CHAPTER 9

Error Detection and Correction

9.1 REVIEW QUESTIONS

1. In a single bit error only one bit of a data unit is corrupted, i.e. changed from 1 to 0 or from 0 to 1, whereas the term burst error means more than one corrupted bit.
2. Redundancy is a technique of adding extra bits into each data unit for the purpose of determining the accuracy of transmission.
3. Vertical redundancy check, longitudinal redundancy check, cyclical redundancy check, and checksum.
4. A parity bit is added to every data unit so that the total number of 1s in the unit becomes even (in even-parity checking). If after transmission the number of 1s is odd, then there must be an error.
5. In even parity the number of 1s should be even; in odd parity the number of 1s should be odd.
6. VRC is the least expensive and most common type of error detection. It can detect single bit errors as well as burst errors if the total number of bits changed is odd.
7. Like VRC, LRC also uses the parity bit technique for error detection. The difference is in the way the parity bits are calculated. The original data bits are organized in a table of rows and columns. The parity bit is then calculated for each column.
8. An LRC of n bits can detect single bit errors as well as most burst errors of n bits except for damage to corresponding bits of different data units.
9. The CRC remainder is added to the data unit.
10. The length of the divisor should be one bit more than the length of the CRC.
11. The CRC checker divides the received data by the predetermined divisor and accepts the data if the remainder is zero.
12. A polynomial should not be divisible by x and should be divisible by (x+1).
13. CRC can detect all burst errors that affect an odd number of bits, all errors of length less than or equal to the degree of the polynomial, and most burst errors of length greater than the degree of the polynomial.

14. Checksum.
15. One's complement arithmetic.
16.
    a. The checksum generator divides the data into equal segments.
    b. The segments are added together using one's complement arithmetic.
    c. The sum is complemented and sent to the receiver along with the data.
17. The checker divides the received data into the same number of segments. The segments are added using one's complement to get the sum. The checker then complements the sum. If the result is 0 the data are accepted, otherwise the data are discarded.
18. If a bit inversion in one data segment is balanced by an opposite bit inversion on the corresponding bit of another segment, the error can't be detected by checksum.
19. $2^r \geq m + r + 1$, where r is the number of redundancy bits, and m is the number of data bits.
20. The purpose of the Hamming code is to correct one or more corrupted bits.

9.2 MULTIPLE CHOICE QUESTIONS

21. b	22. b	23. d	24. a	25. c	26. b	27. c	28. a	29. d	30. c
31. b	32. a	33. b	34. a	35. d	36. b	37. c	38. d	39. d	40. d
41. b	42. a	43. d	44. d						

9.3 EXERCISES

45.
    a. $1{,}500 \times 2 \times 0.001 = 3$ bits /data unit
    b. $12{,}000 \times 2 \times 0.001 = 24$ bits/data unit
    c. $96{,}000 \times 2 \times 0.001 = 192$ bits/data unit
46.
    a. 0
    b. 0
    c. 1
    d. 0
47. Yes, because the number of 1s is odd.
48. 11110110
49. 001
50. No error
51. 1101010000011110

52. 0001101110001100
53. No error
54.
    a. 5
    b. 5
    c. 5
    d. 7
55. $r_1 = 1$
    $r_2 = 1$
    $r_4 = 1$
    $r_8 = 0$
    The code is 100101101111
56. 00111010 11001111 11111111 00000000 00001010
57. The error cannot be detected because the number of corrupted bits is even.
58. Bits 1, 5, 7, and 8 (from the right) are in error.
59. 8 bits (not considering VRC bits). The ratio is 8 to 9
60. 5 bits long
61. 100001011
62. $x^{12} + x^6 + x^5 + x^4 + 1$
63. The 5th bit is in error. The correct code is 11001110111.
64. Two redundancy bits, one data bit.
65. Four redundancy bits, six data bits.
66. No errors in the received code; the original code is the same as the received code.