

Notes on Boolean Operations - Physics 120 - David Kleinfeld - 2018

Arithmetic and logic

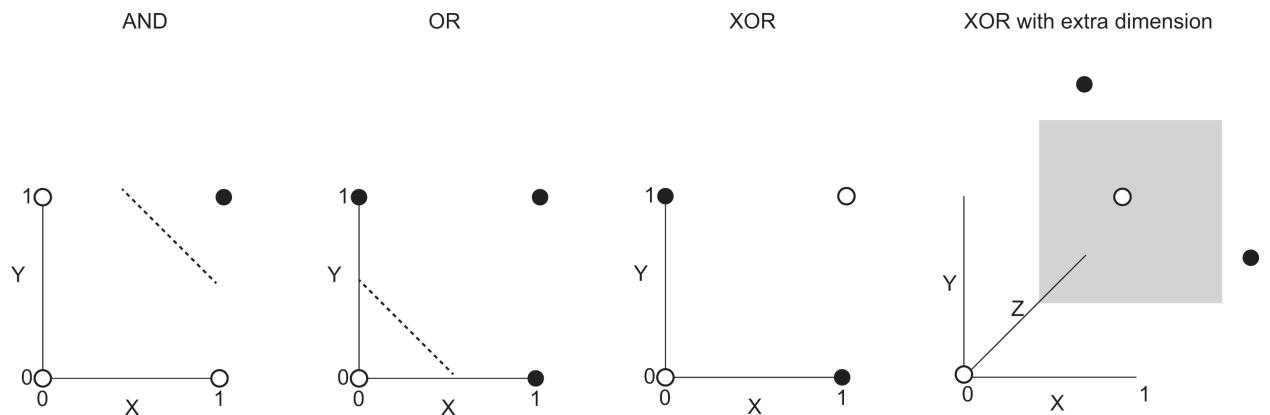
Addition of two bits, either a 0 (LOW) or 1 (HIGH), serves to illustrate XOR and AND:

$\begin{array}{r} 0 \\ \underline{0} \\ 00 \end{array}$	$\begin{array}{r} 1 \\ \underline{0} \\ 01 \end{array}$	$\begin{array}{r} 0 \\ \underline{1} \\ 01 \end{array}$	$\begin{array}{r} 1 \\ \underline{1} \\ 10 \end{array}$
---	---	---	---

The 1's place is the logic function Exclusive OR (XOR) while the 2's place is the function AND, i.e.,

X	Y	AND		X	Y	XOR		X	Y	OR		X	\bar{X}
0	0	0		0	0	0		0	0	0		0	1
0	1	0	and	0	1	1	, along with	0	1	1	and	0	1
1	0	0		1	0	1		1	0	1		1	0
1	1	1		1	1	0		1	1	1			

means "NOT". Plots of AND and OR show that they can be computed by dividing the part of the plane with 0's from that with 1's by a single line. This is not true for XOR; we must add a dimension so that a plane can separate the 0's from the 1's. This will require two levels of logic.



For simplicity of notation, we write X AND Y as XY and X OR Y as X+Y.

Identities:

- $0+X = X$
- $1+X = 1$
- $0X = 0$
- $1X = X$
- $X\bar{X} = 0$
- $X+\bar{X} = 1$

Distribution:

- $X(Y+Z) = XY+XZ$
- $X+(YZ) = (X+Y)(X+Z)$

Absorption:

- $X+(XY) = X$
- $X+(\bar{X}Y) = X+Y$
- $X(X+Y) = X$
- $X(\bar{X}+Y) = XY$

Association:

$$X+(Y+Z) = (X+Y)+Z = (X+Z)+Y$$

DeMorgan:

- $\overline{X+Y} = \bar{X}\bar{Y}$
- $\overline{XY} = \bar{X}+\bar{Y}$
- $\overline{\bar{X}} = X$

Logical depth

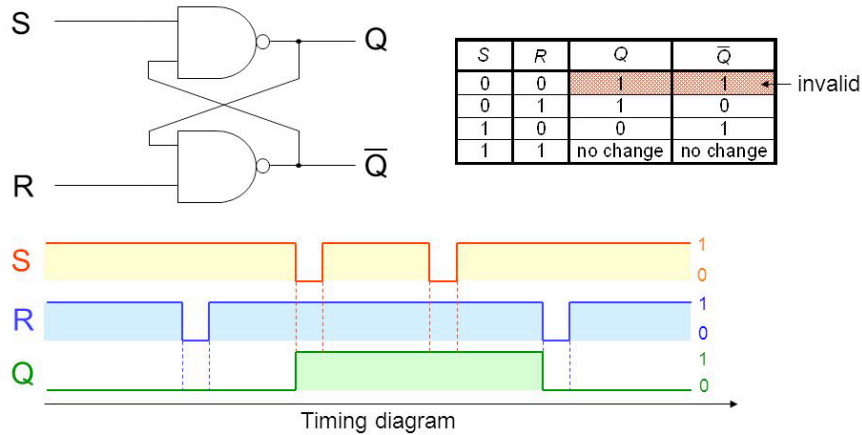
Any Boolean function can be written as the OR of multiple ANDs and thus can be computed with two levels of logic, albeit with arbitrarily large fan outs from the first layer of ANDs and fan ins to the second layer of ORs.

X_1	X_2	X_3	X_4	...	X_N	F
1	1	0	1	...	0	1
0	1	0	1	...	0	0
0	0	0	0	...	1	0
0	1	0	1	...	1	1
⋮	⋮	⋮	⋮		⋮	⋮

so that $F = (X_1 X_2 \bar{X}_3 X_4 \dots \bar{X}_N) + (\bar{X}_1 X_2 \bar{X}_3 X_4 \dots X_N) + \dots$. Thus, looking back, $XOR = \bar{Y}X + Y\bar{X}$.

Flip flops

The simplest storage device is the set-reset or "S-R" flip-flop, which can be formed by inhibitory, or "NOT"ed, feedback between two AND gates, implemented using two NAND gates. Note that this can also be realized with two NOR gates.



The "D" (data) flip flop adds circuitry to ensure that the "S" and "R" inputs are never the same. The "D" input is used in place of the "Set" signal in a S-R flip-flop. To prevent the output of the flip flop from changing on every pulse applied to "D", an additional input called the "CLK" (clock or enable) isolates the "D" input from the flip flop's latching circuitry after the desired value of "D" has been stored. The flip flop will "SET" if "D" is held HIGH and will "RESET" if "D" is held LOW so long as the clock input is HIGH. The flip-flop will not change state once the clock input goes LOW.

A variant of the D flip-flop triggers on the edge of the clock pulse, rather than the level, for added speed and some added danger!

