Оглавление

Лабораторная работа № 1

ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ И СХЕМЫ

ЦЕЛЬ РАБОТЫ: ознакомление с основными характеристиками логических элементов и основами синтеза логических схем.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ И РАСЧЕТНЫЕ ФОРМУЛЫ

Устройства, выполняющие функции *алгебры логики*, называют логическими, либо цифровыми, а также классифицируют в соответствии с разными характерными показателями.

В соответствии с характером вводимых и выводимых данных, логические устройства делятся на устройства последовательного, параллельного, а также смешанного действия, а согласно диаграммному заключению, а также способу взаимосвязи среди входными, а также выходными непостоянными вместе с учетом их перемены, согласно тактам, деятельность — *на комбинационные и последовательностные*.

В *комбинационных* устройствах значения (0 либо 1) сигналов на выходах в любой момент времени целиком создаются смыслами (комбинацией, набором) работающих в этот момент числовых сигналов на входе. В последовательностных устройствах значения сигналов на выходах в n-такте получаются не только значениями входных сигналов в этом такте, но также зависят от внутреннего состояния устройства, которые привелись из-за воздействия входных сигналов в предшествующие такты.

Эта работа посвящена изучению простых комбинационных логических устройств, выполняющих логические функции сложения, умножения, а также отрицания.

1. ОСНОВНЫЕ ЭЛЕМЕНТЫ АЛГЕБРЫ ЛОГИКИ

Анализ комбинационных устройств удобно производить пользуясь возможностями алгебры логики, работающей только с 2-мя значениями: истина (логическая 1) и ложь (логический 0). Поэтому, функции, воспроизводящие сведения, получают в любой момент времени исключительно значения 1 или 0. Такие функции называются *логичными*, а сигналы (входящие, а также выходящие переменные) — *двоичными* (бинарными).

Элементы схемы, при помощи которых реализовывается изменение поступающих в их входы двоичных сигналов, а также непосредственное исполнение задуманных логических операций, называются логическими устройствами.

В общем случае логическое устройство может иметь *n* входов, а также m выходов. Анализируя входящие сигналы $x_1, x_2, \ldots, x_n$ как аргументы, можно соответствующие им выходные сигналы представить как функцию

$$y_i = f\left(x_1, x_2, \ldots, x_n\right),$$

используя средства алгебры логики.

Функции алгебры логики (ФАЛ), в определённых ситуациях называются переключательными функциями, обычно представляются в алгебраической форме (в качестве математических выражений), например, или как таблица истинности (комбинационных таблиц).

$$y_i = \left(x_0 \wedge x_1\right) \vee \left(x_1 \wedge x_2\right)$$

Таблица истинности содержит все композиции (комплекты) бинарных смыслов входных данных вместе с соответствующими им бинарными смыслами выходных данных; каждому набору входных данных соответствует определённое значение выходящего сигнала — результат выполнения логической функции *y*. Максимальное число возможных разных наборов (строчек) зависит от количества входных переменных *n* и равняется 2^n.

В двоичной алгебре существуют три основные функции: конъюнкция, дизъюнкция, отрицание. Остальные функции являются производными от вышеперечисленных.

Главные логические операции представляют собой следующие элементарные преобразования двоичных сигналов:

Логическое сложение, или дизъюнкция, представлено символом «$\vee$» (или «+»), также используется название операция ИЛИ. Количество аргументов данной функции (слагаемых *x*) может быть разным. Такая операция, представленная функцией двух переменных x_1 и x_2, реализуется в качестве логической формулы:

$$y = x_1 \vee x_2 = x_1 + x_2$$

Данная функция принимает значение истина (логическая 1), в случае если истина хотя бы одна из переменных. В случае если переменные x_1 и x_2 обе принимают значение лож (логический 0), функция также будет ложной.

Логическое умножение, или конъюнкция, представлено символом «$\wedge$» (или «·»), также используется название операция И. Количество аргументов данной функции (множителей x) может быть разным. Такая операция, представленная функцией двух переменных x_1 и x_2, реализуется в качестве логической формулы:

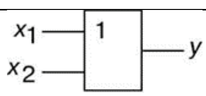
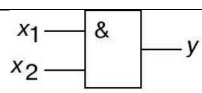
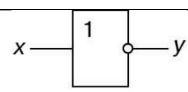
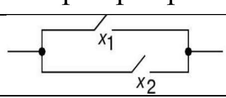
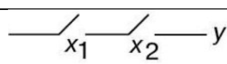
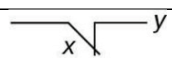
$$y = x_1 \wedge x_2 = x_1 \bullet x_2 = x_1 x_2$$

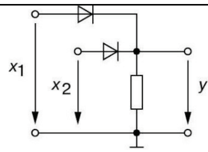
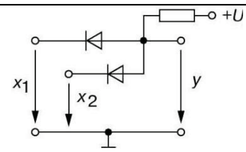
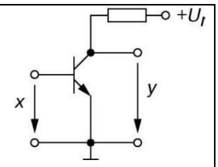
Данная функция принимает значение истина (логическая 1), в случае если истины обе переменные. В случае если одна из переменных x_1 или x_2 примет значение лож (логический 0), функция также будет ложной. Логическое отрицание, либо инверсия, представлено символом черточки над переменной, также используется название операция НЕ. Такая операция, представленная функцией одной переменной x, реализуется в качестве логической формулы $y = x$.

Данная функция принимает значение истина (логическая 1), в случае если переменная принимает значение лож (логический 0), и наоборот.

Таблица истинности функция логического сложения, умножения и отрицания представлена в таблице 1.1

Таблица 1.1. Формы отображения основных логических функций

<table>
<tr><td>Название функции</td><td>Дизъюнкция</td><td>Конъюнкция</td><td>Инверсия</td></tr>
<tr><td>Символ, обозначающий функцию</td><td>$\vee$ или +</td><td>$\wedge$ или •</td><td>$\neg x$</td></tr>
<tr><td>Буквенное обозначение</td><td>ИЛИ</td><td>И</td><td>НЕ</td></tr>
<tr><td>Условно графическое обозначение</td><td></td><td></td><td></td></tr>
<tr><td>Запись при помощи формулы</td><td>$y = x_1 \vee x_2 = x_1 + x_2$</td><td>$y = x_1 \wedge x_2 = x_1 \bullet x_2 = x_1 x_2$</td><td>$y = x$</td></tr>
<tr><td>Таблица истинности</td><td>x_1 x_2 y
0 0 0
0 1 1
1 0 1
1 1 1</td><td>x_1 x_2 y
0 0 0
0 1 0
1 0 0
1 1 1</td><td>x y
0 1
1 0</td></tr>
<tr><td>Контактная</td><td></td><td></td><td></td></tr>
</table>

Схемотехниче- ская			

Главные логические операции ИЛИ, И, НЕ реализуются при помощи логических элементов ИЛИ (*дизъюнктор*), И (*конъюнктор*) и НЕ (*инвертор*). Таким образом можно создать комбинационное устройство любого уровня

сложности, таким образом функции $y = x_1 + x_2$, $y = x_1x_2$, $y = x$ располагают функциональной полнотой и представляют собой функционально полный набор.

Например, разберём функцию неравнозначности y от двух переменных x_1 и x_2, которая принимает значение истина (логическая 1), если $x_1 \neq x_2$ и значение ложь (логический 0), если $y \equiv x_1x_2 + x_1x_2$.

$x_1 = x_2 = 0$ или если $x_1 = x_2 = 1$, таким образом

Функцию неравнозначности обычно именуют *суммированием по модулю 2* и записывают как $y = x_1 \oplus x_2$

Способы реализации контактной и простейшей схемы дизъюнктора, конъюнктора и инвертора показаны в предпоследней и последней строках таблицы 1.1.

2. БАЗОВЫЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

Важное место в цифровой электронике отведено универсальным (базовым) логическим элементам, из которых можно сформировать высокофункционально полный набор, используя который можно реализовать сочетание устройств всех уровней сложности. При интегральной технологии производства утилитарность создания одного базового компонента обладает решающим значением. Из-за этого базовые логические устройства являются основой для большинства цифровых ИМС.

Многоцелевыми логическими устройствами также считаются 2 типа базовых компонентов:

• *функцию Пирса*, представлена символом вертикальной стрелки, направленной вниз $\downarrow$ (стрелка Пирса) и характеризующая операцию ИЛИ-НЕ.

Для элементарной функции от двух переменных x_1 и x_2 функция y принимает значение истина (логическая 1) только если:

$$x_1 = x_2 = 0; y = x_1 \downarrow x_2 = \overline{x_1 + x_2}$$

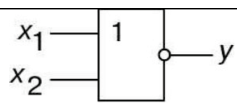
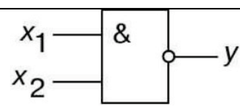
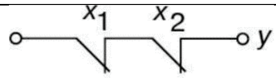
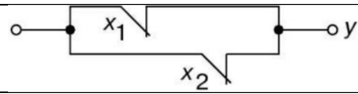
• *функцию Шеффера*, представлена символом вертикальной черты | (штрих Шеффера) и характеризующая операцию И-НЕ.

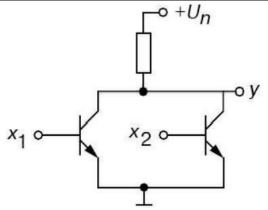
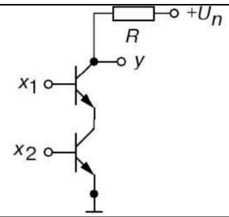
Для элементарной функции от двух переменных x_1 и x_2 функция y принимает значение ложь (логический 0) только если:

$$x_1 = x_2 = 1$$

$$y = x_1 \mid x_2 = \overline{x_1 x_2}$$

Таблица 1.2. Формы отображения базовых логических функций

Название функции	Функция Пирса	Функция Шеффера
Символ, обозначающий функцию	$\downarrow$	$\mid$
Буквенное обозначение	ИЛИ-НЕ	И-НЕ
Условно графическое обозначение		
Запись при помощи формулы	$y = x_1 \downarrow x_2$	$y = x_1 \mid x_2 = \overline{x_1 x_2}$
Таблица истинности	x_1 x_2 y 0 0 1 0 1 0 1 0 0 1 1 0	x_1 x_2 y 0 0 1 0 1 1 1 0 1 1 1 0
Контактная		

<table>
<tr>
<td>Схемотехническая</td>
<td>
</td>
<td>
</td>
</tr>
</table>

В случае равенства значений аргументов данные функции представляют собой операцию инверсии. Основные свойства функции Шеффера и функции Пирса представлены в таблице 1.2.

В последней строке таблицы 1.2 изображены примеры построения двухвходовой схемы ИЛИ-НЕ, где к нагрузочному резистору R подсоединены коллекторы 2-ух единовременно подключённых биполярных транзисторов вида *p-n-p*, эмиттеры которых заземлены. В схеме И-НЕ поочередно подключены 2 биполярных транзистора (эмиттер нательного транзистора заземлён), а также в схеме имеется нагрузочный резистор R.

3. ПРЕДСТАВЛЕНИЕ ЛОГИЧЕСКИХ ФУНКЦИЙ МАТЕМАТИЧЕСКИМИ ВЫРАЖЕНИЯМИ

Наиболее распространённым способом описания логических выражений является запись в виде таблицы. Таблица истинности позволяет полноценно и однозначно определить абсолютно все существующие логические связи.

В табличной (таблица 1.3) форме логические функций принято фиксировать в одной из канонических форм: совершенной дизъюнктивной нормальной форме (СДНФ) либо совершенной конъюнктивной нормальной форме (СКНФ).

Таблица 1.3. Представлении логических функций

№	*a*	*b*	*c*	*y*
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Математическая реализация логической функции в СДНФ выводится из таблицы истинности таким методом: все наборы переменных, при которых функция имеет значение истина (логическая 1), вносятся в логические произведения переменных, при этом переменные, принимающие значение лож (логический 0), записывают под знаком инверсии. Эти произведения, называемые конституентами единицы, либо минтермами, суммируют.

Составим логическую функцию y из трёх переменных a, b, c представленной в таблице 1.3, используя СДНФ:

$$y(a,b,c) = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$$

Совершенной конъюнктивной нормальной форме называется логическое произведение логических сумм, в каждую из которых переменная или её отрицание входят единожды.

При этом для всех наборов переменных входящих в таблицу истинности, в которых функция y принимает значение ложь (логический 0), входят в логическую сумму, переменные, принимающие значение истина (логическая 1), входят в функцию под знаком логического отрицания. Такие логические суммы называются конституентами нуля, либо макстермами, объединяются при помощи операции логического умножения.

Для функции (таблицы 1.3) СКНФ:

$$y(a,b,c) = (\bar{a}+b+c)(a+\bar{b}+c)(a+b+\bar{c})(a+b+c)$$

4. ПРЕОБРАЗОВАНИЕ ЛОГИЧЕСКОЙ ФУНКЦИИВ ЛОГИЧЕСКУЮ СХЕМУ

Для получения логической схемы (рисунок 1) необходимо логические элементы, предназначенные для проведения логических операций, размещать, начиная от входа, в порядке их следования в булевом выражении.

Построим логическое устройство, выполняющее логическую функцию трёх переменных:

$$y = (a+b+c)(a+b+\bar{c})(\bar{a}+b+c)(\bar{a}+\bar{b}+c)$$

Слева находятся входы, *a*, *b* и *c* с ветвями, предназначенными для трёх инверторов, далее четыре элемента операции ИЛИ и, наконец, элемент операции И, расположенный на выходе (рис. 1.1).

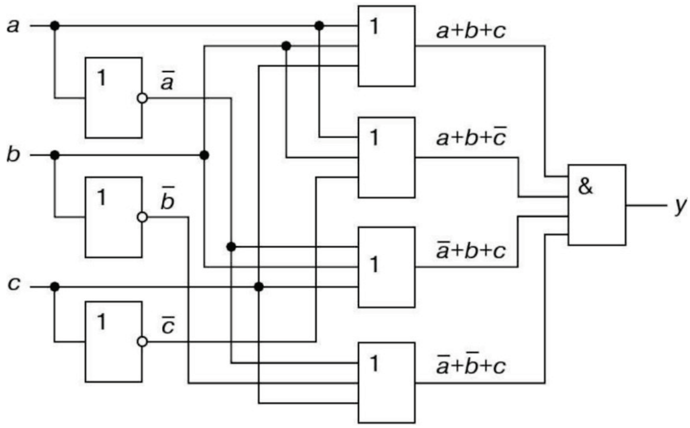


Рис. 1.1. Логическая схема

Таким образом, любую логическую функцию можно составить в соответствии с выражениями, полученными при нахождении СДНФ или СКНФ. Но полученная согласно этому методу схема, зачастую, не является оптимальной с точки зрения её фактической реализации, так как она имеет большие габариты, состоит из большого числа логических элементов, а также из-за возникающих проблем в обеспечении её высокой надежностью.

Алгебра логики позволяет упрощать формулы, описывающие сложные выражения. Это помотает в итоге получить наиболее выгодную структуру

любого логического устройства. Принято считать подходящей структурой так собранные логические приборы, количество которых для данного выражения минимально.

УЧЕБНЫЕ ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ИХ ВЫПОЛНЕНИЮ

Задание 1. Запустить комплекс лабораторных работ **Micro-Cap 12**. Создать на рабочем поле среды **Micro-Cap 12** схему для проверки *основных и базовых логических элементов* (рисунок 1.2) и установить в диалоговых окнах компонентов их параметры или режимы работы. **Скопировать** схему (рисунок 1.2) в отчёт. Для включения отображение номеров узлов схемы нужно нажать кнопку , находящуюся на основной панели.

Схема (рисунок 1.2) состоит из основных двоичных функций [**OR** (ИЛИ), **AND** (И) и **NOT** (НЕ)] и универсальных (базовых) [**NAND** (И-НЕ) и **XOR** (ИЛИ-НЕ)], собрана при помощи логических элементов, которые находятся в библиотеке **Digital Primitives/ Standart Gates**. В схеме используются ключи ***1, 2, ..., 9***, пробники ***X1, X2, ..., X5***, которые находятся в библиотеке **Animation/ Animated Digital LED** с пороговыми напряжениями 5В, генератор прямоугольных сигналов **E1** с амплитудой $E = 5B$, длительностьюимпульса $t_u = 0{,}16c$ и периодом $T = 4c$.

Для простоты измерения сигналов используем «Transient Analysis». Для этого во вкладке «Analysis» выбираем «Transient», указываем номера узлов, связанных с пробниками ***X1, X2, ..., X5*** (d(№ узла)) (рисунок 1.2), нажимаем кнопку «Run». Используя ключи ***1, 2, ..., 9***, **сформировать** все существующие наборы аргументов x_1 и x_2 (00, 01, 10, 11) на входах дизъюнктора (**OR**), конъюнктора (**AND**), штриха Шеффера (**NAND**) и стрелки Пирса (**NOR**) и записать значения выходных логических функций V_k (0 или 1) в таблицу 1.4.

Отметим, что в случае, когда ключ замкнут, то на вход элемента будет подано значение истина (логическая 1) (положительное напряжение = 5 В), а в случае разомкнутого ключа будет подано значение лож (логический 0). Так как инвертор (**NOT**) обладает одним входом, то для получения двух значений входного сигнала (истина (логическая 1) или лож (логический 0)) необходим один ключ под номером 5.

Значения функций изучаемых элементов можно проверять с помощью пробников ***X1, X2, ..., X5***: если выходной сигнал элемента принимает значение истина (логическая 1), то подсоединённый на этом выходе элемента пробник будет светится. Поэтому, при изменении ключей схемы (рисунок 1.2) функции элементов **OR**, **AND** и **NOR** примут значения истина (логическая 1)

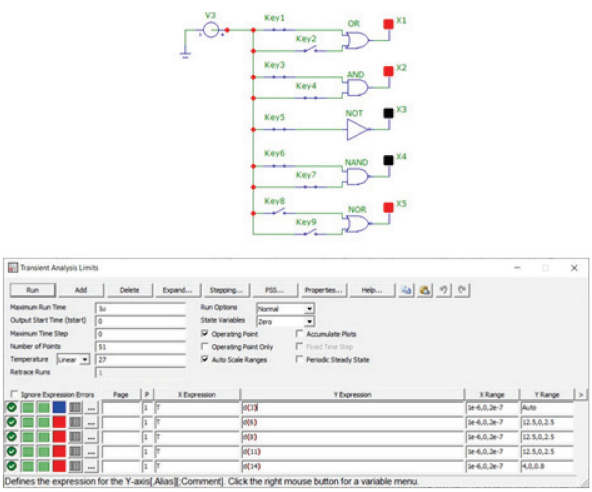


Рис. 1.2. Схема для испытания основных и базовых логических элементов

Таблица 1.4. Значения выходных логических функций V_k (0 или 1)

Дизъюнктор [ИЛИ (**OR**)]			Конъюнктор [И (**AND**)]			Инвертор [НЕ (**NOT**)]		Штрих Шеффера [И-НЕ (**NAND**)]			Стрелка Пирса [ИЛИ-НЕ (**NOR**)]		
x_1	x_2	y	x_1	x_2	y	x		x_1	x_2		x_1	x_2	y
0	0		0	0		0		0	0		0	0	
0	1		0	1				0	1		0	1	
1	0		1	0		1		1	0		1	0	
1	1		1	1				1	1		1	1	

Задание 2. «Переместить» из библиотеки **Digital Primitives/ Standard Gates** на рабочее поле среды **Micro-Cap 12** необходимые логические элементы и составить схему для осуществления логической функции y с 3 аргументами a, b и c, данной в табл. 1.5. **Скопировать** полученную логическую схему в отчет по лабораторной работе

Таблица 1.5. Данные для задания 2

Вариант	Логическая функция
1, 6, 11, 16, 21, 26	$y=(ab+c)(\overline{a}+b+\overline{c})(\overline{a}+b+\overline{c})$
2, 7, 12, 17, 22, 27	$y=(\overline{a}+\overline{b}+\overline{c})(a+b\overline{c})(a+\overline{b}+c)$
3, 8, 13, 18, 23, 28	$y=(a+bc)(\overline{a}+\overline{b}c)(\overline{a}+b+\overline{c})$
4, 9, 14, 19, 24, 1	$y=(\overline{a}\overline{b}+c)(\overline{a}+b+c)(ab+\overline{c})$
5, 10, 15, 20, 25, 30	$y=(\overline{a}+b\overline{c})(a+\overline{b}+c)(ab+c)$

Для примера составим схему для получения значения логической функции вида:

$$y=(ab+c)(\overline{a}+\overline{b}+\overline{c})(a+b+c)$$

Проанализировав функцию очевидно, что для получения логической схемы потребуется три инвертора, три дизъюнктора, при этом необходим один дизъюнктор с двумя входами, и два с тремя, а также необходимы два конъюнктора, при этом один вместе с двумя входами, а второй с тремя.

«Перенесём» на рабочее поле среды **Micro-Cap 12** требуемые модели логических элементов из библиотеки **Digital Primitives/ Standard Gates**, выставляя их, начиная со входа:

- три инвертора **NOT** (**NOT1**, **NOT2** и **NOT3**) с целью нахождения значения функций $\overline{a}, \overline{b}, \overline{c}$;
- конъюнктор **AND1** с двумя входами с целью нахождения функции ab;
- три дизъюнктора: **OR2** с целью нахождения функции $y_1 = a + b + c$, **OR3** с целью нахождения функции $y_2 = a + b + c$ и **OR1**, с целью нахождения функции $y_3 = ab + c$, расположив их друг под другом (рисунок 1.3)

С целью нахождения значения функции логического умножения $y = y_1 y_2 y_3$ включим в схему конъюнктор **AND2** имеющий три входа, к выходу которого есть возможность подключения логического пробника ***X1*** (уровень высокого напряжения 5В) для обозначения появления логической единицы на выходе схемы. «Перенесём» из соответствующих библиотек на рабочее поле источник прямоугольных сигналов ***E*1** и ключ **Key1**, разместив их на входе схемы.

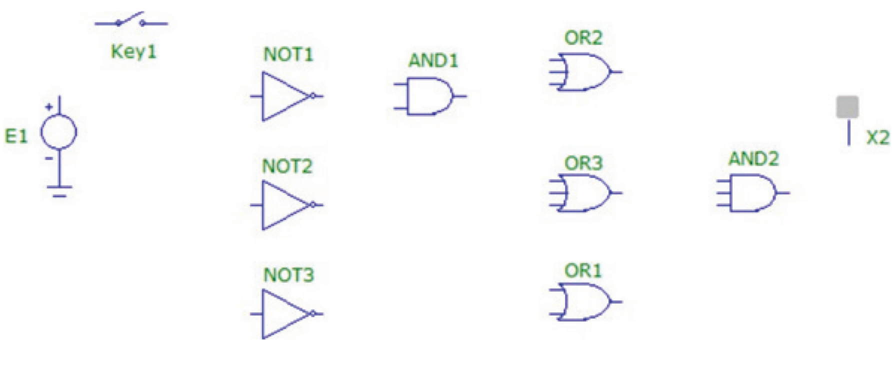


Рис. 1.3. Рабочее поле, исходные данные

Соединив «проводниками» входы и выходы элементов в соответствии с логическими выражениями, соответствующих заданной функции и записав в отчете ожидаемые результаты выполнения операций на выходах элементов (рис. 1.4), приступим к моделированию, открыв вкладку **Analysis/ Transient**.

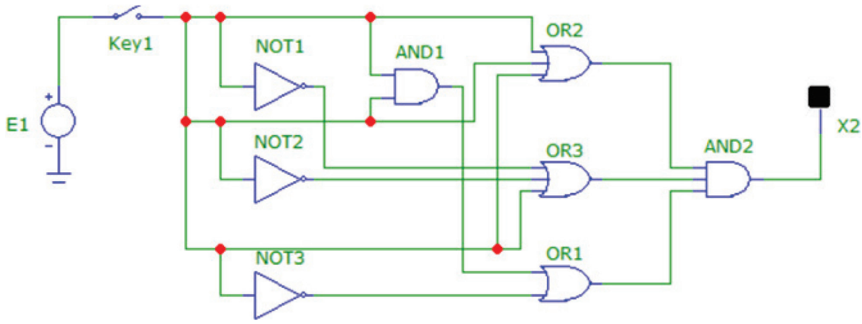


Рис. 1.4. Результаты выполнения операций на выходах элементов

Замкнём ключ **Key1**. Когда соединение элементов выполнено корректно, пробник ***X2*** засветится (на выходе будет получено значение истина (логическая 1)). Если замкнуть ключ **Key1** пробник погаснет. После моделирования закроем вкладку **Transient Analysis (Crtl+F4)**.

ТЕСТОВЫЕ ЗАДАНИЯ К РАБОТЕ 1

1. Укажите **признаки**, описывающие основные логические элементы:

на входах логических элементов аналоговые сигналы, а на выходах – цифровые;
операции логического сложения, логического умножения и инверсия не составляют функционально полный набор;
при помощи основных логических операций И, ИЛИ и НЕ, можно аналитически получить логическую функцию любой сложности;
минимальный логический базис представлен операциями ИЛИ и НЕ или И и НЕ;
входные и выходные сигналы логических элементов могут иметь только два значения
значения истина (логическая 1) и лож (логический 0);
операция логического сложения совпадает с операцией обычного сложения.

2. Выберите **выражение** логической функции двух переменных x_1 и x_2 получаемой при помощи элемента «стрелка Пирса»:

$y = x_1\overline{x_2} \mid \overline{x_1}x_2$	$y = \overline{x_1x_2}$	$y = x_1 \mid x_2$
$y = x_1 \oplus x_2$	$y = x_1 + x_2$	$y = x_1x_2$

3. Выберите **выражение** логической функции двух переменных x_1 и x_2, получаемой при помощи элемента «штрих Шеффера»:

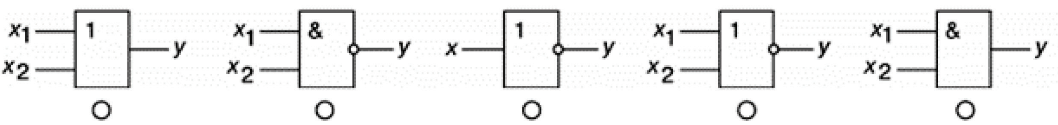
$y = \overline{x_1}x_2 + x_1\overline{x_2}$	$y = \overline{x_1x_2}$	$y = x_1 \oplus x_2$
$y = \overline{x_1 + x_2}$	$y = x_1 + x_2$	$y = x_1x_2$

4. Выберите представление логической функции трёх переменных a, b и c, указанной в совершенной дизъюнктивной нормальной форме (СДНФ):

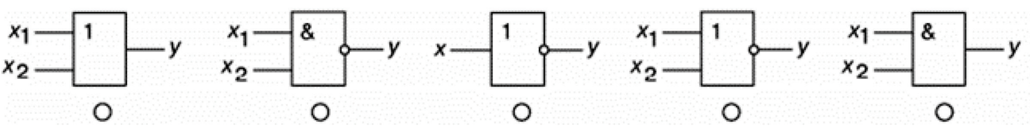
$$y(a,b,c) = abc + \overline{a}bc + a\overline{b}c + ab\overline{c}$$

$y(a,b,c) = (a+b+c)(\overline{a}+b+c)(a+\overline{b}+c)(a+b+\overline{c})$
$y(a,b,c) = (ab+\overline{c}+abc)(a\overline{b}c+\overline{a}\overline{b}+\overline{c}a)$

5. Выберите **элемент** ИЛИ-НЕ:



6. Выберите **элемент** И:



7. Выберите значение **функции**

$$y = \left(\overline{a}b + \overline{c}\right)\left(a + \overline{b}\right),$$

при $a = b = c = 1$

1	0

Лабораторная работа № 2

ПРЕОБРАЗОВАТЕЛИ КОДОВ

Цель работы

Познакомиться с главными характеристиками и испытание интегральных преобразователей кодов (дешифратора, шифратора, демультиплексора и мультиплексора).

Теоретические сведения и расчётные формулы

Кодирование – это процесс преобразования информации в форму, удобную для обработки, хранения и передачи. В технических приложениях, таких как журналы кодовых знаков или программный код, используется двоичная система с символами нуля и единицы. Преобразователи кодов позволяют изменять одну бинарную комбинацию в другую, например, преобразовывать двоично-десятичный код в семисегментный шифр указателя.

Преобразователи кодов соединяют входные и выходные коды с помощью логических функций или таблиц переключений. Существует множество видов преобразователей кодов, но наиболее распространенные используются в цифровой технике.

1. Дешифратор

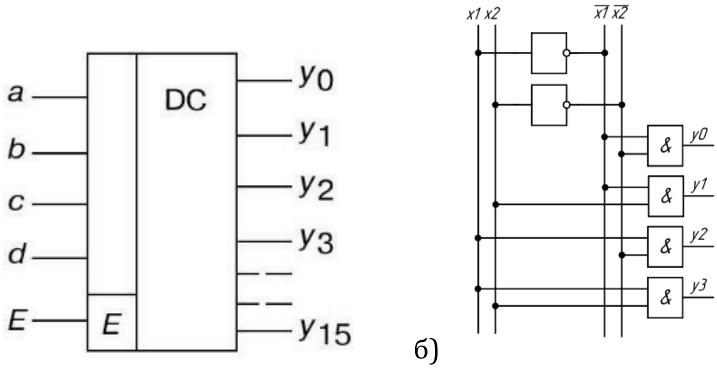
Дешифратор, известный как декодер, является комбинационной схемой, которая имеет n входов и m = 2^n выходов (где m > n). Его основная функция состоит в том, чтобы преобразовать кодовый набор из 0 и 1 (или слово) длиной n в унитарный код на выходах. Только один выход дешифратора будет иметь логическую единицу, соответствующую входному двоичному коду, остальные выходы будут иметь нулевые сигналы.

При использовании декодирующего устройства, происходит выбор цифрового устройства, закодированного в двоичном коде.

Любой дешифратор имеет свое устройство, как на схеме рисунка 2.1а. Данный дешифратор имеет на вход 4 шин значений, и, соответственно, на входе он будет иметь 16 логических адресов всех комбинаций

(4x16 — »4 в 16»),

от $n_0 = \neg n \neg n \neg n \neg n$ до $n_{15} = nnnn$, при $n = 4$, $m = 2^n = 16$.



а) б)

Рис. 2.1 – Дешифратор 4 в 16 и устройство дешифратора 2 в 4

На практике, чаще всего используются дешифраторы с минимальным количеством выходов, такие дешифраторы называются неполными (на вход поступает 4 переменных и из 16 возможных комбинаций используется только 10–12).

Выходы таких дешифраторов показывают конъюнкцию входов или их инверсий, пример на рисунке 2.2б.

Чаще всего декодер состоит из 2-ух компонентов, первый это сам дешифратор, а второй это компонент — управляющий вход Е. Он позволяет управлять работой всего компонента, при установленном значении 1 весь декодер работает, как и всегда, а при значении 0 на все его выходы подаются нулевые значения. Декодеры также часто используется с целью преобразования двоичного программного кода в десятичный.

2. Шифратор

Шифратор / Coder (CD), выполняет противоположную функцию по сравнению с декодером. На рисунке 2.2а показано относительное отображение шифратора 16х4 (шестнадцать в четыре) в схемах. Обычный кодировщик имеет n входов и m выходов (где m < n), и когда сигнал »один» поступает на один из входов (или несколько), на выходе шифратора появляется двоичный код номера активного входа. Количество входов и выходов данного шифратора связано соотношением $n = 2^n$.