

Министерство образования Российской Федерации

Санкт-Петербургский государственный  
политехнический университет

**ПСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ**

*С.Н. Лехин*

**СХЕМОТЕХНИКА ЭВМ**

*ТЕОРЕТИЧЕСКИЙ КУРС*

*часть 2*

Учебное пособие

Санкт-Петербург

2004

УДК 681.3.06

**Рекомендовано к изданию  
Научно-методическим советом  
Псковского политехнического института СПбГПУ**

Рецензенты:

**-Веселков ЕЛ.**, доцент кафедры ЭСА ППИ СПбГПУ

**Ильин С.Н.**, зам. генерального директора АО «СКБ вычислительной техники»

**Схемотехника ЭВМ: Теоретический курс** (часть 2). Учебное пособие. Лехин С.Н.

-СПб/Псков, Изд. СПбГПУ, 2004 – 160 с.:ил.

Учебное пособие (часть 2) по дисциплине «**Схемотехника ЭВМ**» (СД. 03) предназначено для студентов Псковского политехнического института специальности 220100 «Электронно-вычислительные машины, комплексы, системы и сети» очной, очно-заочной и сокращенной форм обучения.

В пособии приведены основные положения логических основ цифровой техники, рассмотрены принципы функционирования цифровых узлов комбинационного типа, методы их синтеза, анализа, схемотехнической реализации, а также принципы построения формирователей и генераторов цифровых сигналов.

Учебное пособие может использоваться учащимися и студентами других специальностей и специализаций, интересующихся вопросами схемотехники узлов цифровых вычислительных машин.

© Псковский политехнический институт  
(филиал) Санкт-Петербургского  
государственного политехнического  
университета, 2004

© С.Н. Лехин

## 1. АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ

Для записей результатов количественных (числовых) измерений используются наборы символов, называемые цифрами. Способ, при котором числа отображаются комбинациями символов, называется системой счисления.

Одно и то же число допускает различные варианты представления. Числовые данные отличаются от остальных тем, что с ними могут производиться арифметические операции (сложение, умножение и т. п.). От выбранного способа записи чисел, то есть от системы счисления, зависит как наглядность их представления, так и сложность выполнения процедур обработки.

Обычно количество символов (цифр) для записи чисел ограничено, поэтому для больших чисел цифры в его записи могут и будут повторяться. Если значение цифры не зависит от ее местоположения в записи, то данная система счисления называется непозиционной. Примером такой системы является римская, с помощью которой размечаются циферблаты часов.

Здесь используются так называемые римские цифры I, V, X, L и другие. Первая соответствует единице, вторая пятерке, третья десятке, а четвертая - сотне. То есть запись XXVII обозначает  $10+10+5+1+1=27$ , запись XI соответствует одиннадцати и т. д. В принципе, применяя достаточное количество символов в этой системе можно записать любое число. Однако, в непозиционных системах счисления имеются определенные трудности при представлении больших чисел и, кроме того, существуют проблемы с выполнением арифметических операций.

Поэтому в цифровой технике используются позиционные системы счисления. Их особенность в том, что значение одной и той же цифры зависит от ее

местоположения в записи числа, то есть от позиции, в которой она располагается. Например, запись 222 в десятичной системе счисления, где используются десять цифр от 0 до 9, означает, что данное число содержит две сотни, два десятка и две единицы. Позиция, которую занимает цифра, называют разрядом. Таким образом, в рассматриваемой записи имеются разряды сотен, десятков и единиц. Числовое значение разряда называется его весом.

То же число 222 можно представить и как  $2 \cdot 100 + 2 \cdot 10 + 2 \cdot 1$ , откуда следует, что веса разрядов отличаются в десять раз. Если отношение соседних весов одинаково и равно  $P$ , то это число называется основанием системы счисления. Обычно в позиционных системах счисления значения весов определяются степенями ее основания и число 222 представляется как  $2 \cdot 10^2 + 2 \cdot 10^1 + 2 \cdot 10^0$ .

Номер разряда соответствует значению показателя степени, в которую требуется возвести основание системы счисления, чтобы получить величину его веса. Количество символов (цифр) для отображения чисел в позиционных системах счисления не может превышать величины основания, в противном случае возникает неоднозначность записи числа.

Веса системы счисления могут и не находиться в пропорциональных отношениях. В этом случае понятие основания не вводится, но запись произвольного числа  $x_{n-1} x_{n-2} \dots x_1 x_0$  может быть представлена следующим образом

$$x_{n-1}P_{n-1} + x_{n-2}P_{n-2} + \dots + x_1P_1 + x_0P_0 = \sum_{i=0}^{n-1} x_iP_i$$

где  $x_i$  – цифра  $i$ -того разряда, а  $P_i$  – его вес.

Если веса находятся в отношениях пропорциональных основанию  $P$ , то аналогичное выражение будет выглядеть так

$$x_{n-1}P^{n-1} + x_{n-2}P^{n-2} + \dots + x_1P^1 + x_0P^0 = \sum_{i=0}^{n-1} x_iP^i$$

Аналогичным образом представляются и числа меньшие единицы, для чего используются отрицательные степени основания. К примеру, число 12,34 можно записать как

$$1 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 4 \cdot 10^{-2}.$$

Основание системы счисления может быть произвольным, но обычно оно выбирается из требований удобства выполнения действий над числовыми данными в конкретной ситуации. В частности, когда требуется обрабатывать данные в цифровых вычислительных машинах, целесообразным является использование двоичной системы счисления.

Это связано с тем, что в цифровой технике используются простейшие элементы, обладающие двумя устойчивыми состояниями, одному из которых можно приписать нулевое значение, а другому – единичное. В двоичной системе основание равно двум и для отображения чисел имеется лишь две цифры 0 и 1. Таким образом, цифровой логический элемент может представлять один разряд двоичного числа.

Любое целое число в такой системе может быть представлено в виде

$$N = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_12^1 + a_02^0$$

Здесь веса разрядов равны соответствующим степеням двойки, то есть 1,2,4,8,16 и т.д.

При необходимости перевода числа из десятичной системы счисления в двоичную его требуется представить в виде суммы степеней основания, то есть двойки. Например:  $21 = 16 + 4 + 1 = 2^4 + 2^2 + 2^0$ . Однако, с учетом того, что в записи числа должны присутствовать все разряды, полученное соотношение требуется дополнить недостающими степенями основания системы счисления. Чтобы результат в целом

остался неизменным, эти разряды необходимо умножить на нуль, а остальные на единицу. В итоге запись числа 21 будет иметь вид  $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$ , а само число в двоичной системе счисления запишется следующим образом 1 0 1 0 1.

Существуют и иные подходы для перевода десятичных чисел в двоичную систему. Один из них основан на вычислении остатков от последовательного деления исходного числа на два. Если 21 поделить на два, то получится целое число 10 и остаток, равный 1. Он будет младшей значащей цифрой двоичного кода этого числа. Далее, полученное целое число опять делится на два, что дает 5 и 0 в остатке. Это вторая значащая цифра. Деление  $21/2=10$  ост.1 продолжают, пока получившееся целое число не станет равным единице, она же будет и цифрой старшего двоичного разряда. Данную процедуру удобно записывать в виде следующей цепочки действий.

Осуществить преобразование двоичного кода в десятичный можно, воспользовавшись ранее приведенной формулой. Например,  $10011_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 2 + 1 = 19_{10}$ .

Кроме двоичной и десятичной систем счисления достаточно часто используют восьмеричную и шестнадцатеричную. В первой для представления разрядов чисел применяют десятичные цифры от 0 до 7, а во второй требуется 16 символов для отображения цифр. В качестве первых десяти используют цифры 0,1,2...9, а остальные шесть обозначаются буквами A,B,C,D,E,F, которые соответствуют числам 10, 11, 12, 13, 14 и 15.

Правила перевода из одной системы счисления в другую аналогичны рассмотренным выше. Например:

$$123_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0 = 64 + 16 + 3 = 83_{10}$$

$$AF_{16} = 10 \cdot 16^2 + 15 \cdot 16^1 + 5 \cdot 16^0 = 10 \cdot 256 + 15 \cdot 16 + 5 = 2805_{10}$$

Так как основания восьмеричной и шестнадцатеричной систем счисления являются степенями двойки, то это существенно облегчает перевод чисел из этих систем в двоичную и обратно. Для перевода из восьмеричной системы в двоичную достаточно каждую цифру числа представить трехразрядным двоичным кодом (триадой) и полученные комбинации нулей и единиц выписать в ряд.

Например:  $123_8 = 1 (1_2) 2 (010_2) 3 (011_2) = 1 010 011_2$ . При необходимости перевода из шестнадцатеричной системы в двоичную требуется проделать аналогичную процедуру, но преобразуя каждую цифру шестнадцатеричного кода в четырехразрядную комбинацию (тетраду) двоичного

$$AF_{16} = A (1010_2) F (1111_2) 5 (0101_2) = 1010 1111 0101_2.$$

Также просто осуществляется и обратное преобразование, то есть трансформация из двоичного кода в восьмеричный и шестнадцатеричный. Для этого требуется двоичное число, начиная справа, разбить на триады либо тетрады и каждой из них поставить в соответствие значение цифры в соответствующем коде. Например:

$$10111100101_2 = 10 111 100 101 = 2745_8$$

$$10111100010_2 = 101 1110 0010 = 5E2_{16}$$

С преобразованием двоичного кода в десятичный и наоборот такой подход не может быть использован, так как отношение между основаниями этих систем счисления не кратно степени двойки.

Формализовать процедуру трансформации десятичного кода в код, аналогичный двоичному, можно преобразовывая каждую цифру десятичного числа в тетраду двоичного кода. Тогда трансформация, к примеру, числа  $175_{10}$  будет выглядеть следующим образом  $1 \rightarrow 0001$ ;  $7 \rightarrow 0111$ ;  $5 \rightarrow 0101$ , а его запись примет вид  $175_{10} = 0001 0111 0101$ .

Однако, это выражение будет существенно отличаться от представления числа  $175$  в двоичном коде, который выглядит следующим образом  $175_{10} = 10101111_2$ . Отличие возникает из-за того, что в предыдущем примере разряды в

пределах каждой тетрады имеют двоичные веса, меняющиеся от единицы до восьми и, кроме того, каждая из них имеет и свой десятичный вес. – 1,10,100. При таком подходе образуется новая система счисления, которая в данном конкретном случае называется двоично-десятичной и относится к позиционным системам счисления с двойным взвешиванием.

В ряде случаев использование такой системы оказывается более удобным, чем двоичной из-за упрощения процедуры преобразования чисел. Однако, выполнение арифметических операций в такой и аналогичных системах счисления является более сложной процедурой, чем в обычных позиционных.

Как уже отмечалось, в позиционных системах счисления, веса разрядов в принципе могут быть произвольными. При этом одна и та же последовательность чисел от 0 до 9 в различных системах, имеющих только две цифры 0 и 1, будет кодироваться по-разному в зависимости от соотношения весов разрядов. В таблице приведены различные, используемые на практике способы кодирования десятичных цифр.

Дес. цифра	Кодированное представление десятичной цифры					
	Код 8-4-2-1	Код 2-4-2-1	Код 7-4-2-1	Код с избытком. 3	Код 2 из5	Код Грея
0	0000	0000	0000	0011	11000	0000
1	0001	0001	0001	0100	01100	0001
2	0010	0010	0010	0101	00110	0011
3	0011	0011	0011	0110	00011	0010
4	0100	0100	0100	0111	10001	0110
5	0101	1011	0101	1000	10100	0111
6	0110	0110	0110	1001	01010	0101
7	0111	0111	1000	1010	00101	0100
8	1000	1110	1001	1011	10010	1100
9	1001	1111	1010	1100	01001	1101



В первом столбце представлены комбинации соответствующие коду 8-4-2-1. Название кода состоит из значений весовых коэффициентов соответствующих разрядов. Если веса имеют значения 2-4-2-1, то для ряда десятичных цифр кодовые комбинации будут выглядеть по иному. Код 7-4-2-1 интересен тем, что любая кодовая комбинация содержит не более двух единиц. Такое свойство кода полезно для выявления ошибок при передаче сигналов, так как, если в принятой тетраде окажется три единицы, это будет сигналом о возникновении ошибки вследствие того, что таких комбинаций в данном коде не существует.

Особенность кодов, представленных в трех последних столбцах таблицы в том, что они относятся к классу непозиционных или невзвешенных. В коде с избытком 3 сумма двоичных чисел в первой и последней строках, второй и предпоследней и т. д. всегда дает кодовую комбинацию 1 1 1 1. Сформировать этот код можно путем прибавления к соответствующей десятичной цифре тройки и преобразования полученного результата в двоичный код.

В коде 2 из 5 используются пятиразрядные комбинации нулей и единиц. Его особенность в том, что в любой кодовой комбинации содержится по две единицы.

Еще одна разновидность довольно часто применяемого кода – код Грея, который обладает тем свойством, что представления соседних чисел отличаются изменением состояния только одного из разрядов.

Все ранее рассмотренные способы кодирования основывались на предположении, что исходное число положительно. Однако, для удобства выполнения ряда арифметических операций в любой системе счисления можно ввести понятие отрицательных чисел, значения которых будут меньше нуля. В десятичной системе счисления для их записи используется знак  $-$ , а такое же по величине (по модулю) положительное число дополняется знаком  $+$ .

Если для представления чисел с целью обработки в цифровых вычислительных машинах используется двоичный код, то ввести аналогичные знаки не представляется возможным, так как любое число и символ должны быть выражены через комбинации нулей и единиц. Таким образом, и знак числа требуется отображать этими же символами.

При одном из подходов положительному знаку ставится в соответствие 0, а отрицательному 1 и знаковый разряд записывается левее самого старшего значащего. Таким образом, число +9 будет выглядеть как 0 1001, а -9 следующим образом 1 1001. Однако в этом случае требуется обязательно указывать, что старший разряд является знаковым, иначе первое число будет воспринято как 9, а второе как 25. В так называемом модифицированном коде под знак отводят два дополнительных разряда, положительный кодируется 00, а отрицательный 11.

Некоторые виды арифметических операций в цифровых вычислительных машинах удобнее производить, используя обратные и дополнительные двоичные коды чисел. Формально обратный код получается из двоичного (прямого) путем замены во всех разрядах нулей на единицы, а единиц на нули. Таким образом, если прямой двоичный код числа 9 выглядит как 1001, то обратный будет представлен комбинацией 0110.

Дополнительный код образуется путем арифметического прибавления единицы в младший разряд обратного кода числа. Отсюда следует, что дополнительный код 9 отобразится кодовой комбинацией 0111. Понятие дополнительного кода можно ввести для любой позиционной системы счисления. Он представляет собой число, дополняющее исходное до значения веса следующего по старшинству разряда системы счисления.

В рассмотренном примере для представления числа 9 используется четырехразрядная двоичная кодовая комбинация. Вес следующего, более старшего разряда в

двоичной системе будет  $2^4 = 16$ . То есть дополнительным кодом девятки будет число  $16 - 9 = 7$ , двоичный код которого имеет вид 0111. Если речь вести о десятичной системе счисления, то дополнительный код 9 должен определяться как  $10^2 - 9 = 91$ .

При математическом описании различных процессов вводится понятие переменной. Это некоторая независимая, величина, которая может принимать ряд значений в определенном диапазоне.

Множество значений переменной может быть как непрерывным, так и дискретным. В первом случае переменная принимает любое значение из области, в которой она определена, а во втором лишь множество конкретных. Примером переменной первого вида является температура. Она меняется непрерывно и принимает любое значение из соответствующего диапазона, причем соседние могут отличаться на бесконечно малую величину. Примером дискретной переменной может служить цена товара. Ее минимальные изменения кратны одной копейке, так как меньших денежных единиц нет.

Над переменными можно проводить определенные математические действия. Совокупность этих действий и правил их выполнения называется алгеброй соответствующих переменных. Значениям одной переменной могут быть поставлены в соответствие значения другой. Закон, определяющий это соответствие называется функцией.

В особую группу выделяются переменные, принимающие лишь два фиксированных значения. Например, если переменная описывает состояния переключателя, который, может находиться либо во включенном, либо в выключенном состояниях. Значению переменной для одного из них можно присвоить название «Вкл», а для другого «Выкл», либо обозначить их по иному «А» и «В», или 0 и 1,

учитывая в последнем случае, что это не цифры, а просто символы для описания состояния переменной.

Такие переменные, имеющие лишь два значения, называются логическими или Булевыми. Первое связано с тем, что они могут выступать как результат анализа логического рассуждения, который бывает истиной или ложью. Совокупность законов преобразования этих переменных и правил действий над ними называется Булевой алгеброй или алгеброй логики.

В обычной алгебре для двух переменных  $A$  и  $B$  существует три возможных отношения между их значениями.  $A$  может быть равно, больше или меньше  $B$ . В алгебре логики определено лишь отношение эквивалентности, то есть переменные здесь могут быть либо равны, либо не равны. Вопрос, какая из них больше, а какая меньше не имеет смысла.

Кроме того, для таких переменных определены три операции или действия: конъюнкция, дизъюнкция и инверсия.

Конъюнкция, иначе называется операцией логического умножения, или операцией «И». Она обозначается значком  $\wedge$ , либо точкой « $\cdot$ », которой в обычной алгебре соответствует умножение. Иногда эту точку не ставят.

Правило выполнения операции конъюнкции для двух логических переменных  $X1$  и  $X2$  может быть описано

$X1$	$X2$	$X1 \cdot X2$
0	0	0
0	1	0
1	0	0
1	1	1

словесно, а также представлено в виде такой таблицы. Результат логического умножения равен единице, только, если обе переменные одновременно принимают единичные значения. Данная таблица полностью совпадает с таблицей умножения чисел 0 и 1.

$X1$	$X2$	$X1 + X2$
0	0	0
0	1	1
1	0	1
1	1	1

Операция дизъюнкции называемая иначе операцией логического сложения, либо операцией «ИЛИ», обозначается

как значком  $V$ , так и «+». Данная операция над двумя переменными определяется таблицей.

Инверсия, называемая также операцией логического отрицания, либо операцией «НЕ», реализуется над одной переменной, преобразуя ее значение в противоположное. Эта операция обозначается в виде черточки сверху над соответствующей переменной.

$X$	$\bar{X}$
$0$	$1$
$1$	$0$

В отличие от алгебры обычных переменных в алгебре логики не существует операций умножения, деления, возведения в степень и никаких других, кроме рассмотренных выше. Это связано с тем, что логические переменные не принимают числовых значений, то есть не могут быть отрицательными, дробными и т.п.

$X+0=X$	$X \cdot 0=0$
$X+1=1$	$X \cdot 1=X$
$X+X=X$	$X \cdot X=X$
$X + \bar{X} = 1$	$X \cdot \bar{X} = 0$
$\bar{0} = 1, \bar{1} = 0$	$\bar{\bar{X}} = X$

В Булевой алгебре справедливы соотношения, которые можно проверить простым перебором значений переменных, воспользовавшись таблицами соответствующих операций.

Для алгебры логики, как и для обычной алгебры, определен ряд законов выполнения действий над переменными, в частности коммутативный, ассоциативный и дистрибутивный.

Первый, иначе называемый переместительным, записывается следующим образом  $X1+X2=X2+X1$ , и  $X1 \cdot X2=X2 \cdot X1$ . Из него вытекает, что при сложении и умножении логических переменных их можно менять местами. Второй, ассоциативный закон иначе называется сочетательным. Для трех переменных его можно представить как

$$X1+X2+X3=(X1+X2)+X3=X1+(X2+X3)$$

$$X1 \cdot X2 \cdot X3=(X1 \cdot X2) \cdot X3=X1 \cdot (X2 \cdot X3),$$

то есть при выполнении логических операций, переменные можно объединять в группы и выполнять соответствующие действия по очереди.

Дистрибутивный, или распределительный закон, устанавливает правила выполнения скобочных действий  $X1 \cdot (X2 + X3) = X1 \cdot X2 + X1 \cdot X3$ , или  $X1 \cdot X2 + X1 \cdot X3 = X1 \cdot (X2 + X3)$ .

Данные выражения представляют собой тождества, то есть они справедливы при любых значениях переменных.

К основным законам алгебры логики относятся и законы или правила де-Моргана, которые связывают операции логического сложения и умножения. Если в обычной алгебре умножение можно представить как многократное сложение, то логическое сложение может быть выражено через логическое умножение следующим образом

$$\overline{X1 + X2 + \dots + Xn} = \overline{X1} \cdot \overline{X2} \cdot \dots \cdot \overline{Xn} \quad \text{и} \quad \text{наоборот}$$

$$\overline{X1 \cdot X2 \cdot \dots \cdot Xn} = \overline{X1} + \overline{X2} + \dots + \overline{Xn}$$

то есть инверсия суммы логических переменных равна логическому произведению их инверсий, а инверсия произведения – сумме инверсий.

Если к обоим частям равенства применить одну и ту же процедуру, то оно не изменится. Отсюда следует, что проинвертировав обе части приведенных соотношений, правила де-Моргана можно представить в такой форме

$$X1 + X2 + \dots + Xn = \overline{\overline{X1} \cdot \overline{X2} \cdot \dots \cdot \overline{Xn}}$$

$$X1 \cdot X2 \cdot \dots \cdot Xn = \overline{\overline{X1} + \overline{X2} + \dots + \overline{Xn}} .$$

Как и в алгебре непрерывных переменных, в алгебре логики под функцией понимается некий закон, или правило по которому переменным из одного набора (множества) ставятся в соответствие переменные из другого набора (множества).

В обычной алгебре аргумент и функция могут принимать, целые и дробные, положительные, отрицательные значения, и количество функций от одного аргумента не ограничено. Например,  $Y=X$ ,  $Y=X^2$ ,  $Y=X^3$ ,  $Y=\sin X$ ,  $Y=\log X$  и т.п. В алгебре логики из-за того, что, как у переменной, так и у функции может быть только два значения, число последних конечно.

От одной логической переменной существует лишь четыре различных функции. Они задаются следующим образом. Каждому из значений переменной может быть произвольным образом поставлено в соответствие значение функции равное 0, либо 1. Если любому значению аргумента функция ( $F1$ ) ставит в соответствие 0, то она называется тождественный ноль.

Функция  $F2$  называется функцией тождества, так как ее значения совпадают со значениями аргумента. Значения функции  $F3$  противоположны, или инверсны по отношению к значениям аргумента. Последняя функция  $F4$  обоим значениям аргумента  $X$  ставит в соответствие единицы, и называется тождественная единица. Других видов функций от одной переменной нет.

Как и в обычной алгебре, в алгебре логики существуют функции от нескольких аргументов или переменных, причем количество функций  $N$  связано с числом переменных  $n$  соотношением  $N = 2^{2^n}$ . Если переменных две  $X1$  и  $X2$ , то их наборов будет четыре, а количество вариантов задания значений функций на этих наборах и, соответственно число самих несовпадающих функций 16. Они представлены в таблице.

$X1$	$X2$	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$F9$	$F10$	$F11$	$F12$	$F13$	$F14$	$F15$
0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	1	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1

Некоторые из функций, приведенных в таблице имеют собственные названия.  $F1$  называется функцией логического умножения, конъюнкцией, функцией И, а  $F7$  – функцией логического сложения, дизъюнкцией, либо функцией ИЛИ. Это объясняется тем, что значения данных функций

эквивалентны результатам выполнения соответствующих логических операций.

С учетом количества обрабатываемых переменных F1 часто называют функцией 2И, а F7 - 2ИЛИ. Алгебраическая (символьная) запись этих функций выглядит следующим образом  $F1=X1 \cdot X2$  и  $F7=X1+X2$ .

Функции F8 отличается от F7 тем, что нули заменены единицами и наоборот. То есть, каждое значение F7 проинвертировано. Поэтому функция F8 называется функцией ИЛИ-НЕ (2ИЛИ-НЕ) и ее связь с F7 можно отобразить таким образом  $F8 = \overline{F7}$ .

X1	X2	F1	F7	F8	F14	F6	F9
0	0	0	0	1	1	0	1
0	1	0	1	0	1	1	0
1	0	0	1	0	1	1	0
1	1	1	1	0	0	0	1

Аналогичная связь наблюдается и между функциями F14 и F1, вследствие чего F14 носит название функции И-НЕ (2И-НЕ) и

соответственно  $F14 = \overline{F1}$ . Отсюда следует, что количество логических функций не только ограничено, но они еще и определенным образом взаимосвязаны между собой.

Функция F6 называется функцией логической неравнозначности, а F9 – функцией логической равнозначности. Первая из них принимает единичное значение в случаях, когда аргументы X1 и X2 не равны, а вторая в противоположной ситуации.

В алгебре логики функции можно определить или задать как с помощью таблицы, отражающей связь значений аргументов и функции, так и в виде совокупности типовых логических операций, записанных как некоторая формула. Первый способ называется табличным, а второй аналитическим.

Вследствие наличия определенной связи между операциями конъюнкции и дизъюнкции, которая описывается



законами де-Моргана, одна и та же функция аналитически может быть представлена по-разному. Например  $f(X1, X2) = X1 \cdot X2$  можно представить и как  $\overline{\overline{X1} + \overline{X2}}$ . Функция  $f(X1, X2, X3) = X1 + X2 \cdot X3$  после тождественных преобразований может выглядеть следующим образом  $\overline{\overline{X1} + (\overline{X2} + \overline{X3})}$ , или  $\overline{\overline{X1} \cdot (\overline{X2} + \overline{X3})}$ , либо как  $\overline{\overline{X1} \cdot \overline{X2} + \overline{X1} \cdot \overline{X3}}$ . Возможен также вариант  $\overline{\overline{X1 + X2} + \overline{X1 + X3}}$ .

Используя правила де-Моргана, любую логическую функцию можно представить в двух разных, но эквивалентных формах: как сумму произведений логических переменных, или как произведение сумм. Если логическое выражение функции представляет собой сумму компонент, каждая из которых является простой конъюнкцией аргументов, то такая форма называется дизъюнктивной нормальной формой - ДНФ. Когда в выражение, описывающее функцию входят лишь произведения сумм прямых или инверсных значений аргументов, это соответствует второй, так называемой конъюнктивной нормальной форме или КНФ.

Примером ДНФ является запись

$$F1(X1, X2, X3) = X1 + X2 \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot X3 + \overline{X1} \cdot X3, \quad \text{а}$$

КНФ может выглядеть следующим образом

$$F2(X1, X2, X3) = (X1 + X2) \cdot X3 \cdot (\overline{X1} + X2 + \overline{X3}) \cdot (\overline{X1} + \overline{X3})$$

Некоторые выражения не подпадают под эти определения, например

$$F3(X1, X2, X3) = X1 + X2 \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot X3 + (\overline{\overline{X1} \cdot \overline{X3}}),$$

так как здесь последнее слагаемое не является простой конъюнкцией, то есть произведением соответствующих логических переменных. Однако после небольших преобразований его можно перевести в ДНФ такого вида  $F3(X1, X2, X3) = X1 + X2 \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot X3 + X1 + X3$ .

ДНФ и КНФ это две эквивалентные формы представления логических функций, которые, используя правила и законы алгебры логики, можно трансформировать

одна в другую. Для функции  $F1$  процедура преобразования будет следующей:

$$F1(X1, X2, X3) = X1 + X2 \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot X3 + \overline{X1} \cdot X3 = \overline{\overline{X1} \cdot (\overline{X2} \cdot \overline{X3})} \\ = \overline{\overline{X1} \cdot (\overline{X2} + X3)} \cdot (\overline{X1} + X2 + \overline{X3}) \cdot (X1 + \overline{X3})$$

Полученное выражение по определению не является КНФ. Однако, если проинвертировать обе части равенства, то КНФ получится для функции  $\overline{F1}$ .

После замены в КНФ логического умножения на сложение, запись  $F2$  примет вид,

$$F2(X1, X2, X3) = \overline{(X1 + X2)} + \overline{X3} + \overline{(X1 + X2 + X3)} + \overline{(X1 + X3)},$$

при котором функция оказывается представленной с использованием лишь двух операций – логического сложения (ИЛИ) и инверсии (НЕ). При замене сложения на умножение, получится соотношение, в которое войдут лишь операции логического умножения (И) и инверсии (НЕ).

Отсюда следует, что любая, сколь угодно сложная логическая функция представима, с помощью двух простейших - ИЛИ и НЕ, либо И и НЕ. Наборы функций, через которые можно выразить все остальные, называются базисом.

Следуя правилам алгебры логики, функцию как НЕ, так и ИЛИ можно представить, используя лишь одну операцию ИЛИ-НЕ. Действительно,  $\overline{X + X} = \overline{X}$ ,  $\overline{(X_1 + X_2)} = X_1 + X_2$

Таким образом, набор из двух функций ИЛИ и НЕ является избыточным, так как после соответствующих преобразований, любую функцию можно реализовать, используя лишь функцию ИЛИ-НЕ. Поэтому она является представительницей минимального базиса.

Аналогичные рассуждения можно провести и по поводу функции И-НЕ. Действительно  $\overline{X} \cdot \overline{X} = \overline{X}$ ,  $\overline{(X_1 \cdot X_2)} = X_1 + X_2$ , а следовательно и эта функция также может служить в качестве минимального базиса. Отсюда следует, что любую сколь угодно сложную функцию от произвольного

количества логических переменных можно представить используя только одну, причем любую из рассмотренных функций. Это обстоятельство в ряде случаев существенно облегчает построение устройств для обработки цифровых сигналов.

Кроме представления функций в форме ДНФ и КНФ существуют так называемые совершенная дизъюнктивная нормальная форма (СДНФ) и совершенная конъюнктивная нормальная форма (СКНФ). ДНФ функции называется совершенной, если в каждом ее слагаемом присутствуют все аргументы или их инверсии.

Функция  $F1(X1, X2) = X1 + \overline{X1} \cdot \overline{X2}$  не представлена в СДНФ, так как в первое слагаемое не входит переменная  $X2$ . А функция  $F2(X1, X2) = X1 \cdot X2 + \overline{X1} \cdot \overline{X2}$  записана в совершенной дизъюнктивной нормальной форме. Аналогичная ситуация справедлива и для конъюнктивных нормальных форм.

Любая функция, представленная в несовершенной форме, всегда может быть приведена к совершенной, причем единственным образом. В частности для функции  $F1(X1, X2)$  это делается умножением первого слагаемого на выражение вида  $X + \overline{X}$ . Так как оно равно единице, то умножение на нее ничего не изменит, но в итоге  $F1$  окажется представленной в виде СДНФ

$$F1(X1, X2) = X1 + \overline{X1} \cdot \overline{X2} = X1 \cdot (X2 + \overline{X2}) + \overline{X1} \cdot \overline{X2} = X1 \cdot X2 + X1 \cdot \overline{X2} + \overline{X1} \cdot \overline{X2}$$

Несмотря на то, что первый вариант функции выглядит проще, в ряде случаев представление в форме СДНФ является необходимым, и, кроме того, при алгебраическом описании функций, заданных в табличной форме, они автоматически приводятся к виду СДНФ.

Пусть некоторая функция  $Y$  от трех переменных представлена в виде таблицы. Полное количество ее значений определяется числом переменных  $n$  и будет равно  $2^n$  (в данном случае  $n=3$ ,  $2^3=8$ ). Для преобразования табличной

X1	X2	X3	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

формы представления функции в алгебраическую выбирается строка, где функция принимает единичное значение и записывается логическое произведение всех переменных.

Те из них, которые принимают единичные значения, вводятся в соответствующее произведение без инверсии, а равные нулю - с инверсией. Получившаяся при этом компонента называется конstituентой единицы.

Далее эти компоненты логически суммируют. В итоге выражение для функции будет иметь вид, соответствующий СДНФ

$Y = X1 \cdot \overline{X2} \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot X3 + X1 \cdot X2 \cdot X3$ . Проверка правильности полученного результата может быть произведена простым перебором значений переменных и вычислением функции. Первое слагаемое, а значит и вся функция обращается в единицу, когда  $X1=1$ ,  $X2=0$ ,  $X3=0$ . Поэтому  $X2$ ,  $X3$  и входят в него с инверсиями, так как только в таком случае  $X1 \cdot \overline{X2} \cdot \overline{X3} = 1 \cdot \overline{0} \cdot \overline{0} = 1 \cdot 1 \cdot 1 = 1$ .

Аналогичным образом выглядит ситуация для последних двух строк таблицы. На остальных наборах ни одна из компонент формулы в единицу не обращается, следовательно функция будет равна нулю.

Рассмотренную функцию можно представить и в конъюнктивной нормальной форме. В этом случае для каждого набора переменных, на котором она обращается в нуль, записывают логическую сумму всех переменных. Если значения переменных равны единице, то они должны входить

туда с инверсией, а если нулю – то в прямом виде. Полученные суммы называются конституентами нуля. Далее их логически перемножают. Для приведенной ранее функции  $Y$  запись в виде КНФ имеет вид, который одновременно представляет собой и СКНФ

$$Y = (X_1 + X_2 + X_3) \cdot (X_1 + X_2 + \bar{X}_3) \cdot (X_1 + X_2 + \bar{X}_3) \cdot (X_1 + \bar{X}_2 + \bar{X}_3) \cdot (\bar{X}_1 + X_2 + \bar{X}_3).$$

Такое представление абсолютно эквивалентно предыдущему, но сложнее по структуре из-за того, что на восьми наборах переменных функция лишь три раза обращается в единицу и пять раз принимает нулевое значение. При ее записи в виде СДНФ в выражении войдет три компонента, а в форме СКНФ – пять. Поэтому на практике часто используют ту форму представления, которая позволяет получить выражение минимальной сложности.

Однако, в ряде случаев форма представления функции не может быть выбрана произвольной. В такой ситуации для минимизации числа ее компонент используют следующий

X1	X2	X3	Y	$\bar{Y}$
0	0	0	1	0
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	0

подход. Если требуется представление в форме СДНФ, а количество единичных значений функции больше, чем нулевых, то вводят новую функцию, инверсную по отношению к исходной.

Единичных значений у нее будет меньше, чем нулевых, и СДНФ окажется проще. Для функции, представленной в данной таблице при

использовании обычного подхода получим

$$Y = \bar{X}_1 \cdot \bar{X}_2 \cdot \bar{X}_3 + \bar{X}_1 \cdot \bar{X}_2 \cdot X_3 + \bar{X}_1 \cdot X_2 \cdot X_3 + X_1 \cdot \bar{X}_2 \cdot \bar{X}_3 + X_1 \cdot X_2 \cdot \bar{X}_3 + X_1 \cdot X_2 \cdot X_3$$

а для инверсной функции выражение примет вид

$$\bar{Y} = X_1 \cdot X_2 \cdot \bar{X}_3 + X_1 \cdot \bar{X}_2 \cdot X_3.$$

Чтобы вернуться к исходной функции достаточно проинвертировать обе части данного равенства  $Y = \overline{X1} \cdot X2 \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot X3$ . Хотя это и не СДНФ, но данное выражение гораздо проще, предыдущего, что в ряде случаев важнее канонической формы представления.

Последнее связано с тем, что при разработке устройств, работающих с цифровыми сигналами, их функционирование описывают в виде совокупности логических функций, а затем реализуют с помощью электронных узлов. Чем компактнее удастся получить выражение для функции, тем проще аппаратно реализовать соответствующую ей процедуру обработки.

В алгебре логики применяются специальные процедуры минимизации логических функций, позволяющие в ряде случаев представить их с использованием минимального количества логических операций. Некоторые функции минимизировать, то есть сократить количество, входящих в их состав компонент нельзя, а для других это возможно и разными способами.

Пусть исходная функция представлена в ДНФ и имеет вид  $Y = X1 \cdot X2 \cdot X3 + X1 \cdot X2 \cdot \overline{X3} + X1 \cdot \overline{X2}$ . Для ее преобразования можно воспользоваться правилами и основными законами алгебры логики. Если из первых двух слагаемых за скобки вынести произведение  $X1 \cdot X2$ , то функция примет вид  $Y = X1 \cdot X2 \cdot (X3 + \overline{X3}) + X1 \cdot \overline{X2}$ . Так как сумма прямого и инверсного значений одной и той же переменной  $X3$  равна единице, а умножение на единицу оставляет результат неизменным, то

$$Y = X1 \cdot X2 \cdot (X3 + \overline{X3}) + X1 \cdot \overline{X2} = X1 \cdot X2 \cdot 1 + X1 \cdot \overline{X2} = X1 \cdot X2 + X1 \cdot \overline{X2}$$

В оставшемся выражении за скобки можно вынести  $X1$ , выражение в скобках опять будет равно единице и в итоге  $Y = X1 \cdot X2 + X1 \cdot \overline{X2} = X1 \cdot (X2 + \overline{X2}) = X1$ . То есть данная

конкретная функция от трех переменных, ранее содержащая три компоненты будет равна  $X1$ .

В ходе выполнения процедуры минимизации часть переменных исчезает. Это происходит при обработке пар слагаемых, отличающихся тем, что какая-либо переменная входит в одно из них в прямом, а в другое - в инверсном виде, причем все остальные компоненты слагаемых совпадают. В этом случае из двух слагаемых получается одно с уменьшенным на единицу количеством переменных.

Отсюда следует, что выражение  $X1 \cdot X2 \cdot X3 + X1 \cdot X2 \cdot \overline{X3}$  минимизировать можно, а  $X1 \cdot X2 \cdot X3 + X1 \cdot \overline{X2} \cdot \overline{X3}$  нет. Если за скобки вынести  $X1$ , то в выражении  $X2 \cdot X3 + \overline{X2} \cdot \overline{X3}$  число переменных не уменьшится.

Таким образом, для минимизации требуется просмотреть все компоненты, входящие в состав функции и попарно сгруппировать слагаемые, отличающиеся значениями лишь одной переменной. Затем вместо них записать выражение с уменьшенным на единицу числом переменных. Эта процедура может повторяться несколько раз. В итоге форма представления исходной функции будет содержать минимальное количество слагаемых и переменных.

Аналогичным образом проводится минимизация функций, представленных в конъюнктивной форме. Формализовать этот процесс можно используя так называемые карты Карно. Данный прием особенно удобен, если число аргументов логической функции не превышает четырех-пяти.

X1	X2	X3	F1
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Для применения карт Карно исходная функция должна быть представлена в совершенной дизъюнктивной нормальной форме, то

есть в виде сумм логических произведений, куда входят все комбинации переменных.

Пусть имеется таблица некоторой функции  $F1$  от трех переменных. В виде СДНФ функция содержит пять слагаемых и выглядит следующим образом. Ее можно минимизировать аналитически, так как в данном выражении существуют пары слагаемых, в которых  $F1 = \overline{X1} \cdot \overline{X2} \cdot \overline{X3} + \overline{X1} \cdot X2 \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot \overline{X3} + X1 \cdot \overline{X2} \cdot X3 + X1 \cdot X2 \cdot X3$  меняется значение лишь одной переменной. Это первое и третье, четвертое и пятое. Прделав необходимые действия, получим  $F1 = \overline{X2} \cdot \overline{X3} + \overline{X1} \cdot X2 \cdot \overline{X3} + X1 \cdot X3$

Однако для рассматриваемой функции процесс минимизации можно продолжить дальше. Если в исходном выражении рассмотреть первое и второе слагаемые, то можно сделать вывод, что обрабатывая их, удалось бы сократить переменную ( $X2$ ), но в преобразованном выражении первое слагаемое уже изменено и данная процедура формально не выполнима.

В то же время в соответствии с законами алгебры логики, в частности  $X+X=X$ , в любое выражение можно без изменения результата логически прибавлять любые имеющиеся там слагаемые. Следовательно, если в первоначальную форму представления функции прибавить  $\overline{X1} \cdot \overline{X2} \cdot \overline{X3}$ , то после обработки этой компоненты со вторым слагаемым, получится  $\overline{X1} \cdot \overline{X3}$ . Выражение для  $F1$  примет вид  $\overline{X2} \cdot \overline{X3} + \overline{X1} \cdot \overline{X3} + X1 \cdot X3$ , соответствующий минимальной дизъюнктивной форме представления рассматриваемой функции.

Объединение слагаемых с одновременным уменьшением числа входящих в их состав переменных, часто называется склеиванием. В целом аналитическая процедура минимизации оказывается достаточно длительной даже простых функций.

Карта Карно представляет собой таблицу, количество клеток или ячеек в которой  $K$  равно числу значений



принимается функцией, которое связано с количеством переменных  $n$  соотношением  $K = 2^n$ . Для функции от трех переменных их будет восемь. Ячейкам карты приписываются все возможные значения комбинаций аргументов. Совокупность аргументов в каждой комбинации разбивается на две группы. У функции  $F1$  в качестве одного из возможных вариантов разбиения в одну группу можно объединить  $X1, X2$ , и отдельно рассматривать  $X3$ , либо сгруппировать  $X1, X3$ , а  $X2$  представлять отдельно. Возможны и иные варианты.

Столбцы обозначаются комбинациями логических произведений прямых и инверсных значений соответствующих переменных группы. Для первого случая разбиения они будут такими  $X1X2, X1\overline{X2}, \overline{X1}X2, \overline{X1}\overline{X2}$ .

Комбинации аргументов, используемые в обозначении соседних столбцов должны отличаться лишь в одном разряде. То есть  $X1X2$  и  $X1\overline{X2}$  но не  $X1X2$  и  $\overline{X1}X2$ , так как здесь меняют значения сразу обе переменные. Верхнюю строку можно обозначить  $X3$ , а нижнюю  $\overline{X3}$ , однако возможен и вариант  $\overline{X3}, X3$ . В итоге таблица будет иметь следующий вид.

$F1$	$X1X2$	$X1\overline{X2}$	$\overline{X1}X2$	$\overline{X1}\overline{X2}$
$X3$				
$\overline{X3}$				

Далее карту Карно заполняют значениями функции, которые она принимает на соответствующих наборах переменных. Если аргумент в наборе равен единице, то в обозначение строки или столбца он входит без инверсии, а если нулю – то с инверсией. Таким образом, для рассматриваемой функции  $F1(0,0,0) = F1(\overline{X1}, \overline{X2}, \overline{X3}) = 1$ ,

$F1(0,0,1) = F1(\overline{X1}, \overline{X2}, X3) = 0$  и т.д. Процедура минимизации заключается в том, что

$F1$	$X1X2$	$X1\overline{X2}$	$\overline{X1}X2$	$\overline{X1}\overline{X2}$
$X3$	1	1	0	0
$\overline{X3}$	0	1	1	1

25

расположенные рядом единицы охватываются так называемыми контурами склейки, причем некоторые из единиц могут входить сразу в несколько контуров, а некоторые ни в один. Количество ячеек в контуре должно быть равно одному из чисел ряда 1, 2, 4, 8, 16... $2^k$ .

Из приведенной ранее таблицы следует, что функция  $F1$  принимает единичное значение когда  $X1=X2=X3=1$ , то есть на наборе  $X1X2X3$ , а также при  $X1=X3=1, X2=0$ . Таким образом, в выражение для функции будут входить компоненты  $X1X2X3 + X1\overline{X2}X3$  и при их склеивании исчезнет переменная  $X2$ .

Процедура минимизации с использованием карт Карно проводится следующим образом. Проверяются переменные в контурах склейки и если они меняют свое значение, то их не вносят в запись соответствующей компоненты функции. Рассмотрение верхнего контура дает произведение  $X1X3$ , так как  $X2$  меняет свое значение. Из следующего контура получится выражение  $\overline{X2}X3$ .

Оставшаяся единица соответствует комбинации переменных  $\overline{X1}X2\overline{X3}$ . Таким образом минимизированное выражение для функции будет иметь вид  $F1 = X1X3 + \overline{X2}X3 + \overline{X1}X2\overline{X3}$  совпадающий с полученным в ходе первого этапа ее минимизации аналитическим способом. Однако там добавление вспомогательной компоненты, которая не изменила значения функции, позволило устранить еще одну переменную и окончательное выражение получилось проще.

Аналогичный подход возможен и при использовании карт Карно. Для этого вводятся дополнительные контура,

$F1$	$x1x2$	$x1\overline{x2}$	$\overline{x1}\overline{x2}$	$\overline{x1}x2$
$x3$	1	1	0	0
$\overline{x3}$	0	1	1	1

охватывающие уже склеенные единицы. Если ввести такой контур для нижней строки то вместо  $\overline{X1}X2\overline{X3}$  получится  $\overline{X1}X3$  и функция примет вид,

$F1 = X1X3 + \overline{X1X3} + X1\overline{X2}$  полностью совпадающий с результатом аналитической минимизации.

Контра склейки можно выбрать и по другому. В этом случае выражение для функции станет таким  $F1 = X1X3 + X1\overline{X2} + \overline{X1X3}$ . Оно не совпадает с предыдущим, но также является минимальной

$F1$	$X1X2$	$X1\overline{X2}$	$\overline{X1X2}$	$\overline{X1X2}$
$X3$	1	1	0	0
$\overline{X3}$	0	1	1	1

дизъюнктивной формой представления той же функции. Отсюда следует, что минимальных форм может быть несколько.

Если, как показано дополнительной контуром уже склеенные единицы, то структура выражения

$F1 = X1X3 + X1\overline{X2} + \overline{X1X3} + \overline{X2X3}$ , описывающего функцию, усложнится, хотя сама функция не изменится. Поэтому число контуров обычно стремятся выбрать минимальным.

$F2$	$X1X2$	$X1\overline{X2}$	$\overline{X1X2}$	$\overline{X1X2}$
$X3$	1	1	0	0
$\overline{X3}$	1	1	0	1

Для некоторой функции  $F2$  картина распределения ее значений может выглядеть следующим образом. В этом случае формируются два горизонтальных, контура

склейки, а нижняя правая единица остается одна. Выражение для функции примет вид.  $F2 = X1X3 + X1\overline{X3} + \overline{X1X2X3}$ . В нем первые два слагаемых отличаются значением переменной  $X3$  и полученное соотношение можно аналитически минимизировать до  $F2 = X1 + \overline{X1X2X3}$ .

Эта же процедура реализуется и с использованием карты Карно, для чего потребуется образовать контур склейки, включающий в себя четыре рядом расположенных единицы и, проанализировав какие из аргументов не меняются, оставить только их в выражении для функции. В данном

$F2$	$x_1x_2$	$x_1\bar{x}_2$	$\bar{x}_1x_2$	$\bar{x}_1\bar{x}_2$
$x_3$	1	1	0	0
$\bar{x}_3$	1	1	0	1

случае неизменной остается лишь переменная  $X1$ , которая и войдет в окончательное выражение.

Как уже отмечалось, склеивать расположенные соответствующим образом

единицы допускается, если их количество кратно степени двойки, то есть 2,4,8,16 и т.д.

Если сравнить комбинации аргументов для крайних клеток одной строки карты Карно, то можно убедиться, что они также отличаются значением лишь одного из них. Для левой нижней клетки комбинация имеет вид  $x_1x_2\bar{x}_3$ , а для правой -  $\bar{x}_1x_2\bar{x}_3$ . Так как эти компоненты входят в выражение описывающее ДНФ функции, то в ходе дальнейшей минимизации должно остаться произведение  $x_2\bar{x}_3$ . Таким образом, форма представления рассматриваемой функции может быть дополнительно упрощена до вида  $F2 = X1 + X2\bar{X}3$ .

Карта Карно является как бы разверткой объемной фигуры, поэтому ее крайние клетки на самом деле располагаются рядом и комбинации соответствующих переменных отличаются значением лишь одной из них. Это позволяет вводить контура склейки, охватывающие и крайние группы ячеек. На карте такой контур условно представляется как разорванный. Его введение для приведенного примера трансформирует компоненту  $\bar{x}_1x_2\bar{x}_3$  в  $x_2\bar{x}_3$ .

Как уже отмечалось, рассмотренная картина

$F$	$x_1x_2$	$\bar{x}_1x_2$	$\bar{x}_1\bar{x}_2$	$x_1\bar{x}_2$
$\bar{x}_3$	1	1	0	1
$x_3$	1	0	0	1

чередования переменных для обозначения строк и столбцов не является единственно возможной. Альтернативный вариант карты Карно для рассматриваемой функции выглядит следующим образом. Единицы и нули здесь

будут располагаться по другому, иными станут контура

склейки, но результат минимизации окажется тем же  $F2 = X1 + X2X3$ .

Пусть имеется некоторая функция  $Y$  от четырех переменных и карта Карно для нее выглядит следующим образом. Здесь одним контуром можно охватить восемь единиц, находящихся в нижней части карты, а вторым четыре единицы в правом столбце.

Единицу, расположенную в ячейке  $X1\overline{X2}X3\overline{X4}$  можно объединить с одной из единиц нижней группы, образовав контур из двух клеток. Склеить эту пару с единицами правого крайнего столбца нельзя, так как они расположены не рядом и при переходе от одного столбца к другому меняются сразу две переменных.

Левую верхнюю единицу можно объединить с правой верхней, либо с левой нижней, однако целесообразнее создать контур из четырех единиц. Такая склейка краевых клеток возможна, так как карта Карно данной конфигурации формально представляет собой сферу, то есть у нее все края состыкованы и крайние четыре клетки на самом деле располагаются рядом. Отсюда следует, что минимизированное выражение для функции будет иметь вид  $Y = \overline{X3} + \overline{X1}X2 + X1X2\overline{X4} + X2X4$

Для функции от пяти переменных, получится карта с 32 клетками, а если переменных шесть, то карта Карно будет

$Y$	$x1x2$	$x1\overline{x2}$	$\overline{x1}\overline{x2}$	$\overline{x1}x2$
$x3x4$	1	0	0	1
$x3\overline{x4}$	0	1	0	1
$\overline{x3}\overline{x4}$	1	1	1	1
$\overline{x3}x4$	1	1	1	1

содержать 64 ячейки. При этом простота и наглядность рассмотренного способа минимизации теряются, и поэтому используются иные подходы.

В ряде случаев работа некоторых устройств с двумя возможными состояниями

описывается так называемыми неполностью заданными или недоопределенными функциями. Они отличаются от ранее рассмотренных тем, что некоторые наборы переменных не могут реализовываться.

Пусть для функции  $Z$  от трех переменных комбинации  $X1=1, X2=0, X3=1$  и  $X1=1, X2=1, X3=1$  никогда не появляются. В этом случае сказать о том, какие значения будут у функции на этих наборах переменных нельзя, так как она на них не задана (не определена). Формально в таблице определяющей функцию, это отмечается записью в соответствующие

X3	X2	X1	Z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	*
1	1	0	1
1	1	1	*

клетки каких-либо значков, к примеру звездочек

Такие функции также можно минимизировать, и в ряде случаев, получающееся при этом выражение оказывается проще, чем у полностью заданных.

Карта Карно для недоопределенных функций строится стандартным образом, в ее клетки записываются нули, единицы и звездочки. Как и для обычной функции, единицы можно охватить контурами склейки, их для приведенного примера будет два и в итоге получится выражение  $Z = \overline{X1}X3 + X2X3$ .

Однако из-за того, что функция не определена, на наборах, которые не могут реализоваться, ей в этом случае

Z	$x1x2$	$x1\overline{x2}$	$\overline{x1}x2$	$\overline{x1}\overline{x2}$
$x3$	*0	*0	1	1
$\overline{x3}$	0	1	1	0

можно приписать любое значение, то есть либо ноль, либо единицу. В принципе это ничего не меняет, так как данные наборы, а, следовательно, и приписанные им значения функции не реализуются.

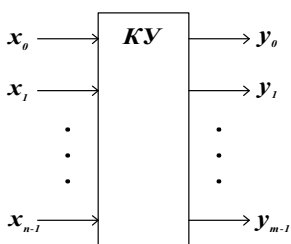
Отсюда следует, что для неполностью определенных

функций звездочки в клетках карты Карно можно, в зависимости от ситуации, заменять нулями, или единицами.

Если для рассматриваемой функции в клетки, помеченные звездочками, записать единицы, то получится два контура склейки из четырех клеток каждый и функция после минимизации примет вид  $Z = X3 + \overline{X}2$ .

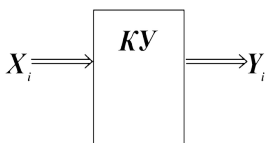
Обычно функция доопределяется таким образом, чтобы это было наиболее выгодно с точки зрения минимизации. Такая новая функция на заданных наборах переменных совпадает с исходной, а на запрещенных (не появляющихся) комбинациях ее значения не играют роли.

## 2. КЛАССИФИКАЦИЯ ЦИФРОВЫХ УСТРОЙСТВ.



Все цифровые устройства делятся на два класса: комбинационные и последовательностные. К первому относятся узлы, выходное состояние которых, является функцией лишь входного набора переменных. Если в  $i$ -тый момент времени на вход такого устройства поступает некоторая кодовая комбинация  $X_i = (x_0 x_1 x_2 \dots x_{n-1})_i$ , то его работу можно описать соотношением  $Y_i = f(X_i)$ , где  $Y_i = (y_0 y_1 y_2 \dots y_{m-1})_i$  - выходная кодовая комбинация или слово.

Структурную схему комбинационного узла можно представить следующим образом. Здесь каждому набору входных переменных будет соответствовать некоторая комбинация выходных, то есть с помощью комбинационного устройства устанавливается функциональная связь между ними, а вид реализуемой функции зависит от внутренней структуры устройства. В общем

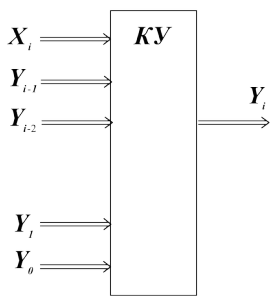


случае разным наборам входных переменных могут соответствовать одинаковые значения выходных.

Для анализа работы и синтеза комбинационных устройств используется стандартный аппарат алгебры логики, в том числе способы табличного и аналитического представления функций, карты Карно, диаграммы Вейча и т.п.

В отличие от комбинационных, выходное состояние последовательностных устройств в текущий  $i$ -тый момент времени зависит не только от входного воздействия  $X_i$ , но и от состояний в которых это устройство находилось в предшествующие этапы. Формально это можно записать следующим образом  $Y_i = f(X_i, Y_{i-1}, Y_{i-2}, \dots, Y_1, Y_0)$ . Такой узел как бы переносит на выходной сигнал всю предысторию (последовательность) входных воздействий, начиная с исходного состояния, то есть с момента  $i = 0$ .

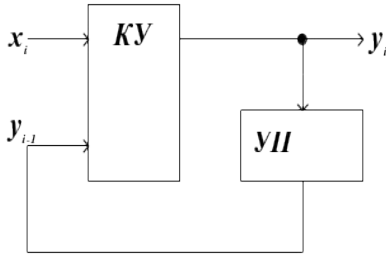
Так как такое устройство устанавливает некоторую функциональную связь между выходными сигналами  $Y_i$  и



набором воздействий  $X_i, Y_{i-1}, Y_{i-2}, \dots, Y_1, Y_0$ , в его состав должен входить некоторый комбинационный узел, который обрабатывает эти воздействия.

Однако, если  $X_i$  это входные сигналы, поступающие в  $i$ -тый момент времени извне, то  $Y_{i-1}$  и ему подобные - это выходные сигналы устройства, сформированные в предыдущие моменты времени. В комбинационном устройстве при формировании текущего значения  $Y_i$  предшествующее ему  $Y_{i-1}$  исчезает, а для последовательностного устройства оно требуется. Данная проблема решается запоминанием предшествующих состояний в некотором дополнительном устройстве (узле памяти), с выходов которого соответствующие сигналы



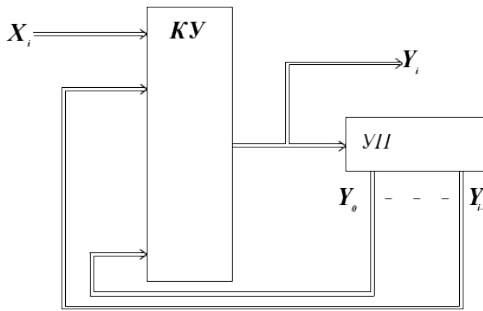


поступают на входы комбинационной схемы.

Отсюда следует, что структура последовательного узла должна иметь следующий

вид. Здесь сигналы  $Y_{i-1}, Y_{i-2} \dots Y_0$ , ранее сформированные на выходе подаются на вход устройства по цепям обратной связи. Так как ее воздействие сказывается через определенный временной интервал после поступления входного сигнала, то данный вид обратной связи называется запаздывающей.

Таким образом, отличительными признаками последовательных устройств являются наличие узла памяти и цепей обратной связи. Их количество и разрядность обрабатываемых сигналов могут быть различными.



В простейшем варианте на вход комбинационной схемы может подаваться одноразрядный внешний сигнал, и запоминаться лишь

предшествующее состояние, описываемое также одноразрядной двоичной функцией. Структура такого простейшего последовательного узла будет иметь следующий вид и оно относится к классу триггеров.

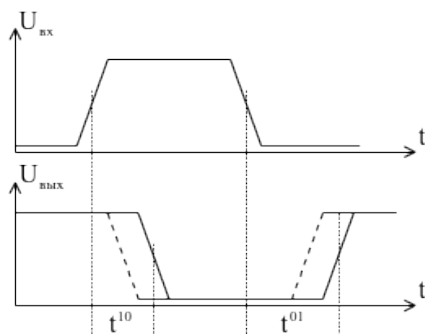
### 3. СОСТЯЗАНИЯ В КОМБИНАЦИОННЫХ СХЕМАХ.

Любой реальный логический элемент реагирует на комбинацию внешних воздействий не мгновенно, а с некоторой задержкой, обусловленной конечным быстродействием транзисторов и диодов, процессами перезаряда паразитных емкостей и, в ряде случаев, режимами работы и особенностями элементов схемы.

Временные диаграммы процесса переключения простейшего логического элемента – инвертора под действием сигнала с конечной длительностью фронтов можно представить следующим образом. Здесь  $t^{10}$  соответствует времени включения, а  $t^{01}$  – выключения. Их величины определяют максимальное значение частоты следования управляющих сигналов и быстродействие всей системы в целом.

Игнорирование задержек переключения при проектировании цифровых устройств, может привести либо к неустойчивой их работе, либо к полной неработоспособности, несмотря на правильно, с логической точки зрения спроектированную схему.

Значения этих времен зависят от большого количества факторов: напряжения питания, уровней управляющих сигналов, температуры, нагрузки, и могут сильно отличаться и для однотипных вентилях. В паспортных и справочных данных на цифровые микросхемы приводятся максимальные значения этих параметров для наихудшей комбинации влияющих факторов.



При проектировании устройств от которых требуется надежная работа в широком диапазоне внешних воздействий необходимо обладать информацией и о минимальных значениях времен переключения. Так

как они не нормируются, то при разработке цифровых устройств их приходится принимать равными нулю.

Уровень выходного сигнала вентиля в течение отрезка времени, когда фактическое состояние выхода не известно, называется состоянием неопределенности и обозначается \*. Этот сигнал, поступая на входы других логических элементов может приводить к появлению на их выходах либо определенных состояний 0, 1, либо также неопределенного. В этом

$x_1$	$x_0$	$y$
0	0	0
0	1	0
0	*	0
1	0	0
1	1	1
1	*	*
*	0	0
*	1	*
*	*	*

случае поведение логического элемента будет задаваться законами не двоичной, а троичной логики.

Из таблицы функционирования элемента 2И следует, что если значение одной из переменных, к примеру  $x_1$  равно нулю, то никакие вариации второй переменной  $x_0$  не вызовут изменений нулевого сигнала на его выходе, что можно записать как  $x_0 \cdot 0 = 0$ . В то же время при  $x_1 = 1$ , изменение значения  $x_0$  приведет к смене выходного состояния так как  $x_0 \cdot 1 = x_0$ . Считая, что  $x_0$  принимает любые значения - 0, 1, \*, соответствующая запись для этой ситуации выглядит следующим образом  $x_0 \cdot 1 = *$ . Аналогичные рассуждения

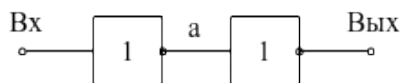
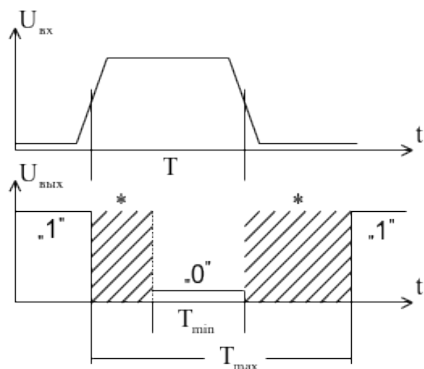
$x_1$	$x_0$	$y$
0	0	0
0	1	1
1	0	1
1	1	1

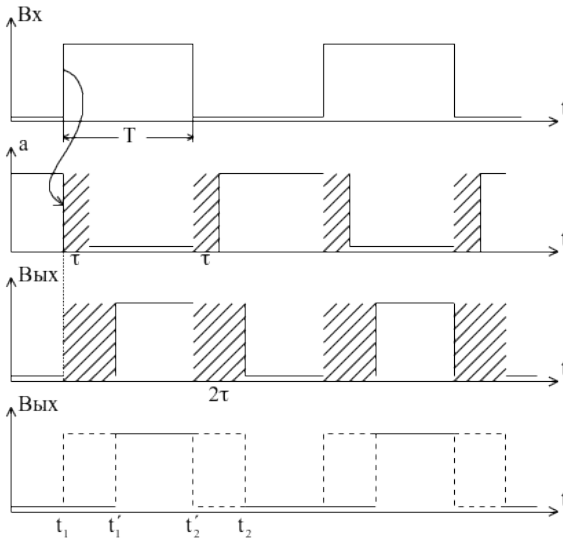
справедливы и по отношению к переменной  $x_0$ , таким образом,  $x_0 \cdot x_1 = *$ . Полная таблица истинности элемента 2И при таком подходе имеет вид.

Анализ аналогичных таблиц для элемента 2ИЛИ приводит к выводу, что  $x_0 + 1 = 1$ ,  $x_0 + 0 = *$  и  $x_0 + x_1 = *$ .

С учетом состояния неопределенности, процесс переключения инвертора можно представить следующим образом. В заштрихованных зонах состояния выходов могут быть как единичными, так и нулевыми, то есть длительность выходного сигнала логического нуля при фиксированной длительности единичного импульса на входе, может лежать в пределах от  $T_{\min} = T - t_{\max}^{10}$ , до  $T_{\max} = T + t_{\max}^{01}$ . Такой разброс может наблюдаться и у идентичных инверторов из разных партий и от разных производителей.

В цепочке из двух последовательно соединенных инверторов ситуация оказывается еще сложнее, так как соотношение их задержек переключения может быть произвольным. Временные диаграммы работы такой цепочки с учетом состояния неопределенности элементов имеют вид. Как следует из графика, отображающего состояние выхода второго инвертора, область неопределенности его состояний получается в два раза больше, чем у одного





Таким образом, выходной сигнал может оказаться как на  $2\tau$  короче, так и длиннее входного, причем предугадать ситуацию заранее невозможно. Если цепочка содержит  $k$  элементов, то зона неопределенности окажется равной

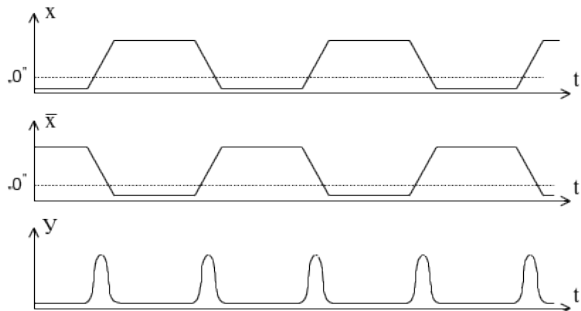
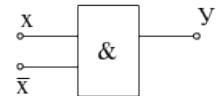
$k\tau$ .

При массовом производстве цифровых устройств эти эффекты требуется учитывать и проектировать схемы таким образом, чтобы разброс параметров элементов не влиял на конечный результат работы.

Разброс задержек срабатывания логических элементов, конечные фронты входных и выходных сигналов, наличие эффекта неопределенности могут приводить к тому, что во время переходных процессов логика работы устройства может нарушаться. Формально для него перестают действовать законы алгебры логики.

Данный эффект принято называть гонками или состязаниями.

Если, к примеру, на



входы элемента 2И подать противофазные сигналы с относительно пологими фронтами, то в определенные промежутки времени состояния обоих входов будут восприниматься как единичные.

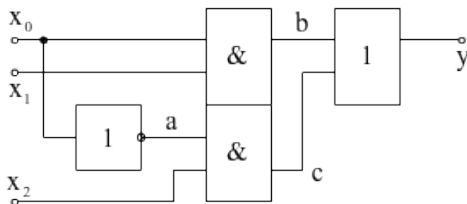
Это приведет к появлению на выходе инвертора коротких положительных импульсов, хотя по логике работы их не должно быть, так как  $x \cdot \bar{x} \equiv 0$ . Таким образом здесь возникает неалгоритмический переход – состязание.

$x_2$	$x_1$	$x_0$	$y$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Пусть требуется реализовать функцию, заданную следующей таблицей. Карта Карно для нее будет иметь вид, откуда

$y$	$x_0x_1$	$\bar{x}_0x_1$	$\bar{x}_0\bar{x}_1$	$x_0\bar{x}_1$
$x_2$	1	1	1	0
$\bar{x}_2$	1	0	0	0

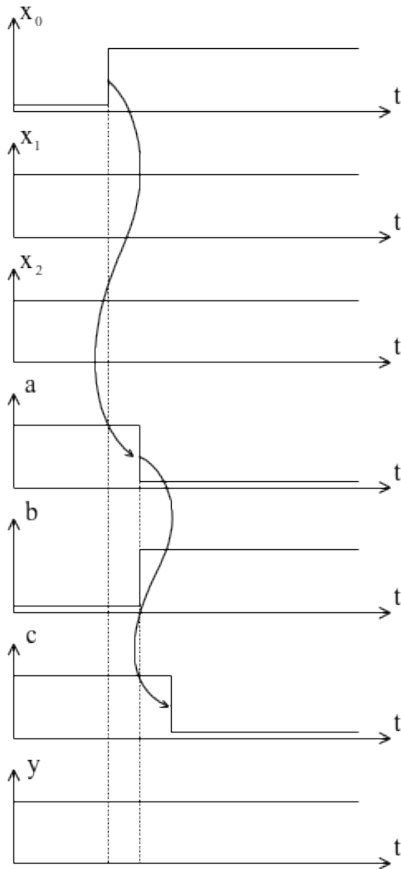
$y = x_0x_1 + \bar{x}_0\bar{x}_2$ , а схема устройства для ее реализации имеет такую структуру. Как видно из таблицы, смена комбинаций входных сигналов 1 1 0 на 1 1 1 и обратно



не должна вызывать изменения значения функции. Однако это справедливо лишь в предположении, что состязания в схеме отсутствуют, то есть

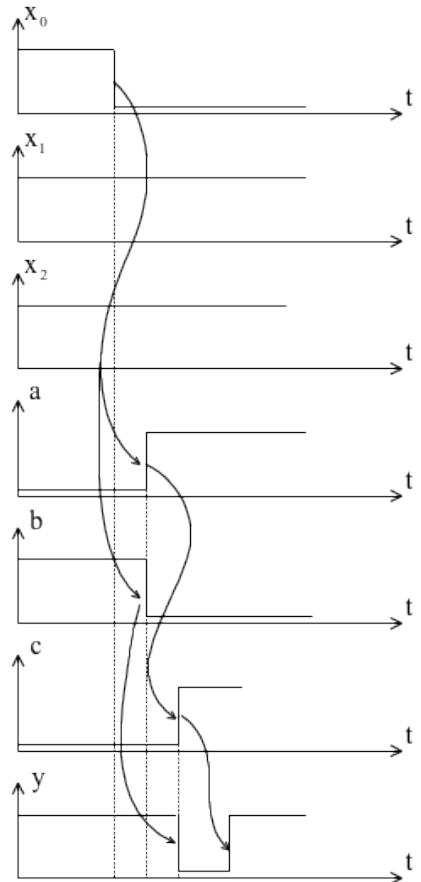
логические элементы обладают нулевыми задержками.

Если предположить, что у всех элементов, входящих в схему они одинаковы и равны  $\tau$ , то временные диаграммы для перехода 1 1 0 → 1 1 1 будут выглядеть следующим образом



Из них видно, что в данном конкретном случае наличие задержек не приводит к нарушению логики работы схемы, то есть состояния здесь отсутствуют.

Однако анализ ситуации для перехода  $1\ 1\ 1 \rightarrow 1\ 1\ 0$ , показывает, что в течение интервала длительностью  $2\tau$  на выходе схемы будет



присутствовать сигнал нулевого уровня, хотя теоретически функция своего единичного значения менять не должна. Таким образом, для этой ситуации состязания приведут к нарушению логики работы схемы, что может повлиять и на функционирование связанных с ней узлов.

Состязания классифицируются следующим образом. Они называются опасными, если на выходе устройства появляются переходы не предусмотренные алгоритмом его работы и приводящие к нарушению функционирования связанных с ним узлов. Если же их работа не нарушается даже при возникновении неалгоритмических переходов, то состязания относят к неопасным. Таким образом, в рассмотренной схеме однозначно классифицировать состязание невозможно, так как для этого требуется знать конфигурацию связанных с ней устройств.

В зависимости от места возникновения различают состязания на входах и состязания в логических элементах устройства. Состязания на входах появляются, если по условиям работы на входе комбинационного устройства одновременно меняются несколько переменных. Эта ситуация была рассмотрена на примере одновременного поступления на элемент 2И противофазных значений переменной  $x$ . Состязания в логических элементах появляются из-за конечных задержек их переключения и вследствие различных путей прохождения управляющих сигналов.

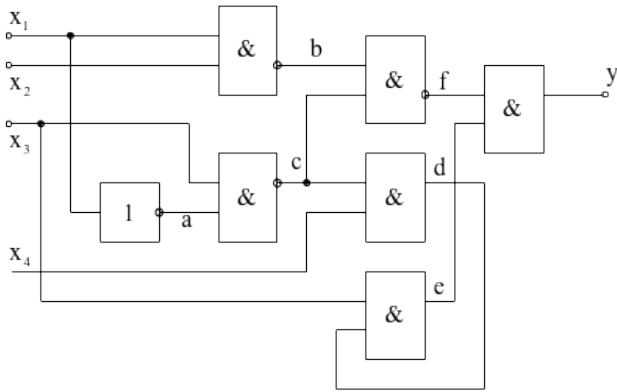
В зависимости от условий перехода комбинационного устройства от одного состояния к другому состязания делятся на статические и динамические. Они относятся к первой группе, когда для двух последовательных во времени значений входных сигналов, состояние выхода системы должно оставаться неизменным. Если же при смене входных воздействий состояние выхода должно измениться на противоположное, то состязания относятся к динамическим. Пример статических состязаний был рассмотрен выше, когда



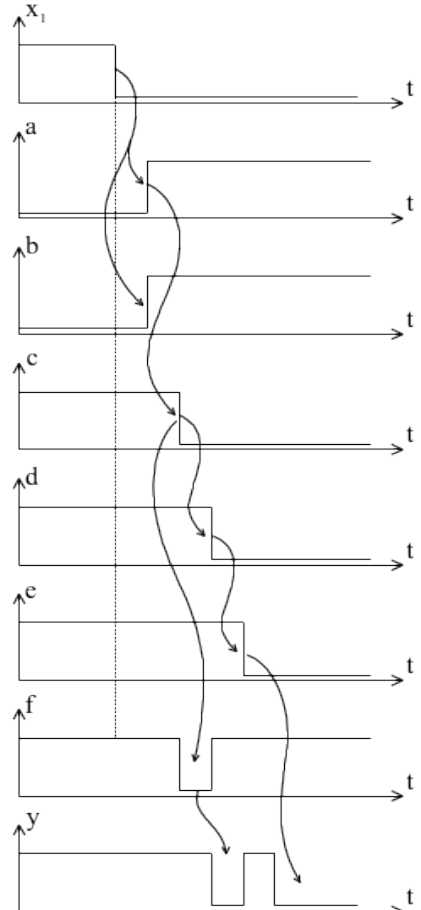
при переходе  $1\ 1\ 1 \rightarrow 1\ 1\ 0$ , на выходе схемы должен сохраняться уровень логической единицы.

Динамические состязания могут возникнуть в схеме, реализующей на двухвходовых элементах логическую функцию.

$$y = \left( \overline{\overline{x_1 x_2} \cdot \overline{x_1 x_3}} \right) \cdot \left( \left( \overline{x_1 x_3} \cdot x_4 \right) \cdot x_3 \right)$$

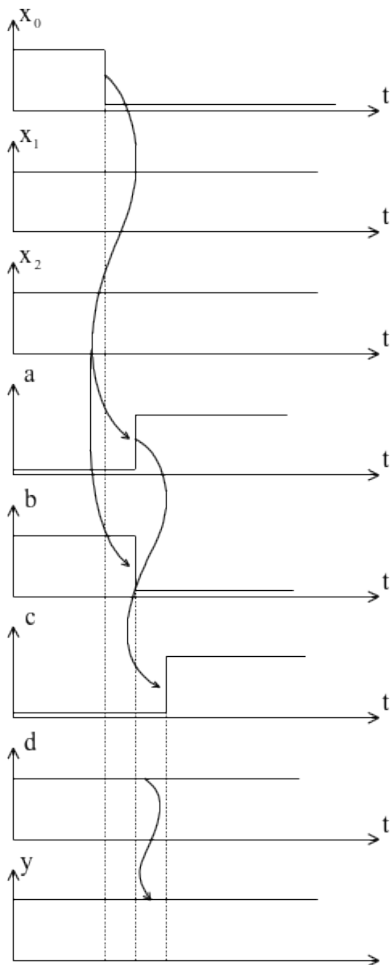


Если с единичного на нулевое меняется значение только одной переменной  $x_1$  то, как следует из временных диаграмм, на



выходе возникнет кратковременный положительный импульс и лишь затем система придет в требуемое состояние. Здесь вначале возникает правильный (алгоритмический) переход, а затем он сменяется неалгоритмическим.

Кроме этого состязания классифицируются как единичные (1 – состязания) и нулевые (0 – состязания). Первые соответствуют ситуации, когда при изменении входных сигналов на выходе устройства должно сохраняться нулевое значение сигнала, а в результате состязаний появляется 1. При 0 - состязаниях во время переходного процесса возникает нулевое состояние, при значениях входных сигналов формально поддерживающих единичный уровень на выходе.



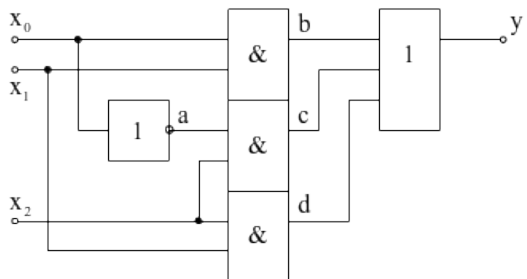
Анализ работы комбинационных схем по временным диаграммам с целью выявления опасных состязаний, особенно при больших количествах логических элементов в их составе оказывается достаточно трудоемким. Поэтому чаще используют аналитические подходы, либо методы, основанные на анализе ситуации с помощью карт Карно.

Для рассматриваемой ранее функции, заданной табличным образом, карта Карно имеет следующий вид. При стандартном подходе получаются два контура склейки и функция описывается следующим образом  $y = x_0x_1 + \overline{x_0x_2}$ .

Как уже отмечалось, здесь наблюдаются состязания при переходе от набора переменных 1 1 1 к 1 1 0. Это объясняется тем, что при  $x_1 = x_2 = 1$  и изменении  $x_0$  происходит переход из одного контура склейки в другой. В этом случае функция на время переходного периода может принять любое значение, хотя формально должно выполняться соотношение  $y = x_0 + \overline{x_0} \equiv 1$ .

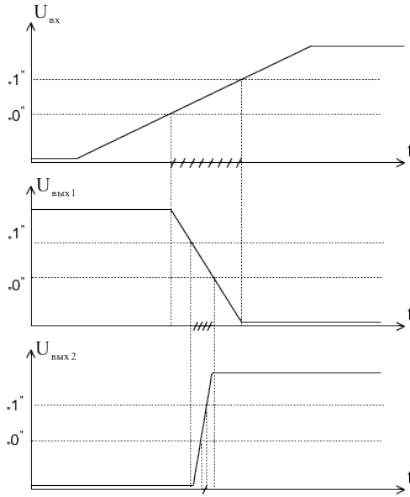
y	$x_0x_1$	$\overline{x_0}x_1$	$\overline{x_0}\overline{x_1}$	$x_0\overline{x_1}$
$x_2$	1	1	1	0
$\overline{x_2}$	1	0	0	0

Для того чтобы в данной ситуации избежать от состязаний, необходимо принудительно зафиксировать единичное значение функции. Это возможно при



введении дополнительных (связующих) контуров склейки (как показано штриховой линией), так как такой контур фиксирует единичное значение функции на соответствующих наборах переменных.

Вертикальный контур выполняет это условие при переходах  $1\ 1\ 1 \leftrightarrow 0\ 1\ 1$ , прежний горизонтальный контур для  $1\ 1\ 0 \leftrightarrow 1\ 0\ 0$ , а дополнительный для  $\overline{1\ 1\ 1} \leftrightarrow 1\ 1\ 0$ . Функция при этом будет иметь вид  $y = x_0x_1 + x_0x_2 + x_1x_2$  и для ситуации  $x_1 = x_2 = 1$ , она примет значение  $y = x_0 + \overline{x_0} + 1 \equiv 1$  независимо от состояния переменной  $x_0$ .



Для реализации такой функции потребуется более сложная схема. Анализ ее работы по временным диаграммам показывает отсутствие состязаний, присущих предыдущему варианту. Если контура в карте Карно не имеют общих границ, то статических состязаний не возникнет. Это связано с тем, что ни при каких комбинациях значений переменных не возникнет

состояние при котором функция примет вид  $y = x_i + x_i$ .

Наиболее эффективным способом борьбы с состязаниями является представление функции в неминимизированной форме. Для рассматриваемой ситуации она будет выглядеть следующим образом  
 $y = x_0x_1x_2 + x_0x_1x_2 + x_0x_1x_2 + x_0x_1x_2$ .

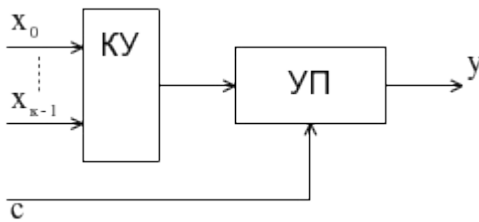
Одной из причин возникновения состязаний является неравенство задержек срабатывания логических элементов. Данная ситуация возникает, в частности, когда совместно используются элементы разных серий, к примеру ТТЛ и ТТЛШ, ТТЛ и КМОП и т. п.

Задержки в соответствующих цепях можно до определенного предела уменьшить пропуская управляющие сигналы с относительно длинными фронтами через цепочку последовательно соединенных инверторов, либо буферных элементов. Процессы, происходящие при этом представлены на временной диаграмме.

Так как переключение вентиля происходит при прохождении входным сигналом зоны напряжений "0" - "1", то время в течение которого второй вентиль изменит

выходное состояние будет меньше длительности фронта управляющего сигнала  $U_{BX}$ . Аналогичная ситуация справедлива и для последующих элементов. Однако, в любом случае длительность фронта формируемого сигнала и соответственно задержка в изменении состояния вентиля не смогут стать меньше минимальных значений присущих элементу данной конфигурации.

Выровнять задержки сигналов в соответствующих цепях можно и искусственно увеличивая их там, где они меньше, но этот путь требует индивидуального подбора элементов схем и



уменьшает  
быстродействие  
устройства в целом.

Если  
длительность ложного  
сигнала,  
обусловленного

состязаниями короче, чем интервал времени между моментами изменений управляющих сигналов, то для подавления помех можно использовать устройства, не пропускающие импульсы с длительностью меньше определенной.

Еще один способ борьбы с состязаниями состоит в подборе определенного порядка смены сигналов и состояний комбинационного устройства. Он основан на том, что опасные состязания проявляются в виде помех лишь при смене конкретных комбинаций сигналов на входах схемы. Поэтому в ряде случаев можно предусмотреть такой порядок их чередования, чтобы опасные состязания отсутствовали.

Один из путей реализации данного способа состоит в использовании специальных видов кодов, у которых при переходе к следующему значению меняется либо минимальное количество разрядов, либо эти изменения являются такими, что состязания в цепях разрядов частично

или полностью компенсируют друг друга. В качестве примера такого кода можно привести код Грея.

Универсальным способом борьбы с состязаниями является использование идеи синхронизации. При этом результат работы комбинационного устройства фиксируется в некотором запоминающем узле по сигналу синхронизации «С» когда все переходные процессы и состязания закончатся.

#### 4. КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА.

Процедура создания (синтеза) комбинационного цифрового устройства подразумевает разработку его принципиальной схемы и реализацию устройства на основе типовых логических элементов, работающих в том или ином базисе.

На первом этапе синтеза определяется общий алгоритм работы проектируемого устройства, то есть описывается функция, которую оно должно выполнять. На втором этапе этот алгоритм конкретизируется, при этом устанавливаются взаимосвязи между входными и выходными сигналами проектируемого устройства. Обычно их задают в табличной форме. На третьем этапе определяются функции, описывающие взаимосвязь входных и выходных сигналов. Они представляются в виде соответствующих комбинаций простейших логических операций. Четвертый этап заключается в том, что каждой простейшей операции, входящей в выражение для полученной функции, ставится в соответствие конкретный логический элемент и устанавливаются связи между ними.

На этом этапе логические элементы обычно идеализируются, то есть считается, что их задержки переключения равны нулю, нагрузочная способность неограниченна и т.п.

В то же время при создании реальных цифровых устройств, разработчик имеет дело с реальными узлами,

которым присущи определенные ограничения и особенности. Поэтому формально правильно спроектированная схема может на практике оказаться не работоспособной. Это вызывает необходимость после завершения процедуры синтеза, проводить анализ особенностей функционирования разработанной схемы с учетом параметров и характеристик реальных логических элементов, в ряде случаев оговаривая и их конкретные разновидности, то есть ТТЛ, КМОП и т.п.

#### 4.1 Дешифраторы.

Широкий класс комбинационных устройств представляют собой преобразователи кодов, к которым можно отнести все комбинационные устройства. Они ставят в соответствие коду входного слова определенное значение выходного кода, то есть устанавливают функциональную связь между входной и выходной переменными. Одной из разновидностей преобразователей кодов является дешифратор.

Обобщенный алгоритм его работы может быть определен следующим образом. При любой комбинации входных сигналов, выходной (в виде логического нуля или единицы) формируется лишь на одном из выходов дешифратора, причем номер этого выхода определяется самой кодовой комбинацией.

Так как посредством  $n$ -разрядного двоичного кода можно задать  $2^n$  комбинаций, то при  $n$  входах у дешифратора должно быть  $N = 2^n$  выходов. Дешифраторы такого типа называются полными. Если количество выходов  $N < 2^n$ , то такой дешифратор относится к неполным.

Для конкретизации алгоритма работы дешифратора

$x_1$	$x_0$	$y_3$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	1
0	1	0	1	0	0
1	0	1	0	0	0
1	1	0	0	1	0

требуется установить связи между наборами его входных и выходных сигналов. Это удобно делать табличным способом.



Таблица,  
описывающая

<i>Дес. предст. вх. кода</i>	$x_1$	$x_0$	$y_3$	$y_2$	$y_1$	$y_0$	<i>Номер акт. выхода</i>
0	0	0	0	0	0	1	0
1	0	1	0	0	1	0	1
2	1	0	0	1	0	0	2
3	1	1	1	0	0	0	3
<i>веса</i>	$2^1$	$2^0$					

$x_1$	$x_0$	$y_3$	$y_2$	$y_1$	$y_0$	
0	0	0	0	0	1	функционирование дешифратора с двумя управляющими входами и четырьмя выходами (дешифратора 2→4) имеет следующий вид. Здесь под выходным сигналом понимается наличие на соответствующем выходе
0	1	0	0	1	0	
1	0	0	1	0	0	
1	1	1	0	0	0	

логической единицы.

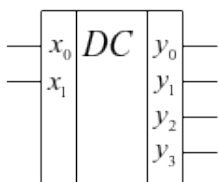
Из нее следует, что сигнал формируется только на одном из выходов и его номер однозначно связан с видом входной комбинации. Однако данная таблица не является единственной. Можно, в частности, задать алгоритм работы дешифратора и следующим образом. То есть существует несколько вариантов устройств, которые по определению относятся к классу дешифраторов.

Однако под дешифратором понимается лишь одно из них, работа которого описывается последней таблицей. Это связано с тем, что если входное слово представлять как двоичный код (присвоив разряду  $x_0$  вес  $2^0$ , а  $x_1$  -  $2^1$ ), то номер выхода на котором появится единичный сигнал будет соответствовать десятичному числу, определяемому этим кодом.

Выходной код, формируемый таким дешифратором называется унитарным кодом логических единиц.

Используя правила преобразования табличного представления функции в аналитическое, уравнения, связывающие значения входных переменных  $x_1, x_2$  с функциями, описывающими состояние каждого из выходов можно представить в таком виде. Отсюда вытекает, что в состав дешифратора  $2 \rightarrow 4$  должны входить четыре двухвходовых элемента И, и два инвертора, связанные между собой следующим образом.

Дешифратор, как функциональный элемент отображается на принципиальных схемах в виде прямоугольника, в центре которого помещается символическое буквенное обозначение DC, соответствующее функциональной принадлежности элемента. В левой области располагают символы, обозначающие



функции входов, а в правой – выходов. В ряде случаев буквенные

$$y_0 = \overline{x_0} \overline{x_1} \overline{x_3}$$

$$y_1 = \overline{x_0} \overline{x_1} x_3$$

$$y_2 = \overline{x_0} x_1 \overline{x_3}$$

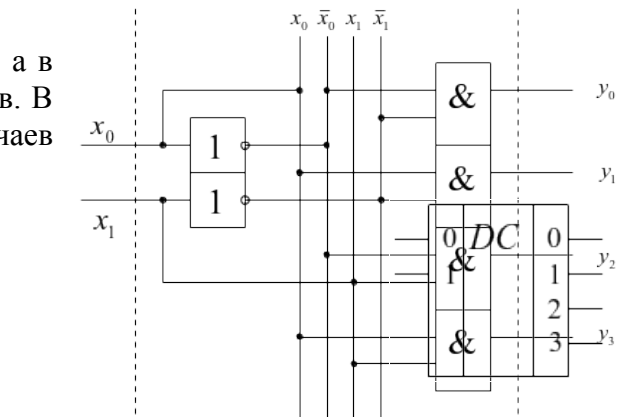
$$y_3 = \overline{x_0} x_1 x_3$$

$$y_4 = \overline{x_0} x_1 \overline{x_3}$$

$$y_5 = \overline{x_0} x_1 x_3$$

$$y_6 = \overline{x_0} \overline{x_1} \overline{x_3}$$

$$y_7 = \overline{x_0} \overline{x_1} x_3$$

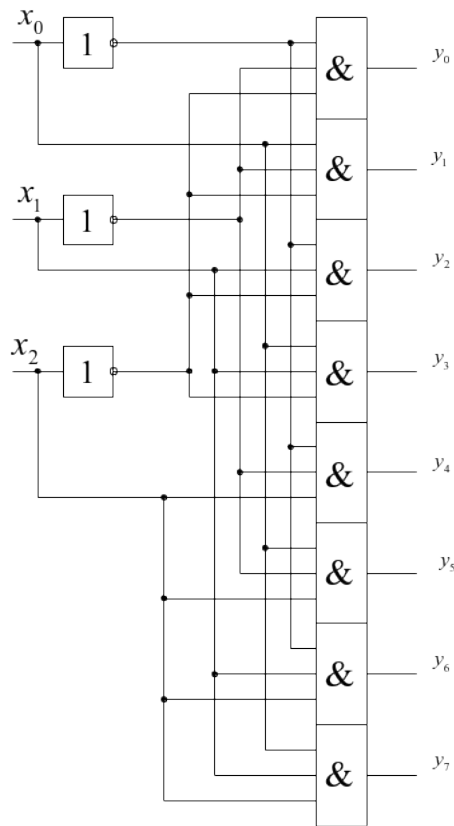


обозначения могут опускаться.

Аналогичным образом, можно синтезировать дешифратор  $3 \rightarrow 8$ . Состояния его выходов описываются таким набором функций, а схема выглядит следующим

образом. Дешифраторы данной структуры относятся к классу линейных, так как все элементы, формирующие выходные сигналы оказываются как бы выстроенными в одну линию.

Между входными и выходными сигналами дешифратора



имеется достаточно простая взаимосвязь, и для дешифратора с  $n$  входами ее можно выразить общей

$$\text{формулой } y_j = \bigwedge_{i=0}^{n-1} ((x_i a_{ij}) \vee (\bar{x}_i \bar{a}_{ij}))$$

Здесь  $i$  и  $j$  – это номера входов и выходов дешифратора  $a_{ij}$  – коэффициенты разложения номера соответствующего выхода в двоичный код.

Определить функцию, описывающую к примеру, состояние десятого выхода дешифратора  $4 \rightarrow 16$  можно следующим образом. Здесь  $n=4$ ,  $i$  меняется от нуля до 3, а  $j$  – от нуля до 15. Для десятого выхода представление его номера в двоичном коде будет выглядеть как  $10_{10} = 1\ 0\ 1\ 0_2$  и коэффициенты  $a_{ij}$

окажутся равными:  $a_{0,10} = 0, a_{1,10} = 1, a_{2,10} = 0, a_{3,10} = 1$ . То есть функцию  $y_{10}$  можно определить

$$\begin{aligned} \text{следующим образом: } y_{10} &= \bigwedge_{i=0}^3 ((x_i \cdot a_{i,10}) \vee (\bar{x}_i \cdot \bar{a}_{i,10})) = \\ &= [(x_0 \cdot 0) + (\bar{x}_0 \cdot 1)][(x_1 \cdot 1) + (\bar{x}_1 \cdot 0)][(x_2 \cdot 0) + (\bar{x}_2 \cdot 1)][(x_3 \cdot 1) + (\bar{x}_3 \cdot 0)] = \\ &= \bar{x}_0 x_1 \bar{x}_2 x_3. \end{aligned}$$

Как уже отмечалось, вследствие того, что цифровые устройства строятся на реальных логических элементах, параметры которых имеют определенные ограничения, после завершающего этапа синтеза требуется провести анализ характеристик разработанного узла.

Одними из важных показателей любого цифрового устройства являются его быстродействие, энергопотребление и нагрузка на внешние линии по которым поступают управляющие сигналы.

Если принять, что средняя задержка переключения логического элемента равна  $\tau$ , то для дешифратора линейной структуры любой разрядности время между изменением входного сигнала и появлением выходного не превысит  $2\tau$ , так как через  $\tau$  сформируются инверсные значения входных переменных и еще через  $\tau$  выходные сигналы.

Средняя мощность потребления определяется количеством логических элементов, так как мощность потребляемая каждым из них  $P_0$  практически не зависит от числа входов, то есть количества обрабатываемых переменных.

Одной из серьезных проблем в многоразрядных дешифраторах и других сложных цифровых устройствах может стать конечный коэффициент разветвления реальных логических элементов и количество входов, подключаемых к линиям, по которым подаются управляющие сигналы (коэффициент нагрузки). Кроме того, при реализации цифровых устройств на реальных микросхемах малой степени интеграции важным является как количество используемых при этом логических элементов, так и число корпусов. Первый фактор влияет на потребляемую мощность, а второй на габаритные размеры реального узла.

Если рассмотреть линейный дешифратор  $4 \rightarrow 16$ , то для его создания понадобится 16 четырехвыходовых элементов И и четыре инвертора (элемента НЕ). В составе ТТЛ, ТТЛШ и КМОП серий выпускаются микросхемы содержащие в одном

корпусе по два элемента 4И (К555ЛИ6, КР1561ЛИ2) и по шесть элементов НЕ (К155ЛН1, К561ЛН2). Таким образом, для реализации дешифратора 4→16 понадобится 9 (8+1) корпусов микросхем, причем 2 инвертора окажутся невостребованными.

Как уже отмечалось, максимальная задержка в таком дешифраторе составит  $2\tau$ , потребляемая мощность будет пропорциональна количеству логических элементов и при среднем потреблении каждым из них  $P_0$ , составит  $20P_0$ . Количество входов логических элементов, подключенных к линиям управляющих сигналов, будет равно 9, так как каждый из сигналов поступает на один инвертор и восемь логических элементов. Любой инвертор оказываются нагруженными на восемь входов схем 4И. То есть в данном дешифраторе коэффициенты нагрузки и разветвления не превышают допустимых пределов.

Однако, в дешифраторе линейной структуры 8→256, каждая из линий управляющих сигналов должна обеспечивать работу 129 входов логических элементов, а инвертор – 128. Так как коэффициент разветвления стандартного ТТЛ вентиля равен 10, то для решения данной проблемы потребуются либо их объединение, либо использование сложных буферных устройств. Требуемое количество корпусов для создания такого дешифратора равно  $256+2$ , так как элемент 8И занимает корпус и дополнительно потребуются 8 инверторов, размещенных в двух корпусах. С ростом разрядности данные проблемы будут усугубляться. Это вызвало необходимость использования других подходов к построению дешифраторов.

Логические функции, описывающие состояния выходов линейного дешифратора  $4 \rightarrow 16$  выглядят следующим образом. В каждую из них входит одна из компонент  $\overline{X_0 X_1}, X_0 \overline{X_1}, \overline{X_0} X_1, X_0 X_1$ .

$$\begin{aligned}
 Y_0 &= \overline{X_3} \overline{X_2} \overline{X_1} \overline{X_0} \\
 Y_1 &= \overline{X_3} \overline{X_2} \overline{X_1} X_0 \\
 Y_2 &= \overline{X_3} \overline{X_2} X_1 \overline{X_0} \\
 Y_3 &= \overline{X_3} \overline{X_2} X_1 X_0 \\
 Y_4 &= \overline{X_3} X_2 \overline{X_1} \overline{X_0} \\
 Y_5 &= \overline{X_3} X_2 \overline{X_1} X_0 \\
 Y_6 &= \overline{X_3} X_2 X_1 \overline{X_0} \\
 Y_7 &= \overline{X_3} X_2 X_1 X_0 \\
 Y_8 &= X_3 \overline{X_2} \overline{X_1} \overline{X_0} \\
 Y_9 &= X_3 \overline{X_2} \overline{X_1} X_0 \\
 Y_{10} &= X_3 \overline{X_2} X_1 \overline{X_0} \\
 Y_{11} &= X_3 \overline{X_2} X_1 X_0 \\
 Y_{12} &= X_3 X_2 \overline{X_1} \overline{X_0} \\
 Y_{13} &= X_3 X_2 \overline{X_1} X_0 \\
 Y_{14} &= X_3 X_2 X_1 \overline{X_0} \\
 Y_{15} &= X_3 X_2 X_1 X_0
 \end{aligned}$$

Их можно представить как новые логические функции  $K_i$ . Тогда исходные выражения примут вид.

$$\begin{aligned}
 Y_0 &= \overline{X_3} L_0 \\
 Y_1 &= \overline{X_3} L_1 \\
 Y_2 &= \overline{X_3} L_2 \\
 Y_3 &= \overline{X_3} L_3 \\
 Y_4 &= \overline{X_3} L_4 \\
 Y_5 &= \overline{X_3} L_5 \\
 Y_6 &= \overline{X_3} L_6 \\
 Y_7 &= \overline{X_3} L_7 \\
 Y_8 &= X_3 L_0 \\
 Y_9 &= X_3 L_1 \\
 Y_{10} &= X_3 L_2 \\
 Y_{11} &= X_3 L_3 \\
 Y_{12} &= X_3 L_4 \\
 Y_{13} &= X_3 L_5 \\
 Y_{14} &= X_3 L_6 \\
 Y_{15} &= X_3 L_7
 \end{aligned}$$

Из анализа данной записи следует, что в состав каждой из функций входит одна из компонент,  $L_j$ .

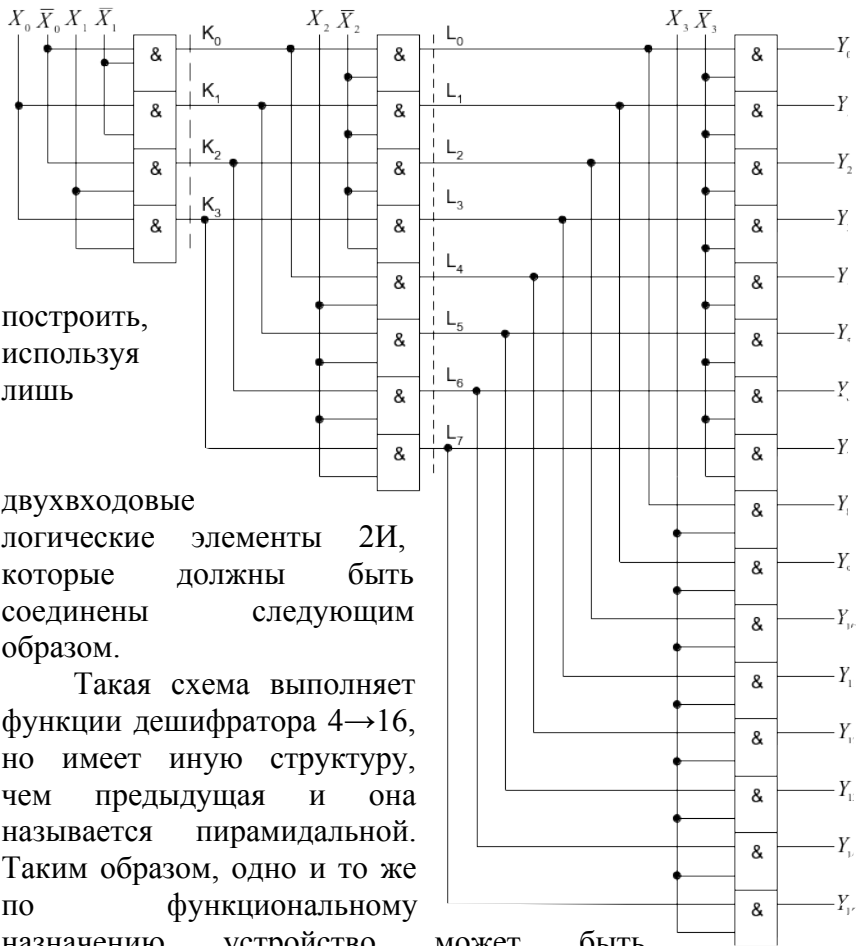
$$\begin{aligned}
 \overline{X_1} \overline{X_0} &= K_0 \\
 \overline{X_1} X_0 &= K_1 \\
 X_1 \overline{X_0} &= K_2 \\
 X_1 X_0 &= K_3
 \end{aligned}$$

$$\begin{aligned}
 Y_0 &= \overline{X_3} \overline{X_2} K_0 \\
 Y_1 &= \overline{X_3} \overline{X_2} K_1 \\
 Y_2 &= \overline{X_3} \overline{X_2} K_2 \\
 Y_3 &= \overline{X_3} \overline{X_2} K_3 \\
 Y_4 &= \overline{X_3} X_2 K_0 \\
 Y_5 &= \overline{X_3} X_2 K_1 \\
 Y_6 &= \overline{X_3} X_2 K_2 \\
 Y_7 &= \overline{X_3} X_2 K_3 \\
 Y_8 &= X_3 \overline{X_2} K_0 \\
 Y_9 &= X_3 \overline{X_2} K_1 \\
 Y_{10} &= X_3 \overline{X_2} K_2 \\
 Y_{11} &= X_3 \overline{X_2} K_3 \\
 Y_{12} &= X_3 X_2 K_0 \\
 Y_{13} &= X_3 X_2 K_1 \\
 Y_{14} &= X_3 X_2 K_2 \\
 Y_{15} &= X_3 X_2 K_3
 \end{aligned}$$

$$\begin{aligned}
 \overline{X_2} K_0 &= L_0 \\
 \overline{X_2} K_1 &= L_1 \\
 \overline{X_2} K_2 &= L_2 \\
 \overline{X_2} K_3 &= L_3 \\
 X_2 K_0 &= L_4 \\
 X_2 K_1 &= L_5 \\
 X_2 K_2 &= L_6 \\
 X_2 K_3 &= L_7
 \end{aligned}$$

Воспользовавшись этим, выражения для функций  $Y$  можно представить в виде.

Отсюда следует, что дешифратор  $4 \rightarrow 16$  можно



построить,  
используя  
лишь

двухвходовые  
логические элементы 2И,  
которые должны быть  
соединены следующим  
образом.

Такая схема выполняет  
функции дешифратора 4→16,  
но имеет иную структуру,  
чем предыдущая и она  
называется пирамидальной.  
Таким образом, одно и то же  
по функциональному  
назначению устройство может быть  
реализовано различными способами.

Особенностью данной схемы является каскадное  
соединение ступеней, переменный коэффициент нагрузки на  
входы (он меняется от 3 в первой до 9 в последней) и  
меньшее быстродействие. Задержка в формировании  
выходного сигнала здесь составит  $4\tau$  ( $\tau$  в инверторе и по  $\tau$  в  
каждой из ступеней). При построении дешифратора такой  
структуры на реальных микросхемах понадобится только 8

(7+1) корпусов, так как элементы, содержащие 4 двухвходовых конъюнктора, выпускаются в одном корпусе.

Еще один вариант дешифратора можно построить, используя несколько иное разбиение исходных функций. Если выражения, описывающие состояния выходов дешифратора разделить на две примерно равные части (в случае дешифратора 4→16 пополам), и ввести новые функции  $A_i$ ,  $B_i$ , то в каждую из исходных  $Y_j$

$$A_0 = \bar{X}_3 \bar{X}_2 \quad B_0 = \bar{X}_1 \bar{X}_0$$

$$A_1 = \bar{X}_3 X_2 \quad B_1 = \bar{X}_1 X_0$$

$$A_2 = X_3 \bar{X}_2 \quad B_2 = X_1 \bar{X}_0$$

$$A_3 = X_3 X_2 \quad B_3 = X_1 X_0$$

$$Y_0 = A_0 B_0 \quad Y_4 = A_1 B_0 \quad Y_8 = A_2 B_0 \quad Y_{12} = A_3 B_0$$

$$Y_1 = A_0 B_1 \quad Y_5 = A_1 B_1 \quad Y_9 = A_2 B_1 \quad Y_{13} = A_3 B_1$$

$$Y_2 = A_0 B_2 \quad Y_6 = A_1 B_2 \quad Y_{10} = A_2 B_2 \quad Y_{14} = A_3 B_2$$

$$Y_3 = A_0 B_3 \quad Y_7 = A_1 B_3 \quad Y_{11} = A_2 B_3 \quad Y_{15} = A_3 B_3$$

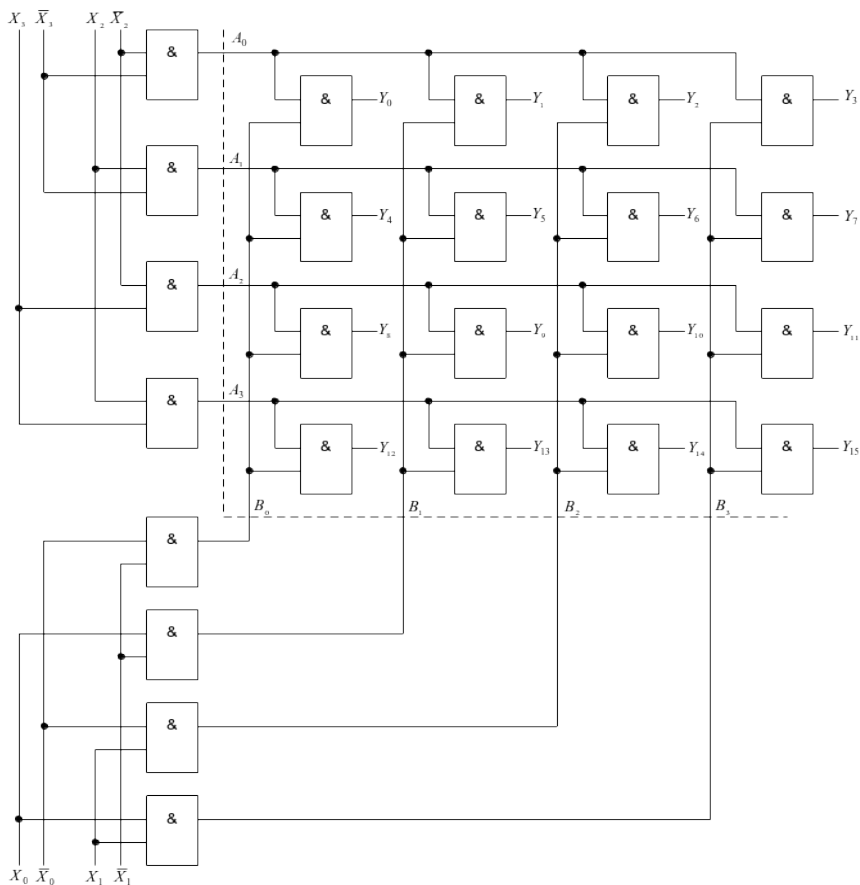
будет входить одна из их конъюнкций.

При этом дешифратор может быть представлен в виде матрицы из 16 двухвходовых элементов И, формирующих выходные сигналы. На их входы должны поступать функции  $A_i$ ,  $B_i$ , формируемые двумя вспомогательными дешифраторами 2→4.

Общее количество логических элементов 2И в такой схеме оказывается равным 28, а количество корпусов микросхем, требуемых для ее создания 7+1. Задержка формирования выходного сигнала составит  $3\tau$ , при коэффициенте нагрузки, 2,3 и 4. Из сравнительных характеристик трех структур дешифраторов, следует, что матричная в целом является более оптимальной.

	Дешифратор 4→16		
	Линейный	Пирамидальный	Матричный
К-во ЛЭ	16 (4И), 4 (НЕ)	28 (2И), 4 (НЕ)	24 (2И), 4 (НЕ)
К-во корп.	8 + 1	7 + 1	6 + 1
К-т нагр.	8, 9	2, 3, 4, 5, 8, 9	2, 3, 4
Макс. зад.	2τ	4τ	3τ
Мощн. потр.	20 P <sub>0</sub>	32 P <sub>0</sub>	28 P <sub>0</sub>

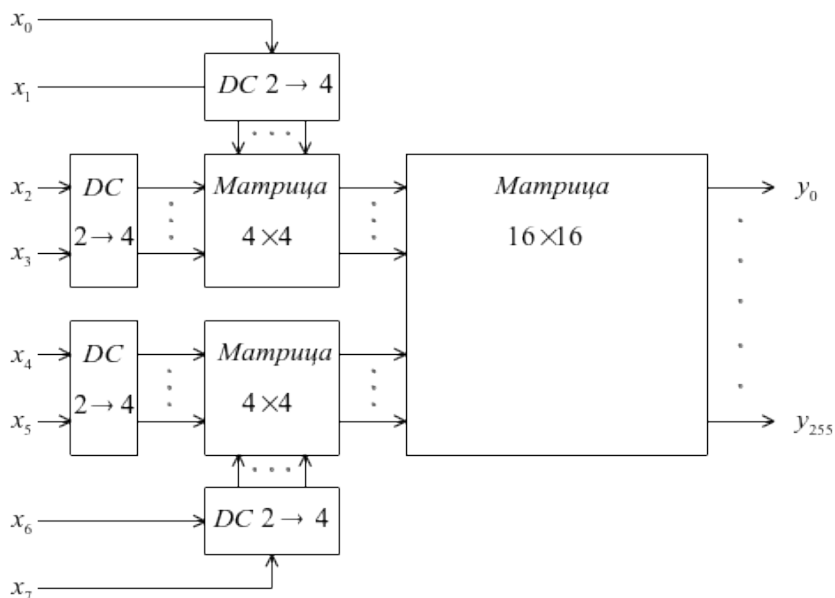




С ростом разрядности ее преимущества становятся еще более значительными. Так для дешифратора 8→256 аналогичная таблица выглядит следующим образом. При

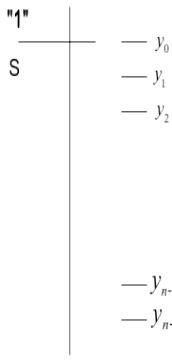
	<i>Дешифратор 8 → 256</i>		
	<i>Линейный</i>	<i>Пирамидальный</i>	<i>Матричный</i>
К-во ЛЭ	256 (8И), 8 (НЕ)	508 (2И), 8 (НЕ)	304 (2И), 8 (НЕ)
К-во корп.	256 + 2	127 + 2	76 + 2
К-т нагр.	128, 129	2 ÷ 129	2 ÷ 16
Макс. зад.	2τ	9τ	4τ
Мощн. потр.	264 P <sub>0</sub>	516 P <sub>0</sub>	312 P <sub>0</sub>

этом предполагается, что матричный дешифратор имеет каскадно-матричную структуру.



Приведенные рассуждения справедливы в случае проектирования дешифраторов на отдельных логических элементах. При их изготовлении методами интегральной технологии, как встроенных узлов сложных цифровых устройств, в качестве схем И применяются диодные сборки и поэтому основные характеристики разных структур дешифраторов будут иными.

Разновидностью дешифратора является дешифратор со с стробированием или дешифратор-демультиплексор. Работу обычного дешифратора можно представить (с моделировать), используя многопозиционный переключатель, на центральный контакт которого подан сигнал логической единицы, предполагая, что номер неподвижного контакта с которым он замыкается определяется управляющим кодом.



В модели дешифратора-демультиплектора на подвижный контакт подается произвольный логический сигнал стробирования  $S$ , который может принимать значение, как логического нуля, так и единицы.

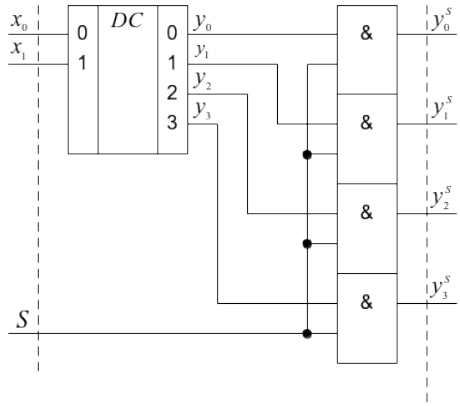
В дешифраторе-демультиплексоре он должен появиться только на выходе, выбранном в соответствии с конкретной комбинацией управляющего кода. Если обозначить сигналы на

выходах дешифратора  $y_i$ , а на выходах демультиплексора -  $y_i^s$ , то связь между ними может быть задана следующим образом.

$y_i$	$S$	$y_i^s$
0	S	0
1	S	S

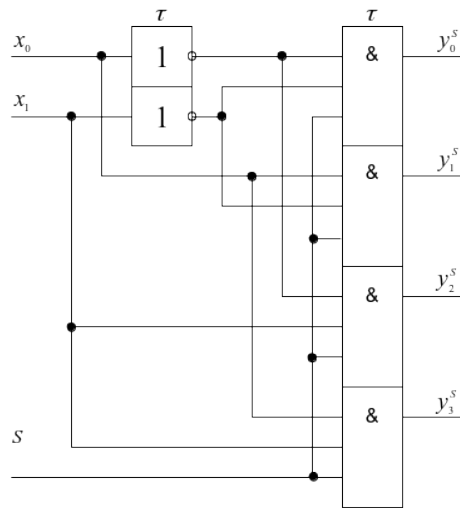
Данная форма табличной записи не является каноническая, так как здесь не конкретизируется значение переменной  $S$ , но из нее можно сделать вывод, что аналитическая связь между значениями функции  $y_i^s$  и  $y_i$  должна иметь вид  $y_i^s = S \cdot y_i$ . Таким образом, дешифратор со стробированием может быть построен с использованием обычного дешифратора выходные сигналы которого обрабатываются в элементах 2И совместно с сигналом стробирования.

Время задержки в такой схеме будет порядка  $3\tau$  ( $2\tau$  в дешифраторе и  $\tau$  в элементе И). Это объясняется тем, что обработка сигналов осуществляется последовательно. На первом этапе формируется выходной



сигнал дешифратора, а затем реализуется его конъюнкция с сигналом стробирования. Аналитическая запись этой процедуры выглядит следующим образом  $y_i^s = y_i \cdot S = (x_k x_l)S$ .

Повысить быстродействие можно перейдя к параллельной обработке соответствующих сигналов, для чего в приведенном выражении



требуется раскрыть скобки. В итоге

$$y_0^s = \overline{x_0} \overline{x_1} S$$

$$y_1^s = x_0 \overline{x_1} S$$

$$y_2^s = \overline{x_0} x_1 S$$

$$y_3^s = x_0 x_1 S$$

получится система уравнений, которой будет соответствовать следующая схема дешифратора-демультиплектора с задержкой формирования выходного сигнала -  $2\tau$ .

В составе серий цифровых микросхем

выпускается большой ассортимент дешифраторов, маркируемых буквами ИД:

К155ИД3 – дешифратор-демультиплексор 4→16; К155ИД4 – комбинированный дешифратор-демультиплексор 2→4, либо 3→8; К555ИД6 – дешифратор 4→10 (неполный); К1533ИД7 - дешифратор-демультиплексор 3→8; К555ИД10 - дешифратор 4→10 (выходы с открытым коллектором) и т.д.

При необходимости создания дешифраторов большой разрядности помимо рассмотренного подхода, подразумевающего их построение на логических элементах, можно использовать готовые функционально законченные узлы.

Пусть требуется реализовать дешифратор 3→8 на основе дешифраторов-демультиплексоров 2→4. Количество необходимых узлов можно в первом приближении оценить

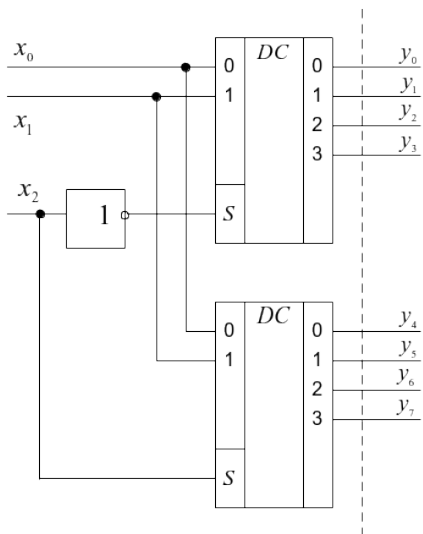
путем деления числа выходов у проектируемого дешифратора на их количество у каждого из исходных модулей. Однако, у дешифратора 3→8 число управляющих входов – 3, а у каждого из исходных по два и дополнительный вход стробирования. Таким образом основным здесь является вопрос объединения модулей в систему.

Таблица, описывающая связь номера активизированного выхода дешифратора 3→8 с входной кодовой комбинации выглядит следующим образом. Если младшие разряды  $x_0, x_1$  подать на входы одного из

Вых. DC1    Вых. DC2

$x_2$	$x_1$	$x_0$	Актив. вых.	$y_0$	$y_1$	$y_2$	$y_3$	$y_0$	$y_1$	$y_2$	$y_3$	$S_0$	$S_1$
0	0	0	$y_0$	1	0	0	0	0	0	0	0	1	0
0	0	1	$y_1$	0	1	0	0	0	0	0	0	1	0
0	1	0	$y_2$	0	0	1	0	0	0	0	0	1	0
0	1	1	$y_3$	0	0	0	1	0	0	0	0	1	0
1	0	0	$y_4$	0	0	0	0	1	0	0	0	0	1
1	0	1	$y_5$	0	0	0	0	0	1	0	0	0	1
1	1	0	$y_6$	0	0	0	0	0	0	1	0	0	1
1	1	1	$y_7$	0	0	0	0	0	0	0	1	0	1

дешифраторов 2→4, то на первых четырех наборах переменных  $x_0 \div x_2$  он будет функционировать в



соответствии с логикой работы дешифратора 3→8. Однако, так как на остальных наборах значения переменных  $x_0, x_1$  станут повторяться, то на выходах этого дешифратора вновь начнут появляться единичные сигналы, которых не должно быть, если он входит в состав дешифратора 3→8.

Данную проблему можно решить, заблокировав работу дешифратора 2→4 на соответствующих наборах переменных. Для этого на его вход стробирования необходимо подать сигнал логического нуля. Второй дешифратор 2→4 должен вступить в работу при  $x_2 = 1$  и тех же комбинациях переменных  $x_0, x_1$ . Их потребуется подать на его входы управления, а на вход стробирования - логический ноль при  $x_2 = 0$  и единицу в случае  $x_2 = 1$ . Таким образом, схема дешифратора 3→8 будет иметь вид.

#### 4.2 Преобразователи кода.

При решении ряда практических задач, в частности определении двоичного кода номера устройства, сформировавшего на соответствующей линии сигнал, к примеру логической единицы, требуется преобразовать кодовую комбинацию из унитарного кода в двоичный. Эта процедура обратна той, которую выполняет дешифратор.

Так как большинство логических элементов не обладают свойством обратимости и на их активные выходы не допускается подача сигналов, то для решения данной задачи необходимо проектирование специального устройства, которое относится к классу преобразователей кодов.

Таблица его функционирования для восьмиразрядного входного кода выглядит следующим образом. Хотя

$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$	количество возможных
0	0	0	0	0	0	0	1	0	0	0	комбинаций входных
0	0	0	0	0	0	1	0	0	0	1	сигналов 256, она
0	0	0	0	0	1	0	0	0	1	0	содержит всего 8
0	0	0	0	1	0	0	0	0	1	1	строк, так как иных
0	0	0	1	0	0	0	0	1	0	0	комбинаций, чем
0	0	1	0	0	0	0	0	1	0	1	представленные, в
0	1	0	0	0	0	0	0	1	1	0	унитарном коде не
1	0	0	0	0	0	0	0	1	1	1	

существует. Отсюда следует, что набор выходных функций  $y_i$  такого устройства относится к классу недоопределенных.

Применение карт Карно для их представления в алгебраической форме нецелесообразно, из-за их сложности вследствие большого количества компонент. В таких случаях удобнее использовать нестандартные подходы, позволяющие получить в алгебраическом виде одну из возможных форм связи между входными и выходными переменными. При этом гарантий того, что она окажется минимальной не будет. Однако, имея результат, его в дальнейшем можно трансформировать с целью оптимизации.

Из анализа таблицы видно, что функция  $y_0$  становится равной единице когда  $x_1$  либо  $x_3, x_5, x_7$  принимают единичные значения, а остальные переменные обращаются в нуль. При этом из-за недоопределенности можно считать, что  $y_1 = x_2 + x_3 + x_6 + x_7$  на всех остальных наборах переменных  $y_2 = x_5 + x_4 + x_6 + x_7$  также равна нулю.

Тогда выражения для нее и остальных выходных функций преобразователя кода представленные в дизъюнктивной нормальной форме будут выглядеть следующим образом. Эти же функции можно представить и в конъюнктивной форме. Здесь в выражения входят логические произведения инверсий переменных при единичном значении которых функция обращается в нуль.

$$y_0 = x_0 x_2 x_4 x_6$$

$$y_1 = x_0 x_1 x_4 x_5$$

$$y_2 = x_0 x_1 x_2 x_3$$

Рассматривая работу цифровых устройств с несколько иной точки зрения, их можно использовать для реализации, на первый взгляд не свойственных им функций. Рассмотрим таблицу, описывающую работу преобразователя унитарного кода логических единиц в двоичный и обратное преобразование, выполняемое дешифратором.

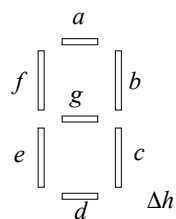
Если кодовые комбинации, представленные в строках таблицы, представлять не как наборы нулей и единиц, а как

Вх. код ПК Вых. код ДШ		Преобр. кода $\Leftrightarrow$ ДШ								Вых. код ПК Вх. код ДШ			Число
Число	$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$		
1	0	0	0	0	0	0	0	1	0	0	0	0	
2	0	0	0	0	0	0	1	0	0	0	1	1	
4	0	0	0	0	0	1	0	0	0	1	0	2	
8	0	0	0	0	1	0	0	0	0	1	1	3	
16	0	0	0	1	0	0	0	0	1	0	0	4	
32	0	0	1	0	0	0	0	0	1	0	1	5	
64	0	1	0	0	0	0	0	0	1	1	0	6	
128	1	0	0	0	0	0	0	0	1	1	1	7	

двоичные коды чисел, то преобразователь кода ставит в соответствие числам 1,2,4,8, и т.п. числа 0,1,2,3..., являющиеся показателями степени в которую требуется возвести двойку, чтобы получить исходное число.

С этой точки зрения преобразователь кода можно рассматривать как устройство, осуществляющее процедуру логарифмирования определенного набора чисел. Дешифратор осуществляет обратное преобразование, то есть числу  $k$  ставит в соответствие  $2^k$ .

Часто встречающейся разновидностью преобразователей кода являются схемы для управления семисегментными индикаторами, который представляет собой набор обычно из семи полосок-сегментов, расположенных следующим образом. Каждый из них определенным сигналом может быть переведена в одно из двух состояний, отличающихся тем, что в одном из них сегмент может светиться, либо изменить свой контраст.





Сегменты обозначаются латинскими буквами от a до g. Часто в такой индикатор вводится дополнительный сегмент h, выполняющий роль десятичной точки. Комбинируя совокупности, к примеру, светящихся, сегментов можно сформировать изображение любой десятичной цифры и ряда буквенных символов.

Двоич. код				Цифра	a b c d e f g	Отабражение
X3	X2	X1	X0			
0	0	0	0	0	1 1 1 1 1 1 0	
0	0	0	1	1	0 1 1 0 0 0 0	
0	0	1	0	2	1 1 0 1 1 0 1	
		.				
		.				
		.				
1	0	0	1	9	1 1 1 1 0 1 1	
1	0	1	0	A	1 1 1 0 1 1 1	
1	0	1	1	B	0 0 1 1 1 1 1	
1	1	0	0	C	1 0 0 1 1 1 0	
1	1	0	1	D	0 1 1 1 1 0 1	
1	1	1	0	E	1 0 0 1 1 1 1	
1	1	1	1	F	1 0 0 0 1 1 1	

Обычно вид символа задается двоичным кодом. Соотношения между значениями кодовых комбинаций, цифрами и их представлением (отображением) приведено в таблице. Считая, что единичному значению сигнала управления сегментом соответствует его свечение или изменение контраста, комбинациям двоичного кода символов можно поставить в соответствие наборы значений этих

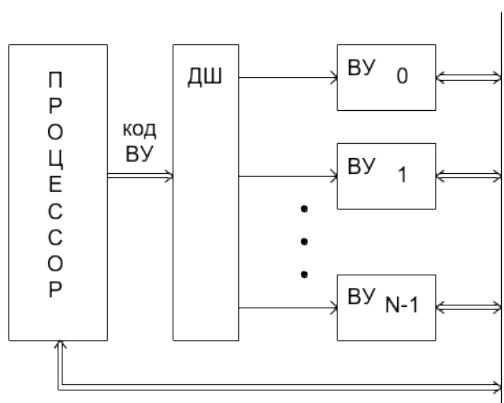
сигналов. Далее необходимо определить функциональные связи  $a = f_1(x_0, x_1, x_2, x_3)$ ,  $b = f_2(x_0, x_1, x_2, x_3)$  и т.п., после чего можно выбрать логические элементы и разработать принципиальную схему такого устройства.

Такие преобразователи кодов выпускаются промышленностью в составе различных серий микросхем – К514ИД1, К514ИД2, К155ПП4, К555ИД18 и другие.

### 4.3 Приоритетный шифратор.

Данное устройство также относится к группе преобразователей кодов и обычно используется как узел аппаратной поддержки при обработке прерываний. Обобщенный алгоритм его работы можно определить следующим образом.

Пусть имеется система, содержащая процессор и ряд внешних устройств ВУ<sub>0</sub> – ВУ<sub>N-1</sub> с которыми ему требуется



осуществлять обмен информацией. Данные при обмене передаются по общей шине, но работа с каждым конкретным устройством имеет свои особенности. В качестве внешних устройств в персональном компьютере могут

выступать клавиатура, монитор, модем и т. п.

Один из способов организации обмена состоит в том, что процессор формирует сигнал, переводящий соответствующее устройство в режим передачи или приема данных. Так как таких устройств обычно несколько, то

сигнал процессора должен в любой момент времени поступать только на одно из них.

С целью уменьшения количества линий связи процессор формирует двоичный код номера устройства, который преобразуется в сигнал, поступающий по отдельной линии к каждому из них, для начала процесса обмена. Преобразование кода номера в соответствующий сигнал может быть реализовано с помощью дешифратора.

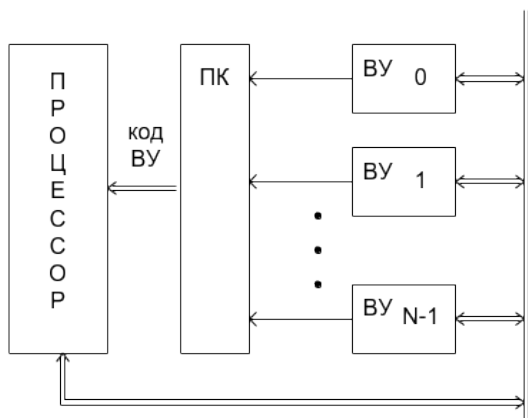
При таком варианте обмена, процессор выступает в качестве активного узла. Для обмена данными ему требуется прервать работу над текущей задачей и это необходимо делать каждый раз при опросе любого из устройств, независимо от того состоится ли с их стороны передача, либо прием данных или нет. Даже, если ни одно из устройств не готово к обмену, либо в данный момент времени он не требуется, процедура опроса все равно должна осуществляться, так как устройство не может сообщить процессору необходим ли в данный момент времени обмен, или нет. При таком подходе будут достаточно велики непроизводительные потери времени процессора.

Другой вариант организации обмена заключается в том, что активными выступают сами внешние устройства и они по мере необходимости формируют запросы прерывания. Эти сигналы, поступающие по определенным линиям должны быть преобразованы в двоичные коды номеров устройств, чтобы иметь возможность определить его разновидность и производить обмен данными в соответствии с установленными правилами.

Возможны и комбинированные варианты решения этой задачи, в частности при передаче данных к устройствам в качестве активного элемента может выступать процессор, а при обратной передаче – сами устройства.

Однако в последних двух случаях возникает проблема, когда сигналы запроса приходят от нескольких устройств одновременно. Логически эта задача неразрешима, так как

если устройства равнозначны, то выбрать одно из них невозможно.



Чтобы выйти из этого положения, объекты выбора требуется каким-либо образом различать. Один из подходов заключается во введении принципа приоритетности, то есть неравнозначности, или старшинства.

При этом объектам присписывается разная степень значимости, или важности и требования самого значимого из них должны выполняться в первую очередь, независимо от ситуации с другими.

Данный принцип можно реализовать следующим образом. Линии сигналов прерывания и соответствующие объекты нумеруются, причем линии с меньшим номером присписывается более старший приоритет. Сигнал запроса на прерывание обычно формируется в виде уровня логического нуля.

Так как комбинации сигналов, поступающих на преобразователь кода могут быть самыми различными, то для синтеза шифратора приоритетов необходимо в таблице соответствия входных и выходных переменных рассмотреть все возможные их варианты.

Однако такая таблица получится очень громоздкой, в частности для восьми входных переменных она будет содержать 256 строк и определить из нее связь между входными и выходными переменными затруднительно. В таких случаях выходом является поиск закономерностей, позволяющих упростить структуру таблицы, для получения

какого либо варианта аналитической связи аргументов и функций.

Гарантий, что полученная форма представления окажется минимальной не будет, но ее можно использовать как основу для дальнейшей модификации.

Используя данный подход, таблицу функционирования шифратора приоритетов, можно представить следующим образом. Здесь крестиками обозначены произвольные

Таблица истинности и структурная схема приоритетного шифратора 8→3.

Приоритет	Состояния входов								Состояния выходов		
	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$y_2$	$y_1$	$y_0$
0	0	x	x	x	x	x	x	x	0	0	0
1	1	0	x	x	x	x	x	x	0	0	1
2	1	1	0	x	x	x	x	x	0	1	0
3	1	1	1	0	x	x	x	x	0	1	1
4	1	1	1	1	0	x	x	x	1	0	0
5	1	1	1	1	1	0	x	x	1	0	1
6	1	1	1	1	1	1	0	x	1	1	0
7	1	1	1	1	1	1	1	0	1	1	1

состояния входов, то есть либо логический ноль, либо единица. Из таблицы следует, что появление нуля на линии с меньшим номером (более высоким приоритетом) независимо от сигналов на остальных входах должно устанавливать на выходе код номера этой линии.

Так, если нулевой сигнал поступал на вход  $x_4$  (при этом независимо от сигналов на входах  $x_5$ - $x_7$  формировался выходной код 100.), а затем появился нулевой уровень на входе  $x_1$ , то в соответствии с таблицей, будет формироваться

код 001, соответствующий номеру линии, по которой пришел сигнал запраса.

То есть такая таблица действительно описывает работу шифратора приоритетов. Использование карт Карно для выявления аналитических связей между входными и выходными переменными в данном примере затруднительно из-за большого количества переменных. Поэтому здесь более целесообразным здесь является использование неформальных подходов к анализу свойств соответствующих функций.

Если рассмотреть функцию  $y_2$ , то можно сделать вывод, что она принимает единичное значение, лишь при одновременном обращении в единицы переменных  $x_0, x_1, x_2$  и  $x_3$ . Отсюда следует, что  $y_2 = x_0 x_1 x_2 x_3$ .

Функция  $y_1 = f(x_0, x_1, \dots, x_6, x_7)$  принимает нулевое значение, если  $x_0$ , либо  $x_1$  равны нулю, независимо от состояния других переменных. Это возможно, если она имеет вид  $y_1 = x_0 x_1 f_1(x_2, x_3, \dots, x_6, x_7)$ . В следующих двух строках  $y_1 = 1$  при  $x_0, x_1$  равных единице и нулевых значениях  $x_2$ , либо  $x_3$ , что возможно, если  $y_1 = x_0 x_1 ((\overline{x_2 + x_3}) + f_2(x_4 \dots x_6, x_7))$ , так как только при этом  $\overline{x_2 + x_3} = 1$  и  $y_1 = 1 \cdot 1 \cdot ((\overline{x_2 + x_3}) + f_2(x_4 \dots x_6, x_7)) = 1 + f_2(x_4 \dots x_6, x_7) = 1$ .

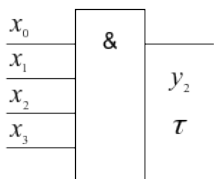
Когда  $x_4$ , или  $x_5$  равны нулю, а  $x_0 - x_3$  единице, функция  $y_1$  должна обратиться в нуль, независимо от значений  $x_6, x_7$ . Отсюда следует, что  $y_1 = x_0 x_1 ((\overline{x_2 + x_3}) + x_4 x_5 f_3(x_6, x_7))$ , так как  $y_1 = 1 \cdot 1 \cdot ((0 + 0) + x_4 x_5 \cdot f_3(x_6, x_7)) = 1 \cdot (0 + 0 \cdot f_3(x_6, x_7)) = 0$ . Эта функция принимает единичное значение на последних двух наборах переменных, если  $f_3(x_6, x_7) = \overline{x_6 + x_7}$ . Таким образом, ее окончательный вид будет  $y_1 = x_0 x_1 ((\overline{x_2 + x_3}) + x_4 x_5 (\overline{x_6 + x_7}))$ .

Как уже отмечалось, использование нестандартных подходов не дает гарантии того, что полученное выражение будет представлять функцию в минимизированной форме.

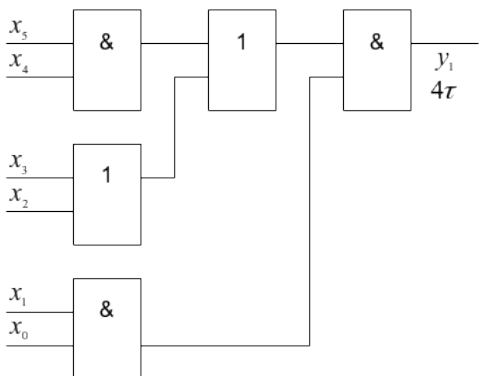
Однако, в данной ситуации из-за отсутствия комбинации переменных  $x_6 = x_7 = 1$ , логическая сумма  $(\overline{x_6} + \overline{x_7})$  будет тождественно равна единице, а следовательно эту компоненту можно исключить и функция примет вид  $y_1 = x_0 x_1 ((\overline{x_2} + \overline{x_3}) + x_4 x_5)$ .

Используя аналогичные рассуждения, можно прийти к выводу, что функция  $y_0$  будет выглядеть следующим образом  $y_0 = x_0 (\overline{x_1} + x_2 (\overline{x_3} + x_4 (\overline{x_5} + x_6 (\overline{x_7}))))$ . Из таблицы видно, что в последней строке  $x_7$  принимает нулевое значения, то есть  $\overline{x_7} \equiv 1$  и функция  $y_0$  также может быть упрощена до выражения  $y_0 = x_0 (\overline{x_1} + x_2 (\overline{x_3} + x_4 (\overline{x_5} + x_6)))$ .

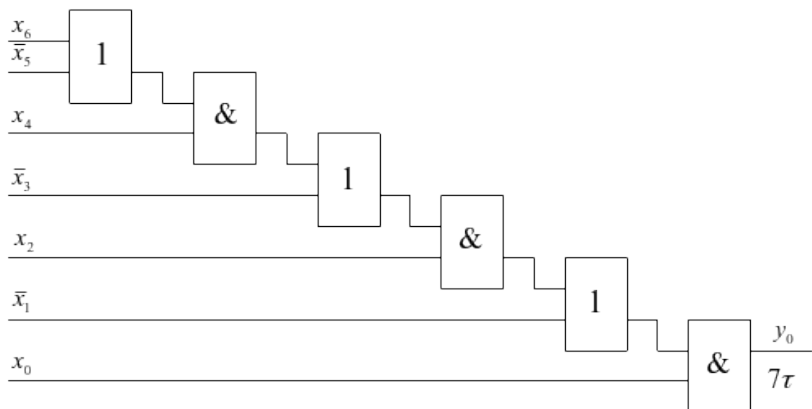
Из полученных соотношений следует, что для реализации шифратора приоритетов требуется три узла, схемы которых выглядят таким образом.



Существенным недостатком данной схемы шифратора является большое время задержки появления выходного сигнала  $y_0$ , оно составляет  $7\tau$ . Это время определяет и максимальное



быстродействие системы, так как двоичный код номера линии, по которой приходит запрос, может быть считан, лишь когда сформируются все его разряды.



Причина этого объясняется структурой выражений, описывающих функции  $y_0$  и  $y_1$ .

Они задают последовательное выполнение действий над входными переменными. Для функции  $y_2$  все четыре аргумента обрабатываются одновременно (параллельно), а так как задержка логического элемента практически не зависит от числа его входов, то значение функции сформируется через  $\tau$ . Однако, используя законы алгебры логики,  $y_2$  можно представить и в таком виде  $y_2 = x_0 x_1 x_2 x_3 = x_0 (x_1 (x_2 x_3))$ . Здесь задержка ее формирования окажется равной  $3\tau$ .

Если логические элементы считать идеальными, то разницы в задержках появления выходного сигнала для первого и второго выражений не будет, однако для реальных элементов это не так.

Для повышения быстродействия шифратора приоритетов требуется представить выражения, полученные для функций  $y_0$  и  $y_1$  в таком виде, чтобы как можно больше переменных обрабатывались параллельно. Применительно к функции  $y_1$  это можно сделать путем последовательной замены действий в соответствии с правилами Де-Моргана. Цепочка преобразований будет при этом выглядеть



следующим образом. Уголками сверху помечены действия, которые меняются на каждом следующем шаге преобразований.

$$\begin{aligned}
 y_1 &= x_0 x_1 ((\bar{x}_2 \hat{+} \bar{x}_3) + x_4 \hat{\cdot} x_5) = x_0 x_1 (\overline{x_2 x_3} \hat{+} (\overline{\bar{x}_4 + \bar{x}_5})) = x_0 x_1 (\overline{x_2 x_3 \cdot (\bar{x}_4 + \bar{x}_5)}) = \\
 y_0 &\equiv x_0 x_1 (\overline{\overline{\overline{\bar{x}_1 \hat{+} \bar{x}_2} (\bar{x}_3 \hat{+} \bar{x}_4)} (\bar{x}_5 \hat{\cdot} \bar{x}_6)})} = x_0 x_1 (\overline{\bar{x}_1 + x_2 (\bar{x}_3 + x_4 \hat{\cdot} x_5 \cdot \bar{x}_6)}) = \\
 &= x_0 (\overline{\bar{x}_1 + x_2 (\overline{\bar{x}_3 \hat{+} \bar{x}_4 + x_5 \bar{x}_6})}) = x_0 (\overline{\bar{x}_1 + x_2 (\overline{x_3 \cdot (\bar{x}_4 + x_5 \bar{x}_6)})}) = \\
 &= x_0 (\overline{\bar{x}_1 + x_2 (\overline{x_3 \cdot \bar{x}_4 \hat{+} x_3 x_5 \bar{x}_6})}) = x_0 (\overline{\bar{x}_1 + x_2 \hat{\cdot} x_3 \bar{x}_4 \hat{\cdot} x_3 x_5 \bar{x}_6}) = \\
 &= x_0 (\overline{\bar{x}_1 \hat{+} \bar{x}_2 + x_3 \bar{x}_4 + x_3 x_5 \bar{x}_6}) = x_0 (\overline{x_1 \cdot (\bar{x}_2 + x_3 \bar{x}_4 + x_3 x_5 \bar{x}_6)}) = \\
 &= x_0 (\overline{x_1 \bar{x}_2 \hat{+} x_1 x_3 \bar{x}_4 \hat{+} x_1 x_3 x_5 \bar{x}_6}) = x_0 \cdot \overline{x_1 \bar{x}_2} \cdot \overline{x_1 x_3 \bar{x}_4} \cdot \overline{x_1 x_3 x_5 \bar{x}_6} .
 \end{aligned}$$

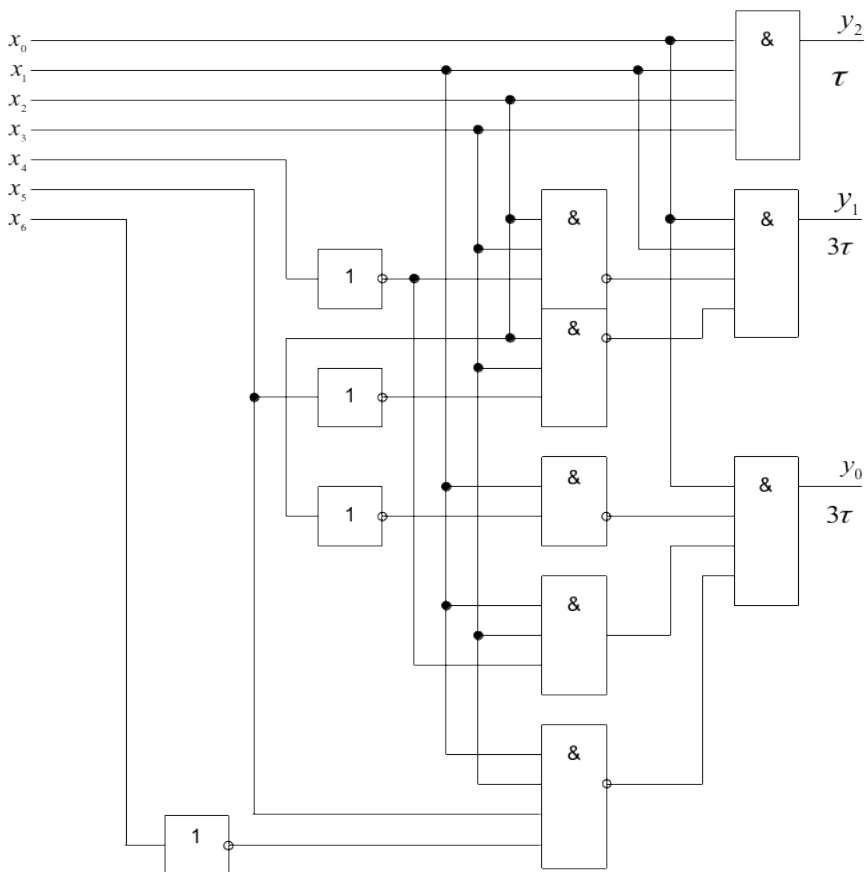
Аналогичные преобразования можно провести и для  $y_0$

Исходя из полученных выражений структура шифратора приоритетов будет иметь вид, представленный на стр. 71. Максимальные задержки формирования выходных сигналов в таком устройстве уменьшатся до  $3\tau$ .

В реальных шифраторах, реализованных в микросхемах К155ИВ1 (шифратор 8→3) и К555ИВ3 (шифратор 10→4), имеются дополнительные входы расширения, позволяющие наращивать разрядность обрабатываемых сигналов путем соответствующего объединения функционально законченных устройств.

Из данного примера следует, что, как и в случае с дешифраторами, устройство, выполняющее одну и ту же функцию, может иметь различные конфигурации и обладать разными характеристиками.

Если вернуться к таблице функционирования шифратора приоритетов, то из нее следует, что, кроме своей основной функции он позволяет осуществлять преобразование унитарного кода логических единиц в двоичный.

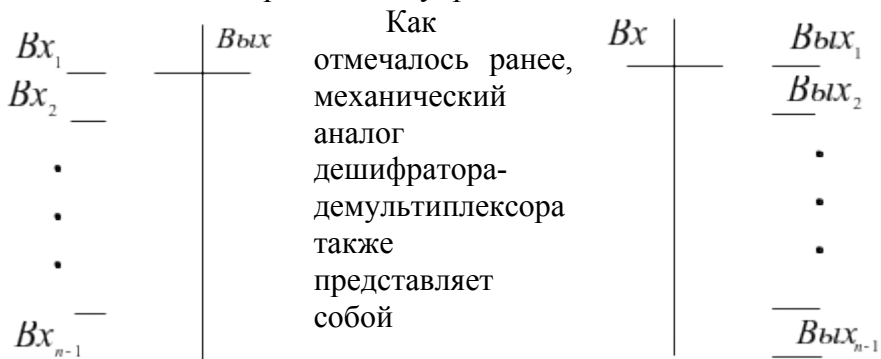


#### 4.4 Мультиплексоры.

Мультиплексором или коммутатором называется устройство, осуществляющее переключение (мультиплексирование) множества информационных сигналов на один выход. Выбор входного сигнала производится заданием с помощью набора управляющих сигналов его номера, то есть номера соответствующего входа.

Обычно этот номер представляется в двоичном коде, хотя могут быть и другие варианты.

Если рассмотреть механическую аналогию мультиплексора, то его можно представить в виде многопозиционного переключателя, неподвижные контакты которого являются входами, а подвижный – выходом, причем его положение определяется управляющим кодом.



многопозиционный переключатель с одним входом и множеством выходов. Отсюда следует, что применяя механические узлы, мультиплексор и дешифратор-демультиплексор можно построить на одном и том же переключателе поменяв местами входы и выходы.

Однако при использовании логических элементов для переключения цифровых сигналов, мультиплексор требуется проектировать как самостоятельный узел, так как большинство вентилях не обладают свойством двунаправленной передачи сигналов. В то же время, в составе КМОП микросхем имеются элементы, называемые двунаправленными ключами, которые таким свойством обладают и на их основе можно создать универсальные переключающие устройства.

Пусть требуется построить мультиплексор коммутирующий сигналы с четырех информационных входов на выход. В этом случае номер входа в двоичном коде будет

задаваться двухразрядной комбинацией управляющих сигналов, которые обычно обозначаются  $S_0$  и  $S_1$ .

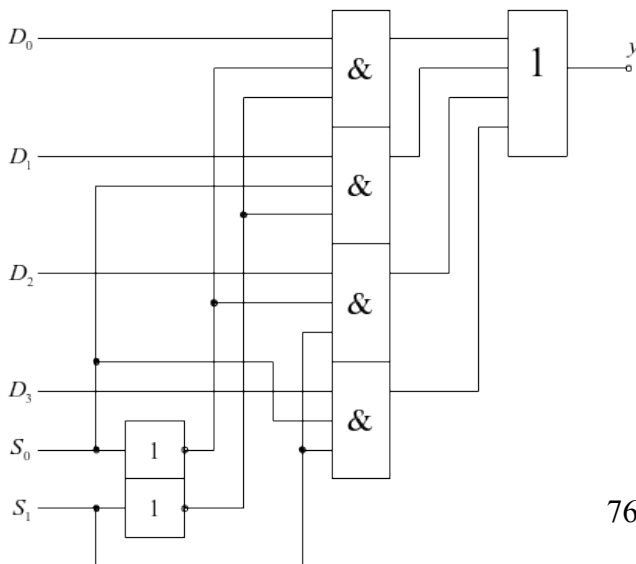
$S_1$	$S_0$	Инф. входы				$y$
0	0	$D_0$	$D_1$	$D_2$	$D_3$	$D_0$
0	1	$D_0$	$D_1$	$D_2$	$D_3$	$D_1$
1	0	$D_0$	$D_1$	$D_2$	$D_3$	$D_2$
1	1	$D_0$	$D_1$	$D_2$	$D_3$	$D_3$

Таблица функционирования такого мультиплексора 4→1 выглядит следующим образом. При наличии на информационных входах произвольных сигналов  $D_i$  нулевая кодовая комбинация

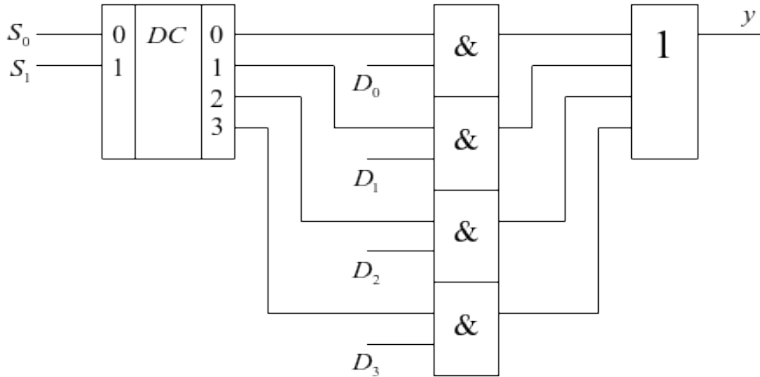
на управляющих входах вызовет подключение к выходу мультиплексора сигнала  $D_0$ , комбинация 0 1 -  $D_1$  и т.д.

Отсюда следует, что функция, описывающая состояние выхода мультиплексора может быть представлена в виде  $y = \bar{S}_0 \bar{S}_1 D_0 + S_0 \bar{S}_1 D_1 + \bar{S}_0 S_1 D_2 + S_0 S_1 D_3$ . Приведенная таблица функционирования является упрощенной, так как у мультиплексора 4→1 шесть входных переменных - две управляющих и четыре информационных. Однако, как и в случае с шифратором приоритетов, такая упрощенная запись позволяет найти один из возможных вариантов функции  $y = f(S_0, S_1, D_j)$ .

Из полученного соотношения следует, что структура

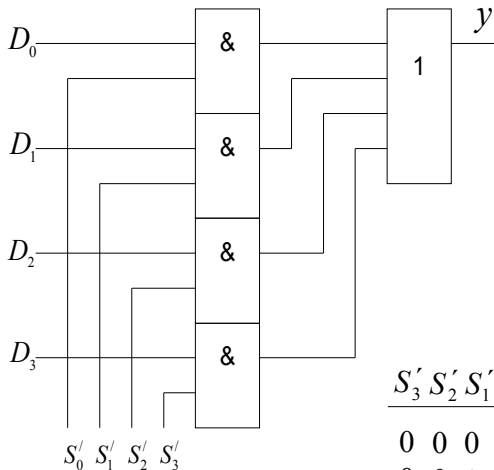


мультиплектора будет такой. В приведенной выше записи прямые и инверсные значения переменных  $S_i$  входят в конъюнкции в комбинациях, характерных для выходных функций  $y_j$  дешифратора  $2 \rightarrow 4$ , ( $y_0 = \bar{x}_0 \bar{x}_1$ ,  $y_1 = x_0 \bar{x}_1$  и т. п.). Таким образом мультиплексор может быть реализован с



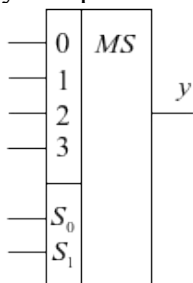
использованием вспомогательного дешифратора, а его выходной сигнал описан соотношением  $y = y_0 D_0 + y_1 D_1 + y_2 D_2 + y_3 D_3$ .

В ряде случаев требуются мультиплексоры, управляемые унитарным кодом. Таблица функционирования такого устройства и соотношение



$S'_3$	$S'_2$	$S'_1$	$S'_0$	Инф. входы				$y$
0	0	0	1	$D_0$	$D_1$	$D_2$	$D_3$	$D_0$
0	0	1	0	$D_0$	$D_1$	$D_2$	$D_3$	$D_1$
1	1	0	0	$D_0$	$D_1$	$D_2$	$D_3$	$D_2$
1	0	0	0	$D_0$	$D_1$	$D_2$	$D_3$	$D_3$

$y = S_0'D_0 + S_1'D_1 + S_2'D_2 + S_3'D_3$ , описывающее его работу имеют вид. Схема данной разновидности мультиплексора отличается от рассмотренной ранее отсутствием дешифратора, так как он выполняет функцию преобразователя двоичного кода управляющих сигналов в унитарный.

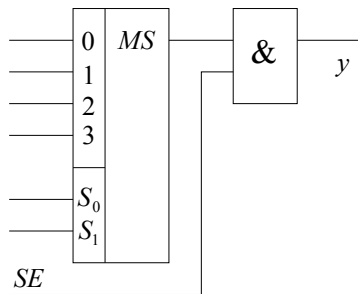


Мультиплексор  $4 \rightarrow 1$ , как функциональный элемент изображается на принципиальных схемах следующим образом.

Дополнительной разновидностью мультиплексоров являются мультиплексоры со стробированием. Они имеют специальный вход  $SE$ , при поступлении на который определенного сигнала, прохождение информационного на выход блокируется. В частности независимо от его значения, на выходе формируется логический ноль.

Данный мультиплексор может быть реализован путем подключения к выходу стандартного устройства элемента 2И, как показано на рисунке.

Ряд мультиплексоров имеют выходной каскад с тремя состояниями и, соответственно дополнительный вход управления  $EO$ , определенный на котором либо активизирует выходной каскад, либо переводит его в состояние высокого импеданса.



При необходимости построения мультиплексоров более высокой разрядности ( $N \rightarrow 1$ ) можно использовать соотношение, описывающее связь между  $N = 2^n$  входными переменными,  $n$  управляющими и функцией, формируемой на выходе.

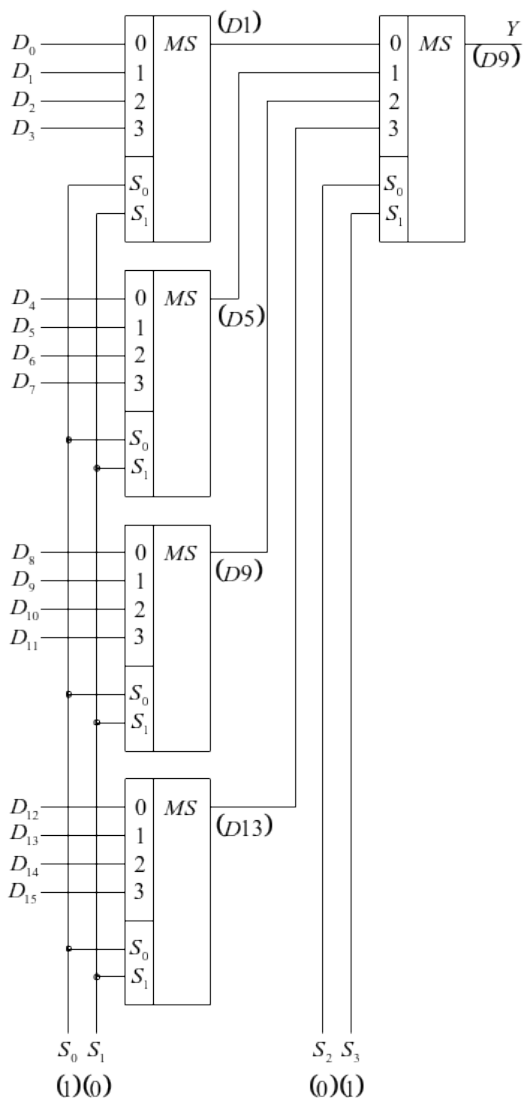
$$y = \bigvee_{j=0}^{N-1} \bigwedge_{i=0}^{n-1} ((x_i \cdot a_{ij}) \vee (\bar{x}_i \cdot \bar{a}_{ij}) \cdot D_j) = \bigvee_{j=0}^{N-1} y_j D_j$$

Здесь  $j$  – номер информационного,  $i$  управляющего входов,  $a_{ij}$  – коэффициенты разложения номера входа  $j$  в двоичный код,  $y_j$  – функции, описывающие сигналы на выходах дешифратора  $n \rightarrow N$ .

Однако, для построения многоразрядных мультиплексоров в ряде случаев более удобным является применение типовых функционально законченных узлов, выпускаемых промышленностью в составе различных серий ТТЛ, ТТЛШ и КМОП микросхем, в частности: К155КП1 (мультиплексор  $16 \rightarrow 1$  со стробированием); К155КП2 (сдвоенный мультиплексор  $4 \rightarrow 1$  со стробированием); К155КП7 (мультиплексор  $8 \rightarrow 1$  со стробированием); К555КП15 (мультиплексор  $8 \rightarrow 1$  с тремя состояниями выходов) и т.п.

Если используются мультиплексоры с обычными выходными каскадами, то вариант схемы мультиплексора 16→1 на основе четырехразрядных блоков выглядит следующим образом.

Здесь к линейке мультиплексоров подключаются все информационные



и на них подаются младшие разряды управляющих  $S_0, S_1$ . Правый

мультиплексор является как бы собирающим и управляется старшими разрядами  $S_2, S_3$ .

Пусть

$S_3$	$S_2$	$S_1$	$S_0$
1	0	0	1

требуется передать на выход сигнал с информационного входа с номером 9. Этому будет соответствовать следующая комбинация управляющих сигналов.

Каждый из мультиплексоров линейки воспримет



набор сигналов  $S_0 = 1, S_1 = 0$  как код числа 1 и они подключат к выходам информационные сигналы со своих входов с номером 1.

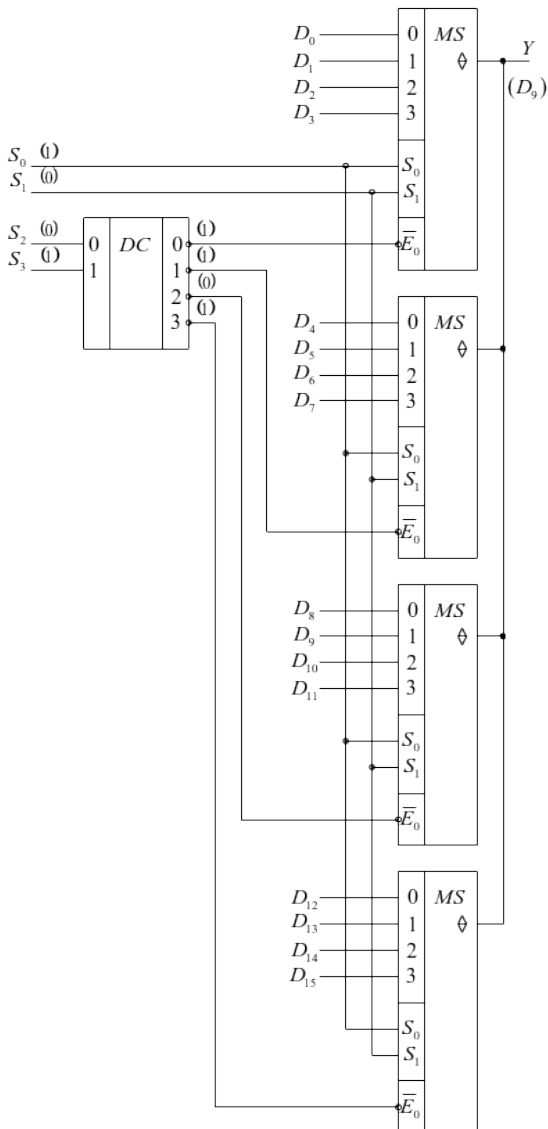
На их выходах появятся сигналы  $D_1, D_5, D_9$  и  $D_{13}$ . Собирающий мультиплексор, для которого кодовая комбинация  $S_2 = 0, S_3 = 1$  соответствует числу 2, передаст на выход устройства данные со своего информационного входа с номером 2, то есть  $D_9$ .

Подобным образом можно каскадировать блоки для создания мультиплексоров  $64 \rightarrow 1$  и т.п. Однако с ростом разрядности увеличивается количество последовательно соединяемых линеек мультиплексоров и, соответственно, задержка прохождения информационных сигналов.

Более высоким быстродействием обладает устройство иной конфигурации. Для его реализации потребуются мультиплексоры, выходы которых могут переводиться в третье состояние и вспомогательный дешифратор. Если активизация выхода мультиплексора происходит при подаче на вход  $\overline{EO}$  сигнала логического нуля, то дешифратор должен формировать унитарный код логических нулей.

Схема такого устройства представлена на стр.78. Здесь младшими разрядами выбирается один из четырех информационных сигналов каждого из мультиплексоров. Наличие трех состояний у выходного каскада обозначается значком в виде перечеркнутого ромба. Для кодовой комбинации, то также будут  $D_1, D_5, D_9$  и  $D_{13}$ .

Дешифратор при  $S_0 = 1, S_1 = 0$  сформирует логический ноль только на выходе с номером 2 и активизирует выходной каскад третьего по счету мультиплексора. У остальных они сохраняют третьье, высокоимпеданное, состояние. При этом на



выход устройства поступит информационный сигнал со входа  $D_9$ , что соответствует номеру, задаваемому комбинацией управляющих сигналов  $S_0 \div S_3$ .

Такой способ управления не приводит к конфликтам на выходах мультиплексоров, так как в любой момент времени активизирован лишь один из них. Задержка информационного сигнала здесь определяется скоростью его прохождения через одну ступень устройства.

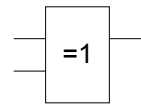
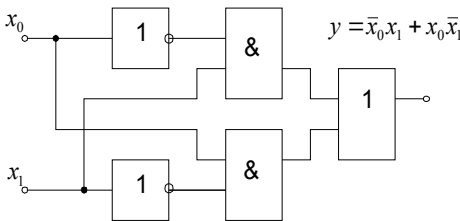
Мультиплексор может быть использован не только по своему прямому назначению, то есть для коммутации цифровых сигналов, но и как универсальный логический элемент, способный реализовать любую функцию от  $n$  логических переменных, где  $n$  - число его управляющих входов. Как уже отмечалось, в отличие от алгебры непрерывных переменных количество логических функций ограничено и для  $n$  переменных составляет  $N = 2^{2^n}$ .

Пусть требуется реализовать функцию логической неравнозначности, таблица истинности которой выглядит следующим образом.

$x_1$	$x_0$	$y$	$\bar{y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Алгебраическая запись этой функции имеет вид  $y = \bar{x}_0 x_1 + x_0 \bar{x}_1$ . Ее можно представить и в несколько иной форме, обратившись к функции логической равнозначности, которая инверсна по отношению к предыдущей. Так как  $\bar{y} = \bar{x}_0 \bar{x}_1 + x_0 x_1$  то  $y = \overline{\bar{x}_0 \bar{x}_1 + x_0 x_1}$ .

Из первого выражения следует, что функцию логической неравнозначности, часто называемую «исключающее ИЛИ», либо «сумма по модулю два» и обозначаемую как  $y = x_0 \oplus x_1$ , можно реализовать посредством такой схемы



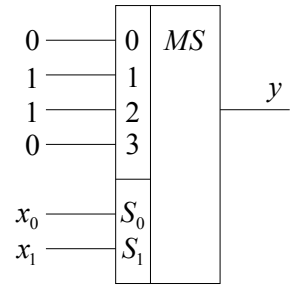
Выпускаемые промышленностью в составе серий микросхем элементы, выполняющие

данную операцию, графически изображаются следующим образом.

$S_1$	$S_0$	$y$	
0	0	$D_0$	(0) 0
0	1	$D_1$	(1) 1
1	0	$D_2$	(1) 1
1	1	$D_3$	(0) 0
$x_1$	$x_0$		$y$

Как следует из таблицы, данный элемент может выполнять и функцию управляемого инвертора. Действительно, при  $x_1 = 0, y = x_0$ , а когда  $x_1 = 1, y = \overline{x_0}$ .

С помощью мультиплексора 4→1 функцию логической неравнозначности можно реализовать следующим образом. Так как при переборе кодовых комбинаций управляющих сигналов  $S_0, S_1$ , на выходах мультиплексора появляются данные с информационных входов  $D_0, D_1, D_2$  и  $D_3$ , то если на них подать сигналы логических нулей и единиц, соответствующие значениям требуемой функции, то на выходе мультиплексора будут формироваться ее значения. При этом переменные  $x_0, x_1$  должны подаваться на входы  $S_0, S_1$ .



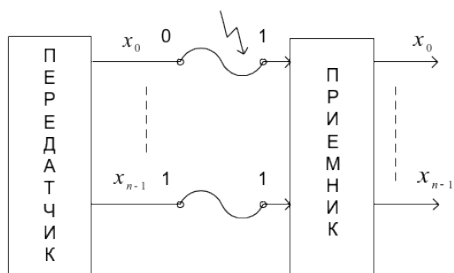
Меняя комбинации сигналов на информационных входах, с помощью мультиплексора можно воспроизвести любую их 16 возможных функций от двух переменных.

С этой точки зрения мультиплексор является универсальным логическим элементом с программируемыми свойствами, так как без изменения конфигурации устройства выполняемая им функция может меняться.

#### 4.5 Схемы контроля четности.

Данные узлы служат для определения четности количества единиц, либо нулей в многоразрядных двоичных словах. Если их число четное, то устройство вырабатывает на выходе определенный сигнал, например логический ноль, в противном случае – единицу. Таким образом, результатом обработки слова 101100011101 будет 1, а для 1100101 – ноль.

Такие устройства находят широкое применение в



системах передачи цифровых данных для обнаружения ошибок.

В процессе передачи данных в системе, состоящей из источника сигнала (передатчика),

приемника и линии

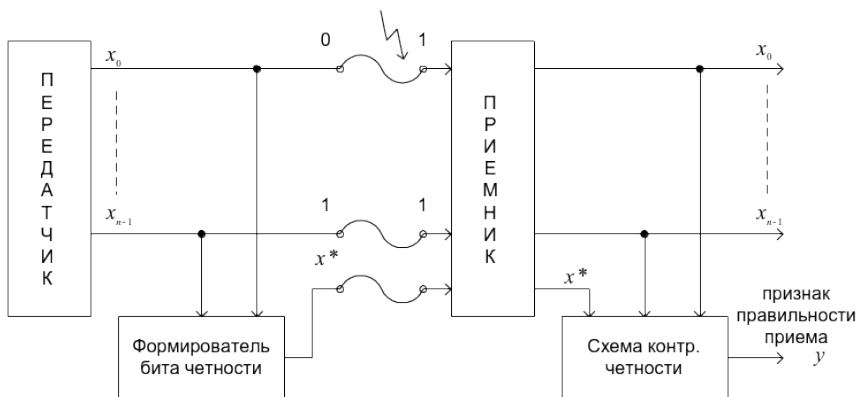
передачи из-за воздействия помех может произойти изменение значения передаваемого бита. То есть по какой-либо линии передается логический ноль, а будет принята единица, либо наоборот, что приведет к искажению полученной информации.

Так как передаваемые данные заранее неизвестны, то приемник воспримет полученную информацию как истинную, что может привести к существенному нарушению работы связанных с ним устройств.

Однако, если появление ошибки в переданном слове удастся обнаружить, то его можно будет передать еще раз и производить эту процедуру до тех пор, пока принятое слово не окажется идентичным переданному.

В этом случае требуется введение критерия, по значению которого на принимающей стороне можно судить о наличии или отсутствии ошибки при передаче. Одним из таких критериев и является четность количества единиц в передаваемом и принятом словах. Если она при передаче не меняется, то с достаточно большой вероятностью можно считать, что и само слово не претерпело трансформации.

Данным способом можно определить одиночные ошибки, либо ошибки сразу в трех, пяти и любом нечетном количестве разрядов. Если инверсия (искажение) значений произойдет у четного их числа, то четность количества единиц в измененном слове сохранится. Так как наиболее



вероятными и частыми являются одиночные ошибки, их обнаружению уделяют особое внимание.

Вариант структуры устройства, выявляющего одиночные ошибки при передаче данных, выглядит следующим образом.

На передающей стороне используется формирователь бита четности (схема контроля четности), вырабатывающий признак  $x^*$ , равный логическому нулю, если количество единиц в передаваемом слове четное и логической единице в противном случае. Этот бит передается вместе со всеми разрядами слова и поступает на приемник, к выходам которого подключается аналогичная схема, проверяющая на четность количество единиц в принятом совместно с дополнительным разрядом слове.

Если число единиц в слове, формируемом передатчиком было четным, то бит четности равен нулю и общее количество единиц в передаваемом слове окажется четным. В случае, когда в исходном слове нечетное число единиц, бит четности примет единичное значение и общее количество единиц в передаваемом слове опять окажется четным.

Таким образом, в отсутствии помех при передаче число единиц в принятом слове всегда четно. При этом схема контроля четности будет формировать признак правильности приема, равный нулю.

Если при передаче произойдет изменение значения (инверсия) одного из разрядов, то четность числа единиц в принятом совместно с разрядом  $x^*$  слове окажется нарушенной. На выходе схемы контроля четности сформируется единичный сигнал, являющаяся признаком возникновения ошибки при передаче, что будет служить основанием для повторной передачи слова принятого с ошибкой.

Как уже отмечалось, когда инверсия возникает в двух разрядах, то четность числа единиц в принятом слове сохраняется, и этап передачи будет определен как безошибочный. Однако вероятность возникновения двух или более ошибок гораздо ниже, чем одиночной.

Синтез схемы контроля четности удобно начать с ее простейшего варианта для двухразрядного слова. Функция, описывающая четность количества единиц в нем, задается такой таблицей.

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

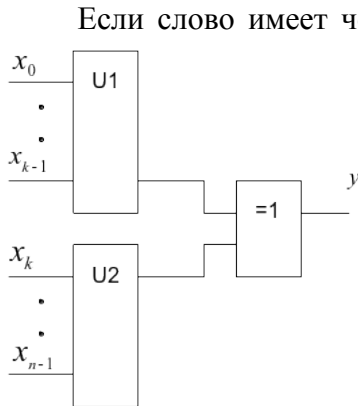
Так как разряды слова можно считать логическими переменными, то эта функция является одной из 16 от двух переменных, в частности – функцией логической неравнозначности. Таким образом, для контроля четности числа единиц в двухразрядном слове требуется использовать сумматор по модулю два, а алгебраически данная процедура записывается следующим образом  $y = \bar{x}_0 \cdot x_1 + x_0 \cdot \bar{x}_1 = x_0 \oplus x_1$ .

Для синтеза схем контроля четности в многоразрядных словах можно задать соответствующую функцию в табличной форме, затем определить ее аналитический вид и разработать схему на стандартных логических элементах. Однако такой способ является достаточно трудоемким, так как с увеличением разрядности слова на единицу, число строк таблицы удваивается. Поэтому в данной ситуации удобнее использовать нестандартный подход.

Пусть имеется  $n$ -разрядное двоичное слово  $x_{n-1} x_{n-2} \dots x_1 x_0$ . Разобьем его на две части  $(x_{n-1} x_{n-2} \dots x_k)$   $(x_{k-1} \dots x_1 x_0)$  и предположим, что имеются устройства  $U_1, U_2$  позволяющие формировать признаки четности количества единиц в них. Очевидно, что четность или нечетность числа единиц в частях слова, определенным образом зависит от ситуации с данным показателем для всего слова

Это можно отразить следующим образом.

Количество единиц в слове	Количество единиц в частях слова	
Четное (0)	Четное (0)	Четное (0)
Четное (0)	Нечетное (1)	Нечетное (1)
Нечетное (1)	Четное (0)	Нечетное (1)
Нечетное (1)	Нечетное (1)	Четное (0)

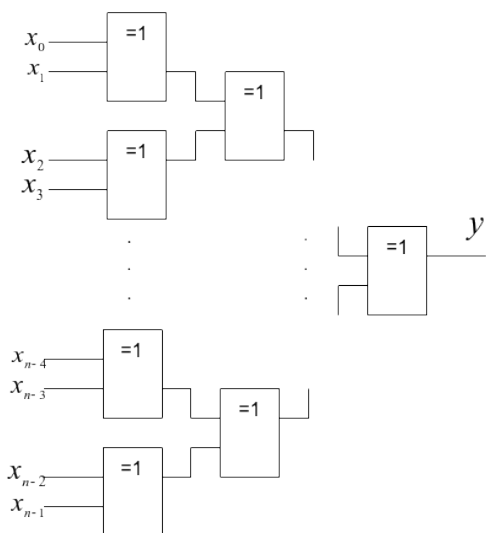


Если слово имеет четное число единиц, то при любом его разбиении в частях слова окажется по четному, либо по нечетному их количеству. При нечетном числе единиц во всем слове, в одной из его частей всегда окажется четное их количество, а в другой нечетное.

В скобках в таблице приведены значения сигналов, формируемых устройствами  $U_1, U_2$  и тех, которые должны

появиться на выходе схемы контроля четности числа единиц во всем слове. Видно, что эта таблица описывает функцию логической неравнозначности над переменными, формируемыми вспомогательными устройствами. Таким образом, структура схемы контроля четности должна иметь следующий вид.





Разбивая части слова на более мелкие компоненты и проводя аналогичные рассуждения, можно прийти к выводу, что в итоге потребуется первоначальный контроль четности количества единиц лишь в двух соседних разрядах слова и дополнительная обработка полученных результатов схемами

логической неравнозначности. Отсюда следует, что один из вариантов принципиальной схемы устройства контроля четности будет выглядеть следующим образом.

При синтезе такого устройства стандартным способом оно получилось бы достаточно громоздким, так как функция, описывающая его работу относится к классу неминимизируемых в базисе И-НЕ, ИЛИ-НЕ.

Промышленностью выпускаются схемы контроля четности в интегральном исполнении, в частности микросхема К155ИП2. Она обрабатывает восьмибитовые слова и имеет входы расширения, позволяющие наращивать разрядность. Аналогичные функции выполняет микросхема К153ЗИП5, работающая с девятиразрядными числами.

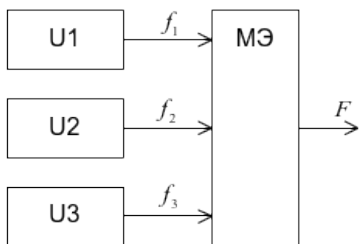
#### 4.6 Мажоритарные элементы.

Еще одним узлом, которое относится к схемам контроля и используется для построения высоконадежных цифровых схем, является мажоритарный элемент.

В любом устройстве возможны нарушения нормальной работы – сбои (ошибки), которые могут возникать как из-за возникновения неисправностей, так и по причинам воздействия помех. Один из методов повышения надежности функционирования цифровых систем заключается в обнаружении и последующем устранении возникающих ошибок.

Однако в ряде случаев их появление недопустимо, то есть вероятность возникновения ошибки должна быть сведена к нулю. В таких ситуациях используется резервирование узлов, блоков и отдельных элементов аппаратуры. То есть несколько устройств включаются в работу параллельно и решают одну и ту же задачу. Если они функционируют правильно, то формируют идентичные результаты. В большинстве ситуаций, в случае нарушения работоспособности или выхода из строя ряда из них, остальные продолжают нормальное функционирование. В этом случае для обеспечения надежной работы системы потребуется некоторый узел, который пропускал бы на ее выход сигналы только с правильно работающих блоков.

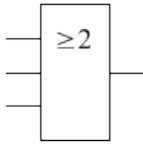
Один из способов решения этой задачи заключается в использовании мажоритарных элементов, работу которых можно описать следующим образом. Пусть в системе используется трехкратное дублирование, то есть имеется три одинаковых узла  $U_1, U_2, U_3$ , формирующих при нормальном функционировании идентичные одноразрядные двоичные сигналы  $f_1, f_2, f_3$ .



При нарушении работы одного из них, его сигнал окажется противоположен формируемым двумя остальными. В этом случае правильное значение выходного сигнала должно совпадать с

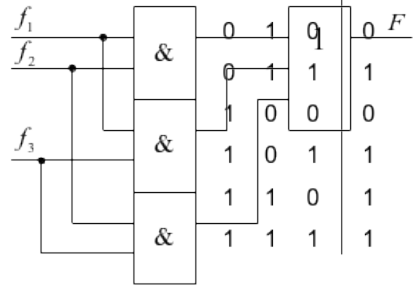
состоянием большинства из устройств.

Таблица, описывающая работу трехвходового мажоритарного элемента, имеет вид. На его выходе формируется сигнал, идентичный большинству из поступивших на входы.



Функция, описывающая работу такого устройства, описывается выражением

$$F = \overline{f_1}f_2f_3 + f_1\overline{f_2}f_3 + f_1f_2\overline{f_3} + f_1f_2f_3.$$



$f_1$	$f_2$	$f_3$	$F$
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	1

Воспользовавшись правилами алгебры логики, либо построив карту Карно, данное выражение можно упростить до  $F = f_1f_2 + f_1f_3 + f_2f_3$ . Внутренняя структура такого элемента имеет вид, а на принципиальных схемах он изображается следующим образом.

Очевидно, что мажоритарный элемент может иметь только нечетное количество входов, 3, 5, 7 и т.п., так как в противном случае при равенстве числа правильных и неверных сигналов, принять решение о выборе невозможно. Кроме того, от самого мажоритарного элемента требуется очень высокая надежность, так как в случае выхода его из строя, вся схема резервирования окажется бесполезной.

Достаточно просто можно создать устройство, которое в схеме с мажоритарным элементом будет формировать двоичный код номера узла, вырабатывающего ложный сигнал. Таблица его функционирования имеет вид. Нулевому коду соответствует правильное функционирование всех узлов, то есть совпадение всех сигналов, в противном случае формируется код номера устройства, состояние выхода которого не совпадает с остальными.

$f_1$	$f_2$	$f_3$	$e_1$	$e_0$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Аналитические выражения для функций, описывающих соответствующие разряды  $e_1, e_0$ , можно получить, используя карту Карно, однако проанализировав структуру таблицы, можно увидеть, что значения  $e_1$  не зависят от  $f_1$  и описываются функцией логической неравнозначности для сигналов  $f_2, f_3$ , то есть  $e_1 = f_2 \oplus f_3$ . Разряд  $e_0$  в первых четырех строках таблицы совпадает с  $f_3$ , а на остальных принимает инверсные по отношению к этой функции значения. При этом прямые реализуются при  $f_1=0$ , а инверсные в случае  $f_1=1$ , независимо от состояния  $f_2$ . Таким образом, для реализации  $e_0$  можно использовать управляемое инвертирование  $f_3$  по единичному значению  $f_1$ . Данная процедура соответствует выполнению функции логической неравнозначности  $e_0 = f_1 \oplus f_3$  и реализуется логическим элементом – сумматором по модулю два.

#### 4.7 Цифровые компараторы.

Компаратором называется устройство для сравнения двух однородных величин, в частности чисел А и В, и формирования признака отношения между ними. В этом случае, не имея информации о самих числах, по сигналам компаратора можно установить:  $A > B$ ,  $A < B$ , либо  $A = B$ .

Цифровые компараторы предназначены для сравнения чисел, представленных в двоичном коде.

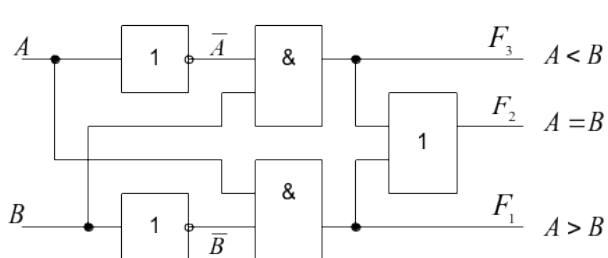
A	B	$F_1$ $A > B$	$F_2$ $A = B$	$F_3$ $A < B$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Работу простейшего из них, компаратора одноразрядных чисел, можно описать

$$F_1 = A\bar{B}$$

$$F_2 = \bar{A}\bar{B} + AB = \overline{A\bar{B} + \bar{A}B}$$

$$F_3 = \bar{A}B$$



следующей таблицей.  
 Такое устройство должно формировать три функции  $F_1, F_2, F_3$ ,

принимаящие единичное значение при соответствующих соотношениях между числами  $A$  и  $B$ .

Алгебраическая форма записи этих функций будет иметь вид, а схема устройства для их формирования, то есть компаратор, выглядит следующим образом.  $F_2$  является функцией логической равнозначности, и так как ее значения инверсны по отношению к функции логической неравнозначности, то возможны две формы записи  $F_2$ .

На основе такой данной ячейки может быть реализована процедуры сравнения чисел любой разрядности.

Пусть имеются два числа  $A$  и  $B$  одинаковой разрядности  $a_{n-1}a_{n-2} \dots a_1a_0$  и  $b_{n-1}b_{n-2} \dots b_1b_0$  и требуется установить отношение между ними. Если разрядность чисел разная, то ее можно выровнять, заполнив старшие разряды более короткого числа нулями. В то же время сравнение таких чисел не имеет смысла, так как если у числа с большей разрядностью, в старших имеется хотя бы одна единица, то оно обязательно будет больше. Это можно продемонстрировать на примере из десятичной системы счисления, если  $A=120$  и  $B=85$  и  $A>B$ . Здесь последнее число можно записать как  $085$ , что не изменит результата.

При анализе отношений между многоразрядными числами на первом этапе сравниваются старшие разряды  $a_{n-1}$  и  $b_{n-1}$ . При этом возможны три ситуации:  $a_{n-1} > b_{n-1}$ ,  $a_{n-1} < b_{n-1}$  и  $a_{n-1} = b_{n-1}$ . Очевидно, что при реализации первых двух,

дальнейшее сравнение можно прекратить, так как, если старший разряд одного числа больше, либо меньше старшего разряда другого, то так же будут соотноситься и числа, например  $\underline{236} > \underline{128}$ ,  $\underline{321} < \underline{480}$ .

В случае, когда старшие разряды одинаковы ( $\underline{456}$  и  $\underline{479}$ ), потребуется перейти к анализу более младших  $a_{n-2}$  и  $b_{n-2}$ . Если они окажутся равными ( $\underline{128}$  и  $\underline{129}$ ), то необходимо сделать еще один шаг и так далее. Таким образом, дольше всего будут сравниваться числа, если они равны. Анализ в одноразрядном компараторе требуется продолжать до появления единичного сигнала на выходах  $F_1$  или  $F_3$ , либо нулевого на  $F_2$ .

Для практической реализации такого устройства кроме одноразрядного компаратора потребуется узел, который осуществляет поочередный ввод разрядов чисел. Такое устройство обладает невысоким быстродействием из-за последовательной процедуры обработки поступающих данных.

Для сокращения времени выполнения процедуры сравнения многоразрядных чисел используются параллельные компараторы. Промышленностью в интегральном исполнении выпускаются микросхемы четырехразрядных устройств (К555СП1, К561ИП2), которые могут соединяться между собой для увеличения разрядности сравниваемых чисел. Такие устройства имеют три стандартных выхода – больше ( $>$ ), равно ( $=$ ) и меньше ( $<$ ), а также дополнительные входы с аналогичными обозначениями.

На принципиальных схемах они изображаются следующим образом, а их таблица функционирования имеет такой вид. Звездочки обозначают произвольные состояния сигналов, то есть либо логический ноль, либо единицу. Из таблицы следует, что данный компаратор кроме отношений  $A > B$ ,  $A < B$ ,  $A = B$ , может фиксировать нестрогие неравенства  $A \geq B$  и  $A \leq B$ . Первое реализуется при комбинации управляющих сигналов 1 0 0, а второе – 0 0 1. В этих случаях на выходах  $A > B$  и  $A < B$  единичный сигнал будет формироваться, когда  $A$  больше, либо равно  $B$ , и соответственно, если  $A$  меньше или равно  $B$ . Для оценки строгих отношений требуется задать набор управляющих сигналов \* 1 \*.

Наращивание разрядности сравниваемых чисел при использовании рассмотренных узлов возможно различными способами, схема реализации одного из них выглядит следующим образом.

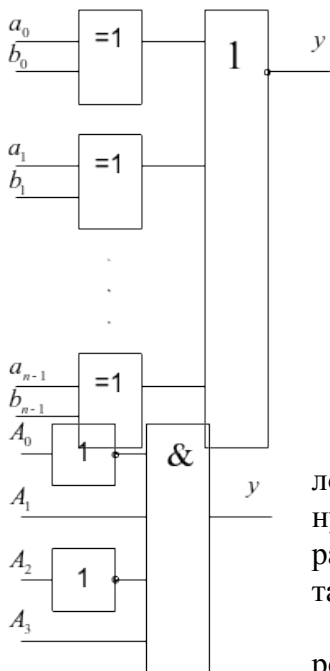


Здесь младшие разряды чисел подаются на левый компаратор, а

старшие на крайний правый. Это связано с тем, что когда старшие разряды не равны, результат сравнения не должен зависеть от значений более младших, что и реализуется в данной схеме, так

Входы				Выходы		
>	=	<	A, B	>	=	<
*	*	*	$A > B$	1	0	0
1	0	0	$A = B$	1	0	0
*	1	*	$A = B$	0	1	0
0	0	1	$A = B$	0	0	1
*	*	*	$A < B$	0	0	1
0	0	0	$A = B$	1	0	1

как при этом выходные сигналы компараторов не зависят от значений управляющих. Такая структура имеет не очень высокое быстродействие, что обусловлено последовательным распространением сигналов управления.



Когда требуется выявить лишь факт равенства чисел, структура параллельного компаратора может быть существенно упрощена, так как в данном случае у чисел должны быть попарно равны все одноименные разряды. Определить, равны они или нет, можно используя схему логической неравнозначности (сумматор по модулю два). Далее необходимо сформировать сигнал логической единицы при наличии нулевых результатов сравнения пар разрядов. Отсюда следует, что схема такого устройства будет иметь вид.

Для ряда приложений требуется решать еще более простую задачу, а именно сравнения числа с некоторой константой. Здесь можно использовать рассмотренные выше многоразрядные компараторы, однако это нецелесообразно из-за значительных аппаратных затрат.

Пусть к примеру, необходимо сформировать сигнал логической единицы, если некоторое число  $A=K$ . То есть требуется создать устройство, на выходе которого появится единичный сигнал, на наборе переменных  $A_i$ , совпадающих с фиксированными значениями разрядов  $K_i$ . Для этой цели можно использовать элемент И с количеством входов равным разрядности чисел, причем разряды  $A_i$ , соответствующие



единичным значениям  $K_i$  необходимо подавать непосредственно, а соответствующие нулевым – в проинвертированном виде. Тогда только при  $A=K$  на все входы элемента И поступят единицы и на выходе также сформируется единичный сигнал. Если, к примеру,  $K=10$  ( $1010_2$ ), то устройство будет выглядеть следующим образом.

### 4.8 Сумматоры двоичных кодов чисел.

$A_0$	$B_0$	$A_0+B_0$	$C_1$	$S_0$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	2	1	0
$2^0$	$2^0$		$2^1$	$2^0$

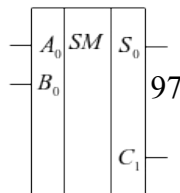
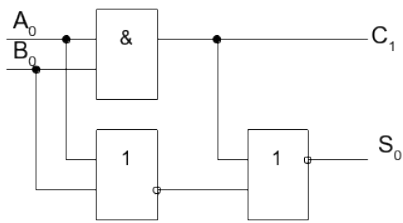
Сумматором называется устройство для арифметического сложения чисел. В

вычислительных машинах они представляются в двоичном коде и таблица истинности простейшего сумматора одноразрядных двоичных чисел выглядит так. Из нее следует, что разрядность суммы совпадает с разрядностью слагаемых если они оба не равны единице. Когда  $A=B=1$  результат сложения получается равным 2 и разрядность суммы становится на единицу больше чем у слагаемых. Это явление называется переполнением разрядной сетки, а появляющаяся в старшем разряде единица – сигналом переноса. Обычно он обозначается буквой  $C$  (от английского слова carry), а нижний индекс характеризует его вес.

Из таблицы следует, что алгебраические представления выходных функций сумматора имеют вид, причем значения сигнала суммы совпадают с функцией логической неравнозначности от операндов. Чтобы избавиться от инверсий входных переменных  $S_0$  удобно представить в

несколько иной форме.

Из полученных



соотношений вытекает, что структура сумматора будет иметь вид. На принципиальных схемах сумматор, как функциональный узел, обозначается следующим образом.

Однако, правильное название данного устройства – полусумматор. У полного одноразрядного сумматора должен быть дополнительный вход для обработки сигнала переноса  $C_0$ , имеющего тот же вес, что и разряды операндов. Это необходимо при построении многоразрядных сумматоров, так как появляющиеся переносы требуется учитывать при суммировании более старших разрядов.

При возникновении переноса  $C_1 = 1$ , а при его отсутствии  $C_1 = 0$ . Однако для удобства считают, что сигнал переноса присутствует всегда, но в зависимости от ситуации имеет либо нулевое, либо единичное значение. В качестве примера можно привести процедуру сложения чисел в десятичной системе счисления.

$$\begin{array}{r}
 + 18 \\
 + 13 \\
 \hline
 31 \quad C=1
 \end{array}
 \quad
 \begin{array}{r}
 + 16 \\
 + 13 \\
 \hline
 29 \quad C=0
 \end{array}$$

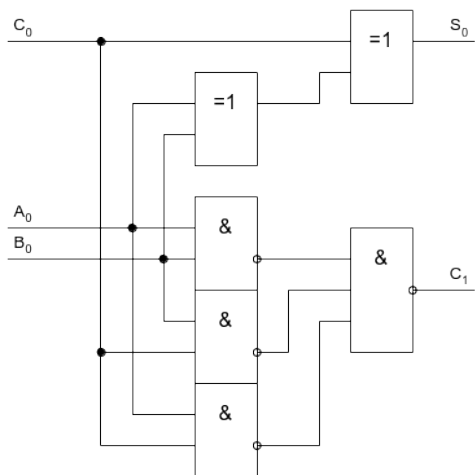
представления функций  $S_0$  и  $C_1$  воспользоваться Карно. Из них можно установить, что  $S_0$  неминимизируема.

Так как все суммируемые разряды имеют одинаковые веса, то таблица истинности полного одноразрядного сумматора имеет следующий вид. Для получения алгебраического

$C_0$	$A_0$	$B_0$	$S_0$	$C_1$
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
$2^0$	$2^0$	$2^0$	$2^0$	$2^1$

$S_0$	$A_0 B_0$	$\bar{A}_0 B_0$	$\bar{A}_0 \bar{B}_0$	$A_0 \bar{B}_0$	выходных функций
$C_0$	1	0	1	0	удобно картами
$\bar{C}_0$	0	1	0	1	

$C_1$	$A_0 B_0$	$\bar{A}_0 B_0$	$\bar{A}_0 \bar{B}_0$	$A_0 \bar{B}_0$	функция
$C_0$	1	1	0	1	функция
$\bar{C}_0$	1	0	0	0	



Однако, выражение для нее можно преобразовать таким образом, чтобы при представлении  $S_0$  использовать операцию суммирования по модулю два (функцию логической неравнозначности). В этом случае запись получается достаточно компактной и схема

сумматора будет иметь вид.

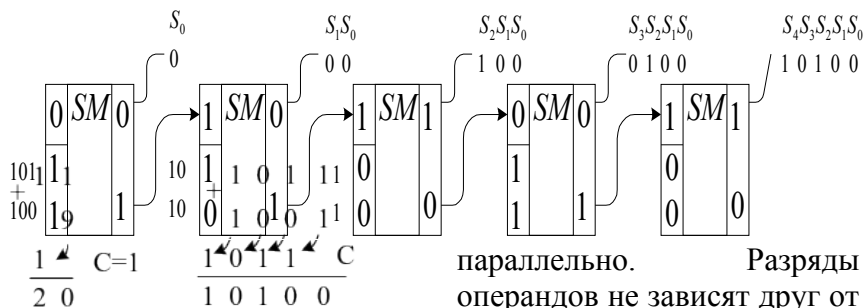
Структура ее верхней части представляет собой схему контроля четности. Если проанализировать по таблице свойства функции  $S_0$ , то можно увидеть, что она действительно принимает единичное значение, когда количество единиц в трехразрядной комбинации  $C_0, A_0, B_0$  нечетное.

Промышленностью в составе серий цифровых микросхем выпускаются одноразрядные сумматоры – К155ИМ1, К555ИМ5 (два в одном корпусе) и ряд других.

$$\begin{aligned}
 S_0 &= A_0 B_0 C_0 + \bar{A}_0 \bar{B}_0 C_0 + \bar{A}_0 B_0 \bar{C}_0 + A_0 \bar{B}_0 \bar{C}_0 = \\
 &= C_0 (A_0 B_0 + \bar{A}_0 \bar{B}_0) + \bar{C}_0 (\bar{A}_0 B_0 + A_0 \bar{B}_0) = \\
 &= C_0 (\overline{A_0 \oplus B_0}) + \bar{C}_0 (A_0 \oplus B_0) = \\
 &= C_0 \oplus A_0 \oplus B_0
 \end{aligned}$$

$$\begin{aligned}
 C_1 &= A_0 B_0 + C_0 A_0 + C_0 B_0 = \\
 &= \overline{A_0 B_0} \cdot \overline{C_0 A_0} \cdot \overline{C_0 B_0}
 \end{aligned}$$

В цифровых системах обработки информации требуется осуществлять процедуру сложения многоразрядных чисел. Ее можно реализовать либо последовательным образом, либо



параллельно. Разряды операндов не зависят друг от друга, поэтому процесс их

суммирования сводится к сложению разрядов, имеющих одинаковые веса. Если при этом возникает перенос, он должен быть добавлен к сумме следующих по старшинству разрядов, как это показано на примере суммирования чисел в десятичной и двоичной системах счисления.

Отсюда следует, что для сложения чисел произвольной разрядности можно использовать одноразрядный сумматор, подавая разряды на обработку последовательно друг за другом и учитывая сигналы переносов.

В этом случае, процедура сложения чисел 11 и 9, представленных двоичными кодами 1011 и 1001 должна происходить так, как показано на рисунке, где представлен один и тот же сумматор, в разные моменты времени. На первом этапе на него поступают младшие разряды операндов и нулевой сигнал переноса. Младший разряд суммы, равный нулю, может быть считан с соответствующего выхода  $S_0$ , а единичный сигнал переноса должен быть прибавлен на следующем шаге к значениям соответствующих разрядов операндов. При этом сформируется разряд суммы  $S_1$  и так далее.

Так как разрядность суммы на единицу больше разрядности слагаемых, то для считывания с выхода  $S_0$

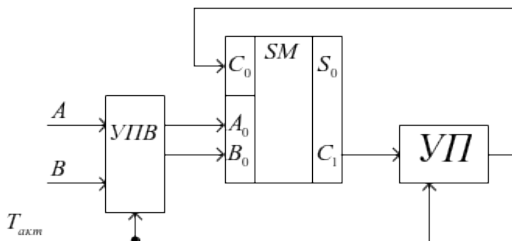
последнего разряда суммы  $S_i$  потребуется дополнительный этап, на котором сигнал переноса необходимо передать на выход  $S_0$ . Для этого его надо просуммировать с двумя нулями.

Процесс сложения  $n$ -разрядных чисел в таком сумматоре займет  $N+1$  шаг. Если среднюю задержку формирования выходных сигналов сумматора  $SM$  принять за  $\tau$ , то время, требуемое для выполнения процедуры сложения может быть оценено как  $(N+1)\tau$ .

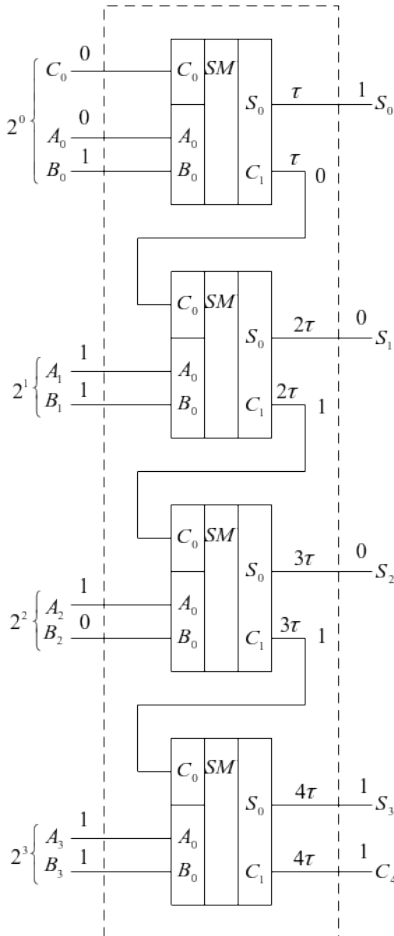
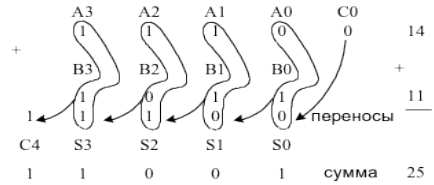
Такой сумматор должен содержать узел для последовательного ввода разрядов слагаемых (УПВ), а также устройство для запоминания на один такт разряда переноса (УП). Сам сумматор является комбинационным устройством, то есть при поступлении новых данных сразу же изменится и результат обработки, а так как сигнал переноса требуется в следующем такте, то его необходимо запомнить.

Несмотря на относительную простоту, существенным недостатком такого сумматора является его низкое быстродействие, поэтому подобные узлы находят применение там, где скорость вычислений не играет существенной роли, например в калькуляторах.

При работе с большими массивами цифровых данных, скорость вычислений становится очень важной, и в этом случае используются параллельные методы обработки.



Так как в ходе суммирования разряды связаны только через сигналы переносов, то структура параллельного



связано с тем, что здесь осуществляется обработка арифметических данных и поэтому значение разряда определяется не только его состоянием (0,1), но и местоположением, то есть весом. Отсюда следует, что для

двоичного сумматора на примере четырехразрядного устройства может быть представлена следующим образом.

Здесь разряды операндов поступают одновременно (параллельно) и сигналы переносов с предшествующих разрядов подаются на соответствующие входы сумматоров, обрабатывающих последующие. Такая структура называется параллельным сумматором с последовательным распространением переносов.

Проверить правильность работы данного устройства можно на примере сложения двоичных кодов чисел 14 и 11.

Несмотря на идентичность одноразрядных сумматоров, их функции несколько отличаются. Это

получения правильного результата разряды данных на такой сумматор требуется подавать в строго определенном порядке.

Быстродействие этой схемы можно оценить, зная время задержки появления сигнала суммы  $\tau_s$  и переноса -  $\tau_c$ . В первом приближении их можно считать равными  $\tau$ . Тогда условная диаграмма процедуры сложения будет выглядеть следующим образом.

Через время  $\tau$  после прихода операндов сформируется правильное значение нулевого разряда суммы  $S_0$  и переноса в первый разряд  $C_1$ . Остальные разряды суммы и переносов, отмеченные штрихами, могут иметь неверные значения, так как к этому моменту правильное значение  $C_1$  только поступит на соответствующий вход второго сумматора и на его обработку совместно с разрядами  $A_1, B_1$  также понадобится интервал времени  $\tau$ .

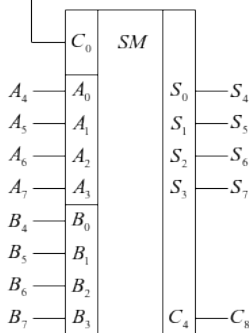
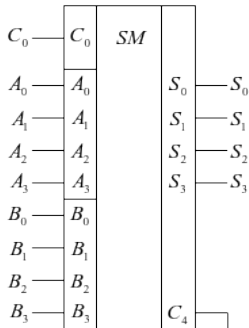
Таким образом через  $2\tau$  истинными будут значения

$C_0 = 0$				
$A_0 B_0$	$S_0 C_1$	$S_0, C_1$	$S_0, C_1$	$S_0, C_1$
$A_1 B_1$	$S'_1 C'_2$	$S_1, C_2$	$S_1, C_2$	$S_1, C_2$
$A_2 B_2$	$S'_2 C'_3$	$S''_2, C''_3$	$S_2, C_3$	$S_2, C_3$
$A_3 B_3$	$S'_3 C'_4$	$S'''_3, C'''_4$	$S''_3, C''_4$	$S_3, C_4$

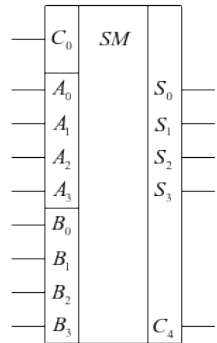
разрядов суммы  $S_0, S_1$ , через  $3\tau$  -  $S_0, S_1, S_3$  и т. д. Отсюда следует, что несмотря на параллельное (одновременное) поступление всех разрядов операндов, правильное значение суммы может быть считано лишь через  $4\tau$ . В общем случае этот интервал зависит от разрядности суммируемых чисел и среднее время задержки такого устройства равно  $t_3 = n\tau$ , где  $n$  - количество разрядов слагаемых. Наличие множителя  $n$

объясняется тем, что перенос на сумматор, обрабатывающий последний разряд, поступает пройдя  $n - 1$  последовательно соединенных аналогичных узлов.

На принципиальных схемах четырехразрядный сумматор изображается следующим образом. Такие устройства можно использовать для построения сумматоров большей разрядности, соединяя их соответствующими выходами и входами переносов, как это показано для восьмиразрядного варианта.



Для суммирующих блоков с внутренним последовательным переносом такой способ соединения является единственно возможным. При этом задержка формирования результата будет возрастать пропорционально их количеству. В общем случае, если использовать  $k$  штук  $n$  - разрядных блоков, то время задержки сложения  $N = kn$  разрядных чисел будет определяться



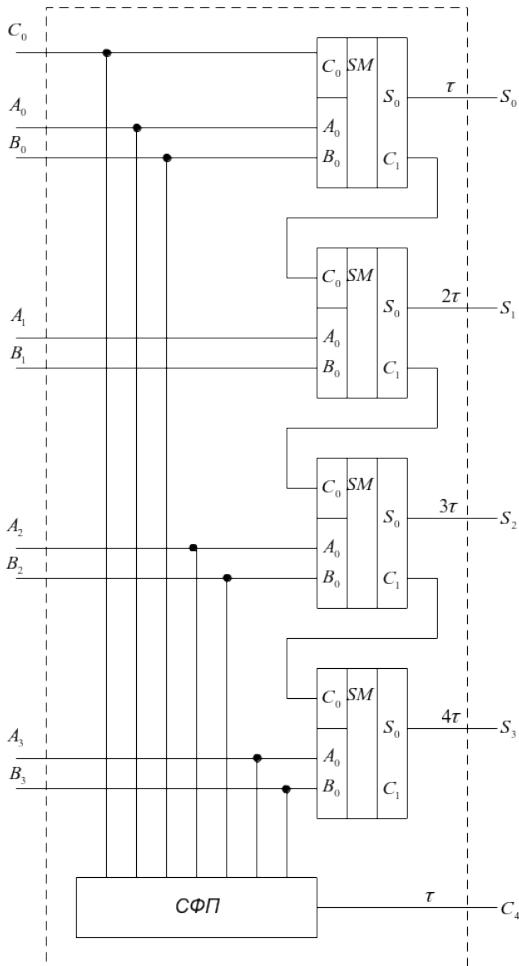
соотношением  $t_3 = kn\tau$ .

Повысить быстродействие многоразрядного сумматора можно, ускорив процесс формирования и передачи сигналов переноса по разрядам.



Пусть имеется устройство – схема формирования переноса (СФП), которое по результатам анализа разрядов операндов, способно сформировать сигнал выходного переноса с задержкой  $\tau$ . Структура четырехразрядного сумматора будет в этом случае выглядеть следующим образом.

Однако к повышению быстродействия суммирующего блока данное решение не приведет, так как несмотря на то,



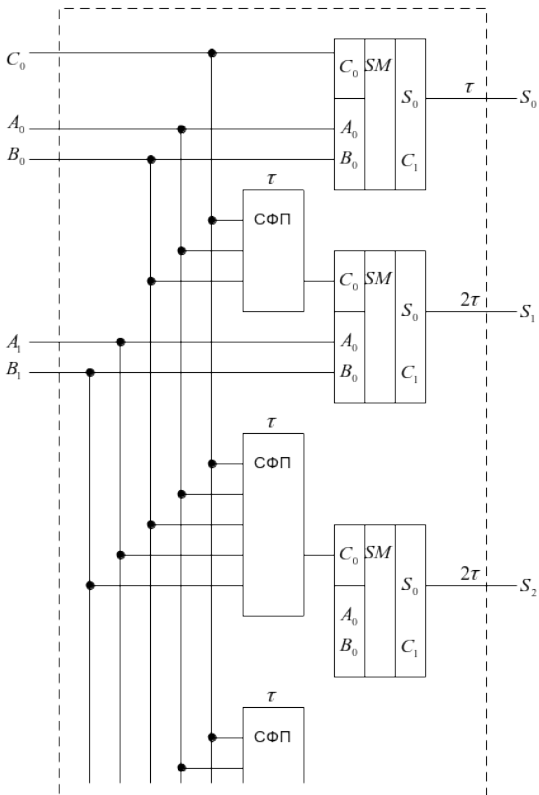
что перенос  $C_4$  появится с задержкой  $\tau$ , результат сложения сформируется не ранее, чем через  $4\tau$ , ( $n\tau$  в суммирующем блоке соответствующей разрядности).

В то же время при последовательном соединении  $k$  таких блоков время сложения будет определяться соотношением  $t_3 = (k - 1)\tau + n\tau$ . За  $(k - 1)\tau$  перенос достигнет последнего блока и  $n\tau$  понадобится для завершения процедуры сложения старших разрядов операндов.

Если аналогичные схемы формирования переносов подключить ко входам переносов всех одноразрядных сумматоров внутри блока, то задержка формирования суммы станет равной  $2\tau$  при любой его разрядности.

Для структуры из  $k$  штук  $n$  - разрядных блоков она составит  $t_s = (k - 1)\tau + 2\tau = (k + 1)\tau$  то есть быстродействие такого многоразрядного сумматора окажется существенно выше, чем у предыдущих вариантов.

Можно ли создать устройства, формирующие переносы

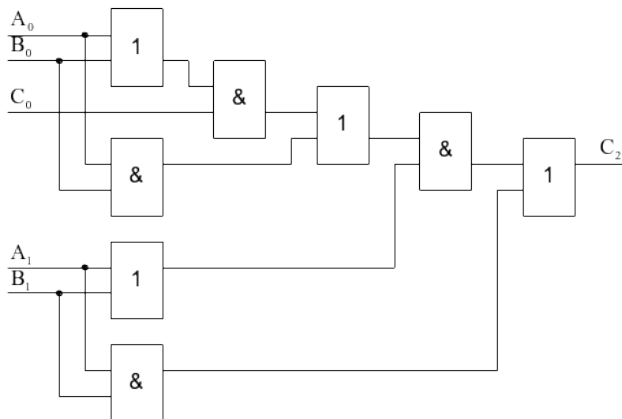


в старшие разряды, не дожидаясь промежуточных результатов сложения? Сигнал переноса, одноразрядного сумматора описывается соотношением

$C_1 = A_0B_0 + C_0(A_0 + B_0)$  Если аналогичное устройство применяется для сложения следующих по старшинству

разрядов, то в данной формуле изменятся индексы, то есть  $C_2 = A_1B_1 + C_1(A_1 + B_1)$ .

Это выражение можно преобразовать следующим образом



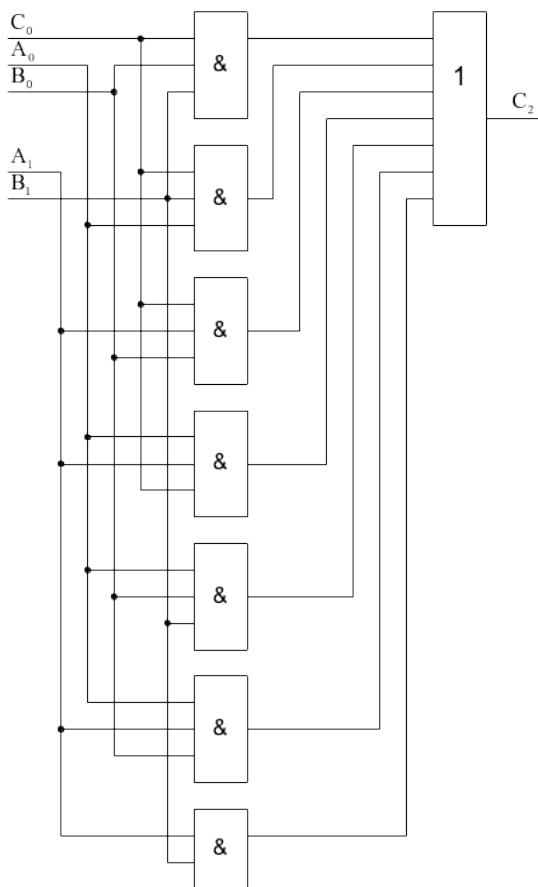
$C_2 = A_1B_1 + (A_1 + B_1)(A_0B_0 + C_0(A_0 + B_0))$ , то есть перенос  $C_2$  можно получить не дожидаясь сигнала  $C_1$ , а обработав все разряды операндов и сигнал входного переноса  $C_0$ .

При такой форме представления функции переноса выигрыша в скорости формирования  $C_2$  не будет, так как из-за последовательной процедуры обработки задержка появления выходного сигнала составит  $5\tau_{лД}$  или примерно  $2\tau$ , если под  $\tau$  понимать задержку в сумматоре.

Однако если это выражение преобразовать, в частности раскрыть скобки, то оно трансформируется к виду

$$C_2 = A_1B_1 + A_0A_1B_0 + A_0B_0B_1 + A_0A_1C_0 + B_0A_1C_0 + A_0B_1C_0 + B_0B_1C_0.$$

Реализация функции  $C_2$  в новом представлении позволит осуществить параллельную обработку компонент, что существенно уменьшит время формирования сигнала переноса (до  $2\tau_{лД}$ ). Аналогичным образом решалась задача повышения быстродействия приоритетного шифратора.



Сумматоры с такой внутренней структурой называются сумматорами с параллельным переносом. Однако их каскадирование возможно только путем соединения выхода переноса предыдущего блока с соответствующим входом последующего, то есть минимальная задержка формирования результата будет равна  $(k + 1)\tau$ .

Если подобные формироваватели переносов устанавливать

между суммирующими блоками, то при теоретической задержке в  $3\tau$  структура сумматора из  $n$  блоков будет нерегулярной, так как каждой последующей схеме придется обрабатывать весь предшествующий набор операндов. То есть они должны иметь разную конфигурацию и, кроме того, резко возрастет количество связей между элементами такого многоуровневого сумматора.

Повысить быстродействие при сохранении регулярности структуры оказалось возможным после разработки сумматоров с ускоренным переносом. В них схемы формирования переносов выносятся за суммирующие блоки,

а с целью уменьшения числа связей на сумматоры возлагается частичная обработка сигналов переносов.

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

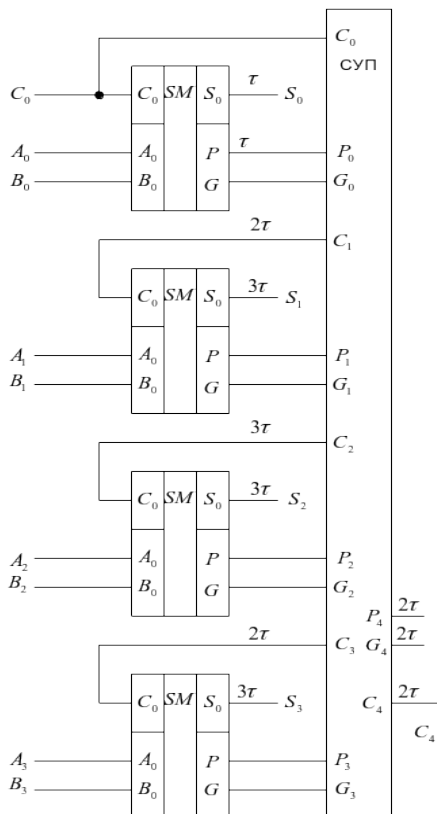
$C_i = A_i B_i + (A_i + B_i) C_{i-1}$ , описывающая сигнал переноса в  $i$ -том разряде представляется в виде  $C_i = G_i + P_i C_{i-1}$ .

Компонента  $G_i = A_i B_i$  называется функцией генерации, а  $P_i = A_i + B_i$  - распространения переноса и задача их

формирования возлагается на одноразрядные сумматоры.

Система уравнений, описывающая работу схемы ускоренного формирования переносов для четырех разрядов сумматора, выглядит следующим образом, а структура сумматора имеет вид.

Последнее выражение представляет сигнал переноса в следующий суммирующий блок и может быть записано как  $C_4 = G_4 + P_4 C_0$ , где



$G_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$ , а  $P_4 = P_3P_2P_1P_0C_0$ . Это

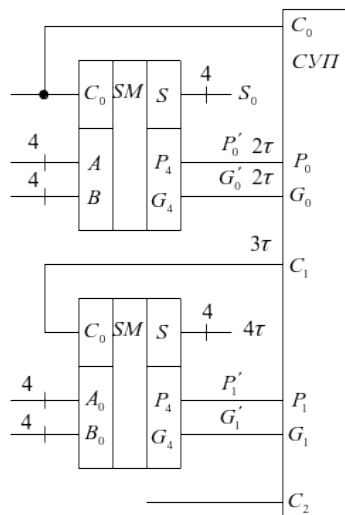
позволяет рассматривать каждый такой блок как эквивалент некоторого сумматора со стандартными значениями функций распространения и генерации переноса, что дает возможность объединять их в многоразрядную структуру посредством аналогичных схем ускоренного переноса (СУП).

Задержка формирования сигнала в представленном четырехразрядном сумматоре составит  $3\tau$  (по  $\tau$  на формирование функций  $P, G$ , переносов во все разряды и самого результата).

Время задержки в шестнадцатиразрядном сумматоре, четырехразрядные блоки объединяются через схемы ускоренного переноса, составит  $4\tau$ , а в шестидесятичетырехразрядной структуре  $5\tau$ .

Промышленностью выпускается большой ассортимент микросхем сумматоров в интегральном исполнении: К155ИМ3 – четырехразрядный параллельный сумматор с последовательными переносами, К555ИМ6 – аналогичный сумматор, с параллельными переносами между разрядами, К155ИП3 – многофункциональное устройство, включающее четырехразрядный сумматор с ускоренным переносом, К155ИП4 – четырехразрядная схема ускоренного переноса.

В приведенной таблице представлены результаты сравнения быстродействия 16 разрядных параллельных сумматоров, построенных на основе четырехразрядных блоков.



N=16, n=4, k=4 Структура	Время задержки теоретическое	Время задержки реальное	Тип микросхем
С последовательным переносом	16τ	180 нС	K155ИМ3
С паралл. переносом в сумматоре и последоват. между блоками	5τ	75 нс	K555ИМ6, K155ИП3
С ускоренным переносом	4τ	50 нС	K155ИП3 K155ИП4

#### **4.9 Устройства вычитания двоичных кодов чисел.**

Операцию вычитания в любой позиционной системе счисления можно представить как сложение уменьшаемого с отрицательным вычитаемым  $15-7=15+(-7)$ . То есть, процедуру вычитания двоичных кодов чисел можно реализовать с использованием сумматоров. Однако при этом необходимо каким-либо образом представлять в двоичном коде отрицательные числа.

Так как цифровые логические элементы воспринимают два уровня сигналов, один из которых обозначается нулем, а другой – единицей, то знак числа может быть закодирован лишь одним из них. В частности, в разрядную сетку добавляется знаковый разряд, который располагается правее самого старшего и принимает нулевое значение, если число положительное и единичное, когда оно меньше нуля.

Формально с кодами знаков могут проводиться те же действия, что и с разрядами числа, однако их результат требуется дополнительно интерпретировать. Как уже отмечалось, кроме двоичного кода для представления чисел используются инверсный (обратный) и дополнительный.

Инверсный образуется путем преобразования нулей во всех, кроме знакового, разрядах в единицы и наоборот, а дополнительный получается из инверсного арифметическим прибавлением к его младшему разряду единицы. Таким образом, запись прямого кода числа +12 будет выглядеть как 0 1100, инверсного - 0 0011, а дополнительного – 0 0100 (здесь при прибавлении единицы образуются переносы в старшие разряды).

Последний вид кода называется дополнительным потому, что число которое им представлено дополняет исходное до ближайшей старшей степени основания системы счисления. В приведенном примере последняя запись соответствует числу 4, которое дополняет исходное 12 до  $2^4$ , то есть до 16. Дополнительный код может быть сформирован в системе счисления с любым основанием. К примеру, в десятичной дополнительный код 13 будет равен 87, так как  $87=100-13$ .

Используя понятие дополнительного кода, операцию вычитания можно заменить сложением уменьшаемого с дополнительным кодом вычитаемого, если не учитывать переполнение разрядной сетки.

Например:  $64-13 \rightarrow 64+87=64+(100-13)=64-13+100 \rightarrow 64-13=51$ .  
 Когда вычитаемое больше уменьшаемого, то ситуация будет следующей  $13-64 \rightarrow 13+36=13+(100-64)=100-(64-13)=49$ . То есть, если результат отрицательный, он оказывается представленным в дополнительном коде.

Аналогичные примеры можно привести и для двоичных кодов чисел со знаком.

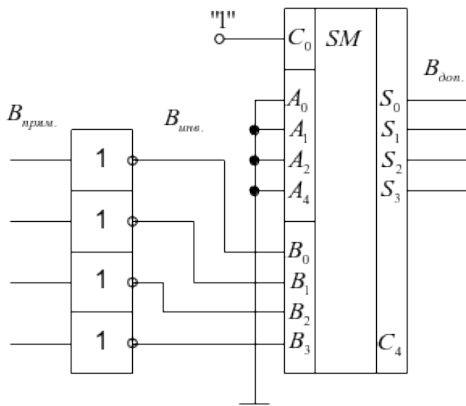
$$13-5=(+13)+(-5)=+8$$

Дополнительный код 5 это прямой код  $\frac{1\ 1011}{1\ 0\ 1000}$  одиннадцати, то есть комбинация 1011. В предложенном примере наблюдается перенос из знакового разряда, который не учитывается.

$$6-13=(+6)+(-13)=-7$$

$$\begin{array}{r} 0\ 0110 \\ 1\ 0011 \\ \hline 1\ 1001 \end{array}$$



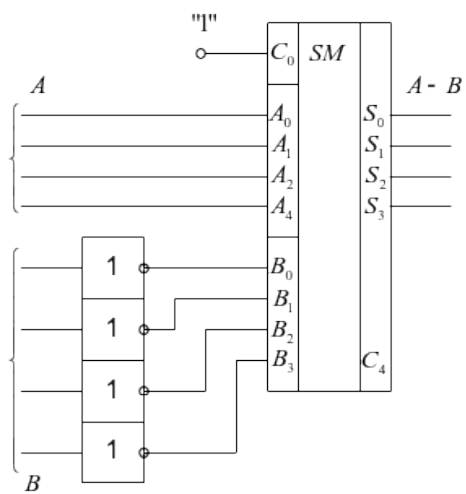


Дополнительный код 13 равен 0011. В данном случае результат оказался равным  $-9$ , но так как он отрицателен, то представлен в дополнительном коде. При этом числу 9 будет соответствовать 7 в прямом.

Таким образом, чтобы реализовать

устройство для вычитания двоичных кодов чисел, потребуется сумматор соответствующей разрядности и преобразователь прямого кода в дополнительный. Его синтез можно провести, используя алгоритм последовательного получения дополнительного кода из прямого, то есть на первом этапе инвертирование разрядов, а затем прибавление к младшему единицы. Схема такого устройства выглядит следующим образом.

Здесь операнд  $A$  равен нулю, а единица прибавляется через разряд переноса  $C_0$ , хотя это можно сделать и через вход  $A_0$ . Разрядность сумматора в таком преобразователе должна быть равна разрядности числа, так как для некоторых кодов, к примеру  $B = 0000$ ,



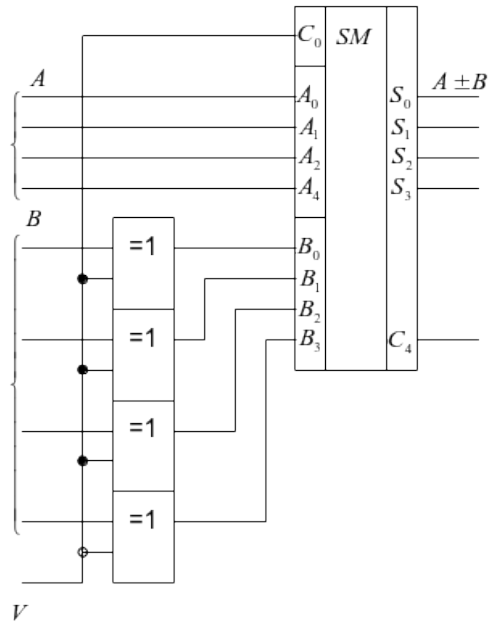
( $B_{инв.} = 1111$ ) и возникающий в младшем разряде перенос будет перемещаться по всем остальным вплоть до старшего.

Этот же узел может быть использован и для преобразования

дополнительного кода в прямой, так как если  $A^*$  дополняет  $A$  до некоторого числа, то верно и обратное утверждение, что  $A$  дополняет  $A^*$  до этого же числа.

В данном устройстве при преобразовании кодов реализуется операция  $0 + (1 + \overline{B})$ . Если на входы  $A_i$  сумматора вместо нулей подавать произвольные числа, то оно будет выполнять операцию вычитания  $A - B$ , формируя положительный результат в прямом коде, а отрицательный в дополнительном. Хотя знаковые разряды здесь не представлены, но информация о знаке результата формируется на выходе переноса  $C_4$  в инверсном виде.

Данную схему можно превратить в универсальный суммирующе-вычитающий узел. Если на входы  $B_i$  подать прямой код числа и установить сигнал входного переноса равным нулю, то выполнится операция  $A+B$ , а если инверсный и единичное значение переноса, то  $A-B$ . Для



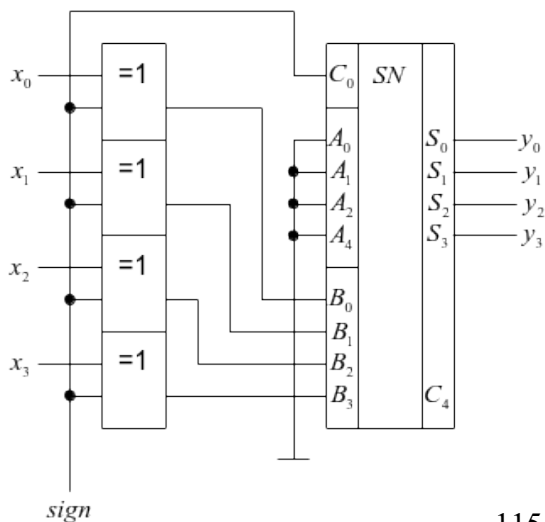
реализации такого устройства потребуется, при нулевом к примеру, значении управляющего сигнала  $V$ , передавать входной операнд без изменений, а если  $V = 1$  инвертировать его разряды.

Эту функцию может выполнить управляемый инвертор, реализованный на сумматорах по модулю два, и схема суммирующе-вычитающего блока будет иметь следующий вид.

Используя управляемые инверторы, можно создать устройство автоматически преобразующее результат вычитания в прямой код в прямой. Здесь, если значение сигнала  $sign = 0$ , то преобразования кода не происходит, а при  $sign = 1$  дополнительный код, которым представляется отрицательный результат вычитания, трансформируется в прямой.

#### 4.10 Сумматоры двоично-десятичных кодов.

Цифровые данные, предназначенные, в частности, для систем отображения информации часто представляются и обрабатываются в кодах, отличных от двоичного, например в двоично-десятичном. Его особенность в том, что каждая цифра десятичного числа преобразуется в четырехразрядную двоичную комбинацию (тетраду) независимо от других.



Так как цифр десять, то некоторые комбинации, имеющиеся в четырехразрядном двоичном коде, в двоично-десятичном будут отсутствовать. Как следует из таблицы, для чисел в пределах от 0 до 9 они совпадают, а далее начинаются отличия, связанные как со структурой тетрад, так и с формированием переносов в старшие разряды кодов.

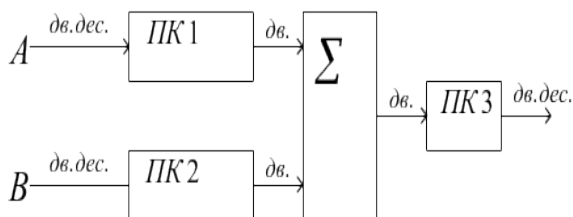
<i>число</i>	<i>дв. код</i>	<i>дв. дес. код</i>
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	1 0000
11	1011	1 0001
12	1100	1 0010
13	1101	1 0011
14	1110	1 0100
15	1111	1 0101
16	1 0000	1 0110
17	1 0001	1 0111
18	1 0010	1 1000
19	1 0011	1 1001
20	1 0100	10 0000

В стандартных позиционных кодах каждый разряд имеет свой вес, равный соответствующей степени основания

системы счисления. В двоично-десятичном коде, используется двойное взвешивание. Одноименные разряды всех тетрад имеют одинаковые веса 8,4,2 и 1, а самим тетрадам приписываются веса кратные степеням 10.

десятичный код		1	7	5								
веса разрядов		$10^2$	$10^1$	$10^0$								
двоичный код		1	0	1	0	1	1	1	1			
веса разрядов		$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$			
двоично - дес. код	0	0	0	1	0	1	1	1	0	1	0	1
веса разрядов	$2^3$	$2^2$	$2^1$	$2^0$	$2^3$	$2^2$	$2^1$	$2^0$	$2^3$	$2^2$	$2^1$	$2^0$
веса тетрад	$10^2$				$10^1$					$10^0$		

Суммировать числа, представленные в двоично-десятичном коде и получать аналогичный результат, можно используя стандартные подходы, либо уже имеющиеся



технические решения. В частности, если преобразовать двоично-десятичные коды операндов в

двоичные, то для их сложения можно использовать двоичные сумматоры, а результат трансформировать в двоично-десятичный код с помощью дополнительного преобразователя. Однако такое решение получается громоздким.

Если создать сумматор двоично-десятичных кодов для одной тетрады, то объединяя такие узлы по входам и выходам переноса, можно наращивать разрядность обрабатываемых операндов. Так как комбинации двоично-десятичного кода тетрады совпадают с соответствующими для двоичного, то их можно складывать в четырехразрядном двоичном сумматоре.

Результат будет верным, пока значение суммы не превысит 9, то есть вплоть до кодовой комбинации 1001. Далее на выходах сумматора начнут формироваться кодовые комбинации, которых не существует в двоично-десятичном коде. Кроме того, пока результат лежит в пределах от 10 до

15 переноса в двоичном сумматоре нет, а в двоично-десятичном он должен появиться.

Таким образом, здесь потребуется коррекция результата. При сложении, к примеру, чисел 7 и 5 (их коды 0111 и 0101) в двоичном сумматоре получится код числа 12 (1100). В двоично-десятичном коде результат должен выглядеть как 1 0010, то есть как 10 и 2 по отдельности. Такое преобразование можно осуществить, если из исходного числа вычесть 10 и сформировать сигнал переноса в следующую тетраду.

Для реализации корректора потребуется устройство анализа выходного кода двоичного сумматора, которое формировало бы, к примеру, сигнал равный нулю, если код меньше или равен 9, и единичный в остальных случаях. Здесь можно использовать параллельный четырехразрядный

S	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	y
0	0	0	0	0	0
1	0	0	0	1	0
·					·
·					·
·					·
8	1	0	0	0	0
9	1	0	0	1	0
<hr/>					
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

цифровой компаратор, но целесообразнее построить специальное устройство. Оно окажется проще, так как в данной ситуации требуется сравнивать код с конкретным числом.

Таблица, определяющая связь между выходными сигналами сумматора и функцией  $y$  анализатора имеет вид.

Из нее следует, что  $y=1$ , если  $S_3$  и  $S_2$ , либо  $S_3$  и  $S_1$  равны единице. В остальных случаях искомая функция обращается в нуль. Таким образом она имеет вид  $y = S_3S_2 + S_3S_1$ .

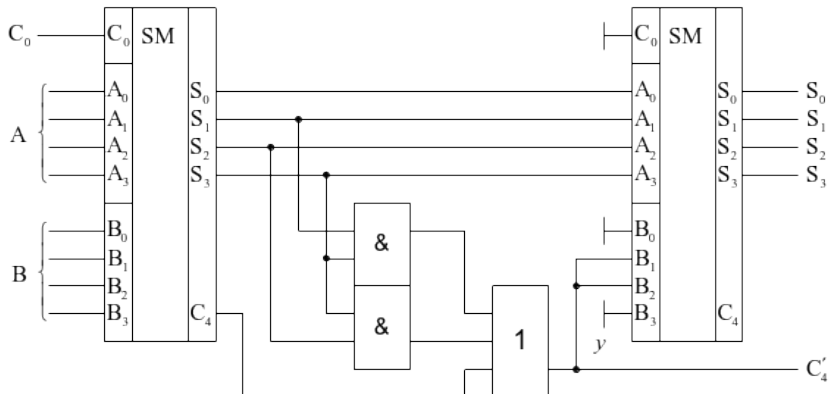
Чтобы провести коррекцию необходимо в случае, когда  $y=1$  вычесть из кода, сформированного

сумматором число 10. С этой целью потребуется прибавить к его результату дополнительный код десяти, то есть число 6 (кодвая комбинация 0110). Таким образом, в схеме потребуется дополнительный двоичный сумматор, в котором, если  $y=0$  исходный результат складывался бы с нулем, а при  $y=1$  - с шестеркой.

Сравнивая коды этих чисел (0110 и 0000), можно сделать вывод, что на входы  $B_0$  и  $B_3$  второго сумматора должны быть поданы логические нули, а на входы  $B_1$  и  $B_2$  сигнал, соответствующий значению функции  $y$ .

Результат сложения двоично-десятичных кодов может оказаться больше 15. Например, при сложении чисел 9 и 8, сумма получается равной 17. В двоичном сумматоре при этом появится сигнал переноса и на его выходах сформируется кодовая комбинация 1 0001.  $C_4=1, S_3=0, S_2=0, S_1=0, S_0=1$ , и результат не будет скорректирован, хотя это требуется.

Процедура в данной ситуации проводится следующим образом. Двоичный сумматор представляет число 17 как  $16+1$ , но  $16=10+6$ , а  $17=10+6+1$ . Наличие 10 говорит о том, что должен сформироваться сигнал переноса в следующую тетраду, а кроме того, к результату работы первого сумматора требуется прибавить 6. Это можно сделать, принудительно установив при возникновении переноса  $C_4$  единичное



значение функции  $Y$ , для чего ее необходимо представить в виде  $y = S_3S_2 + S_3S_1 + C_4$ . При этом схема сумматора двоично-десятичных кодов чисел будет выглядеть следующим образом.

#### ***4.11 Арифметико-логические устройства.***

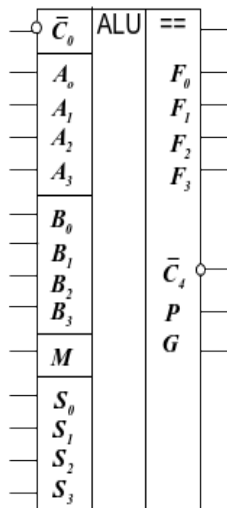
Центральной частью процессора любой цифровой вычислительной машины от простейших контроллеров до современных персональных компьютеров является арифметико-логическое устройство - АЛУ. Его основой служит многоразрядный сумматор, схема которого дополнена логикой, расширяющей функциональные возможности АЛУ и обеспечивающей перестройку с одной операции на другую.

Она осуществляется путем подачи соответствующего набора управляющих сигналов, называемого командой. По одной команде АЛУ может складывать числа, по другой вычитать, по третьей выполнять логические операции над разрядами входных данных, к примеру, инвертировать их и т. п. Если команды и данные подавать в определенной последовательности, то будет производиться соответствующая обработка поступающей цифровой информации. Комбинируя команды, управляющие работой АЛУ, можно реализовывать и разные процедуры обработки. С этой точки зрения АЛУ является программируемым устройством, то есть оно способно работать по заранее сформированной последовательности команд, которая и является программой.

АЛУ в составе серий цифровых микросхем выпускаются в основном для обработки четырехразрядных операндов. Это связано с числом внешних выводов корпуса, требуемых для подачи входных, управляющих сигналов и получения результата. Одной из таких микросхем является К155ИПЗ, изображаемая на схемах следующим образом. У нее имеются входы операндов, а также вход и выход переноса, сигналы на



которых воспринимаются и формируются в инверсном виде. Нулевой уровень на входе  $\overline{C}_0$  соответствует его наличию, а единица – отсутствию. Так как аналогичная ситуация наблюдается и с выходным сигналом  $\overline{C}_4$ , то для увеличения разрядности обрабатываемых чисел, АЛУ по этим входам и выходам можно соединять непосредственно.



Кроме этого в АЛУ формируются функции генерации и распространения переноса, что позволяет для увеличения разрядности объединять их группами по четыре через схемы ускоренного переноса. Для выбора операции, выполняемой АЛУ, служат пять управляющих входов. Сигнал на одном из них –  $M$  переключает вид реализуемой функции (логическая, либо арифметическая). Комбинируя сигналы на остальных –  $S_0 \div S_3$ , можно при  $M=1$  выбрать любую из шестнадцати логических функций от двух переменных, либо, если  $M=0$ , 16 вариантов

арифметических и арифметико-логических операций, что отражено в таблице.

Логические операции над данными производятся поразрядно, то есть для каждой пары одноименных разрядов операндов. В этом случае четырехразрядные кодовые комбинации, поступающие на входы  $A$  и  $B$ , воспринимаются как наборы пар независимых логических переменных.

Если, к примеру, задать операцию  $A_i \vee B_i$ , то произойдет поразрядное логическое сложение и результаты будут переданы на соответствующие выходы АЛУ. В данной ситуации разряды выходного кода считаются не связанными друг с другом.

Выбор функции				M = 1	M = 0	
$S_3$	$S_2$	$S_1$	$S_0$	Логические опер.	Арифметические опер.	
					$\overline{C_0} = 1$	$\overline{C_0} = 0$
0	0	0	0	$F_i = \overline{A_i}$	$F = A$	$F = A + 1$
0	0	0	1	$F_i = \overline{A_i} \vee B_i$	$F = A \vee B$	$F = (A \vee B) + 1$
0	0	1	0	$F_i = \overline{A_i} \wedge B_i$	$F = A \vee \overline{B}$	$F = (A \vee \overline{B}) + 1$
0	0	1	1	$F_i = 0$	$F = -1$	$F = 0$
0	1	0	0	$F_i = \overline{A_i} \wedge \overline{B_i}$	$F = A + (A \wedge \overline{B})$	$F = A + (A \wedge \overline{B}) + 1$
0	1	0	0	$F_i = \overline{B_i}$	$F = (A \vee B) + (A \vee \overline{B})$	$F = (A \vee B) + (A \vee \overline{B})$
0	1	1	0	$F_i = A_i \oplus B_i$	$F = A - B - 1$	$F = A - B$
0	1	1	1	$F_i = A_i \wedge \overline{B_i}$	$F = (A \wedge \overline{B}) - 1$	$F = A \wedge \overline{B}$
1	0	0	0	$F_i = \overline{A_i} \vee B_i$	$F = A + (A \wedge B)$	$F = A + (A \wedge B) + 1$
1	0	0	1	$F_i = \overline{A_i} \oplus B_i$	$F = A + B$	$F = A + B + 1$
1	0	1	0	$F_i = B_i$	$F = (A \vee \overline{B}) + (A \wedge B)$	$F = (A \vee \overline{B}) + (A \wedge B)$
1	0	1	1	$F_i = A_i \wedge B_i$	$F = (A \wedge B) - 1$	$F = A \wedge B$
1	1	0	0	$F_i = 1$	$F = A + A$	$F = A + A + 1$
1	1	0	1	$F_i = A_i \vee \overline{B_i}$	$F = (A \vee B) + A$	$F = (A \vee B) + A + 1$
1	1	1	0	$F_i = A_i \vee B_i$	$F = (A \vee \overline{B}) + A$	$F = (A \vee \overline{B}) + A + 1$
1	1	1	1	$F_i = A_i$	$F = A - 1$	$F = A$

При реализации арифметических операций операнды  $A_i$  и  $B_i$ , воспринимаются как двоичные коды чисел и если реализуется сложение, то происходит арифметическое суммирование разрядов с учетом как входного, так и межразрядных переносов. Так как логическая единица воспринимается как отсутствие переноса на входе АЛУ, а ноль – его наличие, то в зависимости от значения  $\overline{C_0}$ , итог арифметической операции будет отличаться на единицу. При этом положительные результаты формируются в прямом коде, а отрицательные в дополнительном

Арифметико-логические (смешанные) операции реализуются АЛУ в два этапа. Сначала производится поразрядная логическая обработка операндов, а затем арифметическое действие с учетом переносов между разрядами. К примеру операция  $(A \vee B) + (A \vee \bar{B})$  при  $A=0110$  и  $B=0100$  будет выполняться следующим образом. Сначала сформируются функции  $A \vee B = 0110$  и  $A \vee \bar{B} = 1111$ , затем произойдет арифметическое сложение получившихся чисел и, в зависимости от состояния входа переноса, на выходах  $F_i$  получится результат 0101, либо 0110. Так как сумма в обоих случаях больше пятнадцати, то сигнал на выходе  $\bar{C}_4$  примет нулевое значение.

В таком АЛУ имеется выход с открытым коллектором, обозначаемый двумя значками равенства. Сигнал логической единицы на нем будет формироваться при подключении внешнего резистора и равенстве чисел, поступающих на входы  $A_i$  и  $B_i$  АЛУ, если задана операция их вычитания.

#### ***4.12 Умножители двоичных кодов чисел.***

Процедура умножения двоичных кодов чисел достаточно часто встречается при цифровой обработке сигналов. Если рассматривать целые числа, то умножение можно представить как многократное сложение. К примеру, чтобы умножить 3 на 5, требуется тройку пять раз сложить с самой собой. Данным способом умножение можно выполнять, используя, в частности, накапливающие сумматоры. Однако время выполнения этой процедуры будет достаточно велико и, кроме того, оно окажется зависимым от величины множителя.

Другие подходы к реализации процедуры перемножения чисел требуют многократного последовательного выполнения операций сложения и сдвига. Их количество, а

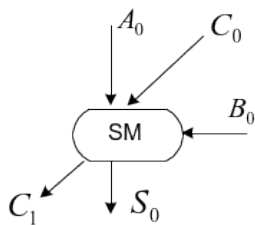
следовательно, скорость выполнения умножения, также зависят от разрядности множителя.

Разработка специальных алгоритмов умножения позволила создать так называемые параллельные или матричные умножители. Одна из их разновидностей называется множительным, а другая - множительно-суммирующим блоком. Последние обеспечивают возможность наращивания разрядности обрабатываемых операндов.

Работу матричного множительного блока можно пояснить на примере перемножения двух трехразрядных чисел, представленных в двоичном коде. На первом этапе разряды множимого умножаются на младший разряд множителя, затем на старший и т.д. Полученные наборы частичных произведений суммируются со сдвигом на один разряд вправо.

$$\begin{array}{r}
 \begin{array}{r}
 a_2 \quad a_1 \quad a_0 \\
 b_2 \quad b_1 \quad b_0 \\
 \hline
 a_2 b_0 \quad a_1 b_0 \quad a_0 b_0 \\
 a_2 b_1 \quad a_1 b_1 \quad a_0 b_1 \\
 a_2 b_2 \quad a_1 b_2 \quad a_0 b_2 \\
 \hline
 P_5 \quad P_4 \quad P_3 \quad P_2 \quad P_1 \quad P_0
 \end{array}
 \end{array}$$

Так как таблица умножения в двоичной системе счисления идентична таблице, описывающей операцию конъюнкции двух логических переменных, то получение компонент частичных произведений можно реализовать на элементах 2И. Для данного примера их понадобится 9.



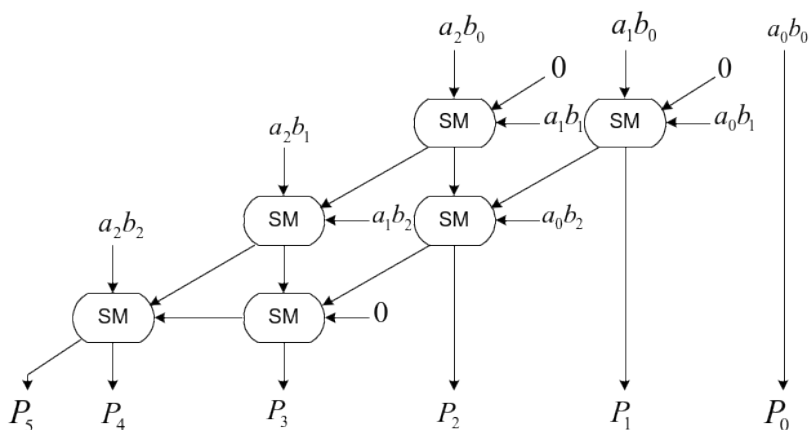
Далее произведения разрядов необходимо просуммировать по столбцам с учетом переносов и в итоге получится результат, равный произведению чисел. Его разрядность будет равна сумме разрядностей сомножителей.

Формирование разрядов произведения можно произвести путем последовательного

сложения пар соответствующих компонент, используя для этого полные одноразрядные двоичные сумматоры.

Как следует из записи процедуры перемножения, разряд  $P_1$  получается при сложении компонент  $a_1b_0$  и  $a_0b_1$ , на вход переноса  $C_0$  сумматора при этом потребуется подать ноль. При суммировании компонент  $a_2b_0$ ,  $a_1b_1$ , необходимо учесть перенос из предыдущего разряда и передать сигнал переноса в следующий.

Структура параллельного умножителя, реализующего данный алгоритм, будет выглядеть таким образом. Задержка



формирования произведения здесь определяется наиболее длинной цепочкой прохождения промежуточных результатов и для разрядов  $P_4$  и  $P_5$  она составит  $4\tau_S + \tau_{ЛЭ}$ . У перемножителя чисел с разрядностями  $m$  и  $n$  она будет равна  $(m + n - 2)\tau_S + \tau_{ЛЭ}$ .

Построить параллельные умножители большой разрядности можно наращивая рассмотренную структуру, либо объединяя множительно-суммирующие блоки, которые реализуют операцию  $P = A * B + C + D$ . Примером такого блока является микросхема К555ИП8, выполняющая

умножение четырехразрядного числа на двухразрядное и прибавление к результату пары чисел с разрядностью 4 и 2.

Возрастание задержек получения результата с увеличением разрядности чисел привело к необходимости разработки иных алгоритмов быстрого умножения. При реализации одного из них, называемого модифицированным алгоритмом Бута, умножение происходит сразу на два разряда. Это позволяет сократить количество операций сложения при формировании частичных произведений и уменьшить длины цепочек последовательного прохождения сигналов.

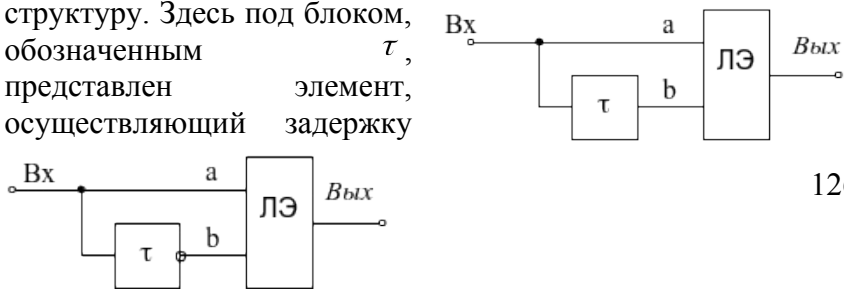
Промышленностью выпускаются в интегральном исполнении функционально законченные блоки для быстрого умножения двух восьмиразрядных чисел - микросхема К1802ВР3, двенадцатиразрядных - К1802ВР4 и шестнадцатиразрядных - К1802ВР5. Время выполнения операции умножения у них составляет порядка 100÷120 нС. В настоящее время аналогичные устройства встраиваются в сопроцессоры и ряд микроконтроллеров.

## 5. ФОРМИРОВАТЕЛИ И ГЕНЕРАТОРЫ НА ЛОГИЧЕСКИХ ЭЛЕМЕНТАХ.

### 5.1 Формирователи импульсов.

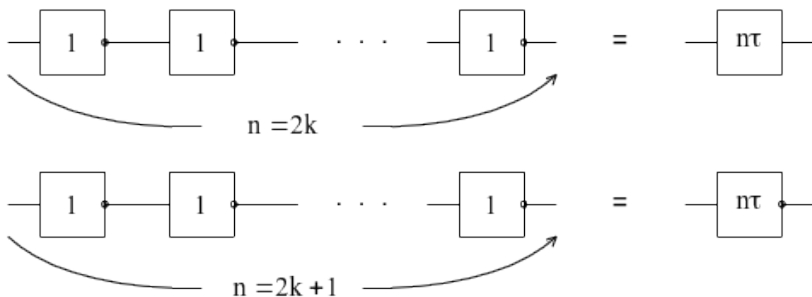
В цифровой технике используются лишь прямоугольные импульсные сигналы, поэтому под формирователями понимаются узлы, которые сохраняя форму импульса, изменяют его длительность, либо осуществляют сдвиг во времени.

Простейшие формирователи имеют следующую структуру. Здесь под блоком, обозначенным  $\tau$ , представлен элемент, осуществляющий задержку

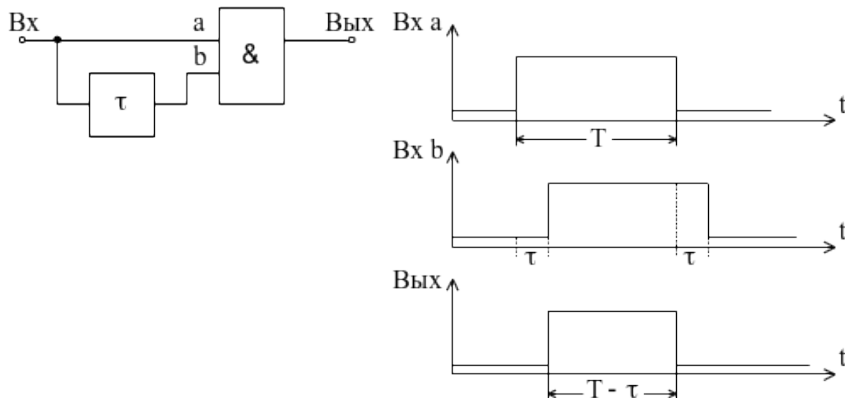


входного сигнала на время  $\tau$ , меньше длительности входного импульса. В качестве таких элементов можно использовать интегрирующие цепочки, а также группу последовательно соединенных инверторов, либо буферных вентилях.

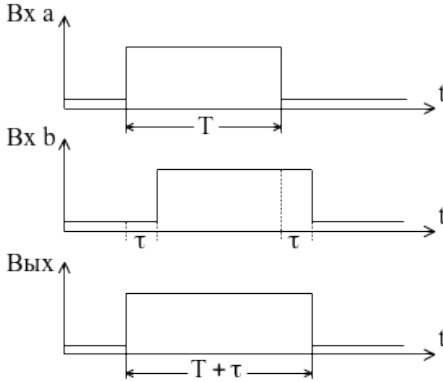
Если задержка прохождения сигнала в одном инверторе составляет  $\tau$ , то при их четном количестве  $n$ , общая задержка без инверсии сигнала будет равна  $n\tau$ , а если  $n$  нечетное, то при задержке  $n\tau$  выходной сигнал окажется проинвертированным.



Временные диаграммы работы первого варианта формирователя при использовании логического элемента 2И



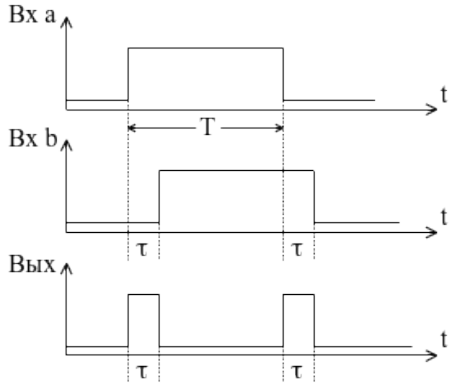
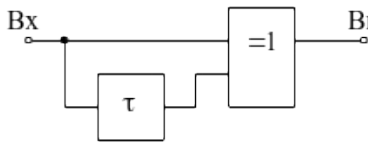
выглядят следующим образом. Так как высокий уровень напряжения формируется лишь при наличии сигналов логических единиц на обоих входах, то длительность



выходного импульса в этом случае окажется на  $\tau$  меньше, чем входного.

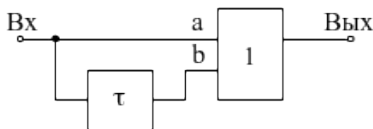
Выходной сигнал элемента 2ИЛИ принимает единичное значение, если на любом из входов, или на обоих сразу присутствуют логических единицы. В случае его использования в

формирователе, длительность выходного импульса окажется

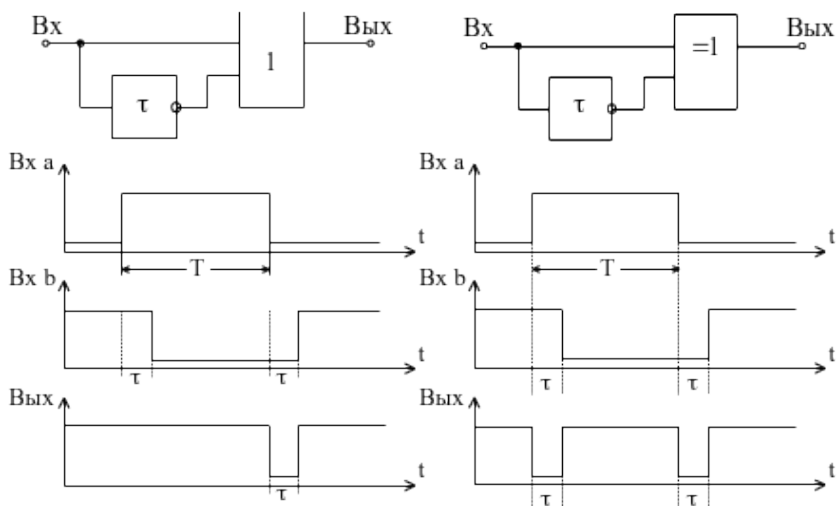


на  $\tau$  больше, чем входного.

У элемента логической неравнозначности (сумматора по модулю два) единичный сигнал на выходе появляется при неравенстве входных. Отсюда следует, что формирователь с таким элементом будет вырабатывать короткие импульсы, длительностью  $\tau$  по фронту и спаду входного сигнала.

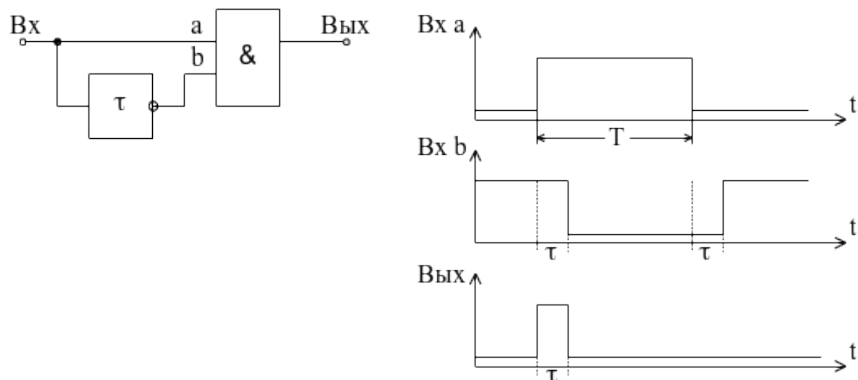


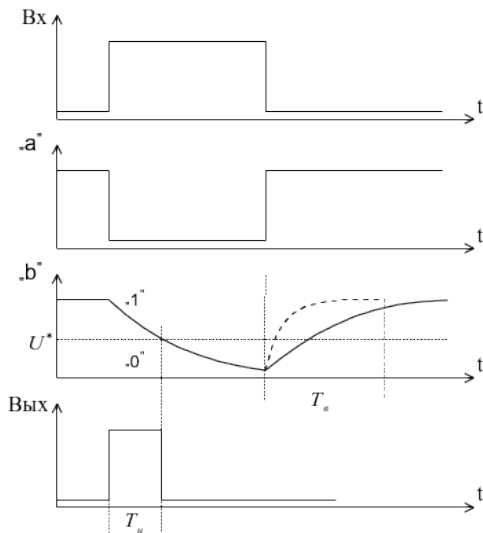




Для варианта формирователя с инвертирующим элементом задержки временные диаграммы при использовании схемы 2И имеют вид. Здесь логические единицы на обоих входах появятся лишь на время  $\tau$  по переднему фронту входного сигнала, поэтому такой же будет и длительность выходного импульса.

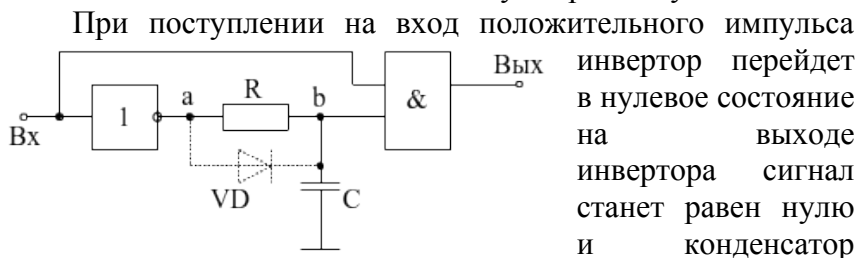
Использование элемента 2ИЛИ приведет к формированию короткого импульса нулевого уровня по спаду входного сигнала, а применение сумматора по модулю два вызовет появление двух аналогичных импульсов (по фронту и спаду).





На практике в цепочках из небольшого количества логических элементов трудно обеспечить задержки на время большее сотен наносекунд. В то же время достаточно часто возникает необходимость формирования импульсов выходной с длительностями порядка сотен микросекунд и более.

В этом случае можно использовать формирователь такой структуры. Здесь в качестве элемента задержки выступает интегрирующая цепочка, состоящая из резистора  $R$  и конденсатора  $C$ . Из временных диаграмм работы такой схемы следует, что в отсутствие входного сигнала на выходе элемента НЕ присутствует высокий уровень, и конденсатор окажется заряженным практически до напряжения логической единицы. Сигнал на выходе схемы будет равен нулю.



При поступлении на вход положительного импульса инвертор перейдет в нулевое состояние на выходе инвертора сигнал станет равен нулю и конденсатор начнет разряжаться через его выходной каскад с постоянной времени  $\tau = RC$ . Пока напряжение в точке «в» не станет ниже порога переключения  $U^*$  элемента 2И, сигналы на его входах воспринимаются как логические единицы и в течение этого

времени ( $T_u$ ) на выходе формируется высокий уровень напряжения.

Далее сигнал на нижнем входе элемента 2И перейдет в область, воспринимаемую как уровень логического нуля, он переключится в нулевое состояние, а конденсатор будет продолжать разряжаться в течение всей длительности входного импульса. По его окончании система вернется в исходное состояние, однако для этого понадобится время, называемое временем восстановления ( $T_6$ ), которое требуется для заряда конденсатора.

Ускорить процесс восстановления можно уменьшив постоянную времени цепи перезаряда, в данном случае величину сопротивления  $R$ . Для этой цели параллельно резистору включается диод, как показано на схеме пунктирной линией. Он открывается когда напряжение на выходе инвертора становится более положительным, чем на верхней обкладке конденсатора, его небольшое прямое сопротивление шунтирует резистор, чем обеспечивается ускорение процесса заряда.

Разряд конденсатора происходит по экспоненциальному закону,

описываемому соотношением

$$U(t) = U_{\text{вых}}^1 - (U_{\text{вых}}^1 - U_{\text{вых}}^0) \left(1 - \exp^{-\frac{t}{RC}}\right), \quad \text{где } U_{\text{вых}}^1 \text{ и } U_{\text{вых}}^0$$

напряжения логического нуля и единицы. Уровень порога переключения  $U^*$  инвертора и схемы 2И достигается за время, равное длительности импульса  $T_H$ , откуда

$$U^* = U_{\text{вых}}^1 - (U_{\text{вых}}^1 - U_{\text{вых}}^0) \left(1 - \exp^{-\frac{T_H}{RC}}\right).$$

Раскрыв скобки и

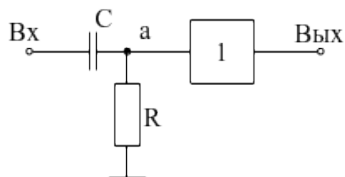
прологарифмировав, получим, что  $T_H = RC \ln \frac{U_{\text{вых}}^1 - U_{\text{вых}}^0}{U^* - U_{\text{вых}}^0}$ .

Считая напряжение логического нуля равным нулю, а порог

переключения  $U^* = \frac{U_{\text{вых}}^1}{2}$ , длительность импульса можно определить по формуле  $T_{\text{И}} = RC \ln 2 \approx 0,7RC$ .

При расчете параметров такого формирователя следует учесть ряд обстоятельств. На вход элемента 2И поступает медленно меняющееся напряжение, что может вызвать его паразитную генерацию и появление пачки импульсов на спаде выходного сигнала. Использование ТТЛ и ТТЛШ вентилей накладывает определенные ограничения на максимальную величину сопротивления резистора, а для КМОП на минимальную.

Это связано с тем, что через входную цепь ТТЛ логического элемента протекает ток, который вызывает падение напряжения на резисторе. Если при низком уровне сигнала на выходе инвертора, оно превысит максимальное значение потенциала логического нуля, то на входе схемы 2И будет присутствовать сигнал, воспринимаемый как единичный и длительность выходного импульса окажется равной длительности входного.



В первом приближении максимальное сопротивление резистора определяется соотношением

$$R_{\text{max}} \approx \frac{0,7B - 0,2B}{1,6mA} \approx 300 \text{ Ом}. \text{ Напряжение } 0,2\text{В} \text{ в числителе}$$

соответствует выходному уровню логического нуля.

При использовании КМОП элементов небольшие сопротивления резистора могут вызвать перегрузку выходного каскада вентиля и нарушение режима его работы. Максимальная величина его сопротивления теоретически не ограничена, так как входные токи КМОП вентиля пренебрежимо малы.

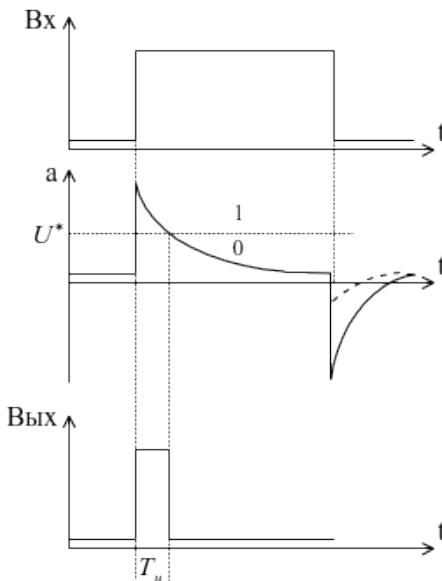
Еще один вариант формирователя, уменьшающий длительность входного импульса, использует

дифференцирующую цепочку и выглядит следующим образом.

Во время действия положительного фронта входного сигнала скачок напряжения через разряженный конденсатор поступает на логический элемент (повторитель) и вызывает появление на его выходе сигнала логической единицы. По мере заряда конденсатора, напряжение на входе вентиля будет уменьшаться, и когда его уровень опустится ниже порога переключения, это воспримется как появление сигнала логического нуля, что вызовет установление на выходе аналогичного сигнала. Длительность импульса, формируемого такой схемой примерно равна  $T \approx 0,7RC$ .

В момент исчезновения входного сигнала, заряженный до напряжения  $U_{ВХ} - U_0$  конденсатор, окажется подключенным параллельно входу логического элемента, где появится отрицательный потенциал, уменьшающийся по мере разряда конденсатора до величины  $I_{ВХ}^0 R$ . Установка диода, или их наличие в структуре логического элемента позволяет

снизить длительность процесса восстановления (это показано на рисунке штриховой линией).

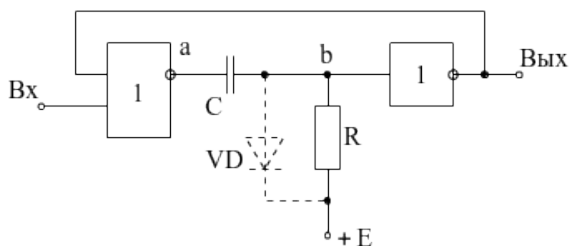


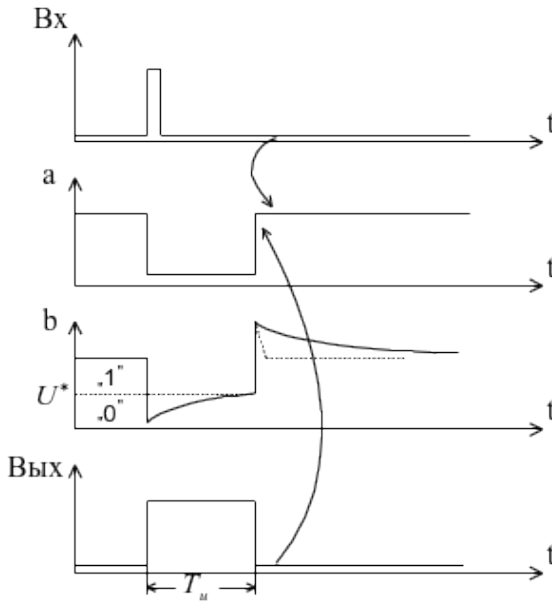
Рассмотренные формирователи позволяют преобразовать относительно длинный входной импульс в более короткий выходной. В ряде случаев требуется решить обратную задачу. Такие устройства могут быть реализованы либо с использованием времязадающих цепочек с

разными постоянными времени заряда и разряда, или введением в схему цепи положительной обратной связи.

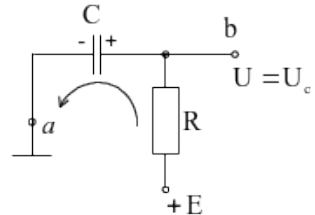
Один из вариантов формирователя второго типа выглядит следующим образом. В исходном состоянии через резистор  $R$  вход инвертора подключен к положительному полюсу источника питания и на его выходе формируется напряжение логического нуля. Оно совместно с нулевым входным сигналом поступает на элемент 2ИЛИ-НЕ, что приводит к появлению на его выходе (в точке «а») высокого уровня напряжения. Такое состояние является устойчивым. Конденсатор  $C$  при этом практически разряжен, так как обе его обкладки имеют высокий потенциал.

Приход положительного фронта входного импульса переводит элемент 2ИЛИ-НЕ в нулевое состояние, а так как заряд на конденсаторе мгновенно измениться не может, то аналогичный скачок напряжения появится и на входе инвертора





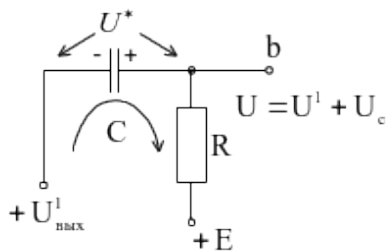
Это вызовет появление на его выходе единичного сигнала, который по цепи обратной связи поступит на верхний вход 2ИЛИ-НЕ. Новое состояние устройства сохранится пока на нем будет поддерживаться сигнал логической единицы.



С этого момента времени конденсатор  $C$  начнет заряжаться по цепи - плюс источника питания, резистор, выходной каскад элемента ИЛИ-НЕ. Напряжение на его правой обкладке станет повышаться и когда оно достигнет порога переключения, инвертор перейдет в состояние логического нуля.

Этот сигнал поступит на верхний вход элемента ИЛИ-НЕ, что приведет к его переходу в единичное состояние и возврату устройства в исходное. Длительность сформированного импульса определяется соотношением

$$T_u \approx RC \ln \frac{U_{\text{Вых}}^1}{U^*}.$$



Из-за того, что к этому моменту на конденсаторе сохраняется некоторый заряд, уровень сигнала в точке «b» скачком увеличится и станет равным

$$U_{Вых}^1 + U_C = U_{Вых}^1 + U^* \quad \text{В}$$

течение времени, которое называется временем восстановления, на входе инвертора будет действовать напряжение, превышающее потенциал источника питания. Для уменьшения длительности этого процесса в схему можно ввести диод VD, который откроется в момент скачка напряжения и образует цепь разряда конденсатора, обладающую малым сопротивлением.

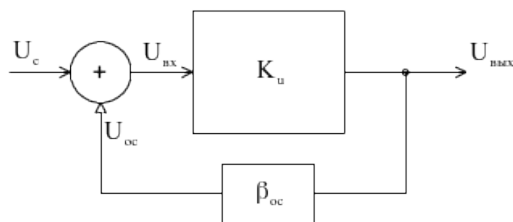
## 5.2 Генераторы цифровых сигналов.

Такие генераторы являются источниками однополярных прямоугольных импульсов определенной длительности и с заданной частотой следования. Они подразделяются на две группы: автогенераторы и преобразователи частоты. Первые после подключения к источнику питания формируют последовательность импульсов с некоторой частотой  $f_0$ , а частота сигналов на выходе вторых определяется соотношением  $f = \frac{f_0}{M} K$ , где  $M$  и  $K$  целочисленные коэффициенты.

Любой автогенератор может быть представлен как система, состоящая из усилителя с коэффициентом усиления

$$K_U = \frac{\Delta U_{вых}}{\Delta U_{вх}}, \quad \text{охваченного цепью обратной связи с}$$

коэффициентом





передачи  $\beta_{oc} = \frac{U_{oc}}{U_{вых}}$ . Коэффициент передачи такой системы

определяется соотношением  $\dot{K}_{u oc} = \frac{\Delta U_{вых}}{\Delta U_c} = \frac{\dot{K}_u}{1 - \beta_{oc} K_u}$  где

$\dot{K}_u(\omega) = K_u(\omega)e^{i\varphi_k(\omega)}$ ,  $\dot{\beta}_{oc}(\omega) = \beta_{oc}(\omega)e^{i\varphi_\beta(\omega)}$ . В общем случае

коэффициенты усиления и передачи цепи обратной связи зависят от частоты и описываются комплексными функциями. Физически это означает, что между входными и выходными сигналами имеются фазовые сдвиги. С учетом этого формула для коэффициента передачи усилителя с обратной связью примет вид

$$\dot{K}_{u oc}(\omega) = \frac{K_u(\omega)e^{i\varphi_k(\omega)}}{1 - K_u(\omega)\beta_{oc}(\omega)e^{i[\varphi_k(\omega)+\varphi_\beta(\omega)]}}$$

Из данного соотношения следует, что при определенных условиях знаменатель дроби может обратиться в ноль и коэффициент усиления системы станет равным бесконечности. В этом случае при нулевом входном сигнале уровень выходного может быть отличен от нуля, что и является свойством и особенностью генератора.

Данная ситуация реализуется, если 
$$\begin{cases} K_u(\omega)\beta_{oc}(\omega) = 1 \\ \varphi_k(\omega) + \varphi_\beta(\omega) = 2k\pi \end{cases}$$

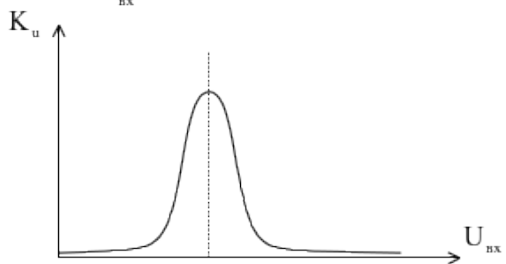
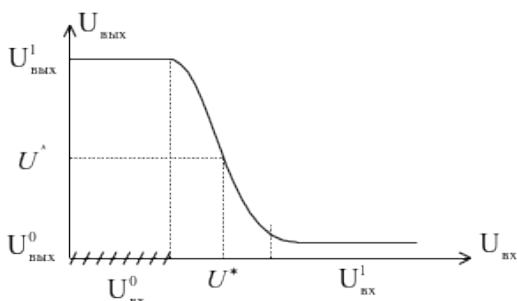
где  $k$  – любое целое число. Это связано с тем, что выражение  $e^{ix} = \cos x + i \sin x$  будет равно единице при  $x = 0, 2\pi, 4\pi, \dots, 2k\pi$ .

Первое соотношение называется условием самовозбуждения по амплитуде и означает, что потери в цепи обратной связи полностью компенсируются усилителем. Второе является условием самовозбуждения по фазе. При его выполнении суммарный сдвиг фаз между входным и сигналом обратной связи таков, что они суммируются. Это соответствует наличию положительной обратной связи, когда сигнал обратной связи находится в фазе со входным.

Если данные условия выполняются на какой-либо одной частоте  $f_1 = \frac{\omega_1}{2\pi}$ , то система будет формировать сигнал в спектре которого содержится лишь одна компонента, то есть выходное напряжение должно иметь синусоидальную форму. Когда эти условия выполняются сразу на нескольких частотах, то одновременно будет генерироваться набор синусоидальных сигналов с соответствующими частотами, амплитудами и фазами.

Отсюда следует, что для формирования периодической последовательности импульсов прямоугольной формы с частотой  $f_0$  условия самовозбуждения должны выполняться для  $f_0, 3f_0, 5f_0$  и т. д., так как все эти компоненты содержатся в спектре такой последовательности.

В качестве активного узла (усилителя) в генераторах цифровых сигналов обычно используются простейшие логические элементы, в частности инверторы. При

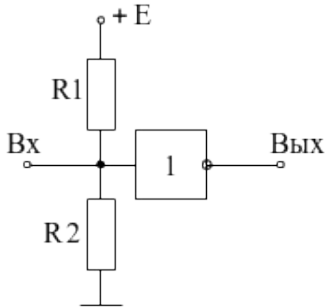


определенных условиях они могут выполнять функции усилителя с фазовым сдвигом  $\pi$  и коэффициентом усиления до нескольких сотен. Примерная форма выходной характеристики инвертора и зависимость его коэффициента

$$\text{усиления } K_u = \frac{\Delta U_{\text{вых}}}{\Delta U_{\text{вх}}}$$

от уровня входного сигнала имеют следующий вид.

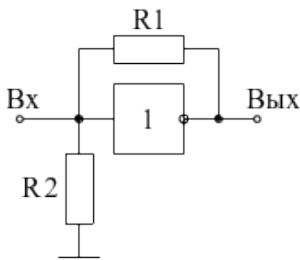
Из приведенных графиков следует, что в диапазонах входных и выходных сигналов, соответствующих уровням логического нуля и единицы, коэффициент усиления инвертора практически равен нулю. Лишь для напряжений близких к порогу переключения, когда все транзисторы вентиля работают в активном режиме, он может принимать достаточно большие значения.



Чтобы использовать инвертор в качестве элемента генератора, его требуется перевести в активный режим. Это можно сделать подав на его вход с делителя на резисторах  $R_1, R_2$  напряжение близкое к порогу переключения, которое для ТТЛ схем составляет 1,2-1,4 В. Так как входной ток при таком

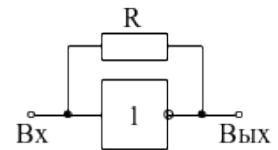
напряжении практически равен нулю, то  $U_{вх} = E \frac{R_2}{R_1 + R_2}$ .

Второй вариант формирования напряжения смещения выглядит следующим образом. Здесь



$U_{вх} \approx U_{вых} \frac{R_2}{R_1 + R_2}$ . В третьем

варианте используется лишь один резистор, который является элементом параллельной отрицательной

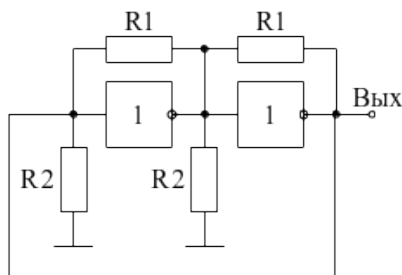


обратной связи по напряжению, вследствие чего оно будет автоматически поддерживаться на уровне  $U_{вых} \approx U^* + I_{вх} R \approx U^*$ . Это объясняется тем, что любые изменения выходного напряжения передаются на вход, а так как активным элементом является инвертор, то рост входного сигнала

приведет к снижению выходного и наоборот. Данный вид обратной связи не влияет на коэффициент усиления по напряжению, уменьшая его лишь по току.

Таким образом, если в качестве усилительного элемента использовать инвертор, работающий в активном режиме и ввести цепь обратной связи с фазовым сдвигом равным  $\pi$  и коэффициентом передачи большим  $\beta_{oc} \geq 1/K_U$ , то схема должна функционировать как автогенератор. Элементом обратной связи может выступать и инвертирующий усилитель, то есть аналогичный инвертор, работающий в активном режиме. При этом суммарный фазовый сдвиг окажется равным  $2\pi$ , а условие самовозбуждения по амплитуде выполнится с большим запасом.

Однако на практике данная система функционировать как автогенератор не будет. Это связано со специфической зависимостью коэффициента усиления вентиля от уровня сигнала. Он становится практически равным нулю,

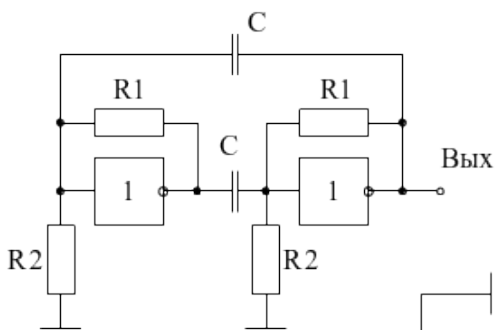


когда входное напряжение находится в зонах, соответствующих логическому нулю и единице. При этом нарушаются условия самовозбуждения по амплитуде так как  $K_U K_U \ll 1$ .

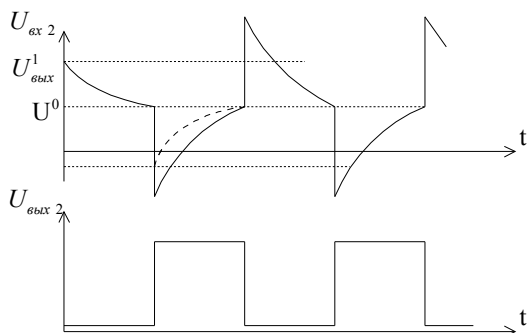
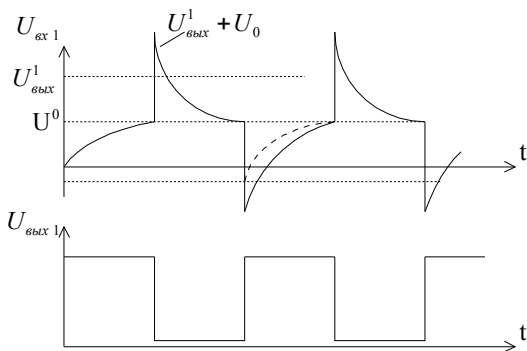
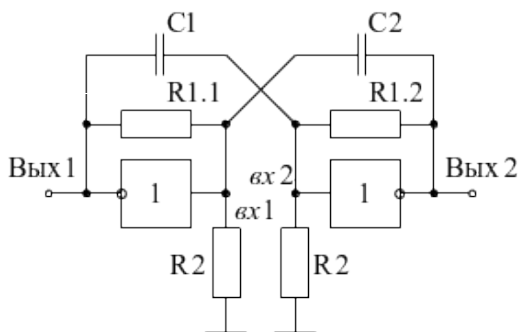
Положительная обратная связь по постоянному току в схеме действует, в связи с чем такая система может находиться в двух устойчивых состояниях. Если на входе первого инвертора напряжение равно нулю, то на его выходе и соответственно на входе второго оно будет иметь высокий уровень, а, напряжение на выходе второго инвертора и на входе первого окажется нулевым. Аналогичная картина получится, если предположить, что на входе присутствует

напряжение логической единицы. Формально можно считать, что при этом генерируется сигнал с нулевой частотой.

Превратить данную схему в автогенератор можно ограничив диапазон изменения входных и выходных напряжений таким образом, чтобы вентили все время находились в активной области, либо принудительно выводить их из состояний при которых коэффициенты усиления обращаются в нуль. Однако в первом случае выходные сигналы не будут соответствовать стандартным значениям уровней логического нуля и единицы. Второй способ реализуется устранением обратной связи по постоянному току с сохранением ее по переменному, для чего используются разделительные конденсаторы.



В несколько видоизмененном виде схема данного устройства может быть представлена следующим образом.

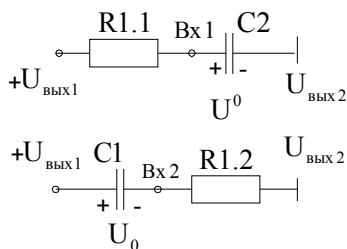


Ее конфигурация похожа на схему транзисторного мультивибратора, однако здесь в качестве ключей выступают инверторы. Элементы схемы обычно выбираются попарно одинаковыми.

Работа такого устройства может быть описана следующим образом. Пусть в исходном состоянии на первом выходе присутствует сигнал логической

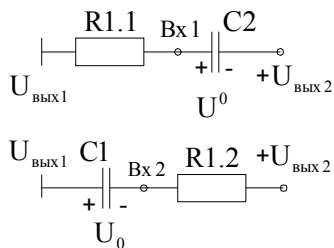
единицы, на втором – нуля, на входе левого инвертора напряжение равно нулю, а на входе правого близко к  $U^1_{\text{вх}}$ .

Практически разряженные конденсаторы  $C1$  и  $C2$  начнут заряжаться по цепям которые показаны без учета резисторов  $R2$ , а также входных и выходных сопротивлений вентилях. Потенциал входа левого инвертора ( $Bx1$ ) станет



увеличиваться, а напряжение на входе второго инвертора будет падать. Через некоторое время напряжение на  $Bx1$  превысит порог переключения вентиля  $U_0$ , а напряжение на втором входе станет меньше его. При этом оба инвертора перейдут в противоположные состояния.

Так как разность потенциалов на обкладках конденсатора мгновенно измениться не может, то скачки напряжения, возникающие на выходах инверторов, поступят на соответствующие входы. Конденсаторы начнут

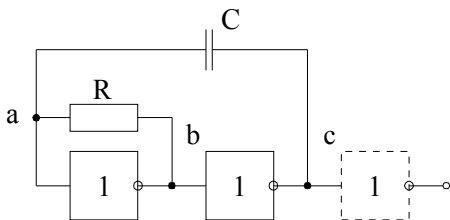


перезаряжаться, вследствие чего потенциал первого входа станет падать, а второго увеличиваться. В момент времени, когда они достигнут порогов переключения логических элементов, те перейдут в противоположные (исходные) состояния и процессы будут периодически повторяться. Штриховыми линиями на временных диаграммах показан ход перезаряда конденсаторов при наличии на входах вентилях антизвонных диодов.

Частота следования прямоугольных импульсов формируемых такой схемой зависит от параметров всех элементов, входящих в состав времязадающих цепей, но в

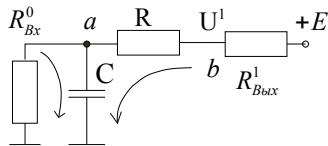
основном определяется емкостью конденсаторов и сопротивлениями резисторов R1. На величины емкостей не накладывается принципиальных ограничений, но сопротивления резисторов должны быть не больше, либо не меньше определенных значений, что зависит от свойств логических элементов. В частности для вентилях серии K155 при R2=1КОм, R1=2КОм частота генерируемых колебаний в килогерцах может быть определена из соотношения  $f \approx 1/1,6C$ , если емкость учитывается в микрофарадах.

Обратную связь по постоянному току можно разорвать в одной точке схемы, и при измененной конфигурации цепи смещения она примет вид.



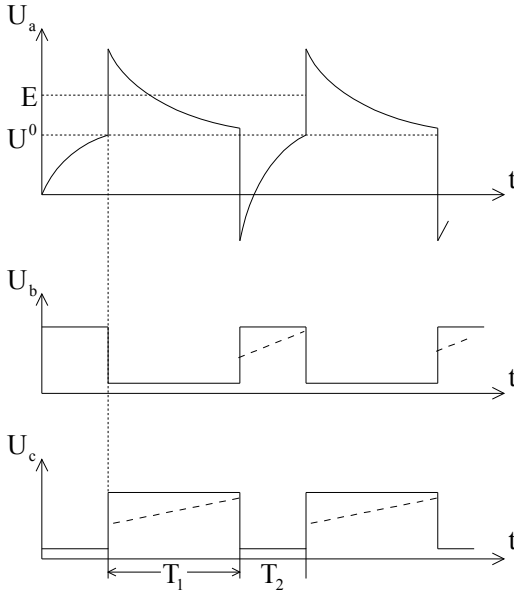
Если считать, что в исходном состоянии напряжение в точке «а»

равно нулю, то в точке «b» оно будет иметь высокий уровень, а на выходе второго инвертора опять будет нулевым. В данной ситуации разность потенциалов на обкладках конденсатора равна нулю и он начнет заряжаться через резистор R и выходное сопротивление первого инвертора, находящегося в состоянии логической единицы. Параллельно конденсатору оказывается входное сопротивление инвертора и, с учетом того, что входной ток логического нуля вытекает из соответствующей цепи ТТЛ элемента, постоянная времени заряда конденсатора будет определяться соотношением  $\tau \approx C \left[ (R + R_{Вых}^1) \parallel R_{Вх}^0 \right]$ .

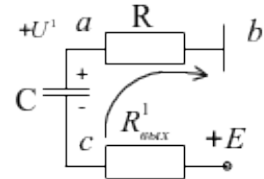


Когда напряжение на входе первого инвертора достигнет порога переключения, его состояние изменится на противоположное. При этом в точке «b» появится низкий





уровень напряжения, а на выходе второго инвертора сформируется сигнал логической единицы.

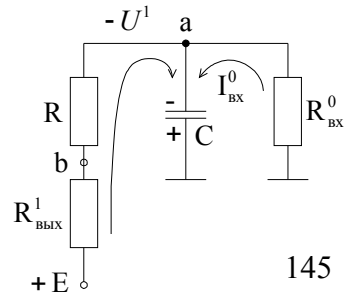


Вследствие

того, что разность потенциалов на обкладках конденсатора мгновенно

измениться не может, в момент переключения второго инвертора на входе первого логического элемента сформируется скачок напряжения величиной  $U_{\text{вых}}^1 - U_{\text{вых}}^0$ . Конденсатор начнет перезаряжаться через резистор  $R$  и выходное сопротивление инвертора, находящегося в состоянии логической единицы. Постоянная времени цепи перезаряда будет равна  $\tau \approx C(R + R_{\text{вых}}^1)$ .

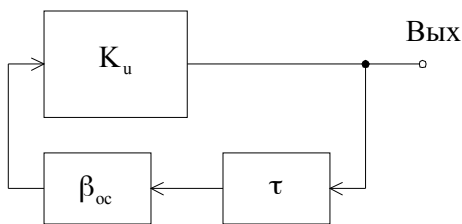
Уменьшающийся потенциал точки «а» через время  $T_1$  достигнет порога переключения логического элемента и схема перейдет в исходное состояние. На входе первого инвертора появится отрицательный скачок напряжения величиной  $U_{\text{вых}}^1 - U_{\text{вых}}^0$  и конденсатор начнет перезаряжаться. Когда напряжение в точке  $a$  достигнет порога переключения  $U_0$ , схема изменит свое состояние на



противоположное и эти процессы будут периодически повторяться.

Как следует из временных диаграмм, длительность формируемого схемой положительного импульса больше интервала времени в течение которого напряжение близко к нулю. Это связано с разными постоянными времени цепей перезаряда конденсатора. При величине сопротивления  $R \approx 390 \text{ Ом}$ , частота выходного сигнала может быть рассчитана по формуле  $f \approx \frac{1200}{C_{(МКФ)}} (Гц)$ . Применение КМОП элементов, у которых входные токи практически отсутствуют ( $R_{вх} \approx \infty$ ), устраняет асимметрию и позволяет использовать резисторы с относительно высокими значениями сопротивлений.

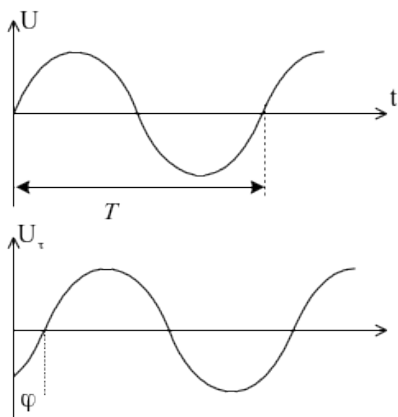
В моменты бросков тока на начальных этапах перезаряда конденсаторов, выходные каскады инверторов работают с перегрузкой, что приводит к снижению уровня логической единицы. Временные диаграммы сигналов с учетом этого обстоятельства представлены на рисунке штриховыми линиями. Для получения стандартных значений уровней к выходу генератора может быть подключен формирующий каскад, на третьем логическом элементе.



Разновидностью генераторов импульсных колебаний являются генераторы с запаздывающей обратной связью, структура которых имеет

следующий вид. У них в цепи обратной связи используется элемент, который задерживает появление выходного сигнала относительно входного на время  $\tau$ .

Задержка во времени эквивалентна появлению фазового сдвига, величина которого определяется соотношением



$$\varphi = 2\pi \frac{\tau}{T} = 2\pi f\tau.$$

Отсюда

следует, что на некоторых частотах фазовый сдвиг  $\varphi_\tau$  станет равным  $\pi$ , и при использовании инвертирующего усилителя и цепи обратной связи с  $\varphi_\beta = 0$ , в схеме будет выполняться условие самовозбуждения по фазе.

Коэффициент передачи

такой системы определяется соотношением

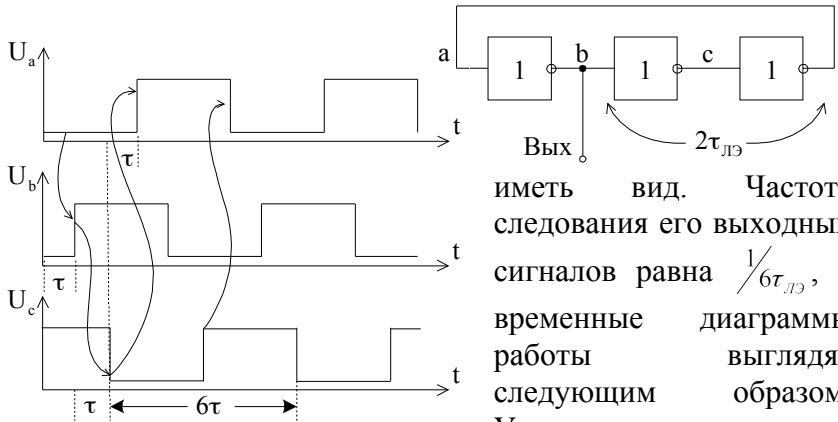
$$K_{Uoc} = \frac{K_U e^{i\pi}}{1 - \beta_{oc} K_U e^{i\pi} e^{-i2\pi f\tau}}.$$

Знаменатель обращается в ноль, если  $\beta_{oc} K_U = 1$  и  $e^{-i(2\pi f\tau - \pi)} = 1$ . Учитывая, что  $e^{-ix} = \cos x - i \sin x$ , это произойдет при  $2\pi f\tau - \pi = 0, 2\pi, 4\pi, \dots, 2k\pi$ , откуда следует, что частоты, для которых будут выполняться условия самовозбуждения по фазе, равны  $f_k = \frac{2k+1}{2\tau}$ . Если при этом выполнены условия самовозбуждения по амплитуде, то система начнет генерировать набор синусоидальных сигналов с частотами  $f_k$  и суммарный сигнал может иметь прямоугольную форму.

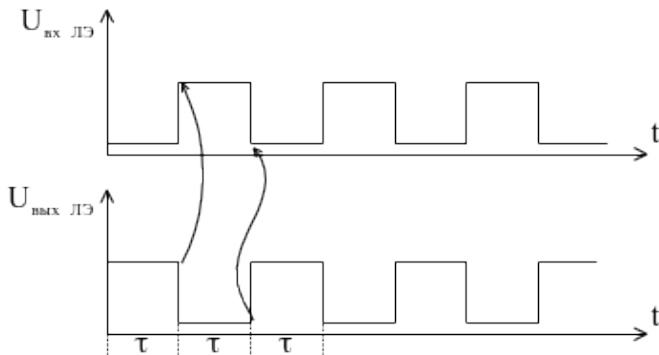
При использовании в качестве инвертирующего усилителя идеального инвертора, не вносящего задержек, частота следования прямоугольных импульсов в такой схеме будет равна  $\frac{1}{2\tau}$ .

Появление логического нуля на входе инвертора переведет его в единичное состояние, но этот сигнал поступит на его вход через время  $\tau$ . На выходе сформируется нулевой уровень, который через  $\tau$  поступит на вход и т.д.

Если в качестве усилителя применен инвертор с временем переключения  $\tau$ , а элемент задержки выполнен на группе из четного количества последовательно соединенных инверторов, то простейший вариант такого генератора будет

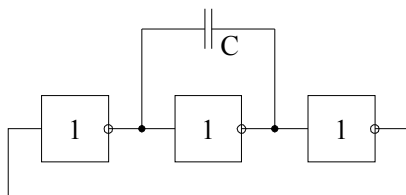


иметь вид. Частота следования его выходных сигналов равна  $\frac{1}{6\tau_{лэ}}$ , а временные диаграммы работы выглядят следующим образом. Увеличение периода колебаний на  $2\tau$



объясняется тем, что первый инвертор вносит задержку  $\tau$  при формировании как положительного, так и отрицательного фронтов сигнала.

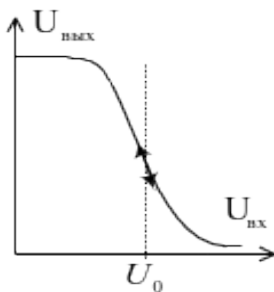
Для клапанов серии К155 с задержками порядка 20 нс, частота генерируемых колебаний составит около 16 МГц. Уменьшить ее можно, лишь увеличивая задержку в цепи обратной связи, путем добавления цепочек из четного числа инверторов, либо замедлением процесса их переключения.



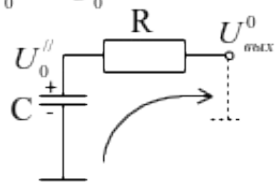
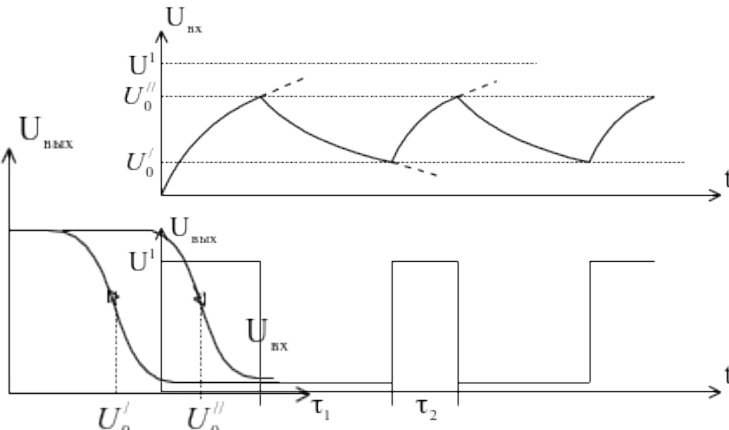
Второй вариант реализуется установкой конденсатора небольшой емкости между входом и выходом любого из инверторов.

В генераторах такого типа сигналы можно снимать с выхода любого из элементов, они будут идентичны, но сдвинуты друг относительно друга на соответствующие интервалы времени.

Достаточно часто в качестве активного элемента генераторов прямоугольных импульсов используется инвертирующий триггер Шмидта. В отличие от обычного инвертора он обладает гистерезисной характеристикой, то есть порог переключения из одного состояния в другое



зависит от направления изменения входного напряжения. Для стандартных ТТЛ клапанов порог переключения составляет  $1,3 \div 1,5 \text{ В}$ , а у элемента с гистерезисной характеристикой  $U_0'' \approx 1,6 \div 1,8 \text{ В}$ , а  $U_0' \approx 0,8 \div 0,9 \text{ В}$ .



логического  
напряжения

заряжаться и  
инвертора

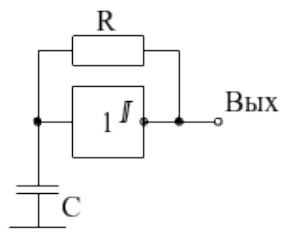
закону

Когда он

срабатывания  $U_0^{II}$  триггер Шмидта переключится в противоположное состояние.

Заряженный к этому моменту до  $U_0^{II}$  конденсатор начнет разряжаться через резистор  $R$  и выходной каскад логического элемента, напряжение на котором равно  $U_{вых}^0$ . Процесс разряда описывается соотношением

$$U_C(t) = U_{вых}^0 + (U_0^{II} - U_{вых}^0) e^{-t/RC}.$$



Принципиальная схема варианта генератора на триггере Шмидта выглядит следующим образом. Если в исходном состоянии конденсатор разряжен ( $U_{вх} = 0$ ), то на выходе

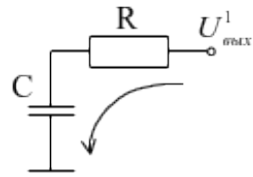
элемента присутствует высокого уровня  $U_{Вых}^1$ .

Конденсатор начнет потенциал входа станет возрастать по

$$U_C(t) = U_{Вых}^1 (1 - e^{-t/RC}).$$

достигнет порога

срабатывания  $U_0^{II}$  триггер Шмидта переключится в



Когда напряжение на конденсаторе достигнет нижнего порога переключения триггера Шмидта, он перейдет в исходное состояние, далее процессы будут периодически повторяться и на выходе сформируется последовательность прямоугольных импульсов с периодом  $T = \tau_1 + \tau_2$ , зависящим от параметров элементов схемы.

Длительность интервала  $\tau_1$  можно определить из формулы, описывающей процесс разряда конденсатора, учитывая, что напряжение на нем к концу этого промежутка времени станет равным  $U_0'$ . Таким образом

$$U_0' = U_{\text{ввх}}^0 + (U_0'' - U_{\text{ввх}}^0) e^{-\tau_1/RC}, \quad \text{откуда} \quad e^{-\tau_1/RC} = \frac{(U_0' - U_{\text{ввх}}^0)}{(U_0'' - U_{\text{ввх}}^0)} \quad \text{и}$$

$$\tau_1 = RC \ln \frac{(U_0'' - U_{\text{ввх}}^0)}{(U_0' - U_{\text{ввх}}^0)}.$$

Длительность  $\tau_2$  положительного импульса определяется из соотношения, описывающего заряд конденсатора, с учетом того, что он начинается от исходного напряжения  $U_0'$  под действием разности потенциалов  $U_{\text{ввх}}^1 - U_0''$ . Таким образом, в установившемся режиме

$$U_C(t) = U_0' + (U_{\text{ввх}}^1 - U_0'')(1 - e^{-t/RC}).$$

Так как за время  $\tau_2$  конденсатор зарядится до напряжения  $U_0''$ , то

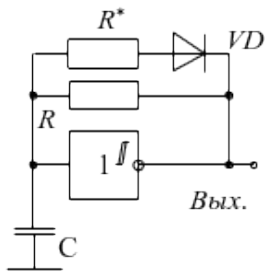
$$U_0'' = U_0' + (U_{\text{ввх}}^1 - U_0'')(1 - e^{-\tau_2/RC}), \quad \text{откуда} \quad 1 - e^{-\tau_2/RC} = \frac{(U_0'' - U_0')}{(U_{\text{ввх}}^1 - U_0'')},$$

и следовательно  $\tau_2 = RC \ln \frac{(U_{\text{ввх}}^1 - U_0')}{(U_{\text{ввх}}^1 - U_0'')}$ . Для триггера Шмидта

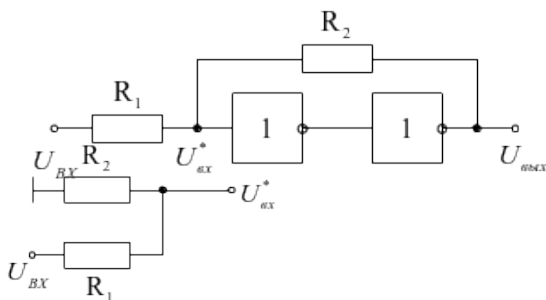
ТТЛ типа  $\tau_1 \approx 0,64RC$ ,  $\tau_2 \approx 0,42RC$ , при этом следует иметь ввиду, что сопротивление резистора должно быть ограничено сверху величиной порядка одного килоома.

Форма выходных колебаний такого генератора несимметрична, что объясняется соответствующей зависимостью длительностей интервалов  $\tau_1, \tau_2$  от порогов срабатывания триггера Шмидта. Симметрия будет наблюдаться лишь при выполнении условия  $U_0' + U_0'' \approx U_{\text{вых}}^1$ , которое не реализуется в рассмотренном триггере Шмидта. Сформировать колебания симметричной формы можно переключая постоянные времени  $\tau = RC$  при заряде и разряде конденсатора, либо схемотехнически реализовав триггера Шмидта с заданным соотношением порогов переключения.

В первом случае вводится цепочка из дополнительного резистора  $R^*$  и диода. В процессе заряда конденсатора напряжение на его верхней обкладке меньше, чем на выходе логического элемента, диод заперт и данная цепочка не влияет на работу схемы. Когда на выходе появляется нулевой уровень, напряжение на входе оказывается более положительным, диод открывается и сопротивление цепи становится равным  $R_{\text{разр}} \approx \frac{RR^*}{R + R^*} < R$ . Разряд конденсатора произойдет быстрее, что приведет к уменьшению  $\tau_1$ .



Так как гистерезисная характеристика в триггере Шмидта формируется за счет наличия положительной обратной связи, то его неинвертирующий вариант можно построить следующим образом.



Для переключения логического элемента из одного состояния в другое необходимо,

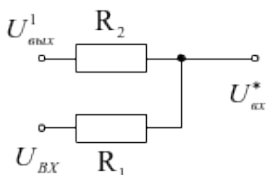


чтобы напряжение на его входе  $U^*$  стало бы больше, либо меньше порогового  $U_0$ . В данной схеме  $U^*$  будет зависеть не только от уровня входного сигнала, но и от состояния выхода второго инвертора. При нулевом выходном напряжении эквивалентная схема устройства без учета входного сопротивления логического элемента выглядит следующим

образом и  $U_{\text{вх}}^* = \frac{U_{\text{ВХ}}}{R_1 + R_2} R_2$ . Это напряжение достигнет порога

переключения  $U_0$  при  $U_{\text{ВХ}} = U_0'' = U_0 \left(1 + \frac{R_1}{R_2}\right)$ .

Если на выходе формируется сигнал логической единицы, то на вход инвертора через резистор  $R_2$  будет поступать напряжение высокого уровня и



$$U_{\text{ВХ}}^* = U_{\text{ввых}}^1 - IR_2 = U_{\text{ввых}}^1 - \frac{U_{\text{ввых}}^1 - U_{\text{ВХ}}}{R_1 + R_2} R_2.$$

В данном случае  $U_{\text{вх}}^*$  достигнет уровня порога переключения  $U^0$  вентиля,

если  $U_{\text{ВХ}} = U_0''$ , то есть,  $U_0 = U_{\text{ввых}}^1 - \frac{U_{\text{ввых}}^1 - U_0''}{R_1 + R_2} R_2$ . Приведа это

выражение к общему знаменателю, получим  $U_0 R_2 + U_0 R_1 = U_{\text{ввых}}^1 R_1 + \cancel{U_{\text{ввых}}^1 R_2} - \cancel{U_{\text{ввых}}^1 R_2} + U_0'' R_2$ , откуда

$U_0' = U_0 \left(1 + \frac{R_1}{R_2}\right) - U_{\text{ввых}}^1 \frac{R_1}{R_2}$ . Ширина петли гистерезиса в таком

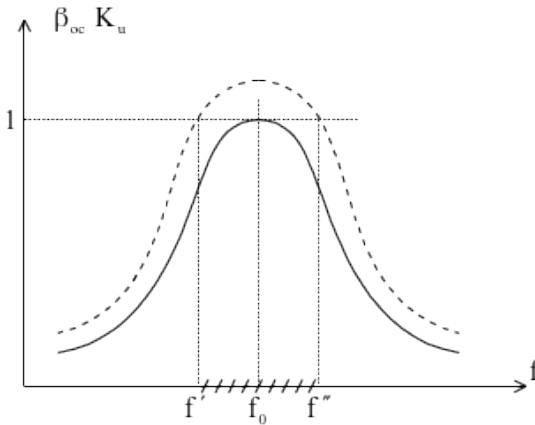
устройстве составит  $\Delta U = U_0'' - U_0' = U_{\text{ввых}}^1 \frac{R_1}{R_2}$ . В зависимости от

типа используемых логических элементов на величины сопротивлений резисторов должны накладываться соответствующие ограничения.

Особенностью всех рассмотренных генераторов является относительно низкая стабильность частоты формируемых сигналов. Одной из причин этого является сильная зависимость параметров логических элементов, а также резисторов и конденсаторов от температуры. В то же время для построения измерительных устройств, систем обмена данными, узлов синхронизации требуются источники высокостабильных колебаний, частоты которых должны очень слабо зависеть от изменений температуры, и от влияния иных дестабилизирующих факторов.

Как уже отмечалось, если в системе с усилителем, охваченным положительной обратной связью, условия самовозбуждения выполнены на одной частоте, то будет генерироваться синусоидальный сигнал с этой частотой. Данное условие можно обеспечить, используя в цепи обратной связи элементы с избирательными свойствами, которыми, в частности, обладает колебательный контур.

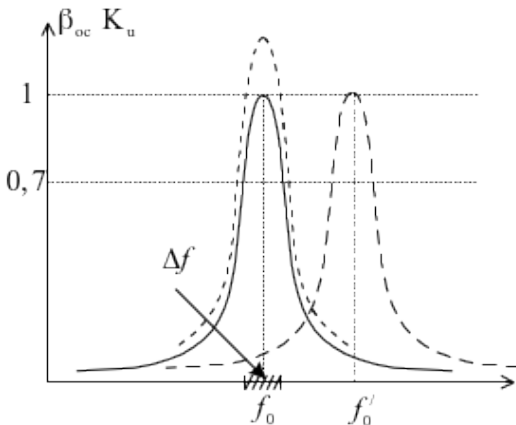
При использовании его в цепи обратной связи, подбором коэффициента усиления усилителя можно добиться, чтобы условие  $\beta_{oc}K_U = 1$  выполнялось только на одной частоте  $f_0$ , на которой и возникнет генерация. Графически это означает, что кривая, соответствующая зависимости  $\beta_{oc}K_U$  от частоты касается горизонтальной линии с ординатой равной единице лишь в одной точке.



Если по каким-либо причинам коэффициент усиления  $K_U$  возрастет, то соответствующий график пойдет выше и условия самовозбуждения станут выполняться для множества

частот, лежащих в пределах от  $f'$  до  $f''$  (в случае выполнения для них условия самовозбуждения по фазе). Это приведет к появлению в спектре выходного сигнала дополнительных компонент и отличию его формы от синусоидальной.

С другой стороны данную ситуацию можно рассматривать, как способность частоты принимать любое значение в диапазоне  $f' \div f''$ . Таким образом, если, к примеру с ростом температуры будут меняться условия самовозбуждения по фазе, то начнет изменяться и частота генерируемого сигнала.



Получить стабильное значение частоты можно, если условия самовозбуждения выполняются для очень узкой полосы частот, и центральная частота этого диапазона остается неизменной при воздействии

дестабилизирующих факторов. Первое условие можно реализовать, используя в цепи обратной связи высокоизбирательные элементы, имеющие узкую резонансную характеристику, то есть обладающие высокой добротностью  $Q = \frac{f_0}{\Delta f}$ , где  $\Delta f$  - ширина полосы пропускания по уровню 0,7. Из второго условия следует, что параметры избирательной системы должны быть неизменными, то есть стабильными.

Этим требованиям удовлетворяют так называемые кварцевые резонаторы. Они представляют собой пластинки из кристаллов кварца, на поверхностях которых нанесены металлические контактные площадки. Наличие у кварца пьезоэлектрических свойств приводит к тому, что при сжатии или растяжении вырезанной из него пластинки на ее краях появится разность потенциалов того либо иного знака. Если же к металлизированным краям пластинки приложить напряжение, то она сожмется или растянется в зависимости от его полярности.

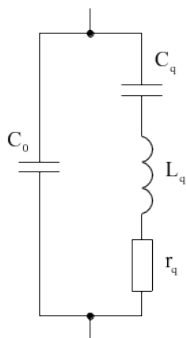
Кроме того, пластинка из кварца обладает свойством упругости, то есть после кратковременного воздействия электрического поля, она вернется в исходное состояние не сразу, а после достаточно большого числа колебаний, то есть изменений геометрических размеров. Частота этих колебаний определяется геометрическими размерами пластинки, а время их затухания - добротностью, которая вследствие малых внутренних потерь в кварце достигает очень больших значений.

При изменениях размеров кварцевой пластинки на ее обкладках будет возникать переменная разность потенциалов, то есть электрические колебания с частотой, равной частоте механического резонанса. Если это напряжение усилить и в фазе вновь подать на электроды кварцевого резонатора, то в такой системе установится режим непрерывной генерации

электрических колебаний с частотой  $f_0 \approx \frac{1600 \div 3600}{d} \text{кГц}$ , где  $d$  - толщина пластинки в миллиметрах. Конкретное значение частоты зависит как от направления под которым вырезается пластинка по отношению к кристаллографическим осям, так и от типа колебаний, которые могут быть колебаниями растяжения – сжатия, кручения, изгиба и т.п.

Вследствие того, что коэффициент линейного расширения кварца очень мал, размеры кварцевой пластинки при изменении температуры, а следовательно и ее резонансная частота будут меняться слабо,  $\frac{\Delta f}{f_0 \Delta t^0}$  составляет

$10^{-6} \div 10^{-8} \text{ 1/град}$ . Для обычных колебательных



контуров эта величина на несколько порядков больше.

Кварцевый резонатор при определенных условиях ведет себя подобно колебательному контуру и может быть представлен в виде следующей эквивалентной схемы, где  $C_q, L_q$

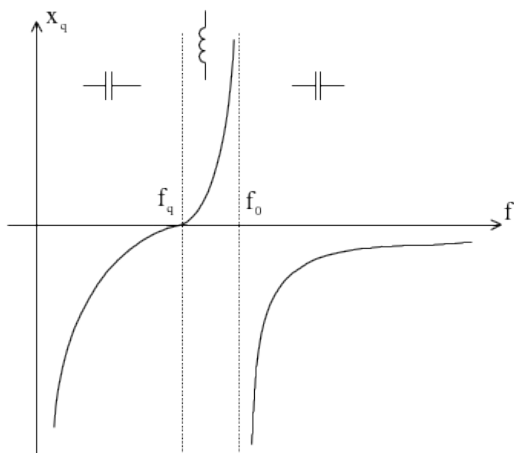
- эквивалентные емкость и индуктивность,  $r_q$  - сопротивление потерь, а  $C_0$  - емкость

электродов и внешних проводников. Параметры этих элементов зависят от резонансной частоты кварца, типа среза и вида колебаний. В частности для кварцевого резонатора с частотой 1 МГц, они имеют значения  $C_q \approx 0,015 \text{ нФ}$ ,  $L_q \approx 1,5 \text{ Гн}$ ,  $r_q \approx 5 \text{ Ом}$ ,  $C_0 \approx 5 \text{ нФ}$ . При этом

добротность  $Q = \frac{\sqrt{L_q / C_q}}{r_q}$  оказывается порядка  $2 \cdot 10^6$ , в то время

как у обычного колебательного контура она не превышает нескольких сотен.

В эквивалентной схеме кварцевого резонатора можно выделить два колебательных контура – один



последовательный с резонансной частотой

$$f_q = \frac{1}{2\pi\sqrt{L_q C_q}}, \quad \text{а}$$

другой параллельный с частотой резонанса

$$f_0 = \frac{1}{2\pi\sqrt{L_q \frac{C_q C_0}{C_q + C_0}}}. \quad \text{На}$$

частоте последовательного резонанса полное сопротивление

резонатора близко к нулю, а на частоте параллельного стремится к бесконечности.

Графики зависимости полного сопротивления кварцевого резонатора от частоты имеют следующий вид. В промежутке  $f_q \div f_0$  оно имеет индуктивный характер, а в остальной области частот – емкостной.

Противоположные значения потенциалов на краях кварцевой пластинки возникают, если по ее толщине  $d$  укладывается нечетное количество полуволн деформации. При скорости их распространения  $v$ , длины волн

определяются соотношением  $\lambda = \frac{v}{f}$  и условия генерации

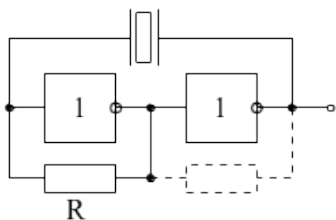
будут выполняться если  $d = (2k + 1)\frac{\lambda}{2}$ . То есть автогенератор с

одним и тем же кварцевым резонатором может формировать сигнал как с частотой  $f = \frac{v}{2d}$ , так и с частотами  $3f, 5f$  и т.д.

Если условия самовозбуждения выполняются лишь для одной из них, то будет генерироваться синусоидальный сигнал, а если для всех, то близкий к прямоугольному. Эквивалентная схема кварцевого резонатора для всех этих частот выглядит

одинаково, однако параметры элементов, входящих в ее состав будут зависеть от конкретного значения частоты.

Один из вариантов кварцевого генератора на логических элементах представляет собой мультивибратор, у которого конденсатор заменен кварцевым резонатором. Резистор выводит инверторы в активный режим и в схеме возникает колебательный процесс на частоте последовательного резонанса кварцевой пластинки.



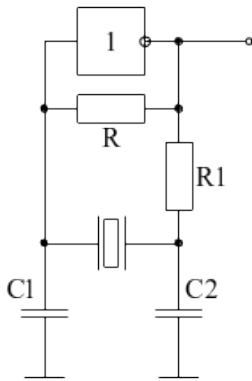
Это связано с тем, что только на этой частоте фазовый сдвиг в цепи обратной связи равен нулю и эквивалентное сопротивление кварцевого резонатора минимально (примерно равно  $r_q$ ), то есть сигнал

положительной обратной связи имеет максимальный уровень. Так как дополнительных избирательных цепей в такой схеме нет, то колебания возникнут на всех резонансных частотах кварца и их форма на выходе генератора будет прямоугольной. В ряде случаев в схеме параллельно второму инвертору устанавливают дополнительный резистор, что облегчает процесс запуска генератора.

Такая схема будет работоспособна при выполнении условия  $r_q \ll R$ , которое при использовании ТТЛ элементов выполняется лишь для кварцевых резонаторов с частотами в единицы мегагерц и более. Это объясняется тем, что с уменьшением резонансной частоты эквивалентное сопротивление  $r_q$  возрастает, а величина сопротивления  $R$  ограничена сверху значениями в сотни Ом. При использовании КМОП вентиля данные ограничения отсутствуют, но возникают определенные проблемы с генерацией высокочастотных колебаний вследствие невысокого быстродействия таких логических элементов.

Одна из разновидностей схем низкочастотных кварцевых генераторов на КМОП логических элементах

называется схемой Пирса и требует для своего построения лишь одного инвертора, работающего в активном режиме, который обеспечивается введением резистора  $R$  сопротивлением в несколько мегаом. Данное устройство представляет собой вариант генератора по схеме емкостной трехточки. Здесь условия самовозбуждения будут выполняться, если реактивное сопротивление между конденсаторами имеет индуктивный характер. В этом случае фазовый сдвиг в цепи обратной связи в определенной полосе частот становится равным  $\pi$ , то есть в схеме действует положительная обратная связь.



Сопротивление кварцевого резонатора имеет индуктивный характер лишь в диапазоне частот  $f_q \div f_0$ , который очень мал вследствие того, что  $C_0 \gg C_q$ . Данная схема будет генерировать последовательность прямоугольных импульсов с частотой следования  $f_q \leq f \leq f_0$ . Подбором емкостей конденсаторов частоту можно в небольших пределах регулировать, так как эквивалентная емкость  $C = \frac{C_1 C_2}{C_1 + C_2}$  добавляется к емкости резонатора  $C_0$ , что несколько уменьшает значение частоты  $f_0$ .

Резистор  $R1$  имеет сопротивление порядка сотен килом и служит для того, чтобы относительно низкое выходное сопротивление инвертора не шунтировало низкочастотный кварцевый резонатор с большим значением  $r_q$ .



## Список литературы.

- Зельдин Е.А. Цифровые интегральные микросхемы в информационно-измерительной аппаратуре.- Л.: Энергоатомиздат, 1986.- 280 с., ил.
- Цифровая вычислительная техника. Учебник для вузов. Под. Ред. Евреинова. – М.: Радио и связь, 1991.- 464 с., ил.
- Опадчий Ю.Ф. и др. Аналоговая и цифровая электроника (Полный курс): Учебник для вузов. - М.: Горячая линия – Телеком, 2000. – 768 с.: ил.
- Шило В.Л. Популярныe цифровые микросхемы. Справочник М.: Радио и связь, 1987.- 252 с., ил.
- Преснухин Л.Н. Расчет элементов цифровых устройств. Учеб. Пособие. 2-е изд., перераб. и доп.- М.: Высш. шк., 1991.
- Цифровые устройства на комплементарных МДП интегральных микросхемах.- М.: Радио и связь, 1983.- 272 с., ил.
- Угрюмов Е.П. Цифровая схемотехника. – СПб.: БХВ – Санкт – Петербург, 2000.- 528 с.: ил.
- Проектирование импульсных и цифровых устройств радиотехнических систем: Учеб. пособие для радиотехнич. спец. /вузов./ Под ред. Казаринова.- М.: Высш. шк., 1985.- 319 с., ил.
- Потемкин И.С., Функциональные узлы цифровой автоматики – М.: Энергоатомиздат, 1988. – 320 с.: ил.
- Бунтов В.Д., Емельянов А.Ф., Макаров С.Б. Проектирование цифровых радиотехнических устройств на интегральных схемах. Учебное пособие. – Л.:ЛПИ, 1985. – 72 с.

## ОГЛАВЛЕНИЕ

	Стр.
1. Арифметические и логические основы цифровой техники.....	3
2. Классификация цифровых устройств.....	30
3. Состязания в комбинационных схемах.....	33
4. Комбинационные цифровые устройства.....	44
4.1. Дешифраторы.....	46
4.2. Преобразователи кодов.....	59
4.3. Шифратор приоритетов.....	63
4.4. Мультиплексоры.....	71
4.5. Схемы контроля четности.....	80
4.6. Мажоритарные элементы.....	85
4.7. Цифровые компараторы.....	88
4.8. Сумматоры двоичных кодов чисел.....	92
4.9. Устройства для вычитания двоичных кодов чисел.....	106
4.10. Сумматоры двоично-десятичных кодов чисел.....	110
4.11. Арифметико-логические устройства.....	114
4.12. Умножители двоичных кодов чисел.....	117
5. Формирователи и генераторы на логических элементах.....	120
5.1. Формирователи импульсов.....	120
5.2. Генераторы цифровых сигналов.....	129
Список литературы.....	152





**ЛЕХИН Сергей Никифорович**

## **СХЕМОТЕХНИКА ЭВМ**

### **ТЕОРЕТИЧЕСКИЙ КУРС**

#### *часть 2*

**Учебное пособие**

Для студентов специальности 220100  
Вычислительные машины, комплексы, системы и сети

Технический редактор Л.Г. Голохвастова  
Компьютерная верстка С.Н. Лехин

Лицензия ЛР № 020593 от 07.08.97

---

Формат 60×84/16. Печать офсетная.

Гарнитура Times New Roman . Уч. изд. п.л. .

Тираж 150 экз. Заказ №

Псковский политехнический институт (филиал) Санкт-Петербургского  
государственного политехнического университета.

Издательство СПбГПУ, член Издательско-полиграфической  
ассоциации вузов Санкт-Петербурга.

Адрес университета и издательства:

Россия, 195251, Санкт-Петербург, ул. Политехническая, 29

Отпечатано с готового оригинал-макета, предоставленного автором