Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА

Кафедра электрических машин
и электропривода

ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ СИСТЕМ УПРАВЛЕНИЯ

Конспект лекций
для студентов специальности 5В071800 – Электроэнергетика

Алматы 2019

СОСТАВИТЕЛИ: Мустафин М.А., Бестерекова А.Н. Логические основы цифровых систем управления. Конспект лекций по специальности 5В071800. – Алматы: АУЭС имени Гумарбека Даукеева, 2019. – 68 с.

В конспекте лекций кратко изложены основные определения, задачи и области применения алгебры логики, основные логические операции (булевы функции).

Рассмотрены законы и правила булевой алгебры и их применение для преобразования логических функций. Приведены сведения о релейных и бесконтактных логических элементах, применяемых в автоматике и электротехнике, методах минимизации логических функций и схем, использовании логических элементов и схем в электротехнических схемах, в системах автоматического управления, релейной защиты, диагностирования и поиска неисправностей.

Ил. 48, табл. 10, библиогр. – 9 назв.

Рецензент: доцент кафедры ЭСиЭЭС Б.К. Курпенов

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи имени Гумарбека Даукеева» на 2019г.

© НАО «Алматинский университет энергетики и связи имени Гумарбека Даукеева», 2019 г.

Содержание

Лекция № 1. Алгебра логики (булева алгебра). Основные определения задачи и области применения..... 4
Лекция № 2. Основные логические операции (булевы функций)..... 7
Лекция № 3. Преобразование логических функций. Законы и правила булевой алгебры..... 12
Лекция № 4. Релейные и бесконтактные логические элементы автоматики..... 18
Лекция № 5. Логические элементы..... 23
Лекция № 6. Анализ и синтез логических устройств. Методы минимизации логических функций и схем..... 32
Лекция № 7. Анализ и синтез логических устройств. Минимизация количества логических элементов..... 36
Лекция № 8. Дизъюнктивная и конъюнктивная формы логических функций. Стандартная минимизация булевых функций..... 41
Лекция № № 9. Основные принципы проектирования релейных схем в соответствии с заданным алгоритмом их работы..... 44
Лекция № 10 Использования логических элементов в электрических схемах..... 48
Лекция № 11 Примеры использования логических элементов в электрических схемах..... 52
Лекция № 12. Триггеры. RS-триггер..... 55
Лекция № 13. Применение алгебры логики для синтеза САР..... 57
Лекция № 14. Применение алгебры логики в релейной защите..... 59
Лекция № 15. Использование алгебры логики в системах диагностирования и поиска неисправностей..... 62
Список литературы..... 67

Лекция №1. Алгебра логики (булева алгебра). Основные определения, задачи и области применения

Цели лекции:

- освоить основные положения, понятия и определения алгебры логики;
- уяснить возможность использования булевой алгебры в технике и технологиях.

Содержание лекции:

- логические высказывания, истинные и ложные высказывания, их исчисление;
- логические связки «не», «и», «или», «если... , то»;
- логические 1 и 0 в технических приложениях и контакторных схемах.

Логика (от греч. logos — слово, понятие, рассуждение, разум) — наука о законах и операциях правильного мышления. Логика составляет основу всякого управления. Управление отдельными технологическими операциями и сложными производственными процессами происходит по принципу «ЕСЛИ…, ТО…». Например, «ЕСЛИ» температура ниже заданной, «ТО» необходимо включить нагрев», либо «ЕСЛИ» выдал сигнал датчик А «ИЛИ» нажата кнопка В, «ТО» включить устройство Х «И» сигнальную лампу «Y». Когда операции простые, а следствия из них и управляющие воздействия достаточно очевидны, то надобности в какой-либо логической обработке информации и преобразовании логических функций не возникает. Но когда производственные процессы характеризуются большим числом параметров и для управления создается автоматическая система, то необходима логическая обработка поступающей информации.

Еще в античные времена была известна формальная логика, позволяющая судить об истинности (правильности) какого – либо высказывания не по содержанию, а по форме его построения. В 19 веке английский математик Джорж Буль разработал математическую логику, или алгебру логики, в которой каждая переменная может принимать одно из двух значений – «истинно» или «ложно». В дальнейшем алгебра логики явилась также основой для математической теории переключательных систем и последующей разработки цифровых вычислительных машин.

Алгебра логики — это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Так, например, предложение «*Нур-Султан – столица Казахстана»* следует считать высказыванием, так как оно истинное. Предложение «*Муратов А.А. — студент АУЭС»* тоже высказывание, так как оно или

истинно или ложно. «2+2=5» - тоже высказывание, но ложное. Высказываниями не являются предложения:

- ничего не утверждающие («*студент второго курса»*);

- использующие неопределенные понятия («*информатика — интересный предмет»*). Первое предложение ничего не утверждает об ученике, а второе использует слишком неопределённое понятие «*интересный предмет»*;

- вопросительные и восклицательные предложения.

В предложении «*этот параметр равен десяти*» необходимо уточнить название параметра, чтобы определить истинность или ложность высказывания. Такие предложения называются *высказывательными формами*.

Высказывательная форма — это повествовательное предложение, которое прямо или косвенно содержит хотя бы одну переменную и становится высказыванием, когда все переменные замещаются своими значениями.

Алгебра логики рассматривает любое высказывание только с одной точки зрения — является ли оно истинным или ложным. Заметим, что зачастую трудно установить истинность высказывания. Так, например, высказывание «*площадь поверхности Индийского океана равна 75 млн км*^{2}» в одной ситуации можно посчитать ложным, а в другой — истинным. Ложным — так как указанное значение неточное и вообще не является постоянным. Истинным — если рассматривать его как некоторое приближение, приемлемое на практике.

Употребляемые в обычной речи слова и словосочетания «не», «и», «или», «если..., то» и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются *логическими связками.*

Высказывания, образованные из других высказываний с помощью логических связок, называются составными. Высказывания, не являющиеся составными, называются элементарными.

Так, например, из элементарных высказываний «Тимур - электрик», «Тимур - IT специалист» при помощи связки «*и»* можно получить составное высказывание «Тимур – электрик *и* IT специалист», понимаемое, как «Тимур – электрик, знающий IT технологии».

При помощи связки «*или*» из этих же высказываний можно получить составное высказывание «Тимур – электрик или IT специалист», понимаемое в алгебре логики как «Тимур – электрик *или* IT специалист»*, или и* электрик*, и* IT специалист одновременно».

Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через А обозначено высказывание «Тимур - электрик» «Тимур -

электрик», а через В — высказывание «Тимур - IT специалист». Тогда составное высказывание «Тимур – электрик *и* IT специалист» можно кратко записать, как А *и* В. Здесь «и» — логическая связка, А, В — логические переменные, которые могут принимать только два значения — «истина» или «ложь», обозначаемые, соответственно, «1» и «0».

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение, которые будут рассмотрены далее.

Булева алгебра применима к любым переменным, которые могут принимать два и только два значения (включено – выключено, годен – брак, принять – отвергнуть, ДА – НЕТ, и т.д.). Обозначаются эти два значения символами 1 и 0, которые совпадают по написанию с обычными арифметическими единицей и нулем, но только внешне, так как смысл их различен. Логическая 1 не есть одна штука чего – либо реального или абстрактного. В технических и многих других практических приложениях логическая единица – это скорее знак того, что совершилось какое – то событие или ряд событий: замкнулись контакты и по цепи течет ток, заработала аппаратура, принято решение, система выдержала испытания и т.д. В своем первоначальном смысле логическая 1 означала, что какое – то высказывание истинно. В противоположность этому логический 0 означал, что высказывание не соответствует истине, т.е. ложно. В технических приложениях логический 0 означает, что контакты разомкнуты, аппаратура не работает, цепь не проводит, проект отвергнут, система не выдержала испытания и т.д.

В дальнейшем для удобства чтения логические 1 и 0 набраны жирным шрифтом.

По существу, исчисление высказываний представляет собой алгебру двух чисел, т. е. такую алгебру, в которой каждый отдельный аргумент и любая функция могут иметь одно из двух значений.

Исторически первыми устройствами, для описания действий которых использовали логические функции, были устройства, выполненные на релейно-контактных элементах. Для проектирования таких устройств была разработана теория релейно-контактных схем. Затем появились бесконтактные устройства, предназначенные только для логических преобразований сигналов и представляющие собой конструктивно оформленные изделия.

Возможность использования булевой алгебры для преобразования контактных схем определяется тем, что каждый из входящих в структурную формулу аргументов (контактов) может принимать всего два значения, т. е. может быть замкнутым или разомкнутым, и вся функция, представленная структурной формулой, может выражать или замкнутую, или разомкнутую цепь. Лампочка, как простейший приемник электроэнергии, в контактных схемах также может иметь только два состояния: включена или выключена.

На рисунке 1.1 изображены условные обозначения элементов релейных схем. При подаче питания на катушку реле (рисунок 1.1,а), контакты его замыкаются или размыкаются (рисунок 1.1,б).

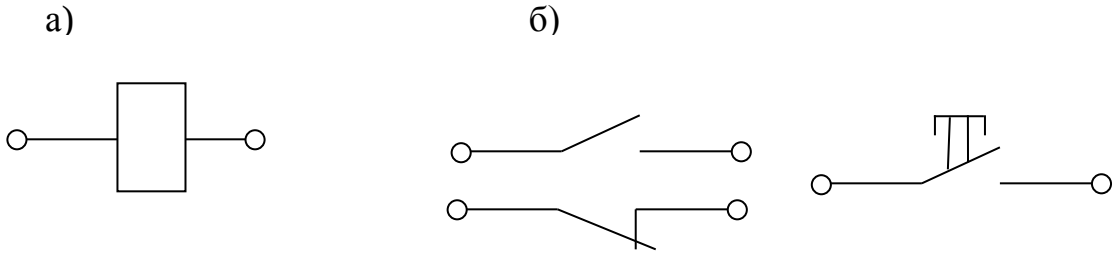


Рисунок 1.1 - Условные обозначения элементов релейных схем

Лекция №2. Основные логические операции (булевы функции).

Цели лекции:
- изучить виды представления булевых функций;
- освоить основные операции булевой алгебры.

Содержание лекции:
- словесное, табличное и алгебраическое описание булевых функций;
- выполнение основных логических операций «ИЛИ», «И», «НЕ»;
- производные функции: импликация и эквиваленция;
- условные обозначения и контакторные аналоги логических элементов.

Результат выполнения логических операций над двоичными переменными называется булевой функцией *F*. Она может принимать только два значения – «0» или «1». Задать булеву функцию – значит указать ее значение при всех возможных комбинациях переменных (аргументов). Если число переменных равно n, то число возможных комбинаций равно 2^n . Когда значение функции известно для всех комбинаций, она называется полностью определенной. В противном случае – частично определенной.

Булевы функции необходимы для синтеза цифровых устройств, содержащих только логические элементы. Для представления булевых функций часто применяют словесное описание, табличное и алгебраическое представление.

Словесное описание функции должно однозначно определять ее значение при различных значениях аргументов.

Табличное представление – это перечисление всех возможных комбинаций входных сигналов (аргументов) и соответсвующих им значений булевой функции. Такая таблица называется таблицей истинности данной функции.

Алгебраическая форма представления булевых функций используется для минимизации (упрощения формул) и для построения логических схем.

В булевой алгебре имеется три основных функции, или операции: «ИЛИ», «И», «НЕ».

1. Операция «ИЛИ».

Результат операции «ИЛИ» равен 0 тогда и только тогда, когда все аргументы операции равны 0, в остальных случаях результат равен 1.

Эта операция является *логическим сложением* и поэтому может обозначаться знаком «+». Ее называют также «дизъюнкцией» (лат. disjunctio — разделение) и иногда обозначают знаком $\vee$. Таким образом, выражения

$$X=A+B \quad \text{и} \quad X=A\vee B$$

эквивалентны и читаются, как «икс равен А или В».

Задавая различные значения переменных, результаты операции «ИЛИ» можно представить в следующем виде:

$$0\vee0 = 0; \quad 0\vee1 = 1; \quad 1\vee0 = 1; \quad 1\vee1 = 1.$$

Используя другое обозначение логического сложения:

$$0+0 = 0; \quad 0+1 = 1; \quad 1+0 = 1; \quad 1+1 = 1.$$

За исключением последнего, все эти выражения являются такими же, как в элементарной алгебре. Результат четвертой операции не совпадает ни с обычным арифметическим сложением, ни со сложением двоичных чисел. Это является следствием того, что логическая 1 является не числом «один», а только символом, смысл которого был пояснен выше. Следует обратить внимание на то, что при сложении двух логических единиц получается логическая единица. Алгебра логики оперирует только двумя значениями - ложью (логический 0) и истиной (логическая 1). Истина не может быть двойной или истиной в квадрате, поэтому при сложении двух истин мы получаем просто истину. Точно также при сложении двух логических сигналов высокого уровня мы получаем логический сигнал высокого уровня. Если задать все значения переменных и выписать их вместе со значением функции, то получится таблица истинности для данной операции (рисунок 2.1).

A	B	X=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Рисунок 2.1- Таблица истинности операции «ИЛИ»

Из таблицы видно, что функция «ИЛИ» принимает значение 0, только если все переменные равны 0, иными словами, на нулевом наборе переменных. Контактный (переключательный) аналог, изображение логического элемента «ИЛИ» на функциональных и принципиальных схемах приведено на рисунке 2.2. На рисунке 2.2,в показано, как обозначается этот элемент в РК и Европе (IEC), на рисунке 2.2,с - условное обозначение, принятое в Америке (ANSI).

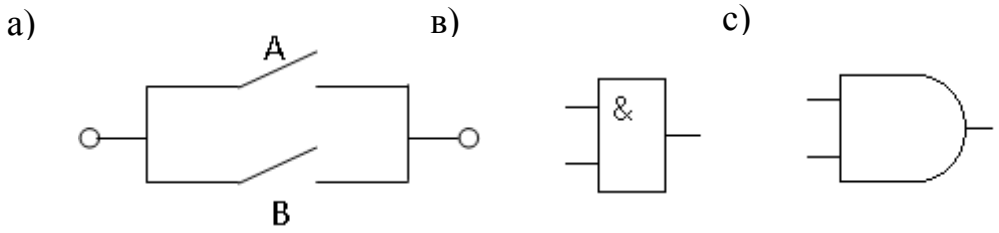


Рисунок 2.2 - Условное обозначение логического элемента «ИЛИ»

2. Операция «И».

Результат операции «И» равен 1 тогда и только тогда, когда все аргументы операции равны 1, в остальных случаях результат равен 0. Это операция логического умножения, поэтому часто она обозначается так же, как произведение в элементарной алгебре, но ее называют также «конъюнкцией (лат. conjunctio — соединение) и обозначают знаком ^. Операция конъюнкции на структурных схемах обозначается знаком «&», являющимся сокращенной записью английского слова «and».

Таким образом, выражения:

$$X = A \wedge B;\ X = A\&B;\quad X = A \cdot B;\quad X = A \times B;\ X = AB$$

эквивалентны и читаются одинаково: «икс равен А и В».

На рисунке 2.3 приведена таблица истинности этой функции. Из таблицы истинности следует, что результат конъюнкции принимает значение 1 только при единичном наборе переменных.

A	B	X=A·B
0	0	0
0	1	0
1	0	0
1	1	1

Рисунок 2.3 – Таблица истинности логического элемента «И»

Контактный аналог и изображение логических элементов «И» на функциональных и принципиальных схемах по различным стандартам показано на рисунке 2.4.

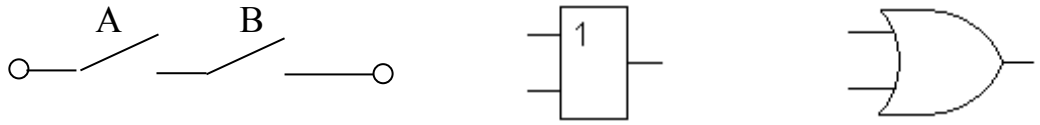


Рисунок 2.4 – Условное обозначение логического элемента «И»

3. Операция «НЕ».

Результат операции «НЕ» имеет значение, противоположное значению аргумента. *Логическое отрицание*, или операция «НЕ» называется также *инверсией*. Эта операция не имеет аналога в элементарной алгебре и является специфической операцией булевой алгебры. Она обозначается чертой над обозначением переменной $\bar{A}$ или как $\neg$ A и читается «не А» или «инверсия А», так как возможны только два значения переменной, то $\overline{0} = 1$, а $\overline{1} = 0$. Таблица истинности проста и приведена на рисунке 2.5.

A	$\bar{A}$
0	1
1	0

Рисунок 2.5 - Таблица истинности логического элемента «НЕ»

Три рассмотренные логические операции являются базовыми, на их основании могут быть реализованы и другие, производные от них операции.

4. Операция *«ЕСЛИ-ТО».*

Операция, выражаемая связками *«если ..., то», «из ... следует»,* называется *импликацией* (лат. *implico* — тесно связаны) и обозначается знаком $\rightarrow$. Высказывание A→B ложно тогда и только тогда, когда *A* истинно, а *B* ложно.

Каким же образом импликация связывает два элементарных высказывания? Покажем это на примере высказываний: *«данный четырёхугольник — квадрат»* (A) и *«около данного четырёхугольника можно описать окружность»* (B). Рассмотрим составное высказывание, понимаемое как *«если данный четырёхугольник квадрат, то около него можно описать окружность».* Есть три варианта, когда высказывание A→B истинно:

а) A истинно и B истинно, то есть данный четырёхугольник квадрат, и около него можно описать окружность;

в) А ложно и В истинно, то есть данный четырёхугольник не является квадратом, но около него можно описать окружность (разумеется, это справедливо не для всякого четырёхугольника);

с) А ложно и В ложно, то есть данный четырёхугольник не является квадратом, и около него нельзя описать окружность.

Ложен только один вариант, когда А истинно, а В ложно, то есть данный четырёхугольник является квадратом, но около него нельзя описать окружность.

В обычной речи связка *«если ..., то»* описывает причинно-следственную связь между высказываниями. Но в логических операциях смысл высказываний не учитывается. Рассматривается только их истинность или ложность.

5.Операция «РАВНОСИЛЬНО»

Операция, выражаемая связками «*тогда и только тогда*», «*необходимо и достаточно*», «*... равносильно ...*», называется *эквиваленцией* или двойной импликацией и обозначается знаком ↔ или **~.** Высказывание А↔В истинно тогда и только тогда, когда значения А и В совпадают. Например, высказывания *«24 делится на 6 тогда и только тогда, когда 24 делится на 3», «23 делится на 6 тогда и только тогда, когда 23 делится на 3»* истинны, а высказывания *«24 делится на 6 тогда и только тогда, когда 24 делится на 5», «21 делится на 6 тогда и только тогда, когда 21 делится на 3»* ложны. Высказывания А и В, образующие составное высказывание А↔В, могут быть совершенно не связаны по содержанию, например: *«три больше двух»* (А), *«пингвины живут в Антарктиде»* (В). Отрицаниями этих высказываний являются высказывания *«три не больше двух»* ($\bar{A}$), *«пингвины не живут в Антарктиде»* ($\bar{B}$). Образованные из высказываний А и В составные высказывания А↔В и $\bar{A}$ ↔$\bar{B}$истинны, а высказывания А↔$\bar{B}$ и $\bar{A}$ ↔В - ложны.

Итак, нами рассмотрены пять логических операций: отрицание, конъюнкция, дизъюнкция, импликация и эквиваленция.

Импликацию можно выразить через дизъюнкцию и отрицание:

$$A \leftrightarrow B = \bar{A} \vee B.$$

Эквиваленцию можно выразить через *отрицание*, *дизъюнкцию* и *конъюнкцию*:

$$A \leftrightarrow B = (\bar{A} \vee B) \cdot (\bar{B} \vee A).$$

Таким образом, операций отрицания, дизъюнкции и конъюнкции достаточно, чтобы описывать и обрабатывать логические высказывания.

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания «НЕ», затем конъюнкция «И»),

после конъюнкции — дизъюнкция «ИЛИ» и в последнюю очередь — импликация.

Лекция №3. Преобразование логических функций. Законы булевой алгебры

Цели лекции:

- освоить основные законы и правила логических операций;
- понять представление логических операций равносильными схемами.

Содержание лекции:

- законы логических операций: переместительный, сочетательный, распределительный, инверсии;
- правила логических операций: идемпотенции, поглощения, склеивания;
- правила работы с константами, работы с переменной и ее инверсией, проведения двойной инверсии.

Законы алгебры логики записываются в виде формул, которые позволяют проводить равносильные преобразования логических выражений.

В булевой алгебре существуют четыре пары основных законов: два переместительных, два сочетательных, два распределительных и два закона инверсии. Эти законы устанавливают равносильность различных выражений, т. е. рассматривают такие выражения, которые можно взаимно заменить подобно замене для тождеств в обычной алгебре. В качестве символа равносильности примем символ, одинаковый с символом равенства в обычной алгебре (=). В дальнейшем будем использовать привычные знаки алгебраических преобразований, что позволит упростить запись и понимание логических преобразований.

Ниже приводятся основные законы для логических операций. Используя законы алгебры логики, можно осуществлять тождественные преобразования формул, упрощать такие формулы. Это необходимо при создании логических схем.

Переместительный закон. Для логической суммы и произведения порядок расположения переменных безразличен:

$$A + B = B + A;$$

$$A \cdot B = B \cdot A.$$

Соответствующие этим выражениям схемы представлены на рисунке 3.1. Рисунок 3.1,а иллюстрирует переместительный закон для логического сложения. Левая и правая схемы представляют нормально разомкнутые цепи, каждая из которых при срабатывании любого из элементов (А или В) замыкается, т. е. эти схемы равносильны.

Соответствующие логическому умножению схемы представлены на рисунке 3.1, б их равносильность также очевидна.

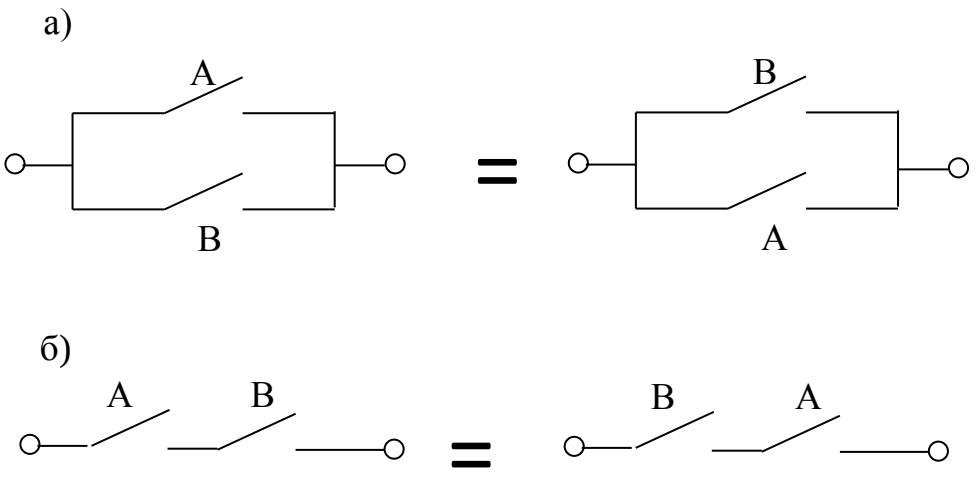


Рисунок 3.1 - Схема логического умножения

Сочетательный закон. Результат последовательного сложения переменных или умножения их не зависит от порядка этих действий:

$$A + (B + C) = (A + B) + C;$$

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C.$$

Рисунок 3.2,а иллюстрирует сочетательный закон для логического сложения. Левая и правая схемы представляют нормально разомкнутые цепи, каждая из которых при срабатывании любого из элементов (A или B) замыкается, т. е. эти схемы равносильны.

Соответствующие логическому сложению схемы представлены на рисунке 3.2,б, их равносильность также очевидна.

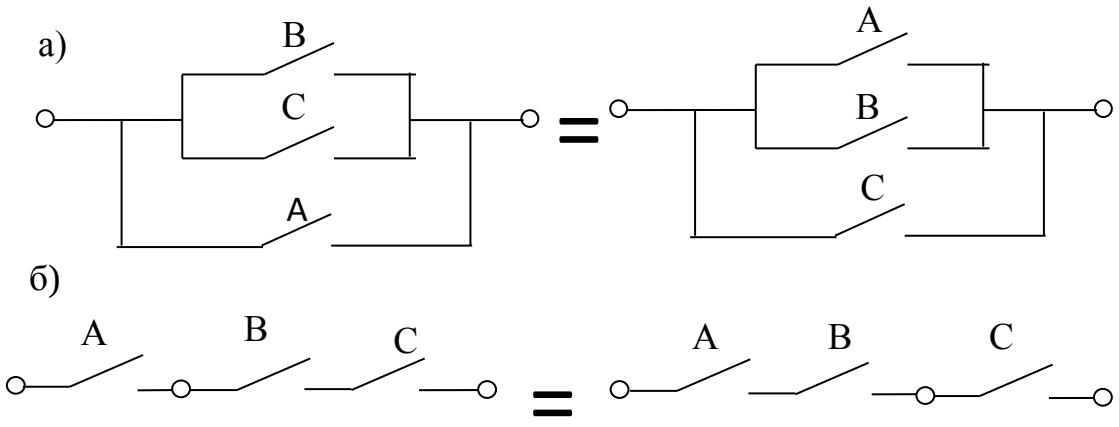


Рисунок 3.2 - Схема логического сложения

Распределительный закон. Общий множитель можно выносить за скобки, как в обычной алгебре:

$$A \cdot (B + C) = (A \cdot B) + A \cdot C;$$

$$A \cdot (B + C) = (A + B) \cdot (A + C).$$

Вторая запись не имеет аналога в обычной алгебре и доказывается преобразованиями «справа налево»:

$$(A + B) \cdot (A + C) = A \cdot A + A \cdot C + A \cdot B + B \cdot C =$$

$$A + A \cdot C + A \cdot B + B \cdot C \quad = A \cdot (1 + C + B) + B \cdot C.$$

Контакторная иллюстрация распределительного закона приведена на рисунке 3.3.

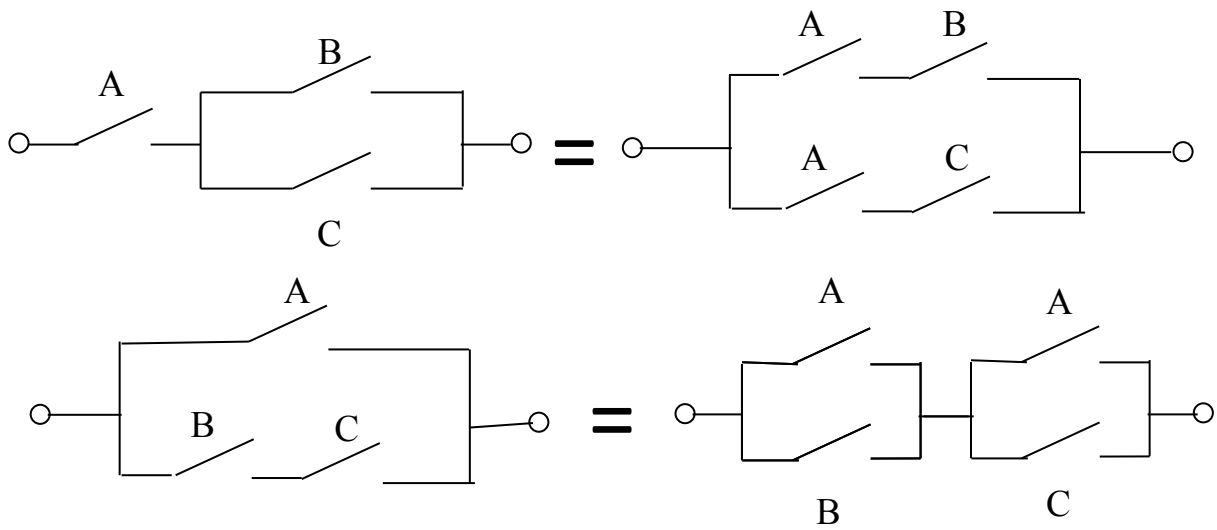


Рисунок 3.3 - Контакторная иллюстрация распределительного закона

Преобразования логических функций, выполняемые с применением распределительного закона, являются основным методом упрощений, так как вынесение общего множителя за скобки сокращает общее число переменных выражения, следовательно, позволяет сократить число элементов в схемах логических устройств.

Закон инверсии (закон Моргана — Шеннона). Отрицание логического сложения равносильно произведению отрицаний слагаемых, и, наоборот, отрицание логического умножения равносильно сумме отрицаний сомножителей:

$$\overline{A + B} = \overline{A} \cdot \overline{B};$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}.$$

Релейная интерпретация закона инверсии представлена на рисунке 3.4.

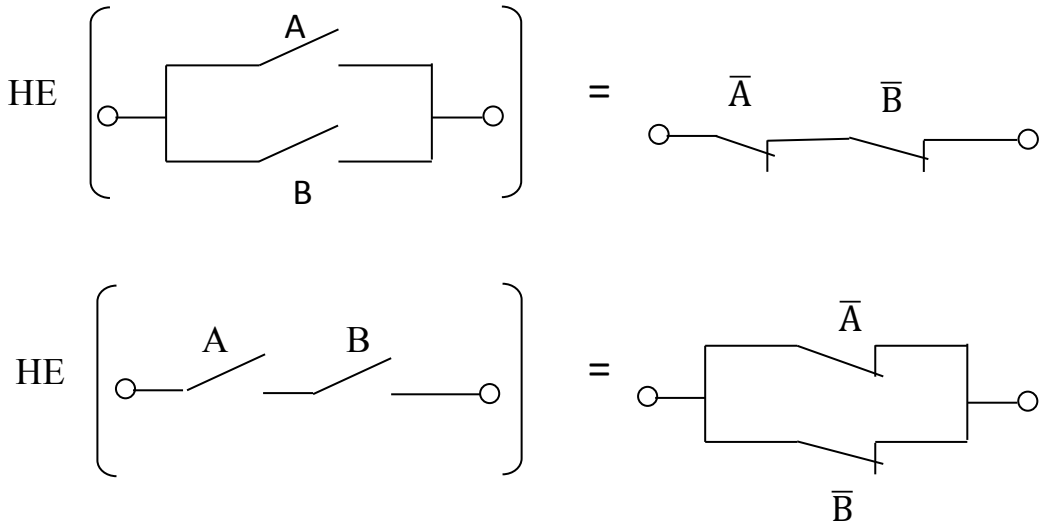


Рисунок 3.4 - Релейная интерпретация закона инверсии

Инверсия произвольной комбинации двоичных переменных, соединенных знаком «плюс» или «умножение», эквивалентна замене в ней значений переменных их инверсиями при одновременном изменении знака «плюс» на знак «умножение» и наоборот. Закон инверсии встречается только в алгебре логики.

Таким образом, закон инверсии позволяет заменить операцию «ИЛИ» операцией «И», а при необходимости — наоборот. Это особенно важно, поскольку при широком использовании интегральных логических элементов в построении логических устройств наиболее часто используют элементы базисов «И-НЕ», «ИЛИ-НЕ».

Закон поглощения. Сложение переменной с этой же переменной, умноженной на другую переменную, или умножение переменной на сумму этой же переменной и другой переменной равно первой переменной:

$$A + A \cdot B = A;$$

$$A \cdot (A + B) = A.$$

Доказательство получим, используя распределительный закон:

$$A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A;$$

$$A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A;$$

$$A \cdot (\overline{A} + B) = A \cdot B;$$

$$A + (\overline{A} \cdot B) = A + B.$$

Релейная интерпретация закона поглощения представлена на рисунке 3.5.

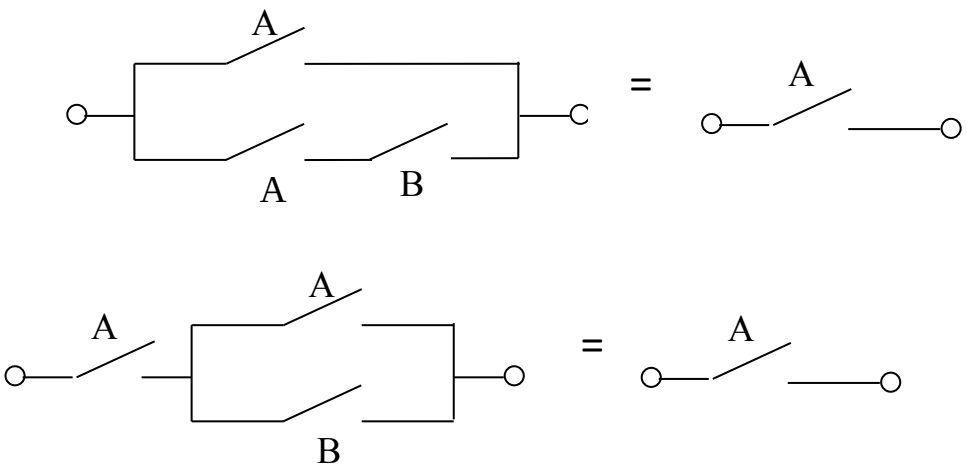


Рисунок 3.5 - Релейная интерпретация закона поглощения

Закон склеивания. Сумма произведений первой и второй переменных и второй переменной и инверсии первой переменной равна второй переменной. Произведение суммы двух переменных и суммы инверсии первой переменной со второй переменной равно второй переменной:

$$A \cdot B + A \cdot \overline{B} = A;$$

$$(A + B) \cdot (A + \overline{B}) = A;$$

$$A \cdot B + A \cdot \overline{B} = A(B + \overline{B}) = A \cdot 1 = A;$$

$$(A + B) \cdot (A + \overline{B}) = A \cdot A + B \cdot A + A \cdot \overline{B} = A + A(B + \overline{B}) + 0 = A + A \cdot 1 = A.$$

Закон склеивания иллюстрируется рисунком 3.6.

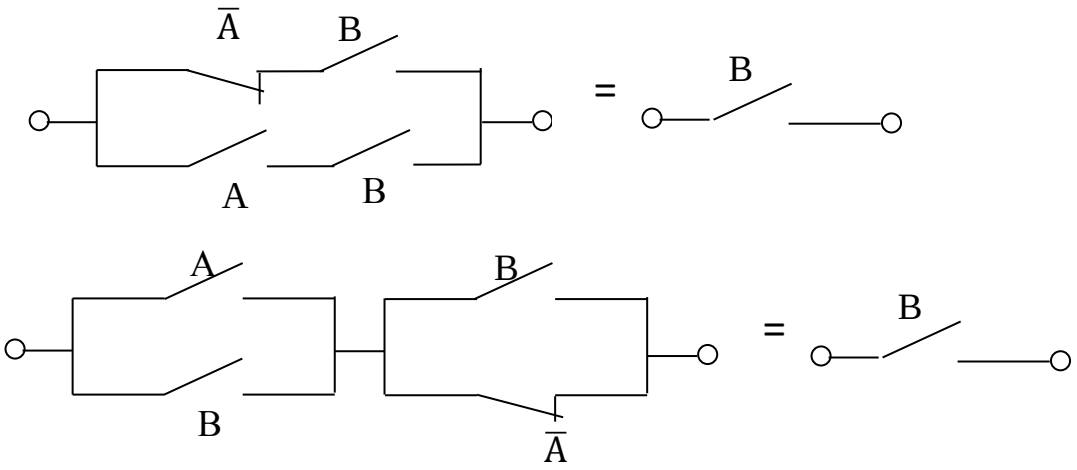


Рисунок 3.6 - Иллюстрация закона склеивания

Выполняя минимизацию, пользуются также следствиями законов алгебры логики, основные из которых следующие:

Правило идемпотентности:

$$A + A = A;$$
$$A \cdot A = A.$$

Правила операций с константами:

$$A + 0 = A;$$
$$A + 1 = 1;$$
$$A \cdot 0 = 0;$$
$$A \cdot 1 = A.$$

Законы инверсии (отрицания):

$$A + \overline{A} = 1;$$
$$A \cdot \overline{A} = 0.$$

Снятие двойного отрицания:

$$\overline{\overline{A}} = A.$$

Последнее следствие может быть представлено и следующим образом:

$$A + \overline{A} \cdot \overline{B} = A + \overline{B};$$
$$\overline{A} + A \cdot B = \overline{A} + B;$$
$$\overline{A} + A \cdot \overline{B} = \overline{A} + \overline{B}.$$

Используя эти простые правила следствия, можно проводить операции, аналогичные школьной алгебре для упрощения логических схем.

Знак отрицания над выражением дает возможность опустить скобки, в которые это выражение заключено (отрицание является самой старшей логической операцией).

При упрощении выражений следует помнить старшинство операций: инверсия, конъюнкция, дизъюнкция.

Для удобства применения приведенного выше материала, сведем его в таблицу 3.1.

Таблица 3.1 – Законы и правила алгебры логики

Закон	Для «ИЛИ»	Для «И»
Переместительный	$A + B = B + A$	$A \cdot B = B \cdot A$
Сочетательный	$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Распределительный	$A \cdot (B + C) = (A \cdot B) + A \cdot C$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
Де Моргана	$\overline{A + B} = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$
Правило		
Идемпотенции	$A + A = A$	$A \cdot A = A$
Поглощения	$A + A \cdot B = A$	$A \cdot (A + B) = A$
Склеивания	$A \cdot B + A \cdot \overline{B} = A$	$(A + B) \cdot (A + \overline{B}) = A$
Операция переменной с ее инверсией	$A + \overline{A} = 1,$	$A \cdot \overline{A} = 0.$
Операция с константами	$A + 0 = A,$ $A + 1 = 1,$	$A \cdot 0 = 0,$ $A \cdot 1 = A.$

Лекция №4. Релейные и бесконтактные логические элементы автоматики

Цели лекции:

- получить представление о видах и характеристиках релейных элементов;
- ознакомиться с видами и областью применения релейных элементов в электротехнике.

Содержание лекции:

- релейные элементы, предназначенные для логических преобразований;
- релейные характеристики идеальных релейных элементов;
- электромагнитные реле постоянного и переменного тока;
- другие виды реле, используемых в электроэнергетике.

Исторически первыми устройствами, для описания действий которых использовали логические функции, были устройства, выполненные на релейно-контактных элементах. Для проектирования таких устройств была разработана теория релейно-контактных схем. Затем появились бесконтактные устройства, предназначенные только для логических преобразований сигналов и представляющие собой конструктивно оформленные изделия.

Релейный элемент (реле) — это элемент автоматики, в котором при плавном изменении входного сигнала выходной изменяется скачкообразно, принимая одно из двух или трех стабильных значений.

Релейные характеристики — характеристики кусочно-линейного вида, соответствующие преобразованию в техническом устройстве (системе) непрерывной входной величины x в дискретные значения выходной величины y_n, где n — число возможных ее значений (уровней 2 или 3).

На рисунке 4.1 приведены релейные характеристики идеальных двухпозиционных релейных элементов, которые рассматриваются в данном курсе.

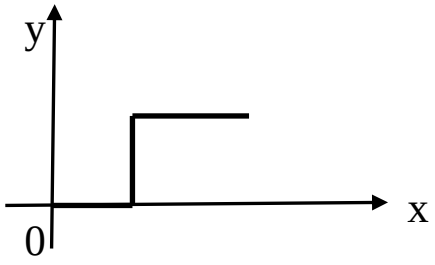


Рисунок 4.1 - Релейные характеристики двухпозиционных релейных элементов

Некоторые релейные элементы могут обладать свойством фиксации, т.е. оставаться в занятом ими состоянии и после снятия воздействия на входе. В этом случае релейный элемент возвращается в первоначальное состояние обычно после подачи воздействия на другой его вход (или воздействия другого знака на тот же вход). Релейный элемент с фиксацией применяют, например, для реализации памяти вычислительных и управляющих машин. Рассмотрим наиболее распространенные контактные электромеханические реле, в которых электрическая величина преобразуется в механическое перемещение, вызывающее замыкание или размыкание контактов.

Рисунок 4.2 - Внешний вид электромеханических реле

Существуют различные виды электромеханических реле:
- электромагнитные реле постоянного и переменного тока;
- магнитоэлектрические реле;
- электродинамические реле;
- индукционные реле;
- электротермические реле.

Электромагнитное нейтральное реле постоянного тока (рисунок 4.3, а) состоит из неподвижного магнитопровода 1, являющегося сердечником обмотки 4, и подвижного якоря 2. Якорь и сердечник реле изготовляют из магнит омягкого материала. При протекании через обмотку 4 тока *I* определенной величины якорь 2 притягивается к сердечнику, что приводит к замыканию контактов 6. При снятии напряжения якорь 2 и контакты 6 возвращаются в исходное положение под действием пружины 5. Для устранения влияния остаточного магнитного потока (отлипания якоря от сердечника) на якоре предусмотрен штифт 3 из немагнитного материала (латунь, медь). Контакты реле, показанные на рисунке 4.3, а, замыкаются при срабатывании реле и размыкаются при отпускании. Такие контакты называют замыкающими. Если контакты при срабатывании реле размыкаются, а при отпускании замыкаются, их называют размыкающими. Реле может иметь не один контакт, а контактную группу, в которой есть замыкающиеся и размыкающиеся контакты.

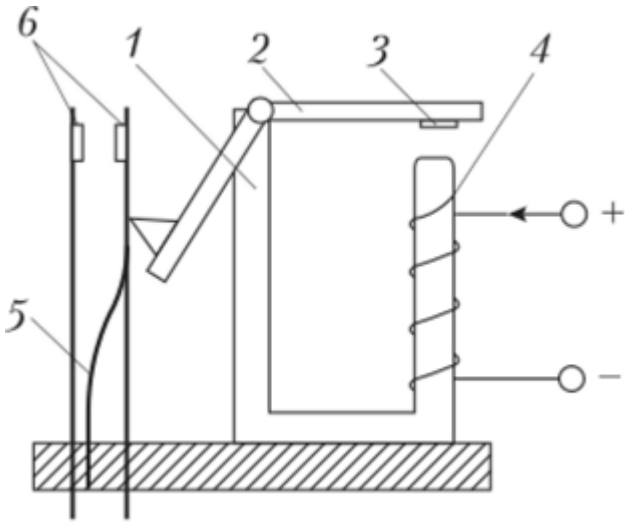


Рисунок 4.3 - Электромагнитные реле постоянного тока

Электромагнитное реле переменного тока (рисунок 4.4) по устройству аналогично электромагнитному реле постоянного тока, но для уменьшения вихревых токов и потерь на перемагничивание стали его сердечник и якорь изготовляют из листовой электротехнической стали.

Если обычное реле постоянного тока включить в цепь переменного тока, то якорь реле будет вибрировать, поскольку два раза за период ток *I*, магнитный поток *Ф*, а следовательно, и сила притяжения F будут проходить через нуль. Вибрация вызовет шум, утяжелит работу контактов и ускорит их износ.

Чтобы устранить вибрации якоря реле и уменьшить искрение контактов, предусматривают такую конструкцию реле, чтобы на якорь действовали два переменных магнитных потока, сдвинутых друг относительно друга по фазе, с тем чтобы тяговое усилие никогда не падало до нуля. Конструкция такого реле приведена на рисунке 4.4.

Полюс сердечника реле раздвоен. Поэтому магнитный поток Ф сердечника раздваивается на два потока: и $Ф_2$. На одну половину раздвоенного сердечника с потоком $Ф_t$ надет медный короткозамкнутый виток. В витке наводится *ЭДС* и возникает ток, который, в свою очередь, создает магнитный поток $Ф_к$. В результате магнитный поток $Ф_{1?}$ проходящий через короткозамкнутый виток, отстает по фазе на угол ф от потока $Ф_2$, проходящего через свободную половину полюса.

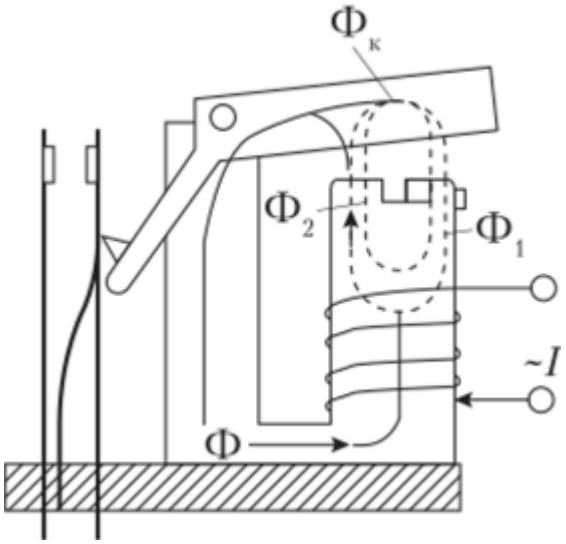


Рисунок 4.4 - Электромагнитное реле переменного тока

Магнитоэлектрические, электродинамические и индукционные реле по своей конструкции обычно напоминают магнитоэлектрические, электродинамические или индукционные измерительные приборы, но только вместо стрелки установлен подвижный контакт.

Электротермическое реле основано на расширении твердого тела, жидкости или газа при нагревании их электрическим током. Механическое перемещение, вызванное этим нагреванием, приводит к замыканию или размыканию электрического контакта.

Герконовые реле выполняются на основе герконов, герметизированных магнитоуправляемых ферромагнитных контактов, замыкающихся под действием магнитного поля.

Реле времени — это устройства автоматики, предназначенные для задержки во времени сигнала включения (отключения) цепей управления и исполнительных механизмов. Чтобы задержать сигнал на время от долей секунды до нескольких десятков секунд, используют конденсаторное реле времени (ВЛ-43, ВЛ-44).

Конденсаторные реле основаны на явлении заряда конденсатора, а также на принципе пересчета импульсов при заряде конденсатора. В реле с механическим устройством замедления выдержка времени создается с помощью часового механизма или синхронных двигателей. Аналогичное устройство имеют циклические программные реле времени, обеспечивающие включение и отключение ИМ с заданной периодичностью в течение конкретного цикла времени.

Также имеются и другие релейные элементы: контакторы и магнитные пускатели.

Контакторы — это специальные электромагнитные реле постоянного или переменного тока, используемые для подключения силовых токоприемников к источнику питания. Их изготовляют с числом полюсов от 1

до 5 и оснащают блок-контактами. Наиболее распространены контакторы постоянного тока типа КП и переменного тока типа КТ.

Магнитные пускатели — это контакторы, оборудованные защитными устройствами. Широко применяют магнитные пускатели типа ПМЛ, ПМА, ПА, предназначенные для управления трехфазными установками переменного тока и защиты от перегрузок с помощью тепловых реле.

Износ контактов силовой аппаратуры и необходимость их постоянного обслуживания являются недостатками электромеханической аппаратуры. Поэтому она все чаще заменяется бесконтактной аппаратурой – бесконтактными реле, компараторами, пускателями и т.д.

Процессы при включении и отключении реле очень сложны. Они описываются нелинейными дифференциальными уравнениями. Подробное изучение этих процессов проводится в специальных дисциплинах специальности («Релейная защита» и другие). В нашей дисциплине рассматриваются идеализированные релейные элементы с двумя состояниями: «контакт замкнут» (1) и «контакт разомкнут» (0).

Лекция №5. Логические элементы

Цели лекции:

- ознакомиться с базовыми и комбинированными логическими элементами;
- изучить функции, условные обозначения и компоновку логических элементов.

Содержание лекции:

- базовые логические элементы «И», «ИЛИ», «НЕ»;
- комбинированные логические элементы «И-НЕ», «ИЛИ-НЕ», «Исключающее ИЛИ».

Логические элементы выполняют логические операции. С помощью логических элементов можно составлять логические схемы, реализующие любые логические функции [1,4,6].

Базовых логических элементов (логических вентилей) всего три: элемент «И», элемент «ИЛИ», элемент «НЕ». Каждый из логических элементов реализует соответствующую логическую операцию и имеет условное графическое обозначение. Логические элементы, работают как самостоятельные элементы в виде микросхем малой степени интеграции, так и входят в виде компонентов в микросхемы более высокой степени интеграции. Таких элементов можно насчитать не один десяток.

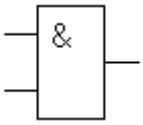
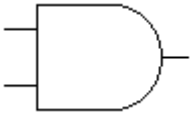
Условное обозначение, принятое в РК (ГОСТ) и Европе (IEC)	Условное обозначение, принятое в Америке (ANSI)
	

Рисунок 5.1 - Условное обозначение логического элемента «И»

Элемент «И» реализует операцию логического умножения.

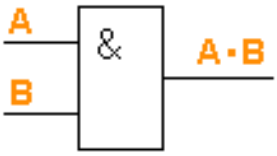


Рисунок 5.2 – Операция логического умножения

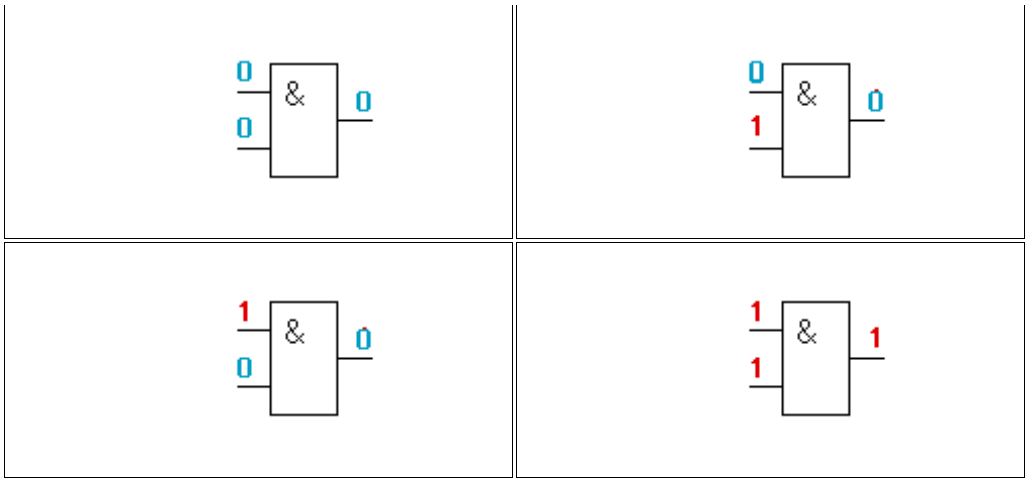


Рисунок 5.3 - Логический элемент «И»

Если входов у элемента будет 4 или 8, то указанное условие (наличие высокого уровня), должно выполняться на всех входах: «И»-на входе 1, «И»-на входе 2, «И»-на входе3 …..«И»-на входе N. Лишь в этом случае на выходе будет также высокий уровень.

Для того чтобы было проще разобраться в логике работы элемента «И», на рисунке 5.1 представлен его аналог в виде контактной схемы. Здесь выход элемента «И» представлен лампой «С». Если лампа светится, то это соответствует высокому уровню на выходе элемента «И». Часто такие

элементы называют 2-И, 3-И, 4-И, 8-И. Первая цифра указывает на количество входов.

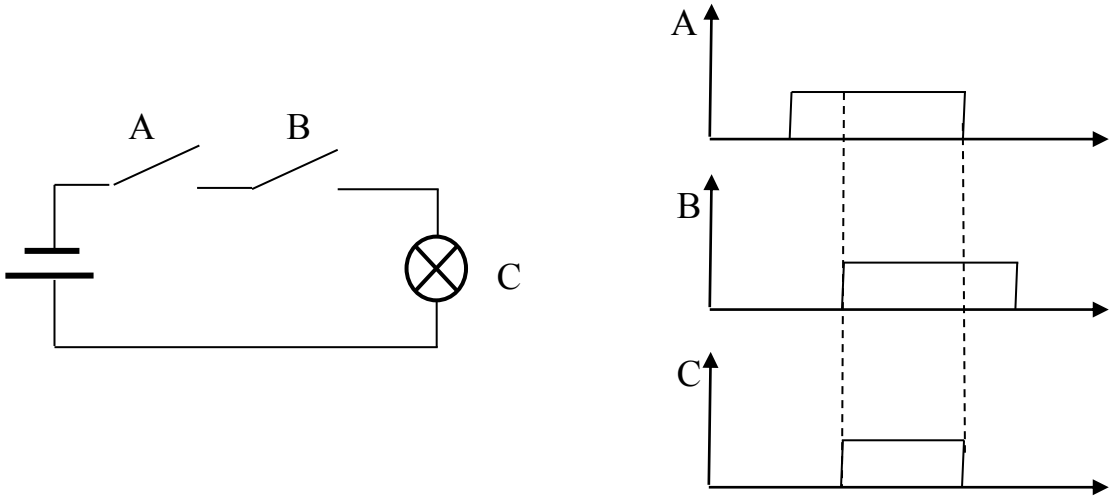


Рисунок 5.4 - Контактная схема элемента «И»

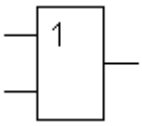
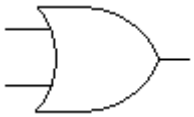
Условное обозначение, принятое в РК (ГОСТ) и Европе (IEC)	Условное обозначение, принятое в Америке (ANSI)
	

Рисунок 5.5 - Условное обозначение логического элемента «ИЛИ»

Элемент «ИЛИ» реализует операцию логического сложения.

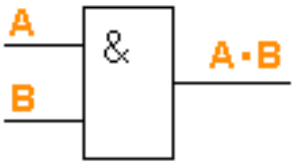


Рисунок 5.6 – Операция логического сложения

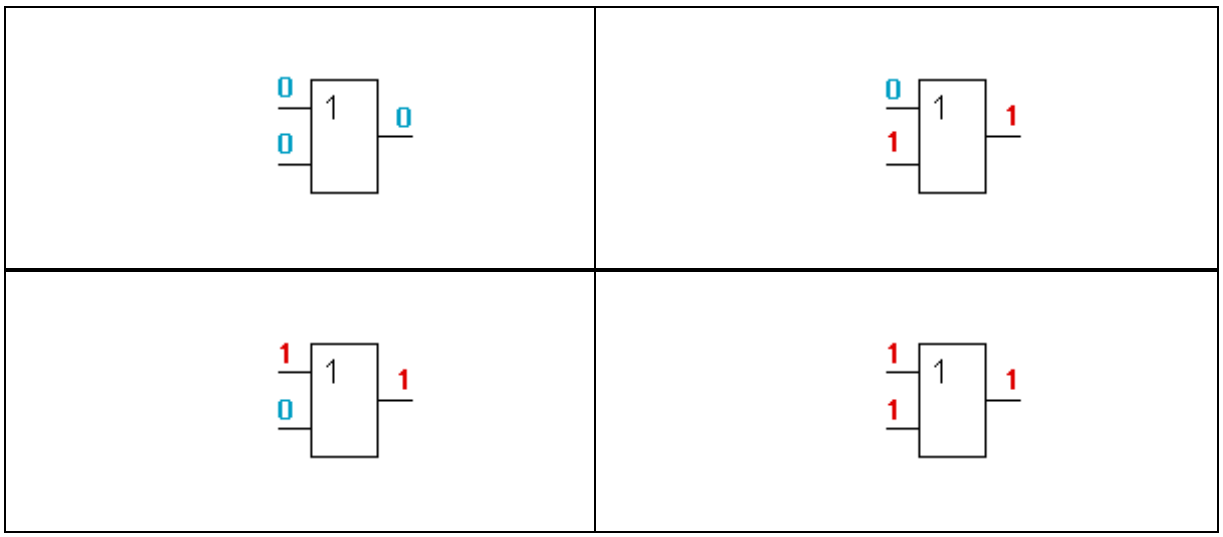


Рисунок 5.7 – Принцип работы логического элемента «ИЛИ»

Контактная схема, представленная на рисунке 5.2, - это аналог функции: нажатие на любую из кнопок (высокий уровень) или на обе кнопки сразу приведет к свечению лампочки (высокий уровень). В данном случае кнопки - это входные сигналы A и B, а лампочка - выходной сигнал C. Здесь же приведен временная диаграмма и таблица истинности соответственно.

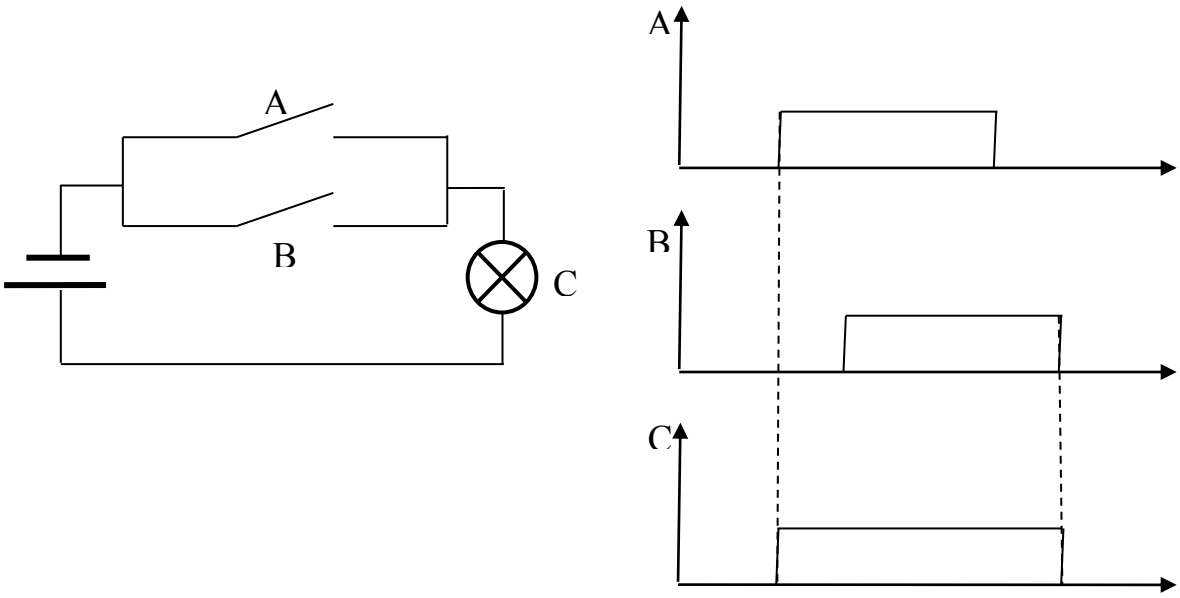


Рисунок 5.8 - Контактная схема логического элемента «ИЛИ»

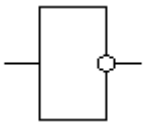
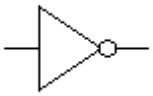
Условное обозначение, принятое в РК (ГОСТ) и Европе (IEC)	Условное обозначение, принятое в Америке (ANSI)
	

Рисунок 5.9 – Условное обозначение элемента «НЕ» (инвертор)

Элемент НЕ реализует операцию логического отрицания (инверсии).

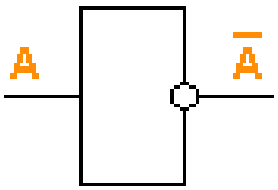


Рисунок 5.10 – Элемент «НЕ»

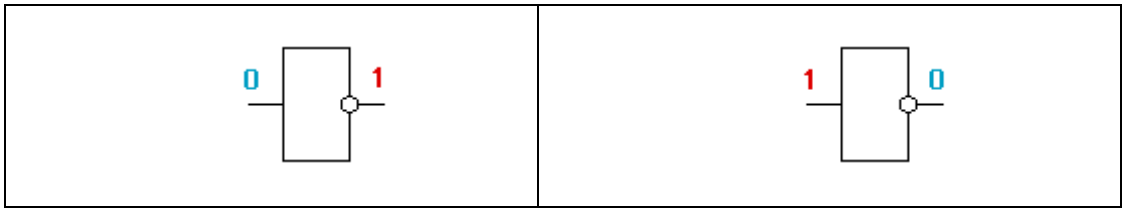


Рисунок 5.11- Принцип работы инвертора

Как только что было сказано – инвертор самая сложная схема цифровой техники. И это подтверждает его контактная схема: если до этого было достаточно лишь только кнопок, то теперь к ним добавилось реле. Пока кнопка А не нажата (логический ноль на входе), реле К1 обесточено, и его нормально-замкнутые контакты включают лампочку В, что соответствует логической единице на выходе.

Если же нажать кнопку (подать на вход логическую единицу), то реле включится, контакты К1 разомкнутся, лампочка погаснет, что соответствует логическому нулю на выходе. Сказанное подтверждают временная диаграмма на рисунке 3в и таблица истинности на рисунке 3 г.

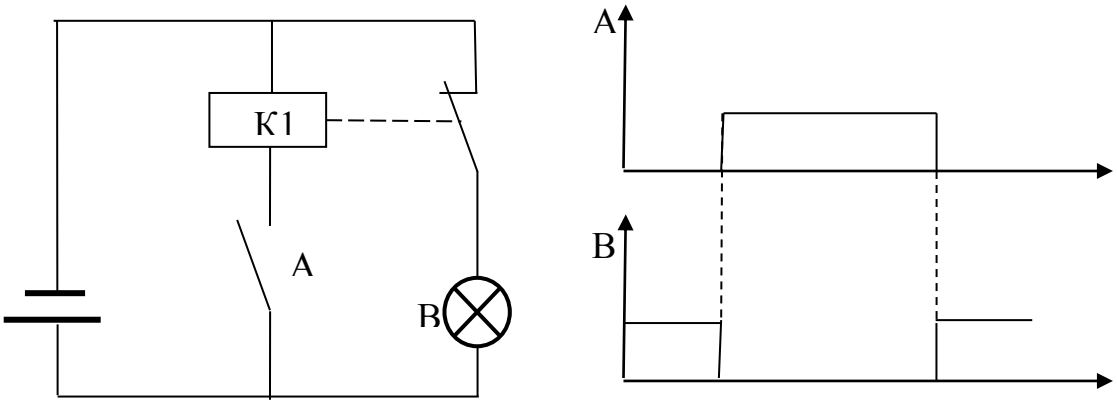


Рисунок 5.12 - Контактная схема инвертора

На практике часто используются комбинированные элементы «И-НЕ» и «ИЛИ-НЕ». С помощью логических элементов «И-НЕ» можно реализовать любую из базовых логических операций, а значит, и построить любую логическую схему. То же самое можно сделать и с применением элемента «ИЛИ-НЕ».

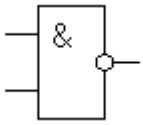
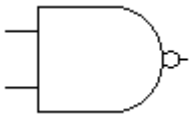
Условное обозначение, принятое в РК (ГОСТ) и Европе (IEC)	Условное обозначение, принятое в Америке (ANSI)
	

Рисунок 5.13 - Условное обозначение логического элемента «И-НЕ»

Элемент «И-НЕ» последовательно реализует операцию логического умножения, а затем инверсию полученного результата. С помощью базовых элементов «И-НЕ» можно представить следующим образом:

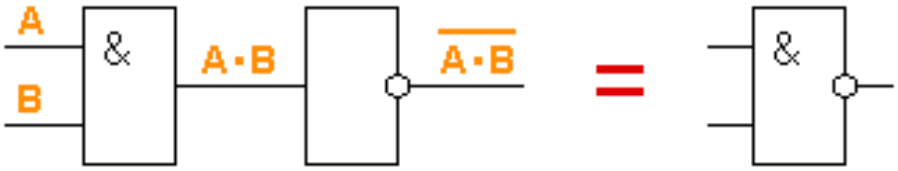


Рисунок 5.14 – Логический элемент «И-НЕ»

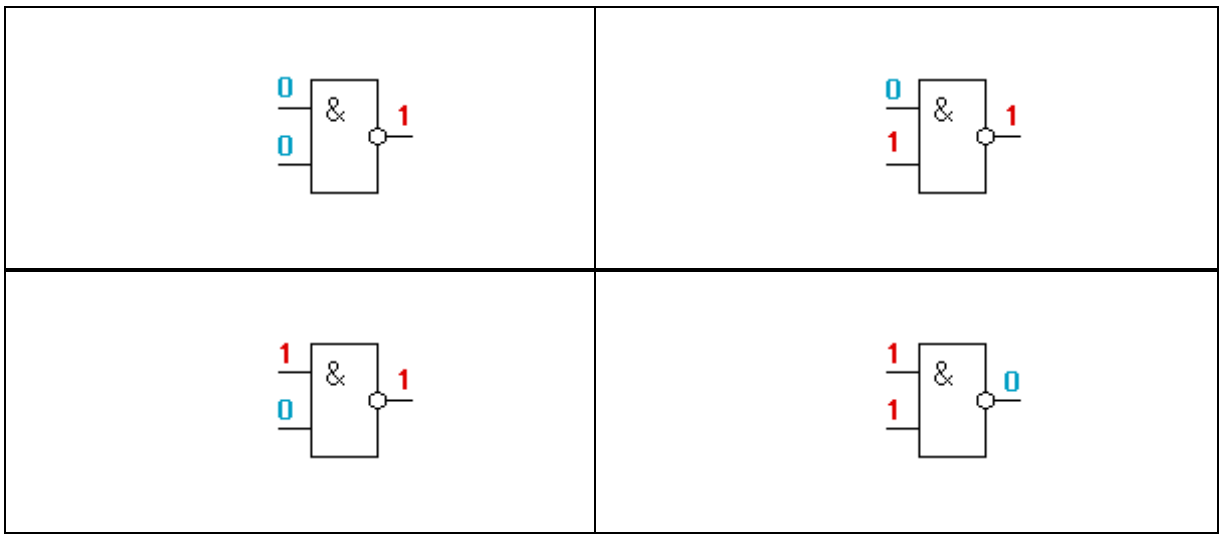


Рисунок 5.15 – Принцип работы логического элемента «И-НЕ»

Базовые логические элементы, построенные на основе элементов «И-НЕ».

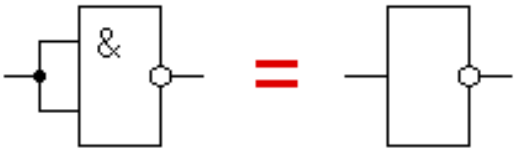


Рисунок 5.16 – Логический элемент «НЕ» из элемента «И-НЕ»

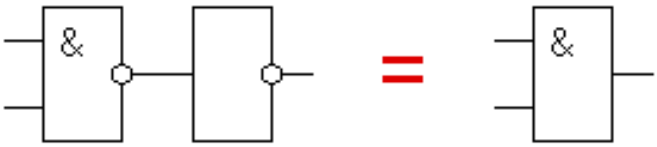


Рисунок 5.17 – Логический элемент «И» из элементов «И-НЕ»

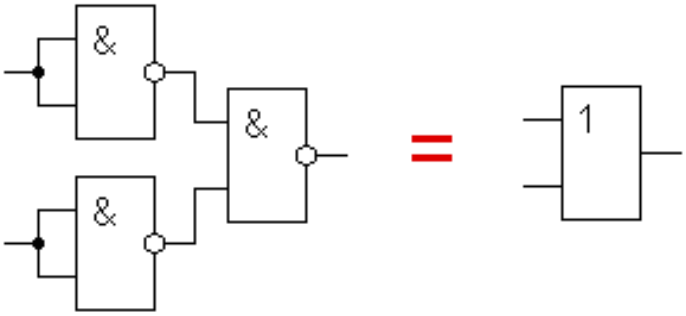


Рисунок 5.18 – Логический элемент «ИЛИ» из элементов «И-НЕ»

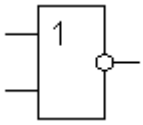
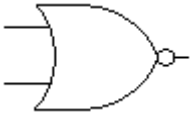
Условное обозначение, принятое в РК (ГОСТ) и Европе (IEC)	Условное обозначение, принятое в Америке (ANSI)
	

Рисунок 5.19 – Условное обозначение логического элемента «ИЛИ-НЕ»

Элемент «ИЛИ-НЕ» последовательно реализует операцию логического сложения, а затем инверсию полученного результата. С помощью базовых элементов «ИЛИ-НЕ» можно представить следующим образом:

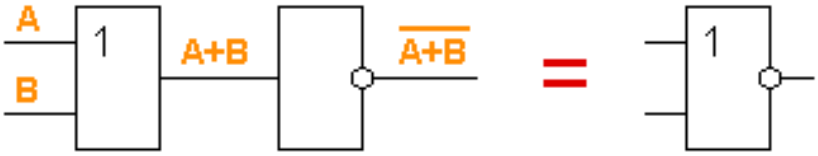


Рисунок 5.20 – Логический элемент «ИЛИ-НЕ»

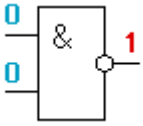
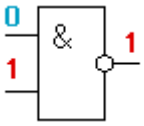
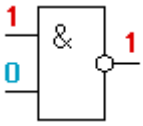
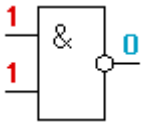
	
	

Рисунок 5.21 – Принцип работы логического элемента «ИЛИ-НЕ»

Базовые логические элементы, построенные на основе элементов «ИЛИ-НЕ»:

операция «НЕ» операция «ИЛИ» операция «И»

$$\bar{x} = \overline{x + x}$$

$$x + y = \overline{\overline{x + y}}$$

$$x \cdot y = \overline{\bar{x} + \bar{y}}$$

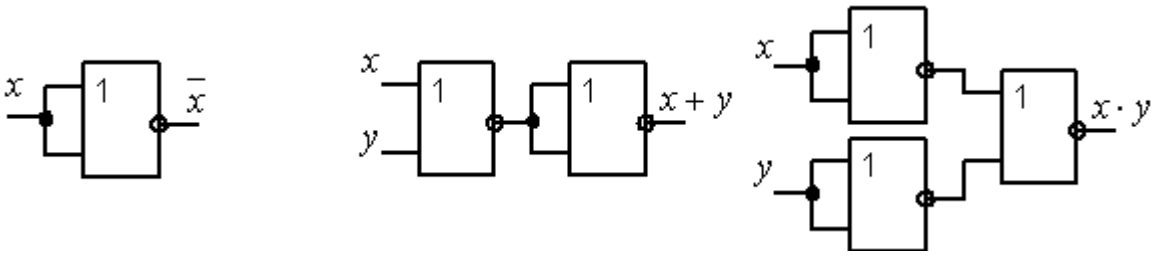


Рисунок 5.22 – Базовые логические элементы

Условное обозначение, принятое в РК (ГОСТ) и Европе (IEC)	Условное обозначение, принятое в Америке (ANSI)
=1	

Рисунок 5.23 — Условное обозначение исключающего «ИЛИ» (XOR)

Элемент «Исключающее ИЛИ» реализует операцию логического сложения по модулю 2.

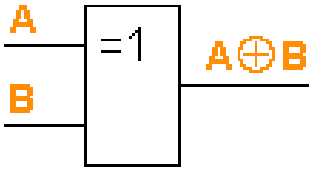


Рисунок 5.24 – Элемент «Исключающее ИЛИ»

На выходе элемента «Исключающее ИЛИ» будет логическая 1, если только один из входов равен 1, во всех остальных случаях на выходе будет 0.

В данном разделе рассмотрены двухвходовые логические элементы, но существуют также элементы с тремя и более входами.

Лекция №6. Анализ и синтез логических устройств. Методы минимизации логических функций и схем

Цели лекции:
- получить понятия о синтезе и анализе логических устройств;
- освоить приемы и способы упрощения логических формул.

Содержание лекции:
- виды логических устройств;
- процесс синтеза и анализа логических устройств;
- равносильные преобразования логических формул;
- приемы и способы, применяемые при упрощении логических формул.

Логические устройства — это устройства, у которых значения выходных сигналов зависят только от комбинации входных сигналов.

Комбинационные логические устройства — это устройства, у которых значения выходных сигналов зависят только от комбинации входных сигналов в данный момент времени.

Последовательностные логические устройства — это устройства, выходные сигналы которых зависят от значений входных сигналов не только в данный момент времени, но и в предыдущие моменты времени. В состав этих устройств обязательно входят элементы памяти — триггеры. Различают несколько видов триггеров в зависимости от того, какую элементарную функцию памяти они реализуют.

При разработке логического устройства сначала формулируют словесное описание его алгоритма действия. Затем составляют удовлетворяющую этому описанию логическую функцию *(абстрактный синтез)* и далее разрабатывают структурную логическую схему устройства *{структурный синтез)* [1,4,6].

В процессе абстрактного синтеза осуществляется переход от словесного описания технологического процесса (его нормальный ход и аварийные ситуации) к составлению алгоритма функционирования в виде таблицы, циклограммы, графика и т.п. *Циклограмма* представляет собой ряд горизонтальных строк, равных числу входов и выходов логического устройства. Для составления логического алгоритма управления технологическим оборудованием необходимо иметь полную информацию о ТП каждой технологической операции и применяемом оборудовании. На этой стадии уточняют последовательность операций и необходимые временные задержки для всех режимов работы объекта управления, определяют параметры, подлежащие контролю и учету в ходе процесса; формулируют требования управляемого объекта к логическому устройству. Эти требования представляют в виде значений двоичных сигналов, которые должны быть поданы на исполнительные устройства системы управления в зависимости от состояния управляемого объекта.

В процессе структурного синтеза происходит переход от логической функции, описывающей алгоритм функционирования, к структурной схеме логического устройства.

Однако прежде чем приступить к разработке схемы, необходимо попытаться преобразовать исходную логическую функцию к максимально простому виду. На основе структурной схемы логического устройства разрабатывают его принципиальную схему с использованием конкретной элементной базы, например в базисе «ИЛИ-НЕ» или «И-НЕ». Завершающий этап создания схемы логического устройства — разработка и согласование узлов связи устройства с оператором и управляемым объектом, защита от помех и т.п.

Исторически первыми устройствами, для описания действий которых использовали логические функции, были устройства, выполненные на релейно-контактных элементах. Для проектирования таких устройств была разработана теория релейно-контактных схем. Затем появились бесконтактные устройства, предназначенные только для логических преобразований сигналов и представляющие собой конструктивно оформленные изделия.

Устройства автоматики, действие которых описывается элементарными логическими функциями, обычно называют в соответствии с реализуемой ими логической операцией элементами «НЕ», «И», «ИЛИ», «И-НЕ», «ИЛИ-НЕ». Имея необходимые элементы, по логической функции можно синтезировать логическое устройство любой сложности. Однако построенная схема может оказаться неоправданно сложной, требующей использования большого числа логических элементов, что может повлиять на стоимость и надежность устройства. Во многих случаях удается так упростить логическую функцию, что соответствующая ей схема устройства оказывается существенно более простой и выполняющей поставленную задачу.

Методы упрощения комбинационных устройств называют методами минимизации логических функций. Метод минимизации основан на применении законов алгебры логики, или булевой алгебры, которые приведены ниже для минимального числа переменных. Эквивалентность левой и правой части уравнений обозначена знаком равенства. Одновременно изображены релейные эквиваленты рассматриваемых законов алгебры логики.

Равносильные преобразования логических формул имеют то же назначение, что и преобразования формул в обычной алгебре. Они служат для упрощения формул или приведения их к определённому виду путем использования основных законов алгебры логики.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т.п.), тогда как другие преобразования основаны на свойствах, которыми не обладают операции обычной алгебры (использование распределительного закона для конъюнкции, законов поглощения, склеивания, де Моргана и др.).

Покажем на примерах некоторые приемы и способы, применяемые при упрощении логических формул [1,4,6].

Для упрощения в (6.1) законы алгебры логики применяются в следующей последовательности: правило де Моргана, сочетательный закон, правило операций переменной с её инверсией и правило операций с константами:

$$\overline{X+Y}\cdot(X\cdot\bar{Y})=\bar{X}\cdot\bar{Y}\cdot(X\cdot\bar{Y})=\bar{X}\cdot X\cdot\bar{Y}\cdot\bar{Y}=0\cdot\bar{Y}\cdot\bar{Y}=0\cdot\bar{Y}=0. \quad (6.1)$$

Для упрощения в (2) применяется правило де Моргана, выносится за скобки общий множитель, используется правило операций переменной с её инверсией:

$$\bar{X}\cdot Y+\overline{X+Y}+X=\bar{X}\cdot Y+\bar{X}\cdot\bar{Y}+X=\bar{X}\cdot(\mathrm{Y}+\bar{\mathrm{Y}})+\mathrm{X}=\bar{X}+X=1. \quad (6.2)$$

Для упрощения в (3) повторяется второй сомножитель, что разрешено законом идемпотенции; затем комбинируются два первых и два последних сомножителя и используется закон склеивания:

$$(\mathrm{X}+\mathrm{Y})\cdot(\bar{\mathrm{X}}+\mathrm{Y})\cdot(\bar{\mathrm{X}}+\bar{\mathrm{Y}})=(\mathrm{X}+\mathrm{Y})\cdot(\bar{\mathrm{X}}+\mathrm{Y})\cdot(\bar{\mathrm{X}}+\mathrm{Y})\cdot(\bar{\mathrm{X}}+\bar{\mathrm{Y}})=Y\cdot\bar{X}. \quad (6.3)$$

Для упрощения в (6.4) вводится вспомогательный логический сомножитель $(Y+\bar{Y})$; затем комбинируются два крайних и два средних логических слагаемых и используется закон поглощения:

$$\begin{gathered}\mathrm{X}\cdot\bar{\mathrm{Y}}+\overline{\mathrm{X}}\cdot\mathrm{Y}\cdot\mathrm{Z}+\mathrm{X}\cdot\mathrm{Z}=\mathrm{X}\cdot\bar{\mathrm{Y}}+\overline{\mathrm{X}}\cdot\mathrm{Y}\cdot\mathrm{Z}+\mathrm{X}\cdot\mathrm{Z}\cdot(\mathrm{Y}+\bar{\mathrm{Y}})=\\ \mathrm{X}\cdot\bar{\mathrm{Y}}+\overline{\mathrm{X}}\cdot\mathrm{Y}\cdot\mathrm{Z}+\mathrm{X}\cdot\mathrm{Y}\cdot\mathrm{Z}+\mathrm{X}\cdot\bar{\mathrm{Y}}\cdot\mathrm{Z}=\\ (\mathrm{X}\cdot\bar{\mathrm{Y}}+\mathrm{X}\cdot\bar{\mathrm{Y}}\cdot\mathrm{Z})+(\overline{\mathrm{X}}\cdot\mathrm{Y}\cdot\mathrm{Z}+\mathrm{X}\cdot\mathrm{Y}\cdot\mathrm{Z})=\\ \mathrm{X}\cdot\bar{\mathrm{Y}}+\mathrm{Y}\cdot\mathrm{Z}.\end{gathered} \quad (6.4)$$

Для упрощения в (6.5) сначала добиваемся, чтобы знак отрицания стоял только перед отдельными переменными, а не перед их комбинациями, для этого дважды применяем правило де Моргана; затем используем закон двойного отрицания:

$$\overline{X\cdot Y+\bar{Z}}=\overline{X\cdot Y}\cdot\bar{\bar{Z}}=(\bar{X}+\bar{Y})\cdot Z. \quad (6.5)$$

Для упрощения в (6.6) выносятся за скобки общие множители; применяется правило операций с константами:

$$\mathrm{X}\cdot\mathrm{Y}+\mathrm{X}\cdot\mathrm{Y}\cdot\mathrm{Z}+\mathrm{X}\cdot\mathrm{Z}\cdot\mathrm{P}=\mathrm{X}\cdot(\mathrm{Y}\cdot(1+\mathrm{Z})+\mathrm{Z}\cdot\mathrm{P})=\mathrm{X}\cdot(\mathrm{Y}+\mathrm{Z}\cdot\mathrm{P}). \quad (6.6)$$

Для упрощения в (6.7) к отрицаниям неэлементарных формул применяется правило де Моргана; используются законы двойного отрицания и склеивания:

$$X + \overline{Y \cdot \bar{Z}} + \overline{\bar{X} + Y + \bar{Z}} = X + \bar{Y} + \bar{\bar{Z}} + \bar{\bar{X}} \cdot \bar{Y} \cdot \bar{\bar{Z}} = \\ X + \bar{Y} + Z + X \cdot \bar{Y} \cdot Z = X + Z + (\bar{Y} + X \cdot \bar{Y} \cdot Z) \\ = \mathrm{X} + \mathrm{Z} + \bar{\mathrm{Y}}. \tag{6.7}$$

Для упрощения в (6.8) общий множитель x выносится за скобки, комбинируются слагаемые в скобках — первое с третьим и второе с четвертым, к дизъюнкции $Y \cdot Z + \overline{Y \cdot Z}$ применяется правило операции переменной с её инверсией:

$$X \cdot \bar{Y} + X \cdot Y \cdot Z + X \cdot \bar{Y} \cdot Z + X \cdot \overline{Y \cdot Z} = \\ X(\bar{Y} + Y \cdot Z + \bar{Y} \cdot Z + \overline{Y \cdot Z}) = \\ \mathrm{X} \cdot ((\bar{\mathrm{Y}} + \bar{\mathrm{Y}} \cdot \mathrm{Z}) + (\mathrm{Y} \cdot \mathrm{Z} + \overline{\mathrm{Y} \cdot \mathrm{Z}}) = \mathrm{X} \cdot (\bar{\mathrm{Y}} + \bar{\mathrm{Y}} \cdot \mathrm{Z} + 1) = \mathrm{X} \cdot 1 = \mathrm{X}. \tag{6.8}$$

Для упрощения в (6.9) используются распределительный закон для дизъюнкции, правило операции переменной с ее инверсией, правило операций с константами, переместительный закон и распределительный закон для конъюнкции:

$$(X \cdot \bar{Y} + Z) \cdot (\bar{X} + Y) + \bar{Z} = \\ X \cdot \bar{Y} \cdot \bar{X} + X \cdot \bar{Y} \cdot Y + Z \cdot \bar{X} + Z \cdot Y + \bar{Z} = \\ 0 + 0 + Z \cdot \bar{X} + Z \cdot Y + \bar{Z} = Z \cdot \bar{X} + (Z + \bar{Z}) \cdot (Y + \bar{Z}) = \\ Z \cdot \bar{X} + 1 \cdot (Y + \bar{Z}) = Z \cdot \bar{X} + Y + \bar{Z} = (Z \cdot \bar{X} + \bar{Z}) + Y = \\ (Z + \bar{Z}) \cdot (\bar{X} + \bar{Z}) \cdot Y = 1 \cdot (\bar{X} + \bar{Z}) + Y \\ = \bar{X} + \bar{Z} + Y. \tag{6.9}$$

Для упрощения в (6.10) используются правило де Моргана, закон двойного отрицания и закон поглощения:

$$X \cdot Y\left(\bar{X} \cdot Z + \overline{\overline{X \cdot Y} \cdot Z} + Z \cdot T\right) = X \cdot Y \cdot \left(\bar{X} \cdot Z + \overline{\overline{X \cdot Y}} + \bar{Z} + Z \cdot T\right) = \\ X \cdot Y \cdot \ (\bar{X} \cdot Z + X \cdot Y + \bar{Z} + Z \cdot T) X \cdot Y + X \cdot Y \cdot \bar{Z} + X \cdot Y \cdot Z \cdot T = X \cdot Y. \tag{6.10}$$

Из этих примеров видно, что при упрощении логических формул не всегда очевидно, какой из законов алгебры логики следует применить на том или ином шаге. Навыки приходят с опытом.

Лекция №7. Анализ и синтез логических устройств. Минимизация количества логических элементов

Цели лекции:

- получить понятия о номенклатуре логических элементов и ее минимизации;
- освоить правила реализации логических функций с помошью комбинированных элементов.

Содержание лекции:

- реализация логических функций элементами «И-НЕ», «ИЛИ-НЕ»;
- переход из базиса «И», «ИЛИ», «НЕ» в базис «ИЛИ-НЕ», а также в базис «И-НЕ»;
- преобразования логических формул для минимизации логических схем.

При проектировании всегда стремятся сузить количество и номенклатуру элементов. В связи с этим созданы логические элементы, способные выполнить простейшую функцию двух аргументов «ИЛИ-НЕ», а также «И-НЕ». С помощью каждого из этих элементов можно выразить все основные операции булевой алгебры, а значит реализовать любую логическую функцию. Покажем это.

Для элемента «ИЛИ-НЕ».

операция «НЕ» $\bar{x} = \overline{x + x}$

операция «ИЛИ» $x + y = \overline{\overline{x + y}}$

операция «И» $x \cdot y = \overline{\bar{x} + \bar{y}}$

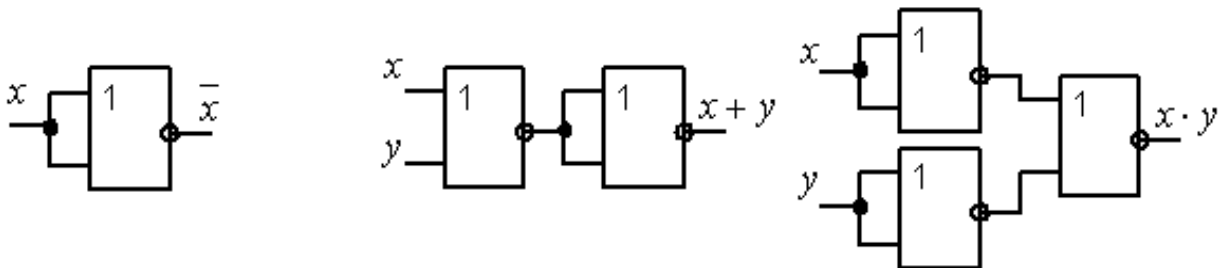


Рисунок 7.1 – Функции двух аргументов для «ИЛИ-НЕ»

Для элемента «И-НЕ».

операция «НЕ» $\bar{x} = \overline{x \cdot x}$

операция «ИЛИ» $x + y = \overline{\bar{x} \cdot \bar{y}}$

операция «И» $x \cdot y = \overline{\overline{x \cdot y}}$

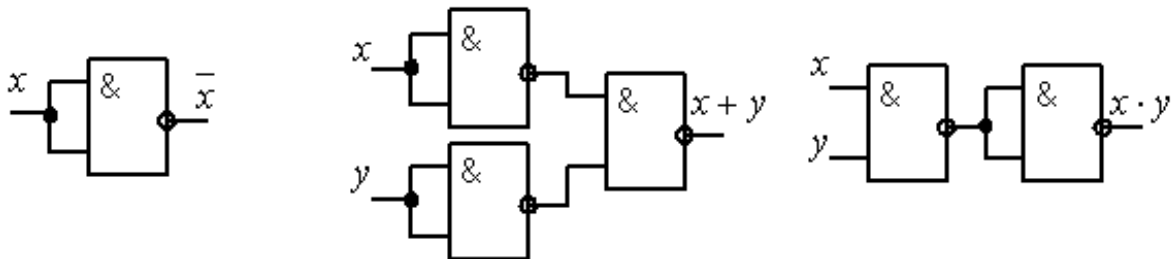


Рисунок 7.2 – Функции двух аргументов для «И-НЕ»

В микросхемном исполнении элементы «ИЛИ-НЕ» обозначаются индексами ЛЕ, элементы «И-НЕ» – индексами ЛА. Например, микросхема К555 ЛЕ1 имеет в своем составе четыре элемента «ИЛИ-НЕ» на два входа каждый.

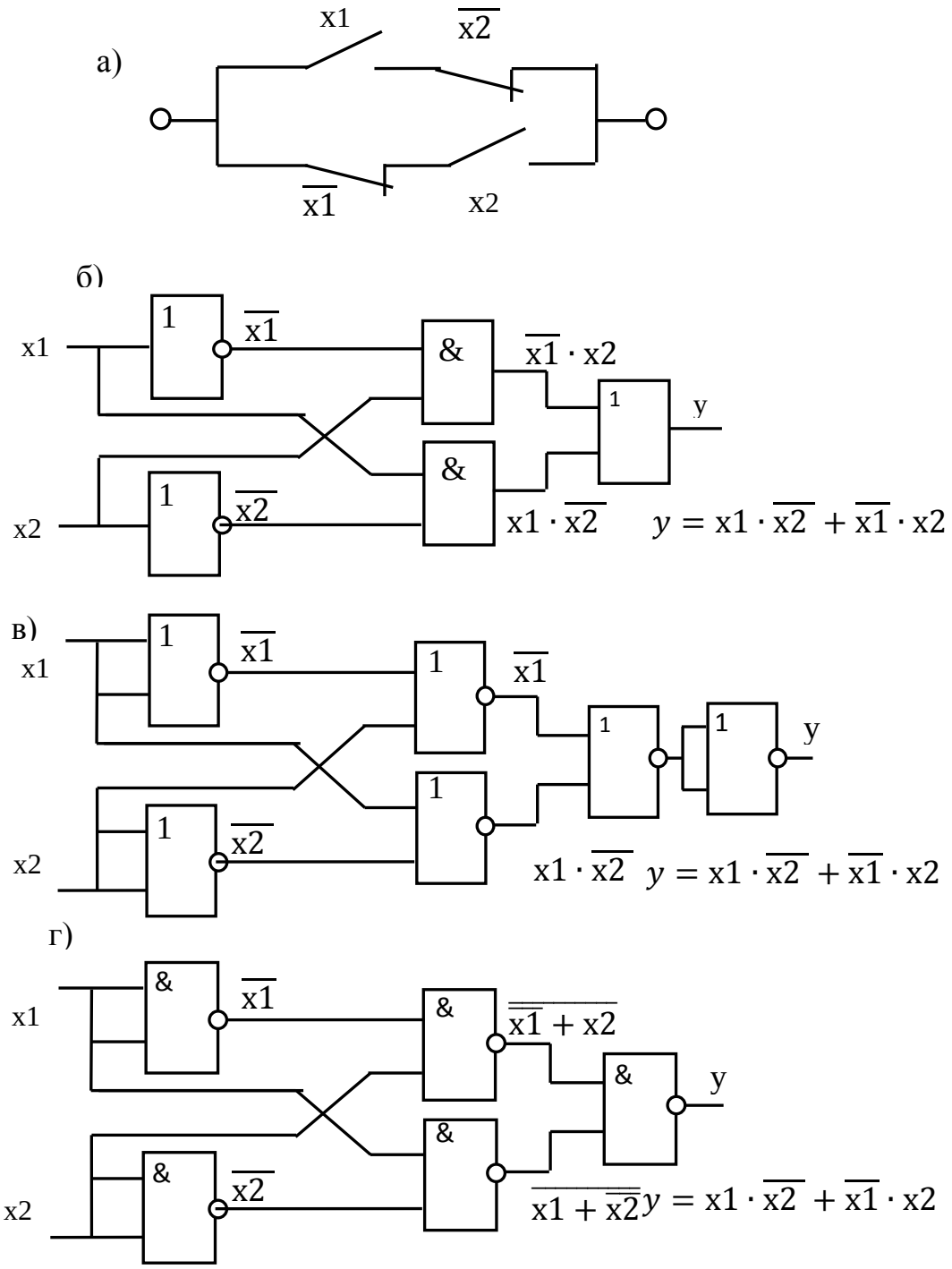
Для перехода из базиса «И», «ИЛИ», «НЕ» в базис «ИЛИ-НЕ», а также в базис «И-НЕ» также выполняется преобразование логической формулы с использованием двойного отрицания. Рассмотрим пример перехода для релейной схемы на рисунке 7.3, а, реализованной в базисе «И», «ИЛИ», «НЕ» (рисунок 7.3, б), в базис «ИЛИ-НЕ» (рисунок 7.3, в) [1,4,6]:

$$y = \mathrm{x1} \cdot \overline{\mathrm{x2}} + \overline{\mathrm{x1}} \cdot \mathrm{x2} = \overline{\overline{\mathrm{x1} \cdot \overline{\mathrm{x2}}}} + \overline{\overline{\overline{\mathrm{x1}} \cdot \mathrm{x2}}} = \overline{\overline{\mathrm{x1}} + \mathrm{x2}} + \overline{\mathrm{x1} + \overline{\mathrm{x2}}}$$
$$= \overline{\overline{\overline{\mathrm{x1}} + \mathrm{x2}} + \overline{\mathrm{x1} + \overline{\mathrm{x2}}}}$$

и в базис «И-НЕ» (рисунок 7.3, г):

$$y = \mathrm{x1} \cdot \overline{\mathrm{x2}} + \overline{\mathrm{x1}} \cdot \mathrm{x2} = \overline{\overline{\mathrm{x1} \cdot \overline{\mathrm{x2}} + \overline{\mathrm{x1}} \cdot \mathrm{x2}}} =$$
$$\overline{\overline{(\mathrm{x1} \cdot \overline{\mathrm{x2}})} \cdot \overline{(\overline{\mathrm{x1}} \cdot \mathrm{x2})}}.$$

Количество черточек сверху формул равно количеству элементов отрицания, т.е. элементов «ИЛИ-НЕ» и «И-НЕ». В первой формуле шесть отрицаний, и соответственно схема на рисунке 7.3, в содержит шесть элементов «ИЛИ-НЕ». Во второй формуле пять отрицаний, и соответственно схема на рисунке 7.3, г содержит пять элементов «И-НЕ».



а) на релейных элементах; б) на элементах «ИЛИ», «И», «НЕ»; в) на элементах «ИЛИ-НЕ»; г) на элементах «И-НЕ».

Рисунок 7.3 - Реализация структурной формулы логического элемента

Ниже приведены два примера упрощения релейных схем [1,4,6]. Пусть необходимо упростить выражение:

$$f = (x + y) \cdot (x + z) \tag{7.1}$$

и начертить релейный эквивалент до упрощения и после него. Здесь f — выходной сигнал (состояние замыкающего контакта) релейного элемента F. До упрощения релейный эквивалент в соответствии с заданным выражением выглядит следующим образом.

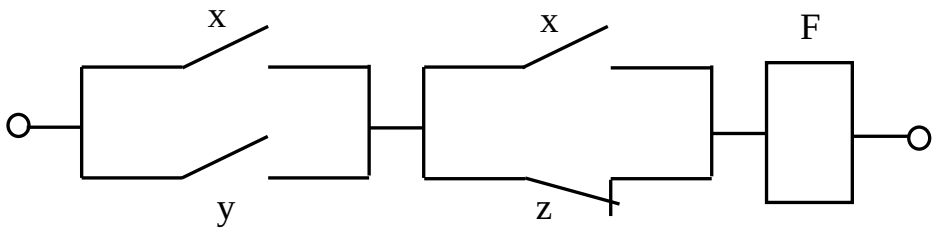


Рисунок 7.4 –Релейный элемент до упрощения

Упростим заданное выражение в соответствии с законами алгебры логики:

$$f = (x + y) \cdot (x + z) = x \cdot x + x \cdot y + x \cdot z + y \cdot z. \qquad (7.2)$$

Учитывая, что $x \cdot x = x$, запишем:

$$f = x + x \cdot y + x \cdot z + y \cdot z = (1 + y + z) + y \cdot z. \qquad (7.3)$$

Учитывая, что

$$1 + y + z = 1$$

окончательно запишем:

$$f = x + y \cdot z \qquad (7.4)$$

После упрощения релейный эквивалент выглядит следующим образом.

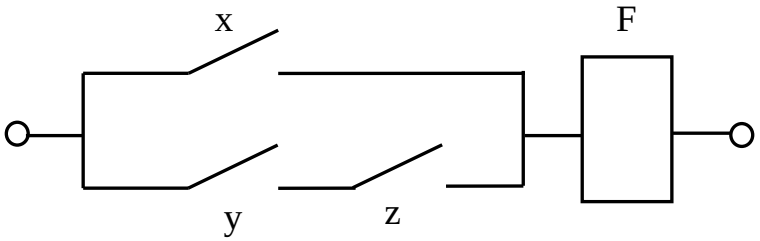


Рисунок 7.5 – Упрощенный релейный элемент

Упростим выражение:

$$f = x \cdot y + \bar{x} \cdot y \cdot \bar{z} + y \cdot z. \qquad (7.5)$$

До упрощения релейный эквивалент в соответствии с заданным выражением выглядит следующим образом:

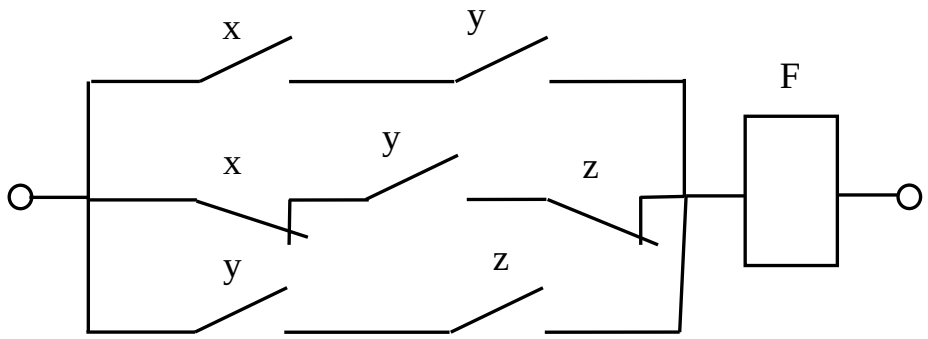


Рисунок 7.6 – Релейный элемент до упрощения

Упростим заданное выражение в соответствии с законами алгебры логики, вынося общий множитель за скобки:

$$f = x \cdot y + \bar{x} \cdot y \cdot \bar{z} + y \cdot z = y \cdot (x + \bar{x} \cdot \bar{z} + z). \qquad (7.6)$$

Релейно-контактная схема этого выражения примет вид:

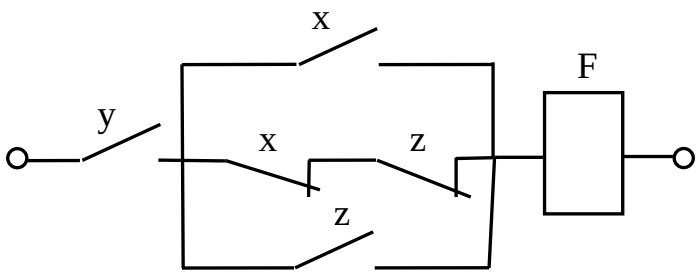


Рисунок 7.7 – Упрощенный релейный элемент

Далее преобразуем полученное выражение:

$$f = y \cdot (x + \bar{x} \cdot \bar{z} + z) = y \cdot (x + z + \overline{x + z}). \qquad (7.7)$$

Здесь учтено, что:

$$\bar{x} \cdot \bar{z} = \overline{x + z}\,. \qquad (7.8)$$

Далее из закона инверсии:
После преобразования упрощенное выражение примет вид:

$$f = y. \tag{7.9}$$

После упрощения выражения релейный эквивалент выглядит так:

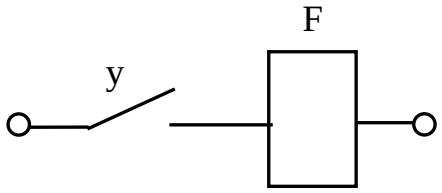


Рисунок 7.8 – Упрощенный релейный элемент

Лекция №8. Дизъюнктивная и конъюнктивная формы алгебраических логических функций. Стандартная минимизация булевых функций

Цели лекции:

- понять наличие и суть двух форм алгебраических функций – дизъюнктивной и конъюнктивной;
- научиться применять карту Карно для минимизации логических функций.

Содержание лекции:

- две формы алгебраических функций – дизъюнктивная и конъюнктивная;
- избыточность дизъюнктивная и конъюнктивная форм и их минимизация;
- минимизация логических схем с применением карт Карно.

Алгебраическая форма представления булевых функций используется для минимизации (упрощения формул) и для построения логических схем [1,4,6]. Существует две формы алгебраических функций – дизъюнктивная и конъюнктивная. Дизъюнктивная нормальная форма представляет собой сумму элементарных произведений аргументов, например:

$$F(x1, x2, x3) = x1 \cdot \overline{x2} + x2 \cdot x3 + \overline{x1} \cdot \overline{x2} \cdot \overline{x3}. \tag{8.1}$$

Если каждое слагаемое содержит все аргументы или их отрицания, то получаем совершенную дизъюнктивную нормальную форму (СДФН), например:

$$F(x1, x2, x3) = x1 \cdot \overline{x2} \cdot x3 + \overline{x1} \cdot x2 \cdot \overline{x3} + \overline{x1} \cdot \overline{x2} \cdot x3. \tag{8.2}$$

Совершенная конъюнктивная нормальная форма (СКНФ) представляет собой логическое произведение элементарных логических сумм, причем каждая сумма содержит все аргументы или их отрицания, например:

$$F(x1, x2, x3) = (\overline{x1} + x2 + \overline{x3}) \cdot (x1 + \overline{x2} + x3) \cdot (\overline{x1} + \overline{x2} + x3). \quad (8.3)$$

Для перехода от таблицы истинности к СДНФ учитываются только те состояния, для которых функция равна «1». Для каждого такого состояния записывается элементарное произведение всех аргументов. Если аргумент имеет значение «0», то записывается его отрицание. Для приведенного примера СДНФ имеет вид:

$$F(x1, x2, x3) = (\overline{x1} \cdot x2 \cdot x3) + (x1 \cdot \overline{x2} \cdot x3) + (x1 \cdot x2 \cdot \overline{x3}) + x1 \cdot x2 \cdot x3. \quad (8.4)$$

Для перехода от таблицы истинности к СДНФ учитываются только те состояния, для которых функция равна «0». Для каждого такого состояния записывается элементарная сумма аргументов. Если аргумент имеет значение «1», то пишется его отрицание. Для приведенного примера СКНФ имеет вид:

$$F(x1, x2, x3) = (x1 + x2 + x3) \cdot (x1 + x2 + \overline{x3}) \cdot (x1 + \overline{x2} + x3) \cdot (\overline{x1} + x2 + x3). \quad (8.5)$$

На основании полученных формул (8.1) или (8.2) можно построить логическую схему, состоящую из элементов «ИЛИ», «И», «НЕ». Для функции (8.1) сначала изображаются инверторы, затем-ячейки «И» и потом ячейки «ИЛИ» (см. рисунок 8.1).

Булевы функции в СДНФ и в СКНФ обычно избыточны. Поэтому этапу построения схемы должно предшествовать упрощение формул или минимизация. Цель минимизации – получить минимально необходимое количество логических элементов в схеме. В основу минимизации положены правила и законы булевой алгебры. Чаще других применяется теорема склеивания:

$$x \cdot \bar{y} + x \cdot y = x. \quad (8.6)$$

Для применения этой теоремы в функции, представленной в СДНФ отыскиваются слагаемые, отличающиеся только одним аргументом, и склеиваются. Когда все операции склеивания выполнены, можно проверить возможность применения закона поглощения. Для примера проведем минимизацию функции (8.1). Добавим в выражение (8.1) еще два слагаемых x1*x2*x3. От этого значение функции не изменится (правило 3):

$$F = \overline{x1} \cdot x2 \cdot x3 + x1 \cdot \overline{x2} \cdot x3 + x1 \cdot x2 \cdot \overline{x3} + x1 \cdot x2 \cdot x3 + x1 \cdot x2 \cdot \quad x3 + x1 \cdot \quad x2 \cdot x3. \quad (8.7)$$

Проведем группирование и возможные склеивания:

$$F = (\overline{x1} \cdot x2 \cdot x3 + x1 \cdot x2 \cdot x3) + (x1 \cdot \overline{x2} \cdot x3 + x1 \cdot x2 \cdot x3) + (x1 \cdot x2 \cdot \overline{x3} + x1 \cdot x2 \cdot x3). \quad (8.8)$$

Вместо четырех слагаемых третьего ранга (8.1) получили три слагаемых второго ранга. Схема, соответствующая (8.3), приведена на рисунке 8.3.

В инженерной практике для минимизации логических функций, как правило, применяют карты (матрицы) Карно. Карта Карно представляет прямоугольник, разбитый на квадраты. Число квадратов равно числу возможных комбинаций, т. е. $N=2^n$. Каждый квадрат соответствует определенной комбинации аргументов (рисунок 8.4,а). Комбинации соседних квадратов должны отличаться не более чем одним аргументом. Для примера на рисунке 8.4,а все возможные комбинации функции F(x1,x2,x3) внесены в квадраты карты Карно.

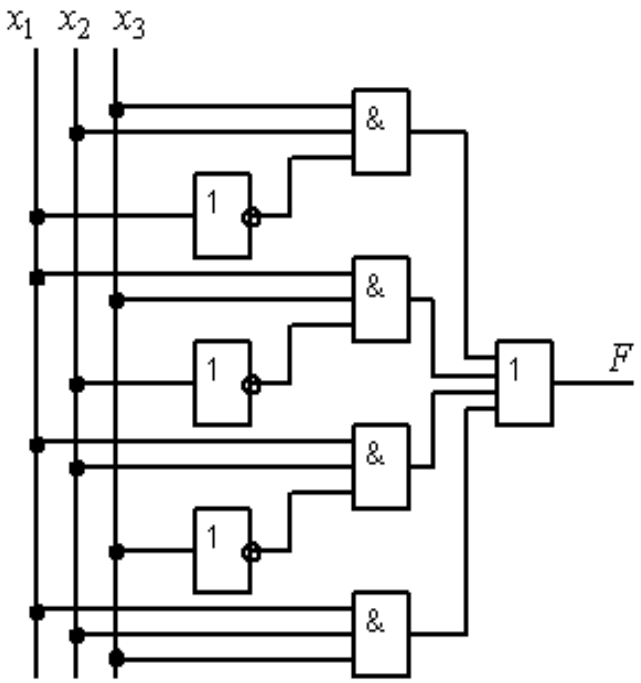


Рисунок 8.2

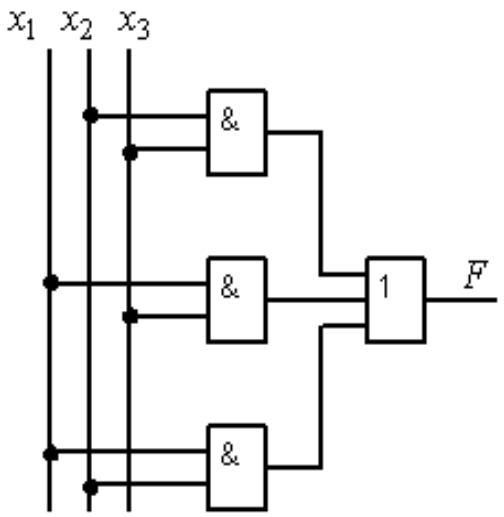


Рисунок 8.3

В целях минимизации карта Карно заполняется «1» и «0». Знак «1» записывается в тот квадрат, комбинация которого соответствует значению F = 1. В остальные квадраты записываются «0» (рисунок 8.4,б). После заполнения квадраты с «1» объединяют в контуры.

x_3 \ x_2x_1	00	01	11	10
0	$\bar{x}_1\bar{x}_2\bar{x}_3$	$x_1\bar{x}_2\bar{x}_3$	$x_1x_2\bar{x}_3$	$\bar{x}_1x_2\bar{x}_3$
1	$\bar{x}_1\bar{x}_2\bar{x}_3$	$x_1\bar{x}_2x_3$	$x_1x_2x_3$	$\bar{x}_1x_2x_3$

а)

x_3 \ x_2x_1	00	01	11	10
0	0	0	1	0
1	0	1	1	1

б)

Рисунок 8.4

Объединить можно 2, 4, 8 квадратов и т. д. Это равносильно объединению слагаемых функции для склеивания. Каждый квадрат может входить в несколько соседних контуров. Возможно объединение крайних квадратов на противоположных сторонах карты.

Объединением двух квадратов исключается один аргумент, четырех квадратов – два аргумента и т. д. В минимизированном выражении функции остаются только те аргументы, значение которых одинаково во всех квадратах контура. Например, для рисунка 8.4,б результат минимизации будет иметь вид:

$$F(x1, x2, x3) = x1 \cdot x2 + x1 \cdot x3 + x2 \cdot x3 \ ,$$

и полностью совпадает с выражением (8.3).

Лекция №9. Основные принципы проектирования релейных схем в соответствии с заданным алгоритмом их работы.

Цели лекции:

- научиться анализировать существующие электрические схемы автоматов;
- научиться составлять электрические схемы в соответствии с заданным алгоритмом.

Содержание лекции:

- проектирование релейных схем;
- применение законов булевой алгебры при анализе релейных схем;
- применение законов булевой алгебры при проектировании релейных схем.

На основе изученных законов и правил алгебры логики можно разрабатывать структурные формулы, а по ним типовые контактные схемы. Задача может состоять в составлении структурной формулы для анализа готовой схемы или синтезе по готовой структурной формуле электрической схемы будущего автомата. Наиболее часто встречающиеся варианты

контактных схем и их эквиваленты, рассмотренные ранее, помогут при составлении электрических схем автоматов, а также анализировать уже готовые конструкции, например, в процессе их ремонта.

Рассмотрим схему, приведенную на рисунке 9.1, а [1,4,6]. Соответствующая ей структурная формула имеет вид:

$$(A + B) \cdot (C + D).$$

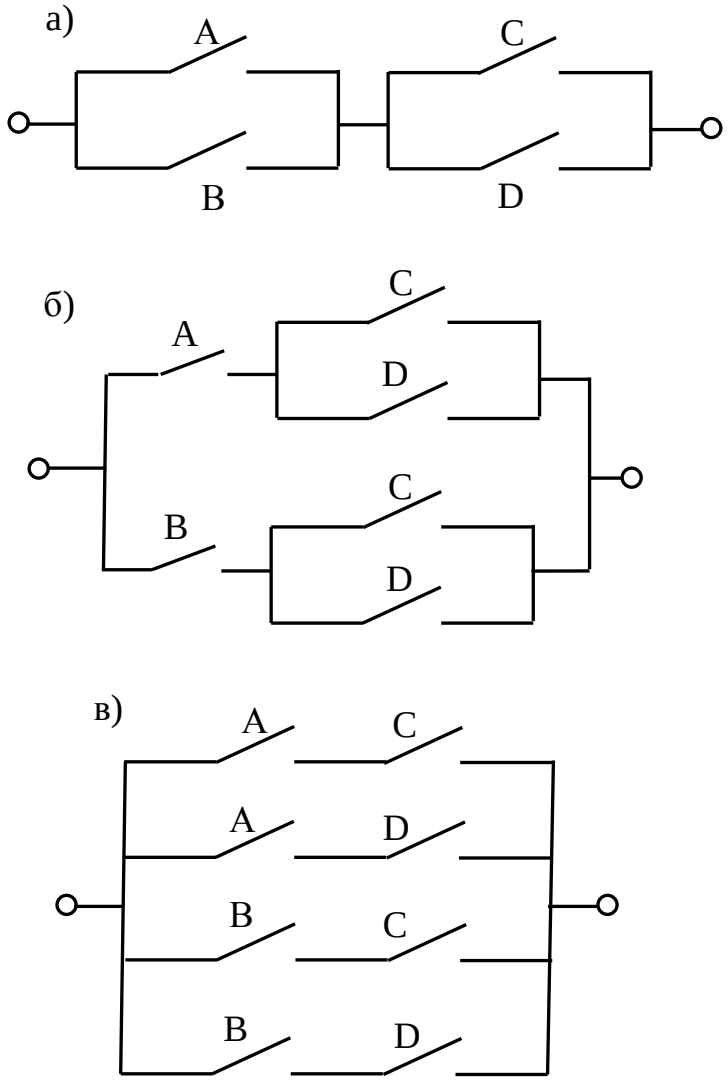


Рисунок 9.1 – Электрические схемы автоматов

Пользуясь распределительным законом алгебры Буля, раскроем скобки в этом выражении и получим:

$$A \cdot (C + D) + B \cdot (C + D),$$

что соответствует схеме, изображенной на рисунке 9.1, б. Далее, за счет перемножения, можем получить формулу

$$A \cdot C + A \cdot D + B \cdot C + B \cdot D,$$

соответствующую рисунку 9.1, в.

Все три схемы эквивалентны, то есть оказываются замкнутыми при одних и тех же условиях. Однако по сложности они разные.

Первая из схем самая простая: она требует четырех реле, каждое из которых должно иметь по одному нормально разомкнутому контакту (для упрощения рисунков катушки реле не показаны).

Схема «б» требует реле с двумя контактными группами. Собственно, основной задачей алгебры контактных схем является отыскание всех эквивалентных схем с тем, чтобы можно было выбрать из них наиболее простую.

Перейдем к построению схем по данным условиям их работы с помощью релейной алгебры [1,4,6].

Как и прежде, условия работы схемы сначала всегда задаются словесно. Конструктор, прежде всего, должен уметь выразить словами то, что хочет. Если такой ясности у него нет, то никакая алгебра не поможет. Начинать нужно всегда с четкой формулировки требований, которые ставятся перед новой схемой. Как и в любом деле, эта задача, пожалуй, самая сложная. Если условия достаточно просты, то мы можем сразу же написать выражение структурной формулы, удовлетворяющей этим требованиям.

Допустим, что мы должны построить схему, содержащую 4 контакта A, B, C и D так, чтобы цепь была включена тогда, когда замкнуты контакт A, и какой-нибудь из остальных трех контактов. В этом простом случае работа схемы в словесной записи будет выглядеть так: «Схема должна проводить ток, если замкнуты контакты A и B, или контакты A и C, или контакты A и D. Структурная формула будет выглядеть так:

$$A \cdot B + A \cdot C + A \cdot D = 1,$$

или

$$A \cdot (B + C + D) = 1.$$

У схемы два варианта. Они показаны на рисунке 9.2. Второй вариант не требует реле с тремя нормально разомкнутыми контактами.

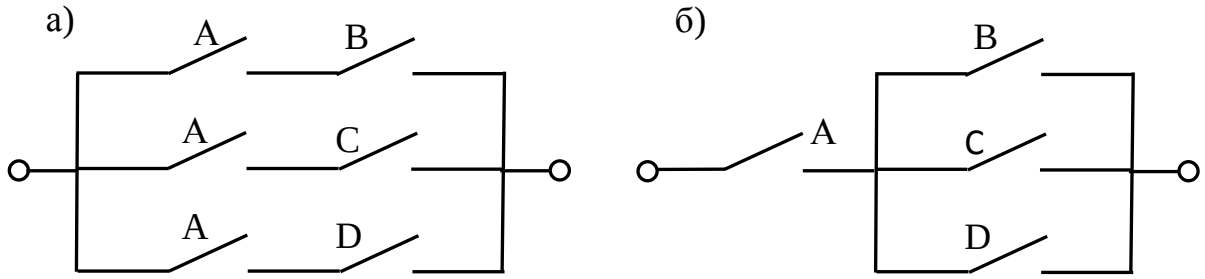


Рисунок 9.1 – Электрические схемы из трех контактов

Пусть вам нужно сконструировать переключатель, позволяющий включать свет при входе в подъезд и выключать после того, как вы поднялись на нужный этаж, или, наоборот, включать при выходе из квартиры и выключать после того, как спуститесь вниз. Задача сводится к управлению одной лампочкой из разных мест двумя переключателями.

Выберем следующий порядок решения задачи: сначала четко сформулируем условия работы переключателей, затем запишем их в виде формулы, и уже по ним начертим электрическую схему.

Чтобы лампочка горела (1), нужно, чтобы было выполнено одно из двух условий:

1) Включить переключатель внизу (A) и выключить наверху ($\overline{B}$). Входите в подъезд.

2) Включить переключатель наверху (B) и выключить внизу ($\overline{A}$) Выходите из квартиры.

Структурная формула запишется так:

$$A \cdot \overline{B} + \overline{A} \cdot B = 1.$$

Схема переключателя показана на рисунке 9.6. В настоящее время такие переключатели выпускаются промышленно, это так называемые проходные выключатели.

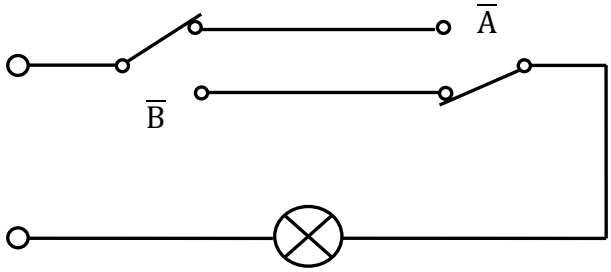


Рисунок 9.6 - Схема переключателя

Составим схему, позволяющую включать и выключать свет в помещении любым из трех различных выключателей, расположенных в разных местах. Рассуждая точно так же, как в случае двух переключателей получаем структурную формулу:

$$A \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C = 1.$$

Схема, составленная по этой формуле, представлена на рисунке 9.7.

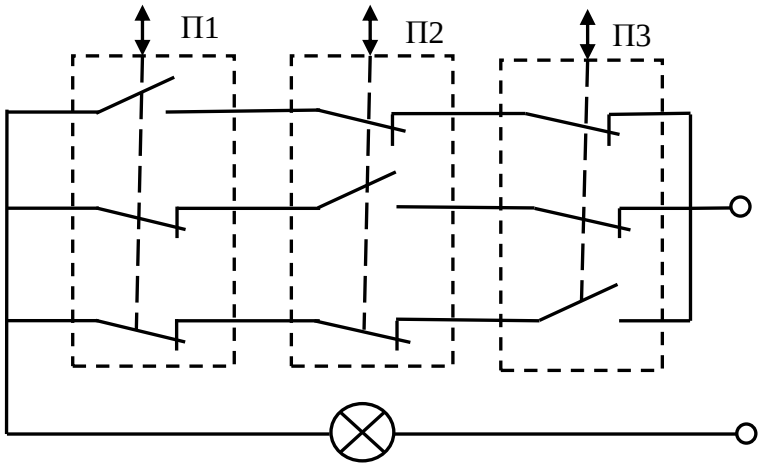


Рисунок 9.7 - Схема из трех различных выключателей

Лекция №10. Использование логических элементов в электрических схемах

Цели лекции:

- научиться читать релейные схемы и освоить их алгебраическое описание;
- научиться составлять релейные электрические схемы в контакторном и бесконтактном вариантах.

Содержание лекции:

- использование логических элементов в типовых электрических схемах;
- электрические схемы с применением базовых и комбинированных логических элементов.

Рассмотрим несколько узлов электрических схем, наиболее часто встречающихся в электротехнике [1,4,6]. На рисунке 10.1 показан узел питания катушки контактора К.

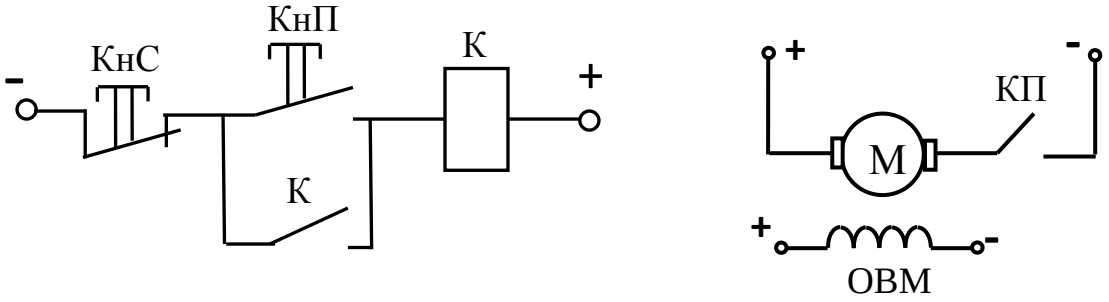


Рисунок 10.1 - Узел питания катушки контактора К

При нажатии кнопки КнП ток проходит по цепи и срабатывает контактор К. Его главный контакт КП подключает двигатель к сети, а контакт К, замыкаясь, шунтирует кнопку КнП. Теперь ток будет протекать по этим контактам, а кнопку КнП можно отпустить. Под действием пружины она размыкает свои контакты, но катушка будет продолжать питаться через контакты К. При нажатии кнопки КнС линия разрывается и контактор отпускается.

Этот узел можно выполнить на логических элементах (рисунок 10.2).

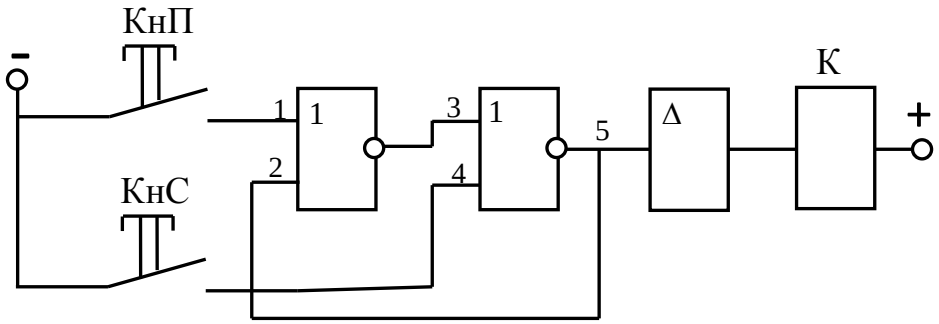


Рисунок 10.2 - Узел на логических элементах

В схему входят катушка контактора К, кнопки КнП и КнС, два логических элемента ИЛИ — НЕ и усилитель. Исходное состояние

X1 = 0 и X2 = 0,

тогда на выходе элемента 1 получим:

$$Y1 = \overline{X1 + X2} = \overline{0 + 0} = 1.$$

На выходе элемента 2

$$Y5 = \overline{X3 + X4} = \overline{1 + 0} = 0,$$

,

т. е. катушка обесточена, реле не срабатывает.

Если нажать КнП, то:

$$Y1 = \overline{X1 + X2} = \overline{1 + 0} = 0.$$

На выходе элемента 2

$$Y5 = \overline{X3 + X4} = \overline{0 + 0} = 1.$$

По катушке проходит ток и контактор срабатывает. Сигнал Y2 подается на вход X2 но Y1 от этого не меняется, так как:

$$Y1 = \overline{X1 + X2} = \overline{1 + 1} = 0.$$

Таким образом, катушка контактора находится под током.

Если нажать кнопку КнС, то на вход второго элемента будет подаваться сигнал:

$$X4 = 1,$$

$$Y2 = \overline{X3 + X4} = \overline{0 + 1} = 0,$$

и контактор отпускается.

Рассмотренная схема способна «запоминать» команды: сигнал Y2 остается неизменным, даже если отпустить кнопку.

Такую же функцию памяти можно осуществить с помощью триггера Т (рисунок 10.3).

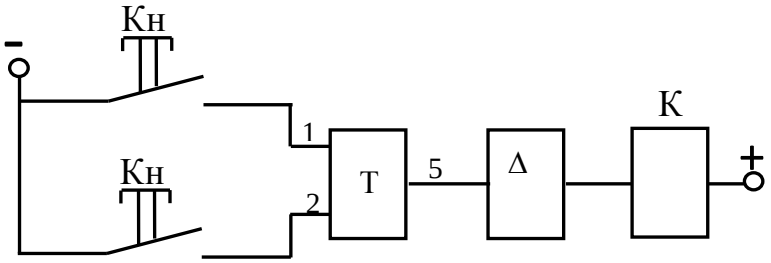


Рисунок 10.3 – Применение триггера

Если на вход подать сигнал X1 = 1, то на выходе появится сигнал Y = 1 и будет оставаться неизменным до тех пор, пока не нажмем кнопку КнС. Тогда триггер переключается и на выходе появляется сигнал Y = 0. Он будет оставаться без изменения до тех пор, пока снова не нажмем кнопку КнП.

На рисунке 10.4 показан узел электрической блокировки двух реле РВ (вперед) и РН (назад), исключающий их одновременное срабатывание, так как это приведет к короткому замыканию.

Действительно, при нажатой кнопке КнВ срабатывает реле РВ, а его блок-контакты размыкаются, и катушка РН не сможет оказаться под током, даже если нажать кнопку КнН. Отметим, что шунтирование замыкающих контактов кнопок здесь не предусмотрено, т. е. узел памяти отсутствует.

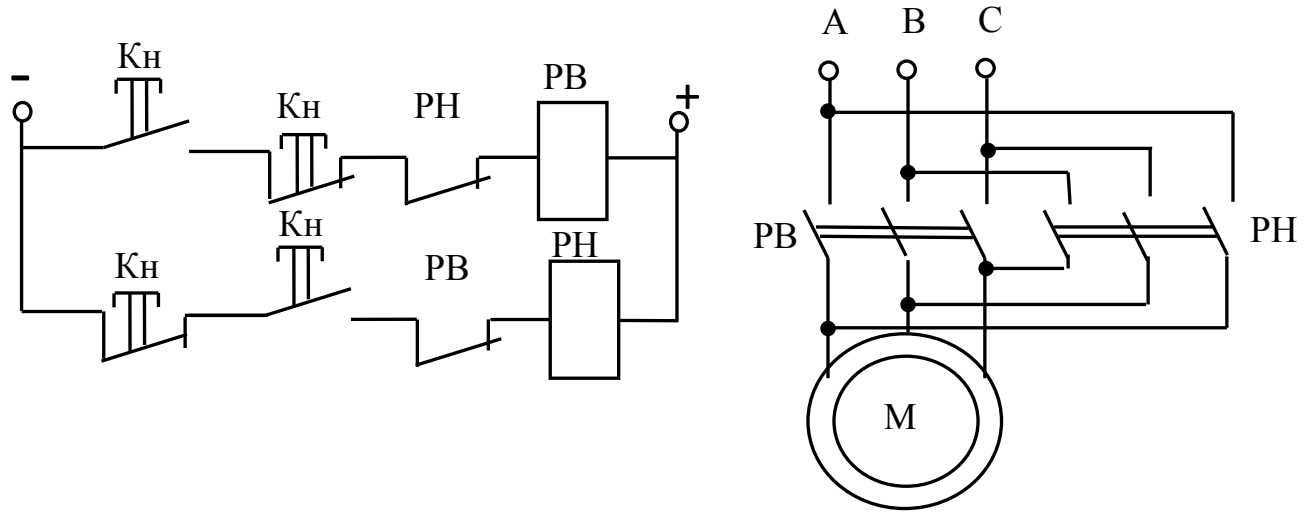


Рисунок 10.4 - Узел электрической блокировки двух реле РВ

В схеме с логическими элементами (рисунок 10.5) при нажатии кнопки КнВ на первом элементе «ИЛИ-НЕ» получим:

$$X1 = 1.$$

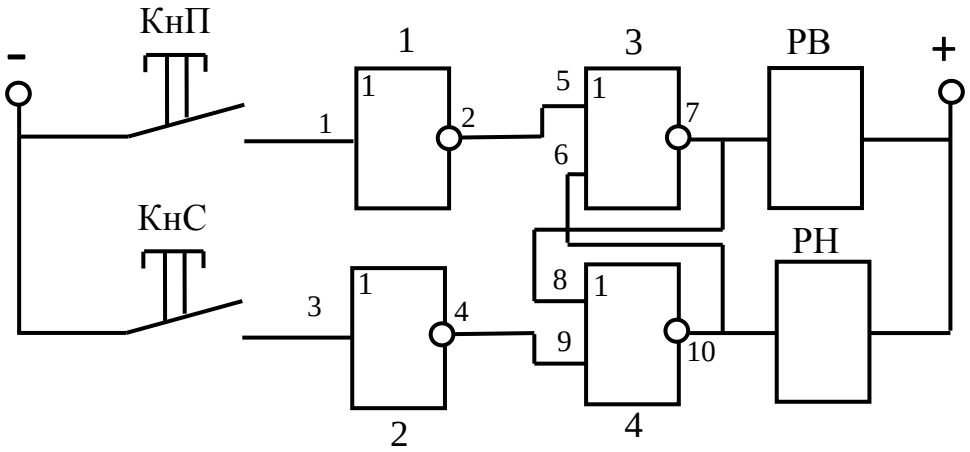


Рисунок 10.5 - Схема с логическими элементами

На втором элементе ИЛИ-НЕ:

$$Y7 = \overline{X5 + X6} = \overline{Y2 + X6} = \overline{0 + 0} = 1.$$

Реле РВ срабатывает и сигнал Y7 подается на вход элемента 4. На входе элемента 3 сигнал отсутствует, тогда:

$$Y4 = \overline{X2} = 1.$$

На четвертом элементе

$$Y10 = \overline{X8 + X9} = \overline{X8 + Y4} = \overline{1 + 1} = 0,$$

т. е. реле РН сработать не может, даже если нажать кнопку КнН. Тогда получим такой же результат:

$$Y10 = \overline{X8 + X9} = \overline{X8 + Y4} = \overline{1 + 0} = 0.$$

Лекция №11. Примеры использования логических элементов в электрических схемах

Цели лекции:

- научиться читать релейные схемы и освоить их алгебраическое описание;
- научиться составлять релейные электрические схемы в контакторном и бесконтактном вариантах.

Содержание лекции:

- использование логических элементов в типовых электрических схемах;
- электрические схемы с применением базовых и комбинированных логических элементов.

На рисунке 11.1 показан узел отпускания реле в случае нажатия кнопки КнС или размыкания контактов конечного выключателя ВК [1,4,6]. В схеме с логическими элементами в исходном положении:

$$Y3 = \overline{X1 + X2} = \overline{0 + 0} = 1,$$

т. е. катушка реле находится под током. При нажатии кнопки КнС получим:

$$Y3 = \overline{X1 + X2} = \overline{1 + 0} = 0$$

и реле отпуска:

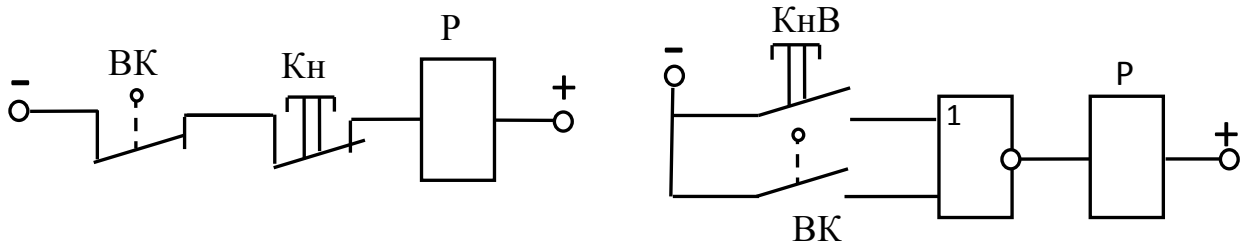


Рисунок 11.1 – Узел отпускания реле

На рисунке 11.2 показан узел включения реле в случае нажатия кнопки КнП при замкнутом контакте ВК. В схеме с логическими элементами в нормальном состоянии контактов получим:

$$Y7 = \overline{X6} = \overline{Y5} = \overline{\overline{X4}} = \overline{\overline{Y3}} = \overline{\overline{X1 \cdot X2}} = X1 \cdot X2.$$

В разомкнутом состоянии КнП и ВК:

$$Y7 = \overline{\overline{X1 \cdot X2}} = \overline{\overline{0 \cdot 0}} = 0.$$

Если нажата только кнопка КнП, то:

$$Y7 = \overline{\overline{X1 \cdot X2}} = \overline{\overline{1 \cdot 0}} = 0.$$

Если замкнут только контакт ВК, то:

$$Y7 = \overline{\overline{X1 \cdot X2}} = \overline{\overline{0 \cdot 1}} = 0.$$

При замыкании КнП и ВК получим:

$$Y7 = \overline{\overline{X1 \cdot X2}} = \overline{\overline{1 \cdot 1}} = 1.$$

Это означает, что реле срабатывает.

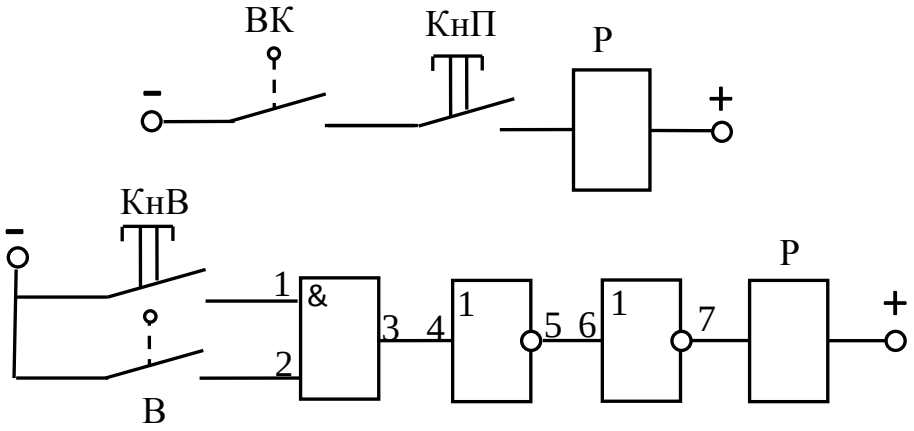


Рисунок 11.2 - Узел включения реле

На рисунке 11.3 показана схема управления двумя реле Р1 и Р2. При подаче напряжения на цепь срабатывает реле времени РВ, его контакты в линии 3 размыкаются мгновенно. Схема готова к работе. При нажатии кнопки КнП срабатывает реле Р1, его контакты замыкаются, шунтирут кнопку. Другие контакты в линии 2 размыкаются, а в линии 3 замыкаются. Реле РВ

отпускается и его контакты с выдержкой времени замыкаются, срабатывает реле Р2. Таким образом, после нажатия кнопки КнП реле Р1 срабатывает сразу, а Р2 — через некоторое время.

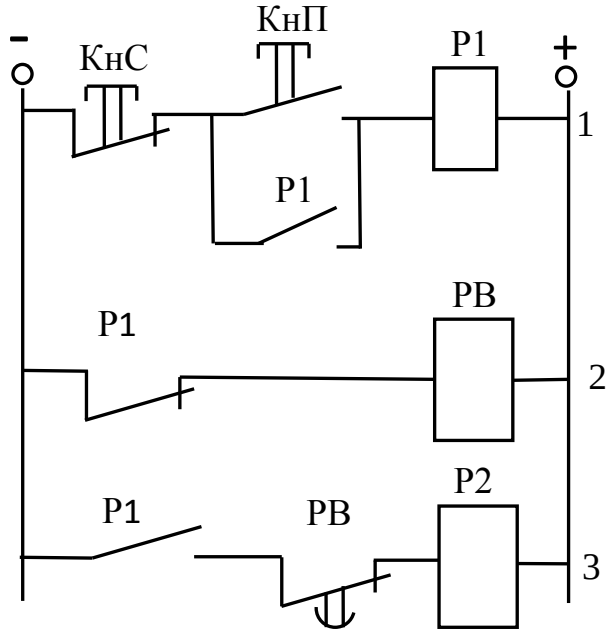


Рисунок 11.3 - Схема управления двумя реле Р1 и Р2

В схеме с логическими элементами (рисунок 11.4) узел «Память» построен на триггере. Пусть на выходе у него сигнал отсутствует (Y3 = 0), реле Р1 и Р2 обесточены. Нажимаем кнопку КнП, появляется сигнал на выходе триггера. Срабатывает реле Р1 и начинает отсчет времени элемент ЭВ.

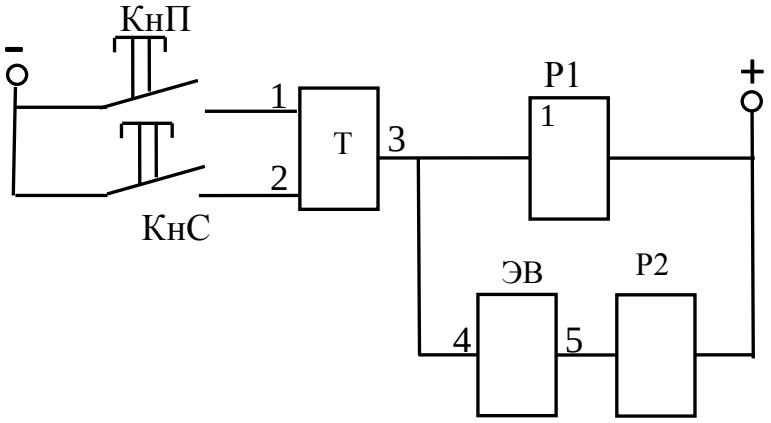


Рисунок 11.4 - Схема с логическими элементами

Когда появится сигнал Y5 = 1, срабатывает реле Р2. При нажатии кнопки КнС триггер переключается и тогда Y3 = 0. Реле Р1 и Р2 отпускаются.

Типовые узлы с логическими элементами широко применяют в более сложных схемах, причем такие схемы гораздо проще, чем схемы на релейно-контакторной аппаратуре.

Лекция №12. Триггеры. RS-триггер

Цели лекции:

- получить понятия о принципе действия и использовании триггеров в логических схемах;
- научиться анализировать возможные комбинации значений входов R S - триггера.

Содержание лекции:

- применение триггеров и виды триггеров;
- принцип действия и таблица истинности RS – триггера;
- реализация RS - триггера с помощью вентилей ИЛИ—НЕ.

Триггер — это электронная схема, имеющая два устойчивых состояния, одно из которых соответствует двоичной единице, а другое — двоичному нулю [1,4,6].

Термин *триггер* происходит от английского слова *trigger* — защёлка, спусковой крючок. Для обозначения этой схемы в английском языке чаще употребляется термин *flip-flop,* что в переводе означает «хлопанье». Это звукоподражательное название электронной схемы указывает на её способность почти мгновенно переходить («перебрасываться») из одного электрического состояния в другое и наоборот.

Самый распространённый тип триггера — так называемый RS-триггер (S и R, соответственно, от английских *set* — установка, и *reset* — сброс). Условное обозначение триггера — на рисунке 12.1.

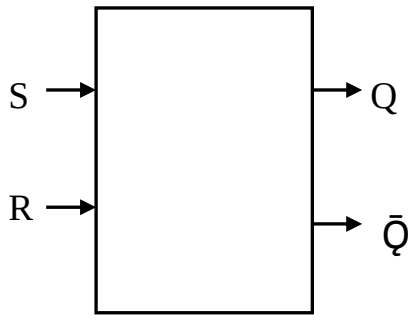


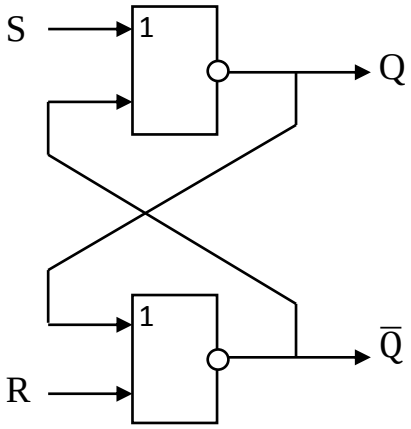
Рисунок 12.1 - RS-триггер

Он имеет два симметричных входа S и R и два симметричных выхода Q и Q̄, причем выходной сигнал Q является логическим отрицанием сигнала Q̄.

На каждый из двух входов S и R могут подаваться входные сигналы в виде кратковременных импульсов ( ).

Наличие импульса на входе будем считать единицей, а его отсутствие — нулем.

На рисунке 12.2 показана реализация триггера с помощью вентилей ИЛИ—НЕ и соответствующая таблица истинности.



S	R	Q	Q̄
0	0	запрещено	
0	1	1	0
1	0	0	1
1	1	хранение бита	

Рисунок 12.2 - Реализация триггера с помощью вентилей

Проанализируем возможные комбинации значений входов R и S триггера, используя его схему и таблицу истинности схемы ИЛИ—НЕ (таблица 5.5):

1) Если на входы триггера подать S=«1», R=«0», то (независимо от состояния) на выходе Q верхнего вентиля появится «0». После этого на входах нижнего вентиля окажется R=«0», Q= «0» и выход Q̄ станет равным «1».

2) Точно так же при подаче «0» на вход S и «1» на вход R на выходе Q̄ появится «0», а на Q — «1».

3) Если на входы R и S подана логическая «1», то состояние Q и Q̄ не меняется.

4) Подача на оба входа R и S логического «0» может привести к неоднозначному результату, поэтому эта комбинация входных сигналов запрещена.

Поскольку один триггер может запомнить только один разряд двоичного кода, то для запоминания байта нужно 8 триггеров, для запоминания килобайта, соответственно, $8x2^{10} = 8192$ триггеров. Современные микросхемы памяти содержат миллионы триггеров.

Лекция №13. Применение алгебры логики для синтеза САР

Цели лекции:

- получить понятия о создании релейных систем автоматического регулирования;
- научиться использовать временные диаграммы и алгебраические преобразования для синтеза САР.

Содержание лекции:

- применение алгебры логики для создания системы автоматического регулирования;
- анализ условий работы автоматической системы.

Рассмотрим пример применения алгебры логики для создания системы автоматического регулирования уровня воды в резервуаре Р (рисунок 13.1) [1,4,6]. Исполнительный механизм ИМ осуществляет подачу воды в резервуар путем полного открытия или закрытия подающего вентиля А. В резервуаре имеются два датчика уровня воды: датчик верхнего уровня В и датчик нижнего уровня Н. Когда уровень воды достигнет или превысит положение датчика, сигнал его становится равным единице. Если уровень воды опустится ниже уровня датчика, сигнал на его выходе становится равным нулю.

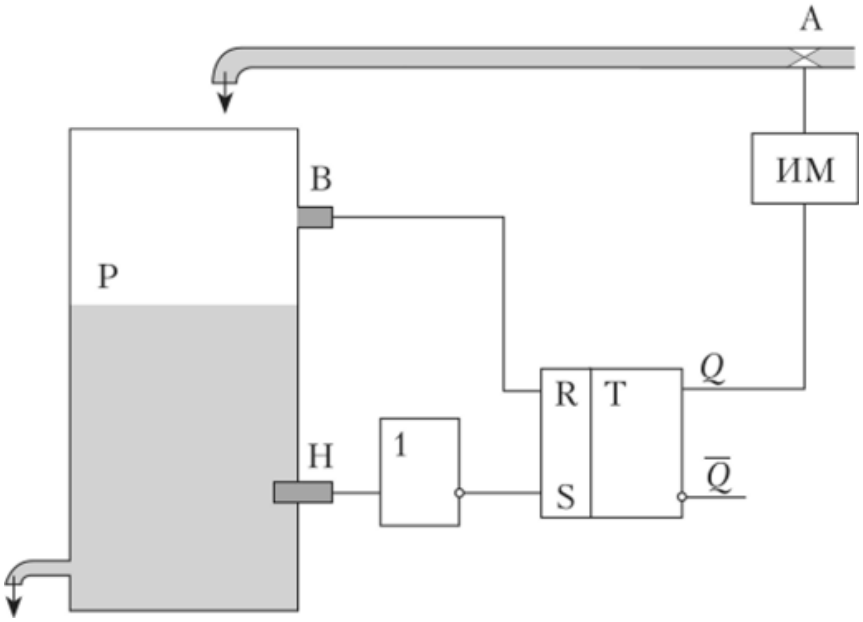


Рисунок 13.1 - Релейная система автоматического регулирования уровня воды в резервуаре

Проанализируем условия работы автоматической системы. Если уровень воды достигнет нижнего уровня Н, то необходимо включить подачу. Если уровень воды достигнет верхнего уровня В, то подачу необходимо отключить. Если уровень воды занимает промежуточное положение между В и Н, то подача должна остаться включенной, если она была включена от

датчика Н. Если же подача была выключена датчиком В, то она должна оставаться выключенной. Временная диаграмма сигналов с выхода датчиков и управляющего сигнала Q приведена на рисунке 13.2.

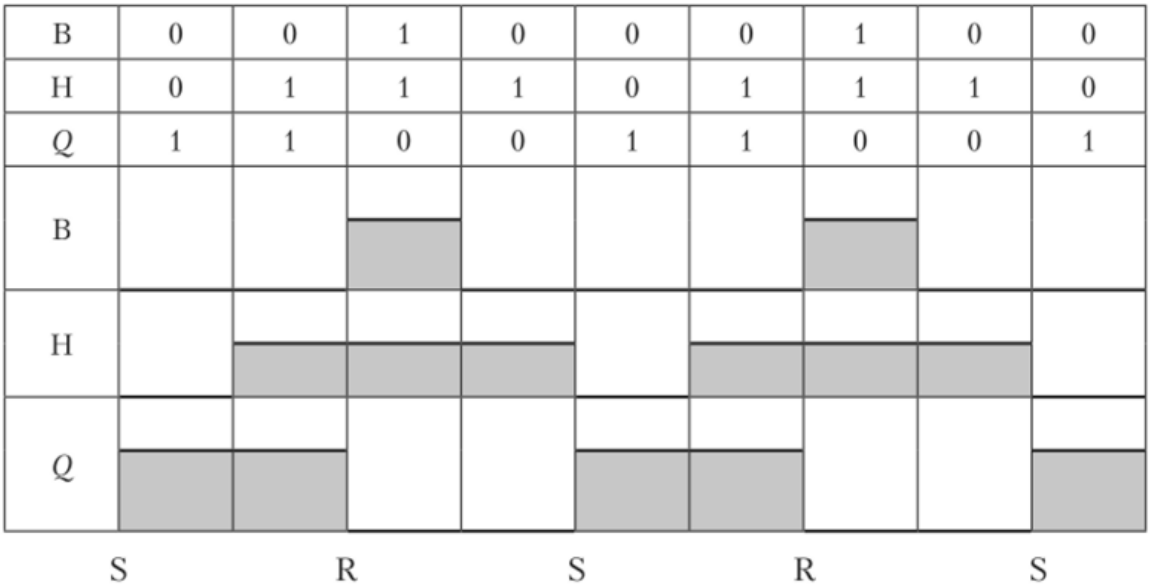


Рисунок 13.2 - Временные диаграммы сигналов в релейной САР

Условия работы, т.е. все комбинации входных сигналов и сигнала управления, переведены на язык алгебры логики и представлены на рисунке 13.2. в верхней таблице в виде единиц и нулей. В таблице указано, при каких соотношениях входных сигналов имеется или отсутствует сигнал Q на выходе релейной САР. Сигнал на выходе является результатом логических операций над входными сигналами.

Если по данным таблицы мы попытаемся записать условия работы в виде логических функций, то обнаружим, что включенному сигналу управления соответствуют два различных соотношения входных сигналов. То же относится и к выключенному сигналу управления. Получается неоднозначность выходного сигнала в зависимости от сочетания входных сигналов. При В = 0 и Н = 1 есть положение, когда Q = 0 и есть положение, когда Q=l. Это значит, что в схеме должен быть элемент памяти, в качестве которого можно использовать уже знакомый нам RS-триггер Т. Для включения триггера используем появление нулевого сигнала на выходе 11 (II = 0). Этот сигнал инвертируется и подается на устанавливающий вход S триггера Т. Поскольку сигнал В не изменяется, то его не будем учитывать и запишем условие для включения S = Н. Условия для сброса триггера и снятия сигнала управления записываем как R = В.

По этому же принципу строятся системы для регулирования температуры при охлаждении электрических машин и трансформаторов, а также силовых установок автомобилей и тракторов с помощью вентиляторов.

Схема может использоваться и для автоматического поддержания температуры за счет подогрева в жилых и животноводческих помещениях.

Лекция №14. Применение алгебры логики в релейной защите

Цели лекции:

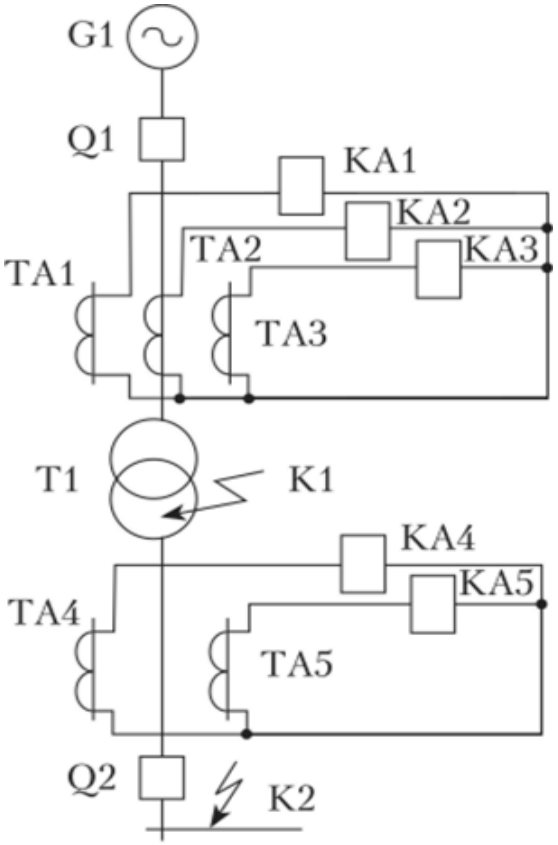
- получить понятия о релейной защите электротехнических объектов;
- уметь использовать алгебру логики в релейной защите.

Содержание лекции:

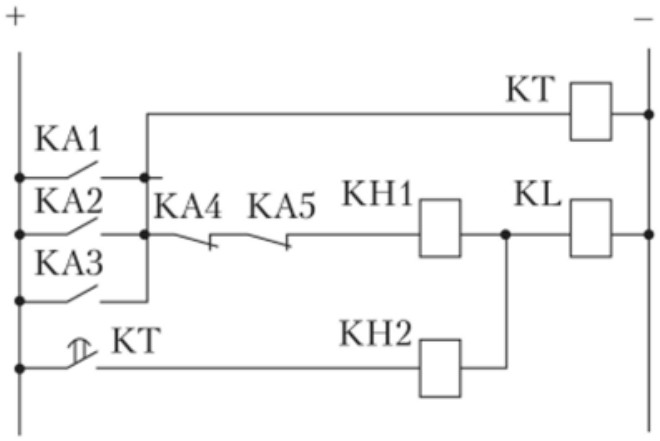
- применение алгебры логики в релейной защите;
- анализ системы логической релейной защиты силового трансформатора.

Рассмотрим еще один пример применения алгебры логики для создания логических релейных защит электротехнических объектов на примере релейной защиты силового трансформатора, приведенной на рисунке 14.1 [1,4,6].

Правила устройства электроустановок предусматривают для ответственных объектов основную и резервную защиту. Основная защита должна отключать объект без выдержки времени, а резервная — с выдержкой времени.



а



б

а — силовая схема; б — схема цепей защиты.

Рисунок 14.1 - Принципиальная схема максимальной токовой защиты трансформатора с временной селективностью

Основной защитой трансформатора Т1 при коротком замыкании в трансформаторе (КЗ в точке К1) служит дифференциальная релейная защита (на схеме она не показана). Резервной защитой при коротком замыкании на отходящих шинах подстанции за выключателем Q2 (КЗ в точке К2) служит максимальная токовая защита, действующая при срабатывании токовых реле КА1—КАЗ. Короткое замыкание в трансформаторе Т1 должно отключаться выключателем Q1 от действия резервной защиты без выдержки времени, т.е. «мгновенно». Короткое замыкание в точке К2 должно без выдержки времени отключаться выключателем Q2 (защита выключателя Q2 на схеме не показана). Если по каким-либо причинам защита, воздействующая на выключатель Q2 или сам выключатель Q2, не сработает, то от резервной защиты с выдержкой времени должен отключиться выключатель Q1.

Рассмотрим, как можно повысить быстродействие рассматриваемой резервной защиты, если КЗ произошло в трансформаторе и основная защита не сработала. Для этого измерительные органы ставят на входе и выходе трансформатора Т1. Они выполняют функцию определения места КЗ: на защищаемом объекте или на участке внешней сети. При КЗ на защищаемом объекте (КЗ в основной зоне) они разрешают работу резервной защиты без выдержки времени, а при внешнем КЗ они блокируют цепь мгновенного отключения, и защита работает как резервная с выдержкой времени.

Определение места КЗ выполняется следующим образом. При КЗ в Т1 (точка К1) трансформаторы тока ТА 1—ТАЗ обтекаются током КЗ, и срабатывают реле тока КА1—КАЗ. Трансформаторы тока ТА4—ТА5 на выходе трансформатора Т1 не обтекаются током КЗ. Реле тока КА4 и КА5 не срабатывают, их размыкающие контакты замкнуты. В такой ситуации защита

должна сработать без выдержки времени. Промежуточное реле KL подает сигнал на отключение выключателя Q1.

Условия работы промежуточного реле KL для отключения без выдержки времени словесно можно сформулировать так: реле KL сработает, если сработает реле КЛ1, ИЛИ сработает реле КА2 ИЛИ, сработает реле КАЗ И НЕ сработают реле КА4 И реле КА5.

В символах математической логики условие срабатывания реле KL записывается так:

$$KL = (KA1 + KA2 + KA3) \times (\overline{KA4} \times \overline{KA5}).$$

В выражении (4.12) и далее логические переменные в целях наглядности записаны прописными буквами в соответствии с обозначениями на схеме.

При КЗ на участке внешней сети (точка К2) трансформаторы тока ТА4 и ТА5 обтекаются током КЗ, что приводит к срабатыванию реле тока КА4 и КА5 и размыканию их размыкающих контактов в цепи релейной защиты без выдержки времени. Таким образом, работа защиты без выдержки времени блокируется. Резервная защита при КЗ в точке К2 работает с выдержкой времени.

Условие срабатывания реле времени резервной защиты формулируется словесно так: реле времени КТ сработает, если сработает реле КА1, ИЛИ сработает реле КА2, ИЛИ сработает реле КАЗ.

В символах математической логики условие срабатывания реле времени записывается как:

$$KT = (KA1 + KA2 + KA3).$$

Полностью условие срабатывания промежуточного реле KL, отключающего выключатель Q1 без выдержки времени и с выдержкой времени, записывается так:

$$KL = (KA1 + KA2 + KA3) \times (\overline{KA4} \times \overline{KA5}) + KT.$$

Срабатывание защиты с выдержкой времени фиксируется указательным реле КН.

Лекция №15. Использование алгебры логики в системах диагностирования и поиска неисправностей в электрических цепях

Цели лекции:

- получить понятия о техническом диагностировании объектов электроэнергетики;

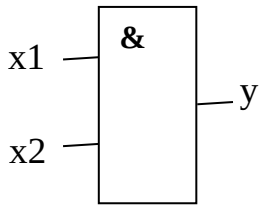
- понять перевод алгоритмов диагностирования объектов в математическую модель в виде логических формул.

Содержание лекции:

- использование алгебры логики в системах диагностирования;
- алгоритм поиска неисправностей в неразветвленных электрических цепях с релейными элементами;
- составление программы поиска неисправностей и анализ результатов с помощью прикладных компьютерных программ.

Процесс определения технического состояния объекта называется техническим диагностированием. Различают следующие виды технического состояния: исправное и неисправное, работоспособное и неработоспособное, а также состояния правильного и неправильного функционирования [1].

Для построения тестов и алгоритмов диагностирования необходимо иметь формальное описание объекта и его поведения в исправном состоянии. В качестве безусловного алгоритма тестового диагностирования приведем пример проверки исправности логического элемента «И», имеющего два входа x_x и x_2 и один выход y (рисунок 15.1, а). Логический единичный сигнал на выходе y этого элемента появится только тогда, когда будут присутствовать единичные логические сигналы на обоих его входах в соответствии с таблицей истинности, приведенной на рисунке 15.1, б. Таблица истинности является математическим описанием объекта диагностирования — логического элемента «И». Если на одном из входов будет нулевой сигнал, то сигнал на выходе также будет равен нулю. Проверка исправности элемента «И» осуществляется поочередной подачей на его входы трех тестовых воздействий: 01, 10 и 11 — и измерением при этом выходных значений y. Если полученные значения на выходе элемента совпадут с соответствующими значениями в правом столбце y таблицы, то элемент исправен, а если не совпадут, то неисправен.



n	x1	x2	y
1	0	1	0
2	1	0	0
3	1	1	1

Рисунок 15.1 - Алгоритм проверки исправности логического элемента «И»

Рассмотрим алгоритм поиска неисправностей в неразветвленных электрических цепях с релейными элементами. Пусть имеется участок цепи,

состоящий из последовательно включенных замыкающих контактов электромагнитных реле PI—P8 (рисунок 15.2), через которые подается сигнал на срабатывание реле P [1]. Устройство, в которое входит этот участок электрической цепи, будет работать только в том случае, если все восемь реле будут исправны, на катушки реле будут поданы напряжения и контакты этих реле окажутся замкнутыми. Если хотя бы один контакт не будет замкнут, то схема не будет работать. Схема имеет контрольные точки К1—К8, в которых можно измерять величину сигнала.

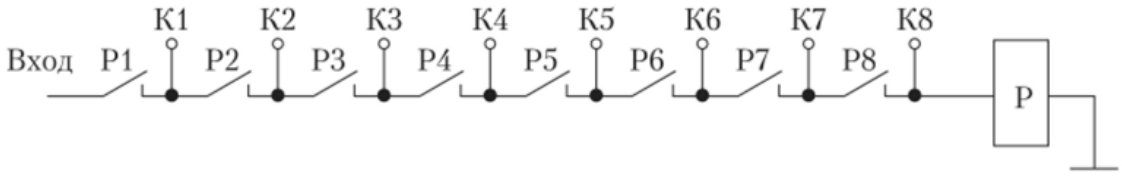


Рисунок 15.2 - Последовательно включенных замыкающих контактов электромагнитных реле PI—P8

Проверим работоспособность схемы, подав напряжение на катушки всех реле PI—P8. Пусть выходное реле P при этом не сработало и измерение показало, что в контрольной точке К8 напряжение отсутствует, т.е. сигнал в контрольной точке К8 равен нулю. Необходимо определить отказавший элемент цепи. Допустим, что произошел отказ реле Р8 и его контакты оказались разомкнутыми. Если мы начнем поиск неисправности с реле Р1, затем будем проверять контакты реле Р2, РЗ и т.д. в контрольных точках К2, КЗ и т.д., то в этом случае мы сможем обнаружить неисправный контакт только через семь шагов, т.е. через семь проверок.

Таким образом, при последовательной проверке контактов время поиска неисправности будет зависеть от места расположения неисправности и от направления ее поиска. Для минимизации числа проверок применяется метод половинного деления или метод дихотомии.

Рассмотрим суть этого метода на том же самом примере, но заменив для общности рассуждений контакты реле прямоугольниками с теми же самыми номерами, что и реле, считая, что в прямоугольнике в общем случае может быть любой дискретный направленный элемент электротехнического устройства (рисунок 15.3) [1]. Сигнал в элементе может передаваться только в одном направлении: с входа элемента на его выход. Эти прямоугольники будем считать блоками, работоспособность которых требуется определить.

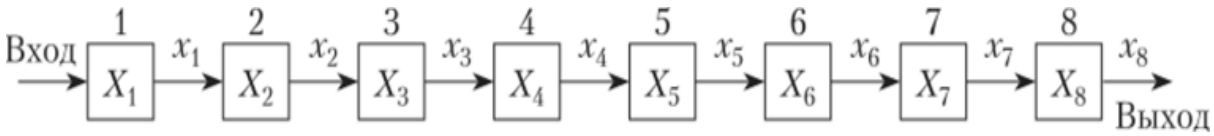


Рисунок 15.3 - Последовательная проверка контактов

Состояние блока обозначим прописной буквой X с индексом, равным номеру блока. Например, состояние блока 2 обозначим X_2. Величину выходного сигнала блока будем обозначать строчной буквой x с индексом, соответствующим номеру блока.

Каждый блок может находиться только в двух состояниях: исправном «1» и неисправном «0». Например, выражение $X_5 = 0$ означает, что пятый блок неисправен, а $x_3 = 1$ означает, что на выходе третьего блока имеется рабочий (допустимый) сигнал, который принимают за единицу. В исправном состоянии при подаче единичного сигнала на вход блока на его выходе также появляется единичный сигнал.

Для начала будем считать, что в схеме имеется только одна неисправность, т.е. только один отказавший блок. На рисунке 15.4 [1] упрощенно показан принцип выбора проверяемого элемента по методу половинного деления, в соответствии с которым вместо семи последовательных проверок всех элементов оказывается достаточным сделать только три проверки для обнаружения неисправности в схеме.

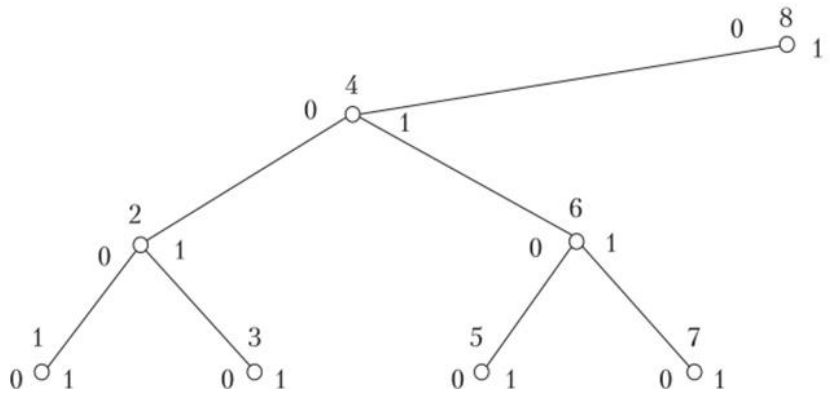


Рисунок 15.4 - Принцип выбора проверяемого элемента

Алгоритм проверки показан на рисунке 15.5 [1].

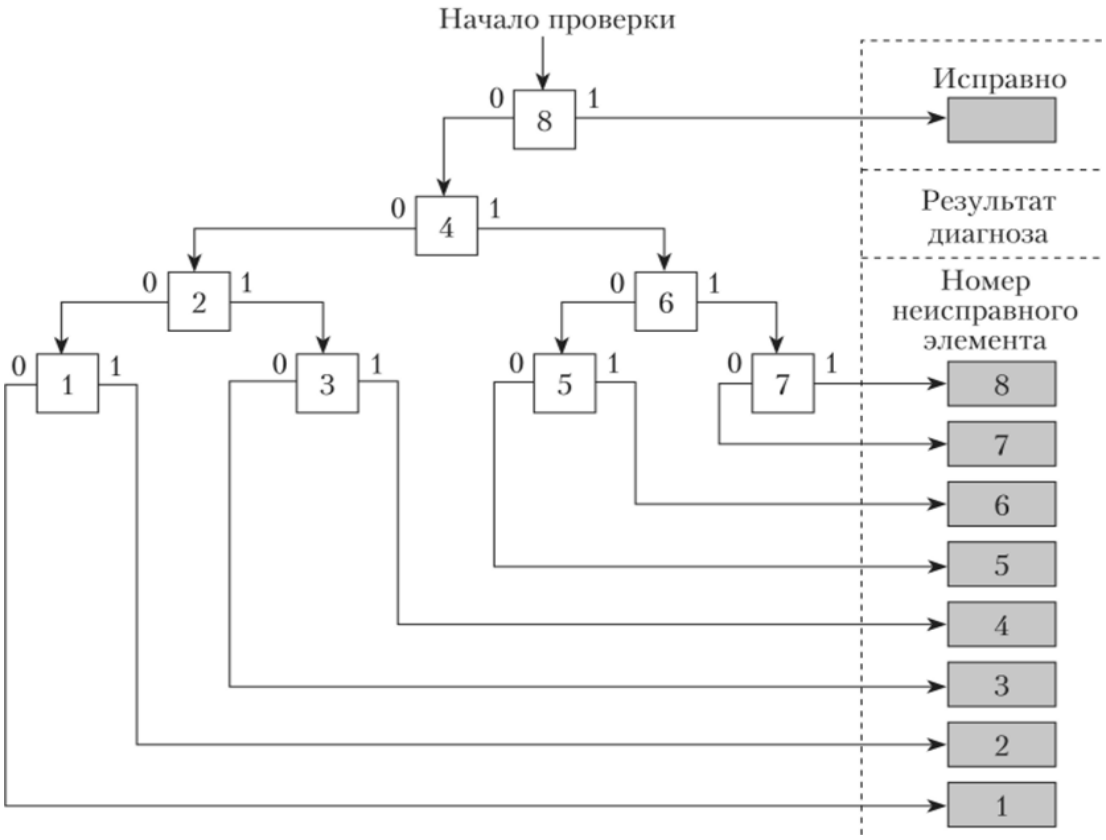


Рисунок 15.5 - Алгоритм проверки

Результаты измерений можно представить в виде таблицы неисправностей, аналогичным таблице истинности элемента «И», но они становятся громоздкими для схем, имеющих большое количество элементов. Гораздо проще представлять математическую модель в виде формул, составленных для схемы с помощью алгебры логики, и анализировать эти формулы с помощью прикладных компьютерных программ.

Мы можем записать выражения для математической модели объекта диагностирования с дискретными элементами следующим образом:

$$x1 = 1 \cdot X1 = X1; \quad x2 = x1 \cdot X2; \quad x3 = x2 \cdot X3; \quad x4 = x3 \cdot X4;$$

$$x5 = x4 \cdot X5 = X1; \quad x6 = x5 \cdot X6; \quad x7 = x6 \cdot X7; \quad x8 = x7 \cdot X8.$$

В [1] приведена составленная в системе *Mathcad* математическая модель процесса диагностирования неразветвленной электрической цепи. В ней приведен вектор состояний X элементов электрической цепи, логические уравнения выходных сигналов x этих элементов (математическая модель объекта диагноза), а также логические уравнения, описывающие процесс диагностирования.

Для составления уравнений, описывающих процесс поиска неисправных элементов электрической цепи, используют алгоритм проверки,

изображенный на рисунке 15.5. В соответствии с этим алгоритмом составляют сначала словесное описание для отыскания каждой неисправности. Например, элемент 2 будет неисправен в том случае, если нет сигнала на восьмом, четвертом и втором элементах и есть сигнал на первом элементе. Это условие в Mathcad записывается с помощью функции с условием if (если):

$$y3 := \text{if}(\neg x8 \wedge \neg x4 \wedge x2 \wedge \neg x3 = 1, \text{"NO"}$$

Результат этой функции зависит от того, выполняется или не выполняется заданное условие. В скобках этой функции сначала записывается условие, затем два выражения: if (условие, выражение 1, выражение 2). Если условие, записанное в скобках, выполняется, то функция принимает значение, равное выражению 1. Если же условие не выполняется, то функция принимает значение, равное выражению 2. Условием для каждого неисправного элемента является истинность или равенство логической 1 словесного описания неисправности данного элемента, записанное с помощью логических операторов MathCad, находящихся на панели булевой логики. Жирный знак равенства в функции if — это не обычный знак равенства, который утверждает, что выражение слева равно единице. Жирный знак равенства входит в условие, которое может выполняться и может не выполняться. Этот жирный знак равенства, или знак выполнения условия, также набирается на панели булевой логики или одновременным нажатием клавиш [Ctrl] + [=].

Аналогичная модель может быть реализована и в приложении Excell (рисунок 15.6).

	№ элемента	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	Состояние элемента X	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
	Выходные сигналы x	1	1	0	0	0	0	0	0
		ИСТИНА	ИСТИНА	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ
		ЛОЖЬ	ЛОЖЬ	ИСТИНА	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ
	Неисправный элемент	НОРМ	НОРМ	АВАРИЯ	НОРМ	НОРМ	НОРМ	НОРМ	НОРМ
	Система в целом	АВАРИЯ							

Рисунок 15.6 - Аналогичная модель реализованная в приложении Excell

Здесь запись логической формулы для того же третьего элемента выглядит:

*y3=ЕСЛИ (*И(НЕ(*x8*);НЕ(*x4*); *x2*;НЕ(*x3*)), *«АВАРИЯ 3», «НОРМ»).*

Этот вариант поиска неисправностей будет выполняться в лабораторном практикуме.

Основная литература:

1 Серебряков А. С. Автоматика: учебник и практикум для академического бакалавриата / А. С. Серебряков, Д. А. Семенов, Е. А. Чернов; под общ. ред. А. С. Серебрякова. — М. : Издательство Юрайт, 2017. - 431 с.

2 Собакин Е.Л. Цифровая схемотехника. Учеб. пособие. Ч.I. – Томск: Изд. ТПУ, 2002. - 160с.

3 Сборник задач по дискретной математике. Для практических занятий в группах: Ю. П. Шевелев, Л. А. Писаренко, М. Ю. Шевелев — Санкт-Петербург, Лань, 2013 г.- 528 с.

4 Савельев А. Я. Прикладная теория цифровых автоматов. М. : Высш. шк. , 2007. 272 с.

5 Б.Аладышкин. Статьи о булевой алгебре для электриков. http://electrik.info

Дополнительная литература

1 Горбатов В. А. Основы дискретной математики. М. : Высш. шк. , 1986.

Сводный план 2019г, поз.221

Марат Аскарович Мустафин
Алтын Нурмолдаевна Бестерекова

ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ СИСТЕМ УПРАВЛЕНИЯ

Конспект лекций
для студентов специальности 5В071800 – Электроэнергетика

Редактор Л.Т. Сластихина
Специалист по стандартизации Г.И.Мухаметсариева

Подписано в печать Формат 60х84 1/16
Тираж _50_экз. Бумага типографская №1
Объем _3,88_ уч. изд. л. Заказ № Цена 1940тенге

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи имени
Гумарбека Даукеева»
050013 Алматы,ул. Байтурсынова, 126/1