FPGA Project

Error Correction using (7,4,3)-Hamming Code

Sanjay and Zohan

March 8, 2019

Introduction

Encoding

Decoding

Implementation

Trial on Arduino

Introduction

Error Correction is used to correct errors caused by the channel.

(7,4,3)-Hamming Code encodes a 4-bit message onto 7 bits. The hamming distance of this code is 3.

Thus, it can correct one error.

Encoding

If the input code-word is: $(\mathrm{m}_1, m_2, m_3, m_4)$, Then the encoded code-word $(\mathrm{c}_1, c_2, c_3, c_4, c_5, c_6, c_7)$, is as follows:

$$c_1 = m_1$$

$$c_2 = m_2$$

$$c_3 = m_3$$

$$c_4 = m_4$$

$$c_5 = m_1 + m_2 + m_4$$

$$c_6 = m_1 + m_3 + m_4$$

$$c_7 = m_2 + m_3 + m_4$$

Where all elements come from the Binary Field. Thus, $+$ indicates XOR.

Decoding

If the received code-word is $(y_1, y_2, y_3, y_4, y_5, y_6, y_7)$, then the error code-word$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$, is calculated using the syndrome vector, which is calculated by multiplying the received code-word vector with the transpose of the parity check matrix(H), of (7,4,3)-Hamming Code.

$$H_{(7,4,3)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

The syndrome vector is a 3 bit vector. It has 8 possible values, where each value corresponds to one error vector of hamming weight one.

The error vector when added(XOR) to the received code-word gives us the corrected code-word.

Implementation

First, we generate input bit stream using Python. This input bit stream is sent to the FPGA via Arduino, where encoding takes place. The encoded bit stream is sent back to the Computer via Arduino, and is stored in a file.

Now, randomized bit flips are introduced to the encoded code-words based on a given Probability of Bit Error (P_b) (which depends on SNR). This data is stored in a new file and is sent to the FPGA via Arduino for decoding.

After decoding, the decoded bit stream is sent back to the Computer via Arduino and is stored in another file. The Total Probability of Error (P_e) can be calculated using the input file and the output file from the decoded bit stream. We can find various values of (P_e) with respect to SNR.

Trial on Arduino

Before implementing Encoding and Decoding on FPGA, we would implement the same on an Arduino-UNO. This is because, it's easier to implement the whole algorithm and communicate in Arduino.