

Логические основы алгоритмизации

История

Начало исследований в области **формальной логики** было положено **Аристотелем** в IV в. до н.э.

Однако **математические подходы** к этим вопросам были впервые указаны **Джорджем Булем**, который положил в основу математической логики **алгебру логики**.

Алгебра логики оперирует с высказываниями, т.е. повествовательными предложениями, о которых можно сказать (И) - Истинно оно или (Л) - Ложно.

Элементарные логические операции

1. Конъюнкция

Опред. Соединение двух (или нескольких) высказываний в одно с помощью союза И (AND) называется конъюнкцией (или операцией логического умножения). Обозначаются \wedge , $\&$, \cdot .

Значения логических операций определяются по правилам, задаваемым в таблице истинности.

Истинность конъюнкции задается следующей таблицей.

A	B	A&B
0	0	0
0	1	0
0	0	0
1	1	1

2. Дизъюнкция

Опред. Соединение двух (или нескольких) высказываний в одно с помощью союза ИЛИ (OR) называется дизъюнкцией (или логического сложения). Обозначаются \parallel , \vee , $+$.

Таблица истинности

A	B	$A \vee B$	$A \text{ xor } B$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

Хор-модифицированная операция «ИЛИ» исключающее или (xor), от обычного «ИЛИ» отличается последней строкой.

3. Отрицание (инверсия)

Опред. Присоединение частицы НЕ (NOT) к данному высказыванию называется операцией отрицания (инверсии). \bar{A} , $\neg A$ – «не A»

Таблица истинности

A	$\neg A$
1	0
0	1

4. Эквивалентность

Служит для задания высказываний «*тогда и только тогда, когда ...*» или «*если и только если*» ... Обозначаются $A \sim B$
($A \equiv B, A \text{ eqv } B$)

Таблица истинности

A	B	$A \sim B$
T	T	T
T	F	F
F	T	F
F	F	T

4. Импликация

Служит для задания так называемых условных высказываний «если..., то ...» «когда ..., тогда ». Обозначаются $A \rightarrow B$, ($A \text{ imp } B$).

Таблица истинности

A	B	$A \rightarrow B$
T	T	T
T	F	F
F	F	T
F	T	T

Опред. Высказывания, образованные с помощью нескольких логических операций называются **сложными**. Истинность их устанавливают, используя **таблицы истинности** соответствующих операций.

Таблица истинности - таблица, с помощью которой устанавливается истинностное значение сложного высказывания при данных значениях входящих в него простых высказываний.

Пример: определим истинность сложного высказывания $A \& B$

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

Опред. Высказывания, у которых таблицы истинности совпадают, называются **равносильными**. Для обозначения используют знак $=$ ($A=B$).

Пример: рассмотрим сложное высказывание $(A \& B) \vee (\bar{A} \& \bar{B})$ и составим таблицу истинности

A	B	\bar{A}	\bar{B}	$A \& B$	$\bar{A} \& \bar{B}$	$(A \& B) \vee (\bar{A} \& \bar{B})$
0	0	1	1	0	1	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	0	1

если сравнить с таблицей истинности, для эквивалентности, то видно, что высказывания равносильны

Свойства логических операций

1. коммутативность

$$A \& B = B \& A$$

$$A \vee B = B \vee A$$

2. закон идемпотентности

$$A \& A = A, \quad A \vee A = A$$

3. ассоциативные законы

$$A \vee (B \vee C) = (A \vee B) \vee C = A \vee B \vee C$$

$$A \& (B \& C) = (A \& B) \& C = A \& B \& C$$

4. дистрибутивные законы

$$A \& (B \vee C) = (A \& B) \vee (A \& C)$$

$$A \vee (B \& C) = (A \vee B) \& (A \vee C)$$

5. законы де Моргана

$$\neg (A \& B) = \neg A \vee \neg B$$

$$\neg (A \vee B) = \neg A \& \neg B$$

6. закон универсального множества

$$X \vee 1 = 1$$

$$X \& 1 = X$$

7. закон нулевого множества

$$X \vee 0 = X$$

$$X \& 0 = 0$$

Схемная реализация базовых ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

Логические элементы (ЛЭ) - это электронные схемы с одним или несколькими входами и одним выходом, через которые проходят электрические сигналы, представляющие 0,1.

Для реализации любой логической операции над двоичными сигналами достаточно элементов трех типов: **И**, **ИЛИ**, **НЕ**, **функционально полная система**, все остальные можно построить через них.

Существуют микросхемы, реализующие более сложные логические функции: **И-НЕ**, называемая операцией **Шеффера** ($\bar{A}B$), и **ИЛИ-НЕ**, называемая **Стрелка Пирса** ($A+B$).

Реальная аппаратура строится из логических элементов подобно тому, как сложная логическая функция получается путем комбинации более простых функций.

Схемная реализация базовых ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ.

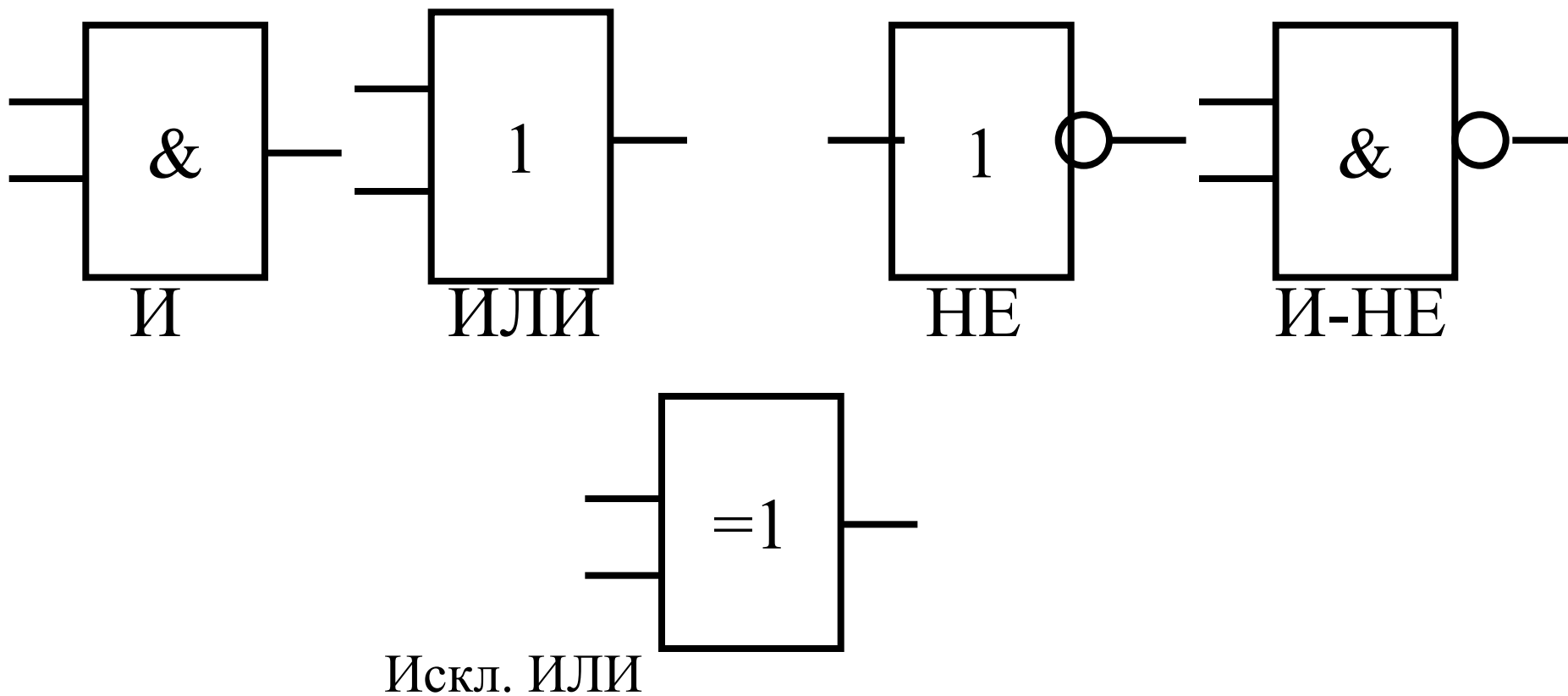
Из логических элементов путем их комбинации строятся основные схемы компьютера.

Триггер - электронный прибор, имеющий два устойчивых состояния, является типичным запоминающим элементом, способным хранить 1 бит информации.

Регистр - совокупность триггеров, предназначенных для хранения числа в двоичном коде.

Сумматор - устройство, обеспечивающее суммирование двоичных чисел с учетом переноса из предыдущего разряда.

Обозначения базовых логических элементов



Полусумматор

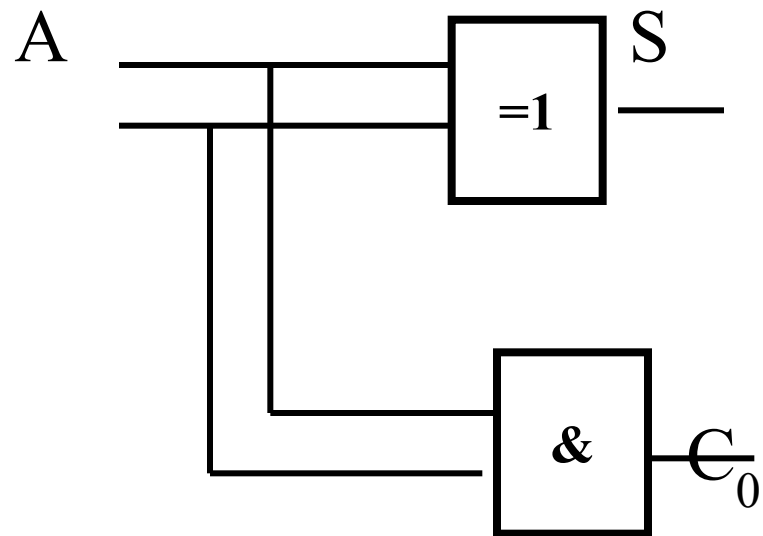
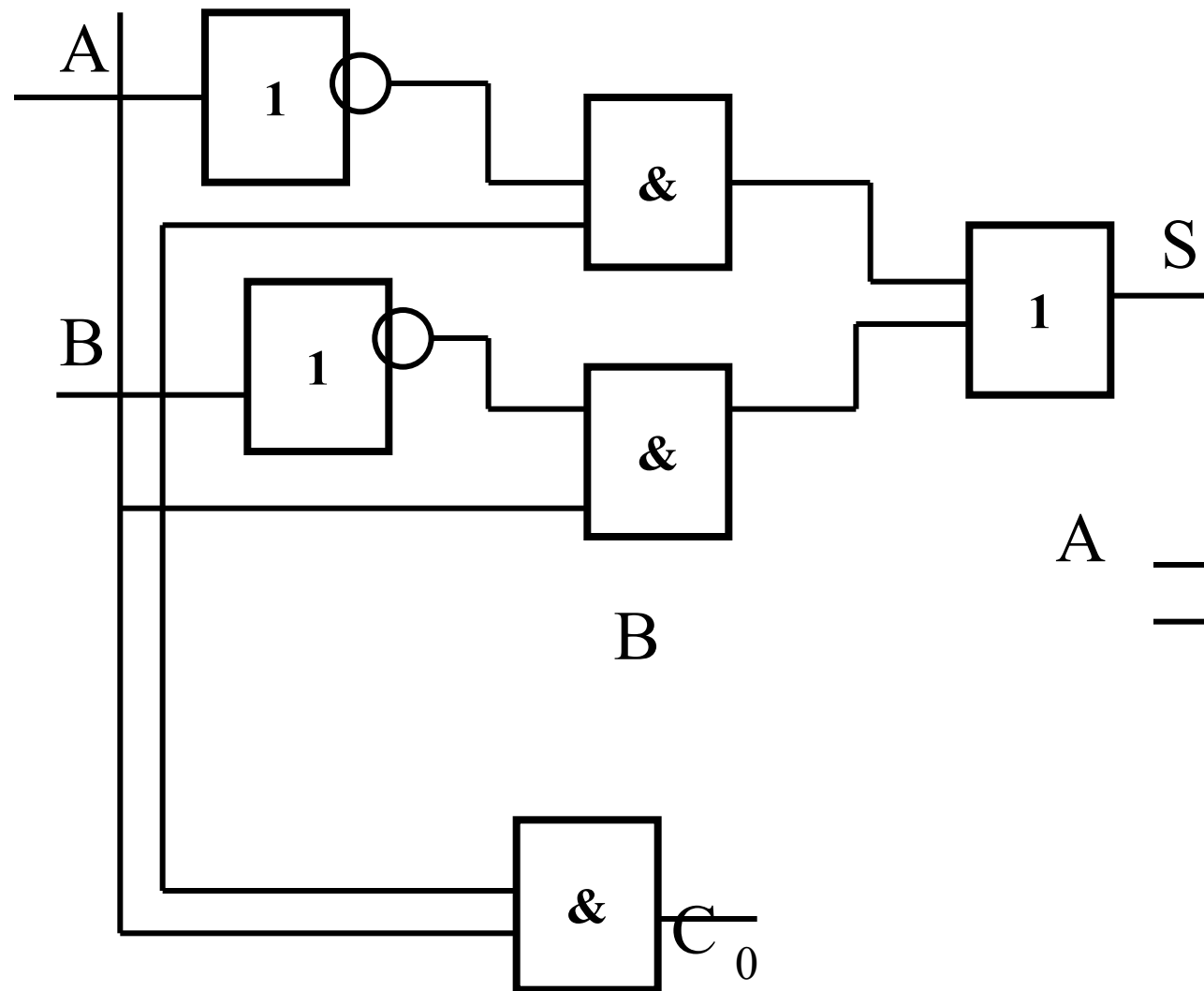
Полусумматор реализует сложение 2-х одноразрядных двоичных чисел A и B. В результате получается, вообще говоря, 2-х разрядно двоичное число. Его младшую цифру обозначим S, а старшую, которая при сложении многоразрядных чисел будет перенесена в старший разряд, через C_0

Тогда используя таблицы истинности и перебирая все четыре (00 01 10 00) возможные случая значений A, B обе цифры S и C_0 можно, оказывается, получить по следующим логическим формулам $S = (\neg A \& B) \vee (A \& \neg B)$, $C_0 = (A \& B)$

Таблица истинности для полусумматора.

A	B	S	C_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Логическая схема полусумматора



Логические основы построения цифровых автоматов.

Логический синтез переключательных и вычислительных схем. Синтез переключательных схем

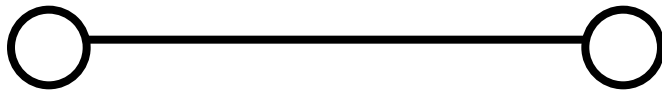
Переключательная схема — схематическое изображение некоторого устройства, состоящего из переключателей и соединяющих их проводников, а также входов и выходов, на которые передается и с которых снимается электрический сигнал.

Каждый переключатель имеет только два состояния: замкнутое разомкнутое. Переключателю X поставим в соответствие логическую переменную x , которая принимает значение 1 только в том случае, когда переключатель X замкнут и схема проводит ток; если же переключатель разомкнут, то переменная x равна нулю. При этом два переключателя X и \bar{X} связаны таким образом, что когда X замкнут, то \bar{X} разомкнут, и наоборот, Следовательно, если переключателю X поставлена в соответствие логическая переменная x , то переключателю \bar{X} должна соответствовать переменная \bar{x} .

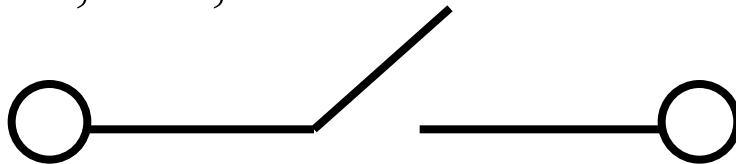
Всей переключательной схеме также можно поставить в соответствие логическую переменную, равную единице, если схема проводит ток, и равную нулю — если не проводит. Эта переменная является функцией от переменных, соответствующих всем переключателям схемы, и называется **функцией проводимости.**

Функции проводимости F некоторых переключательных схем

1. Схема не содержит переключателей и проводит ток всегда, следовательно, $F = 1$;



2. Схема содержит один постоянно разомкнутый контакт, следовательно, $F = 0$;



3. Схема проводит ток, когда переключатель x замкнут, и не проводит, когда x разомкнут, следовательно, $F(x) = x$;



4. Схема проводит ток, когда переключатель \bar{x} разомкнут, и не проводит, когда \bar{x} замкнут, следовательно, $F(x) = \bar{x}$;



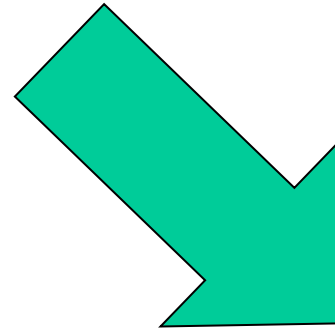
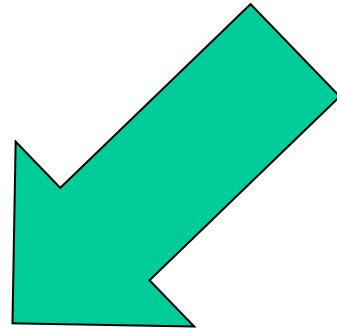
5. Схема проводит ток, когда оба переключателя замкнуты, следовательно, $F(x) = x \cdot y$;



6. Схема проводит ток, когда хотя бы один из переключателей замкнут, следовательно, $F(x) = x \vee y$.



Задачи, возникающие при рассмотрении переключательных схем



Задача синтеза

**Задача анализа
схемы**

Этапы синтеза переключательной схемы

1. Составление функции проводимости по заданным условиям
2. Упрощение этой функции.
3. Построение соответствующей схемы.

Этапы анализа схемы

1. Определение значений функции проводимости при всех возможных наборах, входящих в эту функцию переменных.
2. Получение упрощенной формулы.

СКНФ и СДНФ

Совершенной дизъюнктивной нормальной формой (СДНФ)

называют наиболее полную форму записи логического выражения. Эта форма записи представляет собой сумму, каждое слагаемое которой является произведением всех входных аргументов или их инверсий, например:

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

СДНФ является избыточной, но логические функции, записанные в СДНФ, легко сравнивать между собой и их удобно преобразовывать в таблицы истинности. Булево выражение, полученное из таблицы истинности логической функции, имеет совершенную дизъюнктивную нормальную форму.

Еще одной формой записи логического выражения является **совершенная конъюнктивная нормальная форма (СКНФ)**. Это произведение сомножителей, каждый из которых является суммой всех входных аргументов или их отрицаний, например:

$$F = (\neg A + B + \neg C) (\neg A + B + C) (A + \neg B + C) (A + B + C).$$

Этапы синтеза переключательных схем

1. Образование **СДНФ*** (**СКНФ****) функции по заданной таблице истинности.
 - * совершенной дизъюнктивной нормальной формой
 - ** совершенной конъюнктивной нормальной формой
2. Упрощение этой функции (преобразованию СДНФ (СКНФ) в формулу с наименьшим числом вхождений переменных);
3. Построение соответствующей схемы.

Образование СДНФ функции по заданной таблице истинности

Этот этап включает в себя:

1. в заданной таблице истинности выделяют наборы значений аргументов, при которых функция принимает единичное значение;
2. для каждого выделенного набора образуется **конституэнта** единицы, принимающая единичное значение при данном наборе значений аргументов;
3. составляется логическая сумма образованных конституэнт единицы.

Для образования конституэнты единицы C_i^1 , принимающей единичное значение в i -ом наборе значений аргументов необходимо составить логическое произведение аргументов, в которое аргументы, принимающие в i -м наборе единичное значение, входят без знака отрицания, а аргументы, принимающие в i -м наборе новое значение, входят со знаком отрицания.

Образование СКНФ функции

1. В таблице выделяются наборы значений аргументов, при которых функции принимает нулевое значение;
2. Для каждого выделенного набора образуется конституэнта поля, принимавшая нулевое значение при данном наборе значений аргументов;
3. Составляется логическое произведение образованных конституэнт поля.

Упрощение функции

При преобразовании СДНФ (СКНФ) в формулу с наименьшим числом вхождений переменных (минимизация формулы) используют аксиомы и законы булевой алгебры:

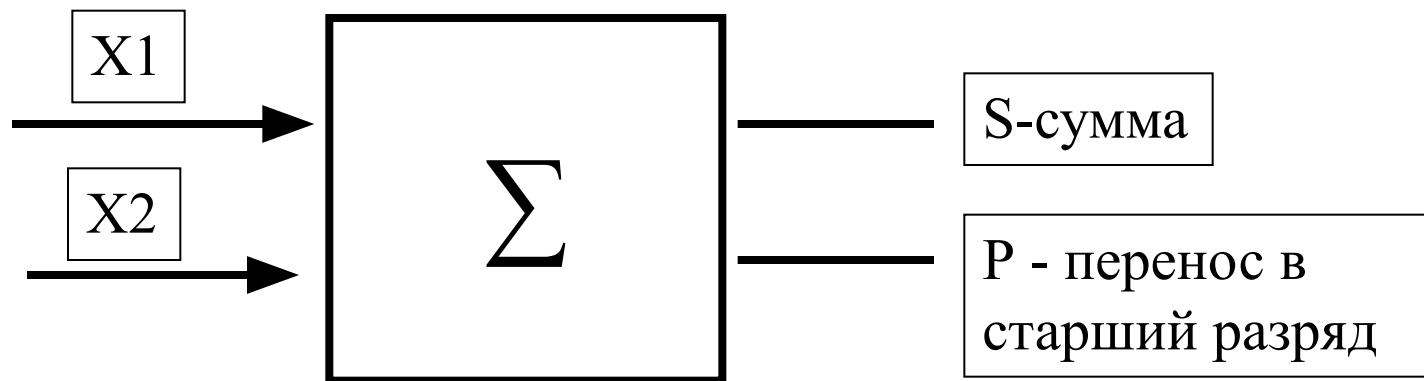
- вынос за скобки $XY \vee XZ = X(Y \vee Z)$;
- полное склеивание $XY \vee X\bar{Y} = X$;
- поглощение $X \vee XY = X$;
- минимизация по методу Квайна;
- минимизация с использованием карт Карно или диаграмм Вейча.

При минимизации по методу Квайна предполагается, что исходная функция задана в СДНФ.

Введем несколько определений. Конъюнкция, получаемая в результате склеивания двух конституэнт единицы, называется **импликантой**. Импликанта поглощает конституэнты единицы, при склеивании которых она образовалась.

Построение схемы

В качестве примера логического синтеза вычислительных схем рассмотрим построение одноразрядного **двоичного сумматора**, имеющего два входа (x_1 и x_2) и два выхода (S и P)



Зададим таблицу истинности сумматора.

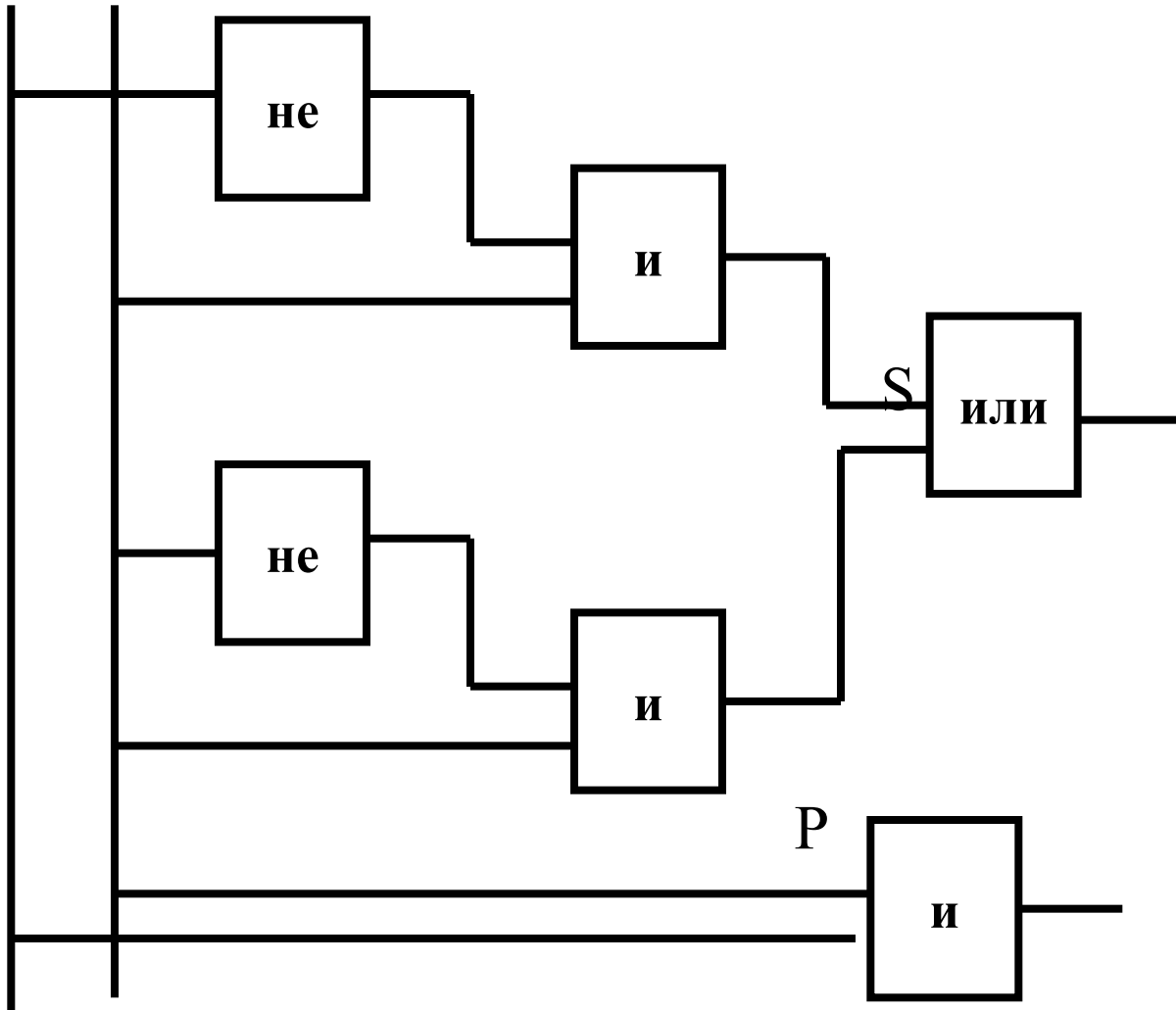
x1	0	0	1	1
x2	0	1	0	1
S=f1(x1,x2)	0	1	1	0
P=f2(x1,x2)	0	0	0	1

Представим выходные функции S и P в виде СДНФ:

$$S=f1(x1,x2) = \neg x1 \wedge x2 + x1 \wedge \neg x2 = x1 \text{ xor } x2$$

$$P=f2(x1, x2) = x1 \wedge x2$$

Логическая схема сумматора, реализующего данные функции, представлена на рисунке



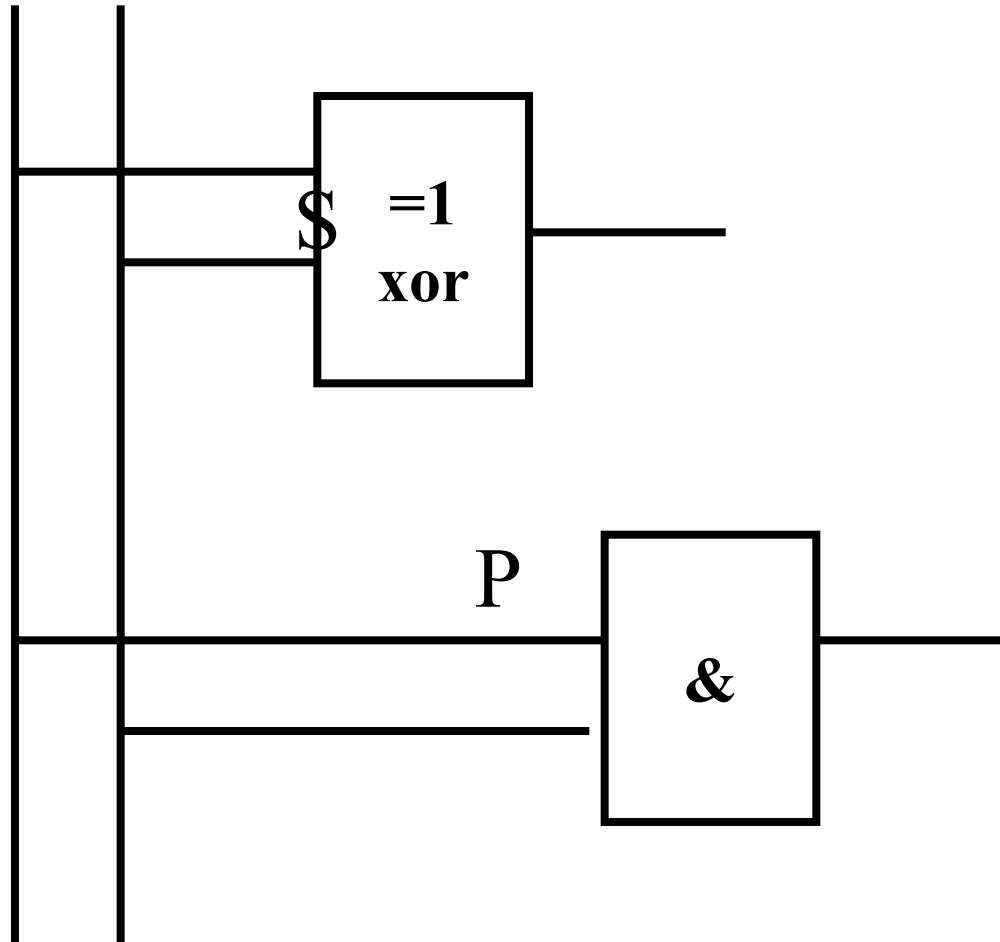
Для логических схем (И, ИЛИ, НЕ) существуют типовые технические схемы (логические элементы), реализующие их на полупроводниковых структурах, т.е. аппаратно.

Использование знаков 0 и 1 подчеркивает некоторое соответствие между значениями логических переменных и функций в математической логике и цифрами в двоичной системе счисления.

Это позволяет описывать работу логических схем ПК и проводить их анализ и синтез с помощью математического аппарата алгебры логики. Любое устройство ПК - некоторый функциональный преобразователь. Причем входы- значения логических переменных, выход - значения логических функций, т.е. устройство ПК \sim функция.

Логический элемент- часть электронной логической схемы, которая реализует элементарную логическую функцию.

Тогда структурная схема сумматора будет иметь вид,
показанный на рисунке



<http://www.seun.ru/faculty/FIT/KTOIT/informatika>
Лекция №5