# Switching Circuits and Logic Design
# CS21002 (section-1)
# Tutorial 1

---

1. Practice subtraction of unsigned numbers and addition, subtraction of numbers represented in signed 2's complement. For the same pair of numbers, try to find complements with various number of digits.

2. Simplify the following switching expressions as much as you can by using the switching identities done in the class. This and the next problem is to highlight and demonstrate the ad-hoc nature of algebraic simplification and to motivate the more systematic simplification methods that we will study later.
   (a) $x' + y' + xyz'$.
   (b) $(x' + xyz') + (x' + xyz')(x + x'y'z)$.
   (c) $xy + wxyz' + x'y$.
   (d) $a + a'b + a'b'c + a'b'c'd$.
   (e) $w'x' + x'y' + w'z' + yz$.

3. Find, by inspection, the complement of each of the following expressions and then simplify it.
   (a) $x'(y' + z')(x + y + z')$.
   (b) $(x + y'z')(y + x'z')(z + x'y')$.
   (c) $w' + (x' + y + y'z')(x + y'z)$.

4. Demonstrate, without using perfect induction, whether each of the following equations is valid.
   (a) $(x + y)(x' + y)(x + y')(x' + y') = 0$.
   (b) $xy + x'y' + x'yz = xyz' + x'y' + yz$.
   (c) $xyz + wy'z' + wxz = xyz + wy'z' + wxy'$.
   (d) $xy + x'y' + xy'z = xz + x'y' + x'yz$.

5. Find all values of switching variables $A, B, C$, and $D$ by solving the following set of simultaneous equations:

$A' + AB = 0$,
$AB = AC$,
$AB + AC' + CD = C'D$.

6. Prove that if $w'x + yz' = 0$, then $wx + y'(w' + z') = wx + xz + x'z' + w'y'z$.

7. (a) Prove that each weight in a weighted code is a non-zero integer in the range $[-9, +9]$.

   (b) Can a weighted code have two different sets of weights that explain it? Justify your answer.

   (c) Prove that Gray code and excess-3 codes are not weighted codes.

   (d) Prove that in a self-complementing weighted code, the sum of the weights is 9.

8. The steps of the **shift and add-3** algorithm (also called the *double-dabble algorithm*) for converting from binary to BCD that we saw in the class can be reversed to obtain an algorithm for converting BCD to binary. Work out the details of the algorithm. Run the algorithm on the BCD codeword 00101001 to produce its binary equivalent. Show each step.

9. We defined the Gray code by repeated reflection of shorter codewords. From that definition prove the conversion formula from binary to Gray code that we wrote down in class. Also derive an inverse formula for converting Gray code to binary.

   [Hint for the first part: Use mathematical induction on the length of codewords.]

10. Prove that there is no single error correcting 4-bit code for decimal digits. More generally, show that for each positive integer $n$, the number of codewords in a single error correcting $n$-bit code is at most $2^{n-1}$.

11. In the canonical sum of product representations we express a switching function as OR of minterms.
    (a) Argue that if we replace the OR with Exclusive-OR, the resultant expression continues to be logically equivalent to the same function.
    (b) Argue that each switching function can be expressed as XOR of products of unprimed (i.e. uncomplemented) literals and switching constants (0 and 1).

    Example: The NOR of two switching variables $x_1$ and $x_2$ is equivalent to $1 \oplus x_1 \oplus x_2 \oplus x_1x_2$.

    You may solve this first for 2-input functions to gain some insight. Use the various nice algebraic properties of the XOR function that we wrote down in class.

12. The majority function $M(x, y, z)$ is equal to 1 when two or three of its arguments equal 1, that is, $M(x, y, z) = xy + xz + yz = (x + y)(x + z)(y + z)$.
    (a) Show that $M(a, b, M(c, d, e)) = M(M(a, b, c), d, M(a, b, e))$.

(b) Show that $M(x, y, z)$, the complementation operation and the constant 0 form a functionally complete set of operations.