



Алфавит – конечное множество различных знаков (букв), символов, для которых определена операция конкатенации (присоединения символа к символу или цепочке символов).

Знак (буква) – любой элемент алфавита (элемент x алфавита X , где $x \in X$).

Слово – конечная последовательность знаков (букв) алфавита.

Словарь (словарный запас) – множество различных слов над алфавитом.



Кодирование данных — процесс преобразования символов алфавита X в символы алфавита Y.

Декодирование — процесс, обратный кодированию.

Символ — наименьшая единица данных, рассматриваемая как единое целое при кодировании/декодировании.

Кодовое слово – последовательность символов из алфавита Y, однозначно обозначающая конкретный символ алфавита X.

Средняя длина кодового слова – это величина, которая вычисляется как взвешенная вероятностями сумма длин всех кодовых слов.

$$L = \sum_{i=1}^N p_i * l_i$$

Если все кодовые слова имеют одинаковую длину, то код называется **равномерным** (фиксированной длины).

Если встречаются слова разной длины, то – **неравномерным** (переменной длины).



Сжатие данных — процесс, обеспечивающий уменьшение объёма данных путём сокращения их избыточности.

Сжатие данных — частный случай кодирования данных.

Коэффициент сжатия — отношение размера входного потока к выходному потоку.

Отношение сжатия — отношение размера выходного потока ко входному потоку.

Пример. Размер входного потока равен 500 бит, выходного равен 400 бит.

Коэффициент сжатия = $500 \text{ бит} / 400 \text{ бит} = 1,25$.

Отношение сжатия = $400 \text{ бит} / 500 \text{ бит} = 0,8$.

Случайные данные невозможно сжать, так как в них нет никакой избыточности.



Сжатие без потерь (полностью обратимое) — сжатые данные после декодирования (распаковки) не отличаются от исходных.

Сжатие с потерями (частично обратимое) — сжатые данные после декодирования (распаковки) отличаются от исходных, так как при сжатии часть исходных данных была отброшена для увеличения коэффициента сжатия.

Статистические методы — кодирование с помощью усреднения вероятности появления элементов в закодированной последовательности.

Словарные методы — использование статистической модели данных для разбиения данных на слова с последующей заменой на их индексы в словаре.

Теорема Шеннона об источнике шифрования

Теорема Шеннона об источнике шифрования устанавливает предел максимального сжатия данных и числовое значение энтропии (меры) Шеннона: невозможно сжать данные настолько, что оценка кода (среднее число бит на символ) меньше, чем энтропия Шеннона исходных данных, без потери точности информации.

$$i(S) = -\sum_{i=1}^N p_i \cdot \log_2 p_i$$

P1	P2	Энтропия
0.50	0.50	1.00
0.60	0.40	0.97
0.70	0.30	0.88
0.80	0.20	0.72
0.90	0.10	0.47
0.99	0.01	0.08

Теорема Шеннона об источнике шифрования (2)



Пример. Дан алфавит «ABCDE» с вероятностями встречаемости символов 0,4, 0,2, 0,2, 0,1 и 0,1 соответственно.

Вероятность строки «AAAABBBCCDE» = $0,4^4 * 0,2^2 * 0,2^2 * 0,1^1 * 0,1^1 =$
 $= 4,096 * 10^{-7}$.

$\log_2 P = -21.21928$.

Наименьшее в среднем число для кодирования строки равно 22 бит.

Энтропийный кодер — кодер, достигающий сжатия максимально близкого к энтропии.



Роберт
Фано
(1917–2016)

Условие Фано: если в код входит слово a , то для любой непустой строки b слова ab в коде не существует.

Символ	Вероятность	Код 1	Код 2
a_1	0,5	1	1
a_2	0,3	01	01
a_3	0,1	010	000
a_4	0,1	001	001

$a_1 \ a_3 \ a_2 \ a_1 a_1 \ a_4 \ a_2 \ a_2 \ a_1 a_1$

1|010|01|1|1|001|01|01|1|1 — Код 1

1|000|01|1|1|001|01|01|1|1 — Код 2

Префиксный код — это код, в котором никакое кодовое слово не является префиксом любого другого кодового слова. Эти коды имеют переменную длину.

Оптимальный префиксный код — это префиксный код, имеющий минимальную среднюю длину.



Дана последовательность символов: AAABCCCCDEEEFG

Встречаемость символов: A = 3, B = 1, C = 4, D = 1, E = 3, F = 1, G = 1.

Построим таблицу с вероятностями, отсортируем в порядке уменьшения вероятностей.

Символ	Вероятность
A	3/14
B	1/14
C	4/14
D	1/14
E	3/14
F	1/14
G	1/14

Символ	Вероятность
C	4/14
A	3/14
E	3/14
B	1/14
D	1/14
F	1/14
G	1/14

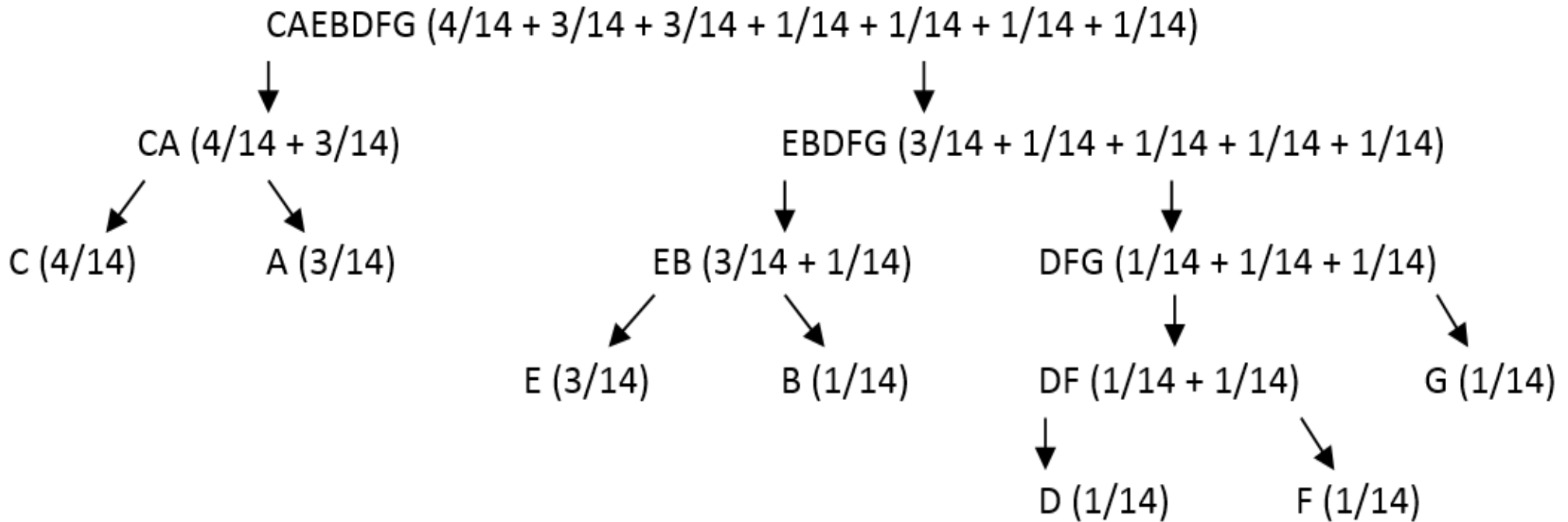
Построим кодовое дерево по методу Шеннона-Фано (от корня к листьям).

Разбивать на каждом уровне дерева на 2 ветки с максимально близкой суммарной вероятностью.

Алгоритм Шеннона-Фано (2)



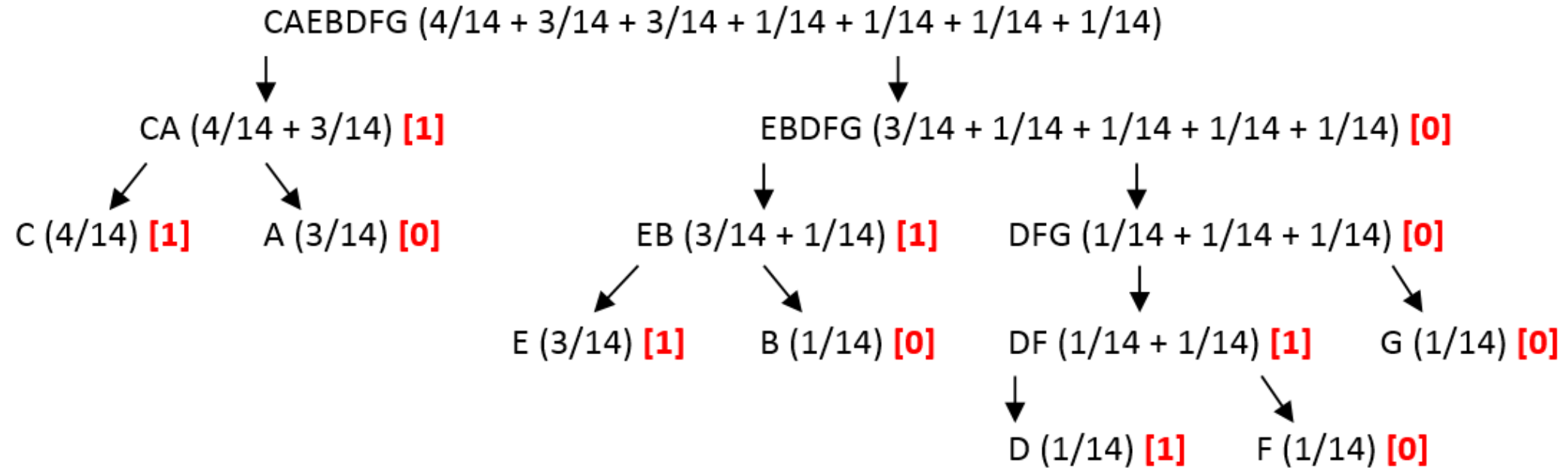
Ветвление осуществлять используя последовательные символы, полученные в результате сортировки вероятностей.
(т.е. CA + EBDGFG, а не CBDF + AEG).



Алгоритм Шеннона-Фано (3)



Левому символу (с большей вероятностью) присвоим значение 1, правому – 0.





Составим таблицу кодировки.

СИМВОЛ	1	2	3	4	Код
C	1	11			11
A		10			10
E	0	01	011		011
B			010		010
D		00	001	0011	0011
F				0010	0010
G			000		000

Дана последовательность символов: AAABCCCCDEEEFG

10.10.10.010.11.11.11.11.0011.011.011.011.0010.000
— 37 бит

Коэффициент сжатия = $(14 * 16 \text{ бит}) / 37 \text{ бит} \approx 6,054$.

Средняя длина кодового слова = $(C) 4/14 * 2 + (A) 3/14 * 2 + (E) 3/14 * 3 + (B) 1/14 * 3 + (D) 1/14 * 4 + (F) 1/14 * 4 + (G) 1/14 * 3 = 8/14 + 6/14 + 9/14 + 3/14 + 4/14 + 4/14 + 3/14 = 37 / 14 \approx 2,643 \text{ бит/символ}$



Дана последовательность символов:

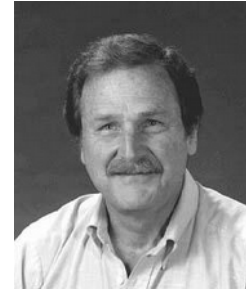
AAABCCCCDEEEFG

Встречаемость символов:

A = 3, B = 1, C = 4, D = 1, E = 3, F = 1, G = 1.

Построим таблицу с вероятностями,
отсортируем в порядке уменьшения
вероятностей.

Символ	Вероятность
C	4/14
A	3/14
E	3/14
B	1/14
D	1/14
F	1/14
G	1/14



Дэвид
Хаффман
(1925–1999)

Построим кодовое дерево по методу Хаффмана с оптимальными префиксными кодами.

1. Выберем 2 элемента с минимальной вероятностью.
2. Формируем новый узел с вероятностью, равной сумме предыдущих 2 элементов. Полученная сумма становится новым элементом таблицы, занимающим соответствующее место в списке убывающих по величине вероятностей.
3. Эта процедура продолжается до тех пор, пока в таблице не останутся всего два элемента.

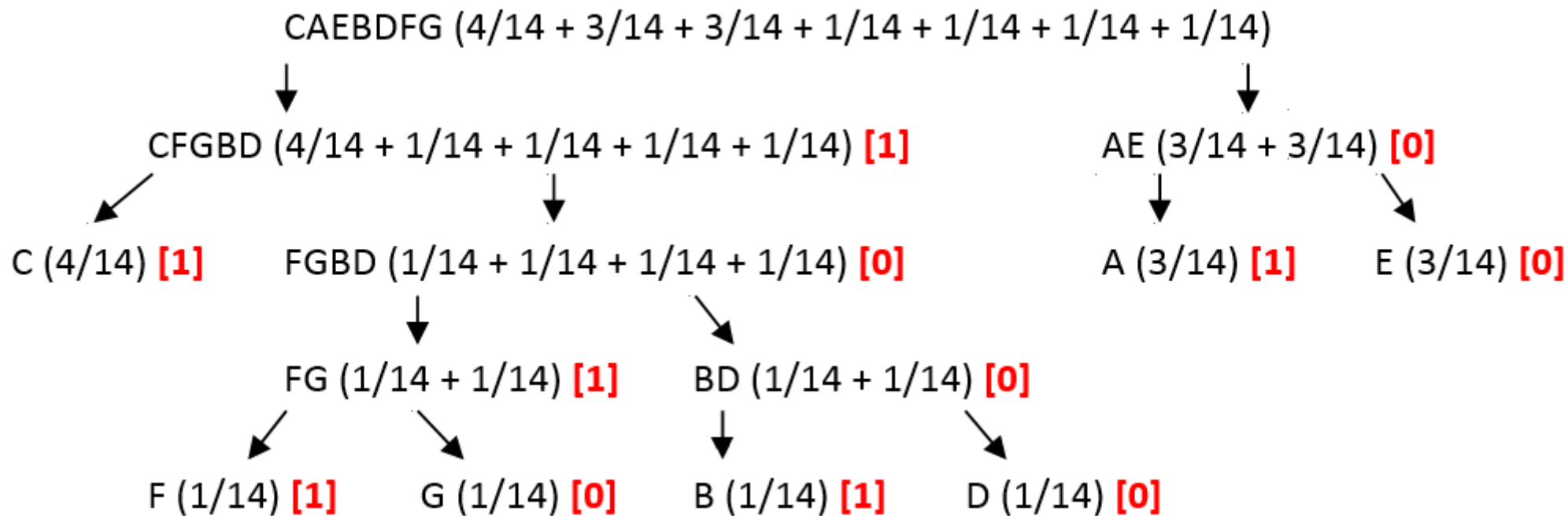
[illegible]



Двум наименьшим символам или узлам на каждом уровне присвоим значение 0 (меньшему) или 1 (большему).

[illegible]

Построим кодовое дерево (от корня).





Составим таблицу кодировки.

СИМВОЛ	1	2	3	4	Код
C	1	11			11
A	0	01			01
E		00			00
B	1	10	100	1001	1001
D				1000	1000
F			101	1011	1011
G				1010	1010

Дана последовательность символов: AAABCCCCDEEEFG

01.01.01.1011.11.11.11.11.1000.00.00.00.1011.1010 —
36 бит

Коэффициент сжатия = $(14 * 16 \text{ бит}) / 36 \text{ бит} \approx 6,222$.

Средняя длина кодового слова = $(C) 4/14 * 2 + (A) 3/14 * 2 + (E) 3/14 * 2 + (B) 1/14 * 4 + (D) 1/14 * 4 + (F) 1/14 * 4 + (G) 1/14 * 4 = 8/14 + 6/14 + 6/14 + 4/14 + 4/14 + 4/14 + 4/14 = 36 / 14 \approx 2,571 \text{ бит/символ}$



Причины:

- Альфа-частицы от примесей в чипе микросхемы.
- Нейтроны из фонового космического излучения.

Частота единичных битовых ошибок (на 1 GB):

- От 1 раза в час до 1 раза в тысячелетие (по данным исследования Google получилось 1 раз в сутки).

Способы обработки данных:

- Использовать полученные данные без проверки на ошибки.
- Обнаружить ошибку, выполнить запрос повторной передачи поврежденного блока.
- Обнаружить ошибку и отбросить поврежденный блок.
- Обнаружить и исправить ошибку.
- Тройная модульная избыточность.



Помехоустойчивые коды — коды, позволяющие обнаружить и (или) исправить ошибки в кодовых словах, которые возникают при передаче по каналам связи.

1) Блочные — фиксированные блоки длиной i символов преобразуются в блоки длиной n символов:

- Неравномерные — редко используемые символы кодируются большим количеством символов (имеют большую длину).
- Равномерные — длина блока (символа) постоянна:
 - а) Неразделимые — коды с постоянной плотностью единиц.
 - б) Разделимые — можно отделить (выделить) служебные биты r от информационных битов i .

2) Непрерывные (свёрточные) — передаваемая информационная последовательность не разделяется на блоки.

Коэффициент избыточности — отношение числа проверочных разрядов (r) к общему числу разрядов (n).

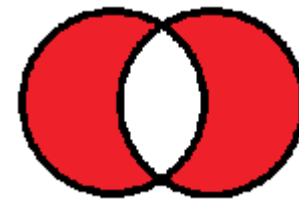
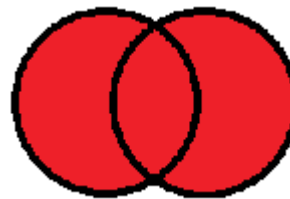
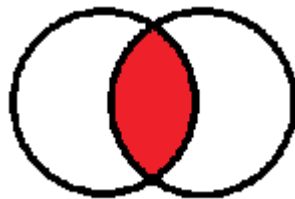


Контрольная сумма — некоторое число, рассчитанное путем применения определенного алгоритма к набору данных и используемое для проверки целостности этого набора данных при их передаче или хранении.

Бит чётности — частный случай контрольной суммы, представляющий из себя 1 контрольный бит, используемый для проверки четности количества единичных битов в двоичном числе.

Сумма по модулю 2 — исключающее «ИЛИ» (для двух операндов), логическое сложение или битовое сложение, разность двух/трёх множеств.

$$A \bmod 2 B = A \oplus B = (\neg(A \wedge B)) \wedge (A \vee B) = \neg((A \wedge B) \vee (\neg A \vee \neg B))$$





Пример. Есть 1 информационный бит $i = 1$.
К нему идёт один бит чётности r_1 .
 $i = r_1, i \oplus r_1 = 0$.

i исх	r_1 исх	i рез	r_1 рез	$i \text{ рез} \oplus r_1 \text{ рез}$
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

A	B	C	$A \oplus B \oplus C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



Код Хэмминга — блочный равномерный делимый самокорректирующийся код. Исправляет одиночные битовые ошибки, возникшие при передаче или хранении данных.

Синдром последовательности S — набор контрольных сумм информационных и проверочных разрядов.

Пример. Есть 1 информационный бит $i = 1$.

К нему идут два бита чётности r_1 и r_2 .

$$i = r_1 = r_2, s_1 = i \oplus r_1, s_2 = i \oplus r_2.$$



Ричард
Уэсли
Хэмминг
(1915–1998)

i исх	r_1 исх	r_2 исх	i рез	r_1 рез	r_2 рез	s_1	s_2
1	1	1	0	0	0	0	0
1	1	1	0	0	1	0	1
1	1	1	0	1	0	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	1	1
1	1	1	1	0	1	1	0
1	1	1	1	1	0	0	1
1	1	1	1	1	1	0	0



r_1	r_2	r_3	i_1	i_2	i_3	i_4
-------	-------	-------	-------	-------	-------	-------

$$r_1 = i_1 \oplus i_2 \oplus i_4$$

$$r_2 = i_1 \oplus i_3 \oplus i_4$$

$$r_3 = i_2 \oplus i_3 \oplus i_4$$

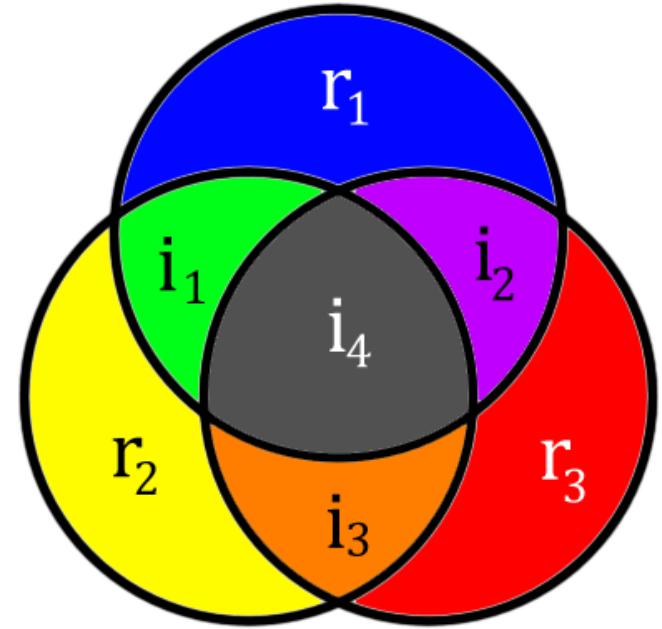
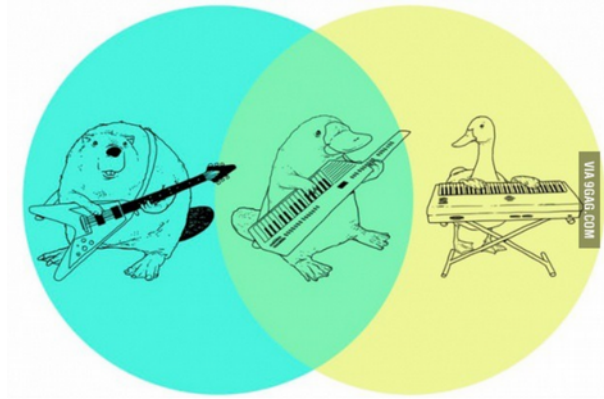


Таблица кода Хэмминга



	1	2	3	4	5	6	7	
2^x	r_1	r_2	i_1	r_3	i_2	i_3	i_4	S
1	X		X		X		X	s_1
2		X	X			X	X	s_2
4				X	X	X	X	s_3

$$r_1 = i_1 \oplus i_2 \oplus i_4$$

$$r_2 = i_1 \oplus i_3 \oplus i_4$$

$$r_3 = i_2 \oplus i_3 \oplus i_4$$

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4$$

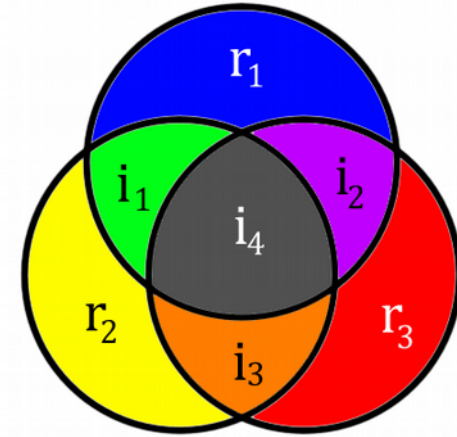
$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4$$

Синдром S (s1, s2, s3)	000	001	010	011	100	101	110	111
Конфигурация ошибок (позиция в сообщении)	НЕТ	0001000	0100000	0000010	1000000	0000100	0010000	0000001
Ошибка в символе	НЕТ	r_3	r_2	i_3	r_1	i_2	i_1	i_4



Таблица кода Хэмминга (2)

	1	2	3	4	5	6	7	
Пример полученного сообщения	1	1	1	0	0	0	1	
2^x	r_1	r_2	i_1	r_3	i_2	i_3	i_4	S
1	X		X		X		X	s_1
2		X	X			X	X	s_2
4				X	X	X	X	s_3




$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

Ошибка в бите i_4 .

Таблица кода Хэмминга (3)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
2^x	r_1	r_2	i_1	r_3	i_2	i_3	i_4	r_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	S
1	X		X		X		X		X		X		X		X	s_1
2		X	X			X	X			X	X			X	X	s_2
4				X	X	X	X					X	X	X	X	s_3
8								X	X	X	X	X	X	X	X	s_4

По таблице видно, за какие информационные биты отвечает каждый проверочный бит: контрольный бит с номером N контролирует все последующие N бит через каждые N бит, начиная с позиции N .

Аналогично с ошибочным битом.

Пример. Имеем синдром $S(0,0,1,1)$. Проверяем, за какой бит отвечают только r_3 и r_4 .

Ответ: i_8 (12-й символ сообщения).



Определение минимального числа контрольных разрядов: $2^r \geq r + i + 1$.

Классические коды Хэмминга с маркировкой $(n; i)$:
(7,4); (15,11); (31,26)...

Диапазон информационных разрядов, i	Минимальное число контрольных разрядов, r
1	2
2-4	3
5-11	4
12-26	5
27-57	6

Коэффициент избыточности — отношение числа проверочных разрядов (r) к общему числу разрядов ($n = i + r$).