

ГЛАВА 1 ОСНОВЫ ИНФОРМАЦИОННОЙ ЭЛЕКТРОНИКИ

1.1 Основы алгебры логики. Логические выражения. Логические элементы

Законы алгебры логики

Законы алгебры логики базируются на аксиомах и позволяют преобразовывать логические функции. Логические функции преобразуются с целью их упрощения, а это ведет к упрощению цифровой схемы.

АКСИОМЫ алгебры логики описывают действие логических функций "И" и "ИЛИ" и записываются следующими выражениями:

$$0 * 0 = 0 \quad 0 + 0 = 0$$

$$0 * 1 = 0 \quad 0 + 1 = 1$$

$$1 * 0 = 0 \quad 1 + 0 = 1$$

$$1 * 1 = 1 \quad 1 + 1 = 1$$

Всего имеется пять законов алгебры логики:

Закон одинарных элементов

$$1 * X = X \quad 0 * X = 0 \quad 1 + X = 1 \quad 0 + X = X$$

Этот закон непосредственно следует из приведённых выше выражений аксиом алгебры логики.

Верхние два выражения могут быть полезны при построении коммутаторов, ведь подавая на один из входов элемента "И" логический ноль или единицу можно либо пропускать сигнал на выход, либо формировать на выходе нулевой потенциал.

Второй вариант использования этих выражений заключается в возможности избирательного обнуления определённых разрядов многоразрядного числа. При поразрядном применении операции "И" можно либо оставлять прежнее значение разряда, либо обнулять его, подавая на соответствующие разряды единичный или нулевой потенциал. Например, требуется обнулить 6, 3 и 1 разряды. Тогда:

$$\begin{array}{r} 11010111 \\ \wedge 10110101 \\ \hline 10010101 \end{array}$$

В приведённом примере отчётливо видно, что для обнуления необходимых разрядов в маске (нижнее число) на месте соответствующих разрядов записаны нули, в остальных разрядах записаны единицы. В исходном числе (верхнее число) на месте 6 и 1 разрядов находятся единицы. После выполнения операции "И" на этих местах появляются нули. На месте третьего разряда в исходном числе находится ноль. В результирующем числе на этом месте тоже присутствует ноль. Остальные разряды, как и требовалось по условию задачи, не изменены.

Точно так же можно записывать единицы в нужные нам разряды. В этом случае необходимо воспользоваться нижними двумя выражениями закона одинарных элементов.

При поразрядном применении операции "ИЛИ" можно либо оставлять прежнее значение разряда, либо обнулять его, подавая на соответствующие разряды нулевой или единичный потенциал. Пусть требуется записать единицы в 7 и 6 биты числа. Тогда:

$$\begin{array}{r} 01010111 \\ \vee \\ 11000000 \\ \hline 11010111 \end{array}$$

Здесь в маску (нижнее число) мы записали единицы в седьмой и шестой биты. Остальные биты содержат нули, и, следовательно, не могут изменить первоначальное состояние исходного числа, что мы и видим в результирующем числе под чертой.

Первое и последнее выражения позволяют использовать логические элементы с большим количеством входов в качестве логических элементов с меньшим количеством входов. Для этого неиспользуемые входы в схеме "И" должны быть подключены к источнику питания, как это показано на рисунке 1:

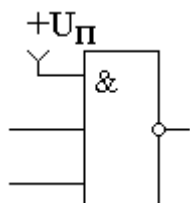


Рис.1 Схема "2И-НЕ", реализованная на логическом элементе "3И-НЕ"

а неиспользуемые входы в схеме "ИЛИ" должны быть подключены к общему проводу схемы, как это показано на рисунке 2.

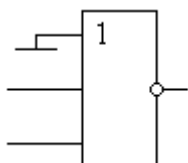


Рис. 2 Схема "НЕ", реализованная на элементе "2И-НЕ"

Законы отрицания

а. Закон дополнительных элементов

$$x + \bar{x} = 1 \quad x \cdot \bar{x} = 0$$

Выражения этого закона широко используется для минимизации логических схем. Если удаётся выделить из общего выражения логической функции такие подвыражения, то можно сократить необходимое количество входов элементов цифровой схемы, а иногда и вообще свести всё выражение к логической константе.

б. Двойное отрицание

$$\bar{\bar{1}} = 0 \quad \bar{\bar{0}} = 1 \quad \bar{\bar{1}} = 1 \quad \bar{\bar{x}} = x$$

с. Закон отрицательной логики

$$\overline{\overline{a + b + c}} = a \cdot b \cdot c$$

$$\overline{\overline{a} \cdot \overline{b} \cdot \overline{c}} = a + b + c$$

Закон отрицательной логики справедлив для любого числа переменных. Этот закон позволяет реализовывать логическую функцию "И" при помощи логических элементов "ИЛИ" и наоборот: реализовывать логическую функцию "ИЛИ" при помощи логических элементов "И". Это особенно полезно в ТТЛ схемотехнике, так как там легко реализовать логические элементы "И", но при этом достаточно сложно логические элементы "ИЛИ". Благодаря закону отрицательной логики можно реализовывать элементы "ИЛИ" на логических элементах "И". На рисунке 3 показана реализация логического элемента "2ИЛИ" на элементе "2И-НЕ" и двух инверторах.

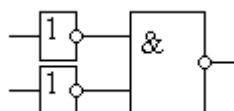


Рис.3 Логический элемент "2ИЛИ", реализованный на элементе "2И-НЕ" и двух инверторах

То же самое можно сказать и о схеме монтажного "ИЛИ". В случае необходимости его можно превратить в монтажное "И", применив инверторы на входе и выходе этой схемы.

Комбинационные законы

Комбинационные законы алгебры логики во многом соответствуют комбинационным законам обычной алгебры, но есть и отличия.

а. закон тавтологии (многократное повторение)

$$X + X + X + X = X$$

$$X * X * X * X = X$$

Этот закон позволяет использовать логические элементы с большим количеством входов в качестве логических элементов с меньшим количеством входов. Например, можно реализовать двухвходовую схему "2И" на логическом элементе "3И", как это показано на рисунке 4:

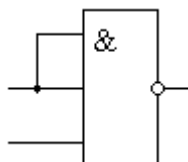


Рис.4 Схема "2И-НЕ", реализованная на логическом элементе "3И-НЕ"

или использовать схему "2И-НЕ" в качестве обычного инвертора, как это показано на рисунке 5:

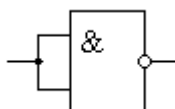


Рисунок 5. Схема "НЕ", реализованная на логическом элементе "2И-НЕ"

Однако следует предупредить, что объединение нескольких входов увеличивает входные токи логического элемента и его ёмкость, что увеличивает ток потребления предыдущих элементов и отрицательно сказывается на быстродействии цифровой схемы в целом.

Для уменьшения числа входов в логическом элементе лучше воспользоваться законом одинарных элементов, как это было показано выше.

б. закон переместительности

$$A + B + C + D = A + C + B + D$$

с. закон сочетательности

$$A + B + C + D = A + (B + C) + D = A + B + (C + D)$$

д. закон распределительности

$$X1(X2 + X3) = X1X2 + X1X3X1 + X2X3 = (X1 + X2)(X1 + X3) = \text{/докажем это путём раскрытия скобок/} = X1X1 + X1X3 + X1X2 + X2X3 = X1(1 + X3 + X2) + X2X3 = X1 + X2X3$$

Правило поглощения (одна переменная поглощает другие)

$$X1 + X1X2X3 = X1(1 + X2X3) = X1$$

Правило склеивания (выполняется только по одной переменной)

$$A\bar{B}C + AB\bar{C} = AC(\bar{B} + B) = AC$$

1

Также как в обычной математике в алгебре логики имеется старшинство операций.

При этом первым выполняется:

1. Действие в скобках
2. Операция с одним операндом (одноместная операция) — "НЕ"
3. Конъюнкция — "И"
4. Дизъюнкция — "ИЛИ"
5. Сумма по модулю два.

Операции одного ранга выполняются слева направо в порядке написания логического выражения. Алгебра логики линейна и для неё справедлив принцип суперпозиции.

Логические элементы

- ИЛИ – логическое сложение (**дизъюнкция**) – **OR**;
- И – логическое умножение (**конъюнкция**) – **AND**;
- НЕ – логическое отрицание (**инверсия**) – **NOT**.

Логический элемент И.

Вход X1	Вход X2	Выход Y
---------	---------	---------

0	0	0
1	0	0
0	1	0
1	1	1

На принципиальных схемах логический элемент "И" обозначают так.

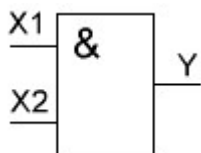


Рис.7

На зарубежных схемах обозначение элемента "И" имеет другое начертание. Его кратко называют **AND**.



Рис.8

Логический элемент ИЛИ.

Вход X1	Вход X2	Выход Y
0	0	0
1	0	1
0	1	1
1	1	1

На схемах элемент "ИЛИ" изображают так.

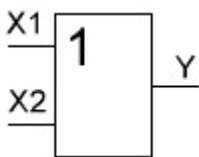


Рис.9

На зарубежных схемах его изображают чуть по-другому и называют элементом **OR**.



Рис.10

Логический элемент НЕ.

Вход X	Выход Y
0	1
1	0

Вот таким образом его показывают на схемах.

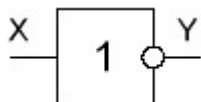


Рис.11

В зарубежной документации элемент "НЕ" изображают следующим образом. Сокращённо называют его **NOT**.

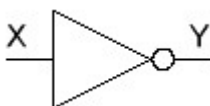


Рис.12

Все эти элементы в интегральных микросхемах могут объединяться в различных сочетаниях. Это элементы: И–НЕ, ИЛИ–НЕ, и более сложные конфигурации.

Логический элемент "исключающее ИЛИ".

К числу базовых логических элементов принято относить элемент реализующий функцию «исключающее ИЛИ». Иначе эта функция называется «неравнозначность».

Высокий потенциал на выходе возникает только в том случае, если входные сигналы не равны. То есть на одном из входов должна быть единица, а на другом ноль. Если на выходе логического элемента имеется инвертор, то функция выполняется противоположная – «равнозначность». Высокий потенциал на выходе будет появляться при одинаковых сигналах на обоих входах.

Таблица истинности.

Вход X1	Вход X2	Выход Y
0	0	0
1	0	1
0	1	1
1	1	0

Эти логические элементы находят своё применение в сумматорах. «Исключающее ИЛИ» изображается на схемах знаком равенства перед единицей "=1".

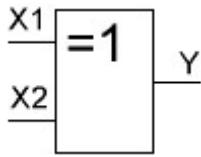


Рис.13

На зарубежный манер "исключающее ИЛИ" называют **XOR** и на схемах рисуют вот так.



Рис.14

Логические операции реализуются с использованием электрических схем, которые называются логическими элементами. Они изготавливаются в виде интегральных микросхем в большинстве своем на базе диодов и транзисторов, либо только транзисторов. В связи с этим различают два типа логических элементов: ДТЛ (диодно-транзисторная логика) и ТТЛ (транзисторно-транзисторная логика). На рис. 15 представлены схемные обозначения элементов, выполняющих простейшие логические операции. При этом элементы "ИЛИ" и "И" – двухвходовые, т.е. выполняющие операции с двумя входными переменными. Однако промышленностью выпускаются логические элементы "ИЛИ" и "И" с большим числом входных клемм.

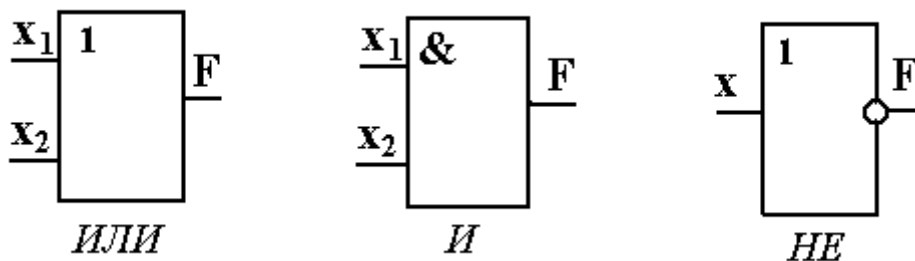


Рис. 15 Схемное обозначение логических элементов

На рис. 15 приведена схема на диодах, с помощью которой реализуется операция логического сложения. Действительно, при отсутствии на входах этой схемы сигналов ($x_i = 0$, где $i = 1, 2, \dots, n$) все диоды будут находиться в закрытом состоянии и на резисторе падение напряжения равно нулю ($F = 0$). При подаче хотя бы на один вход схемы сигнала положительной полярности диод, установленный на этом входе, перейдет в открытое состояние. Через него и последовательно включенный с ним резистор R будет протекать ток, падение напряжения на резисторе будет отличаться от нуля. $F = 1$.

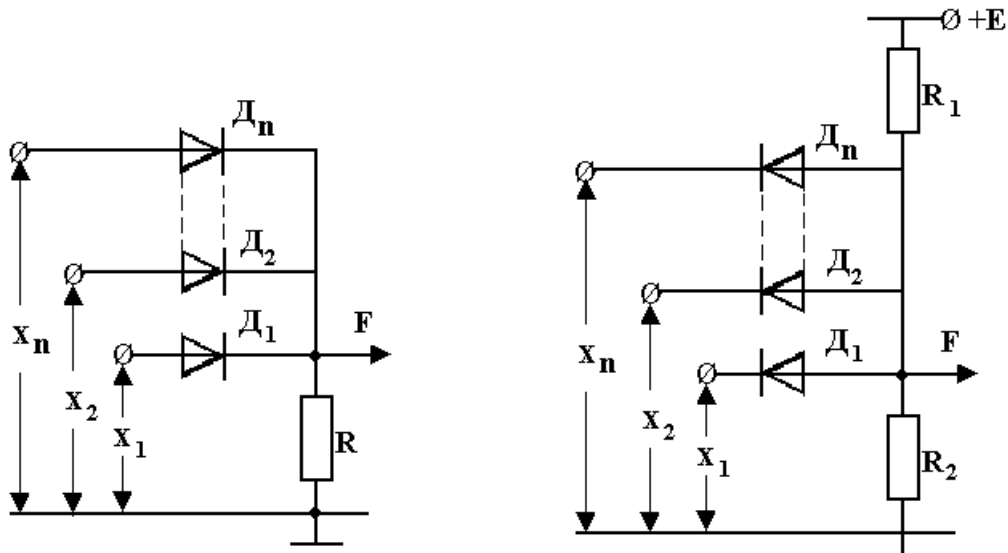


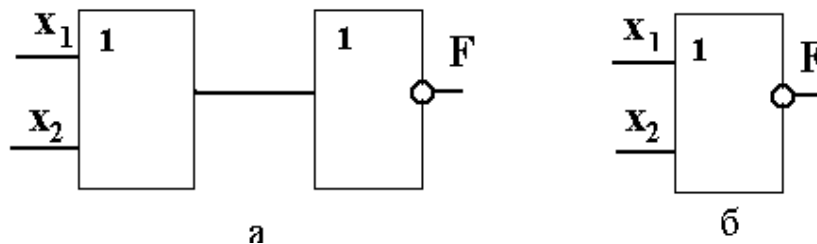
Рис. 15 Реализация логических операций И , ИЛИ

Через резистор R_2 будет протекать ток, и с этого резистора будет сниматься напряжение, $= ER_2 / (R_1 + R_2)$, т.е. $F = 1$, когда катоды всех диодов будут под положительным потенциалом, превышающим величину $U_{\text{вых}} (x_i = 1)$, а, следовательно, все диоды будут закрыты. Если хотя бы на одном входе схемы сигнал отсутствует, соответствующий диод будет открыт, и через него будет протекать ток в обход резистора R_2 , поскольку его сопротивление выбирается много больше сопротивления открытого диода. Падение напряжения на резисторе R_2 в данном случае будет равно нулю. $F = 0$.

Логические элементы выпускаются промышленностью в виде серий. Элементы каждой серии, выполняющие различные операции, могут сопрягаться в сложную схему логического устройства. Элементы серии создаются на базе основного элемента, который обычно выполняет операции “ИЛИ-НЕ” или “И-НЕ”. Математическая форма представления этих операций имеет вид:

$$F = \overline{x_1 + x_2 + x_3 \dots + x_n} \quad \text{и} \quad F = \overline{x_1 \cdot x_2 \cdot x_3 \dots x_n}$$

Схемы, реализующие операции “ИЛИ-НЕ” и “И-НЕ”, могут быть получены при последовательном соединении двух схем, выполняющих операции соответственно “ИЛИ”-“НЕ” и “И”-“НЕ”, что показано на рис.16 и 17,а для случая двухвходовых



элементов.

а

б

Рис. 16 Логическая операция «ИЛИ-НЕ»:

а – принцип реализации операции,

б - схемное обозначение логического элемента.

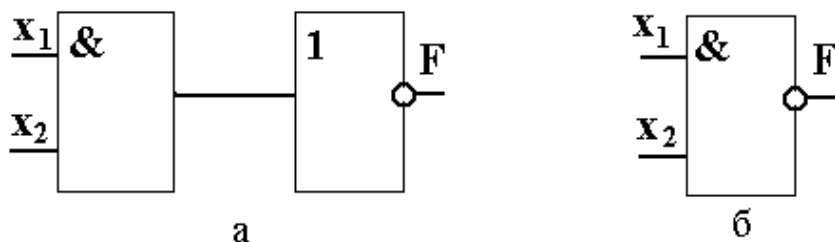


Рис. 17 Логическая операция «И-НЕ»:

а – принцип реализации операции,

б - схемное обозначение логического элемента.

Синтез цифровых комбинационных схем по произвольной таблице истинности.

Любая цифровая комбинационная схема (логическая схема без памяти) полностью описывается таблицей истинности. При этом необязательно, чтобы все комбинации входных цифровых сигналов были полезными. Возможна ситуация, когда только часть комбинаций входных логических сигналов является полезной. В этом случае выходные сигналы цифрового устройства для оставшихся комбинаций входных логических сигналов могут быть доопределены произвольно. Обычно при этом стараются выбирать цифровые значения выходных сигналов таким образом, чтобы схема цифрового устройства получилась простейшей.

Для реализации цифровых логических схем с произвольной таблицей истинности используется сочетание простейших логических элементов "И" "ИЛИ" "НЕ". Существует два способа синтеза цифровых схем, реализующих произвольную таблицу истинности. Это СКНФ (логическое произведение суммы входных сигналов) и СДНФ (сумма логических произведений входных сигналов).

При синтезе цифровой схемы, реализующей произвольную таблицу истинности, каждый выход анализируется (и строится схема) отдельно и независимо.

Совершенная дизъюнктивная нормальная форма (СДНФ)

Для реализации таблицы истинности при помощи логических элементов "И" (СДНФ) достаточно рассмотреть только те строки таблицы истинности, которые содержат логические "1" в выходном сигнале. Строки, содержащие в выходном сигнале логический 0 в построении цифровой схемы не участвуют. Каждая строка, содержащая в выходном сигнале логическую "1", реализуется схемой логического элемента "И" с количеством входов, совпадающим с количеством входных сигналов в таблице истинности.

Входы				Выходы	
In0	In1	In2	In3	Out0	Out1
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	1

Рис 18. Произвольная таблица истинности.

Для синтеза цифровой схемы, реализующей сигнал Out0, достаточно рассмотреть строки, выделенные красным цветом. В таблице истинности три строки, содержащие единицу в выходном сигнале Out0, поэтому в формуле СДНФ будет содержаться три произведения входных сигналов — **термов**, а точнее минтермов:

$$Out0 = \overline{In0} \cdot \overline{In1} \cdot \overline{In2} \cdot In3 + \overline{In0} \cdot In1 \cdot \overline{In2} \cdot \overline{In3} + \overline{In0} \cdot In1 \cdot In2 \cdot In3$$

Так как количество переменных в каждом терме (ранг терма) данного логического выражения равно, то такое логическое выражение называется совершенным. (Совершенная Дизъюнктивная Нормальная Форма — СДНФ)

Полученное логическое выражение реализуются микросхемой D2 в схеме, приведенной на рисунке 19.

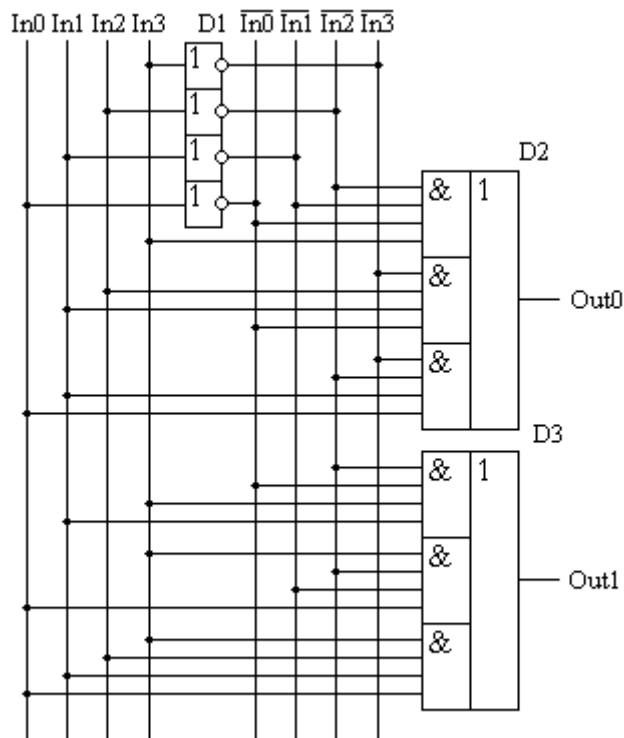


Рис.19 Принципиальная схема

Совершенная конъюнктивная нормальная форма (СКНФ)

Еще одним способом реализации цифровых комбинационных схем является запись логического выражения в совершенной конъюнктивной нормальной форме (СКНФ). Применение СКНФ оправдано при большом количестве логических единиц в выходном сигнале проектируемой цифровой схемы, как это показано в качестве примера в таблице истинности .

Таблица Пример таблицы истинности

№ комбинации	Входы				Выходы	
	8	4	2	1	a	b
0	0	0	0	0	1	1
1	0	0	0	1	0	1
2	0	0	1	0	1	1
3	0	0	1	1	1	1
4	0	1	0	0	0	1
5	0	1	0	1	1	0
6	0	1	1	0	1	0
7	0	1	1	1	1	1
8	1	0	0	0	1	1
9	1	0	0	1	1	1

Для реализации цифрового комбинационного устройства по таблице истинности при помощи логических элементов "ИЛИ" (СКНФ) достаточно рассмотреть только те строки таблицы истинности, которые содержат логические "0" в выходном сигнале.

$$a = (8 + 4 + 2 + \bar{1}) \cdot (8 + \bar{4} + 2 + 1)$$

Полученная формула в схеме на рисунке 20 реализуется микросхемой D2.

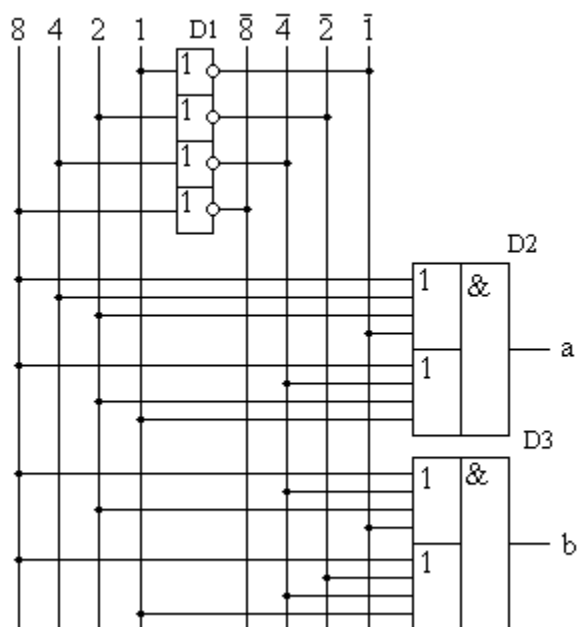


Рис.20 Принципиальная схема мультиплексора, выполненная на логических элементах

ГЛАВА 2 ЦИФРОВЫЕ УСТРОЙСТВА

Триггеры (RS, T, D, JK), регистры, счетчики (суммирующие, вычитающие, реверсивные, кольцевые). Сумматоры (одноразрядные и многоразрядные), шифраторы, дешифраторы, мультиплексоры, демультиплексоры, цифровые компараторы, преобразователи кодов, индикаторы.

2.1 Триггеры

Простейшая схема, позволяющая запоминать двоичную информацию, может быть построена на двух инверторах, охваченных положительной обратной связью. Эта схема приведена на рисунке 21.

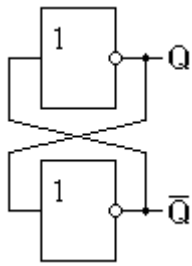


Рис.21 Схема простейшего триггера, построенного на инверторах

В этой схеме может быть только два состояния - на выходе Q присутствует логическая единица и на выходе Q присутствует логический ноль. Если логическая единица присутствует на выходе Q, то на инверсном выходе будет присутствовать логический ноль, который после очередного инвертирования подтверждает уровень логической единицы на выходе Q. И наоборот, если на выходе Q присутствует логический ноль, то на инверсном выходе будет присутствовать логическая единица.

RS триггер

RS триггер получил название по названию своих входов. Вход S (Set — установить *англ.*) позволяет устанавливать выход триггера Q в единичное состояние. Вход R (Reset — сбросить *англ.*) позволяет сбрасывать выход триггера Q (Quit — выход *англ.*) в нулевое состояние.

Для реализации RS триггера воспользуемся логическими элементами "2И-НЕ". Его принципиальная схема приведена на рисунке 22.

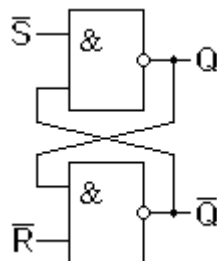


Рис.22 Схема простейшего rs триггера на схемах "2И-НЕ". Входы R и S инверсные (активный уровень '0')

Составим таблицу истинности RS триггера. Входы R и S в этой таблице будем использовать прямые, то есть и запись нуля, и запись единицы будут осуществляться единичными потенциалами (таблица 15).

Таблица 15. Таблица истинности RS триггера.

RS	Q(t)	Q(t+1)	Пояснения
0 0	0	0	Режим хранения информации R=S=0
0 0	1	1	
0 1	0	1	Режим установки единицы S=1

0	1	1	1	
1	0	0	0	Режим записи нуля R=1
1	0	1	0	
1	1	0	*	R=S=1 запрещенная комбинация
1	1	1	*	

RS триггер можно построить и на логических элементах "ИЛИ". Схема RS триггера, построенного на логических элементах "ИЛИ" приведена на рисунке 23. Принцип работы RS триггера, собранный по этой схеме будет точно таким же, как и рассмотренный ранее. Единственное отличие в работе этой схемы по сравнению с предыдущей схемой RS триггера будет заключаться в том, что сброс и установка триггера будет производиться единичными логическими уровнями. Эти особенности связаны с принципами работы инверсной логики, которые рассматривались ранее.

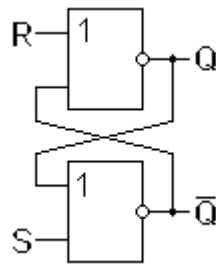


Рис.23 Схема простейшего RS триггера на схемах "ИЛИ". Входы R и S прямые (активный уровень '1')

Так как RS триггер при построении его на логических элементах "И" и "ИЛИ" работает одинаково (принцип работы RS триггера от схемы не зависит), то его условно-графическое изображение на принципиальных схемах тоже одинаково. Условно-графическое изображение RS триггера на принципиальных схемах приведено на рисунке 24.

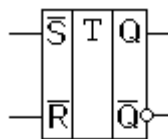


Рис.24 Условно-графическое обозначение RS триггера

D триггеры, работающие по фронту (динамические D триггеры)

Фронт сигнала синхронизации, в отличие от высокого (или низкого) потенциала, не может длиться продолжительное время. В идеальном случае длительность фронта импульса равна нулю. Поэтому в триггере, запоминающем входную информацию по фронту не нужно предъявлять требования к длительности тактового сигнала.

Динамический D триггер, запоминающий входную информацию по фронту, может быть построен из двух D триггеров, работающих по потенциалу (статических D триггеров).

Сигнал синхронизации C будем подавать на статические D триггеры в противофазе. Схема триггера, запоминающего входную информацию по фронту (динамического D триггера) приведена на рисунке 25.

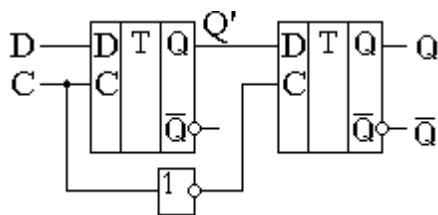


Рис.25 Схема динамического D триггера, работающего по фронту

Рассмотрим работу схемы триггера, приведенной на рисунке 1 подробнее. Для этого воспользуемся временными диаграммами, показанными на рисунке 26.

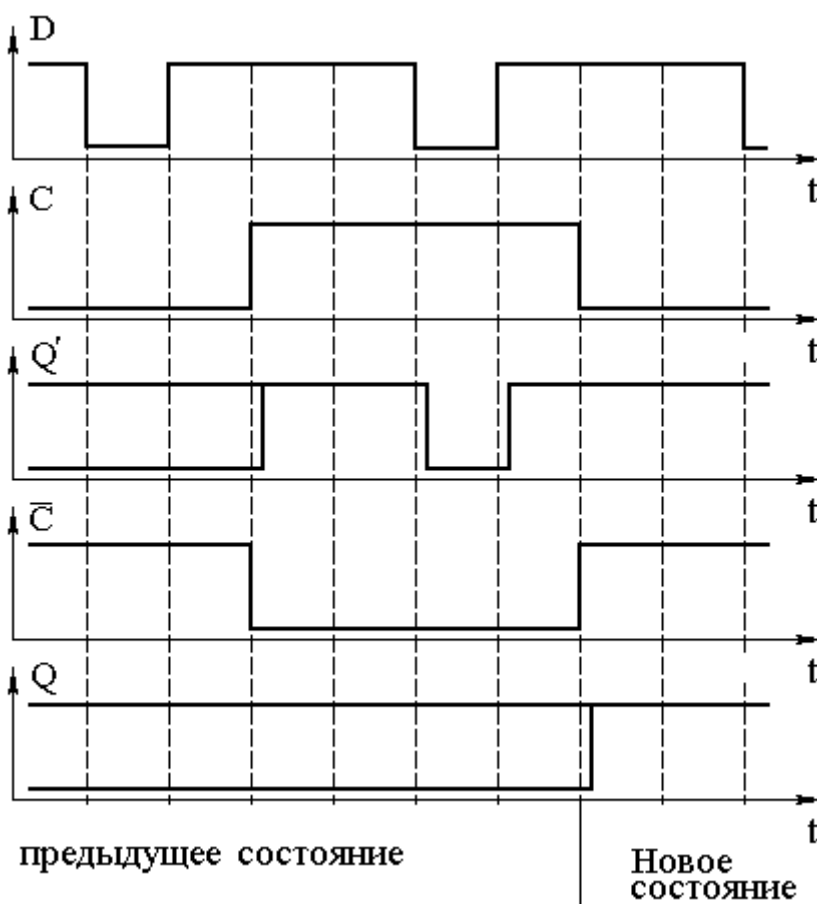


Рис.52. Временные диаграммы D триггера

Условно-графическое обозначение динамического D триггера, запоминающего информацию по фронту тактового сигнала, приведено на рисунке 53.

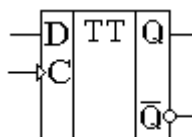


Рис.53. Условно-графическое обозначение динамического D триггера

То, что триггер запоминает входной сигнал по фронту, отображается на условно-графическом обозначении треугольником, изображённым на выводе входа синхронизации. То, что внутри этого триггера находится два триггера, отображается в среднем поле условно-графического изображения двойной буквой Т.

Т триггеры

Т триггер — это счетный триггер. У Т триггера имеется только один вход. После поступления на этот вход импульса, состояние Т триггера меняется на прямо противоположное. Счётным он называется потому, что он как бы подсчитывает количество импульсов, поступивших на его вход. Жаль только, что считать этот триггер умеет только до одного. При поступлении второго импульса Т триггер снова сбрасывается в исходное состояние.

Т триггеры строятся только на базе двухступенчатых триггеров, подобных рассмотренному ранее D триггеру. Использование двух триггеров позволяет избежать неопределенного состояния схемы при разрешающем потенциале на входе синхронизации "С", так как счетные триггеры строятся при помощи схем с обратной связью

Т триггер можно синтезировать из любого двухступенчатого триггера. Рассмотрим пример синтеза Т триггера из динамического D триггера. Для того чтобы превратить D триггер в счётный, необходимо ввести цепь обратной связи с инверсного выхода этого триггера на вход, как показано на рисунке 54.

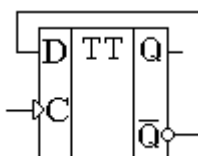


Рис.54 Схема Т триггера, построенная на основе D триггера

Временная диаграмма Т триггера приведена на рисунке 55. При построении этой временной диаграммы был использован триггер, работающий по заднему фронту синхронизирующего сигнала.

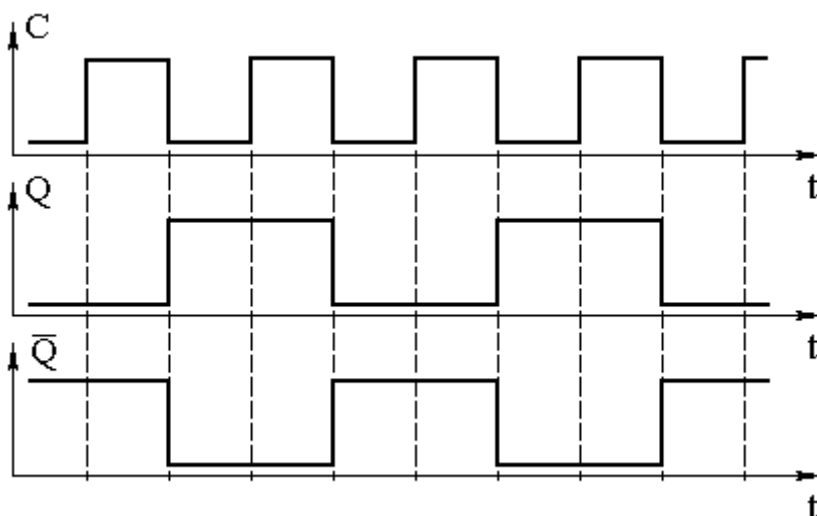


Рис.26 Временные диаграммы Т триггера

T-триггеры используются при построении схем различных счётчиков, поэтому в составе БИС различного назначения обычно есть готовые модули этих триггеров. Условно-графическое обозначение T триггера приведено на рисунке 27.

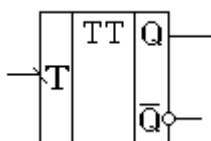


Рис.27 Условно-графическое обозначение T триггера

Подобная схема счетного триггера может быть реализована и на JK триггере. Временная диаграмма синхронного T триггера приведена на рисунке 58, а его условно-графическое обозначение — на рисунке 28.

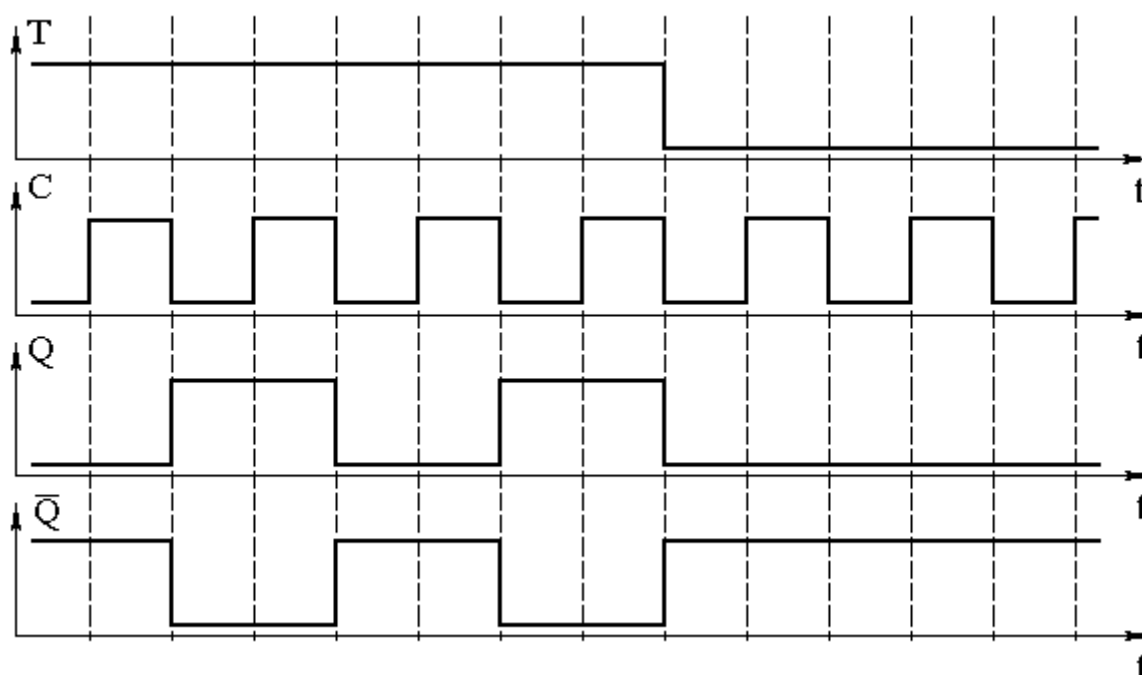


Рис.28 Временные диаграммы синхронного T триггера

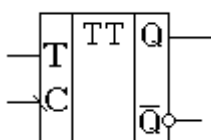


Рис.29 Условно-графическое обозначение синхронного T триггера

jk триггер

Таблица истинности jk триггера практически совпадает с таблицей истинности синхронного RS-триггера. Для того чтобы исключить запрещённое состояние, схема триггера изменена таким образом, что при подаче двух единиц jk триггер превращается в счётный триггер. Это означает, что при подаче на тактовый вход C импульсов jk триггер изменяет своё состояние на противоположное. Таблица истинности jk триггера приведена в таблице 16.

Таблица 16. Таблица истинности jk триггера.

C	K	J	Q(t)	Q(t+1)	Пояснения
0	x	x	0	0	Режим хранения информации
0	x	x	1	1	
1	0	0	0	0	Режим хранения информации
1	0	0	1	1	
1	0	1	0	1	Режим установки единицы J=1
1	0	1	1	1	
1	1	0	0	0	Режим записи нуля K=1
1	1	0	1	0	
1	1	1	0	1	K=J=1 счетный режим триггера
1	1	1	1	0	

Один из вариантов внутренней схемы JK-триггера приведен на рисунке 29.

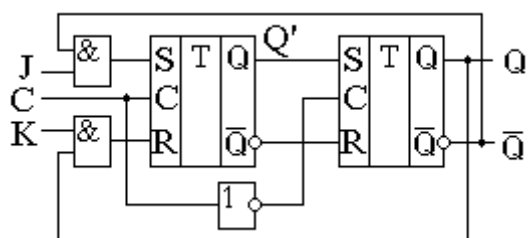


Рис.29. Внутренняя схема jk триггера

Для реализации счетного режима в схеме jk триггера, приведенной на рисунке 59, введена перекрестная обратная связь с выходов второго триггера на входы R и S первого триггера. Благодаря этой обратной связи на входах R и S первого триггера никогда не может возникнуть запрещенная комбинация. При подаче на входы j и k логической единицы одновременно триггер переходит в счетный режим, подобно T триггеру.

Приводить временные диаграммы работы JK-триггера не имеет смысла, так как они совпадают с приведёнными ранее временными диаграммами RS- и T-триггера. Условно-графическое обозначение JK-триггера приведено на рисунке 30.

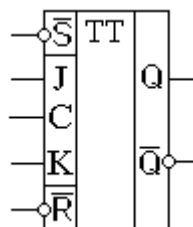


Рис.30. Условно-графическое обозначение jk триггера

На этом рисунке приведено обозначение типовой цифровой микросхемы jk триггера, выполненной по ТТЛ технологии. В промышленно выпускающихся микросхемах обычно кроме входов jk триггера реализуются входы RS-триггера, которые позволяют устанавливать триггер в заранее определённое исходное состояние.

Регистры

Регистром называется последовательное или параллельное соединение триггеров. Регистры обычно строятся на основе D триггеров. При этом для построения регистров могут использоваться как динамические (flip-flop), так и статические D-триггеры (защелки — lath).

Параллельные регистры

Параллельный регистр служит для запоминания многоразрядного двоичного (или недвоичного) слова. Количество триггеров, входящее в состав параллельного регистра определяет его разрядность. Схема четырёхразрядного параллельного регистра приведена на рисунке 66, а его условно-графическое обозначение — на рисунке 31.

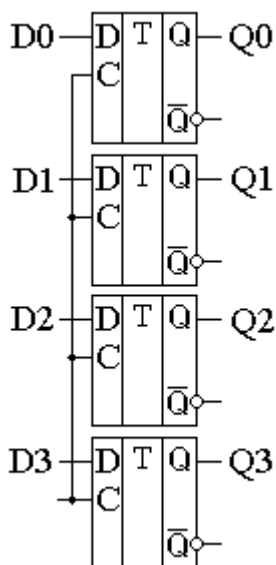


Рис.31 Схема параллельного регистра

В условно-графическом обозначении возле каждого входа D указывается степень двоичного разряда, который должен быть запомнен в этом триггере регистра. Точно таким же образом обозначаются и выходы регистра. То, что микросхема является регистром, указывается в центральном поле условно-графического обозначения символами RG.

В приведённом на рисунке условно-графическом обозначении параллельного регистра инверсные выходы триггеров не показаны. В микросхемах параллельных регистров инверсные выходы триггеров часто не выводятся наружу для экономии количества выводов корпуса.

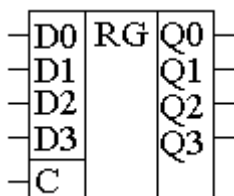


Рис.32 Условно-графическое обозначение параллельного регистра

При записи информации в параллельный регистр все биты (двоичные разряды) должны быть записаны одновременно. Поэтому все тактовые входы триггеров, входящих в состав регистра, объединяются параллельно. Для уменьшения входного тока вывода синхронизации C на этом входе в качестве усилителя часто ставится инвертор.

При решении практических задач часто требуется разрядность параллельных регистров большая восьми. В таком случае можно увеличивать разрядность регистров параллельным соединением готовых микросхем. Принципиальная схема параллельного соединения четырёх регистров приведена на рисунке 33.

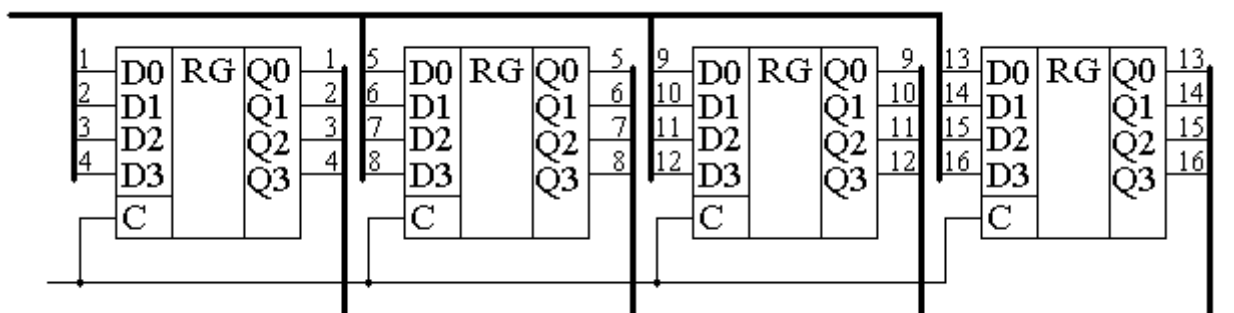


Рис.33 Увеличение разрядности параллельного регистра

Последовательные (сдвиговые) регистры

Кроме параллельного соединения триггеров для построения регистров используются последовательное соединение этих элементов.

Последовательный регистр (регистр сдвига или сдвиговый регистр) обычно служит для преобразования последовательного кода в параллельный и наоборот. Применение последовательного кода связано с необходимостью передачи большого количества двоичной информации по ограниченному количеству соединительных линий. При параллельной передаче разрядов требуется большое количество соединительных проводников. Если двоичные разряды последовательно бит за битом передавать по одному проводнику, то можно значительно сократить размеры соединительных линий на плате (и размеры корпусов микросхем).

Принципиальная схема последовательного (сдвигового) регистра, собранного на основе D триггеров позволяющего осуществить преобразование последовательного кода в параллельный, приведена на рисунке 34. Обратите внимание, что если для параллельных регистров подходили как триггеры работающие по потенциалу (триггеры-защелки), так и триггеры, работающие по фронту, то для реализации последовательного (сдвигового) регистра подходят D триггеры, работающие по фронту!

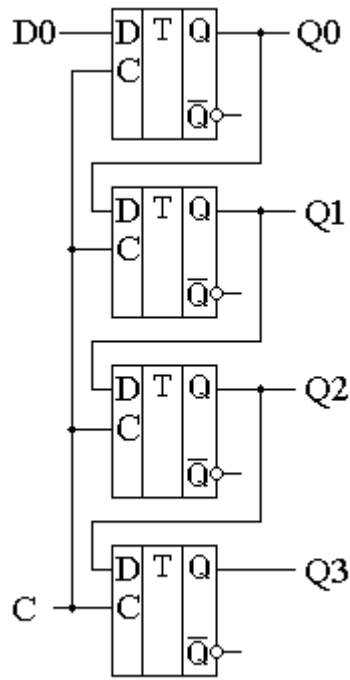


Рис.34 Схема последовательного (сдвигового) регистра

Внутри сдвигового регистра триггеры соединены последовательно, то есть выход первого соединён с входом второго и т.д. Условно-графическое изображение рассмотренного последовательного регистра приведено на рисунке 35.

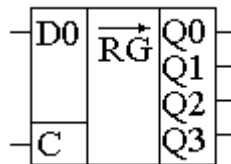


Рис.35 Условно-графическое обозначение последовательного (сдвигового) регистра

Входы синхронизации в последовательных (сдвиговых) регистрах, как и в параллельных регистрах, объединяются. Это обеспечивает одновременность смены состояния всех триггеров, входящих в состав последовательного (сдвигового) регистра.

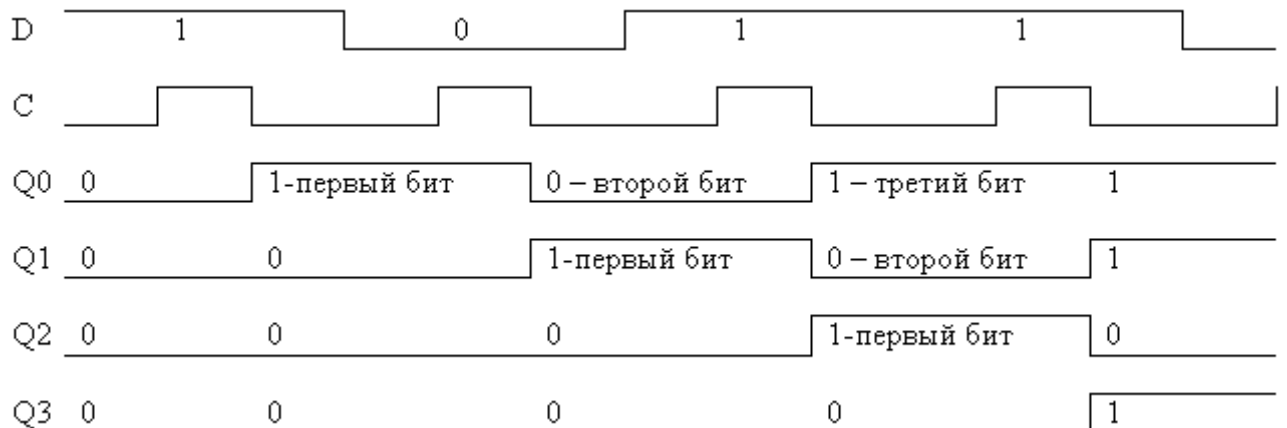


Рис.36 Временная диаграмма работы сдвигового регистра

На рисунке 37 вместе с логическими уровнями записываются значения бит, которые передаются по соединительной линии или присутствуют на выходах сдвигового регистра.

№ такта	1	2	3	1
Q0	1	0	1	1
Q1	X	1	0	1
Q2	X	X	1	0
Q3	X	X	X	1

Счётчики

Счётчики используются для построения схем таймеров или для выборки инструкций из ПЗУ в микропроцессорах. Они могут использоваться как делители частоты в управляемых генераторах частоты (синтезаторах). При использовании в цепи ФАП счётчики могут быть использованы для умножения частоты как в синтезаторах, так и в микропроцессорах.

Двоичные асинхронные счётчики

Простейший вид счётчика — двоичный может быть построен на основе Т-триггера. Т-триггер изменяет своё состояние на прямо противоположное при поступлении на его вход синхронизации импульсов. Для реализации Т-триггера воспользуемся универсальным D-триггером с обратной связью, как это показано на рисунке 37.

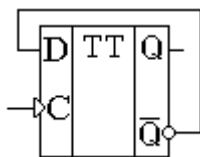


Рис.37 Реализация счетного Т-триггера на универсальном D-триггере

Так как эта схема, при поступлении на вход импульсов меняет свое состояние на противоположное, то её можно рассматривать как счётчик, считающий до двух. Временные диаграммы сигналов на входе и выходах Т-триггера приведены на рисунке 68.

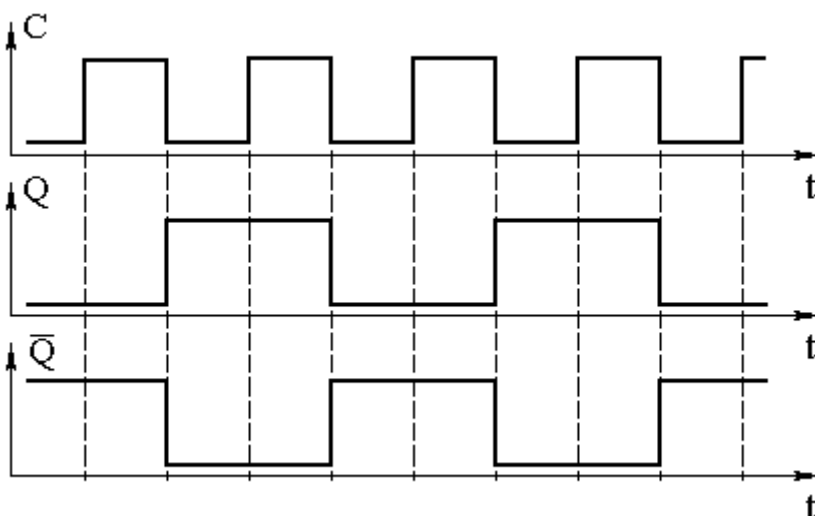


Рис. 38 Временные диаграммы сигналов на входе и выходах Т-триггера.

Обычно требуется посчитать большее количество импульсов. В этом случае можно использовать выходной сигнал первого счетного триггера как входной сигнал для следующего триггера, то есть соединить триггеры последовательно. Так можно построить любой счётчик, считающий до максимального числа, кратного степени два.

Схема счётчика, позволяющего посчитать любое количество импульсов, меньше шестнадцати, приведена на рисунке 39. Количество поступивших на вход импульсов можно узнать, подключившись к выходам счётчика Q0 ... Q3. Это число будет представлено в двоичном коде.

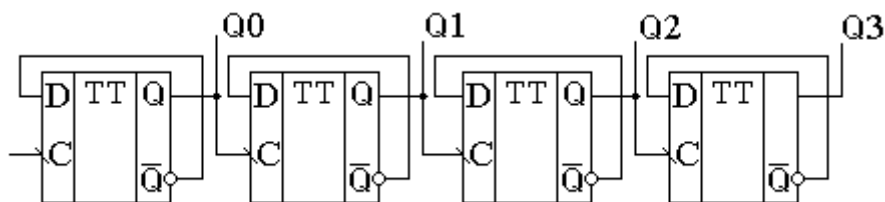


Рис.39 Схема четырёхразрядного счётчика, построенного на универсальных D-триггерах

Для того чтобы разобраться, как работает схема двоичного счётчика, воспользуемся временными диаграммами сигналов на входе и выходах этой схемы, приведёнными на рисунке 40.

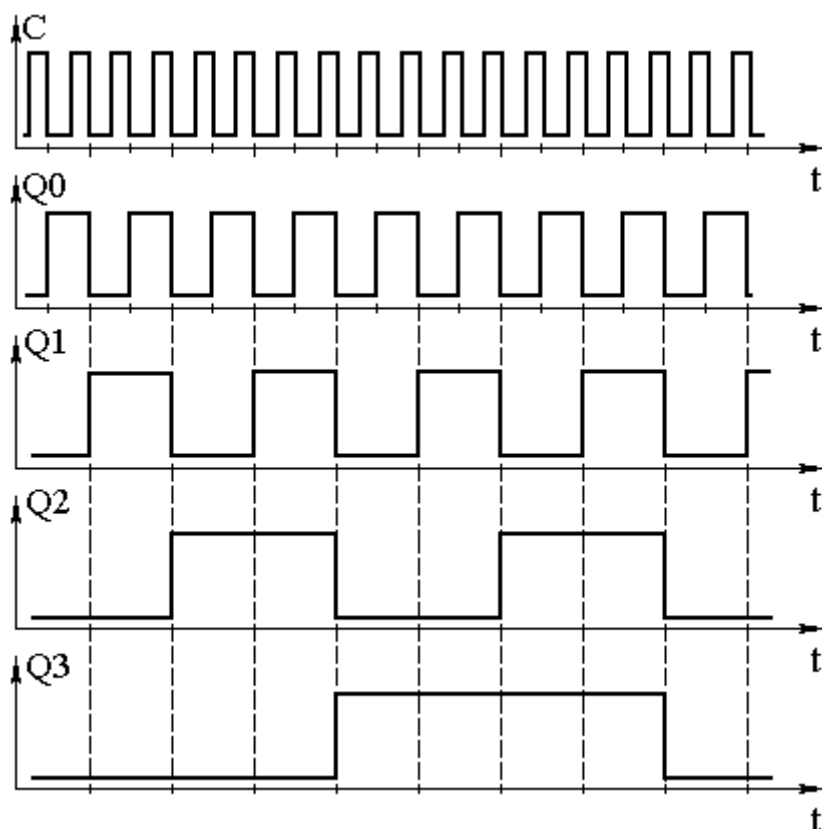


Рис.40 Временная диаграмма четырёхразрядного счётчика

Таблица 18. Изменение уровней на выходе суммирующего счётчика при поступлении на его вход импульсов.

номер входного импульса	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Условно-графическое обозначение суммирующего двоичного счетчика на принципиальных схемах приведено на рисунке 41. В двоичных счётчиках обычно предусматривают вход обнуления микросхемы R, который позволяет записать во все триггеры счётчика нулевое значение. Это состояние иногда называют исходным состоянием счётчика.

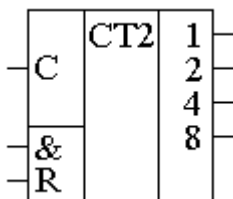


Рис.41 Четырёхразрядный двоичный счётчик.

Таблица 19. Изменение уровней на выходе вычитающего счётчика при поступлении на его вход импульсов.

номер входного импульса	Q3	Q2	Q1	Q0
0	0	0	0	0
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	0	1	1	1
10	0	1	1	0
11	0	1	0	1
12	0	1	0	0
13	0	0	1	1
14	0	0	1	0
15	0	0	0	1

Сумматоры

Построение двоичных сумматоров обычно начинается с сумматора по модулю 2. На рисунке 44 приведена таблица истинности этого сумматора. Ее можно получить исходя из правил суммирования в двоичной арифметике. Предполагается, что читатель знаком с основами двоичной арифметики. Более подробно операции над двоичными числами будут рассмотрены позднее.

X	Y	Out
0	0	0
0	1	1
1	0	1
1	1	0

Таблица истинности сумматора по модулю 2

В соответствии с принципами построения произвольной таблицы истинности, рассмотренными в предыдущей главе, получим схему сумматора по модулю 2. Эта схема приведена на рисунке 44.

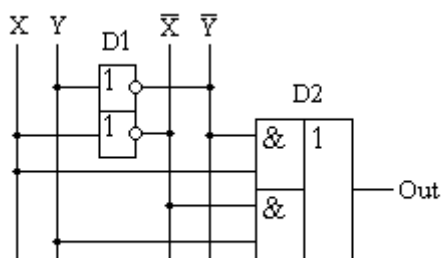


Рис.44 Принципиальная схема сумматора по модулю 2

Сумматор по модулю 2 (для двоичной арифметики его схема совпадает со схемой исключающего "ИЛИ") изображается на схемах как показано на рисунке 45.

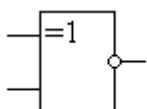


Рис.45 Условно-графическое изображение схемы исключающего "ИЛИ"

Сумматор по модулю 2 выполняет суммирование без учета переноса. В полном двоичном сумматоре требуется учитывать перенос, поэтому требуются схемы, позволяющие формировать перенос в следующий двоичный разряд. Таблица истинности такой схемы, называемой полусумматором, приведена на рисунке 46.

A	B	S	PO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Рисунок 46. Таблица истинности полусумматора

В соответствии с принципами построения произвольной таблицы истинности получим схему полусумматора. Эта схема приведена на рисунке 47.

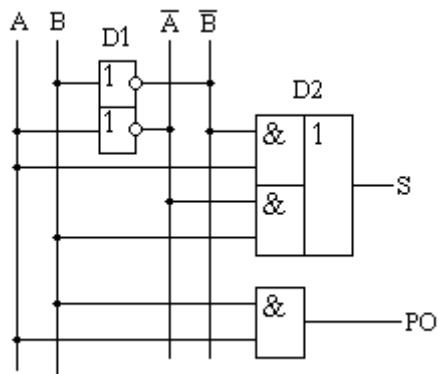


Рис.48 Принципиальная схема, реализующая таблицу истинности полусумматора.

Полусумматор изображается на схемах как показано на рисунке 49.

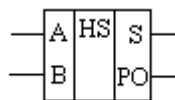


Рис.49 Изображение полусумматора на схемах.

Схема полусумматора формирует перенос в следующий разряд, но не может учитывать перенос из предыдущего разряда, поэтому она и называется полусумматором. Таблицу истинности полного двоичного одноразрядного сумматора можно получить из правил суммирования двоичных чисел.

PI	A	B	S	PO
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

Рис. 50 Таблица истинности полного двоичного одноразрядного сумматора.

В соответствии с принципами построения принципиальной схемы по произвольной таблице истинности получим схему полного двоичного одноразрядного сумматора. Эта схема приведена на рисунке 51. Ее можно минимизировать, но это несколько усложняет принципы построения сумматоров, поэтому вопросы минимизации рассматриваться не будут.

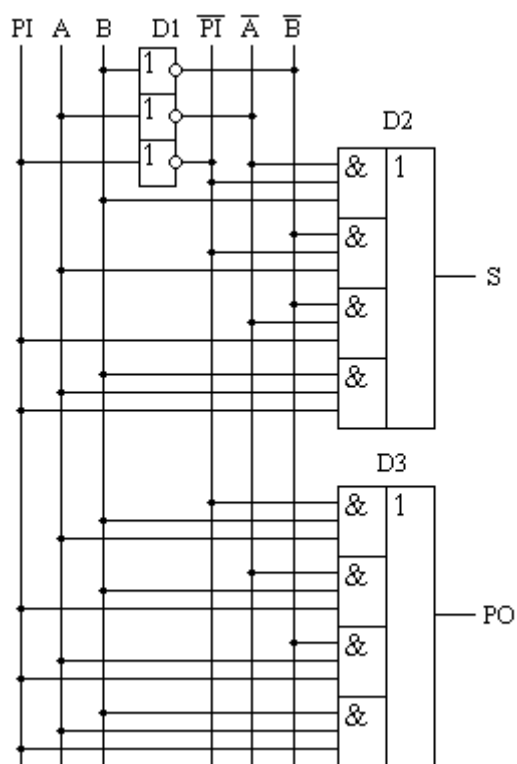


Рис.51 Принципиальная схема, реализующая таблицу истинности полного двоичного одноразрядного сумматора.

Полный двоичный одноразрядный сумматор изображается на схемах как показано на рисунке 52.

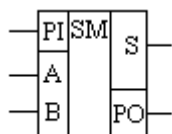


Рис.52 Условно-графическое изображение полного двоичного одноразрядного сумматора.

Шифраторы (кодеры)

Достаточно часто перед разработчиками цифровой аппаратуры встаёт обратная по сравнению с декодированием задача. Требуется преобразовать восьмиричный или десятичный линейный код в двоичный. Линейный восьмиричный код может поступать с выхода механического переключателя. Составим таблицу истинности такого устройства.

Таблица 12. Таблица истинности восьмиричного шифратора (кодера)

№ комбинации	Входы							Выходы		
	1	2	3	4	5	6	7	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	0

3	0	0	1	0	0	0	0	0	1	1
4	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	1	1	1	1

Ещё одним источником линейного восьмеричного кода могут стать аналоговые компараторы с различными порогами срабатывания. Такая линейка компараторов может служить для преобразования аналогового сигнала в цифровой код. Однако двоичный код более компактен. Поэтому требуется преобразователь кода.

Таблица 13. Таблица истинности десятичного шифратора (кодера).

№ комбинации	Входы							Выходы		
	1	2	3	4	5	6	7	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0	0	1	0
3	1	1	1	0	0	0	0	0	1	1
4	1	1	1	1	0	0	0	1	0	0
5	1	1	1	1	1	0	0	1	0	1
6	1	1	1	1	1	1	0	1	1	0
7	1	1	1	1	1	1	1	1	1	1

Таблицы истинности можно объединить. В этом случае ячейки таблицы, где неважно, будет ли записан ноль или будет записана единица, помечены символом 'X'.

Таблица 14. Таблица истинности восьмеричного универсального шифратора (кодера).

№ комбинации	Входы							Выходы		
	1	2	3	4	5	6	7	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1
2	X	1	0	0	0	0	0	0	1	0
3	X	X	1	0	0	0	0	0	1	1
4	X	X	X	1	0	0	0	1	0	0
5	X	X	X	X	1	0	0	1	0	1
6	X	X	X	X	X	1	0	1	1	0
7	X	X	X	X	X	X	1	1	1	1

Теперь можно составить схему устройства. То, что практически во всех строках есть неопределённые значения, позволяет значительно упростить схему восьмеричного кодера. Результирующая схема восьмеричного кодера приведена на рисунке 53.

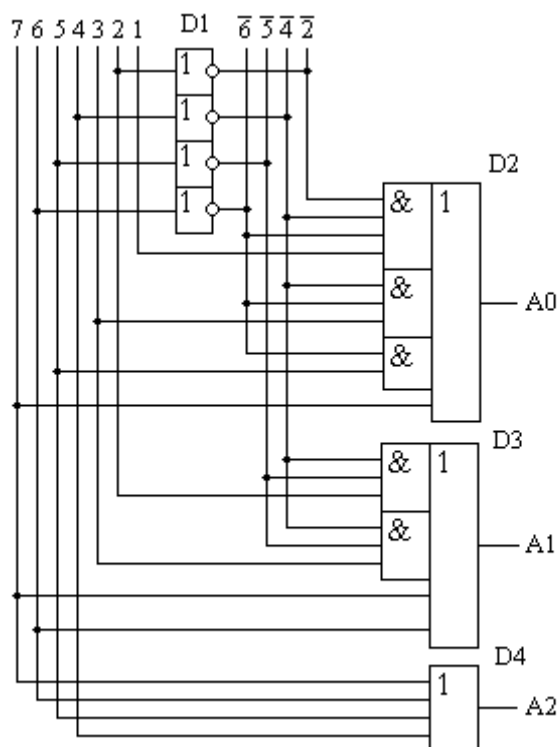


Рис.53. Принципиальная схема восьмеричного кодера.

В настоящее время шифраторы выпускаются в виде отдельных микросхем или используются в виде готовых блоков составе других микросхем, таких как параллельные АЦП. Условно-графическое обозначение шифратора приведено на рисунке 54.

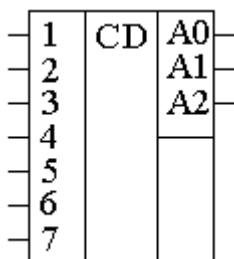


Рис.54 Условно-графическое обозначение восьмеричного шифратора.

Дешифраторы (декодеры)

Декодеры (дешифраторы) позволяют преобразовывать одни виды бинарных кодов в другие. Например, преобразовывать позиционный двоичный код в линейный восьмеричный или шестнадцатеричный. Преобразование производится по правилам, описанным в таблицах истинности, поэтому построение дешифраторов не представляет трудностей. Для построения дешифратора можно воспользоваться правилами синтеза логической схемы для произвольной таблицы истинности.

Десятичный дешифратор (декодер)

Рассмотрим пример построения дешифратора (декодера) из двоичного кода в десятичный. Десятичный код обычно отображается одним битом на одну десятичную цифру. Это классический пример, иллюстрирующий, что нулями и единицами описываются не только двоичные коды. В десятичном коде десять цифр, поэтому для отображения одного десятичного разряда требуется десять выходов дешифратора. В простейшем случае над светодиодом можно просто подписать индицируемую цифру. На входе дешифратора двоичный код записывается в соответствии с правилами двоичной системы счисления. Таблица истинности десятичного декодера приведена в таблице 11.

Таблица 11. Таблица истинности десятичного декодера.

Входы				Выходы									
8	4	2	1	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

В соответствии с принципами построения схемы по произвольной таблице истинности получим схему декодера, реализующего таблицу истинности, приведённую в таблице 11. Эта схема приведена на рисунке 55.

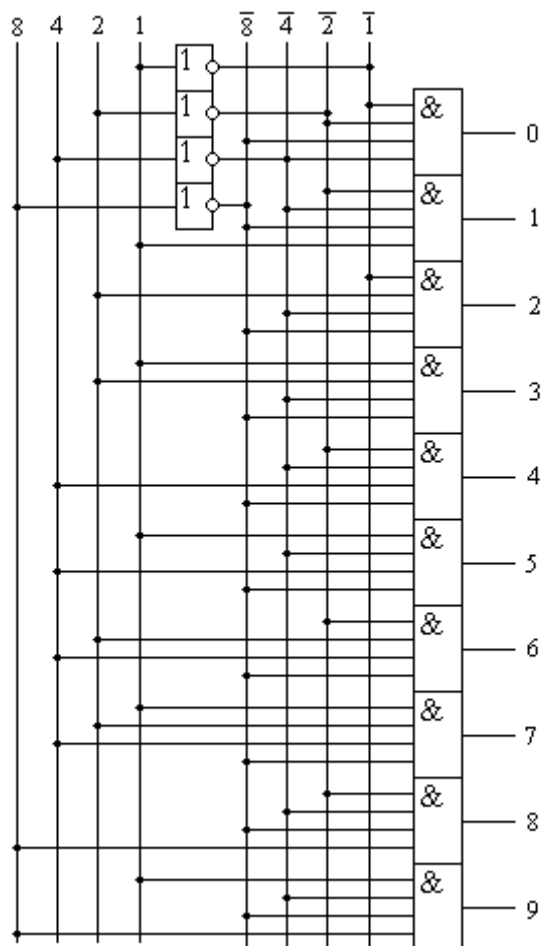


Рис.55. Принципиальная схема двоично-десятичного дешифратора (декодера)

Как видно на этой схеме для реализации каждой строки таблицы истинности потребовалась схема "И". Схема "ИЛИ" не потребовалась, так как в таблице истинности на каждом выходе присутствует только одна единица.

Условно-графическое обозначение микросхемы дешифратора на принципиальных схемах приведено на рисунке. На этом рисунке приведено обозначение двоично-десятичного дешифратора, полная внутренняя принципиальная схема которого изображена на рисунке 56.

Рис.56. Условно-графическое обозначение двоично-десятичного дешифратора

Точно таким же образом можно получить принципиальную схему и для любого другого декодера (дешифратора). Наиболее распространены схемы восьмеричных и шестнадцатеричных дешифраторов. Для индикации такие дешифраторы в настоящее время практически не используются. В основном такие дешифраторы используются как составная часть более сложных цифровых модулей.

Мультиплексоры

Мультиплексорами называются устройства, которые позволяют подключать несколько входов к одному выходу. Демультиплексорами называются устройства,

которые позволяют подключать один вход к нескольким выходам. В простейшем случае такую коммутацию можно осуществить при помощи ключей:

Такой коммутатор одинаково хорошо будет работать как с аналоговыми, так и с цифровыми сигналами. Однако скорость работы механических ключей оставляет желать лучшего, да и управлять ключами часто приходится автоматически при помощи какой-либо схемы.

В цифровых схемах требуется управлять ключами при помощи логических уровней. То есть нужно подобрать устройство, которое могло бы выполнять функции электронного ключа с электронным управлением цифровым сигналом.

Получившийся вариант схемы коммутатора с управлением логическими уровнями приведен на рисунке 56.

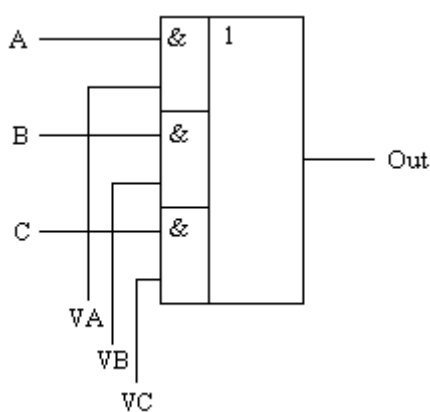


Рис.56 Принципиальная схема мультиплексора, выполненная на логических элементах

В схемах можно одновременно включать несколько входов на один выход. Однако обычно это приводит к непредсказуемым последствиям. Кроме того, для управления таким коммутатором требуется много входов, поэтому в состав мультиплексора обычно включают двоичный дешифратор, как показано на рисунке 57. Этот дешифратор получен нами ранее при помощи синтеза логических схем (СДНФ). Это позволяет управлять переключением информационных входов при помощи двоичных кодов, подаваемых на управляющие входы. Количество информационных входов в таких схемах выбирают кратным степени числа два.

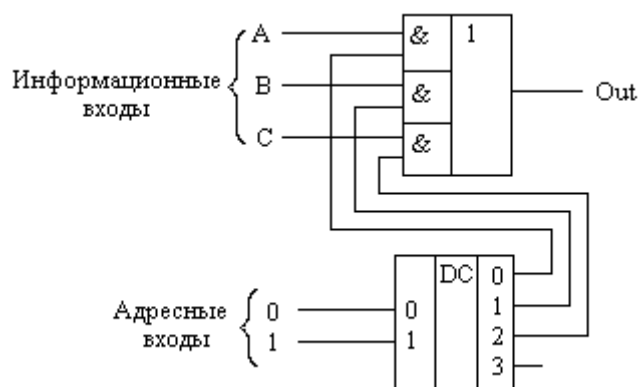


Рис.57. Принципиальная схема мультиплексора, управляемого двоичным кодом

Условно-графическое обозначение четырёхходового мультиплексора с двоичным управлением приведено на рисунке 44. Входы A0 и A1 являются управляющими входами мультиплексора, определяющими адрес входного сигнала, который будет соединён с выходным выводом мультиплексора Y. Сами входные сигналы обозначены как X0, X1, X2 и X3.

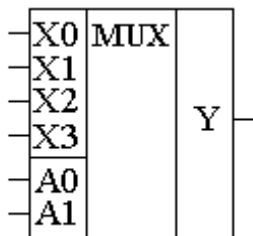


Рис.44 Условно графическое обозначение четырёхходового мультиплексора

В условно-графическом обозначении названия информационных входов A, B, C и D заменены названиями X0, X1, X2 и X3, а название выхода Out заменено на название Y. Такое название входов и выходов более распространено в отечественной литературе. Адресные входы обозначены как A0 и A1.

Демультимплексоры

Демультимплексорами называются устройства, которые позволяют подключать один вход к нескольким выходам. Демультимплексор можно построить на основе точно таких же схем логического "И", как и при построении мультиплексора. Существенным отличием от мультиплексора является возможность объединения нескольких входов в один без дополнительных схем. Однако для увеличения нагрузочной способности микросхемы, на входе демультимплексора для усиления входного сигнала лучше поставить инвертор.

Схема демультимплексора приведена на рисунке 58. В этой схеме для выбора конкретного выхода демультимплексора, как и в мультиплексоре, используется двоичный дешифратор.

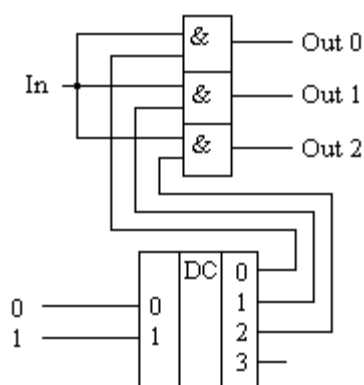


Рис.58 Принципиальная схема демультимплексора, управляемого двоичным кодом

Однако, если рассмотреть принципиальную схему самого дешифратора, то можно значительно упростить демультимплексор. Достаточно просто к каждому логическому элементу 'И', входящему в состав дешифратора просто добавить ещё один вход – In. Такую схему часто называют дешифратором с входом разрешения работы. Условно-графическое изображение демультимплексора приведено на рисунке 59.

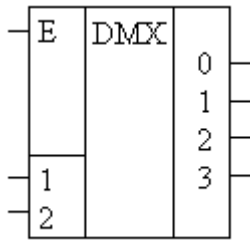


Рис.59 Условно графическое обозначение демультиплексора с четырьмя выходами

В этом обозначении вход I_n обозначен как вход E , а выходы не названы никак, оставлены только их номера. Триггеры предназначены для запоминания двоичной информации. Использование триггеров позволяет реализовывать устройства оперативной памяти (то есть памяти, информация в которой хранится только на время вычислений). Однако триггеры могут использоваться и для построения некоторых цифровых устройств с памятью, таких как счётчики, преобразователи последовательного кода в параллельный или цифровые линии задержки.

Компараторы сигналов.

Компараторы сигналов на ОУ.

Для определения момента равенства двух сигналов используются схемы включения ОУ в компараторных режимах, в которых можно использовать как один вход ОУ, так и оба. Петля ООС обычно не замыкается. Если в схеме компаратора ОУ охватывается слабой положительной ОС, то передаточная характеристика компаратора приобретает гистерезисные свойства.

Одновходовой компаратор сигналов предназначен для сравнения разнополярных входных напряжений, причем в момент их равенства по абсолютной величине выходное напряжение компаратора $U_{\text{ВЫХ}}$ переключается в другое предельное состояние. До момента времени τ напряжение U_c меньше по модулю, чем опорное напряжение $E_{\text{оп}}$, поэтому последнее определяет состояние выхода. В данном случае $E_{\text{оп}}$ больше 0, поэтому $U_{\text{ВЫХ}}$ примерно равно E_k^- . После достижения входным сигналом U_c порогового значения $-E_{\text{оп}}(R1/R2)$, выходное напряжение определяется входным сигналом U_c , при этом напряжение $U_{\text{ВЫХ}}$ становится равным E_k^+ . В момент точного равенства $U_c = E_{\text{оп}}(R1/R2)$, усилитель компаратора находится в неустойчивом линейном режиме. Переключение состояния выхода происходит с некоторой задержкой $\Delta\tau$, которая определяется временем перезаряда паразитных емкостей схемы ОУ.

Наклон линейной переходной характеристики определяется коэффициентом усиления ОУ и его скоростью отклика ρ . Чтобы увеличить скорость переключения, ОУ в схемах компараторов используется без ООС. Одновходовой компаратор имеет ограниченное входное сопротивление, однако позволяет сравнивать большие по амплитуде сигналы без появления ошибок синфазной составляющей $\Delta U_{\text{синф}}$. Амплитуда сигналов между входами ОУ не должна превышать допустимого уровня для входных дифференциальных сигналов, однако относительная точность сравнения сигналов тем выше, чем больше амплитуда. В цепь неинвертирующий вход — земля необходимо включать балансирующий резистор $R_{\text{бал}} = R1 \parallel R2$.

Двухвходовой компаратор позволяет сравнивать сигналы одинаковой полярности.

Уровень этих сигналов должен находиться в пределах допустимого для данного ОУ синфазного входного напряжения $E_{\text{синф}}$. Когда сигналы, подаваемые на разные входы, уравниваются, выходной сигнал компаратора должен быть равен нулю.

В схеме на основе ОУ типа μ А709 амплитуда сигнала, возникающего на выходе, ограничена с помощью диодов VD1 и VD2, поэтому сигнал от компаратора можно непосредственно подавать на входы низкоуровневых цифровых ИС. Эту же схему выходной цепи можно применить для одноходового компаратора. Для данной схемы компаратора минимальная разность входных сигналов $|U_c - E_{\text{оп}}|$, при которой будет достигнут порог срабатывания цифровой ИС (U_n примерно равно 2 В, при $K_0 = 40$ тыс.), составляет

$$|U_c - E_{\text{оп}}|_{\text{min}} = U_n / K_0 = 0,05 \text{ мВ}$$

Для двухкаскадных ОУ, аналогичных LM101, амплитуда выходного сигнала ограничивается с помощью стабилитрона, подключаемого ко второму каскаду усиления. В реальной схеме к рассчитанному значению порога срабатывания компаратора необходимо прибавить ошибки, возникающие за счет входных токов ОУ, напряжение смещения нуля и синфазной составляющей сигнала. Последняя особенно велика для двухходового компаратора, определяющего момент равенства двух однополярных сигналов большого уровня. Время нарастания выходного сигнала компаратора, построенного на стандартном ОУ, до уровня +4 В при превышении $U_0 - E_{\text{оп}} = 10$ мВ составляет примерно 0,5 мкс, время задержки выходного сигнала 3 ... 5 мкс.

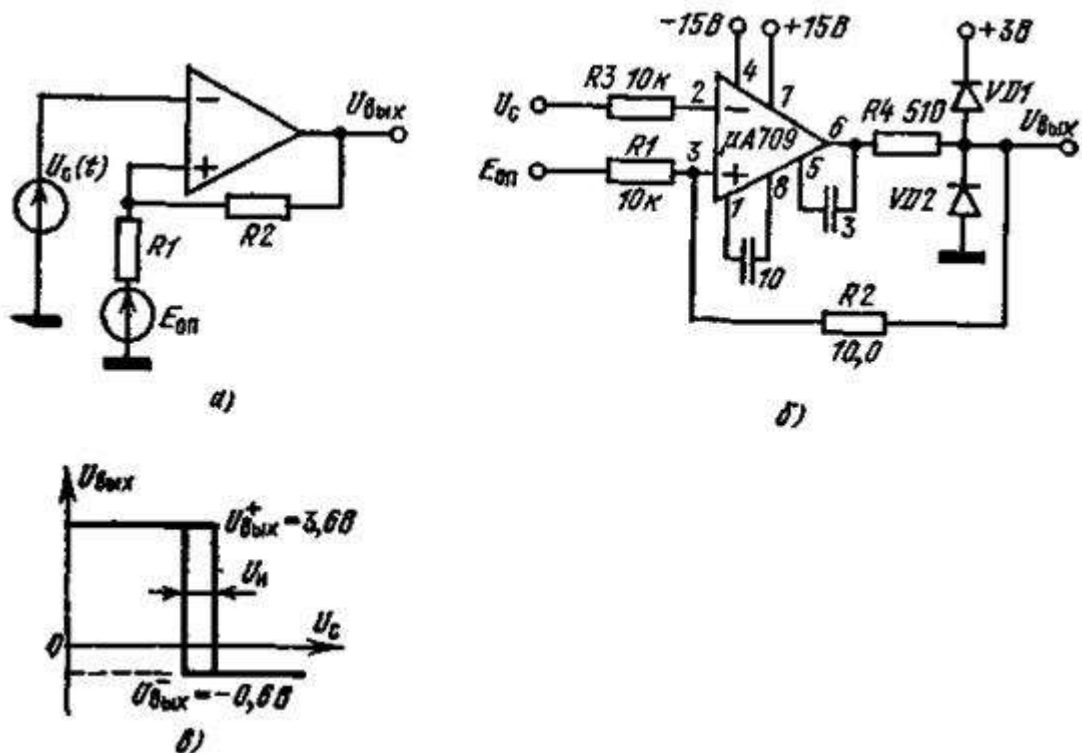
Регенеративные компараторы (триггеры Шмитта).

Регенеративный компаратор (рисунок 1, а) — это схема с положительной ОС. Если опорное напряжение на его входе отсутствует ($E_{\text{оп}} = 0$), то компаратор срабатывает по собственному порогу, т. е. является триггером Шмитта. Уровень этого порога определяется предельными уровнями выходного сигнала $U_{\text{вых max}}^{\pm}$ и коэффициентом положительной ОС

Напряжение порога срабатывания регенеративного компаратора (триггера Шмитта)

$$U_{\text{п}} = U_{\text{н}} / 2 = U_{\text{вых max}} / (2\beta).$$

Напряжение $U_{\text{н}}$ является напряжением гистерезиса триггера Шмитта. Вводить гистерезис в компаратор целесообразно в том случае, если сигнал приходит на фоне помех: после переключения выходного сигнала компаратора относительно большим импульсом сигнала напряжение $U_{\text{н}}$ служит запасом на помехоустойчивость, поскольку возврат компаратора в исходное состояние произойдет лишь когда сигнал уменьшится на заданную величину $U_{\text{н}}$



а — функциональная схема; б — принципиальная схема;
в — передаточная характеристика.

Рисунок 60 - Компаратор с гистерезисной характеристикой.

На рисунке 60, б представлены полная схема регенеративного компаратора и его характеристика переключения. Выходной сигнал данной схемы (рисунок 60, в) совместим со входными сигналами цифровых ИС. Напряжение гистерезиса в этой схеме

$$U_H^\pm = (U_{\text{ВЫХ}}^+ - U_{\text{ВЫХ}}^-) / [R_1 / (R_1 + R_2)] = 4.2\text{мВ}$$

Схема регенеративного компаратора с положительной ОС является основой триггерных и мультивибраторных схем, строящихся на ОУ.:

Преобразователи произвольных кодов

Преобразователь произвольных кодов - это логическая схема, реализующая определённую математическую формулу. На вход такой схемы подаётся двоичный код, соответствующий значению переменной $x_{\text{вх}}$. На выходе схемы получается двоичный код, соответствующий значению функции $x_{\text{вых}}$, вычисленному по указанной математической формуле.

В качестве примера синтеза логической схемы рассмотрим 3-входовую схему, реализующую увеличение входного кода в три раза: $x_{\text{вых}} = 3x_{\text{вх}}$. Последовательность действий при решении подобных задач следующая

1. Определим максимально возможный код на выходе 3-входовой схемы: $7_{10} \times 3_{10} = 21_{10} = 10101_2$ - это число. Поэтому количество выходов для данной схемы будет равно пяти.

2. Заполним таблицу истинности для синтезируемой схемы (табл.1). Поскольку количество выходов данной схемы больше одного, таблица включает в себя несколько (здесь *пять*) столбцов, соответствующих двоичным разрядам выходного сигнала.
3. Для каждого выхода найдем минимальное выражение с помощью карт Карно (рис. 1).
4. По полученным выражениям построим логическую схему на пять выходов, каждый из которых соответствует двоичному разряду вычисляемого по заданной формуле числа (рис. 2).

Таблица .1. Таблица истинности трех-входовой схемы умножения на 3

В десятичном выражении	Входной код			В десятичном выражении	Выходной код				
	В двоичном коде				В двоичном коде				
	a 2^2	b 2^1	c 2^0		Q₄ 2^4	Q₃ 2^3	Q₂ 2^2	Q₁ 2^1	Q₀ 2^0
0	0	0	0	0	0	0	0	0	0
1	0	0	1	3	0	0	0	1	1
2	0	1	0	6	0	0	1	1	0
3	0	1	1	9	0	1	0	0	1
4	1	0	0	12	0	1	1	0	0
5	1	0	1	15	0	1	1	1	1
6	1	1	0	18	1	0	0	1	0
7	1	1	1	21	1	0	1	0	1

Рис.6 1. Минимизация логических выражений для выходных сигналов преобразователя, реализующего формулу $x_{\text{вых}} = 3x_{\text{вх}}$

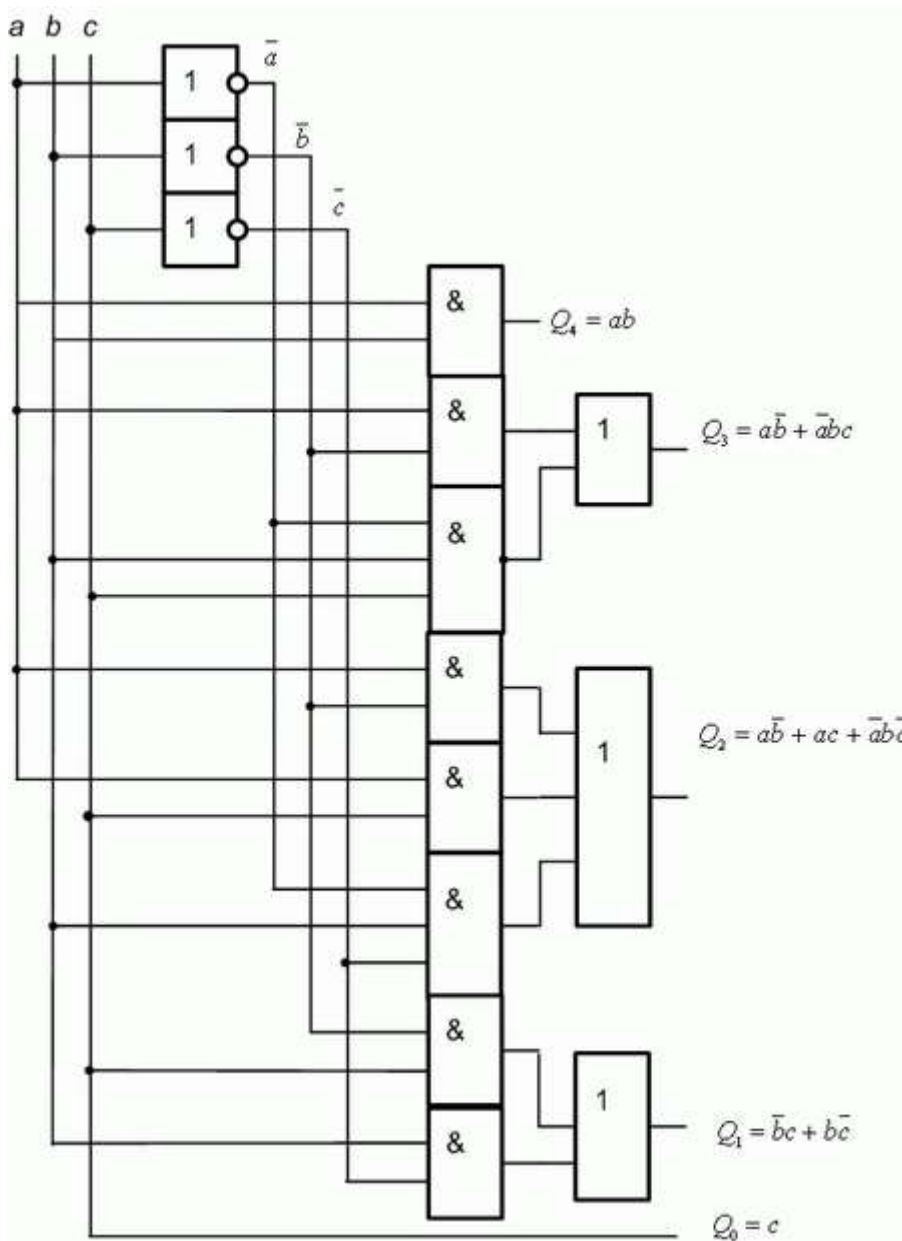


Рис. 61. Логическая схема преобразователя на 3 входа, реализующего формулу умножения на 3

Световые индикаторные устройства.

Практически любое микропроцессорное устройство содержит элементы световой индикации. В качестве световых индикаторов в настоящее время чаще всего применяются светодиоды. На рынке имеется огромный выбор светодиодов, самых разных видов и размеров: светодиоды повышенной яркости, мигающие, двух- и даже трехцветные. Светодиоды выделяются среди других светоизлучательных элементов благодаря особой экономичности и долговечности.

В микропроцессорных устройствах световые индикаторы могут служить для отображения различных режимов работы: предупреждать о критических ситуациях, отражать ход приема управляющих сигналов. Подключить одиночный светодиодный индикатор к микропроцессору очень просто. На рисунке 62 приведена схема подключения светодиода непосредственно к выводу порта микроконтроллера.

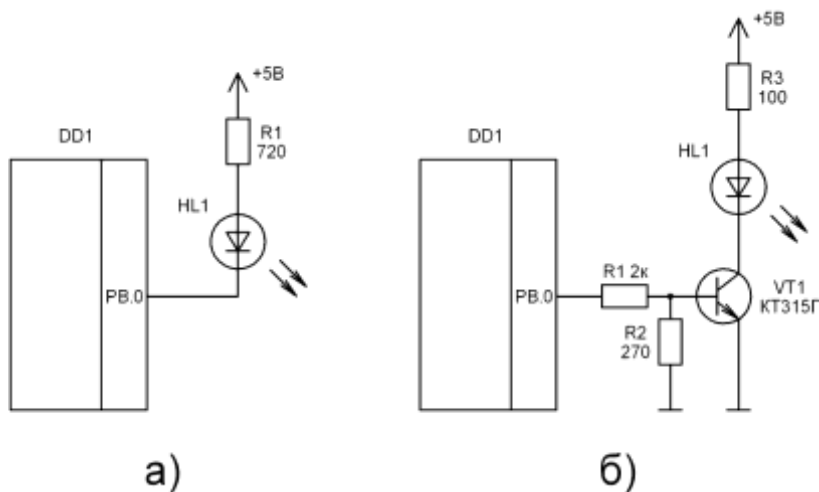


Рис. 62 а) Простейшая схема подключения светодиодного индикатора;
 б) Схема подключения светодиода посредством электронного ключа.

Все выходные каскады микроконтроллера имеют достаточную нагрузочную способность для того, чтобы выдержать подключение одного светодиодного индикатора. Тут имеется в виду светодиод среднего и миниатюрного размера с током потребления в рабочем режиме не больше 20 мА. Если вы желаете подключить по такой схеме несколько светодиодов на несколько разных выходов микроконтроллера, то суммарный ток, проходящий через все эти выходы, не должен превышать 80 мА.

Если нужно подключить индикатор большого размера или повышенной яркости, который потребует большего рабочего тока, придется применить буферный элемент. Одним из вариантов буферного элемента является электронный ключ. Схема подключения светодиода при помощи электронного ключа приведена на рисунке 62б).

Основой электронного ключа служит транзистор VT1. Резистор R1 - токоограничивающий в цепи базы. Резистор R2 служит для надежного запираения ключа при нулевом сигнале на выходе PB.0. Резистор R3 ограничивает рабочий ток светодиода. Его номинал может изменяться в зависимости от типа применяемого светодиода. Если в качестве VT1 использовать транзистор KT315, схема способна обеспечивать рабочий ток светодиода до 250 мА.

Кроме максимально допустимого тока нагрузки, обе приведенные выше схемы отличаются способом управления. Схема, изображенная на рис. а), с инверсным способом управления. Для того, чтобы зажечь светодиод, нужно подать на выход PB.0 низкий логический уровень. Схема, изображенная на рис. б), обеспечивает прямое управление для того, чтобы зажечь светодиод, включенный по этой схеме, на выход PB.0 нужно подать сигнал логической единицы.

С программной точки зрения управление светодиодом, включенным по любой из двух вышеприведенных схем, не представляет никаких трудностей. Например, для того, чтобы установить линию PB.0 в единичное состояние, процессор должен просто выполнить команду:

```
sbi          PORTB          ,          0
```

При этом в первом случае (а) светодиод потухнет, а во втором (б) - загорится. Для сброса сигнала в нулевое состояние достаточно подать команду:

```
cbi          PORTB          ,          0
```

Для тех же целей можно использовать команды вывода в порт:

```
ldi r16, 0b00000001
out PORTB, r16
```

Однако в этом случае одновременно устанавливаются значения всех восьми разрядов порта В. Иногда именно это и требуется.

Еще одним способом подключения мощных световых индикаторных устройств является использование микросхем с открытым коллектором. Например микросхема ULN2003 (DD2) способна коммутировать до семи индикаторов, причем ток через каждый не должен превышать 500ма, а напряжение 50В. Схема подключения мощных индикаторов при помощи данной микросхемы приведена на рисунке 62. Каждый ключ представляет собой транзисторную сборку Дарлингтона.

Для управления индикатором необходимо подать высокий уровень напряжения на соответствующий вход коммутатора.

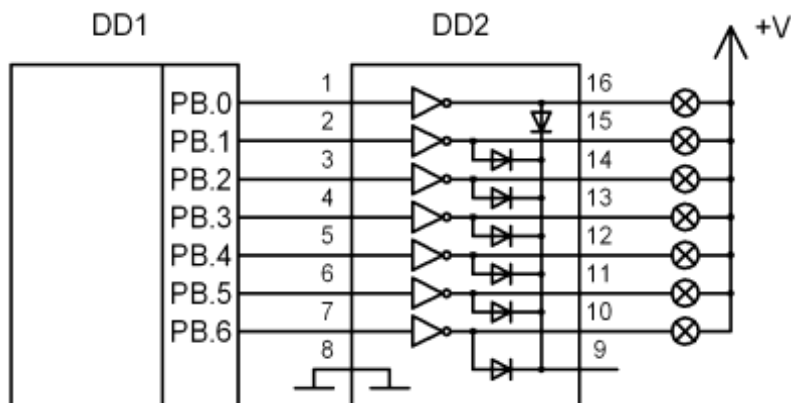


Рис.62. Схема подключения индикаторов посредством сборки ключей с открытым коллектором.

Кроме одиночных световых индикаторов в микропроцессорных устройствах часто применяют знакосинтезирующие матрицы, которые еще называют цифровыми индикаторами. Простейшим примером цифрового индикатора может служить так называемый семисегментный индикатор. Вам наверняка хорошо знаком такой тип индикаторов. Он широко применяется в самых разных устройствах цифровой техники: от калькулятора, до электронных часов. Семисегментный индикатор представляет собой матрицу из семи светодиодов продолговатой формы, размещенных таким образом, чтобы, зажигая их в разных сочетаниях, можно было бы отобразить любую десятичную цифру (имеются в виду арабские цифры от 0 до 9). Кроме семи основных сегментов индикатор чаще всего дополняют восьмым маленьким сегментом, который предназначен для отображения десятичной точки (запятой). Расположив в ряд несколько таких индикаторов, можно отображать любое десятичное число с плавающей запятой.

ГЛАВА 3 ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

Запоминающие устройства. Структурные схемы запоминающих устройств. Микросхемы ОЗУ.

Арифметико-логические устройства. Организация и принципы действия. Операции в АЛУ.

Постоянные запоминающие устройства (ПЗУ)

Очень часто в различных применениях требуется хранение информации, которая не изменяется в процессе эксплуатации устройства. Это такая информация как программы в микроконтроллерах, начальные загрузчики (BIOS) в компьютерах, таблицы коэффициентов цифровых фильтров в сигнальных процессорах, DDC и DUC, таблицы синусов и косинусов в NCO и DDS. Практически всегда эта информация не требуется одновременно, поэтому простейшие устройства для запоминания постоянной информации (ПЗУ) можно построить на мультиплексорах. Иногда в переводной литературе постоянные запоминающие устройства называются ROM (readonlymemory — память доступная только для чтения). Схема такого постоянного запоминающего устройства (ПЗУ) приведена на рисунке 83.

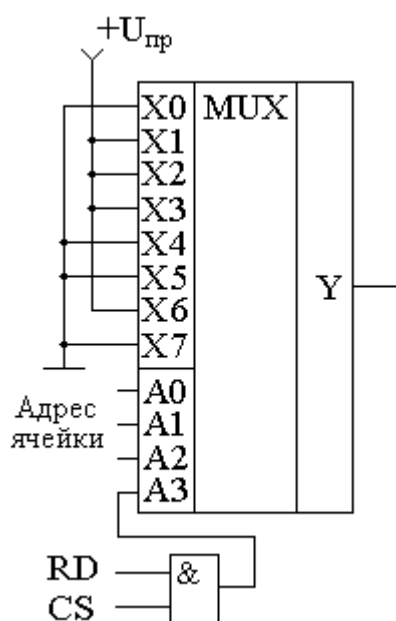


Рис. 63. Схема постоянного запоминающего устройства (ПЗУ), построенная на мультиплексоре.

В этой схеме построено постоянное запоминающее устройство на восемь одноразрядных ячеек. Запоминание конкретного бита в одноразрядную ячейку производится запайкой провода к источнику питания (запись единицы) или запайкой провода к корпусу (запись нуля). На принципиальных схемах такое устройство обозначается как показано на рисунке 6.

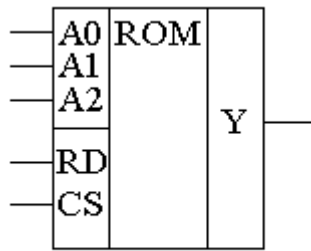


Рис. 64. Обозначение постоянного запоминающего устройства на принципиальных схемах.

Для того, чтобы увеличить разрядность ячейки памяти ПЗУ эти микросхемы можно соединять параллельно (выходы и записанная информация естественно остаются независимыми). Схема параллельного соединения одноразрядных ПЗУ приведена на рисунке 65.

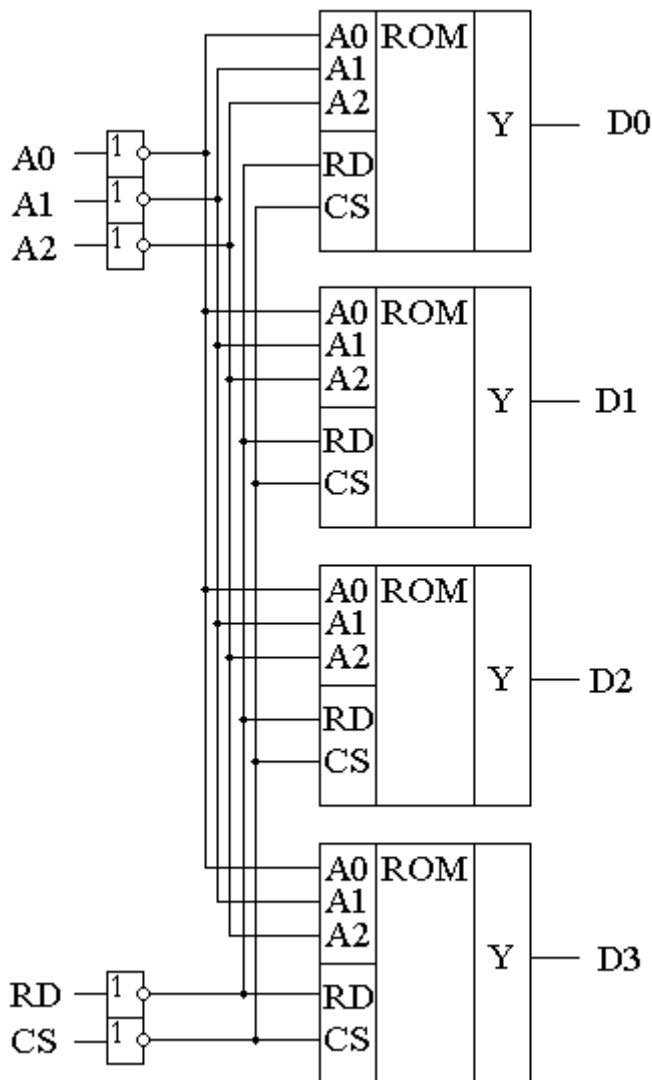


Рис.65 Схема многоразрядного ПЗУ (ROM).

В реальных ПЗУ запись информации производится при помощи последней операции производства микросхемы — металлизации. Металлизация производится при помощи маски, поэтому такие ПЗУ получили название **масочных ПЗУ**. Еще одно отличие реальных микросхем от упрощенной модели, приведенной выше — это использование кроме мультиплексора еще и демультиплексора. Такое решение позволяет превратить одномерную запоминающую структуру в двухмерную и, тем самым, существенно сократить объем схемы дешифратора, необходимого для работы схемы ПЗУ. Эта ситуация иллюстрируется следующим рисунком:

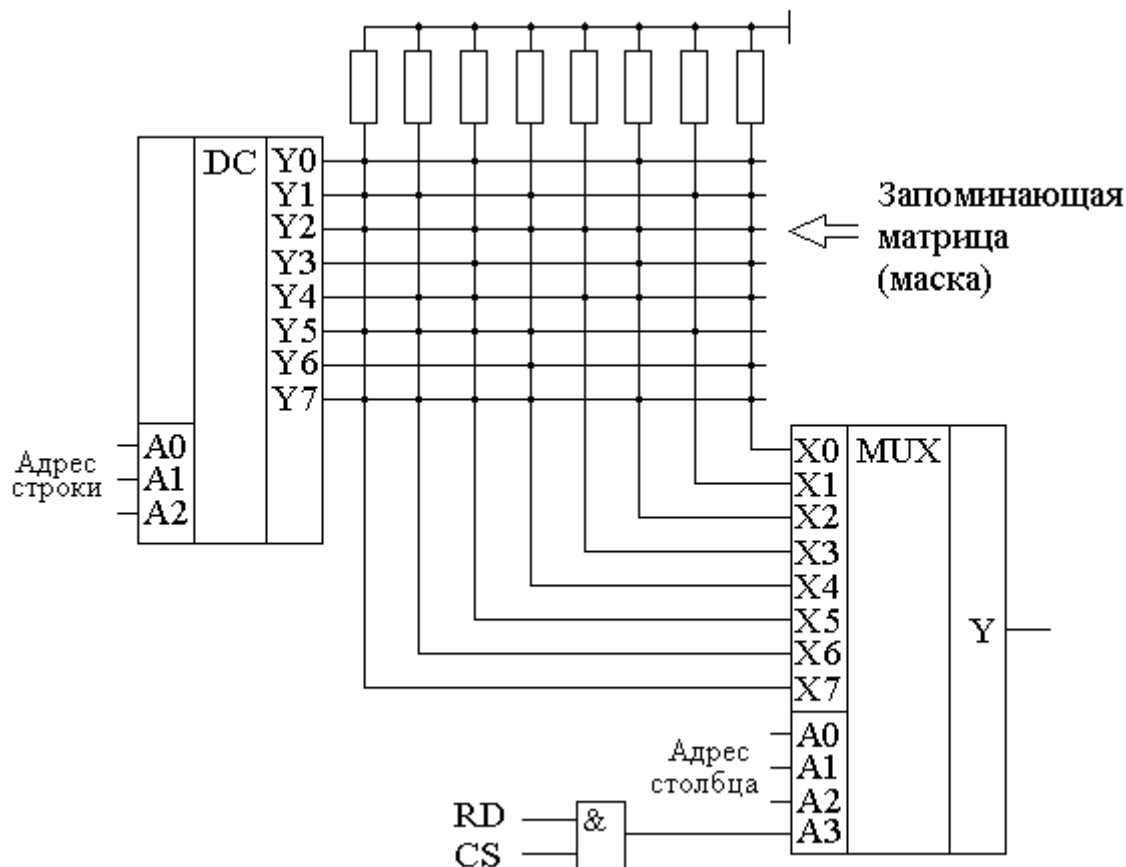


Рис.66 Схема масочного постоянного запоминающего устройства (ROM).

Масочные ПЗУ изображаются на принципиальных схемах как показано на рисунке87. Адреса ячеек памяти в этой микросхеме подаются на выходы A0 ... A9. Микросхема выбирается сигналом CS. При помощи этого сигнала можно наращивать объем ПЗУ (пример использования сигнала CS приведён при обсуждении [ОЗУ](#)). Чтение микросхемы производится сигналом RD.

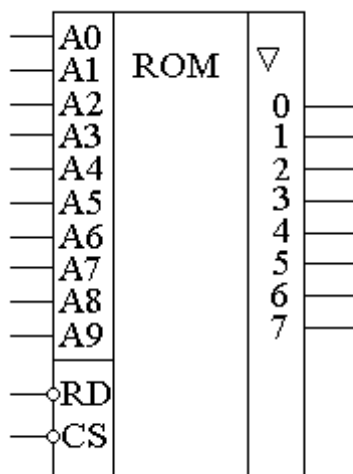


Рисунок 87. Условно-графическое обозначение масочного ПЗУ (ROM) на принципиальных схемах.

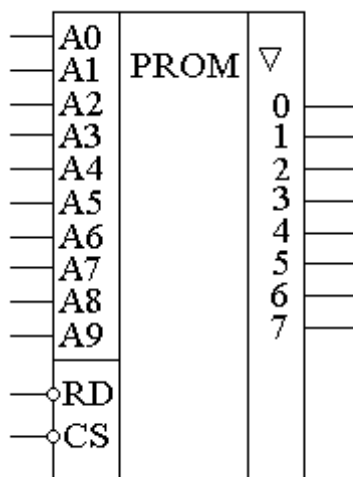


Рис.67 Условно-графическое обозначение программируемого постоянного запоминающего устройства (PROM) на принципиальных схемах.

Программируемые ПЗУ оказались очень удобны при мелкосерийном и среднесерийном производстве. Однако при разработке радиоэлектронных устройств часто приходится менять записываемую в ПЗУ программу. ППЗУ при этом невозможно использовать повторно, поэтому раз записанное ПЗУ при ошибочной или промежуточной программе приходится выкидывать, что естественно повышает стоимость разработки аппаратуры. Для устранения этого недостатка был разработан еще один вид ПЗУ, который мог бы стираться и программироваться заново.

Арифметико-логические устройства

Основными арифметическими операциями являются сложение и вычитание. Разработаны коды дополнительный и обратный, которые позволяют выполнять операцию вычитания методом суммирования. Для выполнения операции вычитания, при использовании дополнительного кода, вычитаемое следует перевести в дополнительный код и

просуммировать с первым слагаемым. Полученный результат (разность) будет представлен в дополнительном коде. Затем его следует перевести в прямой код.

Прямой и дополнительный код положительных чисел совпадают. При преобразовании отрицательного числа в дополнительный код все разряды прямого кода следует проинвертировать и к младшему разряду добавить единицу. При обратном преобразовании дополнительного кода в прямой результат следует также проинвертировать и к младшему разряду добавить единицу.

Таким образом, выполнение операции вычитания методом суммирования требует дополнительных затрат времени и снижает быстродействие вычислительных средств.

Для повышения быстродействия ЭВМ разработаны и используются комбинированные арифметико-логические устройства, которые обеспечивают выполнение ряда арифметических и логических операций над прямыми кодами чисел без их преобразования.

Методика построения одноразрядного арифметического устройства для выполнения операций суммирования и вычитания показана на рисунке 68. Из сравнения логических выражений операций суммирования (Рисунок 88, а) и вычитания (Рисунок 88, б) следует, что выражения для суммы и разности совпадают, а выражение для заёма является частью операции суммирования или вычитания.

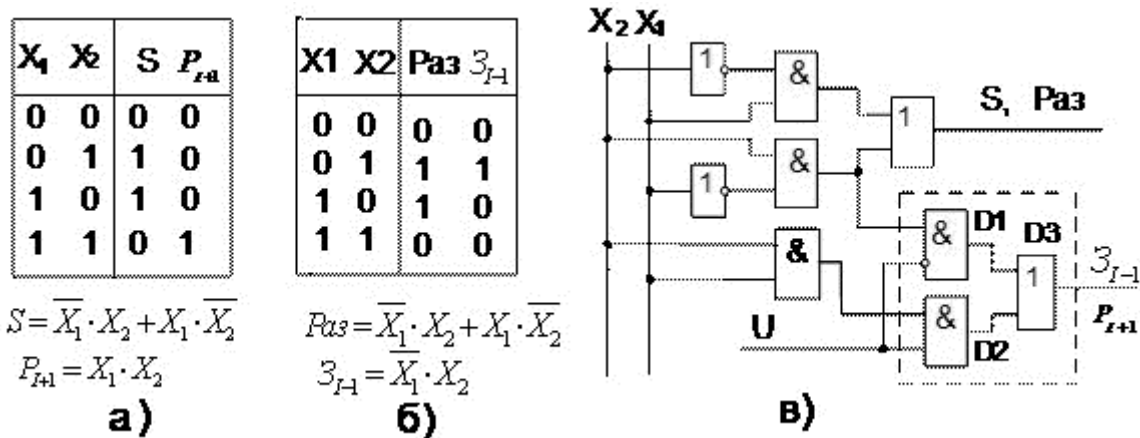


Рисунок 68 Таблицы истинности и структурные формулы операции суммирования а), вычитания б) и схема одноразрядного АЛУ в).

Таким образом, для выполнения операции вычитания не требуется получение дополнительных сигналов, поэтому и не требуются дополнительные аппаратные затраты. Необходимо лишь обеспечить коммутацию сигналов переноса и заёма в соответствии с кодом операции.

На рисунке 88, в) приведена схема простейшего АЛУ, на которой роль устройства управления выполняют два клапана, управляемые разнополярными сигналами от управляющего напряжения U. Эта часть схемы на рисунке 88, в) выделена пунктирной

линией. При $U=0$ выполняется операция вычитания, а при $U=1$ — операция суммирования.

Многоразрядные АЛУ выпускаются в виде интегральных микросхем или входят в состав процессоров, являясь их основой.

МС 564ИПЗ (Рисунок 69,а) — это 4-разрядное параллельное АЛУ, выполняющая 16 арифметических и 16 логических операций.

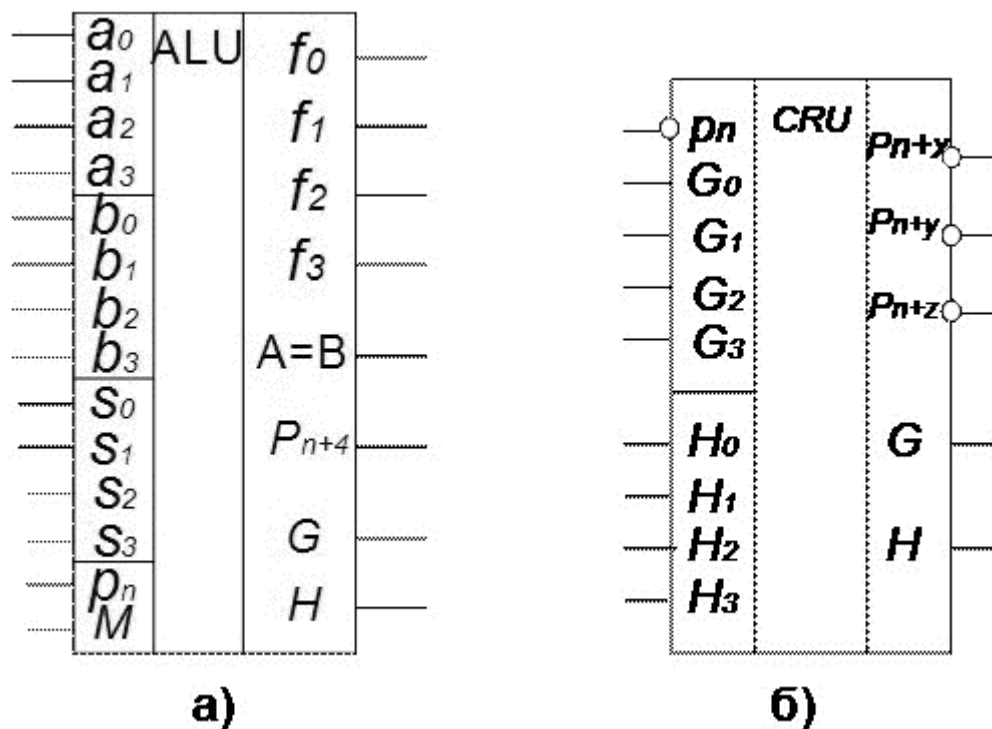


Рисунок 89 Схема 4-разрядного АЛУ 564ИПЗ а) и схема ускоренного переноса 564ИП4 б).

$A(a_0-a_3)$ — первый операнд,

$B(b_0-b_3)$ — второй операнд,

$S(s_0-s_3)$ — код операции — 4 разряда.

Если $M=0$, то выполняются арифметические операции: $2^4=16$, при $M=1$ выполняются логические операции: $2^4=16$. Итого $16+16=32$ операции.

$F(f_0-f_3)$ — результат операции. На выходе $A=B$ появляется «1», если при выполнении операции вычитания результат операции будет равен «0», то есть $A=B$. Поскольку АЛУ параллельного типа, то имеются выходы генерации G и распространения переноса H . P_n и P_{n+4} — входной и выходной переносы.

Для увеличения разрядности обрабатываемых слов МС АЛУ можно соединять последовательно, как и в параллельных сумматорах с последовательным переносом. При этом, конечно, увеличивается время выполнения операций.

ЛИТЕРАТУРА:

1. Амосов, В. Схемотехника и средства проектирования цифровых устройств [Электронный ресурс] / В. Амосов. - СПб. : БХВ-Петербург, 2010. - 560 с.
2. Угрюмов, Е. Цифровая схемотехника [Электронный ресурс]: учебное пособие / Е. Угрюмов. - СПб. : БХВ-Петербург, 2010. - 816 с.