

Н. И. Сорока, Г. А. Кривинченко

# ТЕЛЕМЕХАНИКА

Конспект лекций  
для студентов специальности  
1-53 01 07 «Информационные технологии и управление  
в технических системах»  
всех форм обучения

Часть 2

## Коды и кодирование



Минск



## ВВЕДЕНИЕ

В настоящее время широкое применение нашли цифровые телемеханические системы, в которых измеряемая величина передается в виде определенной кодовой комбинации. Цифровые методы передачи информации по сравнению с другими имеют ряд преимуществ. Главными из них являются следующие:

- 1) прием сигнала сводится не к измерению, а к обнаружению 1 или 0;
- 2) сообщения в цифровой форме легко обрабатываются, запоминаются, коммутируются и регистрируются;
- 3) возможна многократная передача без накопления ошибок;
- 4) применение помехоустойчивого кодирования позволяет значительно увеличить достоверность передачи телемеханических сообщений;
- 5) упрощаются требования, предъявляемые к радиолиниям в отношении калибровки эталонных уровней;
- 6) улучшается использование канала связи в случае применения специальных кодов, статистически согласованных с передаваемыми сообщениями.

Под кодированием в широком смысле понимается переход от одного способа задания информации к другому, допускающий восстановление исходной информации. Теория кодирования получила большое развитие, начиная с 40-х годов XX века после работ К.Шеннона.

В данном конспекте большое внимание уделено теоретическим основам построения кодовых комбинаций, а также преобразованию кода передаваемой и обрабатываемой информации с сохранением его числового эквивалента. Преобразование может осуществляться программным или аппаратным способом. Программный способ отличается универсальностью и высокой производительностью, но требует определенных затрат машинного времени и дополнительно загружает память ЭВМ, что отрицательно сказывается на выполнении машиной других операций. В последние годы большое значение придается аппаратному (схемотехническому) способу преобразования кодов, что связано в первую очередь с разработкой специализированных микросхем, а также интегральных схем среднего и большого уровней интеграции. В разделах данного конспекта основное внимание уделено именно этому перспективному способу преобразования кода в код. Преобразователи могут быть последовательного и параллельного типов. В преобразователях второго типа предусматривается параллельный ввод всех разрядов преобразуемого кода и последующее выполнение комбинационной логикой операций преобразования по алгоритму, задаваемого таблицей истинности. Метод отличается высоким быстродействием, гибкостью и приемлемыми затратами. При параллельном вводе информации для преобразования кодов могут использоваться не только комбинационные схемы, но и элементы памяти, чему в конспекте уделено соответствующее внимание.

Материалы данного конспекта будут полезны студентам при выполнении ими курсового проекта по телемеханике.

# 1. КОДЫ И КОДИРОВАНИЕ

## 1.1. Основные понятия

**Кодирование** – преобразование дискретного сообщения в дискретный сигнал, осуществляемое по определенному правилу. Восстановление дискретного сообщения по сигналу на выходе дискретного канала, осуществляемое с учетом правил кодирования, называется **декодированием**.

**Код** (от лат. *codex* – свод законов) есть совокупность условных сигналов, обозначающих дискретные сообщения.

**Кодовая последовательность (комбинация)** – представление дискретного сигнала.

Целями кодирования сообщений обычно являются:

- 1) передача по общему каналу связи нескольких или многих сообщений для кодового разделения сигналов;
- 2) повышение помехоустойчивости и достоверности передачи сообщений;
- 3) более экономное использование полосы частот канала связи, т.е. уменьшение избыточности;
- 4) уменьшение стоимости передачи и хранения сообщений;
- 5) обеспечение скрытности передачи и хранения информации;
- 6) преобразование любой информации независимо от ее происхождения и назначения в единую систему символов;
- 7) приведение исходных символов в соответствие с характеристиками канала связи.

Любая кодовая комбинация содержит определенный набор элементов или символов (1 и 0, а и б), которые называются **буквами алфавита**, а весь набор букв образует **алфавит кода**. Для двоичного кода алфавит состоит из двух символов, для троичного их число увеличивается до трех (а, б, в или 1, 2, 3), а в десятичном оно равно десяти. Таким образом, **основание кода  $X$**  – это количество признаков или число букв (цифр). Кодовая комбинация, составленная из  $n$  символов или  $n$  элементов, называется **кодовым словом (кодовым блоком)**, имеющим длину  $n$  или число разрядов  $n$ . Если длина всех кодовых комбинаций одинакова, то такие коды называют **равномерными (комплектными)**. Например, код 001, 011, 101 является комплектным, а код 1, 11, 101 – **некомплектным**. В телемеханике обычно используют только равномерные коды.

Кроме указанных характеристик, коды имеют и другие характеристики, которые приведены на рис. 1.1.

Для передачи различных символов, составляющих алфавит кода, могут использоваться импульсы с различными признаками (табл. 1.1).

Передачу кодовых комбинаций можно осуществить последовательно во времени или параллельно, т.е. одновременно во времени. В последнем случае передача должна осуществляться по нескольким проводам или с использованием частотных признаков для разделения элементарных сигналов.

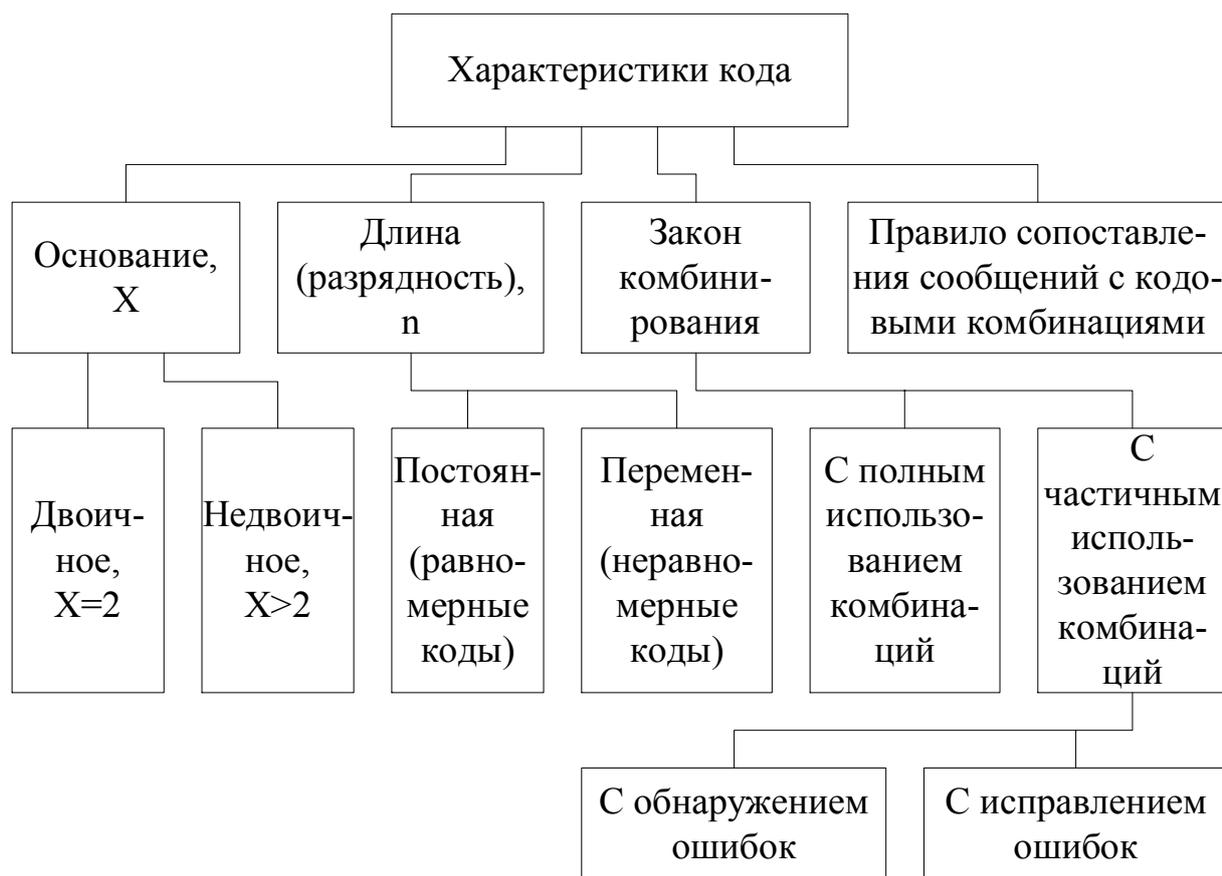


Рис. 1.1. Классификация характеристик кода

Таблица 1.1

Импульсные признаки, используемые для передачи двоичных кодов

Символ	Амплитудные		Временные			Полярные	Частотные		Фазовые
1									
0									

На рис. 1.2 показана последовательная передача кодовой комбинации 1011001 видеоимпульсами, а на рис. 1.3 – передача этой же комбинации радиоимпульсами. В обоих случаях передача осуществляется с пассивными паузами между элементами кодовых комбинаций.

Для передачи кодовых комбинаций параллельно во времени каждому разряду присваивается своя частота (табл. 1.2). Однако признаки у каждого разряда должны быть не частотными, а амплитудными или временными.



Рис. 1.2. Последовательная передача кодовой комбинации видеоимпульсами

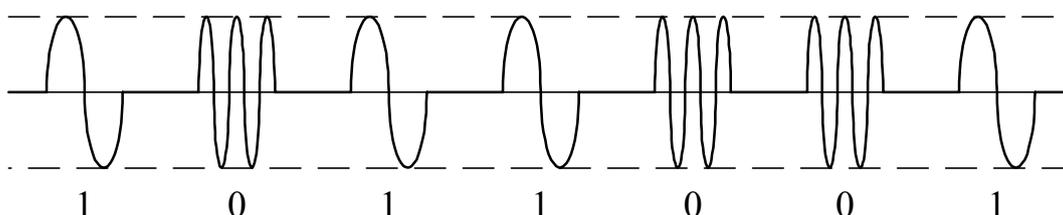


Рис. 1.3. Последовательная передача кодовой комбинации радиоимпульсами

Таблица 1.2

Параллельная передача кодовых комбинаций

Номер разряда	Частота	Номер кодовой комбинации и время её передачи	
		1- $t_1$	2- $t_2$
1	$f_1$	1	1
2	$f_2$	0 —	1
3	$f_3$	0 —	1
4	$f_4$	1	0 —

Первая кодовая комбинация 1001 передается в течение первого интервала времени  $t_1$  частотами  $f_1$  и  $f_4$ , посылаемыми одновременно, а вторая – 1110 передается в течение второго интервала времени одновременной посылкой частот  $f_1, f_2$  и  $f_3$ .

По способу образования кодовых комбинаций коды разделяются на **числовые** и **нечисловые**. В числовых кодах, получивших название **цифровых**, кодовые комбинации образуют ряд возрастающих по весу чисел, определяемый системой счисления. Они применяются в системах измерений, кон-

троля, ЭВМ и т.д. Нечисловые (невзвешенные) коды не имеют систем счисления и применяются в системах управления и телеуправления, где команды и сигналы независимы.

## 1.2. Цифровые коды

В основу правил соответствия кодовых комбинаций числам цифровых кодов положены математические системы счисления, поэтому данные коды называются также арифметическими или взвешенными.

**1.2.1. Запись кодовых комбинаций в виде многочлена.** Любое число в системе счисления с основанием  $X$  можно представить в виде многочлена. Так,  $n$  – разрядное число запишется в виде

$$F(X) = \sum_{i=0}^{n-1} A_i X^i = A_{n-1} X^{n-1} + A_{n-2} X^{n-2} + \dots + A_0 X^0, \quad (1.1)$$

где  $A$  – цифровые коэффициенты, имеющие значения от 0 до  $X - 1$ .

В десятичной системе ( $X = 10$ )

$$F(10) = \sum_{i=0}^{n-1} A_i 10^i. \quad (1.2)$$

Так, число 1408 запишется следующим образом:

$$1408 = 1 \cdot 10^3 + 4 \cdot 10^2 + 0 \cdot 10^1 + 8 \cdot 10^0.$$

В двоичной системе счисления

$$F(2) = \sum_{i=0}^{n-1} A_i 2^i. \quad (1.3)$$

Так, десятичное число 47 запишется следующим образом:

$$47 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

или в виде многочлена

$$G(X) = 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^5 + x^3 + x^2 + x^1 + 1. \quad (1.4)$$

Таким образом, члены многочленов записываются только при наличии коэффициента единицы.

**1.2.2. Сложение.** Над многочленами можно производить все алгебраические операции. Обычное сложение с переносом числа в высший разряд здесь неприменимо, так как это может привести к образованию более высокого разряда, чем принято в данном коде, что недопустимо. Поэтому применяется так

называемое сложение двоичных чисел по модулю 2, обозначаемое знаком  $\oplus$ . При двух слагаемых правила сложения следующие:  $0 \oplus 0 = 0$ ;  $0 \oplus 1 = 1$ ;  $1 \oplus 0 = 1$ ;  $1 \oplus 1 = 0$ .

При сложении многозначных чисел складывают разряды, занимающие одинаковые места. При этом сложение сводится к сложению только коэффициентов при членах совпадающих степеней.

Если складываются несколько чисел, то четное число единиц в сумме дает нуль, а сумма нечетного числа единиц приравнивается к единице. Иногда в результате сложения нескольких чисел сумма выражается меньшим двоичным числом, чем какое-либо из слагаемых. Для примера произведем сложение следующих многочленов:

$$x^6 + x^5 + x^3 + x^2 + 1; x^5 + x^4 + x^2; x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + 1.$$

Выразим эти многочлены в двоичных числах и, расположив их соответствующим образом в столбцы, произведем сложение:

$$\begin{array}{r} x^6 + x^5 + 0 + x^3 + x^2 + 0 + 1 \rightarrow 1101101 \rightarrow 109_{10} \\ \oplus \\ 0 + x^5 + x^4 + 0 + x^2 + 0 + 0 \rightarrow 0110100 \rightarrow 52_{10} \\ \oplus \\ x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \rightarrow 1111111 \rightarrow 127_{10} \\ \hline 0 + x^5 + 0 + 0 + x^2 + x + 0 \rightarrow 0100110 \rightarrow 38_{10} \end{array}$$

**1.2.3. Умножение.** Для того чтобы при умножении многочленов не увеличилась разрядность степени многочлена выше заданной, производят так называемое символическое умножение, или умножение в конечном поле двоичных чисел, состоящее из двух этапов. Первый этап заключается в умножении многочленов по обычным правилам алгебры, за исключением сложения, которое производится по модулю 2. Перемножим два многочлена:

$$\begin{array}{r} x^6 + x^5 + x^3 + x^2 + 1 = \quad 1101101 \\ \times \quad \quad \quad \quad \quad \quad \quad \times \\ x^5 + x^4 + x^2 = \quad 110100 \\ \hline x^8 + x^7 + 0 + x^5 + x^4 + 0 + x^2 \quad 1101101 \\ x^{10} + x^9 + 0 + x^7 + x^6 + 0 + x^4 \quad 1101101 \\ x^{11} + x^{10} + 0 + x^8 + x^7 + 0 + x^5 \quad 1101101 \\ \hline x^{11} + 0 + x^9 + 0 + x^7 + x^6 + 0 + 0 + 0 + x^2 = \quad 101011000100 \end{array}$$

Произведем теперь умножение многочлена на  $x^n$ . Например,  $(x^5 + x^4 + x^2) \times x^3 = x^8 + x^7 + x^5$ . В результате умножения степень каждого члена многочлена повышалась на  $n$ . В двоичной форме записи  $110100 \times 1000 = 110100000$ . Таким образом, умножение многочлена на  $x^n$  означает приписывание справа  $n$  нулей.

**1.2.4. Деление.** При делении в двоичной записи делитель умножается на частное и подписывается под делимым так, чтобы совпадали старшие разряды. В частное записывается единица. Для нахождения первого остатка из делимого вычитается делитель (что эквивалентно их сложению по модулю 2) и к остатку справа сносится очередной разряд делимого. Далее под первым остатком снова подписывается делитель и в частное приписывается еще одна единица, если число разрядов в остатке равно числу разрядов делителя. В противном случае в частном записывается нуль и к остатку подписывается очередной член делимого. Деление продолжается до тех пор, пока степень остатка не станет меньше степени делителя, т.е. число разрядов остатка не окажется меньше числа разрядов делителя. Например,

$$\begin{array}{r}
 110110101 \quad | \quad 101011 \\
 \underline{101011} \quad \vdots \quad | \quad 1110 \rightarrow x^4 + x^2 + x \text{ — частное } Q(X) \\
 111011 \quad \vdots \\
 \underline{101011} \quad \vdots \\
 100000 \quad \vdots \\
 \underline{101011} \quad \vdots \\
 10111 \rightarrow x^4 + x^2 + x + 1 \text{ — остаток}
 \end{array}$$

При составлении циклических кодов необходимо уметь находить только остатки без определения частного. Ниже дается пример нахождения нескольких остатков при делении единицы с нулями на случайно выбранный многочлен  $P(X) = x^3 + x + 1$ . Следует помнить, что число разрядов у остатков на единицу меньше, чем у делителя.

$$\begin{array}{r}
 100000000 \quad | \quad 1011 \\
 \underline{1011} \quad \vdots \\
 01100 \quad \vdots \quad \text{— первый остаток } R_1(X) = 011 \\
 \underline{1011} \quad \vdots \quad \text{— второй остаток } R_2(X) = 110 \\
 1110 \quad \vdots \quad \text{— третий остаток } R_3(X) = 111 \\
 \underline{1011} \quad \vdots \\
 1010 \quad \vdots \quad \text{— четвертый остаток } R_4(X) = 101 \\
 \underline{1011} \quad \vdots \\
 0010 \quad \vdots \quad \text{— пятый остаток } R_5(X) = 001 \\
 \quad \quad \quad \vdots \quad \text{шестой остаток } R_6(X) = 010 \\
 \quad \quad \quad \vdots \quad \text{седьмой остаток } R_7(X) = 100
 \end{array}$$

Дальнейшее деление нецелесообразно, так как остатки начнут повторяться.

**1.2.5. Перенос слагаемых.** Понятие отрицательной цифры при операциях в конечном поле двоичных чисел отсутствует, так как это привело бы к увеличению признаков с двух до трех, т. е. к троичной системе счисления. Поэтому перенос слагаемых из одной части в другую производится без изменения знака. Например, справедливо как выражение  $(x^4+x+1)+(x^3+x)=x^4+x^3+1$ , так и выражение, отличающееся тем, что второе слагаемое левой части перенесено в правую без изменения знака, т. е.  $(x^4+x+1) = (x^3+x)+(x^4+x^3+1)$ . Справедливость этих равенств проверяется сложением по модулю 2 одночленов с одинаковыми степенями.

**1.2.6. Матричная запись кодовых комбинаций.** Всю совокупность комбинаций  $n$ -разрядного двоичного кода, насчитывающего  $2^n$  различных комбинаций, можно записать в виде матрицы, содержащей  $2^n$  строк и  $n$  столбцов. Так, все комбинации трехразрядного кода запишутся в матрице  $a$ :

$$\begin{array}{l}
 \text{а) } \left\| \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array} \right\| \quad
 \text{б) } \oplus \begin{array}{r} 111 \\ 110 \\ \hline 001 \end{array} \quad
 \text{в) } \oplus \begin{array}{r} 001 \\ 011 \\ \hline 010 \end{array} \quad
 \text{г) } \oplus \begin{array}{r} 001 \\ 010 \\ \hline 011 \end{array} \quad
 \text{д) } \oplus \begin{array}{r} 010 \\ 110 \\ \hline 100 \end{array} \quad
 \text{е) } \oplus \begin{array}{r} 011 \\ 110 \\ \hline 101 \end{array} \\
 \text{ж) } \oplus \begin{array}{r} 010 \\ 100 \\ \hline 110 \end{array} \quad
 \text{з) } \oplus \begin{array}{r} 001 \\ 110 \\ \hline 111 \end{array} \quad
 \text{и) } \left\| \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right\| \quad
 \text{к) } \left\| \begin{array}{ccc} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{array} \right\|
 \end{array}$$

Если взять любые две или более строки матрицы  $a$  и сложить их по модулю 2, то получим одну из остальных строк, записанных в этой матрице (пункты б-з). Например, складывая вторую и третью строки, получим четвертую строку (пункт г). Из матрицы  $a$  можно выбрать комбинации, состоящие из одной единицы. Такие комбинации образуют матрицу, называемую единичной матрицей (матрица и). Матрица  $k$  является транспонированной единичной матрицей, т. е. зеркальным отображением матрицы  $i$ . Интересным свойством обладает единичная матрица  $i$ : если сложить по модулю 2 в различном сочетании строки, то получим все остальные строки матрицы  $a$  без нулевой.

При исследовании кодов иногда оказывается полезным графическое и геометрическое представление кодов.

**1.2.7. Графическое представление кода** часто указывает пути и методы кодирования и декодирования комбинаций и представляет собой древовидный график, состоящий из точек и расходящихся от них линий, заканчивающихся также точками. Точки графа называются вершинами, а соединяющие их линии – ребрами. Начальная вершина, от которой начинается расхождение ребер, называется корнем дерева, а число ребер, которое надо пройти от корня к некоторой вершине – порядком этой вершины. Максимальное число ребер, которые могут выходить из каждой вершины дерева, равно основанию кода, а макси-

мальный порядок вершин, которое оно содержит, равен максимальной длине кодовых комбинаций. Значения разрядов комбинации, приписываемой каждой вершине, соответствующей направлениям движения по ребрам от корня дерева к данной вершине. Ребра, ведущие от корня к вершинам первого порядка, определяют значение первого слева разряда комбинации; ребра, соединяющие вершины первого и второго порядков, дают значение второго разряда комбинации, и т.д.

На рис. 1.4 показано кодовое дерево для двоичного трехразрядного кода.

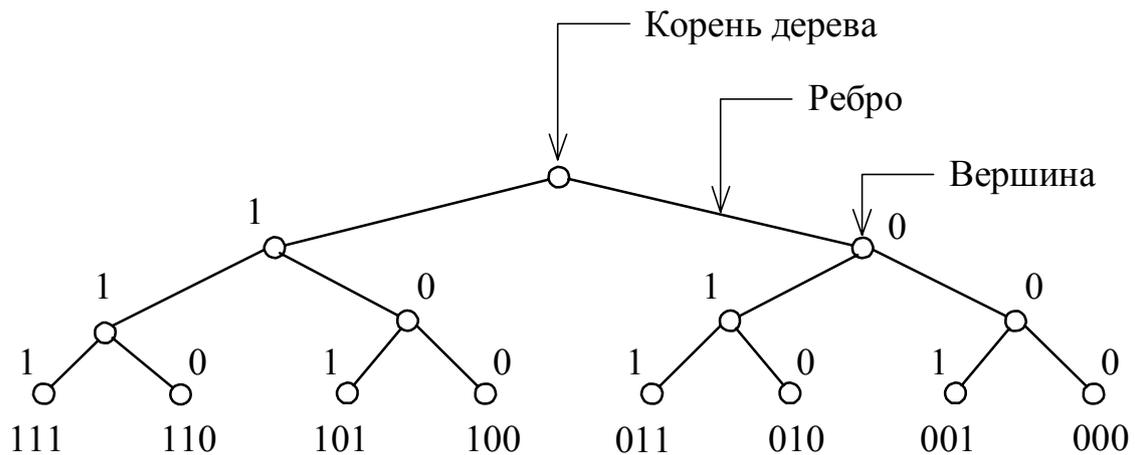


Рис. 1.4. Графическое представление кодового дерева

**1.2.8. Геометрическая модель кода** является более наглядной, чем графическое представление кода. Она дает наглядное представление о возможностях перехода одной комбинации в другую в результате искажения, и поэтому по ней легко судить о корректирующих возможностях кода, т.е. о его способности обнаруживать и исправлять ошибки.

Любая  $n$ -разрядная двоичная кодовая комбинация может быть интерпретирована как вершина  $n$ -мерного единичного куба, т.е. куба с длиной ребра, равной 1.

При  $n=2$  кодовые комбинации располагаются в вершинах квадрата (рис. 1.5), при  $n=3$  – в вершинах единичного куба (рис. 1.6).

В общем случае  $n$ -мерный единичный куб имеет  $2^n$  вершин, что равно наибольшему возможному числу кодовых комбинаций. Такая модель дает простую геометрическую интерпретацию и кодовому расстоянию  $d$  между отдельными кодовыми комбинациями. Оно соответствует наименьшему числу ребер единичного куба, которое необходимо пройти, чтобы попасть от одной комбинации к другой.

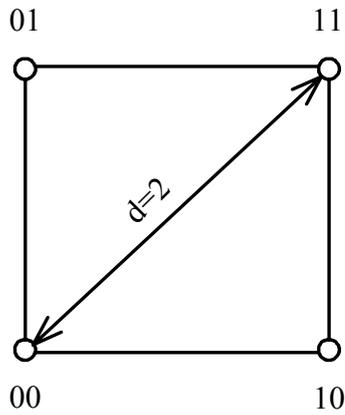


Рис. 1.5. Геометрическая модель двухразрядного двоичного кода

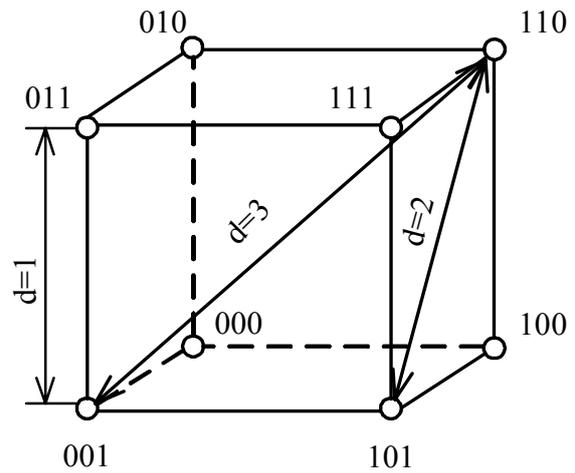


Рис. 1.6. Геометрическая модель трёхразрядного двоичного кода

На рис. 1.7 и 1.8 представлены геометрические модели троичного двухразрядного и трёхразрядного кодов соответственно.

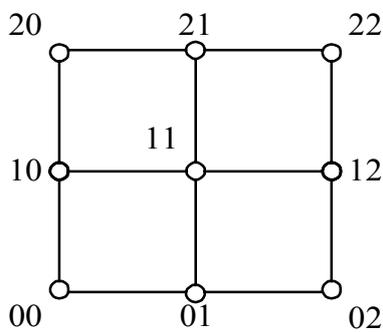


Рис. 1.7. Геометрическая модель троичного двухразрядного кода

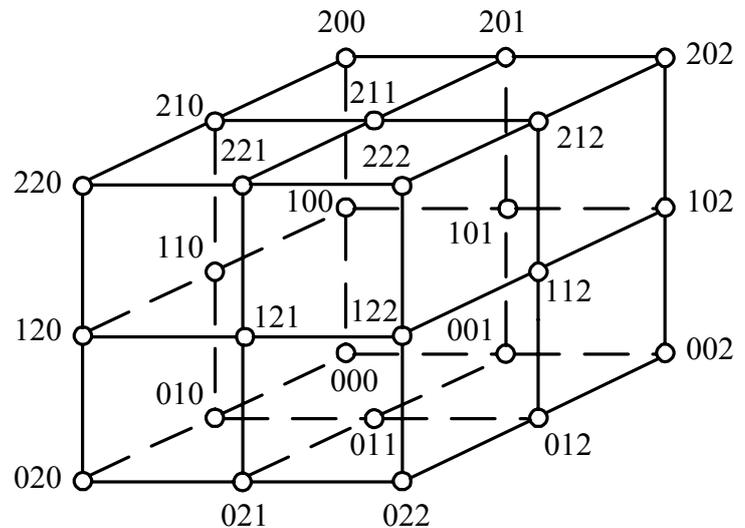


Рис. 1.8. Геометрическая модель троичного трёхразрядного кода

**1.2.9. Классификация двоичных кодов.** По возможности обнаружения и исправления ошибок различают простые и корректирующие коды. Дальнейшая классификация приведена на рис. 1.9.

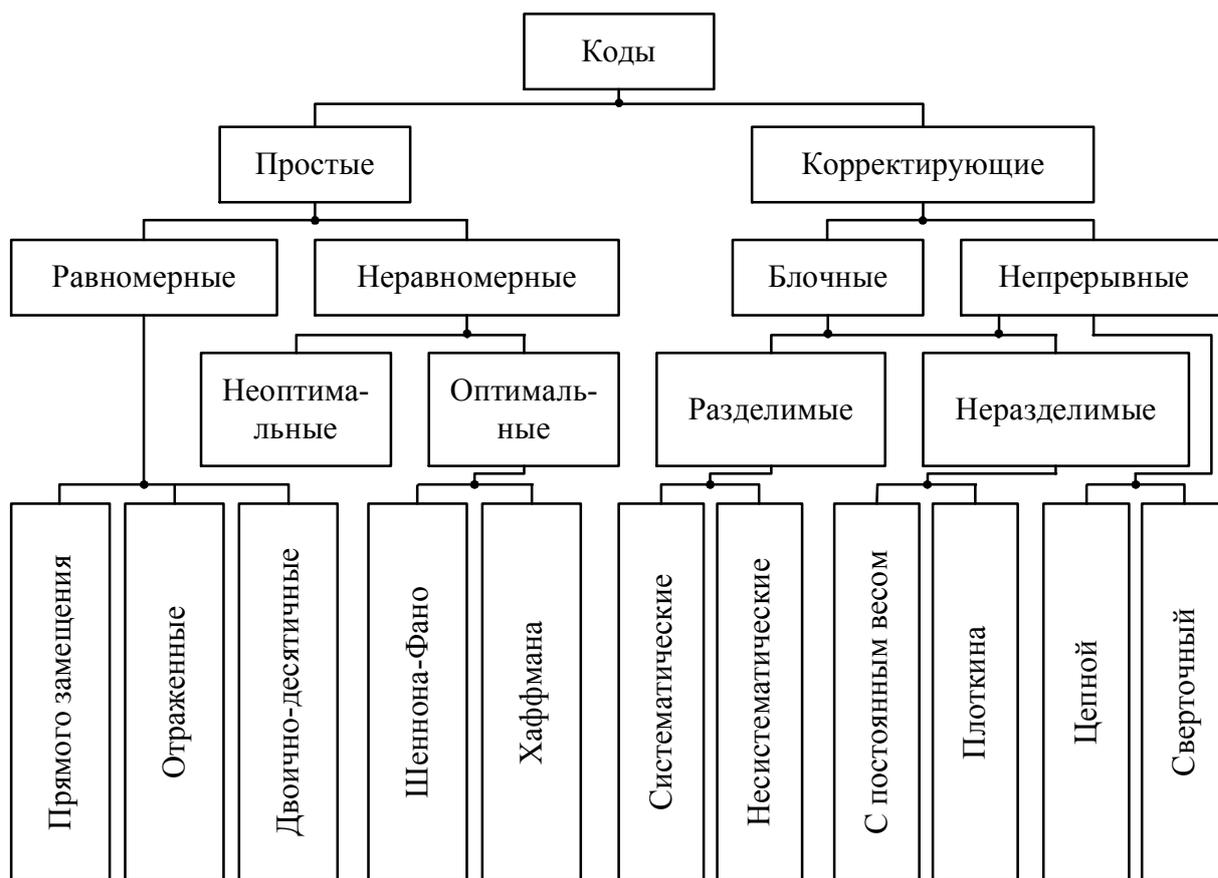


Рис. 1.9. Классификация двоичных кодов

Корректирующий код называется **блочным**, если каждая его комбинация имеет ограниченную длину, и непрерывным, если его комбинация имеет неограниченную, а точнее, полубесконечную длину.

Коды в зависимости от методов внесения избыточности подразделяются на *разделимые* и *неразделимые*. В разделимых кодах четко разграничена роль отдельных символов. Одни символы являются информационными, другие являются проверочными и служат для обнаружения и исправления ошибок. Разделимые блочные коды называются обычно  $n, k$  – кодами, где  $n$  – длина кодовых комбинаций,  $k$  – число информационных символов в комбинациях.

Неразделимые коды не имеют четкого деления кодовой комбинации на информационные и проверочные символы.

Разделимые блочные коды делятся на систематические и несистематические. Несистематические коды строятся таким образом, что проверочные символы определяются как сумма подблоков длины  $l$ , на которые разделяется блок информационных символов. У систематических кодов проверочные символы определяются в результате проведения линейных операций над определенными информационными символами.

**1.2.10. Основные характеристики двоичных кодов.** Двоичные коды характеризуются весом кода  $w$ , кодовым расстоянием  $d$  и весовой характери-

стикой  $F(w)$ . *Весом кода*  $w$  называется количество единиц в кодовой комбинации. Например, для кодовой комбинации 101110 вес кода  $w = 5$ .

Число одноименных разрядов двух кодовых комбинаций, в которых значения символов не совпадают, есть *кодвое расстояние*  $d$  между этими комбинациями. Для определения кодвоего расстояния необходимо сложить эти комбинации по модулю 2. Например, для кодовых комбинаций 10101 и 00110  $d=3$ , так как  $10101 \oplus 00110 = 10011$  ( $w=3$ ). Таким образом, кодвое расстояние определенного кода – это минимальное число элементов, которыми любая кодовая комбинация отличается от другой (по всем парам кодовых слов). Например, для кода, состоящего из комбинаций 1100, 1000, 1011, 1101,  $d_{min}=1$ , так как  $1100 \oplus 1101=0001$  ( $w=1$ ).

*Весовая характеристика кода*  $F(w)$  – число кодовых комбинаций определенного веса  $w$ . Например, для кода, представленного комбинациями 00001 ( $w = 1$ ), 11010 ( $w = 3$ ), 10110 ( $w = 3$ ), 11110 ( $w = 4$ ), имеем  $F(1) = 1$ ,  $F(3) = 2$ ,  $F(4) = 1$ , т.е. код состоит из одного кодового слова веса 1, двух слов веса 3 и одного слова веса 4.

Корректирующие коды имеют и некоторые дополнительные характеристики.

*Абсолютная избыточность кода* определяется числом проверочных символов ( $r$ ), т.е. количеством разрядов, отводимых для коррекции ошибок.

*Относительная избыточность кода* ( $R$ ) есть отношение числа проверочных символов к длине кода:  $R = r/n$ . В общем случае относительную избыточность рассчитывают по формуле  $R = 1 - \log_2 N_p / \log_2 N$ , где  $N_p$  – число кодовых комбинаций, используемых для передачи сообщений (рабочая мощность кода);  $N$  – полное число кодовых комбинаций (мощность кода).

### 1.3. Простые двоичные коды

Эти коды относятся к непомехозащищенным кодам. *Непомехозащищенным кодом* называется код, в котором искажение одного разряда кодовой комбинации не может быть обнаружено. Рассмотрим примеры двоичных непомехозащищенных кодов.

**1.3.1. Двоичный код на все сочетания.** Кодовые комбинации этого кода соответствуют записи натурального ряда чисел в двоичной системе счисления. Вес разряда кода определяется из выражения

$$q_i = 2^{i-1}, \quad (1.5)$$

где  $i = 1, 2, 3, \dots, n$ .

Общее число комбинаций

$$N = 2^n. \quad (1.6)$$

**1.3.2. Единично-десятичный код.** Каждый разряд десятичного числа записывается в виде соответствующего числа единиц (табл. 1.3). При этом разря-

ды разделяются интервалами. Например,  $2\ 4 \rightarrow 11\ 1111$ . Этот код неравномерный. Для преобразования в равномерный необходимо в каждом разряде слева дописать столько нулей, чтобы общее число символов в каждом десятичном разряде было равно 9. Например,  $2\ 4 \rightarrow 000000011\ 000001111$ .

**1.3.3. Двоично-десятичный код.** Каждый разряд десятичного числа записывается в виде комбинации двоичного кода.

В табл.1.3 представлены двоично-десятичные коды с весовыми коэффициентами: 8-4-2-1; 2-4-2-1; 4-2-2-1; 5-1-2-1.

Число 576 различными двоично-десятичными кодами будет записано следующим образом:

в коде 8-4-2-1	$576 \rightarrow 010101110110$ ;
в коде 2-4-2-1	$576 \rightarrow 101111011100$ ;
в коде 4-2-2-1	$576 \rightarrow 100111011010$ ;
в коде 5-1-2-1	$576 \rightarrow 100010101001$ ;

Коды с весовыми коэффициентами 2-4-2-1 называются самодополняющимися, так как инвертированный код, полученный заменой 0 на 1 и 1 на 0 в каждом разряде, всегда дополняет основной до числа 9 (1111). Например, если инвертировать комбинацию 0100 (цифра 4 в коде 2-4-2-1), то получится комбинация 1011, соответствующая цифре 5. При этом сложение прямой и инвертированной комбинации 0100 и 1011 дает в сумме комбинацию 1111, что соответствует цифре 9.

**1.3.4. Числоимпульсный.** Иногда его называют единичным (или унитарным) кодом. Кодовые комбинации отличаются друг от друга числом единиц. Примеры для 12-разрядного кода даны в табл. 1.3 (столбец 8б). Очевидно, что число кодовых комбинаций в этом коде равно разрядности, т.е.  $N = n$ .

**1.3.5. Код Джонсона.** Этот код применяется в устройствах, преобразующих линейные и угловые перемещения в кодовые комбинации. Записи цифр от 0 до 9 приведены в табл. 1.3 (столбец 7). Таким образом, число 137 в коде Джонсона будет представлено в виде 00001 00111 11100.

**1.3.6. Код Грея.** Этот код, который иногда называют рефлексным (отраженным), применяют для преобразования линейных и угловых перемещений в кодовые комбинации. Если при таком преобразовании используется обычный двоичный код, то некоторые расположенные рядом кодовые комбинации различаются в нескольких разрядах. Например, комбинации 0111 (цифра 7) и 1000 (цифра 8) различаются во всех разрядах. При считывании кода с кодового диска может возникнуть большая ошибка от неоднозначности считывания, обусловленная неточностью изготовления кодового диска или неточностью установки считывающих элементов. Допустим, что третий считывающий элемент установлен с отставанием, тогда при считывании цифры 8 получим кодовую комбинацию 1100, что соответствует цифре 12, а следовательно, ошибка будет равна 50 %.

Таблица 1.3

Запись кодовых комбинаций десятичных чисел от 0 до 15 различными кодами

Десятичный	8-4-2-1 на все сочетания	2-4-2-1 (Айкена)	4-2-2-1	5-1-2-1	Код Грея 15-7-3-1	Джонсона	Единично- десятичный неравномерный
1	2	3	4	5	6	7	8
0	0000	0000	0000	0000	0000	00000	25 → 11 11111
1	0001	0001	0001	0001	0001	00001	14 → 1 1111
2	0010	0010	0010	0010	0011	00011	Единично- десятичный равномерный
3	0011	0011	0101	0011	0010	00111	
4	0100	0100	0110	0111	0110	01111	
5	0101	1011	1001	1000	0111	11111	
6	0110	1100	1010	1001	0101	11110	8a
7	0111	1101	1101	1010	0100	11100	25 → 0000000110000011111
8	1000	1110	1110	1011	1100	11000	14 → 0000000010000001111
9	1001	1111	1111	1111	1101	10000	Унитарный 12 - разрядный
10	1010	10000	10000	10000	1111	1000000	
11	1011	10001	10001	10001	1110	100001	8б
12	1100	10010	10010	10010	1010	100011	
13	1101	10011	10101	10011	1011	100111	12 → 1111111111111
14	1110	10100	10110	10111	1001	101111	11 → 0111111111111
15	1111	11011	11001	11000	1000	111111	8 → 0000111111111

Построение кода Грея при отображении десятичных чисел от 0 до 15 четырехразрядным двоичным кодом поясняется табл. 1.3. Столбец старшего разряда делят пополам, в верхнюю половину вписывают нули, в нижнюю – единицы. Затем столбец следующего разряда делят на четыре равные части, которые заполняются единицами и нулями зеркально (с отражением) относительно линии разряда колонки старшего разряда. Аналогичная процедура выполняется в столбцах младших разрядов – единицы и нули заносятся зеркально относительно линий раздела колонки предыдущего разряда. В результате этих простых операций получили двоичный код, в котором соседние комбинации отличаются значением только в одном разряде. Например, те же цифры 7 и 8 в коде Грея запишутся как 0100 и 1100. Допустим, что 1-й считывающий элемент установлен с опережением, тогда вместо комбинации 1100 (цифра 8) получим комбинацию 1101 (цифра 9). Таким образом, ошибка в коде Грея не превосходит цены младшего разряда.

Код Грея, как и другие отраженные коды, относится к системам счисления с неестественным распределением весов разрядов, что затрудняет обработку информации, представленной этими кодами, в ЭВМ и дешифраторах. В силу этого отраженные коды перед обработкой преобразуются в простой двоичный код.

Вес разрядов кода Грея определяется выражением

$$q_i = 2^i - 1, \quad (1.7)$$

где  $i = 1, 2, 3, \dots, n$ .

То есть начиная с младшего разряда веса разрядов запишутся следующим образом: 1, 3, 7, 15, 31,.... Чтобы прочесть число в коде Грея, под каждым разрядом записывают его десятичный эквивалент, старший значащий разряд берется со знаком плюс, перед остальными значащими разрядами знаки чередуются. Например, перевод комбинации кода Грея 101111 и 010011 в десятичный код производится следующим образом:

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 1 & 1 \\ 63 & 31 & 15 & 7 & 3 & 1 \\ & & & & & = 63 - 15 + 7 - 3 + 1 = 53; \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 63 & 31 & 15 & 7 & 3 & 1 \\ & & & & & = 31 - 3 + 1 = 29. \end{array}$$

Код Грея относится к неарифметическим кодам. Поэтому перед обработкой информации производят преобразование в двоичный код.

Существует несколько алгоритмов перевода кода Грея в двоичный код и обратного преобразования. В общем виде число в двоичном коде можно записать как  $a_n a_{n-1} \dots a_i \dots a_1$ , а в коде Грея  $b_n b_{n-1} \dots b_i \dots b_1$ . Преобразование из кода Грея в двоичный код можно осуществлять по следующему правилу: цифра старшего разряда записывается без изменений, т.е.  $a_n = b_n$ ; значение каждого последующего разряда двоичного числа находят путем сложения по модулю

2 этого же разряда в коде Грея с предыдущими, т.е.  $a_{n-1} = b_n \oplus b_{n-1}$ . В общем случае можно записать:

$$a_i = \sum_{j=i}^n b_j \pmod{2}. \quad (1.8)$$

В качестве примера рассмотрим преобразование кодовой комбинации 101111, записанной в коде Грея, в двоичный код:

$$\begin{aligned} 101111 &= b_6 b_5 b_4 b_3 b_2 b_1 \rightarrow a_6 a_5 a_4 a_3 a_2 a_1 = b_6 (b_6 \oplus b_5) (b_6 \oplus b_5 \oplus b_4) \times \\ &\times (b_6 \oplus b_5 \oplus b_4 \oplus b_3) (b_6 \oplus b_5 \oplus b_4 \oplus b_3 \oplus b_2) (b_6 \oplus b_5 \oplus b_4 \oplus b_3 \oplus b_2 \oplus b_1) = \\ &= 1(1 \oplus 0)(1 \oplus 0 \oplus 1)(1 \oplus 0 \oplus 1 \oplus 1)(1 \oplus 0 \oplus 1 \oplus 1 \oplus 1)(1 \oplus 0 \oplus 1 \oplus 1 \oplus 1) = 110101 \end{aligned}$$

Произведем проверку правильности преобразования, для чего воспользуемся правилами чтения чисел, записанных в коде Грея и двоичном коде на все сочетания.

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 1 & 1 & \rightarrow & 1 & 1 & 0 & 1 & 0 & 1 \\ 63 & 31 & 15 & 7 & 3 & 1 & & 32 & 16 & 8 & 4 & 2 & 1 \end{array}$$

$$\begin{array}{ccc} 63 - 15 + 7 - 3 + 1 & \rightarrow & 32 + 16 + 4 + 1 \\ & & 53 = 53 \end{array}$$

Левая часть равна правой, следовательно, преобразование произведено верно.

Обычный двоичный код преобразуется в код Грея путем суммирования по модулю 2 данной комбинации с такой же, но сдвинутой вправо на один разряд. Например, преобразование двоичных чисел 110011 и 111011 в код Грея производится следующим образом:

$$\begin{array}{r} \oplus \begin{array}{r} 110011 \\ 110011 \\ \hline 101010 \end{array} \qquad \oplus \begin{array}{r} 111011 \\ 111011 \\ \hline 100110 \end{array} \end{array}$$

При сложении младший разряд второго слагаемого отбрасывается. Произведем проверку правильности преобразования:

$$\begin{array}{ccc} 1 & 1 & 0 & 0 & 1 & 1 & \rightarrow & 1 & 0 & 1 & 0 & 1 & 0; & 32+16+2+1 & \rightarrow & 63-15+3; & 51=51. \\ 32 & 16 & 8 & 4 & 2 & 1 & & 63 & 31 & 15 & 7 & 3 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & \rightarrow & 1 & 0 & 0 & 1 & 1 & 0; & 32+16+8+2+1 & \rightarrow & 63-7+3; & 59=59. \\ 32 & 16 & 8 & 4 & 2 & 1 & & 63 & 31 & 15 & 7 & 3 & 1 \end{array}$$

Преобразование двоичного числа в код Грея можно осуществить и по такому признаку. Если в старшем, соседнем по отношению к преобразуемому

разряде двоичного числа стоит 0, то в данном разряде кода Грея сохраняется цифра, записанная в двоичном коде, если же 1, то цифра меняется на обратную. Например, при переводе комбинации 110011 предыдущего примера в младшем разряде кода Грея 1 изменится на 0; во втором сохранится 1, так как в третьем разряде двоичного числа записан 0. В третьем сохранится 0, так как в четвертом разряде двоичного кода 0. В четвертом 0 изменится на 1, в пятом – 1 на 0 из-за того, что в пятом и шестом разряде двоичного кода стоит 1. Шестой разряд останется без изменения, так как подразумевается, что левее шестого разряда двоичного числа стоит 0.

На основании рассмотренных выше примеров значение разряда в коде Грея можно получить из выражения

$$b_i = a_{i+1} \oplus a_i. \quad (1.9)$$

В качестве примера рассмотрим преобразование двоичного числа 1011001 в код Грея

$$\begin{aligned} 1011001 &= a_7 a_6 a_5 a_4 a_3 a_2 a_1 \rightarrow b_7 b_6 b_5 b_4 b_3 b_2 b_1 = a_7 (a_7 \oplus a_6) (a_6 \oplus a_5) \times \\ &\times (a_5 \oplus a_4) (a_4 \oplus a_3) (a_3 \oplus a_2) (a_2 \oplus a_1) = 1(1 \oplus 0)(0 \oplus 1)(1 \oplus 1)(1 \oplus 0) \times \\ &\times (0 \oplus 0)(0 \oplus 1) = 1110101. \end{aligned}$$

#### 1.4. Оптимальные коды

Оптимальные по длине коды относятся к неравномерным непомяхозащищенным кодам. Оптимальным кодом считается код, имеющий минимальную среднюю длину кодового слова

$$L = \sum_{i=1}^n \mu_i P(x_i), \quad (1.10)$$

где  $\mu_i$  – длина кодового слова, сопоставляемая сообщению  $x_i$ ;  $P(x_i)$  – вероятность появления этого сообщения.

Очевидно, что  $\mu_i$  и  $L$  зависят от того, каким образом осуществляются формирование кодовых слов и их сопоставление с сообщениями  $x_i$ . Наиболее вероятные сообщения кодируются кодом меньшей длины, а менее вероятные – кодом большей длины. Тогда, учитывая, что по каналу связи чаще будут передаваться кодовые комбинации меньшей длины, получаем экономию во времени при передаче последовательности сообщений.

В оптимальном коде энтропия на символ должна быть максимальной, а это возможно в том случае, когда вероятности появления единиц и нулей  $P(1)$  и  $P(0)$  приблизительно одинаковы. Рассмотрим алгоритмы составления опти-

мальных кодов, удовлетворяющих максимальной энтропии на символ, при допущении, что время передачи единицы и нуля одинаковы  $t(1) = t(0)$ .

**1.4.1. Код Шеннона.** Суть метода Шеннона применительно к двоичному кодированию состоит в следующем. Все сообщения выписываются в порядке убывания их вероятностей. Далее множество дискретных сообщений делится на две части таким образом, чтобы сумма вероятностей сообщений, включенных в первую часть, была приблизительно равна сумме вероятностей сообщений второй части. После этого первому слева (старшему) разряду кода каждого сообщения первой части присваивается значение, равное нулю, а старшему разряду кода каждого сообщения второй части присваивается значение, равное единице. На этом считается законченным кодирование первого сообщения  $x_1$ . Затем остальные сообщения  $x_2, x_3 \dots x_n$  также делятся на две по возможности равновероятные подгруппы; одной из них присваивается значение 0, другой 1. На этом заканчивается кодирование второго сообщения  $x_2$ . Так продолжается до тех пор, пока не будут закодированы все сообщения.

Пример для кодирования 9 сообщений кодом Шеннона приведен в табл. 1.4.

После пятой разбивки кодирование можно приостановить, так как нет двух одинаковых кодовых комбинаций.

Подсчитаем среднее число нулей и единиц и вероятности их появления.

Среднее число нулей

$$L(0) = 0,35 + 0,15 + 0,13 + 0,18 + 0,18 + 0,16 + 0,15 + 0,08 + 0,02 = 1,4.$$

Среднее число единиц

$$L(1) = 0,15 + 0,26 + 0,18 + 0,27 + 0,24 + 0,10 + 0,12 + 0,08 = 1,4.$$

Средняя длина кодового слова

$$L = 0,35 + 0,30 + 0,39 + 0,36 + 0,45 + 0,40 + 0,25 + 0,20 + 0,10 = 2,8.$$

Тогда

$$P(1) = L(1)/L = 1,4/2,8 = 0,5 \quad P(0) = L(0)/L = 1,4/2,8 = 0,5.$$

Таким образом, получим код с максимальной энтропией на символ, но более короткие комбинации являются началом более длинных, что требует передачи разделительных пауз между кодовыми сообщениями, а следовательно приводит к снижению эффективности. От этого недостатка свободен метод Шеннона-Фано.

**1.4.2. Код Шеннона-Фано.** Для построения этого кода все сообщения  $x_i$  выписываются в порядке убывания их вероятностей (табл. 1.5).

Таблица 1.4

## Кодирование сообщений кодом Шеннона

Сообщения $x_i$	Вероятность появления со- общений $P(x_i)$	Разбиение сообщений на подмножества					$\mu_i$
		$x_1, x_2 \rightarrow 0$ $x_3, \dots, x_9 \rightarrow 1$	$x_2, x_3, x_8 \rightarrow 1$ $x_4, \dots, x_7, x_9 \rightarrow 0$	$x_3, x_6, x_8 \rightarrow 0$ $x_4, x_5, x_7, x_9 \rightarrow 1$	$x_4, x_7, x_8 \rightarrow 0$ $x_5, x_6, x_9 \rightarrow 1$	$x_5, x_7 \rightarrow 0$ $x_6, x_8, x_9 \rightarrow 1$	
		I	II	III	IV	V	
$x_1$	0,35	0					1
$x_2$	0,15	0	1				2
$x_3$	0,13	1	1	0			3
$x_4$	0,09	1	0	1	0		4
$x_5$	0,09	1	0	1	1	0	5
$x_6$	0,08	1	0	0	1	1	5
$x_7$	0,05	1	0	1	0	0	5
$x_8$	0,04	1	1	0	0	1	5
$x_9$	0,02	1	0	1	1	1	5

## Построение кода Шеннона-Фано

$x_i$	$P(x_i)$	Разбиение сообщений на подгруппы					Код	$\mu_i$	$L_{xi}$
$x_1$	0,35	1	1				11	2	0,70
$x_2$	0,15	1	0				10	2	0,30
$x_3$	0,13	0	1	1			011	3	0,39
$x_4$	0,09	0	1	0			010	3	0,27
$x_5$	0,09	0	0	1	1		0011	4	0,36
$x_6$	0,08	0	0	1	0		0010	4	0,32
$x_7$	0,05	0	0	0	1		0001	4	0,20
$x_8$	0,04	0	0	0	0	1	00001	5	0,20
$x_9$	0,02	0	0	0	0	0	00000	5	0,10

Записанные так сообщения затем разбиваются на две по возможности равновероятные подгруппы. Всем сообщениям первой подгруппы присваивают цифру 1 в качестве первого кодового символа, а сообщениям второй подгруппы – цифру 0. Аналогичное деление на подгруппы продолжается до тех пор, пока в каждую подгруппу не попадает по одному сообщению.

Найденный код весьма близок к оптимальному. В самом деле, энтропия сообщений

$$\begin{aligned}
 H(X) = -\sum_{i=1}^9 P(x_i) \log P(x_i) = & -(0,35 \log 0,35 + 0,15 \log 0,15 + 0,13 \log 0,13 + \\
 & + 0,09 \log 0,09 + 0,09 \log 0,09 + 0,08 \log 0,08 + 0,05 \log 0,05 + 0,04 \log 0,04 + \\
 & + 0,02 \log 0,02) \cong 2,75 \frac{\text{бит}}{\text{сообщение}}.
 \end{aligned}$$

Средняя длина кодового слова

$$L = \sum_{i=1}^9 L_{X_i} = 0,70 + 0,30 + 0,39 + 0,27 + 0,36 + 0,32 + 0,20 + 0,20 + 0,10 = 2,84.$$

Среднее число нулей

$$L(0) = 0,15 + 0,13 + 0,18 + 0,18 + 0,24 + 0,15 + 0,16 + 0,10 = 1,29.$$

Среднее число единиц

$$L(1) = 0,70 + 0,15 + 0,26 + 0,09 + 0,18 + 0,08 + 0,05 + 0,04 = 1,55.$$

Вероятность появления нулей

$$P(0) = \frac{L(0)}{L} = \frac{1,29}{2,84} = 0,455.$$

Вероятность появления единиц

$$P(1) = \frac{L(1)}{L} = \frac{1,55}{2,84} = 0,545.$$

Таким образом, получен код, близкий к оптимальному.

**1.4.3. Код Хаффмана.** Для получения кода Хаффмана все сообщения выписывают в порядке убывания вероятностей. Две наименьшие вероятности объединяют скобкой (табл. 1.6) и одной из них присваивают символ 1, а другой – 0. Затем эти вероятности складывают, результат записывают в промежутке между ближайшими вероятностями. Процесс объединения двух сообщений с наименьшими вероятностями продолжают до тех пор, пока суммарная вероятность двух оставшихся сообщений не станет равной единице. Код для каждого сообщения строится при записи двоичного числа справа налево путем обхода по линиям вверх направо, начиная с вероятности сообщения, для которого строится код.

Таблица 1.6

Построение кода Хаффмана

$x_i$	$P(x_i)$	Объединение сообщений						Код	
$x_1$	0,35	0,35	0,35	0,35	0,35	0,35	0,37	0,63 <sup>1</sup>	11
$x_2$	0,15	0,15	0,15	0,17	0,20	0,28	0,35 <sup>1</sup>		0,37 <sup>0</sup>
$x_3$	0,13	0,13	0,13	0,15	0,17	0,20 <sup>1</sup>	0,28 <sup>0</sup>		100
$x_4$	0,09	0,09	0,11	0,13	0,15 <sup>1</sup>	0,17 <sup>0</sup>			010
$x_5$	0,09	0,09	0,09	0,11 <sup>1</sup>	0,13 <sup>0</sup>				001
$x_6$	0,08	0,08	0,09 <sup>1</sup>	0,09 <sup>0</sup>					000
$x_7$	0,05	0,06 <sup>1</sup>	0,08 <sup>0</sup>						0110
$x_8$	0,04	0,05 <sup>1</sup>							01111
$x_9$	0,02								01110

Средняя длина кодового слова (см. табл. 1.6)  $L=2,82$ , что несколько меньше, чем в коде Шеннона-Фано ( $L=2,84$ ). Кроме того, методика Шеннона-Фано не всегда приводит к однозначному построению кода. Ведь при разби-

нии на подгруппы можно сделать большей по вероятности как верхнюю, так и нижнюю подгруппы.

От этого недостатка свободна методика Хаффмана. Она гарантирует однозначное построение кода с наименьшим для данного распределения вероятностей средним числом символов на букву.

## 2. КОРРЕКТИРУЮЩИЕ КОДЫ

### 2.1. Основные понятия

Помехоустойчивыми (корректирующими) называются коды, позволяющие обнаружить и исправить ошибки в кодовых комбинациях. Отсюда и деление кодов на две большие группы: 1) коды с обнаружением ошибок; 2) коды с обнаружением и исправлением ошибок.

Принципы обнаружения и исправления ошибок кодами проиллюстрируем с помощью геометрической модели трехразрядного двоичного кода (см. рис. 1.6). Если использовать все восемь кодовых комбинаций, записанных в вершинах куба, то образуется двоичный код на все сочетания. Как было показано выше, такой код является непомяоустойчивым. Если же уменьшить число используемых комбинаций с восьми до четырех, то появится возможность обнаружения одиночных ошибок. Для этого выберем только такие комбинации, которые отстоят друг от друга на расстояние  $d = 2$ , например, 000, 110, 011 и 101. Остальные кодовые комбинации не используются. Если будет принята комбинация 100, то очевидно, что при ее приеме произошла одиночная ошибка. Представленные комбинации построены по определенному правилу, а именно содержат четное число единиц, а принятая комбинация 100 – нечетное. Можно утверждать, что комбинация 100 образовалась при искажении разряда одной из разрешенных комбинаций, но определить, какая именно комбинация искажена, невозможно. Поэтому такие или подобные им коды называют кодами с обнаружением ошибок. Таким образом, в помехозащищенных кодах есть комбинации разрешенные, составленные по определенному правилу, и запрещенные, не соответствующие этому правилу. В общем случае при необходимости обнаруживать ошибки кратности до  $m$  включительно минимальное кодовое (хеммингово) расстояние между разрешенными кодовыми комбинациями должно быть по крайней мере на единицу больше  $m$ , т.е.

$$d_{\min} \geq m + 1. \quad (2.1)$$

Действительно, в этом случае ошибка, кратность которой не превышает  $m$ , не в состоянии перевести одну разрешенную кодовую комбинацию в другую.

Для исправления одиночной ошибки с каждой разрешенной кодовой комбинацией необходимо сопоставить подмножество запрещенных кодовых комбинаций. Чтобы эти подмножества не пересекались, хеммингово расстояние

между разрешенными кодовыми комбинациями должно быть не менее трех. Примем за разрешенные комбинации 000 и 111 (см. рис. 1.6). В результате возникновения единичной ошибки образуются подмножества:

Разрешенные комбинации  $\left\{ \begin{array}{l} 000 \rightarrow 001, 010, 100 \\ 111 \rightarrow 110, 101, 011 \end{array} \right\}$  запрещенные комбинации.

В общем случае для обеспечения возможности исправления всех ошибок кратности до  $S$  включительно каждая из ошибок должна приводить к запрещенной комбинации, относящейся к подмножеству исходной разрешенной кодовой комбинации.

Подмножество каждой из разрешенных  $n$ -разрядных комбинаций  $A_i$  (рис. 2.1) складывается из запрещенных комбинаций, являющихся следствием воздействия:

1) единичных ошибок (они располагаются на сфере радиусом  $d = 1$ , и их число равно  $C_n^1$ );

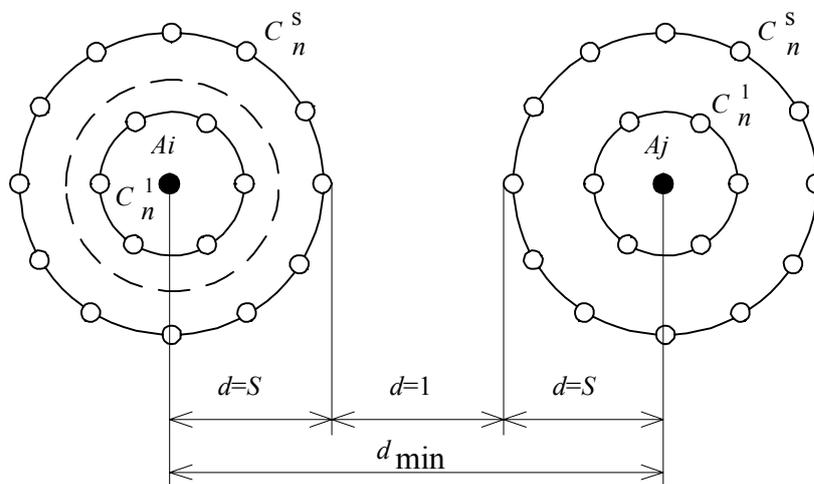


Рис. 2.1. Минимальное кодовое расстояние для исправления ошибок кратности  $S$

2) двойных ошибок (они располагаются на сфере радиусом  $d = 2$ , и их число равно  $C_n^2$ ) и т.д.

Внешняя среда подмножества имеет радиус  $d = S$  и содержит  $C_n^S$  запрещенных кодовых комбинаций.

Поскольку указанные подмножества не должны пересекаться, минимальное хеммингово расстояние между разрешенными комбинациями должно удовлетворять соотношению

$$d_{\min} \geq 2S + 1. \quad (2.2)$$

Нетрудно убедиться в том (рис. 2.2), что для исправления всех ошибок кратности  $S$  и одновременного обнаружения всех ошибок кратности  $m$  ( $m \geq S$ ) минимальное хеммингово расстояние нужно выбирать из условия

$$d_{\min} \geq m + S + 1. \quad (2.3)$$

Вопрос о минимально необходимой избыточности, при которой код обладает нужными корректирующими свойствами, является одним из важнейших в теории кодирования. Для некоторых частных случаев Хемминг указал простые соотношения, позволяющие определить необходимое число проверочных символов:

$$r_{d=3} \geq E \log(n + 1). \quad (2.4)$$

$$r_{d=3} \geq E \log((k + 1) + E \log(k + 1)). \quad (2.5)$$

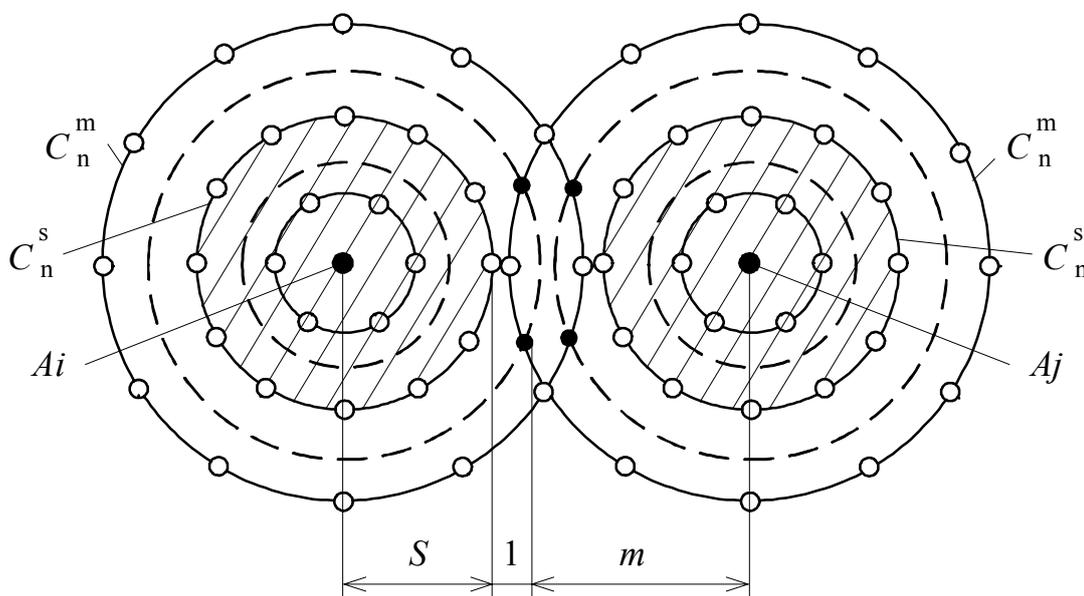


Рис. 2.2. Минимальное кодовое расстояние для одновременного исправления ошибок кратности  $S$  и обнаружения ошибок кратности  $m$

В реальных каналах связи длительность импульсов помехи часто превышает длительность символа. При этом одновременно искажаются несколько расположенных рядом символов комбинации. Ошибки такого рода получили название пачек ошибок или пакетов ошибок. Длиной пакета ошибок  $b$  называется число следующих друг за другом символов, левее и правее которых в кодовой комбинации искаженных символов не содержится. Если, например, кодовая комбинация 10101100111011 в результате действия помех трансформировалась в комбинацию 10101010101011, то длина пачки ошибок  $b$  составляет пять символов.

## 2.2. Коды с обнаружением ошибок

Особенностью этих кодов является то, что кодовые комбинации, входящие в их состав, отличаются друг от друга не менее чем на  $d=2$ .

Коды с обнаружением ошибок условно можно разбить на две группы:  
1) коды, построенные путем уменьшения числа используемых комбинаций;  
2) коды, в которых используются все комбинации, но к каждой из них по определенному правилу добавляются контрольные  $r$ -символы.

Рассмотрим сначала некоторые примеры кодов первой группы.

**2.2.1. Код с постоянным весом (код на одно сочетание).** Общее число кодовых комбинаций в данном коде

$$N = C_n^m = \frac{n!}{m!(n-m)!}, \quad (2.6)$$

где  $m$  – число единиц в слове длиной  $n$ .

В табл. 2.1 представлен код  $C_4^2$ . Правильность принятых кодовых комбинаций определяется путем подсчета количества единиц, и если их число отличается от  $m$ , то в передаче произошла ошибка. Необнаруженная ошибка имеет место, если произошло искажение типа “смещения”, т. е. когда единица переходит в нуль, а нуль – в единицу.

**2.2.2. Распределительный код с  $C_n^1$ .** Это также разновидность кода с постоянным весом, равным единице. Число кодовых комбинаций в данном коде

$$N = C_n^1 = n. \quad (2.7)$$

Кодовая комбинация при  $n=6$  представлена в табл. 2.1 (столбец 3). Сложение по модулю 2 двух комбинаций показывает, что они отличаются друг от друга на кодовое расстояние  $d=2$ . В системах телемеханики этот код нашел широкое применение из-за простой реализации.

Таблица 2.1

Код с постоянным числом единиц

Номер кодовой комбинации	Код $C_4^2$	Код $C_6^1$
1	0011	000001
2	0119	000010
3	1100	000100
4	1001	001000
5	1010	010000
6	0101	100000

Рассмотрим теперь коды второй группы.

**2.2.3. Код с проверкой на четность.** Код с проверкой на четность образуется путем добавления к передаваемой комбинации одного контрольного символа (0 или 1), так чтобы общее количество единиц в передаваемой комбинации было четным. Примеры представления кодовых комбинаций в данном коде приведены в табл. 2.2.

Таблица 2.2

Код с проверкой на чётность

Информационные символы $k$	Контрольные символы $r$	Код с проверкой на чётность $n=k+r$
01101	1	011011
01111	0	011110

Такой код состоит из  $N = 2^k$  комбинаций и имеет минимальное кодовое расстояние  $d_{\min} = 2$ . Коэффициент избыточности кода с проверкой на четность зависит от числа информационных символов:

$$K_{изб} = 1 - \frac{k}{k+1} = 1 - \frac{k}{n}. \quad (2.8)$$

Обнаружение ошибок на приемной стороне осуществляется подсчетом количества единиц в принятой комбинации, и если оно четное, считается что искажений нет. Тогда контрольный символ отбрасывается и исходная  $k$ -рядная комбинация выдается получателю информации. В противном случае кодовая комбинация бракуется.

Данный код может обнаружить любое нечетное число искажений.

Рассмотренный код является простейшим помехоустойчивым кодом, однако, принцип проверки на четность используется во многих достаточно сложных помехоустойчивых кодах.

**2.2.4. Код с проверкой на нечетность.** Особенностью кода является то, что каждая комбинация содержит нечетное число единиц (табл. 2.3). К проверке этого факта и сводится обнаружение ошибок в кодовых комбинациях. Другие основные характеристики кода такие же, как и у кода с одной проверкой на четность.

Таблица 2.3

Код с проверкой на нечётность

Информационные символы $k$	Контрольные символы $r$	Полная кодовая комбинация $n=k+r$
10101	0	101010
11101	1	111011

**2.2.5. Код с двумя проверками на четность.** Данный код является разновидностью кода с проверкой на четность и образуется путем добавления к передаваемой комбинации двух контрольных символов (табл. 2.4). Первый символ добавляет 0 или 1 так, чтобы общее количество единиц в передаваемой комбинации было четным, а второй символ добавляет 0 или 1 так, чтобы количество единиц в нечетных разрядах передаваемой комбинации было четным.

Обнаружение ошибок осуществляется подсчетом количества единиц в информационной части кодовой комбинации и первом контрольном разряде, а также в нечетных разрядах информационной части и втором контрольном символе, и если оно четное в первом и втором случае, то считается, что искажений нет. В противном случае принятая кодовая комбинация бракуется. Данный код позволяет обнаруживать все нечетные искажения и искажения в смежных разрядах, т.е. стоящих рядом.

Таблица 2.4

Код с двумя проверками на четность

Информационные символы $k$	Контрольные символы		Полная кодовая комбинация
	$r1$	$r2$	
101011	0	1	10101101
111101	1	0	11110110
100010	0	0	10001000
101010	1	1	10101011

**2.2.6. Код с повторением.** Этот код имеет две разновидности. В одной из них имеет место  $m$  – кратное повторение комбинаций простого кода  $a_1, a_2 \dots a_k$  :

$$\underbrace{a_1 a_2 \dots a_k}_1 \underbrace{a_1 a_2 \dots a_k}_2 \dots \underbrace{a_1 a_2 \dots a_k}_m$$

Например, при  $m = 3$  кодовая комбинация 1011 в коде с повторением комбинаций будет 1011 1011 1011.

Вторая разновидность кода с повторением характеризуется  $m$ –кратной передачей каждого разряда (код с повторением элементов кода):

$$\underbrace{a_1 a_1 \dots a_1}_{m \text{ раз}} \underbrace{a_2 a_2 \dots a_2}_{m \text{ раз}} \dots \underbrace{a_k a_k \dots a_k}_{m \text{ раз}}$$

Например, при  $m = 3$  кодовая комбинация 1011 в коде с  $m$ –кратной передачей каждого разряда будет 111 000 111 111.

Код с повторением имеет длину  $n = m \cdot k$ , число контрольных разрядов  $r = k(m - 1)$ . Избыточность этих кодов равна  $(m - 1)/m$ . Весьма высокая избыточность является недостатком кодов с повторением. Даже при двукратном повторении она составляет 0,5:

$$K_{изб} = 1 - \frac{k}{n} = 1 - \frac{k}{2k} = 0,5.$$

Код имеет минимальное кодовое расстояние  $d_{\min} = m$  и может использоваться как для обнаружения, так и для исправления ошибок. Для обнаружения ошибок применяют, как правило, код с четным  $d_{\min}$ , для исправления – с нечетным  $d_{\min}$ .

Правильность принятой информации определяется при проведении поэлементного сравнения информационных и контрольных символов, и при наличии хотя бы одного несовпадения вся принятая комбинация бракуется.

Код с повторением позволяет обнаруживать ошибки любой кратности за исключением случаев, когда искажается один информационный символ и все соответствующие ему контрольные, два информационных символа и соответствующие им контрольные и т.д.

При исправлении ошибок в комбинациях обычно применяется мажоритарный принцип исправления для каждого информационного символа, т.е. за истинное значение информационного символа принимается то, которое большее число раз встречается в этом информационном и соответствующих ему контрольных символах. При трехкратном повторении мажоритарный принцип реализуется как решение по двум символам из трех, при пятикратном – как решение по трем из пяти и т.д.

При увеличении числа повторений увеличивается минимальное кодовое расстояние, соответственно улучшаются корректирующие свойства кода, но значительно увеличивается и избыточность. Поэтому кратность повторений больше трех практически не используется.

В условиях коррелированных ошибок обычно применяют первую разновидность кода с повторением (код с повторением комбинаций), имеющую в этом случае более высокую помехоустойчивость. Это обусловлено тем, что входящие в одну проверку на четность разряды достаточно далеко отстоят друг от друга и с малой вероятностью поражаются одним пакетом ошибок.

**2.2.7. Код с числом единиц, кратным трем.** Этот код образуется добавлением к  $k$  информационным символам двух дополнительных контрольных символов ( $r = 2$ ), которые должны иметь такие значения, чтобы сумма единиц, посылаемых в линию кодовых комбинаций, была кратной трем. Примеры комбинаций такого кода представлены в табл. 2.5.

Таблица 2.5

Код с числом единиц, кратным трем

Информационные символы $k$	Контрольные символы		Полная кодовая комбинация
	$r1$	$r2$	
001000	1	1	00100011
011000	1	0	01100010
011001	0	0	01100100

Он позволяет обнаружить все одиночные ошибки и любое четное количество ошибок одного типа (например, только переход 0 в 1) не обнаруживаются двойные ошибки разных типов (смещения) и ошибки одного типа, кратные трем. На приемной стороне полученную комбинацию проверяют на кратность трем. При наличии такой кратности считают, что ошибок не было, два контрольных знака отбрасывают и записывают исходную комбинацию. Данный код обладает дополнительной возможностью обнаруживать ошибки: если первый контрольный символ равен нулю, то и второй тоже должен быть равен нулю.

**2.2.8. Инверсный код (код с повторением инверсии).** Это разновидность кода с двукратным повторением. При использовании данного кода комбинации с четным числом единиц повторяются в неизменном виде, а комбинации с нечетным числом единиц – в инвертированном.

Примеры представления кодовых комбинаций в инверсном коде приведены в табл. 2.6.

Таблица 2.6

Инверсный код

Информационные символы $k$	Контрольные символы $r$	Инверсный код $n=k+r$
111100	111100	111100111100
011100	100011	011100100011
110111	001000	110111001000
111010	111010	111010111010

Прием инверсного кода осуществляется в два этапа. На первом этапе суммируются единицы в первой половине кодовой комбинации. Если их количество окажется четным, то вторая половина кодовой комбинации принимается без инверсии, а если нечетным – то с инверсией. На втором этапе обе зарегистрированные комбинации поэлементно сравниваются, и при обнаружении хотя бы одного несовпадения комбинация бракуется. Это поэлементное сравнение эквивалентно суммированию по модулю 2. При отсутствии ошибок в обеих группах символов их сумма равна нулю.

Рассмотрим процесс обнаружения ошибок на следующем примере. Пусть передана последняя кодовая комбинация из табл. 2.6. Ниже показано суммирование для трех вариантов приема переданной комбинации:

$$\begin{array}{r}
 1) \oplus \begin{array}{r} 111010 \\ 111010 \\ \hline 000000 \end{array}
 \end{array}
 \quad
 \begin{array}{r}
 2) \oplus \begin{array}{r} 101010 \\ 000101 \\ \hline 101111 \end{array}
 \end{array}
 \quad
 \begin{array}{r}
 3) \oplus \begin{array}{r} 111010 \\ 101010 \\ \hline 010000 \end{array}
 \end{array}$$

В первом варианте принята комбинация 111010111010. В первой половине кодового слова (информационных символах) четное количество единиц, поэтому производится ее суммирование по модулю 2 с неинвертируемыми контрольными символами  $r$ , что в результате дает нулевую сумму, т.е. комбинация принята без искажений.

Во втором варианте принята комбинация 101010111010. Подсчитывая количество единиц в информационных символах и замечая, что оно нечетное, контрольные символы инвертируют и суммируют с информационными символами. Присутствие единиц в результате свидетельствует о наличии ошибки, а нуль в этой сумме показывает ее место.

В третьем варианте принята комбинация 111010101010. Поскольку в информационной последовательности четное количество единиц, при проверке контрольные символы суммируются с информационными без инверсии. В этом случае в итоге появляется одна единица. Ее место указывает номер искаженной позиции в принятой последовательности контрольных символов.

Таким образом, если при суммировании в результате среди единиц появляется один нуль – ошибка появилась в первой половине принятой кодовой комбинации (в информационных символах) и нуль указывает ее место. Если в результате среди нулей появляется одна единица – ошибка во второй половине кодовой комбинации (в контрольных символах) и ее место указывает единица. Если в результате суммирования имеется несколько единиц или нулей, это означает, что комбинация принята с несколькими искажениями.

Кодовое расстояние инверсного кода равно количеству разрядов исходного кода при  $k < 4$  и равно 4 при  $k \geq 4$ . Например, при  $d=4$  код может обнаруживать двойные ошибки и исправлять одиночные. Обычно этот код используется только для обнаружения ошибок. Он позволяет обнаруживать ошибки любой кратности за исключением таких, когда искажены 2 информационных символа и соответствующие им 2 контрольных, 4 информационных и соответствующие им 4 контрольных и т.д.

Коэффициент избыточности инверсного кода равен 0,5.

**2.2.9. Корреляционный код (код с удвоением числа элементов).** В рассматриваемом коде символы исходного кода кодируются повторно. Правило вторичного кодирования таково: если в исходном кодовом слове на какой-либо позиции стоит 0, в новом помехоустойчивом коде на эту позицию записывается пара символов 01, а если в исходном коде была 1, она записывается как 10.

Например, кодовое слово 1001 в корреляционном коде будет выглядеть следующим образом: 10010110. Корреляционный код будет всегда иметь вдвое больше элементов, чем исходный. Поэтому его коэффициент избыточности всегда равен 0,5:

$$K_{изб} = 1 - \frac{k}{n} = 1 - \frac{k}{2k} = 0,5.$$

На приеме ошибка обнаруживается в том случае, если в парных элементах содержатся одинаковые символы, т.е. 11 или 00 (вместо 10 и 01). При правильном приеме вторые (четные) элементы отбрасываются и остается первоначальная комбинация.

Код обладает сравнительно высокой помехоустойчивостью, поскольку ошибка не будет обнаружена только в том случае, если будут искажены два рядом стоящие элемента, соответствующие одному элементу исходного кода, т.е. 0 перейдет в 1, а 1 – в 0.

Наибольшая эффективность корреляционного кода проявляется при применении его на каналах, у которых вероятность искажения элементов (единиц и нулей) непрерывно меняется и в отдельные интервалы времени существенно различна.

**2.2.10. Код Бергера.** Контрольные символы в этом коде представляют разряды двоичного числа в прямом или инверсном виде количества единиц или нулей, содержащихся в исходной кодовой комбинации. Число контрольных символов определяется из выражения

$$r = E \log(k + 1), \quad (2.9)$$

где  $E$  – знак округления в большую сторону.

Примеры составления комбинаций в коде Бергера из обычного шестиразрядного двоичного кода представлены в табл. 2.7.

На приемной стороне подсчитывается число единиц (нулей) в информационной части и сравнивается с контрольной кодовой комбинацией (складывается по модулю 2).

При отсутствии ошибок в обеих комбинациях их сумма равна нулю. Ниже показана проверка для шести вариантов приема переданной комбинации из табл. 2.7. Искаженные символы отмечены точкой.

101011100	$100 \oplus 100 = 000$ - искажений нет.
100̇011100	$011 \oplus 100 = 111$ - искажение обнаружено.
11̇0011100	$100 \oplus 100 = 000$ - искажение не обнаружено.
11̇1111100	$110 \oplus 100 = 010$ - искажение обнаружено.
101011101̇	$100 \oplus 101 = 001$ - искажение обнаружено.
10101110̇10	$100 \oplus 010 = 110$ - искажение обнаружено.

## Код Бергера

Информационные символы	Контрольные символы				Полная кодовая комбинация $n=k+r$
	Количество единиц в двоичном коде		Количество нулей в двоичном коде		
	прямом	инверсном	прямом	инверсном	
101011	100				101011100
101011		011			101011011
101011			010		101011010
101011				101	101011101

Данный код обнаруживает все одиночные и большую часть многократных ошибок.

### 2.3. Коды с обнаружением и исправлением ошибок

Если кодовые комбинации составлены так, что отличаются друг от друга на кодовое расстояние  $d \geq 3$ , то они образуют корректирующий код, который позволяет по имеющейся в кодовой комбинации избыточности не только обнаруживать, но и исправлять ошибки. Большую группу кодов, исправляющих ошибки, составляют систематические коды. Рассмотрим общие принципы построения этих кодов.

**2.3.1. Систематические коды.** Систематическими кодами называются блочные  $(n, k)$  коды, у которых  $k$  (обычно первые) разрядов представляют собой двоичный избыточный код, а последующие  $r$ -контрольные разряды сформированные путем линейных комбинаций над информационными.

Основное свойство систематических кодов: сумма по модулю 2 двух и более разрешенных кодовых комбинаций также дает разрешенную кодовую комбинацию.

Правило формирования кода обычно выбирают так, чтобы при декодировании имелась возможность выполнить ряд проверок на четность для некоторых определенным образом выбранных подмножеств информационных и контрольных символов каждой кодовой комбинации. Анализируя результаты проверок, можно обнаружить или исправить ошибку ожидаемого вида.

Информацию о способе построения такого кода содержит проверочная матрица, которая составляется на базе образующей матрицы.

Образующая матрица  $M$  состоит из единичной матрицы размерностью  $k \cdot k$  и приписанной к ней справа матрицы дополнений размерностью  $k \cdot r$ :

$$M = \left\| \begin{array}{ccc|ccc} 1 & 0 & 0 \dots 0 & b_{11} & b_{12} & \dots & b_{1r} \\ 0 & 1 & 0 \dots 0 & b_{21} & b_{22} & \dots & b_{2r} \\ 0 & 0 & 1 \dots 0 & b_{31} & b_{32} & \dots & b_{3r} \\ \cdot & \cdot & \dots & \cdot & \cdot & \dots & \cdot \\ 0 & 0 & 0 \dots 1 & b_{k1} & b_{k2} & \dots & b_{kr} \end{array} \right\| . \quad (2.10)$$

Разрядность матрицы дополнений выбирается из выражения (2.4) или (2.5). Причем вес  $w$  (число ненулевых элементов) каждой строки матрицы дополнений должен быть не меньше чем  $d_{\min} - 1$ .

Проверочная матрица  $N$  строится из образующей матрицы следующим образом. Строками проверочной матрицы являются столбцы матрицы дополнений образующей матрицы. К полученной матрице дописывается справа единичная матрица размерностью  $r \times r$ . Таким образом, проверочная матрица размерностью  $r \times k$  имеет вид

$$N = \left\| \begin{array}{cccc|ccc} b_{11} & b_{21} & b_{31} & \dots & b_{k1} & 1 & 0 & 0 \dots 0 \\ b_{12} & b_{22} & b_{32} & \dots & b_{k2} & 0 & 1 & 0 \dots 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \dots \\ b_{1r} & b_{2r} & b_{3r} & \dots & b_{kr} & 0 & 0 & 0 \dots 1 \end{array} \right\| . \quad (2.11)$$

Единицы, стоящие в каждой строке, однозначно определяют, какие символы должны участвовать в определении значения контрольного разряда. Причем единицы в единичной матрице определяют номера контрольных разрядов.

**Пример 2.1.** Получить алгоритм кодирования в систематическом коде всех четырехразрядных кодовых комбинаций, позволяющего исправлять единичную ошибку. Таким образом, задано число информационных символов  $k = 4$  и кратность исправления  $S = 1$ . По выражению (2.5) определим число контрольных символов:

$$r \geq E \log((4 + 1) + E \log(4 + 1)) = 3 .$$

Минимальное кодовое расстояние определим из выражения (2.2)

$$d_{\min} \geq 2 \cdot 1 + 1 = 3 .$$

Строим образующую матрицу

$$M = \left\| \begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right\| .$$

Проверочная матрица будет иметь вид

$$N = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Обозначим символы, стоящие в каждой строке, через  $a_i$  ( $a_1 a_2 a_3 a_4 a_5 a_6 a_7$ ). Символы  $a_5$ ,  $a_6$  и  $a_7$  примем за контрольные, так как они будут входить только в одну из проверок.

Составим проверки для каждого контрольного символа. Из первой строки имеем

$$a_5 = a_2 \oplus a_3 \oplus a_4. \quad (2.12)$$

Из второй строки получим алгоритм для формирования контрольного символа  $a_6$ :

$$a_6 = a_1 \oplus a_2 \oplus a_4. \quad (2.13)$$

Аналогично из третьей строки получим алгоритм для формирования контрольного символа  $a_7$ :

$$a_7 = a_1 \oplus a_3 \oplus a_4. \quad (2.14)$$

Нетрудно убедиться, что все результаты проверок на четность по выражениям (2.12)–(2.14) дают нуль, что свидетельствует о правильности составления образующей и проверочной матриц.

**Пример 2.2.** На основании алгоритма, полученного в примере 2.1, закодировать кодовую комбинацию  $G(x) = 1101 = a_1 a_2 a_3 a_4$  в систематическом коде, позволяющем исправлять одиночную ошибку.

По выражениям (2.12), (2.13) и (2.14) найдем значения для контрольных символов  $a_5$ ,  $a_6$  и  $a_7$ .

$$\begin{aligned} a_5 &= 1 \oplus 0 \oplus 1 = 0, \\ a_6 &= 1 \oplus 1 \oplus 1 = 1, \\ a_7 &= 1 \oplus 0 \oplus 1 = 0. \end{aligned} \quad (2.15)$$

Таким образом, кодовая комбинация  $F(x)$  в систематическом коде будет иметь вид:

$$F(x) = 1101010. \quad (2.16)$$

На приемной стороне производятся проверки  $S_i$  принятой кодовой комбинации, которые составляются на основании выражений (2.12)–(2.14):

$$\begin{aligned} S_1 &= a_2 \oplus a_3 \oplus a_4 \oplus a_5, \\ S_2 &= a_1 \oplus a_2 \oplus a_4 \oplus a_6, \\ S_3 &= a_1 \oplus a_3 \oplus a_4 \oplus a_7. \end{aligned} \quad (2.17)$$

Если синдром (результат проверок на четность)  $S_1S_2S_3$  будет нулевого порядка, то искажений в принятой кодовой комбинации  $F'(x)$  нет. При наличии искажений синдром  $S_1S_2S_3$  указывает, какой был искажен символ. Рассмотрим всевозможные состояния  $S_1S_2S_3$ :

$S_1$	$S_2$	$S_3$	
0	0	0	– искажений нет,
1	0	0	– искажен символ $a_5$ ,
0	1	0	– искажен символ $a_6$ ,
0	0	1	– искажен символ $a_7$ ,
1	1	0	– искажен символ $a_2$ ,
0	1	1	– искажен символ $a_1$ ,
1	1	1	– искажен символ $a_4$ ,
1	0	1	– искажен символ $a_3$ .

(2.18)

**Пример 2.3.** Кодовая комбинация  $F(x) = 1101010$  (пример 2.2) при передаче была искажена и приняла вид  $F'(x) = 1111010 = a_1a_2a_3a_4a_5a_6a_7$ . Декодировать принятую кодовую комбинацию.

Произведем проверки согласно выражениям (2.17)

$$\begin{aligned} S_1 &= 1 \oplus 1 \oplus 1 \oplus 0 = 1, \\ S_2 &= 1 \oplus 1 \oplus 1 \oplus 1 = 0, \\ S_3 &= 1 \oplus 1 \oplus 1 \oplus 0 = 1. \end{aligned}$$

Полученный синдром  $S_1S_2S_3 = 101$  согласно (2.18) свидетельствует об искажении символа  $a_3$ . Заменяем этот символ на противоположный и получаем исправленную кодовую комбинацию  $F(x) = 1101010$ , а исходная кодовая комбинация имеет  $G(x) = 1101$ , что совпадает с кодовой комбинацией, подлежащей кодированию в примере 2.2.

**2.3.2. Код Хемминга.** Данный код относится к числу систематических кодов. По существу, это целая группа кодов, при  $d_{\min} = 3$  исправляющая все

одинокими или обнаруживающая двойные ошибки, а при  $d_{\min} = 4$  исправляющая одиночные и обнаруживающая двойные ошибки.

В качестве исходных берут двоичный код на все сочетания с числом информационных символов  $k$ , к которому добавляют контрольные символы  $r$ . Таким образом, общая длина закодированной комбинации  $n = k + r$ .

Рассмотрим последовательность кодирования и декодирования кода Хемминга.

*Кодирование. Определение числа контрольных символов.* При передаче по каналу с шумами может быть или искажен любой из  $n$  символов кода, или слово передано без искажений. Таким образом, может быть  $n + 1$  вариантов принятых сообщений. Используя контрольные символы, необходимо различить все  $n + 1$  вариантов. С помощью контрольных символов  $r$  можно описать  $2^r$  событий. Значит, должно быть выполнено условие

$$2^r \geq n + 1 = k + r + 1. \quad (2.19)$$

В табл. 2.8 представлена зависимость между  $k$  и  $r$ , полученная из этого неравенства.

Таблица 2.8

Число контрольных символов  $r$  в коде Хемминга в зависимости от числа информационных символов

$k$	1	2	3	4	5	6	7	8	9	10	11	12	13
$r$	2	3	3	3	4	4	4	4	4	4	4	5	5

Чаще всего заданными является число информационных символов, тогда число контрольных символов можно определить из выражения (2.5).

*Размещение контрольных символов.* К построению кодов Хемминга обычно привлекают производящие матрицы, а процедура проверок при обнаружении и исправлении ошибок проводится с помощью проверочных матриц.

Ниже приводится более простой алгоритм, получивший широкое распространение.

В принципе, место расположения контрольных символов не имеет значения: их можно приписывать и перед информационными символами, и после них, и чередуя информационные символы с контрольными. Для удобства обнаружения искаженного символа целесообразно размещать их на местах, кратных степени 2, т.е. на позициях 1, 2, 4, 8 и т.д. Информационные символы располагаются на оставшихся местах. Поэтому, например, для девятиэлементной закодированной комбинации можно записать

$$r_1, r_2, k_5, r_3, k_4, k_3, k_2, r_4, k_1, \quad (2.20)$$

где  $k_5$  – старший (пятый) разряд исходной кодовой комбинации двоичного кода, подлежащий кодированию;

$k_1$  – младший (первый) разряд.

*Определение состава контрольных символов.* Какой из символов должен стоять на контрольной позиции (1 или 0), выявляют с помощью проверки на четность. Для этого составляют колонку ряда натуральных чисел в двоичном коде, число строк в которой равно  $n$ , а рядом справа, сверху вниз проставляют символы комбинации кода Хемминга, записанные в такой последовательности (2.20):

$$\begin{array}{ll}
 0001 - r_1 & 0110 - k_3 \\
 0010 - r_2 & 0111 - k_2 \\
 0011 - k_5 & 1000 - r_4 \\
 0100 - r_3 & 1001 - k_1 \\
 0101 - k_4 &
 \end{array} \quad (2.21)$$

Затем составляются проверки по следующему принципу: первая проверка – коэффициенты с единицей в младшем разряде ( $r_1, k_5, k_4, k_2, k_1$ ); вторая – коэффициенты во втором разряде ( $r_2, k_5, k_3, k_2$ ); третья – коэффициенты с единицей в третьем разряде ( $r_3, k_4, k_3, k_2$ ); четвертая – коэффициенты в четвертом разряде ( $r_4, k_1$ ). Рассматривая проверки, видим, что каждый контрольный символ входит только в одну из проверок, а поэтому для определения состава контрольных символов суммируют информационные символы, входящие в каждую строку. Если сумма единиц в данной строке четная, то значение символа  $r$ , входящего в эту строку, равно нулю, если нечетная, то единице. Таким образом,

$$\begin{aligned}
 r_1 &= k_5 \oplus k_4 \oplus k_2 \oplus k_1, \\
 r_2 &= k_5 \oplus k_3 \oplus k_2, \\
 r_3 &= k_4 \oplus k_3 \oplus k_2, \\
 r_4 &= k_1.
 \end{aligned} \quad (2.22)$$

В случае кодирования более длинных кодовых комбинаций нужно лишь увеличить число разрядов двоичного кода в колонках (2.21).

*Декодирование.* Для проверки правильности принятой комбинации производят  $S_i$  проверок на четность:

$$\begin{aligned}
 S_1 &= r_1 \oplus k_5 \oplus k_4 \oplus k_2 \oplus k_1, \\
 S_2 &= r_2 \oplus k_5 \oplus k_3 \oplus k_2, \\
 S_3 &= r_3 \oplus k_4 \oplus k_3 \oplus k_2, \\
 S_4 &= r_4 \oplus k_1.
 \end{aligned} \quad (2.23)$$

Если комбинация принята без искажений, то сумма единиц по модулю 2 дает нуль. При искажении какого-либо символа суммирование при проверке дает единицу. По результату суммирования каждой из проверок (2.23) составляют двоичное число  $S_4, S_3, S_2, S_1$  (синдром), указывающее на место искажения. Например, первая и вторая проверки показали наличие искажения, а суммирования при третьей и четвертой проверках (2.23) дали нули. Записываем число  $S_4, S_3, S_2, S_1 = 0011$ , которое означает, что в третьем символе кодовой комбинации (2.20), включающей и контрольные символы (счет производится слева направо), возникло искажение, значит, этот символ нужно исправить на обратный ему. После этого контрольные символы, стоящие на заранее известных местах, отбрасываются.

Код Хемминга с  $d_{\min} = 4$  строится на базе кода Хемминга с  $d_{\min} = 3$  путем добавления дополнительного контрольного символа к закодированной комбинации, который позволяет производить проверку на четность всей комбинации. Поэтому контрольный символ должен быть равен единице, если число единиц в закодированной комбинации нечетное, и нулю, если число единиц четное, т.е. закодированная комбинация будет иметь вид

$$r_1, r_2, k_5, r_3, k_4, k_3, k_2, r_4, k_1, r_5, \quad (2.24)$$

где

$$r_5 = r_1 \oplus r_2 \oplus k_5 \oplus r_3 \oplus k_4 \oplus k_3 \oplus k_2 \oplus r_4 \oplus k_1. \quad (2.25)$$

При декодировании дополнительно к проверкам (2.23) производится проверка

$$S_{\Sigma} = r_1 \oplus r_2 \oplus k_5 \oplus r_3 \oplus k_4 \oplus k_3 \oplus k_2 \oplus r_4 \oplus k_1 \oplus r_5. \quad (2.26)$$

При этом возможны следующие варианты:

- 1) частные проверки (2.23)  $S_i = 0$  и общая (2.25)  $S_{\Sigma} = 0$  — ошибок нет;
- 2)  $S_i \neq 0$  и  $S_{\Sigma} = 0$  — двойная ошибка, принятая кодовая комбинация бракуется;
- 3)  $S_i \neq 0$  и  $S_{\Sigma} \neq 0$  — одиночная ошибка, синдром указывает номер в двоичном коде искаженного разряда, который корректируется;
- 4)  $S_i = 0$  и  $S_{\Sigma} \neq 0$  — искажен последний разряд общей проверки на четность, информационные символы поступают потребителю.

**Пример 2.4.** Закодировать в коде Хемминга с  $d = 4$  кодовую комбинацию  $G(X) = 10011$  т.е.  $k = 5$ .

**Решение.** Согласно табл. 2.8, число контрольных символов  $r_{d=3} = 4$ , размещаются они на позициях 1, 2, 4 и 8, и информационные — на позициях 3,

5, 6, 7, 9. Учитывая, что  $G(X)$  необходимо закодировать в коде Хемминга с  $d = 4$ , добавляют пятый контрольный разряд общей проверки на четность (2.25). Тогда последовательность в общем виде можно записать так:

$$\begin{array}{cccccccccc} r_1, & r_2, & k_5, & r_3, & k_4, & k_3, & k_2, & r_4, & k_1, & r_5 \\ ? & ? & 1 & ? & 0 & 0 & 1 & ? & 1 & ? \end{array} \quad (2.27)$$

Для определения контрольных символов  $r_1 - r_4$  подставим значения  $k_1 \div k_5$  в (2.22) и получим

$$\begin{aligned} r_1 &= 1 \oplus 0 \oplus 1 \oplus 1 = 1, \\ r_2 &= 1 \oplus 0 \oplus 1 = 0, \\ r_3 &= 0 \oplus 0 \oplus 1 = 1, \\ r_4 &= 1. \end{aligned}$$

Контрольный символ  $r_5$  определим из выражения (2.25):

$$r_5 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0.$$

Таким образом, в линию связи будет послан код

$$F(X) = 1011001110.$$

**Пример 2.5.** В приемник поступила кодовая комбинация  $F'(X) = 1010001110$  в коде Хемминга с  $d = 4$ . Декодировать ее; если имеются искажения, то обнаружить и при возможности исправить.

**Решение.** Произведем  $S_i$  проверки согласно (2.23) и  $S_\Sigma$  согласно (2.26), в результате получим:

$$\begin{aligned} S_1 &= 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0, \\ S_2 &= 0 \oplus 1 \oplus 0 \oplus 1 = 0, \\ S_3 &= 0 \oplus 0 \oplus 0 \oplus 1 = 1, \\ S_4 &= 1 \oplus 1 = 0, \\ S_\Sigma &= 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1. \end{aligned}$$

Таким образом, получим синдром  $S_4 S_3 S_2 S_1 = 0100$  и  $S_\Sigma = 1$ , что указывает на то, что искажен четвертый разряд кодовой комбинации  $F'(X)$ . После исправления получим  $F(X) = 1011001110$ , а следовательно, информационная последовательность будет иметь вид  $G(X) = 10011$ , что соответствует исходной кодовой комбинации примера 2.4.

**2.3.3. Циклические коды.** *Общие понятия и определения.* Циклические коды относятся к числу блочных систематических кодов, в которых каждая комбинация кодируется самостоятельно (в виде блока) таким образом, что информационные  $k$  и контрольные  $r$  символы всегда находятся на определенных местах.

Любой групповой код  $(n, k)$  может быть записан в виде матрицы, включающей  $k$  линейно независимых строк по  $n$  символов, и, наоборот, любая совокупность  $k$  линейно независимых  $n$ -разрядных кодовых комбинаций может рассматриваться как образующая матрица некоторого группового кода. Среди всего многообразия таких кодов можно выделить коды, у которых строки образующих матриц связаны дополнительным условием цикличности.

Все строки образующей матрицы такого кода могут быть получены циклическим сдвигом одной комбинации, называемой образующей для данного кода. Коды, удовлетворяющие этому условию, получили название циклических кодов. Сдвиг осуществляется справа налево, причем крайний левый символ каждый раз переносится в конец комбинации. Запишем, например, совокупность кодовых комбинаций, получающихся циклическим сдвигом комбинации 001011:

$$G = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

При описании циклических кодов  $n$ -разрядные кодовые комбинации представляются в виде многочленов фиктивной переменной  $x$  (см. подразд. 1.2.1). Тогда циклический сдвиг строки матрицы с единицей в старшем ( $n$ -м) разряде (слева) равносильно умножению соответствующего строке многочлена на  $x$  с одновременным вычитанием из результата многочлена  $X^n + 1 = X^n - 1$ , т.е. с приведением по модулю  $X^n + 1$ . Умножив, например, первую строку матрицы (001011), соответствующую многочлену  $G_0(X) = x^3 + x + 1$ , на  $x$ , получим вторую строку матрицы (010110), соответствующую многочлену  $X \cdot G_0(X)$ . Нетрудно убедиться, что кодовая комбинация, получающаяся при сложении этих двух комбинаций, также будет соответствовать результату умножения многочлена  $x^3 + x + 1$  на многочлен  $x + 1$ . Действительно,

$$\begin{aligned} 001011 \oplus 010110 &= 011101 = x^4 + x^3 + x^2 + 1, (x^3 + x + 1)(x + 1) = \\ &= x^4 + x^3 + x^2 + 1 = 011101. \end{aligned}$$

Отсюда ясно, что любая разрешенная кодовая комбинация циклического кода может быть получена в результате умножения образующего многочлена на некоторый другой многочлен с приведением результата по модулю  $x^n + 1$ . Иными словами, при соответствующем выборе образующего многочлена, любой многочлен циклического кода будет делиться на него без остатка.

Ни один многочлен, соответствующий запрещенной кодовой комбинации, на образующий многочлен без остатка не делится. Это свойство позволяет обнаружить ошибку. По виду остатка можно определить и вектор ошибки.

Умножение и деление многочленов весьма просто осуществляется на регистрах сдвига с обратными связями и сумматорах по модулю 2.

В основу циклического кодирования положено использование неприводимого многочлена  $P(X)$ , который применительно к циклическим кодам называется образующим, генераторным или производящим многочленом (полиномом).

Многочлен в поле двоичных чисел называется неприводимым, если он делится без остатка только на себя или на единицу. Неприводимые полиномы приведены в прил. 1.

*Методы построения циклического кода.* Существует несколько различных способов кодирования. Принципиально наиболее просто комбинации циклического кода можно получить, умножая многочлены  $G(X)$ , соответствующие комбинациям безызбыточного кода (информационным символам), на образующий многочлен кода  $P(X)$ . Такой способ легко реализуется, однако он имеет тот существенный недостаток, что получающиеся в результате умножения комбинации кода не содержат информационных символов в явном виде. После исправления ошибок такие комбинации для выделения информационных символов приходится делить на образующий многочлен кода. Ситуацию можно значительно упростить, если контрольные символы переписать в конце кода, т.е. после информационных символов. Для этой цели прибегают к следующему искусственному приему.

Умножаем кодовую комбинацию  $G(X)$ , которую мы хотим закодировать, на многочлен  $X^r$ , имеющий ту же степень, что и образующий многочлен  $P(X)$ . Делим произведение  $G(X)X^r$  на образующий полином  $P(X)$ :

$$\frac{G(X) \cdot X^r}{P(X)} = Q(X) + \frac{R(X)}{P(X)}, \quad (2.28)$$

где  $Q(X)$  – частное от деления;  $R(X)$  – остаток.

Умножая выражение (2.28) на  $P(X)$  и перенося  $R(X)$  в другую часть равенства, согласно правилам алгебры двоичного поля, т.е. без перемены знака на обратный, получаем

$$F(X) = Q(X) \cdot P(X) = G(X) \cdot X^r + R(X). \quad (2.29)$$

Таким образом, согласно равенству (2.29), циклический код можно образовать двумя способами:

1) умножением одной из комбинаций двоичного кода на все сочетания (комбинация  $Q(X)$  принадлежит к той же группе того же кода, что и заданная комбинация  $G(X)$ ) на образующий многочлен  $P(X)$ ;

2) умножением заданной комбинации  $G(X)$  на одночлен  $X^r$ , имеющий ту же степень, что и образующий многочлен  $P(X)$ , с добавлением к этому произведению остатка  $R(X)$ , полученного после деления произведения  $G(X) \cdot X^r$  на генераторный полином  $P(X)$ .

**Пример 2.6.** Закодировать кодовую комбинацию  $G(X) = 1111 = x^3 + x^2 + x + 1$  циклическим кодом.

**Решение.** Не останавливаясь на выборе генераторного полинома  $P(X)$ , о чем будет сказано подробно далее, возьмем из прил. 1 многочлен  $P(X) = x^3 + x + 1 = 1011$ . Умножая  $G(X) \cdot X^r$ , получаем

$$G(X) \cdot X^n = (x^3 + x^2 + x + 1)x^3 = x^6 + x^5 + x^4 + x^3 \rightarrow 1111000.$$

От умножения степень каждого члена повысилась, что равносильно приписыванию трех нулей к многочлену, выраженному в двоичной форме.

Разделив на  $G(X) \cdot X^r$  на  $P(X)$ , согласно (2.28) получим

$$\frac{x^6 + x^5 + x^4 + x^3}{x^3 + x + 1} = (x^3 + x^2 + 1) + \frac{x^2 + x + 1}{x^3 + x + 1},$$

или в двоичном эквиваленте

$$1111000/1011 = 1101 + 111/1011.$$

Таким образом, в результате деления получаем частное  $Q(X) = 1101$  той же степени, что и  $G(X) = 1111$ , и остаток  $R(X) = 111$ . В итоге комбинация двоичного кода, закодированная циклическим кодом, согласно (2.29) примет вид

$$F(X) = 1101 \cdot 1011 = 1111000 \oplus 111 = 1111111.$$

Действительно, умножение  $1101 \cdot 1011$  (первый способ) дает тот же результат, что и сложение  $1111000 \oplus 111$  (второй способ).

**Циклические коды, обнаруживающие одиночную ошибку ( $d = 2$ ).** Код, образованный генераторным полиномом  $P(X) = x + 1$ , обнаруживает любое нечетное число ошибок.

Закодируем сообщение  $G(X) = 1101$  с помощью многочлена  $P(X) = 11$ . Поступая по методике, рассмотренной выше, получим

$$F(X) = G(X) \cdot X^r + R(X) = 11010 + 1 = 11011,$$

т.е. на первых четырех позициях находятся разряды исходной комбинации  $G(X)$ , а на пятой – контрольный символ.

Сообщение 1101 является одной из 16 комбинаций четырехразрядного кода. Если требуется передать все эти сообщения в закодированном виде, то каждое из них следует кодировать так же, как и комбинацию  $G(X) = 1101$ . Однако проделывать дополнительно 15 расчетов (в общем случае  $2^r$  расчетов) нет необходимости. Это можно сделать проще, путем составления образующей матрицы. Образующая матрица составляется из единичной транспонированной и матрицы дополнений, составленной из остатков от деления единицы с нулями на образующий многочлен  $P(X)$ , выраженный в двоичном эквиваленте (см. подразд. 1.2.4.). Образующая матрица в данном случае имеет вид:

$$M = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{matrix}$$

Единичная транспонированная матрица

Матрица дополнений

Четыре кодовые комбинации, из которых состоит образующая матрица, являются первыми кодовыми комбинациями циклического кода. Пятая комбинация нулевая, а так как в четырехразрядном непомехозащищенном коде всего  $N = 2^4 = 16$  комбинаций, то остальные 11 ненулевых комбинаций находят суммированием по модулю 2 всевозможных комбинаций строк матрицы  $M$ :

- |                               |   |  |
|-------------------------------|---|--|
| 5. 0000,                      | 9. $a_2 \oplus a_3 = 01100$ ,             | 13. $a_2 \oplus a_3 \oplus a_4 = 11101$ ,            |
| 6. $a_1 \oplus a_2 = 00110$ , | 10. $a_2 \oplus a_4 = 10100$ ,            | 14. $a_1 \oplus a_3 \oplus a_4 = 11011$ ,            |
| 7. $a_1 \oplus a_3 = 01010$ , | 11. $a_3 \oplus a_4 = 11000$ ,            | 15. $a_1 \oplus a_2 \oplus a_4 = 10111$ ,            |
| 8. $a_1 \oplus a_4 = 10010$ , | 12. $a_1 \oplus a_2 \oplus a_3 = 01111$ , | 16. $a_1 \oplus a_2 \oplus a_3 \oplus a_4 = 11110$ . |

Рассмотрение полученных комбинаций показывает, что все они имеют четное число единиц. Таким образом, циклический код с обнаружением одиночной ошибки  $d = 2$  является кодом с проверкой на четность.

**Циклический код с  $d = 3$ .** Эти коды могут обнаруживать одиночные и двойные ошибки или обнаруживать и исправлять одиночные ошибки. Можно предложить следующий порядок кодирования кодовых комбинаций в циклическом коде с  $d = 3$ .

1) *выбор числа контрольных символов.* Выбор  $r$  производят, как и для кода Хемминга, с исправлением одиночной ошибки, по выражению (2.4) или (2.5);

2) *выбор образующего многочлена  $P(X)$ .* Степень образующего многочлена не может быть меньше числа контрольных символов  $r$ . Если в прил. 1 имеется ряд многочленов с данной степенью, то из них следует выбирать самый короткий. Однако число ненулевых членов многочлена  $P(X)$  не должно быть меньше кодового расстояния  $d$ ;

3) *нахождение элементов дополнительной матрицы.* Дополнительную матрицу составляют из остатков, полученных от деления единицы с нулями на образующий полином  $P(X)$ . Порядок получения остатков показан в подразд. 1.2.4. При этом должны соблюдаться следующие условия:

а) число остатков должно быть равно числу информационных символов  $k$ ;

б) для дополнительной матрицы пригодны лишь остатки с весом  $w$ , не меньшим числа обнаруживаемых ошибок  $t$ , т.е. в данном случае не меньшим  $2(w \geq 2)$ ;

в) количество нулей, приписываемых к единице при делении ее на многочлен  $P(X)$ , определяется из условий а и б;

г) число разрядов дополнительной матрицы равно числу контрольных символов  $r$ ;

4) *составление образующей матрицы.* Берут транспонированную единичную матрицу размерностью  $k \times k$  и справа приписывают к ней дополнительную матрицу размерностью  $k \times r$ ;

5) *нахождение всех комбинаций циклического кода данного сомножества.* Это достигается суммированием по модулю 2 всевозможных сочетаний строк образующей матрицы, как было показано при рассмотрении циклического кода с  $d = 2$ ;

б) при индивидуальном кодировании любой из кодовых комбинаций, принадлежащей к сомножеству  $k$  разрядных комбинаций, поступают по общей методике в соответствии с (2.29).

**Пример 2.7.** Образовать циклический код, позволяющий обнаруживать двукратные ошибки или исправлять одиночные ошибки из всех комбинаций двоичного кода на все сочетания с числом информационных символов  $k = 5$ .

**Решение.** По уравнению (2.5) находим число контрольных символов

$$r = E \log((k + 1) + E \log(k + 1)) = E \log((5 + 1) + E \log(5 + 1)) = 4.$$

Из прил. 1 выбираем образующий многочлен  $P(X) = x^4 + x + 1$ . Находим остатки от деления единицы с нулями на  $P(X)$ , которые соответственно равны

0011, 0110, 1100, 1011, 0101.

Строим образующую матрицу

$$M = \begin{pmatrix} 00001 & 0011 \\ 00010 & 0110 \\ 00100 & 1100 \\ 01000 & 1011 \\ 10000 & 0101 \end{pmatrix} \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{matrix}.$$

Так как все члены единичной матрицы являются комбинациями заданного пятиразрядного двоичного кода, то пять комбинаций образующей матрицы представляют собой пять комбинаций требуемого циклического кода. Остальные 26 комбинаций циклического кода (начиная с шестой) могут быть получены путем суммирования по модулю 2 строк образующей матрицы в различном сочетании.

**Пример 2.8.** Закодировать комбинацию  $G(X) = 111011$  циклическим кодом с  $d = 3$ .

**Решение.** Находим число контрольных символов по (2.5)

$$r = E \log((6 + 1) + E \log(6 + 1)) = 4.$$

Из прил. 1 выбираем образующий многочлен  $P(X) = x^4 + x^3 + 1$ .

Умножая  $G(X)$  на  $x^r$ , получим  $G(X) \cdot x^r = 1110110000$ .

Разделив полученный результат, на  $P(X) = 11001$ , найдем остаток  $R(X) = 1110$ . И тогда окончательно в соответствии с (2.29) получаем кодовую комбинацию в циклическом коде с  $d = 3$ :

$$F(X) = 1110111110.$$

**Циклические коды с  $d=4$ .** Эти коды могут обнаруживать одиночные, двойные и тройные ошибки или обнаруживать двойные и исправлять одиночные. При построении данного кода придерживаются следующего порядка:

1) *выбор числа контрольных символов.* Число контрольных символов в этом коде должно быть на единицу больше, чем для кода с  $d = 3$ :

$$r_{d=4} = r_{d=3} + 1; \tag{2.30}$$

2) *выбор образующего многочлена.* Образующий многочлен  $P(X)_{d=4}$  равен произведению двучлена  $(x + 1)$  на многочлен  $P(X)_{d=3}$ :

$$P(X)_{d=4} = P(X)_{d=3}(x + 1). \tag{2.31}$$

Это объясняется тем, что двучлен  $(x + 1)$  позволяет обнаруживать все одиночные и тройные ошибки, а многочлен  $P(X)_{d=3}$  – двойные ошибки.

В общем случае степень генераторного полинома  $P(X)_{d=4}$  равно числу  $r$ . Дальнейшая процедура кодирования остается такой же, как и при образовании кода с  $d = 3$ .

**Пример 2.9.** Требуется закодировать сообщение  $C(X) = 10101010101010$  циклическим кодом с  $d = 4$ .

**Решение.** Определяем число контрольных символов по уравнению (2.5):

$$r_{d=3} = E \log((14 + 1) + E \log(14 + 1)) = E \log(15 + 4) = 5.$$

Из уравнения (2.30) следует, что  $r_{d=4} = 5 + 1 = 6$ .

Выбираем из прил. 1 образующий полином для  $d = 3$ . Пусть  $P(X)_{d=3} = x^5 + x^2 + 1$ . Тогда

$$P(X)_{d=4} = (x + 1)(x^5 + x^2 + 1) = x^6 + x^5 + x^3 + x^2 + x + 1 = 1101111.$$

Так как необходимо закодировать только одно сообщение  $G(X)$ , а не весь ансамбль двоичных кодов с  $k = 14$ , то в дальнейшем будем придерживаться процедуры кодирования, выполняемой по уравнению (2.29). Выбираем одночлен  $x^r = x^6$ . Тогда

$$x^r \cdot G(X) = x^{19} + x^{17} + x^{15} + x^{13} + x^{11} + x^9 + x^7 \rightarrow 10101010101010000000.$$

Разделив полученное выражение на  $P(X)_{d=4}$ , находим остаток:

$$R(X) = x^4 + x^3 + x^2 + x + 1 \rightarrow 011111.$$

Следовательно, передаваемая закодированная комбинация будет иметь вид

$$F(X) = \underbrace{10101010101010}_{\text{информационные символы}} \underbrace{011111}_{\text{контрольные символы}}.$$

**Циклические коды с  $d \geq 5$ .** Эти коды, разработанные Боузом, Чоудхури и Хоквинхемом (сокращенно код БЧХ), позволяют обнаруживать и исправлять любое число ошибок. Заданными при кодировании является число исправляемых ошибок  $s$  и длина слова  $n$ . Число информационных символов  $k$  и контрольных символов  $r$ , а также состав контрольных символов подлежат определению.

Методика кодирования такова:

1) *выбор длины слова*. При кодировании по методу БЧХ нельзя выбирать произвольную длину слова  $n$ . Первым ограничением является то, что слово может иметь только нечетное число символов. Во-вторых, при заданном  $n$  должно соблюдаться одно из равенств:

$$2^h - 1 = n; \quad (2.32)$$

$$(2^h - 1)/q = n, \quad (2.33)$$

где  $h > 0$  – целое число,  $q$  – нечетное положительное число, при делении на которое  $n$  получается целым нечетным числом.

Так, при  $h=6$  длина слова может быть равна не только 63 (2.32), но и 21 при  $q = 3$  (2.33).

2) *определение кодового расстояния*. Кодовое расстояние определяют согласно (2.2), т.е.  $d = 2s + 1$ ;

3) *определение образующего многочлена  $P(X)$* . Образующий многочлен есть наименьшее общее кратное (НОК) так называемых минимальных многочленов  $M(X)$  до порядка  $2s - 1$  включительно, причем берутся все нечетные:

$$P(X) = \text{НОК}[M_1(X) M_3(X) \dots M_{2s-1}(X)]. \quad (2.34)$$

Таким образом, число минимальных многочленов равно  $L = s$ , т.е. равно числу исправленных ошибок. Минимальные многочлены являются простыми неприводимыми многочленами (прил. 2);

4) *определение старшей степени  $t$  минимального многочлена*. Степень  $t$  есть такое наименьшее целое число, при котором  $2^t - 1$  нацело делится на  $n$  или  $nc$ , т.е.  $n = 2^t - 1$  или  $2^t - 1 = cn$ . Отсюда следует, что

$$t = h; \quad (2.35)$$

5) *выбор минимальных многочленов*. После того как определено число минимальных многочленов  $L$  и степень старшего многочлена  $t$ , многочлены выписывают из прил. 2. При этом НОК может быть составлено не только из многочленов старшей степени  $t$ . Это, в частности, касается многочленов четвертой и шестой степеней;

6) *определение степени  $\beta$  образующего многочлена  $P(X)$* . Степень образующего многочлена зависит от НОК и не превышает произведения  $t \cdot s$ ;

7) *определение числа контрольных символов*. Так как число контрольных символов  $r$  равно степени образующего полинома, то в коде длины  $n$

$$\beta = r \leq t \cdot s; \quad (2.36)$$

8) *определение числа информационных символов.* Его производят обычным порядком из равенства

$$k = n - r. \quad (2.37)$$

Дальнейшие этапы кодирования аналогичны рассмотренным для циклических кодов с  $d < 4$ .

**Пример 2.10.** Закодировать все комбинации двоичного кода, чтобы  $n = 15$ , а  $s = 2$ .

**Решение.** Определяем кодовое расстояние по (2.2):  $d = 2s + 1 = 2 \cdot 2 + 1 = 5$  (код БЧХ). Число минимальных многочленов  $L = s = 2$ . Старшая степень минимального многочлена по (2.35)  $t = h = 4$ , так как  $15 = 2^4 - 1$ . Выписываем из прил. 2 минимальные многочлены:  $M_1(X) = x^4 + x + 1$  и  $M_3(X) = x^4 + x^3 + x^2 + x + 1$ . Образующий многочлен определяем по (2.34):

$$P(X) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) = x^8 + x^7 + x^6 + x^4 + 1.$$

Число контрольных символов  $r$  равно по (2.36) степени  $\beta$  образующего многочлена, т. е.  $r = \beta = 8$ , а значит, число информационных символов  $k$  по (2.37) равно:  $k = n - r = 15 - 8 = 7$ . Таким образом, получаем код БЧХ (15, 7) с  $s = 2$ .

После нахождения остатков получаем образующую матрицу

$$\left( \begin{array}{cccc|cccccccc} k_7 & k_6 & k_5 & k_4 & k_3 & k_2 & k_1 & r_8 & r_7 & r_6 & r_5 & r_4 & r_3 & r_2 & r_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{array} \right) \quad (2.38)$$

**Пример 2.11.** Найти образующий полином для циклического кода, исправляющего двукратные ошибки,  $s = 2$ , если общая длина кодовых комбинаций  $n = 21$ .

**Решение.** Определяем, что  $d = 2 \cdot 2 + 1 = 5$ ,  $L = 2$ . Наименьшее значение  $t$ , при котором  $2^t - 1$  нацело делится на 21, есть число 6. Из таблицы прил. 2 выписываем два минимальных многочлена, номера которых определяют следующим образом: берут многочлены  $M_1(X)$  и  $M_3(X)$  и их индексы умножают на  $q = c = 3$ . В результате получаем  $M_3(X)$  и  $M_9(X)$ . Таким образом,

$$P(X) = \text{НОК}[M_3(X)M_9(X)] = (x^6 + x^4 + x^2 + x + 1)(x^3 + x^2 + 1) = \\ = x^9 + x^8 + x^7 + x^5 + x^4 + x + 1,$$

откуда  $r = 9$ , а  $k = 12$ . Получаем код БЧХ (21, 12). Кроме того, доказано, что этот код, имеющий  $d = 7$ , может также обнаруживать и исправлять ошибки кратностью  $s = 3$ .

Построение кодов БЧХ возможно и с помощью таблицы [1], которая приведена в прил. 3 в сокращенном виде. В соответствии с изложенной ранее методикой в таблице по заданным длине кодовой комбинации  $n$  и числу исправляемых ошибок  $s$  рассчитаны число информационных символов  $k$  и образующий многочлен  $P(X)$ . Число контрольных символов  $r$  определяется из уравнения  $r = n - k$ , и запись образующего многочлена в виде десятичных цифр преобразуется путем перевода каждой десятичной цифры в трехразрядное двоичное число. Например, во второй строке таблицы  $P(X) = 23$ . Цифре 2 соответствует двоичное число 010, а цифре 3 – число 011. В результате получаем двоичное число 010011, которое записывается в виде многочлена  $x^4 + x + 1$ . Таким образом, в двоичный эквивалент переводится каждая из десятичных цифр, а не все десятичное число. Действительно, числу 23 соответствует уже многочлен  $P(X) = x^4 + x + 1$ . Из прил. 3 следует, что при  $n = 31$ ,  $k = 21$  и  $s = 2$  образующий многочлен

$$P(X) = x^{10} + x^9 + x^8 + x^6 + x^5 + x + 1 = 011101101001 .$$

**Коды БЧХ для обнаружения ошибок.** Их строят следующим образом. Если необходимо образовать код с обнаружением четного числа ошибок, то по заданному числу обнаруживаемых ошибок  $m$  согласно (2.1) и (2.2) находят значения  $d$  и  $s$ . Дальнейшее кодирование выполняют, как и раньше. Если требуется обнаружить нечетное число ошибок, то находят ближайшее меньшее целое число  $s$  и кодирование производят так же, как и в предыдущем случае, с той лишь разницей, что найденный согласно (2.34) образующий многочлен дополнительно умножают на двучлен  $x + 1$ .

**Пример 2.12.** Построить код БЧХ, обнаруживающий пять ошибок при длине кодовых комбинаций  $n = 15$ .

**Решение.** Находим, что  $d = m + 1 = 5 + 1 = 6$ , а ближайшее меньшее значение  $s$  определим из выражения  $d = 2s + 1$ , откуда  $s = (d - 1)/2 = 5/2 = 2$ . Далее определяем многочлен  $P(X)$ , как указано в примере 2.10, и умножаем его на двучлен  $(x + 1)$ , т. е. получаем

$$P(X) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x + 1) = (x^8 + x^7 + x^6 + x^4 + 1)(x + 1) = \\ = x^9 + x^8 + x^7 + x^5 + x + x^8 + x^7 + x^6 + x^4 + 1 = x^9 + x^6 + x^5 + x^4 + x + 1.$$

Таким образом, получаем код БЧХ (15, 6).

**Коды Файра.** Это циклические коды, обнаруживающие и исправляющие пакеты ошибок. Определение пакета (пачки) ошибки дано в подразд. 2.1. Непременным условием пакета данной длины  $b$  является поражение крайних символов и нахождение между ними  $b - 2$  разрядов.

Коды Файра могут исправлять пакет ошибок длиной  $b_s$  и обнаруживать пакеты ошибок длиной  $b_m$ . Заметим, что в кодах Файра понятие кодового расстояния  $d$ , а следовательно, и уравнение (2.3) не используются.

Образующий многочлен кода Файра  $P(X)_\Phi$  определяется из выражения

$$P(X)_\Phi = P(X)(X^C + 1), \quad (2.39)$$

где  $P(X)$  – неприводимый многочлен степени

$$t \geq b_s, \quad (2.40)$$

принадлежащий показателю степени

$$E = 2^t - 1; \quad (2.41)$$

$$C \geq b_s + b_m - 1. \quad (2.42)$$

Неприводимый многочлен  $P(X)$  выбирают из прил. 1 согласно уравнению (2.40).

Длина слова  $n$  равна наименьшему общему кратному чисел  $E$  и  $C$ , так как только в этом случае многочлен  $x^n + 1$  делится на  $P(X)_\Phi$  без остатка. При  $n^* < n$  никакой многочлен  $x^{n^*} + 1$  не делится на  $P(X)_\Phi$ . Таким образом,

$$n = \text{НОК}(E, C). \quad (2.43)$$

Число контрольных символов

$$r = t + C. \quad (2.44)$$

Дальнейшая процедура кодирования такая же, как и для циклического кода с  $d = 3$ .

**Пример 2.13.** Найти образующий полином и определить общую длину кодовых комбинаций  $n$ , а также число контрольных и информационных символов для кода, позволяющего исправлять пакеты ошибок длиной  $b_s = 4$  и обнаруживать пакеты ошибок длиной  $b_m = 5$ .

Исправить пакет  $b_s = 4$  – значит исправить одну из следующих комбинаций ошибок, пораженных помехами: 1111, 1101, 1011 и 1001. В то же время этот код может обнаруживать одну из комбинаций в пять символов: 11111, 10111, 10011, 10001 и т.д.

На основании (2.40) и (2.42)  $t \geq 4$ ,  $C \geq 8$ . Из прил. 1 находим неприводимый многочлен четвертой степени:  $P(X) = x^4 + x + 1$ . Согласно (2.39) образующий многочлен  $P(X)_\phi = (x^4 + x + 1)(x^8 + 1) = x^{12} + x^9 + x^8 + x^4 + x + 1$ . По выражению (2.41) находим  $E = 2^4 - 1 = 15$ . Поэтому длина кода из (2.43)  $n = 15 \cdot 8 = 120$ . Из (2.44) число контрольных символов  $r = 4 + 8 = 12$ . В итоге получаем циклический код (120, 108). Избыточность такого кода, если учитывать его исправляющую способность, невелика:  $R = 12/120 = 0,1$ .

**Сравнение кодов БЧХ и Файра.** Представляет интерес сравнение по избыточности кода при исправлении того же числа ошибок, но не сгруппированных в пакет, т.е. рассеянных по всей длине слова. Если воспользоваться для этой цели кодами БЧХ и близким значением  $n = 127$ , то при  $s = 4$  можно по изложенной методике подсчитать, что число контрольных символов  $r = 28$ , т.е. получен код (127, 99). Избыточность такого кода  $R = 28/127 = 0,22$ , т.е. значительно выше, чем у кода Файра. Это очевидно: исправить четыре ошибки, находящиеся в одном месте, проще, чем ошибки, рассредоточенные по всей длине комбинации.

Заметим, что существует следующее правило: если циклический код рассчитан на обнаружение независимых ошибок, он может обнаруживать также пакет ошибок длиной  $b_m$ .

**Укороченные циклические коды.** Предположим, что требуется получить 15 комбинаций, закодированных так, чтобы в любой из них могло исправляться по две ошибки, т.е.  $s = 2$ ,  $d = 5$ . Для этого следует взять код с числом информационных символов  $k = 4$ . Код (7, 4) не подходит, так как он исправляет только одну ошибку. Как указывалось, число  $n$ , промежуточное между 7 и 15, в коде БЧХ брать нельзя. Поэтому необходимо взять код (15, 7), рассмотренный в примере 2.10. Однако разрешенных комбинаций в таком коде ( $2^7$ ) значительно больше 15, поэтому код (15, 7) укорачивают путем вычеркивания трех столбцов слева и трех строк снизу, как это показано пунктирной линией в образующей матрице (2.38). В результате образующая матрица укороченного кода (12, 4) принимает вид

$$M = \begin{vmatrix} k_4 & k_3 & k_2 & k_1 & r_8 & r_7 & r_6 & r_5 & r_4 & r_3 & r_2 & r_1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{vmatrix} \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{matrix} \quad (2.45)$$

В матрице (2.45)  $d_{\min} = 5$ . И она представляет четыре кодовые комбинации в коде БЧХ, остальные 11 комбинаций укороченного циклического кода (12, 4) могут быть получены суммированием комбинаций образующей матрицы.

Корректирующая способность укороченного циклического кода, по крайней мере, не ниже корректирующей способности исходного полного циклического кода. Техника кодирования и декодирования в обоих случаях одна и та же. Однако циклический сдвиг кодовой комбинации укороченного циклического кода не всегда приводит к образованию разрешенной комбинации, поэтому укороченные коды относят к числу *псевдоциклических*.

**Декодирование циклических кодов. Обнаружение ошибок.** Идея обнаружения ошибок в принятом циклическом коде заключается в том, что при отсутствии ошибок закодированная комбинация  $F(X)$  делится на образующий многочлен  $P(X)$  без остатка. При этом контрольные символы  $r$  отбрасываются, а информационные символы  $k$  используются по назначению. Если произошло искажение принятой комбинации, то эта комбинация  $F(X)$  преобразуется в комбинацию

$$F^*(X) = F(X) + E(X), \quad (2.46)$$

где  $E(X)$  – многочлен ошибок, содержащий столько единиц, сколько элементов в принятой комбинации не совпадает с элементами переданной комбинации.

Пусть, например, была передана комбинация кода  $(7, 4)$   $F(X) = 1101001$ , закодированная с помощью  $P(X) = 1011$ . Если она принята правильно, то деление на  $P(X)$  дает остаток, равный нулю. Если же комбинация принята как  $F^*(X) = 1101011$ , то при делении на  $P(X)$  образуется остаток  $R(X) = 010$ , что свидетельствует об ошибке, и принятая комбинация бракуется.

**Обнаружение и исправление ошибок.** Существует несколько вариантов декодирования циклических кодов [2]. Один из них заключается в следующем:

1) *вычисление остатка*. Принятую кодовую комбинацию делят на  $P(X)$ , и если остаток  $R(X) = 0$ , то комбинация принята без искажений. Наличие остатка свидетельствует о том, что комбинация принята искаженной. Дальнейшая процедура исправления рассматривается ниже;

2) *подсчет веса остатка  $w$* . Если вес остатка равен или меньше числа исправляемых ошибок, т.е.  $w \leq s$ , то принятую комбинацию складывают по модулю 2 с остатком и получают исправленную комбинацию;

3) *циклический сдвиг на один символ влево*. Если  $w > s$ , то производят циклический сдвиг на один символ влево и полученную комбинацию снова делят на  $P(X)$ . Если вес полученного остатка  $w \leq s$ , то циклически сдвинутую комбинацию складывают с остатком и затем циклически сдвигают ее в обратную сторону вправо на один символ. В результате получают исправленную комбинацию;

4) *дополнительные циклические сдвиги влево*. Если после циклического сдвига на один символ по-прежнему  $w > s$ , производят дополнительные циклические сдвиги влево. При этом после каждого сдвига сдвинутую комбинацию делят на  $P(X)$  и проверяют вес остатка. При  $w \leq s$  выполняют действия,

указанные в подразд. 3, с той лишь разницей, что обратных циклических сдвигов вправо делают столько, сколько их было сделано влево.

**Пример 2.14.** Пусть исходная комбинация  $G(X) = 1001$ , закодированная с помощью  $P(X) = 1011$  и  $s = 1$ , имела вид  $F(X) = 1001110$ . При передаче по каналу связи была искажена и в приемник поступила в виде  $F^*(X) = 1101110$ . Проверить наличие ошибки и в случае обнаружения исправлять ее.

Делим комбинацию 1101110 на 1011 и находим, что остаток  $R(X) = 111$ . Так как  $w = 3 > s = 1$ , то сдвигаем комбинацию 1101110 циклически на один символ влево. Получаем 1011101. В результате деления этой комбинации на  $P(X)$  находим остаток  $R(X) = 101$ . Вес этого остатка  $w = 2 > s = 1$ . Осуществляем новый циклический сдвиг влево. Получаем 0111011. Деление на  $P(X)$  дает остаток  $R(X) = 001$ , вес которого равен  $s$ . Складываем:  $0111011 \oplus 001 = 0111010$ . Теперь осуществляем два циклических сдвига последней комбинации вправо: после первого она принимает вид 0011101, после второго 1001110, т.е. получается уже исправленная комбинация. Проверка показывает, что эта комбинация делится на  $P(X)$  без остатка.

**Пример 2.15.** При передаче комбинации, представленной в седьмой строке матрицы (2.38), исказились два символа и комбинация была принята в виде  $\overset{\cdot\cdot}{111000011101000}$  (искаженные символы помечены точками). Непосредственное деление этой комбинации на  $P(X) = x^8 + x^7 + x^6 + x^4 + 1$  дает остаток весом  $w = 4$ . После первого циклического сдвига комбинация принимает вид  $\overset{\cdot\cdot}{110000111010001}$ . Деление этой комбинации на  $P(X)$  снова дает остаток с весом  $w = 4$ . После второго сдвига и повторного деления ничего не меняется. Вес остатка  $w = 4$ . Делаем третий сдвиг, комбинация принимает вид  $\overset{\cdot\cdot}{000011101000111}$ . И вновь делим на  $P(X)$ . На этот раз остаток  $R(X) = 00000011$  имеет вес  $w = 2 = s = 2$ . Складываем  $\overset{\cdot\cdot}{000011101000111} \oplus 00000011$ , получаем  $\overset{\cdot\cdot}{000011101000100}$ . Произведя три циклические сдвига комбинации вправо, получаем исходную комбинацию 100000011101000.

Второй метод определения номеров элементов, в которых произошла ошибка, основан на свойстве, которое заключается в том, что остаток  $R(X)$ , полученный при делении принятой кодовой комбинации  $F^*(X)$  на  $P(X)$ , равен остатку  $R^*(X)$ , полученному в результате деления соответствующего многочлена ошибок  $E(X)$  на  $P(X)$ .

Многочлен ошибок может быть представлен в следующем виде  $E(X) = F(X) + F^*(X)$ , где  $F(X)$  - исходный многочлен циклического кода. Так, если ошибка произошла в первом символе, то  $E_1(X) = 100\dots 0$ , если во втором -  $E_2(X) = 010\dots 0$  и т.д. Остатки от деления каждого многочлена  $E_i(X)$



$$F(X) = 10111111100 \oplus 00001000000 = 10110111100.$$

**Мажоритарное декодирование циклических кодов.** Мажоритарный способ исправления ошибок основан на принятии решения о значении того или иного разряда декодируемой кодовой комбинации по большинству результатов проверок на четность.

Проверки на четность для каждого разряда составляются на основании некоторой матрицы  $L$ , которая составляется из проверочной матрицы  $N$  путем  $\mu$  – линейных операций над строками.

Матрица  $L$  характеризуется двумя свойствами:

- 1) один из столбцов содержит только единичные элементы;
- 2) все остальные столбцы содержат не более чем по одному единичному элементу.

Проверочная матрица может быть построена путем вычисления так называемого проверочного полинома

$$h(X) = \frac{X^n + 1}{P^{-1}(X)}, \quad (2.46)$$

где  $P^{-1}(X)$  – полином, сопряженный с  $P(X)$ .

В сопряженных  $P^{-1}(X)$  – полиномах члены расположены в обратном порядке. Так, например,  $P(X) = 100011$ , а  $P^{-1}(X) = 110001$ .

Первая строка проверочной матрицы циклического кода есть проверочный полином  $h(X)$ , умноженный на  $X^{r-1}$  (т.е. дополненный справа  $r-1$  нулями). Последующие строки проверочной матрицы есть циклический сдвиг вправо первой. Число сдвигов равно числу дописанных справа нулей. Матрица  $L$  определяет  $\mu$  проверок на четность для разряда, соответствующего единичному столбцу. Добавив к этой совокупности проверок тривиальную проверку  $a_i = a_i$ , получим  $\mu + 1$  независимых проверочных соотношений для одного разряда  $a_i$ , причем свойства матрицы  $L$  таковы, что каждый разряд кодовой комбинации входит только в одну проверку. Такая совокупность проверок называется системой разделенных (ортогональных) проверок относительно разряда  $a_i$ . Системы разделенных проверок для остальных разрядов получают циклическим сдвигом строк матрицы  $L$ , что равносильно добавлению единицы к индексу разряда предыдущей проверки, причем при добавлении единицы к номеру старшего разряда номер последнего заменяется на нуль.

Мажоритарное декодирование осуществляется следующим образом. Если в принятой кодовой комбинации ошибки отсутствуют, то при определении значения разряда  $a_i$  все  $\mu + 1$  проверки укажут одно и то же значение (либо 1, либо 0). Одиночная ошибка в кодовой комбинации может вызвать искажение

лишь одной проверки, двойная ошибка – двух и т. д. Решения о значении разряда  $a_i$  принимаются по большинству (т.е. мажоритарно) одноименных результатов проверок. При этом декодирование безошибочно, если число ошибок в кодовой комбинации не превышает  $\mu/2$ , т.е. искажено не более  $\mu/2$  проверок. Если все системы разделенных проверок для каждого разряда кодовой комбинации содержат не менее  $\mu + 1$  разделенных проверок, то реализуемое минимальное кодовое расстояние

$$d_{\min} = \mu + 1. \quad (2.47)$$

Поясним принцип мажоритарного декодирования на конкретных примерах.

**Пример 2.17.** Построить матрицы  $N$  и  $L$  и найти систему проверок для циклического кода  $(7, 3)$ , образованного с помощью полинома  $P(X) = (x^3+x+1) \times (x+1) = x^4 + x^3 + x^2 + 1$  и позволяющего обнаруживать двойные и исправлять одиночные ошибки.

**Решение.** Находим проверочный полином

$$h(X) = \frac{X^n + 1}{P^{-1}(X)} = \frac{x^7 + 1}{x^4 + x^2 + x + 1} = x^3 + x + 1 \rightarrow 1011.$$

Строим проверочную матрицу

$$N = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Для построения матрицы  $L$  преобразуем матрицу следующим образом. Сложим 2-, 3- и 4-ю строки матрицы:

$$0101100 \oplus 0010110 \oplus 0001011 = 0110001.$$

Аналогично сложим 1-, 3- и 4-ю строки:

$$1011000 \oplus 0010110 \oplus 0001011 = 1000101.$$

Составим матрицу  $L$ , используя для ее построения две полученные суммы и 4-ю строку проверочной матрицы  $N$ :

$$L = \begin{vmatrix} a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}.$$

Легко видеть, что в этой матрице один из столбцов состоит только из единиц, а все остальные столбцы содержат не более одной единицы. Матрица  $L$  дает три независимых проверочных соотношений с разделенными относительно члена  $a_0$  проверками. Добавив к этим соотношениям тривиальную проверку  $a_0 = a_0$ , получим систему разделенных относительно  $a_0$  проверок:

$$\left. \begin{array}{l} a_0 = a_4 \oplus a_5 \\ a_0 = a_2 \oplus a_6 \\ a_0 = a_1 \oplus a_3 \\ a_0 = a_0 \end{array} \right\}. \quad (2.48)$$

Систему проверок для  $a_1$  получим из (2.48) в виде

$$\left. \begin{array}{l} a_1 = a_5 \oplus a_6 \\ a_1 = a_3 \oplus a_0 \\ a_1 = a_2 \oplus a_4 \\ a_1 = a_1 \end{array} \right\}. \quad (2.49)$$

Для остальных разрядов  $a_2 \dots a_6$  можно получить аналогичные системы проверок.

**Пример 2.18.** Исходная комбинация  $G(X) = 101$ , закодированная генераторным полиномом  $P(X) = x^4 + x^3 + x^2 + 1$ , поступила в канал связи в виде  $F(X) = 1010011$ . В результате действия помех была искажена (одиночная ошибка) и в приемник поступила в виде  $F^*(X) = 101001\overset{\bullet}{0}$ . Воспользовавшись системой проверок примера 2.17, определить номер искаженного разряда и исправить его.

**Решение.** Пронумеруем разряды принятой кодовой комбинации следующим образом:

$$F^*(X) = \begin{array}{cccccccc} & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{array}.$$

Произведем проверку правильности приема символа  $a_0$  по выражениям (2.48):

$$\begin{aligned}
 a_0 &= 1 \oplus 0 = 1, & a_0 &= 1 \oplus 0 = 1, \\
 a_0 &= 0 \oplus 1 = 1, & 0 &= 0.
 \end{aligned}$$

Большинство проверок указывают, что разряду  $a_0$  должен быть присвоен символ 1. Таким образом, исправленная комбинация будет  $F(X) = 1010011$ , что соответствует переданной в канал связи.

Для остальных разрядов проверки не проводились, так как в условии задачи указано, что имела место одиночная ошибка.

**Пример 2.19.** Найти систему проверок для символа  $a_0$  кода БЧХ (15, 7), образованного генераторным полиномом  $P(X) = x^8 + x^7 + x^6 + x^4 + 1 = 111010001$  и позволяющего исправлять двойные ошибки.

**Решение.** Вычислим проверочный полином:

$$h(X) = \frac{X^n + 1}{P^{-1}(X)} = x^7 + x^3 + x + 1 \rightarrow 10001011.$$

Построим проверочную матрицу, в качестве первой строки которой используем проверочный полином, умноженный на  $X^{r-1}$ , а остальные строки получим циклическим сдвигом первой:

$$N = \begin{pmatrix}
 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1
 \end{pmatrix}.$$

Преобразуем проверочную матрицу следующим образом. Сложим по модулю два 1-, 5-, 7- и 8-ю; 2-, 3-, 6-, 7- и 8-ю; 4-, 6-, 7- и 8-ю строки матрицы и в результате получим кодовые комбинации соответственно:

$$\begin{aligned}
 &1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1, \\
 &0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1, \\
 &0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1.
 \end{aligned}$$

Составим матрицу  $L$ , используя для ее построения три полученные суммы и 8-ю строку проверочной матрицы  $N$ :

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$a_{14} \ a_{13} \ a_{12} \ a_{11} \ a_{10} \ a_9 \ a_8 \ a_7 \ a_6 \ a_5 \ a_4 \ a_3 \ a_2 \ a_1 \ a_0$$

Полученная матрица удовлетворяет требованиям, предъявляемым к матрице  $L$ .

Данная матрица  $L$  дает четыре независимых проверочных соотношения с разделенными относительно члена  $a_0$  проверками; добавив к ним тривиальную проверку  $a_0 = a_0$ , получим следующую систему для проверки  $a_0$ :

$$\begin{aligned} a_0 &= a_1 \oplus a_3 \oplus a_7, \\ a_0 &= a_2 \oplus a_6 \oplus a_{14}, \\ a_0 &= a_4 \oplus a_{12} \oplus a_{13}, \\ a_0 &= a_8 \oplus a_9 \oplus a_{11}, \\ a_0 &= a_0. \end{aligned}$$

Пусть при передаче был искажен разряд  $a_6$ . Этот разряд входит только во вторую проверку, поэтому четыре проверки дадут правильный результат, а вторая проверка – неправильный. Решение о значении разряда  $a_0$  принимается по критерию большинства и поэтому будет правильным. Ошибочная регистрация разряда произойдет при действии трех и более ошибок, приводящих к неправильным результатам трех и более проверок.

Системы отдельных проверок для остальных разрядов получаются циклическим сдвигом строк матрицы  $L$ .

**2.3.4. Итеративные коды.** Данные коды характеризуются наличием двух или более систем проверок внутри каждой кодовой комбинации. Принцип построения итеративного кода проще всего представлять на конкретном примере. Запишем все информационные разряды блока, подлежащего передаче, в виде таблицы (рис. 2.3).

Каждая строка этой таблицы кодируется каким-либо кодом, а затем кодируется каждый столбец, причем не обязательно тем же кодом. Символы, расположенные в правом нижнем углу таблицы, получаются в результате проверки проверочных символов. Они могут быть построены на основе проверки по строкам и тогда будут удовлетворять проверке по столбцам, и наоборот.

В качестве примера рассмотрим итерированные коды (рис. 2.4) с одной проверкой на четность для каждого столбца и строки. Такой код имеет большую корректирующую способность по сравнению с кодом с одной проверкой на четность, который позволяет только обнаруживать нечетно-кратные ошибки.

Итерированный код позволяет исправить все одиночные ошибки, так как пересечение строки и столбца, содержащих ошибку, однозначно указывает ее место. Передача комбинации итеративного кода обычно происходит по строкам последовательно, от первой строки к последней.

Свойства итеративного кода полностью определяются параметрами итерируемых кодов, в зависимости от которых итеративный код может быть как систематическим, так и несистематическим, как разделимым, так и неразделимым. Длина кодовой комбинации, число информационных разрядов и минимальное кодовое расстояние итеративного кода очень просто выражаются через соответствующие параметры этих кодов:

$$n = \prod_{i=1}^S n_i, \quad k = \prod_{i=1}^S k_i, \quad d_{\min} = \prod_{i=1}^S d_i, \quad (2.50)$$

где  $n_i, k_i, d_i$  – параметры итерируемых кодов;  
 $S$  – кратность итерирования.

Информационные символы	Проверочные символы по строкам
Проверочные символы по столбцам	Проверка проверок

Рис. 2.3. Расположение символов итеративного кода

1	0	0	1	1	1
1	1	0	0	1	1
1	0	0	0	1	0
1	1	1	1	1	1
1	9	9	9	9	1
0	0	1	1	0	0
1	0	0	1	0	0

Рис. 2.4. Итеративный код

Таким образом, простейший итеративный код, образованный путем проверок на четность (нечетность) строк и столбцов, обладает минимальным кодовым расстоянием  $d_{\min} = 4$  и поэтому позволяет обнаруживать все ошибки кратности до 3. Не обнаруживаются четырехкратные ошибки, располагающиеся в вершинах правильного четырехугольника, а также некоторые шестикратные, восьмикратные и т.д. ошибки (рис. 2.5).

Простейший итеративный код обладает довольно высокими обнаруживающими способностями при действии пакетных ошибок – обнаруживается любой пакет ошибок длиной  $l + 1$  и менее, где  $l$  – длина строки.

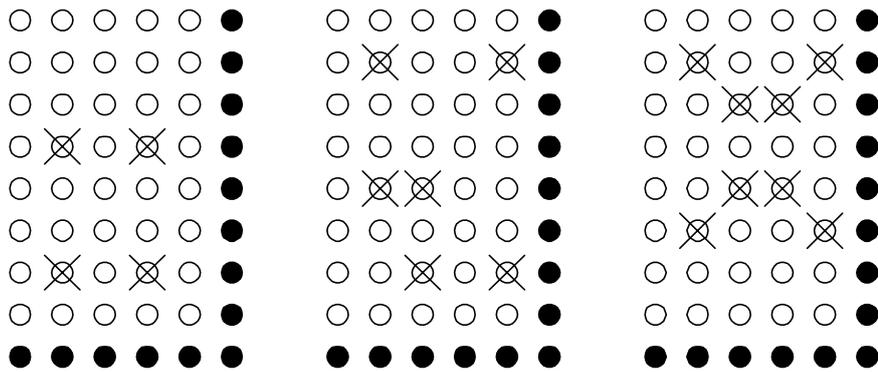


Рис. 2.5. Ошибки, не обнаруживаемые простейшим итеративным кодом

Могут быть образованы также многомерные итеративные коды, в которых каждый информационный разряд входит в комбинации трех, четырех и т.д. итерируемых кодов.

На рис. 2.6 показан пример применения третьей проверки по диагонали. Порядок формирования контрольных символов  $P_i$  показан сплошными линиями.

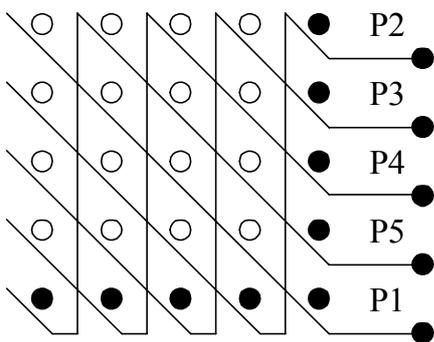


Рис. 2.6. Итеративный код с тремя проверками

На практике наибольшее распространение получили двумерные итеративные коды. Длина строки обычно выбирается равной длине одного знака первичного кода. В качестве итерируемых кодов чаще всего используются коды с одной и двумя проверками на четность и коды Хемминга.

**2.3.5. Рекуррентные коды.** Эти коды относятся к числу непрерывных кодов, в которых операции кодирования и декодирования производятся непрерывно над последовательностью информационных символов без деления на блоки. Рекуррентные коды применяются для обнаружения и исправления пакетов ошибок. В данном коде

после каждого информационного элемента следует проверочный элемент. Проверочные элементы формируются путем сложения по модулю два двух информационных элементов, отстоящих друг от друга на шаг сложения, равный  $b$ .

Рассмотрим процесс кодирования на примере кодовой комбинации, приведенной на рис. 2.7 (верхняя строка), если шаг сложения  $b = 2$ . Процесс образования контрольных символов показан на этом же рисунке (нижняя строка).

Кодирование и декодирование производится с помощью регистров сдвига и сумматоров по модулю два.

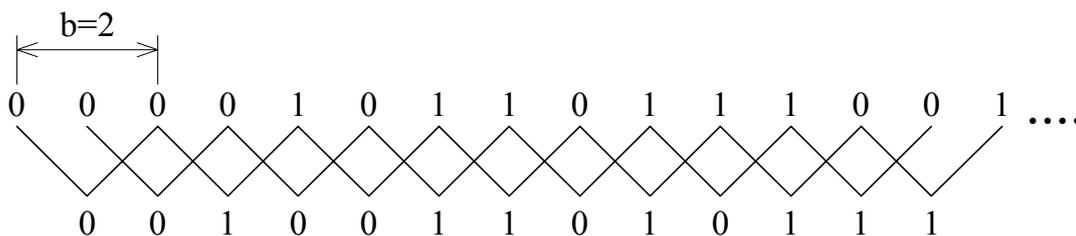


Рис. 2.7. Схема построения рекуррентного кода

На выходе кодирующего устройства (рис. 2.8) получим последовательность символов

$$1000111000111110010011. \quad (2.51)$$

Эта последовательность поступает в дискретный канал связи.

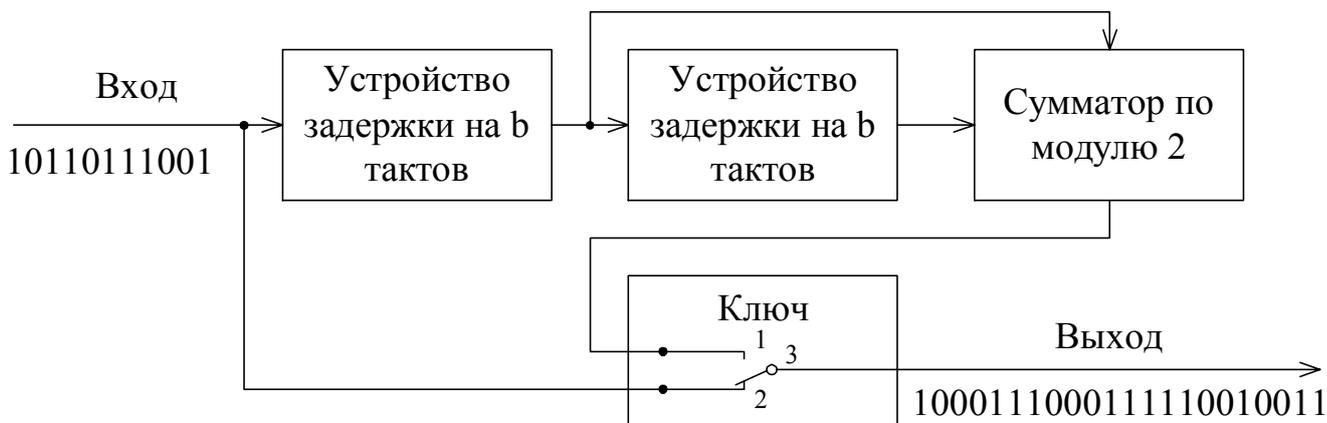


Рис. 2.8. Структурная схема кодера рекуррентного кода

Структурная схема декодера приведена на рис. 2.9. Процесс декодирования заключается в выработке контрольных символов из информационных, поступивших на декодер, и их сравнении с контрольными символами, пришедшими из канала связи. В результате сравнения вырабатывается корректирующая последовательность, которая и производит исправление информационной последовательности. Рассмотрим этот процесс более подробно. Пусть из дискретного канала связи на вход подается искаженная помехами последовательность (искаженные символы обозначены сверху чертой)

$$10001\bar{0}1\bar{1}10111110010011. \quad (2.52)$$

Устройство разделения (рис. 2.9) разделяет последовательность (2.52) на информационные

$$00001011\bar{1}111001 \quad (2.53)$$

и контрольные символы

$$00\bar{0}10110101. \quad (2.54)$$

Последовательности символов (2.53) и (2.54) содержат ошибочные символы, которые подчеркнуты сверху. Формирователь контрольных символов из (2.53) выдает последовательность символов

$$00100100001, \quad (2.55)$$

которая в сумме по модулю два с последовательностью (2.54) дает исправляющую последовательность

$$00110010100. \quad (2.56)$$

Исправляющая последовательность (2.56) подается на инвертор, который выдает последовательность (2.57) и одновременно поступает на устройства задержки на  $b$  и  $2b$  тактов. На выходе устройств задержки появляются последовательности (2.58) и (2.59) соответственно. На выходе схемы совпадения получаем последовательность (2.60)

$$11001101011... \quad (2.57)$$

$$..001100101... \quad (2.58)$$

$$\underline{...0011001...} \quad (2.59)$$

$$00000000001... \quad (2.60)$$

Точки в последовательности слева обозначают задержку символов на соответствующее число тактов. Единица на выходе схемы совпадения возникает только в тех случаях, когда на все ее три входа подаются единицы. Она представляет собой команду исправить ошибку.



Рис. 2.9. Структурная схема декодера рекуррентного кода

Исправленная последовательность вырабатывается устройством коррекции в виде суммы по модулю два последовательности (2.60) и (2.53), задержанной на  $b$  тактов.

$$\begin{array}{r}
 00000000001000000 \\
 \dots\dots\dots 10111111001 \\
 \hline
 10110111001
 \end{array} \tag{2.61}$$

Точки в последовательности слева означают задержку на 6 тактов относительно входа в устройство разделения на информационные и контрольные символы.

После автоматического исправления последовательности (2.61) совпадает с последовательностью на рис. 2.7 (верхняя строка). Как следует из (2.61), на пути информационных символов включено  $3b = 6$  ячеек регистра сдвига. При этом для вывода всех ошибочных символов необходим защитный интервал  $6b + 1 = 13$  символов.

Рассмотренный код позволяет исправлять пакет ошибок длиной  $l = 2b = 4$ .

В заключение следует отметить, что рекуррентный код находит применение в системах связи.

## 2.4. Частотные коды

Частотные коды относятся к нецифровым кодам и применяются для передачи независимых команд, когда нет необходимости во взвешенных кодах. Используются как двухпозиционные, так и многопозиционные коды. На практике находят применение одночастотные коды и коды, в основу принципов комбинирования которых положены математические законы теории соединений. Используются перестановки  $P_n$ , размещения  $A_n^m$ , сочетания  $C_n^m$  и другие законы комбинирования.

**2.4.1. Одночастотный код.** В системах телемеханики с небольшим числом команд часто используют данный код, при котором каждое сообщение передается радиоимпульсом определенной частоты, число сообщений  $N = n_{\text{ч}}$ , где  $n_{\text{ч}}$  – число частот. Во время передачи данного сообщения остальные частоты не передаются.

**2.4.2. Коды, образованные по закону перестановок.** Перестановки  $P_n$  из  $n$  различных частот образуют кодовые комбинации, отличающиеся только порядком следования этих частот. Число элементов во всех комбинациях всегда одинаково. Длина сообщения равна числу частот, т.е.  $m = n_{\text{ч}} = \text{const}$ . Отличительной особенностью этого кода является отсутствие одинаковых частот в одном сообщении. Такой код часто называется **аккордным**. Общее число комбинаций:

$$N = n_{\text{ч}}! \tag{2.62}$$

Например, при трех частотах получается шесть комбинаций:  $f_1f_2f_3$ ,  $f_1f_3f_2$ ,  $f_2f_1f_3$ ,  $f_2f_3f_1$ ,  $f_3f_1f_2$ ,  $f_3f_2f_1$ . Данный код позволяет обнаруживать одиночные искажения, так как в сообщении каждый элемент встречается только один раз.

**2.4.3. Коды, образованные по закону размещений.** Размещения  $A_n^m$  образуют комбинации, которые отличаются друг от друга либо частотами, либо порядком их следования. Количество кодовых комбинаций:

$$N = A_n^m = n!/(n - m)! \quad (2.63)$$

Если, например,  $n = 3$ ,  $m = 2$ , то общее число комбинаций равно шести:  $f_1f_2$ ,  $f_1f_3$ ,  $f_2f_1$ ,  $f_2f_3$ ,  $f_3f_1$ ,  $f_3f_2$ . Комбинации передаются последовательно. Этот код позволяет обнаруживать одиночные ошибки путем счета символов, содержащихся в сообщении.

**2.4.4. Коды на определенное число сочетаний.** С помощью сочетаний  $C_n^m$  можно образовать комбинации, отличающиеся друг от друга только самими частотами. Общее число сообщений, которое можно передать из  $n$  частот по  $m$  частот:

$$N = C_n^m = n!/(m!(n - m)!). \quad (2.64)$$

Так, например если  $n = 4$ ,  $m = 2$ , то можно организовать шесть сообщений:  $f_1f_2$ ,  $f_1f_3$ ,  $f_1f_4$ ,  $f_2f_3$ ,  $f_2f_4$ ,  $f_3f_4$ . Данные коды имеют постоянное число радиоимпульсов (частот) и поэтому могут обнаруживать любые искажения за исключением искажений типа смещение, когда радиоимпульс заменяется на радиоимпульс другой частоты, используемой при формировании всех комбинаций.

**2.4.5. Сменно–качественные коды.** Данные коды широко применяются в устройствах ТУ (ТС) как обладающие свойствами самораспределения. В сменно–качественных кодах соседние символы не могут быть одинаковы, а поэтому дешифратор кода легко может различить различные разряды в сообщении. Пусть необходимо передать кодовую комбинацию  $G(x) = 10011101$  сменно–качественным кодом. Для этой цели 1 передается частотой  $f_1$ , 0 –  $f_2$ , а повторение символа (0 или 1) –  $f_3$ . Тогда комбинация принимает вид  $f_1f_2f_3f_1f_3f_1f_3f_1f_2f_1$ . Нетрудно установить лишь те искажения, в результате которых соседние радиоимпульсы получают одинаковое значение частотного признака, другие искажения не обнаруживаются.

Так, например, если  $n=4$ ,  $m=2$ , то можно передать шесть сообщений:  $f_1f_2$ ,  $f_1f_3$ ,  $f_1f_4$ ,  $f_2f_3$ ,  $f_2f_4$ ,  $f_3f_4$ . Данные коды имеют постоянное число радиоимпульсов (частот) и поэтому могут обнаруживать любые искажения за исключением искажений типа смещение, когда радиоимпульс заменяется на радиоимпульс другой частоты, используемой при формировании всех комбинаций.

### 3. ТЕХНИЧЕСКИЕ СРЕДСТВА ПРЕОБРАЗОВАНИЯ ДЛЯ НЕПОМЕХОЗАЩИЩЕННЫХ КОДОВ

Кодирующим устройством называется преобразователь дискретных сообщений или сигналов (например, импульсов) в кодовые комбинации заданного кода, а *декодирующим* – обратный преобразователь кодовых комбинаций заданного кода в дискретные сообщения или сигналы, выдаваемые на индивидуальные входы.

Кодирующее устройство называют также *кодером* или *шифратором*, а декодирующее – декодером или дешифратором. Кодер формирует, а декодер разделяет кодовые комбинации по индивидуальным выходным цепям.

Наиболее широкое понятие, используемое при разработке технических средств кодирования и декодирования, – это кодопреобразователи (преобразователи кодов). *Преобразователем кодов* называется функциональный узел, преобразующий один код в другой. Такие функциональные узлы преобразуют, например, двоичный или двоично–десятичный код в десятичный, либо в код отображения информации на цифровом индикаторе, непомехозащищенный код в код с обнаружением или исправлением ошибок, а также производят обратное преобразование в приемном устройстве.

#### 3.1. Шифраторы кода $C_n^1$ в двоичный код

Функциональная схема этого шифратора представлена на рис. 3.1, а на рис. 3.2 показано его условное обозначение.

Шифратор работает следующим образом. Если ни одна из кнопок  $SB1...SB7$  не замкнута, то это означает, что передается сигнал 0, которому соответствует двоичное число 000. При замыкании одной из кнопок, соответствующий триггер устанавливается в 1, на инверсном выходе его будет 0, который поступает на соответствующие входы элементов 4И–НЕ (табл. 3.1), работающие как схемы ИЛИ по 0. В результате на выходе элементов  $DD8–DD10$  будет сформирован двоичный код, соответствующий номеру нажатой кнопки.

Таблица 3.1

Состояние элементов шифратора

Позиционное обозначение замкнутой кнопки		$SB1$	$SB2$	$SB3$	$SB4$	$SB5$	$SB6$	$SB7$
Вход	$DD8$	0111	1111	1011	1111	1101	1111	1110
	$DD9$	1111	0111	1011	1111	1111	1101	1110
	$DD10$	1111	1111	1111	0111	1011	1101	1110
Выход	$DD8$	1	0	1	0	1	0	1
	$DD9$	0	1	1	0	0	1	1
	$DD10$	0	0	0	1	1	1	1

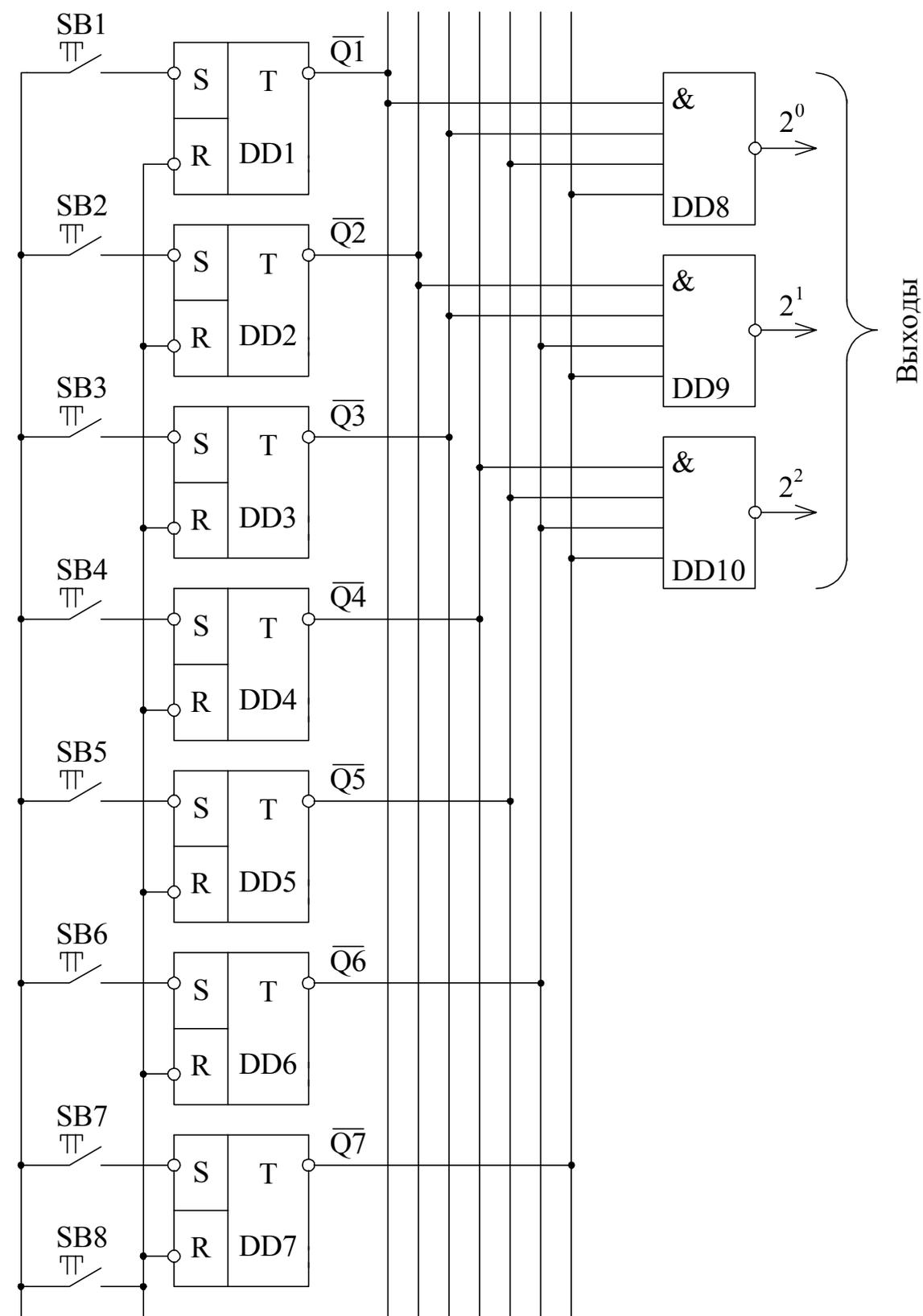


Рис. 3.1. Функциональная схема шифратора кода  $C_7^1$  в код 1-2-4

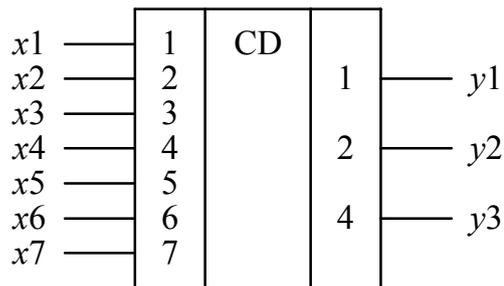


Рис. 3.2. Условное обозначение шифратора  $C_7^1$  в код 1-2-4

В системах ТУ–ТС при формировании адреса объекта и команды находят применение шифраторы кода  $C_n^1$  в двоичный код с использованием мультиплексоров (рис. 3.3).

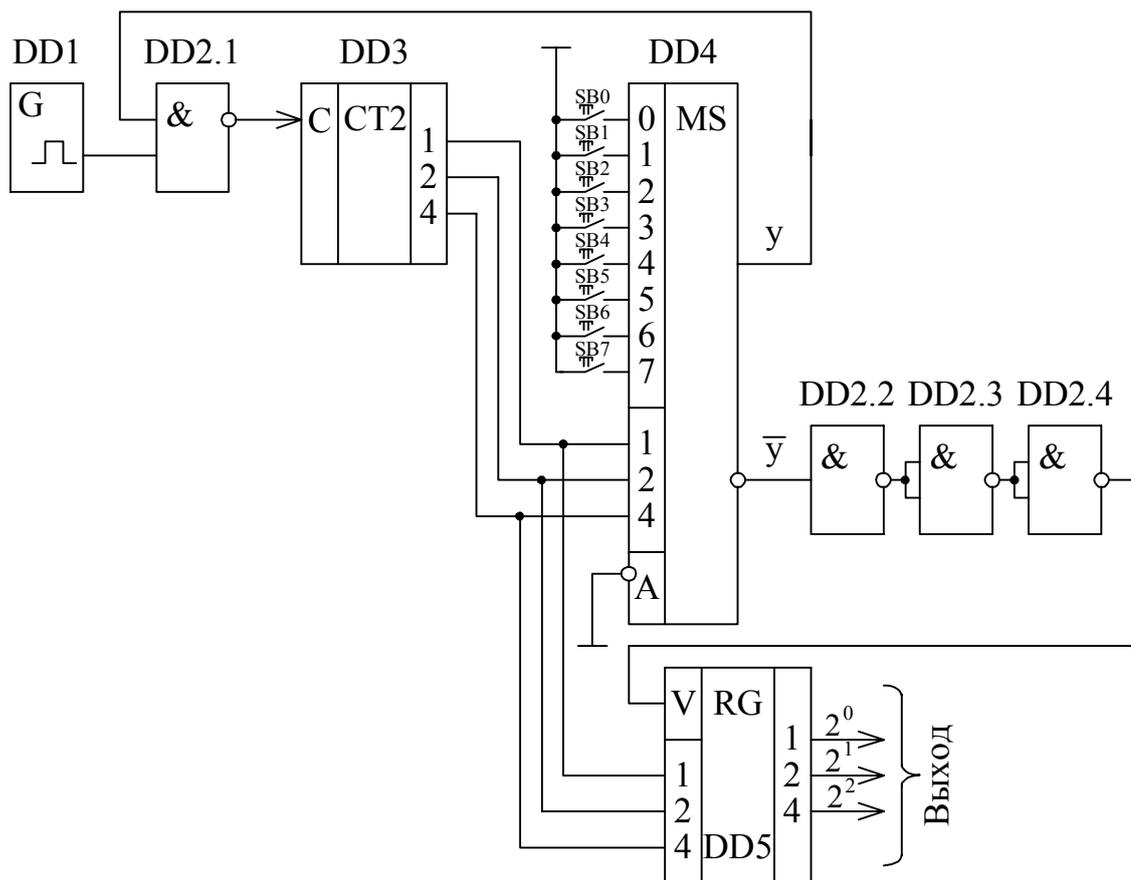


Рис. 3.3. Функциональная схема преобразователя десятичного кода в двоичный код

В исходном состоянии ни одна из кнопок  $SB0...SB7$  не нажата, при этом на входе  $Y$  мультиплексора  $DD4$  – логическая 1, которая поддерживает в открытом состоянии схему И–НЕ  $DD2.1$ . Импульсы от генератора  $DD1$  поступают на вход двоичного счетчика  $DD3$ , на выходе которого формируется код, пропорциональный числу импульсов, поступивших от генератора  $DD1$ . Теперь пусть нажата клавиша  $SB5$ . Тогда при поступлении на адресные входы мультиплексора  $DD4$  от  $DD3$  кода 101 на выходе  $Y$  появляется 0, который закрывает схему  $DD2.1$ , и прекращается поступление импульсов от  $DD1$  на счетчик  $DD3$ . Задержанный элементами  $DD2.2...DD2.4$  импульс с выхода  $\bar{Y}$  мультиплексора  $DD4$  дает разрешение на запись кода 101 с выхода  $DD3$  в регистр  $DD5$ . Данный код в регистре будет сохраняться до тех пор, пока в последующем не будет нажата любая из кнопок  $SB0...SB7$ .

Таким образом, практически произошло преобразование номера кнопки в двоичный избыточный код. При необходимости данный шифратор можно сделать на любое число входов путем параллельного соединения мультиплексоров и регистров.

В интегральном исполнении имеются шифраторы  $ИБ1$  и  $ИБ3$ . Более подробно остановимся на микросхеме  $ИБ1$ . Это приоритетный шифратор 8 в 3 (рис. 3.4), принимающий напряжение низкого уровня на один из восьми параллельных адресных входов  $\bar{I}1... \bar{I}8$ .

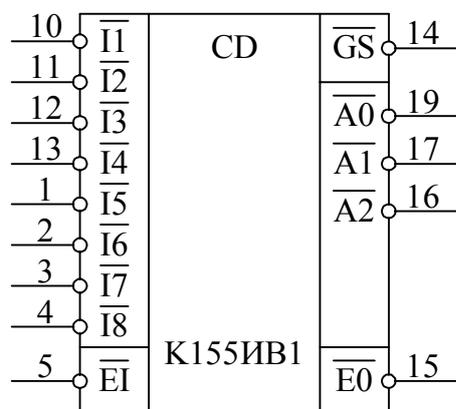


Рис. 3.4. Условное обозначение микросхемы  $ИБ1$

На трех выходах  $\bar{A}0... \bar{A}2$  появляется двоичный код, пропорциональный номеру входа, оказавшегося активным. Состояния дешифратора приведены в табл. 3.2.

## Состояния шифратора ИВ1

Входы									Выходы				
$\overline{E1}$	$\overline{I1}$	$\overline{I2}$	$\overline{I3}$	$\overline{I4}$	$\overline{I5}$	$\overline{I6}$	$\overline{I7}$	$\overline{I8}$	$\overline{GS}$	$\overline{A0}$	$\overline{A1}$	$\overline{A2}$	$\overline{E0}$
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	x	x	x	x	x	x	x	0	0	0	0	0	1
0	x	x	x	x	x	x	0	1	0	1	0	0	1
0	x	x	x	x	x	0	1	1	0	0	1	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	0	1	1	1	1	0	0	0	1	1
0	x	x	0	1	1	1	1	1	0	1	0	1	1
0	x	0	1	1	1	1	1	1	0	0	1	1	1
0	0	1	1	1	1	1	1	1	0	1	1	1	1

Приоритет в том случае, если несколько входов получили активные уровни, будет иметь «старший» среди них по номеру. Высший приоритет у входа  $\overline{I8}$ .

Используя совместно выход  $\overline{E0}$  и разрешающий вход  $\overline{E1}$ , можно строить многоразрядные приоритетные шифраторы.

### 3.2. Дешифратор двоичного кода в десятичный код

Простейший дешифратор, выполненный на элементах И–НЕ, НЕ (рис. 3.5), называется линейным. Когда на входы подается комбинация 000, с выхода элемента  $DD1.4$  должен быть снят сигнал 1, а с остальных выходов – сигналы 0. Для этого на элемент 3И–НЕ  $DD3.1$  сигналы поступают не непосредственно со входов, а через инверторы  $DD1.1...DD1.3$ , в которых нули преобразуются в единицы.

Три сигнала “1” на входе  $DD3.1$  дают на его выходе сигнал 0, который инвертируется элементом  $DD1.4$ , и в результате получаем сигнал на выходе  $DD1.4$ , равный “1”. На выходах всех остальных элементов будут нули, так как на один из входов элементов  $DD3.2...DD5.2$  подаются сигналы 0, минуя инверторы. Порядок формирования сигналов на выходе при других сигналах на входе приведен в табл. 3.3.

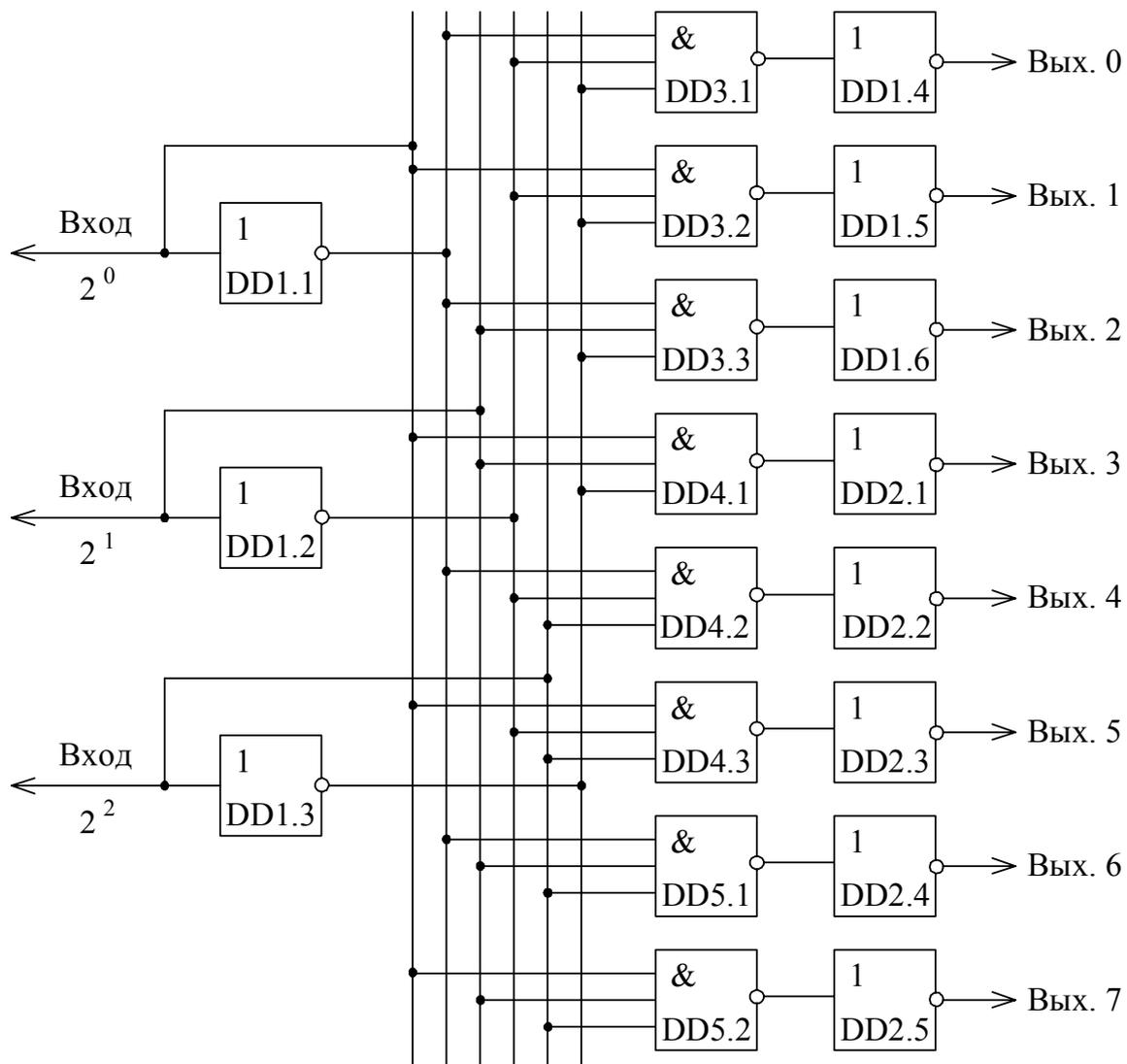


Рис. 3.5. Функциональная схема дешифратора двоичного кода в десятичный код

Таблица 3.3

Состояние дешифратора кода 4–2–1 в десятичный код

Входы			Выходы							
$2^0$	$2^1$	$2^2$	DD1.4	DD1.5	DD1.6	DD2.1	DD2.2	DD2.3	DD2.4	DD2.5
0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

В интегральном исполнении разработана большая гамма двоично-десятичных дешифраторов, допускающих параллельное соединение. Рассмотрим параллельную работу двоично-десятичных дешифраторов на базе микросхемы ИД7.

Логическая структура, цоколевка и условное обозначение дешифратора представлены на рис. 3.6.

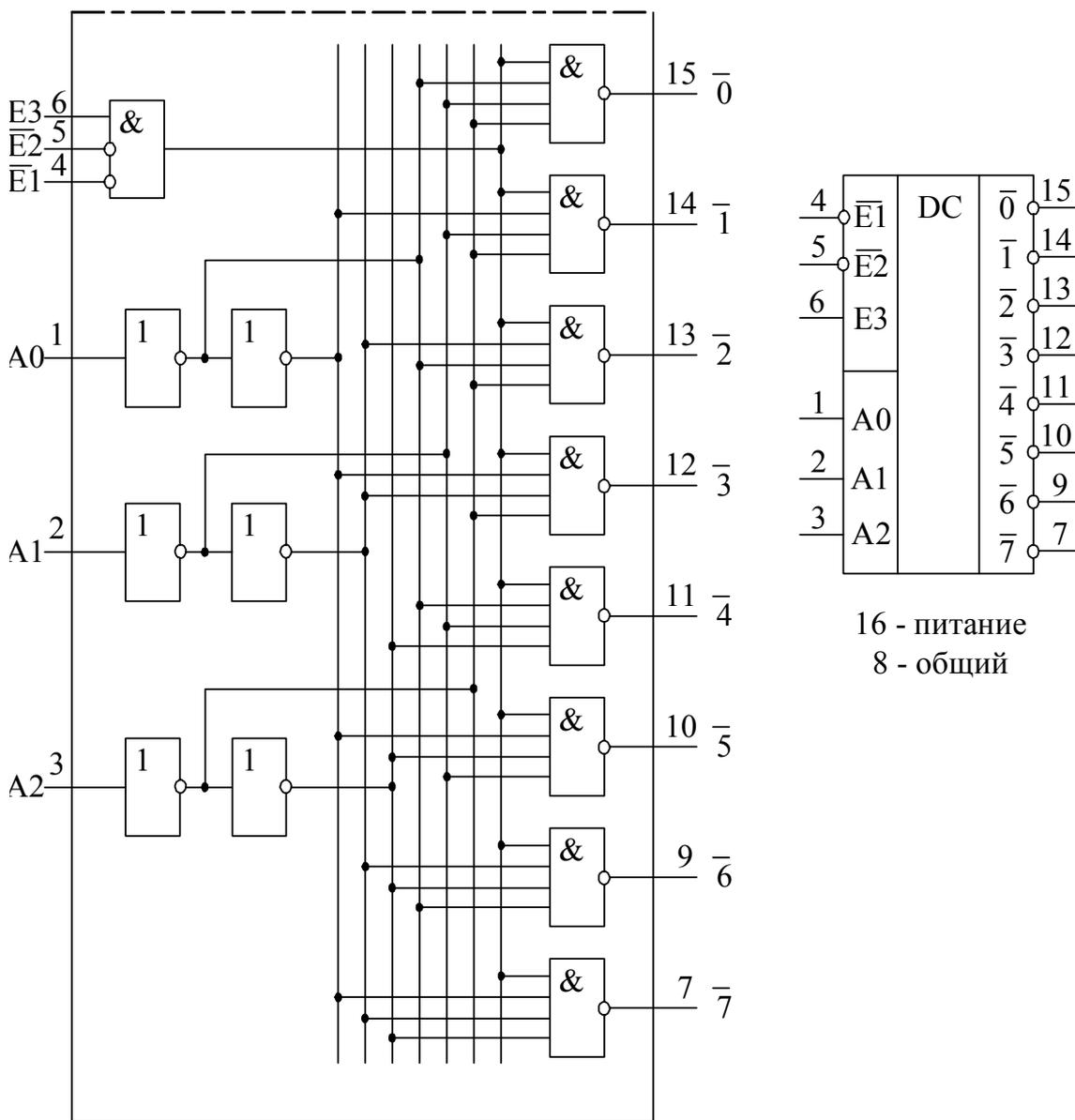


Рис. 3.6. Структура, условное обозначение и цоколевка микросхемы ИД7

Дешифрация происходит тогда, когда на входах  $\overline{E1}$  и  $\overline{E2}$  действует напряжение низкого уровня, а на входе E3 – высокого. При других сочетаниях уровней на входах разрешения  $E_i$  на всех выходах будет напряжение высокого уровня.

Состояния ИД7 приведены в табл. 3.4.

Состояния дешифратора ИД7

Входы						Выходы							
$\overline{E1}$	$\overline{E2}$	E3	A0	A1	A2	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	0	x	x	x	1	1	1	1	1	1	1	1
0	0	1	0	0	0	<b>0</b>	1	1	1	1	1	1	1
0	0	1	1	0	0	1	<b>0</b>	1	1	1	1	1	1
0	0	1	0	1	0	1	1	<b>0</b>	1	1	1	1	1
0	0	1	1	1	0	1	1	1	<b>0</b>	1	1	1	1
0	0	1	0	0	1	1	1	1	1	<b>0</b>	1	1	1
0	0	1	1	0	1	1	1	1	1	1	<b>0</b>	1	1
0	0	1	0	1	1	1	1	1	1	1	1	<b>0</b>	1
0	0	1	1	1	1	1	1	1	1	1	1	1	<b>0</b>

Наличие трехвходового логического элемента разрешения позволяет соединить параллельно несколько дешифраторов с целью увеличения количества выходов.

На рис. 3.7 показан дешифратор на 32 выхода. Для его реализации потребовалось 4 дешифратора ИД7 и дополнительный инвертор DD5.

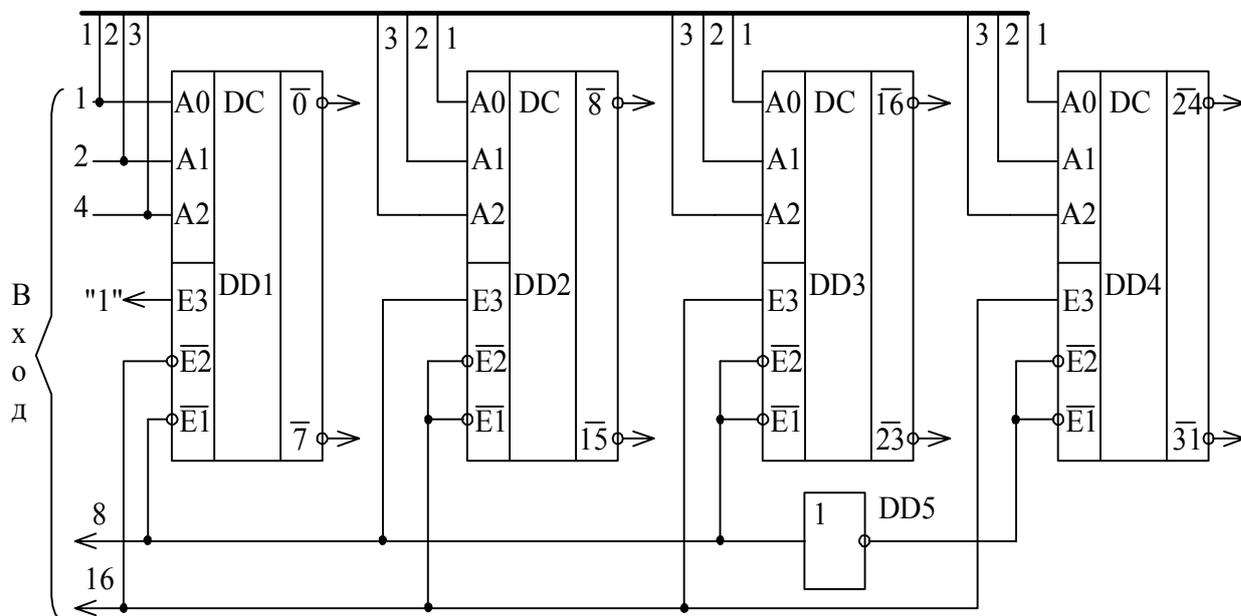


Рис. 3.7. Параллельное соединение ИД7

На входы  $A_0A_1A_2$  дешифраторов  $DD1...DD4$  параллельно подаются три младших разряда преобразуемого кода, а два старших разряда  $2^3$  и  $2^4$  определяют очередность работы каждого дешифратора, т.е. являются второй ступенью дешифрации. Состояния на выходе элементов разрешения приведены в табл. 3.5.

Таблица 3.5

Состояния элементов разрешения дешифраторов

Вход		Дешифратор							
		DD1				DD2			
		Входы			Выход	Входы			Выход
$\overline{E1}$	$\overline{E2}$	E3	$\overline{E1}$	$\overline{E2}$		E3			
23 = 8	24 = 16								
0	0	0	0	1	1	0	0	0	0
1	0	1	0	1	0	0	0	1	1
0	1	0	1	1	0	1	1	0	0
1	1	1	1	1	0	1	1	1	0

Вход		Дешифратор							
		DD3				DD4			
		Входы			Выход	Входы			Выход
$\overline{E1}$	$\overline{E2}$	E3	$\overline{E1}$	$\overline{E2}$		E3			
23 = 8	24 = 16								
0	0	0	0	0	0	1	1	0	0
1	0	1	1	0	0	0	0	0	0
0	1	0	0	1	1	1	1	1	0
1	1	1	1	1	0	0	0	1	1

### 3.3. Дешифратор двоично–десятичного кода в десятичный

Так как в двоично–десятичном коде, например, число 97 записывается как 1001 0111, то для его расшифровки требуется два дешифратора: первый для преобразования десятков, второй – единиц. Для дешифрации трехзначного числа нужны три дешифратора и т.д. При этом каждый дешифратор должен преобразовывать кодовые комбинации от 0000 до 1001 в числа 0...9 соответственно (рис. 3.8). Принцип работы поясним на примере дешифратора ИД6, который преобразует двоичный код, поступающий на входы  $A0...A3$ , в сигнал низкого уровня, появляющийся на десятичном выходе  $\overline{0}...9$ .

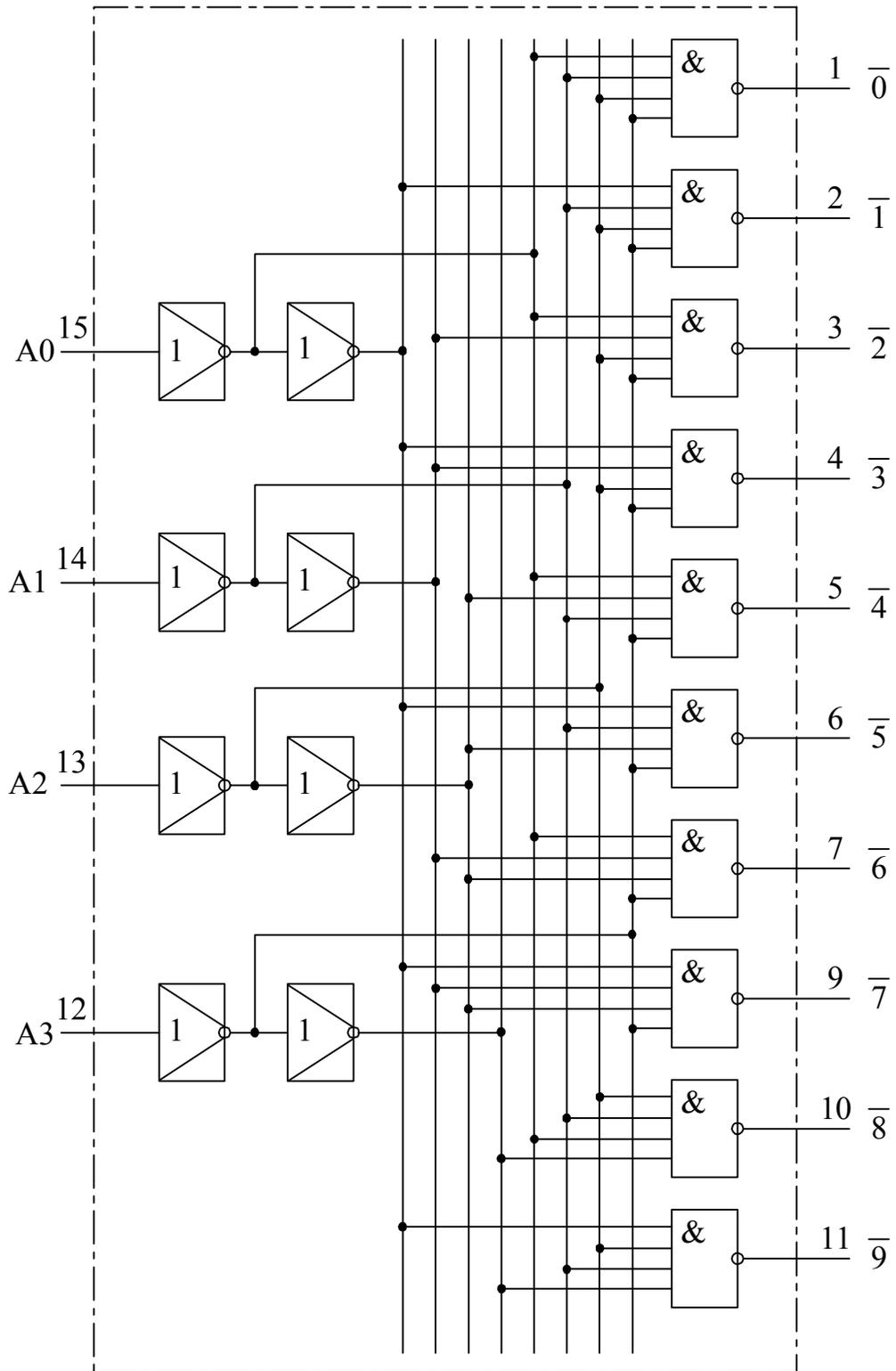


Рис. 3.8. Структура и цоколевка ИД6

Состояния дешифратора приведены в табл. 3.6.

Состояния дешифратора ИД6

Входы				Выходы									
$A_3$	$A_2$	$A_1$	$A_0$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$	$\bar{8}$	$\bar{9}$
0	0	0	0	<b>0</b>	1	1	1	1	1	1	1	1	1
0	0	0	1	1	<b>0</b>	1	1	1	1	1	1	1	1
0	0	1	0	1	1	<b>0</b>	1	1	1	1	1	1	1
0	0	1	1	1	1	1	<b>0</b>	1	1	1	1	1	1
0	1	0	0	1	1	1	1	<b>0</b>	1	1	1	1	1
0	1	0	1	1	1	1	1	1	<b>0</b>	1	1	1	1
0	1	1	0	1	1	1	1	1	1	<b>0</b>	1	1	1
0	1	1	1	1	1	1	1	1	1	1	<b>0</b>	1	1
1	0	0	0	1	1	1	1	1	1	1	1	<b>0</b>	1
1	0	0	1	1	1	1	1	1	1	1	1	1	<b>0</b>
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Если десятичный эквивалент входного кода превышает 9, то на всех выходах  $\bar{0} \dots \bar{9}$  появляется напряжение высокого уровня.

### 3.4. Преобразователи двоичного кода в двоично–десятичный код и обратно

Наибольшее применение нашли два типа преобразователей – на базе пересчетных схем и на базе интегральных микросхем *ПП6* и *ПП7*.

На рис. 3.9 приведена функциональная электрическая схема восьмиразрядного преобразователя двоичного кода в двоично–десятичный первого типа.

По сигналу с устройства управления (УУ) преобразуемый восьмиразрядный двоичный код записывается в двоичный вычитающий счетчик *DD1*, и этим же сигналом, задержанным на время окончания переходных процессов во входных цепях, устанавливается в 1 триггер *DD2*. Единичный сигнал с прямого выхода триггера *DD2* открывает схему И *DD4*, и импульс от генератора *DD6* поступает на суммирующий вход двоично–десятичного счетчика *DD7* и на вычитающий вход двоичного счетчика *DD1*. В момент времени, когда в двоичном счетчике *DD1* будет нулевая комбинация, сигналом с выхода  $\leq 0$  счетчика *DD1* триггер *DD2* устанавливается в исходное положение. Закрывается схема И

*DD4*, и прекращается поступление импульсов в двоично–десятичный счетчик *DD7*. Таким образом, в счетчике *DD7* будет зафиксирован в двоично–десятичном коде эквивалент двоичного кода, поступившего на вход счетчика *DD1*.

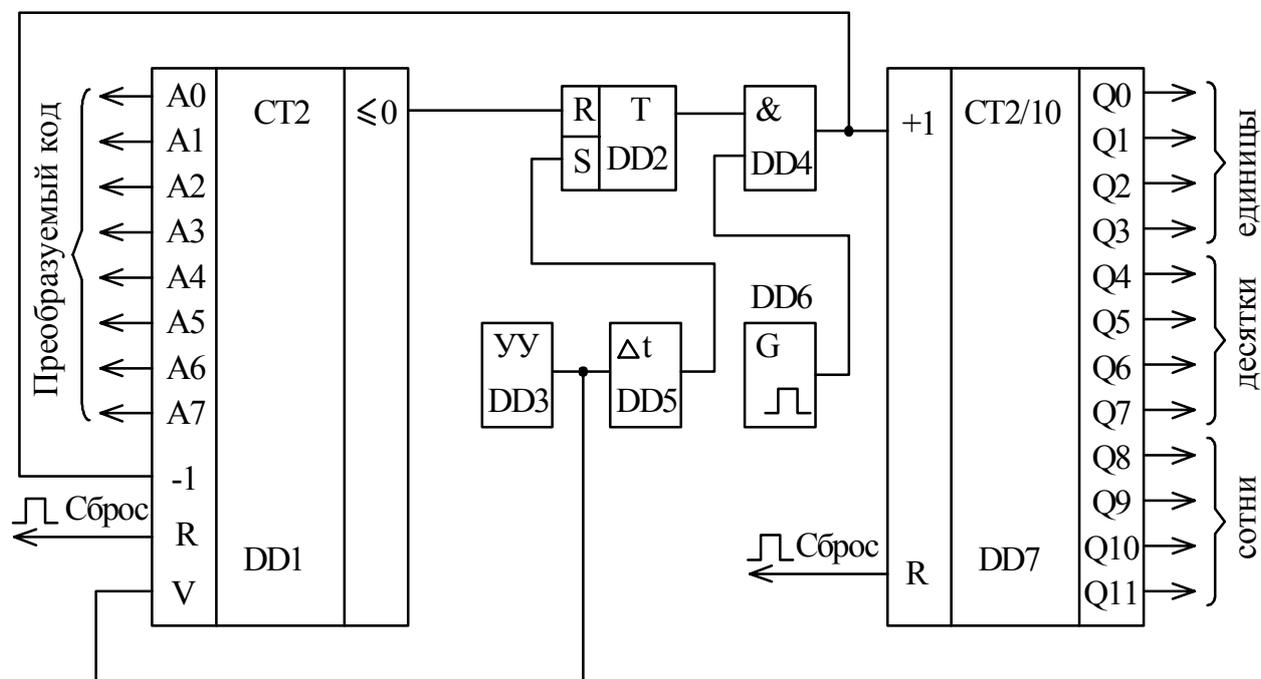


Рис. 3.9. Функциональная электрическая схема преобразования двоичного кода в двоично-десятичный

Данная схема является одной из самых распространенных схем, позволяющих сравнительно просто производить преобразование любых *n*–разрядных двоичных последовательностей. В качестве двоичного счетчика *DD1* можно применить микросхемы *ИЕ6*, а в качестве двоично–десятичного счетчика *DD7* микросхемы *ИЕ7*.

Для преобразования двоично–десятичного кода в двоичный код можно воспользоваться преобразователем, схема которого приведена на рис. 3.9, но поменять местами счетчики *DD1* и *DD7*.

Преобразователи двоично–десятичного кода в двоичный и двоичного кода в двоично–десятичный код можно построить на микросхемах *ПП6* (рис. 3.10) и *ПП7* (рис. 3.11) соответственно. Основой таких преобразователей является запоминающая матрица емкостью 256 бит. Ячейки матрицы соединены в соответствии с программой преобразования. Матрицей управляет дешифратор с 5 входами и 32 выходами.

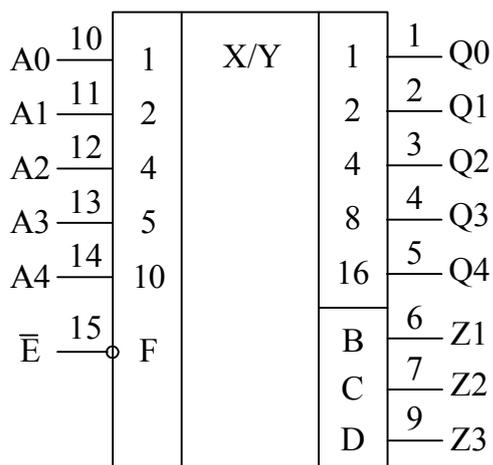


Рис. 3.10. Условное обозначение и цоколевка микросхемы ПР6

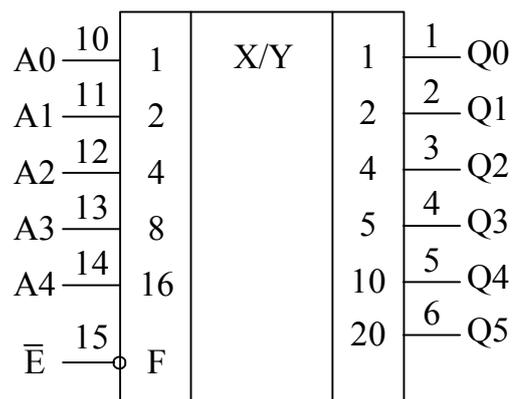


Рис. 3.11. Условное обозначение и цоколевка микросхемы ПР7

На входы  $A_0...A_4$  ПР6 подается двоично–десятичный код с весом 1–2–4–5–10. Вход  $\bar{E}$  разрешает преобразование, если на него подан низкий уровень напряжения. Когда на вход  $\bar{E}$  подано напряжение высокого уровня, то преобразование запрещено, а на выходах  $Q_i$  появляется напряжение высокого уровня. Состояния микросхем ПР6 и ПР7 представлены в табл. 3.7 и 3.8 соответственно.

Таблица 3.7

Состояния логических уровней при преобразовании двоично–десятичных слов в двоичные в ПР6

№ слова	Вход						Выход				
	A4	A3	A2	A1	A0	$\bar{E}$	Q4	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	1
2	0	0	0	1	0	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	0	1	1
4	0	0	1	0	0	0	0	0	1	0	0
5	0	1	0	0	0	0	0	0	1	0	1
6	0	1	0	0	1	0	0	0	1	1	0
7	0	1	0	1	0	0	0	0	1	1	1
8	0	1	0	1	1	0	0	1	0	0	0
9	0	1	1	0	0	0	0	1	0	0	1
10	1	0	0	0	0	0	0	1	0	1	0
11	1	0	0	0	1	0	0	1	0	1	1
12	1	0	0	1	0	0	0	1	1	0	0

№ слова	Вход						Выход				
	A4	A3	A2	A1	A0	$\bar{E}$	Q4	Q3	Q2	Q1	Q0
13	1	0	0	1	1	0	0	1	1	0	1
14	1	0	1	0	0	0	0	1	1	1	0
15	1	1	0	0	0	0	0	1	1	1	1
16	1	1	0	0	1	0	1	0	0	0	0
17	1	1	0	1	0	0	1	0	0	0	1
18	1	1	0	1	1	0	1	0	0	1	0
19	1	1	1	0	0	0	1	0	0	1	1
Любое	x	x	x	x	x	1	1	1	1	1	1

Интегральная микросхема ПР6 имеет еще два применения при использовании выходов Z1, Z2, Z3: при A4 = 0 производится преобразование двоично-десятичного числа X = (A3, A2, A1, A0) в дополнение W до числа 9 по правилу: W = 9 - X = (Z3, Z2, A1, Z1) (рис. 3.12); при A4 = 1 – преобразование двоично-десятичного числа X=(A3, A2, A1, A0) в дополнение W1 до числа 10 (рис. 3.13) по правилу:

$$W1 = (Z3, Z2, Z1, A0) = \begin{cases} 10 - X, & \text{если } 1 \leq X \leq 9, \\ 0, & \text{если } X = 0. \end{cases}$$

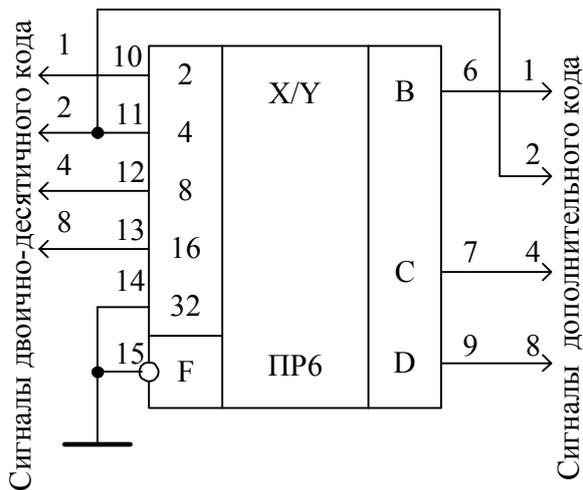


Рис. 3.12. Преобразование в дополнение до числа 9

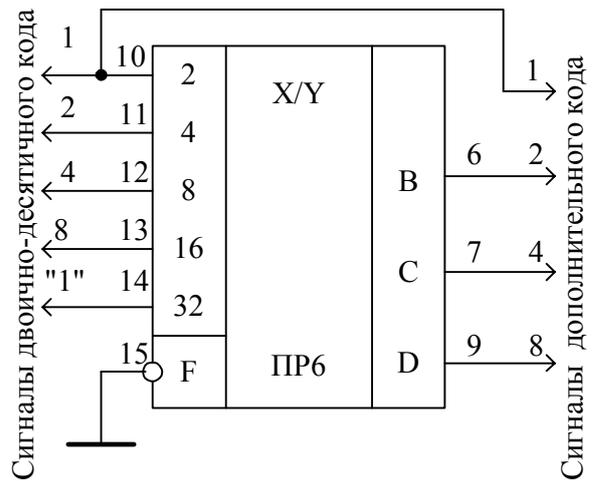


Рис. 3.13. Преобразование в дополнение до числа 10

Таблица 3.8

Состояния логических уровней при преобразовании двоичного  
кода в двоично–десятичный в ПП7

№ слова	Вход						Выход					
	A4	A3	A2	A1	A0	$\bar{E}$	Q5	Q4	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	1
2	0	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	0	0	1	1
4	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	0	1	0	0	0
6	0	0	1	1	0	0	0	0	1	0	0	1
7	0	0	1	1	1	0	0	0	1	0	1	0
8	0	1	0	0	0	0	0	0	1	0	1	1
9	0	1	0	0	1	0	0	0	1	1	0	0
10	0	1	0	1	0	0	0	1	0	0	0	0
11	0	1	0	1	1	0	0	1	0	0	0	1
12	0	1	1	0	0	0	0	1	0	0	1	0
13	0	1	1	0	1	0	0	1	0	0	1	1
14	0	1	1	1	0	0	0	1	0	1	0	0
15	0	1	1	1	1	0	0	1	1	0	0	0
16	1	0	0	0	0	0	0	1	1	0	0	1
17	1	0	0	0	1	0	0	1	1	0	1	0
18	1	0	0	1	0	0	0	1	1	0	1	1
19	1	0	0	1	1	0	0	1	1	1	0	0
20	1	0	1	0	0	0	1	0	0	0	0	0
21	1	0	1	0	1	0	1	0	0	0	0	1
22	1	0	1	1	0	0	1	0	0	0	1	0
23	1	0	1	1	1	0	1	0	0	0	1	1
24	1	1	0	0	0	0	1	0	0	1	0	0
25	1	1	0	0	1	0	1	0	1	0	0	0
26	1	1	0	1	0	0	1	0	1	0	0	1
27	1	1	0	1	1	0	1	0	1	0	1	0
28	1	1	1	0	0	0	1	0	1	0	1	1
29	1	1	1	0	1	0	1	0	1	1	0	0
30	1	1	1	1	0	0	1	1	0	0	0	0
31	1	1	1	1	1	0	1	1	0	0	0	1
Любое	x	x	x	x	x	1	1	1	1	1	1	1

Как правило, разрядности одной микросхемы в большинстве случаев недостаточно, поэтому преобразователи соединяют каскадно. Соединение микросхем *K155 ПП6* для обработки сигналов двоично–десятичных чисел 0–99 показано на рис. 3.14.

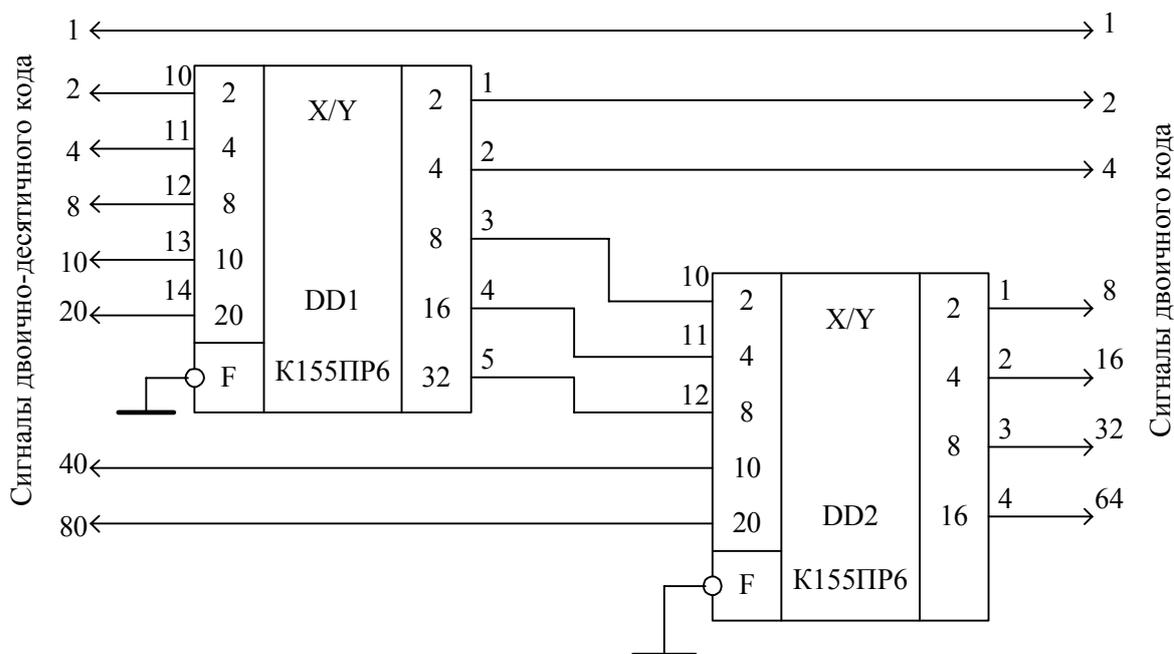


Рис. 3.14. Преобразование двоично-десятичных чисел 0 - 99 в двоичный код

На рис. 3.15 приведена схема преобразователя на микросхемах K155PP7 для двоичных чисел 0–255.

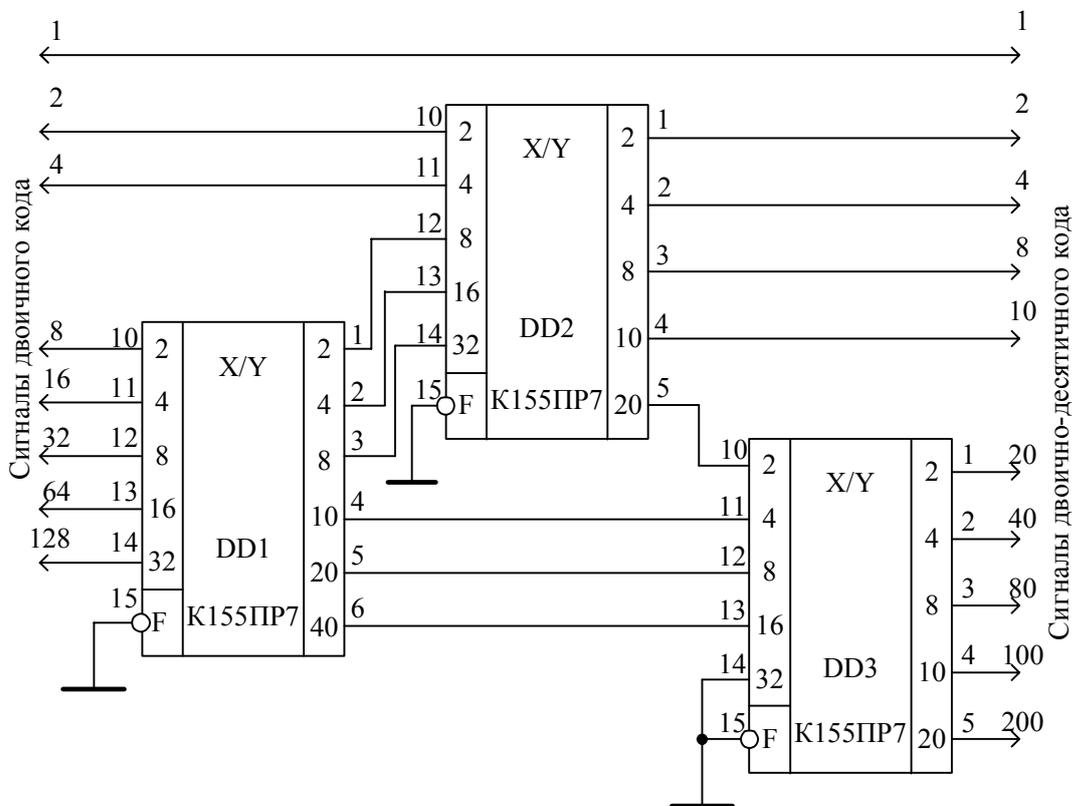


Рис. 3.15. Преобразование двоичных чисел 0 - 255 в двоично-десятичный код

### 3.5. Преобразователь двоичного кода 8–4–2–1 в самодополняющийся двоично–десятичный код 2–4–2–1

Представим числа в двоичном коде как  $y_4y_3y_2y_1$ , а числа в самодополняющемся коде как  $x_4x_3x_2x_1$ . Тогда разряды  $x_i$  через  $y_i$  будут определяться из выражений:

$$x_1 = y_1;$$

$$x_2 = \bar{y}_4\bar{y}_3y_2 + y_4\bar{y}_3\bar{y}_2 + \bar{y}_4y_3\bar{y}_2y_1;$$

$$x_3 = \bar{y}_4y_3y_2 + \bar{y}_4y_3\bar{y}_1 + y_4\bar{y}_3\bar{y}_2;$$

$$x_4 = \bar{y}_4y_3y_2 + \bar{y}_4y_3y_1 + y_4\bar{y}_3\bar{y}_2.$$

(3.1)

Функциональная электрическая схема преобразователя кода 8–4–2–1 в код 2–4–2–1, построенного по выражению (3.1), приведена на рис. 3.16.

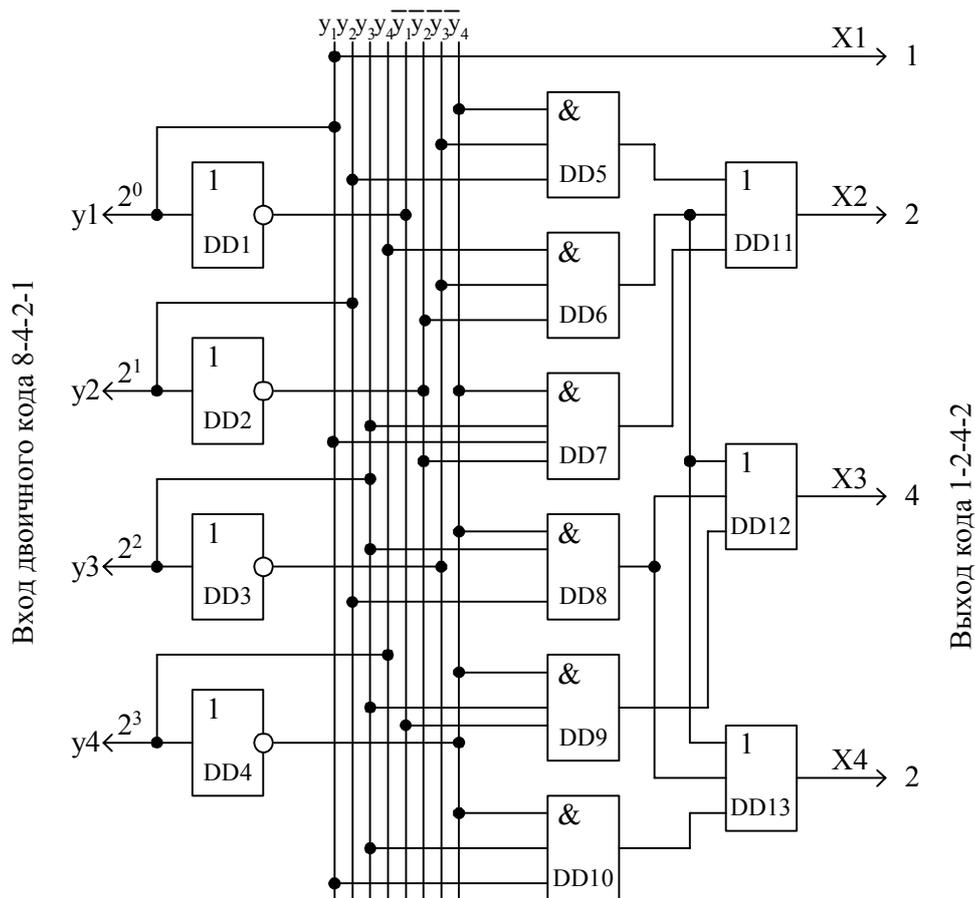


Рис. 3.16. Преобразователь кода 8-4-2-1 в код 2-4-2-1

Табл. 3.9, поясняющая работу схемы, составлена таким образом. В верхних четырех строках записан преобразуемый двоичный код, а в самой нижней – десятичные числа. Во второй – пятой строках снизу зафиксированы комбинации кода 2–4–2–1, которые образуются на выходах схем  $DD11\dots DD13$  и  $X1$ . В остальных строках таблицы записаны единицы или нули, которые образуются на выходах промежуточных элементов.

Таблица 3.9

Преобразование кода 8–4–2–1 в код 2–4–2–1

<b>СОСТОЯНИЯ</b>	<b>Входов</b>	$y_1(2^0)$	0	1	0	1	0	1	0	1	0	1	
		$y_2(2^1)$	0	0	1	1	0	0	1	1	0	0	
		$y_3(2^2)$	0	0	0	0	1	1	1	1	0	0	
		$y_4(2^3)$	0	0	0	0	0	0	0	0	0	1	1
	<b>Промежуточных элементов</b>	$DD1$	1	0	1	0	1	0	1	0	1	0	
		$DD2$	1	1	0	0	1	1	0	0	1	1	
		$DD3$	1	1	1	1	0	0	0	0	0	1	1
		$DD4$	1	1	1	1	1	1	1	1	1	0	0
		$DD5$	0	0	1	1	0	0	0	0	0	0	0
		$DD6$	0	0	0	0	0	0	0	0	0	1	1
		$DD7$	0	0	0	0	0	0	1	0	0	0	0
		$DD8$	0	0	0	0	0	0	0	1	1	0	0
		$DD9$	0	0	0	0	1	0	1	0	0	0	0
		$DD10$	0	0	0	0	0	1	0	1	0	0	0
	<b>Выходов</b>	$x_1(1)$	0	1	0	1	0	1	0	1	0	1	
		$x_2(2)$	0	0	1	1	0	1	0	0	1	1	
		$x_3(4)$	0	0	0	0	1	0	1	1	1	1	
$x_4(2)$		0	0	0	0	0	1	1	1	1	1		
<b>Десятичное число</b>		0	1	2	3	4	5	6	7	8	9		

### 3.6. Преобразователь самодополняющего двоично–десятичного кода 2–4–2–1 в двоичный код 8–4–2–1

На основании анализа столбцов 2 и 3 табл. 1.3 и введенных обозначений разрядов в подразделе 3.5, разряды кода 2–4–2–1 связаны с разрядами кода 8–4–2–1 следующими выражениями:

$$\begin{aligned}
 y_1 &= x_1; \\
 y_2 &= \bar{x}_4 \bar{x}_3 x_2 + x_4 x_3 \bar{x}_2 \\
 y_3 &= x_4 \bar{x}_3 x_2 x_1 + x_4 x_3 \bar{x}_2 + x_3 \bar{x}_2 \bar{x}_1; \\
 y_4 &= x_4 x_3 x_2.
 \end{aligned}
 \tag{3.2}$$

Функциональная электрическая схема преобразователя, построенная на основании выражений (3.2), приведена на рис. 3.17, а принцип работы поясняется табл. 3.10.

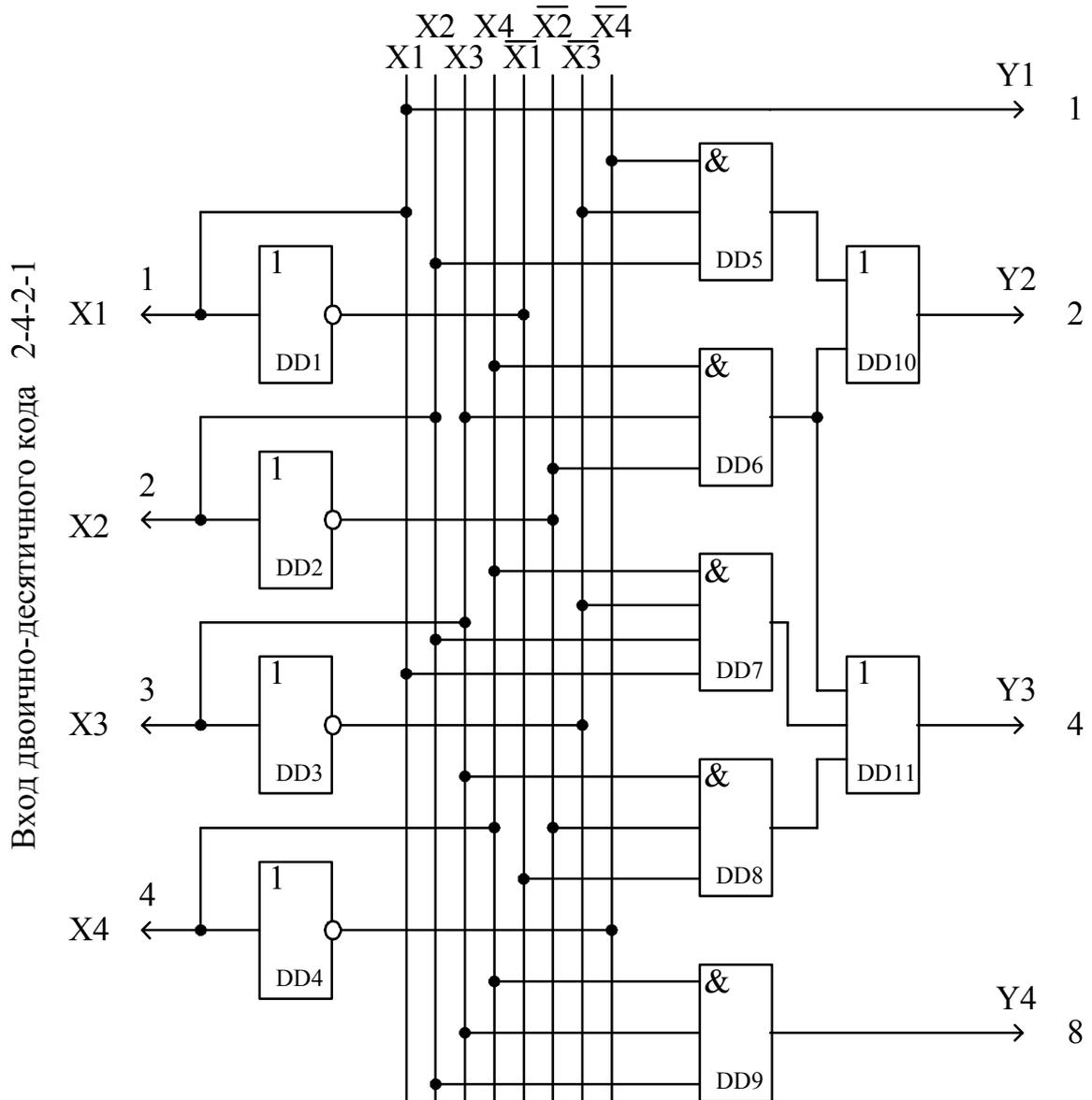


Рис. 3.17. Схема преобразователя кода 2-4-2-1 в код 8-4-2-1

<b>СОСТОЯНИЯ</b>	<b>Входов</b>	$x_1(1)$	0	1	0	1	0	1	0	1	0	1
		$x_2(2)$	0	0	1	1	0	1	0	0	1	1
		$x_3(4)$	0	0	0	0	1	0	1	1	1	1
		$x_4(2)$	0	0	0	0	0	1	1	1	1	1
	<b>Промежуточных элементов</b>	$DD1$	1	0	1	0	1	0	1	0	1	0
		$DD2$	1	1	0	0	1	0	1	1	0	0
		$DD3$	1	1	1	1	0	1	0	0	0	0
		$DD4$	1	1	1	1	1	0	0	0	0	0
		$DD5$	0	0	1	1	0	0	0	0	0	0
		$DD6$	0	0	0	0	0	0	1	1	0	0
		$DD7$	0	0	0	0	0	1	0	0	0	0
		$DD8$	0	0	0	0	1	0	1	0	0	0
		$DD9$	0	0	0	0	0	0	0	0	1	1
	<b>Выходов</b>	$y_1(1)$	0	1	0	1	0	1	0	1	0	1
		$y_2(2)$	0	0	1	1	0	0	1	1	0	0
		$y_3(4)$	0	0	0	0	1	1	1	1	0	0
		$y_4(8)$	0	0	0	0	0	0	0	0	1	1
<b>Десятичное число</b>		0	1	2	3	4	5	6	7	8	9	

### 3.7. Преобразователь кода Грея в двоичный код и обратно

Как указано в подразд. 1.3.5, значение каждого последующего разряда двоичного числа находится путем сложения по модулю 2 этого же разряда в коде Грея с предыдущим (1.8), т.е.

$$a_i = \sum_{j=i}^i b_j (\text{mod} 2),$$

а для определения  $i$ -го разряда в коде Грея необходимо сложить по модулю 2 значение преобразуемого разряда двоичного кода со значением предыдущего разряда этого же кода (1.9), т.е.

$$b_i = a_{i+1} \oplus a_i.$$

Оба эти выражения определяют алгоритмы преобразования кода Грея в двоичный код и обратно.

На рис. 3.18 показана функциональная схема преобразователя четырех разрядного кода Грея в четырех разрядный двоичный код, а на рис. 3.19 схема обратного преобразования. Соответствие кода Грея двоичному и наоборот показано в табл. 3.11.

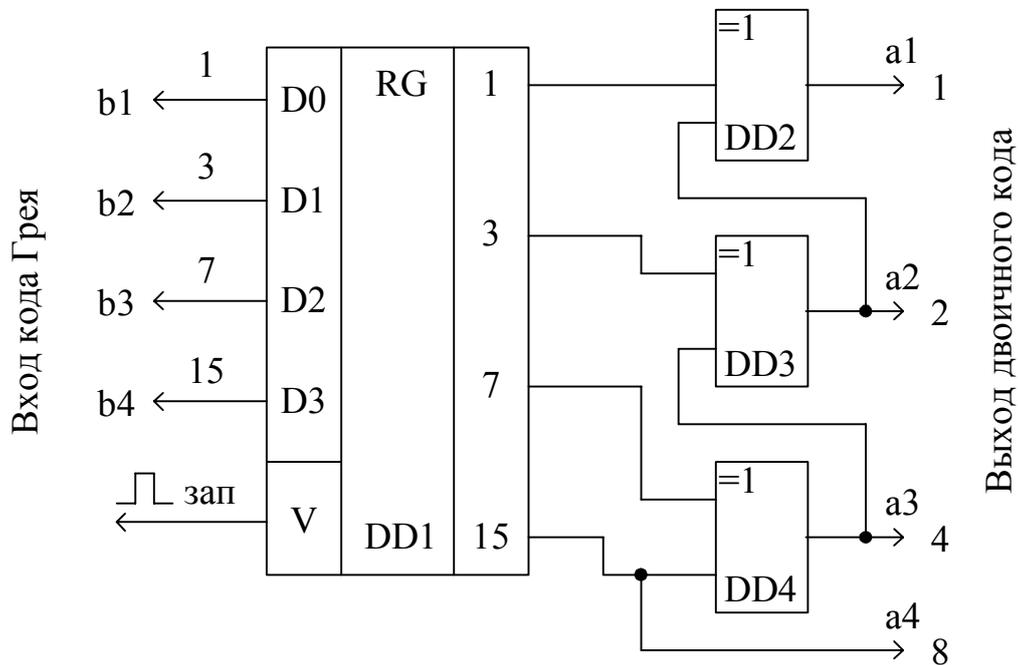


Рис. 3.18. Преобразователь кода Грея в двоичный

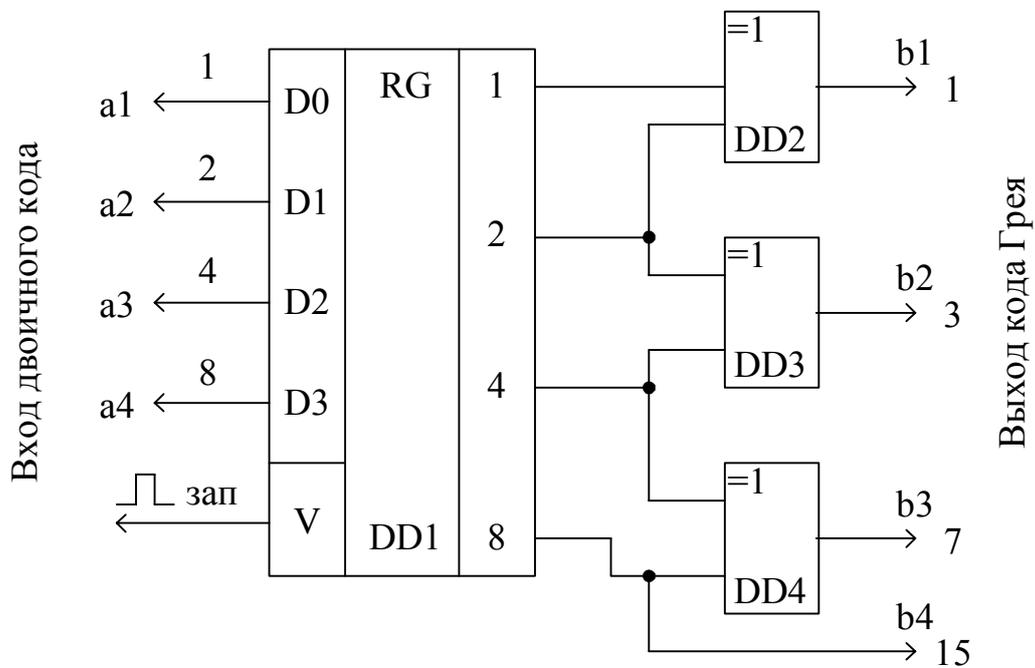


Рис. 3.19. Преобразователь двоичного кода в код Грея

Соответствие четырехразрядного кода Грея двоичному коду 8–4–2–1

Вход	$a_1$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	Двоичный код
	$a_2$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
	$a_3$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
	$a_4$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
Выход	$b_1$	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	Код Грея
	$b_2$	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	
	$b_3$	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	
	$b_4$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
Десятичный код	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		

Преобразование последовательного кода Грея в двоичный можно осуществить с помощью преобразователя, схема которого приведена на рис. 3.20. Кодовые комбинации на вход триггера *DDI* поступают начиная со старшего разряда. Триггер осуществляет операцию суммирования по модулю 2. Принцип работы наглядно поясняется временными диаграммами, приведенными на рис. 3.21.

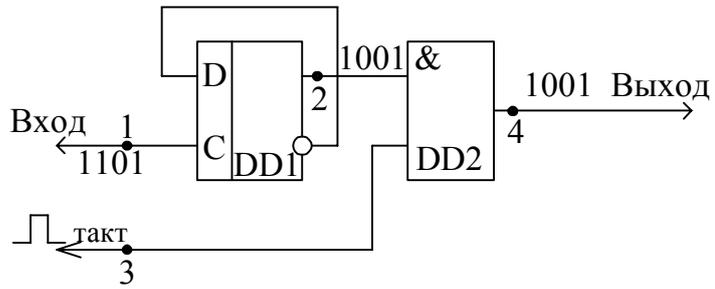


Рис. 3.20. Преобразователь последовательного кода Грея в двоичный

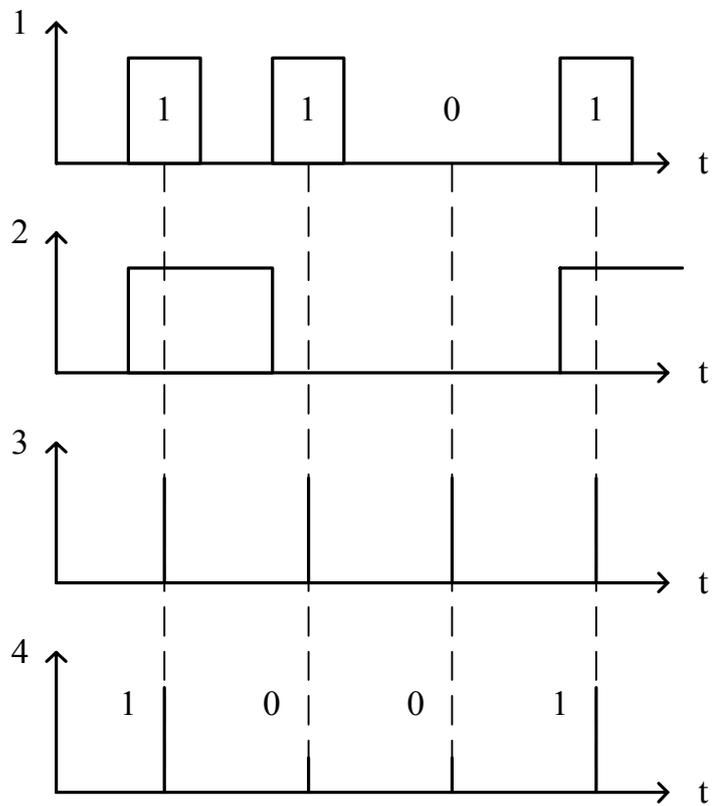


Рис. 3.21. Временные диаграммы работы преобразователя кода Грея в двоичный

### 3.8. Технические средства кодирования и декодирования эффективных кодов

Из материала подразд. 1.4.3 следует, что в общем случае кодер источника должен содержать следующие блоки:

- 1) устройство декорреляции, ставящее в соответствие исходной последовательности знаков другую последовательность знаков;
- 2) буферное устройство, выравнивающее плотность символов перед их поступлением в линию связи.

Декодер источника соответственно должен содержать:

- 1) устройство преобразования последовательности кодовых комбинаций в последовательность знаков;
- 2) буферное устройство, выравнивающее интервалы между знаками;
- 3) устройство рекорреляции, осуществляющее операцию, обратную декорреляции.

Рассмотрим кодер и декодер применительно к коду Хаффмана, приведенному в табл. 1.6.

Схема кодера источника приведена на рис. 3.22. В ней можно выделить основной шифратор на элементах  $DD10...DD13$  с регистром  $DD16$ , и вспомогательную схему управления считыванием информации, содержащую шифратор на элементах  $DD14, DD15$  с регистром  $DD17$ . Основной шифратор обеспечивает запись в регистр  $DD16$  кодовой комбинации, соответствующей сообщению  $x_i$  (см. табл. 1.6). Вспомогательный шифратор обеспечивает запись единицы в такую ячейку регистра  $DD17$ , чтобы длина кодовой комбинации в линии связи соответствовала сообщению  $x_i$ . Пусть необходимо закодировать сообщение  $X8$ . При подаче с устройства управления (УУ) логической единицы открывается схема И  $DD8$ , остальные схемы И будут закрыты, так как на других выходах источника информации сообщения будут отсутствовать. В результате на выходах микросхем  $DD10, DD11$  и  $DD12$  появятся единицы, которые поступят на входы  $D1, D2$  и  $D3$  регистра  $DD16$ , а на входы  $D4$  и  $D0$  соответственно с  $DD13$  и шины 8 поступят 0 и 1. Сигналом 1 с выхода УУ кодовое сообщение на входах  $D0...D4$  будет записано в регистр  $DD16$ . Таким образом, в регистр будет записана кодовая комбинация 01111, что соответствует кодовой комбинации сообщения  $X8$  табл. 1.6. Кроме того, в ячейку  $D0$ , пятую с конца регистра  $DD17$  с выхода схемы ИЛИ  $DD14$ , будет записана 1, что соответствует длине ( $\mu_i = 5$ ) кодовой комбинации, записанной в  $DD16$ . После этого тактовыми импульсами с генератора  $DD19$  кодовая комбинация считывается с  $DD16$  на выход, и одновременно 1, записанная в ячейку  $D0$  импульсами с элемента задержки  $DD20$ , продвигается по регистру  $DD17$ . На пятом такте она появляется на выходе, открывает УУ, которое подает управляющий сигнал на входы схем И  $DD1...DD9$ , что свидетельствует о том, что преобразование сообщения  $X8$  закончено и соответствующая ему кодовая комбинация передана в линию связи, и схема готова к кодированию следующего сообщения.

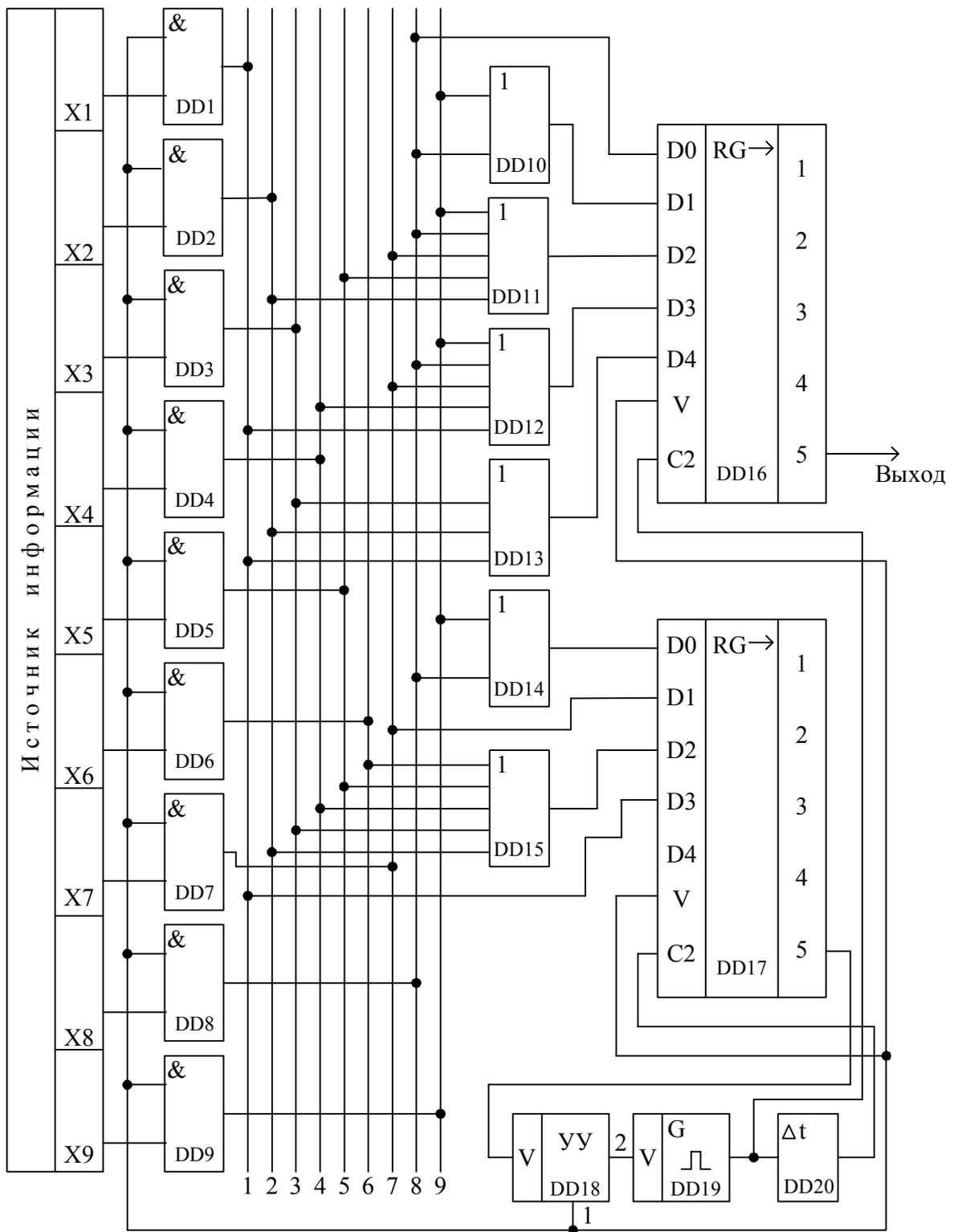


Рис. 3.22. Кодировальное устройство эффективных кодов

На рис. 3.23 приведена схема декодирующего устройства. Символы декодируемой кодовой комбинации, поступающие на вход  $D$  регистра  $DD1$ , продвигаются по нему импульсами тактового генератора  $DD2$ . Так как некоторые

из поступающих кодовых комбинаций начинаются с одного или нескольких нулей, то непосредственно по содержанию регистра невозможно определить начало этих комбинаций, а следовательно, и правильно их декодировать. Для однозначного определения начала каждой кодовой комбинации число ячеек регистра берут на единицу больше числа символов в самой длинной комбинации используемого эффективного кода. В дополнительной первой ячейке регистра  $DD1$  перед поступлением в него очередной декодируемой комбинации всегда записывают единицу (вход  $S1$ ). Продвигаясь по регистру, она сигнализирует о начале кодовой комбинации, а следовательно, и о ее длине. Дешифратор на элементах И  $DD4...DD12$  построен в соответствии с комбинациями используемого кода, в котором со стороны старшего разряда приписана лишняя единица. Например, для декодирования рассматриваемой комбинации 01111 на вход  $DD11$  должна будет поступить кодовая комбинация 101111. При поступлении в регистр последнего символа декодируемой первой комбинации (напоминаем, что код Хаффмана является префиксным) появляется 1 на выходе одной из схем И  $DD4...DD12$ , что соответствует приему сообщения  $X_i$ . Через схему ИЛИ  $DD14$  этот импульс дает на запись сообщения в соответствующую ячейку регистра  $DD13$ , и через элемент задержки  $DD3$  все ячейки регистра  $DD1$  устанавливаются в исходное положение (в первой ячейке 1, в остальных 0). Далее поступает следующая кодовая комбинация, и процесс декодирования повторяется

### 3.9. Схемы равнозначности кодов

Пусть заданы две совокупности переменных:  $X = (x_n, \dots, x_p, \dots, x_1)$  и  $Y = (y_n, \dots, y_p, \dots, y_1)$ . Тогда комбинационная схема, реализующая функцию  $F(X, Y) = (x_n, \dots, x_1, y_n, \dots, y_1)$ , которая равна 1 только при  $x_p = y_p$  для всех  $p = 1, \dots, n$ , называется схемой равнозначности кодов. Разряды  $x_p$  и  $y_p$  равны только в том случае, если  $x_p \oplus \bar{y}_p = 1$ . Поэтому функция

$$F(X, Y) = \prod_{p=1}^n (x_p \oplus \bar{y}_p), \quad (3.3)$$

или

$$F(X, Y) = \overline{\bigcup_{p=1}^n (x_p \oplus y_p)}, \quad (3.4)$$

принимает значение, равное 1, только при попарном равенстве всех одноименных разрядов кодов.

На рис. 3.24 и 3.25 показаны две схемы, реализующие функцию  $F(X, Y)$ , которые построены для  $n = 4$  на основании полученных выражений (3.3) и (3.4) соответственно.

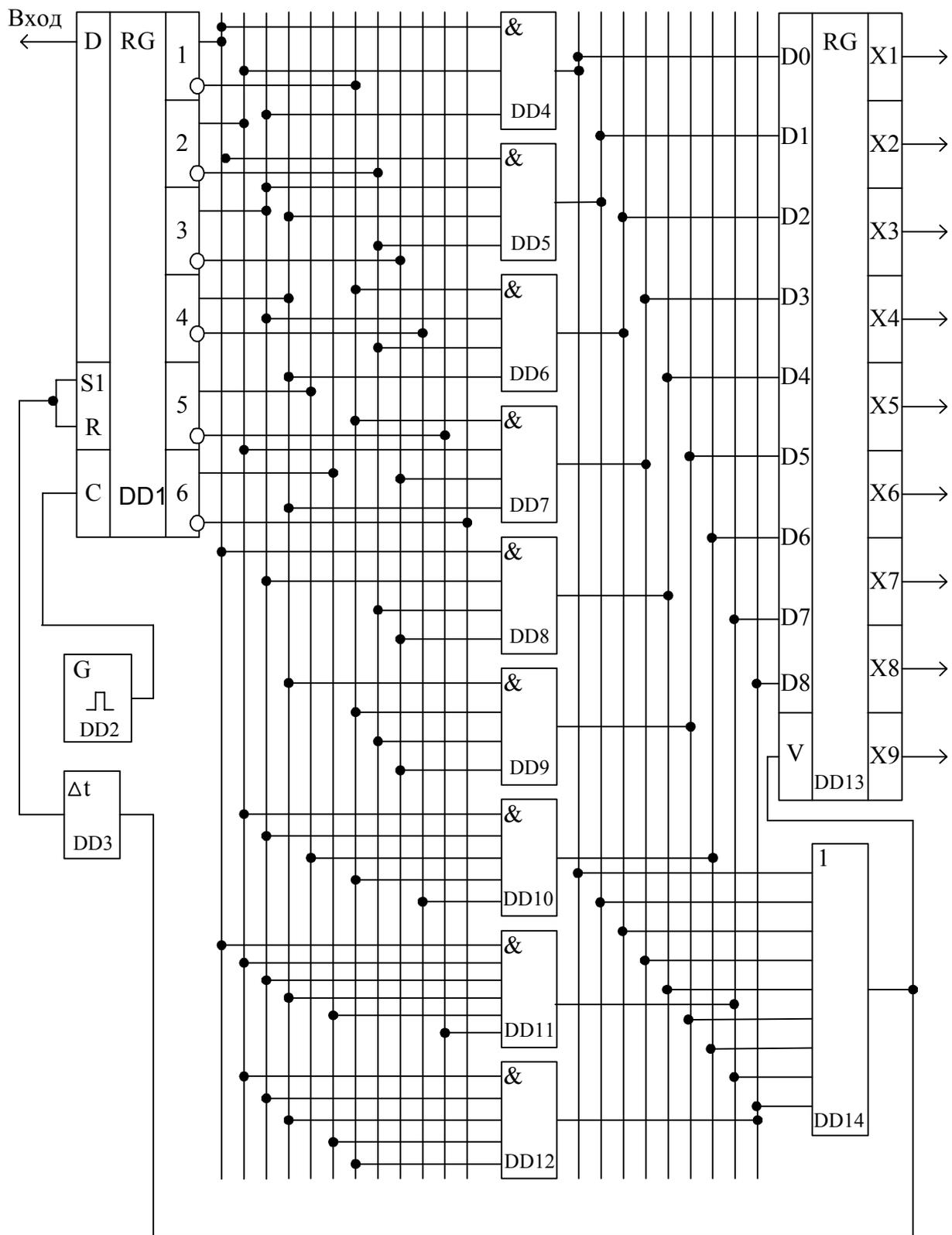


Рис. 3.23. Декодирующее устройство эффективных кодов

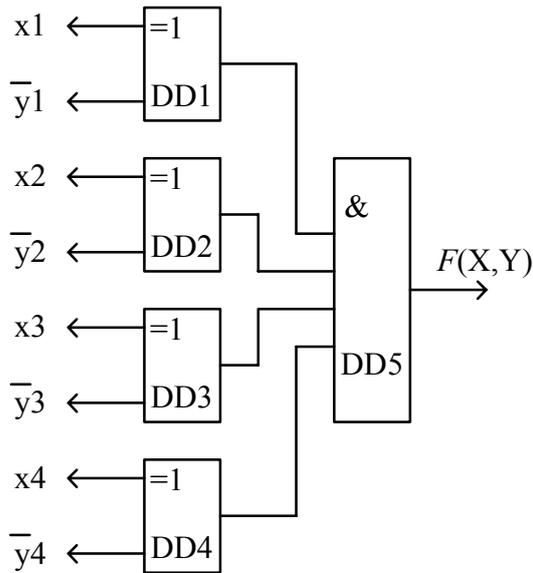


Рис. 3.24. Схема, реализующая функцию  $F(X,Y)$  по выражению (3.3)

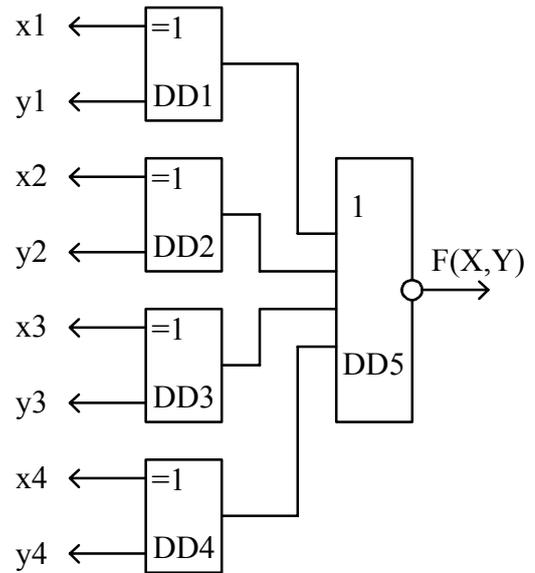


Рис. 3.25. Схема, реализующая функцию  $F(X,Y)$  по выражению (3.4)

Схема равнозначности упрощается при использовании сумматоров по модулю 2 с открытым коллектором (рис. 3.26). В качестве сумматоров по модулю 2 используется “исключающее ИЛИ” на два входа.

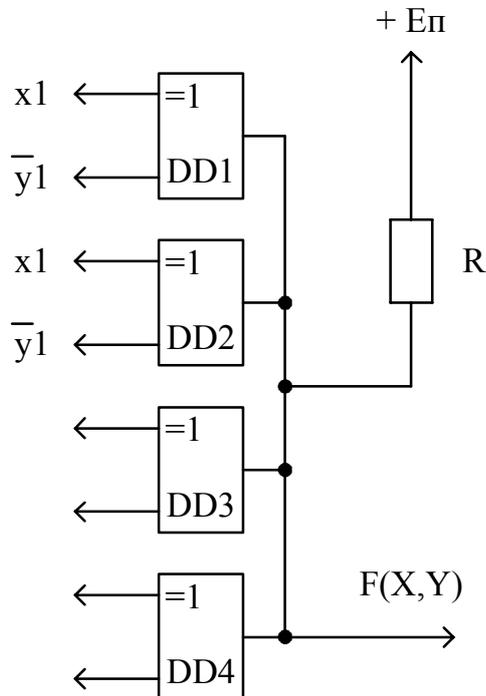


Рис. 3.26. Схема равнозначности на сумматорах по модулю 2 с открытым коллектором

### 3.10. Преобразователь параллельного кода в последовательный и обратно

Универсальным преобразователем является регистр сдвига. Принцип работы рассмотрим на базе четырехразрядного регистра сдвига К155ИР1 (рис. 3.27). Каждый разряд образован синхронным  $RS$ -триггером, включенным по схеме  $D$ -триггера с прямым динамическим входом синхронизации. Он имеет четыре параллельных входа данных  $D_0...D_3$  (выводы 2..5) и один последовательный вход данных  $S_1$  (вывод 1), а также четыре выхода  $Q_0...Q_3$  (выводы 13...10) от каждого из триггеров. Регистр имеет два тактовых входа  $\overline{C1}$  и  $\overline{C2}$ , управляемых отрицательным перепадом (спадом) тактового импульса, и вход разрешения параллельной загрузки  $\overline{PE}$ , который служит для выбора режима работы регистра.

Если на вход  $\overline{PE}$  подано напряжение высокого уровня, то разрешается работа тактовому входу  $\overline{C2}$ . В момент прихода на вход отрицательного перепада импульса в регистр загружаются данные от параллельных входов  $D_0...D_3$ .

Если на вход  $\overline{PE}$  подать напряжение низкого уровня, то разрешается работа тактовому входу  $\overline{C1}$ . С приходом отрицательного перепада тактового импульса на вход  $\overline{C1}$  данные последовательно сдвигаются от входа  $S_1$  на выход  $Q_0$ , затем на  $Q_1$ ,  $Q_2$ ,  $Q_3$  (т.е. вправо). Сдвиг данных по регистру влево будет происходить в том случае, если соединить выход  $Q_3$  и вход  $D_2$ ,  $Q_2$  и  $D_1$ ,  $Q_1$  и  $D_0$ .

Такой регистр можно использовать в качестве элемента буферной памяти арифметических устройств, элемента задержки, преобразователя последовательных кодов в параллельные, наоборот, делителя частоты, распределителя импульсов и других устройств.

Преобразователь параллельного кода в последовательный практически на любое число выходов можно реализовать на мультиплексорах. Рассмотрим восьмиканальный преобразователь параллельного кода в последовательный на ИМС К155КП7 (рис. 3.28). Данный мультиплексор представляет собой восьми-позиционный переключатель, имеющий три адресных входа  $S_0...S_2$  с высоким активным уровнем, один стробирующий вход  $\overline{E}$  и восемь информационных входов  $X_1...X_8$ .

Данные мультиплексоры позволяют коммутировать сообщения от восьми информационных входов на общую выходную линию. Логическое уравнение имеет вид

$$Y = x_1 \overline{S_0} \overline{S_1} \overline{S_2} + x_2 S_0 \overline{S_1} S_2 + x_3 \overline{S_0} S_1 \overline{S_2} + x_4 S_0 S_1 \overline{S_2} + x_5 \overline{S_0} \overline{S_1} S_2 + \\ + x_6 S_0 \overline{S_1} S_2 + x_7 \overline{S_0} S_1 S_2 + x_8 S_0 S_1 S_2.$$

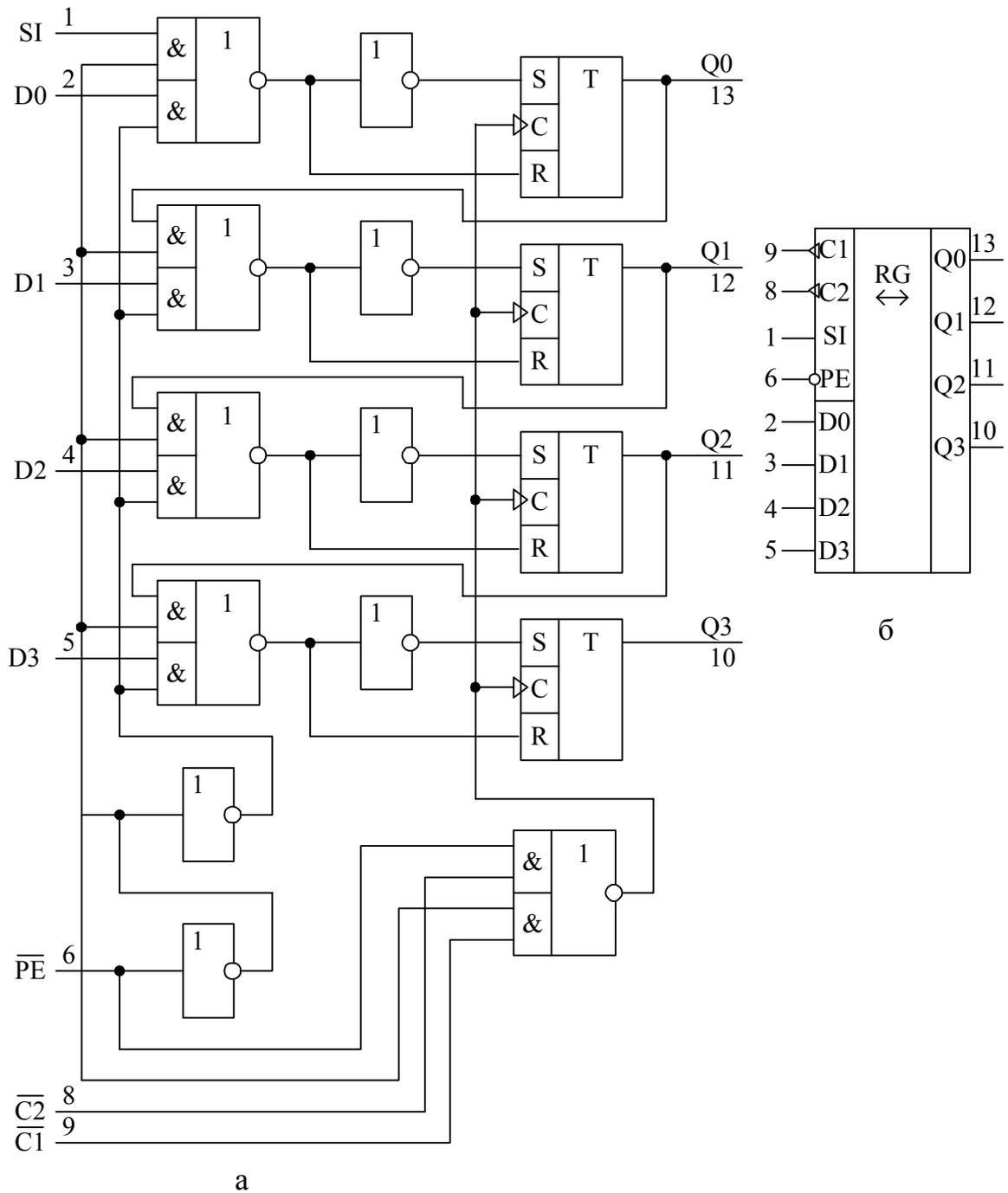


Рис. 3.27. Структура (а) и условное обозначение (б) универсального регистра ИР1

На адресные входы управляющие сигналы поступают от двоичного счетчика *DD2*. В зависимости от кода на входах *S0...S2* к выходу *Y* подключается соответствующий вход  $x_i$  (табл. 3.12). При передаче кодовых сообщений в линию связи без разделительных пауз на вход  $\bar{E}$  постоянно подается 0, а передача с пассивными паузами осуществляется путем подачи на вход  $\bar{E}$  тактовых импульсов (как показано на рис. 3.28). Для обслуживания устройств с большим числом выходов производится параллельное подключение мультиплексов.

Состояния мультиплексора КП7

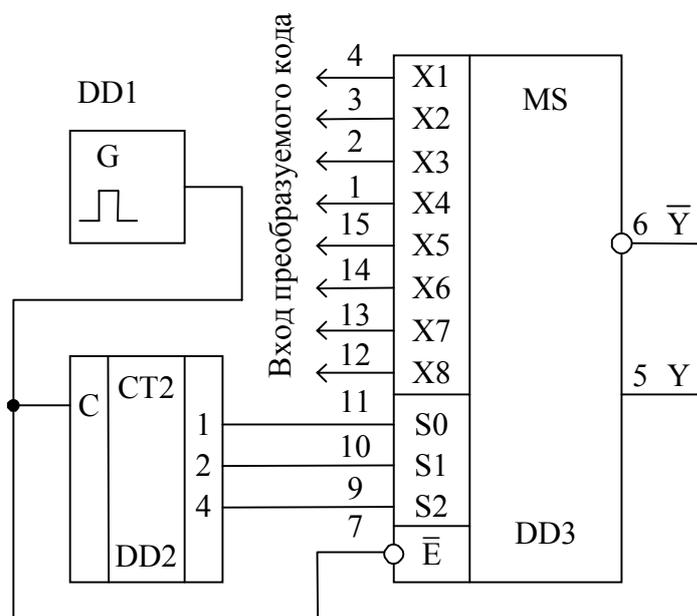


Рис. 3.28. Преобразователь параллельного кода в последовательный

Входы			E	Y	$\bar{Y}$
Выбор					
S2	S1	S0			
x	x	x	1	0	1
0	0	0	0	x1	$\bar{x1}$
0	0	1	0	x2	$\bar{x2}$
0	1	0	0	x3	$\bar{x3}$
0	1	1	0	x4	$\bar{x4}$
1	0	0	0	x5	$\bar{x5}$
1	0	1	0	x6	$\bar{x6}$
1	1	0	0	x7	$\bar{x7}$
1	1	1	0	x8	$\bar{x8}$

На рис. 3.29 показана схема преобразователя на 16 входов, а на рис. 3.30 – на 64 входа.

На входы  $S_0$ ,  $S_1$ ,  $S_2$  поступает код, выбирающий тот или иной вход из восьми возможных в каждом коммутаторе, т.е. выбирающий сразу два одинаковых входа (рис. 3.29) или восемь входов (рис. 3.30). На какой коммутатор –  $DD1$ ,  $DD2$  (см. рис. 3.29) или  $DD2...DD9$  (см. рис. 3.30) поступает адрес от двоичного счетчика, зависит от сигнала на входе стробирования  $\bar{E}$ . Для преобразователя на 64 выхода сигнал стробирования формируется дешифратором  $DD1$  (рис. 3.30) при подаче на его вход кода с весами разрядов 8–16–32.

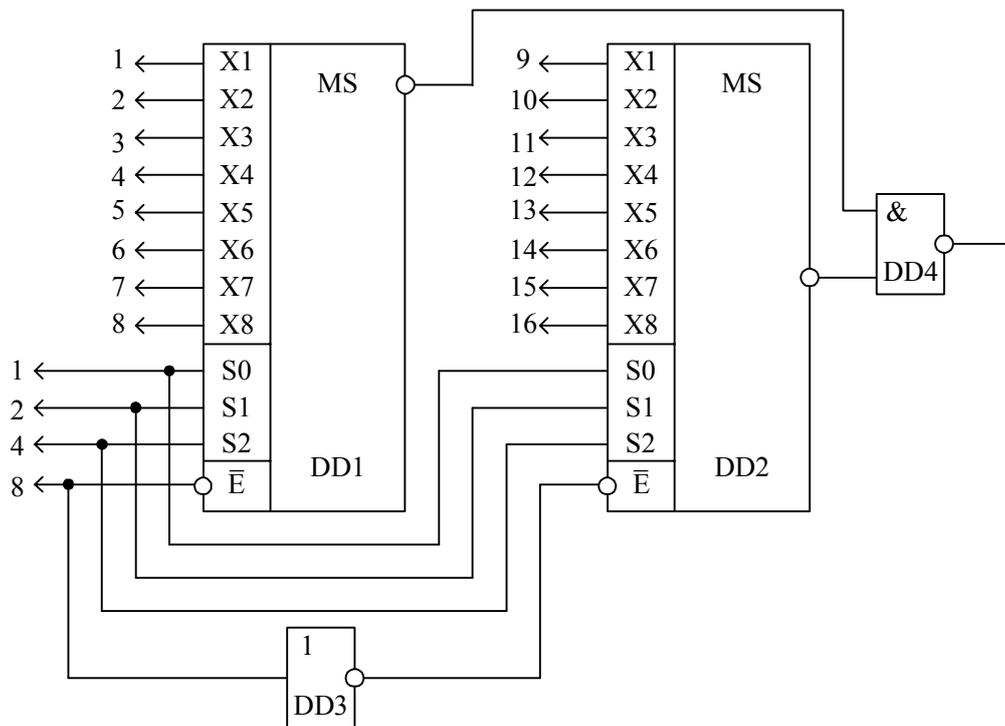


Рис. 3.29. Схема преобразователя на 16 входов

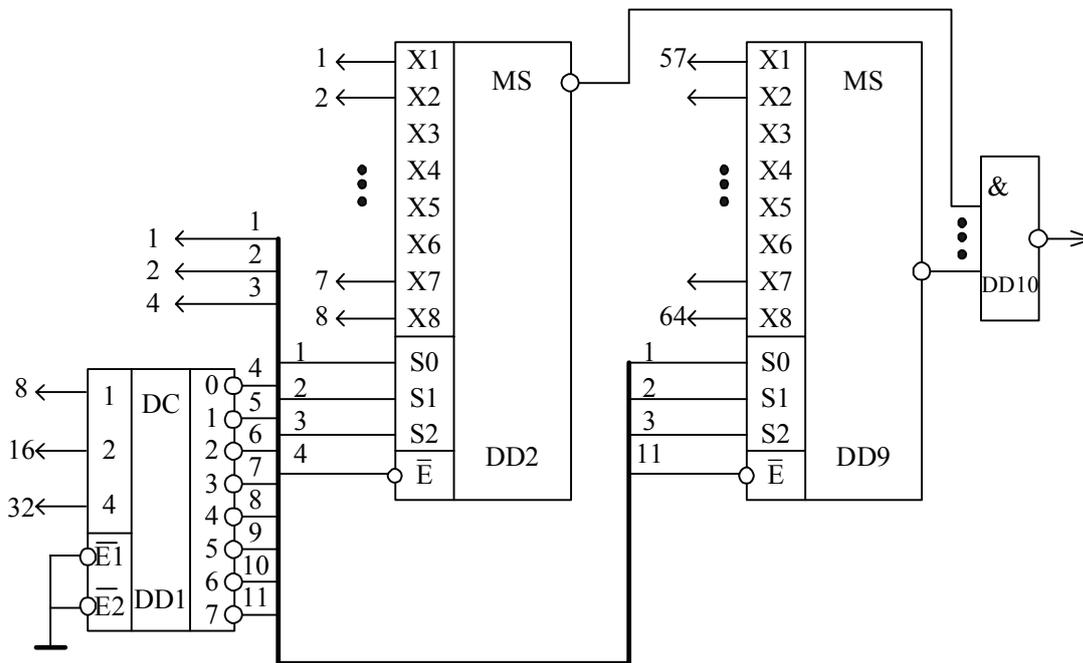


Рис. 3.30. Схема преобразователя на 64 входа

## 4. ТЕХНИЧЕСКИЕ СРЕДСТВА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ КОРРЕКТИРУЮЩИХ КОДОВ

### 4.1. Кодер и декодер кода с защитой на четность

Функциональная схема такого кодера для четырехразрядного кода приведена на рис. 4.1.

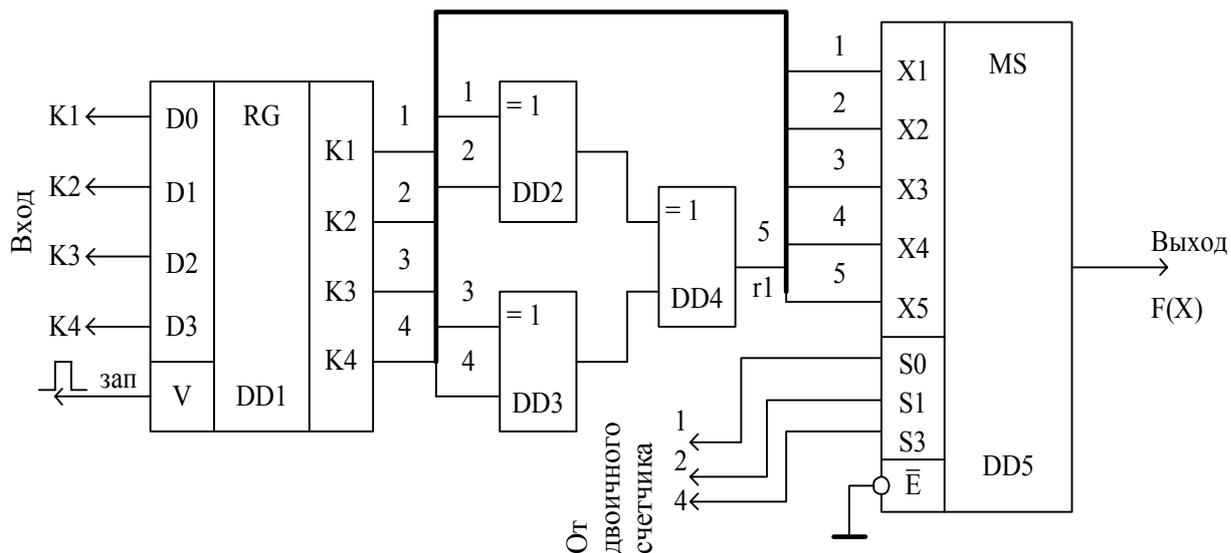


Рис. 4.1. Кодер кода с защитой на четность

Схема работает следующим образом. При подаче импульса запуска исходная кодовая комбинация  $k_1k_2k_3k_4$  записывается в регистр  $DD1$  и одновременно поступает на информационные входы мультиплексора  $DD5$ . На элементах  $DD2...DD4$  собрана схема контроля четности. Если число единиц в исходной кодовой комбинации четное, то на выходе  $DD4$  нуль, если нечетное – единица. Результат этой проверки является контрольным символом  $r_1$ , который поступает на вход  $X5$  мультиплексора  $DD5$ . При поступлении двоичного кода на адресные входы  $S0, S1, S3$  мультиплексор последовательно передает на выход кодовую комбинацию, находящуюся на входах  $X1 - X5$ .

Например, пусть необходимо закодировать кодовую комбинацию 1101. После ее записи в  $DD1$  на выходе  $DD4$  появится контрольный символ, равный единице, а на выходе мультиплексора – последовательный код  $F(x) = 11011 = k_1k_2k_3k_4r_1$ .

Для формирования контрольного разряда  $r_1$  можно использовать счетный триггер вместо ИМС  $DD2...DD4$ , и тогда схема кодера будет иметь вид, представленный на рис. 4.2.

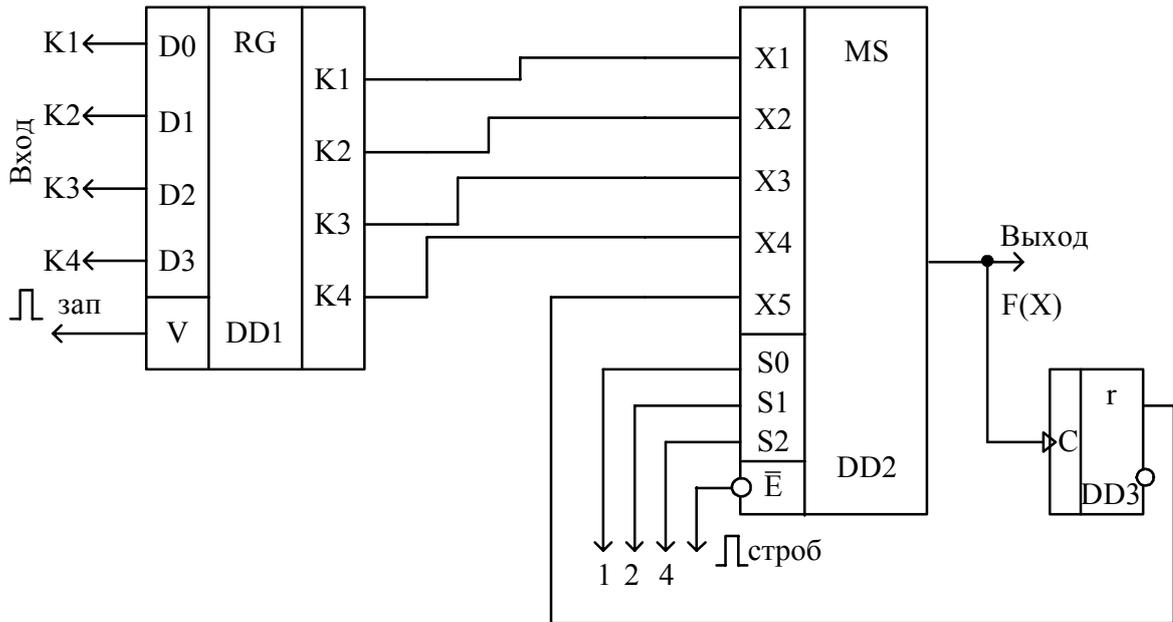


Рис. 4.2. Схема формирования контрольного символа с помощью счетного триггера

Для работы триггера в счетном режиме необходимо иметь паузы между символами, что достигается путем подачи стробирующих импульсов на вход  $\bar{E}$  мультиплексора.

При декодировании принятая кодовая комбинация  $F^*(x)$  проверяется на четность. Если число единиц четное, то искажений нет, в противном случае кодовая комбинация бракуется. Схема декодера на 5 разрядов приведена на рис. 4.3.

В качестве схемы контроля четности используется сумматор по модулю 2. Если в комбинации  $F^*(x)$  четное количество единиц, то на инверсном выходе  $DD2$  будет единица, которая поступит на один из входов схемы И  $DD3$ . На второй вход на пятом такте поступит импульс опроса. При наличии двух единиц на входе, на выходе  $DD3$  появляется единица, которая откроет схемы И  $DD4.1 \dots DD4.4$  и информационные символы поступят в приемник. Если в комбинации  $F^*(x)$  будет нечетное число единиц, то сигнал с выхода формирователя  $DD5$  сбросит регистр в исходное положение.

Сумматор по модулю 2 на любое число входов может быть построен на двухвходовых схемах “исключающее ИЛИ”, или можно использовать схемы контроля четности  $ИП2$ ,  $ИП5$

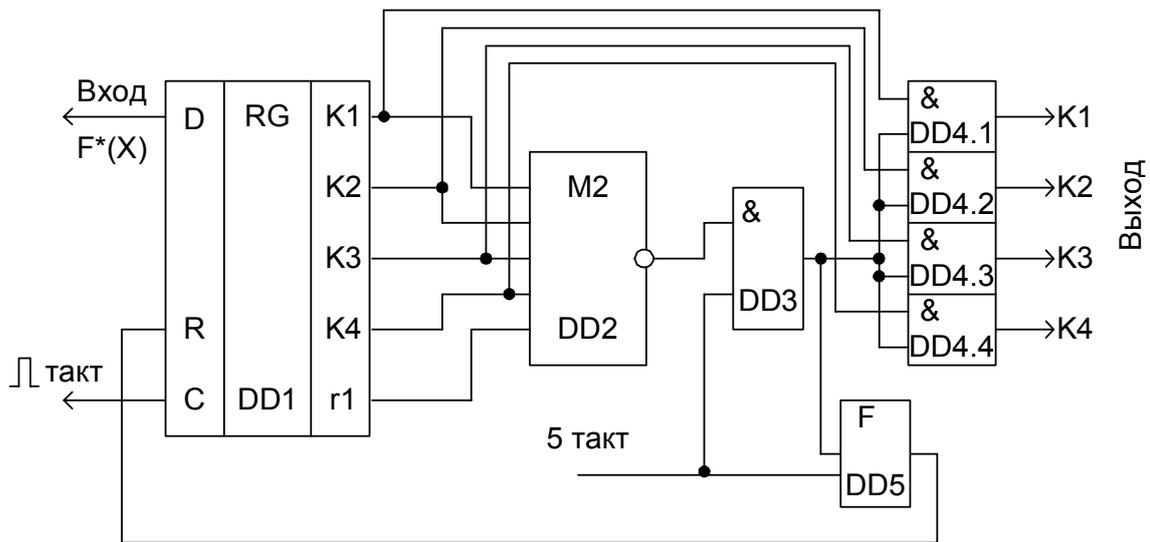


Рис. 4.3. Схема декодера кода с защитой на четность

На рис. 4.4 показано условное обозначение и цоколевка микросхемы К155ИП2, а в табл. 4.1 – состояния ИМС.

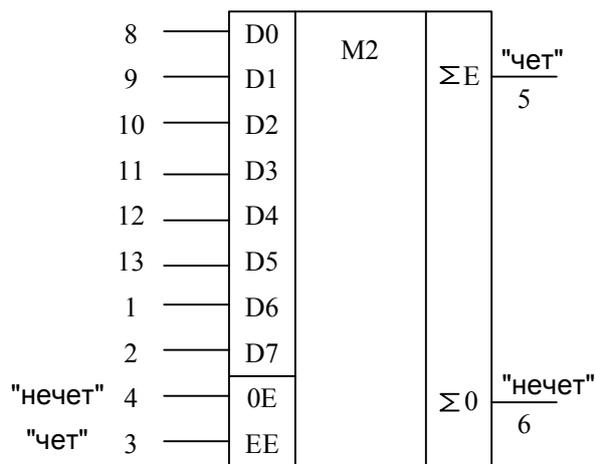


Рис. 4.4. Условное обозначение ИМС ИП2

Таблица 4.1

Состояния ИП

Сумма единиц на входах D0...D7	Входы		Выходы	
	EE	OE	ΣE	Σ0
Четная	1	0	1	0
Нечетная	1	0	0	1
Четная	0	1	0	1
Нечетная	0	1	1	0
Неопределенная	1	1	0	0
Неопределенная	0	0	1	1

## 4.2. Кодер и декодер кода с постоянным весом

Кодер с постоянным весом применяется в системах ТУ. Принцип построения этого кода и корректирующие возможности изложены в подразд. 2.2.1. Рассмотрим кодер для кода  $C_4^2$ . С его помощью можно передать 6 команд (2.6). Поставим в соответствие каждой команде (сообщению) свою кодовую комбинацию:

$$\begin{aligned}
 x_1 = 0011 &= \bar{y}_4 \bar{y}_3 y_2 y_1, & x_2 = 0101 &= \bar{y}_4 y_3 \bar{y}_2 y_1, \\
 x_3 = 1001 &= y_4 \bar{y}_3 \bar{y}_2 y_1, & x_4 = 0110 &= \bar{y}_4 y_3 y_2 \bar{y}_1, \\
 x_5 = 1010 &= y_4 \bar{y}_3 x_2 \bar{x}_1, & x_6 = 1100 &= y_4 y_3 \bar{y}_2 \bar{y}_1.
 \end{aligned}
 \tag{4.1}$$

Откуда получим выражения для разрядов  $y_i$  через сообщения  $x_i$  в виде:

$$\begin{aligned}
 y_1 &= x_1 + x_2 + x_3, & y_2 &= x_1 + x_4 + x_5, \\
 y_3 &= x_2 + x_4 + x_6, & y_4 &= x_3 + x_5 + x_6.
 \end{aligned}
 \tag{4.2}$$

Кодер, построенный в соответствии с приведенными выражениями, приведен на рис. 4.5.

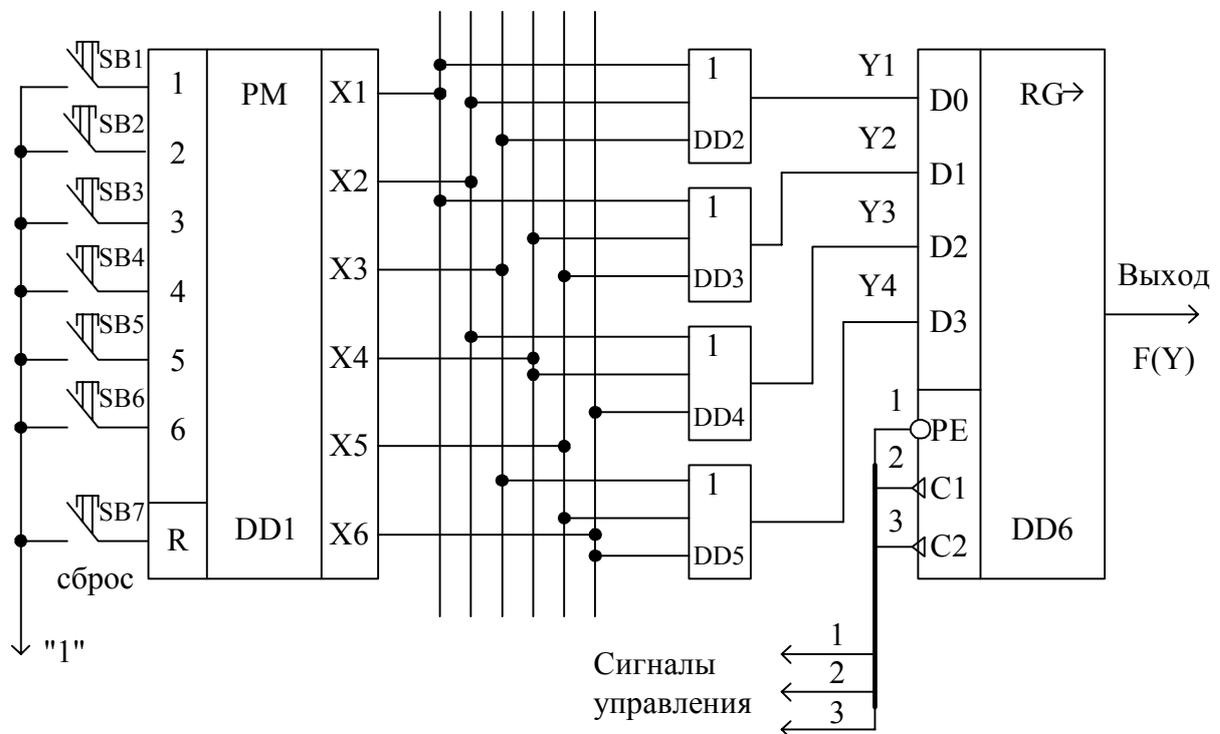


Рис. 4.5. Кодер кода  $C_4^2$

Подлежащие кодированию сообщения заносятся в ОЗУ  $DD1$  и с выходов  $x_1, x_2, x_3, x_4, x_5, x_6$  поступают на кодер, который собран на элементах  $DD2...DD5$ . Входы этих элементов соединены с выходами ОЗУ  $DD1$ , в соответствии с выражениями (4.2). Закодированная кодовая комбинация  $y_4, y_3, y_2, y_1$  записывается в регистр  $DD6$ , а затем последовательно выдвигается в линию связи.

Функциональная схема декодера для кода  $C_4^2$  приведена на рис. 4.6.

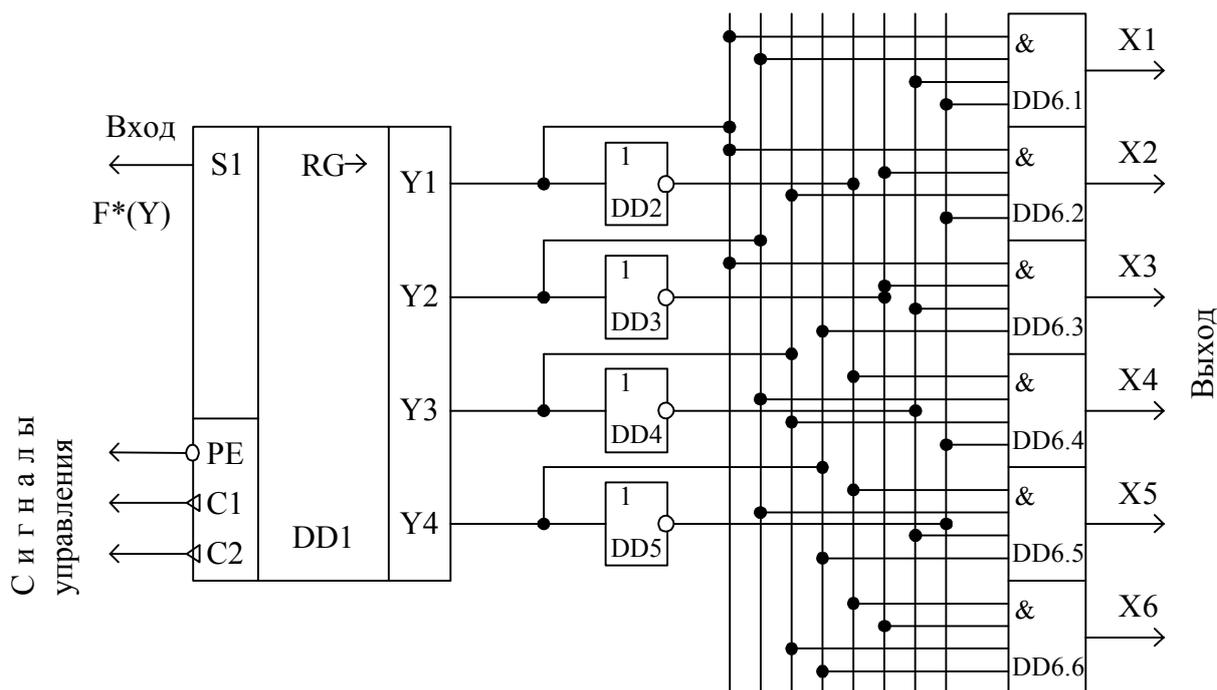


Рис. 4.6. Декодер кода  $C_4^2$

Кодовая комбинация  $F^*(Y)$  из линии связи поступает в приемный регистр  $DD1$  и далее на собственно сам дешифратор, собранный на элементах И  $DD6.1 \dots DD6.6$ , входы которых заведены в соответствии с выражениями (4.1). На выходе схем И  $DD6.1 \dots DD6.6$  получаем сигналы, соответствующие передаваемым сообщениям  $x_1, \dots, x_6$ . Нетрудно видеть, что данная схема обладает защитным отказом, т.е. при поступлении на вход кодовых комбинаций, содержащих количество единиц, отличное от двух, ни на одном из выходов  $x_1, \dots, x_6$  сигнала не будет.

### 4.3. Кодер и декодер кода с двумя проверками на четность

Принцип образования кодовых комбинаций в данном коде описан в подразд. 2.2.5.

Кодирующее устройство для  $k = 6$  показано на рис. 4.7. Оно состоит из входного регистра  $DD1$ , схем контроля четности  $DD2, DD3$  и преобразователя параллельного кода в последовательный  $DD4$ .

Схемой  $DD2$  формируется первый контрольный разряд  $r_1$ , дополняющий до четности всех информационных разрядов, а схемой  $DD3$  формируется второй контрольный разряд  $r_2$ , дополняющий до четности всех нечетных информационных разрядов, т.е.  $k_1, k_3$  и  $k_5$

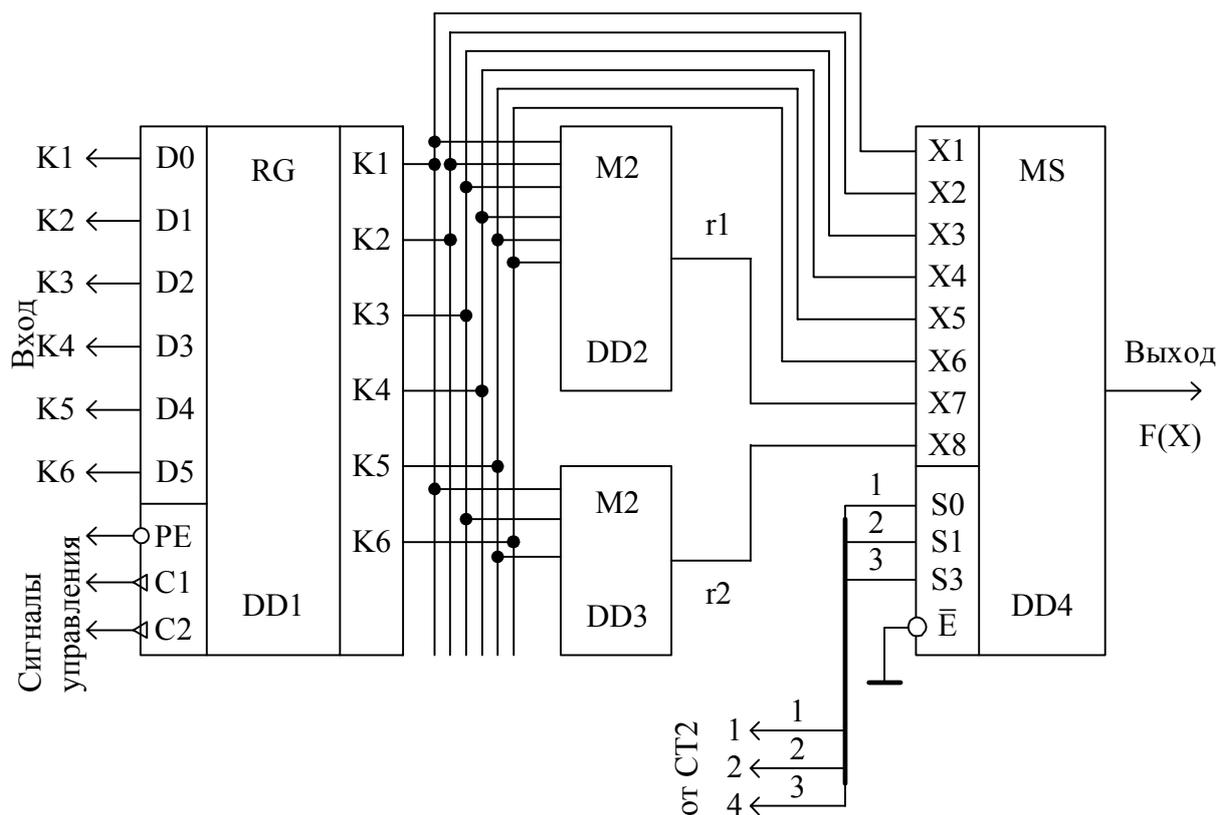


Рис. 4.7. Кодер кода с двумя проверками на четность

Декодирующее устройство рассматриваемого кода приведено на рис. 4.8. Оно состоит из приемного регистра  $DD1$ , двух схем контроля четности  $DD2$  и  $DD3$ , определителя синдрома  $DD4$ , формирователя сигнала “сброс”  $DD5$  и схем вывода информационных символов  $DD6...DD11$ .

Если кодовая комбинация  $F^*(x)$  поступает в приемник без искажений, то на инверсном выходе схемы контроля четности  $DD2$ , осуществляющей проверку  $k_1 \oplus k_2 \oplus k_3 \oplus k_4 \oplus k_5 \oplus k_6 \oplus r_1$ , появится 1. На инверсном выходе схемы контроля четности  $DD3$ , осуществляющей проверку  $k_1 \oplus k_3 \oplus k_5 \oplus r_2$ , тоже будет 1. Эти оба сигнала поступят на вход определителя синдрома  $DD4$ , и единичный сигнал с его выхода дает разрешение на вывод получателю информационных символов через элементы И  $DD6...DD11$ . Если  $F^*(x)$  была искажена, то на инверсном выходе одной или двух схем контроля четности будет сигнал 0, что вызовет запрет на выдачу информационных символов получателю информации, а 1 на выходе  $DD5$  вызовет сброс регистра  $DD1$  в исходное положение. Следует отметить, что опрос определителя синдрома  $DD4$  происходит на девятом такте, т.е. после приема всей кодовой комбинации из линии связи.

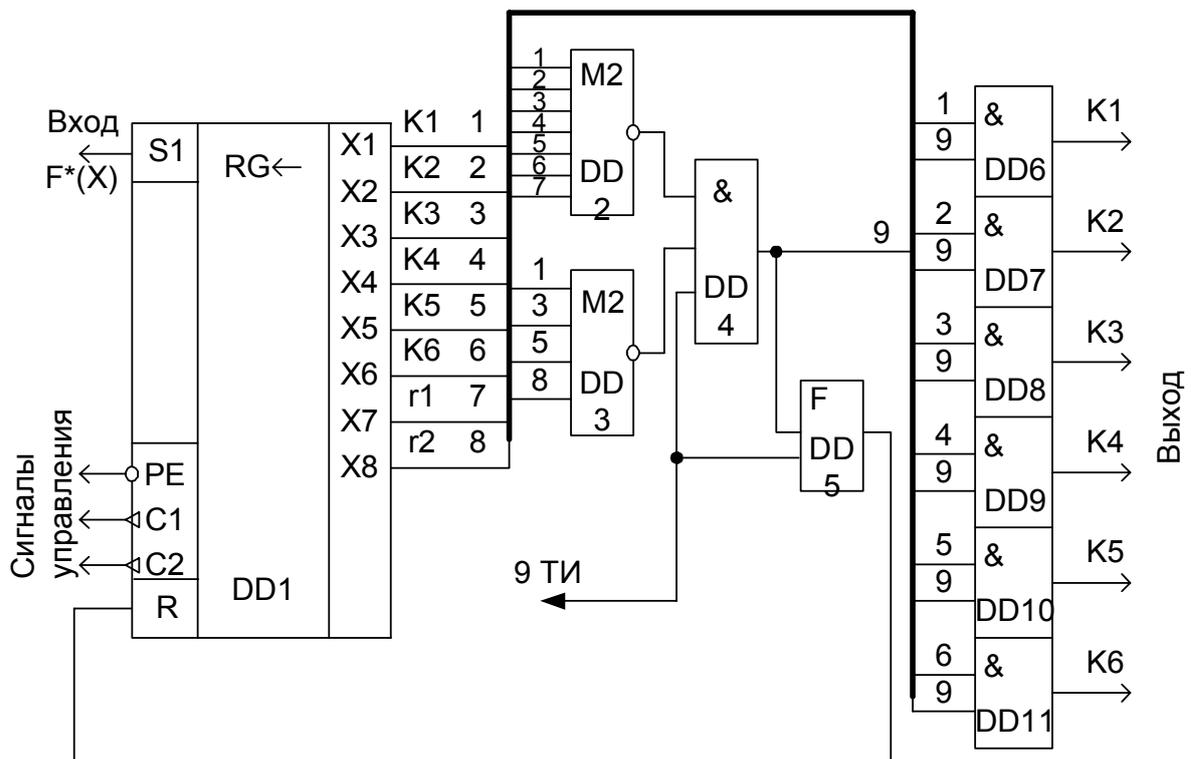


Рис. 4.8. Декодирующее устройство кода с двумя проверками на четность

#### 4.4. Кодер и декодер кода с повторением

Как известно из подразд. 2.2.6, существуют две разновидности этого кода. Первая – это когда исходная комбинация повторяется  $m$  раз и вторая – когда каждый элемент повторяется  $m$  раз. Кодированные устройства для первого и второго вариантов представлены на рис. 4.9 и 4.10 соответственно.

Кодер рис. 4.9 работает следующим образом. Кодовая комбинация, подлежащая кодированию, заносится в регистр DD1. По сигналу “Пуск” триггер DD2 устанавливается в 1, открывается схема И DD4, и тактовые импульсы от генератора DD3 поступают на вход счетчика тактов DD5 и одновременно на вход C2 кольцевого регистра DD1. Исходная кодовая комбинация выдвигается на выход  $m$  (в данном случае  $m = 3$ ) раз. Число повторений устанавливается счетчиком тактов DD5. После того как счетчиком DD5 будет зафиксировано  $3k$  тактов, на выходе счетчика DD5 появляется сигнал, который устанавливает триггер DD2 в исходное положение и схема готова к кодированию следующего сообщения.

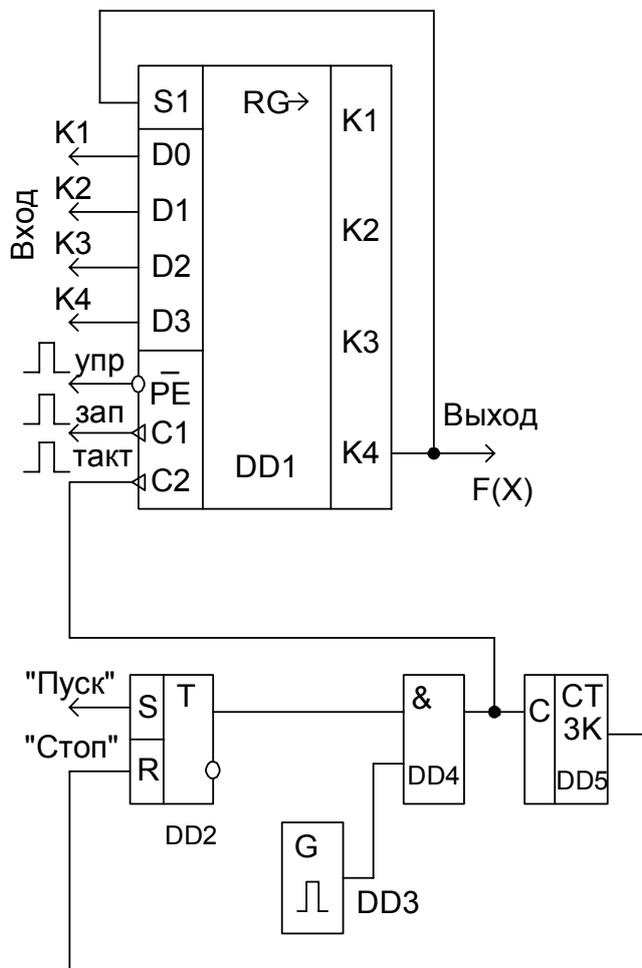


Рис. 4.9. Кодер кода с трехкратным повторением исходной комбинации

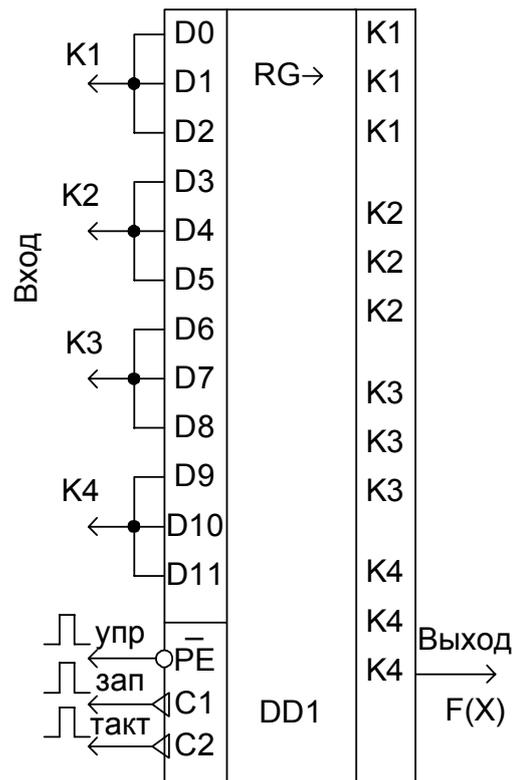


Рис. 4.10. Кодер кода с трехкратным повторением каждого элемента исходной комбинации

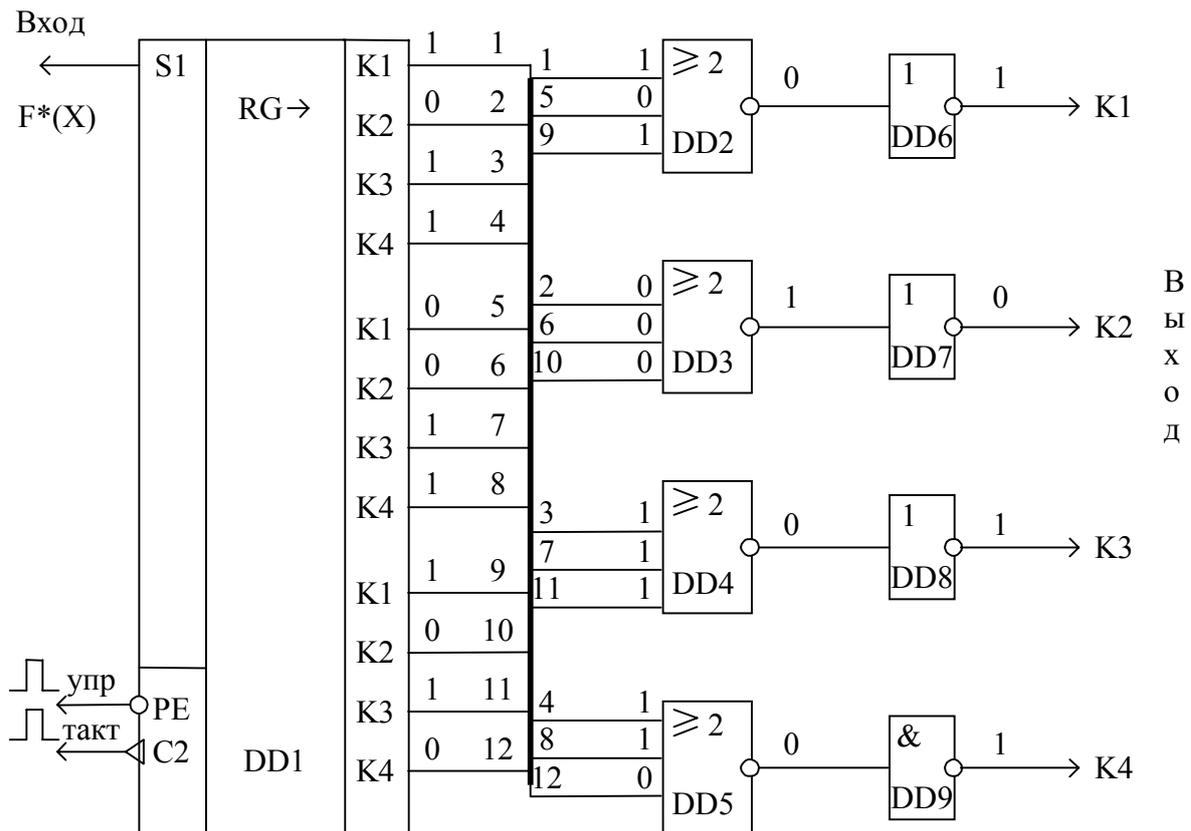
Например, если на вход поступило сообщение  $G(x) = 1101 = k_4k_3k_2k_1$ , то в результате трехкратной передачи в линию связи поступит кодовая комбинация  $F(x) = 110111011101$ .

В кодере рис. 4.10 каждый символ  $k_i$  исходной комбинации записывается в три рядом стоящих ячейки. Таким образом, после подачи  $3k$  импульсов на вход  $C2$  регистра  $DD1$  на выход поступит кодовая комбинация  $k_4k_4k_4k_3k_3k_3k_2k_2k_2k_1k_1k_1$ . Например, если подлежала кодированию сообщение  $G(x) = k_4k_3k_2k_1 = 1001$ , то в линию связи поступит кодовая комбинация  $F(x) = 111\ 000\ 000\ 111$ .

Декодирование заключается в обнаружении и исправлении ошибок. Для исправления ошибок применяется мажоритарный принцип, т.е. за истинное значение информационного символа принимается то, которое большее число раз встречается в этом информационном и соответствующих ему контрольных символах. При трехкратном повторении решение принимается по двум символам из трех.

Как указано в подразд. 2.2.6 наибольшее применение нашел код с повторением комбинаций, как обеспечивающий более высокую помехоустойчивость. Поэтому декодирующее устройство рассмотрим для этого случая, схема которого приведена на рис. 4.11 для  $k = 4$ ,  $m = 3$  и  $n = 12$ .

Кодовая комбинация  $F^*(x)$  из линии связи в последовательном коде заносится в регистр  $DD1$ . С выхода  $DD1$  каждый информационный символ  $k_i$ , поступает на один из входов трехвходового мажоритарного элемента, на остальные два входа подаются соответствующие ему контрольные символы.



$DD1$  К155ИР1 - 3 шт,  $DD2...DD5$  К533ЛПЗ (К1533ЛПЗ),  
 $DD6...DD9$  К155ЛН1

Рис. 4.11. Функциональная схема декодера кода с 3-кратным повторением исходной комбинации

Трехвходовые мажоритарные элементы  $DD2...DD5$  с инверсным выходом выполняют в общем виде функцию  $y = \overline{x_1x_2 \cup x_1x_3 \cup x_2x_3}$ . С учетом инверторов  $DD6...DD9$  на выходе каждого элемента функция будет описываться выражением  $y = x_1x_2 \cup x_1x_3 \cup x_2x_3$ , т.е. сигнал на выходе инвертора будет равен 1(0) только при поступлении на вход мажоритарного элемента двух и более

входных сигналов  $x_i$ , равных 1(0). После принятия решения каждым мажоритарным элементом о присвоении значения тому или иному информационному символу они поступают в приемник информации. Рассмотрим на примере передачи кодовой комбинации  $F(x) = 110111011101$ , которая под действием помех была искажена и на вход регистра  $DD1$  поступила в виде  $F^*(x) = \dot{0}101110\dot{0}1101$ , искаженные символы помечены точкой. Сигналы на входе и выходе каждого элемента указаны на рис. 4.11.

Как видно из рис. 4.11 в результате принятия решения элементами  $DD2...DD5$  исходное сообщение имеет вид  $G(x) = k_4k_3k_2k_1 = 1101$ , что соответствует информационной части  $F(x)$ .

В заключение следует указать, что построение мажоритарных элементов на число входов больше пяти целесообразно на двоичных сумматорах, например, К155ИМ3, К155ИМ2.

#### 4.5. Кодер и декодер кода с числом единиц, кратным трем

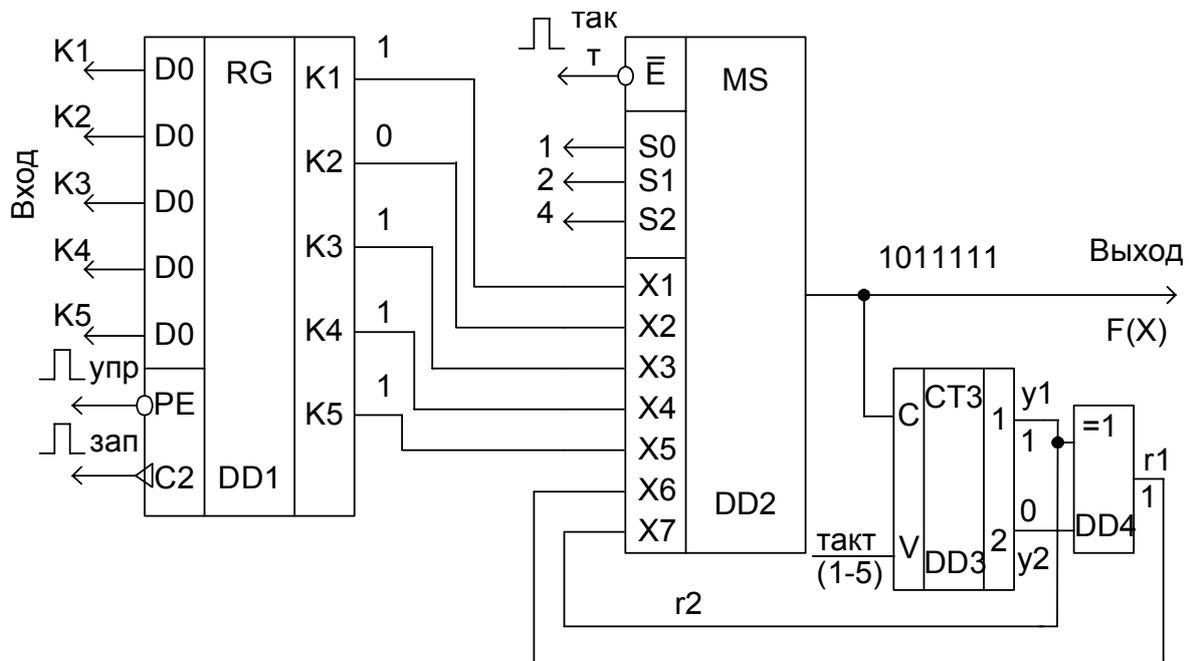


Рис. 4.12. Кодер кода с числом единиц, кратным трем

Как указано в подразд. 2.2.7, кодовые комбинации в данном коде содержат два контрольных символа, причем если первый  $r_1$  символ равен 0, то и второй  $r_2$  тоже должен быть равен 0. Кодировующее устройство для  $k = 5$  приведено на рис. 4.12.

Основой кодера является тактируемый счетчик  $DD3$  с коэффициентом счета 3, который подсчитывает число единиц в информационной части. Возможны следующие состояния счетчика:  $y_2y_1 = 00$ ,  $y_2y_1 = 01$ ,  $y_2y_1 = 10$ , что соответствует комбинациям контрольных символов соответственно:  $r_1r_2 = 00$ ,

$r_1 r_2 = 11$ ,  $r_1 r_2 = 10$ . Таким образом, формирователь контрольных символов на элементах  $DD3$  и  $DD4$  описывается выражениями

$$r_1 = \bar{y}_2 y_1 + y_2 \bar{y}_1, \quad r_2 = y_1.$$

Мультиплексор  $DD2$  осуществляет преобразование параллельного кода в последовательный. Процесс кодирования сообщения  $G(x) = k_1 k_2 k_3 k_4 k_5 = 10111$  показан на рисунке кодера.

Основой декодера (рис. 4.13) является счетчик  $DD2$  с коэффициентом деления 3.

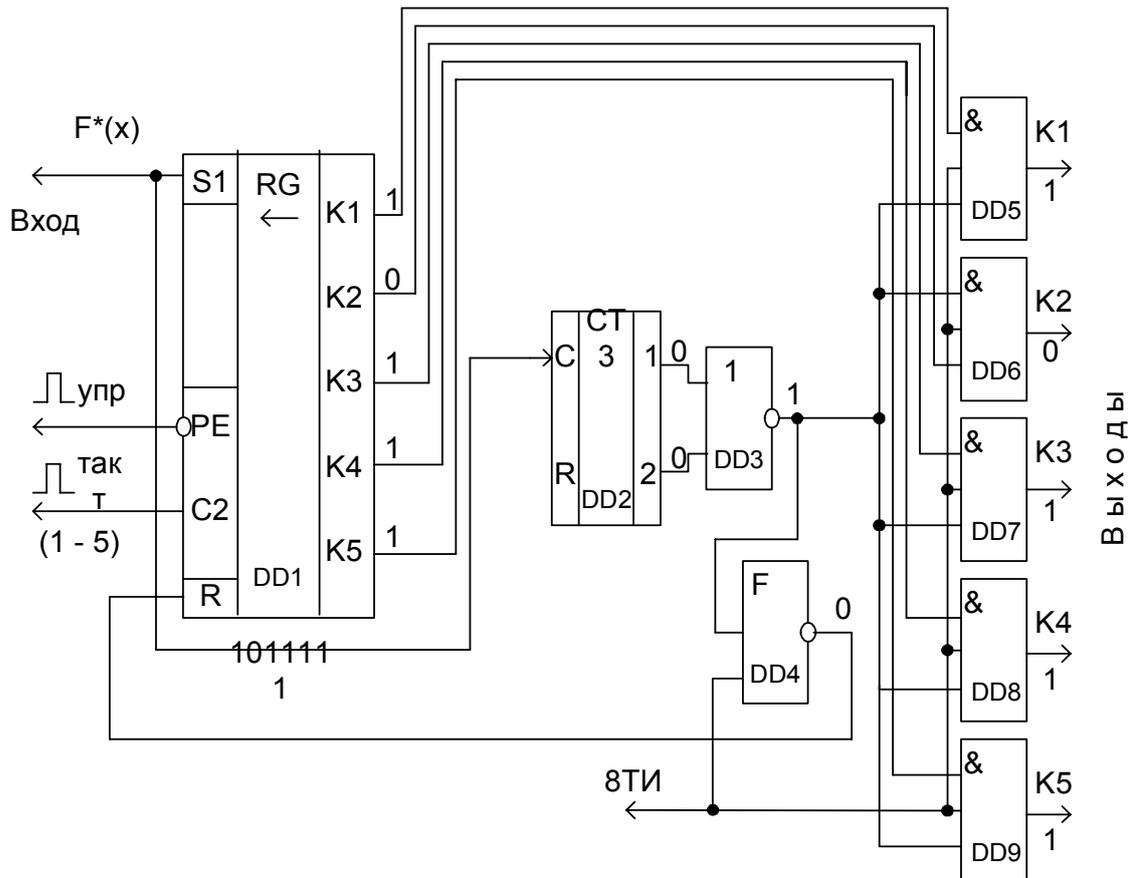


Рис. 4.13. Схема декодера кода с числом единиц, кратным трём

На первых пяти тактах информационные символы заносятся в регистр  $DD1$ , а полная кодовая комбинация  $F^*(x)$  на 1–7–м тактах поступает в счетчик  $DD2$ . Если в кодовой комбинации  $F^*(x)$  искажений нет, то после 7–го такта в счетчике будет зафиксирован синдром 00. На выходе элемента  $DD3$  появится 1, которая разрешает вывод информационных символов  $k_5, k_4, k_3, k_2, k_1$  на такте 8 в приемник информации. В противном случае, при наличии ошибок в  $F^*(x)$ , на выходе элемента  $DD3$  появится 0, что запретит вывод информации в приемник через элементы И  $DD5 \dots DD9$ , а 1 на выходе формирователя  $DD4$  сбросит ре-

гистр  $DD1$  в исходное положение. Процесс декодирования кодовой комбинации  $F^*(x)=k_1k_2k_3k_4k_5r_1r_2 = 1011111$  показан на рис. 4.13.

#### 4.6. Кодер и декодер инверсного кода

Теоретические вопросы построения данного кода рассмотрены в подразд. 2.2.8, а функциональная схема кодера для четырехразрядных сообщений приведена на рис. 4.14.

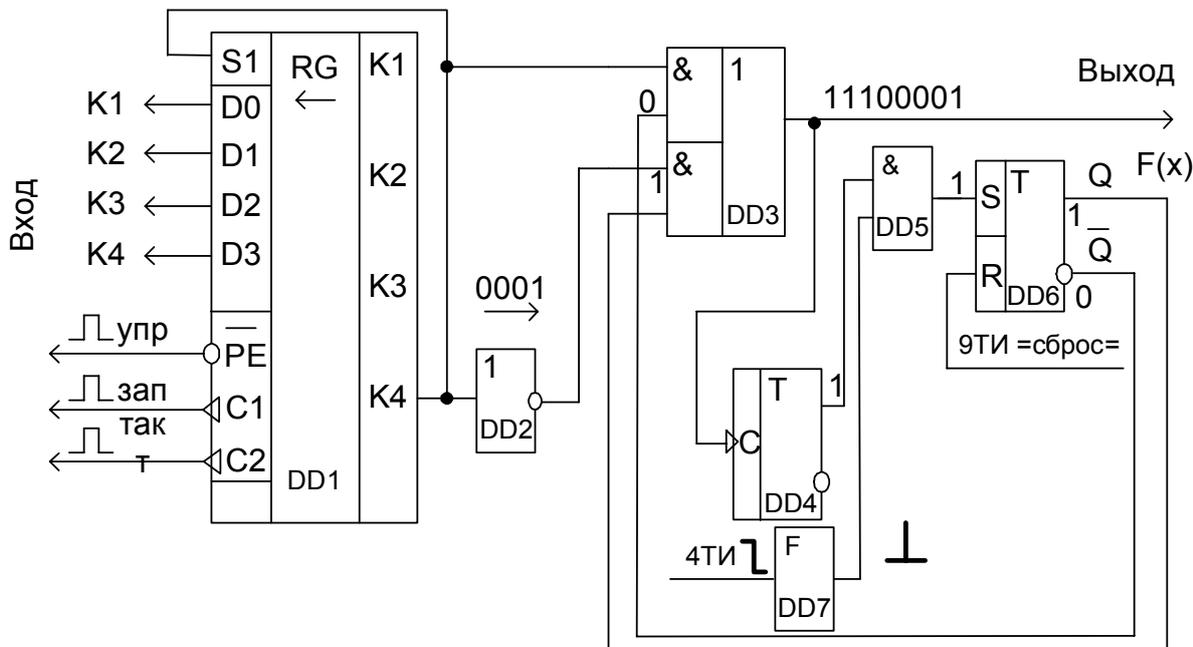


Рис. 4.14. Кодер инверсного кода для  $k = 4$

Подлежащее кодированию сообщение записывается в кольцевой регистр  $DD1$ , а затем на первых четырех тактах через верхнюю по схеме 2И-ИЛИ  $DD3$  выдвигается в линию связи и одновременно повторно через вход  $S1$  записывается в регистр  $DD1$ . Триггер  $DD4$ , работающий в счетном режиме, является сумматором по модулю 2. Спадом 4ТИ через формирователь  $DD7$  производится опрос состояния триггера  $DD4$ . Если число единиц в информационной части было четным, то триггер  $DD4$  окажется в нулевом положении, а следовательно, состояние  $RS$ -триггера  $DD6$  не изменится. По-прежнему будет открыта верхняя схема И  $DD3$  сигналом  $\bar{Q}$  триггера  $DD6$ , и контрольные символы в неизменном виде повторяют информационные, которые на предыдущих четырех тактах поступают в линию связи. Если в информационной части было нечетное

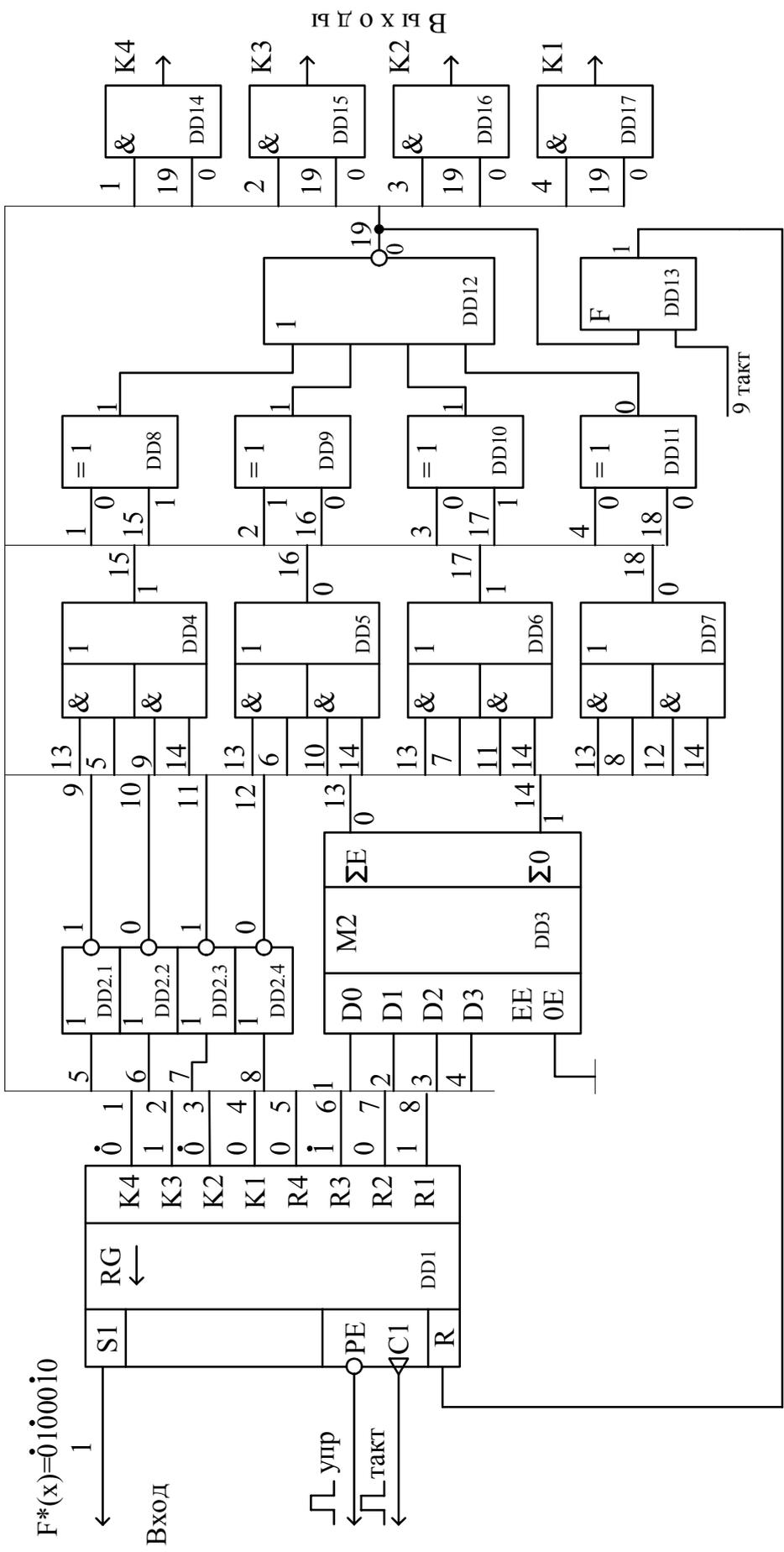


Рис. 4.15. Декодер инверсного кода

число единиц, то триггер  $DD4$  будет в единичном положении, и спадом 4ТИ триггер  $DD6$  тоже устанавливается в 1. Сигнал, снимаемый с выхода  $Q$   $DD6$ , открывает нижнюю схему И  $DD3$ , и вторая часть кодовой комбинации будет поступать в линию связи с выхода регистра через элемент НЕ  $DD2$ , т.е. контрольные символы повторяют информационные в инверсном виде. После передачи всей кодовой комбинации схема устанавливается в исходное положение девятым тактовым импульсом и готова к кодированию следующего полезного сообщения. Состояния элементов схемы при передаче сообщения  $G(x) = k_4k_3k_2k_1=1110$  указано на схеме рис. 4.14.

Функциональная схема декодера 8-разрядных кодовых комбинаций приведена на рис. 4.15. Кодовая комбинация  $F^*(x)$ , поступающая из линии связи, заносится в регистр  $DD1$ . После чего схемой контроля четности  $DD3$  анализируется первая половина  $(k_1k_2k_3k_4)$  комбинации  $F^*(x)$ . Если в ней четное число единиц, то с выхода  $\Sigma E$   $DD3$  снимается 1, которая открывает верхние схемы 2И элементов  $DD4...DD7$ , и тем самым ко входу сумматоров по модулю 2  $DD8...DD11$  поступают контрольные символы в прямом виде. В случае, если схемой  $DD3$  будет зафиксировано в первой половине комбинации  $F^*(x)$  нечетное число единиц, то сигнал, равный 1, появляется на выходе  $\Sigma 0$ , который откроет нижние схемы 2И элементов  $DD4...DD7$ , и на вход сумматоров по модулю 2  $DD8...DD11$  поступят контрольные символы в инверсном виде с выхода элементов НЕ  $DD2.1...DD2.4$ . Сумматоры  $DD8...DD11$  осуществляют поэлементное сравнение информационного и соответствующего ему контрольного символа. При отсутствии ошибок в комбинации  $F^*(x)$  на выходе всех сумматоров будут нули, а на выходе элемента 4ИЛИ-НЕ появится единица, которая откроет схемы И  $DD14...DD17$ , и информационные символы поступят в приемник. В случае наличия ошибок в принятой комбинации на выходе элемента 4ИЛИ-НЕ появится нуль, который запретит выдачу информации потребителю через элементы  $DD14...DD17$ , а единичный сигнал с выхода формирователя  $DD13$  сбросит приемный регистр  $DD1$  в исходное положение. В результате декодер будет подготовлен к приему следующей кодовой комбинации.

Процесс декодирования кодовой комбинации  $F^*(x) = \dot{0}1\dot{0}00\dot{1}01$  показан на рис. 4.15 в виде состояния элементов декодера. В данном случае на выходе сумматоров по модулю 2  $DD8...DD11$  получим синдром 1110, что свидетельствует о наличии ошибок. В соответствии с этим синдромом на выходе схемы ИЛИ-НЕ  $DD12$  появился сигнал, равный 0, который запрещает вывод информационных символов потребителю, а сигнал с выхода  $DD13$  сбросит в исходное состояние приемный регистр  $DD1$ .

#### 4.7. Кодер и декодер корреляционного кода

Как показано в подразд. 2.2.9, при кодировании в данном коде символ 0 заменяется на 01, а символ 1 – на 10. Данная процедура решается довольно простыми техническими приемами.

Функциональная схема кодера для четырехразрядных сообщений приведена на рис. 4.16.

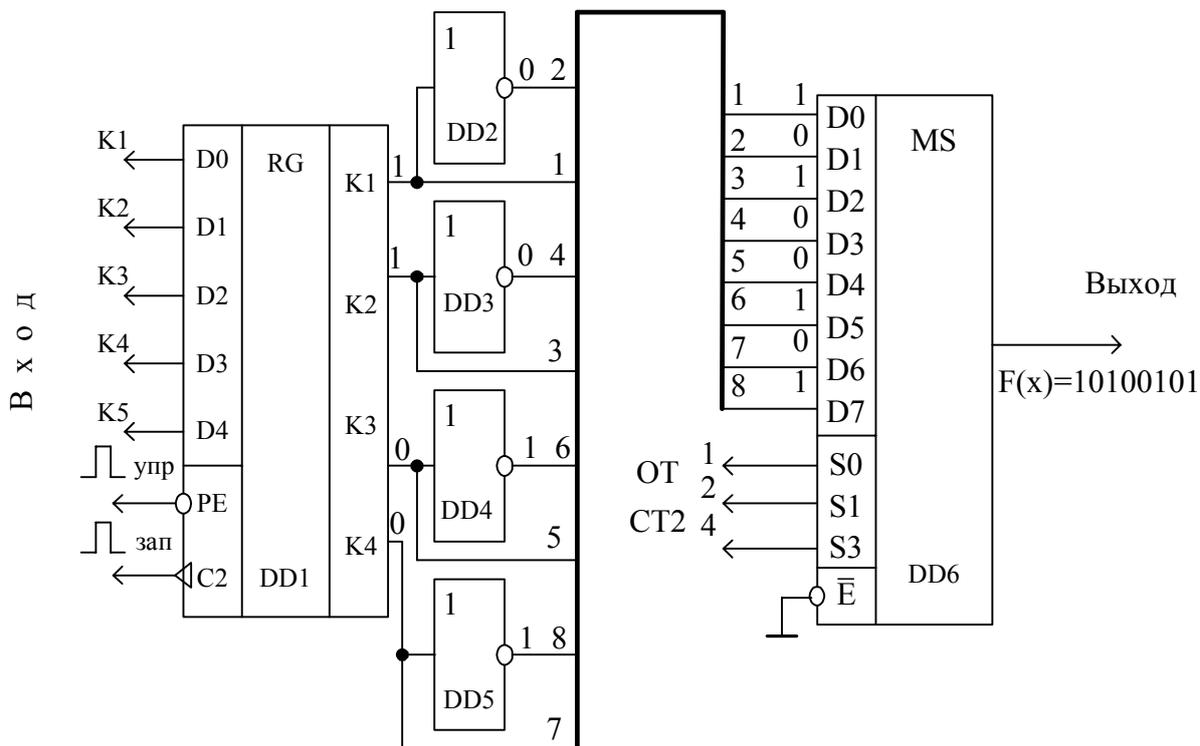


Рис. 4.16. Кодер корреляционного кода

Символы исходного сообщения  $k_1, k_2, k_3$  и  $k_4$  из регистра  $DD1$  поступают на схемы НЕ  $DD2...DD5$  и одновременно на нечетные входы мультиплексора  $DD6$ . Проинвертированные символы поступают на нечетные входы мультиплексора. При поступлении управляющих сигналов на адресные входы мультиплексора  $S_0, S_1, S_2$  от двоичного счетчика он поочередно к выходу подключает входы  $D_0...D_7$ . А так как попеременно чередуются прямые и их инверсные сигналы, то на выходе получаем корреляционный код. Процесс преобразования сообщения  $G(x) = k_1k_2k_3k_4 = 1100$  показан на схеме в виде состояния элементов. На выходе мультиплексора получаем кодовую комбинацию  $F(x)=10100101$ .

Процесс декодирования заключается в поразрядном сравнении двух стоящих рядом символов относящихся к одному и тому же информационному разряду. Учитывая, что при отсутствии искажений один из них равен 0, а другой, соответствующий ему,  $-1$ , в результате сложения по модулю 2 получим единичный синдром, который указывает на отсутствие искажений. Функциональная схема преобразователя приведена на рис. 4.17.

Преобразователь состоит из входного регистра  $DD1$ , в который заносится кодовая комбинация из линии связи; устройства поэлементного сравнения, со-

бранного на элементах “исключающее ИЛИ”  $DD2...DD5$ ; дешифратора синдрома на элементе И  $DD6$ ; схемы управления сбросом регистра на формирователе НЕ  $DD11$ ; устройства вывода на элементах И  $DD7...DD10$ . При отсутствии ошибок в принятом сообщении на выходе дешифратора синдрома появляется 1, которая разрешает вывод информационных символов через элементы  $DD7...DD10$  потребителю.

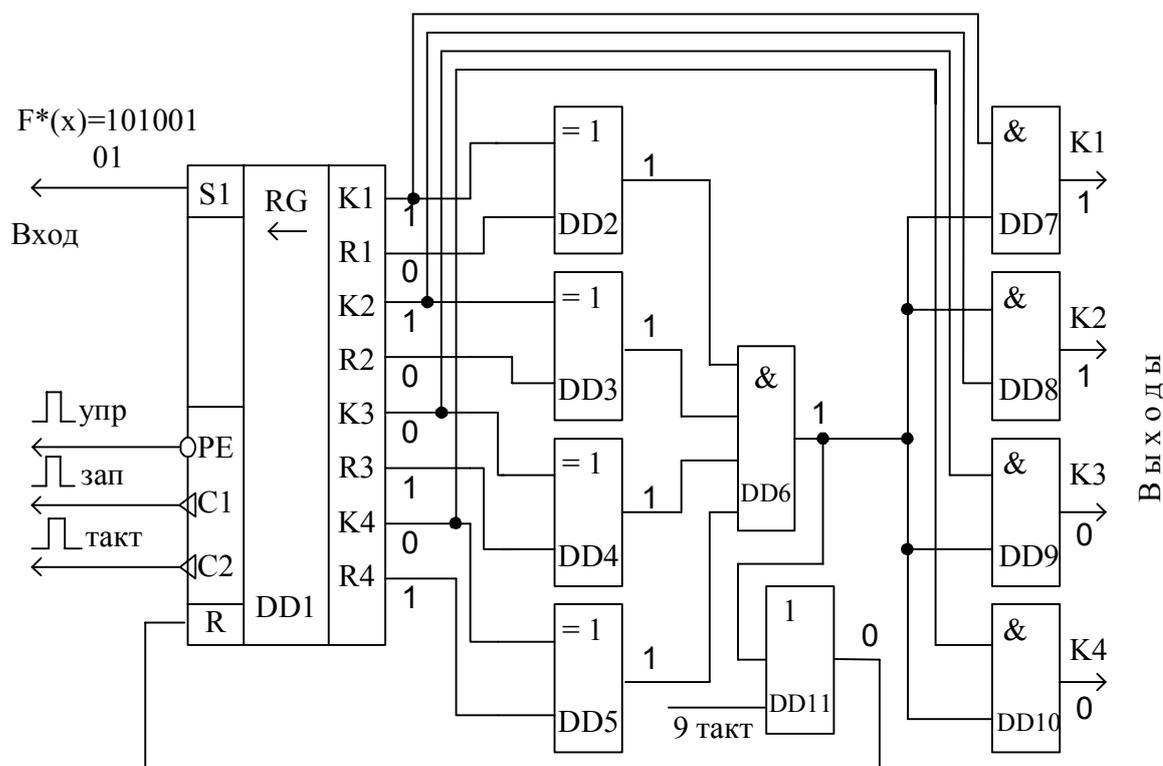


Рис. 4.17. Декодер корреляционного кода

Если в принятой комбинации  $F^*(x)$  имеются ошибки, то на выходе хотя бы одной схемы сумматора по модулю 2 будет 0, который приведет к закрытию схем И  $DD7...DD8$ , что запретит вывод информационных символов, а единичный сигнал с выхода  $DD11$  установит приемный регистр в исходное состояние. Пример дешифрации кодовой комбинации  $F^*(x) = 10100101$  показан на схеме в виде состояния элементов. В данном случае дешифратор синдрома не зарегистрировал ошибок и к потребителю поступила кодовая комбинация 1100, которая соответствует переданному сообщению (см. рис. 4.16).

#### 4.8. Кодер и декодер кода Бергера

Функциональная схема кодера приведена на рис. 4.18. В состав кодирующего устройства входят: входной регистр  $DD1$ , предназначенный для хранения преобразуемых сообщений, счетчик  $DD5$  для подсчета числа единиц в ис-

ходном сообщении и преобразователь  $DD2$  параллельного кода в последовательный.

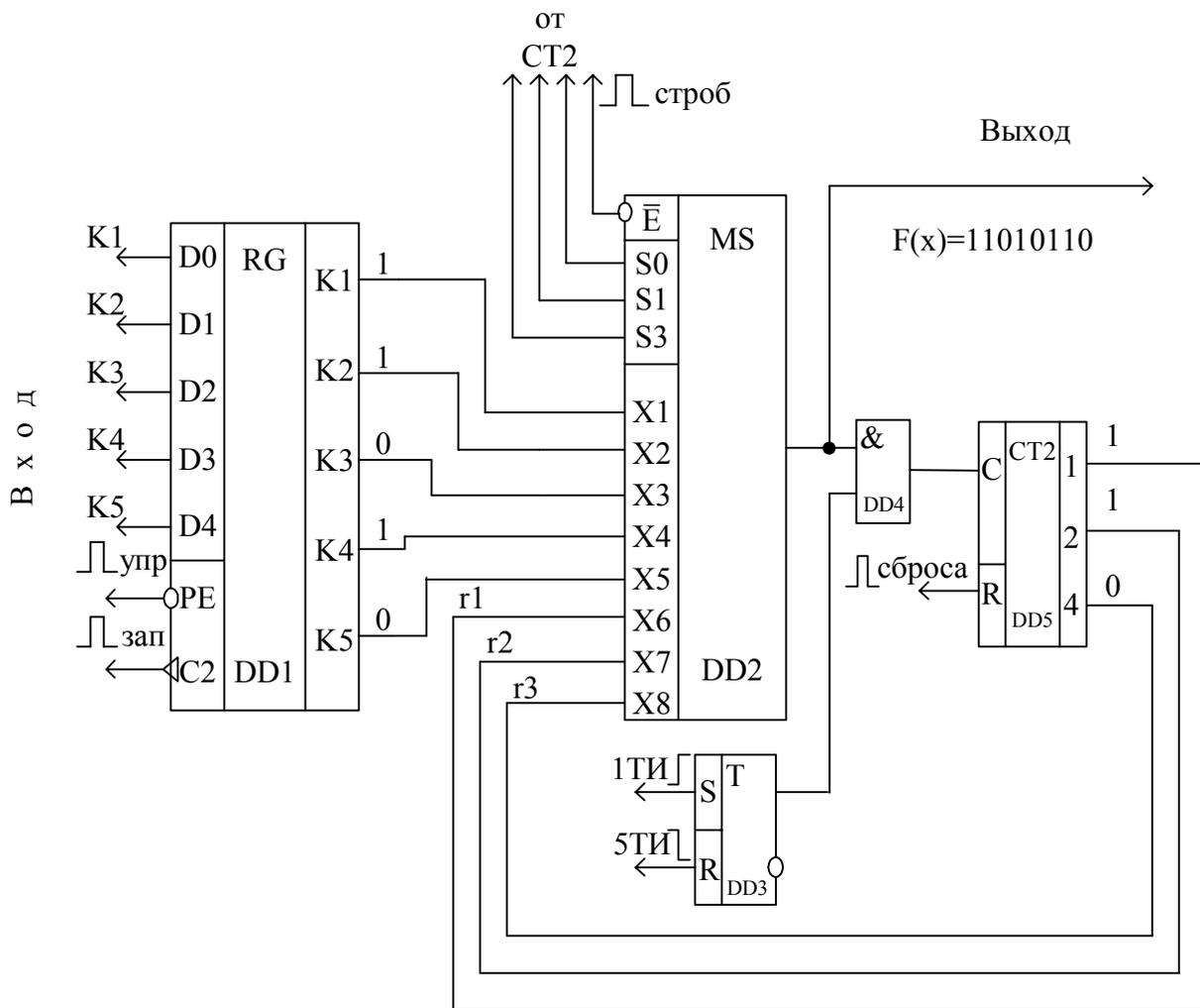


Рис. 4.18. Кодер кода Бергера для  $k=5$

Исходная кодовая комбинация, представляющая, как правило, двоичный избыточный код, через мультиплексор  $DD2$  поступает на выход и одновременно через схему И  $DD4$  на вход счетчика  $DD5$ , который в данном случае подсчитывает число единиц в передаваемом сообщении. После прохождения информационных  $k$  символов спадом пятого тактового импульса триггер  $DD3$  устанавливается в 0, схема И  $DD4$  закрывается и контрольные символы из счетчика  $DD5$  через мультиплексор  $DD2$  поступают в линию связи.

Процесс кодирования кодовой комбинации  $G(x) = 11010$  в виде состояния элементов кодера показан на схеме рис. 4.18.

Декодирование сводится к определению числа единиц в информационной части, т.е. к формированию контрольных символов из пришедших на приемную сторону информационных символов, с последующим сравнением этой

последовательности контрольных символов с контрольными символами, поступившими из линии связи. В случае их совпадения, что говорит об отсутствии ошибок, информационные символы поступают потребителю.

Схема декодера приведена на рис. 4.19.

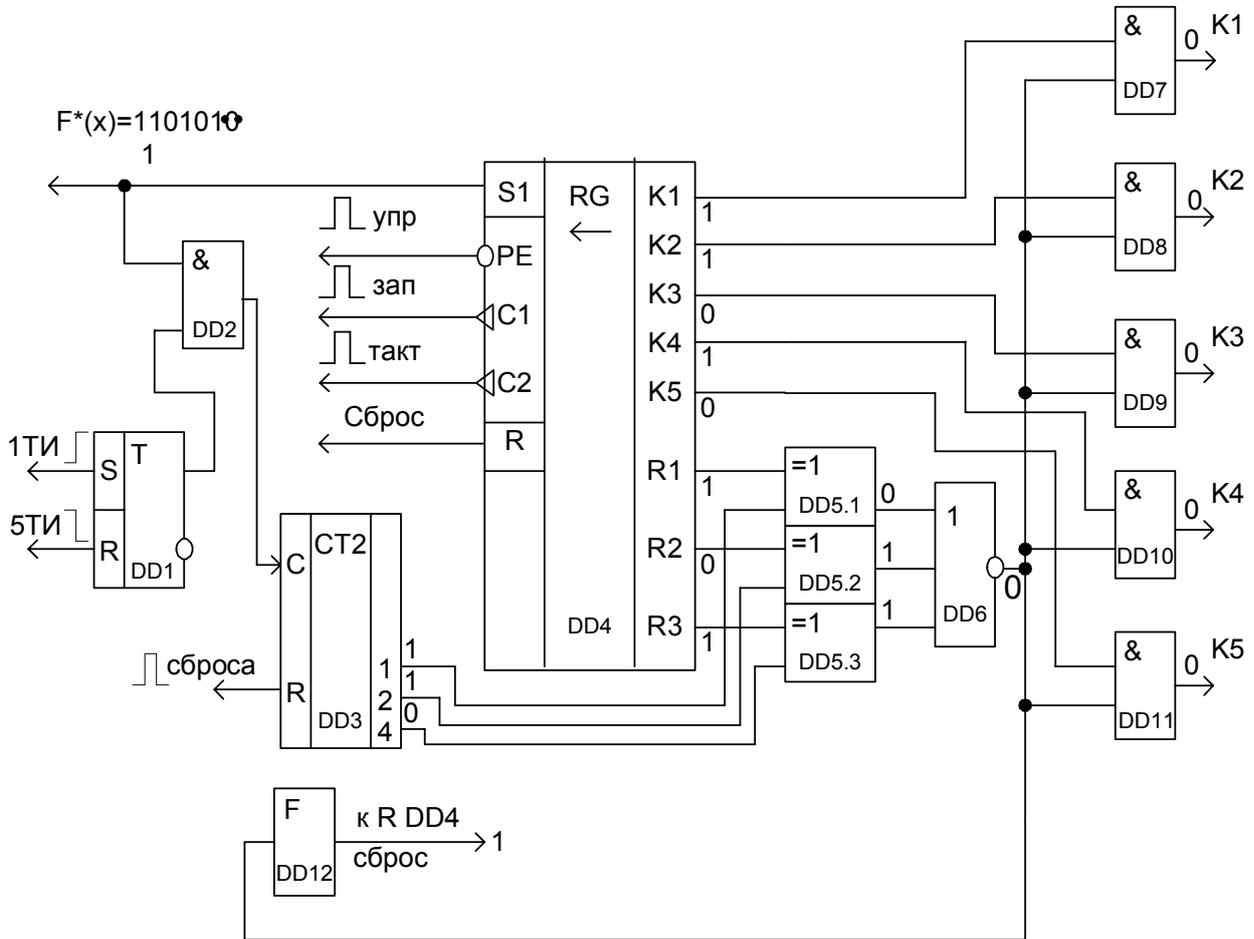


Рис. 4.19. Декодер кода Бергера

Кодовая комбинация в коде Бергера записывается в регистр  $DD4$ . На первых пяти тактах счетчиком  $DD3$  подсчитывается число единиц в информационных символах. После этого сумматорами по модулю 2  $DD5.1 \dots DD5.3$  складываются две последовательности контрольных символов, записанных в регистре  $DD4$  и зафиксированных счетчиком  $DD3$ . При полном их совпадении что говорит об отсутствии ошибок, на выходе ИЛИ-НЕ  $DD6$  появляется 1, которая открывает элементы И  $DD7 \dots DD11$ , и информационные символы поступают потребителю. В случае несоответствия двух последовательностей контрольных символов на выходе формирователя  $DD12$  появляется 1, которая сбрасывает приемный регистр  $DD4$  в исходное положение, а сигнал, равный 0, на выходе  $DD6$  запрещает выдачу информационных символов потребителю. Процесс декодирования кодовой комбинации  $F^*(x) = 1101010\dot{1}$  показан на схеме в виде

состояния элементов. Элементами  $DD5.1 \dots DD5.3$  зафиксировано несовпадение двух последовательностей, в результате чего комбинация бракуется.

#### 4.9. Кодирующее и декодирующее устройства систематического кода

Как следует из подразд. 2.3.1, алгоритм кодирования и декодирования определяется составом образующей и проверочной матриц. Схема кодера, использующего алгоритм, полученный по выражениям: (2.12)  $a_5 = a_2 \oplus a_3 \oplus a_4$ , (2.13)  $a_6 = a_1 \oplus a_2 \oplus a_4$ , (2.14)  $a_7 = a_1 \oplus a_3 \oplus a_4$ , приведена на рис. 4.20.

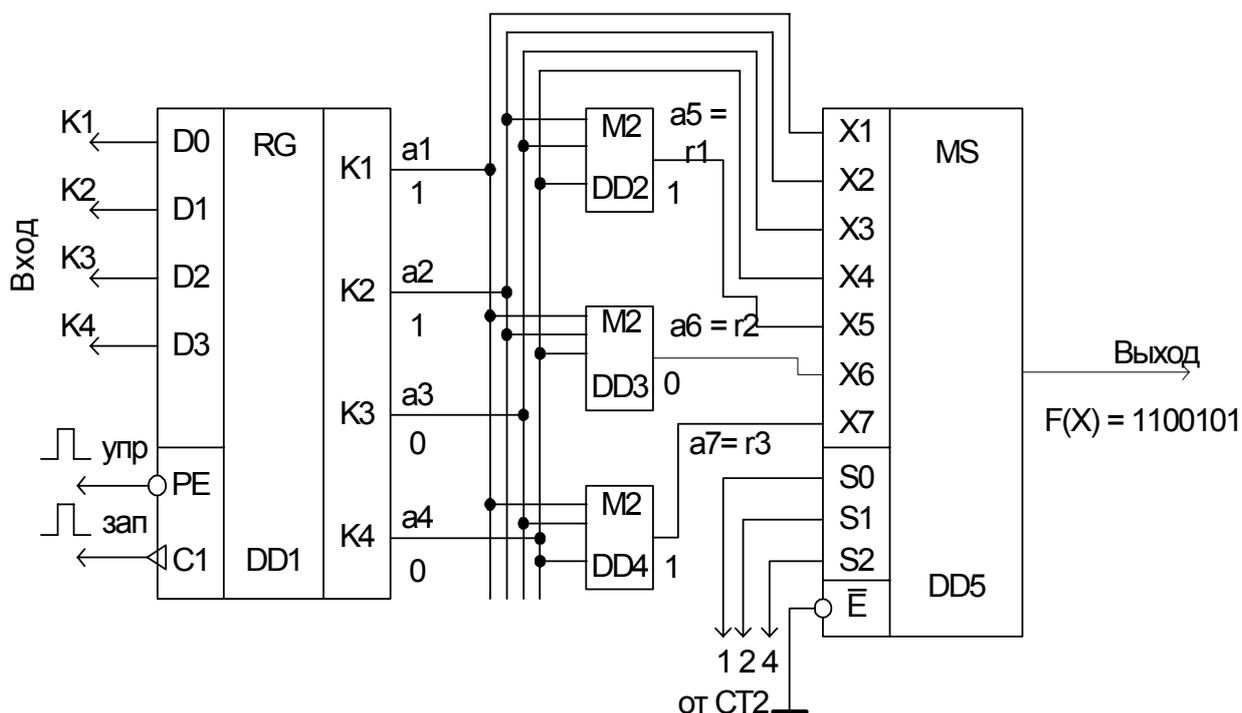


Рис. 4.20. Кодер систематического кода исправляющего одиночные ошибки

Сумматорами по модулю два  $DD2$ ,  $DD3$ ,  $DD4$  формируются контрольные символы  $r_1$ ,  $r_2$  и  $r_3$  в соответствии с выражениями (2.12)...(2.14). Преобразователем параллельного кода в последовательный  $DD5$  сначала в линию связи выдвигаются информационные символы  $k_1, k_2, k_3, k_4$ , а затем контрольные  $r_1, r_2, r_3$ . На этом процесс кодирования данной комбинации заканчивается, и кодер ожидает поступления в регистр  $DD1$  следующего кодового сообщения. Порядок формирования комбинации  $F(x)$  показан на схеме в виде состояния элементов кодера для сообщения  $G(x) = 1100$ .

Декодирование заключается в определении синдрома по выражениям (2.17):

$$\begin{aligned} S_1 &= a_2 \oplus a_3 \oplus a_4 \oplus a_5, \\ S_2 &= a_1 \oplus a_2 \oplus a_4 \oplus a_6, \\ S_3 &= a_1 \oplus a_3 \oplus a_4 \oplus a_7. \end{aligned}$$

Если синдром будет нулевого порядка, то ошибок нет, в противном случае синдром должен указать номер искаженного разряда.

Декодер систематического (7,4) кода показан на рис. 4.21.

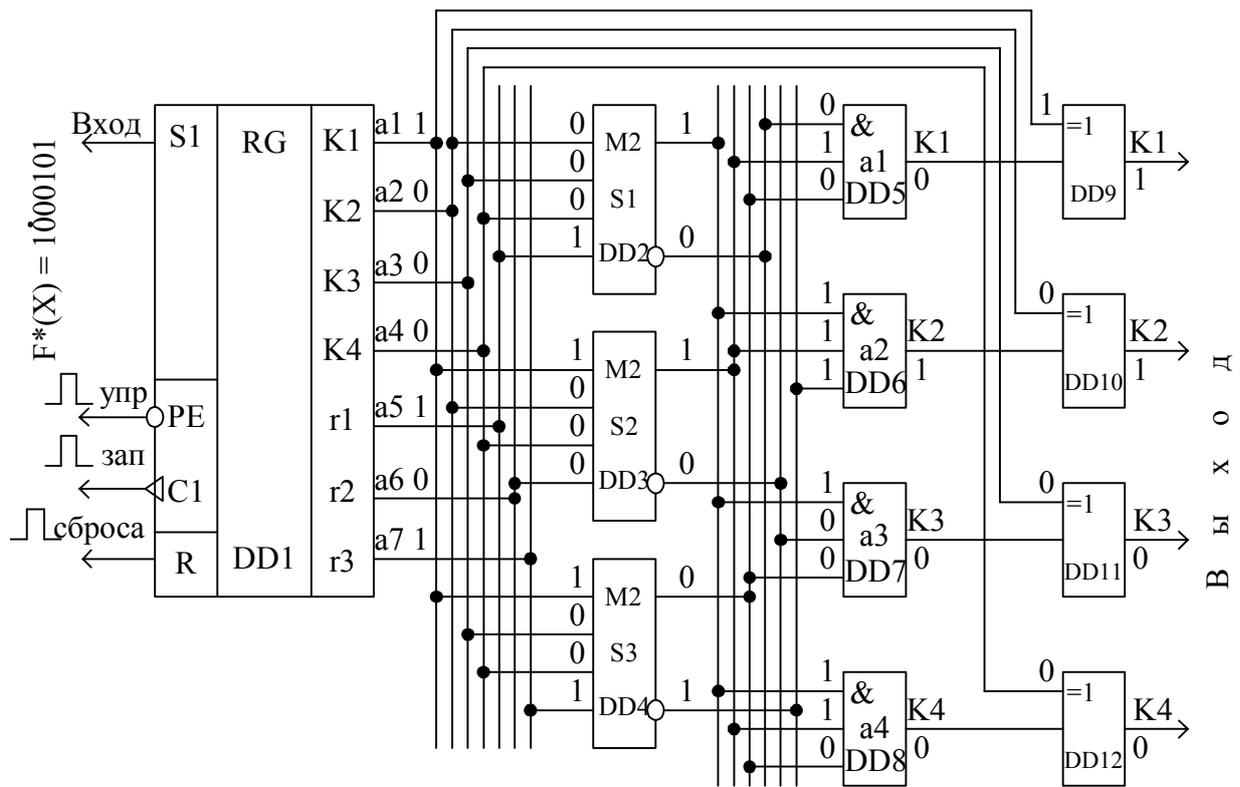


Рис. 4.21. Декодер систематического кода (7, 4) для исправления одиночных ошибок

Кодовая комбинация из линии связи записывается в регистр  $DD1$  и поступает на входы определителя синдрома, собранного на элементах  $DD2$ ,  $DD3$ ,  $DD4$ . На сумматоре по модулю 2  $DD2$  осуществляется вычисление  $S1$ , на  $DD3$  –  $S2$ , на  $DD4$  –  $S3$ . Далее сигналы с определителя синдрома  $S_1S_2S_3$  поступает на дешифратор синдрома, собранный на элементах И  $DD5$ ,  $DD6$ ,  $DD7$ ,  $DD8$ , каждый из которых настроен на кодовую комбинацию (2.18), соответствующую одному из информационных символов  $a_1=k_1$ ,  $a_2=k_2$ ,  $a_3=k_3$  и  $a_4=k_4$ . Дешифратора синдрома для контрольных символов в схеме не предусмотрено, так как исправление контрольных символов не влияет на информацию, поступающую потребителю. При отсутствии ошибок на выходах всех элементов И  $DD5...DD8$  будут 0, которые не влияют на вывод информационных символов. При наличии ошибки в одном из информационных символов комбинации  $F^*(x)$  на выходах соответствующего элемента И  $DD5...DD8$  появится единичный сигнал, который при прохождении информационного символа через выходной сумматор по модулю 2  $DD9...DD12$  изменит его на противополож-

ный. Процесс декодирования кодовой комбинации  $F^*(x) = 1000101$  показан на схеме в виде состояния отдельных элементов. В данном случае зафиксировано искажение символа  $k_2$ , который скорректирован выходным сумматором по модулю 2  $DD10$  с 0 на 1.

#### 4.10. Кодирующее и декодирующее устройство кода Хемминга

Принцип построения кодирующего устройства не зависит от числа информационных разрядов передаваемого кода. Поэтому рассмотрим схему кодирующего устройства (рис. 4.22) для числа информационных символов  $k = 4$ , контрольных символов  $r = 4$  и  $d = 4$ , хотя она без принципиальных изменений может быть использована для кодирования любого числа  $k$  за счет увеличения числа сумматоров по модулю 2 и числа входов отдельных элементов.

Кодер состоит из входного регистра  $DD1$ , куда записываются комбинации, подлежащие кодированию; формирователя контрольных символов на элементах  $DD2...DD5$  и преобразователя параллельного кода в последовательный на мультиплексоре  $DD6$ .

В соответствии с методикой формирования контрольных символов, изложенной в подразд.2.3.2, можно записать, что  $r_1 = k_4 \oplus k_3 \oplus k_1$ ,  $r_2 = k_4 \oplus k_2 \oplus k_1$ ,  $r_3 = k_3 \oplus k_2 \oplus k_1$ ,  $r_4 = k_4 \oplus k_3 \oplus k_2 \oplus k_1 \oplus r_1 \oplus r_2 \oplus r_3$ . Согласно этим выражениям осуществляем подключение входов сумматоров по модулю 2 к информационным шинам  $k_1...k_4$ . Порядок подачи информационных и контрольных символов на вход мультиплексора, а следовательно, и очередность их передачи в линию связи может быть различна: сначала информационные, а потом контрольные или наоборот, или классический вариант – на местах, кратных  $2^i$ , где  $i = 0, 1, 2 \dots r$ , контрольные, а на остальных – информационные.

Порядок кодирования комбинации  $G(x) = k_4k_3k_2k_1 = 1011$  показан на рис. 4.22 в виде состояния элементов. В результате в линию связи поступит кодовая комбинация  $F(x) = 01100110$  с классическим порядком следования контрольных и информационных символов.

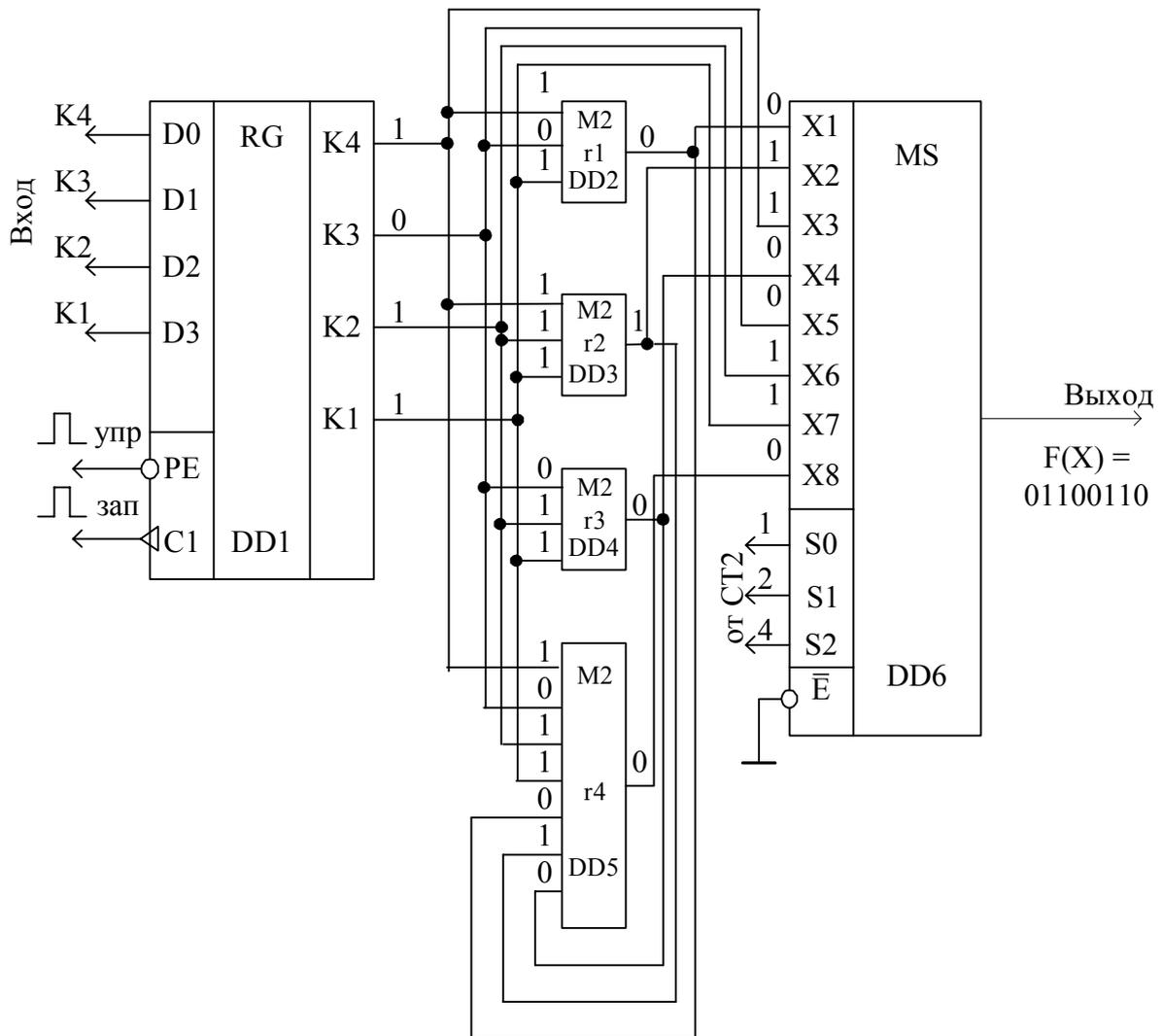


Рис. 4.22. Кодер кода Хемминга с  $d = 4$

Декодирование заключается в нахождении ошибок, их исправлении и выводе полезной информации потребителю. Схема декодера для кода (8,4), позволяющего исправлять одиночные и обнаруживать двойные ошибки, приведена на рис. 4.23. Декодер состоит из входного регистра  $DD1$ , определителя синдрома  $S_1S_2S_3$  на элементах  $DD2...DD4$ , определителя общей проверки на четность  $S_\Sigma$  на элементе  $DD15$ , дешифратора синдрома  $S_1S_2S_3$  на элементе  $DD5$ , дешифратора двойной ошибки на элементе И  $DD6$ , устройства коррекции ошибок на элементах "исключающее ИЛИ"  $DD7...DD10$  и устройства вывода на элементах И  $DD11...DD14$ .

Входы определителя синдрома  $DD2...DD4$  подключаются в соответствии с принятым алгоритмом кодирования. Для кода (8,4) сумматор по модулю 2  $DD2$  осуществляет проверку  $S_1 = r_1 \oplus k_4 \oplus k_3 \oplus k_1$ , сумматор  $DD3$  —

$S_2 = r_2 \oplus k_4 \oplus k_2 \oplus k_1$ , сумматор  $DD4 - S_3 = r_3 \oplus k_3 \oplus k_2 \oplus k_1$ . Общая проверка на четность принятой кодовой комбинации производится сумматором  $DD15 - S_y = r_1 \oplus r_2 \oplus k_4 \oplus r_3 \oplus k_3 \oplus k_2 \oplus k_1 \oplus r_4$ . Дешифратор синдрома представляет обычный дешифратор двоичного кода 4–2–1 в десятичный.

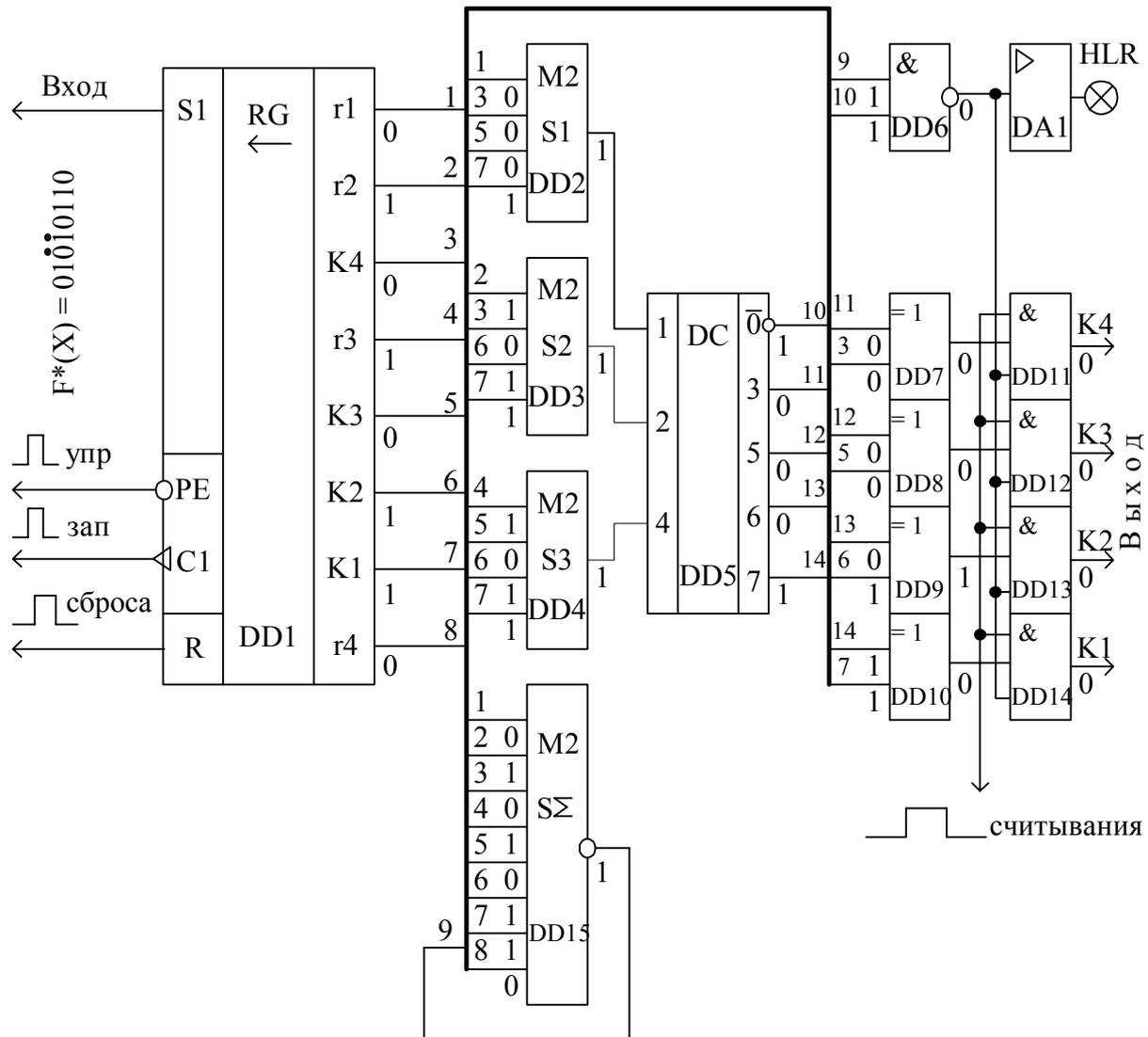


Рис. 4.23. Декодер кода Хемминга (8,4), позволяющего исправлять одиночные ошибки и обнаруживать двойные

В данном дешифраторе  $DD5$  показаны прямые выходы, соответствующие информационным символам, и инверсный выход  $\bar{0}$ , на котором нулевой сигнал появляется только в случае, когда  $S_1 = 0, S_2 = 0, S_3 = 0$ ; выходы, соответствующие контрольным символам, не показаны, так как их коррекция не производится. На выходе дешифратора двойной ошибки, элементе 2И-НЕ  $DD6$ , сигнал 0 (запрета) появляется только в том случае, когда на инверсном выходе

$DD15$  будет 1 и на выходе  $\bar{0}$  дешифратора  $DD5$  тоже будет 1. Этот сигнал поступает на один из входов схем И  $DD11 \dots DD14$  и запрещает выдачу информации потребителю.

При всех других соотношениях  $S_i$  и  $S_\Sigma$ , указанных в подразд. 2.3.2, на выходе И  $DD6$  будет сигнал, равный 1.

Процесс декодирования кодовой комбинации  $F^*(x) = 010\dot{1}0110$  показан на схеме в виде состояния элементов. В данном случае на выходе схемы И  $DD6$  будет 0, а следовательно, схемы И  $DD11 \dots DD14$  будут закрыты, информация потребителю не поступит и будет включен индикатор  $HLR$ , свидетельствующий о двойной ошибке.

#### 4.11. Технические средства умножения и деления многочлена на многочлен

Устройства для умножения и деления многочлена на многочлен составляют основу кодирующих и декодирующих устройств циклических кодов. Эти устройства строятся на базе регистров сдвига с обратными связями и сумматоров с приведением коэффициентов по модулю 2. Такие регистры также называют многотактными линейными переключательными схемами и линейными кодовыми фильтрами Хаффмана.

Основные правила построения схем умножителей и делителей:

1) число ячеек регистра равно старшей степени многочлена, на который происходит умножение или деление. Ячейка регистра для старшей степени многочлена отсутствует, но всегда присутствует ячейка  $x^0$ ;

2) число сумматоров на единицу меньше числа ненулевых членов многочлена, на который производится умножение или деление, или на единицу меньше его веса;

3) при делении отбрасывается сумматор, соответствующий старшему члену многочлена, а при умножении – младшему;

4) сумматоры устанавливаются перед ячейками регистра, соответствующими ненулевым членами многочлена тех же степеней;

5) при умножении множимое подается одновременно на вход и на все сумматоры;

6) при делении делимое подается только на первый сумматор, а частное – на выход и на все сумматоры;

7) множимое или делимое поступает на вход начиная со старшего разряда.

Функциональная схема делителя на многочлен  $P(x) = x^4 + x^3 + 1$  приведена на рис. 4.23.

Разделим на этот многочлен (делитель) многочлен  $G(x) = x^7 + x^5 + x^4 + x^3 + x^1 + 1$  (делимое). Результат деления представлен в виде табл. 4.2, где стрелками показаны направления процессов между элементами.

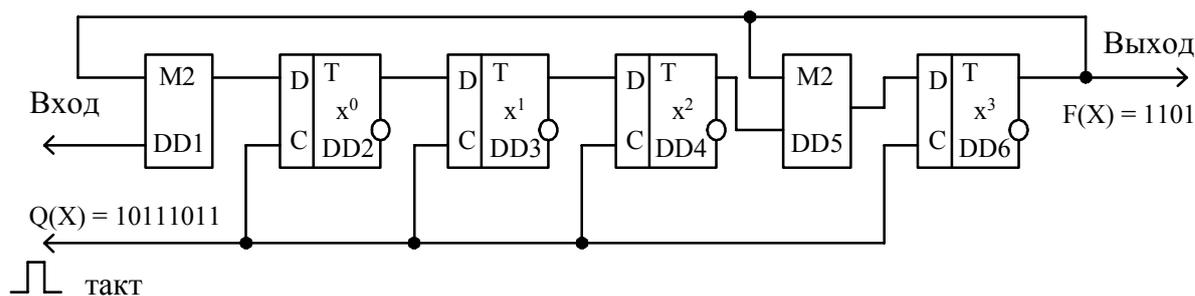


Рис. 4.24. Схема для деления на многочлен  $P(x) = x^4 + x^3 + 1$

Из табл. 4.2 следует, что в такте 1 единица старшего разряда делимого записывается в ячейку  $DD2$ , в такте 2 эта единица считывается с ячейки  $DD2$  и записывается в ячейку  $DD3$  (косая стрелка из ячейки  $DD2$  в ячейку  $DD3$ ). Одновременно нуль делимого записывается в ячейку  $DD2$ , а нули из ячеек  $DD3$  и  $DD4$  переходят соответственно в ячейки  $DD4$  и  $DD6$ , что также показано косыми стрелками. Нуль из ячейки  $DD6$  появляется на выходе.

Таблица 4.2

### Деление многочленов

Номер такта	Вход	Состояние ячеек регистра				Частное (Выход)
		DD2	DD3	DD4	DD6	
0		0	0	0	0	
1	1	1	0	0	0	0
2	0	0	1	0	0	0
3	1	1	0	1	0	0
4	1	1	1	0	1	0
5	1	0	1	1	1	1
6	0	1	0	1	0	1
7	1	1	1	0	1	0
8	1	0	1	1	1	1

Частное

В тактах 3 и 4 ячейки регистра продолжают заполняться, но на выход пока поступают только сигналы 0. Вследствие того что с ячейки  $DD6$  сигнал 1 через сумматор  $DD1$  поступает в ячейку  $DD2$  одновременно с 1 делимого, в ней записывается 0 (такт 4). В этом же такте на выходе появляется 1 и через сумматор  $DD5$  происходит запись 1 в ячейку  $DD6$  (обратная связь с ячейки  $DD6$  на ячейку  $DD2$  регистра и на ту же ячейку  $DD6$  показана косыми стрелками влево и вниз). В такте 6, хотя на вход поступает 0 делимого, по обратной связи с ячейки  $DD6$  в ячейку  $DD2$  записывается 1. Однако из-за той же обратной связи в ячейке  $DD6$  происходит запись 0, так как сумматор  $DD5$  не пропустил два сигнала 1.

Заполнение ячеек регистра в такте 7 происходит без обратной связи, которая вновь сказывается в такте 8. Частное читается сверху вниз. Остатки от деления начинают записываться в ячейки регистра начиная с такта 5. Последний остаток  $R(x) = 1110$  записан в такте 8.

На рис. 4.25 изображена схема умножителя на многочлен  $P(x) = x^4 + x^3 + 1$ . Рассмотрим процесс умножения многочлена  $G(x) = x^7 + x^5 + x^4 + x^3 + x + 1$  на данный многочлен  $P(x)$ . Процесс умножения представлен в табл. 4.3.

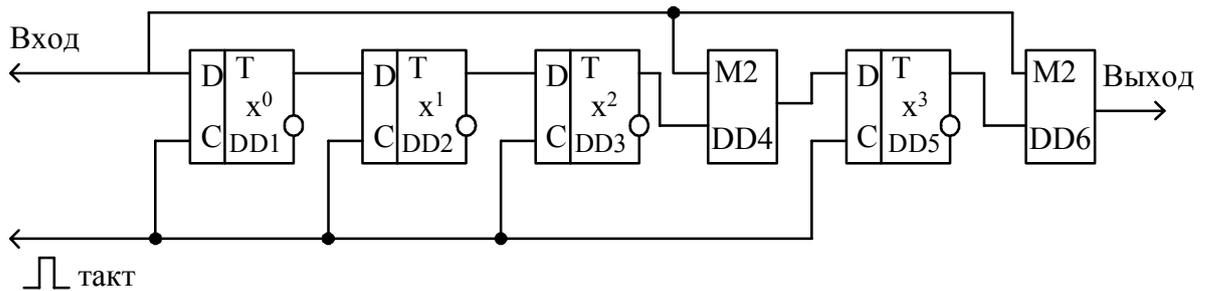


Рис. 4.25. Схема для умножения на многочлен  $P(x) = x^4 + x^3 + 1$

В такте 1 единица старшего разряда записывается одновременно в ячейки  $DD1$ ,  $DD5$  и поступает на выход. В такте 2 на выход проходит сигнал 1 с ячейки  $DD5$ , а с ячейки  $DD1$  единица переходит в ячейку  $DD2$ . В такте 3 сигнал 1 записывается с ячейки  $DD1$  и  $DD5$  и проходит на выход, а сигнал 1 с ячейки  $DD2$  переходит в ячейку  $DD3$ . В такте 4 сигнал 1 записывается только в ячейку  $DD1$ , но на выход он не проходит и не записывается в ячейку  $DD5$ . Этому препятствуют сигналы 1 с ячеек  $DD3$  и  $DD5$ . Начиная с такта 9 информация в регистр не поступает и регистр очищается, т.е. информация, записанная в такте 8, такт за тактом подается на выход. Результат умножения  $F(x) = 111011010011$  читается сверху вниз.

Таблица 4.3

## Умножение многочленов

Номер такта	Вход (Множимое)	Состояние ячеек регистра				Выход (произведение)
		DD1	DD2	DD3	DD5	
0		0	0	0	0	
1	1	1	0	0	1	1
2	0	0	1	0	0	1
3	1	1	0	1	1	1
4	1	1	1	0	0	0
5	1	1	1	1	1	1
6	0	0	1	1	1	1
7	1	1	0	1	0	0
8	1	1	1	0	0	1
9		0	1	1	0	0
10		0	0	1	1	0
11		0	0	0	1	1
12		0	0	0	0	1

## 4.12. Кодер и декодер циклического кода

Как указывалось в подразд.2.3.3, образование циклического кода состоит из двух операций: умножение комбинации обычного двоичного кода  $G(x)$  на одночлен  $X^r$  и последующего деления этого произведения на выбранный образующий многочлен  $P(x)$ . Полученные в остатке от деления контрольные символы приписываются к кодируемой комбинации (2.29). Таким образом, кодирующее устройство должно совмещать функции умножения и деления.

Схема кодирующего устройства циклического кода (7,4), образованного с помощью генераторного полинома  $P(x) = x^3 + x + 1 = 1011$ , показана на рис. 4.26.

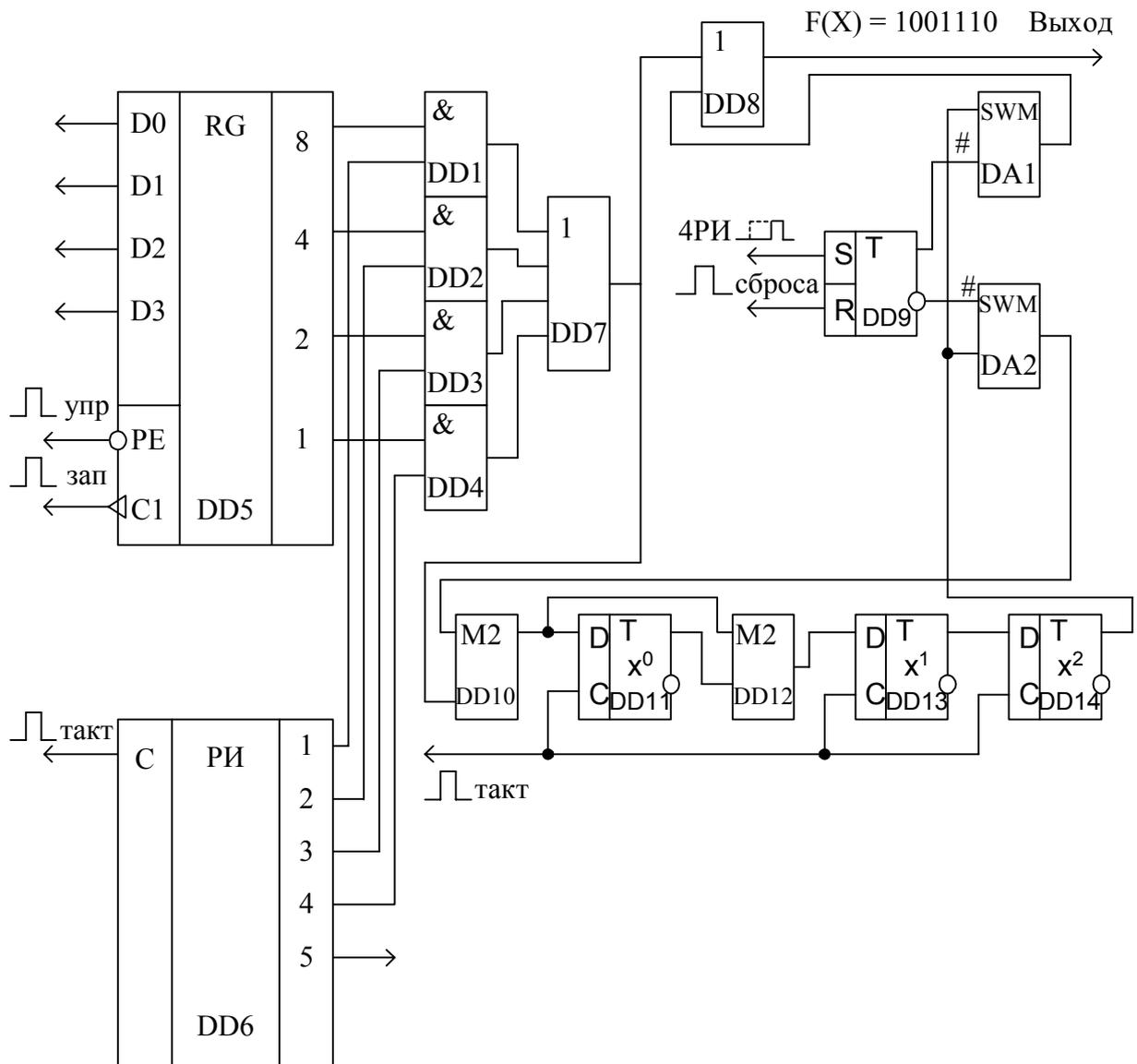


Рис. 4.26. Кодер циклического кода (7,4)

В состав его входит  $r$ -разрядный регистр сдвига ( $DD11, DD13, DD14$ ), который совместно с сумматорами по модулю 2  $DD10, DD12$  осуществляет деление на полином  $P(x)$ , два ключа  $DA1, DA2$ , входной регистр  $DD5$ , для записи  $G(x)$ , коммутатор входных сообщений ( $DD1 \dots DD4, DD6$ ) и триггер управления  $DD9$ .

Схема работает следующим образом. В начале работы ключ  $DA2$  замкнут сигналом 1 с инверсного выхода триггера  $DD9$ . Информационная последовательность под действием управляющих сигналов с распределителя импульсов  $DD6$  через схемы И  $DD1 \dots DD4$  начиная со старшего разряда поступает на вы-

ход и входной сумматор DD10. В процессе ее прохождения за  $k$  тактов в ячейках регистров сдвига DD11, DD13, DD14 накапливается  $r$  проверочных разрядов. После 4-го такта ключ DA2 закрывается, а ключ DA1 открывается. Записанные в ячейках регистра  $r = 3$  проверочных разрядов тремя тактами поступают на выход кодирующего устройства.

Процесс кодирования входного сообщения  $G(x)=1001$  с помощью схемы рис. 4.26 показан в табл. 4.4.

Таблица 4.4

Образование циклического кода

Номер такта	Вход	Состояние ячеек			Выход
		DD11	DD13	DD14	
					после четвертого такта
0		0	0	0	
1	1	1	1	0	1
2	0	0	1	1	0
3	0	1	1	1	0
4	1	0	1	1	1
5		0	0	1	1
6		0	0	0	1
7		0	0	0	0

Декодирование комбинаций циклического кода можно проводить различными методами, существуют методы (см. подразд. 2.3.3), основанные на использовании рекуррентных соотношений, на мажоритарном принципе, на вычислении остатка от деления принятой комбинации на образующий многочлен кода и др. Целесообразность применения каждого из них зависит от конкретных характеристик используемого кода.

Рассмотрим сначала устройства декодирования, в которых для обнаружения и исправления ошибок производится деление произвольного многочлена  $F^*(x)$ , соответствующего принятой комбинации, на образующий многочлен кода  $P(x)$ . В этом случае при декодировании могут использоваться те же регистры сдвига, что и при кодировании.

Декодер для обнаружения ошибок показан на рис. 4.27. В состав декодера входят: буферный регистр на  $k$  разрядов  $DD1$ , декодирующий регистр на элементах  $DD3$ ,  $DD5$ ,  $DD6$  с сумматорами по модулю два  $DD2$ ,  $DD4$ , схема которого подобна схеме кодера, схема ИЛИ–НЕ  $DD7$  и схемы вывода информации на элементах И  $DD8...DD11$ .

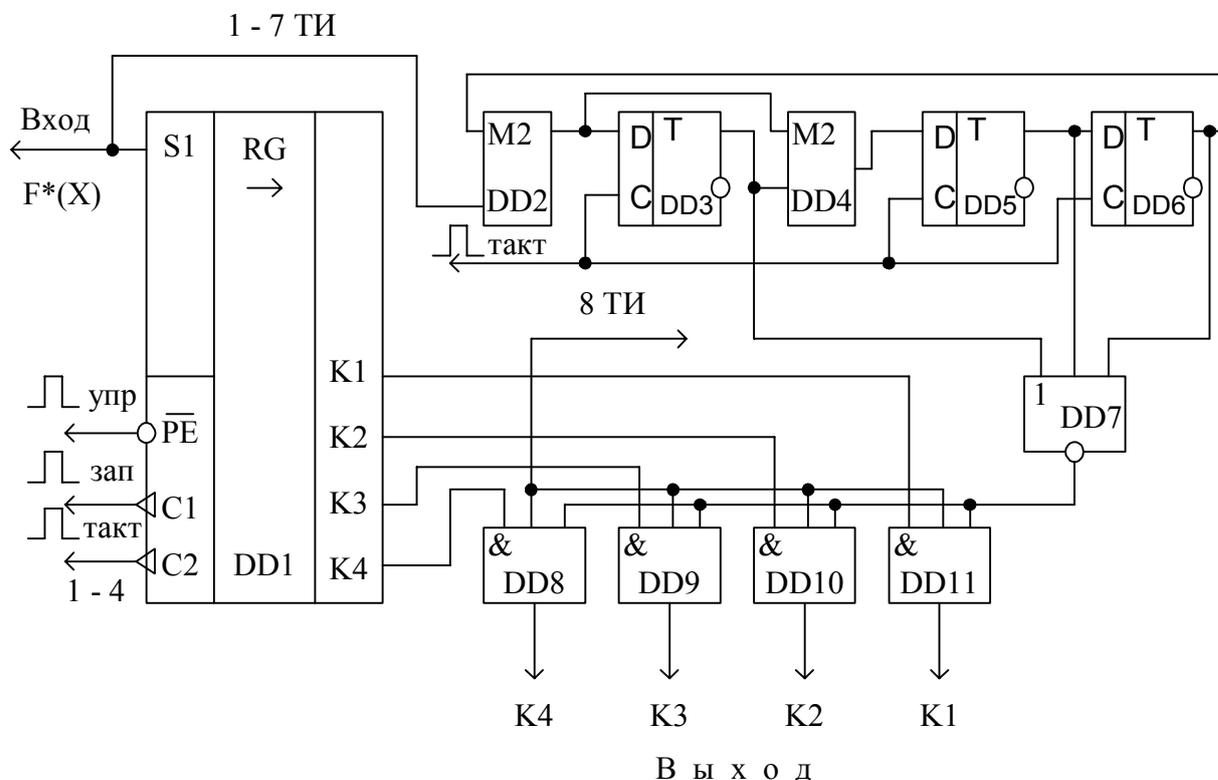


Рис. 4.27. Декодер циклического кода (7,4) для обнаружения ошибок

Принимаемая последовательность записывается в ячейки буферного регистра  $DD1$  на первых четырех тактах и одновременно поступает в декодирующий регистр на 1–7–м тактах. Таким образом, в регистре  $DD1$  оказываются лишь информационные разряды. На 8–м такте после приема последнего разряда кодовой комбинации, открываются схемы И  $DD8...DD11$ . Если комбинация принята без ошибок, то в ячейках декодирующего регистра будут записаны нули, а на выходе схем ИЛИ–НЕ появится 1, которая дает разрешение на вывод информационных разрядов  $k_1...k_4$  через схемы И  $DD8...DD11$  потребителю. Наличие же в тех или иных ячейках декодирующего регистра единиц свидетельствует об ошибках в принимаемой информации. На выходе схемы ИЛИ–НЕ  $DD7$  в таком случае появляется сигнал 0, который запрещает вывод информации из буферного регистра.

Если декодер предназначен для исправления ошибок, то необходимо указать местоположение ошибочных разрядов. С этой целью в состав декодера вместо схемы ИЛИ включают дешифратор синдрома, вырабатывающий на своем выходе сигнал 1 при фиксации в ячейках декодирующего регистра комбинаций определенного вида. Последние выбираются с таким расчетом, чтобы момент их появления в ячейках декодирующего регистра совпадал с моментом прохождения ошибочного разряда через выходной сумматор, включенный на выходе буферного регистра. При этом ошибочный разряд, проходя через выходной сумматор, меняет знак на обратный (т.е. исправляется).

Проиллюстрируем принцип построения декодеров для исправления ошибок на примере кода (7,4) с образующим полиномом  $P(x) = x^3 + x + 1$ . Минимальное кодовое расстояние  $d_{\min} = 3$ , следовательно, код способен исправлять однократные ошибки. Укажем, что дешифратор синдрома должен быть настроен на комбинацию 001.

Схема декодера показана на рис. 4.28.

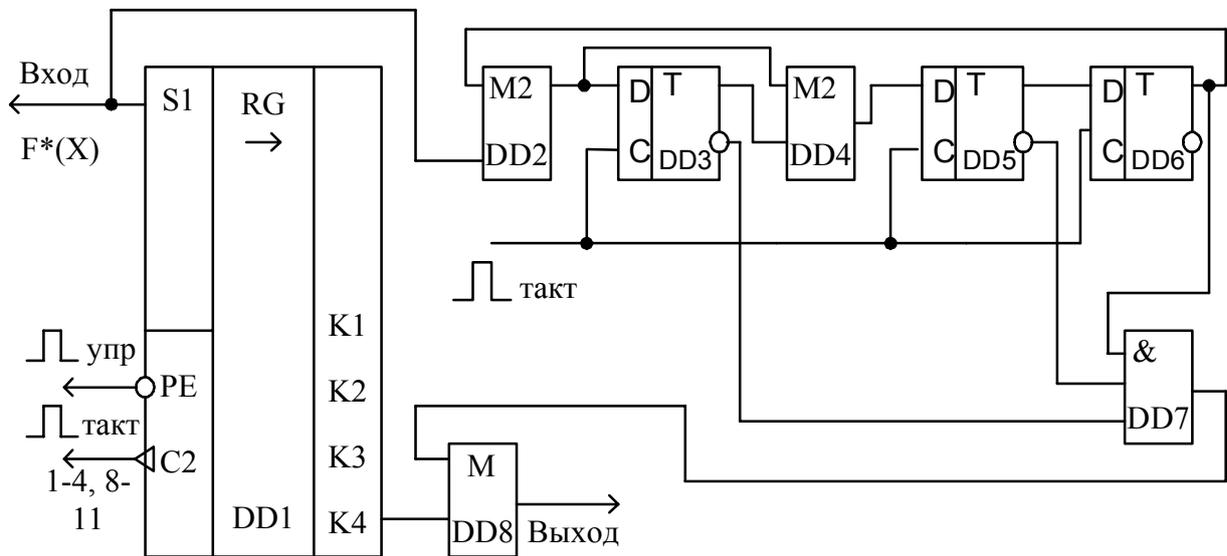


Рис. 4.28. Декодер циклического кода (7,4) для исправления одиночной ошибки

Пусть по каналу связи была передана комбинация  $F(x) = 1001110$ , которая под действием помех приняла вид  $F^*(x) = 10\dot{1}1110$ . Процесс исправления ошибки представлен в табл. 4.5.

На 7-м такте в ячейках декодирующего регистра завершается формирование комбинации синдрома. Комбинация 100 отлична от нуля, что свидетельствует о наличии ошибки. Далее в буферный и декодирующий регистры подается еще  $k = 4$  тактов, которые, во-первых, выдвигают информационные разряды через выходной сумматор DD8 на выход декодера, а во-вторых, переформируют информацию в ячейках декодирующего регистра. Как видно из

табл. 4.5, на 9–м такте в ячейках декодирующего регистра сформирована комбинация 001, на 10–м такте эта комбинация поступает на вход дешифратора синдрома, на выходе которого возникает сигнал 1. В этот же момент на выходной сумматор поступает искаженный 3–й разряд, который, проходя через сумматор, меняет знак на обратный. Исправленная комбинация имеет вид 1001.

Таблица 4.5

Декодирование циклического кода (7,4)

Номер такта	Вход	Состояние ячеек			Вход дешифратора синдрома DD7	Выход дешифратора синдрома DD7	Выход декодера
		DD3	DD5	DD6			
0		0	0	0			---
1	1	1	1	0	110		---
2	0	0	1	1	000		---
3	1	0	0	1	101		---
4	1	0	0	0	111		---
5	1	1	1	0	110		---
6	1	1	0	1	000		---
7	0	1	0	0	011		---
8	---	0	1	0	010	0	1
9	---	0	0	1	100	0	0
10	---	1	1	0	111	1	0
11	---	0	1	1	000	0	1

На рис. 4.29 представлена функциональная схема мажоритарного декодирования кода (7, 3) (см. пример 2.17). В процессе заполнения регистра декодируемой кодовой комбинации ключ *SWT* находится в положении 1. После заполнения регистра сдвига на выходах сумматоров формируются результаты проверок относительно разряда  $a_0$  (2.48):  $a_1 \oplus a_3$  (*DD8*),  $a_2 \oplus a_6$  (*DD9*) и  $a_4 \oplus a_5$  (*DD10*);  $a_0$  поступает непосредственно в схему, где происходит под-

счет чисел 1 и 0, т.е. в мажоритарный элемент  $M \geq 3$ , который выносит решение о значении разряда  $a_0$ . Далее ключ  $SWT$  переводится во второе положение, подается еще один тактовый импульс в регистр, комбинация сдвигается на один разряд вправо, создаются условия (2.49) по проверке разряда  $a_1$  и мажоритарный элемент выносит решение о значении разряда  $a_1$  и т.д. вплоть до декодирования разряда  $a_6$ . Таким образом, декодирование кодовой комбинации осуществляется за  $2n$  тактов: в течение первых  $n$  тактов заполняется регистр  $DD1 \dots DD7$ , а в течение последующих определяется значение каждого из  $n$  разрядов. Вывод информации потребителю осуществляется с выхода мажоритарного элемента через схему И  $DD12$ .

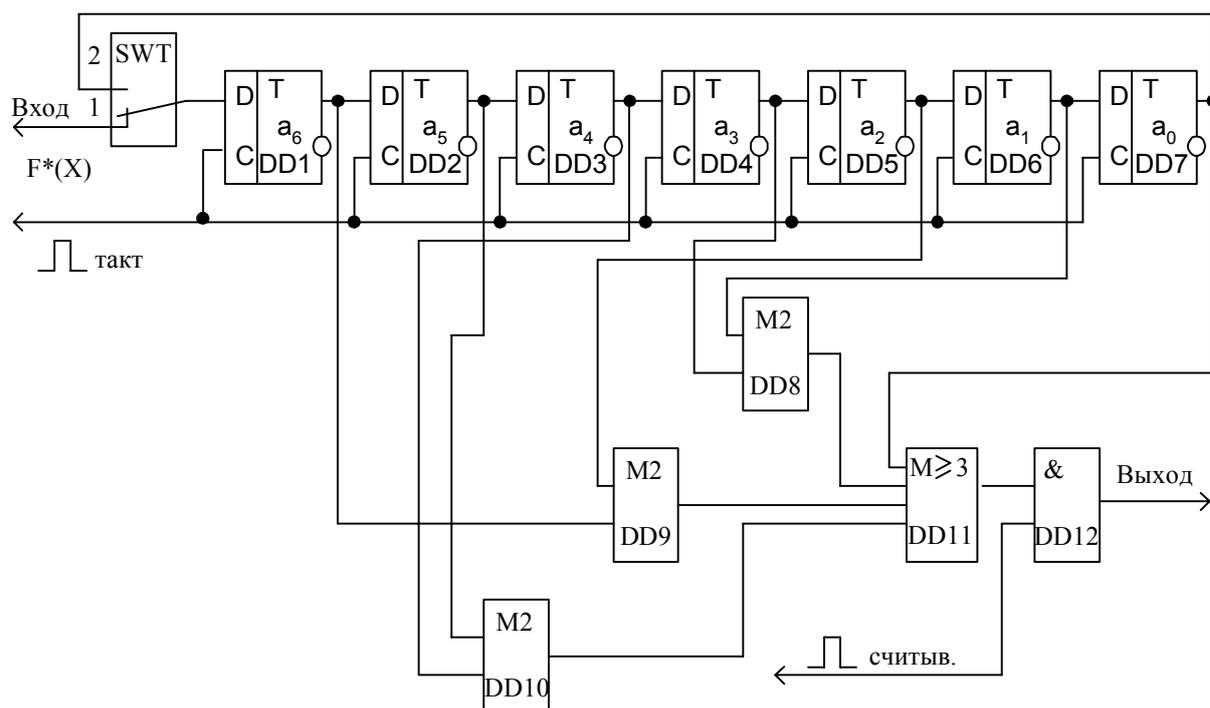


Рис. 4.29. Мажоритарный декодер циклического кода (7,3)

### 4.13. Кодер и декодер итеративного кода

Рассмотрим кодер итеративного кода для структуры кодовой комбинации приведенной в табл. 4.6.

Таблица 4.6

Определение контрольных символов итеративного кода

$k_1$	1	$k_2$	0	$k_3$	1	$r_1$	0
$k_4$	1	$k_5$	0	$k_6$	0	$r_2$	1
$r_3$	0	$r_4$	0	$r_5$	1	$r_6$	1

Кодер для данной структуры приведен на рис. 4.30.

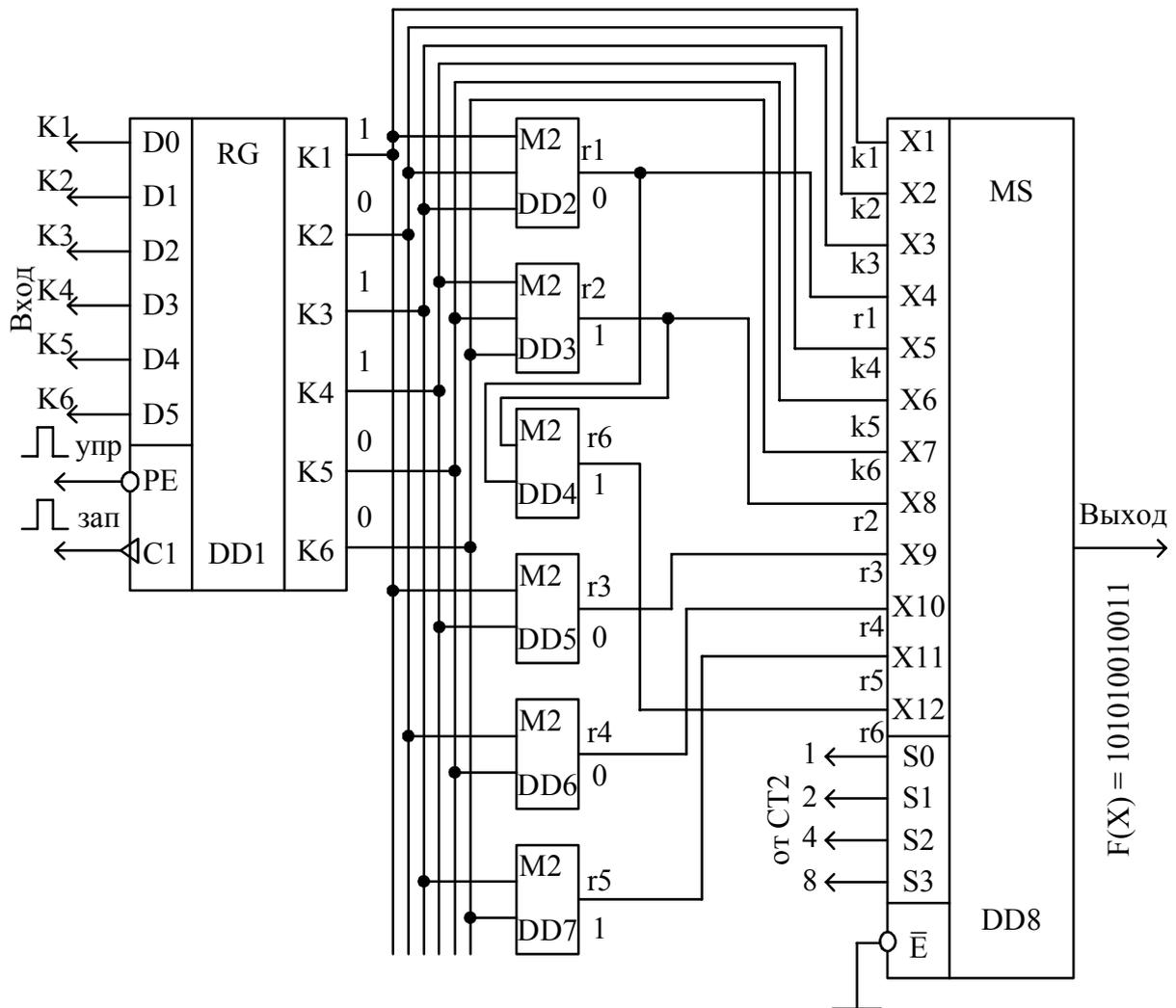


Рис. 4.30. Кодер итеративного кода

Кодовая комбинация, подлежащая кодированию, записывается в буферный регистр  $DD1$ . Формирование контрольных символов  $r_1 \dots r_6$  сумматорами по модулю два  $DD2 \dots DD7$  осуществлена в соответствии строкам и столбцам табл. 4.6. Мультиплексор  $DD8$  осуществляет преобразование параллельного кода в последовательный.

Декодер для рассматриваемой структуры кодовой комбинации приведен на рис. 4.31. Кодовая комбинация в итеративном коде записывается в буферный регистр  $DD1$ . Затем сумматорами по модулю два  $DD2 \dots DD6$  осуществляются проверки в соответствии с табл. 4.6, т.е.  $S_1 = k_1 \oplus k_2 \oplus k_3 \oplus r_1$ ,  $S_2 = k_4 \oplus k_5 \oplus k_6 \oplus r_2$ ,  $S_3 = k_1 \oplus k_4 \oplus r_3$ ,  $S_4 = k_2 \oplus k_5 \oplus r_4$ ,  $S_5 = k_3 \oplus k_6 \oplus r_5$ . При отсутствии искажений синдром  $S_5 S_4 S_3 S_2 S_1$  должен быть нулевого порядка. При наличии одиночной ошибки на выходе соответствующий

щих сумматоров  $DD2...DD6$  появляются 1, которые поступят на дешифратор синдрома, собранный на элементах И  $DD7...DD12$ . Дешифратор однозначно указывает номер искаженного символа, и 1 с соответствующей схемы И  $DD7...DD12$  поступит на один из входов схем коррекции ошибки  $DD13...DD18$ , на второй вход которых поступают соответствующие информационные символы  $k_1 ... k_6$ . Информационный символ, проходя через сумматор по модулю 2, сложится с 1 соответствующей схемы И  $DD7...DD12$  и изменит свое значение на противоположное.

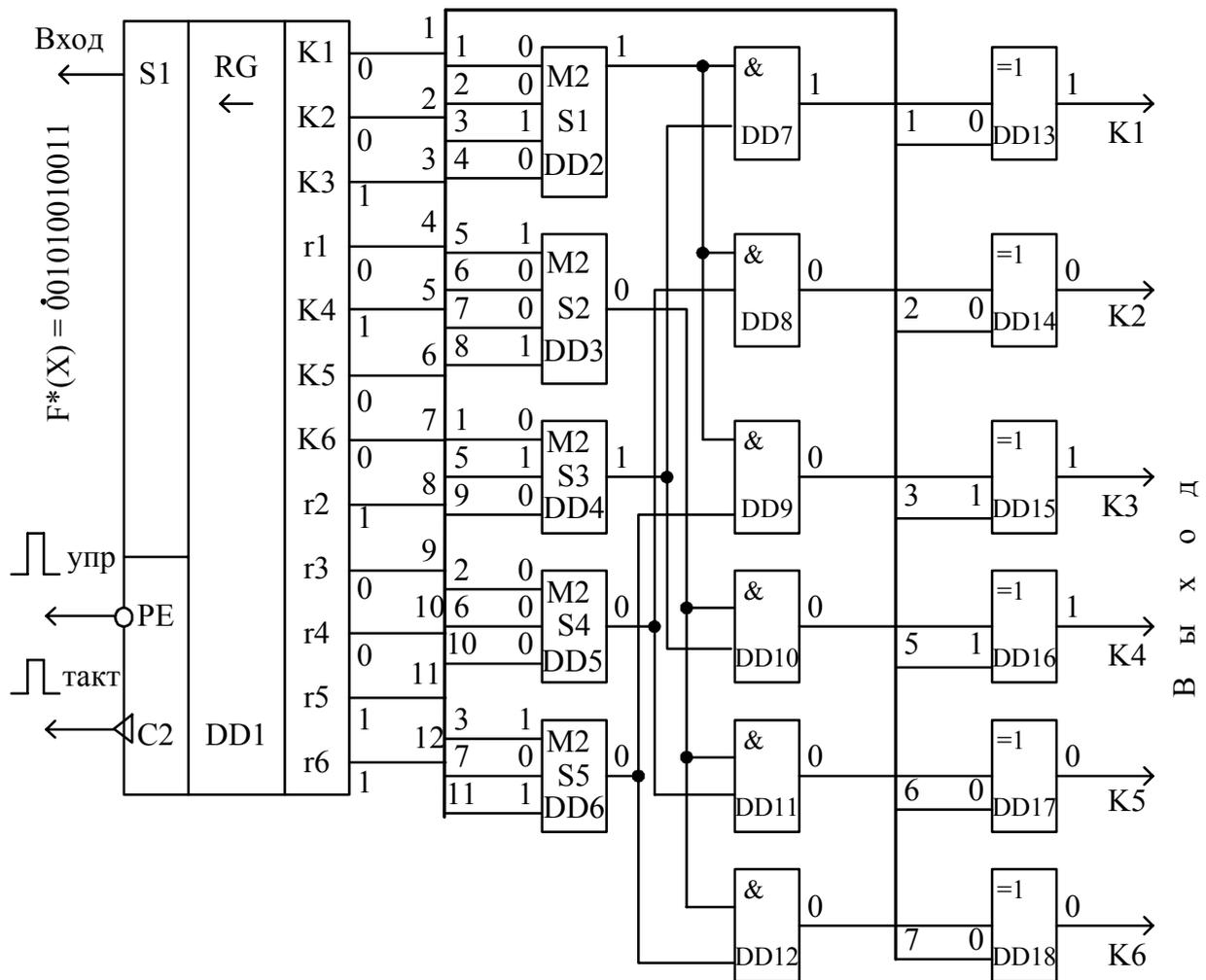


Рис. 4.31. Декодер итеративного кода

На схеме показана дешифрация кодовой комбинации  $F^*(x) = 001010010011$  в виде состояния элементов. В результате дешифрации получим синдром  $S_5S_4S_3S_2S_1 = 00101$ , что вызвало появление 1 на выходе схемы

И  $DD7$ , которая поступила на один из входов схемы  $DD13$ . На второй вход поступил информационный символ  $k_1 = 0$ , который при прохождении через схему  $DD13$  изменил свое значение на 1 и на вход потребителя поступила кодовая комбинация  $k_1k_2k_3k_4k_5k_6 = 101100$ , что соответствует исходной, указанной в табл. 4.6.

#### 4.14. Кодер и декодер рекуррентного кода

Процесс образования и декодирования кодовых комбинаций рекуррентного кода достаточно полно рассмотрен в подразд. 2.3.5. Там же приведены структурные схемы кодирующих и декодирующих устройств рекуррентного кода при шаге сложения  $b=2$ . Для более глубокого понимания процессов обнаружения и исправления ошибок рассмотрим функциональные схемы кодеров и декодеров при шаге сложения  $b=3$  на примере исходной кодовой комбинации  $G(x) = 1111000011111100$ .

Кодирующее устройство такого кода представлено на рис. 4.32. Процесс образования контрольных символов  $r(x)$  с помощью данного кодера представлен в табл. 4.7.

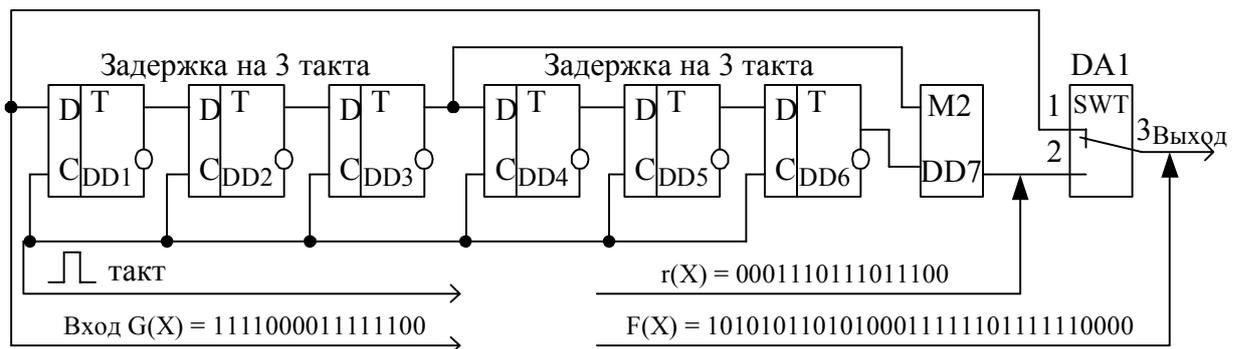


Рис. 4.32. Функциональная схема кодера рекуррентного кода (2, 1) при  $b = 3$

Ключ  $DA1$  находится в положении 1, когда на вход кодера поступает информационный символ, и в положении 2, когда с выхода сумматора по модулю два поступает контрольный символ. Таким образом, выходная последовательность  $F(x)$  в точке 3 представляет собой чередование информационных и контрольных символов.

Декодирующее устройство представлено на рис. 4.33.

Таблица 4.7

Образование контрольных символов при  $b = 3$ 

Номер такта	Вход	Состояние ячеек регистра						Выход DD7
		DD1	DD2	DD3	DD4	DD5	DD6	
		→	↘	↘	↘	↘	↘	↘
1	1	1	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0
3	1	1	1	1	0	0	0	0
4	1	1	1	1	1	0	0	1
5	0	0	1	1	1	1	0	1
6	0	0	0	1	1	1	1	1
7	0	0	0	0	1	1	1	0
8	0	0	0	0	0	1	1	1
9	1	1	0	0	0	0	1	1
10	1	1	1	0	0	0	0	1
11	1	1	1	1	0	0	0	0
12	1	1	1	1	1	0	0	1
13	1	1	1	1	1	1	0	1
14	1	1	1	1	1	1	1	1
15	0	0	1	1	1	1	1	0
16	0	0	0	1	1	1	1	0

Как известно из подразд. 2.3.5, процесс декодирования заключается в формировании контрольных символов из информационных, поступивших на декодер, и их сравнении с контрольными символами, пришедшими из канала связи. В результате сравнения вырабатывается корректирующая последовательность, которая и производит исправление информационной последовательности.

Рассмотрим работу декодера. Входная кодовая комбинация  $F^*(x)$  делителем  $DA1$  разделяется на последовательности информационных и контрольных символов. Посредством линейного преобразователя на элементах  $DD1...DD6$  и  $DD10$  аналогично преобразователю кодирующего устройства, снова формируются проверочные символы  $r^{**}(X)$ , которые сравниваются (суммируются по модулю 2) элементом  $DD11$  с проверочными символами  $r^*(X)$ , поступающими непосредственно из канала связи. Если ошибок нет, то на выходе формирователя синдрома  $DD11$  имеем последовательность, состоящую из одних нулей. Каждой конкретной пачке ошибок соответствует свой синдром. Определим его структуру. Будем считать, что произошел наихудший случай: исказилось  $2b$  символов. Следовательно, будет поражено  $b$  информационных и  $b$  проверочных символов. До поступления первого ошибочного символа на входе регистр содержит безошибочные информационные символы.

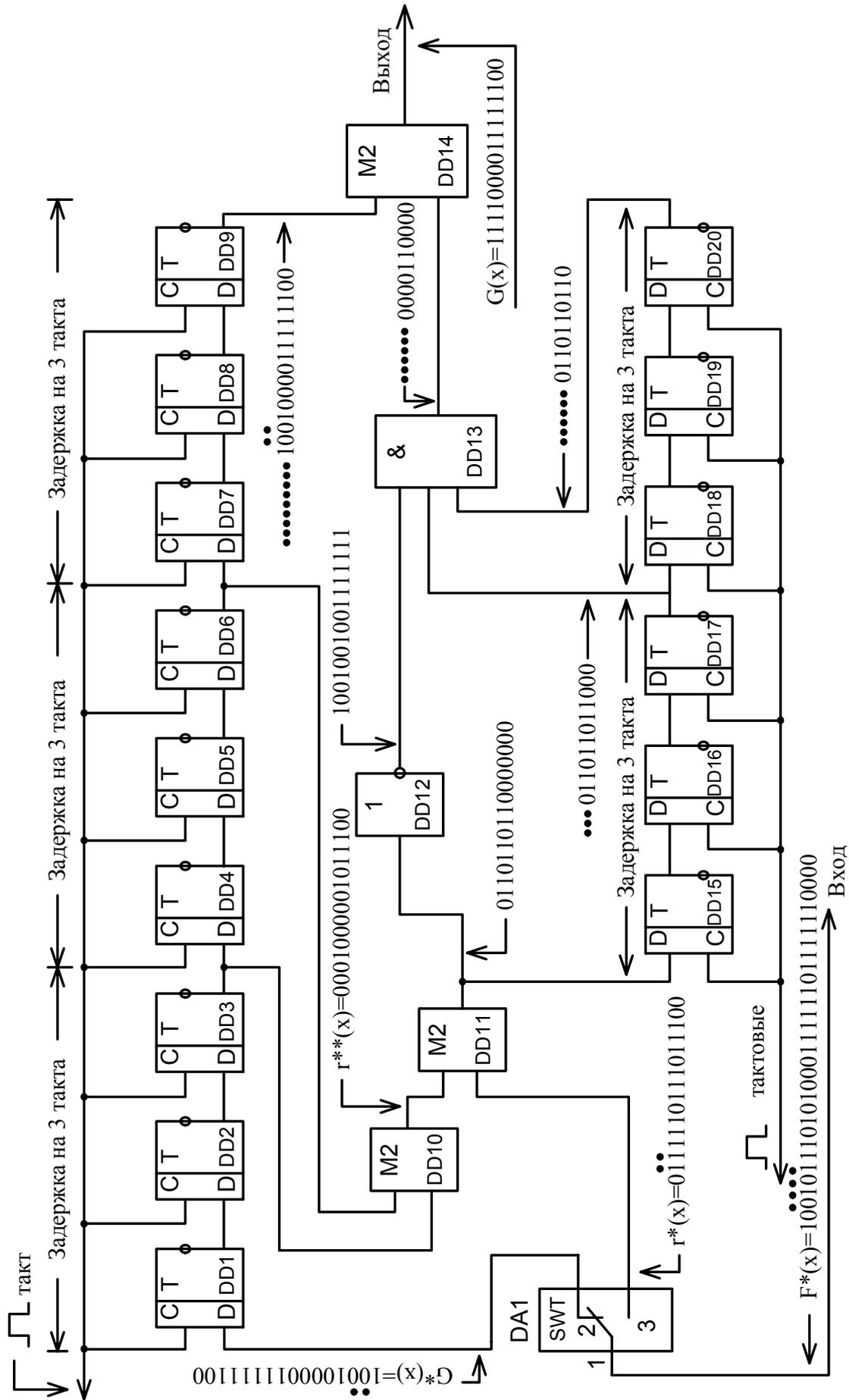


Рис. 4.33. Функциональная схема декодера рекуррентного кода (2,1) при  $b=3$

Поэтому в течение первых  $b$  тактов в синдроме возникают единицы из-за ошибок в проверочных символах. На этом пачка ошибок заканчивается, и в дальнейшем на выходной сумматор формирователя синдрома  $DD11$  будут поступать лишь безошибочные проверочные символы. За следующие  $2b$  тактов единицы формируются в синдроме сначала из-за поступления ошибочных информационных символов из первого полурегистра  $DD1...DD3$ , а затем из второго  $DD4...DD6$ . Таким образом, синдром содержит: 1) единицы на местах ошибок в проверочных символах; 2) со сдвигом на  $b$  символов – единицы на местах ошибок в информационных символах; 3) еще со сдвигом на  $b$  повторяется комбинация, полученная в предыдущем случае.

Как видно из рис. 4.33, анализатор синдрома на элементах  $DD15...DD20$  и  $DD13$  построен в точном соответствии с его структурой. Поскольку корректирующий сигнал формируется через  $3b$  тактов, а информационные символы в формирователе синдрома  $DD1...DD6$  задерживаются только на  $2b$  тактов, то возникает необходимость в дополнительной задержке информационных символов на  $b$  тактов, что производится элементами  $DD7...DD9$ .

Таким образом, на пути информационных символов в декодере имеется всего  $3b$  ячеек  $DD1...DD9$ . Это соответствует  $6b$  символам во входной последовательности  $F^*(x)$ .

Следовательно, чтобы вывести все ошибочные символы из схемы, требуется промежуток  $6b+1$  безошибочных символов. Чтобы не проводилось исправлений в случае появления ошибочных символов в этот период, предусмотрен элемент НЕ  $DD12$ .

Функционирование декодирующего устройства при дешифрации конкретного сообщения  $F^*(x)$  показано на рис. 4.33 в виде конкретных комбинаций на входе и выходе отдельных элементов, которые наглядно демонстрируют исправление двух информационных символов (помеченных точкой сверху), искаженных помехой. Точки спереди кодовых комбинаций означают задержку на соответствующее число тактов.

## 5. ТЕХНИЧЕСКИЕ СРЕДСТВА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ЧАСТОТНЫХ КОДОВ

### 5.1. Кодер и декодер кода на перестановки

Теоретические аспекты построения данного кода рассмотрены в подразд. 2.4.2, а кодер для трех частот приведен на рис. 5.1. В соответствии с выражением (2.62) число команд, которые могут быть представлены в данном коде, равно 6, а именно:  $f_1f_2f_3$ ,  $f_1f_3f_2$ ,  $f_2f_1f_3$ ,  $f_2f_3f_1$ ,  $f_3f_1f_2$ ,  $f_3f_2f_1$ , причем элементы команд передаются последовательно. Выбор любой команды осуществляется одним из ключей  $SA1...SA6$ . Схемами ИЛИ  $DD1...DD3$  формируются элементы команды первой ступени, схемами  $DD4...DD6$  – второй ступени и схемами  $DD7...DD9$  – третьей ступени. Опрос состояния схем каждой

ступени производится импульсами с распределителя импульсов путем подачи их на входы схем И  $DD10...DD12$ ,  $DD13...DD15$ ,  $DD16...DD18$  соответственно. Видеосигналы с выхода схем  $DD10...DD18$  объединяются схемами ИЛИ  $DD19...DD21$ , которые подключают через ключи  $DA4...DA5$  ко входу сумматора  $DA7$  соответствующие генераторы  $DA1...DA3$ . Таким образом, в зависимости от замкнутого ключа на выходе будет сформирована одна из 6 команд.

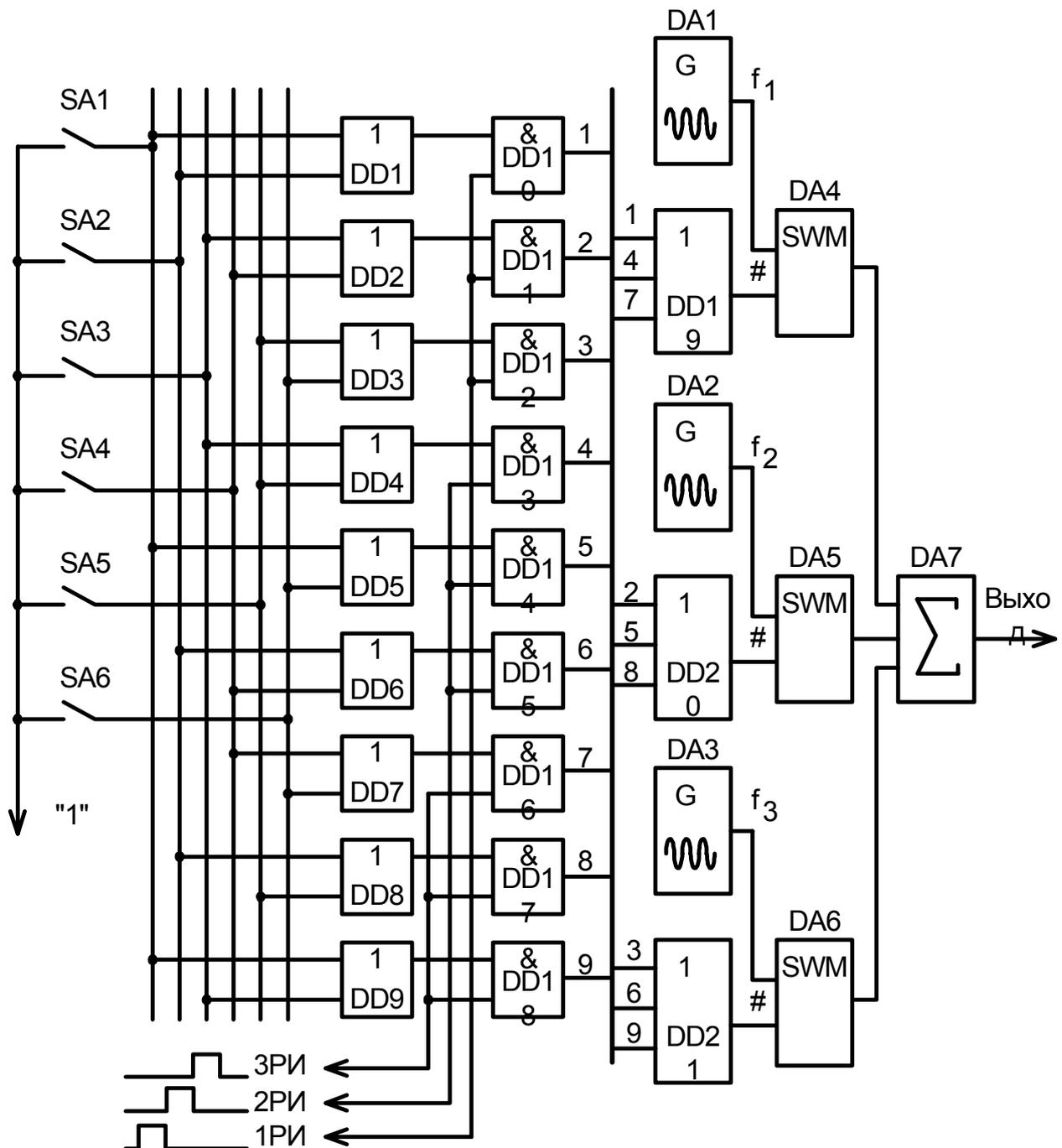


Рис. 5.1. Схема кодера кода на перестановки  $P_3$

Схема декодера приведена на рис. 5.2. Сигналы, пришедшие из канала связи, селективируются полосовыми фильтрами  $DA1...DA3$ , преобразуются в видеоимпульсы преобразователями  $DA4...DA6$ , которые в общем случае представляют собой амплитудные детекторы. Затем видеоимпульсы, соответствующие радиоимпульсам, пришедшим на соответствующих временных позициях, записываются в трехразрядные регистры  $DD5...DD7$ . Дешифрация сигналов с выхода регистров производится схемами И  $DD8...DD13$ . Подключение входов каждого элемента производится в соответствии с правилами формирования команд кодером рис. 5.1 и наглядно видно из рис. 5.2, где на выходе каждого элемента подписаны частоты и очередность их следования.

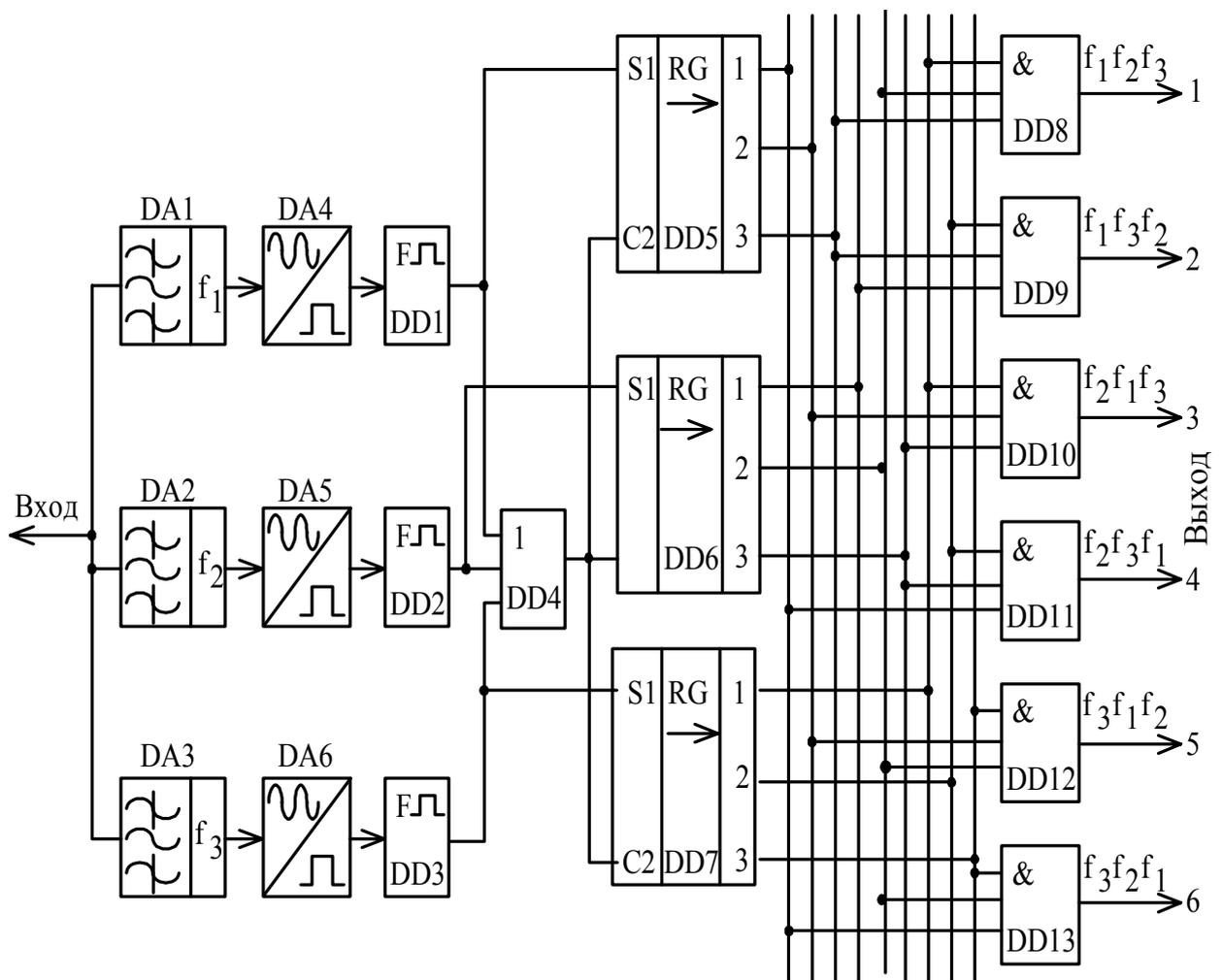


Рис. 5.2. Схема декодера кода на перестановки  $P_3$

## 5.2. Кодер и декодер кода на размещения

Рассмотрим кодер кода на размещение  $A_3^2$ . С помощью данного кодера можно передать 6 команд следующими комбинациями частот:  $f_1f_2$ ,  $f_1f_3$ ,  $f_2f_3$ ,  $f_2f_1$ ,  $f_3f_1$ ,  $f_3f_2$ .

Функциональная схема такого кодера приведена на рис. 5.3.

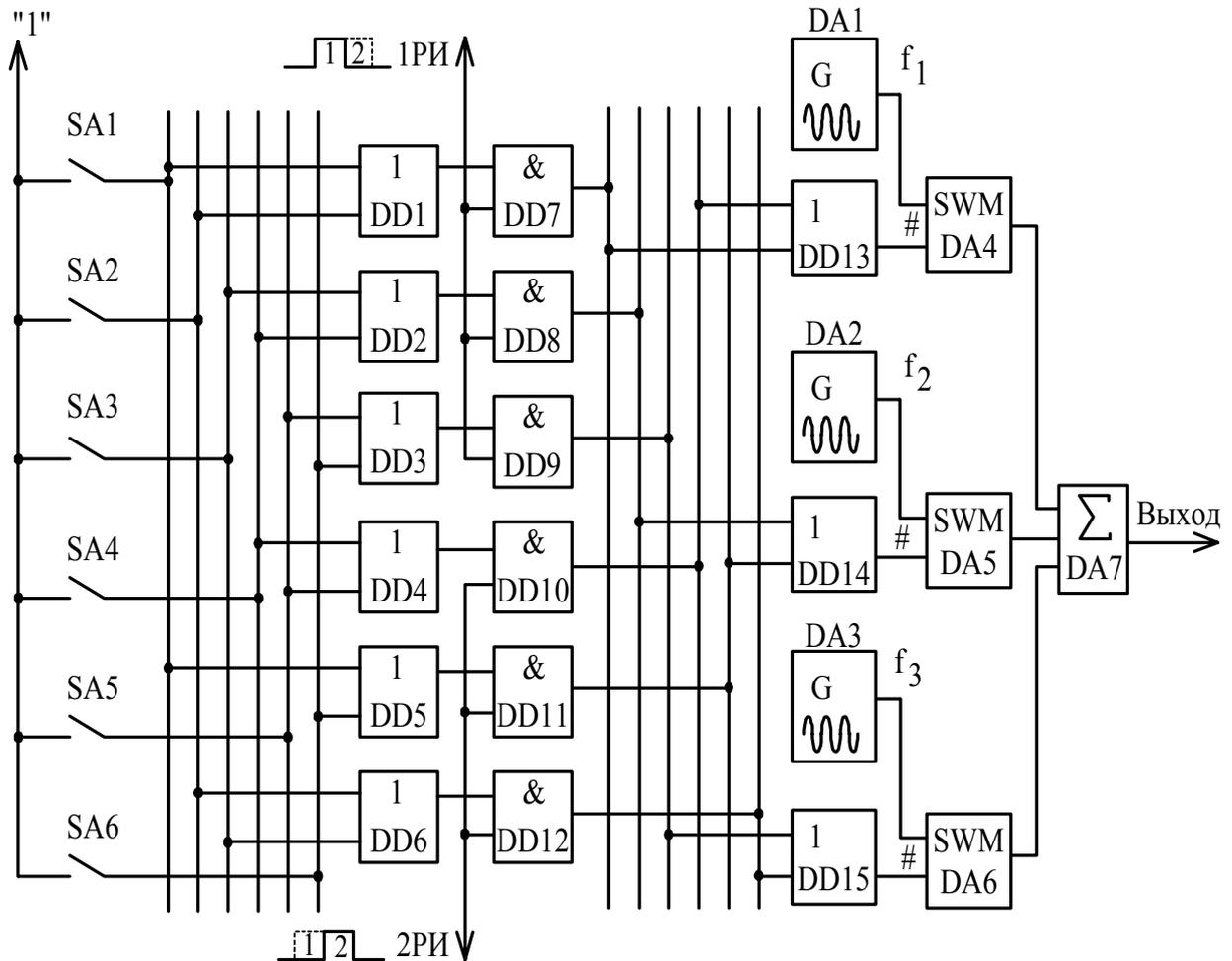


Рис. 5.3. Кодер кода на размещение  $A_3^2$

Элементами ИЛИ  $DD1...DD6$  формируются видеосигналы, соответствующие приведенным выше комбинациям частот. Для организации передачи радиоимпульсов на первой и второй временной позициях служит распределитель импульсов (на схеме показаны только его выходы), имеющий два выхода  $1PI$  и  $2PI$ . Так, при замыкании ключа  $SA1$  сигнал через схемы ИЛИ  $DD1$  и  $DD5$  подготовит к работе схемы ИЛИ  $DD7$  и  $DD11$ . Опрос  $DD7$  осуществляется на первом шаге, а  $DD11$  – на втором шаге  $PI$ .

Сигнал с выхода  $DD7$  через  $DD13$  замыкает ключ  $DA4$ , а сигнал с выхода  $DD11$  через  $DD14$  замыкает ключ  $DA5$ , что обеспечивает поступление частотных посылок сначала  $f_1$ , а затем  $f_2$  через выходной сумматор  $DA7$ . Таким образом, в линию связи поступит последовательный двухчастотный код  $f_1 f_2$ . Аналогично работает кодирующее устройство и при замыкании любого другого ключа  $SA2 \dots SA6$ .

На рис. 5.4 приведена схема декодирующего устройства последовательного двухчастотного кода на размещение из  $n = 3$  частот по две ( $m = 2$ ).

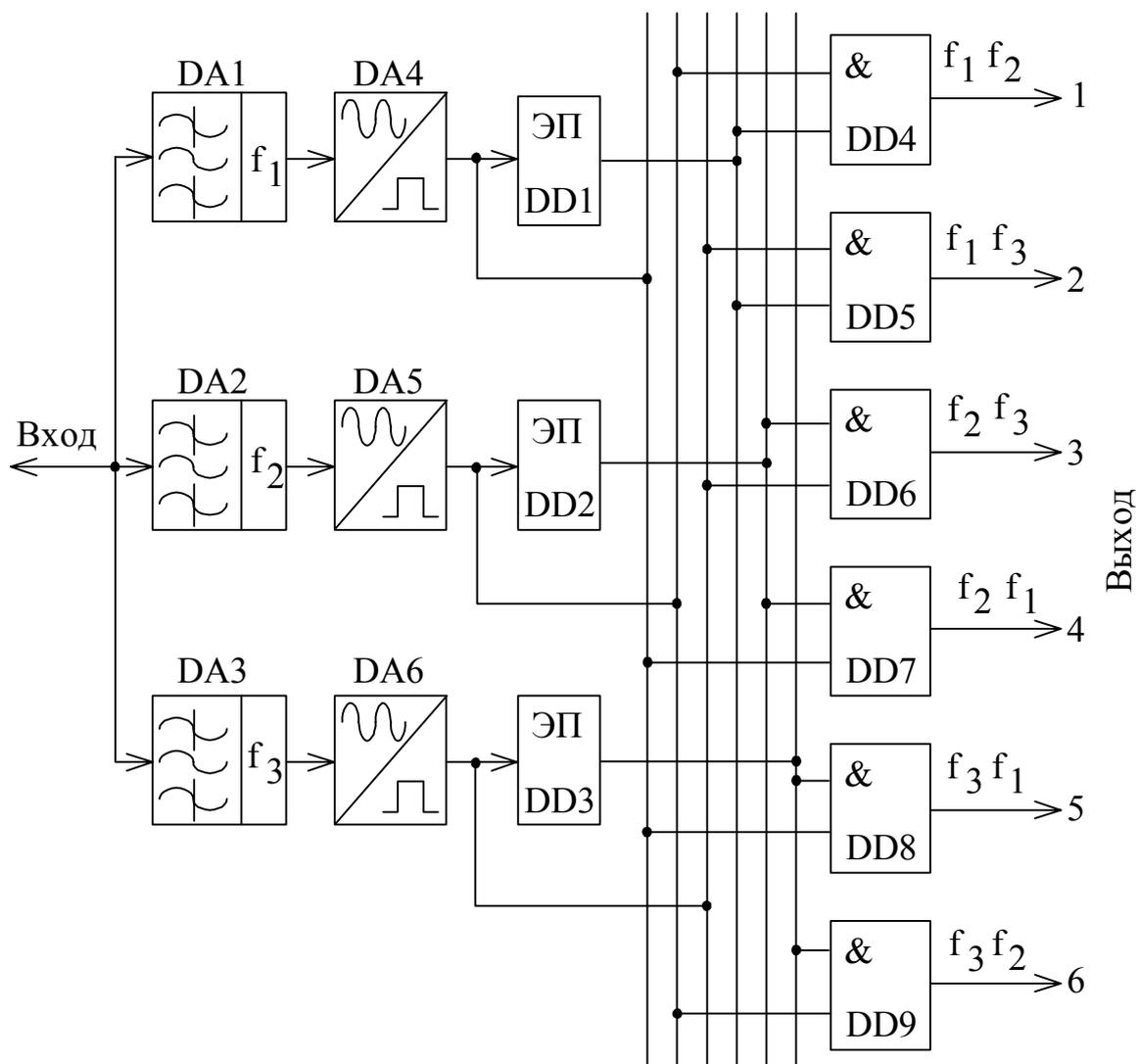


Рис. 5.4. Декодеры кода на размещение  $A_3^2$

Пришедшие от линейного блока радиоимпульсы разделяются полосовыми фильтрами  $DA1 \dots DA3$ , затем выделяется огибающая радиоимпульса и формируется видеоимпульс с параметрами, необходимыми для работы интегральных

микросхем преобразователями  $DA4...DA6$ , в качестве которых можно использовать амплитудные детекторы с триггерами Шмитта.

Видеосигналы с выхода преобразователей  $DA4...DA6$ , пришедшие на первых позициях, запоминаются элементами кратковременной памяти (ЭП)  $DD1...DD3$ , в качестве которых могут использоваться регистры сдвига как в схеме рис. 5.2. Дешифровка сигналов производится схемами И  $DD4...DD9$ , входы которых подключаются в соответствии с кодовыми комбинациями, сформированными при кодировании.

### 5.3. Кодер и декодер кода на сочетании

Принципы построения данных кодов приведены в подразд. 2.4.4. Рассмотрим кодер кода на сочетание  $C_4^2$ , схема которого представлена на рис. 5.5.

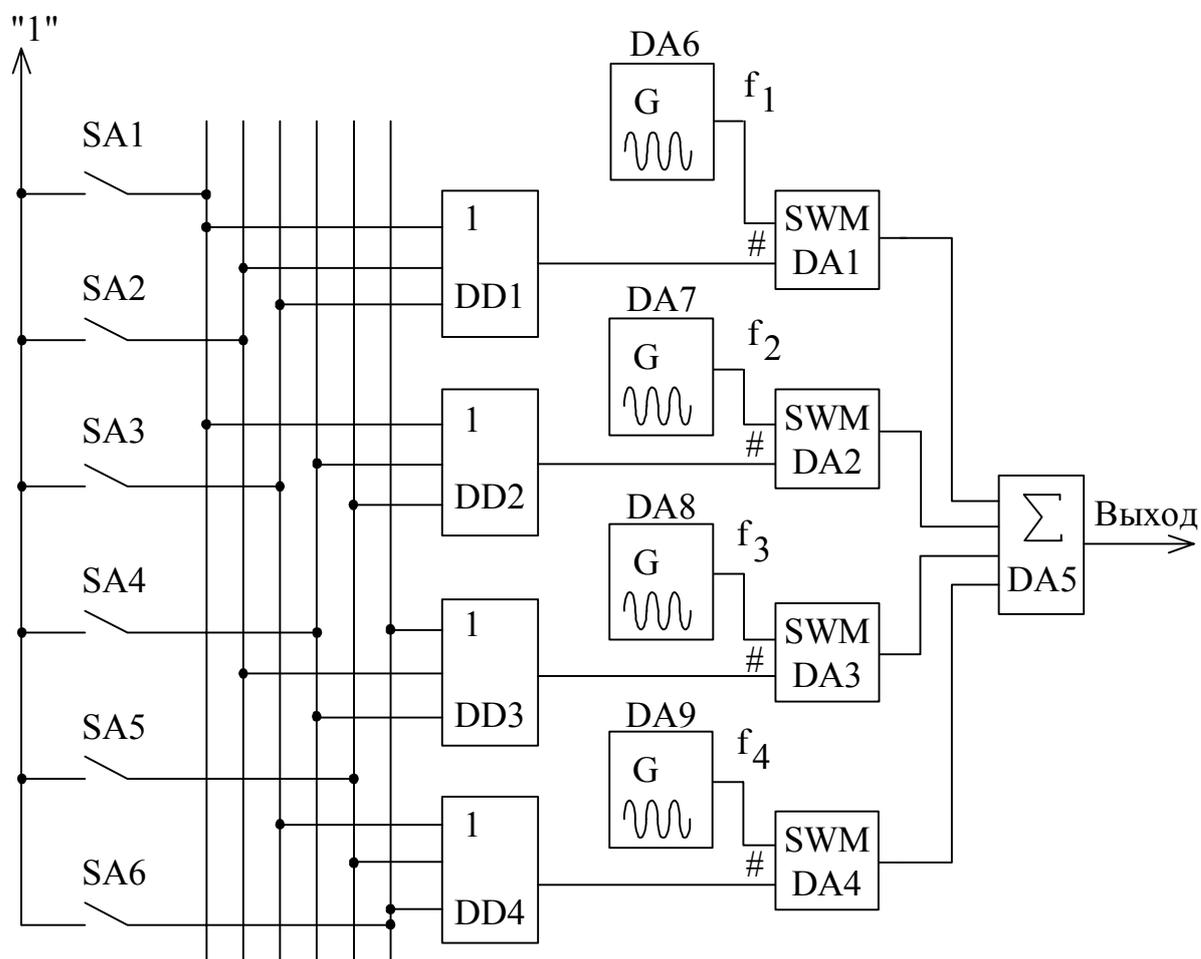


Рис. 5.5. Кодер кода на сочетание  $C_4^2$

Как известно, в данном коде может быть представлено шесть комбинаций:  $f_1f_2$ ,  $f_1f_3$ ,  $f_1f_4$ ,  $f_2f_3$ ,  $f_2f_4$ ,  $f_3f_4$ . Радиоимпульсы в данном коде пере-

даются в параллельном виде. Формирование закона комбинирования радиоимпульсов в соответствии с приведенными комбинациями производится схемами ИЛИ  $DD1 \dots DD4$  при замыкании соответствующего ключа  $SA1 \dots SA6$ . Например, замыкая ключ  $SA1$ , плюс источника питания  $E_{\text{п}}$  через схемы ИЛИ  $DD1$  и  $DD2$  поступает на цифровой вход ключей  $DA1$  и  $DA2$ , открывает их, и сигналы с частотами  $f_1$  и  $f_2$  от генераторов  $DA6$  и  $DA7$  соответственно поступают через выходной сумматор  $DA5$  к линейному блоку. При замыкании ключа  $SA2$  в линейный блок поступит сигнал с частотами  $f_1 f_3$  и т.д.

Декодер кода  $C_4^2$  приведен на рис. 5.6.

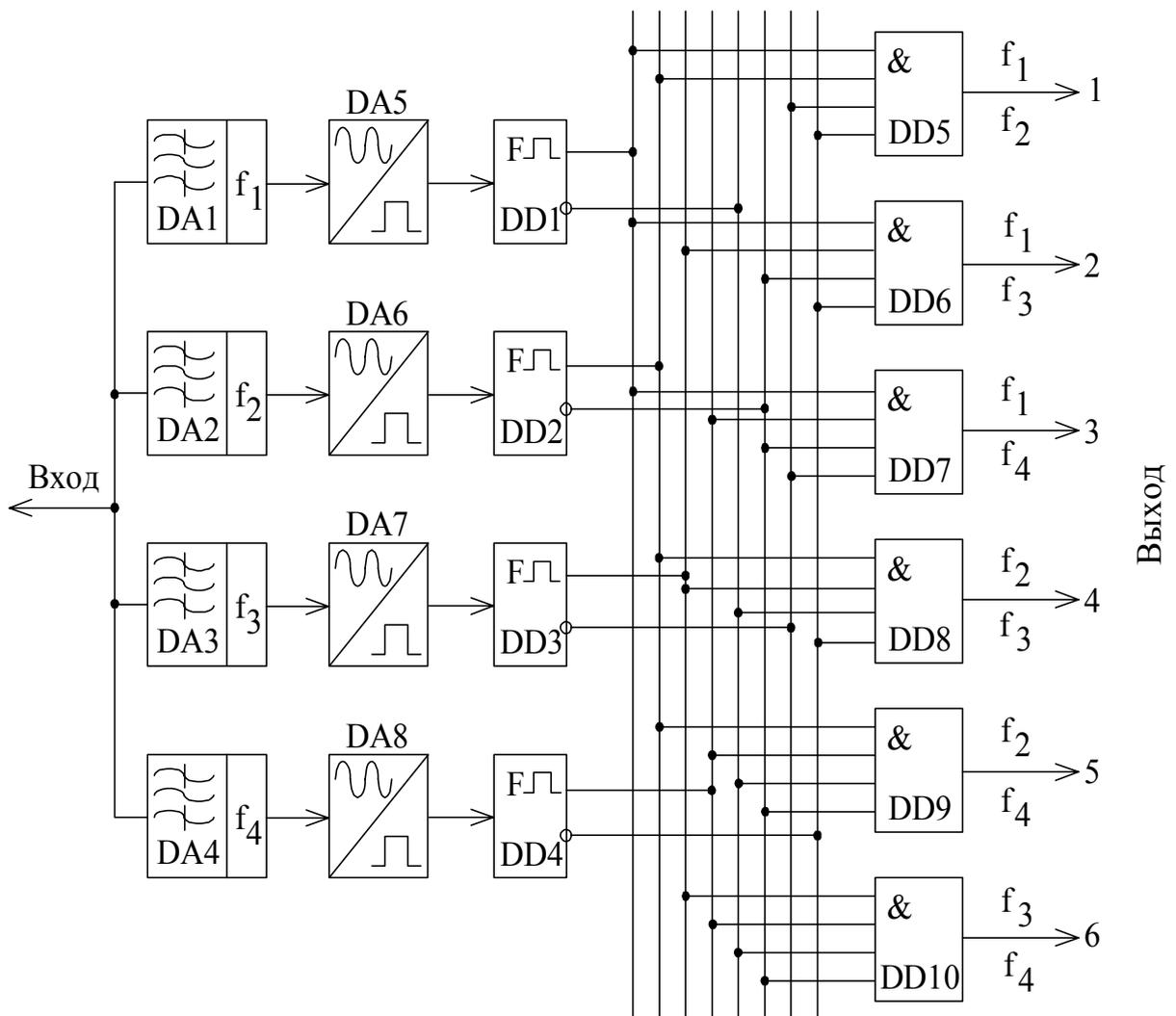


Рис. 5.6. Декодер кода на сочетание  $C_4^2$

Назначение полосовых фильтров  $DA1 \dots DA4$ , преобразователей  $DA5 \dots DA8$  и формирователей импульсов  $DD1 \dots DD4$  такое же, как и в предыдущих схемах декодеров на перестановки (см. рис. 5.2) и на размещение (см. рис. 5.4).

При построении данного декодера большое внимание уделено обеспечению защитного отказа в случае сбоев. Для этой цели на входы дешифраторов, собранных на схемах И  $DD5\dots DD10$ , заведены два прямых (информационных) и два инверсных (неинформационных для данного канала) сигнала с выходов формирователей  $DD1\dots DD4$ . Информационные входы на каждую схему И заведены в соответствии с законом формирования комбинаций кодером. Таким образом, при приходе более или менее двух радиоимпульсов ни одна из схем И  $DD5\dots DD10$  на выходе не будет иметь информационного сигнала, а следовательно, не произойдет ложного срабатывания объекта.

#### 5.4. Дешифратор одночастотного кода

Дешифровка одночастотных сигналов может быть произведена с помощью селективных электронных реле и цифровых частотных избирателей. В данном подразделе рассмотрим вторые как наиболее перспективные при построении систем телемеханики на элементах цифровой вычислительной техники. Схема такого дешифратора приведена на рис. 5.7.

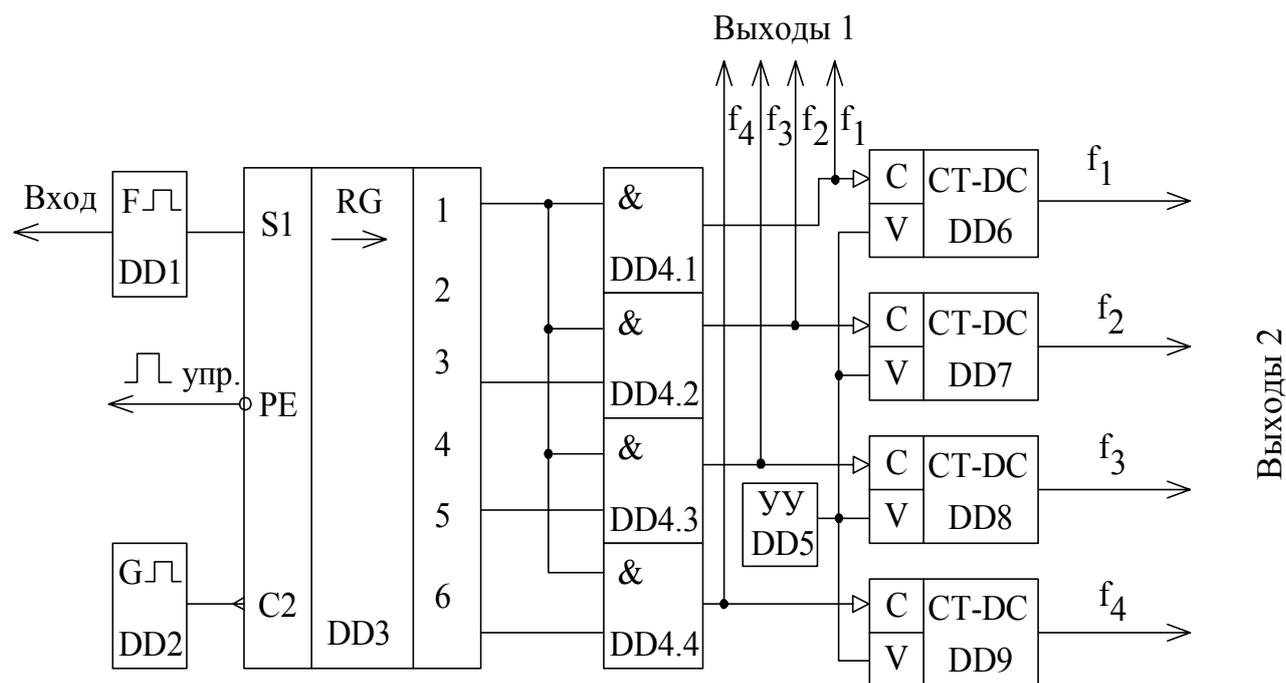


Рис. 5.7. Дешифратор одночастотного сигнала

Из синусоидального сигнала, поступившего от линейного блока, формируются прямоугольные импульсы формирователем  $DD1$ . Первый импульс записывается в первую ячейку регистра сдвига  $DD3$ , который затем продвигается

тактовыми импульсами от генератора  $DD2$ . Второй импульс записывается снова в первую ячейку и одновременно поступает на входы всех схем И. Появившийся при этом сигнал на выходе одной из схем И  $DD4.1 \dots DD4.4$  однозначно связан с частотой радиоимпульса. Для увеличения надежности управления сигналы с выхода схем И  $DD4.1 \dots DD4.4$  могут поступать на пересчетные схемы  $DD6 \dots DD9$ , которые состоят их счетчиков и дешифраторов, где определяется соответствие сигнала на нескольких периодах следования входных радиоимпульсов. Вывод сигнала для управления выходными исполнительными элементами производится по командам с устройства управления  $DD5$  (УУ).

### 5.5. Кодер и декодер сменно–качественного кода

Как известно из подразд. 2.4.5, в данном коде соседние символы не могут быть одинаковыми. Таким образом, при передаче информации бинарными кодами необходимо надлежащим образом производить преобразование передаваемых сообщений.

В схеме кодера, приведенной на рис. 5.8, это решено следующим образом. Кодовая комбинация, подлежащая передаче, записывается в буферный регистр  $DD1$  и при подаче тактовых импульсов от генератора  $DD2$  поступает на входы  $R$  установки в исходное положение счетчиков  $DD3$  и  $DD4$ .

Таким образом, счетчики  $DD3$  и  $DD4$  подсчитывают соответственно число единиц и число нулей, следующих подряд в передаваемой комбинации. Дешифратором  $DD5$  при нечетном числе 1 выходы 1, 3, 5, 7 объединяются схемой ИЛИ  $DD7$ , этими сигналами подключается генератор  $DA1$ , и в канал связи поступают радиоимпульсы с частотой  $f_1$ . Все четные выходы дешифраторов  $DD5$  и  $DD6$  объединяются схемой ИЛИ  $DD8$ , и в результате создаются условия для подачи в канал связи радиоимпульсов с частотой  $f_3$ . Все нечетные выходы дешифратора  $DD6$  объединяются схемой ИЛИ  $DD9$ , тем самым обеспечивается поступление в канал связи радиоимпульсов с частотой  $f_2$ . Процесс преобразования кодовой комбинации  $G(x) = 1110010$  в сменно–качественный код  $F(x) = f_1 f_3 f_1 f_2 f_3 f_1 f_2$  показан на временных диаграммах рис. 5.9.

Основой декодера (рис. 5.10) являются полосовые фильтры  $DA1 \dots DA3$ , которые селективируют радиоимпульсы, поступившие из канала связи, и преобразователи радиоимпульсов в видеоимпульсы  $DA4 \dots DA6$ , которые совместно с формирователями  $DD1 \dots DD3$  формируют видеоимпульсы с параметрами, необходимыми для работы элементов ТТЛ. Триггер  $DD5$  формирует зоны логических единиц. Так как каждой временной зоне соответствует активный видеоимпульс, то последовательно поступающие импульсы с выхода схемы ИЛИ  $DD4$  можно использовать в качестве тактовых. С помощью этих импульсов последовательность информационных символов с выхода триггера  $DD5$  записывается в регистр  $DD6$ . Процесс преобразования последовательности радиоимпульсов  $F(x) = f_1 f_3 f_1 f_2 f_3 f_1 f_2$  в последовательность видеоимпульсов показан на временных диаграммах рис. 5.11.

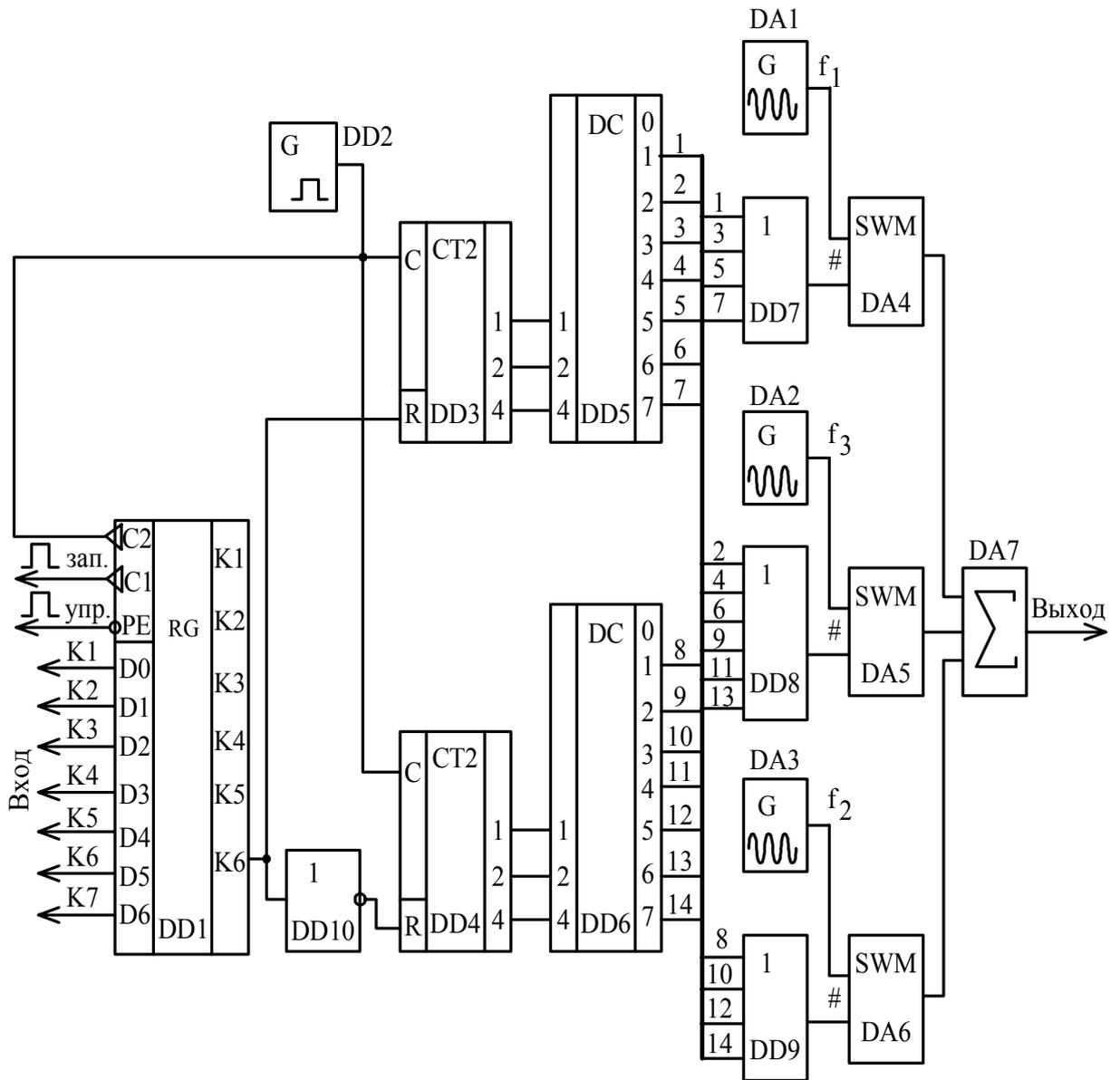


Рис. 5.8. Кодер сменно-качественного кода

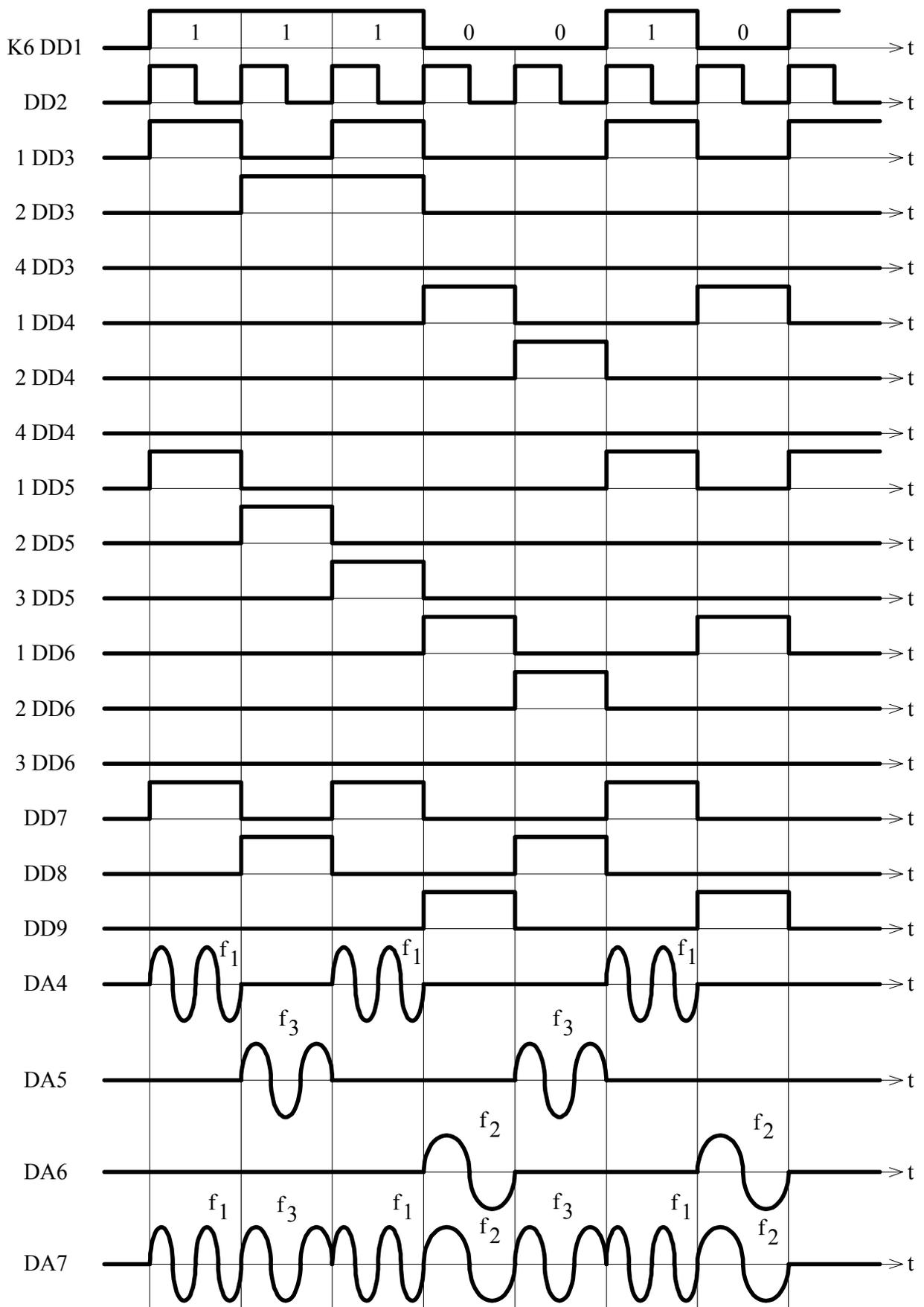


Рис. 5.9. Временные диаграммы работы кодера сменно-качественного кода

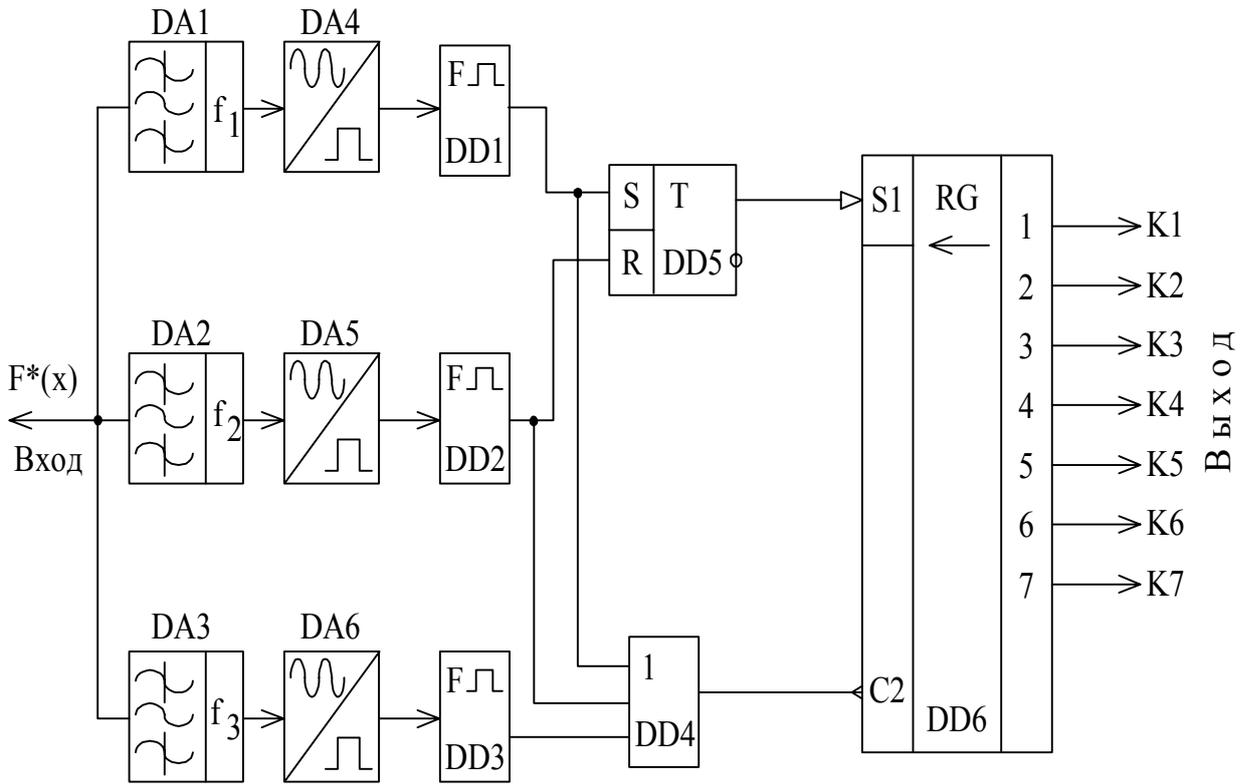


Рис. 5.10. Декодер сменно-качественного кода

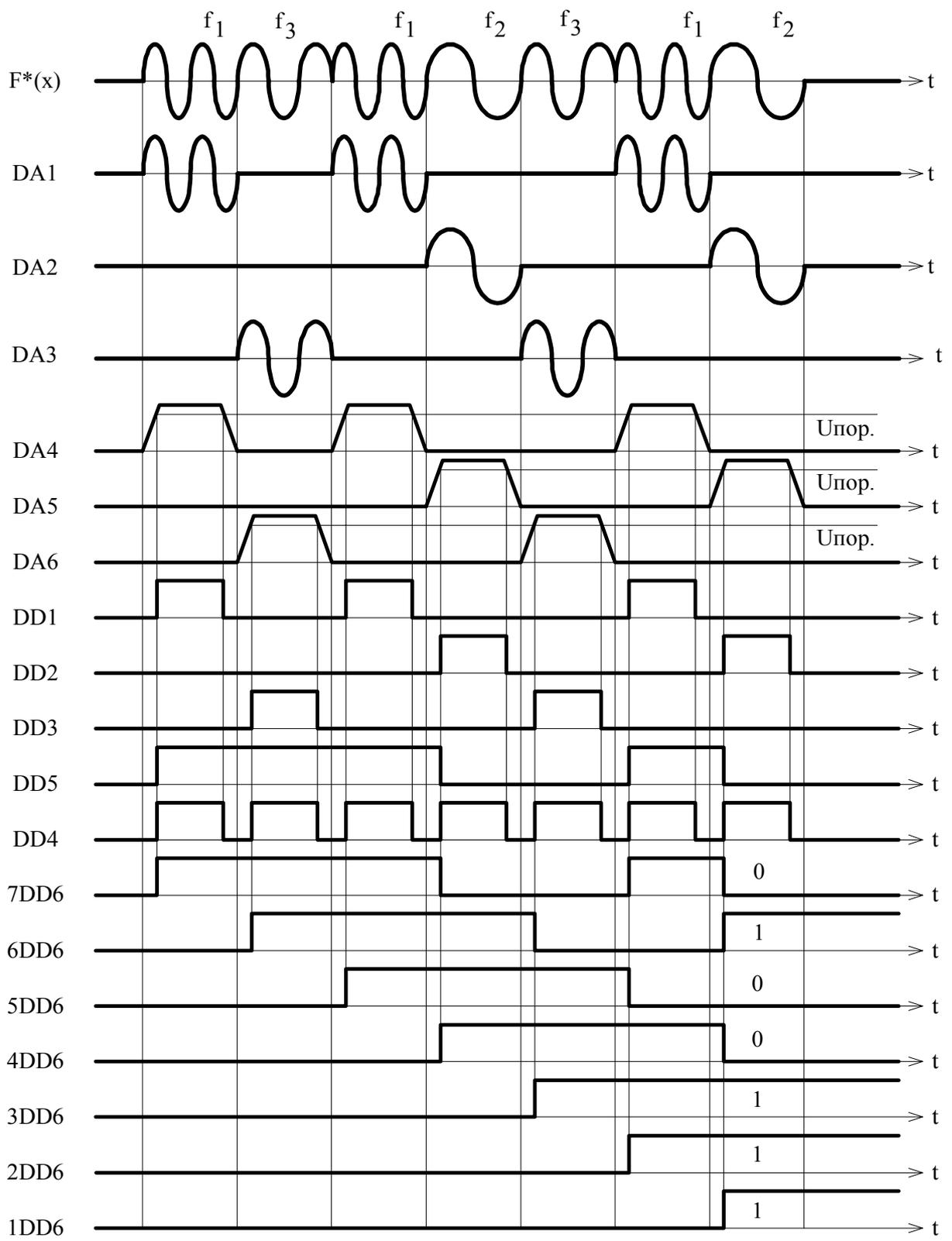


Рис. 5.11. Временные диаграммы работы декодера сменно-качественного кода

## 6. КОДЫ ДЛЯ ПЕРЕДАЧИ ЦИФРОВОЙ ИНФОРМАЦИИ ПО ПОСЛЕДОВАТЕЛЬНЫМ КАНАЛАМ СВЯЗИ

### 6.1. Методы кодирования

Передача информации между двумя достаточно удаленными устройствами требует представления ее в виде последовательного потока бит, характеристики которого зависят от особенностей конкретной системы. Физической основой такой системы является линия связи, которая обычно выполняется в виде витой пары проводов, коаксиального кабеля либо оптического световода. В зависимости от расстояния данные, передаваемые по линии, могут однократно или многократно подвергаться ретрансляции с целью восстановления амплитуды и временных характеристик. Алгоритмы работы передатчика, ретранслятора и приемника определяются выбранным кодом, предназначенным для передачи по линии или линейным кодом. Простейшим линейным кодом является униполярный код типа *NRZ* (non return to zero, рис. 6.1, а). В этом коде нули представлены отсутствием импульса (напряжение, близкое к нулю), а единицы – наличием импульса. Этот код имеет четыре недостатка:

1) средняя мощность, выделяемая на нагрузочном резисторе  $R$ , равна  $U^2/2R$ , что в два раза превышает мощность при биполярном кодировании;

2) большинство линий связи сопрягаются с аппаратурой через реактивные элементы, такие как трансформаторы. Поскольку униполярные сигналы всегда содержат постоянную составляющую и значительную долю низкочастотных компонентов в спектре при передаче длинной последовательности единиц, такое сопряжение затруднено или вовсе невозможно – реактивные элементы на достаточно низких частотах представляют собой либо обрыв, либо короткое замыкание;

3) ретрансляторы и приемники способны надежно восстановить синхронизирующую временную сетку только тогда, когда паузы между импульсами не слишком велики. Другими словами, при передаче достаточно большой последовательности нулей, приемник (или ретранслятор) теряет синхронизацию с передатчиком (или ретранслятором);

4) отсутствие возможности оперативной регистрации ошибок, таких как пропадание или появление лишних импульсов из-за помех.

Биполярный сигнал *NRZ* (рис. 6.1, б) обладает лучшими энергетическими характеристиками. Единица представлена положительным уровнем напряжения, нуль – отрицательным. Средняя мощность равна  $U^2/4R$ , т.е. половине средней мощности униполярного сигнала, хотя перепад уровней тот же самый. Остальные три недостатка сохраняются. Для их ликвидации необходимо введение избыточности одним из двух способов:

1) скорость передачи сигналов по линии выбирается большей, чем скорость передачи информации, без использования дополнительных электрических уровней сигналов;

2) скорость передачи сигналов по линии выбирается равной скорости передачи информации, однако вводятся дополнительные электрические уровни сигналов.

Примером кода с избыточностью, введенной согласно способу 1, является код Манчестер–2.

Форма биполярного сигнала при передаче кода Манчестер–2 показана на рис. 6.1, в.

Единица кодируется отрицательным перепадом сигнала в середине битового интервала, нуль – положительным перепадом. На границах битовых интервалов сигнал, если это необходимо, меняет значение, "готовясь" к отображению очередного бита в середине следующего битового интервала. С помощью кода Манчестер–2 решаются сразу все указанные проблемы. Поскольку число положительных и отрицательных импульсов на любом достаточно большом отрезке времени равно (отличается не более чем на один импульс, что не имеет значения), постоянная составляющая равна нулю. Подстройка синхронизма приемника или ретранслятора производится при передаче каждого бита, т.е. снимается проблема рассинхронизации. Спектр сигнала содержит только две логические составляющие  $F$  и  $2F$ , где  $F$  – скорость передачи информационных бит. Наличие только двух, а не трех или более электрических уровней напряжения позволяет надежно их распознавать (хорошая помехозащищенность).

Второй способ введения избыточности связан с добавлением дополнительных электрических уровней, а в простейшем случае – третьего нулевого уровня.

На рис. 6.1, г приведена форма сигнала с попеременной инверсией знака, так называемого АМІ–сигнала (alternative mark inversion). Нули кодируются отсутствием импульсов, а единицы – попеременно положительными и отрицательными импульсами. Постоянная составляющая равна нулю, проблема передачи последовательности единиц отсутствует, обнаруживаются ошибки нарушающие правильную последовательность знакочередующихся сигналов. Единственная оставшаяся проблема – потеря синхронизации при передаче последовательности нулей, как и в коде  $NRZ$ . Эта проблема решается очень просто: цепочки нулей передатчик заменяет определенными вставками стандартных временных диаграмм. Коды  $AMI$ , в которых цепочка из  $N$  нулей заменяется определенной подстановкой, называются  $BZS$ –кодами (bipolar with  $N$  zeroes substitution).

В коде  $BZS$  каждые три последовательно расположенных нуля подменяются либо комбинацией  $BOV$ , либо  $OOV$ . Символ  $B$  обозначает импульс, который отвечает правилам кодирования  $AMI$  (совпадает по полярности с предыдущим). Выбор одной из этих двух вставок производится так, чтобы, во-первых, число импульсов  $B$  между двумя последовательно расположенными импульсами  $V$  было нечетным и, во-вторых, чтобы полярность  $V$  чередовалась (рис. 6.1, д).

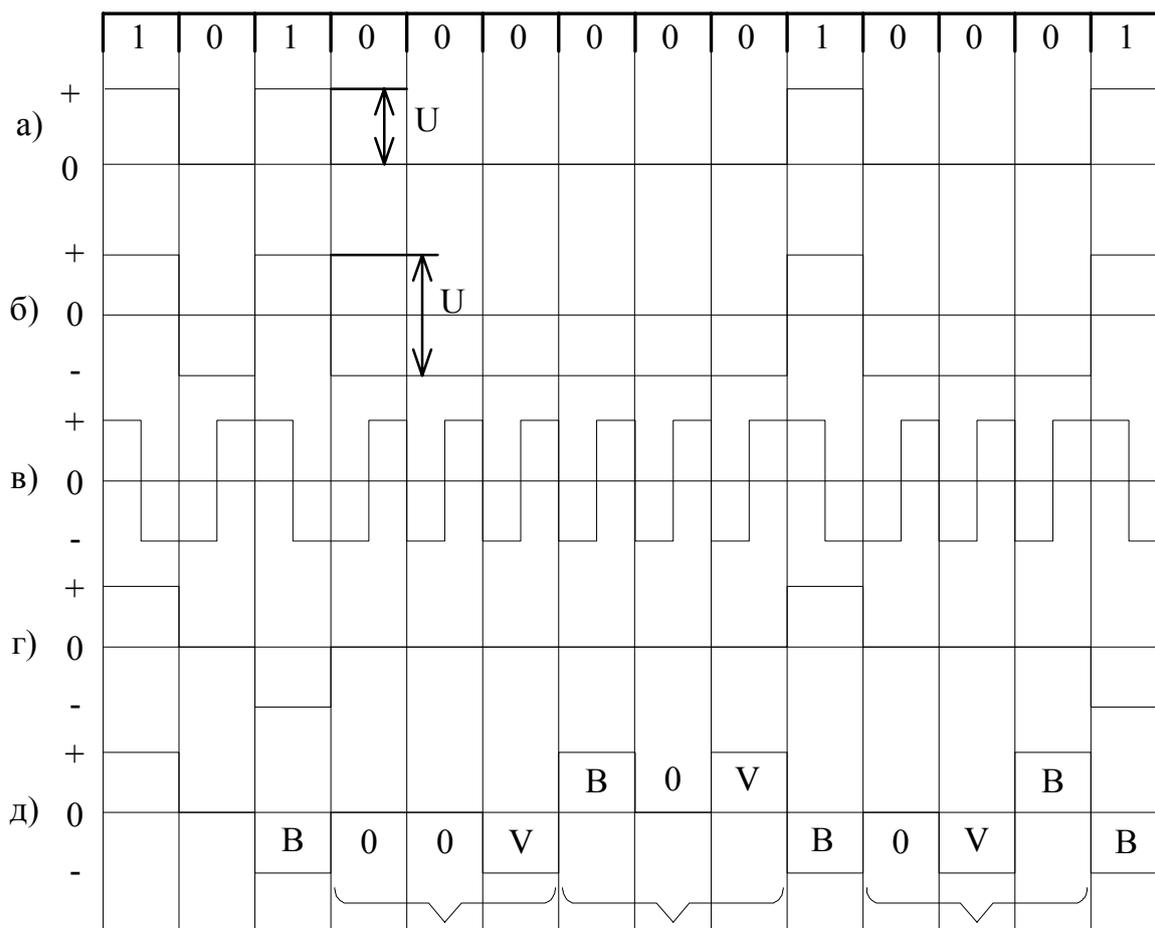


Рис. 6.1. Линейные коды:

а - униполярный код NRZ; б - биполярный код NRZ;  
 в - код "Манчестер - II"; г - код AMI; д - код B3ZS

Существуют также другие распространенные коды, такие как *CMI*, *PST*, *4B3T* и т.п. Все они являются разновидностями кода *AMI* и созданы с целью минимизации требований к полосе пропускания каналов связи и увеличения обнаруживающей способности по отношению к ошибкам при передаче информации.

## 6.2. Шифратор и дешифратор кода Манчестер–2

Сигнал в коде Манчестер–2 может быть получен суммированием по модулю 2 сигналов *NRZ* и синхросигнала *C*. Другими словами, сигнал, представленный в коде Манчестер–2, принимает единичные значения в тех интервалах времени, в которых сигналы *NRZ* и *C* имеют противоположные логические значения (01 или 10). Вследствие этого схема шифратора кода Манчестер–2 чрезвычайно проста (рис. 6.2).

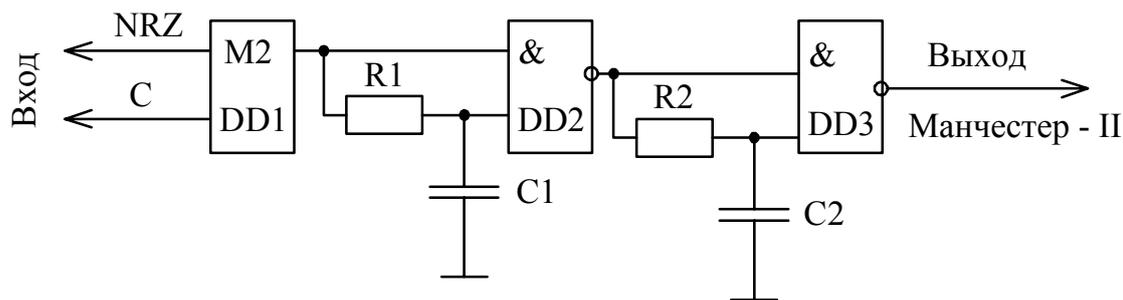


Рис. 6.2. Шифратор кода Манчестер - 2

Временные диаграммы работы шифратора показаны на рис. 6.3. Схема подавления помех ( $R1C1$  и  $R2C2$ ) предназначена для фильтрации результирующего сигнала от кратковременных импульсов, которые могут возникнуть из-за неидеального совпадения отрицательного фронта сигнала  $C$  с отрицательным или положительным фронтом сигнала  $NRZ$ .

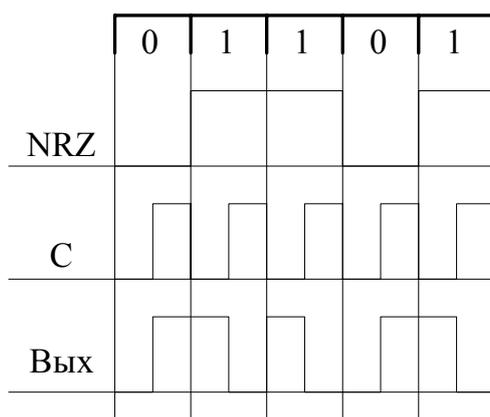


Рис. 6.3. Временные диаграммы работы шифратора

Дешифратор кода Манчестер-2 представляет из себя более сложную схему, содержащую формирователь импульсов  $DD0$ , счетный триггер  $DD1$  и  $D$ -триггер  $DD2$  (рис. 6.4). Как следует из временной диаграммы, приведенной на рис. 6.5, отрицательные импульсы 1 на выходе формирователя импульсов возникают всякий раз, когда сигнал Манчестер-2 меняет свое значение ( $0 \rightarrow 1$  или  $1 \rightarrow 0$ ). Сигнал 2 получают из сигнала 1 с помощью логической схемы. Так как импульс 2 поступает на установочный вход  $S$  счетного триггера  $DD1$ , то в момент  $t_0$  этот триггер обязательно перейдет в единичное состояние и в даль-

нейшем сигнал  $C^*$ , снимаемый с его инверсного выхода, будет в точности повторять сигнал  $C$ , выдаваемый ПЭВМ.

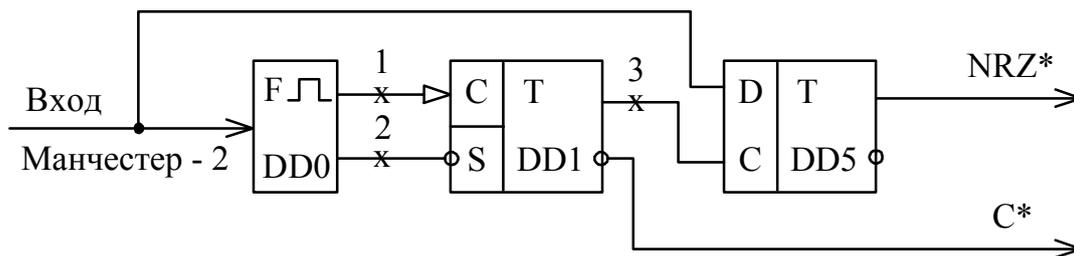


Рис. 6.4. Дешифратор кода Манчестер - 2

Начиная с момента  $t_1$ , т.е. по прошествии одного периода тактовых импульсов от момента  $t_0$ , код  $NRZ^*$ , снимаемый с выхода триггера  $DD2$ , полностью совпадает с кодом  $NRZ$ , поступающим из ПЭВМ на шифратор (с точностью до задержки передачи).

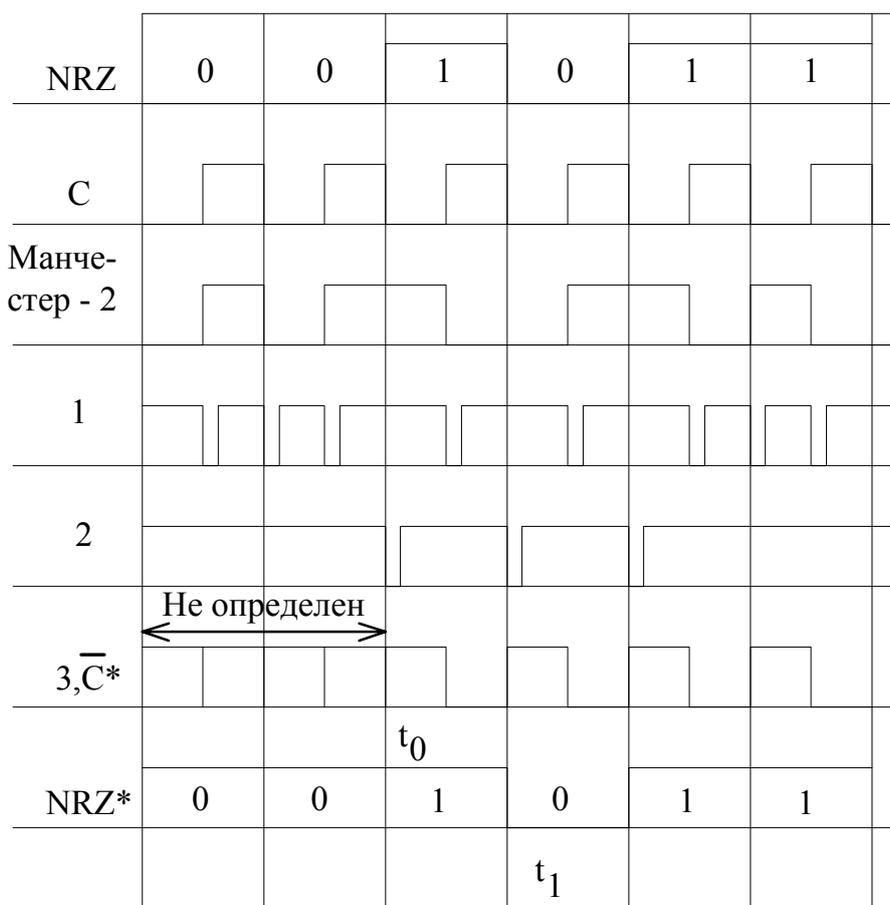


Рис. 6.5. Временные диаграммы работы дешифратора

Таким образом, чтобы заставить приемник войти в синхронизм с передатчиком, достаточен переход сигнала на линии NRZ из 0 в 1. Последующая цепочка бит любой длины, передаваемая по линии NRZ, будет в точности повторена на линии NRZ\* приемника. Это же относится и к синхросигналам: сигнал C\* в точности повторяет исходный сигнал C.

Рассмотрев шифратор и дешифратор кода Манчестер–2, укажем теперь более подробно преимущества данного кода перед кодом NRZ:

1) синхросигнал и информация передаются по одному каналу, в то время как при использовании кода NRZ нужны два канала;

2) диапазон логических частот NRZ начинается от нуля и не превышает половины тактовой частоты (рис. 6.6). Сигнал Манчестер–2 содержит только две логические составляющие  $\frac{f_c}{2}$  и  $f_c$ . Постоянная составляющая при использовании биполярных сигналов равна нулю. Из этого следует, что приемник кода Манчестер–2 может быть узкополосным и поэтому более помехоустойчивым;

3) критерием ошибки передачи является наличие постоянного уровня сигнала в течение времени, превышающего один период тактовой частоты (в коде NRZ подобного критерия не существует). При наличии стартового импульса, равного 1,5 периода критерий ошибки пересматривается;

4) побитовая синхронизация, рассогласование синхронизации может достигать 25 %, а не 4 % и не зависит от длины посылки;

5) при передаче по волоконно–оптическим линиям связи обеспечивается возможность работы светоизлучающего элемента с двукратной перегрузкой по мощности, так как в среднем 50% времени элемент находится в выключенном состоянии.

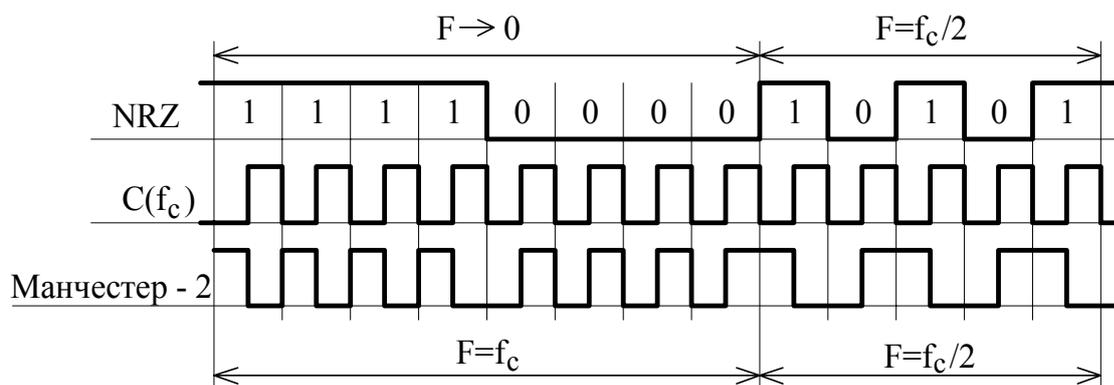


Рис. 6.6. Сравнение частотных характеристик сигналов NRZ и Манчестер - 2

Недостатком кода Манчестер–2 является удвоенная по сравнению с необходимой пропускная способность.

## ЗАКЛЮЧЕНИЕ

Дальнейшее развитие теории и техники кодирования и декодирования кодов связано прежде всего, с появлением новых дискретных кодов и расширением сфер их применения. Существенное влияние на процесс в этой области оказывает также и состояние элементной базы. В конспекте лекций применительно к аппаратным методам преобразования кодов рассмотрены некоторые тенденции выполнения кодеров и декодеров на специализированных ИС, микросхемных матрицах ПЗУ, сумматорах и т.д. В ближайшие годы нужно ожидать значительного расширения номенклатуры заказных БИС для преобразования кодов. Что касается преобразователей кодов с параллельным вводом информации, отличающихся повышенным быстродействием, то при их разработке и практической реализации возникают определенные трудности с увеличением разрядности кодов. Эти трудности могут быть преодолены при помощи аппарата автоматизации проектирования цифровых устройств с привлечением ЭВМ. Методика машинного синтеза схем и соответствующее математическое обеспечение в настоящее время разработаны достаточно хорошо и позволяют синтезировать схемы преобразователей кодов практически на любое количество разрядов входного и выходного кодов. Перспективными следует считать программируемые логические матрицы.

Наряду с аппаратными методами совершенствуются и программные способы преобразования кодов. Возможности программных методов существенно расширились с появлением микропроцессоров. В конспекте лекций приведены достаточно четкие алгоритмы процесса преобразования кода в код, которые могут быть использованы при разработке соответствующих программ. В конспекте лекций отмечались ряд ограничений и недостатков, присущих программным методам. Нужно еще добавить, что в программных методах раскрываются лишь алгоритмы преобразования, в то время как аппаратные методы позволяют проследить все аспекты схемотехнической реализации преобразователей кодов, а это представляет интерес для студентов специальности "Автоматика и телемеханика". В целом можно рассчитывать, что для решения подчас сложных задач теории и техники преобразования кодов в ближайшие годы потребуются обращение как к программным, так и к аппаратным методам.

Приведенные в работе схемы не являются единственным решением, а лишь одним из возможных вариантов технического осуществления преобразования кодов.

Неприводимые многочлены и их эквиваленты

Степень	Многочлен	Двоичная последовательность	Степень	Многочлен	Двоичная последовательность	
1	$x+1$	11	7	$x^7+x+1$	10000011	
2	$x^2+x+1$	111		$x^7+x^3+1$	10001001	
3	$x^3+x+1$	1011		$x^7+x^3+x^2+x+1$	10001111	
	$x^3+x^2+1$	1101		$x^7+x^4+x^3+x^2+1$	10011101	
4	$x^4+x+1$	10011		$x^7+x^5+x^2+x+1$	10100111	
	$x^4+x^3+1$	11001		$x^7+x^5+x^3+x+1$	10101011	
	$x^4+x^3+x^2+x+1$	11111		$x^7+x^6+x^3+x+1$	11001011	
5	$x^5+x^2+1$	100101		$x^7+x^6+x^4+x+1$	11010011	
	$x^5+x^3+1$	101001		8	$x^8+x^4+x^3+x+1$	100011011
	$x^5+x^3+x^2+x+1$	101111			$x^8+x^4+x^3+x^2+1$	100011101
	$x^5+x^4+x^2+x+1$	110111	$x^8+x^5+x^3+x+1$		100101011	
	$x^5+x^4+x^3+x+1$	111011	$x^8+x^5+x^3+x^2+1$		100101101	
	$x^5+x^4+x^3+x^2+1$	111101	$x^8+x^6+x^5+x^2+1$		101100101	
			$x^8+x^7+x^3+x+1$		110001011	
6	$x^6+x+1$	1000011	$x^8+x^7+x^5+x^3+1$	110101001		
	$x^6+x^3+1$	1001001	9	$x^9+x+1$	1000000011	
	$x^6+x^4+x^2+x+1$	1010111		$x^9+x^4+1$	1000010001	
	$x^6+x^4+x^3+x+1$	1011011		$x^9+x^4+x^2+x+1$	1000010111	
	$x^6+x^5+1$	1100001		$x^9+x^4+x^3+x+1$	1000011011	
	$x^6+x^5+x^2+x+1$	1100111		$x^9+x^5+x^4+x+1$	1000110011	
	$x^6+x^5+x^3+x^2+1$	1101101		$x^9+x^6+x^5+x^2+1$	1001100101	
	$x^6+x^5+x^4+x+1$	1110011		10	$x^{10}+x^3+1$	10000001001
	$x^6+x^5+x^4+x^2+1$	1110101				

Минимальные многочлены циклических кодов

Номер $M(x)$	Минимальные многочлены различных степеней, записанные в виде кодовых комбинаций.								
	2	3	4	5	6	7	8	9	
$M_1(x)$	111	1011	10011	100101	1000011	10001001	100011101	1000010001	
$M_3(x)$		1101	11111	111101	1010111	10001111	101110111	1001011001	
$M_5(x)$			111	110111	1100111	10011101	111110011	1100110001	
$M_7(x)$			11001	101111	1001001	11110111	101101001	1010011001	
$M_9(x)$				110111	1101	10111111	110111101	1100010011	
$M_{11}(x)$				111011	1101101	11010101	111100111	1000101101	
$M_{13}(x)$						10000011	100101011	1001110111	

## Параметры циклических кодов БЧХ

$n$	$k$	$s$	$r$	Образующий многочлен
7	4	1	3	13
15	11	1	4	23
	7	2	8	721
	5	3	10	2467
31	26	1	5	45
	21	2	10	3551
	16	3	15	107657
	11	5	20	5423325
	6	7	25	313365047
63	57	1	6	103
	51	2	12	12471
	45	3	18	1701317
	39	4	24	166623567
	36	5	27	1033500423
	30	6	33	1574641656547
	24	7	39	17323260404441
	18	10	45	1363026512351725
127	120	1	7	211
	113	2	14	41567
	106	3	21	11554743
	99	4	28	3447023271
	92	5	35	624730022327
	85	6	42	130704476332273
	78	7	49	26230002166130115
	71	9	56	6255010713253127753
	64	10	63	1206534025570773100045
255	247	1	8	435
	239	2	16	267543
	231	3	24	156720665
	223	4	32	75626641375
	215	5	40	2315754726421
	207	6	48	16176560567636227
	199	7	56	7633031270420722341
	191	8	64	2663470176115333714567
	187	9	68	52755313540001322236351
	179	10	76	22624710717340432416300455

## ЛИТЕРАТУРА

1. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976.
2. Дмитриев В.И. Прикладная теория информации. М.: Высш. шк., 1989. 320 с.
3. Аршинов М.Н., Садовский Л.Е. Коды и математика. М.: Наука, 1983. 144 с.
4. Колесников В.Д., Мирончиков Е.Т. Декодирование циклических кодов. М.: Связь, 1968.
5. Новик А.А. Эффективное кодирование. М.: Энергия, 1965.
6. Хемминг Р.В. Теория кодирования и теория информации. М.: Радио и связь, 1983.
7. Тутевич В.Н. Телемеханика. М.: Высш. шк., 1985. 423 с.
8. Пшеничников А.М., Портнов М.Л. Телемеханические системы на интегральных микросхемах. М.: Энергия, 1977. 296 с.
9. Гуров В.С., Емельянов Г.А., Етрухин Н.Н., Осипов В.Г. Передача дискретной информации и телеграфия. М.: Связь, 1974. 526 с.
10. Мак–Вильямс Ф., Слоэн Н.Дж. Теория кодов исправляющих ошибки. М.: Связь, 1979
11. Касами Т., Токура Н. и др. Теория кодирования. М.: Мир; 1978.
12. Берлекэмп Э. Алгебраическая теория кодирования. М.: Мир, 1971.
13. Блох Э.Л., Зяблов В.В. Обобщенные каскадные коды. М.: Связь, 1976.
14. Марков А.А. Введение в теорию кодирования. М.: Наука, 1982.
15. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения: Справочник. М.: Радио и связь, 1990. 512 с.
16. Богданович М.Н. и др. Цифровые интегральные микросхемы: Справочник. Мн.: Беларусь, 1991. 492 с.
17. Пухальский Г.Н., Новосельцева Т.Я. Проектирование дискретных устройств на интегральных микросхемах: Справочник. М.: Радиосвязь, 1990. – 304 с.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ	1
1. КОДЫ И КОДИРОВАНИЕ	2
1.1. Основные понятия	2
1.2. Цифровые коды	5
1.3. Простые двоичные коды	12
1.4. Оптимальные коды	17
2. КОРРЕКТИРУЮЩИЕ КОДЫ	24
2.1. Основные понятия	24
2.2. Коды с обнаружением ошибок	27
2.3. Коды с обнаружением и исправлением ошибок	34
2.4. Частотные коды	67
3. ТЕХНИЧЕСКИЕ СРЕДСТВА ПРЕОБРАЗОВАНИЯ ДЛЯ НЕПОМЕХОЗАЩИЩЕННЫХ КОДОВ	69
3.2. Дешифратор двоичного кода в десятичный код	73
3.3. Дешифратор двоично–десятичного кода в десятичный	77
3.4. Преобразователи двоичного кода в двоично–десятичный код и обратно	79
3.5. Преобразователь двоичного кода 8–4–2–1 в самодополняющийся двоично–десятичный код 2–4–2–1	85
3.6. Преобразователь самодополняющегося двоично–десятичного кода 2–4–2–1 в двоичный код 8–4–2–1	86
3.7. Преобразователь кода Грея в двоичный код и обратно	88
3.8. Технические средства кодирования и декодирования эффективных кодов	92
3.9. Схемы равнозначности кодов	94
3.10. Преобразователь параллельного кода в последовательный и обратно	97
4. ТЕХНИЧЕСКИЕ СРЕДСТВА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ КОРРЕКТИРУЮЩИХ КОДОВ	101
4.1. Кодер и декодер кода с защитой на четность	101
4.2. Кодер и декодер кода с постоянным весом	103
4.3. Кодер и декодер кода с двумя проверками на четность	105
4.4. Кодер и декодер кода с повторением	107
4.5. Кодер и декодер кода с числом единиц, кратным трем	110
4.6. Кодер и декодер инверсного кода	112
4.7. Кодер и декодер корреляционного кода	114
4.8. Кодер и декодер кода Бергера	116
4.9. Кодирование и декодирование устройства систематического кода	119
4.10. Кодирование и декодирование устройства кода Хемминга	121
4.13. Кодер и декодер итеративного кода	133
4.14. Кодер и декодер рекуррентного кода	136
5. ТЕХНИЧЕСКИЕ СРЕДСТВА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ЧАСТОТНЫХ КОДОВ	139
5.1. Кодер и декодер кода на перестановки	139
5.2. Кодер и декодер кода на размещения	142
5.3. Кодер и декодер кода на сочетания	144
5.4. Дешифратор одночастотного кода	146
5.5. Кодер и декодер сменно–качественного кода	147

6. КОДЫ ДЛЯ ПЕРЕДАЧИ ЦИФРОВОЙ ИНФОРМАЦИИ ПО ПОСЛЕДОВАТЕЛЬНЫМ КАНАЛАМ СВЯЗИ _____	152
6.1. Методы кодирования _____	152
6.2. Шифратор и дешифратор кода Манчестер–2 _____	154
ЗАКЛЮЧЕНИЕ _____	158
ЛИТЕРАТУРА _____	162