Block error correcting codes

Hamming codes

Hamming codes are a family of (n, k) block error correcting codes which are binary, linear and cyclic. They have the following parameters:

Number of check bits:	$n - k = m$
Block length:	$n = 2^m - 1$
The number of information symbols:	$k = 2^m - 1 - m$
Minimum distance:	$d_{min} = 3$

$m \leq 3$

Hamming codes

• A Hamming code word is generated by multiplying the data bits, **M**, by a ***generator matrix,* G**, using modulo2 arithmetic

• This multiplication's result is called the ***code word vector***, consisting of the ***original data bits*** and the ***calculated parity bits***.

• The ***generator matrix,* G** used in constructing Hamming codes consists of **I** (***identity matrix***) and a ***parity generation matrix,* A**:

G = [**I** : **A**]

An example of a (7,4) Hamming code generator matrix:

$$G = \begin{array}{l} 1\ 0\ 0\ 0 \mid 1\ 1\ 1 \\ 0\ 1\ 0\ 0 \mid 0\ 1\ 1 \\ 0\ 0\ 1\ 0 \mid 1\ 0\ 1 \\ 0\ 0\ 0\ 1 \mid 1\ 1\ 0 \end{array}$$

Hamming codes

The multiplication of a 4-bit data vector, **M** = (m_1, m_2, m_3, m_4), by **G** results in a 7-bit ***code word vector*** of the form **X** = (m_1, m_2, m_3, m_4, c_1, c_2, c_3).

X = **MG**

$$c_1 = 1*m_1 \oplus 1*m_2 \oplus 1*m_3 \oplus 0*m_4$$

$$c_2 = 0*m_1 \oplus 1*m_2 \oplus 1*m_3 \oplus 1*m_4$$

$$c_3 = 0*m_1 \oplus 1*m_2 \oplus 0*m_3 \oplus 1*m_4$$

• It is clear that the **A** partition of **G** is responsible for the generation of the actual parity bits.

• The Hamming rule requires that n-k = 3 for a (7,4) code, therefore **A** must contain three columns to produce three parity bits.

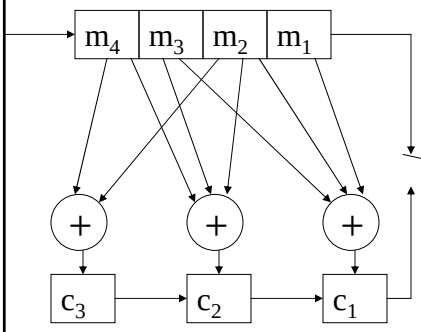
Hamming codes

The operation can be implemented as shift register:

$c_1 = 1*m_1 \oplus 1*m_2 \oplus 1*m_3 \oplus 0*m_4$

$c_2 = 0*m_1 \oplus 1*m_2 \oplus 1*m_3 \oplus 1*m_4$

$c_3 = 0*m_1 \oplus 1*m_2 \oplus 0*m_3 \oplus 1*m_4$



Information bits

Code word

Parity bits

Hamming codes

When a code is created the question is how to choose the matrix **A** in such a way that it has good error correcting properties. For a (7,4) Hamming code it is simple, put up the binary values for value 1 to 7 in a table.

$1 \rightarrow 0\ 0\ 1$
$2 \rightarrow 0\ 1\ 0$
$3 \rightarrow 0\ 1\ 1$
$4 \rightarrow 1\ 0\ 0$
$5 \rightarrow 1\ 0\ 1$
$6 \rightarrow 1\ 1\ 0$
$7 \rightarrow 1\ 1\ 1$

Take the binary value in the table that can form the identity matrix first and then put in the rest of the binary values in arbitrary order. In the parity check matrix the columns are in order 7, 6, 5, 3, 1, 2 and 4. Then the generator matrix is easily constructed.

$$\mathbf{H} = \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

$$\mathbf{G} = \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline \end{array}$$

Hamming codes

The decoding, validating the ***received code word,*** **Y**, involves multiplying it by a parity check to form **S**, the syndrome or parity check vector.

The received code word can be seen as:

$$\mathbf{Y} = \mathbf{X} + \mathbf{E}$$

Where **X** is the transmitted code word and **E** is the error pattern which is the **0**-vector if the received code word contains no error

In order to find a method for decoding the code word it is assumed that there exists a matrix H such that

$$\mathbf{X}\mathbf{H}^T = \mathbf{0}$$

for all code word **X**.

Hamming codes

The decoding matrix H must fulfill:

$$\mathbf{G}\mathbf{H}^T = \mathbf{0}$$

where H is the parity check matrix defined as:

$\mathbf{H} = [-A^T \mid I]$

$$\mathbf{H} = [-A^T \mid I] = \begin{array}{|c|c|} \hline 1\ 1\ 1\ 0 & 1\ 0\ 0 \\ \hline 0\ 1\ 1\ 1 & 0\ 1\ 0 \\ \hline 0\ 1\ 0\ 1 & 0\ 0\ 1 \\ \hline \end{array}$$

The syndrome, **S**, is defined as:

$$\mathbf{S} = \mathbf{Y}\mathbf{H}^T = \mathbf{X}\mathbf{H}^T + \mathbf{E}\mathbf{H}^T = \mathbf{E}\mathbf{H}^T$$

An important observation is that the syndrome is not dependent on the code word, **Y**, its is only dependent on the error vector, **E**.

Hamming codes

If all elements of ***syndrome***, **S**, are zero, the code word was received correctly. If **S** contains non-zero elements, the bit error can be determined by analyzing which parity check that has failed, as long as the error involves only a single bit.

It is *2n-k –1* different syndrome pattern, which means that there are no direct representation between the syndrome and the error pattern.

The strategy is that a certain syndrome pattern maps to the most probable error.

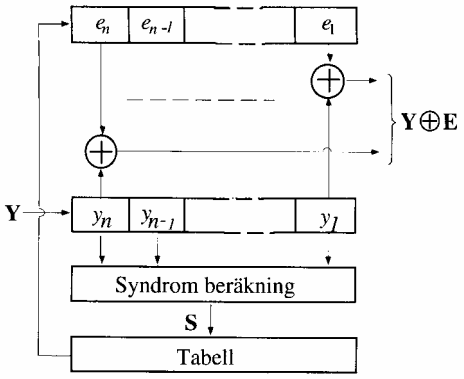
This is implemented as a table that gives which error pattern that match which syndrome, this table is called a ***standard array***. The error pattern with the ***lowest hamming weight***, the most probable match for each syndrome is called the ***co-set leader***.

Hamming codes

The received code word is read into a shift register where the syndrome calculation is performed.

The result is an address to a ROM where the most probable error patterns has been stored.

The chosen error pattern is read into a shift register where subtraction from the received word is performed



Cyclic codes
(Polynomial representation)

Cyclic codes

- Most of the error correcting block codes that are in use are in the category called cyclic codes.
- If the *n*-bit sequence **c**=(c_0, c_1…c_{n-1}) is a valid code word, then (c_{n-1}, c_0, c1….c_{n-2}), which is formed by cyclically shifting **c** one place to the right, is also a valid code word.
- This class of codes can be easily encoded and decoded using linear feedback shift registers (LFSR).
- Examples of cyclic codes are Bose Chaudhuri-Hocquenhem (BCH) and Reed-Solomon (RS) codes.

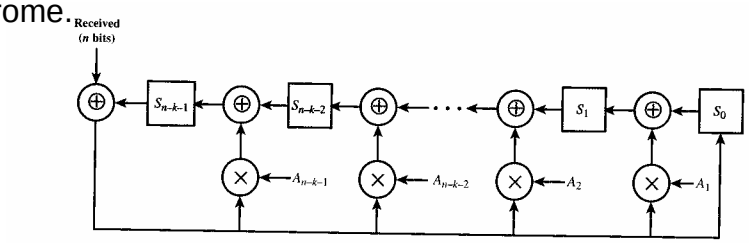
Cyclic codes

- The LFSR implementation of a cyclic error correcting encoder is the same as that of the CRC error detecting code.
- The key difference is that CRC code takes an input of arbitrary length and produces a fixed-length CRC check code, while a cyclic error correcting code takes a fixed-length input (*k* bits) and produces a fixed length check code (*n*-*k* bits).

Cyclic codes

For the encoder, the *k* bits are treated as input to produce a (*n* - *k*) code of check bits in the shift register.

For the decoder, the input is the received bit stream of *n* bits, consisting of *k* data bits followed by (*n*-*k*) parity bits. If there have been no errors after the *k* first steps, the shift register contains the pattern of check bits that were transmitted. After the remaining (*n* - *k*) steps, the shift register contains the syndrome.



Cyclic codes

For decoding a cyclic code, the following procedure is used:

- Process received bits to compute the syndrome in exactly the same fashion as the encoder process the data bits to produce the check code.
- If the syndrome bits are all zero, no error has been detected.
- If the syndrome is nonzero, perform additional processing on the syndrome for error correction.

Cyclic codes

The cyclic code can also be represented on polynomial form (recall the CRC):

$$\frac{X^{n-k}D(X)}{P(X)} = Q(X) + \frac{C(X)}{P(X)}$$

The data block is shifted to the left by *n-k* bits and divided by P(X). This produces a quotient Q(X) and a remainder C(X) of length (*n-k*) bits. The transmitted block is formed by concatenating D(X) and C(X).

$$T(X) = X^{n-k}D(X) + C(X)$$

Cyclic codes

If there are no errors on the reception T(X) will be exactly divisible by P(X) with ***no remainder***.

$$\frac{T(X)}{P(X)} = \frac{X^{n-k}D(X)}{P(X)} + \frac{C(X)}{P(X)} = \left(Q(X) + \frac{C(X)}{P(X)} \right) + \frac{C(X)}{P(X)} = Q(X)$$

$$\frac{X^{n-k}D(X)}{P(X)} = Q(X) + \frac{C(X)}{P(X)}$$

The last equality is valid because of the modulo2 arithmetic (a+a=0), so far we have the same function as for the CRC.

Cyclic codes

If one or more bit errors occur, then the received block Z(X) will be of the form:

$$Z(X) = T(X) + E(X)$$

Where E(X) is an *n*-bit error polynomial with a value '1' in each bit position that is an error in Z(X).

When the Z(X) is decoded we are performing Z(X)/P(X), which produces the (*n*-*k*) syndrome S(X):

$$\frac{Z(X)}{P(X)} = B(X) + \frac{S(X)}{P(X)}$$

where B(X) is the quotient and S(X) is the remainder. S(X) is a function of Z(X). But how does that help us to perform error correction.

Cyclic codes

Lets expand: $\frac{Z(X)}{P(X)} = B(X) + \frac{S(X)}{P(X)}$ $\quad \{ Z(X) = T(X) + E(X) \}$

$$\frac{T(X) + E(X)}{P(X)} = B(X) + \frac{S(X)}{P(X)}$$

remembering

$$\left\{ \frac{T(X)}{P(X)} = \frac{X^{n-k} D(X)}{P(X)} + \frac{C(X)}{P(X)} = \left(Q(X) + \frac{C(X)}{P(X)} \right) + \frac{C(X)}{P(X)} = Q(X) \right\}$$

We get:

$$Q(X) + \frac{E(X)}{P(X)} = B(X) + \frac{S(X)}{P(X)} \quad => \quad \frac{E(X)}{P(X)} = [Q(X) + B(X)] + \frac{S(X)}{P(X)}$$

Cyclic codes

$$\frac{E(X)}{P(X)} = [Q(X) + B(X)] + \frac{S(X)}{P(X)} \qquad \frac{Z(X)}{P(X)} = B(X) + \frac{S(X)}{P(X)}$$

E(X)/P(X) produces the same remainder as Z(X)/P(X). Therefore regardless of the initial pattern of bits, T(X), the syndrome value S(X) depends only on the error bits E(X).

If we can recover the E(X) from S(X), then we can correct the errors in Z(X) by simple addition:

$$Z(X) + E(X) = T(X) + E(X) + E(X) = T(X)$$

This is done with a table that map S(X) to E(X).

Cyclic codes example (pp 226)

Consider a (7,4) code with the generator polynomial $P(X)=X^3+X^2+1$

For the data block 1010, we have $D(X) = X^3+X$ and $T(X)=X^6+X^4$

Using: $$\frac{T(X)}{P(X)} = \frac{X^{n-k}D(X)}{P(X)} + \frac{C(X)}{P(X)}$$ we get:

$$
\begin{array}{rl}
 & X^3 + X^2 + 1 \quad \leftarrow Q(X) \\
P(X) \rightarrow X^3 + X^2 + 1 \; \big) & X^6 \qquad\quad X^4 \quad \leftarrow 2^3 D(X) \\
 & \underline{X^6 + X^5 + \qquad X^3} \\
 & \quad X^5 + X^4 + X^3 \\
 & \quad \underline{X^5 + X^4 + \qquad X^2} \\
 & \qquad\qquad X^3 + X^2 \\
 & \qquad\qquad \underline{X^3 + X^2 + \qquad 1} \\
 & \qquad\qquad\qquad\qquad 1 \quad \leftarrow C(X)
\end{array}
$$

$T(X) = 2^{n-k}D(X) + C(X) = X^6 + X^4 + 1 = 1\ 0\ 1\ 0\ 0\ 0\ 1$

Cyclic codes example (pp 226)

If we then calculate the code word for all possible blocks of data we get the following table.

This table can actually be used to directly map data block into code words instead of performing the actual calculation for each new data block transmission.

Data Block	Codeword
0000	0000000
0001	0001101
0010	0010111
0011	0011010
0100	0100011
0101	0101110
0110	0110100
0111	0111001
1000	1000110
1001	1001011
1010	1010001
1011	1011100
1100	1100101
1101	1101000
1110	1110010
1111	1111111

Cyclic codes example (pp 226)

For error correction we need to construct the syndrome table for single errors using:

$$\frac{E(X)}{P(X)} = [Q(X) + B(X)] + \frac{S(X)}{P(X)}$$

Lets start with 1000000 => $E(X)=X^6$

$$\begin{array}{r} X^3 + X^2 + X \quad \leftarrow Q(X) + B(X) \\ P(X) \rightarrow X^3 + X^2 + 1 \,\big/\, \overline{X^6 \qquad\qquad\qquad\qquad} \quad \leftarrow Z(X) \\ \underline{X^6 + X^5 + \quad\; X^3} \\ \quad X^5 + \quad\; X^3 \\ \quad \underline{X^5 + X^4 + \quad\; X^2} \\ \quad\quad X^4 + X^3 + X^2 \\ \quad\quad \underline{X^4 + X^3 + \quad\; X} \\ \quad\quad\quad\quad X^2 + X \quad \leftarrow S(X) \end{array}$$

The syndrome for the error pattern 1 0 0 0 0 0 0 0 is the 1 0 1. We do the same for all single error patterns and get a table of syndromes.

Cyclic codes example (pp 226)

Error pattern E	Syndrome S
0000001	001
0000010	010
0000100	100
0001000	101
0010000	111
0100000	011
1000000	110

Cyclic codes example (pp 226)

Now suppose that we received block 1101101 =>

$Z(X) = X^6+X^5+X^3+X^2+1$

Using: $\frac{Z(X)}{P(X)} = B(X) + \frac{S(X)}{P(X)}$ we get

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad\quad X^3 \qquad\qquad\qquad\qquad\qquad\qquad\quad \leftarrow B(X) \\
P(X) \rightarrow X^3 + X^2 + 1 \,\big/\, X^6 + X^5 + \qquad\quad X^3 + X^2 + 1 \quad \leftarrow Z(X) \\
\qquad\qquad\qquad\qquad\quad X^6 + X^5 + \qquad\quad X^3 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad X^2 + 1 \quad \leftarrow S(X)
\end{array}
$$

We get a syndrome S=101, which according to the syndrome table is equal to the error pattern E=0 0 0 1 0 0 0. Then:

T = 1 1 0 1 1 0 1 $\oplus$ 0 0 0 1 0 0 0 =1 1 0 0 1 0 1 which are equally to the data block 1 1 0 0.

BCH code

BCH codes are among the most powerful cyclic block codes and are widely used in wireless applications. For any pair of integers *m* and *t*, there is a binary (*n*,*k*) BCH codes with the following parameters:

Block length: $n = 2^m - 1$

Number of check bits: $n - k \leq mt$

Minimum distance: $d_{min} \geq 2t+1$

This code can correct all combinations of ***t*** or fewer errors. The generator polynomial for this code can be constructed from the factors of $X^{2m-1}+1$.

BCH code

Some BCH polynomial generators is:

n	k	t	$P(X)$
7	4	1	$X^3 + X + 1$
15	11	1	$X^4 + X + 1$
15	7	2	$X^8 + X^7 + X^6 + X^4 + 1$
15	5	3	$X^{10} + X^8 + X^5 + X^4 + X^2 + X + 1$
31	26	1	$X^5 + X^2 + 1$
31	21	2	$X^{10} + X^9 + X^8 + X^6 + X^5 + X^3 + 1$

Reed-Solomon codes

Reed Solomon (RS) codes are a widely used subclass of non-binary BCH codes.

With RS codes, data are processed in chunks of m bits, called symbols. An ($n.k$) RS code has the following parameters:

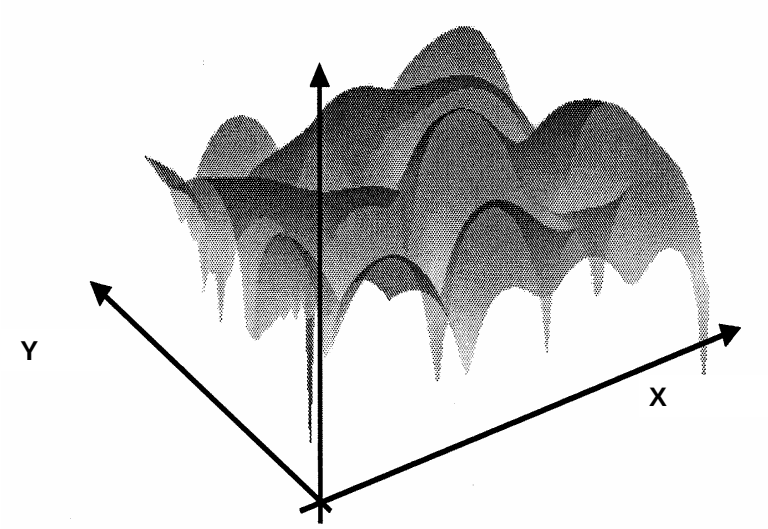
Symbol length:	m bits per symbol
Block length:	$n=2^m$-1 symbols=$m(2^m - 1)$ bits
Data length:	k symbols
Size of check code:	$n - k = 2t$ symbols = $m(2t)$ bits
Minimum distance:	$d_{min} = 2t + 1$ symbols

The encoding algorithm expands a block of k symbols to n symbols by adding $n - k$ redundant check symbols. Typically m is a power of 2.

Interleaving

Interleaving

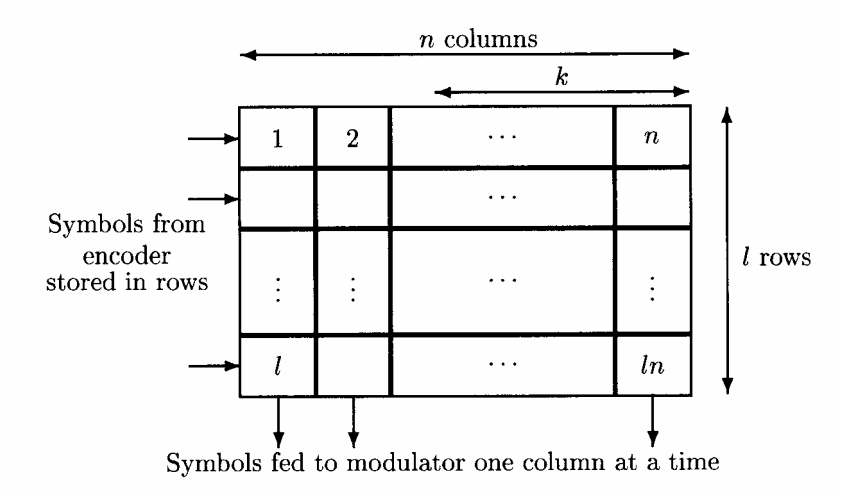
Interleaving



Interleaving

- The advantage of interleaving is that a burst error that affect a sequence of bits is spread out over a number of separate blocks at the receiver so that error correction is possible.
- Interleaving is accomplished by reading and writing data from a memory in different orders block interleaving.
- Or by convolutional interleaving, where there is no fixed block structure.

Block interleaving



Convolutional interleaving