

|  |  |
| --- | --- |
| Name | Aryan Dilipbhai Langhanoja |
| Roll No | 92200133030 |

# Problems

### Outline the different steps involved in addition of larger bit binary numbers for the following two cases:

**(a) The larger of the two numbers is positive**

### and the other number is negative. (b) The larger of the two numbers is negative and the other number is positive.

**Ans:-**

1. Addition of a positive number and a negative number:
   * Determine the larger of the two numbers by comparing their magnitudes.
   * Take the 2's complement of the negative number to obtain its additive inverse.
   * Add the larger number and the 2's complement of the negative number using standard binary addition rules.
   * If there is a carry-out from the most significant bit (MSB), discard it, as it represents an overflow.
   * If the result is positive, it is the correct answer. If the result is negative, take its 2's complement to obtain the correct answer.
2. Addition of a negative number and a positive number:
   * Determine the larger of the two numbers by comparing their magnitudes.
   * Take the 2's complement of the negative number to obtain its absolute value.
   * Subtract the larger number from the 2's complement of the negative number using standard binary subtraction rules.
   * If there is a borrow from the MSB, it means that the result is negative, and the answer is the 2's complement of the result.
   * If there is no borrow from the MSB, it means that the result is positive, and it is the correct answer.

### Outline the different steps involved in sub- traction of larger bit binary numbers for the following two cases:

* 1. **The minuend is positive. The subtrahend is negative and smaller in magnitude.**

### The minuend is positive. The subtrahend is negative and larger in magnitude.

**Ans:-**

1. Subtraction of a negative number smaller in magnitude from a positive number:
   * Take the 2's complement of the negative number to obtain its absolute value.
   * Add the absolute value of the negative number to the positive number using standard binary addition rules.
   * If there is a carry-out from the MSB, discard it, as it represents an overflow.
   * The result is positive, and it is the correct answer.
2. Subtraction of a negative number larger in magnitude from a positive number:
   * Take the 2's complement of the negative number to obtain its absolute value.
   * Add the absolute value of the negative number to the positive number using standard binary addition rules.
   * If there is a carry-out from the MSB, it represents an overflow, and the result is negative.
   * If there is no carry-out from the MSB, the result is positive, and it is the correct answer. However, to express it in two's complement notation, take the 2's complement of the result to obtain the negative of the result.

### What decides whether a particular binary addition or subtraction operation would be possible with 2's complement arithmetic?

**Ans:-**

In 2's complement arithmetic, whether a particular binary addition or subtraction operation is possible or not depends on the bit-width of the operands and the resulting sum or difference.

For example, if two 8-bit signed numbers are added using 2's complement arithmetic, the resulting sum should also be an 8-bit signed number. If the sum exceeds the range of 8-bit signed numbers, it is considered overflow, and the result is considered invalid. Similarly, in subtraction, the difference must also fall within the same range as the operands.

It is important to ensure that the bit-width of the operands and the resulting sum or difference are compatible in 2's complement arithmetic to ensure that the operation can be carried out correctly.

### Why in microprocessors and microcomputers is the 'repeated add and right-shift' algorithm preferred over the 'repeated left-shift an add' algorithm for binary multiplication? Briefly outline the procedure for mul tiplication in case of the former.

**Ans:-**

* because it requires fewer operations and is faster.
* The procedure for multiplication using the 'repeated add and right-shift' algorithm is as follows:

1. Load the multiplicand and multiplier into two registers.
2. Initialize a third register to zero, which will hold the product.
3. Check the value of the least significant bit (LSB) of the multiplier. If it is 1, add the multiplicand to the product register.
4. Right-shift the multiplier by 1 bit, discarding the LSB.
5. Left-shift the multiplicand by 1 bit.
6. Repeat steps 3-5 until the multiplier is zero.
7. The final value in the product register is the result of the multiplication.

### 5. The result of adding two BCD numbers rep- resented in the excess-3 code is 0111 1011when the two numbers are added using simple binary addition. If one of the numbers a (12)10, find the other.

**Ans:-**

To convert it back to BCD, we can subtract 0011 from each 4-bit segment, since excess-3 is obtained by adding 0011 to each BCD digit. This gives us:

0111 - 0011 = 0100 (represents digit 4 in BCD)

1011 - 0011 = 1000 (represents digit 8 in BCD)

So the excess-3 code 0111 1011 represents the BCD number (48)10.

To find the other BCD number, we can subtract the known BCD number (12)10 from (48)10:

48 - 12 = 36

Therefore, the other BCD number is (36)10, which in excess-3 code is represented as:

0011 0101

To verify, we can add the excess-3 codes of (12)10 and (36)10 using simple binary addition:

0010 0100 (represents 12 in excess-3 code)

0011 0101

# Objective’s

1. (6)8 + (7)8 yields

a (13)

b. (15) c.(0110111

d. None of these

### Ans:-B(15)

2. (7)8+(1)2 yields

(8)10

b. (8) c. (1111)

d. None of these

### Ans:-B(8)

3.(1)2+(1)2+ (1)2 + (1)2 would yield

a (1)2

b. (100)

c. (1111)2

d. None of these

### Ans:-b(100)

4. (2)8 + (2)8+ (2)8equals

a. (6)8

b. (6)10

c. 6H

d. (a), (b), and (c)

### Ans:-a (6)8

6. (5)8 × (3)8equals

a. (15)

b. (17)8

c. (111)2

d. (b) and (c)

### Ans:-B(17)

7. The excess-3 BCD code for (1111)2 is a. 0100 1000

b. (10010)

c. 0001 0101

d. None of these

### Ans:- C

8. (1111)2X (3)8 equals

a. (45)10

b. (100111)2

c. (55)8

d. (a) and (c)

### Ans:-A

1. The 16-bit excess-3 BCD equivalent of 5 is a.0000000000001000

b.0000000000000101 c. 0011001100111000

d. 0011001100110101

**Ans:- A**

Fill in The Blanks

* 1. Adding 2H to FEH produces **100H.**
  2. Subtracting (3)g from (100) yields **(97)g.**
  3. 110101 is a **necessarily** binary number**.**
  4. In 2's complement addition or subtraction, the result is in **2's complement form.**
  5. Binary multiplication in computers is a process of **shift** and **add operations.**

## True- False

1. Adding 2's complement of a binary number X to another binary number Y achieves Y- X. **Ans:- False**
2. Adding 2's complement of a binary number X to 2's complement of another binary number Y achieves X + Y. **Ans:- True**
3. The addition of -32 and -17 is possible using six-bit 2's complement arithmetic. **Ans:- False**
4. While 1101.1011 is a fixed-point binary representation, 11011011 x 21 is a float- ing-point representation of the same number. **Ans:- False**
5. BCD numbers can be conveniently added using 2's complement notation.

### Ans:- False