Hamming code example

Dr J. Vaughan

February 11, 2013

1 The Problem

Calculate a Hamming codeword that can correct 1-bit errors in the ASCII code for a line feed, LF, 0x0a.

We are going to calculate a codeword that is capable of correcting all single-bit errors in an 8-bit data element. In the codeword, there are m data bits and r redundant (check) bits, giving a total of n codeword bits.

n = m + r

Each valid codeword of n bits contains m valid data bits. For each valid m-bit data entity, there are n bits that can be changed to give an invalid codeword. Thus the total number of codewords corresponding to a valid data entity is n+1 (n invalid plus 1 valid codeword). As there are 2^m valid data patterns, the total number of codewords is $(n+1)2^m$. In an n-bit codeword, the possible number of patterns is 2^n, and this limits the number of valid + invalid codes that can exist. Thus

$(n+1)2^m \leq 2^n$

and, since n = m + r,

$(m+r+1)2^m \leq 2^{m+r}$

so

$(m+r+1) \leq 2^r$

The least number of redundant bits that satisfies this inequality when m = 8 is r = 4 bits.

Thus, we have a 12-bit codeword. Bits 1, 2, 4 and 8 will be the check bits.

The codeword will be of the form rr0r 000r 1010 and we shall use even parity.

Codeword	r	r	0	r	0	0	0	r	1	0	1	0
Bit No.	1	2	3	4	5	6	7	8	9	10	11	12

2 Approach to a solution

1. Decide on the number of bits in the codeword
2. Determine the bit positions of the check bits
3. Determine which parity bits check which positions
4. Calculate the values of the parity bits

The bit positions covered by each parity bit can be calculated by writing each bit position as a sum of the powers of 2:
$1 = 1$
$2 = 2$
$3 = 1 + 2$
$4 = 4$
$5 = 1 + 4$
$6 = 2 + 4$
$7 = 1 + 2 + 4$
$8 = 8$
$9 = 1 + 8$
$10 = 2 + 8$
$11 = 3 + 8$
$12 = 4 + 8$

Thus,
Check bit 1 governs positions 1, 3, 5, 7, 9: Value = 1 (0 0 0 1)
Check bit 2 governs positions 2, 3, 6, 7, 10: Value = 0 (0 0 0 0)
Check bit 4 governs positions 4, 5, 6, 7, 12 = 0 (0 0 0 0)
Check bit 8 governs positions 8, 9, 10, 11, 12 = 0 (1 0 1 0)

Thus, the complete codeword is 1000 0000 1010.

A 1-bit error in this codeword can be corrected as follows:

First, calculate the parity bits. If all are correct, there is either no error, or errors in more than 1 bit. However, we are limiting ourselves to single-bit errors here. If the parity bits show an error, add up all the erroneous parity bits, counting 1 for bit 1, 2 for bit 2, 4 for bit 4 and 8 for bit 8. The result gives the position of the erroneous bit, which can be complemented to give the correction.

For example, assume the codeword is corrupted to give 1010 0000 1010.
Check parity:

Bit 1: $B1 \oplus B3 \oplus B5 \oplus B7 \oplus B9 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1$
Bit 2: $B2 \oplus B3 \oplus B6 \oplus B7 \oplus B10 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 1$
Bit 4: $B4 \oplus B5 \oplus B6 \oplus B7 \oplus B12 = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$
Bit 8: $B8 \oplus B9 \oplus B10 \oplus B11 \oplus B12 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$

Thus the error is in bit position $1 + 2 = 3$, and bit 3 can be inverted to give the correct codeword:
1000 0000 1010