Кодирование

Кодирование

Кодирование – преобразование дискретного сообщения в дискретный сигнал, осуществляемое по определенному правилу.

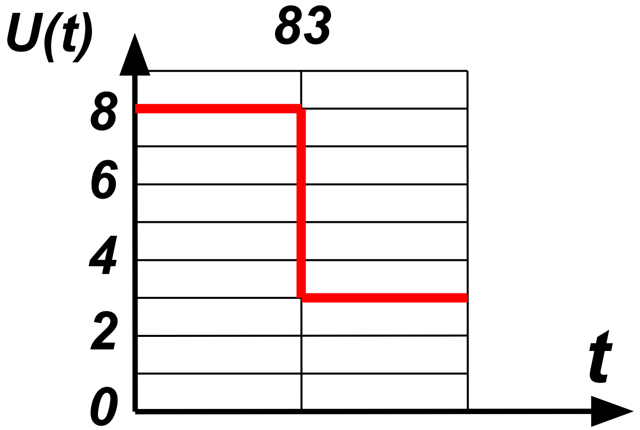
Декодирование – восстановление дискретного сообщения по сигналу на выходе дискретного канала, осуществляемое с учетом правила кодирования.

Код – совокупность условных сигналов, обозначающих дискретные сообщения.

Десятичные и двоичные коды

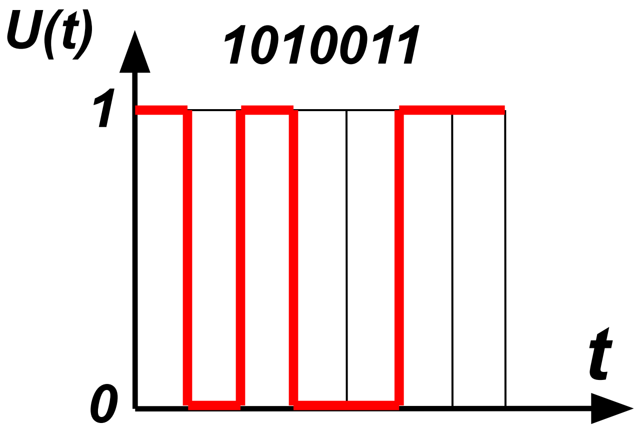
Десятичные коды:

0, 1, …, 10, 11, …, 99



Двоичные коды:

0, 1, 10, 11, 100, 101, …, 1100011



Равномерные и неравномерные коды

Равномерные коды **– коды, при использовании которых, длина всех кодовых комбинаций (кодовых слов) одинакова.**

001, 010, 011, 100 – равномерные коды.

1, 10, 11, 100 – неравномерные коды.

Системы счисления

Шестнадцате-ричная	Десятичная	Восьмеричная	Двоичная
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

Непомехозащищенные коды

Непомехозащищенные коды **– коды, содержащие кодовые комбинации, отличающиеся друг от друга в одном разряде.**

0010 и 0011 отличаются в первом разряде;

1110 и 0110 – в четвертом разряде;

1110 и 0100 – во втором и четвертом разрядах.

Двоичный код на все комбинации

Кодовые комбинации соответствуют записи натурального ряда чисел в двоичной системе счисления.

Пример: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

Количество кодовых комбинаций: $N = 2^n$

Единично-десятичный код

Каждый разряд десятичного числа записывается в виде соответствующего числа единиц; разряды при передаче по каналу связи разделяются интервалами.

Неравномерный единично-десятичный код
352: 111 11111 11
149: 1 1111 111111111

Равномерный единично-десятичный код
352: 000000111 000011111 000000011
149: 000000001 000001111 111111111

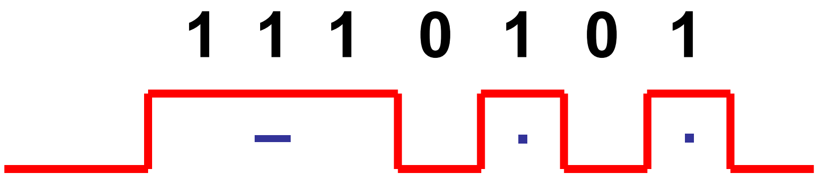
Двоично-десятичный код

Каждый разряд десятичного числа записывается в виде комбинации двоичного кода.

Пример
352: 0011 0101 0011
149: 0001 0100 1001

Код Морзе

Неравномерный код, в котором сигналы передаются в виде точек и тире.



А	·–	A	И	··	I	С	···	S	Щ	––·–	Q
Б	–···	B	К	–·–	K	Т	–	T	Ы	–·––	Y
В	·––	W	Л	·–··	L	У	··–	U	Ю	··––	U
Г	––·	G	М	––	M	Ф	··–·	F	Я	·–·–	
Д	–··	D	Н	–·	N	Х	····	H	Й	·–––	J
Е	·	E	О	–––	O	Ц	–·–·	C	ЬЪ	–··–	X
Ж	···–	V	П	·––·	P	Ч	–––·		Э	··–··	
З	––··	Z	Р	·–·	R	Ш	––––				

Помехозащищенные коды

Помехозащищенные коды (корректирующие коды) **– коды, позволяющие обнаружить ошибки в кодовых комбинациях.**

Помехозащищенные коды разделяются на две группы:

- ***коды с обнаружением ошибок;***
- ***коды с обнаружением и исправлением ошибок.***

Кодовое расстояние

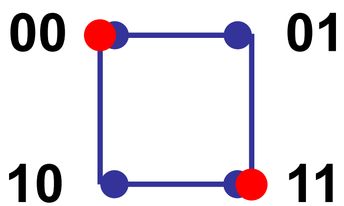
Кодовое расстояние **– минимальное число элементов, в которых могут отличаться друг от друга две кодовые комбинации в используемом коде.**

***n* = 1:**

$$\begin{array}{r} 0 \\ \oplus\; 1 \\ \hline 1 \end{array}$$

$d = 1$

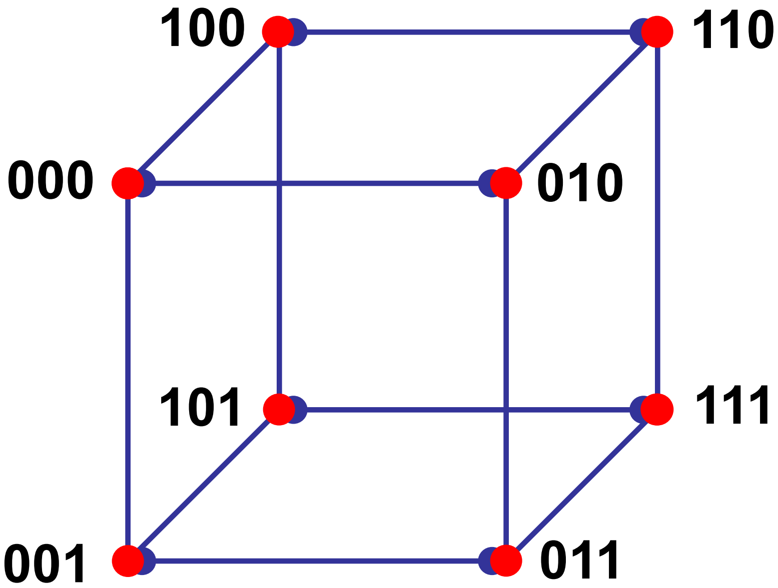
***n* = 2:**



$$\begin{array}{r} 00 \\ \oplus\; 01 \\ \hline 01 \end{array} \quad \begin{array}{r} 00 \\ \oplus\; 10 \\ \hline 10 \end{array} \quad \begin{array}{r} 00 \\ \oplus\; 11 \\ \hline 11 \end{array} \quad \begin{array}{r} 01 \\ \oplus\; 11 \\ \hline 10 \end{array} \quad \begin{array}{r} 01 \\ \oplus\; 10 \\ \hline 11 \end{array} \quad \begin{array}{r} 10 \\ \oplus\; 11 \\ \hline 01 \end{array}$$

$d = 1$ $d = 1$ $d = 2$ $d = 1$ $d = 2$ $d = 1$

Кодовые расстояния при n = 3



000 001 010 011
100 101 110 111

$$\begin{array}{r} 000 \\ \oplus\ 111 \\ \hline 111 \end{array}$$
$d = 3$

$$\begin{array}{r} 100 \\ \oplus\ 011 \\ \hline 111 \end{array}$$
$d = 3$

$$\begin{array}{r} 001 \\ \oplus\ 110 \\ \hline 111 \end{array}$$
$d = 3$

$$\begin{array}{r} 010 \\ \oplus\ 101 \\ \hline 111 \end{array}$$
$d = 3$

$$\begin{array}{r} 000 \\ \oplus\ 011 \\ \hline 011 \end{array}$$
$d = 2$

$$\begin{array}{r} 000 \\ \oplus\ 101 \\ \hline 101 \end{array}$$
$d = 2$

$$\begin{array}{r} 000 \\ \oplus\ 110 \\ \hline 110 \end{array}$$
$d = 2$

$$\begin{array}{r} 011 \\ \oplus\ 110 \\ \hline 101 \end{array}$$
$d = 2$

$$\begin{array}{r} 000 \\ \oplus\ 100 \\ \hline 100 \end{array}$$
$d = 1$

$$\begin{array}{r} 001 \\ \oplus\ 101 \\ \hline 100 \end{array}$$
$d = 1$

$$\begin{array}{r} 100 \\ \oplus\ 110 \\ \hline 010 \end{array}$$
$d = 1$

$$\begin{array}{r} 111 \\ \oplus\ 110 \\ \hline 001 \end{array}$$
$d = 1$

Корректирующая способность кода

$$d_{\min} = r + s + 1$$

$d_{\min}$ – минимальное кодовое расстояние;

r – количество обнаруживаемых ошибок;

s – количество исправляемых ошибок;

$r \geq s.$

Коды с обнаружением ошибок

$$d_{\min} \geq 2$$

- коды, построенные путем уменьшения количества используемых кодовых комбинаций;
- коды, в которых используются все возможные кодовые комбинации, но к каждой комбинации добавляются контрольные символы.

Код с постоянным числом единиц и нулей в комбинациях

Общее число кодовых комбинаций:

$$N = C_n^l = \frac{n!}{l!(n-l)!}$$

l **– число единиц в слове длиной** n**.**

C_5^2 **:** $N = 10$

11000 01010 01100 00101 00110
10010 00011 01001 10001 10100

C_7^3 **:** $N = 35$

1010100 0101010 1110000 0000111 1001001
0010101 1101000 1011000 0110100 0101100
…

Распределительный код

Код с постоянным весом, равным единице

$$N = C_n^1 = n.$$

C_5^1: **00001 00010 00100 01000 10000**

$$d_{\min} = 2$$

Код с проверкой на четность

$$n = k + 1,$$

k – количество информационных символов (разрядов) в кодовой комбинации.

Информационные символы	Контрольные символы	Кодовая комбинация
1 1 0 1 1	0	1 1 0 1 1 0
1 0 1 0 1	1	1 0 1 0 1 1
0 0 0 1 0	1	0 0 0 1 0 1
1 1 0 0 0	0	1 1 0 0 0 0
1 1 1 1 0	0	1 1 1 1 0 0
1 1 1 1 1	1	1 1 1 1 1 1

$$N = 2^k = 2^{n-1} \qquad d_{\min} = 2$$

Код с числом единиц, кратным трем

$$n = k + 2,$$

k – количество информационных символов (разрядов) в кодовой комбинации.

Информационные символы	Контрольные символы	Кодовая комбинация
1 1 0 1 1	1 1	1 1 0 1 1 1 1
1 0 1 0 1	0 0	1 0 1 0 1 0 0
0 0 0 1 0	1 1	0 0 0 1 0 1 1
1 1 0 0 0	0 1	1 1 0 0 0 0 1
1 1 1 1 0	1 1	1 1 1 1 0 1 1
1 1 1 1 1	0 1	1 1 1 1 1 0 1

$$N = 2^k = 2^{n-2} \qquad d_{\min} = 2$$

Код с удвоением элементов (корреляционный код)

Каждый элемент двоичного кода на все сочетания передается двумя символами: 1 преобразуется в 10, а 0 – в 01.

Двоичный код на все сочетания	Корреляционный код
1 1 0 1 1	1 0 1 0 0 1 1 0 1 0
1 0 1 0 1	1 0 0 1 1 0 0 1 1 0
0 0 0 1 0	0 1 0 1 0 1 1 0 0 1
1 1 0 0 0	1 0 1 0 0 1 0 1 0 1
1 1 1 1 0	1 0 1 0 1 0 1 0 0 1
1 1 1 1 1	1 0 1 0 1 0 1 0 1 0

$$N = 2^{n/2} \qquad d_{\min} = 2$$

Инверсный код

$$n = k \cdot 2,$$

k – количество информационных символов (разрядов) в кодовой комбинации.

Информационные символы	Контрольные символы	Кодовая комбинация
1 1 0 1 1	1 1 0 1 1	1 1 0 1 1 1 1 0 1 1
1 0 1 0 1	0 1 0 1 0	1 0 1 0 1 0 1 0 1 0
0 0 0 1 0	1 1 1 0 1	0 0 0 1 0 1 1 1 0 1
1 1 0 0 0	1 1 0 0 0	1 1 0 0 0 1 1 0 0 0
1 1 1 1 0	1 1 1 1 0	1 1 1 1 0 1 1 1 1 0
1 1 1 1 1	0 0 0 0 0	1 1 1 1 1 0 0 0 0 0

$$N = 2^k = 2^{n/2}$$

k	2	3	≥4
$d_{\min}$	2	3	4

Коды с обнаружением и исправлением ошибок

$$d_{\min} \geq 3$$

Образуются путем добавления к кодовой комбинации контрольных символов

- **коды Хэмминга;**
- **циклические коды;**
- **итеративные коды.**

Коды Хэмминга

В качестве исходного используется k-разрядный двоичный код на все сочетания. К нему добавляются m контрольных символов.

$$n = k + m$$

При передаче кодовой комбинации может быть искажен любой из n символов, т.е. число вариантов искажения равно n+1 (включая передачу без искажений).

$$2^m \geq n + 1 \quad \Rightarrow \quad 2^m \geq k + m + 1$$

k	1	2,3,4	5…11	12…26
m	2	3	4	5

Коды Хэмминга: кодирование и декодирование

$k_4 k_3 k_2 k_1 \Longrightarrow k_4 k_3 k_2 m_3 k_1 m_2 m_1$

Разряды двоичных чисел			Сим-волы кода
3	2	1	
0	0	1	m_1
0	1	0	m_2
0	1	1	k_1
1	0	0	m_3
1	0	1	k_2
1	1	0	k_3
1	1	1	k_4

$\Longrightarrow$

m_1	k_1	k_2	k_4
m_2	k_1	k_3	k_4
m_3	k_2	k_3	k_4

Кодиро-вание:

$$m_1 = k_1 \wedge k_2 \wedge k_4$$
$$m_2 = k_1 \wedge k_3 \wedge k_4$$
$$m_3 = k_2 \wedge k_3 \wedge k_4$$

Декоди-рование:

$$l_1 = m_1 \wedge k_1 \wedge k_2 \wedge k_4$$
$$l_2 = m_2 \wedge k_1 \wedge k_3 \wedge k_4$$
$$l_3 = m_3 \wedge k_2 \wedge k_3 \wedge k_4$$

$l_3 l_2 l_1$ – номер искаженного бита

Контрольная сумма блока данных

123 0111 1011	47 0010 1111	170 1010 1010

125 0111 1101	45 0010 1101	170 1010 1010

31535 / 271 = 116 + 99 / 271

0111 1011 0010 1111 / 100001111 = 0111 0100 (0110 0011)

32045 / 271 = 118 + 67 / 271

0111 1101 0010 1101 / 100001111 = 0111 0110 (0100 0011)

Циклические коды

101101 = $X^5 + X^3 + X^2 + 1$**,** $X=2$

Приводимый полином **– полином, который можно представить в виде произведения многочленов низших степеней**

Неприводимый полином **– полином, который нельзя представить в виде произведения многочленов низших степеней**

$P(X^1)$	$X+1$	11	3
$P(X^2)$	X^2+X+1	111	7
$P(X^3)$	X^3+X+1	1011	11
$P(X^3)$	X^3+X^2+1	1101	13
$P(X^4)$	X^4+X+1	10011	19

Сложение полиномов

При операциях с полиномами применяется сложение двоичных чисел по модулю 2, эквивалентное операции «исключающее ИЛИ» с каждым разрядом.

0^0=0 0^1=1 1^0=1 1^1=0

$$
\begin{array}{rl}
\wedge & 1\cdot X^7 + 0\cdot X^6 + 1\cdot X^5 + 0\cdot X^4 + 0\cdot X^3 + 1\cdot X^2 + 1\cdot X^1 + 0\cdot X^0 \\
 & 0\cdot X^7 + 1\cdot X^6 + 0\cdot X^5 + 0\cdot X^4 + 1\cdot X^3 + 1\cdot X^2 + 1\cdot X^1 + 1\cdot X^0 \\
\hline
 & 1\cdot X^7 + 1\cdot X^6 + 1\cdot X^5 + 0\cdot X^4 + 1\cdot X^3 + 0\cdot X^2 + 0\cdot X^1 + 1\cdot X^0
\end{array}
$$

$$
\begin{array}{rl}
\wedge & 1010\ 0110 \\
 & 0100\ 1111 \\
\hline
 & 1110\ 1001
\end{array}
$$

Деление полиномов

Деление двоичных полиномов аналогично делению целых чисел. При этом операция вычитания эквивалентна операции «исключающее ИЛИ».

```
11100110   | 1010
1010       |------
 1000      | 11010
 1010
  01011
   1010
    0010
```

Метод построения циклического кода

$G(X)$ – исходная кодовая комбинация

$P(X)$ – образующий полином

$$\frac{G(X)X^m}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

X^m – одночлен той же степени, что и $P(X)$

$Q(X)$ – частное от деления

$R(X)$ – остаток от деления

$$F(X) = G(X)X^m + R(X)$$

$F(X)$ – закодированное сообщение

Пример построения циклического кода

$P(X) = X + 1 \rightarrow 11$

$G(X) = X^2 + X \rightarrow 0110$

$G(X) \cdot X^1 \rightarrow 01100$

```
01100 | 11
 11   |-----
 0000 | 0100
```

$F(X) \rightarrow 01100$

$G(X) = X^3 + X + 1 \rightarrow 1011$

$G(X) \cdot X^1 \rightarrow 10110$

```
10110 | 11
11    |-----
 11   | 1101
 11
  010
   11
    1
```

$F(X) \rightarrow 10111$

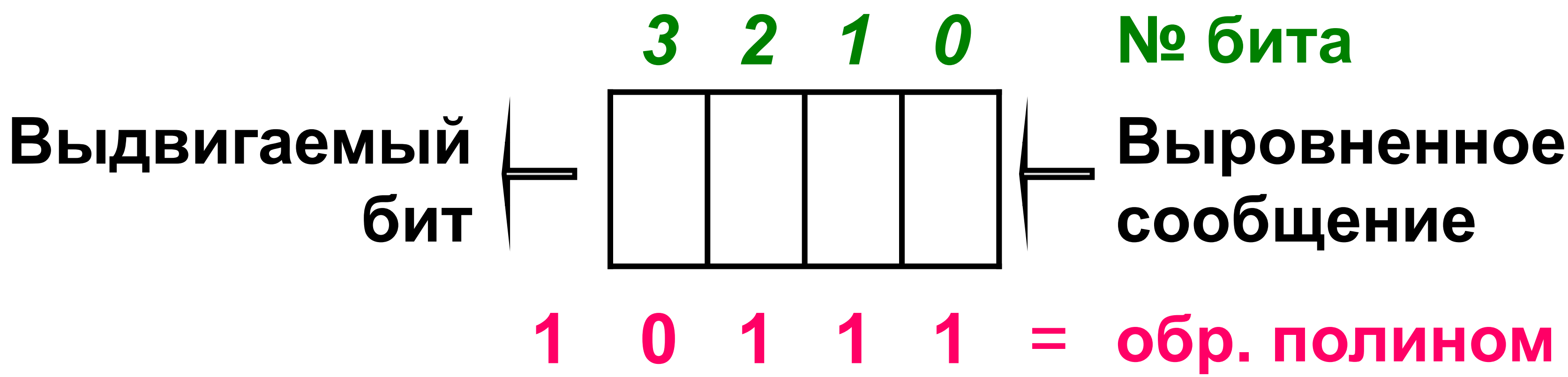
Пример циклического кода

$$P(X) = X + 1 \to 11$$

0→00000

1→00011	2→00101	7→01111
3→00110	5→01010	F→11110
6→01100	A→10100	E→11101
C→11000	4→01001	D→11011
8→10001	9→10010	B→10111

Алгоритм построения циклического кода



1. $R = 0$
2. В хвостовую часть сообщения добавляется m нулевых битов
3. Сдвиг влево на 1 бит
4. Если выдвинут бит со значением 1, $R=R$^$P(X)$
5. Если обработаны не все биты, переход к п.3

Алгоритм построения циклического кода

$P(X) = X^4 + X + 1 \rightarrow 10011$

$G(X) \rightarrow 1010\ \ 0110$

```
1010 0110 0000 | 10011
1001 1         |-----------
 011 111       | 10111010
  10 011
   1 1000
   1 0011
     1011 0
     1001 1
      010 100
       10 011
        0 1110
```

$F(X) \rightarrow 1010\ \ 0110\ \ 1110$

```
0 1010 0110 0000
1 0100 110 0000
----------------
  0111 110 0000
0 1111 10 0000
1 1111 0 0000
----------------
  1100 0 0000
1 1000 0000
----------------
  1011 0000
1 0110 000
----------------
  0101 000
0 1010 00
1 0100 0
----------------
  0111 0
0 1110
```

Выявление ошибок в блоке данных при помощи избыточного циклического кода (CRC)

Контрольные символы добавляются в начало или конец блока данных. Комбинация контрольных символов называется контрольной суммой (CRC).

CRC-12	$X^{12}+X^{11}+X^{3}+X^{2}+X+1$	1 80F
CRC-16[1]	$X^{16}+X^{15}+X^{2}+1$	1 8005
CRC-16[2]	$X^{16}+X^{15}+ X^{13}+1$	1 A001
CRC-CCITT	$X^{16}+X^{12}+X^{5}+1$	1 1021
CRC-32[1]		1 04c11db7
CRC-32[2]		1 edb88320

Алгоритм вычисления 16-битного избыточного циклического кода

1. CRC = FFFF
2. С использованием значения *X* очередного байта выполняется операция CRC=CRC^*X*
3. Сохраняется значение младшего бита CRC: L=CRC&1
4. CRC сдвигается вправо на 1 бит
5. Если L=1, выполняется операция CRC=CRC^A001
6. Если выполнено меньше 8 сдвигов CRC, происходит переход к п.3
7. Если обработаны не все байты блока данных, происходит переход к п.2

Вычисление 16-битного избыточного циклического кода на языке C++

```
unsigned CalcCRC( unsigned char *Buf, unsigned Len ) {
  unsigned CRC = 0xFFFF;
  for( unsigned i=0; i<Len; ++i, ++Buf ) {
    CRC ^= *Buf;
    for( unsigned j=0; j<8; ++j ) {
      bool LSB = CRC & 0x0001;
      CRC >>= 1;
      if(LSB) CRC ^= 0xA001;
    }
  }
  return CRC;
}
```