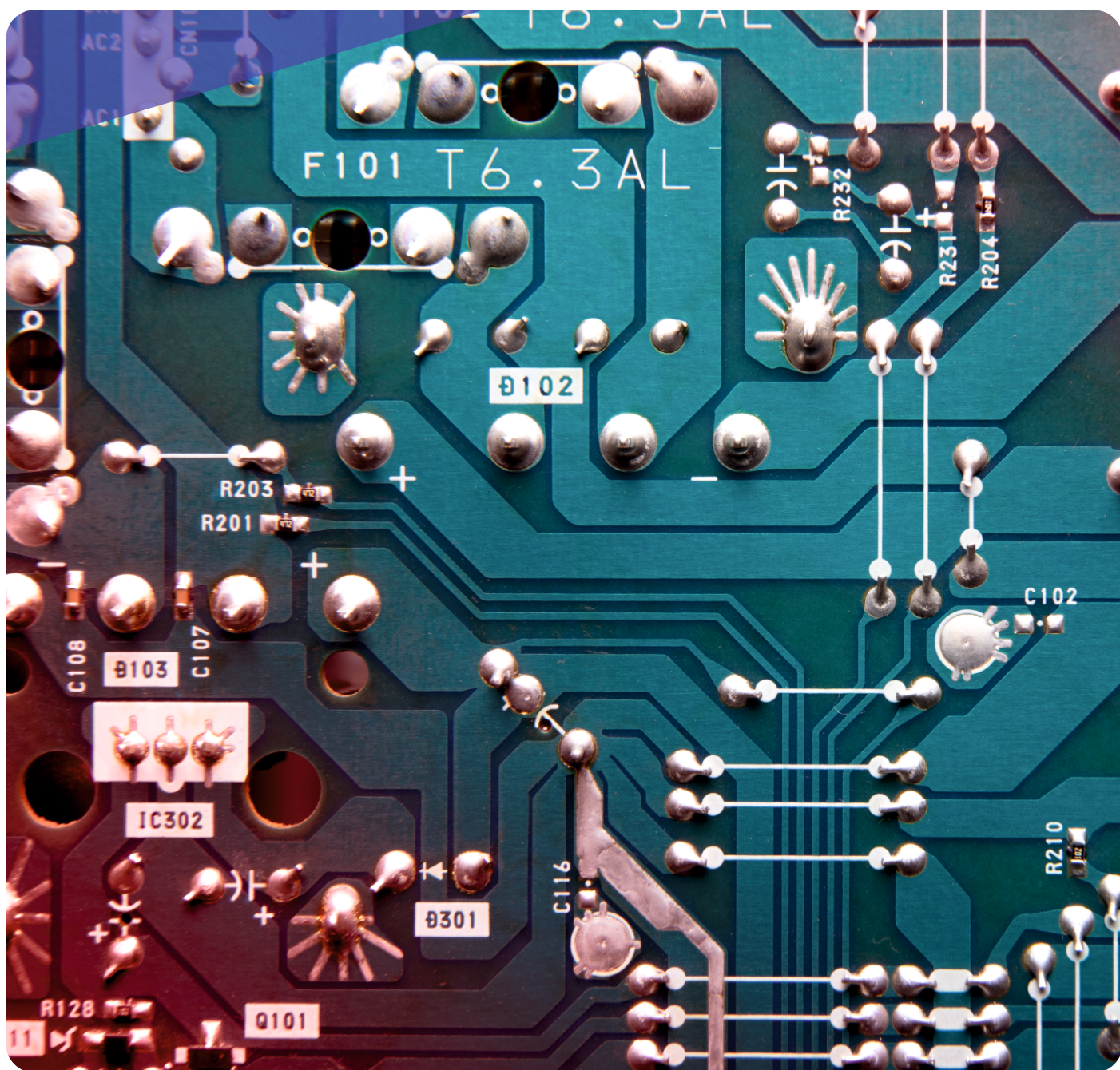


С.В. Федоров, А.Р. Фахруллина

ЦИФРОВЫЕ ЭЛЕКТРОННЫЕ УСТРОЙСТВА СИСТЕМ УПРАВЛЕНИЯ

Учебное пособие



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ**

С.В. Федоров, А.Р. Фахруллина

**ЦИФРОВЫЕ ЭЛЕКТРОННЫЕ УСТРОЙСТВА СИСТЕМ
УПРАВЛЕНИЯ**

Учебное пособие

**Уфа
РИЦ УУНиТ
2024**

УДК 621.38
ББК 32.85
Ф33

*Печатается по решению кафедры
«Технологии производства летательных аппаратов» УУНиТ.
Протокол № 7 от 19.03.2024 г.*

Рецензенты:

Л.А. Влацкая – канд. техн. наук, доцент кафедры «Электро- и теплоэнергетика» ФГБОУ ВО «Оренбургский государственный университет»;
Л.А. Герасимова – канд. техн. наук, доцент кафедры «Информационные технологии и системы управления» ФГБОУ ВО «Московский государственный университет технологий и управления имени К.Г. Разумовского»

Федоров С.В., Фахруллина А.Р.

Ф33 Цифровые электронные устройства систем управления: учебное пособие / С.В. Федоров, А.Р. Фахруллина. – Уфа: РИЦ УУНиТ, 2024. – 92 с.

ISBN 978-5-7477-5876-6

Целью учебного пособия является изучение основ современной теории и формирование знаний, умений и навыков для того, чтобы принимать участие в разработке и модернизации электронных устройств систем управления.

В учебном пособии приводятся основные понятия цифровой электроники. Представлены интегральные схемы логических элементов. Рассмотрены основы алгебры логики. Подробно описаны методы синтеза логических устройств в заданном базисе логических элементов. Приводятся примеры комбинационных логических схем и последовательностных цифровых устройств.

Предназначено для обучающихся по направлению 15.03.04 «Автоматизация технологических процессов и производств».

УДК 621.38
ББК 32.85

ISBN 978-5-7477-5876-6

© Федоров С.В.,
Фахруллина А.Р., 2024
© УУНиТ, 2024

Целью учебного пособия является изучение основ современной теории и формирование знаний, умений и навыков для того, чтобы принимать участие в разработке и модернизации электронных устройств систем управления. Для достижения поставленной цели были сформулированы задачи освоения материала данного учебного пособия:

1. Углубление знаний в области схмотехнических решений электронных устройств в системах управления.

2. Развитие умений и практических навыков построения электронных устройств.

3. Ознакомление с теорией проектирования узлов и элементов электронных устройств.

В соответствии с первой задачей в учебном пособии приводятся основные понятия цифровой электроники, основы алгебры логики и схемные реализации булевых функций. Практические навыки построения электронных устройств, в соответствии со второй задачей, приобретаются при получении знаний о синтезе логических устройств в заданном базисе логических элементов. Изучение разделов «комбинационные логические схемы» и «последовательностные цифровые устройства» позволяет получить теоретические знания о проектировании узлов и элементов электронных устройств систем управления, в соответствии с третьей задачей по освоению материала.

В результате изучения дисциплины обучаемые должны:

Знать: принцип работы и характеристики электронных приборов; основы цифровой электроники, общие принципы и алгоритмы синтеза и реализации цифровых устройств.

Уметь: проектировать электронные цифровые устройства.

Владеть: методами проектирования навыками проектирования электронных цифровых устройств.

Учебное пособие предназначено для обучающихся по направлению 15.03.04 «Автоматизация технологических процессов и производств». Может быть полезным для других специальностей, связанных с проектированием и разработкой систем управления.

1 Цифровые электронные устройства систем управления

1.1 Основные понятия цифровой электроники

Цифровые логические элементы на интегральных микросхемах (ИМС) – это микроэлектронные изделия, предназначенные для преобразования и обработки дискретных сигналов.

Интегральные схемы можно классифицировать по ряду независимых параметров.

По типу сигналов различают элементы с различными способами электрического кодирования двоичной информации:

- потенциальные;
- импульсные;
- импульсно-потенциальные.

При потенциальном способе кодирования при положительной логике за единицу принимают высокий потенциал, за ноль – низкий потенциал. Сигнал сохраняется неизменным на время не менее одного периода следования сигналов синхронизации (рис. 1.1, *а*).

При импульсном кодировании двоичной чаще всего единице соответствует импульс, синфазный с сигналом синхронизации, а ноль – отсутствие импульса; значение сигнала в паузе между сигналами синхронизации не рассматривается (рис. 1.1, *б*).

Одной из разновидностей импульсного способа является динамическое кодирование сигналов, когда единице соответствует последовательность импульсов между двумя импульсами синхронизации, а их отсутствие соответствует нулю (рис. 1.1, *в*).

При импульсно-потенциальном способе кодирования двоичные сигналы в одних элементах (чаще запоминающих элементах) кодируются потенциалами, а в других (логических) – импульсами.

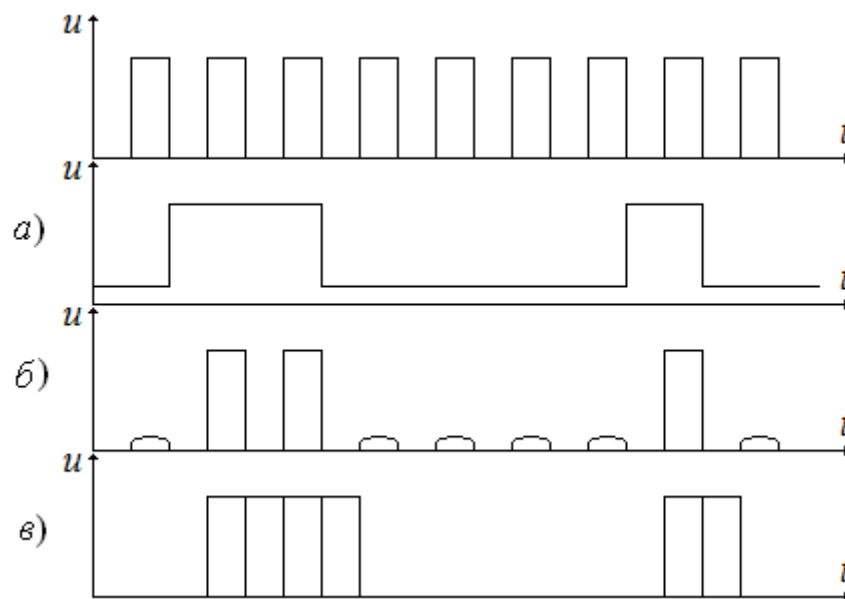


Рис. 1.1. Способы представления двоичной информации: а) при потенциальном, б) при импульсом и в) при импульсно-потенциальном способах кодирования

По конструктивно-технологическому исполнению:

1) монолитные (полупроводниковые) интегральные схемы, элементы которых выполнены в объеме и (или) на поверхности полупроводниковой подложки, выполняющие функции каких-либо электронных элементов;

2) гибридные интегральные схемы предполагают формирование на единой диэлектрической подложке навесных миниатюрных бескорпусных ИС, что часто используется для научных исследований;

3) пленочные (толсто- и тонкопленочные). Эти виды схем очень схожи между собой, но отличаются в изготовлении. Тонкопленочные устройства имеют лучшие рабочие характеристики, но дороже толстопленочных. Такие интегральные схемы используют, когда с помощью $p-n$ -перехода нельзя получить необходимые параметры резисторов, конденсаторов.

По типу используемых активных элементов полупроводниковые интегральные схемы подразделяются на:

1) униполярные: полевые с управляющим $p-n$ -переходом, МОП-транзисторы с n - и p -каналами и КМОП-транзисторы;

2) биполярные, которые в зависимости от используемого типа логических структур и конструктивно-технологических решений делятся на:

- резистивно-транзисторную логику (РТЛ);

- диодно-транзисторную логику (ДТЛ);
- транзисторно-транзисторную логику (ТТЛ и ТТЛШ);
- интегрально-инжекционную логику (И²Л);
- эмитерно-связанную логику (ЭСЛ).

Следующими независимыми признаками классификации является плотность упаковки и степень интеграции.

Плотность упаковки – количество элементов (N) в 1 см³ полупроводника. При N равном до 100 элементов интегральные схемы получили название – ИС малой степени интеграции. При N до 1000 элементов называется интегральной схемой средней степени интеграции (СИС). При N более 10000 элементов интегральные схемы получили название больших интегральных схем (БИС). Плотность упаковки характеризует функциональную сложность интегральной схемы, и такой параметр называют степенью интеграции, он определяется через десятичный логарифм от числа элементов ИС:

$$Q = \log(N).$$

По функциональному назначению интегральные схемы делятся на четыре класса:

1) цифровые, которые предназначены для обработки сигналов, заданных в виде дискретных функций:

- логические интегральные схемы;
- запоминающие устройства (ЗУ);
- триггеры;
- устройства для обработки цифровой информации.

2) аналоговые интегральные схемы предназначены для обработки сигналов, заданных в виде непрерывной функции:

- генераторы;
- усилители;
- фильтры частот;
- устройства задержки сигналов;
- формирователи и т. д.

3) микропроцессоры – класс интегральных схем для вычислительных устройств.

4) источники питания.

По применяемости в аппаратуре интегральные схемы делятся на:

1) схемы общего применения;

2) схемы специального применения.

1.2 Интегральные схемы логических элементов

Интегральная схема представляет собой конструктивно законченное изделие электронной техники, содержащее совокупность электрически связанных в функциональную схему транзисторов, диодов, конденсаторов, резисторов и других электронных элементов, выполненных на поверхности или в объеме полупроводника в едином технологическом процессе.

Интегральные схемы являются элементной базой электроники для преобразования, обработки и хранения информации. Постоянное повышение требований к увеличению быстродействия и уменьшению мощности потребления вычислительных средств привело к созданию различных интегральных схем, разработка которых, как правило, проводится сериями.

Серия ИС объединяет интегральные схемы по функциональному назначению и конструктивно-технологическим признакам. Все интегральные схемы одной серии согласованы по источнику напряжения питания, по входному и выходному сопротивлениям, технологии изготовления, монтажу. В серии объединяются интегральные схемы, из которых можно создать любой сложности законченное электронное устройство.

Принята система условного обозначения интегральных схем, например, КР 155ИД7, цифры и буквы расшифровываются следующим образом:

- буквы КР характеризуют условия приемки интегральной схемы заводом-изготовителем;
- буква К обозначает микросхему общепромышленного (широкого) применения;
- перед буквой К может стоять цифра (если эта цифра 1, то она может быть опущена): 1 (5, 6, 7) – полупроводниковые ИС, 2 (4, 8) – гибридные;
- цифры 155 определяют номер серии;
- буквы ИД определяют функциональное назначение интегральной схемы, буквы ИД являются обозначением дешифратора

(И – ЛИ, НЕ – ЛН, ИЛИ – ЛЛ, И-НЕ – ЛА, ИЛИ-НЕ – ЛЕ, И-ИЛИ-НЕ – ЛР);

- цифра после вторых букв показывает номер разработки.

1.3 Основы алгебры логики

Логическая функция $f(x_1, x_2, \dots, x_n)$ – это функция, принимающая значения 0 и 1, аргументы которой x_1, x_2, \dots, x_n также принимают значения 0 и 1. Здесь 0 и 1 – не арифметические величины, а истинностные значения:

0 – «нет» – ЛОЖЬ;

1 – «да» – ИСТИНА.

Вторым названием логических функций является название булевы функции, которые названы так в честь английского математика Джорджа Буля, который впервые в 1849 году описал использование подобных функций.

Первоначально логические функции использовались для описания схем на основе переключательных (двухстабильных) элементов, которые назывались *переключательные схемы* (switching circuits). Отдавая дань традиции, современные цифровые схемы также называются переключательными, и для их описания используются *переключательные функции*. В дальнейшем изложении термины «булевы функции», «переключательные функции» и «логические функции» используются в тексте как синонимы.

Рассмотрим область определения и область значений булевой функции. Аргументы булевой функции n переменных можно рассматривать как выборку из n элементов (выборку размерности n), каждый из которых принимает два значения $\{0, 1\}$. *Область определения* такой булевой функции (всевозможные наборы аргументов) можно рассматривать как множество перестановок с повторениями в выборке размерности n из двух элементов $\{0, 1\}$. Таким образом, количество входных наборов m булевой функции n переменных вычисляется по формуле

$$m = 2^n.$$

1.3.1 Способы представления логических функций

Всякая логическая функция может быть задана одним из нижеперечисленных способов.

Словесный – при этом способе словесное описание однозначно определяет все случаи, при которых функция принимает значения 0 или 1. Например, многовходовая функция ИЛИ может иметь такое словесное описание: функция принимает значение 1, если хотя бы один из аргументов принимает значение 1, иначе – 0.

Числовой – функция задается в виде десятичных (или восьмеричных, или шестнадцатеричных) эквивалентов номеров тех наборов аргументов, на которых функция принимает значение 1. Условие, что функция $f(x_1, x_2, x_3) = 1$ на наборах 1, 3, 5, 6, 7 записывается $f(1, 3, 5, 6, 7) = 1$. Аналогичным образом булева функция может быть задана по нулевым значениям. При нумерации наборов переменным x_1, x_2, x_3 ставится в соответствие веса $2^2, 2^1, 2^0$, т. е. 6 набору соответствует двоичный эквивалент 110, а 1 набору – 001.

Табличный – функция задается в виде таблицы истинности (соответствия), которая содержит 2^n строк (по числу наборов аргументов), n столбцов по числу переменных и один столбец значений функции. В такой таблице каждому набору аргументов соответствует значение функции. Например, при $n = 3$, число строк $2^3 = 8$, число возможных функций трех переменных $2^{2^3} = 2^8 = 256$.

Аналитический – функция задается в виде алгебраического выражения, получаемого путем применения каких-либо логических операций к переменным алгебры логики. Применяя операции конъюнкции и дизъюнкции, можно задать функцию выражением $f(x_1, x_2, x_3) = x_1x_2 + x_3$.

Координатный – при этом способе задания таблица истинности функции представляется в виде координатной карты состояний, которая часто называется *картой Карно*. Такая карта содержит 2^n клеток по числу наборов всевозможных значений n переменных функции. Переменные функции разбиваются на две группы так, что одна группа определяет координаты столбца, а другая – координаты строки. При таком способе построения клетка определяется координатами переменных, соответствующих определенному двоичному набору. Внутри клетки карты Карно ставится значение

функции на данном наборе. Переменные в строках и столбцах располагаются так, чтобы соседние клетки карты Карно различались только в одном разряде переменных, т. е. были соседними. Такой способ представления очень удобен для наглядности при минимизации булевых функций.

		$x_2 x_3$			
		00	01	11	10
x_1	0	0	1	1	0
	1	0	1	1	1

1.3.2 Законы булевой алгебры. Основные аксиомы, теоремы и тождества

Как любая алгебраическая система *булева алгебра* базируется на совокупности некоторых предположений, которые принято называть *аксиомами*, т. е. предположениями не требующими доказательств. Аксиомы определяются для двух логических значений 1 («ИСТИНА») и 0 («ЛОЖЬ») и операций логического умножения (конъюнкции), которая обозначается «&», « \wedge », « \cdot » или не обозначается вовсе; логического сложения (дизъюнкции), которая обозначается « \vee », «+», и отрицания (инверсии), которая обозначается горизонтальной чертой («-») над переменной или выражением, например, \bar{x} . Булевой переменной, обозначаемой обычно x_i , называется переменная принимающая два логических значения $\{0, 1\}$.

Ниже приведены аксиомы булевой алгебры относительно дизъюнкции, конъюнкции и отрицания:

- 1) аксиомы *конъюнкции*: $0 \cdot 0 = 0$; $1 \cdot 1 = 1$; $0 \cdot 1 = 1 \cdot 0 = 0$;
- 2) аксиомы *дизъюнкции*: $0 + 0 = 0$; $1 + 1 = 1$; $0 + 1 = 1 + 0 = 1$;
- 3) аксиомы *отрицания*: Если $x=0$, то $\bar{x}=1$; Если $x=1$, то $\bar{x}=0$.

Следующие 5 правил обычно называют *теоремами* булевой алгебры. Особенностью теорем булевой алгебры является то, что для их доказательства пользуются простой подстановкой значений булевых переменных. Это обусловлено тем, что переменные могут принимать только 2 значения – 0 и 1.

- 4) *операции с константами*:
 - $x \wedge 1 = x$;

- $x \wedge 0 = 0$;
- $x \vee 1 = 1$;
- $x \vee 0 = x$;
- $\bar{0} = 1$;
- $\bar{1} = 0$.

5) *идемпотентность* (тавтология, повторение):

- $x \wedge x = x$;
- $x \wedge x \wedge \dots \wedge x = x$;
- $x \vee x = x$;
- $x \vee x \vee \dots \vee x = x$.

6) *противоречие*:

$$x \wedge \bar{x} = 0$$

7) *правило «исключенного третьего»*:

$$x \vee \bar{x} = 1$$

8) *двойное отрицание* (инволюция):

$$\bar{\bar{x}} = x$$

Следующие 4 правила обычно называют *законами* или тождествами булевой алгебры.

9) *ассоциативность* (ассоциативный закон):

- $x_1(x_2x_3) = (x_1x_2)x_3$;
- $(x_1 + x_2) + x_3 = x_1 + (x_2 + x_3)$.

10) *коммутативность* (коммутативный закон):

- $x_1x_2 = x_2x_1$;
- $x_1 + x_2 = x_2 + x_1$.

11) *дистрибутивность* (дистрибутивный закон):

- конъюнкции относительно дизъюнкции — $x_1(x_2 + x_3) = x_1x_2 + x_1x_3$;
- дизъюнкции относительно конъюнкции — $x_1 + (x_2x_3) = (x_1 + x_2)(x_1 + x_3)$.

12) *законы де-Моргана* (законы инверсии или отрицания):

- $\overline{x_1x_2} = \bar{x}_1 + \bar{x}_2$;
- $\overline{x_1 + x_2} = \bar{x}_1\bar{x}_2$.

Расширенный закон де-Моргана:

- $\overline{x_1x_2 \dots x_n} = \bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_n$;
- $\overline{x_1 + x_2 + \dots + x_n} = \bar{x}_1\bar{x}_2 \dots \bar{x}_n$.

Следующие 3 правила доказываются на основе законов дистрибутивности, противоречия и «исключенного третьего».

13) *поглощение* (элиминация):

- $x_1 + x_1x_2 = x_1$;
- $x_1(x_1 + x_2) = x_1$.

14) *закон Блейка-Порецкого*:

- $x_1 + \bar{x}_1x_2 = x_1 + x_2$;
- $x_1(\bar{x}_1 + x_2) = x_1x_2$.

15) *склеивание* (объединение):

- $(x_1 + x_2)(x_1 + \bar{x}_2) = x_1$;
- $x_1x_2 + x_1\bar{x}_2 = x_1$.

Приведенные аксиомы, теоремы и тождества булевой алгебры позволяют осуществлять любые преобразования булевых функций, получая в результате переключательные схемы нужной структуры и свойств.

1.3.3 Аналитическое представление булевых функций

Дизъюнктивная и конъюнктивная нормальные формы. В данном подразделе более подробно рассматривается аналитическое представление булевых функций в виде уравнений (*булевых уравнений*) с использованием операций дизъюнкции (ИЛИ), которую принято обозначать « \vee », конъюнкции (И), которую принято обозначать « $\&$ », « \cdot » или не обозначать вовсе, и отрицания (инверсии), которую обозначают горизонтальной чертой (« $\bar{}$ ») над выражением, например, \bar{x} . Данные операции образуют *булев базис*.

Рассмотрим основные понятия и определения, используемые при аналитическом представлении булевых функций.

Элементарное произведение – произведение (конъюнкция) любого числа букв (переменных) булевой функции, взятых с отрицанием или без. Например, $x_1x_2x_3$.

Элементарная сумма – логическая сумма (дизъюнкция) любого числа букв (переменных) булевой функции, взятых с отрицанием или без. Например, $x_1 + x_2 + x_3$.

Дизъюнктивная нормальная форма (ДНФ) – дизъюнкция элементарных произведений. Термин «нормальная» означает, что в данном выражении отсутствуют групповые инверсии, т. е. инверсия над несколькими переменными сразу. Пример ДНФ $f = x_1x_2x_3 + \bar{x}_1x_3$.

Минтерм (конституэнта 1) – произведение всех переменных, взятых с отрицанием или без, соответствующих двоичным наборам, на которых функция принимает значение 1. Минтерм можно назвать полной элементарной конъюнкцией.

Совершенной ДНФ (СДНФ) называется ДНФ, содержащая все полные элементарные конъюнкции (конституэнты 1) данной булевой функции, в которой нет одинаковых элементарных конъюнкций, и каждая из них содержит все переменные данной булевой функции, причем каждую переменную – только один раз (включая вхождения с отрицанием или без отрицания). Другими словами СДНФ – это дизъюнкция всех минтермов булевой функции.

Конъюнктивная нормальная форма (КНФ) – конъюнкция элементарных сумм. Термин «нормальная» означает, что в данном выражении отсутствуют групповые инверсии, т.е. инверсия над несколькими переменными сразу. Пример КНФ $f = (x_1 + x_2 + x_3)(\bar{x}_1 + x_3)$.

Макстерм (конституэнта 0) – сумма всех переменных, взятых с отрицанием или без, соответствующих двоичным наборам, на которых функция принимает значение 0. Макстерм можно назвать полной элементарной дизъюнкцией.

Совершенной КНФ (СКНФ) называется КНФ, содержащая все полные элементарные дизъюнкции (конституэнты 0) данной булевой функции, в которой нет одинаковых элементарных дизъюнкций, и каждая из них содержит все переменные данной булевой функции, причем каждую переменную – только один раз (включая вхождения с отрицанием или без отрицания). Другими словами СКНФ – это конъюнкция всех макстермов булевой функции.

В связи с тем, что одной и той же булевой функции могут соответствовать различные формы аналитической записи, возникает задача нахождения такой формы записи, при которой каждой функции будет соответствовать одна и только одна формула стандартного типа, и каждой формуле стандартного типа будет соответствовать одна и только одна функция. Такие формы записи булевых функций называются *каноническими*. СДНФ и СКНФ являются каноническими формами представления булевых функций.

Переход от табличной формы задания булевых функций к аналитическим. Особый интерес представляет переход от табличных формы представления булевых функций к аналитическим.

Для получения СДНФ и СКНФ, исходя из *таблицы истинности*, можно сформулировать следующие правила.

Для получения СДНФ на основе таблицы истинности необходимо:

1) каждый из входных наборов, на которых булева функция принимает значения 1, представить в виде элементарного произведения (конъюнкции), причем если переменная равна 0, то она входит в конъюнкцию с инверсией, а если 1 – то без инверсии;

2) полученные элементарные конъюнкции объединяются знаками дизъюнкции.

3) Для получения СКНФ на основе таблицы истинности необходимо:

1) каждый из входных наборов, на которых булева функция принимает значения 0, представить в виде элементарной логической суммы (дизъюнкции), причем если переменная равна 1, то она входит в дизъюнкцию с инверсией, а если 0 – то без инверсии;

2) полученные элементарные дизъюнкции объединяются знаками конъюнкции.

3) В качестве примера рассмотрим булеву функцию трех переменных, $f(1, 3, 5, 6, 7)=1$. Ниже приведены таблица истинности и полученные на ее основе СДНФ и СКНФ.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

СДНФ: $f = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3$

СКНФ: $f = (x_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + x_2 + x_3)$

Инверсные функции. Так как в булевой алгебре функции принимают только два значения 0 и 1, то существует особый класс функций, являющихся *инверсными* по отношению к рассматриваемым функциям, т. е. на тех наборах, где данная функция принимает значение 0 (1), инверсная функция принимает значение 1 (0) соответственно.

На основании закона де-Моргана можно сформулировать правило получения инверсной функции.

Для получения аналитического выражения инверсной функции необходимо в исходной функции все переменные заменить на инверсные им, все знаки дизъюнкции заменить на знаки конъюнкции и наоборот.

Например, для ДНФ $f = x_1x_2 + x_3$, $\bar{f} = (\bar{x}_1 + \bar{x}_2)\bar{x}_3$. Полученную таким образом инверсную функцию называют *обратной КНФ*.

Для КНФ $f = (x_1 + x_3)(x_2 + x_3)$, $\bar{f} = \bar{x}_1\bar{x}_3 + \bar{x}_2\bar{x}_3$. Полученную таким образом инверсную функцию называют *обратной ДНФ*.

1.3.4 Минимизация логических функций

Под *минимизацией* будем понимать процесс нахождения такого эквивалентного выражения логической функции, которое содержит минимальное число вхождений переменных. Хотя в общем случае под минимизацией может иметься в виду получение выражений с минимальным числом инверсных переменных либо с минимальным числом вхождений какой-либо одной переменной и т. п. Большинство методов минимизации ориентированы на получение минимальных ДНФ (минимальных КНФ), однако доказано, что минимальное выражение в классе ДНФ будет также минимальным, либо будет отличаться от минимального на одно вхождение переменной в классе других форм функции.

Правило склеивания. Упрощение выражений булевых функций (минимизация) основывается на понятии несущественности переменных. Переменная называется *несущественной* на паре наборов, если при изменении ее значения на противоположное булева функция сохраняет свое значение.

Например, для булевой функции трех переменных, $f(1, 3, 5, 6, 7)=1$, 6-я и 7-я конъюнкции имеют вид: $x_1x_2\bar{x}_3$, $x_1x_2x_3$. По дистрибутивному закону: $x_1x_2\bar{x}_3 + x_1x_2x_3 = x_1x_2(\bar{x}_3 + x_3) = x_1x_2$.

Таким образом, две конъюнкции, содержащие несущественную переменную, заменяются одной, в которой несущественная переменная отсутствует.

Приведем основные определения, используемые при минимизации булевых функций. Данные определения используют понятия нормальных (канонических) форм булевых функций.

Число переменных, входящих в элементарную конъюнкцию (для ДНФ) или в элементарную дизъюнкцию (для КНФ), называется ее *рангом*.

В основе любых методов минимизации лежит операция *склеивания*. Два элементарных произведения одного ранга (для ДНФ) или элементарных сумм одного ранга (для КНФ) склеиваются, если они различаются только по одной переменной.

Операция $Ax + A\bar{x} = A$ называется *полным склеиванием*, а операция $Ax + A\bar{x} = A + Ax + A\bar{x}$ – *неполным склеиванием* (для ДНФ).

Операция $(A+x)(A+\bar{x}) = A$ называется *полным склеиванием*, а операция $(A+x)(A+\bar{x}) = A(A+x)(A+\bar{x})$ – *неполным склеиванием* (для КНФ).

Импликантой называется элементарное произведение, равное 1 на одном или нескольких наборах, где данная функция равна 1, и равное 0 на всех наборах, где данная функция равна 0. Импликанта покрывает один или несколько минтермов рассматриваемой булевой функции. Обычно импликанта – это результат склеивания соответствующих минтермов или импликант.

Простая импликанта – это импликанта, которая содержит хотя бы минтерм функции, но перестает быть импликантой после удаления любого аргумента (иными словами, это импликанта, к которой не нельзя применить операцию склеивания).

Сокращенная ДНФ – это дизъюнкция всех простых импликант.

Существенная импликанта – это простая импликанта, образованная склеиванием таких минтермов, чтобы по крайней мере для одного из них эта операция была единственной. Существенные импликанты образуют ядро функции.

Тупиковая ДНФ – это дизъюнкция простых импликант, из которых ни одна не является лишней.

МДНФ (минимальная ДНФ) – тупиковая ДНФ с минимальным числом вхождений переменных (минимальным числом букв) по сравнению с другими тупиковыми формами этой функции.

Имплицентой называется элементарная логическая сумма, равная 0 на одном или нескольких наборах, где данная функция равна 0, и равная 1 на всех наборах, где данная функция равна 1.

Имплицента покрывает один или несколько макстермов рассматриваемой булевой функции. Обычно имплицента – это результат склеивания соответствующих макстермов.

Простая имплицента – это имплицента, которая содержит хотя бы макстерм функции, но перестает быть имплицентой после удаления любого аргумента (иными словами, это имплицента, к которой не нельзя применить операцию склеивания).

Сокращенная КНФ – это конъюнкция всех простых имплицент.

Существенная имплицента – это простая имплицента, образованная склеиванием таких макстермов, что, по крайней мере, для одного из них эта операция была единственной. Существенные имплиценты образуют ядро функции.

Тупиковая КНФ – это конъюнкция простых имплицент, из которых ни одна не является лишней.

МКНФ (минимальная КНФ) – тупиковая КНФ с минимальным числом вхождений переменных (минимальным числом букв) по сравнению с другими тупиковыми формами этой функции.

Минимизация булевых функций с использованием карт Карно. Рассмотрим визуальный метод минимизации булевых функций с помощью карт (диаграмм) Карно, который является одним из наиболее удобных методов при небольшом числе переменных (до 6). Данный метод был разработан в 1953 г. американским ученым Морисом Карно.

Карта Карно является *координатным* способом представления булевых функций. При этом способе задания таблица истинности функции представляется в виде координатной карты состояний, которая содержит 2^n клеток (по числу входных наборов булевой функции n переменных). Переменные функции разбиваются на две группы так, что одна группа определяет координаты столбца карты, а другая – координаты строки. При таком способе построения каждая клетка определяется значениями переменных, соответствующих определенному двоичному набору. Внутри каждой клетки карты Карно ставится значение функции на данном наборе. Переменные в строках и столбцах располагаются так, чтобы соседние клетки карты Карно *различались только в одном разряде переменных*, т. е. были соседними. Поэтому значения переменных в столбцах и в строках карты образуют соседний код Грея. Такой способ представления очень удобен для наглядности при минимизации булевых функций.

Карта Карно 4 переменных, с точки зрения определения «соседства» переменных, геометрически представляет собой пространственную фигуру тор или, проще говоря, «бублик».

Отметим, что метод карт Карно применим к минимизации булевых функций до 6 переменных (до 4 переменных на плоскости) и до 6 – в трехмерной интерпретации.

Правила минимизации с использованием карт Карно:

1) в карте Карно группы единиц (для получения ДНФ) и группы нулей (для получения КНФ) необходимо обвести четырехугольными контурами. Внутри контура должны находиться только одноименные значения функции. Этот процесс соответствует операции *склеивания* или нахождения импликант данной функции;

2) количество клеток внутри контура должно быть целой степенью двойки (1, 2, 4, 8, 16...);

3) при проведении контуров крайние строки карты (верхние и нижние, левые и правые), а также угловые клетки, считаются соседними (для карт до 4 переменных);

4) каждый контур должен включать максимально возможное количество клеток. В этом случае он будет соответствовать простой импликанте;

5) все единицы (нули) в карте (даже одиночные) должны быть охвачены контурами. Любая единица (нуль) может входить в контуры произвольное количество раз;

6) множество контуров, покрывающих все 1 (0) функции, образуют тупиковую ДНФ (КНФ). *Целью минимизации* является нахождение минимальной из множества тупиковых форм;

7) в элементарной конъюнкции (дизъюнкции), которая соответствует одному контуру, остаются только те переменные, значение которых не изменяется внутри обведенного контура. Переменные булевой функции входят в элементарную конъюнкцию (для значений функции 1) без инверсии, если их значение на соответствующих координатах равно 1, и с инверсией – если 0. Для значений булевой функции, равных 0, записываются элементарные дизъюнкции, куда переменные входят без инверсии, если их значение на соответствующих координатах равно 0, и с инверсией – если 1.

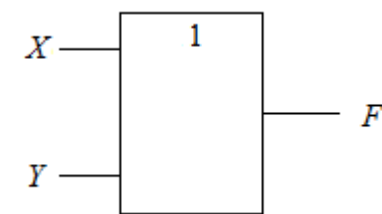
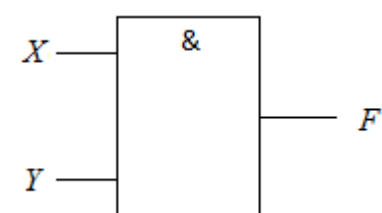
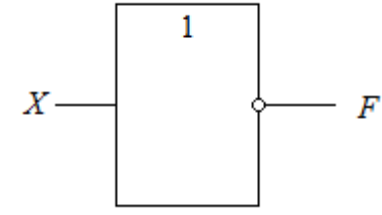
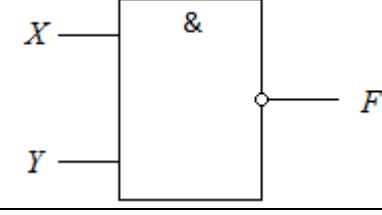
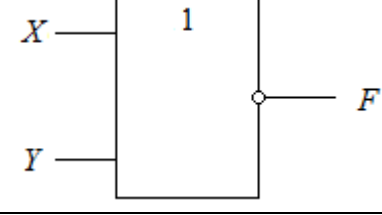
1.4 Схемные реализации булевых функций

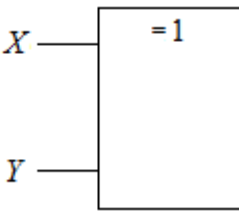
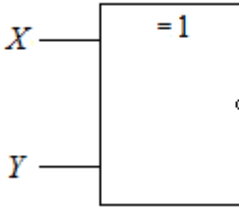
Логические элементы интегральной схемы – это электронные схемы, выполняющие простейшие логические операции. Логические элементы используют в цифровых интегральных схемах в качестве основных элементов и определяют параметры микросхемы.

Логические элементы, выполняющие простейшие логические операции приведены в табл. 1.1.

Таблица 1.1

Логические элементы

Логическая функция	Условное обозначение	Булево выражение	Таблица истинности		
			Входы		Выход
			X	Y	F
1	2	3	4	5	6
ИЛИ		$X + Y = F$	0	0	0
			0	1	1
			1	0	1
			1	1	1
И		$XY = F$	0	0	0
			0	1	0
			1	0	0
			1	1	1
НЕ		$\bar{X} = F$	0	-	1
			1	-	0
И-НЕ		$\overline{XY} = F$	0	0	1
			0	1	1
			1	0	1
			1	1	0
ИЛИ-НЕ		$\overline{X + Y} = F$	0	0	1
			0	1	0
			1	0	0
			1	1	0

1	2	3	4	5	6
Исключающее ИЛИ		$F \quad X \oplus Y = F$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Исключающее ИЛИ–НЕ		$F \quad \overline{X \oplus Y} = F$	0	0	1
			0	1	0
			1	0	0
			1	1	1

1.5 Синтез логических устройств в заданном базисе логических элементов

Реализация любого цифрового устройства предусматривает прежде всего необходимость выбора элементной базы этого устройства в зависимости от требований заказчика к его быстродействию, помехоустойчивости, энергопотреблению, надежности, нагрузочной способности и т. п. После выбора элементной базы и серии ИС схемотехническая реализация устройства идет по определенной схеме с логическими элементами, изображенными по ЕСКД.

На примере покажем последовательность действий при реализации цифрового устройства, работающего по определенному алгоритму:

1) словесный алгоритм работы цифрового устройства и его таблица истинности.

Цифровое устройство должно сформировать на выходе единичный сигнал (функция истинна), если количество истинных аргументов в наборе операндов будет нечетным. Этот словесный алгоритм записывается в виде таблицы истинности работы данного устройства (табл. 1.2).

Таблица истинности

№	Набор входных операндов				Значение функции устройства
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	
0	0	0	0	0	*
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	*
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	*

Задание функции знаком * показывает, что функция при таком наборе аргументов неопределенна и поэтому может быть либо ложной, либо истинной, и принимается такое значение функции, которое позволит минимизировать характеристическое уравнение, описывающее работу этого цифрового устройства.

2) составление характеристического уравнения и его минимизация.

Функция F в СДНФ записывается как сумма минтермов для истинных значений функции:

$$F_{\text{СДНФ}} = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D} + ABCD$$

В выражении $F_{\text{СДНФ}}$ добавлены минтермы неопределенного значения функции ($\overline{A}\overline{B}\overline{C}\overline{D}$, $\overline{A}\overline{B}\overline{C}D$, $ABC\overline{D}$), принятые за истинные. Возможно, это поможет лучше минимизировать функцию, в противном случае этими минтермами можно пренебречь.

Минимизация функции осуществляется с помощью карт Карно. При четырех аргументах карта Карно будет иметь размерность 4×4 .

Таблица 1.3

Карта Карно для функции F

	CD			
AB	00	01	11	10
00	*	1	0	1
01	1	*	0	0
11	0	0	*	1
10	1	0	1	1

Склеиваем минтермы, соответствующие истинным значениям функции в соответствии с изложенным правилом. Получаем

Таблица 1.4

Карта Карно для функции F

	CD			
AB	00	01	11	10
00	*	1	0	1
01	1	*	0	0
11	0	0	*	1
10	1	0	1	1

Получаем 3 контура (контур № 1: 0000, 0001, 0100, 0101; контур № 2: 1111, 1110, 1011, 1010; контур № 3: 0000, 0010, 1000, 1010). Тогда минимизированная функция примет вид:

$$F = \bar{A}\bar{C} + AC + \bar{B}\bar{D}.$$

3) реализация функции.

При реализации функции F на базовых элементах типа И-НЕ необходимо представить функцию F в форме удобной для реализации на выбранной элементной базе. Используя закон инверсии, функцию можно записать в виде:

$$F = \overline{\overline{\overline{B\bar{D}}\bar{A}\bar{C}}\bar{A}C}.$$

В соответствии с этим уравнением реализуем функцию на элементах И-НЕ (рис. 1.2).

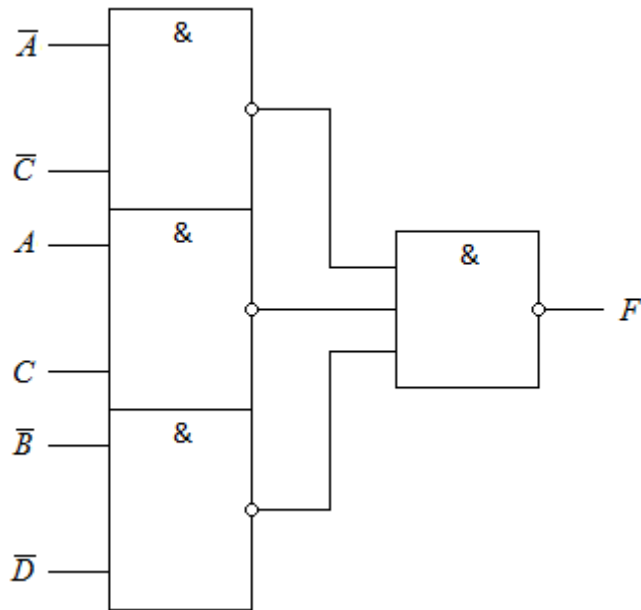


Рис. 1.2. Реализация функции на элементах И-НЕ

Характеристическое уравнение цифрового устройства в форме, удобной для реализации на элементной базе И-ИЛИ-НЕ, приведено ниже.

$$F = \overline{\overline{BD} + \overline{AC} + AC}.$$

В соответствии с этим уравнением реализуем функцию на элементах И-ИЛИ-НЕ (рис. 1.3).

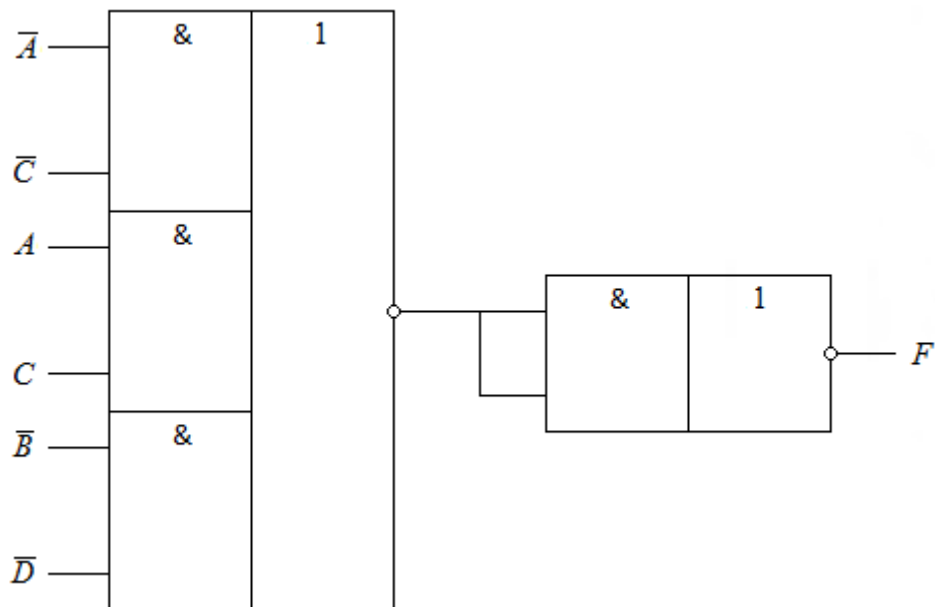


Рис. 1.3. Схема функции на элементах И-ИЛИ-НЕ

1.6 Комбинационные логические схемы

1.6.1 Шифраторы

Шифратор – это комбинационная схема, преобразующая входной управляющий сигнал в кодовую комбинацию на выходе.

Если в шифраторе имеется N входов и n выходов, то связь между количеством входов и выходов устанавливается соотношением

$$N = 2^n .$$

Схематическое изображение шифратора, имеющего 8 входов и 3 выхода, приведено на рис. 1.4.

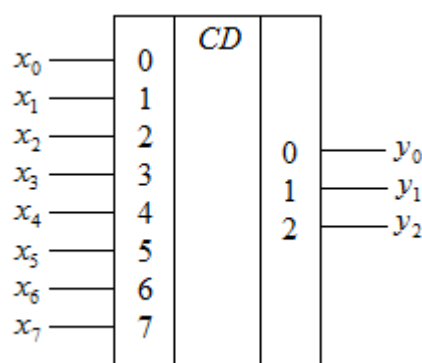


Рис. 1.4. Схематическое изображение шифратора

В его условном обозначении буквы CD – сокращение от слова *coder* (шифратор). Примером использования шифратора является устройство ввода двоичных кодов в цифровое устройство (например компьютер) с клавиатуры. При нажатии клавиши на одном из входов шифратора появляется логическая 1 и на выходе устанавливается двоичный код, соответствующий нанесенному на клавише знаку (букве, цифре и т. д.).

Рассмотрим синтез шифратора на примере преобразования восьмиразрядного унитарного кода в трехразрядный двоичный код. Таблица истинности такого шифратора приведена в табл. 1.5.

Таблица истинности шифратора

Вход								Выход		
x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y_2	y_1	y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Запишем выражение для выходных сигналов в СДНФ:

$$y_0 = \bar{x}_0 x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 x_5 \bar{x}_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 x_7,$$

$$y_1 = \bar{x}_0 \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 x_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 x_7,$$

$$y_2 = \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \bar{x}_5 \bar{x}_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 x_5 \bar{x}_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 x_6 \bar{x}_7 + \bar{x}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 x_7.$$

Анализ таблицы показывает, что для шифраторов должно выполняться условие $x_i x_j = 0$ при $i \neq j$ и что в данном случае для получения аналитической записи логических функций в минимизированном виде не требуется применение карт Карно (которые были бы достаточно сложными для восьми аргументов).

Запишем выражение для младшего разряда y_0 . Он принимает значение 1 только тогда, когда хотя бы на один из входов с нечетным номером (x_1, x_3, x_5, x_7) подается 1. Отсюда следует, что

$$y_0 = x_1 + x_3 + x_5 + x_7.$$

Средний разряд y_1 принимает значение 1 только тогда, когда хотя бы на один из входов с x_2, x_3, x_6, x_7 подается 1. Отсюда следует, что

$$y_1 = x_2 + x_3 + x_6 + x_7.$$

Старший разряд y_2 принимает значение 1 только тогда, когда хотя бы на один из входов с x_4, x_5, x_6, x_7 подается 1. Получаем, что

$$y_2 = x_4 + x_5 + x_6 + x_7.$$

На основании полученных уравнений можно составить схему шифратора. Она будет состоять из трех дизъюнкторов с четырьмя входами (рис. 1.5).

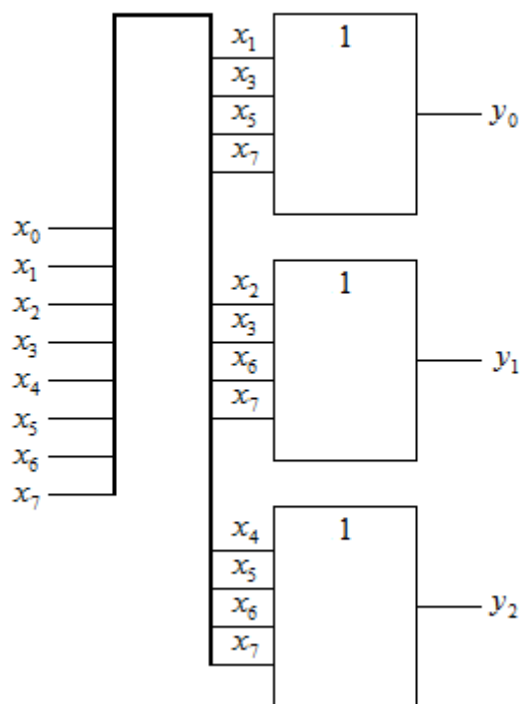


Рис. 1.5. Схема шифратора

1.6.2 Дешифраторы

Дешифратор – это комбинационная схема, преобразующая входной двоичный код в код унитарный на выходе.

Дешифратор часто называют адресным устройством, так как по адресу в двоичном коде на входе, на выходе, номер которого определяется этим адресом, появляется сигнал 1, а на всех других – 0.

Если в шифраторе имеется n входов и N выходов, то связь между количеством входов и выходов устанавливается соотношением

$$N = 2^n.$$

Схематическое изображение дешифратора, имеющего 3 входа и 8 выходов, приведено на рис. 1.6.

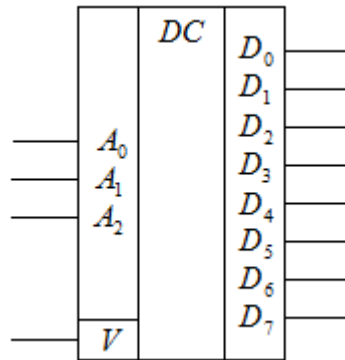


Рис. 1.6. Схематическое изображение дешифратора

Обозначение дешифратора DC – сокращение от *decoder* – дешифратор.

Вход V (от *veto* – запрет) дешифратора позволяет не только запрещать работу устройства, но и наращивать его разрядность. Когда на этот вход подается логическая единица, независимо от состояния информационных входов на выходе образуются нули.

Когда на вход V дешифратора подается 0, на выходе, номер которого соответствует двоичному коду адреса, обозначенного на входе, образуется логическая единица, на других выходах формируется логический нуль. Таблица истинности дешифратора для 3 входов и 8 выходов имеет вид:

Таблица 1.6

Таблица истинности дешифратора

Вход				Выход							
V	A_2	A_1	A_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0	0	1
1	*	*	*	0	0	0	0	0	0	0	0

Запишем выражение для выходных сигналов в СДНФ:

$$D_0 = \bar{V} \bar{A}_2 \bar{A}_1 \bar{A}_0;$$

$$D_1 = \bar{V} \bar{A}_2 \bar{A}_1 A_0;$$

$$D_2 = \bar{V} \bar{A}_2 A_1 \bar{A}_0 ;$$

$$D_3 = \bar{V} \bar{A}_2 A_1 A_0 ;$$

$$D_4 = \bar{V} A_2 \bar{A}_1 \bar{A}_0 ;$$

$$D_5 = \bar{V} A_2 \bar{A}_1 A_0 ;$$

$$D_6 = \bar{V} A_2 A_1 \bar{A}_0 ;$$

$$D_7 = \bar{V} A_2 A_1 A_0 .$$

Из полученных выражений видно, что для синтеза дешифратора необходимо восемь конъюнкторов и четыре инвертора. Схема дешифратора приведена на рис. 1.7.

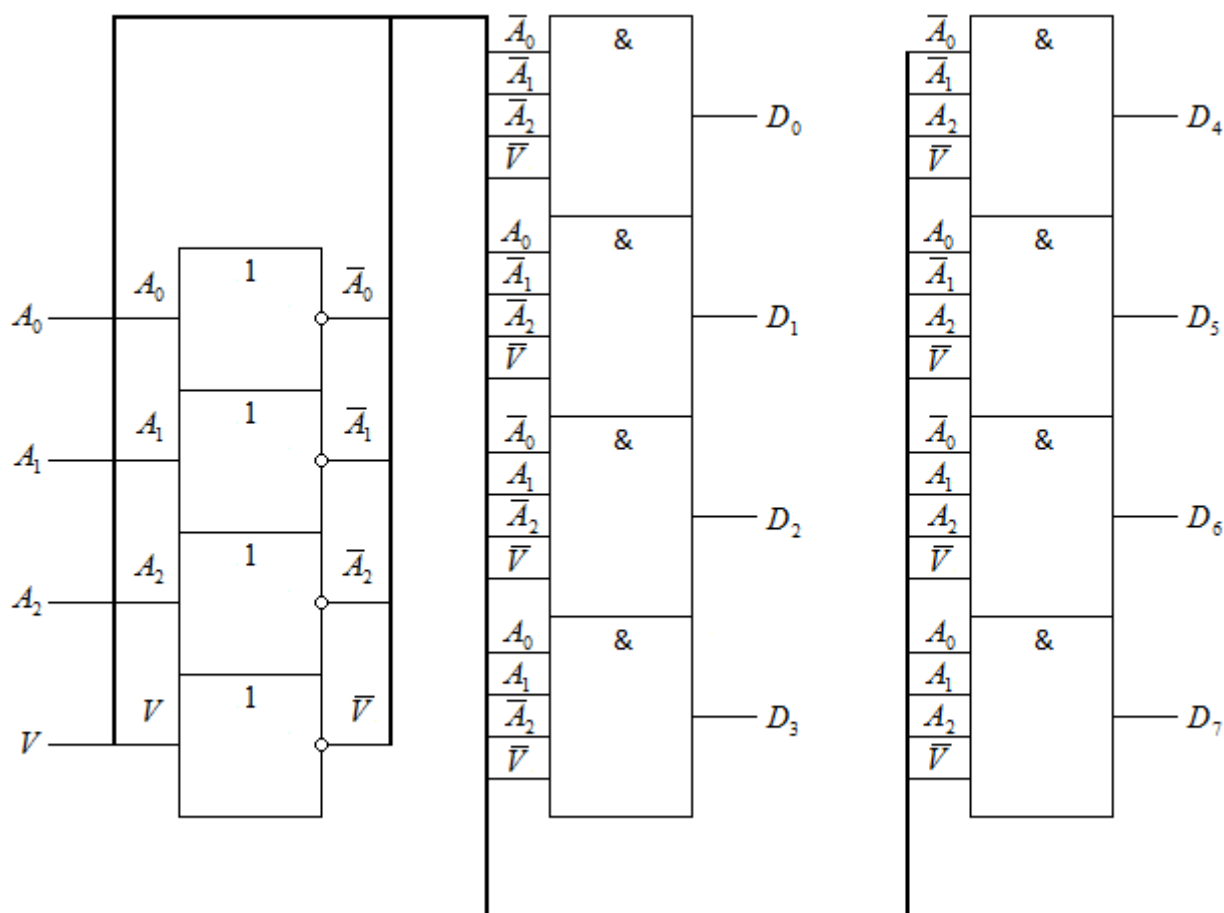


Рис. 1.7. Схема дешифратора

1.6.3 Мультиплексоры

Мультиплексорами (*multiplexer* – коммутатор, *multi* – много, *plex* – сеть) называют устройства, передающие сигнал с одного из информационных входов, указанного в адресе, на общий выход, т. е. выполняющий роль селектора каналов.

Мультиплексоры нашли широкое применение как универсальные переключатели и устройства, реализующие произвольные логические функции.

Схематическое изображение мультиплексора, приведено на рис. 1.8.

На этой схеме входы $D_0, D_1 \dots D_7$ – это информационные входы, A_0, A_1, A_2 – адресные. Число информационных входов N равно

$$N = 2^n,$$

где n – число адресных входов.

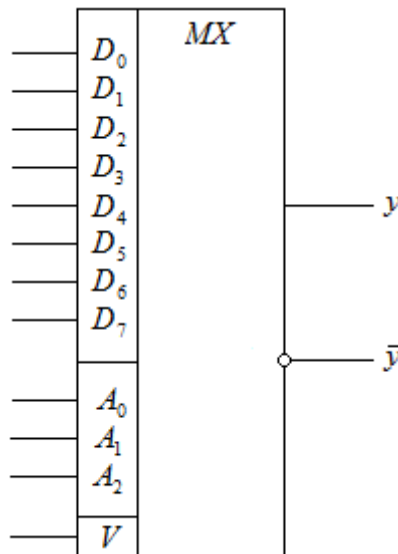


Рис. 1.8. Схематическое изображение мультиплексора

Таблица истинности мультиплексора, изображенного на рис. 1.8, приведена ниже.

Таблица 1.7

Таблица истинности мультиплексора

Вход				Выход
V	A_2	A_1	A_0	y
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7
1	*	*	*	0

Запишем выражение для выходного сигнала в СДНФ с вынесенным за скобки сигналом запрета V :

$$y = \bar{V}(D_0\bar{A}_2\bar{A}_1\bar{A}_0 + D_1\bar{A}_2\bar{A}_1A_0 + D_2\bar{A}_2A_1\bar{A}_0 + D_3\bar{A}_2A_1A_0 + D_4A_2\bar{A}_1\bar{A}_0 + D_5A_2\bar{A}_1A_0 + D_6A_2A_1\bar{A}_0 + D_7A_2A_1A_0)$$

Схема мультимплексора, реализующего данную функцию, приведена на рис. 1.9.

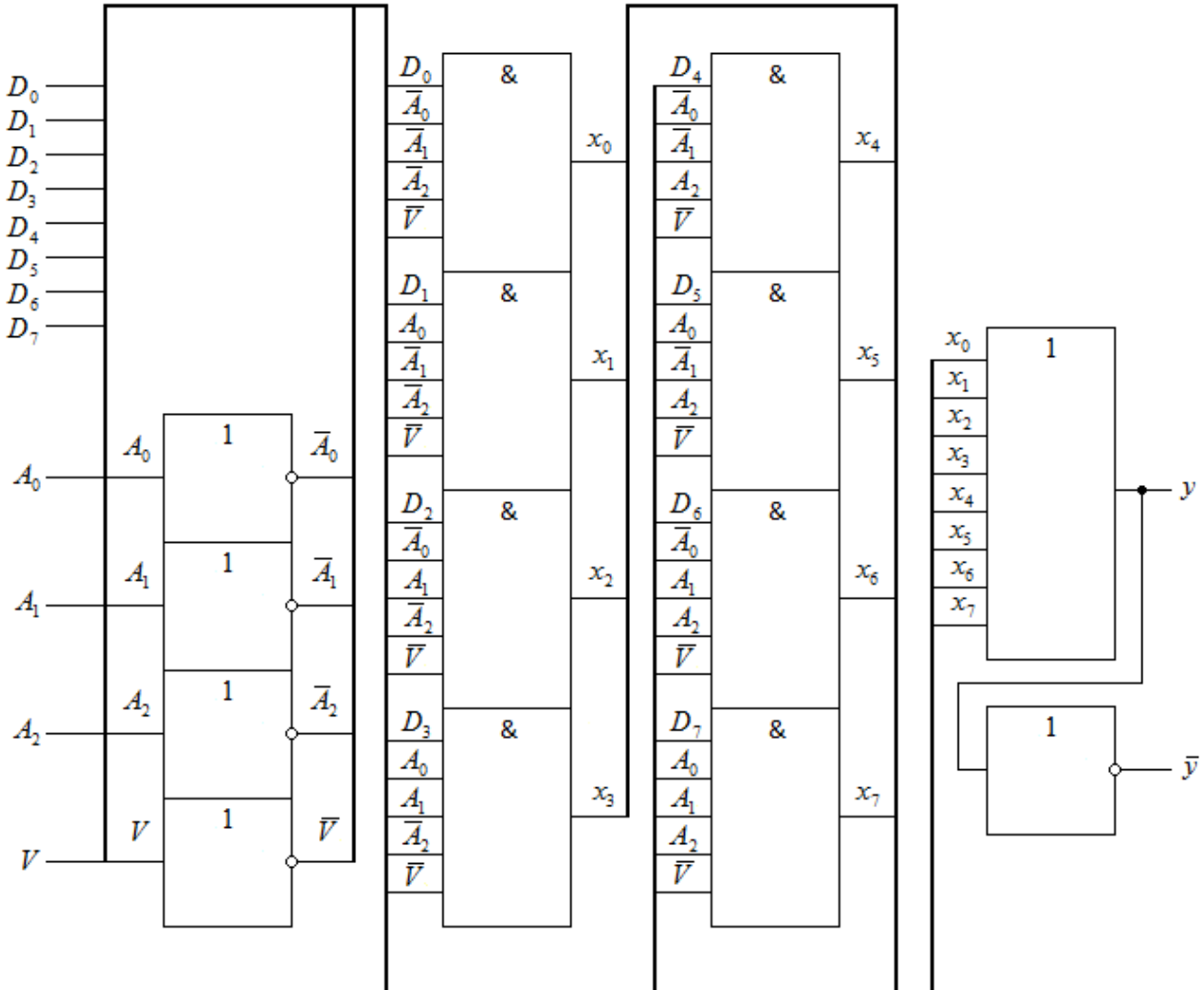


Рис. 1.9. Схема реализации мультимплексора

Мультимплексоры выпускаются промышленностью на число информационных входов не более 16. Большое число коммутируемых сигналов достигается путем наращивания мультимплексоров. Одним из способов является устройство многоразрядного мультимплексора с добавлением одного мультимплексора, устанавливаемого на общем выходе (рис. 1.10).

В этой схеме выходы мультимплексоров $DD1...DD4$ подключены к информационным входам объединяющего мультимплексора $DD5$, адресными входами которого служат старшие разряды аргументов функции. Достоинством данного устройства является то, что оно

состоит из одноименных интегральных микросхем мультиплексоров.

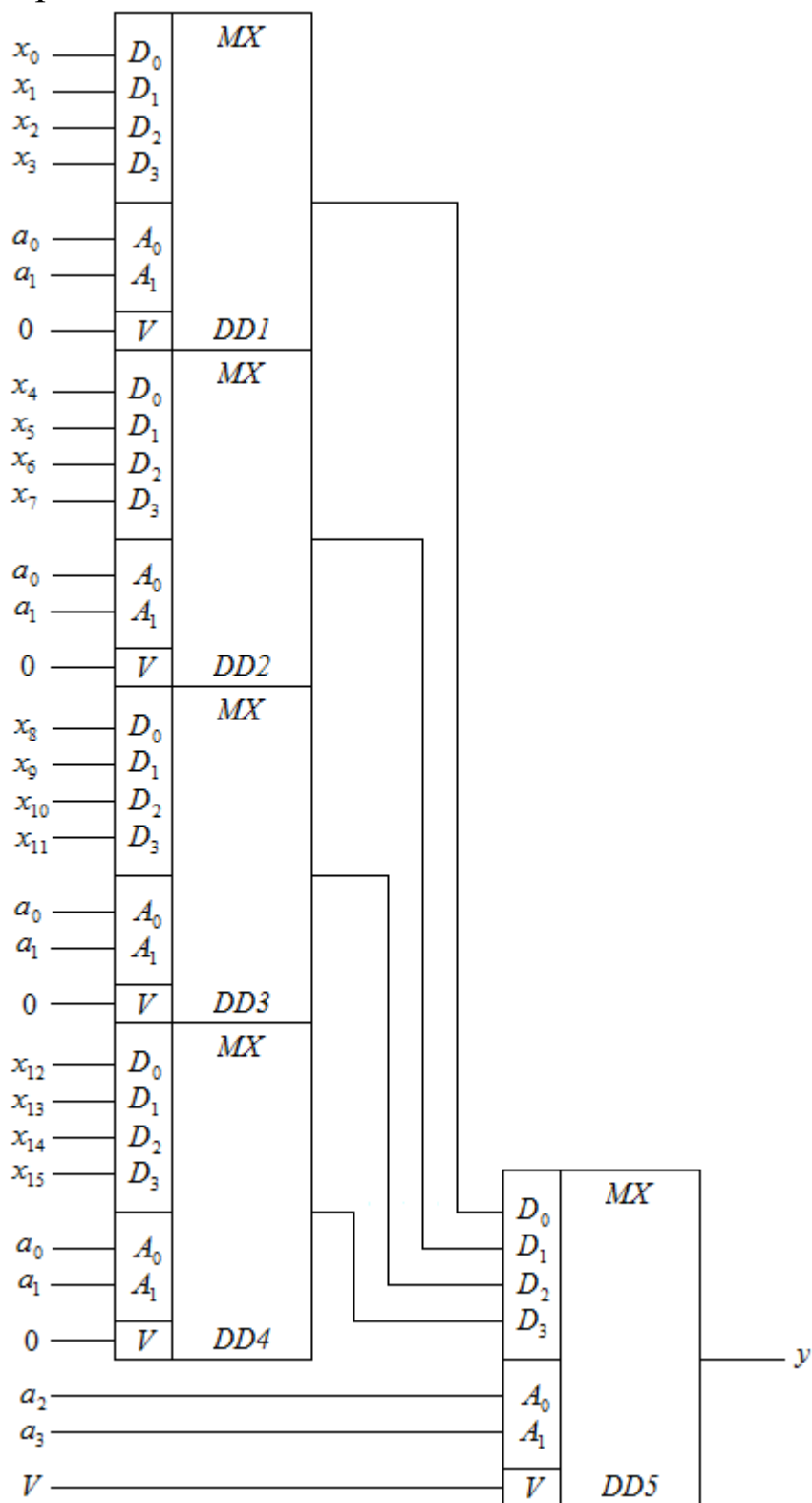


Рис. 1.10. Схема многоразрядного мультиплексора с добавлением одного мультиплексора на выходе

1.6.4 Демультимплексоры

Демультимплексоры выполняют функцию, обратную мультиплексорам: переключают сигнал с единственного входа на один из выходов – тот, который указан в адресе.

Схематическое изображение демультимплексора, приведено на рис. 1.11.

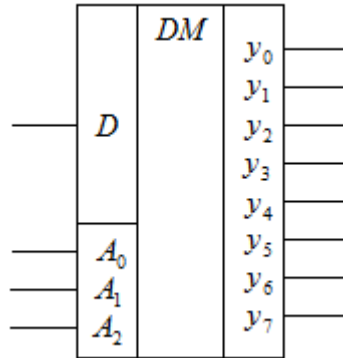


Рис. 1.11. Схематическое изображение демультимплексора

На этой схеме вход D – это информационный вход, A_0 , A_1 , A_2 – адресные. Число информационных выходов N равно $N=2^n$, где n – число адресных входов.

Таблица истинности демультимплексора с тремя адресными входами и восемью информационными выходами представлена ниже (табл. 1.8).

Таблица 1.8

Таблица истинности демультимплексора

Вход			Выход							
A_2	A_1	A_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	D	0	0	0	0	0	0	0
0	0	1	0	D	0	0	0	0	0	0
0	1	0	0	0	D	0	0	0	0	0
0	1	1	0	0	0	D	0	0	0	0
1	0	0	0	0	0	0	D	0	0	0
1	0	1	0	0	0	0	0	D	0	0
1	1	0	0	0	0	0	0	0	D	0
1	1	1	0	0	0	0	0	0	0	D

Запишем выражение для выходных сигналов в СДНФ:

$$y_0 = D\bar{A}_2\bar{A}_1\bar{A}_0;$$

$$y_1 = D\bar{A}_2\bar{A}_1A_0;$$

$$y_2 = D\bar{A}_2A_1\bar{A}_0;$$

$$y_3 = D\bar{A}_2A_1A_0;$$

$$y_4 = DA_2\bar{A}_1\bar{A}_0;$$

$$y_5 = DA_2\bar{A}_1A_0;$$

$$y_6 = DA_2A_1\bar{A}_0;$$

$$y_7 = DA_2A_1A_0.$$

Из полученных выражений видно, что схема демультиплексора практически полностью аналогична дешифратору, только вместо управляющего входного сигнала V на вход демультиплексора подается информационный сигнал D . Схема демультиплексора приведена на рис. 1.12.

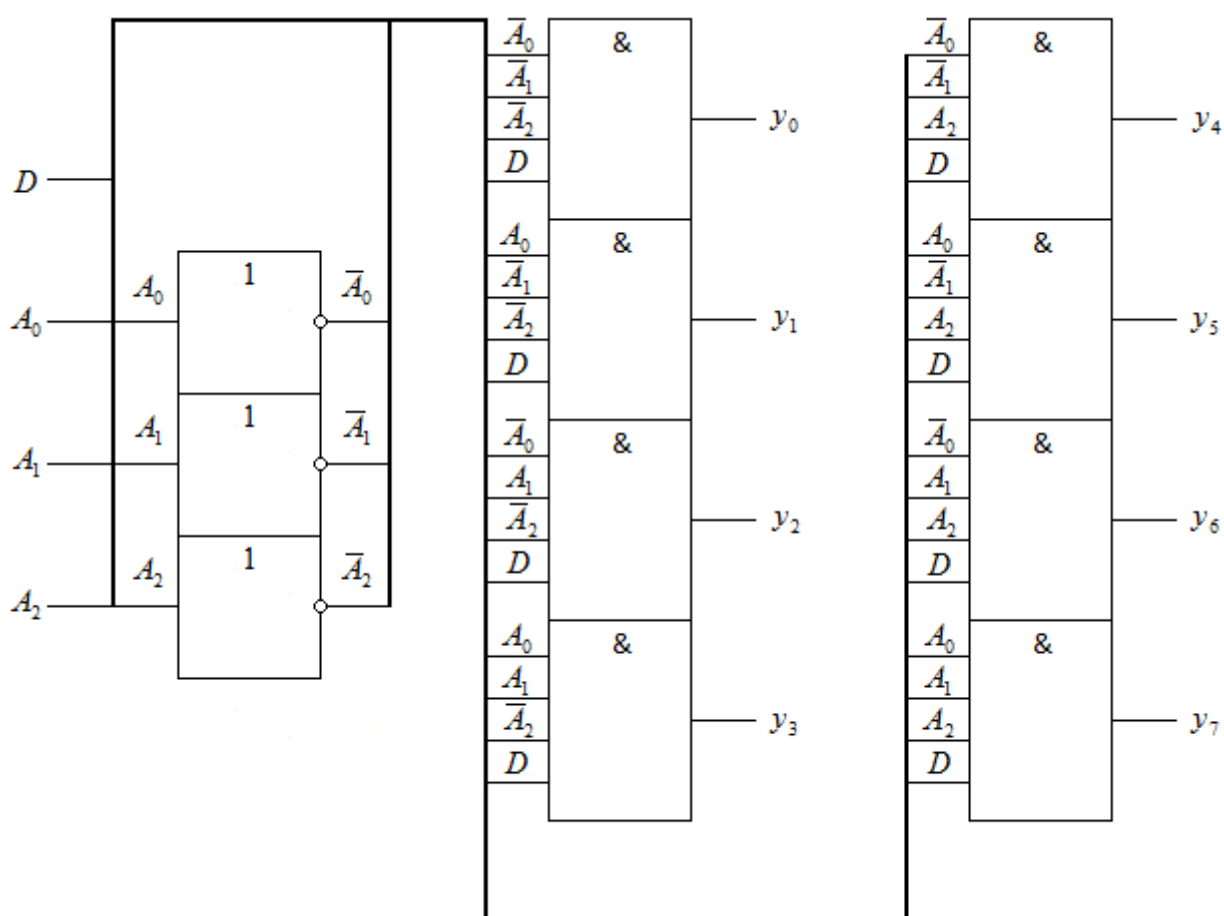


Рис.1.12. Схема демультиплексора

Схему демультиплексора можно представить в виде дешифратора, управляемого адресом, и конъюнкторов, управляемых информационным сигналом и выходами дешифратора (рис. 1.13).

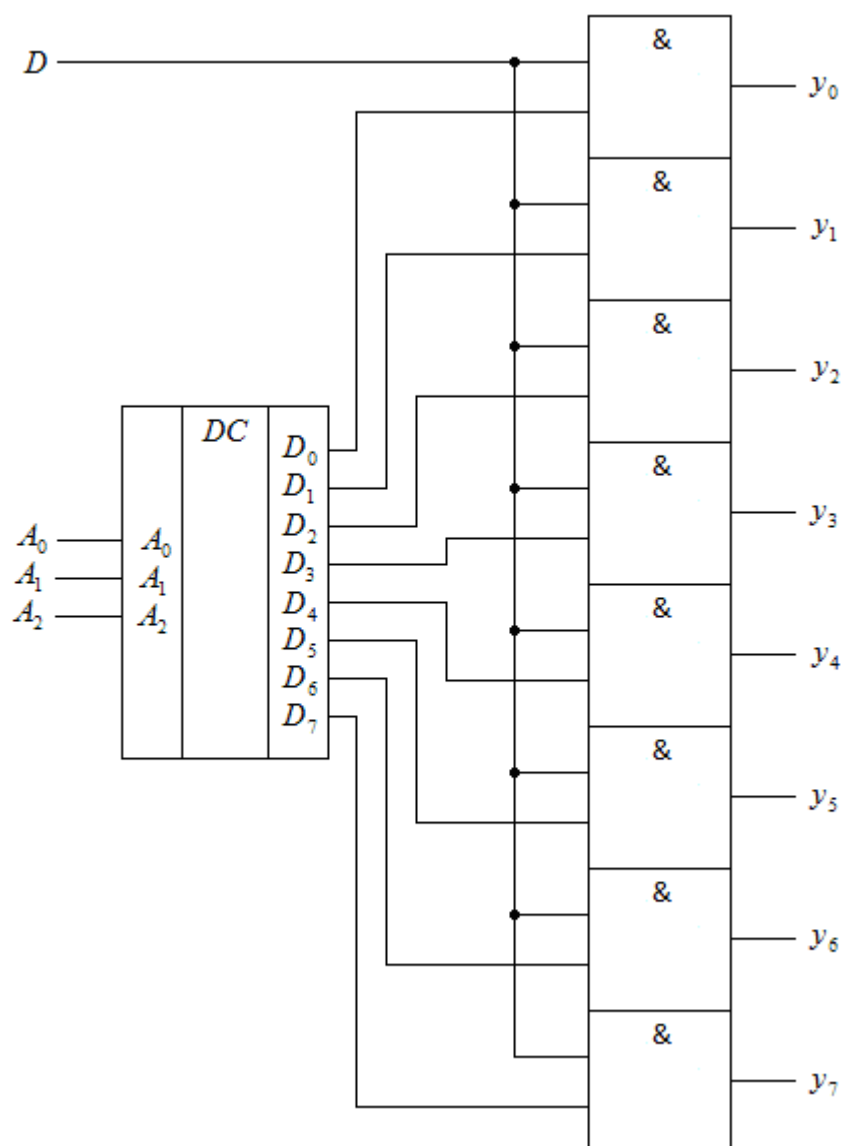


Рис. 1.13. Схема демультиплексора с дешифратором

Демультиплексоры используются в системах центрального оповещения подчиненных и абонентов, а также в линиях связи с временным разделением каналов между постоянными адресатами.

1.6.5 Сумматоры

Полусумматор. Полусумматор – арифметическое устройство, выполняющее сложение двух операндов с организацией переноса.

Полусумматор имеет два входа и два выхода. Схематическое изображение полусумматора, приведено на рис. 1.14.

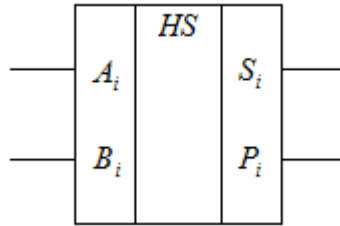


Рис. 1.14. Схематическое изображение полусумматора

Под сокращением *HS* подразумевается полусумматор (*half-summator*). Таблица истинности полусумматора представлена ниже (табл. 1.9).

Таблица 1.9

Таблица истинности полусумматора

Вход		Выход	
A_i	B_i	S_i	P_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Из таблицы истинности следуют аналитические выражения в СДНФ функции суммы S_i и сигнала переноса разряда P_i :

$$S_i = \bar{A}_i B_i + A_i \bar{B}_i;$$

$$P_i = A_i B_i.$$

По этим формулам синтезирована схема, показанная на рис. 1.15. Используя закон Де-Моргана можно минимизировать функцию S_i :

$$S_i = \bar{A}_i B_i + A_i \bar{B}_i = (A_i + B_i)(\bar{A}_i + \bar{B}_i) = (A_i + B_i)\overline{A_i B_i} = (A_i + B_i)\bar{P}_i.$$

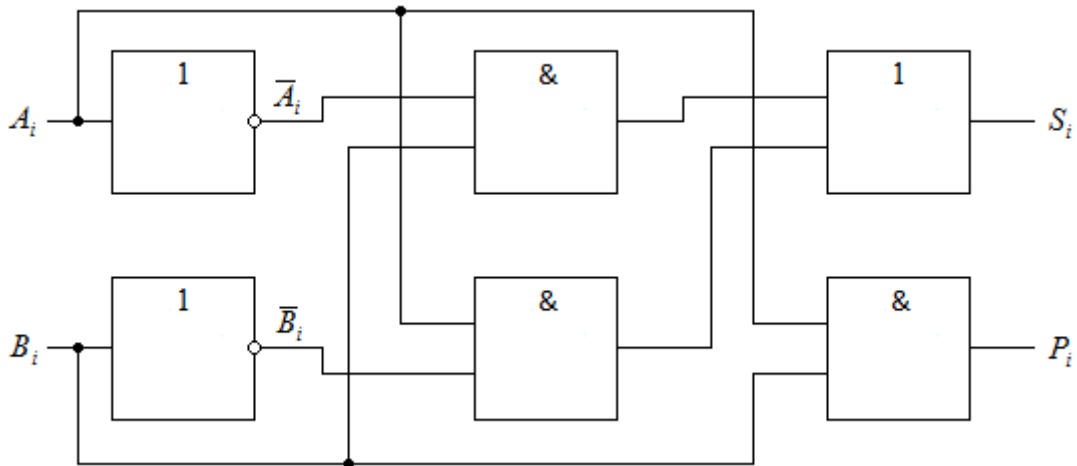
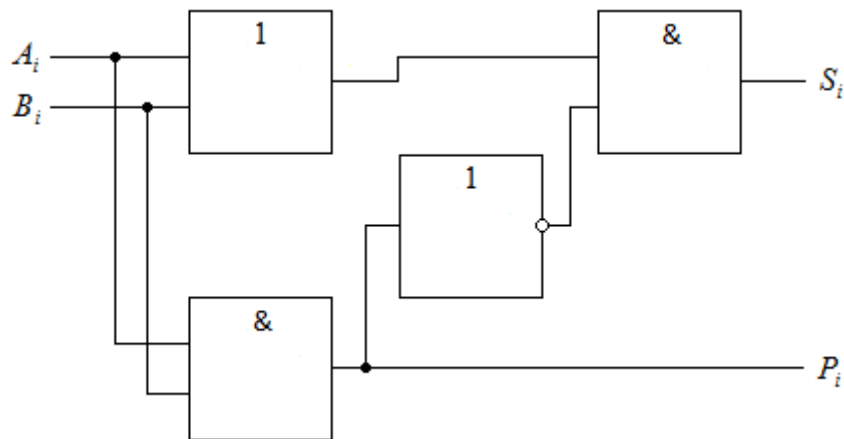


Рис. 1.15. Синтезированная схема полусумматора

Минимизированная схема полусумматора будет иметь вид:



Полный сумматор. При поразрядном суммировании полусумматор используется только при суммировании младшего разряда, когда на входах только два операнда A_0 и B_0 . Но начиная со второго разряда, с операндами A_1 и B_1 складывается перенос нулевого разряда P_0 , который может быть равен 0 или 1, это сложение выполняется на полном сумматоре.

Одноразрядный полный сумматор в отличие от полусумматора имеет три входа: для двух разрядов слагаемых A_i и B_i , а также сигнала переноса младшего разряда P_{i-1} . Условное обозначение сумматора SM – сокращение от *sum*mator. Схематическое изображение сумматора, приведено на рис. 1.16.

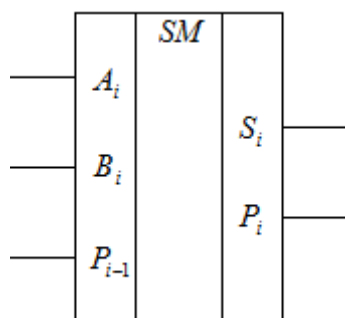


Рис. 1.16. Схематическое изображение сумматора

Таблица истинности полного сумматора представлена ниже (табл. 1.10).

Таблица 1.10

Таблица истинности полного сумматора

Вход			Выход	
A_i	B_i	P_{i-1}	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Характеристические уравнения полного сумматора в СДНФ будут иметь вид:

$$S_i = \bar{A}_i \bar{B}_i P_{i-1} + \bar{A}_i B_i \bar{P}_{i-1} + A_i \bar{B}_i \bar{P}_{i-1} + A_i B_i P_{i-1},$$

$$P_i = \bar{A}_i B_i P_{i-1} + A_i \bar{B}_i P_{i-1} + A_i B_i \bar{P}_{i-1} + A_i B_i P_{i-1}.$$

Схема полного сумматора, соответствующая данным уравнениям, представлена на рис. 1.17.

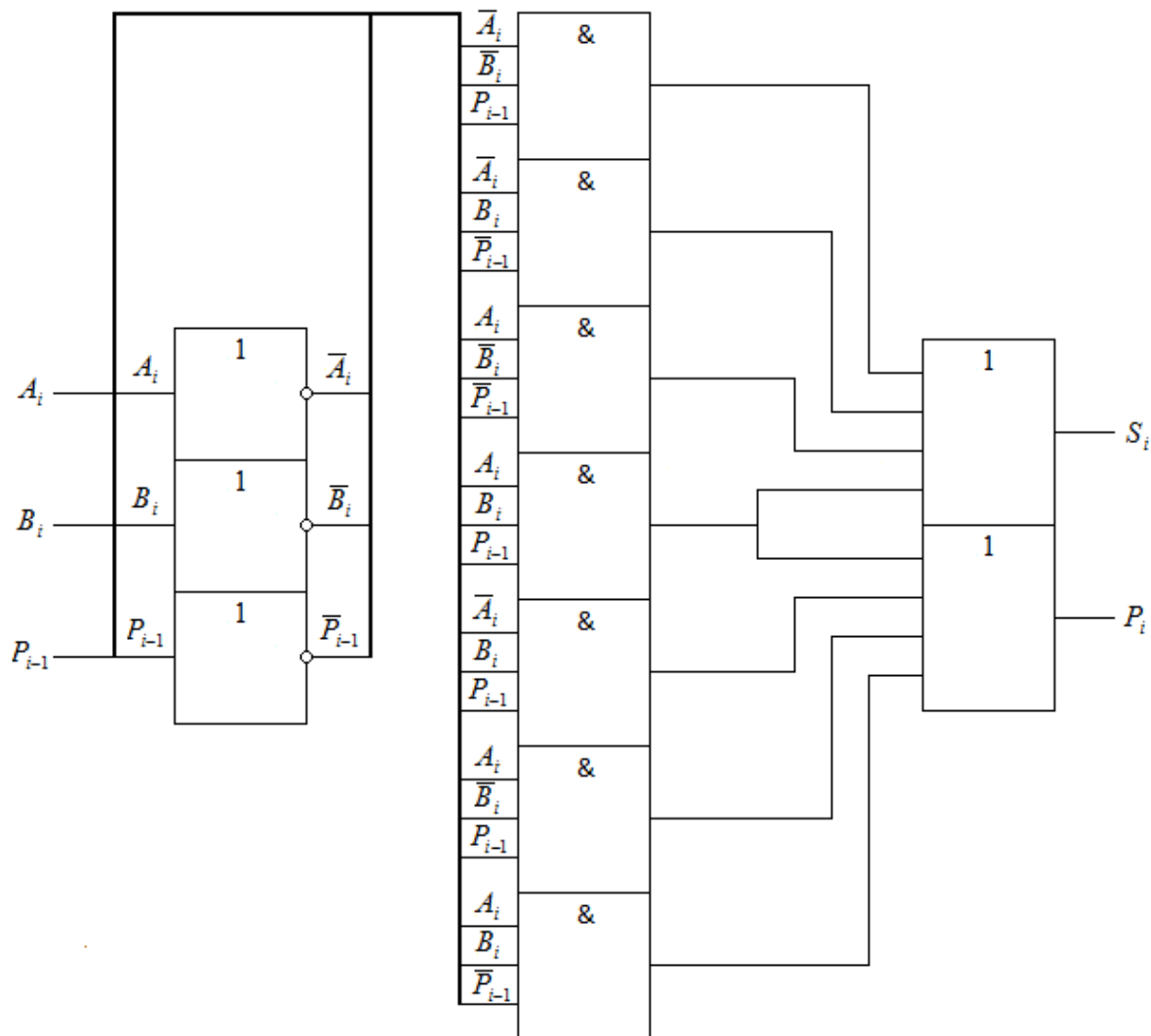


Рис. 1.17. Синтезированная схема сумматора

Минимизируем схему полного сумматора по правилам карт Карно. Составим карту Карно для функции S_i :

Таблица 1.11

Карта Карно для функции S_i

A_i	$B_i P_{i-1}$			
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Из таблицы истинности для S_i следует, что она не минимизируется. Но при этом ее можно представить как сумму трех аргументов по модулю 2:

$$S_i = A_i \oplus B_i \oplus P_{i-1}.$$

Составим карту Карно для P_i :

Таблица 1.12

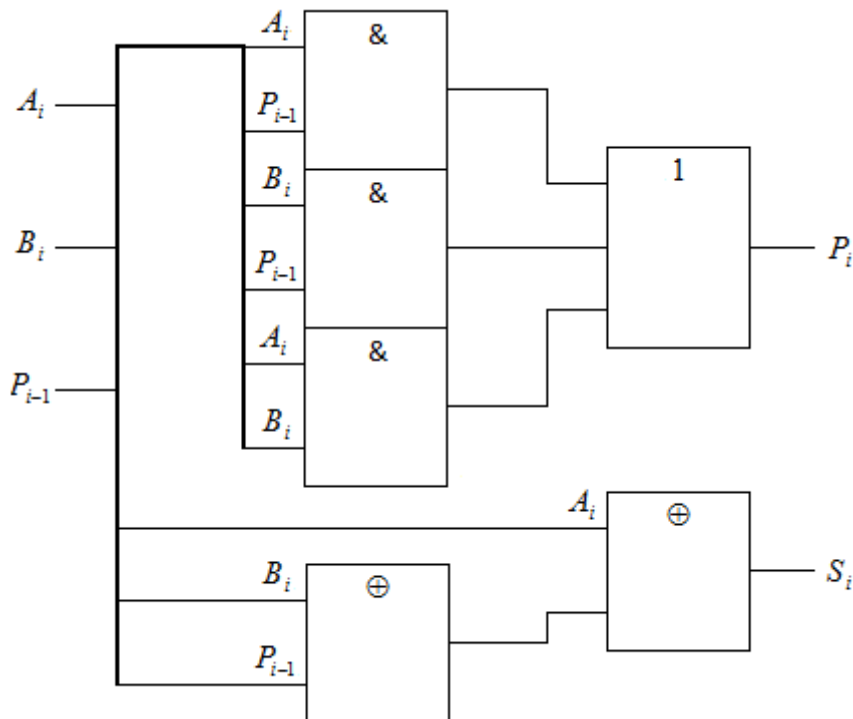
Карта Карно для P_i

	$B_i P_{i-1}$			
A_i	00	01	11	10
0	0	0	1	0
1	0	1	1	1

В результате склеивания получаем три контура по две 1 в каждом. В соответствии с правилами минимизации уравнение для P_i примет вид

$$P_i = A_i P_{i-1} + B_i P_{i-1} + A_i B_i.$$

Таким образом, минимизированная схема полного сумматора будет иметь вид:



Есть и другой вариант реализации полного сумматора. Из таблицы истинности полного сумматора следует, что одноразрядный полный сумматор можно синтезировать на основе трех одноразрядных полусумматоров по схеме, представленной на рис. 1.18.

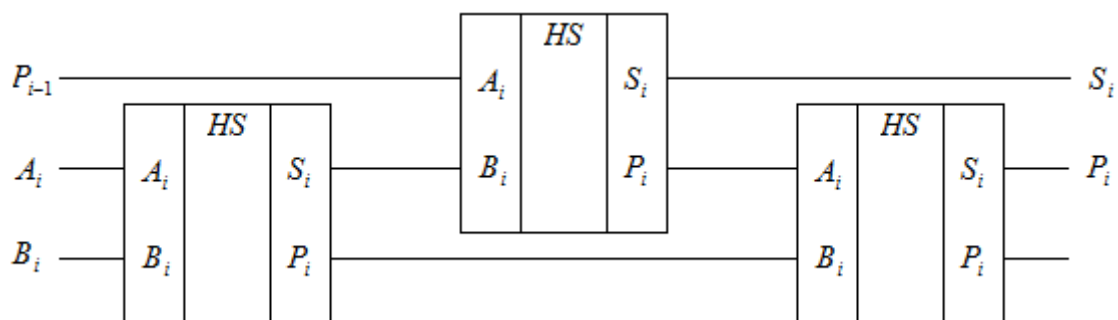


Рис. 1.18. Схема сумматора на основе трех одноразрядных полусумматоров

1.7 Последовательные цифровые устройства

В отличие от комбинационных цифровых устройств последовательные цифровые устройства (ПЦУ) обладают памятью по предыдущему состоянию, поэтому они реализуют функцию, зависящую не только от вектора внешних аргументов X^n , действующих в данном n -м такте, но и от внутреннего состояния устройства Q^{n-1} в предыдущем $(n-1)$ -м такте:

$$Q^n = f(X^n, Q^{n-1}).$$

Если последовательно производить замену Q^{n-1} на $f(X^{n-1}, Q^{n-2})$ и т. д. до начального состояния Q^0 , получим выражение

$$Q^n = f(X^n, Q^{n-1}) = f(X^n, X^{n-1}, Q^{n-2}) = \dots = f(X^n, X^{n-1}, \dots, X^1, Q^0),$$

свидетельствующее о том, что функция в данном n -м такте зависит от *последовательности векторов аргументов, поданных в предыдущих тактах*. Отсюда и произошло название *последовательные цифровые устройства*.

Основным критерием качества ПЦУ является быстродействие, оцениваемое двумя параметрами:

1) *временем установления* (записи, задержки) T_z — длительностью задержки изменения выходного сигнала относительно момента появления входного сигнала;

2) *разрешающим временем* T_p — минимальным периодом повторения изменений входного сигнала.

Последовательные цифровые устройства классифицируются по следующим признакам: функциональному, способу управления, наличию синхронизации, структуре.

По функциональному признаку ПЦУ подразделяются на триггеры, регистры, счетчики.

По способу управления ПЦУ делятся на управляемые потенциалом и управляемые импульсами (перепадами напряжения). В свою очередь ПЦУ, управляемые потенциалом, могут быть прямого и инверсного управления.

Прямое управление потенциалом означает, что активным является потенциал U_H (*high* – высокий), превосходящий пороговый уровень, и ему присваивается значение логической единицы (рис. 1.19, а). При инверсном управлении активным потенциалом является потенциал низкого уровня U_L (*low* – низкий), и ему присваивается значение логической единицы (рис. 1.19, б). Управление потенциалом еще называется *статическим*.

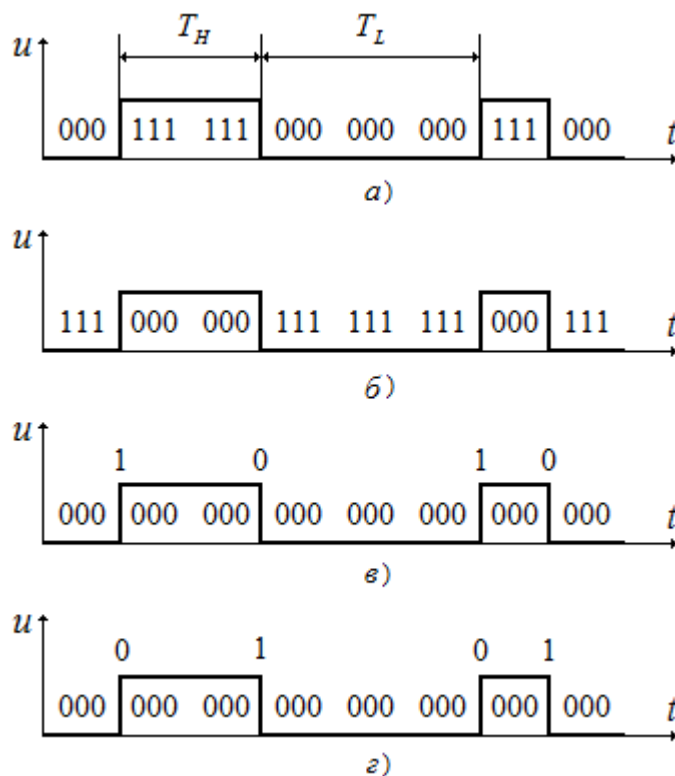


Рис. 1.19. Классификация видов управления ПЦУ:

а – по высокому потенциалу; б – по низкому потенциалу;
 в – по положительному перепаду напряжения; г – по отрицательному перепаду напряжения

ПЦУ, управляемые по перепаду напряжения, также делятся на две категории: управляемые по положительному (рис. 1.19, в) и отрицательному (рис. 1.19, г) перепадам. Такое управление называют *динамическим*, или *импульсным*.

По наличию синхронизации ПЦУ делятся на синхронные и асинхронные. Синхронными называются ПЦУ, на которые помимо

информационных входных сигналов подаются импульсы синхронизации, позволяющие устанавливать соответствие момента переключения данного устройства с определенными событиями в системе и обеспечивать временную упорядоченность действий параллельных процессов. В синхронных ПЦУ любые изменения на информационных входах не приводят к изменению состояния выходов, пока не подан соответствующий сигнал синхронизации.

В *асинхронных* ПЦУ входы синхронизации отсутствуют, поэтому переключения происходят вслед за изменениями потенциалов на информационных входах.

По *структурному признаку* различают ПЦУ параллельного, последовательного и смешанного (комбинированного) типов.

1.7.1 Триггеры

Триггер – элементарный цифровой автомат, обладающий памятью и служащий для записи и хранения одного бита информации.

Триггер может находиться в одном из двух устойчивых состояний: 0 или 1 и переходить из одного состояния в другое под воздействием входных сигналов. При отсутствии входных сигналов состояние триггера определяется только сигналами обратной связи, что отождествляется с внутренним состоянием триггера или его памятью.

Основные свойства триггерных устройств:

1) возможное число внутренних состояний два: 0 или 1 и обозначается Q ;

2) число выходных переменных – одно – Q (прямой выход), но есть и инверсный – \bar{Q} ;

3) число входов определяется типом триггера (от 2 до 5). Входные сигналы могут быть информационные, управляющие, синхроимпульсы;

4) основные типы триггеров по логическому функционированию: RS , D , JK , T , RST , R , E , V и т. д.

Асинхронный RS -триггер. Асинхронный RS -триггер имеет два входа:

1) вход S (от *set* – устанавливать);

2) вход R (от *reset* – восстанавливать).

Отсюда и второе название RS -триггера – *триггер с установочными входами*.

Приведем описание функции RS -триггеров:

- если в n -м такте на оба входа триггера поданы сигналы логических нулей, он сохраняет состояние предыдущего $(n-1)$ -го такта;
- если на вход S подан сигнал логической единицы, а на вход R – логического нуля, триггер устанавливается в состояние единицы независимо от того, в каком состоянии он находился в предыдущем такте;
- если на вход R подан сигнал логической единицы, а на вход S – логического нуля, триггер устанавливается в состояние нуля независимо от того, в каком состоянии он находился в предыдущем такте;
- одновременная подача логических единиц на оба входа запрещена.

Таблица 1.13

Таблица истинности асинхронного RS -триггера

Вход			Выход
R	S	Q^{n-1}	Q^n
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	*
1	1	1	*

В более компактной форме таблицу истинности можно переписать следующим образом (табл. 1.14).

Таблица 1.14

Свернутая таблица истинности

Вход		Выход
R	S	Q^n
0	0	Q^{n-1}
0	1	1
1	0	0
1	1	*

Условное графическое изображение RS -триггера приведено на рис. 1.20.

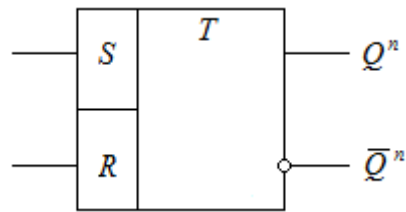


Рис. 1.20. Условное графическое изображение RS -триггера

На основании таблицы истинности RS -триггера составим его карту Карно: (табл. 1.15).

Таблица 1.15

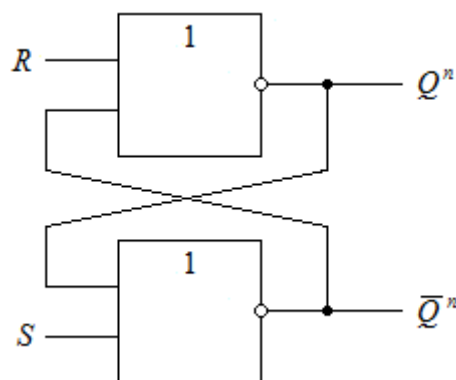
Карта Карно RS -триггера

	SQ^{n-1}			
R	00	01	11	10
0	0	1	1	1
1	0	0	*	*

В результате склеивания получаем два контура по две и четыре 1 в каждом. В соответствии с правилами минимизации уравнение для Q^n примет вид

$$Q^n = S + \bar{R}Q^{n-1}.$$

Таким образом, минимизированная схема RS -триггера будет иметь вид:



Устройство с такой схемой представляет собой запоминающий элемент, который называется *дьюнктивной бистабильной ячейкой*.

Доопределяя таблицу истинности RS -триггера в состояниях неопределенности, можно сформировать новые разновидности

триггеров, которые отличаются от *RS*-триггера только своим поведением при двух входных единичных сигналах:

Таблица 1.16

Вход			Выход			
<i>R</i>	<i>S</i>	Q^{n-1}	Q^n			
1	1	0	0	1	0	1
1	1	1	0	1	1	0
Тип триггера			<i>R</i>	<i>S</i>	<i>E</i>	<i>JK</i>

Синхронный *RST*-триггер. Синхронный *RST*-триггер отличается от асинхронного *RS*-триггера наличием дополнительного входа синхронизации, обозначаемого на схемах латинской буквой *C* (*clock* – часы).

Если на входе синхронизации *C* установлен низкий потенциал (логический ноль), то триггер сохраняет предыдущее состояние независимо от значения информационных сигналов *R* и *S*. При подаче на вход синхронизации *C* логической единицы триггер становится управляемым аналогично асинхронному.

Таблица 1.17

Таблица истинности синхронного *RST*-триггера

Вход				Выход
<i>C</i>	<i>R</i>	<i>S</i>	Q^{n-1}	Q^n
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	*
1	1	1	1	*

В более компактной форме таблицу истинности можно переписать следующим образом (табл. 1.18).

Таблица 1.18

Свернутая таблица истинности

Вход			Выход
C	R	S	Q^n
0	*	*	Q^{n-1}
1	0	0	Q^{n-1}
1	0	1	1
1	1	0	0
1	1	1	*

Условное графическое изображение RST -триггера, приведено на рис. 1.21.

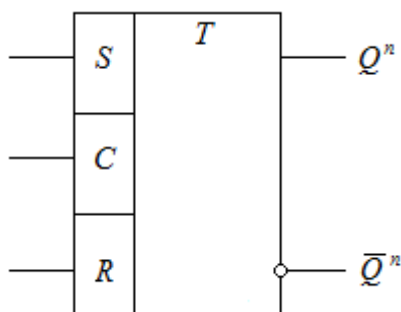


Рис. 1.21. Условное графическое изображение RST -триггера

На основании таблицы истинности RST -триггера составим его карту Карно.

Таблица 1.19

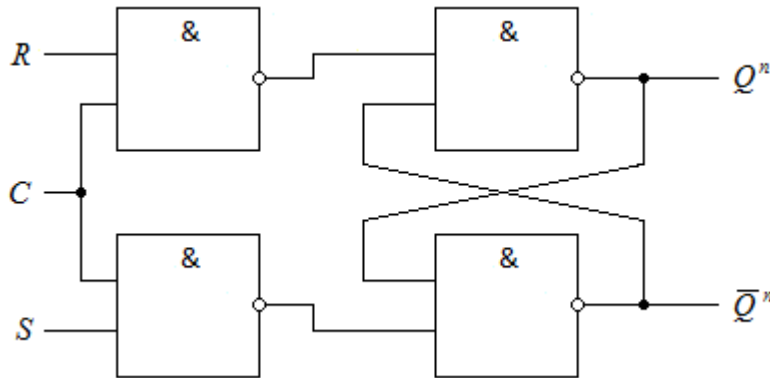
Карта Карно RST -триггера

CR	SQ^{n-1}			
	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	0	*	*
10	0	1	1	1

В результате склеивания получаем три контура, содержащего по четыре 1. В соответствии с правилами минимизации уравнение для Q^n примет вид

$$Q^n = \bar{C}Q^{n-1} + CS + \bar{R}Q^{n-1}.$$

Таким образом, минимизированная схема *RST*-триггера будет иметь вид:



Синхронный *D*-триггер. Триггера типа *D* бывают только синхронными. На их базе строятся такие вычислительные устройства, как регистры, счетчики, делители.

Синхронный *D*-триггер работает следующим образом. Когда на вход синхронизации *C* подается логический нуль, устройство сохраняет предыдущее состояние. При наличии логической единицы на входе синхронизации на выходе триггера устанавливается потенциал, равный потенциалу на информационном входе триггера *D*. Поскольку сигнал на выходе появляется с некоторой задержкой, *D*-триггеры называются триггерами задержки (от *delay* – задержка).

Таблица 1.20

Таблица истинности синхронного *D*-триггера

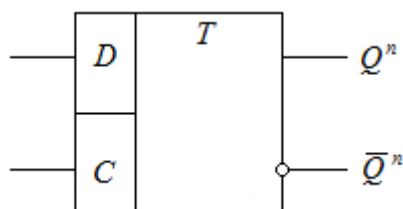
Вход		Выход	
<i>C</i>	<i>D</i>	Q^{n-1}	Q^n
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
↑	0	0	0
↑	0	1	0
↑	1	0	1
↑	1	1	1

В более компактной форме таблицу истинности можно переписать следующим образом (табл. 1.21).

Свернутая таблица истинности

Вход		Выход
C	D	Q^n
0	*	Q^{n-1}
↑	0	0
↑	0	0
↑	1	1
↑	1	1

Условное графическое изображение D -триггера, приведено на рис. 1.22.

Рис. 1.22. Условное графическое изображение D -триггера

На основании таблицы истинности D -триггера составим его карту Карно:

Таблица 1.22

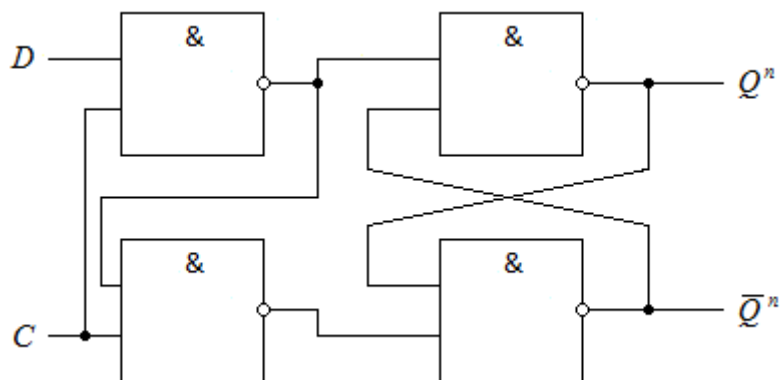
Карта Карно D -триггера

C	DQ^{n-1}			
	00	01	11	10
0	0	1	1	0
1	0	0	1	1

В результате склеивания получаем два контура, содержащие по две 1. В соответствии с правилами минимизации уравнение для Q^n примет вид

$$Q^n = \bar{C}Q^{n-1} + CD.$$

Таким образом, минимизированная схема D -триггера будет иметь вид



T -триггер. T -триггер не существует как самостоятельный элемент серии интегральных схем. Обычно D или JK триггеры переводят в режим T -триггера при введении определенных связей. Название T -триггер происходит от английского *toggle flip-flop* – опрокидываться.

T -триггер функционирует следующим образом: если на информационный вход T подан логический нуль, триггер сохраняет предыдущее состояние; при подаче на вход T логической единицы триггер переключается в противоположное состояние.

Исходя из этого, таблица истинности триггера имеет вид (табл. 1.23):

Таблица 1.23

Таблица истинности T -триггера

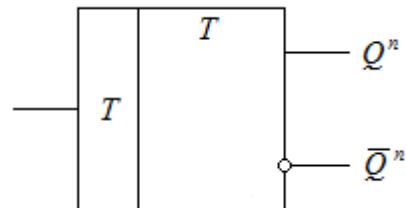
Вход		Выход
T	Q^{n-1}	Q^n
0	0	0
0	1	1
1	0	1
1	1	0

В более компактной форме таблицу истинности можно переписать следующим образом (табл. 1.24).

Свернутая таблица истинности

Вход	Выход
T	Q^n
0	Q^{n-1}
1	\overline{Q}^{n-1}

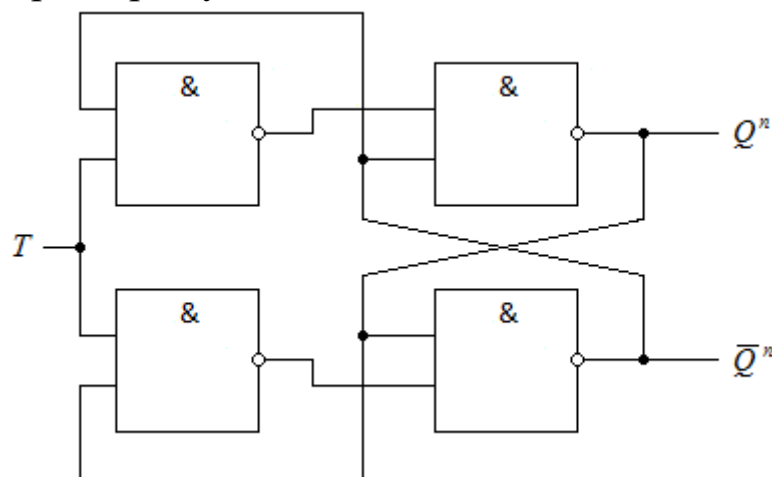
Условное графическое изображение T -триггера, приведено на рис. 1.23.

Рис. 1.23. Условное графическое изображение T -триггера

Из таблицы истинности следует уравнение T -триггера в СДНФ

$$Q^n = \overline{T}Q^{n-1} + T\overline{Q}^{n-1}.$$

Схема T -триггера будет иметь вид



1.7.2 Регистры

Регистром называется устройство, которое служит для запоминания машинного слова (или его части) в процессе выполнения вычислительных операций, а также для преобразования слов посредством сдвига их вдоль регистра на один или несколько разрядов.

Собираются регистры обычно на триггерах или логических элементах, число которых, как правило, соответствует числу разрядов в машинном слове.

Обозначаются регистры *RG* от английского слова *registor*.

Применение регистров:

1) РОНЫ (регистры общего назначения) как адресная сверхоперативная память;

2) регистр команд для хранения кода операции из программы, записанной в оперативной памяти, для дальнейшего ввода в схему управления, где организуется микропрограмма для выполнения кода операции;

3) буферные регистры для хранения операндов, с которыми работает арифметико-логическое устройство;

4) флажковый регистр, куда заносится информация, по которой реализуются условные переходы программы;

5) аккумулятор – специализированный регистр для хранения и различных преобразований результатов арифметических и логических операций над операндами;

6) регистр адреса, в который заносится адрес программы пользователя для обращений к оперативной памяти;

7) регистры для организации ускоренного умножения двоичных чисел;

8) ввод в регистр машинного слова, хранение и вывод его из регистра;

9) преобразование последовательного кода в параллельный и наоборот;

10) сдвиг двоичного числа вправо или влево на определенное число разрядов;

11) выполнение поразрядно логических операций над хранимым машинным словом и вновь вводимым.

Регистры по способу ввода и вывода информации можно подразделить на:

1) параллельные регистры памяти, когда информация вводится и выводится параллельным кодом;

2) последовательные регистры сдвига осуществляют последовательный, поразрядный сдвиг кодовой комбинации в сторону старшего (вправо) или младшего (влево) разряда с помощью тактирующих импульсов;

3) параллельно-последовательные (универсальные) регистры способны вводить параллельный код, а выводить в виде последовательного; либо вводить последовательный код, а выводить параллельный.

Параллельный регистр. В параллельном регистре каждый из триггеров имеет свой независимый информационный вход и свой независимый информационный выход.

Параллельные регистры делятся на две группы:

- 1) регистры, срабатывающие по фронту тактового импульса;
- 2) регистры, срабатывающие по уровню тактового импульса.

Чаще всего в цифровых схемах используют регистры, управляемые фронтом тактовых импульсов.

Условное графическое изображение параллельного регистра, приведено на рис. 1.24.

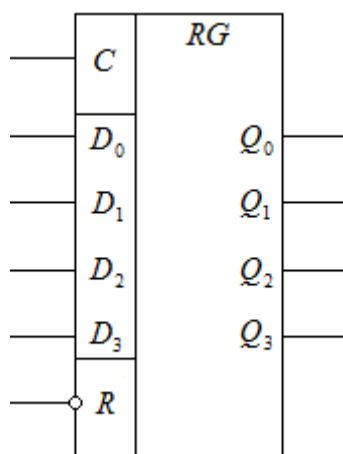


Рис. 1.24. Условное графическое изображение параллельного регистра

В данной схеме вход C синхронизирующий вход, \bar{R} – вход обнуления регистра (обнуление происходит при подаче на вход 0). D_0, D_1, D_2, D_3 – это информационные входы, на них одновременно подается двоичный код вида $D_3 D_2 D_1 D_0$, который записывается в регистр (при поступлении синхронизирующего сигнала C). При этом выходы Q_0, Q_1, Q_2, Q_3 принимают значение этого двоичного кода. Т. е. каждый разряд работает как D -триггер.

Таблица истинности параллельного регистра

Вход		Выход			
C	\bar{R}	Q_0^n	Q_1^n	Q_2^n	Q_3^n
1	1	D_0	D_1	D_2	D_3
0	1	Q_0^{n-1}	Q_1^{n-1}	Q_2^{n-1}	Q_3^{n-1}
*	0	0	0	0	0

Уравнения для регистра в СДНФ:

$$Q_0^n = D_0 \bar{C} \bar{R} + Q_0^{n-1} \bar{C} \bar{R} = \bar{R} (D_0 C + Q_0^{n-1} \bar{C});$$

$$Q_1^n = D_1 \bar{C} \bar{R} + Q_1^{n-1} \bar{C} \bar{R} = \bar{R} (D_1 C + Q_1^{n-1} \bar{C});$$

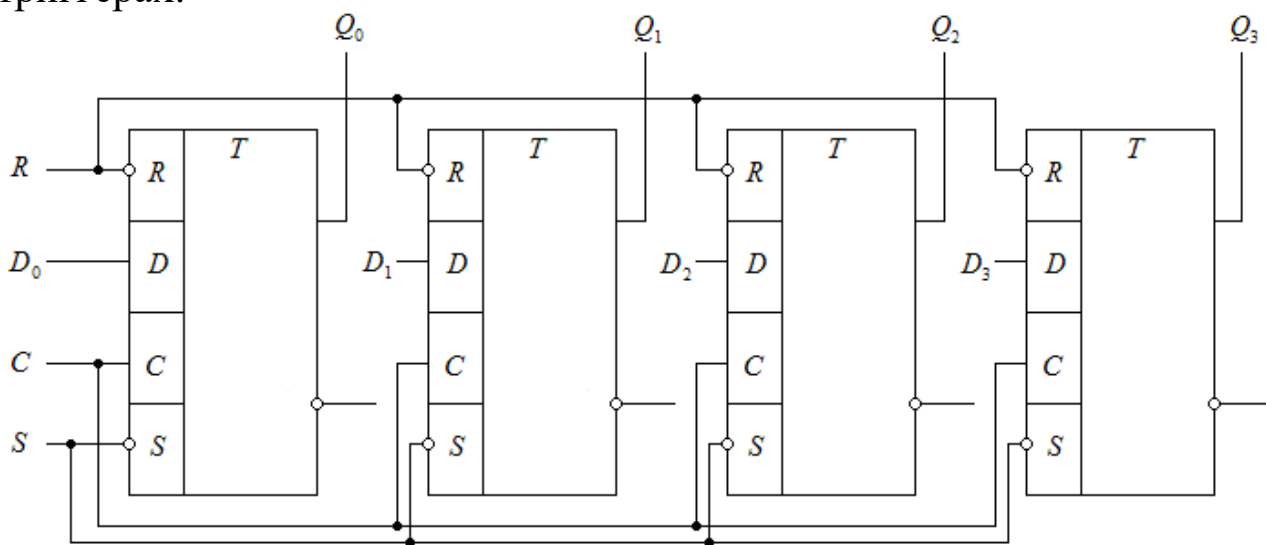
$$Q_2^n = D_2 \bar{C} \bar{R} + Q_2^{n-1} \bar{C} \bar{R} = \bar{R} (D_2 C + Q_2^{n-1} \bar{C});$$

$$Q_3^n = D_3 \bar{C} \bar{R} + Q_3^{n-1} \bar{C} \bar{R} = \bar{R} (D_3 C + Q_3^{n-1} \bar{C}).$$

Каждое уравнение соответствует уравнению D -триггера

$$Q^n = \bar{R} (DC + Q^{n-1} \bar{C}).$$

Поэтому схему параллельного регистра можно построить на D -триггерах.



Регистр сдвига (последовательный регистр).

Последовательные регистры имеют один вход и n -параллельных выходов. В таких регистрах все триггеры соединены последовательно. Тактовые входы всех триггеров объединены между собой.

Условное графическое изображение последовательного регистра приведено на рис. 1.25.

Принцип работы последовательного регистра: при подаче на вход ввода последовательного кода регистра каждый тактовый сигнал

сдвигает разряд регистра из одного триггера в другой в сторону младшего разряда (влево и тогда этот вход обозначается DL), либо в сторону старшего разряда (вправо и вход регистра обозначается DR).

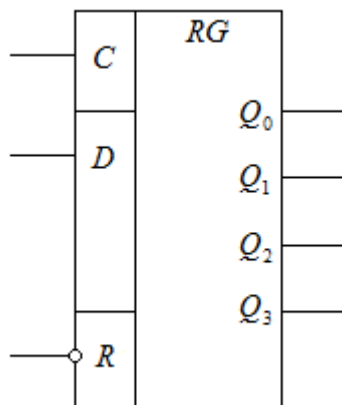


Рис. 1.25. Условное графическое изображение последовательного регистра

В последовательном регистре, в отличие от параллельного, существует связь между триггерами:

- 1) для регистра со сдвигом вправо связь входа триггера с выходом предыдущего триггера описывается уравнением $D_i = Q_{i-1}$;
- 2) для регистра со сдвигом влево связь входа триггера с выходом последующего записывается уравнением $D_i = Q_{i+1}$.

Регистры сдвига также наиболее часто реализуются на основе синхронных D -триггеров.

Построим регистр сдвига в сторону старшего разряда. Условное изображение такого регистра будет выглядеть, как показано на рис. 1.26.

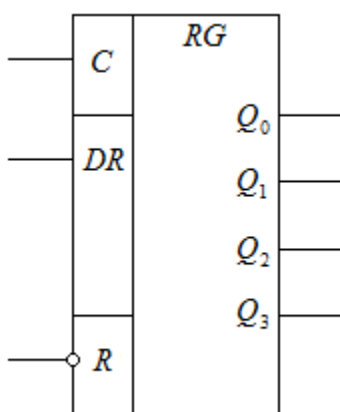


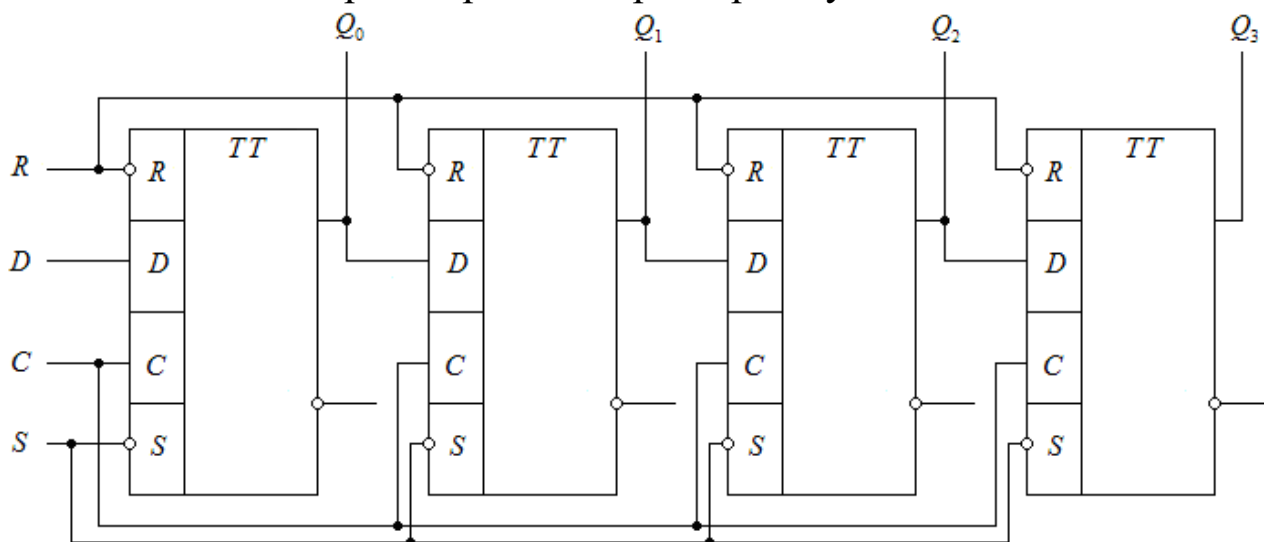
Рис. 1.26. Регистр сдвига в сторону старшего разряда

Пусть в регистр нужно записать двоичный код $D_0 D_1 D_2 D_3$, где D_0 – самый старший разряд кода, а D_3 – самый младший.

Таблица перехода состояний последовательного регистра в сторону старшего разряда

Вход		Выход			
C	D	Q_3^n	Q_2^n	Q_1^n	Q_0^n
0	*	Q_3^{n-1}	Q_2^{n-1}	Q_1^{n-1}	Q_0^{n-1}
↑	D_0	Q_2^{n-1}	Q_1^{n-1}	Q_0^{n-1}	D_0
↑	D_1	Q_1^{n-1}	Q_0^{n-1}	D_0	D_1
↑	D_0	Q_0^{n-1}	D_0	D_1	D_2
↑	D_0	D_0	D_1	D_2	D_3

Схема такого регистра на D -триггерах будет иметь вид:



Построим регистр сдвига в сторону младшего разряда. Условное изображение такого регистра будет выглядеть, как показано на рис. 1.27.

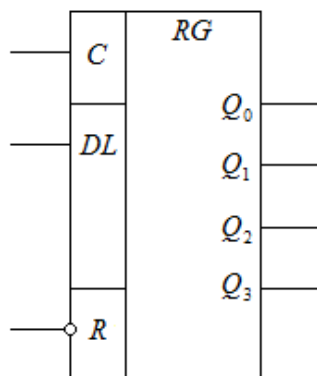


Рис. 1.27. Регистр сдвига в сторону младшего разряда

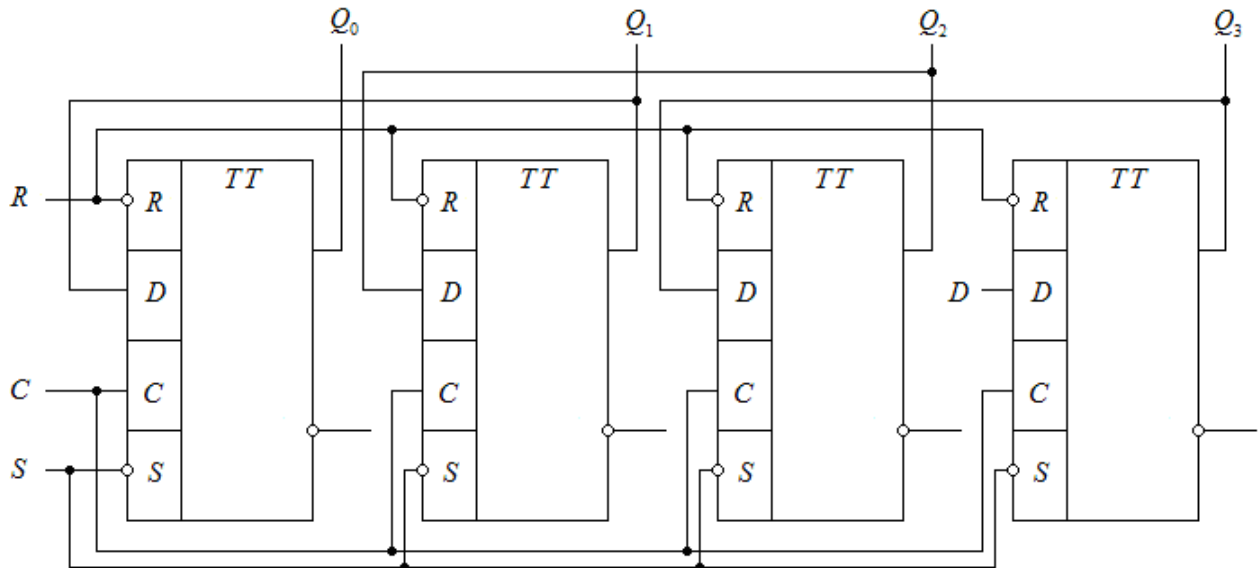
Пусть в регистр нужно записать двоичный код $D_0 D_1 D_2 D_3$, где D_0 – самый старший разряд кода, а D_3 – самый младший.

Таблица 1.27

Таблица перехода состояний последовательного регистра в сторону младшего разряда

Вход		Выход			
C	D	Q_3^n	Q_2^n	Q_1^n	Q_0^n
0	*	Q_3^{n-1}	Q_2^{n-1}	Q_1^{n-1}	Q_0^{n-1}
↑	D_3	D_3	Q_3^{n-1}	Q_2^{n-1}	Q_1^{n-1}
↑	D_2	D_2	D_3	Q_3^{n-1}	Q_2^{n-1}
↑	D_1	D_1	D_2	D_3	Q_3^{n-1}
↑	D_0	D_0	D_1	D_2	D_3

Схема такого регистра на D -триггерах будет иметь вид



1.7.3 Счетчики

Счетчиком называется автомат, выполняющий функции подсчета количества импульсов единичных сигналов, поступающих на его вход, а также функции формирования и запоминания некоторого двоичного кода, соответствующего этому количеству.

Сокращенное обозначение CT происходит от английского *counter* – счетчик.

Счетчики – наиболее распространенное цифровое устройство, используемое, например, для подсчета шагов программы, циклов сложения и вычитания при выполнении арифметических операций,

для преобразования кодов, в аналоговоцифровых и цифроаналоговых преобразователях, в организации сверхоперативной памяти и других цифровых устройствах.

Счетчики могут быть построены на основе счетных триггеров со специальными межразрядными связями на основе сдвигающих регистров (кольцевые счетчики).

Основные параметры счетчика:

1) емкость счетчика $K_{сч}$ характеризует максимальное число сигналов, которое может быть им сосчитано;

2) максимальная частота поступления счетных импульсов – f ;

3) время перехода счетчика из одного состояния в другое – $t_{сч}$.

Классификационные признаки счетчиков.

1) по порядку счета:

- с естественным порядком счета, когда значение кода каждого последующего состояния счетчика отличается на 1 от кода предыдущего состояния;

- с произвольным порядком счета, когда значения кодов соседних состояний счетчика могут отличаться более чем на 1.

2) счетчики с естественным порядком счета могут быть:

- простые: суммирующие, если коды расположены в нарастающем порядке (*Up – counter*) и вычитающие, если в убывающем (*Down – counter*);

- реверсивные, у которых направление перебора может меняться (*Up/Down counter*).

3) по коэффициенту счета $K_{сч}$ (модуль счета определяется числом разрешенных состояний):

- недвоичные, где $K_{сч} \leq 2^n$;

- двоичные, где $K_{сч} = 2^n$ (n – число триггеров в счетчике).

4) по способу переключения триггеров:

- асинхронные счетчики не требуют тактовых импульсов;

- синхронные счетчики требуют наличия тактовых импульсов.

5) по способу организации переноса между разрядами (связями между триггерами) и подачей счетных импульсов на входы триггеров, счетчики делятся:

- последовательные (непосредственные);

- параллельные;

- последовательно-параллельные для получения счетчиков с большим $K_{сч}$.

Асинхронные счетчики. Рассмотрим асинхронный счетчик для суммирования трех разрядов с $K_{сч}=8$ и с последовательным переносом между T -триггерами в счетном режиме.

Условное изображение данного асинхронного суммирующего счетчика показано на рис. 1.28.

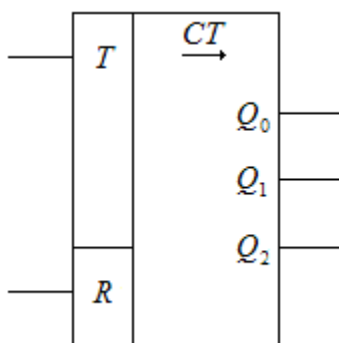


Рис. 1.28. Условное изображение асинхронного суммирующего счетчика

В данной схеме T – информационный (счетный) код, R – вход обнуления (сброса) счетчика и Q_0, Q_1, Q_2 – это выходы счетчика.

Таблица 1.28

Таблица функционирования счетчика

n	Q_2^{n-1}	Q_1^{n-1}	Q_0^{n-1}	Q_2^n	Q_1^n	Q_0^n
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0

Произведем синтез синхронного суммирующего счетчика на T -триггерах.

Таблица истинности синхронного T -триггера

Вход		Выход
T	Q^{n-1}	Q^n
0	0	0
0	1	1
1	0	1
1	1	0

Перепишем таблицу истинности T -триггера таким образом, чтобы были отображены значения входного параметра T при переходах триггера из предыдущего состояния в последующее.

Таблица 1.30

Упрощенная таблица истинности

$Q^{n-1} \rightarrow Q^n$	T
0→0	0
0→1	1
1→0	1
1→1	0

Составим карту Карно для состояния Q_0^n на основе таблицы функционирования счетчика. При этом вместо значения Q_0^n в клетках запишем значения $Q_0^{n-1} Q_0^n$, т. е. переход от предыдущего состояния к следующему.

Таблица 1.31

Карта Карно для Q_0^n

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	01	10	10	01
1	01	10	10	01

Аналогичным образом составим карты Карно для состояний Q_1^n и Q_2^n (табл. 1.32 и 1.33).

Таблица 1.32

Карта Карно для Q_1^n

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	00	01	10	11
1	00	01	10	11

Таблица 1.33

Карта Карно для Q_2^n

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	00	00	01	00
1	11	11	10	11

В полученные карты Карно запишем значения входов D_0 , D_1 , D_2 для триггеров, соответствующие этим переходам. В результате получим карты Карно для входов D_0 , D_1 , D_2 (табл. 1.34 – 1.36).

Таблица 1.34

Карта Карно для D_0

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	1	1	1	1
1	1	1	1	1

Таблица 1.35

Карта Карно для D_1

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Карта Карно для D_2

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	0	0	1	0
1	0	0	1	0

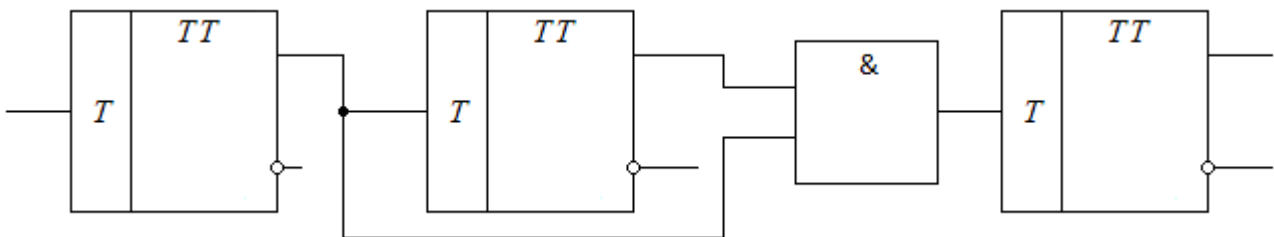
В результате склеивания получим уравнения для D_0 , D_1 , D_2 :

$$D_0 = 1;$$

$$D_1 = Q_0^{n-1};$$

$$D_2 = Q_1^{n-1} Q_0^{n-1}.$$

Исходя из полученных уравнений, построим схему суммирующего счетчика на T -триггерах:



Синхронные счетчики. Рассмотрим синхронный счетчик для суммирования трех разрядов с $K_{сч}=8$ и с последовательным переносом между D -триггерами в счетном режиме. Условное изображение данного синхронного суммирующего счетчика показано на рис. 1.29.

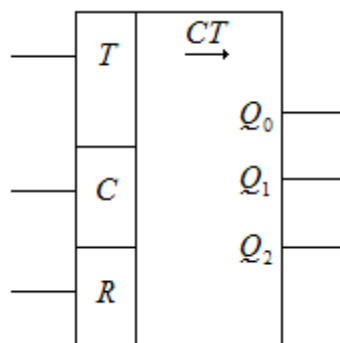


Рис. 1.29. Условное изображение синхронного суммирующего счетчика

В данной схеме T – информационный (счетный) код, R – вход обнуления (сброса) счетчика, C – синхронизирующий вход и Q_0 , Q_1 , Q_2 – это выходы счетчика.

Таблица 1.37

Таблица функционирования данного счетчика

n	Q_2^{n-1}	Q_1^{n-1}	Q_0^{n-1}	Q_2^n	Q_1^n	Q_0^n
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0

Произведем синтез синхронного суммирующего счетчика на D -триггерах.

Таблица 1.38

Таблица истинности синхронного D -триггера

Вход			Выход
C	D	Q^{n-1}	Q^n
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
↑	0	0	0
↑	0	1	0
↑	1	0	1
↑	1	1	1

Перепишем таблицу истинности D -триггера таким образом, чтобы были отображены значения входного параметра D при переходах триггера из предыдущего состояния в последующее ($C=1$).

Таблица 1.39

Упрощенная таблица истинности

C	$Q^{n-1} \rightarrow Q^n$	D
↑	0→0	0
↑	0→1	1
↑	1→0	0
↑	1→1	1

Составим карту Карно для состояния Q_0^n на основе таблицы функционирования счетчика. При этом вместо значения Q_0^n в клетках

запишем значения $Q_0^{n-1} Q_0^n$, т. е. переход от предыдущего состояния к следующему (табл. 1.40).

Таблица 1.40

Карта Карно для Q_0^n

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	01	10	10	01
1	01	10	10	01

Аналогичным образом составим карты Карно для состояний Q_1^n и Q_2^n (табл. 1.41, 1.42).

Таблица 1.41

Карта Карно для Q_1^n

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	00	01	10	11
1	00	01	10	11

Таблица 1.42

Карта Карно для Q_2^n

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	00	00	01	00
1	11	11	10	11

В полученные карты Карно запишем значения входов D_0, D_1, D_2 для триггеров, соответствующие этим переходам. В результате получим карты Карно для входов D_0, D_1, D_2 (табл. 1.43 – 1.45).

Таблица 1.43

Карта Карно для D_0

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	1	0	0	1
1	1	0	0	1

Таблица 1.44

Карта Карно для D_1

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	0	1	0	1
1	0	1	0	1

Таблица 1.45

Карта Карно для D_2

	$Q_1^{n-1} Q_0^{n-1}$			
Q_2^{n-1}	00	01	11	10
0	0	0	1	0
1	1	1	0	1

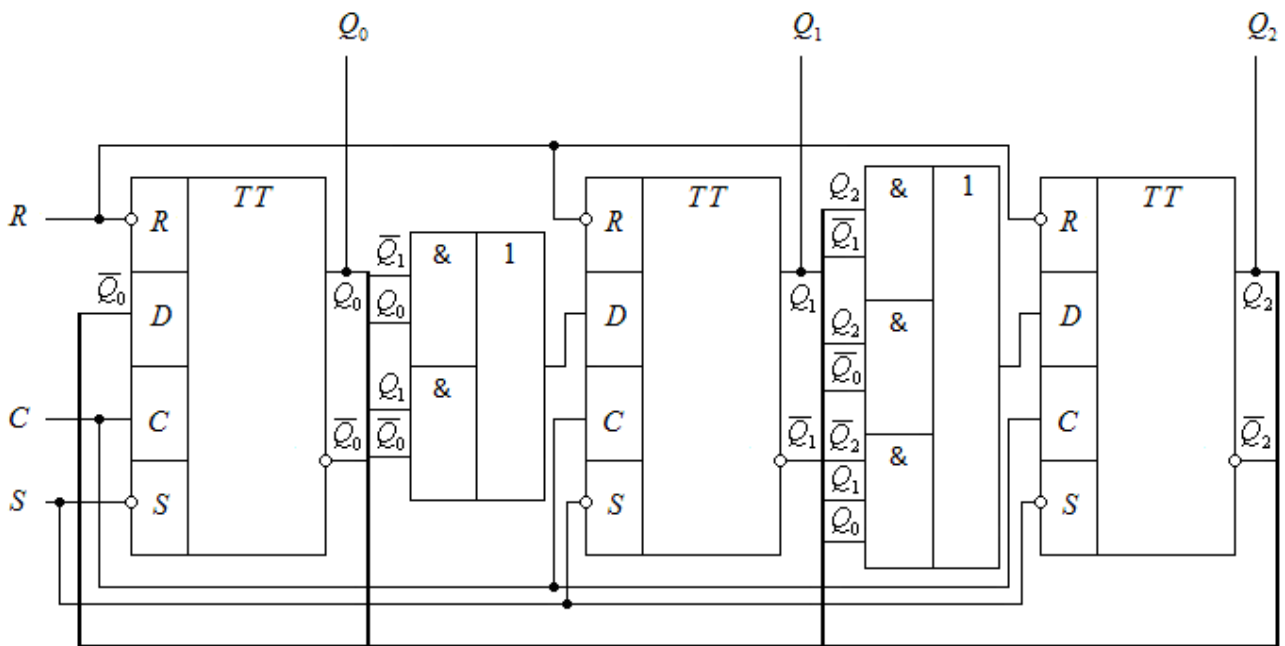
В результате склеивания получим уравнения для D_0 , D_1 , D_2 :

$$D_0 = \bar{Q}_0^{n-1};$$

$$D_1 = \bar{Q}_1^{n-1} Q_0^{n-1} + Q_1^{n-1} \bar{Q}_0^{n-1};$$

$$D_2 = Q_2^{n-1} \bar{Q}_1^{n-1} + Q_2^{n-1} \bar{Q}_0^{n-1} + \bar{Q}_2^{n-1} Q_1^{n-1} Q_0^{n-1}.$$

Исходя из полученных уравнений, построим схему суммирующего счетчика на D -триггерах:



Таким же образом можно построить синхронный счетчик для вычитания трех разрядов с $K_{сч}=8$ и с последовательным переносом между D -триггерами в счетном режиме. Условное изображение данного синхронного вычитающего счетчика показано на рис. 1.30.

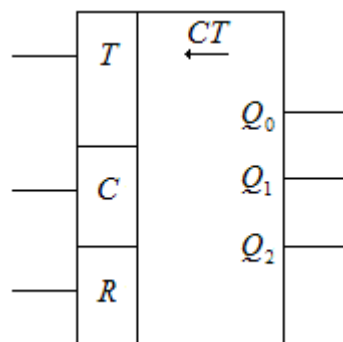


Рис. 1.30. Условное изображение синхронного вычитающего счетчика

Таблица 1.46

Таблица функционирования данного счетчика

n	Q_2^{n-1}	Q_1^{n-1}	Q_0^{n-1}	Q_2^n	Q_1^n	Q_0^n
0	1	1	1	1	1	0
1	1	1	0	1	0	1
2	1	0	1	1	0	0
3	1	0	0	0	1	1
4	0	1	1	0	1	0
5	0	1	0	0	0	1
6	0	0	1	0	0	0
7	0	0	0	1	1	1

Синтез схемы на *D*-триггерах, осуществляется аналогично синтезу суммирующего счетчика.

1.8 Преобразователи кодов

Преобразователи кодов предназначены для перевода чисел из одной формы представления в другую. Например, при вводе информации в ЭВМ необходимо преобразовывать десятичные числа в двоичные, а при выводе информации на индикаторы или печатающее устройство – двоичные или двоично-десятичные коды в коды управления знакогенератором, светодиодными или жидкокристаллическими индикационными панелями, механизмом печати.

Отправным пунктом для построения преобразователя кодов является таблица истинности, в которой записывается полный набор входных и соответствующий набор выходных слов. Если входные и выходные слова записаны двоичными символами, то синтез преобразователя кода сводится к нахождению для каждого разряда выходного слова булевой функции, устанавливающей связь данного разряда с входными наборами двоичных переменных. Нахождение такой связи и минимизация булевого выражения осуществляется с помощью карт Карно (диаграмм Вейча). На заключительном этапе, полученная функция преобразуется к виду, удобному для реализации в заданном (выбранном) элементарном базисе.

В табл. 1.47 приведены наиболее распространенные в цифровой схемотехнике двоичные коды. В обозначениях кодов 8421, 7421, 5421, 2421 указан десятичный вес p_i двоичной единицы x_i соответствующего разряда.

Таблица 1.47

Наиболее распространенные двоичные коды

Десятичное число N	Код 8421	Код 7421	Код 5421	Код Айкена 2421	Код Грея	Код «с избытком» N+3	Дополнение до 9 «9-N»	Дополнение до 10 «10-N»	Код Джонсона
0	0000	0000	0000	0000	0000	0011	1001	1010	00000
1	0001	0001	0001	0001	0001	0100	1000	1001	00001
2	0010	0010	0010	0010	0011	0101	0111	1000	00011
3	0011	0011	0011	0011	0010	0110	0110	0111	00111
4	0100	0100	0100	0100	0110	0111	0101	0110	01111
5	0101	0101	1000	1011	0111	1000	0100	0101	11111
6	0110	0110	1001	1100	0101	1001	0011	0100	11110
7	0111	1000	1010	1101	0100	1010	0010	0011	11100
8	1000	1001	1011	1110	1100	1011	0001	0010	11000
9	1001	1010	1100	1111	1101	1100	0000	0001	10000

Код Грея образован последовательностью двоичных чисел, в которой два любых соседних числа (первое и последнее число также считаются соседними) отличаются только одним разрядом. В коде Джонсона переход к последующему числу осуществляется последовательной заменой 0 на 1, начиная справа, а после установки во всех разрядах 1 – заменой 1 на 0. Коды $N+3$, $9-N$, $10-N$ используются в арифметических устройствах для выполнения операций сложения и вычитания двоично-десятичных чисел.

Пусть необходимо построить преобразователь двоичного кода 421 в трехразрядный код Грея. Запишем таблицу истинности 1.48.

Таблица 1.48

Таблица истинности кодов

Десятичное число N	Код 421 $x_2x_1x_0$	Код Грея $y_2y_1y_0$
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Каждый разряд y_i получаемого на выходе кода является независимой функцией входных наборов $x_2x_1x_0$, которую необходимо найти и минимизировать.

Характеристические уравнения в СДНФ будут иметь вид:

$$y_2 = x_2\bar{x}_1\bar{x}_0 + x_2\bar{x}_1x_0 + x_2x_1\bar{x}_0 + x_2x_1x_0,$$

$$y_1 = \bar{x}_2x_1\bar{x}_0 + \bar{x}_2x_1x_0 + x_2\bar{x}_1\bar{x}_0 + x_2\bar{x}_1x_0,$$

$$y_0 = \bar{x}_2\bar{x}_1x_0 + \bar{x}_2x_1\bar{x}_0 + x_2\bar{x}_1x_0 + x_2x_1\bar{x}_0.$$

Составим карту Карно для состояния y_2 на основе таблицы истинности 1.48.

Таблица 1.49

Карта Карно для y_2

	x_1x_0			
x_2	00	01	11	10
0	0	0	0	0
1	1	1	1	1

Минимизировав, получим

$$y_2 = x_2.$$

Аналогичным образом составим карты Карно для состояний y_1 и

y_0 .

Таблица 1.50

Карта Карно для y_1

	x_1x_0			
x_2	00	01	11	10
0	0	0	1	1
1	1	1	0	0

Минимизировав, получим

$$y_1 = x_2\bar{x}_1 + \bar{x}_2x_1.$$

Таблица 1.51

Карта Карно для y_0

	x_1x_0			
x_2	00	01	11	10
0	0	1	0	1
1	0	1	0	1

Минимизировав, получим

$$y_0 = \bar{x}_1x_0 + x_1\bar{x}_0.$$

Исходя из полученных уравнений, построим схему преобразователь двоичного кода 421 в трехразрядный код Грея (рис. 1.31).

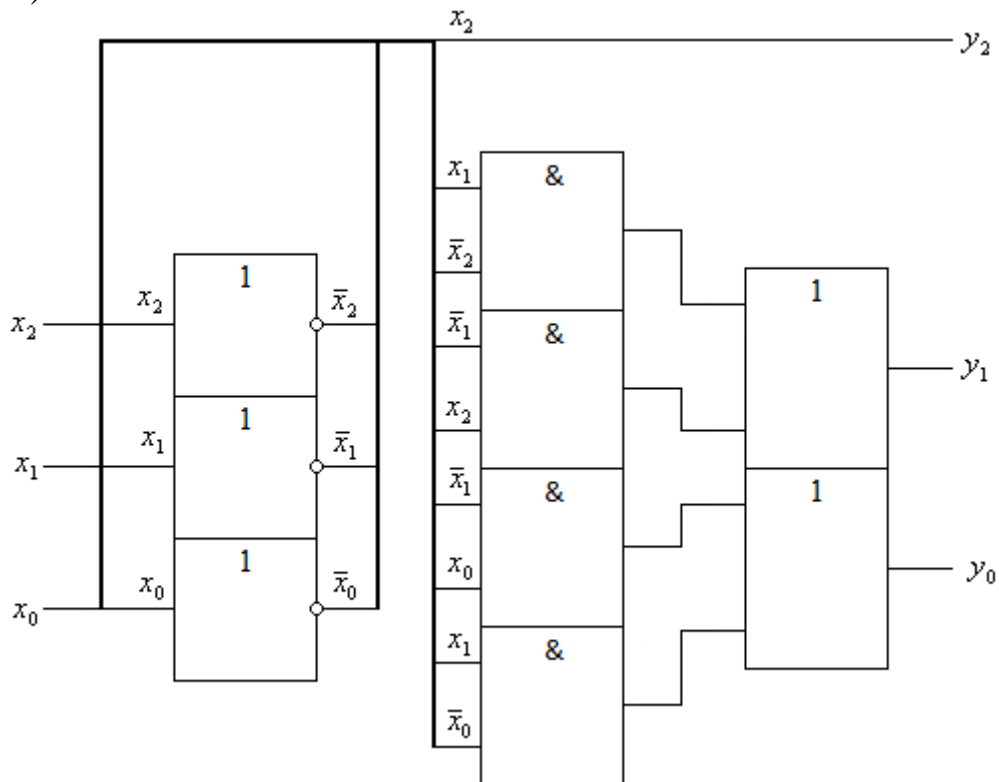


Рис. 1.31. Схема преобразователь двоичного кода 421 в трехразрядный код Грея

1.9 Индикаторы

Визуализация двоично-десятичных чисел часто производится с помощью семисегментных панелей на основе жидких кристаллов или светодиодов (рис. 1.32, а), широко используемых в микрокалькуляторах и т.д.



а)

Рис. 1.32. Семисегментные панели

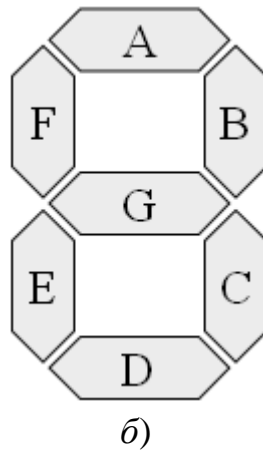


Рис. 1.32. Семисегментные панели

Если сегменты обозначены буквами, как показано на рис. 1.32, б, то таблица истинности будет иметь следующий вид (табл. 1.52).

Таблица 1.52

Таблица истинности семисегментного выходного кода

Десятичное число	Входной код 8421				Семисегментный выходной код						
	x_0	x_1	x_2	x_3	A	B	C	D	E	F	G
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Составим карты Карно для сегментов A , B , C , D , E , F , G на основе таблицы истинности 1.52.

Таблица 1.53

Карта Карно для А

	x_2x_3			
x_0x_1	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	*	*	*	*
10	1	1	*	*

Таблица 1.54

Карта Карно для В

	x_2x_3			
x_0x_1	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	*	*	*	*
10	1	1	*	*

Таблица 1.55

Карта Карно для С

	x_2x_3			
x_0x_1	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	*	*	*	*
10	1	1	*	*

Таблица 1.56

Карта Карно для D

	x_2x_3			
x_0x_1	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	*	*	*	*
10	1	1	*	*

Таблица 1.57

Карта Карно для E

	x_2x_3			
x_0x_1	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	*	*	*	*
10	1	0	*	*

Таблица 1.58

Карта Карно для F

	x_2x_3			
x_0x_1	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	*	*	*	*
10	1	1	*	*

Таблица 1.59

Карта Карно для G

x_0x_1	x_2x_3			
	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	*	*	*	*
10	1	1	*	*

Минимизировав, получим уравнения сегментов (табл. 1.60).

Таблица 1.60

Уравнения сегментов

Сегмент	Уравнение	Сегмент	Уравнение
A	$A = x_0 + x_2 + x_1x_3 + \bar{x}_1\bar{x}_3$	E	$E = \bar{x}_1\bar{x}_3 + x_2\bar{x}_3$
B	$B = x_0 + \bar{x}_1 + \bar{x}_2\bar{x}_3 + x_2x_3$	F	$F = x_0 + \bar{x}_2\bar{x}_3 + x_1\bar{x}_2 + x_1\bar{x}_3$
C	$C = x_0 + x_1 + \bar{x}_2 + x_3$	G	$G = x_0 + x_1\bar{x}_2 + x_2\bar{x}_3 + \bar{x}_1x_2$
D	$D = x_0 + x_2\bar{x}_3 + \bar{x}_1\bar{x}_3 + \bar{x}_1x_2 + x_1\bar{x}_2x_3$		

Схема преобразователя кода 8421 в код семисегментного индикатора показана на рис. 1.33.

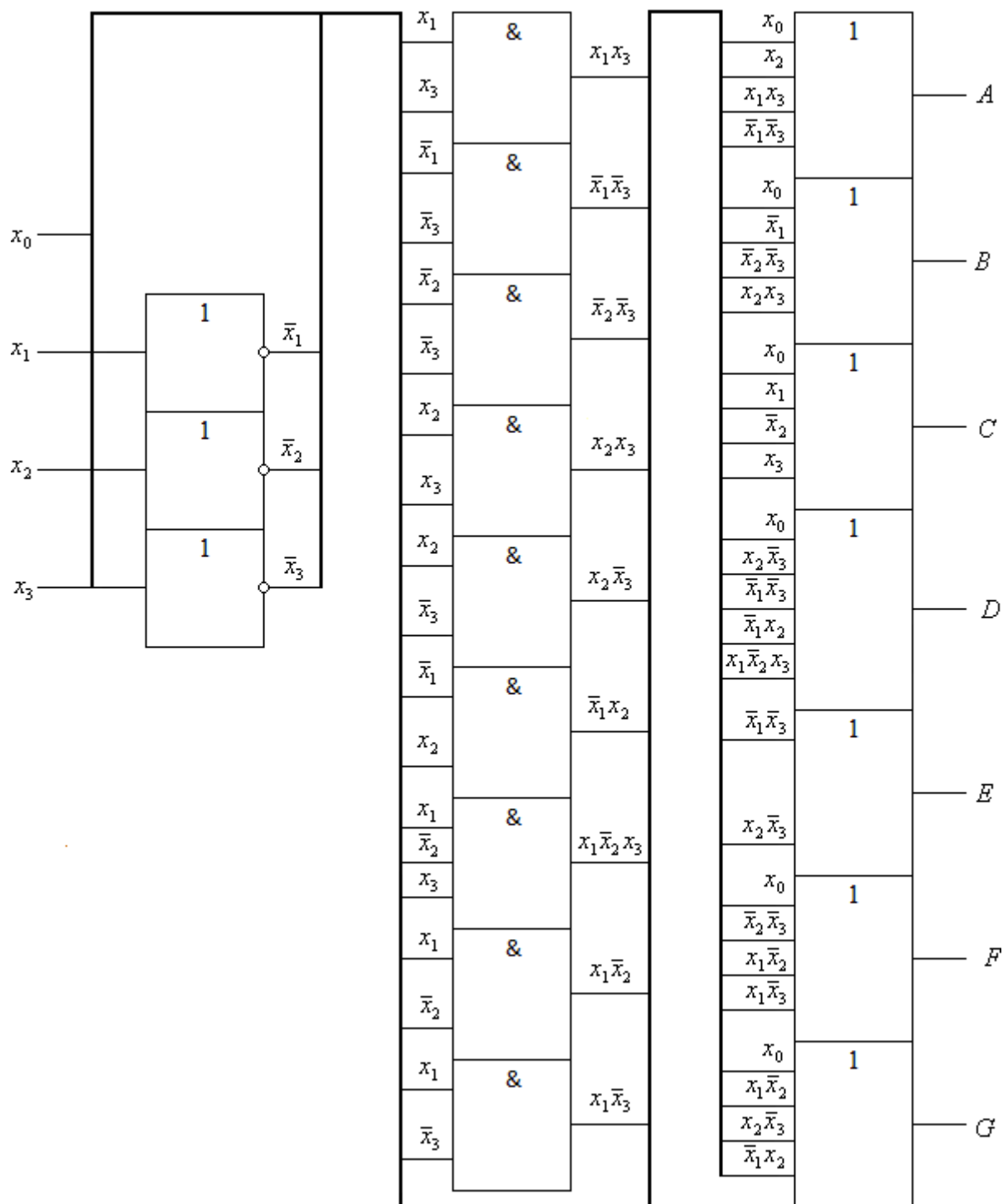


Рис. 1.33. Схема преобразователя кода 8421 в код семисегментного индикатора

ЗАКЛЮЧЕНИЕ

В настоящее время во всех сферах человеческой деятельности в той или иной степени используются достижения полупроводниковой электроники. Это, прежде всего, промышленное производство, в котором электроника выполняет очень важные функции – управления, координации работы всех технологических объектов. Непосредственное управление работой станков, роботов, автоматических транспортных и складских систем, технологических установок – все это выполняется современными средствами автоматизации технологических процессов, выполненных на базе полупроводниковых электронных элементов, устройств и целых систем.

В основе всех современных информационных технологий также лежат достижения полупроводниковой электроники: компьютерная техника, информационно-вычислительные сети, спутниковые каналы связи и многое другое. В информационных технологиях очень важную роль играют средства оптоэлектроники. Передача огромных объемов информации обеспечивается благодаря применению волоконно-оптических линий связи. Для взаимодействия человека и информационно-вычислительной системы всё шире применяются цветные жидкокристаллические мониторы, позволяющие не только отображать информацию, но и вводить её непосредственно с экрана с помощью пленочной клавиатуры, расположенной на поверхности экрана. На их основе выполняются ноутбуки, промышленные компьютеры и компьютеры специального назначения, конструктивно представляющие собой моноблок.

Основой элементной базы для разработки самых различных электронных устройств и систем становятся БИС/СБИС микропроцессорных комплектов и программируемые ИС, среди которых наиболее перспективными являются ПЛИС (система на кристалле). Совершенствуется технология БИС, повышая степень интеграции, тактовая частота и производительность СБИС. В области микропроцессорных средств продолжает действовать закон, сформулированный одним из основателей фирмы *Intel* Гордоном Муром еще в 1965 г.: «Мощность процессора удваивается каждые полтора года при сохранении его стоимости».

Возрастает роль аналого-цифровых и цифроаналоговых преобразователей, обеспечивающих взаимодействие вычислительной сети с объектом управления. Они могут выполняться как в виде отдельных БИС/СБИС, так и встроенным в СБИС по аналоговым входам и выходам. Такие СБИС получили название аналоговых микропроцессоров.

Список использованных источников

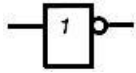
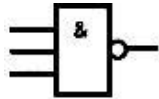
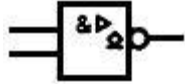
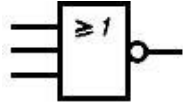
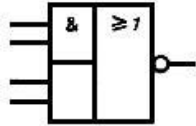
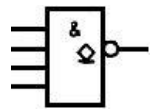
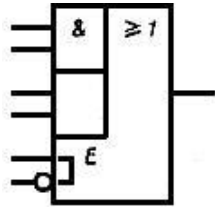
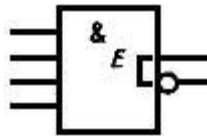

1. Алексенко, А.Г. Микросхемотехника / И.И. Шагурин. – М.: Радио и связь, 1990. – 396 с.
2. Гусев, В.Г. Электроника и микропроцессорная техника: учеб. для вузов / В.Г. Гусев, Ю.М. Гусев. – 5-е изд., стер. – М.: Высш. шк., 2008. – 798 с.
3. Ибрагим, К.Ф. Основы электронной техники. Схемы. Системы / К.Ф. Ибрагим. – М.: Мир. 2001.
4. Информационно-измерительная техника и электроника: учебник для студ. высш. учеб. заведений / [Г.Г. Раннев, В.А. Суругина, В.И. Калашников и др.]; под ред. Г.Г. Раннева. – 2-е изд., стер. – М.: Издательский центр «Академия», 2007. – 512 с.
5. Касаткин, А.С. Электротехника: учеб. для вузов / А.С. Касаткин, М.В. Немцов. – 9-е изд., стер. – М.: Издательский центр «Академия», 2005. – 544 с.
6. Кауфман, М., Сидман, А. Практическое руководство по расчетам схем в электронике. В 2-х томах: пер. с англ. / Ф.Н. Покровского. М.: Энергоатомиздат, 1991.
7. Лачин, В.И. Электроника / В.И. Лачин – М.: Ростов-на-Дону: Феникс; 2001.
8. Метрология и радиоизмерения: учебник для вузов / В.И. Нефедов [и др.]; под ред. В.И. Нефедова. – М.: Высш. шк., 2003. – 526 с.
9. Нарышкин, А.К. Цифровые устройства и микропроцессоры: учеб. пособие для студ. высш. учеб. заведений / А.К. Нарышкин. – М.: Издательский центр «Академия», 2006. – 320 с.
10. Наумкина, Л.Г. Цифровая схемотехника. Конспект лекций по дисциплине «Схемотехника» / Л.Г. Наумкина. – М.: Издательство «Горная книга», Издательство Московского государственного горного университета, 2008. – 308 с.
11. Немцов, М.В. Электротехника и электроника: учебник для вузов. – М.: Издательство МЭИ, 2003. – 593 с.
12. Опадчий, Ю.Ф. Аналоговая и цифровая электроника: учеб. для вузов / Ю.Ф. Опадчий, О.П. Глудкин, А.И. Гуров; под ред. О.П. Глудкина. – М.: Горячая линия – Телеком, 2005. – 768 с.
13. Потемкин, И.С. Функциональные узлы цифровой автоматики. – М.: Энергоатом издат. 1988. – 320 с.

14. Прянишников, В.А. Электроника: полный курс лекций / В.А. Прянишников. – 4-е изд. – СПб.: Корона принт, 2006. – 416 с.
15. Раннев, Г.Г. Методы и средства измерений: учебник для студ. высш. учеб. заведений / Г.Г. Раннев, А.П. Тарасенко. – 5-е изд., стер. – М. : Издательский центр «Академия», 2008. – 336 с.
16. Соклоф, С. Аналоговые интегральные схемы: пер. с англ. / С. Соклоф – М.: Мир, 1988.
17. Справочное пособие по основам электротехники и электроники / под ред. А.В. Нетушила. М.: Высшая школа, 1987.
18. Токхейм, Р. Основы цифровой электроники: пер. с англ. / Р. Токхейм – М.: Мир. 1988. – 392 с.
19. Хоровиц, П. Искусство схемотехники/ П. Хоровиц, У. Хилл. – Издание 5-е, переработанное. – М.: Мир, 1998.
20. Электротехника и основы электроники / под ред. О.П. Глудкина, Б.П. Соколова, М.: Высшая школа, 1993.
21. Электротехника и основы электроники / О.А. Антонова, О.П. Глудкин, П.Д. Давидов и др.; под ред. О.П. Глудкина, В.П. Соколова. – М.: Высшая школа, 1993. – 445 с.
22. Электротехника и электроника. В 3-х книгах / под ред. В.Г. Герасимова. – М.: Энергоатомиздат, 1997.

ПРИЛОЖЕНИЕ

ГОСТ 2.743-91 Единая система конструкторской документации (ЕСКД). Обозначения условные графические в схемах. Элементы цифровой техники

Примеры УГО логических элементов

Наименование	Обозначение
1. Элемент «НЕТ»	
2. Элемент 3И-НЕ	
3. Элемент 2И-НЕ с открытым коллекторным выходом и повышенной нагрузочной способностью	
4. Элемент 3ИЛИ-НЕТ	
5. Комбинированный элемент 2И-ИЛИ с инвертированным выходом	
6. Элемент 4И-НЕТ с открытым коллектором на выходе	
7. Элемент 2И-ИЛИ с инвертированным выходом и расширительным входом	
8. Расширитель	
9. Элемент проверки четности или нечетности	

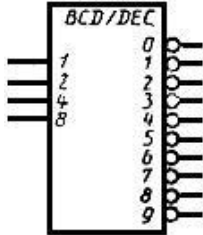
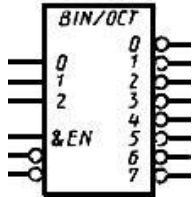
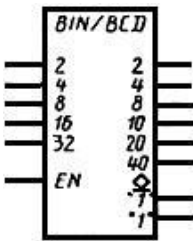
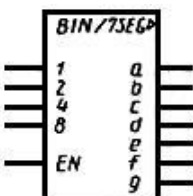
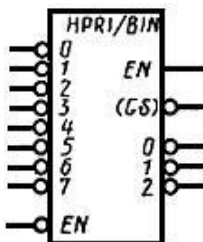
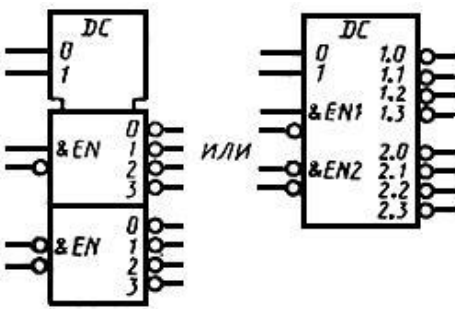
Примеры УГО приемопередающих элементов

Наименование	Обозначение
1. Четыре шинных усилителя с двухпороговым входом и выходом на три состояния с общим входом разрешения третьего состояния	
2. Двухнаправленный шинных приемопередатчик восьмиканальный	
3. Шестиканальный буферный элемент с тремя состояниями на выходе с сигналом разрешения по выходу	

Примеры УГО гистерезисных элементов

Наименование	Обозначение
1. Инвертирующий усилитель с порогом Шмитта	
2. Триггер Шмитта с логическим элементом 4И на входе	

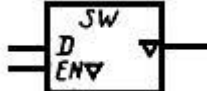
Примеры УГО преобразователей (дешифраторов) и кодирующих устройств (шифраторов)

Наименование	Обозначение
1. Преобразователь двоично-десятичного кода в десятичный код	
2. Преобразователь с трех линий на восемь	
3. Преобразователь двоичного кода в двоично-десятичный	
4. Преобразователь-усилитель двоичного кода в семисегментный. Примечание. Допускается заменить строчные буквы прописными: <i>A, B, C, D, E, F, G</i>	
5. Кодирующее устройство приоритета (приоритетный шифратор) с 8 линий на 3 линии (<i>GS</i> – «групповой сигнал»)	
6. Два дешифратора, принимающих двухразрядный код. Примечание. Допускается обозначение дешифраторов <i>A</i> и <i>B</i> , которые изображаются в качестве групповой метки выходов соответствующего дешифратора	

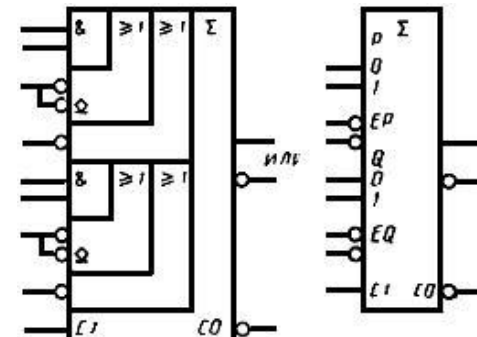
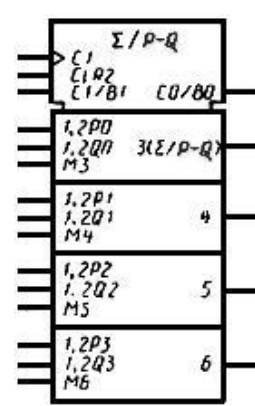
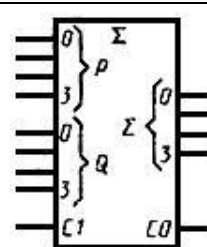
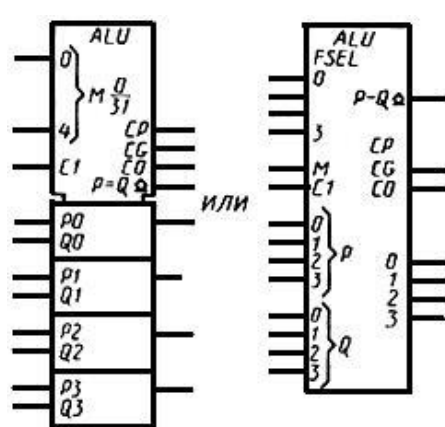
Наименование	Обозначение
7. Преобразователь уровней <i>TTL</i> в уровни <i>МОП</i>	
8. Преобразователь уровней <i>ЭСЛ</i> в уровни <i>TTL</i>	

Примеры УГО мультиплексов и демультиплексов, а также коммутаторов цифровых и аналоговых сигналов

Наименование	Обозначение
1. Мультиплексор на 8 входов со стробированием Примечание. Вход стробирования <i>EN</i> допускается обозначать <i>STR</i>	
2. Демультиплексор на 8 линий	
3. Мультиплексор четырехканальный по два входа каждый	
4. Мультиплексор двухканальный по 4 входа каждый.	

Наименование	Обозначение
5. Электронный коммутатор	

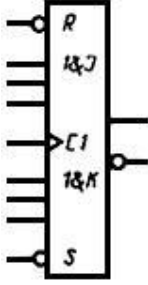
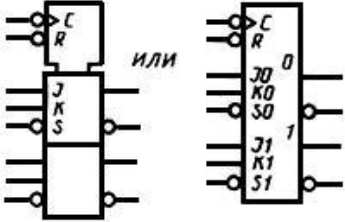
Примеры УГО арифметических элементов

Наименование	Обозначение
1. Полный одноразрядный сумматор	
2. Четырехразрядный сумматор-вычитатель	
3. Полный сумматор на 4 бита	
4. Четырехразрядное скоростное АЛУ	

Наименование	Обозначение
5. Генератор ускоренного переноса для АЛУ	
6. Четырехразрядный цифровой компаратор	

Примеры УГО триггеров (бистабильных элементов)

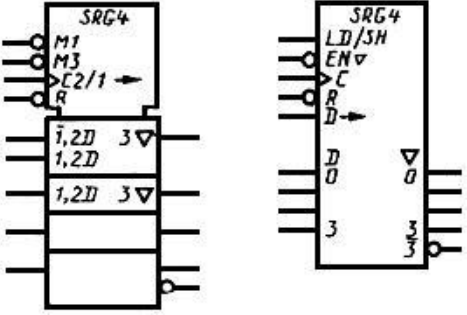
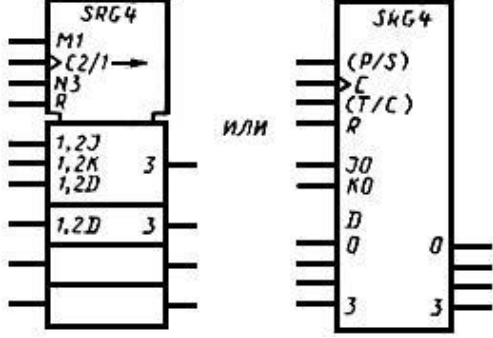
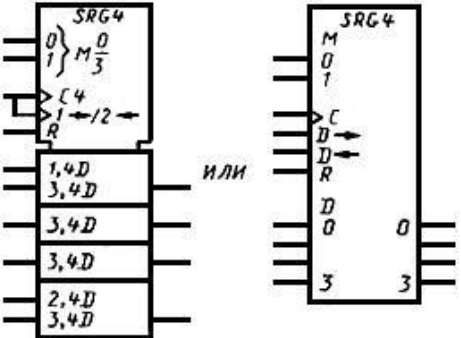
Наименование	Обозначение
1. Два триггера с отдельным запуском (<i>RS</i> -типа), один с дополнительным входом	
2. Два триггера задержки <i>D</i> -типа	
3. Шесть <i>D</i> -триггеров с общими входами управления и сброса	
4. Триггер <i>D</i> -типа, запускаемый по фронту	
5. Триггер <i>JK</i> -типа, запускаемый по фронту	

Наименование	Обозначение
6. Универсальный <i>JK</i> -триггер со структурой «мастер-помощник»	
7. Два <i>JK</i> -триггера с общими входами управления и сброса	

Примеры УГО моностабильных (мультивибраторов) и нестабильных элементов

Наименование	Обозначение
1. Ждущий мультивибратор с перезапуском	
<p>2. Два генератора, управляемых напряжением, с указанием выводов питания</p> <p>(<i>LOG</i> – питание цифровой части элемента, <i>OSC</i> – питание аналоговой части элемента, <i>FC</i> – управление частотой)</p>	

Примеры УГО регистров и счетчиков

Наименование	Обозначение
<p>1. Сдвиговый 4-разрядный регистр с параллельными входами</p>	
<p>2. Сдвиговый 4-разрядный последовательно-параллельный регистр с прямым и дополнительным кодом на выходе (T/C – вход переключения кода на выходах: прямой или дополнительный; P/S вход, управляющий соединением разрядов регистров последовательно или параллельно)</p>	
<p>3. Сдвиговый 4-разрядный двунаправленный универсальный регистр</p>	

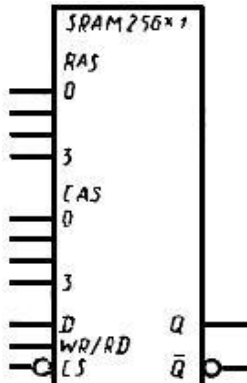
Наименование	Обозначение
<p>4. Универсальный 8-разрядный регистр</p>	
<p>5. Сдвиговый 8-разрядный регистр с двойным последовательным входом и параллельными выходами</p>	
<p>6. Сдвиговый 8-разрядный универсальный регистр с последовательным и параллельными входами и выходами (A/S – вход переключения режимов: асинхронного или синхронного; ALD – вход разрешения параллельной записи информации в канал A)</p>	

Наименование	Обозначение
7. Сдвиговый 8-разрядный регистр с параллельной загрузкой	
8. Двоичный 14-разрядный счетчик со сквозным переносом	
9. Асинхронный десятичный счетчик, состоящий из делителей на 2 и на 5 с параллельной записью	
10. 4-разрядный асинхронный десятичный счетчик, состоящий из делителей на 2 и на 5 с предварительной установкой и синхронным сбросом	

Наименование	Обозначение
11. Синхронный десятичный счетчик с параллельной загрузкой	<p>Two schematic diagrams for a CTRDIV10 synchronous decade counter with parallel load. The left diagram shows inputs M1, 1,3,4+, C2, G3, G4, R and outputs 1,2,D. The right diagram shows inputs LD, T, &CE, R and outputs D0, D1, D2, D3, CT=9.</p>
12. Десятичный синхронный реверсивный счетчик	<p>Two schematic diagrams for a CTRDIV10 synchronous decade reversible counter. The left diagram shows inputs R, 2+, G1, 1-, G2, C3 and outputs iCT=9, 2CT=0, 3,D. The right diagram shows inputs R, +, -, LD and outputs CT=9, CT=0, D0, D1, D2, D3, CT=1,2,3.</p>

Примеры УГО запоминающих устройств (ЗУ)

Наименование	Обозначение
1. ПЗУ на 32 слова по 8 битов	
2. ОЗУ с произвольной выборкой на 16 слов по 4 бита Примечание. Обозначения выводов:	<p style="text-align: center;">ИЛИ</p>
3. ОЗУ с произвольной выборкой на 4 слова по 4 бита с отдельной адресацией при записи и при считывании	<p style="text-align: center;">ИЛИ</p>
4. Статическое ОЗУ на 4 слова по 4 бита	

Наименование	Обозначение
5. Статическое ОЗУ на 256 слов по 1 биту	

ДЛЯ ЗАМЕТОК

Учебное издание

**ФЕДОРОВ Сергей Витальевич
ФАХРУЛЛИНА Альмира Раисовна**

**ЦИФРОВЫЕ ЭЛЕКТРОННЫЕ УСТРОЙСТВА
СИСТЕМ УПРАВЛЕНИЯ**

Учебное пособие

*За достоверность информации, изложенной в учебном пособии,
ответственность несут авторы*

Подписано в печать 10.06.2024 г. Формат 60х84/16.
Усл. печ. л. 5,29. Уч.-изд. л. 5,52.
Тираж 300 экз. (1-й завод 35 экз.). Изд. № 8. Заказ 21.

*Редакционно-издательский центр
Уфимского университета науки и технол.
450008, Башкортостан, г. Уфа, ул. Карла Мар.*

*Отпечатано в отделе полиграфии
редакционно-издательского центра
Уфимского университета науки и технологий
450008, Башкортостан, г. Уфа, ул. Карла Маркса, 12.*