ДИСЦИПЛИНА	Архитектура комьютера и язык ассемблера
ИНСТИТУТ	Передовая инженерная школа СВЧ-электроники
КАФЕДРА	Передовых технологий
ВИД УЧЕБНОГО МАТЕРИАЛА	Методические указания по дисциплине
ПРЕПОДАВАТЕЛЬ	Астафьев Рустам Уралович
СЕМЕСТР	1 семестр, 2025/2026 уч. год

Ссылка на материал:
https://github.com/astafiev-rustam/computer-architecture-and-assembly-language/tree/practice-1-2

Практическое занятие №2: Основы булевой логики

Основы булевой алгебры и логики

Булева алгебра, основанная Джорджем Булем в XIX веке, представляет собой математический аппарат для работы с логическими высказываниями, которые могут принимать лишь два значения: **истина (1)** или **ложь (0)**. Эта система стала теоретической основой проектирования цифровых схем и процессоров, поскольку идеально соответствует двоичной природе вычислительной техники.

Базовые логические операции

Логическое И (AND, конъюнкция)

Результат истинен только когда оба операнда истинны

x_1	x_2	$x_1 \land x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Логическое ИЛИ (OR, дизъюнкция)

Результат ложен только когда оба операнда ложны

x_1	x_2	$x_1 \lor x_2$
0	0	0
0	1	1
1	0	1

x_1	x_2	$x_1 \vee x_2$
1	1	1

Логическое НЕ (NOT, отрицание)

Унарная операция, преобразующая истину в ложь и наоборот

x	¬x
0	1
1	0

Производные логические операции

Логическое И-НЕ (NAND, штрих Шеффера)

Результат противоположен конъюнкции

x_1	x_2	x_1 NAND x_2
0	0	1
0	1	1
1	0	1
1	1	0

Логическое ИЛИ-НЕ (NOR, стрелка Пирса)

Результат противоположен дизъюнкции

x_1	x_2	x_1 NOR x_2
0	0	1
0	1	0
1	0	0
1	1	0

Исключающее ИЛИ (XOR)

Результат истинен когда операнды различны

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1

x_1	x_2	x_1 XOR x_2
1	1	0

Основные законы булевой алгебры

Коммутативность:

$x_1 \wedge x_2 = x_2 \wedge x_1$

$x_1 \vee x_2 = x_2 \vee x_1$

Ассоциативность:

$(x_1 \wedge x_2) \wedge x_3 = x_1 \wedge (x_2 \wedge x_3)$

$(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$

Дистрибутивность:

$x_1 \wedge (x_2 \vee x_3) = (x_1 \wedge x_2) \vee (x_1 \wedge x_3)$

$x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3)$

Законы де Моргана:

$\neg(x_1 \wedge x_2) = \neg x_1 \vee \neg x_2$

$\neg(x_1 \vee x_2) = \neg x_1 \wedge \neg x_2$

Закон двойного отрицания:

$\neg(\neg x) = x$

Практическое применение булевой логики в архитектуре компьютера проявляется в создании **логических вентилей** — физических реализаций булевых операций в виде электронных схем. Комбинации этих вентилей образуют более сложные устройства: сумматоры, мультиплексоры, дешифраторы. Именно на этом принципе строится арифметико-логическое устройство процессора, способное выполнять как арифметические операции, так и логические преобразования над двоичными данными.

Таким образом, булева алгебра служит связующим звеном между абстрактной логикой и физической реализацией вычислительных процессов, обеспечивая математический базис для проектирования цифровых систем любой сложности.

Самостоятельная работа

В качестве базового задания необходимо реализовать в среде Logisim представленную функцию из списка. К заданным функциям можно применять законы и свойства логических функций для упрощения задачи.

1. Находите свой вариант по списку.
2. Составляете таблицу истинности целевой функции.

3. Проектируете в Logisim схему, которая будет воспроизводить поведение заданной функции.
4. Тестируете схему по составленной таблице истинности.
5. В случае, если схема проходит тесты - прикрепляете результаты в СДО.

Функцию необходимо выбирать исходя из своего варианта по списку:

1. `F(A,B,C) = (A ∨ B) ∧ (¬A ∨ C) ∧ (B ∨ C)`
2. `F(A,B,C) = A ∧ B ∨ ¬A ∧ C ∨ B ∧ C`
3. `F(A,B,C,D) = (A ∨ B) ∧ (¬B ∨ C ∨ D) ∧ (A ∨ C)`
4. `F(A,B,C,D) = A ∧ B ∧ C ∨ A ∧ ¬B ∧ D ∨ A ∧ C`
5. `F(A,B,C) = (A ⊕ B) ∧ (B → C)`
6. `F(A,B,C,D) = (A ∨ C) ∧ (B ∨ ¬C) ∧ (¬A ∨ D)`
7. `F(A,B,C) = ¬(A ∧ B) ∨ ¬(¬A ∧ C)`
8. `F(A,B,C,D) = A ∧ B ∨ ¬A ∧ C ∨ B ∧ ¬C ∧ D`
9. `F(A,B,C) = (A ∨ B) ∧ ¬(A ∧ C) ∧ (B ∨ ¬C)`
10. `F(A,B,C,D) = (A → B) ∧ (C ⊕ D) ∨ A ∧ ¬C`
11. `F(A,B,C) = ¬(A ∨ ¬B) ∨ ¬(B ∧ ¬C)`
12. `F(A,B,C,D) = A ∧ (B ∨ C) ∨ ¬A ∧ (C ⊕ D)`
13. `F(A,B,C) = (A ∧ B) ⊕ (A ∨ C)`
14. `F(A,B,C,D) = (A ∨ B ∧ C) ∧ (¬A ∨ ¬B ∨ D)`
15. `F(A,B,C) = A ∧ ¬B ∨ ¬A ∧ B ∨ B ∧ C`
16. `F(A,B,C,D) = (A ⊕ B) ∨ (C ∧ D) ∨ (A ∧ C)`
17. `F(A,B,C) = ¬(A → B) ∨ (B ∧ ¬C)`
18. `F(A,B,C,D) = A ∧ B ∨ ¬B ∧ C ∨ A ∧ ¬C ∧ D`
19. `F(A,B,C) = (A ∨ B) ∧ (¬A ∨ ¬B) ∧ (B → C)`
20. `F(A,B,C,D) = (A ∧ B) → (C ∨ D)`
21. `F(A,B,C) = A ⊕ (B ∧ C) ∨ A ∧ B`
22. `F(A,B,C,D) = (A ∨ ¬B) ∧ (C ∨ D) ∧ (B ∨ ¬C)`
23. `F(A,B,C) = ¬A ∧ B ∨ A ∧ ¬B ∨ A ∧ C`
24. `F(A,B,C,D) = A ∧ (B ⊕ C) ∨ ¬A ∧ (C ∧ D)`
25. `F(A,B,C) = (A → B) ∧ (B → C) ∨ A ∧ ¬C`
26. `F(A,B,C,D) = (A ∨ B) ∧ (¬B ∨ C) ∧ (A ∨ ¬C ∨ D)`
27. `F(A,B,C) = ¬(A ∧ B ∧ C) ∨ (A ⊕ B)`
28. `F(A,B,C,D) = A ∧ B ∨ ¬A ∧ ¬B ∨ B ∧ C ∧ ¬D`
29. `F(A,B,C) = (A ∨ ¬B) ∧ (A ∨ C) ∧ (¬B ∨ ¬C)`
30. `F(A,B,C,D) = (A ⊕ B) ∧ (C ∨ D) ∨ A ∧ ¬C`