

BLG 231E - Digital Circuits Assignment 1

Student ID : 150200054
Full Name : Aslı Yel

Part 1 - Computer Arithmetic

1) a) Using signed 2's-complement representation, convert the decimal numbers (-102) and (-27) to 8-bit binary integers. Show ALL work.

```
(-102)
                                                                                    (-27)
               102 / 2 = 51
                               remainder 0
                                                                                        remainder 1
               51/2 = 25
                                                                        27 / 2 = 13
                               remainder 1
                                                                                        remainder 1
               25 / 2 = 12
                               remainder 1
                                                                         13 / 2 = 6
                                                                                        remainder 0
                12 / 2 = 6
                               remainder 0
                                                                          6/2 = 3
                                                                                        remainder 1
                               remainder 0
                 6/2 = 3
                                                                          3/2 = 1
                                                                                        remainder 1
                 3/2 = 1
                               remainder 1
                                                                          1/2 = 0
                               remainder 1
                 1/2 = 0
             natural binary weighted coding:
                                                                      natural binary weighted coding:
102_{10} = 1.2^{6} + 1.2^{5} + 0.2^{4} + 0.2^{3} + 1.2^{2} + 1.2^{1} + 0.2^{0} = 1100110_{2}
                                                                27_{10} = 1.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 11011_2
            8-bit signed binary representation:
                                                                    8-bit signed binary representation:
                 (+102)_{10} = (01100110)_2
                                                                           (+27)_{10} = (00011011)_2
                2's complement operation:
                                                                        2's complement operation:
        (-102)_2 = [1's complement of (+102)_2] + 1
                                                                 (-27)_2 = [1's complement of (+27)_2] + 1
               = 2's complement of (+102)_2
                                                                        = 2's complement of (+27)_2
                                                                    8-bit (+27)_{10}
            8-bit (+102)<sub>10</sub>
                                    : 01100110
                                                                                            : 00011011
            1's complement
                                                                     1's complement
                                                                                            : 11100100
                                    : 10011001
                                                                    Result: 8-bit (-27)<sub>10</sub> : 11100101
           Result: 8-bit (-102)_{10}: 10011010
            8-bit signed binary representation:
                                                                    8-bit signed binary representation:
                  (-102)_{10} = (10011010)_2
                                                                           (-27)_{10} = (11100101)_2
```

- **b)** Carry out the **binary** operations given below, and explain your answers using terms such as *carry*, *borrow*, and *overflow*. To interpret the results, use only binary numbers.
- P.S. I assumed that the operations are carried out on **8-bit** binary numbers and I used mentioned terms accordingly.

```
i. (-102) + (-27)
```

```
8-bit signed binary representation (from 1.a): (-102)_{10} = 10011010_2 (-27)_{10} = 11100101_2
```

addition of 8-bit signed numbers

```
10011010 : -102

+11100101 : -27

101111111 : cannot be represented

8th bit is the sign bit. Sign of the result is positive.

9th bit is ignored.
```

While both operands are negative, result is positive (neg. + neg. \rightarrow pos. overflow case) Therefore there is an OVERFLOW.

ii.
$$(-102) - (-27)$$

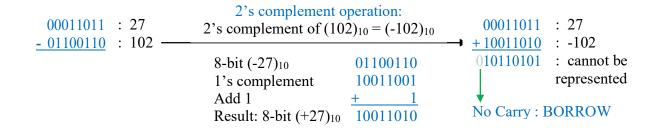
```
8-bit signed binary representation (from 1.a): (-102)_{10} = 10011010_2 (-27)_{10} = 11100101_2
```

subtraction of 8-bit signed numbers

Operation of subtracting two negative operands (neg. - neg.) is not one of the overflow cases. So there is NO OVERFLOW.

8-bit unsigned representation: $27_{10} = 00011011_2$ $102_{10} = 01100110_2$

subtraction of 8-bit signed numbers



9th bit of the result is zero. There is NO CARRY. There is a BORROW.

Part 2 – Boolean Algebra

2) Simplify each Boolean expression using algebraic manipulation (axioms and theorems). Show ALL work.

Show the steps of the simplification, and write which axiom/theorem you used in each step next to the simplification.

a)

-----RESULT

= X'Y' + Y'T + XZT'-----