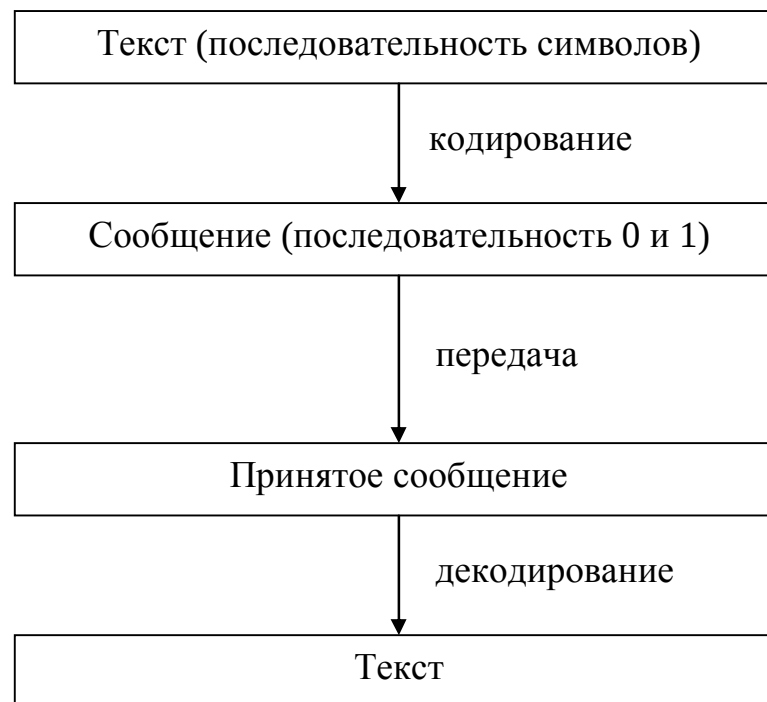


Кодирование



Задача 1. Кодирование в сообщение минимальной длины

Кодирование Хаффмана

Дано: вероятности появления символов

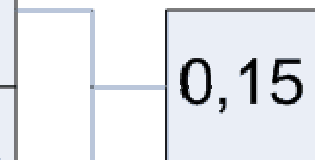
Символ	A	B	C	D	E	F
Вероятность	0,05	0,45	0,1	0,1	0,15	0,15

Отсортируем по убыванию вероятностей:

В	0,45
Е	0,15
F	0,15
С	0,1
D	0,1
A	0,05

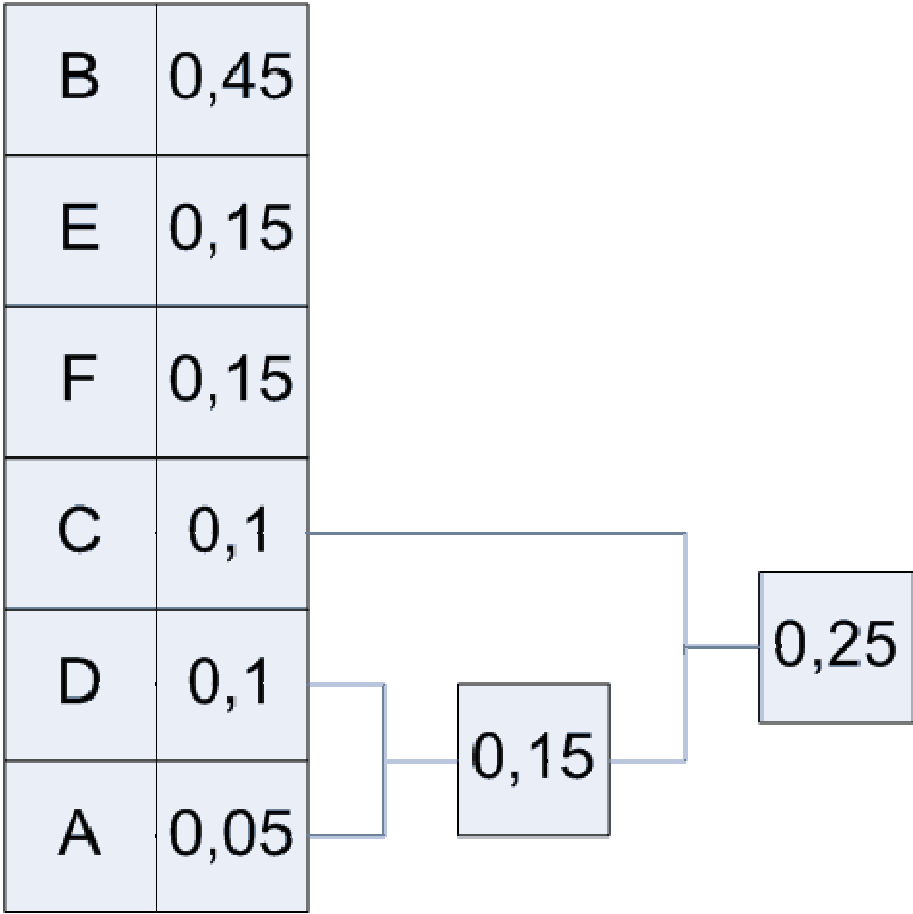
Объединим два наименьших значения:

В	0,45
Е	0,15
F	0,15
С	0,1
D	0,1
A	0,05

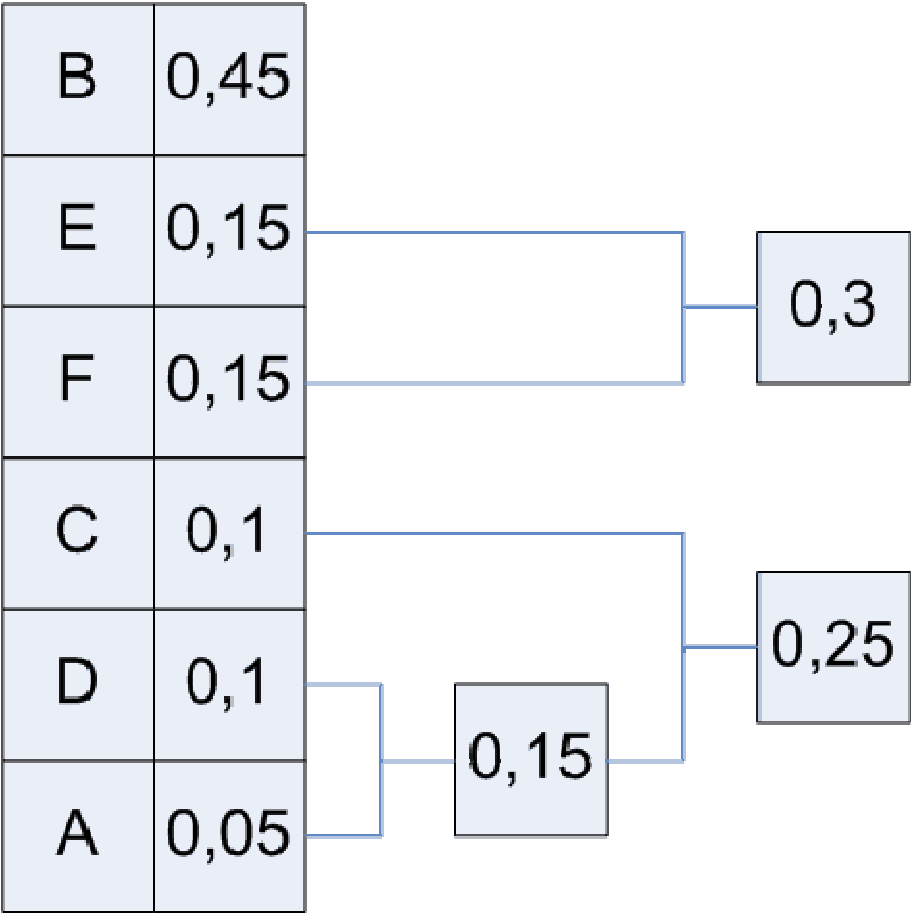


The diagram illustrates the merging of the two smallest probabilities from the table above. A bracket connects the values 0,1 (from row D) and 0,05 (from row A). A line from this bracket points to a separate box containing the value 0,15, representing the combined probability.

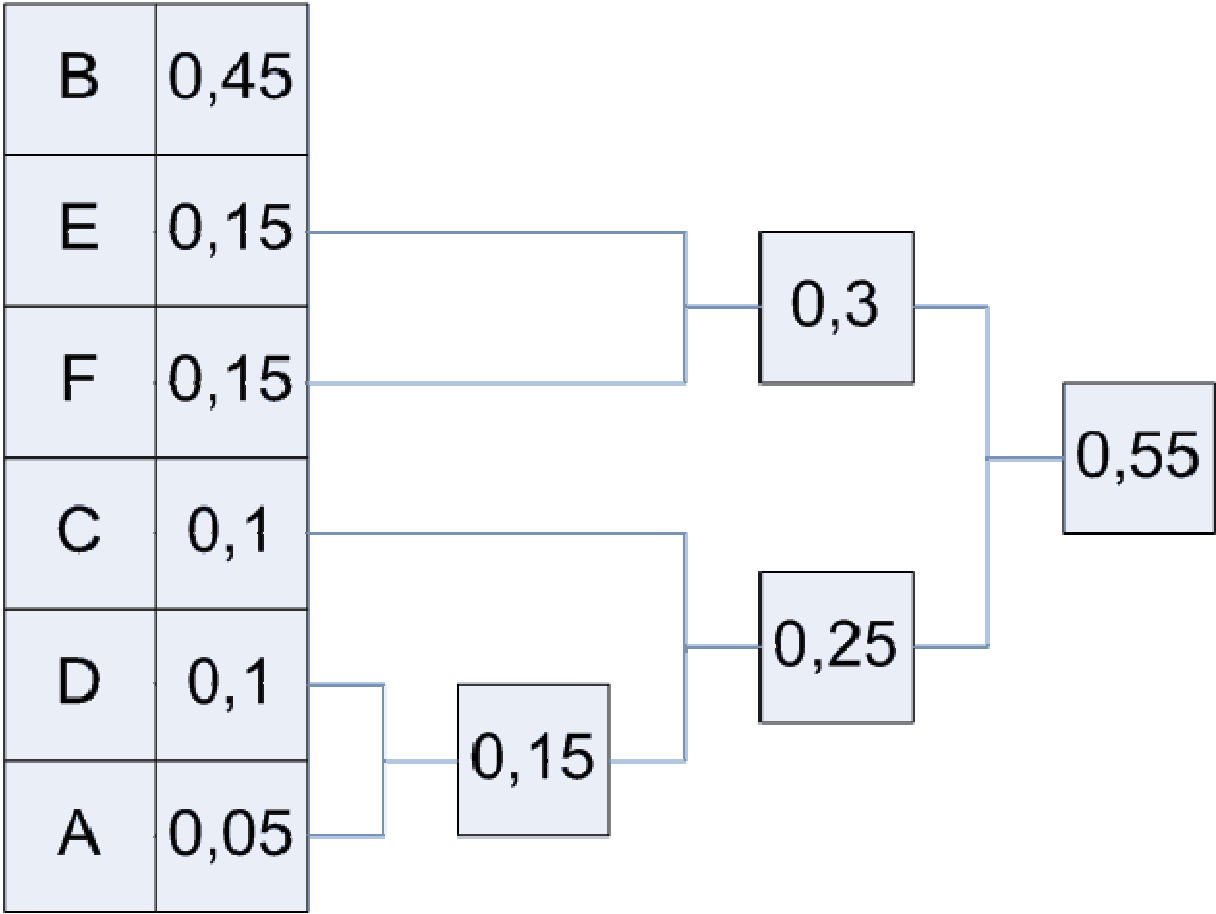
Объединим два наименьших значения:



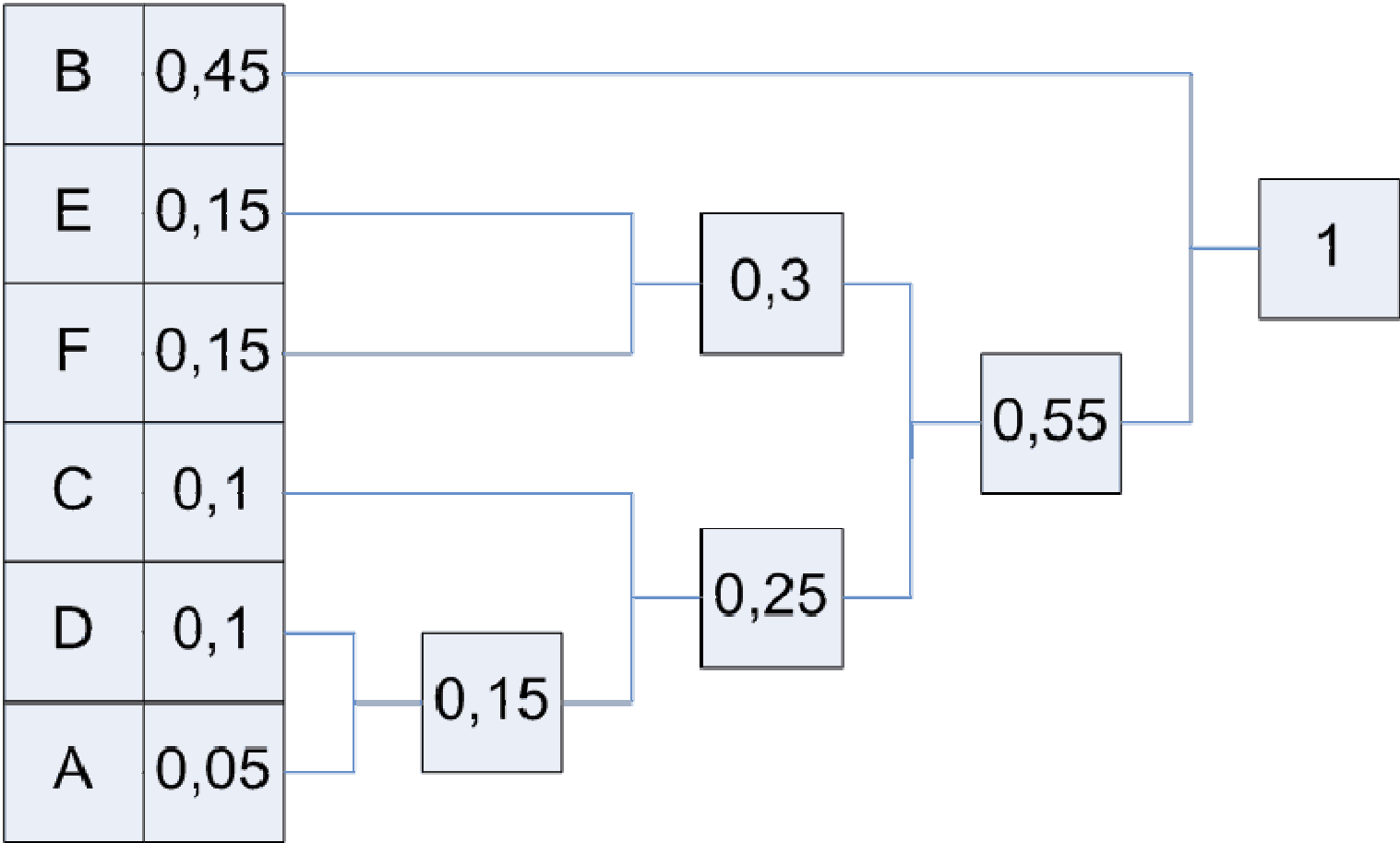
Объединим два наименьших значения:



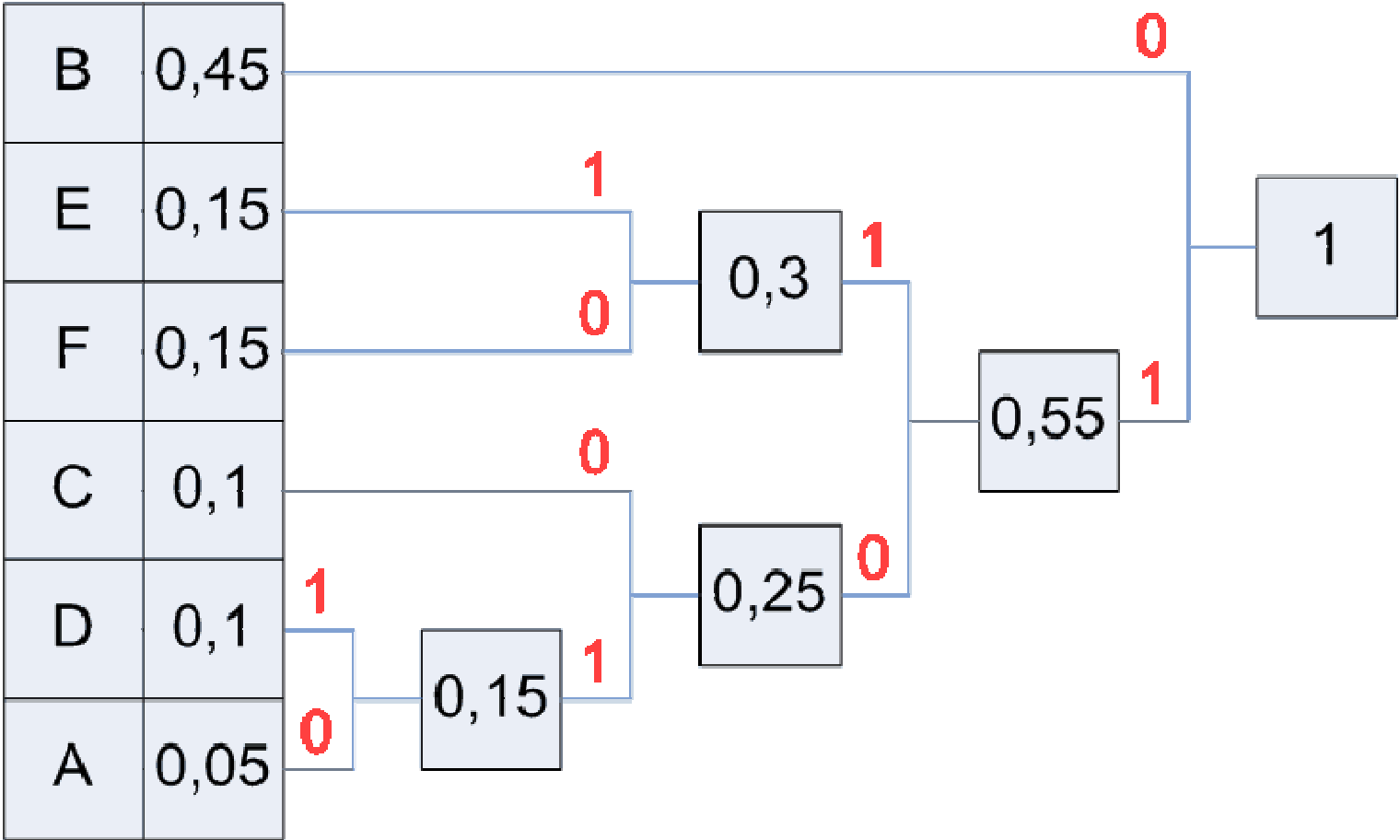
Объединим два наименьших значения:



Объединим два наименьших значения:



Расставим 0 и 1 каким-нибудь способом, например, 0 – для меньшей вероятности, 1 – для большей вероятности:



Запишем коды символов, **справа налево**:

Символ	A	B	C	D	E	F
Код	1010	0	100	1011	111	110

Рассчитаем среднюю длину:

Символ	A	B	C	D	E	F
Код	1010	0	100	1011	111	110
Длина	4	1	3	4	3	3
Вероятность	0,05	0,45	0,1	0,1	0,15	0,15

$$\bar{L} = 0,05 \cdot 4 + 0,45 \cdot 1 + 0,1 \cdot 3 + 0,1 \cdot 4 + 0,15 \cdot 3 + 0,15 \cdot 3 = 2,25$$

Label	Probability
B	0,45
E	0,15
F	0,15
C	0,1
D	0,1
A	0,05

Decision Tree Structure:

- Root Node (Table) branches into:
 - Branch 0 (Top): Leads to a node with probability 0,3.
 - Branch 0: Leads to a node with probability 0,55.
 - Branch 1: Leads to a node with probability 0,25.
 - Branch 1 (Bottom): Leads to a node with probability 0,15.
- Intermediate nodes (0,3, 0,55, 0,25, 0,15) branch into:
 - Branch 0: Leads to the final leaf node labeled '1'.
 - Branch 1: Leads to the final leaf node labeled '1'.

Символ	A	B	C	D	E	F
Код	1111	0	110	1110	100	101

Пример декодирования

Принятое сообщение:

...	0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	1	0	0	...
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----

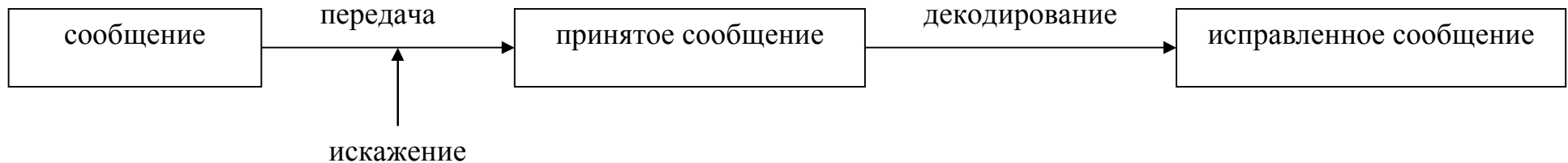
Коды символов:

Символ	A	B	C	D	E	F
Код	1111	0	110	1110	100	101

Результат:

0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	0	0
B	A? C? D? E? F?	A? C? D?	A? D?	A	A? C? D? E? F?	A? C? D?	A? D?	D	A? C? D? E? F?	E? F?	F	A? C? D? E? F?	A? C? D?	A? D?	A	A? C? D? E? F?	A? C? D?	C	A? C? D? E? F?	E? F?	E
B	A			D				F			A			C			E				

Задача 2. Исправление ошибок



Код Хэмминга

Исходное сообщение: k разрядов

Передаваемое сообщение: k -разрядный код + m контрольных символов

Длина сообщения: $n = k + m$

Тип ошибок, которые может исправить код: искажение одного символа.

Передаваемое сообщение: $a_1 a_2 \dots a_{n-1} a_n$

Возможные принимаемые сообщения:

$b_1 a_2 \dots a_{n-1} a_n$,
 $a_1 b_2 \dots a_{n-1} a_n$,
...

n вариантов с искажением
+ вариант отсутствия искажений

Итого $n + 1$ вариантов принимаемых сообщений для каждого передаваемого сообщения.

$a_1 a_2 \dots b_{n-1} a_n$,

$a_1 a_2 \dots a_{n-1} b_n$,

$a_1 a_2 \dots a_{n-1} a_n$

Для того, чтобы код мог исправлять такую ошибку, необходимо:

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

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

$k = 1$	$m = 2$	$n = 3$
$k = 2, 3, 4$	$m = 3$	$n = 5, 6, 7$
$k = 5..11$	$m = 4$	$n = 9..15$
$k = 12..26$	$m = 5$	$n = 17..31$

$$k_4 k_3 k_2 k_1 \longrightarrow k_4 k_3 k_2 \overset{4}{m_3} k_1 \overset{2}{m_2} \overset{1}{m_1}$$

Пример 1.

Принят код 0111011 (слева – младший разряд, справа – старший)

Восстановить переданный код сообщения и исходное сообщение (информационную часть), если кодирование было осуществлено методом Хэмминга и в сообщении было не более одной ошибки типа замещения разряда.

1. Декодирование

0	1	1	1	0	1	1
---	---	---	---	---	---	---

1 2 3 4 5 6 7
 m_1 m_2 k_1 m_3 k_2 k_3 k_4

Номера разрядов, значение которых равно 1, запишем в двоичном виде.

0	1	1	1	0	1	1
---	---	---	---	---	---	---

1 2 3 4 5 6 7

	010	011	100		110	111
--	-----	-----	-----	--	-----	-----

Просуммируем (алгебраически, без переносов)

1 1 1

1 1 0

1 0 0

0 1 1

0 1 0

1 0 0

100 – следовательно, искажён 4-й бит

1 2 3 4 5 6 7

принятое сообщение

0	1	1	1	0	1	1
---	---	---	---	---	---	---

исправленное сообщение

0	1	1	0	0	1	1
---	---	---	---	---	---	---

2. Проверим: рассмотрим код 0110011 – если бы искажения не было.

Номера разрядов, значение которых равно 1, запишем в двоичном виде и просуммируем.

0	1	1	0	0	1	1
---	---	---	---	---	---	---

1 2 3 4 5 6 7

	010	011			110	111
--	-----	-----	--	--	-----	-----

1 1 1

1 1 0

0 1 1

0 1 0

0 0 0

000 – значит, не было искажений

3. Выделим информационную часть

Переданное сообщение:

0	1	1	0	0	1	1
---	---	---	---	---	---	---

1 2 3 4 5 6 7

m_1 m_2 k_1 m_3 k_2 k_3 k_4

Исходное сообщение (информационная часть):

1	0	1	1
---	---	---	---

k_1 k_2 k_3 k_4

4. Проверим: запишем процедуру кодирования.

Исходное сообщение (информационная часть):

1	0	1	1
---	---	---	---

k_1 k_2 k_3 k_4

*	*	1	*	0	1	1
---	---	---	---	---	---	---

1 2 3 4 5 6 7
 m_1 m_2 k_1 m_3 k_2 k_3 k_4

0	1	1	0	0	1	1
---	---	---	---	---	---	---

1 2 4
 m_1 m_2 m_3

Номера разрядов, значение которых равно 1, запишем в двоичном виде и просуммируем

1	1	1
1	1	0
0	1	1
0	1	0
m_3	m_2	m_1

Пример 2.

Принят код 010101 (слева – младший разряд, справа – старший)

Восстановить переданный код сообщения и исходное сообщение (информационную часть), если кодирование было осуществлено методом Хэмминга и в сообщении было не более одной ошибки типа замещения разряда.

Декодирование

0	1	0	1	0	1
---	---	---	---	---	---

1 2 3 4 5 6
 m_1 m_2 k_1 m_3 k_2 k_3

Номера разрядов, значение которых равно 1, запишем в двоичном виде.

0	1	0	1	0	1
---	---	---	---	---	---

1 2 3 4 5 6

	010		100		110
--	-----	--	-----	--	-----

Просуммируем (алгебраически, без переносов)

1 1 0

1 0 0

0 1 0

0 0 0

000 – следовательно, не было искажений

	1	2	3	4	5	6
принятое сообщение	0	1	0	1	0	1
переданное сообщение	0	1	0	1	0	1
исходное сообщение			0		0	1

Проверим: запишем процедуру кодирования.

*	*	0	*	0	1
1	2	3	4	5	6
m_1	m_2	k_1	m_3	k_2	k_3

Номера разрядов, значение которых равно 1, запишем в двоичном виде и просуммируем

1	1	0
1	1	0
m_3	m_2	m_1

0	1	0	1	0	1
1	2		4		
m_1	m_2		m_3		

Пример 3.

Принят код 110010 (слева – младший разряд, справа – старший)

Восстановить переданный код сообщения и исходное сообщение (информационную часть), если кодирование было осуществлено методом Хэмминга и в сообщении было не более одной ошибки типа замещения разряда.

Декодирование

1	1	0	0	1	0
1	2	3	4	5	6
m_1	m_2	k_1	m_3	k_2	k_3

Номера разрядов, значение которых равно 1, запишем в двоичном виде.

1	1	0	0	1	0
1	2	3	4	5	6
001	010			101	

Просуммируем (алгебраически, без переносов)

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

110 – следовательно, искажён 6-й бит

	1	2	3	4	5	6
принятое сообщение	1	1	0	0	1	0
переданное сообщение	1	1	0	0	1	1
исходное сообщение			0		1	1

Проверим: запишем процедуру кодирования.

*	*	0	*	1	1
1	2	3	4	5	6
m_1	m_2	k_1	m_3	k_2	k_3

1	1	0	0	1	1
1	2		4		
m_1	m_2		m_3		

Номера разрядов, значение которых равно 1, запишем в двоичном виде и просуммируем

1	1	0
1	0	1
0	1	1
m_3	m_2	m_1

Циклические коды

Для построения циклического кода необходим так называемый образующий (производящий) многочлен (полином), который должен быть примитивным неразлагаемым.

Образующие многочлены для обнаружения единичных ошибок:

$k = 2, 3, 4$	$m = 3$	$n = 5, 6, 7$	$x^3 + x + 1$
$k = 5..11$	$m = 4$	$n = 9..15$	$x^4 + x + 1$
$k = 12..26$	$m = 5$	$n = 17..31$	$x^5 + x^4 + 1$

Пример.

Часть 1. Кодирование

Сообщение: 1011 (слева – младший разряд, справа – старший)

Используем образующий многочлен: $g(x) = x^3 + x + 1$

1	0	1	1
---	---	---	---

x^0 x^1 x^2 x^3

$$A(x) = x^3 + x^2 + 1$$

$$k = 4 \Rightarrow m = 3, n = 7$$

Сдвиг информационной части:

$$x^m A(x) = x^6 + x^5 + x^3$$

			1	0	1	1
--	--	--	---	---	---	---

x^0 x^1 x^2 x^3 x^4 x^5 x^6

Разделим $x^m A(x) = x^6 + x^5 + x^3$ на $g(x) = x^3 + x + 1$:

$$\begin{array}{r|l} x^6 + x^5 + x^3 & x^3 + x + 1 \\ x^6 + x^4 + x^3 & x^3 \\ \hline x^5 + x^4 & \end{array}$$

Разделим $x^m A(x) = x^6 + x^5 + x^3$ на $g(x) = x^3 + x + 1$:

$$\begin{array}{r|l} x^6 + x^5 + x^3 & x^3 + x + 1 \\ \hline x^6 + x^4 + x^3 & x^3 + x^2 \\ \hline x^5 + x^4 & \\ x^5 + x^3 + x^2 & \\ \hline x^4 + x^3 + x^2 & \end{array}$$

Разделим $x^m A(x) = x^6 + x^5 + x^3$ на $g(x) = x^3 + x + 1$:

$$\begin{array}{r|l} x^6 + x^5 + x^3 & x^3 + x + 1 \\ \hline x^6 + x^4 + x^3 & x^3 + x^2 + x \\ \hline x^5 + x^4 & \\ x^5 + x^3 + x^2 & \\ \hline x^4 + x^3 + x^2 & \\ x^4 + x^2 + x & \\ \hline x^3 + x & \end{array}$$

Разделим $x^m A(x) = x^6 + x^5 + x^3$ на $g(x) = x^3 + x + 1$:

$$\begin{array}{r|l} x^6 + x^5 + x^3 & x^3 + x + 1 \\ \hline x^6 + x^4 + x^3 & x^3 + x^2 + x + 1 \\ \hline x^5 + x^4 & \\ x^5 + x^3 + x^2 & \\ \hline x^4 + x^3 + x^2 & \\ x^4 + x^2 + x & \\ \hline x^3 + x & \\ x^3 + x + 1 & \\ \hline 1 & \end{array}$$

Получили

$$p(x) = x^3 + x^2 + x + 1$$

$$r(x) = 1$$

Передаваемое сообщение:

$$p(x) \cdot g(x) = (x^3 + x^2 + x + 1) \cdot (x^3 + x + 1) = x^6 + x^5 + x^3 + 1$$

1	0	0	1	0	1	1
---	---	---	---	---	---	---

x^0 x^1 x^2 x^3 x^4 x^5 x^6

$r(x) = 1$ $x^m A(x) = x^6 + x^5 + x^3$

Часть 2. Декодирование

Пусть в принятом сообщении ошибка в 3 слева разряде

переданное сообщение

1	0	0	1	0	1	1
x^0	x^1	x^2	x^3	x^4	x^5	x^6

принятое сообщение

1	0	1	1	0	1	1
---	---	---	---	---	---	---

$B(x) = x^6 + x^5 + x^3 + x^2 + 1$

Разделим $B(x) = x^6 + x^5 + x^3 + x^2 + 1$ на $g(x) = x^3 + x + 1$:

$x^6 + x^5 + x^3 + x^2 + 1$

$x^6 + x^4 + x^3$

$x^5 + x^4 + x^2 + 1$

$x^5 + x^3 + x^2$

$x^4 + x^3 + 1$

$x^4 + x^2 + x$

$x^3 + x^2 + x + 1$

$x^3 + x + 1$

x^2

$r^*(x) = x^2$

Код ошибки

0	0	1
x^0	x^1	x^2

Эту процедуру можем повторить для всех вариантов ошибок:

Переданное сообщение	Ошибка	Принятое сообщение	Многочлен в остатке	Код ошибки
1001011	—	1001011	0	000
	в первом слева разряде	0001011	1	100
	во втором слева разряде	1101011	x	010
	в третьем слева разряде	1011011	x^2	001
	в четвёртом слева разряде	1000011	$x + 1$	110
	в пятом слева разряде	1001111	$x^2 + x$	011
	в шестом слева разряде	1001001	$x^2 + x + 1$	111
	в седьмом слева разряде	1001010	$x^2 + 1$	101

(слева — младший разряд, справа — старший)

Почему код называется «циклическим»?

Рассмотрим случай без ошибок.

переданное сообщение	1	0	0	1	0	1	1
	x^0	x^1	x^2	x^3	x^4	x^5	x^6
сдвиг →	1	1	0	0	1	0	1
сдвиг →	1	1	1	0	0	1	0
сдвиг →	0	1	1	1	0	0	1
сдвиг →	1	0	1	1	1	0	0
сдвиг →	0	1	0	1	1	1	0
сдвиг →	0	0	1	0	1	1	1
сдвиг →	1	0	0	1	0	1	1

$$\begin{array}{r}
 1\ 1\ 0\ 1\ 0\ 0\ 1\ |\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 1\ 1\ 0\ 0\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 1\ 1\ 1\ 0\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 1\ 0\ 1\ 1\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0
 \end{array}$$

→

$$\begin{array}{r}
 1\ 0\ 1\ 0\ 0\ 1\ 1\ |\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0\ 0\ 1\ 0\ 1\ 1\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0
 \end{array}$$

→

$$\begin{array}{r}
 0\ 1\ 0\ 0\ 1\ 1\ 1\ |\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0\ 1\ 0\ 1\ 1\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0
 \end{array}$$

↓

$$\begin{array}{r}
 1\ 0\ 0\ 1\ 1\ 1\ 0\ |\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0\ 1\ 0\ 1\ 1\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0\ 0
 \end{array}$$

↓

$$\begin{array}{r}
 0\ 0\ 1\ 1\ 1\ 0\ 1\ |\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 1\ 0\ 1\ 1\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0
 \end{array}$$

←

$$\begin{array}{r}
 0\ 1\ 1\ 1\ 0\ 1\ 0\ |\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 1\ 0\ 1\ 1\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0\ 0
 \end{array}$$

←

$$\begin{array}{r}
 1\ 1\ 1\ 0\ 1\ 0\ 0\ |\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 1\ 0\ 1\ 1\ \\
 1\ 0\ 1\ 1\ \underline{\hspace{1cm}} \\
 0\ 0\ 0
 \end{array}$$

↑

Таким образом, при любом сдвиге код остаётся корректным. Если нет ошибок, то остаток всегда равен 0.

Более того, если есть ошибка в каком-то разряде, то код ошибки для каждого разряда не зависит от сдвига.

Нет ошибок	Ошибка в 1 разряде	Ошибка во 2 разряде	Ошибка в 3 разряде	Ошибка в 4 разряде	Ошибка в 5 разряде	Ошибка в 6 разряде	Ошибка в 7 разряде
1101001	1101000	1101011	1101101	1100001	1111001	1001001	0101001
сдвиг →							
1010011	1010010	1010001	1010111	1011011	1000011	1110011	0010011
сдвиг →							
0100111	0100110	0100101	0100011	0101111	0110111	0000111	1100111
сдвиг →							
1001110	1001111	1001100	1001010	1000110	1011110	1101110	0001110
сдвиг →							
0011101	0011100	0011111	0011001	0010101	0001101	0111101	1011101
сдвиг →							
0111010	0111011	0111000	0111110	0110010	0101010	0011010	1111010
сдвиг →							
1110100	1110101	1110110	1110000	1111100	1100100	1010100	0110100
код ошибки 000	код ошибки 001	код ошибки 010	код ошибки 100	код ошибки 011	код ошибки 110	код ошибки 111	код ошибки 101

(справа – младший разряд, слева – старший)