Name: Id#

COE 202, Term 151

Digital Logic Design

Quiz# 4

Date: Tuesday, Nov. 3

# 

# **Q1**. It is required to design a Tripler circuit. The circuit receives an n-bit number X and computes the result Y=3\*X.

# If the input is an *n-*bit unsigned number, what is the size of the output “*y*” in bits?

# 



# The circuit can be constructed using *n identical copies* of the basic 1-bit cell shown to the right. The cell processes one input bit (X*i*) and produces one output bit (Y*i*) and two output carry bits (CO0 and CO1). To allow for cascading *n* such cells to implement an *n-*bit Tripler, the basic cell also accepts two input carry bits (CI0 and CI1). When the output carry equals 1 then CO1 CO0 = 01, while when it equals 2 then CO1 CO0 = 10.

# The Figure below shows how a 4-bit Tripler circuit is implemented using 4 copies of the basic 1-bit cell.



# Derive the truth table for the basic one-bit cell.

# (Hint: *As the initial input carries = 00, the maximum carry from one cell to the next is 2*)

# Derive a minimized sum-of-product expressions for the outputs of the basic one-bit cell.

**Q2.**

1. Fill in all blank cells in the two tables below. All binary representations use 7 bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Binary | Equivalent decimal value with the binary interpreted as: | | | |
| Unsigned number | Signed-magnitude number | Signed-1’s complement number | Signed-2’s complement number |
| 1011010 |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal | Binary representation in: | | |
| Signed-magnitude notation | Signed-1’s complement notation | Signed-2’s complement notation |
| - 59 |  |  |  |

b. Using 2’s-complement signed arithmetic in **5 bits**, perform the following operations in binary. Show all your work. Verify that you get the expected decimal results.

**Check for overflow and mark clearly any occurrences of it.**

|  |  |
| --- | --- |
| (i)  11010  + 11001 | (ii)  00101  - 10100 |
| (iii)  (+5)  + (-9) | (iv)  (-6)  - (+8) |

c. When doing signed 2’s complement arithmetic in **6 bit**s, the *smallest* binary number **that will cause overflow** when *subtracted* from 101000)2 is \_\_\_\_\_\_\_\_\_\_\_.