Introduction

- Once the data is dispatched over the transmission medium, it may be altered in various ways so that the signals received at the remote end of a link differs from the transmitted signals.
- The effects of these adverse characteristics of a medium are known as transmission impairments and they often reduce transmission efficiency.
- In the case of binary data they may lead to errors, in that some binary zeros are transformed into binary ones and vice versa.
- The three main impairments are attenuation, distortion and noise which is the main factor and constrains the operation of any communications system.

Introduction

- To overcome the effects of such impairments it is necessary to introduce some form of error control.
- The first step in any form of error control is to detect whether any errors are present in the received data, a process which has been explored in some detail in previous lecture.
- Having detected the presence of errors there are two strategies commonly used to correct them:
  - Either further computations are carried out at the receiver to correct the errors, a process known as forward error control (correction)
  - Or a message is returned to the transmitter indicating that errors have occurred and requesting a retransmission of the data, which is known as feedback error control.

Introduction

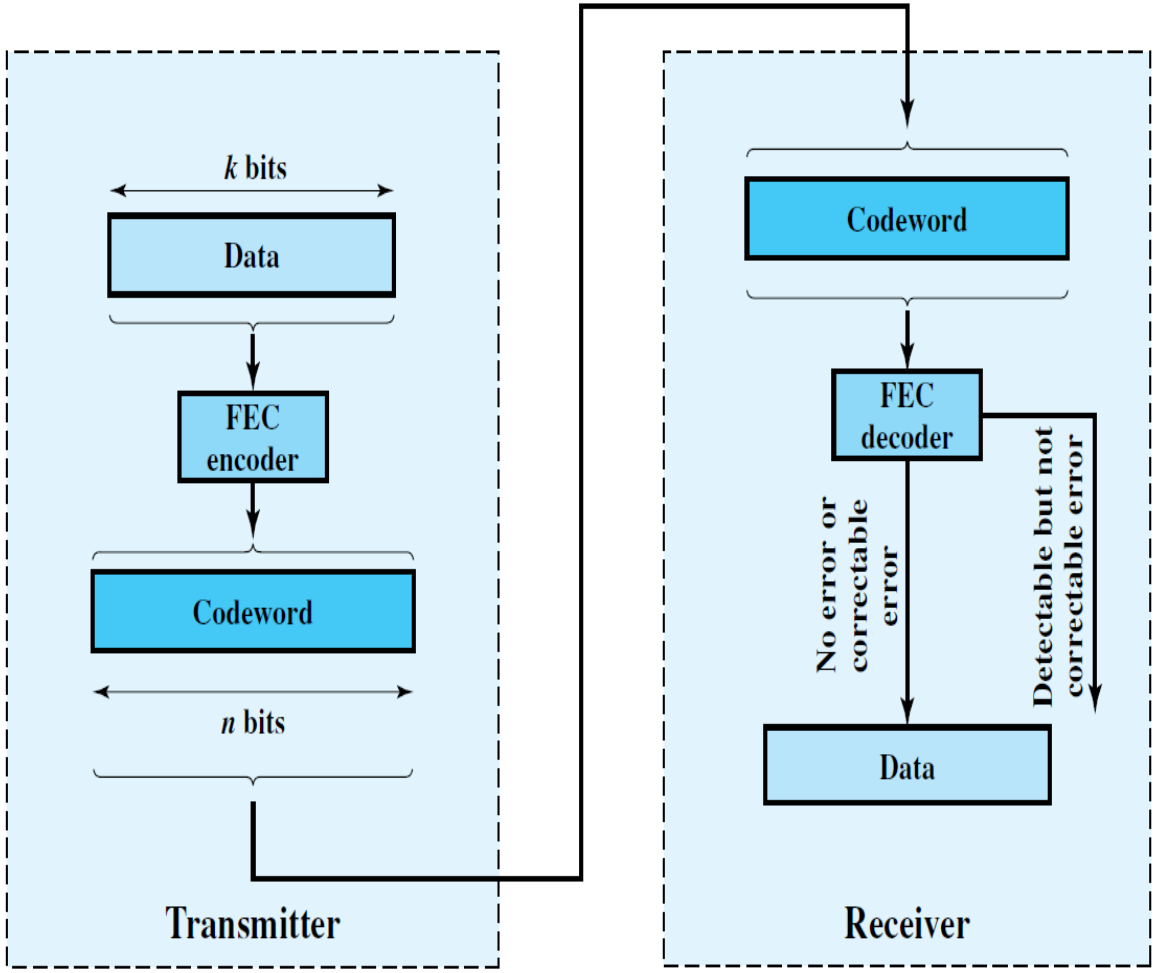
- It is possible to use codes that not only detect the presence of errors but also enable errors to be corrected.
- On a channel that are highly reliable, such as fiber optics, it is cheaper to use error detection code and just retransmit occasional block found to be faulty.
- However, on channels that make many errors, it is better to add enough redundancy to each block for the receiver to be able to figure out what the original block was.
- At a glance, it would seem that correction is always better, since with detection we are forced to discard the message and ask for another copy to be retransmitted. This uses bandwidth and may introduce latency while waiting for retransmission.
- Error correction tends to be more useful when:
  - Errors are quite probable
  - The cost of retransmission is too high such as latency involved in retransmitting a packet over a satellite.

Channel Coding

- Error control is also known as channel coding.
- Channel coding is the process of coding data prior to transmission over a communications channel so that if errors do occur during transmission it is possible to detect and possibly even to correct those errors once the data has been received.
- In order to achieve this error detection/correction some bit patterns need to be identified as error free at the receiver, whereas other bit patterns will be identified as erroneous.
- To increase the number of identifiable bit patterns at the receiver , additional bits, known as redundant bits, are added to the data or information bits prior to transmission.
- $code\ rate = \frac{k}{n}$; measurement of how many additional bandwidth is required to carry data at the same data rate as without the code.
- $redundancy = \frac{n-k}{k}$; ratio of redundant bits to data bits.

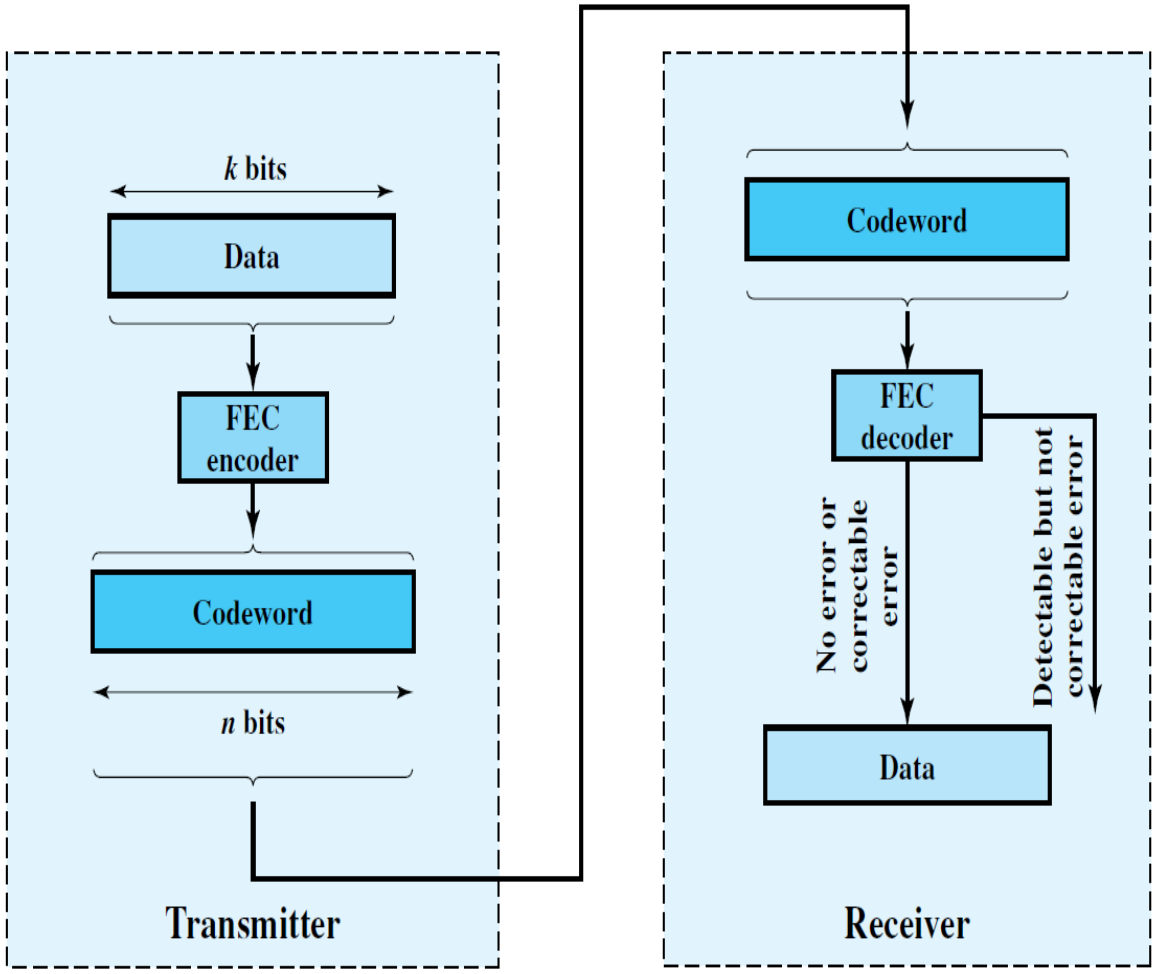
Channel Coding

- Figure shows in general how coding is done.
- On the transmission end, each k bit block of data is mapped into an n bit $(n > k)$ block called a codeword, using an FEC (forward error correction) encoder.
- The codeword is then transmitted.



Channel Coding

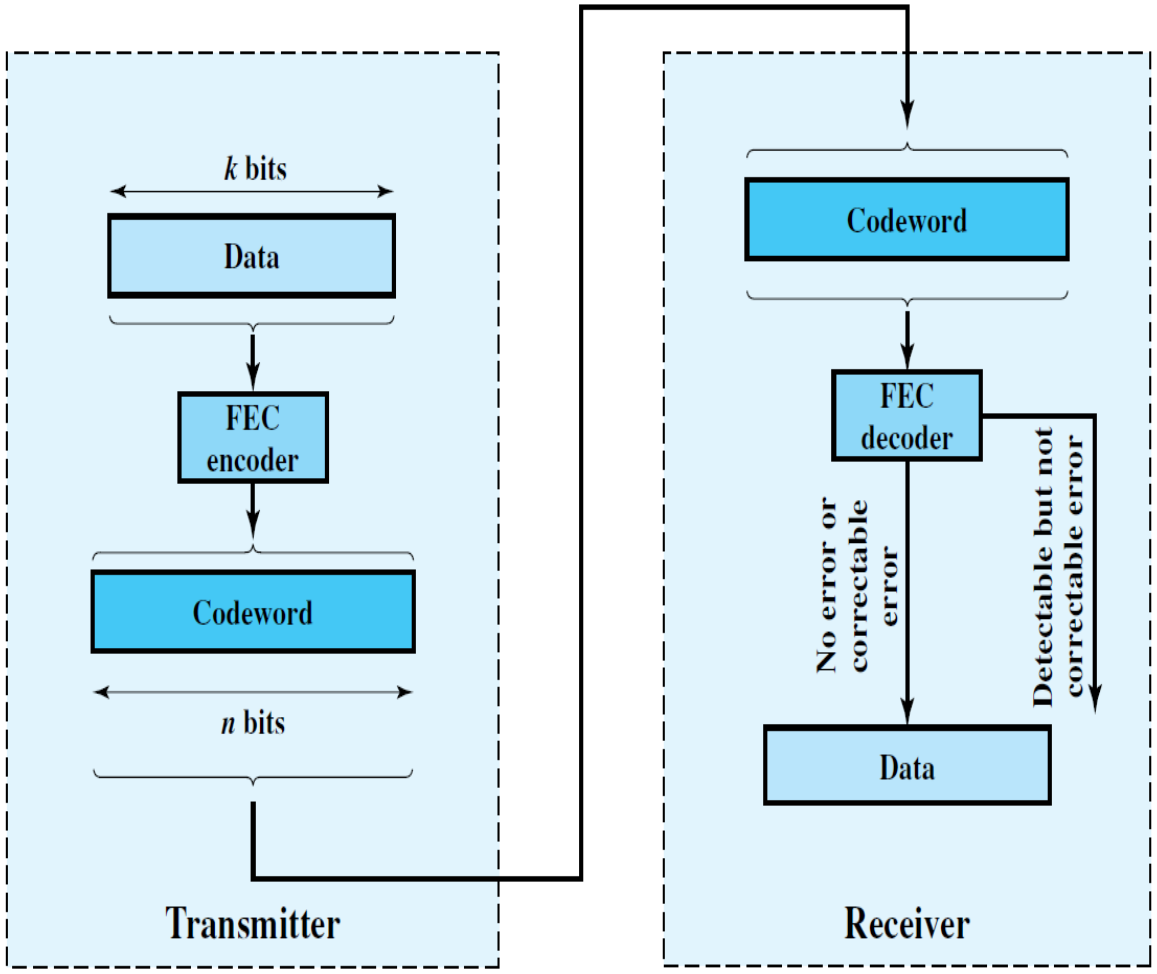
- During transmission, the signal is subject to impairments, which may produce bit errors in the signal.
- At the receiver, the incoming signal is demodulated to produce a bit string that is similar to the original codeword but may contain errors.



Channel Coding

- This block is passed through an FEC decoder, with one of four possible outcomes:

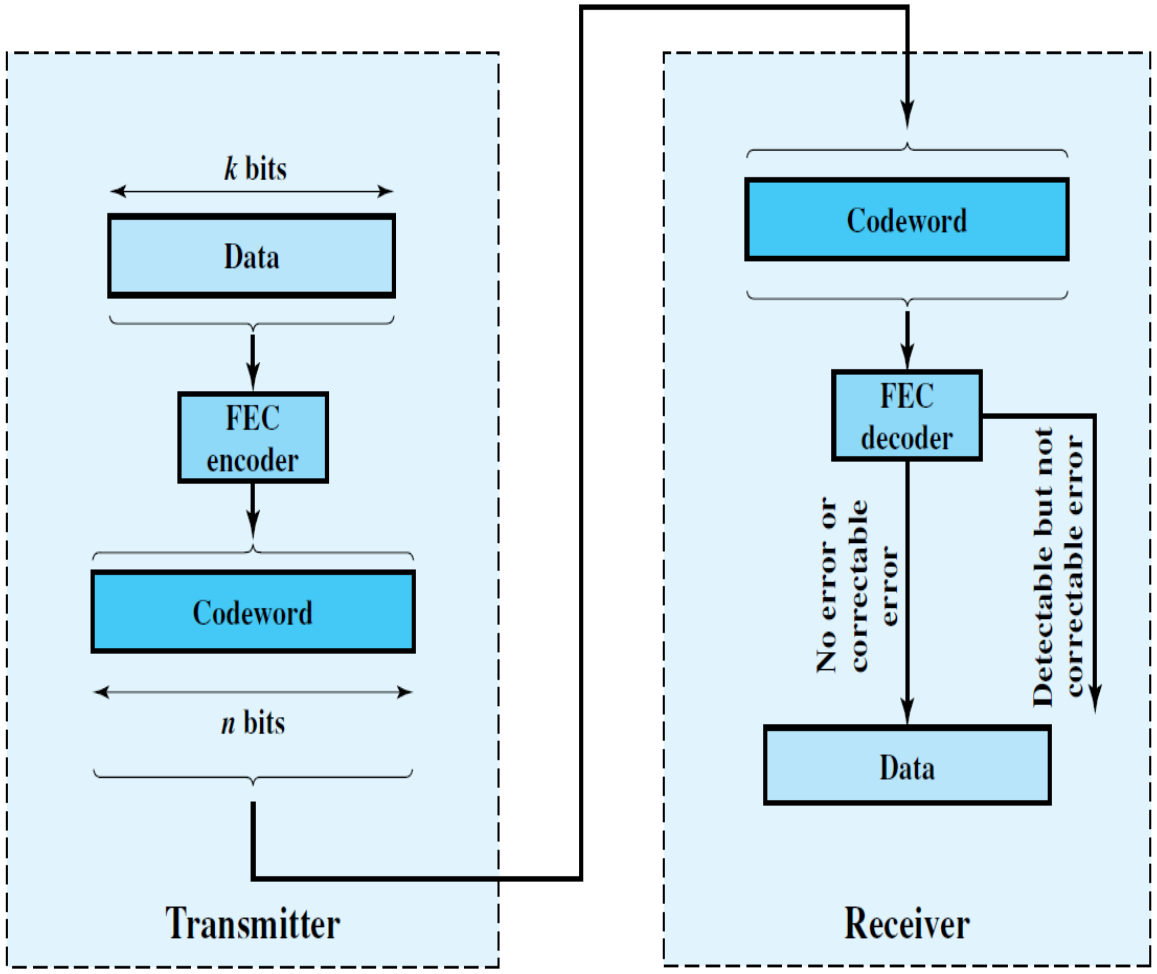
1. If there are no bit errors, the input to the FEC decoder is identical to the original codeword, and the decoder produces the original data block as output.



Channel Coding

- This block is passed through an FEC decoder, with one of four possible outcomes:

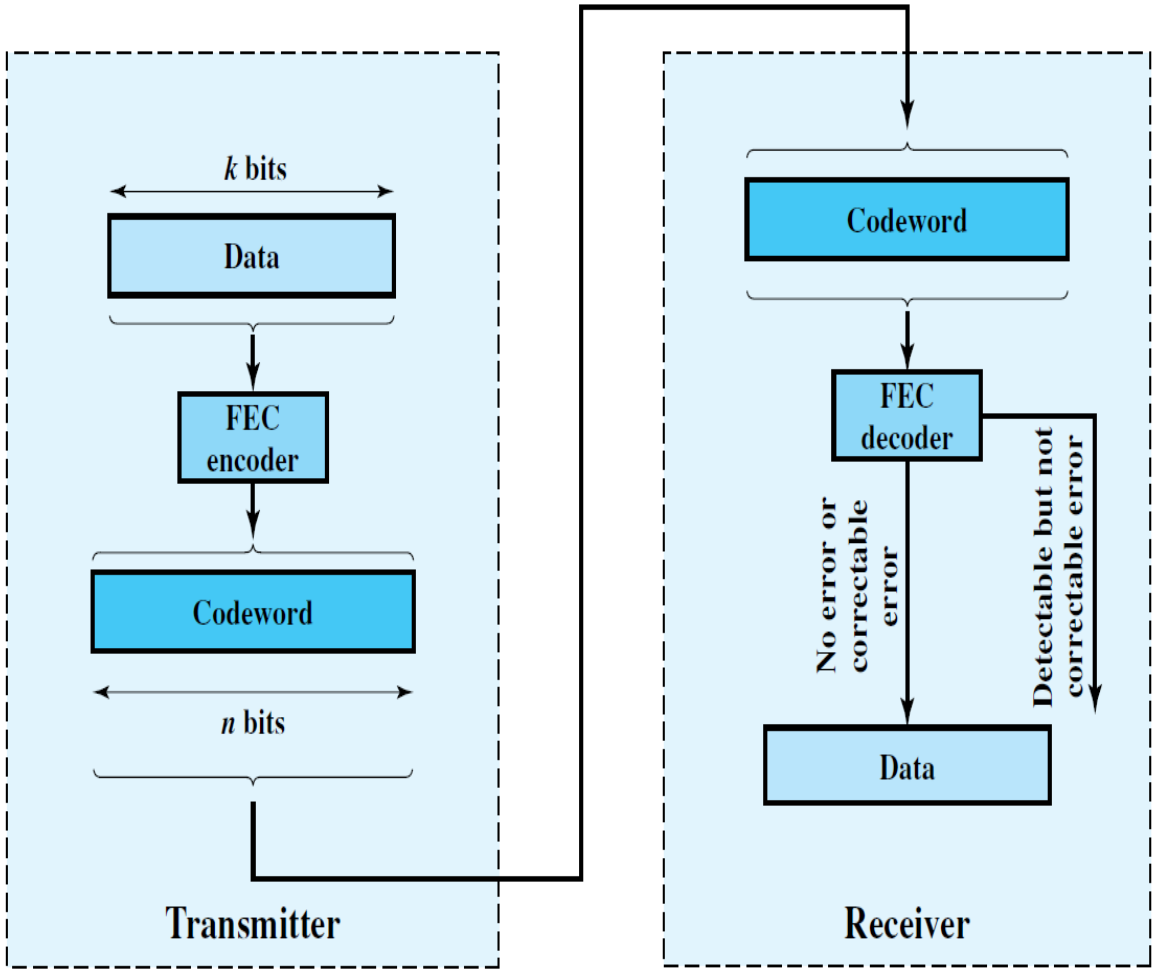
2. For certain error patterns, it is possible for the decoder to detect and correct those errors. Thus, even though the incoming data block differs from the transmitted codeword, the FEC decoder is able to map this block into the original data block.



Channel Coding

- This block is passed through an FEC decoder, with one of four possible outcomes:

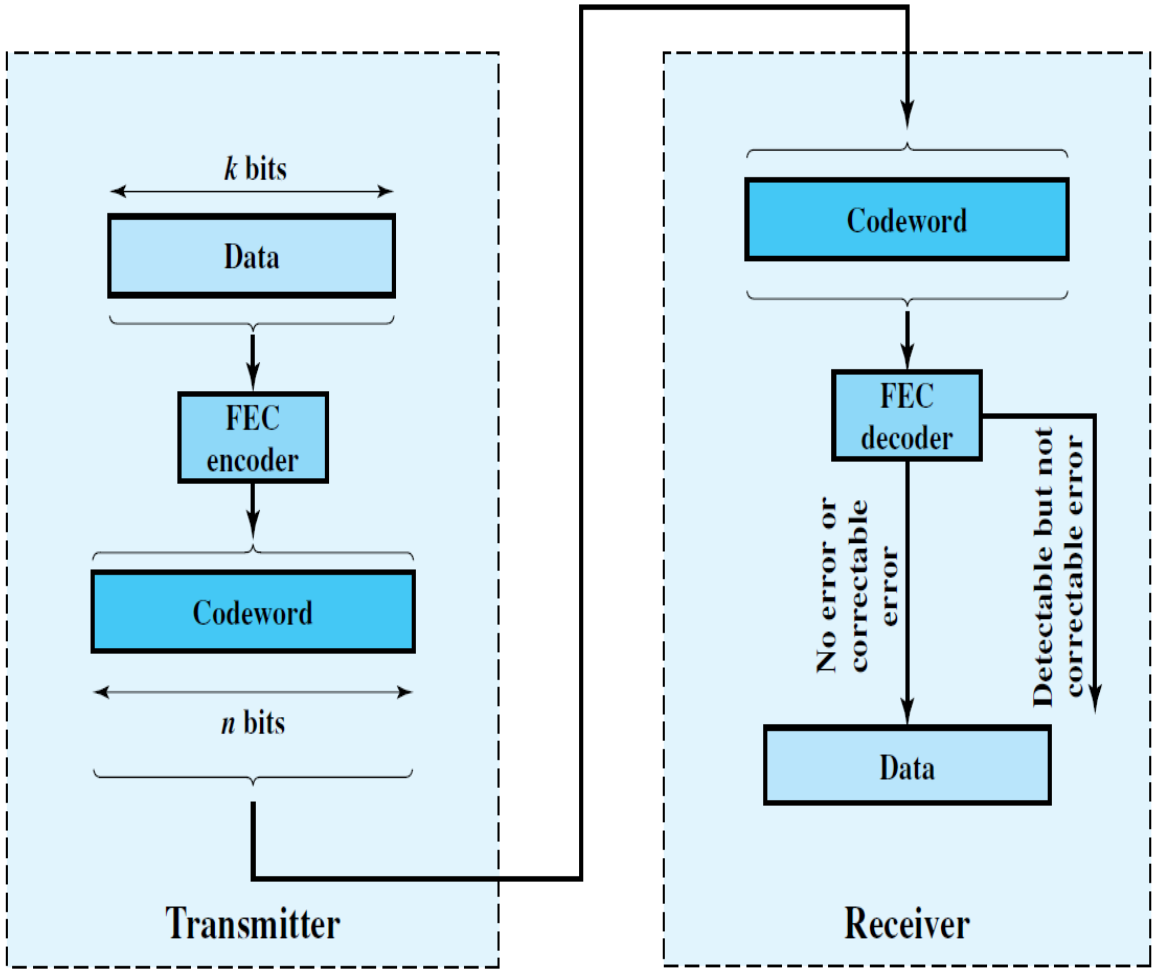
3. For certain error patterns, the decoder can detect but not correct the errors. In this case, the decoder simply reports an uncorrectable error.



Channel Coding

- This block is passed through an FEC decoder, with one of four possible outcomes:

4. For certain, typically rare, error patterns, the decoder does not detect that any errors have occurred and maps the incoming n bit data block into a k bit block that differs from the original k bit block.



Linear Block Codes

- Various different types of code are available for use in channel coding but the simplest and most commonly used are called linear block codes.
- In a block code:
  - The user data stream is segmented into blocks of k bits
  - Each k bit block is encoded independently of other blocks to an n bit codeword $(n > k)$
  - The code rate is k/n
  - The set of all possible source words has size 2^k
  - The set of all possible words in the code space has size 2^n
  - Out of these the code uses only 2^k out of 2^n elements
  - These are called valid codewords

Linear Block Codes

- Channel errors can turn:
  - a valid codeword into a word from the set of $2^n - 2^k$ unused codewords, then the decoder must guess which was the transmitted codeword.
- When facing an unused codeword y, decoders essentially look for the valid codeword that is "closest" to y.

The Hamming Codes

- This is an important group of error correcting codes pioneered by R.W. Hamming in the 1950s.
- They involve the production of check(redundant) bits by adding together different groups of data bits.
- No of redundant bits = $2^r > k + r + 1$ ('k' are data bits, 'r' are redundant bits)
- The type of addition used is known as modulo 2 addition and is equivalent to normal binary addition without any carries.
- Hamming codes can detect up to two bit errors or correct one bit error.

The Hamming Codes

- We shall consider a Hamming $(7,4)$ code, in which three check bits ($c1$, $c2$ and $c3$) are combined with four information bits ($k1$, $k2$, $k3$ and $k4$) to produce a block of data of length $n = 7$.
- Three check equations are used to obtain the three check bits of this Hamming $(7,4)$ code as follows:

$$c1 = k1 \oplus k2 \oplus k4$$
$$c2 = k1 \oplus k3 \oplus k4$$
$$c3 = k2 \oplus k3 \oplus k4$$

- Given data bits = 1010 then $k1 = 1$, $k2 = 0$, $k3 = 1$ and $k4 = 0$ and the check bits obtained from the three check equations above are as follows:

$$c1 = k1 \oplus k2 \oplus k4 = 1 \oplus 0 \oplus 0 = 1$$
$$c2 = k1 \oplus k3 \oplus k4 = 1 \oplus 1 \oplus 0 = 0$$
$$c3 = k2 \oplus k3 \oplus k4 = 0 \oplus 1 \oplus 0 = 1$$

- The codeword is obtained by adding the check bits to the end of the information bits and therefore the data 1010101 will be transmitted (data bits first).

The Hamming Codes

- A complete set of codewords can be obtained in a similar way:

Code word #	K1	K2	K3	K4	C1	C2	c3	Code word #	K1	K2	K3	K4	C1	C2	c3
0	0	0	0	0	0	0	0	8	1	0	0	0	1	1	0
1	0	0	0	1	1	1	1	9	1	0	0	1	0	0	1
2	0	0	1	0	0	1	1	10	1	0	1	0	1	0	1
3	0	0	1	1	1	0	0	11	1	0	1	1	0	1	0
4	0	1	0	0	1	0	1	12	1	1	0	0	0	1	1
5	0	1	0	1	0	1	0	13	1	1	0	1	1	0	0
6	0	1	1	0	1	1	0	14	1	1	1	0	0	0	0
7	0	1	1	1	0	0	1	15	1	1	1	1	1	1	1

Hamming Distance

- An error that occurs in a transmitted codeword can be detected only if the error changes the codeword into some other bit pattern that does not appear in the code.
- The number of positions by which any two codewords in a code differ is known as the Hamming distance(d_H).
- Taking codewords 3 and 8 as an example, we have:
  - Codeword 3 0 0 1 1 1 0 0
  - Codeword 8 1 0 0 0 1 1 0
- These two codewords differ in positions 1, 3, 4 and 6 , so that the distance between these two words is four.

Hamming Distance

- Since all linear block codes contain the all zeros codeword, then an easy way to find the minimum distance(d_{min}) of a code is to compare a non zero codeword which has the minimum number of ones with the all zeros codeword.
- Thus the minimum distance of a code is equal to the smallest number of ones in any non zero codeword, which in the case of this Hamming $(7,4)$ code is three.
- If the codewords of a code differ in three or more positions then error correction is possible since an erroneous bit pattern will be 'closer' to one codeword than another.

Hamming Codes - Properties

- If we take codewords 8 and 10 as an example, we have:
  - Codeword 8 1 0 0 0 1 1 0
  - Codeword 10 1 0 1 0 1 0 1

1000110 (codeword 8) (d_H = 1)

1010110(received codeword)

- The distance between these two codewords is three.
- If codeword 8 is transmitted and an error occurs in bit 3 then the received data will be: 1 0 1 0 1 1 0
- This is not one of the other 15 Hamming (7,4) codewords since an error has occurred.
- Furthermore, the most likely codeword to have been transmitted is codeword 8 since this is the nearest(hamming distance of 1) to the received bit pattern. Thus, it should also be possible to correct the received data by making the assumption that the transmitted codeword was number 8.

Hamming Codes - Properties

- If, however, a second error occurs in bit 7 then the received bit pattern will be $1\ 0\ 1\ 0\ 1\ 1\ 1$
- It should still be possible to detect that an error has occurred since this is not one of the 16 codewords.
- However, it is no longer possible to correct the errors since the received bit pattern has changed in two places.
- Thus, this Hamming $(7,4)$ code is able to detect two errors but correct only one error.
- In general, if the minimum distance of a code is d_{min}, then $d - 1$ errors can normally be detected using a linear block code and $mod(d - 1)/2$ can be corrected.

Hamming Codes - Properties

- All Hamming codes (indeed, all linear block codes) possess the mathematical property that if we add any two codewords together (modulo 2 addition) then the resulting sum is also a codeword.
- For example, if we add codewords 1 and 2 :

$$\begin{array}{rl} & 000111 \\ \oplus & 001001 \\ \hline & 001110 \end{array}$$ Which is codeword 3

- This allows us to represent a whole code by means of a subset of codewords, since further codewords can simply be obtained by modulo 2 addition.
- The subset of codewords is often expressed as a matrix known as a generator matrix, *G*.
- The codewords chosen are normally powers of 2, that is codewords 1, 2,4, 8.

Hamming Codes – Generator Matrix

- A suitable generator matrix for the Hamming (7,4) code consists of the following four codewords:

$$G = \begin{bmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{bmatrix}$$

- The matrix has four rows and seven columns, that is it has dimensions 4×7 $(k \times n)$.
- The whole code can be generated from this matrix just by adding together rows, and it is for this reason that it is called a generator matrix.
- A further reason for the generator matrix being so named is that it can be used to generate codewords directly.

Hamming Codes – Encoding

- Information consisting of the bits 1010 is to be encoded using the Hamming $(7,4)$ code.
- Use the generator matrix to obtain the codeword to be transmitted.
- The codeword is obtained by multiplying the four information bits (expressed as a row vector) by the generator matrix as follows:

$$[1\ 010] \times \begin{bmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{bmatrix}$$

- The multiplication is achieved by multiplying each column of the generator matrix in turn by the row vector as follows:

$$[(1 \times 1 \oplus 0 \times 0 \oplus 1 \times 0 \oplus 0 \times 0), (1 \times 0 \oplus 0 \times 1 \oplus 1 \times 0 \oplus 0 \times 0), (1 \times 0 \oplus 0 \times 0 \oplus 1 \times 1 \oplus 0 \times 0), (1 \times 0 \oplus 0 \times 0 \oplus 1 \times 0 \oplus 0 \times 1), (1 \times 1 \oplus 0 \times 1 \oplus 1 \times 0 \oplus 0 \times 1), (1 \times 1 \oplus 0 \times 0 \oplus 1 \times 1 \oplus 0 \times 1), (1 \times 0 \oplus 0 \times 1 \oplus 1 \times 1 \oplus 0 \times 1)] = 1\ 0\ 1\ 0\ 1\ 0\ 1$$

- Note that this process is, in fact, the same as using the check equations to obtain the check bits and then add the check bits to the end of the information bits.

Hamming Codes – Check Matrix

- It is also possible to express the three check equations of the Hamming $(7,4)$ code in the form of a matrix known as the check matrix H, as follows:

Check equations

$$c1 = k1 \oplus k2 \oplus k4$$

$$c2 = k1 \oplus k3 \oplus k4$$

$$c3 = k2 \oplus k3 \oplus k4$$

Check Matrix

$$\mathrm{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$k1 k2 k3 k4 c1 c2 c3$

- The check matrix is obtained by having each row of the matrix correspond to one of the check equations in that if a particular bit is present in an equation, then that bit is marked by a one in the matrix. This results in a matrix with dimensions 3×7 $(c \times n)$.

Hamming Codes – Check Matrix

- If we now compare the two types of matrix we note that the generator matrix has an identity matrix consisting of a diagonal of ones to its left and the check matrix has this identity matrix to its right.

$$G = \begin{bmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{bmatrix} \qquad \mathrm{H} = \begin{bmatrix} 1101100 \\ 1011010 \\ 0111001 \end{bmatrix}$$

- When a generator or check matrix conforms to this pattern, it is in standard echelon form.
- A further point to note is that if the echelons are removed from the two matrices, then what remains is the transpose of each other.
- In the case of the Hamming $(7,4)$ code:

From Check Matrix $\begin{bmatrix} 1101 \\ 1011 \\ 0111 \end{bmatrix}$ $is\ the\ transpose\ of$ $\begin{bmatrix} 110 \\ 101 \\ 011 \\ 111 \end{bmatrix}$ From Generator Matrix

Hamming Codes – Check Matrix

Example: The generator matrix for a Hamming (15,11) code is as follows:

$$G = \begin{bmatrix} 100000000001100 \\ 010000000000110 \\ 001000000000011 \\ 000100000001101 \\ 000010000001010 \\ 000001000000101 \\ 000000100001110 \\ 000000010000111 \\ 000000001001111 \\ 000000000101011 \\ 000000000011001 \end{bmatrix}$$

Obtain the check matrix.

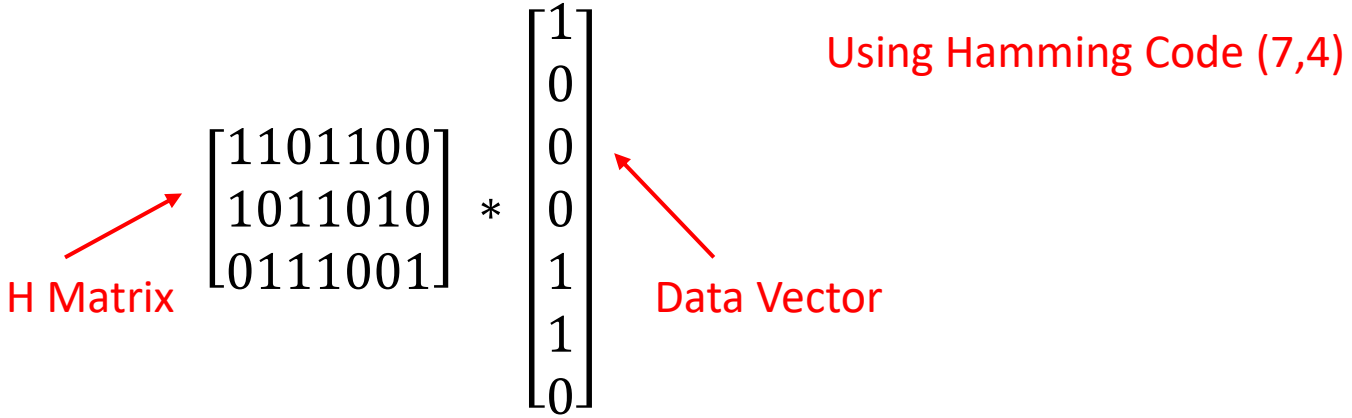
Hamming Codes – Check Matrix

- The code has a block length $n = 15$, consisting of $k = 11$ information bits and $c = 4$ check bits.
- The generator matrix has dimensions 11×15, and includes an 11×11 identity matrix (an echelon) to the left.
- The check matrix has dimensions 4×15 and contains a 4×4 identity matrix to its right hand side.
- The rest of the check matrix is obtained by removing the identity matrix from the generator matrix and transposing what is left. The check matrix is as follows:

$$H = \begin{bmatrix} 100110101111000 \\ 110101111000100 \\ 011010111100010 \\ 001101011110001 \end{bmatrix}$$

Hamming Codes – Decoding

- To determine whether received data is error free or not, it is necessary for all of the check equations to be verified.
- This can be done by recalculating the check bits from the received data or, alternatively, received data can be checked by using the check matrix.
- As its name implies, the check matrix can be used to check the received data for errors in a similar way to using the generator matrix to generate a code word.
- The check matrix, H, is multiplied by the received data expressed as a column vector:



$$\begin{bmatrix} 1101100 \\ 1011010 \\ 0111001 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Hamming Codes – Decoding

- This time the multiplication is achieved by multiplying each row of the check matrix in turn by the received data vector, as follows:

$$\begin{matrix}(1 \times 1 \oplus 1 \times 0 \oplus 0 \times 0 \oplus 1 \times 0 \oplus 1 \times 1 \oplus 0 \times 1 \oplus 0 \times 0) \\ (1 \times 1 \oplus 0 \times 0 \oplus 1 \times 0 \oplus 1 \times 0 \oplus 0 \times 1 \oplus 1 \times 1 \oplus 0 \times 0) \\ (0 \times 1 \oplus 1 \times 0 \oplus 1 \times 0 \oplus 1 \times 0 \oplus 1 \times 0 \oplus 0 \times 1 \oplus 1 \times 0)\end{matrix} = \begin{bmatrix}0 \\ 0 \\ 0\end{bmatrix}$$

- If, as is the case here, the received data is error free then the result of this multiplication is zero.
- This result, which in this case is a 3 bit vector, is known as the syndrome.

Hamming Codes – Decoding

Example: Information consisting of four ones is to be transmitted using the Hamming $(7,4)$ code. (a) Determine the transmitted codeword.

(b) If an error occurs in bit 4 during transmission, determine the syndrome.

(c) Show how the syndrome can be used to correct the error.

To determine the transmitted codeword we can use the three check equations to determine the check bits:

$$c1 = k1 \oplus k2 \oplus k4 = 1 \oplus 1 \oplus 1 = 1$$

$$c2 = k1 \oplus k3 \oplus k4 = 1 \oplus 1 \oplus 1 = 1$$

$$c3 = k2 \oplus k3 \oplus k4 = 1 \oplus 1 \oplus 1 = 1$$

- The transmitted codeword is therefore seven ones as follows:

$$k1\ k2\ k3\ k4\ c1\ c2\ c3 = \quad 1\ 1\ 1\ 1\ 1\ 1\ 1$$

Hamming Codes – Decoding

(b) If an error occurs in bit 4 during transmission, determine the syndrome.

If an error occurs in bit 4 then the received data is 1110111. To check whether there is an error in the received data, we multiply by the check matrix:

$$\begin{bmatrix}1101100\\1011010\\0111001\end{bmatrix} * \begin{bmatrix}1\\1\\1\\0\\1\\1\\1\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix}$$

syndrome

Column 4

- An error that was occurred has caused the syndrome to be non zero. Furthermore the position of the error can be located by comparing the syndrome with the columns of the check matrix. In this case the syndrome provides us with all we need to know about the error since its numerical value equates with column 4 of the check matrix, thus indicating an error in bit 4.