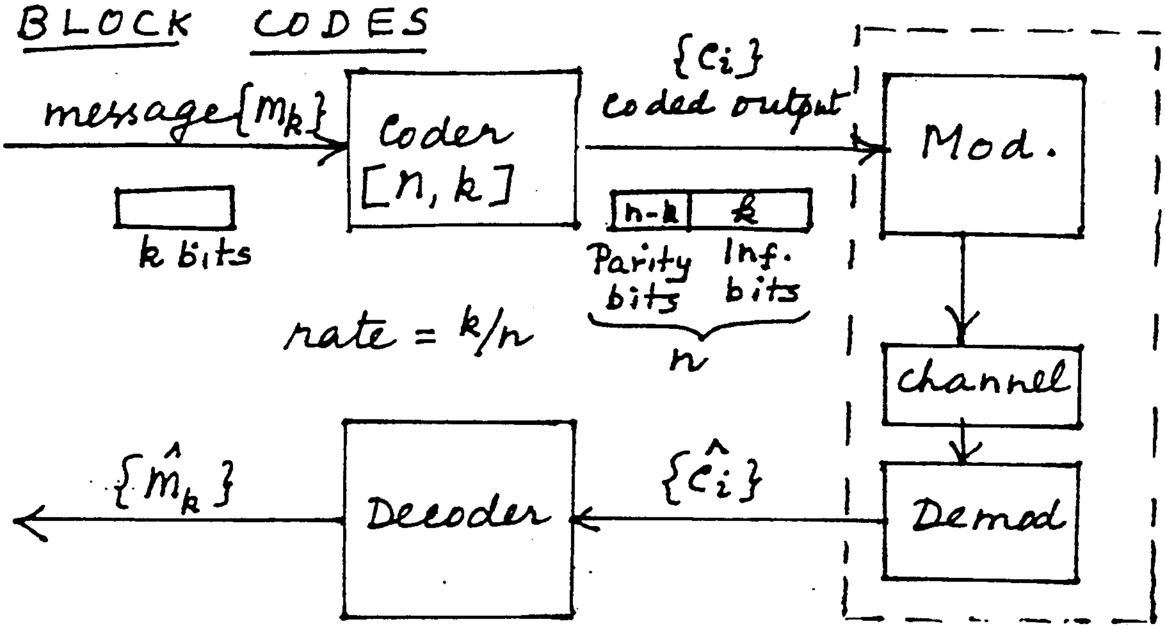
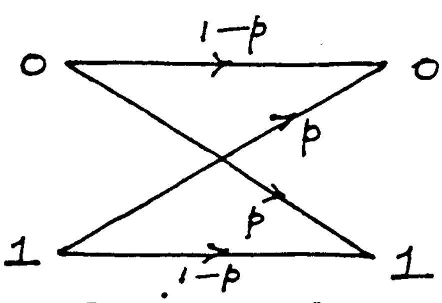
24. ERROR CONTROL CODING

- Error detection (ARQ – Automatic Repeat Request)
- Error correction (FEC – Forward Error Correction)
- Block, convolutional codes
- BLOCK CODES



Discrete channel.

- Parity bits are related to information bits; decoder uses this "relation" to determine the message



p = error probability $= \text{Prob}(\hat{c}_i \neq c_i)$

Binary Symmetric channel (hard decision)

- Assume $p = 8 \cdot 10^{-4}$

  Want $\text{Prob}[\hat{m}_k \neq m_k] < 10^{-4}$

- <u>Repetition code</u>

$$\underbrace{0}_{\substack{1\\ \text{inf. bit}}} \rightarrow \underbrace{0\,0}_{\text{2 parity bits}}\,0 \qquad 1 \rightarrow 1\,1\,1$$

- Decoding rule : max. likelihood, min. distance, majority logic
- <u>Hamming Weight</u> of a <u>code word</u>

$$\underline{s}_1 = \text{codeword}_1 = 1\,0\,1\,1\,0\,1$$
$$\underline{s}_2 = \text{codeword}_2 = 0\,0\,1\,1\,0\,0$$

$W[\underline{s}_i]$ = no. of 1's in code word $\underline{s}_i$

e.g. $W[\underline{s}_1] = 4$; $W[\underline{s}_2] = 2$

- If $\underline{s}_i = [s_{i1}\ s_{i2}\ \dots\ s_{in}]$

$$W[\underline{s}_i] = \sum_{k=1}^{n} s_{ik}$$

- Hamming weight is a proper norm
- <u>Hamming Distance</u> between code words

$d(\underline{s}_i, \underline{s}_j)$ = no. of positions in which $\underline{s}_i$ and $\underline{s}_j$ differ

e.g. $d(\underline{s}_1, s_2) = 2$

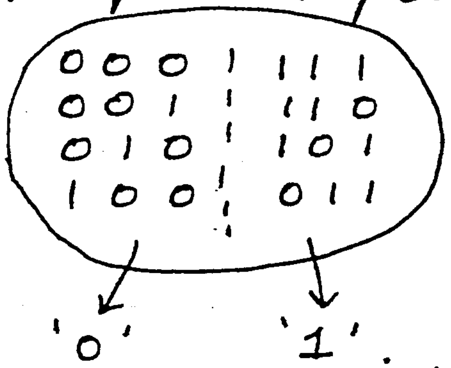
- $d(\underline{s}_i, \underline{s}_j) = W(\underline{s}_i \oplus \underline{s}_j)$ ↳ mod 2

$$= \sum_{k=1}^{n} (s_{ik} \oplus s_{jk})$$

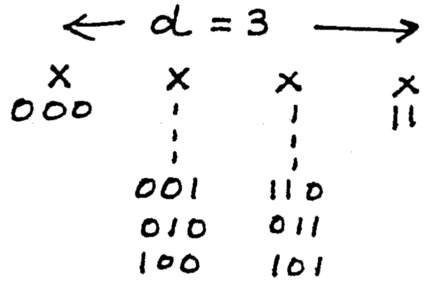
e.g. $d(\underline{s}_1, s_2) = W(\underline{s}_1 \oplus \underline{s}_2)$

$$= W(100001)$$
$$= 2$$

- Minimum distance decoding $\equiv$ decoded codeword is closest in Hamming distance to received word.
- Example : Repetition code



Space of received words



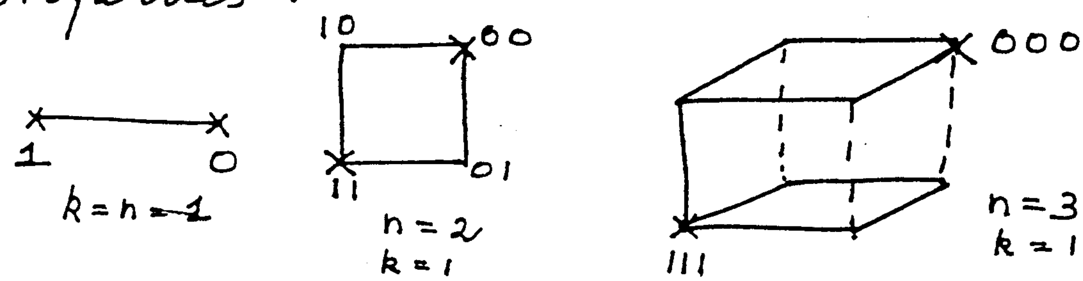
- min. distance $\equiv$ majority logic
  $\equiv$ majority of '1's or '0's

- $d = 3 \Rightarrow$ can correct 1 error
  can detect 2 errors

correct · $\boxed{d = 2M+1 \Rightarrow \text{correct upto } M \text{ errors}}$

detect $\boxed{= M+1 \Rightarrow \text{detect upto } M \text{ errors}}$

- Given no. of errors to be corrected, have to choose codewords to satisfy distance properties.



- rate $= k/n$ $\quad 0 \to 000$ $\quad$ rate $= 1/3$
  $1 \to 111$
- choose 2^k codewords out of 2^n possibilities

- P_{we} = word error rate
  = Pr (2 or more bits in error in a triplet)
  $= \binom{3}{2} p^2 (1-p) + \binom{3}{3} p^3$
  $= 3p^2 - 2p^3$
- $\binom{n}{k}$ = binomial coefficient $= \dfrac{n!}{k!\,(n-k)!}$
- For $p = 8 \cdot 10^{-4}$
  $P_{we} = 3p^2 - 2p^3 \approx 3p^2$
  $= 0.0192 \times 10^{-4} < 10^{-4}$ as required
- Prob. of undetected errors
  = Prob (3 errors) $= p^3 = 512 \cdot 10^{-12}$
- Performance - efficiency tradeoff.
- <u>Single - Parity</u> <u>check code</u>
- Detection of single error
- $\underbrace{b_1}_{1} \; \underbrace{m_1 \; m_2 \; \ldots \; m_k}_{k}$ $(k+1, k)$ code

  parity bit (check bit)
- check digit added such that the Hamming weight of each codeword is <u>even</u> or odd
  ↳ even no. of '1's (even parity)
- e.g

b_1	m_1	m_2
0	0	0
1	0	1
1	1	0
0	1	1

$b_1 = m_1 \oplus m_2$

or $\underbrace{m_1 \oplus m_2 \oplus b_1 = 0}$

algebraic relation between parity bit and information bits

- can detect 1 error (e.g $110 \to 100$)
  can't detect 2 errors (e.g $110 \to 000$)
- If there are odd no. errors (e.g. 3), can we detect ?
- For k-bit messages, 2^{k+1} codewords
  $\Rightarrow \frac{1}{2} \cdot 2^{k+1} = 2^k$ will have <u>odd parity</u>

Parity check excludes this half

000	100
011	111
101	001
110	010

$\to$ odd parity

$\to$ even parity

- $d \geq 2$ as required for single-error detection
- Each word has even no. of 1's

$\boxed{\text{min. distance} = \text{min wt. of any nonzero codeword}}$

$\Rightarrow$ min dist $= 2$

- <u>Advantage</u> : high efficiency $\frac{k}{k+1} \simeq 1$ if $k \gg 1$

<u>Disadvantage</u> : can't correct

- <u>Repetition code</u> : poor efficiency $\left(\frac{1}{n}\right)$
  but can correct upto $\left[\frac{n-1}{2}\right]$ errors
- Want a reasonable combination of the good properties of the two types of codes
  $\Rightarrow [n, k]$ codes