ASC 5

2’S COMPLEMENT

**Def**: Mathematically, the two’s complement REPRESENTATION of a NEGATIVE number is the value of 2^n – V where V is the absolute value of the represented number

Why do we need it? If we start from the definition, we can understand what was the official method that the 2’s complement was computed. The key words: 2’s complement, Representation and negative.

* REPRESENTATION: It is referred to a representation and not to an interpretation
* NEGATIVE: it represents the case (the 10%) where you find a necessity to compute 2’s complement. Only when you deal with a negative number or a representation of a negative number do you need to compute the 2’s complement

What do you do with what you obtain? We need to understand when we need to compute the 2’s complement and when to use it.

EXAMPLE:

1001 0011 – A REPRESENTATION IN BASE 2

**93h** – A REPRESENTATION IN BASE 16 (Base 16 is a zipped notation for base 2, it is an equivalent, therefore it is a REPRESENTATION. When you see base 16 what you see is practically base 2. Base 16 is JUST A NOTATION)

We have 2 **interpretations** possible in base 10:

* Unsigned INTERPRETATION
  + You start from base 16
  + 9 \* 16 + 3 => 147 in HEXAZECIMAL
* Signed INTERPRETATION
  + The signed interpretation is obtained by computing: – (2’s complement of 1001 0011)
  + There are 4 methods for computing the 2’s complement for a representation
    1. We subtract 2^8(n) = 256 (does not fit anymore in a byte) from our value
       - 1 0000 0000 – 1001 0011 = 0110 1101 which is **6dh** in base 16 and 109 in base 10
       - We receive the absolute value of this configuration to which we need to transform it to negative numbers
       - => -109
    2. We reverse all the bits and then we add 1 to what we calculated
       - 1001 0011 -> 0110 1100
       - 0110 1100 + 1 = 0110 1101
    3. Going from right to left, you start to pass your initial configuration, keeping all the starting 0’s and transform the first 1 you find to 0. All the other bits that follow will be complemented
       - 0110 1100 - > 0110 1101 (?)
       - This is the best for obtaining the binary configuration for computing the 2’s complement of a binary configuration. Both are REPRESENTATIONS!
    4. The sum of the absolute values of the 2 complementary binary configurations/representations is the cardinal of the set representable on that dimension(size)
       - The cardinal of the set of values representable on a byte is 2^n = 256. On one word it is 2^65536.
       - The sum of the 2 complementary values is 256
       - You do the sum – 1001 0110 (147) and you get the absolute value of the 2’s complementary, 109
       - It can be applied only in cases in which you won’t obtain a binary configuration or in cases in which you don’t need to compute the binary configuration

Amissible representation intervals:

* ON N POSITIONS (bits in our case): -> 2 ^ n values
* In the unsigned interpretation:
  + [0, 2^n-1] (integer numbers)
* In the signed interpretation:
  + [-2^(n-1), 2^(n-1)-1]
* **INTERPRETATION ON 1 BYTE**
  + [0,255]
  + [-128, 127]
* **INTERPRETATION ON 1 WORD**
  + [0,65535]
  + [-32768, 32767]

THE OFFICIAL DEF: A REPRESENTATION IS THE 2’S COMPLEMENT OF ANOTHER REPRESENTATION

In practice a sequence of 32 bits is too big to understand for us. It is incorrect to say that -109 is the 2’s complement of 147. But this is used in practice because you cannot say that -147 is the 2’s complement of 109. This 2’s complement thing is a one directional thing and it is not true in both ways.

ALWAYS IDENTIFY THE KEY WORDS IN DEFINITION AND IN A QUESTION (or you get a restanta ***moron/bitch***)

EXERCISE:

1. Which is the signed interpretation of (…)
   * (…) we will have:
     1. 1001 0011
     2. 93h
     3. 147d
   * Possible options:
     1. 0110 1101
     2. -109
     3. 6Dh
     4. +147
     5. None of the above
   * CORRECT ANSWER
     1. B)
     2. B) (base 16 is nothing else than base 2 so the 2 questions are the same)
     3. E (THE KEY WORDS FROM THIS QUESTIONS ARE: INTERPRETATION AND (…). IF WE ARE ASKED TO GIVE AN INTERPRETATION, (…) MUST ALWAYS BE A REPRESENTATION. THEREFORE, THIS IS A ”STUPID QUESTION”
2. A configuration that starts with 0: 0xxx…x

* The values in unsigned?
  + The same as in the signed representation: +abc
* The values in signed?
  + The same as in the unsigned representation: +abc
* Why are they the same? The number starts with ZERO
* By intersecting the two domains of signed and unsigned representations of a byte you get [0,127]. These are all the values in binary representations that start with 0. The others are the values that start with 1

1. 0xxx…x

* Which is the representation for -abc?
* KEYWORDS: representation, -abc
* ANSWER: the 2’s complement of 0xxx..x (always a number that starts with 1 EXCEPT 0)

1. 1xxx…x

* The values in unsigned?
  + +def – a positive value
* The values in signed?
  + – (2’s complement of 1xxx…x)

1. 1xxx…x
   * Which is the representation for -def?
   * KEYWORDS: REPRESENTATION, -def
   * How do we do this? We compute the complement and we obtain a number that starts with 0 in binary, except 1000…0
   * The proposed answer would be 2’s complement of 1xxx…x, but it is WRONG
   * The answer is WRONG!
   * What would be the correct one? We don’t have an 8 bits representation. The proof? ADMISSIBLE REPRESENTATION INTERVALS. -147 is representable only in sth bigger than a byte
   * The correct answer would be the 2’s complement of 1xxx…x of the UNSIGNED EXTENSION to an immediate superior size of the initial 1xxx…x configuration (000...01xxx…x)
   * THE PROOF: we start from 147 (1001 0011) and extend it: 0000 0000 1001 0011 and we get it’s complement. We take the first one and then we change everything else –> 1111 1111 0110 1101 = FF6Dh
   * This is da invatatura de minte
2. Which is the binary representation that makes complementary 109 with -147
   * -109 can’t really be expressed as the complement of 147, but there is no binary representation in 1 byte which makes complementary 109 with -147
3. Which is the MINIMUM number of BITS on which we can represent (...)
   * Example: 3
   * Proof: we need to determine the minimum n for which 0 <= 3 <= 2^n-1
   * On 1 bit we represent 2^1 values
   * On 2 bits we represent 2^2 values = 4 values [0,3] in unsigned and [-2, +1]
   * Answer the value in which you represent 3 is 2 bits and it’s value is 10
   * Example: -3
   * Answer: [-4, +3] – signed, [0, 7] – unsigned
   * Answer: the value in which you represent -3 is 3 bits and it’s value is 101, which is 5 in unsigned representation
   * Example: -147
   * 1111 1111 0110 1101: we take all the values until the first 1 after the number!
4. Who is for 2 the second complement?
   * One binary configuration cmpared to another -TRUE
   * 2 Absolute values relative to eachother – TRUE
   * A signed decimal number vs an unsigned one – TRUE BUT FORCED