ЭиУА

Дәуренов ӘД

Лекция 8. Логические элементы автоматики

Цель лекции

Изучить

- назначение,
- принцип работы и
- основные виды логических элементов, применяемых в системах автоматизации, а также
- рассмотреть их использование в релейных схемах,
- программируемых логических контроллерах (PLC) и промышленных системах управления.

План лекции

1. Понятие логических элементов автоматики.
2. Двоичная логика и логические сигналы.
3. Основные логические функции: NOT, AND, OR.
4. Дополнительные логические элементы: NAND, NOR, XOR, XNOR.
5. Таблицы истинности и логические выражения.
6. Реализация логических функций с помощью реле и контакторов.
7. Логические операции в программируемых логических контроллерах.
8. Применение логических элементов в промышленных системах.
9. Значение логических элементов в современной автоматизации.

Понятие логических элементов автоматики

Логический элемент (Logic Gate) — это устройство или функциональный блок, выполняющий определённую логическую операцию над одним или несколькими входными сигналами и формирующий выходной сигнал.

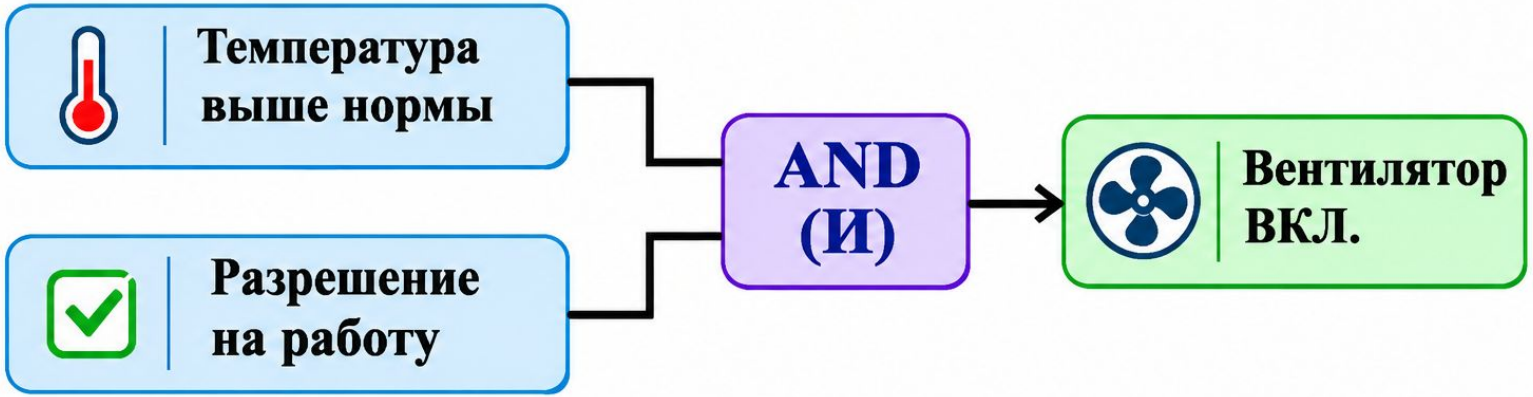
Логические элементы используются для принятия простых решений в автоматических системах.
Например, промышленный вентилятор должен включаться только при выполнении двух условий:

- Температура превышает установленное значение.
- Система охлаждения разрешена к работе.

Если оба условия выполнены, вентилятор включается. Если хотя бы одно условие не выполнено, вентилятор остаётся выключенным.

Такое управление реализуется логической операцией **AND (И)**.

Пример логического управления вентилятором



Вентилятор включается только тогда, когда оба входных условия имеют логическое значение 1.

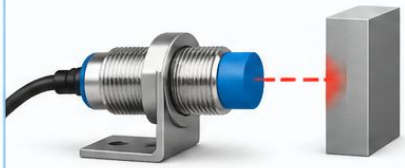
Логические элементы могут быть реализованы с помощью электромеханических реле, полупроводниковых микросхем, программируемых контроллеров (PLC) и других технических средств.

Большинство логических систем работают с двумя состояниями:

Логическое значение	Обозначение	Возможное физическое состояние
0 Логический ноль	0 / FALSE	Нет сигнала, условие не выполнено
1 Логическая единица	1 / TRUE	Есть сигнал, условие выполнено

В промышленной автоматизации логические сигналы могут представлять состояние датчика, кнопки, контакта реле, исполнительного механизма или аварийного сигнала.

Пример: цифровой датчик положения



Может выдавать:

- 0 — объект не обнаружен.
- 1 — объект обнаружен.

Пример уровней логического сигнала на 24 В DC



В системе управления на 24 В DC логическим уровням могут соответствовать определённые диапазоны напряжений, установленные технической документацией оборудования.

Важно: логическая единица не всегда означает наличие именно 24 В, а логический ноль — строго 0 В. Допустимые электрические уровни зависят от типа входа, выхода и используемого оборудования.

3. Основные логические функции

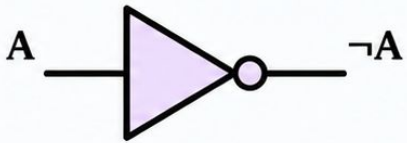
К основным логическим функциям относятся:

NOT (НЕ)

Логическое отрицание.

Инвертирует логическое значение (меняет 0 на 1 и 1 на 0).

A	¬A
0	1
1	0

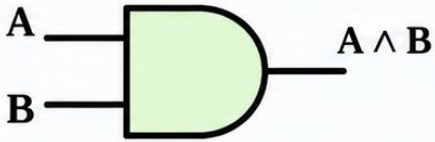


AND (И)

Логическое умножение.

Результат равен 1, только если оба входных значения равны 1.

A	B	A ∧ B
0	0	0
0	1	0
1	0	0
1	1	1

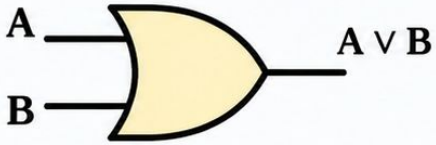


OR (ИЛИ)

Логическое сложение.

Результат равен 1, если хотя бы одно из входных значений равно 1.

A	B	A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1



Пример применения: контроль состояния двери промышленного оборудования

Условие задачи

Предположим, на защитной двери станка установлен датчик, который выдаёт:

- **1** — дверь закрыта;
- **0** — дверь открыта.

Необходимо включить сигнальную лампу, когда дверь открыта.

Логическое решение

Для этого используется логический элемент **NOT (НЕ)**.

$$Y = \overline{A}$$

где:

- **A** — сигнал датчика закрытия двери;
- **Y** — команда включения сигнальной лампы.

Схема логического управления

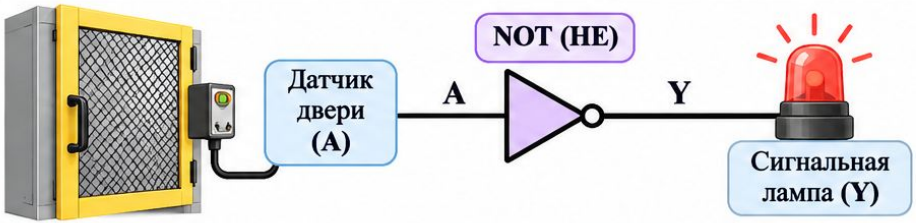
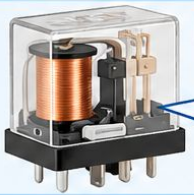


Таблица истинности

A (дверь закрыта)	Y (сигнальная лампа)
0 (открыта)	**1** (включена)
1 (закрыта)	**0** (выключена)

Реализация в релейной схеме



NC (нормально замкнутый контакт)



В релейной схеме такую функцию можно реализовать с помощью нормально замкнутого (NC) контакта реле, катушка которого управляется сигналом датчика.

Таким образом, логический элемент NOT позволяет автоматически включать предупреждающий сигнал при открытии двери.

Примечание: это пример логической сигнализации, а не полноценной системы защитной блокировки станка.

Пример применения логического элемента AND (И): управление насосом

Пример: водяной насос должен включаться только в том случае, если уровень воды в резервуаре низкий **И** в источнике подачи достаточно воды.

Исходные условия

- **A = 1** — уровень воды в резервуаре низкий.
- **B = 1** — в источнике подачи достаточно воды.

Логическое выражение

$$Pump = A \wedge B$$

Схема управления насосом

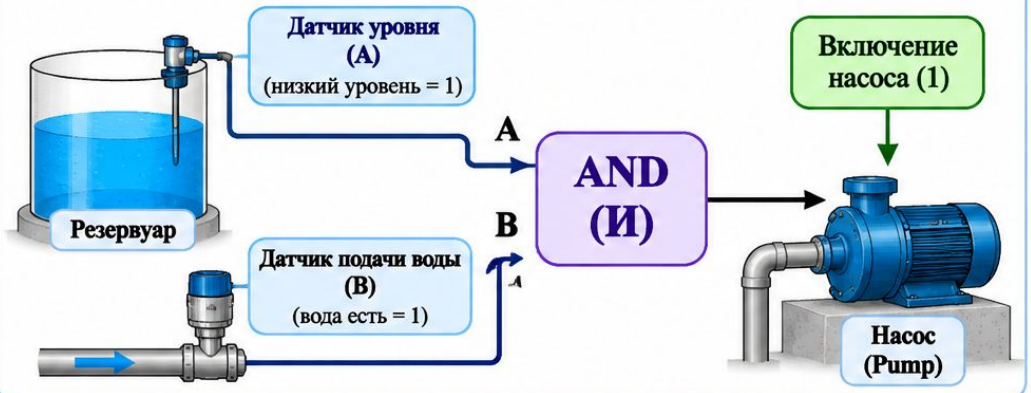


Таблица истинности

A (уровень низкий)	B (вода есть)	Насос (Y) (Pump)
0	0	0 — Выключен
0	1	0 — Выключен
1	0	0 — Выключен
1	1	**1 — Включён**

Принцип работы: насос включается только при одновременном выполнении двух условий: необходимо пополнить резервуар и имеется достаточно воды для перекачивания.

Если хотя бы одно условие равно нулю, команда включения насоса не формируется.

Пример: сигнал аварии должен появляться, если превышена температура двигателя или обнаружено недопустимо низкое давление масла.



Превышение температуры
(HighTemperature, A)

Низкое давление масла
(LowOilPressure, B)

OR (ИЛИ)



Аварийный сигнал
(Alarm, Y)

Логическое выражение

$$Alarm = HighTemperature \vee LowOilPressure$$

Таблица истинности

A (высокая T)	B (низкое P)	Y = A ∨ B (авария)
0	0	0
0	1	1
1	0	1
1	1	1

Аварийный сигнал включится при выполнении хотя бы одного условия.

Дополнительные логические функции

Кроме основных функций NOT, AND и OR, в автоматике применяются NAND, NOR, XOR и XNOR.

NAND (И-НЕ)

Выполняет операцию AND с последующим отрицанием результата.

$$Y = \overline{A \cdot B}$$



A	B	Y (NAND)
0	0	1
0	1	1
1	0	1
1	1	0

Выход равен нулю только тогда, когда оба входа равны единице.

NOR (ИЛИ-НЕ)

Выполняет операцию OR с последующим отрицанием результата.

$$Y = \overline{A \vee B}$$



A	B	Y (NOR)
0	0	1
0	1	0
1	0	0
1	1	0

Выход равен единице только тогда, когда оба входа равны нулю.

XOR (Исключающее ИЛИ)

Формирует единицу, если два входных сигнала различаются.

$$Y = A \oplus B$$



A	B	Y (XOR)
0	0	0
0	1	1
1	0	1
1	1	0

Например, XOR можно использовать для обнаружения несовпадения состояний двух цифровых сигналов.

XNOR (Эквивалентность)

Формирует единицу, когда оба входных сигнала одинаковы.

$$Y = \overline{A \oplus B}$$



A	B	Y (XNOR)
0	0	1
0	1	0
1	0	0
1	1	1

Например, XNOR может применяться для проверки совпадения двух логических состояний.

Таблица истинности (Truth Table)

Таблица истинности (Truth Table) — это таблица, показывающая значения выходного сигнала для всех возможных комбинаций входных сигналов.

Если логическая функция имеет n двоичных входов, количество возможных комбинаций составляет:

$$N = 2^n$$

Пример: количество комбинаций

Количество входов (n)	Количество комбинаций ($N = 2^n$)
1	2
2	4
3	8
4	16
5	32

Пример системы с тремя входными условиями

Рассмотрим систему с тремя входными условиями:

- **A** — команда запуска.
- **B** — разрешение на работу.
- **C** — сигнал блокировки.

Выходное выражение:

$$Y = A \wedge B \wedge \overline{C}$$

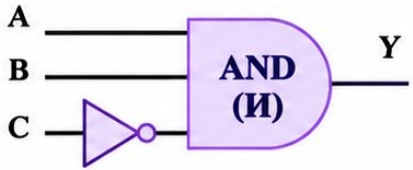


Таблица истинности для примера

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Принцип работы: система включится только тогда, когда активна команда запуска ($A = 1$), получено разрешение ($B = 1$) и отсутствует блокировка ($C = 0$).

Такие таблицы используются при проектировании алгоритмов управления, проверке программ PLC и анализе работы автоматических устройств.

Реализация логических функций с помощью реле и контакторов

Логические функции AND (И), OR (ИЛИ) и NOT (НЕ) можно реализовать не только с помощью электронных микросхем и программируемых контроллеров (PLC), но и с помощью обычных электромагнитных реле и контакторов.

Этот способ широко применяется в релейно-контактных схемах управления промышленным оборудованием.

https://www.youtube.com/watch?v=1yE8X9CvolI&list=PLtpvR7-pDSPGRVVpfBS8eW2g6WQRiZd66&index=1
https://www.youtube.com/watch?v=Tl1rnivmqVE&list=PLtpvR7-pDSPGRVVpfBS8eW2g6WQRiZd66&index=3
https://www.youtube.com/watch?v=4htZ3e-gaUo&list=PLtpvR7-pDSPGRVVpfBS8eW2g6WQRiZd66&index=4

Как реле выполняет логические операции?



Электромагнитное реле состоит из **катушки** и **контактов.**

При подаче напряжения на катушку →  → **реле срабатывает и изменяет положение своих контактов.**

Основные типы контактов

NO
(Normally Open)

Нормально открытый контакт.
При несработавшем реле разомкнут, при срабатывании замыкается.

NC
(Normally Closed)

Нормально закрытый контакт.
При несработавшем реле замкнут, при срабатывании размыкается.

Состояния контактов реле

NO-контакт

Разомкнут



Катушка выключена — цепь разомкнута.

NC-контакт

Замкнут



Катушка включена — цепь замкнута.

Именно с помощью таких контактов можно реализовать основные логические операции.

Реализация функции AND (И) с помощью релейных контактов

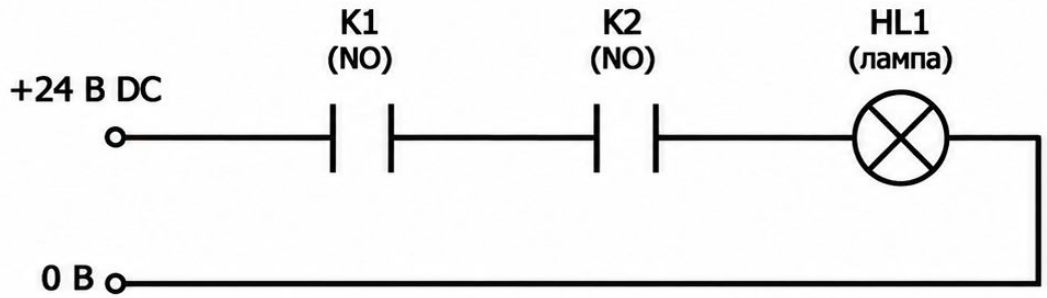
Для реализации функции AND два нормально открытых контакта реле соединяют **последовательно**.

$$Y = K1 \wedge K2$$

Пример: управление сигнальной лампой с помощью двух реле.

Условие: лампа должна гореть только тогда, когда одновременно сработают реле K1 и K2.

Принципиальная электрическая схема (точное соединение)



Обозначения и элементы

	K1, K2 — нормально открытые контакты реле (NO)
	HL1 — сигнальная лампа (нагрузка)
+24 В DC, 0 В	Источник питания постоянного тока (например, 24 В DC)

Логическое выражение:

$$HL1 = K1 \wedge K2$$

K1	K2	Лампа HL1
0	0	Выключена
0	1	Выключена
1	0	Выключена
1	1	Включена

Принцип работы:

1. Если K1 и K2 не сработали, цепь разомкнута, лампа не горит.
2. Если сработало только K1, цепь остаётся разомкнутой.
3. Если сработало только K2, цепь также остаётся разомкнутой.
4. Если сработали оба реле, цепь замыкается и лампа загорается.

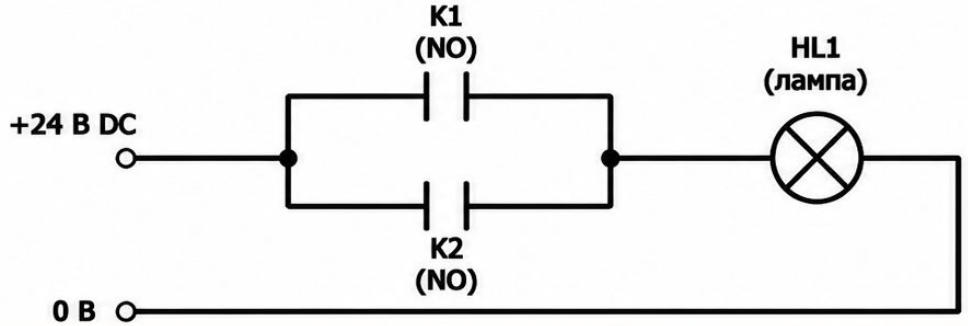
Реализация функции OR (ИЛИ) с помощью релейных контактов

Для реализации функции OR два нормально открытых контакта реле соединяют **параллельно**.

$$Y = K1 \vee K2$$

Пример: сигнальная лампа должна включаться при срабатывании любого из двух реле.

Принципиальная электрическая схема (точное соединение)



Обозначения и элементы

	K1, K2 — нормально открытые контакты реле (NO)
	HL1 — сигнальная лампа (нагрузка)
+24 В DC, 0 В	Источник питания постоянного тока (например, 24 В DC)

Логическое выражение:

$$HL1 = K1 \vee K2$$

K1	K2	Лампа HL1
0	0	Выключена
0	1	**Включена**
1	0	**Включена**
1	1	**Включена**

Принцип работы:

1. При срабатывании K1 ток проходит через верхнюю ветвь цепи, лампа включается.
2. При срабатывании K2 ток проходит через нижнюю ветвь цепи, лампа включается.
3. При срабатывании обоих реле лампа также включается.

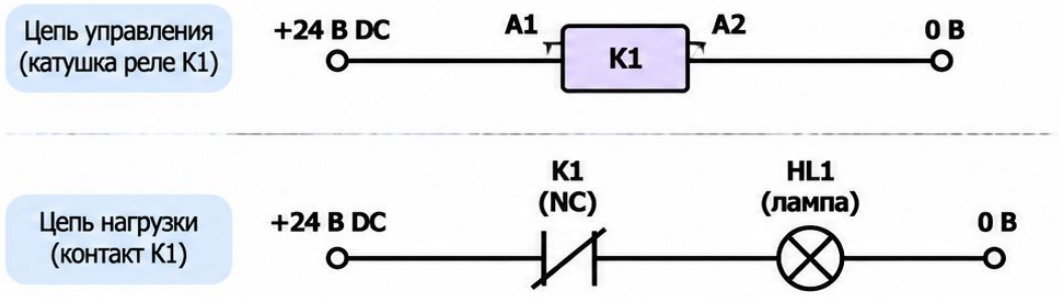
Реализация функции NOT (НЕ) с помощью релейного контакта

Для реализации функции NOT можно использовать **нормально закрытый** контакт реле.

$$Y = \overline{K1}$$

Пример: сигнальная лампа должна гореть, когда реле K1 не сработало, и выключаться, когда реле сработало.

Принципиальная электрическая схема (точное соединение)



Обозначения и элементы

A1 — K1 — A2	K1 — катушка электромагнитного реле (управляется сигналом A)
	K1 (NC) нормально закрытый контакт реле (замкнут при A = 0)
	HL1 — сигнальная лампа (нагрузка)
+24 В DC, 0 В	Источник питания постоянного тока (например, 24 В DC)

Логическое выражение:

$$HL1 = \overline{A}$$

A (катушка K1)	Контакт K1 (NC)	Лампа HL1
0	Замкнут	**Включена**
1	Разомкнут	**Выключена**

Принцип работы:

1. Если на катушку K1 не подано напряжение (A = 0), её NC-контакт замкнут, лампа включена.
2. Если на катушку K1 подано напряжение (A = 1), её NC-контакт разомкнут, лампа выключена.

Как контактор используется в логических схемах автоматики?

Контактор работает по электромагнитному принципу, похожему на реле, но предназначен прежде всего для коммутации силовых электрических цепей, например питания электродвигателей.

У контактора могут быть:

- Главные силовые контакты — для подключения нагрузки.
- Вспомогательные NO- и NC-контакты — для построения логики управления, сигнализации и блокировок.
- Электромагнитная катушка — для управления положением контактов.

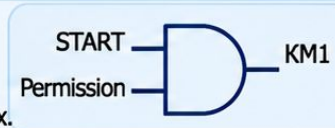
Управление контактором: реализация логической функции AND (И)

Условие: двигатель должен включаться, когда нажата кнопка **START** и получено разрешение на работу (Permission).

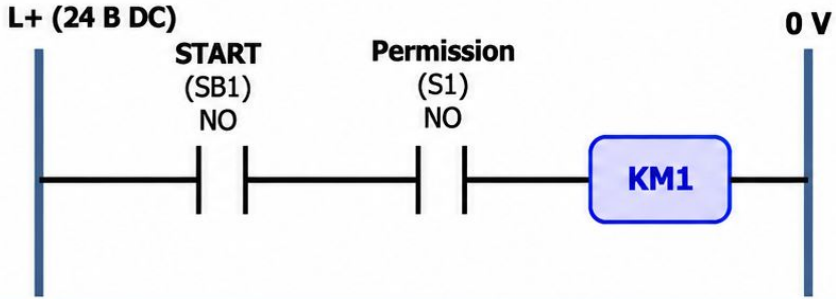
Логическое выражение:

$$KM1 = START \wedge Permission$$

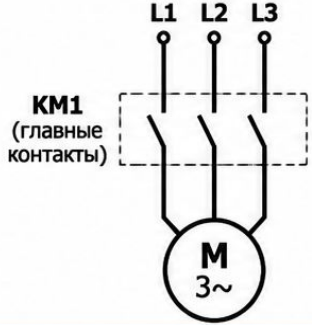
Принцип: два нормально открытых контакта соединены последовательно (логическое И). Катушка контактора KM1 получает питание только при замыкании обоих.



Принципиальная схема управления (цепь катушки KM1)



Силовая схема (подключение двигателя)



Обозначения и элементы



NO — нормально открытый контакт (замыкается при нажатии или при срабатывании устройства)



KM1 — катушка контактора (управляющий элемент)



Главные контакты контактора (коммутируют силовую цепь двигателя)



M — электродвигатель (трёхфазный)

Таблица истинности

START	Permission	KM1 (катушка)	Двигатель
0	0	0	Выключен
0	1	0	Выключен
1	0	0	Выключен
1	1	**1**	**Включен**

Принцип работы схемы

1. **Пуск**: при нажатии START и наличии разрешения (Permission) замыкаются оба контакта, катушка KM1 получает питание, контактор срабатывает.
2. **Остановка**: при отпускании START или при снятии разрешения катушка обесточивается, контактор возвращается в исходное состояние.

Вывод

Схема реализует логическую функцию AND (И):

$$KM1 = START \wedge Permission$$

Контактор включается только при одновременном выполнении двух условий.

Логические операции в программируемых логических контроллерах

Современные системы промышленной автоматизации часто используют ***программируемые логические контроллеры*** — PLC.

Контроллер получает сигналы от датчиков, кнопок, переключателей и других устройств, обрабатывает их по заданной программе и формирует команды для исполнительных механизмов.

Логическое управление на основе ПЛК (PLC)

Суть: ПЛК получает дискретные сигналы от датчиков и кнопок, выполняет логические функции (AND, OR, NOT, XOR) в программе и управляет исполнительными устройствами.

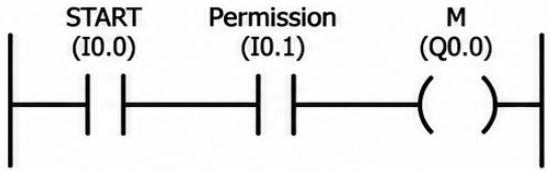


Пример: управление двигателем

Условие: двигатель включается, если нажата кнопка START и получено разрешение на работу.

$$M = START \wedge Permission$$

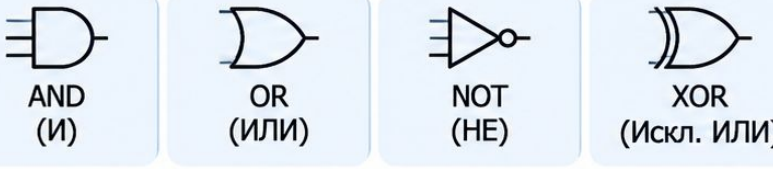
Программа в ПЛК (Ladder, LD)



Назначение сигналов (пример)

Адрес ПЛК	Сигнал	Тип
I0.0	Кнопка START	Вход (дискретный)
I0.1	Разрешение (Permission)	Вход (дискретный)
Q0.0	Двигатель (M)	Выход (дискретный)

Поддерживаемые логические функции



Преимущества реализации на ПЛК:

- Гибкая настройка логики в программе.
- Простое изменение условий работы без переделки электросхемы.
- Расширение функций (таймеры, счётчики, блокировки и др.).
- Соответствие стандарту IEC 61131-3 (языки LD, FBD, ST, SFC, IL).

Языки программирования ПЛК для реализации логики управления

Задача: включить двигатель, если есть команда запуска, разрешение на работу и нет неисправности.

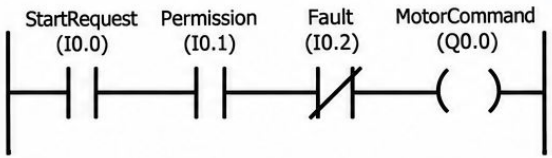
Логическое выражение:

$$MotorCommand = StartRequest \land Permission \land \neg Fault$$

Ladder Diagram (LD)

Язык релейно-контактных схем. Логика представляется в виде контактов, катушек и соединений, напоминающих электрические релейные схемы.

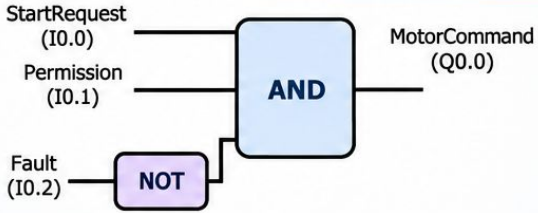
Пример на LD (лестничная диаграмма)



Function Block Diagram (FBD)

Язык функциональных блоков. Логические операции представлены отдельными блоками (AND, OR, NOT и др.).

Пример на FBD (диаграмма функциональных блоков)



Structured Text (ST)

Текстовый язык программирования. Позволяет описывать логические условия с помощью операторов (AND, OR, NOT и др.).

Пример на ST (структурированный текст)

```
MotorCommand := StartRequest AND
                Permission AND
                NOT Fault;
```

Обозначения сигналов (пример)

Сигнал	Адрес ПЛК	Назначение
StartRequest	**I0.0**	Команда запуска (кнопка START)
Permission	**I0.1**	Разрешение на работу (например, от системы безопасности)
Fault	**I0.2**	Сигнал неисправности (1 — есть неисправность)
MotorCommand	**Q0.0**	Команда включения двигателя (выход на контактор)

Ключевые выводы

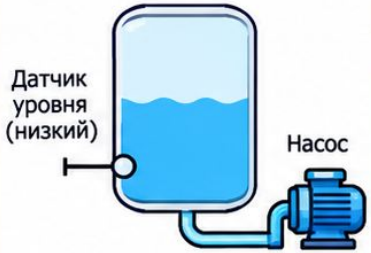
- Одну и ту же логику можно реализовать на разных языках: LD, FBD или ST.
- Во всех случаях выполняется одна и та же функция:

$$MotorCommand = StartRequest \land Permission \land \neg Fault$$

- Выбор языка зависит от требований проекта, опыта разработчика и стандартов предприятия (IEC 61131-3).
- Программная логика ПЛК не заменяет обязательные аппаратные средства защиты и функции безопасности.

Применение логических элементов в промышленной автоматизации

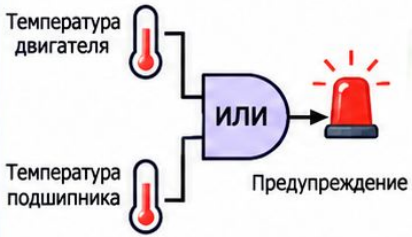
Пример 1. Автоматическое управление насосом



Логика: И (AND)

Насос включается, если уровень воды низкий и нет аварии.

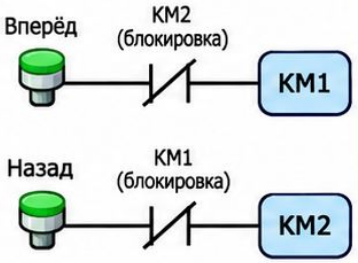
Пример 2. Контроль температуры оборудования



Логика: ИЛИ (OR)

Предупреждение включается, если температура двигателя или подшипника превышает допустимую.

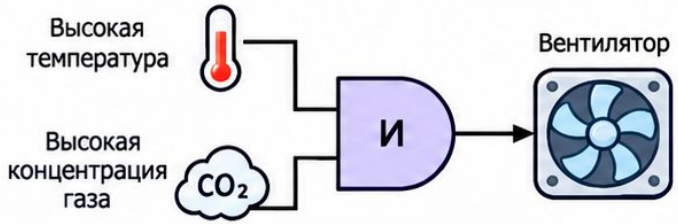
Пример 3. Блокировка двух контакторов



Логика: И (AND)

Каждый контактор включается, только если нажата своя кнопка и второй контактор отключён.

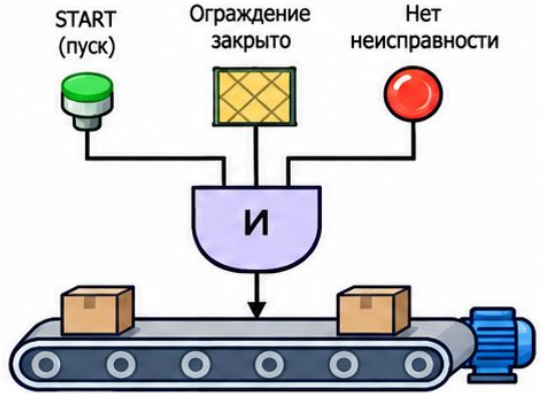
Пример 4. Система вентиляции



Логика: И (AND)

Вентилятор включается при одновременном превышении температуры и концентрации газа и наличии разрешения на работу.

Пример 5. Конвейерная система



Логика: И (AND)

Конвейер запускается, если:

- нажата кнопка START;
- защитное ограждение закрыто;
- нет сигнала неисправности.

В сложных системах управления одна логическая операция часто не может описать все необходимые условия.

Поэтому несколько логических функций объединяются в общую схему.

Комбинирование логических элементов: пример автоматического охлаждения

Задача

Реализовать автоматическое охлаждение оборудования с учётом нескольких условий.

Условия

- **A** — температура выше нормы.
- **B** — оператор включил ручное охлаждение.
- **C** — система исправна (нет неисправностей).

Логика

Вентилятор включается, если температура высокая ИЛИ есть ручная команда, и при этом система исправна.

Структура логики

(A ИЛИ B) И C

Логическая схема (комбинация элементов)

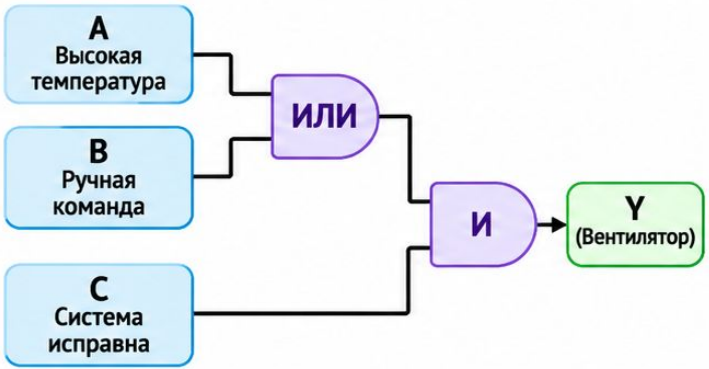
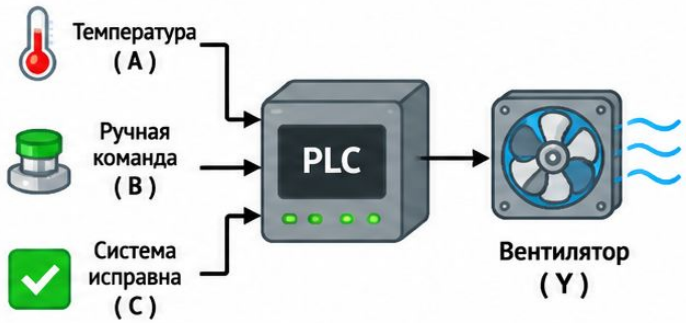


Таблица истинности

A	B	C	Y (Вентилятор)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Пример применения: охлаждение оборудования



Принцип работы

1. Сначала выполняется операция ИЛИ: активен сигнал A или B.
2. Затем результат объединяется с условием C операцией И.

В итоге вентилятор включается, если (A или B) и система исправна (C).

Итоги лекции

Логические элементы используются для обработки дискретных сигналов и формирования управляющих команд.
Основными логическими операциями являются NOT, AND и OR. Дополнительные функции NAND, NOR, XOR и XNOR позволяют реализовывать другие логические зависимости.

Логические функции могут выполняться с помощью реле, электронных схем и программируемых логических контроллеров.
Знание логических элементов необходимо для понимания релейных схем, разработки программ PLC, построения блокировок и проектирования автоматизированных систем управления.

Контрольные вопросы

Что такое логический элемент автоматики?
Какие логические состояния используются в двоичной логике?
Как работает логический элемент NOT?
В каком случае элемент AND формирует единицу на выходе?
Чем логический элемент OR отличается от AND?
Что такое таблица истинности?
Сколько комбинаций имеет логическая функция с тремя входами?
Как работает элемент NAND?
В каком случае XOR формирует логическую единицу?
Для чего используется XNOR?
Как реализовать функцию AND с помощью релейных контактов?
Как реализовать функцию OR с помощью релейных контактов?
Какую функцию может выполнять нормально закрытый контакт?
Как логические элементы используются в PLC?
Чем отличаются языки LD и FBD?
Как применяется логика в системе управления насосом?
Для чего необходимы логические блокировки?
Как реализовать включение вентилятора по двум условиям?
Почему программная логика не заменяет аппаратную защиту?
Какое значение имеют логические элементы в промышленной автоматизации?