МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Казанский государственный
энергетический университет

Л.В. АХМЕТВАЛЕЕВА

Утверждено
учебным управлением КГЭУ
в качестве учебного пособия
для студентов

ЦИФРОВЫЕ УСТРОЙСТВА

УЧЕБНОЕ ПОСОБИЕ
по курсу

Электронные промышленные устройства

Казань 2002

УДК 621.37
А 95

Цифровые устройства: Учеб. пособие / Ахметвалеева Л.В. Казань: Казан. гос. энерг. ун-т, 2002.

Представлены теоретические основы синтеза и анализа цифровых устройств, элементы алгебры логики, принципы и методы построения цифровых узлов. Приведены структура и варианты схем цифровых устройств комбинационного и последовательного типа.

Предназначено для студентов специальностей 200400 направления 654100 “Электроника и микроэлектроника” дневного, вечернего и заочного обучения.

Рецензент

Д-р физ.-мат. наук, ведущий научный сотрудник КФТИ КНЦ РАН
Х.Г. Богданова

Рекомендовано секцией РИС факультета электронной техники и автоматизации

Председатель секции В.Л. Матухин

© Казанский государственный энергетический университет, 2002 г.

ВВЕДЕНИЕ

Интенсивное развитие разнообразных цифровых систем (ЭВМ, микропроцессорные системы, цифровая связь и т.д.) и в связи с этим широкое распространение цифровых методов формирования и обработки электрических сигналов настоятельно требует изучения основ цифровой электроники.

При этом необходимо принять во внимание требования обязательного изучения традиционных понятий, методов анализа и синтеза функциональных узлов цифровых устройств. Без знания теоретических основ невозможно проектирование оптимального по своим характеристикам современного цифрового устройства. Также необходимо уделить внимание современной элементной базе, обеспечивающей реализацию методов проектирования. Перспективы отечественной микроэлектроники предсказать сложно. Отсутствие отечественных микросхем современного уровня компенсируются сейчас доступностью зарубежной элементной базы, поэтому изучение цифровых устройств имеет практическое значение.

Данное пособие предназначено для базовой подготовки студентов специальности 200400 «Промышленная электроника» по направлению «Электроника и микроэлектроника», для изучения раздела курса «Электронные промышленные устройства», посвященного вопросам цифровой электроники.

В пособии рассмотрены теоретические основы цифровой электроники, вопросы анализа и синтеза цифровых устройств, особенности комбинационных схем, структура и функционирование основных последовательных схем: триггеров, регистров, счетчиков. Пособие дает ответы на множество вопросов, возникающих при разработке цифровых схем, при выборе типовых ИМС для их реализации.

В силу ограниченности объема в пособии рассмотрены не все компоненты цифровой электроники. Такие разделы, как запоминающие устройства, ЦАП и АЦП, устройства программируемой логики и ряд других тем предполагается рассмотреть в последующих изданиях.

При написании настоящего пособия использованы опыт работы кафедр “Промышленная электроника” МЭИ (ТУ) и КГЭУ, а также учебники и учебные пособия других авторов по рассматриваемой тематике.

Наше пособие может быть полезно и для студентов других специальностей, изучающих цифровую электронику.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ЭЛЕКТРОНИКИ

1.1. Кодирование цифровой информации

Понятие кодирования. Обозначение различной цифровой информации соответствующими символами называется кодированием, а состав символов для данной информации – алфавитом этого кода. Запись числа в двоичной системе счисления называется его натуральным двоичным кодом. Это название применяется для того, чтобы отличить один код от других двоичных кодов, в которых соответствие между числом и рядом нулей и единиц может быть иным. Из множества возможных интерпретаций на практике используются только те, которые нужны для получения специальных средств кода в каждом конкретном случае. Код выбирают таким образом, чтобы дальнейшее преобразование информации выполнялось возможно проще: чтобы легче было реализовать арифметические операции, подсчет сигналов, расшифровку кода (декодирование), обнаружение и исправление ошибок, приведение к виду, удобному для использования внешними устройствами, и т.д.

Известно большое количество кодов, систематизация и классификация которых из-за их многочисленных признаков является довольно затруднительной. Мы рассмотрим лишь некоторые разновидности кодов. Все коды можно разделить на две самостоятельные группы. К первой относятся коды, в которых используются все возможные комбинации – неизбыточные коды. В литературе их называют еще простыми и первичными. Ко второй относятся коды, в которых используется лишь часть всех возможных комбинаций. Такие коды называются избыточными. Оставшаяся часть комбинаций не несет информации о числах, а служит для обнаружения ошибок, возникающих при передаче или обработке сообщений.

Те и другие коды, в свою очередь, подразделяются на равномерные и неравномерные. Равномерные коды – это коды, все комбинации которых содержат постоянное количество разрядов. Неравномерные коды содержат кодовые комбинации с различным числом разрядов. Типичным представителем неравномерных кодов является используемая в телеграфии азбука Морзе, в которой каждому символу соответствует разное общее количество тире и точек. Ввиду ограниченного применения неравномерных кодов мы их рассматривать не будем.

Все коды, служащие для представления чисел, можно разбить на две группы – взвешенные и невзвешенные коды. Взвешенные коды – такие коды, в которых каждому разряду присваивается определенный вес. Например, натуральный двоичный код является взвешенным кодом, поскольку каждому разряду соответствует вес, равный степени числа 2. Невзвешенные коды – это коды,

разрядам которых нельзя присвоить веса, а соответствие кода и числа устанавливается каким-либо иным способом.

Двоично-десятичные коды. Для выражения каждой десятичной цифры (0, 1, 2, …, 9) нужны четыре двоичные цифры, то есть четыре бита, которые позволяют получить $2^4 = 16$ комбинаций. Так как из 16 комбинаций 6 являются избыточными,то общее число всех возможных кодов из 4 бит равно $16!/6! \approx 2{,}9 \cdot 10^{10}$.

Однако на практике используются только такие коды, которые обеспечивают самую рациональную и простую обработку цифровой информации. Чаще всего используются взвешенные двоично-десятичные коды, позволяющие достаточно просто переводить десятичные цифры $(X)_{10}$ в двоичные $(X)_2$ по формуле:

$$(X)_{10} = a_3 \cdot x_3 + a_2 \cdot x_2 + a_1 \cdot x_1 + a_0 \cdot x_0, \qquad (1.1)$$

где символы a_3, ..., a_0 являются постоянными весовыми коэффициентами соответствующего кода, а символы x_3, ..., x_0 – двоичные цифры 1 или 0. Существуют различные взвешенные двоично-десятичные коды a_3, ..., a_0, например: 8421, 7421, 7321, 6421, 6321, 6311, 6221, 5421, 5311, 5221, 5211, 4421, 4321, 4311, 4221 и 3331. В коде 8421 $a_3 = 8$, $a_2 = 4$, $a_1 = 2$, $a_0 = 1$. Кодовые комбинации, соответствующие тем или иным символам, в различных кодах легко определяются из выражения (1.1). Например,

$$(7)_{10} = 8 \cdot 0 + 4 \cdot 1 + 2 \cdot 1 + 1 \cdot 1 = (0111)_{8421};$$
$$(7)_{10} = 5 \cdot 1 + 2 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 = (1100)_{5211};$$
$$(7)_{10} = 5 \cdot 1 + 2 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 = (1011)_{5211}.$$

Как видно, представление числа в коде 5211 не является однозначным, т.е. одному и тому же числу соответствуют две кодовых комбинации. Неоднозначность имеют все взвешенные двоично-десятичные коды, за исключением кода 8421. При использовании неоднозначных кодов необходимо заранее обусловливать, какая комбинация соответствует данной цифре. В табл. 1.1 приведены кодовые комбинации, соответствующие представлению десятичных цифр в различных двоично-десятичных кодах.

Коды I – IV, VI, VII являются взвешенными, коды V и VIII – невзвешенными.

Коды VI – VIII называются самодополняющимися. Дополняющие свойства этих кодов заключаются в том, что двум десятичным цифрам, сумма которых равна 9, соответствуют две дополняющие комбинации. Например, числу 3

в коде 2421 соответствует комбинация 0011, а числу 6, являющемуся дополнением числа 3 до 9 – комбинация 1100. В самодополняющихся кодах дополнение возникает простой заменой единицы нулем и нуля единицей. Самодополняющимся кодом является также невзвешенный код с избытком 3. Он получается добавлением числа $(3)_{10} = (0011)_2$ к числу в коде 8421 и удобен для арифметических операций, так как при сложении довольно просто можно определить необходимость коррекции результата. А так как это самодополняющийся код, то он может быть использован при вычитании, основанном на сложении в обратном и дополнительном кодах.

Таблица 1.1. Некоторые двоично-десятичные коды

$(X)_{10}$	Несамодополняющиеся коды					Самодополняющиеся коды		
	I	II	III	IV	V	VI	VII	VIII
	8421	6121	5211	4221	невзвешенный	2421	4221	с избытком 3
0	0000	0000	0000	0000	0000	0000	0000	0011
1	0001	0001	0001	0001	0001	0001	0001	0100
2	0010	0010	0011	0010	0010	0010	0010	0101
3	0011	0011	0101	0011	0011	0011	0101	0110
4	0100	0100	0111	0110	0110	0100	1000	0111
5	0101	0101	1000	0111	0111	1011	0111	1000
6	0110	1000	1001	1010	1110	1100	1010	1001
7	0111	1001	1011	1011	1111	1101	1101	1010
8	1000	1010	1101	1110	1100	1110	1110	1011
9	1001	1011	1111	1111	1101	1111	1111	1100

Кроме представленных двоично-десятичных кодов на практике применяются и другие как четырехразрядные (четырехэлементные) коды, так и коды с большим числом разрядов.

Код Грея. Этот код относится к циклическим кодам, характеризующимся тем, что комбинации двоичных цифр, отображающие числа, которые отличаются друг от друга на единицу, разнятся только в одном разряде. Это свойство кода Грея весьма ценно для преобразователей линейного перемещения или кругового движения какого-либо устройства в цифровой код, так как погрешность не превышает единицы младшего разряда.

Код Грея легко получается из натурального двоичного кода путем суммирования по модулю два цифр соседних разрядов. Если число $x_3\ x_2\ x_1\ x_0$ – представление числа X в двоичной системе счисления (в натуральном двоичном коде), а число $y_3\ y_2\ y_1\ y_0$ – представление того же числа в коде Грея, то между ними существуют следующие соотношения:

$$\begin{aligned} y_3 &= x_3; & x_3 &= y_3; \\ y_2 &= x_3 \oplus x_2; & x_2 &= y_3 \oplus y_2; \\ y_1 &= x_2 \oplus x_1; & x_1 &= y_3 \oplus y_2 \oplus y_1; \\ y_0 &= x_1 \oplus x_0; & x_0 &= y_3 \oplus y_2 \oplus y_1 \oplus y_0. \end{aligned} \qquad (1.2)$$

Код Грея, полученный с помощью выражений (1.2), представлен в табл. 1.2. Этот код является неизбыточным кодом.

Таблица 1.2. Код Грея

Число	Натуральный двоичный код				Код Грея			
S	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Существуют также избыточные коды Грея, например, двоично-десятичные. Можно сформировать коды Грея для произвольного числа разрядов m. В этом случае существуют следующие формы перехода от числа в натуральном двоичном коде $x_{m-1}, x_{m-2}, \ldots, x_0$ в код Грея:

$$y_{m-1} = x_{m-1};$$
$$y_{m-2} = x_{m-1} \oplus x_{m-2};$$
$$\ldots$$
$$y_0 = x_1 \oplus x_0.$$

Переход от m-разрядного кода Грея к натуральному двоичному коду осуществляется по формулам

$$x_{m-1} = y_{m-1};$$
$$x_{m-2} = y_{m-1} \oplus y_{m-2};$$
$$\ldots$$
$$x_0 = y_{m-1} \oplus y_{m-2} \oplus \ldots \oplus y_0.$$

Код Джонсона. Последовательность чисел в этом коде моделируется односторонним последовательным заполнением его разрядов вначале единицами, а затем нулями (табл. 1.3). Код Джонсона легко формируется с помощью регистров сдвига и легко дешифруется.

Весьма интересным кодом является код “1 из m”, представленный для случая $m = 8$ в табл. 1.3. Этот код характерен тем, что в любой кодовой комбинации присутствует только одна единица, что позволяет легко находить ошибки в случае искажения кода, и не требуется его дешифрации. Этот код, как и код Джонсона, является избыточным, требующим для своего изображения больше разрядов, чем соответствующие неизбыточные коды.

Таблица 1.3. Код “1 из m”

S	Код Джонсона	Код “1 из 8”
0	0 0 0 0	0 0 0 0 0 0 0 1
1	0 0 0 1	0 0 0 0 0 0 1 0
2	0 0 1 1	0 0 0 0 0 1 0 0
3	0 1 1 1	0 0 0 0 1 0 0 0
4	1 1 1 1	0 0 0 1 0 0 0 0
5	1 1 1 0	0 0 1 0 0 0 0 0
6	1 1 0 0	0 1 0 0 0 0 0 0
7	1 0 0 0	1 0 0 0 0 0 0 0

1.2. Классификация цифровых устройств

Цифровые устройства (ЦУ) обрабатывают информацию, представленную цифровыми кодами. ЦУ преобразует совокупность цифровых входных сигналов *X* в выходные сигналы *Y*. Для формирования цифровых выходных сигналов используются ЦУ двух классов:

- ЦУ, выходные сигналы *Y* которых в некоторый момент времени *t* зависит только от совокупности (комбинации) сигналов *X*, присутствующих на их входах в тот же момент времени *t*, и не зависят от входных сигналов, поступающих в предшествующие моменты времени. Такие ЦУ принято называть комбинационными (КЦУ).

- ЦУ, выходные сигналы *Y* которых в момент *t* определяются не только комбинациями входных сигналов *X*, воздействующих в тот же момент времени *t*, но и сигналами, поступающими на входы в предшествующие моменты времени. В составе таких ЦУ обязательно присутствуют элементы памяти, внутреннее состояние которых отражает предысторию поступления последовательности входных сигналов. Такие ЦУ называются последовательностными (ПЦУ). Говорят, что ПЦУ обладают памятью. Элемент памяти помимо входных и выходных сигналов характеризуется состоянием, которое может изменяться в дискретные моменты времени под воздействием сигналов на входе. ПЦУ называются также цифровыми автоматами, конечными автоматами или автоматами с памятью.

ЦУ строятся из некоторых элементарных компонентов – элементов. При этом элементы соединяются по определенным правилам. Характер элементов и их соединение определяют функционирование устройства в целом. Логическое проектирование (синтез) ЦУ на всех этапах имеет дело с различными моделями. В настоящее время существует достаточное количество таких моделей, однако основой, служащей базой для всех остальных, является модель конечного автомата, предложенная Клини.

ЦУ, описанное моделью конечного автомата, имеет конечное число входов и выходов, на каждом из которых может быть два уровня физического воздействия и конечное число функциональных элементов. При переходе от ЦУ к модели удобно абстрагироваться от физической природы входов и выходов. Для этого каждому входу сопоставляется двоичная входная переменная (принимает значение 0 или 1) в соответствии с двумя фактическими воздействиями на входе, а каждому выходу сопоставляется выходная переменная. В самом общем виде цифровой (конечный) автомат (ЦА) представляется в виде многополюсного «черного ящика» с n входами и m выходами, преобразующего информацию (рис. 1.1.).

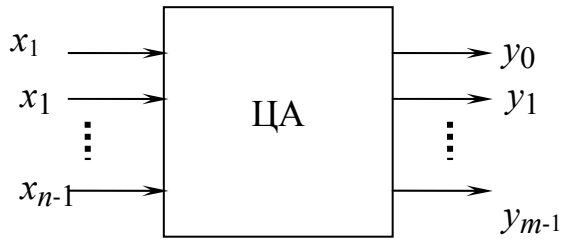


Рис. 1.1. Общий вид цифрового автомата

ЦА состоит из отдельных элементов и имеет операционную структуру. Для описания ЦА применяется структурная теория автоматов. Символы x_i (i=0,1,2,…,n-1) образуют алфавит входных переменных, а y_j (j=0,1,2,…,m-1) алфавит выходных переменных. Различают неинверсные переменные x_i или y_i и инверсные $\overline{x}_i$ или y_i. Последние имеют всегда значение, обратное к значению неинверсных переменных. Индексы i и j используются для нумерации переменных.

Конечную упорядоченную последовательность букв называют словами в заданном алфавите. Если представить в виде упорядоченной последовательности значения символов на всех входах ЦА, то получаем входное слово в виде вектора

$$X = (x_0, x_1, \ldots, x_{n-1}).$$

Аналогично для всех выходов ЦА выходное слово представляется в виде

$$Y = (y_0, y_1, \ldots, y_{m-1}).$$

Каждое слово – набор на входе или выходе автомата, отличается от другого значения хотя бы одной буквой. Каждая переменная поставлена в соответствии с определенным входом или выходом и таким образом определяет значение сигнала на нем. Множество наборов – слов, которые могут появляться на входах и выходах автомата, определяется числом входов и выходов алфавита переменных. Для входов автомата число входных слов – наборов равно 2^n, для выходов число выходных слов равно 2^m.

Для задания автомата необходимо указать соответствие между элементами входного и выходного алфавитов, т.е. между наборами входных и выходных сигналов с учетом автоматного времени.

$$Y(t) = f[X(t)].$$

Однако следует учесть, что в состав многих ЦА входят специальные элементы хранения информации – запоминающие элементы. Чаще всего элементы памяти имеют два состояния, отождествленные с цифрами 1 и 0. Если ЦА содержит в своем составе k элементов памяти, состояние всех элементов составляет слово $S=(s_0, s_1,\ldots, s_{k-1})$, определяющее внутреннее состояние, то общее число внутренних состояний равно $K=2^k$. Множество $S=(S_0, S_1,\ldots, S_{K-1})$ будет представлять множество внутренних состояний ЦА.

Итак, для задания автомата с точки зрения переработки информации на трех множествах входного, выходного и внутреннего алфавитов необходимо определить функциональные зависимости – соответствие между входными, внутренними и выходными символами – наборами. Кроме того, необходимо определить последовательность формирования слов внутренних состояний под воздействием входных слов с учетом внутренних состояний. Другими словами на трех множествах входных, выходных и внутренних состояний необходимо определить две функции:

1) функцию выходов (f), которая парам входное состояние – внутреннее состояние ставит в соответствие выходное состояние

$$Y(t) = f\,[X(t), S(t)]\ ;$$

2) функцию переходов (φ), которая парам входное состояние – внутреннее состояние ставит в соответствие новое внутреннее состояние

$$S(t) = \varphi\,[X(t), S(t-1)].$$

Существуют две модели ЦА, которые отличаются друг от друга функциями выходов:

- автомат Мили, функционирующий по уравнениям:

$$Y(t) = f\,[S(t), X(t)]\ ;$$

$$S(t) = \varphi\,[S(t-1), X(t)].$$

- автомат Мура, функционирующий по уравнениям:

$$Y(t) = f\,[S(t)]\ ;\ \overline{y}_i$$

$$S(t) = \varphi\,[S(t-1), X(t)].$$

В автомате Мура состояния выхода определяются только внутренними состояниями, а в автомате Мили они определяются как внутренними, так и входными состояниями.

Итак, общей математической моделью ЦА является абстрактный автомат, имеющий один вход и один выход, состоящий из:

- множества входных состояний $X = (x_0, x_1, \ldots, x_{n-1})$;
- множества выходных состояний $Y = (y_0, y_1, \ldots, y_{n-1})$;
- множества внутренних состояний $S = (s_0, s_1, \ldots, s_{k-1})$;
- функции выходов f;
- функции переходов φ.

Исходя из сказанного, можно сделать вывод о том, что существует два типа ЦА, с которыми приходится иметь дело в технике: комбинационные и последовательностные автоматы. Все автоматы должны быть построены на элементах двух типов: перерабатывающих информацию по определенным правилам и хранящих информацию.

Первые представляют собой простые комбинационные автоматы, вторые – самые простые последовательностные автоматы. Последовательностный автомат можно представить состоящим из двух частей: комбинационного автомата (КА) и элементов памяти (ЭП) (рис.1.2.).

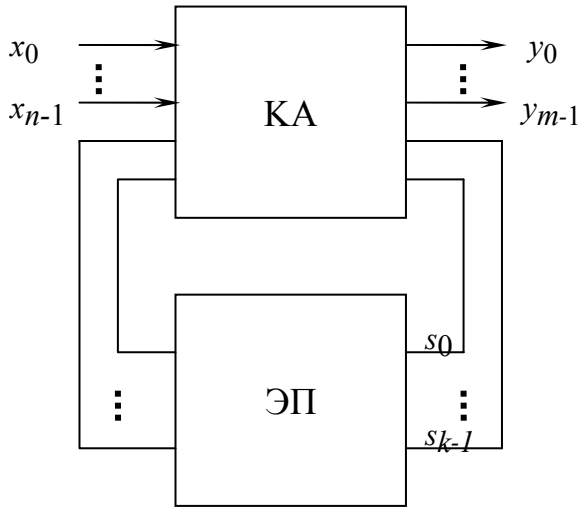


Рис. 1.2. Структура последовательностного цифрового автомата

В общем виде задача синтеза последовательностного цифрового автомата сводится к определению числа элементов памяти и синтезу комбинационной части. Всякий комбинационный ЦА с конечным числом входов $X = (x_0, x_1, \ldots, x_{n-1})$,

и выходов $Y = (y_0, y_1, \ldots, y_{m-1})$, задается путем указания функциональной связи между элементами этих двух множеств: $Y(t) = f[X(t)]$.

При задании функционирования комбинационного автомата номер такта работы опускается, так как совокупность выходных сигналов Y в n-ом такте работы однозначно определяется входными сигналами X на том же такте и его функционирование представляется системой функций:

$$\begin{cases} y_0 = f_0(x_0, x_1, \ldots, x_{n-1}); \\ y_1 = f_1(x_0, x_1, \ldots, x_{n-1}); \\ \quad \ldots \\ y_{m-1} = f_{m-1}(x_0, x_1, \ldots, x_{n-1}). \end{cases}$$

Для синтеза комбинационных автоматов применяются методы алгебры логики.

Исходной информацией к синтезу цифровых устройств обычно является описание его функционирования, определяющее характер связи входных и выходных переменных.

Цифровые устройства, независимо от сложности реализуемых ими операторов, выполняются на основе простейших (элементарных) комбинационных автоматов, называемых логическими элементами. Совокупность элементов и взаимосвязи между ними определяют структуру (структурную или логическую схему) ЦА.

Задача синтеза (логического проектирования) цифрового автомата, реализующего требуемый оператор связи входных и выходных переменных, сводится к определению оптимальной структуры автомата при заданном перечне логических элементов (заданной элементной базе). Наиболее общим критерием оптимальности является стоимость автомата, включая все затраты на его изготовление и последующую эксплуатацию. Известные в настоящее время методы логического проектирования обеспечивают, однако, лишь частичный учет этого критерия, а именно, минимизацию числа аргументов, входящих в функцию. В ходе синтеза автомата осуществляется переход от содержательного описания автомата к формализованному заданию его оператора в виде функций, таблиц, графов, матриц. Сущность задания оператора, независимо от его формы, состоит в определении перечня входных и выходных переменных автомата и установления связей между ними. Далее, исходя из упоминавшихся форм задания оператора (наиболее распространенной является табличная форма), определяется структурная схема автомата.

1.3. Основы алгебры логики

Основным математическим аппаратом, используемым для синтеза цифровых устройств, является булева алгебра (алгебра логики), названная по имени Джорджа Буля, разработавшего ее в середине XIX века.

Возможность применения булевой алгебры к задачам проектирования цифровых автоматов обусловлена аналогией понятий и категорий этой алгебры и двоичной системой счисления. В алгебре логики рассматриваются два символа: 0 и 1. Выражаясь языком формальной логики, символ 0 означает ложное высказывание, 1 – истинное высказывание. Эти простые высказывания обозначаются строчными буквами какого-либо алфавита *x, y, z, p, ….* Из элементарных высказываний образуются сложные высказывания, которые являются функциями простых логических переменных. Эти сложные высказывания называются логическими функциями, или функциями алгебры логики (ФАЛ). Другими словами, под логической (булевой или переключательной) функцией понимается двоичная переменная *y*, значение которой зависит от значений других двоичных переменных (x_{n-1}, ..., x_1, x_0), именуемых аргументами.

Булева алгебра строится на базе трех основных операций (функций) – инверсии, дизъюнкции и конъюнкции.

- Инверсия (отрицание) обозначается чертой, которая ставится над переменной. Например, отрицание переменной *x*, читаемое НЕ *x*, записывается в виде $\overline{x}$.

- Дизъюнкция (логическое сложение) обозначается знаком "$\vee$" (читается: ИЛИ).

- Конъюнкция (логическое умножение) обозначается так же, как произведение в обычной алгебре: знаками " $\cdot$ ", "$\times$" или символами "$\wedge$" или "&" (читается: И).

Постулативно полагается, что при выполнении перечисленных операций равенства имеют вид:

$$\begin{array}{lll} \overline{0} = 1; & 0 \vee 0 = 0; & 0 \cdot 0 = 0; \\ \overline{1} = 0; & 1 \vee 0 = 1; & 1 \cdot 0 = 0; \\ & 0 \vee 1 = 1; & 0 \cdot 1 = 0; \\ & 1 \vee 1 = 1; & 1 \cdot 1 = 1. \end{array} \qquad (1.3)$$

На основании этих постулатов можно вывести тождества алгебры Буля, представленные в табл. 1.4. Тождества 1, …, 14 вытекают непосредственно из равенств (1.3); последующие могут быть выведены на основании предыдущих. Например, для доказательства тождества 15 к его правой части применим 14-й, 4-й и 5-й законы:

$$(x_2 \vee x_0) \cdot (x_1 \vee x_0) = x_2 \cdot x_1 \vee x_0 \cdot x_1 \vee x_2 \cdot x_0 \vee x_0 \cdot x_0 =$$
$$= x_2 \cdot x_1 \vee x_0 \cdot x_1 \vee x_0 \cdot x_2 \vee 1 \cdot x_0.$$

Далее воспользуемся тождествами 14 и 1:

$$x_2 \cdot x_1 \vee x_0 \cdot x_1 \vee x_0 \cdot x_2 \vee 1 \cdot x_0 =$$
$$= x_2 \cdot x_1 \vee x_0 \cdot (x_1 \vee x_2 \vee 1) = x_2 \cdot x_1 \vee x_0.$$

Таким образом, тождество 15 доказано.

Таблица 1.4. Логические тождества

№ п/п	Название	Аналитическая запись
1	Закон сложения с единицей	$x \vee 1 = 1$
2	Закон сложения с нулем	$x \vee 0 = x$
3	Законы тавтологии	$x \vee x = x$
4		$x \cdot x = x$
5	Закон умножения на единицу	$x \cdot 1 = x$
6	Закон умножения на нуль	$x \cdot 0 = 0$
7	Законы дополнительности	$x \vee \overline{x} = 1$
8		$x \cdot \overline{x} = 0$
9	Закон двойного отрицания	$\overline{\overline{x}} = x$
10	Сочетательные законы	$(x_2 \vee x_1) \vee x_0 = x_2 \vee (x_1 \vee x_0)$
11		$(x_2 \cdot x_1) \cdot x_0 = x_2 \cdot (x_1 \cdot x_0)$
12	Переместительные законы	$x_1 \vee x_0 = x_0 \vee x_1$
13		$x_1 \cdot x_0 = x_0 \cdot x_1$
14	Распределительные законы	$(x_1 \vee x_0) \cdot x_2 = x_1 \cdot x_2 \vee x_0 \cdot x_2$
15		$x_2 \cdot x_1 \vee x_0 = (x_2 \vee x_0) \cdot (x_1 \vee x_0)$
16	Теорема Моргана	$\overline{x_1 \vee x_0} = \overline{x_1} \cdot \overline{x_0}$
17		$\overline{x_1 \cdot x_0} = \overline{x_1} \vee \overline{x_0}$

Весьма важными для синтеза цифровых устройств являются тождества 16 и 17, носящие название теоремы Моргана. Теорема Моргана может быть доказана с учетом того, что количество комбинаций аргументов x_1 и x_0 конечно, в данном случае равно 4. Справедливость теоремы следует из табл. 1.5, в которой приведены все возможные комбинации x_1 и x_0. Теорема верна, так как столбцы,

соответствующие выражениям $x_1 \cdot x_0$ и $x_1 \vee x_0$, полностью идентичны. То же можно сказать в отношении столбцов $x_1 \vee x_0$ и $x_1 \cdot x_0$.

Таблица 1.5. Последовательное выполнение операции Моргана

x_1	x_0	$x_1 \cdot x_0$	$\overline{x_1 \cdot x_0}$	$\overline{x_1}$	$\overline{x_0}$	$\overline{x_1} \vee \overline{x_0}$	$\overline{x_1 \vee x_0}$	$\overline{x_1} \cdot \overline{x_0}$
0	0	0	1	1	1	1	1	1
0	1	0	1	1	0	1	0	0
1	0	0	1	0	1	1	0	0
1	1	1	0	0	0	0	0	0

Теорема Моргана обобщается на произвольное количество аргументов

$$\overline{x_{n-1} \cdot x_{n-2} \cdot \ldots \cdot x_1 \cdot x_0} = \overline{x_{n-1}} \vee \overline{x_{n-2}} \vee \ldots \vee \overline{x_1} \vee \overline{x_0};$$

$$\overline{x_{n-1} \vee x_{n-2} \vee \ldots \vee x_1 \vee x_0} = \overline{x_{n-1}} \cdot \overline{x_{n-2}} \cdot \ldots \cdot \overline{x_1} \cdot \overline{x_0}.$$

Число наборов (комбинаций) аргументов логических функций. Поскольку каждый аргумент может принимать лишь одно из двух значений, то общее количество комбинаций P_n из n двоичных аргументов составит

$$P_n = 2^n.$$

Комбинации обозначаются числом S, которое является десятичным эквивалентом входного двоичного кода. При $n = 2$ число S принимает четыре значения 0, 1, 2, 3. В общем случае S должно принимать значения:

$$S = 0, 1, 2, 3, \ldots, 2^{n-1}$$

Число логических функций. Заданному числу входных аргументов n соответствует m логических функций. Функции считаются различными, если их значения отличаются, по крайней мере, для одного набора аргументов. Поскольку каждому из наборов могут соответствовать два значения y (0 или 1), то общее число различных функций n аргументов будет равно:

$$m_n = 2^{Pn} = 2^{2^2}.$$

Например, число логических функций, зависящих от одного аргумента (переменной) $y = f(x)$ равно четырем, каждая из которых определена на двух наборах. Булевы функции от одной переменной (сингулярные функции), а также их условное обозначение и название приведены в табл. 1.6.

Таблица 1.6. Логические функции от одной переменной

Функция	x		Условное обозначение	Название функций
	0	1		
$f_0(x)$	0	0	0	константа 0
$f_1(x)$	0	1	x	переменная x
$f_2(x)$	1	0	$\overline{x}$	инверсия x
$f_3(x)$	1	1	1	константа 1

Действительно, как следует из табл. 1.6, существует всего четыре сингулярные функции от одной переменной. Функции константа 0 и константа 1 названы так по той причине, что любому из двух ее наборов аргументов $x = 0$ и $x = 1$ ставится в соответствие постоянное значение функции, равное 0 для функции константа 0 и равное 1 для функции константа 1. Функция $f_1(x)$ на одном наборе $x = 0$ равна 0, а на другом наборе $x = 1$ равна 1, т.е. повторяет переменную x. Функция $f_2(x)$ принимает значение 1 на наборе $x = 0$ и значение 0 на наборе $x = 1$. Функция, выполняющая такую операцию, носит название отрицания или инверсии и является одной из основных функций булевой алгебры.

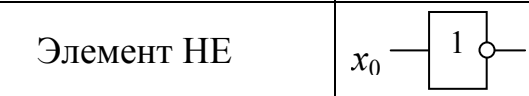
Логические функции двух аргументов и двухвходовые логические элементы. Число комбинаций входных аргументов $P_2 = 2^2 = 4$, число возможных булевых функций $m_2 = 2^4 = 16$. В табл. 1.7 сведены все возможные функции с указанием набора аргументов x_0, x_1 и приведены наименования реализующих их логических элементов. Из табл. 1.7 необходимо выделить три функции (f_1, f_7, f_{12}), которые называются основными функциями булевой алгебры. На рис. 1.3 представлены таблицы истинности и условные изображения логических элементов, выполняющих основные логические функции.

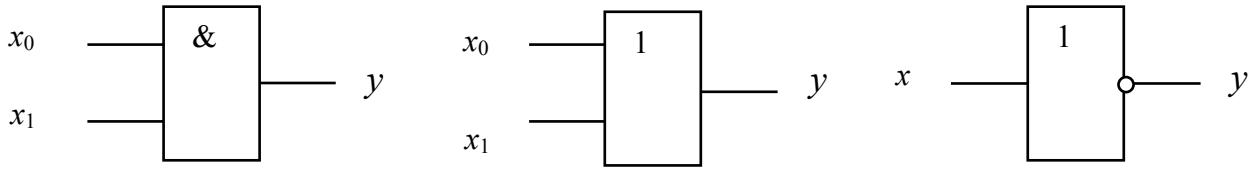
На схемах логические функции, выполняемые элементом в соответствии с ГОСТ 2.743–82, отображаются условными символами: конъюнкция – знаком &, который заносится в левый верхний угол поля элемента; дизъюнкция – цифрой единица; инверсия обозначается кружочком для соответствующего входа или выхода.

Таблица 1.7. Логические функции двух аргументов

Функция	Наборы x_0 0011 x_1 0101	СДНФ	Представление в базисе И, ИЛИ, НЕ
$f_0(x_0,x_1)$	0000	–	0
$f_1(x_0,x_1)$	0001	$x_0 \cdot x_1$	$x_0 \cdot x_1$
$f_2(x_0,x_1)$	0010	$x_0 \cdot \overline{x_1}$	$x_0 \cdot \overline{x_1}$
$f_3(x_0,x_1)$	0011	$x_0 \cdot x_1 \vee x_0 \cdot \overline{x_1}$	x_0
$f_4(x_0,x_1)$	0100	$x_0 \cdot \overline{x_1}$	$\overline{\overline{x_0} \cdot x_1}$
$f_5(x_0,x_1)$	0101	$x_0 \cdot x_1 \vee \overline{x_0} \cdot x_1$	x_1
$f_6(x_0,x_1)$	0110	$x_0 \cdot x_1 \overline{\vee} x_0 \cdot x_1$	$x_0 \cdot \overline{x_1} \vee \overline{x_0} \cdot x_1$
$f_7(x_0,x_1)$	0111	$x_0 \cdot x_1 \vee x_0 \cdot x_1 \vee x_0 \cdot x_1$	$x_0 \vee x_1$
$f_8(x_0,x_1)$	1000	$x_0 \cdot \overline{x_1}$	$\overline{x_0 \vee x_1}$
$f_9(x_0,x_1)$	1001	$x_0 \cdot x_1 \vee \overline{x_0} \cdot \overline{x_1}$	$x_0 \cdot x_1 \vee \overline{x_0} \cdot \overline{x_1}$
$f_{10}(x_0,x_1)$	1010	$x_0 \cdot \overline{x_1} \overline{\vee} x_0 \cdot x_1$	$\overline{x_1}$
$f_{11}(x_0,x_1)$	1011	$x_0 \cdot \overline{x_1} \vee x_0 \cdot x_1 \vee x_0 \cdot x_1$	$x_0 \vee \overline{x_1}$
$f_{12}(x_0,x_1)$	1100	$x_0 \cdot \overline{x_1} \overline{\vee} x_0 \cdot \overline{x_1}$	$\overline{x_0}$
$f_{13}(x_0,x_1)$	1101	$x_0 \cdot \overline{x_1} \vee x_0 \cdot \overline{x_1} \vee x_0 \cdot x_1$	$\overline{x_0} \vee x_1$
$f_{14}(x_0,x_1)$	1110	$x_0 \cdot \overline{x_1} \overline{\vee} x_0 \cdot \overline{x_1} \vee x_0 \cdot \overline{x_1}$	$\overline{x_0 \cdot x_1}$
$f_{15}(x_0,x_1)$	1111	$x_0 \cdot \overline{x_1} \vee x_0 \cdot \overline{x_1} \vee x_0 \cdot \overline{x_1} \vee x_0 \cdot x_1$	1

Продолжение табл. 1.7

Обозначение функции	Название функции	Название логического элемента	Обозначение элемента
0	Константа нуля	Генератор нуля	
$x_0 \cdot x_1,\ x_0 \wedge x_1$	Конъюнкция (логическое умножение И)	Элемент И	
$x_0 \Delta x_1$	Запрет по *x0*	Запрет	
x_0	Переменная x_0	Повторитель	
$x_1 \Delta x_0$	Запрет по x_0	Запрет	
x_1	Переменная x_0	Повторитель	
$x_0 \oplus x_1$	Сложение по модулю 2 (неравнозначность)	Исключающее ИЛИ сумматор по модулю 2	
$x_0 \vee x_1,\ x_0 + x_1$	Дизъюнкция (логическое сложение ИЛИ)	Элемент ИЛИ	
$x_0 \downarrow x_1$	Функция Вебба (стрелка Пирса)	Элемент ИЛИ-НЕ	
$x_0 \sim x_1,\ x_0 \equiv x_1$	Эквивалентность (логическая равнозначность)	Эквивалентность	
$\bar{x}_1$	Инверсия x_1 (отрицание x_0)	Элемент НЕ	
$x_1 \rightarrow x_0$	Импликация от x_1 к x_2	Импликатор	
$\bar{x}_0$	Инверсия x_0 (отрицание x_0)	Элемент НЕ	
$x_0 \rightarrow x_1$	Импликация от x_0 к x_1	Импликатор	
x_0 / x_1	Функция Шеффера (штрих Шеффера)	Элемент И-НЕ	
1	Константа единицы	Генератор единицы	



x_1	x_0	y
0	0	0
0	1	0
1	0	0
1	1	1

а)

x_1	x_0	y
0	0	0
0	1	1
1	0	1
1	1	1

б)

x	y
0	1
1	0

в)

Рис. 1.3. Условные обозначения и таблицы истинности логических элементов, реализующих основные функции: конъюнктор (а), дизъюнктор (б), инвертор (в)

Необходимо заметить, что из двухвходовых конъюкторов и дизъюнкторов можно построить трехвходовые элементы, как это показано на рис. 1.4, или элементы с еще большим числом входов.

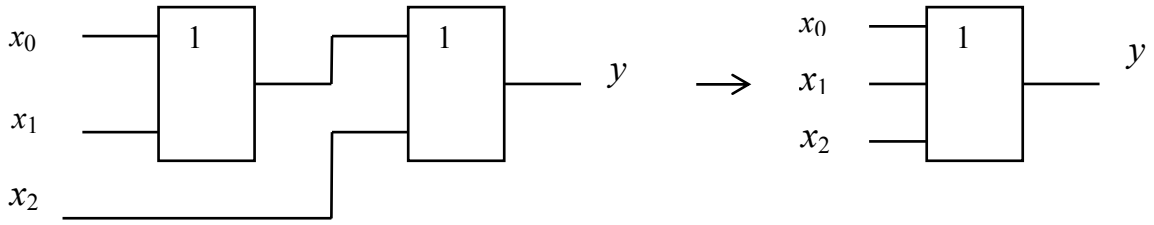


Рис. 1.4. Построение трехвходового дизъюнктора и его условное обозначение

Из табл. 1.7. следует, что, например, функция f_1 принимает значение 1 только на одном из четырех наборов переменных x_0 и x_1. Применительно к преобразованию сигналов это означает, что сигнал на выходе устройства, имеющего два входа x_0 и x_1, появляется только тогда, когда сигнал логической 1 будет одновременно присутствовать на двух входах. Элемент, реализующий эту функцию, носит название схемы И (логическое умножение) или конъюнктор. Функция f_7 принимает значение 1 на всех наборах, кроме одного $x_0 = 0$ и $x_1 = 0$. Применительно к преобразованию сигналов это означает, что сигнал на выходе устройства, имеющего два входа x_0 и x_1, появляется только тогда, когда сигнал 1 будет присутствовать хотя бы на одном из входов. Элемент реализующий эту функцию, носит название схемы ИЛИ (логическое сложение) или дизъюнктор.

Элемент, реализующий функцию f_{10} или f_{12}, носит название схемы НЕ или инвертор. Применительно к преобразованию сигналов данная функция означает, что сигнал на выходе устройства, имеющего один вход x_1 (или x_2 для f_{10}), появится только в том случае, если сигнал 1 на выходе отсутствует. При наличие сигнала 1 на входе, на выходе схемы НЕ будет действовать сигнал 0.

Рассмотрение логических функций можно продолжить и для трёх аргументов. Однако в этом нет необходимости, так как установлено, что достаточно точно иметь только несколько функций одной или двух переменных, и можно путем суперпозиции этих функций и переменных, записать произвольную функцию алгебры логики.

Сложные логические функции. Булева алгебра допускает возможность образования сложных функций, т.е. функций, аргументы которых являются функциями других двоичных аргументов. Например, если $y = y(z_1,\ z_0)$, а $z_1 = z_1(x_1, x_0)$ и $z_0 = z_0(x_3, x_2)$, очевидно, что $y = y(x_3, x_2, x_1, x_0)$. Операция замены аргументов одной функции другими функциями называется суперпозицией функций. Эта операция позволяет с помощью функций меньшего числа аргументов получать функции большего их числа.

Логические элементы. Под логическим аргументом (иногда называемым вентилем) понимают устройство (обычно электронное), реализующее какую-либо простую логическую функцию. Исходя из суперпозиции булевых функций, можно сделать вывод о том, что любую сложную булеву функцию можно представить в виде комбинации простых функций и, следовательно, любой цифровой автомат можно реализовать из простых логических элементов.

Кроме того, из табл. 1.7 следует, что любую функцию можно выразить через три основные, или, что то же самое, любое цифровое устройство можно реализовать из набора элементов, состоящего из дизъюнкторов, конъюнкторов и инверторов.

Например, функцию неравнозначности можно выразить как

$$f_6 = \overline{x_1} \cdot x_0 \vee x_1 \cdot \overline{x_0}.$$

Устройство, выполняющее эту функцию, представлено на рис. 1.5. Таким образом, существует система логических функций (набор логических элементов), позволяющая представить любую сколь угодно сложную функцию (построить сколь угодно сложный цифровой автомат). Такая система функций называется функционально полной системой логических функций, а набор элементов, ей соответствующий, – функционально полным набором (ФПН).Набор логических элементов, состоящий из дизъюнктора, конъюнктора и инвертора

является ФПН. Существуют ФПН, состоящие из двух типов и даже из одного типа элементов. ФПН является набор, состоящий из конъюнктора и инвертора. Для того, чтобы доказать это, покажем, что из этих элементов можно построить дизъюнктор $y = x_1 \vee x_0$ и тогда мы получим уже известный ФПН.

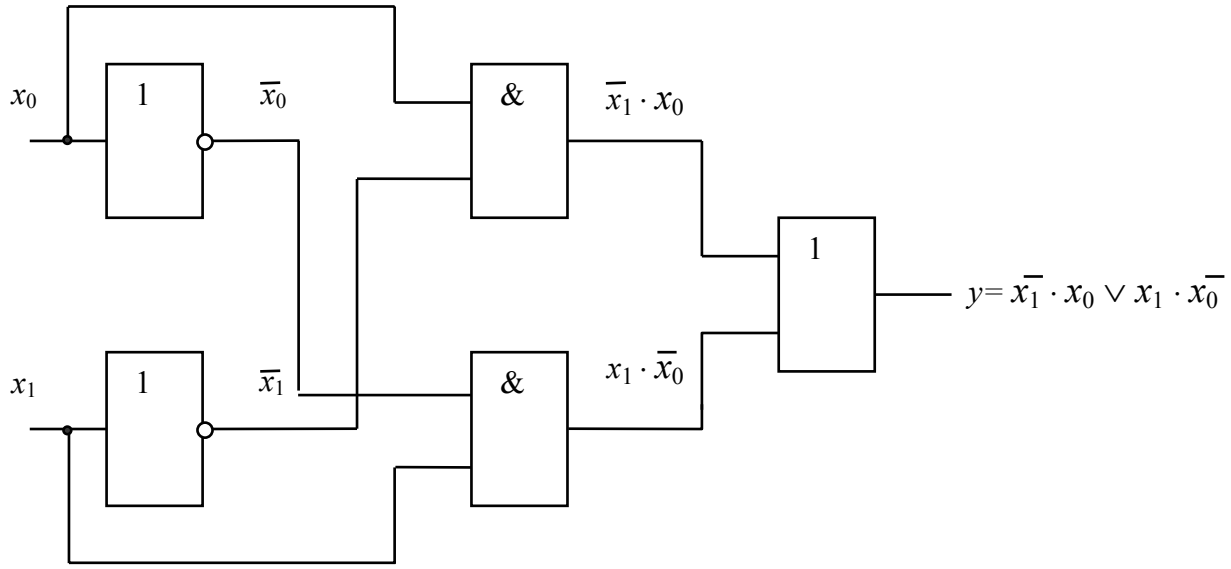


Рис. 1.5. Схема сумматора по модулю “2”

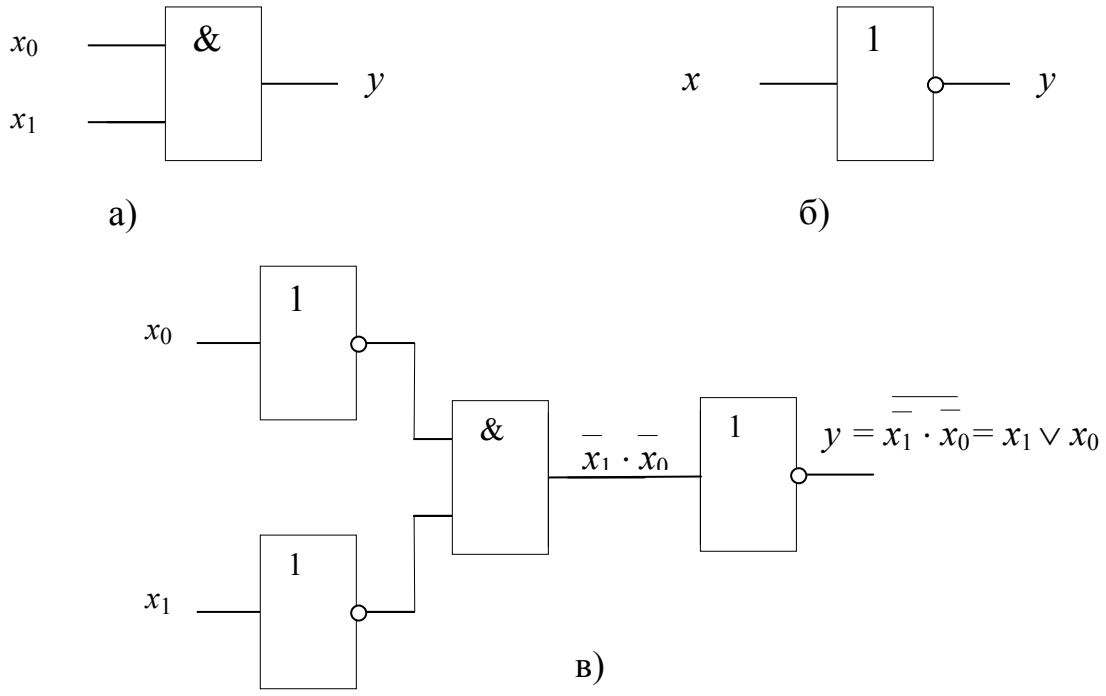


Рис. 1.6. Функционально полный набор, состоящий из двух типов элементов: конъюнктора (а) и инвертора (б); дизъюнктор, реализованный на основе конъюнктора и инвертора (в)

Воспользуемся теоремой Моргана и тождествами алгебры логики:

$$y = x_1 \vee x_0 = \overline{\overline{x_1 \vee x_0}} = \overline{\overline{x_1} \cdot \overline{x_0}}.$$

В полученное выражение входят лишь операции инвертирования (НЕ) и конъюнкции (И), выполняемые имеющимися в наборе элементами. Недостающий в наборе дизъюнктор реализуется по схеме рис.1.6 в. Аналогичным образом можно доказать функциональную полноту набора, состоящего из дизъюнкторов и инверторов.

Рассмотренные ФПН состояли из элементов, выполняющих какую-либо одну из основных функций. Такие элементы называются одноступенчатыми.

Примерами ФПН, состоящими из одного типа элементов, могут служить наборы из элементов Шеффера (И-НЕ) или из элементов Пирса (ИЛИ-НЕ).

Докажем функциональную полноту набора из элементов Шеффера. Для доказательства надо продемонстрировать, что из этих элементов можно построить дизъюнктор, конъюнктор и инвертор.

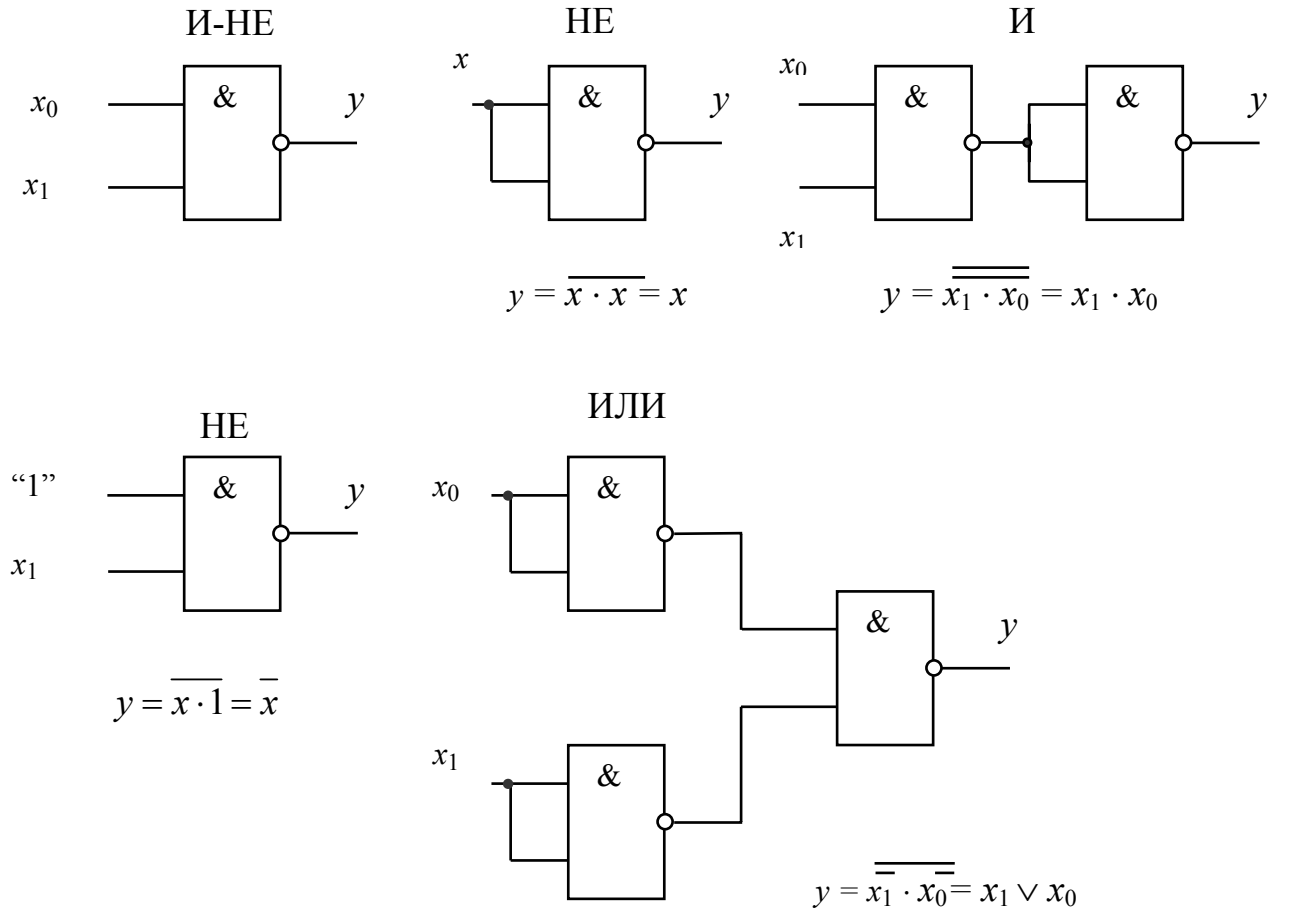


Рис. 1.7. Элемент Шеффера И-НЕ и реализация на его основе инвертора НЕ, конъюнктора И и дизъюнктора ИЛИ

На рис.1.7 представлены условное обозначение элемента Шеффера и реализация на его основе других элементов. Аналогично можно доказать функциональную полноту набора из элементов Пирса. В отличие от одноступенчатых элементов, элементы Шеффера и Пирса выполняют каждый две простые логические функции и поэтому называются двухступенчатыми логическими элементами. Существуют также ФПН, состоящие из других типов двухступенчатых и многоступенчатых элементов, однако наибольший объем выпускаемых серий цифровых интегральных микросхем (ИМС) составляют серии, где базовыми элементами являются элементы Шеффера или Пирса.

Таким образом, любое цифровое устройство может быть построено из ограниченного числа типов логических элементов. Это создает большое их преимущество перед аналоговыми схемами, состоящим из радиоэлементов самой широкой номенклатуры.

Минтермы и макстермы. В булевой алгебре большую роль играют определенные логические функции, называемые минтермами и макстермами.

Минтермом (конституентой единицы) называется логическая функция, которая принимает значение “единица” при одном из всех наборов аргументов и “нуль” при всех остальных. Количество минтермов заданного числа аргументов n, как следует из определения, равно числу P_n возможных комбинаций их значений.

В табл. 1.8 представлена таблица истинности минтермов двух аргументов.

Таблица 1.8. Минтермы функции двух аргументов

S	Аргументы		Минтермы			
	x_1	x_0	Mn_0	Mn_1	Mn_2	Mn_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

Аналитически минтерм может быть выражен в виде конъюнкции (произведения) всех аргументов, причем в прямой форме в него войдут аргументы, имеющие в рассматриваемом наборе значение “единица”, а в инверсной – имеющие значение “нуль”. Для минтермов (табл. 1.8) аналитические выражения имеют вид:

$$Mn_0 = \bar{x}_1 \cdot \bar{x}_0; \quad Mn_1 = \bar{x}_1 \cdot x_0;$$

$$Mn_2 = x_1 \cdot \bar{x}_0; \quad Mn_3 = x_1 \cdot x_0.$$

В общем виде образование минтерма с номером S осуществляется путем перевода S в двоичную систему счисления и записи конъюнкции с аргументами в прямой и инверсной форме. Например, запись Mn_{35} имеет следующий вид:

$$(35)_{10} = (\,1\;\;0\;\;0\;\;0\;\;1\;\;1\,)_2$$
$$\downarrow\;\;\underline{\downarrow}\;\;\underline{\downarrow}\;\;\underline{\downarrow}\;\;\downarrow\;\;\downarrow$$
$$Mn_{35} = \;\; x_5 \cdot x_4 \cdot x_3 \cdot x_2 \cdot x_1 \cdot x_0 \; .$$

Макстермом (конституентой нуля) называется логическая функция, равная нулю на одном из возможных наборов аргументов и единице на всех остальных. Количество макстермов так же, как и минтермов, определяется числом входных аргументов.

В табл. 1.9 представлена таблица истинности макстермов двух аргументов.

Аналитически макстерм выражается в виде дизъюнкции всех аргументов, причем в прямой форме в него войдут аргументы, имеющие в рассматриваемом наборе значение “нуль”, в инверсной – “единица”.

Для макстермов (табл. 1.9) аналитические выражения имеют вид

$$Mx_0 = \underline{x}_1 \vee x_0; \quad Mx_1 = \underline{x}_1 \vee \bar{x}_0;$$
$$Mx_2 = \underline{x}_1 \vee x_0; \quad Mx_3 = \underline{x}_1 \vee \bar{x}_0.$$

Таблица 1.9. Макстермы функции двух аргументов

S	Аргументы		Минтермы			
	x_1	x_0	Mx_0	Mx_1	Mx_2	Mx_3
0	0	0	0	1	1	1
1	0	1	1	0	1	1
2	1	0	1	1	0	1
3	1	1	1	1	1	0

Образование макстерма с номером S осуществляется так, что аргументы логически складываются и единице в двоичном представлении соответствует переменная в инверсной, а нулю – переменная в прямой форме

$$(35)_{10} = (\,1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1\,)_2$$
$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$
$$Mx_{35} = \bar{x}_5 \vee x_4 \vee x_3 \vee x_2 \vee \bar{x}_1 \vee \bar{x}_0 .$$

Как будет показано ниже, произвольная логическая функция может быть представлена в виде комбинации минтермов или макстермов.

1.4. Способы задания и преобразования логических функций

Словесный способ. При этом способе задания функция определяется словами, причем описание должно однозначно определять все случаи, в которых она принимает значение 0 или 1. Например, “Функция равна единице, если любые два или более аргумента равны единице, и нулю – во всех остальных случаях”.

Табличный способ. Способ задания булевых функций с помощью таблиц истинности позволяет достаточно легко перейти к любому другому способу. Таблица истинности представляет собой некоторую таблицу, в которой отмечены наборы (комбинации) входных переменных и значения функции на каждом наборе. Например, для функции, заданной выше словесным описанием, таблица истинности будет иметь вид, приведенный в табл. 1.10.

Аналитический способ. Под аналитическим способом задания логических функций подразумевается запись в виде алгебраического выражения. Для примера, рассмотренного выше, заданная функция может иметь вид

$$y = x_2 \cdot x_1 \vee \overline{\overline{x_1 \cdot x_0} \cdot \overline{x_2 \cdot x_0}} . \qquad (1.4)$$

Таблица 1.10. Таблица истинности функции

S	x_2	x_1	x_0	y
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

Аналитический способ задания логических функций имеет несколько форм.

Дизъюнктивно-нормальная форма (ДНФ) задания логической функции состоит из дизъюнкции простых конъюнкций (импликант) аргументов. Конъюнкция называется простой, если она является логическим произведением переменных в прямой или инверсной форме. ДНФ функции (1.4) имеет вид

$$y = x_2 \cdot x_1 \vee \overline{x_2} \cdot x_1 \cdot x_0 \vee x_2 \cdot \overline{x_1} \cdot x_0. \qquad (1.5)$$

ДНФ имеет свой завершенный вид, называемый совершенной дизъюнктивной нормальной формой (СДНФ), когда в конъюнкции входят все аргументы, т.е. они являются минтермами. Так, функция (1.5) в СДНФ описывается выражением

$$y = \overline{x_2} \cdot x_1 \cdot x_0 \vee x_2 \cdot \overline{x_1} \cdot x_0 \vee x_2 \cdot x_1 \cdot \overline{x_0} \vee x_2 \cdot x_1 \cdot x_0.$$

Конъюнктивно-нормальная форма (КНФ) представления функций содержит конъюнкцию простых дизъюнкций. Для функции (1.4) КНФ представляется выражением

$$y = (x_2 \vee x_1) \cdot (\overline{x_2} \vee x_1 \vee \overline{x_0}) \cdot (x_2 \vee x_1 \vee x_0).$$

Если каждая из простых конъюнкций является полной, т.е. макстермом, то говорят, что функция задана в совершенной конъюнктивной нормальной форме (СКНФ):

$$y = (x_2 \vee x_1 \vee x_0) \cdot (x_2 \vee x_1 \vee \overline{x_0}) \cdot (x_2 \vee \overline{x_1} \vee x_0) \cdot (x_2 \vee x_1 \vee x_0). \qquad (1.6)$$

Числовой способ. Функция в СДНФ может быть условно записана в виде

$$y = \sum (3, 5, 6, 7) = Mn_3 \vee Mn_5 \vee Mn_6 \vee Mn_7$$

В этой условной записи под знак суммы вводятся номера S минтермов (см.табл. 1.9), присутствующих в СДНФ функции.

Аналогично функция (1.6) в СКНФ условно записывается в виде произведения макстермов

$$y = \prod (0, 1, 2, 4) = Mx_0 \cdot Mx_1 \cdot Mx_2 \cdot Mx_4.$$

Задание логических функций на диаграммах Вейча или картах Карно. Очень часто удобным оказывается представление функции в специальных матрицах или диаграммах, называемых диаграммами Вейча или картами Карно. Диаграмма Вейча (карта Карно) – прямоугольная таблица, число ячеек которой равно числу возможных комбинаций аргументов 2^n. Подробно об этом методе будет рассказано ниже в разделе минимизации булевых функций.

Задание функций в виде временных диаграмм. Такая форма часто бывает удобной при задании функционирования какого-либо цифрового устройства. В этой форме строятся временные диаграммы, представляющие возможные комбинации на входе устройства, и временная диаграмма, представляющая значения выходной переменной.

Методы перехода от одного способа задания логических функций к другому. Каждая логическая функция может быть задана любым из перечисленных способов, и поэтому необходимо определить способы перевода функции от одного способа задания к другому.

Переход от табличного способа задания к аналитическому базируется на основной теореме булевой алгебры, утверждающей, что любая булева функция может быть представлена в виде логической суммы конъюнкций значений функции $y(S)$ с минтермами Mn_S, соответствующими этим наборам, или в виде логического произведения дизъюнкций значений функции $y(S)$ и макстермов Mx_S:

$$y = \bigvee_{S=0}^{2^{n-1}} y(S) \cdot Mn_S = \bigwedge_{S=0}^{2^{n-1}} [y(S) \vee Mx_S].$$

Доказательство теоремы. В соответствии с теоремой разложения Шеннона любую логическую функцию в общем виде можно разложить по одной из переменных на два слагаемых или на два сомножителя

$$f(x_{n-1}, \dots, x_1, x_0) = f(x_{n-1}, \dots, x_1, 1) \cdot x_0 \vee f(x_{n-1}, \dots, x_1, 0) \cdot \overline{x}_0;$$
$$f(x_{n-1}, \dots, x_1, x_0) = [f(x_{n-1}, \dots, x_1, 0) \vee x_0] \cdot [f(x_{n-1}, \dots, x_1, 1) \vee \overline{x}_0].$$

Справедливость этой теоремы легко установить, подставляя в обе части соотношений возможные значения переменной x (0 или 1). Подобным образом можно разложить по другой переменной каждое из полученных слагаемых, например:

$$f(x_{n-1}, \dots, x_1, x_0) = f(x_{n-1}, \dots, 1, 1) \cdot x_1 \cdot x_0 \vee f(x_{n-1}, \dots, 0, 1) \cdot \overline{x}_1 \cdot x_0 \vee$$
$$\vee f(x_{n-1}, \dots, 1, 0) \cdot x_1 \cdot \overline{x}_0 \vee f(x_{n-1}, \dots, 0, 0) \cdot \overline{x}_1 \cdot \overline{x}_0.$$

Если эту операцию проделать для всех переменных x_i то результатом будет

$$f(x_{n-1}, ..., x_1, x_0) = f(1, 1, ..., 1, 1) \cdot x_{n-1} \cdot x_{n-2} \cdot ... \cdot x_1 \cdot x_0 \vee ... \vee$$
$$\vee f(0, 0, ..., 0, 0) \cdot \overline{x_{n-1}} \cdot \overline{x_{n-2}} \cdot ... \cdot x_1 \cdot x_0 = \bigvee_{S=0}^{2^{n-1}} y(S) \cdot Mn_S.$$

Аналогичным образом можно доказать соотношение для конъюнктивного представления функции

$$y = \bigwedge_{S=0}^{2^{n-1}} [y(S) \vee Mx_S].$$

Таким образом, справедливость теоремы доказана.

Использование теоремы для перехода от табличного задания к аналитическому покажем на следующем примере (табл. 1.11):

Таблица 1.11. Таблица истинности функции

S	x_2	x_1	x_0	y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

$$y = \bigvee_{S=0}^{7} y(S) \cdot Mn_S = 0 \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} \vee 0 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 \vee 1 \cdot \overline{x_2} \cdot x_1 \cdot \overline{x_0} \vee$$
$$\vee 0 \cdot \overline{x_2} \cdot x_1 \cdot x_0 \vee 1 \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} \vee 1 \cdot x_2 \cdot \overline{x_1} \cdot x_0 \vee 0 \cdot x_2 \cdot x_1 \cdot \overline{x_0} \vee$$
$$\vee 1 \cdot x_2 \cdot x_1 \cdot x_0 = \sum (2, 4, 5, 7).$$

$$y = \bigwedge_{S=0}^{7} [y(S) \vee Mx_S] = (0 \vee x_2 \vee x_1 \vee x_0) \cdot (0 \vee x_2 \vee x_1 \vee \overline{x_0}) \times$$
$$\times (1 \vee x_2 \vee \overline{x_1} \vee x_0) \cdot (0 \vee x_2 \vee \overline{x_1} \vee \overline{x_0}) \cdot (1 \vee \overline{x_2} \vee x_1 \vee x_0) \times$$
$$\times (1 \vee \overline{x_2} \vee x_1 \vee \overline{x_0}) \cdot (0 \vee \overline{x_2} \vee \overline{x_1} \vee x_0) \cdot (1 \vee \overline{x_2} \vee \overline{x_1} \vee \overline{x_0}) = \prod (0, 1, 3, 6).$$

Из рассмотренного примера видно, что табличный способ задания функции преобразуется в СДНФ или СКНФ.

Переход от аналитического задания к табличному может быть осуществлен путем подстановки в выражение функции поочередно всех возможных комбинаций входных аргументов и определения с использованием логических тождеств значения функции на этих наборах. В качестве примера рассмотрим переход от аналитического способа задания функции к табличному:

$$y = \overline{x_1 \cdot x_0} \vee (\underline{\overline{x_1} \vee x_0) \cdot (\, x_2 \cdot x_0\,}).$$

Таблица 1.12. Таблица истинности функции

S	x_2	x_1	x_0	y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Из заданного выражения видно, что функция имеет три аргумента, и поэтому возможны 2^3 =8 комбинаций. В левой части таблицы истинности (табл. 1.12) переберем все возможные комбинации.

Затем найдем $y(0) = y(0, 0, 0) = 0 \cdot 0 \vee (\, 0 \vee \overline{0}\,) \cdot 0 \cdot 0 = 1$. Таким образом, $y(S)$ при $S = 0$ равна 1.

Проводим эту операцию для всех наборов и получаем функцию, заданную таблично.

1.5. Минимизация логических функций

Необходимость минимизации продемонстрируем на примере синтеза комбинационного цифрового автомата в базисе, состоящем из элементов И, ИЛИ, НЕ, функционирование которого задается функцией

$$y = \sum (0, 2, 3) = \overline{x_1} \cdot \overline{x_0} \vee x_1 \cdot \overline{x_0} \vee x_1 \cdot x_0. \qquad (1.7)$$

По данному выражению рисуем схему устройства (рис. 1.8а).

Как видно, нам потребовалось 6 логических элементов. Попытаемся упростить (1.7), используя тождества алгебры логики:

$$y = \overline{x_1} \cdot \overline{x_0} \vee x_1 \cdot \overline{x_0} \vee x_1 \cdot x_0 = \overline{x_0} \cdot (\overline{x_1} \vee x_1) \vee x_1 \cdot x_0 =$$
$$= \overline{x_0} \vee x_1 \cdot x_0 = (\overline{x_0} \vee x_0) \cdot (\overline{x_0} \vee x_1) = x_1 \vee \overline{x_0}.$$

Схема устройства (рис. 1.8б), построенного по упрощенному выражению, реализует ту же функцию существенно меньшим числом логических элементов (всего необходимо два элемента). Такое устройство будет экономичней и дешевле, чем автомат на рис. 1.8а. Следовательно, перед реализацией цифрового автомата логическую функцию необходимо упростить или минимизировать.

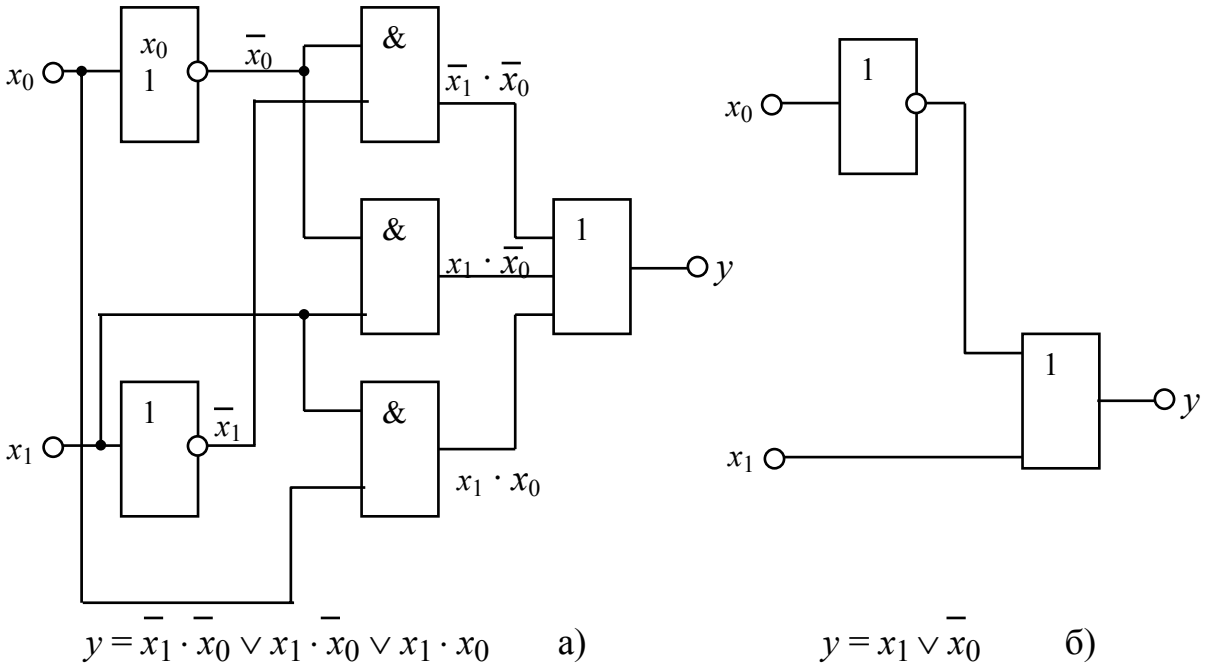


Рис.1.8. Реализация цифрового устройства:
по СДНФ (а), по минимизированной функции (б)

Строгое решение задачи минимизации должно учитывать конкретные особенности логических схем, применяемой элементной базы (в частности, значения коэффициента объединения, нагрузочной способности логических элементов, число элементов в корпусах, стоимость каждого корпуса и т.п.). Однако применяемые в настоящее время математические методы упрощения не решают задачу в таком объеме.

Наиболее детально разработаны методы решения канонической задачи минимизации функций, которая заключается в отыскании ДНФ функции, содержащей минимальное число вхождений аргументов (минимальное число букв). Такие формы принято называть оптимальными или минимальными ДНФ. Исходной формой функции при решении задачи минимизации является ее СДНФ (если функция задана в другой форме, ее переводят в СДНФ). Систематические методы минимизации (в настоящее время известно несколько таких методов, такие как метод Квайна, метод Квайна-Мак-Класки) предполагают применение формализованного порядка упрощения формул. Эти методы описываются строгими алгоритмами и поддаются программированию; их применение дает возможность использовать ЭВМ, что является неизбежным при минимизации формул функций большого числа аргументов (больше шести). При меньшем числе аргументов логические функции могут быть легко минимизированы по методу Вейча-Карно.

Метод Вейча - Карно. Метод минимизации функций, предложенный Вейчем и усовершенствованный Карно, основан на упомянутом выше представлении функции в специальных матрицах, именуемых диаграммами Вейча или картами Карно. Метод Вейча - Карно облегчает процедуру склеивания соседних минтермов благодаря тому, что члены СДНФ и СКНФ (минтермы или макстермы) размещаются на плоскости таким образом, что соседние члены, для которых возможно склеивание, оказываются в геометрической близости друг от друга.

Диаграммы Вейча (карты Карно) представляют собой прямоугольные таблицы, разделенные горизонтальными и вертикальными линиями на ячейки, общее число которых совпадает с числом минтермов или возможных комбинаций аргументов. Каждая ячейка соответствует своему минтерму, причем размещение последних осуществляется таким образом, чтобы два соседних минтерма находились в смежных ячейках. Такой порядок размещения минтермов (макстермов) обеспечивается принятым способом образования наборов аргументов, соответствующих различным ячейкам диаграммы. На рис. 1.9 приведены примеры диаграмм Вейча для двух, трех и четырех переменных, а также показано соответствие ячеек и минтермов.

Для облегчения процесса нахождения нужной ячейки по краям диаграммы ставятся значения переменных (рис. 1.9). Например, в верхнюю строку в диаграмме Вейча для функции трех переменных, обозначенную символом x_1, вносятся все минтермы, в которые x_1 входит в прямой форме, в нижнюю – минтермы, куда x_1 входит в инверсной форме.

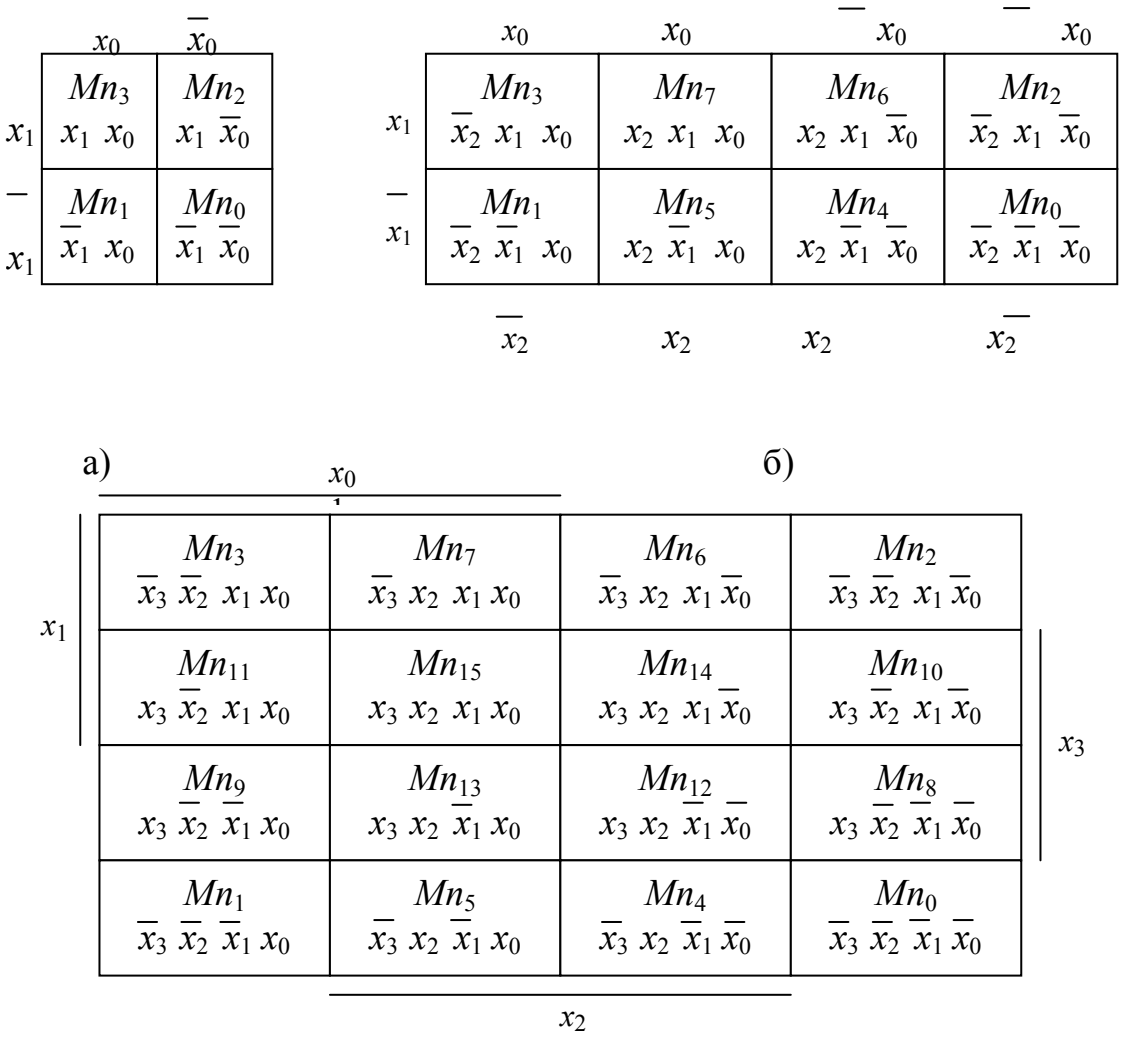


Рис. 1.9. Диаграмма Вейча для функции двух (а), трех (б) и четырех (в) переменных

В дальнейшем условно будем обозначать строки и столбцы, соответствующие переменным в прямой форме, линиями (рис. 1.9в).

Представление функций на диаграмме Вейча. Для представления функции на диаграмме Вейча ее необходимо перевести в СДНФ или задать таблично. Затем на диаграмме отметить единицами те ячейки, которые соответствуют имеющимся в СДНФ функции минтермам. Остальные ячейки либо остаются незаполненными, либо отмечаются нулями. В качестве примера представим функцию, заданную таблично на диаграмме Вейча (рис. 1.10).

S	x_2	x_1	x_0	y
	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

Функция в СДНФ

$$y = \overline{x_2} \cdot x_1 \cdot x_0 \vee \overline{x_2} \cdot \overline{x_1} \cdot x_0$$

Диаграмма Вейча

с занесенной функцией

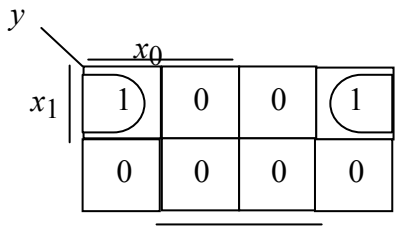


Рис. 1.10. Представление логической функции на диаграмме Вейча

Считывание минимизированной функции с диаграммы. После занесения функции в диаграмму Вейча обычно сразу видна минимальная форма функции. Так, если функция содержит единицы в соседних ячейках, то они могут быть объединены в контур и в конечном выражении функции предстанут в виде импликанты, в которой будет отсутствовать переменная, меняющая знак для заданного объединения. Контур образуют также ячейки, находящиеся на противоположных краях диаграммы Вейча.

Для рассмотренного примера могут быть объединены в контур две единицы, отображающие соседние минтермы, как показано на рис. 1.10:

$$y = Mn_2 \vee Mn_3.$$

На рис. 1.11 представлены некоторые возможные варианты объединения единиц в контуры и минимальные функции, которые при этом получаются.

В общем виде можно сформулировать следующие правила считывания минимизированной функции с диаграмм Вейча:

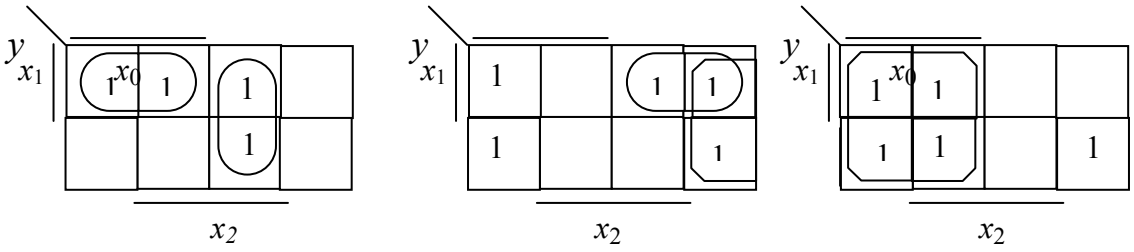
а) минимизированная функция равна дизъюнкции импликант, соответствующих объединенным в контуры единицам;

б) в контуры можно объединять 1, 2, 4, 8, 16, …, 2^n единиц, причем они должны составлять строку, столбец (или их часть) – прямоугольник или квадрат. В эти контуры объединяют и единицы, расположенные на краях диаграммы Вейча;

в) контурами должны охватываться все без исключения единицы, и ни в один контур не должен быть включен нуль;

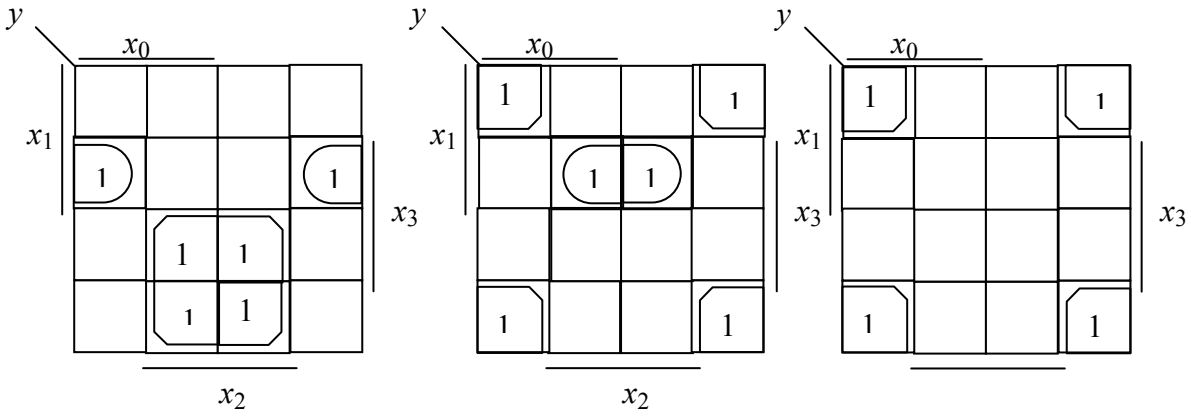
г) одна и та же единица может быть включена в несколько контуров;

д) в импликанту, представляющую данный контур, входят только те переменные, которые для этого контура не меняют своих значений.



$y = x_1 \cdot x_0 \vee x_2 \cdot \overline{x}_0$ $\quad y = \overline{x}_2 \vee x_1 \cdot \overline{x}_0$ $\quad y = x_0 \vee \overline{x}_2 \cdot \overline{x}_1$

Рис. 1.11. Возможные варианты объединения единиц в контуры и результаты склеивания



$y = x_3 \cdot \overline{x}_2 \cdot x_1 \vee x_2 \cdot \overline{x}_1$ $\quad y = \overline{x}_3 \cdot \overline{x}_2 \vee x_3 \cdot x_2 \cdot x_1$ $\quad y = x_2$

Рис. 1.12. Примеры считывания минимизированных функций с диаграмм Вейча для четырех переменных

На рис. 1.12 показаны возможные варианты объединения единиц и получающиеся функции для четырех аргументов.

Еще большие возможности для объединения единиц в контур представляются в диаграммах Вейча для функций пяти аргументов. Поскольку число соседних минтермов для функции n аргументов равно n, то соседними ячейками диаграмм Вейча для пяти переменных будут не только смежные по вертикали и горизонтали, но и ячейки, расположенные симметрично относительно вертикальной оси, разделяющей диаграмму надвое (рис. 1.13).

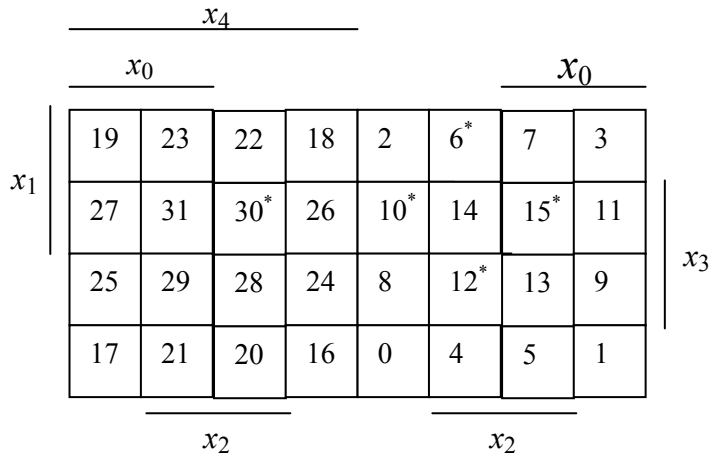


Рис. 1.13. Диаграмма Вейча для функций пяти переменных.
Цифрами обозначены номера минтермов, заносимых в ячейки

Так, например, для ячейки 14, отображающей Mn_{14}, соседними будут ячейки с номерами 6, 10, 12, 15, 30 (обозначены звездочкой). Для ячейки с номером 25 – ячейки 9, 17, 24, 27, 29. Это позволяет производить объединения, некоторые из которых представлены на рис. 1.14.

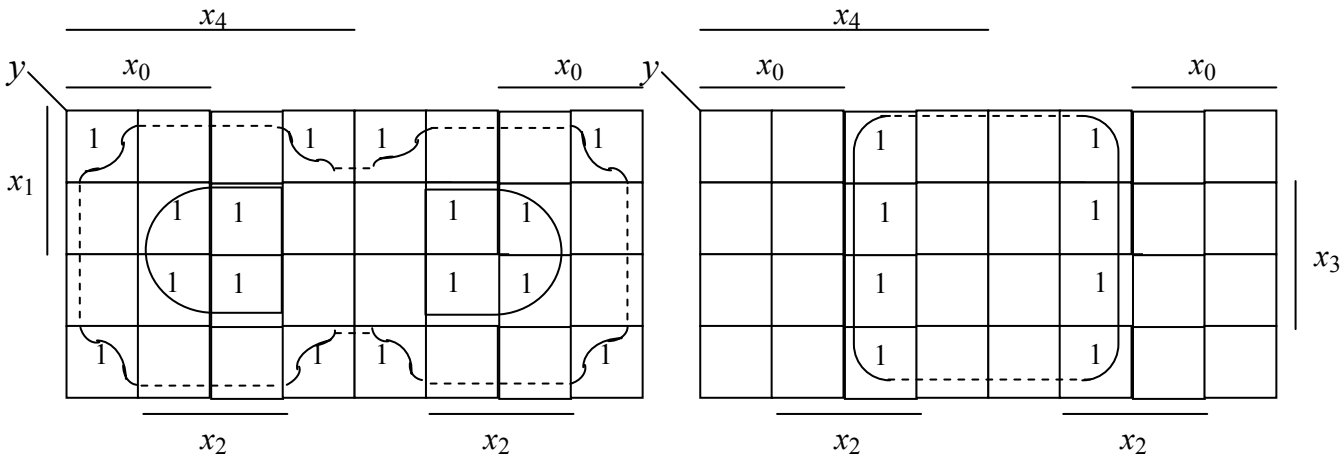


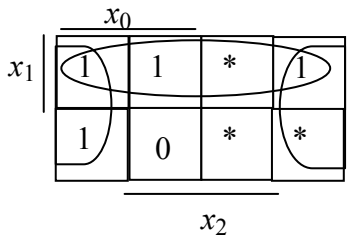
Рис. 1.14. Примеры считывания минимизированных функций
с диаграмм Вейча для пяти переменных

Минимизация частично определенных функций. Логическая функция может быть определена на части возможных комбинаций значений аргументов. Комбинации, на которых функция не определена, называются запрещенными или избыточными.

Частичное определение функций создает возможность факультативного доопределения функций на избыточных комбинациях, что используется для получения по возможности минимальной ДНФ. Рассмотрим пример использования факультативного доопределения частично определенной функции.

На рис. 1.15 исходная функция задана в виде таблицы, где звездочкой обозначены значения функций, которые соответствуют запрещенным комбинациям аргументов. Данные комбинации отражаем на диаграмме Вейча звездочкой. При минимизации доопределяем функцию таким образом, чтобы вся функция покрывалась минимальным числом наибольших по размерам контуров (в данном случае двумя контурами по четыре единицы). Вариант доопределения функции на запрещенных комбинациях позволяет получить минимальную ДНФ. Необходимо заметить, что полученная функция определена уже на всех комбинациях так, как это представлено в правом столбце таблицы на рис. 1.15.

S	x_2	x_1	x_0	y	$y_д$
0	0	0	0	*	1
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	1	1
4	1	0	0	*	0
5	1	0	1	0	0
6	1	1	0	*	1
7	1	1	1	1	1



$$y_д = \overline{x_2} \vee x_1$$

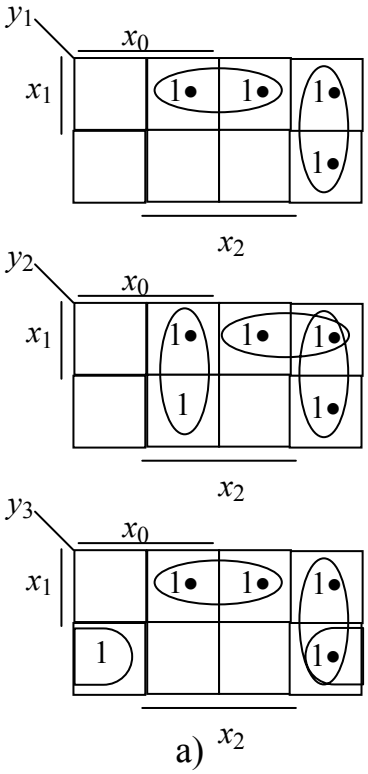
Рис. 1.15. Примеры минимизации частично определенной функции

Совместная минимизация функций. В ряде случаев необходимо синтезировать цифровой автомат, имеющий несколько входов и несколько выходов. Это обстоятельство можно использовать для совместной минимизации булевых функций, поскольку во все функции могут входить одинаковые минтермы и импликанты. Эту часть функции можно использовать многократно при реализации всего цифрового устройства с несколькими выходами. Способ совместной минимизации рассмотрим на примере.

На рис. 1.16а представлены на диаграммах Вейча три функции y_0, y_1, y_2 одних и тех же аргументов x_0, x_1, x_2.

Если подходить к минимизации этих функций раздельно, то три сокращенные функции вместе будут иметь 16 вхождений аргументов (рис. 1.16б). При совместной минимизации целесообразно отметить общие для всех функций минтермы (на диаграммах Вейча они отмечены точкой), а все логические функции представить в виде логической суммы общей части z и необходимых импликант, как это показано на рис. 1.16в. Как видно, при совместной минимизации, удалось уменьшить суммарное число вхождений аргументов в функции для данного конкретного примера в 2 раза.

Минимизация функций “по нулям”. Был рассмотрен способ считывания минимизированных функций, заключающийся в объединении в контуры тех ячеек диаграммы Вейча, где функция определена единицей. Однако возможно объединение по тем же правилам ячеек, где функция определена нулем. В этом случае получается минимальная ДНФ инвертированной функции (рис. 1.17).



а)

$y_0 = x_2 \cdot x_1 \vee \overline{x_2} \cdot \overline{x_0}$
(4 вхождения)

$y_1 = x_2 \cdot x_0 \vee x_1 \cdot \overline{x_0} \vee \overline{x_2} \cdot \overline{x_0}$
(еще 6 вхождений)

$y_2 = \overline{x_2} \cdot \overline{x_1} \vee x_2 \cdot x_1 \vee \overline{x_2} \cdot \overline{x_0}$
(еще 6 вхождений)
ИТОГО:
16 вхождений

б)

$z = x_2 \cdot x_1 \vee \overline{x_2} \cdot \overline{x_0}$
$y_0 = z$
(4 вхождения)

$y_1 = z \vee x_2 \cdot x_0$
(еще 2 вхождения)

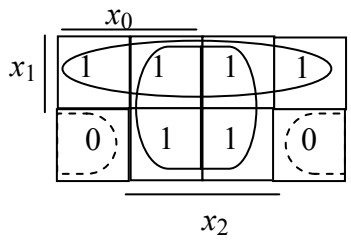
$y_2 = z \vee \overline{x_2} \cdot \overline{x_1}$
(еще 2 вхождения)
ИТОГО:
8 вхождений

в)

Рис.1.16. Диаграммы Вейча с занесенными функциями (а), минимизация без использования общей части (б), совместная минимизация (в)

С помощью теоремы Моргана можно показать, что функции, полученные двумя способами, тождественно равны:

$$y = \overline{\overline{y}} = \overline{\overline{x_1} \cdot \overline{x_2}} = x_2 \vee x_1.$$



считывание по “единицам”

$$y = x_2 \vee x_1$$

считывание по “нулям”

$$\overline{y} = \overline{x}_2 \cdot \overline{x}_1$$

Рис. 1.17. Пример считывания функции

Карты Карно. Разновидностью таблиц, на которых представляются логические функции, являются карты Карно. Они отличаются от диаграмм Вейча лишь способом размещения переменных. В картах Карно переменные перечисляются в циклическом коде или коде Грея. Занесение функций на карты Карно и считывание минимизированных форм ведется так же, как при использовании диаграмм Вейча.

1.6. Вопросы и упражнения

1. Дайте определения цифровых автоматов комбинационного и последовательностного типов.

2. Докажите справедливость логического тождества

$$(x_2 \vee x_1) \cdot (x_1 \vee x_0) = x_1 \vee x_2 \cdot x_0 \vee x_1 \cdot x_0.$$

3. Считая теорему Моргана для двух аргументов доказанной, обобщите ее для произвольного числа аргументов.

4. Определите общее число логических функций одного аргумента, перечислите их и назовите соответствующие логические элементы.

5. Определите число логических функций восьми аргументов.

6. Дайте определение минтерма и макстерма. Запишите аналитические выражения для Mn_{239} и Mn_{534}.

7. Определите таблицу истинности функции и представьте в СДНФ и СКНФ $y = x_2 \cdot x_1 \cdot \overline{x}_0 \vee x_2 \cdot x_1 \vee \overline{x}_0 \vee \overline{x}_2 \cdot x_1$.

8. Дайте определение функциональной полноты набора логических элементов.

9. Исследуйте функциональную полноту набора логических элементов, реализующих функцию $y = x_1 \vee x_0$.

10. Исследуйте функциональную полноту набора логических элементов, реализующих функцию неравнозначности.

11. Исследуйте функциональную полноту набора логических элементов, реализующих функцию Шеффера.

12. Исследуйте функциональную полноту набора логических элементов, реализующих функцию Пирса.

13. Определите взаимосвязь функций $y_0 = \sum (0, 1, 3, 7)$ и $y_1 = \sum (2, 4, 5, 6)$.

14. С какой целью проводится минимизация логических функций?

15. Минимизируйте функцию $y = \sum (1, 3, 4, 5, 9, 11, 12, 13)$ методом Вейча - Карно.

16. Минимизируйте функцию $y = \sum (1, 3, 12, 14, 17, 19, 28, 30)$ методом Вейча - Карно.

17. Проведите совместную минимизацию функций

$$y_0 = \sum (0, 2, 6, 7) = \quad \text{и} \quad y_1 = \sum (0, 1, 2, 6, 7).$$

18. Пользуясь правилом де Моргана, преобразуйте следующие логические выражения:

а) $\overline{x_1 \cdot x_2 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot x_3}$;

б) $\overline{(x_1 \cdot x_2) \vee (\bar{x}_1 \cdot x_2 \cdot x_3) \vee (x_1 \cdot \bar{x}_2 \cdot \bar{x}_3)}$;

в) $\overline{(x_1 \cdot x_2 \cdot x_3 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot x_3) \cdot (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \vee x_1 \cdot \bar{x}_2 \cdot x_3)}$.

2. КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

2.1. Логический синтез комбинационных цифровых устройств

У комбинационных цифровых устройств состояние выхода зависит только от состояния их входов в данный тактовый момент времени. Синтез (логическое проектирование) комбинационных устройств осуществляется на основе математического аппарата, изложенного в предыдущей главе. Для построения разрабатываемых устройств используется какой-либо функционально полный набор логических элементов (простейших комбинационных цифровых автоматов).

Логическое проектирование схем комбинационных цифровых устройств состоит из следующей последовательности этапов:

1) кодирование входных и выходных переменных и переход от словесного задания функции к табличному;

2) переход от табличной формы задания к СДНФ;

3) минимизация функции или получение ее минимальной ДНФ;

4) переход от минимальной ДНФ к минимальной форме в используемом базисе (функционально полном наборе) или перевод функции из ДНФ в структурный вид в используемом базисе;

5) составление схемы устройства.

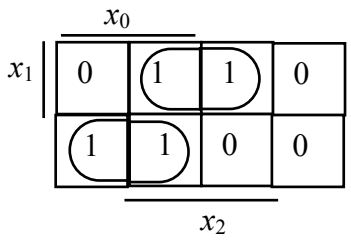
В качестве примера рассмотрим синтез устройства, задаваемого следующим описанием: имеются три датчика, которые одновременно формируют на своих выходах сообщение “да” или “нет”. Необходимо синтезировать на элементах Шеффера цифровой автомат, обеспечивающий принятие решения по следующему алгоритму.

Таблица 2.1. Табличное задание функции

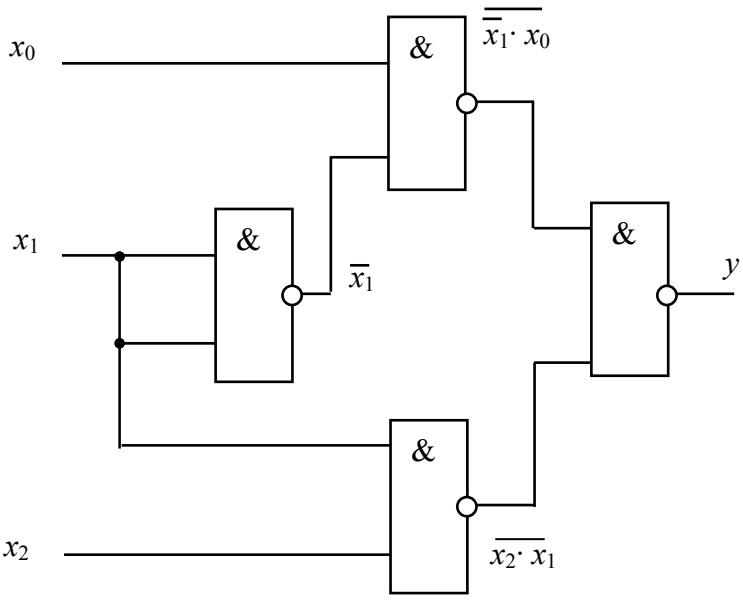
	2	1	0	

Принимается решение “да”, если сообщение “да” имеется или только на выходах первого, или второго и третьего, или на выходах всех трех датчиков.

1. Число входов устройства комбинационного цифрового автомата, очевидно, равно трем (x_2, x_1, x_0), а выход один (y). Условимся, что сообщению “да” соответствует сигнал “логическая единица” ($x_i = 1$, для $i = 0, 1, 2$; $y = 1$), сообщению “нет” – “логический нуль” ($x_i = 0$, $i = 0, 1, 2$; $y = 0$). Исходя из описания функционирования устройства и принятого кодирования входных и выходных переменных заполняем таблицу истинности (табл. 2.1).



а)



б)

Рис. 2.1. Синтез комбинационного цифрового автомата.
Диаграмма Вейча (а), схема (б)

2. По изложенному правилу в первой главе переходим к СДНФ функции:

$$y = \sum (1, 5, 6, 7) = \overline{x_2}\cdot x_1 \cdot x_0 \vee \overline{x_2} \cdot x_1 \cdot x_0 \vee x_2 \cdot \overline{x}_1 \cdot x_0 \vee x_2 \cdot x_1 \cdot x_0.$$

3. Минимизируем данную функцию по методу Вейча - Карно (рис. 2.1а). Из диаграммы Вейча находим минимальную ДНФ функцию

$$\overline{y} = x_1 \cdot x_0 \vee x_2 \cdot x_1.$$

4. Для перевода полученного выражения в структурный вид в базисе элементов Шеффера производим двухкратное инвертирование функции и преобразование по теореме Моргана

$$\overline{\overline{y}} = \overline{\overline{x_1 \cdot x_0 \cdot x_2 \cdot x_1}}.$$

5. По полученному выражению изобразим схему синтезированного устройства (рис. 2.1б).

2.2. Элементная база для практической реализации цифровых устройств

Для практической реализации цифровых автоматов обычно используются необходимые элементы или целые устройства, входящие в серии цифровых ИМС. Основой каждой серии является базовый логический элемент. На первом этапе развития элементной базы цифровых устройств были разработаны многочисленные и довольно разнородные системы логических элементов, затем преимущественное распространение получили несколько систем элементов, обеспечивающих наилучшее сочетание важнейших для потребителей и изготовителей характеристик.

В настоящее время выпускается огромное количество разнообразных интегральных микросхем (ИМС) от простейших логических элементов до сложнейших процессоров, микроконтроллеров и специализированных БИС (больших интегральных схем)/СБИС (сверхбыстродействующих интегральных схем) с программируемой структурой. Выпуском ИМС занимается множество фирм как отечественных, так и зарубежных. Наибольшее распространение получили две технологии цифровых микросхем: ТТЛ (TTL) и ТТЛШ (TTLS) – биполярная транзистор-транзисторная логика и ТТЛ с диодами Шоттки; КМОП (CMOS) – комплементарные транзисторы со структурой “металл – окисел - полупроводник”.

Элементной базой, используемой в комбинационных устройствах является набор ИМС малой и средней степени интеграции, в основе которой лежат ТТЛ серии. Эти серии включают в себя функционально полный комплекс ИМС, используя который можно создавать самые разные цифровые устройства.

Все цифровые ИМС работают с логическими сигналами, имеющими два разрешенных уровня напряжения. Один из этих уровней называется уровнем логической единицы (или единичным уровнем), другой – уровнем логического нуля (или нулевым уровнем). Чаще всего логическому нулю соответствует низ-

кий уровень напряжения U^0 или U_L (L – *LOW* – низкий), а логической единице – высокий уровень напряжения U^1 или U_H (H – *HIGH* – высокий). В этом случае говорят, что принята «положительная логика». Однако, при передаче сигналов на большие расстояния и в системных шинах микропроцессорных систем используют обратное представление, когда логической единице соответствует низкий уровень напряжения, а логическому нулю – высокий уровень напряжения. В этом случае говорят об отрицательной логике. В дальнейшем будем пользоваться соглашениями положительной логики.

В настоящее время существуют системы автоматизированного проектирования, которые не только рассчитывают готовые схемы, но способны и проектировать новые схемы по формализованным описаниям функций, которые данное устройство должно выполнить. Однако, по-настоящему эффективные, оптимизированные по числу используемых аппаратных моделей, наконец, красивые схемы может разработать только человек, который всегда подходит к проектированию творчески и использует оригинальные идеи.

Цифровые логические элементы. Элементной базой, применяемой при построении комбинационных цифровых устройств являются логические элементы. Их выбор производится в соответствии с определенным набором их параметров. Рассмотрим основные из них.

Коэффициент объединения по входу ($K_{об}$) определяет число входов элемента, предназначенных для подачи логических переменных. Элемент с большим коэффициентом объединения по входу имеет более широкие функциональные возможности.

Нагрузочная способность или коэффициент разветвления по выходу ($K_{раз}$) определяет число входов аналогичных элементов, которые одновременно могут быть подключены к выходу данного логического элемента. Чем выше нагрузочная способность элементов, тем меньшее число элементов может потребоваться при построении цифрового устройства.

Быстродействие ($t_{здр}$) логического элемента является одним из важнейших его параметров. Оно оценивается задержкой распространения сигнала от входа к выходу элемента. Определяется двумя параметрами $t_{здр}^{1,0}$ – время задержки переключения выхода элемента из состояния логической 1 в состояние логического 0 и $t_{здр}^{0,1}$ – задержка переключения из состояния 0 в состояние 1. Время задержки измеряется на уровне, среднем между уровнями логического 0 и логической 1.

Средняя задержка распространения сигнала $t_{здр\ ср} = 0{,}5\ (t_{здр}^{0,1} + t_{здр}^{1,0})$. Этот параметр используется при расчете задержки распространения сигналов в сложных логических схемах.

Временные диаграммы переключения инвертирующего логического элемента (рис. 2.2) показывают длительности характерных этапов переходных процессов.

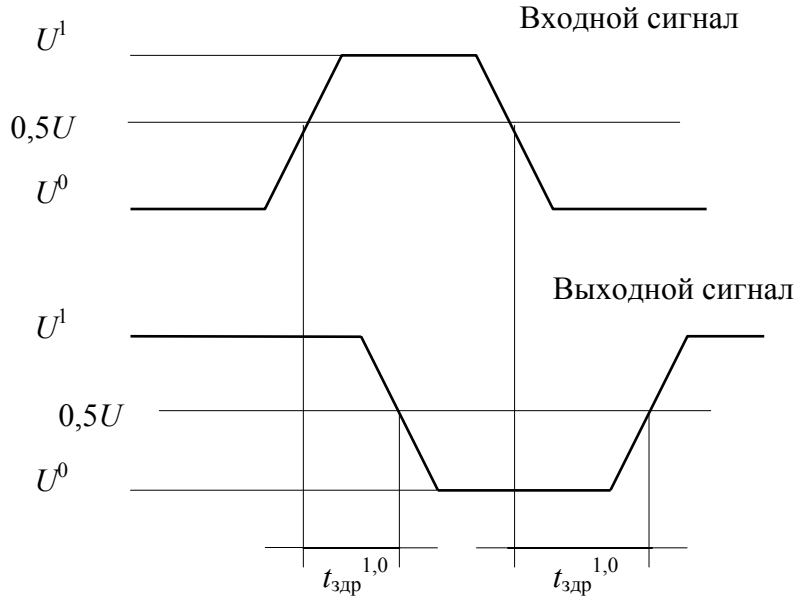


Рис. 2.2. Временные диаграммы процессов переключения логического элемента НЕ

Для правильного проектирования и эксплуатации цифровых устройствы необходимо знать систему статических параметров логических элементов, которые оговариваются в технических условиях.

СТАТИЧЕСКИЕ ПАРАМЕТРЫ ЭЛЕМЕНТОВ. В КАЧЕСТВЕ ВАЖНЕЙШИХ СТАТИЧЕСКИХ ПАРАМЕТРОВ ПРИВОДЯТСЯ ЧЕТЫРЕ ЗНАЧЕНИЯ НАПРЯЖЕНИЙ И ЧЕТЫРЕ ЗНАЧЕНИЯ ТОКОВ.

Четыре значения напряжений задают границы отображения переменных (0 и 1) на выходе и входе элемента. Для нормальной работы элемента требуется, чтобы напряжение, отображающее логическую 1, было достаточно высоким, а напряжение, отображающее 0, – достаточно низким. Эти требования задаются параметрами $U^1_{\text{вых. min}}$ и $U^0_{\text{вых. max}}$. Входные напряжения данного элемента есть выходные напряжения предыдущего (источника сигналов). Уровни, гарантируемые на выходе элемента при соблюдении допустимых нагрузочных условий, задаются параметрами $U^1_{\text{вх. min}}$ и $U^0_{\text{вх. max}}$. Выходные уровни несколько "лучше" входных, что обеспечивает определенную помехоустойчивость элемента. Для уровня U^1 опасны отрицательные помехи, снижающие его, причем допустимая статическая помеха (т.е. помеха любой длительности)

$$U^{-}_{\text{пом}} = U^1_{\text{вых.min}} - U^1_{\text{вх.min}} .$$

Для уровня U^0 опасны положительные помехи, причем допустимая статическая помеха

$$U^{+}_{\text{пом}} = U^{0}_{\text{вых.max}} - U^{0}_{\text{вх.max}}.$$

Четыре значения токов – входные и выходные токи в обоих логических состояниях. При высоком уровне выходного напряжения из элемента – источника ток вытекает, цепи нагрузки ток поглощают. При низком уровне выходного напряжения элемента-источника ток нагрузки втекает в этот элемент, а из входных цепей элементов-приемников токи вытекают. Зная токи $I^{1}_{\text{вых.max}}$ и $I^{0}_{\text{вых.max}}$, характеризующие возможности элемента – источника сигнала, и токи $I^{1}_{\text{вх.max}}$ и $I^{0}_{\text{вх.max}}$, потребляемые элементами-приемниками, можно контролировать соблюдение нагрузочных ограничений, обязательное для всех элементов схемы цифровых устройств.

Мощности потребления ($P_{\text{пот}}$). При разработке цифровых устройств требуется оценивать мощности потребления логических элементов, чтобы сформулировать требования к источникам питания и конструкции теплоотвода. При этом суммируются мощности, рассеиваемые логическими и другими элементами схемы, а также межсоединениями. Мощности, потребляемые элементами, делят на статические и динамические. Статическая мощность потребляется элементом во включенном состоянии. При переключении потребляется дополнительно динамическая мощность, которая пропорциональна частоте переключения элемента. Таким образом, полная мощность зависит от частоты переключения элемента, что и следует учитывать при ее подсчете. Обычно не возникает трудностей при подсчете мощностей, потребляемых биполярными схемами. При подсчете мощностей, потребляемых элементами типа КМОП, положение намного сложнее и данных, приведенных в справочниках, может не хватить.

Вид реализуемой логической функции. Используется широкий набор логических элементов: И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ. Перед обозначением логического элемента обычно указывается число его входов, по которым реализуется соответствующая логическая функция (то есть коэффициент объединения по входу). Число выходов по И и ИЛИ лежит в пределах от 2 до 16. В одной микросхеме может быть несколько логических элементов. В табл. 2.2 приведены основные логические функции, обозначения и их схемы.

Некоторые параметры ИМС наиболее распространенных серий приведены в табл. 2.3. Каждая технология непрерывно совершенствуется с целью увеличения быстродействия ИМС, уменьшения потребляемой мощности и увеличения степени интеграции. При проектировании цифровых устройств, значения параметров используемых ИМС необходимо находить в справочных материалах.

По сочетанию параметров наибольшее распространение получили ИМС серии ТТЛ.

Таблица 2.2. Основные логические функции

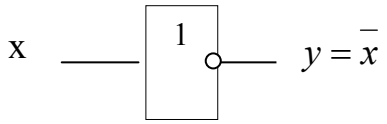
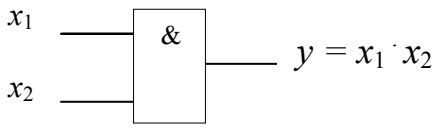
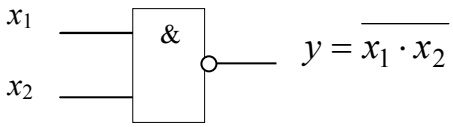
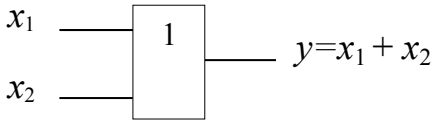
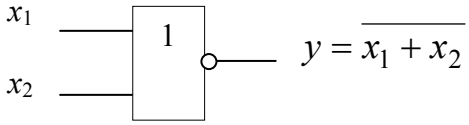
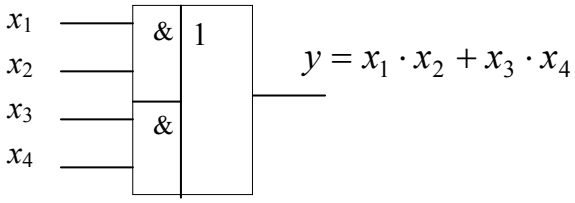
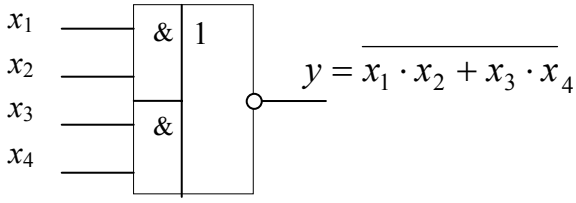
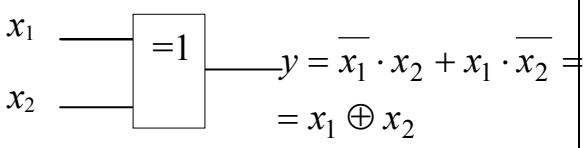
Элемент	Обозначение	Выполняемая функция и схема
НЕ	ЛН	
И	ЛИ	
И-НЕ	ЛА	
ИЛИ	ЛЛ	
ИЛИ-НЕ	ЛЕ	
И-ИЛИ	ЛС	
И-ИЛИ-НЕ	ЛР	
Исключающее ИЛИ	ЛП	

Таблица 2.3. Основные параметры ИМС

Серия ИМС	Потребляемая мощность $P_{пот}$, мВт	Время задержки $t_{зар}$, нс	Максимальная частота f_{max}, МГц	Коэффициент разветвления
155 ТТЛ	10	10,0	35	10
555 ТТЛ	2	9,5	45	20
531 ТТЛ	19	3,0	125	10
1533 ТТЛШ	1	4,0	50	40
1531 ТТЛШ	4	2,0	130	33
176, 564 КМОП	< 1	200	< 2	> 50
1561 КМОП	< 1	250	< 2	> 50

ИМС серии ТТЛ работают при напряжении +5В ± 5%. По сравнению с ИМС серии ТТЛ микросхемы КМОП имеют: малую потребляемую мощность в диапазоне частот до 2 МГц (мощность в статическом режиме не превышает 1 мкВт), большой диапазон значений напряжения питания от 3 до 15 В, очень высокое входное сопротивление (более 1 МОм); большую нагрузочную способность (коэффициент разветвления больше 50). К недостаткам ИМС КМОП относятся: большие значения временных задержек (до 200 нс); повышенное выходное сопротивление (до 1 кОм); значительный разброс параметров. Уровни выходных сигналов зависят от напряжения питания. Уровень логической 1 равен $\approx 0{,}8 \cdot E_{пит.}$, а уровень логического 0 – от 0,3 до 2,5 В.

2.3. Типы выходных каскадов цифровых устройств

Характеристики и параметры входов и выходов цифровых микросхем определяются прежде всего технологией и схемотехникой внутреннего строения микросхем. Но для разработчика цифровых устройств любая микросхема представляет собой всего лишь «черный ящик», внутренности которого знать не обязательно. Ему важно только четко представлять себе, как поведет себя та или иная микросхема в данном конкретном включении, будет ли она правильно выполнять требуемую от нее функцию.

ИМС серий ТТЛ и КМОП различаются типами используемых транзисторов и схемотехническими решениями внутренних каскадов микросхем. Отметим также, что микросхемы КМОП потребляют значительно меньший ток от

источника питания, чем такие же микросхемы ТТЛ (или ТТЛШ), правда, только в статическом режиме или на небольших рабочих частотах. На рис. 2.3 и 2.4 показаны примеры схем входных и выходных каскадов микросхем, выполненных по этим технологиям.

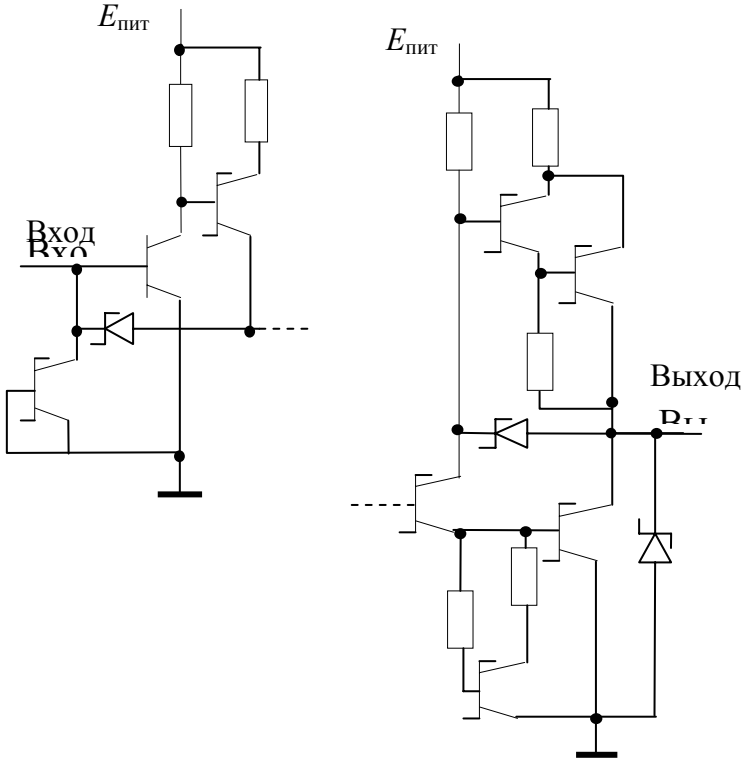


Рис. 2.3. Входной и выходной каскады микросхем ТТЛШ

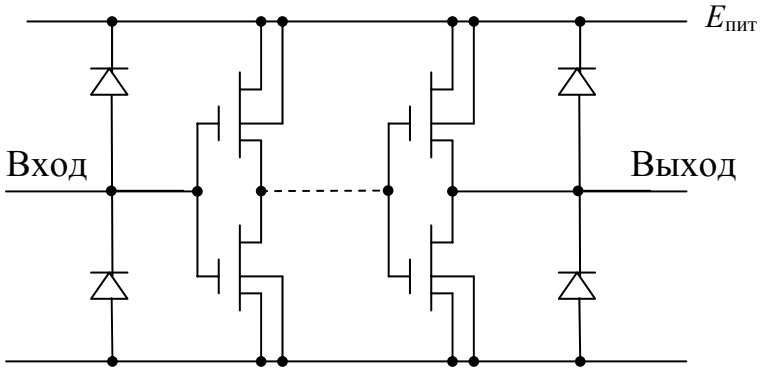


Рис. 2.4. Входной и выходной каскады микросхем КМОП

Понятно, что точный учет всех эффектов в этих схемах, включающих в себя множество транзисторов, диодов и резисторов, крайне сложен, но обычно он просто не нужен разработчику цифровых схем.

В подавляющем большинстве случаев разработчики цифровых схем используют три модели, три уровня представления о работе цифровых устройств:

1) логическая модель;

2) модель с временными задержками;

3) модель с учетом электрических параметров (или электрическая модель).

Опыт показывает, что первая, простейшая модель оказывается достаточной примерно в 20% всех случаев. Она применима для всех цифровых схем, работающих с низкой скоростью, в которых быстродействие не принципиально. Привлечение второй модели, учитывающей задержки срабатывания логических элементов, позволяет охватить около 80% всех возможных схем. Ее применение необходимо для всех быстродействующих устройств и в случае одновременного изменения нескольких входных сигналов. Наконец, добавление третьей модели, учитывающей входные и выходные токи, входные и выходные сопротивления и емкости элементов, позволяет проектировать практически 100% цифровых схем. В первую очередь эту третью модель надо применять при объединении нескольких входов и выходов, при передаче сигналов на большие расстояния и при нетрадиционном включении логических элементов (с переводом их в аналоговый, в линейный режим).

Рассмотрим входы микросхем. На первом уровне представления (логическая модель) и на втором уровне представления (модель с временными задержками) о входах микросхем вообще ничего знать не надо. Вход рассматривается как бесконечно большое сопротивление, никак не влияющее на подключенные к нему выходы. Правда, количество входов, подключаемых к одному выходу, влияет на задержку распространения сигнала, но, как правило, незначительно, поэтому это влияние учитывается редко.

Даже на третьем уровне представления (электрическая модель) в большинстве случаев не надо знать о внутреннем строении микросхемы, о схемотехнике входов. Достаточно считать, что при подаче на вход сигнала логического нуля из этого входа вытекает ток, не превышающий $I^0_{\text{вх}}$, а при подаче сигнала логической единицы в этот вход втекает ток, не превышающий $I^1_{\text{вх}}$. А для правильной логики работы микросхемы достаточно, чтобы уровень напряжения входного сигнала логического нуля был меньше $U^0_{\text{вх}}$, а уровень напряжения входного сигнала логической единицы был больше $U^1_{\text{вх}}$.

Особым случаем является ситуация, когда какой-нибудь вход не подключен ни к одному из выходов, ни к общему проводу, ни к шине питания (так называемый висящий вход). Иногда возможности микросхемы используются не полностью, и на некоторые входы не подаются сигналы. Однако при этом микросхема может не работать или работать нестабильно, так как ее правильное включение подразумевает наличие на всех входах логических уровней, пусть

даже и неизменных. Поэтому рекомендуется подавать на неиспользуемые входы напряжение питания микросхемы $E_{пит}$ или подключать их к общему проводу (земле) в зависимости от того, какой логический уровень необходим на этом входе. Но для некоторых серий микросхем, выполненных по технологии ТТЛ (например, К155 или КР531), неиспользуемые входы надо подключать к шине питания не прямо, а только через резистор величиной около 1 кОм (достаточно одного резистора на 20 входов).

На неподключенных входах микросхем ТТЛ формируется напряжение около 1,5 - 1,6 В, которое иногда называют висячим потенциалом. Обычно этот уровень воспринимается микросхемой как сигнал логической единицы, но надеяться на это не стоит. Потенциал, образующийся на неподключенных входах микросхем КМОП, может восприниматься микросхемой и как логический нуль, и как логическая единица. В любом случае все входы надо куда-то подключать. Неподключенными допускается оставлять только те входы (ТТЛ, а не КМОП), состояние которых в данном включении микросхемы не имеет значения.

Выходы микросхем принципиально отличаются от входов тем, что учет их особенностей необходим даже на первом и втором уровнях представления.

Существуют три разновидности выходных каскадов, существенно различающихся как по своим характеристикам, так и по областям применения:

1) стандартный выход или выход с двумя состояниями (обозначается 2С, 2*S* или, реже, ТТЛ, *TTL*);

2) выход с открытым коллектором (обозначается ОК или *OC*);

3) выход с тремя состояниями или (что то же самое) с возможностью отключения (обозначается ЗС или 3*S*).

Стандартный выход 2С имеет всего два состояния: логический нуль и логическую единицу, причем оба этих состояния активны, то есть выходные токи в обоих этих состояниях ($I^0_{вых}$ и $I^1_{вых}$) могут достигать заметных величин. На первом и втором уровнях представления такой выход можно считать состоящим из двух выключателей, которые замыкаются по очереди (рис. 2.5), причем замкнутому верхнему выключателю соответствует логическая единица на выходе, а замкнутому нижнему – логический нуль.

Выход с открытым коллектором ОК тоже имеет два возможных состояния, но только одно из них (состояние логического нуля) активно, то есть обеспечивает большой втекающий ток $I^0_{вых}$. Второе состояние сводится к тому, что выход полностью отключается от присоединенных к нему входов. Это состояние может использоваться в качестве логической единицы, но для этого между выходом ОК и напряжением питания необходимо подключить нагрузочный резистор *R* (так называемый *pull-up*) величиной порядка сотен Ом. На первом и втором уровнях представления такой выход можно считать состоящим из одно-

го выключателя (рис. 2.5), замкнутому состоянию которого соответствует сигнал логического нуля, а разомкнутому – отключенное, пассивное состояние. Правда, от величины резистора R зависит время переключения выхода из нуля в единицу, что влияет на задержку $t^{0,1}_{\text{здр}}$, но при обычно используемых номиналах резисторов это не слишком важно.

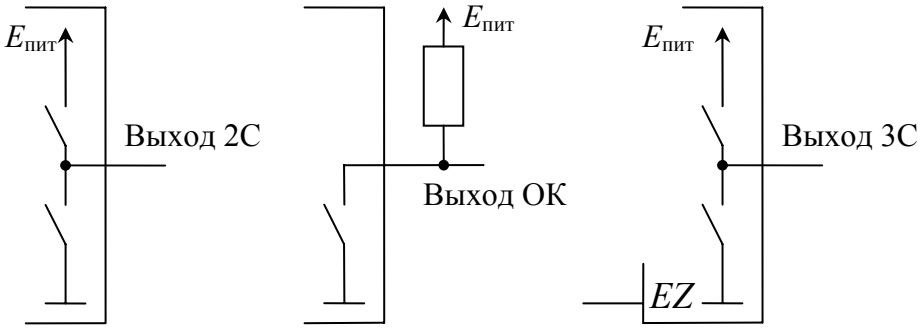


Рис. 2.5. Три типа выходов цифровых микросхем

Наконец, выход с тремя состояниями ЗС очень похож на стандартный выход, но к двум состояниям добавляется еще и третье – пассивное, в котором выход можно считать отключенным от последующей схемы. На первом и втором уровнях представления такой выход можно считать состоящим из двух переключателей (рис. 2.5), которые могут замыкаться по очереди, давая логический нуль и логическую единицу, но могут и размыкаться одновременно. Это третье состояние называется также высокоимпедансным или *Z*-состоянием. Для перевода выхода в третье *Z*-состояние используется специальный управляющий вход, обозначаемый *OE* (*Output Enable* – разрешение выхода) или *EZ* (*Enable Z-state* – разрешение *Z*-состояния, или третьего состояния).

Выходы, имеющие помимо активных состояний еще и пассивное состояние, очень удобны для объединения их между собой. Например, если на один и тот же вход надо по очереди подавать сигналы с двух выходов (рис. 2.6), то выходы 2*С* для этого не подходят, а вот выходы ОК и ЗС подходят.

При объединении двух или более выходов 2С вполне возможна ситуация, при которой один выход стремится выдать сигнал логической единицы, а другой – сигнал логического нуля. Легко заметить, что в этом случае через верхний замкнутый ключ выхода, выдающего единицу, и через нижний замкнутый ключ выхода, выдающего нуль, пойдет недопустимо большой ток короткого замыкания $I_{\text{кз}}$. Это аварийная ситуация, при которой уровень получаемого выходного логического сигнала точно не определен, он может восприниматься

последующим входом и как нуль, и как единица. Конфликтующие выходы могут даже выйти из строя, нарушив работу микросхем и схемы в целом.

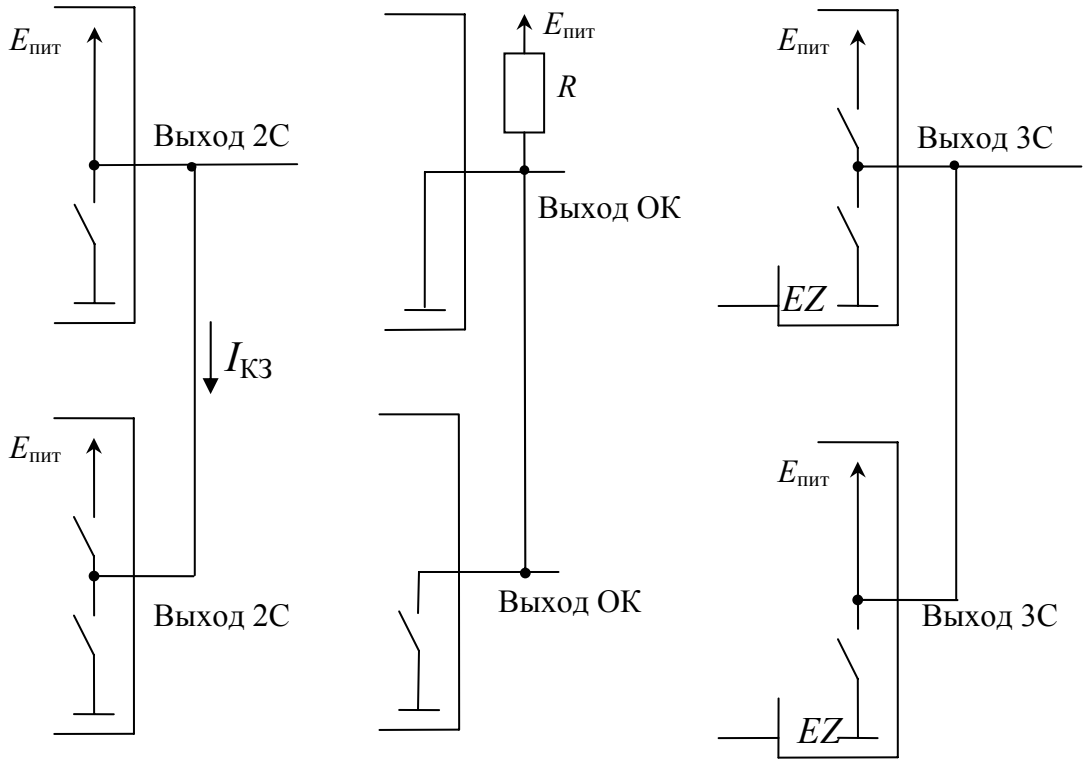


Рис. 2.6. Объединение выходов цифровых микросхем

Зато в случае объединения двух выходов ОК такого конфликта в принципе произойти не может. Даже если ключ одного выхода замкнут, а другого разомкнут, аварийной ситуации не произойдет, так как недопустимо большого тока не будет, а на объединенном выходе будет сигнал логического нуля. А при объединении двух выходов ЗС аварийная ситуация хотя и возможна (если оба выхода одновременно находятся в активном состоянии), но ее легко можно предотвратить, если организовать схему так, что в активном состоянии всегда будет находиться только один из объединенных выходов ЗС.

Объединение выходов цифровых микросхем совершенно необходимо также при шинной (или, как еще говорят, магистральной) организации связей между цифровыми устройствами. Шинная организация связей применяется, например, в компьютерах, других микропроцессорных системах. Суть ее сводится к следующему.

При классической организации связей (рис. 2.7) все сигналы между устройствами передаются по своим отдельным линиям (проводам). Каждое устройство передает свои сигналы всем другим устройствам независимо от других устройств. В этом случае обычно получается очень много линий связи, к тому

же правила обмена сигналами по этим линиям (или протоколы обмена) чрезвычайно разнообразны.

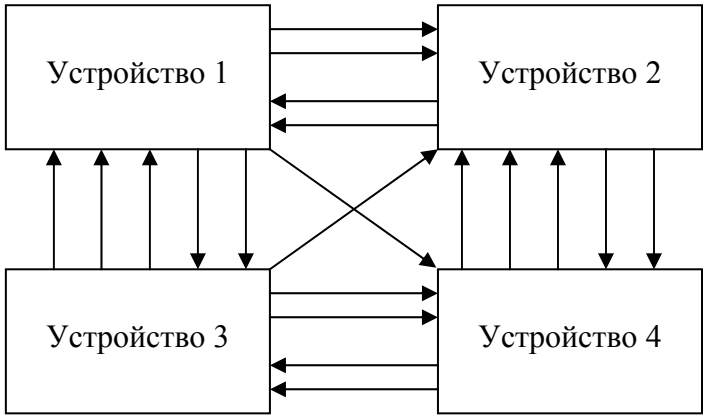


Рис. 2.7. Классическая организация связей

При шинной организации связей (рис. 2.8) все сигналы между устройствами передаются по одним и тем же линиям (проводам), но в разные моменты времени (это называется временным мультиплексированием).

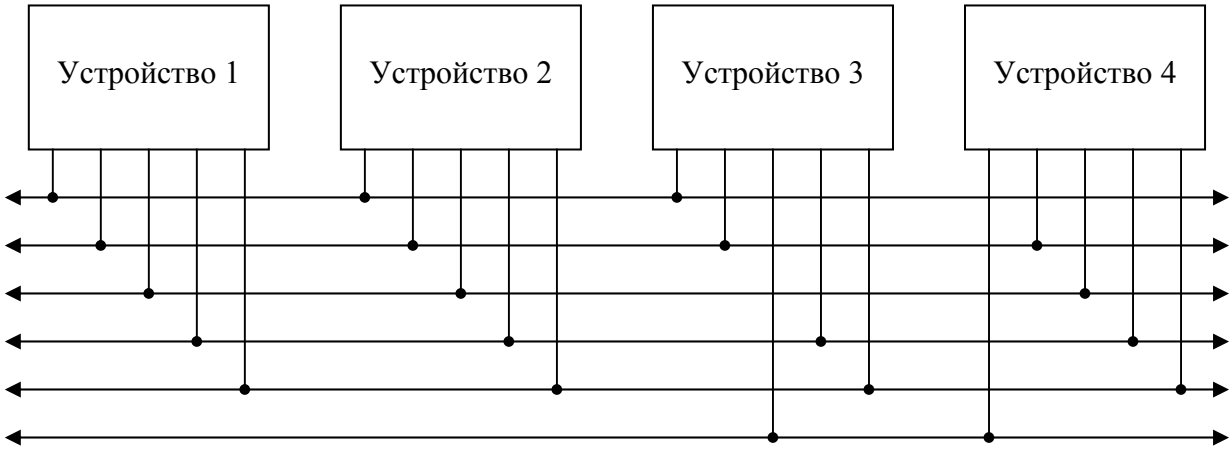


Рис. 2.8. Шинная организация связей

В результате количество линий связи резко сокращается, а правила обмена сигналами существенно упрощаются. Группа линий (сигналов), используемая несколькими устройствами, как раз и называется шиной. Понятно, что объединение выходов в этом случае совершенно необходимо, ведь каждое устройство должно иметь возможность выдавать свой сигнал на общую линию. К недостаткам шинной организации относится прежде всего невысокая (по сравне-

нию с классической структурой связей) скорость обмена сигналами. При простых структурах связи шинная организация может быть избыточна.

На третьем уровне представления (электрическая модель) необходимо уже учитывать, что выходные ключи (рис. 2.5) представляют собой не простые тумблеры (как на первых двух уровнях представления), а транзисторные ключи со своими специфическими параметрами. Однако в большинстве случаев достаточно знать, какой ток может выдать данный выход в состояниях логического нуля ($I^0_{\text{вых}}$) и логической единицы ($I^1_{\text{вых}}$). Величины этих токов не должны превышать суммы токов всех входов, подключенных к данному выходу (соответственно $I^0_{\text{вх}}$ и $I^1_{\text{вх}}$). Количество входов, которое можно подключить к одному выходу, определяет коэффициент разветвления или нагрузочную способность микросхемы. Существуют микросхемы с обычной нагрузочной способностью и с повышенной нагрузочной способностью (больше обычной в два раза и более). Выходы ЗС, как правило, имеют повышенную нагрузочную способность (то есть обеспечивают большие выходные токи). Выходы 2С и ОК могут быть как с обычной, так и с повышенной нагрузочной способностью.

На третьем уровне представления (электрическая модель) необходимо также учитывать выдаваемые выходом микросхемы величины выходных напряжений $U^0_{\text{вых}}$ и $U^1_{\text{вых}}$. Выходы ОК могут быть рассчитаны как на обычное выходное напряжение логической единицы ($U^0_{\text{вых}} = E_{\text{пит}} = 5$ В), так и на повышенное напряжение логической единицы (до 30 В). В последнем случае внешний резистор этого выхода (рис. 2.5) подключается к источнику повышенного напряжения.

При работе с элементами типа ОК проектировщик должен задать сопротивление резистора R, которое не является стандартным, а определяется для конкретных условий. Анализ статических режимов задает ограничения величины сопротивления R снизу и сверху. Значение сопротивления резистора R выбирается в этом диапазоне с учетом быстродействия схемы и потребляемой ею мощности.

Ограничение снизу величины сопротивления резистора R связано с тем, что ее уменьшение может вызвать перегрузку насыщенного транзистора по току. На рис. 2.9, *а* показан режим, в котором нулевое состояние выхода схемы обеспечивается элементом 1 с ОК. Из этого рисунка видно, что через выход элемента 1 протекает суммарный ток, складывающийся из токов резистора, входных токов логических элементов (ЛЭ1,..., ЛЭ$_n$) и токов выходов запертых транзисторов элементов с ОК 2, ..., m, т. е.

$$I^1_{\text{вых}} = I_R + n \cdot I^0_{\text{вх}} + (m-1) \cdot I_z \approx I_R + n \cdot I^0_{\text{вх}},$$

где $I^0_{\text{вх}}$ – входные токи элементов-приемников сигнала при низком уровне входных напряжений; I_z – токи запертых выходов ОК (обычно пренебрежимо малые); $I_R = (E_{\text{пит}} - U^0)/R$. Чтобы ток выхода элемента 1 не превысил допустимого значения $I^0_{\text{вых.max}}$, следует соблюдать следующее условие

$$R \geq (E_{\text{пит}} - U^0)/(I^0_{\text{вых.max}} - n \cdot I^0_{\text{вх.max}}).$$

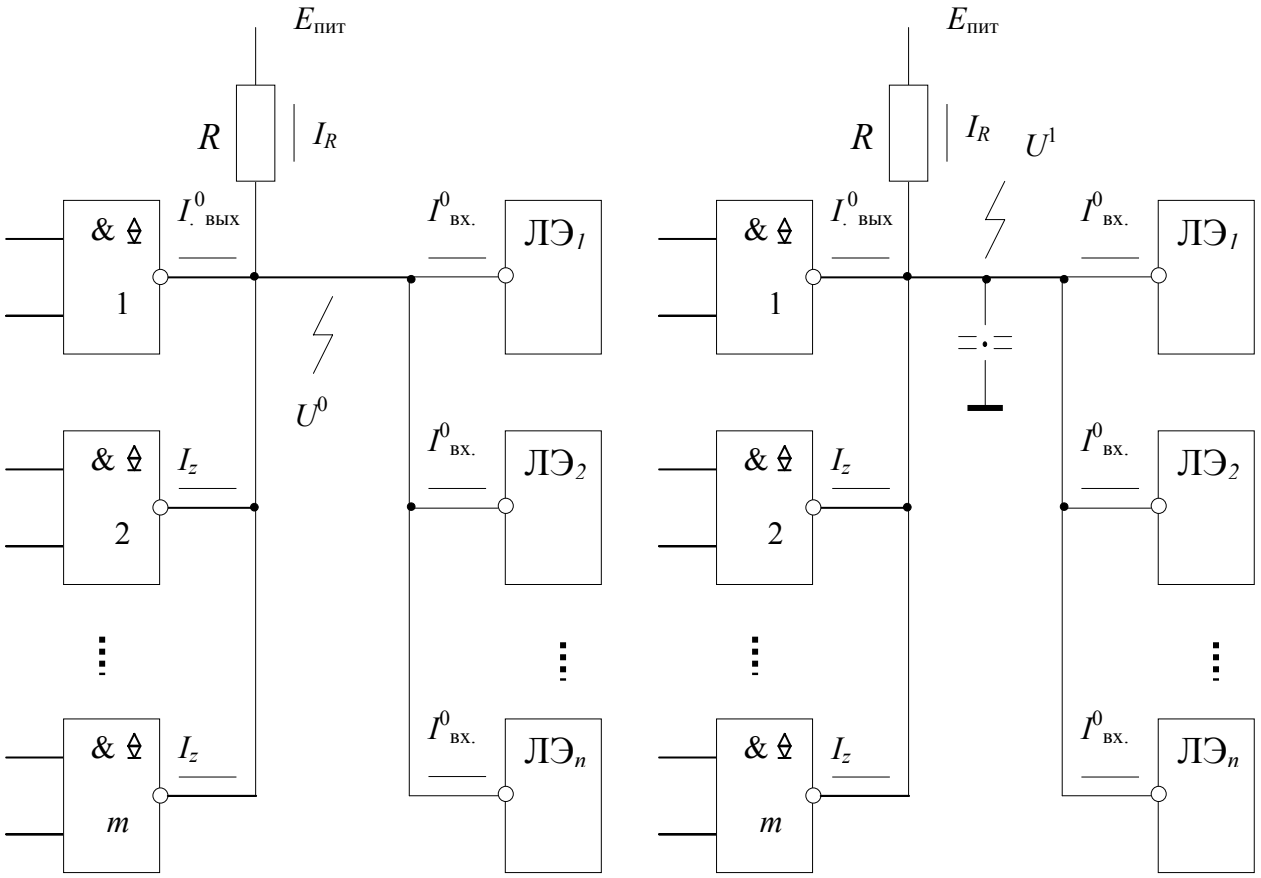


Рис. 2.9. Схемы для расчета минимального (а) и максимального (б) значений сопротивления внешней цепи в каскадах с открытым коллектором

Ограничение сверху величины сопротивления резистора R связано с необходимостью гарантировать достаточно высокий уровень напряжения U^1, формируемого в схеме при запертом состоянии всех выходов элементов с ОК. Из схемы (рис. 2.9б) видно, что:

$$U^1 = E_{\text{пит}} - I_R \cdot R.$$

В то же время $I_R = m \cdot I_z + n \cdot I^1_{\text{вх. max}}$. Из полученных выражений следует

$$R \leq (E_{\text{пит}} - U^1_{\text{вых.min}})/(m \cdot I_z + n \cdot I^1_{\text{вх.max}}),$$

где $U^1_{\text{вых.min}}$ — паспортный параметр элемента.

Имея границы диапазона значений сопротивления резистора R**,** полученные, как показано выше, проектировщик должен выбрать некоторое конкретнее его значение. Выбор вблизи нижней границы улучшает быстродействие схемы, а выбор вблизи верхней уменьшает потребляемую схемой мощность.

Только в сложных случаях, например при переводе логического элемента в линейный режим за счет обратных связей, нужен учет других параметров входных и выходных каскадов. Но в этих редких случаях гораздо проще и надежнее не считать ничего самому, а воспользоваться стандартными схемами включения микросхем или подобрать режимы работы и номиналы внешних элементов (резисторов, конденсаторов) непосредственно на макете проектируемого устройства. В отличие от расчетов такой подход даст полную гарантию работоспособности выбранного решения.

2.4. Основные обозначения на схемах

Для изображения электронных устройств и их узлов применяется три основных типа схем: принципиальная схема; структурная схема; функциональная схема.

Различаются эти три вида схем своим назначением и степенью детализации изображения устройств.

Принципиальная схема – это наиболее подробная схема. Она обязательно показывает все использованные в устройстве элементы и все связи между ними. Если схема строится на основе микросхем, то должны быть показаны номера выводов всех входов и выходов этих микросхем. Принципиальная схема должна позволять полностью воспроизвести устройство. Обозначения принципиальной схемы наиболее жестко стандартизованы, отклонения от стандартов не рекомендуются.

Структурная схема – это наименее подробная схема. Она предназначена для отображения общей структуры устройства, то есть его основных блоков, узлов, частей и главных связей между ними. Из структурной схемы должно быть понятно, зачем нужно данное устройство, и что оно делает в основных режимах работы, как взаимодействуют его части. Обозначения структурной схемы могут быть довольно произвольными, хотя некоторые общепринятые правила все-таки лучше выполнять.

Функциональная схема представляет собой гибрид структурной и принципиальной схем. Некоторые наиболее простые блоки, узлы, части устройства отображаются на ней, как на структурной схеме, а остальные – как на принципиальной схеме. Функциональная схема позволяет понять всю логику работы устройства, все его отличия от других подобных устройств, но не позволяет без дополнительной самостоятельной работы воспроизвести это устройство. Что

касается обозначений, используемых на функциональных схемах, то в части, показанной как структура, они не стандартизованы, а в части, показанной, как принципиальная схема, они стандартизованы.

В технической документации обязательно приводится структурная или функциональная схема, а также обязательно принципиальная схема. В научных статьях и книгах чаще всего ограничиваются структурной или функциональной схемой, приводя принципиальные схемы только некоторых узлов.

А теперь рассмотрим основные обозначения, используемые на схемах. Все узлы, блоки, части, элементы, микросхемы показываются в виде прямоугольников с соответствующими надписями. Все связи между ними, все передаваемые сигналы показываются в виде линий, соединяющих эти прямоугольники. Входы и входы/выходы должны быть расположены на левой стороне прямоугольника, выходы - на правой стороне, хотя это правило часто нарушают, когда необходимо упростить рисунок схемы. Выводы и связи питания, как правило, не показывают, если, конечно, не используются нестандартные включения элементов схемы. Это самые общие правила, касающиеся любых схем.

Введём некоторые определения.

- Положительный сигнал (сигнал положительной полярности) – это сигнал, активный уровень которого – логическая единица, то есть: нуль – это отсутствие сигнала, единица – наличие сигнала (рис. 2.10).

- Отрицательный сигнал (сигнал отрицательной полярности) – это сигнал, активный уровень которого – логический нуль, то есть: единица – это отсутствие сигнала, нуль – наличие сигнала (рис. 2.10).

- Активный уровень сигнала – это уровень, соответствующий приходу сигнала, то есть выполнению этим сигналом соответствующей ему функции.

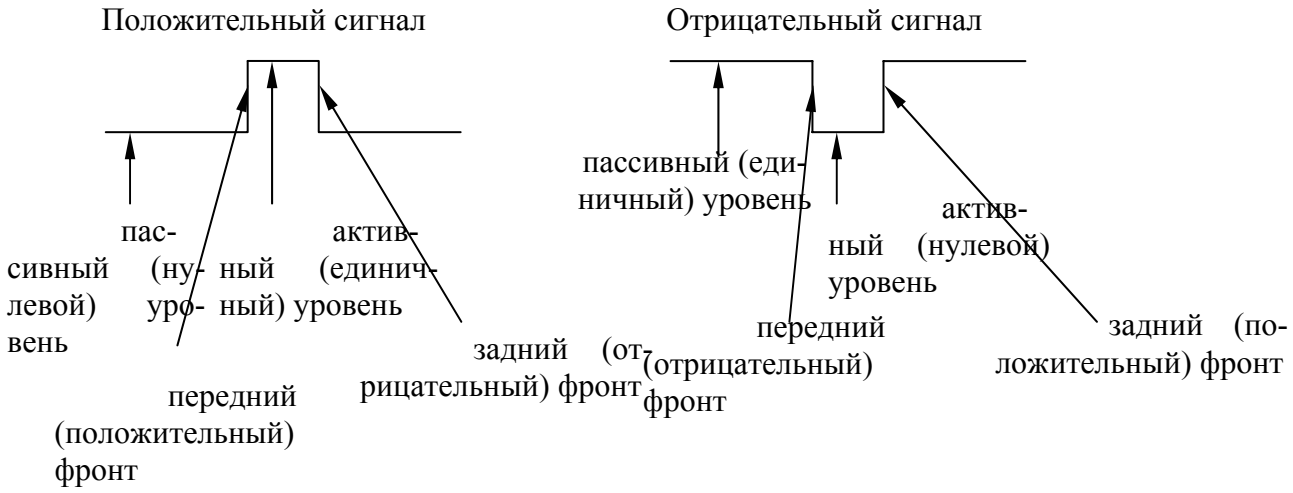


Рис. 2.10. Элементы цифрового сигнала

Пассивный уровень сигнала – это уровень, в котором сигнал не выполняет никакой функции.

Инвертирование или инверсия сигнала – это изменение его полярности.

Инверсный выход – это выход, выдающий сигнал инверсной полярности по сравнению с входным сигналом.

Прямой выход – это выход, выдающий сигнал такой же полярности, какую имеет входной сигнал.

Положительный фронт сигнала – это переход сигнала из нуля в единицу.

Отрицательный фронт сигнала (спад) – это переход сигнала из единицы в нуль.

Передний фронт сигнала – это переход сигнала из пассивного уровня в активный уровень.

Задний фронт сигнала – это переход сигнала из активного уровня в пассивный уровень.

Тактовый сигнал (или строб) – управляющий сигнал, который определяет момент выполнения элементом или узлом его функции.

Шина – группа сигналов (и соответствующих физических линий передачи этих сигналов), объединенных по какому-то принципу. Например, шиной называют сигналы, соответствующие всем разрядам какого-то двоичного кода.

Инверсия сигнала обозначается кружочком на месте входа или выхода Существуют инверсные входы и инверсные выходы (рис. 2.11).

Если какая-то микросхема выполняет функцию по фронту входного сигнала, то на месте входа ставится косая черта (под углом 45°), причем наклон вправо или влево определяется тем, какой фронт – положительный или отрицательный – используется в данном случае (рис. 2.11).

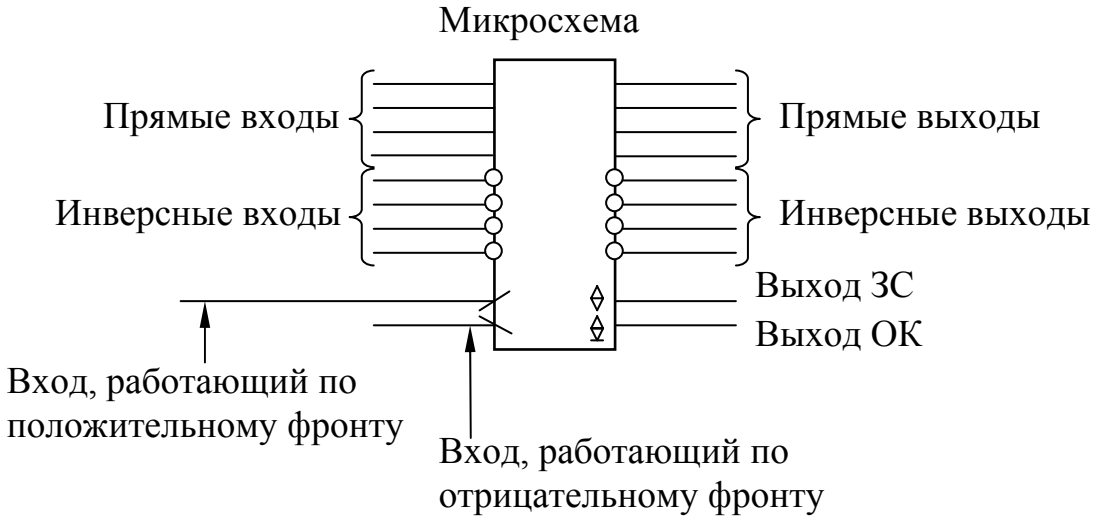


Рис. 2.11. Обозначение входов и выходов

Тип выхода микросхемы помечается специальным значком выход ЗС – перечеркнутым ромбом, а выход ОК – подчеркнутым ромбом (рис. 2.11). Стандартный выход (2С) никак не помечается.

Наконец, если у микросхемы необходимо показать неинформационные выводы, то есть выводы, не являющиеся ни логическими входами, ни логическими выходами, то такой вывод помечается косым крестом (две перпендикулярные линии под углом 45°). Это могут быть, например, выводы для подключения внешних элементов (резисторов, конденсаторов) или выводы питания (рис. 2.12).

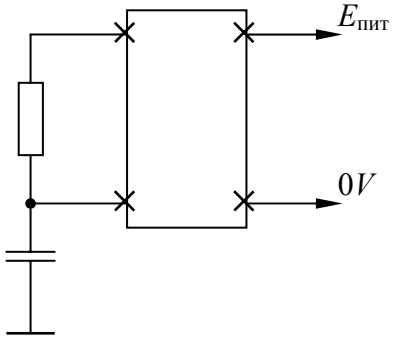


Рис. 2.12. Обозначение неинформационных выводов

В схемах также предусматриваются специальные обозначения для шин (рис. 2.13).

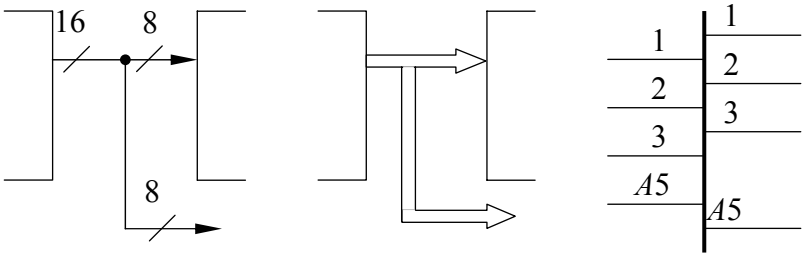


Рис. 2.13. Обозначение шин

Таблица 2.4. Некоторые обозначения сигналов и микросхем

Обозначение	*Название*	*Назначение*
&	And	Элемент И
1	*Exclusive Or*	Элемент Исключающее ИЛИ
1	*Or*	Элемент ИЛИ
A	*Address*	Адресные разряды
BF	*Buffer*	Буфер

Продолжение табл.2.4

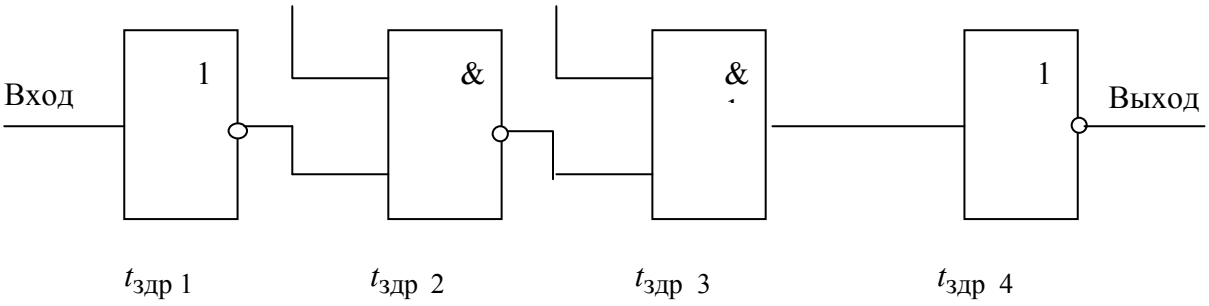
Обозначение	Название	Назначение
CE	*Clock Enable*	Разрешение тактового сигнала
CT	*Counter*	Счетчик
CS	*Chip Select*	Выбор микросхемы
D	*Data*	Разряды данных, данные
EZ	*Enable Z-statc*	Разрешение третьего состояния
G	*Generator*	Генератор
I	*Input*	Вход
I/O	*Input/Output*	Вход/Выход
OE	*Output Enable*	Разрешение выхода
Q	*Quit*	Выход
R	*Reset*	Сброс (установка в нуль)
S	*Set*	Установка в единицу
TC	*Terminal Count*	Окончание счета
Z	*Z-statc*	Третье состояние выхода

На структурных и функциональных схемах шины обозначаются толстыми линиями или двойными стрелками, причем количество сигналов, входящих в шину, указывается рядом с косой чертой, пересекающей шину. На принципиальных схемах шина тоже обозначается толстой линией, а входящие в шину и выходящие из шины сигналы показываются в виде перпендикулярных к шине тонких линий с указанием их номера или названия (рис. 2.13). При передаче по шине двоичного кода нумерация начинается с младшего разряда кода.

2.5. Быстродействие КЦУ

Так как логические элементы, входящие в состав КЦУ, переключаются с задержкой $t_{здр}$, то при изменении в некоторый момент времени комбинации входных сигналов выходные сигналы устройства (если, они изменяются в результате этого) примут установившиеся значения только после того, как закончатся переходные процессы в соответствующих логических элементах. На пути от входов устройства к его выходам отдельные логические элементы включены последовательно. Поэтому длительность переходных процессов будет зависеть от числа логических элементов, которые включены в такой цепочке. Применяемая в настоящее время методика определения $t_{здр}$ в логических элементах, предусматривающая использование при измерении цепочки включенные друг

за другом однотипных логических элементов, позволяет при оценке общей задержки в такой цепочке суммировать задержки отдельных логических элементов.



$$t_{здр} = t_{здр\ 1} + t_{здр\ 2} + t_{здр\ 3} + t_{здр\ 4}$$

Рис. 2.14. Суммирование задержек элементов

При оценке быстродействия КЦУ необходимо выявить ту цепочку логических элементов между входами и выходами устройства, которая будет задавать наибольшую задержку, и сложить между собой задержки логических элементов этой цепочки. Обычно она содержит наибольшее число включенных друг за другом от входов до выходов логических элементов. Пример расчета временных задержек элементов на пути прохождения сигналов на рис.2.14. Но могут быть исключения, связанные, например, с наличием в более короткой цепочке отдельных инерционных логических элементов с большим $t_{здр}$. Поэтому в общем случае необходимо проанализировать все цепочки логических элементов от входов до выходов и выявить такую, которая дает наибольшую задержку.

Рассмотрим пример синтеза КЦУ. Синтезировать на логических элементах И-НЕ КЦУ, имеющее четыре входных сигнала x_0, x_1, x_2, x_3 и один выходной y, заданное таблицей истинности (табл. 2.5). Найти максимальную задержку распространения сигнала $t_{здр}$ считая, что задержки во всех логических элементах одинаковы и составляют $t_э$.

Таблица 2.5. Таблица истинности заданной функции

Номер набора	$x_3\ x_2\ x_1\ x_0$	y	Номер набора	$x_3\ x_2\ x_1\ x_0$	y
0	0 0 0 0	0	8	1 0 0 0	1
1	0 0 0 1	1	9	1 0 0 1	1
2	0 0 1 0	1	10	1 0 1 0	1
3	0 0 1 1	0	11	1 0 1 1	1
4	0 1 0 0	0	12	1 1 0 0	1
5	0 1 0 1	0	13	1 1 0 1	1
6	0 1 1 0	0	14	1 1 1 0	1
7	0 1 1 1	1	15	1 1 1 1	1

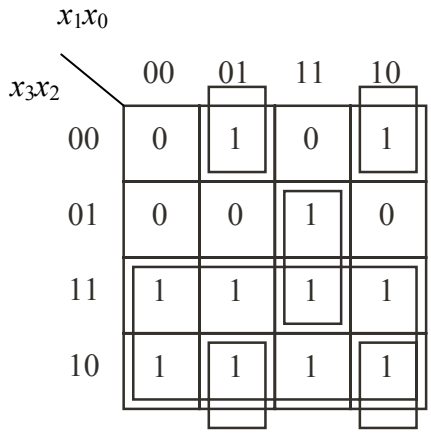


Рис. 2.15. Карта Карно заданной функции

Соответствующая карта Карно для рассматриваемого КЦУ приведена на рис. 2.15. На ее основе получаем минимизированную функцию

$$y = x_3 \vee \bar{x}_2 \cdot \bar{x}_1 \cdot x_0 \vee \bar{x}_2 \cdot x_1 \cdot \bar{x}_0 \vee x_2 \cdot x_1 \cdot x_0 .$$

Если поставить два знака инверсии над правой частью функции, то с помощью правила де Моргана получим

$$y = \overline{\bar{x}_3 \cdot \overline{\bar{x}_2 \cdot x_0} \cdot \overline{x_1 \cdot x_0} \cdot \overline{x_1 \cdot x_1} \cdot \overline{x_0} \cdot \overline{x_2 \cdot x_1 \cdot x_0}}$$

Логическая схема и временные диаграммы на основе этой формулы приведены на рис. 2.16.

Цепочки логических элементов от входов до выхода содержат от двух до пяти элементов. В данном случае небольшая задержка в $5t_{здр}$ связана с прохождением сигнала от входов до выхода по самой длинной цепочке

На временной диаграмме рассмотрено прохождение сигнала в схеме, когда на входе код 0001 изменяется на код 0011. Для упрощения картины переходных процессов предполагается, что сигналы в каждом элементе имеют одинаковую задержку $t_{здр}$, а сами сигналы прямоугольные. Задержка прохождения сигнала снижает быстродействие КЦУ, так как во многих случаях очередное изменение входных сигналов может быть допущено лишь после завершения переходных процессов и фиксации на выходе установившихся значений сигналов

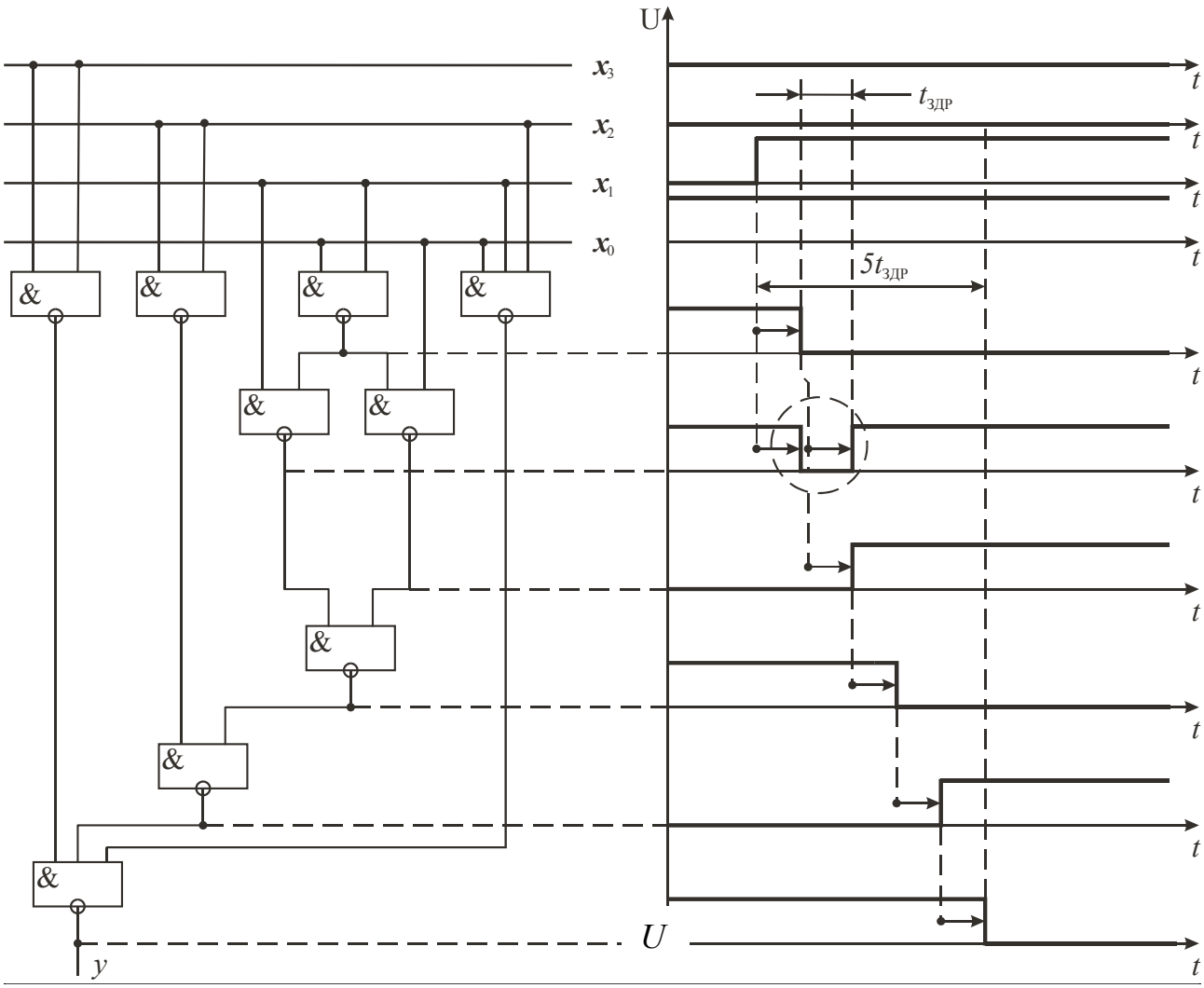


Рис. 2.16. Логическая схема и временные диаграммы синтезируемого КЦУ

2.6. Состязания в КЦУ

Неодинаковую задержку прохождения сигнала в отдельных частях КЦУ иногда ассоциируют с «состязаниями» в скорости переключения логических элементов. В результате этого явления на выходах некоторых элементов могут появляться импульсные помехи. На рис. 2.16 штриховым кружком отмечена импульсная помеха на выходе одного из логических элементов. В данном случае это не привело к появлению помехи на выходе КЦУ. В других случаях неодинаковая задержка прохождения сигнала в отдельных цепочках логических элементов может привести к импульсной помехе и на выходе КЦУ. Покажем это на примере. Рассмотрим временные диаграммы в КЦУ, заданном структурной формулой:

$$y = \overline{x_2 \cdot \overline{x_0 \cdot \overline{x_3 \cdot \overline{x_1 \cdot x_1}}}}.$$

Входной сигнал изменяется с кода 1000 на код 1111. Задержка в каждом логическом элементе $t_{здр}$. Логическая схема КЦУ и временные диаграммы для данного примера приведены на рис. 2.17. В данном случае на выходе КЦУ при изменении кода входного сигнала 1000 на код 1111 кратковременно устанавливается нулевое значение (на время $3t_{здр}$), хотя по условиям работы в соответствии с полученной функцией выходное значение должно было сохраниться на единичном уровне ($y = 1$).

Это непосредственно вытекает из временных диаграмм, если пренебречь задержками переключения логических элементов. Когда длительность помехи превысит определенную величину, возможно нарушение работоспособности подключенных к выходу КЦУ устройств (триггеров, счетчиков и т. д.). В подобных случаях «состязания» называют «опасными». Для борьбы с «опасными состязаниями» можно принять различные меры. Во-первых, иногда можно предусмотреть такой порядок смены входных кодов, при котором либо импульсные помехи не будут появляться на выходе вообще, либо их длительность уменьшится до безопасной величины.

Можно показать, что в рассмотренном на рис. 2.17 примере задержка момента перехода в единичное состояние сигнала x_3 по отношению к сигналам x_1 и x_2 приводит к пропорциональному уменьшению длительности помехи на выходе. В необходимых случаях такую задержку можно ввести искусственно включением цепочки из четного числа инверторов. Во-вторых, при синтезе соответствующего КЦУ можно, иногда найти такой вариант структурной формулы (а следовательно, и структурной схемы), при котором удается уменьшить величину импульсной помехи. Например, выполняя тождественные преобразования полученной функции, можно получить ее другой вариант:

$$y = \overline{x_2 \cdot x_0 \cdot \overline{x_3 \cdot x_3} \cdot \overline{x_1 \cdot x_0}}.$$

Можно показать (соответствующие временные диаграммы предлагается построить самостоятельно), что при смене входного кода 1000 на код 1111 импульсная помеха по выходе уменьшится до $t_{здр}$. В-третьих, можно организовать синхронную передачу сигналов от одного устройства к другому. Для этого вводятся специальные импульсы синхронизации, которые задают моменты передачи информации между отдельными устройствами.

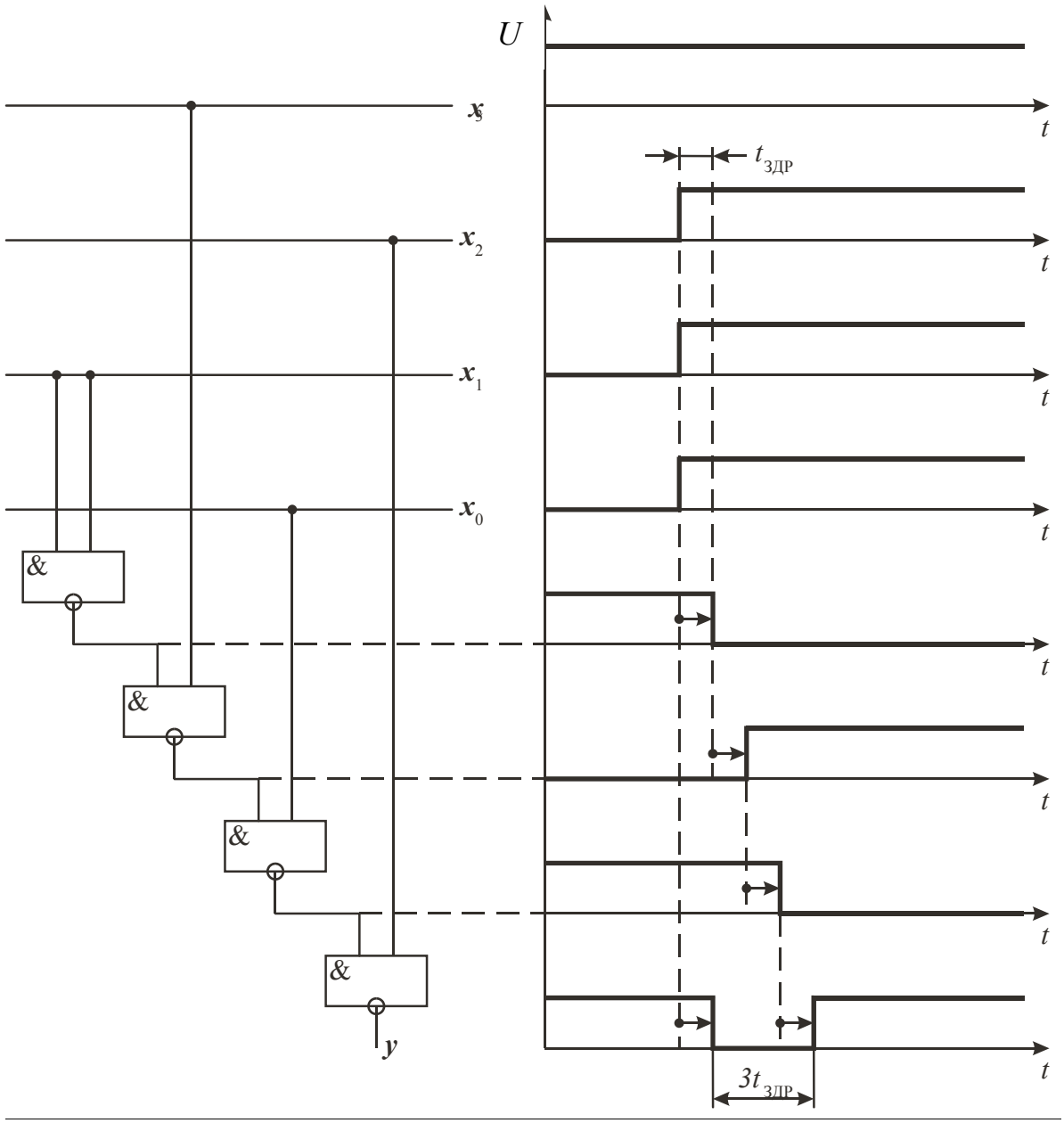


Рис. 2.17. Логическая схема и временные диаграммы КЦУ

Пауза между импульсами синхронизации может быть выбрана такой, чтобы за ее время закончились переходные процессы и на выходах устройств установились стационарные значения. Синхронная передача информации в современных цифровых устройствах применяется очень широко. Этот вопрос уже обсуждался ранее, например при рассмотрении типовых КЦУ, в том числе мультиплексоров. Отметим, что наличие «состязаний» и возникающие при этом импульсные помехи влияют на надежность цифровых устройств.

2.7. Надёжность КЦУ

Под надежностью КЦУ, как и вообще любого технического устройства, понимается свойство выполнять заданные функции в определенных условиях работы. Нарушение нормального функционирования может быть постоянным, когда происходит частичная или полная утрата работоспособности, тогда его называют отказом. В других случаях нарушение носит кратковременный характер (например, из-за нарушения в какой-то момент функционирования под действием импульсной помехи), тогда его называют сбоем. Показателями надежности для указанных случаев чаще всего служит средняя наработка устройства соответственно на один отказ или сбой. Эти показатели могут быть рассчитаны известными методами, если заданы показатели надежности отдельных компонентов, входящих в состав устройства. Каждое КЦУ состоит из некоторого числа интегральных микросхем, соединенных друг с другом заданным образом на печатной плате. Зная показатели надежности отдельных микросхем, характер их соединений друг с другом точки зрения надежности (последовательное или параллельное соединение), показатели надежности составных частей печатной платы (точек присоединения микросхем к плате, печатных проводников), можно рассчитать надежность КЦУ. Для повышения надежности КЦУ можно ввести дополнительные логические элементы, с помощью которых обнаруживаются, а иногда и устраняются ошибочные значения выходных сигналов. С этой целью широко используют особенности сигналов конкретных КЦУ. Например, выходной код дешифратора с единичными активными значениями не может одновременно иметь больше одного единичного разряда. Можно ввести элементы, обнаруживающие ошибочное появление единичных значений в двух и более разрядах, что позволит повысить надежность работы дешифратора. Для устранения сбоев в работе КЦУ необходимо принимать меры, препятствующие возникновению «опасных состязаний». На основе КЦУ часто создаются схемы, обеспечивающие повышение надежности функционирования различных цифровых устройств (например, схемы проверки передаваемых кодов на четность).

2.8. Вопросы и упражнения

1.Чем отличается комбинационное цифровое устройство от последовательностного?

2.Что такое коэффициэнт разветвления по выходу логического элемента?

3.Что такое коэффициент объединения по входу логического элемента?

4.Какие временные параметры характеризуют быстродействие логических элементов?

5.Поясните особенности входных и выходных каскадов микросхем.

6.Дайте определение структурной, функциональной, принципиальной схем.

7.Какова последовательность синтеза комбинационных цифровых устройств?

8.Как оценивается быстродействие комбинационных цифровых устройств?

9.В чем проявляются состязания комбинационных цифровых устройств?

10.Реализуйте в базисе И-НЕ логическую функцию: $f(x_0, x_1, x_2, x_3) = x_0 \cdot x_1 \vee x_2 \cdot x_3 \vee \bar{x}_1(x_0 \cdot x_3 \vee x_2 \cdot \bar{x}_3)$. Найдите максимальную задержку распространения сигналов $t_{\text{здр}}$.

11.Реализуйте в базисе ИЛИ-НЕ логическую функцию: $f(x_0, x_1, x_2, x_3) = x_2 \vee \bar{x}_3 \vee \bar{x}_0 \cdot x_1 \vee x_2 \cdot x_3$. Найдите максимальную задержку распространения сигналов $t_{\text{здр.}}$

3. ТИПОВЫЕ КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

3.1. Преобразователи кодов

На основе логических элементов можно строить цифровые комбинационные узлы электронных устройств любой сложности, реализуемые на ИМС малой степени интеграции. Однако выпускаются типовые (стандартные) комбинационные устройства, выполняемые на одном кристалле ИМС средней степени интеграции, на которых можно строить интересующие нас устройства.

На входы таких устройств могут подаваться информационные сигналы и сигналы управления. Сигналы управления могут определить, например, порядок передачи информационных входных сигналов на выход или играть роль сигналов синхронизации.

Во многих случаях, особенно при использовании в устройствах выходных цепей с тремя состояниями, в качестве сигналов синхронизации выступают сигналы “Выбор микросхемы” (*CS*). Наличие активного значения такого сигнала управления (логический нуль или логическая единица) разрешает устройству выполнение заданных функций, отсутствие его переводит в “неактивное” состояние, при котором устройство не выполняет обработку информации, а его выходы отключены от нагрузки.

В настоящее время находят применение такие типовые комбинационные устройства, выполненные на одном кристалле, как преобразователи кодов, дешифраторы, шифраторы, мультиплексоры, демультиплексоры, цифровые компараторы, арифметические устройства и ряд других менее распространенных схем. Используя методы и особенности логического синтеза цифровых устройств, описанные ранее в главах 1 и 2, рассмотрим функционирование типовых комбинационных устройств.

Преобразователи кодов. Комбинационные преобразователи кодов предназначены для преобразования n – элементного входного кода в m – элементный код на выходе. Задача преобразования кодов возникает в результате того, что отдельные узлы цифровых устройств могут работать в различных кодах и необходимо иметь связующее звено (преобразователь кодов). Связь между входными и выходными сигналами можно задать таблицами истинности или логическими функциями. Преобразователи кодов легко синтезируются методами логического проектирования комбинационных устройств. Для наиболее распространенных операций выпускаются интегральные микросхемы преобразователей кодов: преобразователи двоично-десятичного в двоичный код; преобразователи двоичного в двоично-десятичный код; преобразователи двоично-

го кода в код Грея; преобразователи двоичного или двоично-десятичного кода в код семисегментного индикатора и др.

В качестве примера рассмотрим преобразователь двоичного кода в код управления семисегментным цифровым индикатором, приведенный на рис. 3.1.

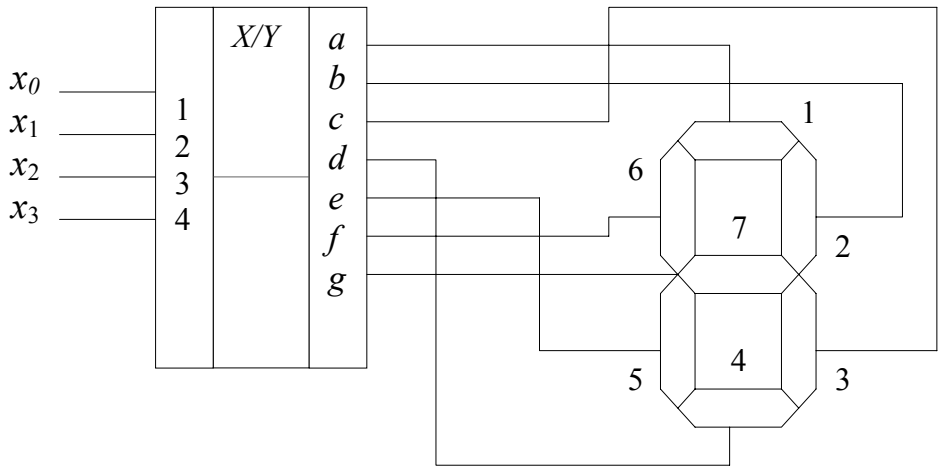


Рис. 3.1. Схема преобразователя кода для семисегментного индикатора

Сам индикатор представляет собой полупроводниковый прибор, в котором имеются семь сегментов, выполненных из светодиодов. Включением и выключением отдельных сегментов можно получить светящееся изображение отдельных цифр или знаков. Конфигурация и расположение сегментов индикатора показаны на рис. 3.1. Каждой цифре соответствует свой набор включения определенных сегментов индикатора. Закон функционирования преобразователя двоично-десятичного кода в код семисегментного индикатора показан в табл 3.1.

Такие индикаторы позволяют получить светящееся изображение не только цифр от 0 до 9, но других знаков, используемых в шестнадцатеричной системе счисления. Для управления такими индикаторами выпускаются интегральные микросхемы следующих типов: КР514ИД1, К514ИД2, 176ИД2, 176ИД3, 564ИД4, 564ИД5 и др. Преобразователи кодов, выполненные по технологии КМОП, можно использовать не только со светодиодными индикаторами, но и с жидкокристаллическими или катодолюминесцентными.

Примерами простейших преобразователей кодов, которые широко применяются в цифровых устройствах, являются шифраторы и дешифраторы.

Таблица 3.1. Закон функционирования преобразователя

десятичная цифра	входной код				сегменты						
	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
	8	4	2	1	1	2	3	4	5	6	7
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

3.2. Шифраторы и дешифраторы

Шифратор. Шифратором называют кодовый преобразователь, который имеет n входов и k выходов, и при подаче сигнала на один из входов (входным кодом является унитарный код “1 из m”) на выходах появляется двоичный код возбужденного входа. Очевидно, что число выходов и входов в полном шифраторе связано соотношением $n = 2^k$. В условном схематическом обозначении шифратора используются буквы *CD* от слов *coder*.

Рассмотрим принцип построения шифратора на примере преобразования унитарного кода “1 из 8” в двоичный код. Состояния такого шифратора приведены в табл. 3.2., а схема и условное графическое обозначение – на рис. 3.2. Если все входные сигналы имеют нулевое значение, то на выходе шифратора будет нулевой код 000.

Младший выход, т.е. выход с весовым коэффициентом, равным 1, должен возбуждаться при входном сигнале на любом из нечетных входов, так как все нечетные номера в двоичном представлении содержат единицу в младшем разряде. Следовательно, младший выход — это выход логического элемента ИЛИ, к входам которого подключены все входы с нечетными номерами. Следующий выход имеет вес два. Он должен возбуждаться при подаче сигналов на входы с номерами 2, 3, 6, 7, т.е. с номерами, имеющими в двоичном представлении единицу во втором разряде. Таким образом, входы логического элемента ИЛИ должны быть подключены к входным сигналам, имеющим указанные номера.

Старший разряд выходного двоичного кода формируется из входных сигналов с номерами 4, 5, 6 и 7, имеющими в двоичном представлении единицу в одном из четырех старших разрядов унитарного кода.

Таблица 3.2. Состояния входов и выходов шифратора 8x3

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

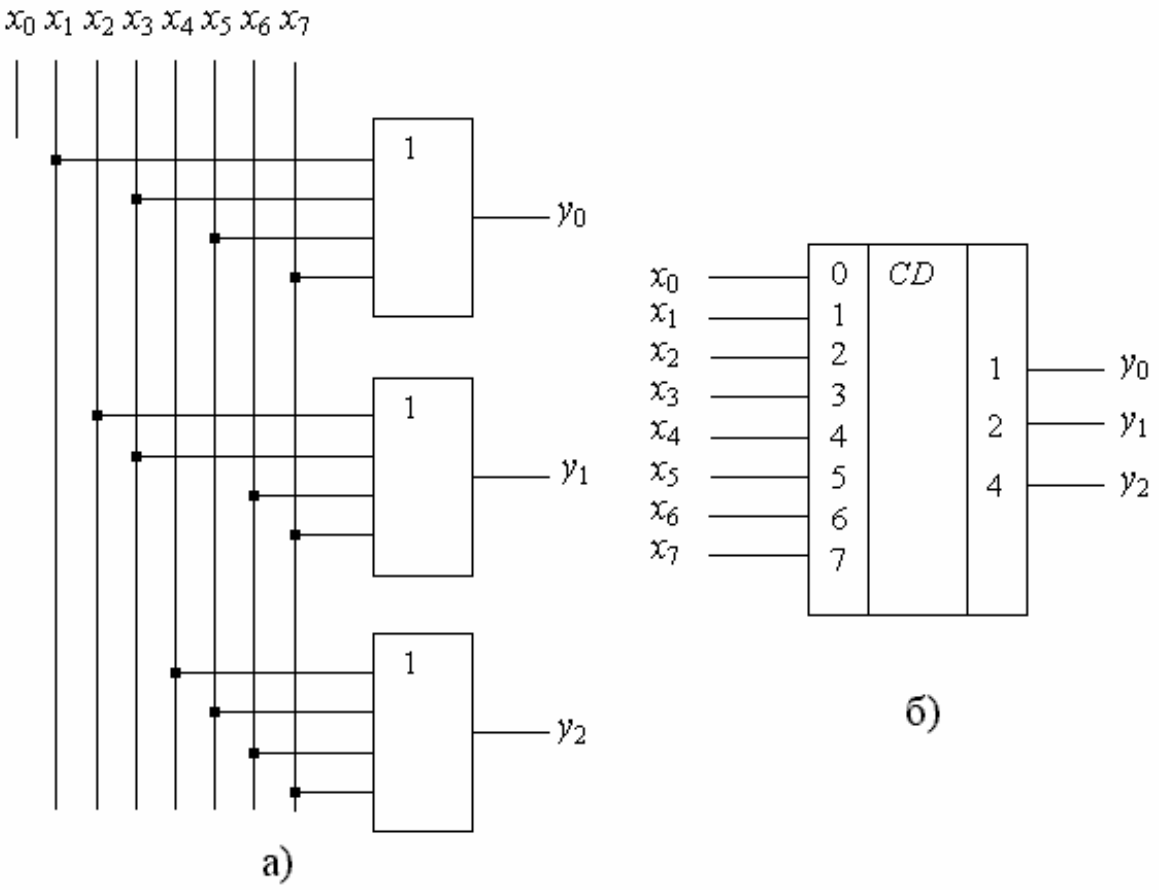


Рис. 3.2. Схема шифратора 8x3 (а) и его условное графическое обозначение (б)

Как показывает рассмотренный пример, при помощи шифратора можно сократить (сжать) информацию для передачи ее по меньшему числу линий связи, так как $k<n$. Обратное преобразование, т. е. восстановление информации в первоначальный вид можно выполнить с помощью дешифратора. Очевидно, что максимальное число входов шифратора не может превышать количество возможных комбинаций выходных сигналов, т.е. необходимо выполнение условия $n \le 2^k$.

Наибольшее применение шифратор находит применение в устройствах ввода информации. Так, например, в кнопочных пультах управления ввод числовых данных обычно выполняется в унитарном коде посредством нажатия одной из десяти кнопок, а данные вводятся в микропроцессор в двоичном коде.

Если сигналы, поступающие на вход шифратора, являются независимыми, что бывает, например, при нажатии одновременно нескольких кнопок на кнопочном пульте управления, то в этом случае каждому входу x шифратора назначают свой приоритет. Обычно считают, что чем выше номер входа, тем выше его приоритет. В этом случае шифратор должен выдавать на выходе двоичный код входа с большим номером. Например, если на входе шифратора установлен код 0011, то на выходе будет код 01. Такие шифраторы называются приоритетными. В связи со специфической областью применения шифраторов имеется небольшой выбор микросхем шифраторов в стандартных сериях. В отечественных сериях шифраторы имеют в названии буквы ИВ.

Таблица 3.3. Состояния входов и выходов шифратора К555ИВ1

$\overline{E}$	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y_2	y_1	y_0	$\overline{G}$	$\overline{E}_o$
1	*	*	*	*	*	*	*	*	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	*	0	0	1	0	1
0	0	0	0	0	0	1	*	*	0	1	0	0	1
0	0	0	0	0	1	*	*	*	0	1	1	0	1
0	0	0	0	1	*	*	*	*	1	0	0	0	1
0	0	0	1	*	*	*	*	*	1	0	1	0	1
0	0	1	*	*	*	*	*	*	1	1	0	0	1
0	1	*	*	*	*	*	*	*	1	1	1	0	1

Примечание: * = 0 или 1.

Рассмотрим пример функционирования приоритетного шифратора К555ИВ1, выполняющего преобразование унитарного кода “1 из 8” в трехзнач-

ный двоичный код. Состояние входов и выходов шифратора приведены в табл. 3.3.

Условное графическое обозначение шифратора К555ИВ1 приведено на рис. 3.3.

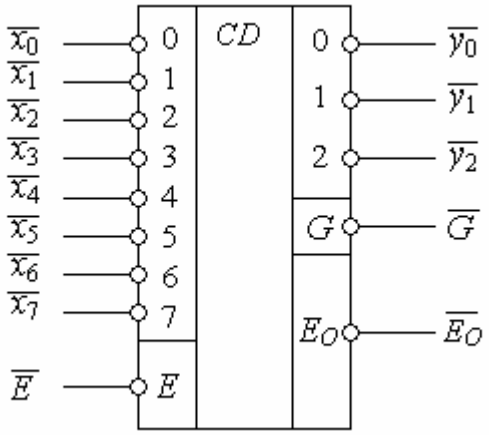


Рис. 3.3. Условное графическое обозначение шифратора К555ИВ1

Назначение сигналов на входе шифратора: $\overline{E}$ – сигнал стробирования шифратора (0 – активен, 1 – неактивен). Сигналы на выходе: $\overline{G}$ – сигнал, свидетельствующий о наличии хотя бы одного возбужденного входа x_i, при включенном состоянии шифратора ($\overline{G}$ = 0 при x_i = 1, хотя бы для одного i при $\overline{E}$ = 0); $\overline{E}_{\text{O}}$ – сигнал разрешения, свидетельствующий об отсутствии возбужденных входов x_i при включенном состоянии шифратора ($\overline{E}_{\text{O}}$ = 0 при $\overline{E}$ = 0 и x_i = 0 для i = 0, … , 7).

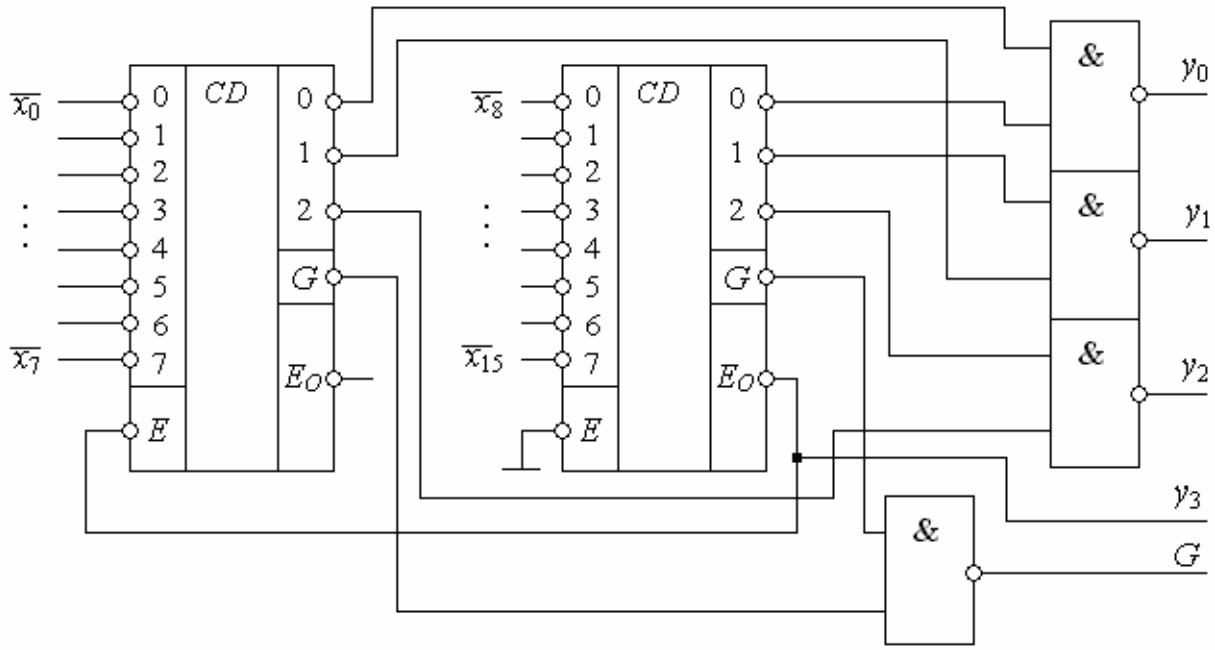


Рис. 3.4. Схема расширенного шифратора 16×4

Таким образом, трехразрядный двоичный код можно считывать с выхода шифратора только при условии $\overline{G} = 0$. Выходной сигнал $\overline{E}_O$ можно использовать при каскадном включении шифраторов. Схема расширенного шифратора 16х4 на микросхемах К555ИВ1 приведена на рис.3.4.

Стробированием называется выделение сигнала в определенный момент времени. В данном случае это – появление выходного сигнала в момент, когда на входе стробирования есть разрешающий уровень $\overline{E} = 0$. На выходе G этой микросхемы появляется сигнал (с нулевым уровнем) при подаче единичного сигнала на любой из входов $x_0, x_1, \ldots, x_7$. Наивысший приоритет имеет вход x_{15}. Первый шифратор включается только в том случае, если не возбужден ни один вход второго шифратора. Сигнал $\overline{G} = 0$, если возбужден хотя бы один вход из $x_0, \ldots, x_{15}$.

Аналогично функционирует приоритетный шифратор К555ИВ2, отличительной особенностью которого является наличие выходов с тремя состояниями, что облегчает каскадирование шифраторов.

Некоторые микросхемы шифраторов перечислены в табл. 3.4.

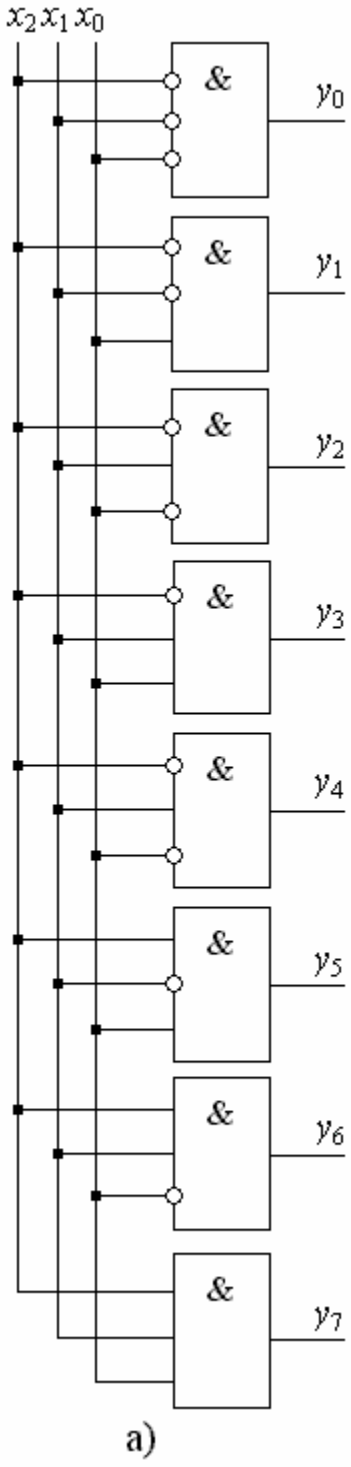
Дешифратор. Дешифратором называют преобразователь двоичного n-разрядного кода в унитарный n-разрядный код, все разряды которого, за исключением одного, равны нулю. Другими словами, дешифратор при наличии определенной комбинации на входах обеспечивает появление логической единицы на одном из его выходов, то есть преобразует двоичный код в номер выходного сигнала – дешифрирует код. В условном графическом обозначении дешифратор отображается буквами *DC* от слова *decoder*. Дешифраторы бывают полные и неполные. Для полного дешифратора выполняется условие $N = 2^n$, где $n -$ число входов, а $N -$ число выходов.

В неполных дешифраторах имеется n входов, но реализуется $N < 2^n$ выходов. Так, например, дешифратор, имеющий 4 входа и 10 выходов будет неполным, а дешифратор, имеющий 2 входа и 4 выхода, будет полным.

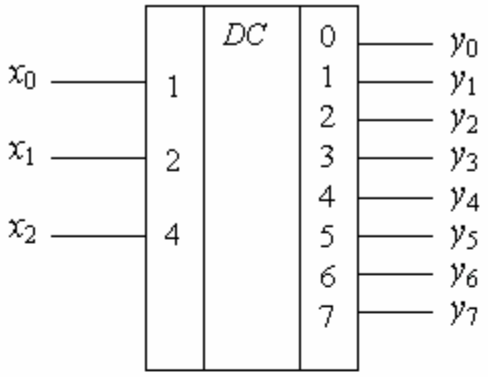
Рассмотрим принцип построения дешифратора на примере преобразования трехразрядного двоичного кода в унитарный код “1 из 8”. Если считать, что входы и выходы упорядочены по возрастающим номерам, т. е. считать, что коду 000 ($\overline{x}_2, \overline{x}_1, \overline{x}_0$) соответствует выход y_0, коду 001($\overline{x}_2, \overline{x}_1, x_0$) – выход y_1 и т.д., то для полного дешифратора можно записать восемь логических функций, которые являются минтермами входных аргументов:

$$y_0 = \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0; \qquad y_4 = x_2 \cdot \bar{x}_1 \cdot \bar{x}_0;$$
$$y_1 = \bar{x}_2 \cdot \bar{x}_1 \cdot x_0; \qquad y_5 = x_2 \cdot \bar{x}_1 \cdot x_0;$$
$$y_2 = \bar{x}_2 \cdot x_1 \cdot \bar{x}_0; \qquad y_6 = x_2 \cdot x_1 \cdot \bar{x}_0;$$
$$y_3 = \bar{x}_2 \cdot x_1 \cdot x_0; \qquad y_7 = x_2 \cdot x_1 \cdot x_0.$$

Реализовать эти восемь логических функций можно с помощью восьми трехвходовых логических элементов И.



а)



б)

S	x_2	x_1	x_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

в)

Рис. 3.5. Схема дешифратора 3×8 (а), его графическое обозначение (б) и таблица истинности (в)

Полученная схема дешифратора, его условное графическое обозначение и таблица истинности приведены на рис. 3.5.

Для увеличения разрядности используют каскадное включение дешифраторов. На рис. 3.6 показана схема дешифратора 4х16, включающая пять дешифраторов 2х4, соединенных последовательно в два каскада. Все дешифраторы одинаковые. Кроме кодовых входов каждый дешифратор имеет вход стробирующего сигнала, разрешающего выработку выходных сигналов (вход C). Сигнал на выходе дешифратора появляется только при $C = 1$. Если разрешающий сигнал отсутствует $C = 0$, то на всех выходах дешифратора устанавливаются сигналы логического нуля независимо от сигнала на кодовых входах, т. е. дешифратор запирается.

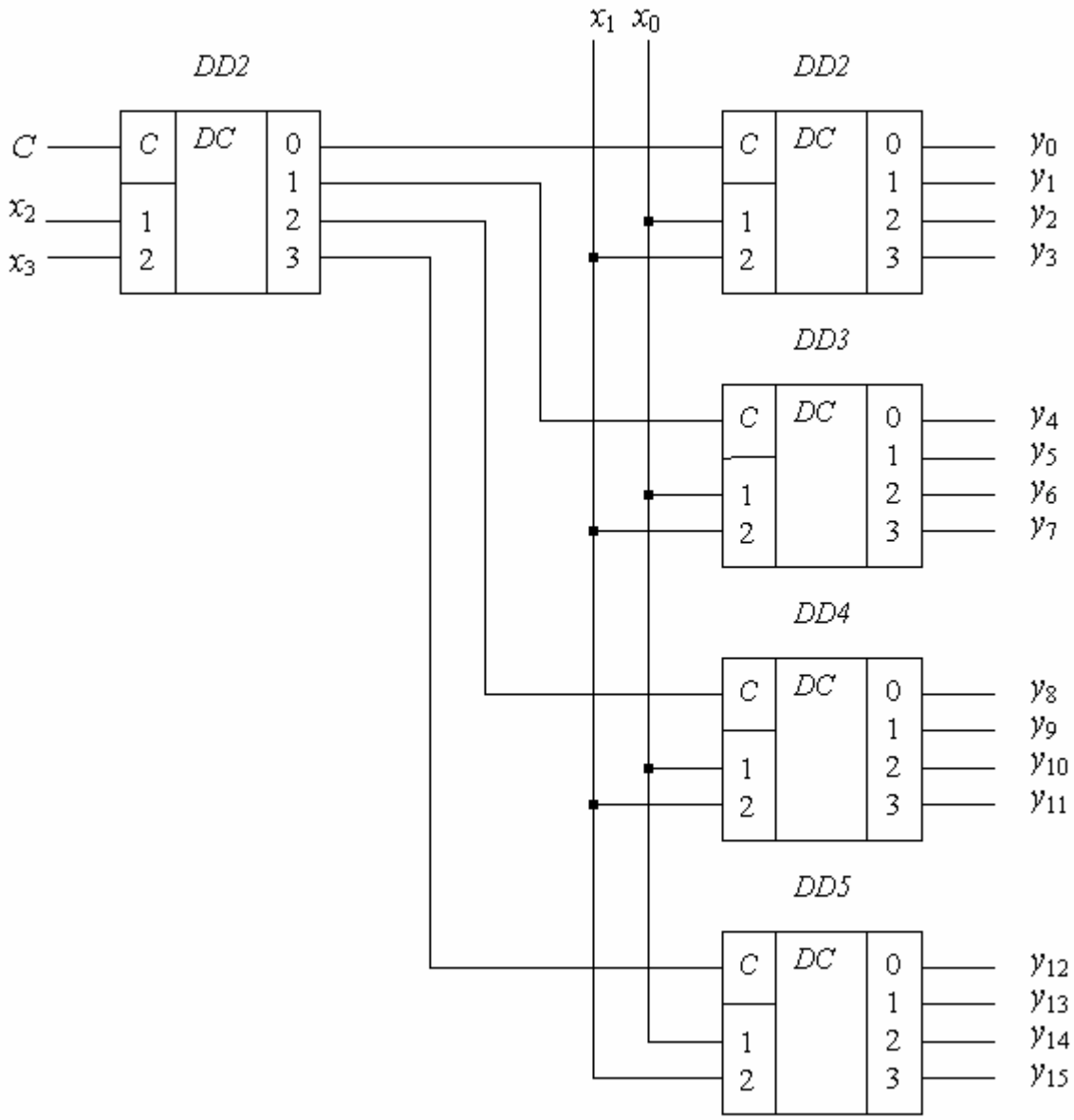


Рис. 3.6. Каскадное включение дешифраторов

Младшие разряды дешифрируемого числа x_0 и x_2 подаются на входы дешифраторов *DD*2...*DD*5.

Однако выполнять дешифрирование этих разрядов будет только тот дешифратор, который включен сигналом, поданным на вход *C* от дешифратора старших разрядов.

Так, например, при дешифрировании числа 1001 на вход поступает код 10, которым возбуждается выход 2. В этом случае включается дешифратор *DD*4, на вход которого подан код 01 младших разрядов дешифрируемого числа. В результате будет возбужден выход 1 дешифратора *DD*4, при этом на выходе появится сигнал y_9, что соответствует выбранному входному коду.

Для расширения числа входов и выходов дешифраторов можно воспользоваться параллельным или прямоугольным дешифратором, схема которого приведена на рис. 3.7.

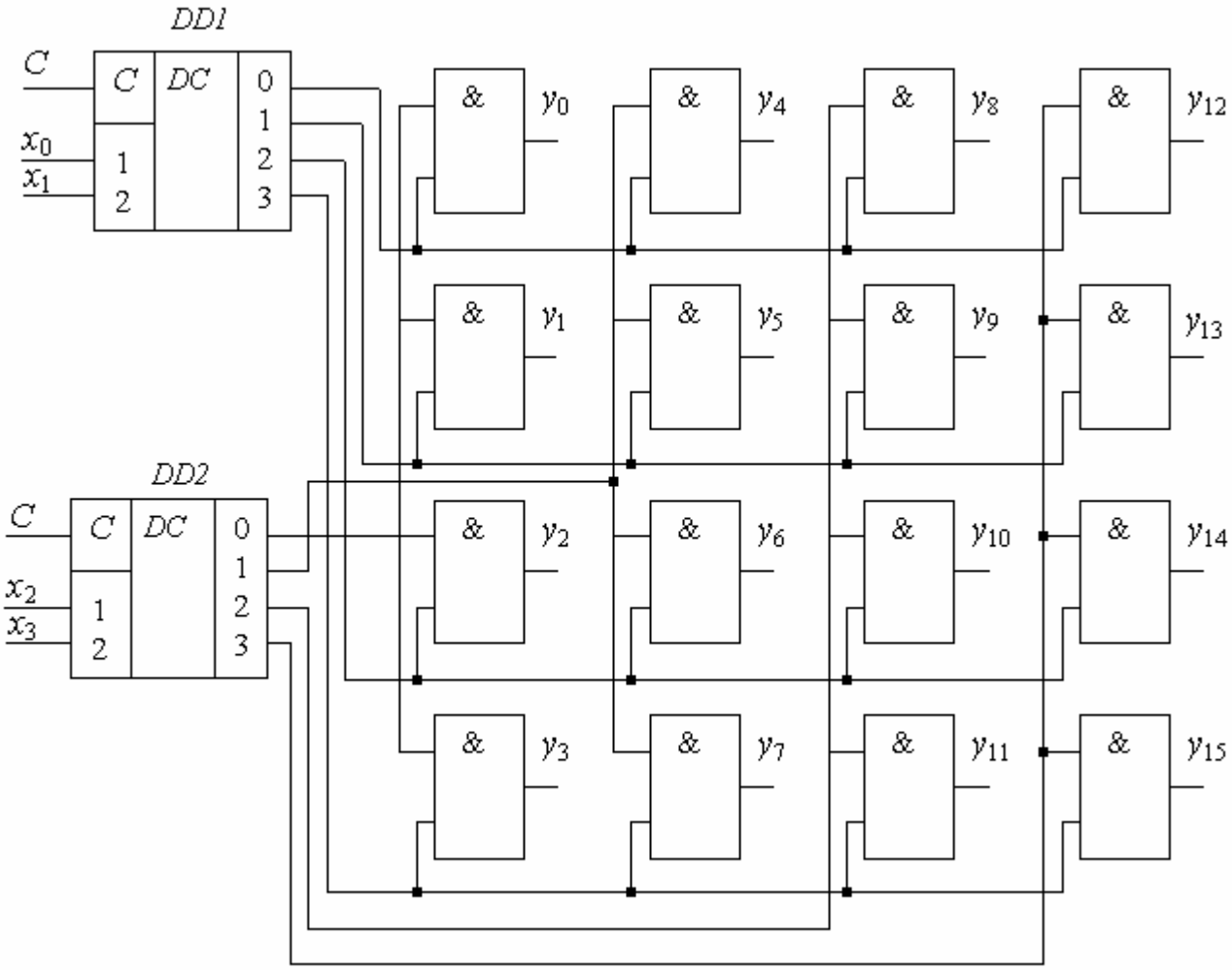


Рис. 3.7. Схема прямоугольного дешифратора

Схема прямоугольного дешифратора состоит из двух ступеней. Первая ступень состоит из двух дешифраторов *DD*1 и *DD*2, первый из которых дешифрует младшие разряды x_0 и x_1 входного числа, а второй – старшие разряды x_2 и x_3. Вторая ступень состоит из *N* элементов 2И-НЕ. Все элементы 2И-НЕ разделены на строки и столбцы: строками управляет дешифратор первой ступени на *DD*1, а столбцами управляет дешифратор *DD*2. Схема, приведенная на рис. 3.7, соответствует полному дешифратору. Если исключить некоторые из элементов 2И-НЕ, то получим неполный дешифратор с уменьшенным числом выходов.

В стандартные серии ИМС входят дешифраторы 2x4, 3x8, 4x16. Различаются микросхемы – дешифраторы входами управления (разрешения/ запрета

входных сигналов), а также типом выхода (с двумя активными состояниями или открытым коллектором). В отечественных сериях микросхемы дешифраторов обозначаются буквами ИД. Микросхемы дешифраторов приведены в табл. 3.4.

Наиболее типичное применение дешифраторов состоит в дешифрировании входных кодов. Еще одним из распространенных применений дешифраторов является селекция (выбор) заданных входных кодов. Рассмотрим пример селектирования 8-разрядного двоичного кода 00101011 на микросхемах К555ИД3 4х16 (рис. 3.8).

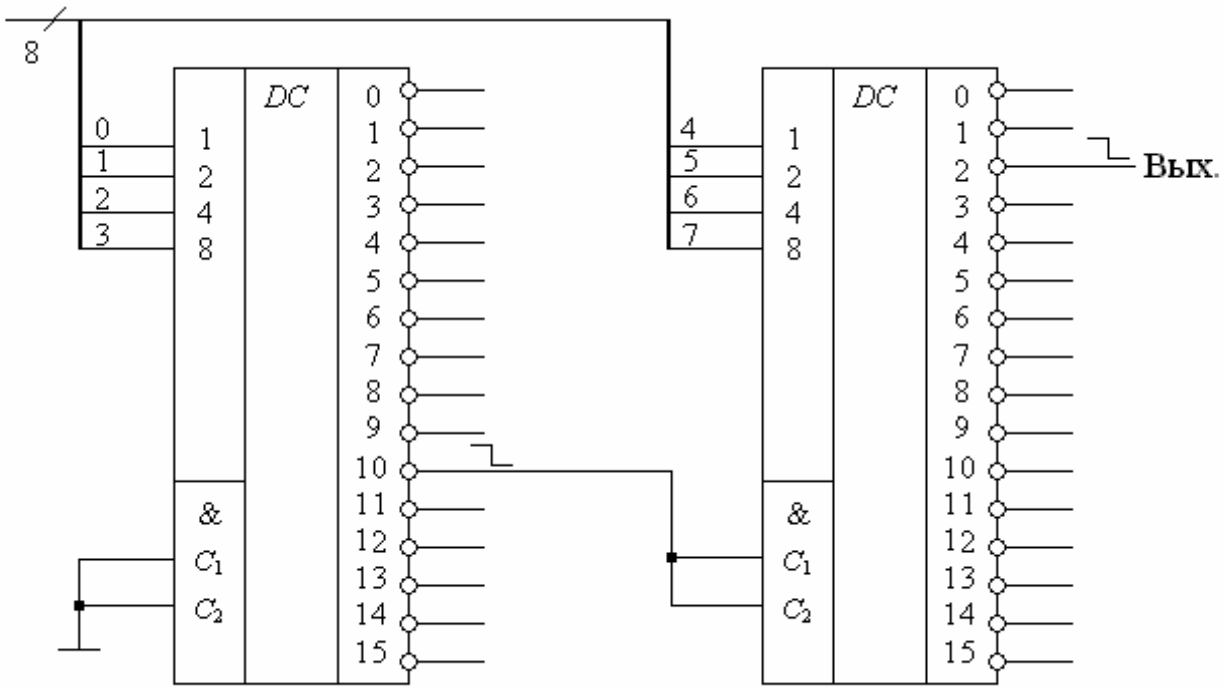


Рис. 3.8. Селектирование кода на дешифраторах

Таблица 3.4. Интегральные микросхемы преобразователей кодов, шифраторов и дешифраторов

Наименование микросхемы	Функциональное назначение	Количество входов	Количество выходов
К155ИД1	Высоковольтный дешифратор для управления газоразрядными индикаторами	4	10
К555ИД3	Полный дешифратор 4х16 со стробированием	4	16

К555ИД4	Сдвоенный дешифратор 2х4 со стробированием	2	8
К555ИД5	Сдвоенный дешифратор 2х4 с открытым коллекторным выходом	2	8
К555ИД6	Дешифратор 4х10	4	10
К155ИД8	Преобразователь кода для управления светодиодной матрицей 7х5	4	18
К555ИД10	Дешифратор 4х10 с открытым коллекторным выходом	4	10
К155ИД11	Преобразователь кода для управления шкальным индикатором с заполнением	3	8
К155ИД12	Преобразователь кода для управления шкальным индикатором с одной точкой	3	8
К155ИД13	Преобразователь кода для управления шкальным индикатором с двумя точками	3	8
1531ИД14	Сдвоенный дешифратор 2х4 со стробированием	2х2	4х4
К155ИД15	Преобразователь кода для управления шкальным индикатором	4	5
К555ИВ1	Приоритетный шифратор 8х3	8	3
К555ИВ2	Приоритетный шифратор 8х3 с тремя состояниями на выходе	8	3
К555ИВ3	Приоритетный шифратор 10х4	10	4
К155ПР6	Преобразователь двоично-десятичного кода в двоичный	6	8
К155ПР7	Преобразователь двоичного кода в двоично-десятичный	6	6

Появление отрицательного сигнала на выбранном выходе дешифратора будет означать поступление на вход заданного кода. Объединяются дешифраторы так, что один из них разрешает работу другого по входам C_1 и C_2.

Ещё одно важное применение дешифраторов состоит в перекоммутации одного входного сигнала на несколько выходов. Дешифратор в данном случае выступает в качестве демультиплексора входных сигналов, который позволяет разделить входные сигналы, приходящие в разные моменты времени, на одну входную линию (мультиплексированные сигналы). При этом входы 1,2,4,8 дешифратора используются в качестве управляющих, адресных, определяющих, на какой выход переслать пришедший в данный момент входной сигнал (рис.

3.9). А сам сигнал подается на один из входов *C* и пересылается на заданный выход. Если у микросхемы имеется несколько стробирующих входов *C*, то оставшиеся входы *C* можно использовать в качестве разрешающих работу дешифратора.

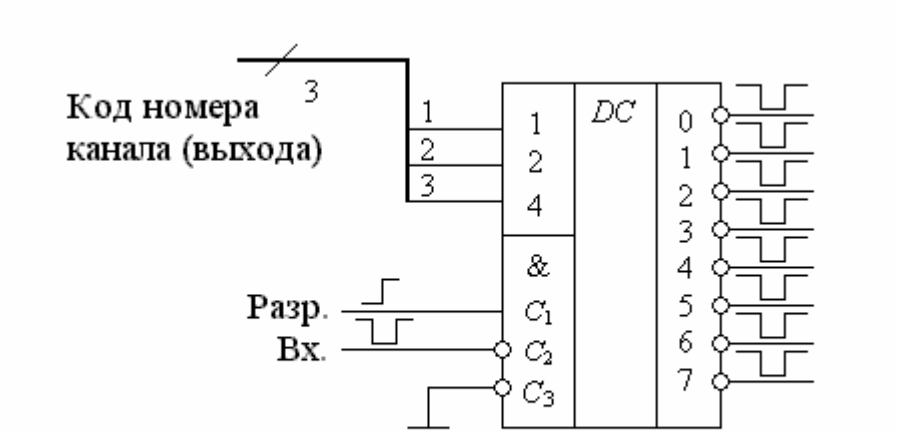


Рис. 3.9. Включение дешифратора как демультиплексора

3.3. Мультиплексоры и демультиплексоры

Мультиплексор. Мультиплексором называют цифровое устройство, которое обеспечивает поочередную передачу цифровой информации, поступающей по нескольким входным линиям связи, на одну выходную линию. Выбор входной линии, информация с которой поступает на выход, осуществляется при помощи сигналов, поступающих на адресные (управляющие) входы. В условных графических обозначениях мультиплексор называется *MUX* или *MS* (от слова *multiplexer*).

Обобщенная схема мультиплексора приведена на рис. 3.10.

Рис. 3.10. Обобщенная схема мультиплексора

Мультиплексор *MUX* (*Multiplexer*) в общем случае можно представить в виде коммутатора, управляемого входной логической схемой. Входные логические сигналы x_0, $x_1,\ldots,x_{n-1}$ поступают на входы коммутатора и через коммутатор передаются на выход y. Управление коммутатором осуществляется входной логической схемой. На вход логической схемы подаются адресные сигналы a_0, $a_1,\ldots, a_{m-1}$. Мультиплексоры могут иметь дополнительный управляющий вход E (*Enable*), который может выполнять стробирование (разрешение) выхода y. Кроме этого некоторые мультиплексоры могут иметь выход с тремя состояниями: два состояния 0 и 1 и третье состояние – пассивное, отключенное. Перевод мультиплексора в третье состояние производится сигналом *OE* (*Output Enable*).

Как правило, мультиплексоры передают информационные сигналы x_i, $i=(0, n-1)$ только в одном направлении – со входа на выход. Однако имеются мультиплексоры, которые могут передавать информационные сигналы в обоих направлениях. Такие мультиплексоры называются двунаправленными.

Двунаправленные мультиплексоры способны передавать не только цифровые, но и аналоговые сигналы. В литературе такие мультиплексоры часто называют селекторами-мультиплексорами.

Мультиплексоры со стробирующим входом E выполняют функции передачи сигналов $x_i \to y$ только при наличии активного сигнала строба E. Выходы мультиплексоров бывают прямыми и инверсными. Мультиплексоры, имеющие три состояния выхода, можно каскадировать, то есть объединять выходы с выходами других мультиплексоров.

Для обозначения коммутационных возможностей мультиплексора пользуются условной записью ($n \to 1$), где n – число входов. Так, например, мультиплексор с функцией $(1 \to 1)$ является одиночным ключом, а мультиплексор $(4 \to 1)$ имеет четыре входа и один выход.

В зависимости от соотношения числа информационных входов n и числа адресных входов m мультиплексоры делятся на полные и неполные. Для полного мультиплексора выполняется условие $n = 2^m$. Если это условие не выполняется, т. е. $n < 2^m$, то мультиплексор неполный. Наибольшее распространение получили мультиплексоры $(2 \to 1)$ с $n = 2$ и $m = 1$, $(4 \to 1)$ с $n = 4$ и $m = 2$, $(8 \to 1)$ с $n = 8$ и $m = 3$ и $(16 \to 1)$ с $n = 16$ и $m = 4$. Для неполных мультиплексоров число входных линий может быть любым, не больше 2^m.

В качестве примера рассмотрим функционирование мультиплексора $(4 \to 1)$, таблица истинности которого приведено в табл. 3.5.

Таблица 3.5. Таблица истинности мультиплексора (4 - 1)

a_1	a_0	y
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

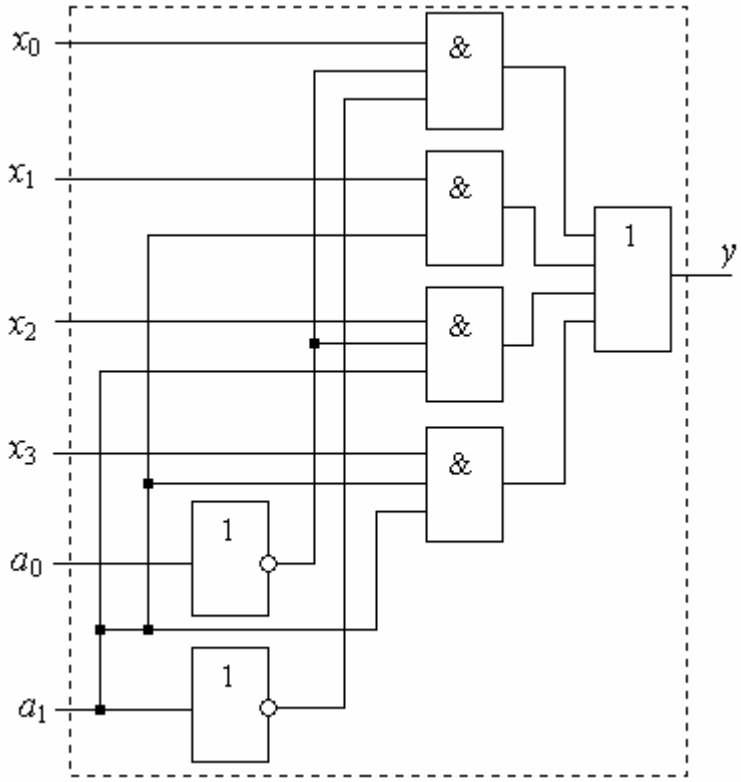


Рис. 3.11. Схема мультиплексора (4→1)

Таблица 3.6. Интегральные микросхемы мультиплексора

Наименование микросхемы	Функциональное назначение	Количество входов	Количество выходов
К155КП1	Стробируемый мультиплексор с инверсным выходом	16	1
К555КП2	Сдвоенный мультиплексор со стробированием	4	2

К155КП7	Стробируемый мультиплексор с прямым и инверсным выходами	8	1
К155КП5	Мультиплексор с прямым выходом	8	1
К555КП11	Четыре стробируемых мультиплексора с тремя состояниями выхода	2	4
К555КП12	Два мультиплексора с тремя состояниями выхода	4	2
К555КП13	Стробируемый мультиплексор с памятью на *D*-тригерах	4	1
КР531КП15	Мультиплексор с тремя состояниями, прямым и инверсным выходами	8	1
К555КП16	Мультиплексор со стробированием	2	4
К555КП17	Два мультиплексора с тремя состояниями, прямым и инверсным выходами	4	2
К531КП18	Четыре мультиплексора со стробированием и инверсными выходами	2	4
К561КП3	Двунаправленный мультиплексор со стробированием	8	1
К561КП1	Два двунаправленных мультиплексора со стробированием	4	2

Логическая функция, выполняемая мультиплексором имеет вид:

$$y = x_0 \cdot \overline{a}_1 \cdot \overline{a}_0 \vee x_1 \cdot \overline{a}_1 \cdot a_0 \vee x_2 \cdot a_1 \cdot \overline{a}_0 \vee x_3 \cdot a_1 \cdot a_0 . .$$

Для реализации полученной выходной функции четырехвходового мультиплексора потребуются четыре трехвходовых логических элемента И и один четырехвходовый элемент ИЛИ. Схема такого мультиплексора приведена на рис. 3.11. Для получения прямых и инверсных адресных сигналов используются два дополнительных инвертора.

В общем виде выходная функция мультиплексора может быть представлена как:

$$y = x_0 \cdot \overline{a}_{n-1} \cdot \ldots \cdot \overline{a}_1 \cdot \overline{a}_0 \vee x_1 \cdot \overline{a}_n \cdot \ldots \cdot \overline{a}_1 \cdot \overline{a}_0 \vee \ldots \vee x_{n-1} \cdot a_n \cdot \ldots \cdot a_1 \cdot a_0 . .$$

В отечественных сериях мультиплексоры имеют код типа микросхемы КП. Выпускается большое количество различных микросхем мультиплексоров, некоторые из которых перечислены в табл. 3.6.

Интегральные микросхемы мультиплексоров можно разделить на группы по следующим признакам: по числу входов: 2-, 4-, 8- и 16-входовые; по числу мультиплексоров в одном корпусе (числу разрядов); по наличию стробирующего входа *E;* по наличию выхода с тремя состояниями (наличию входа *OE*); по способности передавать сигналы в двух направлениях.

Применение мультиплексоров с тремя состояниями выходов позволяет легко увеличить число коммутируемых каналов.

На рис. 3.12 показана схема мультиплексора $(16 \rightarrow 1)$, выполненная на мультиплексорах $(8 \rightarrow 1)$ и инверторе на входе разрешения *OE*. Выходы *y* мультиплексоров *DD*1 и *DD*2 соединены вместе для организации функции «монтажное ИЛИ». При значении адресного сигнала $a_3 = 0$ включается микросхема *DD*1, а при значении $a_3 = 1$– микросхема *DD*2. При включении микросхемы *DD*1 на общий выход поступает один из информационных сигналов x_i, i=0,…,7, подключенных ко входам *DD*1. При включении микросхемы *DD*2 на общий выход поступают сигналы x_i, i=8,…,15. В качестве элементов *DD*1 и *DD*2 в этой схеме можно использовать микросхемы КР531КП15 (или более медленные К555КП15).

Другой способ каскадирования микросхем мультиплексоров основан на использовании схемы, приведенной на рис. 3.13. Рассмотрим пример построения мультиплексора $(64 \rightarrow 1)$, используя мультиплексоры $(8 \rightarrow 1)$. Можно взять восемь мультиплексоров $(8 \rightarrow 1)$, не имеющих третьего состояния выхода (например, К555КП7), на их адресные входы a_2, a_1, a_0, подать одни и те же адресные сигналы, производящие выбор одного из восьми каналов в каждом мультиплексоре, тогда общее число входов будет равно 64.

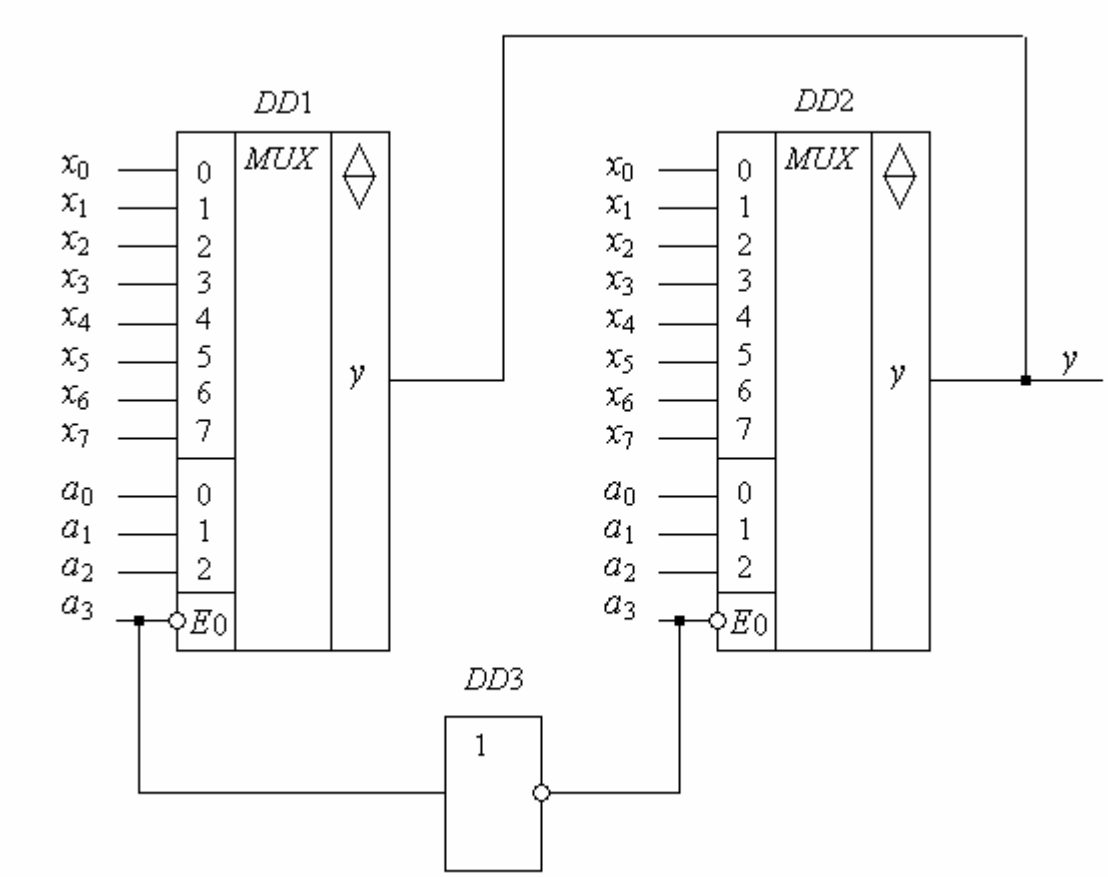


Рис. 3.12. Каскадирование мультиплексоров с тремя состояниями выхода

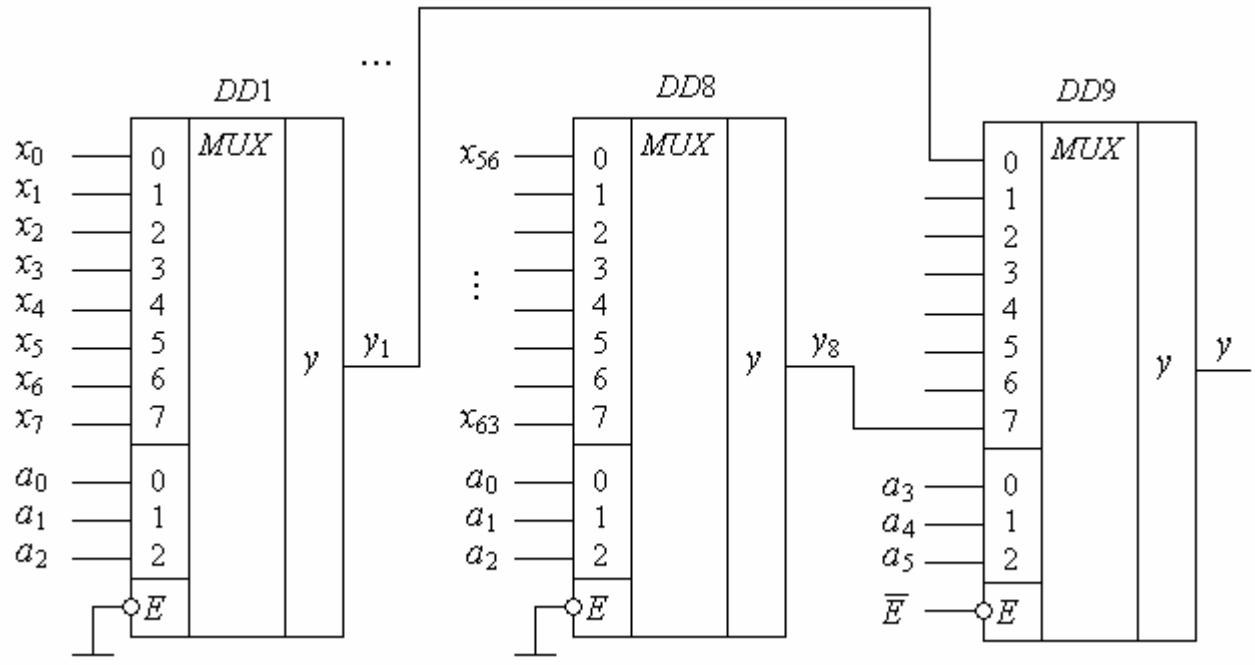


Рис. 3.13. Схема мультиплексора (64→1) на мультиплексорах (8→1)

Последний мультиплексор *DD*9 управляется адресными сигналами a_5, a_4, a_3 и определяет, какой из восьми мультиплексоров *DD*1...*DD*8 будет подключен к выходу *y*. Вход стробирования *E* можно использовать только у последнего мультиплексора *DD*9. Схема стробируемого мультиплексора с форматом $(64 \to 1)$ показана на рис. 3.13.

Мультиплексор помимо основного назначения коммутации входных сигналов может быть использован как универсальный логический модуль, т.е. как устройство, на основе, которого можно реализовать произвольную логическую функцию.

Реализация логических функций на мультиплексорах. Мультиплексор, помимо своей основной функции - передачи сигнала с того или иного входа на выход, может быть использован как универсальный логический модуль (УЛМ), то есть как устройство, на основе которого можно реализовать произвольную логическую функцию. Рассмотрим реализацию цифрового устройства на основе восьмивходового мультиплексора, воспроизводящего, например, логическую функцию, равную единице, если равны единице два и только два аргумента, и равную нулю на всех остальных комбинациях значений аргументов (рис. 3.14).

Входами устройства будут управляющие входы мультиплексора $x_2=a_2$, $x_1=a_1$, $x_0=a_0$, а на информационные входы необходимо подать сигналы, соответствующие 0 или 1 в зависимости от реализуемой функции (рис. 3.14).

а)

x_2 (a_2)	x_1 (a_1)	x_0 (a_0)	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

б)

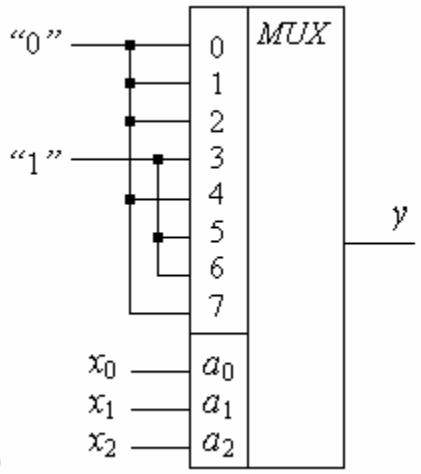


Рис. 3.14. Таблица истинности цифрового устройства (а) и его реализация на мультиплексоре $(8 \to 1)$ (б)

Увеличить число входов можно, подавая на информационные входы мультиплексора , не постоянные сигналы из множества {0, 1}, а сигналы из множества $\{0, 1, x_i, \overline{x}_i\}$, где x_i – один из аргументов воспроизводимой функции.

Один из вариантов синтеза четырехвходового цифрового устройства на основе мультиплексора $(8 \to 1)$ представлен на рис. 3.15.

На основе заданной таблицы истинности цифрового устройства (рис. 3.15а) заполняется таблица программирования (рис. 3.15б). В данной реализации три первых входа x_0, x_1, x_2 цифрового устройства подключаются к управляющим входам мультиплексора, а x_3 подается в прямом или инверсном виде на соответствующие информационные входы. По таблице программирования (рис. 3.15б) строится схема цифрового устройства (рис. 3.15г). Необходимо отметить, что возможен другой вариант его построения, например, когда на информационные входы подаются сигналы x_0, x_1, x_3, а оставшийся вход x_2 подается на соответствующие информационные входы мультиплексора (рис. 3.15в, д).

x_3	x_2	x_1	x_0	y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

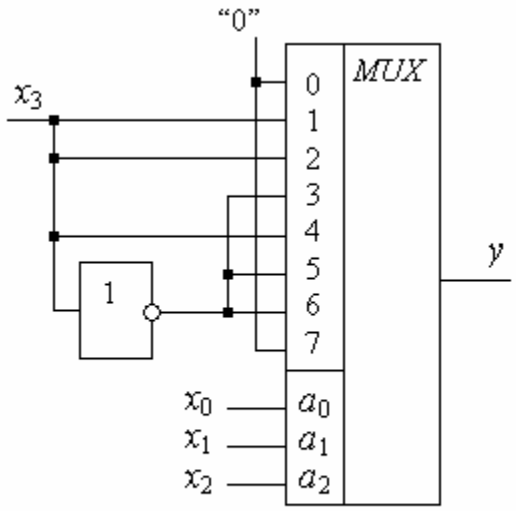
а)

x_2 (a_2)	x_1 (a_1)	x_0 (a_0)	y
0	0	0	0
0	0	1	x_3
0	1	0	x_3
0	1	1	$\overline{x}_3$
1	0	0	x_3
1	0	1	$\overline{x}_3$
1	1	0	$\overline{x}_3$
1	1	1	0

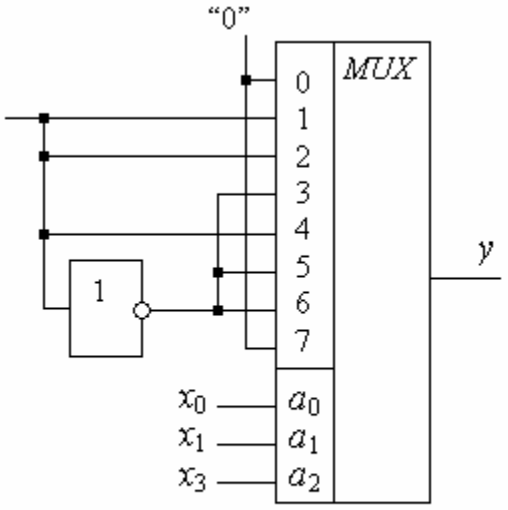
б)

x_3 (a_2)	x_1 (a_1)	x_0 (a_0)	y
0	0	0	0
0	0	1	x_2
0	1	0	x_2
0	1	1	$\overline{x}_2$
1	0	0	x_2
1	0	1	$\overline{x}_2$
1	1	0	$\overline{x}_2$
1	1	1	0

в)



г)



д)

Рис. 3.15. Реализация четырехвходового цифрового устройства на мультиплексоре $(8 \to 1)$. Таблица истинности (а), таблицы программирования (б, в) и схемы (г, д)

Выбор того или иного варианта определяется видом реализуемой логической функции. Общих методов выбора наилучшего варианта пока не имеется. Для реализации КЦА с большим числом аргументов требуются мультиплексоры с большим числом входов, которые могут быть построены по приведенным выше алгоритмам. Так, на основе мультиплексоров с числом входов 64 рассмотренным способом можно построить КЦА, реализующие произвольную логическую функцию семи аргументов.

Временная задержка в данных устройствах может быть определена суммой задержек конкретных ИМС, входящих в последовательную цепь распространения сигналов.

Демультиплексор. Демультиплексором называют цифровое устройство, которое обеспечивает передачу цифровой информации, поступающей по одной линии, на несколько выходных линий. Выбор выходной линии осуществляется при помощи сигналов, поступающих на адресные входы. Таким образом, демультиплексор выполняет преобразование, обратное мультиплексору. В условных графических обозначениях демультиплексор называется *DMX* или *DMS*.

Обобщенная схема демультиплексора, приведенная на рис. 3.16, сходна со схемой мультиплексора. Входной сигнал x поступает на вход коммутатора и через него передается на выходы $y_0,\ldots, y_{n-1}$. Адресные сигналы $a_0,\ldots, a_m$ имеют то же назначение, что и у мультиплексора. Сигнал стробирования E разрешает передачу входного сигнала через коммутатор.

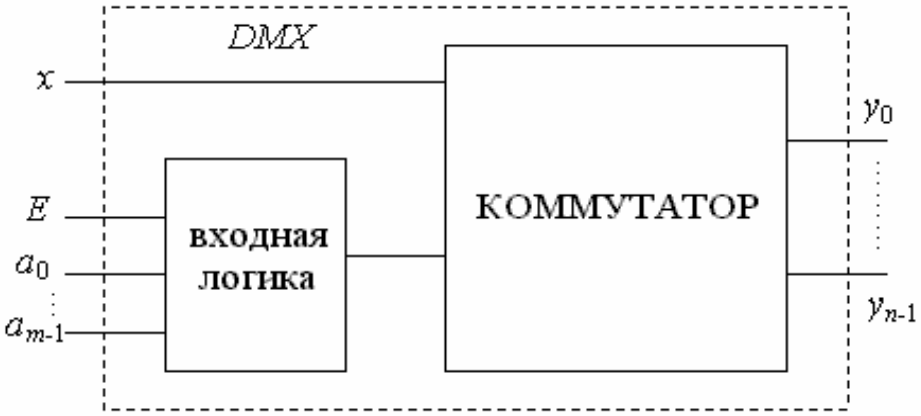


Рис. 3.16. Обобщенная схема демультиплексора

Для обозначения коммутационных возможностей демультиплексоров можно пользоваться записью, аналогичной условной записи мультиплексоров ($1\to n$), где n – число выходов демультиплексора. Так, например, демультиплексор ($1\to 2$) имеет два выхода, а демультиплексор ($1\to 4$) – четыре выхода. Условное графическое обозначение демультиплексора ($1\to 4$) приведено на рис. 3.17а.

Демультиплексоры, как и мультиплексоры, могут быть полными и неполными. Деление мультиплексоров на эти две категории производится так же, как и у мультиплексоров, с той лишь разницей, что под *n* понимается число выходов, а не входов, как в мультиплексоре.

В качестве примера рассмотрим функционирование демультиплексора (1→4), таблица истинности которого приведена в табл. 3.7. Используя данные этой таблицы, получим выходные логические функции:

$$y_0 = x \cdot (\overline{a_0} \cdot \overline{a_1}) = \overline{\overline{x} + a_0 + a_1}\,; \quad y_1 = x \cdot (a_0 \cdot \overline{a_1}) = \overline{\overline{x} + \overline{a_0} + a_1}\,;$$

$$y_2 = x \cdot (\overline{a_0} \cdot a_1) = \overline{\overline{x} + a_0 + \overline{a_1}}\,; \quad y_3 = x \cdot (a_0 \cdot a_1) = \overline{\overline{x} + \overline{a_0} + \overline{a_1}}\,.$$

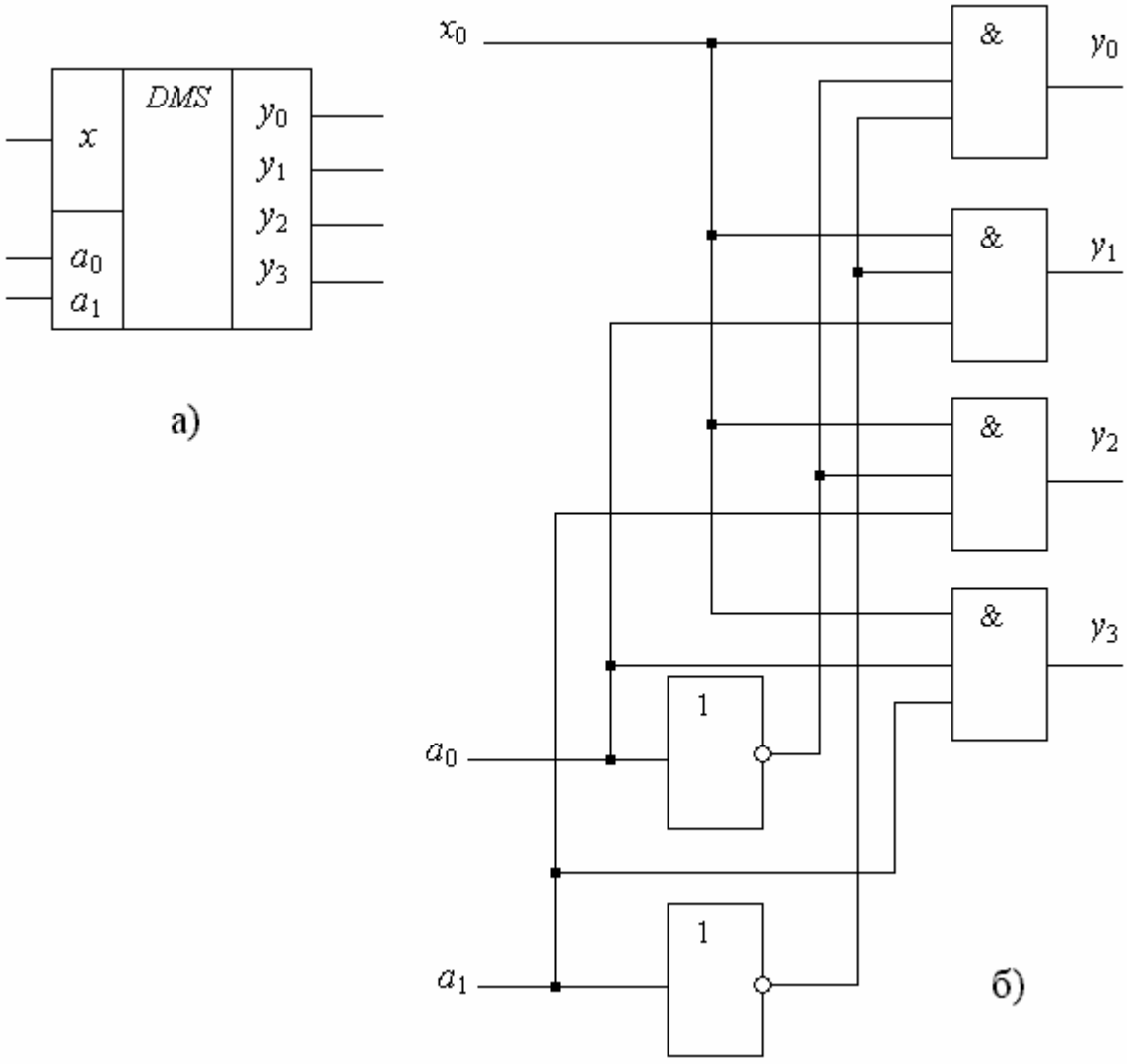


Рис. 3.17. Условное графическое обозначение (а) и схема демультиплексора (1→4) на логических элементах И (б)

Таблица 3.7. Состояния демультиплексора (1→ 4)

a_1	a_0	y_0	y_1	y_2	y_3
0	0	x	0	0	0
0	1	0	x	0	0
1	0	0	0	x	0
1	1	0	0	0	x

Схемы демультиплексора на логических элементах И и ИЛИ-НЕ, реализующие полученные функции, приведены на рис. 3.17б и рис. 3.18. Инверторы в этих схемах обеспечивают формирование необходимых сигналов управления. В каждой схеме И два входа задействованы для адресных сигналов a_0 и a_1, а на третий вход подается входной сигнал x.

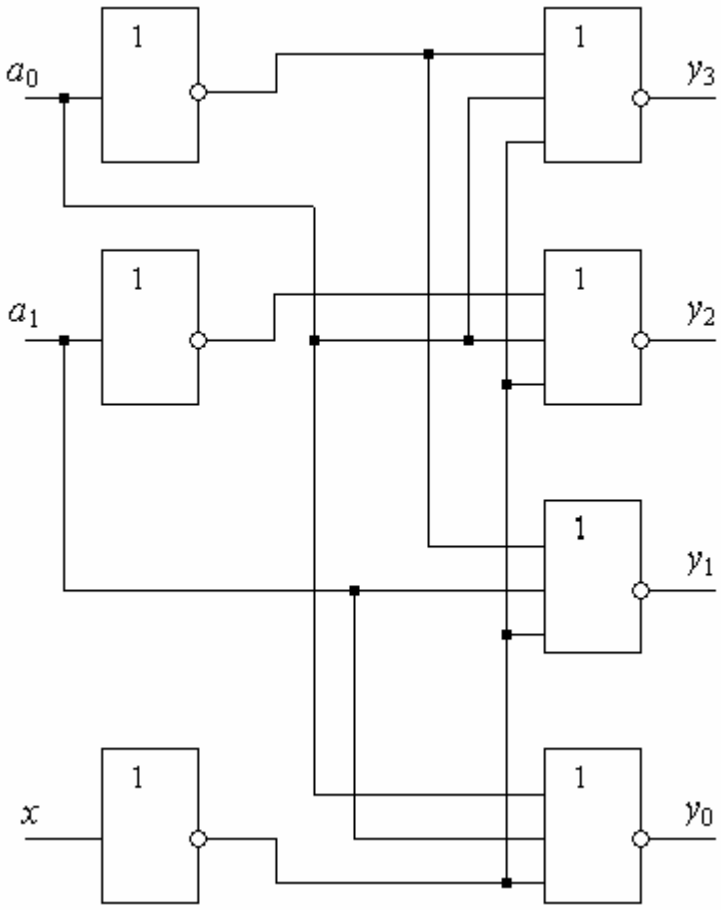


Рис. 3.18. Схема демультиплексора (1→4) на логических элементах ИЛИ-НЕ

Интегральные микросхемы демультиплексоров, так же, как и схемы мультиплексоров, можно разделить на группы по следующим признакам: по

числу выходов; по числу демультиплексоров в одном корпусе; по наличию стробирующего сигнала *E*; по способности передавать сигналы в двух направлениях.

Поскольку выходные функции демультиплексоров сходны с выходными функциями дешифраторов, у них одинаковое условное обозначение ИД. Такие микросхемы часто называют дешифраторами - демультиплексорами. Поэтому в качестве демультиплексора можно применить дешифратор, в котором информационный входной сигнал подается на разрешающий (стробирующий) вход. Так, например, дешифратор К155ИД3 можно использовать в качестве демультиплексора с форматом $(1 \to 16)$. При этом входы разрешения дешифрации используются в качестве основного входа демультиплексора *x*, а адресные входы и выходы используются по прямому назначению. В табл. 3.8 перечислены некоторые схемы демультиплексоров и дешифраторов, которые можно использовать качестве демультиплексоров.

Таблица 3.8. Интегральные схемы демультиплексора

Наименование микросхемы	Функциональное назначение	Количество входов	Количество выходов
К155ИД3	Дешифратор-мультиплексор со стробированием	16	1
К155ИД4	Два дешифратор-мультиплексора со стробированием	4	2
К531ИД7	Скоростной дешифратор-мультиплексор со стробированием	8	1
К531ИД14	Скоростной дешифратор-мультиплексор	4	2

Мультиплексоры-демультиплексоры. Среди схем коммутации можно особо выделить схемы, которые способны пропускать сигналы в обоих направлениях. К таким элементам относится коммутационные микросхемы, выполненные по технологии КМОП. Коммутаторы КМОП способны пропускать как аналоговые, так и цифровые сигналы, в них можно менять местами вход и выход. К ним относятся элементы отечественных серий микросхем: К176, К561, К564, КР1561, 1564, 590 и 591.

Для обозначения коммутационных возможностей мультиплексоров-демультиплексоров можно пользоваться записью $(1 \leftrightarrow n)$, в котором двунаправленная стрелка указывает на двунаправленную передачу сигналов. В табл. 3.9 приведены сведения о некоторых ИМС мультиплексоров-демультиплексоров.

Таблица 3.9. Интегральные схемы мультиплексоров-демультиплексоров

Наименование микросхемы	Функциональное назначение	Количество входов	Количество выходов
564КП1	Двухразрядный мультиплексор-демультиплексор	4	2
564КП2	Мультиплексор-демультиплексор с тремя состояниями выхода.	8	1
590КН1	Мультиплексор-демультиплексор	8	1

3.4. Компараторы

Цифровым компаратором называют цифровое устройство, выполняющее операцию определения отношения между двумя числами (устройство сравнения чисел). Основными отношениями можно считать "равно" и "больше". Другие отношения могут быть определены через основные. Так, например признак неравенства чисел можно получить как инверсию признака равенства $y_{A\neq B}= \overline{y}_{A=B}$, отношение «меньше» - путем перемены местами аргументов в функции $y_{A>B}$ ($y_{A<B} = y_{B>A}$). Для остальных можно записать: $y_{B\geq A}= \overline{y}_{B>A}$; $y_{B\leq A}= \overline{y}_{B<A}$. Эти отношения используются как логические условия в устройствах контроля и диагностики цифровых систем, в микропроцессорных системах.

Устройства сравнения на равенство строятся на основе поразрядных операций над одноименными разрядами обоих чисел. Признак равенства чисел имеет единичное значение ($y_{A=B} = 1$), если во всех одноименных разрядах чисел $A=\sum_{i=0}^{m-1} 2^i \cdot a_i$; $B=\sum_{i=0}^{m-1} 2^i \cdot b_i$ содержатся либо единицы, либо нули. В этом случае логическая функция, реализуемая устройством сравнения на равенство, имеет вид:

$$y_{A=B} = \overline{a_{m-1} \oplus b_{m-1} \vee \ldots \vee a_1 \oplus b_1 \vee a_0 \oplus b_0}\,.$$

Полученную функцию устройства сравнения можно реализовать в заданном базисе логических элементов, используя конкретные микросхемы. Логическим элементом, устанавливающим равенство каждого разряда двух чисел, является сумматор по модулю два, для которого справедливо $y = \bar{a} \cdot b \vee a \cdot \bar{b}$, показывающее, что выходной сигнал *y* имеет низкий уровень при *a* = *b*. Затем определяется, имеют ли выходные сигналы всех сумматоров по модулю два низ-

кий уровень. Выходной сигнал, реализуется многовходовым логическим элементом И.

Обычно оказывается удобным иметь на выходе цифрового компаратора все три возможных признака сравнения: $y_{A<B}$, $y_{A=B}$, $y_{A>B}$. Таблица истинности такого компаратора для сравнения двухразрядных двоичных чисел (операндов) приведена в табл. 3.10.

Таблица 3.10. Таблица истинности компаратора

A		B		Входы		
a_1	a_0	b_1	b_0	$y_{A<B}$	$y_{A=B}$	$y_{A>B}$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

По заданной таблице истинности компаратор может быть синтезирован методом логического проектирования цифровых устройств. На схеме цифровые компараторы обозначаются двумя символами равенства: “= =”. Код типа микросхемы компаратора в отечественных сериях – СП. В ТТЛ серии входят микросхемы К555СП1, КР1533СП1, КР531СП1, представляющие собой четырехразрядные компараторы (рис. 3.19).

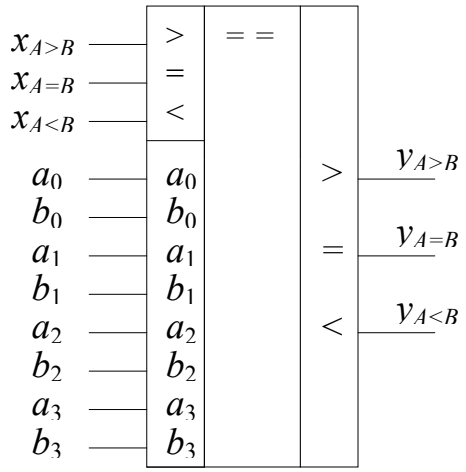


Рис. 3.19. Условное графическое обозначение цифрового компаратора с тремя выходами

Четыре пары входов принимают для анализа два четырехразрядных числа A и B. Три входа $x_{A>B}$, $x_{A=B}$, $x_{A<B}$ необходимы для наращивания разрядности компаратора. Компаратор имеет три выхода результатов сравнения $y_{A<B}$, $y_{A=B}$, $y_{A>B}$. Функционирование такого компаратора представлено таблицей истинности (табл. 3.11).

Таблица 3.11. Таблица истинности цифрового компаратора СП1

Входы сравниваемых чисел				Входы наращивания			Выходы сравнения		
$a_3,\ b_3$	$a_2,\ b_2$	$a_1,\ b_1$	$a_0,\ b_0$	$x_{A<B}$	$x_{A=B}$	$x_{A>B}$	$y_{A<B}$	$y_{A=B}$	$y_{A>B}$
$a_3>b_3$	*	*	*	*	*	*	0	0	1
$a_3<b_3$	*	*	*	*	*	*	1	0	0
$a_3=b_3$	$a_2>b_2$	*	*	*	*	*	0	0	1
$a_3=b_3$	$a_2<b_2$	*	*	*	*	*	1	0	0
$a_3=b_3$	$a_2=b_2$	$a_1>b_1$	*	*	*	*	0	0	1
$a_3=b_3$	$a_2=b_2$	$a_1<b_1$	*	*	*	*	1	0	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0>b_0$	*	*	*	0	0	1
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0<b_0$	*	*	*	1	0	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	0	0	0	1	0	1
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	0	0	1	0	0	1
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	1	0	0	1	0	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	1	0	1	0	0	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	*	1	*	0	1	0

* - неопределенное состояние (0 или 1)

Так, например, если старшие разряды находятся в отношении $a_3 > b_3$, то при любых сигналах на остальных входах на выходах $y_{A>B}$ будет присутствовать сигнал единица и нули на остальных выходах. Пять последних строк табл. 3.11 отображают режим наращивания разрядности, которое может быть последовательным и параллельным.

При последовательном наращивании выходы $y_{A<B}$, $y_{A=B}$, $y_{A>B}$ компаратора, сравнивающего младшие четыре разряда операндов, следует присоединить к одноименным входам последующего каскада (рис. 3.20а).

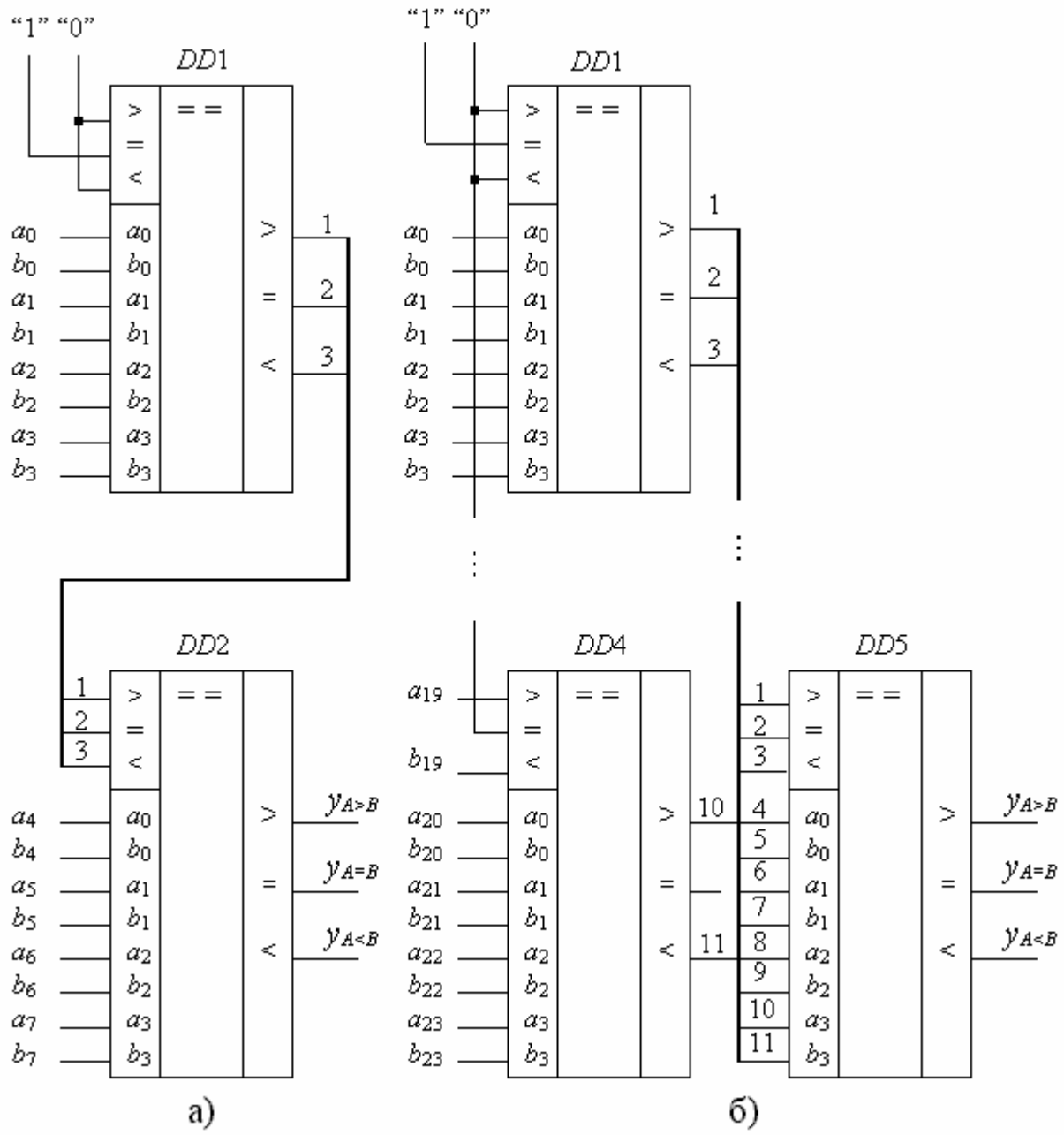


Рис. 3.20. Увеличение разрядности компараторов:
последовательное (а) и параллельное (б)

Для правильной работы многокаскадного компаратора на входы первой микросхемы $x_{A<B}$ и $x_{A>B}$ следует подать сигналы, соответствующие логическому нулю, а на вход $x_{A=B}$ – логической единице. Этим способом при двух компараторах СП1 можно сравнивать два восьмиразрядных числа. Аналогично можно строить последовательный компаратор для сравнения чисел большей разрядности. Однако каждый последовательный каскад добавит время задержки компаратора. Для уменьшения общей задержки компаратора целесообразно использовать параллельное включение компараторов (рис. 3.20б). Здесь компаратор младших разрядов (*DD*1) используется как четырехразрядный, четыре старших *DD*2-*DD*5 как пятиразрядные. При этом входы $x_{A>B}$ и $x_{A<B}$ служат пятой парой разрядных входов. На входы $x_{A=B}$ подано напряжение, соответствующее логическому нулю. Таким образом, компаратор (рис. 3.20б) может сравнивать двадцатичетырехразрядные операнды *A* и *B*. Его время задержки равно удвоенной задержке каждого компаратора.

В зарубежные стандартные серии входят также и 8-разрядные компараторы, сравнивающие два кода по величине (имеющие выходы =, >, <). Примером может служить микросхема *SN*74*AS*885.

Одно из основных применений цифровых компараторов состоит в селектировании входных кодов. В этом случае достаточно иметь информацию только о совпадении кодов на входах компаратора. Используется только выход равенства кодов *A=B*. Для подобных применений выпускаются специальные компараторы, определяющие только совпадение кодов (8-разрядные микросхемы *SN*74*ALS*520, *SN*74*ALS*521).

3.5. Арифметические устройства

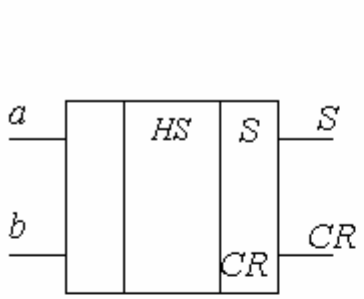
К арифметическим устройствам относятся сумматоры, арифметико-логические устройства (АЛУ) и перемножители многоразрядных двоичных чисел (операндов).

Сумматоры. Сумматор – цифровое устройство, обеспечивающее суммирование двух чисел. Сложение в двоичной системе счисления является самой важной арифметической операцией, так как оно лежит в основе других арифметических операций: вычитания, умножения, деления.

Основными устройствами, обеспечивающими элементарные операции, являются одноразрядные сумматоры. Все сумматоры подразделяются на две основные группы: комбинационные – не обладающие функцией хранения и накопительные – обладающие функцией хранения информации. Одноразрядные сумматоры составляют основу схем сумматоров как последовательных, так и параллельных операндов. Рассмотрим схемы одноразрядных сумматоров и

сумматоров многоразрядных чисел с последовательным и параллельным переносом. В связи с тем, что сумматоры широко используются в цифровой аппаратуре, они выпускаются также в виде интегральных схем.

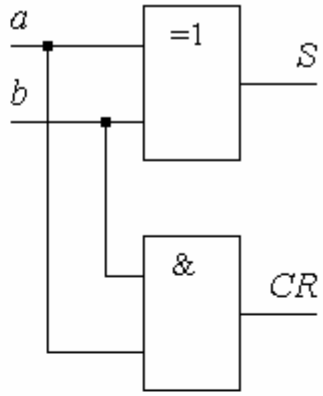
Полусумматор. Полусумматор реализует операцию сложения двух двоичных одноразрядных чисел и формирует на своих выходах сигналы суммы и переноса в старший разряд.



а)

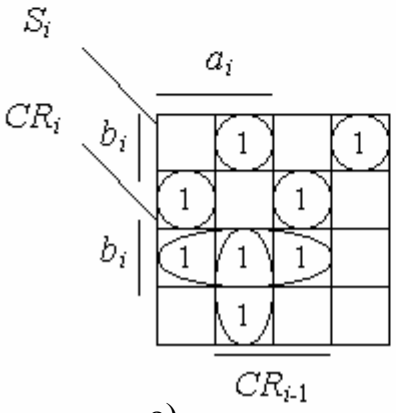
a	b	S	CR
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

б)



в)

Рис. 3.21. Полусумматор. Условное графическое обозначение (а), таблица истинности (б) и схема (в)



а)

a_i	b_i	CR_{i-1}	S_i	CR_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

б)

$$S_i = \overline{\overline{a_i \cdot b_i \cdot CR_{i-1}} \cdot \overline{a_i \cdot b_i} \cdot \overline{CR_{i-1}} \cdot \overline{a_i} \cdot b_i \cdot \overline{CR_{i-1}} \cdot \overline{a_i} \cdot \overline{b_i} \cdot CR_{i-1}}$$

$$CR_{i-1} = \overline{\overline{a_i \cdot b_i} \cdot \overline{a_i \cdot CR_{i-1}} \cdot \overline{b_i \cdot CR_{i-1}}}$$

в)

Рис. 3.22. Синтез одноразрядного полного сумматора. Таблица истинности (а), диаграммы Вейча (б), логические функции в структурной форме (в)

В условном обозначении (рис. 3.21а) полусумматор отображается буквами *HS* от слова *halfsummator*. Исходя из правил сложения двоичных чисел, можно заполнить таблицу истинности полусумматора (рис. 3.21б). Из таблицы видно, что полусумматор выполняет логические функции $S=a\oplus b;\ CR=a\cdot b$ и поэтому его схема может быть реализована в виде, изображенном на рис. 3.21в.

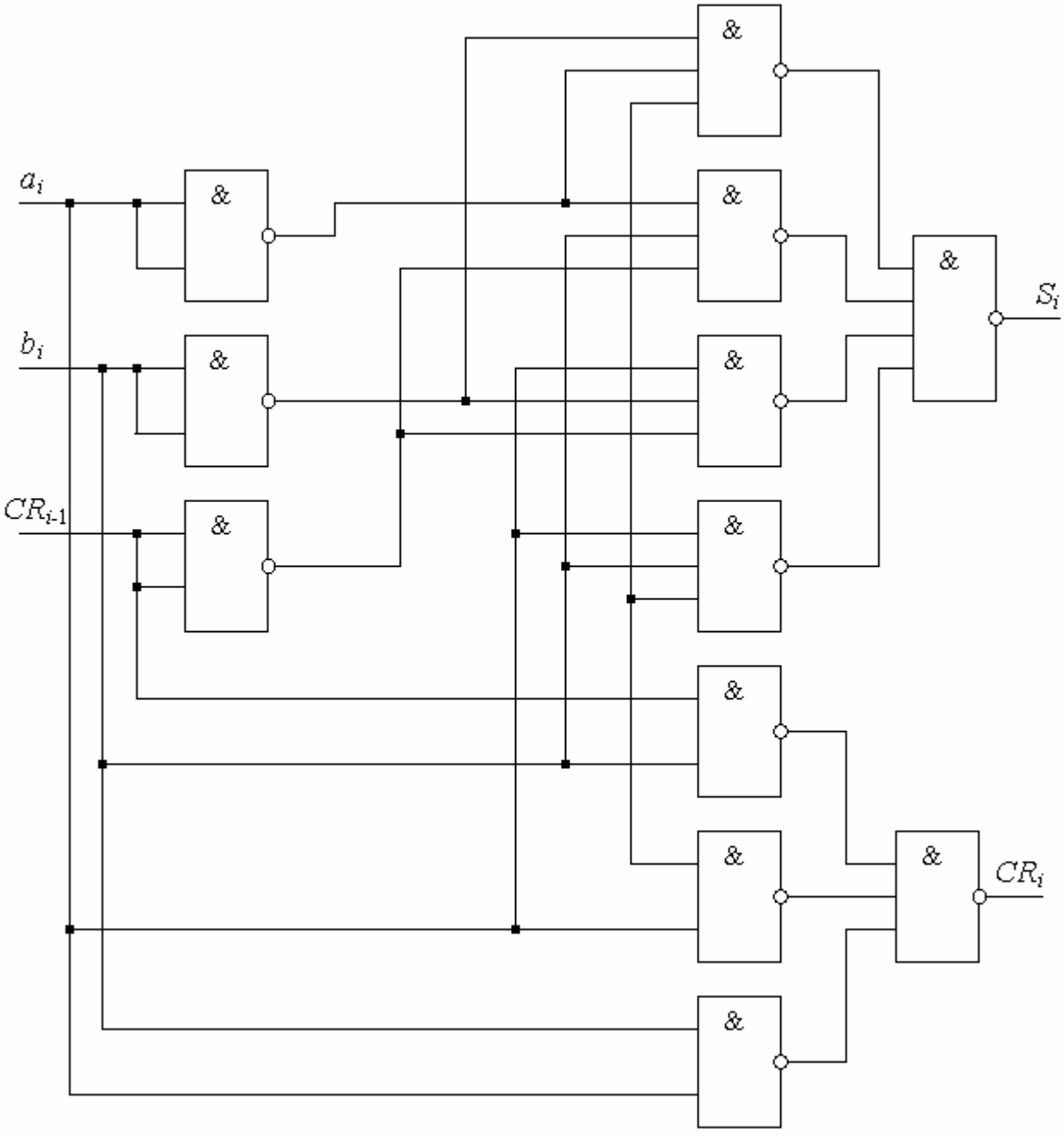


Рис. 3.23. Схема сумматора на элементах Шеффера

Одноразрядные полные сумматоры. Одноразрядный полный сумматор реализует сложение трех двоичных чисел - слагаемых a_i, b_i и сигнала переноса

$CR_{i\text{-}1}$ из предыдущего младшего разряда, а на выходе формирует сумму трех цифр S_i и сигнал переноса в следующий разряд CR_i. Данные синтеза сумматора в базисе элементов Шеффера приведены на рис. 3.22.

Считая задержки логических элементов сумматора в базисе элементов Шеффера одинаковыми и равными $t_{\text{з.ср.}}$ (рис. 3.23), можно определить задержки распространения сигналов от входов a_i, b_i, $CR_{i\text{-}1}$ до выходов S_i, CR_i.

Одноразрядный полный сумматор входит в серии интегральных схем. Например, в серии ТТЛ входят одноразрядные полные сумматоры К155ИМ1, К555ИМ5. Параметры быстродействия таких сумматоров приводятся в справочниках, и обычно они определяют задержки, возникающие при распространении сигнала с какого-либо входа на выход. В наихудшем случае они будут составлять величины:

$$t_{\text{з}.Si} = 3 \cdot t_{\text{з.ср}}; \qquad t_{\text{з}.CRi} = 2 \cdot t_{\text{з.ср}}.$$

Арифметические сумматоры многоразрядных двоичных чисел. Многоразрядные сумматоры должны обеспечивать сложение двоичных многоразрядных операндов A и B. Допустим, необходимо сложить два четырехразрядных операнда:

$$A = \sum_{i=0}^{3} 2^i \cdot a_i \text{ и } B = \sum_{i=0}^{3} 2^i \cdot b_i .$$

Исходя из правил сложения двоичных чисел, мы должны последовательно, начиная с младшего разряда, суммировать три цифры – цифры данного разряда слагаемых a_i, b_i, и сигнал переноса из предыдущего младшего разряда $CR_{i\text{-}1}$. Сумматоры для обработки последовательных и параллельных операндов имеют различные структуры.

Последовательный сумматор. Сумматор для последовательных операндов содержит всего один одноразрядный сумматор, обрабатывающий поочерёдно разряд за разрядом, начиная с младшего.

Сложив младшие разряды, одноразрядный сумматор вырабатывает сумму для младшего разряда результата и перенос. В следующем такте складывается вновь поступившие разряды слагаемых x_i и b_i с переносом из младшего и т. д. Сумматор последовательных операндов (рис. 3.24) также содержит сдвигающие регистры слагаемых и сумм и триггер хранения переноса. Регистры и триггер тактируются синхроимпульсами C.

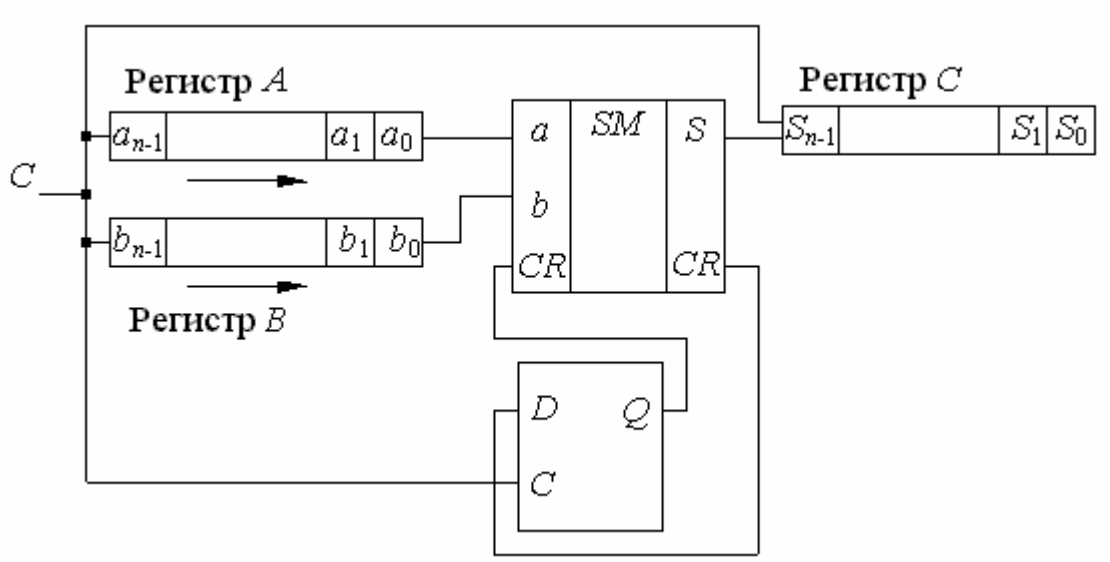


Рис. 3.24. Схема сумматора для последовательных операндов

Параллельный сумматор с последовательным переносом. Сумматор для параллельных операндов строиться как цепочка одноразрядных, соединительных последовательно по цепям переноса. Такой сумматор четырехразрядных операторов может быть построен по следующей схеме (рис. 3.25)

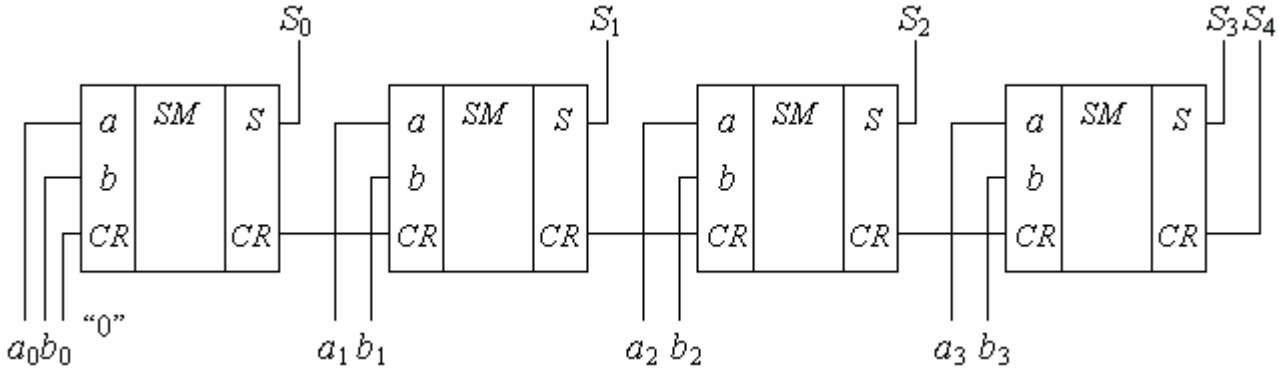


Рис. 3.25. Схема арифметического сумматора четырехразрядных двоичных чисел

На входы a_i, и b_i каждого из сумматоров подаются разрядные коэффициенты (цифры 1 или 0) слагаемых, на вход CR_{i-1} подается сигнал с выхода переноса CR_{i-1} предыдущего сумматора (на вход переноса сумматора младшего разряда постоянно подается сигнал «0»). Сумма двух чисел формируется на выходах S_i одноразрядных сумматоров, причем сигнал старшего разряда суммы формируется на выходе переноса одноразрядного сумматора старшего разряда. Такие сумматоры четырехразрядных двоичных операндов входят в серии ТТЛ ИМС под шифром ИМ3, ИМ6, ИМ7. Основной параметр быстродействия сумматора – задержка суммирования, определяется выражением:

$$t_{З.S} \cong 4 \cdot t_{З.\,SM} ,$$

где $t_{З.\,SM}$ - время задержки одноразрядного сумматора.

Следовательно, чем больше разрядность чисел, которые необходимо сложить, тем большее время затрачивает сумматор на эту операцию. В ряде случаев такое быстродействие оказывается недостаточным, и необходимо принимать меры к его повышению. Основные направления повышения быстродействия: а) уменьшение времени распространения сигнала переноса по тракту внутри одноразрядного сумматора; б) уменьшение или исключение последовательного распространения сигнала переноса вдоль разрядной сетки сумматора.

Уменьшение времени переноса внутри одноразрядного сумматора приводит к проблеме создания схемы, содержащей минимум оборудования и обладающей минимальными временными задержками. В настоящее время не существует методов синтеза таких одноразрядных сумматоров. Поэтому на практике используются схемы, полученные эмпирическим путем. Уменьшение или исключение последовательного распространения сигнала переноса вдоль разрядной сетки осуществляется несколькими способами.

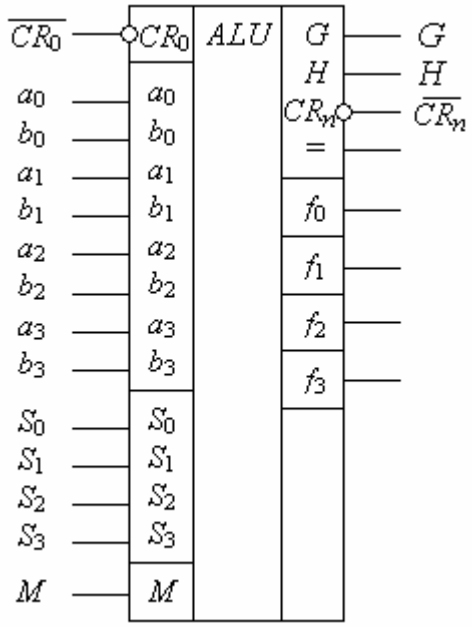


Рис.3.26. Условное графическое обозначение АЛУ

При построении сумматора с групповым переносом все разряды устройства разбиваются на группы, в каждой из которых дополнительными логическими элементами создается так называемый ускоренный перенос.

В сумматорах с параллельным переносом исключение последовательного распространения сигнала переноса осуществляется организацией отдельного от

суммирования сигнала переноса дополнительно введенной логической схемой ускоренного переноса, на входы которой поступают сигналы всех разрядов слагаемых, а на выходах формируются сигналы переноса во все разряды.

Арифметико-логические устройства. Многие серии ИМС имеют в своем составе арифметико-логические устройства, выполняющие над операндами ряд логических и арифметических действии. Примером АЛУ может служить ИМС ИПЗ (рис. 3.26), входящая в серии ТТЛ.

Таблица 3.12. Функции АЛУ

Выбор функции				Логические операции	Арифметические операции M=0	
S_3	S_2	S_1	S_0	M=1	CR_0=0	CR_0=1
0	0	0	0	$\overline{A}$	A	A+1
0	0	0	1	$\overline{A \vee B}$	$A \vee B$	$(A \vee B)$+1
0	0	1	0	$\overline{A}B$	$A \vee B$	$(A \vee \overline{B})$+1
0	0	1	1	0	1	0
0	1	0	0	$\overline{AB}$	$A+\overline{AB}$	$A+\overline{AB}+1$
0	1	0	1	$\overline{B}$	$(A \vee B)+\overline{AB}$	$(A+B)+A\overline{B}+1$
0	1	1	0	$A \oplus B$	A-B-1	A-B
0	1	1	1	$A\overline{B}$	$A\overline{B}$-1	$\overline{AB}$
1	0	0	0	$\overline{A} \vee B$	$A+AB$	$A+AB+1$
1	0	0	1	$A \oplus B$	$A+B$	$A+B+1$
1	0	1	0	B	$(A \vee \overline{B})+AB$	$(A \vee \overline{B})+AB+1$
1	0	1	1	AB	AB-1	AB
1	1	0	0	1	$A+A$	$A+A+1$
1	1	0	1	$A \vee \overline{B}$	$(A \vee B)+A$	$(A \vee B)+A+1$
1	1	1	0	$A \vee B$	$(A \vee \overline{B})+A$	$(A \vee B)+A+1$
1	1	1	1	A	A-1	A

Схема АЛУ довольно громоздка и в данном пособии не приводится. Это АЛУ (английский термин — *arithmetic logic unit*) предназначено для обработки четырехразрядных операндов и имеет кроме входов этих операндов a_i, b_i (i = 0,…, 3), входы выбора режима S_i (i = 0,…, 3) и вход M (*mode control*), сигнал на котором задает характер выполняемых операций. Если M = 0, то в устройстве выполняются арифметические операции. Если M = l, то в устройстве блокируются межразрядные переносы и в зависимости от комбинации сигналов S_i (i = 0,…, 3) может выполняться любая из 16 логических операций АЛУ.

АЛУ имеет инвертированный вход переноса $\overline{CR_n}$ от младших разрядов, что позволяет наращивать разрядность операндов (рис. 3.27).

Полный перечень операций, выполняемых АЛУ, приведен в табл. 3.12.

Логические операции выполняются независимо в каждом разряде (поразрядно). Арифметические операции выполняются с учетом переносов и заемов.

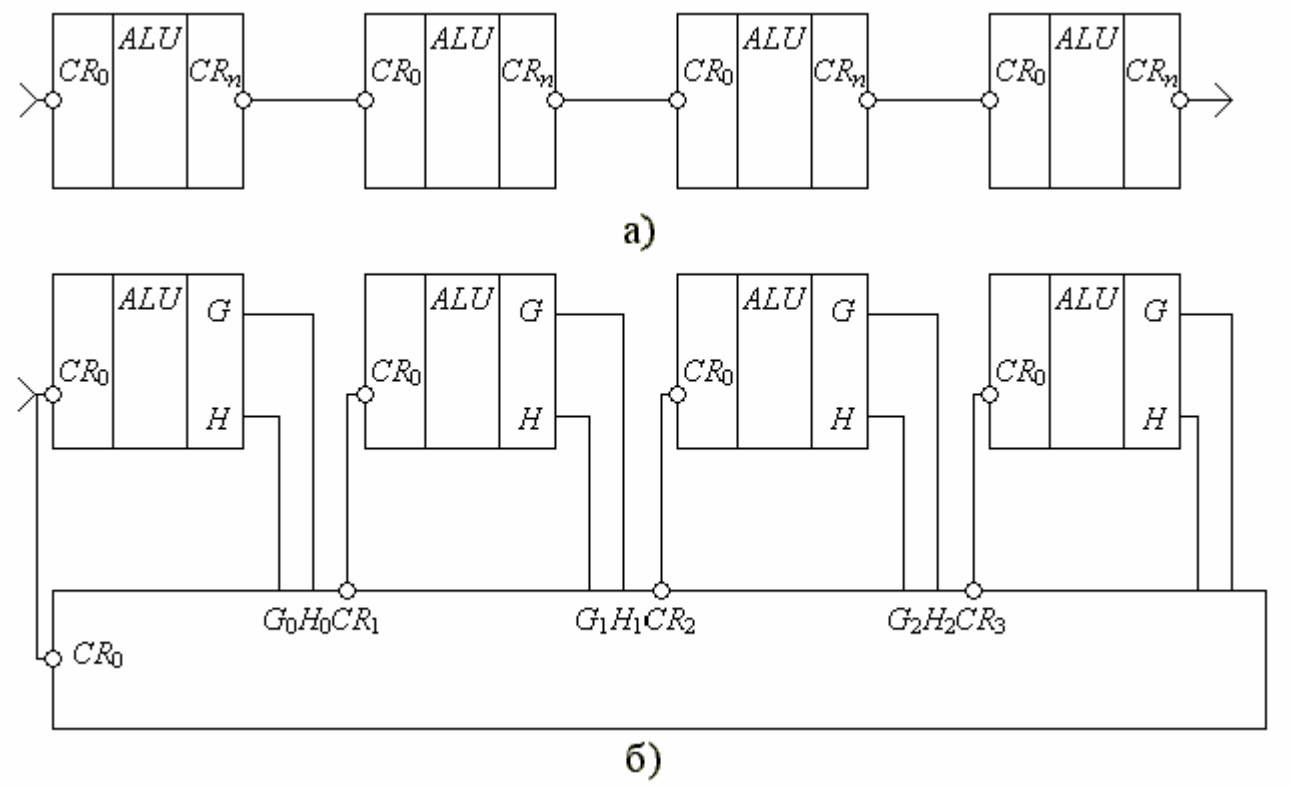


Рис. 3.27. Наращивание разрядности АЛУ с последовательным (а) и ускоренным (б) переносом

Оба типа операции могут встречаться одновременно. Например, запись $(A \vee \overline{B})+AB$ означает, что вначале поразрядно выполняются операции инвертирования ($\overline{B}$), логического сложения ($A \vee \overline{B}$) и логического умножения (AB), а затем полученные указанным образом два четырехразрядных операнда складываются арифметически (с учетом переносов). Операция вычитания в АЛУ заменяется сложением уменьшаемого с вычитанием, представленным в дополнительном коде. Дополнительный код образуется из обратного (инверсного) кода добавлением к нему единицы. Следует учитывать, что результат арифметических действий на выходе АЛУ в этом случае будет представляться в обратном коде. При обработке операндов большей разрядности АЛУ соединяются последовательно (рис. 3.27а).

3.6. Вопросы и упражнения

1.Охарактеризуйте типовые (стандартные) комбинационные цифровые устройства.

2.Дайте определение кодопреобразователя.

3.Какие функции выполняет шифратор?

4.Какие функции выполняет дешифратор?

5.Составьте логические функции, описывающие работу дешифратора при появлении сигналов на выходах 2 и 18, 7 и 23, 15 и 31.

6.Постройте устройство для преобразования десятичных цифр 0, 1, …, 9 в шестнадцатеричный код.

7.Нарисуйте схему дешифратора (4х16), используя только дешифраторы (2х4) с входами стробирования.

8.Реализуйте логическую функцию $f(x_0, x_1, x_2, x_3) = x_1 x_0 \vee x_3 \bar{x}_2 \bar{x}_1 x_0$ на дешифраторе (4х16).

9.Какие функции выполняют мультиплексоры?

10.Синтезируйте схему демультиплексора $(8 \to 1)$ на элементах И - НЕ.

11.Реализуйте логическую функцию $f(x_0, x_1, x_2) = x_2 \bar{x}_1 x_0 \vee \bar{x}_2 x_1 \bar{x}_0$ на мультиплексоре $(8 \to 1)$.

12.Дайте определение и заполните таблицы истинности одноразрядного полного сумматора. Проведите их синтез на элементах Пирса.

13.Как определяется время суммирования двоичных чисел для сумматоров? Как ускорить процесс суммирования?

14.Синтезируйте цифровой компаратор двухразрядных двоичных чисел.

15. Синтезируйте цифровой компаратор, имеющий три выхода $y_{A>B}$, $y_{A=B}$, $y_{A<B}$ на основе компаратора с двумя выходами $y_{A<B}$, $y_{A=B}$.

4. ТРИГГЕРЫ И РЕГИСТРЫ

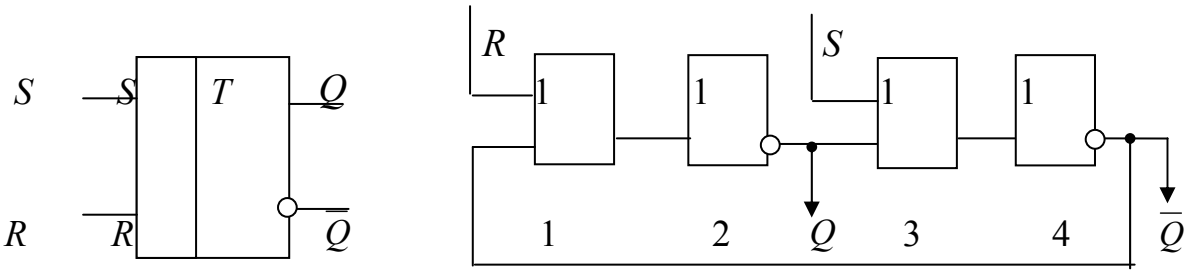
4.1. Основные определения триггера

Триггеры представляют собой простейшие конечные автоматы с памятью, обладающие двумя выходами и одним или несколькими входами.

Триггером называют устройство, которое может находиться в одном из двух устойчивых состояний и переходить из одного состояния в другое под воздействием входного сигнала. Состояние триггера определяется уровнем выходного сигнала. Триггер является базовым элементом последовательностного цифрового устройства (ПЦУ). В нем может храниться либо 0, либо 1.

Для удобства использования в схемах ПЦУ триггеры имеют два выхода: 1) прямой Q (выход 1); 2) инверсный $\overline{Q}$ (выход 0). Состоянию триггера 1 соответствует на выходе Q высокий уровень сигнала (1), а на выходе $\overline{Q}$ – низкий (0). Состоянию триггера 0 соответствует на выходе Q низкий уровень сигнала (0), а на выходе $\overline{Q}$ – высокий (1).

Входы триггера делятся на информационные и вспомогательные (управляющие). Сигналы, поступающие на информационные входы, управляют состоянием триггера. Сигналы на вспомогательных входах используются для предварительной установки триггера в требуемое состояние и синхронизации. Вспомогательные входы могут использоваться и в качестве информационных. Число входов триггера зависит от его структуры и назначения. Информационные входы триггера принято обозначать буквами S, R, J, K, D, T, а управляющие входы – C, V.



а) б)

Рис. 4.1. Стандартное обозначение триггера (а) и построение триггера

с использованием схемы с обратной связью с выхода на вход (б)

Для триггера имеется стандартное обозначение (рис. 4.1а). Здесь S (*Set*) и R (*Reset*) являются информационными входами, Q и $\overline{Q}$ - выходами, T обозначает триггер. Инверсный выход $\overline{Q}$ отмечен кружком.

Триггер может быть построен из логических элементов, в виде схемы с обратной связью с выхода на вход, как это показано на рис.4.1б. Пусть триггер находится в состоянии 0 (Q=0, $\overline{Q}$ =1), и на входах R, S заданы сигналы 0. Это состояние устойчиво и будет сохраняться. Действительно, единичный выходной сигнал $\overline{Q}$ поступает на вход схемы ИЛИ с номером 1. На выходе этой схемы с учетом R =0 будет получен единичный выходной сигнал. Этот сигнал поступит на вход схемы НЕ с номером 2. В результате на выходе этой схемы и, следовательно, на выходе Q будет по-прежнему сигнал 0. С выхода схемы 2 нулевой сигнал поступает на один из входов схемы ИЛИ с номером 3. На второй вход S этой схемы подается также нулевой сигнал. В результате и на выходе схемы 3 будет нулевой сигнал. Этот сигнал поступит на вход схемы НЕ с номером 4, и на выходе ее будет единичный сигнал, т.е. подтверждается нулевое состояние триггера (Q=1, $\overline{Q}$ =0). Аналогично, если бы триггер находился в состоянии 1 и на входах R, S были бы заданы нулевые сигналы, то такое состояние также сохранилось бы.

Пусть теперь на вход триггера, находящегося в нулевом состоянии (Q=0, $\overline{Q}$ =1), поступают информационные сигналы S =1, R =0. Под воздействием сигнала S на выходе схемы ИЛИ с номером 3 будет единичный сигнал. Тогда на выходе схемы НЕ с номером 4 будет сигнал 0. Следовательно, триггер перейдет в единичное состояние (Q =1, $\overline{Q}$ =0). Действительно, выходной сигнал $\overline{Q}$ =0 с выхода схемы НЕ с номером 4 будет поступать на вход схемы ИЛИ с номером 1. С выхода этой схемы нулевой сигнал поступит на вход схемы НЕ с номером 2. В результате на выходе этой схемы и на выходе Q появится единичный сигнал, который поступит на вход схемы ИЛИ с номером 3. Если теперь снять единичный сигнал с входа S, то триггер останется в единичном состоянии. Для установки триггера в нулевое состояние необходимо на вход R подать единичный сигнал, а на вход S – нулевой. Переключение триггера из единичного состояния в нулевое происходит при входах S = 0, R = 1.

Используя способ построения триггера с использованием обратной связи с выхода на вход можно строить другие схемы триггеров.

4.2. Классификация триггеров

Триггеры можно классифицировать по способу приема информации, принципу построения, функциональным возможностям.

По способу приема информации триггеры подразделяются на асинхронные и синхронные. Асинхронные триггеры воспринимают информационные сигналы и реагируют на них в момент появления на входах триггера. Синхронные триггеры реагируют на информационные сигналы при наличии разрешающего сигнала на специальном управляющем входе C, называемом входом синхронизации. Синхронные триггеры подразделяются на триггеры со статическим и динамическим управлением по входу C. Триггеры со статическим управлением воспринимают информационные сигналы при подаче на C-вход уровня 1 (прямой C-вход) или 0 (инверсный C-вход). Триггеры с динамическим управлением воспринимают информационные сигналы при изменении сигнала на C-входе от 0 к 1 (прямой динамический C-вход) или от 1 к 0 (инверсный динамический C-вход).

По принципу построения триггеры со статическим управлением можно разделить на одноступенчатые и двухступенчатые. Одноступенчатые триггеры характеризуются наличием одной ступени запоминания информации. В двухступенчатых триггерах имеются две ступени запоминания информации. Вначале информация записывается в первую ступень, а затем переписывается во вторую и появляется на выходе.

Особенность триггеров с двухступенчатым запоминанием информации состоит в том, что они содержат две триггерные структуры (*MS*-структуры) : одна из них образует так называемый ведущий триггер (*M*), другая – ведомый триггер (*S*) (рис. 4.2). Оба триггера функционируют как синхронные триггеры со статическим управлением. Если на синхронизирующем входе C=1, то ведущий триггер устанавливается в состояние, соответствующее сигналам, поступающим на информационные входы. Ведомый триггер, имеющий инверсный синхронизирующий вход, при этом невосприимчив к информации, поступающей на его вход с выхода ведущего триггера. Он продолжает находиться в состоянии, в которое был ранее установлен (в предыдущем тактовом периоде).

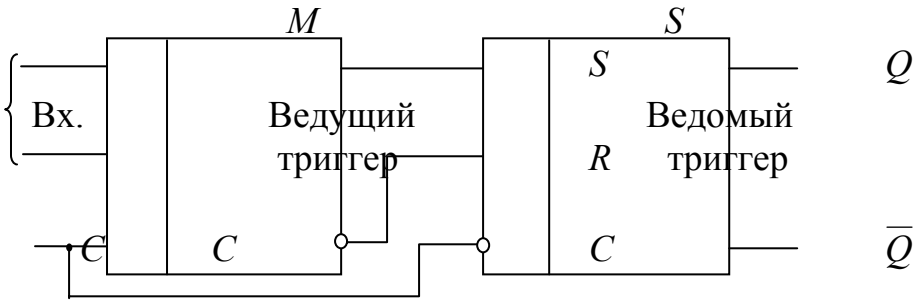


Рис. 4.2. *MS*-структура триггера

При изменении значения C (с C=1 на C=0) ведущий триггер отключается от информационных входов и перестает реагировать на изменение значений сигналов на эти входах; ведомый триггер устанавливается в состояние, в котором находится ведущий триггер. С этого момента на выходах устанавливаются значения, соответствующие входным сигналам, поступающим к моменту рассматриваемого фронта сигнала на синхронизирующем входе.

Таким образом, управление процессами в триггере с двухступенчатым запоминанием информации за время тактового периода осуществляется двумя фронтами сигнала на синхронизирующем входе: на положительном фронте происходит установка ведущего триггера, на отрицательном фронте – ведомого триггера.

Триггеры характеризуются быстродействием, чувствительностью, потребляемой мощностью, помехоустойчивостью, функциональными возможностями. Быстродействие определяется максимальной частотой переключения состояний триггера и достигает сотен мегагерц. Чувствительность триггера определяется наименьшим напряжением на входе (пороговым напряжением), при котором происходит переключение. Помехоустойчивость характеризует способность триггера нормально работать в условиях помех. Функциональные возможности триггера характеризуются числом входных сигналов.

По функциональным возможностям различаются:

1) триггер с раздельной установкой состояний 0 и 1 (RS-триггер);
2) триггер с приемом информации по одному входу D (D-триггер или триггер задержки);
3) триггер со счетным входом T (T-триггер);
4) универсальный триггер с информационными входами J и K (JK-триггер).

Для полного описания триггера достаточно задать его структурную схему из базовых логических элементов и закон функционирования. В качестве базовых логических элементов можно использовать элементы ИЛИ - НЕ, И - НЕ. Поскольку триггер является простейшим автоматом Мура, закон функционирования может быть задан таблицей переходов, в которой входные сигналы в момент их изменения и состояние триггера обозначены индексом t, а после переключения – индексом t+1. Рассмотрим триггеры основных видов.

4.3. Асинхронный *RS*-триггер с прямыми входами

Эти триггеры имеют два информационных входа S и R, используемые для установки соответственно 1 и 0, а также два выхода: прямой Q и инверсный $\overline{Q}$. RS-триггер построен на логических элементах ИЛИ - НЕ, с обратной

связью с замкнутым контуром (рис. 4.3а). Графическое обозначение *RS*-триггера приведено на рис. 4.3, б, таблица переходов *RS*-триггера в табл.4.1, закон функционирования может быть описан табл. 4.2.

Таблица 4.1. Таблица переходов *RS*- триггера

S^t	R^t	Q^t	Q^{t+1}	Режим
0	0	Q^t	Q^t	Хранение
1	0	Q^t	1	Установка 1
0	1	Q^t	0	Установка 0
1	1	Q^t	-	Запрещено

Как следует из таблицы, при комбинации сигналов S^t=1, R^t=0 триггер переходит в состояние 1 (Q^{t+1}=1) независимо от предыдущего состояния Q^t. При наборе сигналов S^t=0, R^t=1 триггер устанавливается в состояние 0 (Q^{t+1}=0). Комбинация сигналов S^t=0, R^t=0 не изменяет состояния триггера, т. е. состояние триггера в момент t+1 равно $Q^{t+1} = Q^t$.

Таблица 4.2. Закон функционирования *RS*- триггера с прямыми входами

S^t	R^t	Q^t	Q^{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

Набор сигналов S^t=1, R^t=1 является запрещенным, так как он приводит к неопределенности состояния триггера.

В *RS*-триггерах с прямыми входами управляющим воздействием обладают единичные уровни сигналов. Сигналы, которые приводят к переключению элемента, называются активными. Для элементов ИЛИ - НЕ, используемых для построения *RS*-триггера, активным сигналом является уровень 1.

Таблица состояний триггера в момент t+1 может быть задана с помощью карт Карно (рис.4.4).

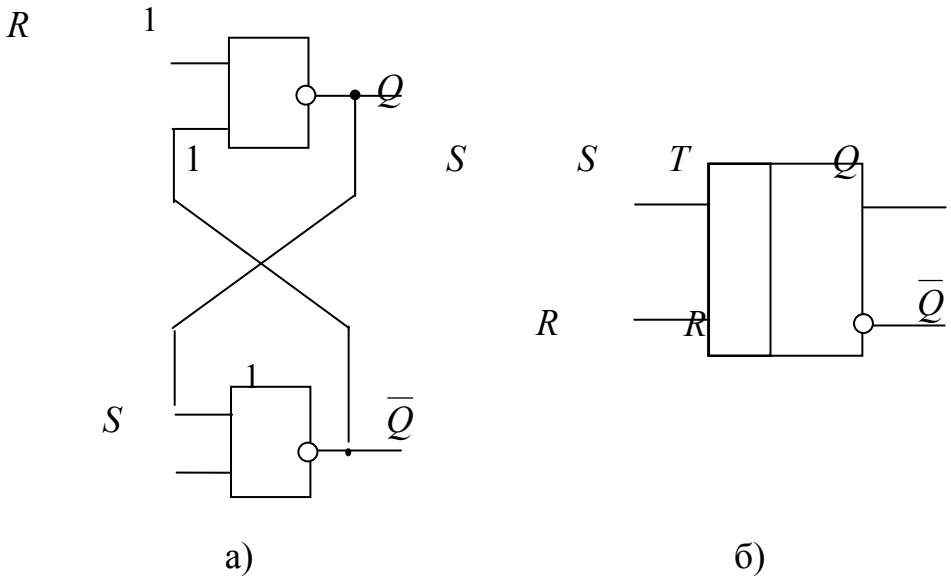
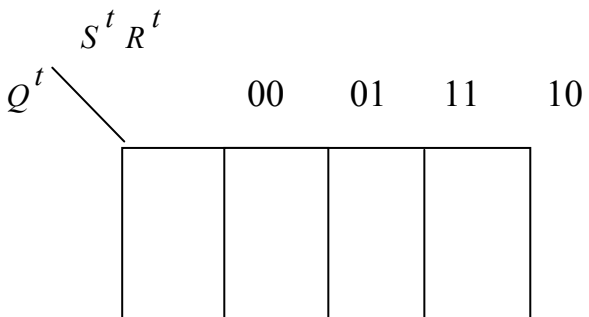


Рис. 4.3. Схема (а) и условное обозначение (б) *RS*-триггера с прямыми входами

Используя карту Карно, можно построить минимизированную логическую функцию для описания функционирования *RS*-триггера:

$$Q^{t+1} = S^t \vee R^t \cdot \overline{Q}^t.$$

RS-триггер устанавливается в состояние 1 при воздействии входного уровня S^t=1 либо остается в состоянии 1, если R^t=0 и триггер был в состоянии Q^t=1.



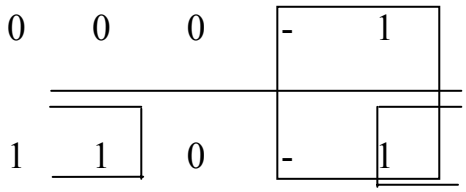


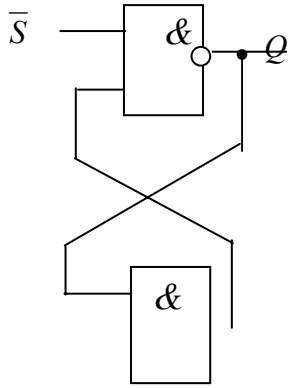
Рис. 4.4. Карта Карно асинхронного *RS*-триггера на элементах ИЛИ-НЕ

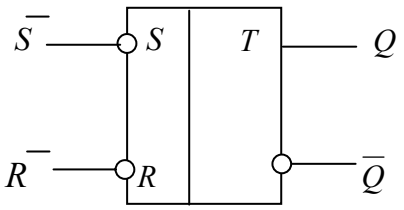
4.4. Асинхронный *RS*- триггер с инверсными входами

Триггеры такого типа строятся на логических элементах И - НЕ. В этом случае уровень 0 является активным входным сигналом, а уровень 1 – пассивным. Информационные входы и соответствующие сигналы таких триггеров принято обозначать как инверсные ($\overline{S}$, $\overline{R}$). В этом случае триггер с инверсными входами будет описываться той же таблицей, что и триггер с прямыми входами. Входные сигналы $S=1$, $R=1$ будут активными, как в триггере с прямыми входами. Схема *RS*-триггера с инверсными входами приведена на рис. 4.5а, условное графическое обозначение такого триггера дано на рис. 4.5б.

Закон функционирования *RS*-триггера на элементах И - НЕ описывается табл. 4.3 из которой следует, что комбинация $\overline{S}=\overline{R}=0$ является запрещенной, а набор $\overline{S}=\overline{R}=1$ – нейтральным. Следовательно, если нулевые сигналы на обоих входах триггера на элементах ИЛИ - НЕ составляют нейтральную комбинацию, то для триггера на элементах И - НЕ они запрещены. Эти особенности следует учитывать при использовании в ПЦУ триггеров на логических элементах разных типов.

Быстродействие асинхронного *RS*-триггера определяется задержкой установки его состояния $t_Т$, равной сумме задержек передачи сигнала через цепочку логических элементов с задержкой $t_Э$ в каждом. В данном случае $t_Т = 2t_Э$.





а) б)

Рис. 4.5. Схема *RS*-триггера с инверсными входами (а) и условное обозначение *RS*-триггера с инверсными входами (б)

Таблица 4.3. Закон функционирования *RS*-триггера с инверсными входами

$\overline{S}^{\,t}$	$\overline{R}^{\,t}$	Q^{t}	Q^{t+1}
0	0	0	-
0	0	1	-
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

4.5. Синхронный *RS*-триггер со статическим управлением

Синхронный *RS*-триггер отличается от асинхронного наличием C-входа, на который поступают синхронизирующие (тактовые) сигналы. Условное графическое обозначение синхронного *RS*-триггера на элементах И-НЕ приведено на рис. 4.6б.

Синхронный триггер состоит из асинхронного *RS*-триггера и комбинационного цифрового устройства (КЦУ) (рис. 4.6а). Как видно из рис. 4.6а синхронный триггер построен из элементов И - НЕ и состоит из КЦУ с тремя входами S, C, R и двумя выходами. КЦУ состоит из двух логических элементов И – НЕ. Элемент памяти (ЭП) представляет собой асинхронный *RS* -триггер на элементах И - НЕ. При C=0 входные логические элементы схемы КЦУ блокированы. Их выходы принимают значения 1 и не зависят от входных сигналов S и R. Для асинхронного RS-триггера на элементах И -НЕ набор единичных сигналов является нейтральным. Триггер будет сохранять свое состояние. При C=1 входные логические элементы схемы КЦУ открыты для передачи информационных сигналов R и S на входы асинхронного RS-триггера. Следователь-

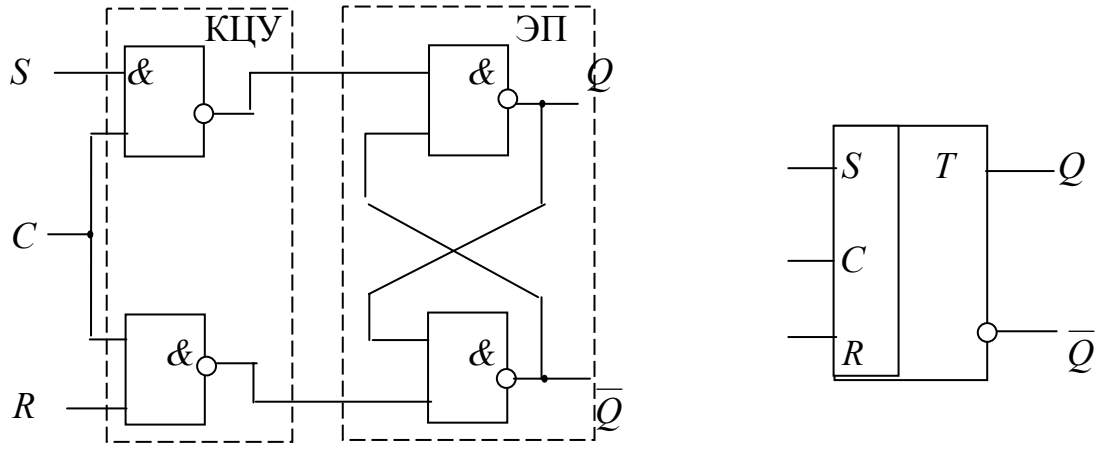
но, синхронный триггер при наличии разрешающего сигнала будет работать по правилам для асинхронного триггера. Закон функционирования синхронного RS-триггера на элементах И - НЕ может быть задан табл. 4.4. Карта Карно, описывающая работу синхронного RS-триггера на элементах И - НЕ, представлена рис. 4.7. Используя карту Карно, можно получить минимальную булеву функцию для описания работы синхронного RS-триггера:

$$Q^{t+1} = \overline{R}^t \cdot Q^t \vee \overline{C}^t \cdot Q^t \vee C^t \cdot S^t = Q^t \cdot (\overline{R}^t \vee \overline{C}^t) \vee C^t \cdot S^t.$$

RS-триггера на элементах И – НЕ

Номер набора	C^t	S^t	R^t	Q^t	Q^{t+1}
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	-
15	1	1	1	1	-

Легко увидеть, что синхронный RS-триггер принимает состояние 1, если на входы C и S поступают уровни 1, или сохраняет единичное состояние Q^t при отсутствии единичных сигналов на входе C или R.



а) б)

Рис. 4.6. Схема синхронного *RS*-триггера со статическим управлением (а)

и условное обозначение синхронного *RS*-триггера со статическим управлением (б)

Синхронные *RS*- триггеры строятся и на логических элементах ИЛИ – НЕ, И-ИЛИ-НЕ, а также на их сочетаниях.

Общее время установления состояния триггера $t_Т$ равно сумме задержек передачи сигнала через цепочку из трех логических элементов с задержкой $t_Э$ в каждом: $t_Т = 3t_Э$. При этом длительность сигнала t_C на входе C должна превышать время переключения $t_Т$: $t_C \geq t_Т = 3\ t_Э$.

Длительность паузы $t_П$ между двумя сигналами на входе C должна быть достаточной для переключения входных элементов в КЦУ (рис.4.5а): $t_П \geq t_Э$.

Следовательно, минимальный период повторения синхронизирующих сигналов на входе C равен $4t_Э$, а наибольшая частота переключения триггера $f = 1/(4\ t_Э)$.

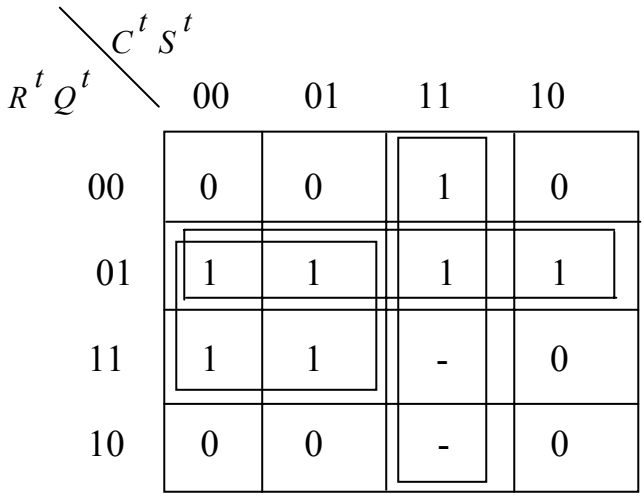


Рис. 4.7. Карта Карно синхронного *RS*-триггера

4.6. Универсальный *JK*-триггер

Универсальный *JK* -триггер, схема которого приведена на рис.4.9а, представляет собой двухступенчатый синхронный триггер. Как видно из рисунка, *JK*-триггер состоит из двух *RS*-триггеров с инверсными входами и двух КЦУ, каждое из которых содержит две схемы И–НЕ, с тремя входами каждая.

Закон функционирования *JK*-триггера задается табл. 4.5. Используя карту Карно (рис. 4.8), можно построить минимизированную логическую функцию для описания функционирования *JK*-триггера:

$$Q^{t+1} = C \cdot J^t \cdot \overline{Q}^t \vee C \cdot K^t \cdot Q^t.$$

Таблица 4.5. Закон функционирования *JK*-триггера

C^t	S^t	R^t	Q^t	Q^{t+1}
0	0	0	Q^t	Q^t
0	0	1	Q^t	Q^t
0	1	0	Q^t	Q^t
0	1	1	Q^t	Q^t
1	0	0	Q^t	Q^t
1	0	1	Q^t	0
1	1	0	Q^t	1
1	1	1	Q^t	$\overline{Q}^t$

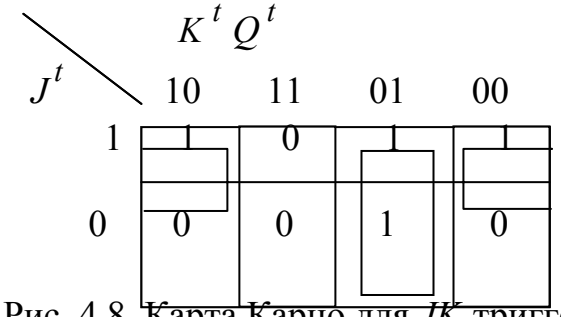


Рис. 4.8. Карта Карно для *JK*-триггера

JK-триггер отличается от синхронного *RS*-триггера тем, что не имеет запрещенных комбинаций сигналов на входах *J* и *K*. При *J* = 1 и *K* = 1 триггер изменяет свое состояние на противоположное.

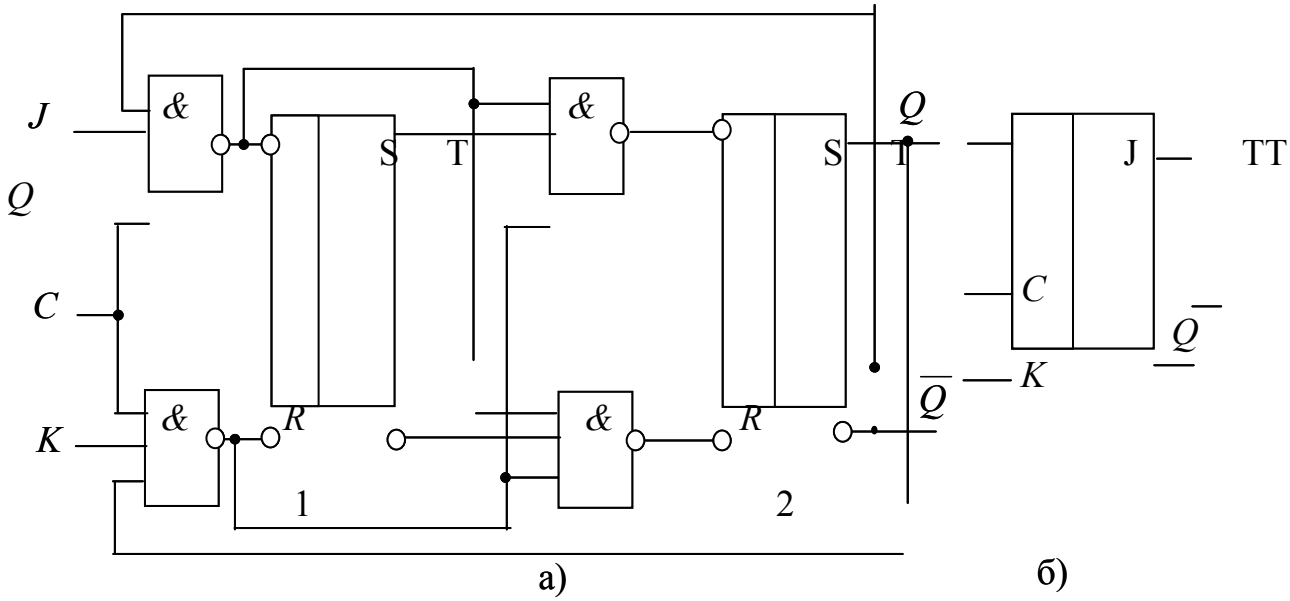


Рис. 4.9. Схема (а) и условное обозначение (б) *JK*-триггера

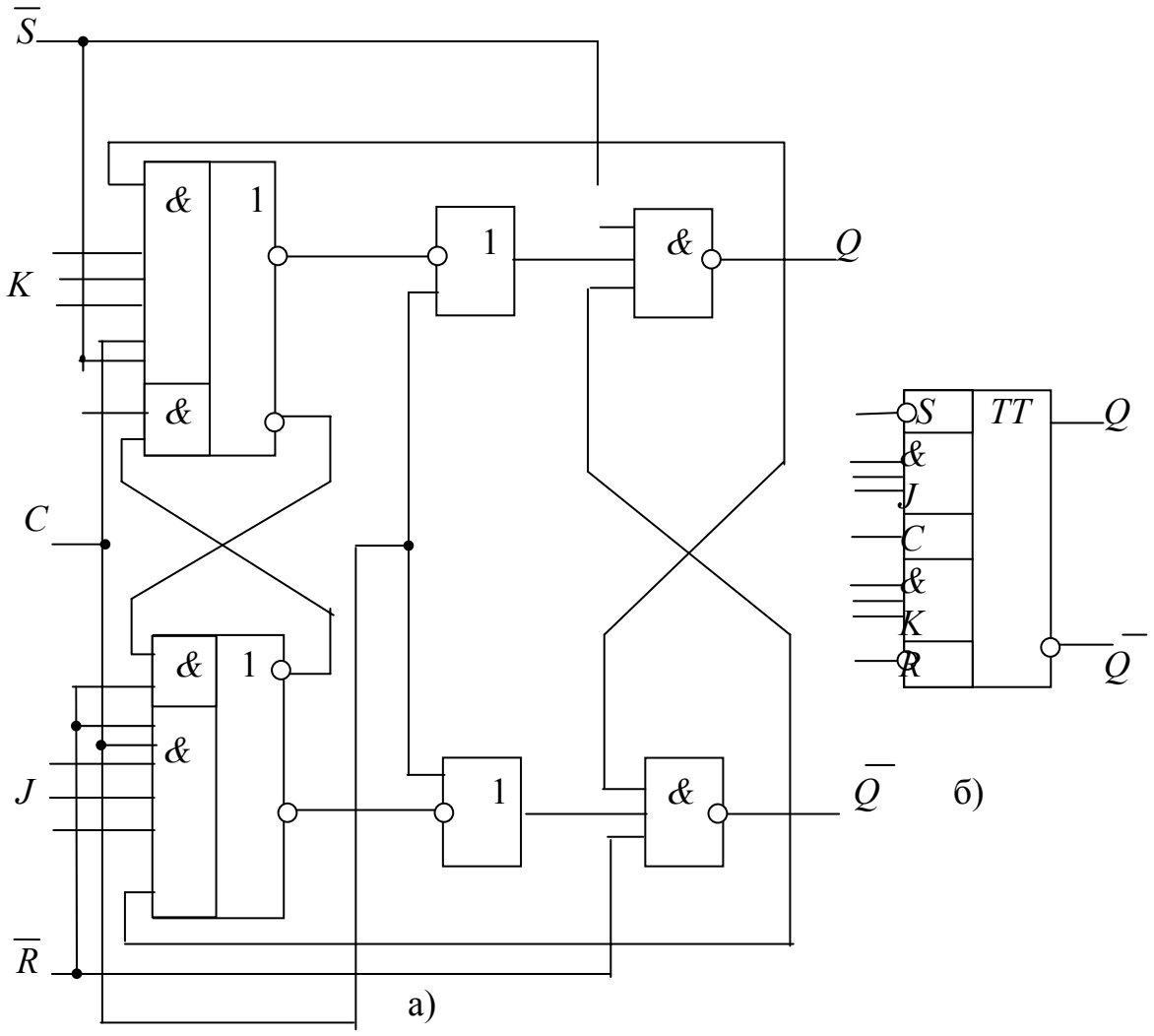


Рис. 4.10. Схема (а) и условное обозначение (б) *JK*-триггера со входной логикой

Условное обозначение *JK*- триггера представлено на рис. 4.9б. Одним из широко используемых вариантов построения является схема универсального *JK*- триггера с входной логикой (рис. 4.10а). Условное обозначение триггера дано на рис. 4.10б. При $C = 1$ вторая ступень блокирована, а информационные сигналы устанавливают состояние первой ступени. При сигнале $C = 0$ вторая ступень воспринимает состояние первой. Требуемое начальное состояние триггера устанавливается с помощью сигналов, подаваемых на входы *RS*-триггеров первой и второй ступени. В схеме триггера имеется по три конъюнктивно связанных входа *J* и *K***.** Другими словами, в структурную схему триггера встроены логические элементы. Их можно использовать для построения сложных схем.

4.7. *T* – триггер

T-триггеры, или триггеры со счетным входом, могут быть получены из *JK*-триггера, при соединении обоих информационных входов *J* и *K* и подаче на них уровня 1 (рис. 4.11а). Функционирование *T*- триггера описывается табл. 4.6. В качестве счетного входа *T*, используется вход *C*. При подаче сигнала на вход *T*-триггер будет переключаться в состояние, противоположное предыдущему. *JK* - триггер в таком режиме выполняет функцию *T*-триггера. Функционирование *T*-триггера можно определить логической функцией:

$$Q^{t+1} = Q^t \cdot \overline{T}^t \vee \overline{Q}^t \cdot T^t, \text{ при } J = K = 1.$$

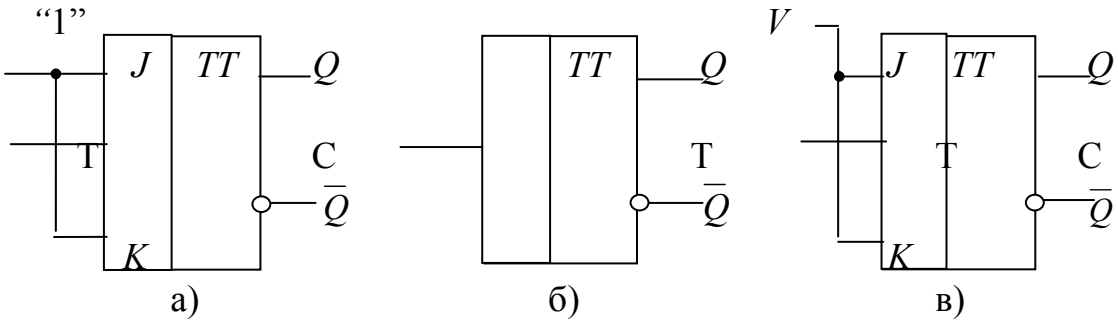


Рис. 4.11. Схема T-*триггера на* JK-*триггере (а),*
T-*триггера (б) и* TV-*триггера (в)*

Следует заметить, что уровень 1 для входов *J* и *K* является пассивным сигналом. Поэтому для получения *T*-триггера из *JK*-триггера можно не предусматривать входы *J* и *K* в схеме (рис. 4.11б).

Разновидностью *T*-триггера является *TV*-триггер, в котором вход *V* является управляющим. При *V* = 1 *TV*-триггер превращается в *T*-триггер. При *V*=0 *TV*-триггер сохраняет свое состояние неизменным (рис. 4.11в).

Таблица 4.6. Функционирование *T*-триггера на *JK*-триггере

T^t	J^t	K^t	Q^t	Q^{t+1}
0	1	1	Q^t	Q^t
1	1	1	Q^t	$\overline{Q}^t$

4.8. *D*-триггер

D-триггер, или триггер задержки, имеет один информационный вход *(D*-вход) и вход для синхронизации *C*. Основное назначение *D*-триггера – задержка сигнала, поданного на вход *D. D*-триггер может быть получен из *JK*-триггера соединением входа *J* со входом *K* через инвертор НЕ. Такой вход называется *D*-входом. Схема *D*-триггера представлена на рис. 4.12а. Условное обозначение *D*-триггера приведено на рис. 4.12б.

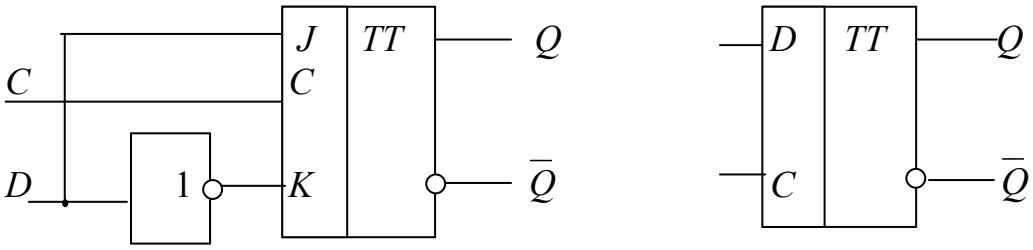


Рис. 4.12. Схема (а) и условное обозначение (б) *D*-триггера

Функционирование *D*-триггера описывается табл. 4.7.

Таблица 4.7. Функционирование *D*-триггера

C^t	D^t	Q^{t+1}
0	0	Q^t
0	1	Q^t
1	0	0
1	1	1

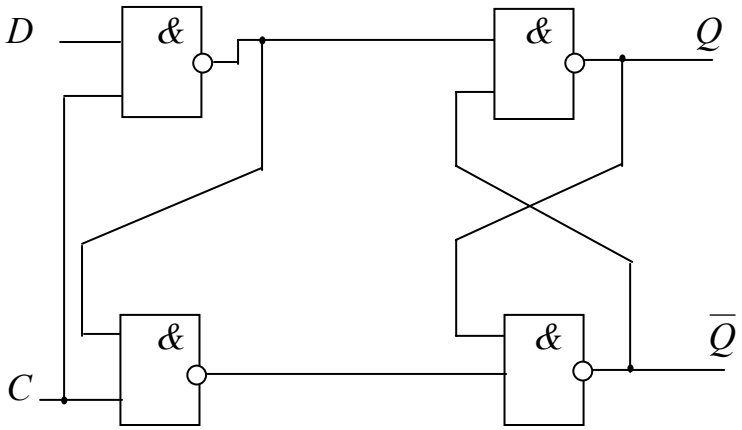


Рис. 4.13. Схема *D*-триггера

Под действием синхросигнала *C* информация, поступающая на вход *D*, принимается в триггер, но на выходе *Q* появляется с задержкой на один такт, т.е. информация, поступающая на вход *D* в момент поступления синхроимпульса на вход *C,* заносится в триггер, а в момент его окончания устанавливается на выходе *Q* и может быть использована в следующем такте.

Функция переходов асинхронного *D*-триггера $Q^{t+1} = D^t$.

Для синхронного *D*-триггера $Q^{t+1} = (C \cdot Q \ \vee C \cdot D)^t$.

Схема *D*-триггера приведена на рис. 4.13. Чаще *D*-триггеры строятся по двухступенчатой схеме.

4.9. Синхронный триггер с динамическим управлением

Синхронный триггер с динамическим управлением по входу *C* воспринимает информацию для изменения состояния лишь тогда, когда на *C*-входе совершается переход с уровня 0 на уровень 1, либо наоборот. Для получения *RS*-триггера с динамическим входом достаточно построить схему, показанную на рис. 4.14.

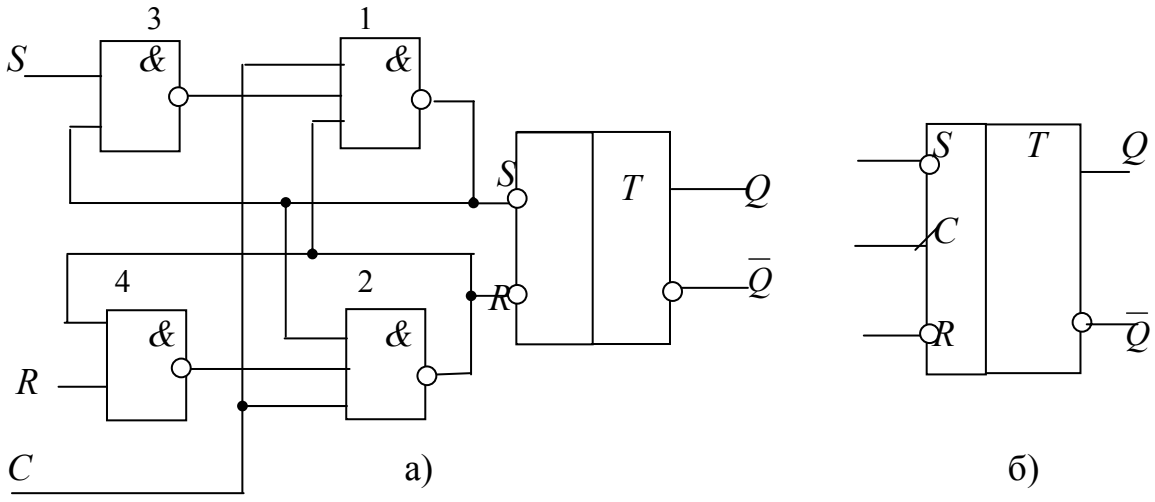


Рис. 4.14. RS-триггер (а) и его условное обозначение (б) с динамическим входом на элементе И – НЕ

Если при *C*=0 на информационные входы поступили какие - либо уровни *S* и *R***,** то при смене уровня на входе *C* с 0 на 1 на выходе элемента 1 будет уровень 0, который поступает на вход элемента 3 и обеспечивает на его выходе уровень 1 независимо от последующих значений уровня на входе *S*. Вход *S* логически отключается, и никакие изменения уровней на входах *S* и *R* триггер не воспринимает, пока на входе *C* не произойдет переход с уровня 0 на уровень 1.

Аналогично можно построить схему *RS*-триггера с динамическим входом на элементах ИЛИ - НЕ (рис. 4.15а). Здесь информация воспринимается триггером со входов *S* и *R* при смене уровней *C*=1 на *C*=0. Условное обозначение такого триггера представлено на рис. 4.15б. Схема *D*-триггера с динамическим входом приведена на рис. 4.16а**,** а его условное обозначение дано на рис. 4.16б. Прием в триггер информации со входа *D* происходит в момент смены на входе *C* уровня 0 на уровень 1.

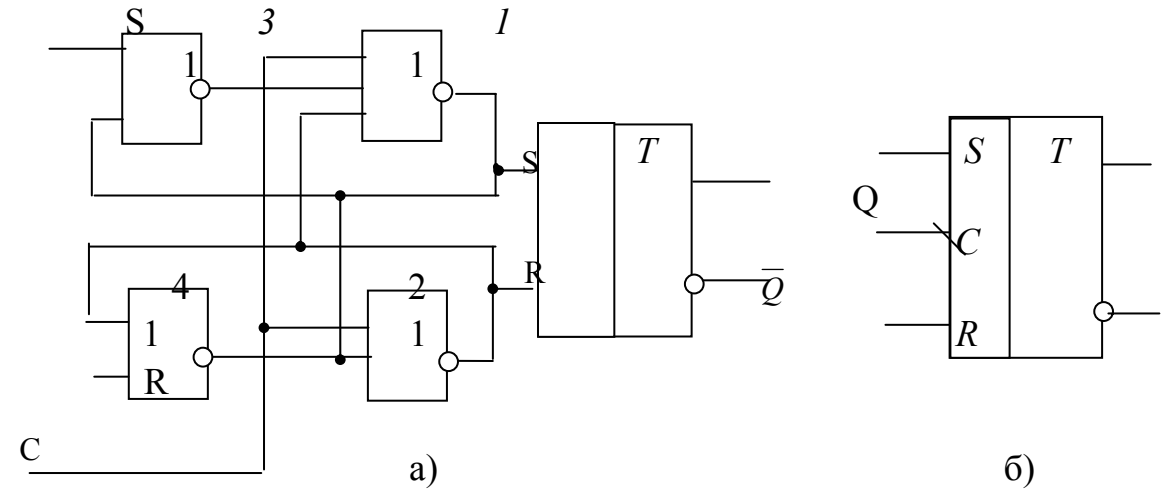


Рис. 4.15. RS-триггер (а) и его условное обозначение (б)

с динамическим входом на элементе ИЛИ – НЕ

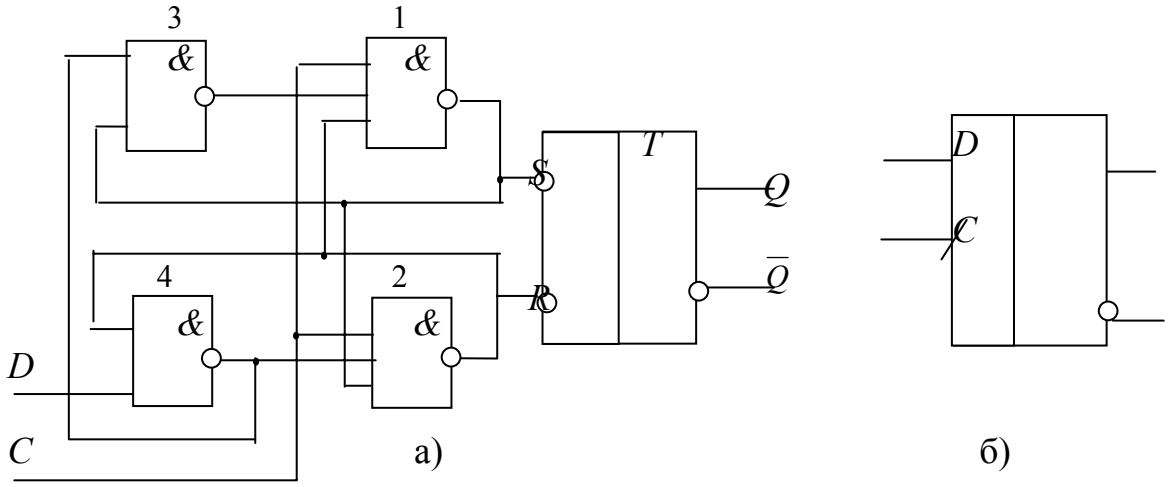


Рис. 4.16. Схема D-триггера (а) и его условное обозначение (б)
с динамическим управлением

Триггеры входят в состав серий ИМС. По функциональному назначению триггеры имеют следующие обозначения: ТР –*RS*-триггеры, ТВ – *JK*-триггеры, ТМ – *D*-триггеры.

Основные параметры интегральных микросхем триггеров можно разделить на две группы: статические и динамические. К статическим параметрам относятся: входное напряжение высокого $U_{\text{вх}}^{1}$ и низкого $U_{\text{вх}}^{0}$ уровней, ток потребления от источника питания, напряжение $U_{\text{пит}}$, нагрузочная способность (ток $I_{\text{н}}$, отдаваемый в нагрузку, или количество $K_{\text{раз}}$ микросхем той же серии, подключаемых к выходу триггера), потребляемая мощность $P_{\text{пот}}$ от источника питания. К динамическим параметрам триггеров относятся: время $t^{0,1}$ переключения из низкого уровня в высокий, время $t^{1,0}$ переключения из высокого уровня в низкий, максимальная частота $f_{\max}$ переключения. Большинство перечисленных параметров определяется серией микросхемы и типом применяемой логики. В табл. 4.8 приведены наименования и параметры некоторых ИМС триггеров.

Таблица 4.8. Параметры интегральных микросхем триггеров

Наименование триггера	Тип логики	Функциональное назначение	$U_{\text{пит}}$, В	$P_{\text{пит}}$, мВт	$U_{\text{вх}}$, В	$t^{0,1}$, нс	$f_{\max}$, МГц
555ТР2	ТТЛ	Четыре *R*-триггера	5,0	35,0	2,5	22,0	25
155ТВ1	ТТЛ	*JK*-триггер	5,0	100,0	2,5	40,0	15
555ТМ2	ТТЛ	Два *D*-триггера	5,0	30,0	2,5	20,0	30
561ТР2	КМОП	Четыре *SR*-триггера	15,0	< 1	7,2	200,0	5
561ТВ1	КМОП	Два *JK*-триггера	15,0	< 1	7,2	150,0	5
561ТМ3	КМОП	Четыре *D*-триггера	15,0	< 1	7,2	150,0	5
1533ТМ2	ТТЛШ	Два *D*-триггера	5,0	20,0	2,5	22,0	34

4.10. Логический синтез триггерных устройств

В практике проектирования цифровых устройств нередко возникает необходимость в разработке новых типов триггеров. Если разработчик встречается с таким типом триггера впервые и, следовательно, у него нет никакого схемного аналога, то эвристические методы синтеза зачастую приводят к неоправ-

данно большим затратам времени. И здесь становится важной задача разработки формальных методов синтеза, которые позволяют достаточно быстро спроектировать требуемый вариант триггера.

Как отмечалось выше, в наиболее общем виде структура триггера представляет собой устройство, содержащее элемент памяти с двумя устойчивыми состояниями (собственно триггер) и схему управления. Описываемый здесь метод учитывает именно такую структуру триггерного устройства и базируется на нахождении логических функций, непосредственно воздействующих на входы собственно триггера и вызывающих его переключение в требуемое состояние. Рассмотрим эту методику на конкретных примерах. В качестве собственно триггера взят элементарный триггер *RS*–типа. Однако последнее не является обязательным, поскольку в качестве собственно триггера может использоваться любой из элементарных триггеров, рассмотренных выше.

Пример 1.

Допустим, что необходимо спроектировать триггерное устройство, имеющее два информационных входа A_1 и A_2, функционирование которого описывается табл.4.9.

Таблица 4.9. Функционирование проектируемого триггерного устройства

Номер строки	$A_1{}^t$	$A_2{}^t$	Q^{t+1}
1	0	0	1
2	0	1	0
3	1	0	1
4	1	1	Q^t

Процедура синтеза начинается с составления обобщенной таблицы состояний триггера, в которой кроме столбцов Q^t, $A^t{}_1$, $A^t{}_2$, Q^{t+1} имеются столбцы R' и S' (табл. 4.10). В столбцы $A^t{}_1$, $A^t{}_2$ заносятся всевозможные комбинации входных сигналов при двух состояниях триггера $Q^t = 0$ (строки 1 - 4) и $Q^t = 1$ (строки 5—8). В столбец Q^{t+1} записываются те состояния триггера, в которые он должен устанавливаться согласно закону его функционирования, приведенного в табл. 4.9. Например, в соответствии с табл. 4.9, при комбинации сигналов $A_1 = A_2 = 0$ (строка 1) триггер должен устанавливаться в состояние $Q^{t+1} = 1$. Следовательно, в строках 1 и 5 столбца Q^{t+1} (табл. 4.10) записывается 1. При

комбинации сигналов $A^t_1 = A^t_2 = 1$ триггер должен переключаться в состояние $\overline{Q}^t$, т.е. в состояние, инверсное исходному (см. табл. 4.10). Следовательно, в строке 4 столбца Q^{t+1} необходимо записать 1, а в строке 8 - 0 и т. д. При заполнении строк столбцов R' и S' табл. 4.10 будем полагать, что собственно триггер выполняется на элементах ИЛИ-НЕ, т.е. его переключение по входам R и S осуществляется сигналами с уровнем 1, использовать матрицу перехода асинхронного RS-триггера на элементах ИЛИ-НЕ (табл. 4.11). Таким образом, чтобы при комбинации сигналов $A^t_1 = A^t_2 = 0$ и исходном состоянии $Q^t = 0$ триггер установился в состояние 1 ($Q^{t+1} = 1$), необходимо, чтобы на его входах R' и S' присутствовала комбинация сигналов $R' = 0$ и $S' = 1$.

Таблица 4.10. Обобщенная таблица состояний проектируемого триггерного устройства

Номер строки	Q^t	$A_1{}^t$	$A_2{}^t$	R'	S'	Q^{t+1}
1	0	0	0	0	1	1
2	0	0	1	*	0	0
3	0	1	0	0	1	1
4	0	1	1	0	1	1
5	1	0	0	0	*	1
6	1	0	1	1	0	0
7	1	1	0	0	*	1
8	1	1	1	1	0	0

Таблица 4.11. Матрица перехода RS-триггера на элементах ИЛИ-НЕ

Q^t	Q^{t+1}	**R**	**S**
0	0	*	0
0	1	0	1
1	0	1	0
1	1	0	*

Именно эта комбинация и заносится в первую строку столбцов R' и S' табл. 4.10. Если триггер находится в состоянии $Q^t = 1$ и поступает комбинация

сигналов $A_1 = A_2 = 0$, то триггер должен подтвердить свое исходное состояние. Поэтому в строке 5 столбца R' должен быть записан 0, а в столбце S' этой строки может стоять либо 0 либо 1, В дальнейшем будем такие ситуации обозначать знаком * (произвольное состояние). В строке 2 этих же столбцов записана комбинация сигналов $R' = *$, $S' = 0$, а в строке 6 $R' = 1$, $S' = 0$. Это следует из того, что при такой комбинации сигналов триггер согласно столбцу Q^{t+1} табл. 4.9 должен принять нулевое состояние.

Последующие строки табл. 4.10 выполняются аналогично.

По данным табл. 4.10 составляются переключательные функции для входов R' и S'. Условимся переключательные функции представлять в виде СДНФ. Тогда для входов R' и S' можно записать

$$R' = \overline{Q} \cdot A_1 \cdot \overline{A}_2^* \vee Q \cdot \overline{A}_1 \cdot A_2 \vee Q \cdot A_1 \cdot A_2 \ ;$$
$$S' = \overline{Q} \cdot A_1 \cdot A_2 \vee \overline{Q} \cdot A_1 \cdot \overline{A}_2 \vee \overline{Q} \cdot \overline{A}_1 \cdot \overline{A}_2 \vee Q \cdot \overline{A}_1 \cdot \overline{A}_2^* \vee Q \cdot A_1 \cdot A_2 \ .$$

Здесь индексы t при переменных R, S, Q, A будем опускать. При составлении выражений для входов R' и S' знак * заменен на 1, а соответствующие минтермы, которые в дальнейшем будем называть избыточными, отмечены знаком *.

Затем осуществляется минимизация полученных выражений для входов R' и S'. Для этой цели воспользуемся картами Карно (рис. 4.17). При занесении функций на карты в клетки, соответствующие избыточным минтермам, будем ставить знак *, а при считывании функций этот знак заменять на 1 или 0, руководствуясь требованием получения минимизированных выражений.

С учетом изложенного при считывании выражения для функции R' необходимо избыточный минтерм заменить на 0, а при считывании выражения для функции S' все избыточные минтермы заменить на 1. В этом случае функции для R' и S' принимают минимальные формы:

$$R' = A_2 \cdot Q \ ;$$
$$S' = \overline{Q} \cdot A_1 \vee \overline{A_2} \ .$$

Для тактируемых триггеров в полученные выражения для функций R' и S' необходимо ввести конъюнктивный член C. В результате для тактируемых триггеров эти выражения запишутся в виде

$$R' = A_2 \cdot Q \cdot C \ ;$$
$$S' = \overline{Q} \cdot A_1 \cdot C \vee \overline{A}_2 \cdot C \ .$$

По полученным выражениям для входов R' и S' строим проектируемое устройство (рис. 4.18) и проверяем его работоспособность. При комбинации сигналов $A_1 = A_2 = 0$ триггер согласно табл. 4.9 обязан установиться в состояние $Q^{t+1} = 1$. При такой комбинации только вентиль B_2 будет подготовлен к включению. В результате при поступлении ТИ ($C = 1$) на выходе вентиля B_2 сформируется сигнал с уровнем 1 и триггер по входу S' установится в состояние $Q^{t+1} = 1$. При комбинации сигналов $A_1 = 0$, $A_2 = 1$ триггер должен установиться в состояние $Q^{t+1} = 0$. Если триггер находился в состоянии $Q = 1$, то только вентиль B_1 будет подготовлен к включению. В результате при поступлении тактирующего сигнала триггер по входу R установится в состояние $Q^{t+1} = 0$.

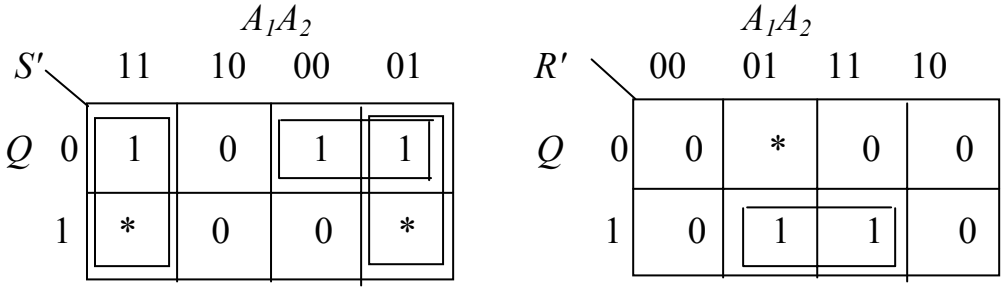


Рис. 4.17. Карты Карно синтезируемого триггера

При комбинации сигналов $A_1 = 1$, $A_2 = 0$ и при $C = 1$ триггер через вентили B_2 и B_3 установится в состояние $Q^{t+1} = 1$; при этом вентиль B_1 будет закрыт сигналом $A_2 = 0$. При комбинации сигналов $A_1 = A_2 = 1$ триггер должен переключиться в инверсное состояние.

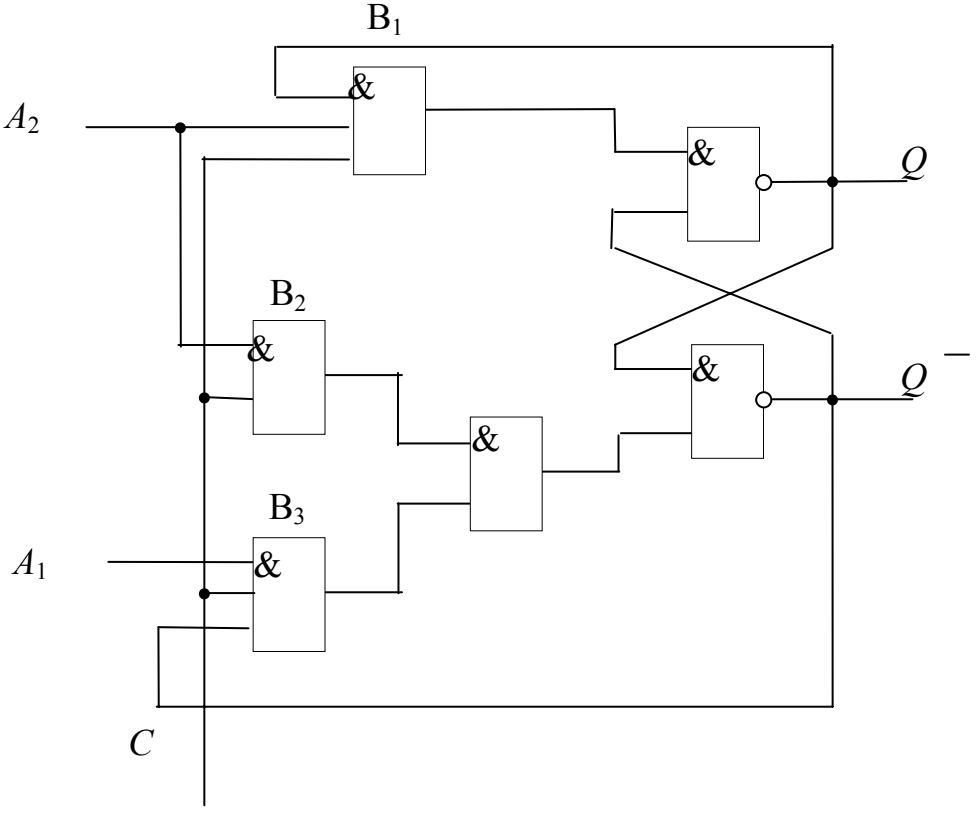


Рис.4.18. Схема синтезированного триггера

Пусть триггер находится в состоянии $Q = 0$. В этом случае только вентиль B_3 будет подготовлен к переключению. Следовательно, с приходом тактирующего сигнала триггер по входу S' установится в состояние $Q = 1$ и тем самым к включению будет подготовлен вентиль B_1. В результате при достаточно большой длительности тактирующего сигнала триггер может за один такт переключаться как минимум 2 раза, что недопустимо.

Для исключения последнего необходимо, чтобы на тактовом входе действовал не потенциальный, а импульсный сигнал, т.е. сигнал, который бы оканчивался непосредственно после переключения триггера в противоположное состояние. Такие триггеры приведены здесь как схемотехнический пример и не находят применения на практике.

Триггер, тактируемый потенциальными сигналами, получается добавлением к триггеру, полученному в результате синтеза, второй ступени, как это показано на рис. 4.19. Другими словами этот триггер построен по способу *MS* структуры.

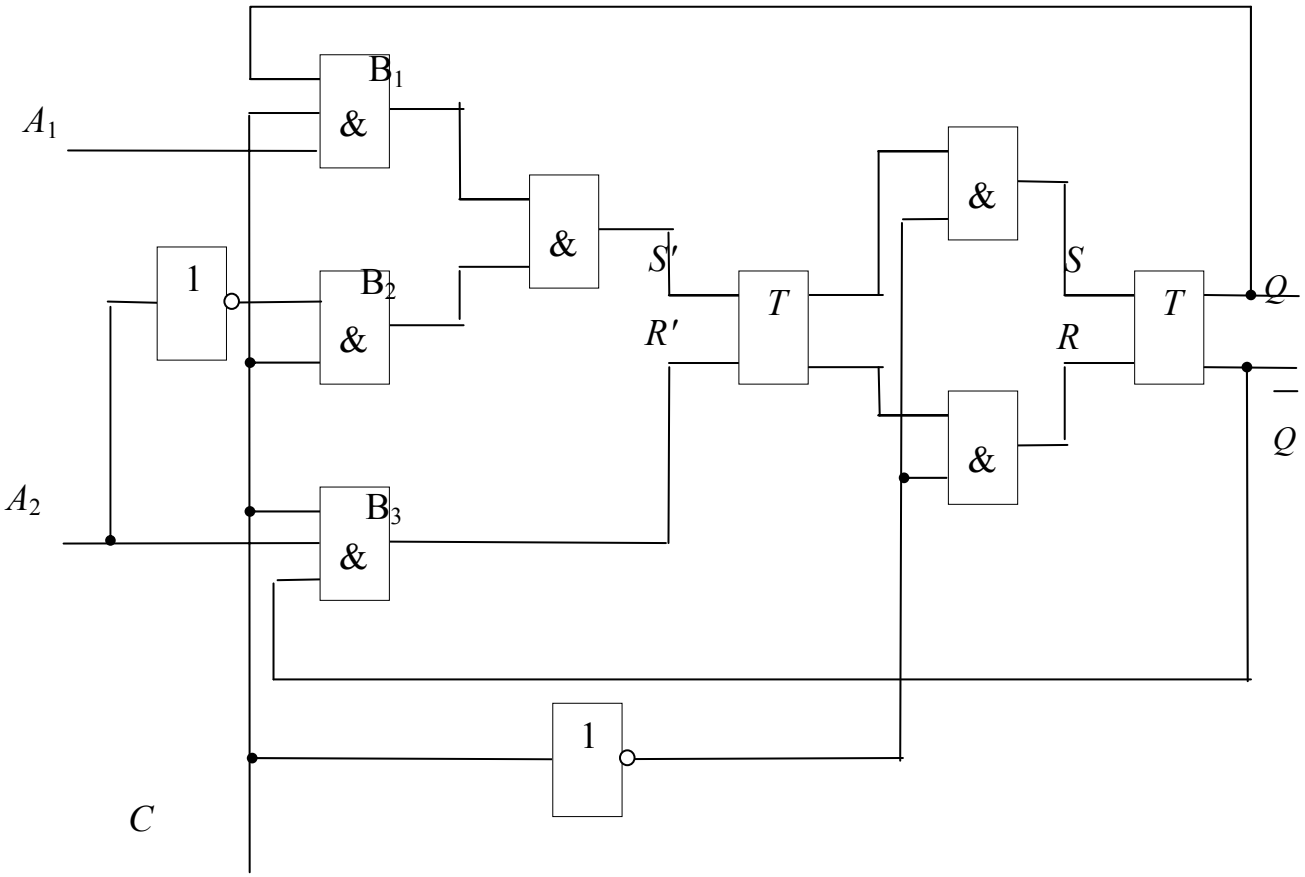


Рис.4.19. Схема синтезированного двухступенчатого триггера

Пример 2.

Для проверки рассмотренной методики синтезируем, например, синхронный *RS*- триггер. Как и в первом случае, составляем обобщенную таблицу состояний *RS*-триггера (табл. 4.12). В строках 4 и 8 столбцов R' и S' записаны 1. Это объясняется тем, что при комбинации входных сигналов $R = S = 1$ триггер согласно закону его функционирования принимает неопределенное состояние, что отмечено знаком * в столбце Q^{t+1} этих же строк.

Таблица 4.12. Обобщенная таблица состояний асинхронного* RS-*триггера

Номер строки	Q^t	R	S	R'	S'	Q^{t+1}
1	0	0	0	*	0	0
2	0	0	1	0	1	1
3	0	1	0	*	0	0
4	0	1	1	1	1	*
5	1	0	0	0	*	1
6	1	0	1	0	*	1
7	1	1	0	1	0	0
8	1	1	1	1	1	*

По результатам таблицы находим выражения для функций R' и S':

$$R' = \overline{Q} \cdot \overline{R} \cdot \overline{S}^{*} \vee \overline{Q} \cdot R \cdot \overline{S}^{*} \vee \overline{Q} \cdot R \cdot S \vee Q \cdot R \cdot \overline{S} \vee Q \cdot R \cdot S \ ;$$

$$S' = \overline{Q} \cdot \overline{R} \cdot S \vee \overline{Q} \cdot R \cdot S \vee Q \cdot \overline{R} \cdot \overline{S}^{*} \vee Q \cdot \overline{R} \cdot S^{*} \vee Q \cdot R \cdot S \ .$$

По картам Карно (рис. 4.20) находим, что функции R' и S' примут минимальные формы $R' = R$ и $S' = S$, если не заполнить клетки всего одного избыточного минтерма, а именно минтерма $\overline{Q} \cdot \overline{R} \cdot \overline{S}$ для функции R' и минтерма $Q \cdot \overline{R} \cdot \overline{S}$ для функции S'.

S' RS R' RS

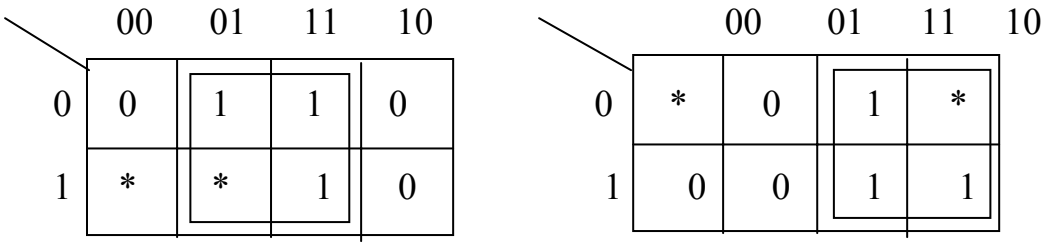


Рис. 4.20. Карты Карно синтезируемого *RS*-триггера

Следовательно, для тактируемого триггера можно записать

$$S' = S \cdot C\,; \quad R' = R \cdot C\,.$$

Схема *RS*-триггера, построенного по полученным выражениям, приведена на рис. 4.21.

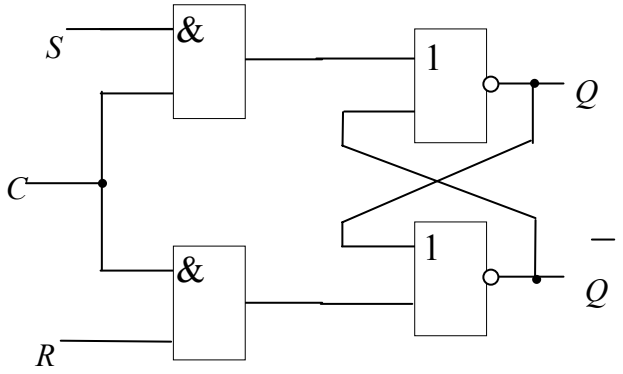


Рис. 4.21. Схема синтезированного *RS*-триггера

Рассмотренные выше методы синтеза можно использовать при проектировании любых переключательных устройств, методы синтеза триггеров с двумя информационными входами можно полностью применить для синтеза триггеров практически с любым числом информационных входов.

4.11. Регистры

Регистры предназначаются для хранения (и, возможно, преобразования) цифрового кода (двоичного числа, слова), поэтому основу их составляют запоминающие элементы – триггеры. В каждом из них должна храниться цифра разряда числа. Кроме хранения регистры могут осуществлять сдвиг принято-

го слова, преобразование двоичного кода из прямого в обратный и наоборот, логические сложение и умножение.

Для приема (записи) информации (т. е. разрядов двоичного кода) в регистр, передачи (считывания) ее, сдвига кода в регистре влево или вправо на определенное число разрядов и других операций применяются вспомогательные логические устройства и соответствующие управляющие сигналы. Операции приема и передачи информации можно реализовать параллельно (все разряды принимаются или передаются одновременно) или последовательно (разряды кода принимаются или передаются последовательно во времени); применяются также регистры, в которых прием (передача) кода осуществляется последовательно, а передача (прием) - параллельно.

Соответственно говорят о последовательных, параллельных, последовательно-параллельных, параллельно-последовательных регистрах.

Параллельные регистры или регистры памяти используются главным образом для запоминания многоразрядного двоичного кода. Последовательные регистры - так называемые сдвиговые регистры - используются как для кратковременного хранения двоичного кода, так и для других целей (для создания генераторов кодов, для синтеза кольцевых счетчиков и управляющих устройств и т.п.).

Любой регистр состоит из отдельных одинаковых блоков, каждый из которых содержит триггер того или иного типа для хранения одного разряда информации (1 или 0) и комбинационное устройство, служащее для управления процессом записи и считывания информации или других ее преобразований (например, сдвиг кода влево или вправо). Поэтому синтез регистров сводится к синтезу отдельных блоков и, возможно, цепей связи между ними.

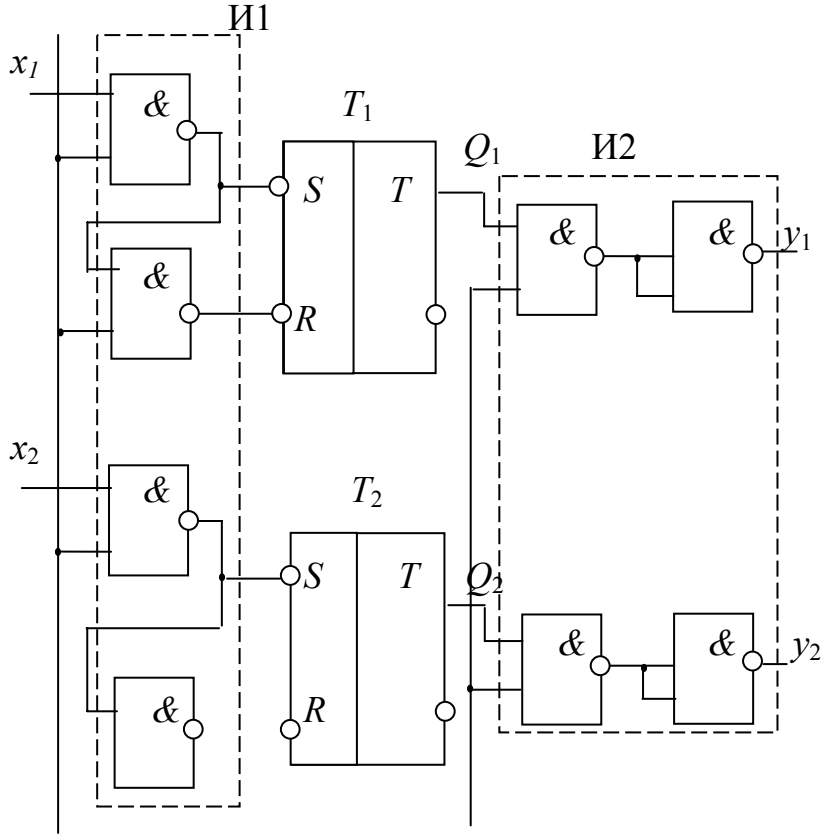


Рис. 4.22. Функциональная схема параллельного регистра

Параллельные регистры. Функциональная схема параллельного регистра (двухразрядного на элементах И-НЕ) изображена на рис. 4.22. Число триггеров $T_1, T_2, ..., T_n$ определяется числом разрядов числа $X = x_1, x_2, ..., x_n$; (где $x_i = 0$ или $x_i = 1$, $i = 1, 2, ..., n$).

При одновременной подаче на входы логических элементов $И_1$ сигналов кода X, и сигнала приема C_1 появляются сигналы на выходах $И_1$, которые устанавливают соответствующие триггеры в положения, определяемые значениями разрядов входного кода. Для того чтобы считать код, зафиксированный в регистре, т. е. передать его из регистра в другие устройства, следует подать сигнал передачи C_2 на входы ЛЭ $И_2$. При этом сигналы 1 появляются на выходах тех схем $И_2$, на входы которых поданы разрешающие сигналы (сигналы 1) с соответствующих триггеров.

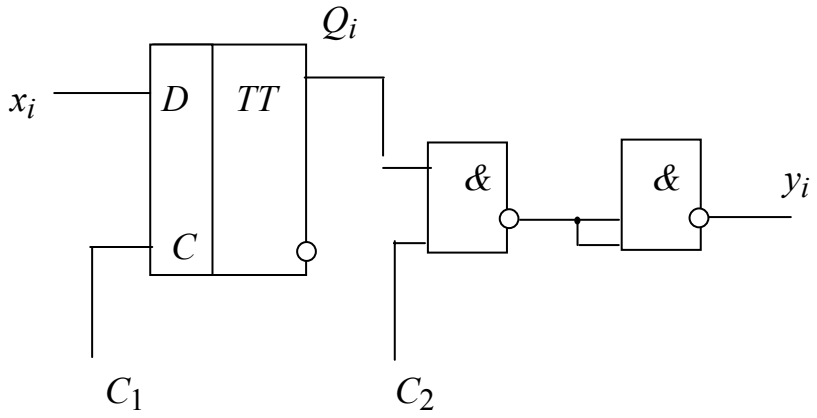


Рис. 4.23. Схема одного разряда регистра на *D*-триггере

Заметим, что вместо прямого кода (снимаемого с прямых выходов триггеров) можно считывать обратный код (с инверсных выходов триггеров). С этой целью используются соответствующие управляющие сигналы. В зависимости от типа используемых триггеров и схем управления можно в регистрах производить и ряд поразрядных логических операций с уже записанным в регистре и вновь вводимым числом.

Таблица 4.13. Функционирование параллельного регистра

C^t_1	C^t_2	Q^t_i	y^t_i

0	0	Q_i^{t-1}	0
0	1	Q_i^{t-1}	Q_i^t
1	0	x_i^t	0
1	1	-	-

Рассмотрим некоторые реализации параллельных регистров.

Схема регистра рис. 4.22 функционирует по правилам, данным в табл. 4.13. Комбинация управляющих сигналов $C^t_1 = 1$, $C^t_2 = 1$, т.е. $C^t_1 C^t_2 = 1$, недопустима, нельзя одновременно записывать и считывать информацию. Схема одного разряда (i-го) параллельного регистра на D-триггерах привелена на рис. 4.23. Этот регистр не требует отдельных управляющих схем ввода разрядов и так же, как и предыдущий, не требует предварительной установки триггеров в нулевое состояние.

Заметим в заключение, что можно реализовать блоки регистра и на других триггерах, в частности на T-триггерах (последние легко получаются из D-триггеров).

Последовательные регистры. Функциональная схема последовательного регистра или регистра сдвига на RS-триггерах приведена на рис. 4.24. На входы R-триггеров подаются тактовые импульсы (ТИ), под воздействием которых все триггеры переключаются в состояние 0. Если триггер T_i переключается при этом из состояния 1 в состояние 0, то на его выходе образуется сигнал переноса (перепад напряжения), вызывающий в свою очередь переключение триггера T_{i+1} из состояния 0 в состояние 1. Время задержки сигнала переноса в элементах задержки достаточно, чтобы завершились переходные процессы в триггере T_{i+1}, вызванные действием ТИ.

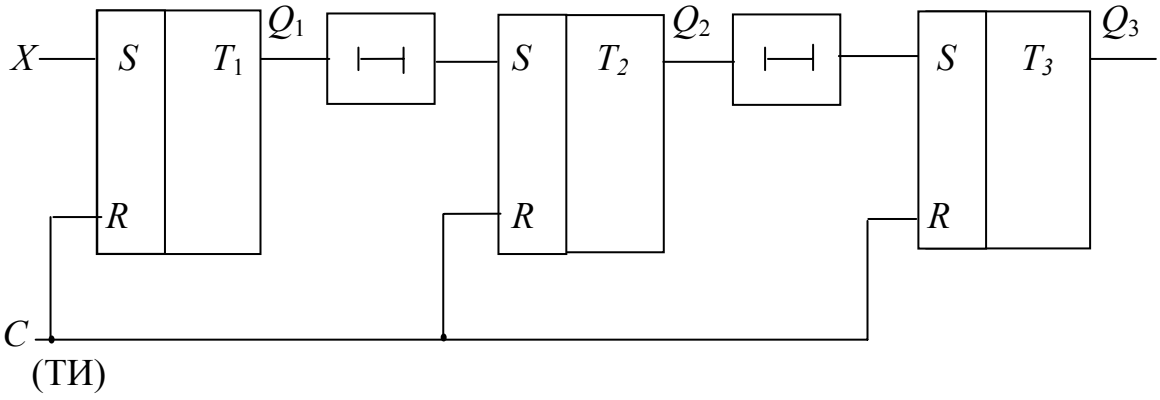


Рис. 4.24. Функциональная схема последовательного регистра

Таким образом, при подаче ТИ на входы R (на такте C^t) информация зафиксированная в регистре на такте C^{t+1}, сдвинется на одну позицию вправо.

Пусть, например, регистр находиться с состоянии 000 ($Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$) и перед первым ТИ подан сигнал $x = 1$; тогда состояние регистра станет 100; при подаче ТИ ($C_1 = 1$) информация из T_1 переписывается в триггер T_2, и состояние регистра будет 010 и т.д.

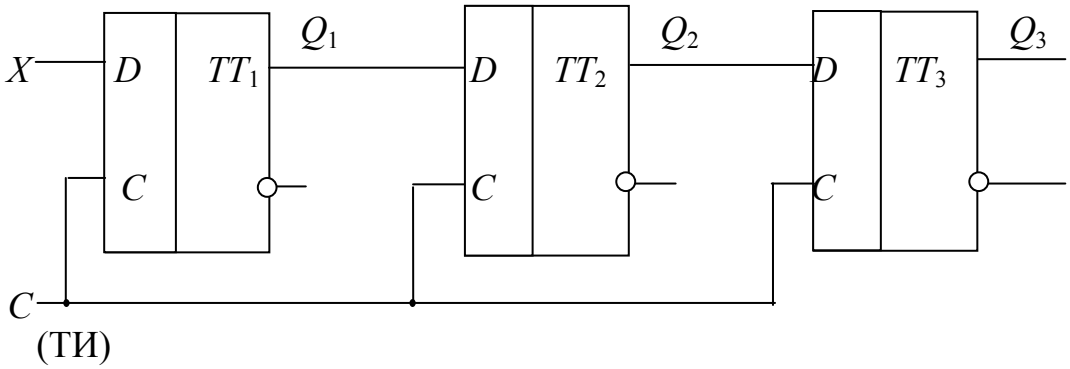


Рис. 4.25. Функциональная схема последовательного регистра на D-триггерах, его входные сигналы

Если же перед подачей второго ТИ подан сигнал $x = 1$, то единица записывается в триггер T_1, и в регистре будет зафиксирован код 110. При действии $C_2 = 1$ в регистре окажется код 011 и т.д. Входной n-разрядный код записывается в регистр, состоящий из n триггеров в течение n-тактов; для полного вывода этого кода из регистра вновь требуется n тактов.

Весьма просто реализуется последовательный регистр на D-триггерах (рис. 4.25); на каждом такте при $C^t = 1$ имеет $Q^t_1 = x^{t-1}_{i-1}$, $Q^t_2 = x^{t-1}_{i}$, $Q^t_3 = x^{t-1}_{i+1}$. Если используются DRS-триггеры, т.е. D-триггеры с установочными входами, то R-входы триггеров используются для подачи сигнала установки 0.

Реверсивные последовательные регистры**.** В рассмотренных выше последовательных регистрах осуществляется сдвиг информации вправо. Однако во многих случаях требуется осуществлять сдвиг кода влево; с этой целью следует так соединить триггеры, чтобы при подаче сигнала сдвига ТИ $(i-1)$-й, триггер T_{i-1} переключался в состояние, в котором до этого был i-й триггер T_i; в регистре на D-триггерах следует реализовать $D^t_{i-1} = Q^t_{i,}$. В реверсивном регистре имеется возможность сдвига информации в обоих направлениях.

Последовательно-параллельные и параллельно-последовательные регистры строятся на базе сочетания последовательных и параллельных регистров.

Регистры сдвига с обратными связями. Последовательные регистры с соответствующими обратными связями могут выполнять роль кольцевых счетчиков, генераторов кодовых комбинаций и др. В качестве примера рассмотрим схему регистра с обратной связью (рис. 4.26) на *JK*- триггерах; пусть в исходном состоянии в регистр записана некоторая информация: например, 1011 (Q_1 = 1, Q_2 = 0, Q_3 = Q_4 = 1). Эта информация сдвигается вправо на один разряд при поступлении каждого ТИ на вход *C;* после первого ТИ в регистре будет информация 1101, после второго ТИ – 1110, после третьего ТИ – 0111, появляется периодическая последовательность сигналов с длиной периода L = 4, после четвертого – вновь 1011 и т. д. Таким образом, в регистре циклически меняется информация, причем длина одного цикла L = 4, на выходе регистра Q_4.

Можно построить регистры, в которых сигнал обратной связи, подаваемый на вход первого триггера, будет являться логической функцией от выходных сигналов нескольких последующих триггеров. При этом можно получить циклы различной длины; максимальная длина цикла в n-каскадном регистре $L_{\max} = 2^n$.

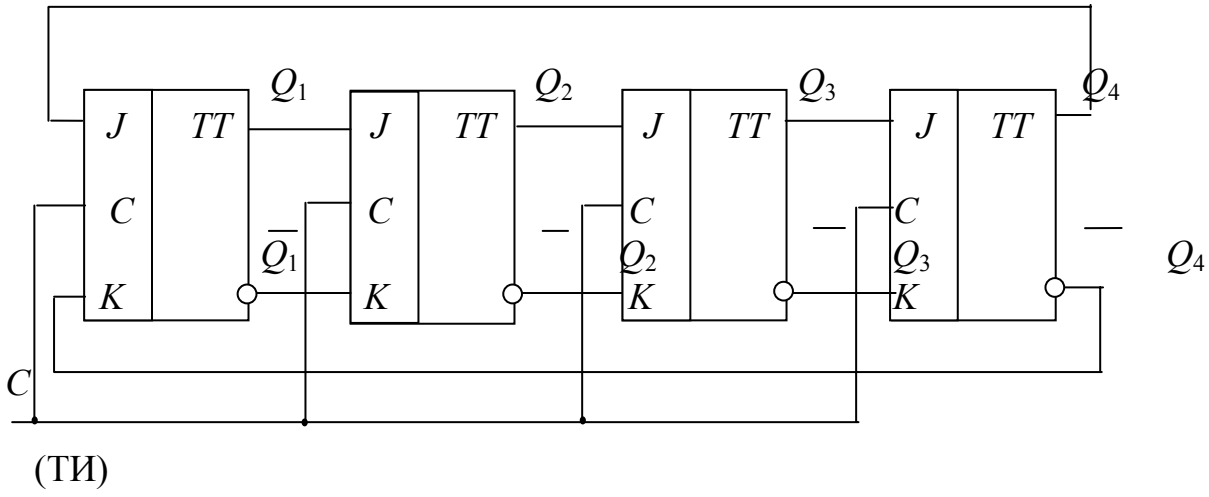


Рис. 4.26. Функциональная схема регистра с обратной связью на JK-триггерах

В простейшей схеме, в которой сигнал обратной связи определяется непосредственно значением выходного сигнала последнего п-го триггера, максимальная длина цикла L = п. Такая простейшая схема, в которой циклически меняется код, выполняет роль так называемого кольцевого счетчика. Порядок переходов, например, для трехразрядного кольцевого счетчика на D-триггерах (рис. 4.27) приведен в табл. 4.14.

Таблица 4.14. Функционирование последовательного регистра

t	C^t_1	C^t_2	C^t_3
1	1	0	0
2	0	1	0
3	0	0	1

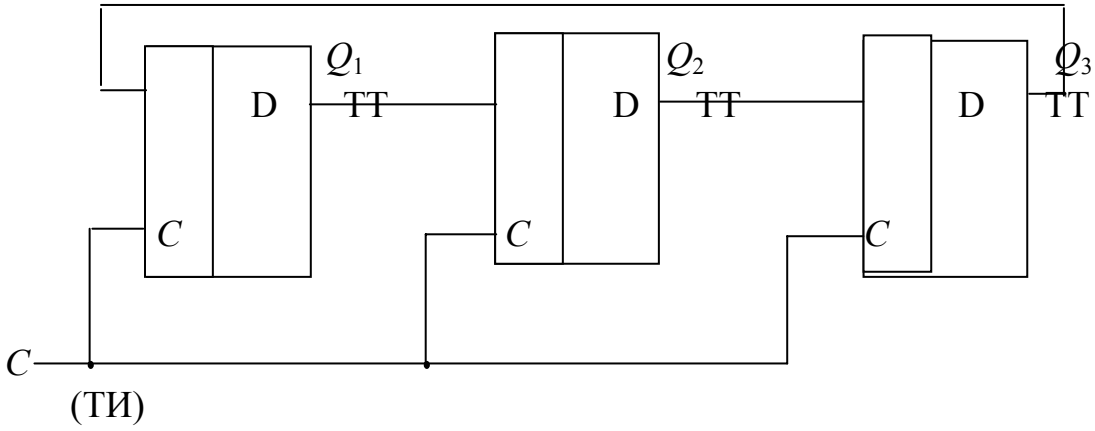


Рис. 4.27. Функциональная схема регистров с обратной связью на D-триггерах

На каждом такте кодовая единица, введенная в первый триггер, сдвигается на одну позицию, и при этом только один триггер переключается из состояния 0 в состояние 1 и находится в этом состоянии в течение одного такта. Можно построить кольцевые счетчики на тактируемых *JK*-триггерах, а также на асинхронных *RS*- и *T*-триггерах.

Регистры входят в состав серии ИМС. Условное обозначение ИМС регистров состоит из обозначения серии (трех или четырех цифр), функционального назначения (двух букв ИР) и порядкового номера разработки (от одной до трех цифр).

Основные параметры интегральных микросхем регистров можно разделить на две группы: статические и динамические. К статическим параметрам

относятся: входное напряжение высокого $U_{вх}^{1}$ и низкого $U_{вх}^{0}$ уровней, ток потребления от источника питания, напряжение $U_{пит}$, коэффициент разветвления $K_{раз}$. К динамическим параметрам триггеров относятся: время $t^{0,1}$ переключения из низкого уровня в высокий, время $t^{1,0}$ переключения из высокого уровня в низкий, максимальная частота f_{max}. Большинство перечисленных параметров определяется серией микросхемы и типом применяемой логики. В табл. 4.15 приведены наименования и параметры ИМС регистров.

Таблица 4.15. Параметры интегральных микросхем регистров

Наименование регистра	Тип логики	Функциональное назначение	Время задержки $t_{зад}$,нс	Потребляемая мощность $P_{пот}$,мВт
К155ИР1	ТТЛ	Четырехразрядный сдвиговый	35	410
К531ИР11	ТТЛШ	Четырехразрядный последовательно-параллельный	22	675
К561ИР9	КМОП	Универсальный четырехразрядный сдвиговый	800	< 1
К1533ИР9	ТТЛШ	Восьмиразрядный сдвиговый с параллельной загрузкой	30	140
КР1533ИР10	ТТЛШ	Восьмиразрядный сдвиговый с последовательно-параллельной записью	21	145

К555ИР10	ТТЛШ	Восьмиразрядный сдвиговый с последовательно-параллельной записью	35	190

4.12. Вопросы и упражнения

1. Чем различаются триггеры на элементах И-НЕ и ИЛИ-НЕ?

2. Чем различаются асинхронные и синхронные триггеры?

3. Поясните принцип действия синхронных триггеров со статическим и динамическим управлением.

4. Почему *JK*-триггер называется универсальным?

5. В каком состоянии будет находиться асинхронный *RS*-триггер при $S = 1$, $R = 0$?

6. Для чего используются *S*- и *R*-входы в микросхеме К155ТМ2 и какую функцию они выполняют?

7. Как реализовать на микросхеме К155ТВ1 *T*-триггер?

8. Объясните работу синхронных триггеров, выполненных по *MS*-структуре.

9. Как построить *T*-триггер на основе *RS*-, *D*-, и *JK*-триггеров?

10. Можно ли в цифровом устройстве заменить синхронный *RS*-триггер на синхронный *JK*-триггер (*E*-, *R*-, *S*-триггер), не нарушая правильности работы устройства?

11. Объясните, как строится временная диаграмма работы триггера.

12. Перечислите основные операции, выполняемые регистром.

13. Определите регистры хранения.

14. Определите регистры сдвига.

15. Что такое длина регистра и сколько различных многоразрядных двоичных чисел можно записать в регистр длиной 12?

16. Какую роль в структуре регистра выполняют триггеры и комбинационные схемы?

17. В чем особенности реверсивных регистров?

5. СЧЕТЧИКИ

5.1. Общие сведения

Последовательностное цифровое устройство, обеспечивающее хранение слова информации и выполнение над ним микрооперации счета, называется счетчиком. Операция счета заключается в изменении значения числа C в счетчике на ±1. Счетчик, в котором выполняется операция счета $C = C+1$, называется суммирующим, а счетчик, реализующий операцию $C = C–1$, – читающим. Счетчик называется реверсивным, если реализуются обе операции.

Основным параметром счетчика является модуль счета K_C, определяемый максимальным числом единичных сигналов, которое может быть сосчитано счетчиком. Счетчик, содержащий n двоичных разрядов, может находиться в состояниях 0, 1,…, 2^n-1. При поступлении на вход суммирующего счетчика 2^n - й единицы он переходит из состояния 2^n-1 в состояние 0. Таким образом, n - разрядный суммирующий двоичный счетчик имеет модуль счета $K_C = 2^n$.

Счетчики характеризуются также быстродействием, которое определяется допустимой частотой входных сигналов и временем установки состояния счетчика.

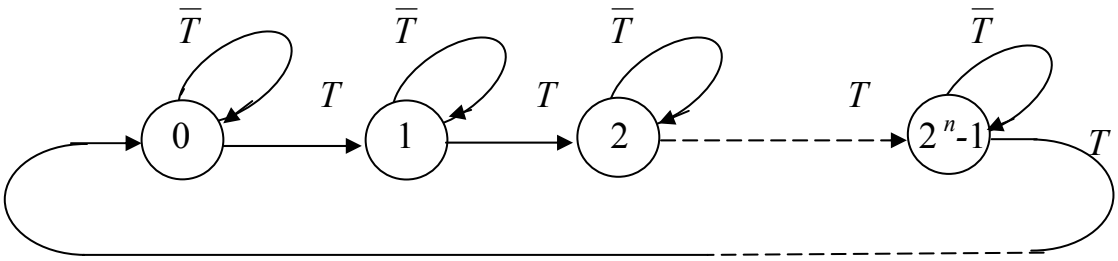


Рис. 5.1. Граф двоичного n-разрядного счетчика

Функционирование n-разрядного двоичного счетчика можно предста-вить графом состояний (рис. 5.1). Вершины графа отмечены значением, характеризующим состояние счетчика и значение выходных сигналов. Микрооперация выполняется по сигналу счета T. При этом счетчик переходит из состояния C в состояние C+1. При отсутствии сигнала ($\overline{T}$ =1) счетчик сохраняет текущее состояние. Счетчики обычно реализуются на T - триггерах. Однако для их построения могут применяться не только триггеры со счетным входом, но и D-триггеры, JK-триггеры.

Счетчики можно классифицировать по нескольким признакам. В зависимости от направления счета различаются суммирующие (с прямым счетом),

вычитающие (с обратным счетом) и реверсивные (с прямым и обратным счетом). По способу организации схемы переноса различаются счетчики с последовательным, параллельным и параллельно-последова-тельным переносом. В зависимости от наличия синхронизации различаются синхронные и асинхронные счетчики.

При маркировке для обозначения счетчика используются буквы ИЕ. Конструктивно счетчики выполняются в виде совокупности интегральных схем-триггеров, соединенных соответствующим образом, или в виде одной интегральной схемы, содержащей многоразрядный счетчик.

5.2. Суммирующие двоичные счетчики

В суммирующем двоичном *n*-разрядном счетчике, состоящем из *n* триггеров, реализуется счетная последовательность чисел. Эта последовательность начинается с 0. Очередное число в этой последовательности получается прибавлением единицы к предыдущему числу. После того как последовательность доходит до максимального числа 2^n-1, она снова проходит через 0 и повторяется. В счетчике с *n* триггерами число возможных состояний равно 2^n, модуль счета K_C также равен 2^n. Каждому состоянию счетчика соответствует число в счетной последовательности от 0 до 2^{n-1}. Рассмотрим устройство двоичного 3-разрядного суммирующего счетчика. В таком счетчике можно реализовать счетную последовательность от 0 до 2^3-1=7. Последовательность чисел может быть задана совокупностью 3-разрядных двоичных чисел $a_3\ a_2\ a_1$: 000, 001, 010, 011, 100, 101, 110, 111. Поставим в соответствие каждому разряду a_i числа выход триггера Q_i. В 3-разрядном счетчике с выходами Q_3, Q_2, Q_1 будет реализовываться счетная последовательность от 0 до 7.

Счетчик может быть реализован с использованием двухступенчатых триггеров *T* со счетным входом. Схема двоичного 3-разрядного суммирующего счетчика представлена на рис. 5.2. В этой схеме исходное состояние счетчика устанавливается подачей сигнала по шине «Уст. 0». Триггеры *T* изменяют свое состояние с окончанием входного сигнала, т. е. после перехода от уровня 1 к уровню 0. Входной сигнал по шине C_0 подается на счетный вход триггера 1. Работа счетчика может быть описана с помощью временной диаграммы (рис. 5.3).

До начала прихода первого сигнала счетчик находился в нулевом состоянии. Это соответствует наличию уровня 0 на выходах Q_3, Q_2, Q_1. С поступлением по шине C_0 входных сигналов на счетный вход первого триггера начинается работа счетчика. С приходом первого сигнала триггер 1 переходит в состояние 1 и на его выходе устанавливается уровень Q=1.

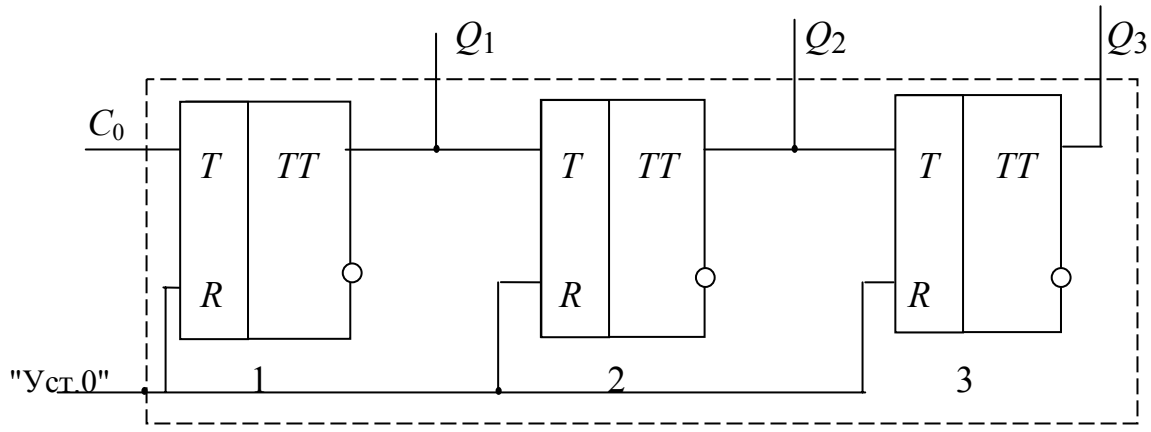


Рис. 5.2. Схема двоичного суммирующего счетчика с K_C=8

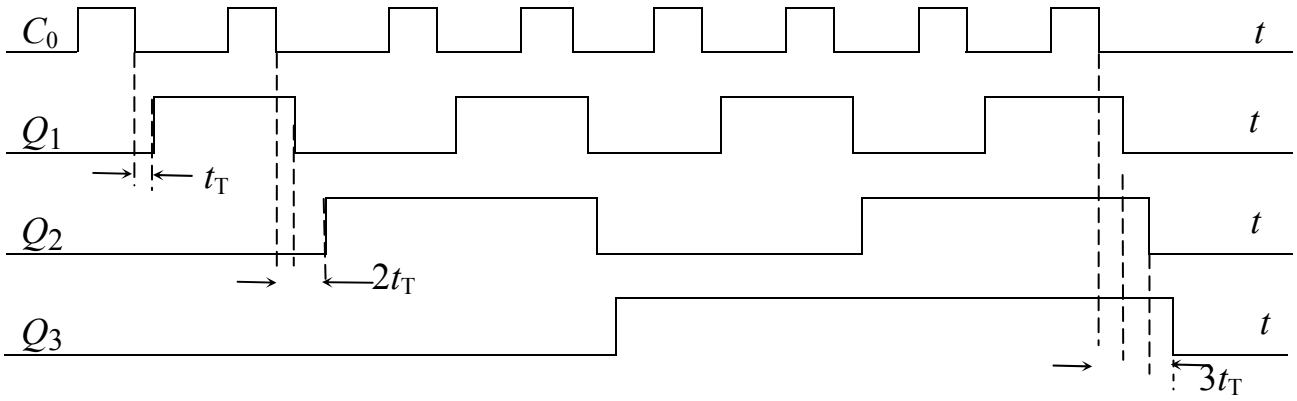


Рис. 5.3. Временная диаграмма двоичного суммирующего счетчика с K_C=8

Поскольку на счетных входах триггеров 2 и 3 не происходит изменения уровня с 1 на 0, эти триггеры сохраняют состояния $Q_2 = 0$, $Q_3 = 0$. С приходом второго сигнала триггер 1 переходит в состояние 0. В момент изменения уровня на его выходе с $Q_1 = 1$ на уровень $Q_1 = 0$ триггер 2 переходит в состояние 1 и на его выходе устанавливается уровень $Q_2 = 1$. Состояние триггера 3 остается неизменным. Триггер 3 перейдет в состояние 1 лишь при поступлении на счетный вход триггера 1 четвертого по счету сигнала. При этом триггер 1 перейдет из состояния 1 в состояние 0. Переход от состояния 1 к состоянию 0 вызовет изменение уровней от 1 к 0 на счетном входе 2. В результате триггер 2 также перейдет из состояния 1 в состояние 0. Такой переход повлечет за собой изменение уровня от 1 к 0 на счетном входе 3. В результате на выходе Q_3 триггера 3 установится уровень 1. При этом на выходах Q_1 и Q_2 триггеров 1 и 2 будут уровни 0. Следовательно, в счетчике будет зафиксировано число 4 в двоичном

представлении. Это соответствует фиксации момента поступления четвертого сигнала.

К моменту прихода восьмого по счету сигнала на выходах триггеров Q_1, Q_2, Q_3 будет установлен уровень 1. Поступление восьмого сигнала на счетный вход триггера 1 вызовет изменение его состояния с 1 на 0. В свою очередь, изменение состояния триггера 1 вызовет изменение состояния триггера 2, а изменение состояния триггера 2 приведет к изменению состояния триггера 3. В результате все триггеры счетчика перейдут в состояние 0. Счетчик будет подготовлен к подсчету новой последовательности из восьми сигналов.

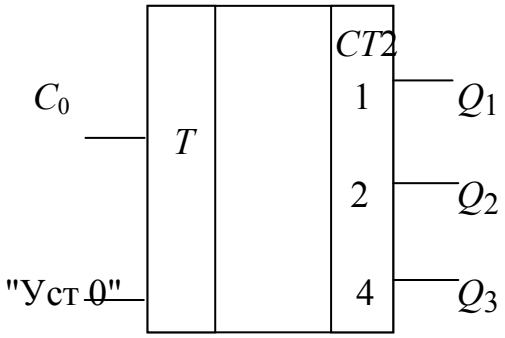


Рис. 5.4. Условное обозначение трехразрядного счетчика

Закон функционирования счетчика можно представить в виде табл. 5.1, условное изображение счетчика приведено на рис. 5.4.

Таблица 5.1. Закон функционирования трехразрядного счетчика

Номер состояния	C_0	Состояние триггеров					
		$Q^t{}_3$	$Q^t{}_2$	$Q^t{}_1$	$Q_3{}^{t+1}$	$Q_2{}^{t+1}$	$Q_1{}^{t+1}$
1	1	0	0	0	0	0	1
2	1	0	0	1	0	1	0
3	1	0	1	0	0	1	1
4	1	0	1	1	1	0	0
5	1	1	0	0	1	0	1
6	1	1	0	1	1	1	0
7	1	1	1	0	1	1	1
8	1	1	1	1	0	0	0

Работу счетчика можно также представить как процесс суммирования предыдущего значения счетчика с единицей. Такое суммирование выполняется по обычным правилам выполнения операции сложения чисел в двоичной системе. При этом можно отметить следующие особенности:

1) если в младшем разряде предыдущего значения счетчика имеется 0, то суммирование изменяет лишь цифру младшего разряда на 1;

2) если в m младших разрядах содержится 1, а в (m+1)-м разряде - 0, то цифры m младших разрядов изменяются на значение 0, а в (m+1)-м разряде - на значение 1.

Например,

$$\begin{array}{rr} 1)\ 110 & \qquad 2)\ 011 \\ + & \qquad + \\ \underline{\quad 1} & \qquad \underline{\quad 1} \\ 111 & \qquad 100 \end{array}$$

Таблица 5.2. Правило сложения для i-го разряда

a_i	P_i	c_i	P_{i+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Пусть $a_1,\ldots, a_n$ - цифры разрядов числа до суммирования, а $c_1, \ldots, c_n$- цифры разрядов числа, полученного в результате суммирования с единицей. Обозначим P_i- значение переноса, формируемого при сложении в (i-1)-ом разряде. Тогда P_{i+1} - перенос, формируем в i-м разряде. Правило сложения для i-го разряда числа с единицей может быть задано табл. 5.2.

Поступление 1 на вход счетчика можно представить как перенос P_1, подаваемый на его первый разряд.

Из табл. 5.2 следует, что значение $c_i = a_i\overline{P_i} \vee a_i\overline{P_i}$ получается в результате подачи переноса P_i; на счетный вход i-го разряда, хранившего a_i. Перенос, передаваемый в (i+1)-й разряд, определяется формулой $P_{i+1} = a_iP_i$.

Рассмотренный счетчик построен на последовательно соединенных T-триггерах. Каждый последующий разряд счетчика переключается сигналом переноса, формируемым на выходе предыдущего разряда. Сигналы для счета подаются на вход триггера самого младшего разряда. Счетчик, построенный таким образом, называется счетчиком с последовательным переносом или асинхронным счетчиком. Из временных диаграмм (рис. 5.3) видно, что в наихудшем случае новое состояние n-разрядного счетчика устанавливается с временной задержкой $t_{уст} = n_i\, t_T$, где $t_{уст}$ – время установки состояния счетчика с

последовательным переносом, а t_T - время переключения триггера. Схема счетчика с реализацией выражений c_i и P_{i+1} приведена на рис. 5.5.

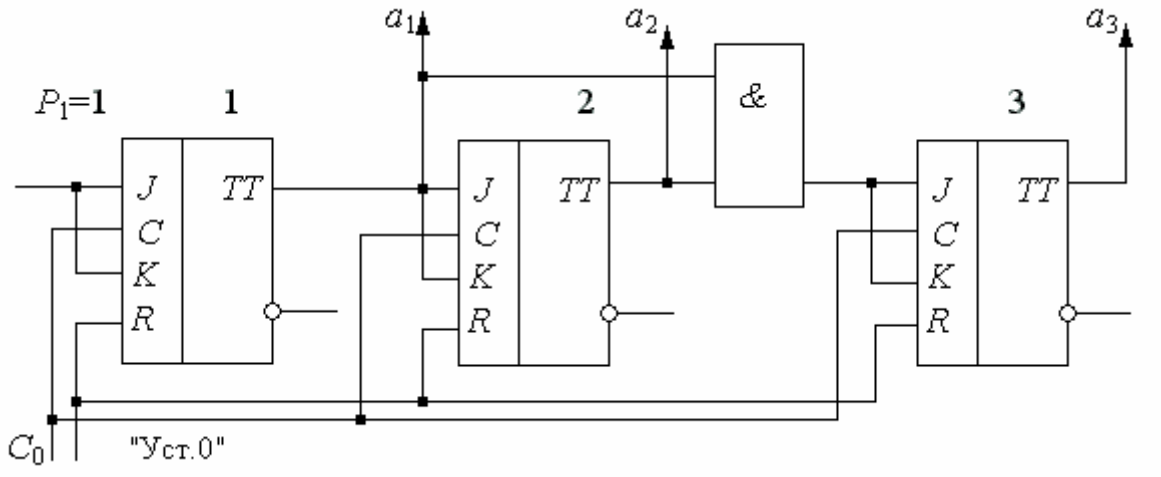


Рис. 5.5. Схема счетчика с реализацией c_i и P_{i+1}

Трехразрядный счетчик построен на *JK*-триггерах. Входы *J* и *K* объединяются, и на объединенный вход подаются переносы. Входные сигналы, число которых надо подсчитать, подаются на входы синхронизации триггеров. Для формирования переносов использованы элементы И.

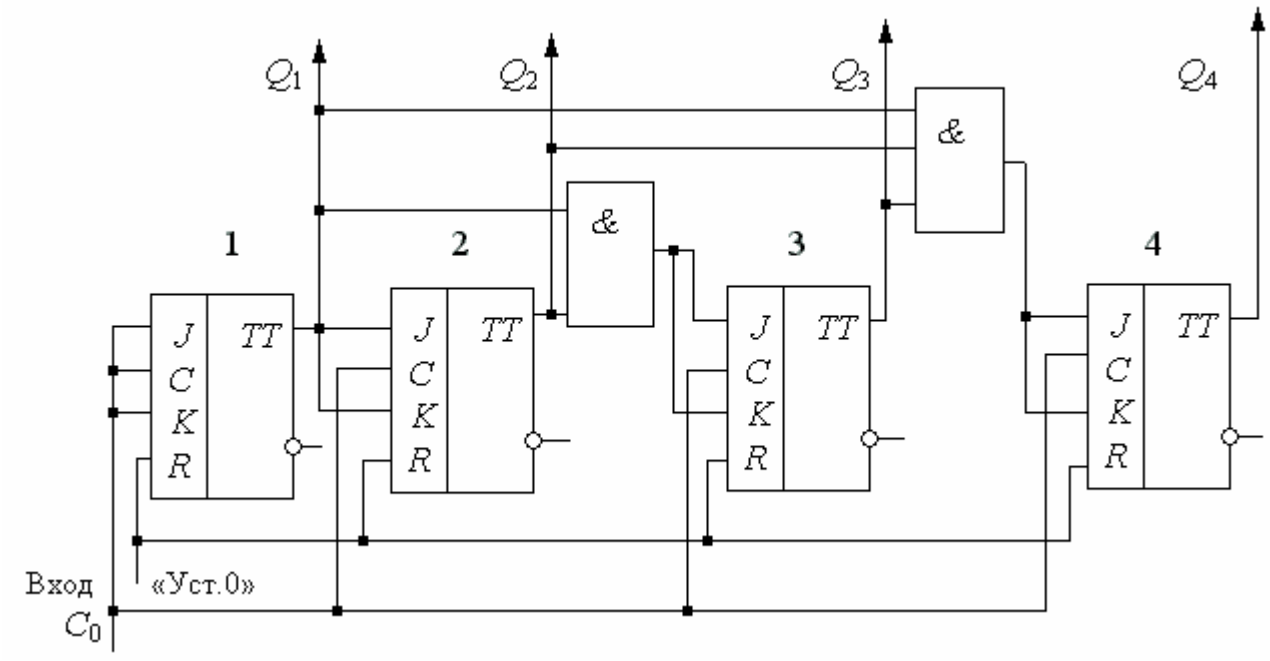


Рис. 5.6. Схема счетчика с параллельным переносом с $K_C=2^4$

Счетчик может быть установлен в нулевое состояние посылкой сигнала по цепи «Уст. 0». С каждым входным сигналом числовое значение в счетчике увеличивается на единицу. С приходом 2^3 сигнала в счетчике устанавливается исходное (нулевое) состояние. В рассматриваемой схеме счетчика процесс переносов также является последовательным. Время задержки переносов растет с ростом числа разрядов в счетчике. Это время задержки ограничивает максимальную частоту подачи сигналов на вход, тем самым ограничивается быстродействие счетчика. Для уменьшения времени задержки распространения переносов могут использоваться счетчики с параллельным переносом (рис. 5.6). В этом случае

$$t_{уст} = t_Т + t_Э ,$$

где $t_Э$- время задержки логического элемента.

Условное обозначение счетчика дано на рис. 5.7, для этой схемы логические выражения переносов будут иметь вид

$$P_1=1; \quad P_2=a_1; \quad P_3=a_1a_2;$$
$$P_4=a_1a_2a_3; \quad \ldots \quad P_n=a_1a_2\ldots a_{n\text{-}1}.$$

Здесь задержка определяется только одном элементе И и не зависит от числа разрядов в счетчике. Необходимо отметить, что такой подход приводит к усложнению счетчика, поскольку используются элементы И с большим числом входов. Кроме того, необходимость включения в схему счетчика элементов И с нарастающим от разряда к разряду числом входов нарушает регулярность его структуры. Поэтому при построении многоразрядных счетчиков используются схемы с параллельно-последовательным переносом.

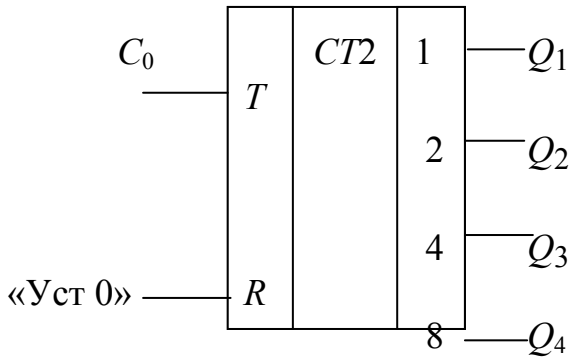


Рис. 5.7. Условное обозначение счетчика с параллельным переносом с $K_С=2^4$

Схема счетчика с параллельно-последовательным переносом состоит из группы триггеров, внутри каждой, из которой организуется параллельный пе-

ренос, а между группами – последовательный. Счетчик, схема которого приведена на рис. 5.8, состоит из 4-разрядных счетчиков с параллельным переносом.

На входе каждого такого счетчика включен элемент И с пятью входами.

В нем формируется сигнал переноса в следующую группу при заполнении предыдущей группы триггеров единицами. Задержка в многоразрядном счетчике будет пропорциональна числу групп в счетчике.

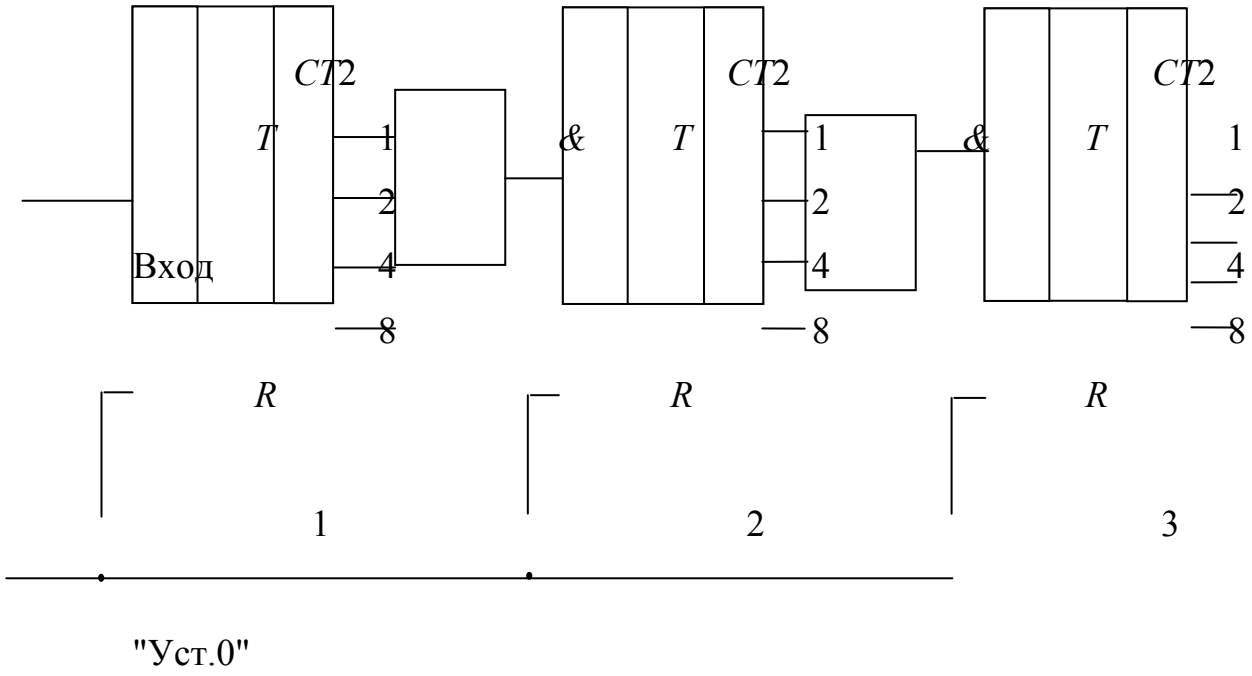


Рис. 5.8. Схема вычитающего счетчика с параллельно-последовательным переносом

5.3. Вычитающие и реверсивные двоичные счетчики

В вычитающих счетчиках с приходом очередного счетного сигнала предыдущий результат уменьшается на единицу, в вычитающем двоичном *n*-разрядном счетчике реализуется счетная последовательность чисел, начиная с 2^n-1 и кончая 0. Очередное число в этой последовательности получается вычитанием единицы из предыдущего числа. После получения значения 0 последовательность повторяется. Порядок смены состояний вычитающего счетчика может быть описан табл. 5.3.

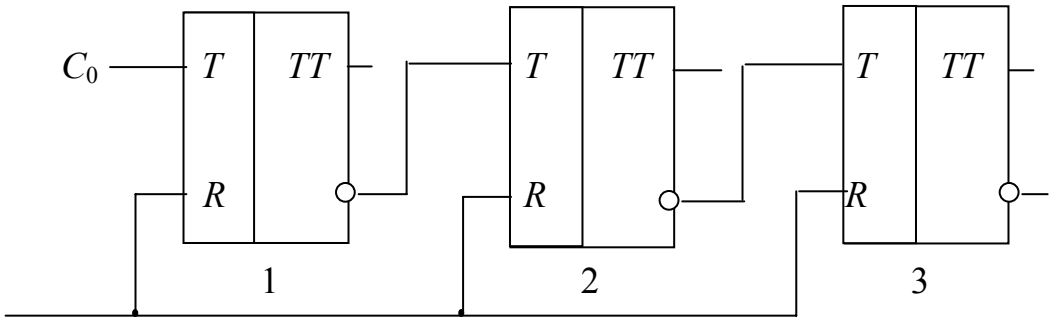
Таблица 5.3. Состояния двоичного вычитающего счетчика

Номер состояния	C_0	Состояние триггеров					
		Q^t_3	Q^t_2	Q^t_1	Q_3^{t+1}	Q_2^{t+1}	Q_1^{t+1}
1	1	1	1	1	1	1	0
2	1	1	1	0	1	0	1
3	1	1	0	1	1	0	0
4	1	1	0	0	0	1	1
5	1	0	1	1	0	1	0
6	1	0	1	0	0	0	1
7	1	0	0	1	0	0	0
8	1	0	0	0	1	1	1

Из таблицы следует еще одно отличие вычитающего счетчика от суммирующего: триггер каждого последующего разряда переходит в другое состояние при сигнале займа, обратном сигналу переноса в суммирующем счетчике. Поэтому вычитающий счетчик в отличие от суммирующего строится так, что со входом каждого последующего триггера соединяется инверсный выход предыдущего триггера. Схема вычитающего счетчика с последовательной передачей переносов приведена на рис. 5.9.

Функционирование *i*-го разряда вычитающего счетчика можно описать логическими выражениями переноса P_{i+1} и разности C_i из табл. 5.4. Таблица получается на основании анализа особенностей вычитания единицы из двоичного числа. Из табл. 5.4 следуют логические выражения :

$$C_i = \bar{a}_i P_i \vee a_i \bar{P}_i ; \qquad P_{i+1} = a_i P_i .$$



"Уст.0"

Рис. 5.9. Схема вычитающего счетчика с последовательным переносом

Разность C_i определяется тем же выражением, что и в суммирующем счетчике, поэтому перенос должен подаваться на счетный вход триггера *T*. В отличие от суммирующего счетчика в выражении для $P_{i+1} \cdot a_i$ используется $\overline{a_i}$. Следовательно, на элементы И**,** формирующие переносы, подаются сигналы с

инверсных выходов триггера. Для ускорения работы вычитающих двоичных счетчиков могут использоваться схемы с параллельным и параллельно-последовательным переносом.

В реверсивном счетчике объединяются схемы суммирующего и вычитающего счетчиков. Кроме того, существует возможность управления направлением счетчика, для чего предусматривается дополнительное КЦУ.

В реверсивном счетчике на *Т*-триггерах (рис. 5.10) счетные сигналы поступают на вход *Т*-триггера через логические элементы в случае, если они открыты единичными сигналами с выходов предыдущих разрядов. Для счетных сигналов предусмотрены два входа. Если счетчик работает как суммирующий, сигналы счета следует подавать на вход +1.

Таблица 5.4. Правило вычитания для *i*-го разряда

b_i	P_i	c_i	P_{i+1}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Для вычитающего счетчика сигналы счета подаются на вход -1. На выходе счетчика, обозначенном >15, сигнал появляется при переходе счетчика в состояние с номером 15, в котором все триггеры установлены в состояние 1. На этом выходе формируется сигнал переноса в следующий счетчик. На выходе <0 сигнал появляется при заполнении счетчика нулями. Это сигнал займа в следующий разряд в схеме вычитающего счетчика. Условное обозначение реверсивного счетчика с двумя входами приведено на рис. 5.11.

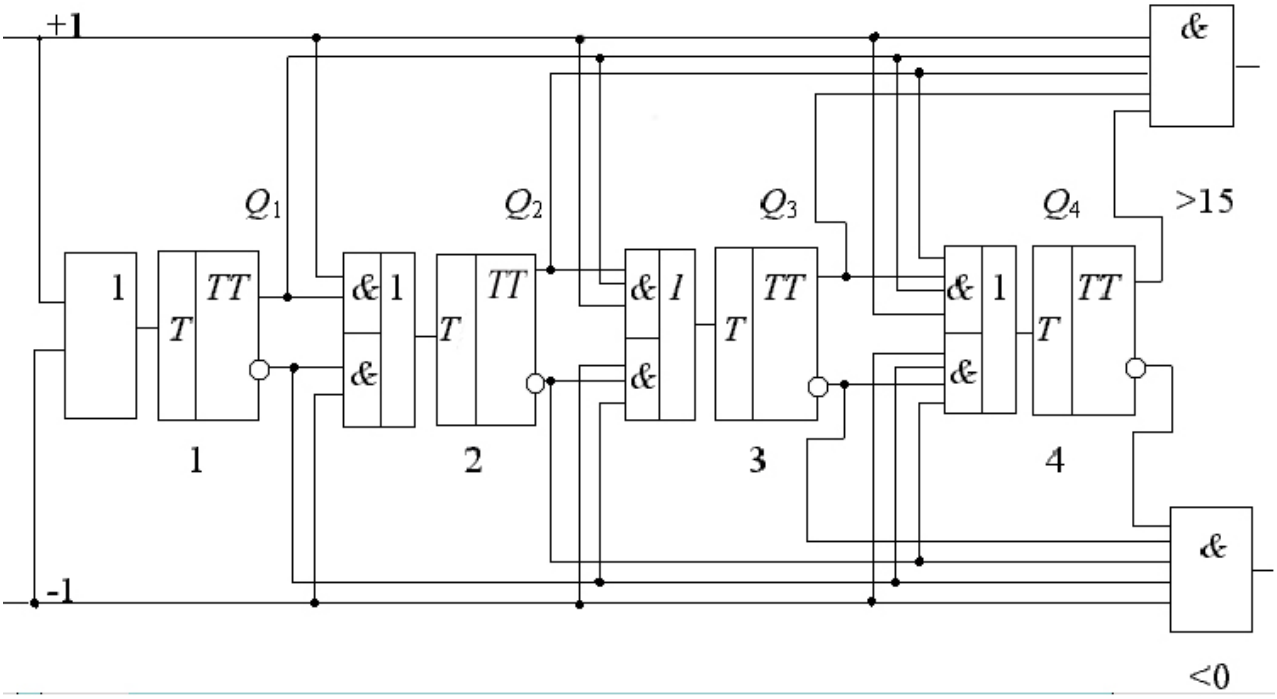


Рис. 5.10. Схема реверсивного счетчика на *T*-триггерах

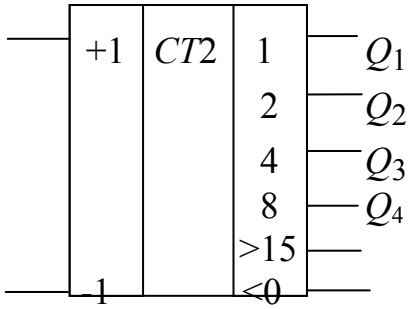


Рис. 5.11. Условное обозначение реверсивного счетчика на *T*-триггерах

Если требуется построить реверсивный счетчик с одним источником сигналов для счета, то необходимо предусмотреть специальное ПЦУ для переключения на суммирующий +1 или вычитающий -1 входы (рис. 5.12).

При подаче сигнала на вход C_c *RS*-триггер установится в единичное состояние. Сигналы счета C_0 будут поступать на вход +1 реверсивного счетчика, который будет работать как суммирующий. При подаче сигнала на вход C_B*RS*-триггер установится в нулевое состояние. Сигналы счета со входа C_0 будут поступать на вход -1, и счетчик будет работать в режиме вычитающего счетчика.

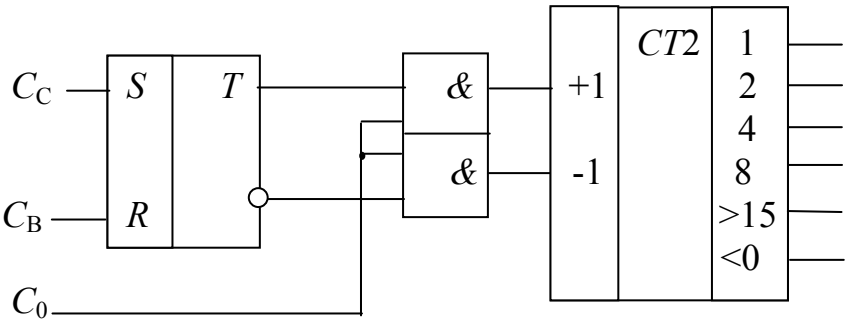


Рис. 5.12. Схема переключения режимов работы счетчиков

Используя интегральные схемы 4-разрядных счетчиков с выходами займа и переноса, можно строить реверсивные счетчики большей разрядности. На рис. 5.13 приведен пример построения 8-разрядного реверсивного счетчика из двух 4-разрядных.

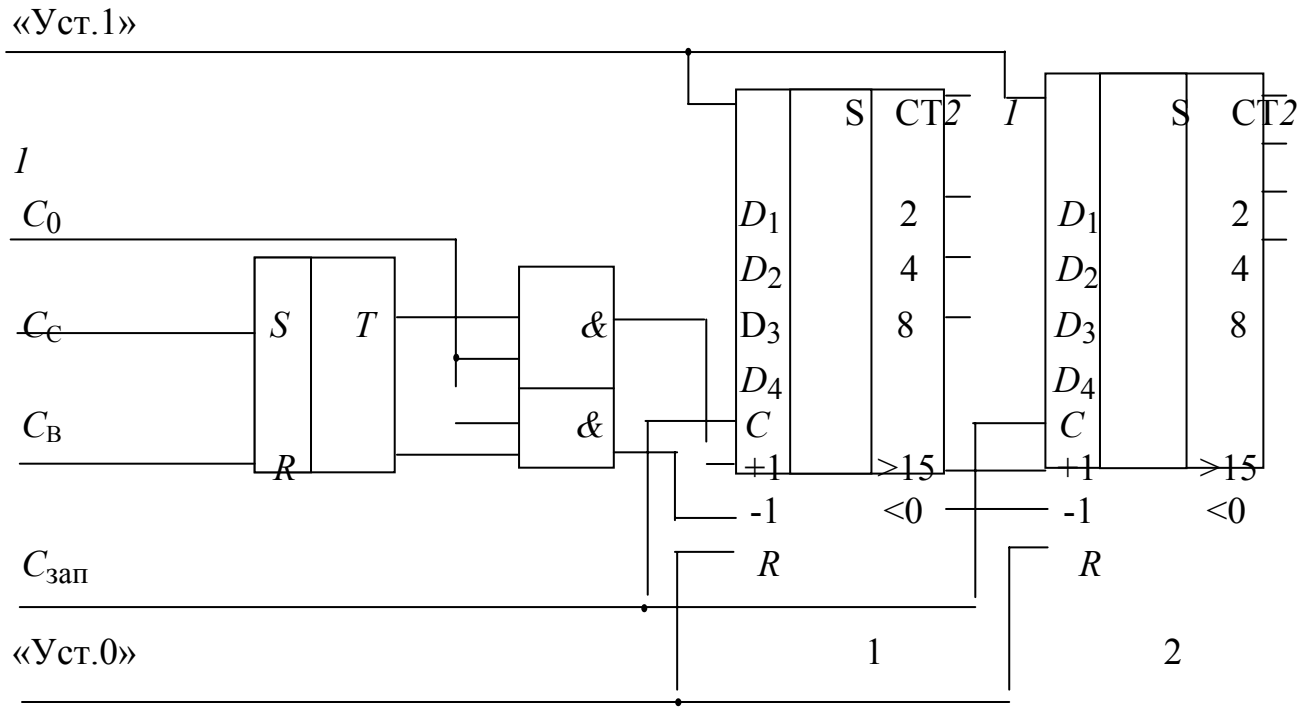
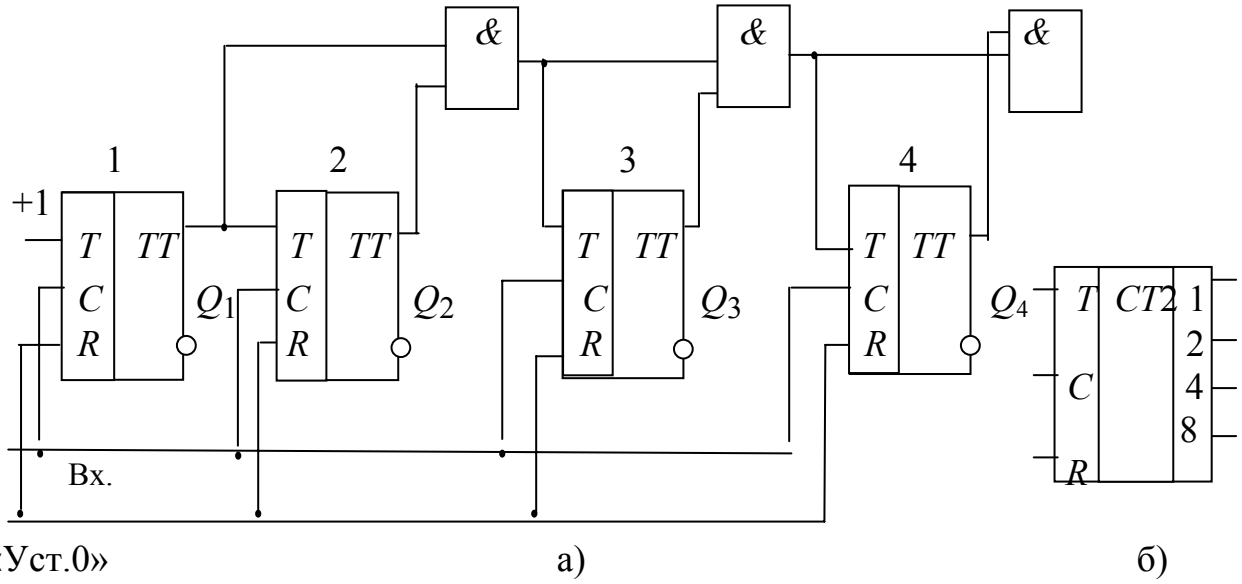


Рис. 5.13. Схема реверсивного восьмиразрядного счетчика

Здесь допускается установка заданного исходного состояния счетчика с помощью ввода в счетчик нужной кодовой комбинации $a_0, a_1, ..., a_7$ при наличии разрешения на входе записи $C_{зап}$. Используя возможность посылки сигналов на вход «Уст. 1» или «Уст. 0» в триггерах счетчика можно установить состояния 0 или 1. При необходимости можно использовать выходы >15 первого или второго 4- разрядных счетчиков. Частота повторения сигналов на этих выходах в 16 или 256 раз меньше частоты с сигналов, поступающих на вход C_0, соответственно.

5.4. Синхронные и асинхронные двоичные счетчики.

Двоичные счетчики, состояние триггеров которых изменяется одновременно под воздействием сигнала синхронизации на входах всех триггеров, получили название синхронных.



«Уст.0» а) б)

Рис. 5.14. Схема а) и условные обозначения б) синхронного счетчика

со сквозным переносом на *T* - триггерах

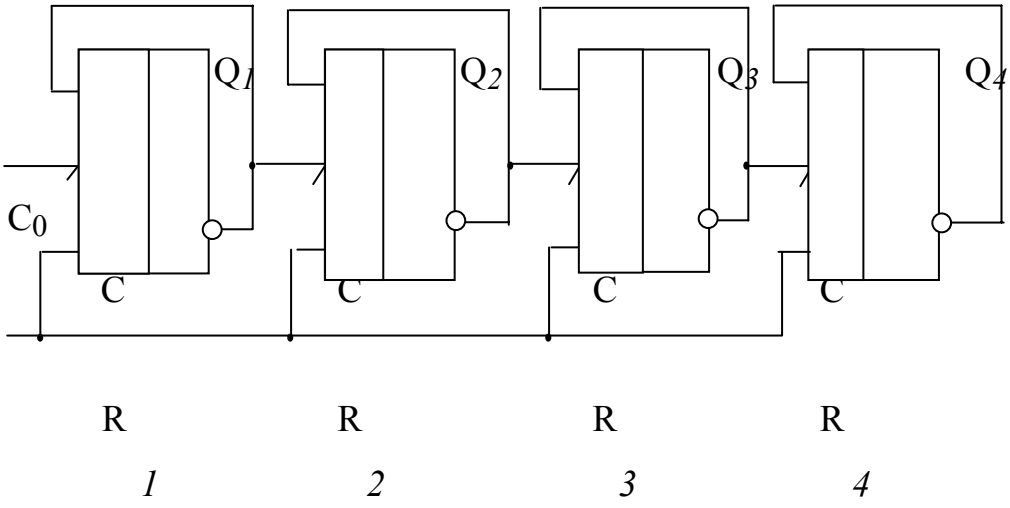
Схема синхронного счетчика со сквозным переносом на *T*-триггерах приведена на рис. 5.14а, его условное обозначение дано на рис. 5.14б.

Синхронные счетчики используются в синхронных цифровых системах. Последовательностные цифровые устройства в этих системах обычно зависят друг от друга и управляются от общего источника синхросигналов. В таких условиях нужно, чтобы все триггеры во всех ПЦУ изменяли свое состояние одновременно но сигналу синхронизации, чтобы текущее состояние триггеров использовалось для определения их следующего состояния. Применяемая здесь схема со сквозным переносом легко наращивается простым добавлением схемы И с двумя входами. Однако для определения значения самого правого входа *T* n-разрядного счетчика необходимо время, равное времени распространения сигнала через одну схему И, умноженному на *n*-1. Время установки счетчика $t_{уст} = t_Т + (n\text{-}1)t_Э$.

Различные схемы синхронных счетчиков были использованы при рассмотрении суммирующих и вычитающих счетчиков (см. рис. 5.5, 5.6, 5.8, 5.10).

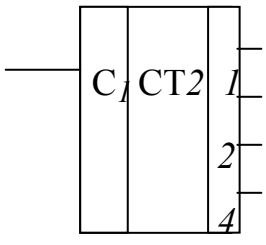
В асинхронных счетчиках синхронизирующие входы триггеров соединяются с входами соседних триггеров. Поэтому состояние триггера меняется в ответ на изменение состояния соседнего триггера, а не в ответ на воздействие сигнала внешней синхронизации.

В асинхронных счетчиках волной изменений состояния распространяется по всей цепочке триггеров, в отличие от синхронных счетчиков, где происходит изменение состояния всех триггеров одновременно. Схема асинхронного счетчика на *D*-триггерах с динамическим управлением приведена на рис. 5.15 его условное обозначение – на рис. 5.16. В триггерах с прямым динамическим входом изменение состояния осуществляется при перепаде от уровня 0 к 1.



«Уст.0»

Рис. 5.15. Схема асинхронного двоичного счетчика



«Уст.0» R *8*

Рис. 5.16. Условное обозначение асинхронного двоичного счетчика

В асинхронных счетчиках с последовательным переносом вход каждого последующего триггера соединяется с инверсным выходом предыдущего. Сигналы счета поступают на вход C_0. *С помощью сигнала, поступающего на вход «Уст. 0», счетчик может быть установлен в начальное состояние.*

Асинхронные счетчики также были рассмотрены при описании суммирующих и вычитающих счетчиков (см. рис. 5.2, 5.9). Асинхронные счетчики позволяют обеспечить большую скорость счета. Объясняется это тем, что после переключения первого триггера счетчика на него можно подавать следующий сигнал, не ожидая распространения воздействия от сигнала через весь счетчик. В синхронном счетчике между сигналами на счетный вход должно проходить время, определяемое переключением одного триггера и установлением значений на всех входах T. *С учетом сказанного можно отметить, что все типы суммирующих, вычитающих, реверсивных счетчиков могут быть реализованы в виде как синхронных, так и асинхронных счетчиков.*

5.5. Десятичные счетчики

На практике возникает потребность в построении счетчиков по произвольному модулю K_C с числом двоичных разрядов, выбираемых исходя из условия $2^n < K_C < 2^{n+1}$.

Это вызывает необходимость исключения лишних состояний в счетчике. Рассмотрим особенности построения таких счетчиков на примере десятичного счетчика.

Для построения счетчика с $K_C = 10$ необходимо иметь 4-разрядный двоичный счетчик, число состоянии которого следует уменьшить с 16 до 10. Счетная последовательность десятичного счетчика может быть представлена в двоично-кодированном десятичном коде ($Q_4\ Q_3\ Q_2\ Q_1$ - 8421), в котором каждая десятичная цифра кодируется 4 разрядным двоичным числом. Счетная последовательность суммирующего десятичного счетчика в этом случае совпадает с двоичной последовательностью от 0000 до 1001, после чего следует 0 и последовательность повторяется. Схема десятичного счетчика на *JK*-триггерах с входной логикой приведена на рис. 5.17.

Десятичный счетчик представляет собой 4-разрядный двоичный суммирующий счетчик с параллельным переносом, схема которого дополнена обратной связью с выхода $\overline{Q}_4$ на вход J-триггера 2. Кроме того, выход Q_1 соединен со всеми выходами К. триггера 4. После приема в счетчик восьми сигналов на его выходах $Q_1\,Q_2\,Q_3\,Q_4$ установятся соответственно уровни 0001. При этом на вход J-триггера 2 будет подаваться нулевой уровень с выхода $\overline{Q}_4$. С приходом девятого сигнала на выходе Q_1 триггера 1 установится единичный уровень. С приходом десятого сигнала триггер 1 и триггер 4 перейдут в нулевое состояние. Триггеры 2 и 3 сохранят нулевое состояние. Десятым сигналом счетчик установится в начальное состояние, и процесс счета будет повторяться. Закон функционирования десятичного счетчика приведен в табл. 5.5.

В общем случае при построении счетчиков по произвольному модулю K_C в схему соответствующего двоичного счетчика вводится обратная связь для исключения лишних состоянии. Двоичный суммирующий или вычитающий счетчик переключается до установки некоторого значения K_C.

Это состояние выявляется специальной схемой, на выходе которой формируется сигнал установки счетчика в нулевое состояние. Схемы счетчика по модулю K_C - наиболее просты при использовании двоичных счетчиков с последовательным переносом. Рассмотрим построенную таким способом схему десятичного счетчика (рис. 5.18).

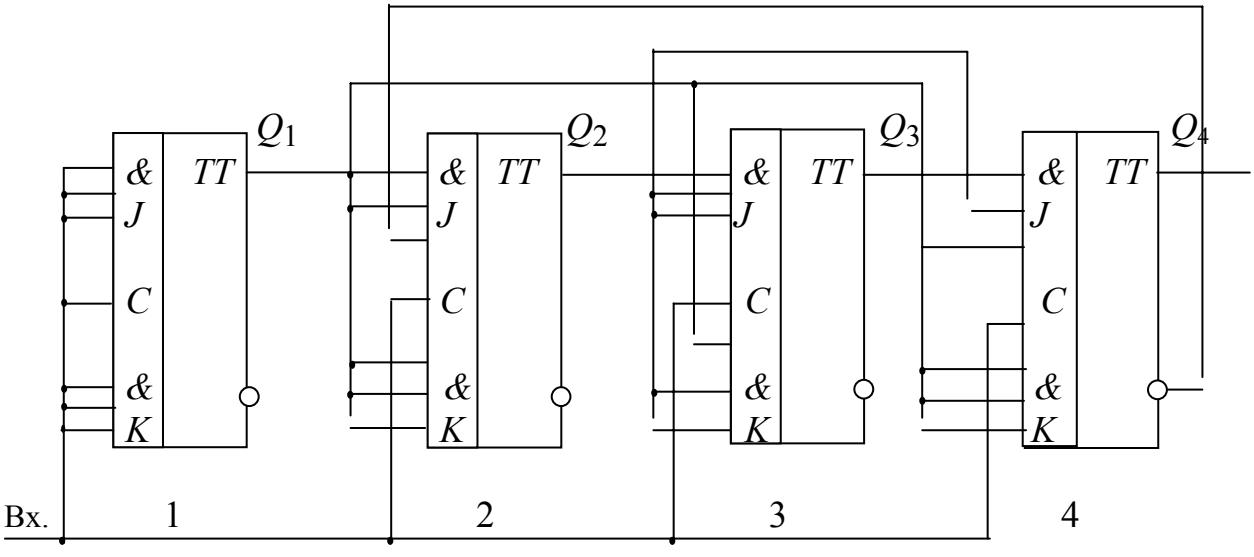


Рис. 5.17. Схема десятичного счетчика

Работа счетчика заключается в следующем. Начиная с исходного нулевого состояния счетчик принимает последовательность сигналов счета. Триггер 5 находится в состоянии 0. Элемент И фиксирует момент прихода десятого сиг-

нала (комбинация уровней на выходах триггеров Q_4, Q_3, Q_2, Q_1 1010) и переводит триггер 5 в состояние 1. До прихода одиннадцатого сигнала все разряды счетчика переводятся в нулевое состояние установочным сигналом с выхода $\overline{Q}_5$. После поступления десяти сигналов счетчик возвращается в исходное положение. Одиннадцатый сигнал положительным перепадом возвращает единичный триггер 5 в исходное состояние 0.

Счетчик-делитель с произвольным модулем счета K_C можно реализовать на одних *JK*-триггерах без дополнительных элементов И (вентилей). Такие счетчики, получившие название безвентильных, можно наращивать без дополнительных логических элементов. Быстродействие безвентильных счетчиков зависит от их структуры.

Таблица 5.5. Закон функционирования десятичного счетчика

Номер комбинации	Вх.	Состояния триггеров							
		$Q^t{}_4$	$Q^t{}_3$	$Q^t{}_2$	$Q^t{}_1$	$Q_4{}^{t+1}$	$Q_3{}^{t+1}$	$Q_2{}^{t+1}$	$Q_1{}^{t+1}$
1	1	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	0	1	0
3	1	0	0	1	0	0	0	1	1
4	1	0	0	1	1	0	1	0	0
5	1	0	1	0	0	0	1	0	1
6	1	0	1	0	1	0	1	1	0
7	1	0	1	1	0	0	1	1	1
8	1	0	1	1	1	1	0	0	0
9	1	1	0	0	0	1	0	0	1
10	1	1	0	0	1	0	0	0	0

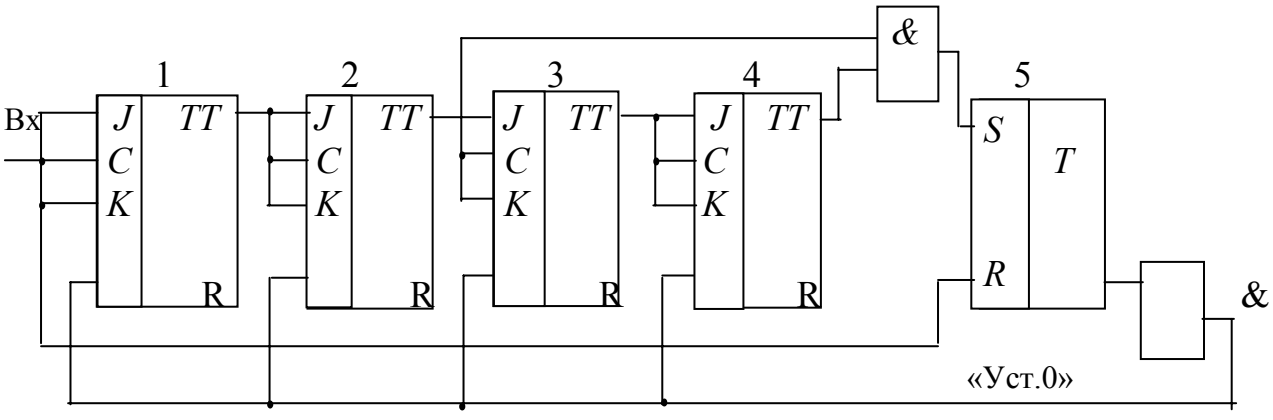


Рис. 5.18. Схема двоичного счетчика с последовательным переносом с K_C

Счетчики-делители строятся в соответствии со следующими правилами:

1) заданный коэффициент счета K_C разлагается на сомножители $K_C=2^c[2^b(2^a+1)+1]$, где *a, b, c*—целые числа 1, 2, 3,… ;

2) функциональная схема счетчика представляется совокупностью схем счетчиков с коэффициентами счета – сомножителями $K_{10} = K_2 \cdot K_5$, где $K_2=2$, $K_5=5$;

3) функциональная схема реализуется на *JK*-триггерах. Внутри каждого из счетчиков *JK*-триггеры соединяются по схеме последовательного переноса

для вычитающего счетчика. Выходной сигнал снимается с выхода 2^a ;

4) для каждого сомножителя строится схема счетчика с коэффициентами счета 2^a, 2^b, 2^c, ... и дополнительными триггерами для увеличения коэффициента счета на единицу. *J*- вход каждого дополнительного *JK*-триггера подсоединяется к прямому выходу последнего разряда счетчика, *C* - вход – к *C* - входу первого разряда счетчика, выход $\overline{Q}$ - к *J*-входу первого разряда счетчика, *K*-вход с уровнем 1.

Схема десятичного счетчика, соответствующая этим правилам, дана на рис. 5.19.

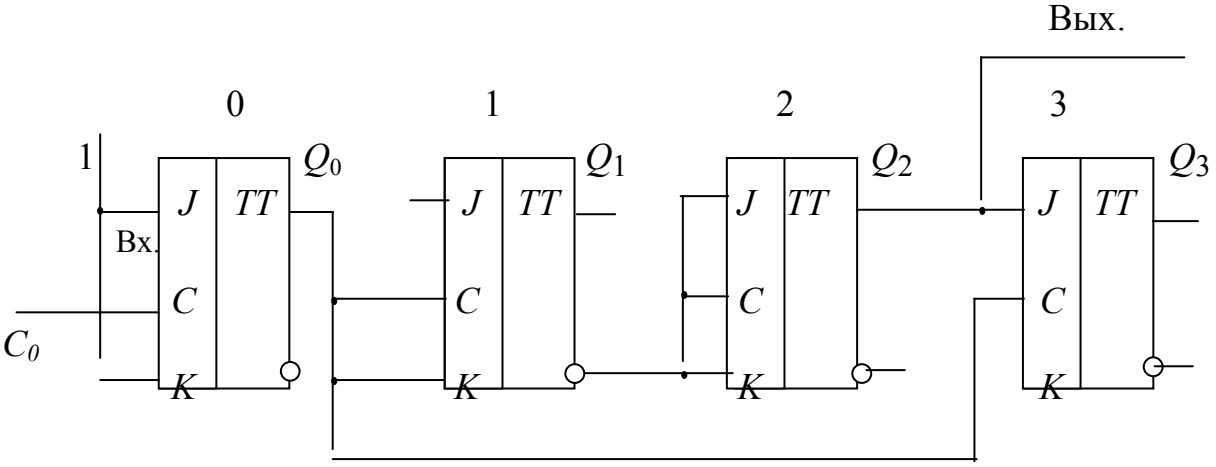


Рис. 5.19. Схема десятичного счетчика

Временная диаграмма для счетчика с коэффициентом счета $K_5 = 2^2 + 1$ приведена на рис. 5.20.

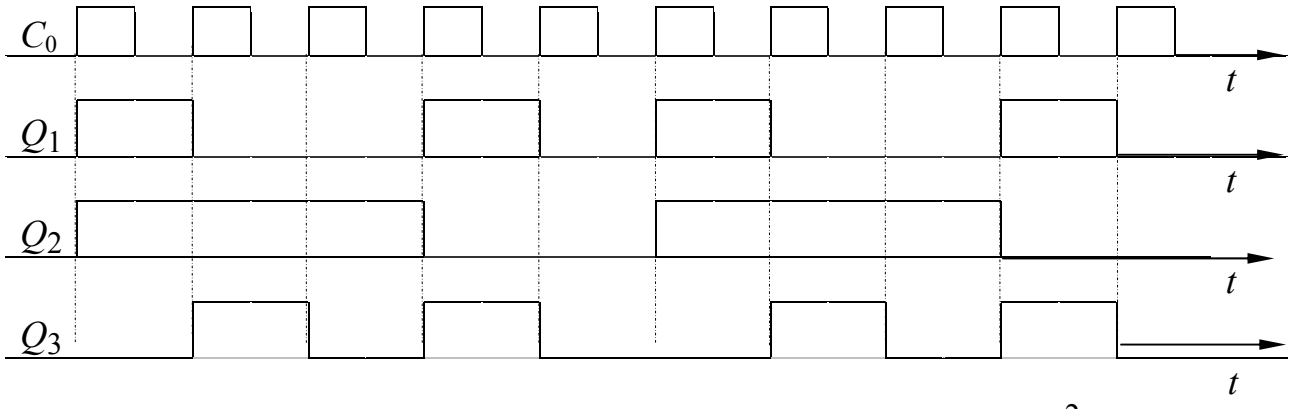


Рис. 5.20. Временная диаграмма для счетчика с $K_5 = 2^2 + 1$

Для реализации такого счетчика-делителя необходимы четыре *JK*-триггера. Триггер 1 осуществляет деление на 2, поступающих на вход сигналов. Выход триггера *J* подсоединяется к входу счетчика-делителя на 5, в котором триггеры 2 и 3 соединены в схему вычитающего счетчика с последовательным переносом. Триггер 4 используется для увеличения коэффициента счета 2^2 на единицу. Выходом делителя является выход триггера 3.

В приведенных схемах счетчиков счет начинался с нуля и завершался после достижения заданной величины модуля счета установкой счетчика в нулевое состояние. Другими словами, в таких счетчиках исключаются старшие состояния. Возможен также вариант построения счетчиков с исключением младших состояний. Здесь счет будет начинаться с некоторого числа, равного разности $M = 2^n - K_C$. При этом в счетчике будет производиться последовательно смена состояний $M, M+1, \ldots, M + K_C - 1, 2^n$. Такой способ известен как способ счета с предварительной установкой. Схема десятичного счетчика с предварительной установкой приведена на рис. 5.21.

В качестве начального состояния на счетчике устанавливается значение $2^4 - 10_{10} = 6_{10}$. Сигнал *x* появится на выходе элемента & тогда, когда на счетчике будет установлено значение $2^n - 1$ и придет очередной сигнал на вход счетчика. Сигналом *x* будут установлены в единичное состояние триггеры 2 и 3, а триггеры 1 и 4 останутся в нулевом состоянии. В начале счетчик необходимо установить в исходное состояние $\overline{Q}_1 Q_2 Q_3 \overline{Q}_4$ с помощью сигнала «Уст. начальное состояние». Этот сигнал непосредственно используется для предварительной установки требуемого начального состояния.

Недостатком данного способа построения десятичного счетчика является то, что нарушается естественный порядок счета (начинается не с значения 0_{10} , а с 6_{10}). Достоинством является простота выработки сигнала окончания счета.

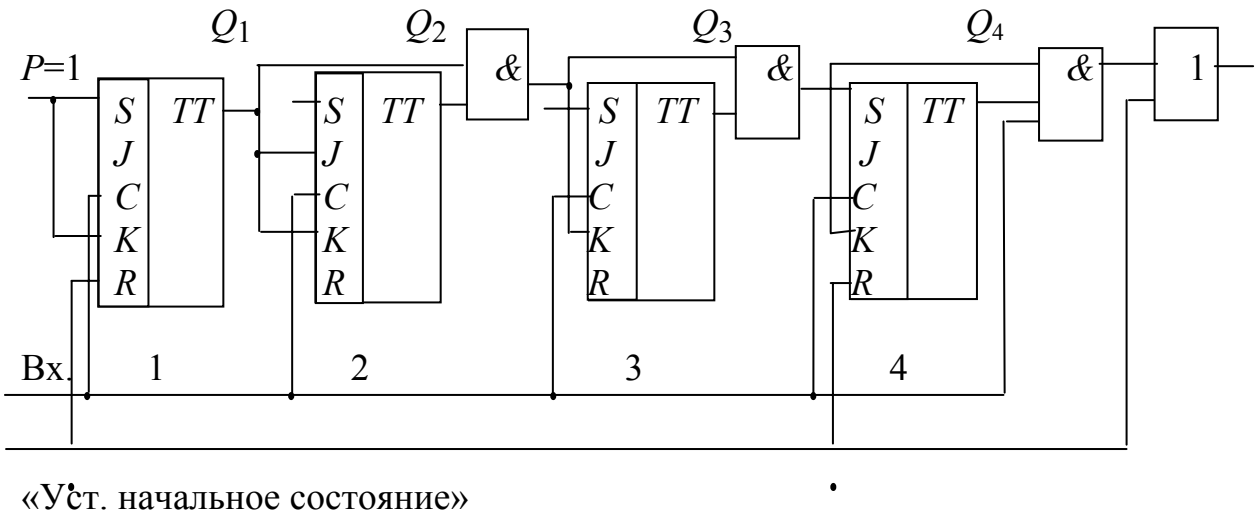


Рис. 5.21. Схема десятичного счетчика с предустановкой

Вполне очевидно, что все способы, использованные для построения десятичных счетчиков на основе двоичных, могут быть распространены и на счетчики с произвольным модулем $K_C<2^n$. Такие счетчики используются в качестве делителей частоты.

5.6. Делители частоты

В делителях частоты входная периодическая последовательность сигналов формирует на выводе периодическую последовательность с частотой, в N раз меньшей входной.

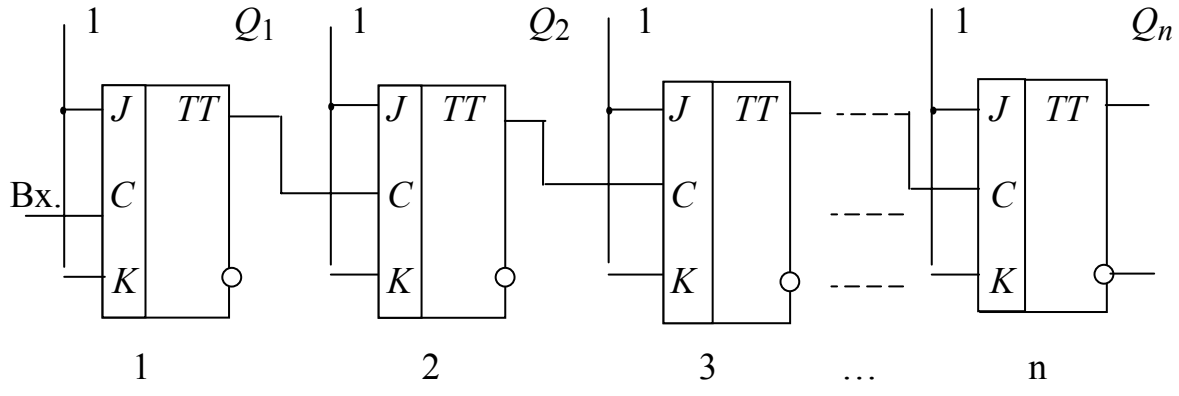


Рис. 5.22. Счетчик- делитель

Последовательность состояний делителя частоты может быть произвольной, важно лишь обеспечить заданный коэффициент счета $K_c = N$.

Счетчик-делитель с $N = 2^n$ может быть реализован по схеме двоичного счетчика с последовательным переносом. На выходе каждого разряда счетчика частота следования сигналов вдвое ниже, чем на входе.

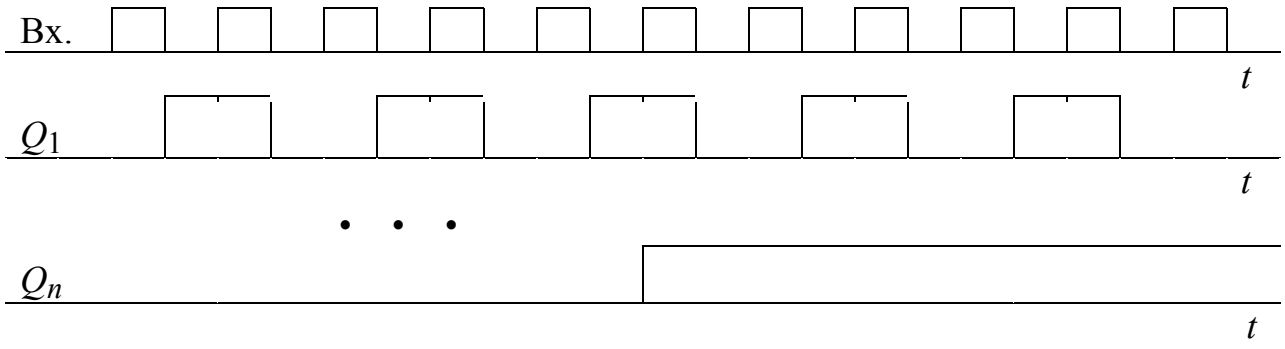


Рис. 5.23. Временная диаграмма счетчика-делителя

При n разрядах в счетчике частота выходной последовательности сигналов окажется в 2^n меньше по сравнению с входной последовательностью.

На рис.5.22 представлена схема счетчика-делителя, и на рис. 5.23 – временная диаграмма.

Счетчик-делитель с $N = 3$ может быть построен в виде безвентильного счетчика на *JK*-триггерах. Модуль $N = 3$ может быть представлен в виде $N = 2 + 1$. Для построения счетчика достаточно двух *JK*-триггеров. Схема счетчика-делителя с $N = 3$ приведена на рис. 5.24.

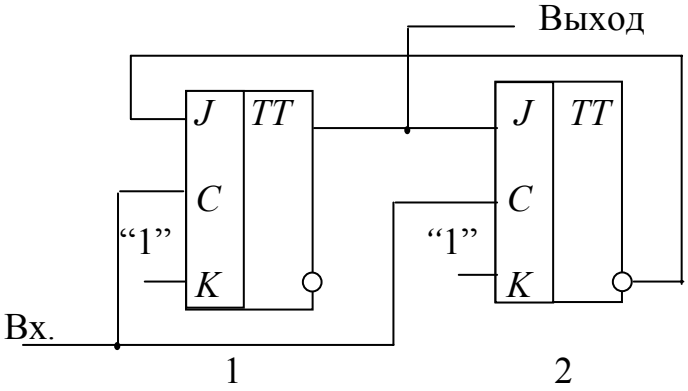


Рис. 5.24. Счетчик делитель $N = 3$

Каскадные счетчики-делители получаются в результате последовательного соединения счетчиков-делителей с коэффициентами деления N_1, N_2, …, N_K для уменьшения частоты следования сигналов в $N = N_1 N_2 \ldots N_K$ раз. Пусть требуется построить счетчик-делитель с $N = 27$ (здесь $N = 27 = 3 \cdot 3 \cdot 3$). Для по-

строения счетчика-делителя с $N = 27$ необходимо соединить последовательно три каскада счетчиков-делителей с $N = 3$ (рис. 5.25).

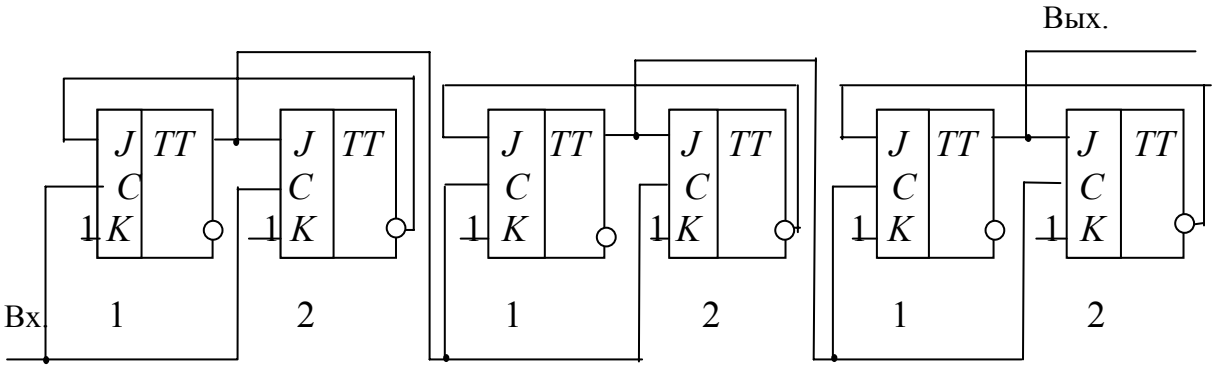


Рис. 5.25. Счетчик-делитель $N = 27$

Кольцевые счетчики представляют собой регистры сдвига, в которых информация циркулирует по замкнутому кругу. Контур образуется соединением старшего разряда регистра с младшим регистром цепью сдвига.

В схеме кольцевого счетчика (рис. 5.26) при подаче сигнала установки только триггер 1 будет находиться в состоянии 0. Состояние счетчика, соответствующее значению $Q_1\overline{Q}_2\overline{Q}_3\overline{Q}_4$, будет называться начальным. С приходом сигнала на вход кольцевого счетчика единица из триггера 1 будет переписана в разряд триггера 2. При этом счетчик перейдет в состояние $\overline{Q}_1Q_2\overline{Q}_3\overline{Q}_4$. После подачи трех сигналов в состоянии 1 окажется триггер 4. С приходом четвертого сигнала счетчик перейдет в начальное состояние, и снова триггер 1 будет в состоянии 1. Период цикла кольцевого счетчика на регистре сдвига определяется числом триггеров в счетчике. Кольцевые счетчики могут использоваться в качестве распределителей сигналов.

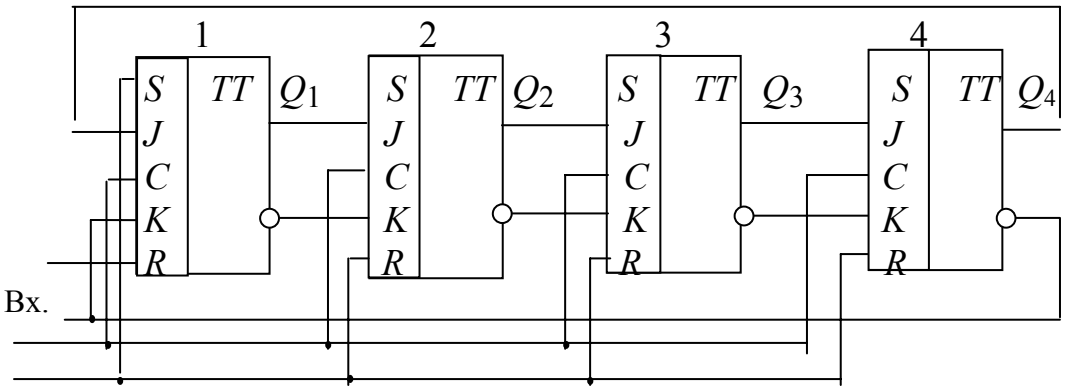


Рис. 5.26. Кольцевой счетчик

5.7. Распределители

В этих устройствах входные сигналы, поступающие на один вход, распределяются по N выходам таким образом, что в каждый такт по очереди выдается один сигнал на один из N выходов в порядке их нумерации.

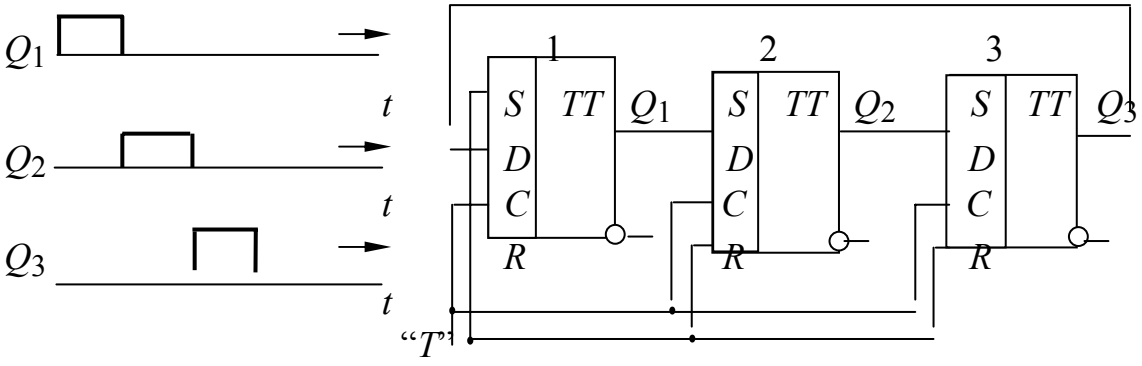


Рис. 5.27. Схема распределителя с кольцевым счетчиком

Схема распределителя с кольцевым счетчиком представлена на рис. 5.27. Распределители являются элементами устройств управления цифровых систем, работающих по заданной программе. Вначале триггер 1 находится в единичном состоянии, а остальные триггеры — в нулевом. Сигналы, поступающие на вход счетчика, переключают его из одного состояния в другое с периодом, равным трем. Соответственно на выходах Q_1, Q_2 и Q_3 будут по очереди появляться выходные сигналы. В каждый данный момент выходной сигнал будет присутствовать только на одном из выходов.

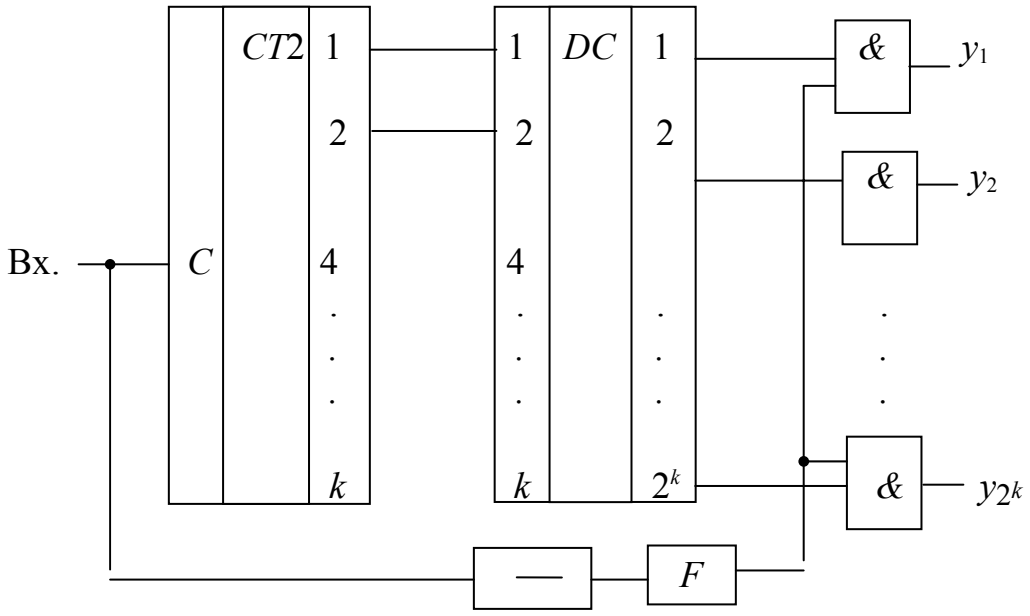


Рис. 5.28. Схема распределителя

Распределители можно строить и с использованием двоичных счетчиков. В этом случае потребуются дешифраторы для выделения и управления выходными цепями. Схема на двоичном счетчике с использованием дешифратора приведена на рис. 5.28 (*F*—формирователь). В этой схеме выходные сигналы появляются периодически (с периодом 2^k) на выходных цепях $y_0, y_1, \ldots, y_2k$. При этом в каждый момент выходной сигнал появляется только на одном из выходов $y_0, y_1, \ldots, y_2k$.

Счетчики входят в состав серии ИМС. Условное обозначение ИМС счетчиков состоит из обозначения серии (трех или четырех цифр), функционального назначения (двух букв ИЕ) и порядкового номера разработки (от одной до трех цифр).

Основные параметры интегральных микросхем счетчиков можно разделить на две группы: статические и динамические. К статическим параметрам относятся: входное напряжение высокого $U_{\text{вх}}^{1}$ и низкого $U_{\text{вх}}^{0}$ уровней, ток потребления от источника питания, напряжение $U_{\text{пит}}$, коэффициент разветвления $K_{\text{раз}}$ и модуль счета $K_{\text{с}}$. К динамическим параметрам триггеров относятся: время $t^{0,1}$ переключения из низкого уровня в высокий, время $t^{1,0}$ переключения из высокого уровня в низкий, максимальная частота $f_{\max}$. Большинство перечисленных параметров определяется серией микросхемы и типом применяемой логики. В табл. 5.5 приведены наименования и основные параметры некоторых ИМС счетчиков.

Таблица 5.5. Параметры интегральных микросхем счетчиков

Наименование счетчика	Тип логики	Функциональное назначение	Модуль счета $K_{\text{с}}$	Время задержки $t_{\text{зад}}$, нс	Потребляемая мощность $P_{\text{пот}}$, мВт
К155ИЕ5	ТТЛ	Асинхронный двоичный	16	135	265
К555ИЕ18	ТТЛШ	Синхронный двоичный	16	24/35	160
К155ИЕ7	ТТЛ	Реверсивный двоичный	16	24	510

К561ИЕ8	КМОП	Счетчик Джонсона с дешифратором	10	350	< 1
К561ИЕ11	КМОП	Двоичный реверсивный	16	400	< 1
К1533ИЕ11	ТТЛШ	Синхронный двоично-десятичный	10	16/24	105
К555ИЕ2	ТТЛШ	Двоично-десятичный	10	50	45
КР1533ИЕ5	ТТЛШ	Асинхронный двоичный	16	70	65

5.8. Логический синтез счетчиков

Логический синтез счетчиков так же, как и триггеров, основываются на применении теории конечных автоматов. Понятно, что с помощью таких методов можно синтезировать любые типы счетчиков, в том числе счетчиков с произвольным коэффициентом счета. Однако следует отметить, что методы синтеза наиболее эффективны в том случае, когда требуется счетчик, работающий в специальных кодах, либо кодирование состояний счетчика осуществляется произвольно. В этом случае рассмотренные выше схемотехнические способы построения счетчиков хотя и приемлемы, но требуют значительных временных затрат. Поскольку в наиболее общем случае целью всякого синтеза является не только разработка той или иной схемы, удовлетворяющей предъявленным к ней требованиям, но и определение в некотором смысле оптимальной ее структуры, то разработчик должен владеть как схемотехническими, так и логическими методами синтеза. Умелое применение обоих методов и их сопоставление позволяет выбрать оптимальный вариант синтезируемой схемы.

Наиболее широко известный логический синтез счетчиков основан на совместном решении прикладных уравнений счетчика с характеристическими уравнениями триггеров. Здесь используются, как правило, *RS-, T-, D-* и *JK-* триггеры, хотя принципиально возможно применение и других типов триггеров. В настоящей главе этот метод иллюстрируется на универсальных *JK-*триггерах по той основной причине, что они входят в состав практически всех отечественных и зарубежных серий ИМС. Синтез счетчика, как и триггеров, включает в себя выполнение следующих этапов:

1) определение числа элементов памяти (объем памяти автомата);

2) кодирование внутренних и выходных состояний автомата;

3) становление связи между входными, внутренними и выходными состояниями (описывается закон функционирования автомата);

4) составление обобщенной таблицы, в которую заносятся внутренние состояния счетчика и порядок их изменения, а также состояния, в которые должен устанавливаться счетчик под воздействием входных сигналов;

5) составление уравнений триггеров, их минимизация и составление схемы синтезируемого счетчика.

Объем памяти автомата вычисляется по формуле: $m = \left[\log_2 K_{\text{сч}}\right]^+$, где m – число триггеров, $K_{\text{Сч}}$ – коэффициент пересчета или количество внутренних состояний счетчика. Процедуру синтеза рассмотрим на конкретных примеров применительно к синхронным счетчикам.

Пример 1. Синтезировать счетчик с $K_{\text{Сч}} = 5$, работающий в двоичном коде.

Счетчик имеет 5 состояний, следовательно для его построения требуется три триггера $m_n = \left[\log_2 5\right]^+ = 3$. Обозначим эти триггеры через *A, B,* С. При поступлении входного импульса счетчик меняет свое состояние. Здесь индексами Q_A^t, Q_B^t, Q_C^t обозначенны сосотояния счетчика до прихода входного сигнала. $Q_A^{t+1}, Q_B^{t+1}, Q_C^{t+1}$ - состояния счетчика после прихода входного сигнала (выходные состояния). Порядок переключения состояний счетчика приведен в табл. 5.6.

В таком счетчике возможны три избыточных кода, которым соответствуют произвольные состояния триггеров, что отмечено знаком * в соответствующих строках столбцов Q^{t+1}.

Таблица 5.6. Таблица состояний синтезируемого счетчика

Состояние счетчика		Q_A^t	Q_B^t	Q_C^t	Q_A^{t+1}	Q_B^{t+1}	Q_C^{t+1}
Рабочие	1	0	0	0	0	0	1
	2	0	0	1	0	1	0
	3	0	1	0	0	1	1
	4	0	1	1	1	0	0
	5	1	0	0	0	0	0
Избыточные	6	1	0	1	*	*	*
	7	1	1	0	*	*	*
	8	1	1	1	*	*	*

На рис. 5.29, а) – в) приведены диаграммы Вейча полученных уравнений триггеров для выражений Q_A^{t+1}, Q_B^{t+1}, Q_C^{t+1}.

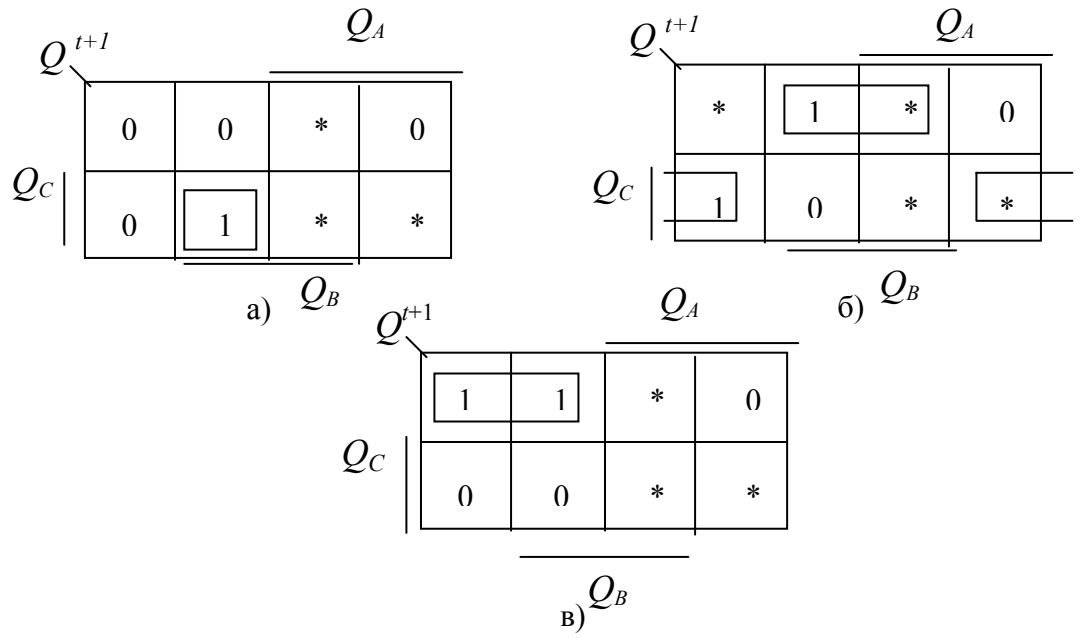


Рис. 5.29. Карты Карно счетчика с $K_{Сч}$=5:
а) для триггера *A;* б) для триггера *B;* в) для триггера *C*

В результате считывания получаем

$$Q_A^{t+1} = Q_B \cdot Q_C \cdot \overline{Q}_A; \quad Q_B^{t+1} = \overline{Q}_B \cdot Q_C \vee Q_B \cdot \overline{Q}_C \; ; \; Q_C^{t+1} = \overline{Q}_A \cdot \overline{Q}_C \cdot$$

Применительно к *JK*- триггеру запишем систему уравнений

$$J_A \cdot \overline{Q}_A \vee \overline{K}_A \cdot Q_A = Q_B \cdot Q_C \cdot \overline{Q}_A \vee Q_A \cdot 0 \; ;$$

$$J_B \cdot \overline{Q}_B \vee \overline{K}_B \cdot Q_B = \overline{Q}_B \cdot Q_C \vee Q_A \cdot \overline{Q}_C \; ;$$

$$J_C \cdot \overline{Q}_C \vee \overline{K}_C \cdot Q_C = \overline{Q}_C \cdot \overline{Q}_A \vee Q_C \cdot 0$$

и приравняв коэффициенты, найдем уравнения входов. Методом сравнения коэффициентов при Q и $\overline{Q}$ правых и левых частей полученных уравнений находим выражение для входов *J* и *K*

$$J_A = Q_B \cdot Q_C; \qquad \overline{K_A} = 0 \text{ или } K_A = 1;$$

$$J_B = Q_C; \qquad \overline{K}_B = \overline{Q}_C \text{ или } K_B = Q_C;$$

$$J_C = \overline{Q}_A; \qquad \overline{K}_C = 0 \text{ или } K_C = 1.$$

Схема, полученная в результате синтеза, приведена на рис. 5.30.

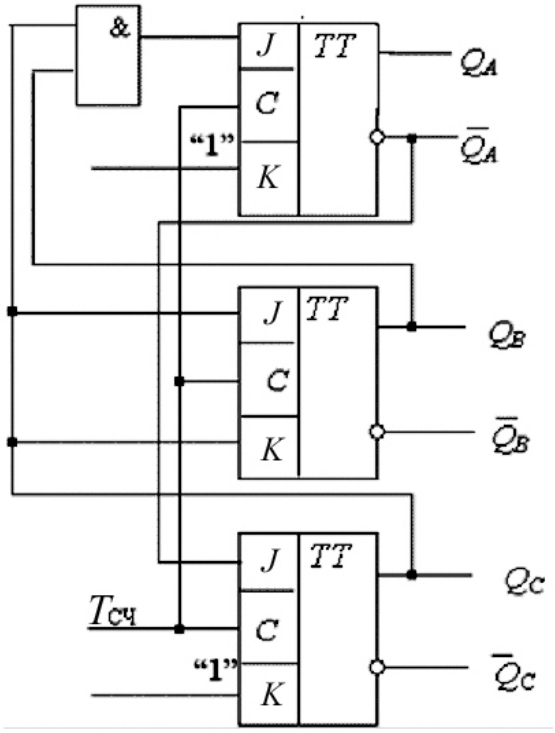


Рис. 5.30. Схема синтезируемого двоичного счетчика с $K_{СЧ} = 5$

Выполним синтез рассмотренного счетчика другим методом, с использованием матрицы переходов базового триггера. Суть метода синтеза в непосредственном нахождении уравнений входов триггеров.

Пример 2. Синтезировать счетчик с естественным порядком счета и $K_{СЧ} = 5$ с использованием матрицы переходов JK - триггера.

Вначале определяется требуемое число элементов памяти (триггеров)

$$m_n = \left[\log_2 5\right]^+ = 3 .$$

Затем составляется обобщенная таблица состояний счетчика (табл. 5.7), описывающая порядок переключения его состояний. Согласно выражению $2^n - K_{СЧ} = 3$ счетчик имеет три неиспользуемых (избыточных) состояния. Как следует из табл. 5.7, состояния сигналов Q_A^{t+1}, Q_B^{t+1}, Q_C^{t+1} на неиспользуемых комбинациях обозначены знаком $*$, т.е. они могут принимать произвольные значения. В таблицу входят столбцы с обозначением информационных входов *JK*- тригера.

После заполнения столбцов с индексами Q^t и Q^{t+1} заполняются столбцы, обозначающие названия информационных входов используемых триггеров. В свою очередь, их заполнение осуществляется на основе матрицы переходов JK- триггера (табл. 5.8).

Таблица 5.7. Обобщенная таблица состояний синтезируемого счетчика

Состояие счетчика		Q_A^t	Q_B^t	Q_C^t	Q_A^{t+1}	Q_B^{t+1}	Q_C^{t+1}	ТгA		ТгB		ТгC	
								J_A	K_A	J_B	K_B	J_C	K_C
Рабочие	1	0	0	0	0	0	1	0	*	0	*	1	*
	2	0	0	1	0	1	0	0	*	1	*	*	1
	3	0	1	0	0	1	1	0	*	*	0	1	*
	4	0	1	1	1	0	0	1	*	*	1	*	1
	5	1	0	0	0	0	0	*	1	0	*	0	*
Избыточные	6	1	0	1	*	*	*	*	*	*	*	*	*
	7	1	1	0	*	*	*	*	*	*	*	*	*
	8	1	1	1	*	*	*	*	*	*	*	*	*

Строка 1. Триггер до и после поступления входного сигнала находится в состоянии 0, т, е. $Q^t = Q^{t+1}$. Это условие может быть выполнено, когда на входе триггера J присутствует уровень 0, а на входе K либо 0, либо 1 (произвольное состояние, отмеченное знаком *). Именно эти символы заносятся в строку 1 столбцов J и K.

Таблица 5.8. Матрица переходов *JK*- триггера

Q^t	Q^{t+1}	J	K
0	0	0	*
0	1	1	*
1	0	*	1
1	1	*	0

Строка 2. Триггер должен перейти из состояния 0 в состояние 1. Это может быть выполнено при условии, когда на входе 1 обязательно действует уровень 1. При этом на входе K может быть уровень либо 0, либо 1 (т.е. знак *).

Строка 3. Триггер переключается из состояния 1 в состояние 0. Это условие выполняется при комбинации сигналов на входах триггера J = * и K = 1.

Строка 4. Триггер находится в состоянии 1 до и после прихода входного сигнала. Это условие может быть выполнено при комбинации сигналов J = 0 и

K = 0. Используя данные табл. 5.8, заполняем столбцы J_A, K_A , J_B , K_B, J_C , K_C обобщенной табл. 5.7.

Порядок заполнения этих столбцов рассмотрим на примере разряда C (триггер ТгС). Триггер ТгС до прихода импульса находится в состоянии 0, т.е. Q^t = 0 (строка 1, столбец Q^{t+1}). После поступления входного сигнала триггер ТгС должен переключиться в состояние 1, те. Q^{t+1}=1 (строка 1, столбец Q^{t+1}). Следовательно, триггер ТгС переходит из состояния 0 в состояние 1. В соответствии с табл. 5.8 для этого переходного состояния вход J = 1, а вход K =*. Именно эти символы заносятся в столбцы J_C и K_C табл. 5.7 (J_C =1, K_C =*) и т.д.

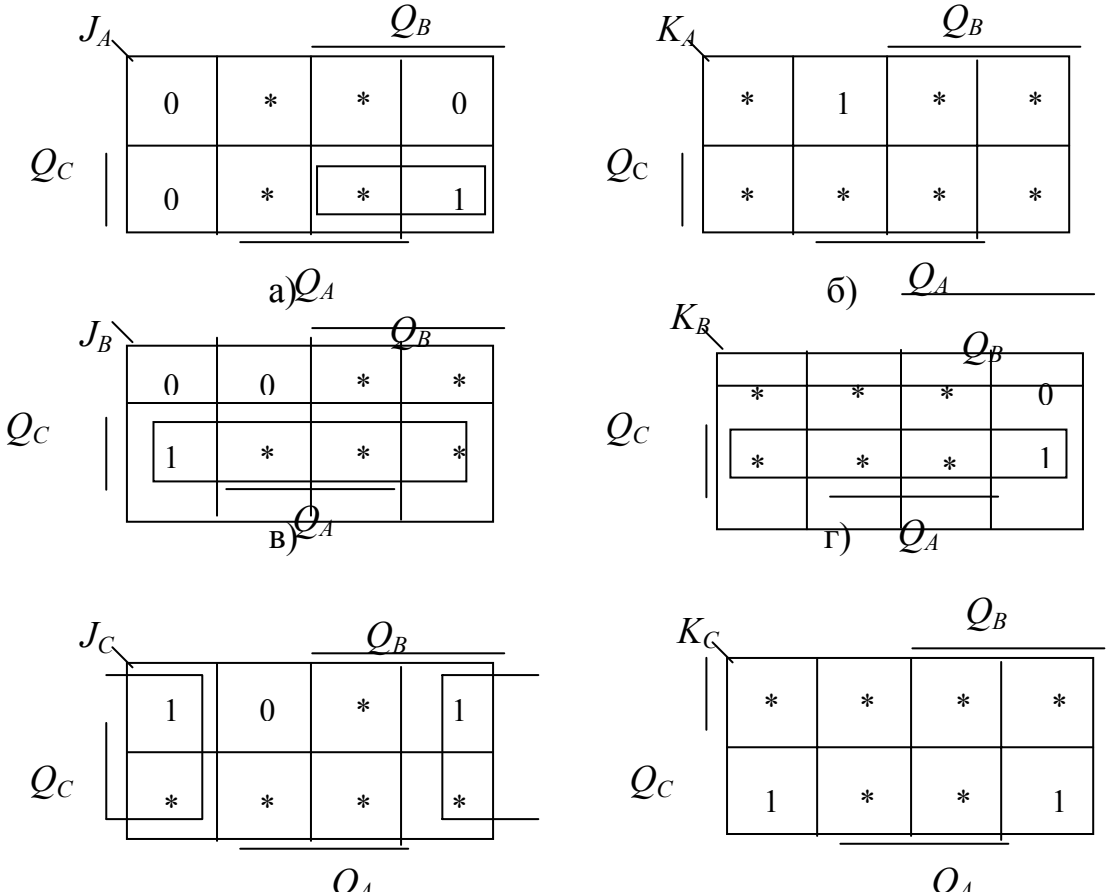
При заполнении столбцов входов J и K на неиспользуемых комбинациях (избыточные состояния) принято считать, что J и K на этих комбинациях принимают произвольные значения. Затем составляем диограммы Вейчадля определения входов J и K каждого триггера (рис. 5.31а - е). В результате считывания имеем следующие уравнения для входов J и K:

$$J_A = Q_B \cdot Q_C; \quad K_A = 1;$$

$$J_B = Q_C; \qquad K_B = Q_C;$$

$$J_C = \overline{Q}_A; \qquad K_C = 1.$$

Полученные уравнения полностью совпадают с выражениями входов J и K счетчика, синтезированного первым способом. В результате и схемы будут совпадать (см. Пример 1).



д) е)

Рис. 5.31. Диаграммы Вейча-Карно для счетчика на *JK* - триггерах:
для входов триггера *A* (а, б); для входов триггера *B* (в, г);
для входов триггера (д, е)

5.9. Вопросы и упражнения

1. Перечислите основные признаки классификации счетчиков.
2. Как можно установить в счетчике исходное состояние?
3. Чем отличается вычитающий счетчик от суммирующего?
4. Сколько триггеров необходимо для счета $K_{\text{Сч}} = 85$?
5. Чем отличается работа счетчика при подсчете числа импульсов и при делении частоты?
6. В чем состоят принципы последовательного, параллельного и группового переносов, их достоинства и недостатки?
7. Какие основные типы микросхем счетчиков Вы знаете? Перечислите их управляющие и информационные входы, выходы?
8. Чем отличаются двоичные и двоично-десятичные счетчики?
9. Определите максимальную частоту двоичного асинхронного восьмиразрядного счетчика, если $t_{\text{ЗД Р}}$ каждого триггера равна 10^{-7}с.
10. Нарисуйте схему асинхронного реверсивного счетчика с *К*сч = 5.
11. Нарисуйте схему двоично-десятичного счетчика с входами предварительной установки.
12. Нарисуйте схему делителя частоты с целочисленным переменным коэффициентом деления а) $N = 14$; б) $N = 18$; в) $N = 20$; г) $N = 14$.
13. Назовите способы построения недвоичных счетчиков.
14. Постройте схему двоичного счетчика с *К*сч = 6 методом логического синтеза КЦУ.
15. Синтезируйте схему счетчика ведущего счет в последовательности 0, 1, 3, 7, 9, 0.

ЛИТЕРАТУРА

1. Агаханян Т.М. Интегральные микросхемы. М.: Энергоатомиздат, 1983.

2. Алексенко А.Г., Шагурин И.И. Микорсхемотехника. М.: Радио и связь, 1990.

3. Березюк Н.Т., Андрушенко А.Г., Мошицкий С.С. и др. Кодирование информации (двоичные коды). Харьков: Высш.шк., 1978.

4. Браммер Ю. А., Пащук И.Н. Импульсные и цифровые устройства. М.: Высш.шк., 1999.

5. Букреев И.Н., Горячев В.И., Мансуров Б.М. Микроэлектронные схемы цифровых устройств. М.: Радио и связь, 1990.

6. Гольденберг Л.М. Импульсные устройства. М.: Радио и связь, 1984.

7. Евринов Э.В., Бутыльский Ю.Т., Мамзелев И.А. и др. Цифровая и вычислительная техника. Учебник для вузов. М.: Радио и связь 1991.

8. Калабеков Б.А. Цифровые устройства и микропроцессорные системы. М.: Горячая линия – Телеком, 2000.

9. Матюшин О.Т. Цифровые комбинационные устройства. М.: Моск. энерг. Ин-т, 1981.

10. Матюшин О.Т. Цифровые последовательные устройства. М.: Моск. энерг. Ин-т, 1982.

11. Нефедов А.В. Интегральные микросхемы и их зарубежные аналоги: Справочник. Т.10. М.: ИП РадиоСофт, 2000.

12. Новиков Ю.В. Основы цифровой схемотехники. Базовые элементы и схемы. Методы проектирования. М.: Мир, 2001.

13. Применение интегральных микросхем в электронной вычислительной технике: Справочник / Р.В.Данилов, С.А.Ельцова, Ю.П.Иванов и др.: Ред. Б.Н.Файзулаев, Б.В.Тарабрин. М. Радио и связь, 1987.

14. Прянишников В.А. Электроника: Курс лекций. Спб.: КОРОНА, 1998.

15. Пухальский Г.И. Проектирование дискретных устройств на интегральных микросхемах. М. Радио и связь, 1990.

16. Савельев А.Я. Арифметические и логические основы цифровых автоматов: Учебник. М.: Высш.шк., 1980.

17.Савельев А.Я. Прикладная теория цифровых автоматов: Учебник для вузов по спец. ЭВМ. М.: Высш.шк., 1987.

18. Угрюмов Е.П. Цифровая схемотехника. СПб.: БХВ – Санкт-Петербург, 2000.

19. Фридман А., Менон П. Теория и проектирование переключательных схем. М.: Мир, 1978.

20. Цифровые и аналоговые интегральные микросхемы: Справочник / С.В. Якубовский и др. М.: Радио и связь, 1989.

21. Шило В.Л. Популярные цифровые микросхемы: Справочник. М.: Радио и связь, 1987.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ....3
1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ЭЛЕКТРОНИКИ 4
1.1. Кодирование цифровой информации 4
1.2. Классификация цифровых устройств 9
1.3. Основы алгебры логики 14
1.4. Способы задания и преобразования логических функций 26
1.5. Минимизация логических функций 30
1.6. Вопросы и упражнения 39
2. КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА 41
2.1. Логический синтез комбинационных цифровых устройств 41
2.2. Элементная база для практической реализации цифровых устройств 43
2.3. Типы выходных каскадов цифровых устройств 48
2.4. Основные обозначения на схемах 57
2.5. Быстродействие КЦУ 61
2.6. Состязания в КЦУ 64
2.7. Надёжность КЦУ 67
2.8. Вопросы и упражнения 67
3. ТИПОВЫЕ КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА 69
3.1. Преобразователи кодов 69
3.2. Шифраторы и дешифраторы 71
3.3. Мультиплексоры и демультиплексоры 82
3.4. Компараторы 94
3.5. Арифметические устройства 98
3.6. Вопросы и упражнения 106
4. ТРИГГЕРЫ И РЕГИСТРЫ 107
4.1. Основные определения триггера 107
4.2. Классификация триггеров 108
4.3. Асинхронный *RS*-триггер с прямыми входами 110
4.4. Асинхронный *RS*- триггер с инверсными входами 113
4.5. Синхронный *RS*-триггер со статическим управлением 114
4.6. Универсальный *JK*-триггер 116
4.7. *T* – триггер 119
4.8. *D*-триггер 120
4.9. Синхронный триггер с динамическим управлением 121
4.10. Логический синтез триггерных устройств 123
4.11. Регистры 130
4.12. Вопросы и упражнения 138

5. СЧЕТЧИКИ 140

5.1. Общие сведения 139
5.2. Суммирующие двоичные счетчики 140
5.3. Вычитающие и реверсивные двоичные счетчики 146
5.4. Синхронные и асинхронные двоичные счетчики. 151
5.5. Десятичные счетчики 153
5.6. Делители частоты 158
5.7. Распределители 161
5.8. Логический синтез счетчиков 163
5.9. Вопросы и упражнения 169
ЛИТЕРАТУРА 170
ОГЛАВЛЕНИЕ 173

Ляля Вахитовна Ахметвалеева

ЦИФРОВЫЕ УСТРОЙСТВА
Учебное пособие по курсу
«Электронные промышленные устройства»
(Кафедра промышленной электроники)

Редактор издательского отдела Приклонская А.Г.

Изд.лиц № НД		Темплан КГЭУ 2002 г.
Подписано к печати		Формат 60×84/16
Гарнитура «Times»	Вид печати РОМ	Бумага «Business»
Физ.печ.л. 11	Усл.печ.л. 10,2	Уч.изд.л. 8,8
Тираж 2000	Заказ	

Издательский отдел КГЭУ
420066, Казань, Красносельская, 51

Типография КГЭУ
420066, Казань, Красносельская, 51