**Binary to Ternary Numerals**

Ryan Young  
CSE 2300w

Lab #3

**Objectives:**

The main objective of this lab was to give us more practice with a number code conversion that relates binary to ternary. It also helped hone our abilities with logic works specifically using and and or gates.

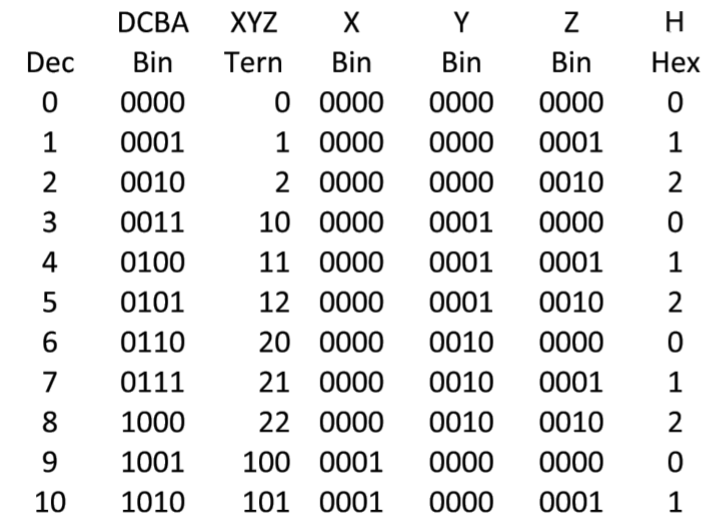
**Introduction:**

In order to convert one system of numbers to another we have learned the easiest way to do this is through a truth table. The input values we have are labeled “DCBA’ and range from 0000 to 1010, while the output values are the ternary equivalents XYZ from 000 to 101. Using binary switches, and and or gates, not inverters and a hex display in this lab I will create a circuit on the logicworks program to convert those binary values to the ternary ones.

**Procedure:**

Step 1.) Identify which values of “Z” always stay at zero when converting from the “DCBA” input. These values will simply be connected to ground because they always have a value of zero.

Step 2.) Create an equation based off the table of values found below that converts each of the outputs in terms of DCBA.



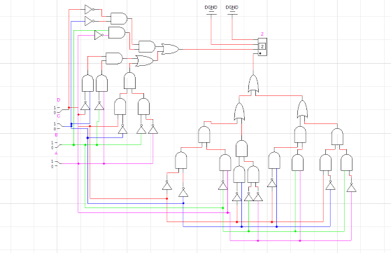
Step 3.) After creating the equations to change the “DCBA” inputs to the proper ternary outputs, draw on paper a rough plan for how your circuit is going to look. If the input is a zero that means we will need a “not inverter” and if the input is a one then we can directly wire to the switch.

Step 4.) Once you have a rough idea of how your circuit is going to look start building it in the logicworks program. Start with the Hex display and work backwards. There should be one or gate per input (the last two the first two are grounded only). And then from there continue to work backwards using the equations that we derived and the table given.

Step 5.) Go back a test each input and make sure it corresponds with the correct output bas on the table above.

**Results:** The conversion from binary in the form of DCBA to was done through the final circuit i created in the logicworks program (seen below). Also shown below is the table I created to help decide which gates needed inversion in front of them or not.

|  |  |  |
| --- | --- | --- |
| Row | DCBA | Converted to |
| 3 | 0010 | D”not”, C”not”, B, A”not” |
| 3 | 0101 | D”not”, C, B”not”, A |
| 3 | 1000 | D, C”not”, B”not”, A”not” |
| 4 | 0001 | D”not”, C”not”, B”not”, A |
| 4 | 0100 | D”not”, C, B”not”, A”not” |
| 4 | 0111 | D”not”, C, B, A |
| 4 | 1010 | D, C”not”, B, A”not” |

****

As you can see in the picture above the circuit the input is “0101” and it has the correct ternary output of “2”.

**DIscussion:**

While my circuit produces the correct output based on an input, I in hindsight created it very efficiently. Instead of simply putting a “not inverter” at “D” switch I put one just before every and gate that the signal went into.

**Conclusion:**

Through my successful completion of this lab I gained more knowledge with regards to how to convert number systems, using logic works, and overall building of circuits. Furthermore I realized just how careful I have to be when building circuits. I had to go back multiple times and make sure that the type of gate I was using the correct one, or make sure my work looked neat.

**Questions:**

1. What do you think you could do to reduce the number of gates in your solution? :

In order to reduce the number of gates in my solution I think using four input and gates instead of multiple two input and gates would significantly reduce the number of gates needed. Furthermore adding the use of a three input or gate would reduce the number of total gates as well.