EC305 Problem Set 5

1. [Problem 2.13] Let $g(D) = D^L + g_{L-1}D^{L-1} + \cdots + g_1 D + 1$ (assume that $L > 0$). Let $z(D)$ be a nonzero polynomial with highest and lowest order terms of degree j and i, respectively; that is, $z(D) = D^j + z_{j-1}D^{j-1} + \cdots + D^i$ [or $z(d) = D^j$, if $i = j$]. Show that g(D) z(D) has at least two nonzero terms. Hint: Look at the coefficients of D^{L+j} and of D^i.

2. [Problem 2.14] Show that if $g(D)$ contains a factor $1 + D$, then all error sequences with an odd number of errors are detected.

3. [Problem 2.15] For a given generator polynomial $g(D)$ of degree L and given data length K, let $c^{(i)}(D) = c^{(i)}_{L-1}D^{L-1} + \cdots + c^{(i)}_1 D + c^{(i)}_0$ be the CRC resulting from the data string with a single 1 in position i [i.e., $s(d) = D^i$ for $0 \leq i \leq K-1$].

   (a) For an arbitrary data polynomial $s(D)$, show that the CRC polynomial is $c(D) = \sum_{i=0}^{K-1} s_i c^{(i)}(D)$ (using modulo 2 arithmetic).

   (b) Letting $c(D) = c_{L-1}D^{L-1} + \cdots + c_1 D + c_0$, show that

   $$c_j = \sum_{i=0}^{K-1} s_i c^{(i)}_j; \quad 0 \leq j < L.$$

   This shows that each c_j is a parity check and that a CRC code is a parity check code.

4. [Problem 2.16(a)] Consider a stop-and-wait ARQ strategy in which, rather than using a sequence number for successive packets, the sending DLC sends the number of times the given packet has been retransmitted. The number of times a packet is retransmitted is 0 the first time a packet is transmitted. The receiving DLC returns an ack or nak (without any request number) for each frame it receives. Show by example that this strategy does not work correctly no matter what rule the receiving DLC uses for accepting packets.

5. [Problem 2.20] Give an example in which go back n ARQ fails if the modulus m is equal to n.

6. Reading: Correctness of Stop and Wait. Page No.69-71

7. [Problem 2.30]

   (a) Apply the bit stuffing rule of Section 2.5.2 to the following frame:
   0110111110111111011111010111111111101111010

   (b) Suppose the following string of bits is received:
   0111111011111011001111100111110111110110001111101011110
   Remove the stuffed bits and show where the actual flags are.

8. [Problem 2.31] Suppose the bit stuffing rule of Section 2.5.2 is modified to stuff a 0 only after the appearance of 01^5 in the original data. Carefully describe how the destuffing rule at the receiver must be modified to work with this change. Show how your rule would destuff the following string:
   01101111101111110111110101111110
   If your destuffing rule is correct, you should remove only two 0's and find only one actual flag.

9. [Problem 2.32] Bit stuffing must avoid the appearance of 01^6 within the transmitted frame, so we accept as given that an original string 01^6 will always be considered into $01^5 01$. Use this, plus the necessity to destuff correctly at the receiver, to show that a 0 must always be stuffed after 01^5. *Hint*: Consider the data string $01^5 01 x_1 x_2 ...$. If a 0 is not stuffed after 01^5, the receiver cannot distinguish this from $01^6 x_1 x_2 ...$ after stuffing, so stuffing is required in this case. Extend this argument to $01^5 0^k x_1 x_2 ...$ for any k>1.

10. [Problem 2.33] Suppose that the string 0101 is used as the bit string to indicate the end of a frame and the bit stuffing rule is to insert a 0 after the appearance of 010 in the original data; thus, 010101 would be modified by stuffing to 01001001. In addition, if the frame proper ends in 01, a 0 would be stuffed after the first 0 in the actual terminating string 0101. Show how the string 11011010010101011101 would be modified by this rule. Describe the destuffing rule required at the receiver. How would the string 11010001001001100101 be destuffed?