Код Хэмминга

Коды Хэмминга — вероятно, наиболее известный из первых самоконтролирующихся и самокорректирующихся кодов. Построены они применительно к двоичной системе счисления.

История

В середине 1940-х годов Ричард Хэмминг работал в знаменитых лабораториях фирмы Белл (Bell Labs) на счётной машине Bell Model V. Это была электромеханическая машина, использующая релейные блоки, скорость которых была очень низка: один оборот за несколько секунд. Данные вводились в машину с помощью перфокарт, поэтому в процессе чтения часто происходили ошибки. В рабочие дни использовались специальные коды, чтобы обнаруживать и исправлять найденные ошибки, при этом оператор узнавал об ошибке по свечению лампочек, исправлял и снова запускал машину. В выходные дни, когда не было операторов, при возникновении ошибки машина автоматически выходила из программы и запускала другую.

Хэмминг часто работал в выходные дни, и все больше и больше раздражался, потому что часто был должен перезагружать свою программу из-за ненадежности перфокарт. На протяжении нескольких лет он проводил много времени над построением эффективных алгоритмов исправления ошибок. В 1950 году он опубликовал способ, который известен как код Хэмминга.

Систематические коды

Систематические коды образуют большую группу из блочных, разделимых кодов (в которых все символы можно разделить на проверочные и информационные). Особенностью систематических кодов является то, что проверочные символы образуются в результате линейных операций над информационными символами. Кроме того, любая разрешенная кодовая комбинация может быть получена в результате линейных операций над набором линейно независимых кодовых комбинаций.

Самоконтролирующиеся коды

Коды Хэмминга являются самоконтролирующимися кодами, то есть кодами, позволяющими автоматически обнаруживать ошибки при передаче данных. Для их построения достаточно приписать к каждому слову один добавочный (контрольный) двоичный разряд и выбрать цифру этого разряда так, чтобы общее количество единиц в изображении любого числа было, например, четным. Одиночная ошибка в каком-либо разряде передаваемого слова (в том числе, может быть, и в контрольном разряде) изменит четность общего количества единиц. Счетчики по модулю 2, подсчитывающие количество единиц, которые содержатся среди двоичных цифр числа, могут давать сигнал о наличии ошибок.

При этом невозможно узнать, в каком именно разряде произошла ошибка, и, следовательно, нет возможности исправить её. Остаются незамеченными также ошибки, возникающие одновременно в двух, четырёх, и т.д. — в четном количестве разрядов. Впрочем, двойные, а тем более четырёхкратные ошибки полагаются маловероятными.

Самокорректирующиеся коды

Коды, в которых возможно автоматическое исправление ошибок, называются самокорректирующимися. Для построения самокорректирующегося кода, рассчитанного на исправление одиночных ошибок, одного контрольного разряда недостаточно. Как видно из дальнейшего, количество контрольных разрядов k должно быть выбрано так, чтобы удовлетворялось неравенство $2^k \geq k + m + 1$ или $k \geq \log_2(k + m + 1)$, где m — количество основных двоичных разрядов кодового слова. Минимальные значения k при заданных

значениях m, найденные в соответствии с этим неравенством, приведены в таблице.

Диапазон m	**k_{min}**
1	2
2-4	3
5-11	4
12-26	5
27-57	6

В настоящее время наибольший интерес представляют двоичные блочные корректирующие коды. При использовании таких кодов информация передаётся в виде блоков одинаковой длины и каждый блок кодируется и декодируется независимо друг от друга. Почти во всех блочных кодах символы можно разделить на информационные и проверочные. Таким образом, все комбинации кодов разделяются на разрешенные (для которых соотношение информационных и проверочных символов возможно) и запрещенные.

Основными характеристиками самокорректирующихся кодов являются:

1. Число разрешенных и запрещенных комбинаций. Если n - число символов в блоке, r - число проверочных символов в блоке, k - число информационных символов, то 2^n - число возможных кодовых комбинаций, 2^k - число разрешенных кодовых комбинаций, $2^n - 2^k$ - число запрещенных комбинаций.
2. Избыточность кода. Величину $\frac{r}{n}$ называют избыточностью корректирующего кода.
3. Минимальное кодовое расстояние. Минимальным кодовым расстоянием d называется минимальное число искаженных символов, необходимое для перехода одной разрешенной комбинации в другую.
4. Число обнаруживаемых и исправляемых ошибок. Если g - количество ошибок, которое код способен исправить, то необходимо и достаточно, чтобы $d \geq 2g + 1$
5. Корректирующие возможности кодов.

Граница Плоткина даёт верхнюю границу кодового расстояния $d \leqslant \frac{n*2^{k-1}}{2^k-1}$ или $r \geq 2 * (d-1) - \log_2 d$ при $n \geq 2 * d - 1$

Граница Хемминга устанавливает максимально возможное число разрешенных кодовых комбинаций $2^k \leq 2^n / \sum_{i=0}^{\frac{d-1}{2}} C_n^i$ где C_n^i - число сочетаний из n элементов по i элементам. Отсюда можно получить выражение для оценки числа проверочных символов: $r \geq log_2(\sum_{i=0}^{\frac{d-1}{2}} C_n^i)$ Для значений $(d/n) \leq 0.3$ разница между границей Хемминга и границей Плоткина невелика.

Граница Варшамова-Гильберта для больших n определяет нижнюю границу числа проверочных символов $r \geq log_2(\sum_{i=0}^{d-2} C_{n-1}^i)$ Все вышеперечисленные оценки дают представление о **верхней границе** d при фиксированных n и k или **оценку снизу** числа проверочных символов

Код Хемминга

Построение кодов Хемминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным. $r_1 = i_1 \oplus i_2 \oplus ... \oplus i_k$. знак $\oplus$ здесь означает сложение по модулю 2 $S = i_1 \oplus i_2 \oplus ... \oplus i_n \oplus r_1$. $S = 0$- ошибки нет, $s = 1$ однократная ошибка. Такой код называется $(k+1, k)$ или $(n, n-1)$. Первое число - количество элементов последовательности, второе - количество информационных символов. Для каждого числа проверочных символов $r = 3, 4, 5..$существует классический код Хемминга с маркировкой $(n, k) = (2^r - 1, 2^r - 1 - r)$ т.е. - $(7, 4), (15, 11), (31, 26)$. При иных значениях k получается так называемый усеченных код, например международный телеграфный код МТК-2, у которого $k = 5$. Для него необходим код Хемминга $(9, 5)$, который является усеченным от классического $(15, 11)$. Для Примера рассмотрим классический код Хемминга $(7, 4)$. Сгруппируем проверочные символы следующим образом:

$r_1 = i_1 \oplus i_2 \oplus i_3$

$r_2 = i_2 \oplus i_3 \oplus i_4$

$r_3 = i_1 \oplus i_2 \oplus i_4$

знак $\oplus$ здесь означает сложение по модулю 2.

Получение кодового слова выглядит следующим образом:

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} i_1 & i_2 & i_3 & i_4 & r_1 & r_2 & r_3 \end{pmatrix}$$

На вход декодера поступает кодовое слово $V = (i'_1, i'_2, i'_3, i'_4, r'_1, r'_2, r'_3)$ где штрихом помечены символы, которые могут исказиться в результате помехи. В декодере в режиме исправления ошибок строится последовательность синдромов:

$S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_3$

$S_2 = r_2 \oplus i_2 \oplus i_3 \oplus i_4$

$S_3 = r_3 \oplus i_1 \oplus i_2 \oplus i_4$

$S = (s_1, s_2, s_3)$ называется синдромом последовательности.

Получение синдрома выглядит следующим образом:

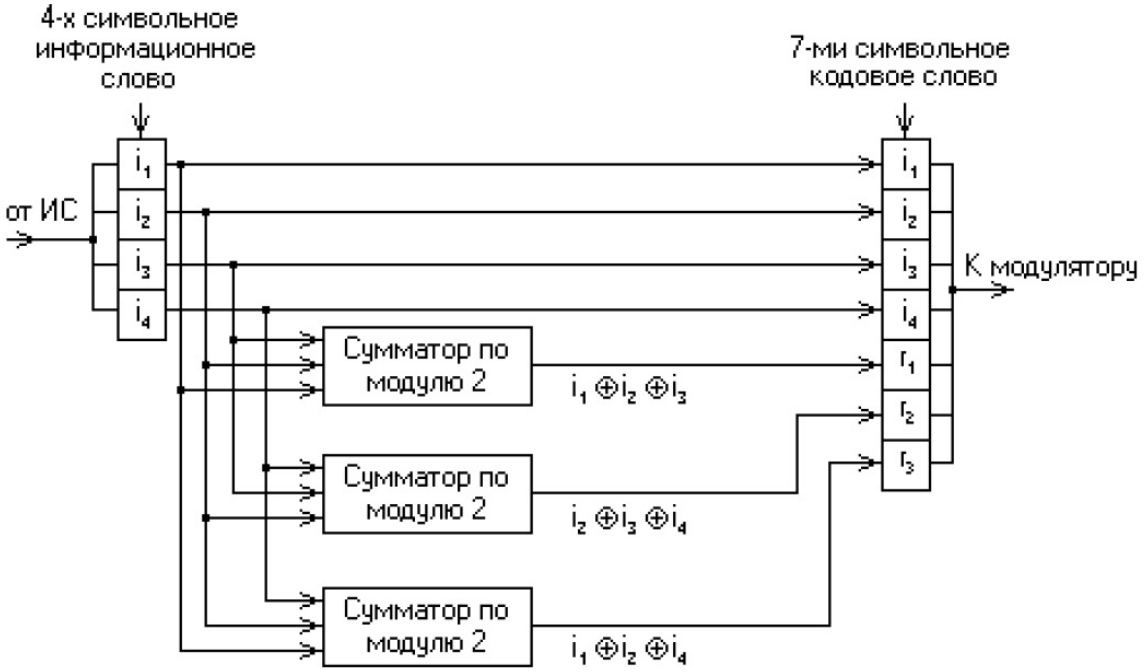
$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & r_1 & r_2 & r_3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_1 & S_2 & S_3 \end{pmatrix}$$

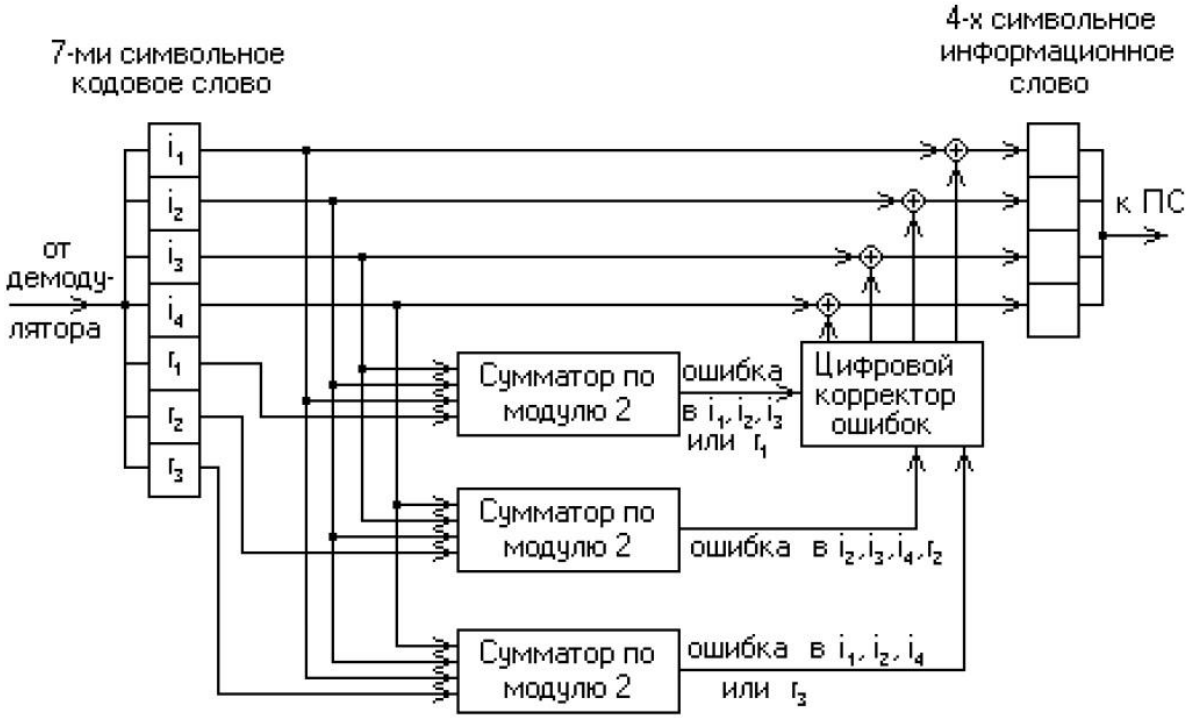
Кодовые слова $(7, 4)$ кода Хемминга

i_1	i_2	i_3	i_4	r_1	r_2	r_3
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	0
1	1	1	1	1	1	1

Синдром $(0, 0, 0)$ указывает на то, что в последовательности нет искажений. Каждому ненулевому синдрому соответствует определенная конфигурация ошибок, которая исправляется на этапе декодирования. Для кода $(7, 4)$ в таблице указаны ненулевые синдромы и соответствующие им конфигурации ошибок (для вида: i_1 i_2 i_3 i_4 r_3 r_2 r_1).

Синдром	001	010	011	100	101	110	111
Конфигурация ошибок	0000001	0000010	0001000	0000100	1000000	0010000	0100000
Ошибка в символе	r_3	r_2	i_4	r_1	i_1	i_3	i_2


4-х символьное информационное слово
7-ми символьное кодовое слово
от ИС
i_1
i_2
i_3
i_4
Сумматор по модулю 2
$i_1 \oplus i_2 \oplus i_3$
Сумматор по модулю 2
$i_2 \oplus i_3 \oplus i_4$
Сумматор по модулю 2
$i_1 \oplus i_2 \oplus i_4$
r_1
r_2
r_3
К модулятору


7-ми символьное кодовое слово
4-х символьное информационное слово
от демоду-лятора
i_1
i_2
i_3
i_4
r_1
r_2
r_3
Сумматор по модулю 2
ошибка в i_1, i_2, i_3 или r_1
Сумматор по модулю 2
ошибка в i_2, i_3, i_4, r_2
Сумматор по модулю 2
ошибка в i_1, i_2, i_4 или r_3
Цифровой корректор ошибок
к ПС

Алгоритм кодирования

Предположим, что нужно сгенерировать код Хемминга для некоторого информационного кодового слова. В качестве примера возьмём 15-битовое кодовое слово $\mathbf{x}_1...\mathbf{x}_{15}$, хотя алгоритм пригоден для кодовых слов любой длины. В приведённой ниже таблице в первой строке даны номера позиций в кодовом слове, во второй — условное обозначение битов, в третьей — значения битов.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$
1	0	0	1	0	0	1	0	1	1	1	0	0	0	1

Вставим в информационное слово контрольные биты $\mathbf{r}_0...\mathbf{r}_4$ таким образом, чтобы номера их позиций представляли собой целые степени двойки: 1, 2, 4, 8, 16... Получим 20-разрядное слово с 15 информационными и 5 контрольными битами. Первоначально контрольные биты устанавливаем равными нулю. На рисунке контрольные биты выделены розовым цветом.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
$\mathbf{r}_0$	$\mathbf{r}_1$	$\mathbf{x}_1$	$\mathbf{r}_2$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{r}_3$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{r}_4$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$
0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	1

В общем случае количество контрольных бит в кодовом слове равно двоичному логарифму числа бит кодового слова (включая контрольные биты), округлённому в большую сторону до ближайшего целого. Например, информационное слово длиной 1 или 2 бита требует двух контрольных разрядов, 3- или 4-битовое информационное слово — трёх, 5...11-битовое — четырёх, 12...26-битовое — пяти и т.д.

Добавим к таблице 5 строк (по количеству контрольных битов), в которые поместим матрицу преобразования. Каждая строка будет соответствовать одному контрольному биту (нулевой контрольный бит — верхняя строка, четвёртый — нижняя), каждый столбец — одному биту кодируемого слова. В каждом столбце матрицы преобразования поместим двоичный номер этого столбца, причём порядок следования битов будет обратный — младший бит расположим в верхней строке, старший — в нижней. Например, в третьем столбце матрицы будут стоять числа 11000, что соответствует двоичной записи числа три: 00011.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	
$\mathbf{r}_0$	$\mathbf{r}_1$	$\mathbf{x}_1$	$\mathbf{r}_2$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{r}_3$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{r}_4$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	
0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	1	
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	$\mathbf{r}_0$
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	$\mathbf{r}_1$
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	$\mathbf{r}_2$
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	$\mathbf{r}_3$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	$\mathbf{r}_4$

В правой части таблицы мы оставили пустым один столбец, в который поместим результаты вычислений контрольных битов. Вычисление контрольных битов производим следующим образом. Берём одну из строк матрицы преобразования (например, $\mathbf{r}_0$) и находим её скалярное произведение с кодовым словом, то есть перемножаем соответствующие биты обеих строк и находим сумму произведений. Если произведение получилось больше единицы, находим остаток от его деления на 2. Иными словами, мы подсчитываем сколько раз в кодовом слове и соответствующей строке матрицы в одинаковых позициях стоят единицы и берём это число по модулю 2.

Если описывать этот процесс в терминах матричной алгебры, то операция представляет собой перемножение матрицы преобразования на матрицу-столбец кодового слова, в результате чего получается матрица-столбец контрольных разрядов, которые нужно взять по модулю 2.

Например, для строки r_0:

$r_0 = (1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 1) \bmod 2 = 5 \bmod 2 = 1$.

Полученные контрольные биты вставляем в кодовое слово вместо стоявших там ранее нулей. Кодирование по Хэммингу завершено. Полученное кодовое слово — 10110010001011110001.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**		
$\mathbf{r_0}$	$\mathbf{r_1}$	$\mathbf{x_1}$	$\mathbf{r_2}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{x_4}$	$\mathbf{r_3}$	$\mathbf{x_5}$	$\mathbf{x_6}$	$\mathbf{x_7}$	$\mathbf{x_8}$	$\mathbf{x_9}$	$\mathbf{x_{10}}$	$\mathbf{x_{11}}$	$\mathbf{r_4}$	$\mathbf{x_{12}}$	$\mathbf{x_{13}}$	$\mathbf{x_{14}}$	$\mathbf{x_{15}}$		
1	0	1	1	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1		
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	$\mathbf{r_0}$	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	$\mathbf{r_1}$	0
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	$\mathbf{r_2}$	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	$\mathbf{r_3}$	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	$\mathbf{r_4}$	1

Алгоритм декодирования

Алгоритм декодирования по Хэммингу абсолютно идентичен алгоритму кодирования. Матрица преобразования соответствующей размерности умножается на матрицу-столбец кодового слова и каждый элемент полученной матрицы-столбца берётся по модулю 2. Полученная матрица-столбец получила название «матрица синдромов». Легко проверить, что кодовое слово, сформированное в соответствии с алгоритмом, описанным в предыдущем разделе, всегда даёт нулевую матрицу синдромов.

Матрица синдромов становится ненулевой, если в результате ошибки (например, при передаче слова по линии связи с шумами) один из битов исходного слова изменил своё значение. Предположим для примера, что в кодовом слове, полученном в предыдущем разделе, шестой бит изменил своё значение с нуля на единицу (на рисунке обозначено красным цветом). Тогда получим следующую матрицу синдромов.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**		
$\mathbf{r_0}$	$\mathbf{r_1}$	$\mathbf{x_1}$	$\mathbf{r_2}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{x_4}$	$\mathbf{r_3}$	$\mathbf{x_5}$	$\mathbf{x_6}$	$\mathbf{x_7}$	$\mathbf{x_8}$	$\mathbf{x_9}$	$\mathbf{x_{10}}$	$\mathbf{x_{11}}$	$\mathbf{r_4}$	$\mathbf{x_{12}}$	$\mathbf{x_{13}}$	$\mathbf{x_{14}}$	$\mathbf{x_{15}}$		
1	0	1	1	0	1	1	0	0	0	1	0	1	1	1	1	0	0	0	1		
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	$\mathbf{s_0}$	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	$\mathbf{s_1}$	1
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	$\mathbf{s_2}$	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	$\mathbf{s_3}$	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	$\mathbf{s_4}$	0

Заметим, что при однократной ошибке матрица синдромов всегда представляет собой двоичную запись (младший разряд в верхней стоке) номера позиции, в которой произошла ошибка. В приведённом примере матрица синдромов (01100) соответствует двоичному числу 00110 или десятичному 6, откуда следует, что ошибка произошла в шестом бите.

Литература

- Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ. М.: Мир, 1976, 600 с.
- Пенин П.Е.,Филиппов Л.Н. Радиотехнические системы передачи информации. М.: Радио и Связь, 1984, 256 с.
- Блейхут Р. Теория и практика кодов, контролирующих ошибки. Пер. с англ. М.: Мир, 1986, 576 с.

Применение

Код Хэмминга используется в некоторых прикладных программах в области хранения данных, особенно в RAID 2; кроме того, метод Хэмминга давно применяется в памяти типа ECC и позволяет «на лету» исправлять однократные и обнаруживать двукратные ошибки.

Источники и основные авторы

Код Хэмминга *Источник*: http://ru.wikipedia.org/w/index.php?oldid=62054989 *Редакторы*: A5b, Aats1988, AdmiralHood, Alik Kirillovich, AntonR, CommonsDelinker, Cot-Pilot, Cthulhu92, Devgru, Euku, Gvozdik, Holop, Inc ru, Insolor, IntKecsk, KAVA, Kochergin v.i., Les, MaxBioHazard, Mcherenkov, Mir76, Panther, RaMaXa, Sergeisemenoff, Sergey Zhuravlev, SolitaryDreamer, Splux, TranvanKhanh, Unomano, Vald, Vanuan, Vlsergey, Vodkadrinker, Volkov, Winterheart, Xber9, ZOGmeister, Zymedo, Ботильда, Голем, ПоЧитатель, Серж Тихомиров, Чръный человек, 101 анонимных правок

Источники, лицензии и редакторы изображений

Файл:My hemcode.png *Источник*: http://ru.wikipedia.org/w/index.php?title=Файл:My_hemcode.png *Лицензия*: Creative Commons Zero *Редакторы*: я

Файл:HemDecode.PNG *Источник*: http://ru.wikipedia.org/w/index.php?title=Файл:HemDecode.PNG *Лицензия*: Creative Commons Zero *Редакторы*: Vodkadrinker

Лицензия