КИПиА ЯМАЛ

Слесарь по контрольно-измерительным приборам и автоматике

5-й разряд · МДК.01.03

Тема 3.1

«Логические основы цифровой техники»

Полный конспект лекций и практической подготовки

6 академических часов: 4 часа теории + 2 часа практики

Тренажёр: /logika.php

Редакция: 06.10.2026

СОДЕРЖАНИЕ

1. Занятие 1. Базовые логические операции и таблицы истинности — 2 часа
2. Занятие 2. Составные функции, упрощение и реализация — 2 часа
3. Лабораторно-практическое занятие — 2 часа
4. Контрольные вопросы и итоговая задача

Цель темы. научить переходить от физического двоичного сигнала к логической операции, таблице истинности, булевой функции и функциональной схеме, а затем проверять результат ручным экспериментом.

1. Занятие 1. Базовые логические операции и таблицы истинности — 2 часа

В цифровой технике логические 0 и 1 — это абстрактные состояния, которым в реальном устройстве соответствуют диапазоны напряжения, тока, состояния контактов или внутренние биты контроллера. Поэтому сначала нужно определить, что именно означает 0 и 1 в данной системе.

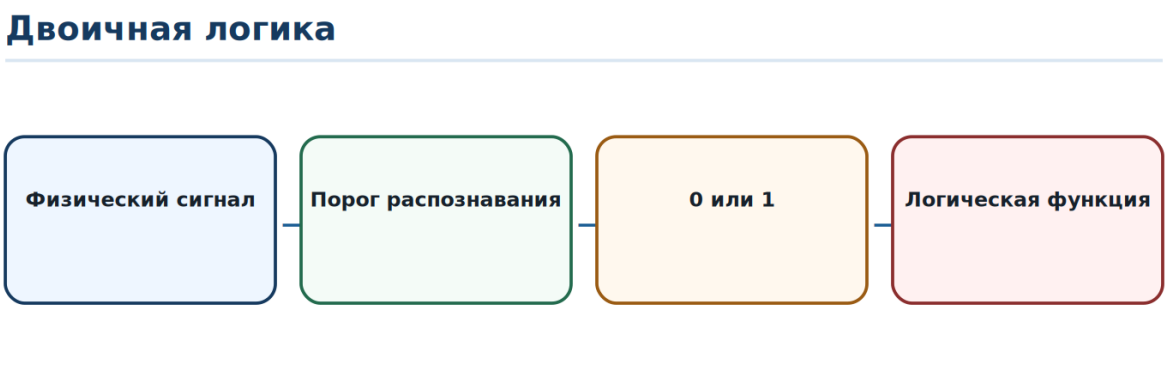


Рисунок 1 — От физического сигнала к логическому состоянию

1.1. Операция НЕ

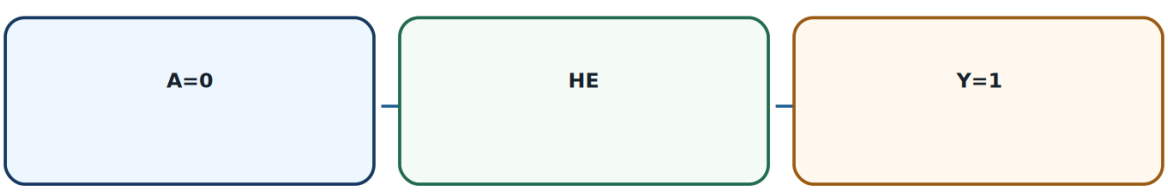


Рисунок 2 — Инвертор меняет состояние на противоположное

НЕ имеет один вход. Если A=0, то Y=1; если A=1, то Y=0. В релейной логике аналогом инверсии часто служит нормально-замкнутый контакт.

A	Y=НЕ A
0	1
1	0

1.2. Операция И

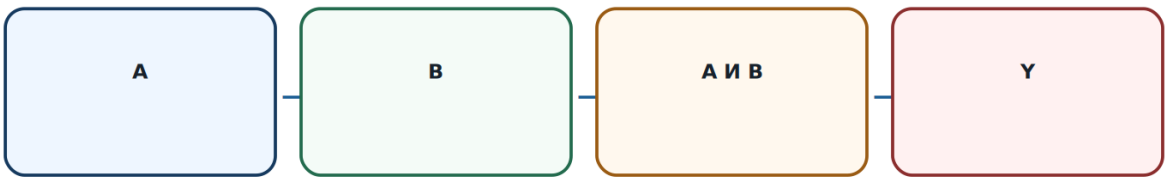


Рисунок 3 — Выход активен только при выполнении всех условий

Операция И соответствует требованию «все условия одновременно». Например, разрешение пуска может формироваться как «защитная дверь закрыта И давление в норме».

A	B	Y=A И B
0	0	0
0	1	0
1	0	0
1	1	1

1.3. Операция ИЛИ



Рисунок 4 — Достаточно хотя бы одного активного условия

ИЛИ даёт единицу, если активен хотя бы один вход. В релейной схеме параллельные нормально-открытые контакты реализуют ИЛИ.

A	B	Y=A ИЛИ B
0	0	0
0	1	1
1	0	1
1	1	1

1.4. XOR и XNOR

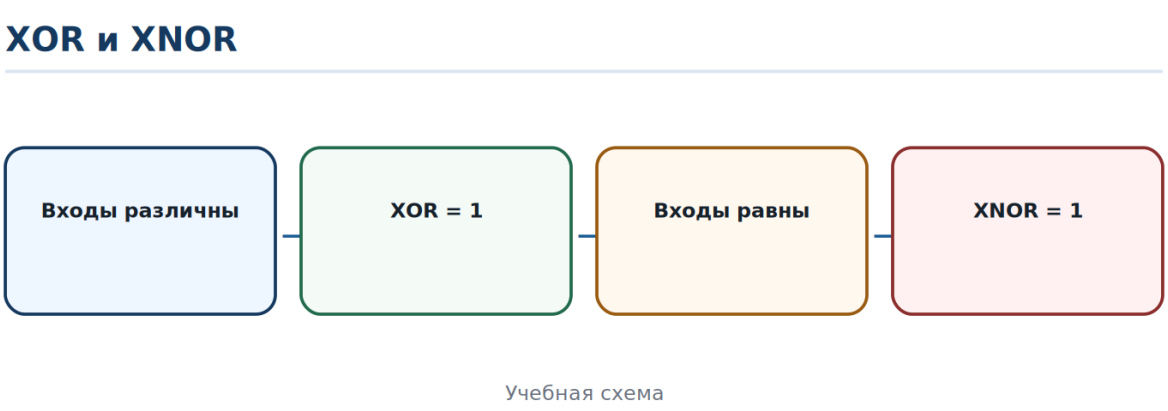


Рисунок 5 — Различие и равенство входов

XOR равен 1, когда два входа различны. XNOR — когда равны. XOR удобно использовать для признака несовпадения двух дискретных состояний.

A	B	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

1.5. NAND и NOR



Рисунок 6 — Инверсия результата И или ИЛИ

NAND — это НЕ(И), NOR — НЕ(ИЛИ). Эти элементы важны тем, что каждый из них способен образовать функционально полный базис: только на NAND или только на NOR можно построить любую булеву функцию.

1.6. Таблица истинности



Рисунок 7 — Алгоритм построения

Для n входов существует 2^n комбинаций. Для трёх входов A,B,C — восемь строк. Таблица истинности является объективной проверкой функции: если схема и формула дают одинаковый Y во всех строках, они логически эквивалентны.

1.7. Основные законы булевой алгебры

• A И 1 = A; A И 0 = 0.

• A ИЛИ 0 = A; A ИЛИ 1 = 1.

• A И A = A; A ИЛИ A = A.

• A И НЕ A = 0; A ИЛИ НЕ A = 1.

• НЕ(НЕ A) = A.

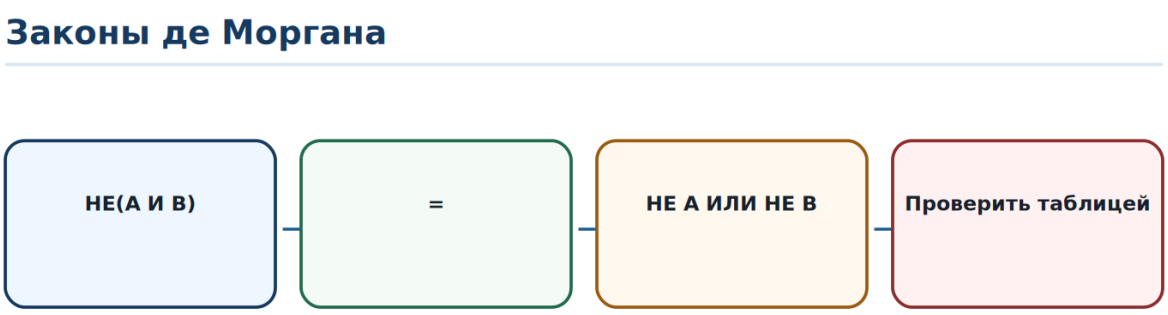


Рисунок 8 — Один из законов де Моргана

Законы де Моргана позволяют переносить инверсию через И/ИЛИ: НЕ(А И В)=НЕ А ИЛИ НЕ В; НЕ(А ИЛИ В)=НЕ А И НЕ В. Полезно не заучивать механически, а проверять таблицей истинности.

2. Занятие 2. Составные функции, упрощение и реализация — 2 часа

2.1. Промежуточные сигналы



Рисунок 9 — Чтение схемы слева направо

Составную схему удобно разбирать по узлам. Например, для Y=(А И В) ИЛИ С сначала вводят S1=А И В, затем Y=S1 ИЛИ С. Такой подход совпадает с тренажёром, где промежуточные сигналы отображаются как S1, S2.

```
Пример 1
A=1, B=0, C=1.
S1=A И B=0.
Y=S1 ИЛИ C=0 ИЛИ 1=1.
```

2.2. Переход от алгоритма к функции

Словесное условие переводят в переменные и операции. Фраза «двигатель разрешён, если есть команда ПУСК и защита исправна, либо включён сервисный режим» может быть записана как Y=(START И OK) ИЛИ SERVICE.

2.3. СДНФ и СКНФ



Рисунок 10 — Построение СДНФ по строкам Y=1

СДНФ строят по строкам таблицы, где Y=1. Для каждой строки записывают конъюнкцию всех входов: переменная берётся без инверсии, если в строке 1, и с инверсией, если 0. Затем минтермы объединяют операцией ИЛИ. СКНФ строится по строкам Y=0 через макстермы.

2.4. Минимизация и карта Карно

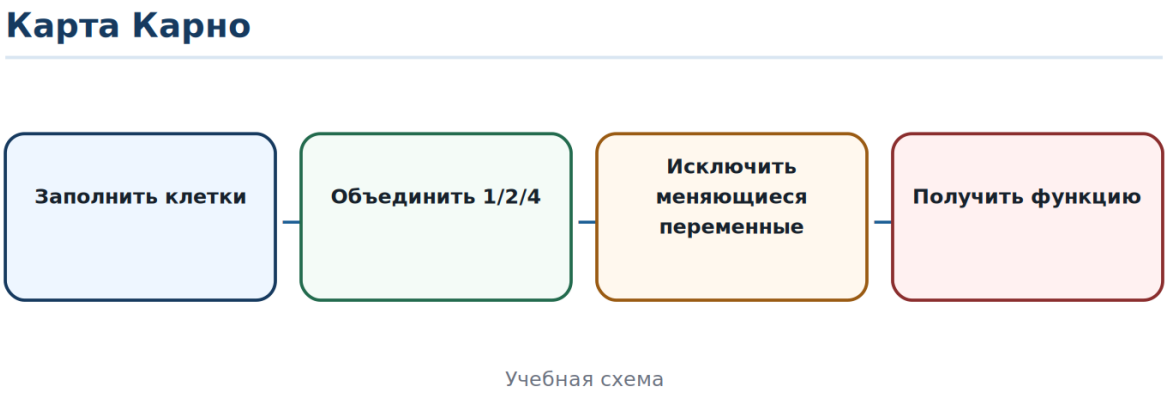


Рисунок 11 — Группировка соседних единиц

Карта Карно — графический способ упрощения функций с небольшим числом переменных. Соседние клетки отличаются одной переменной; единицы объединяют прямоугольными группами размером 1,2,4,8. Переменные, меняющиеся внутри группы, исключаются.

> Важно. минимизация нужна не ради красивой формулы, а чтобы уменьшить число элементов, соединений и потенциальных точек отказа.

2.5. Релейная реализация



Рисунок 12 — Соответствие контактов логическим операциям

Последовательные нормально-открытые контакты соответствуют И, параллельные — ИЛИ, нормально-замкнутый контакт реализует инверсию входного условия. Такая связь помогает перейти от булевой логики к LAD и промышленной автоматике.

2.6. Комбинационная схема и схема с памятью



Рисунок 13 — Ключевое различие

В комбинационной схеме выход определяется только текущими входами. В схеме с памятью результат зависит также от предыдущего состояния. Самоподхват реле, RS-триггер и внутренний бит PLC — примеры хранения состояния.

3. Лабораторно-практическое занятие — 2 часа

Практика выполняется в /logika.php. Основной режим — «Ручной эксперимент», затем «Обучение с разбором» и в конце «Контроль знаний».

1. Выбрать уровень «Логические элементы» и вручную перебрать все входные состояния НЕ, И, ИЛИ, XOR, NAND, NOR и XNOR.

2. Для каждого элемента сравнить наблюдаемый выход Y с таблицей истинности.

3. Перейти в «Составные схемы» и для одной схемы вручную вычислить S1/S2 и Y до переключения входов.

4. Перебрать все наборы A/B/C выбранной схемы и заполнить таблицу истинности.

5. Записать булеву функцию схемы.

6. Для функции с несколькими единицами получить СДНФ.

7. Выполнить простое упрощение или группировку на карте Карно.

8. Перейти в режим «Контроль знаний» и выполнить набор без подсказок.

A	B	C	S1	S2	Y

4. Контрольные вопросы и итоговая задача

1. Что физически может обозначать логическая 1?
2. Как работает НЕ?
3. Когда И даёт единицу?
4. Когда ИЛИ даёт ноль?
5. Чем XOR отличается от ИЛИ?
6. Что показывает XNOR?
7. Что такое NAND и NOR?
8. Почему NAND/NOR называют функционально полными базисами?
9. Сколько строк у таблицы истинности для трёх входов?
10. Как проверить эквивалентность двух функций?
11. Сформулируйте один закон де Моргана.
12. Что такое промежуточный сигнал S1?
13. Как строится СДНФ?
14. Для чего нужна карта Карно?
15. Как операции И/ИЛИ связаны с последовательными и параллельными контактами?
16. Чем комбинационная схема отличается от схемы с памятью?

Итоговая задача
Дано: Y=(A И НЕ B) ИЛИ C.
1) Построить таблицу истинности для 8 наборов.
2) Для A=1,B=0,C=0 вычислить НЕ B=1, затем A И НЕ B=1, значит Y=1.
3) Проверить все строки в ручном режиме /logika.php на эквивалентной составной схеме.