Because our computer uses 12-bits registers to represent numbers, we must use 2’s complement binary representation of 12-bits to represent numbers. The 2's complement representation of a negative number is obtained by taking the binary representation of its positive counterpart, inverting all the bits, and then adding 1.

Let's calculate the 2's complement binary representation for each of the given numbers:

-13:

true code’s binary representation of -13: 1000 0000 1101

one's complementary of 13: 1111 1111 0010

Adding 1: 1111 1111 0011

The 12-bit 2's complement representation of -13: 1111 1111 0011

56:

true code’s binary representation of 56: 0000 0011 1000

Since it is a positive number, the 2's complement representation is the same as the true code’s binary representation: 0000 0011 1000

-1:

true code’s binary representation of -1: 1000 0000 0001

one's complementary of 13: 1111 1111 1110

Adding 1: 1111 1111 1111

The 12-bit 2's complement representation of -1: 1111 1111 1111

-2048:

Since -2048 is not within the range that can be represented by 12 bits , we cannot accurately represent -2048 using true code and one's complementary.

But the 12-bit 2's complement representation of -2048 is 1000 0000 0000.

2048:

Since 2048 is not within the range that can be represented by 12 bits , we cannot accurately represent 2048 using 12-bit registers.

In summary, the 2's complement binary representations for the given numbers using 12-bit registers are:

-13: 1111 1111 0011  
56: 0000 0011 1000  
-1: 1111 1111 1111  
-2048: 1000 0000 0000  
2048: Not representable accurately with 12 bits