Помехоустойчивое кодирование

Определение. ***Помехоустойчивость*** – называется способность системы осуществляющей прием информации в условиях наличия помех в линиях связи.

Определение. ***Помехой*** называется сторонние возмущение, действующее в системе, препятствующее правильному приему сигналов.

Для защиты полезной информации необходимо вводить ***избыточность*** *(смысловая, физическая, статистическая,)*

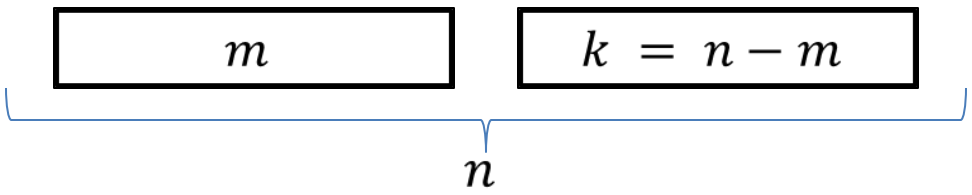
Коды, позволяющие обнаруживать и исправлять ошибки бывают двух видов:

- блоковые коды;
- сверточные коды

Коды, которые обеспечивают возможность обнаружения и исправления ошибки, называют ***помехоустойчивыми****.*

Код, содержащий помимо информационных еще и контрольные разряды называется ***систематическим кодом.***

Длина слова ***систематического кода*** *(n) =*
число информационных разрядов (m) + число контрольных разрядов(k)



Надежность обеспечивается тем, что наряду с битами, непосредственно кодирующими сообщение (будем называть их *информационными битами),* передаются (хранятся) дополнительные биты, по состоянию которых можно судить о правильности передачи (будем называть их *контрольными битами)*. При *равномерном кодировании** сообщения длина кодовой цепочки на знак (или группу знаков) ***n*** складывается из длины информационной части ***m*** и числа контрольных битов ***k***. Очевидно, $n \geq m$.

Подобно введенной ранее величине *Q*, характеризующей относительную избыточность кода при передаче по идеальному каналу, в рассматриваемом случае удобно определить *избыточность сообщения* для реального канала *L* следующим образом:

$$\mathrm{L} = \frac{n}{m} = \frac{m+k}{m} = 1 + \frac{k}{m} \qquad (1)$$

Относительная избыточность сообщения *- это характеристика, показывающая, во сколько раз требуется удлинить сообщение, чтобы обеспечить его надежную (безошибочную) передачу (хранение).*

Очевидно, что *L* характеризует эффективность кодирования при передаче по реальным каналам и при равных надежностях передачи предпочтение должно быть отдано тому способу кодирования, при котором избыточность окажется наименьшей. Правда, для практики важна также простота технической реализации того или иного способа кодирования.

Коды, обнаруживающие ошибку

Задача *обнаружения* ошибки может быть решена довольно легко. Достаточно передавать каждую букву сообщения *дважды.*

Например, при необходимости передачи слова *«гора»* можно передать «*ггоорраа*».

При получении искаженного сообщения, например, «*гготрраа*» с большой вероятностью можно догадаться, каким было исходное слово. Конечно, возможно такое искажение, которое делает неоднозначным интерпретацию полученного сообщения, например, «гпоорраа», «*ггоорреа*» или *«кгоорраа».*

Однако цель такого способа кодирования состоит не в исправлении ошибки, а в фиксации факта искажения и повторной передаче части сообщения в этом случае. Недостаток данного способа обеспечения надежности состоит в том, что избыточность сообщения оказывается очень большой - очевидно, $L = 2$.

Поскольку ошибка должна быть только обнаружена, можно предложить другой способ кодирования. Пусть имеется цепочка информационных бит длиной m. Добавим к ним один контрольный бит $(k = 1)$, значение которого определяется тем, что новая кодовая цепочка из $m + 1$ бит должна содержать *четное количество единиц* - по этой причине такой контрольный бит называется *битом четности.*

Например, для информационного байта 01010100 бит четности будет иметь значение 1, а для байта 11011011 бит четности равен 0.

В случае одиночной ошибки передачи число 1 перестает быть четным, что и служит свидетельством сбоя. Например, если получена цепочка 110110111 (контрольный бит выделен подчеркиванием), ясно, что передача произведена с ошибкой, поскольку общее количество единиц равно 7, т.е. нечетно.

Предложенный способ кодирования не позволяет установить, в каком конкретно бите содержится ошибка и, следовательно, не дает возможности ее исправить. Избыточность сообщения при этом равна:

$$\mathrm{L} = \frac{m+1}{m} = 1 + \frac{1}{m}$$

На первый взгляд кажется, что путем увеличения ***m*** можно сколь угодно приближать избыточность к ее минимальному значению ($L_{min} = 1$). Однако с ростом ***m***,

- во-первых, растет вероятность парной ошибки, которая контрольным битом не отслеживается;
- во-вторых, при обнаружении ошибки потребуется заново передавать много информации. Поэтому обычно ***m*** = 8 или 16 и, следовательно, $L = 1{,}125$ (1,0625).

Коды, исправляющие одиночную ошибку

Можно было бы предложить простой способ установления ошибки – передавать каждый символ трижды, например, «*гггоооррраааа*» – тогда при получении сообщения *«гггоооnрраааа»* ясно, что ошибочной оказывается буква «п» и ее следует заменить на «р».

Безусловно, при этом предполагается, что вероятность появления парной ошибки невелика. Такой метод кодирования приводит к избыточности сообщения $L = 3$, что неприемлемо с экономической точки зрения.

Прежде, чем обсуждать метод кодирования, позволяющий локализовать и исправить ошибку передачи, произведем некоторые количественные оценки.

Наличие шумов в канале связи ведет к частичной потере передаваемой информации на величину возникающей неопределенности, которая при передаче одного бита исходного сообщения составляет

$$\mathrm{H} = -\mathrm{p} * \log_2 p - (1-p) * \log_2 (1-p)$$

где p - вероятность появления ошибки в сообщении. Для восстановления информационного содержания сообщения следует дополнительно передать количество информации не менее величины ее потерь, т.е. вместо передачи каждого 1 бит информации следует передавать $1 + H$ бит. В этом случае избыточность сообщения составит

$$L_{min} = \frac{1+H}{1} = 1 - \mathrm{p} * \log_2 p - (1-p) * \log_2 (1-p) \qquad (2)$$

Приведенную избыточность следует считать *минимальной* (это указывает ее индекс), поскольку при передаче сообщения по каналу, характеризуемому вероятностью искажения p, при избыточности, меньшей L_{min} восстановление информации оказывается невозможным.

Пример

Какое минимальное количество контрольных бит должно передаваться вместе с 16-ю информационными для обеспечения восстановимости информации, если вероятность искажения составляет 1%?

Подставляя $p = 0{,}01$ в (2), находим $L_{min} \approx 1{,}081$. При $m = 16$ из (1) получаем $n = m \cdot L_{min} = 17{,}29$. Следовательно, с учетом того, что количество контрольных бит выражается целым числом, $k \geq n - m = 2$. Реальная избыточность согласно (1) составит $L = 1{,}125$.

Выражение (2) устанавливает границу избыточности, при которой возможно восстановление переданной информации, однако, не указывает, каким образом следует осуществить кодирование, чтобы ошибка могла быть локализована (т.е. определено, в каком бите она находится) и, естественно, устранена.

Схема системы связи

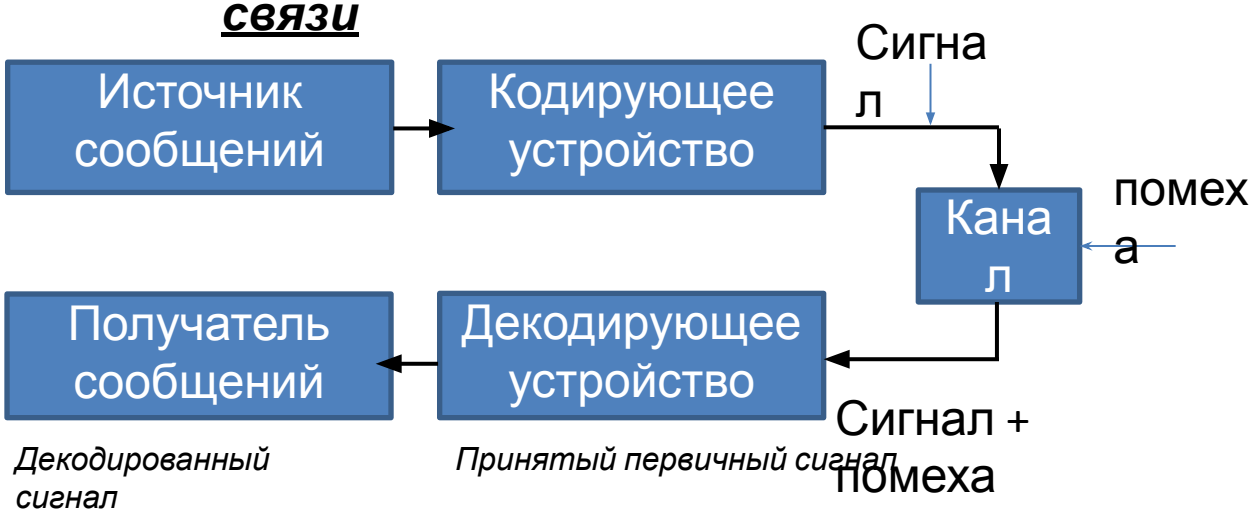


Схема работы кодирующего и декодирующего устройств

Алфавит:

$\Sigma = \{a_1, a_2, \ldots, a_n\}$

$\Sigma' = \{b_1, b_2, \ldots, b_p\}$

Сообщение:

$\alpha = a_{i1}\, a_{i2} \ldots a_{im}$, где $a_{ij} \in \Sigma$

→ Процесс кодирования

→ Процесс декодирования

Кодовая таблица

a_1	$c(a_1) = b_{11} b_{12} .. b_{1r}$
a_2	$c(a_2) = b_{21} b_{22} .. b_{2r}$
..	…
a_k	$c(a_k) = b_{k1} b_{k2} .. b_{kr}$

Закодированное сообщение:

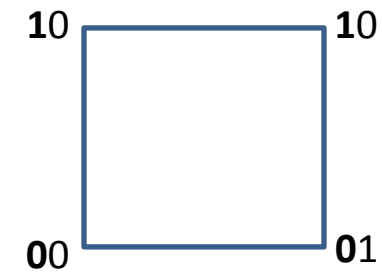
$\beta = c(a_{i1})\, c(a_{i2}) \ldots c(a_{im}) = b_{i1}\, b_{i2} \ldots b_{in}$

Геометрическая интерпретация построения кодовых слов

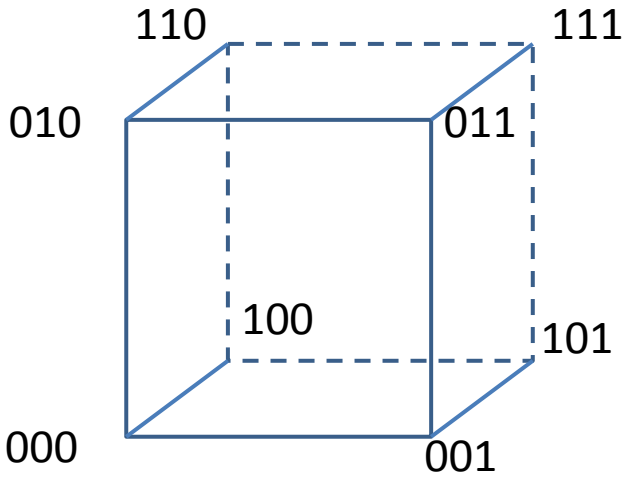
Пусть α= 101011 – вектор в пространстве или точка в пространстве B^n, где n – длина слова (блока); B={0,1}.

Всего точек в пространстве $B^n \rightarrow 2^n$.

Примеры пространств B^n разных размерностей



Пример пространства B^2



Пример пространства B^3

Пример пространства B^1

Код

– это некоторые точки пространства B^n, или некоторое подмножество пространства B^n

Замечания к процедуре построения двоичных кодовых слов:

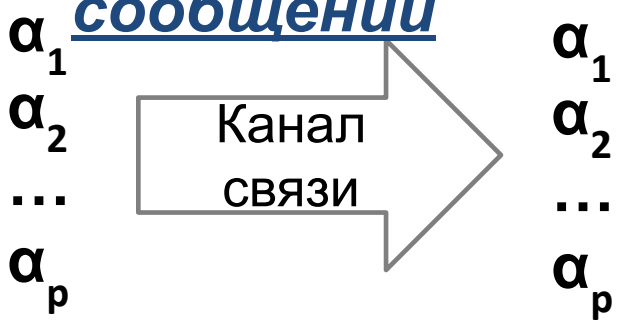
1. На вход кодирующего устройства поступает последовательность из m информационных двоичных символов. На выходе ей соответствует последовательность из n двоичных символов, причем n>m.

2. Всего может быть:
   - $\mathbf{2^m}$ различных входных и
   - $\mathbf{2^n}$ различных выходных последовательностей.

3. Из общего числа 2^n выходных последовательностей только 2^m последовательностей соответствуют входным. Их называют ***разрешенными кодовыми комбинациями***.

4. Остальные $2^n - 2^m$ возможных выходных последовательностей для передачи не используются. Их называют **запрещенными кодовыми комбинациями**.

Передача сообщений



При передаче сообщения от источника к получателю возможны следующие варианты передачи и получения сообщений:

1. Передача с ошибкой, ошибка обнаружена (отправили $\alpha_i \rightarrow$ получили α_j, $\forall\, \boldsymbol{i,j = 1..p, i \neq j}$);
2. Передача без ошибок (отправили $\alpha_i \rightarrow$ получили α_i, $\forall\, \boldsymbol{i = 1..p}$);
3. Передача с ошибкой, ошибка не обнаружена (отправили $\alpha_i \rightarrow$ получили β, $\forall\, \boldsymbol{i = 1..p}$, β - не является кодовым словом);

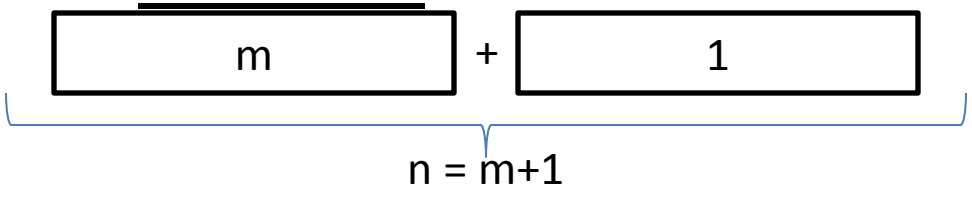
Существуют следующие способы борьбы с ошибками передачи:

- Увеличить расстояние между точками пространства B^n;
- Уменьшить количество кодовых слов.

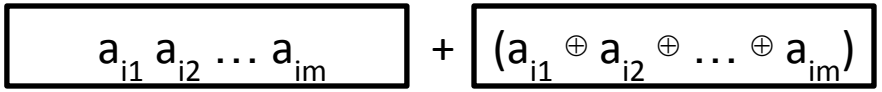
Определение. Код обнаруживает **t** ошибок, если для всякого **r ≤ t r** ошибок переводит кодовое слово в некодовое.

Примеры кодов

Код с проверкой на четность



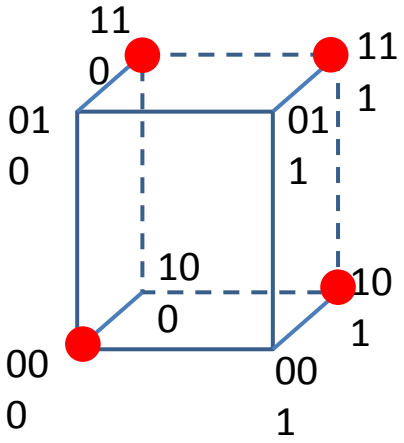
Описание: Код дополняется 1 контрольным разрядом, в который записывается 0 или 1, дополняющие сумму информационных разрядов до четного числа.



Пример кодирования двухразрядных слов:

Кодовая таблица

00 → 000
01 → 011
10 → 101
11 → 110



Код - тетраэдр

Примеры кодов

Модификация кода с проверкой на четность

Кодируется слово $a_{i1}\, a_{i2}\, \ldots\, a_{im}$. Длинное слово разбивается на группы по q разрядов и записываются в таблицу размера $(q \times p)$, где p = $\frac{m}{q}$. Контрольные разряды выделяются всем группам по строкам и столбцам. Количество контрольных разрядов - q*p;

		…		
		…		
…	…	…	…	…
		…		
		…		

Замечания.

1. Увеличение избыточности передаваемых кодов приводит к тому, что появляется возможность не только обнаружить, но и исправить ошибку.
2. Признаком отсутствия искажения в процессе приема-передачи является равенство контрольного разряда нулю

Примеры кодов

Коды с повторением

Один заданный информационный символ повторяется n раз. Это (n, 1)-код. Для него минимальное расстояние равно n, и при предположении, что большинство принятых битов совпадает с переданным информационным битом, может быть исправлено (n — 1)/2 ошибок.

0 ↔ 00000

1 ↔ 11111

Расстояние Хэмминга. Кодовое расстояние

На множестве двоичных слов длины ***m*** расстоянием ***d(a,b)*** – ***расстоянием Хэмминга*** - между двумя словами **a** и **b** называют число несовпадающих позиций этих слов.

Например: расстояние между словами $a = 0110100$ и $b = 0010101$ равно **2**.

$$\begin{array}{r} 0110100 \\ \oplus\ \underline{0010101} \\ 0100001 \end{array}$$

Определение: Минимальное расстояние, взятое по всем парам кодовых разрешенных комбинаций кода, называют (***минимальным)*** ***кодовым расстоянием*** ***(d_{0min}).***

Пример: Рассмотрим код, заданный таблицей

00	↔	10101
01	↔	10010
10	↔	01110
11	↔	11111

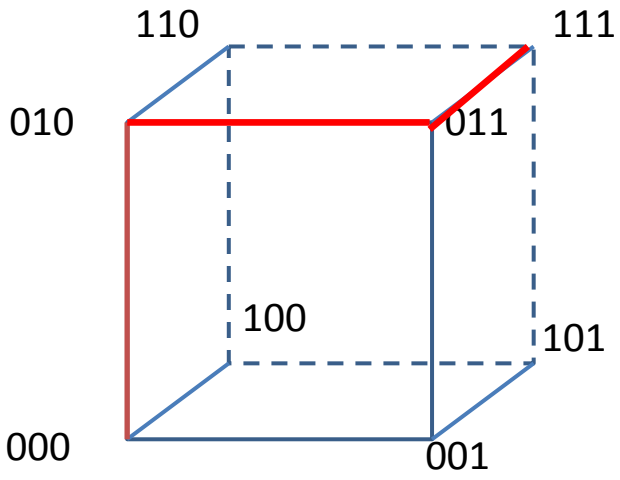
$$d(10101, 10010) = 3$$
$$d(10101, 01110) = 4$$
$$d(10101, 11111) = 2$$
$$d(10010, 01110) = 3$$
$$d(10010, 11111) = 3$$
$$d(01110, 11111) = 2$$

$d_{0min} = 2$ – *кодовое расстояние для данного кода*

Геометрическая интерпретация кодового расстояния и расстояния Хэмминга

При взаимно независимых ошибках наиболее вероятен переход в кодовую комбинацию, отличающуюся от данной в наименьшем числе символов.

Рассмотрим код, исправляющий ошибку. Идея построения такого кода наглядно иллюстрируется геометрической моделью трехзначного двоичного кода на все сочетания, которая представляет собой куб.



Для каждой вершины куба имеются три вершины, которые отстоят от нее на один шаг (на расстоянии одного ребра куба), еще три вершины, которые отстоят на два шага, и одна вершина — на три шага.

Расстояние между ближайшими кодовыми комбинациями *называется* ***кодовым расстоянием.***

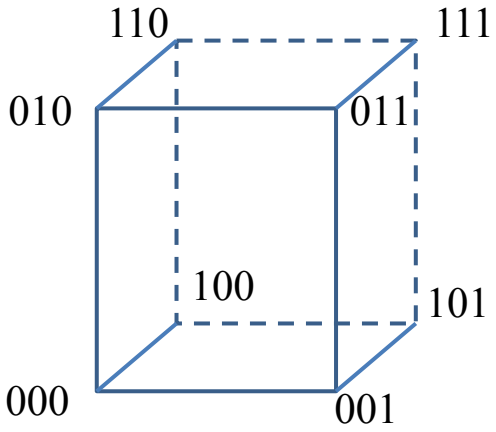
Замечание. Кодовое расстояние — параметр, характеризующий помехоустойчивость кода и заложенную в нем избыточность.

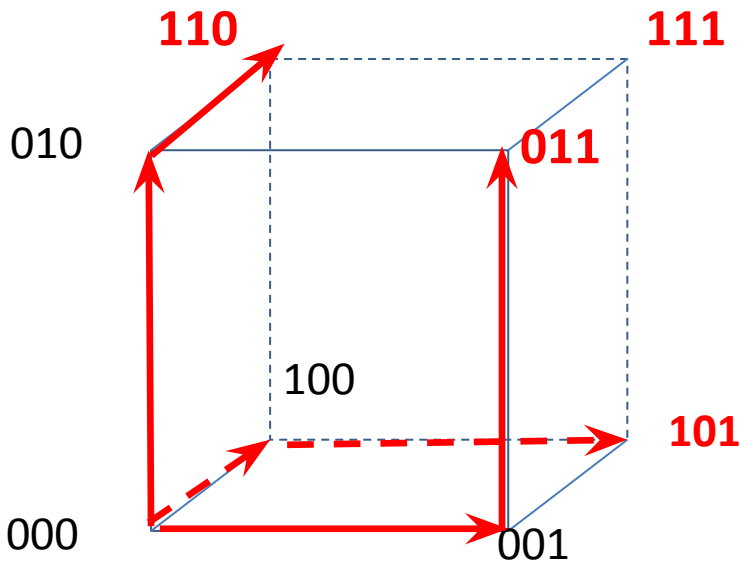
Кодовое расстояние (d)

Если кодовое расстояние $d = 1$ (избыточность в коде отсутствует), то не могут быть обнаружены даже единичные искажения, так как искаженная комбинация будет совпадать с одной из разрешенных.

Если кодовое расстояние $d = 2$, то такой код позволяет обнаруживать одиночные ошибки, так как уже есть возможность сделать так, чтобы искаженная комбинация не входила в число разрешенных.

По рисунку легко определить кодовые комбинации, обнаруживающие ошибку в комбинации **000**. Они должны отличаться друг от друга в двух символах, т. е. отстоять от точки **0** на два шага.





Как видно из рисунка ими являются 110, 011, 101. Для исправления одиночной ошибки расстояние от точки **0** следует увеличить еще на один шаг. Такая комбинация будет только одна — 111.

Для трехмерного куба корректирующие комбинации расположены на противоположных вершинах куба. Это пары 000—111, 010—101, 001—110, 011—100 (Коды-спутники).

Исправление ошибки в кодах-спутниках

Идея исправления ошибки в кодах-спутниках весьма проста. Главное, чтобы при искажении любой комбинации не могла быть образована соседняя рабочая комбинация. Процесс исправления ошибки заключается в том, что искаженная комбинация отождествляется с ближайшей разрешенной комбинацией.

Например, если передавать буквы алфавита, которым соответствуют следующие комбинации двоичного кода: *А* — 00000, *Б* — 00111 и *В* — 11100, то при искажении любого одного знака легко определить, какая комбинация была передана, так как каждая из них отличается друг от друга не меньше чем в трех символах (кодовое расстояние $d \geq 3$).

Декодирование после приема производится таким образом, что принятая кодовая комбинация отождествляется с той разрешенной, которая находится от нее на **наименьшем** кодовом расстоянии.

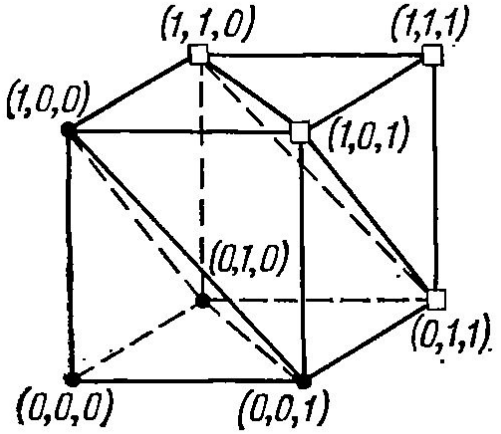
Такое декодирование называется декодированием ***по методу максимального правдоподобия***

В общем случае при необходимости обнаруживать ошибки кратности до r включительно минимальное хэммингово расстояние между разрешенными кодовыми комбинациями должно быть по крайней мере на единицу больше r.

$$d_{0\ min} \geq r+1$$

В общем случае для обеспечения возможности исправления всех ошибок кратности до s включительно при декодировании по методу максимального правдоподобия, каждая из ошибок должна приводить к запрещенной комбинации, относящейся к подмножеству исходной разрешенной кодовой комбинации

Пример. Рассмотрим (2,3)–код с проверкой на четность. Множество кодовых слов есть 000, 101, 011, 110. Минимальное расстояние между кодовыми словами равно 2. Этот код способен обнаруживать однократную ошибку.



Общее выражение для определения кодового расстояния в случае одновременного обнаружения и исправления ошибок

$$d = r + s + 1$$

, где

r — число обнаруживаемых ошибок;

s — число исправляемых ошибок;

d — минимальное количество элементов, в которых одна кодовая комбинация отличается от другой.

Если требуется определить кодовое расстояние исходя только из количества исправляемых ошибок, то применяют формулу

$$d = 2s + 1$$

Примеры кодов

Код Хэмминга. Общее описание

$d_{0min} = 3$

Для $\forall m \exists$ **(n, m)** код Хэмминга

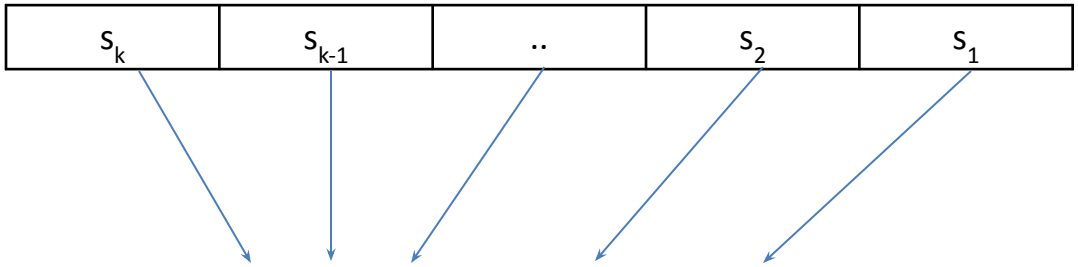
(n,m) = **(2^k-1, 2^k-1-k)**

Это систематический код, с **m** информационными и **k = (n-m)** проверочными битам. Код Хэмминга является кодом с проверкой на четность, с той лишь разницей, что эта проверка производится **k** раз.

При каждой проверке охватывается часть информационных символов и один избыточный, при этом получается один контрольный символ.

Если результат проверки дает четное число, то контрольному символу присваивается значение '**0**', если нечетное – '**1**'.

k - разрядное двоичное число:



Когда при передаче кодового слова возникает одиночная ошибка, окажутся невыполненными те проверочные соотношения S_i, в которые входит значение ошибочного разряда.

Порядок кодирования по методу Хемминга

Пусть по каналу связи передается слово

$$\bar{u} = (u_1, u_2, \dots, u_m),\ u_i \in \Sigma = \{0,1\},$$

тогда закодированное сообщение обозначим

$$\bar{u}' = (u_1, u_2, \dots, u_m, u_{m+1}, \dots, u_n)$$

Зависимость между числом информационных и проверочных разрядов

$$2^m \leq \frac{2^n}{1+n}$$ и $$2^k \geq n+1$$

Соотношение между количеством информационных и контрольных символов в коде Хэмминга

n	m	k	n	m	k
3	1	2	11	7	4
5	2	3	12	8	4
7	4	3	13	9	4
9	5	4	14	10	4
10	6	4	15	11	4

Определение мест расположения и значений контрольных символов

Для определения мест расположения контрольных символов необходимо построить матрицу кода Хемминга. Размер матрицы - (k*n).

Характерной её особенностью является то, что столбцы матрицы являются различными ненулевыми комбинациями символов алфавита {0,1} длины k, выписанные в порядке возрастания их значений.

Пример. Матрица кода Хэмминга (15,11)

k

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

$n=2^k$

Строки матрицы – это проверочные уравнения из которых вычисляются значения контрольных разрядов. Единицы в строке – обозначают разряды, которые будут принимать участие в суммировании (или в контролировании).

$$s_1 = \mathbf{u'_1} \oplus u'_3 \oplus u'_5 \oplus u'_7 \oplus u'_9 \oplus u'_{11} \oplus u'_{13} \oplus u'_{15} = 0$$
$$s_2 = \mathbf{u'_2} \oplus u'_3 \oplus u'_6 \oplus u'_7 \oplus u'_{10} \oplus u'_{11} \oplus u'_{14} \oplus u'_{15} = 0 \qquad (*)$$
$$s_3 = \mathbf{u'_4} \oplus u'_5 \oplus u'_6 \oplus u'_7 \oplus u'_{12} \oplus u'_{13} \oplus u'_{14} \oplus u'_{15} = 0$$
$$s_4 = \mathbf{u'_8} \oplus u'_9 \oplus u'_{10} \oplus u'_{11} \oplus u'_{12} \oplus u'_{13} \oplus u'_{14} \oplus u'_{15} = 0$$

Целесообразно выбирать такое размещение контрольных символов в кодовой комбинации, при которой каждый из них включается в минимальное число проверяемых групп (лучше в одну). На этом основании контрольные разряды это
– u'_1, u'_2, u'_4, u'_8 , то есть те места, где столбцы матрицы Хэмминга является степенью числа 2 - 2^0, 2^1, 2^2, 2^4 и т.д.

Пример. Матрица кода Хэмминга (15,11)

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

Окончательно получаем закодированное сообщение для кода (15,11):

$$u' = (k_1, k_2, u_1, k_3, u_2, u_3, u_4, u_5, u_6, u_7, k_4, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15})$$

Значения контрольных разрядов получаем из системы уравнений (*):

$k_1 = u'_3 \oplus u'_5 \oplus u'_7 \oplus u'_9 \oplus u'_{11} \oplus u'_{13} \oplus u'_{15}$
$k_2 = u'_3 \oplus u'_6 \oplus u'_7 \oplus u'_{10} \oplus u'_{11} \oplus u'_{14} \oplus u'_{15}$
$k_3 = u'_5 \oplus u'_6 \oplus u'_7 \oplus u'_{12} \oplus u'_{13} \oplus u'_{14} \oplus u'_{15}$
$k_4 = u'_9 \oplus u'_{10} \oplus u'_{11} \oplus u'_{12} \oplus u'_{13} \oplus u'_{14} \oplus u'_{15}$

Порядок проведения проверок и декодирования

При получении закодированного по методу Хэмминга сообщения необходимо проверить выполнимость соотношений для контрольных разрядов:

$$s_1 = k_1 \oplus u'_3 \oplus u'_5 \oplus u'_7 \oplus u'_9 \oplus u'_{11} \oplus u'_{13} \oplus u'_{15}$$
$$s_2 = k_2 \oplus u'_3 \oplus u'_6 \oplus u'_7 \oplus u'_{10} \oplus u'_{11} \oplus u'_{14} \oplus u'_{15}$$
$$s_3 = k_3 \oplus u'_5 \oplus u'_6 \oplus u'_7 \oplus u'_{12} \oplus u'_{13} \oplus u'_{14} \oplus u'_{15}$$
$$s_4 = k_4 \oplus u'_9 \oplus u'_{10} \oplus u'_{11} \oplus u'_{12} \oplus u'_{13} \oplus u'_{14} \oplus u'_{15}$$

В результате будет получена k-разрядное число S, которое называется «синдром»:

S:

s_4	s_3	s_2	s_1

Правила интерпретации значения синдрома:

- S=0 – передача сообщения произошла без ошибок;
- S<>0 – во время передачи произошла ошибка, при этом – десятичное значение синдрома – номер разряда , переданного с ошибкой.

Алгоритм декодирования кода Хэмминга:

1. Провести проверку всех битов чётности
2. Если все биты чётности верны, то перейти к п 5.
3. Вычислить сумму номеров всех неправильных битов чётности
4. Инвертировать содержимое бита, номер которого равен сумме, найденной в п.3
5. Исключить биты чётности , передать правильный информационный код

Пример: Закодировать сообщение 1101 кодом Хэмминга.

m=4, k=3, n=7

$$H = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{matrix}$$

В качестве проверенных разрядов выбираем 1й, 2й и 4й. Чтобы закодировать сообщение 1101, нужно определить проверочные разряды в комбинации

$[k_1\ k_2\ 1\ k_3\ 1\ 0\ 1]$. Из матрицы H получаем:

$k_1 = u_3 \oplus u_5 \oplus u_7$
$k_2 = u_3 \oplus u_6 \oplus u_7$
$k_3 = u_5 \oplus u_6 \oplus u_7$

Следовательно, $k_1 = 1$, $k_2 = 0$, $k_3 = 0$ и закодированное сообщение имеет вид: **1010101.**

Предположим, что шестой символ принят ошибочно, тогда будет получено сообщение 1010111. Синдром в этом случае имеет вид 110. т. е. двоичное представление числа 6.

При построении кодов перед разработчиками стоит задача определения числа добавочных, корректирующих символов k, или исходя из числа информационных разрядов m, либо из общей длины кода n.

Для обнаружения и исправления **одиночной** ошибки соотношение между числом информационных разрядов **m** и числом корректирующих разрядов **k** должно удовлетворять следующим условиям:

$$2^k \geq n + 1$$

$$2^m \leq \frac{2^n}{n+1}$$

При этом подразумевается, что общая длина кодовой комбинации $n = m + k$

Для практических расчетов при определении числа контрольных разрядов кодов с минимальным кодовым расстоянием $\mathbf{d_{0min} = 3}$ удобно пользоваться выражениями:

1. Если известна длина полной кодовой комбинации *n*

$$k_1 = [\log_2(n+1)]$$

2. Если при расчетах удобнее исходить из заданного числа информационных символов m

$$k_2 = [\log_2\{(m+1) + [\log_2(m+1)]\}]$$

Для кодов, обнаруживающих все **трехкратные ошибки** ($d_{0min} = 4$)

$$k_3 \geq 1 + \log_2(n+1)$$

$$k_3 \geq 1 + \log_2[(m+1) + \log_2(m+1)]$$

Для кодов длиной в *n* символов, исправляющих **одну или две ошибки** ($d_{0min} = 5$),

$$k_2 \geq \log_2(C_n^2 + C_n^1 + 1)$$

Для практических расчетов можно пользоваться выражением

$$k_2 = [\log_2 \frac{n^2 + n + 1}{2}]$$

Для кодов, исправляющих **три ошибки** ($d_{0min} = 7$),

$$k_2 = [\log_2 \frac{n^3 + n^2 + n + 1}{6}]$$

Для кодов, исправляющих **S ошибок** ($d_{0min} = 2S + 1$),

$$\log_2\left(C_n^s + C^{s-1}_{n} + \cdots + 1\right) < k_s < \log_2\left(C^{2s-1}_{n-1} + C^{2s-2}_{n-1} + \cdots + 1\right)$$

Выражение слева известно как нижняя граница Хэмминга, а выражение справа как верхняя граница ***Варшамова — Гильберта***.

В настоящее время разработаны десятки кодов, которые теоретически могут обнаруживать произвольное количество ошибок.

Циклические коды

Зависимость общего числа разрядов комбинаций от количества информационных и исправляемых разрядов

Общее число битов n	Число полезных битов, m	Число исправляемых битов, s
31	26	1
	21	2
	16	3
63	57	1
	51	2
	45	3
127	120	1
	113	2
	106	3

Примеры порождающих

Степень полинома, r	Порождающий полином, P(x)
2	111
3	1011
4	10011
5	100101, 111101, 110111
6	1000011, 1100111
7	10001001, 10001111, 10011101
8	111100111, 100011101, 101100011