**МОЙ ШИФР 01076**

**ПРЕОБРАЗОВАНИЕ ЛОГИЧЕСКИХ ФУНКЦИЙ**

**1.1. Основные положения алгебры логики**

Практически все устройства железнодорожной автоматики предназначены для контроля событий и выдачи управляющих сигналов при выполнении определенных условий. При этом принято оценивать наличие факта события логической единицей (1), а отсутствие его – логическим нулем (0).

Соотношения между различными событиями представляются в виде логических функций, записываемых в выражениях алгебры логики, особенностью которых является то, что задаваемые в них переменные (аргументы) и результаты выполняемых над ними операций (функции) могут принимать только два значения – 0 или 1.

Основными операциями алгебры логики являются:

1) логическое сложение (дизъюнкция), обозначаемое знаками \* и +;

2) логическое умножение (конъюнкция), обозначаемое знаками \*, &,• или \* , применяемыми при записи логических функций;

3) логическое отрицание (инверсия), обозначаемое чертой над выражением аргумента или функции (например, x).

При записи выражений функций алгебры логики (ФАЛ) чаще используются буквы латинского алфавита, а формализация и преобразование связей между логическими переменными осуществляются на основании ее правил и законов.

Правило 1: x. • 0 = 0. Логическое произведение любого аргумента на 0 всегда равно 0.

Правило 2: x • 1 = x. Логическое произведение любого аргумента на 1 всегда равно значению аргумента.

Правило 3: x • x = x. Логическое произведение одинаковых аргументов равно значению аргумента.

Правило 4: x • x = 0. Логическое произведение аргумента на его инверсию равно 0.

Правило 5: x + 0 = x. Логическая сумма аргумента и константы 0 равна значению аргумента.

Правило 6: x + 1 = 1. Логическая сумма аргумента и константы 1 равна 1.

Правило 7: x + x = x. Логическая сумма одинаковых аргументов рав-на значению аргумента.

Правило 8: x + x = 1. Логическая сумма прямого значения аргумента и его инверсии равна 1.

Правило 9: x = x. Двойное инвертирование аргумента не меняет его значения.

Основными законами алгебры логики являются следующие:

1) переместительный: x + y = y + x; x • y = y • x;

2) сочетательный: x + y + z = x + (y + z) = (x + y) + z; x • у • z = x • (y • z) = (x • y) • z;

3) распределительный: x • (y + z) = x • y + x • z; x + y • z = (x+y) • (x+z);

4) поглощения: x + x • y = x; x • (x + y) = x;

5) склеивания: 

6) правило де Моргана: . 

Физическая реализация функций алгебры логики осуществляется на дискретных элементах автоматики, к которым относятся реле, двухпозиционные переключатели и логические элементы (ЛЭ). Операция логического умножения выполняется путем последовательного соединения контактов реле или с помощью логического элемента И (дизъюнктора) (рис. 1.1). Логическое сложение выполняется при параллельном соединении контактов или на логическом элементе ИЛИ (конъюнкторе) (рис. 1.2), а логическое отрицание представляется инверсным контактом реле или выполняется логическим элементом НЕ (инвертором) (рис. 1.3). Функции, выполняемые представленными на этих рисунках элементами, можно записать в виде таблиц соответствия (табл. 1.1 - 1.3).



Функции, выполняемые представленными на этих рисунках элементами, можно записать в виде таблиц соответствия (табл. 1.1 \_ 1.3).



Временные диаграммы работы логических схем И, ИЛИ и НЕ приведены на рис. 1.4 (а, б).



Система ФАЛ {И, ИЛИ, НЕ} является функционально полной, но не минимальной. Минимальной функционально полной системой называется такая система, исключение из которой хотя бы одной функции делает систему неполной. К ним относятся системы функций {И, НЕ } и {ИЛИ, НЕ}. Любую логическую функцию, записанную в выражениях базиса {И, ИЛИ, НЕ}, можно записать и в базисах {И, НЕ}, {ИЛИ, НЕ}. Переход от одного базиса к другому осуществляется с использованием закона двойного инвертирования и правила де Моргана (см. ниже). Применение мини-мальных базисов удобно, поскольку на их основе можно записать ФАЛ любой сложности. Физическая реализация логических функций, записанных в минимальных базисах, осуществляется на универсальных логических элементах И-НЕ (элемент Шеффера) и ИЛИ-НЕ (элемент Вебба). Условные обозначения и временные диаграммы работы этих элементов приведены на рис. 1.5 и 1.6 соответственно.



Соотношения между входными и выходными сигналами элементов И-НЕ и ИЛИ-НЕ приведены в табл. 1.4.



Современные логические микросхемы в основном составлены на ос-нове элементов Шеффера и Вебба, а элементы базиса {И, ИЛИ, НЕ} чаще всего используются в качестве буферов, ключей и элементов с третьим (высокоомным) состоянием.

**1.2. Способы задания функций алгебры логики**

Функция алгебры логики считается полностью определенной, если заданы ее значения на всех наборах аргументов.

Наиболее распространенными способами задания ФАЛ являются следующие.

1. Алгебраический, при котором функция задается в виде алгебраи-ческого выражения, определяющего порядок выполнения логических опе-раций над переменными алгебры логики:



или



2. Табличный, когда ФАЛ представляется таблицей истинности (соответствия), содержащей 2n строк и (n+1) столбцов при n аргументах функции. В (n+1) столбце проставляются значения функции, соответст-вующие каждому набору (сочетанию) аргументов, записанному в первых n столбцах.

Так, в табл. 1.5 и 1.6 заданы ФАЛ двух и трех аргументов соответственно.

При таком задании номер набора аргументов соответствует двоич-ному числу, составленному из значений аргументов. Например, набор аргументов 101 соответствует числу 5, записанному в двоичной системе счисления, следовательно, этот набор имеет 5-й номер



3. Числовой, когда ФАЛ задается номерами наборов, на которых ее значение равно 1. Так, функцию, заданную табл. 1.5, можно представить в виде



а заданную табл. 1.6 –



В тех случаях, когда значения функции на некоторых наборах не определены или безразличны, в таблице истинности проставляются знаки " ~ ", , или " \* ", а при числовом способе задания не полностью определенной функции указываются множества наборов, на которых ФАЛ принимает значения 1 (обязательные номера), и наборов, на которых функция равна 0 (запрещенные номера) или ее значения не определены (условные номера):



или



В первом случае на наборах аргументов 1, 3, 5, 6 функция f 1 принимает значения 1, а на наборах 2, 4 – значения 0. На остальных (не указанных) наборах 0 и 7 значения функции не определены (безразличны).

Во втором примере обязательными номерами являются 0, 2, 5, условными – 1, 3, запрещенными – 4, 6, 7.

Эти же функции представлены таблицей истинности (табл. 1.7).



4. Координатный, при котором ФАЛ задаётся в виде координатной карты состояний (карты Карно), содержащей 2n клеток, определяемых пересечениями строк и столбцов, соответствующими определенным наборам аргументов. Так,  (табл. 1.7) могут быть представлены картами Карно (рис. 1.7):



Координатными картами удобно задавать логические функции не более чем от пяти аргументов. В частности, карта Карно для функции аргументов а, в, с, d ( аргумент а является старшим) показана на рис. 1.8. Для примера в клеточки вписаны номера наборов. При заполнении такой карты в эти клетки записываются соответствующие наборам аргументов значения ФАЛ.

**1.3. Канонические формы представления функций алгебры логики**

Каноническими формами представления логических функций являются совершенные дизъюнктивная (СДНФ) и конъюнктивная (СКНФ) нормальные формы.

Дизъюнктивной нормальной формой (ДНФ) называется такая форма ФАЛ, при которой функция записывается в виде дизъюнкции простых конъюнкций прямых или инверсных аргументов:

 

Если в каждой конъюнкции ДНФ присутствуют все аргументы функции, то такая форма носит название совершенной:



Для перехода от ДНФ к СДНФ необходимо в каждую конъюнкцию ДНФ ввести недостающие аргументы путём умножения её на выражение вида 



СДНФ может быть получена непосредственно из таблицы истинности, если представить в виде конъюнкций наборы аргументов, на которых функция равна 1, и объединить их знаками дизъюнкции.

Так, из табл. 1.5 можно записать:



а из табл. 1.6 аналогично:



Конъюнктивной нормальной формой (КНФ) называется форма представления функции в виде конъюнкции простых дизъюнкций прямых или инверсных аргументов:



Если в каждой дизъюнкции КНФ содержится полное количество аргументов ФАЛ, то такая форма называется совершенной:



Для перехода от КНФ к СКНФ необходимо к каждой дизъюнкции с недостающими аргументами прибавить выражения вида, где xi – недостающий аргумент:



Для получения СКНФ из таблицы зависимости необходимо представить в виде инверсий наборы аргументов, на которых функция принимает значение 0, и объединить их знаками конъюнкции.

Так, из табл. 1.5 можно получить:



а из табл. 1.6 аналогично:



По каноническим формам (СДНФ и СКНФ) могут быть построены логические устройства, но, как правило, схемы в этих случаях содержат избыточное количество элементов и, прежде чем их составить, функции должны быть упрощены (минимизированы).

**1.4. Методы минимизации и минимальные формы функций алгебры логики**

При минимизации функций алгебры логики, заданных перечисленными выше способами, могут быть использованы методы: алгебраический (последовательный), Карно и Квайна. Алгебраический метод минимизации основан на применении законов алгебры логики к преобразованию выражения функции с целью получения минимальной формы. При этом первоначальная запись ФАЛ может быть любой. Методы Карно и Квайна используются при минимизации совершенных форм ФАЛ, и если они записаны в сокращенных, но не минимальных выражениях, то прежде, чем использовать тот или другой метод, необходимо привести заданную функцию к СДНФ или СКНФ рассмотренными в подразделе 1.3 методами.

Порядок применения всех этих методов описан в [2 – 4]. В качестве примера рассмотрим минимизацию функции трех аргументов:



При алгебраическом методе минимизации данную функцию можно преобразовать путем вынесения за скобки общих конъюнкций и проведения операций склеивания.



Если функция задана в СКНФ, то ее минимизация проводится с использованием распределительного закона:



К алгебраическим относятся методы минимизации с помощью формул разложения первого и второго рода [3].

Формула разложения первого рода основана на равенстве:



а формула второго рода – на равенстве



Формулы разложения используют при минимизации сложных скобочных форм ФАЛ, причем формула первого рода применяется для минимизации функций, имеющих характер дизъюнктивных форм, а второго рода – конъюнктивных.

Рассмотрим примеры минимизации вышеприведенных функций с помощью формул разложения. Минимизация функции СДНФ проводится по формуле разложения первого рода:



а функция СКНФ минимизируется по формуле разложения второго рода:



При использовании метода Карно следует заданную ФАЛ представить координатной картой и провести операции склеивания путем объединения в замкнутые области значений функции, равных 1 или 0, и исключить из выражения функции аргументы, изменяющие свои значения в пределах выделенных областей. Функция запишется в минимальной дизъюнктивной нормальной форме (МДНФ), если операции склеивания проводились с наборами аргументов, на которых функция равна 1, и в минимальной конъюнктивной нормальной форме (МКНФ) при проведении операций склеивания с наборами, на которых функция равна 0.

Следует помнить, что в одну область объединяются 2k клеток, где k = 1, 2, 3 и т. д.

Так, приведенная в предыдущем примере функция может быть представлена координатным способом в виде карты Карно, на которой показаны способы проведения операций склеивания (рис. 1.9). Минимизированные выражения функций будут соответствовать тем, которые получены при алгебраическом методе.



Так, приведенная в предыдущем примере функция может быть представлена координатным способом в виде карты Карно, на которой показаны способы проведения операций склеивания (рис. 1.9). Минимизированные выражения функций будут соответствовать тем, которые получены при алгебраическом методе.

Метод Квайна заключается в последовательном проведении операций склеивания и поглощения. При этом в результате операций склеивания получаются импликанты – минтермы, если наборы аргументов, над которыми проводятся операции склеивания, представлены в виде конъюнкций, и макстермы, если наборы аргументов представлены в виде дизъюнкций. Импликанты вводятся в состав функции, и проводятся операции поглощения, в результате которых получаются минимальные формы ФАЛ.

Метод Квайна используется при минимизации функций и систем ФАЛ, определенных на всех наборах аргументов без ограничения их числа.

Для приведенной в рассматриваемом примере функции



находятся импликанты , полученные в соответствии с законом склеивания, составляется импликантная таблица (табл. 1.8), по которой проводятся операции поглощения в соответствии с [4], и записывается минимизированная функция. Для функции, записанной в СКНФ, 

импликантами будут , а импликантная таблица составляется в виде табл. 1.9.



Поскольку найденные импликанты "перекрывают" столбцы таблицы в единственном числе, то функция в этом случае запишется так:





Функция, записанная в МКНФ по табл. 1.9, представляется в виде:



В том случае, когда остаются не поглощенными какие-то члены функций, они вводятся в состав минимизированных форм без изменения.

**1.5. Задания к разделу 1**

Вариант задания выбирается из табл. 10 в соответствии с порядковым номером в журнале или по заданию преподавателя. Для студентов заочного отделения вариант выбирается по сумме цифр шифра, если она не превышает 26. В противном случае – по сумме двух последних цифр.



В соответствии с заданием требуется:

1) представить заданную функцию таблицей истинности, СДНФ, СКНФ, координатным способом;

2) минимизировать функцию методами: алгебраическим, Карно, Квайна;

3) записать минимизированные функции в МДНФ и МКНФ;

4) обосновать применяемые методы с точки зрения законов алгебры логики и дать их сравнительные характеристики;

5) сделать выводы о целесообразности применения того или иного метода при различных формах ФАЛ;

6) составить задачу и решить ее наиболее рациональным методом

**2. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ**

**2.1. Синтез комбинационных схем на релейно-контактных элементах**

На релейно-контактных элементах реализуются функции, записанные в базисе {И, ИЛИ, НЕ} в МДНФ и МКНФ, причем прямое значение аргумента представляется замыкающим (фронтовым) контактом, инверсное– размыкающим (тыловым), произведение (конъюнкция) аргументов– последовательным, а сумма (дизъюнкция) – параллельным соединением контактов. Релейно-контактная схема должна иметь столько входов, сколько аргументов содержится в выражении функции. На входах схемы включаются воспринимающие реле, контакты которых осуществляют физическую реализацию ФАЛ.



На рис. 2.1 приведена схема, реализующая функцию



а на рис. 2.2 – функцию



Поскольку в каждой из функций содержится по три аргумента, то в схемах, реализующих эти функции, включены по три воспринимающих реле (А, Б, С). Подача входных сигналов осуществляется при нажатии (замыкании) соответствующих кнопок, что вызывает включение того или иного реле. Выходная функция формируется контактами реле в соответствии с выражениями (2.1), (2.2).

**2.2. Синтез комбинационных схем на логических элементах базиса {И, ИЛИ, НЕ}**

К логическим элементам (ЛЭ) данного базиса относятся элементы И – конъюнктор (схема логического умножения, или схема совпадения), ИЛИ – дизъюнктор (схема логического сложения), НЕ – инвертор (схема логического отрицания). На них может быть реализована любая, сколь угодно сложная логическая функция, записанная в соответствующем базисе.

Применяя элементы, приведенные в подразделе 2.1, выражения ФАЛ могут быть реализованы схемами, представленными на рис. 2.3 и 2.4. При этом схема, приведенная на рис. 2.3, реализует функцию (2.1), а на рис. 2.4 – функцию (2.2).



Поскольку в выражениях функций имеются кроме прямых и инверсные значения аргументов, то для их получения в схемах включены инверторы, а сами функции формируются при помощи логических элементов И, ИЛИ. Принципиальные схемы ЛЭ рассмотрены в [1, 2, 4]. Задавая комбинации значений входных сигналов a, b, с согласно таблице истинности (см. табл. 1.6), представленной в подразделе 1.2, получим соответствующие значения выходной функции f.

**2.3. Синтез комбинационных схем на логических элементах базиса {ИЛИ, НЕ}**

К логическим элементам данного базиса относятся элементы Вебба (ИЛИ-НЕ). На них удобно реализовывать ФАЛ, записанные в КНФ. Для приведения КНФ к базису {ИЛИ, НЕ} функция дважды инвертируется и записывается в выражениях Вебба, а затем реализуется на ЛЭ ИЛИ-НЕ. Так, МКНФ функции (2.2)  приводится к базису {ИЛИ, НЕ} следующим образом:



При инвертировании функции используется правило де Моргана.

Схема, реализующая данное выражение, приведена на рис. 2.5.



**2.4. Синтез комбинационных схем на логических элементах базиса {И, НЕ}**

К ЛЭ базиса {И, НЕ} относятся элементы Шеффера (И-НЕ). На элементах этого типа удобно реализовать ФАЛ, записанные в ДНФ. Для приведения ДНФ к базису {И, НЕ} функция дважды инвертируется и записывается в выражениях Шеффера, а затем реализуется на ЛЭ И-НЕ. Так, функцию можно привести к базису {И, НЕ} следующим образом:



Схема, реализующая данную функцию, приведена на рис. 2.6.

Операции приведения функций алгебры логики к базисам {И, НЕ} и {ИЛИ, НЕ} основаны на законе двойного инвертирования Первая инверсия проводится с целью преобразования выражения ФАЛ таким образом, чтобы оно содержало только те операции, которые реализуются с помощью логических элементов выбранного базиса. Вторая инверсия применяется для того, чтобы сохранить физический смысл функции при записи её в новом базисе. Как видно из примеров, операция инвертирования проводится только один раз, вторая инверсия остаётся нераскрытой.

**2.5. Задания к разделу 2**

Варианты заданий приведены в табл. 2.1.



Выбор варианта производится согласно указаниям, данным в разд. 1

В соответствии с заданием требуется:

1) минимизировать заданную функцию любым методом;

2) записать МДНФ и МКНФ функции;

3) реализовать ФАЛ на релейно-контактных элементах;

4) реализовать функции на ЛЭ всех базисов.

**3. СИНТЕЗ СПЕЦИАЛЬНЫХ КОМБИНАЦИОННЫХ СХЕМ**

**3.1. Синтез схем с несколькими выходами**

Комбинационные схемы с несколькими выходами могут быть заданы в виде системы ФАЛ или таблицей истинности. Минимизация функций может проводиться всеми известными методами, но следует учитывать, что при не полностью определенных функциях удобнее использовать метод Карно, а в тех случаях, когда функции определены на всех наборах аргументов и их количество превышает четыре или пять, целесообразно использвать метод Квайна, при котором можно получить наиболее оптимальную схему, в которой одни и те же элементы используются для формирования нескольких ФАЛ. Сущность метода изложена в [ 3, 4]. Пример минимизации методом Квайна рассмотрим при составлении комбинационной схемы, реализующей систему функций:



На первом этапе минимизации сведем в табл. 3.1 и 3.2 члены СДНФ (конституенты) и найденные импликанты с указанием функций, содержащих названные элементы. Следующим этапом является составление импликантной таблицы (табл. 3.3) и проведение операций поглощения с целью получения функций в МДНФ. В первую очередь в состав ФАЛ вводятся импликанты, составляющие ядро функции, т. е. перекрывающие столбцы таблицы в единственном числе. Затем записываются импликанты, перекрывающие наибольшее количество столбцов, и в последнюю очередь в состав функций включаются оставшиеся не поглощенными конституенты. Используя правила записи минимальных ФАЛ при методе Квайна, получим:



 

Схемы, реализующие системы функций, составляются по изложенным ранее правилам, и на рис. 3.1 приведен пример реализации полученной системы ФАЛ в базисе {И, ИЛИ, НЕ}. При использовании метода Карно для каждой функции составляется координатная карта и проводятся операции склеивания, в результате кото-рых записываются минимальные выражения ФАЛ. В табл. 3.3 представлен метод минимизации функций алгебры логики в СДНФ, но аналогично минимизируются и системы функций алгебры логики в СКНФ (см. табл. 1.9). Метод Квайна является основным при минимизации логических функций с помощью ЭВМ.



**3.2. Синтез шифратора двоичного кода**

Шифратор (кодер) предназначен для кодирования десятичных цифр двоичным кодом и представляет собой комбинационную схему, имеющую 10 входов и количество выходов, соответствующее числу разрядов кода. Двоичные коды, при помощи которых представляются десятичные цифры в устройствах автоматики, связи и вычислительной техники, приведены в табл. 3.4. Определяющей характеристикой каждого кода является его основание, представляющее вес каждого разряда кода или способ его образования. При известном основании кода легко осуществить перевод числа, записанного в двоичной форме, в десятичное. Для этого достаточно каждый двоичный разряд числа заменить значением его веса и провести сложение всех разрядов. Исключение составляют коды с избыточностью, когда каждой десятичной цифре искусственно присваивается определённая кодовая комбинация.



Рассмотрим пример синтеза шифратора кода 8421. Условное обозначение шифратора приведено на рис. 3.2. Таблица истинности (табл. 3.5) для шифратора 8421 составлена в соответствии с таблицей кодов и представляет собой таблицу четырёх ФАЛ (x1 , x2 , x3 , x4 ) от десяти аргументов (y0 , y1 , y2 ,.. y9 ).



Выражения логических функций для выходов шифратора:



Таким образом, получается система уравнений, представляющая собой математическую модель комбинационной системы, которую можно реализовать на любой элементной базе и, в частности, на логических эле-ментах ИЛИ (рис. 3.3).

**3.3. Синтез дешифратора двоичного кода**

Дешифратор (декодер) преобразует двоичный код в десятичное число. Условное обозначение декодера кода 8421 показано на рис. 3.4. Декодер имеет количество входов, соответствующее разрядности дешифри-руемого кода, и десять выходов по числу десятичных цифр.



При поступлении на входы дешифратора кодовой комбинации сигналов на одном из его выходов, номер которого соответствует этой комбинации, появляется уровень логической единицы.

Таблица истинности дешифратора аналогична табл. 3.5 за исключением того, что функциями будут y0 , y1 , y2 ,.. y9, а аргументами x1 , x2 , x3 , x4.

Выражения для выходов дешифратора также записываются в виде системы ФАЛ:



которая может быть реализована на любой элементной базе. В частности, удобно такой дешифратор выполнить на логических элементах ИЛИ-НЕ, для чего необходимо дважды проинвертировать все функции и получить выражения Вебба:



Схема, составленная в соответствии с приведенными выражениями, показана на рис. 3.5.

**3.4. Синтез преобразователя двоичного кода**

Преобразователи кодов предназначены для преобразования одного двоичного кода в другой. Синтез преобразователя кода (ПК) может осу-ществляться двумя способами. При первом способе ПК составляется из дешифратора и шифратора в соответствии со структурной схемой, приве-денной на рис. 3.6.



 Следовательно, задача в этом случае заключается в соединении дешифратора и шифратора в одну схему.

При втором способе ПК составляется, как комбинационная схема с количеством входов, равным разрядности преобразуемого, и числом выходов, равным количеству разрядов нового кода.

Таблица истинности для ПК составляется аналогично табл. 3.4. В качестве аргументов служат значения разрядов преобразуемого, а функциями являются разряды нового кода. В табл. 3.6 приведено соответствие между входами и выходами преобразователя кода 8421 в код 2 из 5.

Минимизированные формы логических функций для выходов ПК получаются с помощью метода Карно, при котором для каждой ФАЛ составляется координатная карта и проводятся операции склеивания. Так, выражение для y1 можно получить, если заполнить карту Карно так, как показано на рис. 3.7.



Проведя операции склеивания, получим Аналогично заполняются карты для всех остальных функций и получаются их выражения в МДНФ или МКНФ:



исходя из которых составляется схема преобразователя кода.

**3.5. Задания к разделу 3**

**З а д а н и е 1**

Минимизировать методом Квайна систему ФАЛ и реализовать ее на релейно-контактных и логических элементах базисов {И, ИЛИ, НЕ}, {И, НЕ}, {ИЛИ, НЕ}.



З а д а н и е 2

 

П р и м е ч а н и е. Схемы шифраторов, дешифраторов и преобразователей кодов составить на элементах Шеффера и Вебба.

**4. СИНТЕЗ КОНЕЧНОГО АВТОМАТА**

**4.1. Способы задания конечных автоматов**

Конечный автомат может быть задан словесным алгоритмом, таблицами переходов (ТП) и выходов (ТВ), графом состояний и системой уравнений, выражающих зависимости между входами, выходами и внутренними состояниями автомата. Последний (алгебраический) способ задания представляет собой математическую модель, по которой составляется схема, реализующая заданный алгоритм. Следовательно, в любом случае конечной формой задания автомата является алгебраическая, а порядок синтеза сводится к получению математической модели и функциональной схемы автомата и состоит из следующих этапов:

1) составление словесного алгоритма работы автомата;

2) определение числа состояний автомата;

3) составление таблиц переходов и выходов;

4) определение количества элементов памяти автомата;

5) кодирование таблиц переходов и выходов;

6) составление таблицы истинности конечного автомата;

7) получение математической модели автомата;

8) составление функциональной схемы;

9) составление принципиальной схемы.

**4.2. Синтез асинхронного конечного автомата**

**4.2.1. Кодирование асинхронного автомата**

При синтезе асинхронного автомата необходимо решить вопрос исключения критических состязаний элементов памяти (ЭП). Наиболее распространенными способами, предполагающими исключение критических состязаний в процессе синтеза, являются методы кодирования таблиц переходов таким образом, чтобы при функционировании автомат не смог оказаться в незаданных по условиям переходов состояниях. Универсальным является метод кодирования ТП по столбцам [2], использование которого рассмотрим на примере синтеза автомата, заданного ТП (табл. 4.1) и ТВ (табл. 4.2). При этом метода вводится понятие -класса, представляющего собой множество, включающее устойчивое и все неустойчивые состояния, из которых заданы переходы в данное устойчивое состояние. Критические состояния возникают в том случае, когда схема в результате состязаний ЭП попадает вместо одного устойчивого состояния в другое, т. е. из одного -класса схема ошибочно перейдёт в другой. Для исключения этого явления вводятся переменные, разделяющие классы внутри каждого столбца ТП. Они имеют одинаковое значение в кодах состояний одного -класса и различное для кодов состояний других -классов. Для разделения состояний внутри одного -класса вводятся дополнительные переменные, которые одновременно являются разделяющими для -классов другого столбца.



Согласно используемому методу для первого столбца ТП (табл. 4.1)  , а для второго , где в скобках указаны номера строк с устойчивыми и неустойчивыми состоя-ниями. Необходимое количество элементов памяти определится по формуле:



Таким образом, для реализации автомата требуется два элемента памяти (Y1 и Y2), причём Y1 предназначен для разделения -классов первого столбца, Y2 - второго (табл. 4.3), а состояния автомата закодируются согласно табл. 4.4.



Кодированные ТП и ТВ представляются в виде табл. 4.5 и 4.6 соответственно.



В некоторых случаях, при большем количестве состояний автомата, при кодировании ТП появляются дополнительные (промежуточные) состояния, которые используются для исключения критических состязаний. Подробное описание метода изложено в [2]. Рассмотрим пример кодирования таблицы переходов (табл. 4.7), содержащей основные и дополнительные состояния. Основными являются те состояния, в которые схема автомата должна приходить в соответствии с заданием, а дополнительными - те, в которых схема может оказаться вследствие неодновременного срабатывания ЭП. Как видно из табл. 4.7, переходы из дополнительных состояний заданы таким образом, что в случае состязаний элементов памяти схема не сможет перейти из одного -класса в другой, поскольку из всех основных и дополнительных состояний любого -класса переходы заданы в устойчивое состояние этого класса.



**4.2.2. Синтез релейно-контактного автомата**

Схема релейно-контактного автомата составляется на основании таблицы истинности (табл. 4.8).



Таблица истинности формируется из таблиц переходов и выходов (см. табл. 4.5 и 4.6). Функциями в этих таблицах являются Y1(t), Y2(t), Z(t), а x, Y1(t -1), Y2(t -1) - аргументами указанных функций, причем Y1(t), Y2(t) представляют собой значения внутренних состояний, Z(t) - выходов, x - входа в настоящий момент времени, а Y1(t -1), Y2(t -1) - значения внутренних состояний в предыдущий момент. Математическая модель автомата представляется системой уравнений, полученных с помощью рассмотренных ранее методов:



Схема релейноконтактного автомата, составленная в соответствии с его математической моделью, приведена на рис. 4.1.



На этой схеме реле Х является воспринимающим, реле У1 и У1 - элементами памяти, реализованной с помощью обратных связей (цепей самоблокировки). Функции управления ЭП и выхода формируются контактами указанных реле.

**4.2.3. Синтез автомата на бесконтактных элементах**

В асинхронных автоматах на бесконтактных элементах в качестве ЭП используются RS-триггеры. В отличие от релейноконтактных при синтезе бесконтактных конечных автоматов необходимо получить выражения для функций управления S- и R-входами триггеров. Поэтому ТП автомата на RS-триггерах содержит столбцы для S и R функций, в которые записаны значения входных сигналов, переводящих триггеры из одного состояния в другое. Уровни этих сигналов определяются в соответствии с таблицей переходов триггера (табл. 4.9), в которой Y1(t-1) и Y2(t-1) – состояния прямых выходов триггера в предыдущий, а Y1(t) и Y2(t) – в настоящий момент времени. S1, R1 и S2 , R2– управляющие сигналы, переводящие триггеры из предыдущего состояния в настоящее.



Таблица истинности асинхронного автомата представлена табл. 4.10.



Математическая модель автомата представляется системой функций:



а схема автомата, составленная на основании его математического описа-ния, приведена на рис. 4.2.



**4.3. Синтез синхронного автомата**

В синхронных автоматах в качестве ЭП используются триггеры с синхронизирующими входами. Синхронизация работы функциональных узлов автомата является одним из способов исключения критических состязаний элементов памяти, поскольку синхронизирующие импульсы подаются через интервалы времени, в течение которых завершаются переходные процессы и все элементы системы приходят в устойчивое состояние. Следовательно, при синтезе синхронного автомата ТП кодируется произвольно. Так, таблица переходов (см. табл. 4.1) в отличие от табл. 4.5 представлена кодированной таблицей (табл. 4.11), в которой состояния закодированы двоичными числами, соответствующими порядковым номерам состояний, а таблица выходов представляется табл. 4.12.



Таблица истинности (табл. 4.13), составленная по ТП и ТВ, учитывает то, что в качестве ЭП выбраны JK- триггеры, принципы управления которыми представлены в табл. 4.14.



Математическое описание конечного автомата на JK-триггерах представляется системой функций, полученных из табл. 4.13 методом Карно:



Схема синхронного автомата приведена на рис. 4.3.



Синхронизация входа автомата осуществляется импульсами напряжения Uc1, подаваемыми на синхронизирующий вход триггера Т, а блока памяти – импульсами Uc2, которые подаются на синхронизирующие входы триггеров Т1 и Т2, выполняющих функции элементов памяти. Изменение состояния триггеров возможно только при поступлении синхронизирующих импульсов, подаваемых в те моменты времени, когда все переходные процессы в схеме заканчиваются.

Таким образом повышается устойчивость работы автомата.

**4.4. Задания к разделу 4**

В соответствии с номером варианта составить схемы автоматов, представленных в задании графами, на релейно-контактных элементах, RS- и JK-триггерах. Порядок выполнения задания следующий.

1. В соответствии с графом составить таблицы переходов и выходов.

2. Определить необходимое количество элементов памяти.

3. Закодировать состояния автомата с учётом исключения критиче-ских состязаний элементов памяти.

4. Составить кодированные таблицы переходов и выходов.

5. Составить таблицы истинности для автоматов на релейно-контактных и бесконтактных элементах.

6. Записать функции управления и выходов автомата.

7. Составить функциональные схемы автоматов на релейно-контактных и бесконтактных элементах.

