САТБАЕВ УНИВЕРСИТЕТІ

SATBAYEV UNIVERSITY

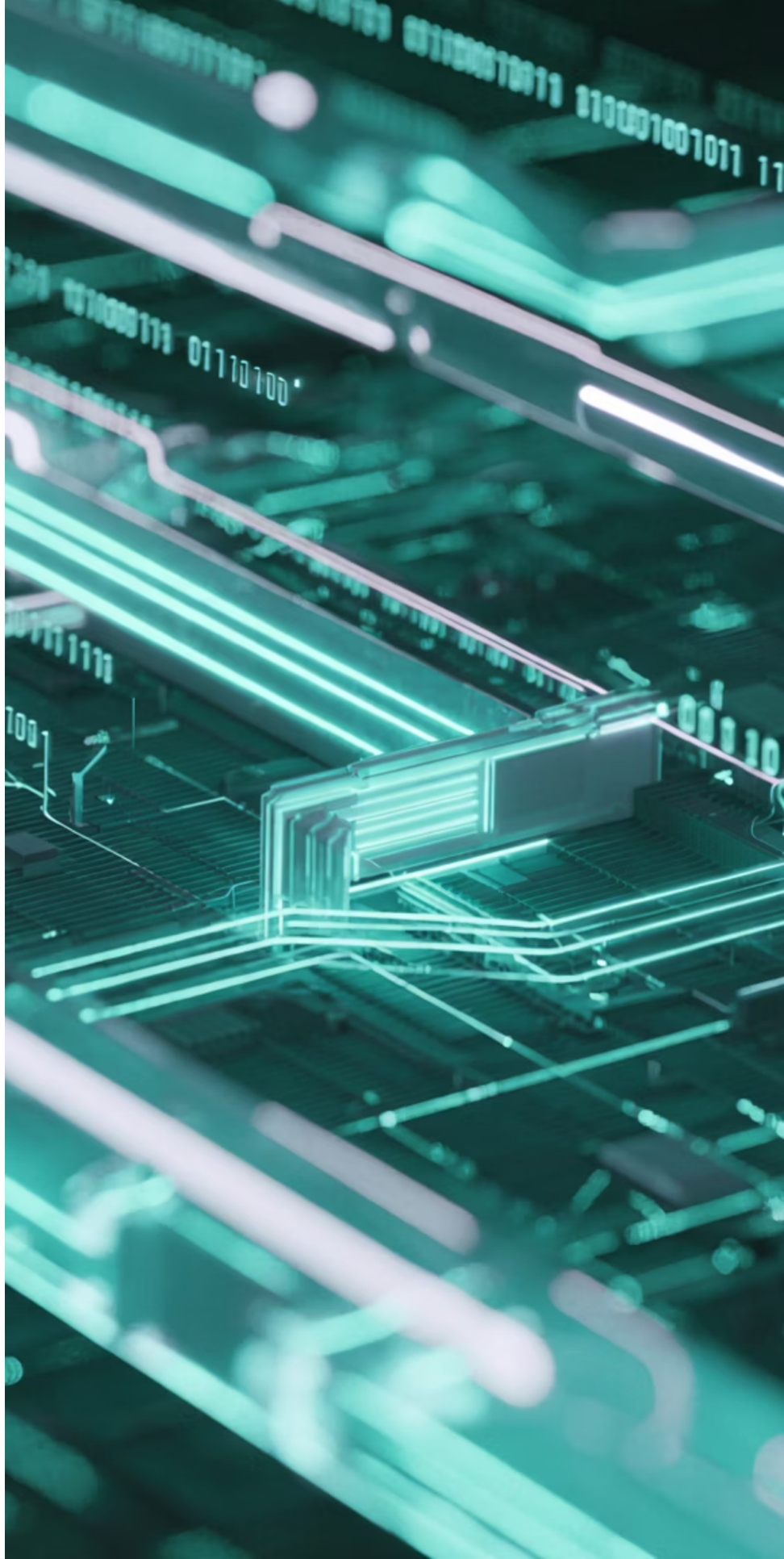
Цифровая схемотехника

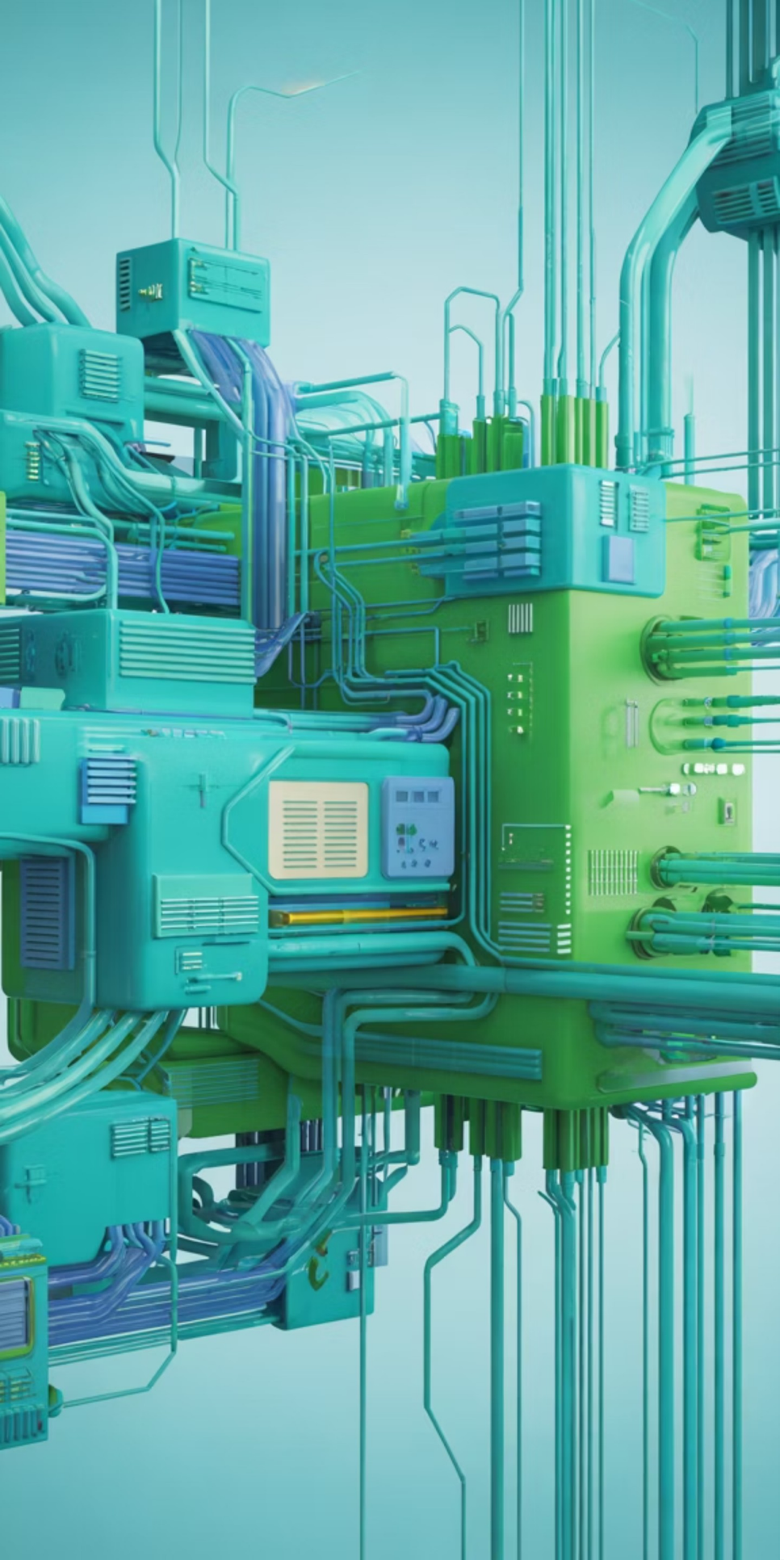
Ержан Асел Ануаркызы,Ph.D., ассоциированный профессор

Assel.yerzhan@satbayev.university

Математическое описание цифровых устройств. Системы счисления. Операции булевой алгебры

Фундаментальные принципы работы современной вычислительной техники





ЧАСТЬ 1

Введение в цифровые устройства и системы счисления

Цифровые устройства окружают нас повсюду — от смартфонов до суперкомпьютеров. Все они работают на основе одних и тех же математических принципов, которые мы рассмотрим в этой части презентации. Понимание систем счисления — это первый шаг к пониманию того, как устройства обрабатывают информацию на самом низком уровне.

Что такое цифровые устройства?

Цифровые устройства — это электронные системы, которые обрабатывают информацию в виде дискретных сигналов, представленных двумя состояниями: 0 и 1. Эта бинарная природа делает их надёжными, устойчивыми к помехам и простыми в проектировании.



Компьютеры

Обработка данных и вычисления

Смартфоны

Коммуникации и приложения

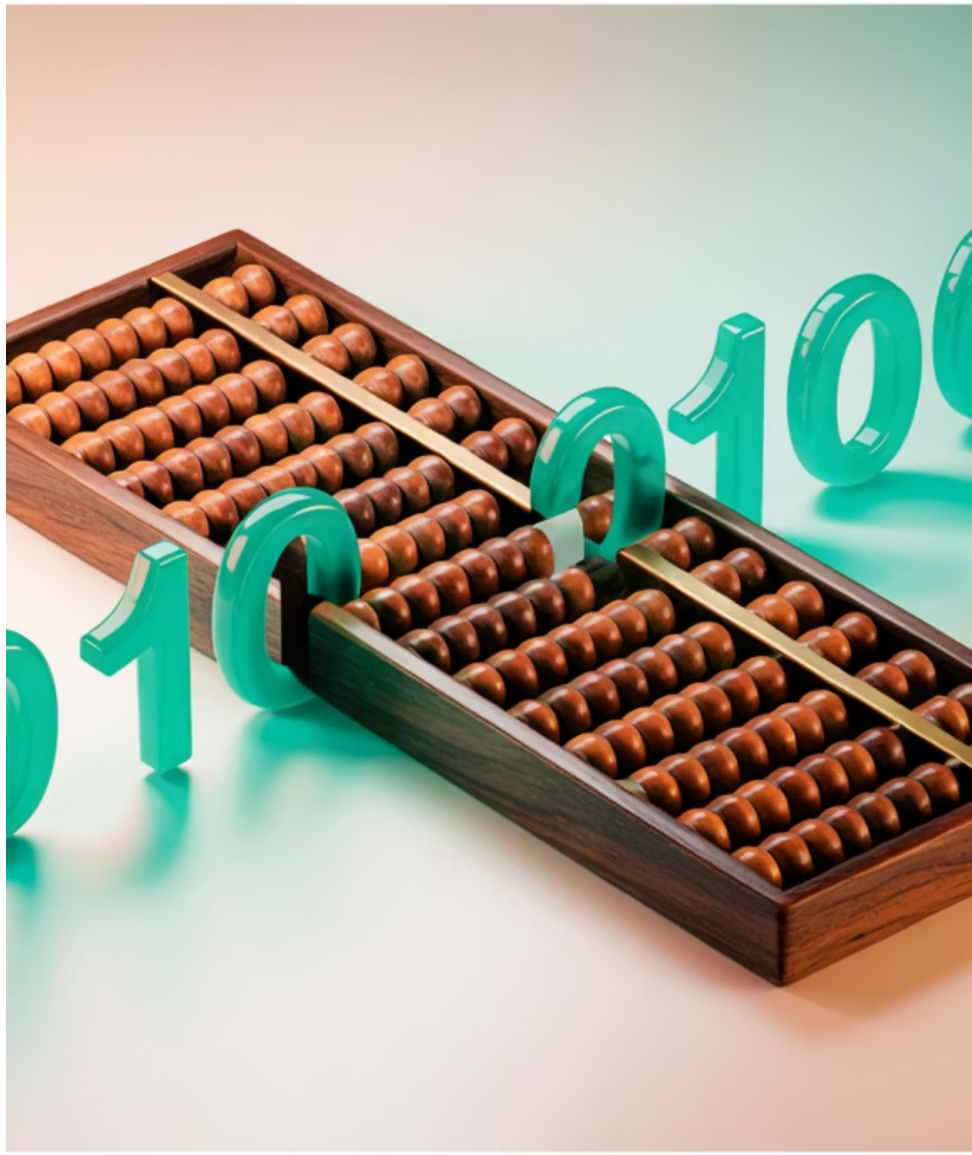
Калькуляторы

Математические операции

Логические элементы являются строительными блоками всех цифровых схем, позволяя реализовывать сложную логику через комбинации простых операций.

Системы счисления: зачем нужны?

Система счисления — это способ записи чисел с помощью определённого набора цифр и правил их позиционного размещения. Выбор системы счисления влияет на эффективность вычислений и удобство представления информации в различных контекстах.



1

Позиционные системы

Значение цифры зависит от её позиции в числе. Каждая позиция имеет свой вес, равный степени основания системы.

2

Непозиционные системы

Значение цифры не зависит от позиции (например, римские цифры). Менее удобны для вычислений.

3

Основание системы

Определяет количество уникальных цифр: десятичная (0-9), двоичная (0,1), троичная (0,1,2).

Сравнение систем счисления

Различные системы счисления используются в разных областях. Десятичная система привычна человеку, двоичная оптимальна для электроники, а другие системы применяются в специализированных приложениях.



Десятичная система

Основание 10. Используется в повседневной жизни. Каждая позиция — степень десяти: 10^3, 10^2, 10^1, 10^0.



Двоичная система

Основание 2. Идеальна для электронных схем. Каждая позиция — степень двойки: 2^3, 2^2, 2^1, 2^0.



Троичная система

Основание 3. Теоретически более эффективна. Используется в некоторых специальных компьютерах.

Двоичная система счисления

Основа цифровой электроники

Двоичная система использует только две цифры — 0 и 1, что идеально соответствует двум физическим состояниям электронных компонентов: напряжение есть или нет, ток течёт или не течёт.

Каждый разряд в двоичном числе называется **бит** (binary digit) — минимальная единица информации в цифровых системах.

Пример преобразования

Десятичное число **9** в двоичной системе:

$$9_{10} = 1 \quad 2^3 + 0 \quad 2^2 + 0 \quad 2^1 + 1 \quad 2^0 = 1001_2$$

Правила сложения

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 10 (0 с переносом 1)

Таблица соответствия десятичных и двоичных чисел

Понимание связи между десятичными и двоичными числами критически важно для работы с цифровыми системами. Таблица ниже демонстрирует закономерности преобразования.

Десятичное	Двоичное
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Десятичное	Двоичное
8	1000
10	1010
15	1111
16	10000
20	10100
100	1100100
255	11111111

Обратите внимание: количество разрядов в двоичном числе растёт логарифмически относительно десятичного значения. Для представления числа N требуется $\lceil \log_2(N+1) \rceil$ битов.

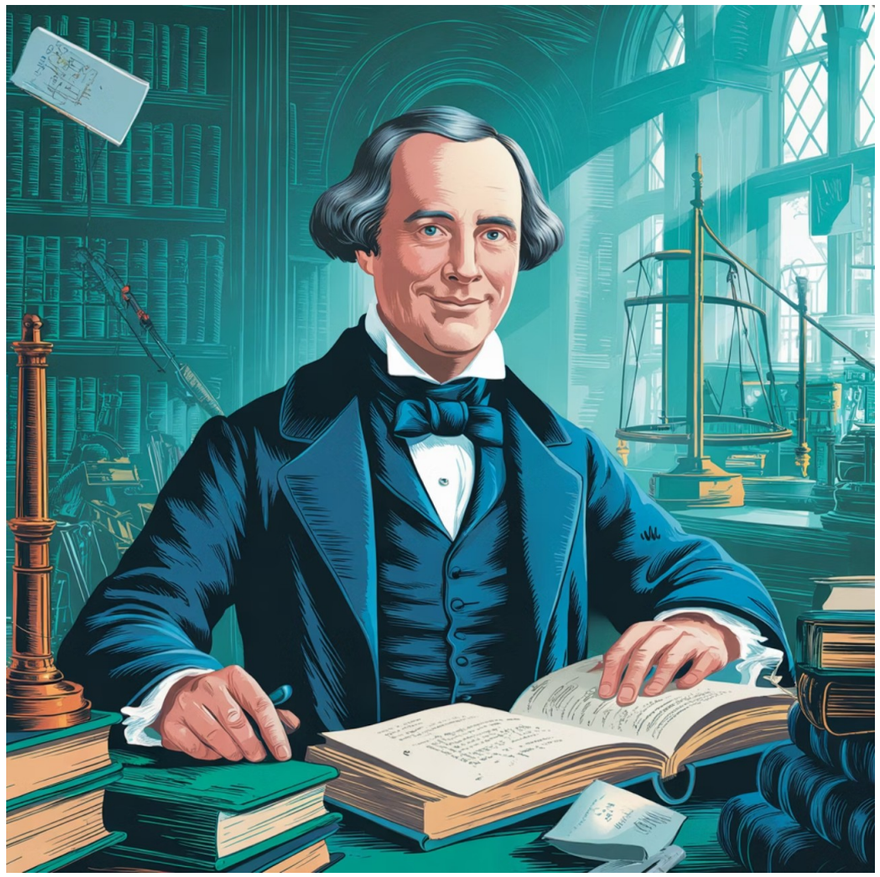
ЧАСТЬ 2

Основы булевой алгебры

Булева алгебра — это математический фундамент, на котором построена вся современная цифровая электроника. Она предоставляет формальный аппарат для описания и анализа логических схем, позволяя проектировать всё — от простейших калькуляторов до процессоров современных суперкомпьютеров.



Что такое булева алгебра?



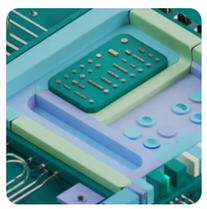
Булева алгебра — это математическая система для работы с логическими значениями, которые могут принимать только два состояния: **Истина (1)** и **Ложь (0)**. Эта алгебра кардинально отличается от привычной нам арифметики.

Система была разработана английским математиком **Джорджем Булем** в 1854 году в его труде "Исследование законов мысли". Первоначально она предназначалась для формализации логических рассуждений, но спустя почти столетие стала основой проектирования цифровых схем.

Булева алгебра позволяет описывать, анализировать и оптимизировать логические функции, что делает её незаменимым инструментом в разработке цифровых устройств любой сложности.

Основные логические операции булевой алгебры

Все сложные цифровые устройства строятся из комбинаций четырёх базовых логических операций. Каждая операция имеет своё графическое обозначение в виде логического элемента.



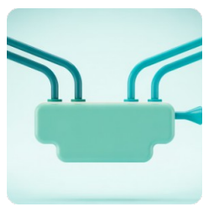
Отрицание (NOT, ¬)

Инверсия входного значения. Если вход равен 0, выход будет 1, и наоборот. Обозначается кружком на выходе элемента.



Конъюнкция (AND, ·)

Логическое умножение. Выход истинен только когда все входы истинны. Аналог операции "И" в естественном языке.



Дизъюнкция (OR, +)

Логическое сложение. Выход истинен, если хотя бы один вход истинен. Аналог операции "ИЛИ" в естественном языке.



Исключающее ИЛИ (XOR, ⊕)

Выход истинен, когда входы различны. Используется в сумматорах и схемах контроля чётности.

Таблицы истинности для основных операций

Таблица истинности — это полное описание поведения логической функции для всех возможных комбинаций входных значений. Для функции от N переменных таблица содержит 2^N строк.

NOT (Отрицание)

A	¬A
0	1
1	0

AND (Конъюнкция)

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

OR (Дизъюнкция)

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

XOR (Исключающее ИЛИ)

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Визуализация таблиц истинности

Таблицы истинности наглядно демонстрируют поведение каждой логической операции. Комбинация этих базовых операций позволяет создавать сколь угодно сложные логические функции, необходимые для работы цифровых устройств.

Законы булевой алгебры

Законы булевой алгебры позволяют преобразовывать и упрощать логические выражения, что критически важно для оптимизации цифровых схем. Эти законы аналогичны законам обычной алгебры, но имеют свои особенности.

1

Закон идемпотентности

Повторение переменной не меняет результата:

$$A + A = A \qquad A \quad A = A$$

2

Закон дополнения

Переменная и её отрицание дают константу:

$$A + \neg A = 1 \qquad A \quad \neg A = 0$$

3

Закон коммутативности

Порядок операндов не важен:

$$A + B = B + A \qquad A \quad B = B \quad A$$

4

Закон ассоциативности

Группировка операндов не влияет на результат:

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

5

Закон дистрибутивности

Распределительный закон:

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

6

Правила де Моргана

Связь между операциями AND и OR через отрицание:

$$\neg(A \quad B) = \neg A + \neg B \qquad \neg(A + B) = \neg A \quad \neg B$$

Пример применения закона де Моргана

Исходное выражение

$$F = \neg(A \quad B \quad C)$$

Применение закона де Моргана

$$F = \neg A + \neg B + \neg C$$

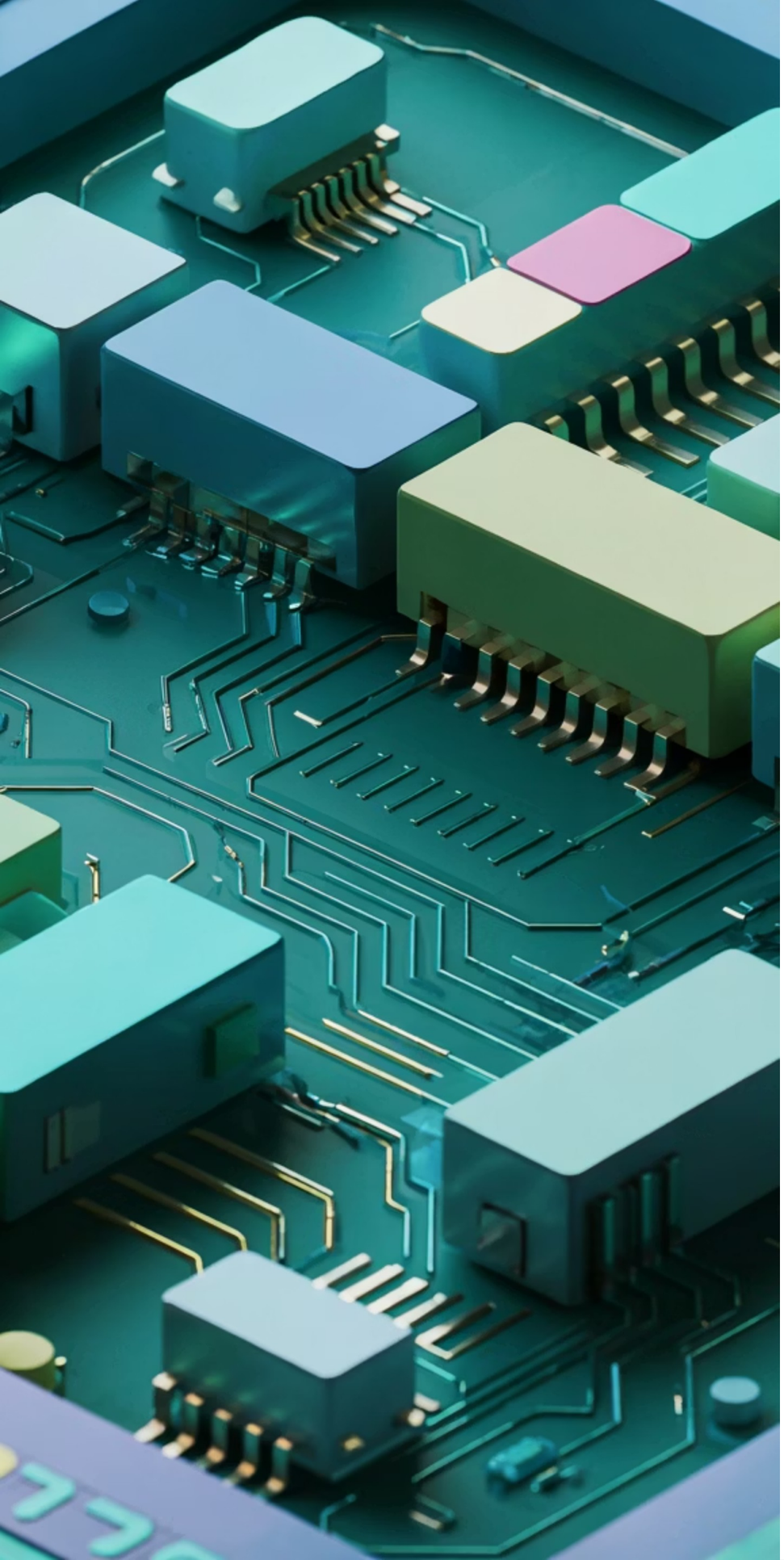
Первая форма требует трёхвходового элемента AND и один инвертор. Вторая форма использует три инвертора и один трёхвходовой элемент OR.

Практическая важность

Преобразование логических выражений с помощью законов де Моргана имеет огромное практическое значение:

- Позволяет выбрать более дешёвую схемную реализацию
- Может уменьшить задержку распространения сигнала
- Даёт возможность использовать доступные типы логических элементов
- Упрощает тестирование и отладку схем

В промышленных проектах оптимизация логических схем может привести к экономии миллионов транзисторов в массовом производстве.



ЧАСТЬ 3

Представление и синтез цифровых устройств

Синтез цифровых устройств — это процесс преобразования абстрактного математического описания в конкретную электронную схему. Этот процесс включает представление логических функций, их оптимизацию и физическую реализацию с помощью логических элементов.

Логические функции и их представление

Булева функция — это отображение из множества булевых переменных в булево значение {0, 1}. Существует несколько эквивалентных способов представления одной и той же функции, каждый из которых удобен для разных задач.

Таблицы истинности

Полное табличное описание функции для всех возможных входных комбинаций. Наиболее наглядный способ, но громоздкий для функций с большим числом переменных.

Алгебраические выражения

Представление в виде булевых формул с операциями AND, OR, NOT. Компактно и удобно для преобразований. Различают дизъюнктивную (СДНФ) и конъюнктивную (СКНФ) нормальные формы.

Карты Карно

Графический метод представления и минимизации функций 2-6 переменных. Соседние ячейки отличаются значением только одной переменной, что упрощает визуальное нахождение упрощений.

Минимизация булевых функций

Минимизация — это процесс нахождения эквивалентного логического выражения с минимальным числом операций. Это критически важно для создания эффективных цифровых устройств.

Зачем нужна минимизация?

- Уменьшение количества логических элементов
- Снижение стоимости производства
- Уменьшение энергопотребления
- Повышение быстродействия схемы
- Увеличение надёжности

Методы минимизации

01

Карты Карно

Визуальный метод для 2-6 переменных

02

Метод Квайна-Мак-Класки

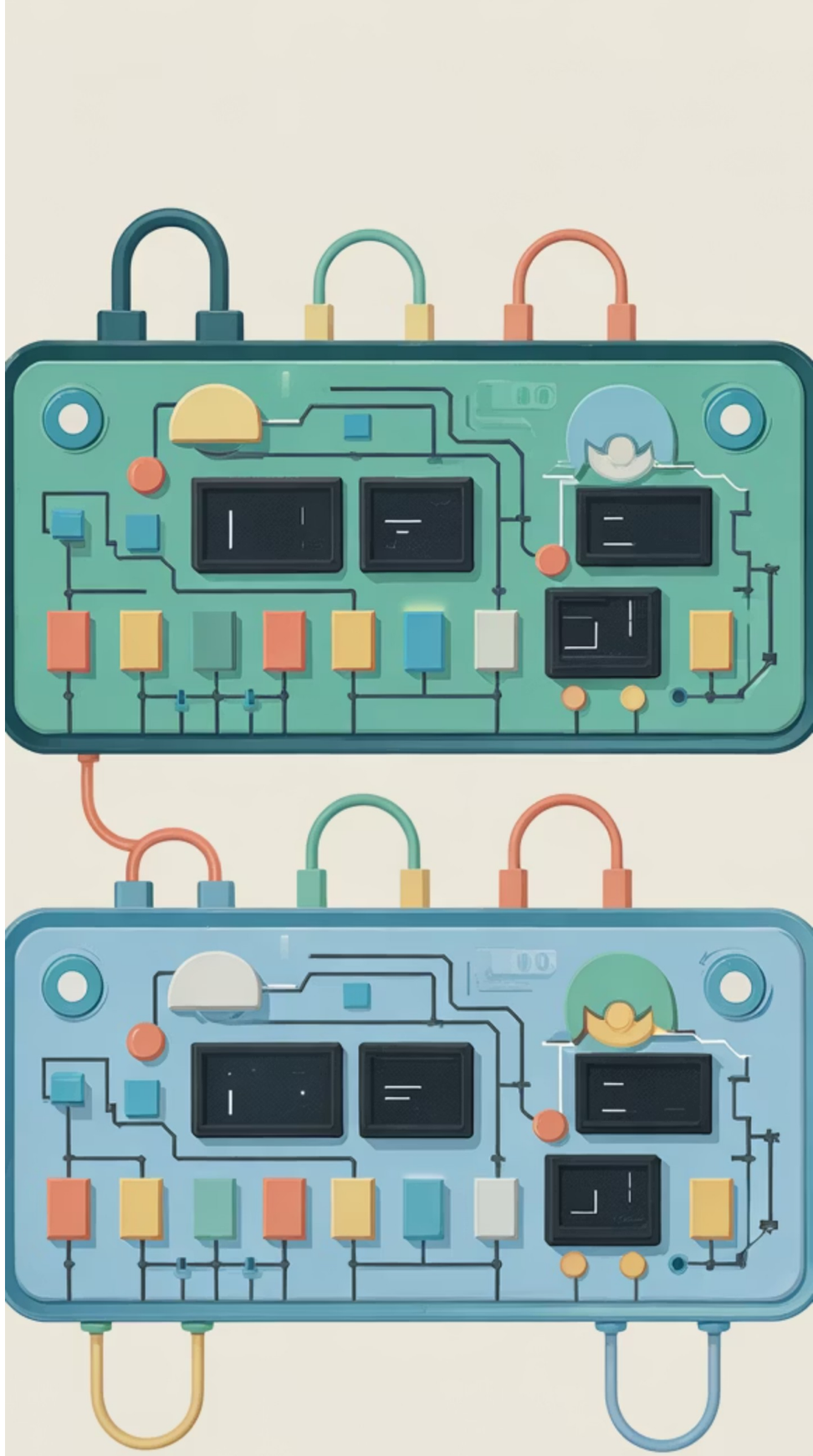
Алгоритмический метод для любого числа переменных

03

САПР

Автоматизированные системы для сложных функций

> Пример: функция от 3 переменных F(A,B,C) = Σ(1,3,5,7) может быть минимизирована с F = A до F = C, что сокращает количество элементов в 3 раза.



Пример карты Карно

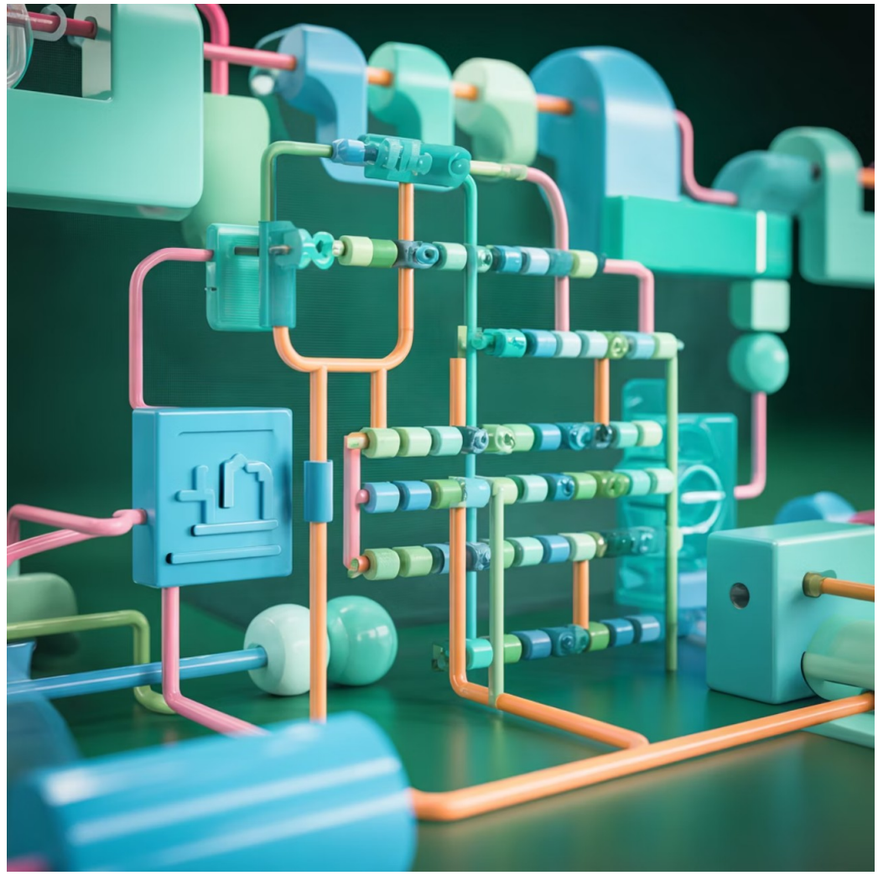
Карта Карно — это двумерная таблица, где каждая ячейка соответствует одной строке таблицы истинности. Соседние ячейки (включая граничащие через край) отличаются только одной переменной, что позволяет визуально находить группы для минимизации.

Правила работы с картой Карно: Группируем единицы (или нули) в прямоугольные группы размером 2^n (1, 2, 4, 8...). Чем больше группа, тем меньше переменных в результирующем выражении. Группы могут перекрываться и замыкаться через края карты.



Комбинационные логические схемы

Комбинационные схемы — это цифровые устройства без памяти, у которых выходные сигналы в любой момент времени однозначно определяются текущими значениями входных сигналов. Они не имеют обратных связей и не зависят от предыстории.



Сумматоры

Выполняют арифметическое сложение двоичных чисел

Мультиплексоры

Выбирают один из нескольких входов на основе управляющего сигнала

Дешифраторы

Преобразуют двоичный код в активацию одной из выходных линий

Компараторы

Сравнивают два числа и определяют отношение между ними

Последовательные логические схемы

Последовательные схемы — это устройства с памятью, выходы которых зависят не только от текущих входов, но и от предыдущих состояний системы. Они содержат элементы памяти и обратные связи.



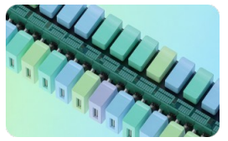
Триггеры

Базовые элементы памяти, хранящие один бит информации. Типы: D-триггер, JK-триггер, RS-триггер. Переключаются по фронту тактового сигнала.



Счётчики

Последовательно проходят через заданную последовательность состояний при каждом тактовом импульсе. Используются для подсчёта событий и формирования временных интервалов.



Регистры

Группы триггеров для хранения многобитовых данных. Могут иметь последовательный или параллельный ввод/вывод. Основа для построения памяти.

Пример синтеза комбинационной схемы: двухразрядный сумматор

Рассмотрим процесс синтеза полного одноразрядного двоичного сумматора — базового элемента любого арифметического устройства. Он принимает два бита слагаемых (A и B) и бит переноса с предыдущего разряда (Cin), выдавая сумму (S) и перенос в следующий разряд (Cout).

Таблица истинности

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Булевы функции

Из таблицы истинности выводим логические выражения:

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + AC_{in} + BC_{in}$$

Сумма реализуется через операцию XOR, а перенос — через комбинацию AND и OR. Для двухразрядного сумматора два таких элемента соединяются последовательно.

Современные процессоры используют более сложные схемы сумматоров (сумматоры с опережающим переносом), которые работают значительно быстрее за счёт параллельного вычисления переносов.

Практическое значение булевой алгебры в цифровой электронике

Булева алгебра — это не просто абстрактная математическая теория. Она является фундаментом всей современной цифровой индустрии, позволяя превращать идеи в работающие электронные устройства.

Оптимизация схем

Применение законов булевой алгебры позволяет минимизировать количество логических элементов, что напрямую снижает стоимость производства чипов. В масштабах миллиардов транзисторов даже 1% оптимизации даёт колоссальную экономию.

Повышение быстродействия

Меньше логических элементов означает меньше задержек распространения сигнала. Оптимизированные схемы работают быстрее, что критично для высокопроизводительных процессоров.

Снижение энергопотребления

Каждое переключение логического элемента потребляет энергию. Минимизация схем ведёт к снижению энергопотребления — ключевому параметру для мобильных устройств и дата-центров.

Проектирование микропроцессоров

Все процессоры, от простейших микроконтроллеров до многоядерных CPU, спроектированы с использованием булевой алгебры. Арифметико-логические устройства, блоки управления, кэш-память — всё это результат применения булевых функций.

Современные тенденции и применение

FPGA — программируемые логические матрицы

Field-Programmable Gate Arrays — это микросхемы, которые можно многократно программировать для реализации любых цифровых схем. Они содержат миллионы настраиваемых логических блоков.

- Быстрое прототипирование цифровых систем
- Обработка сигналов в реальном времени
- Ускорители машинного обучения
- Телекоммуникационное оборудование

ASIC — специализированные интегральные схемы

Application-Specific Integrated Circuits — чипы, спроектированные для конкретной задачи. Обеспечивают максимальную производительность и эффективность.

- Майнинг криптовалют
- Графические процессоры
- Сетевое оборудование
- Специализированные ускорители AI

Автоматизированные средства проектирования

Современные САПР (системы автоматизированного проектирования) используют сложные алгоритмы на основе булевой алгебры для автоматического синтеза, оптимизации и верификации цифровых схем, содержащих миллиарды транзисторов. Инструменты вроде Vivado, Quartus и Synopsis Design Compiler позволяют проектировать схемы, которые было бы невозможно создать вручную.

Архитектура современного цифрового устройства

Современное цифровое устройство представляет собой иерархию взаимосвязанных логических блоков, каждый из которых построен на принципах булевой алгебры. От простейших логических элементов до сложнейших процессоров — всё подчиняется законам, открытым Джорджем Булем более 150 лет назад.

Заключение

Булева алгебра и системы счисления — фундамент цифровой техники

Без понимания двоичной системы и логических операций невозможно разобраться в работе любого цифрового устройства, от простейшего калькулятора до суперкомпьютера.

Математический аппарат для проектирования

Законы булевой алгебры предоставляют формальные методы для синтеза, анализа и оптимизации цифровых схем любой сложности.

Путь к сложным вычислительным системам

Освоение этих основ открывает дорогу к глубокому пониманию архитектуры компьютеров, проектированию микропроцессоров и созданию инновационных цифровых решений.

Вопросы для обсуждения

1. Почему двоичная система оптимальна для электроники?
2. Как минимизация влияет на характеристики схем?
3. В чём разница между комбинационными и последовательными схемами?
4. Какие современные технологии используют булеву алгебру?

> *"Математика — царица наук, а булева алгебра — царица цифрового мира."*