



BLOCOS LÓGICOS BÁSICOS																				
PORTA	Símbolo Usual	Tabela da Verdade	Função Lógica	Expressão																
E AND		<table border="1"> <tr> <th>A</th><th>B</th><th>S</th></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	Função E: Assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.	$S = A \cdot B$	
A	B	S																		
0	0	0																		
0	1	0																		
1	0	0																		
1	1	1																		
OU OR		<table border="1"> <tr> <th>A</th><th>B</th><th>S</th></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	Função E: Assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.	$S = A + B$	
A	B	S																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
NÃO NOT		<table border="1"> <tr> <th>A</th><th>S</th></tr> <tr> <td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td></tr> </table>	A	S	0	1	1	0	Função NÃO: Inverte a variável aplicada à sua entrada.	$S = \bar{A}$										
A	S																			
0	1																			
1	0																			
NE NAND		<table border="1"> <tr> <th>A</th><th>B</th><th>S</th></tr> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0	Função NE: Inverso da função E.	$S = \overline{(A \cdot B)}$	
A	B	S																		
0	0	1																		
0	1	1																		
1	0	1																		
1	1	0																		
NOU NOR		<table border="1"> <tr> <th>A</th><th>B</th><th>S</th></tr> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0	Função NOU: Inverso da função OU.	$S = \overline{(A + B)}$	
A	B	S																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	0																		
OU Exclusivo		<table border="1"> <tr> <th>A</th><th>B</th><th>S</th></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0	Função OU Exclusivo: Assume 1 quando as variáveis assumirem valores diferentes entre si.	$S = A \oplus B$ $S = \bar{A} \cdot B + A \cdot \bar{B}$	
A	B	S																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	0																		
Coincidência		<table border="1"> <tr> <th>A</th><th>B</th><th>S</th></tr> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	1	Função Coincidência: Assume 1 quando houver coincidência entre os valores das variáveis.	$S = A \cdot B$ $S = \bar{A} \cdot \bar{B} + A \cdot B$	
A	B	S																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	1																		

$$O1 = (\neg S1 \wedge \neg S2 \wedge S3 \wedge \neg S4 \wedge \neg S5 \wedge \neg S6 \wedge \neg S7) \mid (S1 \wedge S2 \wedge S3 \wedge S4 \wedge S5 \wedge S6);$$

$$O1 = (\neg a \&\& \neg b \&\& c \&\& \neg d \&\& \neg e \&\& \neg f \&\& \neg g) \mid\mid (a \&\& b \&\& c \&\& d \&\& e \&\& f)$$

Karnaugh Map Solver

**Function Info**

Output Name: O2  
One using the function result: 02

Input Names: S1,S2,S3,S4,S5,S6,S7  
Comma separated list of variable names: S1,S2,S3,S4,S5,S6,S7

**Settings:**

- Sum of Products
- Product of Sums (very slow with >10 variables)
- Draw Kmap
- Draw groupings

**Terms:**  
Minterms: 30,120,127  
Comma separated list of numbers: 0,1,2,3,4,5,6,7

**Solutions:**  
Generic:  $O2(S1, S2, S3, S4, S5, S6, S7) = S1 \bar{S}2 \bar{S}3 \bar{S}4 \bar{S}5 \bar{S}6 \bar{S}7 + S1 S2 \bar{S}3 \bar{S}4 \bar{S}5 \bar{S}6 \bar{S}7 + S1 \bar{S}2 S3 \bar{S}4 \bar{S}5 \bar{S}6 \bar{S}7$

**VHDL:**  
 $O2 \leftarrow (\text{not } S1 \text{ and not } S2 \text{ and not } S3 \text{ and not } S4 \text{ and not } S5 \text{ and not } S6 \text{ and not } S7) \text{ or } (S1 \text{ and } S2 \text{ and } S3 \text{ and not } S4 \text{ and not } S5 \text{ and not } S6 \text{ and not } S7) \text{ or } (S1 \text{ and } S2 \text{ and } S3 \text{ and } S4 \text{ and } S5 \text{ and } S6 \text{ and } S7);$

**Verilog:**  
assign O2 = ( $\neg S1 \& \neg S2 \& \neg S3 \& \neg S4 \& \neg S5 \& \neg S6 \& \neg S7$ ) | ( $S1 \& S2 \& S3 \& \neg S4 \& \neg S5 \& \neg S6 \& S7$ ) | ( $S1 \& S2 \& S3 \& S4 \& S5 \& S6 \& S7$ );

**Karnaugh Map**

$O2 = (\neg S1 \& \neg S2 \& S3 \& S4 \& S5 \& S6 \& \neg S7) \mid (S1 \& S2 \& S3 \& S4 \& \neg S5 \& \neg S6 \& \neg S7) \mid (S1 \& S2 \& S3 \& S4 \& S5 \& S6 \& S7);$

$O2 = (\neg a \&& \neg b \&& c \&& d \&& e \&& f \&& \neg g) \mid (a \&& b \&& c \&& d \&& \neg e \&& \neg f \&& \neg g) \mid (a \&& b \&& c \&& d \&& e \&& f \&& g)$

( $\neg a \& \neg b \& \neg c \& \neg d \& \neg e \& \neg f \& \neg g$ ) || ( $a \& b \& c \& d \& \neg e \& \neg f \& \neg g$ ) || ( $a \& b \& c \& d \& e \& f \& g$ ) - Wolfram|Alpha

b && c && d && e && f && \neg g) || (a && b && c && d && \neg e && \neg f && \neg g) || (a && b && c && d && e && f && g)

**NATURAL LANGUAGE** **MATH INPUT** **EXTENDED KEYBOARD** **EXAMPLES** **UPLOAD** **RANDOM**

**Input**

$(\neg a \wedge \neg b \wedge c \wedge d \wedge e \wedge f \wedge \neg g) \vee$   
 $(a \wedge b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \vee (a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge g)$

((Not a) AND (Not b) AND c AND d AND e AND f AND (Not g))  
Or (a AND b AND c AND d AND (Not e) AND (Not f) AND (Not g))  
Or (a AND b AND c AND d AND e AND f AND g)

$\neg \text{expr}$  is the logical NOT function  
 $e_1 \wedge e_2 \wedge \dots$  is the logical AND function  
 $e_1 \vee e_2 \vee \dots$  is the logical OR function

**Minimal forms**

DNF	$(a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge g) \vee$ $(a \wedge b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \vee (\neg a \wedge \neg b \wedge c \wedge d \wedge e \wedge f \wedge \neg g)$
CNF	$(\neg a \vee b) \wedge (a \vee f) \wedge (a \vee \neg g) \wedge$ $(\neg b \vee \neg e \vee g) \wedge (a \wedge d \wedge (e \vee f)) \wedge (f \wedge \neg g)$
ANF	$(a \wedge b \wedge c \wedge d \wedge e \wedge f) \vee (a \wedge b \wedge c \wedge d \wedge \neg e) \vee (a \wedge b \wedge c \wedge d \wedge \neg f) \vee$ $(a \wedge b \wedge c \wedge d \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge \neg g) \vee (b \wedge c \wedge d \wedge e \wedge f) \vee$ $(c \wedge d \wedge e \wedge f \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge e \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge f \wedge g) \vee$ $(a \wedge c \wedge d \wedge e \wedge f \wedge g) \vee (b \wedge c \wedge d \wedge e \wedge f \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge g)$
NOR	$(\neg a \vee b) \vee (a \vee f) \vee (a \vee \neg g) \vee$ $(\neg b \vee \neg e \vee g) \vee (\neg c \vee \neg d \vee (e \vee f)) \vee (f \vee \neg g)$
NAND	$(a \bar{\wedge} b \bar{\wedge} c \bar{\wedge} d \bar{\wedge} e \bar{\wedge} f \bar{\wedge} g) \bar{\wedge}$ $(a \bar{\wedge} b \bar{\wedge} c \bar{\wedge} d \bar{\wedge} \neg e \bar{\wedge} \neg f \bar{\wedge} \neg g) \bar{\wedge} (\neg a \bar{\wedge} \neg b \bar{\wedge} \neg c \bar{\wedge} \neg d \bar{\wedge} \neg e \bar{\wedge} \neg f \bar{\wedge} \neg g)$
AND	$\neg (a \wedge \neg b) \wedge \neg (\neg a \wedge \neg f) \wedge \neg (\neg a \wedge g) \wedge$ $\neg (b \wedge e \wedge g) \wedge c \wedge d \wedge \neg (e \wedge f) \wedge \neg (f \wedge g)$
OR	$\neg (\neg a \vee \neg b \vee \neg c \vee \neg d \vee \neg e \vee \neg f \vee \neg g) \vee$ $\neg (\neg a \vee \neg b \vee \neg c \vee \neg d \vee \neg e \vee f \vee g) \vee$ $\neg (a \vee b \vee \neg c \vee \neg d \vee \neg e \vee \neg f \vee g) \vee$

(assuming NAND and NOR are n-ary operators)

$e_1 \vee e_2 \vee \dots$  is the logical OR function  
 $e_1 \wedge e_2 \wedge \dots$  is the logical AND function  
 $e_1 \vee e_2 \vee \dots$  is the logical XOR function  
 $e_1 \vee e_2 \vee \dots$  is the logical NOR function  
 $e_1 \wedge e_2 \wedge \dots$  is the logical NAND function

Privacy Policy.

Karnaugh Map Solver

**Function Info**

Output Name: O3  
One using for function result: O3

Input Names: Comma separated list of variable names: S1,S2,S3,S4,S5,S6,S7

**Settings:**

- Sum of Products (selected)
- Product of Sums (very slow with >10 variables)
- Draw Kmap
- Draw groupings

**Terms:** Minterms: Comma separated list of numbers: 24,30,126  
Don't Cares: Comma separated list of numbers: 51,52,53,54,55,56,57

**Solutions:** Generic:  $O3(S1, S2, S3, S4, S5, S6, S7) = S1 \bar{S}2 S3 S4 S5 S6 S7 + S1 S2 \bar{S}3 S4 S5 S6 S7 + S1 S2 S3 S4 S5 S6 S7$

**VHDL:**  $O3 \leftarrow (\text{not } S1 \text{ and not } S2 \text{ and } S3 \text{ and } S4 \text{ and not } S5 \text{ and not } S6 \text{ and not } S7) \text{ or } (\text{not } S1 \text{ and not } S2 \text{ and } S3 \text{ and } S4 \text{ and } S5 \text{ and not } S6 \text{ and not } S7) \text{ or } (S1 \text{ and } S2 \text{ and } S3 \text{ and } S4 \text{ and } S5 \text{ and } S6 \text{ and not } S7)$

**Verilog:** assign O3 = ( $\neg S1 \& \neg S2 \& S3 \& S4 \& \neg S5 \& \neg S6 \& \neg S7$ ) | ( $\neg S1 \& \neg S2 \& S3 \& S4 \& S5 \& S6 \& \neg S7$ ) | ( $S1 \& S2 \& S3 \& S4 \& S5 \& S6 \& \neg S7$ );

**Karnaugh Map**

The Karnaugh map shows the minterms 24, 30, and 126 highlighted. The map has 16 cells corresponding to the 7-variable input space. Minterm 24 is at S1=0, S2=1, S3=0, S4=1, S5=0, S6=1, S7=0. Minterm 30 is at S1=1, S2=0, S3=1, S4=0, S5=1, S6=0, S7=1. Minterm 126 is at S1=0, S2=1, S3=1, S4=1, S5=1, S6=1, S7=1.

$$O3 = (\neg S1 \& \neg S2 \& S3 \& S4 \& \neg S5 \& \neg S6 \& \neg S7) | (\neg S1 \& \neg S2 \& S3 \& S4 \& S5 \& S6 \& \neg S7) | (S1 \& S2 \& S3 \& S4 \& S5 \& S6 \& \neg S7);$$

O3 = (~a and ~b and c and d and ~e and ~f and ~g) or (~a and ~b and c and d and e and f and ~g) or (a and b and c and d and e and f and ~g)

Wolfram Alpha query:  $(\neg a \wedge \neg b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \vee (\neg a \wedge \neg b \wedge c \wedge d \wedge e \wedge f \wedge \neg g) \vee (a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge \neg g)$

Minimal forms

DNF	$(a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge \neg g) \vee (\neg a \wedge \neg b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \vee (\neg a \wedge \neg b \wedge c \wedge d \wedge e \wedge f \wedge \neg g) \vee (a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge \neg g)$
CNF	$(\neg a \vee b) \wedge (\neg a \vee e) \wedge (a \vee \neg b) \wedge c \wedge d \wedge (\neg e \vee f) \wedge (e \vee \neg f) \wedge \neg g$
ANF	$(c \wedge d) \vee (a \wedge c \wedge d) \vee (b \wedge c \wedge d) \vee (c \wedge d \wedge e) \vee (c \wedge d \wedge \neg e) \vee (c \wedge d \wedge f) \vee (c \wedge d \wedge \neg f) \vee (a \wedge b \wedge c \wedge d \wedge e) \vee (a \wedge b \wedge c \wedge d \wedge \neg e) \vee (a \wedge b \wedge c \wedge d \wedge f) \vee (a \wedge b \wedge c \wedge d \wedge \neg f) \vee (b \wedge c \wedge d \wedge e) \vee (b \wedge c \wedge d \wedge \neg e) \vee (b \wedge c \wedge d \wedge f) \vee (b \wedge c \wedge d \wedge \neg f) \vee (c \wedge d \wedge f \wedge g) \vee (c \wedge d \wedge \neg f \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge e \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge \neg e \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge f \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge \neg f \wedge g) \vee (b \wedge c \wedge d \wedge e \wedge g) \vee (b \wedge c \wedge d \wedge \neg e \wedge g) \vee (b \wedge c \wedge d \wedge f \wedge g) \vee (b \wedge c \wedge d \wedge \neg f \wedge g) \vee (a \wedge b \wedge c \wedge d \wedge e \wedge \neg g) \vee (a \wedge b \wedge c \wedge d \wedge \neg e \wedge \neg g) \vee (a \wedge b \wedge c \wedge d \wedge f \wedge \neg g) \vee (a \wedge b \wedge c \wedge d \wedge \neg f \wedge \neg g)$
NOR	$(\neg a \vee b) \vee (\neg a \vee \neg e) \vee (\neg b \vee \neg e) \vee (\neg c \vee \neg d) \vee (\neg e \vee \neg f) \vee \neg g$
NAND	$(a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge \neg g) \wedge (\neg a \wedge \neg b \wedge \neg c \wedge \neg d \wedge \neg e \wedge \neg f \wedge \neg g) \wedge (\neg a \wedge \neg b \wedge \neg c \wedge \neg d \wedge e \wedge f \wedge \neg g) \wedge (\neg a \wedge \neg b \wedge \neg c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \wedge (\neg a \wedge \neg b \wedge c \wedge \neg d \wedge \neg e \wedge \neg f \wedge \neg g) \wedge (\neg a \wedge \neg b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \wedge (\neg a \wedge \neg b \wedge c \wedge d \wedge e \wedge \neg f \wedge \neg g) \wedge (\neg a \wedge \neg b \wedge c \wedge d \wedge e \wedge f \wedge \neg g)$
AND	$\neg (a \wedge \neg b) \wedge \neg (a \wedge \neg e) \wedge \neg (a \wedge \neg f) \wedge \neg (c \wedge \neg d) \wedge \neg (c \wedge \neg e) \wedge \neg (c \wedge \neg f) \wedge \neg g$
OR	$\neg (\neg a \vee b \vee \neg c \vee \neg d \vee \neg e \vee \neg f \vee g) \vee \neg (a \vee b \vee \neg c \vee \neg d \vee \neg e \vee f \vee g)$

(assuming NAND and NOR are n-ary operators)

Other forms

ESOP	$(a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge \neg g) \vee (\neg a \wedge \neg b \wedge c \wedge d \wedge e \wedge f \wedge \neg g) \vee (\neg a \wedge b \wedge \neg c \wedge d \wedge e \wedge f \wedge \neg g)$
------	--

Enlarge Data Customize Plain Text