Your Name: _____ netid:_____ Group #:
Name: _____ netid:_____
Name: _____ netid:_____
Name: _____ netid:_____
Name: _____ netid:_____

ECE 198JL Worksheet 3: Error Detection and Correction

Please be familiar with the material on this page before you go to discussion section.

A brief review of part of Lecture Notes Set 1.5

In any digital system, errors may sometimes occur, in which some bits are altered. Any such error can have serious consequences. We can design representations, also called codes, in such a way that errors could be detected and even corrected.

One of the most convenient classes of error-correcting codes, commonly used in memory chips, was derived by R. W. Hamming and is known as **Hamming Codes**. An (**N**,**k**) Hamming code uses **N** bits to encode **k** data bits. The remaining **N-k** bits are parity bits that enable a system to determine the location of any single-bit error. In other words, any single bit error can be corrected.

The (7,4) Hamming code encodes a 4-bit message $\mathbf{d_3d_2d_1d_0}$ with three additional parity bits $\mathbf{p_4}$, $\mathbf{p_2}$, and $\mathbf{p_1}$, arranged as follows into a 7-bit code word:

code word: $\mathbf{x_7\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1}$

data / parity bits: $\mathbf{d_3\ d_2\ d_1\ p_4\ d_0\ p_2\ p_1}$

where

- p_1 is an **even parity bit** chosen such that $\mathbf{x_7}$ XOR $\mathbf{x_5}$ XOR $\mathbf{x_3}$ XOR $\mathbf{x_1} = 0$
- p_2 is an **even parity bit** chosen such that $\mathbf{x_7}$ XOR $\mathbf{x_6}$ XOR $\mathbf{x_3}$ XOR $\mathbf{x_2} = 0$
- p_4 is an **even parity bit** chosen such that $\mathbf{x_7}$ XOR $\mathbf{x_6}$ XOR $\mathbf{x_5}$ XOR $\mathbf{x_4} = 0$

When a bit pattern stored as a Hamming code is examined, we use check bits $\mathbf{c_4}$, $\mathbf{c_2}$, and $\mathbf{c_1}$ to determine whether the pattern has suffered an error. Each of these check bits is calculated using the formula for the corresponding parity bit. For example, $\mathbf{c_1} = \mathbf{x_7}$ XOR $\mathbf{x_5}$ XOR $\mathbf{x_3}$ XOR $\mathbf{x_1}$. In the absence of error, all of the check bits are 0, since the parity bits are chosen so as to make them so.

Let's use this encoding schema to help Prof. Lumetta send a message to Prof. Kindratenko ☺

Context for the discussion groups

Prof. Lumetta wants to be able to send to Prof. Kindratenko any of the sixteen 4-bit messages 0000, 0001, 0010, 0011, ... , 1111. Unfortunately, the transmission line is noisy, and Prof. Lumetta worries that a bit could get flipped during transmission. For example, if Prof. Lumetta sends 0010, Prof. Kindratenko might receive 1010, or he might receive 0110. It is also possible, although much less likely, that two or more bits in the message are changed. Prof. Lumetta decides to use the (7,4) Hamming code to encode the 4-bit messages.

1. Suppose that Prof. Lumetta wants to send the message 1101 and hence transmits 1100110.
    a. Give an example illustrating that the (7,4) Hamming code can be used to detect 2-bit errors.

    b. Give an example illustrating that the (7,4) Hamming code can not detect some 3-bit errors.

    c. Give an example illustrating that the (7,4) Hamming code can not correct 2-bit errors.

2. Any 1-bit error can be corrected when using the Hamming code. Suppose that Prof. Kindratenko receives 0110001.
    a. Give the check bits $c_4c_2c_1$.

    b. Which Hamming code bit position is wrong?

    c. What codeword did Prof. Lumetta (most likely) send?

3. Suppose that Prof. Kindratenko receives 0111111.
    a. Give the check bits $c_4c_2c_1$.

    b. What codeword does Prof. Kindratenko think Prof. Lumetta sent? What 4-bit message does Prof. Kindratenko think Prof. Lumetta sent?

    c. Suppose that Prof. Lumetta actually sent 0110011. Why did the check bits not convey this information to Prof. Kindratenko?