

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

**Государственное образовательное учреждение
высшего профессионального образования**

**СЕВЕРО-ЗАПАДНЫЙ ГОСУДАРСТВЕННЫЙ ЗАОЧНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

О. Г. АНТОНОВ, Е. Ю. МИГУНОВА

**ЦИФРОВЫЕ УСТРОЙСТВА И
МИКРОПРОЦЕССОРЫ
(ЧАСТЬ 1)**

ЦИ Ф Р О В Ы Е У С Т Р О Й С Т В А

УЧЕБНОЕ ПОСОБИЕ

*Утверждено редакционно - издательским
советом университета в качестве учебного пособия*

САНКТ-ПЕТЕРБУРГ

2006

Антонов О. Г., Мигунова Е. Ю. Цифровые устройства и микропроцессоры. (Часть 1). Цифровые устройства. Учебное пособие. – СПб.: СЗТУ, 2006. – 82 с.

В учебном пособии в краткой форме изложен материал для изучения основных разделов дисциплины «Цифровые устройства и микропроцессоры. (Часть 1). Цифровые устройства».

Учебное пособие ориентировано на студентов факультета радиоэлектроники специальности 200700 – («радиотехника»), направление 654200 (инженер), а также на студентов направления 552500 (бакалавр).

Материал учебного пособия может быть полезен при курсовом и дипломном проектировании, а также специалистам в области цифровой техники.

Р е ц е н з е н т ы: кафедра радиотехники СЗТУ (зав. каф. *Г. И. Худяков*, д-р техн. наук, проф.); *Л. А. Решетов*, канд. техн. наук, доц. СПб Госуниверситета аэрокосмического приборостроения; *С. А. Вылегжанин*, генеральный директор ЗАО «Радиокоммуникационные системы».

ПРЕДИСЛОВИЕ

Дисциплина «Цифровые устройства и микропроцессоры» согласно учебному плану специальности 200700 изучается студентами 3-го курса факультета радиоэлектроники в двух семестрах. В соответствии с этим материал учебного пособия, условно названный авторами «Цифровые устройства» (ЦУ), изучается в 5 – м семестре. Он посвящен изучению логических схем, способам задания логических функций, таким, как словесное описание, таблица истинности и структурная формула, а также способам их минимизации, где упор сделан на графический метод Карно-Вейча.

Из-за ограниченного объема учебного пособия авторы не смогли достаточно подробно остановиться на методе минимизации Квайна-МакКласки, отсылая интересующихся к специальной литературе [3, 7].

Отдельная глава посвящена арифметическим основам цифровой техники, где рассматриваются различные системы счисления, способы кодирования десятичных чисел и способы перевода из одной системы счисления в другую.

Далее излагаются особенности комбинационных устройств, таких, как устройства неравнозначности, равнозначности, полусумматоры, полные сумматоры, дешифраторы, шифраторы, мультиплексоры, демультиплексоры и цифровые компараторы.

В отдельной главе рассматриваются цифровые автоматы, куда входят триггеры, регистры и счетчики.

Специальная глава учебного пособия посвящена изучению аналого-цифровых и цифроаналоговых преобразователей.

В заключительной главе рассмотрены полупроводниковые запоминающие устройства.

Следует отметить, что современный этап развития полупроводниковой электроники отличается массовым выпуском интегральных микросхем с быстро растущей степенью интеграции. Интегральные схемы занимают доминирующее положение в цифровой технике вообще и системах обработки информации в частности.

При изучении дисциплины «Цифровые устройства и микропроцессоры, ч. 1. Цифровые устройства» нужно учитывать, что в основе их построения лежит сравнительно большое число принципов и технических решений. Элементная база ЦУ разнообразна. Для описания ЦУ используется значительное число параметров и характеристик, разобраться в которых, а также спроектировать ЦУ с заданными характеристиками может только квалифицированный специалист, хорошо знакомый с элементной базой современной микросхемотехники.

Изучение дисциплины «Цифровые устройства и микропроцессоры, ч. 1. Цифровые устройства» базируется на знаниях, полученных студентами при изучении математики, физики, информатики, основ теории цепей и электроники.

ВВЕДЕНИЕ

Многие вопросы терминологии для этой части курса соответствуют терминологии импульсных сигналов, что должно найти отражение в предшествующих базовых курсах (понятие длительности импульса, длительности нарастающего и спадающего фронта, скважности импульсов). Поэтому на этих вопросах авторы останавливаются очень кратко, а интересующиеся могут воспользоваться учебной литературой [5,6,8].

Следует отметить, что для рассматриваемых здесь сигналов характерна прямоугольная, или, точнее, трапецеидальная форма импульса при практически любой скважности. Типичная форма такого сигнала показана на рис. 1.1, а, где пунктирной линией показаны сигналы в отрицательной полярности.

Обычно *длительность импульса* $t_{и}$ принято измерять на уровне 0,5 от

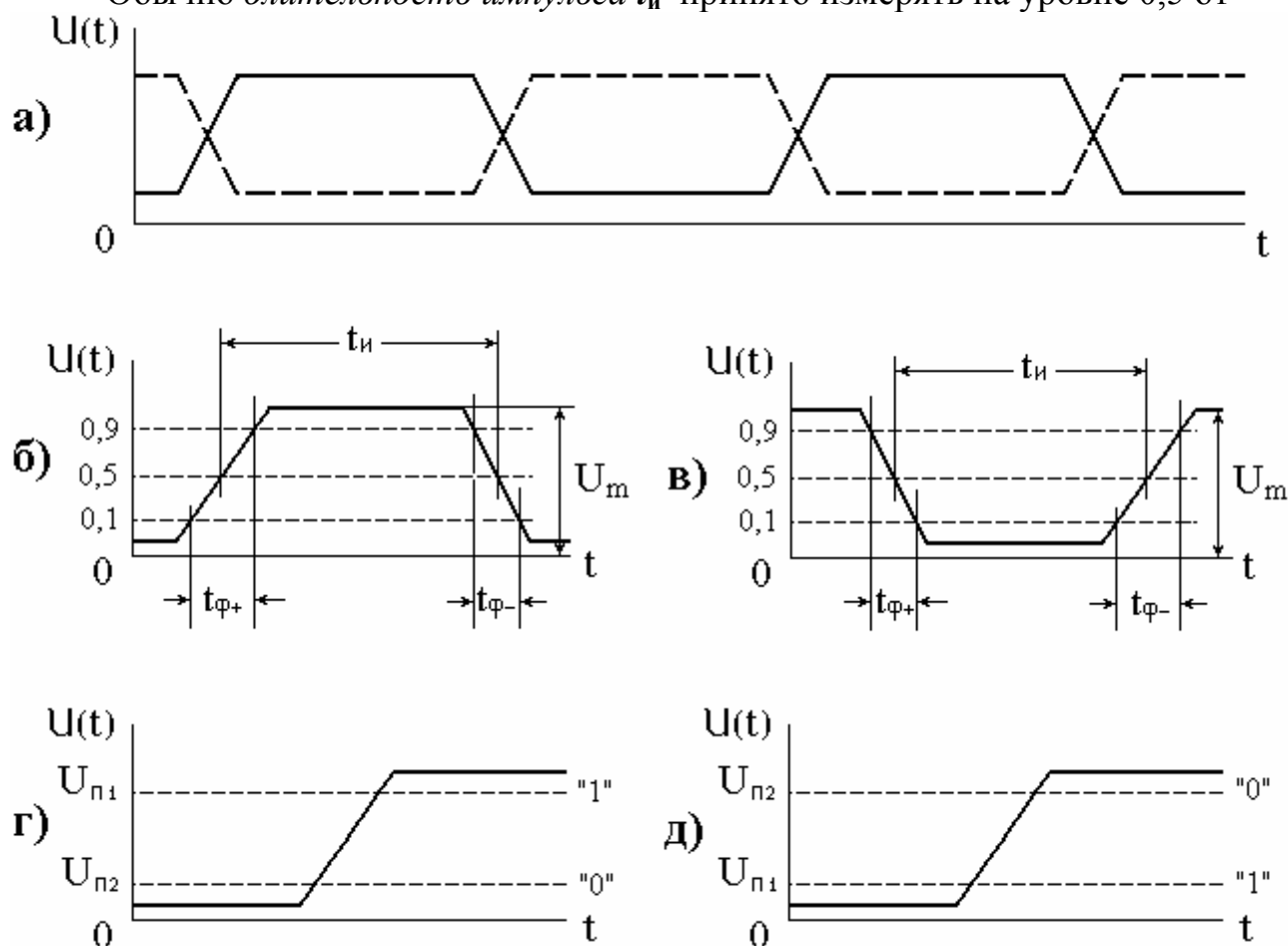


Рис. 1.1

амплитудного значения U_m , а *длительность нарастающего* $t_{\phi+}$ и *спадающего* $t_{\phi-}$ *фронта* между уровнем 0,1 и 0,9 от величины амплитуды, как показано на рис. 1.1, б для импульсов положительной полярности и на рис. 1.1, в – для импульсов отрицательной полярности.

Вместе с тем, следует отметить, что в переводной технической литературе часто применяются термины «*передний фронт*» и «*задний фронт*», которые менее удачны, так как не отражают понятий «*нарастающий*» и «*спадающий*».

Естественно, что для импульса отрицательной полярности передний фронт будет спадающим, а задний – нарастающим.

Характерной особенностью рассматриваемых сигналов является то, что – это сигналы *потенциального* типа. Они формируются в цепях, где отсутствуют разделительные конденсаторы, и поэтому при изменении длительности импульса в несколько раз какого-либо существенного спада вершины не наблюдается. В связи с этим в цифровых устройствах вводится специфическое понятие *информационного уровня*, в качестве которого может быть принят логический «0» или логическая «1». Значения сигналов $u(t)$, превышающие верхний пороговый уровень $U_{п1}$ (рис. 1.1, г), соответствуют логической «1», а значения сигнала, которые меньше, чем нижний пороговый уровень $U_{п2}$, – логическому «0». Этот способ кодирования сигналов носит название *позитивная логика*. При противоположном способе кодирования, который называется *негативная логика*, уровни меняются местами, т. е. низкий уровень напряжения соответствует логической «1», а высокий – логическому «0». Это показано на рис. 1.1, д.

1. ЛОГИЧЕСКИЕ СХЕМЫ

1. 1. Основные понятия алгебры логики

Для математического описания и анализа работы цифровых устройств применяется алгебра логики (булева алгебра). Эта алгебра представляет собой часть математической логики, называемой исчислением высказываний.

Под высказыванием понимается всякое предложение, в котором содержится смысл утверждения (*истинности*) или отрицания (*ложности*). Любое высказывание может быть только истинным, либо ложным. Не может быть высказываний *одновременно* истинных и ложных.

Если высказывание A – истинно, то это можно обозначить $A=1$. Если высказывание B – ложно, то его можно записать $B=0$. Таким образом, любое высказывание определяется либо «0», либо «1». Такие условия алгебры логики позволяют использовать этот математический аппарат для описания процессов в цифровых устройствах.

Таблица 1.1

Функция $f_n(x)$	Аргумент		Условное обозначение функции	Название функции
	0	1		
$f_0(x)$	0	0	0	Константа 0
$f_1(x)$	0	1	x	Переменная x
$f_2(x)$	1	0	\bar{x}	Инверсия x
$f_3(x)$	1	1	1	Константа 1

Переменные величины и функции от них, которые могут принимать только два значения «1» и «0», называются *логическими* или *булевыми* аргументами и функциями. Значение логической функции от n аргументов определяется заданным сочетанием значений всех n аргументов, которое называется *набором аргументов*.

Таблица 1. 2

Функция $f_n(x_1, x_2)$	Аргументы	Условное обозначение функции	Название функции
	0 0 1 1 0 1 0 1		
$f_0(x_1, x_2)$	0 0 0 0	0	Константа 0
$f_1(x_1, x_2)$	0 0 0 1	$x_1 x_2$	Конъюнкция
$f_2(x_1, x_2)$	0 0 1 0	$x_1 \bar{x}_2 = x_1 \Delta x_2$	Запрет по x_2
$f_3(x_1, x_2)$	0 0 1 1	x_1	Переменная x_1
$f_4(x_1, x_2)$	0 1 0 0	$\bar{x}_1 x_2 = x_2 \Delta x_1$	Запрет по x_1
$f_5(x_1, x_2)$	0 1 0 1	x_2	Переменная x_2
$f_6(x_1, x_2)$	0 1 1 0	$x_1 \oplus x_2 = x_1 \bar{x}_2 \vee \bar{x}_1 x_2$	Исключающее ИЛИ
$f_7(x_1, x_2)$	0 1 1 1	$x_1 \vee x_2$	Дизъюнкция
$f_8(x_1, x_2)$	1 0 0 0	$x_1 \downarrow x_2 = \overline{x_1 \vee x_2}$	Стрелка Пирса
$f_9(x_1, x_2)$	1 0 0 1	$x_1 \sim x_2 = x_1 x_2 \vee \bar{x}_1 \bar{x}_2$	Эквивалентность
$f_{10}(x_1, x_2)$	1 0 1 0	\bar{x}_2	Инверсия x_2
$f_{11}(x_1, x_2)$	1 0 1 1	$x_2 \rightarrow x_1 = x_1 \vee \bar{x}_2$	Импликация x_2 к x_1
$f_{12}(x_1, x_2)$	1 1 0 0	\bar{x}_1	Инверсия x_1
$f_{13}(x_1, x_2)$	1 1 0 1	$x_1 \rightarrow x_2 = \bar{x}_1 \vee x_2$	Импликация x_1 к x_2
$f_{14}(x_1, x_2)$	1 1 1 0	$x_1 x_2 = \overline{x_1 x_2}$	Штрих Шеффера
$f_{15}(x_1, x_2)$	1 1 1 1	1	Константа 1

Для любой логической функции от n переменных существует число $z = 2^n$ различных наборов. Так как логическая функция определена на z наборах и может принимать только два значения «0» и «1», то число различных булевых функций от n переменных равно: $2^z = 2^{2^n}$.

Для булевой функции от одного переменного ($n = 1$) существует четыре различных функции (*сингулярные функции*) и для их описания потребуется таблица из 4-х строк (табл. 1. 1). От двух аргументов получим 16 различных булевых функций (*бинарные функции*). Эти функции представлены в табл. 1.2. Для описания бинарных функций потребуется уже 16 строк и т. д. Поэтому слож-

ные логические функции при большом числе *аргументов* обычно представляются как более простые функции от одного или двух аргументов.

1. 2. Логические операции

Операция дизъюнкции. Это операция логического сложения, иначе операция **ИЛИ**. Аналитически операция дизъюнкции записывается следующим образом: $Y = x_1 + x_2 = x_1 \vee x_2$. (1.1)

Чтобы не путать сложение логических функций с операцией сложения в обычной алгебре, применяется специальный знак сложения, но возможно использования и знака алгебраического сложения, что часто делается в технической литературе, если не возникает путаницы между логическими функциями и обычными алгебраическими выражениями.

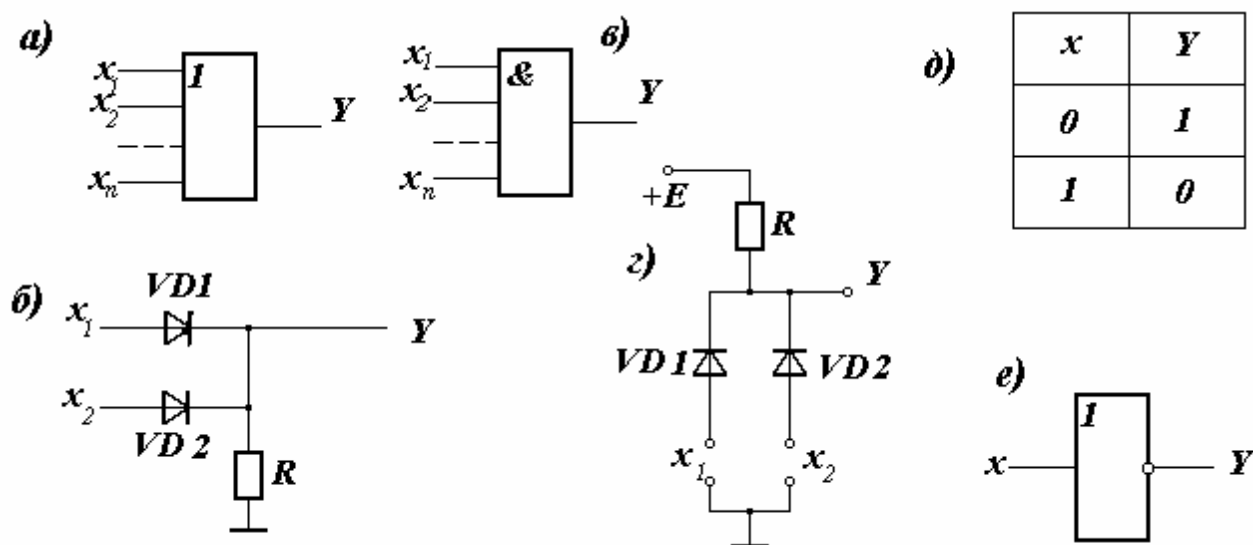


Рис. 1. 2

Выражение (1.1) показывает, Y принимает единичное значение, если хотя бы одна из переменных равна единице. Это выражение можно записать в виде таблицы, которая носит название *таблицы истинности* (табл. 1.3). Эта таблица, дает краткую запись следующих действий:

$$1 \vee 0 = 1; 0 \vee 1 = 1; 1 \vee 1 = 1; 0 \vee 0 = 0. \quad (1.2)$$

Условное обозначение элемента, выполняющего операцию дизъюнкции, и его схемная реализация представлены соответственно на рис. 1.2, *а* и *б*.

Таблица 1.3

x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Таблица 1.4

x_1	x_2	Y
0	0	0
1	0	0
0	1	0
1	1	1

Операция конъюнкции. Эту операцию называют также операцией **И** (операцией логического умножения). Аналитически в случае двух переменных эта операция записывается в виде: $Y = x_1 x_2 = x_1 \wedge x_2$. (1.3)

Смысл операции конъюнкции заключается в том, что $Y=1$, если одновременно $x_1=1$ и $x_2=1$, т. е., если для этой операции составить таблицу истинности (табл. 1.4), то можно сделать вывод, что содержание таблицы кратко говорит о следующем: $0 \cdot 0 = 0$; $0 \cdot 1 = 0$; $1 \cdot 0 = 0$; $1 \cdot 1 = 1$. (1.4)

В случае n переменных: $Y = x_1 x_2 x_3 \dots x_n$. (1.5)

Условное графическое изображение элемента, выполняющего операцию конъюнкции и его схемная реализация на дискретных элементах представлена соответственно на рис. 1.2, в и г.

Операция инверсии. Это операция логического отрицания, иначе операция **НЕ**. Аналитически это записывается в виде: $Y = \bar{x}$. Таблица истинности для этой операции показана на рис. 1.2, д и практически говорит о том, что $\bar{1} = 0$; $\bar{0} = 1$. Схемным примером инвертора может служить ключевой каскад, условное графическое изображение которого представлено на рис. 1.2, е. Кругок около выхода говорит о выполнении операции инверсии в схеме.

Из сопоставления таблиц истинности для операций дизъюнкции, конъюнкции и отрицания можно отметить следующую закономерность. Операции **И** и **ИЛИ** можно поменять местами, если «1» поменять на «0», а логическое умножение – на логическое сложение: если $x_1 \vee x_2 = Y$, то $\bar{x}_1 \bar{x}_2 = \bar{Y}$; если $x_1 x_2 = Y$, то $\bar{x}_1 \vee \bar{x}_2 = \bar{Y}$. (1.6)

Равенства (1.6) отражают принцип двойственности в алгебре логики. Основные правила выполнения операций дизъюнкции и конъюнкции для одной переменной с константами и с инверсией самой переменной можно сформулировать следующим образом:

$$\begin{aligned} x \vee 0 &= x; x \vee 1 = 1; x \vee x = x; x \vee \bar{x} = 1; \\ x \cdot 0 &= 0; x \cdot 1 = x; x \cdot x = x; x \cdot \bar{x} = 0; \bar{\bar{x}} = x; x \vee x = x \vee x \vee x \text{К} = x. \end{aligned} \quad (1.7)$$

1. 3. Законы и теоремы булевой алгебры

Закон коммутативности. Этот закон говорит о том, что порядок записи переменных не влияет на результат: $x_1 \vee x_2 = x_2 \vee x_1$, т. е. у этого закона есть аналог в обычной алгебре.

Закон ассоциативности, или сочетательный закон:

$x_1 \vee x_2 \vee x_3 = x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3$; $x_1 x_2 x_3 = x_1 (x_2 \cdot x_3)$. Этот закон также имеет аналог в обычной алгебре.

Закон дистрибутивности, или распределительный закон, имеет две формы записи: $x_1 (x_2 \vee x_3) = x_1 x_2 \vee x_1 x_3$; (1.8)

$$x_1 \vee x_2 x_3 = (x_1 \vee x_2)(x_1 \vee x_3). \quad (1.9)$$

Форма записи (1.8) имеет аналог в обычной алгебре, а форма (1.9) не имеет аналога. Поэтому справедливость равенства (1.9) требует доказательства: *)

$$\begin{aligned} (x_1 \vee x_2)(x_1 \vee x_3) &= x_1x_1 \vee x_2x_1 \vee x_1x_3 \vee x_2x_3 = \\ &= x_1 \vee x_2x_1 \vee x_1x_3 \vee x_2x_3 = x_1(1 \vee x_2 \vee x_3) \vee x_2x_3 = x_1x_2x_3. \end{aligned} \quad (1.10)$$

Закон поглощения. Этот закон имеет две формы записи. Для операции **ИЛИ**: $x_1 \vee x_1x_2 = x_1$; (1.11) и для операции **И**: $x_1(x_1 \vee x_2) = x_1$. (1.12)

Доказательства равенств (1.11) и (1.12) очень просты:

$$x_1(1 \vee x_2) = x_1 \cdot 1 = x_1; \quad x_1(x_1 \vee x_2) = x_1x_1 \vee x_1x_2 = x_1 \vee x_1x_2. \quad (1.13)$$

Равенство (1.13) соответствует (1.11). Соответственно для операции **ИЛИ**: $x_1 \vee x_1x_2 = x_1(1 \vee x_2) = x_1 \cdot 1 = x_1$. (1.14)

Равенство (1.14) также соответствует (1.11).

Закон склеивания. Как и в предыдущем случае, имеется два варианта:

$$x_1x_2 \vee \overline{x_1}x_2 = x_2; \quad (x_1 \vee x_2)(\overline{x_1} \vee x_2) = x_2. \quad (1.15)$$

Доказательство для (1.15): $x_2(x_1 \vee \overline{x_1}) = x_2 \cdot 1 = x_2$. (1.17)

При доказательстве (1.16) необходимо раскрыть скобки и воспользоваться правилами (1.7) и законом (1.15): $x_1x_1 \vee x_1x_2 \vee x_2x_1 \vee x_2x_2 = x_2$. (1.18)

Правило де Моргана. Справедливость этого правила вытекает из принципа двойственности булевой алгебры, приведенного в (1.6):

$$\overline{x_1 \vee x_2} = \overline{x_1} \overline{x_2}; \quad (1.19) \quad \overline{x_1x_2} = \overline{x_1} \vee \overline{x_2}. \quad (1.20)$$

1. 4. Способы задания логических функций

Словесное описание. Первый способ задания логической функции – это ее словесное описание. Рассмотрим конкретный пример задания функции таким способом, т. е. опишем работу устройства словами.

Функция трех аргументов принимает значение «1», если два любых аргумента или все три равны «1»; во всех других случаях эта функция равна «0».

Таблица 1.5

x_3	x_2	x_1	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

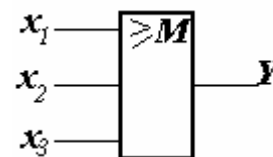


Рис 1.3

*) Примечание. При написании логических выражений в математической форме, чтобы избежать случайных ошибок, операция **ИЛИ** в дальнейшем будет обозначаться логическим символом \vee , а операция **И**, как в обычной алгебре.

Логическая схема, соответствующая функции, описанной таким образом, называется *мажоритарным элементом типа «2 из 3-х»*.

Мажоритарный элемент – это логическое устройство, применяемое для повышения достоверности передаваемой информации по однотипным каналам связи с помехами. На схемах электрических принципиальных он изображается, как показано на рис. 1.3. Принцип действия мажоритарного элемента нагляднее виден, если для его описания применить таблицу истинности (табл. 1.5), т. е. воспользоваться табличным способом задания функции.

Табличный способ. Табличный способ основан на задании логической функции с помощью таблицы истинности. Табл. 1.5 полностью характеризует функционирование мажоритарного элемента типа «2 из 3-х». Из таблицы видно, что функция $Y=1$, в четырех случаях, когда два входных сигнала – единичные, что действительно соответствует словесному описанию.

Аналитический способ. Логическую функцию можно задать аналитически, воспользовавшись правилами и теоремами булевой алгебры. Для удобства рассмотрим тот же самый пример с мажоритарным элементом «2 из 3-х». Из таблицы истинности выпишем четыре строчки, когда $Y=1$:

- 1) $x_1=0, x_2=1, x_3=1$ для сочетания $\bar{x}_1 x_2 x_3 = Y = 1$;
- 2) $x_1=1, x_2=0, x_3=1$ для сочетания $x_1 \bar{x}_2 x_3 = Y = 1$;
- 3) $x_1=1, x_2=1, x_3=0$ для сочетания $x_1 x_2 \bar{x}_3 = Y = 1$;
- 4) $x_1=1, x_2=1, x_3=1$ для сочетания $x_1 x_2 x_3 = Y = 1$.

Каждое из произведений переменных, для которых Y – истинно, называется *минтермом*, и искомую функцию можно представить как дизъюнкцию минтермов:

$$Y = f(x_1, x_2, x_3) = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3. \quad (1.21)$$

Выражение (1.21) содержит конъюнкции всех переменных или их инверсий и называется *совершенной дизъюнктивной нормальной формой (СДНФ)*, иначе структурной формулой, (*1-я стандартная форма*).

Существует другая форма записи, которая носит название *совершенной конъюнктивной нормальной формы (СКНФ)*. Это - *2-я стандартная форма*, для получения которой необходимо в таблице истинности выделить строки с нулевыми значениями функции, когда $Y=0$:

$$Y = (\bar{x}_1 \vee x_2 \vee x_3)(\bar{x}_2 \vee x_3 \vee x_1)(x_1 \vee x_2 \vee \bar{x}_3)(x_1 \vee x_2 \vee x_3). \quad (1.22)$$

1. 5. Переход от структурной формулы к логической схеме.

Понятие функциональной полноты и логического базиса

При построении логической схемы по структурной формуле можно пользоваться обеими стандартными формами, но предпочтение отдать той, где будет меньше элементов. Схема, которая получается на основании структурной формулы, носит название *комбинационной* и обычно содержит все логические элементы: И, ИЛИ, НЕ. В дальнейшем будет показано, что, выполняя заданную

логическую функцию, такая схема не будет оптимальной в смысле конструкции.

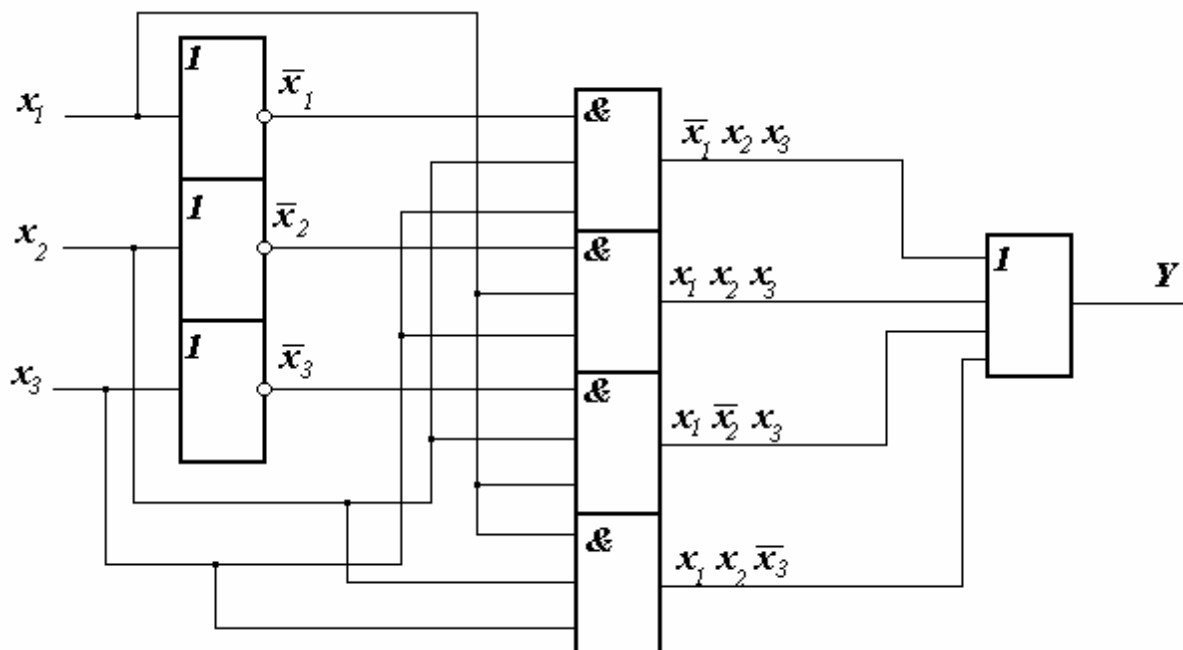


Рис. 1. 4

При переходе от структурной формулы к логической схеме элементы, предназначенные для выполнения предусмотренных в формуле операций, располагаются на схеме, начиная от входов, в таком же порядке, в каком выполняются сами операции.

В этом параграфе удобно продолжить рассмотрение уже знакомого по предыдущему материалу примера с мажоритарным элементом. Воспользуемся структурной формулой (1.21).

На комбинационной схеме сначала изобразим *инверторы* для отрицания входных сигналов, затем элементы **И**, которых согласно (1.21) должно быть четыре и в самом конце один элемент **ИЛИ** с четырьмя входами и одним выходом (рис. 1. 4). Приведенное выше условное изображение мажоритарного элемента (рис. 1. 3) не говорит, из каких элементов он построен, и это уточняется при построении схемы (рис. 1. 4).

Функциональная полнота систем логических элементов. Система логических элементов **И**, **ИЛИ**, **НЕ** *достаточна* для построения любых комбинационных устройств. Такая система будет *функционально полной*, но полнота этой системы *избыточна*. Покажем, что один элемент (**И** или **ИЛИ**) из системы может быть исключен с сохранением ее функциональной полноты.

Предположим, что исключен элемент **ИЛИ**. Остались два элемента **И** и **НЕ**. С помощью этих элементов также можно выполнить операцию *дизъюнкции*, используя *правило де Моргана* (1.19). Представим (1.19) в виде (1.23):

$$x_1 \vee x_2 = \overline{\overline{x_1} \cdot \overline{x_2}} \quad (1.23).$$

Этому выражению соответствует логическая схема (рис. 1.5, а).

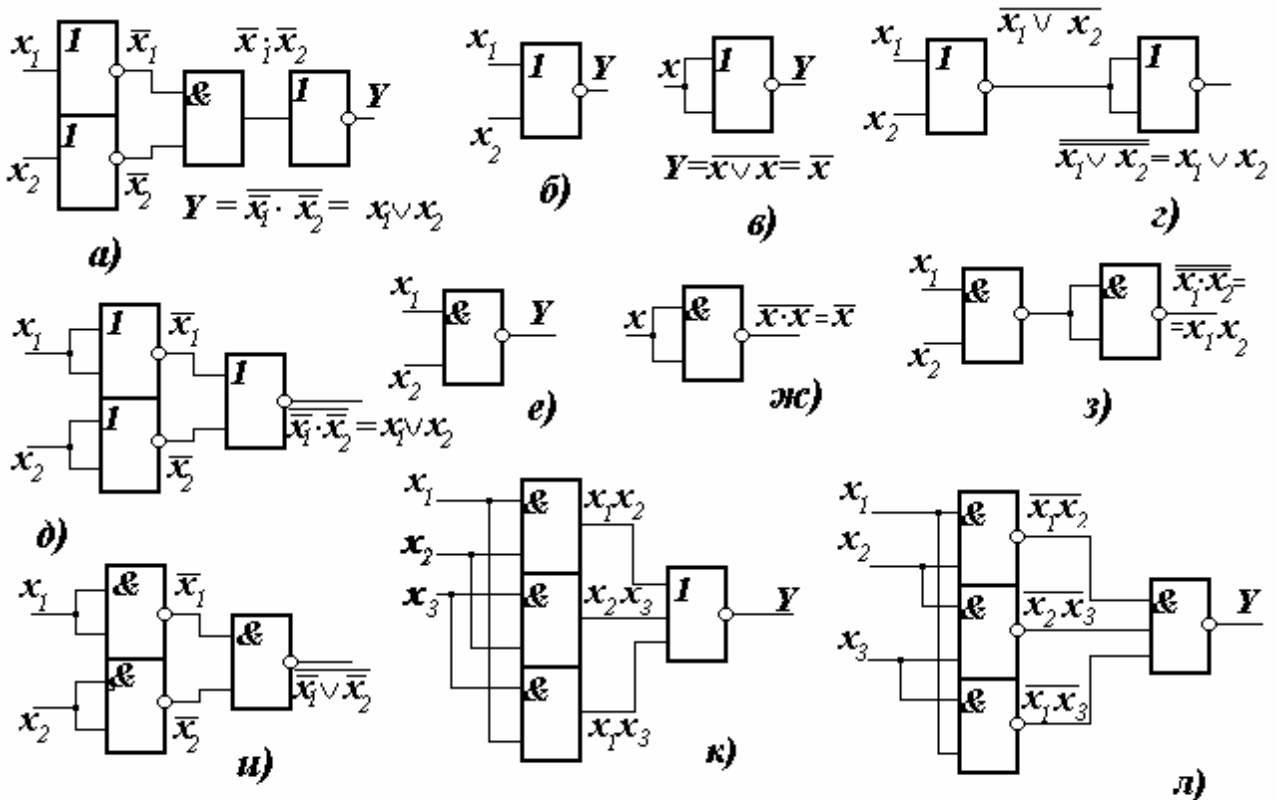


Рис. 1.5

Возможность получения операции ИЛИ в системе элементов И, НЕ говорит о функциональной полноте системы из двух элементов И–НЕ. В этом случае говорят, что логическая схема построена в *базисе* И–НЕ.

1. 6. Универсальные логические элементы

Можно показать, что, уменьшая далее количество логических элементов, приходим к системе из *одного* логического элемента, выполняющего все логические операции. Такие логические элементы называются *универсальными* и бывают двух типов.

Универсальный логический элемент ИЛИ – НЕ. Этот логический элемент реализует логическую функцию *отрицание дизъюнкции*: $Y = x_1 \vee x_2$. В табл. 1. 2 эта функция называется «Стрелка Пирса» и обозначается: $Y = x_1 \downarrow x_2$. Условное графическое обозначение элемента показано на рис. 1.5, б. На рис. 1. 5, в показано, как с помощью универсального элемента полу-

чить функцию НЕ, т. е. выполнить операцию инверсии. Рис. 1. 5, з показывает реализацию функции ИЛИ, а рис. 1. 5, д – функции И.

Универсальный логический элемент И – НЕ. Этот логический элемент также образует функционально-полную систему логических элементов. Как видно из табл. 1.2, этот логический элемент реализует логическую функцию $Y = x_1 \cdot x_2 = x_1 | x_2$ («Штрих Шеффера»). Его условное графическое изображение показано на рис. 1. 5, е, а на рис. 1. 5, ж – выполняемая им операция НЕ. Две другие логические операции, выполняемые этим универсальным элементом И и ИЛИ, показаны соответственно на рис. 1. 5, з и и. В общем случае все универсальные логические элементы могут иметь не два, а n входов.

1. 7. Минимизация логических схем алгебраическим методом

Как было показано на примере мажоритарного элемента типа «2 из 3-х», прямой способ построения схемы по структурной формуле обычно не получается удовлетворительным с практической точки зрения: *применяются разнотипные логические элементы, количество элементов ничем не ограничено.*

Задача минимизации сводится к тому, чтобы после получения аналитической формы записи структурной формулы $Y = f(x_1, x_2, \dots, x_n)$, выполнить ее минимизацию, т. е. найти такую форму записи, которая потребует при реализации наименьшего числа элементов. Типовыми приемами при этом можно считать:

1) прибавление одного или нескольких однотипных членов из числа имеющихся в первой стандартной форме, т. к. $A \vee A \vee A \vee A = A$;

2) умножение отдельных членов функции на сумму $A \vee \bar{A}$, где A может быть одной из переменных $x_1, x_2, x_3 \dots x_n$, так и функцией этих переменных; так как $A \vee \bar{A} = I$, то такое умножение не нарушает тождественности исходного и полученного выражений;

3) выделение слагаемых типа $A \vee \bar{A}$ путем применения закона дистрибутивности; после представления суммы в виде двух сомножителей, один из которых $A \vee \bar{A} = I$, выражение упростится;

4) использование законов поглощения и склеивания.

После проведения всех возможных преобразований получают функцию, не имеющую избыточных членов и не поддающуюся дальнейшей минимизации. Эту форму записи называют *тупиковой*.

Минимизируем для примера алгебраическим путем СДНФ структурной формулы для мажоритарного элемента типа «2 из 3-х». В выражении (1.21) к последнему члену, *не содержащему инверсий*, прибавим два еще таких же:

$$Y = x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 x_2 x_3. \quad (1.24)$$

Произведем группировку членов выражения, которое получили, на основании закона дистрибутивности:

$$Y = x_2 x_3 (\bar{x}_1 \vee x_1) \vee x_1 x_3 (\bar{x}_2 \vee x_2) \vee x_1 x_2 (\bar{x}_3 \vee x_3) = x_1 x_2 \vee x_2 x_3 \vee x_1 x_3. (1.25)$$

Выражение (1.25) – *тупиковая форма* записи. Теперь можно составить логическую схему (рис. 1.5, к), из которой видно, что по сравнению со схемой, построенной по структурной формуле (СДНФ), изображенной на рис. 1.4, общее количество элементов уменьшилось *в два раза*, а также уменьшилось число входов у некоторых элементов. Обычно желательно построить схему на *однотипных элементах*. Покажем для примера, как это выполнить на универсальных логических элементах И – НЕ (рис. 1.5, л). На выходе схемы будет сформирован сигнал, описываемый равенством:

$$Y = (x_1 x_2)(x_2 x_3)(x_1 x_3). (1.26)$$

Воспользуемся правилом де Моргана и преобразуем выражение (1.26) к виду (1.27): $Y = \overline{x_1 x_2} \vee \overline{x_2 x_3} \vee \overline{x_1 x_3}$; (1.27). Это позволит получить *тупиковую форму*: $Y = x_1 x_2 \vee x_2 x_3 \vee x_1 x_3$. (1.28)

Следует отметить, что минимизация алгебраическим методом предполагает наличие некоторых *практических навыков*: бывает не всегда очевидно, что полученная форма записи – тупиковая; часто трудно определить склеивающиеся слагаемые.

1. 8. Графический метод минимизации Карно – Вейча

Графический метод минимизации логических функций использует карты Карно или диаграммы Вейча, что практически то же самое. Этот метод автоматизирует процедуру поиска *склеивающихся слагаемых*.

Карта Карно – это таблица, имеющая ячейки для всех возможных минтермов функции. Метод обычно применяется при небольшом числе минтермов (обычно не более 5 ... 6). Для функции *двух переменных* карта Карно имеет вид рис. 1.6, а и содержит *четыре* клетки: каждая клетка предназначена для своего минтерма.

Если какой-то из этих минтермов в СДНФ присутствует, то в соответствующей клетке ставится «1». Если какого-либо минтерма нет, то проставляется

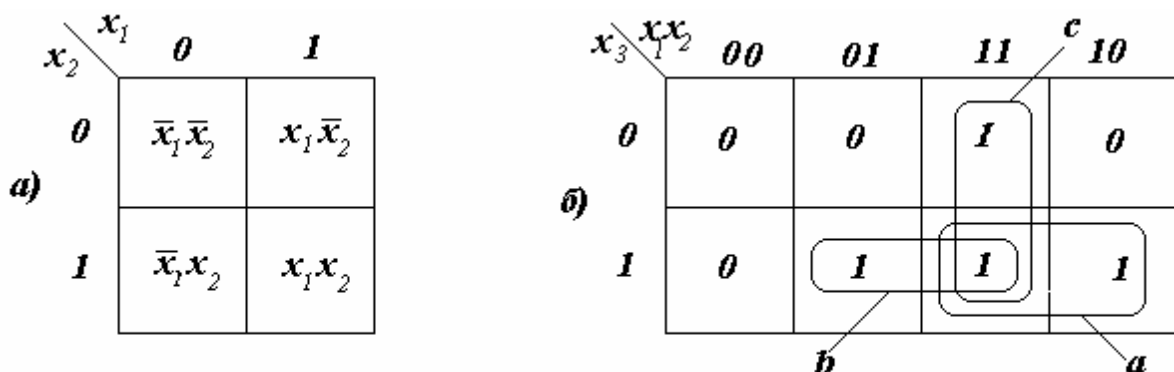


Рис. 1. 6

«0». Пользуясь этим правилом легко построить карту для мажоритарного элемента типа «2 из 3-х», что соответствует функции трех переменных.

Следует учитывать, что нумерация клеток в картах Карно по горизонтали и вертикали должна соответствовать коду Грея (табл. 2.1, с. 25), и для мажоритарного элемента карта Карно показана на рис. 1.6, б. Склеивание производится между теми минтермами, которые записаны в виде «1» в соседних клетках карты по вертикали и горизонтали. На рис. 1.6, б имеется три таких контура: *a*, *b*, *c*.

Контур «*a*» отражает склеивание минтермов:

$$x_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 = x_1 x_3 (x_2 \vee \bar{x}_2) = x_1 x_3 . \quad (1.29)$$

Контур «*b*» отражает склеивание минтермов:

$$x_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 = x_2 x_3 (x_1 \vee \bar{x}_1) = x_2 x_3 . \quad (1.30)$$

Контур «*c*» отражает склеивание минтермов:

$$x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 = x_1 x_2 (x_3 \vee \bar{x}_3) = x_1 x_2 . \quad (1.31)$$

В результате выполненных операций склеивания из четырех слагаемых остались три: $Y = x_1 x_2 \vee x_2 x_3 \vee x_1 x_3$, (1.32)

т. е. получена та же тупиковая форма, найденная алгебраически в виде (1.28).

Особенности минимизации функции 4-х переменных. Карта Карно функции 4-х переменных $Y = f(x_1, x_2, x_3, x_4)$ имеет вид квадрата, содержащего 16 ячеек. Каждая ячейка соответствует определенному минтерму в пер-

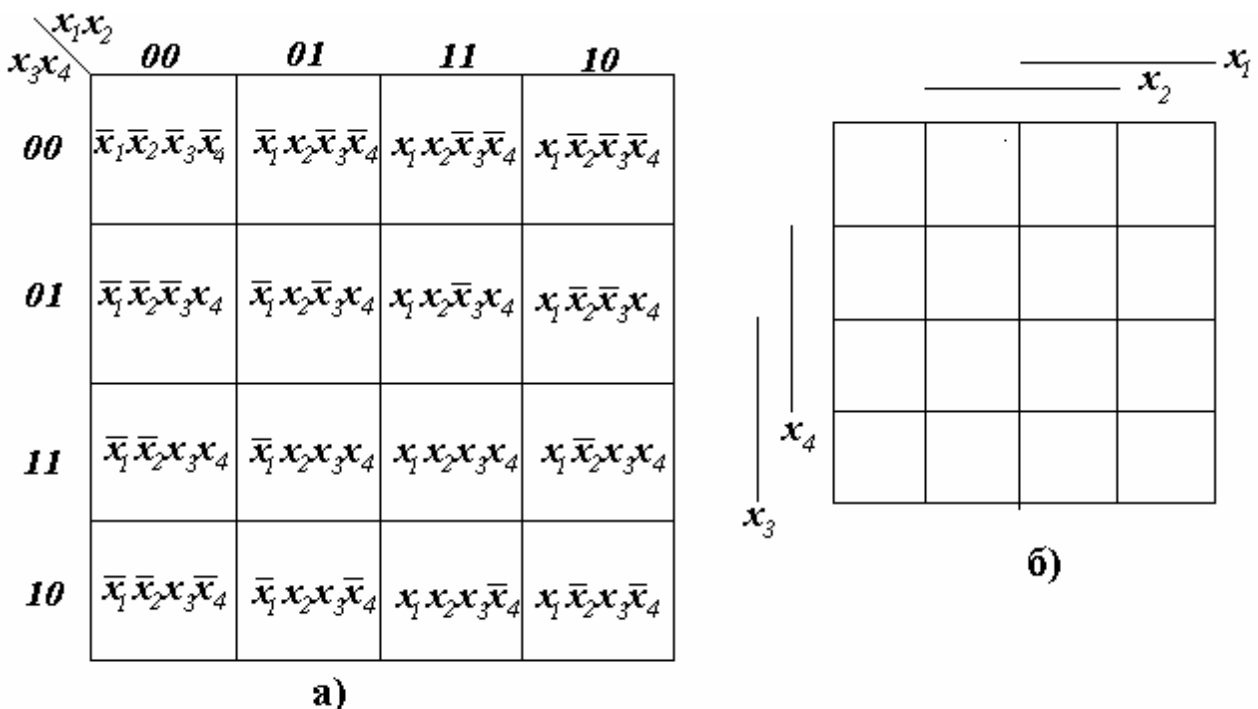


Рис. 1. 7

вой стандартной форме (рис. 1. 7, а).

При практической работе в клетки карты вписывают «1» или «0» по рассмотренному выше принципу. Кроме того, иногда применяют специальные обозначения клеток, в которых аргументы равны «1», как показано на

рис. 1.7, б, в виде отрезков линий, около которых пишутся аргументы. 8 квадратов против черты с буквой соответствуют слагаемым, в которые эта переменная входит без отрицания. В остальные 8 квадратов (наборов) эта переменная входит с отрицанием, т. е. переменная равна «0».

Задание логической функции числовым способом. При этом способе задания функции в углу клетки проставляется номер набора, который должен соответствовать номеру клетки в двоичном коде (рис. 1. 8, а).

Для числового представления функции в первой стандартной форме под знаком суммы перечисляют номера наборов, на которых функция равна «1». Подразумевается, что на остальных наборах функция равна «0».

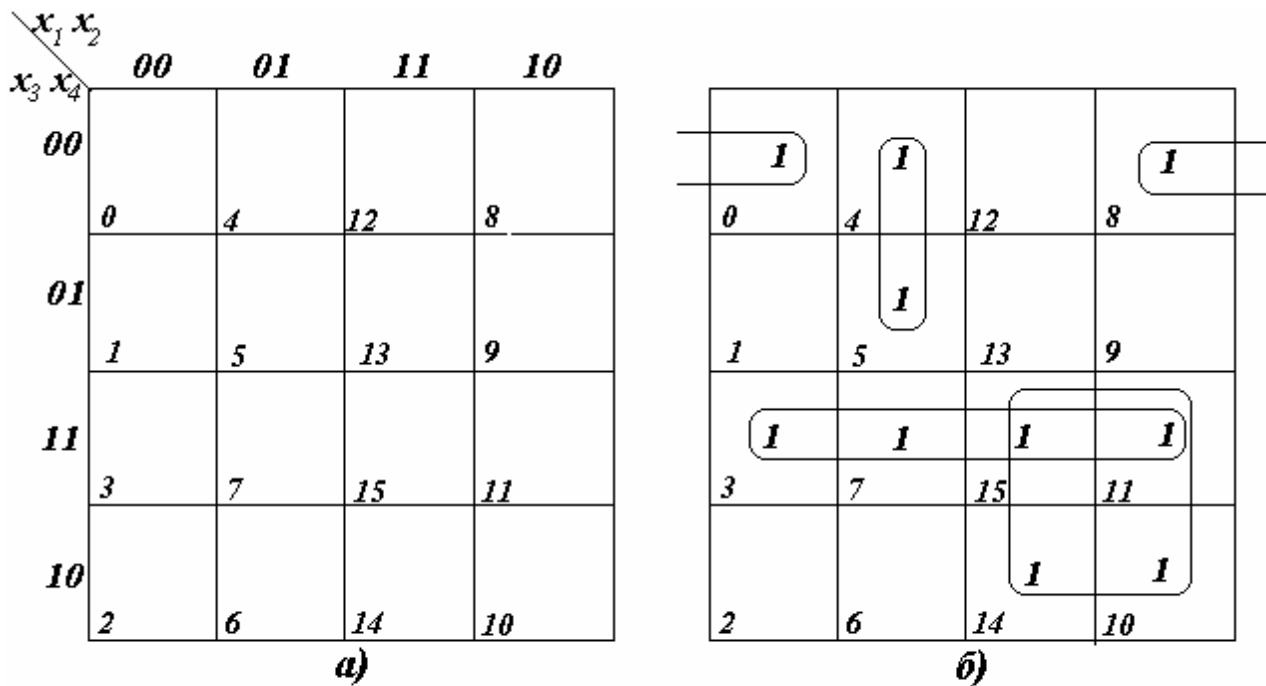


Рис. 1. 8

Например, $f(x_1, x_2, x_3, x_4) = \sum(0, 3, 5, 6, 7)$. Для представления во второй стандартной форме под знаком произведения перечисляются наборы, на которых функция равна «0». Например, $f(x_1, x_2, x_3, x_4) = \prod(1, 2, 4, 8, 9)$. Более подробно процедуру минимизации можно показать на следующем примере. Дана функция $f(x_1, x_2, x_3, x_4) = \sum(0, 3, 4, 5, 7, 8, 10, 11, 14, 15)$. Ее алгебраическая форма для выполнения процедуры минимизации не обязательна, т. к. перечислены наборы, на которых функция равна «1», но для наглядности эффекта минимизации приведем и алгебраическую запись:
$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} \overline{x_2} \overline{x_3} x_4 \vee \overline{x_1} \overline{x_2} x_3 \overline{x_4} \vee \overline{x_1} \overline{x_2} x_3 x_4 \vee \overline{x_1} x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} x_4 \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4 \vee x_1 \overline{x_2} \overline{x_3} \overline{x_4} \vee x_1 \overline{x_2} \overline{x_3} x_4 \vee x_1 \overline{x_2} x_3 \overline{x_4} \vee x_1 \overline{x_2} x_3 x_4 \vee x_1 x_2 \overline{x_3} \overline{x_4} \vee x_1 x_2 \overline{x_3} x_4 \vee x_1 x_2 x_3 \overline{x_4} \vee x_1 x_2 x_3 x_4. \quad (1.33)$$

Нанесем эту функцию на карту Карно и произведем все возможные склеивания (рис. 1. 8, б). Склеиваются две или четыре соседние единицы. При

этом 0-й набор и 8-й будут соседними, т. к. ось симметрии проходит через се-

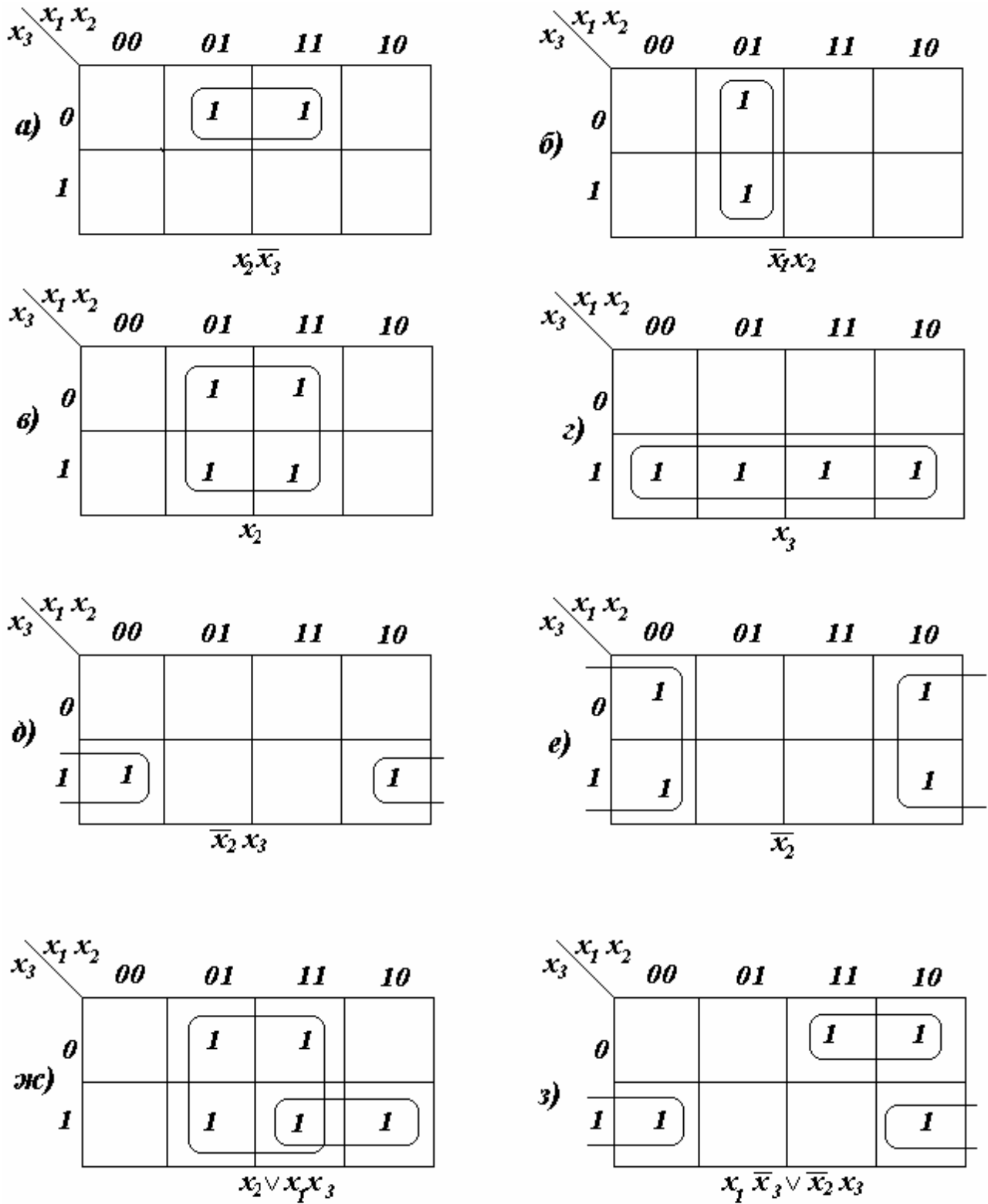


Рис. 1. 9

редину рисунка. В результате получим:

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 x_4 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3. \quad (1.34)$$

Особенности процедуры склеивания по карте Карно. Общее правило выполнения всевозможных склеиваний по карте Карно заключается в том, что склеиваются две единицы в соседних клетках: это уменьшает количество аргументов в минтерме *на один*. Если при склеивании единиц при переходе от одной клетки к другой изменяется знак, как показано на рис. 1. 9, *а* и *б* для случая трех переменных, то один из аргументов выпадает. На рис. 1.9, *а* – это x_1 , а на рис. 1. 9, *б* – это x_3 . В результате склеивания можно получить и один аргумент, если удастся объединить *четыре соседних клетки*, как показано на рис. 1. 9, *в* и *г* (функция трех переменных). При этом, как уже говорилось выше, необходимо учитывать ось симметрии, что позволяет рассматривать крайние клетки как соседние (рис. 1. 9, *д* и *е*).

При объединении единиц на карте Карно сначала надо стремиться получить *максимальную группу*, т. к. при этом получается функция с минимальным числом элементов, но также нужно следить, чтобы не было *отдельно стоящих* единиц, дающих максимальный по количеству элементов минтерм. Если *все единицы* вошли в какие-либо объединения, то создавать *лишние объединения* не следует, т. к. это только увеличивает длину структурной формулы.

Например, на рис. 1. 9, *ж* получена максимальная группа из 4-х единиц, и одна из единиц этой группы объединена с отдельно стоящей единицей в крайнем правом столбце. Это минимизировало структурную формулу. Рис.1.9, *з* иллюстрирует случай, когда две, стоящие рядом единицы в крайнем правом столбце, *не объединяются*. Это объясняется тем, что *все* единицы вошли в объединения, а то объединение, о котором идет речь, было бы избыточным, другими словами, *лишним*.

В случае четырех переменных максимальное объединение содержит 8 единиц (рис. 1. 10, *а*). Кроме того, в этом случае появляется дополнительная, горизонтальная ось симметрии (рис. 1. 10, *б*), т. е. карту Карно можно представить себе обернутой вокруг цилиндра, что приводит к тому, что боковые кромки соединяются.

В случае четырехклеточных объединений будут получаться минимизированные функции в виде конъюнкции двух аргументов, а в двухклеточных, соответственно, конъюнкции трех аргументов (для функции четырех переменных). Но при этом надо следить, чтобы не сделать *лишних* объединений, что приведет к ошибке, как показано на рис. 1. 10, *в* пунктиром. Объединение, обведенное пунктиром, не следует учитывать, т. к. оно – *избыточное*. Результат минимизации в этом случае будет описываться выражением (1.35):

$$Y = \overline{x_1} \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 x_4 \vee x_1 x_2 \overline{x_4} \vee x_1 x_3 x_4. \quad (1.35)$$

На рис. 1. 10, *г* показана минимизация функции, для которой возможны два минимальных представления: (1.36) и (1.37).

$$\text{I-й вар. } Y = \overline{x_1} \overline{x_3} \vee \overline{x_2} \overline{x_3} x_4 \vee x_1 x_3 x_4;$$

(1.36)

$$\text{II-й вар. } Y = \overline{x_1} \overline{x_3} \vee x_1 x_3 x_4 \vee x_1 x_2 x_4. \quad (1.37)$$

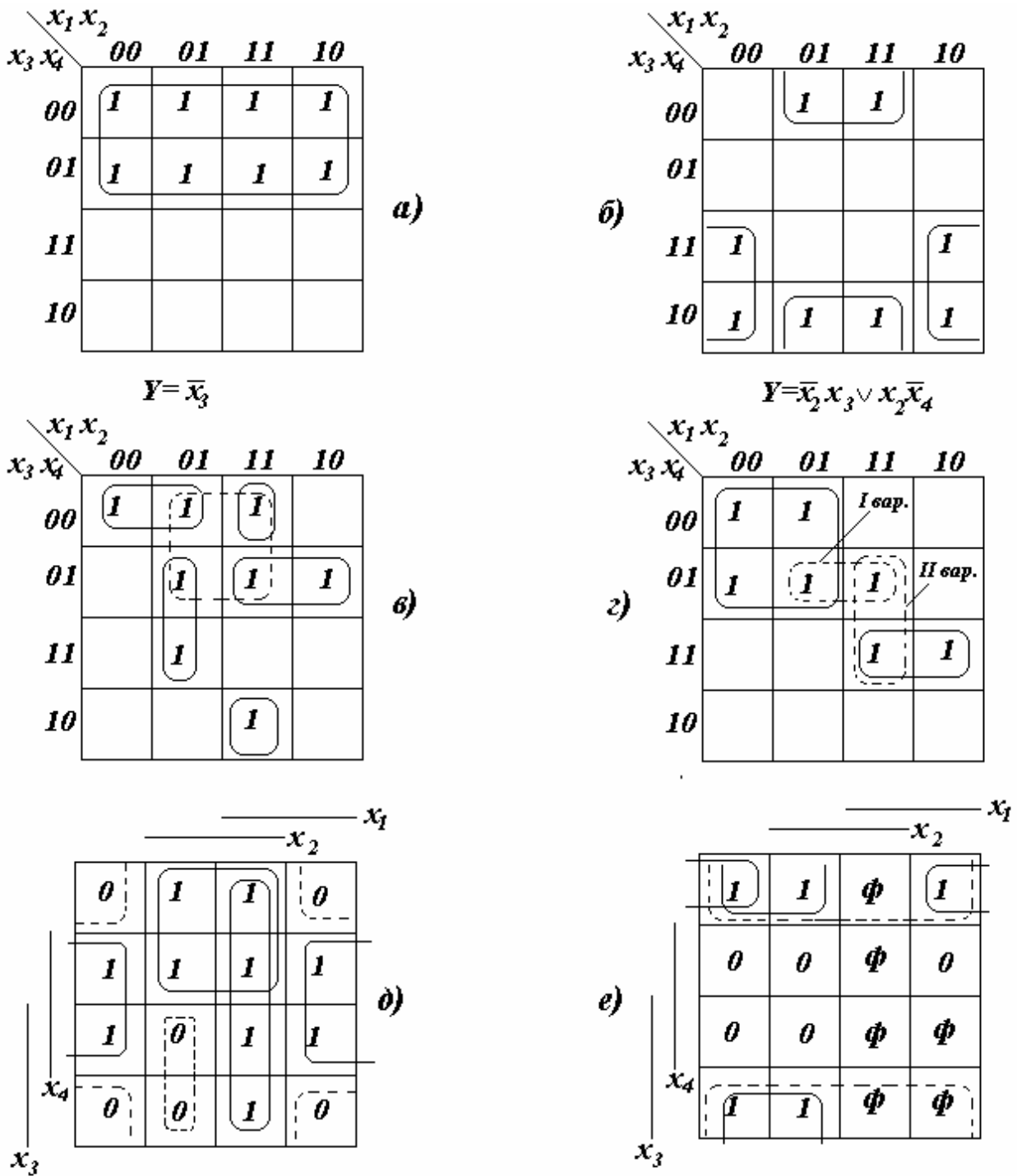


Рис. 1. 10

Сравнение этих вариантов показывает, что они приблизительно равноценны по количеству членов в структурных формулах.

Иногда удобнее при графическом методе минимизации вместо группирования единиц произвести группирование нулей. Это лучше рассмотреть на конкретном примере. Предположим, что необходимо минимизировать функцию

$$\text{вида: } f(x_1, x_2, x_3, x_4) = \sum(1, 3, 4, 5, 9, 11, 12, 13, 14, 15). \quad (1.38)$$

$$\text{Ее отрицание: } \overline{f(x_1, x_2, x_3, x_4)} = \sum(0, 2, 6, 7, 8, 10). \quad (1.39)$$

На карте Карно (рис. 1. 10, д) наглядно виден принцип минимизации группированием нулей. Сначала сгруппируем единицы и обведем контуры сплошной линией. Затем нули – штриховой. Единиц – 10, а нулей – 6. Количество нулей, как правило, еще не является решающим в пользу группирования нулей, но проверить такую возможность, конечно, необходимо в каждом конкретном случае. При группировании единиц получаем минимальную форму:

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 \vee \overline{x_2} x_4 \vee \overline{x_2} x_3 \quad (1.40)$$

$$\text{При группировании нулей: } \overline{f(x_1, x_2, x_3, x_4)} = \overline{x_2} \overline{x_4} \vee \overline{x_1} x_2 x_3. \quad (1.41)$$

$$\begin{aligned} \text{Операция инверсии над (1.41) дает: } \overline{\overline{f(x_1, x_2, x_3, x_4)}} &= \overline{\overline{x_2} \overline{x_4} \vee \overline{x_1} x_2 x_3} = \\ &= \overline{\overline{x_2} \overline{x_4}} \overline{\overline{x_1} x_2 x_3} = (x_2 \vee x_4)(x_1 \vee \overline{x_2} \vee \overline{x_3}). \end{aligned} \quad (1.42)$$

Можно показать, что (1.40) и (1.42) равносильны и могут быть преобразованы одно в другое.

Минимизация недоопределенных функций. К недоопределенным, или не полностью определенным функциям, относятся функции, содержащие факультативные условия, когда некоторые наборы аргументов запрещены или невозможны. Такую функцию можно произвольно доопределить, установив ее значение «0» или «1» на запрещенных наборах по своему усмотрению. Это обычно делается на карте Карно при выполнении минимизации. Факультативные условия обозначаются какой-либо буквой, чаще всего X или Φ . Значение $\Phi=1$ проставляется в тех квадратах, которые могут входить в какие-либо объединения единиц, упрощая процедуру склеивания.

Поясним сказанное на конкретном примере. Составим структурную формулу устройства, которое реагирует только на четные десятичные цифры из всех цифр от 0 до 9 на входе в виде четырехразрядного десятичного кода. На выходе должен появиться сигнал «1», когда на четырех входах будут сигналы, соответствующие наборам 0, 2, 4, 6, 8, а наборы 10, 11, 12, 13, 14, 15 никогда не будут встречаться. Булева функция такого устройства будет иметь вид:

$$F = \sum_{об}(0, 2, 4, 6, 8) \vee \sum_{\phi}(10, 11, 12, 13, 14, 15), \quad (1.43)$$

где: об – обязательные условия; ϕ – факультативные условия.

Нанесем (1.43) на карту Карно (рис. 1. 10, е). Минимизация без учета факультативных условий, если считать на всех запрещенных наборах $\phi=0$, дает:

$$F = \overline{x_1} \overline{x_4} \vee \overline{x_2} \overline{x_3} \overline{x_4}. \quad (1.44)$$

$$\text{Если же принять на 10-м, 12-м и 14-м наборах } \phi=1, \text{ то } F = \overline{x_4}. \quad (1.45)$$

Этот результат означает, что для чисел от 0 до 15 признаком четности будет служить $x_4=0$, т. е. отсутствие единицы в младшем разряде двоичного изображения этих чисел.

1. 9. Метод минимизации Квайна – Мак Класки

Метод минимизации логических функций с помощью карт Карно становится неудобным при количестве аргументов больше 5 ... 6. В этом случае удобнее пользоваться методом Квайна – Мак Класки. Алгоритм минимизации основан на использовании двух операций: неполного склеивания $Ax \vee A\bar{x} = Ax \vee A\bar{x} \vee A$ и поглощения $A \vee Ax = A \vee A\bar{x} = A$.

Если функция представлена в СДНФ и, если в ней произвести все возможные склеивания, а затем все поглощения, то получим более простое выражение, которое называется сокращенной ДНФ.

Для дальнейших рассуждений важно ввести понятие *импликант* функции (конъюнкция переменных с отрицанием или без него), изъятие которого из структурной формулы приводит к появлению тупиковых вариантов ДНФ. Процедура минимизации при этом имеет много общего с методом Карно и заключается в следующих действиях.

1. Структурная формула логического устройства, представленная в СДНФ, записывается в виде столбика двоичных чисел, каждое из которых (минтерм в методе Карно) здесь называется *импликантом*, т. е. \bar{x}_i заменяется на 0, а x_i – на 1.

1. Двоичные числа разбиваются на группы по количеству единиц (нулевая группа состоит из одного числа). Группы отделяются одна от другой и образуют колонку.

2. Производится склеивание чисел колонки. Результат пишется в следующей колонке, а при склеивании необходимо учитывать целый ряд условий:

- склеиваются только числа из соседних групп;
- склеиваются только числа, отличающиеся в одном разряде; например при склеивании $(a0b)$ с $(a1b)$ получается число $(a-b)$, которое записывается в новую колонку, а около выражений предыдущей колонки, для которых выполнено склеивание, ставится какая-либо метка (например, знак \vee);
- каждое число может участвовать в склеивании произвольное количество раз, но необходимо перебрать все возможные варианты склеиваний; повторяющиеся склеивания не выписываются.

4. Количество единиц в последующих колонках должно постепенно уменьшаться, но следует обратить внимание на то, что склеиваются выражения, в которых черточки стоят в одних и тех же позициях. Например, из $(11 - 0 -)$ и $(10 - 0 -)$ получаем $(1 - - 0 -)$.

5. Образование новых колонок прекращается, если в последней колонке нельзя выполнить ни одного склеивания.

6. Выражения, для которых оказалось невозможным выполнить склеивание, заменяются буквенными операциями, обратными исходным, при этом черточки опускаются. Например, выражение $(10 - 0 -)$ заменяется на $x_1 \bar{x}_2 \bar{x}_4$.

7. Буквенные выражения – это простые импликанты минимизируемой функции, а их дизъюнкция – это сокращенная ДНФ.

8. Составляется таблица импликантов, из которой выбираются минимальные тупиковые формы.

Пример минимизации по Квайну–Мак Класки. Исходная функция четырех переменных: $Y = \overline{x_1} \overline{x_2} \overline{x_3} x_4 \vee x_1 \overline{x_2} \overline{x_3} \overline{x_4} \vee x_1 \overline{x_2} x_3 \overline{x_4} \vee x_1 x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4 \vee x_1 x_2 \overline{x_3} x_4 \vee x_1 \overline{x_2} x_3 x_4 \vee x_1 x_2 x_3 \overline{x_4} \vee x_1 x_2 \overline{x_3} \overline{x_4} \vee x_1 x_2 x_3 x_4 \vee x_1 \overline{x_2} \overline{x_3} \overline{x_4} \vee x_1 \overline{x_2} x_3 \overline{x_4} \vee x_1 x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4 \vee \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} \overline{x_2} x_3 \overline{x_4} \vee \overline{x_1} \overline{x_2} x_3 x_4 \vee \overline{x_1} x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} x_4 \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4$. (1.46)

Выпишем в столбик в виде двоичных чисел импликанты, входящие в выражение (1.46), как рекомендовано в п.1, и произведем разбиение на группы:

0 0 0 1√	0 0 0 1√	0 – 0 1√	– – 0 1
0 1 0 0√	0 1 0 0√	– 0 0 1√	0 1 – –
0 1 0 1√	1 0 0 0√	0 1 0 –√	– 1 0 –
0 1 1 0√	-----	0 1 – 0√	1 – 0 –
0 1 1 1√	0 1 0 1√	– 1 0 0√	
1 0 0 0√	0 1 1 0√	1 0 0 –√	
1 0 0 1√	1 0 0 1√	1 – 0 0√	
1 1 0 0√	1 1 0 0√	-----	
1 1 0 1√	-----	0 1 – 1√	
	0 1 1 1√	– 1 0 1√	
	1 1 0 1√	0 1 1 –√	
		1 – 0 1√	
		1 1 0 –√	

Последний, самый короткий столбик, позволяет получить сокращенную ДНФ функции в виде: $Y = \overline{x_3} x_4 \vee x_1 x_2 \vee x_2 x_3 \vee x_1 x_3$. (1.47)

Таблица импликантов, соответствующая (1.47), представлена в табл. 1. 6.

Таблица 1. 6

Простые импликанты	Члены канонической суммы								
	0001	0100	0101	0110	0111	1000	1001	1100	1101
– – 0 1 *	+		+				+		+
0 1 – – *		+	+	+	+				
– 1 0 –		+	+					+	+
1 – 0 – *						+	+	+	+

В табл. 1. 6 отмечены звездочками те простые импликанты, которые обязательно должны войти в любую тупиковую форму (столбцы с единственным крестиком) – это *существенные простые импликанты*. Не отмеченный звездочкой простой импликант не является существенным и не должен входить в минимальную тупиковую форму, т. е. он – лишний. Минимальная тупиковая форма, полученная из таблицы импликант, представлена в виде (1.48):

$$Y = \overline{x_3} x_4 \vee x_1 x_2 \vee x_1 x_3. \tag{1.48}$$

Вопросы и задания для самоконтроля

1. Сформулируйте теорему де Моргана и приведите пример ее использования.
2. Что такое мажоритарный элемент «2 из 3-х»?
3. Приведите пример задания логической функции в первой стандартной форме (СДНФ).
4. Чем отличается запись логической функции в форме СКНФ?
5. Какая система логических элементов называется функционально полной?
6. Как перейти от логической функции к логической схеме?
7. Какой логический элемент называется «универсальным»?
8. Какие типовые приемы можно использовать при алгебраической минимизации логических функций?
9. В чем заключается графический метод минимизации Карно – Вейча?
10. В чем особенность минимизации по Квайну – Мак Класки?

2. ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ

2. 1. Понятие о системах счисления. Представление целых и дробных чисел

В цифровых устройствах находят применение только *позиционные* системы счисления. В этих системах числа представляются последовательностью цифр разрядов, разделенных запятой или точкой на две группы: группу разрядов целой части числа и группу дробной части

$$\Lambda a_2 a_1 a_0, a_{-1} a_{-2} K, \quad (2.1)$$

где $a_0, a_1, a_2 \dots$ - цифры нулевого, первого и т. д. разрядов целой части числа;

$a_{-1}, a_{-2} K$ - цифры первого, второго и т. д. разрядов дробной части.

Единице каждого разряда приписывается определенный *вес* p^k , где p - *основание системы счисления*, k – номер разряда, равный индексу при буквах, изображающих цифры разрядов.

Общий алгоритм представления числа в какой-либо системе счисления определяется равенством

$$N = \dots a_2 p^2 + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + a_{-2} p^{-2}. \quad (2.2)$$

Десятичная система счисления. Для обычной десятичной системы счисления, знакомой всем по повседневной жизни используется набор цифр разрядов $0, 1, 2, \dots, 9$; $p = 10$.

Запись числа $729,324_{10}$. Индекс указывает основание системы счисления. Иногда для этой цели используется буква D. Допускается не указывать основание системы. Тогда по умолчанию подразумевается, что применяется десятичная система счисления:

$$\begin{array}{cccccc}
 7 & 2 & 9 & , & 3 & 2 & 4D \\
 \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\
 & \text{Весовые коэффициенты} & & & & & \\
 10^2 & 10^1 & 10^0 & & 10^{-1} & 10^{-2} & 10^{-3}
 \end{array}$$

Двоичная система счисления. Основание системы счисления $p = 2$. Применяется только два символа 0 и 1. Эти двоичные цифры называются битами (сокращение от английских слов binary digit – двоичная цифра). Любое число может быть записано комбинацией этих символов. Например, запись $111011,101_2$ соответствует согласно (2.2) в десятичной системе счисления следующему числу:

$$\begin{array}{cccccc}
 1 & 1 & 0 & 1 & 1 & , & 1 & 0 & 1B \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\
 & \text{Весовые коэффициенты} & & & & & & & \\
 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} & 2^{-3}
 \end{array}$$

Восьмеричная система счисления. Основание системы счисления $p = 8$. Цифры разрядов используют 8 символов: 0, 1, 2, 3, 4, 5, 6, 7. Символы 8 и 9 не используются и поэтому не должны встречаться в записи. Числа представляются в соответствии с общим алгоритмом (2.2), а вместо индекса 8 иногда применяется латинская буква Q.

Например, записи числа 1357_8 соответствует запись числа в десятичной системе:

$$\begin{array}{cccc}
 1 & 3 & 5 & 7 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 & \text{Весовые коэффициенты} & & \\
 8^3 & 8^2 & 8^1 & 8^0
 \end{array}$$

Шестнадцатеричная система счисления. Основание системы счисления $p = 16$. В этой системе счисления цифры разрядов записываются 16 – ю символами. Первые десять символов записываются десятичными арабскими цифрами: 0 ... 9. Последние шесть символов – латинскими прописными буквами A, B, C, D, E, F, которые соответствуют числам от $10 = A$ до $15 = F$. Соответственно, $B = 11$, $C = 12$, $D = 13$ и $E = 14$.

При записи числа в 16 – ричной системе счисления в конце ставится индекс 16 или латинская (чаще *прописная* буква H , но допускается и применение *строчной* буквы h):

$$A43H = A \quad 4 \quad 3 = 10 \cdot 16^2 + 4 \cdot 16^1 + 3 \cdot 10^0 = 2627_{10} .$$

$$\downarrow \quad \downarrow \quad \downarrow$$

Весовые коэффициенты
 $16^2 \quad 16^1 \quad 16^0$

Применение BCD – кода при кодировании десятичных чисел. Следует отметить, что двоичная система наиболее удобна для выполнения арифметических операций и хранения n – разрядных чисел, т. к. для запоминания цифры каждого разряда используются устройства с двумя устойчивыми состояниями – триггеры.

При хранении десятичных чисел каждая цифра десятичного числа предварительно представляется в десятичной форме. Такая форма представления носит название двоично–кодированной десятичной системы, или *BCD – кода*.

Например, число $765,93_{10}$ в этой системе кодирования будет иметь вид:
 $765,93D = \underbrace{0111}_7 \underbrace{0110}_6 \underbrace{0101}_5 , \underbrace{1001}_9 \underbrace{0011}_3 BCD$

Существует несколько способов двоичного кодирования десятичных цифр. *BCD – код* иначе называется код 8421 (название составлено из весовых коэффициентов разрядов в двоичной системе). Другие, наиболее употребительные способы кодирования приведены в табл. 2. 1.

Таблица 2.1

Десятичные числа	Код 8421 (BCD-код)	Код с избытком 3	Код Айкена 2421	Код 2 из 5-ти	Код Джонсона	Код Грея 7421
0	0000	0011	0000	01100	00000	0000
1	0001	0100	0001	11000	10000	0001
2	0010	0101	0010	10100	11000	0011
3	0011	0110	0011	10010	11100	0010
4	0100	0111	0100	01010	11110	0110
5	0101	1000	1011	00110	11111	0111
6	0110	1001	1100	10001	01111	0101
7	0111	1010	1101	01001	0111	0100
8	1000	1011	1110	00101	00011	1100
9	1001	1100	1111	00011	00001	1101

Особый интерес из приведенных в табл. 2.1 кодов представляет код Грея, т. е. код с обменом единицей: при последовательном переходе от одной цифры этого кода к другой всегда изменяется только один из двоичных разрядов. Этот код применяется при минимизации логических функций с помощью карт Карно, о чем речь уже шла в предыдущей главе, а также будет встречаться в дальнейшем.

2. 2. Перевод чисел из одной системы счисления в другую

Наиболее просто производится перевод из 8 – ричной или 16 – ричной системы счисления в двоичную. Например, при переводе числа из 8 – ричной системы

$$627,23_8 = \underbrace{110}_6 \underbrace{010}_2 \underbrace{111}_7 , \underbrace{010}_2 \underbrace{011}_3_2$$

необходимо каждую цифру заменить двоичным эквивалентом в виде *триады* – (трех бит).

А при переводе числа из 16 – ричной системы в 2 – ичную необходимо каждую цифру заменить двоичным представлением из 4 - х бит – *тетрадой* :

$$AC5, F6_{16} = \underbrace{1010}_A \underbrace{1100}_C \underbrace{0101}_5 , \underbrace{1111}_F \underbrace{0110}_6_2 .$$

В случае необходимости обратного перевода из 2 - ичной системы в 8 – ричную или 16 – ричную производится разбиение двоичного представления на *триады* и *тетрады* , как было показано выше.

Несколько сложнее переводятся числа из двоичной системы счисления в десятичную. Использование общего алгоритма (2.1) приводит к большому количеству вычислений. На практике широко применяется метод удвоения, который понятен из рис. 2.1, *а*.

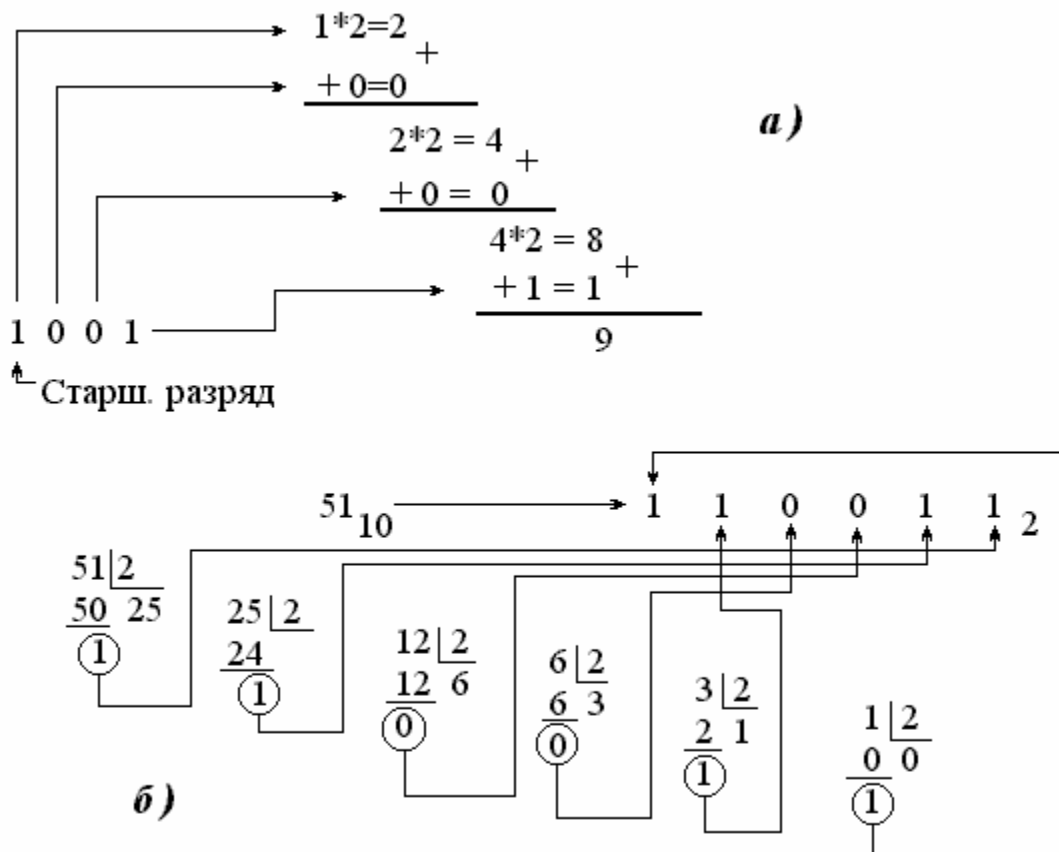


Рис. 2. 1

На рис. 2.1, б представлен способ быстрого пересчета десятичного числа в двоичное методом *деления*.

При пересчете десятичного числа в 8 – ричное или 16 – ричное целая часть числа преобразуется точно, а дробная – приблизительно. При этом при пересчете целой части числа производится операция *деления* на основание системы, а при пересчете дробной части операция – *умножения*.

Проиллюстрируем это примерами. Предположим, что необходимо преобразовать число 751_{10} в 8 – ричную систему. Используем метод деления, реализующий схему Горнера [3] (рис. 2.2, а).

Как видно из рис. 2.2, б, порядок действий при переводе целого числа из десятичной системы счисления в 16 – ричную остается аналогичным

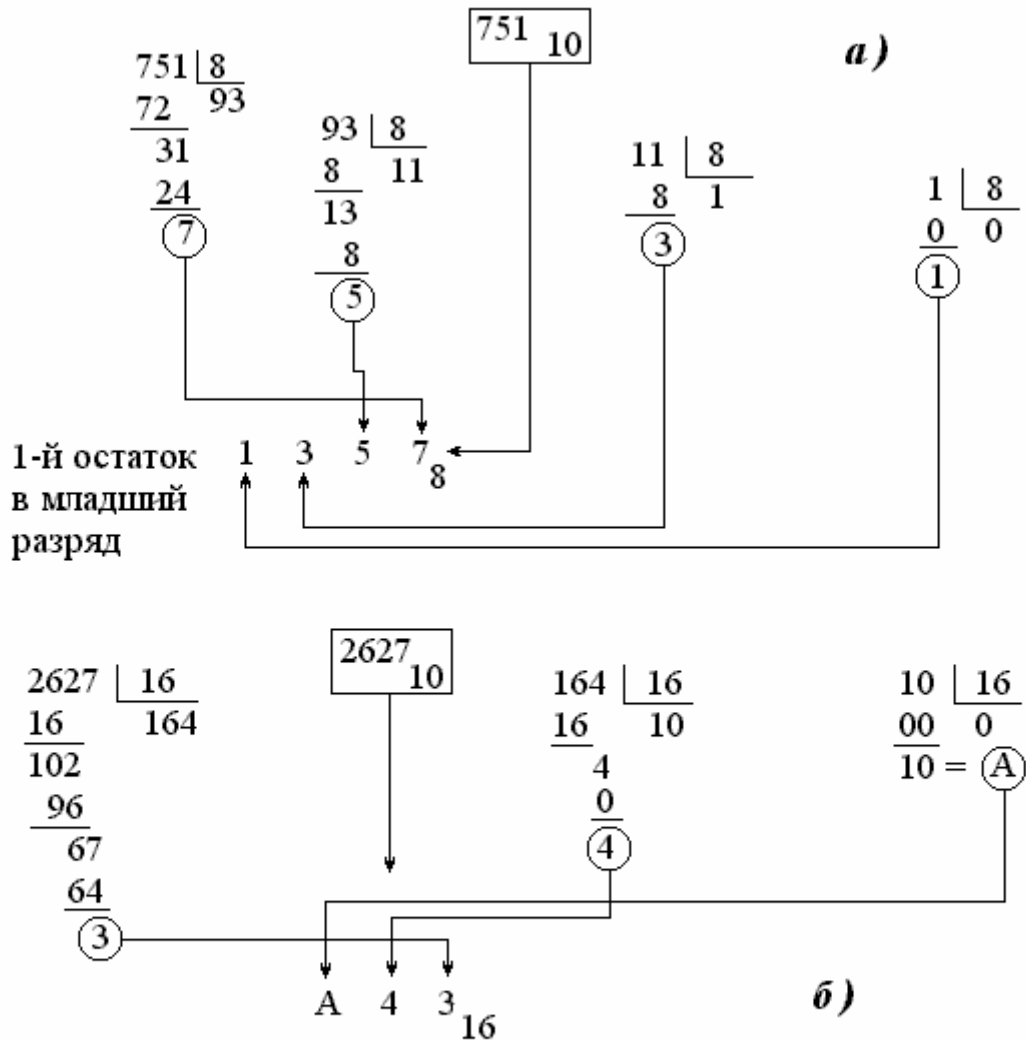


Рис. 2. 2

Пример перевода дробной части числа из десятичной системы счисления в шестнадцатеричную показан на рис. 2.3. на этом рисунке показана процедура перевода числа $0,7821502_{10}$ в шестнадцатеричную систему счисления.

Умножение на основание системы производится в столбик. Следует отметить, что преобразование дробной части числа в другую систему счисления можно выполнить, как это видно из рис. 2.3, с любой, наперед заданной точностью.

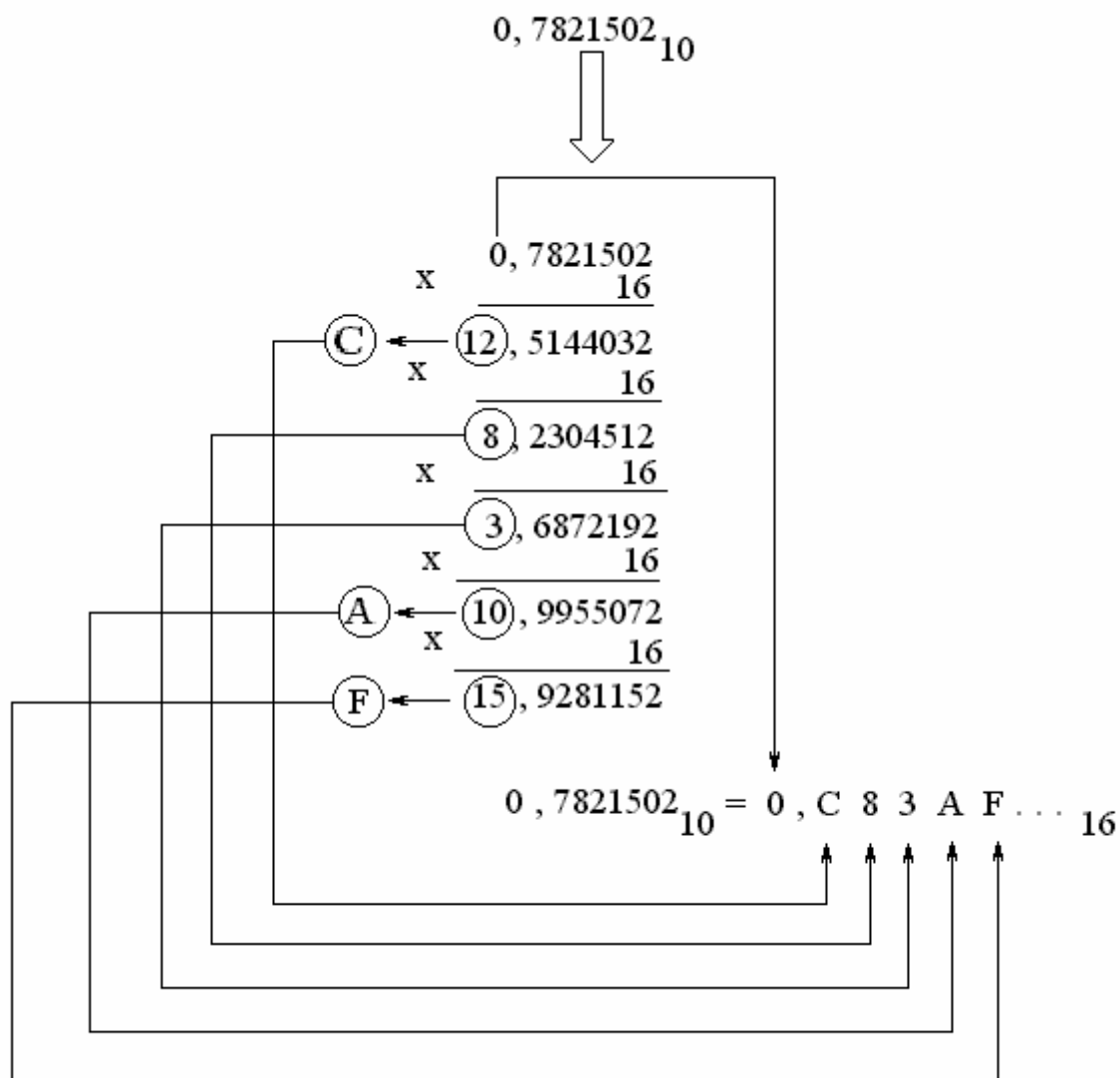


Рис. 2.3

2.3. Формы представления чисел с фиксированной и плавающей точкой

Форма представления числа с фиксированной точкой. Эта форма использует один разряд в качестве знакового в кодированном виде: 0 – положительный знак, 1 – отрицательный знак. Остальные разряды хранят абсолютное значение числа.

Точка, разделяющая целую часть числа и дробную, занимает фиксированное положение или перед старшим разрядом в дробном числе, или после младшего разряда в целой части. Это предполагает, что целая и дробная части числа находятся в различных ячейках памяти. А это приводит к низкой точности представления чисел.

Например, число -0.110_2 следующим образом разместится в элементах ячейки памяти с разрядностью 1 байт (рис. 2. 4, а) .



Рис. 2. 4

На рис. 2. 4, б изображена целая часть числа $0F_{16}$.

Числа с плавающей точкой. В вычислительной математике дробь обычно представляется в виде целого числа, умноженного на основание в степени с отрицательным показателем., который называется порядком. В такой системе представления чисел любое число имеет вид:

$$\text{ЧИСЛО} = \text{МАНТИССА} \times \text{ОСНОВАНИЕ}^{\text{ПОРЯДОК}}$$

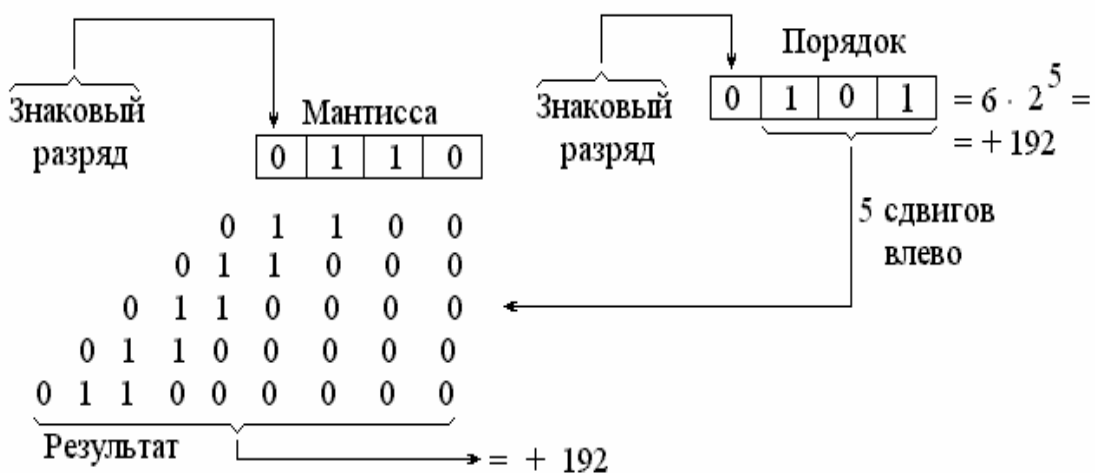


Рис. 2. 5

При этом мантисса – целое число (положительное или отрицательное). Мантисса и порядок имеют отдельные знаковые разряды, и в системе представления с плавающей точкой (запятой) мантисса выражается с помощью одной группы битов, а порядок – с помощью другой, что позволяет расширить диапазон чисел, но при этом уменьшается точность числа, которая определяется мантиссой.

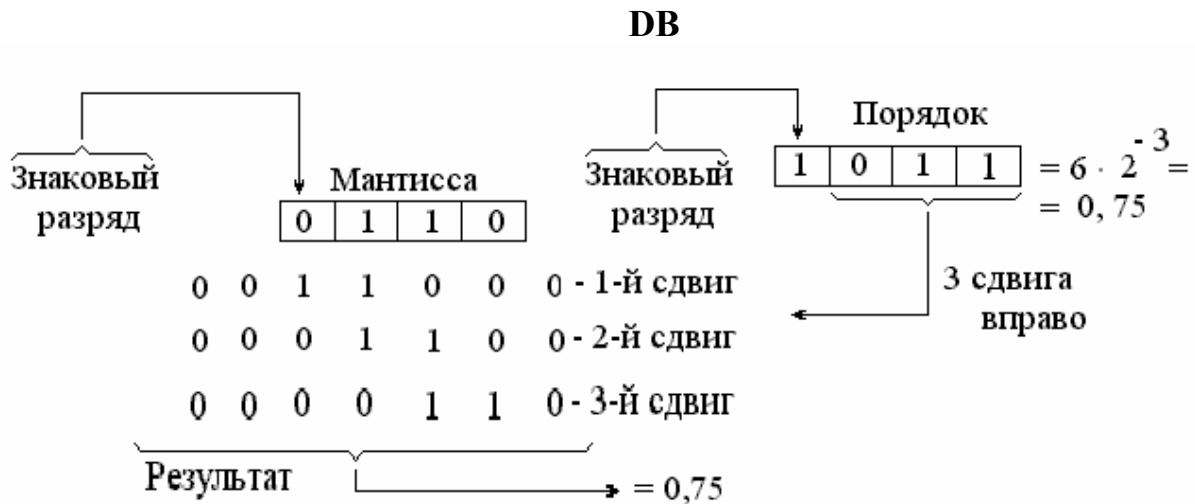


Рис. 2. 6

Вопросы и задания для самоконтроля

1. Что такое позиционная система счисления?
2. Напишите общий алгоритм представления числа в какой-либо системе счисления.
3. В чем особенность двоичной системы счисления?
4. Какие существуют способы двоичного кодирования десятичных цифр?
5. В чем особенность 16-ричной системы счисления?
6. В чем особенность применения BCD-кода при кодировании десятичных чисел?
7. Как производится перевод из 16-ричной системы счисления в двоичную с помощью BCD-кода?
8. Как осуществить быстрый перевод десятичного числа в двоичное или 16-ричное методом деления (схема Горнера)?
9. Дайте понятие о формах представления числа с фиксированной и плавающей точкой.
10. Что такое дополнительный код. Как он используется в арифметических операциях?

3. КОМБИНАЦИОННЫЕ УСТРОЙСТВА

3.1. Вводные понятия

Комбинационное устройство – это логическая схема с m входами ($m \geq 1$) и n выходами ($n \geq 1$), у которой состояния выходов лишь определяются состояниями входов в тот же момент времени. Последовательность действий при синтезе комбинационного устройства обычно заключается в следующем. На основе заданных условий функционирования (словесного описания, таблицы истинности) составляется структурная формула устройства и производится ее минимизация по правилам, описанным в предыдущей главе.

3.2. Устройство неравнозначности

Это устройство с двумя входами x_1, x_2 и одним выходом Y , реализующим логическую функцию, называемую ИСКЛЮЧАЮЩЕЕ ИЛИ: $Y = 1$ только при несовпадении информационных значений входных сигналов в соответствии с таблицей истинности (табл. 7).

Таблица 3.1

x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

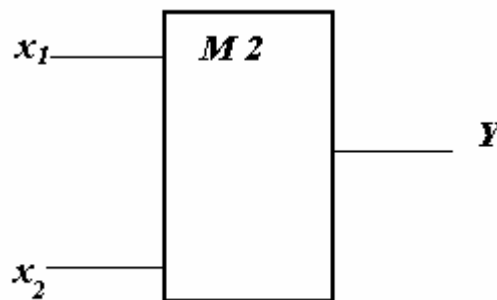


Рис. 3.1

Условное изображение этого объекта на схемах функциональных представлено на рис. 3.1. Эта функция имеет специальное обозначение, как показано в выражении (3.1): $Y = x_1 \oplus x_2$ (3.1)

и чаще называется «сумма по модулю 2». По условиям срабатывания, т. е. для

наборов, где $Y = 1$ можно записать $Y = x_1 \bar{x}_2 \vee \bar{x}_1 x_2$.

$$(3.2)$$

Таблица 3.2

x_2	x_1	Y
0	0	1
0	1	0
1	0	0
1	1	1

Устройство, работающее по структурной формуле (3.1) можно реализовать в базисе И – ИЛИ –НЕ из пяти логических элементов. Путем несложных преобразований можно уменьшить число элементов на единицу. Т. к. $x_1 \bar{x}_1 = 0$; $x_2 \bar{x}_2 = 0$, то преобразуем (3.1) к виду:

$$Y = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 \bar{x}_1 \vee x_2 \bar{x}_2 = (x_1 \vee x_2)(\bar{x}_1 \bar{x}_2). (3.3)$$

Логическая схема, выполняющая эту функцию, представлена на рис. 3. 2, а. Нетрудно показать, что для этой функции существует ряд дополнительных тождеств, которые полезно использовать для упрощения логических выражений:

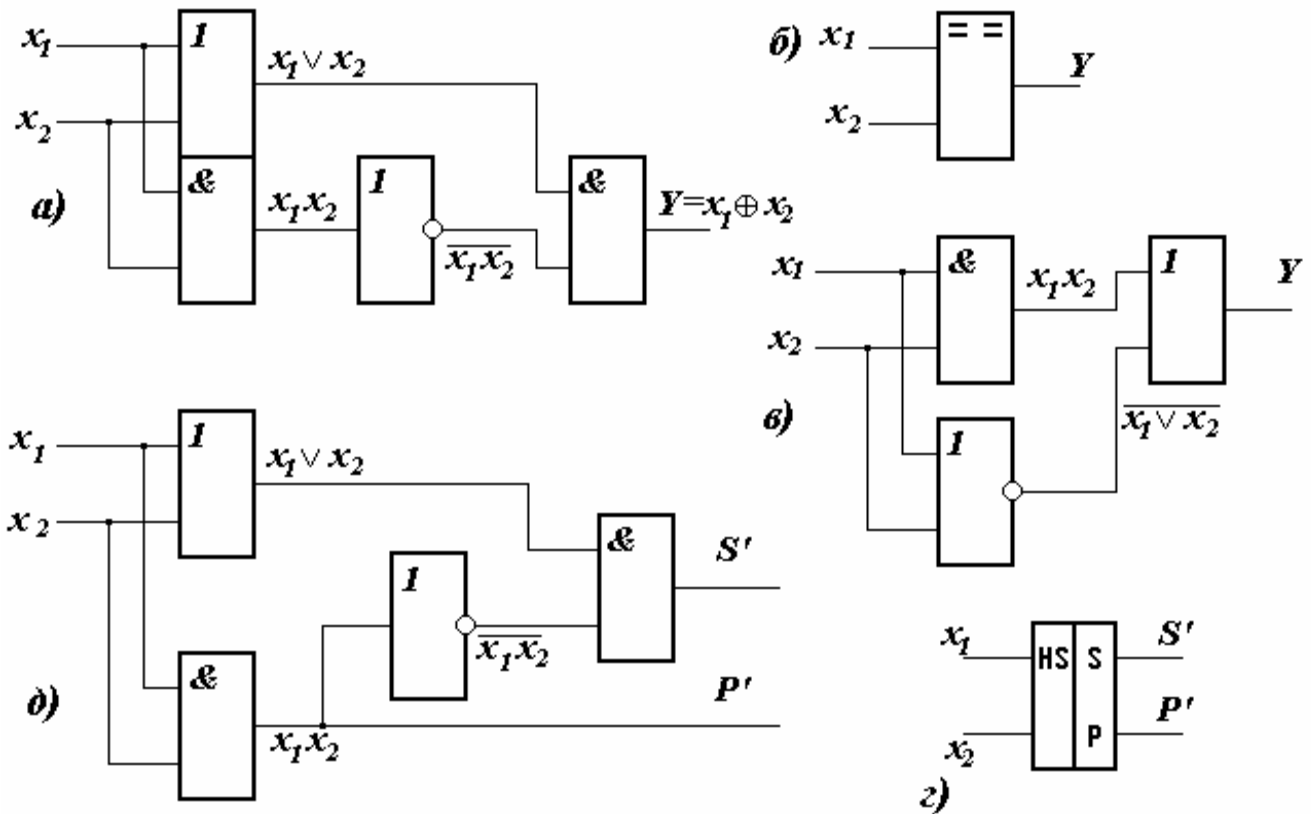


Рис. 3. 2

$$\overline{x_1 \oplus x_2} = \overline{x_1 x_2} \vee x_1 x_2; \quad (3.4)$$

$$x \oplus x = 0; \quad x \oplus \overline{x} = 1; \quad (3.5)$$

$$1 \oplus 1 = 0; \quad 1 \oplus 0 = 1; \quad (3.6)$$

$$0 \oplus 0 = 0; \quad x \oplus 1 = \overline{x}; \quad (3.7)$$

$$x \oplus 0 = x. \quad (3.8)$$

$$\text{Если } x_1 \oplus x_2 = Y, \text{ то } x_1 \oplus Y = x_2; \quad x_2 \oplus Y = x_1. \quad (3.9)$$

3.3. Устройство равнозначности

Так называется устройство с двумя входами x_1 и x_2 и одним выходом Y , условия работы которого таковы: сигнал на выходе имеет место только при совпадении информационных значений входных сигналов. Таблица истинности, соответствующая этому словесному описанию, представлена в виде табл. 8, а условное графическое изображение элемента на схемах – на рис. 3. 2, б .

На основании табл. 8 можно получить структурную формулу:

$$Y = x_1 x_2 \vee \overline{x_1} \overline{x_2}, \quad (3.10)$$

а, применяя правило де Моргана, (3.10) можно преобразовать к виду (3.11):

$$Y = (x_1 x_2) \vee (\overline{x_1} \vee \overline{x_2}). \quad (3.11)$$

Принципиальная схема устройства равнозначности представлена на рис. 3. 2, в.

3. 4. Полусумматор

Полусумматором называется устройство для сложения двух одноразрядных двоичных чисел. В результате сложения в общем случае наряду с суммой может получиться перенос, т. е. это устройство имеет два входа и два выхода. Таблица истинности в соответствии с приведенным словесным описанием

Таблица 3.3 представлена в виде табл. 9, а условное графическое

x_1	x_2	S'	P'
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

изображение – рис. 3. 2, г. На этом рисунке входные сигналы – x_1, x_2 ; S' - сигнал суммы, а выходной сигнал P' - перенос. Из табл. 9 можно видеть, что булева функция S' полусумматора совпадает с функцией Y в случае сумматора по модулю 2 и для построения логической схемы

полусумматора можно воспользоваться уже рассмотренными ранее схемами, дополнив их схемами для формирования переноса. Знаки «штрих», которые стоят около выходов S' и P' , ставятся для отличия полусумматора от полного сумматора, о котором речь будет ниже, в следующем параграфе.

Для экономии числа логических элементов структурную формулу для суммы S' составляют «по нулям», а для переноса – «по единицам», как показано в (3.12) и (3.13): $S' = (x_1 \vee x_2)(\overline{x_1} \vee \overline{x_2}) = (x_1 \vee x_2) \overline{x_1 x_2}$; (3.12)

$$P' = x_1 x_2. \quad (3.13)$$

В этом случае перенос будет составлять часть структурной формулы суммы, что видно из принципиальной схемы полусумматора (рис. 3. 2, д).

3. 5. Полный сумматор

Полный сумматор – это устройство, предназначенное для сложения трех одноразрядных двоичных чисел x_1, x_2, x_3 . Такая задача часто возникает при поразрядном сложении двоичных чисел, когда третье слагаемое – перенос из предыдущего младшего разряда. Условное графическое изображение устройства представлено на рис. 3. 3, а, а принципиальная схема – на рис. 3. 3, б.

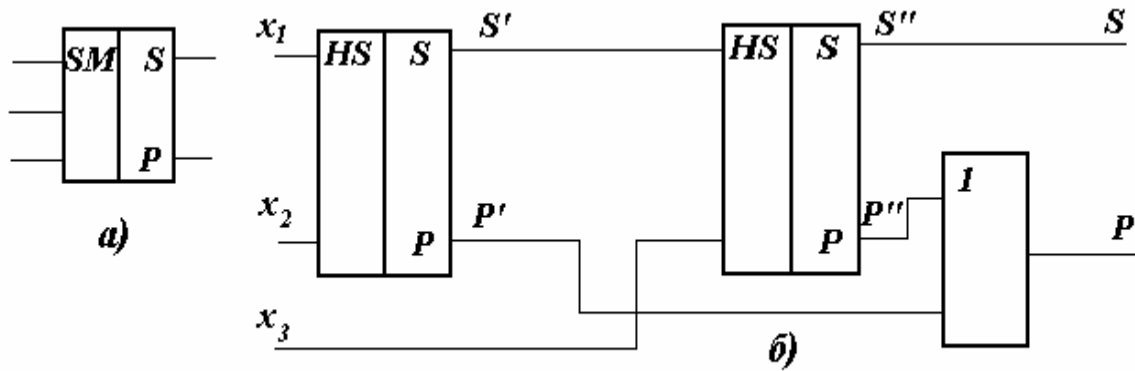


Рис. 3. 3

Как видно из рис. 3. 3, б, полный сумматор строится из двух полусумматоров и одного элемента, выполняющего операцию дизъюнкции двух переносов, возникающих в каждом полусумматоре. Поэтому полный сумматор также имеет два выхода: S – сумма и P – перенос.

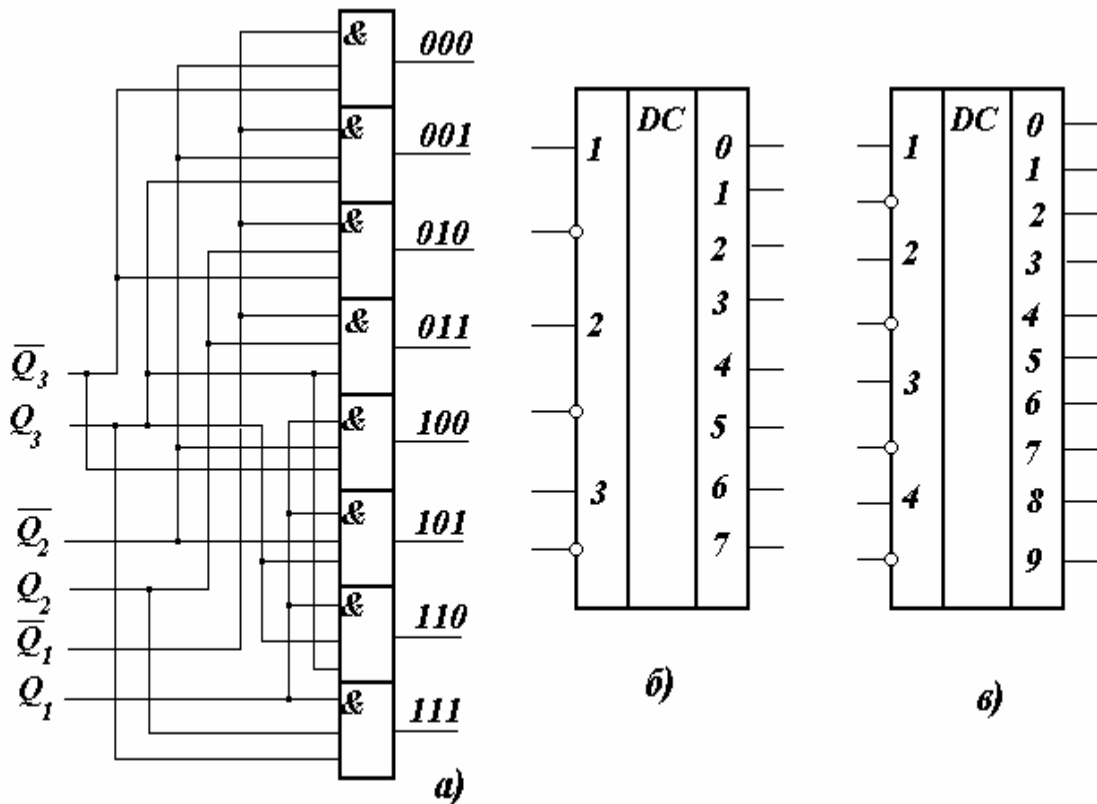


Рис. 3. 4

3. 6. Дешифраторы

Дешифратор – это устройство, предназначенное для распознавания кодовых комбинаций. Дешифратор имеет число входов, равное значности принимаемого кода и число выходов, равное возможному числу кодовых комбинаций.

ций. Дешифратор вырабатывает единичный сигнал на требуемом выходе только тогда, когда на вход поступает код числа, соответствующего этому выходу.

Как видно из приведенного определения, дешифратор – это декодирующее устройство (*декодер*). Его принципиальная схема представлена на рис. 3.4, *а*, а условное графическое изображение – на рис. 3.4, *б*.

Как видно из рис. 3.4, *а*, полный дешифратор должен иметь 2^n элементов И, каждый из которых имеет n входов. В код числа могут входить как единицы, так и нули, а схема И для нормального срабатывания требует единичные сигналы на оба входа. Поэтому для индикации кода приходится использовать единицы с инверсного выхода разряда, и полный дешифратор будет иметь $2n$ входных шин (n для прямых и n для инверсных значений разрядных чисел).

Так, в случае применения трехразрядного кода (рис. 3.4, *б*) у дешифратора шесть входных шин (три прямых и три инверсных). В этом коде может быть записано восемь чисел (0 ... 7). Если необходимо записать 10 цифр (0 ... 9), то потребуются схемы на четыре входа, и входная информация должна содержать 4 двоичных разряда в прямом коде и столько же в инверсном (рис. 3.4, *в*).

3.7. Мультиплексоры и демультиплексоры

Мультиплексор – это комбинационная схема, имеющая $m + 2^m$ входов и один выход, где m – число адресных входов, а 2^m – число информационных входов мультиплексора. Каждому адресу в двоичном коде соответствует свой информационный вход, сигнал с которого при этом адресе приходит на выход.

Основное назначение мультиплексора – это коммутация 2^m сигналов со входа на один общий выход. При этом будет выполняться логическая функция согласно выражению (3.14) [3]:

$$f = \bigvee_{j=0}^{2^m-1} A_j K_j(v), \quad (3.14)$$

где $v = x_1 \dots x_m$; x_p – адресные сигналы; $p = 1, 2 \dots m$; A_j – входные информационные сигналы мультиплексора.

На рис. 3.5, *а* изображен четырехканальный мультиплексор, в котором использованы логические элементы ИЛИ – НЕ и И – ИЛИ (адресные входы x_1 и x_2). Промышленность выпускает мультиплексоры с различным числом адресных входов $m=2, 3$ и 4 [11]. В случае, когда количество адресных входов $m=2$, мультиплексоры выпускаются вдвоенные, четырехканальные, со стробированием, условное графическое изображение которых приведено на рис. 3.5, *б* (микросхема 155КП2). Стробирующие входы \bar{V}_1 и \bar{V}_2 используются для построения коммутаторов с $k \cdot 2^m$ информационными входами ($k = 2, 3, 4 \dots$) на основе 2^m – канальных мультиплексоров.

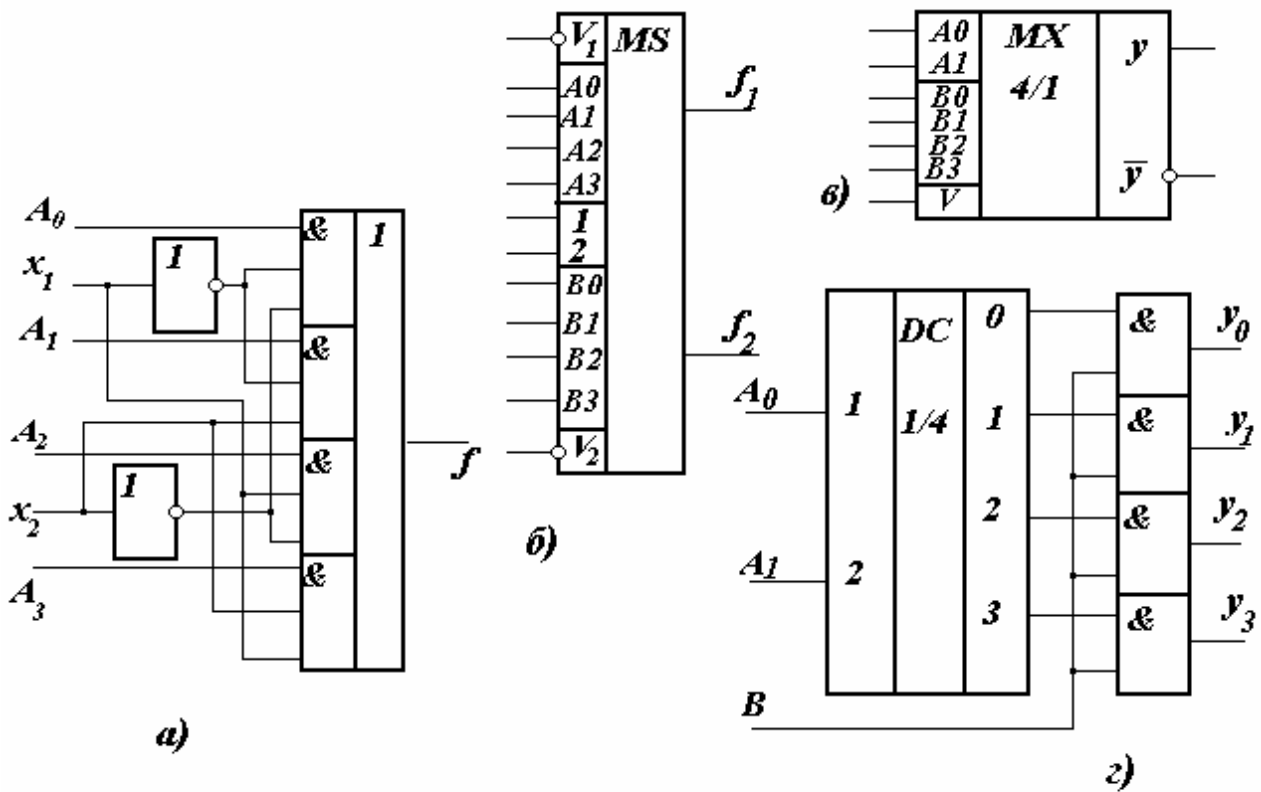


Рис. 3. 5

Демультимплексоры выполняют функцию, обратную функции мультиплектора, т. е. коммутируют один входной сигнал на 2^m выходов. В качестве демультимплексоров могут быть использованы мультиплексоры, выполненные на основе аналоговых двунаправленных ключей. Например, в качестве демультимплексоров могут применяться микросхемы 564КП1 и 564КП2. В этом случае информационные входы A_j являются выходами f_i , а выход f – входом G . Работа этих демультимплексоров, которые могут применяться как для аналоговых, так и для импульсных сигналов, подробно рассмотрена в [3].

И мультиплексоры, и демультимплексоры – это *коммутаторы* логических сигналов. Демультимплексоры иногда называются *распределителями*. На передающей стороне используются мультиплексоры, а на приемной – демультимплексоры, т. е. совместное использование этих устройств позволяет осуществить временное разделение (уплотнение) каналов. Следует отметить, что в технической литературе встречаются обозначения мультиплексоров MX и MS . Кроме того, в структурных схемах иногда пишется и название устройства, например, коммутатор с 4-х направлений на 1-но может быть обозначен, как на рис.3. 5, в.

Демультимплексор обычно обозначается тем же символом, что и дешифратор (декодер), т. е. DC . Иногда пишется и символ, обозначающий, на сколько выходов коммутируется входной сигнал, как показано на рис. 3. 5, б, из которого видно, что при реализации демультимплексоров на выходах дешифраторов предусматриваются схемы И, одни из входов которых соединяются параллель-

но, образуя общий вход, на который подается входной информационный сигнал B . Логическое выражение, связывающее выходные переменные y_i с входными m_i и переменной B на информационном входе, можно представить выражением $y_i = m_i B$,

$$(3.15)$$

где $m_i = A_1 \dots A_p$ – полные правильные конъюнкции, т. е., по-существу, минтермы на управляющих входах; $i = 0, 1 \dots 2^p - 1$; p – число управляющих переменных.

3.8. Устройство сравнения кодов (цифровой компаратор)

В цифровой технике эти устройства применяются в виде схем сравнения двух чисел (слов), двух n -разрядных двоичных кодов $A(a_n, a_{n-1} \dots a_1)$ и $B(b_n, b_{n-1} \dots b_1)$, где a_i, b_i – двоичные переменные «0» или «1».

В результате сравнения кодов чаще всего требуется установить факт равенства $A = B$, т. е. равенства a_i, b_i для всех $i = 1, 2 \dots n$, либо факт неравенства $A \neq B$ – если хотя бы в одном k -ом разряде $a_k \neq b_k$.

Иногда факта неравенства недостаточно и требуется установить знак неравенства, т. е., что действительно имеет место $A > B$ или $A < B$.

Равнозначность разрядов a_i и b_i определяется схемой равнозначности, истинностью функции $f_i = a_i b_i \vee \bar{a}_i \bar{b}_i$.

$$(3.16)$$

Поэтому равенство кодов определяется истинностью конъюнкции всех f_i :

$$f_{A=B} = \bigwedge_{i=1}^n f_i. \quad (3.17)$$

При $f = 1$ имеет место равенство $A = B$. В противном случае при $f_{A=B} = 0$ будет неравенство $A \neq B$. Аналогично можно установить совпадение или несовпадение кодов A и B с помощью функции неравнозначности разрядов a_i, b_i : $y = a_i \bar{b}_i \vee \bar{a}_i b_i$.

$$(3.18)$$

Если дизъюнкция в (3.18) равна единице, то $A \neq B$ и, наоборот, $A = B$, если дизъюнкция будет равна нулю.

Цифровой компаратор можно получить в различных логических базисах, в частности в базисе И – НЕ, получим [6]:

$$y_{A \neq B} = \bigvee_{i=1}^n (\bar{a}_i \bar{b}_i \vee \bar{\bar{a}_i \bar{b}_i}) = \bigwedge_{i=1}^n (\bar{\bar{a}_i \bar{b}_i \cdot \bar{\bar{a}_i \bar{b}_i}}). \quad (3.19)$$

По этой структурной формуле строим логическую схему, как показано на рис. 3.6, а. Условное графическое изображение устройства сравнения кодов представлено на рис. 3.6, б.

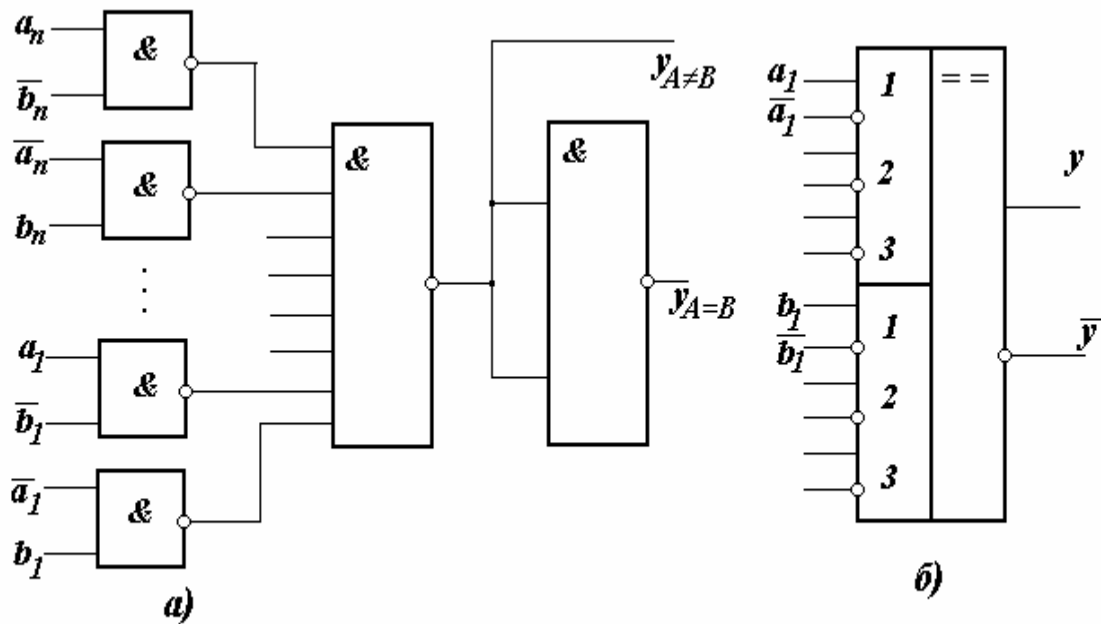


Рис. 3. 6

3. 9. Устройство ЗАПРЕТ

Устройство ЗАПРЕТ - это логическое устройство, в котором сигнал «1», поданный на один из его входов x_1 , называемый *сигнальным*, приводит к появлению сигнала «1» на его выходе Y только в случае, когда на другом входе x_2 , который называется *запрещающим*, будет сигнал «0».

Работа схемы ЗАПРЕТ описывается структурной формулой:

$$Y = x_1 \cdot \bar{x}_2, \quad (3. 20)$$

а условное графическое изображение устройства приведено на рис. 3. 7, а.

Операцию ЗАПРЕТ можно выполнить и по второму входу x_1 . В этом случае он будет инверсным, а x_2 – сигнальным. Этот случай показан на рис. 3 7, б.

Рассмотрим пример использования операции ЗАПРЕТ в конкретном устройстве: амплитудном селекторе импульсов минимальной амплитуды (рис. 3. 8, а). Работа этого устройства становится понятной из рассмотрения временных диаграмм (рис. 3.8, б).

Амплитудный селектор АС пропускает импульсы, превышающие пороговый уровень E - первый и третий, которые запускают расширитель, служащий для получения импульса – пьедестала, совпадающего с входными импульсами.

Входные сигналы (первый и третий) после схемы задержки Δt не будут пропущены схемой ЗАПРЕТ, а второй импульс с амплитудой меньше E , пройдет на выход, т. к. для него нет запрещающего импульса.

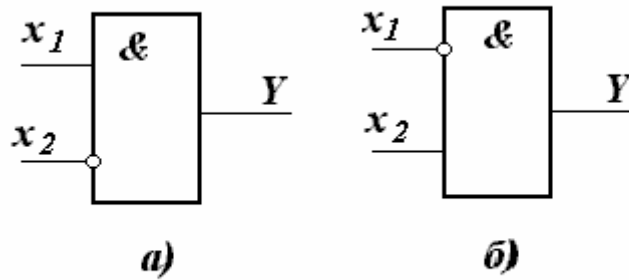


Рис. 3. 7

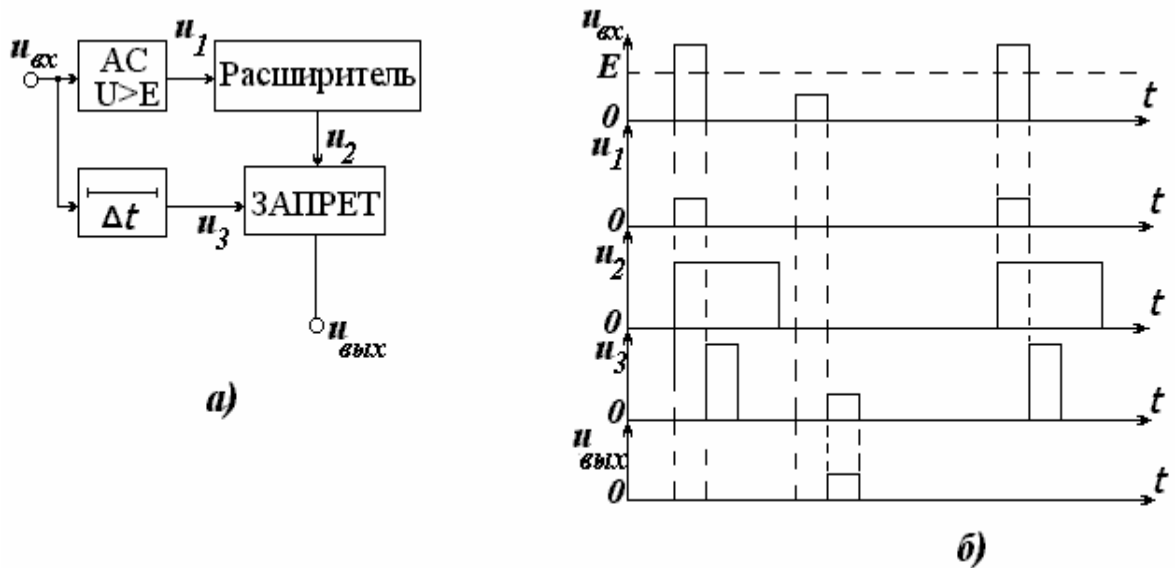


Рис. 3. 8

3. 10. Шифраторы и кодопреобразователи

Шифратор – это устройство, преобразующее сигнал «1» на одном из входов в кодовую комбинацию на выходных шинах. Если с выходных шин шифратора считать m – элементный код, то максимальное число входных шин зависит от возможного числа кодовых комбинаций и составляет 2^m . Шифратор, как и дешифратор, - это схема со многими выходами, но при одинаковом числе элементов кода m число булевых функций, описывающих работу шифратора, меньше, чем для дешифратора, но сами функции намного сложнее.

Предположим, что требуется построить шифратор, позволяющий получить три кодовые комбинации двухэлементного кода: 01; 10; 11. В этом устройстве будет три входных шины x_1, x_2, x_3 и две выходные шины F_1, F_2 . Булевы функции для каждого выхода представлены в табл. 10, а условное графическое изображение шифратора (кодера)– на рис. 3. 9.

Таблица 3.4

№ набора	x_1	x_2	x_3	F_1	F_2
0	0	0	0	\emptyset	\emptyset
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	\emptyset	\emptyset
4	1	0	0	1	1
5	1	0	1	\emptyset	\emptyset
6	1	1	0	\emptyset	\emptyset
7	1	1	1	\emptyset	\emptyset

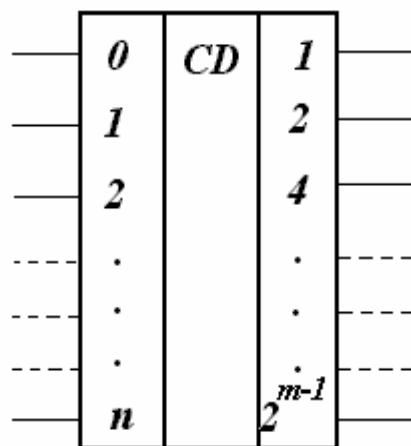


Рис. 3.9

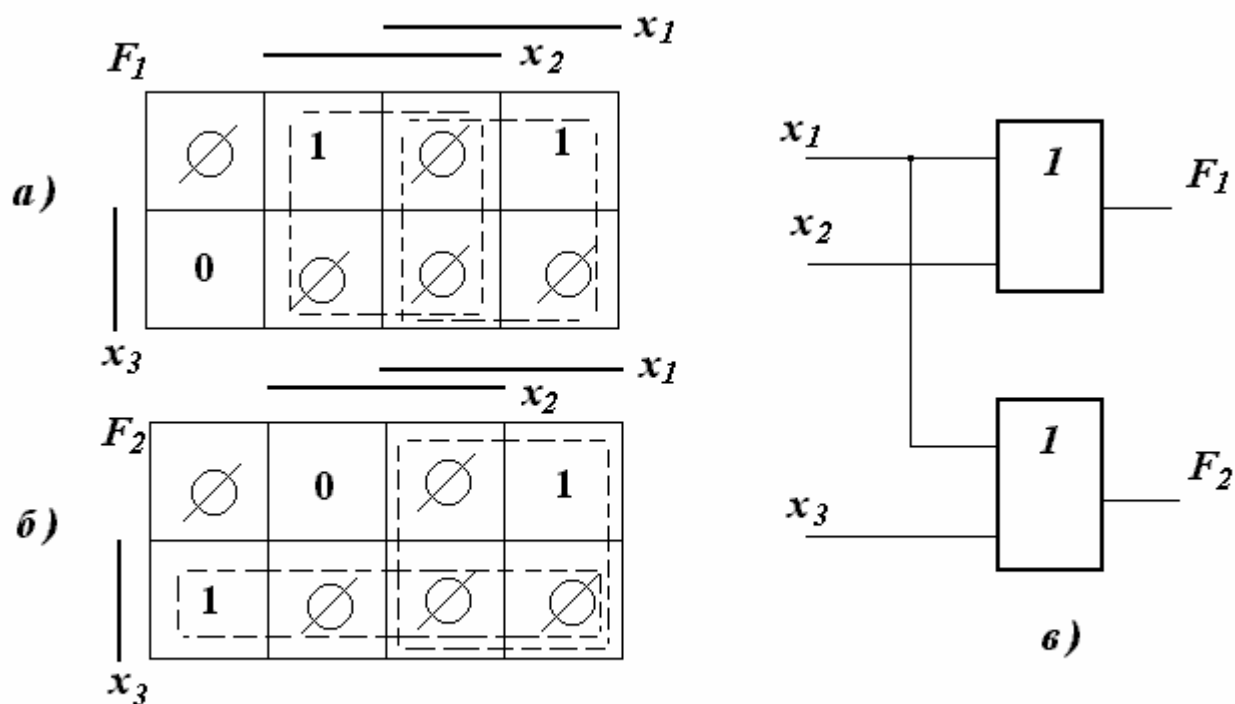


Рис. 3.10

Из табл. 3.4 видно, что все наборы входных сигналов, содержащие две или три единицы, а также нулевой набор для шифратора запрещены. Поэтому выходным сигналам на этих наборах можно давать факультативные значения [11].

Работа такого шифратора описывается системой булевых функций для каждого выхода:

$$F_1 = \sum_{o\phi} (2, 4) \vee \sum_{\phi} (0, 3, 5, 6, 7). \quad (3.21)$$

$$F_2 = \sum_{o\phi} (1, 4) \vee \sum_{\phi} (0, 3, 5, 6, 7). \quad (3.22)$$

Произведя минимизацию (3. 21) и (3. 22) с помощью карт Карно (рис. 3.10, *a* и *б*) с учетом факультативных условий, получим:

$$F_1 = x_1 \vee x_2; \quad F_2 = x_1 \vee x_3. \quad (3. 23)$$

Схема шифратора, описываемого табл. 10 и минимизированного графическим способом, приобретает, согласно (3. 23), очень простой вид, как показано на рис. 3. 10, *в*.

Кодообразователи. Преобразование одного кода в другой может быть выполнено шифратором совместно с дешифратором. При необходимости преобразовать m – элементный код в n – элементный можно воспользоваться схе-

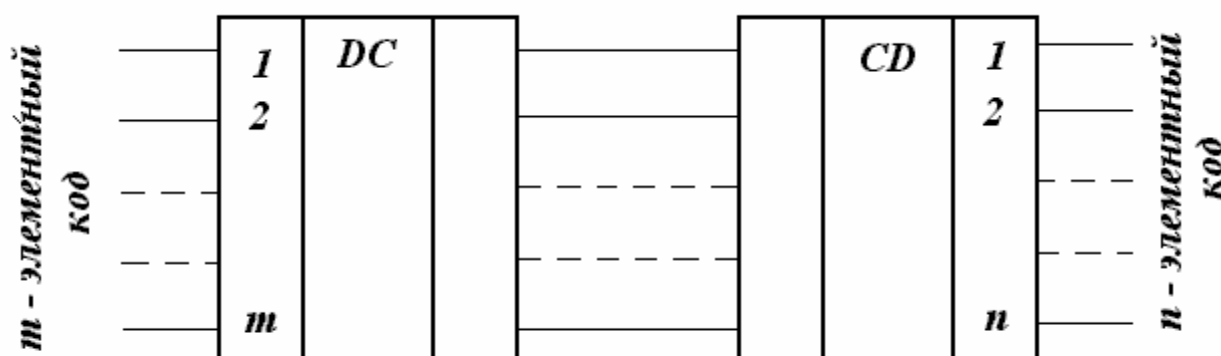


Рис. 3. 11

мой рис. 3. 11.

Такой путь построения кодообразователя – простейший, но он не всегда применяется, так как количество оборудования, которое при этом необходимо использовать, может оказаться значительно бóльшим, как при построении кодообразователя как единой сложной схемы со многими выходами.

3. 11. Программируемые логические устройства с матричной структурой

Для построения логических устройств часто используется универсальный элемент, называемый *программируемая логическая матрица* (ПЛМ). Такая матрица может быть запрограммирована на выполнение логических функций различной степени сложности.

Структура ПЛМ показана на рис. 3. 12. Цепи входных переменных $x_1, x_{2,\Delta}$ и их инверсий $\overline{x_1}, \overline{x_{2,\Delta}}$ составляют горизонтальные цепи матрицы M_1 , вертикальными цепями которой служат *цепи конъюнкции*. Другая матрица M_2 образуется цепями конъюнкции с горизонтальными цепями выходов $y_1, y_{2,\Delta}$. В узлах матрицы M_1 включены элементы, с помощью которых могут формироваться требуемые конъюнкции входных переменных, а элементы в

узлах матрицы M_2 позволяют формировать на выходных цепях требуемые дизъюнкции функций, полученных на цепях конъюнкций.

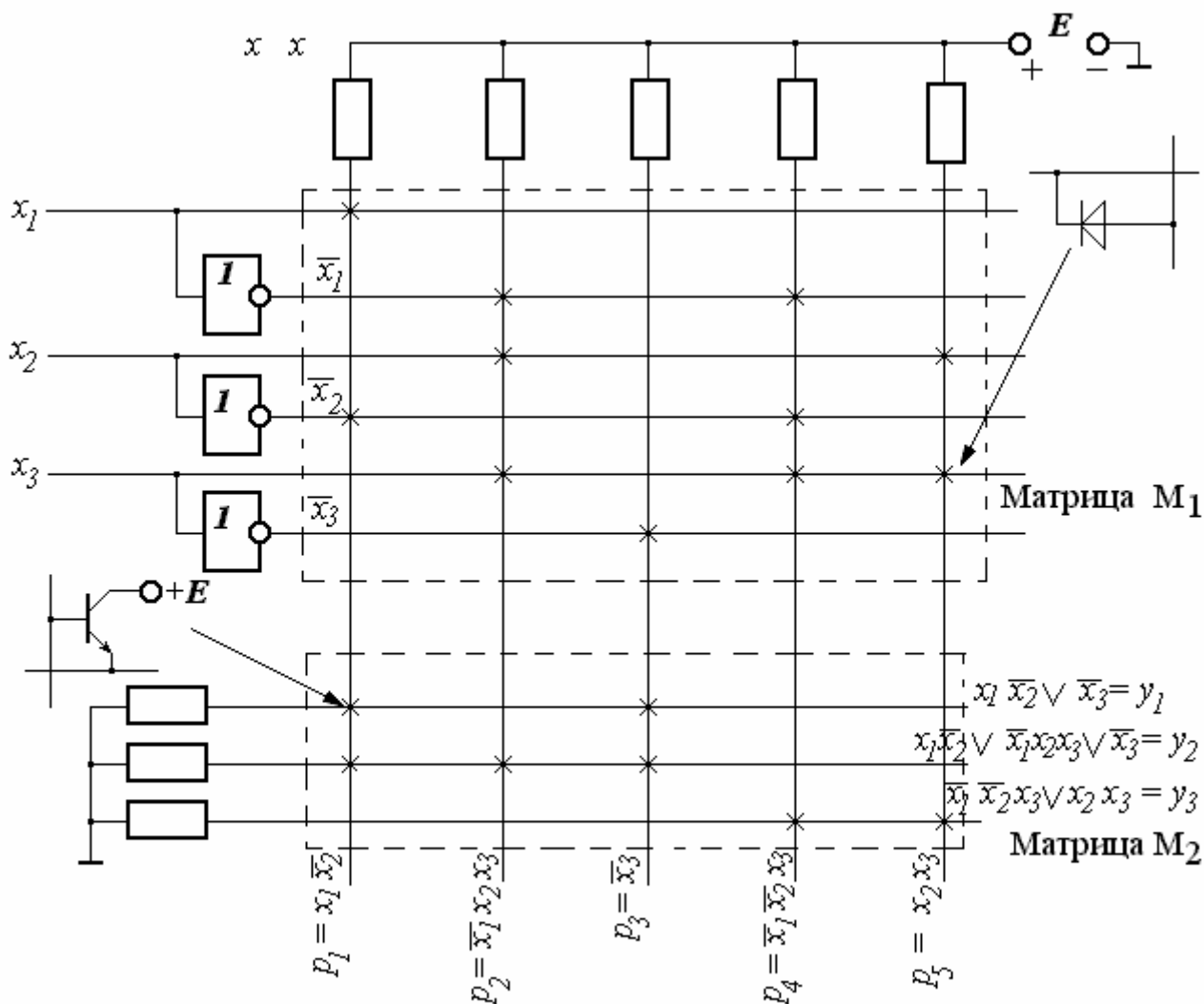


Рис. 3. 12

При программировании ПЛИМ в узлах матриц M_1 и M_2 производится подключение элементов, необходимых для получения выходных логических функций y_1, y_2, \dots, y_K . При необходимости получения инверсной функции в выходные цепи включаются инверторы.

Матрица M_1 содержит горизонтальные цепи, на которых действуют входные переменные x_1, x_2, \dots, x_K и их инверсии $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_K$, и вертикальные цепи, на которых формируются конъюнкции p_1, p_2, \dots, p_K . В отдельных узлах матрицы между ее вертикальными и горизонтальными цепями включаются диоды. На вертикальной цепи образуется высокий потенциал (уровень лог. 1), закрывающий диоды. Если хотя бы на одном из таких входов низкий потенциал (уровень лог. 0), открывается диод, и уровень лог. 0 с этого входа через открытый диод передается на вертикальную цепь матрицы.

На рис. 3. 12 крестиками показаны участки, в которых в процессе программирования создаются соединения, позволяющие получить функции:

$$p_1 = x_1 \overline{x_2}, \quad p_2 = \overline{x_1} x_2 x_3, \quad p_3 = \overline{x_3}, \quad p_4 = \overline{x_1} \overline{x_2} x_3, \quad p_5 = x_2 x_3. \quad (3.24)$$

Если включить в эти узлы диоды, то можно на выводах p_i сформировать любые конъюнкции входных переменных и их инверсий.

В узлах матрицы M_2 между цепями p_i и y_i включены транзисторы, базы которых подключены к цепям p_i , а эмиттеры – к цепям y_i . Если в цепи p_i имеется уровень лог. 1, транзистор будет в открытом состоянии, и высокий потенциал передается в цепь y_i . При этом $y_i = 1$ независимо от уровней на других выходах матрицы M_1 . Это позволяет получить в схеме на рис. 3. 12

$$y_1 = x_1 \overline{x_2} \vee \overline{x_3}, \quad y_2 = x_1 \overline{x_2} \vee x_1 x_2 x_3 \vee \overline{x_3}, \quad y_3 = \overline{x_1} \overline{x_2} x_3 \vee x_2 x_3 \quad (3.25)$$

Программа функционирования комбинационной схемы, построенной на ПЛМ, может быть представлена в виде таблицы. В частности для схемы, представленной на рис. 3. 12, таблица выглядит в виде табл. 3.5.

При построении таблицы следует придерживаться следующих правил. На пересечении j -ой строки и столбца x_i записывается 1, если x_i входит в конъюнкцию p_j на выходе матрицы M_1 без инверсии, и 0 – если с инверсией, если x_i не входит в j -ю конъюнкцию, то ставится прочерк. На пересечении i -ой строки и столбца y_k записывается 1, если i -я элементарная конъюнкция входит в ДНФ y_k , и точка в противном случае.

Таблица 3.5

	x_1	x_2	x_3	y_1	y_2	y_3
p_1	1	0	–	1	1	·
p_2	0	1	1	·	1	·
p_3	–	–	0	1	1	·
p_4	0	0	1	·	·	1
p_5	–	1	1	·	·	1

Программирование ПЛМ может выполняться на заводе-изготовителе в процессе изготовления микросхемы на этапе формирования элементов в узлах матриц, а также самим пользователем. В этом случае завод-изготовитель микросхемы выпускает ПЛМ со вставленными элементами во все узлы матриц. Пользователь в процессе программирования с помощью импульсов тока пере-

жигает плавкие перемычки, последовательно включенные с элементами ПЛМ, что приводит к их отключению. Оставшиеся элементы обеспечивают необходимые функции на выходах микросхемы.

Число входов в выпускаемых промышленностью ПЛМ достигает 24, выходов – 16, а цепей конъюнкции – 96. Структуры, программируемые на заводе – изготовителе главным образом используются при выпуске БИС [3].

Рассмотрим в качестве примера использования ПЛМ в цифровых устройствах некоторые типовые узлы цифровых устройств, выполненные на ПЛМ.

Дешифратор на ПЛМ. Рассмотрим, как реализуется на ПЛМ дешифратор, преобразующий трехразрядный двоичный код (x_1, x_2, x_4) , подаваемый на вход в унитарный 8 – разрядный код на выходе $(y_0 \text{ К } y_7)$.

Логические выражения, определяющие функционирование этого дешифратора можно представить, как показано в [3], в виде (3. 25):

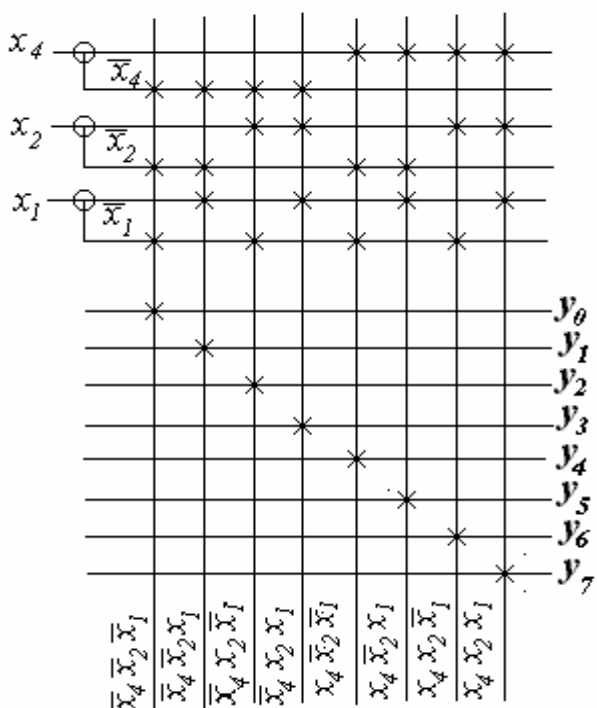


Рис. 3. 13

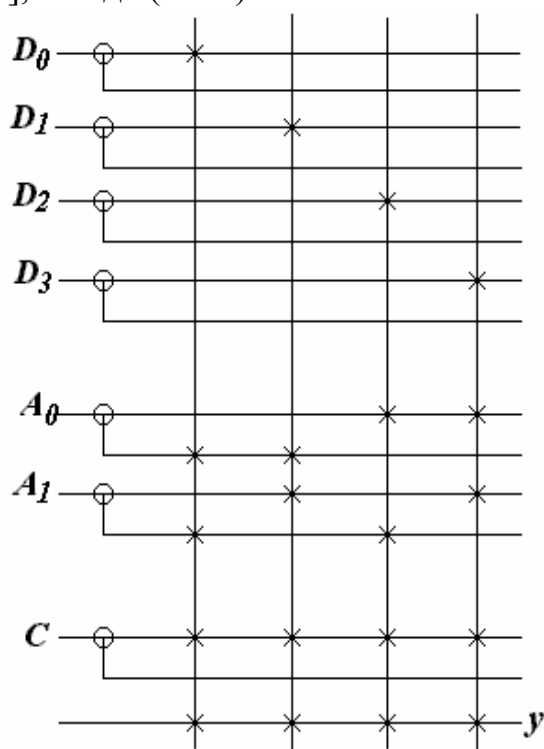


Рис. 3. 14

$$\begin{aligned}
 y_0 &= \overline{x_4} \overline{x_2} \overline{x_1} , & y_1 &= \overline{x_4} \overline{x_2} x_1 , \\
 y_2 &= \overline{x_4} x_2 \overline{x_1} , & y_3 &= \overline{x_4} x_2 x_1 , \\
 y_4 &= x_4 \overline{x_2} \overline{x_1} , & y_5 &= x_4 \overline{x_2} x_1 , \\
 y_6 &= x_4 x_2 \overline{x_1} , & y_7 &= x_4 x_2 x_1 .
 \end{aligned}
 \tag{3.25}$$

ПЛМ, реализующая систему функций (3. 25), представлена на рис. 3. 13, на котором для упрощения схемы отсутствуют некоторые элементы (источник питания и резисторы).

Мультиплексор и демультиплексор на ПЛМ. Схема мультиплексора на

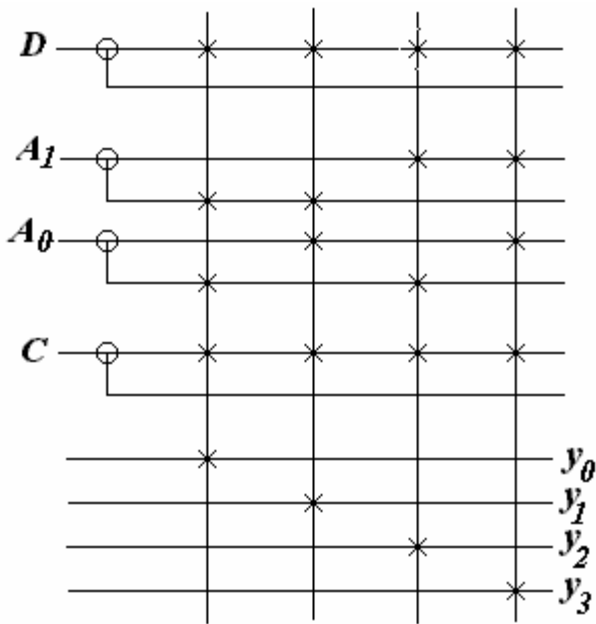


Рис. 3. 15

четыре входа (D_3, D_2, D_1, D_0) показана на рис. 3. 14. На этом рисунке A_1, A_0 – адресные входы; C – вход для подачи сигнала разрешения выдачи; y – выход.

На рис. 3. 15 приведена схема демультиплексора на четыре выхода (y_3, y_2, y_1, y_0). На этом рисунке D – вход; A_1, A_0 – адресные входы; C – вход сигнала разрешения выдачи информации.

Шифратор на ПЛМ. В качестве примера выполнения шифратора на ПЛМ рассмотрим построение шифратора, преобразующего унитарный десятичный код (с отображением десятичной цифры уровнем *лог. 1* на одной из десяти цепей) в двоичный код *8421*.

Для этого воспользуемся логическими выражениями (3. 26), приведенными в [3]:

$$\begin{aligned}
 x_1 &= \overline{y_1} | \overline{y_3} | \overline{y_5} | \overline{y_7} | \overline{y_9}, & x_2 &= \overline{y_2} | \overline{y_3} | \overline{y_6} | \overline{y_7}, \\
 x_4 &= \overline{y_4} | \overline{y_5} | \overline{y_6} | \overline{y_7}, & x_8 &= \overline{y_8} | \overline{y_9},
 \end{aligned}
 \tag{3. 26}$$

где y_i – входные сигналы; x_j – выходные сигналы (значения разрядов кода *8421*).

ПЛМ, реализующая функции шифратора, показана на рис. 3. 16 (для упрощения схемы не показаны источник питания и резисторы).

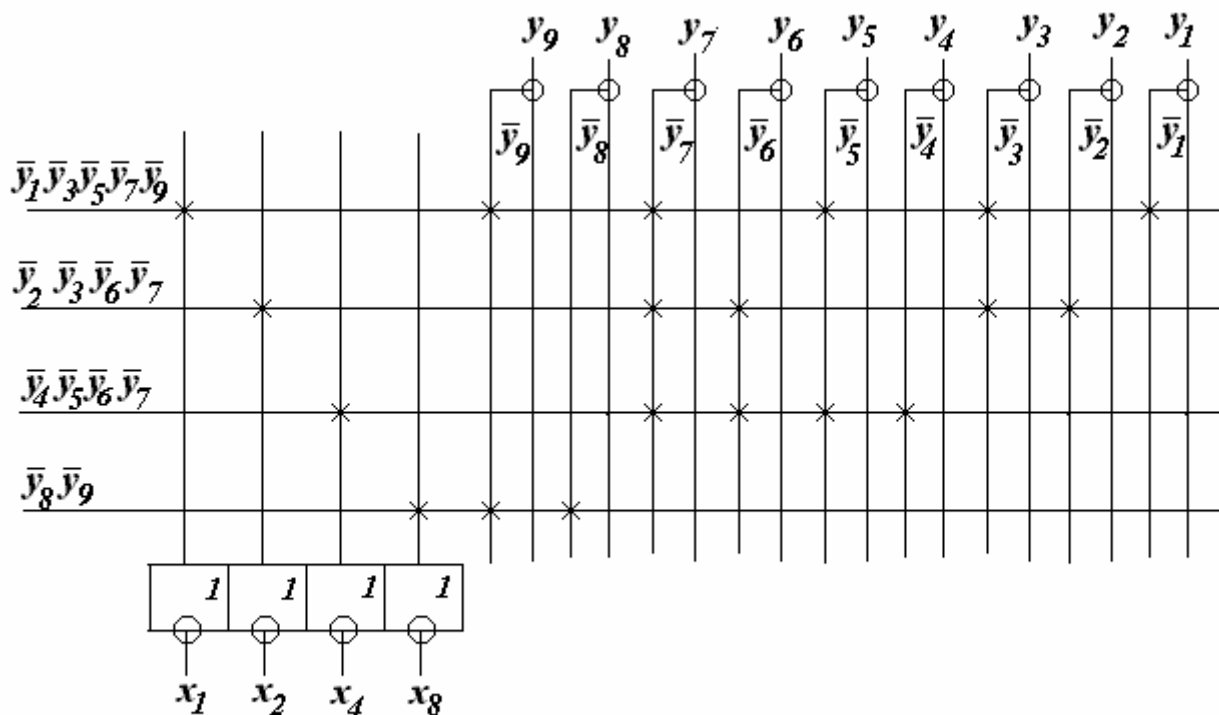


Рис. 3. 16

Вопросы и задания для самоконтроля

1. Что такое комбинационные устройства?
2. Приведите примеры комбинационных устройств.
3. Дайте определение и начертите схему устройства неравнозначности.
4. Начертите таблицу истинности устройства равнозначности.
5. Дайте словесное описание, начертите таблицу истинности и функциональную схему полусумматора.
6. Начертите схему полного сумматора.
7. Что такое дешифратор? Как работает схема простейшего дешифратора?
8. Каково назначение мультиплексора? Опишите его работу.
9. Как работает демультиплексор?
10. Начертите схему цифрового компаратора и опишите его работу.
11. Для чего используются универсальные логические элементы «Программируемые логические матрицы, (ПЛИМ)»?
12. Приведите пример использования ПЛИМ для выполнения преобразования унитарного десятичного кода в код 8421.
13. Что такое устройство ЗАПРЕТ?
14. Как используется устройство ЗАПРЕТ в селекторе импульсов минимальной амплитуды?
15. Покажите, как получить логическую схему цифрового компаратора в базе И – НЕ.

4. ЦИФРОВЫЕ АВТОМАТЫ

4.1. Основные понятия

В рассмотренных в предыдущей главе комбинационных устройствах значения выходных сигналов y_i определялись лишь конкретным сочетанием входных сигналов x_i для одних и тех же моментов времени t_i .

Цифровые автоматы или, иначе, конечные автоматы – это *последовательностные устройства*. Они содержат наряду с комбинационными устройствами (КУ) также запоминающие устройства (ЗУ). В этих устройствах выходные сигналы на k – м такте [6] Y_k зависят не только от входных сигналов на этом такте X_k , но и от внутреннего состояния устройства, т. е. практически от сигналов на элементах памяти Q_k : $Y_k = F(X_k, Q_k)$. (4.1)

А сигналы на элементах памяти Q_k на k – м такте зависят от их состояния в предшествующий момент времени, т. е. на $k-1$ – м такте. С другой стороны, сигналы на элементах памяти Q_k на k – м такте зависят от их состояния на $k-1$ – м такте и сигналов на k -м такте: $Q_k = f(Q_{k-1}, X_k)$. (4.2)

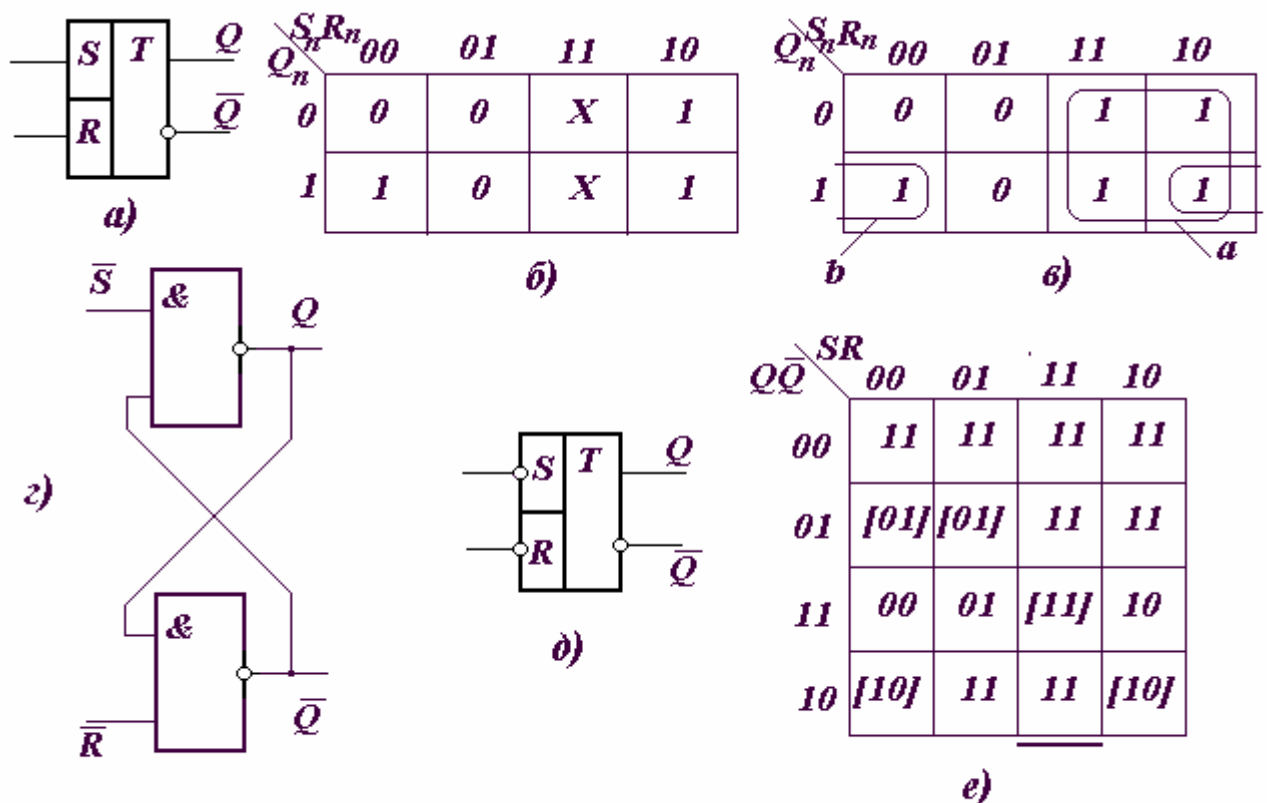


Рис. 4.1

Выражения (4.1) и (4.2) описывают работу конечного автомата (КА), которая кроме аналитических выражений, может быть задана таблично в виде

таблицы переходов КА, т. е. таблицы истинности, содержащей информационные значения входных сигналов, внутренних состояний и выходных сигналов.

В цифровых устройствах входные управляющие сигналы и выходные сигналы применяются *потенциального* типа, т. е. КА управляется высокими и низкими уровнями напряжения.

В *синхронных* КА имеется один или несколько входов C , на которые поступают синхронизирующие сигналы тактовых импульсов (ТИ). КА такого типа срабатывает только в момент действия ТИ (иногда после его окончания).

Быстродействие КА определяется главным образом инерционностью логических элементов, которые входят в его состав. Инерционность зависит от длительности установления стационарных уровней. Если каждый логический элемент задерживает сигнал на время $t_{з. ср.}$, а количество элементов, через которые сигнал распространяется, равно l , то задержка сигнала $t_з = lt_{з. ср.}$. Эта задержка определяет разрешающее время $t_{min} = t_з$ и максимальную частоту переходов КА $f_{max} = 1/t_{min}$.

4.2. Триггеры

Общие сведения. Триггеры относятся к классу бистабильных генераторов прямоугольных импульсов. Они имеют два длительно устойчивых состояния равновесия и переходят из одного состояния в другое скачкообразно под действием внешнего сигнала.

Триггер – это типичный представитель последовательностных устройств. Он может быть тактируемым или асинхронным. Элементная база, которая в настоящее время чаще всего применяется для построения триггерных схем, – это интегральные схемы различной степени интеграции.

Асинхронные триггеры. Асинхронные триггеры – это простейшие КА, имеющие два выхода Q и \bar{Q} и два входа S и R (рис. 4.1, а). Входной сигнал S устанавливает триггер в положение $Q = 1$. S – от английского слова *SET* – установка. Сигнал R от английского слова *RESET* – сброс переключает триггер в положение $Q = 0$. Случай $R = S = 1$ требует для каждой разновидности триггера отдельного рассмотрения, учитывающего особенности схемы.

Для сигналов S, R, Q, \bar{Q} будем различать две совокупности значений:

$$S_n, R_n, Q_n, \bar{Q}_n; \quad S_{n+1}, R_{n+1}, Q_{n+1}, \bar{Q}_{n+1}. \quad (4.3)$$

Совокупность, для которой значения Q и \bar{Q} отмечены индексом n , соответствует состоянию триггера в момент появления входных сигналов S_n, R_n , но это состояние зависит не только от значений входных сигналов, но и от состояния триггера перед поступлением этих сигналов, т. е. от Q_n, \bar{Q}_n .

Работу триггерных схем можно описать с помощью таблиц, аналогичных таблицам истинности. Но в данном случае таблица иллюстрирует переход значений выходного сигнала от Q_n к Q_{n+1} и ее называют *таблицей переключений*. При построении таблицы учтем следующие очевидные положения: если $S = 0, R = 0$, то состояние триггера не изменяется, т. е. $Q_{n+1} = 0$, если $Q_n = 0$; и $Q_{n+1} = 1$, если $Q_n = 1$.

Таблица 4.1

S_n	R_n	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

Сигнал $S = 1$ устанавливает на выходе $Q=1$, и при комбинации на входе $S_n=1, R_n=0$ на выходе получим сигнал $Q_{n+1}=1$. При этом, если $Q_n=0$, то произойдет переключение выходного напряжения, а если $Q_n=1$, то сохранится прежнее состояние.

Сигнал $R = 1$ устанавливает на выходе $Q=0$. Поэтому при $S = 0; R = 1; Q_{n+1} = 0$. Если при этом $Q_n = 1$, то триггер переключается, если $Q_n = 0$, то это состояние сохраняется. Совокупность входных сигналов $S = R = 1$ не дает однозначного определения Q_{n+1} , и на выходе получим неопределенность, которая обычно обозначается буквой X.

Таблица переключений триггера представлена в виде табл. 4.1. Если эту таблицу изобразить в виде карты Карно, то эта карта будет изображать функцию трех переменных, а в ячейках карты будет изображено значение Q_{n+1} , как показано на рис. 4. 1, б . Разновидности статических триггеров при отображении их состояний на карте Карно отличаются значениями Q_{n+1} в столбце неопределенности.

RS-триггер с инверсными входами. Заполним столбец неопределенности на карте Карно (рис. 4. 1, б) сочетанием «11» и проведем минимизацию функции, отображаемой картой, графическим методом, как показано на рис. 4.1, в . Контур «а» говорит о том, что $Q_{n+1} = 1$, когда $S_n = 1$ независимо от того, какие значения принимают сигналы R_n, Q_n . Контур «б» говорит о том, что $Q_{n+1} = 1$, когда $Q_n = 1, R_n = 0$ независимо от сигнала S_n . Минимизированная дизъюнктивная форма записи интересующей нас функции будет иметь вид:

$$Q_{n+1} = S_n \vee Q_n \bar{R}_n. \quad (4.4)$$

Для реализации этой функции на элементах И – НЕ ее необходимо представить в конъюнктивной форме:

$$\overline{Q_{n+1}} = \overline{S_n Q_n \bar{R}_n} = \overline{\overline{S_n} \overline{Q_n} R_n}. \quad (4.5)$$

По этой структурной формуле строим логическую схему в выбранном базисе (рис. 4. 1, з) . Условное графическое изображение устройства представлено на рис. 4. 1, д . Связь выходных сигналов устройства с входными вытекает из логической функции, выполняемой элементом «штрих Шеффера»:

$$Q = \overline{SQ} = S \vee \overline{Q}; \quad \overline{Q} = \overline{RQ} = R \vee \overline{Q} \quad (4.6)$$

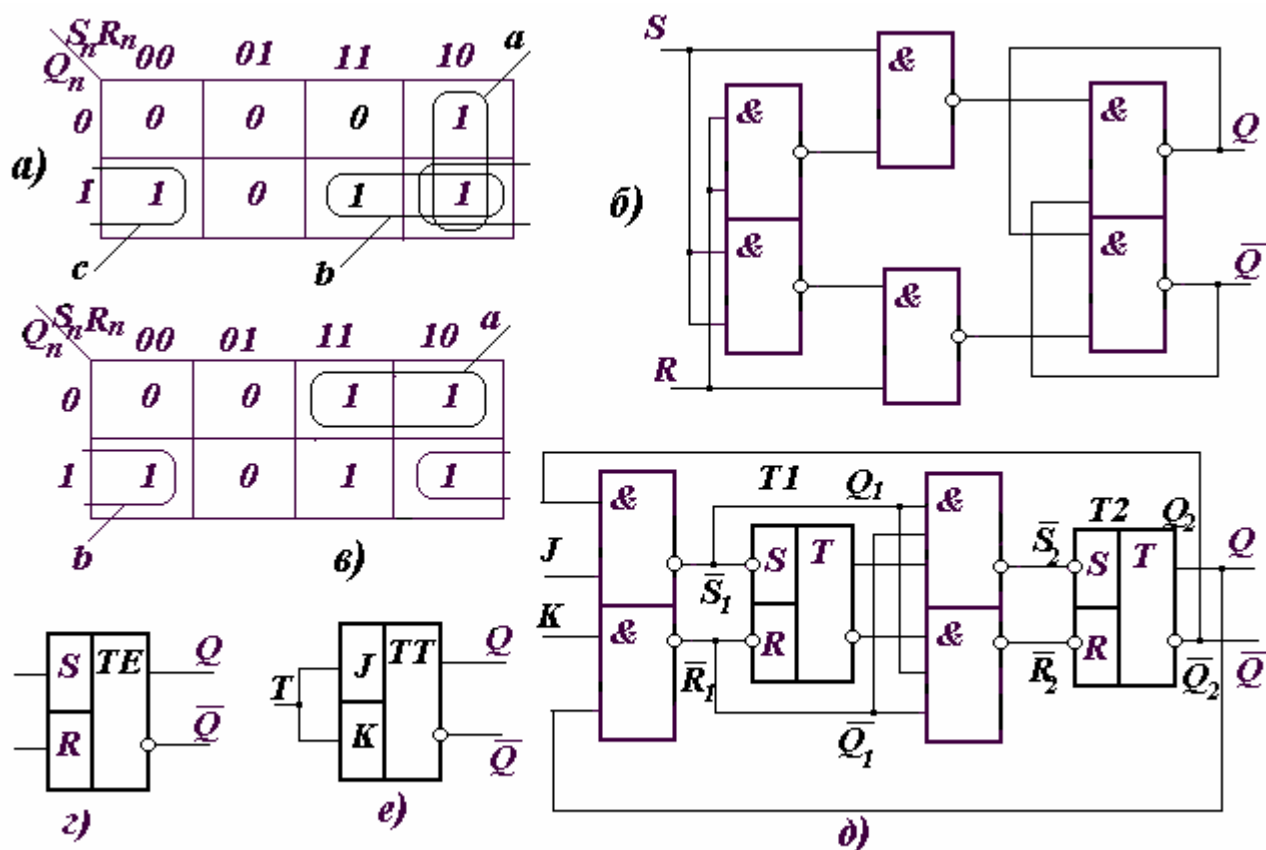


Рис. 4. 2

Уравнения (4.6) называются уравнениями возбуждения. С их помощью строится обобщенная карта Карно (рис. 4. 1, e). Истинные значения Q и \overline{Q} , полученные из уравнений, записываются в той же последовательности, что и аргументы данной строки, т. е. сначала Q , а затем \overline{Q} . Из полученных сочетаний Q и \overline{Q} устойчивыми будут те, которые подобны заданным в качестве аргументов. Они помещены в квадратные скобки.

Из всех возможных комбинаций устойчивых сочетаний Q и \overline{Q} сочетания «00», и «11» следует исключить, т. к. не выполняется требование инверсности. Из карты видно, что сочетание «00» неустойчиво, ему нет подобного, и оно исключается автоматически. Но остается еще сочетание «11», которое может быть устойчиво. Его необходимо исключить принудительно, запрещая сочетание на входах $S = R = 1$. Это получается, если на входе выполняется равенство $SR = 0$ (этот столбец отмечен внизу чертой).

E – триггер. Если заполнить столбец неопределенности карты Карно сочетанием «01» и минимизировать функцию трех переменных (рис. 4. 2, a) $Q_{n+1} = f(Q_n, S_n, R_n)$, то следует обратить внимание на три контура.

Контур «а» дает первый минтерм: $S_n \bar{R}_n$, контур «b» - второй : $S_n Q_n$; контур «с» - третий: $\bar{R}_n Q_n$. Полную структурную формулу можно представить в виде: $Q_{n+1} = S_n \bar{R}_n \vee S_n Q_n \vee \bar{R}_n Q_n = \overline{S_n R_n Q_n} \overline{S_n R_n}$. (4.7)

Выражение (4.7) можно реализовать на универсальных элементах И-НЕ (рис. 4. 2, б), а условное графическое изображение Е- триггера на схемах дано на рис. 4. 2, в.

Если составить уравнения возбуждения триггера в виде:

$$Q = \bar{Q} \vee SR; \quad \bar{Q} = \bar{Q} \vee \bar{S}R, \quad (4.8)$$

то по обобщенной карте Карно нетрудно определить, что для этого типа триггера запретные сочетания «00» и «11» неустойчивы и автоматически исключаются, а устойчивые состояния выходов всегда инверсны.

JK-триггер. Если заменить столбец неопределенности в карте Карно сочетанием «10», то карта Карно приобретет вид, как показано на рис. 4. 2, г. Анализируя этот рисунок приходим к заключению, что контур «а» дает нам истинность Q_{n+1} , если $Q_n = 0$ и $S_n = 1$ независимо от R_n . Это объединение дает первый минтерм $\bar{Q}S_n$. Выделяя контур «b», можно сделать вывод, что Q_{n+1} истинно, если $Q_n = 1$ и $R_n = 0$ независимо от S_n . Это объединение дает второй минтерм $Q_n \bar{R}_n$. Искомая функция будет получена в виде:

$$Q_{n+1} = S_n \bar{Q}_n \vee \bar{R}_n Q_n. \quad (4.9)$$

Схема триггера, функционирующего в соответствии с (4.9), представлена на рис. 4. 2, д. В эту схему входят два **RS**- триггера с инверсными входами, выполненные также на элементах И – НЕ. Эти внутренние триггеры имеют собственные входы **R, S**, на которые подаются сигналы \bar{R}_1, \bar{S}_1 для Т1 и \bar{R}_2, \bar{S}_2 для Т2. Чтобы различать входные сигналы триггера в целом и входные сигналы «внутренних» триггеров, принято входы этого триггера обозначать буквами **J, K**. Вход **J** имеет то же значение, что и вход **S**, а вход **K** – то же значение, что и вход **R**. С учетом последнего замечания уравнение (4.9) можно представить в виде $Q_{n+1} = J_n \bar{Q}_n \vee \bar{K}_n Q_n$. (4.10)

На основании (4.10) можно построить таблицу переключений триггера (табл. 4.2), а также в более компактной форме (табл. 4.3). Анализ табл. 4.2 и табл. 4.3 показывает, что **JK** – триггер не имеет *запрещенных сочетаний входных сигналов J и K*. Это делает его похожим на **E** – триггер, но в отличие от **E** – триггера при $J=K=1$ этот триггер изменяет свое состояние на инверсное, т. е. $Q_{n+1} = \bar{Q}_n$.

Таблица 4.2

J_n	K_n	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Таблица 4.3

J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Это свойство позволяет JK – триггеру работать в счетном режиме. Для этого необходимо объединить входы J, K , и мы получим счетный триггер, или T –

триггер, условное графическое изображение которого дано на рис. 4 2, е. Этот тип триггера имеет более широкие возможности для использования, так как это – универсальный триггер.

4.3. Синхронные триггеры

Синхронный RS- триггер (RSC - триггер). Основа схемы – это асинхронный RS – триггер с инверсными входами (элемент D3 на рис. 4. 3, а). Управляющие сигналы на каждый из входов триггера подаются через элементы И – НЕ (элементы D1, D2 на рис. 4. 3, а). Условное графическое изображение триггера показано на рис. 4. 3, б, а временные диаграммы – на рис. 4. 3, в.

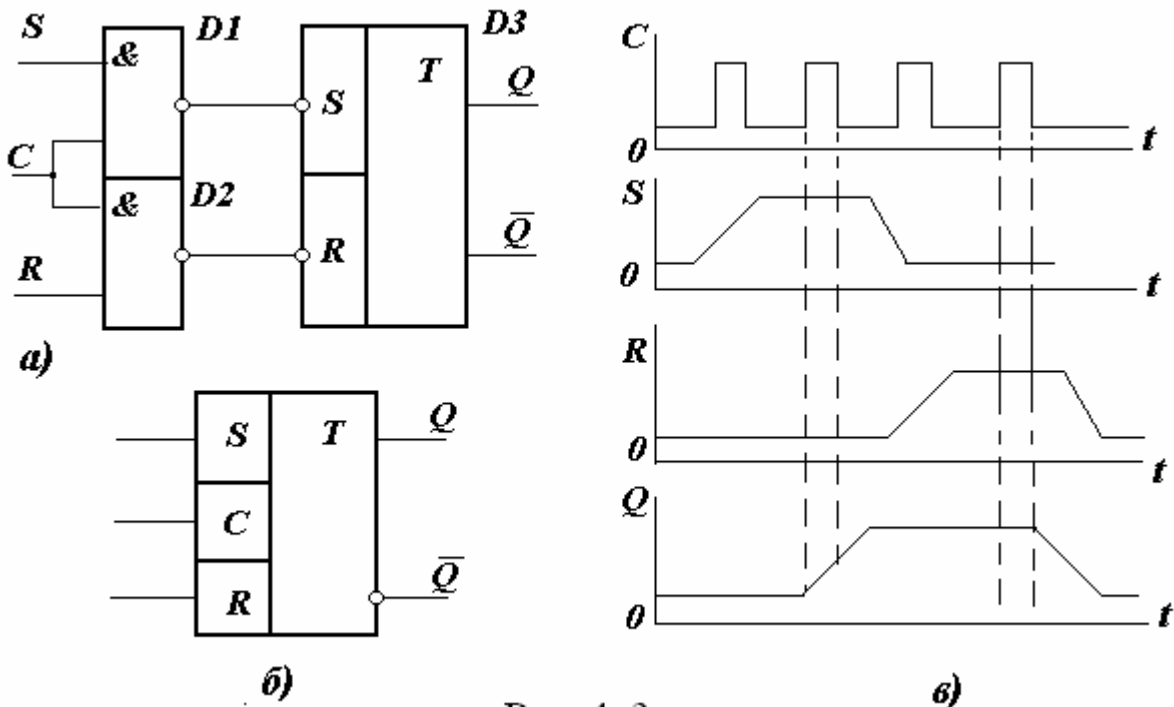


Рис. 4.3

Элемент D1 обеспечивает совпадение синхроимпульсов C и установочных сигналов S , т. е. временное стробирование сигналов S , а элемент D2 стробирует по времени сигналы R .

Уравнение асинхронного RS - триггера с инверсными входами (4.4) для синхронного RSC – триггера приобретает вид (4.12):

$$Q_{n+1} = C_n S_n \vee \overline{C_n R_n} Q_n. \quad (4.12)$$

При этом нужно иметь в виду, что входным сигналом \overline{S} в этом случае будет

выходной сигнал элемента D1, т. е. \overline{CS} ; входным сигналом \overline{R} - выходной сигнал элемента D2, т. е. \overline{RC} .

D – триггер (триггер задержки). В синхронных триггерах в промежутке между синхроимпульсами выходные напряжения не изменяются. Это позволяет создать на основе синхронизируемых RS – триггеров триггеры задержки – D – триггеры (от английского слова DELAY – задержка).

Логическое уравнение этого триггера можно получить, если в (4.12) заменить сигнал S на сигнал D, а сигнал R – на инверсию сигнала D:

$$Q_{n+1} = C_n D_n \vee \overline{C_n D_n} Q_n. \quad (4.13)$$

Воспользовавшись правилом Де Моргана, (4.13) можно представить в более удобной форме, чтобы реализовать D – триггер на логических элементах И – НЕ:

$$Q_{n+1} = \overline{C_n D_n} [(C_n \overline{C_n D_n}) Q_n]. \quad (4.14)$$

На рис. 4.4, а изображена схема, а на рис. 4.4, б – условное графическое

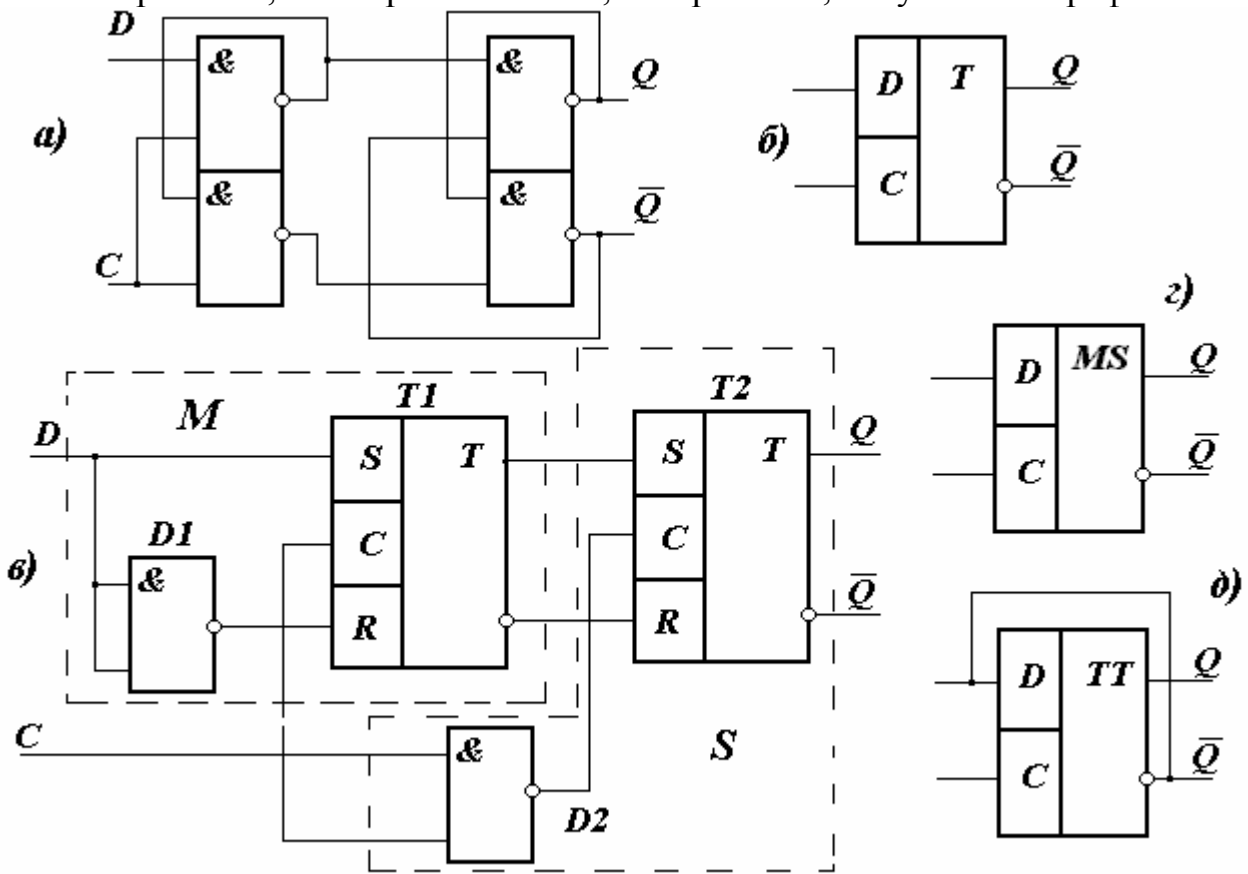


Рис. 4.4

изображение D – триггера, в основе работы которого лежит то, что он принимает информацию по входу D. Информация записывается в триггер при поступлении тактирующего импульса, а на выходе появляется в момент окончания тактирующего импульса. Эта информация может быть использована, когда придет следующий тактирующий импульс, т. е. в следующем такте. Это и объясняет, почему D – триггер – элемент задержки на один такт.

На рис. 4. 4, в показан вариант *D*-триггера, основными элементами которого являются два *RSC*-триггера с инверторами на входе, а сам триггер носит название *MS*-триггера, представляющего собой систему из двух триггеров, один из которых – основной *M* (*MASTER* - хозяин), а другой – вспомогательный *S* (*SLAVE* - раб). Если на входе *MS*-триггера действует сигнал $D=1$, то при поступлении сигнала $C=1$, единица будет записана в *T1*. После окончания тактового импульса на входе *C* триггера *T2* будет действовать сигнал $\bar{C} = 1$, и информация будет переписана из триггера *T1* в триггер *T2*.

Таблица 4.4

Q_{k-1}	T_k	Q_k
0	0	0
0	1	1
1	0	1
1	1	0

При изменении входного сигнала *D* на инверсный, т. е. при $D = 0$, на выходе элемента *D1* получим $\bar{Q} = 1$.

В момент прихода тактирующего импульса этот сигнал переведет *T1* в состояние 0. После окончания тактирующего импульса информация из *T1* будет переписана в *T2*. Если сигнал *D* не изменяется, то информация, записанная в *MS* – триггере, сохраняется. Условное графическое изображение *MS* – триггера показано на рис. 4. 4, з .

Счетный режим *D* – триггера. На рис. 4. 4, д показано, как легко получить из синхронного *D* – триггера триггер, работающий в счетном режиме. Таблица переключений такого триггера показана в виде табл. 4.4. Считаемые импульсы подаются на вход *C*, а выход \bar{Q} соединяется со входом *D* .

4. 4. Регистры

Основные определения . Регистр – это устройство, предназначенное для выполнения операций приема, хранения и передачи слова, т. е. упорядоченной совокупности двоичных переменных (*разрядов*). Разряды регистра обычно нумеруются в том же порядке, как и разряды в слове, т. е. от $0 \dots n$, где *n* определяет информационную емкость.

Слова из регистра поступают на комбинационные устройства для выполнения логических операций, а затем – на другие регистры, куда записываются результаты операций.

По способу записи информации регистры классифицируются на параллельные и последовательные (регистры сдвига).

Параллельные регистры . Это регистры, представляющие собой несколько триггеров с общими сигналами управления и синхронизации и индивидуальными информационными входами для приема разрядов слова. Запись и считывание информации в этом типе регистра разделены во времени, что позволяет использовать в них наиболее простые схемы триггеров с записью информации по прямому или инверсному входам (парафазный вход).

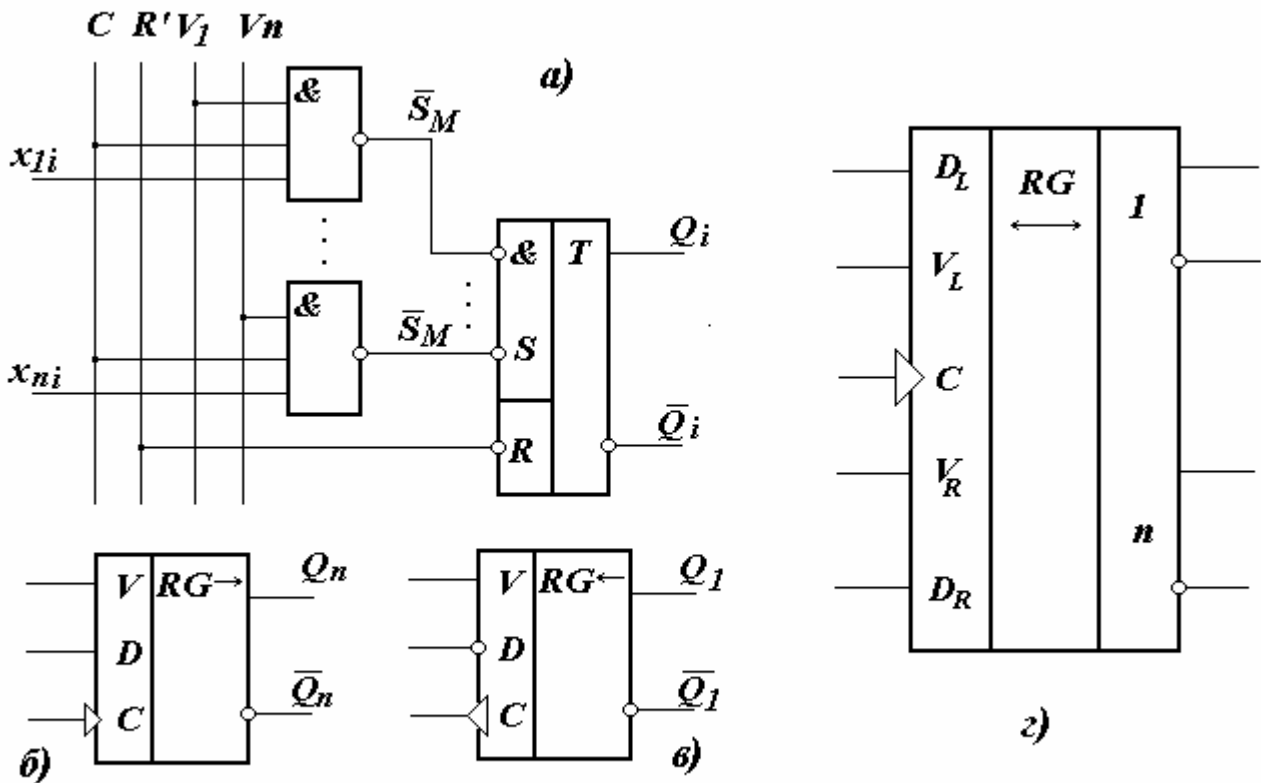


Рис. 4. 5

Для повышения помехоустойчивости регистров часто применяют стробующие управляющие сигналы, т. е. синхронный принцип работы.

На рис. 4. 5, а представлена логическая схема разряда двухтактного однофазного регистра на элементах И-НЕ и RS -триггере с инверсными входами.

В этой схеме информационные входы $x_1, x_2 \dots x_n$ и управляющие входы $R', V_1, V_2 \dots V_n$. Триггер содержит n дополнительных установочных входов \bar{S} , функцию которых выполняют дополнительные входы $\&$. Тактирование регистра производится синхросигналами C , подаваемыми одновременно на третий вход элементов И-НЕ. Для того, чтобы записать в регистр слова x_k , необходимо подать последовательность двух импульсов сначала на управляющий вход R' , затем на управляющий вход V_k . Первым импульсом обеспечивается микрооперация сброса состояния регистра в ноль. Вторым импульсом в регистр записывается слово x_k .

Последовательный регистр сдвига в одном направлении. Этот тип регистра имеет два входа D и V и один выход Y . Вход D называется информационным, вход V – управляющим. Если рассматривать сдвигающий регистр как автомат [6], внутреннее состояние которого задано совокупностью состояний разрядов $Q_1, Q_2, \dots Q_n$, то эта совокупность разрядов представляет собой слово информации, хранимой в регистре.

В случае регистра, сдвигающего вправо, функции переходов и выходов автомата представляются переключательными функциями:

$$Q_k(t+1) = VQ_{k-1}(t) + \bar{V}Q_k(t), k = 1, 2, \dots, n; Q_0 = D; Y = Q_n. \quad (4.15)$$

Для регистра, сдвигающего влево, эти функции можно представить в виде: $Q_k(t+1) = VQ_{k+1}(t) + \bar{V}Q_k(t), k = 1, 2, \dots, n; Q_{k+1} = D; Y = Q_1.$ (4.16)

Операция сдвига выполняется при $V=1$. В освобождающийся разряд будет записана входная переменная D . Выходным будет крайний разряд, в сторону которого происходит сдвиг. При $V=0$ слово информации в регистре остается без изменения.

Условное графическое изображение сдвиговых регистров влево и вправо показано на рис. 4. 5, б и в. Направление стрелки на входе S показывает в первом случае положительный фронт, а во втором – отрицательный.

Реверсивный регистр. Этот тип регистра осуществляет сдвиг информации вправо и влево и имеет соответственно два информационных входа V_R (вправо) и V_L (влево), а также два информационных входа $D - D_R$ (вправо) и D_L (влево). Этот регистр функционирует как однонаправленный регистр сдвига вправо при $V_L=0$ и влево при $V_R=0$. Одновременная подача управляющих сигналов V_R, V_L должна быть запрещена, т. е. нужно обеспечить $V_L V_R=0$, что даст:

$$Q_k(t+1) = V_k Q_{k-1}(t) \vee V_L Q_{k+1}(t) \vee \bar{V}_L \bar{V}_R Q_k(t), k = 1, 2, \dots, n. \quad (4.17)$$

Условное графическое изображение такого типа регистра представлено на рис. 4. 5, г.

4. 5. Счетчики

Общие сведения. Счет импульсов – это одна из наиболее распространенных операций, выполняемых в цифровых устройствах. Счетчики можно классифицировать по разным признакам. Наиболее общая классификация – это три группы счетчиков: суммирующие, вычитающие и реверсивные, т. е. суммирующие или вычитающие в зависимости от управляющих сигналов.

Можно отметить два основных способа построения счетчиков: на основе триггеров и на основе регистров. При этом основными характеристиками будут модуль счета m , т. е. число, характеризующее количество устойчивых состояний счетчика, и быстродействие, которое характеризуется максимальной частотой поступления счетных импульсов.

Двоичный счетчик с последовательным переносом. Этот счетчик относится к счетчикам суммирующего действия. В качестве примера рассмотрим счетчик на микросхеме средней степени интеграции К155ИЕ5. Логическая схема счетчика представлена на рис. 4. 6, а, а условное графическое изображение – на рис. 4. 6, б [11]. Обозначение СТ2 на этом рисунке говорит о том, что это счетчик по $mod 2$, т. е. двоичный (от английского слова COUNTER – счетчик). Счетчик выполнен на синхронных JK - триггерах с асинхронным сбросом. Эту операцию выполняют входы $\&$ и R_0 . Особенностью счетчика является то,

что, если счет идет до 16 импульсов, то выход Q_1 следует соединить со входом C_2 , а считаеые импульсы подавать на вход C_1 . Переменный коэффициент счета – 2, 4, 8, 16 – обеспечивается на выходах Q_1, Q_2, Q_3, Q_4 .

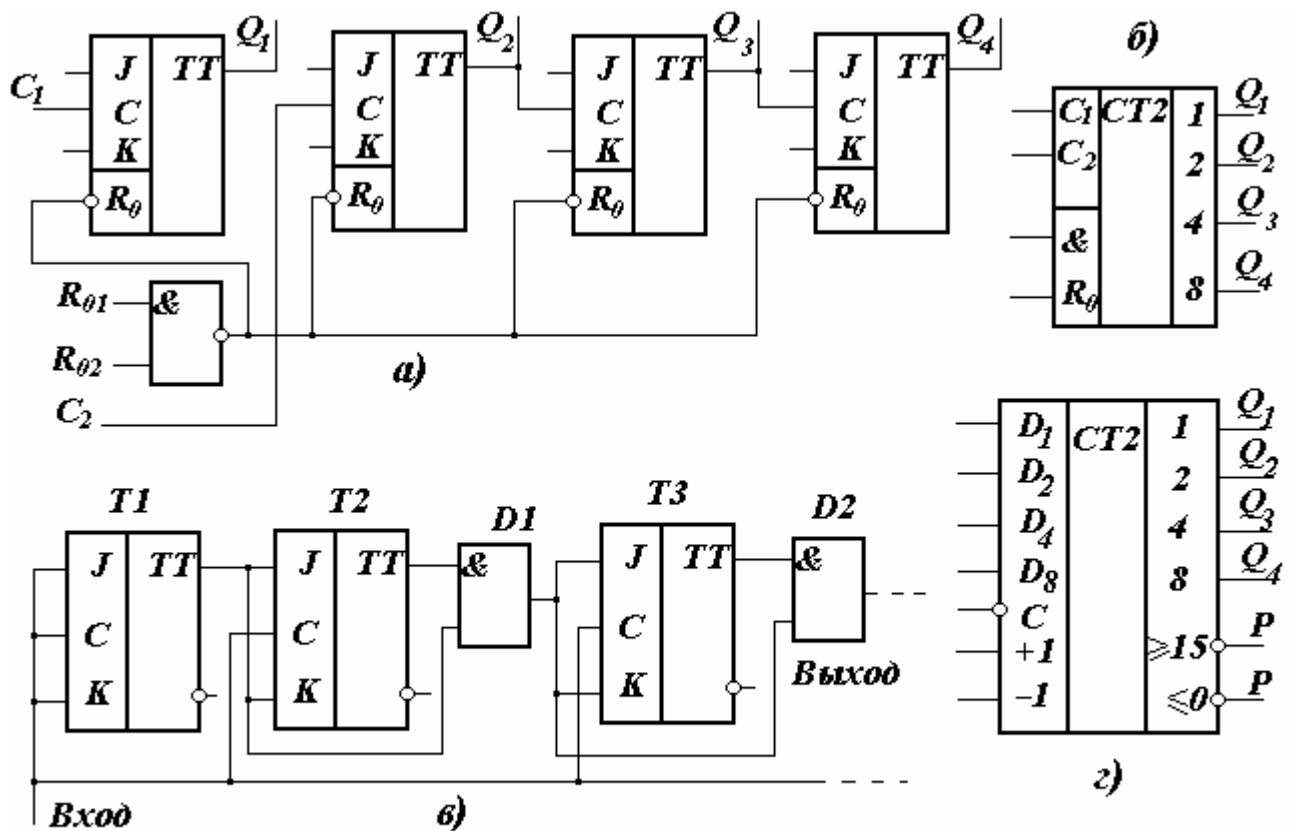


Рис. 4. 6

Быстродействие счетчиков с последовательным переносом характеризуется величиной задержки, зависящей от количества последовательно включенных триггеров: $t_{зад} = nt_{П}$. (4.18)

Двоичные счетчики с параллельным переносом. Параллельный перенос применяется для повышения быстродействия двоичных счетчиков. Здесь используется известная закономерность двоичных чисел: если к разряду с нулем прибавляется единица, то она будет в результате, а, если к разряду с единицей прибавить единицу, то в результате получим нуль, а единица перемещается в следующий разряд. Логическая схема счетчика с параллельным переносом представлена на рис. 4. 6, в.

Наличие элементов D_1, D_2 в этом счетчике объясняется тем, что триггер T_1 срабатывает от каждого счетного импульса на входе. При этом $Q_1 = J = K = 1$ для триггера T_2 после первого импульса. Следовательно T_2 работает от второго счетного импульса $Q_2 = 1; Q_1 = 0$. От третьего счетного импульса $Q_1 = J = K = 1$ для T_3 , который работает от четвертого импульса и т. д. Поэтому при одном и том же числе счетных разрядов предельная частота счетчика с параллельным переносом в n раз выше, чем у счетчика с последовательным переносом.

лельным («сквозным») переносом будет выше, чем в счетчике последовательного действия, и будет равна: $F_{\max} = 1 / (t_{\Pi} + (n - 2)t_{3\Pi})$, (4.19)

где t_{Π} – задержка в срабатывании триггера; $t_{3\Pi}$ – задержка в элементах D1, D2; $t_{\Pi} \gg t_{3\Pi}$.

Реверсивный счетчик. Этот тип счетчика выполняет как операцию суммирования, так и вычитания, а также предварительную запись информации. Рассмотрим в качестве примера интегральный счетчик на микросхеме K155IE7 (рис. 4. 6, з).

Входы D_1, D_2, D_4, D_8 называются информационными входами первого, второго, третьего и четвертого разрядов. Число, которое необходимо предварительно записать в счетчик, подается на эти разряды в параллельном коде. Во время работы счетчика в счетном режиме сигналы с этих входов на установочные входы блокированы. Запись числа на входы $D_1 \dots D_8$ производится по специальной команде – логический «0» на входе «С». В счетном режиме на входе «С» – логическая «1».

Вход «+1» предназначен для счета импульсов в режиме суммирования и называется *входом прямого счета*. Вход «-1» предназначен для подачи и счета импульсов в режиме вычитания – это вход *обратного счета*. Одновременная подача на входы «+1» и «-1» логической «1» запрещена.

В этом счетчике нет специальной команды реверса, и переход от суммирования к вычитанию или наоборот производится только за счет подачи входных импульсов на «+1» или «-1».

Вход R_0 – это асинхронный вход сброса во всех разрядах одновременно. В счетном режиме на входе R_0 должен быть логический «0».

Выходы 1, 2, 4, 8 – это прямые выходы счетных триггеров 1-го, 2-го, 3-го и 4-го разрядов. Инверсный выход « ≥ 15 » называется *выходом прямого переноса P*. на этот выход передается логический «0», когда при прямом счете на вход поступает пятнадцатый импульс. Если все разряды счетчика перед началом работы были установлены в «0», то выход « ≥ 15 » создаст 16-й входной импульс на входе «+1». При малой разрядности счетчика этот сигнал можно использовать как сигнал переноса для подачи на следующий счетчик.

При записи в счетчик предварительно какого-либо числа, отличного от нуля, номер входного импульса, дающего отклик на выходе « ≥ 15 », будет зависеть от этого числа. Под входными импульсами понимаются потенциальные сигналы отрицательной полярности.

Инверсный выход « ≤ 0 » – это выход *обратного переноса \bar{P}* . На этом выходе будет сигнал логического «0», равный по длительности очередному входному импульсу на входе «-1», когда все счетные триггеры в режиме вычитания установятся в логический «0» на прямых входах 1 ... 8.

Счетчики с произвольным модулем счета. К этой группе относятся счетчики, модуль счета которых не равен целой степени числа 2. Чаще всего такие счетчики строятся на регистрах сдвига с обратными связями.

Рассмотрим построение счетчика для случая коэффициента пересчета $N=3$. Для его построения потребуется взять за основу двоичный счетчик с числом разрядов $m = \lceil \log_2 N \rceil$, где выражение в квадратных скобках – ближайшее целое число, большее величины m . В рассматриваемом случае $\log_2 3 = 1,6$, т. е. потребуется два триггера.

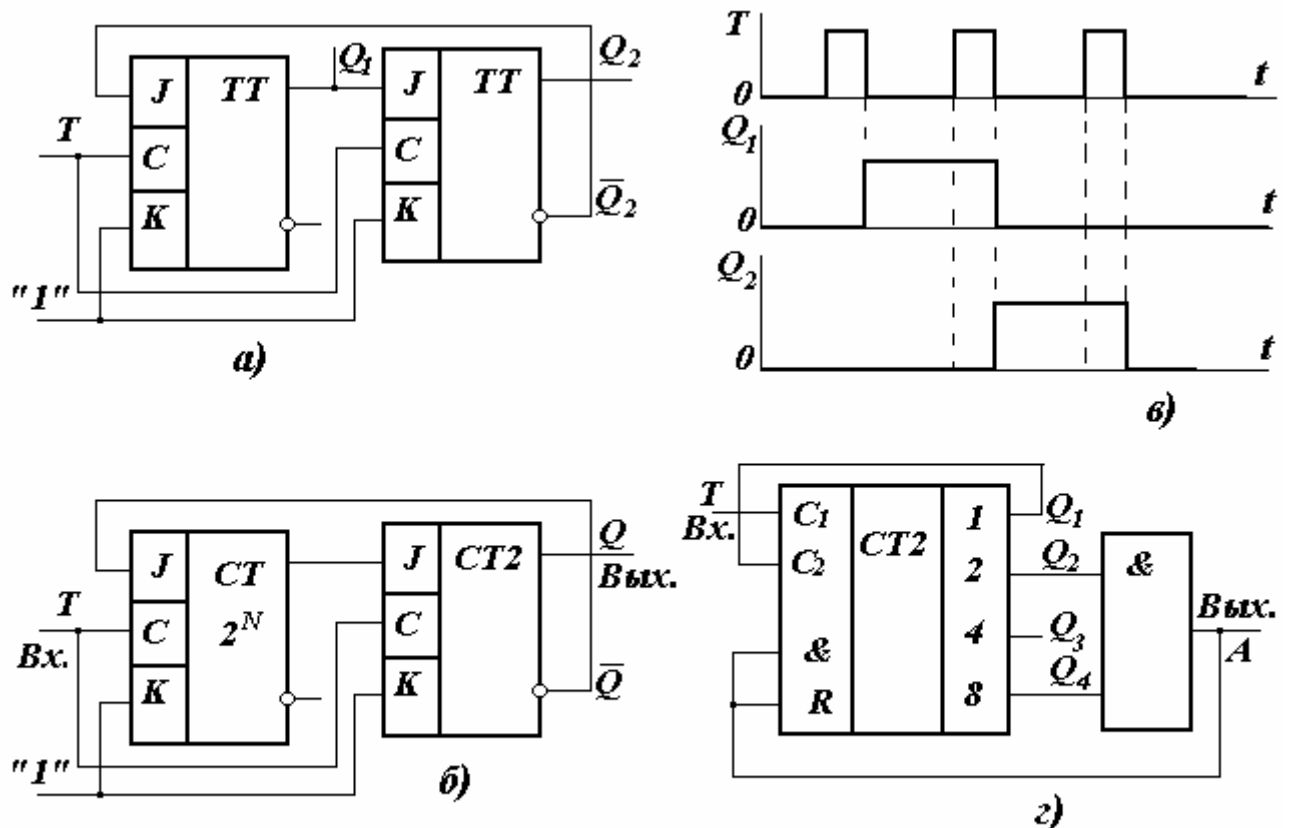


Рис. 4. 7

Если в исходном состоянии выходы триггеров (рис. 4. 7, а и б) $Q_1=Q_2=0$, то первый входной импульс установит счетчик в положение $Q_1 = 1; Q_2 = 0$; второй – в состояние $Q_1 = 0; Q_2 = 1$, а третий – снова в исходное состояние $Q_1=Q_2=0$. На вход K постоянно подается логическая «1».

Построение счетчика $N = 3$ выполнено как бы путем увеличения на 1 коэффициента пересчета $N = 2$. По этой методике можно построить общий алгоритм получения недвоичного счетчика по принципу $2^N + 1$ с помощью дополнительного счетчика СТ2 (рис. 4. 7, в).

Декадный счетчик. Счетчик с коэффициентом пересчета $N = 10$ носит название декадного. Такой коэффициент пересчета применяется сравнительно часто. Его удобно организовать на стандартной микросхеме К155ИЕ5 с коэффициентом пересчета $N = 16$. Для получения требуемого модуля счета необходимо при обнаружении числа 10_{10} (1010_2) сбрасывать счетчик в нулевое положение и начинать счет сначала, как показано на рис. 1. 23, з.

Если построить для этого случая карту Карно, то искомое число 1010_2 будет получаться, когда $A = Q_4 \bar{Q}_3 Q_2 \bar{Q}_1$, или с учетом факультативных условий $A = Q_2 Q_4$.

Следует отметить, что в качестве декадного можно использовать стандартный счетчик на микросхеме K155ИЕ2, имеющий коэффициент пересчета 10 и выполненный в одном корпусе.

Вопросы и задания для самоконтроля

1. Что такое цифровой автомат?
2. Приведите аналитическое выражение, описывающее работу цифрового автомата.
3. Чем определяется быстродействие конечного автомата?
4. Опишите работу триггера как конечного автомата (последовательного устройства).
5. Приведите таблицу переключений простейшего асинхронного триггера.
6. Начертите карту Карно для RS-триггера с инверсными входами.
7. Покажите, как получить уравнение возбуждения для этого триггера.
8. В чем особенность E-триггера?
9. Как построить обобщенную карту Карно для этого типа триггера?
10. Покажите, как построить таблицу переключений для JK- триггера?
11. Что такое синхронный триггер?
12. Приведите схему и временные диаграммы для синхронного RS- триггера?
13. Приведите логическое уравнение для D-триггера.
14. В чем особенность MS-триггера?
15. Дайте определение параллельного и последовательного регистров.
16. Для чего в регистрах применяется синхронный принцип работы?
17. В чем особенность построения регистров сдвига влево и вправо, а также реверсивного регистра?
18. Как классифицируются счетчики импульсов?
19. Приведите схему двоичного счетчика с последовательным переносом.
20. Для чего применяются счетчики с параллельным переносом? Приведите схему на логических элементах.
21. Как строятся счетчики с произвольным модулем счета? Приведите пример.
22. Приведите пример построения декадного счетчика на микросхеме средней степени интеграции (например, K155ИЕ5).

5. АНАЛОГО – ЦИФРОВЫЕ И ЦИФРОАНАЛОГОВЫЕ ПРЕОБРАЗОВАТЕЛИ

5.1. Аналого–цифровые преобразователи (АЦП)

Принцип аналого-цифрового преобразования. Чаще всего получаемый непосредственно от источника информации сигнал представлен в форме непрерывно меняющегося по значению напряжения либо тока (рис. 5.1, а). Таков, например, характер электрического сигнала, соответствующего телефонным, телевизионным и другим видам сообщений. Для передачи таких сообщений по линии связи или для их обработки (например, при фильтрации помех) возможно использование двух форм: *аналоговой* или *цифровой*.

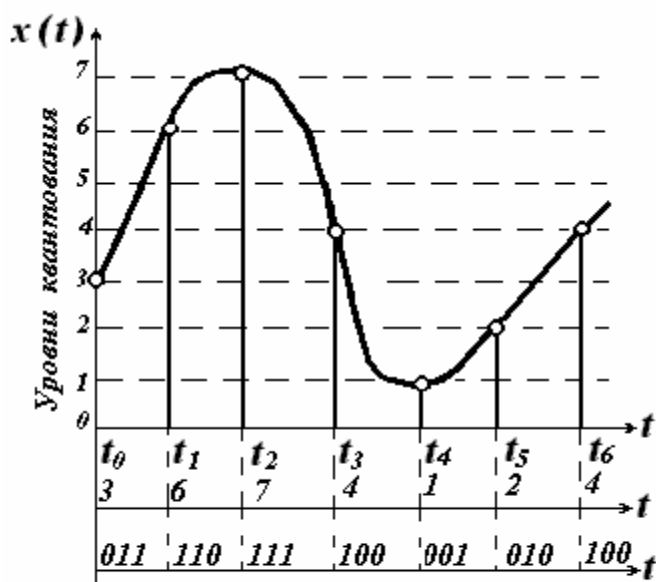


Рис. 5.1

Преобразование сигналов из аналоговой формы в цифровую выполняется в устройстве, называемом *аналого-цифровым преобразователем* (АЦП). В АЦП обычно выделяются следующие процессы: *дискретизация*, *квантование* и *кодирование*.

Дискретизация непрерывных сигналов. В процессе дискретизации из непрерывного во времени сигнала выбираются отдельные его значения, соответствующие моментам времени, следующим через определенный временной интервал T (на рис. 5.1 моменты $t_0, t_1 \dots$). Интервал T называется *тактовым интервалом*, а

моменты t_0, t_1, \dots , в которые берутся отсчеты, — *тактовыми моментами* времени.

Дискретные значения сигнала обычно отсчитываются с таким малым тактовым интервалом T , чтобы по ним можно было восстановить сигнал в аналоговой форме с необходимой точностью.

Квантование и кодирование. При выполнении этих операций создается *сетка уровней квантования* (рис. 5.1), сдвинутых друг относительно друга на величину Δ , называемую *шагом квантования*. Каждому уровню квантования приписывается порядковый номер (0, 1, 2, 3, 4, и т.д.). Полученные в результате дискретизации значения исходного аналогового напряжения заменяются ближайшими к ним уровнями квантования. На рис. 5.1 напряжение в момент t_0 за-

меняется ближайшим к нему уровнем квантования с номером 3, в тактовый момент t_1 значение напряжения близко к уровню 6 и заменяется этим уровнем и т.д. Этот процесс носит название *операции квантования*, которая по-существу состоит в округлении значений аналогового напряжения, выбранных в тактовые моменты времени. Округление приводит к ошибкам (*шум квантования*). При проектировании АЦП стремятся снизить шум квантования до предельно малой величины.

Другая важная операция, выполняемая при АЦП – это *кодирование*. Смысл этой операции состоит в том, что при округлении значений напряжения при квантовании, получаются числа, соответствующие уровням квантования. Так, на рис. 5.1 образуется последовательность чисел: 3, 6, 7, 4, 1, 2 и т.д. Эта последовательность чисел представляется двоичным кодом.

Таким образом, АЦП производит преобразование аналоговой величины, например, мгновенное значение напряжения, в цифровой код. Это происходит при сравнении этого напряжения с эталонным, как показано на рис. 5.2, а.

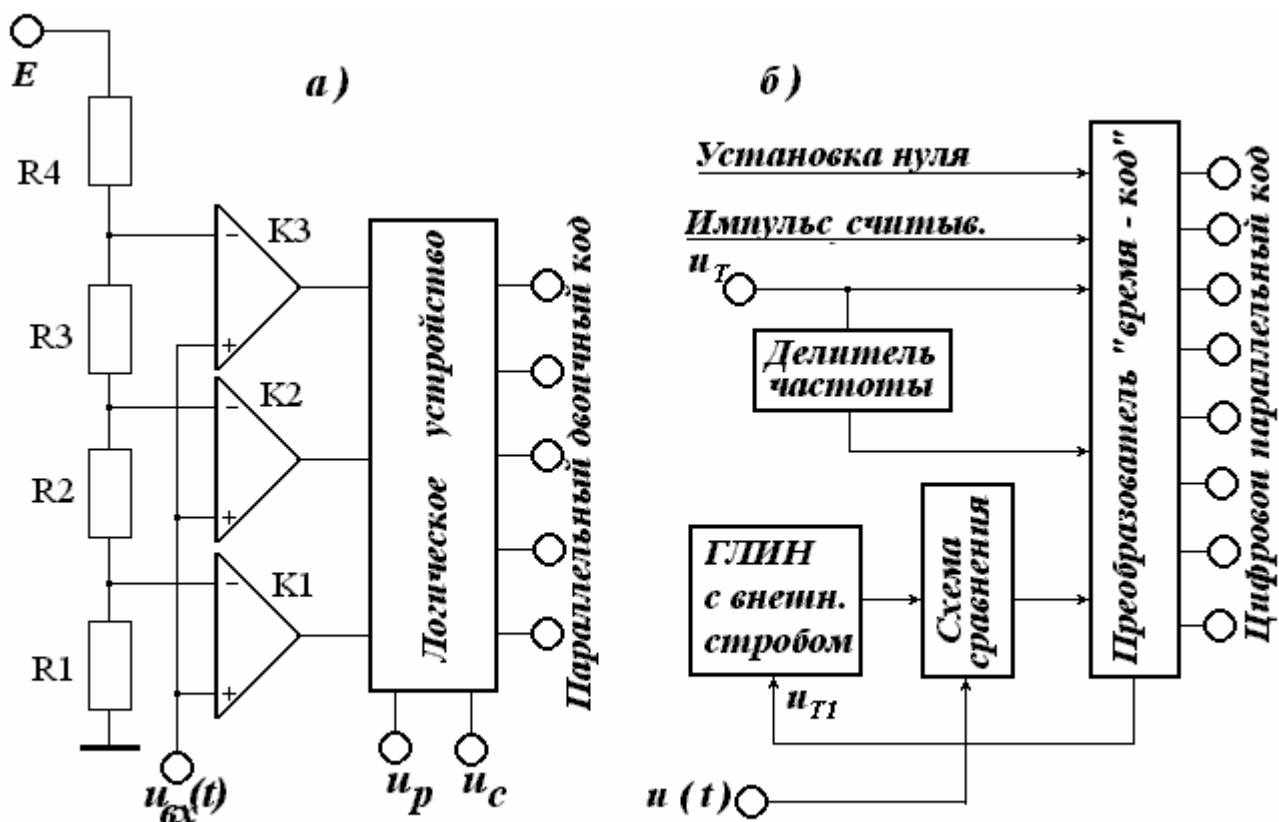


Рис. 5.2

На этом рисунке показан АЦП *параллельного* типа, в котором датчик уровней эталонных напряжений образуется при подаче напряжения на *прецизионный делитель*, состоящий из цепочки резисторов R1 ... R4. Полученные эталонные напряжения поступают на компараторы K1 ... K3. Если сигнал $u_{вх}(t)$

оказывается больше опорного, то происходит срабатывание компаратора: на его выходе получается логическая «1».

Логическое устройство в этой схеме записывает выходные сигналы компараторов в регистр (при наличии сигнала регистрации u_p), преобразует сигнал логической «1» в регистре в двоичный код с того компаратора, у которого был наибольший уровень опорного напряжения, и передает двоичный код на выход (при наличии сигнала списывания u_c).

АЦП параллельного типа отличаются достаточно высоким быстродействием, так как опорные напряжения существуют у них постоянно, и на их формирование не расходуется дополнительное время. Но точность такого АЦП ограничена, так как ограничено число каналов. Требуется, чтобы разница между соседними уровнями квантования была существенно больше взаимной нестабильности порогов срабатывания соседних компараторов.

На рис. 5.2, б показано АЦП с промежуточным преобразованием напряжения во временной интервал. Сам преобразователь «время - код» показан на рис. 5.3, а временная диаграмма, поясняющая его работу – на рис. 5.4.

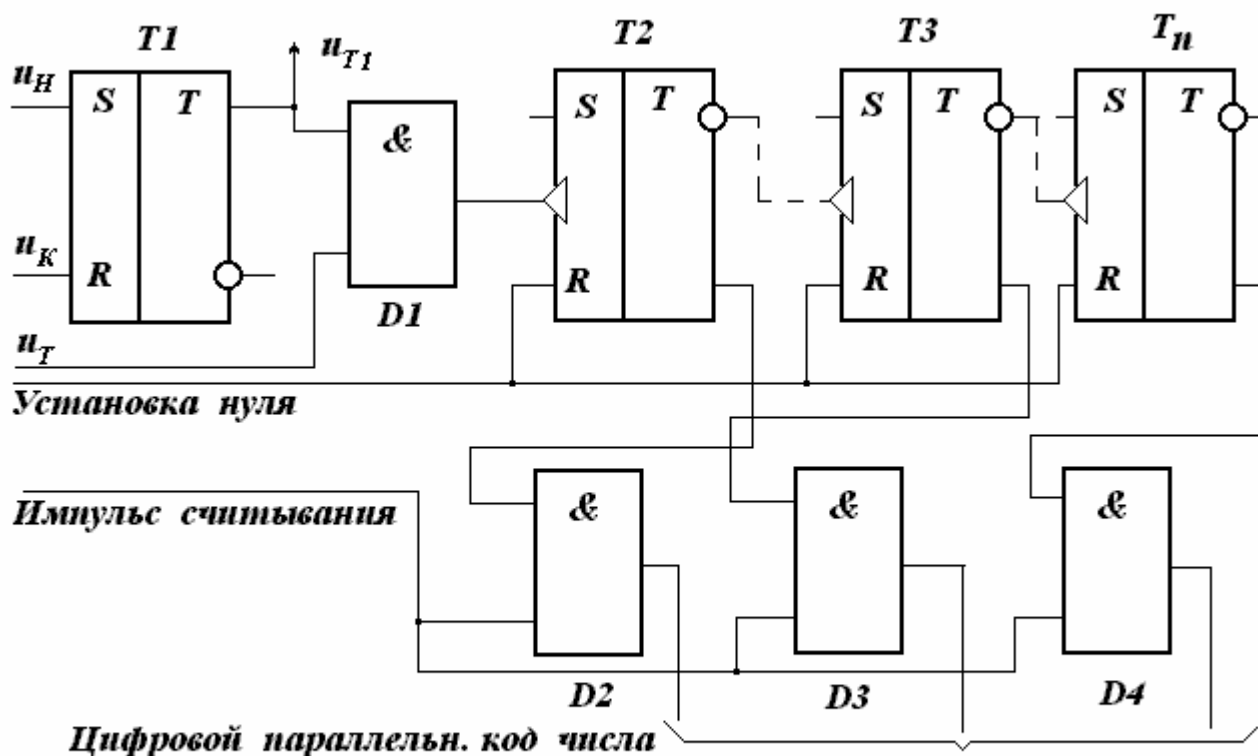


Рис. 5.3

Работу преобразователя «время - код» можно кратко описать следующим образом. Тактовые импульсы u_T поступают на делитель частоты, на выходе которого получают импульсы начала u_H для запуска преобразователя «время - код». При этом на выходе триггера $T1$ формируется сигнал логической «1», который служит стробом для ГЛИН (сигнал u_{T1}). При формировании линейно нарастающего напряжения происходит его сравнение в компараторе (схема

сравнения) с исследуемым напряжением, и в момент равенства этих напряжений вырабатывается сигнал конца (u_k), сбрасывающий триггер $T1$ в ноль.

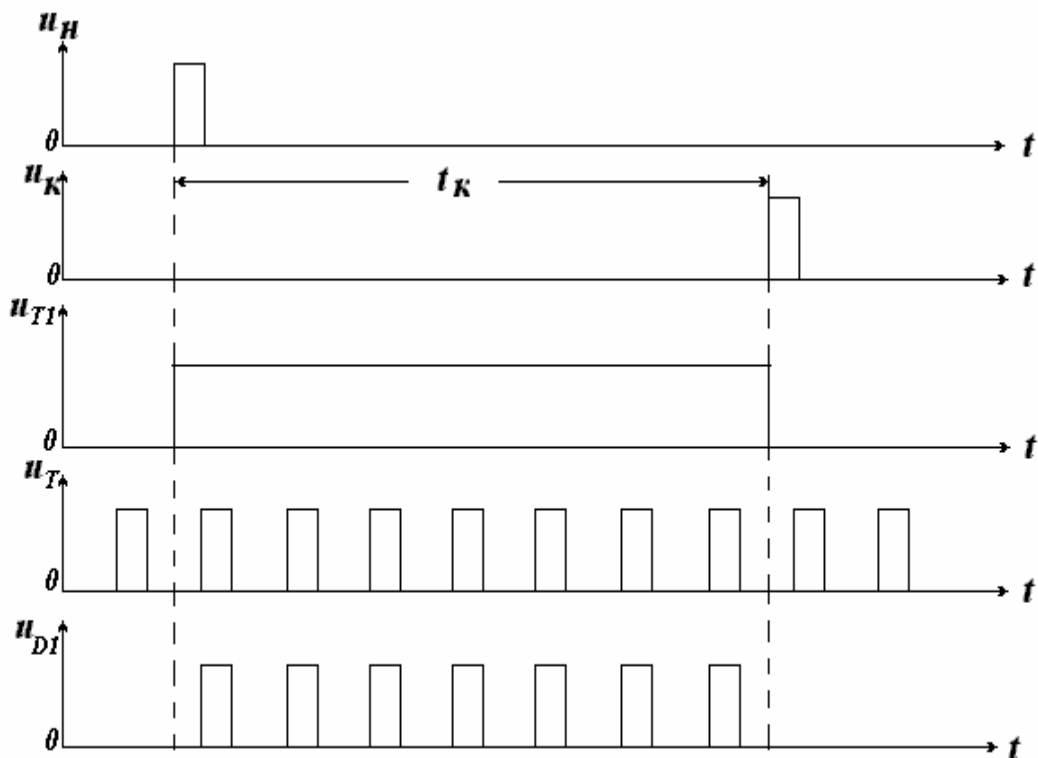


Рис. 5.4

На время действия строба работает счетчик импульсов на триггерах $T2$, $T3 \dots T_n$, сигналы с которых поступают на схемы совпадения $D2, D3 \dots D_n$. При поступлении импульса считывания на выходе схем совпадения формируется параллельный код числа.

5.2. Цифроаналоговые преобразователи (ЦАП)

Среди устройств ЦАП наиболее распространены устройства, преобразующие параллельный двоичный код в напряжение. Если воспользоваться формулой представления двоичных чисел

$$N = a_0 2^0 + \dots + a_{n-1} 2^{n-1} = \sum_{i=0}^{n-1} a_i 2^i, \quad (5.1)$$

операцию преобразования «код - напряжения» можно представить в виде [5]

$$U(N) = (U_{\max} / N_{\max}) \cdot N = U_{\max} \sum_{i=0}^{n-1} \frac{2^i \cdot a_i}{2^n - 1} = \sum_{i=0}^{n-1} u_i a_i, \quad (5.2)$$

где $U_{\max} / N_{\max} = U_{\max} / (2^n - 1) = U(1)$, т. е. представляет собой масштабный коэффициент, под которым подразумевается напряжение, соответствующее единице входного сигнала.

Из (5.2) следует, что рассматриваемое преобразование сводится к суммированию элементарных напряжений:

$$u_i = \frac{U_{max} 2^i}{(2^n - 1)} \approx \frac{U_{max}}{2^{n-1}} = U_{max} \cdot K_i. \quad (5.3)$$

Элементарные напряжения u_i образуются с помощью деления некоторого эталонного напряжения U_{max} резистивными делителями с коэффициентом передачи $K_i = 1/2^{n-1}$. Из операции суммирования исключаются те слагаемые u_i , которые соответствуют нулевым значениям элементов a_i , составляющих входное двоичное число: $\{a_{n-1} a_{n-2} \dots a_1 a_0\}$.

Принцип деления напряжения реализуется с помощью ключевых схем (рис. 5.5, а и б).

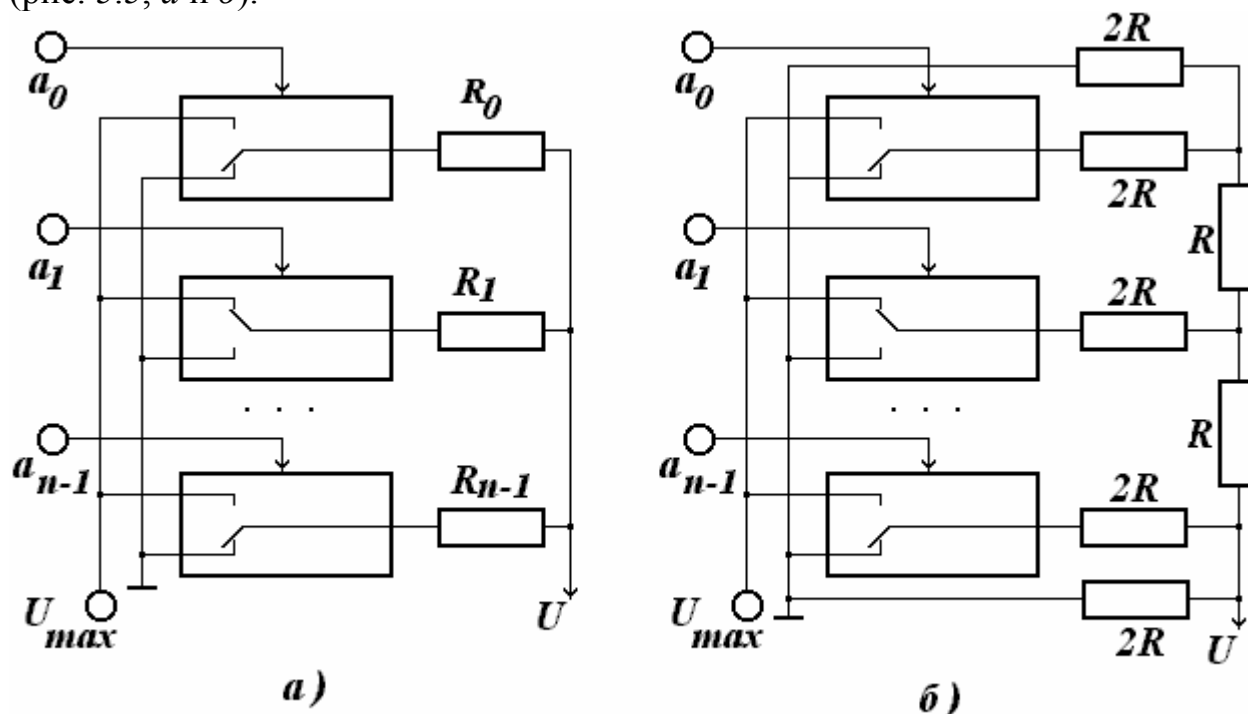


Рис. 5.5

Под ключом подразумевается достаточно быстродействующее электронное устройство. На рис. 5.5, а изображена схема с двоично – взвешенными резистивными цепями, т.е. схема с суммированием напряжений. На рис. 5.5, б изображен преобразователь «код - напряжение» с многозвенной резистивной цепью типа $R - 2R$ (читается: «эр – два эр»). Эта схема чаще находит применение, когда производится не деление напряжений, а деление токов. В этой схеме резисторы имеют только два номинала, что делает ее удобной при интегральной технологии.

На рис. 5.6 приведена структурная схема ЦАП в виде выпускаемой промышленностью микросхемы К572ПА1 [5].

Если на цифровом входе i – го разряда $a_i = 1$, то усилитель откроет ключ, через который разрядный ток этого разряда резистивной матрицы поступит на *Выход 1*. В случае $a_i = 0$ откроется другой ключ, через который ток по-

ступит на *Выход 2*. При наличии сигналов на всех разрядах результирующий ток будет равен:

$$I_{\text{вых1}} = \sum_{i=1}^n a_i I_i = \sum_{i=1}^n a_i 2^{-i} ; \quad I_{\text{вых2}} = I_0 \sum_{i=1}^n a_i 2^{-i} . \quad (5.4)$$

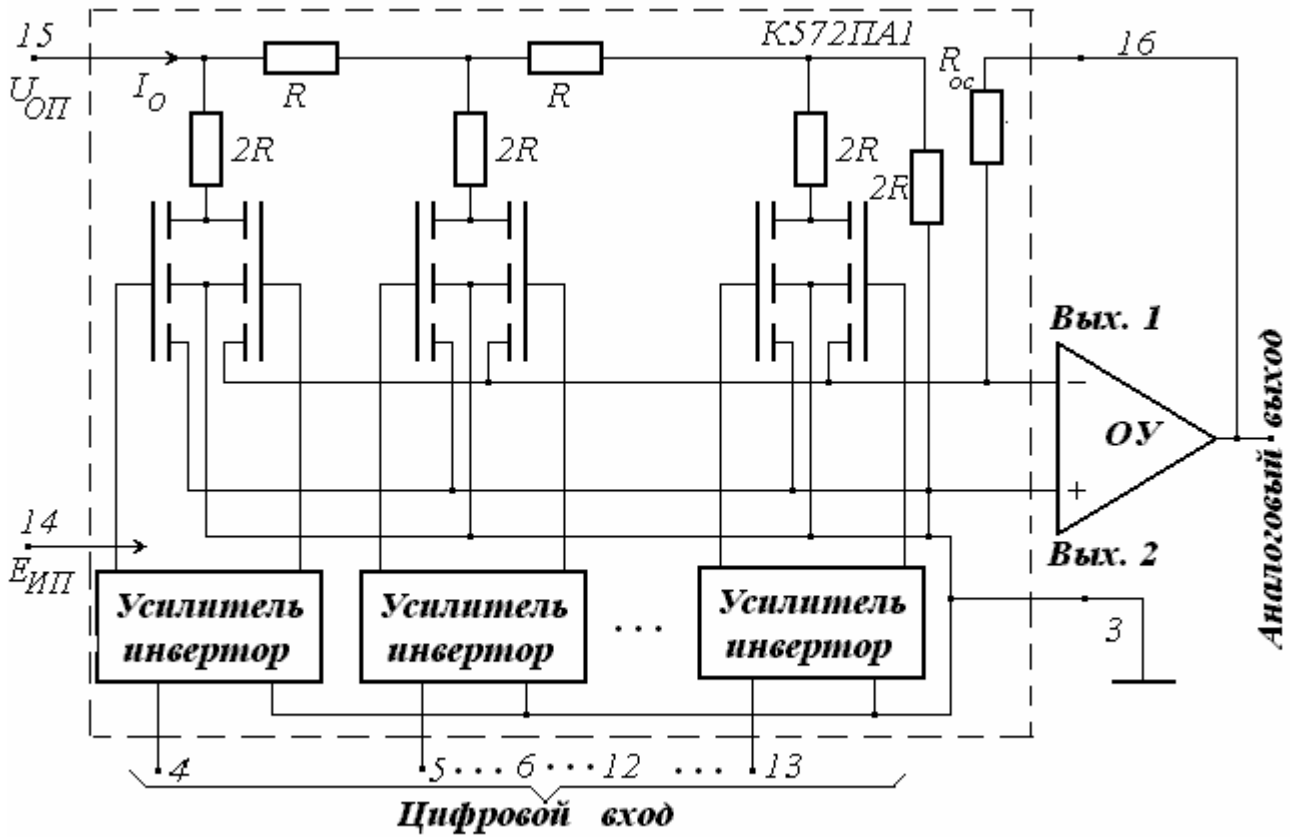


Рис. 5. 6

Если необходимо иметь на выходе напряжение, что встречается чаще, то к микросхеме К572ПА1 подключается операционный усилитель. На аналоговом выходе ЦАП напряжение будет равно:

$$U_{\text{вых}} = I_{\text{вых1}} R_{oc} = I_0 R_{oc} \sum_{i=1}^n a_i 2^{-i} . \quad (5.5)$$

С учетом очевидных соотношений:

$$I = \frac{U_{on}}{R} ; \quad N = \sum_{i=1}^n a_i 2^{-i} . \quad (5.6)$$

выходное напряжение на аналоговом выходе, пропорциональное величине входного числа, можно представить в виде:

$$U_{\text{вых}} = U_{on} R_{oc} N / R . \quad (5.7)$$

Следует также отметить, что использование цифровой формы представления сигналов может обеспечить более высокую помехоустойчивость и стабильность параметров обработки, а также независимость от влияния изменений в окружающей среде.

Вопросы и задания для самоконтроля

4. В чем заключается принцип аналого – цифрового преобразования?
5. Как называется устройство, преобразующее сигнал из аналоговой формы в цифровую?
6. Что такое дискретизация непрерывных сигналов?
7. Какие операции называются квантованием и кодированием?
8. Что понимается под шагом и уровнем квантования?
9. Из-за чего получаются ошибки в работе АЦП?
10. Как связаны ошибка в работе АЦП и шаг квантования?
11. От чего зависит величина аппаратурных погрешностей ЦАП?
12. Как происходит преобразование аналоговой величины в цифровой код?
13. Покажите на простейшей схеме, как происходит сравнение аналоговой величины с эталонным напряжением.
14. Как работает АЦП параллельного типа?
15. Что такое АЦП с промежуточным преобразованием напряжения во временной интервал?
16. Начертите простейшую схему преобразователя «время – код».
17. Начертите временную диаграмму работы преобразователя «время – код».
18. Как называется устройство, преобразующее параллельный двоичный код в напряжение?
19. Начертите схему ЦАП с суммированием напряжений.
20. Что такое преобразователь «код – напряжение» с многозвенной резистивной цепью $R - 2R$?
21. Как выглядит упрощенная структурная схема ЦАП на микросхеме К572ПА1?
22. Покажите, как подключается операционный усилитель к выходу микросхемы К572ПА1, и что этим достигается.
23. Какова помехоустойчивость и стабильность параметров аппаратуры при цифровом способе обработки?

6. ПОЛУПРОВОДНИКОВЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА

6.1. Основные типы и параметры запоминающих устройств

Типы запоминающих устройств. Для хранения небольших массивов кодовых слов обычно используются регистры. Для хранения больших массивов слов строят *запоминающие устройства (ЗУ)* с использованием специальных микросхем, каждая из которых может хранить информацию объемом в тысячи битов.

По выполняемым функциям ЗУ можно классифицировать на следующие типы: *оперативное ЗУ (ОЗУ), постоянное ЗУ (ПЗУ), перепрограммируемое ПЗУ (ППЗУ)*.

ОЗУ предназначено для использования в условиях, когда необходимо выбирать и обновлять хранимую информацию с достаточно высокой скоростью. В ОЗУ предусматривается три режима работы: *режим хранения* при отсутствии обращения к ЗУ, *режим чтения* хранимых слов и *режим записи новых слов*. При этом в режимах чтения и записи ОЗУ должно функционировать с высоким быстродействием (доли микросекунды).

ПЗУ используется для хранения постоянной информации, которая не нарушается и при отключении источников питания. При этом предусмотрены два режима работы: *режим хранения* и *режим чтения с высоким быстродействием*. Режим записи пользователем не предусмотрен.

ППЗУ отличается от ПЗУ тем, что допускает обновление однажды занесенной информации, т. е. здесь предусмотрен *режим записи*. ППЗУ в процессе работы цифрового устройства обычно используется как ПЗУ.

Параметры ЗУ. ЗУ содержит некоторое число N ячеек, в каждой из которых может храниться слово с определенным числом n разрядов. Ячейки нумеруются двоичными числами. Номер ячейки называется *адресом*. Если для представления адресов используются комбинации m – разрядного двоичного кода, то число ячеек в ЗУ составит $N = 2^m$ [3].

Количество информации, которое может храниться в ЗУ, определяет его *емкость*. Емкость выражается числом ячеек N с указанием разрядности n хранимых в них слов в форме $N \times n = M$ бит. Часто разрядность ячеек выбирают кратной байту (8 бит). Большие значения емкости часто выражаются в единицах $K = 2^{10} = 1024$. Например, $M = 64 \text{ Кбайт} = 64 \cdot 1024 \cdot 8$ бит.

Быстродействие ЗУ характеризуется двумя параметрами: *временем выборки* t_B , т. е. интервалом времени между моментом подачи сигнала выборки и появлением данных на выходе, и *циклом записи* $t_{ЦЗ}$, определяемым минимально допустимым временем между моментом подачи сигнала выборки при записи m моментом, когда разрешено последующее обращение к памяти.

Микросхемы ППЗУ дополнительно характеризуются временем хранения записанной в них информации, а также допустимым количеством циклов перезаписи.

6.2. Оперативное запоминающее устройство (ОЗУ)

Типовая структура микросхемы ОЗУ представлена на рис. 6.1 [3]. Информация хранится в *накопителе*, который представляет собой матрицу, составленную из элементов памяти (ЭП), расположенных вдоль строк и столбцов. ЭП может хранить 1 бит информации (*лог. 0* либо *лог. 1*). ЭП снабжен управляющими цепями для установки элемента в одном из трех режимов: *режиме хранения*, в котором он отключается от входа и выхода микросхемы, *режиме чтения*, в котором информация выдается на выход, *режиме записи*, в котором в ЭП записывается новая информация, поступающая на вход.

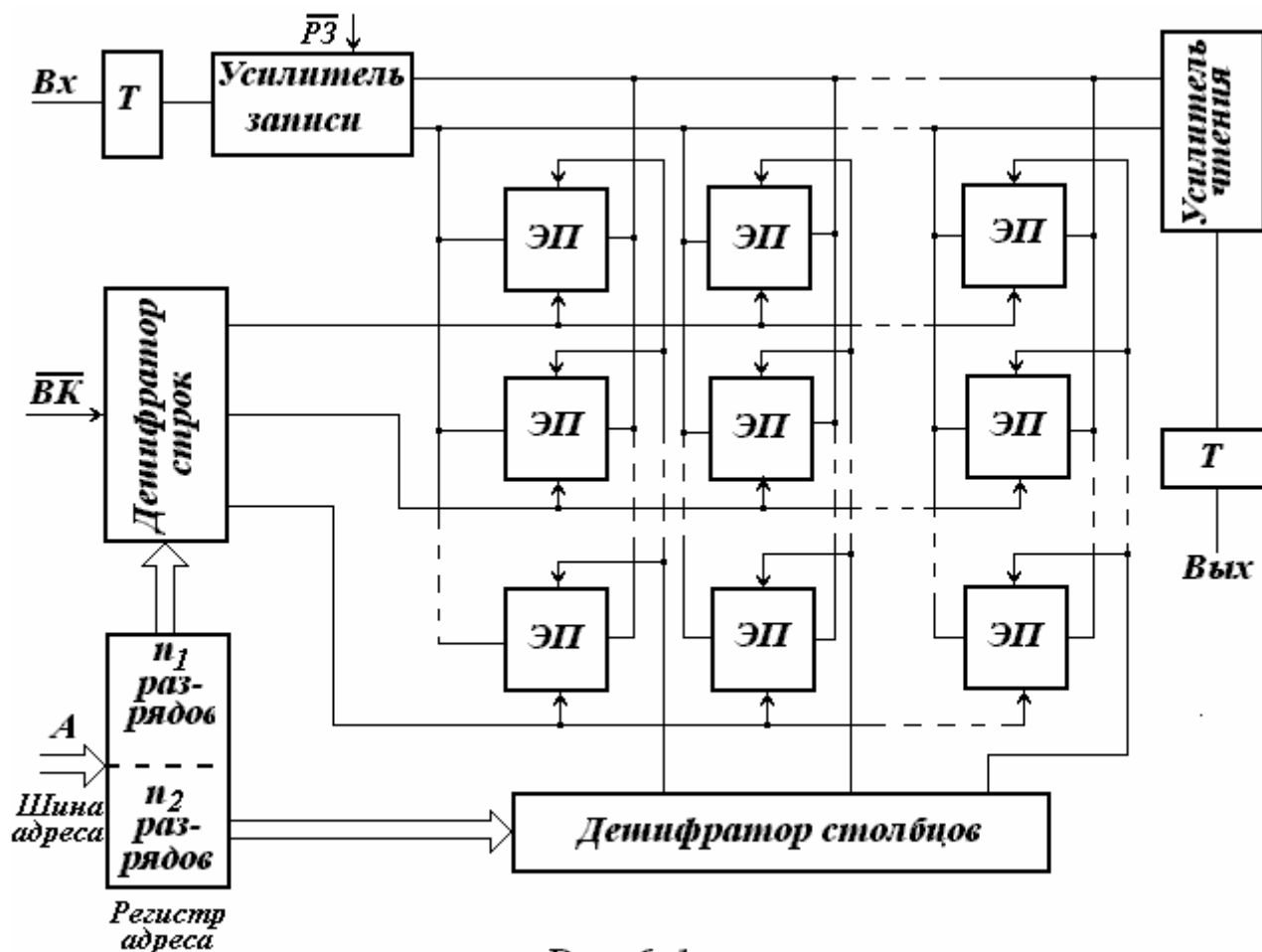


Рис. 6.1

У каждого ЭП имеется номер, который называется *адресом* элемента. Для поиска необходимого ЭП указывается строка и столбец, соответствующие положению ЭП в накопителе. Адрес ЭП в виде двоичного числа принимается по шине адреса в регистр адреса. Число разрядов адреса связано с емкостью накопителя, а число строк и столбцов выбирается равными целой степени числа двух. Если число строк $N_{СТР} = 2^{n_1}$ и число столбцов $N_{СТ} = 2^{n_2}$, то емкость накопителя $N = N_{СТР} \cdot N_{СТ} = 2^{n_1} \cdot 2^{n_2} = 2^{n_1+n_2} = 2^n$, где n - число разрядов адреса, принимаемого в регистр адреса.

Например, при емкости $N = 2^{10} = 1024$ число разрядов адреса $n = 10$; при этом выбирается $n_1 = n_2 = n/2 = 5$. В этом случае число строк и число столбцов накопителя $2^{n_1} = 2^{n_2} = 32$.

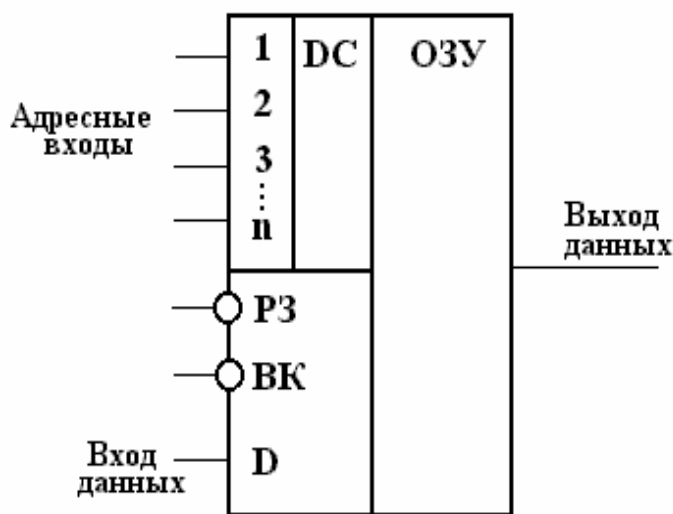


Рис. 6. 2

Разряды регистра адреса делятся на две группы: одна группа из n_1 разрядов определяет двоичный номер строки, в которой в накопителе расположен ЭП, другая группа из n_2 разрядов – двоичный номер столбца, в котором расположен выбираемый ЭП. Каждая группа разрядов адреса подается на соответствующий дешифратор: *дешифратор строк* и *дешифратор столбцов*. Каждый из дешифраторов создает на одной из своих выходных цепей уровень

$\log.1$ (на остальных выходах дешифратора устанавливается уровень $\log.0$), выбранный ЭП оказывается под воздействием уровня $\log.1$ одновременно по цепям строки и столбца.

При чтении содержимое ЭП выдается на *усилитель чтения* и с него – на *выходной триггер* (Т) и выход микросхемы.

Режим записи устанавливается подачей сигнала на вход разрешения записи ($\overline{P3}$). При уровне $\log.0$ на входе $\overline{P3}$ открывается усилитель записи, и бит информации со входа данных поступает в выбранный ЭП и запоминается в нем. Эти процессы происходят в том случае, если на входе выбора кристалла (\overline{BK}) действует активный уровень $\log.0$. При уровне $\log.1$ на этом входе на всех выходах дешифратора устанавливается уровень $\log.0$ и ЗУ оказывается в режиме хранения. Условное графическое обозначение микросхемы ОЗУ показано на рис. 6. 2.

Микросхемы ОЗУ допускают наращивание емкости памяти *наращиванием разрядности* (разрядности хранимых слов), а также *наращиванием числа ячеек* (числа слов в памяти). То есть имеется возможность строить память с требуемой организацией.

Схема наращивания разрядности ячеек представлена на рис. 6. 3. На все микросхемы подается один и тот же адрес. При чтении каждой микросхемой выдается определенный разряд считываемого слова. При записи входное слово поразрядно заносится в ЭП отдельных микросхем.

Если микросхемы имеют организацию $N \times 1$ (N одноразрядных ячеек), то для блока памяти с организацией $N \times n$ (N ячеек с разрядностью каждой n) потребуется n микросхем.

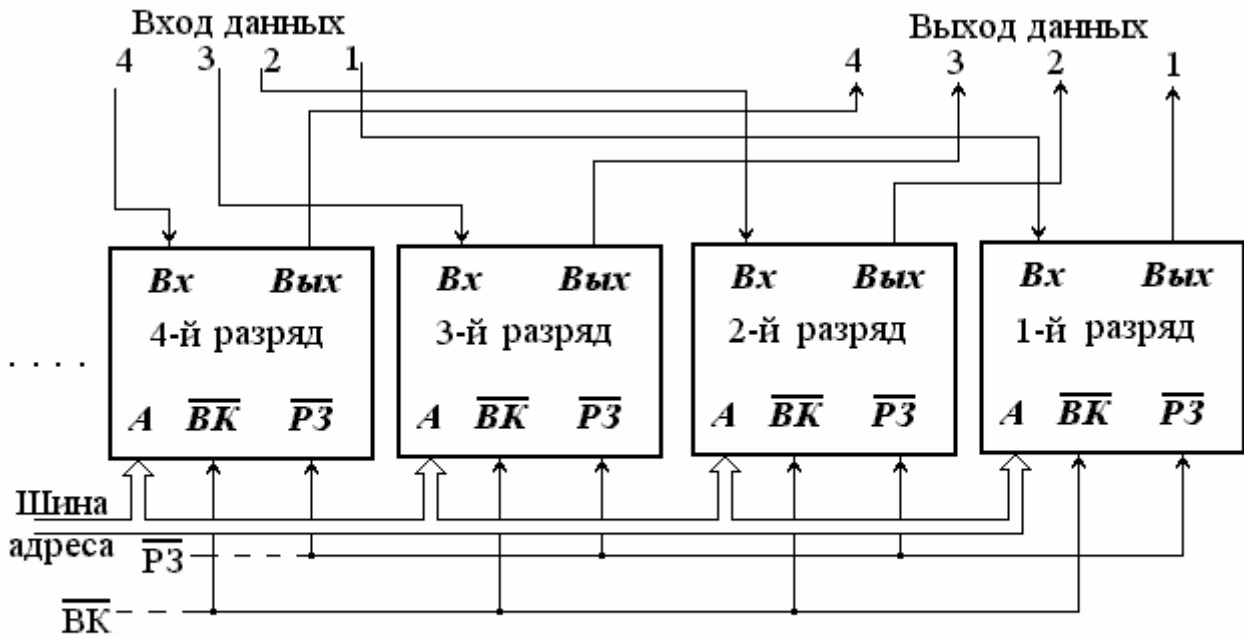


Рис. 6. 3

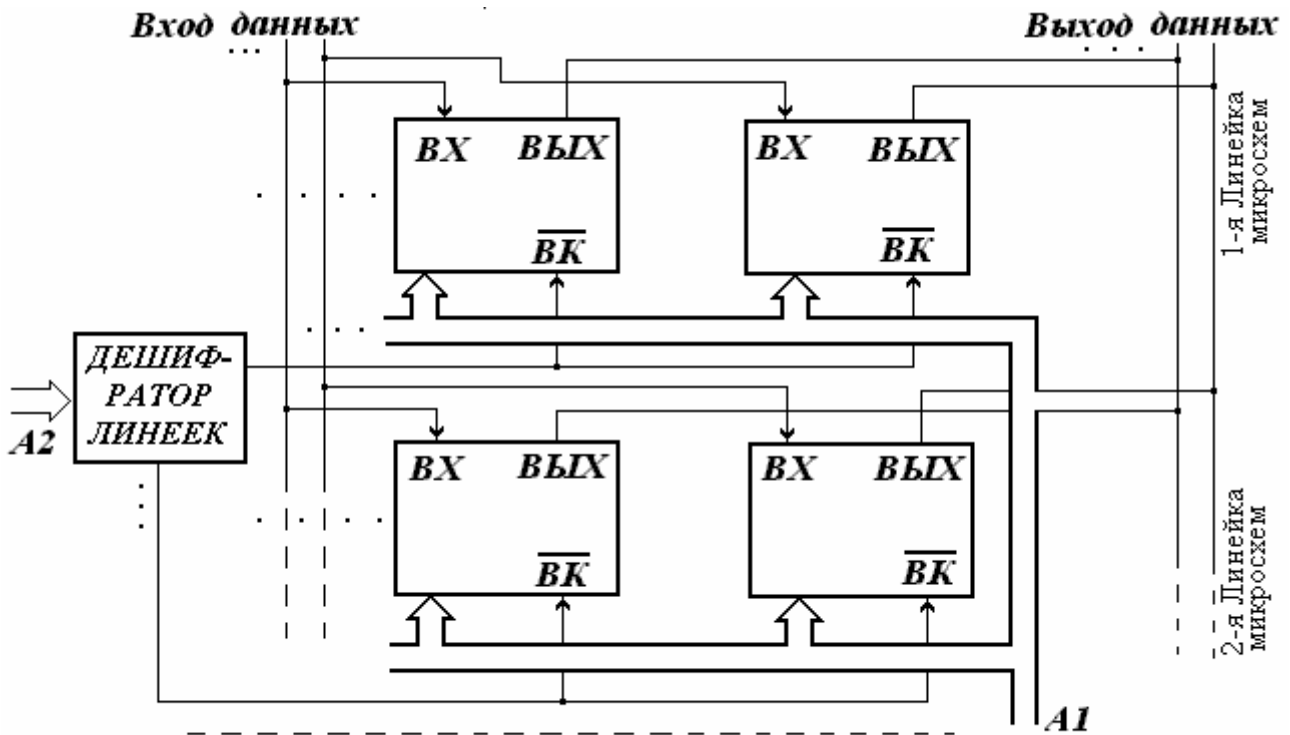


Рис. 6. 4

На рис. 6.4 представлена схема наращивания числа ячеек и их разрядности. Блок памяти состоит из микросхем, образующих отдельные *линейки (ряды)*, каждая из которых строится по схеме наращивания разрядности. Разряды адреса блока памяти делятся на две группы A_1 и A_2 . Группа разрядов A_2 определяет номер линейки, группа разрядов A_1 – номер ячейки в выбранной линейке.

Выбор линейки производится с помощью дешифратора, на вход которого подается A_2 , а каждый из выходов подключен к входу \overline{BK} определенной линейки.

В зависимости от кодовой комбинации, содержащейся в A_2 , на соответствующем выходе дешифратора появляется уровень $\text{лог.}0$, который обеспечивает выбор определенной линейки микросхем. На входы \overline{BK} остальных линеек с выходов дешифратора поступает уровень $\text{лог.}1$, и микросхемы этих линеек устанавливаются в режим хранения. Они не реагируют на адресную группу A_1 .

Для примера предположим, что необходимо наращивать емкость блока памяти на микросхемах с организацией 1024×1 таким образом, чтобы получить организацию 4096×8 , т. е. блок памяти на 4096 8-разрядных ячеек.

Наращивание разрядности потребует в каждой линейке схемы на рис. 6.4 использовать 8 микросхем: для увеличения числа ячеек в четыре раза необходимо предусмотреть четыре линейки микросхем, и общее число требуемых микросхем $8 \cdot 4 = 32$.

Формирование адреса в таком блоке памяти, по которому будет производиться обращение, происходит следующим образом. Для выбора линейки в адресе потребуется двухразрядная группа A_2 , каждой из четырех кодовых комбинаций этой группы (00, 01, 10, 11) будет соответствовать определенная линейка в блоке памяти. Выбор ячейки в линейке микросхем предполагает наличие в адресе 10-разрядной группы A_1 (число комбинаций 10-разрядной группы $2^{10} = 1024$ равно числу ЭП в микросхеме). Адрес рассматриваемого блока памяти должен иметь 12 разрядов.

В каждом столбце матрицы микросхем на рис. 6.4 выходы всех микросхем объединяются в цепь соответствующего разряда *выхода данных* блока, все *входы данных* – в цепь соответствующего разряда входа данных блока памяти.

6.3. Постоянные запоминающие устройства

Главное отличие ПЗУ от ОЗУ заключается в том, что информация в ячейки памяти записывается однократно, а в процессе эксплуатации используется только *режим чтения*.

По способу занесения информации ПЗУ делятся на два вида: ПЗУ, *программируемые маской* на заводе – изготовителе, и ПЗУ, *программируемые пользователем*.

Если ПЗУ программируется в процессе изготовления микросхем, то информация заносится с помощью фотошаблона. Такой способ записи пригоден, когда производится выпуск крупной партии ПЗУ с одинаковой информацией.

В ПЗУ, программируемых пользователем, запись информации производится с помощью специальных устройств – *программаторов*. Программатор выдает в микросхему напряжение для записи информации, набираемой на клавиатуре. Этими напряжениями производится прожигание плавких перемычек в элементах памяти.

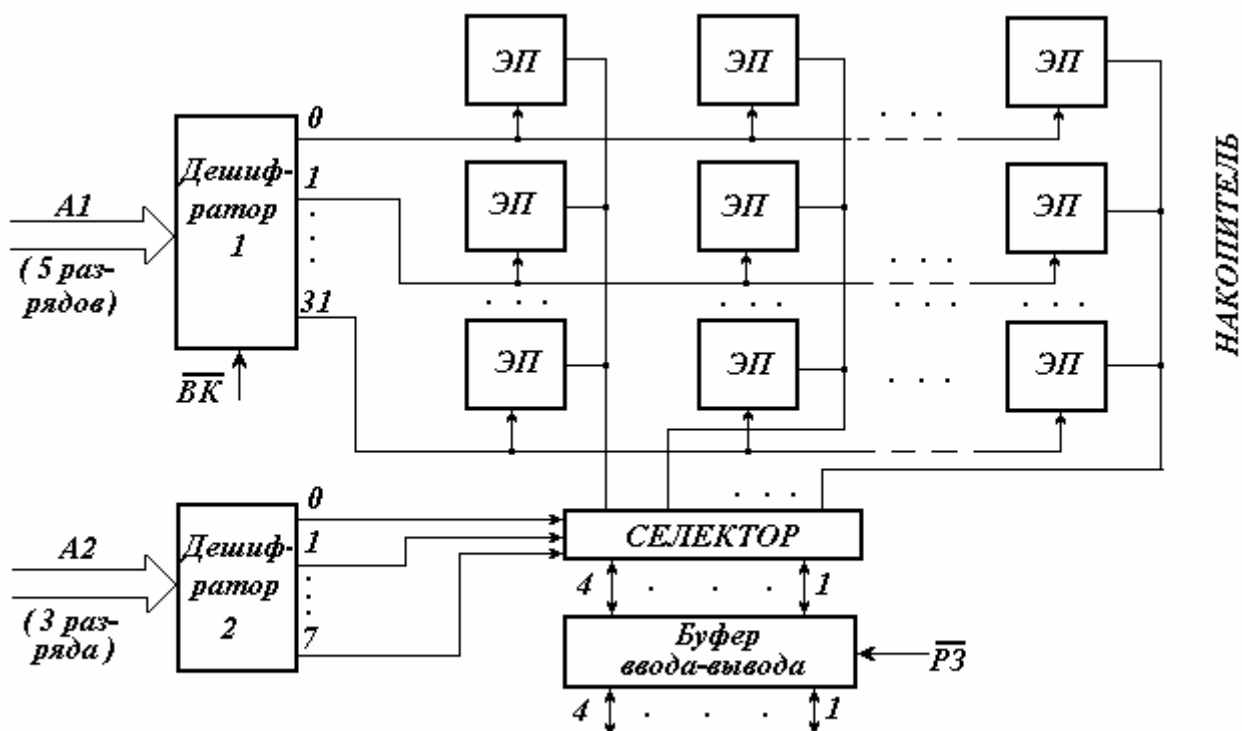


Рис. 6. 5

Структурная схема ПЗУ, программируемого пользователем, приведена на рис. 6.5. Матрица-накопитель состоит из элементов памяти, образующих строки и столбцы (как в ОЗУ), но в отличие от ОЗУ при считывании из накопителя выдается содержимое целой строки элементов памяти. Строка содержит несколько слов. Селектор позволяет выделить из строки и передать на выход требуемое слово.

В качестве примера предположим, что ПЗУ имеет емкость $M = 2^{10}$ бит, разбивающихся на $N = 2^8$ слов по $2^2 = 4$ разрядов в каждом слове. Накопитель будет содержать 2^{10} элементов памяти, расположенных вдоль $2^5 = 32$ строк и такого же количества столбцов.

При обращении указывается адрес слова, который в рассматриваемом примере содержит восемь разрядов, разделяющихся на две группы A_1 (пятиразрядная) и A_2 (трехразрядная). Группа A_1 подается на дешифратор 1, выбирающий одну из 32 строк накопителя. Содержимое строки составляет 32 бит, т. е. восемь 4-разрядных слов. Номер слова в строке задается группой A_2 . Дешифратор 2 преобразует эту адресную группу в сигнал на одном из восьми своих

выходов, который выделяет требуемое слово в селекторе, передаваемое через буфер ввода-вывода на выход микросхемы [3].

6. 4. Перепрограммируемые постоянные запоминающие устройства (ППЗУ)

ППЗУ могут хранить записанную в них информацию неопределенно долго при отключенном питании, а также допускают стирание записанной информации и запись новой. При этом необходимо учитывать, что, если чтение из памяти происходит за доли микросекунды, то запись потребует времени на несколько порядков больше.

На рис 6.6, а показан элемент памяти с электрической записью информации и стиранием ультрафиолетовым светом [3].

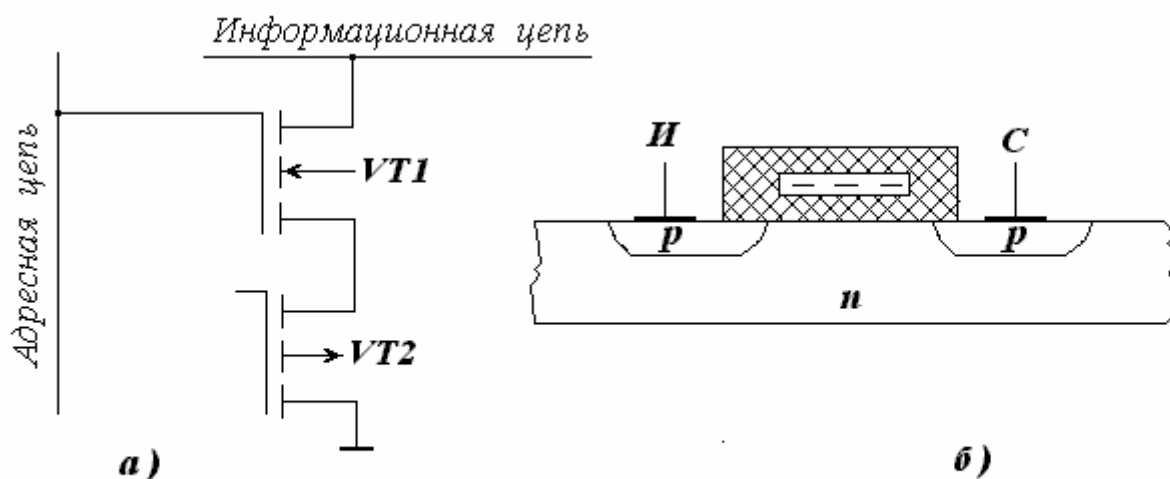


Рис. 6. 6

Транзистор $VT1$ производит выборку элемента памяти. Для хранения информации используется транзистор $VT2$, структура которого показана на рис. 6.6, б. Особенность структуры этого транзистора – изолированный затвор. При подаче достаточно большого напряжения к $p - n -$ переходу истока либо стока происходит *инжекция электронов в затвор*, после чего этот заряд может удерживаться на затворе длительное время.

Отрицательный заряд на затворе притягивает дырки и создает в $n -$ области проводящий $p -$ канал между истоком и стоком. Состояние транзистора соответствует *логическому нулю*.

Если к $p - n -$ переходу не прикладывать повышенного напряжения, заряда на затворе не будет, и транзистор окажется в непроводящем состоянии, что соответствует *логической единице*.

Процедура *стирания информации* зависит от типа микросхемы. Существуют микросхемы с *электрическим* и *ультрафиолетовым* стиранием информации. Под действием напряжения специальной формы или ультрафиолетового излучения, действующего в течение нескольких минут, снимается заряд с за-

творов транзисторов, и все транзисторы накопителя оказываются в непроводящем состоянии.

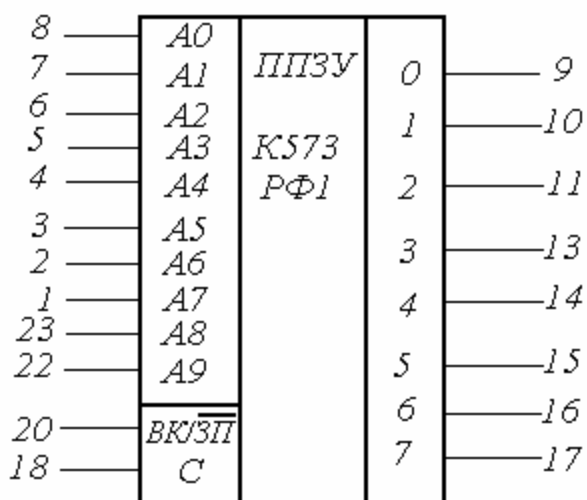


Рис. 6.7

На рис. 6.7 показана в качестве примера микросхема ППЗУ типа К573РФ1 с организацией 1024×8 , в которой предусмотрены 10 адресных входов (выводы с номерами 8, 7, 6, 5, 4, 3, 2, 1, 23, 22) и 8 входов/выходов данных (выводы 9, 10, 11, 13, 14, 15, 16, 17), совместимые с ТТЛ – логикой.

Режим чтения информации. Для работы микросхемы требуется три источника питания: +12 В (вывод 19), +5В (вывод 24), -5 В (вывод 21) относительно общего вывода 12. На входе $\overline{BK} / 3P$ (вывод 20) и программирующем входе (вывод 18) устанавливается напряжение уровня логического 0.

При стирании информации через прозрачную кварцевую крышку в корпусе микросхемы подается излучение с длиной волны около 400 мкм. Источник ультрафиолетового излучения лампы типа ДРТ – 220 или ДРТ – 375. При этом выводы микросхемы должны быть закорочены. В результате стирания во всех разрядах ячеек памяти устанавливается логическая 1.

Режим стирания информации. При стирании информации через прозрачную кварцевую крышку в корпусе микросхемы подается излучение с длиной волны около 400 мкм. Источник ультрафиолетового излучения лампы типа ДРТ – 220 или ДРТ – 375. При этом выводы микросхемы должны быть закорочены. В результате стирания во всех разрядах ячеек памяти устанавливается логическая 1.

Режим записи информации. В цикле записи на вход $\overline{BK} / 3P$ (вывод 20) подается напряжение +12 В. На адресных входах устанавливаются адреса ячеек памяти. Записываемая в ячейки памяти информация поступает в виде параллельных 8 – разрядных кодовых комбинаций. Для этой цели служат выводы 9 ... 11 и 13 ... 17. Эти выводы в режиме записи используются как входы записываемых данных.

Программирующий импульс амплитудой 26 В подается с некоторой задержкой относительно момента подачи кодовой комбинации адреса и записываемого числа на программирующий вход С (вывод 18).

Временная диаграмма импульсов в режиме записи представлена на рис. 6.8. Из диаграммы видно, что цикл записи предусматривает подачу по одному программирующему импульсу по каждому из адресов.

В течение каждого цикла записи *программируются все ячейки памяти.* Программирование отдельных ячеек или групп ячеек не допускается. Число циклов программирование памяти определяется выражением:

$$N = 100 \text{ мс} / t_{\text{и}}, \text{ где } t_{\text{и}} - \text{длительность программирующего импульса, мс.}$$

Например, если $t_{\text{и}} = 0,5 \text{ мс}$, то для надежной записи информации потребуется $N = 200$ циклов записи.

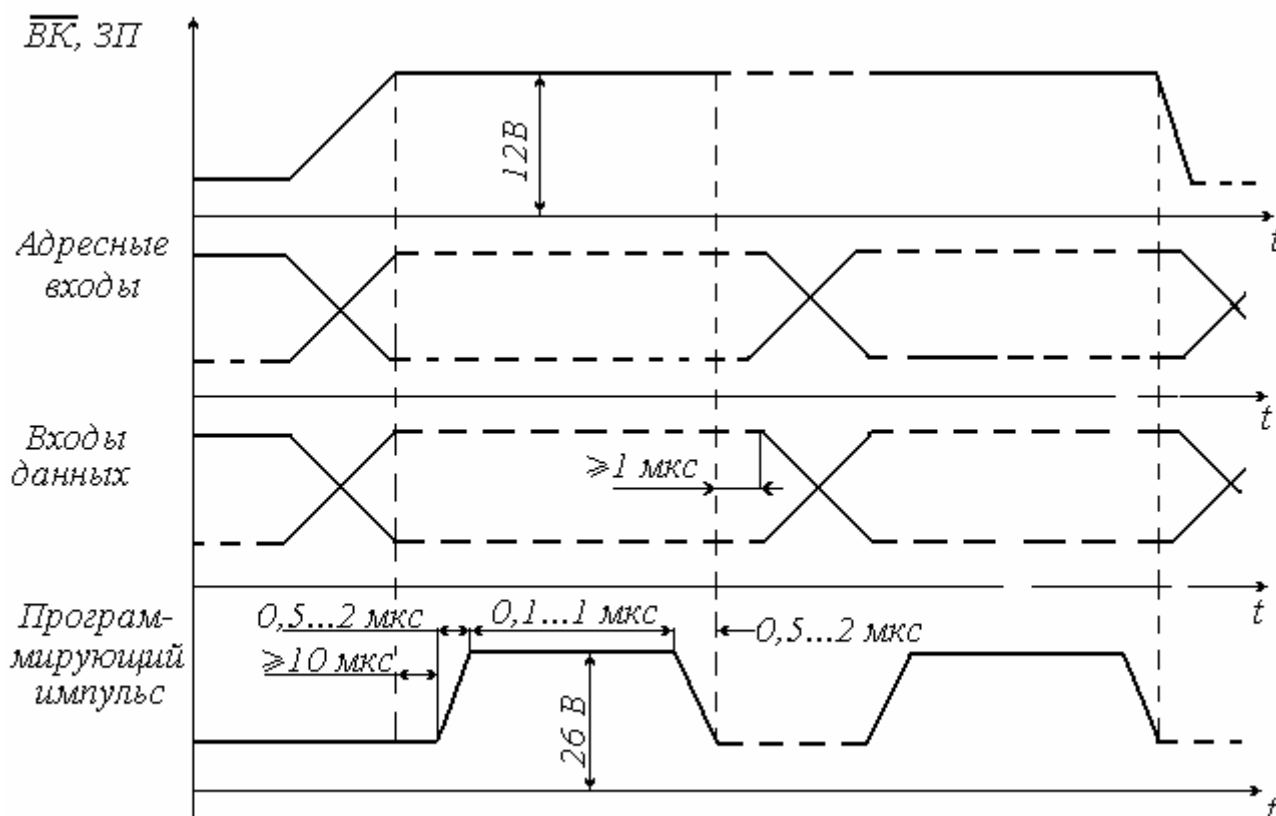


Рис. 6. 8

ППЗУ дороже ПЗУ и обычно находят применение в процессе отладки микропроцессорных устройств с целью уточнения информации, которая должна храниться в памяти. После отладки ППЗУ можно заменить на ПЗУ [3].

Вопросы и задания для самоконтроля

1. В каких случаях для хранения кодовых слов могут использоваться регистры?
2. Как классифицируются ЗУ по выполняемым функциям?
3. Что такое ЗУ с произвольной выборкой?
4. Что такое емкость и разрядность ЗУ?
5. Какими параметрами можно характеризовать быстродействие ЗУ?
6. Приведите схему наращивания разрядности ячеек ОЗУ.
7. Чем отличается ПЗУ от ОЗУ?
8. Как классифицируются ПЗУ по способу программирования?
9. Каковы особенности перепрограммируемых ПЗУ?
10. Какие существуют способы стирания информации в ППЗУ?
11. Как происходит запись информации с помощью программаторов?
12. Какова особенность ПЗУ, программируемого пользователем?

СПИСОК ЛИТЕРАТУРЫ

1. **Алексенко А. Г., Шагурин И. И.** Микросхемотехника: Учеб. пособие для вузов. – 2-е изд., переработ. и доп. – М.: Радио и связь, 1990. – 496 с.
2. **Антонов О. Г., Бабкин А. Ф., Голод О. С.** Цифровые устройства и микропроцессоры. Цифровые устройства: Конспект лекций. – СПб.: СЗПИ, 1995. – 82 с.
3. **Калабеков Б. А.** Цифровые устройства и микропроцессорные системы: Учебник для техникумов связи. – М.: Горячая линия–Телеком, 2002. – 336 с.
4. **Проектирование** импульсных и цифровых устройств радиотехнических систем: Учеб. пособие для радиотехнич. спец. вузов / Под ред. Ю. М. Казаринова. – М.: Высш. шк., 1985. – 319 с.
5. **Гольденберг Л. М.** Импульсные устройства: Учебник для вузов. – М.: Радио и связь, 1981. – 224 с.
6. **Ерофеев Ю. Н.** Импульсные устройства: Учеб. пособие для вузов по спец. «Радиотехника». – М.: Высш. шк., 1989. – 527 с.
7. **Трачик В.** Дискретные устройства автоматики: Пер. с польского / Под ред. Д. А. Поспелова. – М.: Энергия, 1978. – 456 с.
8. **Импульсные** устройства: Учеб. пособие / О. Г. Антонов, А. Ф. Бабкин, О. С. Голод и др. – Л.: СЗПИ, 1990. – 72 с.
9. **Фролкин В. Г., Попов Л. Н.** Импульсные устройства: Учебник для студентов вузов по спец. «Радиотехника». – М.: Сов. радио, 1980. – 367 с.
10. **Гольденберг Л. М. и др.** Цифровые устройства и микропроцессорные системы. Задачи и упражнения: Учеб. пособие для вузов. – М.: Радио и связь, 1992. – 256 с.
11. **Основы** импульсной и цифровой техники: Учеб. пособие для вузов / В. В. Гусев, Л. Г. Зеличенко, К. В. Конев и др. – М.: Сов. радио, 1975. – 440 с.
12. **Интегральные** микросхемы: Справочник / Под ред. Б. В. Тарабрина. – М.: Энергоатомиздат, 1985. – 528 с.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

- Адресный вход 65
Алгебра логики 5
Алгоритм представления числа 23
Аналитический способ задания функции 10
Аналого-цифровые преобразователи (АЦП) 61
Аргумент 7
Асинхронные триггеры 49
АЦП параллельного типа 62
АЦП с промежуточным преобразованием напряжения во временной интервал 63
- Байт 68
Бит 68
Бинарные функции 7
Булева алгебра 5
- Вес разряда 23
Восьмеричная система счисления 23
Время выборки 68
- Горнера схема 27
Графический метод минимизации
Карно-Вейча 14
- Двоичная система счисления 23
Двоично-взвешенная резистивная цепь 65
Двоичные счетчики с параллельным переносом 57
Двоичные счетчики с последовательным переносом 56
Декадный счетчик
Демультимплексор 36
Демультимплексор на программируемых логических матрицах (ПЛМ) 45
Десятичная система счисления..23
Дешифратор 34
Дешифратор линеек 71
Дешифратор на ПЛМ 44
- Дешифратор столбцов 70
Дешифратор строк 70
Дискретизация 61
- Емкость запоминающего устройства 68
- Задание логической функции числовым способом 16
Задний фронт 4
Закон ассоциативности 8
Закон дистрибутивности 8
Закон коммутативности 8
Закон поглощения 9
Закон склеивания 9
Законы и теоремы булевой алгебры 8
Запоминающие устройства (ЗУ) 68
Запрещенное сочетание входных сигналов 51
- Импликант 21
Импликант существенный 22
Инвертор 11
Информационный вход 35
Информационный уровень 5
Исключающее ИЛИ 31
Истинность 5
Истинности таблица 8
- Карта Карно функции 4-х переменных 15
Квантование 61
Кодопреобразователь 41
Комбинационная схема 10
Комбинационные устройства 31
Коммутатор 36
Конечный автомат (КА) 47
КА синхронный 48
КА максимальная частота переходов 48

- Логическая «1» 5
 Логическая функция 6
 Логические операции 7
 Логические схемы 5
 Логический базис 10
 Логический «0» 5
 Логическое умножение 8
 «Лишние» объединения 18
 Ложность 5
- Мажоритарный элемент типа «2 из 3-х» 10
 Максимальная группа 18
 Максимальная частота переходов конечного автомата (КА) 48
 Макстерм «,,,,,,», 10
- Мантисса 29
 Массив кодовых слов 68
 Матрица микросхем ОЗУ 72
 Метод минимизации Квайна-МакКласки 20
 Минимизация группированием нулей 19
 Минимизация логических схем алгебраическим методом 13
 Минимизация недоопределенных функций 20
 Минтерм 10
 Модуль счета 56
 Мультиплексор 35
 Мультиплексор и демультиплексор на ПЛМ 45
- Набор аргументов 6
 Накопитель 69
 Нарращивание разрядности 70
 Нарращивание числа ячеек 70
 Негативная логика 5
 Неполное склеивание 21
 Номер набора 18
- Обобщенная карта Карно 50
 Операция дизъюнкции 7
- Логическая схема 10
 Операция **И** 8
 Операция **ИЛИ** 7
 Операция инверсии (**НЕ**) 8
 Операция квантования 62
 Операция конъюнкции 8
 Операция **НЕ** 8
 Оперативное ЗУ (ОЗУ) 69
 Основание 29
 Основание системы счисления 23
 Основы цифровой техники 23
- Параллельные регистры 54
 Перевод чисел из одной системы счисления в другую 26
 Передний фронт 4
 Перенос 34
 Перепрограммируемые ПЗУ (ППЗУ) 74
 Позитивная логика 5
 Позиционная система счисления 23
 Полный сумматор 33
 Полусумматор 33
 Понятие о системах счисления 23
 Пороговый уровень 5
 Порядок 29
 Последовательностное устройство 47
 Последовательный регистр сдвига в одном направлении 55
 Постоянные запоминающие устройства (ПЗУ) 72
 ПЗУ, программируемые маской 72
 ПЗУ, программируемые пользователем 72
 Потенциальный тип сигнала 5
 Правило де Моргана 9
 Преобразователь «время-код» 63
 Преобразователь «код-напряжение» 65
 Прецизионный делитель 62
 Применение **BCD** – кода при кодировании десятичных чисел 24
 Программатор 73

- Программируемая логическая матрица 41
- Программируемые логические устройства с матричной структурой 41
- Программирующий импульс в ППЗУ 75
- Распределитель 36
- Режим записи новых слов 68
- Режим стирания информации ППЗУ 75
- Режим хранения 68
- Режим чтения 68
- Режим чтения информации ППЗУ 75
- Резистивная цепь $R - 2R$ 65
- Сетка уровней квантования 61
- Сигнал переноса 33
- Сигнал суммы 33
- Сингулярные функции 7
- Система логических элементов **И**, **ИЛИ**, **НЕ** 11
- Синхронные триггеры: 52
- RSC – триггер 52
- D – триггер 53; MS – триггер 54
- Синхронный конечный автомат 48
- Склеивающиеся слагаемые 14
- Словесное описание 9
- Совершенная дизъюнктивная нормальная форма (СДНФ) 11
- Совершенная конъюнктивная нормальная форма (СКНФ) 11
- Сокращенная дизъюнктивная нормальная форма (ДНФ) 21
- Способы задания логических функций 9
- Столбец неопределенности 49
- Стробирующий вход 35
- Структурная формула 10
- Счетный режим D -триггера 54
- Счетчики 56
- Счетчики с произвольным модулем счета 58
- Таблица истинности 7
- Табличный способ задания логической функции 10
- Тактовый интервал 61
- Тактовый момент 61
- Типовые приемы алгебраической минимизации 13
- Типы запоминающих устройств 68
- Триггеры: 48
- RS -триггер с инверсными входами 49
- E -триггер 50; JK -триггер 51
- Тупиковая форма записи 14
- Универсальные элементы 12
- Универсальный элемент «Стрелка Пирса» 12
- Универсальный элемент «Штрих Шеффера» 13
- Уравнения возбуждения RS -триггера 50
- Устройство «ЗАПРЕТ» 38
- Устройство неравнозначности 31
- Устройство равнозначности 32
- Устройство сравнения кодов 37
- Факультативные условия 20
- Формы представления числа 28
- Функциональная полнота 10
- Цикл записи 68
- Цифроаналоговые преобразователи (ЦАП) 64
- Цифровые автоматы 47
- Число разрядов адреса 70
- Шаг квантования 61
- Шестнадцатиричная система 24
- Шифратор 39
- Шифратор на ПЛИС 45
- Шум квантования 62

ОГЛАВЛЕНИЕ

	Стр.
ПРЕДИСЛОВИЕ	3
ВВЕДЕНИЕ	4
1. ЛОГИЧЕСКИЕ СХЕМЫ	5
1.1. Основные понятия алгебры логики	5
1.2. Логические операции.....	7
1.3. Законы и теоремы булевой алгебры	8
1.4. Способы задания логических функций	9
1.5. Переход от структурной формулы к логической схеме. Понятие функциональной полноты и логического базиса	10
1.6. Универсальные логические элементы	12
1.7. Минимизация логических схем алгебраическим методом	13
1.8. Графический метод минимизации Карно-Вейча	14
1.9. Метод минимизации Квайна-МакКласки	20
Вопросы и задания для самоконтроля	23
2. ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ	23
2.1. Понятие о системах счисления. Представление целых и дробных чисел	23
2.2. Перевод чисел из одной системы счисления в другую	26
2.3. Формы представления чисел с фиксированной и плавающей точкой	28
Вопросы и задания для самоконтроля	30
3. КОМБИНАЦИОННЫЕ УСТРОЙСТВА	31
3.1. Вводные понятия	31
3.2. Устройство неравнозначности	31
3.3. Устройство равнозначности	32
3.4. Полусумматор	33
3.5. Полный сумматор	33
3.6. Дешифраторы	34
3.7. Мультиплексоры и демультимплексоры	35
3.8. Устройство сравнения кодов (цифровой компаратор)	37
3.9. Устройство ЗАПРЕТ	38
3.10. Шифраторы и кодопреобразователи	39
3.11. Программируемые логические устройства с матричной структурой	41
Вопросы и задания для самоконтроля	46
4. ЦИФРОВЫЕ АВТОМАТЫ	47
4.1. Основные понятия	47
4.2. Триггеры	48
4.3. Синхронные триггеры	52
4.4. Регистры	54

4.5. Счетчики	56
Вопросы и задания для самоконтроля	60
5. АНАЛОГО – ЦИФРОВЫЕ И ЦИФРОАНАЛОГОВЫЕ ПРЕОБРАЗОВАТЕЛИ	61
5.1. Аналого – цифровые преобразователи (АЦП)	61
5.2. Цифроаналоговые преобразователи (ЦАП)	64
Вопросы и задания для самоконтроля	67
6. ПОЛУПРОВОДНИКОВЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА	68
6.1. Основные типы и параметры запоминающих устройств (ЗУ)	68
6.2. Оперативное запоминающее устройство (ОЗУ)	69
6.3. Постоянные запоминающие устройства (ПЗУ)	72
6.4. Перепрограммируемые постоянные запоминающие устройства (ППЗУ)	74
Вопросы и задания для самоконтроля	76
СПИСОК ЛИТЕРАТУРЫ	77
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	78

**Антонов Олег Гурьевич
Мигунова Екатерина Юрьевна**

**Цифровые устройства и микропроцессоры, ч. 1
Ц и ф р о в ы е у с т р о й с т в а**

Учебное пособие

Редактор И. Н. Садчикова
Сводный темплан 2006 г.
Лицензия ЛР № 020308 от 14.02.97

Санитарно – эпидемиологическое заключение № 78.01.07.953.П.005641. 11.03 от 21.11.2003

Подписано в печать		Формат 60×84 ^{1/16}
Б. кн. – журн.	П. л.	РТП РИО СЗТУ
	Тираж	Заказ

Северо–Западный государственный заочный технический университет
РИО СЗТУ, член Издательско-полиграфической
ассоциации университетов России
191186, Санкт-Петербург, ул. Миллионная, 5