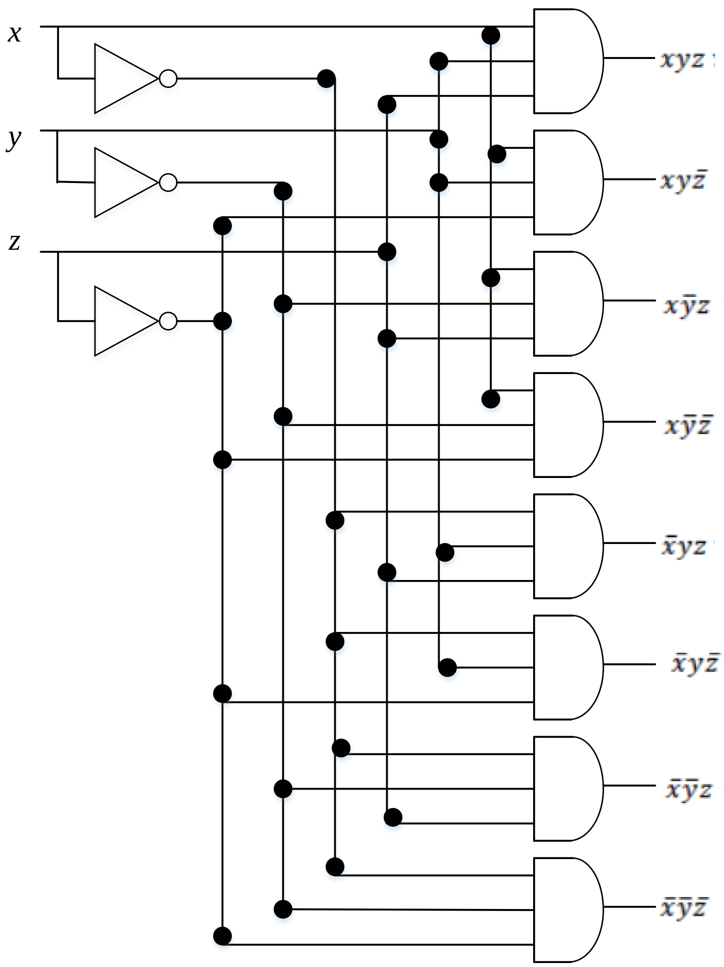

CHAPTER 2

Exercise 1

The decimal integers for all 3-bit binary integers are given below.

3-bit Binary Integer	Decimal Integer
000	0
001	1
010	2
011	3

100	4
101	5
110	6
111	7

Exercise 2

The decimal value of 111011.01111 is given below.

$$1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5}$$

$$= 32 + 16 + 8 + 0 + 2 + 1 + 0 + 0.25 + 0.125 + 0.0625 + 0.03125$$

$$= 59.46875$$

Exercise 3

$12.34 - 2^3 = 4.34$

$4.34 - 2^2 = 0.34$

$0.34 - 2^{-2} = 0.09$

The binary number is 1100.01.

Exercise 4

$12 \div 2 = 6$ with the reminder 0

$6 \div 2 = 3$ with the reminder 0

$3 \div 2 = 1$ with the reminder 1

$1 \div 2 = 0$ with the reminder 1

The binary integer is 1100.

$0.34 \times 2 = 0.68$

$0.68 \times 2 = 1.36.$

The binary number is 0.01. Hence, the entire binary value is 1100.01.

Exercise 5

The signed magnitude representation of -12 is 10001100.

The signed magnitude representation of 12 is 00001100.

Exercise 6

One’s complement representation of -12 is 11110011.

One’s complement representation of 12 is 00001100.

Exercise 7

Two’s complement representation of -12 is 11110100.

Two’s complement representation of 12 is 00001100.

Exercise 8

00010111 – 00001001 is performed using one’s complement method as follows.

    00010111

  + 11110110

= 100001101

The end carry-around is performed as follows.

  00001101

+   1

= 00001110

The result is 00001110 or 14.

Exercise 9

00001001 – 00010111 is performed using one’s complement method as follows.

  00001001

 + 11101000

= 11110001

There is no end carry-around. The result is 11110001 or – 14.

Exercise 10

00010111 – 00001001 is performed using two’s complement method as follows.

  00010111

 + 11110111

= 100001110

After discarding the carry-out, the result is 00001110 or 14.

Exercise 11

00001001 – 00010111 is performed using two’s complement method as follows.

  00001001

 + 11101001

= 11110010.

The result is 11110010 or – 14.

Exercise 12

$m + k + 1 \leq 2^k$

$8 + k + 1 \leq 2^k$

$k + 9 \leq 2^k$

k can be any integer equal to or greater than 4. Let *k* = 4, that is, we can use 4 check bits.

Exercise 13

Based on the result from Exercise 12, we use 4 check bits. Hence, there are 12 bits including 8 data bits and 4 check bits. The Hamming algorithm is carried out as follows.

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Value												

The check bits are at bit positions 1, 2, 4, and 8. We write down each bit position as the sum of numbers that are powers of 2 as follows.

1 = 1

2 = 2

3 = 1 + 2

4 = 4

5 = 1 +4

6 = 2 + 4

7 = 1 + 2 + 4

8 = 8

9 = 1 + 8

10 = 2 + 8

11 = 1 + 2 + 8

12 = 4 + 8

Hence, the check bit at position 1 contributes to bit positions 1, 3, 5, 7, 9, 11. The check bit at position 2 contributes to bit positions 2, 3, 6, 7, 10, 11. The check bit at position 4 contributes to bit positions 4, 5, 6, 7, 12. The check bit at position 8 contributes to bit positions 8, 9, 10, 11, 12. We fill in the data bits 01001011 into bit positions 11, 10, 9, 7, 6 5 and 3 with We now compute the value of each check bit using the even parity.

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Value	0	1	0	0	1	1	0	1	0	1	1	0

The Hamming code is 010011010110.

Exercise 14

For the Hamming code 010111010110 based on the even parity, we find:

- 1 bit error at bit positions 1, 3, 5, 7, 9, 11 using check bit at bit position 1
- No bit error at bit positions 2, 3, 6, 7, 10, 11 using check bit at bit position 2
- No bit error at bit positions 4, 5, 6, 7, 12 using check bit at bit position 4

- 1 bit error at bit positions 8, 9, 10, 11, 12 using check bit at bit position 8.

Hence, one bit error occurs at the bit position 1 + 9 = 9. The Hamming code is corrected to 010011010110.

CHAPTER 3

The round-robin is commonly used to schedule processes/threads on computers.

Pros:

- It gives each process/thread a turn to use CPU with an equal amount of time so that no processes/threads wait too long to be served.
- It is easily implemented.

Cons:

- More processes/threads will result in more waiting time of each process/thread before its turn of using CPU.
- No priority of processes/threads is considered.
- The equal share of CPU time can be exploited by denial of service attacks.

Exercise 2

If a data file is stored entirely on a single block of memory or storage space, the block of memory or storage space left after a data file is deleted may not be usable for another data file of a different size, resulting in unused memory or storage space. Breaking a data file into pages of standard size maximized the use of memory or storage space by allowing a page of any data file