**Билет №1**

**Определение ЦА. Основные понятия теории автоматов: ЦА конечные, синхронные, асинхронные, идеализированные, абстрактные, структурные. Абстрактная и структурная теория автоматов.**

ЦА - устройство, предназначенное для преобразования цифровой информации, способное переходить под воздействием входных сигналов из одного состояния в другое и выдавать выходные сигналы.

ЦА конечны, когда множество входных и выходных сигналов, а также число входных и выходных каналов и множество состояний автомата конечны.

Синхронный ЦА – входные сигналы действуют в строго определенные моменты времени при Т=конст, определяемые генератором синхронизирующих импульсов, в которые возможен переход автомата из одного состояния в другое.

Асинхронный ЦА – Т <> конст и определяется моментами поступления входных сигналов, а переход автомата из одного состояния в другое осуществляется при неизменном состоянии входа.

Идеализированный ЦА – Не учитываются переходные процессы в элементах схемы автомата, разница в фактических величинах Т для правильного функционирования автомата не имеет значения, поэтому для описания законов функционирования ЦА вводят абстрактное время, принимающее целые неотрицательные значения.

Абстрактный ЦА - шестикомпонентный вектор S = {A,z,w,σ,λ,a1}, у которого: А- множество состояний автомата, Z – входные сигналы, W- выходные сигналы, σ- функция переходов, λ- функция выходов, а1 – начальное состояние автомата.

Структурный ЦА – учитывает структуру входных и выходных сигналов, а также его внутреннее устройство на уровне структурных схем.

Структурная теория ЦА изучает общие приемы построения структурных схем автоматов на основе элементарных автоматов.

Абстрактная теория ЦА – изучаются наиболее общие законы их поведения без учета конечной структуры автомата и физической природы информации.

**Билет №2**

**Варианты ЦА: автоматы Мили и Мура, С-автомат, автомат без памяти, автономный автомат, автомат без выхода, управляющие и операционные автоматы, микропрограммные автоматы.**

Автомат Мили – a(t+1) = σ (a(t), z(t)); w(t) = λ (a(t), z(t)); a(0) = a1, t= 0,1,2,...

Автомат Мура – a(t+1) = σ (a(t), z(t)); w(t) = λ (a(t)); a(0) = a1, t= 0,1,2,...

С-автомат: под абстрактным С-автоматом понимают математическую модель цифрового устройства, определяемую восьмикомпонентным вектором S = {A,Z,W,U,σ,λ1, λ2,a1}, где А- множество состояний, Z- входной алфавит, W- выходной алфавит автомата Мили, U- выходной алфавит автомата Мура, σ- функция переходов автомата, λ1- функция выходов автомата Мили, λ2- функция выходов автомата Мура, а1 – начальное состояние.

Автомат без памяти(КС): Алфавит состояний такого автомата содержит единственную букву, поэтому понятие функции переходов вырождается и становится ненужным для описания работы автомата, т.е. выходной сигнал в данном такте зависит только от входного сигнала того же такта и никак не зависит от ранее принятых сигналов.

Автономный автомат: В таком автомате входной алфавит состоит из одной буквы. Автомат задается четырьмя объектами: А, W, σ, λ с возможным выделением начального состояния а1. Если автомат конечен и число его состояний равно к, то среди значений а(1), А(2),…, а(к) найдутся повторяющиеся состояния. АА используются для построения генераторов периодических последовательностей, генераторов синхросерий и в других задающих устройствах.

Автомат без выхода: В таком автомате выходной алфавит содержит только одну букву. Автомат задается тремя объектами: А, Z, σ. Из функций, задающих поведение автомата, сохраняется лишь функция переходов.

Управляющие и операционные автоматы: ОА реализует действия над исходной информацией (словами), т.е. является исполнительной частью операционного устройства, а УА управляет работой ОА, т.е. вырабатывает необходимые последовательности управляющих сигналов в соответствии с алгоритмом выполняемой операции. УА используются не только в операционных устройствах вычислительной техники в системе УА-ОА, но и в разнообразных системах автоматики по управлению технологическими процессами и объектами.

Микропрограммные автоматы: Алгоритм записываемый с помощью микроопераций и логических условий, называется микропрограммой.

**Билет №3**

**Автоматы Мили и Мура. С-автомат. Законы функционирования. Основные различия.**

Автомат Мили – a(t+1) = σ (a(t), z(t)); w(t) = λ (a(t), z(t)); a(0) = a1, t= 0,1,2,...

Автомат Мура – a(t+1) = σ (a(t), z(t)); w(t) = λ (a(t)); a(0) = a1, t= 0,1,2,...

Эти автоматы различаются способом определения выходного сигнала. В автомате Мили функция λ определяет выходной сигнал в зависимости от состояния автомата и входного сигнала в момент времени t, а в автомате Мура накладываются ограничения на функцию λ, заключающиеся в том, что выходной сигнал зависит только от состояния автомата и не зависит от значения входных сигналов. Выходные сигналы ЦА Мура отстают на один такт от выходных сигналов ЦА Мили, эквивалентного ему.

С-автомат: под абстрактным С-автоматом понимают математическую модель цифрового устройства, определяемую восьмикомпонентным вектором S = {A,Z,W,U,σ,λ1, λ2,a1}, где А- множество состояний, Z- входной алфавит, W- выходной алфавит автомата Мили, U- выходной алфавит автомата Мура, σ- функция переходов автомата, λ1- функция выходов автомата Мили, λ2- функция выходов автомата Мура, а1 – начальное состояние.

a(t+1) = σ (a(t), z(t)); w(t) = λ 1(a(t), z(t)); u(t) = λ (a(t)); a(0) = a1, t= 0,1,2,...

Отличие С-автомата в том, что он одновременно реализует две функции выходов λ1 и λ2, каждая из которых характерна для модели Мили и модели Мура в отдельности. От С-автомата легко перейти к автоматам Мили и Мура с учетом возможных сдвигов выходных сигналов на такт, аналогично тому, как возможен переход от автомата Мили к автомату Мура и наоборот. Много реальных автоматов работает по модели С-автомата.

**Билет №4**

**Системы канонических уравнений (СКУ) и системы выходных функций (СВФ). Построение СКУ И СВФ для автоматов Мили и Мура.**

СКУ и СВФ являются аналитической интерпретацией таблиц переходов и выходов или графов ЦА. СКУ определяет функции переходов ЦА, а СВФ определяет функции выходов ЦА.

Каждое состояние ЦА интерпретируется как событие, соответствующее множеству переходов в это состояние: As(t+1) = v Zf(t)&Am(t). Для сокращения записи СКУ и СВФ опускают знаки конъюнкции и времени t в правой части уравнения.

Для автомата Мили: СКУ: a1(t+1) = a1z1 v a2z2 v a4z2; a2(t+1) = a1z2; a3(t+1) = a3z1 v a4z2; a4(t+1) = a2z1 v a3z2

CВФ: w1(t) = a1z1 v a1z2; w2(t) = a2z2; w3(t) = a3z1 v a4z2; w4(t) = a4z1; w5(t) = a4z1 v a3z2

Для автомата Мура: a1(t+1) = a2z1 v a3z1; a2(t+1) = a4z2; a3(t+1) = a6z1; a4(t+1) = a3z2 v a2z2 v a1z2; a5(t+1) = a5z1 v a6z2; a6(t+1)= a4z1 v a5z2

СВФ: w1(t) = a1 v a4; w2(t) = a1; w3(t) = a5; w4(t) = a3; w5(t) = a6

**Билет №5**

**Задание ЦА на стандартных языках: таблицы, графы и их аналитическая интерпретация – СКУ и СВФ. Условия однозначности и полной определенности.**

Стандартные (автоматные) языки задают функции переходов и выходов в явном виде. Для того, чтобы задать автомат, необходимо описать все компоненты вектора S = {A,z,w,σ,λ,a1}.

Табличный способ: *автомат Мили* описывается с помощью двух таблиц: таблицы переходов и таблицы выходов. Таблица переходов задает функцию σ, таблица выходов – λ. Каждому столбцу таблицы поставлено в соответствие одно состояние из множества А, каждой строке – один входной сигнал из множества Z. На пересечении строки и столбца в таблице переходов, записывается состояние as, в которое должен перейти автомат из состояния am, под действием входного сигнала zf, т.е. as = σ(am, zf). На пересечении строки и столбца в таблице выходов записывается выходной сигнал wg, выдаваемый автоматом в состоянии am при поступлении на его вход сигнала zf, т.е. wg = λ(am, zf). Автомат Мура задается одной отмеченной таблицей переходов, в которой каждому столбцу приписаны не только состояния am, но еще и выходной сигнал wg, соответствующий этому состоянию, где wg = λ(am).

Граф: Ориентированный граф, вершины которого соответствуют состояниям, а дуги – переходам между ними. Дуга, направленная из вершины am в вершину as, задает переход в автомате из состояния am в состояние as.

СКУ и СВФ являются аналитической интерпретацией таблиц переходов и выходов или графов ЦА. СКУ определяет функции переходов ЦА, а СВФ определяет функции выходов ЦА.

Каждое состояние ЦА интерпретируется как событие, соответствующее множеству переходов в это состояние: As(t+1) = v Zf(t)&Am(t). Для сокращения записи СКУ и СВФ опускают знаки конъюнкции и времени t в правой части уравнения.

Условие однозначности: означает, что для любой пары (am, zf) задано единственное состояние перехода as и единственный выходной сигнал wg, выдаваемый на переходе.

Условие полной определенности: означает, что для всех возможных пар (am, zf) всегда указано состояние перехода и выходной сигнал.

**Билет №6**

**Задание ЦА на языке граф схем алгоритмов (ГСА) и построение на его основе СКУ и СВФ.**

ГСА – ориентированный связный граф, содержащий одну начальную (А0), одну конечную (Ак) и произвольное конечное множество условных Х={x1,...,xl} и операторных А = {А1,…,Ам} вершин. Любой алгоритм должен начинаться и заканчиваться символами начальной и конечной вершин. Начальная вершина не имеет входов, конечная – выходов. Конечная, операторная и условная вершины имеют по одному входу, причем входящая линия может образовываться слиянием нескольких линий. Начальная и операторная вершины имеют по одному выходу, а условная – два выхода, помеченных символами 1 и 0. Операторной вершине сопоставляется вполне определенный оператор Ам, символизирующий некоторые действия Yt. Yt – подмножество множества Y={y1, y2,..., yn}, называемого множеством микроопераций. Разрешается в различных операторных вершинах запись одинаковых подмножеств множества Y. Внутри условных вершин записывается один из элементов множества X={x1, x2, ..., xl}, называемого множеством логических условий. Разрешается в различных условных вершинах запись одинаковых элементов множества Х.

**Билет №7**

**Задание ЦА на языке логических схем алгоритмов (ЛСА) и построение на его основе СКУ и СВФ.**

Язык ЛСА является аналитической интерпретацией языка ГСА и может быть использован для более компактной формы записи алгоритма функционирования ЦА. Запись алгоритма на языке ЛСА представляет собой конечную строку, состоящую из символов операторов А = {A0, A1,...,Ak}, логических условий X={x1,...,xl} и верхних и нижних стрелок, которым приписаны натуральные числа. В некоторых случаях используются логические, которые всегда принимают нулевое значение, т.е. тождественно ложные логические условия ω (оператор ω). После оператора ω всегда производится переход по стрелке, которая стоит справа от него. Если в ЛСА имеются циклы из логических условий, то вводится пустой оператор Ae(Ye), отмеченный пустым выходным сигналом. Этот оператор помещают в ЛСА путем замены стрелки i, стоящей в начале петли из логических условий на следующую группу символов ЛСА: ω↑k↓iAe(Ye)↓k.

**Билет №12**

**Минимизация полностью определенных автоматов Мили методом Ауфенкампа и Хона. Задача минимизации. Алгоритм. Пример.**

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

*Состояния am и as являются эквивалентными, если λ(am, ξ) = λ(as, ξ) для всевозможных входных слов ξ.*

Алгоритм: 1. Находим последовательные разбиения п1, п2, …, пк, пк+1 множества А на классы одно-, двух-, К-, К+1- эквивалентных состояний до тех пор, пока в каком-то (К+1) шаге не окажется, что пк = пк+1.

2. В каждом классе эквивалентности разбиения п выбирается по одному состоянию, в результате чего получаем множество А’ состояний минимального автомата S’ = {A’,z,w,σ’,λ’,a1’} эквивалентному автомату S.

3. Для определения функции переходов и выходов автомата S’ в таблице переходов и выходов вычеркиваются столбцы, соответствующие не вошедшим в А’ состояниям. В оставшихся столбцах не вошедшие в множество А состояния заменяются на эквивалентные.

4. В качестве начального состояния а1’ выбирается состояние из множества А’, эквивалентное состоянию а1. В частности, удобно за а1’ принимать само состояние а1.

**Билет №13**

**Минимизация полностью определенных автоматов Мура методом Ауфенкампа и Хона. Задача минимизации. Алгоритм. Пример.**

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

*Состояния am и as являются эквивалентными, если λ(am, ξ) = λ(as, ξ) для всевозможных входных слов ξ.*

Алгоритм: При минимизации полностью определенных автоматов Мура вводится понятие 0-эквивалентности состояний и разбиение множества состояний на 0-эквивалентные классы. 0-эквивалентными являются одинаково отмеченные состояния. Если два состояния автомата Мура 0-эквивалентны и под действием одинаковых входных сигналов попадают в 0-эквивалентные состояния, то они называются 1-эквивалентными. Все дальнейшие классы эквивалентности для автомата Мура определяются аналогично, как для автомата Мили

1. Находим последовательные разбиения п1, п2, …, пк, пк+1 множества А на классы одно-, двух-, К-, К+1- эквивалентных состояний до тех пор, пока в каком-то (К+1) шаге не окажется, что пк = пк+1.

2. В каждом классе эквивалентности разбиения п выбирается по одному состоянию, в результате чего получаем множество А’ состояний минимального автомата S’ = {A’,z,w,?’,?’,a1’} эквивалентному автомату S.

3. Для определения функции переходов и выходов автомата S’ в таблице переходов и выходов вычеркиваются столбцы, соответствующие не вошедшим в А’ состояниям. В оставшихся столбцах не вошедшие в множество А состояния заменяются на эквивалентные.

4. В качестве начального состояния а1’ выбирается состояние из множества А’, эквивалентное состоянию а1. В частности, удобно за а1’ принимать само состояние а1.

**Билет №14**

**Алгоритм минимизации ЦА Мили с помощью таблицы пар. Задача минимизации. Алгоритм. Пример.**

Алгоритм:

1. Находят одноэквивалентные разбиения состояний автомата

2. Строим таблицу пар. Строки таблицы пар помечены парами одноэквивалентных состояний, столбцы – входными сигналами. На пересечении строк и столбцов в таблице пар записываются пары состояний, которые являются первоприемниками по отношению к конкретному входному сигналу.

3. Последовательно по строкам отыскиваются отличающиеся пары состояний, которые отсутствуют в первом основном столбце таблицы пар. Если в какой-либо строке имеется хотя бы одна такая пара, то в этой строке зачеркивается пара в первом столбце. Такие строки, в которых зачеркнуты пары в первом столбце, называются выделенными строками.

4. Находятся невыделенные строки, в которых имеются пары, вычеркнутые в первом столбце на предыдущем этапе. Если такие строки имеются, то для них зачеркиваются пары в первом столбце. Такой процесс повторяется до тех пор, пока на очередном этапе не обнаруживаются невыделенные строки, в которых имеются пары, вычеркнутые в первом столбце на предыдущем этапе.

5. Оставшиеся незачеркнутые пары в первом столбце таблицы образуют все пары эквивалентных состояний.

**Билет №15**

**Алгоритм минимизации ЦА Мура с помощью таблицы пар. Задача минимизации. Алгоритм. Пример.**

Алгоритм:

1. Находят 0-эквивалентные разбиения состояний автомата

2. Строим таблицу пар. Строки таблицы пар помечены парами одноэквивалентных состояний, столбцы – входными сигналами. На пересечении строк и столбцов в таблице пар записываются пары состояний, которые являются первоприемниками по отношению к конкретному входному сигналу.

3. Последовательно по строкам отыскиваются отличающиеся пары состояний, которые отсутствуют в первом основном столбце таблицы пар. Если в какой-либо строке имеется хотя бы одна такая пара, то в этой строке зачеркивается пара в первом столбце. Такие строки, в которых зачеркнуты пары в первом столбце, называются выделенными строками.

4. Находятся невыделенные строки, в которых имеются пары, вычеркнутые в первом столбце на предыдущем этапе. Если такие строки имеются, то для них зачеркиваются пары в первом столбце. Такой процесс повторяется до тех пор, пока на очередном этапе не обнаруживаются невыделенные строки, в которых имеются пары, вычеркнутые в первом столбце на предыдущем этапе.

5. Оставшиеся незачеркнутые пары в первом столбце таблицы образуют все пары эквивалентных состояний.

**Билет №16**

**Синтез автоматов без памяти. Основные понятия: КС, логический элемент, функциональная схема, базис. Задачи анализа и синтеза комбинационных логических схем (КЛС). Примеры.**

Реализуемый в этих автоматах способ обработки информации называют комбинационным, а сами автоматы без памяти – комбинационными схемами. КС состоит из логических элементов и реализует булеву функцию или совокупность булевых функций.

Логический элемент: простейшая функциональная единица ЭВМ, реализующая одну элементарную булеву функцию. ЛЭ характеризуются определенными техническими параметрами: а) Коэффициент объединения по входу, показывающий какое число входов имеет логический элемент б) Коэффициент разветвления по выходу характеризующий количество входов логических элементов в) Среднее время задержки распространения сигнала в логическом элементе.

Базис: (совокупность) элементов, выбранных для синтеза КС, всегда должен быть функционально полным, т.е. допускать реализацию любой булевой функции на основе принципа суперпозиции.

Задача анализа: заданной КС сводится к отысканию булевой функции или системы булевых функций, описывающих работу этой схемы, с помощью аппарата алгебры логики.

Задача синтеза: КС состоит в построении оптимальной схемы проектируемого узла устройства, исходя из физического описания его работы.

**Билет №17**

**Основные этапы проектирования автоматов без памяти – КЛС. Критерии качества технической реализации КЛС: сложность оборудования (цена схемы), быстродействие, надежность, минимум применяемых элементов. Пример синтеза КЛС.**

Основные этапы синтеза: 1. Анализ технического задания и составление таблицы истинности.

2. Минимизация логических функций.

3. Преобразование минимальных логических функций для рациональной реализации логической схемы в заданном базисе.

4. Построение функциональной схемы.

5. Проверка работоспособности схемы и ее корректировка.

Критерии качества технической реализации: Сложность (цена) схемы по Квайну: Определяется суммарным числом входов логических элементов в составе схемы.

Быстродействие: Оценивается максимальной задержкой распространения сигнала при прохождении его от входа схемы к выходу.

Надежность: Оценивается интенсивностью отказов: λ = n/N\*t, где n – количество элементов, вышедших из строя за период испытаний t, N- общее количество элементов.

**Билет №18**

**Синтез КЛС в булевом базисе, базисах И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ. Правила преобразования для рациональной реализации. Пример.**

Задача синтеза схемы состоит в преобразовании описывающих ее логических функций в суперпозицию логических элементов заданного типа.

И,ИЛИ,НЕ: В этом случае функция представляется в виде суперпозиции операторов логических элементов И (конъюнкторов), это a1, b1, c1, d1, и оператора логического элемента ИЛИ

И-НЕ: Для реализации исходной булевой функции на элементах И-НЕ необходимо от МДНФ функции взять двойное отрицание и одно из них раскрыть по правилу де Моргана, избавляясь от дизъюнкции между элементарными конъюнкциями.

В этом случае функция представлена в виде суперпозиции только операторов И-НЕ

ИЛИ-НЕ: Для реализации исходной булевой функции на элементах ИЛИ-НЕ необходимо от МКНФ функции взять двойное отрицание и одно из них раскрыть по правилу де Моргана, избавляясь конъюнкции между элементарными дизъюнкциями.

В этом случае функция представлена в виде суперпозиции только операторов ИЛИ-НЕ.

И-ИЛИ-НЕ: Для построения схемы необходимо получить МКНФ в виде МДНФ отрицания функции и в случае необходимости преобразовать.

**Билет №19**

**Дешифраторы: определение, условное графическое обозначение, табличное и аналитическое описание. Синтез КЛС на основе дешифраторов. Примеры.**

Дешифратором называется КС, имеющая n входов и 2 в степени n выходов, осуществляющая преобразование входного двоичного n-разрядного кода в сигнал на одном из выходов. Различают полные и неполные дешифраторы. Число выходов полного Д. N = 2n , неполного – N < 2n.

Аналитическое описание: Yi = αi \* E, i = 0, 1, ..., n, где αi – i-й минтерм n входных переменных; Е – сигнал, разрешающий дешифрирование.

Синтез КЛС на основе дешифратора: Для получения схемы достаточно определить выходы дешифратора, соответствующие входящим в функцию конституентам единицы и соединить их с входами дизъюнктора. Если на входы дешифратора будут поданы входные переменные, то на выходе дизъюнктора сформируется значение функции.

**Билет №20**

**Мультиплексоры: определение, условное графическое обозначение, табличное и аналитическое описание. Синтез КЛС на основе мультиплексоров. Примеры.**

Мультиплексор – адресный коммутатор, который может выполнить коммутацию на выход сигнала с того информационного входа, адрес которого задан сигналами на адресных входах.

Аналитическое описание: Y = v Xi αiE, i = 0, 1, ..., 2n- 1, где αi – минтерм (конституента 1), соответствующий i – му адресному набору. Данная функция далее может быть реализована в заданном базисе элементов.

Синтез КЛС на основе мультиплексоров: Мультиплексор можно использовать для преобразования параллельной информации в последовательную, если последовательно задавать адреса разрядов кода числа. Мультиплексор на большое число входов, как правило, приходится строить из мультиплексоров меньшей размерности. «8-1» На адресные входы подаются входные переменные, а информационные входы, соответствующие входящим в функцию конституентам единицы, соединяются с шинами питания, остальные инф. входы соединяются с шинами земли. На выходе мультиплексора формируется значение функции. «4-1» В качестве управляющих сигналов используются переменные, которые подаются на адресные входы мультиплексора. На инф. входы поступают переменные.

**Билет №21**

**Задача структурного синтеза автоматов с памятью. Канонический метод структурного синтеза. Теорема о структурной полноте. Структурная схема С-автомата.**

Задача структурного синтеза: В общем случае задача структурного синтеза автоматов с памятью сводится к нахождению общих приемов построения структурной схемы полученного на этапе абстрактного синтеза автомата Мили, Мура или С-автомата на основе композиции некоторых элементарных автоматов специального вида.

Канонический метод синтеза позволяет свести задачу структурного синтеза произвольного автомата с памятью к задаче синтеза КС. Он оперирует с элементарными автоматами, разделяющимися на два больших класса. Первый класс составляют элементарные автоматы с памятью, называемые элементами памяти. Второй класс составляют элементарные комбинационные автоматы – логические элементы.

Теорема о структурной полноте: Всякая система элементарных автоматов, которая содержит автомат Мура, обладающий полной системой переходов и выходов, и какую-либо функционально-полную систему логических элементов, является структурно полной (Глушков В.М.).

**Билет №22**

**Основные этапы канонического метода структурного синтеза автоматов с памятью. Особенности синтеза автоматов Мили и Мура. Пример.**

Основные этапы канонического метода: 1. Переход на структурный уровень, т.е. кодирование входных, выходных сигналов и состояний автомата. При этом каждая буква алфавитов A,Z,w,u кодируется двоичными векторами (наборами), длина которых равна числу физически реализованных входных и выходных каналов ( для z,w,u) и числу элементов памяти. При этом две различные буквы одного и того же алфавита должны кодироваться различными двоичными векторами.

2. Выбор элементов памяти автомата. Для правильной работы схемы нельзя допустить, чтобы сигналы на входах элементов памяти участвовали в формировании сигналов, которые по цепям обратной связи подавались бы в тот же самый момент времени на эти входы. В связи с этим элементами памяти должны быть не автоматы Мили, а автоматы Мура, имеющие полную систему переходов и полную систему выходов.

3. Выбор функционально полной системы логических элементов. Система логических элементов должна быть функционально полной.

4. Построение булевых ф-ций возбуждения памяти и ф-ции выхода ( СКУ и СВФ). Получение булевых ф-ций выходов не зависит от типа используемых элементов памяти и может быть сделано непосредственно по структурной таблице выходов автомата. Для построения ф-ций возбуждения памяти, в начале строится таблица ф-ции возбуждения памяти по которой записываются канонические уравнения ф-ции возбуждения. Для построения этой таблицы используется структурная таблица переходов автомата и ф-ция входов, используемого триггера.

5. Построение функциональной схемы автомата. Полученная на этапе 4 СКУ и СВФ, преобразуется для рациональной реализации в выбранном базисе и строится функциональная схема структурного автомата.

Особенности синтеза автоматов Мили и Мура**:**

**Билет №24**

**Гонки в ЦА. Аппаратные и логические методы устранения гонок.**

Если при переходе автомата из одного состояния в другое должны изменить свои состояния сразу несколько запоминающих элементов, то между ними начинаются состязания, или гонки. Тот элемент, который выиграет эти состязания, т.е. изменит свое состояние ранее, чем другие элементы, может через цепь обратной связи изменить сигналы на входах некоторых запоминающих элементов до того, как другие, участвующие в состязаниях элементы изменят свои состояния. Это может привести к переходу автомата в состояние, не предусмотренное законом функционирования. Гонки в автомате связаны с разбросом во временных параметрах сигналов, проходящих через логические и запоминающие элементы, и имеют место в любой реальной логической схеме. Для обеспечения заданного закона функционирования автомата необходимо исключить возможность появления критических гонок.

*Аппаратные методы:*

Импульсная синхронизация: гонки устраняются путем ограничения длительности сигнала с, поступающего в цепь синхронизации.

Использование Двойной (двухступенчатой) памяти: Заключается в разделении во времени процессов выработки сигналов возбуждения и процесса переключения состояний.

*Логические методы*:

Соседнее кодирование: Способ кодирования состояний, при котором соседние состояния автомата кодируются наборами, различающимися состоянием только одного элемента памяти.