Digital Logic (Part 1)

1. For the following circuit:
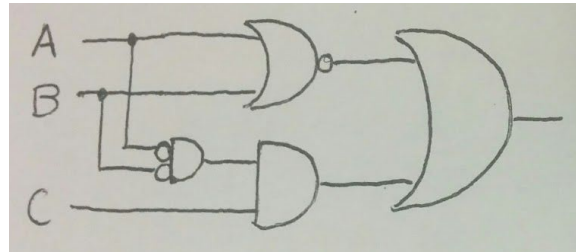
  a. Write its truth table:

A	B	C	F (output)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

   Are there any shortcuts?
   If output = 1, then either (A+B)' = 1 (A = 0, B = 0, C = 0 or 1)
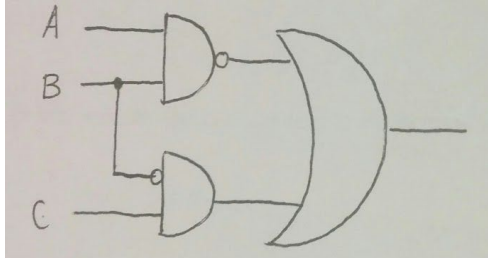   **or A'B'C = 1 (A = 0, B = 0, C = 1)**

  b. What is the equivalent, <u>unsimplified</u> Boolean expression?
   (A + B)' + A'B'C = $\overline{A+B} + \overline{A}\,\overline{B}\,C$

  c. Simplify or shorten your answer in part c, using boolean algebra laws, and <u>write the laws you used</u> next to each step.
   A'B' + A'B'C   DeMorgan's
   A'B'   Absorption

Extra practice: (repeat #1 for this circuit)

A	B	C	F (output)
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

(AB)' + B'C   **(original)**
A' + B' + B'C   **DeMorgan's**
A' + B'   **Absorption**

2. For the boolean equation: F = (A'A + (BC')')'+B $(\overline{\overline{AA} + \overline{BC}} + B)$

    a. Draw the unsimplified circuit:
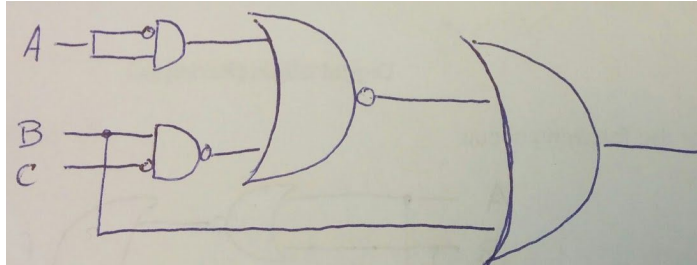
    What was your thought process in implementing this circuit?
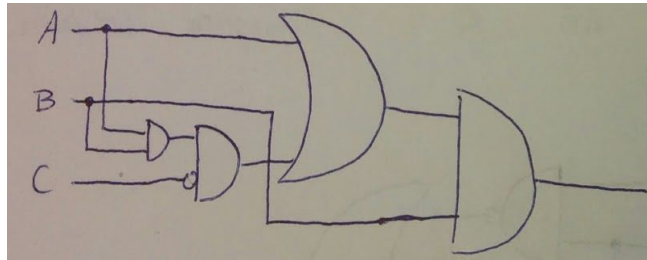   **(ideally) inside-out or outside-in**

    b. Write its truth table:

A	B	C	F (output)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

    c. Simplify the expression by using boolean algebra laws to shorten the number of terms, and write the corresponding laws next to each step:

    F = (A'A + (BC')')'+B   (original)
    F = (0 + (BC')')'+B = ((BC')')' + B   Inverse
    F = BC' + B   Involution
    F = B   Absorption

Extra practice: **(A + ABC')B**

(repeat #2 for this expression)

A	B	C	F (output)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

(A + ABC')B   **(original)**
AB   **Absorption again**

3. What are the different levels of abstraction in the hardware/software interface?
   (Hint: the rainbow)
   [This is just to help you keep the overall context and big idea of the class in mind.]