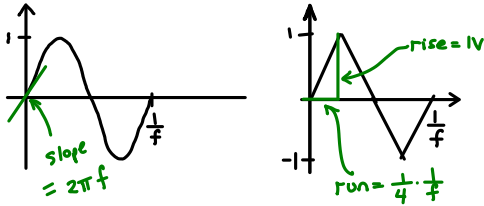


## Solutions to Assignment 5

Revised Dec. 30: bit period in question 5 is 100ns instead of 50ns.

### Question 1

The diagram below shows the two waveforms and their maximum slopes (slew rates):



(a) The slew rate is the slope of the voltage vs time curve. For a sine wave the slope is given by the derivative. The derivative of  $\sin(2\pi ft)$  is  $2\pi f \cos(2\pi ft)$  which has a maximum value, at  $t = 0$  of  $2\pi f$  (about  $6.3f$ ). The slope can also be obtained from the Taylor series expansion of  $\sin(2\pi ft)$  which has value  $2\pi f$  at  $t = 0$ .

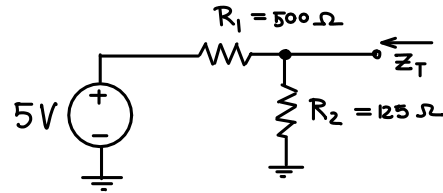
(b) A triangle wave with a peak value of 1 V and frequency  $f$  increases in voltage by 1 V in a time  $\frac{1}{4f}$ . Thus the slope is rise/run =  $4f$ .

### Question 2

The noise margin is the noise voltage required to cause an error. The decision threshold between the voltages would be set half-way between the levels. In the case of a system using voltage levels of +1, 0 and -1 volts, the thresholds would be at -0.5 and +0.5V. Thus a noise voltage of 0.5V could cause an error and this is the noise margin.

### Question 3

The following diagram shows one possible network:



The short-circuit current is given by  $5/R_1$  and the impedance seen by a source driving the termination would be the resistance of the two resistors in parallel  $(\frac{R_1 R_2}{R_1 + R_2})$ .

For a voltage of 5V and a current of 10 mA  $R_1 = 5/0.01 = 500\ \Omega$  and for a termination resistance of  $100\ \Omega$  we can solve for  $R_2 = 125\ \Omega$ .

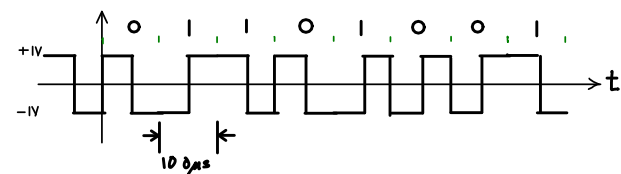
### Question 4

The HDLC frame begins and ends with a flag sequence (0111 1110). HDLC requires bit stuffing after 5 consecutive ones to ensure that data sequences equal to flag are not mistake for flags. Thus there will be two bit-stuffed zeros (after bits 4(5) and 9(10)). The sequence is thus:

0111 1110 1111 1011 1110 1111 0111 1110

### Question 5

The differential Manchester encoding for 0110 1001 is as follows:



### Question 6

A code uses the four codewords 10101010, 01010101, 11111111, and 00000000.

- (a) If there are four codewords then two bits ( $\log_2(4)$ ) can be transmitted per codeword.
- (b) Since each codeword has 8 bits,  $n = 8$  and since there are two data bits  $k = 2$ .
- (c) The code rate is  $\frac{k}{n} = \frac{2}{8} = \frac{1}{4}$ .
- (d) The minimum distance of the code can be found by finding the minimum Hamming distance between all pairs. These distances are (comparing 1-2, 1-3, 1-4, 2-3, etc): 8, 4, 4, 4, 4, 8. The minimum is 4 so this is the minimum distance of the code.
- (e) A code with a minimum distance of 4 can detect up to  $4-1=3$  errors.
- (f) A code with a minimum distance of 4 can correct  $\lfloor \frac{4-1}{2} \rfloor = 1$  errors.
- (g) (*labelled second e*) If the FEC decoder received the codeword 11101011 the Hamming distances to each valid codeword are 2, 6, 2, 6. Since none of these is zero, an error has been detected.
- (h) (*labelled second f*) The FEC decoder would choose the valid codeword that is at the smallest Hamming distance from the received codeword. Since there are two valid codewords with the same distance to the received codeword (2), the FEC decoder would have to pick one at random, either 11111111 or 10101010.
- (i) (*labelled g*) The “corrected” codeword would be either 10101010 or 11111111. In each case two errors (the second and eight bits, or the fourth and sixth bits) would have been corrected.

Note that the receiver doesn't know if the corrected codeword was the one actually transmitted. It's only choosing the codeword that was most likely transmitted.

---

### Question 7

---

The data sequence is 10110 ( $k = 5$ ). The generator polynomial is  $x^4 + 1$ . The CRC is the remainder after dividing by the generator polynomial  $G(x)$ . Since  $G(x)$  has 5 terms, the remainder (and the CRC) will

```

          10111
          -----
10001 | 101100000
        10001
        -----
          01110
          00000
          -----
            11100
            10001
            -----
              11010
              10001
              -----
                10110
                10001
                -----
                  0111

```

So the value of the remainder, the CRC, is 0111. Appending this value to the message polynomial (or equivalently, adding it to the zero-extended version) should result in a zero remainder. We can check by appending the CRC, dividing by  $G(x)$  and making sure the remainder is zero:

```

          10111
          -----
10001 | 101100111
        10001
        -----
          01110
          00000
          -----
            11101
            10001
            -----
              11001
              10001
              -----
                10001
                10001
                -----
                  0000

```