1. Consider the following (5, 2) ECC code which contains a mixture of parity and repetitive bits:

\[ M_1 M_2 P_1 P_2 P_3 \]

where the two repetitive bits are \( P_1 = M_1 \) and \( P_2 = M_2 \) whereas the parity bit is \( P_3 = M_1 \oplus M_2 \).

(a) Construct a table of all valid code-words.

(b) What is the minimum Hamming distance between the valid code-words? What is the maximum Hamming distance between the valid code-words? How many bit-error(s) can we detect? How many bit-error(s) can we correct?

(c) Another (5, 2) ECC code is constructed as follows:

\[ M_1 M_2 P_1 P_2 P_3 \]

where \( P_1 = M_1 \oplus M_2 \), \( P_2 = M_1 M_2 \), and \( P_3 = M_1 + M_2 \).

Again, construct a table of all valid code-words.

(d) What is the minimum Hamming distance for this ECC code? What is the maximum Hamming distance between the valid code-words? How many bit-error(s) can we detect? How many bit-error(s) can we correct?

(e) A purely repetitive (6, 2) ECC code is constructed by simply repeating each message bit twice.

Construct a table of all valid code-words for this (6, 2) repetitive code.

(f) What is the minimum Hamming distance for this ECC code? What is the maximum Hamming distance between the valid code-words? How many bit-error(s) can we detect? How many bit-error(s) can we correct?

(g) Among the 3 ECC codes explored in Part (a), (c) and (e), which one is the best? Justify briefly.
2. Consider the design of a Hamming-style (5, 3) parity-checking code. Each valid code-word has the following form:

\[ X_3X_2X_1P_2P_1 \]

where the 2 parity bits are \( P_2 = X_2 \oplus X_3 \) and \( P_1 = X_1 \oplus X_2 \).

(a) Construct a table of all valid code-words.

(b) What is the minimum Hamming distance between the valid code-words? What is the maximum Hamming distance for this code?

(c) How many bit error(s) can we detect in this case? How many bit error(s) can we correct?

(d) Can we improve the detection/correction capability by adding one additional parity bit as follows: \( P_3 = X_3 \oplus X_1 \)? Why or why not?

Due date: **Nov 30** in class