
ECE380 Digital Logic

Optimized Implementation of
Logic Functions:
Multiple Output Circuits, NAND
and NOR Logic Networks

Multiple output circuits

- In all previous examples we have considered only single output functions
- In practice, these functions may be part of some larger circuit that has many such functions
- Circuits that implement these functions may be combined into a less costly single circuit with multiple outputs by sharing some gates needed in the implementation of the single functions

Multiple output circuit example

ab \ cd	00	01	11	10
00			1	1
01		1	1	1
11	1	1		
10	1	1		

$$f_1(a,b,c,d) = ac' + a'c + bc'd$$

cost=4 gates+10 inputs

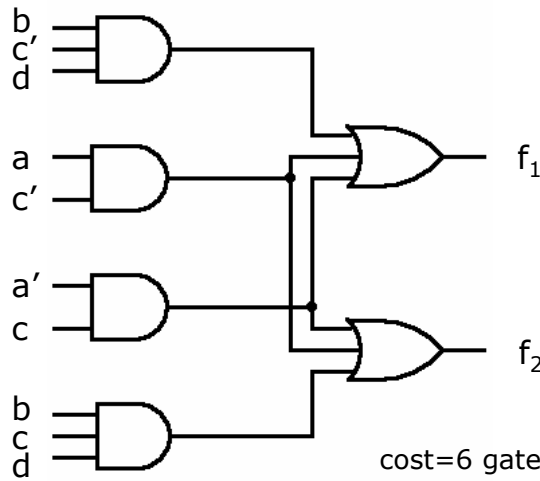
ab \ cd	00	01	11	10
00			1	1
01			1	1
11	1	1		
10	1	1		

$$f_2(a,b,c,d) = ac' + a'c + bcd$$

cost=4 gates+10 inputs

NOTE: cost ignores NOT gates

Multiple output circuit example



cost=6 gates+16 inputs

Multiple output circuit example

- In this case, the minimum combined circuit was derived from the minimum circuit for each function (f_1 and f_2)
- This will not always be the case.
- Consider two functions f_3 and f_4 .

<i>cd</i> \ <i>ab</i>	00	01	11	10
00				
01	1	1	1	
11	1	1	1	
10		1		

Optimal realization of f_3

<i>cd</i> \ <i>ab</i>	00	01	11	10
00				
01	1		1	1
11	1		1	1
10		1		

Optimal realization of f_4

Multiple output circuit example

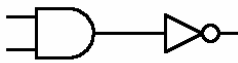
<i>cd</i> \ <i>ab</i>	00	01	11	10
00				
01	1	1	1	
11	1	1	1	
10		1		

<i>cd</i> \ <i>ab</i>	00	01	11	10
00				
01	1		1	1
11	1		1	1
10		1		

Optimal realization of f_3 and f_4 together

NAND and NOR logic networks

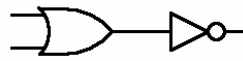
- A **NAND** gate is a functional combination of an AND gate followed by a NOT gate



||

x_1	x_2	$\overline{x_1 \cdot x_2}$
0	0	1
0	1	1
1	0	1
1	1	0

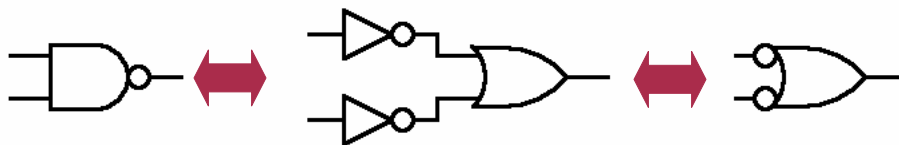
- A **NOR** gate is a functional combination of an OR gate followed by a NOT gate



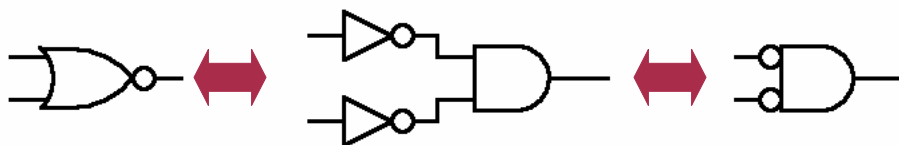
||

x_1	x_2	$\overline{x_1 + x_2}$
0	0	1
0	1	0
1	0	0
1	1	0

DeMorgan's theorem in gate terms



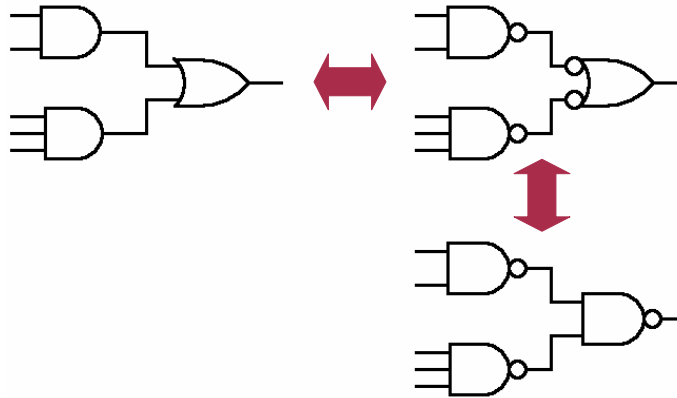
$$(ab)' = a' + b'$$



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

AND-OR and NAND-NAND networks

- If we have a network in AND-OR (SOP) form, we can convert it to a NAND-NAND network



OR-AND and NOR-NOR networks

- If we have a network in OR-AND (POS) form, we can convert it to a NOR-NOR network

