Hamming Codes

By

Steve Edlefsen

Contents

1 INTRODUCTION

The algorithms for generating Hamming codes for canonical and non-canonical data widths are defined in this document. Also, a new descriptor for canonical and non-canonical Hamming codes is defined: (t,k,j), where

t = the total number of bits
k = the number of canonical data bits
j = the actual number of data bits

Obviously, m = t-k, where m = the number of parity bits.

Example:

(15, 11, 8)

Conical Hamming error correction and detection code implementation is summarized in Table 1, which shows the relationship between the number of parity bits (m), the number of data bits (k) and the dimensions of the various matrices used to encode and decode Hamming codewords, which are the concatenation of the data bits and the parity bits.

Non-canonical data widths are those which are not equal to the data widths listed in Table 1.

The table shows that as the number of parity bits increases, the size of the matrices increases. This is an exponential increase.

parity	total bits	data bits	descriptor	A	G	H	S
m	$t=2^m-1$	$k=2^m-m-1$		m x k	$(I_k \mid A^T)$	$(A \mid I_{t-k})$	
2	3	1	3,1	2x1	1x1 | 1x2 = 1x3	2x1 | 1x1 = 2x1	1x2
3	7	4	7,4	3x4	4x4 | 4x3 = 4x7	3x4 | 3x3 = 3x7	1x3
4	15	11	15,11	4x11	11x11 | 11x4 = 11x15	4x11 | 4x4 = 4x15	1x4
5	31	26	31,26	5x26	26x26 | 26x5 = 26x31	5x26 | 5x5 = 5x31	1x5
6	63	57	63,57	6x57	57x57 | 57x6 = 57x63	6x57 | 6x6 = 6x63	1x6
7	127	120	127,120	7x120	120x120 | 120x7 = 120x127	7x120 | 7x7 = 7x127	1x7

TABLE 1 HAMMING CODE CONFIGURATION

2 NOTATION

Matrices and vectors are identified with a capital letter, e.g. A.

Individual elements of a matrix are denoted by a lowercase character of the identifier of the matrix. For example, the element of the A matrix in row 1 column 1 is a11, or a_{11}.

Note: The term matrix and vector are used interchangeably.

A 1 dimensional matrix is a single row matrix or vector. S is a 1 row by 8 column or 1x8 matrix:

$$S = 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$$

bit vector – a 1xn matrix whose elements are the binary numbers 0 and 1, i.e. bits.

The transpose of a matrix is a matrix with the indices of the matrix reversed. S^T is an 8 row by 1 column, or 8x1 matrix:

$$S^T = \begin{matrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

Multidimensional matrices are manipulated in an analogous way:

$$A = \begin{matrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{matrix}$$

$$A^T = \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{matrix}$$

bit value - the unsigned decimal value of a bit vector.

The character “x” means matrix multiplication.

The character “|”, vertical bar means “concatenation” when using matrix variables in an expression.

The superscript “T” means transposed.

A matrix expression is a combination of matrix identifiers with the matrix operations, e.g.

$D|Z \times G = D|Z \times I|A^T$

means matrix D concatenated with matrix Z times matrix G = matrix D concatenated with matrix Z times matrix I concatenated with the transpose of matrix A.

The character “|” is also used as a delimiter, to highlight a part of a matrix:

$d1d2d3d4d5d6d7d8|z1z2z3|$

The following example demonstrates the notation of matrix operations on rows and columns used throughout:

$$
s1 = \begin{array}{cccccccc|ccc|cccc}
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

s1 = 0*0+0*0+1*0+ 1*0+1*1+1*1+1*1+1*1+0*1+0*1+0*1+0*1+0*0+1*0+0*0
= 0 + 0 + 0 + 0 + 1 + 1 + 1 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0
= 0

Where
* = AND
+ = XOR

3 MATRIX DEFINITIONS

A = parity matrix. Creates the parity bits of the data

I = identity matrix. A square matrix of dimension mxm, where m = the number of parity bits, whose diagonal bits are 1 and the rest of the bits are 0.

G = generator matrix. A combination of the A matrix and an the identity matrix. Generates the codewords.

H = parity check matrix. A combination of the A matrix and an identity matrix. Creates the syndromes which indicate which bit is incorrect.

S = syndrome. In the case of a bit error, the bits in the syndrome match the nth column of the H matrix which indicates the nth data bit is in error.

D = the data matrix or vector

Z = the zero matrix or vector

P = the parity bits

C = the codeword matrix or vector which consists of the date vector concatenated with the parity bits.

4 CANONICAL HAMMING CODES

4.1 Cononical "A" and "I" Matrix Configuration

The canonical A (parity) and I (identity) matrices' dimensions are based on the number of parity bits and the bit width of the data.

The I matrix is a square matrix whose diagonal bits are 1 and the rest of the bits are 0.

The A matrix is formatted with mxk entries, i.e. m rows and k columns, each column a mx1 bit vector consisting of bit values from 3 to 2^m -1, excluding bit vectors which appear in the I matrix, i.e. those bit vectors whose bit value is a power of 2.

Table 2 defines the format of the A matrix.

m	bit values	mx1	dim
2	3	2x1	2x1
3	3, 5-7	3x1	3x4
4	3, 5-7, 9-15	4x1	4x11
5	3, 5-7, 9-15, 17-31	5x1	5x26
6	3, 5-7, 9-15,17-31, 33-63	6x1	6x57
7	3, 5-7, 9-15,17-31, 33-63, 65-127	7x1	7x120

TABLE 2 PARITY MATRIX FORMAT

For example, the table of entries for m = 4 would have 11 column vectors, corresponding to 11 columns of 4 bits, corresponding to 4 parity bits

bit value	3	5	6	7	9	10	11	12	13	14	15
	0	0	0	1	1	1	1	1	1	1	1
	0	1	1	1	0	0	0	1	1	1	1
	1	0	1	1	0	1	1	0	0	1	1
	1	1	0	1	1	0	1	0	1	0	1

So, the A matrix for m = 4 would be

$$A = \begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{matrix}$$

4.2 Canonical Codeword Generation

A canonical codeword is generated by the following matrix operations:

$C = D\ G = D\ (I_k \mid A^T) = D\ I_k \mid D\ A^T = D \mid D\ A^T = D \mid P$

D is the data and P is the parity bits.

Example

$$A = \begin{matrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{matrix}$$

$$A^T = \begin{matrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$D = 1 \quad 0 \quad 1 \quad 0$$

$$DA^T = 1010 \quad \begin{matrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} = 101$$

The codeword C = D | P = 1010 | 101

The matrix multiplication is expressed by the following formula:

$$P = \sum_{c=1}^{m} \sum_{r=1}^{k} a_{cr} d_c$$

4.3 Canonical Codeword Decoding

Codeword decoding implemented by the following matrix operations:

$S = H\ C^T$

Where S is the syndrome matrix.

The syndrome shall be decoded to correct the incorrect bit.

Example:

$$\begin{matrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{matrix} \quad \begin{matrix} 1 \\ <1> \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{matrix} \quad = S$$

<1> is the bit in error.

1110101 1110101 1110101
1101100 1011010 0111001
1100100 1010000 0110001
1 0 1 = syndrome => not correct => 2nd col of H => 2nd data bit incorrect

The matrix multiplication is expressed by the following formula:

$$S = \sum_{r=1}^{m} \sum_{c=1}^{k} a_{rc} c_c$$

5 NON-CANONICAL HAMMING CODES

Non-canonical Hamming codes are used for data with bit widths which are not canonical Hamming code bit widths. Table 1 lists the parameters and matrices used in canonical Hamming codes, including the number of data bits.

5.1 Non-cononical "A" and "I" Matrix Configuration

The configuration for the A and I matrices for non-canonical Hamming codes are identical with the canonical case (see Table 2), where he size of the matrices are related to the number of parity bits in the codeword, which implies the upper limit of the number of data bits.

Note that for j=number of actual data bits, $2^{m-1}-1-(m-1)<j \leq 2^{m}-1-m$ where m = the number of parity bits.

For example, the (15,11) Hamming code can generate codewords for data widths of 5 to 11, since the canonical Hamming code with the next lowest number of data bits is (7,4), i.e. 7 total bits and 4 data bits, i.e (15,11,5) to (15,11,11). $2^{m-1}-m+1<j \leq 2^{m}-1-m = 5<j\leq 11$, where m = 4.

Table 3 summarizes the ranges of the non-canonical bit widths.

m	2^{m-1}-m+1	2^{m}-1-m
2	1	1
3	1	4
4	5	11
5	12	26
6	27	57
7	58	120

TABLE 3 RANGES OF NON-CONICAL DATA WIDTHS

5.2 Non-canonical Codeword Generation

For the (15,11,8) Hamming code

t = 15 total bits
k = 11 canonical data bits
j = 8 actual data bits
m = 4 canonical parity bits for (15,11)

An 8 bit data vector, which is not the canonical 7, can be represented by

```
     11
12345678901
ddddddddzzz
```

which is 8 data bits concatenated with 3, 0 bits, identified as z1(bit 9), z2 (bit 10), z3 (bit 11).

The resulting 11 bits is the conical (15,11) Hamming code data bit width with 4 parity bits. The parity bits are generated using the standard generator matrix, where D is the data bits and Z is the zero bits and I is the identity matrix.

D|Z x G = D|Z x I|A^T

$$d1d2d3d4d5d6d7d8z1z2z3 \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

$$d1d2d3d4d5d6d7d8|z1z2z3| \left.\begin{array}{ccccccccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}\right|$$

The matrix operations can be simplified, since

D|Z x I = D|Z

D|Z x G = D|Z x I|A^T = D|Z | D|Z x A^T

$$d1d2d3d4d5d6d7d8|z1z2z3| \; | \, d1d2d3d4d5d6d7d8|z1z2z3| \begin{array}{cccc} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ - & - & - & - \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}$$

The following is the matrix operation in enumerated detail:

$$p1 = \begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 \end{matrix}$$

$$p2 = \begin{matrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 \end{matrix}$$

$$p3 = \begin{matrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 \end{matrix}$$

$$p4 = \begin{matrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 \end{matrix}$$

Since the zs are 0

$$p1 = \frac{0}{d1}\ \frac{0}{d2}\ \frac{0}{d3}\ \frac{0}{d4}\ \frac{1}{d5}\ \frac{1}{d6}\ \frac{1}{d7}\ \frac{1}{d8}\ \frac{1}{0}\ \frac{1}{0}\ \frac{1}{0}$$

$$p2 = \frac{0}{d1}\ \frac{1}{d2}\ \frac{1}{d3}\ \frac{1}{d4}\ \frac{0}{d5}\ \frac{0}{d6}\ \frac{0}{d7}\ \frac{1}{d8}\ \frac{1}{0}\ \frac{1}{0}\ \frac{1}{0}$$

$$p3 = \frac{1}{d1}\ \frac{0}{d2}\ \frac{1}{d3}\ \frac{1}{d4}\ \frac{0}{d5}\ \frac{1}{d6}\ \frac{1}{d7}\ \frac{0}{d8}\ \frac{0}{0}\ \frac{1}{0}\ \frac{1}{0}$$

$$p4 = \frac{1}{d1}\ \frac{1}{d2}\ \frac{0}{d3}\ \frac{1}{d4}\ \frac{1}{d5}\ \frac{0}{d6}\ \frac{1}{d7}\ \frac{0}{d8}\ \frac{1}{0}\ \frac{0}{0}\ \frac{1}{0}$$

Since the last 3 values are always 0, the parity of the bits remain the same if these bits are removed and the final value is

d1 d2 d3 d4 d5 d6 d7 d8 p1 p2 p3 p4

The following equations prove what is enumerated above:

$$\sum_{p=1}^{m}\sum_{q=1}^{k+m} d(q) * h^T\,(q,p) = \sum_{p=1}^{m}\left[\sum_{q=1}^{j} d(q) * h^T\,(q,p) + \sum_{q=1}^{k-j} d(q) * h^T\,(q,p)\right]$$

$$= \sum_{p=1}^{m}\left[\sum_{q=1}^{j} d(q) * h^T\,(q,p) + \sum_{q=1}^{k-j} z(q) * h^T\,(q,p)\right] = \sum_{p=1}^{m}\left[\sum_{q=1}^{j} d(q) * h^T\,(q,p) + 0\right]$$

$$= \sum_{p=1}^{m}\sum_{q=1}^{j} d(q) * h^T\,(q,p)$$

The only difference between the calculation of parity with a non-conical bit width, e.g. 8 and a canonical width, e.g. 11, is that only j bits are used in the calculation. The dimensions of the A matrix, m x k, e.g. 4 x 11, since the (15,11) Hamming code is used in both cases, is the same.

5.3 Non-canonical Codeword Decoding

For the (15,11,8) Hamming code

m = 4 canonical parity bits
k = 11 canonical data bits (2^m-1-m)
j = actual data bits

As in the encoding case, 8 data bits are extended to 11 bits with 3 added 0 bits, identified as z1, z2 …, resulting in 11 bits which are the conical (15,11) Hamming code with 4 parity bits. The syndrome bits are generated using the standard parity check matrix, H, with D, the data bits, Z, the zero bits, P, the received parity bits, I, the identity matrix and S, the syndrome bits.

$$D^T|Z^T|P^T \; x \; H \; = \; D^T|Z^T|P^T \; x \; A^T|I$$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0
0	1	1	1	0	0	0	1	1	1	1	0	1	0	0
1	0	1	1	0	1	1	0	0	1	1	0	0	1	0
1	1	0	1	1	0	1	0	1	0	1	0	0	0	1

$d1$
$d2$
$d3$
$d4$
$d5$
$d6$
$d7$
$d8$
$z1$
$z2$
$z3$
$p1$
$p2$
$p3$
$p4$

A = J | Q
J = those columns of A which multiply D
A = those columns of A which multiply Z

$$D^T|Z^T|P^T \; x \; JQI \; = \; D^TJ + Z^TQ + P^TI \; = \; D^TJ + 0 + P^TI \; = D^TJ + P^T \; = \; S$$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0
0	1	1	1	0	0	0	1	1	1	1	0	1	0	0
1	0	1	1	0	1	1	0	0	1	1	0	0	1	0
1	1	0	1	1	0	1	0	1	0	1	0	0	0	1

$d1$
$d2$
$d3$
$d4$
$d5$
$d6$
$d7$
$d8$
$-$
$z1$
$z2$
$z3$
$-$
$p1$
$p2$
$p3$
$p4$

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{matrix} \begin{matrix} d1 \\ d2 \\ d3 \\ d4 \\ d5 \\ d6 \\ d7 \\ d8 \end{matrix} + \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix} \begin{matrix} z1 \\ z2 \\ z3 \end{matrix} + \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} p1 \\ p2 \\ p3 \\ p4 \end{matrix}$$

$$\begin{array}{cccccccccc|ccc|cccc} & & d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 & p1 & p2 & p3 & p4 \\ s1 & = & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & d5 & d6 & d7 & d8 & 0 & 0 & 0 & p1 & 0 & 0 & 0 \end{array}$$

$$\begin{array}{cccccccccc|ccc|cccc} & & d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 & p1 & p2 & p3 & p4 \\ s2 & = & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ & & 0 & d2 & d3 & d4 & 0 & 0 & 0 & d8 & 0 & 0 & 0 & 0 & p2 & 0 & 0 \end{array}$$

$$\begin{array}{cccccccccc|ccc|cccc} & & d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 & p1 & p2 & p3 & p4 \\ s3 & = & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ & & 0 & d2 & d3 & d4 & 0 & 0 & 0 & d8 & 0 & 0 & 0 & 0 & 0 & p3 & 0 \end{array}$$

$$\begin{array}{cccccccccc|ccc|cccc} & & d1 & d2 & d3 & d4 & d5 & d6 & d7 & d8 & z1 & z2 & z3 & p1 & p2 & p3 & p4 \\ s4 & = & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ & & 0 & d2 & d3 & d4 & 0 & 0 & 0 & d8 & 0 & 0 & 0 & 0 & 0 & 0 & p4 \end{array}$$

The zs are truncated since they have no effect on the result.

The following equations prove what is enumerated above:

$$\sum_{p=1}^{m} \sum_{q=1}^{k+m} d(q) * h\,(q,p) = \sum_{p=1}^{m} \left[\sum_{q=1}^{j} d(q) * h\,(q,p) + \sum_{q=j+1}^{k} d(q) * h\,(q,p) + \sum_{q=k+1}^{m+k} d(q) * h\,(q,p) \right]$$

$$= \sum_{p=1}^{m} \left[\sum_{q=1}^{j} d(q) * h\,(q,p) + \sum_{q=j+1}^{k-j} z(q-j) * h\,(q,p) + \sum_{q=k+1}^{k-j} p(q-k) * i\,(q-k,p) \right]$$

$$= \sum_{p=1}^{m} \left[\sum_{q=1}^{j} d(q) * h\,(q,p) + 0 + \sum_{q=k+1}^{k-j} p(q-k) * i\,(q-k,p) \right]$$

$$= \sum_{p=1}^{m} \left[\sum_{q=1}^{j} d(q) * h\,(q,p) + p(p) \right]$$

$$= \sum_{p=1}^{m} \sum_{q=1}^{j} d(q) * h^T\,(q,p) = S$$

As in the encoding case, the only difference between the calculation of the syndrome bits with a non-conical bit width, e.g. 8 and the canonical width 11, is that only j data bits are used in the calculation. The dimensions of the A matrix, m x k, (4 x 11), since the (15,11) Hamming code is used, is the same for both bit widths.

Example

encoding

$$00101111|000|\ |\ 00101111|000| \begin{matrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ - & - & - & - \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$

$$p1 = \begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{matrix} = 0$$

$$p2 = \begin{matrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix} = 0$$

$$p3 = \begin{matrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{matrix} = 1$$

$$p4 = \begin{matrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{matrix} = 0$$

Since the last 3 values are always 0, the parity of the bits remain the same if these bits are removed and the final value is

00101111 0010

Decoding (no error)

$$
\left[\begin{array}{cccccccc|ccc|cccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}\right]
\begin{array}{c}
0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ - \\ 0 \\ 0 \\ 0 \\ - \\ 0 \\ 0 \\ 1 \\ 0
\end{array}
$$

$$
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0
\end{array}
x
\begin{array}{c}
0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1
\end{array}
+
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1
\end{array}
x
\begin{array}{c}
0 \\ 0 \\ 0
\end{array}
+
\begin{array}{cccc}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}
x
\begin{array}{c}
0 \\ 0 \\ 1 \\ 0
\end{array}
$$

$$
s1 = \begin{array}{cccccccc|ccc|cccc}
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} = 0
$$

$$
s2 = \begin{array}{cccccccc|ccc|cccc}
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} = 0
$$

$$
s3 = \begin{array}{cccccccc|ccc|cccc}
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array} = 0
$$

$$
s4 = \begin{array}{cccccccc|ccc|cccc}
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} = 0
$$

$$
S = \begin{array}{c}
0 \\ 0 \\ 0 \\ 0
\end{array}
$$

Note: the z bits have no effect on the result

Error example
d4 = 0 -> 1

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{matrix} \; x \; \begin{matrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} + \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix} \; x \; \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} + \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \; x \; \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix}$$

$$s1 = \begin{array}{cccccccc|ccc|cccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} = 0$$

$$s2 = \begin{array}{cccccccc|ccc|cccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} = 1$$

$$s3 = \begin{array}{cccccccc|ccc|cccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} = 1$$

$$s4 = \begin{array}{cccccccc|ccc|cccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} = 1$$

$$S = \begin{matrix} 0 \\ 1 \\ 1 \\ 1 \end{matrix}$$ = column 4 of A matrix => bit 4 is flipped!

6 EXTENDED HAMMING CODES

Extended are Hamming codes with an added parity bit. This bit is the parity of all the bits in the codeword, including the parity bit itself. The parity bit, when added to the codeword makes the combination of the parity bit concatenated with the codeword, even parity.

This enables the 2 bit error detection.