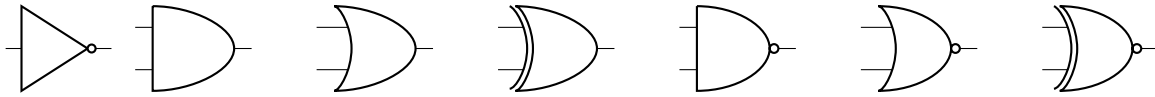


Logic Gates

1. Label the following logic gates:



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

2. Convert the following to boolean expressions:

(a) NAND

Solution: $\bar{A}\bar{B} + \bar{A}B + A\bar{B}$

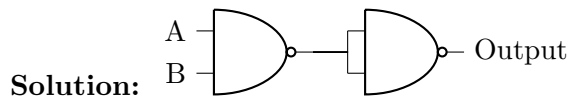
(b) XOR

Solution: $\bar{A}B + A\bar{B}$

(c) XNOR

Solution: $\bar{A}\bar{B} + AB$

3. Create an AND gate using only NAND gates.



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

$$\begin{array}{llll}
 1 + A = 1 & A + \bar{A} = 1 & A + AB = A & (A + B)(A + C) = A + BC \\
 0B = 0 & B\bar{B} = 0 & A + \bar{A}B = A + B & \\
 \text{DeMorgan's Law: } & \overline{AB} = \bar{A} + \bar{B} & \overline{A + B} = \bar{A}\bar{B} &
 \end{array}$$

1. Minimize the following boolean expressions:

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

Solution:

$$(AA + A\bar{B} + AB + B\bar{B})C = (A + A(\bar{B} + B))C = AC \quad (1)$$

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

Solution:

$$\bar{A}\bar{C}(\bar{B} + B) + A\bar{C}(B + \bar{B}) + AC(B + \bar{B}) = \bar{A}\bar{C} + A\bar{C} + AC \quad (2)$$

$$= \bar{A}\bar{C} + A\bar{C} + A\bar{C} + AC \quad (3)$$

$$= (\bar{A} + A)\bar{C} + A(\bar{C} + C) \quad (4)$$

$$= A + \bar{C} \quad (5)$$

(c) DeMorgan's: $\overline{A(\bar{B}\bar{C} + BC)}$

Solution:

$$\overline{A(\bar{B}\bar{C} + BC)} = \bar{A} + \overline{\bar{B}\bar{C} + BC} \quad (6)$$

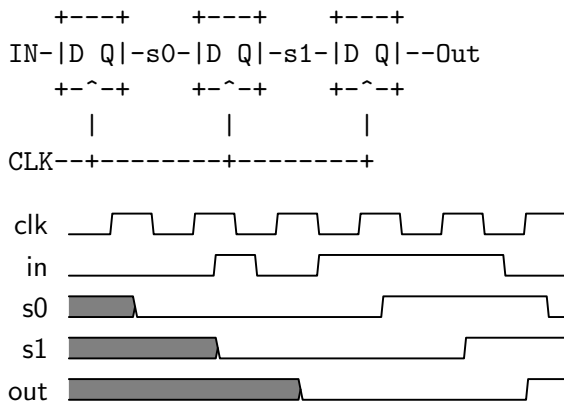
$$= \bar{A} + \overline{\bar{B}\bar{C}}\overline{BC} \quad (7)$$

$$= \bar{A} + (B + C)(\bar{B} + \bar{C}) \quad (8)$$

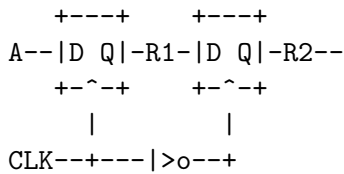
$$= \bar{A} + B\bar{C} + \bar{B}C \quad (9)$$

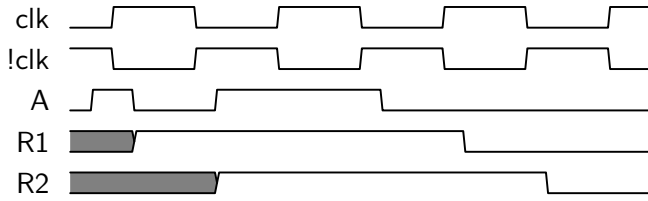
State

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



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





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.

