Block coding: some definitions

- Performed by subdividing the message stream into a sequence of blocks each "k" bits long
- Each "k" bit data block is mapped to an "n" digit block of output digits by the encoder, where $n > k$

SOURCE	k-bit mesage word
ENCODER	n-bit codeword

Block coding: some definitions

- k/n is the *code rate* (or *code efficiency*)
- $1\text{-}k/n$ is the *code redundancy*

- The encoder produces an (n,k) code, e.g. a (7,4) code maps 4 message bits into 7 output bits. The 7-4=3 extra digits are called *parity check bits*

Block coding: some definitions

For example; (15,11) code has ...

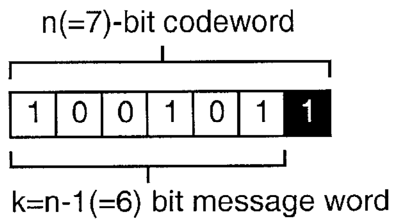
- 15 codeword bits $(=n)$
- 11 message bits $(=k)$
- 4 parity check bits $(=r)$
- 11/15 code rate, or efficiency, <1
- 4/15 code redundancy

Block coding: single parity

- The simplest (trivial!) example of an (n,k) block code is where $n=k+1$ - that is the addition of a single *parity* bit
- The additional bit, a *parity check bit* is used to *detect* an odd number of errors. There is no error *correction* capability
- In many systems it is unlikely that more than one error will occur

Block coding: single parity

- The *even* parity scheme is most commonly used in single extra bit type encoding
- The single parity bit is the modulo-2 sum of the message bits, e.g.,



Modulo-2 (mod-2) arithmetic

- mod-2 *addition*

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

- mod-2 *multiplication*

$0 \otimes 0 = 0$

$0 \otimes 1 = 0$

$1 \otimes 0 = 0$

$1 \otimes 1 = 1$

- mod-2 *subtraction* is the same as *addition*
- mod-2 *division* is the same as *multiplication*

Block coding: single parity

- At the receiver the mod-2 sum is repeated on the *whole codeword,* if the resultant sum is not equal to 0, then an odd number of errors has occurred.
- Note, an even number of errors, e.g. 2 *cannot* be detected.

Block coding: parity interleaving

- In many applications errors tend to occur in *bursts* of several successive digits (e.g. short term fading on radio channels)
- Multiple errors cause havoc on parity checking, so the parity bits are often *interlaced* so that the digits that are being checked are widely spaced ...

m m m c m m m c m m m c m m m c

Block coding: parity correction

- Parity check coding can be readily extended to provide *error correction*
- In order to correct an error, we need to detect the error and find the location of the error. This requires two checks on the same codeword in two different positions
- The simplest system is the square array parity check (or *rectangular block code*) in which the k message bits are arranged in a square matrix, with a parity check for each row and column

Block coding: parity correction

e.g. for k=9 message bits …

0	1	0	1			1	1	0	0
1	0	0	1			1	0	0	1
0	1	1	0			0	1	1	0
1	0	1				0	0	1	

0 1 0 1 1 0 0 1 0 1 1 0 1 0 1   1 1 0 0 1 0 0 1 0 1 1 0 0 0 1

Each complete (n=15) digit codeword is read out row by row for transmission and is re-matrixed at the receiver. A single error in the received data shows up as row and column parity errors

Block coding: parity correction

- For multiple errors it may be possible to detect the presence of the errors without being able to locate and correction them
- No correction and detection scheme can protect against *all* possible combinations of errors
- There are however more effective error detection and correction techniques which we will now investigate ...

Linear block codes

- A message block of k information bits input to the channel encoder is denoted by a k-vector, i.e. a vector k digits long;

$$M = \{m_1, m_2, m_3, \ldots, m_k\} \text{ where } m_i \in \{0,1\}$$

- Assuming the message is binary, there are 2^k distinct encoder input messages
- Each unique message block is mapped into a codeword of length n-bits;

$$C = \{c_1, c_2, c_3, \ldots, c_n\} \text{ where } n > k$$

Linear block codes

- Only 2^k of the possible 2^n codewords are used.
- This makes it possible to generate codewords selected so that a number of transmission e%rOrs must occur before one codeword is confused with another at the receiver
- The most important class of block codes are *linear* block codes

UNIVERSITY OF BATH ELEC0031 Digital Communications 13

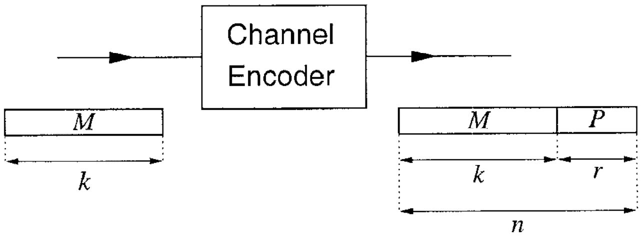
Properties of linear block codes

- The parity bits are given by appropriate mod-2 sums taken from the message block
- Each of the 2^k code vectors can be expressed as a linear combination (mod-2 sum) of k linearly independent code vectors
  e.g. if
  $C_x = 1\,0\,0\,0\,1\,1$, and $C_y = 0\,1\,0\,1\,0\,1$
  are code vectors so is $C_z = C_x \oplus C_y$
- The all-zero vector is a code vector corresponding to the all-zero message vector

UNIVERSITY OF BATH ELEC0031 Digital Communications 14

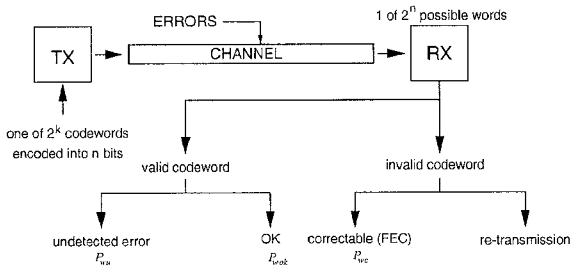
Properties of linear block codes

- Codes in which the message bits appear at the beginning of the codeword are called *systematic linear block codes*
- This constraint does not restrict the performance of the resultant code



Decoding linear block codes

- The decoder is presented with a word of n bits which has been corrupted by errors
- The decoded word is therefore not necessarily one of the 2^k codewords



Hamming Distance & Hamming Weight

A typical block code: $(n,k)=(6,3)$

Message	Codeword
000	*000 000*
001	*001 110*
010	*010 101*
011	*011 011*
100	*100 011*
101	*101 101*
110	*110 110*
111	*111 000*

Code rate $R=3/6$

Hamming Distance & Hamming Weight

- Since $n=6$ there are $2^6=64$ possible codewords, of which only 8 $(2^k=2^3=8)$ have been used
- Each codeword has been chosen so that it differs from any other in at least 3 positions.
- Hence 3 transmission errors can transform one codeword into another …
- However, 1 or 2 errors can be detected …
- A single error can be corrected (as the received codeword will be *closest* to the true codeword)

Hamming Distance & Hamming Weight

- The number of differences between two adjacent codewords is an important parameter and is called the ***Hamming distance***
- The minimum Hamming distance is called the ***minimum distance***
- The ***Hamming weight*** of a codeword C is defined as the number of "1"s in C
- The ***minimum distance*** of a linear block code is equal to the ***minimum weight*** of any non-zero codeword

Hamming Distance & Hamming Weight

- A linear block code with minimum distance $d_{\min}$ can correct up to

$$\left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

errors.

- For single error correction then $d_{\min}=3$, and the $n\text{-}k=r$ check or parity bits have to indicate in which position of the codeword the error has occurred