CHAPTER 10

PRACTICE SET

Questions

Q10-1. In a *single-bit error* only one bit of a data unit is corrupted; in a *burst error* more than one bit is corrupted (not necessarily contiguous).

Q10-3. In this case, k = 20, r = 5, and n = 20. Five redundant bits are added to the dataword to create the corresponding codeword.

Q10-5. The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.

Q10-7. We have $n = 2^r - 1 = 7$ and $k = n - 3 = 7 - 3 = 4$. A dataword has four bits and a codeword has seven bits. Although it is not asked in the question, we give the datawords and valid codewords below. Note that the minimum distance between the two valid codewords is 3.

Data	*Code*	*Data*	*Code*
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

Q10-9.

a. The generator has three bits (more than required). Both the rightmost bit and leftmost bits are 1s; it can detect all single-bit errors.

b. This cannot be used as a generator: the rightmost bit is 0.

c. This cannot be used as a generator; it has only one bit.

Q10-11. In this case $r = 7 - 1 = 6$.

a. The length of the error is L = 5, which means $L \le r$. All burst errors of this size will be detected.

b. The length of the error is L = 7, which means $L = r + 1$. This CRC will detect all burst errors of this size with the probability $1 - (0.5)^5 \approx 0.9688$. Almost 312 out of 10,000 errors of this length may be passed undetected.

c. The length of the error is L = 10, which means $L > r$. This CRC will detect all burst errors of this size with the probability $1 - (0.5)^6 \approx 0.9844$. Almost 156 out of 10,000 errors of this length may be passed undetected. Although the length of the burst error is increased, the probability of errors being passed undetected is decreased.

Q10-13. The value of a checksum can be all 0s (in binary). This happens when the value of the sum (after wrapping) becomes all 1s (in binary).

Q10-15. The following shows that L is the weighted sum of the data items.

Beginning	$R = 1$	$L = 0$
Iteration 1	$R = 1 + D_1$	$L = L + R = 0 + 1 + D1$
Iteration 2	$R = R + D_2 = 1 + D_1 + D_2$	$L = L + R = 2 + 2D_1 + D_2$
…	…	…
Iteration *n*	$R = 1 + D_1 + D_2 + \ldots + D_n$	$L = n + nD_1 + \ldots + D_n$

Problems

P10-1. We have (vulnerable bits) = (data rate) × (burst duration). The last example shows how a noise of small duration can affect a large number of bits if the data rate is high.

a.	vulnerable bits	=	$(1500) \times (2 \times 10^{-3})$	=	3 bits
b.	vulnerable bits	=	$(12 \times 10^{3}) \times (2 \times 10^{-3})$	=	24 bits
c.	vulnerable bits	=	$(100 \times 10^{3}) \times (2 \times 10^{-3})$	=	200 bits
d.	vulnerable bits	=	$(100 \times 10^{6}) \times (2 \times 10^{-3})$	=	200,000 bits

P10-3. The following shows the results. In the interpretation, 0 means a word of all 0 bits, 1 means a word of all 1 bits, and ~X means the complement of X.

a. (10001) ⊕ (10001) = (00000) **Interpretation: X ⊕ X → 0**

b. (11100) ⊕ (00000) = (11100) **Interpretation: X ⊕ 0 → X**

c. (10011) ⊕ (11111) = (01100) **Interpretation: X ⊕ 1 → ~X**

P10-5. Answers are given below:

a. error **b.** error **c.** 0000 **d.** 1101

P10-7. The following shows the result. Part d shows that the Hamming distance between a word and itself is 0.

a. d (10000, 00000) = 1 **b.** d (10101, 10000) = 2

c. d (00000, 11111) = 5 **d.** d (00000, 00000) = 0

P10-9. The CRC-8 is 9 bits long, which means $r = 8$.

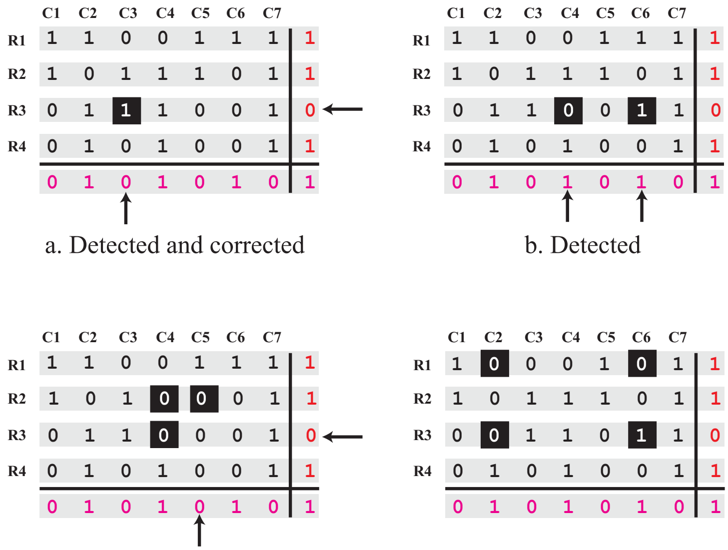
a. It has more than one bit and the rightmost and leftmost bits are 1s; it can detect a single-bit error.

b. Since $6 \leq 8$, a burst error of size 6 is detected.

c. Since $9 = 8 + 1$, a burst error of size 9 is detected most of the time; it may be left undetected with probability $(1/2)^{r-1}$ or $(1/2)^{8-1} \approx 0.008$.

d. Since $15 > 8 + 1$, a burst error of size 15 is detected most of the time; it may be left undetected with probability $(1/2)^{r}$ or $(1/2)^{8} \approx 0.004$.

P10-11. The following shows the errors and how they are detected.



a. Detected and corrected

b. Detected

c. Detected

d. Not detected

a. In the case of one error, it can be detected and corrected because the two affected parity bits can define where the error is.

b. Two errors can definitely be detected because they affect two bits of the column parity.The receiver knows that the message is somewhat corrupted (although not where). It discards the whole message.

c. Three errors are detected because they affect two parity bits, one of the column parity and one of the row parity. The receiver knows that the message is somewhat corrupted (although not where). It discards the whole message.

d. The last case cannot be detected because none of the parity bits are affected.

P10-13.

a. $(x^3 + x^2 + x + 1) + (x^4 + x^2 + x + 1) = x^4 + x^3$

b. $(x^3 + x^2 + x + 1) - (x^4 + x^2 + x + 1) = x^4 + x^3$

c. $(x^3 + x^2) \times (x^4 + x^2 + x + 1) = x^7 + x^6 + x^5 + x^2$

d. $(x^3 + x^2 + x + 1) / (x^2 + 1) = x + 1$ (remainder is 0)

P10-15. To detect single bit errors, a CRC generator must have at least two terms and the coefficient of x^0 must be nonzero.

a. $x^3 + x + 1 \rightarrow$ It meets both critic.

b. $x^4 + x^2$ $\rightarrow$ It meets the first criteria, but not the second.

c. $1 \rightarrow$ It meets the second criteria, but not the first.

d. $x^2 + 1 \rightarrow$ It meets both criteria.

P10-17. This generator is $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$.

a. It has more than one term and the coefficient of x^0 is 1. It detects all single-bit error.

b. The polynomial is of degree 32, which means that the number of checkbits (remainder) $r = 32$. It will detect all burst errors of size 32 or less.

c. Burst errors of size 33 are detected most of the time, but they are slip by with probability $(1/2)^{r-1}$ or $(1/2)^{32-1} \approx 465 \times 10^{-12}$. This means **465 out of 10^{12}** burst errors of size 33 are left undetected.

d. Burst errors of size 55 are detected most of the time, but they are slipped with probability $(1/2)^r$ or $(1/2)^{32} \approx 233 \times 10^{-12}$. This means **233 out of 10^{12}** burst errors of size 55 are left undetected.

P10-19. The following shows the steps:

a. We first add the numbers in two's complement to get 212,947.

b. We divide the above result by 65,536 (or 2^{16}). The quotient is 3 and the remainder is 16,339. The sum of the quotient and the remainder is 16,342.

c. Finally, we subtract the sum from 65,535 (or $2^{16} - 1$), simulating the complement operation, to get 49,193 as the checksum.

P10-21.

a. We calculate R and L values in each iteration of the loop and then concatenate L and R to get the checksum. All calculations are in hexadecimal and modulo 256 or $(FF)_{16}$. Note that R needs to be calculated before L in each iteration ($L = L_{previous} + R$).

Initial values:	`R = 00`	`L = 00`
Iteration 1:	`R = 00 + 2B = 2B`	`L = 00 + 2B = 2B`
Iteration 2:	`R = 2B + 3F = 6A`	`L = 2B + 6A = 95`
Iteration 3:	`R = 6A + 6A = D4`	`L = 95 + D4 = 69`
Iteration 4:	`R = D4 + AF = 83`	`L = 69 + 83 = EC`
Checksum = **`EC83`**		

b. The L and R values can be calculated as shown below (D_i is the corresponding bytes), which shows that L is the weighted sum of bytes.

$$R = D_1 + D_2 + D_3 + D_4 = 2B + 3F + 6A + AF = 83$$
$$L = 4 \times D_1 + 3 \times D_2 + 2 \times D_3 + 1 \times D_4 = EC$$

P10-23. We use modulo-11 calculation to find the check digit:

$$\mathbf{C} = (\mathbf{1} \times 0) + (\mathbf{2} \times 0) + (\mathbf{3} \times 7) + (\mathbf{4} \times 2) + (\mathbf{5} \times 9) + (\mathbf{6} \times 6) + (\mathbf{7} \times 7) + (\mathbf{8} \times 7) + (\mathbf{9} \times 5) \bmod 11 = 7$$

P10-25. The receiver misses samples 21, 23, 25, 27, 29, 31, 33, 35, 37, and 39. However, the even-numbered samples are received and played. There may be some glitches in the audio, but that passes immediately.

P10-27. The redundant bits in this case need to find $(n + 1)$ different states because the corruption can be in any of the n bits or in no bits (no corruption). A set of r bits can define 2^r states. This means that we need to have the following relationship: $2^r \geq n + 1$. We need to solve the equation for each value of k using trial and error to find the minimum value of r.

a. If $k = 1$, then $r = 2$ and $n = 3$ because ($2^2 \geq 3 + 1$), which means C(3, 1).

b. If $k = 2$, then $r = 3$ and $n = 5$ because ($2^3 \geq 5 + 1$), which means C(5, 1).

c. If $k = 5$, then $r = 4$ and $n = 9$ because ($2^4 \geq 9 + 1$), which means C(9, 5).

d. If $k = 50$, then $r = 6$ and $n = 56$ because ($2^6 \geq 56 + 1$), which means C(56, 50).

e. If $k = 1000$, then $r = 10$ and $n = 1010$ because $2^{10} \geq 1010 + 1$, which means C(1010, 1000).

P10-29. If we need to correct m bits in an n bit codeword, we need to think about the combination of n objects taking no object at a time or $Com(n, 0)$, which means the state of no error, the combination of n objects taking one object at a time or $Com(n, 1)$, which means the state of one-bit error, the combination of n objects taking two objects at a time or $Com(n, 2)$, which means the state of two-bit error, and so on. We can have the following relationship between the value of r (number of redundant bits) and the value of m (the number of errors) we need to correct.

$$2^r \geq Com(n, m) + Com(n, m-1) + \ldots + Com(n, 1) + Com(n, 0)$$