ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Каждый может ошибиться, а если о чем-нибудь очень долго размышлять, уж наверняка ошибёшься.

Ярослав Гашек
Похождения бравого солдата Швейка

Необходимость помехоустойчивого кодирования:

Если в канале есть помехи, то при приеме кодовых символов могут произойти ошибки, тогда кодовые комбинации (полученные при эффективном кодировании) будут декодированы неправильно!

Задача: повышение верности передачи

Один из путей ее решения – *помехоустойчивое* (*канальное*) кодирование.

Помехоустойчивыми (корректирующими) кодами называются коды, обеспечивающие автоматическое обнаружение и/или исправление ошибок в кодовых комбинациях.

Такая возможность обеспечивается целенаправленным *введением избыточности* в передаваемые сообщения.

При *кодировании источника* избыточность *уменьшается* или полностью устраняется (достигается увеличение скорости передачи информации за счёт уменьшения средней длины кодовых слов)

При помехоустойчивом кодировании в передаваемые сообщения *вводится избыточность* (количество символов увнличивается!)
За счет этого появляется возможность обнаруживать ошибки и даже исправлять их

Наиболее простой способ:

например, вместо слова стол можно передавать слово *сссттттоооллл*

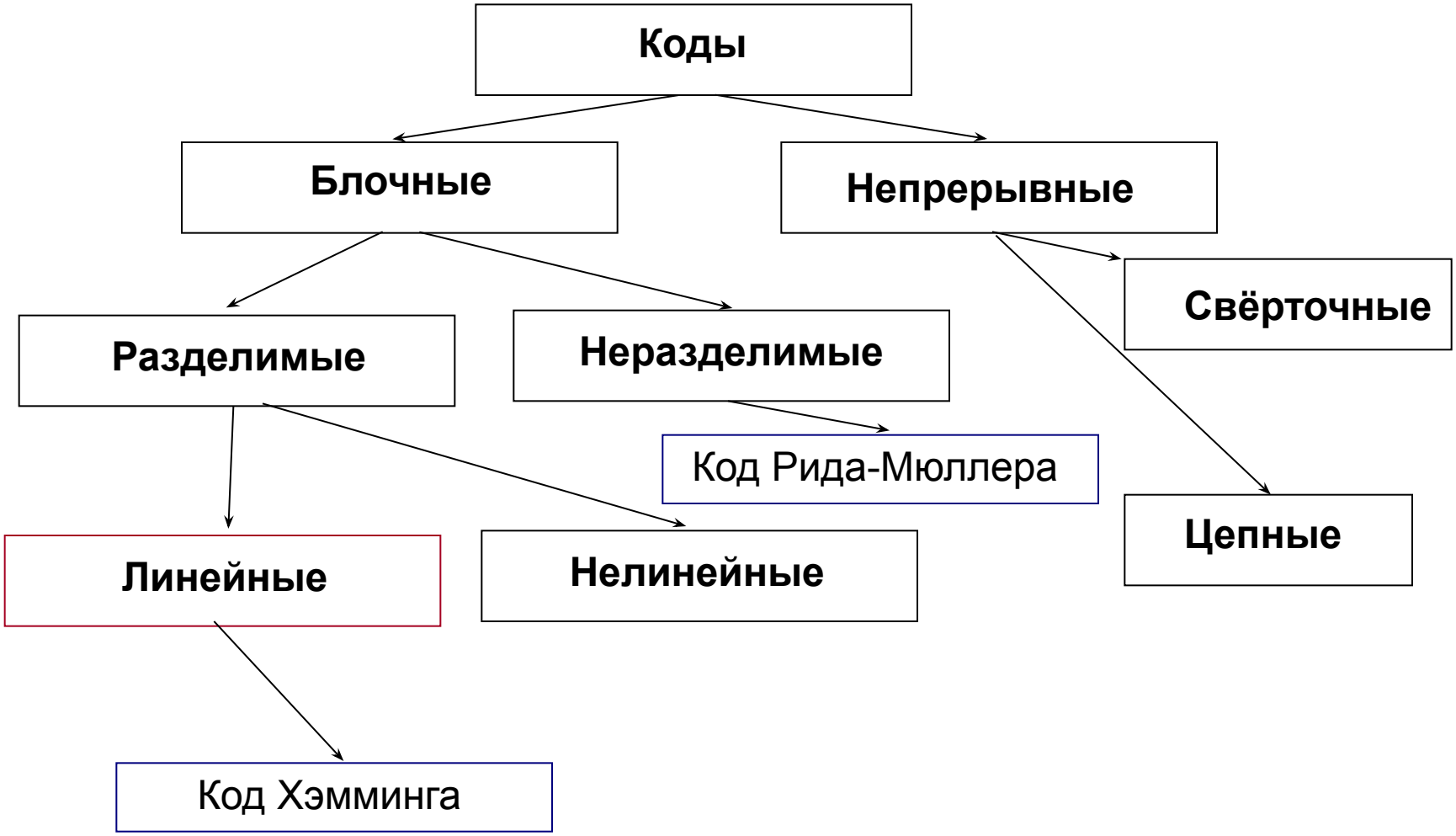
2-я Теорема Шеннона (Основная теорема кодирования для каналов с помехами (шумами)

Если производительность источника $H'(\mathrm{A})$ ***меньше пропускной способности канала*** C ***то*** *существует* ***по крайней мере одна процедура кодирования/декодирования, при которой вероятность ошибочного декодирования и ненадежность*** $H(\mathrm{A}|\mathrm{B})$ ***могут быть сколь угодно малы. Если*** $H'(\mathrm{A}) > C$ ***то такой процедуры не существует.***

В вышеприведённом примере ясно, что при фиксированном уровне помех для стремления вероятности ошибки к нулю количество повторений должно стремиться к бесконечности.

Скорость передачи информации при этом стремится к нулю.

Классификация корректирующих ко́дов



Линейные блочные ко́ды

Блочный равномерный код – множество кодовых слов (комбинаций) одинаковой длины *n*.

Элементы кодовых слов выбираются из некоторого алфавита (канальных) символов объемом *q*.

Если *q* =2, код называется двоичным.
Далее для простоты считается, что *q* =*2.*

Линейные блочные ко́ды

Поскольку все кодовые слова имеют одинаковую длину, удобно считать их векторами, принадлежащими линейному пространству размерности *n*.

00110100101101

……..

01001110100100

Для линейных ко́дов справедливо утверждение: *линейная комбинация кодовых слов является кодовым словом*.

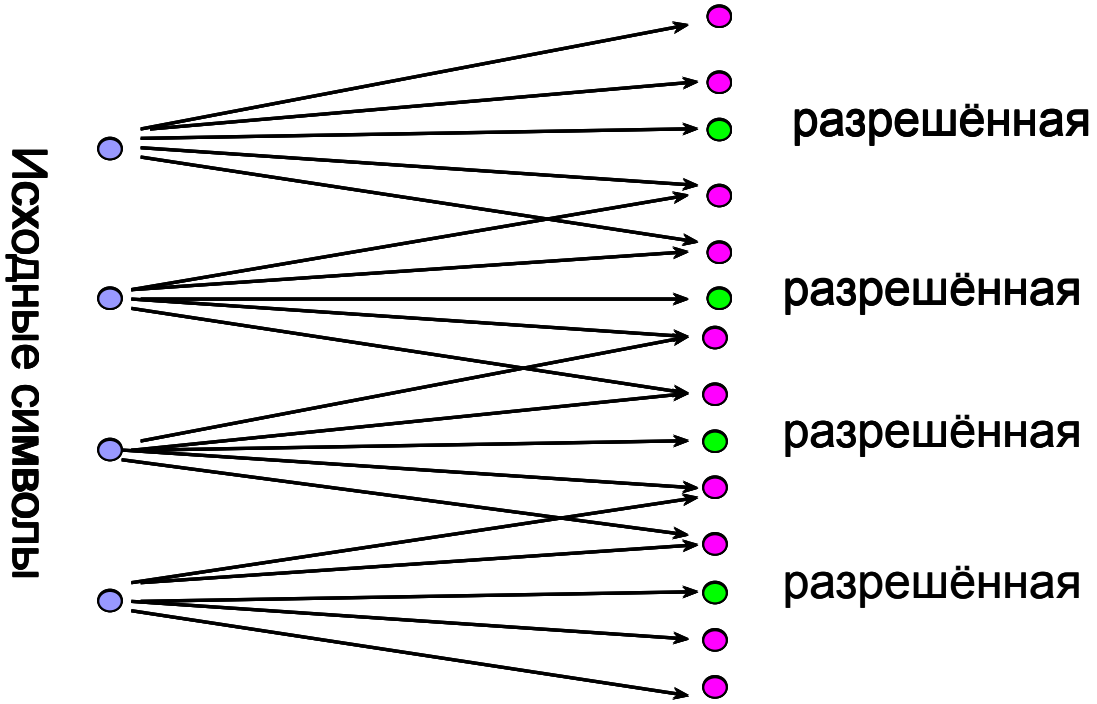
Всего 2^n *n*-мерных векторов с двоичными компонентами (кодовых комбинаций или слов).

Из них только $M = 2^k$ комбинаций являются *разрешёнными* и составляют код, который называется (n,k)-*кодом* (отношение $k/n = R$ называется относительной *скоростью кода*).

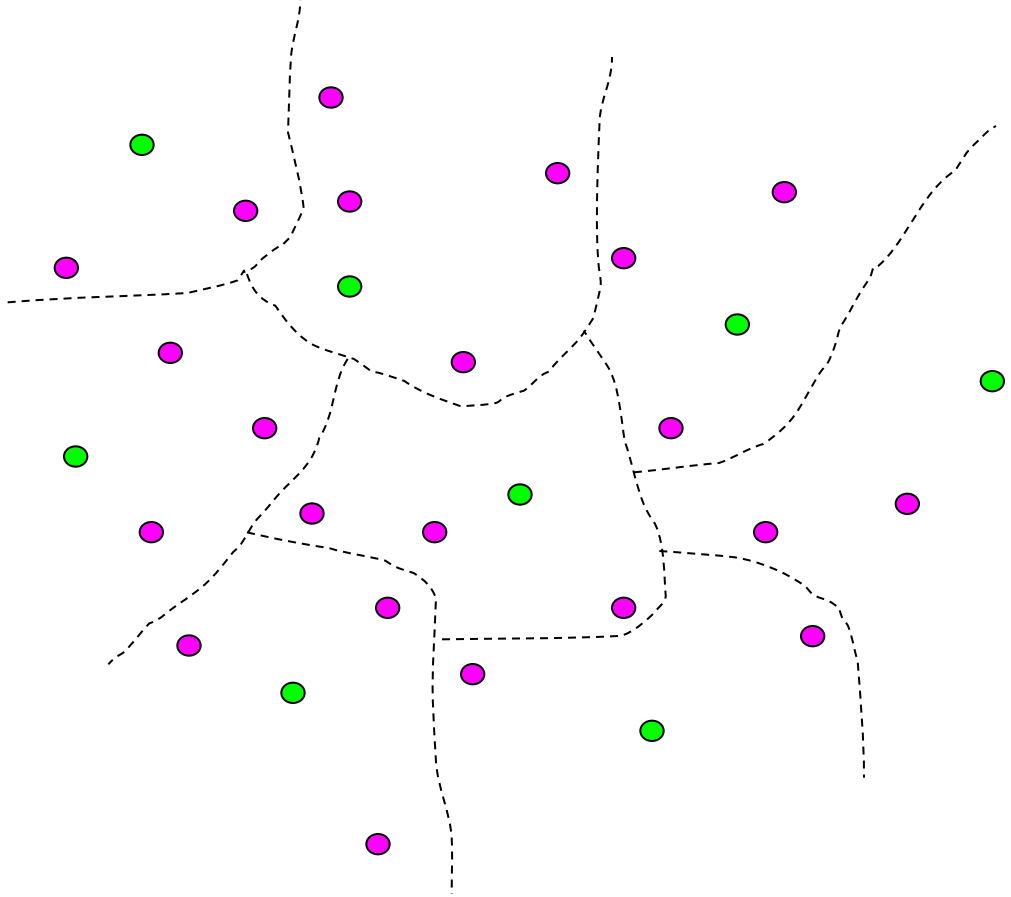
Остальные комбинации являются *запрещёнными*, образуются из разрешённых в канале под воздействием помех.

Разрешённые комбинации – векторы линейного пространства.

Чем больше расстояние между разрешёнными комбинациями, тем меньше вероятность преобразования их друг в друга под действием помех, тем выше способность кода к *обнаружению и исправлению* ошибок.



Работа декодера сводится к разбиению всего пространства на области, каждая из которых содержит одну разрешённую комбинацию



Для кодирования и декодирования линейных блочных кодов применяются действия, описываемые операциями над векторами в линейном пространстве над конечным полем целых чисел

Сложение и умножение в конечном поле понимаются как сложение и умножение по модулю *q*

Простейшее из таких полей, называемых *полями Галуа́* – поле по модулю 2, обозначаемое *GF*(2)

ЭВАРИСТ ГАЛУА
Évariste Galois
(1811-1832)
выдающийся французский математик, основатель современной алгебры.

Таблица сложения в поле $GF(2)$

+	0	1
0	0	1
1	1	0

Таблица умножения в поле $GF(2)$

×	0	1
0	0	0
1	0	1

Заметим, что вычитание по модулю 2 совпадает со сложением по модулю 2

Метрика (расстояние) Хэмминга, определяемая для двух двоичных кодовых векторов выражением

$$d(x,y)=\sum_{i=1}^{n}(x_i-y_i)\bmod 2$$

Расстояние по Хэммингу между двумя двоичными векторами равно количеству несовпадающих элементов

Скалярное произведение можно определить выражением

$$(x,y)=\sum_{i=1}^{n}x_i y_i$$

Итак, множество всех двоичных кодовых слов длины *n* можно рассматривать, как *n*-мерное линейное пространство над конечным полем скаляров *GF*(2).

Линейные коды являются разделимыми, то есть k символов – информационные, остальные (n-k) – проверочные.

Информационные символы зависят от передаваемого сообщения и могут быть какими угодно.

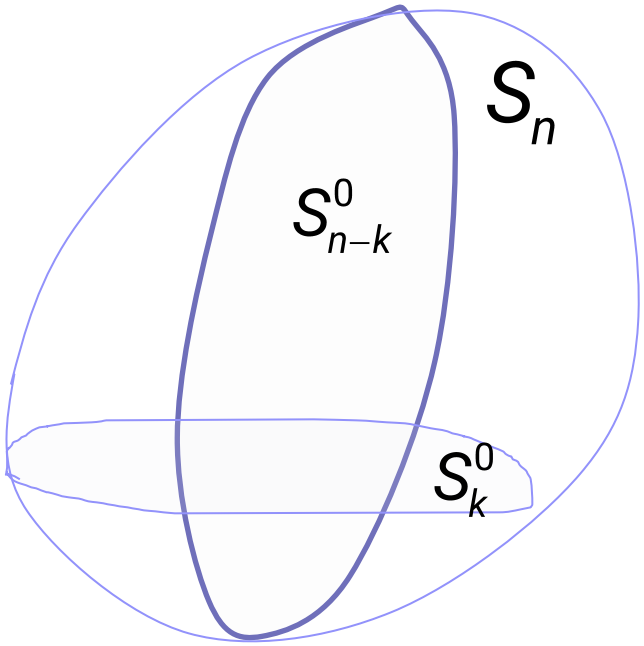
Проверочные символы однозначно определяются информационными (т.к. *формируются из информационных символов* кодером, работающим по определённому алгоритму).

Отсюда следует, что каждая кодовая комбинация, будучи вектором n-мерного пространства, принадлежит его k-мерному подпространству

Обозначим *информационные символы* $x_1, \ldots x_k$

Образуем вектор длины *n* следующим образом:

$$(x_1, \ldots x_k, 0, \ldots 0) \in S_k^0$$



$$S_n = S_k^0 \oplus S_{n-k}^0$$

– прямая сумма пространств

$$S_k^0 \perp S_{n-k}^0$$

Любой вектор одного подпространства ортогонален любому вектору из другого подпространства

Далее мы придем к другому разложению пространства

$$S_n = S_k \oplus S_{n-k}$$

$$S_k \perp S_{n-k}$$

Обозначим *информационный вектор* $\mathbf{X} = (x_1, \ldots, x_k)$

а кодовый вектор $\mathbf{C} = (c_1, \ldots, c_n)$

Кодирование описывается линейным преобразованием (оператором), отображающим векторы, соответствующие подпространству S_k^0, в векторы из S_n: $\mathbf{C} = \mathbf{XG}$

Матрица кодирования

$$\mathbf{G} = \begin{pmatrix} g_{11} & g_{12} & \cdot & \cdot & \cdot & g_{1n} \\ g_{21} & g_{22} & \cdot & \cdot & \cdot & g_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{k1} & g_{k2} & \cdot & \cdot & \cdot & g_{kn} \end{pmatrix}$$

Подробнее:

$$(x_1,\ldots,x_k)\begin{pmatrix} g_{11} & g_{12} & \cdot & \cdot & \cdot & g_{1n} \\ g_{21} & g_{22} & \cdot & \cdot & \cdot & g_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{k1} & g_{k2} & \cdot & \cdot & \cdot & g_{kn} \end{pmatrix} =$$

$$= \left(\sum_{i=1}^{k} x_i g_{i1}, \ldots \sum_{i=1}^{k} x_i g_{in} \right)$$

Или $c_j = x_1 g_{1j} + x_2 g_{2j} + \ldots + x_k g_{kj},\ j = 1,\ldots,n$

Любой кодовый вектор (кодовое слово) является *линейной комбинацией* строк матрицы, а поскольку строк ровно *k,* то
все разрешенные кодовые слова *принадлежат k-мерному подпространству, натянутому на строки матрицы, как на базис.*
Таким образом, при кодировании информационного вектора подпространство S_k^0 преобразуется линейным образом, но остается *k-мерным* подпространством S_k

Очевидно, что кодовая матрица должна состоять из *линейно независимых* строк (иначе размерности подпространства не хватит для представления *k-мерного информационного вектора*).

Путём линейных операций над строками и перестановки столбцов любую такую матрицу можно привести к *систематическому* виду

$$\mathbf{G} = \left(\mathbf{I}_k \quad \boxtimes \quad \mathbf{P}\right) = \begin{pmatrix} 1 & 0 & 0 & . & . & 0 & p_{11} & p_{12} & \cdot & p_{1(n-k)} \\ 0 & 1 & 0 & . & . & 0 & p_{21} & p_{22} & \cdot & p_{2(n-k)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & . & . & 1 & p_{k1} & p_{k2} & \cdot & p_{k(n-k)} \end{pmatrix}.$$

k

k первых символов повторяют символы информационного вектора, а остальные (*n*-*k*) символов формируются из информационных и являются проверочными (паритетными). В этом случае код называют *систематическим*.

Пример. Систематический код (7,4) порождается матрицей

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

Кодовые слова имеют структуру

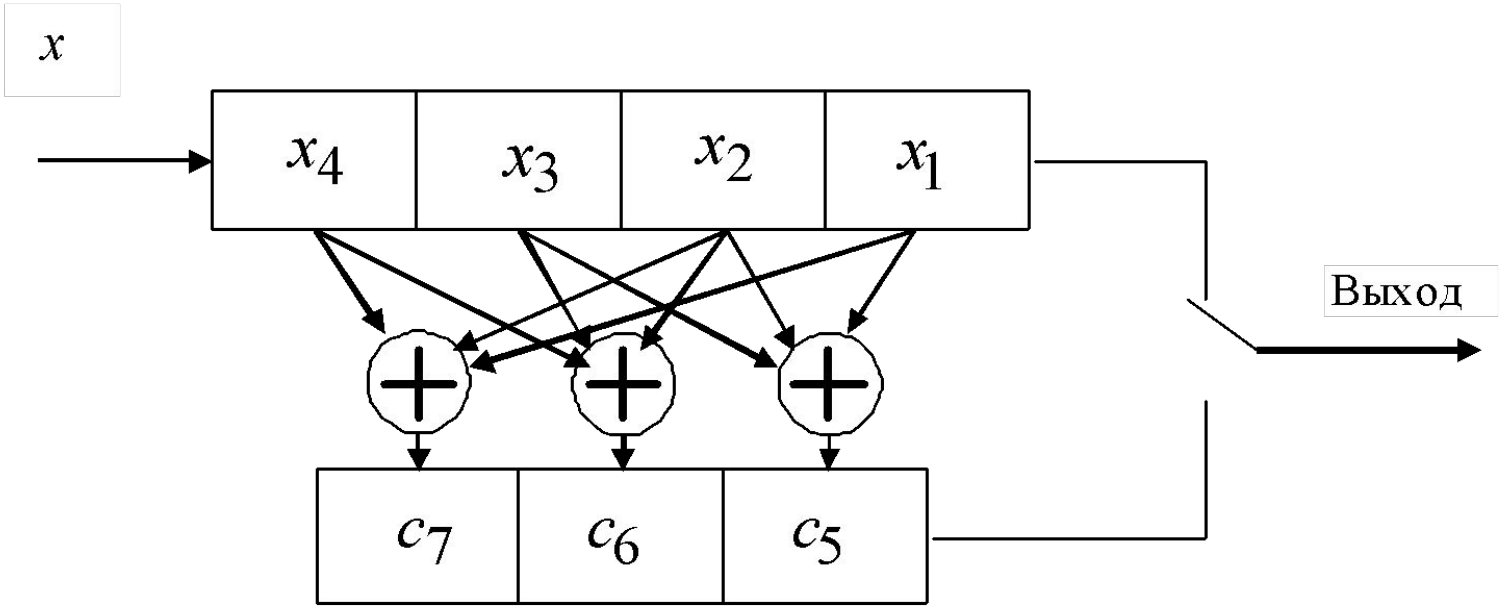
$$\mathbf{C} = (x_1, x_2, x_3, x_4, c_5, c_6, c_7)$$

где $c_5 = x_1 + x_2 + x_3$

$$c_6 = x_2 + x_3 + x_4$$

$$c_7 = x_1 + x_2 + x_4$$

Структурная схема кодера

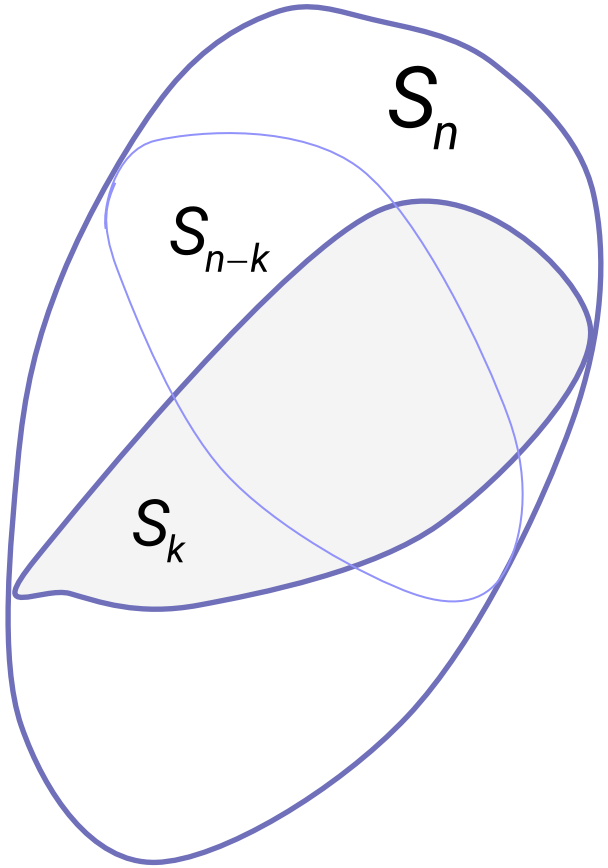


Применение любого кода предполагает реализацию не только кодирования, но и декодирования. Декодирование систематического линейного блочного кода могло бы заключаться в простом отбрасывании проверочных символов, но это не обеспечивало бы обнаружения и исправления ошибок

Подпространство S_k представляет собой множество всех разрешенных кодовых комбинаций – линейную оболочку совокупности вектор-строк порождающей матрицы.

Это подпространство и есть код.

Тогда подпространство S_{n-k}, ортогональное к нему, также можно считать некоторым кодом (*n, n-k*), *дуальным* к данному. Порождающая матрица $\mathbf{H}$ дуального кода содержит *n-k* линейно-независимых строк длины *n*



Любое кодовое слово (*n, k*)-кода ортогонально любому кодовому слову (*n, n-k*)-кода, следовательно

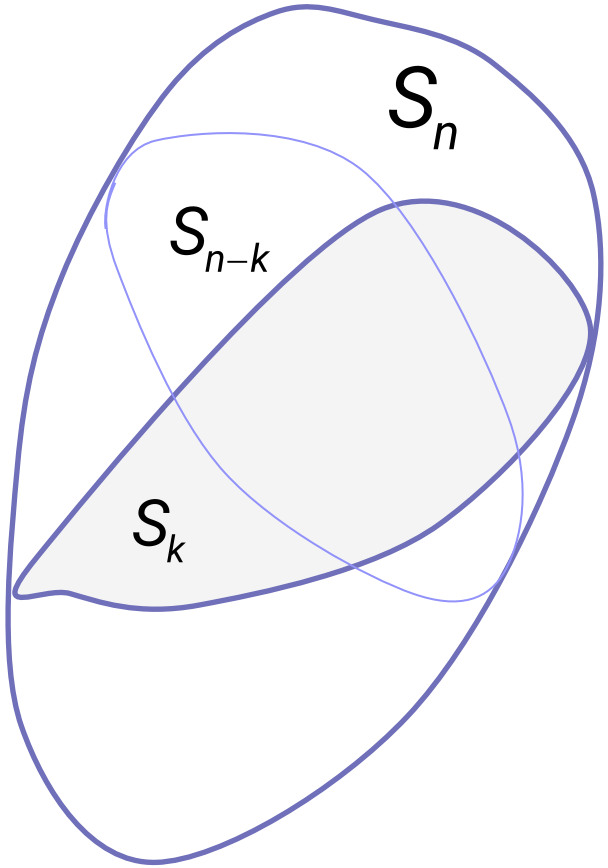
$$\mathbf{G}\mathbf{H}^{\mathbf{T}} = \mathbf{0}$$

$\mathbf{0}$ – матрица размера $k \times (n-k)$, состоящая из нулей

Для систематической матрицы $\mathbf{G} = (\mathbf{I}_k \quad \mathbf{P})$ проверочная матрица также систематическая

$$\mathbf{H} = \left(-\mathbf{P}^{\mathbf{T}} \quad \mathbf{I}_{n-k}\right)$$

(для двоичного кода минус можно опустить, так как сложение и вычитание по модулю 2 совпадают)

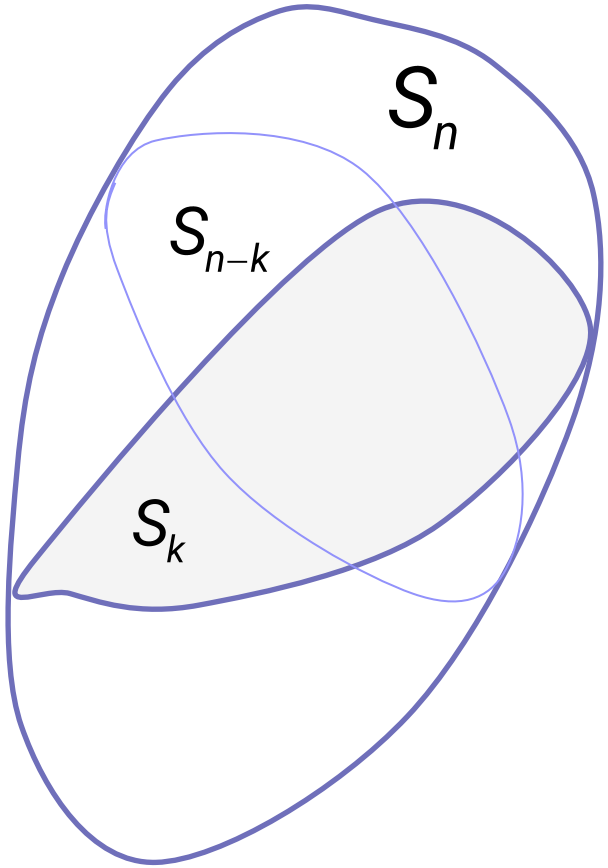


Матрица $\mathbf{H}$ является порождающей матрицей дуального кода; в то же время она может использоваться для обнаружения ошибок в комбинациях исходного кода, прошедших через канал:

$$\mathbf{G}\mathbf{H}^{\mathbf{T}} = \mathbf{0}$$

если принятая кодовая комбинация $\mathbf{Y}$ является разрешённой, то она ортогональна к подпространству S_{n-k}, т.е. ко всем строкам матрицы $\mathbf{H}$:

$$\mathbf{Y}\mathbf{H}^{\mathbf{T}} \stackrel{\Delta}{=} 0$$



Если полученный вектор ненулевой, то при передаче имела место ошибка!

Умножая слева вектор-строку, соответствующую принятой комбинации, на транспонированную матрицу $\mathbf{H}^{\mathbf{T}}$, получаем вектор (называемый *синдромом*), который равен нулевому вектору в том и только в том случае, если комбинация является разрешённой. В противном случае комбинация является запрещённой, следовательно, при передаче произошла ошибка. По значению синдрома можно определить, какой именно разряд кодового слова содержит ошибку (а при двоичном коде – и исправить её!).

Коды Хэмминга

Коды Хэмминга представляют собой (*n, k*)-коды, удовлетворяющие условию

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

В частности, рассмотренный (7,4)-код принадлежит к кодам Хэмминга при $m=3$

Для кода Хэмминга проверочная матрица содержит в качестве $n = 2^m - 1$ столбцов все возможные $(n-k)$-значные комбинации нулей и единиц, исключая нулевой вектор.

Для рассмотренного кода

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

$$\mathbf{H}^{\mathbf{T}} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Пример. Предположим, что передавалась разрешённая комбинация 0100111, а при передаче произошла ошибка, скажем, во втором символе, так что принятá комбинация 0000111.

Умножая вектор-строку (принятую комбинацию), слева на $\mathbf{H}^{\mathbf{T}}$, получим синдром

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

Полученный синдром указывает на второй символ принятой комбинации, как ошибочный. Для исправления достаточно прибавить к кодовой комбинации комбинацию 0100000

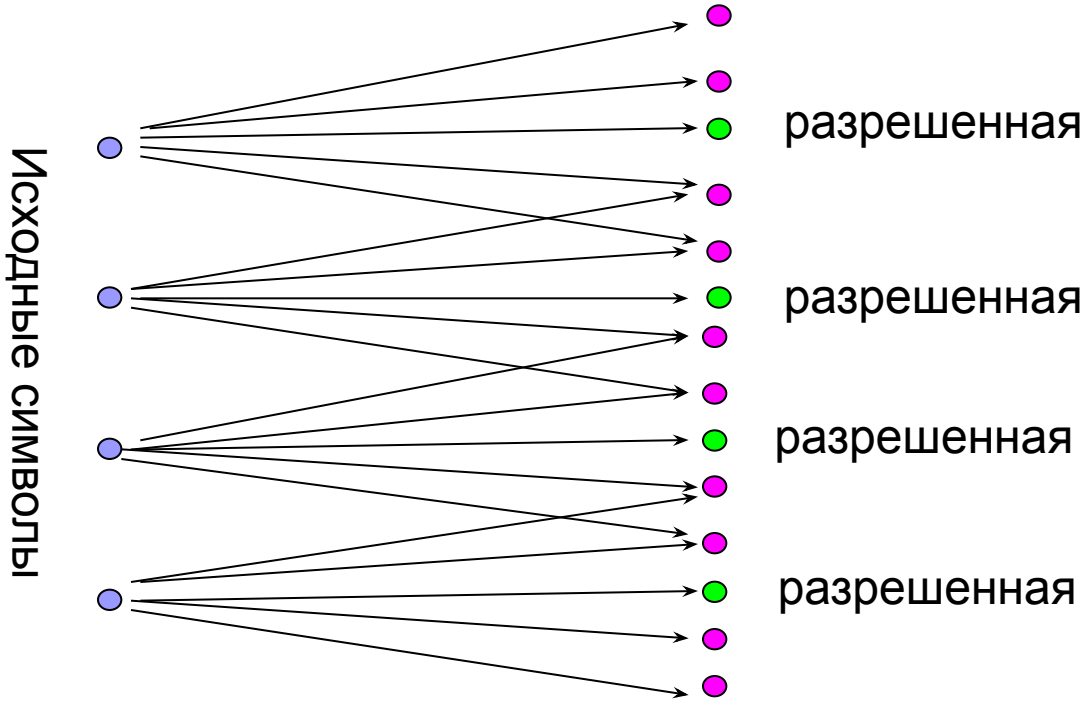
Пример. Предположим, что при передаче разрешенной кодовой комбинации 0100111 произошли две ошибки, скажем, в третьем и пятом символах, так что принята комбинация 0110011. Найдем синдром:

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

Полученный синдром совпадает с шестой строкой матрицы, что указывает на шестой символ принятой комбинации, как ошибочный. Прибавление комбинации 0000010 к принятой кодовой комбинации, очевидно, не исправляет её.

Итак, код Хэмминга (7,4) обнаруживает одно- и двукратные ошибки и исправляет однократные.

Расстояние между любыми двумя разрешенными комбинациями этого кода не менее 3. Поэтому при приёме запрещенной комбинации она заменяется той разрешенной комбинацией, расстояние до которой равно 1. Двукратная ошибка отдаляет принимаемую комбинацию на расстояние, равное 2, что и приводит к ошибочному «исправлению» ошибки. При этом «исправляется» один символ, поэтому «исправленная» комбинация отстоит от принятой на расстояние 1.



Коды, обнаруживающие ошибки, но не исправляющие их, могут использоваться в системах с решающей обратной связью (*системах с переспросом*). В таких системах при обнаружении ошибки во время декодирования по каналу обратной связи передается сигнал переспроса, и тогда передающее устройство повторяет передачу забракованной комбинации.

a $b_{\text{ц}}(t)$ $u(t)$ $z(t)$ $\hat{b}_{\text{ц}}(t)$ $\hat{a}$

При решении вопроса о целесообразности помехоустойчивого кодирования и выборе помехоустойчивого кода следует руководствоваться критерием максимума *скорости передачи информации при заданной верности*.

Введение избыточных символов приводит к:

- *увеличению* времени передачи кодовой комбинации (при постоянной ширине спектра)
- *укорочению* элементарных посылок и расширению спектра(при неизменном времени)
- при укорочении посылок приходится увеличивать мощность передатчика, иначе – *увеличение вероятности ошибочного приема* символа.

Поэтому применение конкретного кода (или помехоустойчивого кодирования вообще) может оказаться нецелесообразным.