1. Hex FAC3 in binary is:
    - F A C 3
    - 15 10 12 3
    - 1111 1010 1100 0011
    - 1111101011000011
2. Hex FAC3 as an unsigned decimal is:
    - 15 x 16^3 + 10 x 16^2 + 12 * 16^1 + 3 *16^0
    - 61,440 + 2560 + 192 + 3 = 64195
3. Hex FAC3 as a signed decimal is:
    - 1111101011000011
    - 0000010100111100 + 1 = 0000010100111101
    - In decimal = 2^10 + 2^8 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0 = 1341
    - Because F = 1111 = Most significant digit is 1, therefore signed as negative, so answer is -1341
4. Hex 0064 in binary is:
    - 0 0 6 4
    - 0000 0000 0110 0100
    - 0000000001100100
5. Hex 0064 as an unsigned decimal is:
    - 6 x 16 + 4 = 100
6. Hex 0064 as a signed decimal is:
    - Also 100, because the leftmost hex digit is 0, implying non-negative sign
7. Hex 8000 in binary is:
    - 8 0 0 0
    - 8 0 0 0
    - 8421 8421 8421 8421
    - 1000 0000 0000 0000
    - 1000000000000000
8. Hex 8000 as an unsigned decimal is:
    - 8 * 16^3 = 32,768
9. Hex 8000 as a signed decimal is:
    - 8 0 0 0
    - 1000 0000 0000 0000
    - 0111111111111111 + 1 = 1000000000000000 = -32,768
10. Decimal 8000 encoded in 16-bits (unsigned) is in hex:
    - 8000 % 2 = 0, 8000 // 2 = 4000
    - 4000 % 2 = 0, 4000 // 2 = 2000
    - 2000 % 2 = 0, 2000 // 2 = 1000

- 1000 % 2 = 0, 1000 // 2 = 500
- 500 % 2 = 0, 500 // 2 = 250
- 250 % 2 = 0, 250 // 2 = 125
- 125 % 2 = 1, 125 // 2 = 62
- 62 % 2 = 0, 62 // 2 = 31
- 31 % 2 = 1, 31 // 2 = 15
- 15 % 2 = 1, 15 // 2 = 7
- 7 % 2 = 1, 7 // 2 = 3
- 3 % 2 = 1, 3 // 2 = 1
- 1 % 2 = 1, 1 // 2 = 0
- 0001 1111 0100 0000
- 0x1F40

11. Decimal 8000 encoded in 16-bits (signed) is in hex:
    - 0x1F40 (no change from previous answer)
12. Decimal -11 encoded in 16-bits (signed) is in hex:
    - 11 % 2 = 1, 11 // 2 = 5
    - 5 % 2 = 1, 5 // 2 = 2
    - 2 % 2 = 0, 2 // 2 = 1
    - 1 % 2 = 1, 1 // 2 = 0
    - 0000 0000 0000 1011
    - 1111 1111 1111 0100
    - 1111 1111 1111 0101
    - 0xFFF5
13. Decimal -32717 encoded in 16-bits (signed) is in hex:
    - 32717 % 2 = 1, 32717 // 2 = 16358
    - 16358 % 2 = 0, 16358 // 2 = 8179
    - 8179 % 2 = 1, 8179 // 2 = 4089
    - 4089 % 2 = 1, 4089 // 2 = 2044
    - 2044 % 2 = 0, 2044 // 2 = 1022
    - 1022 % 2 = 0, 1022 // 2 = 511
    - 511 % 2 = 1, 511 // 2 = 255
    - 255 % 2 = 1, 255 // 2 = 127
    - 127 % 2 = 1, 127 // 2 = 63
    - 63 % 2 = 1, 63 // 2 = 31
    - 31 % 2 = 1, 31 // 2 = 15
    - 15 % 2 = 1, 15 // 2 = 7
    - 7 % 2 = 1, 7 // 2 = 3

- 3 % 2 = 1, 3 // 2 = 1
- 1 % 2 = 1, 1 // 2 = 0
- 0111 1111 1100 1101
- 1000 0000 0011 0010
- 1000 0000 0011 0011
- 0x8033

14. Binary 10111101 in hex is:
    - 1011 1101
    - 0xBD
15. Binary 1011110100000001 as an unsigned decimal is:
    - 1011 1101 0000 0001
    - $(11 * 2^{12}) + (13 * 2^{8}) + 0 + 1 = 48385$
16. Binary 1011110100000001 as a signed decimal is:
    - 1011 1101 0000 0001
    - 1011 1101 0000 0000
    - 0100 0010 1111 1111
    - $-((4 * 2^{12}) + (2 * 2^{8}) + (15 * 2^{4}) + 15) = -17151$
17. If we had 20-bit registers, the smallest signed decimal integer value would be:

In a 20-bit signed system, the smallest value lies on -2^19 which is -524288

18. If we had 20-bit registers, the largest signed decimal integer value would be:

In a 20-bit signed system, the smallest value lies on -2^19 - 1, which is 524287

19. The modular sum of 16-bit hex values 3511 + 4FFC is:

0x3511 + 0x4FFC = 0x850D

20. The saturated sum of 16-bit hex values 3511 + 4FFC is:

If the two values are unsigned, the result of this operation would be 0x850D, the same as the previous answer; otherwise, the resulting value would be clamped to 0x7FFF, the largest signed positive 16-bit integer

21. The 16-bit operation 0x3511 + 0x4FFC has a carry (Y or N):

No there is no carry in this addition

22. The 16-bit operation 0x3511 + 0x4FFC has a overflows (Y or N):

If the two values are unsigned, the answer is no; otherwise, it is yes.

If the two values are signed, the final column would result in 1 (as a 16 carried from previous column) + 3 + 4, which is 8, giving it a negative sign. Because the two given numbers being added are both positive, and the answer is negative, this would cause an overflow.

B-level Problems

1. The modular sum of 16-bit hex values 6159 + F702 is:
   - 0xF702
   - + 0x6159
   - = 0x585B
2. The saturated sum of 16-bit hex values 6159 + F702 is:
   - 0xFFFF
3. The 16-bit operation 0x6159 + 0xF702 has a carry (Y or N):
   - Y
4. The 16-bit operation 0x6159 + 0xF702 has a overflows (Y or N):
   - Y if the values are unsigned, N otherwise
   - If the values are signed, the first value would be signed as a positive, while the second would be signed as a negative, resulting in subtraction, which would not cause an overflow in this case. If the numbers are unsigned, they are added like normal, and 15 + 6 would be over the 16 base limit, requiring a carry.
5. The modular sum of 16-bit hex values EEEE + C00C is:

EEEE = 61198

C00C = 49164

61198 + 49164 = 110362

110362 in hex is 0x1AC9A

Since the sum must be 16 bits in length to match the size of the two given values, the final result would overflow. Using modular overflow, the answer is set as 0xAC9A with a carry value(1) denoting its negativity being passed through to the overflow system register flag

6. The saturated sum of 16-bit hex values EEEE + C00C is:

110362

This exceeds the 16-bit limit 65535 so it saturates to 0xFFFF

7. The 16-bit operation 9EEE + AB0C has a carry (Y or N):
    - 9EEE + AB0c = 146FA
    - There is a carry from the 1 in 9EEE + A. Yes there is a carry
8. The 16-bit operation 9EEE + AB0C has a overflows (Y or N):
    - Yes, regardless of the signedness of the values,
    - Both values would share a negative sign if signed, so 9 + 11 or -9 - 11 would both overflow.
9. The negation of 16-bit word 0xB00F is:
    - Inverted: 0x4FF0
    - Add 1: 0x4FF1
10. The negation of 16-bit word 0x2232 is:
    - Inverted: 0xDDCD
    - Add1: 0xDDCE
11. The negation of 16-bit word 0x8000 is:
    - 8 0 0 0
    - 1000 0000 0000 0000
    - 0111 1111 1111 1111 + 1
    - 1000 0000 0000 0000
    - 8000
12. The negation of 32-bit word 0xFFF329BA is:
    - 15 15 15 3 2 9 11 10
    - 1111 1111 1111 0011 0010 1001 1011 1010
    - 0000 0000 0000 1100 1101 0110 0100 0101 = negation
    - 0000 0000 0000 1100 1101 0110 0100 0110 = + 1
    - 0 0 0 C D 6 4 6
    - 0x000CD646
13. 96.03125 as a 32-bit float, in hex is:
    - 96 = 1100000
    - .03125 * 2 = .0625 * 2 = 0.125 * 2 = 0.25 * 2 = 0.5 * 2 = 1
    - 96.03125 = 1100000 . 00001
    - Move decimal 6 spots to the left = 1.10000000001
    - 127 + 6 = 133

- Sign binary digit + binary 133 + converted number after decimal + 0s until 32 characters reached =
- 0 10000101 10000000001 000000000000
- Group into 4-bit sections for easy hex translation
- 0100 0010 1100 0000 0001 0000 0000 0000
- 4 2 12 0 1 0 0 0
- Hex = 42c01000

14. -16777216 as a 32-bit float, in hex is:
    - 1000000000000000000000000 in binary
    - Move the decimal 24 spaces, 127 + 24 = 151
    - 1 10010111 00000000000000000000000
    - 1100 1011 1000 0000 0000 0000 0000 0000
    - 12 11 8 0 0 0 0 0
    - Hex = cb800000
15. Hex 43700000, when interpreted as an IEEE-754 pattern, is in decimal:
    - First convert to binary
    - 4 3 7 0 0 0 0 0
    - 0100 0011 0111 0000 0000 0000 0000 0000
    - Separate based on sign, exponent(8 bits), then number (until 0 is reached, then the rest is just empty space)
    - 0 10000110 111
    - Positive number, 10000110 is 134, 134-127 = 7 decimal places, 111
    - (assume 1 in front of decimal) 1.1110000 - > 11110000.0 no decimal, 1110000 = 240 in decimal

16. Hex C0FF0000, when interpreted as an IEEE-754 pattern, is in decimal:
    - First convert to binary
    - 12 0 15 15 0 0 0 0
    - 1100 0000 1111 1111 0000 0000 0000 0000
    - Separate
    - 1 10000001 1111111 0000000000000000
    - Negative number, 10000001 is 129 -127 = 2 decimal places
    - 1.1111111 -> 111.11111
    - 111 = 7 0.11111 = 0.5 + 0.25 + 0.125 + 0.0625 + 0.03125 = 0.96875
    - -7.96875

A-level Problems

1. The largest finite IEEE-754 single precision float, in hex is:
    - 2^{127}*(2 - (1 / 2^{23})) ~= 3.4028235 * 10^{38}
    - 0x7F7FFFFF

2. The smallest finite IEEE-754 single precision float, in hex is:
    - 0xFF7FFFFF
3. The largest nonzero negative IEEE-754 single precision float, in hex is:
    - 1000 0000 0000 0000 0000 0000 0000 0001
    - 0x80000001
    - This is the largest negative possible (as in, closest to 0 as possible, but still negative)
4. The smallest non-zero positive IEEE-754 single precision float, in hex is:
    - 1 0000 0001 00000000000000000000000
    - 1000 0000 1000 0000 0000 0000 0000 0000
    - 0x80800000
    - This would be the smallest negative possible (as in farthest from zero as possible, as the value is signed negative in the MSD, and contains all 0s except for the LSD, making it as small as a negative as possible)
5. -5.125 X 2₉₀ as a 32-bit float, in hex is:
    - $-5.125 \times 2^{90} = -2.5625 \times 2^{91} = -1.28125 * 2^{92}$
    - Sign = -, MSB = 1
    - Exponent:
        - 92 + 127 = 219
        - 219 % 2 = 1, 217 // 2 = 109
        - 109 % 2 = 1, 108 // 2 = 54
        - 54 % 2 = 0, 54 // 2 = 27
        - 27 % 2 = 1, 27 // 2 = 13
        - 13 % 2 = 1, 13 // 2 = 6
        - 6 % 2 = 0, 6 // 2 = 3
        - 3 % 2 = 1, 3 // 2 = 1
        - 1 % 2 = 1, 1 // 2 = 0
        - 11011011
    - Mantissa:
        - 0.28125 - 0.5 = negative number, ignore
        - 0.28125 - 0.25 = 0.03125
        - 0.03125 - 0.125 = negative number, ignore
        - 0.03125 - 0.0625 = negative number, ignore
        - 0.03125 - 0.03125 = 0
        - 01001 (and many trailing zeros)
    - 1110 1101 1010 0100 0000 0000 0000 0000

- 0xEDA40000

6. 2^{-138} as a 32-bit float, in hex is:
    - Sign = +, MSB = 0
    - Exponent:
        - Lowest possible: 00000000
    - Mantissa:
        - Current exponent: 2^{-126}
        - Need to multiply by 2^{-12}, so set 12th bit of mantissa (and nothing else)
    - 0000 0000 0000 0000 0000 1000 0000 0000
    - 0x00000800
7. 1.5 X 2^{-143} as a 32-bit float, in hex is:
    - Sign = +, MSB = 0
    - Exponent:
        - Lowest possible: 00000000
    - Mantissa:
        - Need to multiply by $1.5 * 2^{-17} = 0.00001144409$
        - Start with 17th bit:
        - $0.00001144409 - 2^{-17} = 0.0000038147$
        - $0.0000038147 - 2^{-18} < 2^{-23}$, use
        - 0000 0000 0000 0000 1100 000
    - 0000 0000 0000 0000 0000 0000 0110 0000
    - 0x00000060
8. OPTIONAL — Try this for a challenge, a puzzle, or the experience:
   Hex C059000000000000, when interpreted as an IEEE-754 pattern, is in decimal:
    - 1100 0000 0101 1001 0000 0000 0000…
    - Sign: Negative
    - Exponent:
        - 1000 0000
        - 128
        - 2^1
    - Mantissa:
        - 1011001
        - 0.5 + 0.125 + 0.0625 + 0.0078125 = 0.6953125
    - $-2^1 * (1.6953125) = 3.390625$