Assignment - 1

1. (10 marks) If an unsigned decimal integer of n-digit is converted into binary number representation, how many bits are there in the binary representation?

Ans.) Let the n digit decimal number be N, then the number of bits in the binary

representation will be : ë(log2(N) û + 1

Let the binary representation of N = (Bn-1Bn-2.....B2B1B0)2

For n bit representation,

⇒ 2n-1≤ N = 2n-1. Bn-1 + 2n-2 . Bn-2 + ... + 22 . B2 + 21 . B1 + 20 . B0 ≤ 2n-1 + 2n-2 + ... + 20

⇒ 2n-1 ≤ N ≤ 2n - 1

⇒ 2n-1 ≤ N < 2n ( Since, N ≤ 2n - 1 ⇒ N < 2n  )

⇒ n-1 ≤ log2 (N) < n ( Since log2N lies between n and n-1, floor(log2N) = n-1)

⇒ ë n-1 û = ë log2 (N) û

⇒ n - 1 = ë log2(N) û ( floor(n-1) = n-1 since n is a natural number)

⇒ n = 1 + ë log2(N) û

2. (15 marks) Convert decimal number(50.6875)10 to binary representation.

Ans.) XI = 50 and XF  = 0.6875

For the integral part, we have the following:

|  |  |  |
| --- | --- | --- |
| Integer: Decimal-to-binary | | |
| Dividing-by-rd | Quotient | Remainder |
| 50/2 | 25 | 0 |
| 25/2 | 12 | 1 |
| 12/2 | 6 | 0 |
| 6/2 | 3 | 0 |
| 3/2 | 1 | 1 |
| 1/2 | 0 | 1 |

For the fractional part, it follows:

|  |  |  |
| --- | --- | --- |
| Fractional: Decimal-to-binary | | |
| Multiplying-by-rd | Fractional part | Integral part |
| 0.6875 x 2 | 0.375 | 1 = x-1 |
| 0.375 x 2 | 0.75 | 0 = x-2 |
| 0.75 x 2 | 0.5 | 1 = x-3 |
| 0.5 x 2 | 0 | 1 = x-4 |

∴, (50.6875)10 = (11 0010 . 1011)2

3. (15 marks) Represent decimal numbers (50.6875)10 and (−50.6875)10 in sign-magnitude method for k= 8 and m= 4, where k and m indicate the number of integer bits (including sign bit) and fraction bits in the representation, respectively

Ans) Sign-magnitude representations of the numbers (50.6875)10 and (-50.6875)10

can be found out as follows:

Firstly, finding out the magnitudes of integral part and the fractional part:

Here, XI = 50 and XF  = 0.6875

For the integral part, we have the following:

|  |  |  |
| --- | --- | --- |
| Integer: Decimal-to-binary | | |
| Dividing-by-rd | Quotient | Remainder |
| 50/2 | 25 | 0 |
| 25/2 | 12 | 1 |
| 12/2 | 6 | 0 |
| 6/2 | 3 | 0 |
| 3/2 | 1 | 1 |
| 1/2 | 0 | 1 |

For the fractional part, it follows:

|  |  |  |
| --- | --- | --- |
| Fractional: Decimal-to-binary | | |
| Multiplying-by-rd | Fractional part | Integral part |
| 0.6875 x 2 | 0.375 | 1 = x-1 |
| 0.375 x 2 | 0.75 | 0 = x-2 |
| 0.75 x 2 | 0.5 | 1 = x-3 |
| 0.5 x 2 | 0 | 1 = x-4 |

∴, (50.6875)10 = (110010.1011)2

For sign magnitude form, with k=8 and m=4, to get the positive value, we must concatenate “0” with the binary representation i.e. X+ve = 0.XM where Xm is the magnitude of the number. Similarly, to get the negative value, we must concatenate “1” with the binary representation i.e. X-ve = 1.XM where again the Xm is the magnitude of the number. Hence, (50.6875)10 = “0” + “0110010.1011”

Hence,

(50.6875)10 = (0011 0010 . 1011)2  in fixed point sign-magnitude form

(-50.6875)10 = (1011 0010 . 1011)2 in fixed point sing-magnitude form

4. (15 marks) Represent decimal numbers (50.6875)10 and (−50.6875)10 in biased (with Bias= 128) method for k= 8 and m= 4, where k and m indicate the number of integer bits and fraction bits in the representation, respectively.

Ans) Biased representation of the numbers (50.6875)10 and (-50.6875)10 can be obtained as follows: For number 50.6875, XI = 50 and XF  = 0.6875

For the integral part, we have the following:

|  |  |  |
| --- | --- | --- |
| Integer: Decimal-to-binary | | |
| Dividing-by-rd | Quotient | Remainder |
| 50/2 | 25 | 0 |
| 25/2 | 12 | 1 |
| 12/2 | 6 | 0 |
| 6/2 | 3 | 0 |
| 3/2 | 1 | 1 |
| 1/2 | 0 | 1 |

For the fractional part, it follows:

|  |  |  |
| --- | --- | --- |
| Fractional: Decimal-to-binary | | |
| Multiplying-by-rd | Fractional part | Integral part |
| 0.6875 x 2 | 0.375 | 1 = x-1 |
| 0.375 x 2 | 0.75 | 0 = x-2 |
| 0.75 x 2 | 0.5 | 1 = x-3 |
| 0.5 x 2 | 0 | 1 = x-4 |

∴, (50.6875)10 = (11 0010 . 1011)2

Now, to obtain the biased representation or excess-bias representation, we must add the bias=128 to the binary representation:

(50.6875)10 = Binary representation of (128 + (50.6875))

= Binary representation of (178.6875)

= (1011 0010 . 1011)2  in fixed point biased representation

(-50.6875)10 = Binary representation of ( 128 + (-50.6875)) =

= Binary representation of ( 77.3125)

= (0100 1101 . 0101)2 in fixed point biased representation

5. (15 marks) Represent decimal numbers (50.6875)10 and (−50.6875)10 in binary 2’s complement representations for k= 8 and m= 4, where k and m indicate the number of integer and fraction bits in the representation, respectively.

Ans.) 2's complement representation of the numbers (50.6875)10 and (-50.6875)10

can be obtained as follows.

Firstly, finding out the magnitudes of integral part and the fractional part:

Here, XI = 50 and XF  = 0.6875

For the integral part, we have the following:

|  |  |  |
| --- | --- | --- |
| Integer: Decimal-to-binary | | |
| Dividing-by-rd | Quotient | Remainder |
| 50/2 | 25 | 0 |
| 25/2 | 12 | 1 |
| 12/2 | 6 | 0 |
| 6/2 | 3 | 0 |
| 3/2 | 1 | 1 |
| 1/2 | 0 | 1 |

For the fractional part, it follows:

|  |  |  |
| --- | --- | --- |
| Fractional: Decimal-to-binary | | |
| Multiplying-by-rd | Fractional part | Integral part |
| 0.6875 x 2 | 0.375 | 1 = x-1 |
| 0.375 x 2 | 0.75 | 0 = x-2 |
| 0.75 x 2 | 0.5 | 1 = x-3 |
| 0.5 x 2 | 0 | 1 = x-4 |

∴, (50.6875)10 = (11 0010 . 1011)2

As per the definition:

For a given number, a method to find its complement representations is:

If the given number has a positive value (including zero), simply add zero(s) to

its most significant end so that the representation has k digits in length. Otherwise, when the given number has a negative value, say, −Y , then the complement representation is R − Y .

∴, (50.6875)10 = (0011 0010 . 1011)2  in fixed point 2's complement representation

For 2's complement R = 2k  = 28 = (1 0000 0000)2  and Y = 50.6875,

∴, (-50.6875)10 = (1 0000 0000)2 - (0011 0010 . 1011)2

= (1100 1101 . 0101)2 in fixed point 2's complement representation

or (-50.6875)10 = 2's complement of binary representation of (50.6875)10

= 2's complement of (0011 0010 . 1011)2

= 1's complement of (0011 0010 . 1011)2 + ulp

= (1100 1101 . 0100)2  + (0.0001)2

= (1100 1101 . 0101)2

6. (15 marks) Represent decimal numbers (50.6875)10  and (−50.6875)10 in binary 1’s complement representations for k=8 and m=4, where k and m indicate the number of integer and fraction bits in the representation, respectively

Ans.)

1's complement representation of the numbers (50.6875)10  and (-50.6875)10

can be obtained as follows:

Firstly, finding out the magnitudes of integral part and the fractional part:

Here, XI = 50 and XF  = 0.6875

For the integral part, we have the following:

|  |  |  |
| --- | --- | --- |
| Integer: Decimal-to-binary | | |
| Dividing-by-rd | Quotient | Remainder |
| 50/2 | 25 | 0 |
| 25/2 | 12 | 1 |
| 12/2 | 6 | 0 |
| 6/2 | 3 | 0 |
| 3/2 | 1 | 1 |
| 1/2 | 0 | 1 |

For the fractional part, it follows:

|  |  |  |
| --- | --- | --- |
| Fractional: Decimal-to-binary | | |
| Multiplying-by-rd | Fractional part | Integral part |
| 0.6875 x 2 | 0.375 | 1 = x-1 |
| 0.375 x 2 | 0.75 | 0 = x-2 |
| 0.75 x 2 | 0.5 | 1 = x-3 |
| 0.5 x 2 | 0 | 1 = x-4 |

∴, (50.6875)10 = (11 0010 . 1011)2

As per the definition:

For a given number, a method to find its complement representations is:

If the given number has a positive value (including zero), simply add zero(s) to

its most significant end so that the representation has k digits in length. Otherwise, when the given number has a negative value, say, −Y , then the complement representation is R − Y .

∴, (50.6875)10 = (0011 0010 . 1011)2 in fixed point 1's complement representation

For 1's complement R = 2k - 1 = 28 - 1= (1 0000 0000)2 - (0 0000 0001)

= (1111 1111), and Y = 50.6875,

∴, (-50.6875)10 = (1111 1111)2 - (0011 0010 . 1011)2

= (1100 1101 . 0100)2 in fixed point 1's complement representation

(-50.6875)10 = 1's complement of (00110010.1011)2

= (11001101.0100)2

7. (15 marks) Let A= 11001, B= 01100, and C= 10100 be three 2’s complement numbers in 5 bits. Perform 2’s complement operations A+B and A+C to obtain the results. Verify to check whether or not there is an overflow. Provide a method and show how to obtain the correct result with your method if there is an overflow.

Ans ) A = 11001 = -7

B = 01100 = 12

---------

~~1~~ 00101 = +5 => The result is correct if we ignore the carry

Also, no overflow occurs, since magnitude of

result i.e 5 < 16 (2n-1)

A = 11001 = -7

C = 10100 = -12

---------

~~1~~01101 = -19 => Since, the magnitude of the result is

19 > 2n-1 = 16, overflow occurs

If the number of bits in the results Z are to be same as the number of bits in the operands A and B then, there is no way to obtain the correct result.

One solution could be to flag the overflow as an error but that would still require using an extra bit to hold the flag value. To obtain the flag, let As, Bs and Zs represent the sign bit of the operands A and B and the result Z. Then,

Flag = As' . Bs' . Zs + As . Bs . Zs' (where X' represents the complement)

i.e. overflow only occurs when the operands have same sign and the sign of the result is different from the sign of the operands. Overflow does not occur when the operands have different signs. So, for the addition A + C

A = 11001 ⇒ As = 1

+ C = 10100 ⇒ Bs  = 1

--------------

Z = 01101 ⇒ Zs = 0

∴, Flag = As' . Bs' . Zs + As . Bs . Zs' = (1' . 1' . 0 ) + (1 . 1 . 0') = 1

This flag would indicate that an overflow has occured and that the correct result is actually n+1 bits wide. Here, n+1 = 5+1 = 6 bits wide