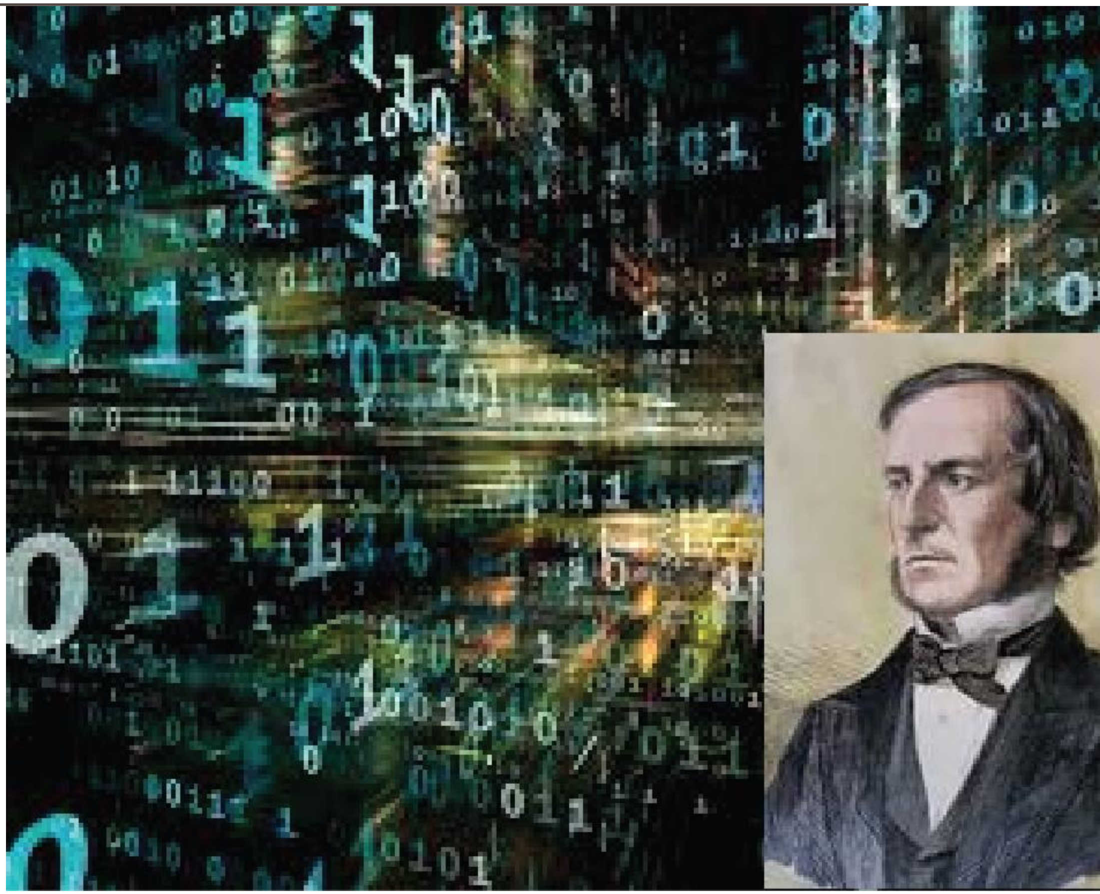
DIGITAL CIRCUITS

Week-4, Lecture-2
Boolean Algebra

Sneh Saurabh
24th August, 2018



Digital Circuits: Announcements/Revision

Digital Circuits

Boolean Algebra

CSOP and CPOS: function and complement

- For complement: take the missing terms

CSOP

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \Sigma m(1,3,4,7)$$

$$\boldsymbol{f}'(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \Sigma m(0,2,5,6)$$

- CSOP to CPOS or vice-versa: take the missing terms

CPOS

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \Sigma m(1,3,4,7) \quad \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \Pi M(0,2,5,6)$$

$$\boldsymbol{f}'(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \Pi M(1,3,4,7)$$

CSOP

$$\mathbf{F}(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) = \Sigma m(1,2,5,7,10,15)$$

$$\mathbf{F}'(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) = \Sigma m(0,3,4,6,8,9,11,12,13,14)$$

CSOP

$$\mathbf{F}(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) = \mathit{\Pi} M(0,3,4,6,8,9,11,12,13,14)$$

$$\mathbf{F}'(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) = \mathit{\Pi} M(1,2,5,7,10,15)$$

- From CSOP to CPOS and vice-versa can be done
- From function in CSOP/CPOS to its complement

SOP to CSOP

Method-1

- Write the truth-table from SOP
- Derive the CSOP from the truth-table

Method-2

- If a variable A is missing in the product term multiply by $(A + A')$
- Expand the expression
- Eliminate the terms occurring more than once
- Put the function in the form required

Find CSOP for the following SOP:

$$\boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = a + b'c$$

Solution:

$$\boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = a(b + b')(c + c') + (a + a')b'c$$

$$= abc + \boldsymbol{ab'c} + abc' + ab'c' + \boldsymbol{ab'c} + a'b'c$$

$$= abc + \boldsymbol{ab'c} + abc' + ab'c' + a'b'c$$

$$\boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = \Sigma m(7,5,6,4,1) = \Sigma m(1,4,5,6,7)$$

SOP to CPOS

Method-1

- Write the truth-table from SOP
- Derive the CPOS from the truth-table

Find CPOS for the following SOP:

$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = xy + x'z$

Solution: $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = xy + x'z = (xy + x')(xy + z)$

$= (x' + y)(x + z)(y + z)$

$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = (x' + y + zz')(x + z + yy')(y + z + xx')$

$= (\boldsymbol{x'} + \boldsymbol{y} + \boldsymbol{z})(x' + y + z')(x + z + y)(x + z + y')(y + z + x)(\boldsymbol{y} + \boldsymbol{z} + \boldsymbol{x'})$

$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \Pi M(4,5,0,2) = \Pi M(0,2,4,5)$

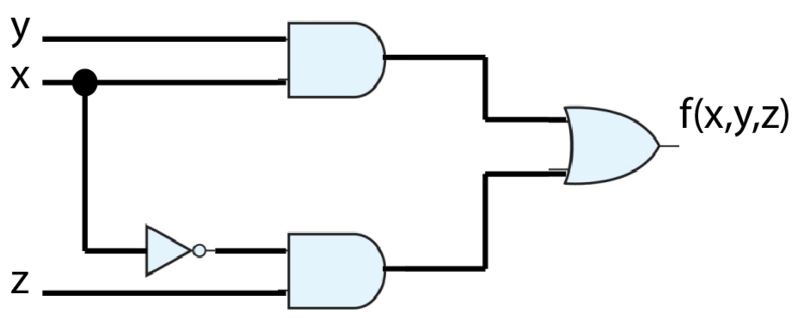
Method-2

- Use the distributive law to represent a function as POS:
  - $A + BC = (A + B)(A + C)$
- If a variable A is missing in the sum term add a term (AA') and expand using distributive law
- Eliminate the terms occurring more than once
- Put the function in the form required

AND-OR and OR-AND implementation

- CSOP and CPOS may not yield minimized implementation (minimum number of gates)
- POS and SOP can be used to realize AND-OR or OR-AND implementations

Implement the following function using 2-inputs AND-OR: $\boldsymbol{f(x,y,z)} = xy + x'z$

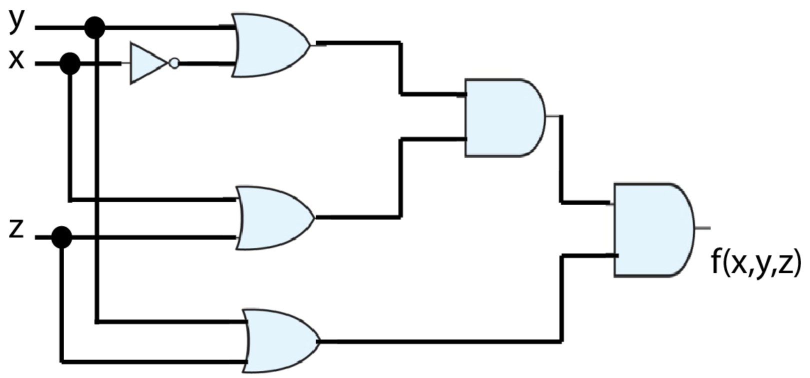


Implement the following function using 2-inputs OR-AND:

$\boldsymbol{f(x,y,z)} = xy + x'z$

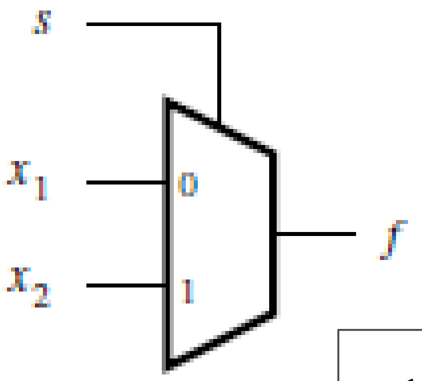
Solution: $\boldsymbol{f(x,y,z)} = xy + x'z = (xy + x')(xy + z)$

$= (x' + y)(x + z)(y + z)$



Different implementations will give different number of literals and gate counts

Multiplexor (for lab)

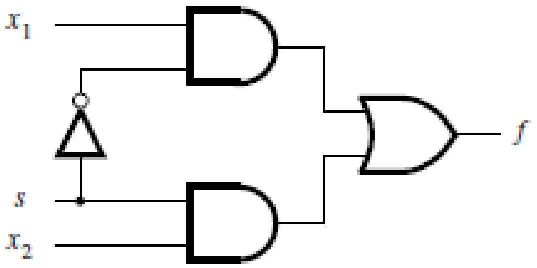


- Output same as the input x_1 if $s = 0$
- Output same as the input x_2 if $s = 1$

$s\ x_1 x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

$$f(s, x_1, x_2) = s'x_1x_2' + s'x_1x_2 + sx_1'x_2 + sx_1x_2$$

$$= s'x_1 + sx_2$$



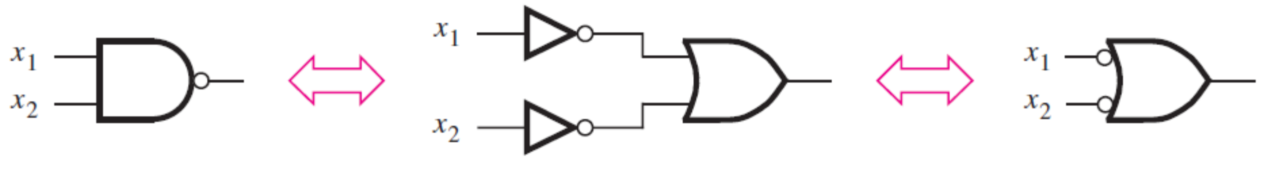
Digital Circuits

Synthesis using NAND/NOR Gates

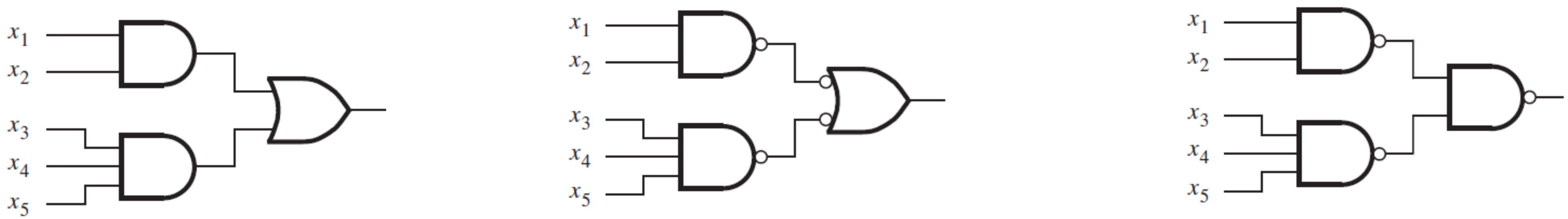
Synthesis: Using NAND/NOR gates

- Any logic function can be implemented in terms of only NAND gates or only NOR gates (universal gates)
- Realizing NAND/NOR gates are easier in terms of transistor than AND/OR
- Applying De Morgan's theorem to realize SOP with NAND gates and POS with NOR gates

Synthesis: SOP using NAND gates (1)



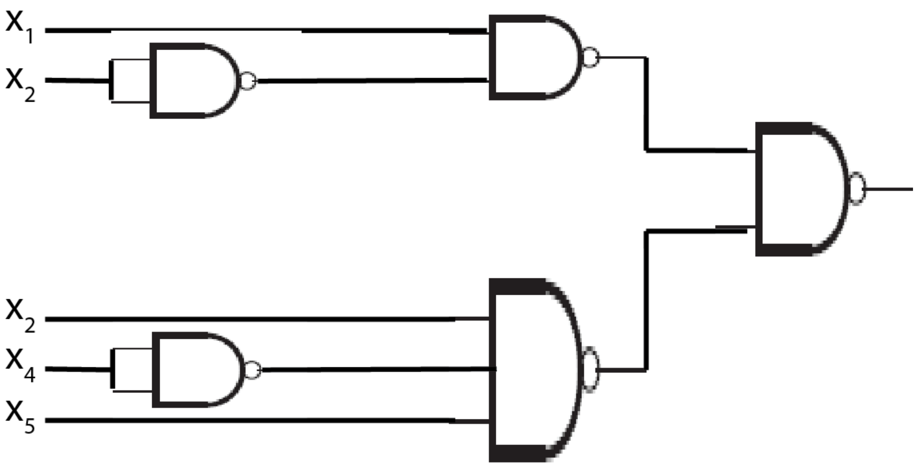
(a) $\overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$



Synthesis: SOP using NAND gates (2)

- Can directly convert an SOP expression to a network of NAND gates

$f(x_1, x_2, x_3, x_4, x_5) = x_1 x_2' + x_2 x_4' x_5$



- For each product term, one NAND gate is used
    - Inputs to the NAND gates are the literals (variable or complemented variables)
    - Complemented variables can be represented with a 2-input NAND gate and both input tied to the same variable (acts as inverter)

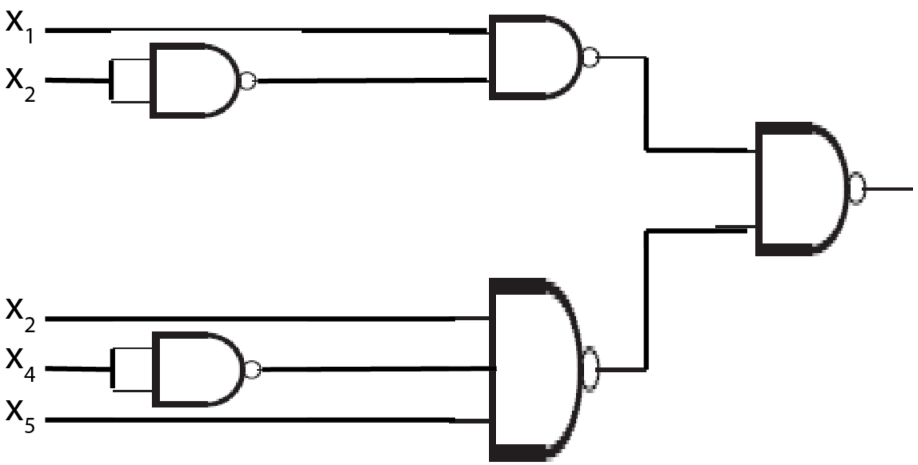
- The outputs of NAND gates representing SOP are given as input to the final NAND gate

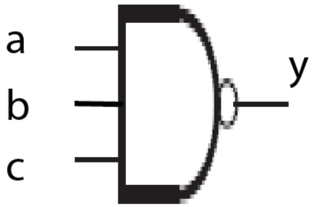
Verify

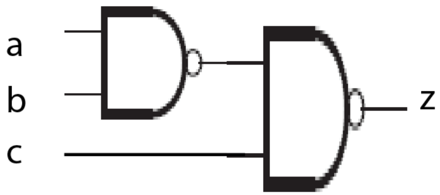
$((x_1 x_2')'.(x_2 x_4' x_5)')'$

$= ((x_1 x_2')')' + ((x_2 x_4' x_5)')' = x_1 x_2' + x_2 x_4' x_5$

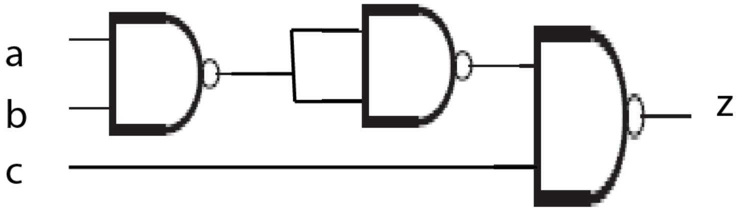
Synthesis: SOP using 2-input NAND gates





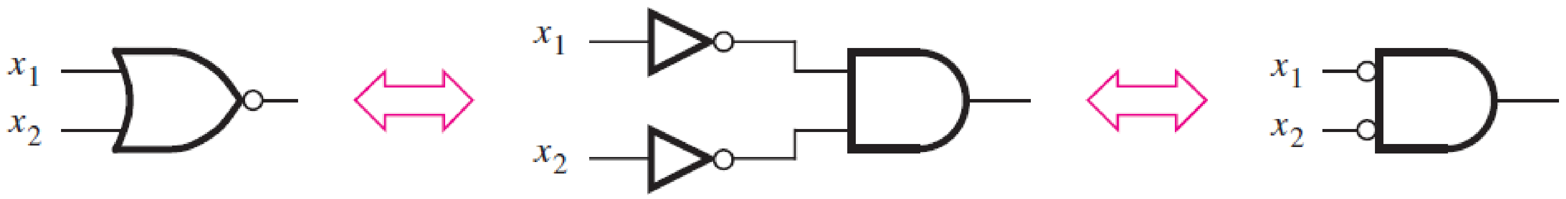


- Wrong
- $z \neq y$

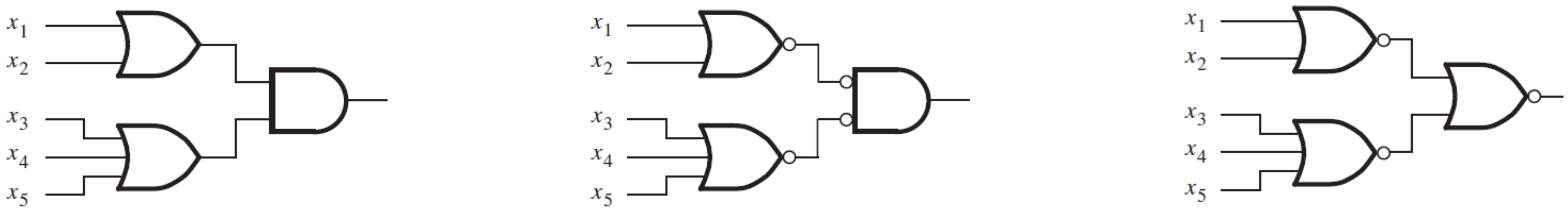


- Correct
- $z = y$

Synthesis: POS using NOR gates



(b) $\overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2$



$$f(x_1, x_2, x_3, x_4, x_5) = (x_1 + x_2)(x_3 + x_4 + x_5)$$

NOR gates: Deriving POS (1)

- To implement a function with NOR, first convert the function in POS
- Different techniques can be used to derive POS

Realize the following function using 2-input NOR gates: $f(a,b,c) = \Sigma m(2,3,4,6,7)$

Method 1:

$f(a,b,c) = \Sigma m(2,3,4,6,7)$

- First minimize SOP:

$= a'bc' + a'bc + ab'c' + abc' + abc$

$= a'b + ab'c' + ab = b + ab'c'$

$= (b+b')(b+ac') = b + ac'$

- Convert to POS:

$f(a,b,c) = (b+a)(b+c')$

Method 2:

$f(a,b,c) = \Sigma m(2,3,4,6,7)$

- Convert to POS:

$= \Pi M(0,1,5)$

$= (a+b+c)(a+b+c')(a'+b+c')$

- Minimize

$= (a+b+c)(a+b+c')(a+b+c')(a'+b+c')$

$= (a+b)(b+c')$

NOR gates: Deriving POS (2)

Realize the following function using 2-input NOR gates: $f(a,b,c) = \Sigma m(2,3,4,6,7)$

Method 3:

$f(a,b,c) = \Sigma m(2,3,4,6,7)$

- First minimize complement of SOP:

$f'(a,b,c) = \Sigma m(0,1,5)$

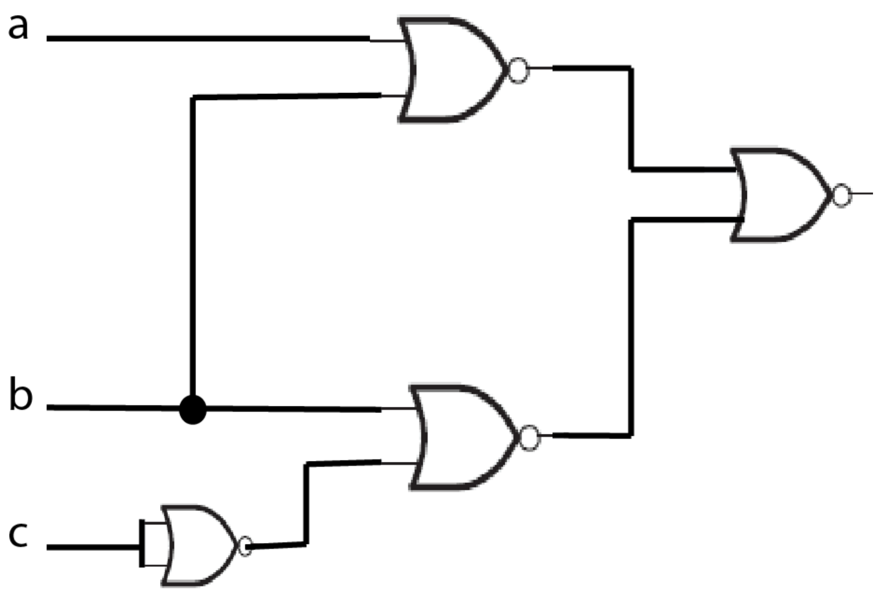
$= a'\,b'c' + a'b'c + ab'c$

$= a'b' + ab'c = b'(a' + ac)$

$= b'(a' + c) = a'b' + b'c$

- Use De Morgan's Law to get POS:

$f(a,b,c) = (a'b' + b'c)' = (a+b)(b+c')$



Digital Circuits

Word Problem to Logic Gate

Word Problem to Logic Network

- Word Problem (Specification) to Boolean expression
- Minimize Boolean expression
- Represent expression using logic network

Problem 1

Assume that a large room has three doors and that a switch near each door controls a light in the room. It has to be possible to turn the light ON or OFF by changing the state of any one of the switches. Design a logic network that controls the light.

Solution:

Detailed specification

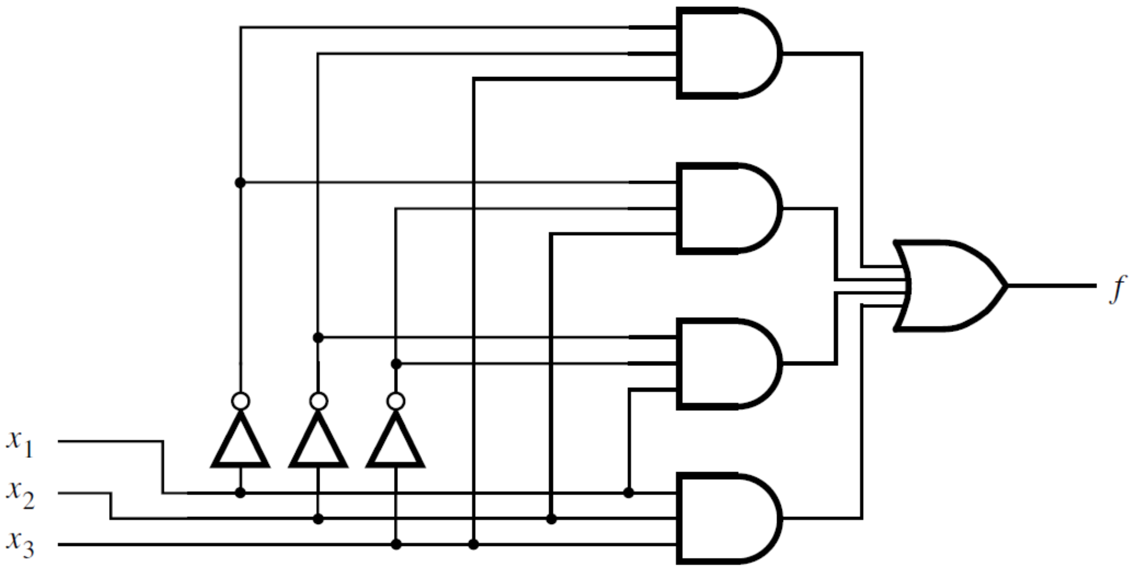
- Suppose the states of the three switches are represented as: x_1, x_2 and x_3. The output of a function $f(x_1, x_2, x_3)$ is taken as the state of the light.
- Assume that light is OFF when all the three switches are OFF, i.e. $x_1 = x_2 = x_3 = 0$
- When only one switch is ON, then the light should be ON
- When only one switch is ON, and then the other switch is turned ON, then the light should turn OFF
- When all three switches are ON, then light should be ON

Problem 1…

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$f(x_1, x_2, x_3) = m_1 + m_2 + m_4 + m_7$$

$$= x_1'x_2'x_3 + x_1'x_2x_3 + x_1\,x_2'x_3' + x_1x_2x_3$$



Practice Problems

1. Problems of Chapter-2 (*Mano and Ciletti*).:
2. Problems of Chapter-2 (*Brown and Vranesic*): Leave sections on CAD, Verilog
3. Using Boolean algebra, show that NAND and NOR gates are not associative
4. Using Boolean algebra, show that XOR gate is associative
5. Draw the truth table of 3-input XOR and XNOR gates