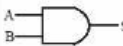

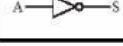

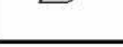
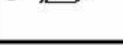



BLOCOS LÓGICOS BÁSICOS																			
PORTA	Símbolo Usual	Tabela da Verdade	Função Lógica	Expressão															
E AND		<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><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><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><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><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><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><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><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.	$S=A \odot B$ $S=\bar{A} \cdot \bar{B} + A \cdot B$
A	B	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

Karnaugh Map Solver

Function Info

Output Name:
O2

Input Names:
S1,S2,S3,S4,S5,S6,S7

Settings:

- Sum of Products
- Product of Sums
- Draw Kmap
- Draw groupings

Reset Everything

Terms

Minterms:
30,120,127

Don't Cares:
None

Reset Terms

Solutions:

Generic:
O2[S1, S2, S3, S4, S5, S6, S7] = S1 S2 S3 S4 S5 S6 S7 + S1 S2 S3 S4 S5 S6 S7 + S1 S2 S3 S4 S5 S6 S7

VHDL:
O2 <= (not S1 and not S2 and S3 and S4 and S5 and S6 and not S7) or (S1 and S2 and S3 and S4 and not S5 and not S6 and not S7) or (S1 and S2 and S3 and S4 and S5 and S6 and S7);

Verilog:
assign O2 = (~S1 & ~S2 & S3 & S4 & S5 & S6 & ~S7) | (S1 & S2 & S3 & S4 & ~S5 & ~S6 & ~S7) | (S1 & S2 & S3 & S4 & S5 & S6 & S7);

Karnaugh Map

O2 S4,S5,S6,S7

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Cost free to send any bugs or feedback to bugs@charlie-coleman.com

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

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

(~a & ~b & c & d & e & f & ~g) || (a & b & c & d & e & ~f & ~g) || (a & b & c & d & e & f & g) - Wolfram|Alpha

26+c+%26%26+d+%26%26+e+%26%26+f+%26%26+~g%29+%7C%7C+%28a+%26%26+b+%26%26+c+%26%26+d+%26%26

b & c & d & e & f & ~g) || (a & b & c & d & e & ~f & ~g) || (a & b & c & d & e & f & g)

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

Input

(~a & ~b & c & d & e & f & ~g) V
(a & b & c & d & e & ~f & ~g) V (a & b & c & d & e & f & 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)

~ expr is the logical NOT function
e1 & e2 ... is the logical AND function
e1 v e2 v ... is the logical OR function

Minimal forms **Text notation**

DNF	(a & b & c & d & e & f & ~g) V (a & b & c & d & e & ~f & ~g) V (~a & ~b & c & d & e & f & ~g)
CNF	(~a V b) & (a V f) & (a V ~g) & ~g (~b V ~e V g) & c & d & e & (e V ~f) & (f V ~g)
ANF	(a & b & c & d) V (c & d & e & f) V (a & b & c & d & e) V (a & b & c & d & f) V (a & b & c & d & g) V (a & c & d & e & f) V (b & c & d & e & f) V (c & d & e & f & g) V (a & b & c & d & e & g) V (a & b & c & d & f & g) V (a & c & d & e & f & g) V (b & c & d & e & f & g) V (a & b & c & d & e & f & g)
NOR	(~a V ~b) V (a V f) V (a V ~g) V (~b V ~e V g) V ~c V ~d V (e V ~f) V (f V ~g)
NAND	(a & b & c & d & e & f & ~g) ~ (a & b & c & d & e & ~f & ~g) ~ (~a & ~b & c & d & e & f & ~g)
AND	~(a & ~b) & ~ (~a & ~f) & ~ (~a & ~g) & ~g ~(b & c & ~g) & c & d & e & (~e & f) & ~ (~f & g)
OR	~ (~a V ~b V ~c V ~d V ~e V ~f V ~g) V ~ (~a V ~b V ~c V ~d V e V f V g) V ~ (a V b V ~c V ~d V ~e V ~f V g)

(assuming NAND and NOR are n-ary operators)

e1 v e2 v ... is the logical OR function
e1 & e2 ... is the logical AND function
e1 ~ e2 ~ ... is the logical XOR function
e1 v e2 ~ ... is the logical NOR function
e1 & e2 ~ ... is the logical NAND function

Privacy Policy.

Function Info

Output Name:
O3

Input Names:
S1,S2,S3,S4,S5,S6,S7

Settings:

- Sum of Products
- Product of Sums
- Draw Kmap
- Draw groupings

Reset Everything

Terms

Minterms:
24,30,126

Don't Cares:
-

Reset Terms

Solutions:

Generic:
O3(S1, S2, S3, S4, S5, S6, S7) = S1'S2'S3455'S6S7 + S1'S2'S345556S7 + S1S23545556S7

VHDL:
O3 <= (not S1 and not S2 and S3 and S4 and not S5 and not S6 and not S7) or (not S1 and not S2 and S3 and S4 and S5 and S6 and not S7) or (S1 and S2 and S3 and S4 and S5 and S6 and not S7);

Verilog:
assign O3 = (~S1 & ~S2 & S3 & S4 & ~S5 & ~S6 & ~S7) | (~S1 & ~S2 & S3 & S4 & S5 & S6 & ~S7) | (S1 & S2 & S3 & S4 & S5 & S6 & ~S7);

Karnaugh Map

O3

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

$$O3 = (\sim a \ \text{and} \ \sim b \ \text{and} \ c \ \text{and} \ d \ \text{and} \ \sim e \ \text{and} \ \sim f \ \text{and} \ \sim g) \ \text{or} \ (\sim a \ \text{and} \ \sim b \ \text{and} \ c \ \text{and} \ d \ \text{and} \ e \ \text{and} \ f \ \text{and} \ \sim g) \ \text{or} \ (a \ \text{and} \ b \ \text{and} \ c \ \text{and} \ d \ \text{and} \ e \ \text{and} \ f \ \text{and} \ \sim g)$$

Input

(~a ^ ~b ^ c ^ d ^ ~e ^ ~f ^ ~g) V (~a ^ ~b ^ c ^ d ^ e ^ f ^ ~g) V (a ^ b ^ c ^ d ^ e ^ f ^ ~g)

((NOT a) AND (NOT b) AND c AND d AND (NOT e) AND (NOT f) AND (NOT g)) OR ((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 e AND f AND (NOT g))

~ e.g. is the logical NOT function
e1 ^ e2 ^ ... is the logical AND function
e1 v e2 v ... is the logical OR function

Minimal forms

DNF (a ^ b ^ c ^ d ^ e ^ f ^ ~g) V (~a ^ ~b ^ c ^ d ^ e ^ f ^ ~g) V (~a ^ ~b ^ c ^ d ^ ~e ^ f ^ ~g)

CNF (~a v b) ^ (~a v e) ^ (a v ~b) ^ (c ^ d ^ ~e) ^ (e v f) ^ (e v ~f) ^ ~g

ANF (c ^ d) v (a ^ c ^ d) v (b ^ c ^ d) v (c ^ d ^ e) v (c ^ d ^ f) v (c ^ d ^ g) v (a ^ b ^ c ^ d) v (a ^ c ^ d ^ e) v (a ^ c ^ d ^ f) v (a ^ c ^ d ^ g) v (b ^ c ^ d ^ e) v (b ^ c ^ d ^ f) v (b ^ c ^ d ^ g) v (c ^ d ^ e ^ f) v (c ^ d ^ e ^ g) v (c ^ d ^ f ^ g) v (a ^ b ^ c ^ d ^ e) v (a ^ b ^ c ^ d ^ f) v (a ^ b ^ c ^ d ^ g) v (a ^ c ^ d ^ e ^ f) v (a ^ c ^ d ^ e ^ g) v (a ^ c ^ d ^ f ^ g) v (a ^ b ^ c ^ d ^ e ^ f) v (a ^ b ^ c ^ d ^ e ^ g) v (a ^ b ^ c ^ d ^ f ^ g) v (a ^ b ^ c ^ d ^ e ^ f ^ g)

NOR (~a v b) v (~a v e) v (a v ~b) v (~c v d) v (~e v f) v (e v ~f) v g

NAND (a ^ b ^ c ^ d ^ e ^ f ^ ~g) ^ (~a ^ ~b ^ c ^ d ^ e ^ f ^ ~g) ^ (~a ^ ~b ^ c ^ d ^ ~e ^ f ^ ~g)

AND (~a ^ ~b) ^ (~a ^ e) ^ (~a ^ b) ^ c ^ d ^ ~e ^ (e ^ f) ^ (~e ^ f) ^ ~g

OR ~(~a v ~b v ~c v ~d v ~e v ~f v g) V (~a v b v ~c v ~d v ~e v f v g) V (~a v b v ~c v ~d v e v f v g)

(assuming NAND and NOR are n-ary operators)

Other forms

ESOP (a ^ b ^ c ^ d ^ e ^ f ^ ~g) v (~a ^ ~b ^ c ^ d ^ e ^ f ^ ~g) v (~a ^ ~b ^ c ^ d ^ ~e ^ f ^ ~g)