John Robertson

jprob

1)

a.

22 = 00010110

5 = 00000101

00010110

+00000101

00011011 = 1 + 2 + 8 + 16 = 27

b.

15 = 1111

14 = 1110

1111

+1110

***1***1101 = 1 + 4 + 8 + 16 = 29

Overflow of 16. If w4 were dumped, the result would be 13.

c.

12 = 1100

8 = 1000

1100

+1000

***1***0100 = 4 + 16 = 20

Overflow of 16. If w4 were dumped, the result would be 4.

2)

a.

7 = 0111

4 = 0100

As Two’s Complement numbers, 0111 remains as is, and 0100 is flipped to 1011 to become Ones’ Complement before one is added, yielding 1100.

0111

+1100

**1**0011

The extra bit is ignored, so the result is 2 + 1 = 3.

b.

19 = 00010011

11 = 00001011

00001011 becomes 11110100 then 11110101.

00010011

+11110101

**1**00001000

Ignoring the bit yields 8.

c.

3 = 0011

8 = 1000

1000 is -8 in Two’s Complement as it is.

0011

+1000

1011 = 1 + 2 – 8 = -5

3)

a.

11 = 01011

-4 = 00100 then 11011 then 11100

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | Q | Q-1 | M | Count |
| 00000 | 01011 | 0 | 11100 | 5 |

Q0 \* Q-1 is 10; A – M = (00000 + 00100) = 00100; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00010 | 00101 | 1 | 11100 | 4 |

Q0 \* Q-1 is 11; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00001 | 00010 | 1 | 11100 | 3 |

Q0 \* Q-1 is 01; A + M = (00001 + 11100) = 11101; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11110 | 10001 | 0 | 11100 | 2 |

Q0 \* Q-1 is 10; A – M = (11110 + 00100) = **1**00010 (ignore the last bit); shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00001 | 01000 | 1 | 11100 | 1 |

Q0 \* Q-1 is 01; A + M = (00001 + 11100) = 11101; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11110 | 10100 | 0 | 11100 | 0 |

Result is 1111010100 = -512 + 256 + 128 + 64 + 16 + 4 = -44

b.

-7 = 0111 then 1000 then 1001

3 = 0011 (negative is 1101)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | Q | Q-1 | M | Count |
| 0000 | 1001 | 0 | 0011 | 4 |

Q0 \* Q-1 is 10; A – M = (0000 + 1101) = 1101; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1110 | 1100 | 1 | 0011 | 3 |

Q0 \* Q-1 is 01; A + M = (1110 + 0011) = **1**0001; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 | 1110 | 0 | 0011 | 2 |

Q0 \* Q-1 is 00; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0000 | 0111 | 0 | 0011 | 1 |

Q0 \* Q-1 is 10; A – M = (0000 + 1101) = 1101; shift Q and A right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1110 | 1011 | 1 | 0011 | 0 |

Result is 11101011 = -128 + 64 + 32 + 8 + 2 + 1 = -21

4)

a)

16 = 010000 (negative = 110000)

5 = 000101

|  |  |  |  |
| --- | --- | --- | --- |
| A | Q | M | Count |
| 0000000 | 000101 | 010000 | 6 |

Shift left A and Q; A – M = (000000 + 110000) = 110000; A is less than zero; A + M = (110000 + 010000) = **1**000000

|  |  |  |  |
| --- | --- | --- | --- |
| 1000000 | 001010 | 010000 | 5 |

Shift left A and Q; A – M = (000000 + 110000) = 110000; A is less than zero; A + M = (110000 + 010000) = **1**000000

|  |  |  |  |
| --- | --- | --- | --- |
| 1000000 | 010100 | 010000 | 4 |

Shift left A and Q; A – M = (000000 + 110000) = 110000; A is less than zero; A + M = (110000 + 010000) = **1**000000

|  |  |  |  |
| --- | --- | --- | --- |
| 000000 | 101000 | 010000 | 3 |

Shift left A and Q; A – M = (000001 + 110000) = 110001; A is less than zero; A + M = (110001 + 010000) = **1**000001

|  |  |  |  |
| --- | --- | --- | --- |
| 000001 | 010000 | 010000 | 2 |

Shift left A and Q; A – M = (000010 + 110000) = 110010; A is less than zero; A + M = (110010 + 010000) = **1**000010

|  |  |  |  |
| --- | --- | --- | --- |
| 000010 | 100000 | 010000 | 1 |

Shift left A and Q; A – M = (000101 + 110000) = 110011; A is less than zero; A + M = (110101 + 010000) = **1**000101

|  |  |  |  |
| --- | --- | --- | --- |
| 000101 | 000000 | 010000 | 0 |

5)

a)

125.125 can be split into two parts: before and after the radix point. The before equals 64 + 32 + 16 + 8 + 4 + 1, or 01111101. The latter equals 1/8, or 0.001. These make 01111101.001. The mantissa is 6 digits wide (not counting the zero pad or the leading one), so the bias (for single-precision) is 127 + 6, or 133. The number is positive, so the sign bit is zero. The exp is 133, or 128 + 4 + 1, or 10000101. The fractional bit is 10 digits long and needs to be padded with 13 zeroes.

The single-precision floating-point representation of the number is:

0 10000101 1111010010000000000000