ECE/CS 4434/6434
Homework 5
Due Date: Friday, October 11, 11:59 PM

Reading: Lectures 9 – 11 — Chapter 3 of I. Koren + Chapter 2 of R. K. Iyer

Problem 1 (10 pts) – For each of the following codes, determine the code distance and the number of bit-errors they can detect or correct.

Part A (5 pts) Repetition code of length n = 4: C1 = {0000, 1111}

Code Distance = 4, Detect = 3, Correct = 2

Part B (5 pts) Single-bit even parity code: C2 = {000, 011, 101, 110}

Code Distance = 2, Detect = 1, Correct = 0

Problem 2 (15 pts) – Consider a (5, 4) cyclic code with the generator polynomial G(X) = X + 1. For each of the following parts, provide an answer in the space provided. Show your work and a justification for your answer to get partial credit.

Part A (5 pts) For data word $D_1(X)$ = (0110), we obtain the codeword $C_1(X)$ = **____01010_____**

$$g(x) = x + 1$$
$$d(x) = x^2 + x$$
$$c(x) = (x + 1)(x^2 + x)$$
$$c(x) = x^3 + (2\%2)x^2 + x$$
$$c(x) = x^3 + x$$
$$c(0110) = 01010$$

Part B (5 pts) The codeword $C_2(X) = X^4+X$ is received with no error, the corresponding data word was $D_2(X) = \boldsymbol{X^3 - X^2 + X = 01110}$ ____

$$G(X) = X + 1$$
$$C_2(X) = X^4 + X$$
$$C_2(X) = D_2(X)G(X)$$
$$D_2(X) = \frac{C_2(X)}{G(X)}$$
$$D_2(X) = \frac{(X^4 + X)}{X + 1}$$
$$D_2(X) = X^3 + \frac{(-x^3 + x)}{X + 1}$$

$$D_2(X) = X^3 - X^2 + \frac{(X^2 + X)}{X + 1}$$
$$D_2(X) = X^3 - X^2 + X$$

Part C (5 pts) The codeword $C_3(X) = X^4+X^3+X^2+X+1$ is received, is an error detected or not?

Check if remainder of $D_3(X) = 0$

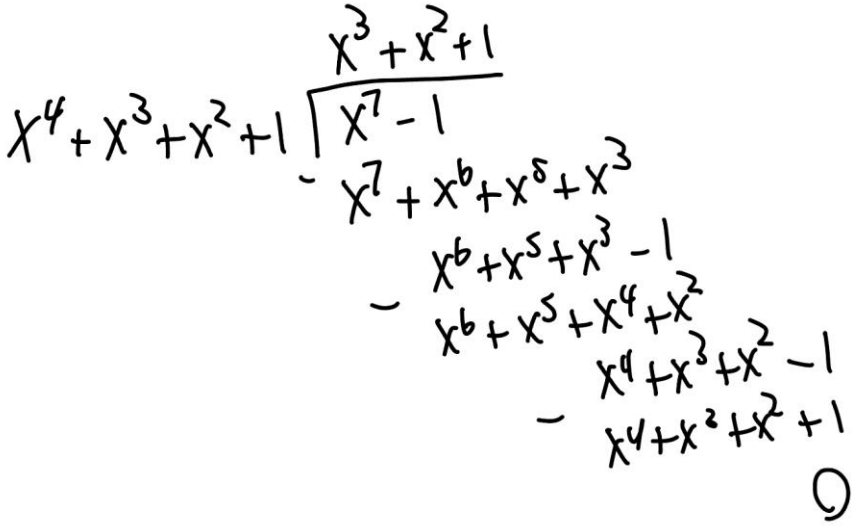
$$C_3(X) = D_3(X)G(X)$$
$$D_3(X) = \frac{C_3(X)}{G(X)}$$
$$D_3(X) = \frac{(X^4 + X^3 + X^2 + X + 1)}{X + 1}$$
$$D_3(X) = X^3 + \frac{X^2 + X + 1}{X + 1}$$
$$D_3(X) = X^3 + X + \frac{1}{X + 1}$$

Remainder not equal to zero, an error is detected.

Hint: Remember that a cyclic code (n, k) encodes data into codewords of length *n*, by multiplying *k*-bit data word by a *n-k* degree polynomial.

Problem 3 (30 pts) – Given that $X^7 - 1 = (X+1)\, g_1(X)\, g_2(X)$, where $g_1(X) = X^3+X+1$,

Part A (10 pts) Calculate $g_2(X)$.



$$g_2(X) = X^3 + X^2 + 1$$

Part B (10 pts) Identify all the (7, k) cyclic codes that can be generated based on the factors of $X^7 - 1$. How many different such cyclic codes exists?

$$G(X) = (X + 1)$$
$$G(X) = X^3 + X + 1$$
$$G(X) = X^3 + X^2 + 1$$
$$G(X) = (X + 1)(X^3 + X + 1)$$
$$G(X) = (X + 1)(X^3 + X^2 + 1)$$
$$G(X) = (X^3 + X + 1)(X^3 + X^2 + 1)$$
$$G(X) = X^7 - 1$$

Part C (10 pts) Show all the codewords generated by the polynomial $G(X)=(X+1)g_1(X)$ along with their corresponding data words. What is the distance for this code? What are the error detection capabilities of this code (how many errors it can detect or correct)?

$$G(X) = (X + 1)(X^3 + X + 1)$$
$$G(X) = X^4 + X^2 + X + X^3 + X + 1$$
$$G(X) = X^4 + X^3 + X^2 + 1$$

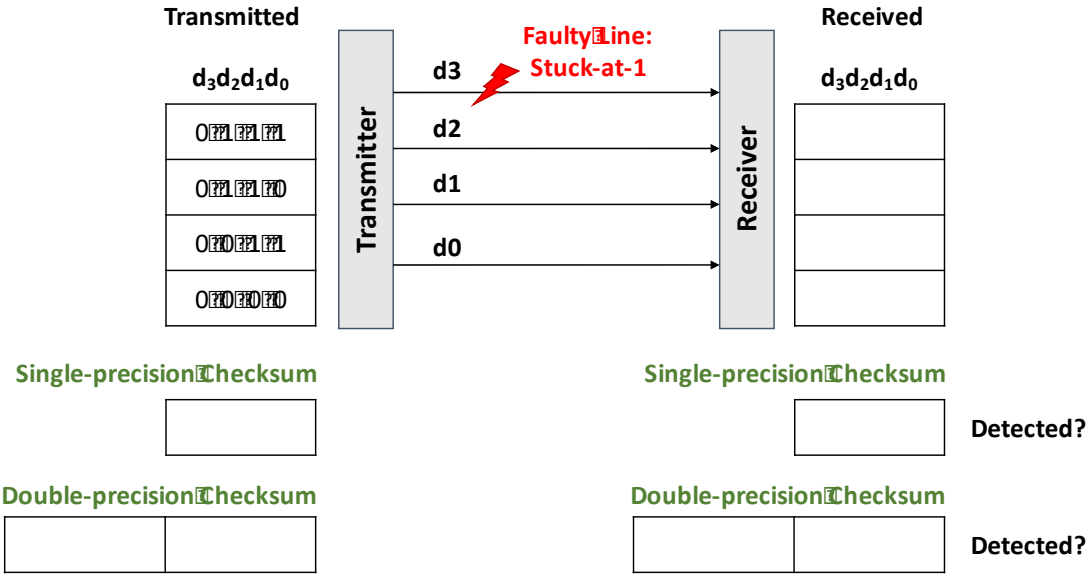
$$D(X) = 000, C(X) = 0 = 0000000$$
$$D(X) = 001, C(X) = X^4 + X^3 + X^2 + 1 = 00011101$$
$$D(X) = 010, C(X) = X^5 + X^4 + X^3 + X = 00111010$$
$$D(X) = 100, C(X) = X^6 + X^5 + X^4 + X^2 = 01110100$$
$$D(X) = 011, C(X) = (X^4 + X^3 + X^2 + 1)(X + 1) = X^5 + X^2 + X + 1 = 00100111$$
$$D(X) = 110, C(X) = (X^4 + X^3 + X^2 + 1)(X^2 + X) = X^6 + X^3 + X^2 + X = 01001110$$
$$D(X) = 101, C(X) = (X^4 + X^3 + X^2 + 1)(X^2 + 1) = X^6 + X^5 + X^3 + 1 = 01101001$$
$$D(X) = 111, C(X) = (X^4 + X^3 + X^2 + 1)(X^2 + X + 1) = X^6 + X^4 + X + 1 = 01010011$$

$$d_{min} = 4$$
$$detect = 3$$
$$correct = FLOOR\left(\frac{4 - 1}{2}\right) = 1$$

Hint: Note that subtraction in modulo-2 arithmetic is identical to addition, thus $X^n - 1$ is identical to $X^n + 1$.

Problem 4 (10 pts) – ***Checksums*** are commonly used in communication applications to detect errors in data transmission. The basic idea is to add up the block of data words to be transmitted and then transmit the sum (called the *checksum*) along with the data. The receiver adds up the data it receives and compares this sum with the checksum received. If the two do not match, an error is indicated. There are several variations of checksums. If the data words are ***d*** bits long, the *single-precision checksum* is calculated by performing **module-2^d** addition of the words, ignoring any overflows in addition. The *double-precision checksum* is calculated by **module-2^{2d} addition** of the data words.

The following figure shows a scenario where four 4-bit words (d_3,d_2,d_1,d_0) are transmitted over a communication channel. Assuming that the line carrying bit d_3 is faulty with a stuck-at-1 value, fill in the empty boxes by showing the received data words and calculating the single-precision and double-precision checksums for both transmitted and received data. Then for each case indicate whether the error will be detected or not.



Received (d3 Stuck at 1):

d3 d2 d1 d0
1 1 1 1
1 1 1 0
1 0 1 1
1 0 0 0

Single Precision Checksum Transmitted: 7+6+3+0 mod 16 = 16 mod 16 = 1 = 0001
Single Precision Checksum Received: 15 + 14 + 11 + 8 mod 16 = 48 mod 16 = 1 = 0001

1==1, No Error Detected for Single-Precision

Double Precision Checksum Transmitted: 7 + 6 + 3 + 0 mod 256 = 16 mod 256 = 16 = 00010000
Double Precision Checksum Received = 15 + 14 + 11 + 8 mod 256 = 48 mod 256 = 48 = 00110000

16 != 48, Error Detected for Double-Precision

Problem 5 (15 pts) – Remember that an (*n*, *k*) cyclic code can detect single bit errors and multiple adjacent bit errors affecting fewer than (*n* - *k*) bits. Show all your work for the following parts to get partial credit.

Part A (5 pts) Show that if the generating polynomial G(X) of a cyclic code has more than one term, all single-bit errors will be detected.

A single bit error at position i can be represented as X^i so a codeword can then be written as $R(X) = D(X)G(X) + X^i$. To detect an error the received codeword polynomial $R(X)$ is divided by $G(X)$ and if the remainder is 0 there is no error detected.

$$R(X) = D(X)G(X) + X^i$$
$$R(X) \div G(X) = (D(X)G(X) + X^i) \div G(X)$$
$$R(X) \div G(X) = \Big(\big(D(X)G(X) \div G(X)\big) + \big(X^i \div G(X)\big)\Big)$$
$$R(X) \div G(X) = \Big(D(X) + \big(X^i \div G(X)\big)\Big)$$

It is impossible for X^i to be evenly divided by $G(X)$ as $G(X)$ has more than 1 term therefore there will not be and will never be a zero remainder. This is because no polynomial with more than one term can divide a single term monomial without a remainder causing all single bit errors to be detected.

Part B (5 pts) Show that if G(X) has a factor with three terms, all *adjacent* double-bit errors will be detected.

Any adjacent double bit error can be represented as $X^i + X^{i+1}$ meaning a received codeword can be written as $R(X) = D(X)G(X) + X^i + X^{i+1}$. To check if $G(X)$ could detect an error in this scenario we would divide $R(X)$ by $G(X)$ to see if it could detect the error.

$$R(X) \div G(X) = (D(X)G(X) + X^i + X^{i+1}) \div G(X)$$
$$R(X) \div G(X) = \Big(\big(D(X)G(X) \div G(X)\big) + \big(X^i + X^{i+1} \div G(X)\big)\Big)$$
$$R(X) \div G(X) = \Big(D(X) + \big(X^i + X^{i+1} \div G(X)\big)\Big)$$

It is impossible for $X^i + X^{i+1}$ to be evenly divided by G(X) as G(X) is a three term polynomial and $X^i + X^{i+1}$ is a two term polynomial meaning there is no way for G(X) to divide with a non-zero remainder.

Part C (5 pts) Show that if G(X) has X + 1 as a factor, all odd numbers of bit errors will be detected. That is, if encoded word contains an odd number of terms (errors), it does not have X + 1 as a factor.
Hint: Note that a single bit error in bit position *i* can be represented by X^i and the received codeword with such an error can be written as D(x)G(x) + X^i.

Any odd count of bit errors can be represented simply as $\boldsymbol{R(X) = D(X)G(X) + X^i + X^j}$ ... where there are an odd number of additional error terms. A unique property of $\boldsymbol{X + 1}$ as a factor of G(X) is that it always perfectly divides even polynomials with an even number of terms but does not cleanly divide polynomials with odd number of terms. This means that our odd count of bit errors represented as $\boldsymbol{X^i + X^j}$ ... would always be caught by the factor of G(X), $\boldsymbol{X + 1}$ as there would always be a remainder when dividing R(X) by G(X).

$$\boldsymbol{R(X) \div G(X) = (D(X)G(X) + X^i + X^j \ldots) \div G(X)}$$
$$\boldsymbol{R(X) \div G(X) = D(X) + \left(X^i + X^j \ldots \div G(X)\right)}$$

Because G(X) has X+1 as a factor and X+1 will never produce a zero remainder when dividing odd term-count polynomials, we know for sure that the $\boldsymbol{X^i + X^j \ldots \div G(X)}$ block of the above equation will not divide evenly and will produce a non-zero remainder signaling an error every time there are an odd number of error bits.

Problem 6 (20 pts) – A CRC code has the following generator polynomial:

$$g(x) = x^{16} + x^{15} + x^2 + 1$$

Calculate the probability of detecting 17-bit burst error patterns and 18-bit burst error patterns.

Hint: Represent the data arriving as a sum *T(x) = D(x)g(x) + e(x)*, where *D(x)* is the original bit string, *e(x)* is a burst error vector, and + (plus) denotes the XOR operator. The error vector has 1's in the bit positions that are in error. A single burst error is characterized by an initial 1, a mixture of 0's and 1's, and a final 1, with all other bits being 0.

Hint: If the remainder of *T(x) / g(x)* is zero, then no error is detected.

Degree of generator polynomial g(x) = 16 with an actual length of 17 (16-0). CRC codes can detect burst errors of length n <= k+1 where k = the degree of the generator polynomial. The +1 comes from the actual number of bits adding 1 to account for the 1's place, so 0-16 bits is 17 total bits. For this example, k = 16 (Because the degree of the generator polynomial is 16). This means that any burst error <=17 bits will always be detected.

This means that for 17-bit burst error patterns we have a 100% chance of detection as the length of the pattern is <=k+1. So P(Detecting 17-bit Burst Error) = 1

For the 18-bit burst error patterns we see a probability of detection that is slightly more nuanced.

$$\boldsymbol{R(X) = T(X)}$$
$$\boldsymbol{R(X) = D(X)g(X) + e(X)}$$
$$\boldsymbol{R(X) \div G(X) = D(X) + \big(e(X) \div g(X)\big)}$$
$$\boldsymbol{P\big(e(X) \div g(X)\big) = 0}$$

17-Bit: 0 extra bits, P(Detecting 17-bit Burst Error) = 1
18-Bit: 1 extra bit, P(Detecting 18-bit Burst Error) = $1 - 2^{-17} = 0.9999923706$

This is because there is a $\frac{1}{2^{17}}$ probability of the burst error perfectly creating the generator code where the lower 17 bits match the generator code. The probability is very low but still possible.