Questions:

1. 1. Total Gate Count: **NOR/NOR,** it has 2 less gates than NAND/NAND
   2. Input Count: **NOR/NOR**, it has 6 less inputs than NAND/NAND
   3. IC Count: **NOR/NOR,** it has 1 less IC than NAND/NAND
   4. Time Required to implement: **NOR/NOR**, it takes less time to plug everything in
   5. Purchase Price of IC’s: **NOR/NOR**, 1 IC is less than one 2-input NAND IC

|  |  |
| --- | --- |
| **AND/OR** | **OR/AND** |
| BD + AD + BC + AC | (A+B)(C+D) |
| **NAND/NAND** | **NOR/NOR** |
| [(BD)’ (AD)’(BC)’(AC)’]’ | [(A+B)’ + (C+D)’]’ |
| **OR/NAND** | **AND/NOR** |
| [(B’+D’)(A’+D’)(B’+C’)(A’+C’)]’ | [(A’B’) + (C’D’)]’ |
| **NOR/OR** | **NAND/AND** |
| (B’+D’)’+(A’+D’)’+(B’+C’)’+(A’+C’)’ | (A’B’)’ (C’D’)’ |

1. The OR/AND circuit uses the fewest gates, 2 OR’s and 1 AND.
2. The OR/AND circuit uses the fewest inputs, 6. This could be a design consideration because it reduces the amount of wire used.
3. 7 input NAND gates aren’t generally produced because few applications call for a NAND gate of that size (at least that we have dealt with so far). Also, 7 is an odd number, and most systems use divisors of 2 for logic inputs.

Conclusion:

Tyler: In this lab I learned how to use DeMorgan’s Theorem to turn a SOP or POS into many different combinations of gates. This will be helpful later on when we have complex circuits and limited numbers of gates. I also used truth tables to test my findings, and to make sure the gates worked correctly. I also practiced using K-maps to find the most minimal implementation of that particular logic.

Susan: Today’s lab took the theorems learned in lecture and allowed us to create a system that gave a desired output. It was interesting to see how many different ways there are to get the same output with the same input. I enjoyed analyzing the different equations and determining which was best based off of different things such as cost, number of inputs, and number of outputs.