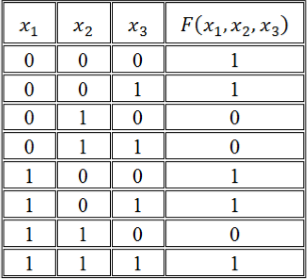
## Архитектура вычислительных систем Лабораторная работа № 1

**Вариант 2**

1. Составить таблицу истинности логического выражения (A∨B)∧(A∨¬C)

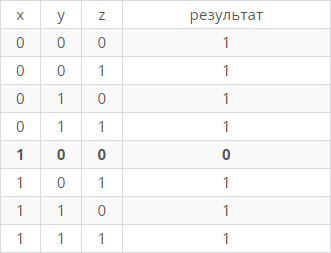
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | A∨B | ¬C | A∨¬C | (A∨B)∧(A∨¬C) |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |

1. Построить логическую функцию по таблице истинности



¬*x*1¬*x*2¬*x*3∨¬*x*1¬*x*2*x*3∨*x*1¬*x*2¬*x*3∨*x*1¬*x*2*x*3∨*x*1*x*2*x*3

1. По таблице истинности построить логическую функцию с помощью СДНФ и минимизировать

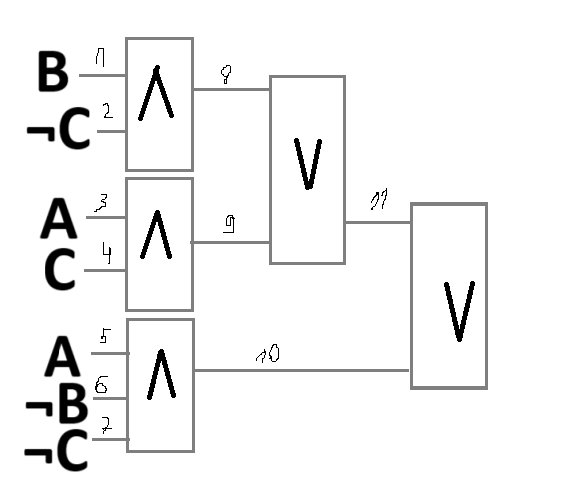


# ¬x¬y¬z ∨ ¬x¬yz ∨ ¬xy¬z ∨ ¬xyz ∨ x¬yz ∨ xy¬z ∨ xyz

Упростим логическую функцию, используя правило склеивания:

# ¬x¬y¬z ∨ ¬x¬yz ∨ ¬xy¬z ∨ ¬xyz ∨ x¬yz ∨ xy¬z ∨ xyz =

¬x¬y ∨ ¬xy ∨ xz ∨ xy¬z = ¬x ∨ xz ∨ xy¬z

1. Сделать оценку сложности функциональной схемы. B¬C ∨ AC ∨ A¬B¬C

С = 11.

**Контрольные вопросы:**

## Для чего используется алгебра логики?

Алгебра логики используется для:

* + - Анализа и синтеза логических схем в цифровых устройствах (компьютерах, процессорах, микроконтроллерах).
    - Описания и упрощения логических выражений.
    - Проектирования алгоритмов и программ.
    - Решения задач в области искусственного интеллекта, теории автоматов и теории информации.

## Дайте определение - Двоичные переменные, Переключательные функции, Логический элемент компьютера

* + - Двоичные переменные - это переменные, которые могут принимать только два значения: 0 (ложь) или 1 (истина). Они используются для описания логических состояний.
    - Переключательные функции - это функции, которые принимают двоичные переменные на входе и возвращают двоичное значение на выходе. Они описывают поведение логических схем.
    - Логический элемент компьютера - это базовый компонент цифровой схемы, который реализует одну из логических функций (например, "И", "ИЛИ", "НЕ"). Логические элементы используются для построения более сложных устройств, таких как процессоры и память.

## Базовые и составные логические функции - краткая характеристика

Базовые логические функции:

* + - "НЕ" возвращает противоположное значение входной переменной.
    - "И" (конъюнкция) возвращает 1, только если все входные переменные равны 1.
    - "ИЛИ" (дизъюнкция) возвращает 1, если хотя бы одна входная переменная равна 1.

Составные логические функции:

* + - "И-НЕ" (NAND) - комбинация "И" и "НЕ". Возвращает 0, только если все входные переменные равны 1.
    - "ИЛИ-НЕ" (NOR) - комбинация "ИЛИ" и "НЕ". Возвращает 1, только если все входные переменные равны 0.
    - "Исключающее ИЛИ" (XOR) возвращает 1, если количество входных переменных, равных 1, нечётное.

## Назовите четыре основных закона булевой алгебры

Коммутативный закон:

A ∧ B = B ∧ A (для "И").

A ∨ B = B ∨ A (для "ИЛИ").

Ассоциативный закон:

A ∧ (B ∧ C) = (A ∧ B) ∧ C (для "И").

A ∨ (B ∨ C) = (A ∨ B) ∨ C (для "ИЛИ").

Дистрибутивный закон:

A ∧ (B ∨ C) = (A ∧ B) ∨ (A ∧ C) (дистрибутивность "И" относительно "ИЛИ"). A ∨ (B ∧ C) = (A ∨ B) ∧ (A ∨ C) (дистрибутивность "ИЛИ" относительно "И"). Закон де Моргана:

¬(A ∧ B) = ¬A ∨ ¬B.

¬(A ∨ B) = ¬A ∧ ¬B.

* 1. **Этапы построения логической схемы** Составление таблицы истинности для функции. Построение логической функции в СДНФ. Минимизация логической функции.

Реализация схемы с использованием заданных базисных элементов (например, "И", "ИЛИ", "НЕ").

## Чем вызвана необходимость упрощения переключательной функции путем приведения ее к дизъюнктивной нормальной форме?

Уменьшением количества логических элементов в схеме, что снижает её сложность и стоимость.

Повышением быстродействия схемы за счёт уменьшения количества уровней логики. Упрощением анализа и проектирования схем.

## Что такое цена схемы? Как ее определить?

Цена схемы — это мера сложности схемы, которая определяется количеством логических элементов (вентилей) и количеством входов в этих элементах.

Чтобы определить цену схемы, нужно:

Подсчитать количество логических элементов (например, "И", "ИЛИ", "НЕ"). Учесть количество входов в каждом элементе.

Суммировать общее количество входов или элементов.