Overview

- typical errors occur at bit or byte level during some sort of transmission
  - communication between computers
  - from memory to CPU
  - from CPU to i/o device
- protecting against such errors involves use of error-detecting or error-correcting codes
- define these codes as addition of specially computed bits for transmission and then computation performed over bits of received message
- result considered “coding” and “decoding”
- originally related to work of Claude Shannon & Information Theory for communication but applies equally to transmission of “information” within computers
example of error-detection vs error-correction given in following email:

- “Meet me in Manhattan at the information desk at Senn Station on July 43. I will arrive at noon on the train from Philadelphia.”

- coding theory requires use of redundancy in the form of added bits which compensate for “noise” in the transmission

  - in memory, address buses, transmission lines typically single bits

  - in disk memories often sequences of bits – “burst errors”

- examples of sources of errors:
  - component failure
  - damage to equipment
  - “cross-talk”
  - lightning
  - power disturbances
  - radiation effects
  - electromagnetic fields
  - electrical noise

- focus of text is what to do to protect against such effects assuming that they do occur

- interesting examples of such errors in airline industry:
- cross-talk between on-board systems
- passenger devices (e.g. cell phones!)
- external signals (military radar)
- lightning
- equipment malfunction

**Terminology**

- **code** – means of representing information or data using self-defined set of rules

- **code word** – collection of symbols used to represent particular piece of data based on specified code

- **binary code** – symbols are binary digits 0 and 1

- **valid** - codeword that follows all rules defining the code, otherwise **invalid**

- **encoding** – process of determining corresponding codeword for particular data item

- **decoding** – process of recovering original data from codeword

- **error-detecting code** – errors in codeword will make it become an invalid codeword & hence will be detected

- **single-error correcting code** – can correct single bit errors

- **Hamming distance** – number of bit positions in which two binary words differ
- **distance** – minimum Hamming distance between any 2 valid codewords – useful for example if distance is 2, then any single error in valid codeword creates a non-codeword; “cube” useful to represent 3-bit words

- **separable code** – check bits appended to information bits, rather than interspersed, thus decoding is simply removing appended bits

**Parity-bit Codes**

- simplest of codes other than one-bit codes with replication

- parity bits added to detect errors when distance is sufficient to produce invalid codewords on specified numbers of errors

- either even-parity or odd-parity

- when used for memory, one parity bit added on write to memory, and then removed on read

- obviously need extra hardware

- Table 2.2

- Figure 2.1 – what is error in first AND gate in part (b)?
  
  o if you receive 0001, can you correct it?

- Applications:
- transmissions over communication links (telephone lines, optical, microwave, satellite

- transmission of data to/from memory

- data exchange over buses within computer

- example: 7-bit ASCII characters with 1-bit parity (Fig. 2.2)

  - if info bits are (1 0 1 0 1 0 1) what is the parity bit?

- purpose of parity-bit checking is to detect errors, so how well this is done is measure of success, and of not doing so, a measure of failure

- $P_{ue}$ is probability of not detecting an error, i.e. no error detection used

- $P'_{ue}$ is the probability of undetected error when error detection used

- to illustrate these, consider an 8-bit byte & one bit of parity where undetected errors will be occurrence of 2, 4, 6 or 8 errors as the parity will be unchanged

- $B(r:n,q)$ is binomial probability of $r$ failures in $n$ occurrences with failure probability of $q$ which in this case is

  $$B(r: 9, q) = (9Cr) q^r(1-q)^{9-r}$$
- Note that for q relatively small \((10^{-4})\) only the case where \(r = 2\) is significant.

- thus \(P'_{ue}\) is \(B(2:9, q) = 36 q^2 (1-q)^7\)

- and \(P_{ue}\) is \(1 - P(0 \text{ errors}) = 1 - B(0: 8, q) = 1 - (1-q)^8\)

- then the ratio of these two gives the improvement ratio due to parity-bit coding

- these can be simplified by replacing \((1 - q)^n\) by \((1 - nq)\) and \((1/(1-q))\) by \((1 + q)\) Why?

- \(P_{ue}/P'_{ue}\) is given by \([2(1 + 7q)/ 9q]\)

- Figure 2.3 shows this ratio for various values, where today the value of q is getting close to \(10^{-7}\) an improvement from \(10^{-5}\) or \(10^{-6}\) in the 1960s and 1970s

- Figure 2.4 gives the actual logic diagram for a parity generator chip

  if the info bits are \((1 1 1 0 1 1 0 0)\) what is the parity bit?