CS305
Computer Architecture

Hamming Codes

Bhaskaran Raman
Room 406, KR Building
Department of CSE, IIT Bombay

http://www.cse.iitb.ac.in/~br

Need for Error Correcting Codes

- Reading and writing may be separated in time: storage systems
  - Main memory
  - Hard disk
- Reading and writing may be separated in space: communication
- Errors could be introduced in bits between writing & reading
- How to correct these errors? Error correcting codes.

Error Correcting Codes: Terminology

Data word
m

Code word
n

m or n
not necessarily
32 bits

- Redundant bits = ECC bits = (n-m) = k
- Notation for ECC: (n, m)
- Typically: code-word = superset of data-word

Data word	ECC bits
m	*k*

- Hamming codes: specific kind of ECC

Error Detection vs Error Correction

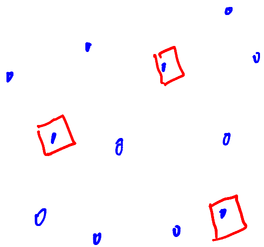
- Suppose data-word = 1 bit
- How many extra bits needed to detect 1-bit error?
- How many extra bits needed to correct 1-bit error?
- Suppose data-word = m bits
- How many extra bits needed to detect 1-bit error?
- How many extra bits needed to correct 1-bit error?
  - Answered through information theory analysis

1-bit Error Correction: Information Theory Analysis

Data word = m
Code word = n
n - m = k

Consider space of all possible code-words

Size of this space = 2^n



possible 0-bit error patterns = 2^m = valid code words

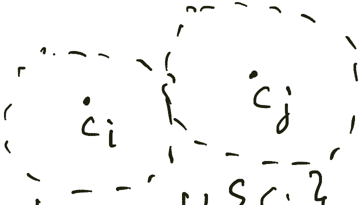
Distance betn 2 pts in this space = # bit flips req'd to go from one n-bit pattern to another = Hamming distance

H.D(0111011, 1011101) = 4

Nbrs of a node = Nodes at Hamming Distance = 1

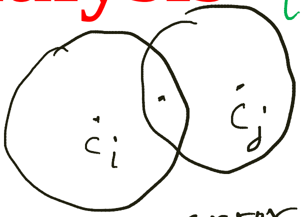
Code-words (valid): $c_0, c_1 \ldots c_{(2^m-1)}$

N_i = set of nbrs of c_i



$N_i' = N_i \cup \{c_i\}$
$N_j' = N_j \cup \{c_j\}$

$N_i' \cap N_j' = \phi$



1-bit error corr. is not possible

$N_i \cap N_j = \phi$

$|N_j'| = n+1$

$\bigcup_{i=0}^{2^m-1} N_i' \subseteq$ all poss n-bit pattern

$2^m \times (n+1) \le 2^n$

$m+k+1 \le 2^k$

$k_{opt} = O(\log_2 m)$

Hamming Code (n, 4)

$(4+k+1) \le 2^k$
$k \geq 3$
$n \geq 7$
$n = 7$

For m=4, $(4+k+1) <= 2^k$, satisfied for k >= 3

Hamming code uses min. reqd. k=3, i.e. (7, 4)

d7	d6	d5	p4	d3	p2	p1
7	6	5	4	3	2	1

7 = 111 b
6 = 110 b
5 = 101 b
4 = 100 b
3 = 011 b
2 = 010 b
1 = 001 b

Bit is set in representation of 7, 6, 5
➔ p4 = d7 + d6 + d5

Bit is set in representation of 7, 6, 3
➔ p2 = d7 + d6 + d3

Bit is set in representation of 7, 5, 3
➔ p1 = d7 + d5 + d3

Error Correction in Hamming Code

Assuming at most 1-bit error

Compute p4, p2, p1 bits again

Add up bit positions of pi, which do not match with code-word read == position of bit error

Scheme works even if error is in pi bit

Hamming Code (7, 4): An Example

Given data word 0110, what is the code-word?

p4 = d7+d6+d5 = 0

p2 = d7+d6+d3 = 1

p1 = d7+d5+d3 = 1

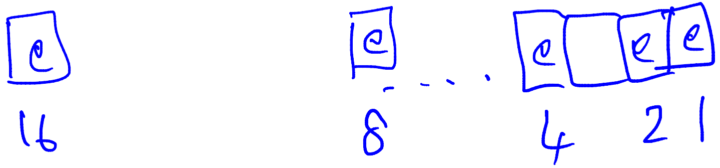
Code word = 0110011

Suppose code word read = 0100011

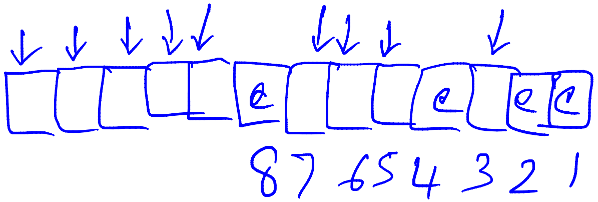
p4 = 1, p2 = 1, p1 = 0

Position of error = 4+1 = d5

Generic Hamming Code

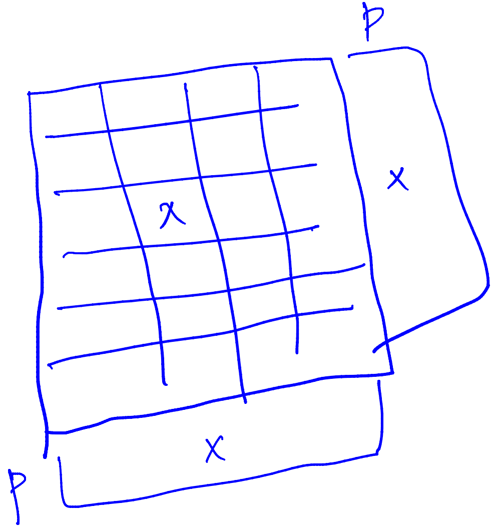


$m = 9$



$(9 + 4 + 1) \leq 2^4$

2-D Parity Scheme



(+) detect 2-bit errors
3-bit errors
detect most 4-bit errors

(−) more bits than Hamming code scheme

Summary

- Error correction: extra bits needed
  - Many more than for error detection
- Hamming code: $k = O(\log(m))$
- 2-D parity: $k = O(\mathrm{sqrt}(m))$