

Цифровые сигналы и цифровые системы все шире используются при решении многих задач, встречающихся практически во всех областях жизни. Это объясняется как развитием радиоэлектроники вообще, так и рядом преимуществ цифровой техники по сравнению с традиционной аналоговой. Можно выделить два главных направления приложения цифровой техники: - **автоматизированное управление** технологическими процессами, включая автоматизированный контроль и диагностику технических средств, и **использование вычислительных технологий** для обработки сигналов, автоматизации проектирования, решения задач административно-организационного управления.

Данный курс ориентирован на подготовку специалистов по первому из них.

Цифровое устройство – устройство, осуществляющее прием, хранение и преобразование дискретной информации по некоторому алгоритму, т.е. вырабатывающее сигнал, определенным образом связанный с совокупностью входных сигналов. Обычно эта связь оказывается достаточно сложной и реализуется путем реализации ряда более простых преобразований сигналов. При этом реализация множества преобразований сигналов в устройстве оказывается возможной как аппаратным путем, т.е. с использованием отдельных узлов для реализации соответствующих преобразований, так и с использованием универсального узла – микропроцессора, где и осуществляются все эти преобразования. По этому принципу различают устройства **с жесткой логикой и программируемой логикой**. Выбор оптимального принципа построения конкретного устройства определяется требованиями, предъявляемыми к данному устройству. В любом случае, при этом, принцип действия этих устройств базируется на использовании законов булевой алгебры, допускающей двоичное представление сигналов.

В первой части курса рассматриваются основы функционирования цифровых устройств и принципы построения их базовых узлов с использованием интегральных схем малой и средней степени интеграции.

СИГНАЛЫ И ТИПОВЫЕ КАСКАДЫ ЦИФРОВОЙ ЭЛЕКТРОНИКИ СИГНАЛЫ В ЦИФРОВЫХ УСТРОЙСТВАХ

Сигнал — это изменение физической величины, используемой для передачи данных. Сигнал образуется на основе некоторой физической величины (электромагнитные или акустические колебания, электрическое напряжение и др.), традиционно называемой энергетическим носителем, путем изменения одного или нескольких ее параметров (амплитуды, частоты, фазы, длительности и др.) по закону передаваемой информации. Считают, что сигнал — это материально-энергетическое воплощение сообщения. Посредством совокупности сигналов можно представить любое сложное сообщение. Сигнал может преобразовываться без изменения смысла информации из одной физической величины в другую, более удобную для передачи по каналу связи и обработки в схемах компьютера. Изменение параметров физической величины по закону передаваемой информации называют **модуляцией**, а изменяемые параметры — **информативными**.

Сигналы **классифицируют** по ряду признаков, например,:

- степени определенности ожидаемых значений — случайные и детерминированные;
- структуре временного изменения — непрерывные и дискретные;
- особенностям спектрального представления — низкочастотные и высокочастотные, узкополосные и широкополосные;
- способу преобразования — кодированные, декодированные, усиленные, дискретизированные и т.д.;

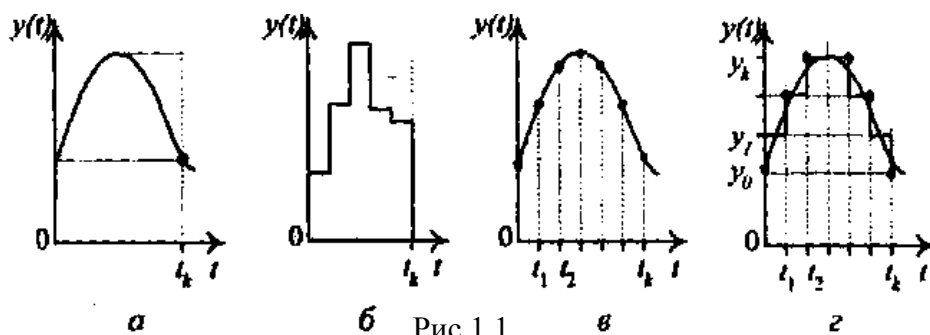
- принадлежности к виду связи — телеграфные, телефонные, радиолокационные,

В процессе передачи сигналов от источника к приемнику физические величины и способы их модуляции могут многократно изменяться, но содержание сообщения остается неизменным, поскольку оно определяется только законом модуляции.

В общем случае способ формализованного описания различных сигналов и соответственно сообщений) называется **представлением информации**. В теории информации рассматривают не физическое, а математическое представление сигналов, то есть их описание с помощью различных функций, формул, графиков, законов распределения вероятностей.

Наиболее распространенными способами представления сигналов являются временной, спектральный, статистический. Во многих случаях информация о протекании некоторого физического процесса поступает от соответствующих датчиков в виде электрических сигналов, которые непрерывно изменяются во времени. Однако в цифровой схемотехнике, изначально предполагающей использование нелинейных преобразований параметров сигналов, преобладает временное их описание, отражающее закон изменения одного из параметров (чаще всего напряжения) во времени. Таким образом, сигналы представляются одномерными функциями времени. Для наглядности при их анализе часто используются временные диаграммы, графически отображающие изменения сигналов на временной оси.

По характеру поведения сигнала на временной оси $y(t)$ различают следующие разновидности сигналов:



1. Непрерывная функция непрерывного аргумента в интервале времени $0 \leq t \leq T$ (Рис.1.1, а).

2. Дискретная функция непрерывного аргумента (рис. 1.1, б). Значения, принимаемые функцией $y(t)$, образуют дискретный ряд чисел $y_i, i = 1, 2, \dots, k$. Значение аргумента может быть любым в заданном интервале времени $0 \leq t \leq T$. Преобразование непрерывной функции $y(t)$ в дискретное множество значений y_i называется **квантованием по уровню**.

3. Непрерывная функция дискретного аргумента (рис. 1.1, в). Значения функции; определяются только на дискретном множестве $t_j, j = 1, 2, \dots, J$. Функция $y(t)$ может принимать любые значения в заданном диапазоне y_{\min}, y_{\max} . Преобразование функции $y(t)$ непрерывного аргумента t в функцию $y(t_j)$ дискретного аргумента t_j называется **дискретизацией (квантованием) во времени**.

4. Дискретная функция дискретного аргумента (рис.1.1, г). Значения, принимаемые функцией и аргументом, образуют дискретные ряды чисел y_0, y_1, \dots, y_k и t_0, t_1, \dots, t_k .

В цифровой электронике используются логические сигналы – цифровые сигналы, число возможных дискретных значений для которых ограничено до минимума, необходимого для передачи информации, **1 бита**. В этом случае сигналы соотносят к одному из лишь двух возможных классов (0 или 1). При этом соответствующее значение бита информации может быть передано как уровнем сигнала (**потенциальные сигналы**), так и в виде перехода из одного состояния в другое (**импульсные сигналы**). В свою очередь, в зависимости от соотношения уровней для потенциальных сигналов или направления изменения уровня сигнала для импульсных различают **положительную логику** ($U^0 < U^1$ для потенциальных сигналов, переход $0 \rightarrow 1$ принимается за 1 для импульсных сигналов), и **отрицательную логику** в противном случае.

Такое неполное использование возможностей электрических сигналов для передачи информации приводит к усложнению алгоритмов для получения тех же результатов, что и в аналоговых устройствах. К тому же аналоговые схемы обладают потенциально более высоким быстродействием. Но современное состояние теории и практики цифровой обработки сигналов позволяет часто несколько более простых и с более высокой точностью решений многих задач. Одним из достоинств цифровой электроники является хорошо развитая элементная база, что позволяет обычно свести проектированию цифровых устройств к отысканию наиболее подходящего алгоритма и его оптимальной реализации.

Согласно теореме Котельникова сигнал, описываемый функцией с ограниченным спектром, определяется своими дискретными значениями, которые отсчитываются через интервалы времени $t = 1/2F_c$, где F_c — ширина спектра. Это означает, что сигнал $y(t)$ можно представить отдельными мгновенными значениями, которые отсчитываются через конечный интервал времени. По этим значениям оказывается возможным полностью восстановить исходный непрерывный сигнал. Таким образом, осуществляя дискретизацию с необходимой частотой и квантование с достаточной точностью, оказывается возможным реализовать достоинства цифровых методов обработки и для аналоговых сигналов. При этом часто используются и самые разные способы представления сигналов. В частности, к дискретно-непрерывным сигналам относят также время-импульсное представление первичного сигнала $y(t)$ прямоугольными импульсами с непрерывным информативным параметром t_i/T , где t_i — длительность импульсов, пропорциональная значению сигнала; T — период повторения импульсов (рис. 1.2 а).

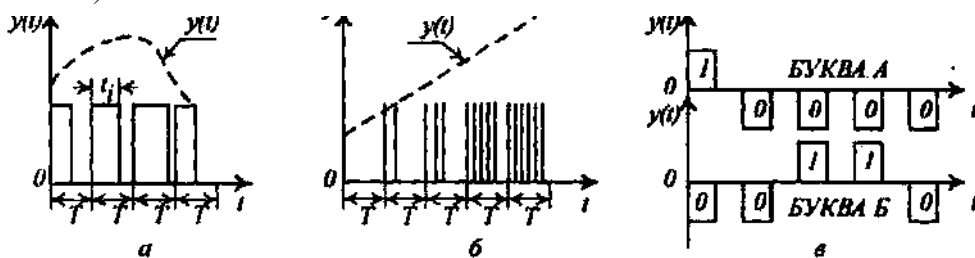


Рис.1.2.

При число-импульсном представлении (рис.1.2 б) информативным параметром является количество импульсов за период T .

В телеграфной связи используют импульсно-кодовое

представление символов. Например, в коде Бодо передачу символов реализуют последовательной посылкой во времени пяти импульсов разной полярности, как показано на рис. 1.2 в для букв Л и Б.

В цифровой схемотехнике используют разрядно-цифровое (или просто цифровое) кодирование, в котором исходный сигнал представляется группой символов, отображающих значения цифр 0 и 1 двоичной системы счисления уровнями электрических сигналов. В этом случае разряды двоичного числа характеризуются весом, кратным степени двойки — 1, 2, 4, 8, ... (в направлении от младших к старшим разрядам), так что, например, для четырехразрядного двоичного числа имеем: $A = A_3A_2A_1A_0 = a_32^3 + a_22^2 + a_12^1 + a_02^0$ ($1101_2 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 13$).

Передача цифрового кода может быть осуществлена как последовательной пере-

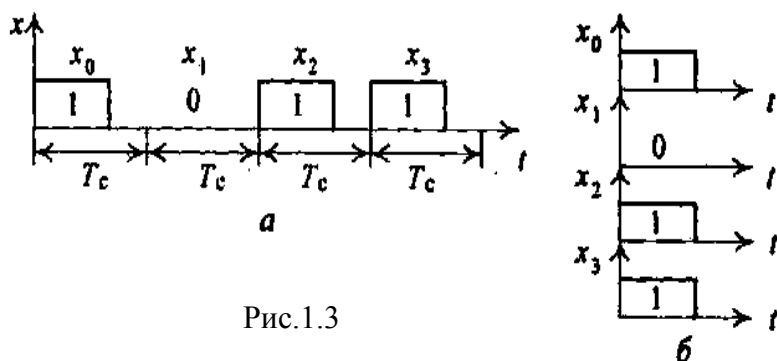


Рис.1.3

дачей его символов во времени (**последовательный код**) с помощью единственного канала передачи (рис. 1.3 а), так одновременной их передачей с использова-

нием многоканальной связи (**параллельный код**) (рис. 1.3 б).

На практике последовательный код используют при передаче информации на большие расстояния (например, между компьютерами), а параллельный код — при передаче информации на малые расстояния (например, между блоками компьютера). Параллельный код обеспечивает, очевидно, наибольшую скорость передачи, но требует существенно большего числа независимых линий связи.

Абсолютные значения уровней напряжений, соответствующих 0 и 1, определяют конкретную элементную базу, используемой при построении цифровых устройств. В частности, в наиболее распространенных ТТЛШ сериях эти напряжения соответствуют 0,4В и 2,4В. Между этими значениями существует некоторая защитная переходная зона с тем, чтобы свести до минимума влияние внешних условий и помех на работу устройств. Менее строго уровни напряжений, соответствующие 0 и 1 определены в КМОП – сериях, допускающих большой разброс возможных напряжений питания схем (3...15)В.

Одной из главных характеристик цифровых устройств является их **быстродействие**, определяющее предельную скорость изменения входных сигналов, при котором устройство нормально функционирует. Для оценки быстродействия обычно используются временные параметры тестовых сигналов, в качестве которых чаще всего используют прямоугольные импульсы.

Под импульсом понимают кратковременное изменение напряжения или тока в электрической цепи. В цифровой технике используются **видеоимпульсы**, не имеющие в отличие от **радиоимпульсов** высокочастотного заполнения.

Основными характеристиками и параметрами импульсов для характеристики быстродействия устройств являются:

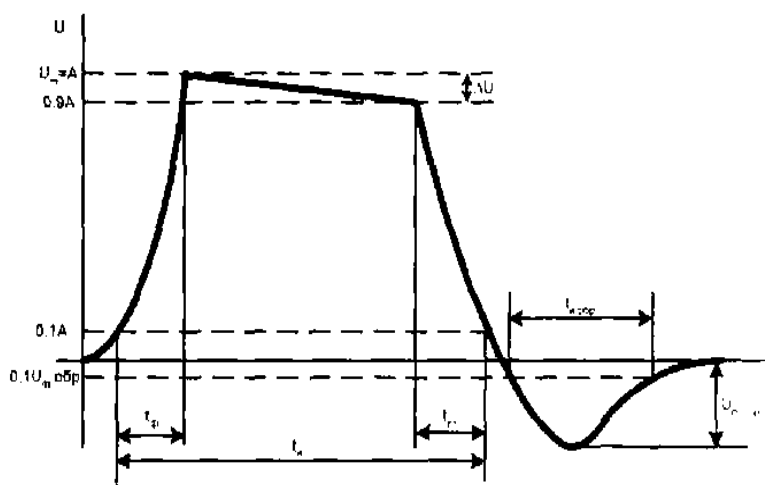


Рис.1.4.

- амплитуда импульса $U_m = A$;
- активная длительность импульса ; (из - меряется на уровне 0,5 A)
- длительность фронта t_f (определяется как время, за которое выходной сигнал изменяется от 0,1A до ,9A);
- длительность среза $t_{ср}$ (определяется как время, за которое выходной сигнал изменяется от 0,9A до ,1A) dt
- время задержки $t_{зд}$ (определяется

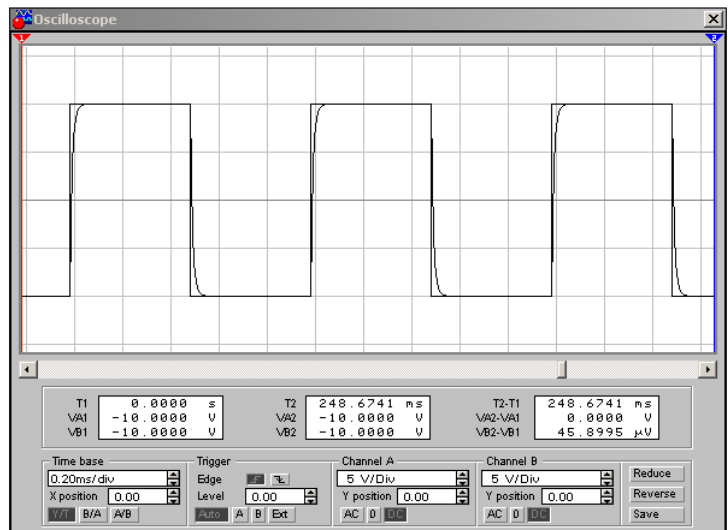
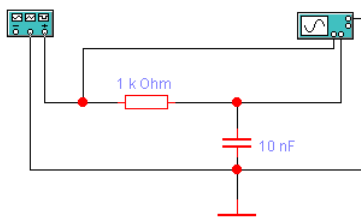
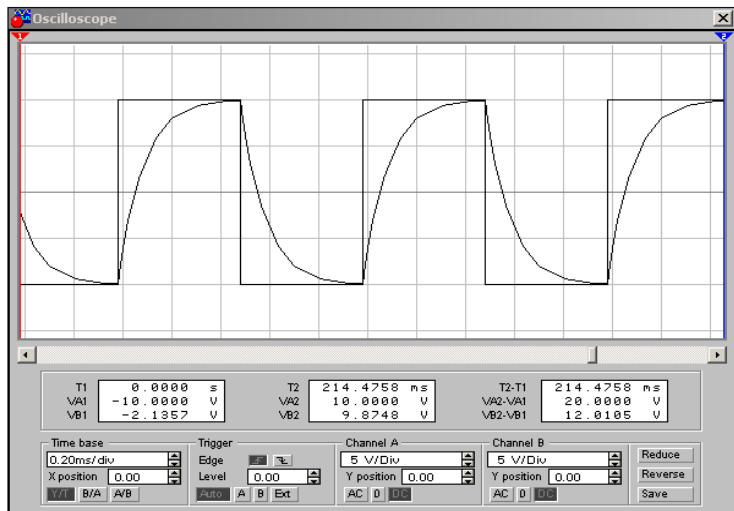
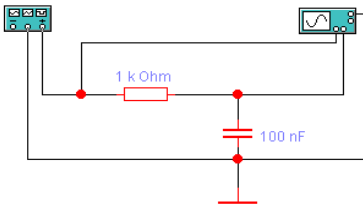
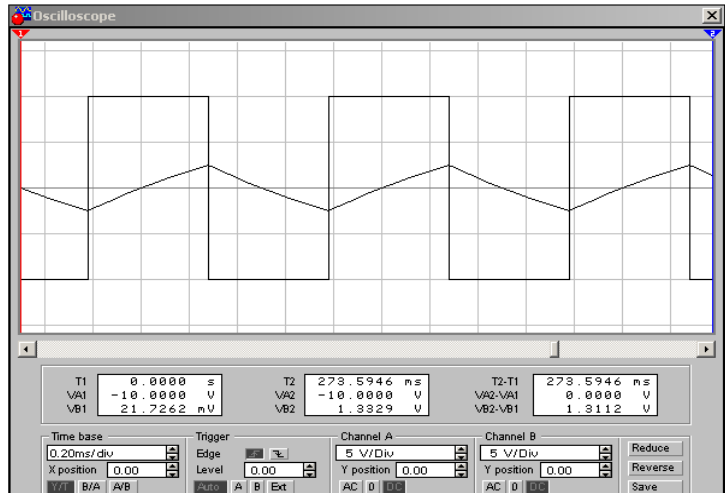
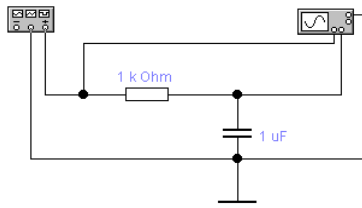
как время между началами активных длительностей импульсов на входе и выходе устройства)

При оценке цифровых устройств с помощью периодической последовательности импульсов, вводятся дополнительные параметры - период следования импульсов T и скважность $Q = T/t_u$.

ТИПОВЫЕ КАСКАДЫ ИМПУЛЬСНЫХ устройств

Прохождение электрических сигналов через цепи сопровождается изменением их формы. Рассмотрим выходные сигналы для отдельных случаев радиотехнических цепей.

А) Интегрирующая цепь



Напряжение на выходе интегрирующей цепи, представляющее собой напряжение на обкладках конденсатора, при импульсном воздействии на входе изменяется по экспоненциальному закону. Для положительных прямоугольных импульсов нарастание напряжения (фронт выходного сигнала) описывается выражением вида $U = A \cdot \exp\left(-t/RC\right)$, уменьшение (спад выходного сигнала) – зависимостью $U = A \cdot \left[-\exp\left(-t/RC\right)\right]$. На приведенных осциллограммах иллюстрируется изменение вида выходного сигнала с изменением параметра - постоянной времени $\tau = R \cdot C$ инте-

грирующей цепи. Это свойство интегрирующей цепи часто используется при реализации цепей, предназначенных для задержки импульсов.

В) Диодные ключи

В цифровой и импульсной технике широко используются схемы на полупроводниковых диодах. Это разного рода защитные устройства, ограничители напряжения, фиксаторы уровня напряжения. Одно из их применений - базовый элемент в схемах диодной логики (ДЛ). Работа этих устройств очевидна, поэтому оценим только их быстродействие на примере последовательного диодного ключа.

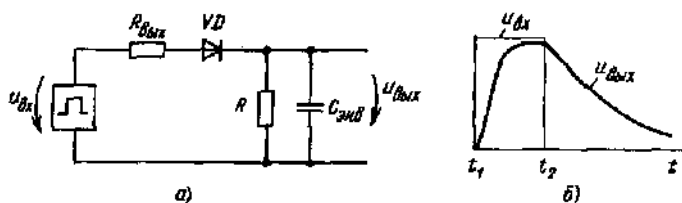


Рис.1.5.

Рассмотрим факторы, определяющие скорости нарастания и спада напряжения на выходе каскада. Будем считать, что к выходу (параллельно R) подключен некоторый эквивалентный емкостной элемент, емкость $C_{экв}$ которого

включает емкость нагрузки, монтажа и закрытых диодов. Для процессов с момента t_1 действия на входе напряжения уровня $\text{лог.}1$ образуется эквивалентная схема, представленная на рис. 1.5. Сопротивление, включающее сопротивление открытого диода VD и выходное сопротивление элемента, с которого напряжение поступает на $V_{x.1}$, обозначим r . Очевидно, из-за наличия емкости $C_{экв}$ напряжение на элементе не может измениться скачком, оно растет по экспоненциальному закону с постоянной времени $\tau_{нар} = C_{экв} rR / (r + R) \approx rC_{экв}$ (так как $r \ll R$), стремясь к значению $u_{вх} r / (r + R) \approx u_{вх}$.

В момент t_2 при переходе напряжения на входе от уровня $\text{лог.}1$ к уровню $\text{лог.}0$ закрывается и диод VD . При этом напряжение на выходе снижается по мере разряда емкости $C_{экв}$ через резистор R с постоянной времени $\tau_{сп} = RC_{экв}$. Так как $\tau_{сп} \gg \tau_{нар}$, скорость спада выходного напряжения оказывается меньше скорости нарастания этого напряжения. В результате очередная подача на вход напряжения уровня $\text{лог.}1$ возможна лишь после того, как остаточное напряжение на выходе снизится до определенного малого значения, Поэтому медленный спад выходного напряжения является критическим параметром, определяющим быстродействие диодного ключа.

С) Транзисторные ключи

Транзисторные ключи широко используются в импульсных устройствах в виде самостоятельного устройства, непосредственно обеспечивающего коммутацию нагрузки. Еще одно из их применений – управляемый элемент в аналогово-дискретных схемах, работающий в ключевом режиме, т.е. с использованием только двух состояний транзистора – закрытого или полностью открытого. В этом случае достигается наиболее щадящий режим для транзистора (снижаются до минимума потери на его разогрев) и, следовательно, наибольшее КПД.

В цифровой схемотехнике транзисторные ключи являются базовой схемой практически всех логических элементов, решающим образом определяя их основные характеристики. Обычно используется ключ, в котором транзистор включён по схеме с общим эмиттером (рис.1.6), где показаны схема такого ключа, входная и семейство выходных характеристик с нагрузочной прямой транзисторного каскада.

a) Статический режим транзисторного ключа

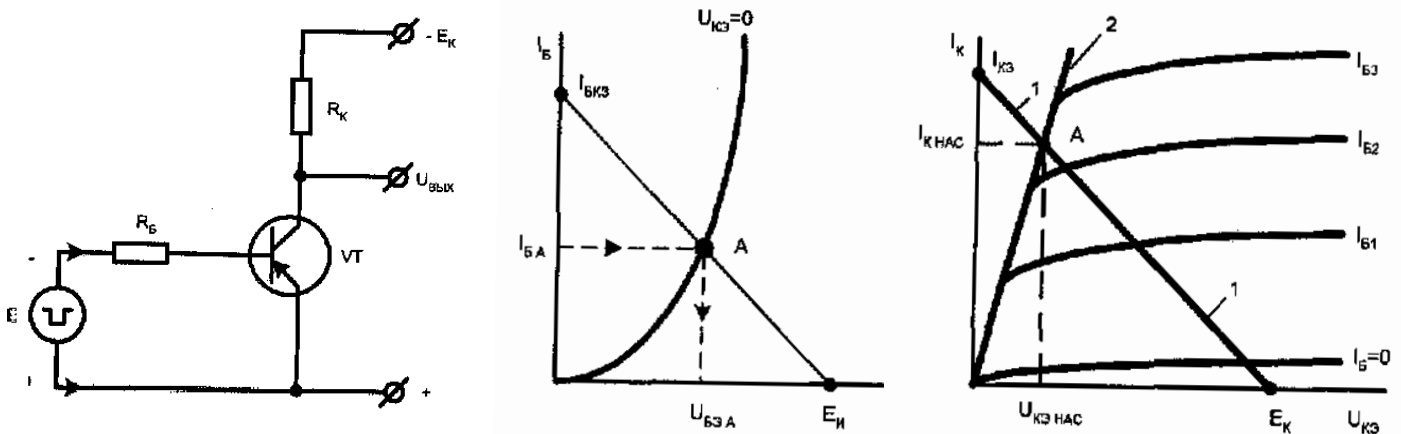


Рис.1.5

Ключевой режим транзистора достигается подачей на его вход напряжения, величина которого обеспечивает только два состояния транзистора.

При $U_{вх} \geq 0$ (логический 0 в системе отрицательной логики) транзистор оказывается закрытым ($I_k = 0$), т.е. выходное напряжение $U_{вых} = U_k = 0$. Для отпирания транзистора на его вход подается $U_{вх} < 0$. Нетрудно видеть, что подобный ключ реализует логическую операции инверсии сигнала в системе отрицательной логики.

Очевидно, что транзисторный ключ не является идеальным ключом, т.к. транзистор не обеспечивает желаемых величин сопротивлений $R_{откр} = 0$ и $R_{закр} = \infty$. Действительно, даже в закрытом состоянии в транзисторе протекают тепловые токи. Но и в пренебрежении ими конечное сопротивление нагрузки R_n приводит к образованию делителя напряжения, так что для закрытого транзистора выходное напряжение не может по абсолютной величине превышать значения $U_{вых} = E_k R_n / (R_k + R_n)$.

В открытом состоянии транзистор характеризуется режимом насыщения, т.е. существует предельная величина тока $I_k = I_{кна}$, определяемая характеристиками транзистора и значениями R_k и E_k (т.А на нагрузочной прямой), так что дальнейшее увеличение уровня входного сигнала оказывается бессмысленным. В результате достаточно обеспечить величину входного тока $I_{бнас} = I_{кна} / \beta$ (т.А на входной характеристике) для максимального отпирания транзистора (β - коэффициент усиления по току транзистора). К сожалению, транзисторы отличаются большим технологическим разбросом, а их характеристики сильно меняются с изменением внешних условий. По этой причине для надежного обеспечения режима насыщения используют повышенное значение тока базы $I = s I_{бнас}$ ($s = 1,5 \dots 3$ - степень насыщения). В лучшем случае, таким образом, минимальное по величине напряжение на выходе транзисторного ключа не может быть меньше некоторого напряжения $U_{кэнас}$, являющегося параметром транзистора.

b) Динамический режим транзисторного ключа

Кроме рассмотренных статических характеристик транзисторных ключей, не меньший интерес представляют их динамические характеристики, отражающие быстродействие этих каскадов. Действительно, переходные процессы в системах обработки

информации приводят к ее искажению, что вынуждает исключить эти моменты времени из рабочего цикла. Очевидно, что чем продолжительней эти t процессы, тем меньше пропускная способность проектируемых цифровых устройств. Кардинальным путем является повышение быстродействия ее элементов.

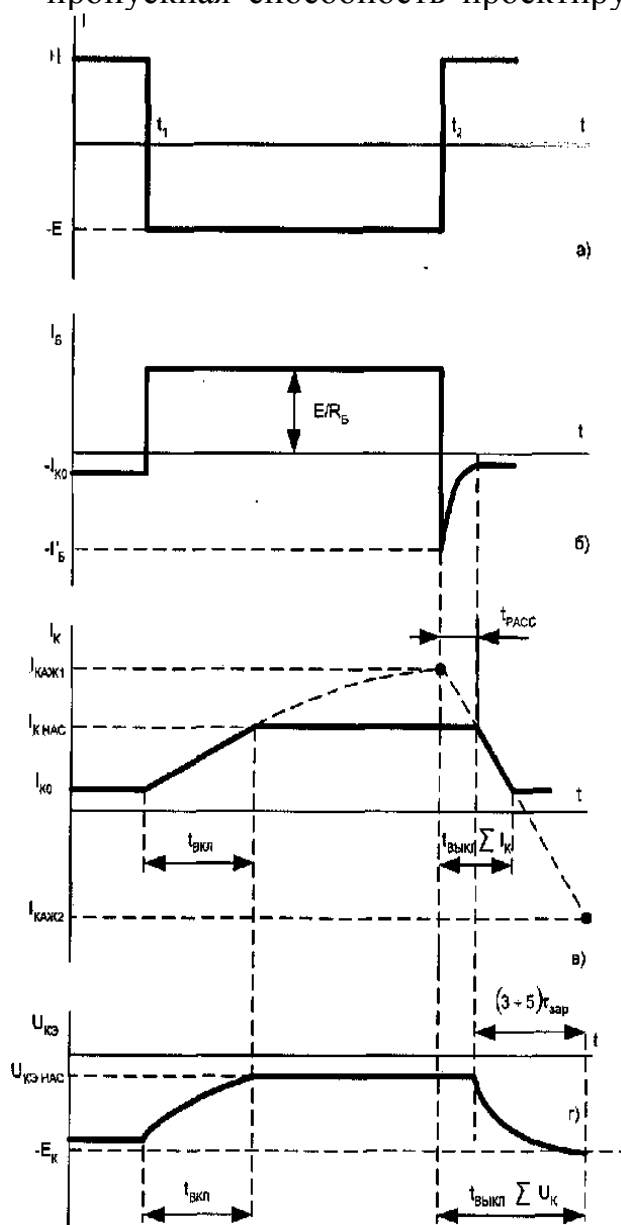


Рис 1.6.

3. $t_{12} < t < t_2$

Этот интервал представляет собой стационарный режим ключа. Ограничение тока коллектора сопровождается накоплением в базе транзистора избыточного заряда, величина которого пропорциональна степени насыщения s .

4. $t_{22} > t > t_2$

Входное напряжение скачком меняется до величины E , соответствующей надежному запирающему напряжению транзистора. Но в базе оказывается избыточный заряд, рассасывание которого представляет отрицательный ток базы. Этот процесс осуществляется с конечной скоростью, т.е. существует конечный промежуток времени – **время выхода из насыщения (время рассасывания)** -, в течении которого ток коллектора, а, следовательно, и выходное напряжение не меняются. Длительность этого процесса при большом запирающем напряжении приближенно описывается соотношением $t_{рас} \approx \tau_{эк} \cdot I_{кнас} \left(-1 \right) / 2 \beta I_б$. Здесь $\tau_{эк} = \tau_\beta + R_k C_k \left(+ \beta \right)$ - характеристика быстродействия

Рассмотрим искажения прямоугольного импульса, сопровождающие его прохождение через транзисторный ключ (рис 1.5). Осциллограммы, иллюстрирующие работу схемы, приведены на рис. 1.6.

1. $t < t_1$

Ключ находится в стационарном состоянии – транзистор закрыт положительным напряжением E .

2. $t_1 < t < t_{12}$

Входное напряжение скачком изменяется до величины $-E$. Это напряжение открывает транзистор – через переход Э-Б протекает ток $I_б = E/R_б > I_{бнас}$. Малое сопротивление перехода Э-Б и малая входная емкость практически не сказываются на форме входного тока, повторяющей входной сигнал. Ток коллектора же по экспоненциальному закону начинает возрастать до некоторого **кажущегося** значения $I_{каж1} = I_б \beta$, но в момент t_2 достигает значения $I_{кнас}$. напряжение на коллекторе транзистора повторяет его форму. Этот момент времени определяется характеристиками ключа и завершает время нарастания выходного импульса. Для случая $s > 1$, что характерно для цифровых схем, длительность фронта может быть записана в виде $t_\phi \approx \tau_\beta / s$, где τ_β - характеристика частотных свойств транзистора.

ключа, складывающаяся из инерционности транзистора и его нагрузки в коллекторной цепи.

5. $t > t_{22}$

Начинается выключение транзистора в активном режиме. Продолжительность этого процесса (длительность среза) составляет величину $t_{cp} \approx \tau_{эк} \cdot I_{кнас} / 2\beta I_{\sigma}$. по завершении этого процесса транзистор снова переходит в стационарный режим.

Таким образом, быстродействие транзисторного ключа снижается за счет наличия трех последовательных переходных процессов и может быть оценено их суммарной длительностью $t_{\Sigma} = t_{\phi} + t_{cp} + t_{расс}$. Как можно заметить, быстродействие транзисторного ключа зависит от многих факторов, причем уменьшение длительности фронта требует повышения степени насыщения ключа в открытом состоянии, тогда как для уменьшения длительности времени рассасывания необходимо ее минимизировать.

с) Схемы транзисторных ключей

Схема с ускоряющим конденсатором (рис.1.7.а).

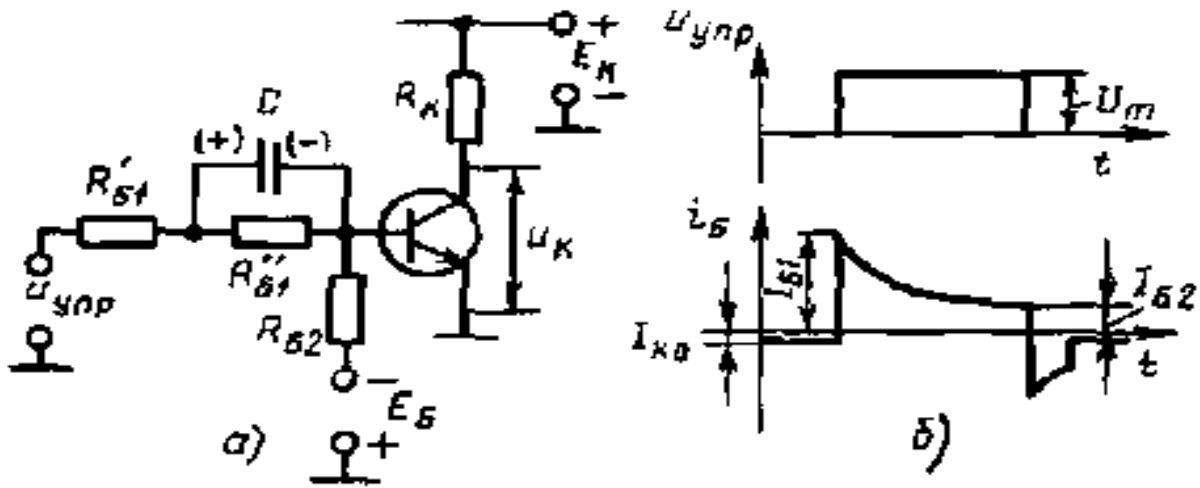


Рис.1.7.

Одним из возможных способов повышения быстродействия является тот факт, что процессы, определяющие инерционность ключа, оказываются разнесенными во времени. Это позволяет скорректировать управляющий ключом сигнал для достижения оптимальных характеристик, т.е. уменьшить ток базы после того, как транзистор вошел в режим насыщения. При этом уменьшается степень насыщения в стационарном включенном режиме и, как следствие, время рассасывания заряда (задержка выключения) после окончания входного импульса.

Указанный эффект объясняется следующим. При включении транзистора составляющая базового тока, обусловленная положительным управляющим импульсом, проходит через резистор $R_{б1}$ и разряженный ускоряющий конденсатор C , т. е. ограничивается только $R_{б1}$. ($R_{б1} \gg R_{бэ}$). Поэтому в базу транзистора втекает значительный ток $I_{б1}$ (рис. 1.6.б), что позволяет существенно сократить длительность фронта.

Реально конденсатор C начинает заряжаться, когда транзистор уже находится в насыщении. После зарядки конденсатора ток управляющего импульса ограничивается уже двумя резисторами: $R_{б1}$ и $R_{б2}$. Поэтому теперь ток базы $I_{б2}$ существенно меньше $I_{б1}$.

С окончанием управляющего импульса базовый ток скачкообразно меняет свое направление, затем экспоненциально уменьшается, а после запираения транзистора уста-

навливается на уровне $I_{к0}$ (рис. 1.7.б). При этом напряжение на зарядившемся конденсаторе (его полярность указана в скобках на рис. 1.7.а) ускоряет выключение транзистора. Таким образом, в каскаде крутой фронт включения транзистора сочетается с уменьшением длительности фронта и задержки выключения.

Ключ с нелинейной отрицательной обратной связью.

Время рассасывания можно полностью устранить, если исключить насыщение транзистора. Это, в свою очередь, может быть реализовано путем обеспечения на переходе К-Б только запирающего напряжения (для схемы на рис.1.7.а, например, коллектор транзистора *n-p-n-типа* должен всегда иметь положительный потенциал относительно базы). Однако при включении транзистора база получает положительный потенциал со стороны входа ключа, а положительный потенциал коллектора убывает по мере увеличения коллекторного тока, так что напряжение $U_{КБ}$ может стать отрицательным. Выполнение этого условия обеспечивается в каскаде (рис.1.8.), в котором за счет диода *VD* реализована нелинейная отрицательная обратная связь. В отсутствие положительных управляющих импульсов транзистор *VT* и диод *VD* заперты — обратная связь отсутствует.

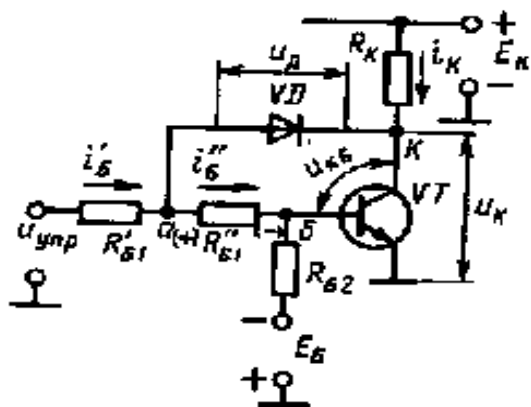


Рис.1.8

опустится несколько ниже значения $u_{aб}$ диод откроется и соединит точки *a* и *к*. После этого, если считать диод идеальным ($u_{VD} = 0$), потенциал коллектора всегда будет оставаться равным потенциалу точки *a*, т. е. более положительным относительно потенциала базы на величину $i_б'' R_{δ1}''$. Реально $u_{VD} \neq 0$, за счет чего потенциал коллектора ниже потенциала точки *a* на u_{VD} , однако при правильно выбранных элементах схемы (когда $u_{aб} > i_б'' R_{δ1}''$) напряжение $u_{кб} > 0$, т.е. насыщение транзистора отсутствует.

В рассмотренном каскаде на этапе включения можно допускать большой базовый ток. После отпирания диода через него ответвляется значительная часть управляющего тока, благодаря чему базовый ток существенно уменьшается. Так как через резистор R_K ток не может изменяться (потенциалы E_K и u_K его выводов постоянны), то ответвившийся через диод ток замыкается на «землю» через транзистор и цепи, присоединяемые к коллектору.

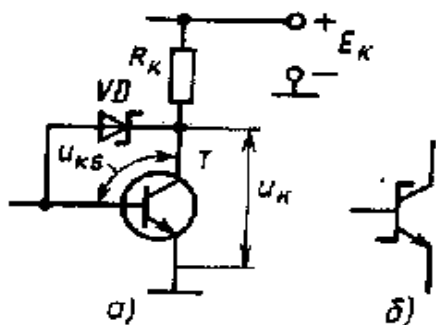


Рис.1.9.

С поступлением положительного управляющего импульса транзистор отпирается, коллекторный ток I_K нарастает, а потенциал коллектора уменьшается. При этом через резисторы $R_{δ1}$ и $R_{δ2}$ проходит одинаковый ток $i_б' = i_б''$, а анод диода имеет положительный потенциал относительно базы ($u_{aб} = i_б'' R_{δ1}'' > 0$; на рис. 1.7, где полярность напряжения на резисторе $R_{δ1}''$ показана и скобках). Когда в процессе включения потенциал коллектора $u_{кб}$

В настоящее время нелинейную отрицательную обратную связь реализуют с помощью диода Шоттки. Он представляет собой алюминий-кремниевый диод с малым падением напряжения в открытом состоянии (менее 0,5 В), в котором практически отсутствует накопление зарядов, благодаря чему

время его переключения составляет доли наносекунды.

В процессе включения транзистора VT (рис.1.9.а) диод Шотки отпирается. Хотя в данном случае $u_{кб} < 0$ это напряжение меньше того, при котором коллекторно-базовый переход смещается в прямом направлении. Иными словами, диод Шотки отпирается раньше, чем коллекторно-базовый переход. Поэтому практически насыщение транзистора и связанная с ним задержка выключения отсутствуют.

Единую интегральную структуру транзистор - диод Шотки называют транзистором Шотки. Его изображение приведено на рис.1.9б.

Ключ на переключателях тока.

Быстродействие транзисторных ключей во многих случаях является главным требованием при реализации цифровых устройств. Одним из возможных путей достижения этой цели является использование ключей на переключателях тока, характеризуемых схемой избыточностью и повышенным энергопотреблением.

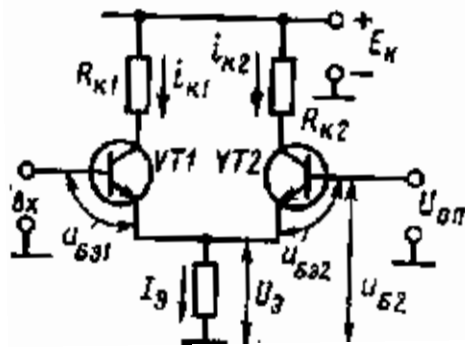


Рис.1.10.

Такой каскад предназначен для переключения тока из одной цепи в другую. Схема переключателя (рис. 1.10) состоит из двух идентичных усилителей на транзисторах $VT1$ и $VT2$, связанных эмиттерами. При изменении напряжения u_{ex} ток I_3 в эмиттерной цепи переключается из цепи транзистора $VT1$ в цепь транзистора $VT2$ или наоборот. На базу $VT2$ подается стабильное опорное напряжение u_{on} . Напряжение $u_э$ на эмиттерах приблизительно повторяет большее из двух напряжений: u_{ex} и u_{on} . При $u_{ex} = u_{on}$ токи и напряжения в плечах схемы соответственно одинаковы:

$$i_{к1} = i_{к2} \approx 0.5I_3, \quad u_{к1} = u_{к2} \approx E_k - 0.5I_3 R_k,$$

$$u_э = u_{on} - e_0,$$

где e_0 (десятые доли вольт) — напряжение $u_{кэ}$ на открытом эмиттерном переходе. Предположим, что $u_{кэ}$ начинает возрастать от значения $u_{ex} = u_{on}$. При этом напряжение $u_э$ тоже увеличивается, а так как $u_{бэ2} = u_{on} = const$, то $u_{бэ2}$ уменьшается, т. е. транзистор $VT2$ начинает запирается. Если считать, что он запрется при $u_{бэ2} = 0$, то это произойдет, когда $u_э = u_{on}$, т. е. при $u_{ex} = u_{on} + e_0$. После запираения транзистора $VT2$ ток I_3 переключится в цепь транзистора $VT1$.

Если u_{ex} начинает уменьшаться от уровня u_{on} , то напряжение $u_{бэ1}$ тоже уменьшается, так как потенциал эмиттеров задается большим сейчас стабильным напряжением u_{on} ($u_э = u_{on} - e_0$). Когда u_{ex} окажется равным $u_э$, т. е. при $u_{ex} = u_{on} - e_0$, транзистор $VT1$ запрется и весь эмиттерный ток переключится в цепь транзистора $VT2$. Заметим, что после изменения u_{ex} транзисторы в процессе переключения управляются со стороны эмиттеров, т. е. работают как каскады с общей базой.

Элементы схемы переключателя выбраны с таким расчетом, чтобы исключить насыщение транзисторов. Однако в данном случае коллекторные токи транзисторов оказываются в активном режиме достаточно стабильными. Это объясняется стабильностью тока эмиттеров ($I_э = u_э / R_э \approx u_{on} / R_э$), с которым коллекторный ток каждого транзистора связан коэффициентом α ($I_к = \alpha I_э$); значения последнего колеблются

незначительно (в десятки раз меньше, чем коэффициент β в схеме с общим эмиттером).

Так как транзисторы работают по схеме с общей базой, их постоянная времени $\tau_\alpha \gg \tau_\beta$, а из-за отсутствия насыщения исключается задержка включения транзистора. В результате быстродействие переключателя тока оказывается значительно выше, чем у ключа с общим эмиттером.

Ключи на полевых транзисторах.

Большое число задач, решаемых с помощью цифровых устройств, предъявляет повышенные требования к энергопотреблению этих устройств, не накладывая особых требований к их быстродействию. Для этих целей широко используются ключи на полевых МОП-транзисторах, выгодно отличающиеся от биполярных возможностью достижения существенно большей степени интеграции элементов при изготовлении интегральных схем.

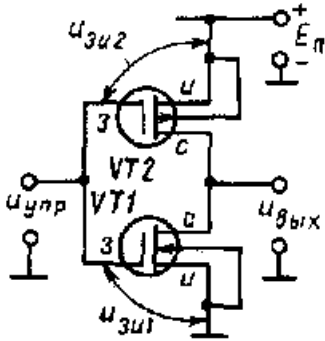


Рис.1.11.

Наибольшее распространение получили ключи на *комплемментарных* структурах (КМОП-транзисторы), составленных из транзисторов разного типа проводимости. Управляющее напряжение подается на объединенные затворы, подложки соединены с истоками. В стационарном режиме один из транзисторов открыт, а другой закрыт. Причем в закрытом состоянии сопротивление полевого транзистора оказывается очень большим, так что энергия от источника сигнала практически не потребляется. Энергия источника питания расходуется лишь на перезаряд паразитных емкостей при переключениях схемы, так что с

повышением частоты переключения энергопотребление растет. Паразитные емкости оказывают влияние и на быстродействие подобных ключей, которое оказывается ниже, чем у биполярных.

Контрольные вопросы

1. Различия между аналоговым, цифровым и логическим сигналом .
2. Различия между аналоговым, цифровым и логическим устройством.
3. Различия между последовательным и параллельным представлением цифровой информации.
4. Назначение и характеристики диодного ключа.
5. Назначение и характеристики транзисторного ключа.
6. Степень насыщения транзисторного ключа и ее влияние на характеристики.
7. Характеристики быстродействия транзисторного ключа.
8. Способы уменьшения длительности фронта транзисторного ключа.
9. Способы уменьшения длительности времени рассасывания транзисторного ключа.
10. Способы уменьшения длительности среза транзисторного ключа.
11. Какие показатели транзисторного ключа и каким образом улучшает ускоряющий конденсатор.
12. Какие показатели транзисторного ключа и каким образом улучшает нелинейная отрицательная обратная связь.
13. Достоинства токовых переключателей в схемах транзисторных ключей.
14. Особенности транзисторных ключей на базе полевых транзисторов.

ЦИФРОВЫЕ МИКРОСХЕМЫ

Основные характеристики цифровых микросхем

Устройства, предназначенные для обработки цифровых сигналов, относятся к *цифровым устройствам* и обычно реализуют сложные алгоритмы обработки сигнала путем последовательной реализации типовых преобразований. В устройствах с «жесткой логикой» для реализации этих преобразований используются аппаратные средства в

виде *функциональных узлов*. Современная элементная база цифровой электроники располагает большим набором готовых функциональных узлов, реализованных в виде интегральных микросхем (ИМС). Для сопряжения между собой стандартных функциональных узлов, выполнения отдельных логических задач или построения нетиповых функциональных узлов широко используются *логические элементы*, входящие в перечень выпускаемых цифровых ИМС и реализующие простейшие логические операции или их комбинации.

Набор цифровых микросхем с общими конструктивно-технологическими и схемотехническими признаками образует *серию ИМС*. В настоящее время имеется широкий выбор между сериями ИМС, различающимися уровнями логических сигналов, характеристиками быстродействия и энергопотребления. При использовании ИМС различных серий в устройстве может возникнуть задача согласования их сигналов. Однако существует ряд серий ИМС, позволяющих реализовать цифровое устройство, не выходя за рамки этой серии. Выбор же конкретной серии определяется характеристикой ее компонентов как элементов электрической цепи. Возможности конкретной серии обычно иллюстрируются на примере характеристик реализованного в этой серии простейшего логического устройства - *элемента НЕ (инвертора)*.

Статические характеристики инвертора определяют с помощью входной, нагрузочной и передаточной характеристик.

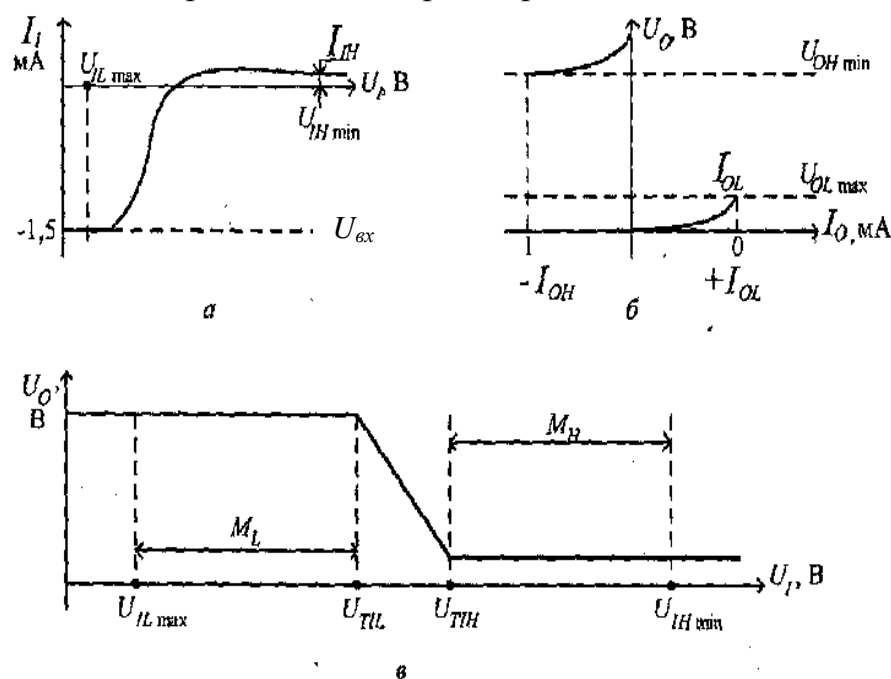


Рис.2.1.

Входная характеристика (рис.2.1.а) $I_{ex} = f(U_{ex})$ - зависимость входного тока от входного напряжения. Токи, втекающие в рассматриваемый элемент, считаются положительными, вытекающие - отрицательными. Эта характеристика позволяет оценить влияние подключения этого элемента к источнику сигнала.

Нагрузочная характеристика (рис.2.1.б) $U_{вых} = f(I_{н})$ - зависимость выходного напряжения от тока нагрузки. На рисунке принято, что токи, втекающие

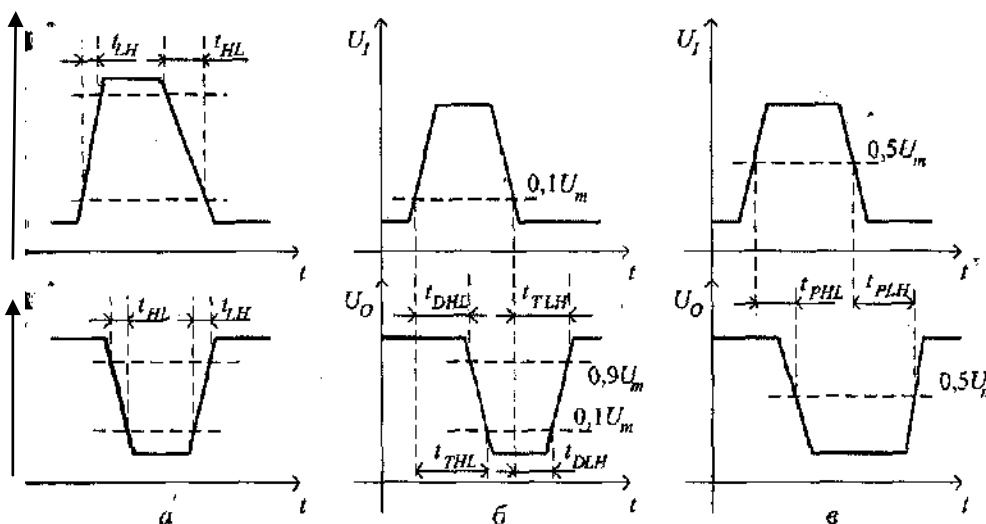
в рассматриваемый элемент, считаются положительными, вытекающие - отрицательными. Эта характеристика позволяет оценить рассматриваемый элемент как источник логического сигнала для последующих устройств.

Передаточная характеристика (рис.2.1.в) $U_{вых} = f(U_{вх})$ - зависимость выходного напряжения от входного. Эта характеристика позволяет сформулировать требования к входным логическим сигналам с учетом технологического разброса при изготовлении цифровых схем $U_{разбр}$ и неизбежного присутствия разного рода помех в реальном устройстве $U_{ном}$. Вводится количественная оценка максимально допустимого значе-

ния логического 0 на входе устройства $U_{\max}^0 = U_{\text{пор}} - U_{\text{разбр}} - U_{\text{ном}}$ и минимально допустимого значения логической 1 $U_{\min}^1 = U_{\text{пор}} + U_{\text{разбр}} + U_{\text{ном}}$, не приводящих к сбоям в работе устройства. ($U_{\text{пор}}$ - значение порогового входного напряжения, из сравнения с которым сигнал должен быть классифицирован в классе логических сигналов, т.е. опознан как лог.0 или лог.1. В реальных устройствах это значение оказывается примерно на середине переходной зоны передаточной характеристики.)

На практике обычно пользуются общей количественной интегральной оценкой – коэффициентом разветвления по выходу, который указывает максимальное число входов логических элементов, которое без нарушения помехоустойчивости работы подключаемых устройств может быть одновременно соединено с выходом данного элемента $K_{\text{разв}} = I_{\text{удоп}} / I_{\text{ex}}$. Причем, количественное значение коэффициента разветвления может быть различным для выходных сигналов лог.0 и лог.1.

Быстродействие цифровых микросхем характеризуют значением длительностей:



1. фронта и среза
 2. задержек распространения сигнала при включении $t_{\text{зад}}^{0,1}$ и выключении $t_{\text{зад}}^{1,0}$
- На практике обычно используют среднее время задержки распространения сигнала
- $$t_{\text{зад}} = \sqrt{t_{\text{зад}}^{0,1} + t_{\text{зад}}^{1,0}}$$

Для оценки качества элементов широко используют обобщенный параметр - работу переключения $A_{\text{П}} = P_{\text{CC}} \cdot t_{\text{зад}}$, где P_{CC} – средняя мощность, потребляемая элементом от источника питания.

В цифровой схемотехнике подавляющее число входов узлов логических устройств являются потенциальными. Эти входы характеризуются следующими особенностями:

- на входах и выходах логических элементов действуют только потенциальные сигналы;
- с выхода одного элемента на вход другого передаются как переходные, так и установившиеся значения сигналов;
- на пути потенциального сигнала не допускается включать конденсаторы и обмотки трансформаторов, поскольку длительность потенциального сигнала в общем не ограничена (такой вид электрической связи называется гальваническим или непосредственным);
- отсутствие конденсаторов и трансформаторов в цепях связи способствует упрощению технологии изготовления микросхем;
- реализуется ограниченный набор булевых функций: НЕ, ИЛИ, И, НЕ ИЛИ, НЕ И, НЕ И-ИЛИ, "Исключающее ИЛИ", что облегчает применение автоматизированных методов проектирования.

Потенциальные элементы различают по схемотехническому признаку — по способу соединения транзисторов, диодов и резисторов между собой в пределах одной схемы типового базового элемента. Принято считать, что совокупность элементов с общим признаком построения образует вид схемной логики или просто логику.

Базовые логические элементы цифровой электроники

Различают следующие виды логики потенциальных элементов:

- диодную (ДЛ) и диодно-транзисторную логику (ДТЛ);
- транзисторную логику (ТЛ), в которой выделяют схемы с непосредственными связями (ТЛНС), с резисторными связями (ТЛРС) и резисторно-конденсаторными связями (ТЛРКС);
- интегральную инжекционную логику (ИИЛ или И²Л);
- транзисторно-транзисторную логику (ТТЛ) и ее модификации с диодами Шотки (ТТЛШ);
- эмиттерно-связанную логику (ЭСЛ);
- МОП-транзисторную логику (р-МОП, л-МОП, КМОП);

Диодные логические элементы

Диодные логические элементы являются исторически первыми и самыми простыми схемами, которые реализуют булевы функции ИЛИ, И, И-ИЛИ, ИЛИ-И. Диодные элементы не усиливают входных сигналов и не могут выполнять операцию НЕ.

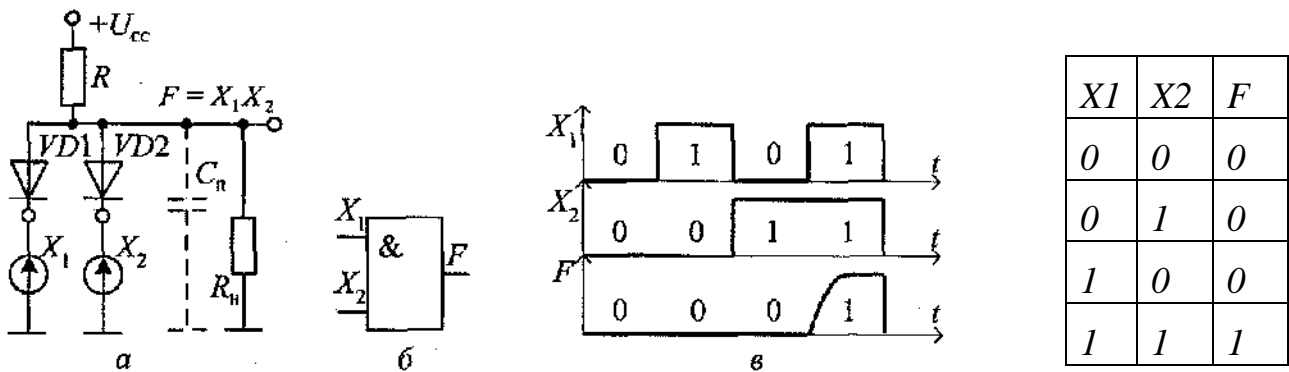


Рис.2.2

При рассмотрении работы логических элементов принимают (если не оговорено особо) положительное кодирование: высокий уровень напряжения U_H отображает лог. 1, а низкий уровень U_L — лог. 0.

Логика работы логического элемента 2И представлена в табл., на основе которой получают выражение для выходной булевой функции элемента $F = X_1 X_2$.

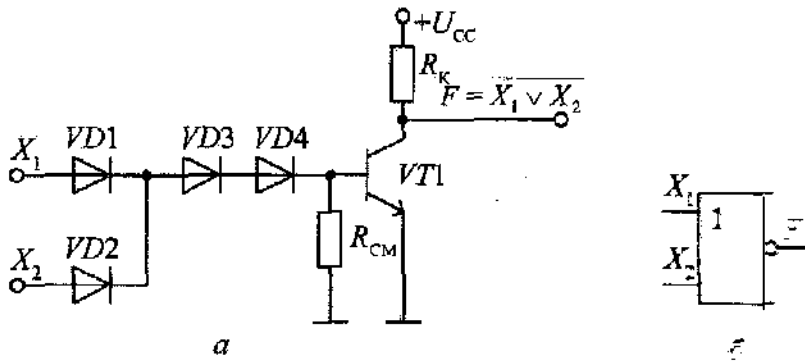
Используя принцип суперпозиции, функцию F можно записать для произвольного числа входных переменных: $F \sim X_1 X_2 \dots X_n$.

Высокий уровень напряжения U на выходе диодного элемента И устанавливается только при одновременной подаче на оба входа высоких уровней напряжения U_{IH} , при которых закрываются кремниевые диоды VD_1 и VD_2 . Выходное напряжение при этом $U_{вых} = U_{CC} R_n / (R + R_n)$.

Длительность фронта выходного сигнала t_{LH} определяется временем заряда паразитной емкости C_n через большое сопротивление резистора R .

Если на один из входов, например, X_1 подан низкий уровень напряжения, то диод VD_1 открывается. При этом на выходе устанавливается напряжение низкого уровня $U_{вых} = U_{VD0}$, где U_{VD0} — падение напряжения на открытом диоде. Длительность среза выходного сигнала определяется временем разряда паразитной емкости через малое сопротивление открытого диода. В результате время среза оказывается существенно меньше длительности фронта.

Элементы диодно - транзисторной логики

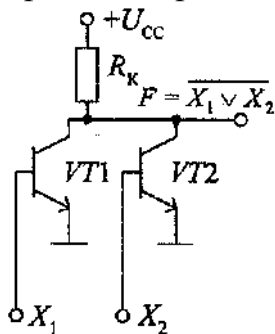


В этих элементах операции дизъюнкции и конъюнкции реализуются с помощью диодных схем, а операцию отрицания выполняет инвертор на основе транзистора: го ключа.

Логика работы двухвходового элемента НЕ ИЛИ может быть

описана в булевой функции $F = \overline{X_1 \vee X_2}$. Используя принцип суперпозиции, функцию F можно записать для произвольного числа переменных: $F = \overline{X_1 \vee X_2 \vee X_3 \vee \dots \vee X_n}$.

Транзисторная логика



Элементы ТЛ строят путем параллельного подключения коллекторов транзисторов к общему резистору коллекторной нагрузки R_k . Выходное напряжение снимается с объединенных коллекторов. Если на один из входов поступает высокий уровень напряжения U_1 , то соответствующий транзистор открывается и на выходе устанавливается низкий уровень сигнала U_0 . Высокий уровень сигнала оказывается только в случае, когда на всех входах присутствует напряжение низкого уровня.

Интегральная инжекционная логика

Разновидностью транзисторных схем являются элементы интегральной инжекционной логики (ИИЛ или И²Л). Схемотехнику И²Л используют для построения микропроцессорных и запоминающих БИС (серии К582, К583, К584 и др.).

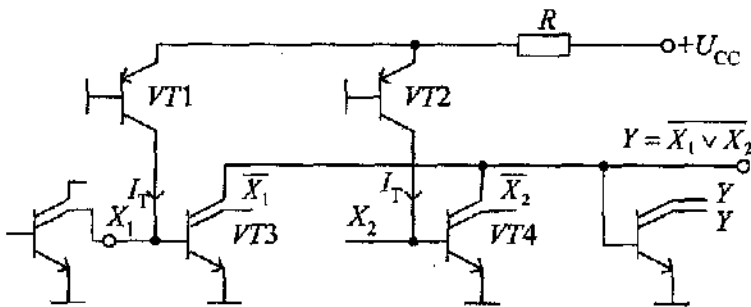


Схема включает инжекционные $p-n-p$ транзисторы $VT1, VT2$, включенные по схеме с общей базой, и входные многоколлекторные $n-p-n$ транзисторы $VT3, VT4$, включенные по схеме с общим эмиттером. Эмиттеры транзисторов $VT1, VT2$ называются инжекторами, а протекающий через них дырочный ток — инжекционным. Каждый из транзисторов $VT1, VT2$ образует вместе с источником питания и внешним резистором R источник тока, который питает индивидуальным током I_T входы транзисторов $VT3, VT4$.

Особенностями элементов И²Л являются:

- "безрезисторность", характерная для МОП-структур, которая впервые была реализована в схемотехнике И²Л;
- соединение областей базы и коллектора инжекционных $p-n-p$ транзисторов соответственно с областями эмиттера и коллектора входных $n-p-n$ транзисторов, а также малое число схемных компонентов и соединений между ними (количество операций маскирования и диффузии в два раза меньше сравнительно с элементами ТТЛ);

- низкий уровень напряжения $U_L \sim 0,01$ В снимается с коллектора насыщенного транзистора, а высокий уровень напряжения $U_H \sim 0,8$ В — с коллектора закрытого транзистора, причем этот уровень ограничивается напряжением базы насыщенного транзистора нагрузки; используется режим микротоков, в котором токи коллектора изменяются от десятков до сотен микроампер; работоспособность элементов сохраняется при изменении значения тока в них на несколько порядков;

- на коллекторах входного транзистора реализуется инверсия переменной, а на соединенных коллекторах транзисторов $VT3, VT4$ выполняются операции НЕ-ИЛИ.

Входные транзисторы управляются переключением тока на их входах. Если ко входу X_1 подключен коллектор левого насыщенного транзистора, то ток I_m замыкается на него и не поступает в базу транзистора $VT3$, который закрывается и создает на своих коллекторах режим разомкнутых контактов. Если ко входу X_1 подключен коллектор левого закрытого транзистора, то ток I_m втекает в базу $VT3$, насыщает его и обеспечивает на коллекторах режим замкнутых контактов.

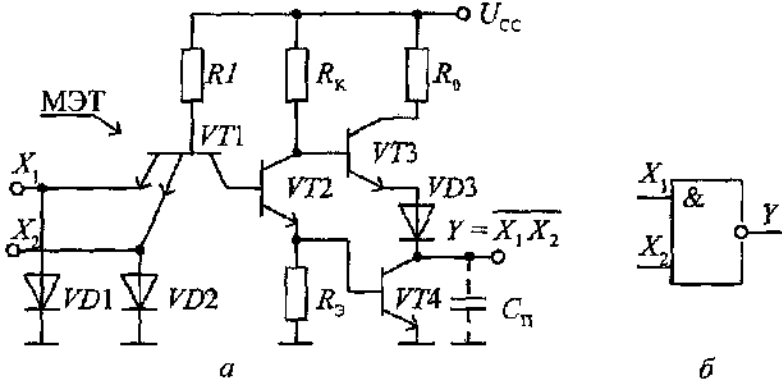
Задержка распространения сигнала в элементе И²Л при токе 100 мкА составляет примерно 5-10 нс, мощность потребления — до 20 мкВт, работа переключения равна 0,1 пДж (например, в ТТЛШ серии КР1533 работа переключения составляет 6 пДж) Отмеченные свойства элементов И²Л и БИС на их основе придают им технологичность и компактность, они имеют невысокую стоимость при большом быстродействии.

Транзисторно-транзисторный логический элемент

Базовый элемент ТТЛ

В типовых микросхемах ТТЛ используют сложные инверторы, которые повышают

быстродействие и нагрузочную способность элементов. Схема базового (типового) элемента ТТЛ со сложным инвертором содержит три основных каскада (рис. 2.13, а)



Входной каскад, реализующий операцию И (транзистор $VT1$, резистор $R1$) Ко всем входам МЭТ подключены

демпфирующие (антизвонные) диоды, ограничивающие влияние импульсов помех отрицательной полярности.

Фазоинверсный каскад (транзистор $VT2$, резисторы R_1 и R_3), управляющий выходными транзисторами с помощью противофазных изменений напряжений на коллекторе и эмиттере $VT2$.

Выходной двухканальный усилитель (транзисторы $VT3, VT4$, смещающий диод $VD3$, резистор R_6). Сложный инвертор образуется совместной работой фазоинверсного и выходного каскадов.

При совпадении на входах элементов высоких уровней напряжений МЭТ переключается в инверсный режим и своим коллекторным током открывает транзистор $VT2$. Часть эмиттерного тока транзистора $VT2$ втекает в базу транзистора $VT4$ и открывает его. После быстрого разряда паразитной емкости C_n через коллектор насыщенного транзистора $VT4$ на выходе устанавливается низкий уровень напряжения. При этом

транзистор $VT3$ — закрытый, поскольку напряжение, прикладываемое к последовательно включенным переходам базы и диода $VD3$, недостаточно для его открывания.

При подаче на один из входов напряжения низкого уровня МЭТ переключается в режим насыщения, ток его коллектора равен нулю, вследствие чего закрываются транзисторы $VT2$ и $VT4$. При этом открытый транзистор $VT3$ работает в режиме эмиттерного повторителя: на его вход поступает высокий уровень напряжения с коллектора закрытого транзистора $VT2$, а нагрузкой служит сопротивление закрытого транзистора $VT4$. Эмиттерный повторитель передает на выход высокое напряжение $U_{on} = U_{cc} - 2U^0$, где $2U^0$ — прямое падение напряжения на двух последовательно включенных переходах базы транзистора $VT3$ и диода $VD3$. Повторитель создает в нагрузке ток, в 50-100 раз превышающий его входное значение. Это также обеспечивает быстрый заряд паразитной емкости $C_{п}$.

В процессе переключения имеется кратковременный интервал, когда транзистор $VT4$ уже открыт, а $VT3$ — еще не успел закрыться. При этом возникает значительный импульс тока от источника питания на землю (сквозной ток). Для уменьшения амплитуды сквозного тока в коллекторе транзистора $VT3$ поставлен ограничительный резистор R_0 - 100...200 Ом.

В настоящее время наибольшее распространение получили элементы ТТЛ с диодами Шоттки.

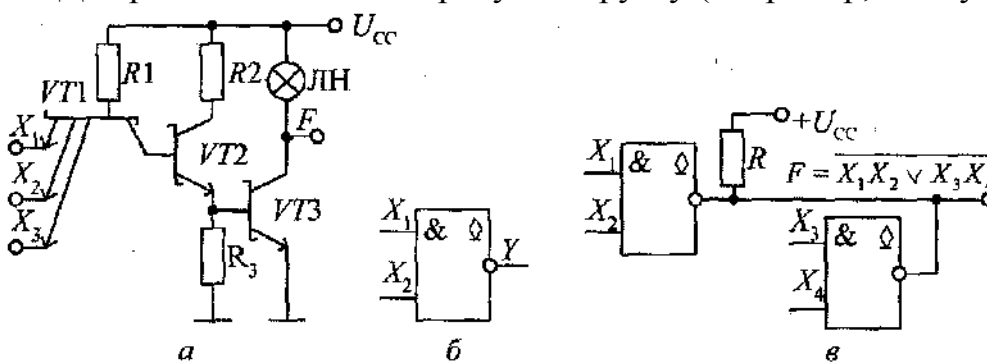
Элементы ТТЛШ в сравнении с ТТЛ имеют более высокое быстродействие и меньшую потребляемую мощность, что достигается применением диодов Шоттки, в которых накопление избыточных зарядов исключено за счет особенностей работы. Время переключения диодов Шоттки очень мало (до 0,1 нс) и не зависит от температуры. В сравнении с кремниевыми диодами прямое падение напряжения в диодах Шоттки вдвое меньше (около 0,3-0,4 В).

Достоинством ТТЛ и ТТЛШ серий является наиболее развитая элементная база. Имеется большое число готовых функциональных узлов, что существенно упрощает разработку цифровых устройств. При этом имеется большое число серий, легко сопрягаемых между собой, и оптимизированных по быстродействию и энергопотреблению для решения отдельных практических задач.

Свободные входы ТТЛ и ТТЛШ элементов эквиваленты входам, на которые поданы высокие уровни логических сигналов, но они оказываются подвержены влиянию помех.

Элементы ТТЛ и ТТЛШ с открытым коллекторным выходом и тремя состояниями

Для работы на нестандартную нагрузку (например, лампу накаливания (ЛН) или обмотку реле) выпускают схемы элементов ТТЛ и ТТЛШ с **открытым коллектором** (рис. 2.19, а).

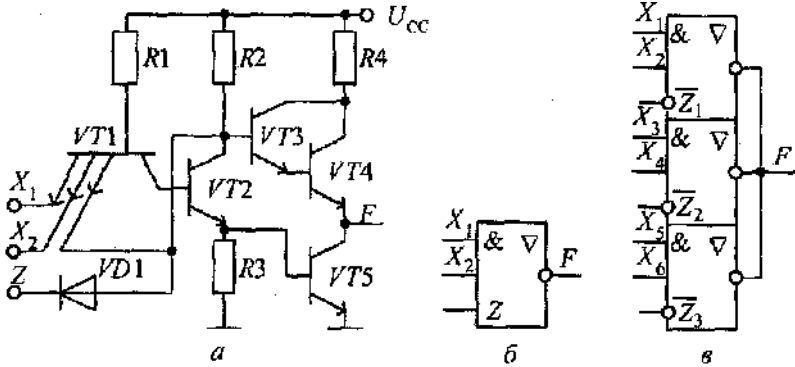


выпускают схемы элементов ТТЛ и ТТЛШ с **открытым коллектором** (рис. 2.19, а). При совпадении высоких уровней на входах транзистор $VT3$ открывается и

ЛН светится. Если хотя бы на одном из входов будет низкий уровень напряжения, то транзистор $VT3$ размыкается и ЛН гаснет.

Если выходы нескольких элементов с открытым коллектором подключить к источнику питания через общий коллекторный резистор, это позволяет дополнительно реализовать логическую функцию $F = F_1 \vee F_2 \vee \dots \vee F_n$. (монтажное ИЛИ-НЕ).

В схемах элементов ТТЛ и ТТЛШ соединение выходов нескольких элементов во



избежание протекания большого тока от выхода с высоким уровнем U_{IH} на выход с низким уровнем U_{OL} недопустимо, поскольку может привести к выходу из строя микросхемы. При необходимости такого прямого соединения выходов (способ "монтажное

ИЛИ") используют элементы с тремя состояниями.

Два состояния выходов — это обычная выдача значений лог. 0 и лог. 1. **Третье состояние (Z-состояние, состояние «выключено», высокоимпедансное состояние)** характеризуется бесконечно большим выходным сопротивлением, когда элемент практически полностью отключается от нагрузки, то есть не потребляет и не выдает тока. Это достигается рядом схемных решений. Простейшая схема приведена на рис. В том случае, когда диод $VD1$ подключен к эмиттеру и коллектору транзисторов $VT1$ и $VT2$, а на катод диода подается управляющий сигнал Z с высоким уровнем напряжения (лог. 1), схема работает как элемент НЕ И.

Если управляющий сигнал Z представлен низким уровнем напряжения (лог. 0), то МЭТ и коллектор транзистора $VT2$ (а соответственно и база транзистора $VT3$) подключаются через открытый диод $VD1$ к шине земли. В этом случае все транзисторы закрыты и элемент переходит в третье состояние (Z-состояние). В сериях микросхем часто управляющий сигнал является инверсным. При объединении выходов элементов ТТЛ с тремя состояниями управляющие сигналы $Z1$, $Z2$ и $Z3$ должны подаваться только поочередно.

Логические элементы эмиттерно-связной логики

Схемотехника элементов ЭСЛ основана на использовании дифференциального усилителя в режиме переключения тока. Элементы ЭСЛ появились в 1967 г. и в настоящее время являются самыми быстродействующими среди полупроводниковых элементов на основе кремния. Задержки распространения сигналов в элементах ЭСЛ уменьшились до субнаносекундного диапазона (приблизительно 1 нс).

Сверхбыстродействие элементов ЭСЛ достигается за счет использования ненасыщенного режима работы транзисторов, выходных эмиттерных повторителей, малых амплитуд логических сигналов (около 0,8 В). В логических элементах ЭСЛ имеется парафазный выход, что позволяет одновременно получать прямое и инверсное значение реализуемой функции. Это дает заметное снижение общего количества микросхем в аппаратуре.

Особенностями схемотехники ЭСЛ и ее характеристик являются:

- возможность объединения выходов нескольких элементов для образования новых функций;
- возможность работы на низкоомную нагрузку благодаря наличию эмиттерных повторителей;

- небольшое значение работы переключения и независимость потребляемой мощности от частоты переключения;
- высокая стабильность динамических параметров при изменении температуры и напряжения питания;
- использование отрицательного источника питания и заземления коллекторных цепей, что уменьшает зависимость выходных сигналов от помех в шинах питания.

К недостаткам элементов ЭСЛ относят сложность схем, значительное потребление мощности и трудности согласования с микросхемами ТТЛ и ТТЛШ.

Элементы на КМОП – структурах

Элементы на комплементарных МОП-транзисторах (см. «Схемные методы улучшения характеристик транзисторных ключей») характеризуются практическим отсутствием потребления энергии в статическом режиме. К достоинствам этой элементной базы относится широкий допустимый диапазон питающих напряжений (3...15 В).

Современные элементы на КМОП структурах характеризуются высоким быстродействием. Однако переключения сопровождаются перезарядом паразитных емкостей, что сопровождается «динамическим» энергопотреблением, так что при большой частоте переключений выигрыш в энергопотреблении по сравнению с ТТЛШ-элементами отсутствует.

К особенностям применения КМОП – структур относится недопустимость «висячих» входов. Большое входное сопротивление элементов делает их очень чувствительным к наводкам.

Контрольные вопросы

1. Различия между аналоговыми и цифровыми микросхемами.
2. Назначение логических микросхем. Выполняемые ими функции.
3. Передаточная характеристика Ее вид для элемента 2-И.
4. Нагрузочная характеристика. Ее вид для буферных логических элементов.
5. Коэффициент разветвления. Результат его превышения.
6. Быстродействие логических элементов. Ограничения, накладываемые этой характеристикой.
7. Помехоустойчивость логических элементов.
8. Свойства элементов ДЛ.
9. Назначение элементов ТТЛ с открытым коллектором.
10. Назначение элементов ТТЛ с третьим Z – состоянием.
11. Особенности логических элементов КМОП.

ЦИФРОВЫЕ УСТРОЙСТВА

Классификация цифровых устройств

Цифровые устройства предназначены для обработки цифровых сигналов, которые изменяются по закону дискретной функции, т.е. являются квантованными по уровню и заданы в дискретные моменты времени. Любые процессы и преобразования цифровой информации, какими бы сложными они ни были, могут быть сведены к простейшим операциям с логическими переменными 0 и 1.

Цифровые устройства классифицируют по ряду признаков, например,:

По способу ввода-вывода информации различают последовательные, параллельные и смешанные

По способу функционирования различают синхронные и асинхронные устройства

По отношению к собственному состоянию при изменении входных сигналов различают устройства с памятью (последовательностные устройства) и без памяти (комбинационные устройства).

В комбинационном устройстве (называемом также автоматом без памяти) цифровой сигнал на выходе определяется лишь значениями сигналов, действующими в данный момент времени на входах устройства, и не зависит от того, какие значения они принимали ранее. В этом смысле комбинационные устройства лишены памяти (они не хранят сведений о прошлом работы устройства).

В последовательных устройствах (или автоматах с памятью) выходной сигнал определяется не только набором символов, действующих на входах в данный момент времени, но и внутренним состоянием устройства, а последнее зависит от того, какие наборы символов действовали на входах во все предшествующие моменты времени в процессе работы устройства. Поэтому можно говорить, что последовательные устройства обладают памятью (они хранят сведения о прошлом работы устройства).

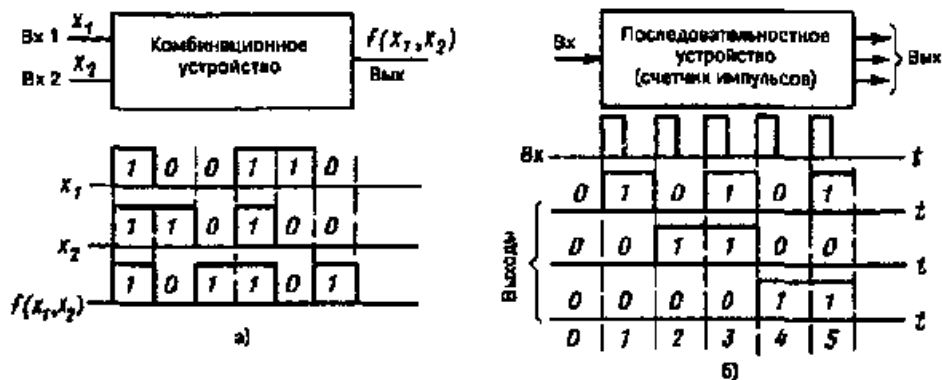


Рис 3.1

Рассмотрим примеры комбинационного и последовательного устройства. Пусть устройство (рис. 3.1.а) предназначено для формирования на выходе сигнала, определяющего совпадение сигналов на входах: на выходе формируется лог.1 в случаях, когда на обоих входах действует либо лог.1, либо лог.0; Если на одном из входов действует лог.1, а на другом — лог.0, то на выходе устройства образуется лог.0. Такое устройство является комбинационным, в котором значение формируемой на выходе логической функ-

ции определяется лишь значениями ее аргументов в данный момент времени.

Рассмотрим другой пример. Счетчик на рис. 4.1.б подсчитывает импульсы. В каждый момент времени его состояние соответствует числу поступивших на вход импульсов. Выходная информация определяется тем, каково было состояние счетчика до данного интервала времени и поступает или нет на вход импульс в данном интервале времени. Таким образом, данное устройство является последовательным устройством.

Математической основой описания работы подобных устройств является алгебра логики или булева алгебра, оперирующая с переменными, принимающими только два значения: событие истинно или событие ложно. В двоичной системе этим понятиям ставят в соответствие две цифры алфавита: 1 – событие истинно и 0 – событие ложно. Выходной сигнал как функция двоичных переменных способна принимать тоже только два значения. Функции двоичных переменных называют логическими функциями, булевыми функциями, функциями алгебры логики, переключательными функциями.

Логические функции

Функция $F = F(X_1, X_2, \dots, X_n)$ называется логической (булевой), если ее аргументы X_1, X_2, \dots, X_n , и значения самой этой функции могут принимать только два значения: логического 0 и логической 1.

Для задания функции алгебры логики, как и любой другой функции необходимо поставить в соответствие значения функции для всех возможных комбинаций входных аргументов. Если число аргументов функции равно n , то число различных сочетаний (наборов) значений аргументов составляет $m=2^n$, а число различных функций n аргументов 2^m . Так, при $n = 1$ число функций $2^1 = 2$, при $n = 2$ число функций $2^2 = 4$, при $n = 3$ число функций $2^3 = 8$, при $n = 4$ число функций $2^4 = 16$, при $n = 5$ число функций $2^5 = 32$, при $n = 6$ число функций $2^6 = 64$, при $n = 7$ число функций $2^7 = 128$, при $n = 8$ число функций $2^8 = 256$.

Существует три способа представления логических функций, используемые на различных этапах разработки логической модели цифрового устройства. В процессе проектирования, прodelьывают переходы от одного способа представления функции к другому так, чтобы оказалось возможным построение логической модели.

1. Словесный способ. Взаимосвязь значений функции и ее аргументов описывается словесной формулировкой. Например, мажоритарная функция «два из трех»: значение функции принимается «большинством голосов» - «лог.1» на выходе возможна только в случае, когда она есть, по крайней мере, на двух входах из трех.

2. Табличный способ.

Таблица функционирования или таблица истинности полностью определяет значения функции для любой возможной комбинации сигналов на входе устройства. Для цифровых схем число возможных наборов всегда конечно, что отличает их от таблично заданных математических функций, предполагающих существование неограниченного числа промежуточных значений.

№ набора	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Комбинации входных сигналов называют наборами и часто для сокращения записи их нумеруют и используют при записи для представления наборов лишь их номера. Для введенного «мажоритарного элемента», например, таблица истинности имеет вид

3. Аналитический способ. Логическая функция записывается в виде аналитического выражения, где показаны логические операции, выполняемые над аргументами функции.

Аналитический способ используется для записи структурной формулы, отражающей работу логического устройства. Логическая функция представляется совокупностью наборов аргументов, при которых эта функция должна принимать значение логической 1. В частности, для рассматриваемого «мажоритарного элемента» можем структурную формулу можем представить в виде

$$F = \bar{A} \wedge B \wedge C \vee A \wedge \bar{B} \wedge C \vee A \wedge B \wedge \bar{C} \vee A \wedge B \wedge C$$

Подобно обычной математической записи аналитический способ задания логической функции показывает, какие и в какой последовательности над аргументами должны выполняться логические операции. Как и в арифметике, при записи структурной формулы используется система приоритетов логических операций («НЕ», «И», «ИЛИ»).

При записи структурной формулы возможны две формы записи логических выражений.

Дизъюнктивной нормальной формой (ДНФ) называется логическая сумма элементарных логических произведений, в каждое из которых аргумент или его отрицание входят один раз. ДНФ может быть получена из таблицы истинности следующим образом: Для каждого набора аргументов, на котором функция равна «1», записывают элементарные произведения переменных, причем переменные, значения которых равны нулю, записывают с инверсией. Полученные произведения, которые носят название *конъюнкта единицы*, или *минтермов*, суммируют. Если в каждой конъюнкции представлены все аргументы функции (или их инверсии), такая форма называется *совершенной дизъюнктивной нормальной формой (СДНФ)*. Каждая логическая функция имеет единственную СДНФ.

Конъюнктивная нормальная форма (КНФ) – представление логической функции в виде логического произведения (конъюнкции) элементарных логических сумм (дизъюнкций). КНФ может быть получена из таблицы истинности - для каждого набора, при которых функция принимает нулевые значения, составляется дизъюнкция. При этом аргументы, имеющие значение 1, записываются с инверсией. Если в каждой дизъюнкции представлены все аргументы функции (или их инверсии), такая форма называется *совершенной конъюнктивной нормальной формой (СКНФ)*. Каждая логическая функция имеет единственную СКНФ.

СДНФ и СКНФ записывают непосредственно из таблицы истинности.

СДНФ имеет столько конъюнкций, сколько единичных значений принимает функция. Для каждого набора переменных, для которого функция принимает единичное значение, составляются элементарные конъюнкции. Если в наборе переменная принимает нулевое значение, то ее записывают с инверсией. Все конъюнкции затем логически суммируют.

СКНФ имеет столько дизъюнкций, сколько нулевых значений у логической функции. Дизъюнкции соответствуют наборам, для которых значение логической функции равно нулю, так что если переменная входит в нее с единичным значением, то используют инверсию этой переменной. Дизъюнкции логически перемножаются.

Схемная реализация цифровых устройств

При проектировании цифровых устройств для учета технических особенностей работы устройства используются 3 модели:

1. логическая модель (описывает работу устройств в предположении идеальности параметров используемой элементной базы)
2. модель с временными задержками (учитывает конечное быстродействие элементов схем и позволяет предусмотреть и устранить возможность сбоев в работе)
3. модель с учетом электрических характеристик применяемых элементов (используется при детализации принципиальной схемы устройства).

Логическая модель цифрового устройства является схемной реализацией алгоритма работы устройства. Работа цифровых устройств описывается с помощью *теории автоматов* – раздела теории управляющих систем, изучающей математические модели преобразователей дискретной информации. При проектировании цифровых устройств для упрощения процесса разработки часто используется метод декомпозиции - представление цифрового устройства совокупностью независимо функционирующих логических устройств для формирования отдельных бит выходного сигнала.

Законы булевой алгебры позволяют выразить любую сколь угодно сложную логическую функцию с помощью всего лишь трех элементарных логических функций («НЕ», «И», «ИЛИ»), образующих функционально полную систему логических элементов. При реализации цифровых устройств оказываются возможными различные функционально полные системы логических функций, отличающиеся друг от друга числом входящих в них функций и видом этих функций. Выбор того или иного базиса для построения логических устройств обусловлен тем, насколько экономически удобно и просто выполнить элементы, технически реализующие входящие в базис функции, и все логическое устройство в целом.

Для одной переменной существует всего 4 логические функции, причем устройства, реализующие 3 из них, оказываются тривиальными

аргумент x	функция			
	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	1
1	0	1	0	1

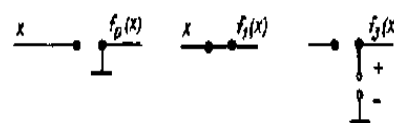


Рис 3 3

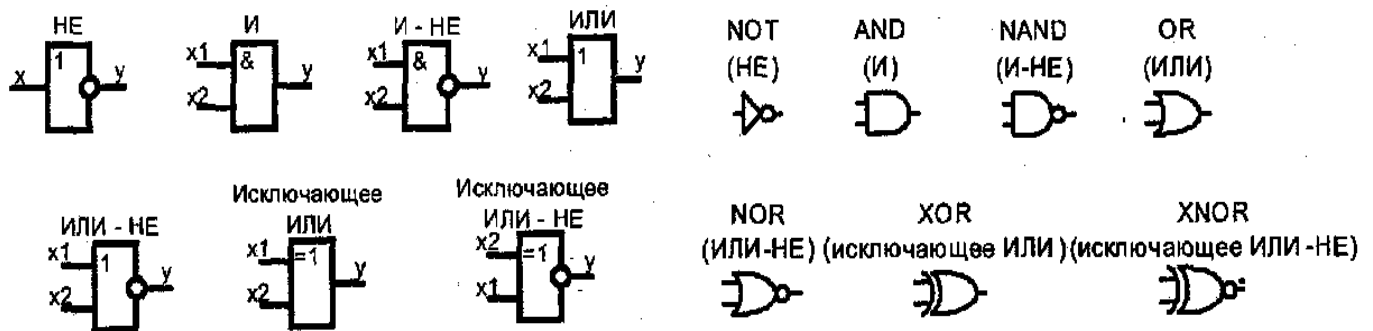
Для двух переменных существует уже 16 логических функций.

Аргументы		Функции															
x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Логические функции одной и двух переменных называются элементарными, так как не содержат более одной логической операции. Наиболее широко используемые логические функции двух переменных приведены в таблице

Обозначение логических операций		Таблица истинности					Как читается	Название операции
		x_1	0	0	1	1		
Основное	Дополнительные	x_2	0	1	0	1		
$x_1 \cdot x_2$	$x_1 x_2$ $x_1 \wedge x_2$ $x_1 \& x_2$	$x_1 \cdot x_2$	0	0	0	1	x_1 и x_2	Конъюнкция; логическое И; логическое произведение
$x_1 \vee x_2$	$x_1 + x_2$	$x_1 \vee x_2$	0	1	1	1	x_1 или x_2	Дизъюнкция; логическое ИЛИ; логическая сумма
$x_1 \rightarrow x_2$	$x_1 \supset x_2$	$x_1 \rightarrow x_2$	1	1	0	1	если x_1 , то x_2 ; x_1 влечет x_2 ; x_1 имплицирует x_2	Импликация
$x_1 \Leftrightarrow x_2$	$x_1 \equiv x_2$ $x_1 \leftrightarrow x_2$	$x_1 \Leftrightarrow x_2$	1	0	0	1	x_1 эквивалентно x_2	Эквивалентность; равнозначность
$x_1 \oplus x_2$	$x_1 \oplus x_2$	$x_1 \oplus x_2$	0	1	1	0	либо x_1 , либо x_2 ; x_1 неэквивалентно x_2	Сумма по модулю; неравнозначность; исключающее ИЛИ
$x_1 \Delta x_2$	$x_1 \nrightarrow x_2$ $x_1 \not\supset x_2$	$x_1 \Delta x_2$	0	0	1	0	x_1 запрет по x_2 ; x_1 , но не x_2	Запрет; отрицание импликации
$x_1 x_2$	—	$x_1 x_2$	1	1	1	0	x_1 и x_2 несовместны	Логическое И-НЕ; элемент (штрих) Шеффера; отрицание конъюнкции
$x_1 \downarrow x_2$	—	$x_1 \downarrow x_2$	1	0	0	0	ни x_1 , ни x_2	Логическое ИЛИ-НЕ; стрелка Пирса; функция Вебба; отрицание дизъюнкции
\bar{x}	$\neg x$	x	0	1			не x	Логическое НЕ; инверсия; логическое отрицание
		\bar{x}	1	0				

Ниже даются условные графические обозначения (УГО) в отечественной и зарубежной документации наиболее распространенных из них.



Через три логические функции инверсии (НЕ), конъюнкции (И), дизъюнкции (ИЛИ) можно выразить любую элементарную функцию и построить любое сложное логическое устройство. Вместе с тем, оказывается, что функционально полная система логических операций может быть реализована и с помощью других элементарных функций. Последовательно исключая элементы, которые можно выразить через другие, получают минимальный базис. Под минимальной функционально полной системой понимают такой набор функций, который обладает функциональной полнотой, но нарушаемой при исключении любой из составляющих ее функций.

Набор из трех функций (НЕ, И, ИЛИ) является функционально полной системой функций, однако, не являющейся минимальной. Из него можно исключить одну из функций И либо ИЛИ. Наборы (НЕ, И), а также (НЕ, ИЛИ) из двух функций служат базисами. Действительно функцию И можно реализовать через функции (НЕ, ИЛИ): $a \wedge b = \overline{\overline{a} \vee \overline{b}}$, Функцию ИЛИ можно реализовать через функции (НЕ, И): $a \vee b = \overline{\overline{a} \wedge \overline{b}}$. Ограничиваясь базисами (НЕ, И); (НЕ, ИЛИ) для выполнения исключенной одной операции требуется проводить дополнительно три операции НЕ, что

экономически нецелесообразно. Поэтому на практике часто используют неминимальный базис трех функций (НЕ, И, ИЛИ).

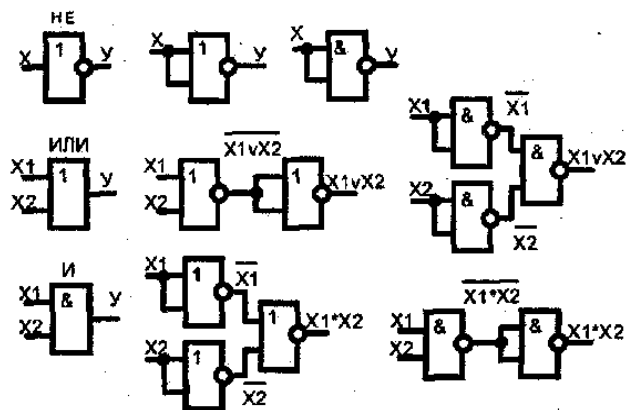


Рис.3.4.

что все логическое устройство построено только на одностипных логических элементах. Получаем логическую схему, которая обладает регулярной структурой. Необходимо только осуществить коммутацию одинаковых логических элементов. Базисы на логических элементах (И-НЕ), (ИЛИ-НЕ) широко используются при проектировании устройств, удобны для реализации в больших интегральных схемах. Уменьшение номенклатуры до одного типа, таким образом, облегчает проектирование устройств. Дополнительные инверторы на входах улучшают нагрузочную способность и служат для формирования сигналов лог. 1 и лог. 0 достаточного уровня.

Принципиальная схема представляет собой аппаратную реализацию структурной формулы устройства, которая должна осуществить необходимые преобразования

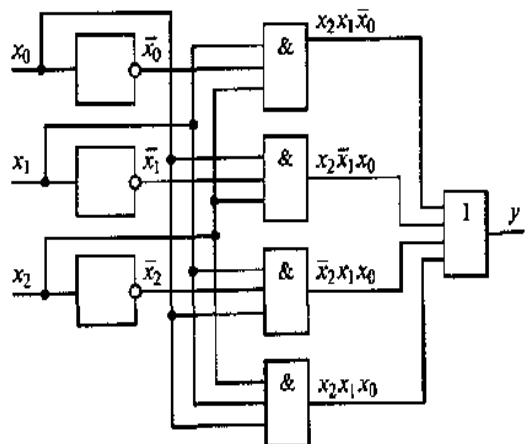


Рис.3.2.

над входными сигналами, соблюдая их последовательность. Вариант схемы приведен на рис.3.2.

Минимизация логических функций

Сложные логические функции обычно допускают неоднозначное представление структурной формулы, служащей основой при аппаратной реализации цифровых устройств. В связи с этим возникает задача отыскания наиболее удобной формы представления структурной формулы, обеспечивающей оптимальное решение для проектируемого цифрового устройства. Оптимальность разрабатываемого устройства определяется предъявляемыми к нему требованиями по энергопотреблению, быстродействию, количеству корпусов дискретных микросхем и т.д. При этом обычно определяют наиболее простое решение, которое корректируется с учетом обеспечения требуемых характеристик, выбора соответствующей элементной базы, минимизации числа соединений, повышения надежности устройств.

Довольно удобно технически реализуются на микросхемах логические элементы, совмещающие в себе указанные функции. Удобство объясняется тем, что транзистор инвертирует фазу входного сигнала, изменяет ее на 180°. Это элементы И-НЕ (штрих Шеффера) и ИЛИ-НЕ (стрелка Пирса).

Каждый из элементов (И-НЕ), (ИЛИ-НЕ) в отдельности является функционально полным базисом, позволяет синтезировать любое сколь угодно сложное устройство. Реализация функций НЕ, И, ИЛИ в базисе (ИЛИ-НЕ) и (И-НЕ) приведена на рис.3.4.

Привлекательность базисов из одной логической функции (И-НЕ) либо (ИЛИ-НЕ) заключается в том,

над входными сигналами, соблюдая их последовательность.

Возможны несколько вариантов составления принципиальной схемы. Продемонстрируем один из способов на примере «мажоритарного элемента», структурная формула которого получена ранее в виде $F = A \wedge B \wedge C \vee A \wedge \bar{B} \wedge C \vee A \wedge B \wedge \bar{C} \vee A \wedge \bar{B} \wedge \bar{C}$. Выходной сигнал F представляет собой логическую сумму 4 слагаемых, что реализуется с помощью логической схемы «ИЛИ». На входы этого логического элемента необходимо подать 4 логических произведения из 3 сомножителей, что требует использования 4 схем «ЗИ». На входы этих схем логического умножения поступают входные сигналы, причем на некоторые входы

Элементы булевой алгебры

Производство логических элементов, реализующих все возможные логические функции, оказывается нецелесообразным. Тем более, что с ростом числа переменных число логических функций быстро растет. Это объясняется тем, что любая сколь угодно сложная функция может быть реализована с помощью ограниченного числа элементарных логических функций, в частности, комбинации только элементов «НЕ», «И», «ИЛИ». Подобные наборы элементарных логических элементов, позволяющие реализовать любую логическую функцию, называют функционально полными наборами. Это оказывается возможным путем использования специфических законов булевой алгебры.

В сложных логических выражениях устанавливается определенный порядок выполнения операций *операции в скобках*, «НЕ», «И», «ИЛИ». Во вложенных скобках сначала выполняются операции во внутренних скобках.

Над логическими выражениями производят тождественные преобразования с использованием законов булевой алгебры.

Функции являются эквивалентными, если они принимают одинаковые значения при одних и тех же наборах входных сигналов.

Две эквивалентные функции, приравненные друг к другу, называются тождеством.

Булева алгебра, оперирующая лишь с логическими сигналами и операциями, характеризуется принципом двойственности.

Если в правилах выполнения логического сложения все переменные и саму функцию заменить на инверсные значения, а операцию сложения заменить на операцию умножения, то равенство в правилах выполнения логических операций сохраняется

$$f = \overline{xy} \vee x\overline{y} \qquad \overline{f} = \overline{\overline{xy} \vee x\overline{y}}$$

Осуществление тождественных преобразований логических функций базируется на специфических теоремах булевой алгебры.

Первые девять теорем относятся к функциям одной переменной, остальные к функциям двух и большего числа переменных. Все теоремы легко доказать путем перебора всех возможных комбинаций переменных.

1 $x \vee 0 = 1$	2 $x \vee 1 = 1$	3 $x \vee x = x$	4 $x + \overline{x} = 1$
5 $x \wedge 0 = 0$	6 $x \wedge 1 = x$	7 $x \wedge x = x$	8 $x \wedge \overline{x} = 0$
9 $\overline{\overline{x}} = x$			

Следующие теоремы, содержащие две и большее число переменных, имеет каждая два варианта: для логического сложения и для логического умножения.

10. $x \vee y = y \vee x$	$x \wedge y = y \wedge x$ («переместительный закон».)
11. $\overline{\overline{x \vee y} \vee z} = x \vee \overline{\overline{y \vee z}}$	$\overline{\overline{x \wedge y} \wedge z} = x \wedge \overline{\overline{y \wedge z}}$ («сочетательный закон»).
12. $x \wedge \overline{\overline{y \vee z}} = x \wedge y \vee x \wedge z$	$x \vee \overline{\overline{y \wedge z}} = (x \wedge y) \vee (x \wedge z)$
13. $x \vee x \wedge y = x$	$x \wedge \overline{\overline{x \vee y}} = x$ («закон поглощения»).
14. $x \vee \overline{x} \wedge y = x \vee y$	$x \wedge \overline{\overline{x \vee y}} = x \wedge y$ («закон свертки»)
15. $\overline{\overline{x \wedge y \vee x \wedge \overline{y}}} = x$	$\overline{\overline{\overline{x \vee y} \wedge \overline{\overline{x \vee y}}}} = x$ («закон склеивания»)
16. $\overline{\overline{x \vee y \vee z \vee \dots}} = \overline{\overline{x \wedge y \wedge z \wedge \dots}}$ $x \wedge y \wedge z \wedge \dots = \overline{\overline{x \vee y \vee z \vee \dots}}$ (теорема де Моргана)	

Для переменной и функции справедлива теорема

$$17. \quad \overline{\overline{x \vee F(x, y, z, \dots)}} = x \vee F(x, y, z, \dots) \qquad \overline{\overline{x \wedge F(x, y, z, \dots)}} = x \wedge F(x, y, z, \dots)$$

Справедливость всех приведенных законов может быть доказана прямой подстановкой.

Минимизацию логических выражений для упрощения структурной формулы можно проводить непосредственно с использованием законов булевой алгебры.

Предварительно введем следующие определения:

I. Два произведения логических переменных называются соседними, если они отличаются значением только одной переменной. Например, произведения ABC и $AB\bar{C}$ являются соседними, т.к. переменные A и B входят в оба произведения без знаков инверсии, а переменная C входит в одно произведение с инверсией, в другое – без инверсии.

II. Количество переменных в произведении называют его рангом.

С использованием этих определений эти правила формулируются следующим образом.

1. Два соседних слагаемых K-го ранга можно заменить одним слагаемым (K-1)-го ранга. Действительно, $A \wedge B \wedge \bar{C} \vee A \wedge B \wedge C = A \wedge B \wedge (C \vee \bar{C}) = A \wedge B$.

2. Поскольку сумма любого числа одинаковых слагаемых равна этому слагаемому, то из m одинаковых слагаемых (m-1) слагаемое являются лишними (избыточными) и их можно отбросить. И, наоборот, что к логической сумме можно добавить еще сколько угодно слагаемых, повторяющих уже в этой сумме содержащихся.

3. Любой несовершенный дизъюнктивный вид записи структурной формулы можно расширить до совершенного вида путем умножения слагаемых на выражение $(x \vee \bar{x})$. Например,

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

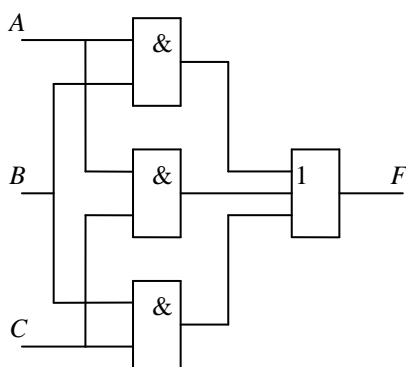
4. Если функция, представленная в СНДФ содержит 2^n слагаемых, где n- количество переменных, то она тождественно равна единице. И обратное: если функция тождественно равна единице $F=1$, то она имеет все 2^n слагаемых и на каждом из 2^n наборов функция принимает значение 1.

5. Если функция F, записанная в СНДФ содержит m слагаемых, то ее отрицание \bar{F} должно содержать остальные $2^n - m$ слагаемых.

Рассмотрим «мажоритарный элемент». Законы булевой алгебры позволяют преобразовать его структурную формулу к виду

$F = A \wedge B \wedge C \vee A \wedge \bar{B} \wedge C \vee A \wedge B \wedge \bar{C} \vee A \wedge \bar{B} \wedge \bar{C} \vee (A \wedge B \wedge C \vee A \wedge B \wedge \bar{C})$, где выражение в скобках представляет собой повторение одного из логических слагаемых. Последовательно производя операцию логического сложения с учетом одного из законов булевой алгебры $x + x = 1$, можем получить

$$F(A,B,C) = \bar{A}BC + A\bar{B}C + \underline{ABC} + \underline{ABC} + ABC + \underline{ABC} = BC + AC + AB, \text{ что, в свою очередь, предполагает достаточно простую его схемную реализацию.}$$



Существует ряд алгоритмов минимизации структурной формулы непосредственно в аналитическом виде, но наиболее наглядным является метод, использующий графическое представление таблицы истинности.

Существует ряд алгоритмов минимизации структурной формулы непосредственно в аналитическом виде, но наиболее наглядным является метод, использующий графическое представление таблицы истинности.

Диаграммы Вейча и карты Карно

Наиболее известными методами для этих целей являются карты Карно и диаграммы Вейча, различие между которыми сводится лишь к различию в графическом представлении таблицы истинности.

сти. Эти методы широко используются для ручной, без применения ЭВМ, минимизации логических функций, число аргументов которой не превышает 4...5.

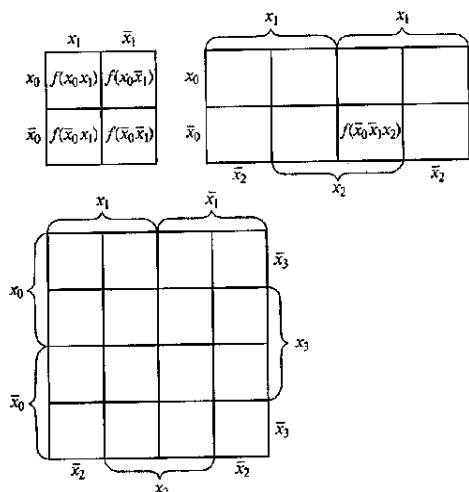


Рис.4.1.

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

Из приведенных рисунков видно, что минтерм представляется минимальным участком площади - одной клеткой на картах Карно (картах минтермов). При этом в смежных клетках наборы оказываются соседними. Соседними же являются наборы, находящиеся в верхней и нижней строках и в правом и левом столбце. Линиями за границами прямоугольника указаны столбцы и строки, в которых соответствующие переменные встречаются в наборах без инверсии. В других столбцах и строках эти переменные входят в наборы с инверсией. Остальные клетки остаются незаполненными или заполняются

нулями. В результате логическая функция представляется совокупностью клеток, заполненных единицами, а инверсия функции – совокупностью нулей или пустых клеток.

Алгоритм минимизации сводится к следующему:

1. На карте выделяют прямоугольные области, объединяющие клетки, в которых функция принимает единичное значение. Каждая область объединения (контур) может содержать только 2^k – клеток (1, 2, 4, 8, 16 и т.д.). При этом контура могут неоднократно пересекаться – каждая клетка может входить в несколько контуров.
2. Каждая из выделенных областей является самостоятельным произведением переменных, значения которых в рамках выделенной области остаются постоянными.
3. Из полученного множества выделенных областей выбирают минимальное количество максимально больших областей. В этом случае сумма полученных произведений образует минимальную структурную формулу.

Рассмотрим пример

$$F(A, B, C, D) = \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D} \vee \bar{A} \wedge B \wedge \bar{C} \wedge \bar{D} \vee \bar{A} \wedge B \wedge C \wedge \bar{D} \vee \bar{A} \wedge B \wedge C \wedge D \vee A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D} \vee A \wedge \bar{B} \wedge C \wedge \bar{D} \vee A \wedge \bar{B} \wedge C \wedge D \vee A \wedge B \wedge C \wedge \bar{D} \vee A \wedge B \wedge C \wedge D$$

Введем эту функцию в диаграмму Вейча, рис.4.6.

Клетки, содержащие единицы, в которых в диаграмме находились соседние наборы, объединяются контурами по следующим правилам:

	A			
		1		1
B		1	1	1
		1	1	
	1	1		1

C

D

1. Если единицы занимают две соседние строки или два соседних столбца, в том числе расположенные по краям диаграммы и разделенные внешними границами, то они могут быть объединены контурами и т.к. такой контур содержит восемь клеток, слагаемые четвертого ранга, находящиеся в этих клетках могут быть заменены одним слагаемым первого ранга, т.е. той переменной, которая является для них общей.
2. Если объединяются четыре квадрата, то четыре слагаемых четвертого ранга заменяются одним слагаемым второго ранга, т.е. переменными, которые являются для них общими.

Упрощение получается за счет того, что общие переменные можно вынести за скобки, а в скобках остается полный набор из оставшихся переменных, который всегда равен 1, но чтобы это произошло, объединять можно только 2^n клеток. В ином случае в скобках не получится полного набора, который можно заменить 1.

3. Если объединяются два квадрата, то слагаемые четвертого ранга заменяются одним слагаемым третьего ранга. Одна и та же единица может входить во сколько угодно контуров. Всегда нужно стремиться объединить большее число клеток с единицами и получить при этом наименьшее число контуров. Все единицы должны быть объединены контурами, даже если какая-либо единица не может быть объединена с другими. В этом случае контуром обводится одна клетка и упрощения слагаемого не получается. Далее из каждого контура выписываются одинаковые переменные. Количество слагаемых в получившейся минимизированной формуле должно совпасть с количеством контуров.

Проделав все действия в приведенном примере, получим следующую минимизированную функцию:

$$F = AC + CD + \overline{ABC} + \overline{BCD}.$$

При использовании диаграммы Вейча или карты Карно можно добиться более существенной минимизации, если использовать условие избыточности. Это условие означает, что некоторые наборы переменных на входах проектируемого устройства никогда не встречаются. Например, бывает известно, что цифровое устройство работает в пределах первого десятка. Для этого необходимо четыре двоичных разряда, из четырех переменных можно составить шестнадцать различных наборов, но наборы, соответствующие десятичным числам от 10 до 15 никогда не будут присутствовать на входах устройства и поэтому совершенно безразлично, что будет на выходе устройства при таких наборах на входах.

Это условие позволяет поставить в клетки диаграммы, где находятся эти наборы, как 0, так и 1 в зависимости от того, что обеспечит лучшую минимизацию, т.е. меньшее количество контуров и большее количество клеток в каждом из них. Например, пусть задана функция, которую нужно минимизировать:

$$F = \overline{ABD} + ABC\overline{D} + \overline{ABC}D.$$

При этом известно, что наборы $\overline{ABC}\overline{D}$, \overline{ABC} и \overline{ABD} никогда не встречаются при работе устройства, т.е. как говорят, являются избыточными. При вводе заданной функции в диаграмму Вейча поставим в клетки, в которых находятся избыточные наборы знак

Часто бывает выгоднее минимизировать не собственно функцию F , а ее инверсию. В этом случае объединяются в контуры нули, что обеспечивает минимизацию функции \overline{F} . Для получения искомой функции достаточно включить дополнительный инвертор, реализующий путем дополнительной операции отрицания получение требуемого результата.

Существует ряд алгоритмов минимизации структурной формулы при аналитической форме ее представления, реализуемые с помощью компьютеров. Кроме того, разработаны алгоритмы минимизации цифровых устройств, позволяющие устранить избыточность аппаратуры, неизбежную при ее проектировании методом декомпозиции.

Синтез логических устройств в заданном базисе логических элементов

На практике в целях уменьшения номенклатуры часто пользуются функционально полной системой логических элементов в составе двух, выполняющих операции И-НЕ, ИЛИ-НЕ.

Если задан базис И-НЕ, то путем двойного инвертирования исходного выражения или его части и применения теорем де Моргана логические функции приводятся к виду, содержащему только операции логического логического умножения и инвертирования. Если же задан базис ИЛИ-НЕ, исходную логическую функцию теми же приемами приводят к виду, содержащему только операции логического сложения и инверсии. Далее логическое выражение записывается через условные обозначения выбранных операций

Решение подобных задач может быть проиллюстрировано на примерах.

Задана логическая функция, приведем ее к базису И-НЕ:

$$\begin{aligned}
 y &= x_0 x_3 + (\bar{x}_0 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2); \\
 y &= y_1 + \bar{y}_2 y_3 = y_1 + \overline{\bar{y}_2 y_3} = y_1 + \overline{(\bar{y}_2 | y_3)} = y_1 + \overline{\overline{(\bar{y}_2 | y_3)}} = \\
 &= \overline{\bar{y}_1 (\bar{y}_2 | y_3)} = \bar{y}_1 | (\bar{y}_2 | y_3); \\
 \bar{y}_1 &= \overline{x_0 x_3} = x_0 | x_3; \\
 \bar{y}_2 &= \overline{\bar{x}_0 + x_2 + \bar{x}_3} = x_0 \bar{x}_2 x_3 = \overline{x_0 \bar{x}_2 x_3} = x_0 | \bar{x}_2 | x_3; \\
 y_3 &= x_1 + \bar{x}_2 = \overline{x_1 + \bar{x}_2} = \bar{x}_1 x_2 = \bar{x}_1 | x_2; \\
 y &= (x_0 | x_3) | ((x_0 | \bar{x}_2 | x_3) | (\bar{x}_1 | x_2)).
 \end{aligned}$$

Пример приведения к базису ИЛИ-НЕ

$$\begin{aligned}
 y &= y_1 + \bar{y}_2 y_3 = y_1 + \overline{\bar{y}_2 y_3} = y_1 + \overline{(y_2 + \bar{y}_3)} = y_1 + \overline{(y_2 + \bar{y}_3)} = y_1 \downarrow (y_2 \downarrow \bar{y}_3); \\
 y_1 &= x_0 x_3 = \overline{\bar{x}_0 \bar{x}_3} = \bar{x}_0 + \bar{x}_3 = \bar{x}_0 \downarrow \bar{x}_3; \\
 y_2 &= \bar{x}_0 + x_2 + \bar{x}_3 = \overline{\bar{x}_0 + x_2 + \bar{x}_3} = \bar{x}_0 \downarrow x_2 \downarrow \bar{x}_3; \\
 y_3 &= x_1 + \bar{x}_2 = \overline{x_1 + \bar{x}_2} = x_1 \downarrow \bar{x}_2 \Rightarrow \bar{y}_3 = x_1 \downarrow \bar{x}_2; \\
 y &= (\bar{x}_0 \downarrow \bar{x}_3) \downarrow ((\bar{x}_0 \downarrow x_2 \downarrow \bar{x}_3) \downarrow (x_1 \downarrow \bar{x}_2)).
 \end{aligned}$$

Контрольные вопросы

1. Логические функции. Формы их представления.
2. Определение элементарной логической функции.
3. Теоремы де Моргана.
4. Определение минимального функционально полного набора логических функций.
5. Цель минимизации логических функций в цифровой электронике.
6. Определение структурной формулы устройства. Ее отыскание.
7. Принципы минимизации структурных формул.
8. Теоремы булевой алгебры, используемые при минимизации логических функций.
9. Определение ранга для СНДФ.
10. Достоинства минимизации с помощью диаграмм Вейча.
11. Требования к контурам при минимизации с помощью диаграмм Вейча.

ФУНКЦИОНАЛЬНЫЕ УЗЛЫ КОМБИНАЦИОННОГО ТИПА

Одним из наиболее удобных методов проектирования цифровых устройств является сведение сложного алгоритма, определяющего необходимые преобразования входного цифрового сигнала, сочетанием ряда более простых. Это объясняется, с одной стороны, трудностями и громоздкостью анализа при формализации задания на устройство, а с другой – возможностью интуитивного его сведения к последовательности типовых преобразований. К преимуществам подобного подхода к разработке цифровых схем относится наличие развитой элементной базы, содержащей большое число готовых типовых функциональных узлов в своем составе.

Выпускаемые в виде законченных ИМС функциональные узлы, как и логические элементы, характеризуется ограниченным числом входов. Более того, часто необходимое преобразование сигнала не полностью реализуется возможностями выбранной ИМС. В результате часто оказывается, что реализация этих устройств на логических элементах предпочтительнее, что особенно существенно при разработке нетиповых узлов. Поэтому знание методов решения некоторых типовых задач цифровой схемотехники позволяет заметно упростить процесс разработки новых устройств.

Комбинационными называют цифровые устройства, в которых значения выходных сигналов определяются присутствующей в данный момент времени комбинацией входных сигналов..

Типовые функциональные узлы

Дешифраторы

Кодирующим устройством называют логический узел, преобразующий многоразрядный входной код в выходной код, построенный по иному закону. Название в большей мере условно, поскольку любое цифровое устройство преобразует некоторый входной код в некоторый выходной, т.е. является кодовым преобразователем. Традиционно это название применяется к узлам, работа которых не описывается достаточно простым алгоритмом, как, например, работа сумматора, а задается таблицей соответствия входов и выходов. В таком смысле термин и будет применяться в дальнейшем.

Дешифратором или декодером (*decoder*) чаще всего называют кодирующее устройство, преобразующее двоичный код в унитарный (из всех m выходов дешифратора активный уровень имеется только на одном, номер которого равен поданному на вход двоичному числу. На всех остальных выходах дешифратора уровни напряжения неактивные). Условное изображение дешифратора на схемах показано на рис. 5.1, а.

О разрешающем входе E будет сказано ниже.

Если декодер имеет n входов, m выходов и использует все возможные наборы входных переменных, то $m=2^n$. Такой декодер называют полным в отличие от неполного, использующего лишь часть возможных наборов и имеющего соответственно меньшее число выходов и внутренних схемных элементов.

Декодер используют, когда нужно обратиться к различным цифровым устройствам, и при этом номер устройства - его *адрес* - представлен двоичным кодом.

Входы декодера (их иногда называют *адресными входами*) часто нумеруют не порядковыми номерами, а в соответствии с весами двоичных разрядов, т.е. не 1, 2, 3, 4, 5, ..., а 1, 2, 4, 8, 16... Число входов и выходов декодера указывают таким образом: декодер 3—8 (читается "три в восемь"); 4—16; 4—10 (это неполный декодер).

Формально описать работу дешифратора можно, задав список функций, обрабатываемых каждым из его выходов Y_k . Так, для дешифратора 3—8

$$Y_0 = a_4 \wedge a_2 \wedge a_1 \quad Y_1 = \bar{a}_4 \wedge a_2 \wedge a_1 \quad Y_2 = a_4 \wedge \bar{a}_2 \wedge a_1 \quad \dots \quad Y_7 = \bar{a}_4 \wedge \bar{a}_2 \wedge \bar{a}_1$$

Реализация этих восьми выражений с помощью восьми трехвходовых элементов И дает наиболее простой по структуре дешифратор, называемый *линейным*, (рис. 5.1,б). Основной объем его оборудования составляют в общем случае m n -входовых элементов И. Кроме того, к оборудованию дешифратора обычно относят n инверторов входных переменных и n буферных входных усилителей, обычно инвертирующих, что характерно для интегральной технологии.

Назначение буферных усилителей — свети к единице кратность нагрузки, которую представляет дешифратор для источника сигнала. Иначе каждый источник сигнала, как следует из рис. 5.1,б, будет нагружен весьма существенно — на $m/2$ входов элементов И.

Если дешифратор состоит из элементов И-НЕ, то на его выходах будут обрабаты-

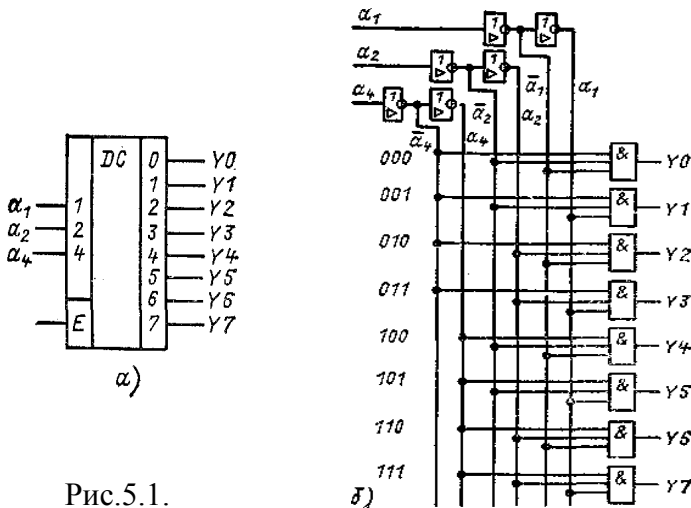


Рис.5.1.

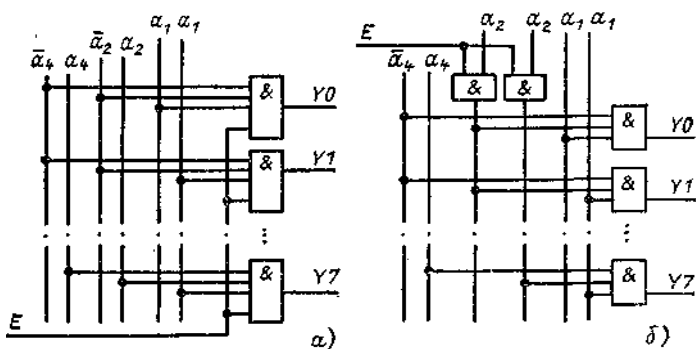


Рис.5.2.

ваться не сами функции Y_i , а их инверсии, т. е. активным уровнем выхода будет низкий. Ликвидировать инверсии на выходах можно или подключив инверторы, или построив дешифратор по двойственной схеме - на элементах ИЛИ-НЕ. Число входных инверторов и буферных усилителей при этом не изменится.

Дешифраторы часто имеют *разрешающий* (управляющий, стробирующий) вход E (от *enable* — давать возможность). При $E=1$ дешифратор работает как обычно, при $E=0$ на всех выходах устанавливаются неактивные уровни независимо от поступившего кода адреса. Вход E часто выполняют инверсным. Дешифратор, имеющий разрешающий вход, иногда называют *декодер-демультиплексор*.

На рис. 5.2, а показан вариант построения разрешающего входа, когда сигнал E воздействует непосредственно на все дешифрирующие элементы. Этот вариант требует увеличения на единицу числа входов дешифрирующих элементов, но не вносит дополнительной задержки. На рис. 5.2, б показан другой вариант, основанный на том, что, как видно из рис. 6.1,6, в дешифраторе не найдется ни одного дешифрирующего элемента, к которому любая переменная не была бы подключена или в своей прямой, или в инверсной форме. Поэтому если и в прямой, и в инверсный тракты любой входной переменной поставить элементы И и завести на них сигналы E , то при $E=0$ будут заперты абсолютно все конъюнкторы, подключенные к выходам.

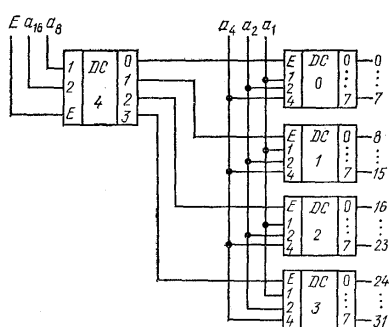


Рис.5.3.

На рис. 5.2, б показано воздействие сигнала E на среднюю переменную a_2 чтобы не создавалось ложного впечатления, что запирает дешифратор надо обязательно или по старшей, или по младшей переменной. Способ управления по одной из входных переменных экономичен по оборудованию, но увеличивает задержку дешифратора.

На рис. 5.3 показана группа из пяти дешифраторов, соединенных в два каскада. Вся группа работает как дешифратор 5—32. Два старших разряда адреса a_{16} и a_8 расшифровываются дешифратором 2—4 DC4,

который по входам E управляет четырьмя дешифраторами 3—8 второго каскада. Младшие разряды адреса a_4 , a_2 , a_1 поступают на все дешифраторы второго каскада, но открытым по входу E оказывается лишь один из них. Ему и будет принадлежать единственный из всех 32 возбужденный выход. Принцип используется при построении дешифраторов на много выходов из микросхем дешифраторов с меньшим числом выходов.

В рассмотренном примере 5-разрядный адрес был разбит на две группы в 2 и 3 разряда, и это определило структуру всей схемы. В общем случае многоразрядный адрес можно разбить на группы различными способами, и каждому способу будет соответствовать свой вариант схемы многокаскадного (не обязательно двухкаскадного) дешифратора. Варианты будут отличаться задержкой и аппаратными затратами, и можно ставить задачу выбора оптимальной в заданной серии элементов структуры.

В предельном случае при числе каскадов, равном разрядности адреса, получается *пирамидальный дешифратор*, имеющий максимально возможную задержку. Достоинством его является использование только двухвходовых элементов И, что определило его широкое распространение на заре цифровой техники, и с тех пор он, как сорняк, неистребим в литературе. В современных многовходовых логических базисах пирамидальная схема дешифратора не выдерживает конкуренции с другими структурами даже по оборудованию, не говоря уже о задержке.

На рис. 5.4 показан двухкаскадный дешифратор 4—16, второй каскад которого собран по схеме *прямоугольного или матричного дешифратора*. Разряды адреса разбиты на две группы, каждая из которых независимо от другой расшифровывается своим *дешифратором первого каскада DC1 и DC2*. При любой комбинации значений входных переменных оказываются выбранными одна строка и один столбец сетки, в узлах которой расположены элементы И второй ступени. В результате каждой входной набор возбуждает выход единственного соответствующего ему элемента И. Такую сетку из элементов И и называют *прямоугольным или матричным дешифратором*.

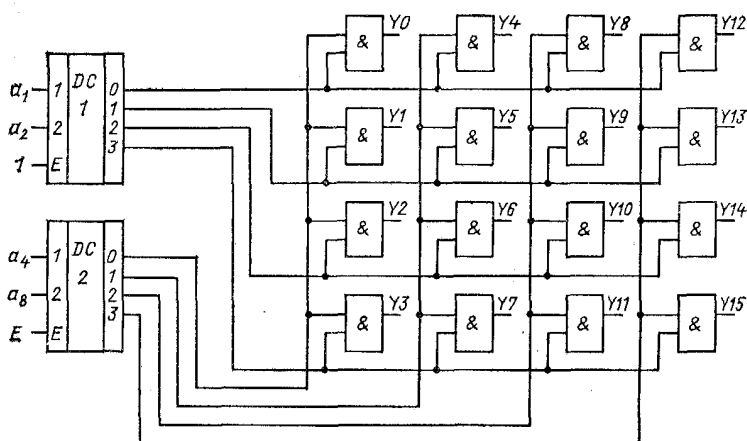


Рис.5.4.

При использовании во второй ступени элементов И-НЕ выходы дешифратора будут инверсными. Их можно сделать прямыми, построив сетку второго каскада по двойственному варианту, на элементах ИЛИ-НЕ; тогда инверсными должны быть выходы дешифраторов первого каскада. Делить разряды адреса между DC1 и DC2 нужно по возможности поровну: чем ближе прямоугольник второго каскада к квадрату, тем при том же числе выходных элементов И меньше сумма его строк и столбцов, т. е. меньше число выходов дешифраторов первого каскада. В качестве входа E всего двухкаскадного дешифратора удобно использовать разрешающий вход одного из дешифраторов первого каскада. При этом запираются или все строки, или все столбцы.

Целесообразно сравнить три рассмотренных типа дешифраторов по величине задержки и аппаратным затратам. Задержка минимальна для линейного дешифратора и в пределе может быть равна $(2 \dots 3)\tau_{\text{зад}}$ в зависимости от числа ступеней инверторов-усилителей. При включении линейных дешифраторов в несколько каскадов задержки всех каскадов складываются. Чем больше число выходов дешифратора, тем большая доля оборудования сосредоточена в элементах И самого последнего каскада. Число элементов предпоследнего каскада уже в несколько раз меньше, чем последнего, а предыдущих - тем более. Число элементов И последнего каскада дешифратора любого типа всегда равно числу его выходов, поэтому в первом приближении аппаратные затраты дешифраторов различных типов соотносятся как аппаратные затраты их элементов И последнего каскада. Они и определяют основную разницу: у линейного дешифратора число входов каждого элемента И последнего (и единственного) каскада равно n (n - число адресных входов), у каскадного оно зависит от способа разбиения на группы, однако оно всегда меньше n , по больше двух, у прямоугольного оно равно двум - минимально возможному числу. Поэтому при большом числе выходов (сотни и более) прямоугольный дешифратор - самый экономичный по оборудованию, чем и объясняется его широкое применение в БИС памяти. При уменьшении числа выходов до нескольких десятков первенство по экономичности переходит к каскадным дешифраторам, а при малом числе выходов самым экономичным (а к тому же и самым быстрым) оказывается линейный дешифратор.

Приведенные оценки справедливы лишь для дешифраторов, построенных непосредственно из логических элементов, например при разработке схемы матричной БИС. При проектировании же блоков из готовых микросхем, когда затраты оборудования оцениваются не числом элементов, а числом корпусов, даже большие дешифраторы экономичнее строить по каскадному принципу, набирая их из микросхем небольших дешифраторов на 8 или 16 выходов.

Шифраторы

Шифратор, или кодер (*encoder*), выполняет функцию, обратную дешифратору.

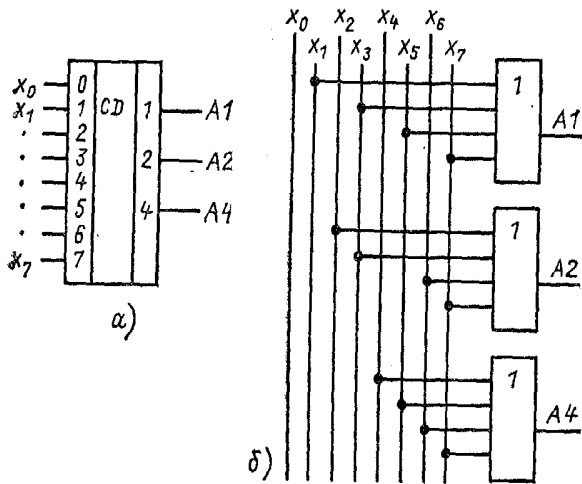


Рис.5.5.

Условное изображение шифратора на схемах показано на рис. 5.5.а. Классический шифратор имеет m - входов и n выходов, и при подаче сигнала на один из входов (обязательно на один, и не более) на выходе узла появляется двоичный код номера возбужденного выхода. Число входов и выходов такого шифратора связано соотношением $n=2^m$. Шифратор можно использовать, например, для отображения в виде двоичного кода номера нажатой кнопки или положения многопозиционного переключателя.

В данном случае существенно, что единица присутствует всегда только на одном из входов. Поэтому в создаваемой схеме не нужны конъюнкторы, выделяющие определенные комбинации нулей и единиц. Схема может начинаться прямо с элементов ИЛИ - по одному на каждый выход. Это соображение сразу разбивает искомую схему на n простых фрагментов. Ко входу элементов ИЛИ каждого выходного разряда должны быть подключены те входы шифратора, в двоичном представлении номера которых есть единица в данном разряде. Так, к ИЛИ младшего разряда формируемого выходного кода должны быть подключены все нечетные входы, поскольку у всех нечетных номеров, и только у них, в младшем разряде содержится единица. Функциональная схема шифратора 8 - 3 показана на рис. 6.5.

На схеме вход x_0 никуда не подключен, поскольку сигналу на этом входе соответствует выходной код «все нули». Недоумение может вызвать то, что схема совершенно не различает ситуаций «подан сигнал на вход X_0 » и «не подано ни одного входного сигнала вообще».

Причина в том, что при построении схемы учтено положение задания об обязательном присутствии сигнала на одном из входов, и оно использовано для минимизации аппаратных затрат.

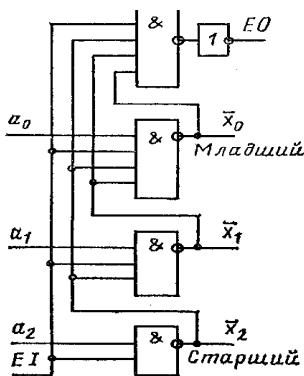


Рис.5.6.

Совместно с шифратором в состав кодирующих узлов может входить схема выделения старшей единицы. Эта схема преобразует m - разрядное слово следующим образом: все старшие нули и самая старшая единица входного кода пропускаются на выход без изменения; все разряды, более младшие, чем старшая единица, заменяются пулями.

Один из возможных вариантов схемы выделения старшей единицы показан на рис. 5.6.. На входы a_0 , a_1 , a_2 поступает преобразуемое слово (« a_0 - младший разряд, a_2 - старший»), на вход EI (от *enable in*) — входной сигнал разрешения. При $EI=1$ схема работает следующим образом. Любое число старших нулей порождает на выходах своих разрядов единицы и никак не влияет на работу элементов И-НЕ более младших разрядов. Любая самая старшая единица порождает на соответствующем выходе 0 (активный низкий уровень выхода) и запирает все более младшие элементы И-НЕ, устанавливая на их выходах неактивный высокий уровень. При этом низкий уровень появляется и на выходе EO (от *enable out*) - выходе разрешения. Если разрядность обрабатываемого слова превышает разрядность схемы, то слово разбивается на группы и выход EO более старшей группы подается на вход EI более младшей. При таком включении единица, поступившая на любой вход любой группы, запрет не только все более младшие разряды своей группы, но по цепочке $EO-EI$ — и все более младшие группы целиком. На выходах всей схемы останется только самая старшая единица входного слова, представленная активным низким уровнем.

Если к выходу схемы выделения старшей единицы подключить шифратор, то в сумме получится функциональный узел *приоритетного шифратора (priority encoder)*, формирующий в двоичном коде номер самой старшей единицы из всех, присутствующих во входном слове. С выходами схемы по рис. 6.6. хорошо стыкуются входы шифратора, если его выполнить по схеме, двойственной по отношению к показанной на рис. 6.5.6 : инверсным выходам одной схемы будут соответствовать инверсные входы другой, и весь приоритетный шифратор будет построен на технологичных элементах без лишних инверторов. Если во входном слове присутствует только одна единица, то приоритетный шифратор будет выполнять функцию обычного шифратора. Поэтому микросхемы обычных шифраторов не встречаются почти ни в одной серии, а приоритетные шифраторы выпускаются в составе многих серий.

Кроме кодирования состояний переключателей и номеров нажатых клавиш приоритетные шифраторы используются для определения номера устройства, подавшего сигнал запроса на обслуживание в микропроцессорных системах, входя в состав микросхем контроллеров прерываний

Преобразователи произвольных кодов

Если закон работы преобразователя не описывается каким-либо достаточно понятным правилом, как, например, работа декодера или шифратора, то единственной практически приемлемой формой задания преобразователя становится таблица истинности. Поскольку таблица воплощает в себе идею полного перебора вариантов, она способна задавать абсолютно любой закон.

Табл. 5.1.

A		B		
a_2	a_1	Z	J	K
0	0	1	0	0
0	1	0	0	1
1	0	0	1	1
1	1	0	0	0

Табл.5.2

A	B
0	4
1	1
2	3
3	0

Правильным правилом, как, например, работа декодера или шифратора, то единственной практически приемлемой формой задания преобразователя становится таблица истинности. Поскольку таблица воплощает в себе идею полного перебора вариантов, она способна задавать абсолютно любой закон.

Пусть табл. 5.1 описывает закон работы некоторого трехцветного светофора, управляемого двухразрядным двоичным кодом. Таблица 5.2. дает сокращенное описание того же закона, где двоичные коды заменены их десятичными (или восьмеричными) эквивалентами. На рис. 5.7.а показано условное изображение кодового преобразователя, заданного табл. 5.1. Изображение по рис. 6.7б допустимо использовать, когда коды имеют общепринятые названия.

К построению кодового преобразователя можно подойти с двух позиций.

При первом подходе преобразователь реализуется как *система булевых функций* группы аргументов. Таблица 5.1. может рассматриваться как таблица истинности системы B функций z, J, k.

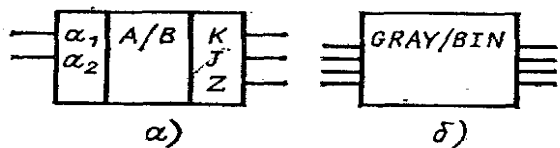


Рис.5.7.

Простейшим способом построения схемы, обрабатывающей систему функций с m выходами, является синтез обычными методами m независимых логических устройств.. Для трех выходов системы B записываются три выражения в дизъюнктивной форме: $z = a_2 \wedge a_1$, $j = a_2 \wedge \overline{a_1}$, $k = \overline{a_2} \wedge a_1 \vee a_2 \wedge \overline{a_1}$.

Легко видеть, что независимая схемная реализация этих выражений скорее всего будет неоптимальной, поскольку элементы, реализующие конъюнкцию $a_2 \wedge a_1$, оказываются дублированными в схемах j и k- Более экономичное решение обычно получается при подходе к системе функций с учетом ее взаимосвязанности.

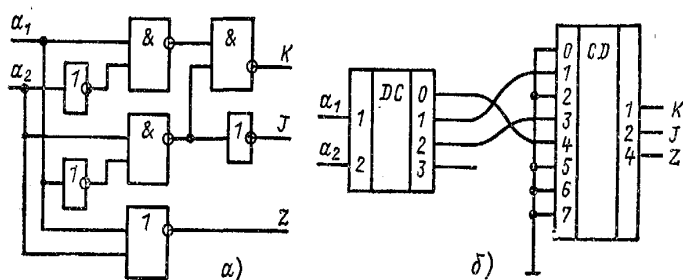
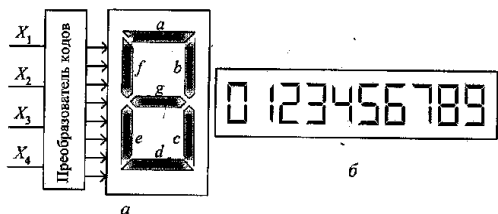


Рис.5.8.

Тогда часто удается выявить общие логические фрагменты, входящие в формулы нескольких выходов. Эти фрагменты достаточно реализовать схемно лишь один раз.

На рис. 5.8. а показана возможная реализация рассматриваемого кодового преобразователя в базе И-НЕ, ИЛИ-НЕ. В более сложных многовыходных схемах экономия от учета связности обычно оказывается существенно большей, чем в рассмотренном очень простом примере, однако каких-либо алгоритмов эффективного целенаправленного выявления общих частей, к сожалению, не существует. Как и синтез оптимальной логической схемы, это поисковая задача, только обычно еще более сложная из-за большего числа возможных вариантов. В литературе описаны некоторые приемы, упорядочивающие и облегчающие поиск экономичных вариантов реализации систем булевых функций.

Пример. Дешифратор двоичного кода в код семисегментного индикатора.

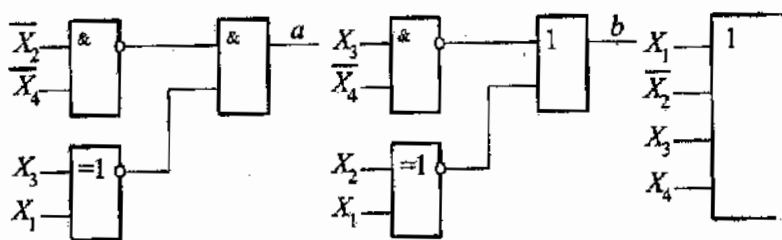


Визуальное отображение двоично-десятичных чисел часто выполняется с помощью семисегментных индикаторов на основе электролюминисцентных приборов, жидких кристаллов или светодиодных матриц. Количество семисегментных индикаторов определяется разрядностью чисел, отображаемых на световом табло, и составляет обычно шесть и более десятичных цифр.

Выводимый из компьютера десятичный код отображаемой цифры поступает вход двоично-десятичного преобразователя, выходы которого *a, b, c, ... g* подключаются к соответствующим сегментам индикатора (рис. 4.38, а).

X_1	X_2	X_3	X_4	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	1	1	0	1
0	1	0	1	1	1	1	0	0	1	1
0	1	1	0	1	1	1	1	0	0	1
0	1	1	1	1	1	1	1	0	0	1
1	0	0	0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	0	0	1	1	1
1	1	0	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$$\begin{aligned}
 a &= X_2 \vee X_4 \vee X_1 X_3 \vee \bar{X}_1 \bar{X}_3; \\
 b &= X_1 X_2 \vee \bar{X}_1 \bar{X}_2 \vee \bar{X}_3 \vee X_4; \\
 c &= X_1 \vee \bar{X}_2 \vee X_3 \vee X_4; \\
 d &= \bar{X}_1 X_2 \vee X_2 \bar{X}_3 \vee \bar{X}_1 \bar{X}_3 \vee X_1 \bar{X}_2 X_3 \vee X_4; \\
 e &= \bar{X}_1 X_2 \vee \bar{X}_1 \bar{X}_3; \\
 f &= \bar{X}_1 \bar{X}_2 \vee \bar{X}_1 X_3 \vee \bar{X}_2 X_3 \vee X_4; \\
 g &= \bar{X}_1 X_2 \vee X_2 \bar{X}_3 \vee \bar{X}_2 X_3 \vee X_4.
 \end{aligned}$$



При втором подходе к построению кодового преобразователя он трактуется как пара *декодер — кодер*. Схема того же преобразователя, построенного по второму способу, показана на рис. 5.8.б. Число входов дешифратора равно числу входов преобразователя, число выходов шифратора - числу выходов преобразователя. Соединения дешифратора и шифратора выполняются в соответствии с таблицей, и в этом случае табл. 5.2. может оказаться удобнее, чем табл. 5.1.. Часть выходов декодера и входов кодера может не использоваться. Если нескольким входным комбинациям соответствует одна и та же выходная, то соответствующие выходы декодера объединяют на элементе ИЛИ и выход последнего подают на нужный вход кодера. Эффективно стыкуются друг с другом декодер и кодер, построенные на элементах И-НЕ: первый имеет инверсный выход, а второй - инверсный вход. В качестве кодера можно использовать приоритетный шифратор.

При синтезе схемы на матричной БИС преобразователь, построенный по принципу реализации булевых функций, оказывается в среднем более экономичным по оборудованию, но менее быстродействующим, чем в варианте декодер - кодер, а при проектировании из готовых микросхем более выгодным и по числу корпусов, и по быстродействию обычно оказывается структура декодер - кодер. Однако потребляемая мощность в этом случае может быть больше, чем у схемы из отдельных логических элементов. За-

траты времени инженера на логическое проектирование по схеме декодер - кодер неизмеримо меньше, чем затраты на проектирование преобразователя из логических элементов.

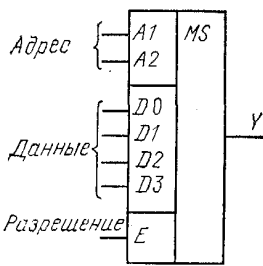
Мультиплекоры и демультиплексоры

Термином “мультиплексирование” называют процесс передачи данных от нескольких источников по общему каналу, и любое устройство, осуществляющее на передающей стороне операцию сведения данных в один канал, принято называть мультиплексором. Это название исторически закрепилось за схемой, способной осуществлять временное мультиплексирование сигналов, передавая их в линию друг за другом в темпе смены кодов на своих адресных входах. Но эта же схема может выполнять и еще одну распространенную операцию — *выбор, селекцию* (от *select* — выбирать) данных из определенного, указанного адресным кодом источника. Любое устройство, выполняющее операцию селекции, называют селектором, и поэтому указанная схема имеет второе название - *селектор*. Кроме того, поскольку схема выполняет коммутацию сигналов, ее еще называют *коммутатором*. Поэтому на функциональных схемах ее обозначают и *MUX* и *MS*.

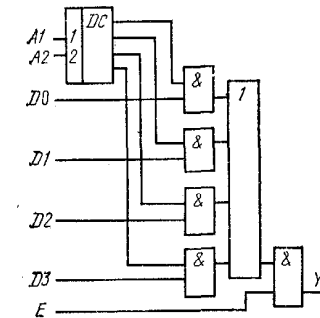
На приемной стороне мультиплексированной магистрали требуется выполнить обратную операцию — демультиплексирование, т. е. распределение порций данных, поступающих в последовательные моменты времени, по своим приемникам. Эту операцию выполняет *демультиплексор*.

Мультиплексор (multiplexor) — это функциональный узел, осуществляющий под-

ключение (*коммутацию*) одного из нескольких *входов данных* к выходу. Номер выбранного входа соответствует коду, поданному на *адресные входы* мультиплексора. Вход *E* — разрешающий: при $E=1$ мультиплексор работает как обычно, при $E=0$ выход узла находится в неактивном состоянии, мультиплексор заперт.



Условное изображение мультиплексора



Возможный вариант структурной схемы мультиплексора

Работу M можно представить в виде многопозиционного ключа. Управляющий код по адресным входам определяет номер входной линии, информационный сигнал с которой поступает на выход. В отличие от него M допускает только одностороннюю передачу данных.

Работа M описывается соотношением

$$F = x_0 \wedge \overline{a_{n-1}} \wedge \overline{a_{n-2}} \wedge \dots \wedge \overline{a_1} \wedge \overline{a_0} \vee x_1 \wedge \overline{a_{n-1}} \wedge \overline{a_{n-2}} \wedge \dots \wedge \overline{a_1} \wedge a_0 \vee \dots \vee x_m \wedge \overline{a_{n-1}} \wedge \overline{a_{n-2}} \wedge \dots \wedge \overline{a_1} \wedge a_0$$

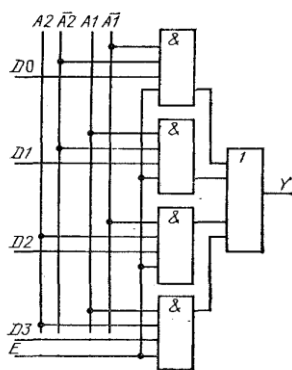
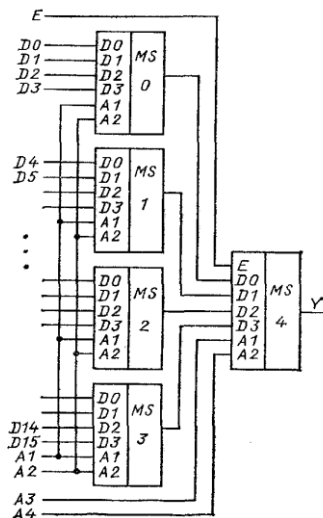


Схема мультиплексора в базисе НЕ, И, ИЛИ



Каскадное соединение мультиплексоров

называемым мультиплексной формулой ($m = 2^n$).

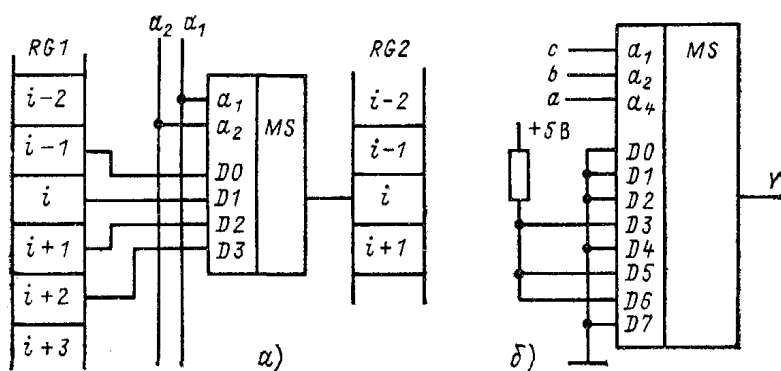
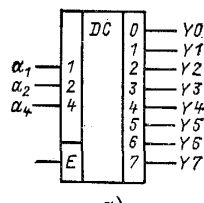
С помощью формул де-Моргана ее легко перевести в базис И-НЕ. Это и будет окончательная схема, по которой часто строят реальные мультиплексоры.

При проектировании цифровых устройств (и вообще решении инженерных задач) часто используется отыскание окончательного решения путем оптимизации некоторого предварительного. При этом оно может быть заведомо неоптимальным, но обязательно должно быть доведенным до конца. Часто это оказывается намного результативнее, чем пытаться найти сразу оптимальное.

В роли демультиплектора успешно выступает декодер, если к его разрешающему входу E подключить мультиплексированную магистраль данных, а на адресные входы подавать друг за другом коды адресов приемников. Поэтому декодер, имеющий разрешающий вход E , иногда называют не просто декодер, а декодер-демультиплексор.

Применение мультиплексов не ограничивается операциями мультиплексирования и селекции.

- Параллельный сдвигатель (одно из назначений – реализация операций умножения или деления на 2, эквивалентных в двоичной арифметике операциям сдвига)



На рис. а показан один, 1-й разряд схемы параллельного сдвигателя.

В полной схеме сдвигателя ко входу каждого разряда регистра $RG2$ подключено по такому же мультиплексу, входы данных которого в свою очередь подключены к выходам нескольких разрядов регистра $RG1$. На адресные входы мультиплексов всех разрядов подается один и тот же код. В результате в зависимости от значения

адресного кода в 1-й разряд $RG2$ будет переписываться содержимое различных разрядов $RG1$.

Возможные применения мультиплексов:

а – один разряд комбинационного сдвигателя; б – реализация произвольной функции, заданной таблицей истинности, в данном случае – табл. 3.2

- Универсальный логический элемент (составляет функционально полный набор логических функций)

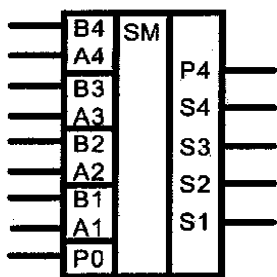
№	a	b	c	Y	№	a	b	c	Y
0	0	0	0	0	4	1	0	0	0
1	0	0	1	0	5	1	0	1	1
2	0	1	0	0	6	1	1	0	1
3	0	1	1	1	7	1	1	1	0

Мультиплексор можно использовать в качестве универсального логического элемента для реализации любой функции от числа аргументов, равного числу адресных входов мультиплексо-ра. Мультиплексор, изображенный на Рис. 7.1.б, реализует функцию, заданную табл. Для этого входы данных мультиплексо-ра подклю-чены к источникам 1 и 0 в такой последова-тельности; которая полностью копирует последовательность единиц и нулей таблицы истинности.. При этом не требуется ни записи СДНФ, ни ее минимизации! Кстати, эта функци, не минимизи-руется (в чем полезно убедиться лично), поэтому для своей реализации требует четырех элемен-тов ЗИ-НЕ и трех инверторов, что в сумме даст почти два корпуса и 3т задержки. Неудиви-тельно, что способ реализации функций трех или четырех аргументов с помощью микросхемы мультиплексо-ра весьма популярен у разработчиков. Следует помнить, что этот способ может дать экономию лишь при использовании микросхем. При разработке схем для кристаллов матричных и других БИС объем оборудования определяется числом базовых логических элементов, поэтому такой способ будет крайне расточительным.;

Сумматоры

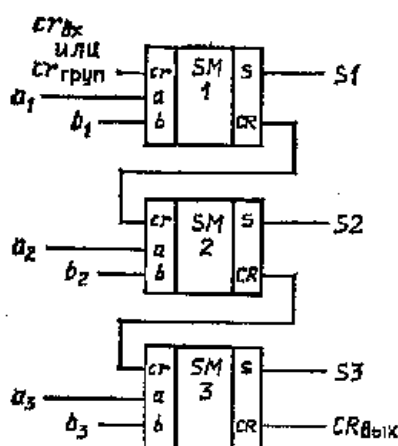
Сумматоры выполняют арифметическую операцию сложения двух чисел.

Они имеют как самостоятельное значение, так и являются составной частью арифметико-логического устройства (АЛУ). При организации различных вычислительных процессов суммированию отводится главная роль, оно является основной операцией.



Например, вычитание — это суммирование с использованием дополнительного либо обратного кода, умножение сводится к сдвигу и сложению (суммированию) двоичных чисел. Следует отметить, что сумматоры являются логическими устройствами, функционируют по законам алгебры логики, но выполняют операцию арифметического, а не логического сложения. В соответствии с определением, сумматор суммирует два числа. Выходной сигнал зависит только от двух входных сигналов, действующих на входе в текущий момент. Следовательно, сумматор является комбинационным устройством.

Однако, сумматоры могут быть реализованы и другими способами, например, содержащим в своем составе элементы памяти.



Простейший способ построения многоразрядного сумматора — последовательное включение соответствующего числа одноразрядных сумматоров, складывающих одноименные разряды слагаемых. Выход переноса CR каждого разряда подключен ко входу переноса cr соседнего старшего разряда. Выходной сигнал переноса старшего разряда является выходом переполнения всего многоразрядного сумматора. В результате задача построения сумматора может быть сведена к построению схемы одноразрядного сумматора.

Таблица истинности одноразрядного сумматора имеет вид

Входы			Выходы	
слагаемые		перенос	сумма	перенос
a_i	b_i	p_i	S_i	P_{i+1}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Это позволяет выразить структурные формулы в СДНФ

$$S_i = \bar{a}_i \bar{b}_i \bar{p}_i \vee a_i \bar{b}_i \bar{p}_i \vee \bar{a}_i b_i p_i \vee a_i b_i p_i,$$

$$P_{i+1} = a_i b_i \bar{p}_i \vee \bar{a}_i b_i p_i \vee a_i \bar{b}_i p_i \vee a_i b_i p_i.$$

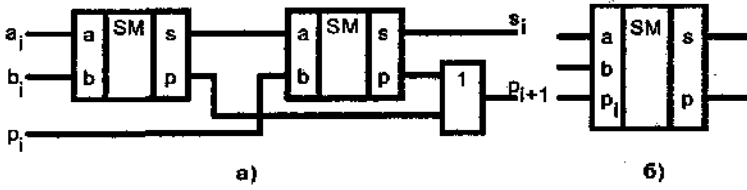
Записанные функции очень компактно выражаются с помощью операции «ИСКЛЮЧАЮЩЕЕ ИЛИ» («СУММАТОР ПО МОДУЛЮ 2», «НЕРАВНОЗНАЧНОСТЬ»)

$$S_i = p_i \overline{(a_i \oplus b_i)} \vee \bar{p}_i (a_i \oplus b_i) = (a_i \oplus b_i) \oplus p_i;$$

$$P_{i+1} = a_i b_i \bar{p}_i \vee p_i (a_i \vee b_i) = a_i b_i \vee p_i (a_i \oplus b_i).$$

Устройство, реализующее одновременно эти структурные формулы, называется **полным** одноразрядным сумматором. С точки зрения технической реализации последние выражения оказывается целесообразным построить его на базе

более простых устройств - **полусумматоров** (частичных сумматоров, неполных сумматоров)



Неполный сумматор является устройством, реализующим логиче-

Слагаемые		Результат	
a	b	Сумма S	Перенос P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

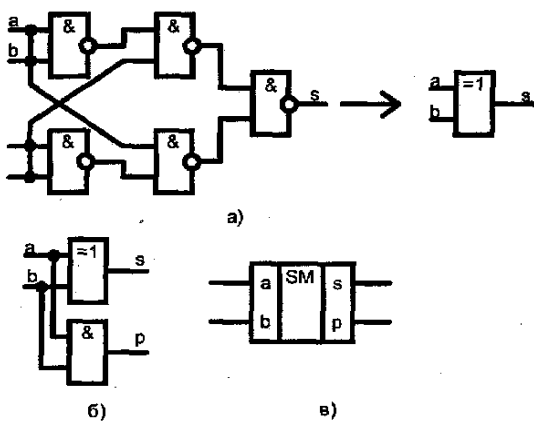
ские функции двух переменных, таблица истинности которого имеет вид

Это позволяет записать структурные формулы в виде

$$S = \bar{a} \wedge b \vee a \wedge \bar{b} = a \oplus b$$

$$p = a \wedge b$$

Используя метод декомпозиции, неполный сумматор может быть реализован в виде совокупности двух автономных логических устройств.



Логическое устройство, формирующее сигнал переноса p , реализуется элементом «И».

Операция «ИСКЛЮЧАЮЩЕЕ ИЛИ», также относится к элементарным логическим функциям. В составе развитых серий интегральных схем всегда имеются готовые ИС, однако, в силу ограниченного использования этой логической операции часто удобнее оказывается ее реализовать. Пример реализации этой функции в базе «И-НЕ» приведен на рис.а.

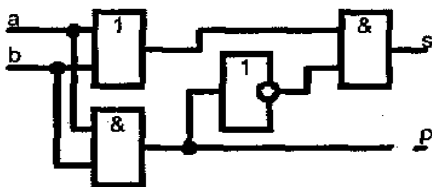
Как уже отмечалось, метод декомпозиции подходит к построению цифрового устройства к совокупности автономных логических

устройств, что предполагает избыточность аппаратных затрат. Применительно к неполному сумматору можно учесть, что сигнал переноса p появляется как промежуточный результат при одном из способов представления структурной формулы для S

$$S = \bar{A} \wedge B \vee A \wedge \bar{B} = \bar{A} \wedge B \vee A \wedge \bar{B} \vee A \wedge \bar{A} \vee B \wedge \bar{B} = A \wedge (\bar{A} \vee \bar{B}) \vee B \wedge (\bar{A} \vee \bar{B}) =$$

$$= (\bar{A} \vee \bar{B})(A \wedge B) = (A \vee B) \wedge \bar{A} \wedge \bar{B}$$

В результате схема может быть реализована в виде (см. рис.)



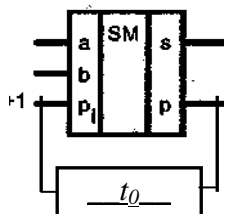
Частичные сумматоры находят самостоятельное применение как полные сумматоры в младших разрядах многоразрядных сумматоров, где сигнал переполнения от предыдущих разрядов заведомо отсутствует. Кроме того, самостоятельное применение находят и элементы «ИСКЛЮЧАЮЩЕЕ ИЛИ».

Например, они широко используются в качестве «управляемых инверторов». Нетрудно видеть, что, рассматривая один из входов элементов в качестве управляющего, выбором значения сигнала на нем, можно получать на выходе элемента прямой или инверсный вид сигнала на другом входе.

Рассмотренный способ реализации многоразрядных сумматоров является параллельным сумматором, не накладывающим ограничений на его разрядность. Быстродействие же подобного сумматора, оцениваемое временем до получения окончательного результата, определяется, очевидно, скоростью передачи сигнала переноса в

старшие разряды. Действительно, выходные сигналы каждого промежуточного разряда оказываются достоверными только после поступления сигнала переноса от младших разрядов. В результате инерционность подобного сумматора с **последовательным переносом** характеризуется величиной $\tau_{зад} = N \cdot \tau_0$, где N – число разрядов сумматора. Существенное повышение быстродействия сумматоров обеспечивается в схемах с **параллельным переносом**, обеспечиваемым, однако, за счет существенного усложнения аппаратуры. В этом случае вводится дополнительное комбинационное устройство, вырабатывающее сигналы переноса для старших разрядов помимо рассмотренных одnorазрядных сумматоров. Это, однако, возможно лишь при учете всех разрядов входных сигналов одновременно, так что это устройство является функцией $2N$ – аргументов.

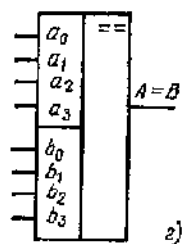
Наряду с **параллельным** сумматором, реализующим операцию арифметического сложения



двух чисел, заданных в параллельном коде, часто используется **последовательный** сумматор для чисел, заданных в последовательном коде. При этом схема устройства оказывается достаточно простой. Схема временной задержки на величину t_0 обеспечивает поступление сигнала переполнения при суммировании следующего старшего разряда.

Цифровые компараторы

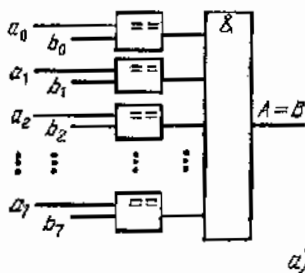
Цифровым компаратором называют функциональный узел сравнения чисел. Простейшие компараторы формируют на выходе однобитовый сигнал совпадения сравниваемых чисел $F_{A=B}$ (лог.1 – совпадение чисел, лог.0 – в противном случае).



Традиционное решение с помощью построения таблицы истинности в этом случае оказывается очень громоздким – размер этой таблицы быстро растет с увеличением разрядности. Поэтому часто используются *эвристические* методы решения, основанные на использовании какой-то дополнительной информации.

Очевидно, что равенство многоразрядных чисел возможно только в том случае, если выполняется поразрядное совпадение этих чисел. Поскольку обнаружение совпадения значений отдельных разрядов реализуется схемой «НЕРАВНОЗНОСТЬ» («ИСКЛЮЧАЮЩЕЕ ИЛИ - НЕ»), схема простейшего компаратора может быть реализована в виде, приведенном на рис.

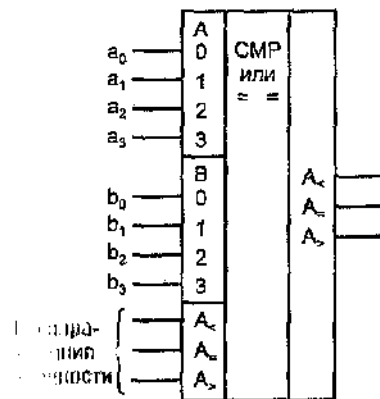
Очевидно, что равенство многоразрядных чисел возможно только в том случае, если выполняется поразрядное совпадение этих чисел.



В развитых сериях цифровых микросхем обычно имеются компараторы, решающие более общую задачу – вырабатывают дополнительные сигналы сравнения чисел $F_{A>B}$, $F_{A<B}$, причем с возможностью наращивания разрядности сравниваемых чисел.

В этом случае опять учитывается, что результат сравнения значений отдельного разряда имеет смысл только в том случае, если совпадают значения более старших разрядов.

В развитых сериях цифровых микросхем обычно имеются компараторы, решающие более общую задачу – вырабатывают дополнительные сигналы сравнения чисел $F_{A>B}$, $F_{A<B}$, причем с возможностью наращивания разрядности сравниваемых чисел.



Проектирование узлов комбинационного типа произвольной логики

Построение логической модели

Прежде всего, задается характер функционирования КЦ. Это может быть сделано различными способами, но задача сводится к ее представлению в виде таблицы функционирования (таблицами истинности), задающими значение искомых функций на всех наборах аргументов. От таблицы легко перейти к СДНФ искомых функций (СДНФ— совершенная дизъюнктивная нормальная форма, т. е. дизъюнкция конъюнктивных членов одинаковой размерности). Для этого составляют логическую сумму тех наборов аргументов, на которых функции принимает единичное значение.

Дальнейшие действия зависят от выбранной для аппаратной реализации элементной базы, в качестве которой могут быть использованы:

1. Постоянные запоминающие устройства (ПЗУ)
2. Программируемые логические матрицы (ПЛМ)
3. Логические блоки на мультиплексорах
4. Функционально полные наборы логических элементов.

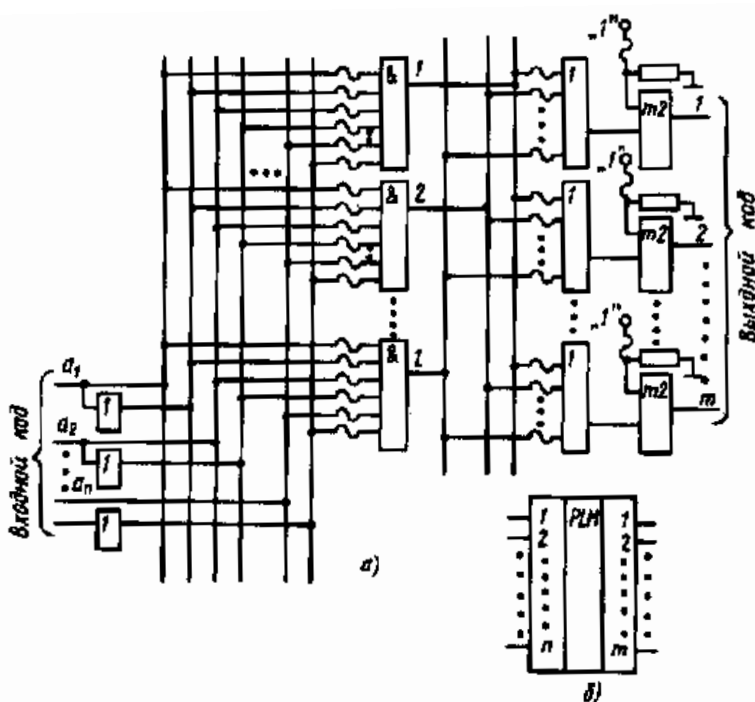
Современная элементная база предлагает большой выбор **ПЗУ**, которые различаются как способом записи информации (изготовителем или пользователем, однократно или с возможностью многократной перезаписи, с возможностью программирования непосредственно в схеме, ..). При соответствующем выборе адресного пространства и



б

организации памяти ПЗУ процесс проектирования как логических, так и цифровых устройств не требует дальнейших операций – значения функции заносятся в память ПЗУ и работа комбинационного устройства в этом случае сводится лишь к извлечению известного

результата для конкретного набора входных сигналов. Более того, если искомая функция задана в какой-то сокращенной форме, то ее необходимо привести к СДНФ. Подобный метод, очевидно, является универсальным, все более привлекательным по мере развития элементной базы.



Более удобным методом реализации комбинационных устройств в настоящее время считается использование ПЛМ, характеризующихся меньшей схемной избыточностью. Действительно, ПЗУ допускает все наборы входных сигналов возможными и отводит отдельные ячейки памяти для хранения результатов для каждого случая. ПЛМ же допускает ограниченное число наборов входных сигналов, для которых устройство должно выдать лог.1.

Программируемые логические матрицы (ПЛМ) выпускаются в микросхемном исполнении. Такая матрица

содержит k конъюнкторов, входы каждого из которых соединены с линиями входных сигналов и их инверсий, и m дизъюнкторов, входы каждого из них соединены с выходами всех конъюнкторов. Выходы дизъюнкторов выведены наружу через элементы «исключающее ИЛИ», позволяющие пропускать на выходы сигналы в прямой или инверсной форме.

Пережигая соответствующие переключки (на рис. *a* они показаны волнистыми линиями) и оставляя необходимые соединения - программируя матрицу, можно организовать конъюнкции любых комбинаций входных сигналов и дизъюнкцию любых наборов полученных конъюнкций.

Через переключку на вход «исключающего ИЛИ» поступает потенциал логической 1, что приводит к инверсии функции, сформированной на выходе дизъюнктора. При разрушении этой переключки на входе «исключающего ИЛИ» оказывается логический 0 и сформированная функция при прохождении через этот элемент не инвертируется.

Будучи запрограммированной, ПЛМ устанавливает постоянное соответствие между выходным и входным кодами, т. е. может использоваться как преобразователь кодов или, что равносильно, как формирователь логических функций на выходах по их аргументам на входах. ПЛМ оказывается способной формировать m выходных функций от n входных аргументов с числом членов в функции, равным l , при гибкой связи между этими числами. ПЛМ допускает получение нескольких разных выходных функций при одном входном коде и одной и той же функции при разных входных кодах.

Вместе с тем ПЛМ не дает возможности формировать функции с числом членов более l , т. е. более числа конъюнкторов. Условное обозначение ПЛМ показано на рис.б.

Часто даже логическая мощность ПЛМ оказывается избыточной, например, при реализации функций, не содержащих больших общих логических слагаемых. Упрощенная схема ПЛМ, известная как ПМЛ (программируемая матричная логика), характеризуется жестким распределением выходов элементов И (первая матрица) со входами элементов ИЛИ (вторая матрица). Это оказывается более удобным при реализации несложных устройств.

При построении устройств на ПЛМ может оказаться необходимым минимизировать структурную формулу до числа слагаемых, определяемых числом входов матрицы ИЛИ выбранной ИС. Дальнейшая минимизация может позволить обойтись ИС меньшей степени интеграции.

Случай применения мультиплексоров близок к случаю использования ПЗУ с той лишь разницей, что значения функции задаются схемным способом, т.е. непосредственным подключением входов мультиплексора к точкам схемы, имеющим потенциал лог.0 или лог.1. Широкий выбор мультиплексоров, возможность наращивания их разрядности делают этот способ принципиально простым.

Реализация цифровых устройств на логических элементах является самым изученным способом и сводится к отысканию оптимальной формы структурной формулы. При этом возможно необозримое множество вариантов, причем наиболее выигрышные не всегда очевидны. Поиск таких вариантов обычно происходит без теоретических подсказок и на основе эвристических рассуждений.

Уточнение модели с учетом конечного быстрогодействия элементов

Выходные величины устройств комбинационной логики зависят только от текущего значения входных величин (аргументов). Предыстория значения не имеет. После завершения переходных процессов на их выходах устанавливаются выходные величины, на которые характер переходных процессов влияния не оказывает. С этой точки зрения переходные процессы в подобных схемах не опасны. Но обычно в цифровом устройстве узлы комбинационного типа функционируют совместно с узлами последовательностного типа, что кардинально меняет ситуацию. Во время переходных процессов на выходах комбинационных схем появляются временные сигналы, не предусмотренные описанием их работы и называемые *рисками*. Со временем они исчезают, и на выходе устанавливается значение, предусмотренное логической формулой, описывающей работу цепи. Однако риски могут быть восприняты элементами памяти, необратимое изменение

состояния которых может радикально изменить работу цифрового устройства в целом, несмотря на исчезновение сигналов рисков.

Различают статические и динамические риски. Статические риски — это кратковременные изменения сигнала, который должен был оставаться пониженным (единичным или нулевым, соответственно чему говорят о 1-риске или 0-риске). Если согласно логике работы узла комбинационного типа состояние выхода должно измениться, но вместо однократного перехода происходят многократные, то имеет место динамический риск. При динамических рисках первый и последний переходы всегда совпадают с алгоритмически-

предусмотренными логикой работы схемы. Статический риск такого свойства не имеет и считается более неблагоприятным.

Простейший пример (см. рис. а) соответствует выработке функции "константа 1" по формуле $F = \overline{x \wedge x} = 1$. В статике при любом значении x на одном из входов элемента И-НЕ имеется логический нуль, обеспечивающий единичное значение выхода. В переходных процессах возможен статический 1-риск.

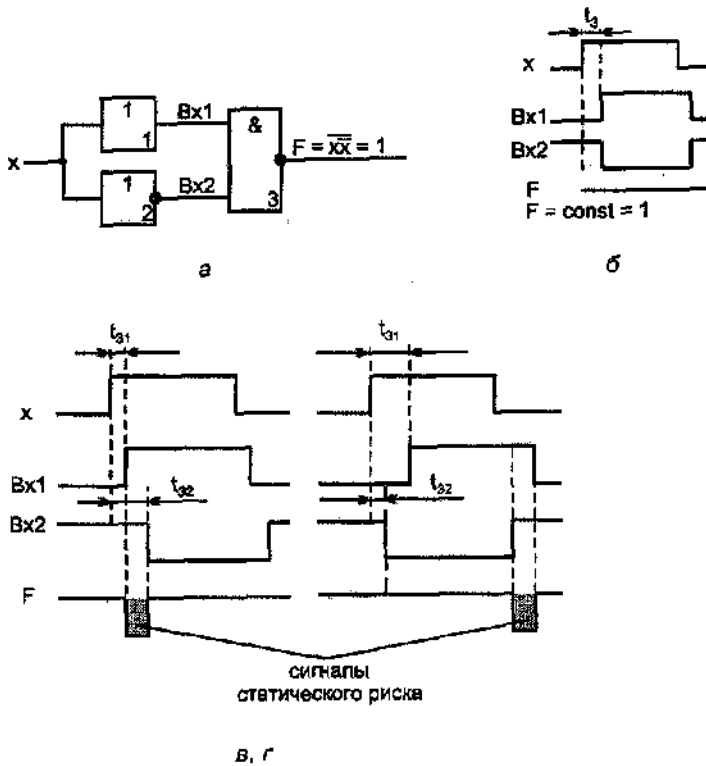
Не учитывая задержку элемента 3, которая здесь не играет роли, рассмотрим временные диаграммы переходных процессов для случаев равенства задержек элементов 1 и 2 ($t_{зад1} = t_{зад2}$) (рис.б), а также их неравенства $t_{зад1} < t_{зад2}$ и $t_{зад1} > t_{зад2}$ показанные на рис. в,г.

Видно, что при различных задержках элементов возникает статический риск после положительного или отрицательного перепада входного сигнала в зависимости от того, задержка какой элемента больше.

В логических схемах встречаются подобные участки, где сигнал разветвляется, каждый из получившихся сигналов распространяется по собственной цепи, а затем они снова сходятся на входах одного элемента. Но при прохождении этих цепей указанные сигналы оказываются задержанными из-за инерционности элементов соответствующих цепей. Причем, величина этой задержки является случайной, так что, например, замена одного из элементов этой цепи может вызвать появление сигнала риска, способного нарушить работу устройства. Описанное явление называют **гонками**.

Одной их разновидностей подобных проблем, связанных с временными характеристиками цифровых устройств, являются гонки по входу.

Гонки по входу возникают, когда ветвящийся сигнал поступает на элементы, имеющие разброс по уровню срабатывания (рис.а), а фронт этого сигнала излишне пологий (рис.б). Если длительность фронта входного сигнала заметно больше времени срабатывания элементов, то где-то в середине фронта будет существовать отрезок времени, когда с точки зрения одного элемента входной сигнал уже равен 1, а с точки зрения другого - еще равен 0. Элементы будут реагировать на один и тот же сигнал как на два различных, а такая ситуация при проектировании схемы ее алгоритмом не предусматривается. В результате схема в течение этого времени может выработать ложные сигналы. Это явление и называют «гонки по входу». Гонки по



срабатывания элементов, то где-то в середине фронта будет существовать отрезок времени, когда с точки зрения одного элемента входной сигнал уже равен 1, а с точки зрения другого - еще равен 0. Элементы будут реагировать на один и тот же сигнал как на два различных, а такая ситуация при проектировании схемы ее алгоритмом не предусматривается. В результате схема в течение этого времени может выработать ложные сигналы. Это явление и называют «гонки по входу». Гонки по

входу не наблюдаются, если логическая схема собрана на элементах одной серии микросхем. Потенциально опасны с этой точки зрения схемы, собранные из элементов различных серий, имеющих одинаковый уровень сигналов, но существенно различные времена задержек и фронтов. Это обычно учитывается наложением требований к минимальной длительности фронтов сигналов, поступающих на вход логических элементов, т.е. к источникам этих сигналов.

Наиболее распространены два способа борьбы с этими гонками:

1. Синтез схем, свободных от гонок. В этом случае необходим анализ процессов в схеме и приводит обычно к введению избыточных элементов для исключения рисков. По этой причине используется только при построении достаточно простых устройств.

2. Использование тактирования работы устройств. В этом случае восприятие информации разрешается только по приходу служебных сигналов, специально вводимых для синхронизации работы устройства. В результате восприятие информации во время переходных процессов оказывается невозможным. Частота следования тактовых импульсов, таким образом, определяет быстродействие устройства и устанавливается с учетом наиболее инерционных процессов в работе устройств.

Контрольные вопросы

1. Назначение функциональных узлов в электронных устройствах.
2. Отличительные особенности узлов комбинационного типа.
3. Назначение шифраторов.
4. Различия между дешифраторами и мультиплексорами.
5. Достоинства матричного дешифратора.
6. Недостатки пирамидального дешифратора.
7. Отличие арифметического сумматора от устройства логического сложения чисел.
8. Принципы построения преобразователей кодов.
9. Пути повышения быстродействия сумматоров.
10. Различия между полным и частичным сумматором.
11. Возможность варианта реализации операции арифметического суммирования на базе мультиплексора.
12. Вариант реализации функции дешифратора на базе мультиплексора.
13. Назначение цифровых компараторов.

Последовательностные устройства

Триггеры

Общая характеристика триггеров

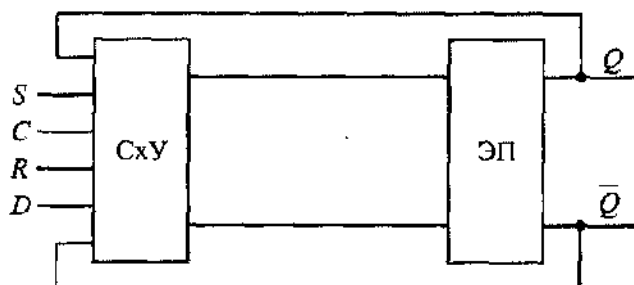
Особенностью последовательностных логических устройств является их зависимость не только от действующих в настоящий момент на входе логических сигналов, но и от тех значений переменных, которые действовали на входах в предыдущие моменты времени. Очевидно, что для выполнения этого условия значения предыдущих переменных должны быть запомнены логическим устройством.

Элементарным устройством памяти является триггер - простейший цифровой автомат с памятью, способный хранить 1 бит (*binary digit* — двоичный разряд) информации, который является неотъемлемой частью любого последовательностного устройства.

Триггером называют устройство, имеющее два устойчивых состояния и способное под действием внешних сигналов переключаться из одного состояния в другое. При этом напряжение на его выходе изменяется скачкообразно. Способность поддерживать устойчивое состояние на выходе без изменения при отсутствии входных сигналов сколь угодно

длительный интервал времени и обуславливает применение триггеров в качестве элементов памяти.

В общем случае триггер содержит собственно элемент памяти и входную



комбинационную схему, преобразующую входные сигналы триггера в сигналы, требуемые для управления элементом памяти. В основе любого триггера находится регенеративное кольцо из двух инверторов, охваченных глубокой положительной обратной связью. Поэтому переход из одного состояния в другое происходит лавинообразно за очень короткое время.

Триггер имеет два выхода: прямой Q и инверсный \bar{Q} . Состояние триггера определяют по значению сигнала на прямом выходе Q .

Изменение состояния триггера (его переключение) обеспечивается внешними сигналами и сигналами обратной связи на выходе триггера, поступающие на входы СхУ. Обычно внешние сигналы, как и входы триггера, обозначают латинскими буквами R, S, T, C, V др.. В простейших схемах триггеров отдельная СхУ может отсутствовать. Поскольку функциональные свойства триггеров определяются их СхУ, то названия основных входов переносятся на всю схему триггера.

Классификация триггеров

Триггеры классифицируют последующим признакам:

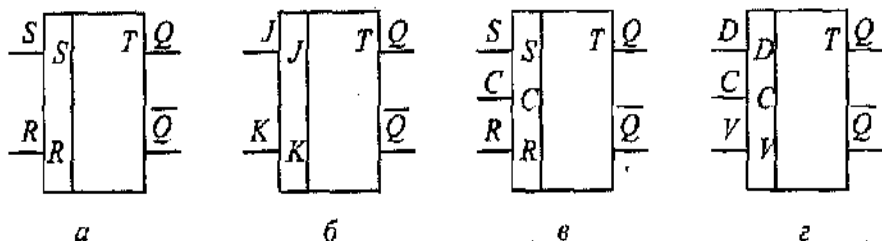
- логике функционирования (RS, JK, D, T и др.);
- способу записи информации (асинхронные и синхронные);
- моменту реакции на тактовый сигнал (статические, динамические);
- количеству ступеней (одно или двухступенчатые триггеры);
- составу логических элементов (триггеры на элементах НЕ И, НЕ ИЛИ, НЕ И ИЛИ и др.).

В соответствии с логикой функционирования различают следующие триггеры:

- с отдельной установкой состояний "0" и "1" (RS-триггеры);
- с одним информационным входом (D-триггеры);
- со счетным входом (T-триггеры);
- универсальные с отдельной установкой состояний "0" и "1" (JK-триггеры);
- комбинированные (RST-, RSJK-триггеры);
- со сложной входной логикой.

Входы триггеров разделяются на информационные (R, S, T и др.) и управляющие (C, V). Информационные (логические) входы предназначены для приема сигналов запоминаемой информации. Названия входных сигналов отождествляют с названиями входов триггера. Управляющие входы служат для управления записью информации. В триггерах может быть два вида управляющих сигналов; синхронизирующий (тактовый) сигнал C , поступающий на C -вход (тактовый вход) и разрешающий сигнал V , поступающий на V вход

По способу записи (приема) информации различают *асинхронные* и *синхронные* (тактируемые) триггеры. Триггеры, не имеющие C -входа (рис.а,б), называются асинхронными.

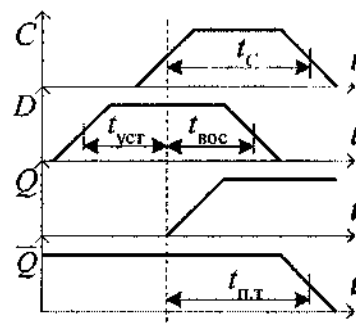
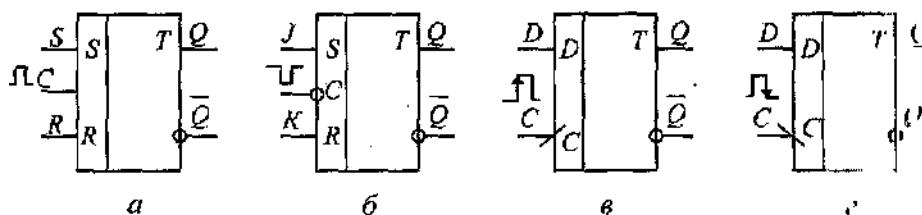


В них запись информации производится в любой момент времени при поступлении сигналов на информационные входы. Триггеры, имеющие C -вход (рис.в,г), называются синхронными.

В синхронном триггер запись информации возможна при совпадении сигналов на информационном входе и входе синхронизации. Это позволяет обеспечить более высокую помехоустойчивость работы устройств на основе триггеров.

На V -входы триггера поступают сигналы, которые разрешают ($V = 1$) или запрещают ($V = 0$) запись информации. В синхронных триггерах с V -входом запись информации возможна при совпадении сигналов на информационном, C - и V - входах.

По способу управления записью (моменту реакции на тактовый сигнал) различают синхронные триггеры со статическим (по уровню) и динамическим (по фронту) управлением.



В асинхронных триггерах запись нуля и единицы возможна в любой момент времени, при этом входной информационный сигнал одновременно является и управляющим.

В синхронных триггерах с управлением, по уровню запись информации возможна только в течение длительности тактового сигнала. При этом тактовые сигналы могут быть прямыми (изменяться от 0 к 1) или инверсными (изменяться от 1 к 0) (рис. а,б).

При управлении фронтами разрешение на запись информации дается только в момент перепада тактового сигнала от нуля к единице (прямой динамический вход) или от единицы к нулю (инверсный динамический вход). В другие моменты времени триггер не реагирует на входные информационные сигналы независимо от уровня тактового импульса.

Динамические параметры триггеров

Важным способом описания функционирования триггеров являются временные диаграммы, отражающие не только логическое функционирование схемы, но и ее поведение во времени. Другими способами это поведение не отображается, поэтому при уточнении логической модели с учетом временных задержек этот способ оказывается незаменимым.

Для характеристики динамических параметров триггеров используется ряд временных параметров, измеряемых на уровне половины амплитуды сигналов:

- минимальная длительность импульса t_c на тактовом входе;
- минимальное время предварительной установки сигнала на информационном входе $t_{уст}$;
- время выдержки (фиксации) $t_{вос}$ — минимальное время между нарастанием синхросигнала C и изменением информационного сигнала D ; для асинхронных триггеров $t_{вос}$ — просто длительность входного сигнала;
- время переключения триггера $t_{..T}$ — временной интервал между фронтом входного переключающего сигнала и фронтом сигнала на выходе, который принимает новое состояние).

Минимальная длительность синхросигнала на входе триггера определяется максимальным временем переключения триггера $t_{..T}$. В двухступенчатом триггере с однократной синхронизацией время переключения второй ступени определяется по отношению к спаду синхроимпульса.

RS-триггеры

RS-триггером называют запоминающий элемент с отдельными информационными входами для установки его в состояние "0" (R -вход) и в состояние 1 (S -вход). Название "RS-триггер" образовано от первых букв слов RESET (сброс) и SET (установка).

R_t	S_t	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	K_6
1	1	1	K_7

Асинхронный RS-триггер

В таблице переходов RS-триггера приняты обозначения: R_t , S_t , Q_t — значения логических переменных в момент времени t на входах R , S и выходе Q , Q_{t+1} — состояние триггера после переключения; K_6 , K_7 — неопределенные коэффициенты на тех набо-

рах, где входные сигналы R_t и S_t одновременно принимают значение лог.1 (запрещенная комбинация сигналов).

Приняв, что запрещенные комбинации исключены, в диаграмме Вейча можно заменить их как лог.1, так и лог.0 в результате получаем логические уравнения асинхронного RS-триггера

$$K_6 = K_7 = 1, \quad Q_{t+1} = S_t \vee \overline{R_t} \wedge Q_t \quad (1)$$

$$K_6 = K_7 = 0, \quad Q_{t+1} = \overline{R_t} \wedge S_t \vee Q_t \quad (2)$$

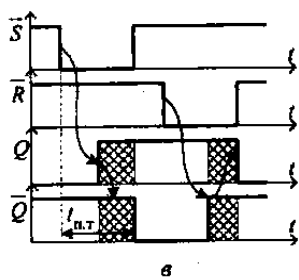
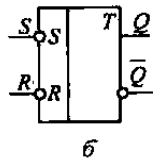
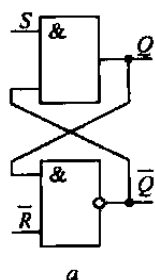
Записанные выражения определяют новое состояние триггера Q_{t+1} в зависимости от его предыдущего состояния Q_t . Для сокращения записи индекс в правой части выражения обычно опускается.

Асинхронный RS-триггер на элементах ИЛИ-НЕ

Преобразуем логическое выражение (2) в виду, удобному для реализации в базе ИЛИ-НЕ

$$Q_{t+1} = \overline{\overline{R} \wedge \overline{S} \vee \overline{Q}} = \overline{\overline{R} \vee \overline{S} \vee \overline{Q}}$$

Асинхронный RS-триггер на элементах И-НЕ



Преобразуем логическое выражение (1) к виду, удобному для реализации на элементах И-НЕ:

$$Q_{t+1} = \overline{\overline{S} \vee \overline{R} \wedge \overline{Q}} = \overline{\overline{S} \wedge \overline{R} \wedge \overline{Q}}$$

особенностью этого триггера является инверсное управление по информационным входам, что отражается в условном графическом обозначении.

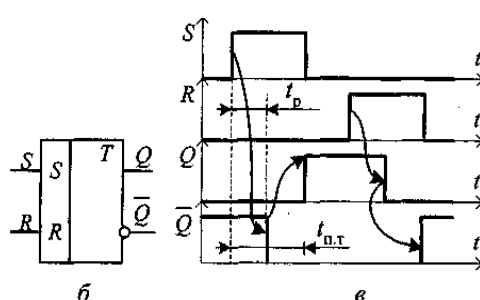
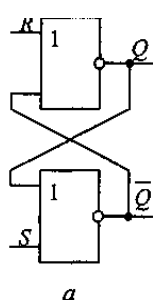
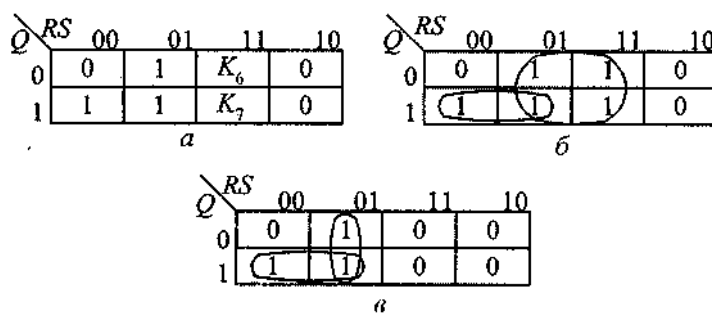
Из анализа диаграмм работы RS-триггера следует, что элементы в схеме переключаются последовательно. Имеется интервал времени, когда на обоих выходах устанавливаются одинаковые сигналы (заштрихованные для последнего случая области) — явление "риск".

Длительность переключения триггера определяется суммой задержек: $t_{nep} = 2t_0$.

Длительность входного сигнала определяется из условия $t_u \geq t_{nep}$. На практике для надежности переключения триггера длительность входного импульса увеличивают на одну задержку, то есть $t_u = 3t_0$. Максимальная и рабочая частоты переключения триггера соответственно равны $f_{max} = 1/(3t_0)$ и $f_p = 1/(2t_0)$.

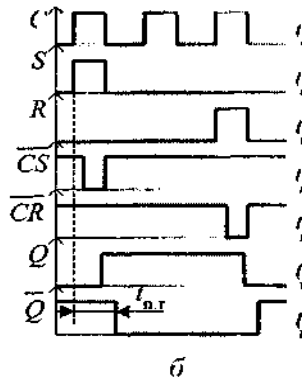
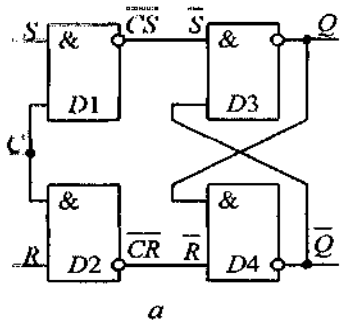
Синхронный RS-триггер

Этот триггер характеризуется наличием дополнительного управляющего входа C , разрешающего изменение состояния триггера, что используется для снижения чувствительности к помехам путем разрешения записи информации в триггер только тогда, когда она достоверна.



к

Для построения синхронного RS-триггера на элементах И-НЕ достаточно заменить переменные S и R на CS и CR соответственно в выражении логического уравнения

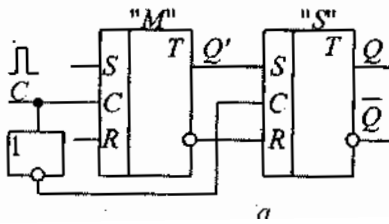


$$Q_{t+1} = \overline{(C \wedge S)} \wedge \overline{(C \wedge R)} \wedge Q$$

Появление дополнительного звена распространения сигнала приводит к снижению быстродействия, так что максимальная и рабочая частоты соответственно равны $f_{\max} = 1/t_{0\text{д}}$ и $f_p = 1/t_{0\text{р}}$.

Синхронный вход этого триггера является *потенциальным*, т.е. запись информации оказывается возможной

на всей длительности разрешающего сигнала по входу C . Для еще большей помехоустойчивости используются триггеры с *динамическим* входом, в которых состояние триггера меняется только на длительности одного из перепадов сигнала на управляющем входе.

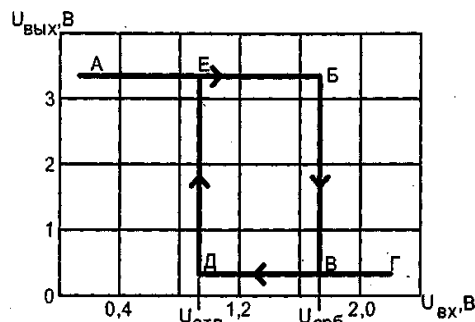
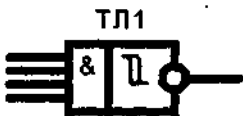


Триггеры с динамическим входом строятся на основе многоступенчатых схем. Простейшая двухступенчатая схема реализуется на базе двух синхронных RS-триггеров по способу «М (master) - S (slave- помощник)». На длительности синхроимпульса на управляющем входе происходит запись информации во входную ступень, а в выходную ступень она переписывается лишь в момент окончания этого импульса.

D – триггеры

Из таблицы переходов для D – триггера следует, что состояние триггера устанавливается значением сигнала на информационном D – входе. В результате оказывается, что асинхронный режим этого триггера лишен смысла – выходной сигнал полностью повторяет входной.

В электронике, использующей переходы от аналоговой формы представления сигналов к цифровой, широко используется специфическое устройство – *триггер Шмитта*. Особенностью этого



устройства является наличие гистерезиса в передаточной характеристике, что может рассматриваться как наличие «памяти» - в переходной зоне $U_{отп}$ $U_{срб}$ выход триггера продолжает сохранять свое предыдущее значение. Наличие гистерезиса позволяет отсекал любой шум, меньшей величины этой зоны, и формировать крутые фронты импульсов. Благодаря своим свойствам: повышенная помехоустойчивость, восстановление искаженной формы импульса, усиление сигналов до стандартных уровней логического нуля и логической единицы, триггер Шмитта находит широкое применение в цифровой технике. Особенно заслуживает внимания его использование в линиях приема и передачи сигналов.

Поскольку триггер – элемент памяти бита цифровой информации, то в цифровой технике под D – триггером понимают только синхронный триггер, что нашло свое отражение в его названии *Delay* (*задержка*) – выходной сигнал повторяет

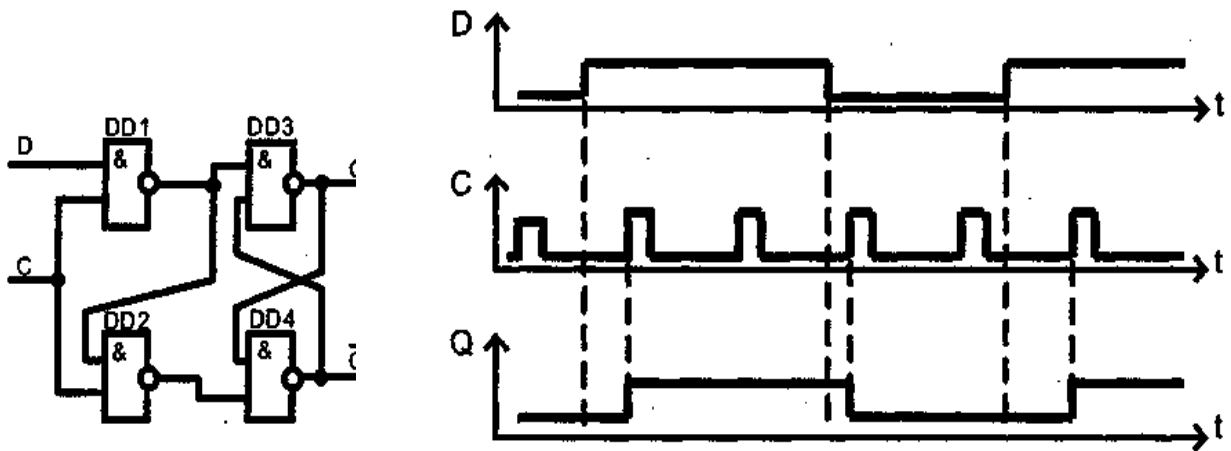
Таблица истинности D-триггера

C	D	Q_{t+1}
0	0	Q_n
0	1	Q_n
1	0	0
1	1	1

только синхронный триггер, что нашло свое отражение в его названии *Delay* (*задержка*) – выходной сигнал повторяет

ет входной с задержкой во времени, определяемой моментов поступления синхросигнала на управляющий C - вход.

Логическое уравнение для D – триггера с потенциальным C - входом имеет вид $Q_{t+1} = Q_t \wedge \bar{C} \vee D \wedge C$

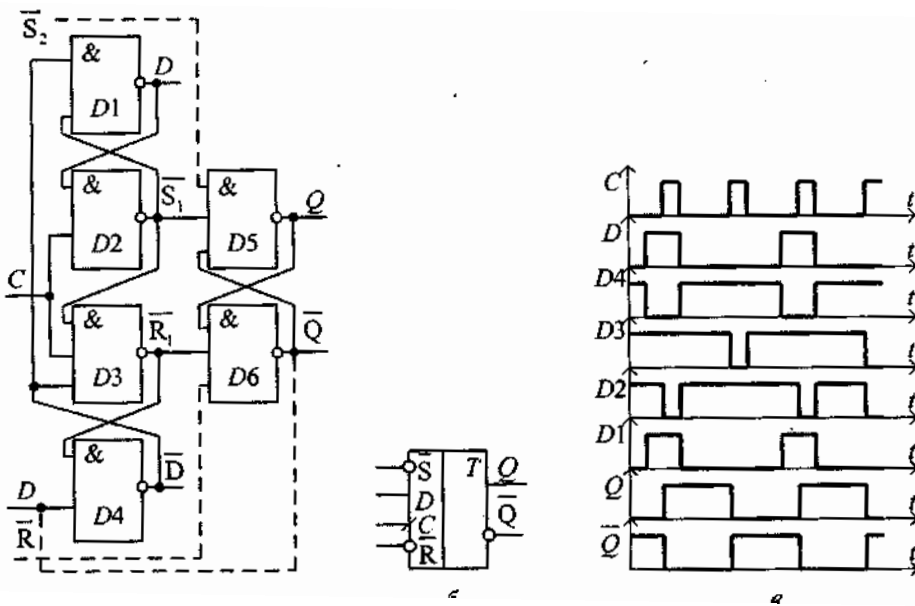


D – триггеры с динамическим C- входом

В триггерах с динамическим управлением записью информации синхроимпульс C активен лишь на коротком интервале времени в окрестности фронта или спада. Поэтому D -триггеры с динамическим управлением обладают высокой помехоустойчивостью. На практике широкое распространение

получили D -триггеры с прямым динамическим управлением по схеме "трех триггеров".

Здесь хранение информации осуществляет основной выходной синхронный RS -триггер (элементы $D5$ и $D6$) с инверсным управлением, а прием тактового и информационного сигналов и задание динамического режима работы обеспечивают два выходных коммутирующих триггера (элементы $D1$, $D2$, $D3$, $D4$). Элемент $D4$



подает инверсное значение входного сигнала D на входы элементов $D1$ и $D3$. Элемент $D1$ повторяет значение сигнала D .

При $C = D = 1$ включается элемент $D2$ и устанавливает основной триггер по входу S_1 в состояние "1"; одновременно блокируется работа элемента $D3$, в связи с чем схема уже не реагирует на изменение входного сигнала. При $C = 1, D = 0$ включается элемент $D3$ и устанавливает основной триггер в состояние "0"; одновременно происходит прием новой информации элементом $D4$. При $C = 0$ обеспечивается режим хранения записанной информации.

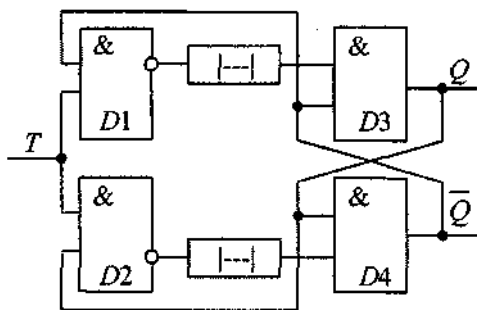
T – триггеры

Триггером типа T называется запоминающий элемент с двумя устойчивыми состояниями и одним информационным T -входом. Состояние T -триггера изменяется на противоположное после каждого поступления счетного сигнала на T -вход. Логика функционирования асинхронного счетного триггера представлена таблицей переходов (табл.

T_t	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

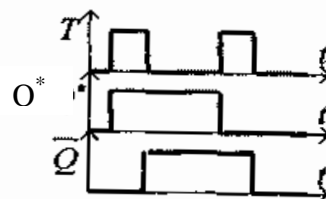
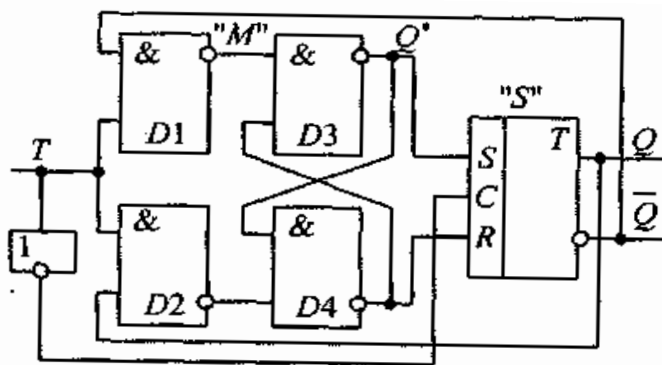
2.10) и описывается логическим уравнением $Q_{t+1} = \bar{T} \wedge Q_t \vee T \wedge \bar{Q}_t$.

Для реализации асинхронного Т-триггера на элементах И-НЕ это выражение преобразуется к виду $Q_{t+1} = \overline{\overline{\overline{T} \wedge Q} \vee \overline{\overline{\overline{T} \wedge \bar{Q}}}} = \overline{\overline{\overline{T} \wedge Q} \wedge \overline{\overline{\overline{T} \wedge \bar{Q}}}}$, где учтено, что $\overline{\overline{\overline{T} \wedge Q}} = \overline{\overline{\overline{T} \wedge \bar{Q}}}$.



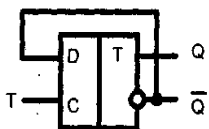
Переключение триггера определяется совместным действием счетных сигналов "Т" и обратной связи выходов. Для исключения "гонок" в триггере сигналы обратной связи не должны изменяться во время действия счетного сигнала "Т". Задержка выходного сигнала может осуществляться линией задержки или дополнительным триггером. На рис. - схема одноступенчатого асинхронного Т-триггера на элементах НЕ И с логическими связями согласно полученному уравнению. Сигналы с выходов элементов D1 и D2 задерживаются на время τ , которое не должно быть меньше длительности поступающих импульсов. В противном случае а длительности входного Т – сигнала оказывается возможным дополнительное опрокидывание триггера, причем многократное. По этой причине Т – триггеры строят с использованием двухступенчатых триггеров или синхронных триггеров с динамическим С-входом.

Сигналы с выходов элементов D1 и D2 задерживаются на время τ , которое не должно быть меньше длительности поступающих импульсов. В противном случае а длительности входного Т – сигнала оказывается возможным дополнительное опрокидывание триггера, причем многократное. По этой причине Т – триггеры строят с использованием двухступенчатых триггеров или синхронных триггеров с динамическим С-входом.



Асинхронный Т-триггер включает в себя два синхронных RS-триггера, при этом на Т-вход основной ступени подается считываемый сигнал, а входы S и R соединены соответственно с выходами вспомогательной ступени.

Пусть в начальном состоянии $Q^* = Q = 0$. При поступлении первого счетного импульса происходит совпадение лог. 1 на входе элемента D1 ($T = 1, Q = 1$), и основной триггер переключается в состояние "1". После окончания входного импульса инвертор в цепи связи между ступенями разрешает перезапись информации во вспомогательный триггер. После поступления второго импульса триггер переключается в состояние "0", то есть осуществляется сложение входных сигналов по модулю два.



Режим Т-триггера может быть просто реализован на базе D – триггера. Действительно, если в логическом уравнении, описывающем работу D – триггера, в качестве сигнала на D – входе использовать собственный инверсный выход, а управляющий С – вход рассматривать как информационный Т – вход, то приходим к уравнению Т-триггера.

JK триггеры

Триггером типа JK называется запоминающий элемент с двумя устойчивыми состояниями и информационными входами J (аналог S) и K (аналог R), которые обеспечивают соответственно раздельную установку состояний "1" и "0". Он функционирует подобно RS триггеру, однако при совпадении

J	K	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

сигналов $JK=1$ переключается в противоположное состояние. Таким образом, JK триггер не имеет запрещенных комбинаций входных сигналов.

Из таблицы переходов асинхронно-следует карта Карно вида (см. рис.), что сать логическое уравнение его функциони-

$$Q_{t+1} = \overline{K} \wedge Q \vee J \wedge \overline{Q} .$$

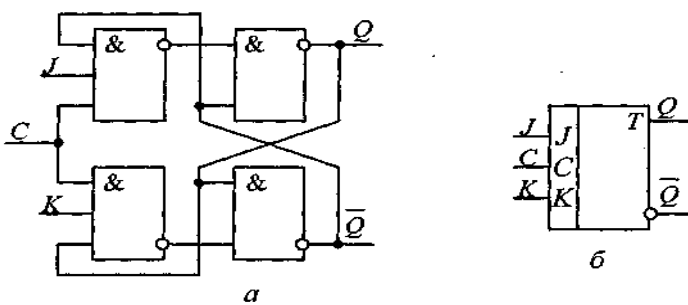
	KJ			
Q	00	01	11	10
0	0	1	1	0
1	1	1	0	0

го JK триггера позволяет записывания

Асинхронный JK триггер так же, как и T триггер, подвержен режиму генерации, поэтому его применение ограничено, а более широко используется синхронный триггер.

Для построения одноступенчатого синхронного JK -триггера на элементах И-НЕ требуется заменить в уравнении для асинхронного варианта переменные K и J на сочетания CK и CJ , после чего выполнить преобразования на основе правил двойной инверсии и правил де Моргана:

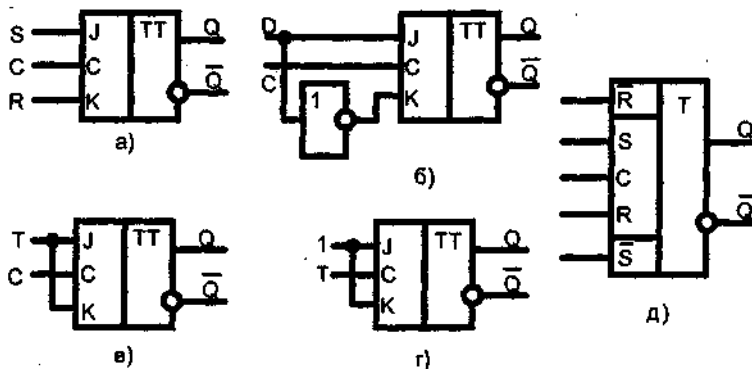
$$Q_{t+1} = \overline{\overline{C \wedge K \wedge Q} \vee \overline{C \wedge J \wedge \overline{Q}}} = \overline{\overline{C \wedge K \wedge Q} \wedge \overline{C \wedge J \wedge \overline{Q}}}$$



Наибольшее распространение получили синхронные JK -триггеры с динамическим C -входом, реализуемые по двухступенчатым схемам.

JK триггер называется универсальным, поскольку позволяет реализовать режим любого из триггеров.:

RS -триггер (рис.а) получается из JK -триггера, когда входы JK используются, как входы S и R соответственно, а запрещенная комбинация не подается.



Если в формуле несинхронного JK -триггера J назвать входом D , а на вход K подать \overline{D} , то получим: $Q^{t+1} = D(\overline{Q^t} + Q^t) = D$, что соответствует несинхронному D -триггеру. Использование синхронного входа позволяет реализовать синхронный D -триггер на основе синхронного JK -триггера (рис.б).

Для получения T -триггера достаточно объединить входы J и K и назвать этот вход входом T , по которому триггер будет переключаться в противоположное состояние, как это должен делать T -триггер (рис.в – синхронный, рис.г – асинхронный)

На рис.д показан JK -триггер с дополнительными RS – входами, обладающими более высоким приоритетом по отношению ко входам собственно JK -триггера.

Применение триггеров в схемах приема внешних сигналов

Источниками сигналов, поступающих на цифровое устройство извне, могут быть контакты тумблеров, кнопок, реле, импульсные сигналы различных электронных схем, сигналы других цифровых устройств. Характеристики этих сигналов, как правило, отличаются от характеристик сигналов, применяемых в данном цифровом устройстве. Входной сигнал должен иметь полярность и амплитуду, которые соответствуют уровням сигналов элементов цифрового устройства. Это требование очевидно как в смысле его постановки, так и в смысле выполнения: нужно лишь ввести делители напряжения или соответствующие усилители. Для взаимных переходов между типовыми уровнями сигналов КМОП, ТТЛ, ЭСЛ выпускаются специальные микросхемы. Менее очевидными оказываются требо-

вания к временным параметрам внешних сигналов. Существенными вопросами здесь являются длительность фронтов входных сигналов, дребезг контактов и привязка входных сигналов к синхросерии.

Формирование длительности фронтов

Длительность фронтов сигналов ряда источников во много раз превышает не только длительность фронтов логических элементов, но даже и длительность тактового периода. Длинные фронты могут вызвать гонки по входу, причем если связанная с этим неодновременность срабатывания настолько велика, что разные элементы срабатывают в различные такты, то любая система синхронизации здесь будет бессильна. Если на длинный фронт наложен высокочастотный шум, то результирующая кривая может за время действия фронта несколько раз пересечь порог срабатывания элемента, вызвав серию срабатываний от существенно однократного сигнала. Сами элементы также плохо реагируют на длинные фронты: когда входной сигнал находится в зоне порога переключения, могут оказаться открытыми сразу оба последовательно включенных выходных транзистора, что вызывает их перегрев; в зоне переключения коэффициент усиления элемента велик (кривая вход-выход идет круто), и нахождение элемента в этом режиме при наличии даже небольшой паразитной обратной связи, например по питанию, приводит к вспышке генерации элемента, перегревающей его самого и создающей помехи. В общем, по целому ряду соображений крутизну фронтов входных сигналов желательно иметь соизмеримой с крутизной фронтов элементов.

Для превращения пологих фронтов в крутые используется *триггер Шмитта*. Он, в частности, является обязательным элементом шинных формирователей (приемопередатчиков) — специальных функциональных узлов, предназначенных для обеспечения передачи сигналов по линиям связи, объединяющим различные устройства.

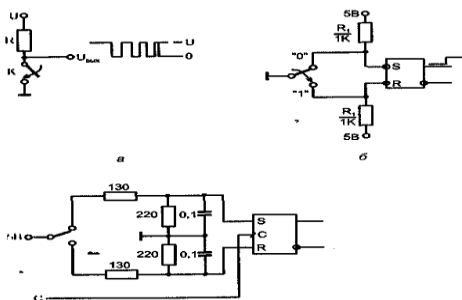
Ввод логических сигналов от механических ключей

Ввод логических сигналов от механических ключей — одно из типовых действий, позволяющее оператору воздействовать на цифровое устройство.

Механические ключи имеют упругость, их коммутация — сложный процесс. После первого соударения контактов происходит ряд упругих отскоков, называемых дребезгом контактов, поэтому вместо однократного перепада напряжения ключи создают целую серию импульсов (рис. а).

Длительность упругих колебаний ключей зависит от их конструкции, обычно она лежит в диапазоне 1...10 мс. Такой сигнал нельзя вводить в цифровое устройство, т. к. он может создать множество ложных переключений.

Для получения "очищенного" от дребезга контактов сигнала принимают специальные меры — программные или схемные. Программные методы! вводят паузу между каждым нажатием ключа и использованием формируемого ключом сигнала (серия пустых команд NOP в программе, выполняемой системой). В схемных методах борьбы с дребезгом контактов используются свойства триггеров.



С помощью триггеров выходное напряжение ключа очищается от паразитных колебаний и превращается в стандартный логический сигнал. Для работы с перекидными ключами (однополюсными ключами на два положения) часто используется схема (рис. б), в которой верхнее положение ключа устанавливает триггер (на входе S нулевое напряжение, на входе R — высокое, задаваемое от источника

ка 5В через резистор R_i ; номиналы напряжений и сопротивлений приводятся здесь с ориентацией на схемотехнику ТТЛ) Нижнее положение ключа ведет к сбросу триггера; (на входе R нулевое напряжение, на входе S — высокое). При изменении состояния ключа возникают упругие отскоки от контактов. Первое же соударение приводит триггер в соответствующее состояние, а при отскоке ключа, когда он находится в воздухе, оба входа триггера получают пассивные сигналы логической единицы (высокие напряжения от цепочек "источник-резистор"), т. е. триггер попадает в режим хранения уже установленного правильного состояния. С помощью подобной схемы производится асинхронный ввод сигнала от механического ключа. Резисторы R_j могут быть достаточно высокоомными, т. к. через них замыкается только относительно малый ток входной цепи триггера при единичном сигнале на нем I_{Bx} .

Синхронизированный с тактовыми сигналами ввод (рис. в) осуществляется с помощью тактирования триггера. Видоизменения схемы в сравнении с предыдущей объясняются тем, что синхронный триггер имеет прямые входы установки и сброса, и тем, что для повышения помехоустойчивости схемы добавлены конденсаторы. Принцип работы схемы сохраняется. Первый же тактовый импульс, пришедший после переключения ключа, формирует выходной сигнал. Резисторы низкоомны, поскольку через них замыкаются входные токи триггера при обоих значениях логического сигнала на них, в том числе и значительные по величине токи $I_{ВХQ}$. Отношение сопротивлений плеч делителей обеспечивает подачу на входы триггера необходимых уровней U_i .

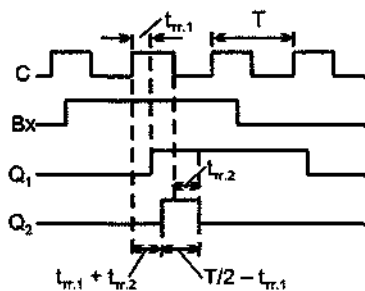
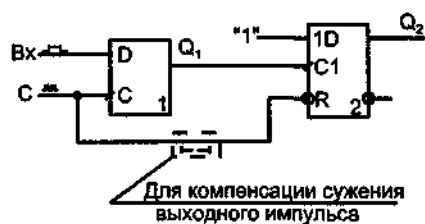
Иногда сигналы от механических ключей вводят с помощью более простых схем, содержащих интегрирующие RC-цепочки. В этом случае переходный процесс при идеальном замыкании ключа имел бы форму экспоненты, а в реальной ситуации эта экспонента будет с горизонтальными участками, соответствующими отскокам ключа, когда он находится в воздухе, и ток емкости нулевой. Так как подобная кривая монотонна, после достижения ею порогового значения логический элемент переключается однократно.

Синхронизаторы одиночных импульсов

Синхронизаторы одиночных импульсов вырабатывают под воздействием асинхронного входного сигнала импульс, принадлежащий тактовой последовательности ТИ. Такой импульс может понадобиться для запуска устройства, реализации пошагового режима его работы и т. д.

Привязка одиночного импульса к тактовой системе обязательна для правильного его восприятия синхронными цифровыми устройствами.

При реализации синхронизаторов следует организовать следующие процессы: разрешить прохождение очередного целого импульса ТИ на вход схемы и затем снять это разрешение после прохождения всего одного импульса. Этим требованиям соответствует структура: синхронизатор момента воздействия входного сигнала на триггер плюс сам триггер, устанавливаемый и сбрасываемый соседними фронтами ТИ (разнополярными). На рис. приведен один из вариантов указанной структуры. Рассмотрение временных диаграмм его работы свидетельствует о сужении входного импульса относительно тактового на время $t_{\text{т.}i}$. Входной сигнал по длительности должен превышать период тактовых импульсов T , иначе он может не дожидаться фронта, который разрешает его прием в первый триггер, и, таким образом, остаться незамеченным.



разрешить прохождение очередного целого импульса ТИ на вход схемы и затем снять это разрешение после прохождения всего одного импульса. Этим требованиям соответствует структура: синхронизатор момента воздействия входного сигнала на триггер плюс сам триггер, устанавливаемый и сбрасываемый соседними фронтами ТИ (разнополярными). На рис. приведен один из вариантов указанной структуры. Рассмотрение временных диаграмм его работы свидетельствует о сужении входного импульса относительно тактового на время $t_{\text{т.}i}$. Входной сигнал по длительности должен превышать период тактовых импульсов T , иначе он может не дожидаться фронта, который разрешает его прием в первый триггер, и, таким образом, остаться незамеченным.

емый соседними фронтами ТИ (разнополярными). На рис. приведен один из вариантов указанной структуры. Рассмотрение временных диаграмм его работы свидетельствует о сужении входного импульса относительно тактового на время $t_{\text{т.}i}$. Входной сигнал по длительности должен превышать период тактовых импульсов T , иначе он может не дожидаться фронта, который разрешает его прием в первый триггер, и, таким образом, остаться незамеченным.

Контрольные вопросы

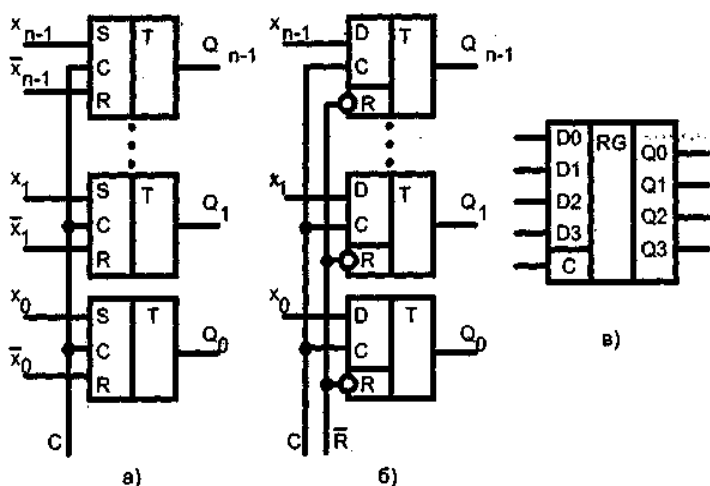
1. Различие между устройствами комбинационного и последовательностного типов.
2. В чем проявляются свойства триггера как элемента памяти.
3. Назначение C – входа триггера.
4. Различие между D и T – триггерами.
5. Различие между RS и JK – триггерами.
6. Происхождение названия D – триггера.
7. Различие между потенциальными и динамическими C – входами триггера.
8. Сигналы, необходимые для записи лог.1 в синхронный JK – триггер.
9. Почему триггер Шмитта относят к классу триггеров.
10. Причины использования триггеров в качестве формирователей фронтов импульсов, поступающих от удаленных источников.
11. Цель применения триггеров в схемах приема сигналов от механических ключей.

Функциональные узлы последовательностного типа

Типовые функциональные узлы

Регистры

Регистром называется последовательное устройство, предназначенное для хранения ограниченного объема информации, представленной в виде многоразрядного двоичного кода. Регистры входят в состав микропроцессоров в качестве сверхоперативных запоминающих устройств (СОЗУ).



Из определения вытекает, что регистры должны содержать элементы памяти и реализуются на триггерах для побитного хранения информации. Но организация регистров предполагает отдельные биты как составляющие некоторой цифровой информации, что требует введения вспомогательных комбинационных схем для сохранения этих связей.

Основная функция регистра — хранение многоразрядного числа, которое представлено в двоичной системе счисления. Таким образом, для хранения n -разрядного двоичного числа регистр должен содержать n -триггеров.

По способу приема информации регистры подразделяются на:

- ◆ параллельные — информация записывается и считывается только в параллельной форме;
- ◆ последовательные (сдвигающие, сдвиговые) — запись и считывание информации происходит только в последовательной форме;
- ◆ последовательно-параллельные, такие универсальные регистры можно использовать для преобразования кода из параллельного в последовательный и наоборот.

По числу каналов передачи информации регистры подразделяются:

- ◆ однофазные — информация вводится либо в прямом, либо в обратном коде;
- ◆ парафазные — одновременно и в прямом, и в обратном кодах.

При этом как входная, так и выходная информация может быть представлена как в прямом, так и инверсном входе.

По способу тактирования (синхронизации) регистры подразделяют:

- ◆ одноктактные, синхронизируемые одной управляющей последовательностью;
- ◆ двуктактные;
- ◆ многотактные, управляемые несколькими последовательностями импульсов.

Сдвиговые регистры могут быть однонаправленные, когда записанную информацию сдвигают только в одном направлении, и двунаправленные, в которых информация сдвигается как вправо, так и влево. Для таких регистров предусматривается специальный вход включения режима направления сдвига.

Чаще всего регистры строят на основе D и JK -триггеров.

Параллельный регистр

В параллельных регистрах прием и выдача двоичных слов осуществляется по всем разрядам одновременно. Поэтому триггеры, соответствующие разным разрядам не связаны между собой.

Каждый триггер в параллельном регистре имеет свои независимые входы и выходы. Тактовые входы всех триггеров соединены между собой.

Подавая на входы *RS*- триггеров (рис. а) парафазные сигналы, в регистр записывается двоичный код. Запись входного кода происходит во время прихода синхроимпульса на вход *C*. Если для записи в данном регистре используется только прямой или только обратный входной код, то запись информации выполняется за два такта синхронизации. По первому такту необходимо сбросить или установить все триггеры регистра подачей на соответствующие входы активного логического уровня, а только потом, по второму синхросигналу, - записать в регистр новую информацию.

На рис. б показан параллельный однокантный регистр с однофазным входом на *D*-триггерах. В таком регистре при уровне логической 1 на *C*-синхровходе все триггеры устанавливаются в состояния, определяемые действующим на *D*-входах входными сигналами. Для записи информации необходим только один импульс синхронизации. Дополнительный вход *R* начальной установки переводит все триггеры в нулевые состояния одним управляющим импульсом.

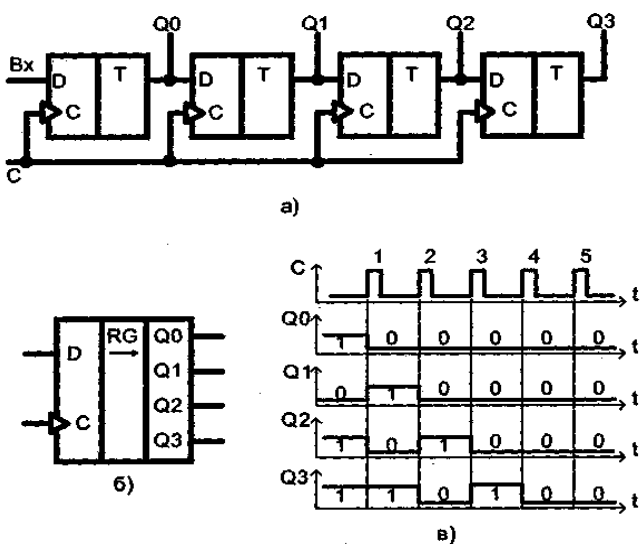
На рис. в показано условное графическое обозначение четырехразрядного параллельного регистра. Этот триггер имеет четыре однофазных прямых входа *DO, D1, D2, D3*, четыре соответствующих выхода *Q0, Q1, Q2, Q3* и вход синхронизации *C*.

Параллельные регистры могут иметь дополнительно вспомогательные комбинационные схемы, которые расширяют функциональные возможности. Они используются для установки начального состояния, режима приема, хранения либо выдачи информации, реализуют запись с двух направлений и т. д.. В этом случае регистр имеет соответствующие входы.

Параллельные регистры позволяют реализовать поразрядные (без переносов) элементарные логические операции.

Действительно, если в регистр, реализованный на *RS*- или *JK*-триггерах и хранящий число *A*, записать единичные биты числа *B* без предварительного сброса, то в регистре окажется записанным число $C = A \vee B$. Для реализации логического умножения достаточно записать нулевые биты числа *B*. Операция инверсии осуществляется использованием инверсных выходов регистра. Для реализации операции «ИСКЛЮЧАЮЩЕЕ ИЛИ» параллельные регистры реализуются на *T*-триггерах.

Сдвиговые регистры



В сдвиговом (последовательном) регистре триггеры соединены последовательно, т. е. выходы предыдущего триггера передают информацию на входы последующего. Простейший однофазный сдвиговой регистр на *D*-триггерах изображен на рис.а.

Управление осуществляется фронтом синхронизирующего сигнала, который является общим для всех триггеров. Пусть триггеры регистра находятся в состоянии: $Q0 = 1, Q1 = 0, Q2 = 1, Q3 = 1$. Предположим, что на вход поступает нулевая информация. При отсутствии синхроимпульса в регистре хранится информация без изменения. В момент положительного фронта синхронизирующего сигнала каждый из триггеров устанавливается в состояние, соответствующее действовавшему на его входе *D* сигналу:

- ◆ по первому тактовому импульсу: лог. 1 из *Q0* в *Q1*, лог. 0 из *Q1* в *Q2*; лог. 1 из *Q2* в *Q3*;
- ◆ по второму тактовому импульсу: лог. 0 из *Q0* в *Q1*, лог. 1 из *Q1* в *Q2*; лог. 0 из *Q2* в *Q3*;
- ◆ по третьему тактовому импульсу: лог. 0 из *Q0* в *Q1*, лог. 0 из *Q1* в *Q2*; лог. 1 из *Q2* в *Q3*.

Работу сдвигового регистра можно проиллюстри-

Таблица состояний сдвигового регистра

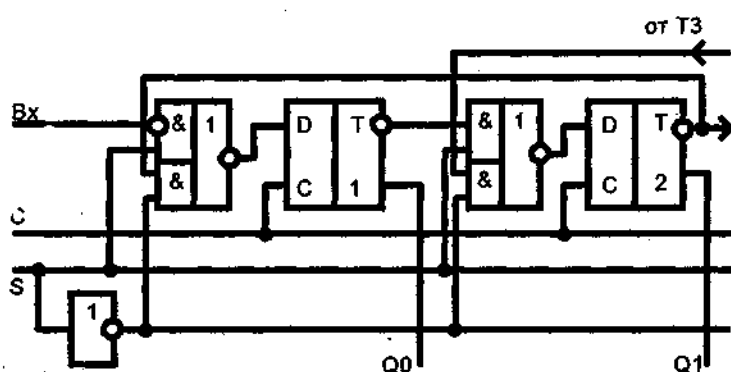
Номер разряда	<i>Q0</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>
1 такт	1	0	1	1
	→	→	→	→
	0	1	0	1
2 такт	0	1	0	1
	→	→	→	→
	0	0	1	0
3 такт	0	0	1	0
	→	→	→	→
	0	0	0	1

ровать рис. в и таблицей, в которых приведено содержимое регистров при выполнении последовательных сдвигов вправо.

Суть сдвига состоит в том, что логическая переменная (1 либо 0), записанная в i -разряде, передается в соседний справа ($i + 1$) разряд. В крайний левый разряд заносится число, подаваемое с внешнего входа. Из крайнего правого разряда двоичный код последовательно поступает из регистра во внешнюю цепь. В данном случае - выход Q_3 .

Для осуществления сдвига влево необходимо в сдвиговом регистре ввести элементы, изменяющие направление передачи информации, т. е. изменить связи между триггерами, подключая выход триггера ко входу соседнего слева разряда, например, с использованием элементов *И-ИЛИ-НЕ*, изменяющих направление передачи сигнала. При $S = 1$ открываются верхние по схеме элементы *И*, осуществляется передача сигнала в регистр из левого триггера в правый. При $S = 0$ открываются нижние по схеме элементы *И*, информация сдвигается влево.

В сдвиговых регистрах используются только двухступенчатые триггеры или триггеры с динамическим управлением. Это гарантирует сдвиг информации строго на



один разряд по каждому импульсу синхронизации. При использовании других триггеров процесс сдвига становится неуправляемым и за один импульс синхронизации кодовое число может быть сдвинуто на несколько разрядов.

При использовании одного входа и только одного выхода двоичное слово вводится в сдвиговый регистр и выводится из него в последовательной форме.

Такое представление позволяет уменьшить разрядность шин в цифровом устройстве. С другой стороны, при этом замедляется быстродействие. Для осуществления ввода либо вывода числа в n -разрядном регистре требуется n -синхронизирующих импульсов по числу триггеров.

Одним из основных применений сдвиговых фильтров является их использование в качестве регулируемых цифровых линий задержки. Действительно, на выходе сдвигового регистра сигнал появляется с задержкой, определяемой как разрядностью регистра, так и частотой следования синхроимпульсов, которые могут рассматриваться в качестве параметров этой линии задержки и могут использоваться для управления величиной времени задержки.

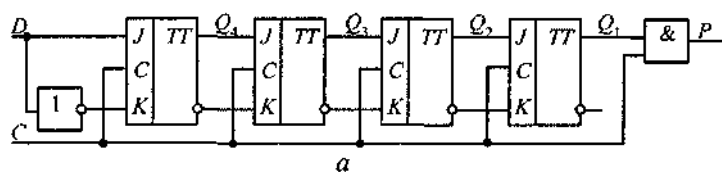
Параллельно-последовательные регистры

В сдвиговом регистре у каждого триггера возможен самостоятельный вывод. Через n -тактов в n -разрядный регистр последовательно будет записано число, которое может быть затем снято в параллельной форме одновременно с выходов триггеров всех разрядов. Так осуществляется **преобразование двоичного кода из последовательной формы в параллельную**.

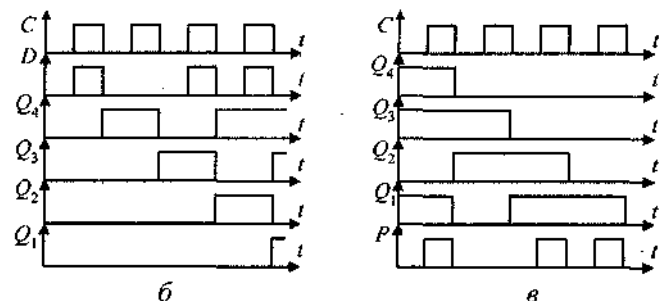
Каждый триггер сдвигового регистра может быть снабжен через вспомогательную комбинационную схему входами начальной установки в любое заданное состояние. Это позволяет осуществить прием числа в параллельной форме. После того как n -разрядное число будет записано в регистр, путем подачи n -синхронизирующих импульсов двоичный код в последовательной форме снимается с выхода старшего разряда регистра. Так происходит **преобразование параллельного кода в последовательный**.

Устройства взаимного преобразования последовательного и параллельного кодов

На рис.а приведен пример четырехразрядного регистра сдвига вправо на JK-триггерах, реализующего преобразование кодов. Старший разряд регистра с помощью инвертора на K-входе работает в режиме D-триггера.

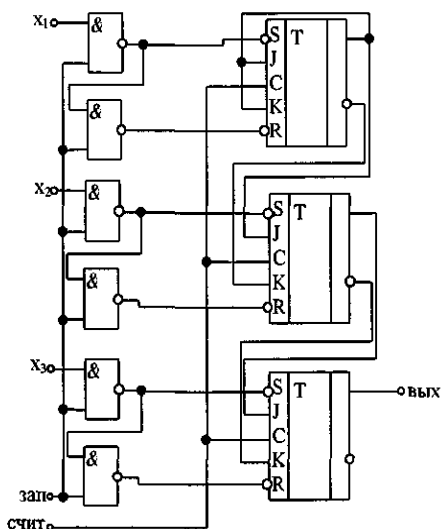


Пусть вход регистра по линии D поступает последовательный код слова $A=1101$ в направлении от младших разрядов к старшим. Значение разрядов слова поступает одновременно с синхроимпульсами, которые обеспечивают как прием кода в старший разряд, так и одновременный сдвиг содержимого регистра вправо (рис. б). После прихода четырех синхроимпульсов на выходах регистра $Q_4 - Q_1$ устанавливается код 1101. Таким



образом осуществляется преобразование последовательного кода в параллельный, называемое последовательным вводом слова в регистр.

Преобразование параллельного кода в последовательный также выполняется сдвигом хранимого слова. Процесс преобразования параллельного кода слова $A=1101$ в последовательный в направлении от младших разрядов к старшим с помощью сдвига вправо показан на рис. в.



В приведенной схеме JK-триггеры дважды превращены в D-триггеры.

Одно превращение осуществлено для организации записи с использованием вспомогательных инверсных входов S и R и, добавлением двух элементов И-НЕ. Запись числа в регистр в параллельном коде происходит при подаче числа на входы x_1, x_2, x_3 и сигнала «1» на вход «зап».

Для считывания информации в последовательном коде JK-триггеры второй раз превращены в D-триггеры, у которых выходы каждого триггера соединены со входом D следующего, как это делается в последовательном регистре. В этом случае JK-триггеры превращены в D-триггеры способом, при котором вместо дополнительного инвертора используется инверсный выход предыдущего триггера. Исключение составляет первый триггер, у которого входы J и K объединены и соединены с основным входом этого же

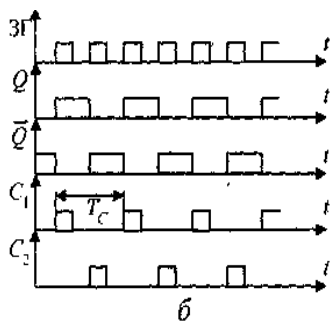
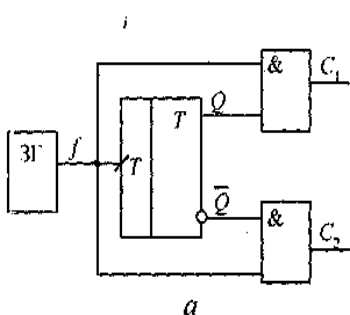
триггера. Благодаря такой схеме после подачи импульсов считывания первый триггер оказывается в состоянии «0».

Одним из известных применений подобных регистров является реализация с их помощью операций умножения и деления чисел на степень числа 2. Действительно, в двоичной арифметике умножение или деление числа на 2 равносильно поразрядному сдвигу соответственно влево или вправо, что и используется для реализации этих операций.

Еще одно из их применений – построение распределителей тактов.

Распределители тактов

В системах синхронизации и устройствах управления используют распределители тактов (РТ), которые вырабатывают серии импульсов. Простейший способ построения распреде-

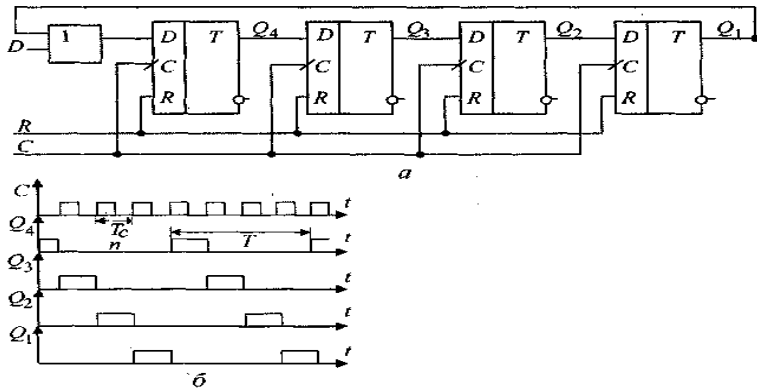


лителя тактов на два канала состоит в делении счетными триггерами частоты прямоугольных импульсов от задающего генератора (ЗГ).

На прямом и инверсном выходах триггера с помощью схем совпадения образуются две последовательности импульсов фаз $C_1 = fQ$ и $C_2 = f\bar{Q}$. Длительность такта синхронизации T_c в два раза больше периода следования импульсов ЗГ.

Распределители на большее число каналов могут быть построены на кольцевых регистрах, в которых выход младшего разряда соединен с информационным входом старшего, если реализован сдвиг вправо (рис. а), или имеется связь от старшего разряда к младшему при сдвиге влево.

Кольцевой регистр в качестве распределителя тактов функционирует следующим образом. Вначале через схему ИЛИ в старший разряд Q_4 записывается лог.1. Затем по фронту каждого синхроимпульса единица сдвигается вправо и поочередно появляется на выходах Q_3 , Q_2 и Q_1 . С приходом четвертого синхроимпульса единица из младшего разряда Q_1 , по кольцевой связи и через схему ИЛИ снова записывается в старший разряд. Число выходов распределителя тактов определяется количеством триггеров кольцевого регистра. В распределителе тактов на основе четырехразрядного кольцевого регистра формируются импульсы фаз C_1 , C_2 , C_3 и C_4 , которые образуют машинный такт T длительностью в четыре периода синхроимпульсов, то есть $T = 4T_c$.



Недостатком приведенной схемы является возможность сбоя - появления лишних единиц в разрядах, например, 1100, 1001. В практических схемах кольцевых регистров используется схема самовосстановления после сбоя, которая при неправильной работе автоматически возвращает регистр в исходное состояние.

Регистры используют также для выполнения и

других функций. В частности, на их основе строятся такие устройства, как линии задержки на заданное число тактов, накапливающие сумматоры, формирователи импульсов большей длительности, генераторы псевдослучайных последовательностей. Регистры применяют в арифметическо-логических устройствах в качестве узлов, выполняющих различные логические операции.

В частности, для подключения к магистрали внешнего устройства используются буферные регистры. В отличие от шинных формирователей буферные регистры содержат в своем составе триггеры и способны хранить поступающие данные, что составляет их важную функцию.

Буферные регистры с тремя состояниями обеспечивают портам возможность отключения от магистрали под воздействием управляющих сигналов E , а также необходимую нагрузочную способность. Запись информации в регистры происходит при сигнале строба $STB = 1$. Через порты ввода данные поступают в магистраль, а через порты вывода данные с магистрали передаются тому либо иному модулю. Порты ввода-вывода могут выполнять обе указанные операции.

Шинные формирователи, приемопередатчики, буферные регистры связывают выходы микропроцессора с внешней средой, так как нагрузочная способность микропроцессора недостаточна.

Счетчики

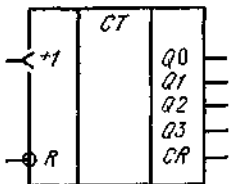
Общая характеристика счетчиков

Счетчиком называется последовательностное устройство, предназначенное для счета входных импульсов и фиксации их числа. Счет импульсов является одной из наиболее распространенных операций в устройствах дискретной обработки информации.

Основное функциональное назначение счетчиков:

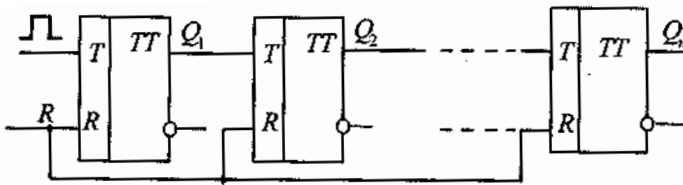
- ◆ регистрация числа импульсов, поступивших на вход;
- ◆ деление частоты (используется однозначная связь между частотами следования периодической последовательности импульсов на входе счетчика и его выходе $F_{вых} = F_{вх} / K$).

По мере поступления входных сигналов счетчик последовательно перебирает свои состояния в определенном для данной схемы порядке. Длину списка используемых состояний (параметр K) называют *модулем пересчета* или *емкостью* счетчика. Одно из возможных состояний счетчика принимается за *начальное (нулевое)*. Если счетчик начал считать с начального состояния, то через каждые K сигналов в нем снова устанавливается начальное состояние, а на выходе счетчика при этом появляется сигнал *K -ичного переноса CR* .



$Q0—Q3$ — выходы счетчика, комбинация значений которых определяет номер его состояния; CR — выход переноса, R - вход общего сброса. Часто счетчик снабжен и входами данных Dj для *параллельной загрузки* произвольного кода. Загрузка осуществляется при подаче сигнала на еще один вход — вход параллельной загрузки L .

Из определения и логики работы счетчиков следует, что их текущее состояние зависит не только от нового пришедшего импульса, но также и от количества предыдущих импульсов, т.е. счетчики относятся к устройствам с памятью.



Строятся счетчики, как и регистры, на основе одноступенчатых связанных между собой триггеров. Наиболее часто используются T - и JK -триггеры. (T -триггеры так и называются — *счетные триггеры*). В JK -триггерах счетный режим обеспечивается при $J=K=1$. Комбинационные элементы в счетчиках используются для управления работой триггеров. Число триггеров определяет максимальное количество импульсов, которое может быть подсчитано счетчиком.

В счетчиках выполняются следующие логические операции:

- ◆ установка в нулевое состояние (сброс);
- ◆ запись входной информации в параллельной форме — начального кода, с которого начинается счет;
- ◆ хранение записанной информации;
- ◆ выдача хранимой информации в параллельной форме;
- ◆ инкремент — увеличение хранящегося числа на единицу;
- ◆ декремент — уменьшение хранящегося числа на единицу.

По значению модуля счета K различают:

- ◆ двоичные счетчики ($K=2^n$, где n — число используемых триггеров);
- ◆ недвоичные ($K < 2^n$).

Различные схемы счетчиков могут перебирать свои состояния в самом различном порядке. В двоичных счетчиках порядок смены состояний триггеров соответствует последовательности двоичных чисел. Кроме того, применяют *одинарное* кодирование, когда состояние счетчика представлено местом расположения единственной единицы (например, сдвигающий регистр с одной двигающейся единицей — *кольцевой счетчик, распределитель тактов*), возможно *унитарное* кодирование, когда состояние представлено числом единиц и более сложные виды кодирования.

Различают *суммирующие, вычитающие* и *реверсивные* счетчики. Суммирующий счетчик перебирает свои состояния в возрастающем порядке, *вычитающий* - в убывающем. В *реверсивном* счетчике предусматривается вход для управляющего сигнала, устанавливающего направление счета.

Счетчики, которые в процессе работы для переключения требуют подачи синхросигналов, называют *синхронными (тактируемыми)*, а счетчики, у которых для переключения достаточно подавать лишь входные сигналы, — *асинхронными*.

Суммирующий вход счетчика обозначается «+1», вычитающий — «-1». Это счетные входы. У асинхронных счетчиков они помечаются таким же символом, как и счетный вход Т-триггера, указывающим полярность перепада входного сигнала, по которому счетчик меняет состояние своего выхода.

Временные характеристики счетчиков оцениваются их *разрешающей способностью, быстродействием и временем установления кода*.

Разрешающая способность t_{pc} определяется минимальным интервалом времени между двумя входными импульсами, при котором еще сохраняется работоспособность счетчика. Очевидно, что этот параметр определяется характеристиками первого триггера счетчика $t_{pc} = t_T$.

Быстродействие характеризуется максимальной частотой входных импульсов в режиме деления частоты $F_{max} = 1/t_T$.

Время установления кода $t_{уст}$ отсчитывается от начала входного импульса до момента получения нового состояния на выходе счетчика.

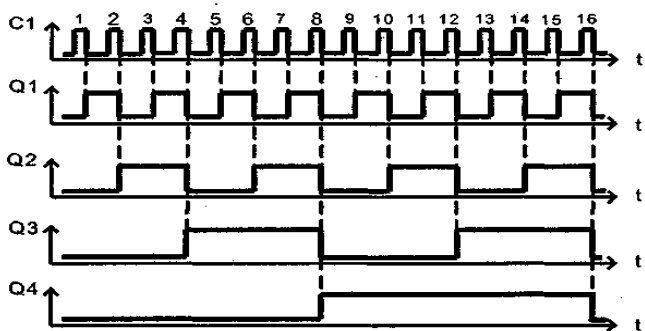
Суммирующий двоичный асинхронный счетчик с последовательным переносом

Временные диаграммы показывают состояния каждого из триггеров при поступлении на вход периодической последовательности импульсов.

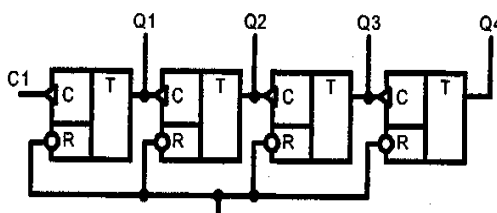
В таблице приведено состояние триггеров, соответствующее числу поступивших на вход импульсов.

Таблица истинности четырехразрядного счетчика

Число поступивших импульсов		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Выходы	Q1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	Q2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	Q3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	Q4	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0



Как можно заметить, при срабатывании по заднему фронту (срезу) триггеры имеют инверсные динамические входы.



Состояние счетчика в двоичном коде по приходу

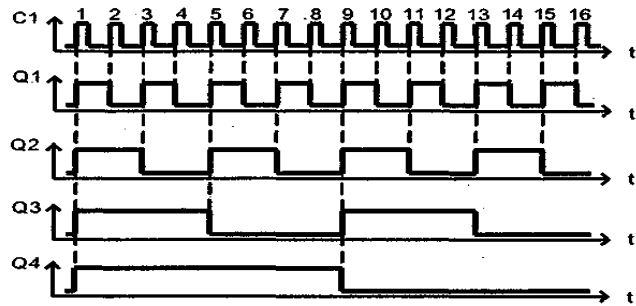
на вход каждого нового импульса увеличивается на единицу - осуществляется операция инкремента.

Так как счетный триггер делит частоту входных импульсов на два, то цепочка из четырех последовательно соединенных триггеров делит частоту на 16. По приходу каждого шестнадцатого импульса счетчик обнуляется и цикл счета начинается сначала.

Вычитающий двоичный асинхронный счетчик с последовательным переносом

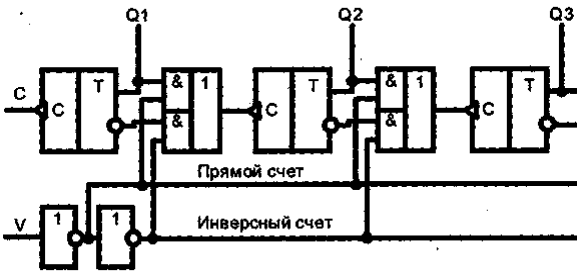
Если исходные асинхронные *T*-триггеры имеют прямые динамические входы (срабатывают по переднему фронту импульса при переходе из 0 в 1), то счетчик превращается в вычитающий. Он выполняет операцию декремента.

Того же эффекта можно достичь, если использовать не прямые, а инверсные выходы предыдущих триггеров счетчика. Это и используется при построении реверсивного счетчика.



Действительно, управляя выбором прямого или инверсного выхода предыдущего триггера, можно реализовать как режим суммирования входных импульсов, так и режим вычитания.

Комбинационная схема для обеспечения режима реверсивного счетчика представляет собой группу их соответствующего числа мультиплексов $2 \rightarrow 1$, причем их схемная может быть различной.



Кроме готовых ИС мультиплексов, они могут быть реализованы на логических элементах. Еще один способ – использование элемента «ИСКЛЮЧАЮЩЕЕ ИЛИ», который является управляемым инвертором, что и требуется для данного случая.

Счетчики с параллельным переносом характеризуются предельной простотой схем, но отличаются большим временем установления кода. Действительно, в предельном случае, когда происходит переполнение счетчика, импульс переполнения на выходе счетчика появится с задержкой, определяемой суммой задержек срабатывания всех его триггеров.

В результате максимальная частота переключения в режиме делителя частоты не может превышать $F_{max} = 1/nt_T$, а время установления $t_{уст} = nt_{здT}$. Это предъявляет завышенные требования к быстродействию триггеров счетчика и может быть устранено за счет схемных решений, обеспечивающих ускоренный перенос сигнала переключения.

Суммирующий двоичный счетчик с параллельным переносом

Свести до минимума время установления можно при условии, что все разряды счетчика будут переключаться одновременно. В этом случае следует отказаться от асинхронных счетчиков и перейти к использованию синхронных.

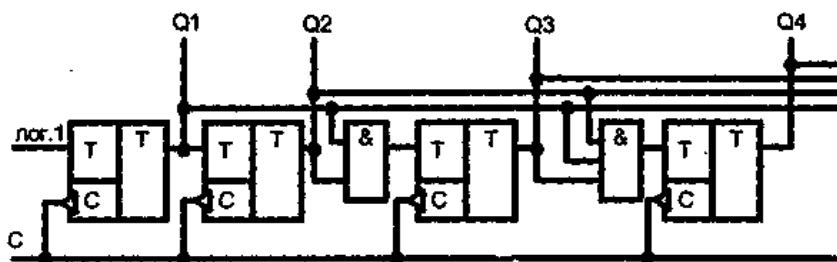
Из анализа таблицы и рис. для суммирующего асинхронного счетчика можно обнаружить, что переключение каждого *i*-го триггера происходит по приходу нового входного импульса при условии, что все предыдущие триггеры находятся в единичном состоянии. Таким образом, оказывается возможным предсказать появление сигнала переполнения от младших разрядов до фактического его появления. Действительно,

$$P_{12} = Q_1 \cdot C_1;$$

$$P_{23} = Q_1 \cdot Q_2 \cdot C;$$

$$P_{34} = Q_1 \cdot Q_2 \cdot Q_3 \cdot C$$

Из этих выражений следует, что сигнал переноса p_i формируется с помощью многовходовых элементов И.

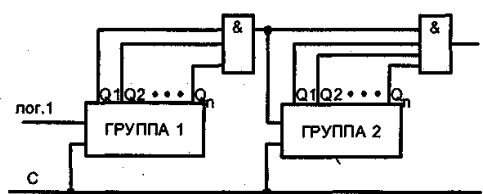


В результате оказывается возможным свести до минимума время установления сигнала на выходе счетчика, исключив влияние задержки срабатывания предыдущих триггеров на момент появления сигнала переноса на входе каждого последующего $t_{уст} = t_{здТ} + t_{здИ}$. Время задержки распространения для элементов «И» заметно меньше задержки распространения для триггера.

Однако, с ростом числа разрядов реализация параллельных счетчиков вызывает затруднения. Растет число входов элементов И, растет нагрузка на выходы триггеров. Кроме того, одновременное переключение сразу многих триггеров в счетчике создает значительный импульс тока в цепях питания.

Суммирующий счетчик со сквозным (групповым) переносом

При большом числе разрядов используют комбинированную структуру последо-



вательно — параллельного переноса. В этом случае триггеры счётчика объединяются в группы, внутри каждой группы осуществляется параллельный перенос, а между группами — последовательный. При такой организации переноса все схемы умножения имеют одинаковое число входов. Сигнал переноса из группы триггеров возникает при

условии, что все триггеры этой группы находятся в единичном состоянии.

Упрощение комбинированного счетчика с последовательно- параллельным переносом достигается некоторым снижением его быстродействия. Время установления кода в счетчике со сквозным переносом определяется задержкой переключения триггера, задержкой переключения схем «И» в одной группе и количеством групп. Таким образом, быстродействие такого счетчика является промежуточным между быстродействиями счетчиков с последовательным и параллельным переносом.

$$t_{уст} = t_{здИ} (l - 1) + t_{сп},$$

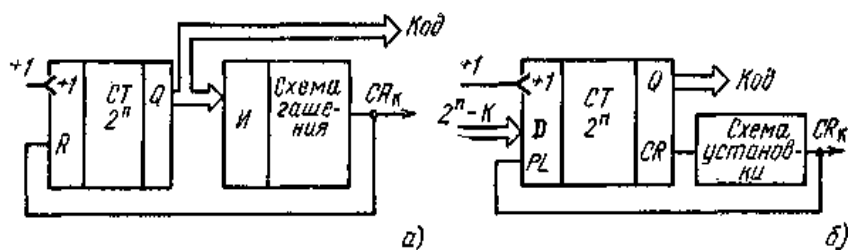
где l — число групп; $t_{сп}$ - время задержки (установления) в группе.

При необходимости формирования счетчиков большой размерности и уменьшения импульсов тока при переключении переходят к формированию счетчиков в коде Грея. Суть кода Грея заключается в том, что при переходе от одной кодовой комбинации к следующей соседней изменяется состояние только одного разряда. Однако полученный результат затем необходимо перевести в двоичный код с использованием преобразователя кода.

Счетчики с модулем счета $K \neq 2^n$

Различные области применения требуют счетчиков с модулями пересчета (основаниями), не только кратными целой степени двойки, но и другими. Например для работы в десятичной системе — 10, для схем часов и календарей — 60, 24, 7... В общем случае требуется строить счетчики по любому заданному основанию K . Иногда пересчет выгоднее реализовать на единственном счетчике, иногда — разложить основание на два сомножителя: целую степень двойки, реализуемую на обычном двоичном счетчике, и оставшееся нечетное число, являющееся основанием счетчика уже меньшего размера, чем он получится, если его строить непосредственно для заданного основания.

На базе ИС счетчиков требуемый счетчик по произвольному основанию можно построить двумя основными способами.



1. Двоичный счетчик разрядности n , такой, чтобы 2^n было больше K , дополняется элементом И, который по состояниям

выходов Q_i обнаруживает код конца счета (обычно $K-1$), после чего по цепи R сбрасывает счетчик в 0. Структурная схема, реализующая этот способ, показана на рис. а. Сигнал, сбрасывающий счетчик, одновременно является и сигналом K -ичного переноса CR_K .

Достоинства способа: - естественная двоичная последовательность кодов от 0 до $K-1$, использование обычно имеющегося в ИС счетчиков входа R . В случае суммирующего счетчика достаточно собрать на элементе И лишь прямые выходы тех триггеров, которые при коде конца счета равны 1. Число входов элемента И, таким образом, зависит от кода конца счета.

2. Второй способ иллюстрирует рис. б. Двоичный счетчик перед началом счета по тракту параллельной загрузки загружается кодом дополнения числа K до 2^n . Кодом конца счета в этом случае является естественное переполнение счетчика, т. е. код *Все единицы*, обнаруживаемый штатным трактом переноса, в результате чего вырабатывается сигнал CR . Сигнал CR , воздействуя на вход L , управляющий параллельной загрузкой, снова устанавливает в счетчике дополнение K до 2^n .

Достоинства способа — использование штатного тракта переноса и имеющихся во многих счетчиках входов параллельной загрузки, легкая смена основания пересчета. Для этого входы D_i нужно подключить не к константам 1 и 0 (питание и общий провод), а к выходу специального регистра начальных состояний.

Недостатком способа является неестественная последовательность получаемых кодов, требующая в случае их использования перекодировки.

Данный способ применяется, когда показания счетчика не важны, а используется лишь сигнал его выходного переноса. Это типично для задачи деления частоты входных сигналов на некоторое число K . Счетчики, выполняющие эту функцию, называют *делителями*.

Попытка реализовать счетчик по произвольному основанию простейшими, наиболее очевидными способами, например путем подачи выхода конъюнктора, обнаруживающего код K , на вход гашения R или выхода переноса CR непосредственно на вход параллельной загрузки L , почти всегда нарушает правила заведения обратных связей в синхронных устройствах и, естественно, порождает сбои из-за гонок.

Ошибки этого типа при построении счетчиков по произвольному основанию очень распространены, поэтому целесообразно подробнее рассмотреть некоторые не совсем очевидные моменты.

Во-первых, если счетчик имеет тракт межразрядного или межгруппового переноса последовательного типа, то в процессе счета триггеры переключаются не строго одновременно. Порядок их переключения может быть любым, и при этом сразу после активного фронта счетного сигнала на некоторые мгновения могут возникнуть коды конца счета. Это приводит к появлению на выходе элемента И помех, способных досрочно сбросить счетчик в 0. Поэтому элемент И или схема, использующая его сигнал, должны открываться лишь спустя время задержки счетчика, например, синхросигналом иной фазы или другим задержанным импульсом. Счетчик с непосредственной связью, имеющий жесткий порядок переключения триггеров, опасных помех такого типа не порождает. Счетчики, кодом конца счета которых являются все единицы, также не имеют этого недостатка, поскольку схемы штатных трактов переноса всегда построены так, что ложных сигналов на выходе CR не появляется.

Во-вторых, для гарантированного гашения счетчика (или для его параллельной загрузки) длительность импульса гашения (или загрузки) должна быть не менее некоторого определенного значения. В то же время, при гашении счетчика сигналом элемента И, реагирующего на код конца счета, любой триггер счетчика, первым переключившийся в 0, тут же вызовет исчезновение кода конца счета и как следствие — сигнала гашения R . Чтобы длительность сигнала гашения была не менее требуемой паспортном, сумма двух величин — задержки счетчика по тракту вход гашения R — выход Q и задержки элемента И конца счета — должна быть гарантированно больше минимально допустимой длительности импульса гашения по R -входу. Это требует гарантии минимальных значений задержек,

входящих в рассматриваемую сумму, а как раз этих гарантий изготовитель практически никогда и не дает.

Несмотря на то что подобные схемы в реальной практике применяются и при этом даже работают, для серийной аппаратуры такие решения нельзя считать грамотными. Как и во всех других случаях, связанных с гонками, эксперимент не может быть критерием правильности решения: при другой температуре или других образцах комплектующих элементов схема может давать сбои.

Чтобы гарантировать правильность работы, сигнал элемента И (или переноса) необходимо или удлинить до требуемого значения, запустив им одновибратор, или запомнить на триггере. Эти узлы и обслуживающие их элементы (гашения триггера и т. п.) на рис. *a* и *б* показаны условно в виде схем гашения и установки.

Счетчики с не двоичными весами разрядов

Многообразие задач, решаемых средствами цифровой электроники, использует счетчики, в которых состояния могут меняться по несколько иным законам. В автоматике, например, широко используется код Грея, характеризующийся тем, что соседние состояния счетчика различаются значениями только одного из разрядов. Такое представление состояний в устройствах силовой автоматики позволяет свести до минимума влияние переходных процессов при управлении мощными установками. Вариант проектирования подобных счетчиков рассмотрим на примере синтеза двоично-десятичного счетчика.

Пусть требуется синтезировать асинхронный счетчик, работающий в коде 5-2-1-1. В соответствии с заданным кодом заполняем левую часть табл. 1 функционирования счётчика (столбцы Q_1, Q_2, Q_3, Q_4).

Таблица 1

<i>n</i>	Q_4	Q_3	Q_2	Q_1	J_1	K_1	J_2	K_2	J_3	K_3	J_4	K_4
0	0	0	0	0	1	-	0	-	-	-	-	-
1	0	0	0	1	-	0	1	-	-	-	-	-
2	0	0	1	1	-	0	-	1	1	-	-	-
3	0	1	0	1	-	0	1	-	-	-	-	-
4	0	1	1	1	-	1	-	1	-	1	1	-
5	1	0	0	0	1	-	0	-	-	-	-	-
6	1	0	0	1	-	0	1	-	-	-	-	-
7	1	0	1	1	-	0	-	1	1	-	-	-
8	1	1	0	1	-	0	0	-	-	-	-	-
9	1	1	1	1	-	1	-	1	-	1	-	1

В таблице *n* - номер состояния счетчика, меняющийся на единицу при подаче каждого счетного импульса; Q_1, Q_2, Q_3, Q_4 - логические переменные на выходе четырех триггеров, первый триггер с выходом Q_1 соответствует первому младшему разряду; *J* и *K* - значения необходимых сигналов на соответствующих входах *JK* - триггеров.

Указанный в задании режим асинхронного счетчика означает, что каждый из универсальных триггеров проектируемого счетчика может переключаться как при подаче на его вход *C* положительного («одиначного») импульса, так и при подаче сигнала на вход *C* с выхода другого триггера., тогда как для синхронного допускается только первый вариант. При этом переключение последующего триггера происходит тогда, когда предыдущий переключается из «1» в «0». Учитывая это, находим необходимое место подключения входов *C* всех четырех триггеров. Первый триггер должен переключаться при подаче первого, пятого и нулевого (десятого) импульса. Ни один из последующих триггеров не может обеспечить всех трех переключений. Поэтому на вход *C* первого триггера нужно подавать счетные импульсы. Они обеспечивают максимальную частоту переключений, а выборка нужных моментов переключения обеспечивается подачей сигналов на вход *J* и *K*.

На вход *C* второго триггера также нужно подать счетные импульсы, так как выходные сигналы ни одного из триггеров не обеспечат его переключения в нужный момент.

Вход *C* третьего триггера нужно соединить с выходом второго, так как его переключение при подаче нулевого, третьего, пятого и восьмого импульсов на вход счетчика из «0» в «0» может быть использовано для переключения третьего триггера.

По тем же соображениям выход третьего триггера нужно соединить со входом четвертого.

Для определения требуемых сигналов на входах *J* и *K* заполняют правую часть табл. 1, используя таблицу функционирования *JK* - триггера (табл. 2). Прочерк в таблице показывает, что значение сигнала на данном входе не вызывает изменения состояния соответствующего триггера. Например, для переключения первого триггера из «0» в «1» при подаче первого счетного импульса требуется, чтобы $J_1 = 1$ в позиции $n = 0$, а значения K_1 может быть любым, что означает прочерк. Если при очередном такте работы на какой-либо триггер сигнал переноса не подается, т.е. триггер, с которого он может прийти, не сбрасывается в «0», то в соответствующих клетках *J* и *K* можно ставить прочерки, так как при любых значениях *J* и *K* триггер не переключается. Таким способом заполняется вся таблица.

Таблица 2

$Q^n \rightarrow Q^{n+1}$	J^n	K^n
0	0	-
0	1	-
1	-	1
1	1	0

Таблица 3

	X_1		X_2
X_3	0101	0111	0110
	1101	1111	1110
X_4	1001	1011	1010
	0001	0011	0010
			0100
			1100
			1000
			0000

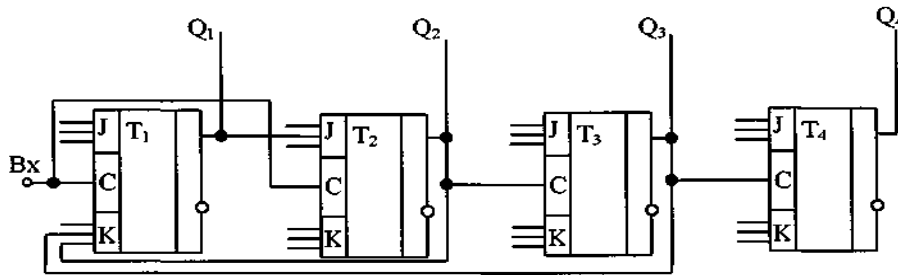
Перенесем теперь данные правой части табл. 1 на диаграммы Вейча.

Таблица 4

J_1	K_1	J_2	K_2
J_3	K_3	J_4	K_4

В табл. 3 приведена диаграмма Вейча для четырех логических переменных. Принимаем за логические переменные значения сигналов на выходах триггеров и заполним восемь диаграмм Вейча (табл. 4), по которым определим сигналы на входах *J* и *K* четырех триггеров. При заполнении диаграмм единицы, нули или прочерк ставятся в тех клетках, в которых находятся соответствующие комбинации выходных переменных *Q*.

В клетках, в которых функция не определена или её значение не играет роли (т.е. ставится прочерк), можно помещать любые значения переменных, чтобы объединить контуром наибольшее количество клеток. Так, в табл. 4 для J_1 во всех клетках можно поставить единицы и объединить одним контуром. Это означает, что $J_1 = 1$. Из всех других диаграмм следует, что $K_1 = Q_2 Q_3$, $J_2 = Q_1$, $K_2 = 1$, $J_3 = 1$, $K_3 = 1$, $J_4 = 1$, $K_4 = 1$. Следовательно, на входы $J_1, K_2, J_3, K_3, J_4, K_4$ надо подать единицы, на вход K_1 - конъюнкцию сигналов с прямых выходов второго и третьего триггеров, а вход J_2 соединить с прямым выходом первого триггера. Если какой-либо вход элементов ТТЛ никуда не подключен, это эквивалентно подаче на этот вход единицы. Таким образом, счетчик синтезирован.

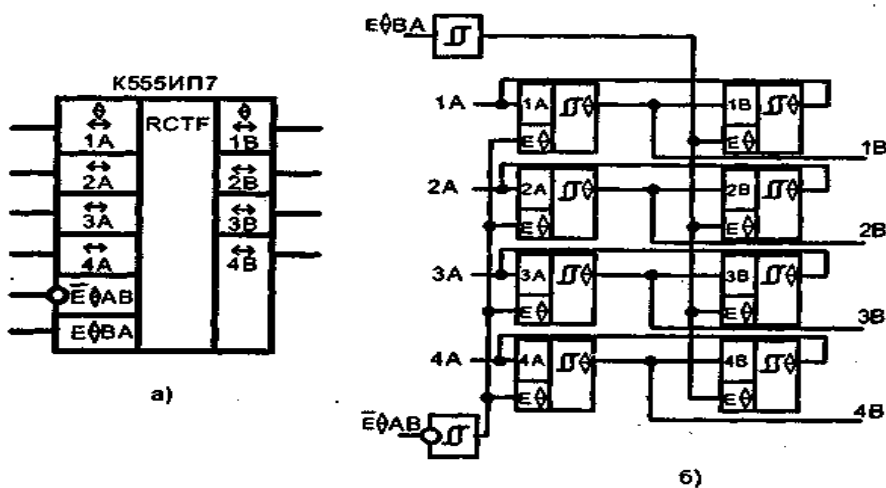


Шинные формирователи

К шине передачи данных обычно подключается много источников и приемников цифровых сигналов. При передаче сигналов по проводникам протекают большие импульсные токи, достигающие десятков миллиампер. Поэтому необходимы специальные микросхемы, обслуживающие шины данных.

Буферные микросхемы не выполняют логических функций, но формируют цифровые сигналы, усиливают импульсы по току, по мощности при работе на шину, отключают источник информации от шины, когда он не участвует в обмене, формируют при необходимости требуемые уровни сигналов логической 1 или логического 0. Буферные усилители передают сигнал без инверсии либо с инверсией, имеют вывод разрешения по входу или по вы-

ходу. Наиболее удобны для применения буферные элементы с тремя состояниями: выходные состояния высокого и низкого уровней, а также третье состояние высокого сопротивления — размыкания выхода по специальной команде. Третье состояние обозначается Z. Выходное сопротивление в режиме Z для буферного элемента составляет сотни кило-



ом. Буферные элементы, включаемые между источником информации и шиной, называются шинные формирователи, приемопередатчики, шинные драйверы или магистральные вентиль-буферы. Двухнаправленные шинные формирователи позволяют в зависимости от управляющего сигнала передавать информацию в шину данных или, наоборот, принимать с шины и передавать приемнику. Различные шинные формирователи отличаются разрядностью, прямыми или инверсными сигналами разрешения-запрета работы, электрическими характеристиками.

На рис приведена схема четырехшинного приемопередатчика, позволяющего передачу информации возможна от одного вывода к другому в любом направлении.

$\bar{E}\phi AB$	$E\phi BA$	A	B	Примечание
1	1	0	0	B→A
1	1	1	1	B→A
0	1	X	X	X
1	0	Z	Z	Z
0	0	0	0	A→B
0	0	1	1	A→B

Схема содержит 10 триггеров Шмитта, два из которых являются управляющими. Восемь триггеров включены попарно. Каждый вход триггера A соединяется с выходом триггера B, и они образуют один вывод A. Выход триггера A соединяется со входом триггера B. Так обра-

зуется вывод B схемы. Эти выводы A-B образуют одну шину приемопередатчика вход-выход или выход-вход.

Работа схемы поясняется таблицей. Если на входах управляющих триггеров подано напряжение высокого уровня

$E\phi AB = 1, \bar{E}\phi AB = 1$, то на $E\phi$ - входы А-триггеров поступает логический нуль, А-триггеры закрыты, а на входы $E\phi$ В-триггеров — логическая единица, В-триггеры открыты. Передача информации осуществляется в направлении от выходов 1B-4B и выходам 1A-4A. При смене напряжения на

управляющих входах с высокого (лог. 1) на низкое (лог. 0) $E \diamond AB = 0, \overline{E} \diamond AB = 0$, лог. 1 поступает на А-триггеры, лог. 0 — на В-триггеры. Передача информации будет осуществляться в обратном направлении от А к В.

При подаче на вход неинвертирующего управляющего триггера напряжения низкого уровня $E \diamond AB = 0$, а на вход инвертирующего — напряжения высокого уровня $\overline{E} \diamond AB = 1$, на выходах управляющих триггеров установится логический ноль — напряжение низкого уровня, что приведет к отключению всех остальных триггеров. Это и есть состояние схемы, когда выводы А и В отключены друг от друга — режим Z высокого выходного сопротивления.

Недостаток подобных шинных формирователей (приемопередатчиков) заключается в том, что возможен прием недопустимой и опасной комбинации сигналов управления $E \diamond AB = 1, \overline{E} \diamond AB = 0$. Тогда на управляющих входах $E \diamond$ всех восьми триггеров установится напряжение высокого уровня лог. 1, и схема сможет пропускать информацию в обоих направлениях, что является недопустимым в работе, так как приводит к сбою аппаратуры.

Проектирование автоматов

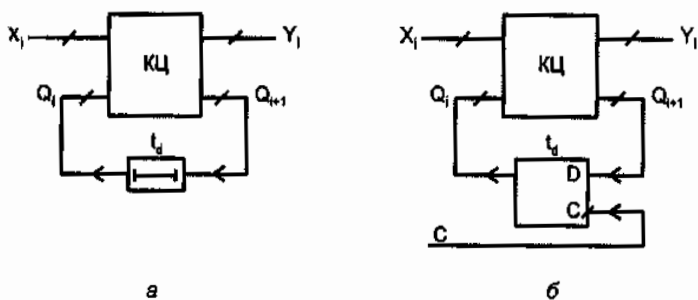
Общая характеристика автоматов

Автоматами, или *последовательностными* (не последовательными!) *схемами*, называют схемы, выходы которых зависят не только от значений входов в данный момент, но и от комбинаций значений входов в определенные прошлые моменты времени. Автомат в некотором смысле помнит прошлые воздействия в отличие от комбинационной схемы (КС), выход которой определяется значениями входов только в настоящее время (разумеется, после окончания переходных процессов). *Автомат имеет память*, КС памяти не имеет. Иногда КС трактуют как частный случай автомата — автомата без памяти, но это, пожалуй, лишь неоправданное усложнение терминологии.

Тем или иным содержимым памяти автомата определяется его *внутреннее состояние*, или просто *состояние*. Внешнее проявление различных состояний — это различные реакции автомата на одни и те же воздействия. Как и мы: находясь в различных состояниях, или улыбаемся, или обижаемся на одну и ту же шутку.

Принципиальным является деление АП на *асинхронные* и *синхронные*. В асинхронных (рис.а) роль элементов памяти играют элементы задержки, через которые сигналы состояния передаются на входы КЦ, чтобы совместно с новым набором входных переменных определить следующую пару значений Y и Q на выходе. Элементы АП переключаются здесь под непосредственным воздействием изменений информационных сигналов. Скорость распространения процесса переключений в цепях асинхронного автомата определяется собственными задержками элементов.

В синхронном АП (рис. б) имеются специальные синхросигналы (тактирующие импульсы)



С, которые разрешают элементам памяти прием данных только в определенные моменты времени. Элементами памяти служат синхронные триггеры. Процесс обработки информации упорядочивается во времени, и в течение одного такта возможно распространение процесса переключения только в строго определенных пределах

тракта обработки информации.

Практическое применение асинхронных автоматов существенно затруднено сильным влиянием на их работу задержек сигналов в цепях АП, создающих статические и динамические риски, гонки элементов памяти (неодновременность срабатывания ЭП даже при одновременной подаче на них входных сигналов) и др. В итоге характерным свойством асинхронного автомата является то, что при

переходе из одного устойчивого состояния в другое он обычно проходит через промежуточные нестабильные состояния. Нельзя сказать, что методы борьбы с нежелательными последствиями рисков и гонок в асинхронных АП отсутствуют, но все же обеспечение предсказуемого поведения АП — сложная проблема. В более или менее сложных АП асинхронные схемы встречаются очень редко, а в простейших схемах применяются. Примером могут служить асинхронные RS-триггеры.

В синхронных автоматах каждое состояние устойчиво и переходные временные состояния не возникают. Концепция борьбы с последствиями рисков и гонок в синхронных автоматах проста — прием информации в элементы памяти разрешается только после завершения в схеме переходных процессов. Это обеспечивается параметрами синхроимпульсов, задающих интервалы времени для завершения тех или иных процессов. В сравнении с асинхронными, синхронные АП значительно проще в проектировании.

На сегодняшний день и достаточно длительную перспективу основным путем построения АП следует считать применение тактирования, т. е. синхронных автоматов.

Цифровые устройства с памятью могут быть инициальными и неинициальными. Инициальные каждый раз начинают функционировать с одного и того же состояния.

По способу формирования выходных сигналов цифровые устройства с памятью делят на два типа: автомат Мура и автомат Мили.

В автоматах Мура выходной сигнал $Y(t)$ зависит от внутреннего состояния $U(t)$ и не зависит явно от входного сигнала $X(t)$. Переключение внутренних состояний $U(t)$ определяется входными сигналами $X(t)$. Функция переходов Φ и функция выходов Ψ записываются в виде:

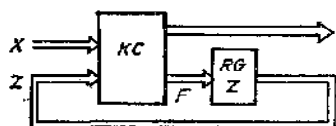
$$\begin{aligned} Y(t) &= \Psi(U(t)); \\ U(t+1) &= \Phi(U(t), X(t)). \end{aligned}$$

Выходной сигнал $Y(t)$ автоматов Мили зависит как от внутреннего состояния $U(t)$, так и от входного сигнала $X(t)$:

$$\begin{aligned} Y(t) &= \Psi(U(t), X(t)); \\ U(t+1) &= \Phi(U(t), X(t)). \end{aligned}$$

В схемотехнике преобладают автоматы Мура, выходы которых являются функциями только состояния автомата.

Некоторые автоматы, в частности, рассмотренные счетчики с недвоичными весами разрядов, принадлежат к числу *автономных автоматов*, которые не имеют информационных входов, и под действием тактовых сигналов меняют свое состояние по алгоритму, определяемому структурой автомата.



Обобщенная структурная схема цифрового автомата показана на рис. Выходной код Y , вырабатываемый КС, есть функция не только входного кода X , но и кода состояния Z , который хранится в *регистре состояний* RGZ . Этот регистр и есть *память* автомата. Чем больше его емкость, тем богаче спектр поведения автомата, шире разнообразие его реакций на одни и те же входные воздействия, полнее учет прошлого опыта. Новое состояние Z^{t+1} в которое переходит автомат после очередного входного воздействия, т. е. новое содержимое RGZ , задается *кодом перехода* F . Код F , так же как и выходной код Y , есть функция и входных сигналов, и состояния автомата непосредственно перед его переходом в новое.

Формально под определение автомата подходит любое цифровое устройство, имеющее хотя бы один триггер. Отдельный триггер, регистр, счетчик любого типа, целая ЭВМ — это все автоматы. Однако рассмотрение триггера или регистра с позиций теории автоматов сегодня имеет скорее академический интерес. Практически схемотехники пользуются типовыми отработанными схемами. Законы функционирования этих схем хорошо известны, и при работе с ними никто не применяет каких-либо специфических приемов теории автоматов. В среде разработчиков понятие «автомат» трактуется уже. О синтезе автоматов начинают говорить когда нужно разработать схему цифрового блока с памятью, закон работы которого, с одной стороны, не очевидно прост, т. е. он не воспроизводится каким-либо стандартным узлом - автоматом типа триггера, счетчика,

распределителя или очевидными их комбинациями, например счетчика с мультиплексором на выходе. Но при этом, с другой стороны, закон поведения проектируемого блока еще не слишком сложен. Под этим понимается ситуация, когда все рабочие комбинации входов, состояний, выходных сигналов как-то еще можно отобразить в виде обозримых списков и таблиц, с которыми реально можно работать. Несмотря на расплывчатость этого определения, именно фактор обозримости имеет здесь решающее значение. Если сложность устройства выходит за рамки обозримости, то его уже не проектируют как единый автомат. Его или разбивают на систему отдельных автоматов, или, чаще, используют общепринятую структуру процессора, а если есть возможность, то готовый микропроцессор. Рассмотрение с позиций теории синтеза автоматов устройств, сложность которых соизмерима со сложностью хотя бы простейшего микропроцессора, также носит или академический, или методологический характер. Построить целый микропроцессор по методике синтеза автоматов практически невозможно. Микропроцессор для этого слишком сложен.

Задачу синтеза автомата средней сложности удобно разбить на три части: *формализация задания, кодирование состояний, синтез комбинационной схемы*. Эти части не полностью автономны, и работа над одной из них обычно требует коррекции результатов двух других.

Формализация задания автомата

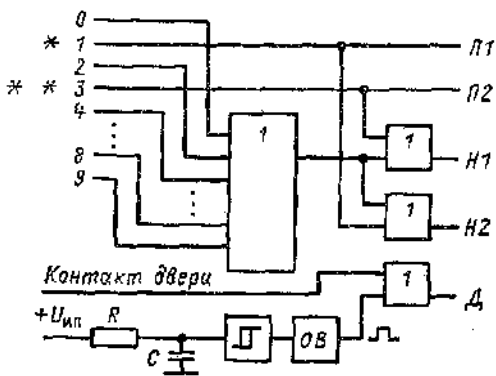
Автоматы обычно строят для управления работой различных технических объектов: исполнительных механизмов, приводов, электронных устройств вплоть до целых ЭВМ. В этой роли автоматы называют программными датчиками, генераторами управляющих сигналов, блоками (устройствами) управления. Управляющим воздействием на объект являются выходы автомата $У$. На входы автомата $Х$ поступают команды оператора (или вышестоящего автомата), сигналы, описывающие окружающую обстановку, сигналы о ситуациях, возникающих в управляемом объекте. Автомат проектируется под конкретный объект с его конкретным алгоритмом управления, и сложность проектирования автомата - именно в его нестандартности. Задание на автомат сначала дается в виде словесного описания закона управления, и, как правило, первый вариант задания оказывается очень неполным, допускающим неоднозначное толкование. Поэтому сначала задание нужно *формализовать*. В процессе этой работы задание уточняется, корректируется. Чтобы с этим справиться, разработчик или сам должен хорошо знать условия работы объекта, или иметь на этапе формализации тесный контакт с заказчиком.

Чтобы проиллюстрировать характер возникающих при синтезе автомата вопросов, процедура синтеза рассмотрим пример простого объекта с хорошо понятными алгоритмом и условиями работы.

Задание. *Спроектировать схему кодового замка двери комнаты. Соленоид должен оттягивать ригель замка после поочередного нажатия двух (а не трех, как обычно: это для упрощения) из десяти кнопок, а именно / и 3. В общем случае обе цифры должны быть различными. В исходно состоянии схема возвращается единичным уровнем сигнала D , снимаемого с контакта при открывании двери.*

Это и есть первичная, словесная, интуитивно ясная и воспринимающаяся как достаточно полная формулировка задания. Требуемый автомат исключительно прост, и схему его можно построить на интуитивном уровне понимания без какой-либо методики. Полезно попытаться сделать это, а затем сравнить затраты времени, схему, а особенно — число ошибок и недоделок с результатом работы по книжке: будет хорошо видно, на каких этапах методика помогает интуиции даже в случае очень простой задачи.

Проектируемая схема не может быть просто комбинационной. Это автомат, поскольку на один и тот же входной сигнал, например на нажатие кнопки 3, схема должна реагировать по-разному: как на неверный сигнал, если это первое нажатие, как на верный, если перед этим была нажата кнопка 1, и никак не реагировать после того, как соленоид уже сработал, а дверь еще не открыта!.



Процесс формализации задания удобно разбить на этапы.

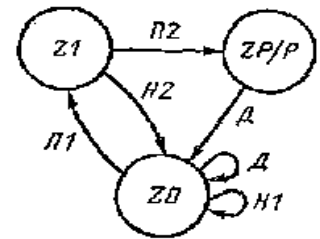
1-й этап формализации —

формирование списка входных сигналов X и выходных Y , которые будут использоваться при разработке схемы. В данном случае формально на вход воздействует сигнал открытой двери D и 10 кнопок. Однако можно ожидать, что избирательно автомат будет реагировать лишь на пять входных ситуаций: сигнал правильной первой цифры $П1$ (цифра 1), сигнал правильной второй цифры $П2$ (цифра 3), сигнал неправильной первой цифры $Н1$ (любая цифра, кроме 1, в том числе и 3), сигнал неправильной второй цифры $Н2$ (любая цифра, кроме 3, в том числе и 1) и сигнал D .

Для упрощения дальнейшей работы рационально весь набор ситуаций X , фактически воздействующих на автомат, сформировать заранее на вспомогательных логических схемах, как показано на рис. Аналогично формируется массив выходных сигналов Y , который в примере представлен единственным выходным уровнем P .

2-й этап формализации — определение требуемого задачей числа состояний автомата.

В данном случае решение очевидно: три состояния, $Z0$ — начальное состояние, состояние ожидания первого правильного сигнала; $Z1$ — первый правильный сигнал получен, ожидание второго; ZP — оба сигнала получены, вырабатывается единичный уровень выходного сигнала P на усилитель соленоида замка. Если бы отпирающая последовательность состояла из трех нажатий кнопок, то добавилось бы еще одно состояние. Можно, если этого требуют условия работы, ввести еще состояние — тревога, в которое автомат переходит в случае набора неверной цифры и при этом выдает сигнал куда следует. Кстати, как автомат должен поступать в случае неправильного набора — это первый вопрос, ответа на который не было в первичной формулировке задания, который возникает при проектировании и требует выяснения у заказчика. Допустим, ответ таков: при неправильном наборе схема сбрасывается в начальное состояние.



При проектировании сложных автоматов число состояний становится очевидным далеко не сразу, а лишь в процессе изучения задания, по мере более глубокого понимания алгоритма автомата и уяснения разнообразия его реакций на одинаковые входные сигналы.

3-й этап формализации — построение предварительного, упрощенного графа или таблицы автомата (синтез абстрактного автомата). В примере с замком лучше начинать с графа, поскольку при числе состояний в пределах 10—15 граф обычно нагляднее таблицы. В графовом представлении функционирования автомата состояния автомата отображаются вершинами графа, а возможные переходы автомата из одного состояния в другое — дугами графа. Направления переходов обозначены стрелками. Дуги помечены входными сигналами-условиями, при которых эти условия выполняются.

Выходные сигналы, если они связаны только с определенным состоянием, изображаются внутри кружка состояния, порождающего этот сигнал. Если кроме состояния выходной сигнал зависит еще и от входных сигналов, то он изображается

выходящей из соответствующего состояния стрелкой, помеченной порождающим входным сигналом, В рассматриваемом примере разумно полагать, что выходной сигнал P однозначно связан с состоянием автомата ZP , т.е. $P=1$, тогда, когда автомат находится в состоянии ZP независимо от действующих при этом входных сигналов. Граф есть символическое отражение словесного описания автомата. Пока на вход не поступит сигнал $\Pi 1$, автомат находится в начальном состоянии $Z0$. Сигнал $\Pi 1$ переводит его в состояние $Z1$, от куда он при поступлении неверного сигнала $H2$ вернется в $Z0$, а при поступлении второго правильного сигнала $\Pi 2$ перейдет в состояние ZP разрешения открыть замок. Сигнал D , возникающий при открывании двери, сбрасывает автомат в $Z0$, подготавливая его к следующему циклу работы. Никакие другие сигналы не выводят автомат из состояния ZP : пусть такова воля заказчика *В принципе можно задать и иные условия.*

Для строгого указания на тот факт, что, несмотря на поступление некоторых сигналов, автомат, тем не менее, не изменяет своего состояния, на графе переходов около этих состояний чертят петли, помеченные соответствующими сигналами. Примеры двух таких петель показаны у вершины $Z0$.

Строгое начертание графа потребовало бы еще одной петли $\Pi 1$ у вершины $Z1$ и трех петель — $\Pi 1$, $\Pi 2$ и $H2$ — у вершины ZP (или хотя бы двух — $\Pi 2$ и $H2$, поскольку сигнал $\Pi 1$ входит в состав $H2$). Петли сильно загромождают чертеж, и их, как правило, не изображают, а лишь подразумевают, полагая, что при поступлении любого сигнала, кроме тех, которыми помечены стрелки перехода из данного состояния, автомат своего состояния не изменит. Несколько позже будет показано, что замалчивание существования петель оправдано и с позиций построения логической схемы.

Входы	Состояния		
	$Z0$	$Z1$	ZP
$\Pi 1$	$Z1$	$Z1$	ZP
$\Pi 2$	$Z0$	ZP	ZP
$H2$	$Z0$	$Z0$	ZP
D	$Z0$	$Z0$	$Z0$
Выход	—	—	P

Наряду с графовым используют табличное описание автомата, общепринятая форма которого приведена в табл. Строки таблицы отображают входные сигналы, столбцы - состояния. На пересечении строки и столбца записывается новое

состояние, в которое автомат перейдет из состояния данного столбца под воздействием входного сигнала данной строки. В последней строке приведены те выходные сигналы, которые однозначно связаны с состоянием. Если выходной сигнал связан еще и со входным, то он изображается на соответствующем пересечении и отделяется от нового состояния косой чертой, как знаменатель дроби. Если в таблице заполнены все клетки, то говорят, что автомат *определен полностью*. В задании на *недоопределенный* автомат в некоторых клетках могут стоять кресты — символы безразличия. В процессе построения логической схемы разработчик доопределяет таблицу исходя из соображений качества схемы — аналогично доопределению таблицы логической функции.

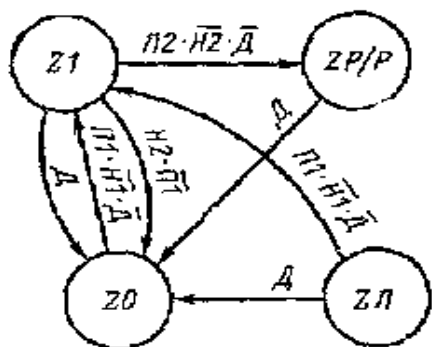
Граф и таблица — два взаимно эквивалентных способа формального описания поведения автомата. Каждый из них задает реакцию проектируемого автомата на каждый входной сигнал уже однозначно, без опасности разночтения. Чтобы граф был строго эквивалентен таблице, на нем нужно изобразить в явном виде все петли.

Описанный процесс построения графа или таблицы называют этапом *абстрактного синтеза* автомата или этапом *синтеза абстрактного автомата*. *Абстрактный автомат* — это еще не устройство и даже не схема. Это лишь математическая модель, это алгоритм функционирования некоторого преобразователя кодовых последовательностей. Если входы в табл. (или на графе) отождествить с буквами некоторого входного алфавита, а выходы — с буквами выходного алфавита (в случае замка выходной

алфавит состоит лишь из двух букв: P и $ПРОБЕЛ$), то абстрактный автомат - это закон преобразования цепочек букв (т. е. слов} входного алфавита в цепочки букв (слова) выходного алфавита. Примерно такая постановка задачи и послужила в свое время исходным толчком к созданию теории автоматов. Создание абстрактного автомата — это первый осязаемый результат на пути построения схемы автомата. Дальнейшие шаги разработки автомата, приблизительно до уровня получения его функциональной схемы, принято называть этапом *структурного синтеза*.

4-й этап формализации — построение полного графа (полной таблицы) автомата. Формальное описание абстрактного автомата, полученное на предыдущем этапе, как правило, оказывается недостаточно полным для того, чтобы непосредственно по нему можно было вести синтез логической схемы. Обычно требуется еще рассмотреть и формализовать следующие не учтенные абстрактной моделью моменты: вопрос о *неиспользуемых состояниях* автомата, *начальная установка* его после включения питания, возможность *совпадения входных сигналов* во времени,

Неиспользуемые (запрещенные) состояния. О выборе кодов регистра Z для представления состояний автомата речь будет впереди, а пока пусть известно, что три рабочих состояния автомата хранятся в двухзарядном триггерном регистре. При этом четвертая из возможных комбинаций состояний триггеров не используется, в цепочке переходов автомата не участвует, и о ней, вроде бы, можно и не говорить. Однако если из-за воздействия помехи а регистре ошибочно возникнет это четвертое состояние, автомат в нем «зависнет», и никакие внешние сигналы уже не смогут вернуть его в рабочий цикл. Говорят, что в графе переходов существует *изолированная вершина*. В грамотно спроектированной схеме изолированных вершин или циклов из нескольких таких вершин быть не должно. От неиспользуемых вершин нужно сделать «стоки» в рабочий цикл графа, и это должно быть отражено на самом графе.



В рассматриваемом примере на новом чертеже полного графа лишняя вершина ZL уже показана явно и подключена к остальной части графа дугами A и $H1 \cdot \overline{H1} \cdot \overline{D}$. Это решение допустимое и неплохое: если в автомате вдруг возникнет состояние ZL , то или первое же открывание двери вернет автомат в начальное состояние, или нажатие первой верной кнопки переведет его в $Z1$. Почему вместо простого условия перехода $H1$ использовано более сложное — $H1 \cdot \overline{H1} \cdot \overline{D}$ будет ясно немного позже.

Начальная установка. При включении питания триггеры регистра, а вместе с ними и весь автомат оказываются в случайном состоянии. При некоторых применениях допустимо устанавливать автоматы в начальное состояние $Z0$, выполнив первый холостой, установочный прогон до конца цикла. Если эффект, вызываемый таким неполным циклом, начавшимся с произвольной промежуточной точки, недопустим, то автомат снабжают цепями *начальной установки (начального сброса)*. На графе переходов для этого вводят дуги переходов от каждой вершины к нулевой, которые помечают еще одним специально введенным входным сигналом — сигналом начального сброса.

В примере с кодовым замком начальная установка путем холостого прогона чревата неприятностями. Если автомат при включении питания с равной вероятностью устанавливается в любое состояние, в том числе и в ZP , то злоумышленник, выворачивая в обеденный перерыв на несколько секунд пробки и пробуя затем дверь, будет проникать в любую защищенную комнату в среднем за че-

тыре попытки. Для начальной установки замка специальной цепи сброса можно не вводить. Роль сигнала сброса может выполнить сигнал D : дуги D от вершин ZP и ZJI к ZO уже есть, и если добавить еще одну дугу D от ZI к ZO , то цель, которой служит автомат, не пострадает: хотя открываемая неожиданно выходящим из комнаты дверь и прервет интеллектуальный процесс набора кода другим лицом, желающим попасть в комнату, но в результате дверь все равно окажется открытой и желание набравшего — выполненным. Отметим, что решение использовать тракт D еще и таким образом требовало знания реальных условий работы автомата, а не вытекало непосредственно из текста задания. Дуга от ZI к ZO показана на последнем рис. Вообще в цифровой аппаратуре цепи начального сброса автоматов делают как с ручным запуском, вводя специальную кнопку *СБРОС*, так с автоматическим, от включения питания. После включения питания RC -цепочка с постоянной времени порядка секунды через триггер Шмитта запускает одновибратор OB , импульс которого через элемент ИЛИ поступает в тракт D и устанавливает автомат в начальное состояние. *И еще один метод: в электрических схемах БИС триггеры иногда выполняют немного асимметричными, так, чтобы в процессе появления напряжения питания они сами устанавливались в 0.*

Совпадение входных сигналов во времени. Если автомат создается не для обработки текстов, где буквы поступают строго по одной, а как управляющее устройство, которое должно реагировать на изменения окружающей обстановки, то возможности совпадения во времени входных сигналов, по крайней мере от независимых источников, исключить нельзя. Однако в таблице переходов и выходов абстрактного автомата реакция на возможные совпадения сигналов никак не отражена, поэтому схема, построенная по такой таблице, может реагировать на совпадения входных сигналов любым, самым неожиданным образом. Чтобы этого не произошло, разработчик должен предусмотреть адекватные реакции автомата на эти случаи и отразить их в таблице или на графе переходов.

Формальным приемом учета возможности совпадения сигналов, который часто рекомендуют к применению на этапе структурного синтеза автомата, является переход к новому набору абстрактных входных сигналов, число которых равно числу всех комбинаций реальных сигналов. В случае замка пять входных сигналов с помощью дешифратора можно превратить в 2^5 , т. е. в 32 абстрактных сигнала, которые уже будут появляться строго по одному. Для описания автомата в этом случае потребуется таблица в 32 строки, столько же дуг (включая петли) будет и у графа. Соответственно усложнится процедура синтеза схемы.

Этот прием оправдан и даже необходим, когда разработчик приписывает определенный смысл именно комбинациям входных сигналов, как, например, при кодировании A -разрядным кодом команды 2^k запускаемых этой командой различных операций. Однако в устройствах автоматики намного чаще определенный смысл приписывается не комбинации сигналов, а именно отдельному сигналу (перегрев, ошибка, дверь открыта и т. п.) вне связи с другими. Поэтому наиболее адекватной реакцией при совпадении сигналов часто оказывается реакция на наиболее важный из них как на единственный в данный момент, а все остальные сигналы игнорируются. Если приоритет не определен четко самим содержанием задачи, он может быть назначен произвольно. Ранжирование входных сигналов по приоритетам снимает с разработчика большой труд по осознанию требуемой реакции автомата на каждую комбинацию различных входных сигналов и существенно упрощает как таблицу, так и схему автомата. При этом ранжировать требуется не все сигналы, а лишь те, которые порождают конфликтные ситуации, т. е. при своем совпадении вызывают на графе переходов ситуации неопределенности или генерации. Для каждого состояния автомата конфликтующие группы сигналов в общем различны, и начинать работу нужно с выявления этих групп.

В рассматриваемом примере в состоянии $Z0$ конфликтуют две пары сигналов: $П1$ с $Д$ и $П1$ с $Н1$. При одновременном появлении любой пары сигнал $П1$ будет стремиться перевести автомат в состояние $Z1$, а $Д$ или $Н1$ — оставить в $Z0$. По смыслу работы замка сигналам $Д$ и $Н1$ нужно назначить более высокий приоритет, чем сигналу $П1$, т. е. разрешить переход в $Z1$ по сигналу $П1$ лишь при условии, что дверь закрыта ($\overline{Д}$) и кроме правильной кнопки не нажата больше никакая другая, а тем более сразу все — ладонью ($\overline{Н1}$). Для этого переход $Z0 \rightarrow Z1$, который на исходном рис. был помечен безусловным сигналом $П1$, на уточненном графе, помечается уже конъюнкцией $П1 \cap \overline{Н1} \cap \overline{Д}$. Снова для формализации задания потребовалось знание условий работы объекта.

При рассмотрении вершин $Z0$ и $Z1$ конфликтующей парой оказываются сигналы $П1$ и $Н2$. Оба они порождаются нажатием кнопки 1, но один из них вызывает переход $Z0 \rightarrow Z1$, а второй — обратно, $Z1 \rightarrow Z0$. Все время, пока нажата кнопка 1, схема будет находиться в режиме генерации. Чтобы состояние $Z1$ сделать устойчивым, необходимо сигналу $П1$ в этом состоянии дать более высокий приоритет, чем сигналу $Н2$. Для этого дуга $Z1 \rightarrow Z0$ помечается функцией $\overline{Н2} \cdot П1$, и тогда вернуться в $Z0$ от неверно нажатой кнопки автомат сможет лишь после отпущения кнопки $П1$.

В состоянии $Z1$ сигналы $Д$ и $\overline{Н2} \cdot П1$ не конфликтуют, поскольку заняты одним общим делом, и совпадение их во времени не вызовет осложнений. Поэтому здесь введения приоритетов не требуется. В неполном графе переход $Z1 \rightarrow ZP$ выполняется по условию $П2$. Однако в этом случае нажимать кнопку $П2$ и переходить в ZP можно, не отпустив предварительно кнопки $П1$. Нажатая кнопка $П1$ не дает автомату сброситься из $Z1$ в $Z0$ даже при одновременном нажатии любой неверной кнопки. Значит, фактически существенным оказывается знание лишь первой цифры. После ее нажатия злоумышленник, не отпуская первой кнопки, нажимает ладонью все остальные. Неверная кнопка замка не сбросит, а верная его откроет. Чтобы вторая цифра была столь же значимой, что и первая, в последнем графе переход $Z1 \rightarrow ZP$ делается по условию $П2 > Н2$, т. е. переход в ZP по нажатию $П2$ осуществится, только если $П1$ отпущена и вообще не нажата никакая другая кнопка, кроме $П2$,

№ ц/п	Состояние в данный момент	Входы автомата					Следующее состояние	Выход Р
		П1	Н1	П2	Н2	Д		
1	Z0	1	0	X	X	0	Z1	0
2	Z1	X	X	X	X	1	Z0	0
3	Z1	0	X	X	1	X	Z0	0
4	Z1	X	X	1	0	0	ZP	0
5	ZP	X	X	X	X	1	Z0	1
6	ZP	X	X	X	X	1	Z0	0
7	ZP	1	0	X	X	0	Z1	0

В $Z1$ конфликт возникает между нажатием кнопки $П2$ (сигнал $П2 \cap \overline{Н2}$) и сигналом $Д$. Конфликт ликвидируется, если одному из сигналов, например $Д$, назначить более высокий приоритет. Это отражено в условии перехода $Z1 \rightarrow ZP$: $П2 \cap \overline{Н2} \cap \overline{Д}$.

Введение приоритетов позволило ликвидировать все возможные сбои при совпадении входных сигналов во времени, не увеличив при этом числа дуг графа. Последний граф считается окончательным. Построенный граф — не обязательно самый правильный или самый удачный. Цель его — проиллюстрировать сам процесс формализации задания. Характерно, что для построения графа понадобился ряд сведений о реальных условиях работы автомата, которые не были изложены в первоначальном задании и, вообще говоря, требовали согласования с заказчиком. Построенный граф представлен в табличной форме. Поскольку входные сигналы теперь уже не абстрактные, таблица не получается столь изящной, как предыдущая табл. Построением полной таблицы автомата, охватывающей все его возможные (в том числе и нежелательные) состояния и все комбинации входных сигналов, способные повлиять на работу, и заканчивается этап формализации задания.

В табл. X обозначает, что как следующее состояние автомата, так и его выходной сигнал, относящиеся к данной строке, не должны зависеть от значения входного сигнала того столбца, в котором стоит этот символ.

Регистр состояний автомата

К вопросам разработки регистровой части автомата принято относить выбор числа триггеров регистра и способа кодирования состояния. Минимально возможное число триггеров равно ближайшему сверху целому от двоичного логарифма числа состояний. Максимальное число триггеров равно числу состояний, при этом каждое состояние кодируется единицей в одном из триггеров. В среднем с ростом числа триггеров в пределах указанного диапазона уменьшается число логических элементов, требующихся для дешифрации состояний. При максимальном числе триггеров дешифрации вообще не требуется. Поэтому высказываемое иногда утверждение, что чем меньше триггеров, тем автомат экономичнее по аппаратным затратам, нельзя считать безусловно верным. Кроме соображений экономичности КС увеличению числа триггеров сверх необходимого минимума способствует использование микросхем регистров с числом разрядов, кратным четырем.

При кодировании состояний, если нет других соображений, вершины графа переходов нумеруются в произвольном порядке и номера кодируются по двоичной системе. Однако объем комбинационной схемы можно уменьшить, если для вершин, связанных большим числом дуг переходов, подбирать коды, отличающиеся возможно меньшим числом разрядов. Тогда для обеспечения каждого перехода придется завести сигналы на входы меньшего числа триггеров. Иногда для уменьшения числа переключаемых триггеров увеличивают общее их число. Алгоритма нахождения наиболее экономичного решения, кроме полного перебора вариантов, не существует. Разработчики используют метод проб и ошибок и свой опыт.

Распространено мнение, что в автомате не будет гонок и он может быть сделан асинхронным, без системы синхронизации, если его состояния закодировать так, чтобы при любом переходе изменялось значение только одного разряда. Это так называемое противогоночное кодирование состояний автомата. В случае разветвленного графа такое кодирование требует введения дополнительных триггеров в регистр. В основе мнения о действенности противогоночного кодирования лежат добротные работы, но большой давности, основанные на характерном для того времени соотношении задержек, когда задержка переключения триггера существенно превышала задержку КС. В этих условиях основной причиной возникновения гонок действительно было неодинаковое время переключения триггеров. Поэтому если при любом переходе автомата в регистре состояний переключался всего только один триггер, гонки не возникало.

В современной элементной базе задержка триггера соизмерима с задержкой одного логического элемента, поэтому задержка комбинационной схемы средней сложности превышает задержку триггера. Еще хуже ситуация внутри БИС, где существенную роль играют еще и трудноконтролируемые задержки в связях. В этих условиях основное число помех гоночного типа зарождается в недрах КС автомата, и противогоночное кодирование его состояний здесь бессильно. Тем не менее поток рекомендаций применять этот метод до сих пор не иссякает. Построение КС, в которых гонки невозможны, весьма трудоемко и реально выполнимо лишь для очень простых схем. Кроме того, такие схемы требуют больше оборудования, и эта разница уже при достаточно простых схемах превышает затраты оборудования на генератор синхросигналов.

Учитывая все сказанное, число триггеров регистра состояний кодового замка принимается равным двум. Кодирование его состояний: $Z0 = 00$; $Z1 = 01$; $ZP = 11$; $ZL = 10$. Коды для $Z0$ и $Z1$ и $Z1$ и ZP выбраны отличающимися лишь одним разрядом (соседнее кодирование), чтобы уменьшить, как уже говорилось, объем логических схем; именно эти состояния связаны наибольшим числом дуг. Поскольку устройство легко разместится на одной плате, т. е. расфазировки активных фронтов можно не опасаться, система синхронизации выбирается однофазной, а триггеры — соответственно синхронные JK -триггеры. Генератором синхросигналов может служить мультивибратор

на логических микросхемах. Входные сигналы привязываются к синхросерии с помощью схем синхронизаторов. Дребезга контактов схема не боится: таблица переходов автомата построена так, что снятие, а затем повторная подача входного сигнала не нарушают работы автомата,

Комбинационная схема автомата

После задания способа кодирования состояний разработчик имеет уже всю необходимую информацию для построения КС автомата. Для кодового замка эта информация представлена в более подробной по сравнению с предыдущей табл. - уже с расшифровкой текущего и следующего состояний автомата в виде значений $Q2, Q1$ — выходов триггеров регистра и $J2, K2, J1, K1$ — их входов. КС имеет семь входов, и полная таблица ее значений имела бы 128 строк. Сократить число строк до шести удалось благодаря двум доста-

точно универсальным приемам.

Во-первых, конфликтующим входным сигналам были присвоены приоритеты. Чем выше приоритетность некоторого сигнала, тем больше в таблице на строке, где этот сигнал проявляется, крестов (X). а каждый крест,

№ п/п	Входы КС								Имя следующего состояния	Выходы КС				
	Состояние Z		Входные сигналы X							Код перехода в следующее состояние, F				Выход Y
	код	код	P1	H1	J2	H2	D	J2		K2	J1	K1	P	
1	Все	X	X	X	X	X	X	1	Z0	0	1	0	1	0
2	Z0	0	0	1	0	X	X	0	Z1	0	0	1	0	0
3	Z1	0	1	0	X	X	1	X	Z0	0	0	0	1	0
4	Z1	0	1	X	X	1	0	0	ZP	1	0	0	0	0
5	ZP	1	1	X	X	X	X	X	ZP	0	0	0	0	1
6	Z1	1	0	1	0	X	X	0	Z1	0	1	1	0	0

относящийся к какой-то переменной, отображает сразу две строки — и для значения этой переменной 0, и для значения 1. Одна строка с четырьмя крестами заменяет сразу 16 строк. Первая строка табл. вобрала в себя все дуги сброса автомата в Z0 сигналом D (строки 2, 5 и 6 предыдущей табл., в том числе и из состояния ZP, поэтому в строке ZP (строка 5) достаточно было отобразить лишь значение выходного сигнала P.

Во-вторых, в качестве триггеров использованы JK-триггеры, которые при $J=K=0$ хранят свое состояние. Следовательно, все строки с такими комбинациями входов КС, которые не вызывают смены состояния автомата, т. е. переключения триггеров, можно в таблицу не вносить: все равно на выходе КС они дают нули и поэтому в состав СДНФ не войдут. Другими словами, строя таблицу, достаточно заносить в нее только метки на дугах смены состояний автомата, и не требуется заносить метки петель, указывающих на сохранение состояния автомата. Поэтому петли на графе можно с самого начала не чертить. Если бы вместо JK- использовать D-триггеры, то число строк таблицы увеличилось бы существенно, поскольку для простого поддержания выхода D-триггера в 1 на его D-вход должна обязательно каждый такт подаваться 1.

Полученная таблица позволяет построить логические функции для всех пяти выходов КС известными методами.

$$J2 = \overline{Q2} \cdot Q1 \cdot P2 \cdot \overline{H2} \cdot \overline{D};$$

$$K2 = D \vee Q2 \cdot \overline{Q1} \cdot P1 \cdot \overline{H1};$$

$$J1 = \overline{Q2} \cdot \overline{Q1} \cdot P1 \cdot \overline{H1} \cdot \overline{D} \vee Q2 \cdot \overline{Q1} \cdot P1 \cdot \overline{H1} \cdot \overline{D} = \overline{Q1} \cdot P1 \cdot \overline{H1} \cdot \overline{D};$$

$$K1 = D \vee \overline{Q2} \cdot \overline{Q1} \cdot \overline{P1} \cdot H2;$$

$$P = Q2 \cdot Q1.$$

Комбинационная схема автомата — это заданный таблицей кодовый преобразователь некоторого входного кода в выходной, и при такой трактовке для его построения годятся все рассмотренные ранее методы. В частности, КС можно строить на ПЛМ или ПЗУ, если это даст экономию затрат. Существуют микросхемы ПМЛ, в состав которых входят программируемая логическая матрица и небольшой триггерный регистр. Запрограммировав такую микросхему, можно весь автомат реализовать на одном корпусе.

Контрольные вопросы

1. Назначение регистров.
2. Область применения преобразователей последовательного кода в параллельный.
3. Принцип реализации операции «И» в регистре.
4. Принцип реализации операции деления на 2^n в регистре.
5. Назначение распределителей тактов.
6. Различия между сумматором и суммирующим счетчиком.
7. Различия при синтезе синхронных и асинхронных двоичных счетчиков.
8. Принципы построения двоичных счетчиков.
9. Принцип построения делителей частоты на счетчиках.
10. Назначение шинных формирователей.
11. Принципы построения автоматов с конечным числом состояний.

Вспомогательные элементы цифровых устройств

К числу вспомогательных узлов цифровых устройств относят функциональные узлы, не выполняющие логических операций или запоминания данных, но необходимые для функционирования этих устройств - элементы задержки, формирования и генерации импульсных сигналов.

Генераторы и формирователи прямоугольных импульсов

Генераторы прямоугольных сигналов относят к классу релаксационных – их выходные сигналы чередуют медленные изменения со скачкообразными. В импульсной технике применяется несколько видов релаксационных генераторов, однако в цифровой электронике наибольшее распространение получили мультивибраторы. Их название объясняется широким спектром частот выходного сигнала. Существует ряд ИС мультивибраторов, требующих лишь подключения вреязадающих цепочек, но простейшие мультивибраторы очень просто реализуются на базе логических элементов.

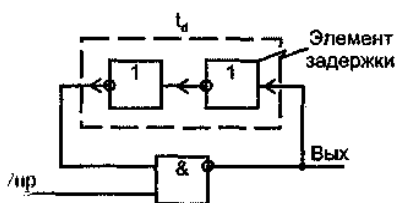
Различают *автоколебательные* и *ждущие (одновибраторы)* мультивибраторы.

Автоколебательный мультивибратор (АГ) характеризуется отсутствием устойчивых состояний. Как и у всякого логического устройства его выходной сигнал может принимать одно из двух значений – лог.0 или лог.1. Однако эти состояния оказываются квазиустойчивыми, т.е. АГ может находиться в каждом из этих состояний лишь некоторое время, определяемое его параметрами, а непрерывный процесс смены этих состояний и представляет собой генерацию прямоугольных импульсов.

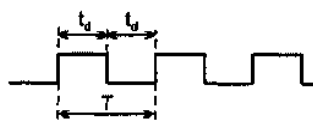
Ждущий мультивибратор (одновибратор ОВ) в одном из состояний выхода может находиться сколь угодно долго (устойчивое состояние). Другое же состояние оказывается квазиустойчивым – под воздействием внешнего сигнала ОВ может перейти в это состояние, но через некоторое время все равно вернется в свое устойчивое состояние. При этом длительность пребывания ОВ в этом состоянии определяется параметрами данного ждущего мультивибратора. Таким образом, однократное внешнее воздействие вызывает появление единственного выходного импульса с длительностью, являющейся основной характеристикой ОВ.

Автоколебательные мультивибраторы

Простейшие генераторы импульсных последовательностей реализуются на эле-



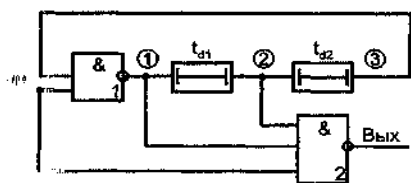
а



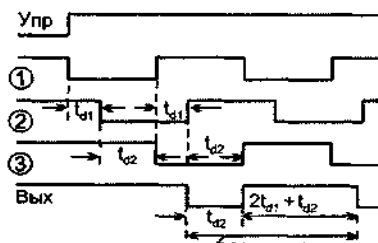
б

ментах задержки и логических. Один из таких вариантов приведен на рис. а. При нулевом значении сигнала управления Упр на выходе элемента И-НЕ имеется логическая единица, которая через обратную

связь с задержкой на t_d передается на верхний вход элемента. Таким образом, в исходном состоянии верхний вход элемента И-НЕ находится в состоянии логической единицы. Изменение управляющего сигнала является командой для начала работы генератора. Появление единицы на нижнем входе Упр элемента И-НЕ дает совпадение единиц на обоих входах, что переводит выход схемы в нулевое состояние. Это состояние длится в течение интервала t_d , т. к. после него нуль с выхода схемы по обратной связи пройдет на верхний вход элемента и поставит его в единичное состояние, которое также сохранится на время t_d , после чего изменится из-за воздействия по цепи обратной связи. Следовательно, схема будет генерировать симметричные импульсы с длительностями импульса и паузы, равными t_d (рис. б).



и



г

При необходимости получения импульсной длительности, отличной от меандра, для формирования длительностей импульса и паузы могут быть использованы различные элементы

задержки: отрезки обычных или специальных коаксиальных кабелей, цепочки логических элементов, RC – цепочки.

Применение логических элементов для задержки сигналов основано на инерционности логических элементов, так что последовательное соединение нескольких элементов позволяет суммировать их задержки. Это очень удобный способ получения задержки сигнала, что объясняется конструктивными особенностями логических элементов и обычно наличием свободных элементов в уже используемых корпусах ИС. Однако величина задержки оказывается случайной величиной из-за разброса параметров логических элементов и не может быть использована в устройствах, где предъявляются значения к точности частоты следования импульсов.

Лучшими в этом отношении характеристиками обладают элементы задержки, реализованные на RC – цепочках. Вариант построения АГ с подобной реализацией задержки приведен на рис. а. Генератор, показанный на рис. а, можно построить на любых инвертирующих элементах.

Резистор R выполняет две функции: смещает рабочую точку элемента 1 на крутой участок передаточной характеристики, обеспечивая мягкое самовозбуждение, и вместе с конденсатором C служит времязадающим элементом. Элемент 3 служит развязывающим буфером.

Регенеративный процесс переключения схемы начинается, когда напряжение на входе элемента 1 достигает порога его срабатывания, показанного на

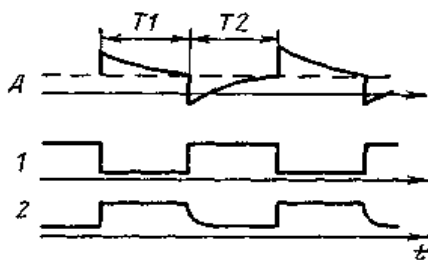
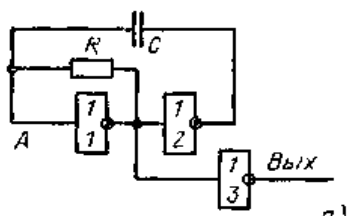
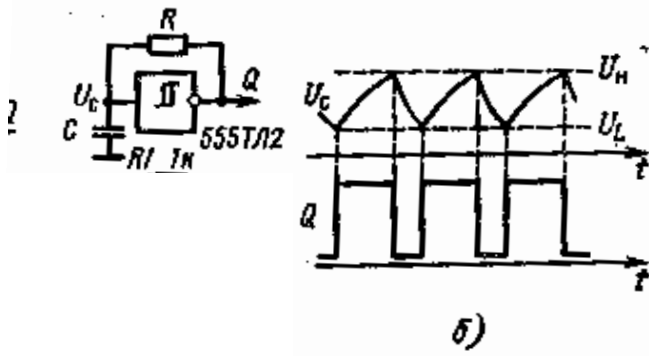


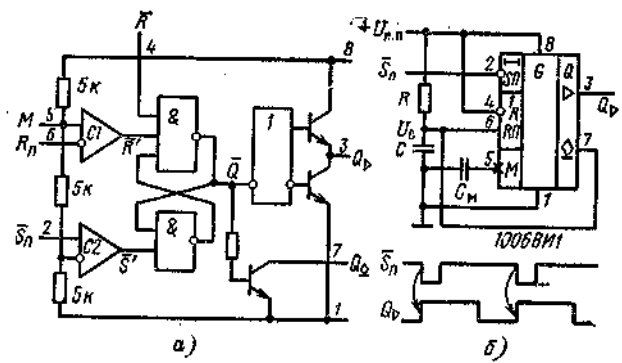
рис. б штриховой линией. Процесс заканчивается переключением элементов 1 и 2 в противоположные состояния и передачей фронта напряжения с выхода элемента 2 через конденсатор C на вход эле-

мента 1, после чего потенциал точки A снова будет стремиться к пороговому уровню с постоянной времени RC . Для ТТЛ-схем процесс несколько асимметричен: выше порогового уровня входной ток элемента 1 не превышает десятков микроампер, а ниже порога может быть более 1 мА, т.е. дает заметную добавку к зарядному току резистора. Поэтому обычно полупериод $T2$ процентов на 10 короче $T1$. Ориентировочная длительность каждого полупериода — около $2RC$. При необходимости получения существенно асимметричной (или, наоборот, более симметричной) временной диаграммы резистор R можно шунтировать цепочкой из включенных последовательно резистора и кремниевого диода. Нестабильность частоты подобных генераторов имеет значение 10—20 %.



Очень удобным логическим устройством для реализации АГ является триггер Шмитта. Работу этого генератора поясняют временные диаграммы. Заряд и разряд конденсатора C может происходить только до напряжений, определяемых пороговыми уровнями U_H и U_L . При достижении напряжением U_C одного из этих уровней происходит скачкообразное изменение выходного сигнала генератора Q . Частота генерируемых импульсов $f = 0,7/RC$. Стабильность частоты генераторов на триггерах Шмитта определяется стабильностью пороговых напряжений U_H и U_L и достаточна для многих практических применений.

Более высокую стабильность частоты за счет большей стабильности порогов срабатывания обеспечивает прецизионный таймер **1006ВИ1**. Структурная схема этого таймера показана на рис. а и состоит из двух аналоговых компараторов $C1$ и $C2$, асинхронного потенциального R-S-триггера, мощного выходного каскада и выходного каскада с открытым коллектором. Опорные напряжения компараторов U_L и U_H задаются делителем с высокой точностью: $U_L = 1/3 U_{un}$ и $U_H = 2/3 U_{un}$. Мощный выходной каскад обеспечивает вы-

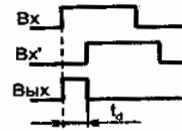
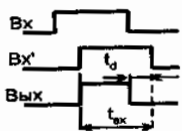
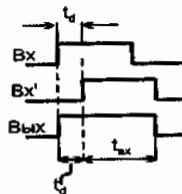
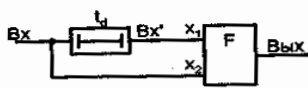


ходной ток $I_{вых} = 200$ мА.

Для обеспечения высокой стабильности частоты применяют кварцевые генераторы, простой вариант которого показан на рис. Резисторы $R1$ и $R2$ — той же величины, что и в предыдущей схеме, выводят рабочие точки на крутые участки характеристик. Конденсатор C позволяет логическим элементам иметь различные по постоянному смещению уровни крутых участков. Его импеданс на рабочей частоте должен быть существенно меньше сопротивления $R2$.

Элемент 3 — буфер. Частота выходного меандра жестко задается частотой кварцевого резонатора, и приведенная схема хорошо работает в диапазоне от 100 кГц до 10 МГц. Частота следования импульсов определяется резонансной частотой выбранного кварцевого резонатора, но существуют ИС синтезатора частот, позволяющего перестройку частоты в широких пределах с сохранением ее стабильности.

Формирователи импульсов по длительности



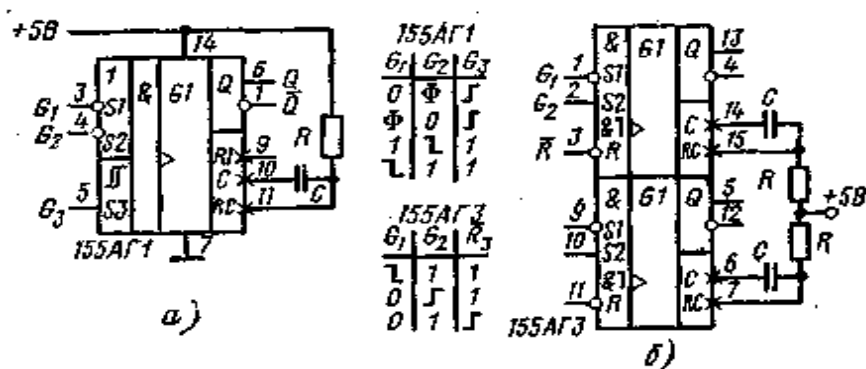
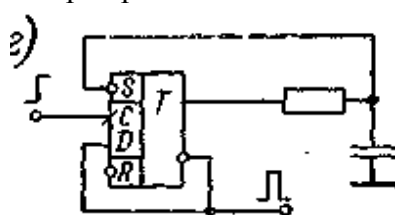
К задачам формирования импульсов по длительности относятся расширение, сужение и стандартизация их длительности.

Эти операции реализуются схемой (рис. а). Если конкретизировать функцию F , считая ее дизъюнкцией, то, как видно из временных диаграмм на рис. б, схема будет расширять входной импульс на интервал, равный времени задержки t_d . Если понимать под функцией F конъюнкцию и

рассмотреть временные диаграммы (рис. 6), то можно видеть, что схема дает сужение входного импульса на величину t_d . Если $F = x_1 x_2$, то будет выполнена стандартизация длительности импульса. Выходной импульс будет иметь длительность t_d , независимо от длительности входного (при $t_{ex} > t_d$). Это иллюстрируется временными диаграммами рис. 2.

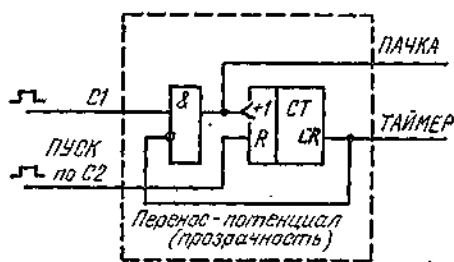
Недостатком приведенной схемы является требование, накладываемое на входной сигнал $t_{ex} > t_d$. Устранить это ограничение позволяют формирователи на основе ждущих мультивибраторов.

Наибольшее распространение имеют одновибраторы с RC-цепочками, не накладывающие требований на длительность запускаящего импульса. Подобный одновибратор просто реализуется на базе D-триггера.



В семействах ИС цифровых устройств ждущие мультивибраторы представлены в виде функционально завершенных устройств, требующих лишь подключения времязадающих элементов. Часто они снабжаются вход-

ной логикой, позволяющей управлять режимом их работы.



При повышенных требованиях к точности формируемой длительности импульсов используются цифровые формирователи. На рис. показана формирователя, использующего подсчет заданного количества импульсов, стабильность частоты следования которых и определяет точность работы

Контрольные вопросы

1. Области применения автогенераторных мультивибраторов в цифровых устройствах.
2. Различие между ждущим мультивибратором и T-триггером.
3. Каким образом в автогенераторных мультивибраторах возможно осуществление регулирования частоты колебаний.
4. Почему в ждущих мультивибраторах оказывается возможным устранить ограничения на соотношения между длительностями входного и выходного сигналов.
5. Существуют ли ограничения на период следования импульсов запуска ждущего мультивибратора.
6. Чему равна скважность импульсов симметричного мультивибратора.
7. Чем обусловлены ограничения на величину R резистора времязадающей цепи в схемах мультивибраторов на логических элементах.
8. В чем преимущество цифровых методов формирования импульсов большой длительности по сравнению с аналоговыми.

Некоторые примеры реализации цифровых устройств

При проектировании цифровых устройств на ИС средней степени интеграции обычные методы синтеза (таблицы истинности, диаграммы Вейча и др.) оказываются непригодными, хотя и могут быть эффективно использованы при решении вспомога-

тельных задач. Основными предпосылками для проектирования цифровых устройств на ИС средней степени интеграции являются глубокое понимание решаемой задачи и детальное знание законов функционирования значительного числа ИС, выраженных в аналитической форме. В этом случае достаточно сложные цифровые устройства могут быть спроектированы *эвристическим методом*, который делает невидимыми все рассуждения, приведшие к получению правильно функционирующей схемы.

Важной особенностью цифровой электроники является многовариантность возможных решений. При этом выделение оптимального решения определяется системой оценок, выбранных в качестве критерия. Причем, в зависимости от назначения устройства эта система оценок видоизменяется, т.е. для какой-то задачи наиболее существенными могут оказаться характеристики, не играющие особой роли для другой. В этом случае оптимальное решение оказывается выбираемым из множества возможных и чем богаче выбор, тем ближе выбранное решение к оптимальному. Приводимые примеры являются вариантами решения типовых задач, но призваны отразить многообразие методов построения устройств цифровой электроники.

Последовательный сумматор

В качестве примера показана схема четырехразрядного двоичного полного сумматора чисел, заданных в последовательном коде. Операция сложения двух чисел в двоичном коде осуществляется с учетом переноса из младшего разряда и выдачей переноса в старший разряд.

Сумматор последовательного действия содержит один одноразрядный сумматор, D-триггер и три сдвиговых регистра. Два сдвиговых регистра используются для хранения слагаемых, а в третий записывается результат суммирования. Регистры обычно выполняются отдельно от сумматора и являются отдельными микросхемами.

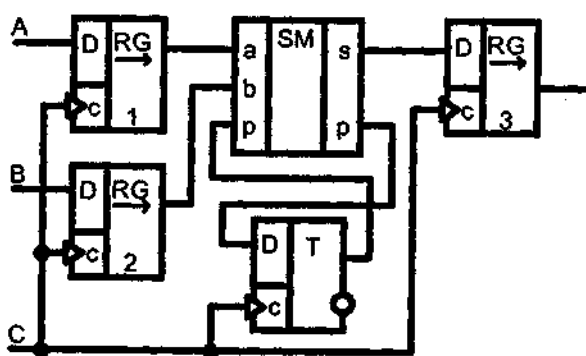
Под действием каждого синхроимпульса на входы a , b одноразрядного сумматора поступают, разряды кодовых слов, начиная с младшего. На вход p подается сигнал переноса с выхода D-триггера. Результаты суммирования в последовательной форме записываются в третий регистр, новый сигнал переноса записывается в D-триггер по приходу следующего синхроимпульса.

Для выполнения суммирования на тактовый вход необходимо подать n синхронизирующих импульсов по числу разрядов суммируемых двоичных чисел. После каждого синхроимпульса выполняется счет одного разряда и формирование переноса в следующий разряд.

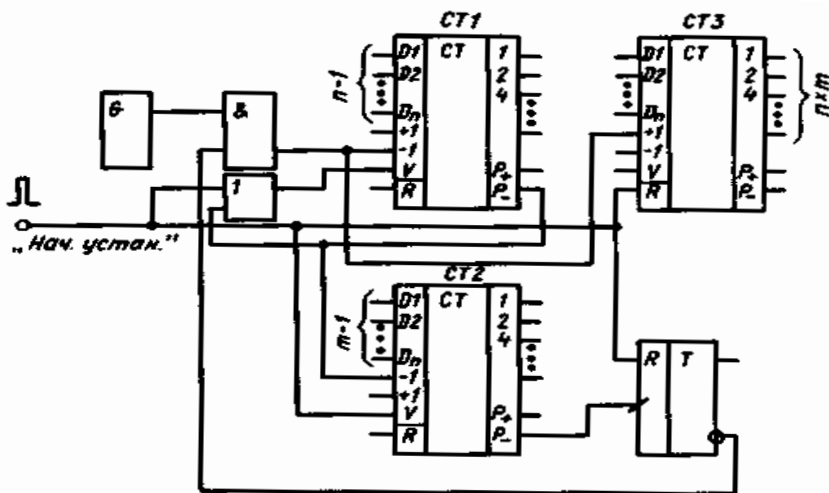
Достоинство последовательных сумматоров — малые аппаратные затраты, однако операция выполняется над числами, заданными в последовательном коде, что и определяет быстродействие сумматора.

Устройство умножения двоичных чисел

Существует большое многообразие устройств для умножения двоичных чисел. Одним из главных критериев обычно является быстродействие устройства, что приводит к реализации умножителя в виде комбинационной схемы. Однако, если к быстродействию

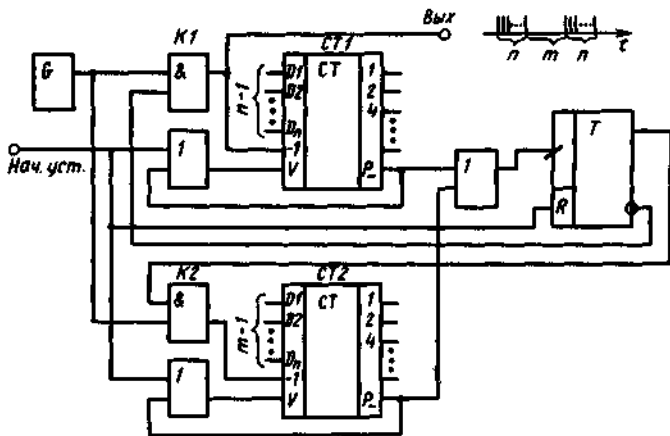


устройства не предъявляется особых требований, а главным требованием является простота схемы, то операция умножения может быть реализована приведенной схемой. В данном случае отвлекаются от правил умножения, а процесс рассматривают как результат необходимого числа суммирований одного из сомножителей.



обнуляется счетчик CT_3 , где накапливается произведение двоичных кодов сомножителей, в счетчики CT_1 и CT_2 заносятся коды и в состояние 0 по прямому выходу устанавливается триггер. После этого логической 1 с инверсного выхода он разрешает поступление импульсов генератора G на суммирующий вход счетчика CT_3 и на вычитающий вход счетчика CT_1 . Когда на этом входе пройдут n импульсов, в счетчик CT_3 запишется число n , а на выходе заема P_- появится импульс, который считает единицу из счетчика CT_2 и даст разрешение на повторное занесение числа $n-1$ в счетчик CT_1 . С этого момента начнется повторение описанного процесса: в счетчик CT_3 еще раз будет занесено число n , из счетчика CT_2 — считана следующая единица, а в счетчик CT_1 вновь записано число $n-1$. Таким образом, каждой единице, считанной из счетчика CT_2 , соответствует число n , заносимое в счетчик CT_3 . Этот процесс происходит до тех пор, пока на вычитающем входе CT_2 не пройдет m импульсов. После этого импульс с выхода P_- переключит триггер, который нулем с инверсного выхода прервет поступление импульсов генератора к счетчикам через конъюнктор. К этому моменту в счетчик CT_3 будет записано число nm .

Формирователь пачек импульсов



Рассматриваемое устройство вырабатывает выдавать пачки из n импульсов с интервалом между пачками, равным m периодов следования импульсов.

На рис.приведена схема одного из возможных вариантов такого устройства. При этом принимается во внимание, что на выход заема P_- каждого счетчика с вычитающего входа проходит положительный импульс вслед за осуществившим обнуление счетчика.

На входы $D_1—D_n$ каждого счетчика постоянно поданы двоичные коды чисел $n-1$ и $m-1$. С поступлением на входы V импульса «Начальная установка» эти числа заносятся в счетчики, а триггер устанавливается в 0. При этом логическая 1 с его инверсного выхода разрешает импульсам генератора G проходить через конъюнктор K_1 на вычитающий вход счетчика CT_1 и одновременно на выход устройства.

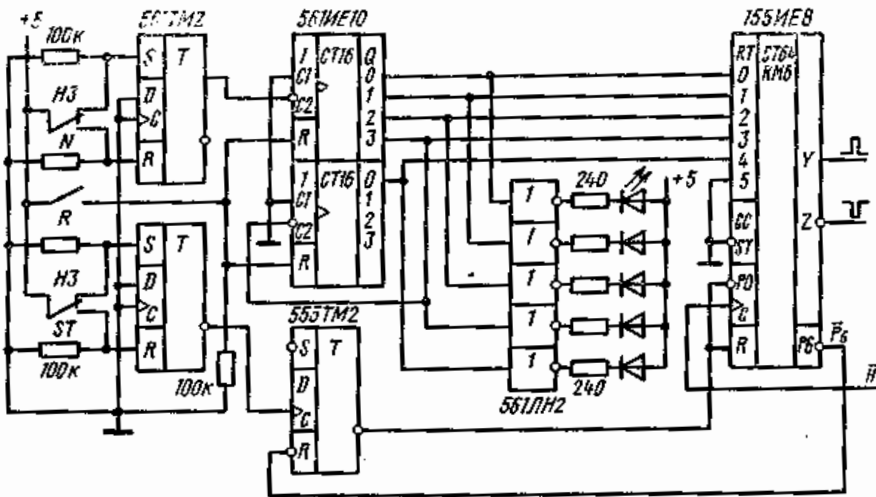
После того как на вычитающий вход поступит $n-1$ импульсов, счетчик CT_1 обнулится. Следующий импульс выделится на выходе заема P_- , через дизъюнктор переключит триггер в 1 и на вход V поступит разрешение на повторный ввод числа $n-1$ в счетчик CT_1 .

Теперь триггер с инверсного выхода блокирует конъюнктор K_1 (на выходе устройства начинается пауза), а с прямого выхода дает разрешение импульсам генератора проходить на вычитающий вход счетчика CT_2 . После поступления $m-1$ импульсов счетчик CT_2 обнулится. Следующий импульс выделится на

выходе P переключит триггер в 0, а на вход V поступит разрешение на повторный ввод числа $m-1$ в счетчик СТ2. С этого момента начнется формирование следующей пачки импульсов.

Вариант устройства с реализацией в конкретной элементной базе приведен на следующем рисунке.

Для решения поставленной задачи наиболее подходит ИС 155ИЕ8. С помощью кнопки «N» и 5-разрядного счетчика (561ИЕ10) устанавливается необходимый двоичный код числа N , который индицируется с помощью пяти светодиодов или шкального индикатора. При нажатии кнопки «ST» (Start) подается

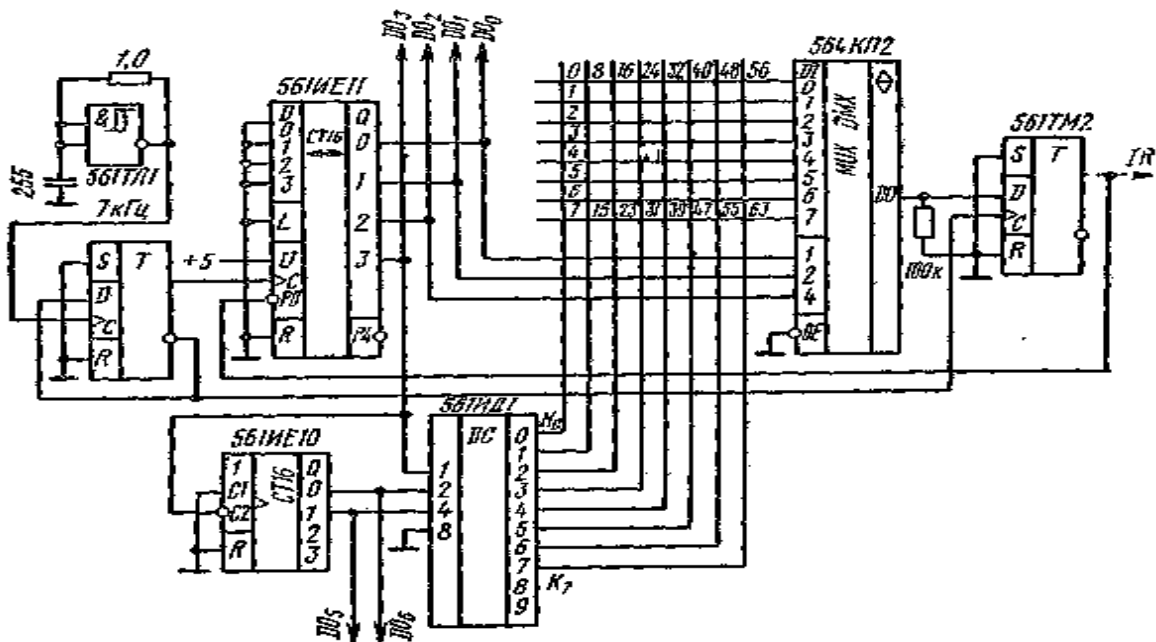


разрешение на счет ИС 155ИЕ8. Сигнал переноса $Pv = 0$ останавливает счет. Кнопкой «R» производится сброс 5-разрядного счетчика в нулевое состояние.

Матричные шифраторы клавиатуры

Пусть клавиатура содержит 64 клавиши, каждой из которых приписан свой 6-разрядный двоичный код. При нажатии любой клавиши на шину данных микро-ЭВМ должен выдаваться двоичный код клавиши, а на шину управления—сигнал запроса прерывания IR (Interrupt Request). Явно напрашивающееся решение задачи построения шифратора 64×6 с помощью приоритетных шифраторов 155ИВ1 нельзя считать приемлемым, так как требуется девять ИС, потребляющих большую мощность, для уменьшения которой следует использовать КМОП ИС.

Наиболее эффективным решением задачи является матричное сканирование клавиатуры, которое легко выполнить с помощью 6-разрядного счетчика, дешифратора 3×8 и мультиплексора $8 \rightarrow 1$. При замыкании любой клавиши $j=0 \dots 63$ выходной



сигнал ИС 564КП2' примет значение 1 в момент времени, когда 6-разрядный двоич-

ный счетчик (561 ИЕ11, 561ИЕ10) установится в состояние 1, что приведет к получению значения сигнала $IR=1$, который блокирует дальнейшие изменения состояния счетчика. Пока клавиша нажата на шину данных, будет выдаваться число $j=DO_5...DO_0$ и сигнал $JR=1$. При отпуске клавиши сигнал IR установится в 0 и счетчик продолжит счет. Дешифратор 561ИД1 периодически вырабатывает с частотой $f=7000/128 \Rightarrow 55$ Гц на восьми выходах сигналы $K_i=1$, производящие сканирование клавиатуры.

Чем ниже частота задающего генератора (триггер Шмитта 561ТЛ1), тем надежнее исключается «дребезг» контактов клавиш. При частоте генератора 7 кГц «дребезг» надежно устраняется даже для невысококачественной клавиатуры. Если требуется шифратор клавиатуры $16*4$, то из схемы следует исключить ИС 561ИЕ10 и 561ИД1, включив вместо них дешифратор $1*2$ (НЕ) состояния триггера Q_3 счетчика 561ИЕ11. В этом случае остаются только клавиши с номерами $j=0...15$.

Литература

1. Угрюмов Е.П. Цифровая схемотехника. СПб: «БХВ – Петербург», 2002г. – 528с.
2. Схемотехника электронных систем. Аналоговые и импульсные устройства: Учебник./ В.И. Бойко, А.Н. Гуржий, В.Я. Жуйков и др.. -СПб.: БХВ-Петербург, 2004.- 496 с
3. Схемотехника электронных систем. Цифровые устройства : Учебник/ В.И. Бойко, А.Н. Гуржий, В.Я. Жуйков и др.. -СПб.: БХВ-Петербург, 2004.-506 с.
4. Ушаков В.Н. Основы аналоговой и импульсной техники. -М.: РадиоСофт, 2004.
5. Новиков, Юрий Витальевич. Основы цифровой схемотехники. Базовые элементы и схемы. Методы проектирования. -М.: Мир, 2001.-379с.
6. Бабич Н.П., Жуков И.А. Компьютерная схемотехника. Киев: МК-Пресс, - 2004г., - 576с.
7. Точчи Р.Дж., Уидмер Н.С. Цифровые системы. Теория и практика. Пер. с англ. М: Издательский дом «Вильямс» - 2004г., 1024с.