Solutions to Assignment 4

Version 2 (May 10): Corrected division result for the example Question 8.

Question 1

The character 'e' has ASCII value 0x45 (binary 01000101) which has an odd number of '1' bits (3). For even parity the number of '1' bits has to be even so the parity bit must be set to '1'.

Question 2

The Hamming distance is the number of bits that differ between two codewords. 0x71 is binary 0111 0001 and 0x93 is binary 1001 0011. The XOR operation between the two codewords is 1110 0010 or 0xe2 which has 4 '1' bits. This means 4 bits differ and the Hamming distance is 4. Note that your calculator may be able to compute the XOR function.

Question 3

The codewords are:

```
000000
010101
101010
111111
```

(a) n, the number of bits per codeword, is 6 and k, the number of data bits per codeword, is 2 ($\log_2 4$).

(b) The code rate is $k/n = 2/6 = 1/3$.

(c) The minimum distance is found by finding the smallest Hamming distance between all possible pairs of codewords. The pairs and distances are:

```
word 1  word 2 distance
------  ------   -
000000  010101   3
000000  101010   3
000000  111111   6
010101  101010   6
010101  111111   3
101010  111111   3
```

The minimum of the distances computed above is 3 so this is the minimum distance of the code.

(d) A block code with minimum distance d is guaranteed to detect $d-1$ or fewer errors. This code will therefore always detect an error if the received codeword has 1 or 2 errors.

A block code with minimum distance d is guaranteed to correct $\lfloor \frac{d-1}{2} \rfloor$ errors. This code will correct $\lfloor \frac{3-1}{2} \rfloor = 1$ error. A minimum-distance decoder is guaranteed to be able to correct any received codeword from this code that has 1 error. If the received codeword has more errors then the decoder may or not correct the error (i.e. it may or may not "guess" the correct codeword).

(e) The codeword with minimum distance to 001010 is the codeword 101010 which is at a distance 1. The decoder would choose that as the codeword most likely to have been sent because that is the transmitted codeword that would require the minimum number of errors to produce the received codeword. If that was indeed the codeword that was sent then one bit error would have been corrected.

(f) The distance from the codeword 001001 to the possible transmitted codewords is 2, 3, 3, and 4. In this case there is only one closest codeword and the receiver would decide that the codeword 000000 had been transmitted, thus detecting and correcting two errors (the two '1' bits).

This is an example where the decoder can correct more errors than are guaranteed to be corrected. If there had been two valid codewords at the same distance to the received codeword then the decoder would not have been able to correct the error.

Question 4

For the first system:

$$E_b = PT_{bit} = 10^3 \times 10^{-6} = 10^{-3}$$

Joules/bit (Watt·seconds/bit)
while for the second system:

$$E_b = PT_{bit} = \frac{600}{1.2 \times 10^6} = 0.5 \times 10^{-3}$$

Joules/bit

The coding gain is the reduction in E_b to achieve the same error rate over the same channel. In this case the coding gain is $10 \log_{10}(1/0.5) = 3$ dB.

Question 5

The bit sequence 100010 can be represented as the polynomial:

$$1x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 0x^0 = x^5 + x$$

Question 6

The polynomial $x^3+x+1 = 1x^3+0x^2+1x^1+1x^0$ can be represented as the bit sequence 1011.

Multiplying the polynomial by x^3 results in $x^6 + x^4 + x^3$ which is 1011000.

Question 7

The generator polynomial $x^3 + 1$ it has 4 bits. The remainder will thus have at most 3 bits. The remainder after dividing by the generator polynomial is the CRC so the CRC will have 3 bits.

To compute the CRC we first extend the message by the number of bits in the CRC (3 in this case) producing 1001000. Then we divide by the generator polynomial. In this case the result is trivial:

```
           1000
       ---------
1001   | 1001000
         1001
         ----
            0000
```

The remainder, 000 is the CRC.

Question 8

Appending the CRC to the message produces the same message. The division produces a remainder of zero indicating there was no error.

Note that this is a trivial example. In most cases the CRC will not be zero. Here is a more realistic example using a message of 1010. We append three 0's and do the division:

```
           1011
       ---------
1001   | 1010000
         1001
         ----
          0110
          0000
          ----
           1100
           1001
           ----
            1010
            1001
            ----
             011
```

In this case the remainder, the CRC, is 011. Appending it to the message, we would transmit 1010011. At the receiver we would divide by the generator polynomial again and check the remainder:

```
           1011
       ---------
1001   | 1010011
         1001
         ----
          0110
          0000
          ----
           1101
           1001
           ----
            1001
            1001
            ----
             000
```

Since the remainder is 0 then no error was detected.