**CS29002 SWITCHING LABORATORY**
**CSE Department, IIT Kharagpur**
**Spring Semester 2015–16**
**Module A: Boolean Logic and Combinational Logic Circuits**
**Assignment 1**
**Date: 11–Jan–2016**

Design a 3-bit to 8-bit encoder using NAND and NOR gates, having the following input/output behavior:

| **Input** | **Output** |
|---|---|
| ***x y z*** | ***a b c d  e f g h*** |
| 0 0 0 | 0 0 0 0  0 0 0 1 |
| 0 0 1 | 0 0 0 0  0 0 1 1 |
| 0 1 0 | 0 0 0 0  0 1 1 1 |
| 0 1 1 | 0 0 0 0  1 1 1 1 |
| 1 0 0 | 0 0 0 1  1 1 1 1 |
| 1 0 1 | 0 0 1 1  1 1 1 1 |
| 1 1 0 | 0 1 1 1  1 1 1 1 |
| 1 1 1 | 1 1 1 1  1 1 1 1 |

**Note:** NAND and NOR are universal gates, that is, any Boolean function can be realized by using only NAND gates or only NOR gates. To see how, let us first take the case of two-input NAND gates. We have:

$$\begin{aligned} x' &= (1x)', \\ xy &= ((xy)')' = (1(xy)')', \text{ and} \\ x+y &= (x'y')' = ((1x)'(1y)')'. \end{aligned}$$

This means that the three basic Boolean operations AND, OR and NOT can be realized by NAND gates. For NOR gates, we have the dual formulas:

$$\begin{aligned} &x' = (0+x)', \\ &x+y = ((x+y)')' = (0+(x+y)')', \text{ and} \\ &xy = (x'+y')' = ((0+x)'+(0+y)')'. \end{aligned}$$

Express each of the eight output bits as a simple Boolean function in the input variables $x$, $y$, $z$, and apply the NAND/NOR implementation whichever you like. As an example, note that the fourth least significant bit $e$ is 1 if and only if $xyz = 011$ or $x = 1$. Therefore this bit has the formula $e = x'yz + x$. Another possibility is to express each bit (except the leftmost) in terms of its previous bit. That is, we have $a = xyz$, $b = a + xyz'$, $c = b + xy'z$, and so on. This second approach adds to the complexity and depth of the circuit. For example, $h$ is always 1, and $d$ is the same as the input $x$. Therefore designing these as $h = 1$ and $d = x$ is more pragmatic than taking $h = g + x'y'z'$ and $d = c + xy'z'$. Use an appropriate mixture of both the techniques so that you have a relatively simple circuit.