

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра систем управления

**Н. И. Сорока, Г. А. Кривинченко**

## **ТЕЛЕМЕХАНИКА. МОДУЛЯЦИЯ И КОДИРОВАНИЕ ИНФОРМАЦИИ**

*Допущено Министерством образования Республики Беларусь  
в качестве учебного пособия для студентов учреждений высшего образования  
по специальностям «Информационные технологии и управление  
в технических системах», «Электронные системы безопасности»,  
«Электронные и информационно-управленческие системы  
физических установок»*

В двух частях  
Часть 2

Минск БГУИР 2020

УДК 621.398(075)  
ББК 32.968я73  
С65

**Рецензенты:**

кафедра радиолокации и приемопередающих устройств  
учреждения образования «Военная академия Республики Беларусь»  
(протокол №18 от 19.04.2018);

заведующий кафедрой телекоммуникационных систем  
учреждения образования «Белорусская государственная академия связи»  
кандидат технических наук, доцент С. И. Половения

**Сорока, Н. И.**

С65 Телемеханика. Модуляция и кодирование информации : учеб. пособие.  
В 2 ч. Ч. 2 / Н. И. Сорока, Г. А. Кривинченко. – Минск : БГУИР, 2020. –  
185–376 с. : ил.  
ISBN 978-985-543-537-3 (ч. 2).

Рассмотрены спектры периодических и непериодических сигналов, методы и технические средства модуляции и демодуляции, применяемые в системах телемеханики. Большое внимание уделено получению соотношений для инженерных расчетов и оценок. Приведены основные характеристики кодов, дана классификация двоичных кодов, указаны методы представления кодов, рассмотрены технические средства преобразований для непомяхозащищенных кодов, кодирования и декодирования корректирующих и частотных кодов.

Предназначено для студентов, а также магистрантов, аспирантов и специалистов, занимающихся проектированием, созданием и эксплуатацией телемеханических комплексов и систем передачи информации.

Издано в двух частях. Часть 2 издания включает разделы 6–10 учебного пособия. Табл. : 40. Ил. : 103. Прил. : 4. Библиогр. : 22.

**УДК 621.398(075)  
ББК 32.968я73**

**ISBN 978-985-543-537-3 (ч. 2)  
ISBN 978-985-543-535-9**

© Сорока Н. И., Кривинченко Г. А., 2020  
© УО «Белорусский государственный  
университет информатики  
и радиоэлектроники», 2020

## Содержание

6 Коды и кодирование .....	187
6.1 Основные понятия .....	188
6.2 Цифровые коды .....	190
6.2.1 Запись кодовых комбинаций в виде многочлена .....	190
6.2.2 Сложение .....	191
6.2.3 Умножение .....	192
6.2.4 Деление .....	193
6.2.5 Перенос слагаемых .....	194
6.2.6 Матричная запись кодовых комбинаций.....	194
6.2.7 Графическое представление кода.....	195
6.2.8 Геометрическая модель кода .....	195
6.2.9 Классификация двоичных кодов .....	197
6.2.10 Основные характеристики двоичных кодов .....	198
6.3 Простые двоичные коды .....	199
6.3.1 Двоичный код на все сочетания .....	199
6.3.2 Единично-десятичный код.....	199
6.3.3 Двоично-десятичный код.....	199
6.3.4 Числоимпульсный код.....	200
6.3.5 Код Джонсона .....	200
6.3.6 Код Грея.....	200
6.4 Оптимальные коды.....	204
6.4.1 Код Шеннона.....	204
6.4.2 Код Шеннона – Фано.....	207
6.4.3 Код Хаффмана.....	208
7 Корректирующие коды.....	209
7.1 Основные понятия.....	209
7.2 Коды с обнаружением ошибок.....	212
7.2.1 Код с постоянным весом (код на одно сочетание) .....	212
7.2.2 Распределительный код с $C_n^1$ .....	212
7.2.3 Код с проверкой на четность .....	213
7.2.4 Код с проверкой на нечетность .....	213
7.2.5 Код с двумя проверками на четность .....	214
7.2.6 Код с повторением.....	214
7.2.7 Код с числом единиц, кратным трем .....	215
7.2.8 Инверсный код (код с повторением и инверсией) .....	216
7.2.9 Корреляционный код (код с удвоением числа элементов).....	218
7.2.10 Код Бергера .....	218
7.3 Коды с обнаружением и исправлением ошибок .....	219
7.3.1 Систематические коды .....	219
7.3.2 Код Хемминга .....	223
7.3.3 Циклические коды .....	227
7.3.4 Итеративные коды .....	253
7.3.5 Рекуррентные коды.....	256
7.3.6 Сверточные коды .....	259
7.3.7 Турбокоды .....	264
7.4 Коды Рида – Соломона .....	268
7.4.1 Общие сведения .....	268

7.4.2 Кодирование сообщений в коде Рида – Соломона.....	274
7.4.3 Декодирование кодов Рида – Соломона.....	277
7.5 Частотные коды.....	278
7.5.1 Одночастотный код.....	278
7.5.2 Коды, образованные по закону перестановок.....	278
7.5.3 Коды, образованные по закону размещений.....	278
7.5.4 Коды на определенное число сочетаний.....	279
7.5.5 Сменно-качественные коды.....	279
8 Технические средства преобразования для непомяхозащищенных кодов.....	280
8.1 Шифраторы кода $C_n^1$ в двоичный код.....	280
8.2 Дешифратор двоичного кода в десятичный код.....	284
8.3 Дешифратор двоично-десятичного кода в десятичный.....	288
8.4 Преобразователи двоичного кода в двоично-десятичный код и обратно.....	290
8.5 Преобразователь двоичного кода 8-4-2-1 в самодополняющийся двоично-десятичный код 2-4-2-1.....	295
8.6 Преобразователь самодополняющегося двоично-десятичного кода 2-4-2-1 в двоичный код 8-4-2-1.....	296
8.7 Преобразователь кода Грея в двоичный код и обратно.....	298
8.8 Технические средства кодирования и декодирования эффективных кодов.....	301
8.9 Схемы равнозначности кодов.....	304
9 Технические средства кодирования и декодирования корректирующих кодов.....	310
9.1 Кодер и декодер кода с защитой на четность.....	310
9.2 Кодер и декодер кода с постоянным весом.....	312
9.3 Кодер и декодер кода с двумя проверками на четность.....	314
9.4 Кодер и декодер кода с повторением.....	316
9.5 Кодер и декодер кода с числом единиц, кратным трем.....	319
9.6 Кодер и декодер инверсного кода.....	320
9.7 Кодер и декодер корреляционного кода.....	323
9.8 Кодер и декодер кода Бергера.....	325
9.9 Кодирование и декодирование устройства систематического кода.....	328
9.10 Кодирование и декодирование устройства кода Хемминга.....	330
9.11 Технические средства умножения и деления многочлена на многочлен.....	332
9.12 Кодер и декодер циклического кода.....	335
9.13 Кодер и декодер итеративного кода.....	343
9.14 Кодер и декодер рекуррентного кода.....	345
10 Технические средства кодирования и декодирования частотных кодов.....	349
10.1 Кодер и декодер кода на перестановки.....	349
10.2 Кодер и декодер кода на размещения.....	351
10.3 Кодер и декодер кода на сочетания.....	353
10.4 Дешифратор одночастотного кода.....	355
10.5 Кодер и декодер сменно-качественного кода.....	356
Приложение В Неприводимые многочлены и их эквиваленты.....	360
Приложение Г Минимальные многочлены циклических кодов.....	361
Приложение Д Параметры циклических кодов БЧХ.....	362
Приложение Е Экспресс-тесты для проверки усвоения лекционного материала.....	363
Литература.....	374

## 6 КОДЫ И КОДИРОВАНИЕ

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 единицы или нуля;

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;

б) 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, допускающему восстановление исходной информации. Теория кодирования получила большое развитие, начиная с 40-х гг. XX в., после работ К. Шеннона.

В части 2 учебного пособия 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 (схмотехническому) способу преобразования кодов, что связано в первую очередь с разработкой специализированных микросхем, а также интегральных схем среднего и большого уровней интеграции. В разделах 6–10 данного учебного пособия 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, задаваемому 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, что также освещено в учебном пособии.

## 6.1 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, коды 001, 011, 101 являются комплектными, а коды 1, 11, 101 – **некомплектными**. В телемеханике обычно используют только равномерные коды.

Кроме указанных характеристик, коды имеют и другие характеристики, которые приведены на рисунке 6.1.

Для передачи различных символов, составляющих алфавит кода, могут использоваться импульсы с различными признаками (таблица 6.1).

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.

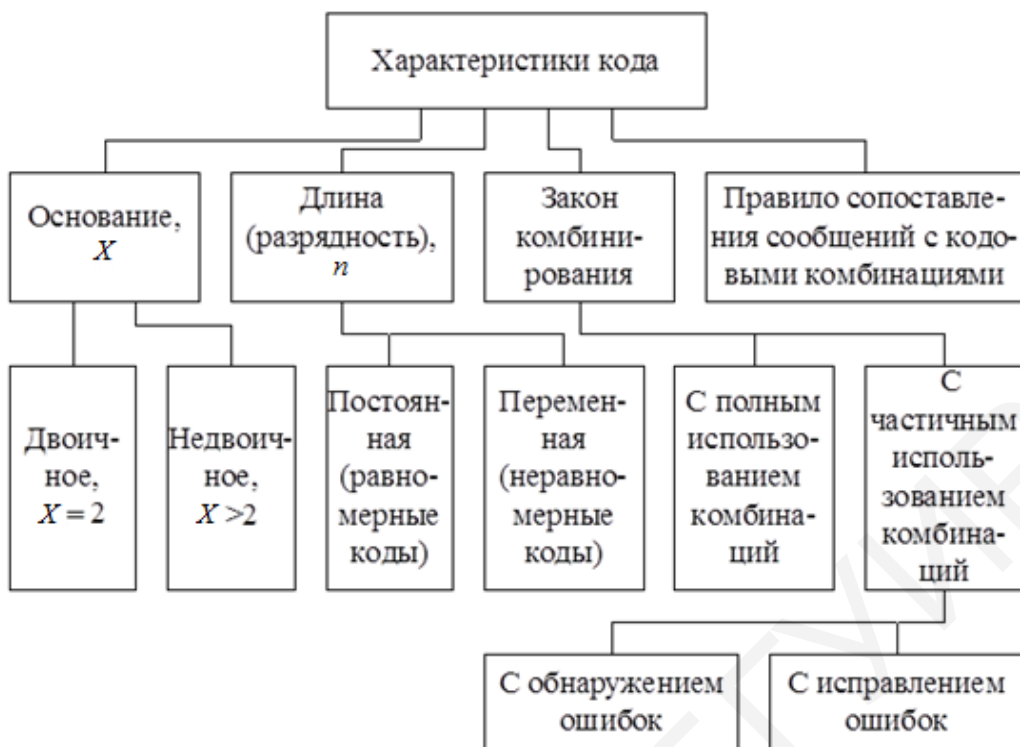


Рисунок 6.1 – Классификация характеристик кода

Таблица 6.1 – Импульсные признаки, используемые для передачи двоичных кодов

Символ	Признаки										
	Амплитудные		Временные			Полярные	Частотные		Фазовые		
1											
0	—	—									

На рисунке 6.2 показана последовательная передача кодовой комбинации 1011001 видеоимпульсами, а на рисунке 6.3 – 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 (таблица 6.2). Однако признаки у каждого разряда должны быть не частотными, а амплитудными или временными.

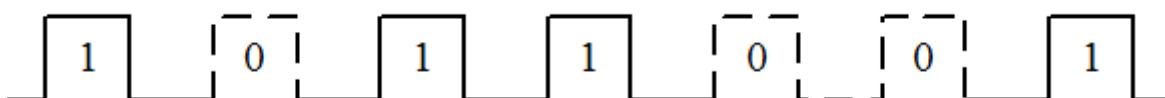







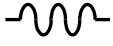
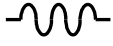


Рисунок 6.2 – Последовательная передача кодовой комбинации видеоимпульсами

Таблица 6.2 – Параллельная передача кодовой комбинации

Номер разряда	Частота	Номер кодовой комбинации и время ее 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 – в течение второго интервала времени одновременной посылкой частот  $f_1, f_2$  и  $f_3$ .

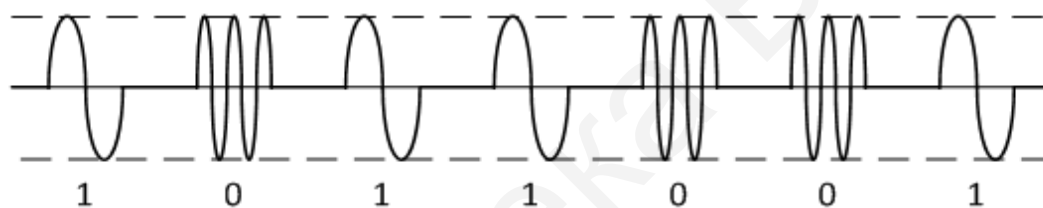


Рисунок 6.3 – Последовательная передача кодовой комбинации радиоимпульсов

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

## 6.2 Цифровые коды

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

### 6.2.1 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



$$F(X) = \sum_{i=0}^{n-1} A_i X^i = A_{n-1} X^{n-1} + A_{n-2} X^{n-2} + \dots + A_0 X^0, \quad (6.1)$$

где  $A$  – цифровые коэффициенты, имеющие значения от 0 до  $X-1$ .  
В десятичной системе ( $X=10$ )

$$F(10) = \sum_{i=0}^{n-1} A_i 10^i. \quad (6.2)$$

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

$$F(2) = \sum_{i=0}^{n-1} A_i 2^i. \quad (6.3)$$

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

$$G(x) = 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^5 + x^3 + x^2 + x^1 + 1. \quad (6.4)$$

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

### 6.2.2 Сложение

Над многочленами можно производить все алгебраические операции. Обычное сложение с переносом числа в старший 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:

	$x^6$	+	$x^5$	+	0	+	$x^3$	+	$x^2$	+	0	+	1	$\rightarrow 1101101$	$\rightarrow 109_{10}$
⊕	0	+	$x^5$	+	$x^4$	+	0	+	$x^2$	+	0	+	0	$\rightarrow 0110100$	$\rightarrow 52_{10}$
⊕	$x^6$	+	$x^5$	+	$x^4$	+	$x^3$	+	$x^2$	+	X	+	1	$\rightarrow 1111111$	$\rightarrow 127_{10}$
	0	+	$x^5$	+	0	+	0	+	$x^2$	+	X	+	0	$\rightarrow 0100110$	$\rightarrow 38_{10}$

### 6.2.3 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 на втором этапе.

Перемножим два многочлена:

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

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

### 6.2.4 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,

$$\begin{array}{r}
 110110101 \quad | \quad 101011 \\
 101011 \quad | \\
 \hline
 111011 \\
 101011 \\
 \hline
 100000 \\
 101011 \\
 \hline
 10111 \quad \rightarrow x^4 + x^2 + x + 1 - \text{остаток}
 \end{array}
 \quad \begin{array}{l}
 \hline
 1110 \rightarrow x^3 + x^2 + x - \text{частное } Q(X)
 \end{array}$$

При составлении циклических кодов необходимо уметь находить только остатки без определения частного. Ниже дается пример нахождения нескольких остатков при делении единиц 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.

$$\begin{array}{r}
 1000000000 \quad | \quad 1011 \\
 1011 \quad | \\
 \hline
 01100 \quad \text{— первый остаток } R_1(X) = 011 \\
 1011 \quad \text{— второй остаток } R_2(X) = 110 \\
 \hline
 1110 \quad \text{— третий остаток } R_3(X) = 111 \\
 1011 \quad | \\
 \hline
 1010 \quad \text{— четвертый остаток } R_4(X) = 101 \\
 1011 \quad | \\
 \hline
 00100 \quad \text{— пятый остаток } R_5(X) = 001 \\
 \quad \quad \text{шестой остаток } R_6(X) = 010 \\
 \quad \quad \text{седьмой остаток } R_7(X) = 100
 \end{array}$$

Дальнейшее деление нецелесообразно, так как остатки начнут повторяться.

### 6.2.5 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.

### 6.2.6 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  $a$ :

$$\begin{array}{l}
 \text{a) } \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix} \quad
 \text{б) } \begin{array}{l} \square \begin{array}{l} 111 \\ 110 \\ 001 \end{array} \\ \\ \square \begin{array}{l} 010 \\ 100 \\ 110 \end{array} \end{array} \quad
 \text{в) } \begin{array}{l} \square \begin{array}{l} 001 \\ 011 \\ 010 \end{array} \\ \\ \square \begin{array}{l} 001 \\ 110 \\ 111 \end{array} \end{array} \quad
 \text{г) } \begin{array}{l} \square \begin{array}{l} 001 \\ 010 \\ 011 \end{array} \\ \\ \square \end{array} \quad
 \text{д) } \begin{array}{l} \square \begin{array}{l} 010 \\ 110 \\ 100 \end{array} \\ \\ \square \end{array} \quad
 \text{е) } \begin{array}{l} \square \begin{array}{l} 011 \\ 110 \\ 101 \end{array} \\ \\ \square \end{array} \\
 \text{ж) } \begin{array}{l} \square \begin{array}{l} 010 \\ 100 \\ 110 \end{array} \\ \\ \square \end{array} \quad
 \text{з) } \begin{array}{l} \square \begin{array}{l} 001 \\ 110 \\ 111 \end{array} \\ \\ \square \end{array} \quad
 \text{и) } \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad
 \text{к) } \begin{vmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{vmatrix}
 \end{array}$$

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  $u$ ). Матрица  $k$  является транспонированной единичной матрицей, т. е. зеркальным отображением матрицы  $u$ . Интересным свойством обладает единичная матрица  $u$ : 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.

## 6.2.7 Графическое представление кода

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 максимальный порядок вершин, которые оно содержит, равен максимальной длине кодовых комбинаций. Значения разрядов комбинации, приписываемой каждой вершине, соответствуют 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 рисунке 6.4 показано кодовое дерево для двоичного трехразрядного кода.

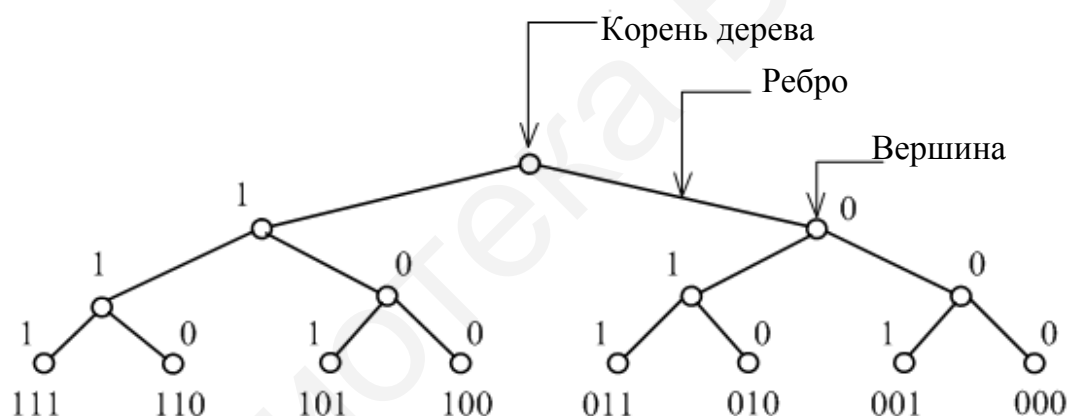


Рисунок 6.4 – Кодовое дерево для двоичного трехразрядного кода

## 6.2.8 Геометрическая модель кода

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 единице.

При  $n = 2$  кодовые комбинации располагаются в вершинах квадрата (рисунок 6.5), при  $n = 3$  – в вершинах единичного куба (рисунок 6.6).

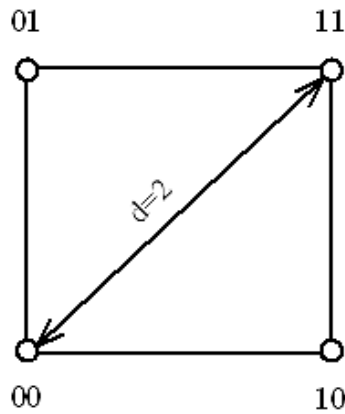


Рисунок 6.5 – Геометрическая модель двухразрядного кода

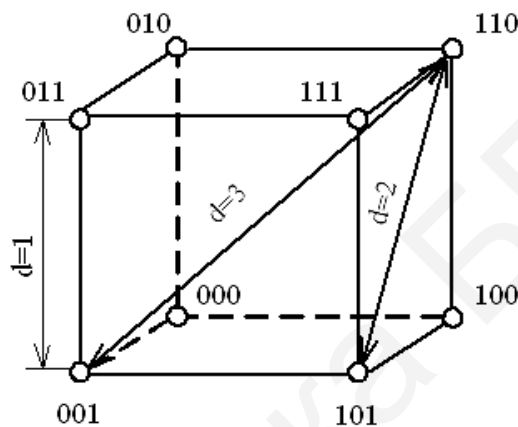


Рисунок 6.6 – Геометрическая модель трехразрядного кода

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.

На рисунках 6.7 и 6.8 представлены геометрические модели троичного двухразрядного и трехразрядного кодов соответственно.

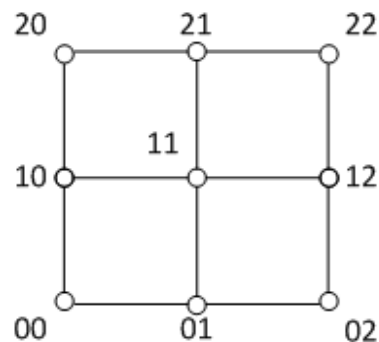


Рисунок 6.7 – Геометрическая модель троичного двухразрядного кода

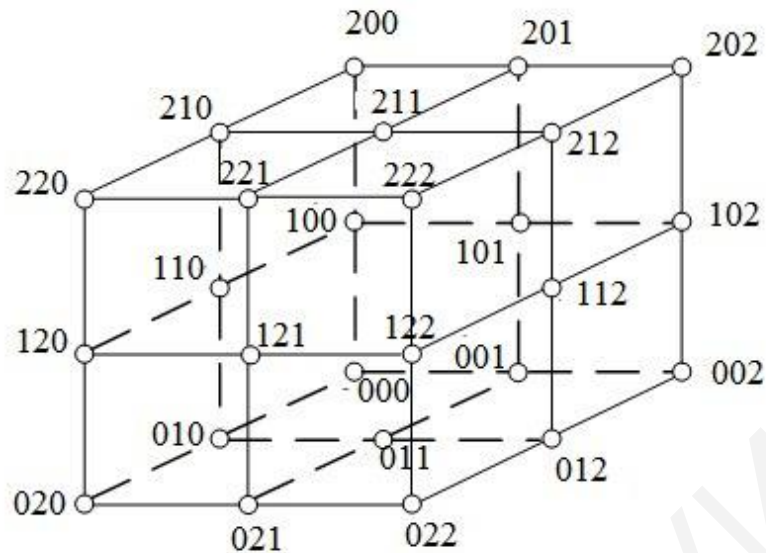


Рисунок 6.8 – Геометрическая модель троичного трехразрядного кода

### 6.2.9 Классификация двоичных кодов

По возможности обнаружения и исправления ошибок различают простые и корректирующие коды. Под простыми кодами следует понимать коды, не позволяющие обнаружить искажения. Корректирующие коды делятся на две группы: позволяющие только обнаружить искажения и позволяющие обнаружить и исправить искажения. Дальнейшая классификация приведена на рисунке 6.9.



Рисунок 6.9 – 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 – 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 длиной  $l$ , на которые разделяется блок информационных символов. У систематических кодов проверочные символы определяются в результате проведения линейных операций над определенными информационными символами.

#### 6.2.10 Основные характеристики двоичных кодов

Двоичные коды характеризуются весом кода  $w$ , кодовым расстоянием  $d$  и весовой характеристикой  $F(w)$ . *Весом кода  $w$*  называется количество единиц в кодовой комбинации. Например, для кодовой комбинации 1011110 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 кодового 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, кодовое 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 весом 1, двух слов весом 3 и одного слова весом 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 избыточ-



ность рассчитывают по формуле  $R = I - \frac{\log_2 N_p}{\log_2 N}$ , где  $N_p$  – число кодовых комбинаций, используемых для передачи сообщений (рабочая мощность кода);  $N$  – полное число кодовых комбинаций (мощность кода).

### 6.3 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.

#### 6.3.1 Двоичный код на все сочетания

Кодовые комбинации этого кода соответствуют записи натурального ряда чисел в двоичной системе счисления. Вес разряда кода определяется из выражения

$$q_i = 2^{i-1}, \quad (6.5)$$

где  $i = 1, 2, 3, \dots, n$ .

Общее число комбинаций

$$N = 2^n. \quad (6.6)$$

#### 6.3.2 Единично-десятичный код

Каждый разряд десятичного числа записывается в виде соответствующего числа единиц (таблица 6.3). При этом разряды разделяются интервалами. Например,  $24 \rightarrow 11\ 1111$ . 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,  $24 \rightarrow 000000011\ 000001111$ .

#### 6.3.3 Двоично-десятичный код

Каждый разряд десятичного числа записывается в виде комбинации двоичного кода.

В таблице 6.3 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 комбинаций 0100 и 1011 дает в сумме комбинацию 1111, что соответствует цифре 9.

### 6.3.4 Числоимпульсный код

Иногда числоимпульсный код называют единичным (или унитарным) кодом. Кодовые комбинации отличаются друг от друга числом единиц. Примеры для 12-разрядного кода даны в таблице 6.3. Очевидно, что число кодовых комбинаций в этом коде равно разрядности, т. е.  $N = n$ .

### 6.3.5 Код Джонсона

Этот код применяется в устройствах, преобразующих линейные и угловые перемещения в кодовые комбинации. Записи цифр от 0 до 9 приведены в таблице 6.3. Таким образом, число 137 в коде Джонсона будет представлено в виде 00001 00111 11100.

### 6.3.6 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 %.

Построение кода Грея при отображении десятичных чисел от 0 до 15 четырехразрядным двоичным кодом поясняется в таблице 6.3. 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 первый 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.

Таблица 6.3 – Запись кодовых комбинаций десятичных чисел от 0 до 15 различными кодами

Десятичный код	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	Код Джонсона	Единично-десятично неравномерный код	Единично-десятично равномерный код	Унитарный 12-разрядный код
0	0000	0000	0000	0000	0000	00000	25→11 11111	25→000000011000011111	12→11111111111111
1	0001	0001	0001	0001	0001	00001	14→1 1111	14→0000000010000001111	11→01111111111111
2	0010	0010	0010	0010	0011	00011			8→00001111111111
3	0011	0011	0101	0011	0010	00111			
4	0100	0100	0110	0111	0110	01111			
5	0101	1011	1001	1000	0111	11111			
6	0110	1100	1100	1001	0101	11110			
7	0111	1101	1101	1010	0100	11100			
8	1000	1110	1110	1011	1100	11000			
9	1001	1111	1111	1111	1101	10000			
10	1010	10000	10000	10000	1111	100000			
11	1011	10001	10001	10001	1110	100001			
12	1100	10010	10010	10010	1010	100011			
13	1101	10011	10101	10011	1011	100111			
14	1110	10100	10110	10111	1001	101111			
15	1111	11011	11001	11000	1000	111111			

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

$$q_i = 2^i - 1, \quad (6.7)$$

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:

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

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

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 осуществляться 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

$$a_i = \sum_{j=i}^n b_j \pmod{2}. \quad (6.8)$$

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:

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

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

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:

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

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

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

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

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 примеров значение разряда в коде Грея можно получить из выражения.

$$b_i = a_{i+1} \oplus a_i. \quad (6.9)$$

В качестве примера рассмотрим преобразование двоичного числа 1011001 в код Грея:

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

## 6.4 Оптимальные коды

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

$$L = \sum_{i=1}^n \mu_i P(x_i), \quad (6.10)$$

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

### 6.4.1 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 таблице 6.4.

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

$$P(1) = \frac{L}{L(1)} = \frac{1,4}{2,8} = 0,5; \quad P(0) = \frac{L(0)}{L} = \frac{1,4}{2,8} = 0,5.$$

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.

Таблица 6.4 – Кодирование сообщений кодом Шеннона

Сообщения $x_i$	Вероятность появления сообщений $P(x_i)$	Разбиение сообщений на подмножества					$\mu_i$
		I $x_1, x_2 \rightarrow 0$ $x_3, \dots, x_9 \rightarrow 1$	II $x_2, x_3, x_8 \rightarrow 1$ $x_4, \dots, x_7, x_9 \rightarrow 0$	III $x_3, x_6, x_8 \rightarrow 0$ $x_4, x_5, x_7,$ $x_9 \rightarrow 1$	IV $x_4, x_7, x_8 \rightarrow 0$ $x_5, x_6, x_9 \rightarrow 1$	V $x_5, x_7 \rightarrow 0$ $x_6, x_8, x_9 \rightarrow 1$	
$x_1$	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



### 6.4.2 Код Шеннона – Фано

Для построения этого кода все сообщения  $x_i$  выписываются в порядке убывания их вероятностей (таблица 6.5).

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 очень близок к оптимальному, так как энтропия сообщений

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

Таблица 6.5 – 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

Средняя длина кодового слова:

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

Среднее число нулей:

$$L(0) = 0,70 + 0,30 + 0,39 + 0,27 + 0,36 + 0,32 + 0,20 + 0,20 + 0,10 = 2,84.$$

Среднее число единиц:

$$L(1) = 0,70 + 0,15 + 0,26 + 0,29 + 0,18 + 0,08 + 0,05 + 0,04 = 1,55.$$

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.

### 6.4.3 Код Хаффмана

Для получения кода Хаффмана все сообщения выписывают в порядке убывания вероятностей. Две наименьшие вероятности объединяют скобкой (таблица 6.6) 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 6.6 – 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	1	11
$x_2$	0,15	0,15	0,15	0,17	0,20	0,28	0,35			
$x_3$	0,13	0,13	0,13	0,15	0,17	0,20	0,28	0,28	0	100
$x_4$	0,09	0,09	0,11	0,13	0,15	0,17	0,17			
$x_5$	0,09	0,09	0,09	0,11	0,13	0,13	0,13	0,13	0	001
$x_6$	0,08	0,08	0,09	0,09	0,09	0,09	0,09			
$x_7$	0,05	0,06	0,08	0,08	0,08	0,08	0,08	0,08	0	0110
$x_8$	0,04	0,05	0,05	0,05	0,05	0,05	0,05			
$x_9$	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0	01110

Средняя длина кодового слова (см. таблицу 6.6)  $L=2,82$ , 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 разбиении на подгруппы можно сделать большей по вероятности как верхнюю, так и нижнюю подгруппу.

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

## 7 КОРРЕКТИРУЮЩИЕ КОДЫ

### 7.1 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. рисунок 6.6). Если использовать все восемь кодовых комбинаций, записанных в вершинах куба, то образуется двоичный код на все сочетания. 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 кратностью до  $m$  включительно минимальное кодовое (хеммингово) расстояние между разрешенными кодовыми комбинациями должно быть по крайней мере на единицу больше  $m$ , т. е.

$$d_{\min} \geq m + 1. \quad (7.1)$$

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. рисунок 6.6). В результате возникновения единичной ошибки образуются подмножества:

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

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

Подмножество каждой из разрешенных  $n$ -разрядных комбинаций  $A_i$  (рисунок 7.1) складывается из запрещенных комбинаций, являющихся следствием воздействия:

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

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

Внешняя сфера 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

$$d_{\min} \geq 2S + 1. \quad (7.2)$$

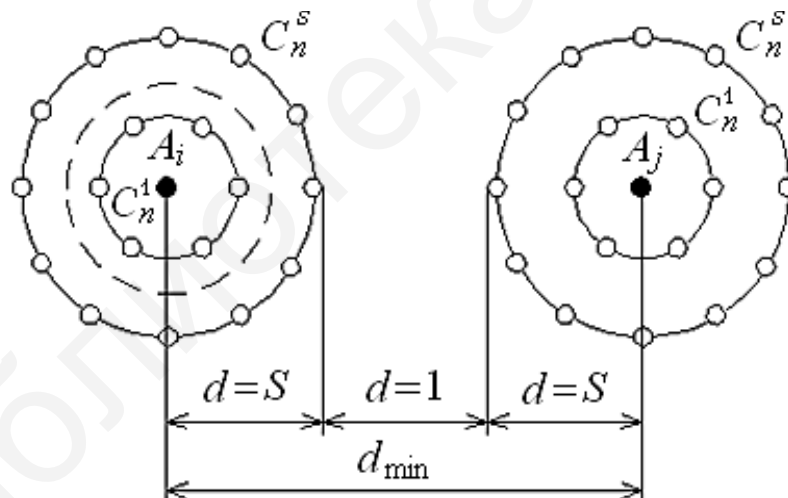


Рисунок 7.1 – Минимальное кодовое расстояние исправления ошибок кратностью  $S$

Нетрудно убедиться в том, что для исправления всех ошибок кратностью  $S$  и одновременного обнаружения всех ошибок кратностью  $m$  ( $m \geq S$ ) (рисунок 7.2) минимальное хеммингово расстояние нужно выбирать из условия

$$d_{\min} \geq m + S + 1. \quad (7.3)$$

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:

$$r_{d=3} = E \log(n+1), \quad (7.4)$$

$$r_{d=3} = E \log((k+1) + E \log(k+1)). \quad (7.5)$$

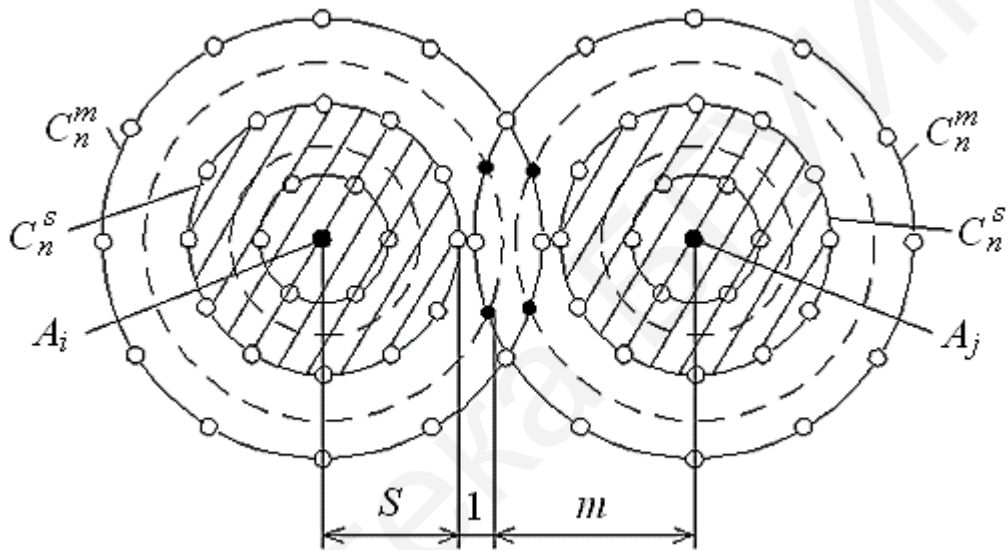


Рисунок 7.2 – Минимальное кодовое расстояние для одновременного исправления ошибок кратностью  $S$  и обнаружения ошибок кратностью  $m$

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.

## 7.2 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 (описаны в пунктах 7.2.1 и 7.2.2);
- 2) коды, в которых используются все комбинации, но к каждой из них по определенному правилу добавляются контрольные  $r$ -символы (описаны в пунктах 7.2.3–7.2.10).

### 7.2.1 Код с постоянным весом (код на одно сочетание)

Общее число кодовых комбинаций в данном коде

$$N = C_n^m = \frac{n!}{m!(n-m)!}, \quad (7.6)$$

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

В таблице 7.1 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.

Таблица 7.1 – Код с постоянным числом единиц

Номер кодовой комбинации	Код $C_4^2$	Код $C_6^1$
1	0011	000001
2	0110	000010
3	1100	000100
4	1001	001000
5	1010	010000
6	0101	100000

### 7.2.2 Распределительный код с $C_n^1$

Это также разновидность кода с постоянным весом, равным единице. Число кодовых комбинаций в данном коде

$$N = C_n^1 = n. \quad (7.7)$$

Кодовая комбинация при  $n=6$  представлена в таблице 7.1 (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.

### 7.2.3 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 таблице 7.2.

Таблица 7.2 – Код с проверкой на четность

Информационные символы $k$	Контрольные символы $r$	Код с проверкой на четность $n=k+r$
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:

$$K_{\text{изб}} = 1 - \frac{k}{k+1} = 1 - \frac{k}{n}. \quad (7.8)$$

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  $k$ -разрядная 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.

### 7.2.4 Код с проверкой на нечетность

Особенностью кода является то, что каждая комбинация содержит нечетное число единиц (таблица 7.3). 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 7.3 – Код с проверкой на нечетность

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

### 7.2.5 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 (таблица 7.4). 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.

Таблица 7.4 – Код с двумя проверками на четность

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

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.

### 7.2.6 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  $(m - 1)/m$ . Слишком высокая избыточность является недостатком кодов с повторением. Даже при двукратном повторении она составляет 0,5:

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

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.

### 7.2.7 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 таблице 7.5.

Таблица 7.5 – Код с числом единиц, кратным трем

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

Код 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.

### 7.2.8 Инверсный код (код с повторением и инверсией)

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 таблице 7.6.

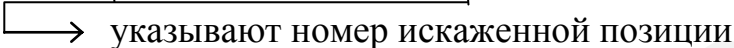
Прием инверсного кода осуществляется в два этапа. На первом этапе суммируются единицы в первой половине кодовой комбинации. Если их количество окажется четным, то вторая половина кодовой комбинации принимается без инверсии, а если нечетным – то с инверсией. На втором этапе обе зарегистрированные комбинации 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.

Таблица 7.6 – 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

Рассмотрим процесс обнаружения ошибок на следующем примере. Пусть передана последняя кодовая комбинация из таблицы 7.6. Ниже показано суммирование для трех вариантов приема переданной комбинации:

$$\begin{array}{r}
 \begin{array}{r}
 111010 \\
 \oplus 111010 \\
 \hline
 000000
 \end{array}
 \qquad
 \begin{array}{r}
 1\dot{0}1010 \\
 \oplus 000101 \\
 \hline
 101111
 \end{array}
 \qquad
 \begin{array}{r}
 111010 \\
 \oplus 1\dot{0}1010 \\
 \hline
 010000
 \end{array}
 \end{array}$$


 указывают номер искаженной позиции

В первом варианте принята комбинация 111010111010. В первой половине кодового слова (информационных символов) 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 два информационных символа и соответствующие им два контрольных, четыре информационных символа и соответствующие им четыре контрольных и т. д.

Коэффициент избыточности инверсного кода равен 0,5.

### 7.2.9 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 символ был 1, записывается 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:

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

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.

### 7.2.10 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

$$r = E \log(k + 1), \quad (7.9)$$

где  $E$  – знак округления в большую сторону.

Примеры составления комбинаций в коде Бергера из обычного шестизрядного двоичного кода представлены в таблице 7.7.

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. Далее показана проверка для шести вариантов приема переданной комбинации из таблицы 7.7.

Таблица 7.7 – Код Бергера

Информационные символы $k$	Контрольные символы $r$				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

Искаженные символы отмечены точкой:

$101011100$      $100 \oplus 100 = 000$  – искажений нет,  
 $10\dot{0}011100$      $011 \oplus 100 = 111$  – искажение обнаружено,  
 $1\dot{1}\dot{0}011100$      $100 \oplus 100 = 000$  – искажение не обнаружено,  
 $1\dot{1}\dot{1}\dot{1}11100$      $110 \oplus 100 = 010$  – искажение обнаружено,  
 $10101110\dot{1}$      $100 \oplus 101 = 001$  – искажение обнаружено,  
 $101011\dot{0}\dot{1}0$      $100 \oplus 010 = 110$  – искажение обнаружено.

Данный код обнаруживает все одиночные и большую часть многократных ошибок.

### 7.3 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.

#### 7.3.1 Систематические коды

Систематическими кодами называются блочные  $(n, k)$ -коды, у которых  $k$  (обычно первые) разрядов представляют собой двоичный избыточный код, а последующие  $n$  разрядов – 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 некото-

рых 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  $k \times k$  и приписанной к ней справа матрицы дополнений размерностью  $k \times r$ :

$$M = \left\| \begin{array}{ccc|ccc} 1 & 0 & 0 \dots 0 & b_{11} & b_{12} & \dots & b_{1r} \\ 0 & 1 & 0 \dots 0 & b_{21} & b_{22} & \dots & b_{2r} \\ 0 & 0 & 1 \dots 0 & b_{31} & b_{32} & \dots & b_{3r} \\ \cdot & \cdot & \dots & \cdot & \cdot & \dots & \cdot \\ 0 & 0 & 0 \dots 1 & b_{k1} & b_{k2} & \dots & b_{kr} \end{array} \right\|. \quad (7.10)$$

Разрядность матрицы дополнений выбирается из выражения (7.4) или (7.5). 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

$$N = \left\| \begin{array}{cccc|ccc} b_{11} & b_{21} & b_{31} \dots b_{k1} & 1 & 0 & 0 \dots 0 \\ b_{12} & b_{22} & b_{32} \dots b_{k2} & 0 & 1 & 0 \dots 0 \\ \cdot & \cdot & \cdot \dots \cdot & \cdot & \cdot & \dots \\ b_{1r} & b_{2r} & b_{3r} \dots b_{kr} & 0 & 0 & 0 \dots 1 \end{array} \right\|. \quad (7.11)$$

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 7.1.** Получить алгоритм кодирования в систематическом коде всех четырехразрядных кодовых комбинаций, позволяющего исправлять единичную ошибку. Таким образом, задано число информационных символов и кратность исправления  $S = 1$ .

**Решение.** По выражению (7.5) определим число контрольных символов при  $k = 4$  :

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

Минимальное кодовое расстояние определим из выражения (7.2):

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

Построим образующую матрицу:

$$M = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Проверочная матрица будет иметь вид

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

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

$$a_5 = a_2 \oplus a_3 \oplus a_4. \quad (7.12)$$

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

$$a_6 = a_1 \oplus a_2 \oplus a_4. \quad (7.13)$$

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

$$a_7 = a_1 \oplus a_3 \oplus a_4. \quad (7.14)$$

Нетрудно убедиться, что все результаты проверок на четность по выражениям (7.12)–(7.14) дают нуль, что свидетельствует о правильности составления образующей и проверочной матриц.

**Пример 7.2.** На основании алгоритма, полученного в примере 7.1, закодировать кодовую комбинацию  $G(x) = 1101 = a_1 a_2 a_3 a_4$  в систематическом коде, позволяющем исправлять одиночную ошибку.

**Решение.** По выражениям (7.12), (7.13) и (7.14) найдем значения для контрольных символов  $a_5$ ,  $a_6$  и  $a_7$ :

$$\begin{aligned}
 a_5 &= 1 \oplus 0 \oplus 1 = 0, \\
 a_6 &= 1 \oplus 1 \oplus 1 = 1, \\
 a_7 &= 1 \oplus 0 \oplus 1 = 0.
 \end{aligned}
 \tag{7.15}$$

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

$$F(x) = 1101010. \tag{7.16}$$

На приемной стороне производятся проверки  $s_i$  принятой кодовой комбинации, которые составляются на основании выражений (7.12)–(7.14):

$$\begin{aligned}
 S_1 &= a_2 \oplus a_3 \oplus a_4 \oplus a_5, \\
 S_2 &= a_1 \oplus a_2 \oplus a_4 \oplus a_6, \\
 S_3 &= a_1 \oplus a_3 \oplus a_4 \oplus a_7.
 \end{aligned}
 \tag{7.17}$$

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  $S_1S_2S_3$ :

$$\begin{array}{l}
 S_1S_2S_3 \\
 0 \ 0 \ 0 \text{ – искажений нет,} \\
 1 \ 0 \ 0 \text{ – искажен символ } a_5, \\
 0 \ 1 \ 0 \text{ – искажен символ } a_6, \\
 0 \ 0 \ 1 \text{ – искажен символ } a_7, \\
 1 \ 1 \ 0 \text{ – искажен символ } a_2, \\
 0 \ 1 \ 1 \text{ – искажен символ } a_1, \\
 1 \ 1 \ 1 \text{ – искажен символ } a_4, \\
 1 \ 0 \ 1 \text{ – искажен символ } a_3.
 \end{array}
 \tag{7.18}$$

**Пример 7.3.** Кодовая комбинация  $F(x) = 1101010$  (пример 7.2) при передаче была искажена и приняла вид  $F'(x) = 1111010 = a_1a_2a_3a_4a_5a_6a_7$ . Декодировать принятую кодовую комбинацию.

**Решение.** Произведем проверки согласно выражениям (7.17):

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



Полученный синдром  $S_1S_2S_3 = 101$  согласно (7.18) 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 7.2.

### 7.3.2 Код Хемминга

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

$$2^r \geq n + 1 = k + r + 1. \quad (7.19)$$

В таблице 7.8 представлена зависимость между  $k$  и  $r$ , полученная из этого неравенства, а также при помощи выражения (7.5).

Чаще всего заданными является число информационных символов, тогда число контрольных символов можно определить из выражения (7.5).

Таблица 7.8 – 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

*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, согласно которому для удобства обнаружения искаженного символа целесообразно размещать контрольные символы на местах, кратных степени 2,

т. е. на позициях 1, 2, 4, 8 и т. д., а информационные символы размещаются на оставшихся местах. Поэтому, например, для девятиэлементной закодированной комбинации можно записать

$$r_1, r_2, k_5, r_3, k_4, k_3, k_2, r_4, k_1, \quad (7.20)$$

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

$k_1$  – младший (первый) разряд.

В принципе, место расположения контрольных символов при передаче по каналу связи не имеет значения: их можно приписывать и перед информационными символами, и после них, и чередуя информационные символы с контрольными.

*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 (7.20):

$$\begin{array}{ll} 0001 - r_1 & 0110 - k_3 \\ 0010 - r_2 & 0111 - k_2 \\ 0011 - k_5 & 1000 - r_4 \\ 0100 - r_3 & 1001 - k_1 \\ 0101 - k_4 & \end{array} \quad (7.21)$$

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,

$$\begin{aligned} r_1 &= k_5 \oplus k_4 \oplus k_2 \oplus k_1, \\ r_2 &= k_5 \oplus k_3 \oplus k_2, \\ r_3 &= k_4 \oplus k_3 \oplus k_2, \\ r_4 &= k_1. \end{aligned} \quad (7.22)$$

В случае кодирования более длинных кодовых комбинаций нужно лишь увеличить число разрядов двоичного кода в колонках (7.21).

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

$$\begin{aligned} S_1 &= r_1 \oplus k_5 \oplus k_4 \oplus k_2 \oplus k_1, \\ S_2 &= r_2 \oplus k_5 \oplus k_3 \oplus k_2, \\ S_3 &= r_3 \oplus k_4 \oplus k_3 \oplus k_2, \\ S_4 &= r_4 \oplus k_1. \end{aligned} \quad (7.23)$$

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 (7.23) 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 (7.23) дали нули. Записываем число  $S_4 S_3 S_2 S_1 = 0011$ , которое означает, что в третьем символе кодовой комбинации (7.20), 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

$$r_1, r_2, k_5, r_3, k_4, k_3, k_2, r_4, k_1, r_5, \quad (7.24)$$

где

$$r_5 = r_1 \oplus r_2 \oplus k_5 \oplus r_3 \oplus k_4 \oplus k_3 \oplus k_2 \oplus r_4 \oplus k_1. \quad (7.25)$$

При декодировании дополнительно к проверкам (7.23) производится проверка

$$S_{\Sigma} = r_1 \oplus r_2 \oplus k_5 \oplus r_3 \oplus k_4 \oplus k_3 \oplus k_2 \oplus r_4 \oplus k_1 \oplus r_5. \quad (7.26)$$

При этом возможны следующие варианты:

- 1) частные проверки (7.23)  $S_i = 0$  и общая (7.25)  $S_{\Sigma} = 0$  – 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 7.4.** Закодировать в коде Хемминга с  $d = 4$  кодовую комбинацию  $G(x) = 10011$  т. е.  $k = 5$ .

**Решение.** Согласно таблице 7.8, число контрольных символов  $r_{d=3} = 4$ , размещаются они на позициях 1, 2, 4 и 8, 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 (7.25). Тогда последовательность в общем виде можно записать так:

$$\begin{matrix} r_1, & r_2, & k_5, & r_3, & k_4, & k_3, & k_2, & r_4, & k_1, & r_5. \\ ? & ? & 1 & ? & 0 & 0 & 1 & ? & 1 & ? \end{matrix} \quad (7.27)$$

Для определения контрольных символов  $r_1 - r_4$  подставим значения  $k_1 - k_5$  в (7.22) 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 (7.25):

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

Таким образом, в линию связи будет послан код  $F(x) = 1011001110$ .

**Пример 7.5.** В приемник поступила кодовая комбинация  $F(x) = 10100011$  в коде Хемминга с  $d = 4$ . Декодировать ее. Если имеются искажения, то обнаружить их и при возможности исправить.

**Решение.** Произведем проверки  $S_i$  согласно (7.23) и  $S_\Sigma$  согласно (7.26), 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 7.4.

### 7.3.3 Циклические коды

*Общие понятия и определения.* Циклические коды относятся к числу блочковых 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:

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

При описании циклических кодов  $r$ -разрядные кодовые комбинации представляются в виде многочленов фиктивной переменной  $x$  (см. пункт 6.2.1). 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  $XG_0(X)$ . Нетрудно убедиться, что кодовая комбинация, получающаяся при сложении этих двух комбинаций, также будет соответствовать результату умножения многочлена  $x^3 + x + 1$  на многочлен  $x + 1$ :

$$001011 \oplus 010110 = 011101 = x^4 + x^3 + x^2 + 1, \quad (x^3 + x + 1)(x + 1) = \\ = x^4 + x^3 + x^2 + 1 = 011101.$$

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, не делится без остатка на образующий многочлен. Это свойство позволяет обнаружить ошибку. По виду остатка можно определить и вектор ошибки.

Умножение и деление многочленов достаточно 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 представлены в приложении В.

*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. Умножают кодовую комбинацию  $G(X)$ , которую необходимо закодировать, на одночлен  $X^r$ , имеющий ту же степень, что и образующий многочлен. Делят произведение  $G(X)X^r$  на образующий полином  $P(X)$ :

$$\frac{G(X)X^r}{P(X)} = Q(X) + \frac{R(X)}{P(X)}, \quad (7.28)$$

где  $Q(X)$  – частное от деления;

$R(X)$  – остаток.

Умножая выражение (7.28) на  $P(X)$  и перенося  $R(X)$  в другую часть равенства, согласно правилам алгебры двоичного поля, т. е. без перемены знака на обратный, получаем

$$F(X) = Q(X) \cdot P(X) = G(X) \cdot X^r + R(X). \quad (7.29)$$

Таким образом, согласно равенству (7.29), 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 7.6.** 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 приложения В многочлен  $P(X) = x^3 + x + 1 = 1011$ . Умножая  $G(X)X^r$ , 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 (7.28) 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

$$\frac{1111000}{1011} = 1101 + \frac{111}{1011}.$$

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 (7.29) 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).

### 7.3.3.1 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  $P(X) = 11$ , 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. пункт 6.2.4). 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  $M$ :

$$\begin{array}{lll} 0000; & a_2 \oplus a_3 = 01100; & a_2 \oplus a_3 \oplus a_4 = 11101; \\ a_1 \oplus a_2 = 00110; & a_2 \oplus a_4 = 10100; & a_1 \oplus a_3 \oplus a_4 = 11011; \\ a_1 \oplus a_3 = 01010; & a_3 \oplus a_4 = 11000; & a_1 \oplus a_2 \oplus a_4 = 10111; \\ a_1 \oplus a_4 = 10010; & a_1 \oplus a_2 \oplus a_3 = 01111; & a_1 \oplus a_2 \oplus a_3 \oplus a_4 = 11110. \end{array}$$

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.



### 7.3.3.2 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  $r$* : производят, как и для кода Хемминга, с исправлением одиночной ошибки, по выражению (7.4) или (7.5);

2) *выбор образующего многочлена  $P(X)$* . Степень образующего многочлена не может быть меньше числа контрольных символов  $r$ . Если в приложении В 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 пункте 6.2.4, 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  $d = 2$ .

При индивидуальном кодировании любой из кодовых комбинаций, принадлежащей к сомножеству  $k$  разрядных комбинаций, поступают по общей методике в соответствии с (7.29).

**Пример 7.7.** 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 (7.5) находим число контрольных символов:

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

Из приложения В 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 7.8.** Закодировать комбинацию  $G(X) = 111011$  циклическим кодом с  $d = 3$ .

**Решение.** Находим число контрольных символов по (7.5):

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

Из приложения В 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 (7.29) получим кодовую комбинацию в циклическом коде с  $d = 3$ :

$$F(X) = 1110111110.$$

### 7.3.3.3 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  $d = 3$ :

$$r_{d=4} = r_{d=3} + 1; \quad (7.30)$$

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

$$P_{d=4}(X) = P_{d=3}(X)(x+1). \quad (7.31)$$

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

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

**Пример 7.9.** Требуется закодировать сообщение  $G(X) = 10101010101010$  циклическим кодом с  $d = 4$ .

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

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

Из уравнения (7.30) следует, что  $r_{d=4} = 5+1 = 6$ .

Выбираем из приложения В образующий полином для  $d = 3$ . Пусть

$$P_{d=3}(X) = x^5 + x^2 + 1.$$

Тогда

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

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 (7.29). 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  $P_{d=4}(X)$ , находим остаток:

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

Следовательно, передаваемая закодированная комбинация будет иметь вид

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

### 7.3.3.4 Циклические коды с $d \geq 5$

Эти коды, разработанные Боузом, Чоудхури и Хоквингемом (коды 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  $n$ , 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, вторым – при заданном  $n$  должно соблюдаться одно из равенств:

$$2^h - 1 = n, \quad (7.32)$$

$$(2^h - 1)/q = n, \quad (7.33)$$

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 (7.32), но и 21 при  $q = 3$  (7.33);

2) *определение кодового расстояния* 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:

$$P(X) = \text{НОК}[M_1(X) M_3(X) \dots M_{2s-1}(X)]. \quad (7.34)$$

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

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

$$t = h; \quad (7.35)$$

5) *выбор минимальных многочленов.* После того как определено число минимальных многочленов  $L$  и степень старшего многочлена  $t$ , многочлены выписывают из приложения Г. 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 длиной  $n$

$$\beta = r \leq t \cdot s; \quad (7.36)$$

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

$$k = n - r. \quad (7.37)$$

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

**Пример 7.10.** Закодировать все комбинации двоичного кода, чтобы  $n = 15$ , а  $s = 2$ .

**Решение.** Определяем кодовое расстояние по (7.2):  $d = 2s + 1 = 2 \cdot 2 + 1 = 5$  (код БЧХ). Число минимальных многочленов  $L = s = 2$ . Старшая степень минимального многочлена по (7.35)  $t = h = 4$ , так как  $15 = 2^4 - 1$ . Выписываем из приложения Г 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 (7.34):

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

Число контрольных символов  $r$  равно по (7.36) степени  $\beta$  образующего многочлена, т. е.  $r = \beta = 8$ , а значит, число информационных символов  $k$  по (7.37) равно:  $k = n - r = 15 - 8 = 7$ . Таким образом, получаем код БЧХ (15,7) с  $s = 2$ .

После нахождения остатков получаем образующую матрицу (7.38)

$$\left\| \begin{array}{cccc|cccccccc} k_1 & k_6 & k_5 & k_4 & k_3 & k_2 & k_1 & r_8 & r_7 & r_6 & r_5 & r_4 & r_3 & r_2 & r_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{array} \right\| \quad (7.38)$$

**Пример 7.11.** 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, приведенной в приложении Г, 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,

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

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 [44], которая приведена в приложении Д в сокращенном виде. В соответствии с изложенной в данном подразделе 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 приложения Д следует, что при  $n=31$ ,  $k=21$  и  $s=2$  образующий многочлен

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

### 7.3.3.5 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 (7.1) и (7.2) 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 (7.34) образующий многочлен дополнительно умножают на двучлен  $x+1$ .

**Пример 7.12.** 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 7.10, и умножаем его на двучлен  $(x+1)$ , т. е. получаем

$$\begin{aligned} P(X) &= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x + 1) = (x^8 + x^7 + x^6 + x^4 + 1)(x + 1) = \\ &= x^9 + x^8 + x^7 + x^5 + x + x^8 + x^7 + x^6 + x^4 + 1 = x^9 + x^6 + x^5 + x^4 + x + 1. \end{aligned}$$

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

### 7.3.3.6 Циклические избыточные коды

Циклический избыточный код (Cyclical Redundancy Check – CRC) имеет фиксированную длину и используется для обнаружения ошибок. Наибольшее распространения получили коды CRC-16 и CRC-32, имеющие длину 16 и 32 бита

соответственно. Код CRC строится по исходному сообщению произвольной длины, т. е. этот код не является блочным в строгом смысле этого слова. Но при каждом конкретном применении этот код – блочный,  $(k + 16, k)$  – код для CRC-16,  $(k + 32, k)$  – код для CRC-32.

Вычисление значения кода CRC происходит посредством деления многочлена, соответствующего исходному сообщению (полином-сообщение), на фиксированный многочлен (полином-генератор). Остаток от такого деления и есть код CRC, соответствующий исходному сообщению. Для кода CRC-16 полином-генератор имеет степень 16, а для CRC-32 – 32. Полиномы-генераторы подбираются специальным образом и для кодов CRC-16/32 стандартизированы Международным консультативным комитетом по телеграфной и телефонной связи (ССИТТ).

Для CRC-16, например, стандартным является полином-генератор

$$x^{16} + x^{12} + x^5 + 1; x^{16} + x^{15} + x^2 + 1,$$

для CRC-32

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1,$$

что соответствует четырехбайтовой контрольной сумме.

Ниже приведен пример построения CRC-4 кода для сообщения 11010111 с использованием полинома-генератора  $x^4 + x^3 + x^2 + 1$ . Исходному сообщению соответствует полином  $x^7 + x^6 + x^4 + x^2 + x + 1$ , т. е. нумерация битов здесь начинается справа:

$$\begin{array}{r} x^7 + x^6 + x^4 + x^2 + x + 1 \\ \underline{x^7 + x^6 + x^5 + x^3} \\ x^5 + x^4 + x^3 + x^2 + x + 1 \\ \underline{x^5 + x^4 + x^3 + x} \\ x^2 + 1 \end{array} \quad \begin{array}{r} | x^4 + x^3 + x^2 + 1 \\ x^3 + x \end{array}$$

Полиному  $x^2 + 1$  соответствуют биты 0101 – это и есть CRC-4 код.

Существуют быстрые алгоритмы для расчета CRC-кодов, использующие специальные таблицы, а не деление многочленов с остатком.

CRC-коды способны обнаруживать одиночную ошибку в любой позиции и, кроме того, многочисленные комбинации кратных ошибок, расположенных близко друг от друга. При реальной передаче или хранении информации ошибки обычно группируются на некотором участке, а не распределяются равномерно по всей длине данных. Таким образом, хотя для идеального случая двоичного симметричного канала CRC-коды не имеют никаких теоретических пре-

имущества по сравнению, например, с простыми контрольными суммами, для реальных систем эти коды являются очень полезными.

Коды CRC используются очень широко: модемами, телекоммуникационными программами, программами архивации и проверки целостности данных и многими другими программными и аппаратными компонентами вычислительных систем.

### 7.3.3.7 Коды Файра

Коды Файра – это циклические коды, обнаруживающие и исправляющие пакеты ошибок. Определение пакета (пачки) ошибки дано в подразделе 7.1. 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 (7.3) не используются.

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

$$P_\Phi(X) = P(X)(x^c + 1), \quad (7.39)$$

где  $P(X)$  – неприводимый многочлен степени

$$t \geq b_s, \quad (7.40)$$

принадлежащий показателю степени

$$E = 2^t - 1, \quad (7.41)$$

$$C \geq b_s + b_m - 1. \quad (7.42)$$

Неприводимый многочлен  $P(X)$  выбирают из приложения В согласно уравнению (7.40).

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,

$$n = \text{НОК}(E, C). \quad (7.43)$$

Число контрольных символов

$$r = t + C. \quad (7.44)$$

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

**Пример 7.13.** Найти образующий полином и определить общую длину кодовых комбинаций  $n$ , а также число контрольных и информационных символов.



лов для кода, позволяющего исправлять пакеты ошибок длиной  $b_s = 4$  и обнаруживать пакеты ошибок длиной  $b_m = 5$ .

**Решение.** 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 (7.40) и (7.42)  $t \geq 4$ ,  $C \geq 8$ . Из приложения В находим неприводимый многочлен четвертой степени:  $P(X) = x^4 + x + 1$ .

Согласно (7.39) образующий многочлен

$$P(X)_\Phi = (X^4 + X + 1)(X^8 + 1) = X^{12} + X^9 + X^8 + X^4 + X + 1.$$

По выражению (7.41) находим  $E = 2^4 - 1 = 15$ . Поэтому длина кода из (2.43)  $n = 15 \cdot 8 = 120$ . Из (7.44) 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  $b_m$ .

### 7.3.3.8 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 7.10. 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 (7.38). В результате образующая матрица укороченного кода (12, 4) принимает вид

$$M = \begin{pmatrix} k_4 & k_3 & k_2 & k_1 & r_8 & r_7 & r_6 & r_5 & r_4 & r_3 & r_2 & r_1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{matrix}. \quad (7.45)$$

В матрице (7.45)  $d_{\min} = 5$ . 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*.

### 7.3.3.9 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

$$F^*(X) = F(X) + E(X), \quad (7.46)$$

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 [45], а именно:

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);

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

**Пример 7.14.** 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 исправить ее.

**Решение.** 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 7.15.** При передаче комбинации, представленной в седьмой строке матрицы (7.38), исказились два символа и комбинация была принята в виде 1 1̇ 1̇ 000011101000 (искаженные символы помечены точками).

**Решение.** 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 11000011101001. 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 00001110100011̇1̇. И вновь делим на  $P(X)$ . На этот раз остаток  $R(X) = 00000011$  имеет вес  $w = 2 = s = 2$ . Складываем  $0001110100011 \oplus 00000011$ , получаем 000011101000100. Произведя три циклических 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  $P^*(X)$  на  $P(X)$ , равен



Для достижения равенства  $R(X) = R^*(X)$  необходимо было дописать четыре нуля. Это означает, что ошибка произошла в пятом разряде, т. е. исправленная кодовая комбинация будет иметь вид

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

### 7.3.3.10 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 производятся 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:

$$h(X) = \frac{X^n + 1}{P^{-1}(X)}, \quad (7.47)$$

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 получаются 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:

$$d_{\min} = \mu + 1. \quad (7.48)$$

Поясним принцип мажоритарного декодирования на конкретных примерах.

**Пример 7.17.** Построить матрицы  $N$  и  $L$  и найти систему проверок для циклического кода  $(7, 3)$ , образованного с помощью полинома  $P(X) = (x^3 + x + 1)(x + 1) = x^4 + x^3 + x^2 + 1$  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:

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

Для построения матрицы  $L$  преобразуем матрицу следующим образом. Сложим вторую, третью и четвертую строки матрицы  $N$ :

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

Аналогично сложим первую, третью и четвертую строки:

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

Составим матрицу  $L$ , используя для ее построения две полученные суммы и четвертую 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 соотношения с разделенными относительно члена  $a_0$  проверками. Добавив к этим соотношениям тривиальную проверку  $a_0 = a_0$ , получим систему разделенных относительно  $a_0$  проверок:

$$\left. \begin{array}{l} a_0 = a_4 \oplus a_5 \\ a_0 = a_2 \oplus a_6 \\ a_0 = a_1 \oplus a_3 \\ a_0 = a_0 \end{array} \right\} \quad (7.49)$$

Систему проверок для  $a_1$  получим из (7.49) в виде

$$\left. \begin{array}{l} a_1 = a_5 \oplus a_6 \\ a_1 = a_3 \oplus a_0 \\ a_1 = a_2 \oplus a_4 \\ a_1 = a_1 \end{array} \right\} \quad (7.50)$$

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

**Пример 7.18.** Исходная комбинация  $G(X) = 101$ , закодированная генераторным полиномом  $P(X) = x^4 + x^3 + x^2 + 1$ , поступила в канал связи в виде  $G(X) = 101001$  в результате действия помех была искажена (одиночная ошибка) и в приемник поступила в виде  $F^*(X) = 101001\dot{0}$ . Воспользовавшись системой проверок примера 7.17, 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 (7.49):

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

Большинство проверок указывает, 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 7.19.** 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 2 первую, пятую, седьмую и восьмую; вторую, третью, шестую, седьмую и восьмую; четвертую, шестую, седьмую и восьмую строки матрицы и в результате получим соответствующие кодовые комбинации:

$$\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1, \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1, \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1. \end{array}$$

Составим матрицу  $L$ , используя для ее построения три полученные суммы и восьмую 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  $L$ .

### 7.3.3.11 Код Голея

Данный код является циклическим, образованным генераторным полиномом  $P(X) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$  (110001110101). Кодирование производится по общим правилам для циклических кодов (см. пункт 2.3.3). Различают две разновидности кода Голея:  $(n, k) = (23, 12)$  с кодовым расстоянием  $d = 7$ , позволяющим исправлять все ошибки кратностью  $s = 3$ , и  $(24, 12)$  с кодовым расстоянием  $d = 8$ , позволяющим исправлять ошибки кратностью  $s = 3$  и обнаруживать все искажения кратностью  $m = 4$ . Код  $(24, 12)$  образовывается из кода  $(23, 12)$  путем добавления одного разряда проверки на четность. Данный код относится к числу наиболее распространенных, несмотря на то, что требует более широкой полосы пропускания канала связи.

На рисунке 7.3 представлена структура кодового слова в коде (23,12).



Рисунок 7.3 – Структура кодового слова в коде Голея (23,12)

Декодирование может производиться двумя методами. Первый метод основан на прямом декодировании по таблице соответствий. Этот метод очень ресурсоемкий, а поэтому рассмотрим наиболее чаще применяемый «метод ловли ошибок». Декодирование по данному методу осуществляется в четыре основных этапа.

*Первый этап.* Принятую кодовую комбинацию в коде (23,12)  $F^*(X)$  делим на образующий полином  $P(X)$ . Если вес синдрома равен нулю, то в кодовом слове искаженных символов нет. Если вес синдрома 1, 2 или 3, то это значит, что все (1, 2 или 3) искажения находятся в младших 11 байтах кодовой комбинации  $F^*(X)$ . Причем единичные биты в синдроме соответствуют искаженным битам в принятой кодовой комбинации. В случае, если вес больше трех, переходим ко второму этапу.

*Второй этап.* Предполагаем, что искажен 17-й бит, остальные (0, 1 или 2) искажения находятся в младших 11 битах. Для проверки предположения находим модифицированный синдром. Для этого складываем по модулю 2 синдром, полученный на первом этапе с остатком от деления  $x^{16}$  на  $P(X)$ , который заранее подсчитан и равен 01101100110 (старшие биты слева). Если вес модифицированного синдрома равен 0, 1 или 2, то это означает, что искажен 17-й бит, а все остальные ошибки в младших 11 битах. Причем единичные биты в модифицированном синдроме соответствуют искаженным битам в принятой кодовой комбинации. В случае, если вес больше трех, переходим к третьему этапу.

*Третий этап.* Предполагаем, что искажен 18-й бит, а остальные (0, 1 или 2) искажения находятся в младших 11 битах. Далее выполняем действия аналогично второму этапу. При этом остаток от деления  $x^{17}$  на  $P(X)$  подсчитан заранее и равен 11011001100 (старшие биты слева). Если вес больше трех, то переходим к четвертому этапу.

*Четвертый этап.* Производим циклический сдвиг принятой кодовой комбинации  $F^*(X)$  в сторону младших разрядов и начинаем с первого этапа. Если на очередном этапе искажения будут определены, необходимо исправленную кодовую комбинацию циклически сдвинуть в сторону старших разрядов в соответствующее количество разрядов, равное количеству разрядов сдвига в сторону младших разрядов.

*Примечание* – Чтобы каждый раз на первом этапе не делить кодовую комбинацию на  $P(X)$ , существует правило для нахождения нового синдрома, которое

заключается в следующем: если в младшем разряде синдрома нуль, то новый синдром получается циклическим сдвигом на один разряд вправо старого синдрома, а если в младшем разряде единица, то для нахождения нового синдрома нужно старый синдром сложить с  $P(X)$ , а потом отбросить младший (справа) бит.

**Пример 7.20.** На приемную сторону поступила кодовая комбинация  $F^*(X) = 11010010100111100001101$ , закодированная кодом Голя. Декодировать данную комбинацию.

**Решение.** Производим деление  $F^*(X)$  на генераторный полином  $P(X) = 110001110101$ . В результате деления получим остаток  $R(X) = 00000000000$ , что в соответствии с первым этапом указывает на то, что в принятой кодовой комбинации искажений нет. Отбрасываем 11 младших разрядов (контрольные разряды) и получаем исходную кодовую комбинацию  $G(X) = 110100101001$ , которая кодировалась на передающей стороне.

**Пример 7.21.** Из канала связи поступила кодовая комбинация  $F^*(X) = 11010010100111101011111$ , закодированная кодом Голя (23,12). Декодировать данную комбинацию.

**Решение.** Делим  $F^*(X)$  на  $P(X)$  и в результате чего получаем остаток (синдром)  $R(X) = 00001010010$ , вес которого  $w = 3$ , что говорит об искажениях в младших битах (см. первый этап). Для получения исправленной кодовой комбинации производим сложение по модулю 2 принятой кодовой комбинации  $F^*(X)$  с синдромом  $R(X)$ , получаем:

$$\begin{array}{r} 11010010100111101011111 \quad - \text{ кодовая комбинация } F^*(X), \\ \oplus \quad \underline{\quad\quad\quad 00001010010 \quad} \quad - \text{ синдром } R(X), \\ 11010010100111100001101 \quad - \text{ исправленная кодовая комбинация } F(X). \end{array}$$

Для проверки правильности принятого решения разделим исправленную кодовую комбинацию  $F(X)$  на  $P(X)$ . В результате получим только остаток  $R(X) = 00000000000$ , что говорит об отсутствии искажений (правильности принятого решения при декодировании).

**Пример 7.22.** В приемник поступила кодовая комбинация  $F^*(X) = 11010110100111100000100$ , закодированная кодом Голя (23,12). Декодировать данную комбинацию.

**Решение.** Делим  $F^*(X)$  на  $P(X)$  и в итоге получаем остаток  $R(X) = 110011000101$  с весом  $w = 6$ . Далее поступаем в соответствии с третьим этапом. Предполагаем, что искажен 18-й бит, а остальные (0, 1 или 2) искажения находятся в младших 11 битах. Находим модифицированный синдром путем сложения  $R(X)$  с остатком от деления  $x^{17}$  на  $P(X)$ , равным 11011001100. В итоге получаем:

$$\oplus \begin{array}{r} 11011000101 \\ \underline{11011001100} \\ 00000001001 \end{array} \begin{array}{l} - \text{остаток от деления } F^*(X) \text{ на } P(X), w = 6, \\ - \text{остаток от деления } x^{17} \text{ на } P(X), w = 6, \\ - \text{модифицированный синдром, } w = 2. \end{array}$$

Вес модифицированного синдрома  $w = 2$  означает что искажен 18-й бит, а остальные искажения находятся в первом и четвертом битах. Для получения исправленной комбинации производим сложение принятой кодовой комбинации  $F^*(X)$  с вектором ошибки в 18-м бите и модифицированным синдромом. В результате получим:

$$\begin{array}{r} 11010110100111100000100 \\ \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline | | | | | | | | | | | | | | | | | | \\ \hline \end{array} 1000000000000000000 \\ \underline{\hspace{10em} 00000001001} \\ 11010010100111100001101 \end{array} \begin{array}{l} - \text{кодовая комбинация } F^*(X), \\ - \text{вектор ошибки в 18-м бите,} \\ - \text{модифицированный полином,} \\ - \text{исправленная кодовая комбинация } F(X). \end{array}$$

Отбрасываем 11 младших (контрольных) разрядов и получаем исходную кодовую комбинацию  $G(X) = 110100101001$ , которая соответствует комбинации  $G(X)$  примера 2.20 при отсутствии искажений.

**Пример 7.23.** На входе приемника зарегистрирована кодовая комбинация  $F^*(x) = 10010011100111110001101$ , закодированная кодом Голея (23,12). Декодировать данную комбинацию.

**Решение.** Делим  $F^*(x)$  на образующий полином  $P(X) = 110001110101$  и в результате чего получаем синдром  $R(X) = 01000101110$ . Учитывая, что вес синдрома  $w = 5$ , переходим к декодированию согласно второму, третьему и четвертому этапам. При этом всю последовательность операций декодирования разобьем на отдельные шаги.

Шаг 1

$$\oplus \begin{array}{r} 01000101110 \\ \underline{01101100110} \\ = 00101001000 \end{array} \begin{array}{l} - \text{синдром,} \\ - \text{остаток от деления } x^{16} \text{ на } P(X), \\ - \text{синдром 16,} \end{array}$$

$$\oplus \begin{array}{r} 01000101110 \\ \underline{11011001100} \\ = 10011100010 \end{array} \begin{array}{l} - \text{синдром шага 1, равный } R(X), \\ - \text{остаток от деления } x^{17} \text{ на } P(X), \\ - \text{синдром 17.} \end{array}$$

Итого:

синдром	вес	
001000101110	5	- синдром
000101001000	5	- синдром 16
010011100010	5	- синдром 17

Вывод: искажения не определены, переходим ко второму шагу.

### Шаг 2

10010011100111110001101 – предыдущее кодовое слово, равное  $F^*(X)$ ,  
11001001110011111000110 – кодовое слово шага 1, полученное циклическим сдвигом вправо

01000101110 – синдром шага 1,  
00100010111 – синдром шага 2, полученный циклическим сдвигом синдрома шага 1,

$$\begin{aligned} & \oplus \begin{array}{l} 00100010111 \\ \underline{01101100110} \end{array} \quad \begin{array}{l} \text{– синдром шага 2,} \\ \text{– остаток от деления } x^{16} \text{ на } P(X), \end{array} \\ = & 01001110001 \quad \text{– синдром 16.} \end{aligned}$$

$$\begin{aligned} & \oplus \begin{array}{l} 00100010111 \\ \underline{11011001100} \end{array} \quad \begin{array}{l} \text{– синдром шага 2,} \\ \text{– остаток от деления } x^{17} \text{ на } P(X), \end{array} \\ = & 11111011011 \quad \text{– синдром 17.} \end{aligned}$$

Итого:

синдром	вес
000100010111	5 – синдром
001001110001	5 – синдром 16
01111011011	9 – синдром 17

Вывод: искажения не определены, переходим к третьему шагу.

### Шаг 3

11001001110011111000110 – кодовое слово шага 2  
0110010011100111100011 – кодовое слово шага 3, полученное циклическим сдвигом вправо

00100010111 – синдром шага 2

В синдроме в младшем разряде 1, поэтому новый вычисляем следующим образом:

$$\begin{aligned} & \oplus \begin{array}{l} 00100010111 \\ \underline{110001110101} \end{array} \quad \begin{array}{l} \text{– синдром шага 2} \\ \text{– образующий полином } P(X) \end{array} \\ = & 110101100010 \end{aligned}$$

отбрасываем младший разряд и получаем  
11010110001 – синдром шага 3

$$\begin{aligned} & \oplus \begin{array}{l} 11010110001 \\ \underline{01101100110} \end{array} \quad \begin{array}{l} \text{– синдром шага 3} \\ \text{– остаток от деления } x^{16} \text{ на } P(X), \end{array} \\ = & 10111010111 \quad \text{– синдром 16.} \end{aligned}$$

$$\begin{aligned} & \oplus \begin{array}{l} 11010110001 \\ \underline{11011001100} \end{array} \quad \begin{array}{l} \text{– синдром шага 3} \\ \text{– остаток от деления } x^{17} \text{ на } P(X), \end{array} \\ = & 00001111101 \quad \text{– синдром 17.} \end{aligned}$$

Итого:

синдром	вес
011010110001	6 – синдром шага 3
010111010111	8 – синдром 16
000001111101	6 – синдром 17

Вывод: искажения не определены, переходим к четвертому шагу.

#### Шаг 4

01100100111001111100011 – кодовое слово шага 3

10110010011100111110001 – кодовое слово шага 4, полученное циклическим сдвигом вправо

11010110001 – синдром шага 3

В синдроме в младшем разряде 1, поэтому новый вычисляем следующим образом:

$$\oplus \begin{array}{r} 11010110001 \text{ – синдром шага 3} \\ 110001110101 \text{ – образующий полином } P(X) \\ \hline \end{array}$$

$$= 101011000100$$

отбрасываем младший разряд и получаем

10101100010 – синдром шага 4

$$\oplus \begin{array}{r} 10101100010 \text{ – синдром шага 4} \\ 01101100110 \text{ – остаток от деления } x^{16} \text{ на } P(X), \\ \hline \end{array}$$

$$= 11000000100 \text{ – синдром 16.}$$

$$\oplus \begin{array}{r} 10101100010 \text{ – синдром шага 4} \\ 11011001100 \text{ – остаток от деления } x^{17} \text{ на } P(X), \\ \hline \end{array}$$

$$= 01110101110 \text{ – синдром 17.}$$

Итого:

синдром	вес	
010101100010	5	– синдром шага 4
011000000100	3	– синдром 16
001110101110	7	– синдром 17

Вывод: искажения не определены, переходим к пятому шагу.

#### Шаг 5

10110010011100111110001 – кодовое слово шага 4

11011001001110011111000 – кодовое слово шага 5, полученное циклическим сдвигом вправо

10101100010 – синдром шага 4

01010110001 – синдром шага 5, получен циклическим сдвигом вправо

$$\oplus \begin{array}{r} 01010110001 \text{ – синдром шага 5} \\ 01101100110 \text{ – остаток от деления } x^{16} \text{ на } P(X), \\ \hline \end{array}$$

$$= 00111010111 \text{ – синдром 16.}$$

$$\oplus \begin{array}{r} 01010110001 \text{ – синдром шага 5} \\ 11011001100 \text{ – остаток от деления } x^{17} \text{ на } P(X), \\ \hline \end{array}$$

$$= 10001111101 \text{ – синдром 17.}$$

Итого:

синдром	вес	
001010110001	5	– синдром шага 5
000111010111	7	– синдром 16
010001111101	7	– синдром 17

Вывод: искажения не определены, переходим к шестому шагу.

Шаг 6

11011001001110011111000 – кодовое слово шага 5

01101100100111001111100 – кодовое слово шага 6, полученное циклическим сдвигом вправо

01010110001 – синдром шага 5

В синдроме в младшем разряде 1, поэтому новый вычисляем следующим образом:

$$\oplus \begin{array}{r} 01010110001 \text{ – синдром шага 5} \\ 110001110101 \text{ – образующий полином } P(X) \\ \hline \end{array}$$

$$= 111011000100$$

отбрасываем младший разряд и получаем

$$11101100010 \text{ – синдром шага 6}$$

$$\oplus \begin{array}{r} 11101100010 \text{ – синдром шага 6} \\ 01101100110 \text{ – остаток от деления } x^{16} \text{ на } P(X), \\ \hline \end{array}$$

$$= 10001111101 \text{ – синдром 16}$$

$$\oplus \begin{array}{r} 11101100010 \text{ – синдром шага 6} \\ 11011001100 \text{ – остаток от деления } x^{17} \text{ на } P(X), \\ \hline \end{array}$$

$$= 00110101110 \text{ – синдром 17.}$$

Итого:

синдром            вес

011101100010    6 – синдром шага 6

010000000100    2 – синдром 16

000110101110    6 – синдром 17

Вывод: вес синдрома 16 равен 2, следовательно, в 17-м разряде ( $x^{16}$ ) ошибка, а остальные ошибки искажения – в младших 11 разрядах.

Исправляем ошибки:

$$\oplus \begin{array}{r} 01101100100111001111100 \text{ – кодовое слово шага 6} \\ 10000000100 \text{ – синдром 16} \\ \hline 000000100000000000000000 \text{ – искажение в 17-м разряде} \\ \hline =01101110100101001111000 \text{ – кодовое слово без искажений шага 6} \end{array}$$

Для получения исходного кодового слова выполняем пять циклических сдвигов влево и получаем:

11010010100111100001101 – кодовое слово без ошибок  $F(X)$ .

Получателю выводятся информационные символы, расположенные на позициях 12–23 (см. рисунок 7.3), т. е.  $G(X) = 110100101001$ .

Для проверки правильности декодирования необходимо кодовое слово  $F(X)$ , полученное на шестом шаге, разделить на генераторный полином  $P(X)$ , и если остаток будет нулевого порядка, то декодирование произведено верно.

### 7.3.4 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 (рисунок 7.4).

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 итеративный код (рисунок 7.5) 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.

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

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	0	0	0	0	1
0	0	1	1	0	0
1	0	0	1	0	0

Рисунок 7.4 – Расположение символов итеративного кода

Рисунок 7.5 – Итеративный код

Итеративный 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 кода, в зависимости от которых итеративный  $n_i, k_i, d_i$ -й 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:

$$n = \prod_{i=1}^S n_i, \quad k = \prod_{i=1}^S k_i, \quad d_{\min} = \prod_{i=1}^S d_i, \quad (7.51)$$

где  $n_i, k_i, d_i$  – параметры итерируемых кодов;

$S$  – кратность итерирования.

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 крат-



ностью до трех. Не обнаруживаются четырехкратные ошибки, располагающиеся в вершинах правильного четырехугольника, а также некоторые шестикратные, восьмикратные и т. д. (рисунок 7.6).

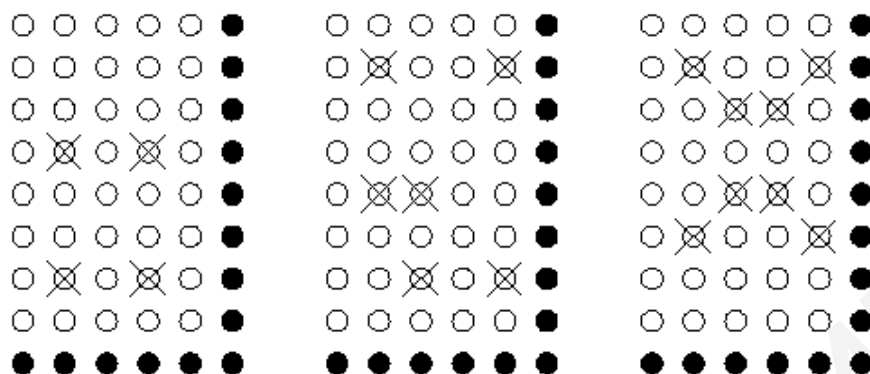


Рисунок 7.6 – Ошибки, не обнаруживаемые простейшим итеративным кодом

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.

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

На рисунке 7.7 показан пример применения третьей проверки по диагонали. Порядок формирования контрольных символов  $P_i$  показан сплошными линиями.

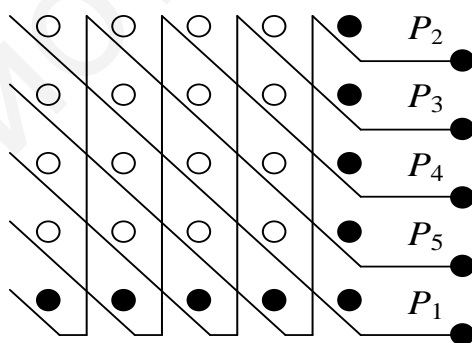


Рисунок 7.7 – Итеративный код с тремя проверками

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.

### 7.3.5 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 2 двух информационных элементов, отстоящих друг от друга на шаг сложения, равный  $b$ .

Рассмотрим процесс кодирования на примере кодовой комбинации, приведенной на рисунке 7.8 (верхняя строка), если шаг сложения  $b = 2$ . Процесс образования контрольных символов показан на этом же рисунке (нижняя строка).

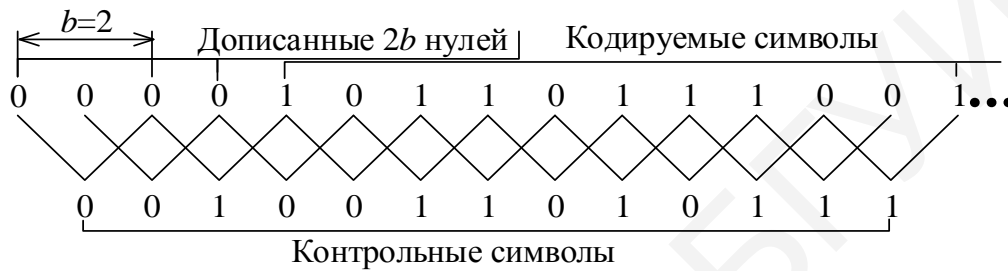


Рисунок 7.8 – Схема построения рекуррентного кода

Структурная схема кодирующего устройства приведена на рисунке 7.9, из которого следует, что входная кодовая комбинация задерживается на  $2b$  тактов, что равносильно дописыванию  $2b$  нулей перед кодируемой кодовой комбинацией (верхняя строка рисунка 7.8).

На выходе кодирующего устройства (см. рисунок 7.9) получим последовательность символов

$$1000111000111110010011, \quad (7.52)$$

которая поступает в дискретный канал связи.

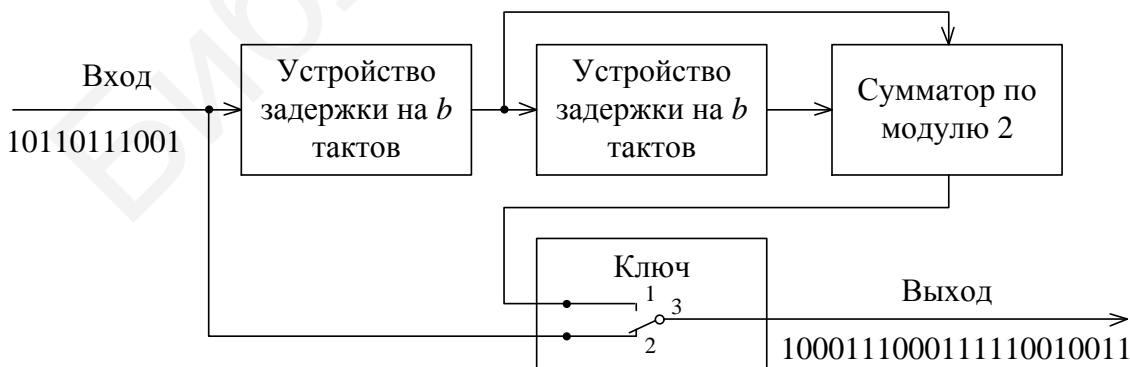


Рисунок 7.9 – Структурная схема кодера рекуррентного кода

Структурная схема декодера приведена на рисунке 7.10.

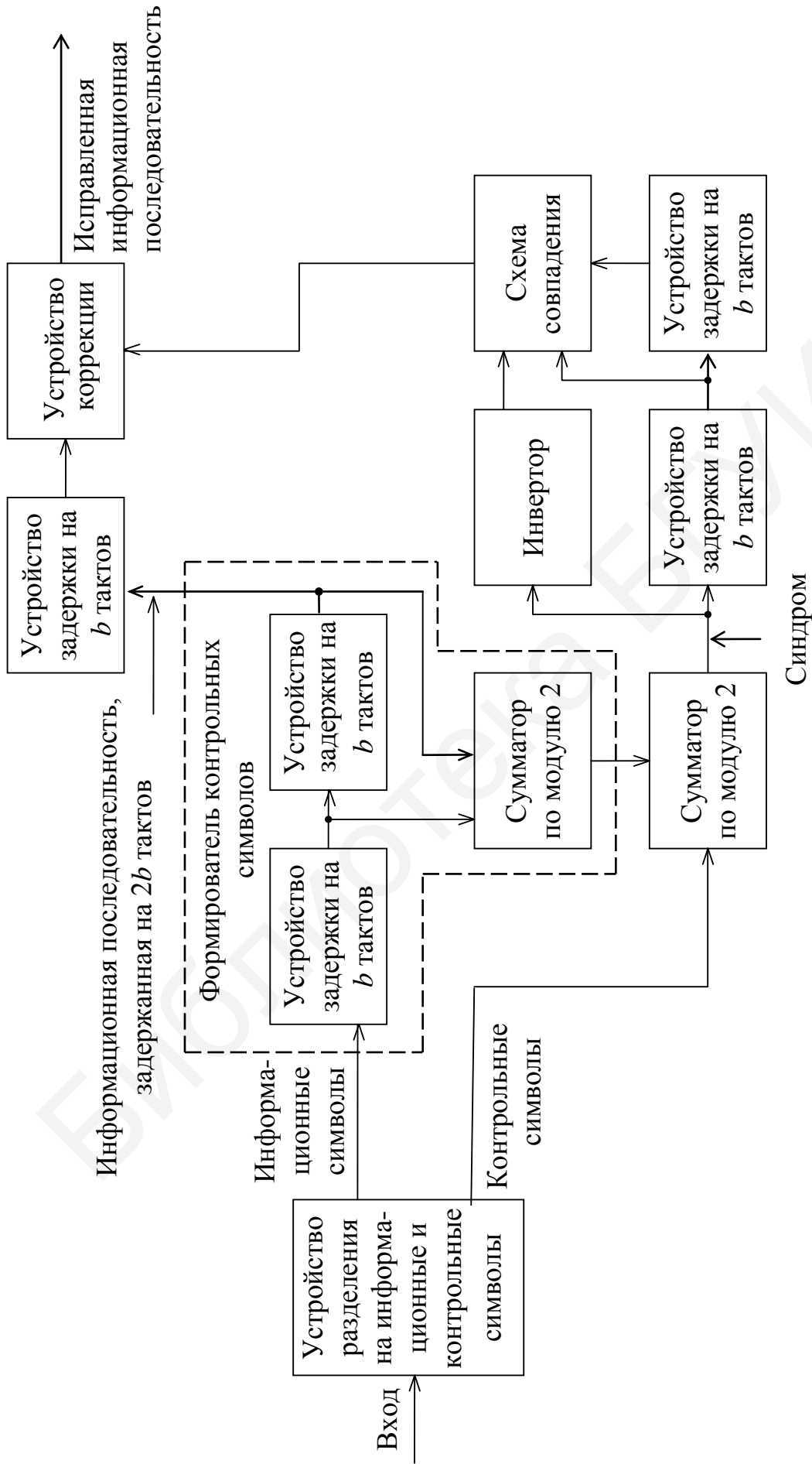


Рисунок 7.10 – Структурная схема декодера рекуррентного кода

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)

$$1000\bar{1}0\bar{1}1\bar{1}0111110010011. \quad (7.53)$$

Устройство разделения (см. рисунок 7.10) разделяет последовательность (7.53) на информационные

$$101\bar{1}1\bar{1}11001 \quad (7.54)$$

и контрольные символы

$$000\bar{1}0110101. \quad (7.55)$$

Последовательности символов (7.54) и (7.55) содержат ошибочные символы, которые подчеркнуты сверху. Формирователь контрольных символов из (7.54) выдает последовательность символов

$$00100100001, \quad (7.56)$$

которая в сумме по модулю 2 с последовательностью (7.55) дает исправляющую последовательность

$$\oplus \begin{array}{r} 00010110101 \\ 00100100001 \\ \hline 00110010100 \end{array} \quad (7.57)$$

Исправляющая последовательность (7.57) подается на инвертор, который выдает последовательность (7.58) и одновременно поступает на устройства задержки на  $b$  и  $2b$  тактов (задержки отмечены соответствующим числом точек слева). На выходе устройств задержки появляются последовательности (7.59) и (7.60) соответственно. На выходе схемы совпадения получаем последовательность (7.61):

$$11001101011... \quad (7.58)$$

$$..001100101... \quad (7.59)$$

$$....0011001... \quad (7.60)$$

$$\hline 00000000001... \quad (7.61)$$

Точки слева в последовательности 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.

Исправленная последовательность вырабатывается устройством коррекции в виде суммы по модулю 2 последовательностей (7.61) и (7.54) (информационные символы), задержанной на  $3b$  тактов:

$$\begin{array}{r} 00000000001000000 \\ \dots\dots\dots .10111111001 \\ \hline 10110111001 \end{array} \quad (7.62)$$

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

После автоматического исправления последовательность (7.62) совпадает с последовательностью, представленной на рисунке 7.8 (верхняя строка). Как следует из (7.62), 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 для передачи факсимильных данных.

### 7.3.6 Сверточные коды

Методы кодирования и декодирования, рассмотренные в пунктах 7.3.1–7.3.4, относились к блочным кодам. При использовании таких кодов информационная последовательность разбивается на отдельные блоки, которые кодируются независимо друг от друга. Таким образом, закодированная последовательность становится последовательностью независимых слов одинаковой длины.

При использовании сверточных кодов поток данных разбивается на гораздо меньшие блоки длиной  $k$  символов (в частном случае  $k_0 = 1$ ), которые называются *кадрами информационных символов*.

Кадры информационных символов кодируются *кадрами кодовых символов* длиной  $n_0$ . При этом кодирование кадра информационных символов в кадр кодового слова производится с учетом предшествующих  $t$  кадров информационных символов. Процедура кодирования, таким образом, связывает между собой последовательные кадры кодовых слов. Передаваемая последовательность становится одним полубесконечным кодовым словом.

Основными характеристиками сверточных кодов являются величины:

- $k_0$  – размер кадра информационных символов;
- $n_0$  – размер кадра кодовых символов;
- $t$  – длина памяти кода;
- $k = (t+1) \cdot k_0$  – информационная длина слова;
- $n = (t+1) \cdot n_0$  – кодовая длина блока.

Кодовая длина блока – это длина кодовой последовательности, на которой сохраняется влияние одного кадра информационных символов.

Наконец, сверточный код имеет еще один важный параметр – скорость  $R = k/n$ , которая характеризует степень избыточности кода, вводимой для обеспечения исправляющих свойств кода.

Как и блочные, сверточные коды могут быть систематическими и несистематическими и обозначаются как линейные сверточные  $(n, k)$ -коды.

Систематическим сверточным кодом является такой код, для которого в выходной последовательности кодовых символов содержится без изменения породившая его последовательность информационных символов. В противном случае сверточный код является несистематическим.

Примеры схем кодеров для систематического  $(8, 4)$  и несистематического сверточных  $(6, 3)$  кодов приведены на рисунках 7.11 и 7.12 соответственно.

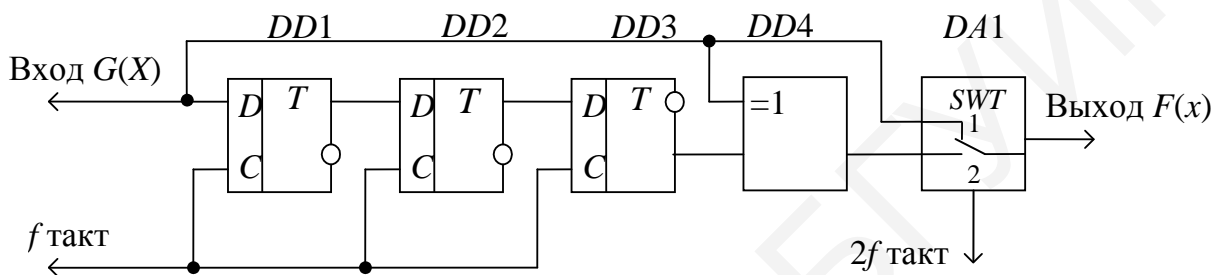


Рисунок 7.11 – Кодер систематического сверточного кода  $(8, 4)$

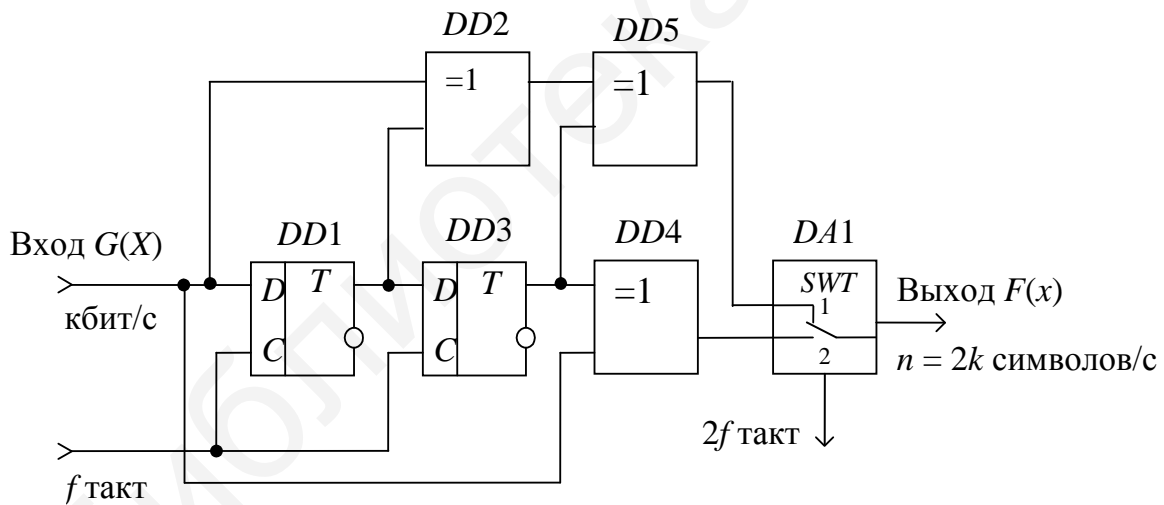


Рисунок 7.12 – Кодер несистематического сверточного кода  $(6, 3)$

Для того чтобы схема на рисунке 7.12 стала систематической, надо убрать один сумматор  $DD4$ . Корректирующие свойства от этого не изменятся, хотя у несистематического кода свертка больше – это выгодно, так как нет информации в открытом виде.

Возможны различные способы описания сверточных кодов, например, с помощью порождающей матрицы. В силу бесконечности кодируемой последовательности порождающая матрица будет иметь бесконечные размеры: она будет состоять из бесконечного числа матриц  $M$  для обычного блочного кода, расположенных

вдоль главной диагонали полубесконечной матрицы. Вся остальная ее часть заполняется нулями.

Однако более удобным способом описания сверточного кода является его задание с помощью *импульсной переходной характеристики* эквивалентного фильтра или соответствующего ей *порождающего полинома*.

Импульсная переходная характеристика (ИПХ) фильтра (кодер сверточного кода является фильтром) есть реакция на единичное воздействие вида  $\bar{\delta} = 10\ 000\dots$ ). Для кодеров, изображенных на рисунках 7.11 и 7.12, соответствующие импульсные характеристики будут иметь вид

$$H_1 = 11.00.00.01.00.00, \quad (7.63)$$

$$H_2 = 11.10.11.00.00.00. \quad (7.64)$$

Еще одна форма задания сверточных кодов – это использование порождающих полиномов, однозначно связанных с ИПХ эквивалентного фильтра:

$$H_1(x) = 1 + x + x^7, \quad (7.65)$$

$$H_2(x) = 1 + x + x^2 + x^4 + x^5. \quad (7.66)$$

При этом кодовая последовательность  $F(x)$  на выходе сверточного кодера получается в результате свертки входной информационной последовательности  $G(x)$  с импульсной переходной характеристикой  $H$ .

Рассмотрим примеры кодирования последовательностей с использованием импульсной характеристики эквивалентного фильтра.

Пусть  $G(x) = (110\dots)$ , тогда для кодера с ИПХ  $F_1 = 11.00.00.01.00.00\dots$

$$\begin{array}{r}
 11.00.00.01.00.00 \dots \\
 11.00.00.01.00.00 \dots \\
 \hline
 F_1 = (11.11.00.01.01.00.00 \dots)
 \end{array}$$

Или для  $G(x) = (1011000)$

$$\begin{array}{r}
 11.00.00.01.00.00 \dots \\
 11.00.00.01.00.00 \dots \\
 11.00.00.01.00.00 \dots \\
 \hline
 F(x) = 11.00.11.10.00.01.10.00.00 \dots
 \end{array}$$

Иногда удобнее рассматривать полный порождающий полином сверточного кода  $P(x)$  как совокупность нескольких многочленов меньших степеней, соответствующих ячейкам выходного регистра кодера. Так, для кодера, представленного на рисунке 7.11, соответствующие частичные порождающие полиномы будут следующими:

$$P_1(X) = 1 + x + x^2, \quad (7.67)$$

$$P_2(X) = 1 + x^2. \quad (7.68)$$

Пусть, например, кодируется последовательность  $G(X) = 1010\dots$

Тогда на входе 1 ключа  $DA1$  при кодировании будет последовательность  $F(X) = (11011000\dots)$ , а на входе 2 – последовательность  $F_2(X) = (10001000\dots)$ .

Легко заметить, что при этом справедливы равенства

$$F_1(X) = G(X)P_1(X), \quad (7.69)$$

$$F_2(X) = G(X)P_2(X). \quad (7.70)$$

Такая форма записи удобна, поскольку видна структура кодирующего устройства (по набору полиномов можно сразу синтезировать устройство).

После того как рассмотрены методы представления кодовых комбинаций в сверточном коде, следует указать, что использование стандартов значительно удешевляет разработку и эксплуатацию сложных технических систем. В настоящее время в телеметрии в качестве международных приняты стандарты консультативного комитета по космическим системам передачи данных – *CCSDS* (Consultative Committee for Space Data Systems).

В *CCSDS* рекомендуется использовать базовый сверточный код со следующими характеристиками:

- скорость кодирования:  $R = 1/2$  битов на символ;
- длина связки:  $k = 7$  битов.

Образующие многочлены:

$$P_1(x) = 1111001 = 1 + x + x^2 + x^3 + x^6,$$

$$P_2(x) = 1011011 = 1 + x^2 + x^3 + x^5 + x^6;$$

- инверсия символов: на выходе  $C2$ .

Функциональная схема сверточного кодера, отвечающая указанным характеристиками, приведена на рисунке 7.13.

Выходные символы базового сверточного кода  $1/2$  – это последовательность  $C_1(1), C_2(1), C_1(2), C_2(2)$ .

Скорость кодирования сверточного кодера ( $r = 1/2$ ), фиксированная длина связки ( $k = 7$ ) могут быть изменены для уменьшения минимально необходимой полосы частот. Это изменение достигается выбиванием битов из выходной последовательности кодера по некоторому правилу  $P(r)$ .



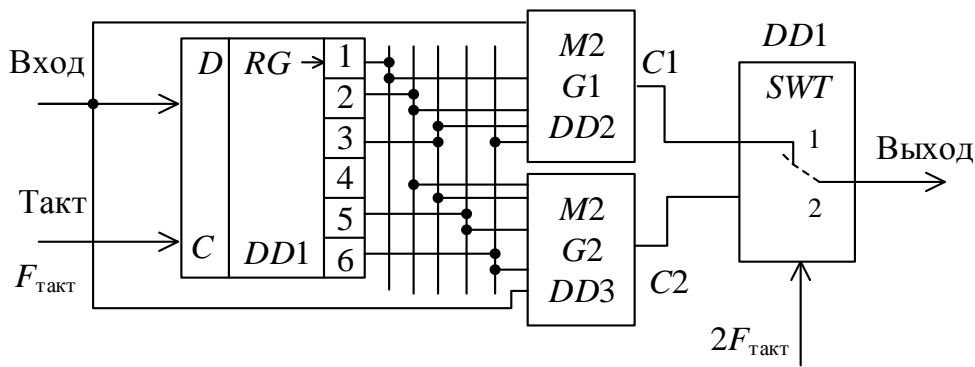


Рисунок 7.13 – Функциональная схема сверточного кодера

**Выбивание** – это удаление некоторых символов, производимых кодером, перед передачей. Выбивание битов уменьшает требуемую для передачи полосу частот, но тем самым ухудшает характеристики канала. Четыре различные схемы выбивания битов предоставляют выбор большинства необходимых скоростей кодирования (2/3, 3/4, 5/6 или 7/8) и соответствующих уровней коррекции ошибок. На рисунке 7.14 изображена схема сверточного кодера с выбиванием битов.

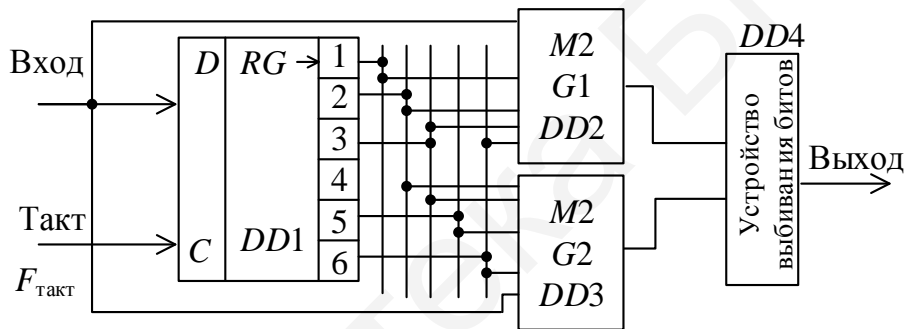


Рисунок 7.14 – Функциональная схема сверточного кодера с выбиванием битов

При применении выбивающих схем инверсия символов  $G2$  пропускается, поэтому, если достаточная для работы системы синхронизации плотность потока битов не обеспечена другими методами, необходимо применение псевдорандомизации.

Структура кода для любой из скоростей кодирования определяется таблицей 7.9.

Таблица 7.9 – Структура кода с выбиванием битов

Правило выбивания: 1 – передаваемый символ, 0 – непередаемый символ	Скорость кодирования	Выходная последовательность
$C_1$ : 10 $C_2$ : 11	2/3	$C_1(1) C_2(1) C_2(2)$
$C_1$ : 101 $C_2$ : 110	3/4	$C_1(1) C_2(1) C_2(2) C_1(3)$
$C_1$ : 10101 $C_2$ : 11010	5/6	$C_1(1) C_2(1) C_2(2) C_1(3) C_2(4) C_1(5)$
$C_1$ : 1000101 $C_2$ : 1111010	7/8	$C_1(1) C_2(1) C_2(2) C_1(3) C_2(4) C_1(5) C_2(6) C_1(7)$

### 7.3.7 Турбокоды

Турбокоды – это систематические блочные коды с кодовыми группами большой длины (сотни-тысячи битов).

Турбокодер предоставляет собой комбинацию двух простых сверточных кодеров, которые генерируют проверочные символы из  $k$ -информационных битов, поступающих на вход. Вместе с проверочными символами в формировании выходной последовательности турбокодера участвуют незакодированные входные информационные биты. Передача информации осуществляется блоками. Длина передаваемых блоков кода

$$n = \frac{k + 4}{R} \quad (7.71)$$

и соответствующая ей скорость передачи приведены в таблице 7.10. Из-за необходимости передачи символов завершения блока кода реальная скорость кодирования несколько меньше номинальной.

Таблица 7.10 – Длины кодовых блоков

Длина информационного блока, $K$	Длина блока $n$ при скорости кодирования $R$			
	1/2	1/3	1/4	1/6
1784	3576	5364	7152	10 728
3568	7144	10 716	14 288	21 434
7136	14 280	21 420	28 560	42 840
8920	17 848	26 772	35 696	53 544
16 384	32 776	49 164	65 552	98 328

Далее под термином скорость кодирования всегда понимается номинальная скорость кодирования.

Основная особенность турбокодера и декодера – это перестановка, т. е. побитовое чередование всего блока данных. В отличие от посимвольного «прямоугольного» чередования в  $RS$ -коде, перестановка в турбокоде чередует отдельные биты и напоминает беспорядочное их поведение.

Перестановка в любом блоке длиной  $k$  задается переупорядочиванием целых чисел  $1, 2, K$ , по следующему алгоритму. Сначала вычисляется  $k$ :  $k = k_1 k_2$ . Параметры  $k_1$  и  $k_2$  приведены в таблице 7.11. Далее для получения номеров перестановок  $\pi(S)$  вычисляются  $s$ : от  $s=1$  до  $s = k$ . В уравнениях (7.73) и (7.74) результат, полученный в квадратных скобках, означает максимальное целое, большее или равное этому числу, а  $p_q$  – одно из следующих простых чисел:

$$p_1 = 31; p_2 = 37; p_3 = 43; p_4 = 47; p_5 = 53; p_6 = 59; p_7 = 61; p_8 = 67.$$

Таблица 7.11 – Параметры кода  $k_1$  и  $k_2$

Длина информационного блока	$k_1$	$k_2$
1784	8	223
3568	8	223·2
7136	8	223·4
8920	8	223·5

$$m = (S - 1) \bmod 2, \quad (7.72)$$

$$i = \lceil (s - 1) / 2k_2 \rceil, \quad (7.73)$$

$$j = \lceil (s - 1) / 2 \rceil - ik_2, \quad (7.74)$$

$$t = (19i + 1) \bmod 2(k_1 / 2), \quad (7.75)$$

$$q = t \bmod 8 + 1, \quad (7.76)$$

$$c = (p_q j + 2m) \bmod k_2, \quad (7.77)$$

$$\pi(s) = 2(t + c(k_1 / 2) + 1) - m. \quad (7.78)$$

Интерпретация полученных чисел такова:  $s$ -й бит входного блока становится  $\pi(s) - m$  выходным битом, как показано на рисунке 7.15.

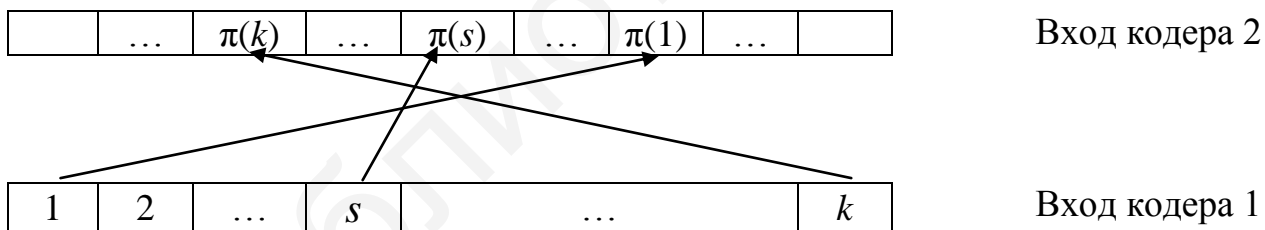


Рисунок 7.15 – Перестановка в турбокоде

Выходные последовательности компонентов турбокодера определяются обратными связями сдвиговых регистров компонентов кодера – «обратными» векторами  $G_0$  и комбинацией их выходов – «прямыми» векторами  $G_1$ ,  $G_2$  и  $G_3$ :

- Обратный вектор для всех скоростей кодирования  $G_0 = 10011$ .
- Прямой вектор для обеих составляющих кодера и скоростей кодирования  $1/2$  и  $1/3$   $G_1 = 11011$ . При скорости кодирования  $1/2$  все символы, кроме символов с выходом  $G_1$  обоих компонентов кодера, выбиваются. Без выбивания скорость кодирования составляет  $1/3$ .

- Прямые векторы для скорости  $1/4$ :  $G_2 = 11011$ ,  $G_3 = 11111$  для первого сверточного кодера и  $G_1 = 11011$  для второго сверточного кодера. Скорость кодирования на этих выходах без выбивания составляет  $1/4$ .

- Прямые вектора для скорости  $1/6$ :  $G_1 = 11011$ ,  $G_2 = 10101$ ,  $G_3 = 11111$  для первого сверточного кодера,  $G_3 = 11111$  для второго сверточного кодера. Скорость кодирования на этом выходе без выбивания составляет  $1/6$ . Функциональная схема турбокодера показана на рисунке 7.16.

Каждый входной блок  $k$ -информационных битов помещается в буфер блока, и биты читаются из буфера в двух разных порядках для двух составляющих кодера. На вход первого кодера поступают биты в исходном порядке, на вход второго составляющего кодера – переставленные биты. Следующий считываемый бит для входа  $A$  определяется простым счетчиком, а для входа  $B$  – некоторым правилом перестановки.

Составляющие кодера – это рекурсивные сверточные кодеры, образованные сдвиговыми регистрами с обратными связями. Ключевое отличие этих кодеров от базового сверточного кодера – рекурсивность, т. е. наличие обратных связей.

Оба компонента кодера, показанные на рисунке 7.16, инициализируются нулями во всех регистрах. Их полный цикл составляет  $k + 4$  битов, за который они производят выходной блок кода из  $(k + 4)/R$  кодовых символов, где  $R$  – номинальная скорость кодирования. Во время передачи  $k$ -битов, входные переключатели находятся в положении «вниз», как показано на рисунке 7.16, для загрузки входных данных.

Во время поступления на выход последних четырех битов информационного блока они находятся в положении «вверх», замыкая входы элемента «исключающее ИЛИ» и тем самым размыкая обратную связь сдвигового регистра. Вследствие этого через четыре такта регистры заполняются нулями.

В течение этих четырех тактов кодер продолжает выводить ненулевые символы кода, в том числе кодируемый выход (выход 01), добавляя таким образом четыре бита  $k$ -информационным битам в блок кода.

Выходы кодеров, как показано в правой части рисунка 7.16, мультиплексируются для выбранной скорости кодирования и формируют блок кода. Для скорости кодирования  $1/3$  выходная последовательность образуется выходами 01, 11, 21, для скорости кодирования  $1/4$  – 01, 12, 13, 21, для скорости кодирования  $1/6$  – 01, 11, 12, 13, 21, 23. Эти последовательности формируются в течение  $k + 4$  тактов.

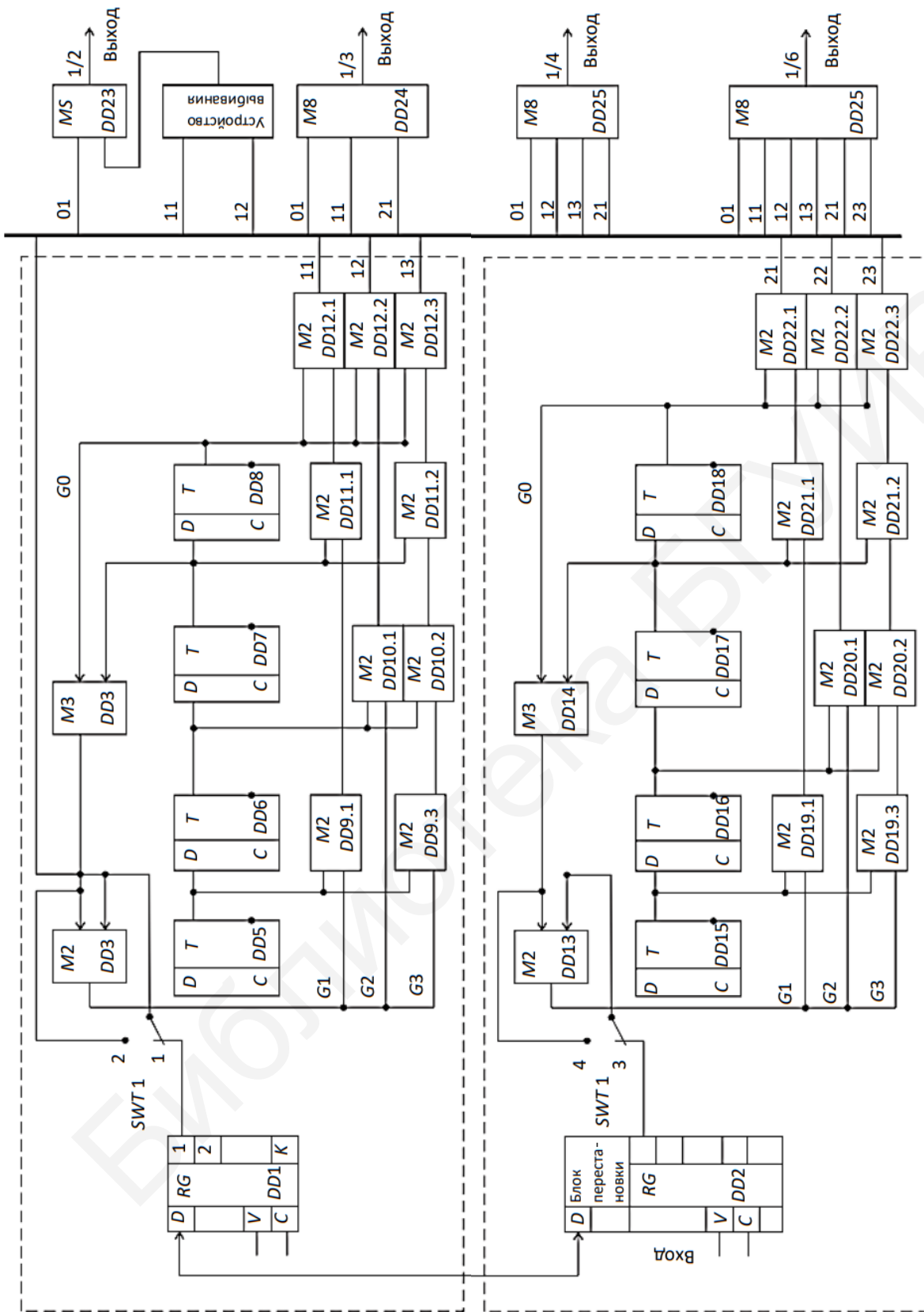


Рисунок 7.16 – Функциональная схема турбокодера

Скорость кодирования  $1/2$  достигается выбиванием каждого второго бита с выходов 11 и 21 по следующему последовательному алгоритму: первым выбивается выход 21, вторым – 11 и т. д. Всего выбивается  $(k + 4)/2$  битов.

Блоки турбокода, сформированные из этих последовательностей для четырех скоростей передачи, изображены на рисунке 7.17.

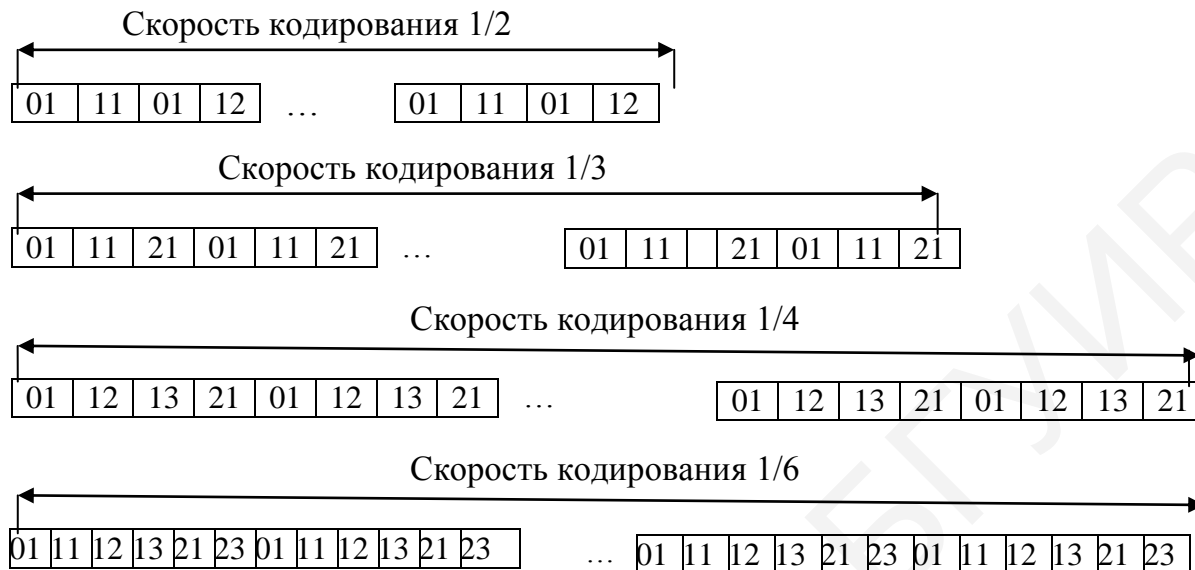


Рисунок 7.17 – Блоки турбокода при различных скоростях кодирования

## 7.4 Коды Рида – Соломона

### 7.4.1 Общие сведения

Коды Рида – Соломона – это недвоичные циклические коды, символы которых представляют собой  $m$ -битовые последовательности, где  $m$  – положительное целое число, большее единицы. Коды Рида – Соломона  $(n, k)$  определены на  $m$ -битовых символах при  $n, k$ , для которых

$$(n, k) = (2^m - 1, 2^m - 1 - 2s), \quad (7.79)$$

где  $s$  – количество искаженных битов в символе, которые могут исправить код, а  $n - k = 2s = r$  – число контрольных символов.

Расширенный код Рида – Соломона можно получить при  $n = 2^m$  или  $n = 2^m + 1$ , но не более того. Минимальное кодовое расстояние определяется следующим образом [1]:

$$d_{\min} = n - k + 1 = 2s + 1 = r + 1. \quad (7.80)$$

Преимущество недвоичных кодов Рида – Соломона можно увидеть в следующем сравнении. Рассмотрим двоичный код  $(n, k) = (7, 3)$ . Общее количество  $n$ -разрядных кодовых комбинаций будет  $2^n = 2^7 = 128$ , из которых  $2^k = 2^3 = 8$  (или  $1/16$  часть от общего) являются информационными. Затем рассмотрим недвоичный код

$(n, k) = (7, 3)$ , где каждый символ состоит из  $m = 8$  битов. Тогда общее количество кодовых комбинаций будет  $2^{mn} = 2^{21} = 2097152$ , из которых  $2^{km} = 2^9 = 512$  (или  $1/4096$  часть от общего) являются информационными. Отсюда можно сделать вывод, что если в качестве информационных комбинаций используется незначительная часть от общего количества  $n$ -разрядных кодовых комбинаций, т. е. можно достичь большего  $d_{\min}$ .

Вероятность появления ошибки в декодированном символе  $P_E$  можно записать через вероятность появления ошибки в канальном символе  $P$ :

$$P_E = \frac{1}{2^m - 1} \sum_{j=t+1}^{2^m-1} C_j^{2^m-1} p^j (1-p)^{2^m-1-j}. \quad (7.81)$$

При описании циклических кодов  $n$ -разрядные кодовые комбинации представляются в виде многочленов фиктивной переменной  $X$ . Показатели степени у  $X$  соответствуют номерам разрядов (начиная с нулевого), а коэффициентами при  $X$  в общем случае являются  $p$  элементов поля  $GF(p)$ . При этом наименьшему разряду числа соответствует фиктивная переменная  $X^0 = 1$ . Для двоичных кодов коэффициентами при  $X$  будут только цифры 0 и 1. Для кодов Рида – Соломона каждый из  $2^m$  элементов конечного поля  $GF(2^m)$  можно представить как отдельный полином степени  $m-1$  или меньше. Степенью полинома называется степень члена максимального порядка. Обозначим каждый ненулевой элемент  $GF(2^m)$  полиномом  $a_i(X)$ , в котором по крайней мере  $m$  коэффициентов  $a_i(X)$  ненулевые. Для  $i = 0, 1, 2, \dots, 2^m - 2$

$$\alpha_i(X) = \alpha_{i,0} + \alpha_{i,1}X + \alpha_{i,2}X^2 + \dots + \alpha_{i,m-1}X^{m-1}. \quad (7.82)$$

Для понимания принципа кодирования и декодирования недвоичных кодов, таких как Рида – Соломона, сделаем экскурс в понятие конечных, известных как поля Галуа ( $GF$ ). Для любого простого числа существует конечное поле, которое обозначается  $GF(p)$  и содержит  $p$  элементов. Понятие  $GF(p)$  можно обобщить как поле из  $p^m$  элементов, именуемое полем расширения  $GF(p)$ . Это поле обозначается  $GF(p^m)$ , где  $m$  – положительное число. Заметим, что  $GF(p^m)$  содержит в качестве подмножества все элементы  $GF(p)$ . Символы из поля расширения  $GF(2^m)$  используются при построении кода Рида – Соломона.

Двоичное поле  $GF(2)$  является подполем расширения  $GF(2^m)$ , точно так же как поле вещественных чисел является подполем комплексных чисел. Кроме чисел 0 и 1, в поле расширения существуют дополнительные однозначные элементы, которые будут представлены новым символом  $\alpha$ . Каждый ненулевой элемент в  $GF(2^m)$  можно представить как степень  $\alpha$ . Бесконечное множество элементов  $F$  образуется из стартового множества  $\{0, 1, \alpha\}$  генерируется дополнительными элементами путем последовательного умножения последней записи на  $\alpha$ .

$$F = \{0, 1, \alpha, \alpha^2, \dots, \alpha^j, \dots\} = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^j, \dots\}. \quad (7.83)$$

Для вычисления из  $F$  *конечного* множества элементов  $GF(2^m)$  необходимо на  $F$  наложить условия: оно может содержать только  $2^m$  элементов и быть замкнутым относительно операции умножения. Условие замыкания множества элементов поля по отношению к операции умножения имеет вид нередуцируемого полинома

$$\alpha^{(2^m-1)} + 1 = 0 \quad (7.84)$$

или, что то же самое,

$$\alpha^{(2^m-1)} = 1 = \alpha^0.$$

С помощью полиномиального ограничения любой элемент со степенью, большей или равной  $2^m - 1$ , можно следующим образом понизить до элемента со степенью, меньшей  $2^m - 1$ :

$$\alpha^{(2^m+n)} = \alpha^{(2^m-1)}\alpha^{(n+1)} = \alpha^{(n+1)}. \quad (7.85)$$

Таким образом, уравнение (7.85) можно использовать для формирования конечной последовательности  $F^*$  из бесконечной последовательности  $F$ .

$$\begin{aligned} F^* &= \{0, 1, \alpha, \alpha^2, \dots, \alpha^{2^m-2}, \alpha^{2^m-1}, \alpha^{2^m}, \dots\} = \\ &= \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^m-2}, \alpha^0, \alpha^1, \alpha^2, \dots\}. \end{aligned} \quad (7.86)$$

Следовательно, из уравнения (7.86) можно видеть, что элементы конечного поля  $GF(2^m)$  даются следующим выражением:

$$GF(2^m) = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^m-1}\}. \quad (7.87)$$

Для описания конечного поля применяются примитивные полиномы. Неприводимый (нередуцируемый) полином  $f(x)$  порядка  $m$  будет примитивным, если наименьшим положительным целым числом  $n$ , для которого  $X^n + 1$  делится на  $f(x)$ , будет  $n = 2^m - 1$ . Заметим, что нередуцируемый полином – это такой полином, который нельзя представить в виде произведения полиномов меньшего порядка.

**Пример 7.24.** Основываясь на предыдущем определении примитивного полинома, указать, какие из следующих нередуцируемых полиномов будут примитивными:

- а)  $1 + X + X^4$ ;
- б)  $1 + X + X^2 + X^3 + X^4$ .



## Решение

а) Мы можем проверить этот полином порядка  $m = 4$ , определив, будет ли он делителем  $X^n + 1 = X^{(2^m - 1)} + 1 = X^{15} + 1$  для значений  $n$  из диапазона  $1 \leq n \leq 15$ . Не трудно убедиться, что  $X^{15} + 1$  делится на  $1 + X + X^4$ , и после повторения вычислений можно проверить, что при любых значениях  $n$  из диапазона  $1 \leq n \leq 15$  полином  $X^n + 1$  не делится на  $1 + X + X^4$ . Следовательно,  $1 + X + X^4$  является примитивным полиномом.

б) Легко проверить, что полином является делителем  $X^{15} + 1$ . Проверив, делится ли  $X^n + 1$  на  $1 + X + X^2 + X^3 + X^4$ , для значений  $n$ , меньших 15, можно также видеть, что указанный полином является делителем  $X^5 + 1$ . Следовательно, несмотря на то, что полином  $1 + X + X^2 + X^3 + X^4$  является неприводимым, он не будет примитивным.

Рассмотрим пример, в котором будут задействованы примитивный полином и конечное поле, которое он определяет. Выберем полином  $f(x) = 1 + X + X^3$  из приложения В, который определяет конечное поле  $GF(2^m)$ , где степень полинома  $m = 3$ . Таким образом, в поле, определяемом полиномом  $f(x)$ , имеется  $2^m = 2^3 = 8$  элементов. Поиск корней полинома  $f(x)$  — это поиск таких значений  $X$ , при которых  $f(x) = 0$ . Привычные нам двоичные элементы 0 и 1 не подходят полиному  $f(x) = 1 + X + X^3$  (они не являются корнями), поскольку  $f(1) = 1$  и  $f(0) = 1$  (в рамках операций по модулю 2). Кроме того, основная теорема алгебры утверждает, что полином порядка  $m$  должен иметь в точности  $m$  корней. Следовательно, в этом примере выражение  $f(0) = 1$  должно иметь три корня. Возникает определенная проблема, поскольку три корня не лежат в том конечном поле, что и коэффициенты  $f(x)$ . А если они находятся где-то еще, то, скорее всего, в поле расширения  $GF(2^3)$ . Пусть  $\alpha$  — элемент поля расширения, который определяется как корень полинома  $f(x)$ . Следовательно, можно записать:

$$f(\alpha) = 0,$$

$$1 + \alpha + \alpha^3 = 0, \quad (7.88)$$

$$\alpha^3 = -1 - \alpha.$$

Поскольку при операциях над двоичным полем  $+1 = -1$ , то  $\alpha^3$  можно представить следующим образом:

$$\alpha^3 = 1 + \alpha. \quad (7.89)$$

Таким образом,  $\alpha^3$  представляется как взвешенная сумма всех  $\alpha$ -членов более низкого порядка. Фактически так можно представить все степени  $\alpha$ . Например, рассмотрим следующее:

$$\alpha^4 = \alpha \cdot \alpha^3 = \alpha \cdot (1 + \alpha) = \alpha + \alpha^2, \quad (7.90)$$

$$\alpha^5 = \alpha \cdot \alpha^4 = \alpha \cdot (\alpha + \alpha^2) = \alpha^2 + \alpha^3. \quad (7.91)$$

Из уравнений (7.9) и (7.91) получаем

$$\alpha^5 = 1 + \alpha + \alpha^2. \quad (7.92)$$

Используя уравнение (7.92), получаем

$$\alpha^6 = \alpha \cdot \alpha^5 = \alpha \cdot (1 + \alpha + \alpha^2) = \alpha + \alpha^2 + \alpha^3 = 1 + \alpha^2. \quad (7.93)$$

А теперь из уравнения (7.93) вычисляем

$$\alpha^7 = \alpha \cdot \alpha^6 = \alpha \cdot (1 + \alpha^2) = \alpha + \alpha^3 = 1 = \alpha^0. \quad (7.94)$$

Заметьте, что  $\alpha^7 = \alpha^0$  и, следовательно, восемью элементами конечного поля  $GF(2^3)$  будут

$$\{0, \alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}. \quad (7.95)$$

Отображение элементов поля в базисные элементы, которое описывается уравнением (7.82), можно проиллюстрировать с помощью схемы линейного регистра сдвига с обратной связью (рисунок 7.18) и табличной записью (таблица 7.12). Схема генерирует (при  $m=3$ )  $2^m - 1$  ненулевых элементов поля и, таким образом, обобщает процедуры, описанные в уравнениях (7.89)–(7.95). Следует отметить, что показанная на рисунке 7.18 обратная связь соответствует коэффициентам полинома  $f(X) = 1 + X + X^3$ . Пусть вначале схема находится в некотором состоянии, например 1 0 0; при выполнении правого сдвига на один такт можно убедиться, что каждый из элементов поля (за исключением нулевого), показанных на рисунке 7.18, циклически будет появляться в разрядах регистра сдвига.

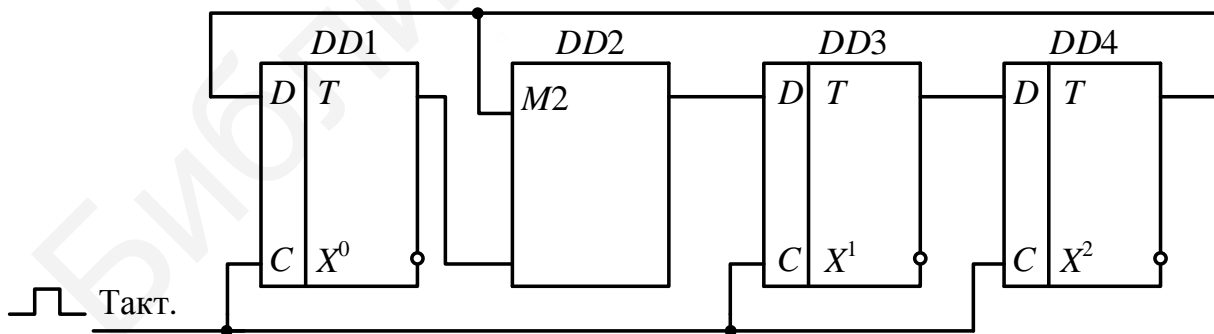


Рисунок 7.18 – Функциональная схема линейного регистра с обратной связью

Таблица 7.12 – Отображение элементов поля в базисные элементы  $GF(8)$  с помощью  $f(X) = 1+X+X^3$

Элемент поля	Образующий элемент		
	$X^0$	$X^1$	$X^2$
0	0	0	0
$a^0$	1	0	0
$a^1$	0	1	0
$a^2$	0	0	1
$a^3$	1	1	0
$a^4$	0	1	1
$a^5$	1	1	1
$a^6$	1	0	1
$a^7$	1	0	0

Каждая строка в таблице 7.12 содержит последовательность двоичных величин, представляющих коэффициенты  $\alpha_{i,0}, \alpha_{i,1}, \alpha_{i,2}$  из уравнения (7.82). Одним из преимуществ использования элементов  $\{\alpha^i\}$  поля расширения вместо двоичных элементов является компактность записи, что оказывается удобным при математическом описании процессов недвоичного кодирования и декодирования.

На данном конечном поле  $GF(2^3)$  можно определить две арифметические операции – сложение и умножение.

В таблице 7.13 показана операция сложения, а в таблице 7.14 – операция умножения, но только для ненулевых элементов. Правила суммирования следуют из уравнений (7.89) и (7.94); их можно доказать, обратившись к таблице 7.12, поскольку сумму двух элементов поля можно рассчитать путем сложения (по модулю 2) соответствующих элементов (7.82).

$$\alpha_i + \alpha_j = (\alpha_{i,0} + \alpha_{j,0}) + (\alpha_{i,1} + \alpha_{j,1})X + \dots + (\alpha_{i,m-1} + \alpha_{j,m-1})X^{m-1}. \quad (7.96)$$

Таблица 7.13 – Таблица сложения для  $GF(8)$  при  $f(x) = 1 + X + X^3$

Элемент поля	$\alpha^0$	$\alpha^1$	$\alpha^2$	$\alpha^3$	$\alpha^4$	$\alpha^5$	$\alpha^6$
$\alpha^0$	0	$\alpha^3$	$\alpha^6$	$\alpha^1$	$\alpha^5$	$\alpha^4$	$\alpha^2$
$\alpha^1$	$\alpha^3$	0	$\alpha^4$	$\alpha^0$	$\alpha^2$	$\alpha^6$	$\alpha^5$
$\alpha^2$	$\alpha^6$	$\alpha^4$	0	$\alpha^5$	$\alpha^1$	$\alpha^3$	$\alpha^0$
$\alpha^3$	$\alpha^1$	$\alpha^0$	$\alpha^5$	0	$\alpha^6$	$\alpha^2$	$\alpha^4$
$\alpha^4$	$\alpha^5$	$\alpha^2$	$\alpha^1$	$\alpha^6$	0	$\alpha^0$	$\alpha^3$
$\alpha^5$	$\alpha^4$	$\alpha^6$	$\alpha^3$	$\alpha^2$	$\alpha^0$	0	$\alpha^1$
$\alpha^6$	$\alpha^2$	$\alpha^5$	$\alpha^0$	$\alpha^4$	$\alpha^3$	$\alpha^1$	0

Правила умножения, указанные в таблице 7.14, следуют из обычной процедуры, в которой произведение элементов поля вычисляется путем сложения по модулю  $(2^m - 1)$  их показателей степеней или для данного случая по модулю 7.

Таблица 7.14 – Таблица умножения для  $GF(8)$  при  $f(x) = 1 + X + X^3$

Элемент поля	$\alpha^0$	$\alpha^1$	$\alpha^2$	$\alpha^3$	$\alpha^4$	$\alpha^5$	$\alpha^6$
$\alpha^0$	$\alpha^0$	$\alpha^1$	$\alpha^2$	$\alpha^3$	$\alpha^4$	$\alpha^5$	$\alpha^6$
$\alpha^1$	$\alpha^1$	$\alpha^2$	$\alpha^3$	$\alpha^4$	$\alpha^5$	$\alpha^6$	$\alpha^0$
$\alpha^2$	$\alpha^2$	$\alpha^3$	$\alpha^4$	$\alpha^5$	$\alpha^6$	$\alpha^0$	$\alpha^1$
$\alpha^3$	$\alpha^3$	$\alpha^4$	$\alpha^5$	$\alpha^6$	$\alpha^0$	$\alpha^1$	$\alpha^2$
$\alpha^4$	$\alpha^4$	$\alpha^5$	$\alpha^6$	$\alpha^0$	$\alpha^1$	$\alpha^2$	$\alpha^3$
$\alpha^5$	$\alpha^5$	$\alpha^6$	$\alpha^0$	$\alpha^1$	$\alpha^2$	$\alpha^3$	$\alpha^4$
$\alpha^6$	$\alpha^6$	$\alpha^0$	$\alpha^1$	$\alpha^2$	$\alpha^3$	$\alpha^4$	$\alpha^5$

### 7.4.2 Кодирование сообщений в коде Рида – Соломона

Наибольшее распространение нашли коды, которые описываются выражением (7.79). Генерирующий полином имеет следующий вид:

$$P(X) = P_0 + P_1X + P_2X^2 + \dots + P_{2S-1}X^{2S-1} + X^{2S}. \quad (7.97)$$

Поскольку  $P(X)$  имеет порядок  $2S$ , мы должны иметь в точности  $2S$  последовательные степени  $\alpha$ , которые являются корнями полинома. Обозначим корни  $P(X)$  как  $\alpha, \alpha^2, \dots, \alpha^{2S}$ . Выразим в качестве примера полиномиальный генератор для кода  $(7, 3)$  через  $2S = n - k = 4$  корня следующим образом:

$$\begin{aligned} P(X) &= (X - \alpha)(X - \alpha^2)(X - \alpha^3)(X - \alpha^4) = (X^2 - (\alpha + \alpha^2) \cdot X + \alpha^3) \times \\ &\times (X^2 - (\alpha^3 + \alpha^4) \cdot X + \alpha^7) = (X^2 + \alpha^4X + \alpha^3)(X^2 + \alpha^6X + \alpha^0) = \\ &= X^4 - (\alpha^4 + \alpha^6) \cdot X^3 + (\alpha^3 + \alpha^{10} + \alpha^0) \cdot X^2 - (\alpha^4 + \alpha^9) \cdot X + \alpha^3 = \\ &= X^4 + \alpha^3X^3 + \alpha^0X^2 - \alpha^1X + \alpha^3. \end{aligned}$$

Поменяв порядок расположения членов полинома на обратный и заменив знаки «минус» на «плюс», так как над двоичным полем  $+1 = -1$ , получим

$$P(X) = \alpha^3 + \alpha^1X + \alpha^0X^2 + \alpha^3X^3 + X^4. \quad (7.98)$$

Так как код Рида – Соломона является циклическим, кодирование в систематической форме аналогично процедуре двоичного кодирования, рассмотренной в пункте 7.3.3, т. е.

$$X^r M(X) = Q(X) \cdot P(X) + R(X), \quad (7.99)$$

где  $M(X)$  – полином сообщения.

Уравнение (7.99) логично переписать следующим образом:

$$R(X) = X^{n-k}M(X) \text{ по модулю } P(X). \quad (7.100)$$

Результирующий полином кодового слова  $F(X)$ , приведенный в уравнении (7.99), можно представить следующим образом:

$$F(X) = R(X) + X^r \cdot M(X). \quad (7.101)$$

**Пример 7.25.** Закодировать сообщение из трех символов  $010 = \alpha^1$ ,  $110 = \alpha^3$ ,  $111 = \alpha^5$  (см. рисунок 7.18) с помощью кода Рида – Соломона (7,3), генератор которого описывается выражением (7.98).

**Решение.** Умножим  $M(X) = \alpha^1 + \alpha^3X + \alpha^5X^2$  на  $X^r = X^4$ , получим

$$M(X) \cdot X^r = \alpha^1X^4 + \alpha^3X^5 + \alpha^5X^6.$$

Разделив  $M(X) \cdot X^r$  на  $P(X)$  с помощью таблиц 7.13 и 7.14, получим

$$R(X) = \alpha^0 + \alpha^2X + \alpha^4X^2 + \alpha^6X^3.$$

В итоге комбинация  $M(X)$ , закодированная кодом Рида – Соломона, согласно (7.101) примет вид

$$F(X) = \alpha^0 + \alpha^2X + \alpha^4X^2 + \alpha^6X^3 + \alpha^1X^4 + \alpha^3X^5 + \alpha^5X^6.$$

Кодирующее устройство для рассмотренного примера можно представить в виде  $(n-k)$ -разрядного регистра сдвига (рисунок 7.19).

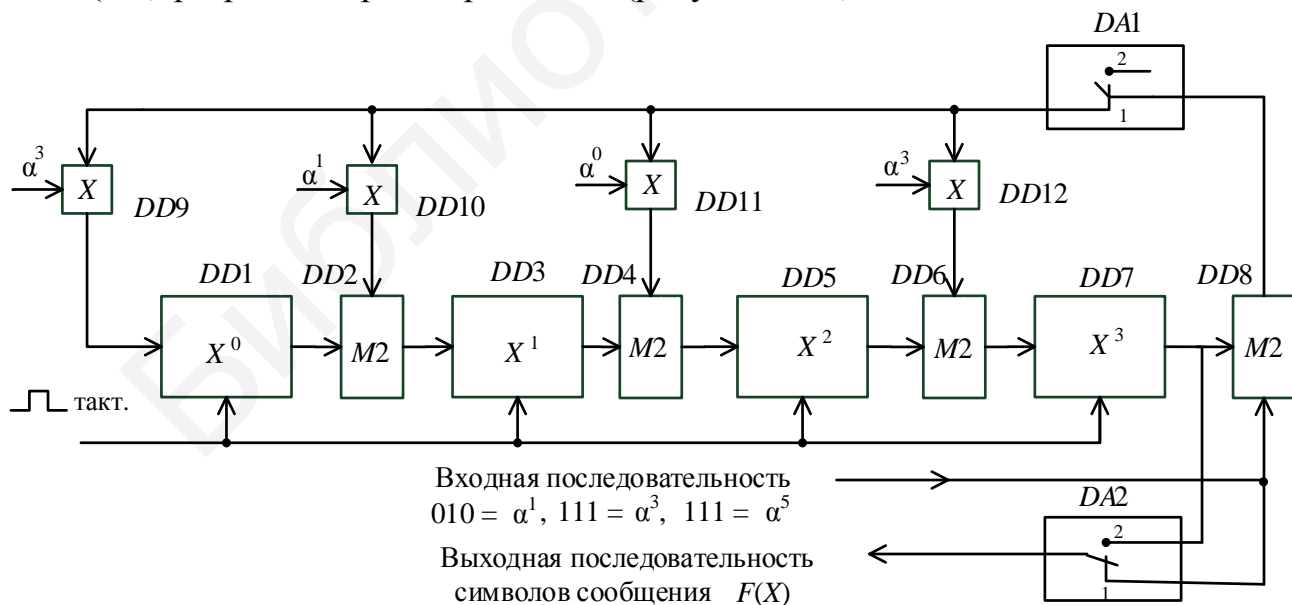


Рисунок 7.19 – Кодер кода Рида – Соломона (7,3)

Элементы умножителя на рисунке 7.19, взятые справа налево, соответствуют коэффициентам полинома в уравнении (7.98). Процесс кодирования является недвоичным, так как каждый разряд регистра сдвига образованный ячейками  $DD1$ ,  $DD3$ ,  $DD5$  и  $DD7$  содержит трехбитовый символ. Сложение и умножение трехбитовых символов производится в соответствии с таблицами 7.13 и 7.14.

В течение первых  $k$  тактовых импульсов ключ  $DA1$  находится в положении 1 и входные символы поступают в  $(n-k)$ -разрядный регистр сдвига. Ключ  $DA2$  находится также в положении 1, что обеспечивает одновременную передачу всех символов сообщения непосредственно на выход и на вход сумматора  $DD8$ . После передачи  $k$ -го символа ключ  $DA1$  размыкается (положение 2), а ключ  $DA2$  переходит в положение 2 и  $(n-k)$  тактовых импульсов выдвигают контрольные символы из регистра на выход. Таким образом, за  $n$  тактовых импульсов на выходе кодирующего устройства получаем кодовую комбинацию  $F(X)$ .

Последовательность действий в кодирующем устройстве показана в таблице 7.15. При этом необходимо учесть, что тактовые импульсы в ячейки  $DD1$ ,  $DD3$ ,  $DD5$  и  $DD7$  подаются с задержкой относительно входной последовательности на время переходных процессов в сумматоре  $DD8$ .

Таблица 7.15 – Последовательность образования кодовых комбинаций в коде (7,3) Рида – Соломона

Вход	Номер такта	Состояние ячеек					Выход
		$DD1$	$DD3$	$DD5$	$DD7$	$DD8$	
–	0	0	0	0	0	0	0
$\alpha^5$	1	$\alpha^1$	$\alpha^6$	$\alpha^5$	$\alpha^1$	$\alpha^5$	$\alpha^5$
$\alpha^3$	2	$\alpha^3$	0	$\alpha^2$	$\alpha^3$	$\alpha^0$	$\alpha^3$
$\alpha^1$	3	$\alpha^0$	$\alpha^2$	$\alpha^4$	$\alpha^6$	$\alpha^4$	$\alpha^1$
–	4	–	$\alpha^0$	$\alpha^2$	$\alpha^4$	–	$\alpha^6$
–	5	–	–	$\alpha^0$	$\alpha^2$	–	$\alpha^4$
–	6	–	–	–	$\alpha^0$	–	$\alpha^2$
–	7	–	–	–	–	–	$\alpha^0$

Как видно из таблицы 7.15, после третьего такта регистр содержит четыре контрольных символа  $\alpha^0$ ,  $\alpha^2$ ,  $\alpha^4$ ,  $\alpha^6$ , которые выдвигаются последующими четырьмя тактовыми импульсами на выход. Таким образом, на выходе появится кодовое слово, записанное в полиномиальной форме в следующем виде:

$$\begin{aligned}
 F(X) &= (100) + (001)X + (011)X^2 + (101)X^3 + (010)X^4 + (110)X^5 + (111)X^6 = \\
 &= \alpha^0 + \alpha^2 X + \alpha^4 X^2 + \alpha^6 X^3 + \alpha^1 X^4 + \alpha^3 X^5 + \alpha^5 X^6.
 \end{aligned}
 \tag{7.102}$$

Корни полиномиального генератора  $P(X)$  должны быть и корнями кодового слова, генерируемого  $P(X)$ , поскольку правильное кодовое слово имеет следующий вид:

$$F(X) = M(X)P(X)
 \tag{7.103}$$

Следовательно, произвольное кодовое слово, выражаемое через корень генератора  $P(X)$ , должно давать нуль. Представляется интересным, действительно ли полином кодового слова в уравнении (7.102) дает нуль, когда он выражается через какой-либо из четырех корней  $P(X)$ . Это означает проверку следующего:

$$F(\alpha) = F(\alpha^2) = F(\alpha^3) = F(\alpha^4) = 0.$$

Независимо выполнив вычисления для разных корней, т. е. подставив в (7.102)  $x = \alpha$ , получим:

$$\begin{aligned} F(\alpha) &= \alpha^0 + \alpha^3 + \alpha^6 + \alpha^9 + \alpha^5 + \alpha^8 + \alpha^{11} = \\ &= \alpha^0 + \alpha^3 + \alpha^6 + \alpha^2 + \alpha^5 + \alpha^1 + \alpha^4 = \\ &= \alpha^1 + \alpha^0 + \alpha^6 + \alpha^4 = \alpha^3 + \alpha^3 = 0, \end{aligned}$$

$$\begin{aligned} F(\alpha^2) &= \alpha^0 + \alpha^4 + \alpha^8 + \alpha^{12} + \alpha^9 + \alpha^{13} + \alpha^{17} = \\ &= \alpha^0 + \alpha^4 + \alpha^1 + \alpha^5 + \alpha^2 + \alpha^6 + \alpha^3 = \\ &= \alpha^5 + \alpha^6 + \alpha^0 + \alpha^3 = \alpha^1 + \alpha^1 = 0, \end{aligned}$$

$$\begin{aligned} F(\alpha^3) &= \alpha^0 + \alpha^5 + \alpha^{10} + \alpha^{15} + \alpha^{13} + \alpha^{18} + \alpha^{23} = \\ &= \alpha^0 + \alpha^5 + \alpha^3 + \alpha^1 + \alpha^6 + \alpha^4 + \alpha^2 = \\ &= \alpha^4 + \alpha^0 + \alpha^3 + \alpha^2 = \alpha^5 + \alpha^5 = 0, \end{aligned}$$

$$\begin{aligned} F(\alpha^4) &= \alpha^0 + \alpha^6 + \alpha^{12} + \alpha^{18} + \alpha^{17} + \alpha^{23} + \alpha^{29} = \\ &= \alpha^0 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha^1 = \\ &= \alpha^2 + \alpha^0 + \alpha^5 + \alpha^1 = \alpha^6 + \alpha^6 = 0. \end{aligned}$$

Эти вычисления показывают, что, как и ожидалось, кодовое слово, выражаемое через любой корень генератора  $P(X)$ , должно давать нуль.

### 7.4.3 Декодирование кодов Рида – Соломона

Декодирование недвоичных кодов отличается от декодирования двоичных кодов, рассмотренных в пункте 7.3.3. При двоичном декодировании декодеру нужно знать лишь расположение ошибки. Если известно, где находится ошибка, бит нужно менять с 1 на 0 или наоборот. При недвоичном декодировании необходимо узнать не только расположение ошибки, но и определить правильное значение символа, расположенного на этой позиции. Процесс декодирования состоит из следующих шагов: вычисление синдрома, локализация ошибки, нахождение значений ошибок и исправление принятого синдрома с помощью найденного полинома ошибок.

## 7.5 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.

### 7.5.1 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.

### 7.5.2 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:

$$N = n_{\text{ч}}! \quad (7.104)$$

Например, при трех частотах получается шесть комбинаций:  $f_1 f_2 f_3$ ,  $f_1 f_3 f_2$ ,  $f_2 f_1 f_3$ ,  $f_2 f_3 f_1$ ,  $f_3 f_1 f_2$ ,  $f_3 f_2 f_1$ . Данный код позволяет обнаруживать одиночные искажения, так как в сообщении каждый элемент встречается только один раз.

### 7.5.3 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:

$$N = A_n^m = n! / (m!(n-m)!). \quad (7.105)$$

Если, например,  $n=3$ ,  $m=2$ , то общее число комбинаций равно шести:  $f_1 f_2$ ,  $f_1 f_3$ ,  $f_2 f_1$ ,  $f_2 f_3$ ,  $f_3 f_1$ ,  $f_3 f_2$ . Комбинации передаются последовательно. Этот код позволяет обнаруживать одиночные ошибки путем счета символов, содержащихся в сообщении.



### 7.5.4 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:

$$N = C_n^m = n! / (m!(n-m)!). \quad (7.106)$$

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.

### 7.5.5 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 разные разряды в сообщении. Пусть необходимо передать кодовую комбинацию  $G(x) = 10011101$  сменно-качественным кодом. Для этой цели 1 передается частотой  $f_1$ , 0 – частотой  $f_2$ , а повторение символа (0 или 1) – частотой  $f_3$ . Тогда комбинация принимает вид  $f_1f_2f_3f_1f_3f_1f_2f_1$ . Нетрудно установить лишь те искажения, в результате которых соседние радиоимпульсы получают одинаковое значение частотного признака, другие искажения не обнаруживаются.

## 8 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.

### 8.1 Шифраторы кода $C_n^1$ в двоичный код

Функциональная схема этого шифратора представлена на рисунке 8.1, а на рисунке 8.2 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 (таблица 8.1), работающие как схемы ИЛИ по 0. В результате на выходе элементов  $DD8–DD10$  будет сформирован двоичный код, соответствующий номеру нажатой кнопки.

В системах ТУ-ТС при формировании адреса объекта и команды находят применение шифраторы кода  $C_n^1$  в двоичный код с использованием мультиплексов (рисунок 8.3).

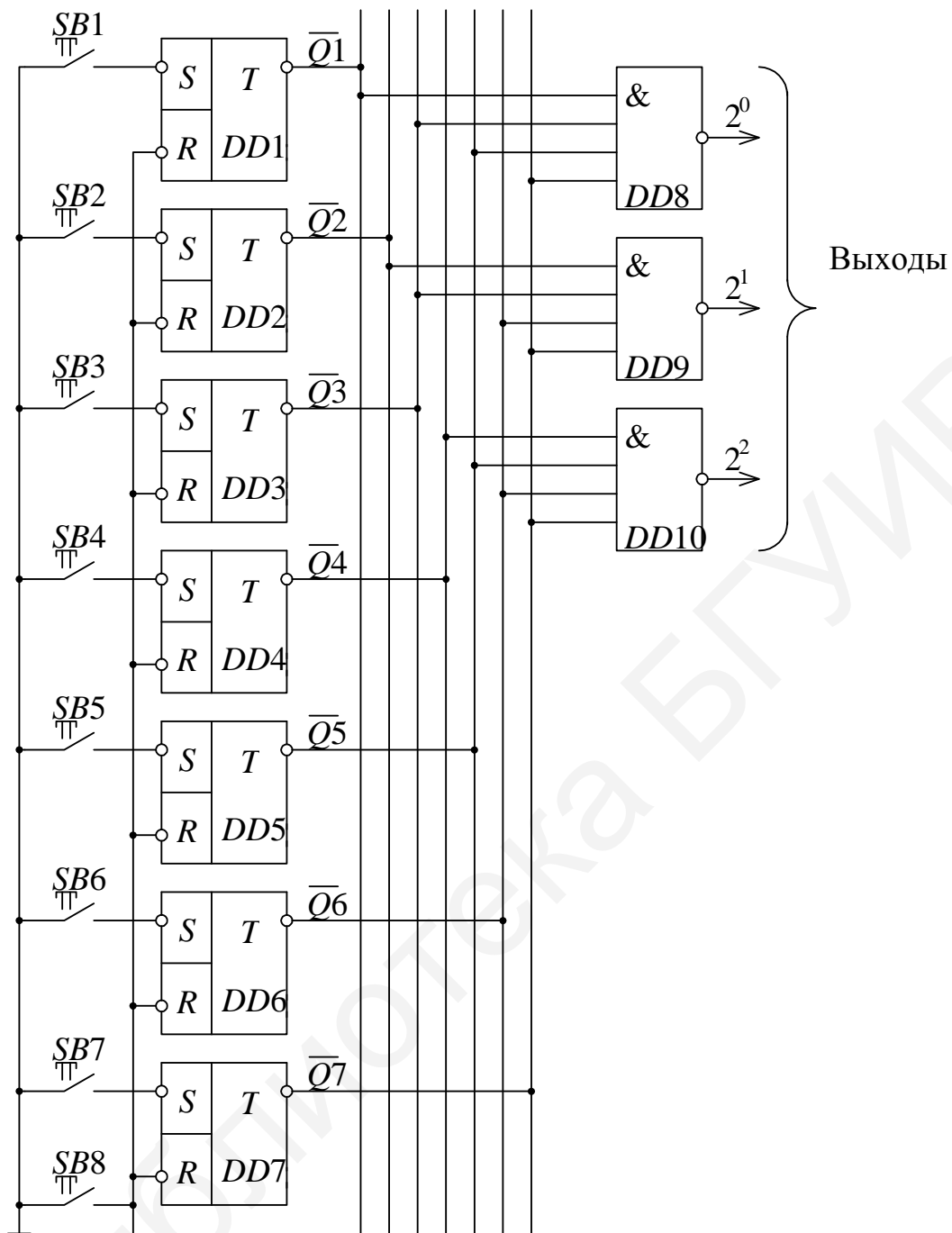


Рисунок 8.1 – Функциональная схема шифратор кода  $C_n^1$  в код 1-2-4

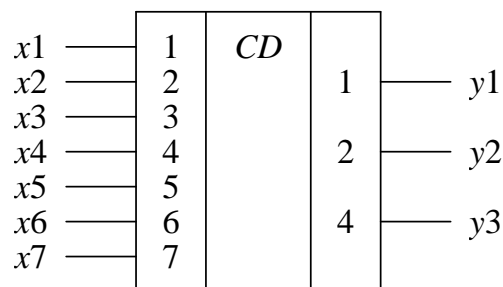


Рисунок 8.2 – Условное обозначение шифратора  $C_7^1$  в код 1-2-4

Таблица 8.1 – Состояние элементов шифратора

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

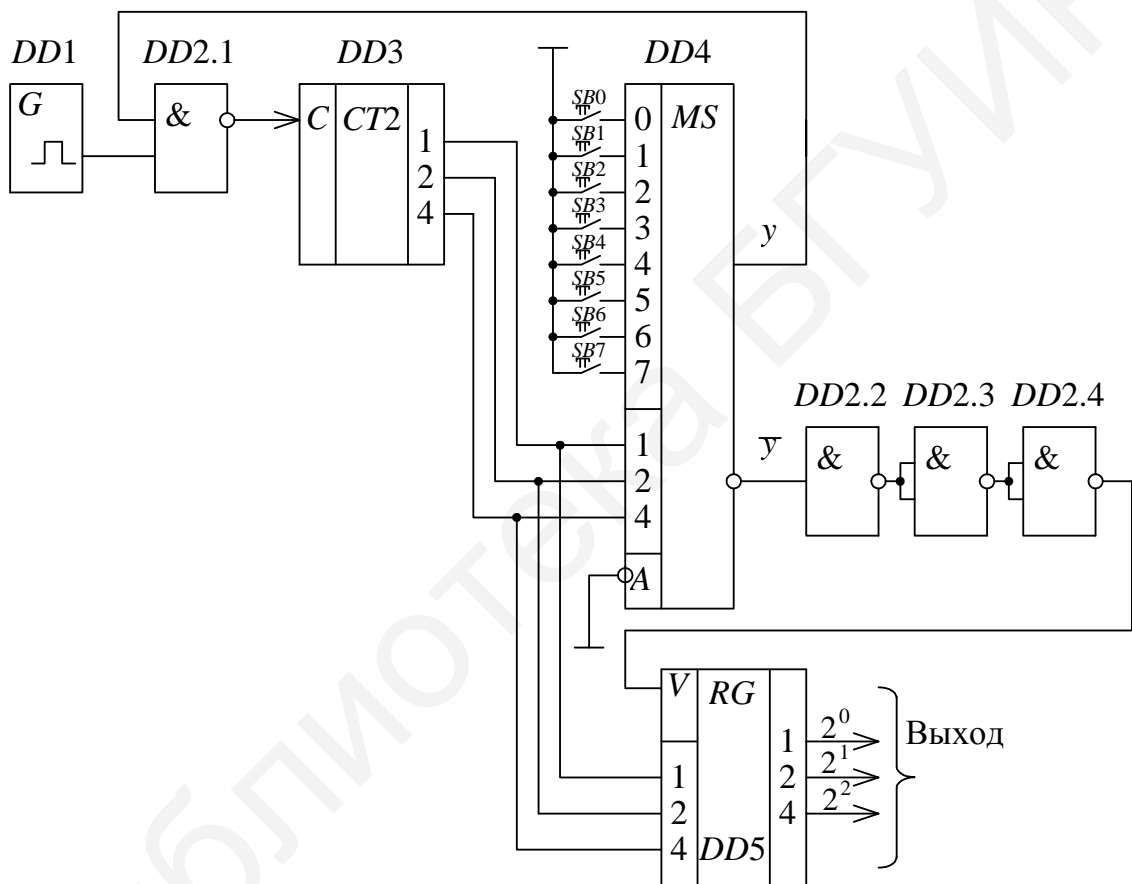


Рисунок 8.3 – Функциональная схема преобразователя десятичного кода в двоичный код

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 кнопка *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 по-

ступление 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 ИВ1 и ИВ3. Более подробно остановимся на микросхеме ИВ1. Это приоритетный шифратор 8 в 3 (рисунок 8.4), принимающий напряжение низкого уровня на один из восьми параллельных адресных входов  $\bar{I1}$ – $\bar{I8}$ .

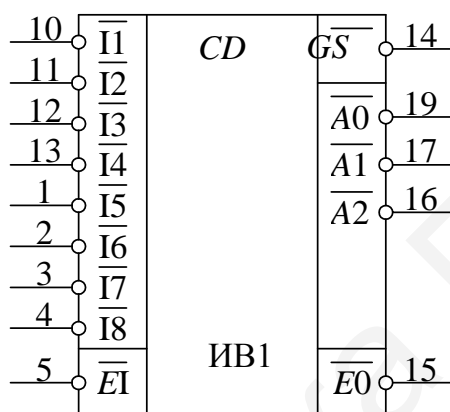


Рисунок 8.4 – Условное обозначение микросхемы ИВ1

На трех выходах  $\bar{A0}$ – $\bar{A2}$  появляется двоичный код, пропорциональный номеру входа, оказавшегося активным. Состояния шифратора приведены в таблице 8.2.

Таблица 8.2 – Состояния шифратора ИВ1

Входы									Выходы				
$\bar{E1}$	$\bar{I1}$	$\bar{I2}$	$\bar{I3}$	$\bar{I4}$	$\bar{I5}$	$\bar{I6}$	$\bar{I7}$	$\bar{I8}$	$\bar{GS}$	$\bar{A0}$	$\bar{A1}$	$\bar{A2}$	$\bar{EO}$
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	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.

## 8.2 Дешифратор двоичного кода в десятичный код

Простейший дешифратор, выполненный на элементах И-НЕ, НЕ (рисунок 8.5), 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.

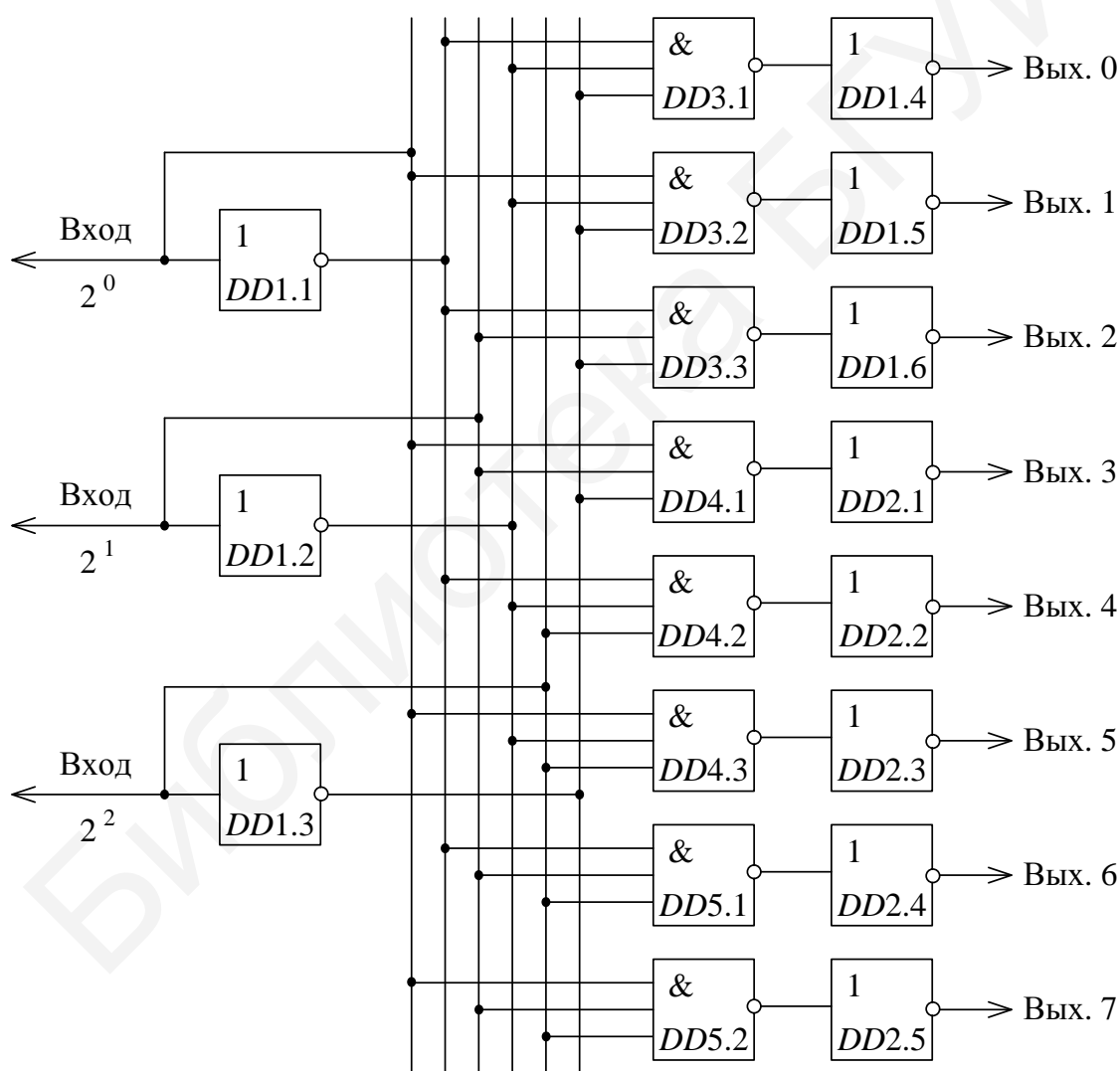


Рисунок 8.5 – Функциональная схема дешифратора двоичного кода в десятичный код

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 таблице 8.3.

Таблица 8.3 – 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 рисунке 8.6.

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.

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

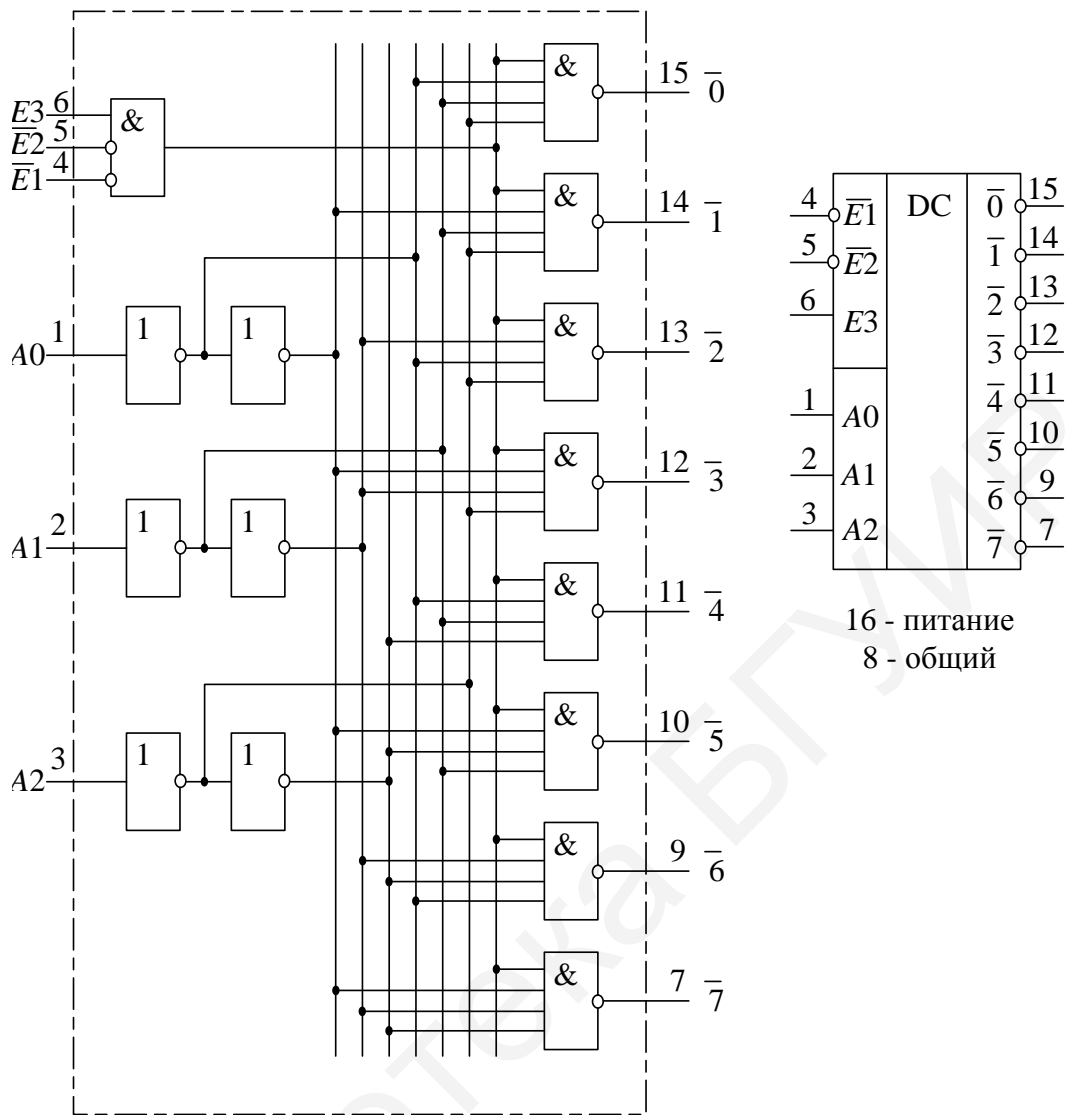


Рисунок 8.6 – Структура, условное обозначение и цоколевка микросхемы ИД7

Таблица 8.4 – Состояния дешифратора ИД7

Входы						Выходы							
$\bar{E}1$	$\bar{E}2$	E3	A0	A1	A2	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$
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>



На рисунке 8.7 показан дешифратор на 32 выхода. Для его реализации потребовалось четыре дешифратора ИД7 и дополнительный инвертор *DD5*.

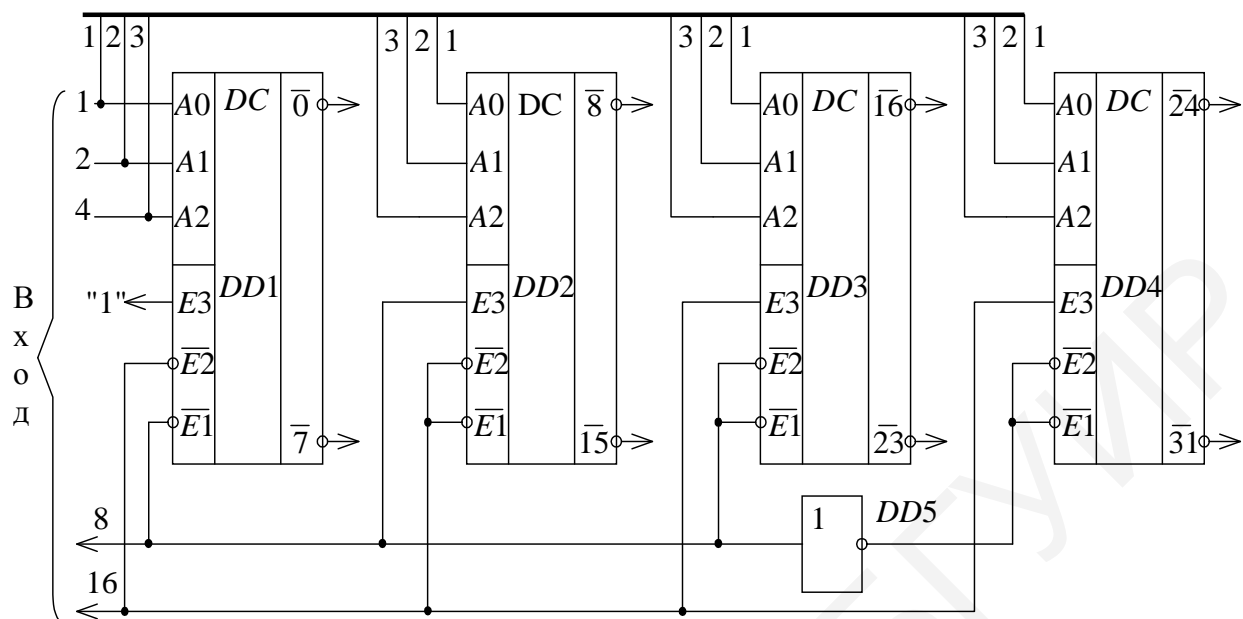


Рисунок 8.7 – Параллельное соединение ИД7

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 таблице 8.5.

Таблица 8.5 – Состояния элементов разрешения дешифраторов

Вход		Дешифратор							
		<i>DD1</i>				<i>DD2</i>			
		Входы			Выход	Входы			Выход
$2^3 = 8$	$2^4 = 16$	$\overline{E1}$	$\overline{E2}$	$E3$		$\overline{E1}$	$\overline{E2}$	$E3$	
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
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

### 8.3 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 (рисунок 8.8). Принцип работы поясним на примере дешифратора ИД6, который преобразует двоичный код, поступающий на входы  $A_0$ – $A_3$ , в сигнал низкого уровня, появляющийся на десятичном выходе  $\bar{0}$ – $\bar{9}$ .

Состояния дешифратора приведены в таблице 8.6.

Таблица 8.6 – 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  $\bar{0}$ – $\bar{9}$  появляется напряжение высокого уровня.

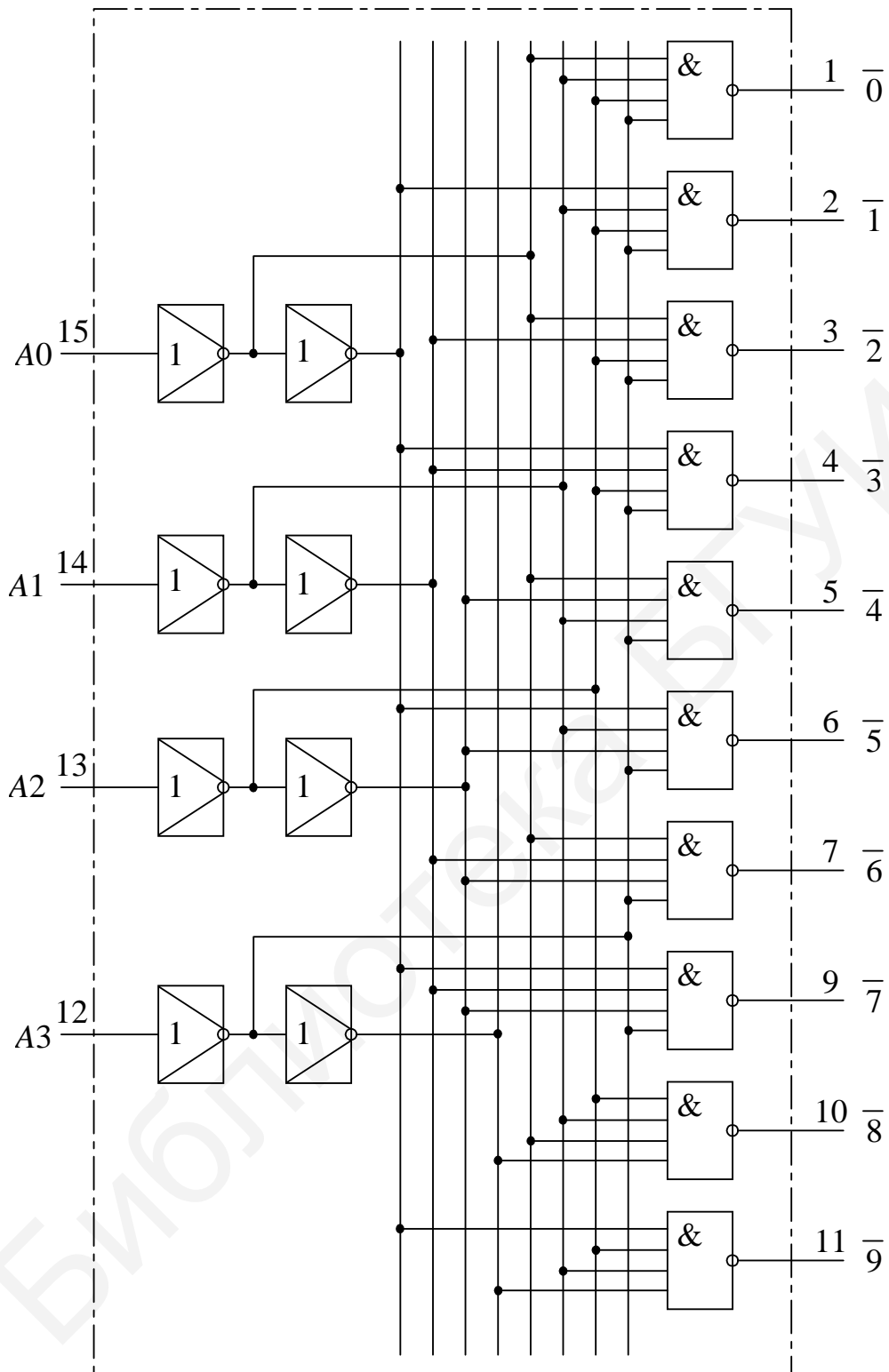


Рисунок 8.8 – Структура и цоколевка ИД6

## 8.4 Преобразователи двоичного кода в двоично-десятичный код и обратно

Наибольшее применение нашли два типа преобразователей – на базе пересчетных схем и на базе интегральных микросхем ПР6 и ПР7.

На рисунке 8.9 приведена функциональная электрическая схема восьмиразрядного преобразователя двоичного кода в двоично-десятичный первого типа.

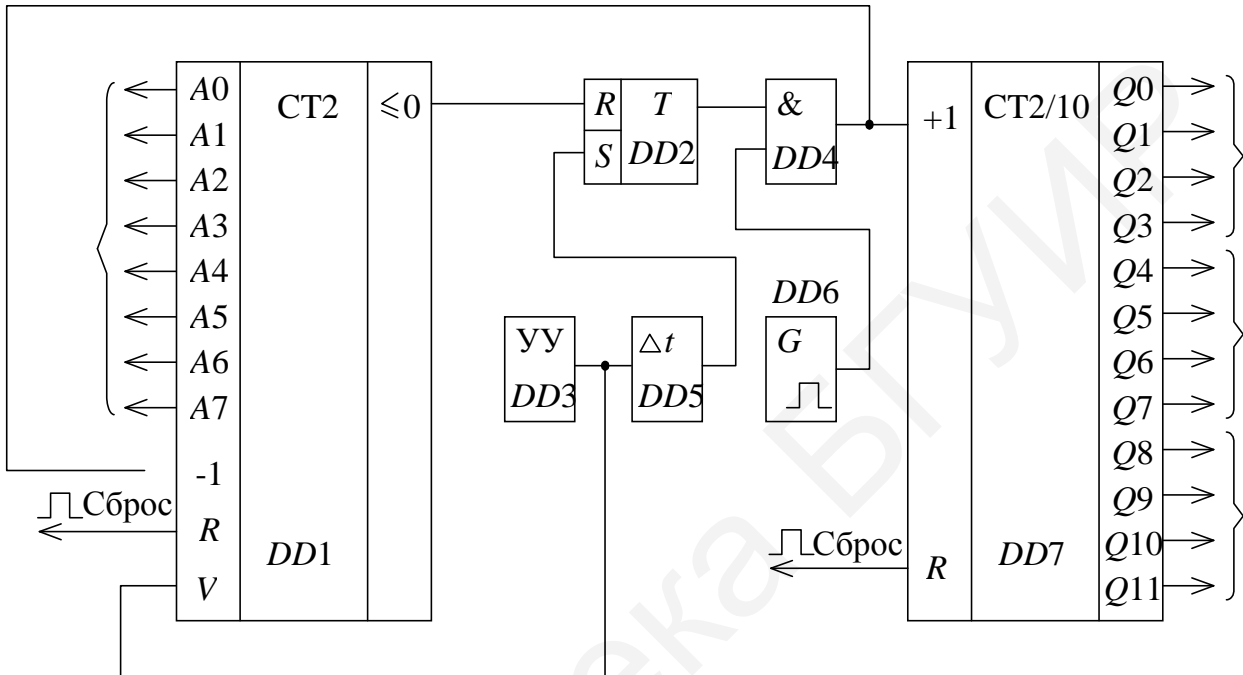


Рисунок 8.9 – Функциональная электрическая схема преобразования двоичного кода в двоично-десятичный

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  $DD1$ .

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 рисунке 8.9, но поменять местами счетчики *DD1* и *DD7*.

Преобразователи двоично-десятичного кода в двоичный и двоичного кода в двоично-десятичный код можно построить на микросхемах ПР6 (рисунок 8.10) и ПР7 (рисунок 8.11) соответственно. Основой таких преобразователей является запоминающая матрица емкостью 256 битов. Ячейки матрицы соединены в соответствии с программой преобразования. Матрицей управляет дешифратор с 5 входами и 32 выходами.

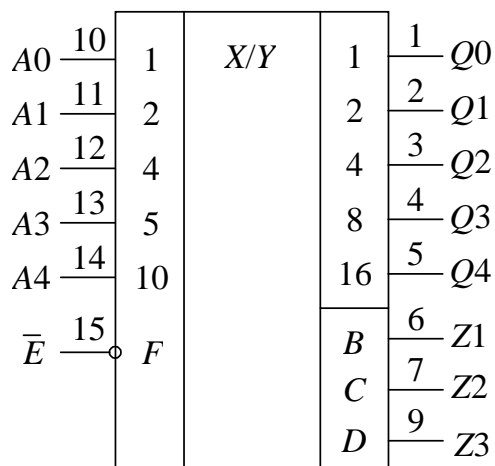


Рисунок 8.10 – Условное обозначение и цоколевка микросхемы ПР6

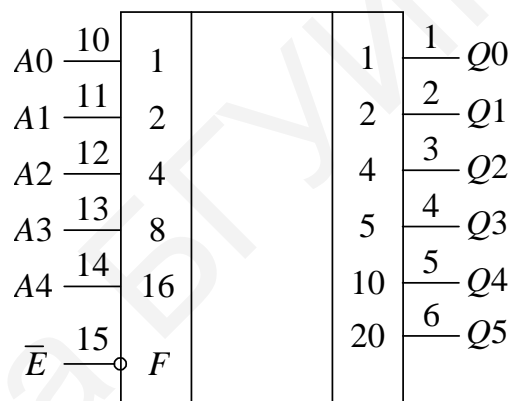


Рисунок 8.11 – Условное обозначение и цоколевка микросхемы ПР7

На входы *A0–A4* 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 таблицах 8.7 и 8.8 соответственно.

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  $W = 9 - X = (Z3, Z2, A1, Z1)$  (рисунок 8.12); при  $A4 = 1$  – преобразование двоично-десятичного числа  $X = (A3, A2, A1, A0)$  в дополнение  $W1$  до числа 10 (рисунок 8.13) по правилу

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

Таблица 8.7 – Состояния логических уровней при преобразовании двоично-десятичных слов в двоичные микросхемой ПР6

Номер 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	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	1	0	0	0	0	0	0	0	1	0	1
6	0	1	0	0	1	0	0	0	0	1	1	0
7	0	1	0	1	0	0	0	0	0	1	1	1
8	0	1	0	1	1	0	0	1	0	0	0	0
9	0	1	1	0	0	0	0	1	0	0	0	1
10	1	0	0	0	0	0	0	1	0	1	1	0
11	1	0	0	0	1	0	0	1	0	1	1	1
12	1	0	0	1	0	0	0	1	1	1	0	0
13	1	0	0	1	1	0	0	1	1	1	0	1
14	1	0	1	0	0	0	0	1	1	1	1	0
15	1	1	0	0	0	0	0	1	1	1	1	1
16	1	1	0	0	1	0	1	0	0	0	0	0
17	1	1	0	1	0	0	1	0	0	0	0	1
18	1	1	0	1	1	0	1	0	0	0	1	0
19	1	1	1	0	0	0	1	0	0	0	1	1
Любое	x	x	X	x	x	1	1	1	1	1	1	1

Таблица 8.8 – Состояния логических уровней при преобразовании двоичного кода в двоично-десятичный микросхемой ПР7

Номер слова	Вход						Выход					
	A4	A3	A2	A1	A0	$\bar{E}$	Q5	Q4	Q3	Q2	Q1	Q0
1	2	3	4	5	6	7	8	9	10	11	12	13
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

Продолжение таблицы 8.8

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

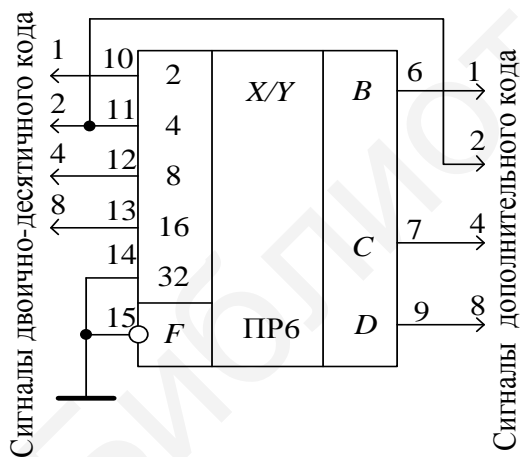


Рисунок 8.12 – Преобразование в дополнение до числа 9

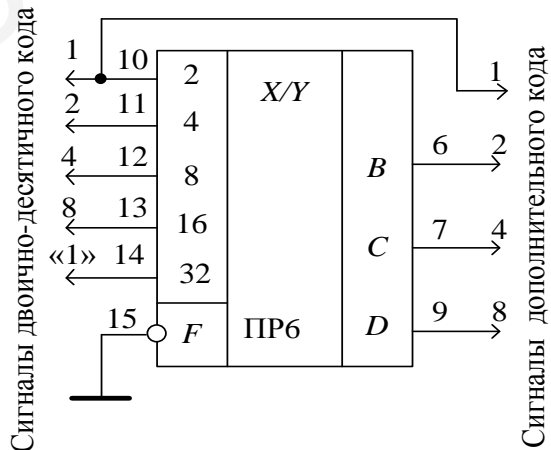


Рисунок 8.13 – Преобразование в дополнение до числа 10

Как правило, разрядности одной микросхемы в большинстве случаев недостаточно, поэтому преобразователи соединяют каскадно. Соединение микросхем ПР6 для обработки сигналов двоично-десятичных чисел 0–99 показано на рисунке 8.14.

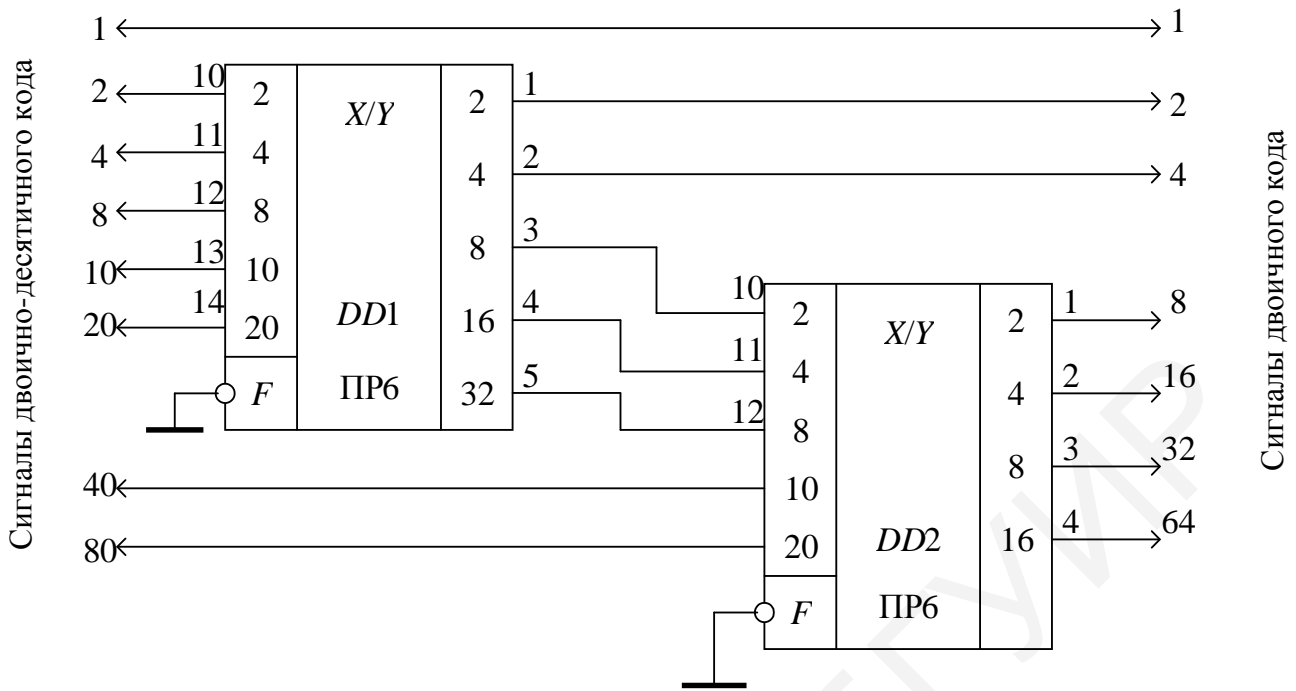


Рисунок 8.14 – Преобразование двоично-десятичных чисел 0–99 в двоичный код

На рисунке 8.15 приведена схема преобразователя на микросхемах PP7 для двоичных чисел 0–255.

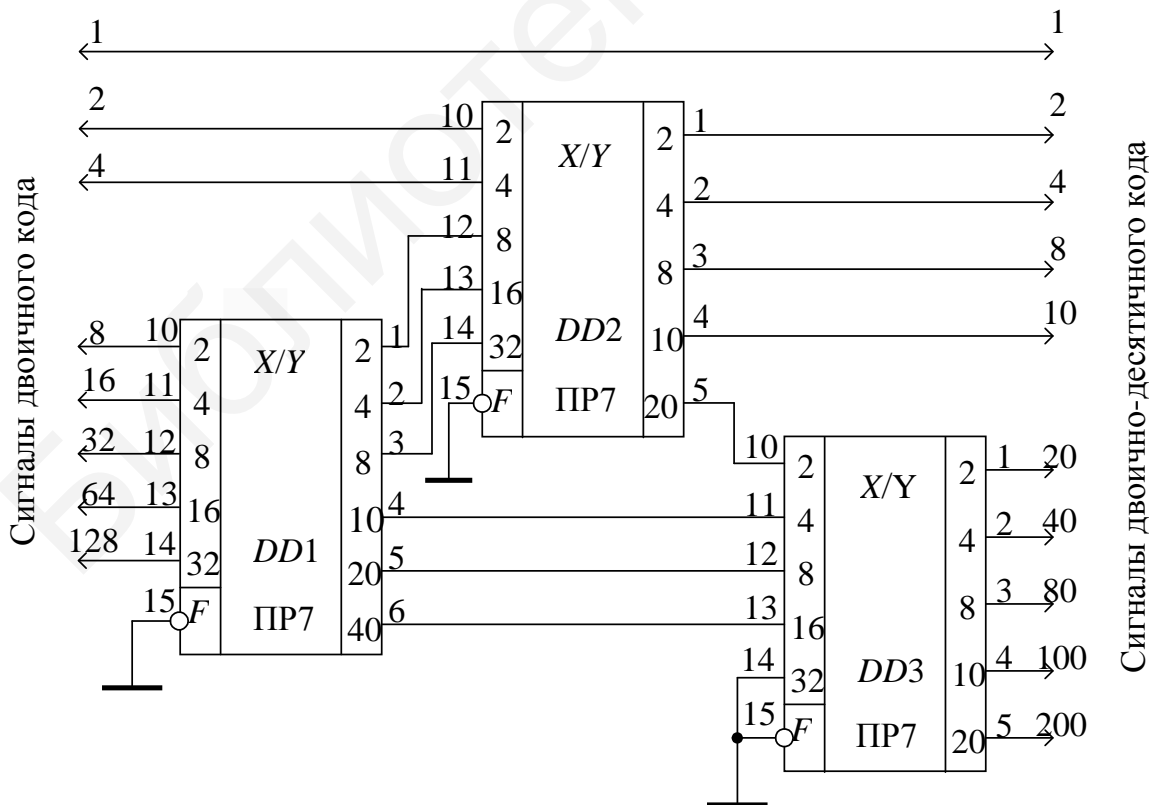


Рисунок 8.15 – Преобразование двоичных чисел 0–255 в двоично-десятичный код



## 8.5 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:

$$\begin{aligned} x_1 &= y_1, \\ x_2 &= \bar{y}_4\bar{y}_3y_2 + y_4\bar{y}_3\bar{y}_2 + \bar{y}_4y_3\bar{y}_2y_1, \\ x_3 &= \bar{y}_4y_3y_2 + \bar{y}_4y_3\bar{y}_1 + y_4\bar{y}_3\bar{y}_2, \\ x_4 &= \bar{y}_4y_3y_2 + \bar{y}_4y_3y_1 + y_4\bar{y}_3\bar{y}_2. \end{aligned} \quad (8.1)$$

Функциональная электрическая схема преобразователя кода 8-4-2-1 в код 2-4-2-1, построенного по выражению (8.1), приведена на рисунке 8.16.

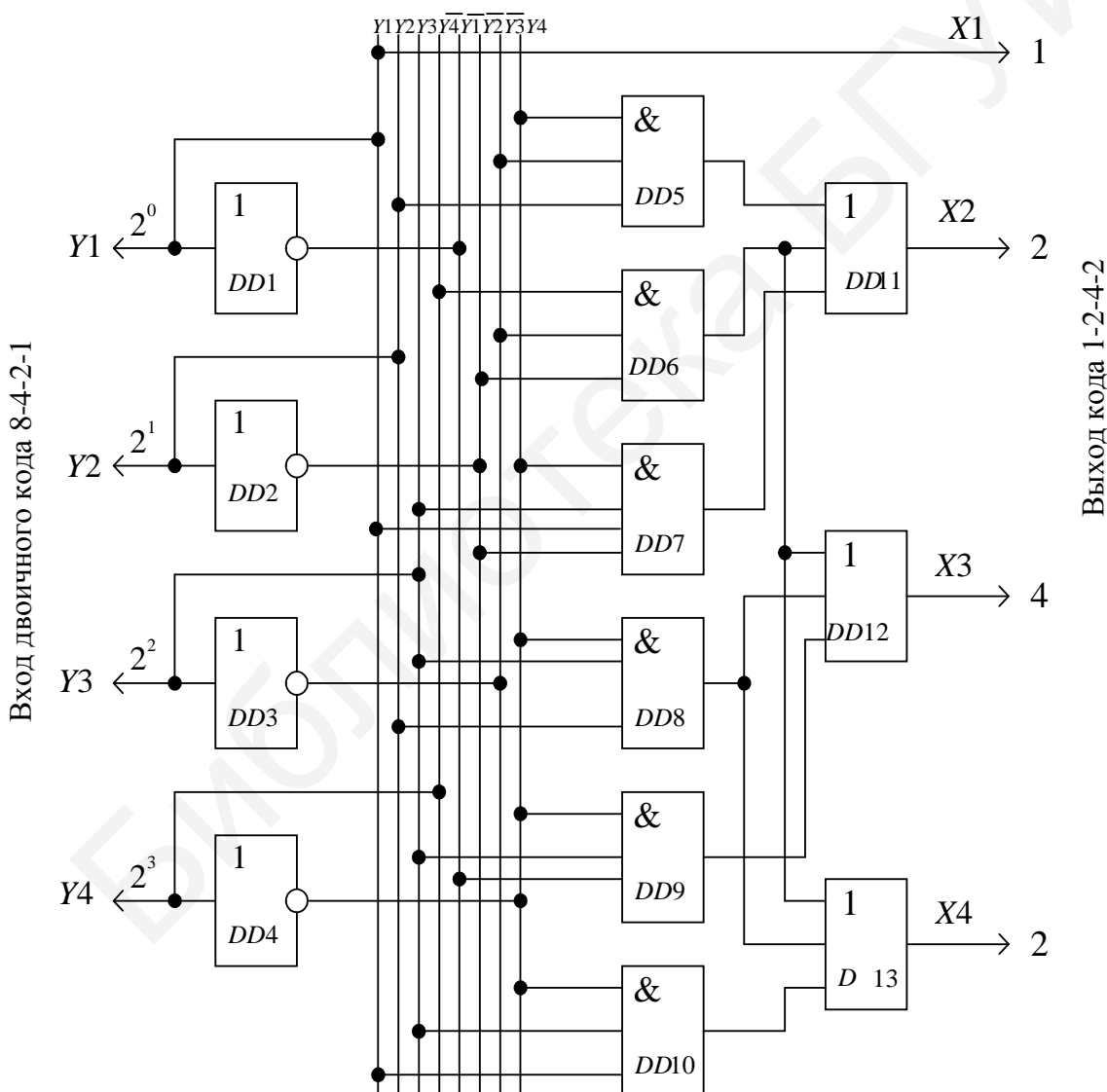


Рисунок 8.16 – Преобразование кода 8-4-2-1 в код 2-4-2-1

Таблица 8.9, 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–DD13* и *X1*. В остальных строках таблицы записаны единицы или нули, которые образуются на выходах промежуточных элементов.

Таблица 8.9 – Преобразование кода 8-4-2-1 в код 2-4-2-1

СОСТОЯНИЕ	ВХОДОВ	$y_1(2^0)$	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	1	1
	ПРОМЕЖУТОЧНЫХ ЭЛЕМЕНТОВ	<i>DD1</i>	1	0	1	0	1	0	1	0	1	0
		<i>DD2</i>	1	1	0	0	1	1	0	0	1	1
		<i>DD3</i>	1	1	1	1	0	0	0	0	1	1
		<i>DD4</i>	1	1	1	1	1	1	1	1	0	0
		<i>DD5</i>	0	0	1	1	0	0	0	0	0	0
		<i>DD6</i>	0	0	0	0	0	0	0	0	1	1
		<i>DD7</i>	0	0	0	0	0	1	0	0	0	0
		<i>DD8</i>	0	0	0	0	0	0	1	1	0	0
		<i>DD9</i>	0	0	0	0	1	0	1	0	0	0
		<i>DD10</i>	0	0	0	0	0	1	0	1	0	0
	ВЫХОДОВ	$x_1(1)$	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	
Десятичное число		0	1	2	3	4	5	6	7	8	9	

### 8.6 Преобразователь самодополняющегося двоично-десятичного кода 2-4-2-1 в двоичный код 8-4-2-1

На основании анализа столбцов 2 и 3 таблицы 6.3 и введенных обозначений разрядов в подразделе 8.5, разряды кода 2-4-2-1 связаны с разрядами кода 8-4-2-1 следующими выражениями:

$$\begin{aligned}
 y_1 &= x_1, \\
 y_2 &= \overline{x_4 x_3 x_2} + x_4 x_3 \overline{x_2}, \\
 y_3 &= x_4 x_3 x_2 x_1 + x_4 x_3 \overline{x_2} + \overline{x_3 x_2 x_1}, \\
 y_4 &= x_4 x_3 x_2.
 \end{aligned}
 \tag{8.2}$$

Функциональная электрическая схема преобразователя, построенная на основании выражений (8.2), приведена на рисунке 8.17, а принцип работы поясняется таблицей 8.10.

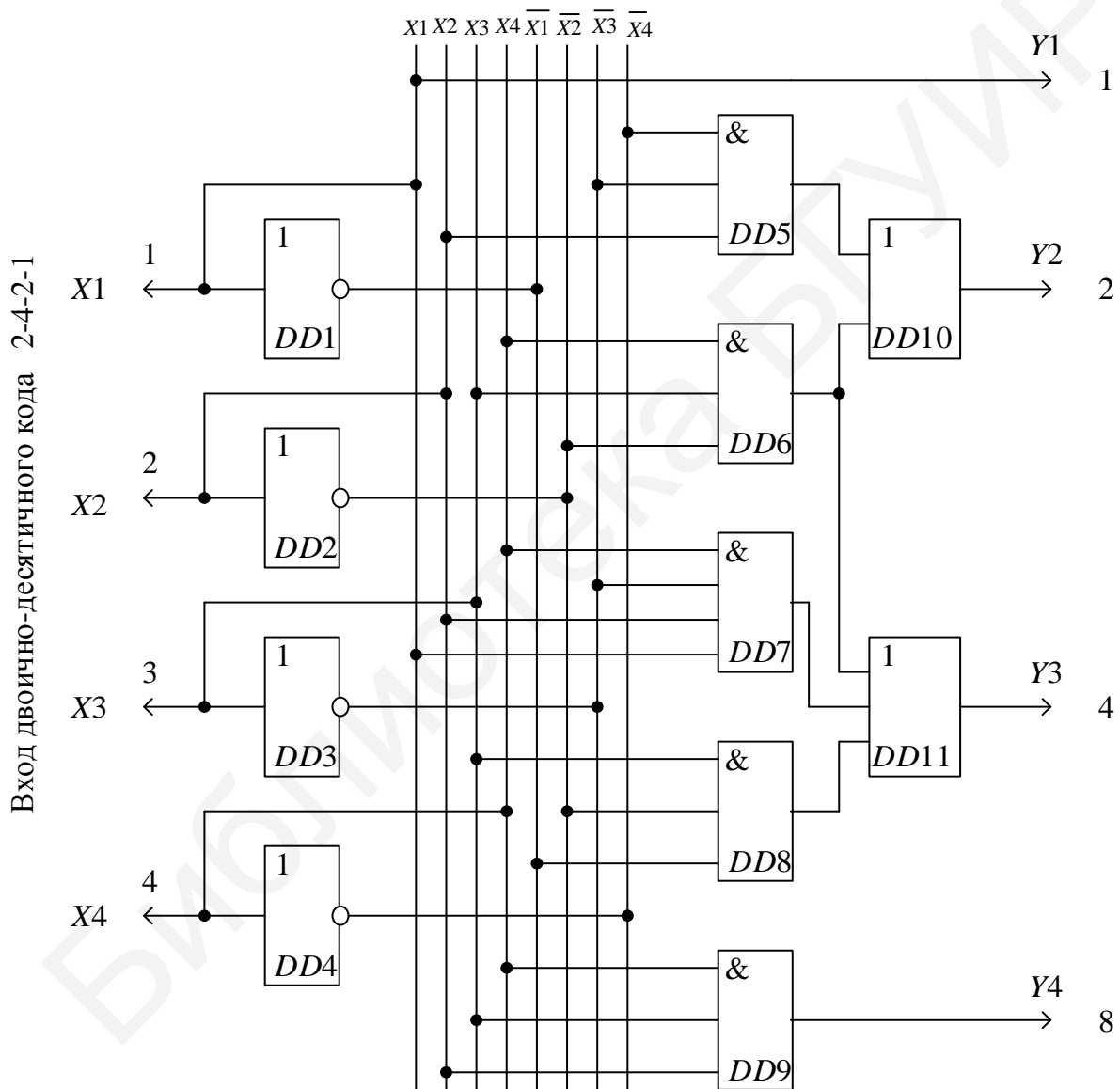


Рисунок 8.17 – Схема преобразователя кода 2-4-2-1 в код 8-4-2-1

Таблица 8.10 – Преобразование кода 2-4-2-1 в код 8-4-2-1

СОСТОЯНИЕ	ВХОДОВ	X1(1)	0	1	0	1	0	1	0	1	0	1
		X2(2)	0	0	1	1	0	1	0	0	1	1
		X3(4)	0	0	0	0	1	0	1	1	1	1
		X4(2)	0	0	0	0	0	1	1	1	1	1
	промежуточных элементов	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
	ВЫХОДОВ	Y1(1)	0	1	0	1	0	1	0	1	0	1
		Y2(2)	0	0	1	1	0	0	1	1	0	0
		Y3(4)	0	0	0	0	1	1	1	1	0	0
Y4(8)		0	0	0	0	0	0	0	0	1	1	
Десятичное число		0	1	2	3	4	5	6	7	8	9	

### 8.7 Преобразователь кода Грея в двоичный код и обратно

Как указано в пункте 6.3.6, значение каждого последующего разряда двоичного числа находится путем сложения по модулю 2 этого же разряда в коде Грея с предыдущим (6.8), т. е.

$$a_i = \sum_{j=i}^i b_j \pmod{2},$$

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 (6.9), т. е.

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

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

На рисунке 8.18 показана функциональная схема преобразователя четырехразрядного кода Грея в четырехразрядный двоичный код, а на рисунке 8.19 – схема обратного преобразования. Соответствие кода Грея двоичному и наоборот показано в таблице 8.11.

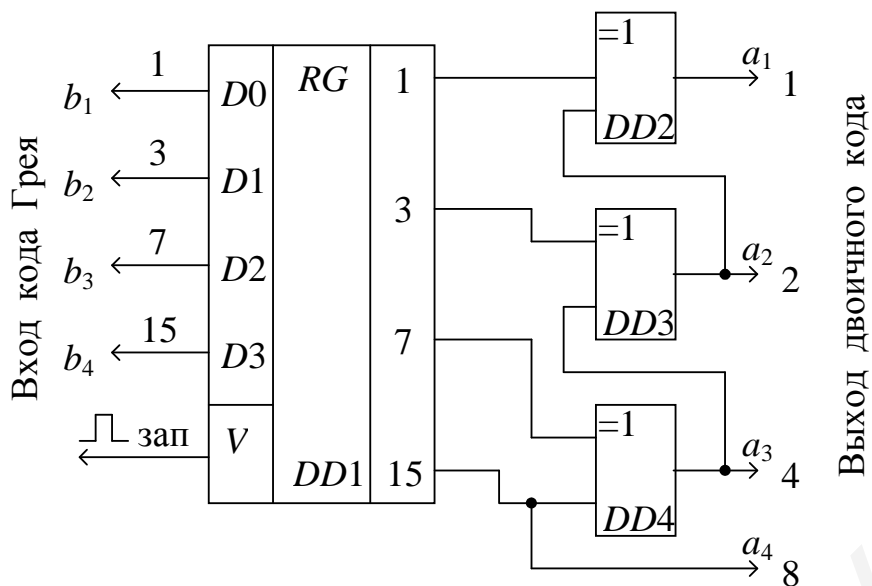


Рисунок 8.18 – Преобразователь кода Грея в двоичный

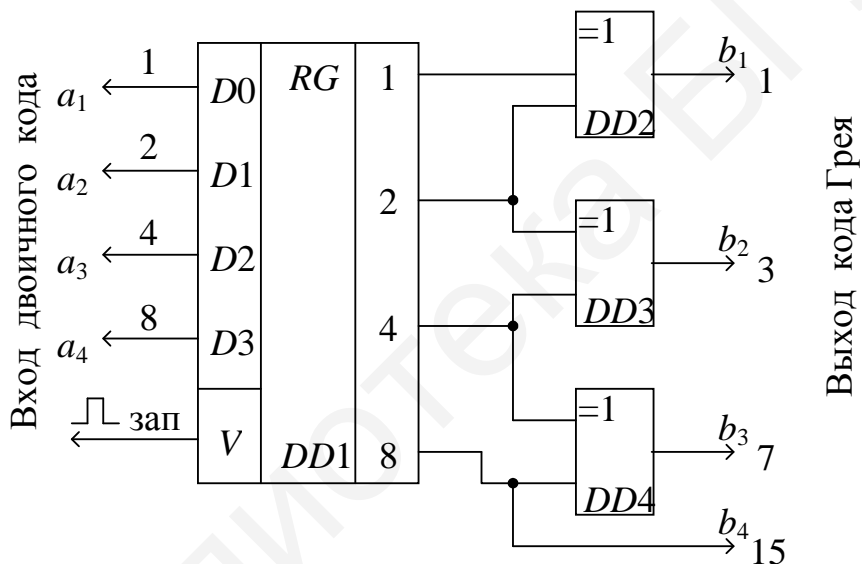


Рисунок 8.19 – Преобразование двоичного кода в код Грея

Таблица 8.11 – 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 рисунке 8.20.

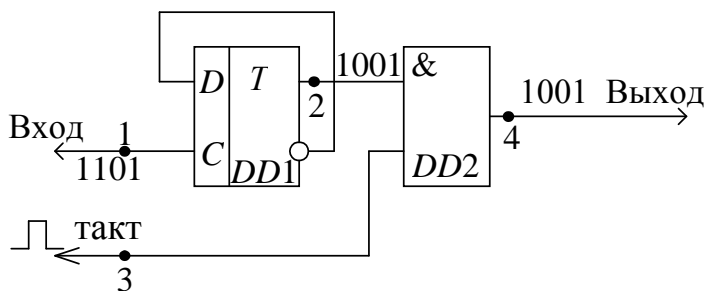


Рисунок 8.20 – Преобразователь последовательного кода Грея в двоичный

Кодовые комбинации на вход триггера *DD1* поступают начиная со старшего разряда. Триггер осуществляет операцию суммирования по модулю 2. Принцип работы наглядно поясняется временными диаграммами, приведенными на рисунке 8.21.

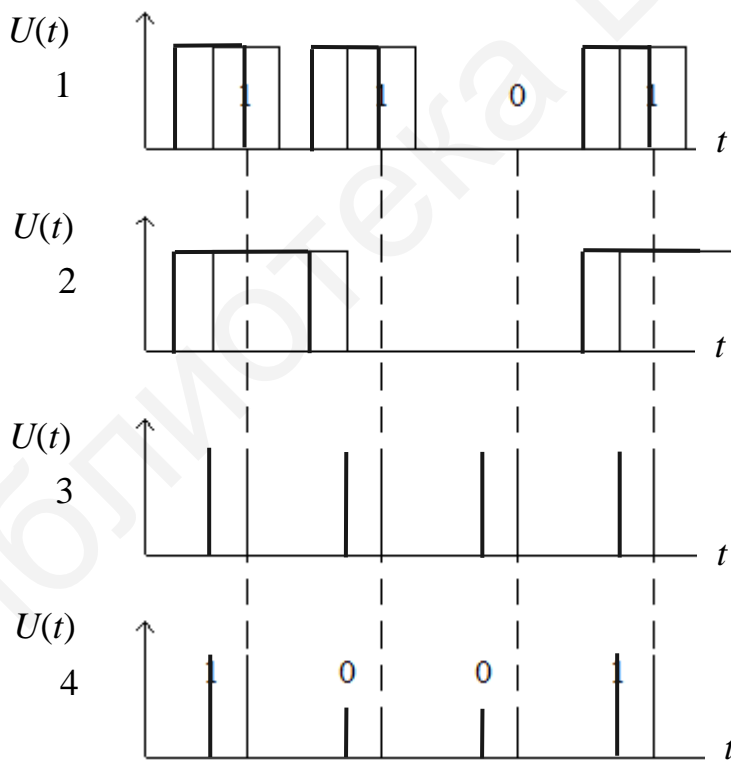


Рисунок 8.21 – Временные диаграммы работы преобразователя кода Грея в двоичный

## 8.8 Технические средства кодирования и декодирования эффективных кодов

Из материала, приведенного в пункте 6.4.3, 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 таблице 6.6.

Схема кодера источника представлена на рисунке 8.22.

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. таблицу 6.6). 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 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  $X8$  таблицы 6.6. 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 ячейки  $D0$  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.

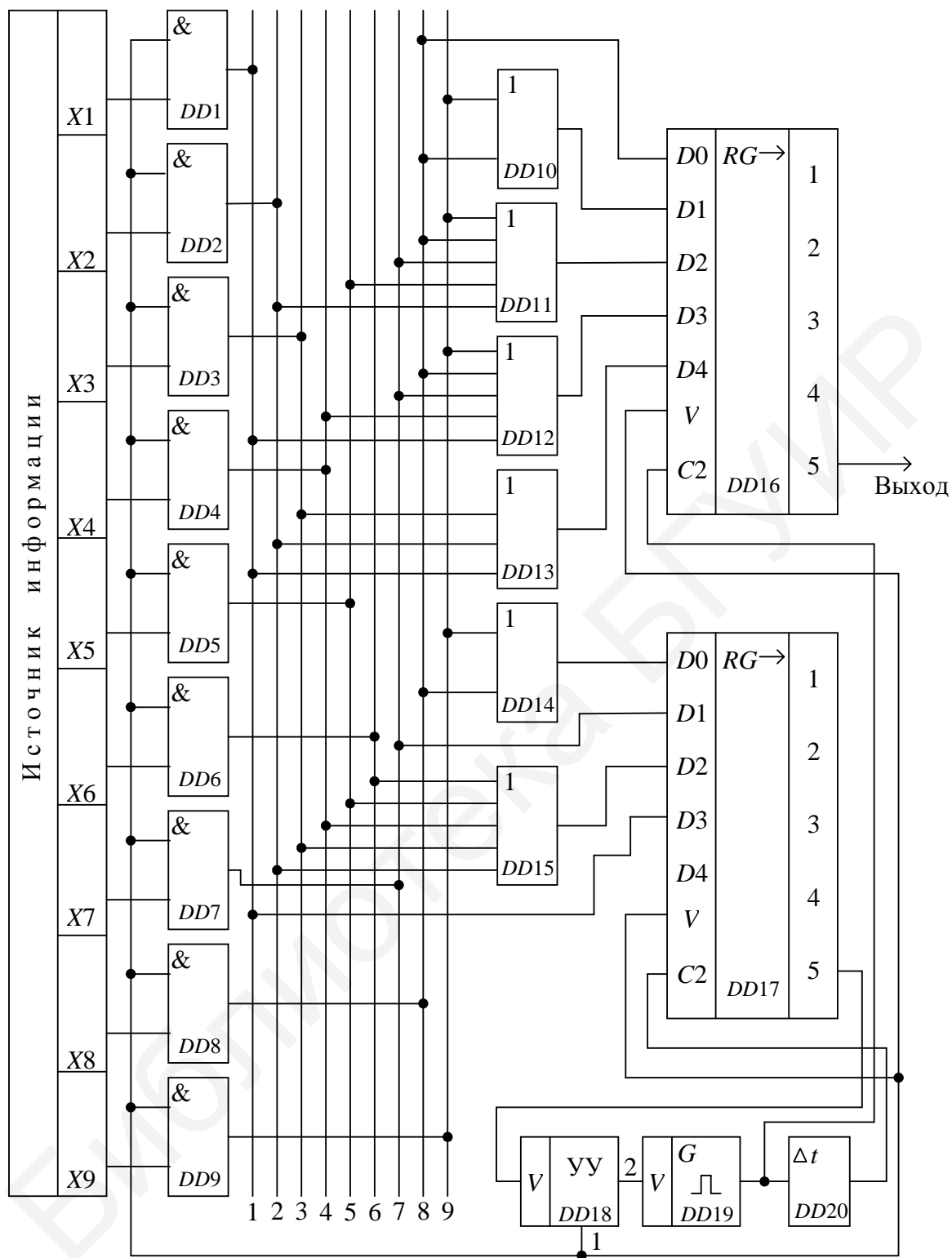


Рисунок 8.22 – Кодирование эффективных кодов



На рисунке 8.23 приведена схема декодирующего устройства.

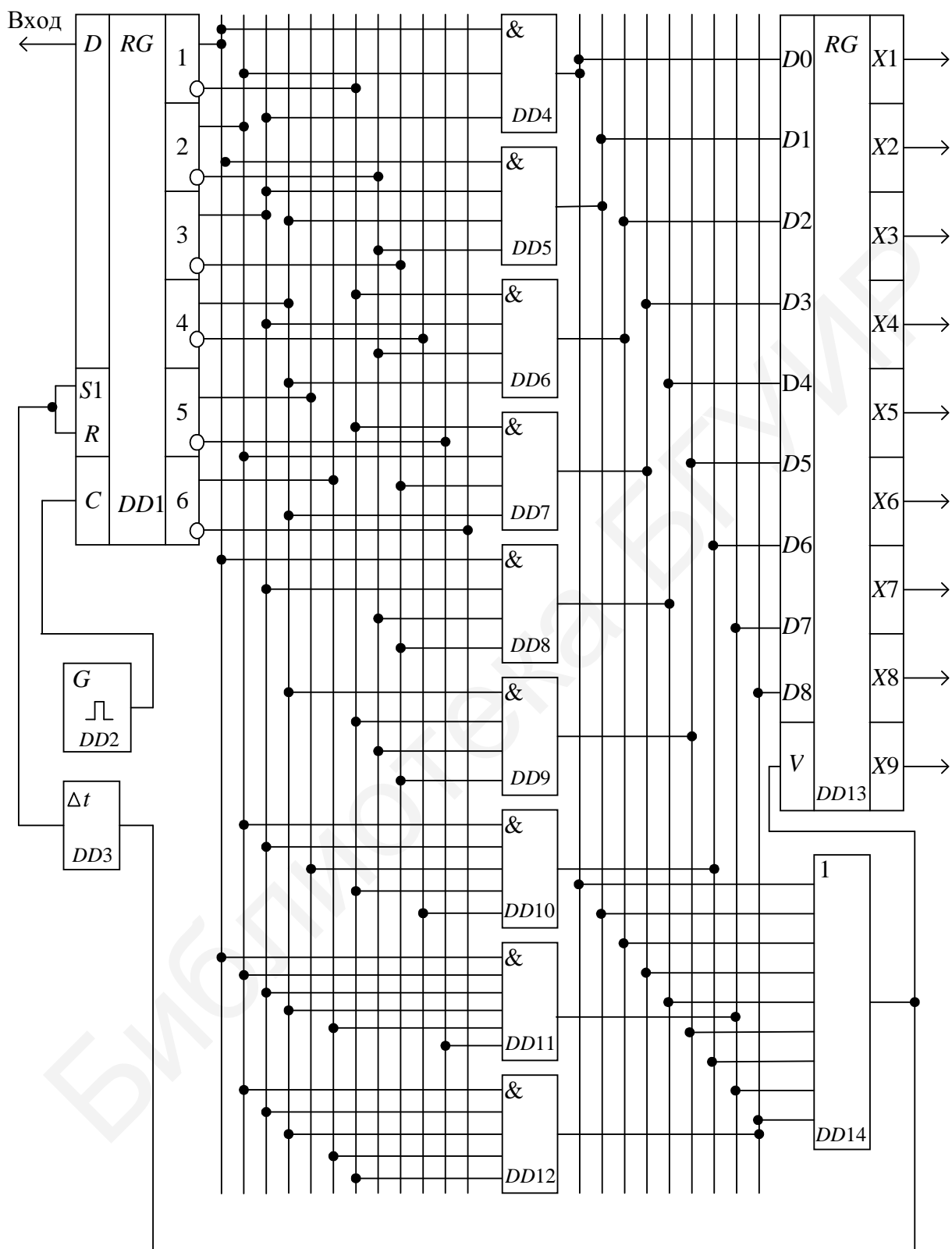


Рисунок 8.23 – Декодирующее устройство эффективных кодов

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.

## 8.9 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

$$F(X, Y) = \prod_{p=1}^n (x_p \oplus \bar{y}_p) \quad (8.3)$$

или

$$F(X, Y) = \bigcup_{p=1}^n (x_p \oplus \bar{y}_p) \quad (8.4)$$

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

На рисунках 8.24 и 8.25 показаны две схемы, реализующие функцию  $F(X, Y)$ , которые построены для  $n = 4$  на основании полученных выражений (8.3) и (8.4) соответственно.

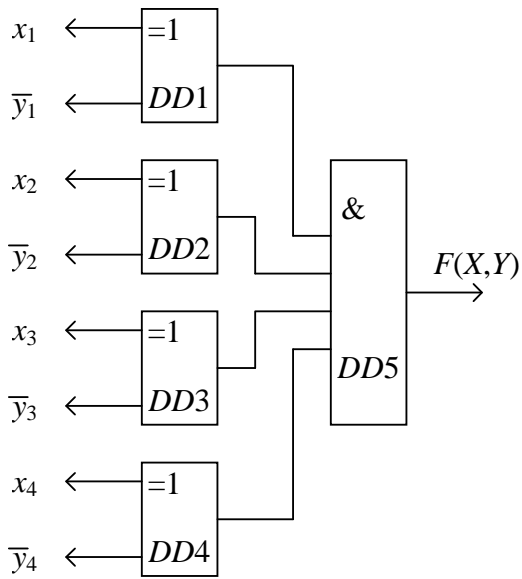


Рисунок 8.24 – Схема, реализующая функцию  $F(X,Y)$  по выражению (8.3)

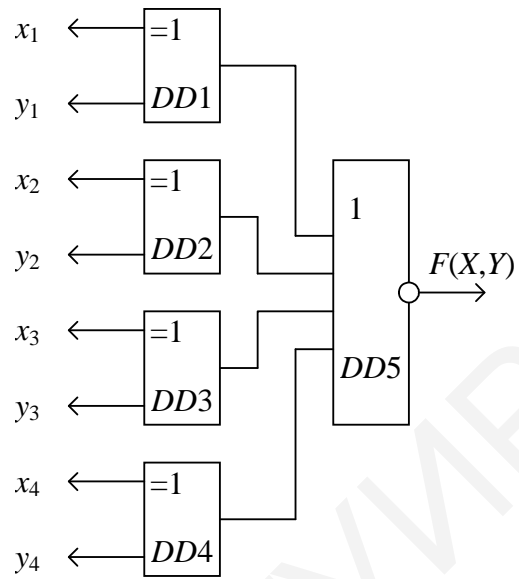


Рисунок 8.25 – Схема, реализующая функцию  $F(X,Y)$  по выражению (8.4)

Схема равнозначности упрощается при использовании сумматоров по модулю 2 с открытым коллектором (рисунок 8.26). В качестве сумматоров по модулю 2 используется исключающее ИЛИ на два входа.

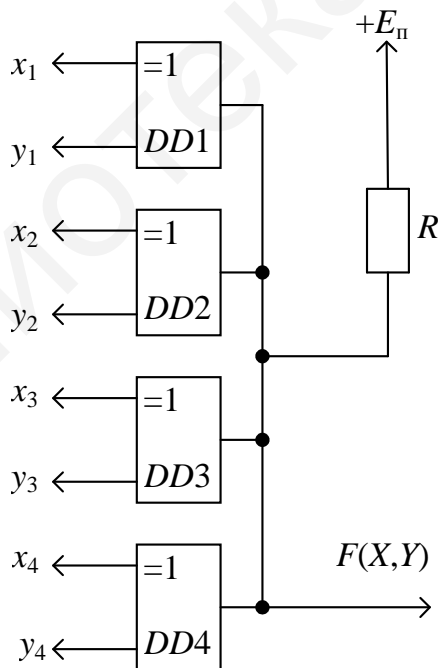
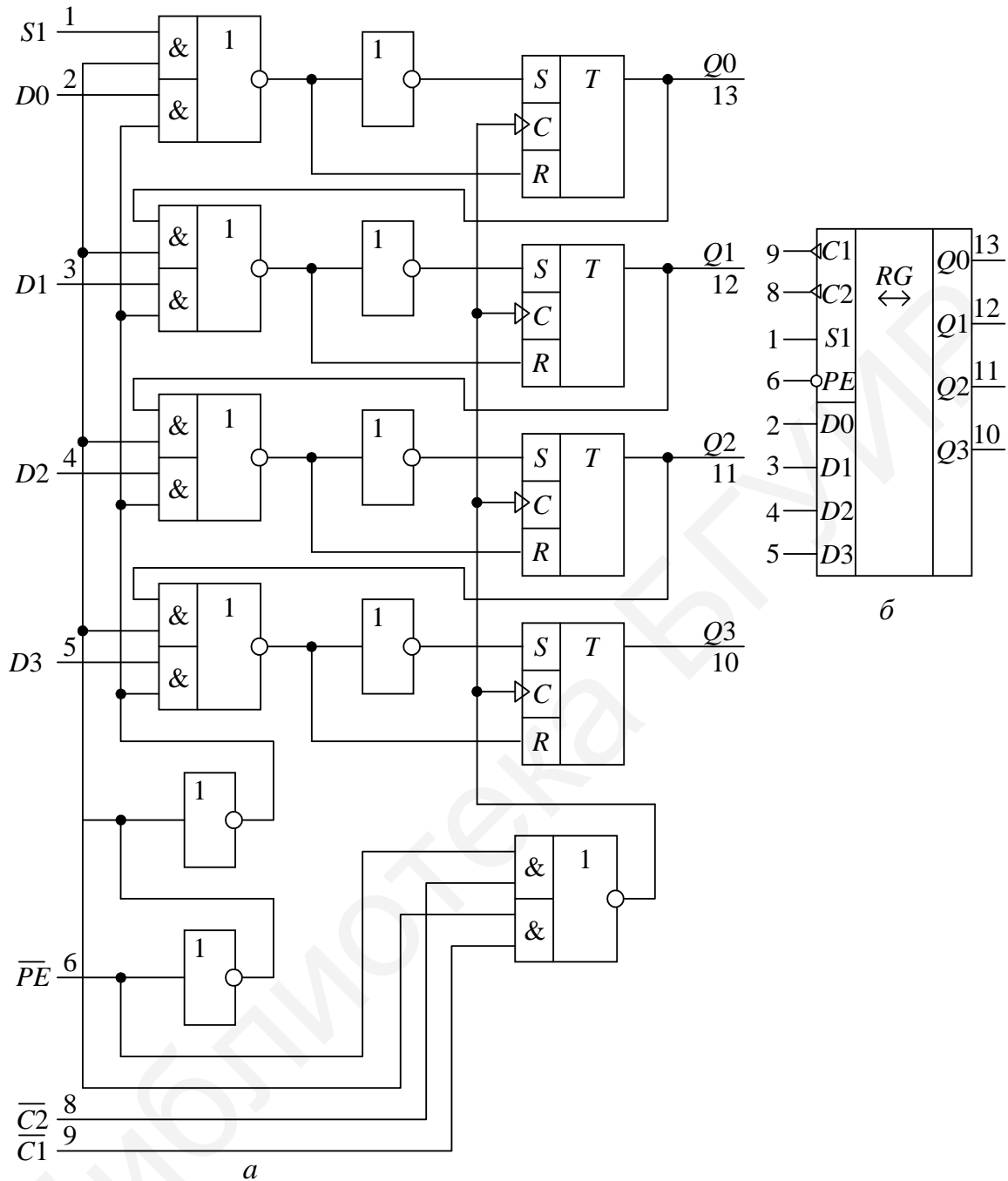


Рисунок 8.26 – Схема равнозначности на сумматорах по модулю 2 с открытым коллектором

Универсальным преобразователем является регистр сдвига. Принцип работы рассмотрим на базе четырехразрядного регистра сдвига ИР1 (рисунок 8.27).



*a* – структура; *б* – условное обозначение

Рисунок 8.27 – Универсальный регистр IP1

Каждый разряд образован синхронным  $RS$ -триггером, включенным по схеме  $D$ -триггера с прямым динамическим входом синхронизации. Он имеет четыре параллельных входа данных  $D0-D3$  (выводы 2–5) и один последовательный вход данных  $S1$  (вывод 1), а также четыре выхода  $Q0-Q3$  (выводы 13–10) от каждого из триггеров. Регистр имеет два тактовых входа  $\bar{C}1$  и  $\bar{C}2$ , 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  $\bar{C}2$ . В момент прихода на вход отрицательного перепада импульса в регистр загружаются данные от параллельных входов  $D0-D3$ . Если на вход  $\overline{PE}$  подать напряжение низкого уровня, то разрешается работа тактовому входу  $\bar{C}1$ . С приходом отрицательного перепада тактового импульса на вход  $\bar{C}1$  данные последовательно сдвигаются от входа  $S1$  на выход  $Q0$ , затем на  $Q1$ ,  $Q2$ ,  $Q3$  (т. е. вправо). Сдвиг данных по регистру влево будет происходить в том случае, если соединить выход  $Q3$  и вход  $D2$ ,  $Q2$  и  $D1$ ,  $Q1$  и  $D0$ .

Такой регистр можно использовать в качестве элемента буферной памяти арифметических устройств, элемента задержки, преобразователя последовательных кодов в параллельные и, 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 КП7 (рисунок 8.28). Данный мультиплексор представляет собой восьмипозиционный переключатель, имеющий три адресных входа  $S0-S2$  с высоким активным уровнем, один стробирующий вход  $\bar{E}$  и восемь информационных входов  $X1-X8$ .

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

$$Y = x_1 \bar{S}_0 \bar{S}_1 \bar{S}_2 + x_2 S_0 \bar{S}_1 S_2 + x_3 \bar{S}_0 S_1 \bar{S}_2 + x_4 S_0 S_1 \bar{S}_2 + x_5 \bar{S}_0 \bar{S}_1 S_2 + x_6 S_0 \bar{S}_1 S_2 + x_7 \bar{S}_0 S_1 S_2 + x_8 S_0 S_1 S_2.$$

На адресные входы управляющие сигналы поступают от двоичного счетчика  $DD2$ . В зависимости от кода на входах  $S0-S2$  к выходу  $Y$  подключается соответствующий вход  $x_i$  (таблица 8.12). 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 рисунке 8.28). Для обслуживания устройств с большим числом выходов производится параллельное подключение мультиплексоров.

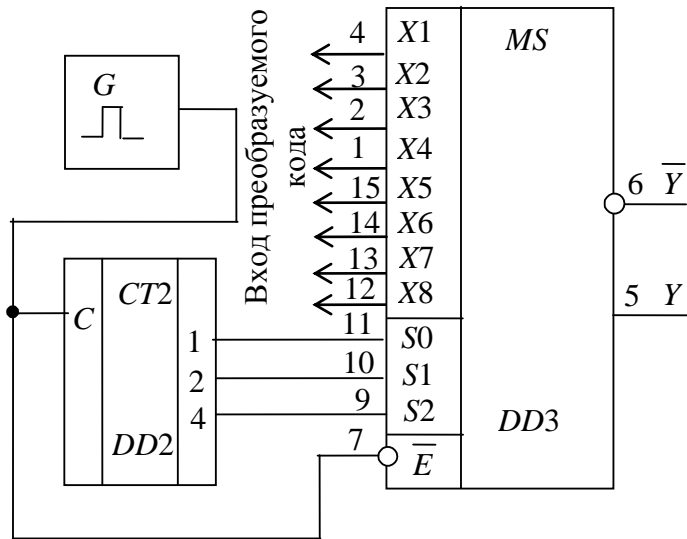


Рисунок 8.28 – Преобразователь параллельного кода в последовательный

Таблица 8.12 – Состояния мультиплексора КП7

Выбор			Выходы		
S2	S1	S0	E	Y	$\bar{Y}$
x	x	x	1	0	1
0	0	0	0	X1	X1
0	0	1	0	X2	X2
0	1	0	0	X3	X3
0	1	1	0	X4	X4
1	0	0	0	X5	X5
1	0	1	0	X6	X6
1	1	0	0	X7	X7
1	1	1	0	X8	X8

На рисунке 8.29 показана схема преобразователя на 16 входов, а на рисунке 8.30 – на 64 входа.

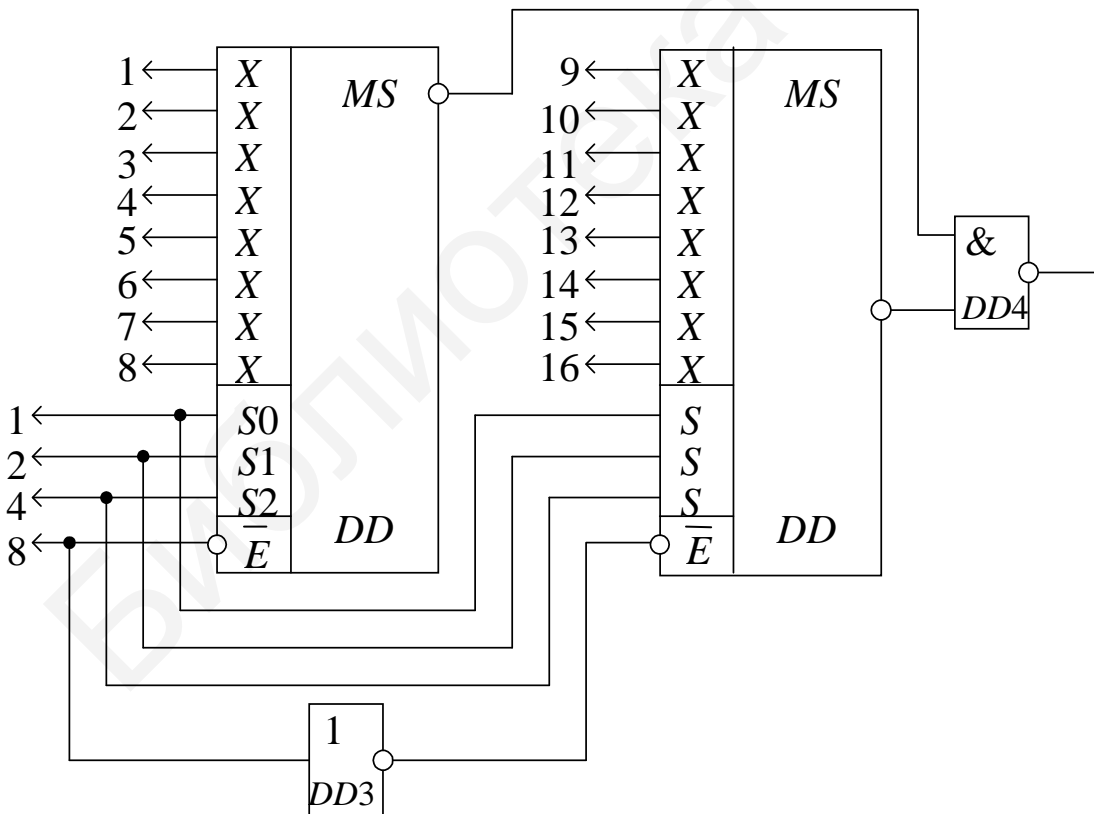


Рисунок 8.29 – Схема преобразователя на 16 входов

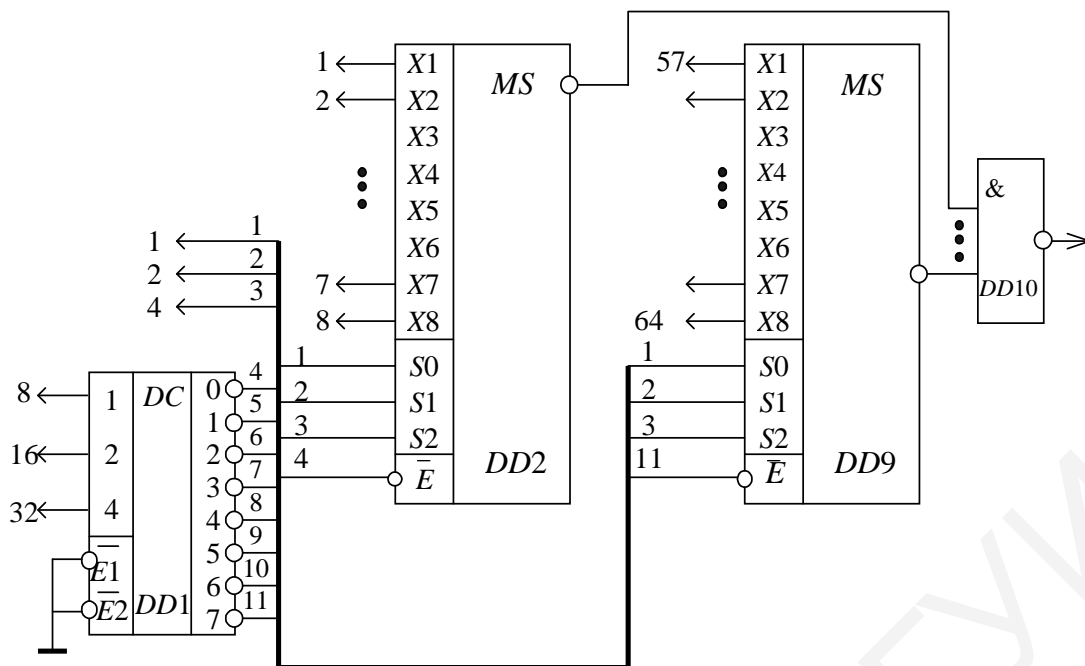


Рисунок 8.30 – Схема преобразователя на 64 входа

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 (см. рисунок 8.29) или восемь входов (см. рисунок 8.30). На какой коммутатор –  $DD_1$ ,  $DD_2$  (см. рисунок 8.29) или  $DD_2$ – $DD_9$  (см. рисунок 8.30) – поступает адрес от двоичного счетчика, зависит от сигнала на входе стробирования  $\bar{E}$ . Для преобразователя на 64 выхода сигнал стробирования формируется дешифратором  $DD_1$  (рисунок 8.30) при подаче на его вход кода с весами разрядов 8-16-32.

## 9 ТЕХНИЧЕСКИЕ СРЕДСТВА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ КОРРЕКТИРУЮЩИХ КОДОВ

### 9.1 Кодер и декодер кода с защитой на четность

Функциональная схема такого кодера для четырехразрядного кода приведена на рисунке 9.1.

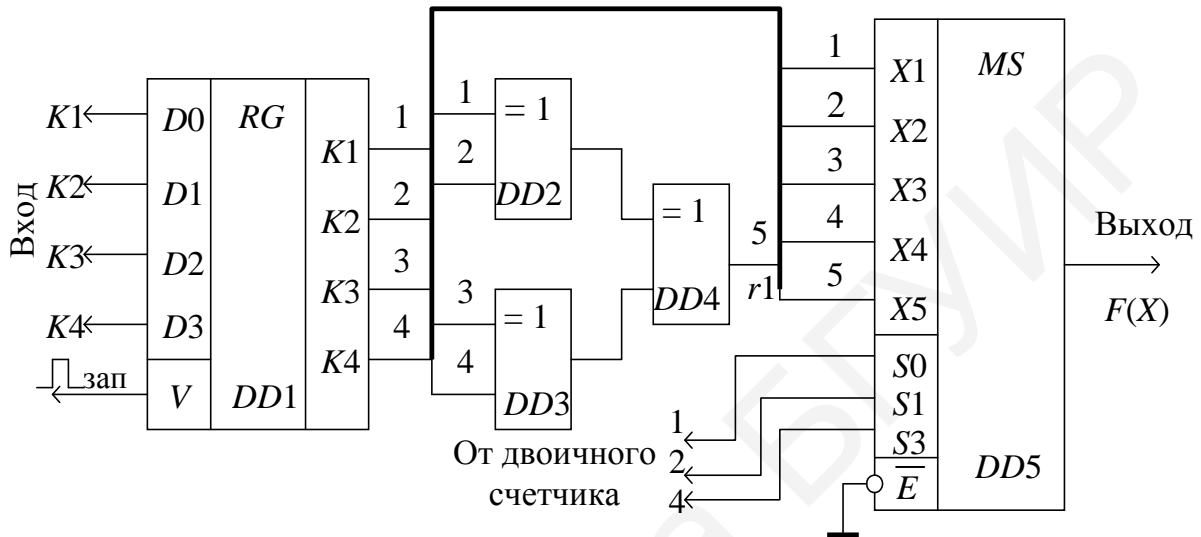


Рисунок 9.1 – Кодер с защитой на четность

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  $S0S1S3$  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 рисунке 9.2.

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



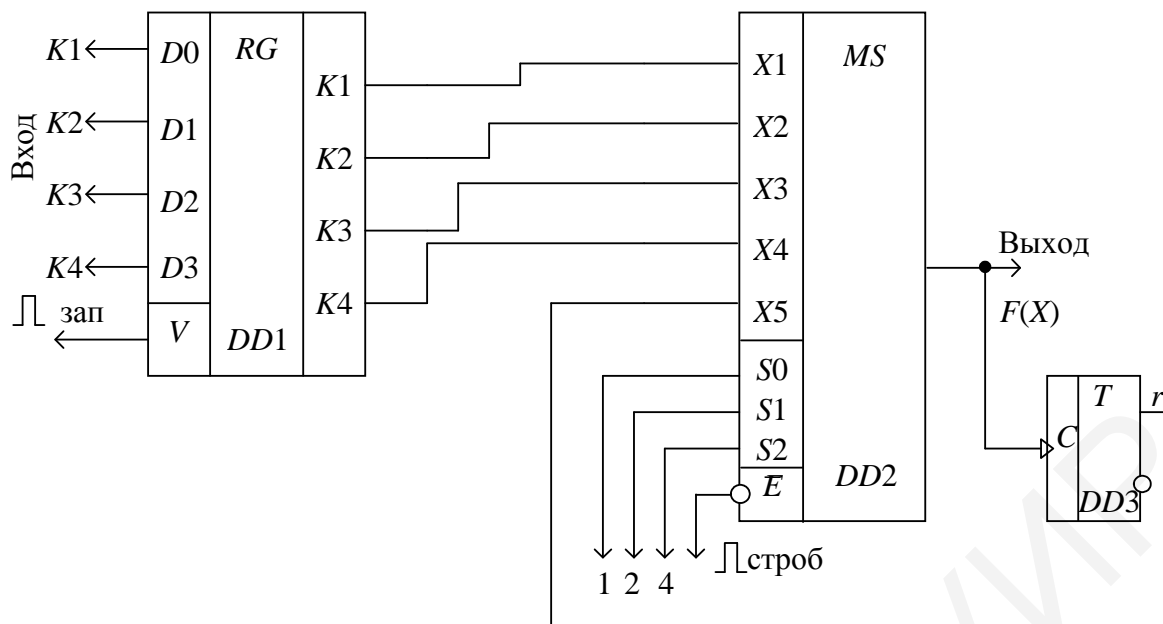


Рисунок 9.2 – Схема формирования контрольного символа с помощью триггера

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 пять разрядов приведена на рисунке 9.3.

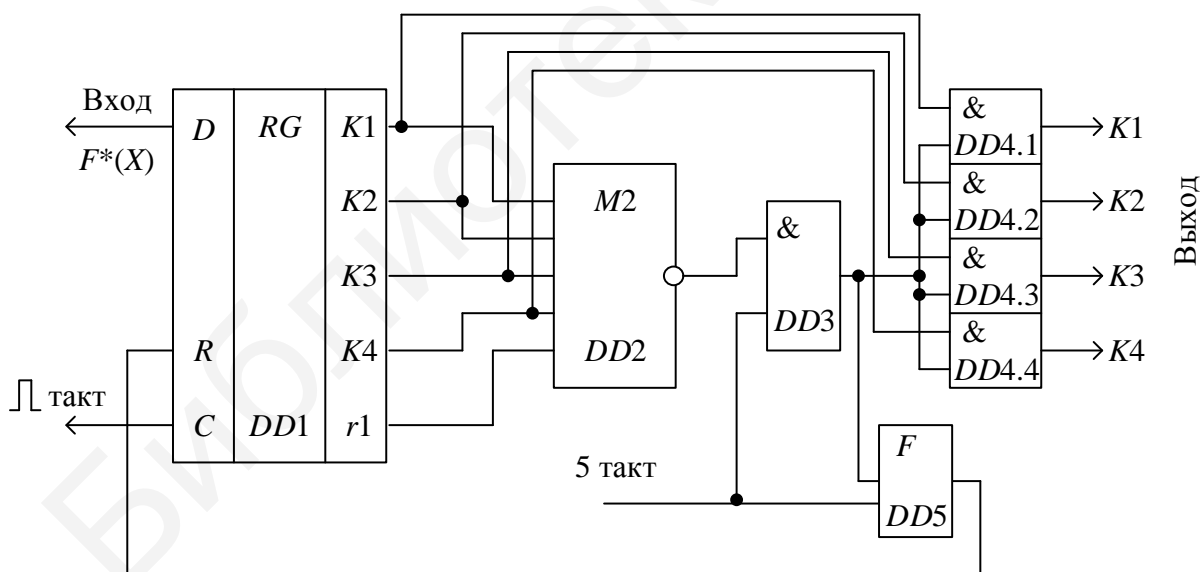


Рисунок 9.3 – Схема декодера кода с защитой на четность

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 открывает схемы И DD4.1–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 исключаяющее ИЛИ, или можно использовать схемы контроля четности ИП2 (рисунок 9.4), состояние входов и выходов которой приведены в таблице 9.1.

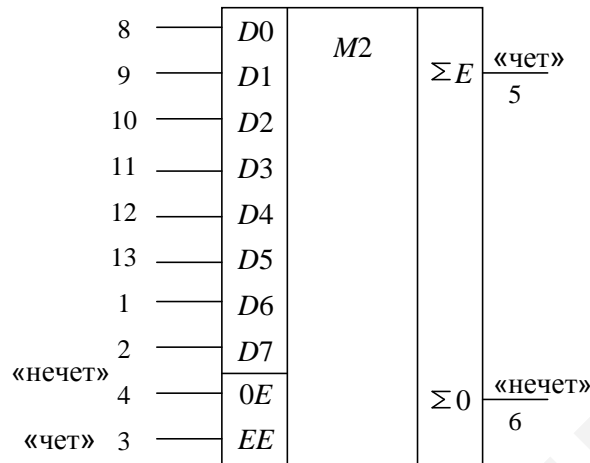


Рисунок 9.4 – Условное обозначение ИМС ИП2

Таблица 9.1 – Состояния ИП

Сумма единиц на входах $D0-D7$	Входы		Выходы	
	$EE$	$0E$	$\Sigma E$	$\Sigma 0$
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

## 9.2 Кодер и декодер кода с постоянным весом

Кодер с постоянным весом применяется в системах ТУ. Принцип построения этого кода и корректирующие возможности изложены в пункте 7.2.1. Рассмотрим кодер для кода  $C_4^2$ . С его помощью можно передать шесть команд (7.6). Поставим в соответствие каждой команде (сообщению) свою кодовую комбинацию:

$$\begin{aligned}
 x_1 = 0011 &= \bar{y}_4 \bar{y}_3 y_2 y_1, & x_2 = 0101 &= \bar{y}_4 y_3 \bar{y}_2 y_1, \\
 x_3 = 1001 &= y_4 \bar{y}_3 \bar{y}_2 y_1, & x_4 = 0110 &= \bar{y}_4 y_3 y_2 \bar{y}_1, \\
 x_5 = 1010 &= y_4 \bar{y}_3 x_2 \bar{x}_1, & x_6 = 1100 &= y_4 y_3 \bar{y}_2 \bar{y}_1.
 \end{aligned} \tag{9.1}$$

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

$$\begin{aligned}
 y_1 &= x_1 + x_2 + x_3, & y_2 &= x_1 + x_4 + x_5, \\
 y_3 &= x_2 + x_4 + x_6, & y_4 &= x_3 + x_5 + x_6.
 \end{aligned}
 \tag{9.2}$$

Кодер, построенный в соответствии с приведенными выражениями, приведен на рисунке 9.5.

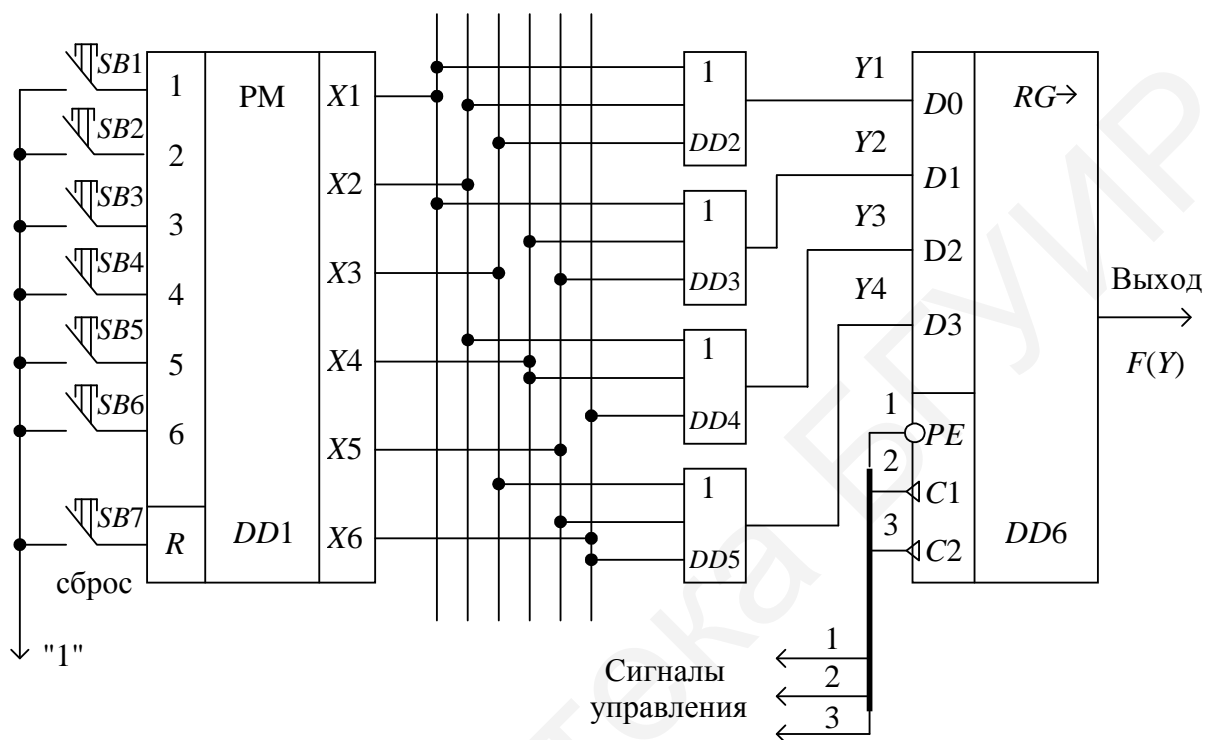


Рисунок 9.5 – Кодер кода  $C_4^2$

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 (9.2). 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 рисунке 9.6.

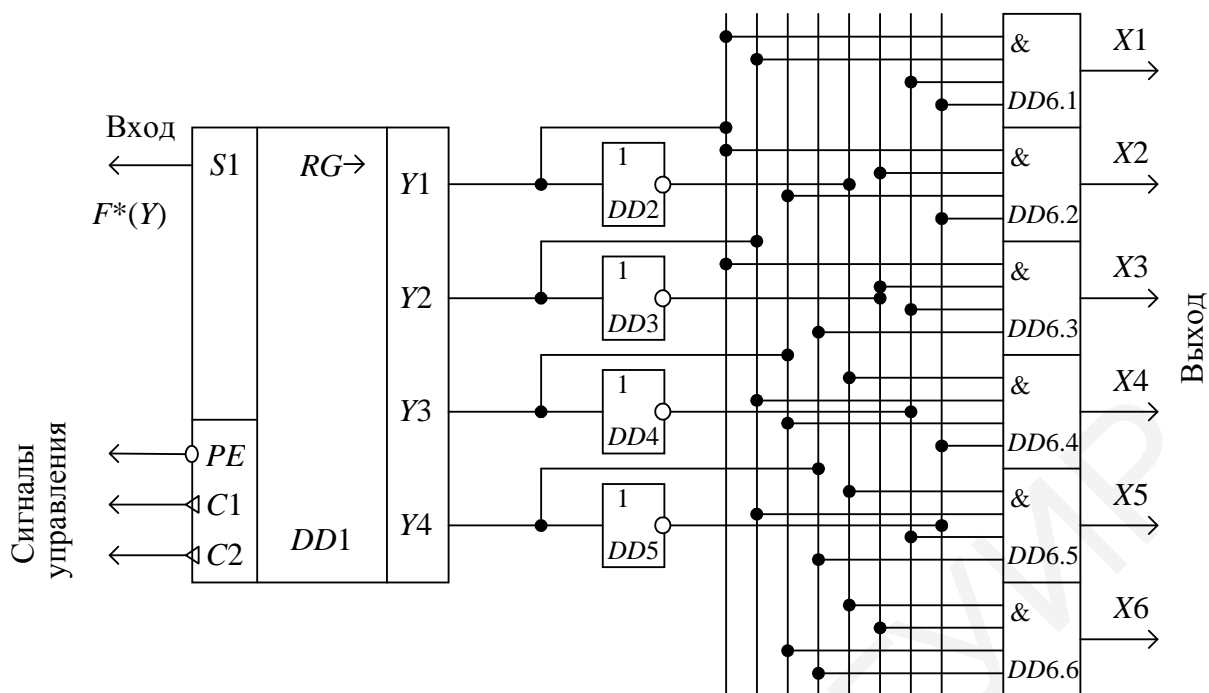


Рисунок 9.6 – Декодер кода  $C_4^2$

Кодовая комбинация  $F^*(Y)$  из линии связи поступает в приемный регистр  $DD1$  и далее на собственно сам дешифратор, собранный на элементах И  $DD6.1-DD6.6$ , входы которых заведены в соответствии с выражениями (9.1). На выходе схем И  $DD6.1-DD6.6$  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.

### 9.3 Кодер и декодер кода с двумя проверками на четность

Принцип образования кодовых комбинаций в данном коде описан в пункте 7.2.5.

Кодирующее устройство для  $k = 6$  показано на рисунке 9.7. 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 все информационные разряды, а схемой  $DD3$  формируется второй контрольный разряд  $r_2$ , дополняющий до четности все нечетные информационные разряды, т. е.  $k_1, k_3$  и  $k_5$ .

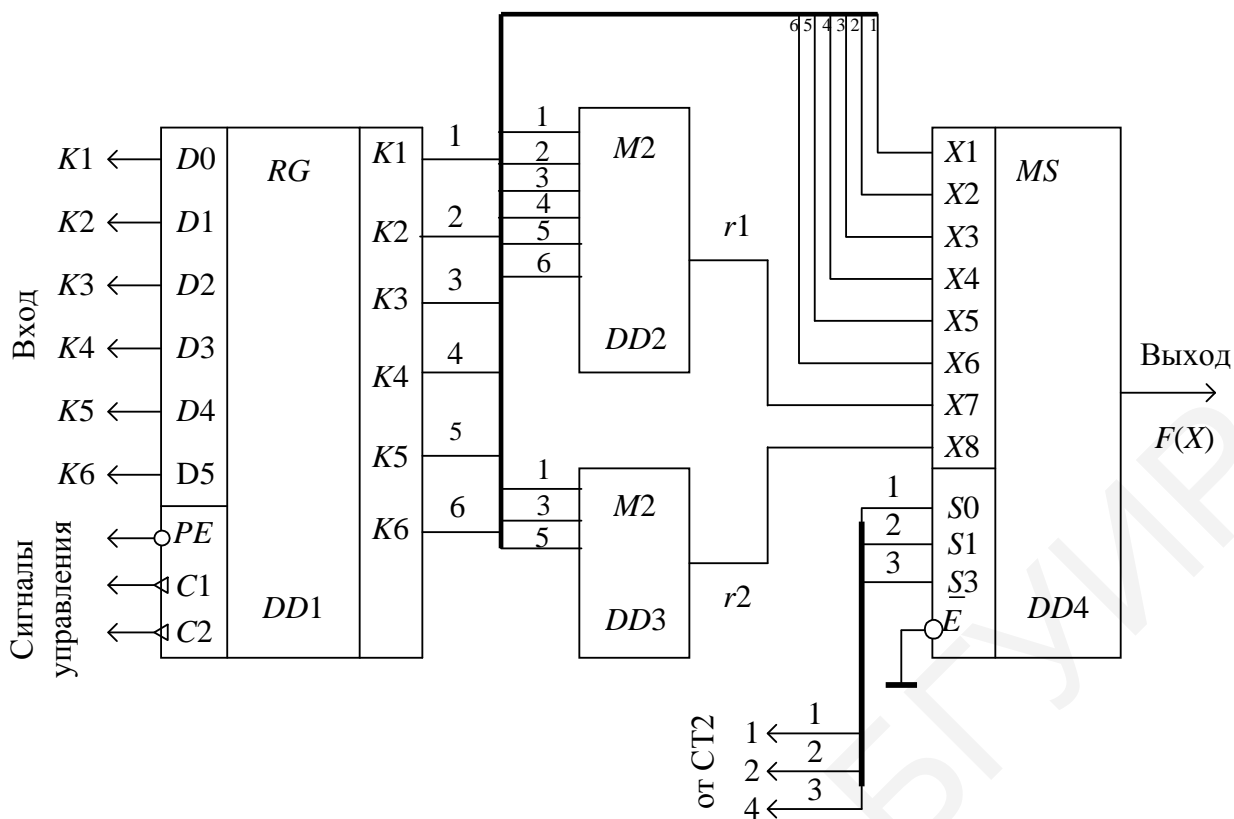


Рисунок 9.7 – Кодер кода с двумя проверками на четность

Декодирующее устройство рассматриваемого кода приведено на рисунке 9.8. 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.

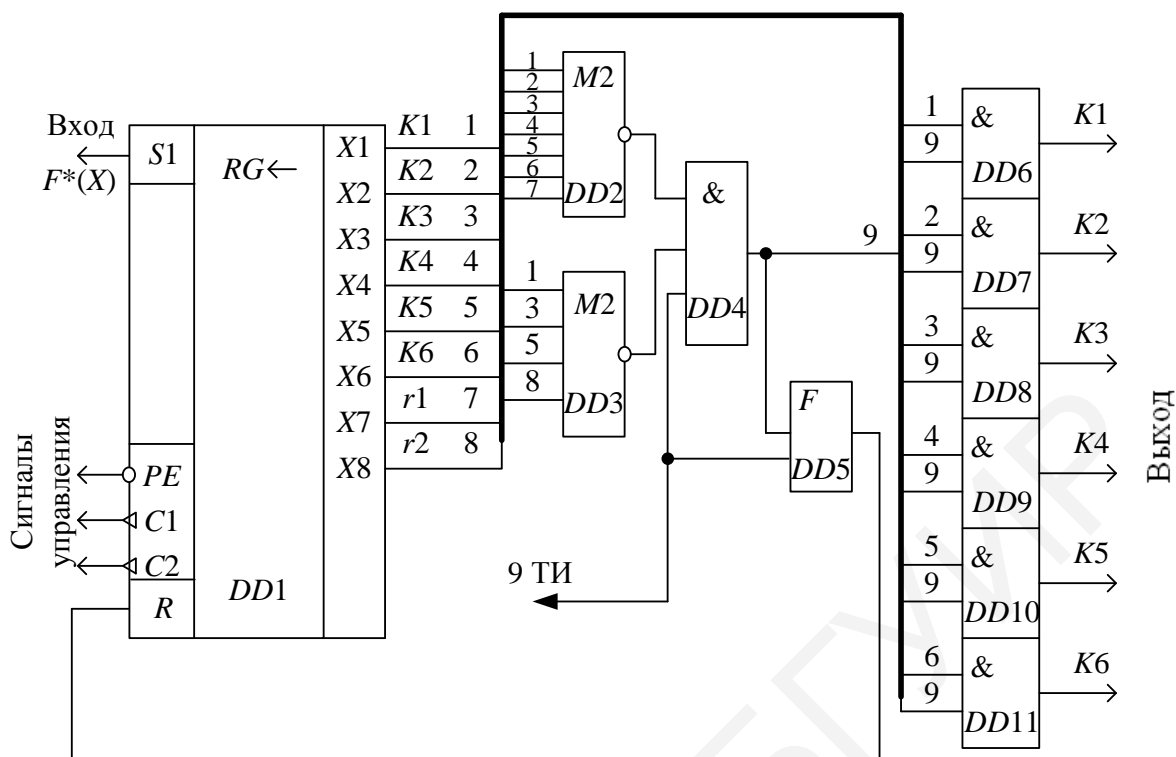


Рисунок 9.8 – Декодирующее устройство с двумя проверками на четность

#### 9.4 Кодер и декодер кода с повторением

Как известно из пункта 7.2.6, 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 рисунках 9.9 и 9.10 соответственно.

Кодер, изображенный на рисунке 9.9, 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.

Например, если на вход поступило сообщение  $G(x) = 1101 = k_4 k_3 k_2 k_1$ , то в результате трехкратной передачи в линию связи поступит кодовая комбинация  $G(x) = 1111000011111100$ .

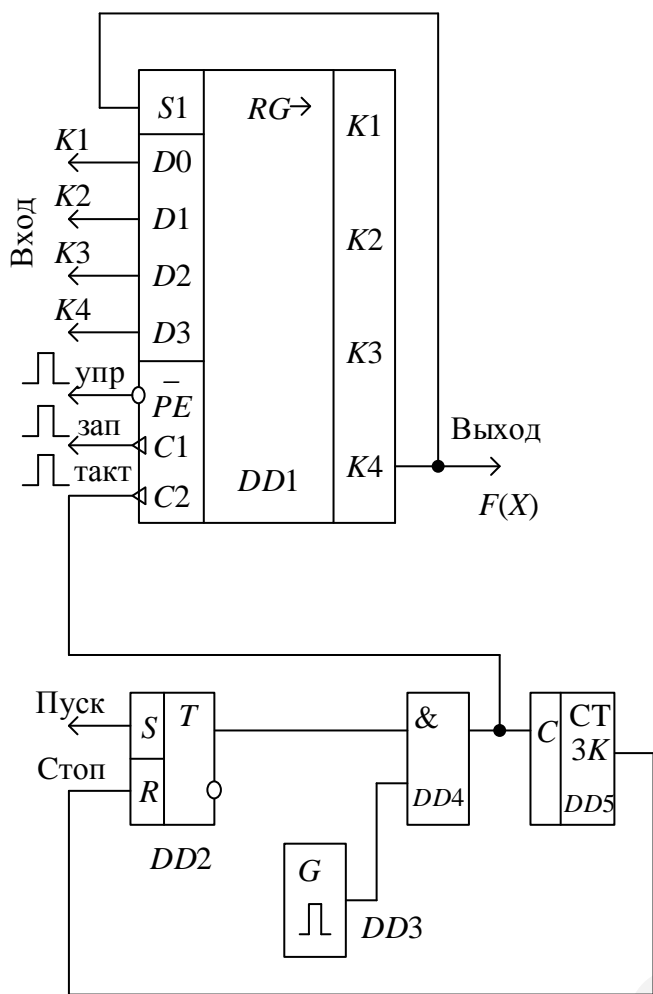


Рисунок 9.9 – Кодер с трехкратным повторением исходной комбинации

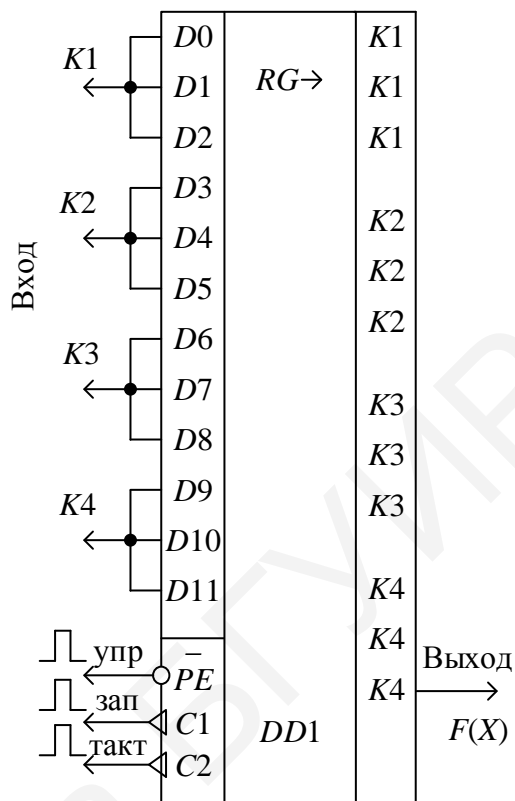


Рисунок 9.10 – Кодер с трехкратным повторением каждого элемента исходной комбинации

В кодере, изображенном на рисунке 9.10, каждый символ  $k_i$  исходной комбинации записывается в три рядом стоящие 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 подлежало кодированию сообщение  $G(x) = k_4k_3k_2k_1 = 1001$ , то в линию связи поступит кодовая комбинация  $F(x) = 111000000111$ .

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 пункте 7.2.6, наибольшее применение нашел код с повторением комбинаций как обеспечивающий более высокую помехоустойчивость. Поэтому де-

кодирующее устройство рассмотрим для этого случая, схема которого приведена на рисунке 9.11 для  $k = 4$ ,  $m = 3$  и  $n = 12$ .

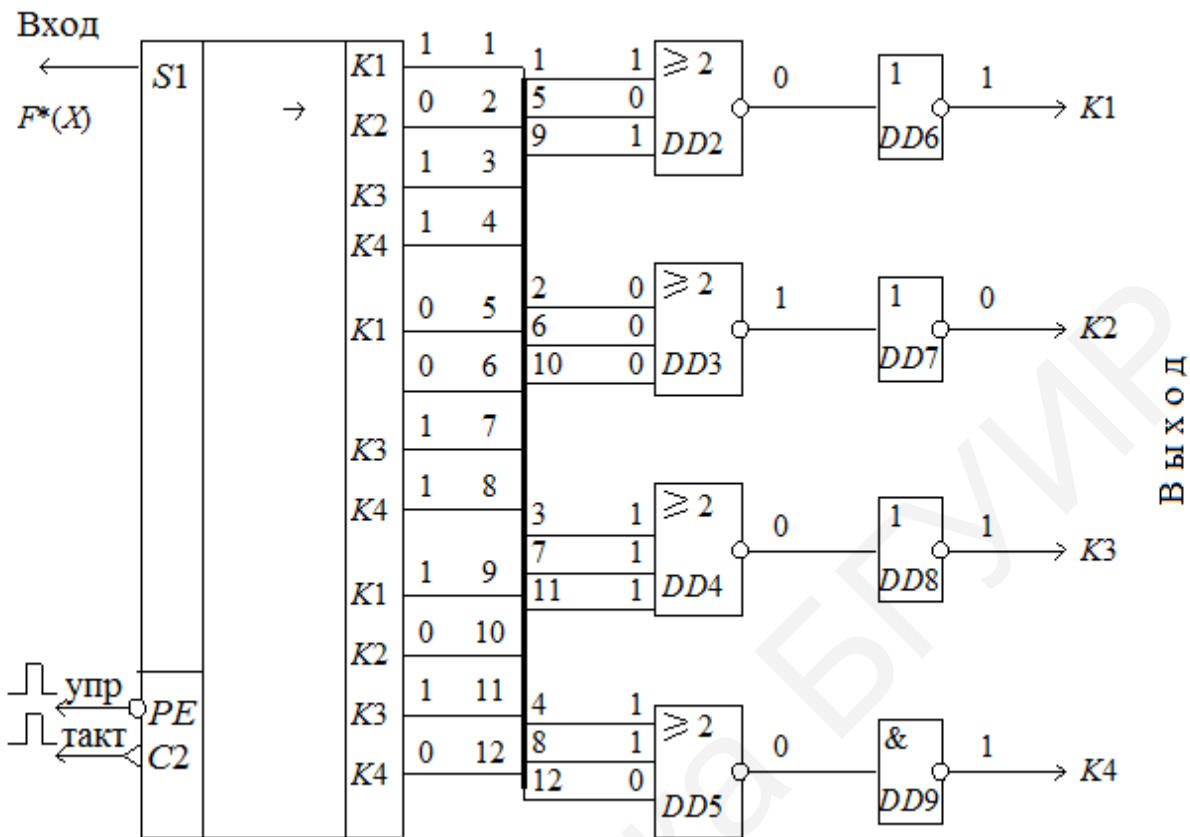


Рисунок 9.11 – Декодер кода с трехкратным повторением исходной комбинации

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.

Трехвходовые мажоритарные элементы  $DD2-DD5$  с инверсным выходом выполняют в общем виде функцию  $y = x_1x_2 \cup x_1x_3 \cup x_2x_3$ . 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 рисунке 9.11.



Как видно из рисунка 9.11, в результате принятия решения элементами  $DD2-DD5$  исходное сообщение имеет вид  $G(x) = k_4k_3k_2k_1 = 1001$ , что соответствует информационной части  $F(x)$ . В заключение следует указать, что построение мажоритарных элементов на число входов больше пяти целесообразно на двоичных сумматорах, например, ИМ3, ИМ2.

### 9.5 Кодер и декодер кода с числом единиц, кратным трем

Как указано в пункте 7.2.7, 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. Кодировальное устройство для  $k=5$  приведено на рисунке 9.12.

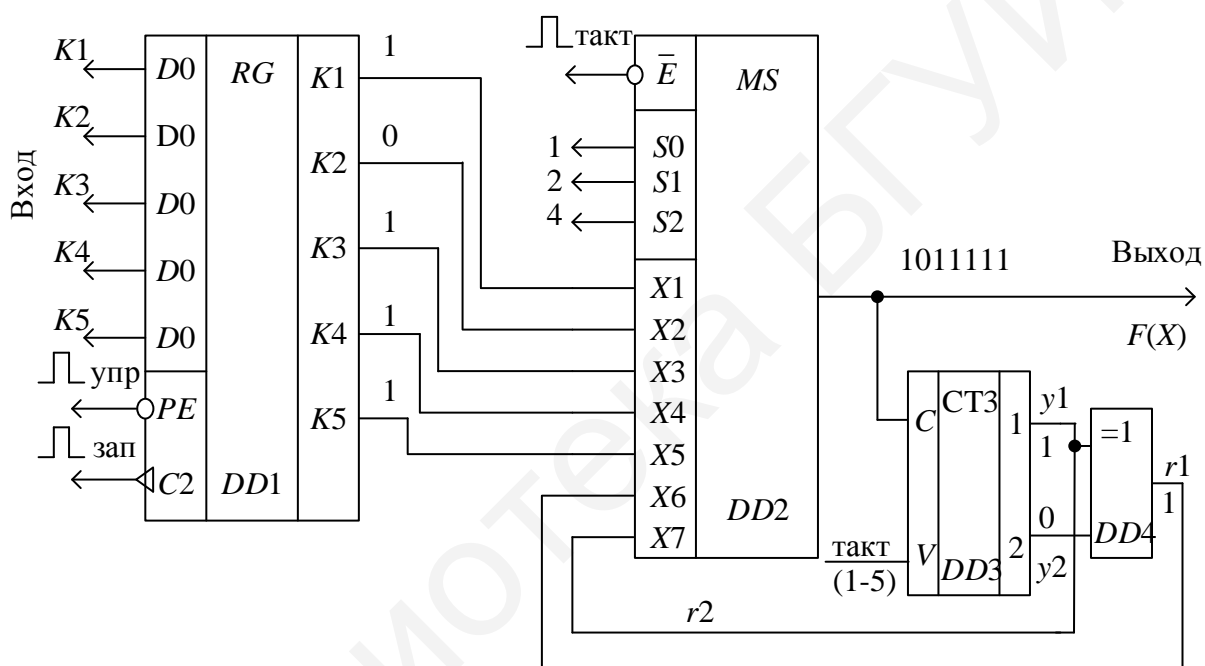


Рисунок 9.12 – Кодер кода с числом единиц, кратным трем

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:  $r_1r_2 = 00$ ,  $r_1r_2 = 11$ ,  $r_1r_2 = 10$ . Таким образом, формирователь контрольных символов на элементах  $DD3$  и  $DD4$  описывается выражением

$$\overline{r_1} = \overline{y_2}y_1 + y_2\overline{y_1}, r_2 = y_1.$$

Мультиплексор  $DD2$  осуществляет преобразование параллельного кода в последовательный. Процесс кодирования сообщения  $G(x) = k_1k_2k_3k_4k_5 = 10111$  показан на рисунке 9.12.

Основой декодера (рисунок 9.13) является счетчик  $DD2$  с коэффициентом деления 3.

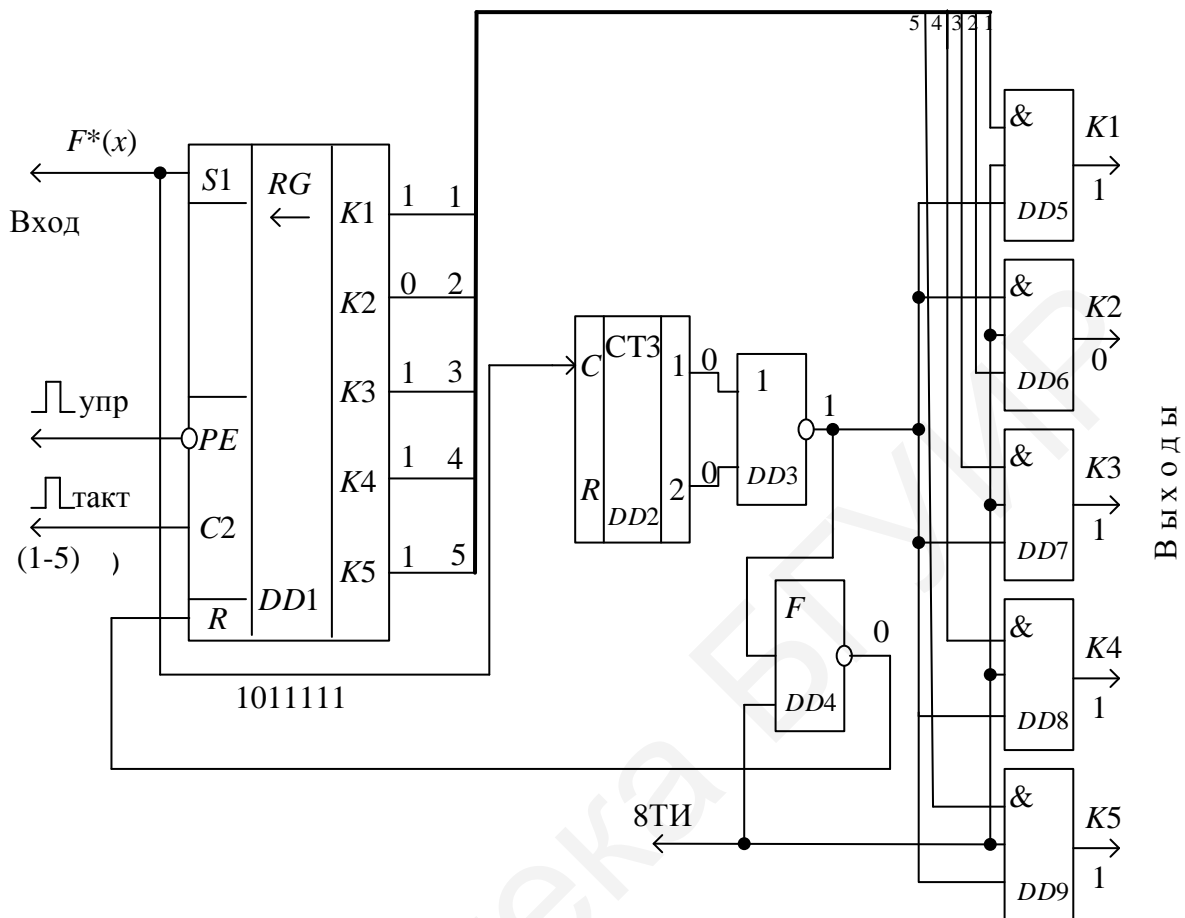


Рисунок 9.13 – Схема декодера кода с числом единиц, кратным трем

На первых пяти тактах информационные символы заносятся в регистр  $DD1$ , а полная кодовая комбинация  $F^*(x)$  на тактах 1–7 поступает в счетчик  $DD2$ . Если в кодовой комбинации  $F^*(x)$  искажений нет, то после седьмого 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  $DD5–DD9$ , а 1 на выходе формирователя  $DD4$  сбросит регистр  $DD1$  в исходное положение. Процесс декодирования кодовой комбинации  $F^*(x) = k_1 k_2 k_3 k_4 r_1 r_2 = 1011111$  показан на рисунке 9.13.

### 9.6 Кодер и декодер инверсного кода

Теоретические вопросы построения инверсного кода рассмотрены в пункте 7.2.8, а функциональная схема кодера для четырехразрядных сообщений приведена на рисунке 9.14.

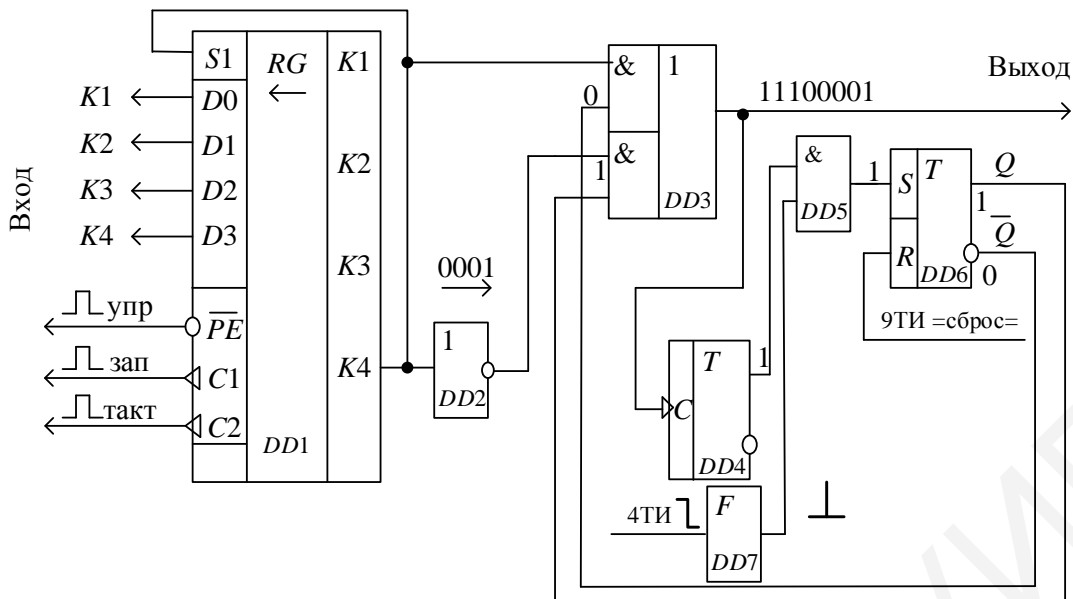


Рисунок 9.14 – Кодер инверсного кода для  $k = 4$

Подлежащее кодированию сообщение записывается в кольцевой регистр  $DD1$ , а затем на первых четырех тактах через верхний по схеме 2И-ИЛИ элемент  $DD3$  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  $DD3$  сигналом  $\bar{Q}$  триггера  $DD6$ , и контрольные символы в неизменном виде будут повторять информационные, которые на предыдущих четырех тактах поступают в линию связи. Если в информационной части было нечетное 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 повторят информационные в инверсном виде. После передачи всей кодовой комбинации схема устанавливается в исходное положение девятым тактовым импульсом и готова к кодированию следующего полезного сообщения. Состояние элементов схемы при передаче сообщения  $G(x) = k_4k_3k_2k_1 = 1110$  указано на схеме (см. рисунок 9.14).

Функциональная схема декодера восьмиразрядных кодовых комбинаций приведена на рисунке 9.15.

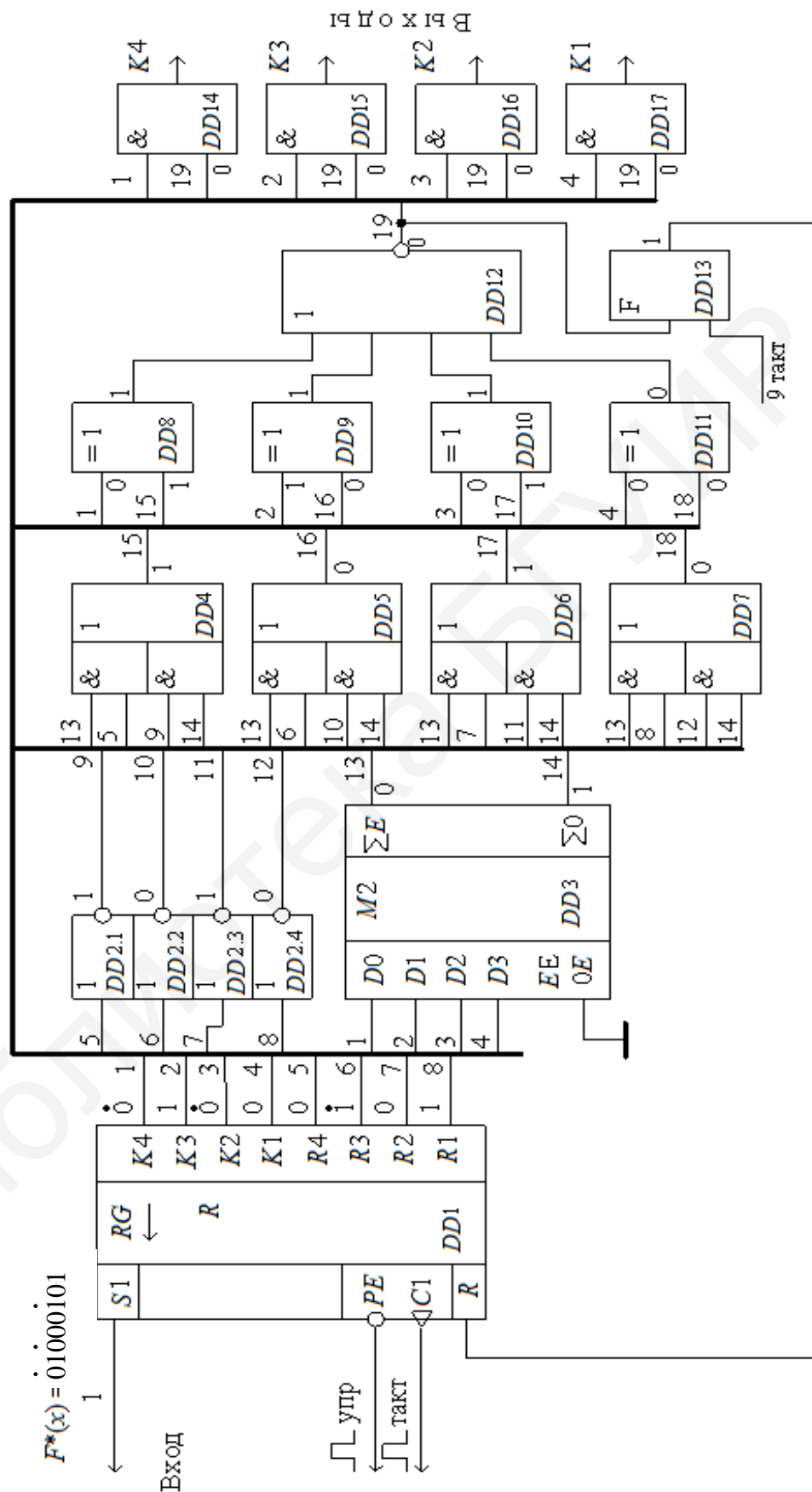


Рисунок 9.15 – Декодер инверсного кода

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 открывает 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  $F^*(x) = 01000101$  показан на рисунке 9.15 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  $DD1$ .

## 9.7 Кодер и декодер корреляционного кода

Как показано в пункте 7.2.9, 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 рисунке 9.16.

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  $S0, S1, S2$  от двоичного счетчика он поочередно к выходу подключает входы  $D0$ – $D7$ . А так как попеременно чередуются их прямые и инверсные сигналы, то на выходе получаем корреляционный код. Процесс преобразования сообщения  $G(x) = k_4k_3k_2k_1 = 1100$  показан на схеме в виде состояния элементов. На выходе мультиплексора получаем кодовую комбинацию  $F(x) = 10100101$ .

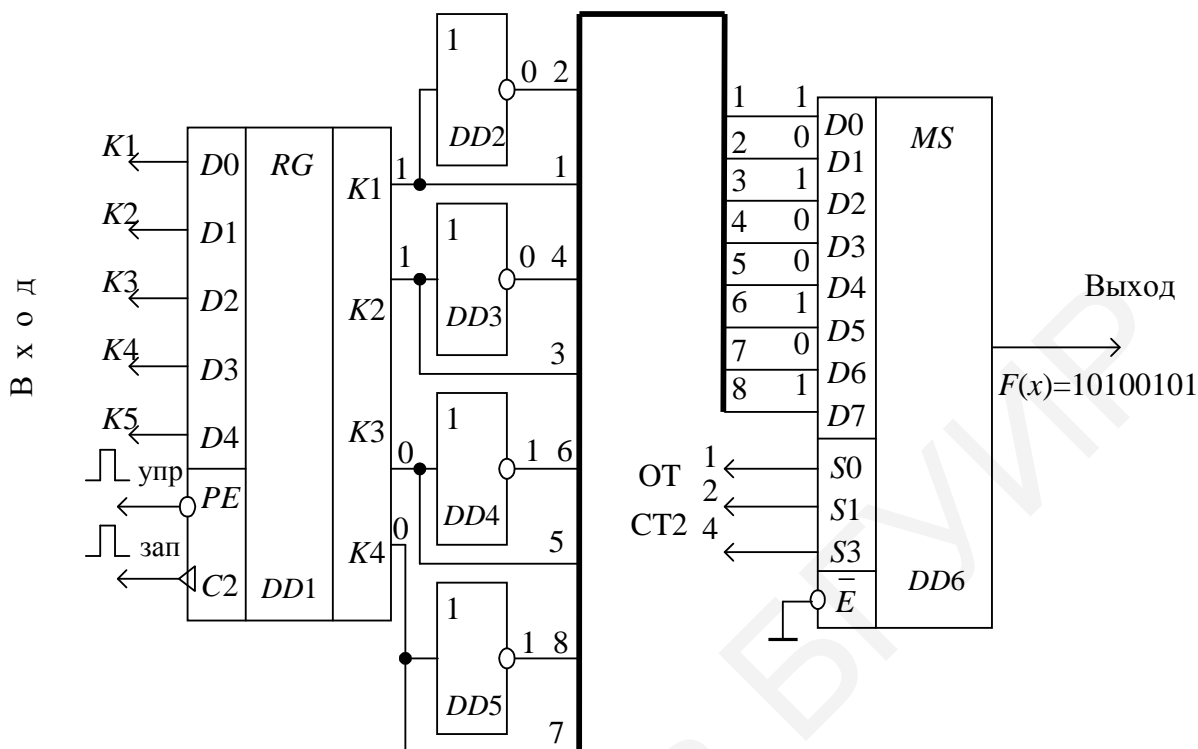


Рисунок 9.16 – Кодер корреляционного кода

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 рисунке 9.17.

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

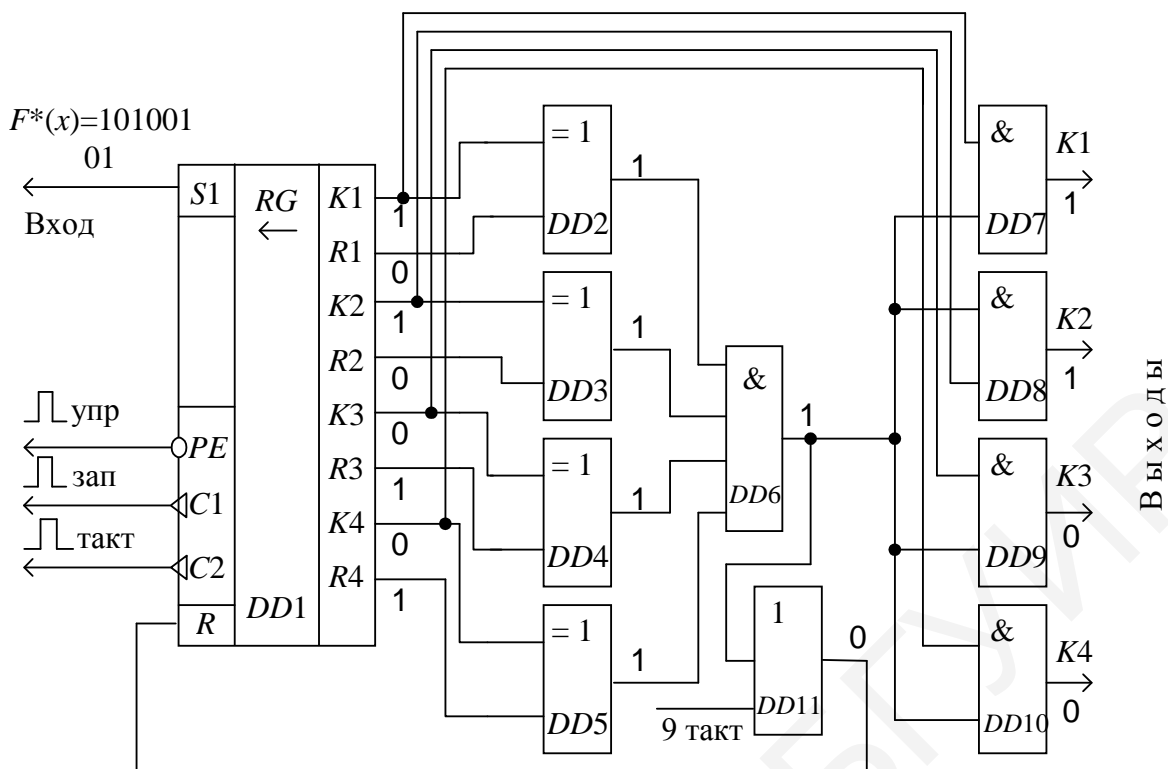


Рисунок 9.17 – Декодер корреляционного кода

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. рисунок 9.16).

## 9.8 Кодер и декодер кода Бергера

Функциональная схема кодера приведена на рисунке 9.18.

В состав кодирующего устройства входят: входной регистр  $DD1$ , предназначенный для хранения преобразуемых сообщений, счетчик  $DD5$  для подсчета числа единиц в исходном сообщении и преобразователь  $DD2$  параллельного кода в последовательный.

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, схе-

ма 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 рисунке 9.18.

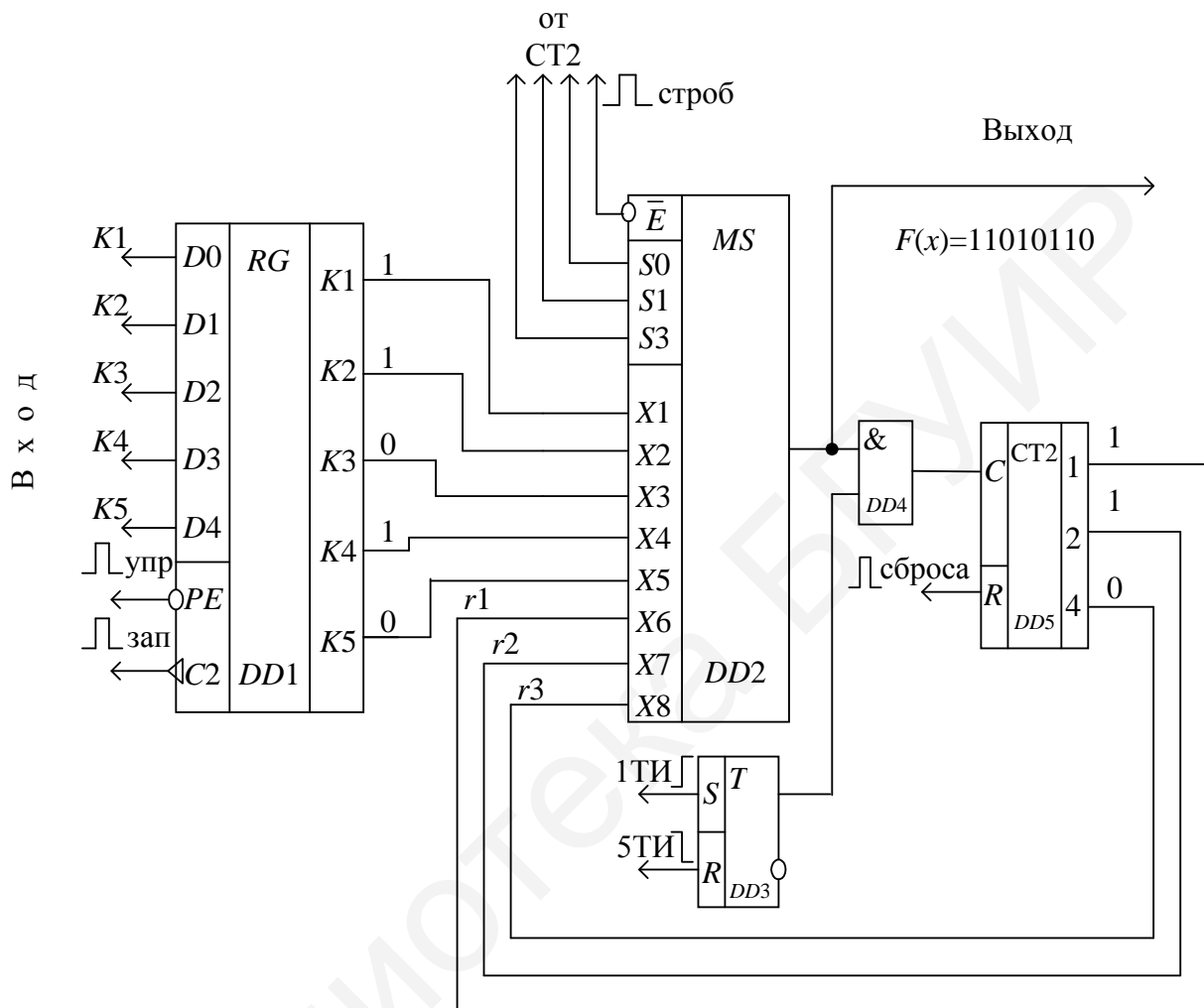


Рисунок 9.18 – Кодер кода Бергера для  $k = 5$

Декодирование сводится к определению числа единиц в информационной части, т. е. к формированию контрольных символов из пришедших на приемную сторону информационных символов с дальнейшим 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 рисунке 9.19.



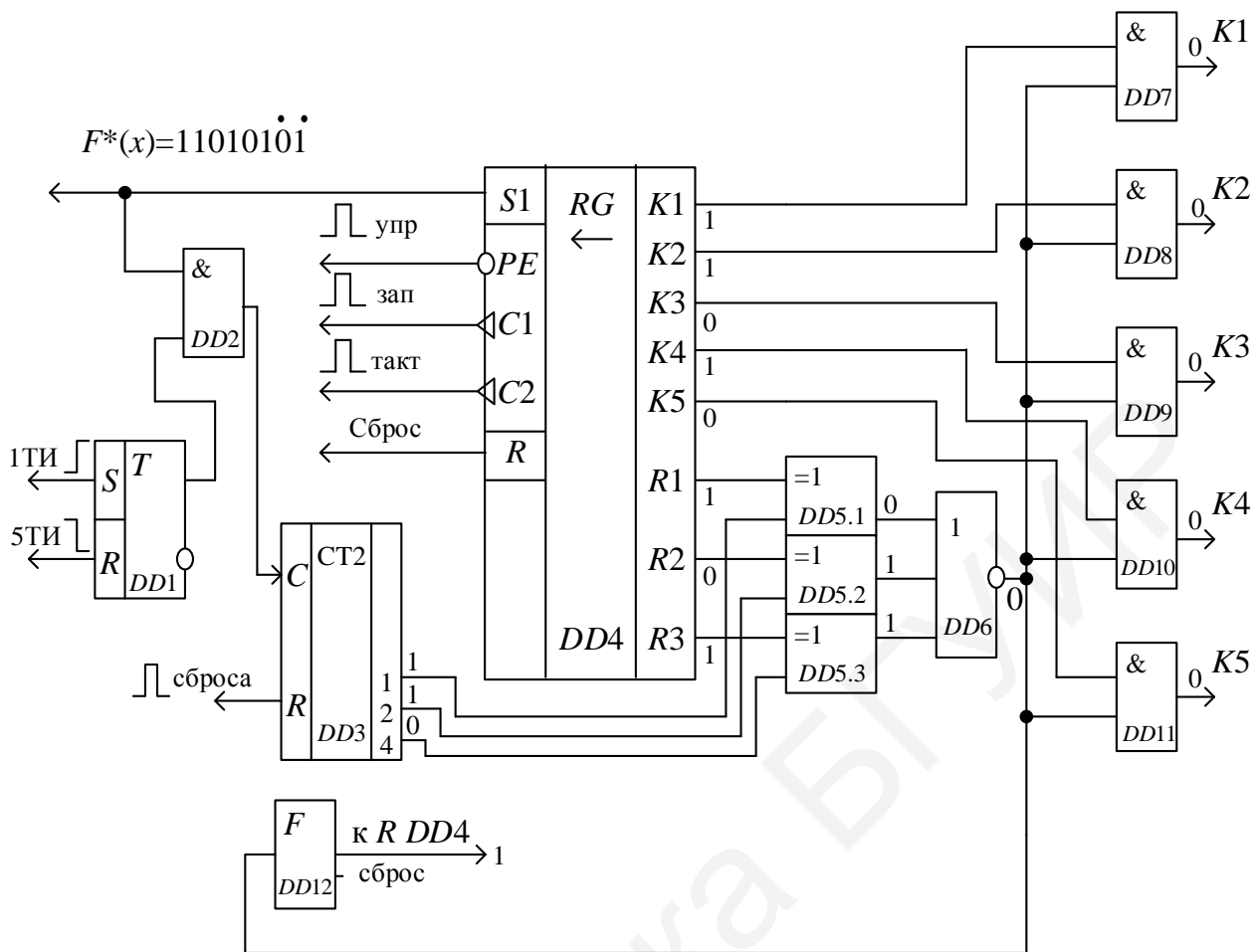


Рисунок 9.19 – Декодер кода Бергера

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  $DD5.1$ – $DD5.3$  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  $DD7$ – $DD11$ , 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  $F^*(x) = 11010101$  показан на схеме в виде состояния элементов. Элементами  $DD5.1$ – $DD5.3$  зафиксировано несовпадение двух последовательностей, в результате чего комбинация бракуется.

## 9.9 Кодирование и декодирующее устройства систематического кода

Как следует из пункта 7.3.1, алгоритм кодирования и декодирования определяется составом образующей и проверочной матриц. Схема кодера, использующего алгоритм, полученный по выражениям (7.12)–(7.14) ( $a_5 = a_2 \oplus a_3 \oplus a_4$ ,  $a_6 = a_1 \oplus a_2 \oplus a_4$ ,  $a_7 = a_1 \oplus a_3 \oplus a_4$ ), приведена на рисунке 9.20.

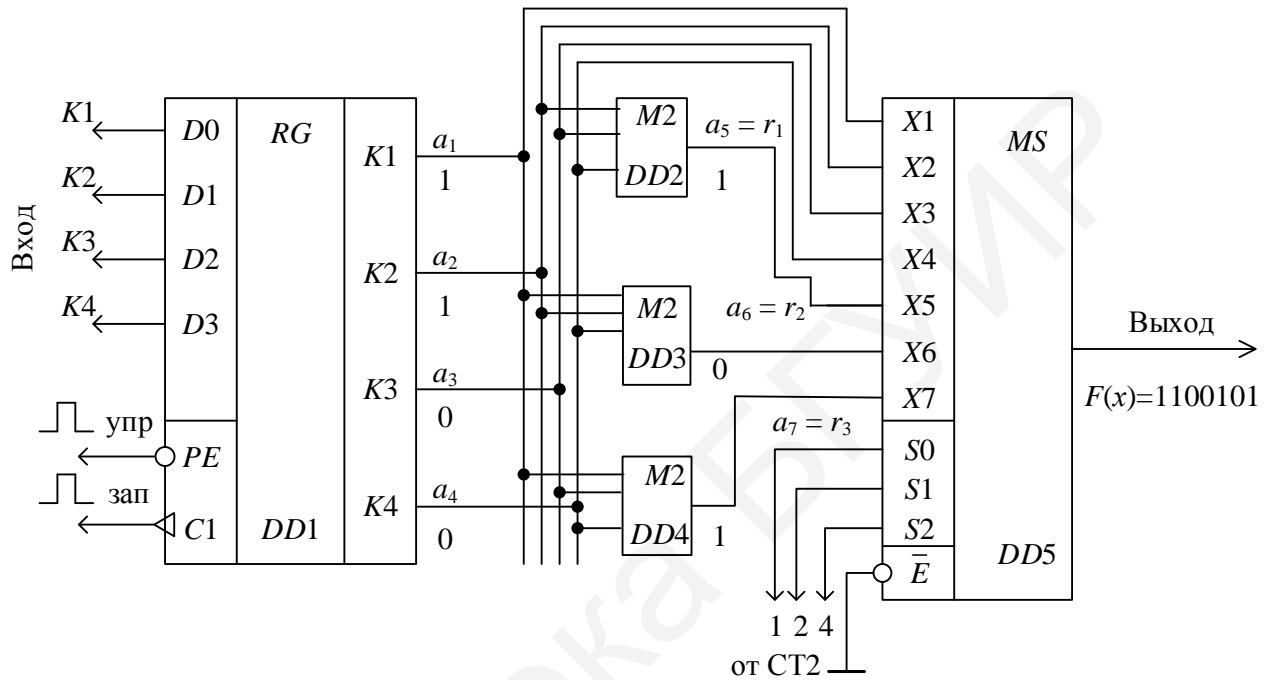


Рисунок 9.20 – Кодер систематического кода, исправляющего одиночные ошибки

Сумматорами  $DD2$ – $DD4$  по модулю 2 формируются контрольные символы  $r_1$ ,  $r_2$  и  $r_3$  в соответствии с выражениями (7.12)–(7.14). 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 (7.17):

$$S_1 = a_2 \oplus a_3 \oplus a_4 \oplus a_5,$$

$$S_2 = a_1 \oplus a_2 \oplus a_4 \oplus a_6,$$

$$S_3 = a_1 \oplus a_3 \oplus a_4 \oplus a_7.$$

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

Декодер систематического кода (7,4) показан на рисунке 9.21.

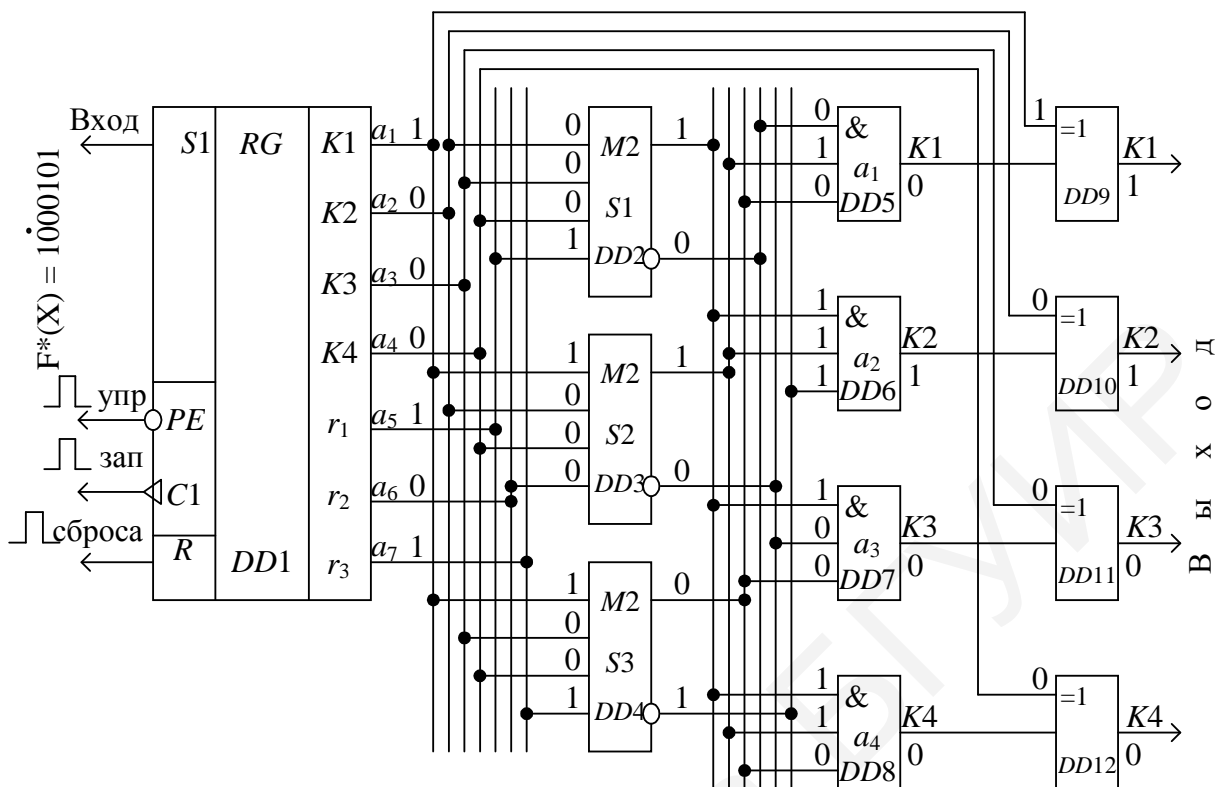


Рисунок 9.21 – Декодер систематического кода (7,4) для исправления одиночных ошибок

Кодовая комбинация из линии связи записывается в регистр  $DD1$  и поступает на входы определителя синдрома, собранного на элементах  $DD2-DD4$ . На сумматоре по модулю 2  $DD2$  осуществляется вычисление  $S_1$ , на  $DD3$  –  $S_2$ , на  $DD4$  –  $S_3$ . Далее сигналы с определителя синдрома  $S_1S_2S_3$  поступают на дешифратор синдрома, собранный на элементах И  $DD5-DD8$ , каждый из которых настроен на кодовую комбинацию (7.18), соответствующую одному из информационных символов  $a_1 = k_1, a_2 = k_2, a_3 = k_3$  и  $a_4 = k_4$ . Дешифратор синдрома для контрольных символов в схеме не предусмотрен, так как исправление контрольных символов не влияет на информацию, поступающую потребителю. При отсутствии ошибок на выходах всех элементов И  $DD5-DD8$  будет 0, который не влияет 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 выходные сумматоры по модулю 2  $DD9-DD12$  изменит его на противоположный. 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.

## 9.10 Кодирующее и декодирующее устройство кода Хемминга

Принцип построения кодирующего устройства не зависит от числа информационных разрядов передаваемого кода. Поэтому рассмотрим схему кодирующего устройства (рисунок 9.22) 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.

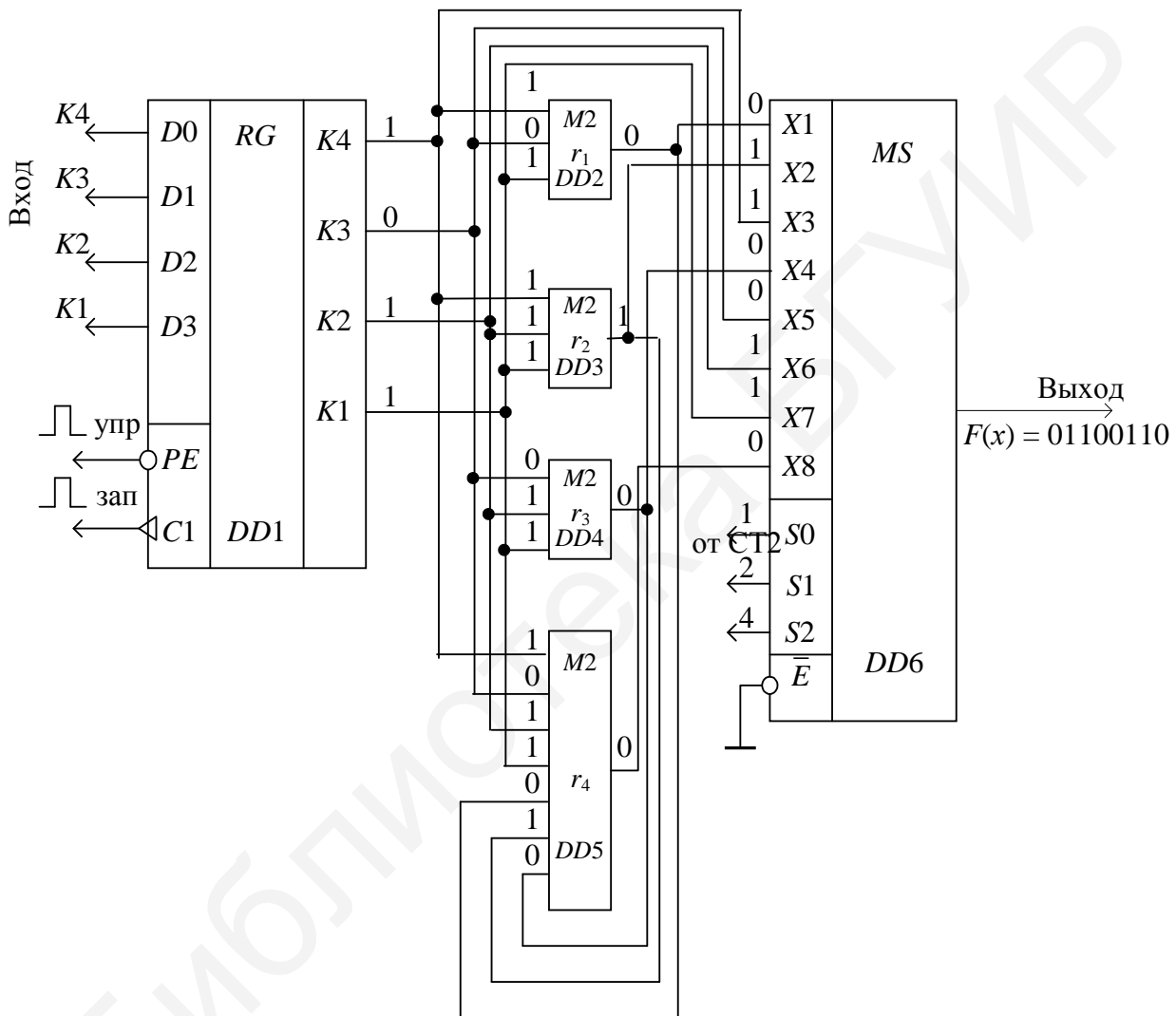


Рисунок 9.22 – Кодер кода Хемминга с  $d = 4$

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 пункте 7.3.2, 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 рисунке 9.22 в виде состояния элементов. В результате в линию связи поступит кодовая комбинация  $F(x) = 01100110$  с классическим порядком следования контрольных и информационных символов.

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 рисунке 9.23.

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

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 на элементе И-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  $DD11-DD14$  и запрещает выдачу информации потребителю.

При всех других соотношениях  $S_i$  и  $S_\Sigma$ , указанных в пункте 7.3.2, на выходе И  $DD6$  будет сигнал, равный 1.

Процесс декодирования кодовой комбинации  $F^*(x) = 010110110$  показан на схеме в виде состояния элементов. В данном случае на выходе схемы И  $DD6$  будет 0, а следовательно, схемы И  $DD11-DD14$  будут закрыты, информация потребителю не поступит и будет включен индикатор  $HLR$ , свидетельствующий о двойной ошибке.

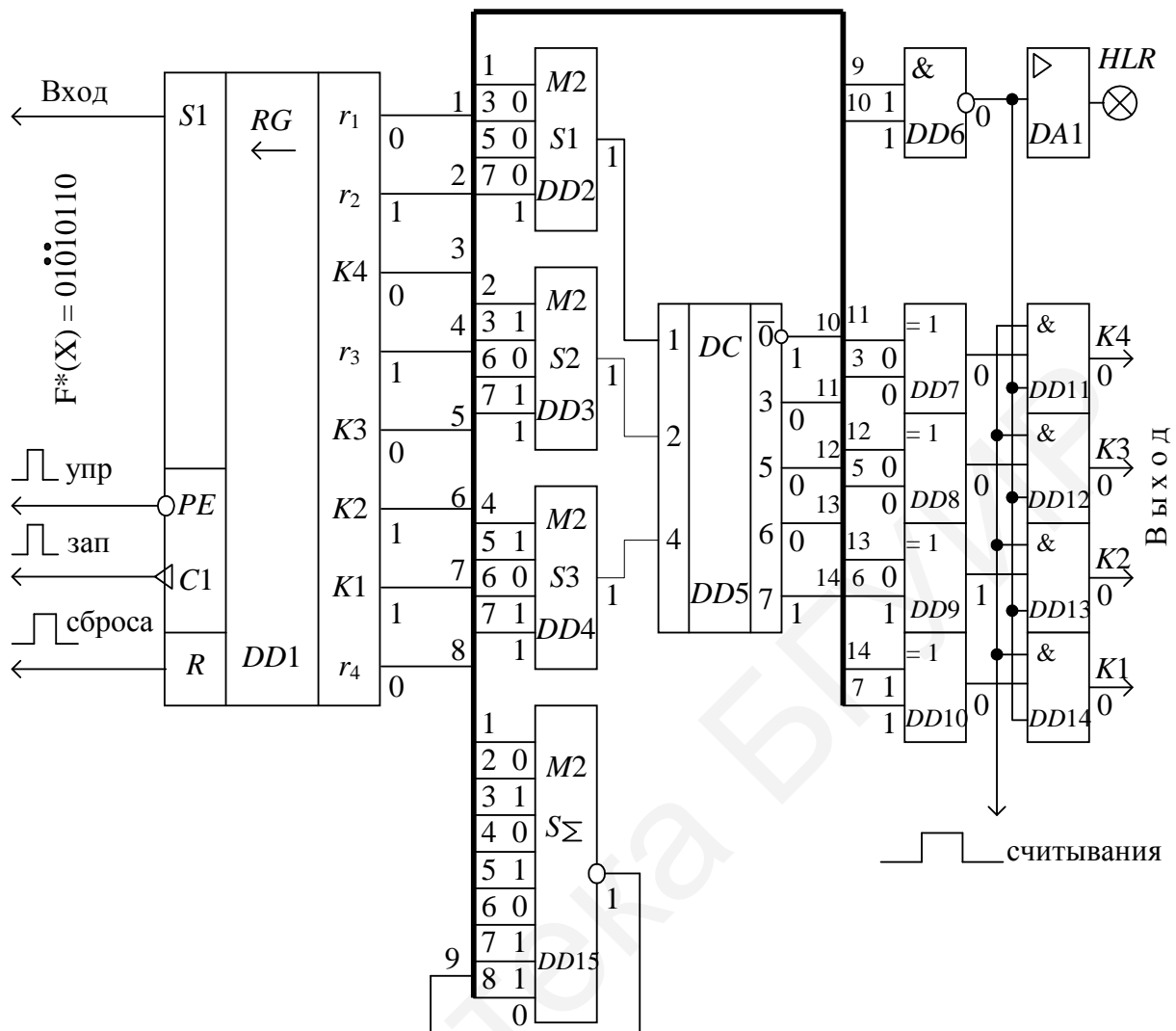


Рисунок 9.23 – Декодер кода Хемминга (8,4), позволяющего исправлять одиночные ошибки и обнаруживать двойные

## 9.11 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 членам 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 рисунке 9.24.

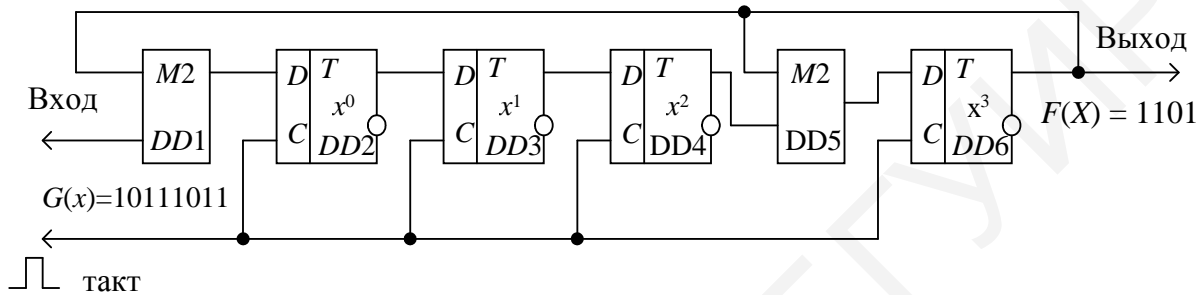


Рисунок 9.24 – Схема для деления на многочлен  $P(x) = (x^4 + x^3 + 1)$

Разделим на этот многочлен (делитель) многочлен  $G(x) = x^7 + x^5 + x^4 + x^3 + x + 1$  (делимое). Результат деления представлен в виде таблицы 9.2, где стрелками показаны направления процессов между элементами.

Из таблицы 9.2 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 9.2 – 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

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 по обратной связи с ячейки  $DD6$  в ячейку  $DD2$  записывается 1, хотя на вход поступает 0 делимого. 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 записываются в ячейки регистра начиная с такта 5. Последний остаток  $R(x) = 1110$  записан в такте 8.

На рисунке 9.25 изображена схема умножителя на многочлен  $P(x) = (x^4 + x^3 + 1)$ .

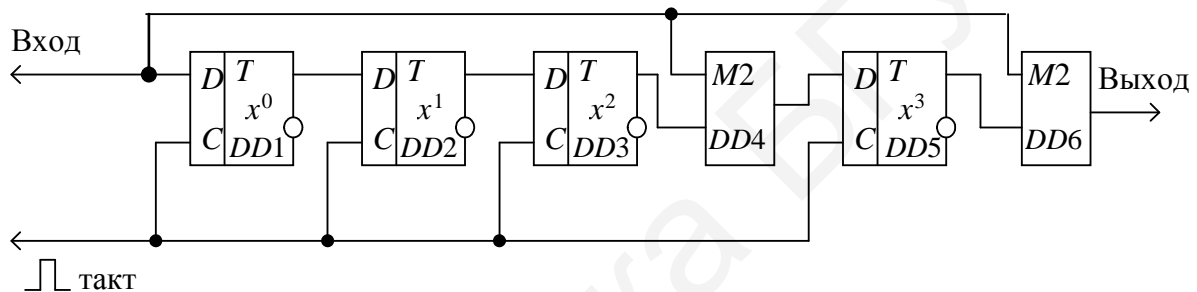


Рисунок 9.25 – Схема для умножения на многочлен

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 таблице 9.3.

Таблица 9.3 – 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



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.

## 9.12 Кодер и декодер циклического кода

Как указывалось в пункте 9.3.3, 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 (7.29). Таким образом, кодирующее устройство должно совмещать функции умножения и деления.

Схема кодирующего устройства циклического кода (7,4), образованного с помощью генераторного полинома  $P(x) = (x^3 + x^2 + 1) = 1011$ , показана на рисунке 9.26.

В его состав 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 ( $DD1-DD4$ ,  $DD6$ ) 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  $DD1-DD4$  начиная со старшего разряда поступает на выход и входной сумматор  $DD10$ . В процессе ее прохождения за  $k$  тактов в ячейках регистров сдвига  $DD11$ ,  $DD13$ ,  $DD14$  накапливается  $r$  проверочных разрядов. После четвертого такта ключ  $DA2$  закрывается, а ключ  $DA1$  открывается. Записанные в ячейках регистра три проверочных разряда тремя тактами поступают на выход кодирующего устройства.

Процесс кодирования входного сообщения  $G(x) = 1001$  с помощью схемы, представленной на рисунке 9.26, показан в таблице 9.4.

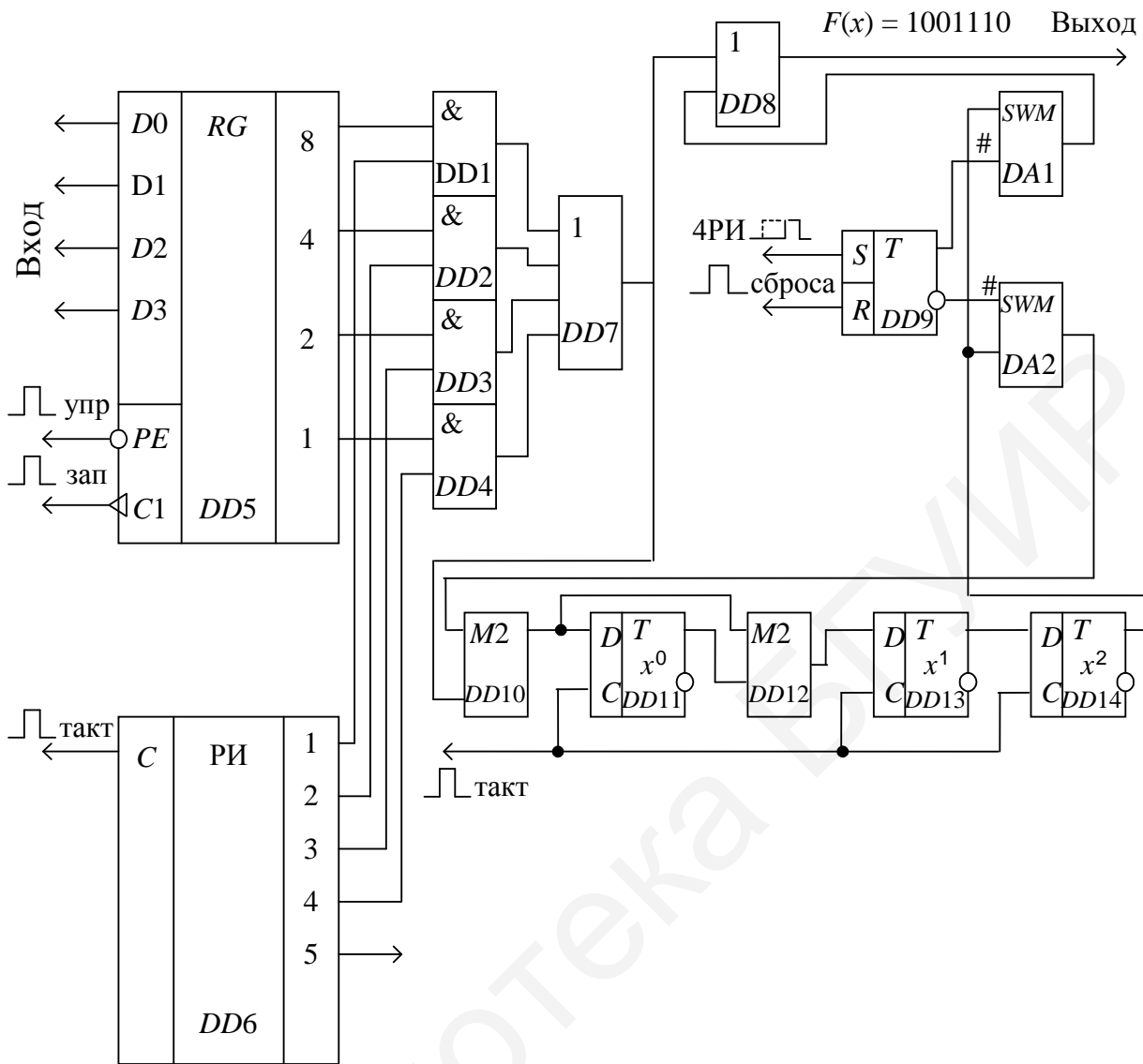


Рисунок 9.26 – Кодер циклического кода (7,4)

Таблица 9.4 – Образование циклического кода

Номер такта	Вход	Состояние ячеек			Выход после четвертого такта
		DD11	DD13	DD14	
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. пункт 7.3.3), 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 рисунке 9.27.

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  $DD8-DD11$ .

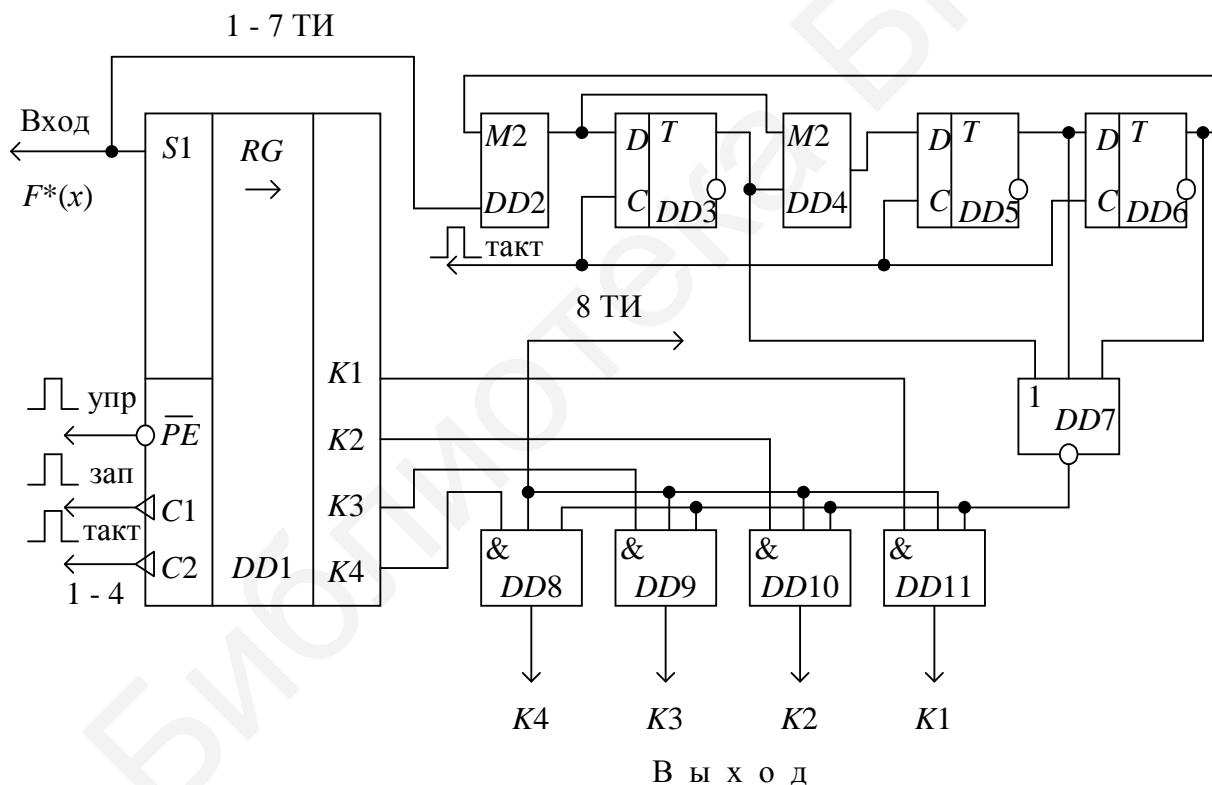


Рисунок 9.27 – Декодер циклического кода (7,4) для обнаружения ошибок

Принимаемая последовательность записывается в ячейки буферного регистра  $DD1$  на первых четырех тактах и одновременно поступает в декодирующий регистр на тактах 1–7. Таким образом, в регистре  $DD1$  оказываются лишь информационные разряды. На восьмом 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  $K1-K4$  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  $P(x) = (x^3 + x^2 + 1)$ . 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 рисунке 9.28.

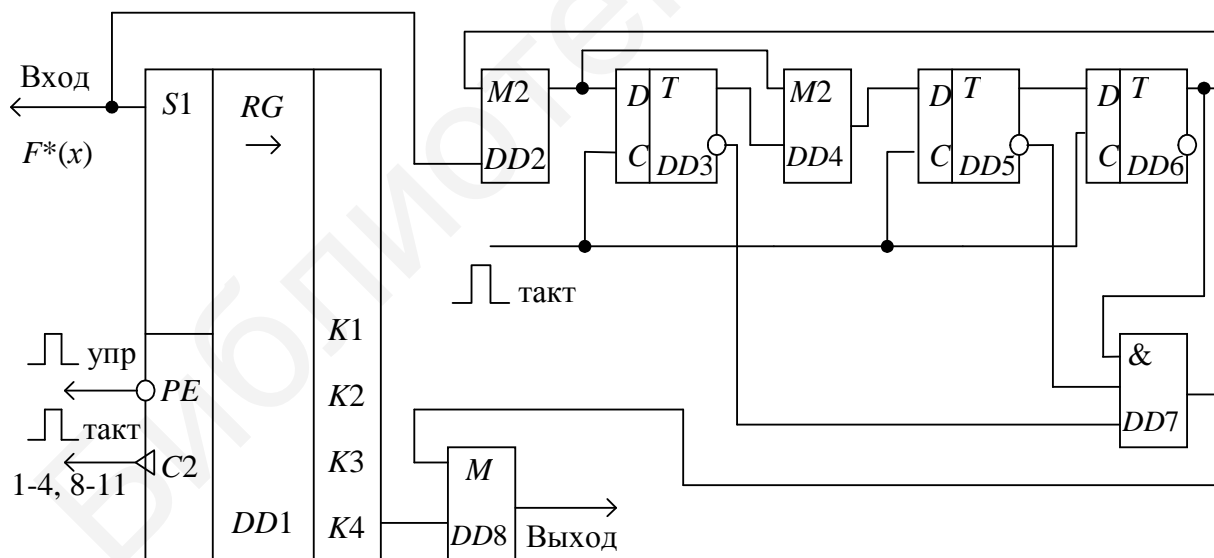


Рисунок 9.28 – Декодер циклического кода  $(7,4)$  для исправления одиночной ошибки

Пусть по каналу связи была передана комбинация  $F^*(x) = 1001110$ , которая под действием помех приняла вид  $F^*(X) = 1011110$ . Процесс исправления ошибки представлен в таблице 9.5.

На седьмом 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 таблицы 9.5, на девятом такте в ячейках декодирующего регистра сформирована комбинация 001, на десятом такте эта комбинация поступает на вход дешифратора синдрома, на выходе которого возникает сигнал 1. В этот же момент на выходной сумматор поступает искаженный третий разряд, который, проходя через сумматор, меняет знак на обратный. Исправленная комбинация имеет вид 1001.

Таблица 9.5 – 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	–	0	–
1	1	1	1	0	110	0	–
2	0	0	1	1	000	0	–
3	1	0	0	1	101	0	–
4	1	0	0	0	111	0	–
5	1	1	1	0	110	0	–
6	1	1	0	1	000	0	–
7	0	1	0	0	011	0	–
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 рисунке 9.29 представлена функциональная схема мажоритарного декодирования кода (7,3) (см. пример 7.17).

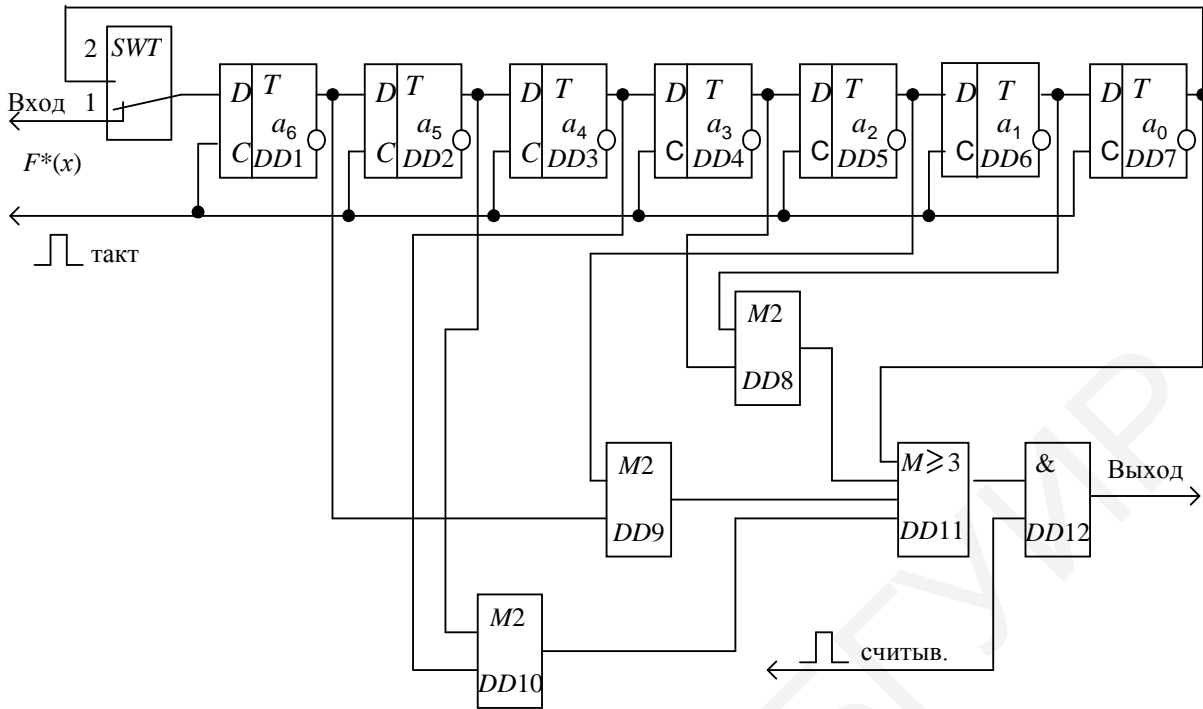


Рисунок 9.29 – Мажоритарный декодер циклического кода (7,3)

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  $a_0$  (7.49):  $a_1 \oplus a_3$  ( $DD8$ ),  $a_2 \oplus a_6$  ( $DD9$ ) и  $a_4 \oplus a_5$  ( $DD10$ );  $a_0$  поступает непосредственно в схему, где происходит подсчет 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 (7.50) 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  $DD1-DD7$ , а в течение последующих определяется значение каждого из  $n$  разрядов. Вывод информации потребителю осуществляется с выхода мажоритарного элемента через схему И  $DD12$ .

Разновидностью циклических кодов является код Файра (см. подпункт 7.3.3.7). Кодирование производится по общим правилам для циклических кодов. Здесь остановимся на специфике декодирования.

Как известно, образующий полином состоит из произведения двух многочленов с различными степенями. Поэтому принятая комбинация  $F^*(x)$  делится одновременно на первый и второй полиномы. Равенство остатков на  $n$ -м такте в двух делителях является условием возможности исправления обнаруженного пакета ошибок. Поясним это более подробно.

Учитывая, что степени этих полиномов различные, то и остатки будут появляться на различных тактах. Так, при делении  $F^*(x)$  на первый многочлен, остаток,

определяемый многочленом ошибок  $E(x)$  (см. подпункт 7.3.3.9), появляется в первый раз на  $(2^{b_s}-1)$ -м такте.

Следовательно, для того чтобы он появился на  $n$ -м такте, общее число разрядов должно быть кратно  $(2^{b_s}-1)$ .

В процессе деления  $F^*(x)$  на многочлен  $x^{2^{b_s}-1}+1$  остаток, определяемый многочленом ошибок  $E(x)$ , впервые появится на  $(2^{b_s}-1)$ -м такте, и затем его появление циклически повторяется с периодом  $(2^{b_s}-1)$  тактов, следовательно, общая длина кодовой комбинации должна быть кратна  $(2^{b_s}-1)$ .

Из приведенных выше рассуждений становится понятно выражение (7.43), что  $n = \text{НОК}(E, C) = \text{НОК}[(2^{b_s}-1), (2^{b_s}-1)]$  для кода, исправляющего пакет ошибок длиной, равной  $b_s$ .

Функциональная схема декодера приведена на рисунке 9.30.

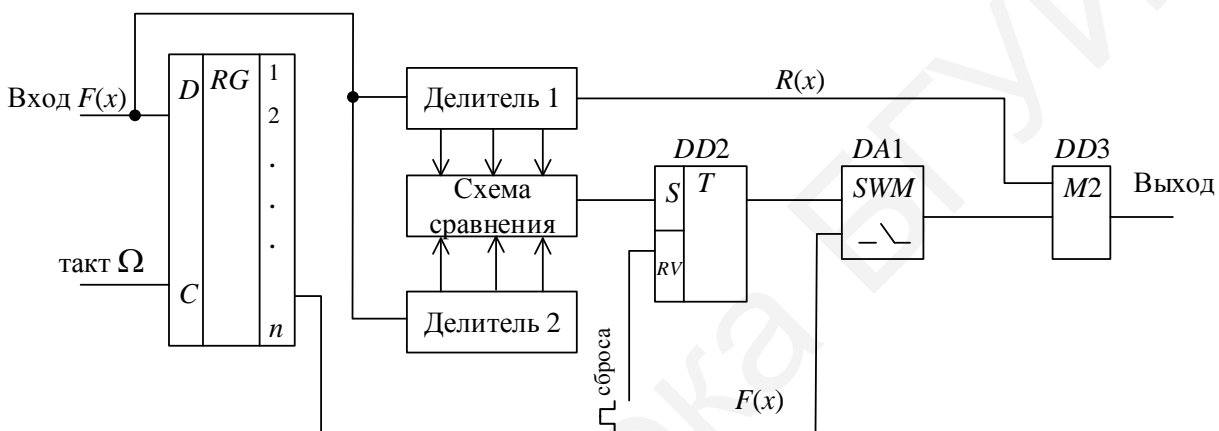


Рисунок 9.30 – Декодер кода Файра

Кодовая комбинация  $F^*(x)$  поступает на вход приемного регистра  $DD1$  и одновременно на вход двух делителей. Остатки с выходов делителей поступают в схему сравнения. При равенстве остатков на выходе схемы сравнения появляется логическая единица, которая устанавливает триггер  $DD2$  в единичное положение.

Сигнал с выхода триггера  $DD2$  открывает ключ  $DA1$ , и на вход корректора ошибок  $DD3$ , который представляет собой сумматор по модулю 2, поступает принятая кодовая комбинация и остаток (синдром). Таким образом, при прохождении через сумматор по модулю 2  $DD3$  происходит коррекция принятой комбинации. Кодовая комбинация с выхода  $DD3$  поступает в приемник информации.

В общем случае, когда  $E(X)$  начинается не со старшего разряда, а с  $j$ -го, для обеспечения равенства остатков, полученных на  $n$ -м такте, нужно проводить последовательное домножение их на  $X$  с приведением по модулю в одной схеме и по модулю в другой.

Если пакет начинается с  $j$ -го разряда, то необходимо сделать дополнительно  $(n-j)$  тактов, прежде чем остатки в делителях сравняются.

За  $(n-j)$  дополнительных сдвигов содержимое приемного регистра  $DD1$  сместится, и ошибочные символы снова окажутся непосредственно перед схемой коррекции  $DD3$ .

Если за время вывода  $F^*(x)$  из приемного регистра условия возможности проведения коррекции не были выполнены, то обнаружена неисправимая ошибка.

Рассмотрим процесс исправления пакетов ошибок кодом Файра с образующим полиномом  $P(X)_\Phi = (x^3 + x^2 + 1)(x^5 + 1)$ .

Код может исправлять пакеты ошибок, состоящие из трех символов ( $b_s = 3$ ). Длина кодовой комбинации  $n = (2^3 - 1)(2 \cdot 3 - 1)$ . Предположим, что в трех старших разрядах нулевой кодовой комбинации возник пакет ошибок типа  $E(x) = 101$ .

При делении  $E(X)$  на  $P(x) = x^3 + x^2 + 1$  остаток  $E(X)$  будет появляться в регистре с периодом семь тактов. Чередование остатков при делении  $E(X)$  на  $x^5 + 1$  происходит с периодом пять тактов.

Последовательность изменения остатков в регистрах делителей 1 и 2 при поступлении на вход декодирующего устройства нулевой кодовой комбинации с пакетом ошибок 101 в старших разрядах приведена в таблице 9.6.

Таблица 9.6 – Последовательность изменения остатков в регистрах делителей 1 и 2

Номер такта	Остаток в регистре делителя 2: $x^5 + 1$	Остаток в регистре делителя 1: $x^3 + x^2 + 1$
1	01001	111
2	10010	011
3	00101	110
4	01010	001
5	<b>10100</b>	010
6	01001	100
7	10010	<b>101</b>
8	00101	111
9	01010	011
10	<b>10100</b>	110
11	01001	001
12	10010	010
13	00101	100
14	01010	<b>101</b>
15	<b>10100</b>	111
...	...	...
...	...	...
...	...	...
28	00101	<b>101</b>
29	01010	111
30	<b>10100</b>	011
31	01001	110
32	10010	001
33	00101	010
34	01010	100
35	<b>10100</b>	<b>101</b>



Условие возможности исправления пакета ошибок выполняется на 35-м такте. За последующие три такта при выводе кодовой комбинации из приемного регистра на схему коррекции DD3 пакет ошибок устраняется.

### 9.13 Кодер и декодер итеративного кода

Рассмотрим кодер итеративного кода для структуры кодовой комбинации, приведенной в таблице 9.7.

Таблица 9.7 – 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 рисунке 9.31.

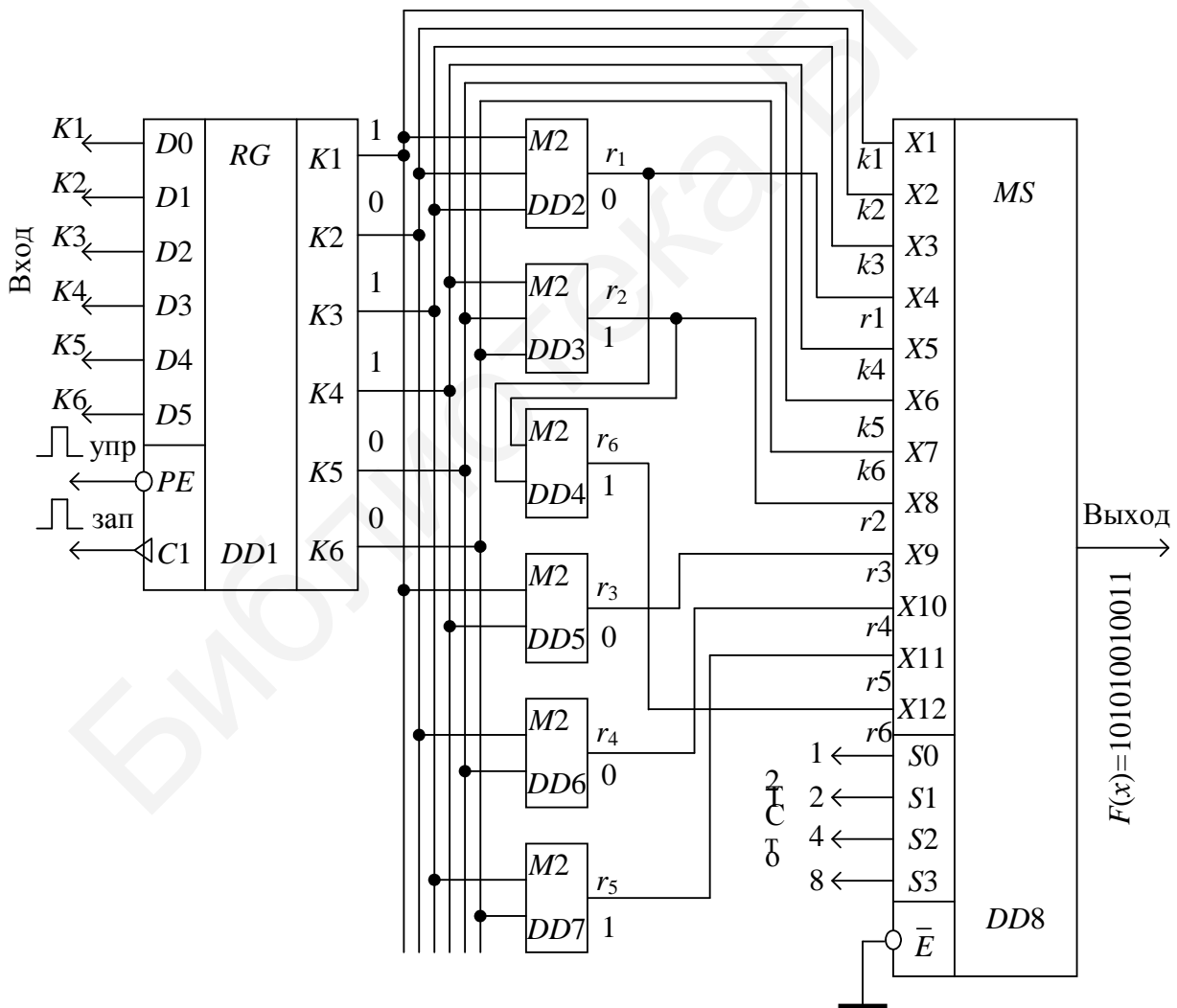


Рисунок 9.31 – Кодер итеративного кода

Кодовая комбинация, подлежащая кодированию, записывается в буферный регистр *DD1*. Формирование контрольных символов  $r_1-r_6$  сумматорами по модулю 2 *DD2-DD7* осуществлено в соответствие со строками и столбцами таблицы 9.7. Мультиплексор *DD8* осуществляет преобразование параллельного кода в последовательный.

Декодер для рассматриваемой структуры кодовой комбинации приведен на рисунке 9.32.

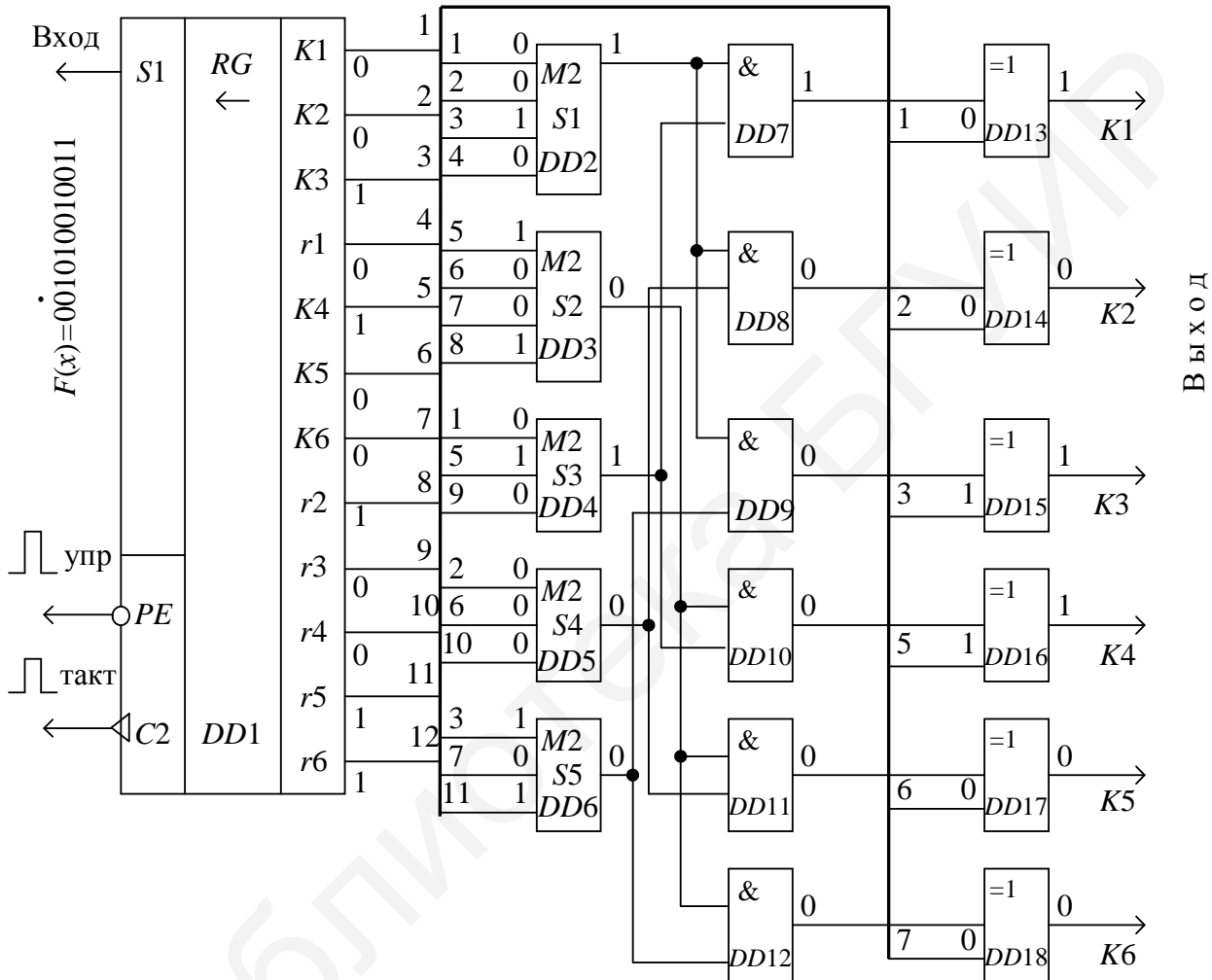


Рисунок 9.32 – Декодер итеративного кода

Кодовая комбинация в итеративном коде записывается в буферный регистр *DD1*. Затем сумматорами по модулю 2 *DD2-DD6* осуществляются проверки в соответствии с таблицей 9.7, 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  $S_5S_4S_3S_2S_1$  должен быть нулевого порядка. При наличии одиночной ошибки на выходе соответствующих сумматоров *DD2-DD6* появляются единицы, которые поступают на дешифратор синдрома, собранный на элементах И *DD7-DD12*. Дешифратор однозначно указывает номер искаженного символа, и единицы с соответствующей схемы И *DD7-DD12* поступают 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. На схеме показана дешифрация кодовой комбинации  $F^*(x) = \dot{0}01010010011$  в виде состояния элементов. В результате дешифрации получим синдром  $S_5S_4S_3S_2S_1 = 00101$ , что вызвало появление единицы 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 единицу и на вход потребителя поступила кодовая комбинация  $k_1k_2k_3k_4k_5k_6 = 101100$ , что соответствует исходной, указанной в таблице 9.7.

### 9.14 Кодер и декодер рекуррентного кода

Процесс образования и декодирования кодовых комбинаций рекуррентного кода достаточно полно рассмотрен в пункте 7.3.5. 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 рисунке 9.33.

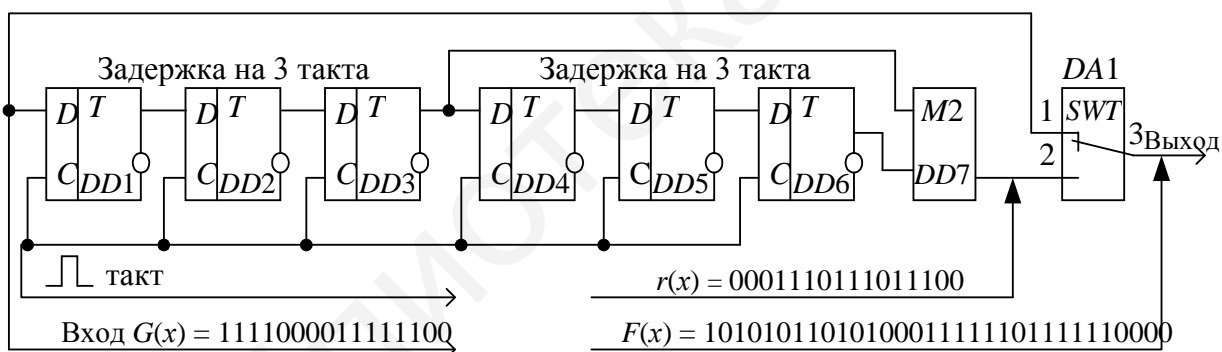


Рисунок 9.33 – Функциональная схема кодера рекуррентного кода при  $b = 3$

Процесс образования контрольных символов  $r(x)$  с помощью данного кодера представлен в таблице 9.8.

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 рисунке 9.34.

Как известно из пункта 7.3.5, 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.

Таблица 9.8 – 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

Рассмотрим работу декодера. Входная кодовая комбинация  $F^*(x)$  разделителем  $DA1$  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 регистра содержатся безошибочные информационные символы. 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  $DD4-DD6$ .

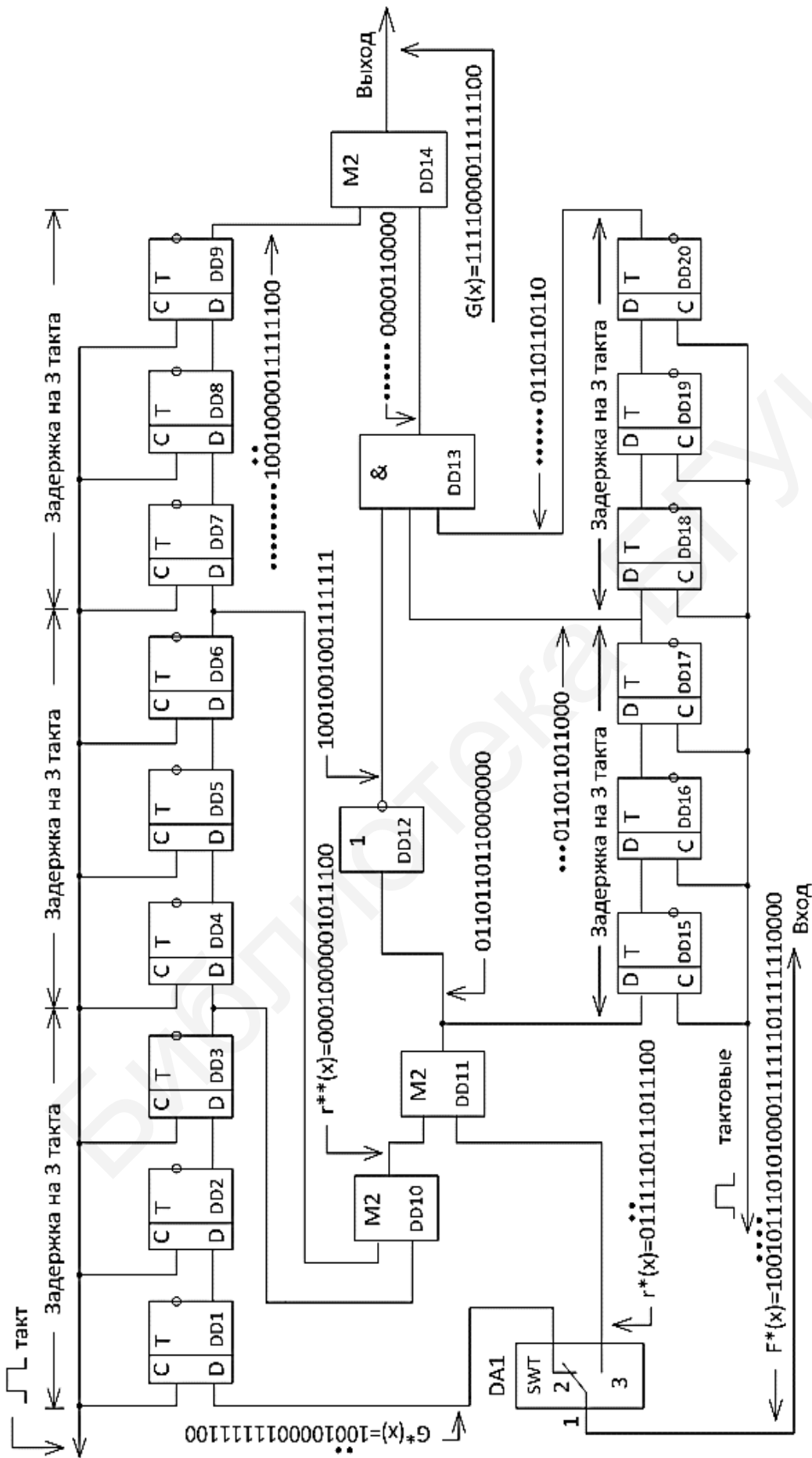


Рисунок 9.34 – Функциональная схема декодера рекуррентного кода при  $b=3$

Как видно из рисунка 9.34, 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 проводились исправления в случае появления ошибочных символов в этот период, предусмотрен элемент НЕ  $DD12$ .

Функционирование декодирующего устройства при дешифрации конкретного сообщения  $F^*(x)$  показано на рисунке 9.34 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.

# 10 ТЕХНИЧЕСКИЕ СРЕДСТВА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ЧАСТОТНЫХ КОДОВ

## 10.1 Кодер и декодер кода на перестановки

Теоретические аспекты построения данного кода рассмотрены в пункте 7.5.2, а кодер для трех частот приведен на рисунке 10.1.

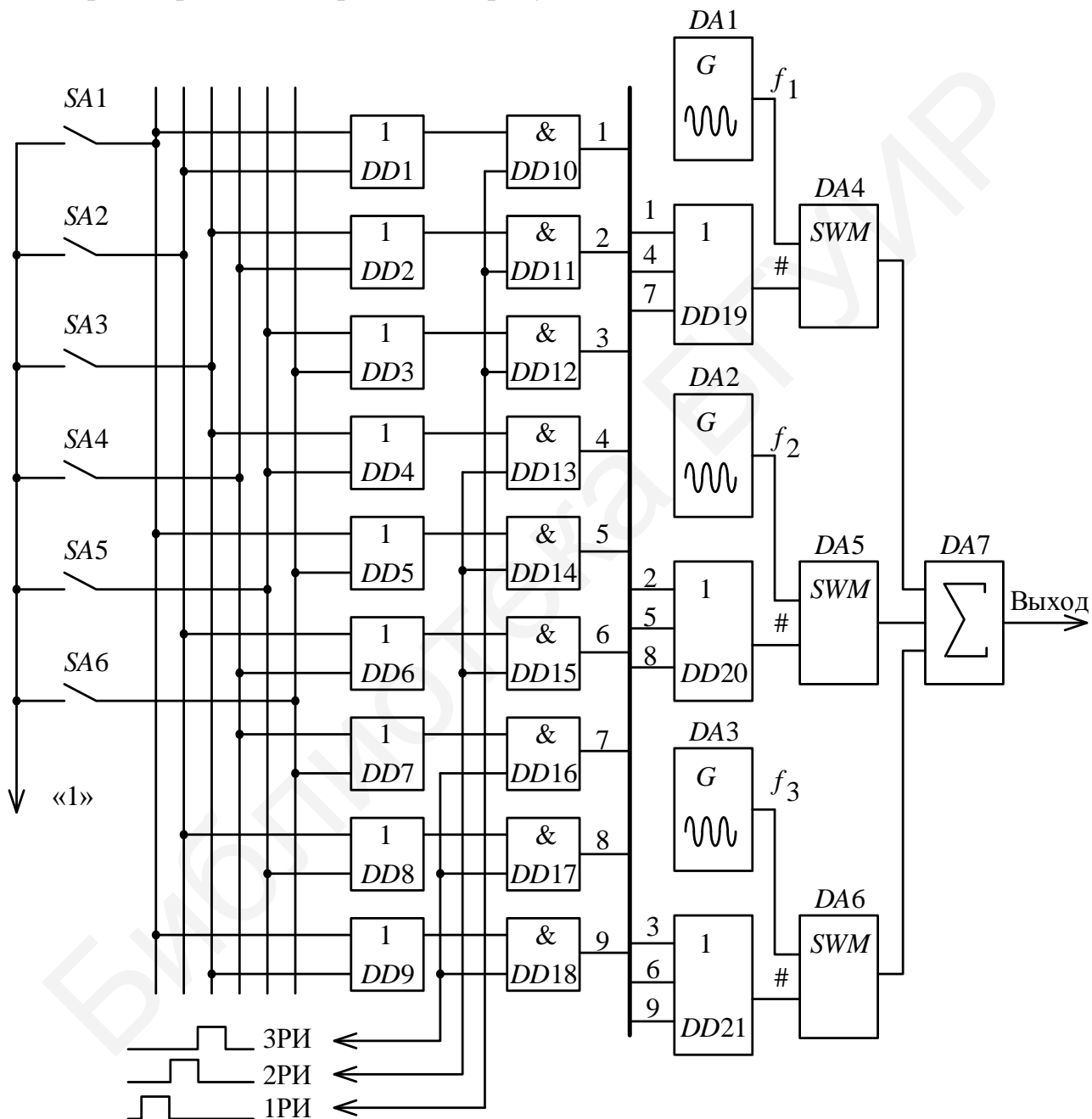


Рисунок 10.1 – Схема кодера кода на перестановки  $P_3$

В соответствии с выражением (7.104) число команд, которые могут быть представлены в данном коде, равно шести, а именно:  $f_1f_2f_3$ ,  $f_1f_3f_2$ ,  $f_2f_1f_3$ ,  $f_2f_3f_1$ ,  $f_3f_1f_2$ ,





ствующие временные позиции, 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, представленным на рисунке 10.1, и наглядно видно из рисунка 10.2, где на выходе каждого элемента подписаны частоты и очередность их следования.

## 10.2 Кодер и декодер кода на размещения

Рассмотрим кодер кода на размещение  $A_3^2$ . С помощью данного кодера можно передать шесть 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 рисунке 10.3.

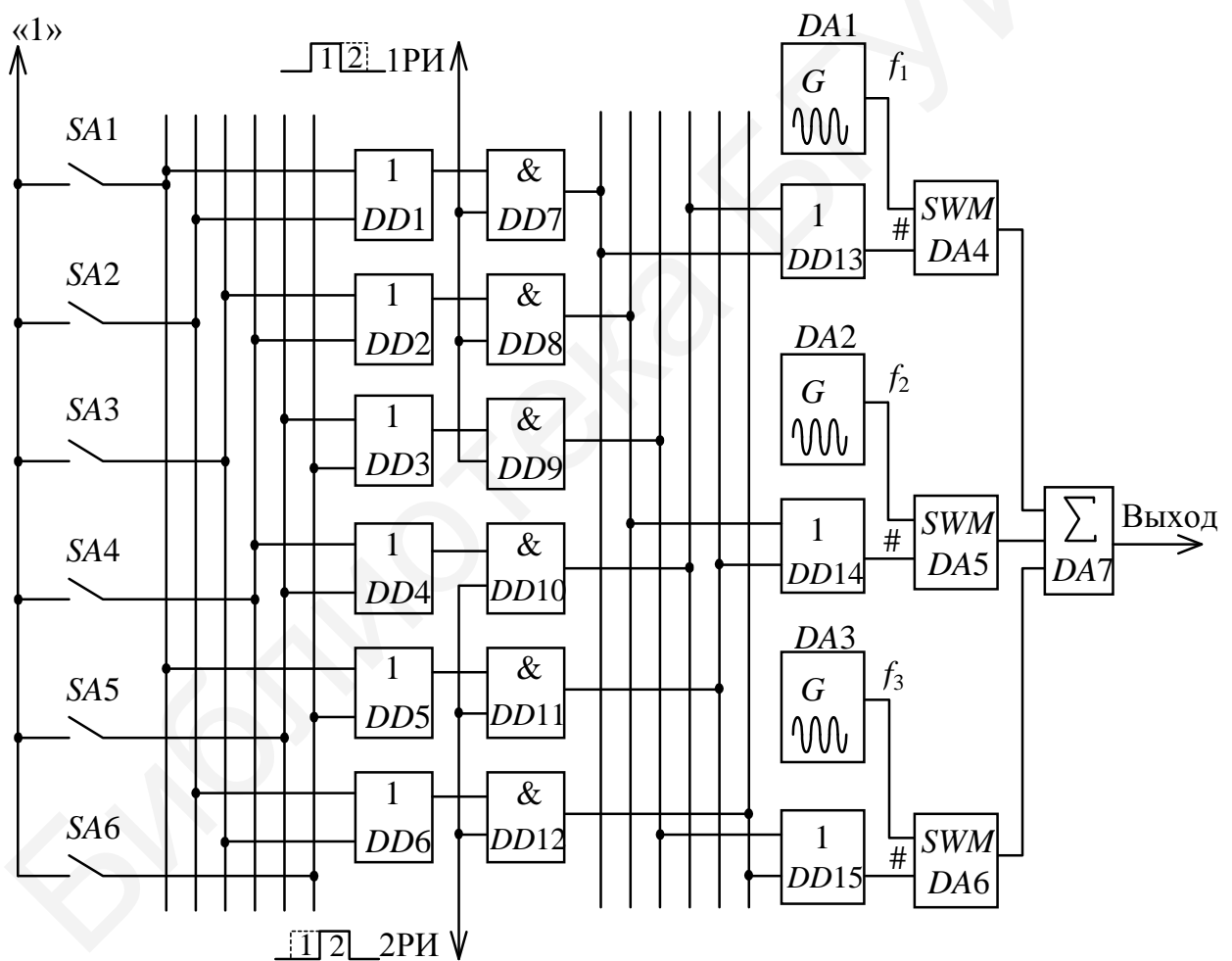


Рисунок 10.3 – Кодер кода на размещение  $A_3^2$

Элементами ИЛИ  $DD1–DD6$  формируются видеосигналы, соответствующие приведенным 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 И 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  $f_1f_2$ . Аналогично работает кодирующее устройство и при замыкании любого другого ключа SA2–SA6.

На рисунке 10.4 приведена схема декодирующего устройства последовательного двухчастотного кода на размещение из  $n = 3$  частот по две ( $m = 2$ ).

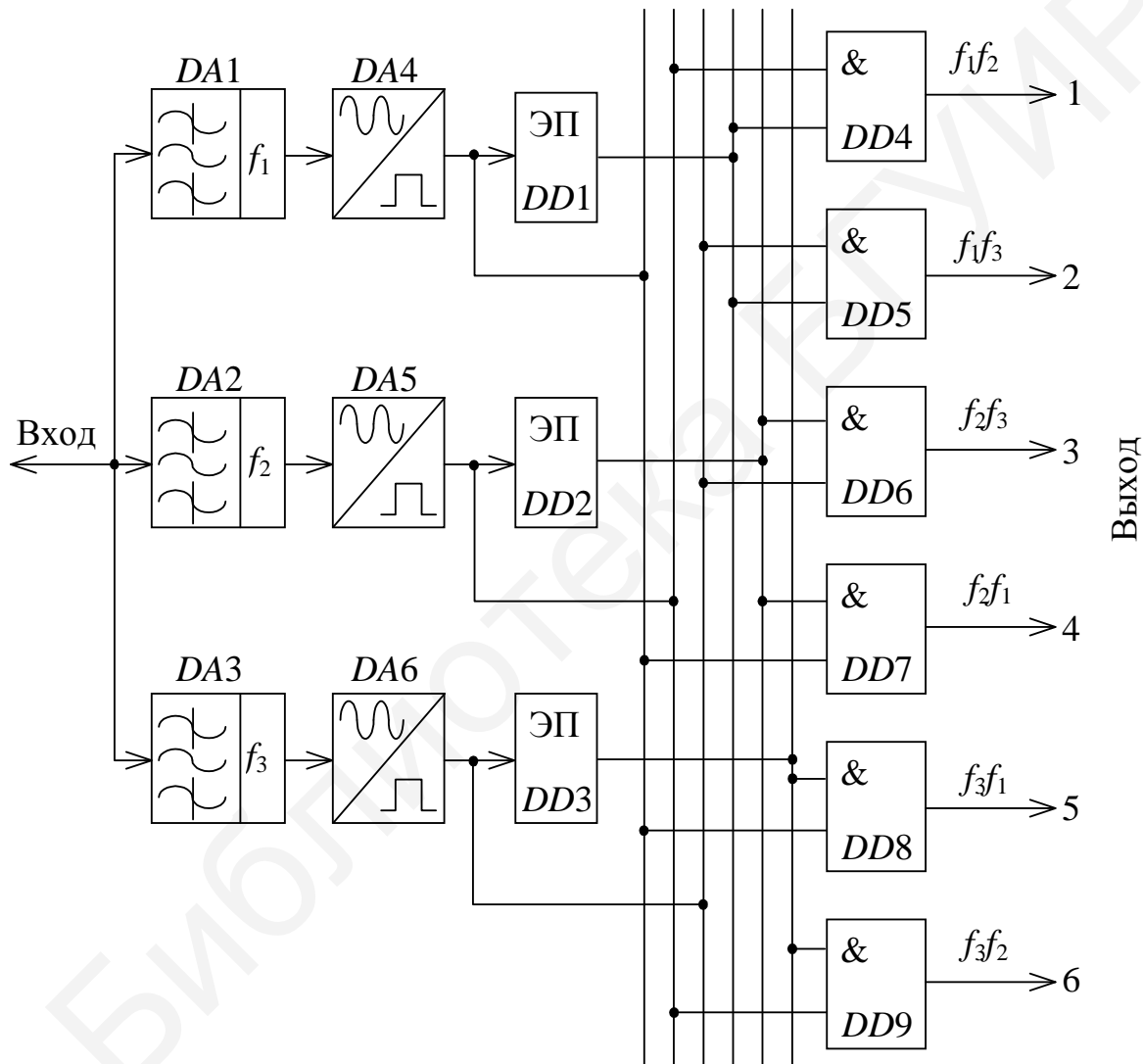


Рисунок 10.4 – Декодеры кода на размещение  $A_3^2$

Пришедшие от линейного блока радиоимпульсы разделяются полосовыми фильтрами DA1–DA3, затем выделяется огибающая радиоимпульса и формируется видеоимпульс с параметрами, необходимыми для работы интегральных микросхем преобразователей 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, представленной на рисунке 10.2. Дешифровка сигналов производится схемами И  $DD4-DD9$ , входы которых подключаются в соответствии с кодовыми комбинациями, сформированными при кодировании.

### 10.3 Кодер и декодер кода на сочетании

Принципы построения данных кодов приведены в пункте 7.5.4. Рассмотрим кодер кода на сочетание  $C_4^2$ , схема которого представлена на рисунке 10.5.

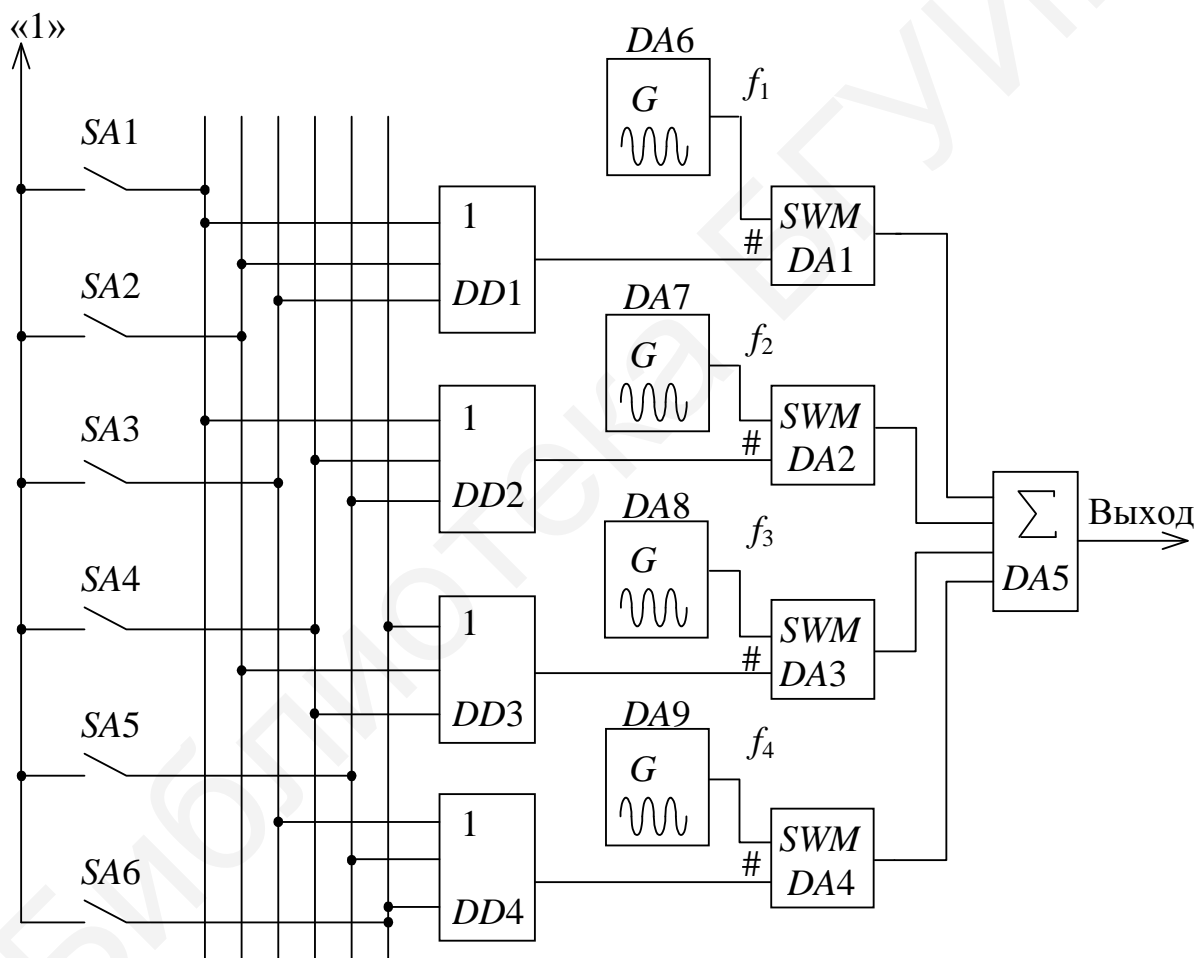


Рисунок 10.5 – Кодер кода на сочетание  $C_4^2$

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 передаются в параллельном виде. Формирование закона комбинирования радиоимпульсов в соответствии с приведенными комбинациями производится схемами ИЛИ  $DD1-DD4$  при замыкании соответствующего ключа  $SA1-SA6$ . Например, замыкая ключ  $SA1$ , плюс источника питания  $E_{п}$  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  $f_1f_3$  и т. д.

Декодер кода  $C_4^2$  приведен на рисунке 10.6.

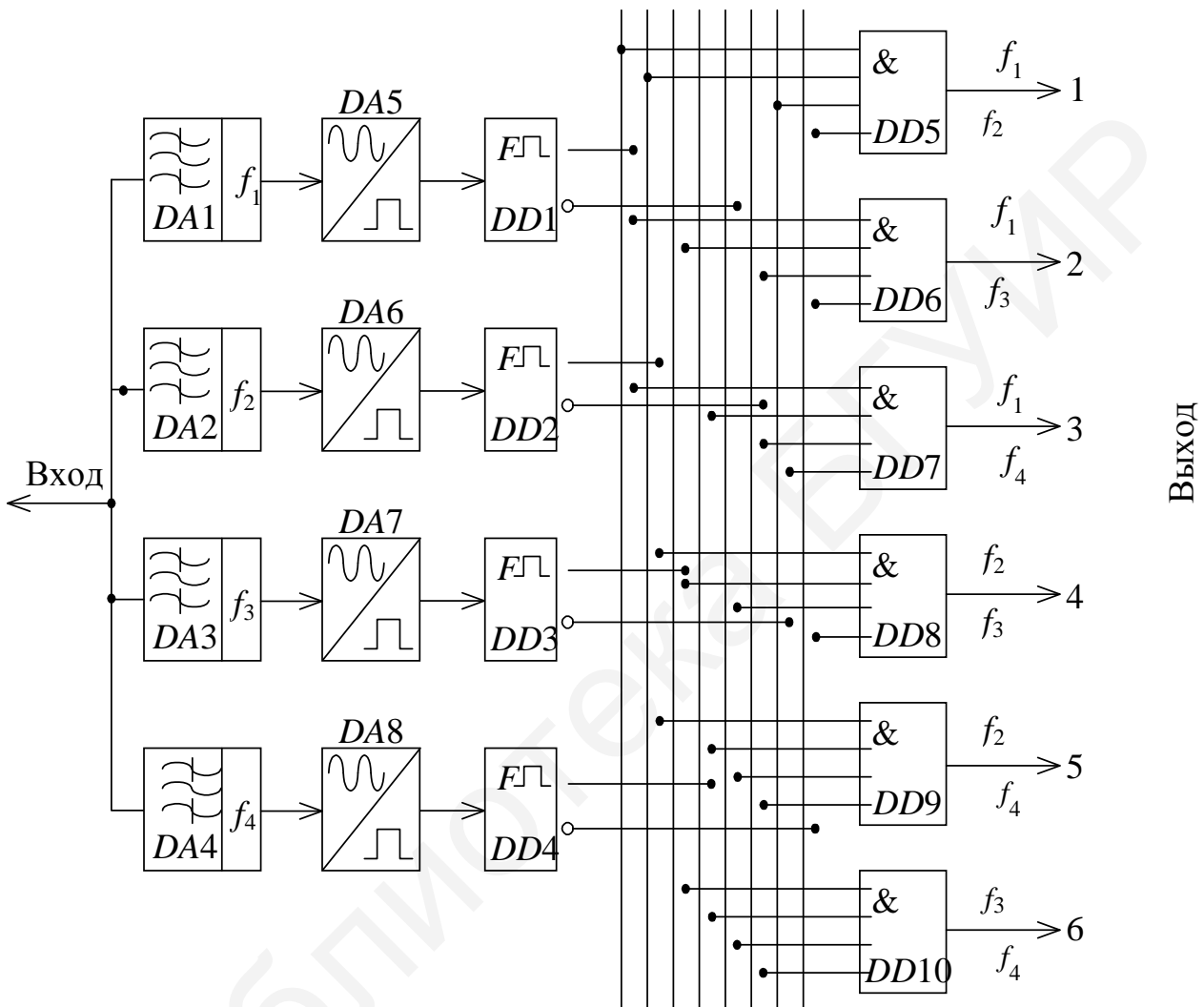


Рисунок 10.6 – Декодер кода на сочетание  $C_4^2$

Назначение полосовых фильтров  $DA1-DA4$ , преобразователей  $DA5-DA6$  и формирователей импульсов  $DD1-DD4$  такое же, как и в предыдущих схемах декодеров на перестановки (см. рисунок 10.2) и на размещение (см. рисунок 10.4).

При построении данного декодера большое внимание уделено обеспечению защитного отказа в случае сбоев. Для этой цели на входы дешифраторов, собранных на схемах И  $DD5-DD10$ , заведены два прямых (информационных) и два инверсных (неинформационных для данного канала) сигнала с выходов формирователей  $DD1-DD4$ . 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

$DD5$ – $DD10$  на выходе не будет иметь информационный сигнал, а следовательно, не произойдет ложное срабатывание объекта.

### 10.4 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 рисунке 10.7.

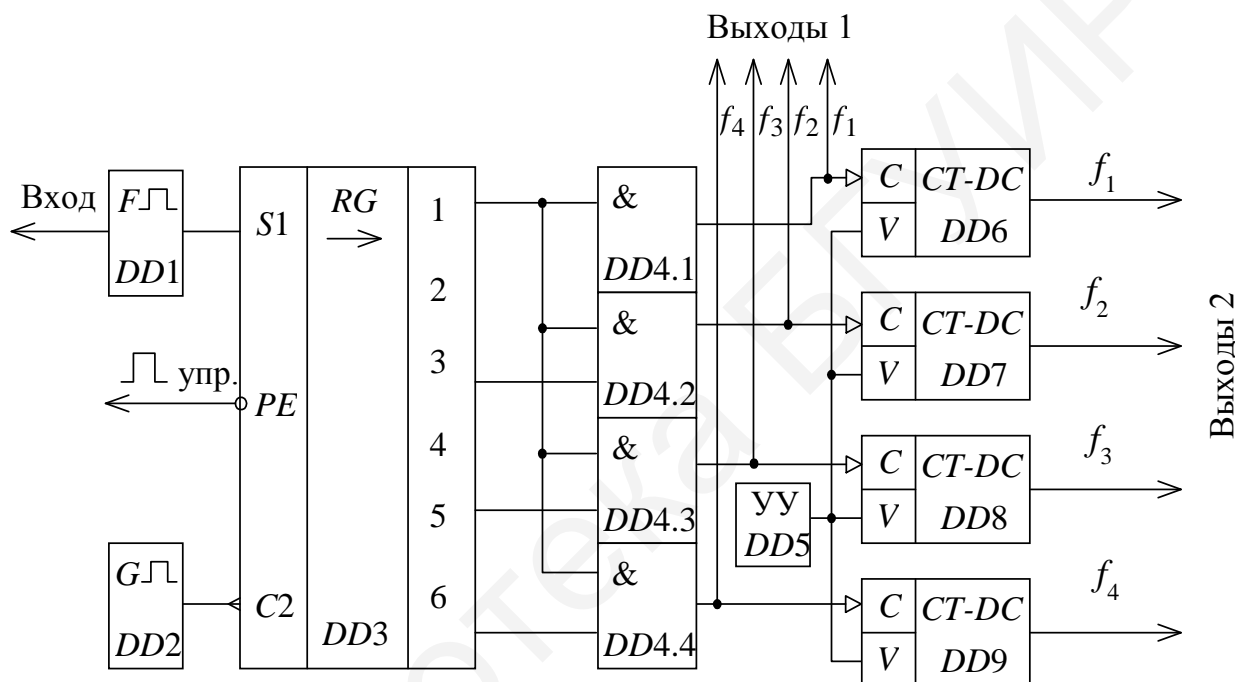


Рисунок 10.7 – Дешифратор одночастотного сигнала

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  $DD4.1$ – $DD4.4$  однозначно связан с частотой радиоимпульса. Для увеличения надежности управления сигналы с выхода схем И  $DD4.1$ – $DD4.4$  могут поступать на пересчетные схемы  $DD6$ – $DD9$ , 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  $DD5$ .

## 10.5 Кодер и декодер сменно-качественного кода

Как известно из пункта 7.5.5, 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 рисунке 10.8, 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  $DD4$ .

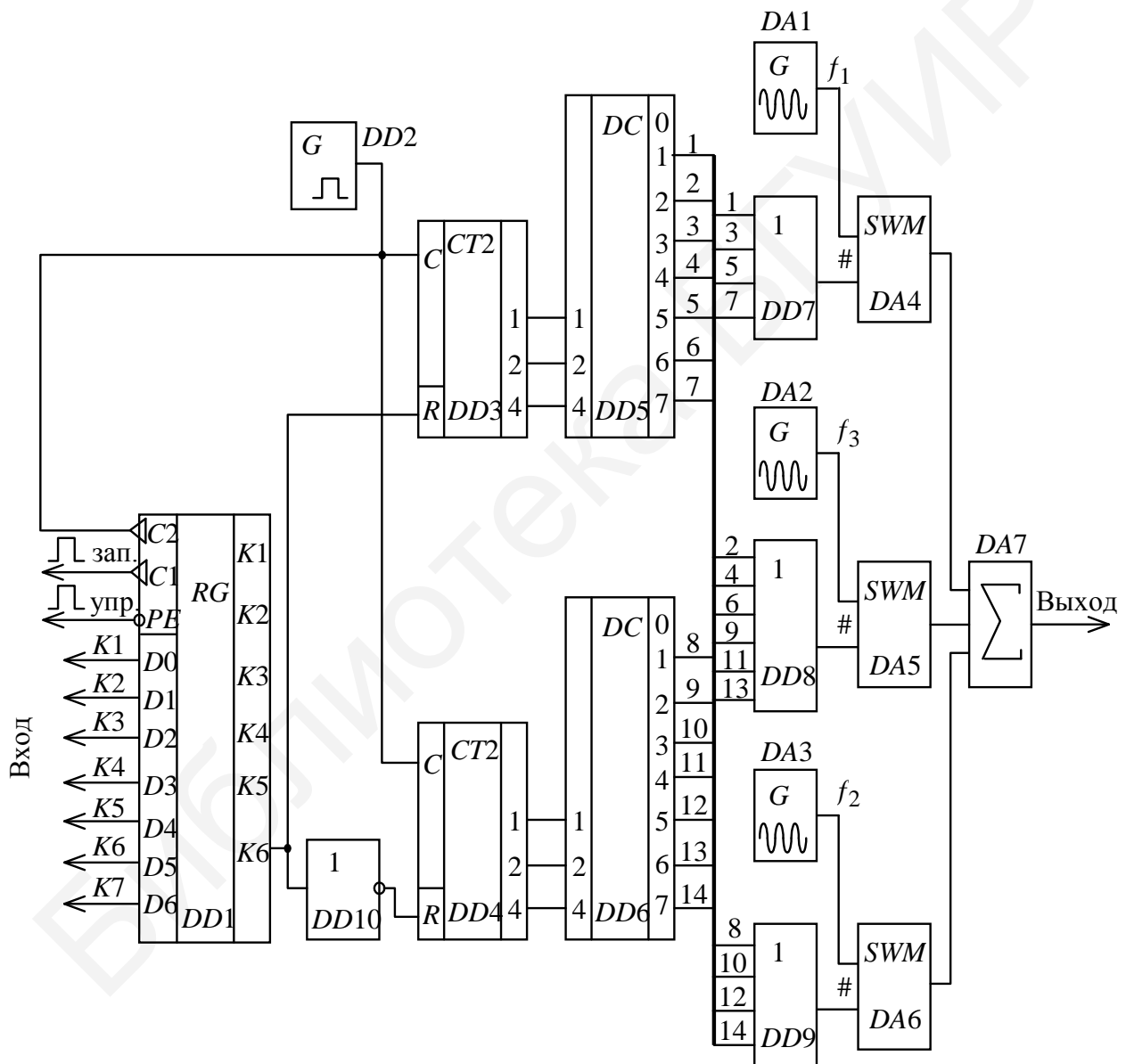


Рисунок 10.8 – Кодер сменно-качественного кода

Таким образом, счетчики  $DD3$  и  $DD4$  подсчитывают соответственно число единиц и нулей, следующих подряд в передаваемой комбинации. Дешифратором  $DD5$  при нечетном числе 1 выходы 1, 3, 5, 7 объединяются схемой ИЛИ  $DD7$ ,

этим 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, представленных на рисунке 10.9.

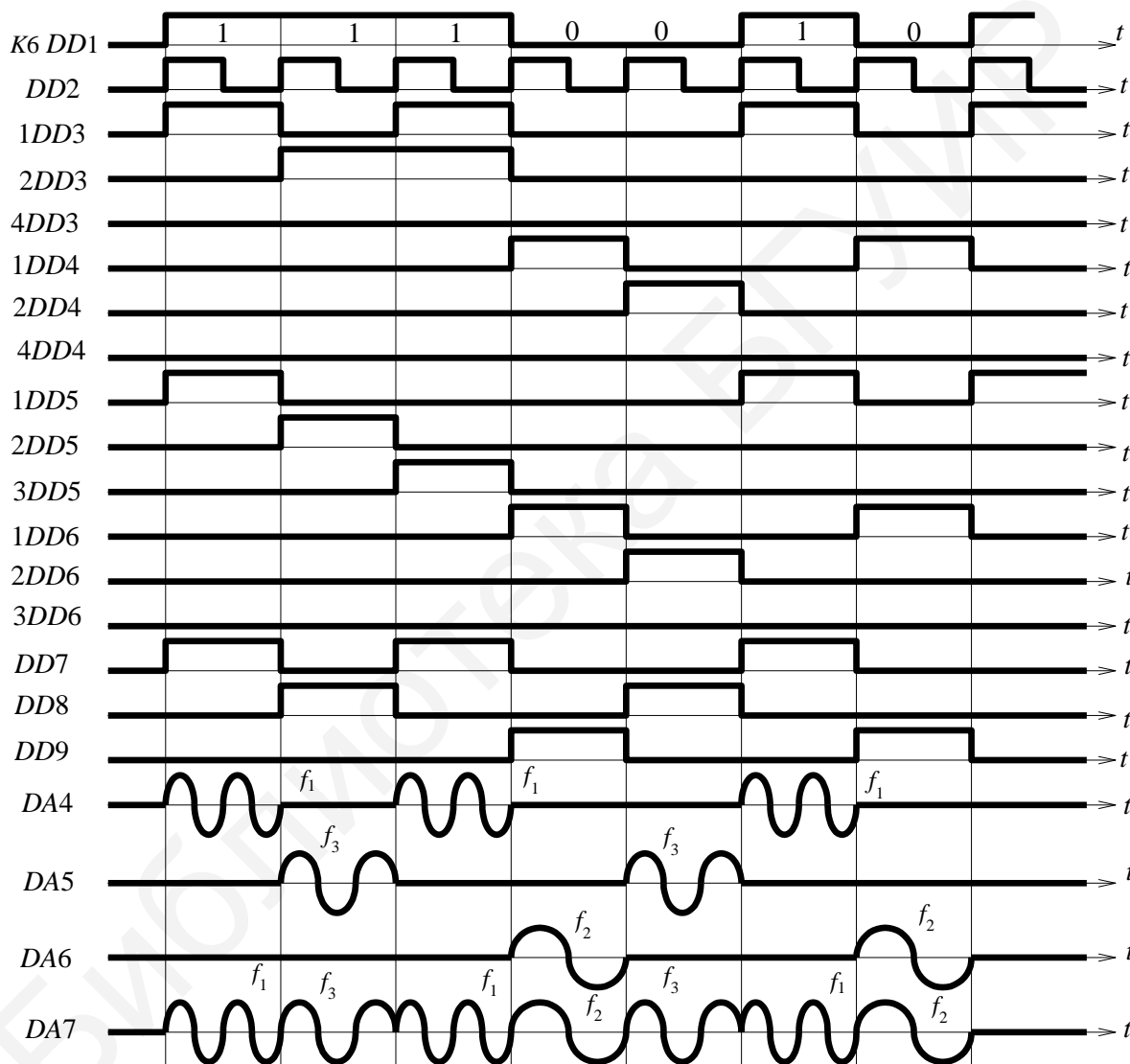


Рисунок 10.9 – Временные диаграммы работы кодера сменно-качественного кода

Основой декодера (рисунок 10.10) являются полосовые фильтры  $DA1$ – $DA3$ , которые селективируют радиоимпульсы, поступившие из канала связи, и преобразователи радиоимпульсов в видеоимпульсы  $DA4$ – $DA6$ , которые совместно с формирователями  $DD1$ – $DD3$  формируют видеоимпульсы с парамет-

рами, необходимыми для работы элементов  $DD4$ ,  $DD5$ . 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  $DD6$ .

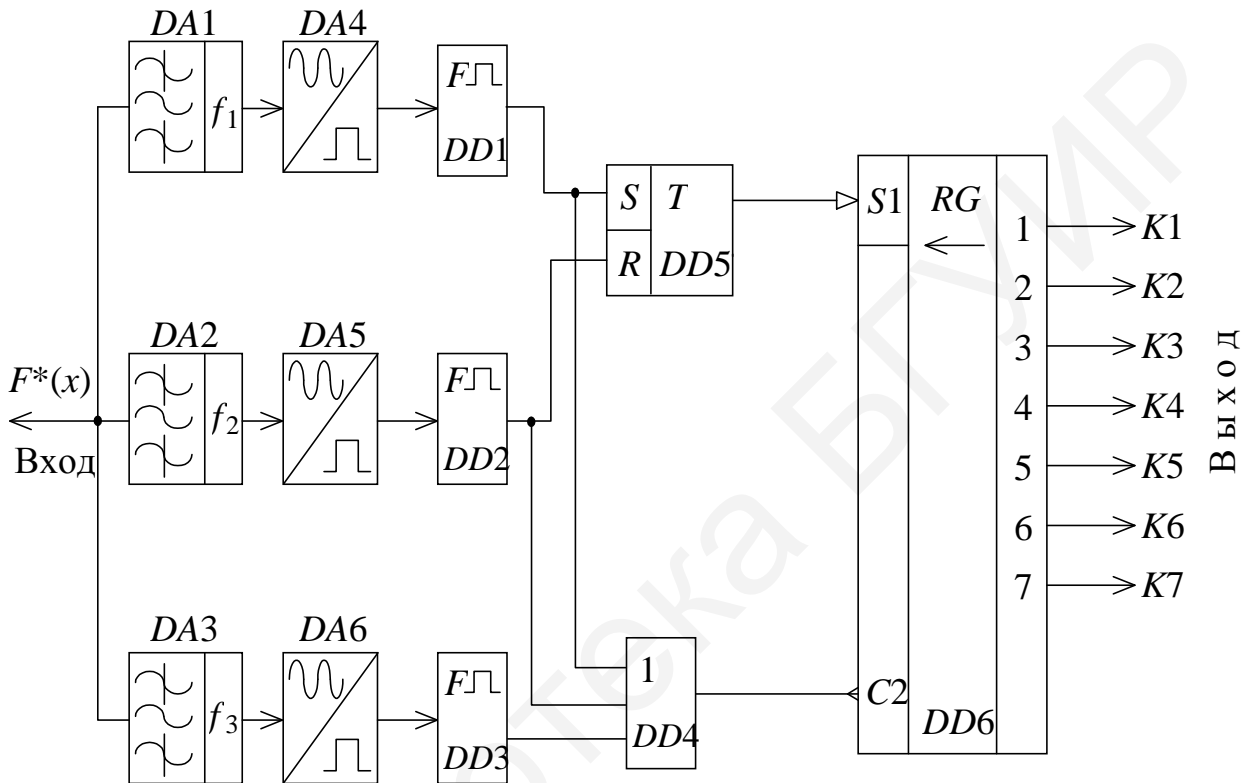


Рисунок 10.10 – Декодер сменно-качественного кода

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, представленных на рисунке 10.11.



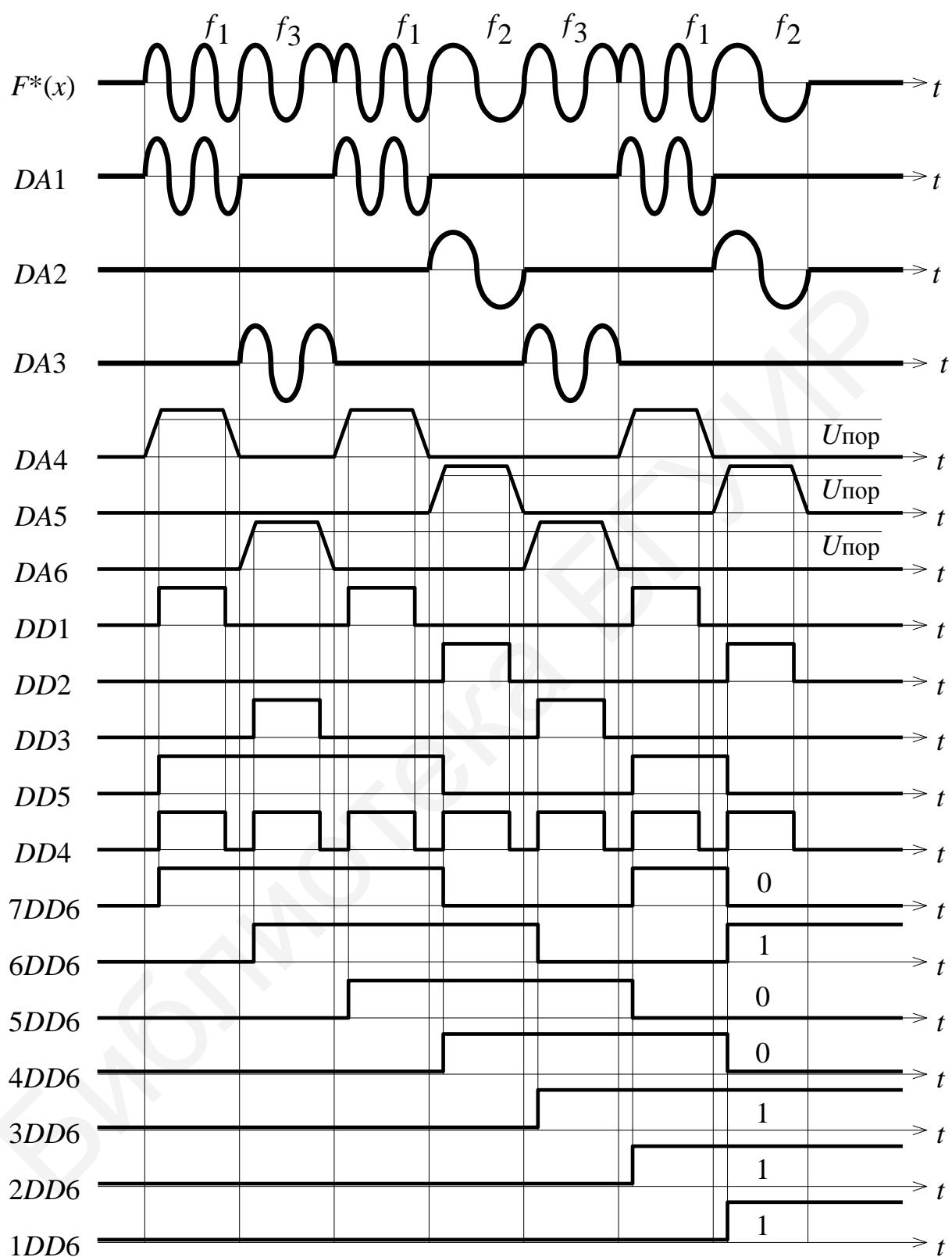


Рисунок 10.11 – Временные диаграммы работы декодера сменно-качественного кода

## ПРИЛОЖЕНИЕ В

(справочное)

### НЕПРИВОДИМЫЕ МНОГОЧЛЕНЫ И ИХ ЭКВИВАЛЕНТЫ

Сте- пень	Многочлен	Двоичная последова- тельность	Сте- пень	Многочлен	Двоичная последова- тельность
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
6	$x^6+x+1$	1000011		$x^8+x^7+x^3+x+1$	110001011
	$x^6+x^3+1$	1001001	$x^8+x^7+x^5+x^3+1$	110101001	
	$x^6+x^4+x^2+x+1$	1010111	9	$x^9+x+1$	1000000011
	$x^6+x^4+x^3+x+1$	1011011		$x^9+x^4+1$	1000010001
	$x^6+x^5+1$	1100001		$x^9+x^4+x^2+x+1$	1000010111
	$x^6+x^5+x^2+x+1$	1100111		$x^9+x^4+x^3+x+1$	1000011011
	$x^6+x^5+x^3+x^2+1$	1101101		$x^9+x^5+x^4+x+1$	1000110011
	$x^6+x^5+x^4+x+1$	1110011		$x^9+x^6+x^5+x^2+1$	1001100101
	$x^6+x^5+x^4+x^2+1$	1110101		10	$x^{10}+x^3+1$

## ПРИЛОЖЕНИЕ Г

(справочное)

### 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	

**ПРИЛОЖЕНИЕ Д**  
(справочное)

**ПАРАМЕТРЫ ЦИКЛИЧЕСКИХ КОДОВ БЧХ**

$n$	$k$	$s$	$r$	Образующий многочлен $P(x)$
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

## ПРИЛОЖЕНИЕ Е

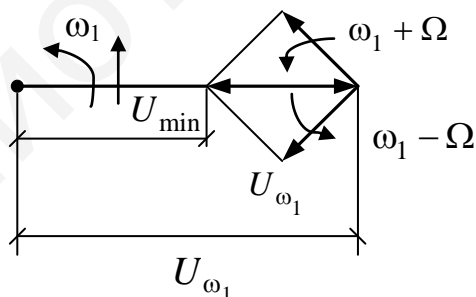
(справочное)

### ЭКСПРЕСС-ТЕСТЫ ДЛЯ ПРОВЕРКИ УСВОЕНИЯ ЛЕКЦИОННОГО МАТЕРИАЛА

Указание. На каждый вопрос теста необходимо отвечать в виде: ДА, НЕТ, НЕ ЗНАЮ. За правильный ответ дается +1 балл, за неверный – минус 1 балл, за НЕ ЗНАЮ – ноль баллов. Сумма баллов и является оценкой по десятибалльной системе за тест.

#### Тест №1

1. Можно ли использовать телемеханическую систему для передачи дискретных команд по линии связи от автомата к автомату?
2. Включает ли система телерегулирования устройства ТИ и ТУ?
3. Можно ли отнести систему, предназначенную для передачи дискретных команд по каналу связи с ПУ на КП, к числу телемеханических?
4. Включает ли понятие «информация» новизну, содержащуюся в сообщении?
5. Может ли быть крутизна характеристики амплитудного модулятора больше единицы?
6. Соответствует ли векторная диаграмма сигналу АМ?



7. Справедливо ли утверждение, что сигнал ОАМ занимает полосу частот, равную  $\Delta\omega = \Omega$ ?
8. Присутствует ли паразитная фазовая модуляция при АМ, если  $m_{\text{АМ}} > 1$ ?
9. Можно ли систему ТУ отнести к категории I3 по помехоустойчивости, если вероятность трансформации команды ТУ составляет  $10^{-12}$ , при  $P_1 = 10^{-4}$ ?
10. Можно ли систему ТИ отнести к категории R3 по безотказности, если среднее время между отказами 4500 часов?

## Тест №2

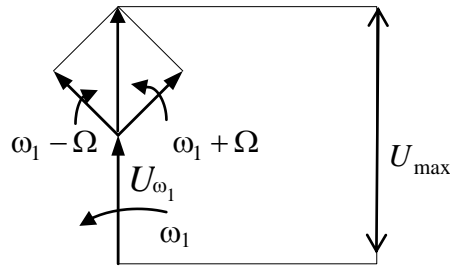
1. Относится ли система к системе телемеханики, если в ней допускается режим запаздывания при получении информации?
2. Включает ли функция телеконтроля функции измерения и управления?
3. Относится ли АМ-сигнал к почти периодическим сигналам?
4. Можно ли отнести систему к числу телемеханических систем, у которых передача дискретных команд от автомата к автомату осуществляется средствами местной автоматики?
5. Справедливо ли наше утверждение, что сигнал ОАМ занимает полосу частот  $\Delta\omega = 2\Omega_{\max}$  ?
6. Можно ли индекс ФМ-сигнала измерять в радианах?
7. Может ли индекс частотной модуляции быть больше 10?
8. Можно ли использовать источники переменного напряжения для питания устройств телемеханики, если напряжение на его выходе изменяется в пределах  $\pm 20\%$ ?
9. Могут ли значения двоичных сигналов ограничиваться номинальным напряжением постоянного тока 6 В?
10. Можно ли сигнал назвать активным, если источник питания находится внутри устройства телемеханики?

## Тест №3

1. Можно ли отнести к телемеханике область науки и техники, занимающуюся автоматическим управлением подвижными объектами средствами местной автоматики?
2. Связана ли эффективность систем телемеханики с передачей информации с малыми искажениями?
3. Можно ли отнести систему к системе телемеханики, если человек присутствует на обеих сторонах, но на одной из сторон человек является объектом исследования?
4. Относятся ли подвижные объекты к числу сосредоточенных?
5. Можно ли отнести конфигурацию линии связи к радиальной многоточечной, если все датчики подключены к контролируемому пункту отдельными проводами?
6. Можно ли отнести систему телемеханики к контролируемому классу I2, если  $P_{\text{н.ош}}$  равно  $\pm 10^{-10}$  при вероятности искажения бита  $10^{-3}$ ?
7. Можно ли систему телемеханики отнести к классу А3 по точности, если только методическая погрешность составляет 1,0 %?
8. Можно ли использовать систему телемеханики, предназначенную для работы на открытом воздухе, в помещениях с кондиционированием?
9. Можно ли отнести сигнал к числу активных, если источник питания находится вне устройства телемеханики?
10. Может ли периодическая передача инициироваться контролируемым пунктом?

### Тест №4

1. Соответствует ли векторная диаграмма ЧМ-сигналу при  $m_{\text{ЧМ}} \ll 1$ ?



2. Возможно ли передать без искажений ЧМ-сигнал по каналу связи, имеющему полосу частот  $F_k = 400$  Гц, если  $m_{\text{ЧМ}} = 3,5$ , а частота модулирующего сообщения  $F_c = 50$  Гц?

3. Можно ли по выражению  $\Delta F = 2(F_{\text{П}} + M_{\text{ЧМ}}F_c)$  определить полосу частот для АМ-ЧМ сигнала?

4. Можно ли восстановить без искажений исходное сообщение, передаваемое методом АМ, если полоса частот канала связи  $F_k = 150$  Гц, частота модулирующего сообщения  $F_c = 10$  Гц, а коэффициент глубины модуляции  $m_{\text{АМ}} = 1,1$ ?

5. Можно ли применять модуляцию в многоканальных кабельных линиях связи?

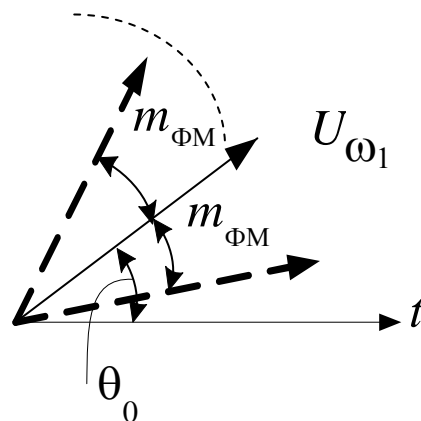
6. Достигает ли при АМ максимальная мощность за период модулирующего сообщения четырехкратного значения мощности молчания?

7. Возможно ли передать без искажений АМ-ЧМ сигнал по каналу связи с полосой частот  $F_k = 12000$  Гц, если  $M_{\text{ЧМ}} = 5$ , частота поднесущей  $F_{\text{П}} = 1000$  Гц,  $F_c = 100$  Гц?

8. Справедливо ли выражение  $\Delta F = 2(M_{\text{ЧМ}}F_{\text{П}} + m_{\text{ЧМ}}F_c)$  для определения полосы частот, занимаемой ЧМ-ЧМ сигналом?

9. Можно ли увеличить амплитуду полезной составляющей при однополосной амплитудной модуляции в два раза?

10. Соответствует ли векторное представление ФМ-сигналу?



### Тест №5

1. Справедливо ли выражение  $A_k = \frac{24}{k\pi} \sin \frac{k\pi u}{T}$  для определения амплитуды  $k$ -й гармоники периодической последовательности прямоугольных импульсов?
2. Будет ли присутствовать 11-я гармоника в спектре периодической последовательности прямоугольных импульсов, если скважность  $Q = 5,5$ ?
3. Можно ли с помощью полосового фильтра получить АМ-сигнал из АИМ-сигнала?
4. Возможно ли выделить из спектра АИМ неискаженную составляющую модулирующего сообщения с частотой  $\Omega$ , если частота следования импульсов поднесущей (несущей) будет  $\omega_1 < 2\Omega$ ?
5. Зависят ли амплитуды гармонических составляющих вида  $k\omega_1$  при АИМ-I от  $m_{\text{АИМ}}$ ?
6. Зависят ли амплитуды гармонических составляющих вида  $k\omega_1 \pm \Omega$  при АИМ-II от частоты модулирующего сообщения  $\Omega$ ?
7. Может ли индекс ФИМ-сигнала быть больше единицы?
8. Может ли индекс ЧМП-сигнала быть больше 10?
9. Справедливо ли наше утверждение, что амплитуда АМ-сигнала  $U_{\text{АМ}} = 5(1 + 0,7 \cos 50t) \cos 960t$  за период модулирующего сообщения изменяется от 0 до 10 В?
10. Справедливо ли наше утверждение, что ЧМ-сигнал  $U_{\text{ЧМ}} = 5 \cos(1000\pi t + 4 \sin 20\pi t)$  имеет практическую полосу частот, равную  $\Delta F = 100$  Гц?

### Тест №6

1. Можно ли непрерывную функцию  $f(t)$ , не содержащую частот выше  $F_{\text{max}}$ , восстановить дискретными отсчетами, взятыми через интервал времени  $\Delta t = \frac{1}{3F_{\text{max}}}$ ?
2. Справедливы ли наши утверждения, что свойства ряда Котельникова  $f(t) = \sum_{k=-\infty}^{\infty} f(k\Delta t) \frac{\sin 2\pi F_{\text{max}}(t - k\Delta t)}{2\pi F_{\text{max}}(t - k\Delta t)}$  основываются на свойстве функции  $\frac{\sin x}{x}$ ?
3. Соответствует ли выражение  $U(t) = U_{\omega_1} (\cos \omega_1 t \cos \Omega t - \sin \omega_1 t \sin \Omega t)$  однополосному АМ-сигналу при передаче верхней боковой составляющей?
4. Можно ли по выражению  $\varphi(t) = \omega_1 t + k_{\text{ЧМ}} \int c(t) dt$  определить полную фазу ЧМ-сигнала?
5. Зависит ли ширина спектра ФМ-сигнала от амплитуды и частоты модулирующего сообщения?



6. Справедливо ли наше утверждение, что спектры ФИМ- и ШИМ-сигналов по своему частотному составу совпадают?

7. Можно ли по выражению  $\Delta\omega = \frac{2\pi}{\tau}$  определить практическую полосу частот ШИМ-сигнала?

8. Справедливо ли наше утверждение, что при уменьшении коэффициента глубины амплитудной манипуляции энергия несущей возрастает, а энергия боковых полос падает?

9. Справедливо ли, что при АМП-амплитуда составляющей на несущей частоте вписывается в огибающую спектра только при  $m_{\text{АМП}} = 1$ ?

10. Всегда ли форма спектра АМП-сигнала симметрична относительно несущей частоты?

### Тест №7

1. Можно ли передать без искажений ЧМ-АМ сигнал по каналу связи, имеющему полосу частот  $\Delta F_{\text{КС}} = 2,1$  кГц, если  $F_{\text{П}} = 1$  кГц, а девиация частоты поднесущей  $F_{\text{ДП}} = 100$  Гц?

2. Справедливо ли наше утверждение, что в выражении для спада импульса при ФИМ  $\tau_2 = \tau/2 + \Delta\tau \sin \Omega(t - \tau)$  время  $t$  заменено на  $t - \tau$  для того, чтобы учесть смещения центра импульса относительно тактовой точки?

3. Определяет ли выражение  $\Delta\omega = \frac{2\pi\tau}{\tau_{\text{min}}}$  полосу частот, занимаемую ФИМ-сигналом?

4. Справедливо ли наше утверждение, что при ШИМ-II индекс  $m_{\text{ШИМ}}$  уменьшается в два раза по сравнению с ШИМ-I?

5. Является ли спектр радиоимпульса симметричным относительно составляющей на несущей частоте?

6. Справедливо ли наше утверждение, что амплитуды гармонических составляющих радиоимпульса в два раза больше соответствующих гармонических составляющих видеосигнала?

7. Верно ли наше утверждение, что спектр ЧМП-сигнала становится несимметричным относительно несущей частоты при скважности, отличной от двух?

8. Можно ли по выражению  $\Delta\omega = 2(m+1)\Omega$  определить полосу частот ЧМП-сигнала?

9. Справедливо ли, что при индексах ЧМП-сигнала, близких к единице, основная энергия содержится в несущей и двух первых боковых гармонических составляющих?

10. Справедливо ли, что при ФМП на  $\Delta\phi = 180^\circ$  и скважности  $Q = 3$  в спектре отсутствует составляющая на несущей частоте?

## Тест №8

1. Справедливо ли, что в цифровых системах преобразование последовательности битов в последовательность электрических импульсов осуществляется по правилу:  $0 \longrightarrow -b$ ,  $1 \longrightarrow +b$ ?
2. Действительно ли символьная скорость больше битовой?
3. Измеряется ли битовая скорость в бодах?
4. Можно ли модулированный радиосигнал представить в виде двух гармонических колебаний, отличающихся начальным фазовым сдвигом, равным  $180^\circ$ ?
5. Можно ли совокупность состояний сигнала назвать диаграммой фазовых переходов?
6. Зависит ли полоса частот АМП-сигнала от коэффициента глубины манипуляций?
7. Действительно ли синфазная составляющая при АМП равна нулю?
8. Справедливо ли наше утверждение, что элемент задержки в структурной схеме АМП манипулятора преобразует параллельный код в последовательный?
9. Действительно ли ЦАП в схеме амплитудного манипулятора служит для формирования амплитуды квадратурной составляющей?
10. Может ли амплитуда сигнала на выходе перемножителя устройства формирования М-АМП иметь нулевое значение?

## Тест №9

1. Действительно ли изменение фазы при ОФМП всегда указывает на передачу двоичной единицы?
2. Применяется ли двоичная фазовая манипуляция при формировании широкополосных сигналов?
3. Возможна ли межсимвольная интерференция при уменьшении полосы частот радиосигнала?
4. Присутствует ли синфазная составляющая в М-ичной ФМП ?
5. Может ли начальная фаза при М-ичной ФМП равняться  $2\pi$  ?
6. Верно ли, что к ФМП-4 действительная часть комплексного бита есть бит четной последовательности?
7. Можно ли избавиться от изменения фазы на  $180^\circ$  при квадратурной ФМП без смещения?
8. Играет ли роль способ демодуляции при сигнальном кодировании?
9. Имеет ли в своем составе сигнальный кадер КФМП-8 аналого-цифровой преобразователь?
10. Можно ли с помощью  $\pi/4$  КФМП осуществить абсолютную манипуляцию?

## Тест №10

1. Будет ли наблюдаться разрыв фазы ЧМП-сигнала при применении независимых генераторов?
2. Можно ли ЧМП-сигнал представить в виде суммы двух АМП-сигналов?
3. Можно ли использовать схемы логического умножения в качестве ключа в частотном манипуляторе с разрывом фазы?
4. Приводит ли разрыв фазы ЧМП-сигнала к расширению полосы частот?
5. Присутствует ли разрыв фазы в генераторе, управляемом напряжением?
6. Справедливо ли выражение для индекса  $m_{\text{ЧМП}} = (F_{\text{max}} - F_0)/F_c$ , если модулирующий сигнал однополярный?
7. Верно ли, что при ЧМП переходная фаза меняется по пилообразному закону?
8. Можно непериодические функции  $f_1(t) = \cos \psi$  и  $f_2(t) = \sin \psi$  разложить в ряды Фурье?
9. Имеют ли одинаковое значение амплитуды гармонических составляющих на частотах  $\omega_1 \pm k\Omega$  для ЧМП и ЧМ?
10. Будет ли спектр ЧМП-сигнала симметричным относительно несущей частоты, если скважность модулирующего сообщения не равна двум?

## Тест №11

1. Можно ли при затворной модуляции осуществить амплитудную модуляцию, если используется линейный участок вольт-амперной характеристики полевого транзистора, т. е.  $a_2 = 0$ ?
2. Может ли конденсатор  $C_1$  в схеме затворной модуляции иметь большое сопротивление для тока несущей частоты  $\omega_1$  и малое – для тока частоты модулирующего сообщения  $\Omega$ ?
3. Справедливо ли утверждение, что в схеме затворного модулятора от источника модулирующего напряжения потребляется меньшая мощность, чем в схеме стокового модулятора?
4. Можно ли осуществить базовую модуляцию путем подачи модулирующего напряжения на базу или эмиттер биполярного транзистора?
5. Справедливо ли утверждение, что в балансном модуляторе ОАМ напряжение несущей частоты подается на затворы транзисторов  $VT_1$  и  $VT_2$  противофазно, а модулирующее напряжение – синфазно?
6. Можно ли при однополосной амплитудной модуляции в заданной полосе частот разместить вдвое большее количество передатчиков?
7. Можно ли каскад на полевом транзисторе назвать «реактивным», если фаза сигнала обратной связи поворачивается на  $180^\circ$  или  $360^\circ$ ?
8. Можно ли осуществить фазовую модуляцию с помощью реактивного биполярного транзистора?

9. Можно ли при фазовом методе получения сигнала с одной боковой полосой получить верхнюю боковую полосу на частоте  $\omega_1 + \Omega$  путем изменения фазы несущего колебания на входе БМ1 или БМ2 на  $180^\circ$ ?

10. Устраняется ли паразитное изменение амплитуды входного сигнала в дробном детекторе?

### Тест №12

1. Зависит ли емкость варикапа от напряжения смещения при отсутствии полезного сообщения?

2. Зависит ли индуктивность обмотки с ферромагнитным сердечником на переменном токе от высоты сердечника?

3. Справедливо ли утверждение, что каскад на транзисторе  $VT2$  частотно-модулятора с управляемой индуктивностью предназначен для термостабилизации?

4. Можно ли получить изменение частоты в цепочечном RC-генераторе путем изменения емкости одного из конденсаторов по закону модулирующего сообщения?

5. Справедливо ли утверждение, что в частотном дискриминаторе на расстроенном контуре сопротивление нагрузки должно удовлетворять условиям  $1/\omega_1 C \ll R < 1/\Omega C$  ?

6. Присутствует ли постоянное напряжение на выходе дискриминатора с одиночным контуром при отсутствии полезного сообщения?

7. Устраняется ли паразитная амплитудная модуляция в частотном дискриминаторе на связанных контурах?

8. Зависит ли полярность выходного напряжения дробного детектора от девиации частоты?

9. Можно ли из сигнала ФИМ получить сигнал ФМ?

10. Справедливо ли наше утверждение, что буферный усилитель в схеме многоканального амплитудно-импульсного модулятора имеет малое входное и большое выходное сопротивления?

### Тест №13

1. Можно ли назвать диаграммой фазовых переходов графическое изображение траекторий перемещения сигнальных точек в сигнальном созвездии?

2. Верно ли наше предположение, что спектр ЧМП-сигнала становится несимметричным относительно составляющей на несущей частоте при скважности модулирующего сообщения, отличной от двух?

3. Можно ли передать без искажений ЧМ-АМ сигнал по каналу связи, имеющему полосу частот  $\Delta F=2,1$  кГц, если  $F_{\text{п}}=1$  кГц, а девиация частоты поднесущей  $F_{\text{г}}=100$  Гц?

4. Справедливо ли наше утверждение, что кодовое расстояние – это число единиц, содержащееся в кодовой комбинации?

5. Может ли степень частного при делении равняться степени делителя, если деление производится с помощью сумматоров по  $M2$ ?

6. Может ли при сложении по  $M2$  нескольких чисел сумма выражаться меньшим двоичным числом, чем какое-либо из слагаемых?

7. Означает ли умножение многочлена  $G(x)$  на  $x^n$  приписывание справа к многочлену  $n$  нулей?

8. Справедливо ли наше утверждение, что если вес остатка больше числа исправляемых ошибок, то принятую кодовую комбинацию складывают по модулю два с остатком и получают исправленную кодовую комбинацию?

9. Можно ли определить номер искаженного разряда в циклическом коде по весу синдрома?

10. Можно ли составить проверку для конкретного разряда при мажоритарном декодировании кодовых комбинаций в циклическом коде на основании некоторой матрицы  $L$ , в которой все столбцы содержат по одному единичному элементу?

### Тест №14

1. Можно ли четырехразрядным двоичным кодом на все сочетания записать десятичное число 16?

2. Равноценны ли записи одного и того же числа 01101 в двоичном коде и 01011 в коде Грея?

3. Могут ли коды с  $d = 3$  обнаруживать двойные и исправлять единичные ошибки?

4. Соответствует ли число 0110, записанное в коде Айкена 2- 4- 2-1 десятичному числу 6?

5. Можно ли в коде Грея 7-3-1 записать число 10?

6. Обнаруживает ли код с постоянным весом все четырехкратные ошибки?

7. Относится ли комбинация 01010001 к числу разрешенных в коде с числом единиц, кратным трем?

8. Зависит ли избыточность корреляционного кода от числа информационных разрядов?

9. Верна ли запись десятичного числа 57 в коде Грея 100101?

10. Результат проверки принятой комбинации, закодированной инверсным кодом: 001000. Верны ли наши утверждения, что искажен третий информационный разряд?

### Тест №15

1. Определяет ли число ненулевых элементов вес кодовой комбинации?

2. Характеризует ли единичная матрица циклического кода размерностью  $k \times k$  комбинации простого кода?

3. Результат проверки принятой комбинации в коде Хемминга показал, что  $S_i = 1, S_\Sigma = 1$ . Соответствует ли это двойному искажению?

4. Может ли степень частного для циклического кода равняться степени исходной кодовой комбинации?

5. Может ли циклический код  $d \geq 5$  иметь четное число разрядов?

6. Может ли вес строки матрицы дополнений циклического кода с  $d = 3$  быть больше, чем  $d+1$ ?

7. Может ли остаток при кодировании в циклическом коде иметь степень, равную степени полинома  $P(X)$ ?

8. Характеризует ли количество проверочных символов корректирующие свойства кода Хемминга при одном и том же числе информационных символов?

9. Можно ли рассматривать циклический код с  $d = 2$  как обычный код с четным числом единиц?

10. Может ли циклический код, образованный двухчленным  $P(X) = x+1$ , обнаруживать нечетные ошибки любой кратности?

### Тест №16

1. Определяет ли кодовое расстояние  $d$  число ненулевых членов образующего многочлена  $P(X)$ ?

2. Может ли вес остатка, используемый для нахождения матрицы дополнений, быть равным числу исправляемых ошибок?

3. Может ли быть число разрядов дополнительной матрицы меньше числа контрольных символов?

4. Справедливо ли равенство  $P(X)_{d=3} = (x+1)P(X)_{d=2}$ ?

5. Справедливо ли равенство  $\frac{2^h + 1}{q} = n$  для определения длины слова по методу БЧХ?

6. Может ли быть число минимальных членов равным кратности исправления  $L = S$ ?

7. Может ли степень образующего многочлена быть больше, чем произведения старшей степени на кратность исправления, т. е.  $\beta > tS$ ?

8. Может ли код Файра исправить пакет ошибок длиной  $b = 5$ , если искажены второй и пятый разряды?

9. Может ли код по закону перестановок обнаруживать одиночные ошибки?

10. Может ли для оптимальных кодов средняя длина кодового слова быть больше, чем  $L \geq \frac{H(X)}{\log m}$ ?

### Тест №17

1. Справедливо ли наше утверждение, что код Хемминга с  $d = 3$  исправляет одиночные и обнаруживает двойные ошибки?

2. Зависит ли выбор образующего многочлена  $P(X)$  от кодового расстояния?

3. Справедлива ли запись числа  $45_{10}=01001011$  в двоично-десятичном коде Айкена?

4. Результат принятой кодовой комбинации в коде Хемминга с  $d = 4$  показал, что  $S_i \neq 0$  и  $S_\Sigma = 0$ . Соответствует ли это одиночной ошибке?

5. Справедливо ли, что для составления образующей матрицы в циклическом коде берут единичную матрицу размерностью  $k \times k$  и справа приписывают к ней дополнительную матрицу размерностью  $k \times r$ ?

6. Справедливо ли наше утверждение, что для построения кодов БЧХ с помощью таблиц запись образующего многочлена в виде десятичных цифр преобразуется путем перевода каждой десятичной цифры в трехразрядное двоичное число?

7. Может ли применяться код БЧХ для обнаружения ошибок?

8. Может ли код Файра исправить пакет ошибок длиной  $b_s = 5$ , если искажены второй и пятый разряды?

9. Может ли для оптимальных кодов средняя длина кодового слова быть больше энтропии сообщений?

10. Справедливо ли наше утверждение, что в условиях коррелированных ошибок рациональнее применить код с повторением комбинаций вместо кода с повторением каждого разряда?

## ЛИТЕРАТУРА

### К РАЗДЕЛАМ 6–10

44. Питерсон, У. Коды, исправляющие ошибки / У. Питерсон, Э. Уэлдон. – М. : Мир, 1976. – 593 с.
45. Дмитриев, В. И. Прикладная теория информации / В. И. Дмитриев. – М. : Высш. шк., 1989. – 320 с.
46. Аршинов, М. Н. Коды и математика / М. Н. Аршинов, Л. Е. Садовский. – М. : Наука, 1983. – 144 с.
47. Колесник, В. Д. Декодирование циклических кодов / В. Д. Колесник, Е. Т. Мирончиков. – М. : Связь, 1968. – 251 с.
48. Новик, А. А. Эффективное кодирование / А. А. Новик. – М. : Энергия, 1965. – 235 с.
49. Хемминг, Р. В. Теория кодирования и теория информации / Р. В. Хемминг. – М. : Радио и связь, 1985. – 176 с.
50. Тутевич, В. Н. Телемеханика / В. Н. Тутевич. – М. : Высш. шк., 1985. – 423 с.
51. Пшеничников, А. М. Телемеханические системы на интегральных микросхемах / А. М. Пшеничников, М. Л. Портнов. – М. : Энергия, 1977. – 296 с.
52. Передача дискретной информации и телеграфия / В. С. Гуров [и др.]. – М. : Связь, 1974. – 526 с.
53. Мак-Вильямс, Ф. Теория кодов исправляющих ошибки / Ф. Мак-Вильямс, Н. Дж. Слоэн. – М. : Связь, 1979. – 744 с.
54. Теория кодирования / Т. Касами [и др.]. – М. : Мир, 1978. – 576 с.
55. Берлекэмп, Э. Алгебраическая теория кодирования / Э. Берлекэмп. – М. : Мир, 1971. – 239 с.
56. Блох, Э. Л. Обобщенные каскадные коды / Э. Л. Блох, В. В. Зяблов. – М. : Связь, 1976. – 281 с.
57. Марков, А. А. Введение в теорию кодирования / А. А. Марков. – М. : Наука, 1982. – 192 с.
58. Шевкопляс, Б. В. Микропроцессорные структуры. Инженерные решения : справочник / Б. В. Шевкопляс. – М. : Радио и связь, 1990. – 512 с.
59. Цифровые интегральные микросхемы : справочник / М. Н. Богданович [и др.]. – Минск : Беларусь, 1991. – 492 с.
60. Пухальский, Г. Н. Проектирование дискретных устройств на интегральных микросхемах : справочник / Г. Н. Пухальский, Т. Я. Новосельцева. – М. : Радиосвязь, 1990. – 304 с.



61. Скляр, Б. Цифровая связь: теоретические основы и практическое применение / Б. Скляр. – М. : Изд. дом «Вильямс», 2003. – 1106 с.

62. Хелд, Г. Технологии передачи данных / Г. Хелд. – 7-е изд. – СПб. : Питер, 2003. – 720 с.

63. Угрюмов, Е. П. Цифровая схемотехника : учеб. пособие / Е. П. Угрюмов. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2007. – 800 с.

64. Джонсон, Г. Высокоскоростная передача данных / Г. Джонсон. – М. : Изд. дом «Вильямс», 2005. – 1024 с.

65. Сорока, Н. И. Телемеханика : конспект лекций. В 5 ч. Ч. 4 : Системы телемеханики / Н. И. Сорока, Г. А. Кривинченко. – Минск : БГУИР, 2005. – 153 с.

Библиотека БГУИР

*Учебное издание*

**Сорока Николай Ильич**  
**Кривинченко Георгий Александрович**

**ТЕЛЕМЕХАНИКА.**  
**МОДУЛЯЦИЯ И КОДИРОВАНИЕ ИНФОРМАЦИИ**

УЧЕБНОЕ ПОСОБИЕ

В двух частях  
Часть 2

Редактор *М. А. Зайцева*

Корректор *Е. Н. Батурчик*

Компьютерная правка, оригинал-макет *Е. Г. Бабичева*

Подписано в печать 08.05.2020. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».  
Отпечатано на ризографе. Усл. печ. л. 11,51. Уч.-изд. л. 12,2. Тираж 150 экз. Заказ 314.

Издатель и полиграфическое исполнение: учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники».

Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий №1/238 от 24.03.2014,  
№2/113 от 07.04.2014, №3/615 от 07.04.2014.

Ул. П. Бровки, 6, 220013, г. Минск



