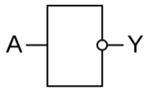
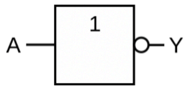
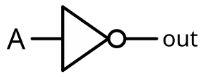
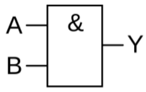
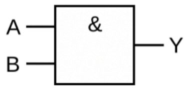
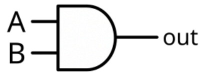
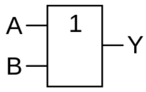
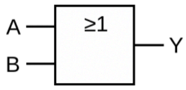
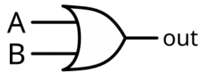
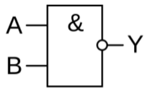
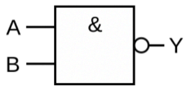
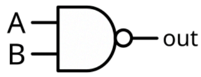
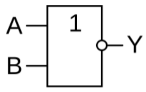
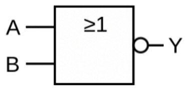
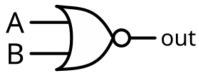
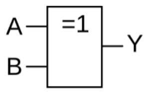
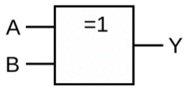
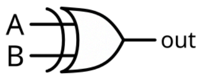
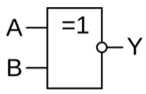
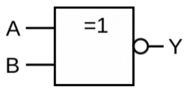
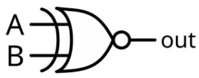
Информатика

Лабораторная работа 3. Логические основы информатики

Для описания логики функционирования аппаратных и программных средств ЭВМ используется алгебра логики или, как ее часто называют, булева алгебра (по имени основоположника этого раздела математики – Дж. Буля).
Алгебра логики – это раздел математической логики, который изучает логические высказывания. Логические высказывания рассматриваются со стороны их логических значений – истинности или ложности – и логических операций над ними.
Логические основы ВМ – это схемная реализация процесса обработки информации в ЭВМ на основе логических функций.

Целью разработки алгебры логики было создание приёмов решения традиционных логических задач алгебраическими методами.
Алгебра логики оперирует логическими высказываниями – это любое предложение, в котором содержится смысл утверждения, т.е. истинности (англ. TRUE), или отрицания, т.е. ложности (англ. FALSE). Одно и то же высказывание не может быть одновременно истинным и ложным или одновременно не истинным и не ложным. Отдельные высказывания обозначают заглавными буквами латинского алфавита (A, B, C и т.д.) Если высказывание A истинно (англ. TRUE), то записывают A = 1. Если высказывание B ложно (англ. FALSE), то записывают B = 0.

Логический вентиль	Условные графические обозначения ГОСТ 2.743-91	IEC 60617-12 : 1997	US ANSI 91-1984	Функция, запись	Таблица истинности
НЕ (англ. *NOT gate*)				Отрицание $Y=\overline{A}$ $Y=\neg A$ $Y=\tilde{A}$	A Y 0 1 1 0
И (англ. *AND gate*)				Конъюнкция $Y=A\wedge B$ $Y=A\cdot B$ $Y=A\&B$ $Y=AB$	A B Y 0 0 0 0 1 0 1 0 0 1 1 1
ИЛИ (англ. *OR gate*)				Дизъюнкция $Y=A\vee B$ $\boldsymbol{Y=A+B}$	A B Y 0 0 0 0 1 1 1 0 1 1 1 1
НЕ И (И-НЕ) (англ. *NAND gate*) Элемент Шеффера				$Y=\overline{A\wedge B}$ $Y=A\overline{\wedge}B$ $Y=\overline{A\cdot B}$ $Y=\overline{AB}$ $Y=A\vert B$	A B Y 0 0 1 0 1 1 1 0 1 1 1 0
НЕ ИЛИ (ИЛИ-НЕ) (англ. *NOR gate*) Элемент Пирса				$Y=\overline{A\vee B}$ $Y=A\overline{\vee}B$ $Y=\overline{A+B}$ $\boldsymbol{Y=A-B}$	A B Y 0 0 1 0 1 0 1 0 0 1 1 0
Исключающее ИЛИ (англ. *XOR gate*) сложение по модулю 2				Строгая дизъюнкция $Y=A\underline{\vee}B$ $Y=A\oplus B$	A B Y 0 0 0 0 1 1 1 0 1 1 1 0
Исключающее ИЛИ с инверсией (англ. *XNOR gate*) равнозначность				Эквиваленция $Y=\overline{A\underline{\vee}B}$ $Y=A\overline{\underline{\vee}}B$ $Y=\overline{A\oplus B}$ $Y=A\odot B$	A B Y 0 0 1 0 1 0 1 0 0 1 1 1

Аудиторные задачи с примерами:

Пример: Найти значение логического выражения $A\ V\ B\ \&\ \overline{C}$ при A=0 (False), B=1 (True), C=0(False). Решение: подставим значения переменных в выражение и вычислим его согласно приоритету выполнения операций: $A\ V\ B\ \&\ \overline{C} = 0\ V\ 1\&\overline{0}\ = 0\ V\ 1\&1 = 0\ V\ 1 = 1$ (True). Ответ: заданное логическое выражение принимает значение True.

1. **Найти значение логического выражения**

а)

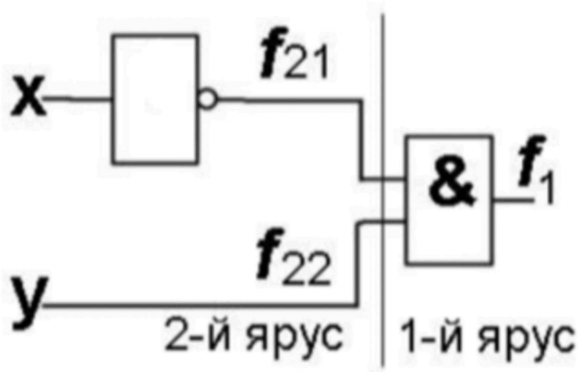
$A\ \&\ B\ V\ \overline{C}$ при A = False, B = True, C = False.

б)

$\overline{A \vee B}\ \&\ C$ при A = False, B = False, C = True.

2. **Найдите булеву функцию логической схемы и составьте таблицу истинности для логической схемы.**

Пример:



Решение. Разбиваем логическую схему на ярусы. Запишем все функции, начиная с 1-го яруса:

$f_1 = f_{21} \wedge f_{22}$
$f_{21} = \bar{x}$
$f_{22} = y$

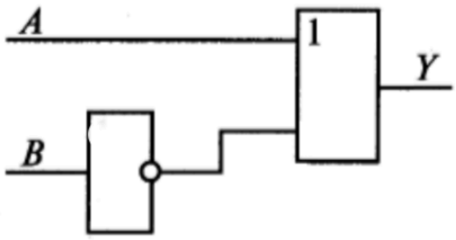
Получаем функцию, которую реализует на выходе логическая схема:

$f_1 = \bar{x} \wedge y$.

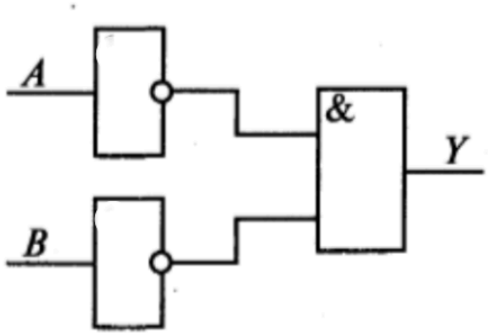
Таблица истинности для данной логической схемы:

x	y	f_{21}	f_{22}	f
1	1	0	1	0
1	0	0	0	0
0	1	1	1	1
0	0	1	0	0

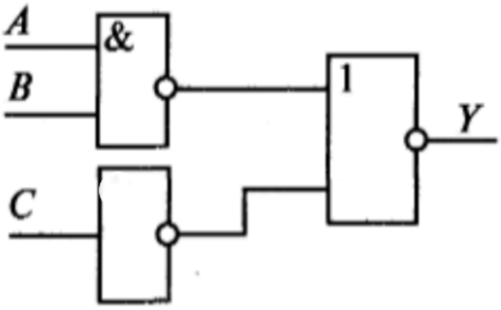
а)



б)



в)



3.

По заданным логическим выражениям составить логические схемы и заполнить таблицы истинности.

$$\overline{X \,\&\, \overline{Y}} \vee Z = F;$$

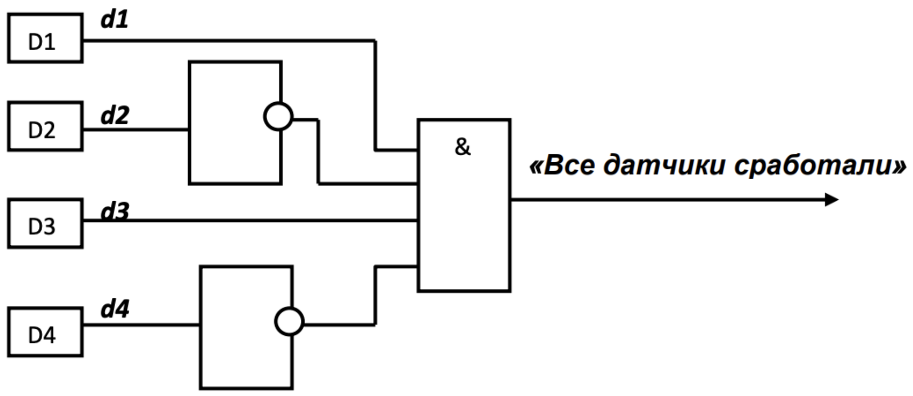
$$X \vee \overline{\overline{Y} \,\&\, Z} = F;$$

$$\overline{\overline{X} \vee Y \,\&\, \overline{Z}} = F.$$

4. **Датчики системы срабатывают следующим образом: датчики D1 и D3 срабатывают из 0 в 1, а датчики D2 и D4 – из 1 в 0. Разработать схему, которая «отследит» срабатывание всех четырех датчиков одновременно и выдаст сигнал логической 1. Подумайте сами, решение на следующей странице.**

Если бы все четыре датчика срабатывали одинаково, то достаточно было бы просто выполнить операцию логического умножения, использовав 4 элемента И. На его выходе появляется логическая 1 тогда, когда на всех его входах - логическая 1.

Однако датчики D2 и D4 выдают при срабатывании логические 0. Изменим логику работы датчиков таким образом, чтобы при срабатывании появлялись не логические 0, а логические 1. Для этого сигналы с датчиков достаточно проинвертировать.



5. **Обозначим датчики дверей D1, D2,D3,D4. Выработать сигнал логической 1 в случае открывания одной из четырех дверей автомобиля. Состояние двери определяют датчики следующим образом: если дверь закрыта, то датчик выдает логический 0, если открыта – логическую 1.**