ЛЕКЦИЯ 9. НЕПРЕРЫВНЫЕ КОДЫ

9.1 Идея построения непрерывного кода Финка–Хегельбергера

9.2. Сверточные коды

9.3. Представление сверточных кодов с помощью полиномов

9.4. Графическое представление сверточных кодов

9.5. Декодирование сверточных кодов

9.1 Идея построения непрерывного кода Финка–Хегельбергера

В этом коде последовательность кодовых символов не разделяется на отдельные блоки. В поток информационных разрядов включаются проверочные, так что между двумя информационными разрядами помещается один проверочный. Обозначая информационные разряды через a_i, а проверочные через b_i, получаем такую последовательность:

$$a_1 b_1 a_2 b_2 a_3 b_3 \ldots a_k b_k a_{k+1} b_{k+1} \ldots$$

Проверочные разряды формируются по следующему правилу:

$$b_i = a_{i-S} \oplus a_{i+S+1}, \tag{9.1}$$

где S – шаг кода (S =0,1,2…).

На приемной стороне производится проверка условия (9.1). При ошибочном приеме проверочного разряда соотношение (9.1) в принятой последовательности не будет выполнено для одного разряда b_i. В случае же ошибочного приема информационного разряда соотношение (9.1) не будет выполняться при двух значениях b_i и b_{i+2S+1}.

Тогда информативный разряд

a_{i+S+1}

должен быть заменен на противоположный.

9.2. Сверточные коды

Сверточным код называется потому, что выходная последовательность является сверткой импульсной характеристики кодера со входной последовательностью.

Сверточный кодер представляет собой устройство, воспринимающее за каждый такт работы m входных информационных разрядов, и выдающее на выход за тот же такт n выходных разрядов, подлежащих передаче по каналу связи.

Относительная скорость кода

$$R = \frac{m}{n}.$$

Основными элементами сверточного кодера являются: регистр сдвига, сумматоры по модулю 2 и коммутатор.

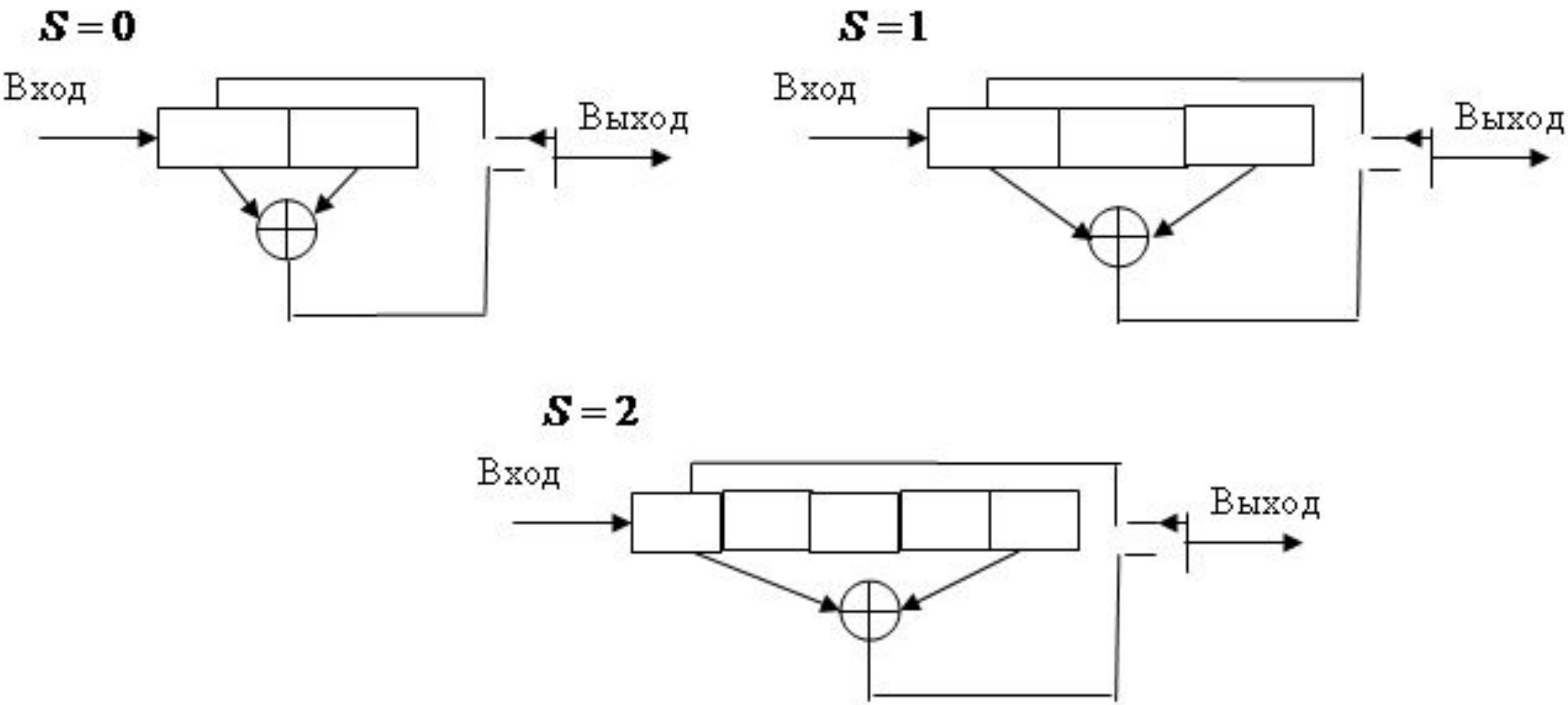


Рис. 9.1. Кодеры для кода Финка

Число ячеек l в регистре сдвига определяет память кода. Маркировка сверточного кода (n, m, l).

В момент поступления на вход регистра нового информационного разряда разряд, хранящийся в крайней правой ячейке, выводится из регистра и сбрасывается. Каждый из остальных, хранящихся в регистре разрядов перемещается на одну ячейку вправо, освобождая тем самым крайнюю левую ячейку, куда и поступает новый информационный разряд.

Сверточные коды бывают систематические и несистематические.

а)

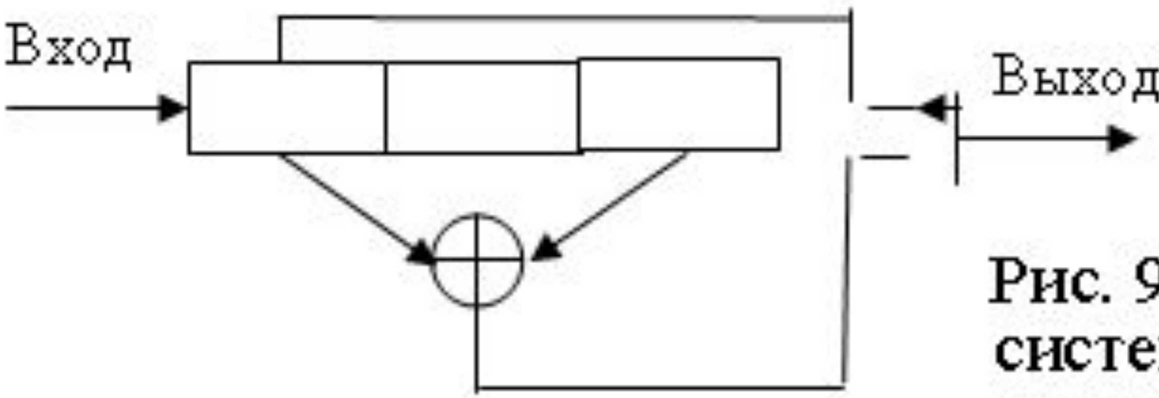
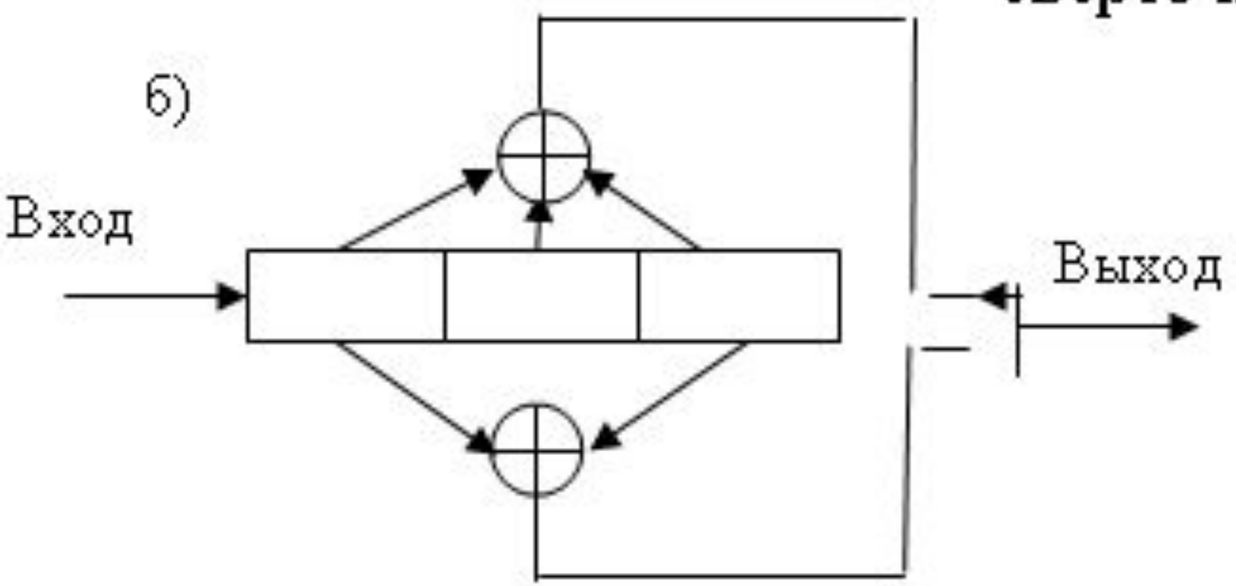


Рис. 9.2. Примеры кодеров систематического (а) и несистематического (б) сверточного кода

б)



Для того чтобы задать структуру сверточного кодера, необходимо указать, какие ячейки регистра сдвига связаны с каждым из сумматоров, счет разрядов ведется слева направо. Связи j-го сумматора описываются путем задания j-й порождающей последовательности

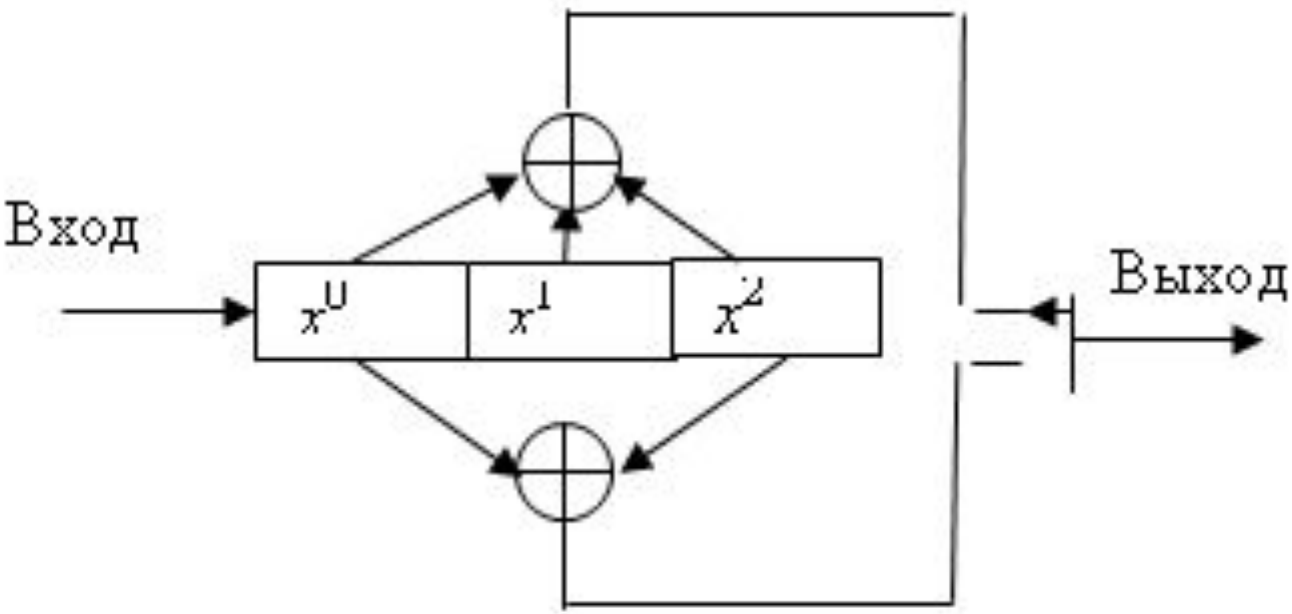
$$g_j = (g_{j0}, g_{j1}, g_{j2}, \dots, g_{jl-1}),$$

где компонента

$$g_{ji} = \begin{cases} 1, & \textit{если } i-\textit{я ячейка связана} \\ & \quad c\ j-\textit{м сумматором}, \\ 0, & \textit{в противном случае}. \end{cases}$$

9.3. Представление сверточных кодов с помощью полиномов

Сверточный код требует для своего описания несколько порождающих полиномов, число которых определяется количеством m входных и n выходных разрядов, передаваемых за каждый такт в канал связи.



Порождающие полиномы

$$g_1(x) = x^2 + x + 1$$

$$g_2(x) = x^2 + 1.$$

Рис. 9.3. Кодер сверточного кода

Порождающие полиномы могут быть объединены в матрицу размера $m \times n$ $G(x) = \left[x^2 + x + 1 \quad x^2 + 1\right]$.

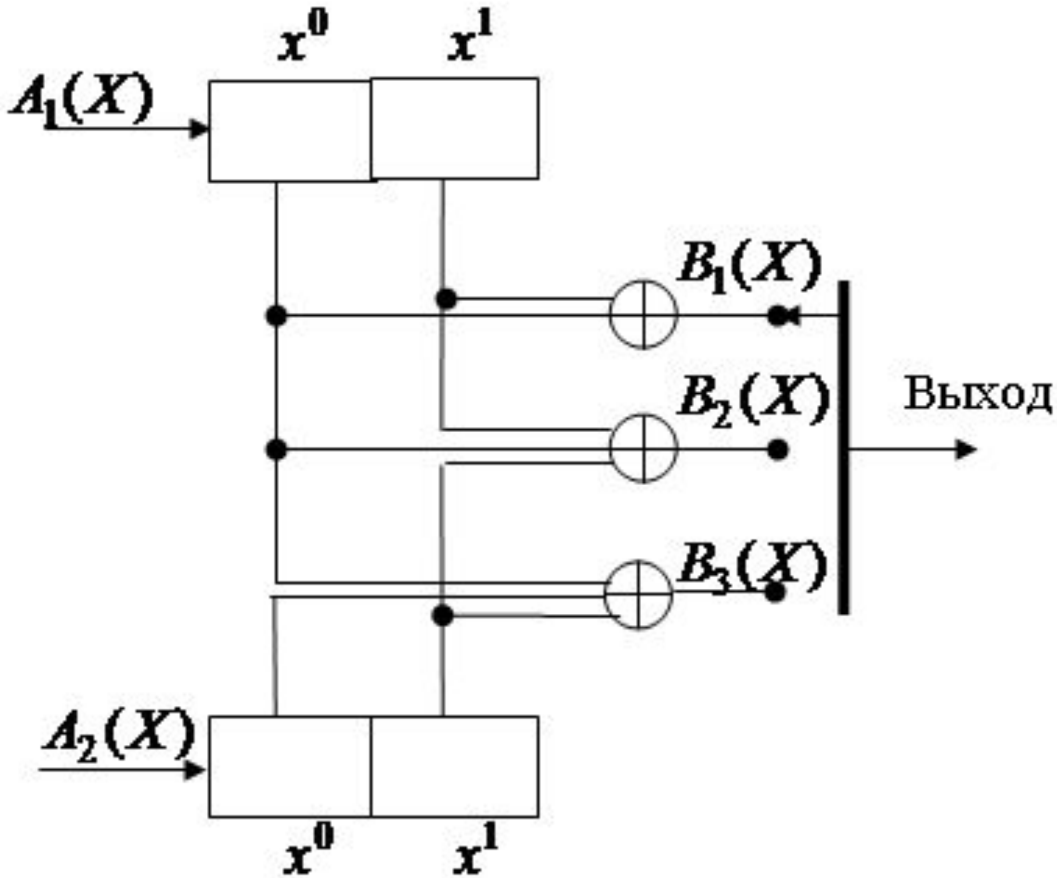


Рис. 9.4. Пример кодера со скоростью 2/3

Порождающая матрица полиномов

$$G(x) = \begin{bmatrix} x+1 & x+1 & 1 \\ 0 & x & x+1 \end{bmatrix}.$$

Пример работы кодера

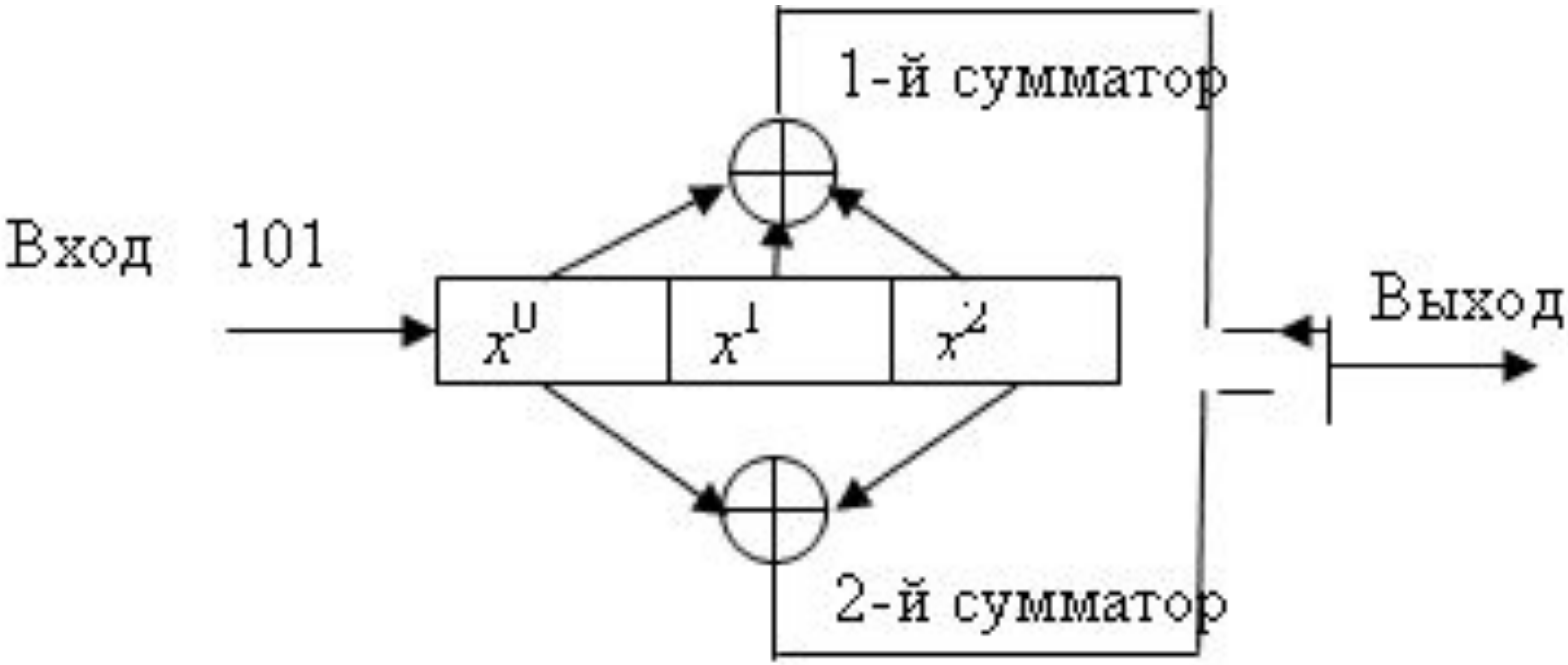


Рис. 9.5. Кодер несистематического кода

Последовательности 101 соответствует многочлен

$A(x) = x^2 + 1$.

Номер такта	Номер выхода кодера	Содержимое выхода кодера
1	1 2	$1 \oplus 0 \oplus 0 = 1$ $1 \oplus 0 = 1$
2	1 2	$0 \oplus 1 \oplus 0 = 1$ $0 \oplus 0 = 0$
3	1 2	$1 \oplus 0 \oplus 1 = 0$ $1 \oplus 1 = 0$
4	1 2	$0 \oplus 1 \oplus 0 = 1$ $0 \oplus 0 = 0$
5	1 2	$0 \oplus 0 \oplus 1 = 1$ $0 \oplus 1 = 1$

Таким образом на выходе кодера будет образована последовательность 1110001011

На выходе первого сумматора соответствует полином $B_1(x) = (x^2+1)(x^2+x+1) = x^4 + x^3 + x + 1$.

Ему соответствует последовательность 11011

На выходе второго сумматора

$B_2(x) = (x^2+1)(x^2+1) = x^4 + 1$.

Ему соответствует последовательность 10001

В итоге на выходе кодера будет сформирована последовательность $B(x)$ выходных символов за 5 тактов нахождения входной последовательности 101 в трехразрядном регистре:

$\boldsymbol{B_1(x)}$	1		1		0		1		1	
$\boldsymbol{B_2(x)}$		1		0		0		0		1
$\boldsymbol{B(x)}$	1	1	1	0	0	0	1	0	1	1
Такты	1		2		3		4		5	

9.4. Графическое представление сверточных кодов

Сверточный кодер как конечный автомат с памятью описывают диаграммой состояний. Диаграмма состояний представляет собой направленный граф, вершины которого отождествляются с возможными состояниями кодера, а ребра, помеченные стрелками, указывают возможные переходы между состояниями. Внутренними состояниями кодера считают разряды в $l-1$ левых ячейках регистра. Состояние 000...0 называется нулевым, остальные – ненулевые. Над каждым из ребер записывают кодовые символы, порождаемые кодером при соответствующем переходе из состояния в состояние.

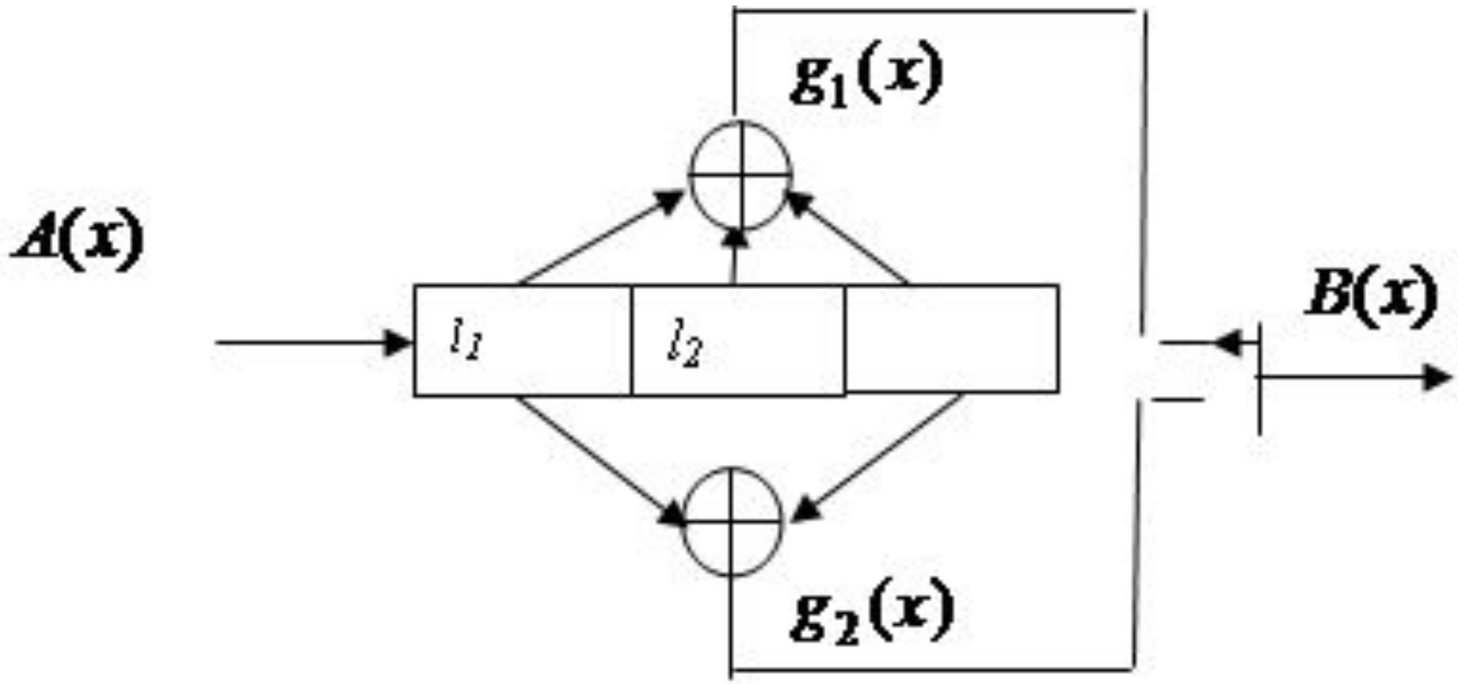


Рис. 9,6. Кодирующее устройство

Кодер может находиться в таких состояниях: l_1, l_2 - 00; 10; 11; 01. Эти состояния соответствуют вершинам графа.

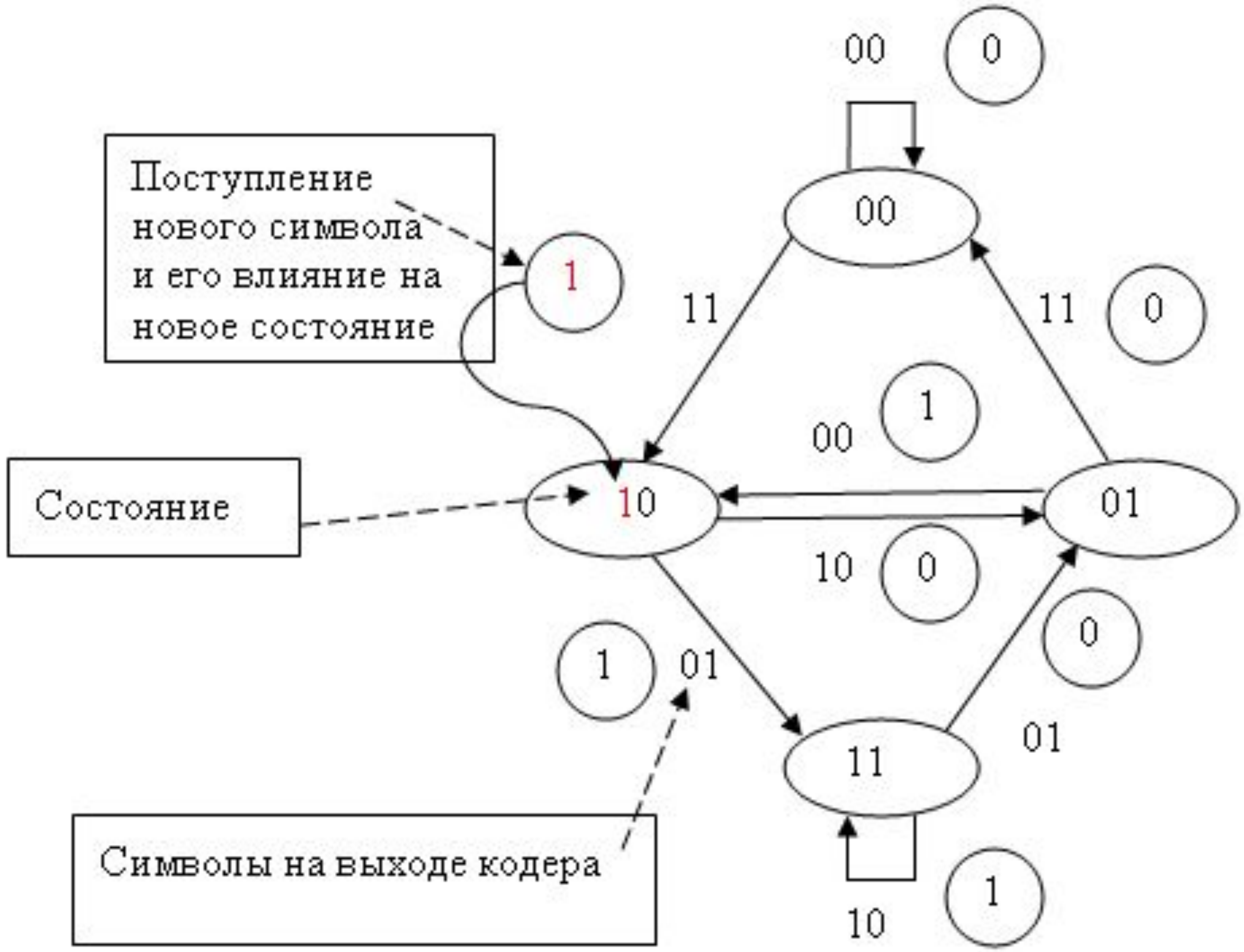


Рис 9.7. Диаграмма состояний

Диаграмму состояний можно развернуть во времени и получить решеточную (решетчатую) диаграмму.

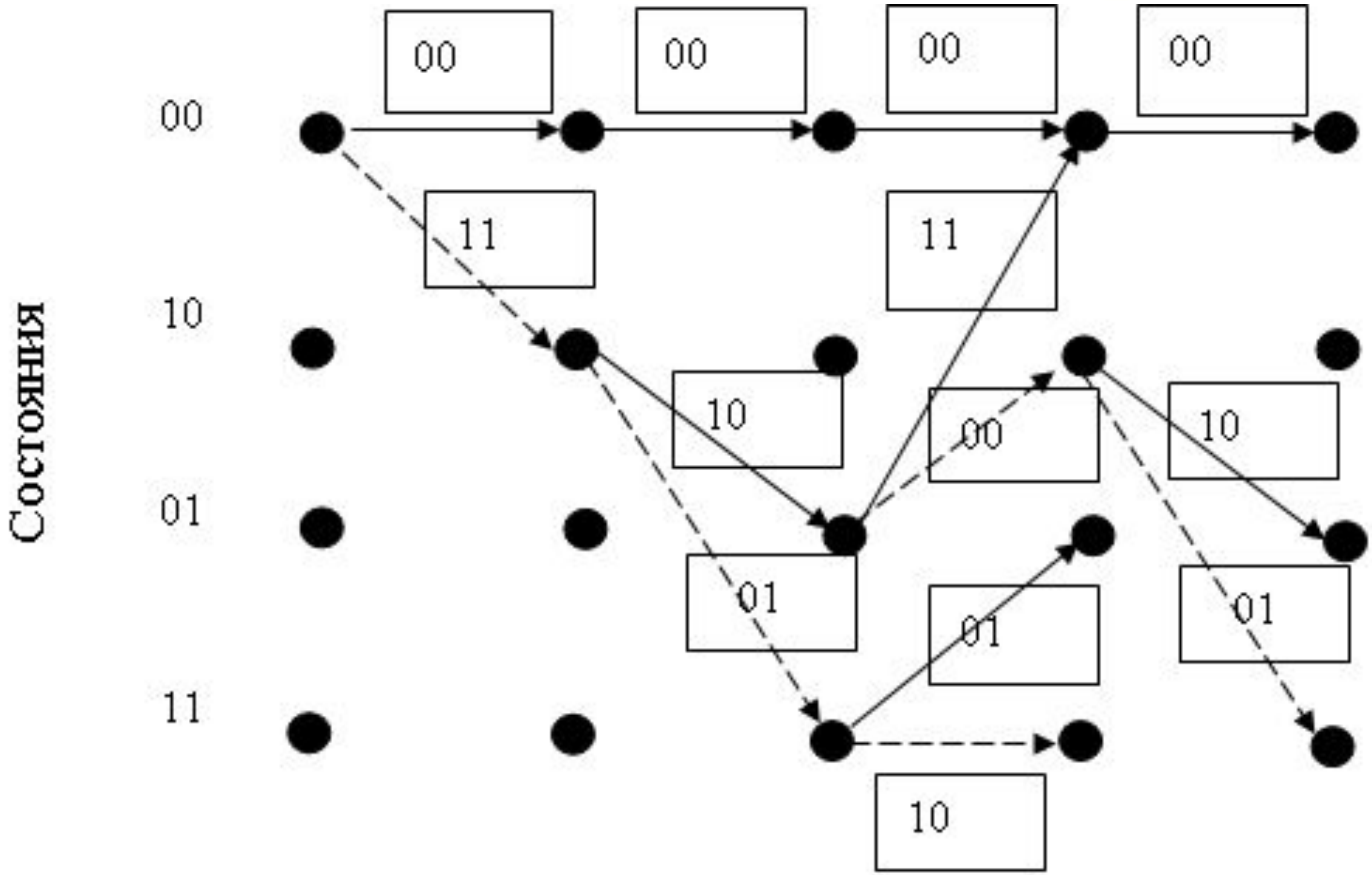


Рис.9.8. Решетчатая диаграмма

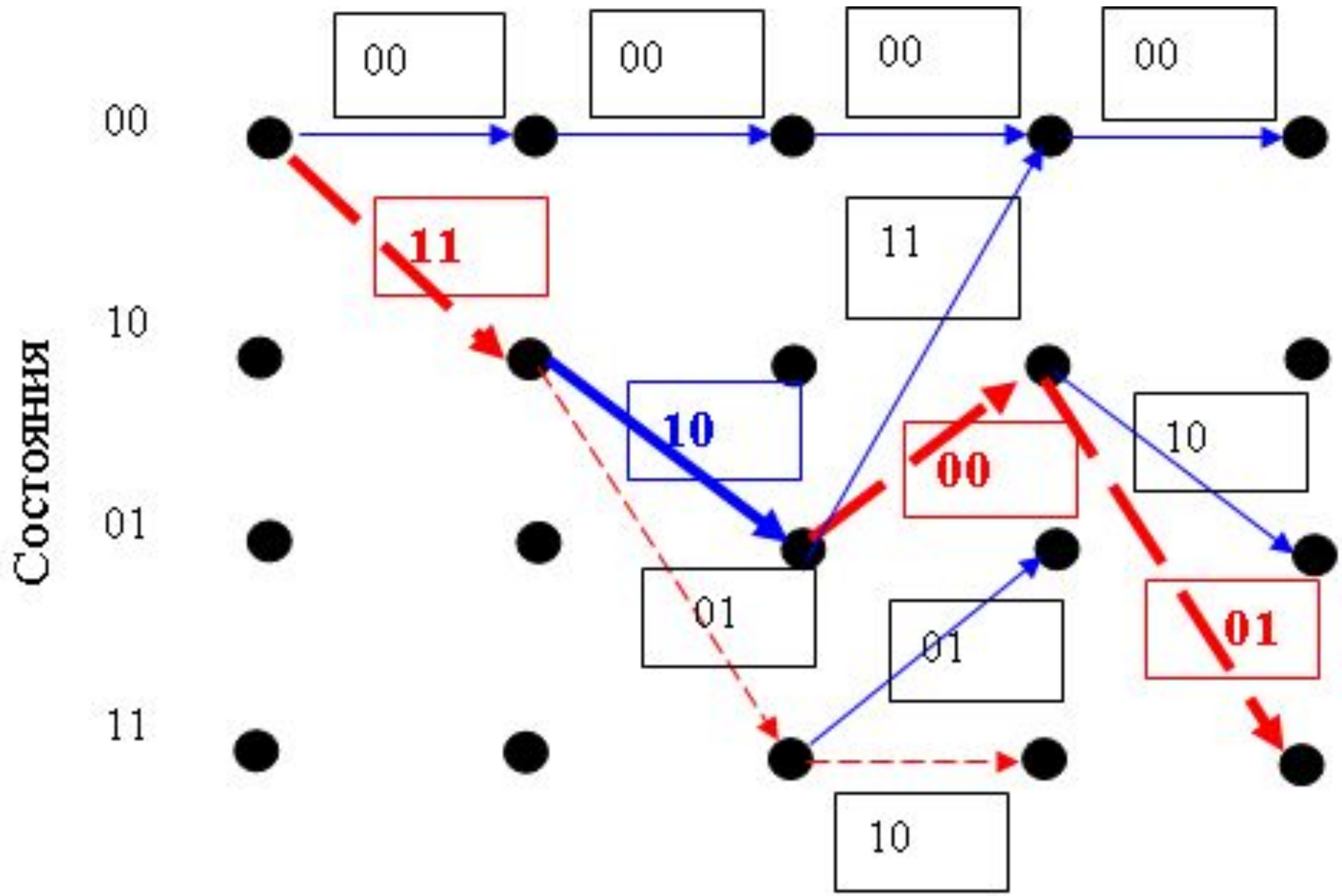


Рис.9.9. Кодирование последовательности 1011

9.5. Декодирование сверточных кодов

Метод декодирования по алгоритму Витерби

Метод представляет собой декодирование по максимуму правдоподобия. Идея алгоритма Витерби состоит в том, что в декодере воспроизводят все возможные пути последовательных изменений состояний сигнала, сопоставляя получаемые при этом кодовые символы с принятыми аналогами по каналу связи и на основе анализа ошибок между принятыми и требуемыми символами определяют оптимальный путь (оптимальной считается та последовательность, расстояние Хемминга которой от принятой последовательности минимально).

Алгоритм Витерби

1. Декодер имеет такую же решетчатую диаграмму, что и кодер. До декодирования декодер находится в нулевом состоянии.

2. Декодер получает n входных разрядов X. Находит возможные переходы из текущего состояния в состояния и считывает выходные разряды $\hat{X}$ из решетчатой диаграммы.

3. Сравнивает полученные X и ситанные $\hat{X}$ разряды и определяет метрику дуги

$$d_i = \sum_{j-1}^{n} X_j \oplus \hat{X}_{ji} .$$

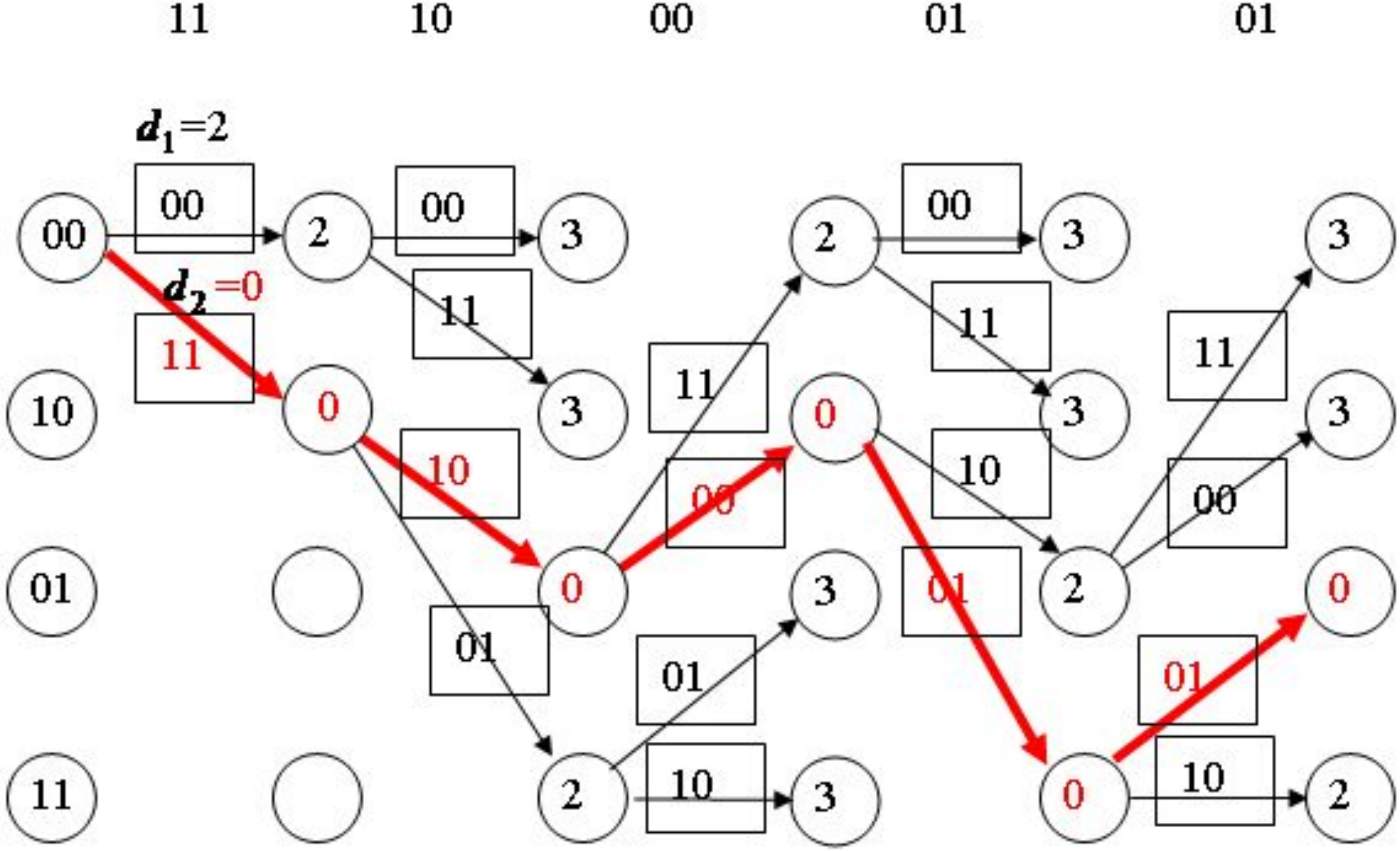
4. Рассчитывается метрика пути

$$d_{\sum} = \sum d_i .$$

5. Выбирается тот путь на решетчатой диаграмме, у которого $d_{\sum}$ минимальна.

6. Идя по выбранному пути на решетчатой диаграмме декодер выдает 1-й левый разряд состояния.

1. Декодирование в случае отсутствия ошибок при приеме



Последовательность на выходе декодера

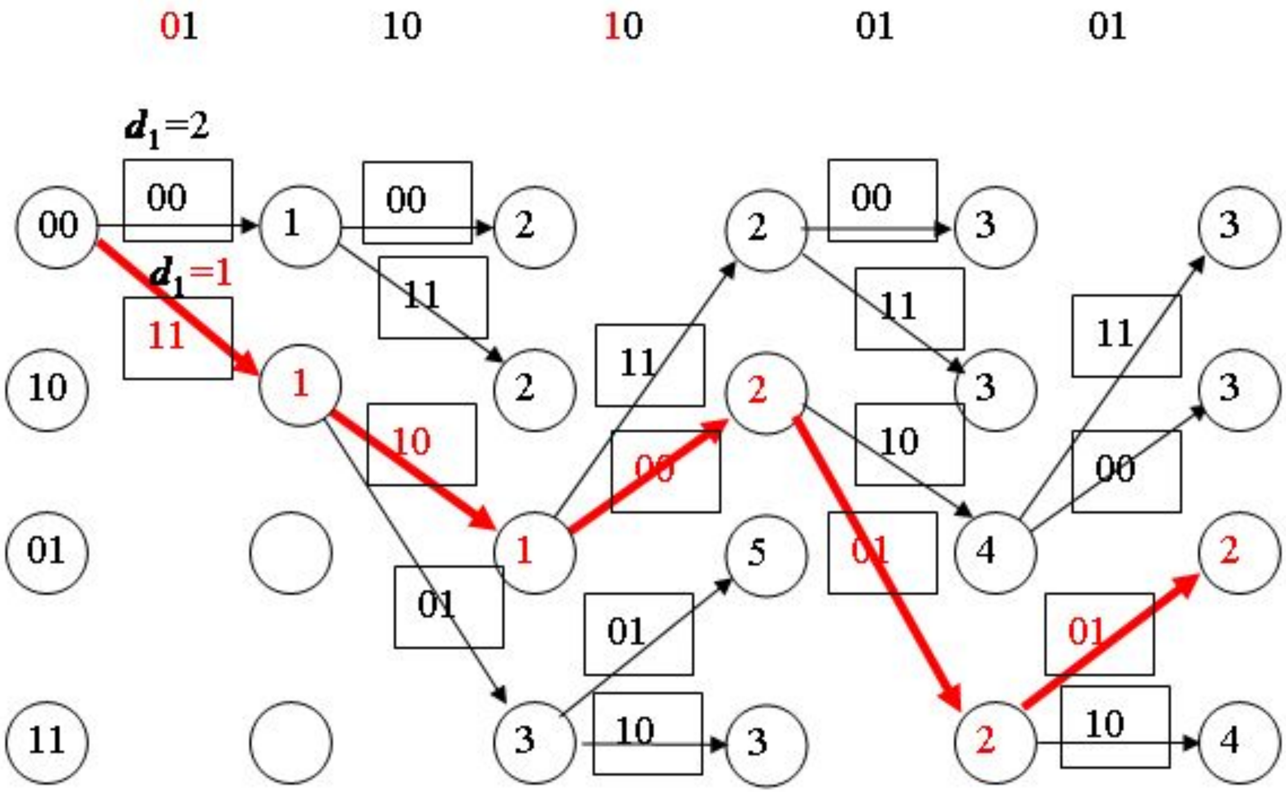
1 0 1 1 0

Рис. 9.10. Декодирование на основе алгоритма Витерби в отсутствие ошибок при приеме

2. Декодирование в случае наличия ошибок при приеме

1 1 1 0 0 0 0 1 0 1

0 1 1 0 1 0 0 1 0 1.



Последовательность на выходе дкеодера

1 0 1 1 0

Рис. 9.11. Декодирование на основе алгоритма Витерби
в случае наличия ошибок при приеме