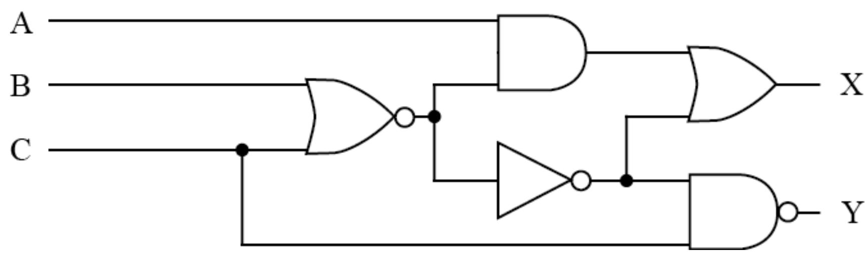
**Ex:**




Complete the truth table, below, for the logic circuit above. (Note that you must fill in some missing input values in the table, too.)

| A | B | C | X | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | | |
| 0 | | 1 | | |
| | 1 | 0 | | |
| | | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |

**SOL'N:** First, we fill in the input values (in red), which correspond to counting up in binary. Second, we look for simple cases where the output values may be determined. If C = 0, for example, Y must be one (green values) because of the NAND gate. If the output of the NOR gate is 0, (if either B or C is a 1), the output of the NOT gate will be 1 and Y will equal C-not (purple values) and the output of the OR gate, X, will be 1 (blue values).

If the output of the NOR gate is 1, (B and C = 0), then X will be 1 if and only if A is 1 , (orange values), owing to the AND gate.

| A | B | C | X | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |