Find A + B using two’s complement addition in base 10 and base 2 (Without truncation)? (Bit vector of length W = 4) Where the values of A is 1001 and B is 1111

Two’s complement is 1001 = -8 + 1 = -7

1111 = -1 = -8

1001

1111

11000

Find A + B using two’s complement addition in base 10 and base 2 (with truncation)? (Bit vector of length W = 4) Where the values of A is 1001 and B is 1111

Remove left most bit -8,1000

We can represent a bit pattern of length w = 4 with a single hex digit 0xE. For a two’s complement interpretation of these digits, Give the additive inverses for 0xE.

The **additive inverse** of a number is what you add to a number to create the sum of zero.

1110 + 10(2) = 0000

In an unsigned multiplication, if there are w bits then the true product will have how many bits?

2 \* w bits

What is the output of the expression when x = -4 and y = -1 using two’s complement multiplication. (There are four values as part of the options, The first is a true product in unsigned integer, the second is the binary representation of the first one. The third one is the unsigned integer of the truncated product (Truncated to 3 bits) and the fourth one is the binary representation of the third one).

-4 \* -1 = 1010 << 1110

True product 100 \* 001 = 00100 = -4 To represent in 2\*w bits wen need to add a zero in the front.

Find ways to express x \* K where x is some integer and K is 31, using only one operation, where we consider both additions and subtractions to be 1 and to have comparable cost. You may need to use some tricks beyond the simple form A and B rules we have considered so far. (Check all that apply).

x \* 31

x(32-1)

x\* 2^5 - x

(x<<5) - x

0x12 34 56 78 Big endian machine will take byte in given order

12 34 56 78

Small endian machine will take byte in reverse order