

Логический элемент ТТЛ (транзисторно-транзисторная логика). Простейший логический элемент ТТЛ строится на базе многоэмиттерного транзистора VT1, выполняющего функцию И для сигналов, подаваемых на его эмиттеры, и транзисторного ключа VT2, выполняющего функцию НЕ

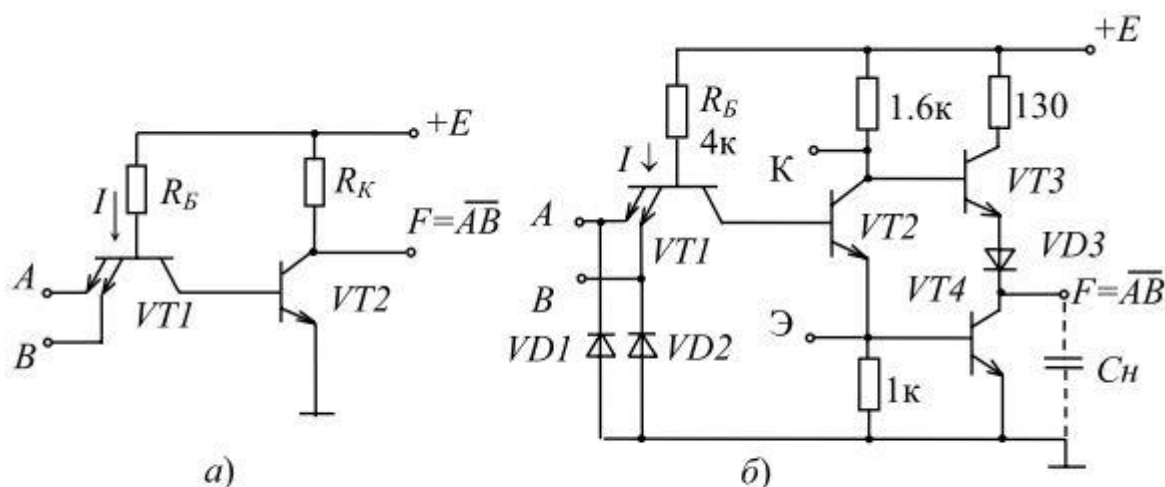


Рис. 10.8—Электрическая схема логического элемента ТТЛ

(рис.10.8, а). Если на всех входах высокие уровни напряжения ($A = 1$ и $B = 1$), закрыты эмитерные переходы $VT1$, открывается переход база-коллектор этого транзистора и его ток I открывает и насыщает ключевой транзистор $VT2$, формируя на выходе низкий уровень напряжения ($F = 0$). При низком уровне сигнала на любом из входов ток I протекает через выбранный эмиттер, т.е. переключается в выходную цепь источника сигнала, коллекторный ток становится равным нулю, закрывая $VT2$. При этом $F = 1$. Таким образом, схема реализует таблицу истинности элемента 2И-НЕ. К сожалению, схема рис. 10.8, а неэкономична. Дело в том, что при входном сигнале 0, когда $VT2$ насыщен, через RK течет большой ток IKH , потребляя от источника питания значительную мощность, которая лишь нагревает схему. Для снижения IKH , казалось бы, достаточно увеличить сопротивление RK . Однако это резко сократило бы нагрузочную способность схемы, так как ток, потребляемый подключенными к выходу нагрузками, создавал бы на RK настолько большое падение напряжения, что потенциал логической единицы на выходе мог бы стать недопустимо низким. Для повышения экономичности и быстродействия выходной ключ выполняют по схеме сложного инвертора (рис.10.8, б). При $A = B = 1$ открыты транзисторы $VT2$ и $VT4$ (транзистор $VT3$ закрыт), и емкость нагрузки CH быстро разряжается через малое сопротивление открытого транзистора $VT4$. Для любой другой комбинации входных сигналов емкость нагрузки имеет возможность быстро зарядиться до высокого уровня напряжения через низкое выходное сопротивление эмиттерного повторителя $VT3$ ($VT2$ и $VT4$ закрыты).

Собственное потребление тока в выходной цепи мало как при логическом нуле на выходе (закрыт VT3), так и при логической 1 на выходе (закрыт VT4). С целью повышения помехоустойчивости эмиттеры VT1 часто соединяют с корпусом через диоды (VD1, VD2), запертые для входных сигналов положительной полярности. Они открываются только при отрицательной полярности импульсов, которые могут возникать при наличии помех и переходных процессах в схеме. Для расширения функциональных возможностей базовый элемент ТТЛ имеет выводы К и Э от коллектора и эмиттера транзистора (рис.10.8, б), к которым могут подключаться логические расширители, представляющие собой как отдельные микросхемы, так и схемы, встроенные в один корпус с базовым элементом. Схема расширителя представляет собой часть базового элемента, включающую входную логику на многоэмиттерном транзисторе и транзистор VT2 инвертора с выводами К и Э. Если соединить одноименные выводы базового элемента и расширителя, то произойдет «расширение по ИЛИ», означающее, что транзистор VT4 базового элемента откроется в любом случае, открыт ли транзистор VT2 базового элемента или транзистор VT2 расширителя, так как они включены параллельно. Логический расширитель. В схеме логического расширителя (рис. 10.9, а) задействованы лишь элементы RB, VT1 и VT2 базовой схемы. Логические расширители (например, К155ЛД1) используются совместно с другими логическими элементами (например, К155ЛР3 или К155ЛР1), имеющими соответствующие входы К и Э (см. точки К и Э базовой схемы ТТЛ). При подключении логического расширителя к базовой схеме логического элемента И-НЕ (рис.10.9, б) формируется элемент двухступенчатой логики И-ИЛИ-НЕ.

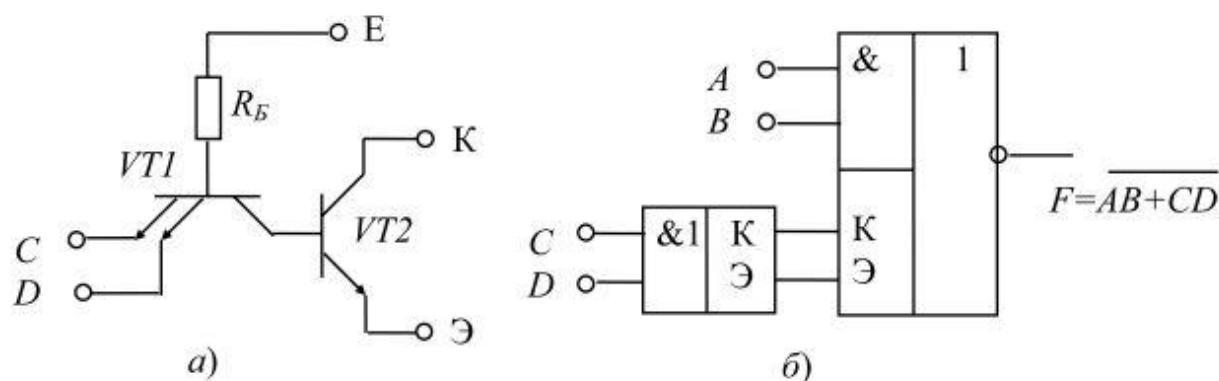


Рис. 10.9–Логический расширитель и его подключение к базовой схеме логического элемента ТТЛ

Логический элемент ТТЛШ (транзисторно-транзисторная логика с диодами Шоттки).

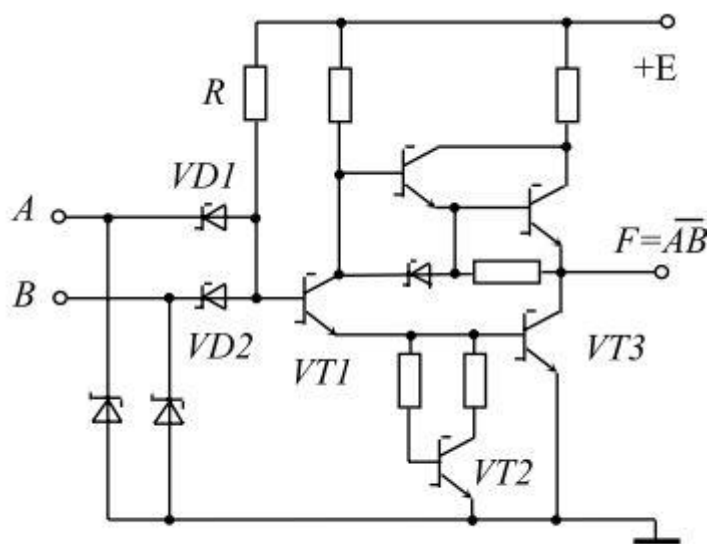


Рис. 10.10–Базовый элемент ТТЛШ

Базовый логический элемент ТТЛШ (на рис. 10.10 обозначена лишь часть элементов типовой схемы, существенных для приведенных ниже пояснений) выполняется подобно элементу ТТЛ, но коллекторные переходы транзисторов зашунтированы диодами Шоттки. Диоды Шоттки исключают насыщение транзисторов, поэтому технология ТТЛШ свободна от недостатков, связанных с рассасыванием избыточных зарядов и позволяет существенно (в 3-5 раз) повысить быстродействие по сравнению с ТТЛ-логикой. В маломощных микросхемах ТТЛШ высокое быстродействие сочетается с умеренным потреблением мощности. Вместо многоэмиттерного транзистора в таких ИМС для выполнения логических функций И-НЕ, ИЛИ-НЕ используются диоды Шоттки (рис. 10.10). При $A = B = 1$ диоды $VD1$ и $VD2$ закрыты, ток от источника питания E через резистор R протекает в базу $VT1$ и открывает его. На эмиттерной нагрузке $VT1$ ($VT2$ со своими резисторами) возникает высокий уровень напряжения, открывающий выходной транзистор $VT3$, напряжение на коллекторе которого принимает уровень логического нуля. В случае, когда на одном из входов действует напряжение логического нуля, низкий уровень напряжения на анодах диодов $VD1$, $VD2$ блокирует транзисторы $VT1$ и $VT2$. На выходе логического элемента появляется напряжение логической единицы. Таблица истинности элемента соответствует логической функции 2И-НЕ. Логический элемент ЭСЛ (эмиттерно-связанная логика). Элементы ЭСЛ являются основной базой для микросхем сверхвысокого быстродействия (до единиц гигагерц). Для уменьшения задержек переключения транзисторы в элементах ЭСЛ не насыщаются. Это достигается путем введения глубокой обратной отрицательной связи по току с помощью резисторов, включаемых в цепи эмиттеров. Уменьшению задержек переключения способствует также ограничение пределов изменения перепадов напряжения и использование

эмиттерных повторителей для ввода и вывода сигналов (рис. 10.11). Эмиттерные повторители на входе элемента способствуют увеличению нагрузочной способности за счет существенного повышения входного сопротивления, на выходе – существенно снижают постоянную времени выходной цепи в связи с очень малым выходным сопротивлением повторителя. Элемент ЭСЛ реализует переключение тока I_0 . При закрытых $VT1$ и $VT2$ ($A = B = 0$) ток I_0 течет через транзистор $VT3$. Для этого на базу $VT3$ подается постоянное опорное напряжение $U_{оп}$. На коллекторе $VT3$ за счет падения напряжения на R_K формируется низкий потенциал, который повторяется на выходе (уровень логического нуля). При подаче высокого уровня (логической единицы) на базу

хотя бы одного из транзисторов VT1 или VT2 ток I_0 течет через них, а транзистор VT3 из-за фиксированного напряжения на его базе закрыт, и его высокий потенциал формирует на выходе элемента логическую единицу. Следовательно, элемент ЭСЛ реализует на выходе логическую функцию ИЛИ. Если эмиттерный повторитель (на схеме не показан) подключить к объединенным коллекторам транзисторов VT1, VT2, то на его выходе можно реализовать логическую функцию ИЛИ-НЕ. Напряжение питания ЭСЛ обычно выбирается равным – 5,2 В, уровню логической единицы соответствует напряжение – 0,98 В, а уровню логического нуля – 1,63 В. Элемент работает в формате положительной логики, так как переход от логического нуля к логической единице совершается с повышением потенциала. Несмотря на малые значения времени переключения, импульсные помехи в цепях питания незначительны, так как потребление тока в этой схеме не изменяется при ее переключении. Логические элементы КМОП (комплементарная полевая логика). Элементы реализуются на полевых транзисторах с изолированным затвором с индуцированными n- и p-каналами. Общая закономерность построения таких структур состоит в том, что параллельное соединение одного типа транзисторов сопровождается последовательным соединением транзисторов противоположного типа. Проиллюстрируем это положение на примере базовых элементов 2ИЛИ – НЕ (рис. 10.12, а) и 2И – НЕ (рис. 10.12, б).

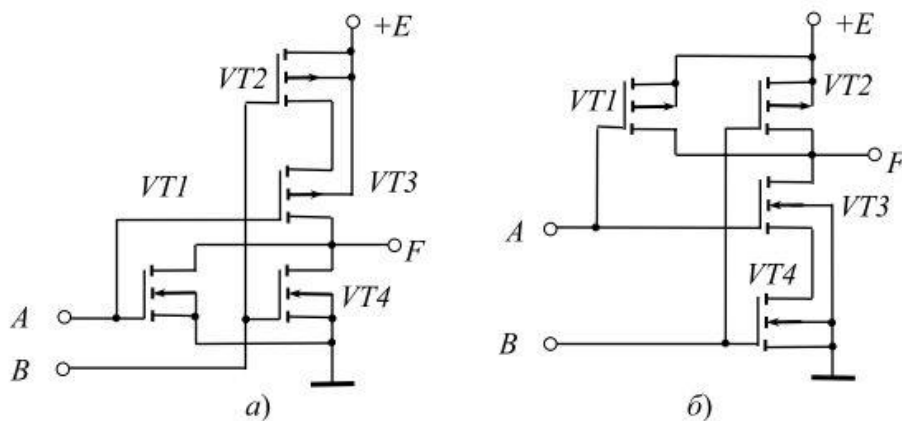


Рис. 10.12–Базовые элементы КМОП логики: а) 2ИЛИ – НЕ, б) 2И – НЕ

В этих схемах пары транзисторов VT1, VT3 и VT2, VT4 комплементарные, когда один из них заперт, другой открыт. Пусть в схеме на рис. 10.12, а на оба входа А и В подан логический ноль (нулевой потенциал в положительной логике). Тогда транзисторы n-типа VT1 и VT4 заперты, так как разность потенциалов между их затворами и истоками, присоединенными к корпусу, близка к нулю. Транзисторы p-типа VT2 и VT3 открыты, потому что нулевой потенциал, подведенный к затворам, относительно их подложек, соединенных с положительным полюсом источника питания, создает отрицательную разность потенциалов, необходимую для индуцирования в них p-канала. Но так как через эти каналы протекают лишь пренебрежимо малые токи запертых транзисторов VT1 и VT4, падения напряжения на транзисторах VT2 и VT3 незначительны, и значит, выходное напряжение, примерно равное + E, соответствует логической единице. Если хотя бы на один из входов, например А, подать логическую единицу, т.е. потенциал близкий к +E, то соответствующий p-канальный транзистор (в данном случае VT3) закроется, отключая источник питания от выхода, на который через открывшийся транзистор с n-каналом (в рассматриваемом случае VT1) подан нулевой потенциал корпуса, т.е. логический ноль. Таким образом, в рассматриваемой схеме логический перепад напряжений выше, чем в любой другой логике (коэффициент использования напряжения питания близок к единице). Аналогично можно проанализировать работу схемы рис. 10.12 б, реализующей в положительной логике функцию И-НЕ. Главным достоинством КМОП

ИМС является ничтожно малое потребление энергии в статическом режиме. Один из выходных транзисторов мик-

росхемы всегда закрыт. Транзисторы включены последовательно, и цепь тока питания разорвана, если не учитывать ничтожно малого тока запертого транзистора. Однако во время переключения схемы из одного стационарного состояния в другое в течение некоторого времени (пока происходит перезаряд паразитных емкостей) открыты оба комплементарных транзистора. Через схему проходит импульс тока, а значит, потребляется небольшая порция энергии от источника питания. С ростом частоты переключения потребление энергии растет. При частоте переключения 5 МГц потребление может доходить до нескольких милливатт на корпус. На комплементарной паре транзисторов строятся аналоговые ключи (рис. 10.13, а). При $W = 1$ он замкнут (между точками X и Y сопротивление равно примерно 100 Ом), при $W = 0$ разомкнут. Такие ключи используются для коммутации не только цифровых, но и аналоговых сигналов. Добавление ключа к выходу обычного логического элемента обеспечивает ему состояние разомкнутого выхода, в которое он переходит при $W = 0$.

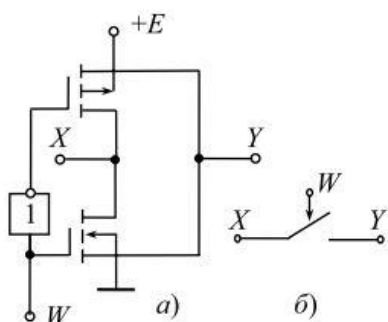


Рис. 10.13—Ключ КМОП:
а) схема, б) электрический эквивалент ключа

Логический элемент И²Л (интегральная инжекционная логика). Электрическая схема элемента 2ИЛИ-НЕ приведена на рис. 10.14.

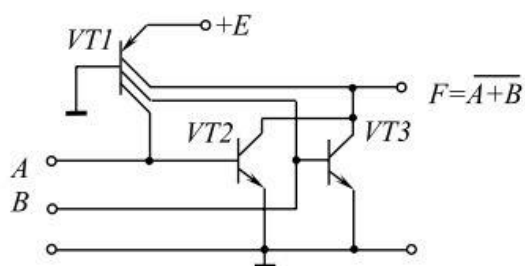


Рис. 10.14—Базовый логический элемент И²Л

Многоколлекторный биполярный транзистор (МКТ) VT1 носит название инжекторного. При наличии логической единицы хотя бы на одном из логических входов, на объединенных коллекторах транзисторов VT1, VT2, питаемых одним из коллекторов МКТ, возникает низкий уровень напряжения (0). Только при $A = B = 0$ оба транзистора VT1 и VT2 закрыты и на выходе возникает высокий уровень напряжения (1), близкий к напряжению источника питания E. Согласно рассмотренной таблице истинности, логический элемент работает в соответствии с логикой 2ИЛИ-НЕ.

Использование Булевой алгебры при построении логических цифровых устройств.

Различные логические переменные могут быть связаны функциональными зависимостями. Функциональные зависимости между логическими переменными могут быть описаны логическими формулами или таблицами истинности. В общем виде логическая *формула* функции двух переменных записывается в виде: $y=f(X_1, X_2)$, где X_1, X_2 — входные переменные. В *таблице истинности* отображаются все возможные сочетания (комбинации) входных переменных и соответствующие им значения функции y , получающиеся в результате выполнения какой-либо логической операции. При одной переменной полный набор состоит из четырёх функций, которые приведены в таблице 2.

Таблица 2 – Полный набор функций одной переменной

Таблица 2 – Полный набор функций одной переменной				
X	Y1	Y2	Y3	Y4
0	1	0	1	0
1	0	1	1	0

Y1 — Инверсия, Y2 — Тожественная функция, Y3 — Абсолютно истинная функция и Y4 – Абсолютно ложная функция.

Инверсия (отрицание) является одной из основных логических функций, используемых в устройствах цифровой обработки информации.

При двух переменных полный набор состоит из 16 функций, однако в цифровых устройствах используются далеко не все.

Основными логическими функциями двух переменных, используемыми в устройствах цифровой обработки информации являются: дизъюнкция (логическое сложение), конъюнкция (логическое умножение), сумма по модулю 2 (неравнозначность), стрелка Пирса и штрих Шеффера. Условные обозначения логических операций, реализующих указанные выше логические функции одной и двух переменных, приведены в таблице 3.

Название операции	Обозначение	
	Слово	Знак
Инверсия, отрицание, дополнение	НЕ (NOT)	\overline{X} , $-X$
Дизъюнкция, логическое сложение	ИЛИ (OR)	$X_1 \vee X_0$, $X_1 + X_0$
Конъюнкция, логическое умножение	И (AND)	$X_1 \wedge X_0$, $X_1 \& X_0$, $X_1 \cdot X_0$
Неравнозначность, сумма по модулю 2	Исключающее ИЛИ	$X_1 \oplus X_0$
Стрелка Пирса	ИЛИ-НЕ	$\overline{X_1 \vee X_0}$, $\overline{X_1 + X_0}$
Штрих Шеффера	И-НЕ	$\overline{X_1 \cdot X_0}$, $\overline{X_1 \& X_0}$

Таблица 4 – Таблицы истинности основных функций двух переменных

Дизъюнкция			Конъюнкция			Исключающее ИЛИ			Стрелка Пирса		Штрих Шеффера			
X1	X2	Y	X1	X2	Y	X1	X2	Y	X1	X2	Y	X1	X2	Y
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	1	1	0	1	0	0	1	1	0	1	0	0	1	1
1	0	1	1	0	0	1	0	1	1	0	0	1	0	1
1	1	1	1	1	1	1	1	0	1	1	0	1	1	0

Дизъюнкция. В отличие от обычного арифметического или алгебраического суммирования здесь наличие двух единиц даёт в результате единицу. Поэтому при обозначении логического суммирования предпочтение следует отдать знаку (\vee) вместо знака (+) [1].

Первые две строчки таблицы истинности операции дизъюнкции ($x_1=0$) определяют *закон сложения с нулём*: $x \vee 0 = x$, а вторые две строчки ($x_1 = 1$) — *закон сложения с единицей*: $x \vee 1 = 1$.

Конъюнкция. Таблица 4 убедительно показывает тождественность операций обычного и логического умножений. Поэтому в качестве знака логического умножения возможно использование привычного знака обычного умножения в виде точки [1].

Первые две строчки таблицы истинности операции конъюнкции определяют *закон умножения на ноль*: $x \cdot 0 = 0$, а вторые две — *закон умножения на единицу*: $x \cdot 1 = x$.

Исключающее ИЛИ. Под функцией «Исключающее ИЛИ» понимают следующее: единица на выходе появляется тогда, когда только на одном входе присутствует единица. Если единиц на входах две или больше, или если на всех входах нули, то на выходе будет ноль.

Надпись на обозначении элемента ИСКЛЮЧАЮЩЕЕ ИЛИ « $\neq 1$ » (Рисунок 1, г) как раз и обозначает, что выделяется ситуация, когда на входах одна и только одна единица.

Эта операция аналогична операции арифметического суммирования, но, как и другие логические операции, без образования переноса. Поэтому она имеет другое название *сумма по модулю 2* и обозначение \oplus , сходное с обозначением арифметического суммирования.

Стрелка Пирса и штрих Шеффера. Эти операции являются инверсиями операций дизъюнкции и конъюнкции и специального обозначения не имеют.

Рассмотренные логические функции являются простыми или элементарными, так как значение их истинности не зависит от истинности других каких либо функций, а зависит только от независимых переменных, называемых *аргументами*.

В цифровых вычислительных устройствах используются сложные логические функции, которые разрабатываются на основе элементарных функций.

Сложной является логическая функция, значение истинности которой зависит от истинности других функций. Эти функции являются аргументами данной сложной функции.

Например, в сложной логической функции $Y = \overline{X_1 \vee X_2} \cdot \overline{X_3 \& X_4}$ аргументами являются $X_1 \vee X_2$ и $\overline{X_3 \& X_4}$

Логические элементы

1.2.2 Логические элементы

Для реализации логических функций в устройствах цифровой обработки информации используются логические элементы. Условные графические обозначения (УГО) логических элементов, реализующих рассмотренные выше функции, приведены на рисунке 1.

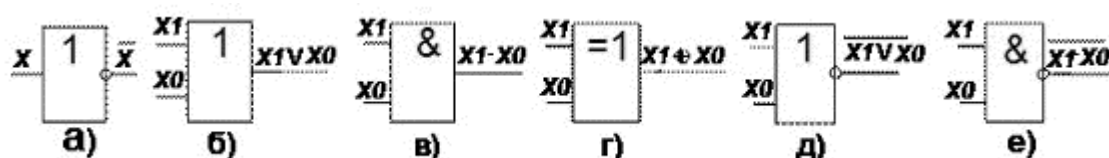


Рисунок 1 – УГО логических элементов: а) Инвертор, б) ИЛИ, в) И, г) Исключающее ИЛИ, д) ИЛИ-НЕ, е) И-НЕ.

Сложные логические функции реализуются на основе простых логических элементов, путём их соответствующего соединения для реализации конкретной аналитической функции. Функциональная схема логического устройства, реализующего сложную

функцию $Y = \overline{X_1 \vee X_2 \cdot X_3 \& X_4}$ приведённую в предыдущем параграфе, приведена на рисунке 2.

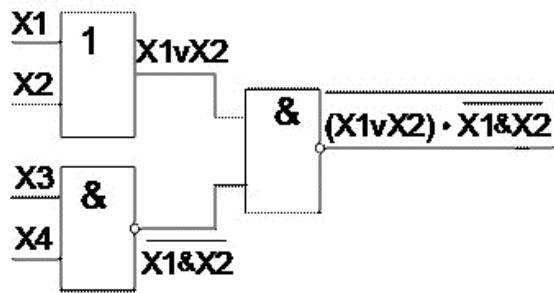


Рисунок 2 – Пример реализации сложной логической функции

Как видно из рисунка 2, логическое уравнение показывает, из каких ЛЭ и какими соединениями можно создать заданное логическое устройство.

Поскольку логическое уравнение и функциональная схема имеют однозначное соответствие, то целесообразно упростить логическую функцию, используя законы алгебры логики и, следовательно, сократить количество или изменить номенклатуру ЛЭ при её реализации.

6. Функционально полные системы логических элементов.

Элементы основного базиса: И, ИЛИ, НЕ.

Схема «И»

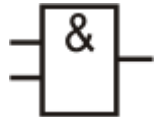


Схема «ИЛИ»

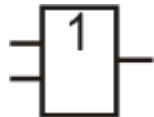
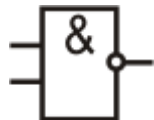


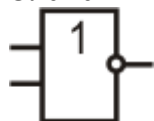
Схема «НЕ» (инвертор)



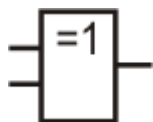
Элемент «И-НЕ»



Элемент «ИЛИ-НЕ»



Элемент «Исключающее ИЛИ»



Триггер- основной элемент электронной памяти.

Триггером называется устройство, имеющее два устойчивых состояния и сохраняющее любое из них сколь угодно долго после снятия внешнего воздействия, вызвавшего переход триггера из одного состояния в другое. Поэтому говорят, что триггер обладает памятью. Триггер можно представить в общем случае состоящим из ячейки памяти и устройства управления (порой весьма сложного), преобразующего входную информацию в комбинацию сигналов, под воздействием которых ячейка памяти принимает одно из двух устойчивых состояний. По способу записи информации триггеры могут быть асинхронными и синхронными. Триггер называют асинхронным, если сам сигнал, несущий информацию, вызывает его переключение. В синхронных (тактируемых) триггерах информация записывается при одновременном воздействии информационного сигнала и синхронизирующего (разрешающего) импульса. Синхронизация может осуществляться импульсом (потенциалом) или перепадом потенциала (фронтом или срезом импульса). В первом случае (статическое управление) сигналы на информационных входах оказывают влияние на состояние триггера в течение всего времени наличия синхроимпульса. Во втором случае (динамическое управление) воздействие информационных сигналов проявляется только в моменты изменения потенциала на входе синхронизации, т.е. при переходе его от 0 к 1 (фронт) или от 1 к 0 (срез). По функциональному признаку различают RS-триггеры, D-триггеры, T-триггеры и JK-триггеры, а также их комбинации. Основу любого триггерного устройства составляет элементарная ячейка памяти на двух инверторах с кольцом положительной обратной связи, называемая защелкой (рис. 12.1).

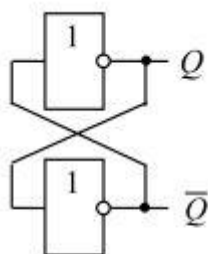


Рис. 12.1—
Элементарная ячейка

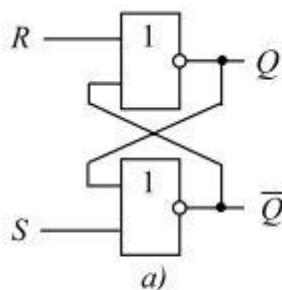
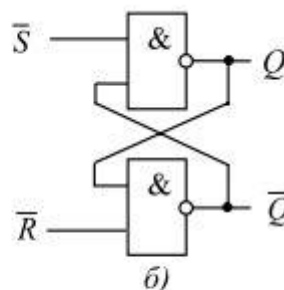


Рис. 12.2—Асинхронный RS-триггер с прямым (а) и инверсным (б) управлением



При подаче питания ЛЭ не могут находиться в активной области передаточной характеристики, так как петлевое усиление много больше единицы, и ячейка переходит в одно из двух устойчивых состояний с инверсными уровнями на выходах Q и \bar{Q} . Это состояние ячейка будет хранить, пока включен источник питания.

Классификация и принцип действия логических триггеров, типы управления. Динамические, двухступенчатые триггеры.

В асинхронном (неактируемом) RS-триггере (рис. 12.2, а), выполненном на логических элементах ИЛИ-НЕ, состояниями описанной выше ячейки памяти можно управлять, подавая логическую 1 либо на вход S (Set) установки в состояние $Q = 1$, либо на вход R (Reset) сброса в состояние $Q = 0$. Если одновременно подать, а затем снять логическую 1 с обоих входов, то состояние триггера после снятия входных сигналов будет неопределенным. Такая комбинация на входах триггера является запрещенной.

Минимальная длительность установочных импульсов должна вдвое превышать время задержки примененных логических элементов, чтобы по петле положительной обратной связи пришло подтверждение о фиксации нового состояния триггера. Аналогичным образом работает асинхронный RS-триггер с инверсным управлением (рис. 12.2, б), выполненный на логических элементах И-НЕ. Установка триггера в единичное и нулевое состояние на выходе Q осуществляется подачей логического нуля на входы S или R . При единичных уровнях на обоих входах триггер не меняет своего состояния (находится в режиме хранения).

Тактируемый (синхронный) RS-триггер (рис. 12.3) может изменить свое состояние только с приходом тактового импульса на вход C . Помехи, действующие на информационных входах R и S между тактовыми импульсами, не влияют на работу триггера. Запрещенным является состояние $CRS = 1$.

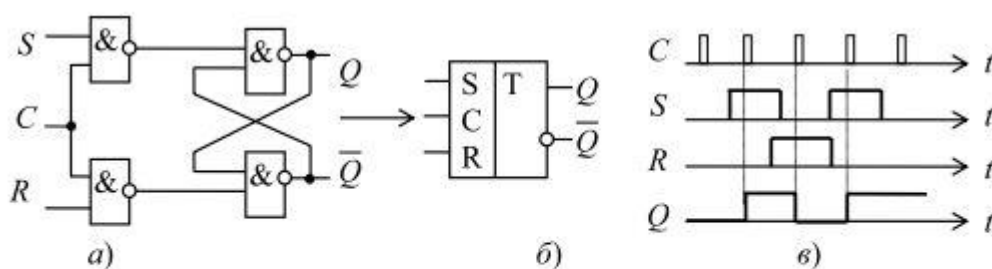


Рис. 12.3–Тактируемый RS-триггер: а) построение триггера на логических элементах И-НЕ; б) обозначение триггера на функциональных схемах; в) временные диаграммы, поясняющие работу триггера

При включении питания состояние триггера остается неопределенным. После совпадения единичных уровней на входах S и C триггер устанавливается в единичное состояние. При совпадении логических единиц на входах R и C триггер сбрасывается в нулевое состояние. D-триггер (от английского Delay – задержка) имеет один информационный (D – Data – данные) и один тактируемый (C – Clock – тактовая последовательность) вход. Такой триггер можно получить из RS-триггера, подав на R -вход инвертированный сигнал с S -входа (рис. 12.4, а). Условное обозначение D-триггера со статическим управлением показано на рис. 12.4, б. Из временных диаграмм, приведенных на рис. 12.4, в можно увидеть, что при $C = 1$ триггер работает как повторитель ($Q = D$), а при переходе сигнала на входе C от логической единицы к логическому нулю триггер запирается и переходит в режим хранения (защелкивает информацию со входа D). В отличие от RS-триггера, D-триггер не имеет запрещенного состояния.

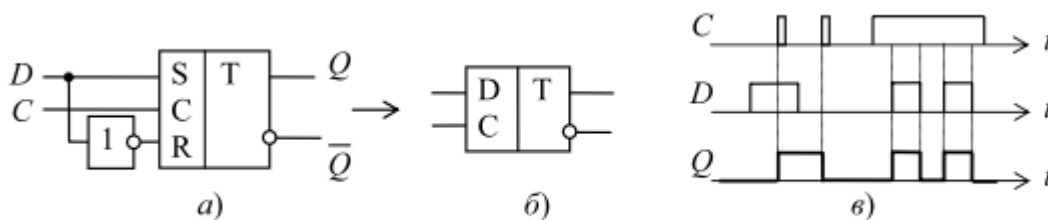


Рис. 12.4–D-триггер со статическим управлением

Статический синхронный триггер реагирует на входные сигналы в течение всего времени, пока тактовый сигнал C равен единице. Часто, однако, требуется триггер, в котором считываемая информация не передавалась бы непосредственно на выход, а появлялась там только тогда, когда все схемы уже заперты. Этим свойством обладают триггеры с динамическим управлением. D-триггер с динамическим управлением, тактируемый фронтом тактового импульса, можно выполнить по двухступенчатой схеме, показанной на рис. 12.5, а. При $C=0$ триггер первой ступени повторяет сигнал D , но триггер второй ступени защелкнут (находится в режиме хранения). При переходе к $C=1$ первый триггер защелкивает на своем выходе информацию со входа D , а второй повторяет ее на своем выходе. Таким образом, двухступенчатый триггер по фронту тактового импульса защелкивает на своем выходе уровень сигнала со входа D и сохраняет его до следующего фронта. Триггеры с динамическим управлением необходимы для построения счетчиков и регистров сдвига. Из двух вариантов УГО динамического входа, приведенных на рис. 12.5, б и в, в дальнейшем будем использовать вариант б, меняя направление косой черты для обозначения входа при тактировании по срезу импульса.

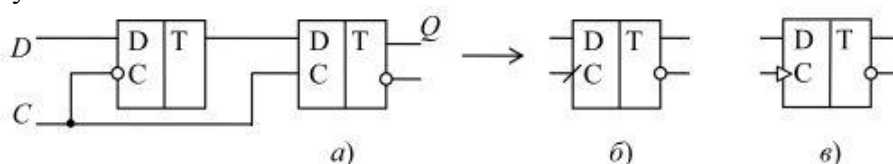


Рис. 12.5–Триггер задержки с динамическим управлением

T-триггер (от английского Toggle – опрокидываться, кувыраться). Он имеет только тактовый вход T и меняет свое состояние на противоположное по фронту или срезу каждого нового тактового импульса (рис. 12.6). На рисунке показано УГО T-триггера и как можно выполнить T-триггер на базе RS- или D-триггеров с динамическим управлением. Каждый раз по фронту сигнала T изменяется уровень напряжения на выходе Q . Частота изменения потенциала на выходе T-триггера в два раза меньше частоты импульсов на его тактовом входе. Это свойство T-триггера используется при построении двоичных счетчиков, а T-триггер называют также счетным триггером.

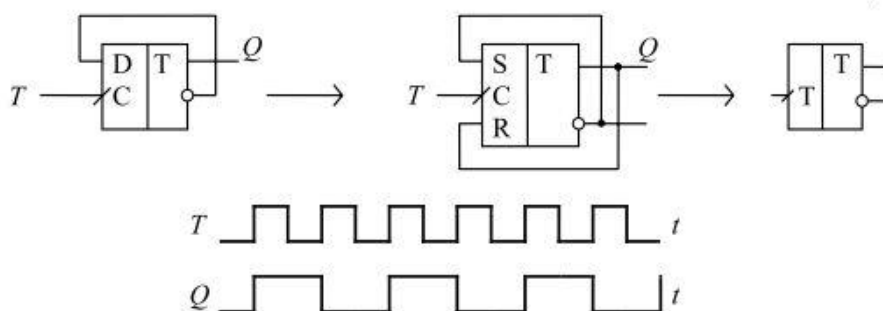


Рис. 12.6–Счетный триггер и временные диаграммы его работы

JK-триггер выполняет наиболее универсальные функции (J – Jerk – резкое движение, толчок; K – Kill – ликвидировать). Он строится на базе RS-триггера с динамическим тактовым входом (рис. 12.7), но, в отличие от него, в JK-триггере устранено запрещенное состояние при $J = K = 1$. При совпадении логических единиц на информационных входах J и K он работает как счетный (режим переключения), т.е. меняет свое состояние на противоположное при каждом новом такте. Логическая 1 на входе J устанавливает триггер в состояние единицы (режим записи 1, установка), логическая 1 на входе K переводит триггер в состояние логического нуля (режим

записи 0, сброс) при наличии тактирования. При наличии логических нулей на входах J и K тактовый импульс не меняет состояние триггера (режим хранения).

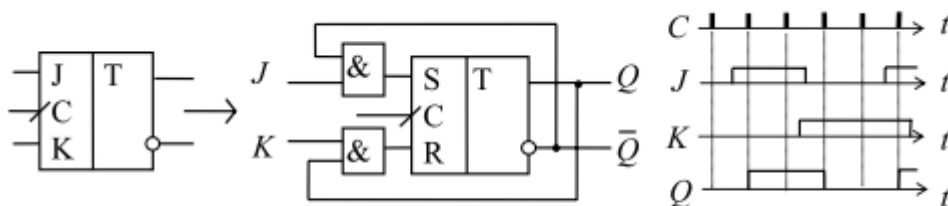


Рис. 12.7–JK-триггер

Работу триггеров можно описать таблицей состояний и функциями переходов (рис. 12.8). Функции переходов синхронных триггеров описывают состояние триггера Q^+ после прихода очередного тактового импульса (его фронта или среза) как логическую функцию входных сигналов и исходного остояния триггера Q до прихода тактового

$S(J)$	$R(K)$	Q^+	
		RS	JK
0	0	\bar{Q}	\bar{Q}
0	1	0	0
1	0	1	1
1	1	x	\bar{Q}

$Q^+ = D$ – для D-триггера;
 $Q^+ = \bar{Q}J + Q\bar{K}$ – для JK-триггера;
 $Q^+ = Q \oplus T$ – для T-триггера;
 $Q^+ = S + Q\bar{R}$, $RS = 0$ – для RS-триггера.

Рис. 12.8–Таблица состояний RS- и JK-триггеров и функции переходов триггеров

импульса.

Примеры микросхем триггеров приведены на рис. 12.9. Наряду с тактируемыми, микросхемы K555M2 и K555TB6 имеют дополнительные входы для предварительной установки в единичное или нулевое состояние (комбинированные триггеры). При этом входы установки R и S являются асинхронными (приоритетными). Предельная частота функционирования триггеров K555TM2 составляет 25 МГц, K555TB6 – 30 МГц. Микросхема K555TP2 содержит четыре асинхронных RS-триггера, причем два из них имеют по два объединенных логикой ИЛИ входа S (подача логического 0 на любой из них устанавливает триггер в единичное состояние).

Последовательные устройства: счетчики, регистры. Комбинационные устройства: сумматоры, шифраторы, преобразователи кодов, мультиплексоры. Принципы построения оперативных запоминающих устройств(ОЗУ), ячейка памяти. Запись, считывание информации. Нарращивание объема памяти. Постоянные запоминающие устройства(ПЗУ).

Счетчики

Цифровое устройство, циклически меняющее свои состояния под действием импульсов, подаваемых на один вход, называется счетчиком. Количество тактов, через которое повторяется исходное состояние счетчика, называют коэффициентом пересчета (модулем счета) Ксч. Счетчики строят из цепочек триггеров с динамическим управлением.

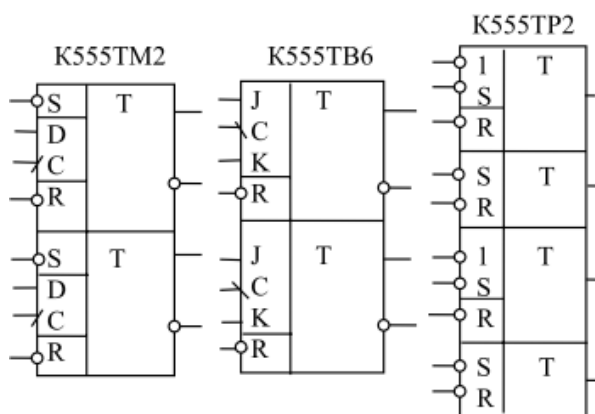


Рис. 12.9–Интегральные микросхемы триггеров

По коэффициенту пересчета различают счетчики двоичные ($K_{сч} = 2^n$, где n – разрядность счетчика), десятичные ($K_{сч} = 10n$, где n – количество декад счетчика), с произвольным постоянным $K_{сч}$, с изменяемым $K_{сч}$ (программируемые). По направлению счета счетчики делятся на суммирующие, вычитающие, реверсивные. По способу организации внутренних связей между триггерами счетчики могут быть асинхронными (с последовательным переносом) и синхронными (с параллельным переносом). Синхронные счетчики обладают большим быстродействием. Асинхронные двоичные счетчики строят из цепочки счетных триггеров, соединяя выход предыдущего со входом последующего. Такой счетчик реализован на микросхеме K155IE5 (рис. 12.10).

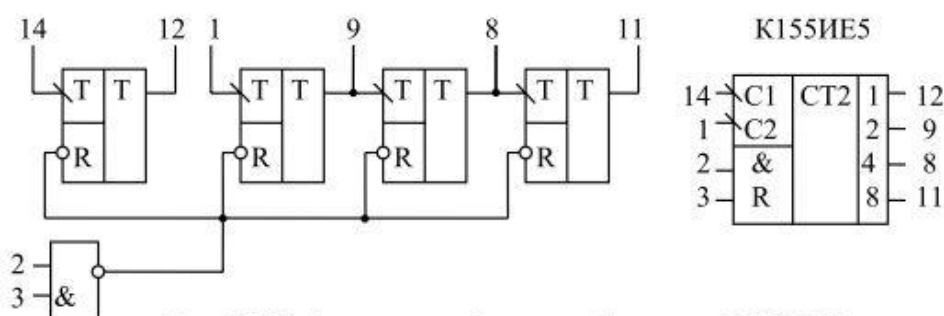


Рис. 12.10–Асинхронный двоичный счетчик K155IE5

При совпадении логических единиц на выходах 2 и 3 счетные триггеры сбрасываются в нулевое состояние. При замыкании выводов 1 и 12 получаем четырехразрядный двоичный счетчик. Частота счетных импульсов последовательно делится в два раза каждым триггером. При этом счетчик проходит 16 состояний (с 0 по 15), каждому из которых соответствует четырехразрядный код на выходах с весовыми коэффициентами 8, 4, 2, 1. Микросхема K155IE2 состоит из счетного триггера (вход – C1, выход – Q1) и

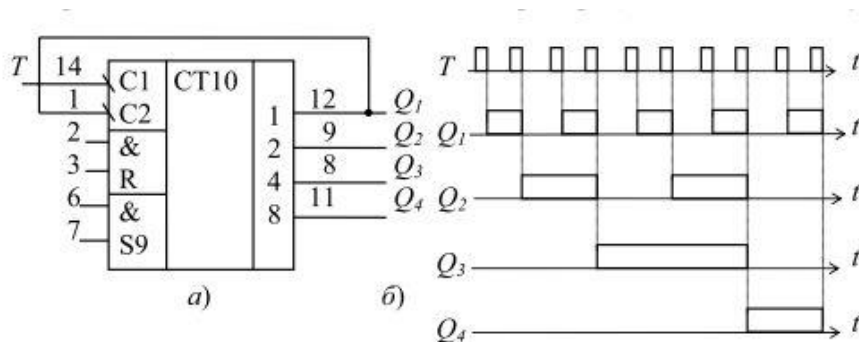


Рис. 12.11–Асинхронный двоично-десятичный счетчик К155ИЕ2

счетчика с коэффициентом пересчета $K_{сч} = 5$ (вход – C2, выходы – Q2, Q3, Q4). Если их соединить между собой так, как это показано на рис. 12.11, а, то получится двоично-десятичный счетчик, временные диаграммы работы которого приведены на рис. 12.11, б. При поступлении десятого импульса (по его срезу) кодовая комбинация $Q_4 Q_3 Q_2 Q_1 = 1001$ сменяется комбинацией 0000, и далее цикл из десяти состояний счетчика (с 0 по 9) периодически повторяется. При совпадении логических единиц на входах R счетчик устанавливается в состояние «0», при совпадении логических единиц на входах S9 – в состояние «9». В синхронном двоичном счетчике (рис. 12.12) счетный импульс T воздействует сразу на все триггеры. Первый триггер работает как счетный. Каждый последующий меняет свое состояние на противоположное, когда все предыдущие находятся в единичном состоянии. Устройство реализует алгоритм работы суммирующего двоичного счетчика с $K_{сч} = 16$.

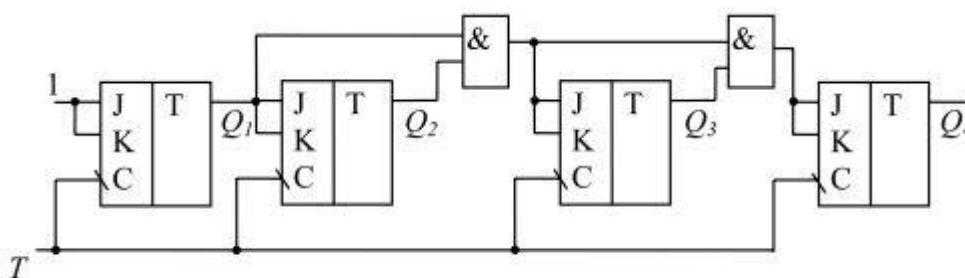


Рис. 12.12–Четырехразрядный синхронный двоичный счетчик

Реверсивные счетчики обладают универсальными возможностями. На рис. 12.13 показаны ИМС синхронных реверсивных десятичного (К555ИЕ6) и двоичного (К555ИЕ7) счетчиков. Уровнем логического нуля на входе L в счетчик записывается четырехразрядный код со входов предустановки 1, 2, 4, 8 (параллельная загрузка). Эта возможность позволяет строить на таких микросхемах счетчики и делители частоты с изменяемым $K_{сч}$. Уровнем логической единицы на входе R счетчик сбрасывается в нулевое состояние. Вход R имеет приоритет по отношению ко входу L. При подаче импульсов на суммирующий вход +1 на вычитающем –1 должен быть высокий уровень и наоборот.

Регистры

Регистры представляют собой цепочки триггеров и предназначены для записи, хранения, сдвига и считывания из них двоичной информации (полубайта, байта и т. д.). Различают регистры сдвиговые (со сдвигом вправо, влево и реверсивные), с параллельной загрузкой, универсальные, кольцевые и файловые. Регистры сдвига строятся на D-триггерах с динамическим управлением. Тактовые входы объединяются, вход D первого триггера служит для приема информации, а входы D последующих подключаются к прямым выходам предыдущих (рис. 12.17).

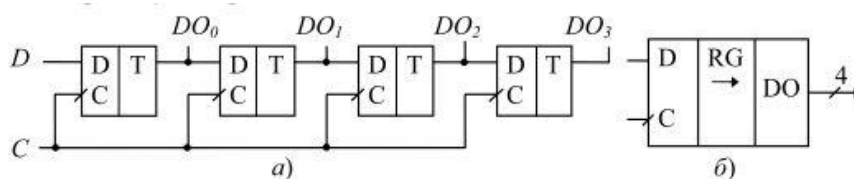


Рис. 12.17–Построение регистра сдвига на D-триггерах (а) и его УГО (б)

За четыре такта С (четыре синхроимпульса) последовательную информацию со входа D можно преобразовать в параллельную на выходах DO0 - DO3. Регистры с параллельной загрузкой также чаще строят на D-триггерах, объединяя их тактовые входы. Микросхемы регистров памяти могут тактироваться потенциалом (K580ИР82) или фронтом (K555ИР27) тактового импульса (рис. 12.18). Обязательным условием при записи данных в регистр является их фиксация до поступления такта. Хранимые данные с выхода первой микросхемы читаются при логическом нуле на входе OE (разрешение выхода) При логической единице на входе OE выходы микросхемы находятся в высокоимпедансном состоянии. Запись информации во вторую микросхему происходит по фронту тактового импульса только при логическом нуле на входе L (Load – разрешение загрузки). При L = 1 имеет место режим хранения данных в регистре.

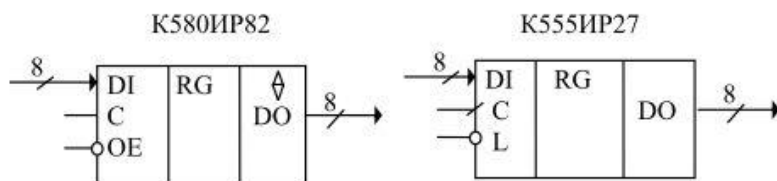


Рис. 12.18–Микросхемы параллельных регистров

Универсальные регистры. Микросхема K155ИР13 (рис. 12.19) является примером универсального регистра. Режим ее работы задается уровнями сигналов на входах SR и SL (см. таблицу состояний регистра). При сдвиге вправо сигнал со входа DR переписывается в младший разряд DO0 по фронту каждого тактового импульса, а старая информация выхода DO7 теряется. При сдвиге влево информация со входа DL записывается на выход DO7, а информация с выхода DO0 теряется. При SL = SR = 1 микросхема работает как параллельный регистр, запись информации в который со входов DI происходит по фронту тактового импульса. При подаче 0 на вход R все триггеры регистра сбрасываются в нулевое состояние.

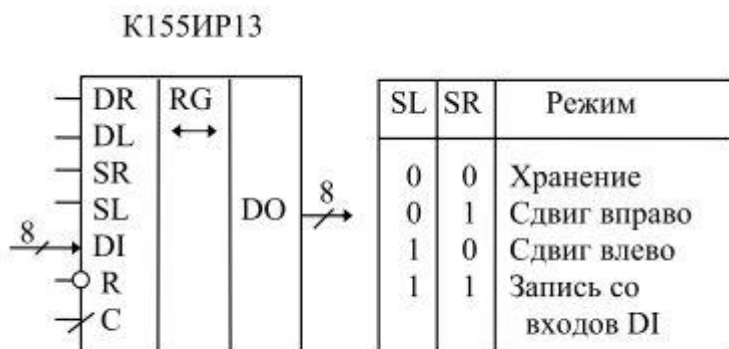
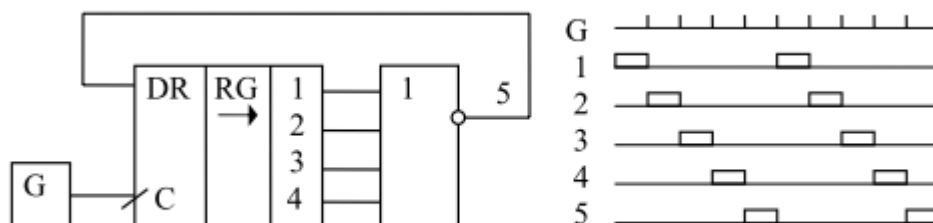


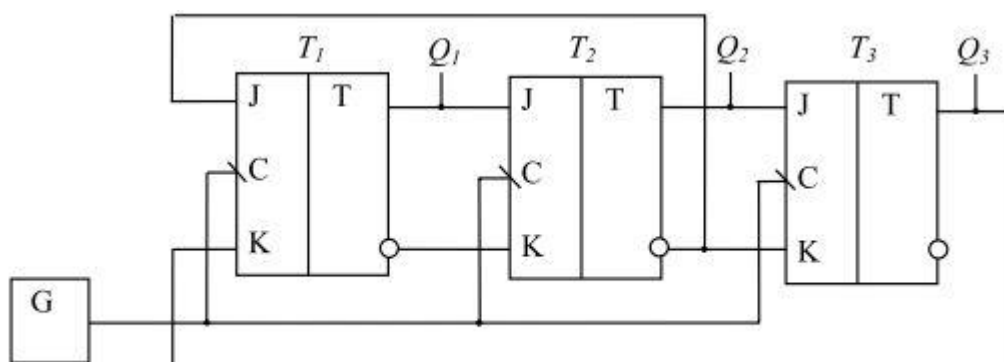
Рис. 12.19–Универсальный регистр

До тех пор, пока на управляющем входе U поддерживается уровень логической 1, $DR = D_{вх}$, обратная связь не действует. За первые n тактов запоминается n -разрядный входной код $D_{вх}$. Если теперь подать $U = 0$, то $DR = Q_n$ и выведенный из регистра код поразрядно поступает на вход. После n тактовых импульсов регистр сдвига опять находится в исходном состоянии. Следовательно, логическое состояние на входе управления определяет, вводится ли новая информация или в регистре циркулирует старая. Помимо основного назначения, регистры имеют другие многочисленные применения. Рассмотрим только некоторые из них.

Кольцевой счетчик. На регистрах сдвига реализуются самые разнообразные счетчики. Простейшим из них является кольцевой счетчик.



Синхронный счетчик. Проанализируем работу другого счетчика на регистре сдвига, в котором используется перекрестная обратная связь (рис. 12.22).



Исследуемое устройство представляет собой синхронный счетчик на

регистре сдвига, построенном на трех JK-триггерах. Состояние первого триггера после подачи очередного среза счетных импульсов генератора G зависит от сигналов обратной связи, поступающих на его информационные входы J и K с выходов Q2 и Q3. Триггер T2 повторяет состояние триггера T1 на предыдущем такте, а триггер T3 – состояние триггера T2.

Удобно анализ работы устройства провести с помощью таблицы состояний, предположив, что первоначально триггеры были сброшены в нулевое состояние ($Q1 = Q2 = Q3 = 0$), а затем на счетный вход поступает очередной k-тый импульс (рис. 12.23). До подачи первого импульса на информационных входах триггера T1: $J = 1$, $K = 0$. После первого импульса T1 переходит в единичное состояние, а логические нули с выходов Q1, Q2 переписываются на выходы Q2, Q3. Уровни сигналов на информационных входах T1 не изменились. Поэтому после второго импульса он снова будет находиться в состоянии $Q1 = 1$. Теперь уже и $Q2 = 1$. На информационных входах T1 устанавливаются уровни $J = K = 0$.

Сумматоры

Двоичный сумматор (SM) (рис. 11.12, а) служит для формирования арифметической суммы n-разрядных двоичных чисел A и B (рис. 11.12, б). Результатом сложения (при $n = 4$) является четырехразрядная сумма S и выход переноса P, который можно рассматривать как пятый разряд суммы.

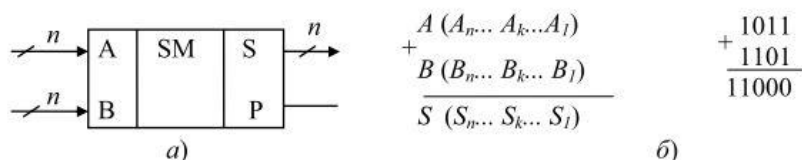


Рис. 11.12–Двоичный сумматор

Полусумматор (HS) служит для сложения битов младших разрядов двух двоичных чисел (его можно реализовать на ЛЭ по таблице истинности, представленной на рис. 11.13).

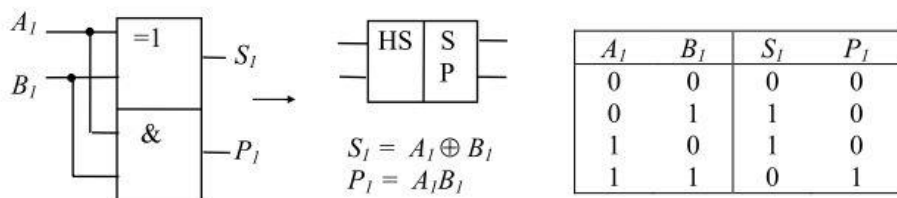


Рис. 11.13–Синтез полусумматора на логических элементах

Полный одноразрядный сумматор суммирует биты соответствующих разрядов двух двоичных чисел и вырабатывает перенос в следующий разряд.

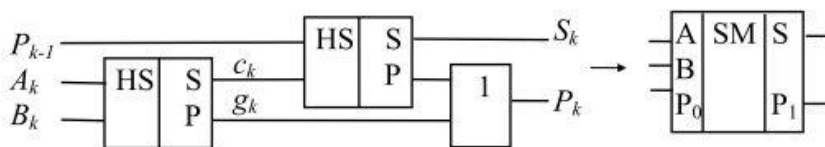


Рис. 11.14–Полный одноразрядный сумматор

Полный одноразрядный сумматор можно построить из двух полусумматоров HS и логического элемента ИЛИ (рис. 11.14).

В корпусе микросхемы K555ИМ6 четыре полных одноразрядных сумматора объединены в схему четырехразрядного сумматора (рис. 11.15).

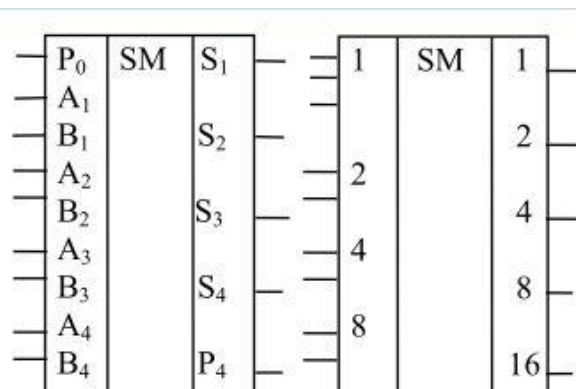


Рис. 11.15—Четырехразрядный последовательный сумматор K555ИМ6

Сигнал переноса последовательно передается с выхода предыдущего разряда сумматора на вход переноса следующего разряда. В дальнейшем будем использовать более простое и наглядное условное графическое обозначение сумматора, приведенное справа. В дополнительных полях микросхемы показаны весовые коэффициенты разрядов входа и выхода сумматора. Время выполнения операции в сумматоре на рис. 11.15 намного больше времени сложения в одноразрядном сумматоре, так как в каждый следующий разряд единица переноса попадает, проходя все более длинную цепочку логических элементов. Чтобы уменьшить время выполнения операции сложения многоразрядных чисел, используют схемы параллельного переноса. При этом сигналы переноса во всех разрядах одновременно вычисляются по значениям входных переменных в данном разряде. Для сигнала переноса из любого k -го разряда справедливо соотношение (см. обозначения на рис. 11.14)

$$p_k = a_k b_k + (a_k \oplus b_k) p_{k-1}, \quad (11.2)$$

$\Downarrow \qquad \Downarrow$
 $g_k \qquad c_k$

где g_k - функция генерации переноса;
 c_k - функция распространения переноса.

Пользуясь рекуррентным выражением (11.2), можно вывести следующие формулы для вычисления сигналов переноса в четырехразрядном сумматоре:

Шифраторы

Шифратор (кодер) — (англ. *encoder*) логическое устройство, выполняющее логическую функцию (операцию) - преобразование позиционного n -разрядного кода в m -разрядный двоичный, троичный или k -ичный код.

Двоичный шифратор выполняет логическую функцию преобразования унарно n -ичного однозначного кода в двоичный. При подаче сигнала на один из n входов (обязательно на один, не более) на выходе появляется двоичный код номера активного входа.

Если количество входов настолько велико, что в шифраторе используются все возможные комбинации сигналов на выходе, то такой шифратор называется полным, если не все, то неполным.

Шифратор (кодер) преобразует единичный сигнал на одном из входов в n -разрядный двоичный код. Наибольшее применение он находит в устройствах ввода информации (пультах управления) для преобразования десятичных чисел в двоичную систему счисления. Предположим, на пульте десять клавиш с гравировкой от 0 до 9. При нажатии любой из них на вход шифратора подается единичный сигнал (X_0, \dots, X_9). На выходе шифратора должен появиться двоичный код (Y_0, \dots, Y_9) этого десятичного числа. Как

видно из таблицы истинности (таблица В), в этом случае нужен преобразователь с десятью входами и четырьмя выходами.

Входы	Выходы			
X	Y ₃	Y ₂	Y ₁	Y ₀
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

На выходе Y₀ единица должна появиться при нажатии любой нечетной клавиши X₁, X₃, X₅, X₇, X₉, т.е. $Y_0=X_1+X_3+X_5+X_7+X_9$.

$$Y_1=X_2+X_3+X_6+X_7; \; Y_2=X_4+X_5+X_6+X_7; \; Y_3=X_8+X_9.$$

Следовательно, для реализации шифратора понадобится четыре элемента ИЛИ: пятиходовый, два четырех входовых и двухвходовый (рис.13.1.)

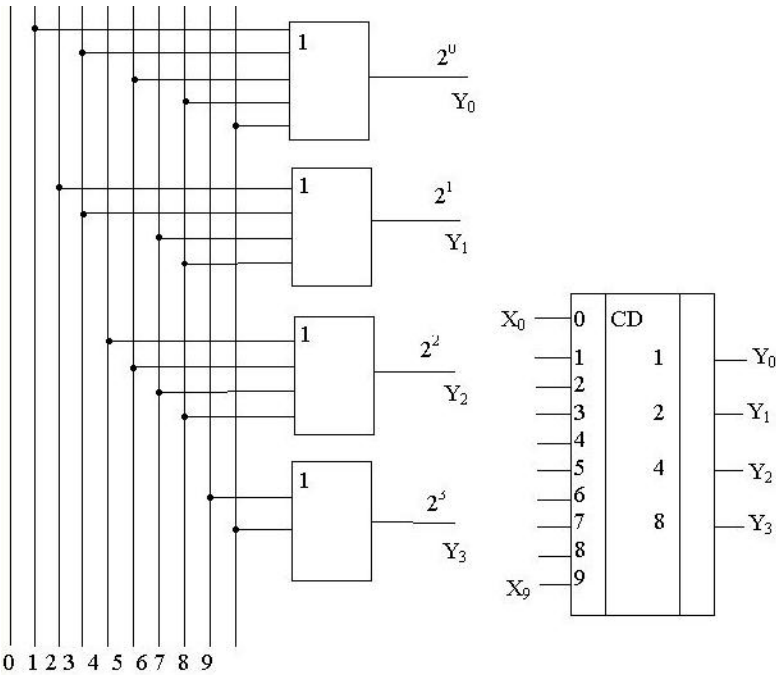


рис.13.1.

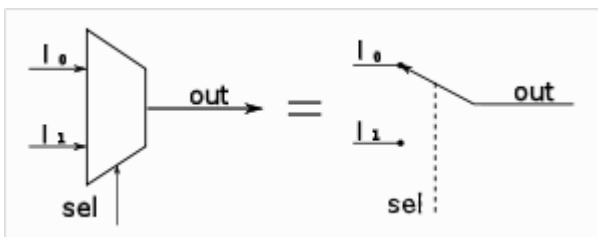
Мультиплексоры

Мультиплексор — устройство, имеющее несколько сигнальных входов, один или более управляющих входов и один выход. Мультиплексор позволяет передать сигнал с *одного*

из входов на выход; при этом выбор желаемого входа осуществляется подачей соответствующей комбинации управляющих сигналов.

[Аналоговые](#) и [цифровые](#) мультиплексоры значительно различаются по принципу работы. Первые электрически соединяют выбранный вход с выходом (при этом [сопротивление](#) между ними невелико — порядка единиц/десятков [Ом](#)). Вторые же не образуют прямого электрического соединения между выбранным входом и выходом, а лишь «копируют» на выход логический уровень ('0' или '1') с выбранного входа. Аналоговые мультиплексоры иногда называют **ключами**.

Устройство, противоположное мультиплексору по своей функции, называется [демультиплексором](#).



Схематически мультиплексор можно изобразить в виде [коммутатора](#), обеспечивающего подключение одного из нескольких входов (их называют **информационными**) к одному выходу устройства. Коммутатор обслуживает управляющая схема, в которой имеются адресные входы и, как правило, разрешающие ([стробирующие](#)).

Сигналы на **адресных** входах определяют, какой конкретно информационный канал подключен к выходу. Если между числом информационных выходов и числом адресных входов действует соотношение $n = 2^m$ то такой мультиплексор называют полным. Если

$n < 2^m$ то мультиплексор называют неполным.

Разрешающие входы используют для расширения функциональных возможностей мультиплексора. Они используются для наращивания разрядности мультиплексора, синхронизации его работы с работой других узлов. Сигналы на разрешающих входах могут разрешать, а могут и запрещать подключение определенного входа к выходу, то есть могут блокировать действие всего устройства.

В качестве **управляющей схемы** обычно используется [дешифратор](#). В цифровых мультиплексорах логические элементы коммутатора и дешифратора обычно объединяются.

Обобщённая схема мультиплексора



Обобщённая схема мультиплексора.

Входные логические сигналы X_i поступают на входы коммутатора и через коммутатор передаются на выход Y . На вход управляющей схемы подаются адресные сигналы A_k (от англ. *Address*). Мультиплексор также может иметь дополнительный управляющий вход E

(от англ. *Enable*), который разрешает или запрещает прохождение входного сигнала на выход *Y*.

Кроме этого, некоторые мультиплексоры могут иметь выход с тремя состояниями: два логических состояния 0 и 1, и третье состояние — отключённый выход (выходное сопротивление равно бесконечности, высокоимпедансное Z-состояние). Перевод мультиплексора в третье состояние производится снятием управляющего сигнала ОЕ (от [англ. Output Enable](#)). Мультиплексоры могут использоваться в делителях частоты, [триггерных](#) устройствах, сдвигающих устройствах и др. Мультиплексоры часто используют для преобразования параллельного двоичного кода в последовательный. Для такого преобразования достаточно подать на информационные входы мультиплексора параллельный двоичный код, а сигналы на адресные входы подавать в такой последовательности, чтобы к выходу поочередно подключались входы, начиная с первого и заканчивая последним.

Принципы построения оперативных запоминающих устройств(ОЗУ)

Одним из ведущих направлений развития современной микроэлектроники элементной базы являются большие интегральные микросхемы памяти, которые служат основой для построения запоминающих устройств в аппаратуре различного назначения. Наиболее широкое применение эти микросхемы нашли в ЭВМ, в которых память представляет собой функциональную часть, предназначенную для записи, хранения, выдачи команд и обрабатываемых данных. Комплекс механических средств, реализующих функцию памяти, называют запоминающим устройством.

Оперативные запоминающие устройства — (ОЗУ — их обозначают английской аббревиатурой RAM) подразделяются на статические – SRAM (Static RAM) и динамические – DRAM (Dynamic RAM).

В статических ОЗУ запоминающая ячейка представляет собой триггер на биполярных или полевых транзисторах, что определяет потенциальный характер управляющих сигналов и возможность считывания информации без ее разрушения.

В динамических ОЗУ элементом памяти является емкость (например, затвора полевого транзистора), что требует периодического восстановления (регенерации) записанной информации в процессе ее хранения. ОЗУ динамического типа позволяют реализовать большой объем памяти, но они сложнее в использовании, так как необходимо наличие специальной схемы управления режимами работы. Динамические ОЗУ характеризуются наибольшей информационной емкостью и невысокой стоимостью, но имеют большее энергопотребление и меньшее быстродействие. В современных динамических ОЗУ имеются встроенные системы регенерации и синхронизации. Такие ОЗУ по внешним сигналам управления не отличаются от статических ОЗУ.

Плотность упаковки динамических элементов памяти в несколько раз выше, чем статических.

Оперативные запоминающие устройства RAM делятся на статические – SRAM (Static RAM) и динамические – DRAM (Dynamic RAM). В статических ОЗУ запоминающими элементами являются триггеры. В динамических ОЗУ данные хранят в виде зарядов конденсаторов, образуемых элементами МОП-структур. Запоминающие конденсаторы разряжаются, поэтому каждые несколько миллисекунд данные должны регенерироваться. Плотность упаковки динамических элементов памяти в несколько раз выше, чем статических. Динамические ОЗУ характеризуются наибольшей информационной емкостью и невысокой стоимостью, но имеют большее энергопотребление и меньшее быстродействие.

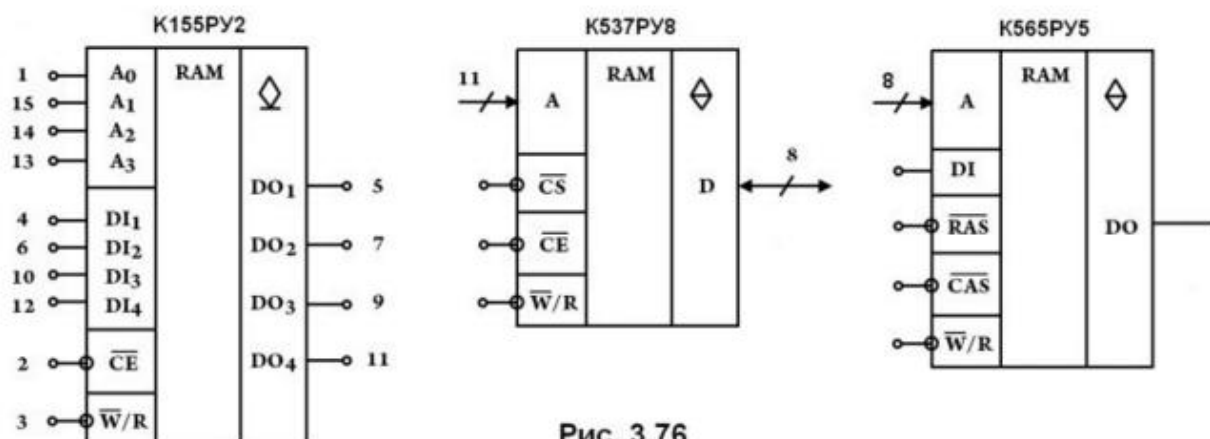


Рис. 3.76

Микросхема K155PY2 — это статическое ОЗУ с открытым коллекторным выходом — выполнена на основе ТТЛ-структур емкостью 64 бит. Имеет структуру 16×4, т. е. может хранить 16 слов длиной 4 разряда каждое.

Микросхема K537PY8 — это статическое ОЗУ объемом 2 Кбайта, выполнена на основе структур КМОП, по входу и выходу совместима с ТТЛ-структурами. Имеет двунаправленную 8-разрядную шину данных, которая используется и для записи, и для считывания информации.

Микросхема K565PY5 — это динамическое ОЗУ на основе n-МОП-структур, по входам и выходам совместима с ТТЛ-структурами, имеет организацию 64Kx1. Шина адреса работает в мультиплексном режиме. Вначале на ней выставляются адреса строк, которые запоминаются во внутреннем регистре по спаду сигнала RAS. Затем выставляются адреса столбцов, которые запоминаются по спаду сигнала CAS.

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 (read only memory - память доступная только для чтения).

Постоянная память типа ROM имеет следующие разновидности:

1. Программируемые при изготовлении ИМС с помощью одной из масок. Эта память типа ПЗУМ (ПЗУ масочные) – ROM(M).
2. Память, программируемая пользователем (ППЗУ – программируемые ПЗУ):
 - PROM – содержимое записывается однократно в память,
 - EPROM и EEPROM – содержимое может быть заменено путем стирания информации и записи новой. В EPROM – стирание происходит путем облучения кристалла ультрафиолетовыми лучами (РПЗУ-УФ – репрограммируемые ПЗУ с УФ стиранием). В EEPROM – стирание происходит электрическими сигналами (РПЗУ-ЭС – репрограммируемые ПЗУ с электрическим стиранием).

Запись данных для EPROM и EEPROM производится электрическими сигналами

Схема такого постоянного запоминающего устройства (ПЗУ) приведена на рисунке 1.

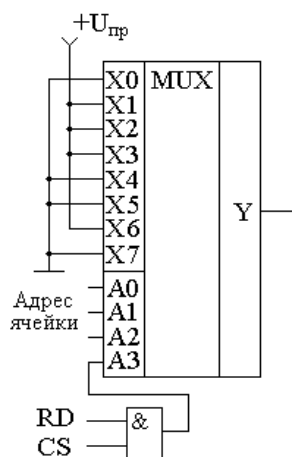


Рисунок 1. Схема постоянного запоминающего устройства (ПЗУ), построенная на мультиплексоре.

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 2

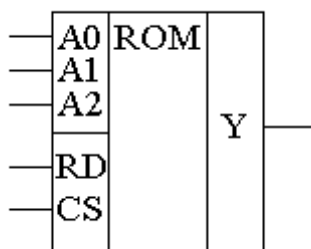


Рисунок 2. Обозначение постоянного запоминающего устройства на принципиальных схемах.

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 3.

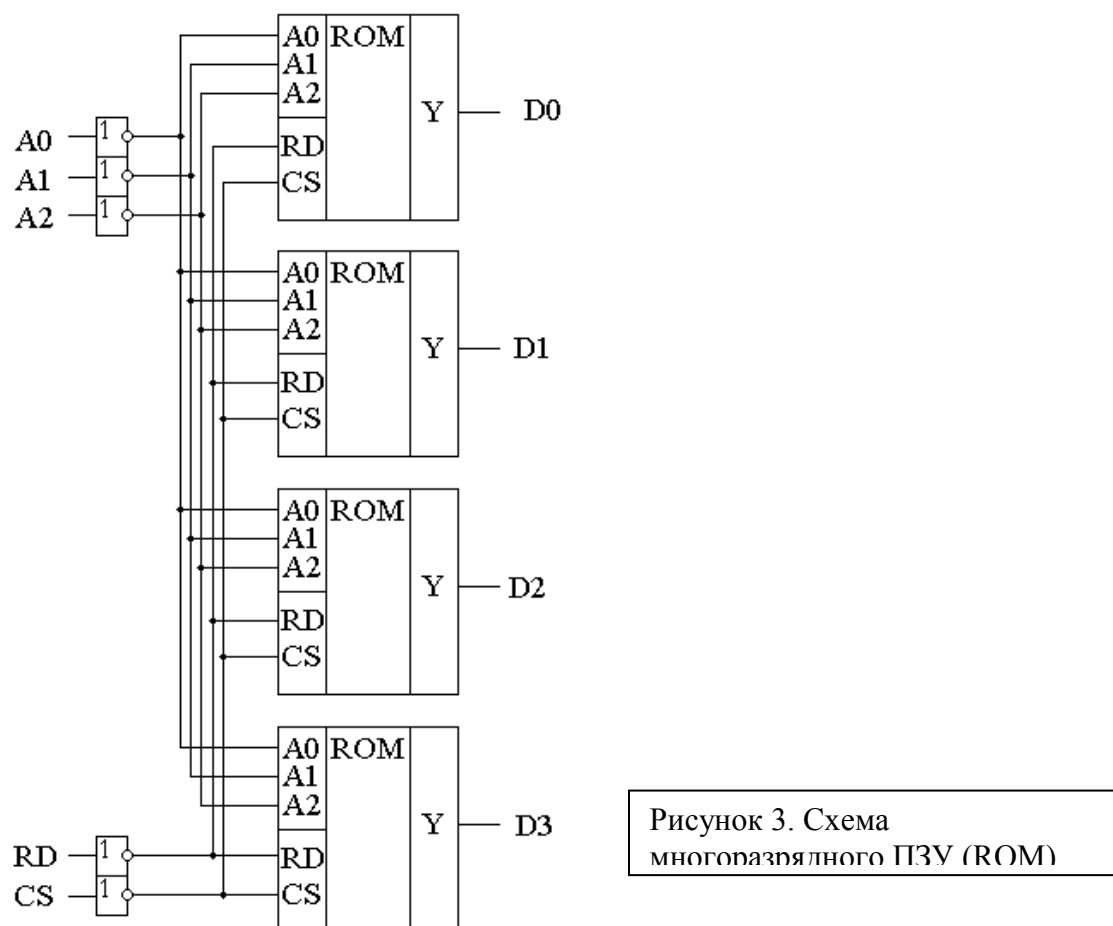


Рисунок 3. Схема
многоадресного ПЗУ (ROM)

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:

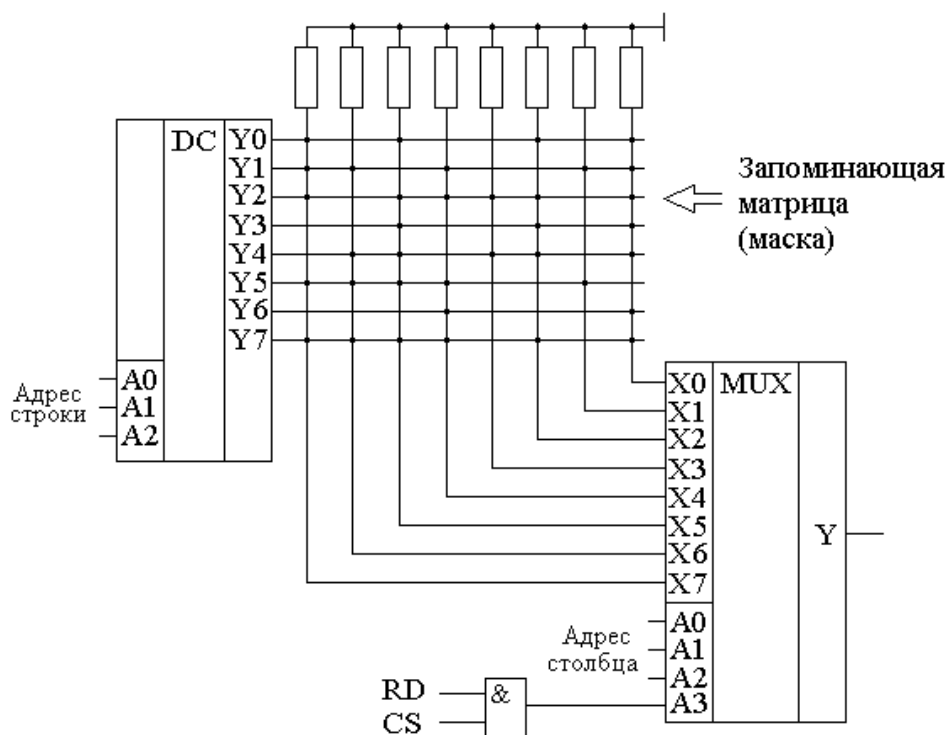


Рисунок 4. Схема масочного постоянного запоминающего устройства (ROM).

Масочные ПЗУ изображаются на принципиальных схемах как показано на рисунке 5. 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.

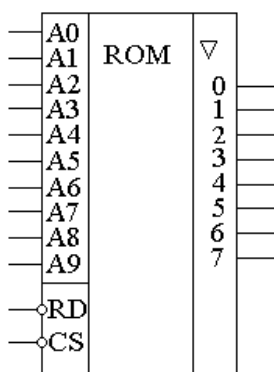


Рисунок 5. Условно-графическое обозначение масочного ПЗУ (ROM) на принципиальных схемах.

Программирование масочного ПЗУ производится на заводе изготовителя, что очень неудобно для мелких и средних серий производства, не говоря уже о стадии разработки устройства. Естественно, что для крупносерийного производства масочные ПЗУ являются самым дешевым видом ПЗУ, и поэтому широко применяются в настоящее время. Для мелких и средних серий производства радиоаппаратуры были разработаны микросхемы, которые можно программировать в специальных устройствах - программаторах. В этих ПЗУ постоянное соединение проводников в запоминающей матрице заменяется плавкими

перемычками, изготовленными из поликристаллического кремния. При производстве ПЗУ изготавливаются все перемычки, что эквивалентно записи во все ячейки памяти ПЗУ логических единиц. В процессе программирования ПЗУ на выводы питания и выходы микросхемы подаётся повышенное питание. При этом, если на выход ПЗУ подаётся напряжение питания (логическая единица), то через перемычку ток протекать не будет и перемычка останется неповрежденной. Если же на выход ПЗУ подать низкий уровень напряжения (присоединить к корпусу), то через перемычку запоминающей матрицы будет протекать ток, который испарит ее и при последующем считывании информации из этой ячейки ПЗУ будет считываться логический ноль.

Такие микросхемы называются **программируемыми ПЗУ (ППЗУ)** или **PROM** и изображаются на принципиальных схемах как показано на рисунке 6. В качестве примера ППЗУ можно назвать микросхемы 155PE3, 556PT4, 556PT8 и другие.

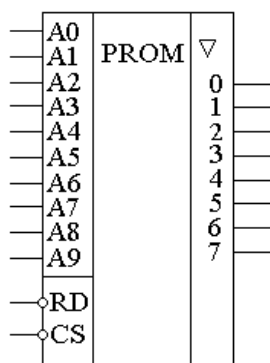


Рисунок 6. Условно-графическое обозначение программируемого постоянного запоминающего устройства (PROM) на принципиальных схемах.

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.

ПЗУ с ультрафиолетовым стиранием строится на основе запоминающей матрицы построенной на ячейках памяти, внутреннее устройство которой приведено на следующем рисунке:

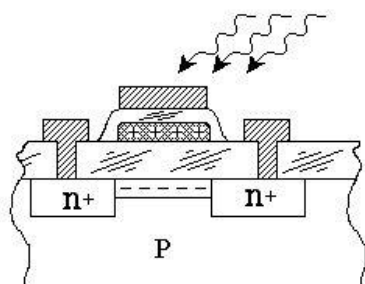


Рисунок 7. Запоминающая ячейка постоянного запоминающего устройства (ПЗУ) с ультрафиолетовым и электрическим стиранием.

Ячейка представляет собой МОП транзистор, в котором затвор выполняется из поликристаллического кремния. Затем в процессе изготовления микросхемы этот затвор окисляется и в результате он будет окружен оксидом кремния - диэлектриком с прекрасными изолирующими свойствами. В описанной ячейке при полностью стертом ПЗУ, заряда в плавающем затворе нет, и поэтому транзистор ток не проводит. При программировании ПЗУ, на второй затвор, находящийся над плавающим затвором, подаётся высокое напряжение и в плавающий затвор за счет туннельного эффекта

индуцируются заряды. После снятия программирующего напряжения индуцированный заряд остаётся на плавающем затворе, и, следовательно, транзистор остаётся в проводящем состоянии. Заряд на плавающем затворе подобной ячейки может храниться десятки лет.

Структурная схема описанного постоянного запоминающего устройства не отличается от описанного ранее масочного ПЗУ. Единственное отличие - вместо плавкой перемычки используется описанная выше ячейка. Такой вид ПЗУ называется репрограммируемыми постоянными запоминающими устройствами (РПЗУ) или EPROM. В РПЗУ стирание ранее записанной информации осуществляется ультрафиолетовым излучением. Для того, чтобы этот свет мог беспрепятственно проходить к полупроводниковому кристаллу, в корпус микросхемы ПЗУ встраивается окошко из кварцевого стекла.

При облучении микросхемы РПЗУ, изолирующие свойства оксида кремния теряются, накопленный заряд из плавающего затвора стекает в объем полупроводника, и транзистор запоминающей ячейки переходит в закрытое состояние. Время стирания микросхемы РПЗУ колеблется в пределах 10 - 30 минут.

Количество циклов записи - стирания микросхем EPROM находится в диапазоне от 10 до 100 раз, после чего микросхема РПЗУ выходит из строя. Это связано с разрушающим воздействием ультрафиолетового излучения на оксид кремния. В качестве примера микросхем EPROM можно назвать микросхемы 573 серии российского производства, микросхемы серий 27сXXX зарубежного производства. В РПЗУ чаще всего хранятся программы BIOS универсальных компьютеров. РПЗУ изображаются на принципиальных схемах как показано на рисунке 8.

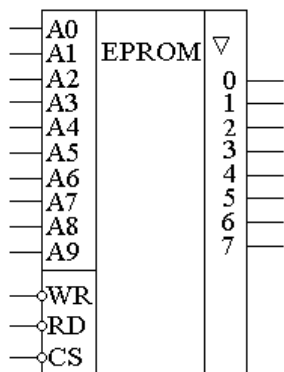


Рисунок 8. Условно-графическое обозначение РПЗУ (EPROM) на принципиальных схемах.

Так как корпуса с кварцевым окошком очень дороги, а также малое количество циклов записи - стирания привели к поиску способов стирания информации из РПЗУ электрическим способом. На этом пути встретилось много трудностей, которые к настоящему времени практически решены. Сейчас достаточно широко распространены микросхемы с электрическим стиранием информации. В качестве запоминающей ячейки в них используются такие же ячейки как и в РПЗУ, но они стираются электрическим потенциалом, поэтому количество циклов записи - стирания для этих микросхем достигает 1000000 раз. Время стирания ячейки памяти в таких ПЗУ уменьшается до 10 мс. Схема управления для электрически стираемых программируемых ПЗУ получилась сложная, поэтому наметилось два направления развития этих микросхем:

1. ЕСППЗУ (EEPROM) - электрически стираемое программируемое постоянное запоминающее устройство
2. FLASH-ПЗУ

Электрически стираемые ПЗУ (EEPROM) дороже и меньше по объему, но зато позволяют перезаписывать каждую ячейку памяти отдельно. В результате эти микросхемы обладают максимальным количеством циклов записи - стирания. Область применения электрически стираемых ПЗУ - хранение данных, которые не должны стираться при выключении питания. К таким микросхемам относятся отечественные микросхемы 573PP3, 558PP3 и зарубежные микросхемы EEPROM серии 28сХХ. Электрически стираемые ПЗУ обозначаются на принципиальных схемах как показано на рисунке 9.

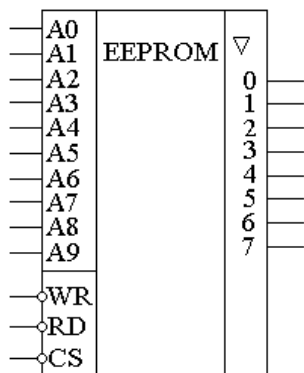


Рисунок 9. Условно-графическое обозначение электрически стираемого постоянного запоминающего устройства (EEPROM) на принципиальных схемах.

В последнее время наметилась тенденция уменьшения габаритов ЭСППЗУ за счет уменьшения количества внешних выводов микросхем. Для этого адрес и данные передаются в микросхему и из микросхемы через последовательный порт. При этом используются два вида последовательных портов - SPI порт и I2C порт (микросхемы 93сХХ и 24сХХ серий соответственно). Зарубежной серии 24сХХ соответствует отечественная серия микросхем 558PPX.

FLASH - ПЗУ отличаются от ЭСППЗУ тем, что стирание производится не каждой ячейки отдельно, а всей микросхемы в целом или блока запоминающей матрицы этой микросхемы, как это делалось в РПЗУ.

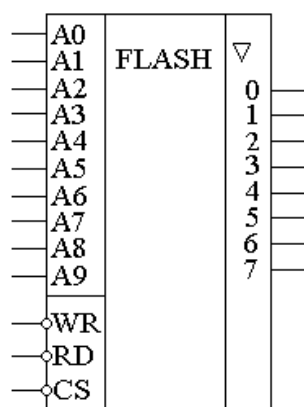


Рисунок 10. Условно-графическое обозначение FLASH памяти на принципиальных схемах.

При обращении к постоянному запоминающему устройству сначала необходимо выставить адрес ячейки памяти на шине адреса, а затем произвести операцию чтения из микросхемы. Эта временная диаграмма приведена на рисунке 11.

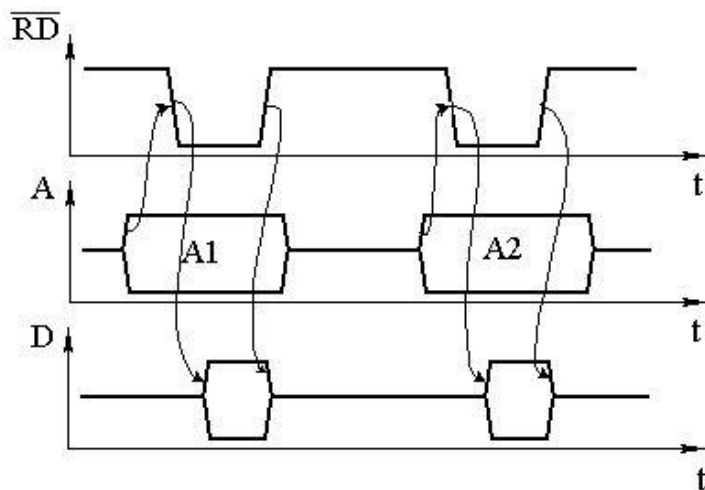


Рисунок 11. Временные диаграммы сигналов чтения информации из ПЗУ.

На рисунке 11 стрелочками показана последовательность, в которой должны формироваться управляющие сигналы. На этом рисунке RD - это сигнал чтения, A - сигналы выбора адреса ячейки (так как отдельные биты в шине адреса могут принимать разные значения, то показаны пути перехода, как в единичное, так и в нулевое состояние), D - выходная информация, считанная из выбранной ячейки ПЗУ.