Logic Gates

1. Label the following logic gates:

   ![Logic Gates Diagram]

   **Solution:** not, and, or, xor, nand, nor, xnor

2. Convert the following to boolean expressions:

   (a) NAND

   **Solution:** $\overline{A}B + \overline{A}B + AB$

   (b) XOR

   **Solution:** $\overline{A}B + AB$

   (c) XNOR

   **Solution:** $\overline{A}B + AB$

3. Create an AND gate using only NAND gates.

   ![AND Gate Using NAND Diagram]

   **Solution:**

4. How many different two-input logic gates can there be? How many n-input logic gates?

   **Solution:** A truth table with $n$ inputs has $2^n$ rows. Each logic gate has a 0 or a 1 at each of these rows. Imagining a function as a $2^n$-bit number, we count $2^{2^n}$ total functions, or 16 in the case of $n = 2$.

Boolean Logic

$1 + A = 1 \
A + \overline{A} = 1 \
A + AB = A \
(A + B)(A + C) = A + BC \
0B = 0 \
B\overline{B} = 0 \
A + \overline{AB} = A + B$

DeMorgan’s Law:

$\overline{AB} = \overline{A} + \overline{B} \
\overline{A + B} = \overline{A} \overline{B}$

1. Minimize the following boolean expressions:

   (a) Standard: $(A + B)(A + \overline{B})C$
Solution:

\[(AA + A\bar{B} + AB + B\bar{B})C = (A + A(\bar{B} + B))C = AC\]  \hspace{1cm} (1)

(b) Grouping & Extra Terms: \(\bar{A}B\bar{C} + \bar{A}B\bar{C} + AB\bar{C} + \bar{A}B\bar{C} + ABC + A\bar{B}C\)

\[
\begin{align*}
\bar{A}\bar{C}(\bar{B} + B) + A\bar{C}(B + \bar{B}) + AC(B + \bar{B}) &= \bar{A}\bar{C} + A\bar{C} + AC \\
&= \bar{A}\bar{C} + A\bar{C} + AC + AC \\
&= (\bar{A} + A)\bar{C} + A(\bar{C} + C) \\
&= A + \bar{C}
\end{align*}
\]  \hspace{1cm} (2)

(c) DeMorgan's: \(A(BC + BC)\)

\[
\begin{align*}
\overline{A(BC + BC)} &= \overline{A + BC + BC} \\
&= \overline{A + B\bar{C}BC} \\
&= \overline{A + (B + C)(\bar{B} + \bar{C})} \\
&= \overline{A + BC + \bar{B}C}
\end{align*}
\]  \hspace{1cm} (3)

State

1. Fill out the timing diagram for the circuit below:

```
+--++  +--++  +--++
IN|D Q|--s0-|D Q|--s1-|D Q|--Out
   +--++  +--++  +--++
|       |       |
CLK-----------
```

clk
in
s0
s1
out
2. Fill out the timing diagram for the circuit below:

```
+-----+  +-----+
A--|D Q|->R1--|D Q|->R2--
    +-+-    +-+-
        |      |
CLK---->o--+
```

**FSM**

1. Fill in the following FSM for outputting a 1 whenever we have two repeating bits as the most recent bits, and a 0 otherwise. You may not need all states.