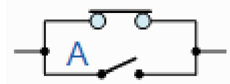
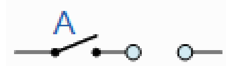
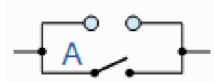
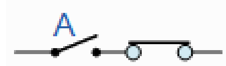
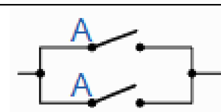
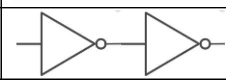
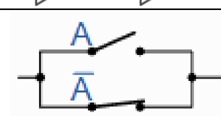
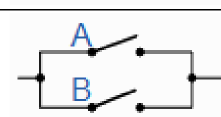
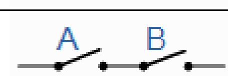
Harold's Boolean Algebra
Cheat Sheet

12 September 2021

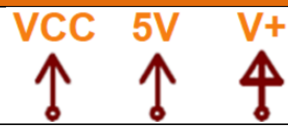
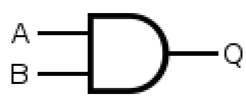
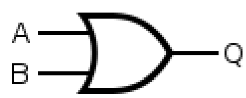
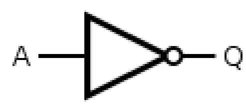
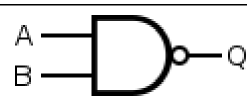
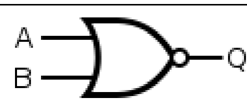
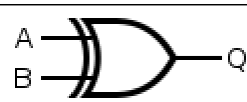
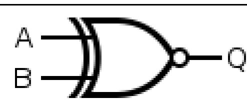
Boolean Algebra

Boolean Expression	Law or Rule	Equivalent Circuit	Description
$A + 1 = 1$	Annulment (OR)		A in parallel with closed = "CLOSED"
$A \bullet 0 = 0$	Annulment (AND)		A in series with open = "OPEN"
$A + 0 = A$	Identity (OR)		A in parallel with open = "A"
$A \bullet 1 = A$	Identity (AND)		A in series with closed = "A"
$A + A = A$	Idempotent (OR)		A in parallel with A = "A"
$AA = A$	Idempotent (AND)		A in series with A = "A"
$\overline{(\overline{A})} = A$	Double Negation		NOT NOT A (double negative) = "A"
$A + \overline{A} = 1$	Complement (OR)		A in parallel with NOT A = "CLOSED"
$A\overline{A} = 0$	Complement (AND)		A in series with NOT A = "OPEN"
$A + B = B + A$	Commutative (OR)		A in parallel with B = B in parallel with A
$AB = BA$	Commutative (AND)		A in series with B = B in series with A
$A(B + C) = AB + AC$	Distributative (OR)		Permits the multiplying or factoring out of an expression
$A + BC = (A + B)(A + C)$	Distributative (AND)		

$A + (B + C) = (A + B) + C = A + B + C$	Associative (OR)		Allows the removal of brackets from an expression and regrouping of the variables
$A(BC) = (AB)C = ABC$	Associative (AND)		
$A + (AB) = A$	Absorptive (OR)		Enables a reduction in a complicated expression to a simpler one by absorbing like terms
$A(A + B) = A$	Absorptive (AND)		
$A + \overline{A}B = A + B$	Absorptive (Derived)		Reduces a complicated expression to a simpler one by absorbing compliment term
$\overline{(A + B)} = \overline{A} \bullet \overline{B}$	De Morgan's Theorem (NOR)		Invert and replace OR with AND
$\overline{AB} = \overline{A} + \overline{B}$	De Morgan's Theorem (NAND)		Invert and replace AND with OR

Source: https://www.electronics-tutorials.ws/boolean/bool_6.html

Boolean Logic Gates

Boolean Logic	Notation	Gate	Description
IDENTITY	1 T True		On, Tautology, High voltage (typically +5V)
NULL	0 F ⊥ False	**GND** 	Off, Contradiction, Low voltage (typically 0V)
Input	A, B, C, D		Line, Wire, Connects to
Output	W, X, Y, Z		Line, Wire, Connects from
AND	A • B AB A. B $A \wedge B$ $A \cap B$		AND, BUT, Multiply, Conjunction, Intersection
OR	$A + B$ $A \vee B$ $A \cup B$ A | B		Inclusive-OR, Add, Disjunction, Union
NOT	$\overline{A}$ A^ A′ ¬A ~A ! A		NOT, Invert, Negation, Change, Difference
NAND	$\overline{AB}$ $A \barwedge B$ A | B*		Not AND
NOR	$\overline{A + B}$ $A \overline{\vee} B$ $A \downarrow B$		Not OR
XOR	$A \oplus B$ $A \veebar B$ $A\overline{B} + \overline{A}B$		Exclusive-OR, Both A and B are different
XNOR	$A \odot B$ $\overline{A \oplus B}$ $AB + \overline{AB}$		Exclusive-NOR, Both A and B are the same

Boolean Logic Truth Tables

Inputs		Outputs								
A	B	AND .	NAND	OR +	NOR	XOR ⊕	XNOR ⊙	NOT $\bar{A}$	VCC 1	GND 0
0	0	0	1	0	1	0	1	A=1	1	0
0	1	0	1	1	0	1	0	A=1	1	0
1	0	0	1	1	0	1	0	A=0	1	0
1	1	1	0	1	0	0	1	A=0	1	0

Blank Truth Tables

Inputs		Output
A	B	X
0	0	
0	1	
1	0	
1	1	

Inputs			Output	
A	B	C	X	Y
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Inputs				Output		
A	B	C	D	X	Y	Z
0	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			
0	1	1	1			
1	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			
1	1	1	1			

Karnaugh Mapping (K-Map)

2-Bit K-Map		**A**	
		0	1
B	0		
	1		

4-Bit K-Map		**AB**			
		00	01	11	10
CD	00				
	01				
	11				
	10				

3-Bit K-Map		**AB**			
		00	01	11	10
C	0				
	1				

2x2 Group

1x4 Group

K-Map Rules

1) Circle only 1s (ones) and don't cares for Sum of Products (SOP), $e.g.\ \overline{A}\,\overline{B}\,\overline{C} + \overline{A}BC + AB\overline{C}$.
   a. Circle only 0s (zeros) and don't cares for Product of Sums (POS), $e.g.\ (A + \overline{B})(\overline{A} + B)$.
   b. Don't cares may be used or ignored.
2) No diagonals, only horizontal or vertical connections.
3) Group only adjacent cells in groups with powers of 2 (1x1, 1x2, 2x1, 2x2, 2x4, 4x2, 1x4, 4x1).
4) Make groups as large as possible.
5) Must group <u>all</u> 1s (ones) for SOP or all 0s (zeros) for POS.
6) Overlapping is allowed.
7) Wrapping around all edges allowed, both top-bottom edges and left-right edges.
8) Fewest groups possible (OPTIMAL).
9) For each circle, determine which inputs do not contribute to the logic (is both 0 and 1).
10) Write down equation as a SOP, $e.g.\ \overline{A}\,\overline{B}\,\overline{C} + \overline{A}BC + AB\overline{C}$