Introduction to Computing

CPIT 201 - Homework 2 - Due **September 19**

1. Store **22** in an 8-bit memory location using unsigned representation? Show your work?

|  |  |
| --- | --- |
| Change the number from Decimal to Binary | (22)10 ------------> (10110)2 |
| Add three bits to the left to make it 8 bits | 0 0 0 1 0 1 1 0 |

1. Store **-31** in an 8-bit memory location using sign-and-magnitude representation? Show your work?

|  |  |
| --- | --- |
| Change the number from Decimal to Binary | (-31)10 ------------> (11111)2 |
| Store -31 in 7 bits | 0 0 1 1 1 1 1 |
| Add the sign | 1 0 0 1 1 1 1 1 |

1. Store **-36** in an 8-bit memory location using two’s complement representation? Show your work?

|  |  |
| --- | --- |
| Change the number from Decimal to Binary | (-36)10 ------------> (100100)2 |
| Store -36 in 8 bits | 0 0 1 0 0 1 0 0 |
| Apply two's complement operation | 0 0 1 0 0 1 0 0  1 1 0 1 1 1 0 0 |

1. Retrieve the integer that stored as 11100111 in memory in unsigned format? Show your work?

|  |  |
| --- | --- |
| Convert the number from Binary to Decimal | (11100111)₂ = (1 × 2⁷) + (1 × 2⁶) + (1 × 2⁵) + (0 × 2⁴) + (0 × 2³) + (1 × 2²) + (1 × 2¹) + (1 × 2⁰) = (231)₁₀ |

1. Retrieve the integer that stored as 11100111 in memory in sign-and-magnitude format? Show your work?

|  |  |
| --- | --- |
| The leftmost bit is 1 so the number in Decimal is negative. | 1 1 1 0 0 1 1 1 1 |
| Convert the number from Binary to Decimal without the leftmost number. | (11001111)₂ = (1 × 2⁷) + (1 × 2⁶) + (0 × 2⁵) + (0 × 2⁴) + (1 × 2³) + (1 × 2²) + (1 × 2¹) + (1 × 2⁰) = (207)₁₀ |
| Add the negative sign | (-207)10 |

1. Retrieve the integer that stored as 11010101 in memory in two’s complement format? Show your work?

|  |  |
| --- | --- |
| The leftmost bit is 1 so the number in Decimal is negative. | 1 1 0 1 0 1 0 1 |
| Apply two's complement operation | 1 1 0 1 0 1 0 1  0 0 1 0 1 0 1 1 |
| Convert the number from Binary to Decimal | (00101011)₂ = (0 × 2⁷) + (0 × 2⁶) + (1 × 2⁵) + (0 × 2⁴) + (1 × 2³) + (0 × 2²) + (1 × 2¹) + (1 × 2⁰) = (43)₁₀ |
| Add the negative sign | (-43)10 |

1. Using floating-point representation (IEEE\_127) (single precision), show the process how (-**36.36**) is represented?

|  |  |  |  |
| --- | --- | --- | --- |
| The number is negative. | | S=1 | |
| Convert the number from Decimal to Binary | | (100100.0101)2 | |
| Normalization (100100.0101)2 | | (1.001000101)2 × 25 | |
| S | E | | M |
| 1 | 5+127=132  10000100 | | 00100010100000000000000 |

1. The bit pattern (010001110 00000000010110101010000)2 is stored in IEEE\_127 format. Show the value in decimal? Show your work?

|  |  |  |  |
| --- | --- | --- | --- |
| S | E | | M |
| 0 | 10001110  142-127=15 | | 00000000010110101010000 |
| (1.00000000010110101010000)2 × 215 | | | |
| Convert the number from Binary to Decimal | | (1000000000101101.01010000)2 = (32813.80)10 | |

1. Use the **last digit** of your student number (6) as **6**-bit memory location in **two’s complement**
2. How many different patterns (symbols) can be represented using **6** bits?

62=36

1. show the range for all numbers that can be represented (minimum and maximum values)?

Maximum = (26-1)-1 = 31

Minimum = - (26-1) = -32

1. Show the binary representation of the **first two numbers** and the **last two numbers**?

|  |  |
| --- | --- |
| (31) 10 = (011111)2 | (30) 10 = (011110)2 |
| (-32)10 is negative, take the two’s complement and then stores it. | 1 0 0 0 0 0  1 0 0 0 0 0 |
| (-31) 10 is negative, take the two’s complement and then stores it. | 0 1 1 1 0 0  1 0 0 1 0 0 |

1. Discuss when an overflow would happen?

Occurs when adding two positive numbers produces a negative result, or when adding two negative numbers produces a positive result. Adding operands of unlike signs never produces an overflow. For example:

01010000 = 80

+ 01010000 = 80

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

10100000 = −96 (not 160 because the sign bit is 1.)