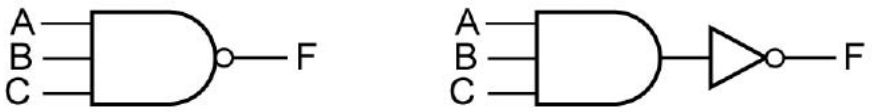
Note Title 3/25/2009

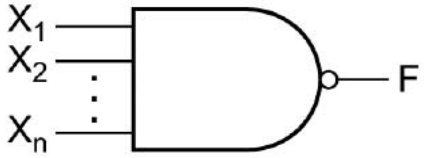
Ch. 7 text

AND-NOT



(a) 3-input NAND gate

(b) NAND gate equivalent



(c) n-input NAND gate

Figure 7-8: **NAND Gates**

(d) a 3-input NAND gate

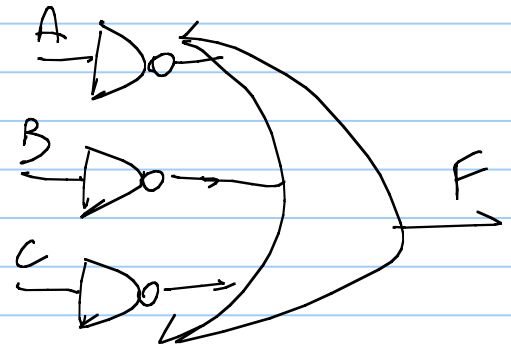
$F = (\text{from } (a)) = (A \cdot B \cdot C)' =$

$= (\text{De Morgan's Law}) =$

$A' + B' + C' = (\text{from } (d)) = F$

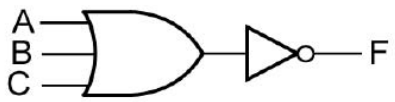
(e)

(e)

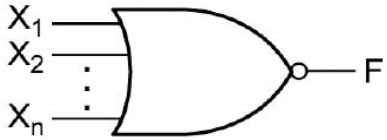


(d) is a shorthand for (e).

(a) 3-input NOR gate

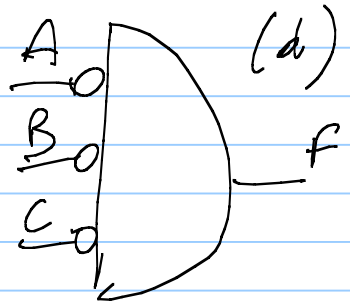


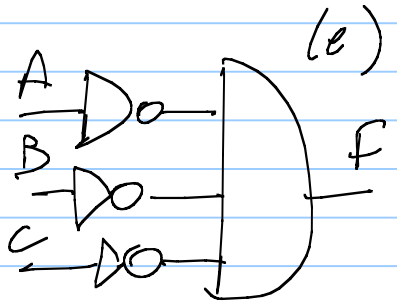
(b) NOR gate equivalent



(c) n-input NOR gate

Figure 7-9: **NOR Gates**





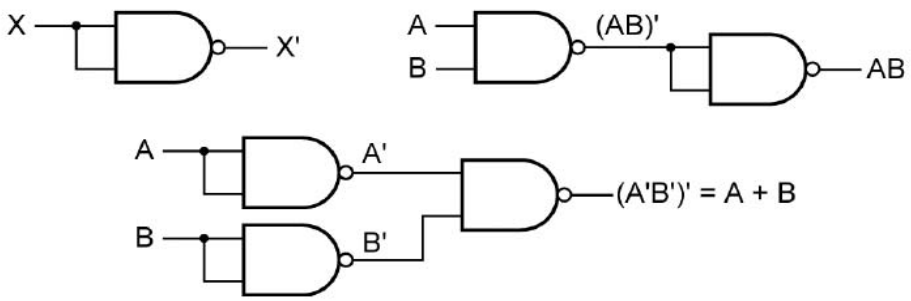
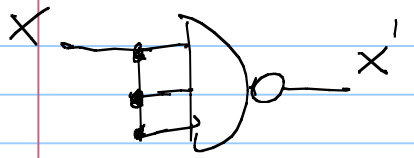


Figure 7-10: **NAND Gate Realization of NOT, AND, and OR**

{NAND} is a complete set of Boolean gates, i.e. this set is sufficient to realize any Boolean circuit.

We know that {AND, OR, NOT} is sufficient.

NAND, as shown in the figure above, can realize NOT, AND, and OR. So, {NAND} is also a complete set of Boolean gates.



An inverter built from a 3-input NAND.

$(x \cdot x \cdot x)' = x'$

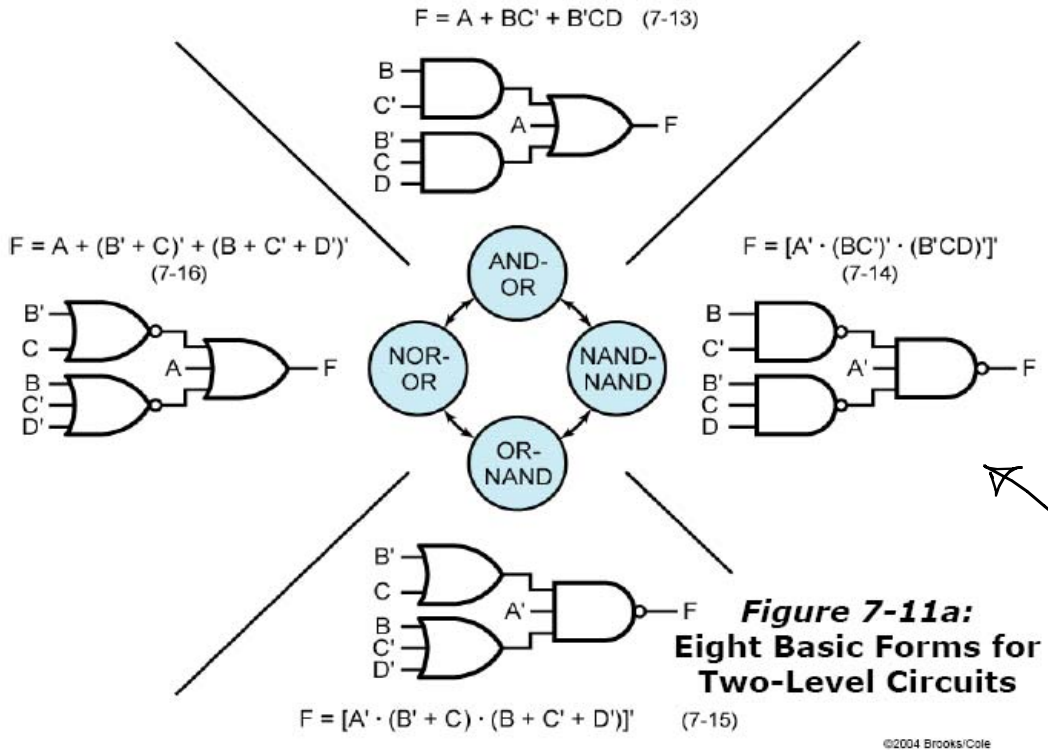


Figure 7-11a:
Eight Basic Forms for Two-Level Circuits

SOP realization on top

Start from SOP

$F = A + BC' + B'CD =$

$[(A + BC' + B'CD)']' =$

$[A' \cdot (BC')' \cdot (B'CD)']' =$

NAND(B, C')

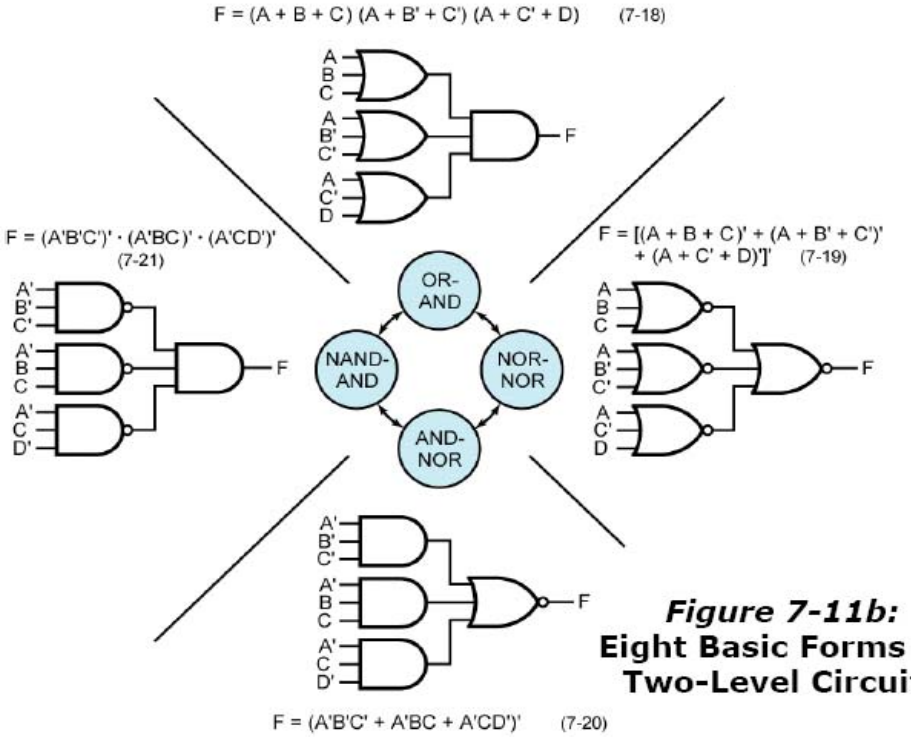
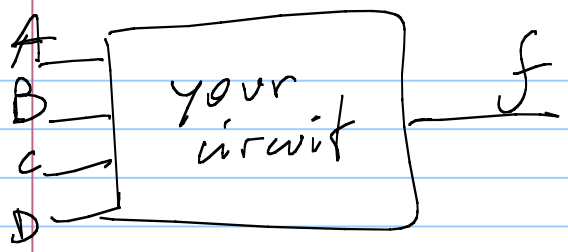


Figure 7-11b:
Eight Basic Forms for Two-Level Circuits

Start with POS (instead of SOP)

Circuit 2 discussion



$f = g(A, B, C, D)$

7404 inverter (note 7405!)

$f(A, B, C, D)$

Vcc

330Ω resistor

LED

74LS05

"f" indicator circuit – treat it as a black box!

(1) Design a minimal SOP (AND-OR) expression for g using a k-map

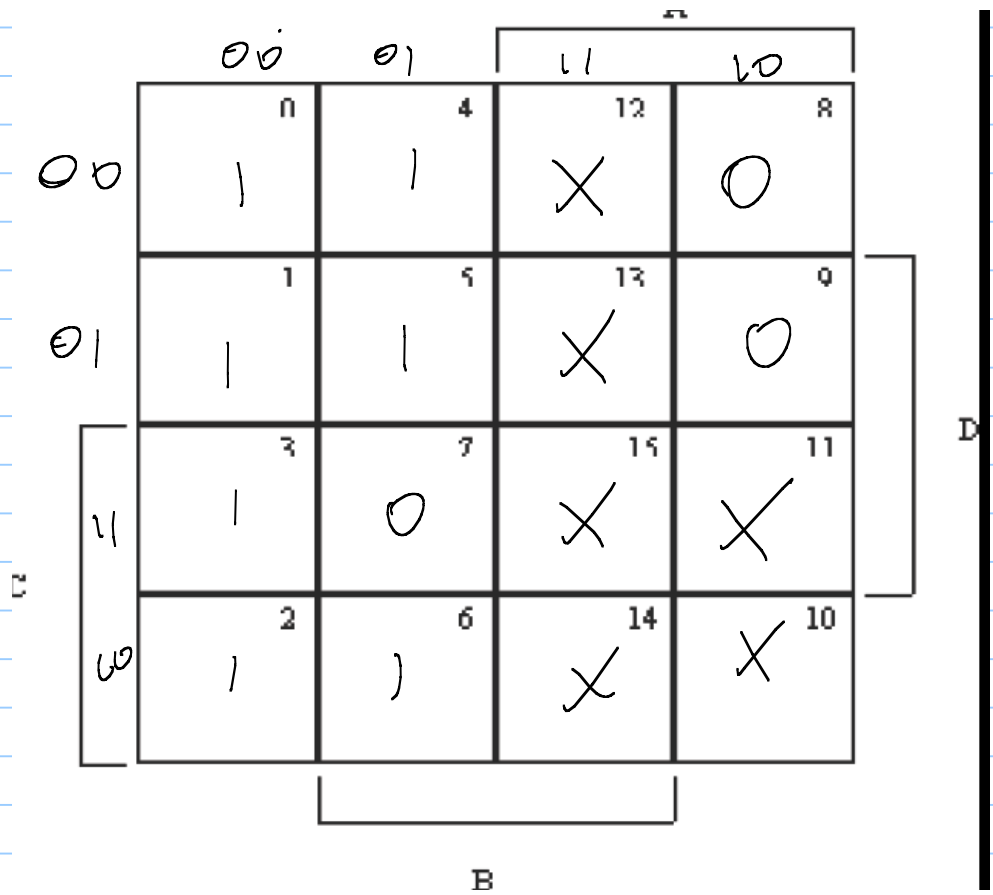
(2) Design the corresponding NAND-NAND circuit

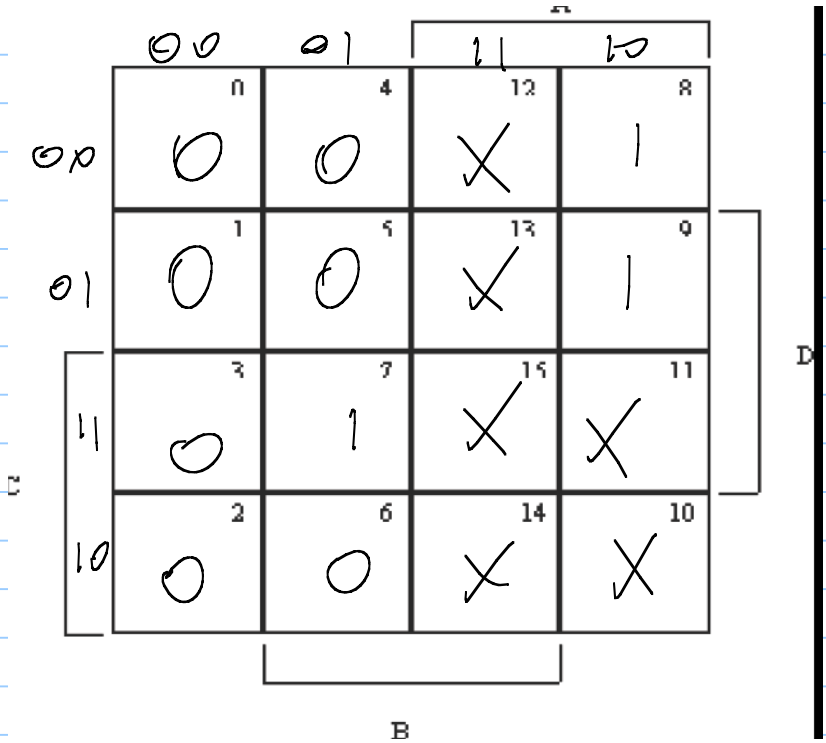
(3) Negate the output with a 7404 inverter

(4) The output of the 7404 inverter will be displayed using the LED indicator circuit.

(3) and (4) are on the previous page

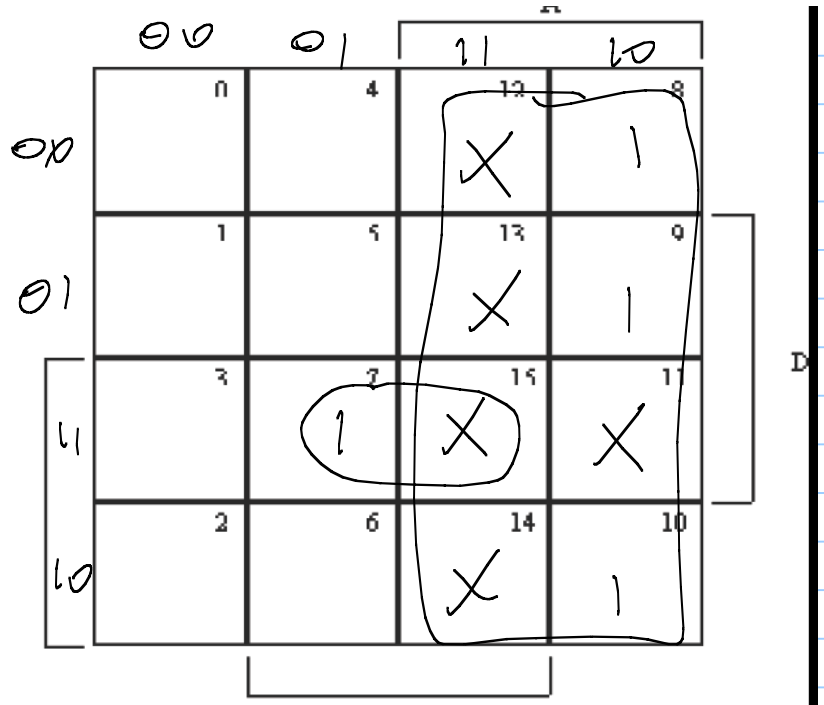
$f(A,B,C,D) = \sum m(0,1,2,3,4,5,6) + \sum d(10,11,12,13,14,15)$





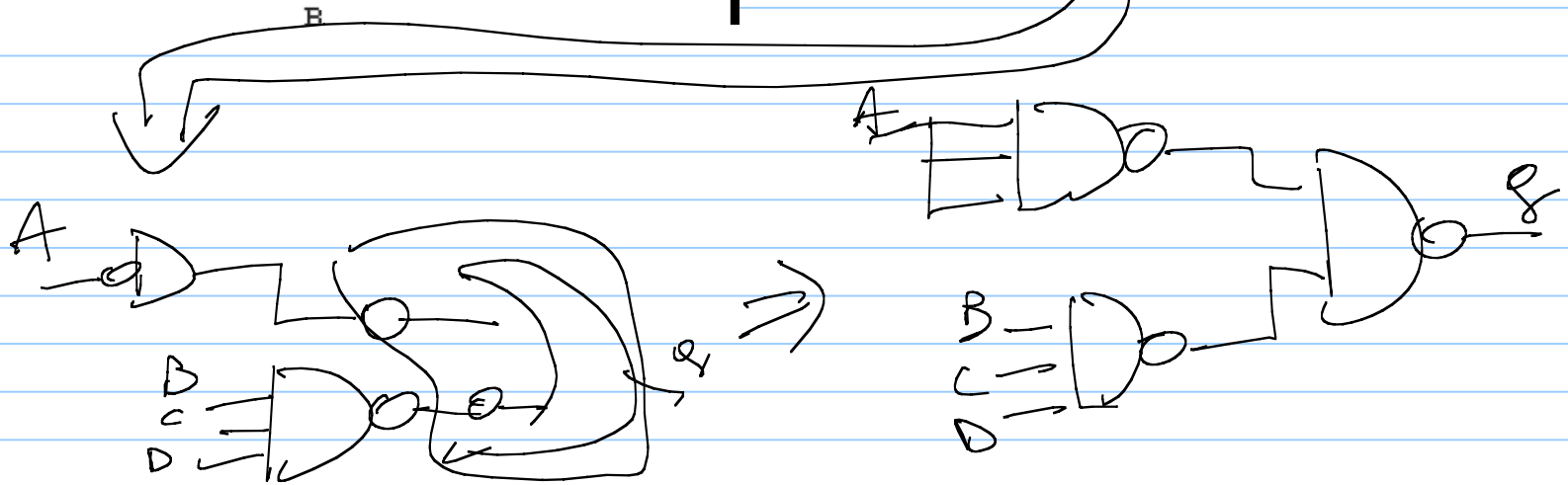
$g(A,B,C,D) = f'(A,B,C,D) :$

$\sum m(7,8,9) + \sum d(10,11,12,13,14,15)$

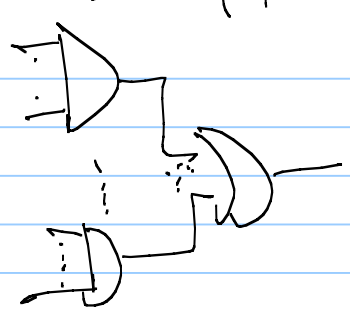


The K-map for g

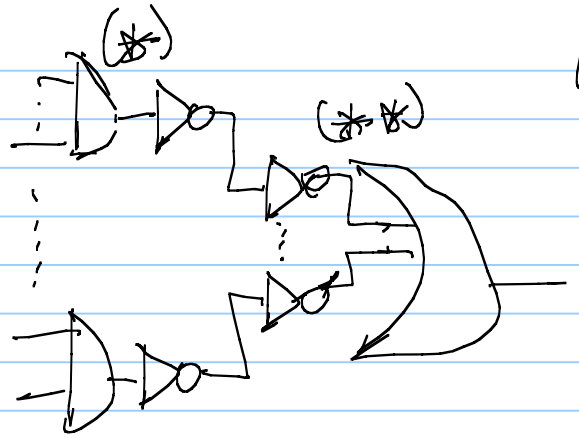
$g = A + BCD$



From SOP (AND-OR) to NAND-NAND

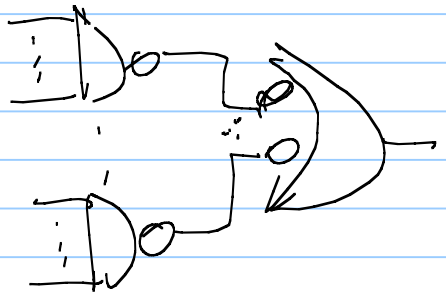


negate both the outputs of the AND gates and the inputs of the OR gate

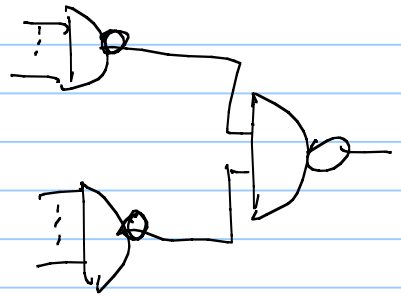


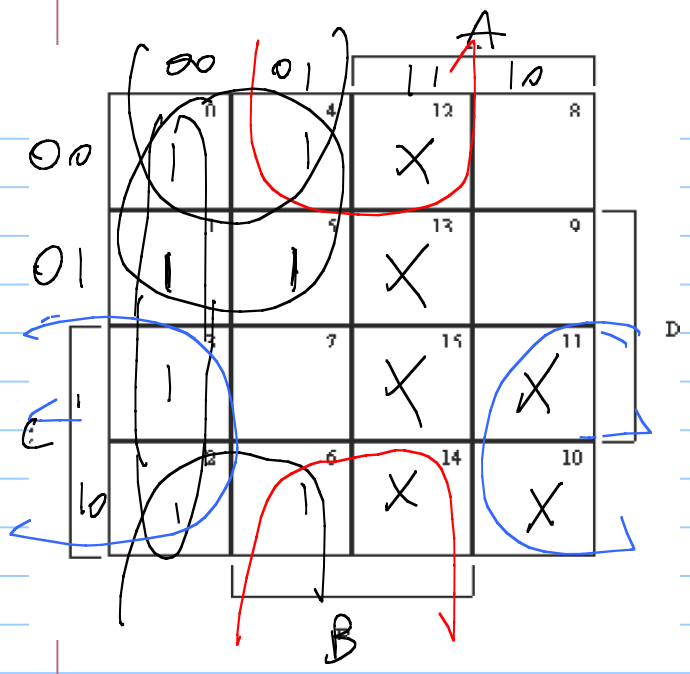
(*) = (**)

notationally equivalent to;



De Morgan's law





$$f(A,B,C,D) = \sum m(0,1,2,3,4,5,6) + \sum d(10,11,12,13,14,15)$$

$$f = A'D' + A'B' + A'C'$$

Check that $g' = f$

$$g' = (A + BCD)' = A' \cdot (BCD)' =$$

$$= A' \cdot (B' + C' + D') = A'B' + A'C' + A'D' = f \quad \checkmark$$

prime implicants:

	m_0	m_1	m_2	m_3	m_4	m_5	m_6
$A'C'$	X	X					
$A'B'$	X	X	X				
$A'D'$	X		X				
BD'							
CB'			X				

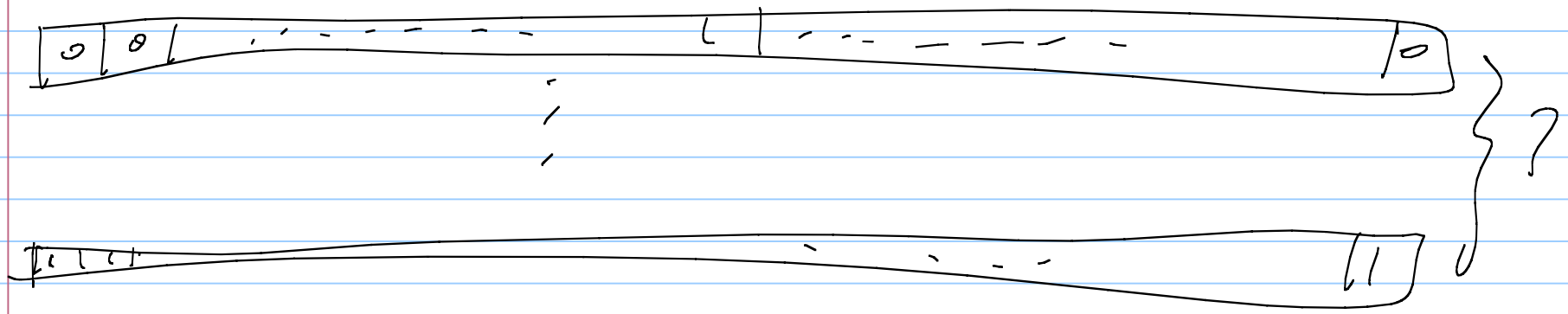
(Quine–McCluskey algorithm from ch. 6)

none of the 1s is covered by only one implicant! So there are no essential prime implicants

How many configuration of 4 bit. vars? 16

How many Boolean functions of 4 binary variables?

0 or 1



A: 2^{16}