IT 327: Lab 11
Error Detection and Coding: Parity, CRC & FEC

Background

Error detection and coding is pervasive in digital communications; it is very rare that digital information is transmitted without some of these protective measures, except in public broadcasting. For this reason, we want to play around with it more than just in class. The best way (presently) is to do this with software.

Objectives

1. To experiment with parity and multi-bit errors in parity.
2. To experiment with CRC and do some actual encoding.
3. To write a simple program for forward error correction coding and to experiment with it.

Equipment Used

This lab will not use any hardware; all experiments will be done using software simulation of circuits. If you prefer, you may do the schematic of the circuit on MultiSim and simulate it using that software.

Procedures

Be sure to take the necessary screenshots and include them in your write-up with an appropriate caption.

Part 1:

a) Write a program (in any language, such as C, C++, Java, or PHP) that generates a parity bit (even or odd - you choose) for each byte of data and verify its correct operation.

b) Write another program to detect correct data based on the parity you have defined. Then use the program to generate a byte of data along with the parity bit (9-bits total), and observe what happens when there are 0, 1, 2, or 3 bits in error in a given byte. Include a screenshot of the results (command line or other) and discuss the probability of an undetected error occurring.

Part 2:

a) Write a program that generates an 8-bit CRC checksum for a data stream. An example of this program follows, in pseudocode. This same program may be used to check correct reception of the data stream. The simplest data stream for which this may be used is a single byte; simulate at least 5 different 1-byte data streams both with and without errors and log the results. Particularly interesting are error data streams which differ by only 1 bit from the correct data.

b) Also, simulate at least one 32-byte data stream with at least one error and log the results. Include a screenshot of the results and discuss the probability of an undetected error occurring.

Pass-off

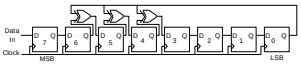
There is no pass-off required for this lab, but your write-up should show the results of your simulations.

Conclusions

Discuss the things you learned about error coding and correction (ECC), and what was the most meaningful (or least meaningful). Discuss what you have learned in this lab with respect to applications of ECC with which you are familiar.

CRC Program Pseudocode

Initialize counter and data stream
Loop for each bit in data stream
Define next states for each bit
Next Bit 1 = next bit of Data In
Next Bit 2 = Present Bit 1
Next Bit 3 = Present Bit 2 XOR present Bit 8
Next Bit 4 = Present Bit 3 XOR present Bit 8
Next Bit 5 = Present Bit 4 XOR present Bit 8
Next Bit 6 = Present Bit 5
Next Bit 7 = Present Bit 6
Next Bit 8 = Present Bit 7
Move next states into present states
Present Bit 1 = Next Bit 1
Present Bit 2 = Next Bit 2
Present Bit 3 = Next Bit 3
Present Bit 4 = Next Bit 4
Present Bit 5 = Next Bit 5
Present Bit 6 = Next Bit 6
Present Bit 7 = Next Bit 7
Present Bit 8 = Next Bit 8
Output CRC encoder byte to screen
End loop
Output final CRC checksum byte



8-Bit CRC Encoder