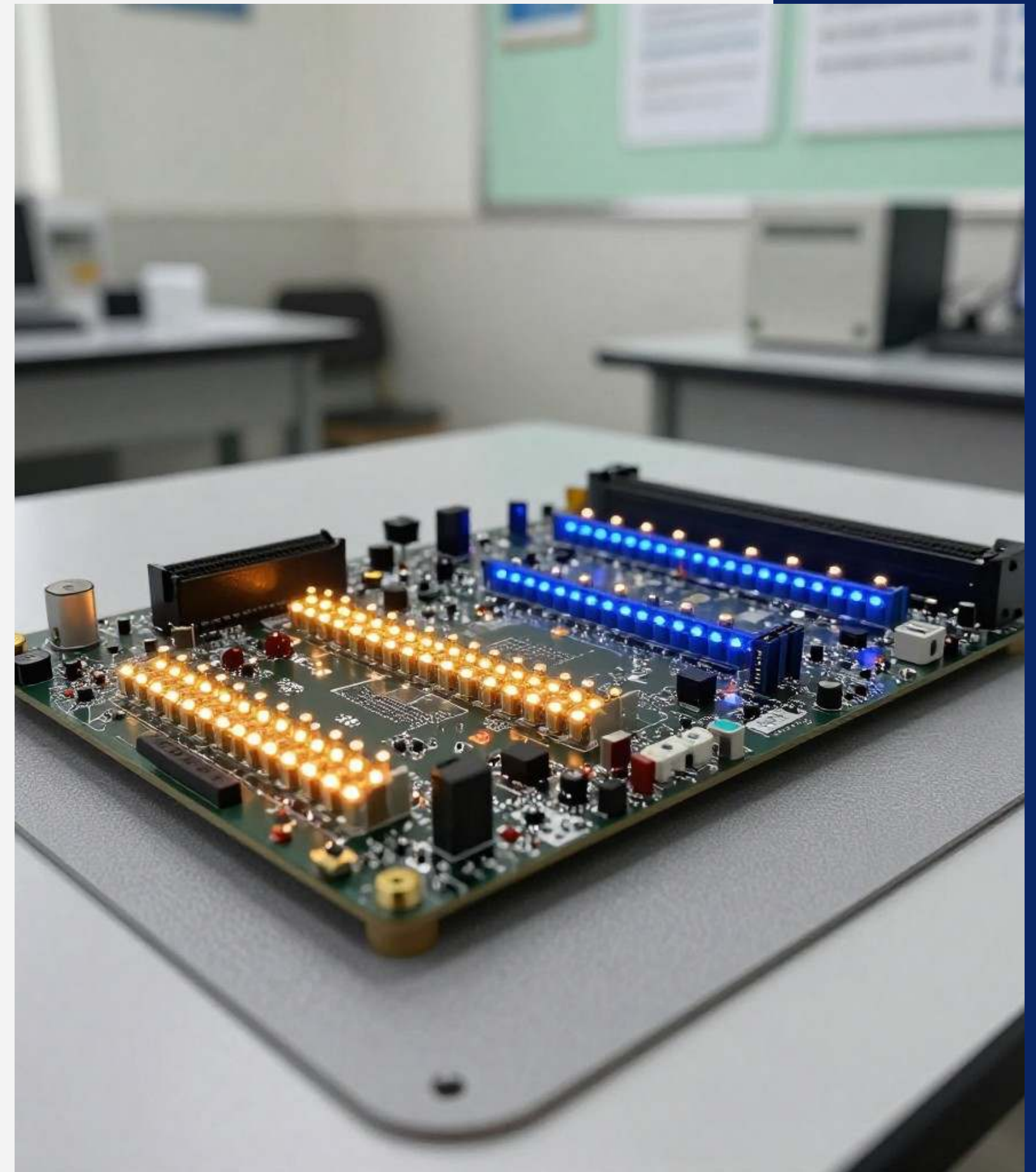


Логические основы устройства компьютера

Компьютер работает как цифровая система:
он считает, хранит и решает задачи с
логикой 0 и 1.



Почему компьютеру нужна логика



Электронные схемы надежнее всего работают с двумя устойчивыми состояниями: есть сигнал и нет сигнала. Двоичный код легко представить электрическими уровнями. Логические операции нужны для арифметики, сравнений, условий и управления.

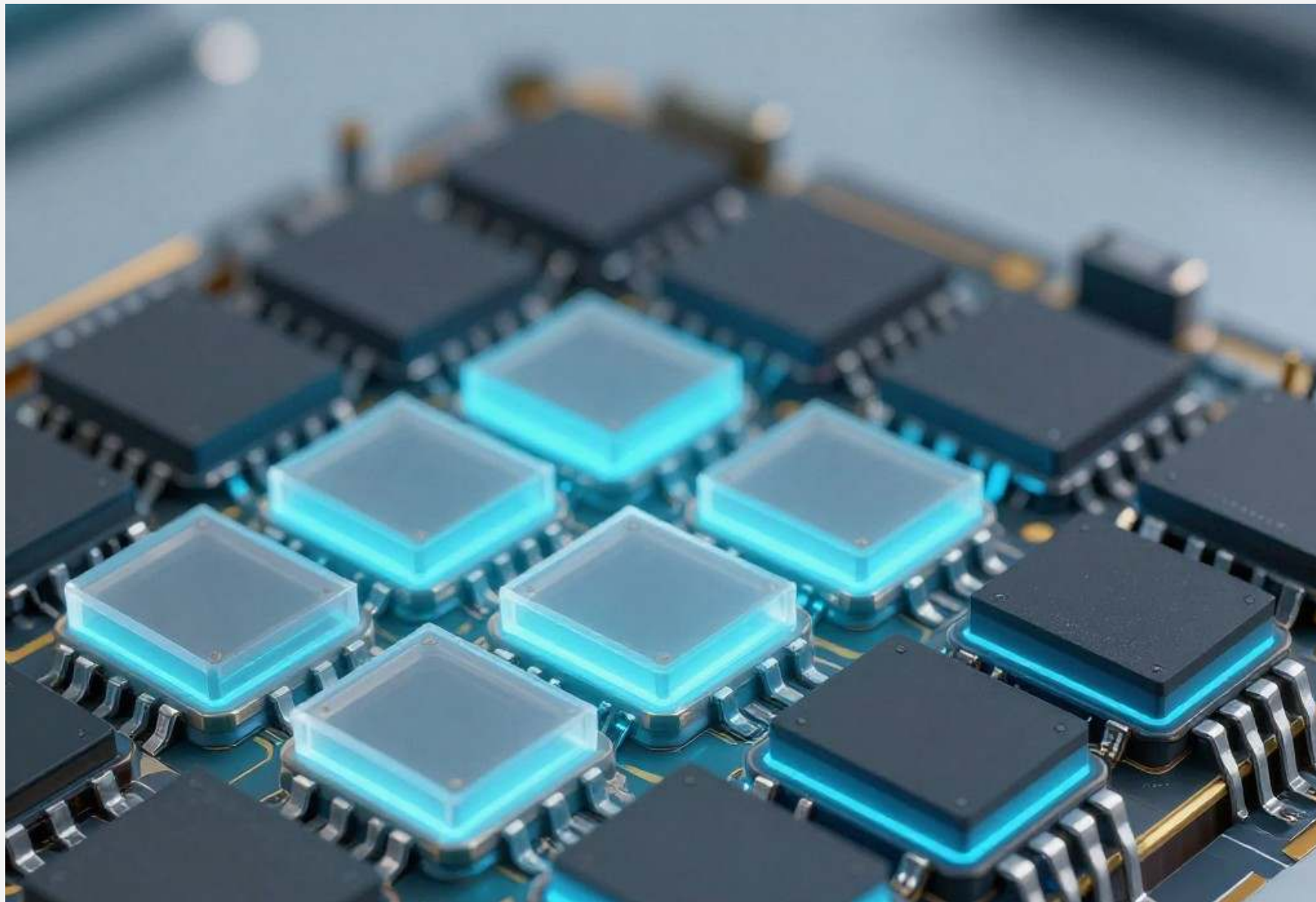
Двоичная система счисления

Бит — минимальная единица информации, принимающая значения 0 или 1. Эти числа образуют «кирпичики», из которых строится всё, что видит и обрабатывает компьютер. Для удобства наборов используют последовательности битов.

8 битов образуют 1 байт. Байты помогают упорядочивать данные при хранении и передаче: чем стабильнее и понятнее правила, тем проще микросхемам выполнять обработку. Поэтому байты широко встречаются в устройствах.

В компьютере двоичными последовательностями кодируются числа, буквы, изображения, звук и команды программы. Электронным схемам удобно различать только два состояния, поэтому двоичная запись особенно подходит для цифровых устройств.

Транзистор как логический переключатель



Пороговое напряжение и значения 1 и 0

Транзистор в микросхеме работает как очень быстрый электронный переключатель. Когда напряжение выше порогового, схема соответствует значению 1, а при более низком — значению 0. Так формируются логические состояния.

Почему в процессорах так много транзисторов

В современных процессорах и устройствах памяти используются миллиарды транзисторов. Такое количество нужно, чтобы одновременно выполнять много операций и хранить много битов. Именно поэтому двоичная логика становится реальной физической технологией.

Как сигнал проходит через компьютер

Путь данных: от ввода к обработке, временной памяти и выводу.



Логические значения и высказывания

В логике высказывание может быть истинным или ложным. В компьютере этим значениям соответствуют 1 и 0. Поэтому каждая проверка превращается в задачу: «верно» или «неверно»?

Из простых условий строятся более сложные проверки в программах. Например, сравнение двух чисел, выбор одного из двух вариантов или проверка нескольких условий подряд. Так формируются понятные правила работы приложения.

Логические условия применяются и для управления повторениями в циклах. Программа может продолжать действия, пока условие истинно, и прекращать, когда оно становится ложным. Это позволяет автоматически находить ответ и реагировать на ситуацию.

Основные логические операции

A,B	И	ИЛИ	НЕ(A) и ИЛИ(а)
00	0	0	НЕ=1, П=0
01	0	1	НЕ=1, П=1
10	0	1	НЕ=0, П=1
11	1	1	НЕ=0, П=0

И, ИЛИ, НЕ и исключающее ИЛИ задают правила, как из входов получается результат 0 или 1. Таблица показывает результат для каждой комбинации входов.

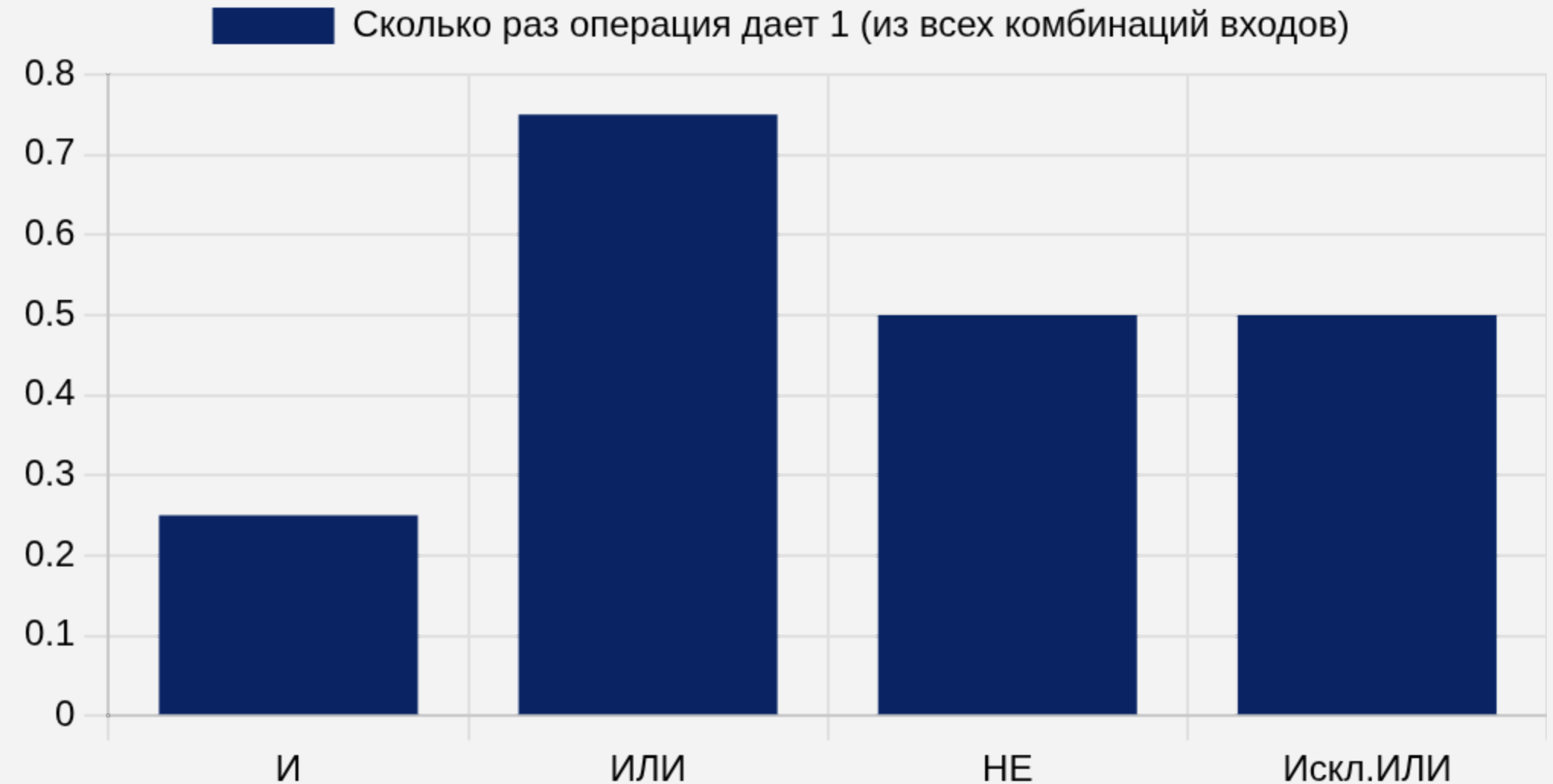
По этим правилам можно собрать любую цифровую схему: комбинируя простые операции, получают сложные проверки и преобразования сигналов.

Базовые правила булевой алгебры

Сравнение логических операций по числу единиц

Чем больше случаев дает 1, тем «чаще» операция пропускает единичный сигнал. Поэтому ИЛИ чаще активирует выход, чем И.

Разные операции по-разному реагируют на входы. Это объясняет, почему одни из них подходят для фильтрации условий, а другие — для проверки различий.



Счёт единиц по таблицам истинности для двух входов (НЕ рассматривается для одного входа).

Логические элементы в схемах



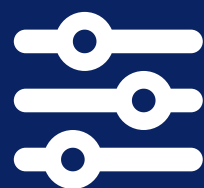
Элементы с входами и выходами

Логические элементы принимают входные сигналы и формируют выход по правилам операций И, ИЛИ, НЕ. На схемах они изображаются условными значками. Это позволяет быстро понять, что делает узел.



Соединение проводниками

Элементы связывают проводниками: выход одного узла становится входом другого. Так строятся цепочки преобразований, где каждое звено меняет логические состояния по своей функции. От этого зависит результат всей схемы.



Проверки и преобразование сигналов

Комбинируя логические элементы, можно выполнять проверки условий и преобразовывать данные. Например, определять, выполняется ли требование, или выбирать вариант действия. То есть узлы «переводят» логику в управление сигналами.



Строительные блоки цифровых устройств

На практике из таких узлов собирают более крупные модули: регистры, сумматоры, контроллеры. Поэтому логические элементы можно считать базовыми кирпичиками цифровой техники. Без них не работает никакая компьютерная логика.

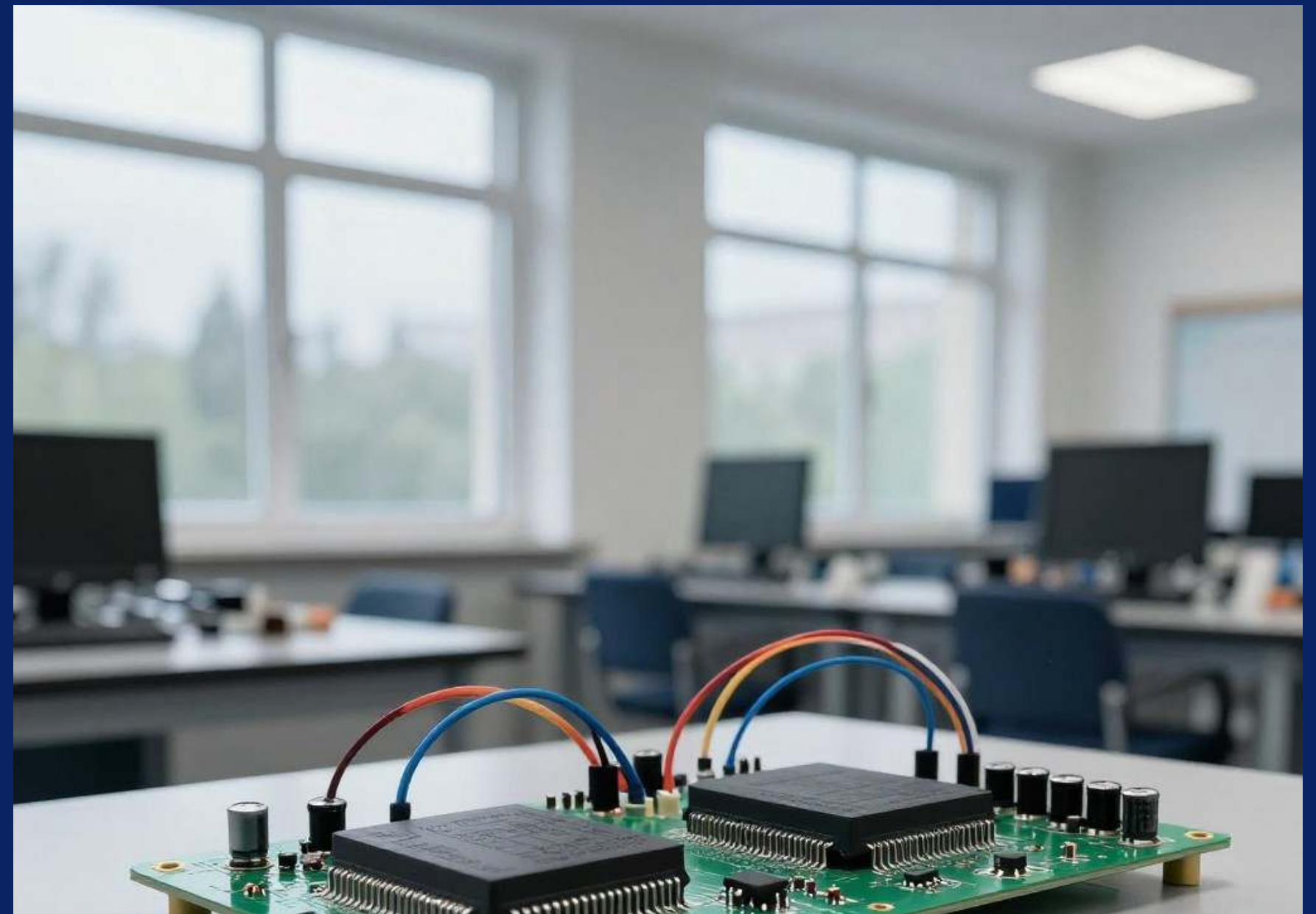
Сумматор: логика в арифметике

Полусумматор и перенос

Сложение двоичных чисел выполняют сумматоры — специальные логические схемы. Полусумматор получает два входа и выдает сумму и перенос. Он использует операции логики для вычислений на каждом разряде.

Полный сумматор в процессоре

Полный сумматор отличается тем, что учитывает входящий перенос из предыдущего разряда. Поэтому он работает в цепочке разрядов, как в настоящем сложении чисел. Эти схемы входят в состав арифметико-логического устройства процессора.



Как компьютер принимает решение

Условие в программе проверяет значение переменной или результат сравнения. Логический блок выбирает один из двух путей: «да» или «нет».



Память и логические состояния

Оперативная память хранит данные в виде устойчивых состояний ячеек: внутри каждой ячейки фиксируется бит. Такое состояние сохраняется, пока компьютер включён, позволяя программе продолжать работу.

Регистры и кэш-память работают по тому же принципу «есть состояние—нет состояния», но размещены ближе к процессору. Поэтому доступ к ним происходит быстрее и уменьшается задержка при обработке.

Одна ячейка отвечает за минимальную единицу информации, а несколько ячеек объединяются в наборы данных и команд. Благодаря этому компьютер быстро читает нужные значения и записывает результаты вычислений.

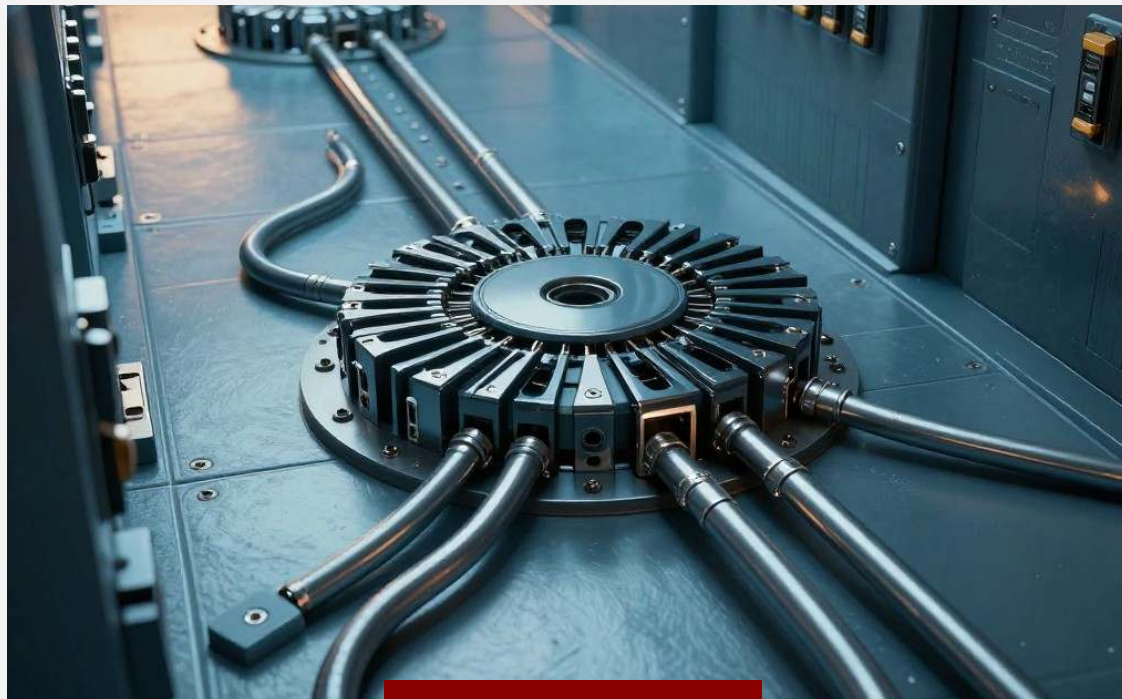
Память участвует не только в хранении, но и в вычислениях: процессор многократно выбирает биты из памяти, меняет их по логическим правилам и возвращает обратно.

Логика в процессоре



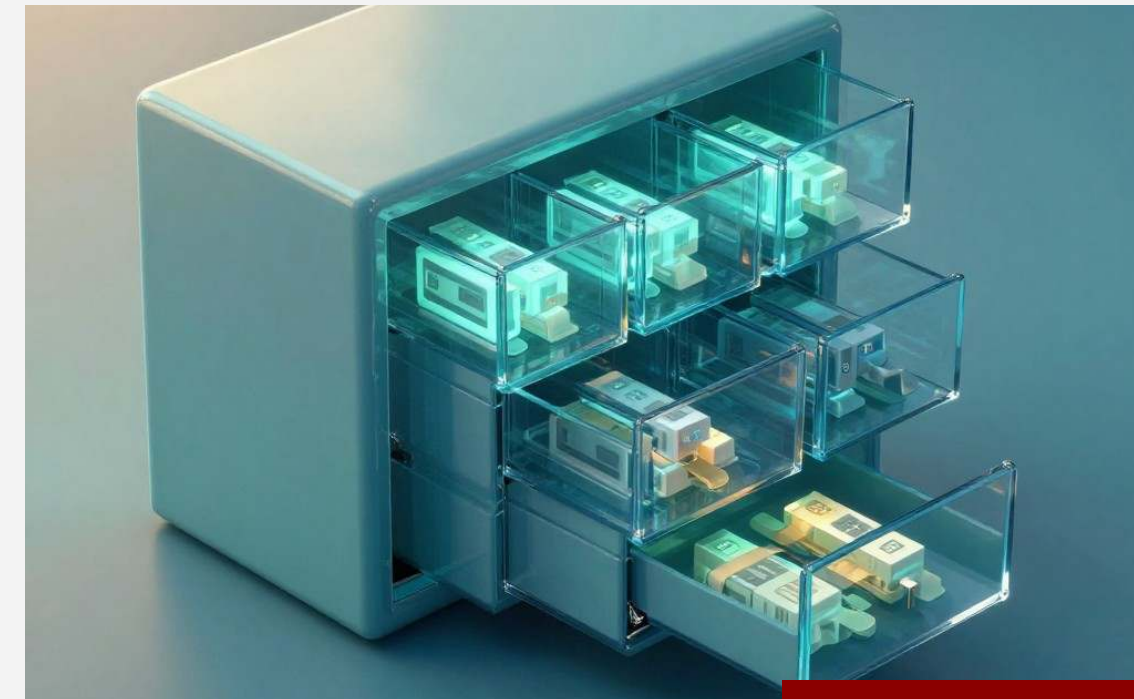
АЛУ: «считает» и сравнивает

Арифметико-логическое устройство выполняет операции над битами: сложение, вычитание, сравнения и логические преобразования. Важная идея: процессор работает с числами как с цепочками двоичных значений, поэтому логика пронизывает арифметику.



Устройство управления: задаёт порядок действий

Устройство управления организует работу процессора: выбирает следующую операцию и распределяет сигналы между узлами. Оно «связывает» команды программы с нужными логическими операциями. Так появляется последовательность шагов, необходимая любой программе.



Регистры: быстрые «карманы» для данных

Регистры временно хранят промежуточные результаты и текущие команды, чтобы процессору не приходилось каждый раз обращаться к основной памяти. Чем ближе данные к вычислителю, тем меньше задержка. В итоге логическая работа выполняется быстрее и стабильнее.

Как кодируются разные виды информации

Рассмотрим несколько примеров: буквы, числа, цвета, звук и изображения. Во всех случаях данные представляются как набор битов и кодов, чтобы компьютер мог их хранить и обрабатывать.

Любая информация представляется одинаковым базовым способом: как биты. Разные форматы лишь по-разному «раскладывают» смысл по этим битам, чтобы потом правильно восстановить текст, цвет, звук или картинку.

Вид информации	Пример	К чему сводится
Текст	Буква «А» в ASCII	Код 65
Числа	Число 5 в двоичном виде	Последовательность бит
Цвет	Пиксель в RGB	Три канала (R, G, B)
Звук	Цифровые отсчёты	Последовательность значений
Изображение	Пиксельная матрица	Набор пикселей

Стандартные способы компьютерного кодирования

От бита к байту и дальше

Чем больше единица измерения, тем сложнее хранить и передавать данные: растёт объём битов, и требуется больше памяти и пропускной способности.

Лестница размеров показывает, как из маленьких единиц собираются большие данные. Понимание масштабов помогает выбирать носители и оценивать требования к хранению.



Упрощённая шкала размеров данных (1 бит → 8 бит → 1024 байта → 1 МБ → 1 ГБ)

Логика в программах



Ветка if/иначе

Оператор if проверяет условие и выбирает, что делать дальше. Это похоже на переключатель: при истинности выполняется один набор действий, при ложности — другой, без путаницы.



Сравнения и условия

При помощи сравнений программа решает, равны ли значения, больше ли одно другому или прошло ли ограничение. Такие проверки помогают выбрать корректный путь обработки данных и действий.



Логические переменные

Переменные могут хранить результат проверки — «истина» или «ложь». Затем их удобно использовать в выражениях, не повторяя сравнения снова и снова в одном месте программы.

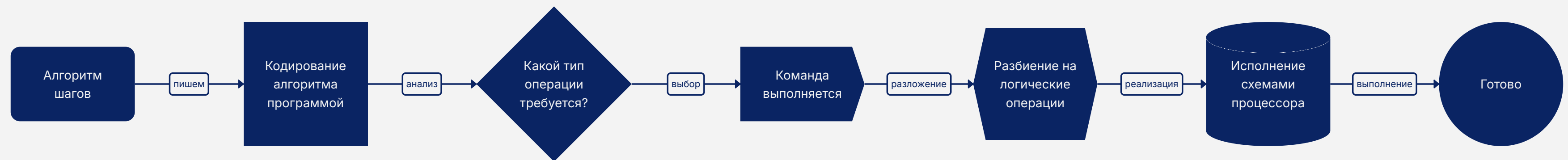


Связки И, ИЛИ, НЕ

Комбинации И, ИЛИ и НЕ позволяют строить сложные условия из простых. Например, можно требовать одновременно несколько правил или наоборот принимать один из нескольких вариантов.

От алгоритма к устройству

Алгоритм переводится в команды программы, команды распадаются на логические операции, а операции реализуются схемами внутри процессора.



Почему двоичная логика надежна

Двоичная система устойчива к помехам, потому что схема различает только два устойчивых состояния. Ей не нужно «угадывать» промежуточные уровни напряжения, что делает распознавание проще.

Если напряжение немного колеблется, достаточно чтобы оно оставалось по разные стороны порога. Тогда компьютер всё равно трактует значения как 0 и 1 и не превращает шум в ошибки данных.

Такая устойчивость важна при длительной работе, хранении и передаче информации. Чем проще различать два состояния, тем стабильнее работа устройств и тем меньше риск искажений.

Где в компьютере работает логика

Узел	Функция
Процессор	Вычисления и сравнения
Оперативная память	Хранит биты
Накопители SSD/жёсткий диск	Сохраняют коды данных
Клавиатура и мышь	Подают сигналы ввода
Монитор	Выводит пиксели изображения

Устройство цифровых систем

Ни один «главный» узел не делает всё сам. Логические состояния присутствуют во всей системе: вычисления, хранение, ввод и вывод опираются на биты и правила обработки.

Логика распределена по компонентам: каждый узел имеет собственную роль, но везде работает один принцип — управление потоками двоичных сигналов.

Компьютер — логическая машина

Компьютер работает на двоичных сигналах и логических операциях, которые реализуются схемами из транзисторов. Логика связывает данные, команды, память и процессор, поэтому устройство считает, сравнивает, хранит и принимает решения.