Учреждение образования
«Гомельский государственный политехнический техникум»
Заочное отделение

Контрольная работа

по дисциплине

«Основы автоматики и микропроцессорной техники»

для группы ЗМЭ-41с

Специальность: **2-36 03 31**
«Монтаж и эксплуатация электрооборудования»

Вариант контрольной работы определяет номер учащегося по списку в учебном журнале

Список литературы, имеющейся в библиотеке ГГПТ

О.И. Головинский
Основы автоматики О.И. Головинский М. «В.ш.», 1987 г.

Б.А. Калабеков
Цифровые устройства и микропроцессорные системы – М. Горячая линия - Телеком, 2005

А.В. Кузин, М.А. Жаворонков
Микропроцессорная техника
Москва 2004

В.В. Стрыгин Л.С. Щарев
Основы вычислительной микропроцессорной техники и программирования – М. «В.ш.», 1989 г.

Задание 1 – Автоматические системы

Для заданной системы:
- составьте структурную схему;
- составьте алгоритм работы;
- выберите типы датчиков;
- выберите типы исполнительных устройств.

Таблица 1 – Варианты заданий

№ варианта	Название системы
0	Система автоматического регулирования частоты вращения электродвигателя
1	Автоматическая система поддержания температуры в помещении на одном уровне
2	Автоматической системы поддержания уровня воды в водонапорной башне
3	Система автоматического поддержания величины освещенности в помещении
4	Автоматическая система охранной сигнализации помещения
5	Система автоматического регулирования частоты вращения электродвигателя
6	Автоматическая система поддержания температуры в помещении на одном уровне
7	Автоматической системы поддержания уровня воды в водонапорной башне
8	Система автоматического поддержания величины освещенности в помещении
9	Автоматическая система охранной сигнализации помещения

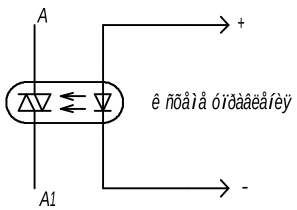
Задание 2 – Бесконтактные системы автоматики

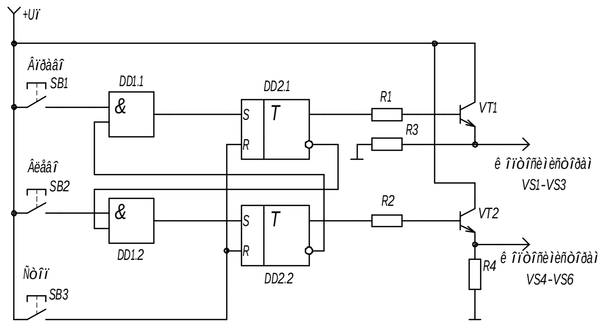
Замените релейно-контакторную цепь управления и силовую цепь на бесконтактную систему с использованием полупроводниковых приборов и устройств (оптосимисторов, оптотиристоров, триггеров, логических элементов, бесконтактных датчиков и т.п.)

Примеры построение схемы:

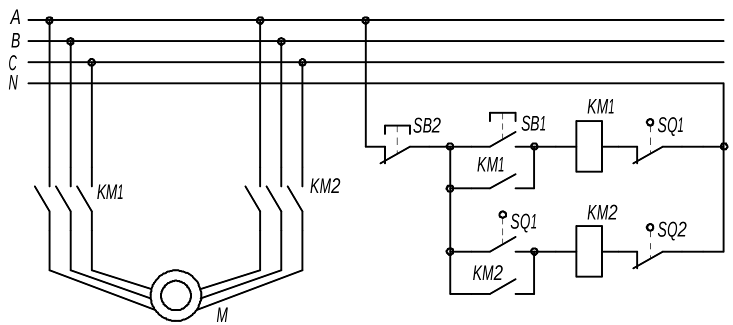
Силовые контакты электромагнитных контакторов заменяют тиристорами (симисторами) совместно с оптопарами (для гальванической развязки силовой цепи и схемы управления) или оптотиристорами (оптосимисторами).

Триггеры выполняют функцию самоблокировки кнопок управления. Логические элементы определяют алгоритм функционирования схемы.





Вариант **1 (7)**



На схеме представлена схема управления двухскоростным двигателем М. При нажатии кнопки **SB1** запитывается КМ**1** и замыкает свои контакты, подключая эктродвигатель к сети (скорость n_1). При срабатывании путевого выключателя **SQ1** выключается КМ**1** и включается КМ**2**, замыкание контактов которого обеспечивает переход двигателя на скорость n_2. При срабатывании **SQ2** или нажатии кнопки стоп **SB2** двигатель отключается.

Вариант **2 (8)**

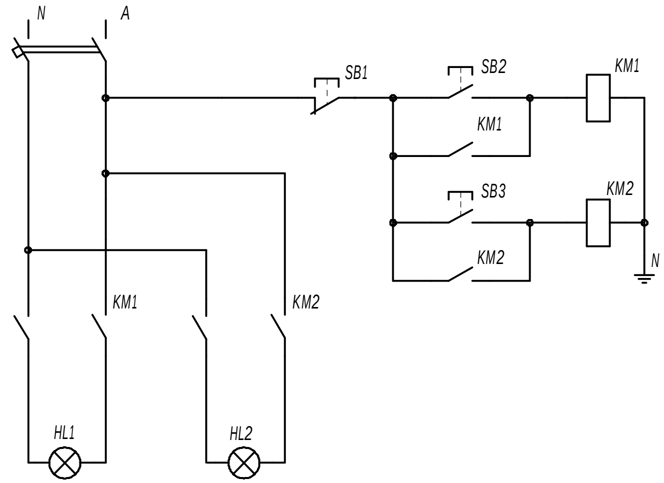


Схема дистанционного включения электроламп с общим отключением

Вариант **3 (9)**

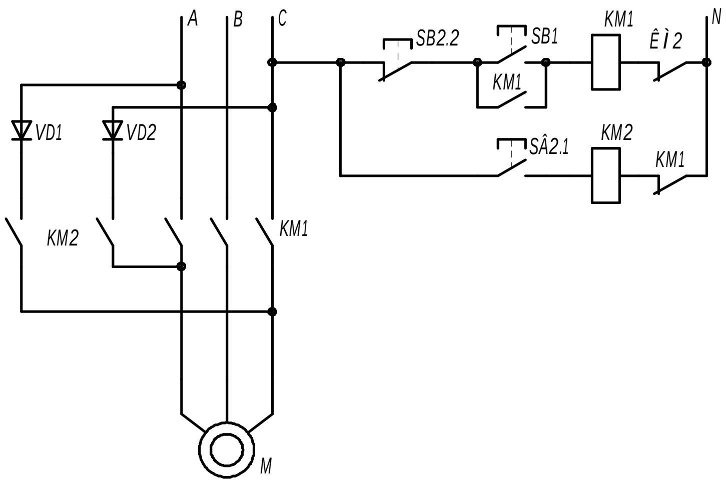


Схема обеспечивает пуск и динамическое торможение асинхронного двигателя М.

При нажатии **SB1** срабатывает контактор КМ**1**, обеспечивая пуск двигателя и самоблокировку кнопки.

При нажатии кнопки **SB2** контактор КМ**1** отключает двигатель от сети, а контактор КМ**2** обеспечивая протекание по обмотке статора постоянного тока – происходит процесс динамического торможения.

Вариант **4 (0)**

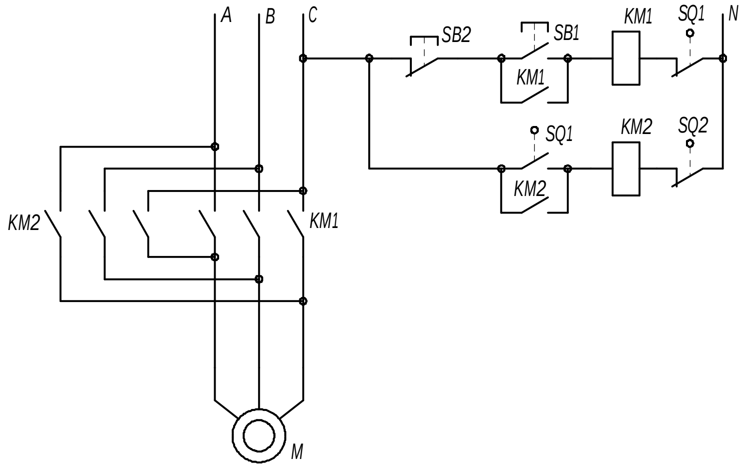


Схема обеспечивает пуск, реверс и остановку асинхронного двигателя М.

При нажатии **SB1** срабатывает контактор КМ**1**, обеспечивая пуск двигателя и самоблокировку кнопки. При достижении определённой точки срабатывает путевой выключатель **SQ1**, отключающий контактор КМ1 и включающий КМ**2**. Двигатель начинает вращаться в обратную сторону. При достижении начальной точки срабатывает **SQ2** и двигатель останавливается.

Вариант **5 (6)**

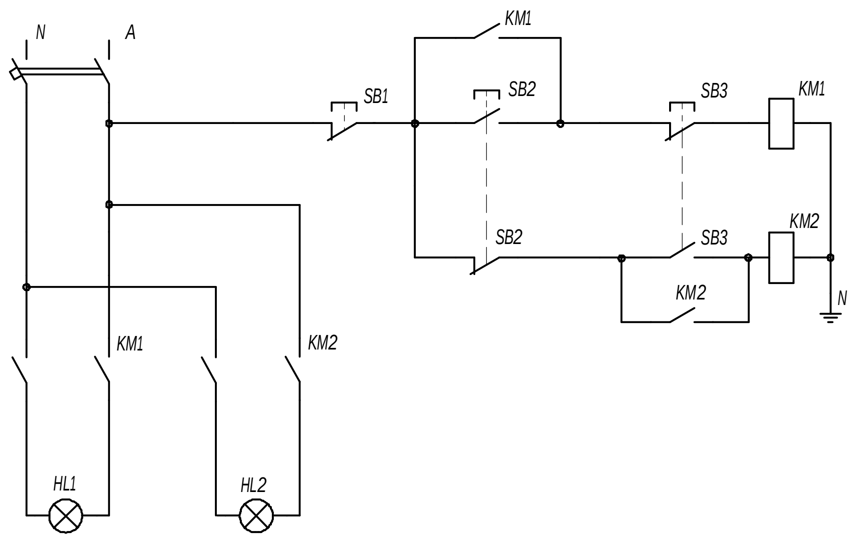


Схема дистанционного переключения электроламп с общим отключением

Задание 3 – Арифметические основы ЭВМ

Произведите необходимые операции над числами в различных системах счисления:

- переведите десятичное число в иную систему счисления;
- проверьте полученные результаты (выполните обратный перевод);
- сложите числа в указанной системе счисления.

Таблица **2** – Варианты заданий

№ варианта	Перевод числа		Сложение чисел	
	Десятичное число	Иная система счисления	Слагаемые	Система счисления
0	**1000**	двенадцатеричная	+**2121** **1212**	троичная
1	**487**	пятнадцатеричная	+**1234** **211**	пятеричная
2	**4000**	четырнадцатеричная	+**2345** **364**	семеричная
3	**285**	четверичная	+**23A6** **975**	одинадцатеричная
4	**5000**	семнадцатеричная	+**7775** **887**	девятеричная
5	**854**	шестеричная	+**1FH** **C12**	восемьнадцатеричная
6	**255**	пятеричная	+**51FD** A**17**	семнадцатеричная
7	**179**	троичная	+**40B** **8C1**	четырнадцатеричная
8	**2940**	восемьнадцатеричная	+**3122** **1232**	четверичная
9	**723**	семеричная	+**7B19** **5A1**	двенадцатеричная

Задание 4 – Логические основы ЭВМ

Алгебра логики, разработанная в середине IX в. ирландским математиком Д. Булем, является научной основой работы цифровых устройств. В ней действуют принципы (правила), схожие с обычной алгеброй, но буквами (символами) обозначаются не числа, а высказывания. В алгебре Буля переменные принимают только два дискретных значения: логическая **«1»** приписывается истинному высказыванию и логический **«0»** — ложному. Символы нельзя рассматривать как арифметические числа, т. е. алгебра логики является алгеброй состояний, а не чисел. Аппарат алгебры логики используют как для анализа, так и проектирования (синтеза) логических устройств любой сложности в системах цифровой обработки информации. В этом случае можно проводить все исследования строго математически.

Символы **«0»** и **«1»** в алгебре логики характеризуют состояния переменных и состояния их функций. Т.о. логической функцией является функция $f(x_1, x_2 \cdots x_n)$, принимающая значения **«0»** либо **«1»**. Переменные $\boldsymbol{x_1}$, $\boldsymbol{x_2 \ldots x_n}$ также имеют значения **«0»** либо **«1»**.

Любую логическую функцию удобно представить в виде таблицы состояний (таблицы истинности), где записываются возможные комбинации аргументов и соответствующая им функция.

Устройства, предназначенные для формирования функций алгебры логики, называются логическими устройствами. Логические устройства строятся на логических элементах, которые реализуют определённую функцию.

Основные логические операции включают следующие элементарные преобразования двоичных сигналов.

1) Логическое сложение или **дизъюнкция** (от англ. **disjunction** — разъединение), обозначаемое символом **v** и называемое также операцией ИЛИ. Эта операция описывается для простейшей функции двух переменных $\boldsymbol{x_1}$ и $\boldsymbol{x_2}$ в виде логической формулы:

$$y_Д = x_1 \vee x_2. \qquad \textbf{(4.1)}$$

Соотношение **(4.1)** означает, что функция $\boldsymbol{y_Д}$ равна **«1»**, если хотя бы один из аргументов ($\boldsymbol{x_1}$ или $\boldsymbol{x_2}$) равен **«1»**.

Элемент ИЛИ - на выходе этого элемента появится логическая единица тогда, когда хотя бы на одном из входов появится единица. То есть, или на первом, или на втором, или на третьем - на любом из имеющихся входов. Логический ноль на выходе будет только тогда, когда на всех входах будет сигнал логического нуля.

Наиболее просто эту операцию можно реализовать с помощью контактной цепи с двумя параллельно включенными контактами. Сигнал $\boldsymbol{y_Д}$ на выходе такой цепи появится только в том случае, если хотя бы один из контактов замкнут.

Условное обозначение, таблица истинности и другие показатели этой логической функции приведены в таблице **4.1**.

2) Логическое умножение или **конъюнкция** (от англ. **conjunction** — соединение), обозначаемое символом **^** и называемое операцией И. Условное обозначение **&** конъюнкции на логических схемах именуют амперсандом. Для удобства записи сложных логических функций символ конъюнкции можно условно отождествлять со знаком обычного умножения. Для функции двух переменных в этом случае имеем

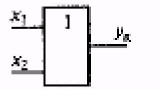
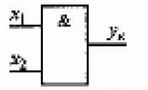
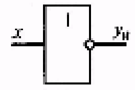
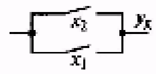
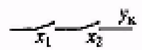
$$y_к = x_1 \wedge x_2 = x_1 \cdot x_2. \qquad \textbf{(4.2)}$$

Соотношение **(4.2)** показывает, что $\boldsymbol{y_к} = \mathbf{1}$ только в том случае, когда оба аргумента ($\boldsymbol{x_1}$ или $\boldsymbol{x_2}$) становятся равными **«1»**.

Элемент И - на выходе этого элемента сигнал логической единицы появляется только тогда, когда на всех входах будет присутствовать логическая единица. То есть, и на первом, и на втором, и на третьем (если он есть), и на всех имеющихся входах. Если хотя бы на одном входе будет ноль, то и на выходе тоже будет ноль.

Условное обозначение и другие показатели функции $\boldsymbol{y_к}$ представлены в третьем столбце (см. табл. **4.1**). Эта операция может быть реализована контактной цепью, состоящей из последовательно включённых контактов.

Таблица **4.1** - Формы отображения основных логических операций

Наименование формы	Вид логической операции		
	Дизъюнкция	Конъюнкция	Инверсия
Символическая	$\vee$; +	$\wedge$; $\cdot$	$\overline{x}$
Буквенная	«ИЛИ»	«И»	«НЕ»
Условная			
Аналитическая	$y_д = x_1 \vee x_2$	$y_к = x_1 \wedge x_2$	$y_и = \overline{x}$
Табличная (истинности)	x_1 x_2 $y_д$ 0 0 0 0 1 1 1 0 1 1 1 1	x_1 x_2 $y_к$ 0 0 0 0 1 0 1 0 0 1 1 1	x $y_и$ 0 1 1 0
Контактная			

3) Логическое отрицание или **инверсия**, обозначаемое черточкой над переменной и называемое операцией НЕ. Эта операция записывается

$$y_и = \overline{x}$$

Как видно, операция выполняется над одной переменной **x** и значение $\boldsymbol{y_и}$ всегда противоположно значению этой переменной. Условное обозначение и другие показатели функции $\boldsymbol{y_и}$ приведены в четвертом столбце (см. табл. **4.1**).

Реализация логической операции НЕ может быть также осуществлена контактной цепью, но (в отличие от цепей, рассмотренных ранее) с помощью нормально замкнутых контактов электромагнитного реле. Отсутствие напряжения на обмотке реле ($\boldsymbol{x} = \mathbf{0}$) предполагает замыкание цепи и появление сигнала на ее выходе, соответствующего логической **«1»** ($\boldsymbol{y_и} = \mathbf{1}$). При наличии напряжения (логической **«1»**) на обмотке реле ($\boldsymbol{x} = \mathbf{1}$) цепь разомкнута, и сигнал на выходе цепи отсутствует ($\boldsymbol{y_и} = \mathbf{1}$).

Сопоставляя таблицы истинности для операций дизъюнкции и конъюнкции (см. табл. **4.1**), можно обосновать следующие соотношения алгебры Буля, имеющие большое практическое значение.

Принцип дуальности, который удобно выразить в виде двух положений:

если $x_1 \vee x_2 = y_Д$, то $\bar{x}_1 \vee \bar{x}_2 = \bar{y}_Д$; **(4.3)**

если $x_1 \wedge x_2 = y_к$, то $\bar{x}_1 \vee \bar{x}_2 = \bar{y}_к$; **(4.4)**

Правило де Моргана вытекает как следствие принципа дуальности и формулируется в виде двух логических соотношений:

$$\overline{x_1 \vee x_2} = x_1 \wedge x_2 = \bar{x}_1 \cdot \bar{x}_2; \quad \textbf{(4.5)}$$

$$\overline{x_1 \cdot x_2} = \overline{x_1 \wedge x_2} = \bar{x}_1 \vee \bar{x}_2.$$

Приведенные соотношения **(4.3)... (4.5)** можно легко обобщить для **n** входных сигналов $x_1, x_2 ..., x_n$. Их широко используют для преобразования сложных логических функций к более простому виду (минимизации функций) при проектировании (синтезе) логических устройств цифровой электроники.

Рассмотренные выше основные логические операции И, ИЛИ и НЕ образуют функционально полный набор, т.е. позволяют реализовать любые логические функции (преобразования) комбинационной логики. Строго говоря, для этой цели можно ограничиться даже двумя операциями, например ИЛИ и НЕ. Однако при использовании только этих трех элементов не всегда удается получить логические устройства наипростейшего вида. Поэтому в логических системах находят применение и другие типовые элементы, реализующие иные логические операции.

Особое значение в цифровой микроэлектронике уделяется двум универсальным логическим операциям, каждая из которых способна самостоятельно образовать функционально полный набор. Как известно, в случае применения только одного базового элемента наблюдается заметное усложнение проектируемых логических устройств. Однако в интегральной технологии удобство изготовления одного базового элемента имеет решающее значение. Поэтому универсальные логические элементы составляют основу большинства интегральных цифровых микросхем.

Универсальные логические операции, реализуемые базовыми элементами, включают две следующие разновидности.

4) Функция Шеффера, обозначаемая символически вертикальной черточкой **|** (штрих Шеффера), отображает операцию И-НЕ. Для простейшей функции двух переменных x_1 и x_2 в этом случае получают

$$x_1 | x_2 = \overline{x_1 \wedge x_2} = \overline{x_1 \cdot x_2} = y_ш \quad \textbf{(4.6)}$$

5) Функция Пирса, обозначаемая символически вертикальной стрелкой (стрелка Пирса), выражает операцию ИЛИ-НЕ. Для функции двух переменных x_1 и x_2 она записывается в виде

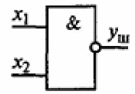
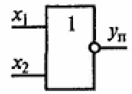
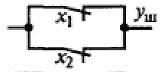
$$x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = y_п \quad \textbf{(4.7)}$$

Важнейшие показатели универсальных логических операций представлены в табл. **4.2.**

Реализацию операций И-НЕ и ИЛИ-НЕ не представляет труда осуществить также в контактной цепи, применяя для этой цели электромагнитные реле с нормально замкнутыми (в отсутствие сигнала на входе управления реле, соответствующее отсутствию напряжения на его обмотке) контактами. Для

реализации операции И-НЕ электромагнитные реле включают в цепь параллельно (см. табл. **4.2**), а в случае операции ИЛИ-НЕ — последовательно.

Таблица **4.2** - Формы отображения универсальных логических операций

Наименование формы	Вид логической операции	
	Функция Шеффера	Функция Пирса
Символическая	|	↓
Буквенная	«И-НЕ»	«ИЛИ-НЕ»
Условная		
Аналитическая	$y_ш = x_1 \| x_2$	$y_п = x_1 \downarrow x_2$
Табличная (истинности)	x_1 x_2 $y_ш$ 0 0 1 0 1 1 1 0 1 1 1 0	x_1 x_2 $y_и$ 0 0 1 0 1 0 1 0 0 1 1 0
Контактная		

Законы алгебры логики

Законы булевой алгебры отражают связи, существующие между операциями, выполняемыми над логическими переменными.

1) $x + 0 = x; \quad x \cdot x = 0$

2) $x + 1 = 1; \quad x \cdot 1 = 0$

3) $x + x = x; \quad x \cdot x = x$

4) $x + \bar{x} = 1; \quad x \cdot \bar{x} = x$

5) $\bar{\bar{x}} = x$

6) $x + y = y + x; \quad x \cdot y = x \cdot y$ (коммутативный закон)

7) $(x + y) + z = x + (y + z); \quad (x \cdot y)z = x(y \cdot z)$ (ассоциативный закон)

8) $\overline{x + y} = \bar{y} \cdot \bar{x}; \quad \overline{x \cdot y} = \bar{x} + \bar{y}$ (правило де Моргана)

9) $x + x \cdot y = x; \quad x(x + y) = x$ (закон поглощения).

10) $x + y \cdot z = (x + y)(x + z); \quad x(y + z) = xy + xz.$

11) $x + \bar{x}y = x + y; \quad x(\bar{x} + y) = xy$ (закон свёртки).

12) $xy + x\bar{y} = x; \quad (x + y)(x + \bar{y}) = x$ (закон склеивания).

Понятие минимизации логических функций

При представлении функции алгебры логики (ФАЛ) математическим выражением используют два вида ее представления.

Дизъюнктивной нормальной формой (ДНФ) называется логическая сумма элементарных логических произведений, в каждое из которых аргумент или его отрицание входят один раз. ДНФ может быть получена из таблицы истинности следующим образом: для каждого набора аргументов, на котором функция равна **«1»**, записывают элементарные произведения переменных, причем переменные, значение которых равно нулю, записывают с инверсией. Полученные произведения, которые носят название конституанта единицы, или минтермов, суммируют.

<u>Пример **4.1**</u>

Таблица **4.3**–Таблица истинности

x_2	x_1	x_0	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Задана логическая функция **3**-х переменных, которая равна единице в случае, если хотя бы две из входных переменных равны единице. Запишем эту функцию в виде таблицы истинности (табл. **4.3**). Для данной логической функции ДНФ имеет вид

$$y(x_2, x_1, x_0) = \overline{x_2} x_1 x_0 + x_2 \overline{x_1} x_0 + x_2 x_1 \overline{x_0} + x_2 x_1 x_0.$$

ДНФ, полученная суммированием конституента единицы, называется совершенной (СДНФ).

Конъюнктивной нормальной формой (КНФ) называется логическое произведение элементарных сумм, в каждую из которых аргумент или его отрицание входят один раз. КНФ может быть получена из таблицы истинности: для каждого набора аргументов, на котором функция равна «О», составляют элементарную сумму, причем переменные, значение которых равно **«1»**, записываются с отрицанием. Полученные суммы, которые носят название конституента нуля или макстермов, объединяют операцией логического умножения.

Пример **4.2**

Для функции из примера **4.1** КНФ:

$$y(x_2, x_1, x_0) = (x_2 + x_1 + x_0)(x_2 + x_1 + \overline{x_0})(x_2 + \overline{x_1} + x_0)(\overline{x_2} + x_1 + x_0).$$

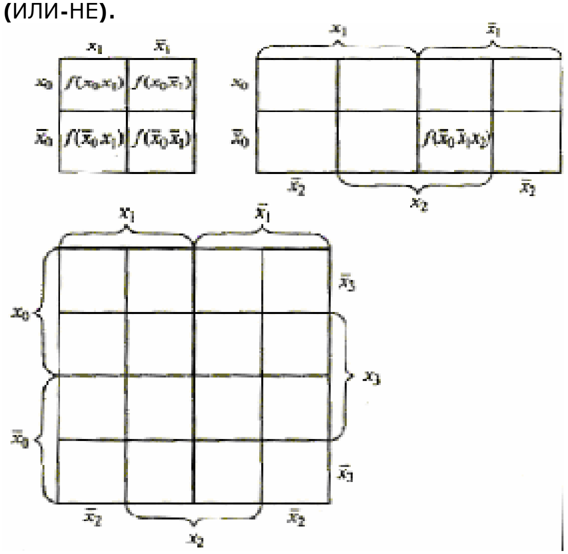
Эта КНФ также называется совершенной (СКНФ), так как каждая элементарная сумма содержит все переменные.

Иногда удобнее пользоваться не самой ФАЛ, а ее инверсией. В этом случае при использовании вышеописанных методик для записи СДНФ надо использовать нулевые, а для записи СКНФ единичные значения функции.

Минимизация ФАЛ. Ранее было показано, что логическую схему, реализующую заданный алгоритм обработки сигналов, можно синтезировать непосредственно по выражению, представленному в виде СДНФ или СКНФ. Однако полученная таким образом схема, как правило, не оптимальна с точки зрения ее практической реализации. Поэтому исходные выражения минимизируют.

Целью минимизации логической функции является минимизация стоимости ее технической реализации. Основным критерием минимизации ФАЛ является критерий уменьшения количества элементарных логических элементов при

использовании только однородных элементов, например, только типа И-НЕ (ИЛИ-НЕ).



Метод минимизации ФАЛ с использованием карт Вейча (Карно) базируется на табличном виде представления ФАЛ. Он широко используется для ручной, без применения ЭВМ минимизации ФАЛ, число переменных в которых не превышает **4...5**.

Картой Вейча (Карно) называется таблица, число клеток которой для функции и переменных равно 2^n, причем каждому минтерму соответствует своя клетка карты (рис. **4.1**).

Рисунок **4.1** - Вид карты Вейча (Карно) для функций **2**, **3** и **4** переменных

Из приведенных рисунков видно, что минтерм представляется минимальным участком площади — одной клеткой на картах Карно (картах минтермов).

Если требуется представить на карте Вейча (Карно) логическую функцию, заданную в виде СДНФ, в соответствующие клетки заносятся единицы. Остальные клетки остаются незаполненными или заполняются нулями. Логическая функция на карте Вейча (Карно) представляется совокупностью клеток, заполненных единицами, а инверсия функции представляется совокупностью пустых или заполненных нулями клеток.

При минимизации ФАЛ используются либо ее единичные, либо нулевые значения. В обоих случаях получаются равносильные выражения, которые, однако, могут отличаться по числу членов и выполненных логических операций.

Алгоритм минимизации сводится к следующему:

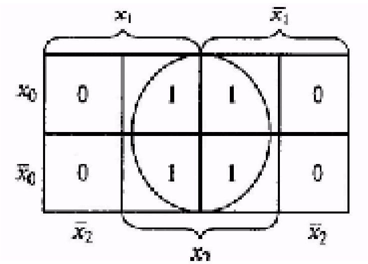
1) На карте Вейча выделяют прямоугольные области, объединяющие выбранные значения функции (**«1»** или **«0»**). Причем каждая область должна содержать 2^K клеток, где к может принимать значения **0**; **1**; **2**; **4**, т.е. **1**; **2**; **4**; **8**; **16** и т.д. клеток. Выделенные области могут пересекаться, т. е. одна клетка может входить в несколько разных областей.

Каждая из выделенных областей является самостоятельным произведением переменных и носит название импликанты.

2) Из полученного множества выделенных областей выбирают минимальное число максимально больших областей, включающих все клетки с выбранным значением ФАЛ. Сумма полученных произведений образует минимальную ДНФ.

При объединении клеток с нулевыми значениями ФАЛ получим ФАЛ, инверсную заданной.

Пример **4.3** Минимизируем ФАЛ:



а) $y = x_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 \bar{x}_0$;

$y = x_2$

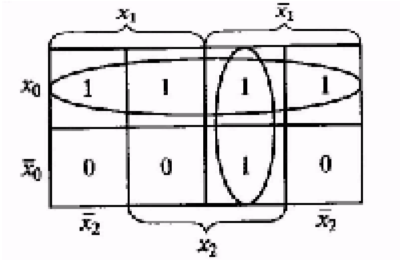
б) $y = \bar{x}_2 x_1 x_0 + x_2 x_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 \bar{x}_1 x_0$;

объединяем по **«1»**

$y = x_0 + x_2 \bar{x}_1$;

объединяем по **«0»**

$\bar{y} = x_1 \bar{x}_0 + \bar{x}_2 \bar{x}_0$



Применяя теорему де Моргана,

$\bar{\bar{y}} = \overline{x_1 \bar{x}_0 + \bar{x}_2 \bar{x}_0} = \overline{x_1 \bar{x}_0} \cdot \overline{\bar{x}_2 \bar{x}_0} = (\bar{x}_1 + x_0)(x_2 + x_0)$.

Из последнего примера видно, что объединение областей по единичным и нулевым значениям функции может привести хотя и к равносильным, но различным минимальным выражениям. Следовательно, может отличаться и схема, реализующая заданный алгоритм. Поэтому для получения минимально простой технической реализации желательно проводить минимизацию с использованием как нулевых, так и единичных значений функции и из полученных минимальных форм выбрать простейшую.

Следует также отметить, что, поскольку карты Вейча (Карно) функций **3** и **4**-х переменных представляют собой объемные фигуры, при формировании областей необходимо помнить, что в одну область объединяются крайние столбцы карты **3**-х переменных, а также четыре угловых клетки карты **4**-х переменных (рис. **4.2**).

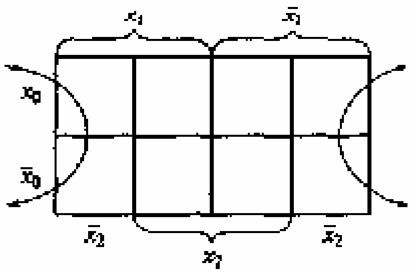
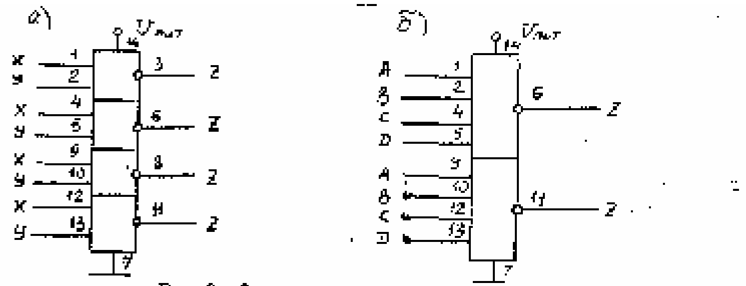


Рисунок **4.2** – Объединение крайних областей в единую область

Синтез логических устройств в заданном базисе логических элементов

На практике в целях уменьшения номенклатуры используемых микросхем часто пользуются функционально полной системой логических элементов в составе двух, выполняющих операции И-НЕ, ИЛИ-НЕ. Любую ФАЛ можно записать в заданном базисе логических элементов.



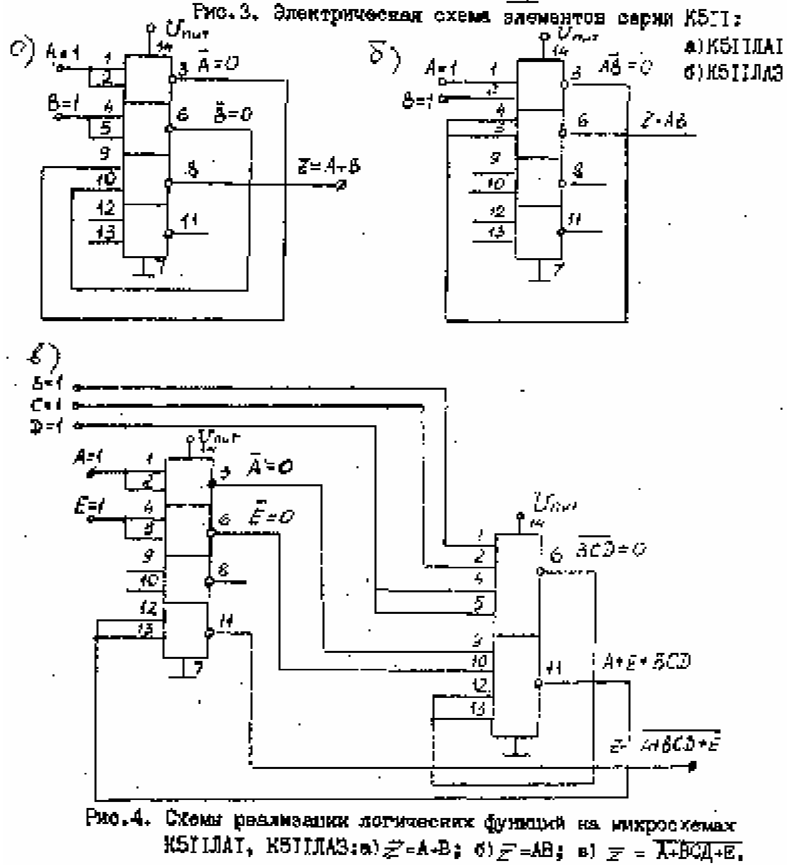


Рисунок **4.3** – Схемы реализации логических функций с помощью интегральных микросхем И-НЕ

Задана логическая функция **3**-х переменных. По заданной таблице истинности:

- составить ДНФ;
- минимизировать функцию с помощью карты Вейча (Карно);
- по полученной функции составить схему реализации функции с помощью интегральных микросхем И-НЕ;
- реализовать данную функцию на контактных элементах.

Таблица **4.4** – Варианты заданий

аргументы			**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**0**
x_0	x_1	x_2	y									
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**0**
0	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**
1	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**
0	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**1**
1	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**
0	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**
1	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**

Задание 5 – Микропроцессорная техника

Раскройте темы в соответствии с вариантом

Таблица **5** – Варианты заданий

№ варианта	Темы	№ варианта	Темы
1	Виды информации, обрабатываемые ЭВМ и способы её кодирования	**11**	Внутренняя архитектура однокристального микропроцессора
2	Микропроцессоры. Назначение, архитектура, основные характеристики	**12**	Контроль работы ЭВМ
3	Шинная структура ЭВМ	**13**	Общая характеристика внешних устройств
4	Структурная схема типовой микропроцессорной системы	**14**	Микроконтроллеры. Назначение, архитектура, основные характеристики
5	Оперативные запоминающие устройства. Динамическая и статическая память	**15**	Алгоритм работы микропроцессорной системы. Основные операции МП
6	Механизм прерываний	**16**	Применение микропроцессорных средств в системах управления
7	Языки программирования	**17**	Структуры программ. Составление программы. Алгоритмы
8	Организация работы процессора с внешними устройствами	**18**	Интерфейс. Классификация, принципы организации
9	Применение ЭВМ в системах управления	**19**	Устройства памяти. Основные характеристики, классификация
10	Интерфейсы **RS-232, RS-485**	**20**	Структура и принцип построения ЭВМ