Lecture 6

Finishing up from last time

We discussed using the minterms to write out a function last class.

The short hand we looked at was:

$f(a, b) = m_1 + m_2 + m_3$ or

$f(a, b) = \sum m(1, 2, 3)$

We also need to represent the don’t cares for these functions

$f(a, b) = m_1 + m_2 + m_3 + d_6 + d_{10}$

$f(a, b) = \sum m(1, 2, 3) + \sum d(6, 10)$

How many different functions can 2 inputs represent?

- What does this mean?

Table 2.13 All two-variable functions.

a	b	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 2.14 Number of functions of *n* variables.

Variables	Terms
1	4
2	16
3	256
4	65,536
5	4,294,967,296

Implementations using AND, OR, and NOT Gates

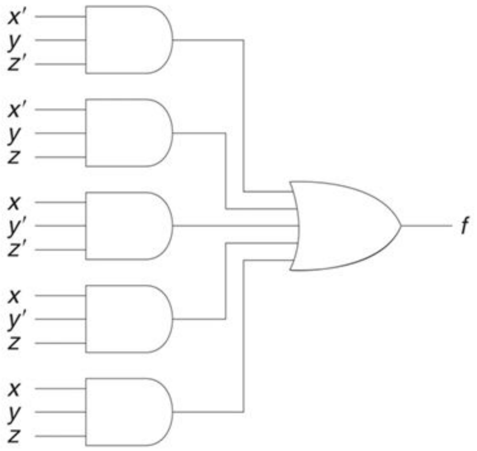
$f = x'yz' + x'yz + xy'z' + xy'z + xyz$

How will this look?

Implementations using AND, OR, and NOT Gates

$f = x'yz' + x'yz + xy'z' + xy'z + xyz$

How will this look?



Implementations using AND, OR, and NOT Gates

Can you draw a diagram for this?

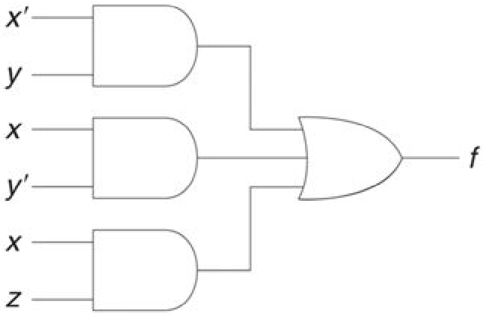
$f = x'y + xy' + xz$

Both this and the previous are 2 level circuits

Implementations using AND, OR, and NOT Gates

Can you draw a diagram for this?

$f = x'y + xy' + xz$



Both this and the previous are 2 level circuits

Implementations using AND, OR, and NOT Gates

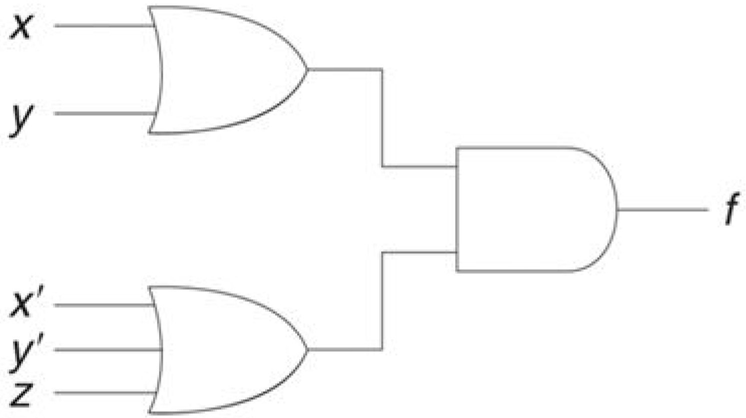
How about this one?

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

Implementations using AND, OR, and NOT Gates

How about this one?

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

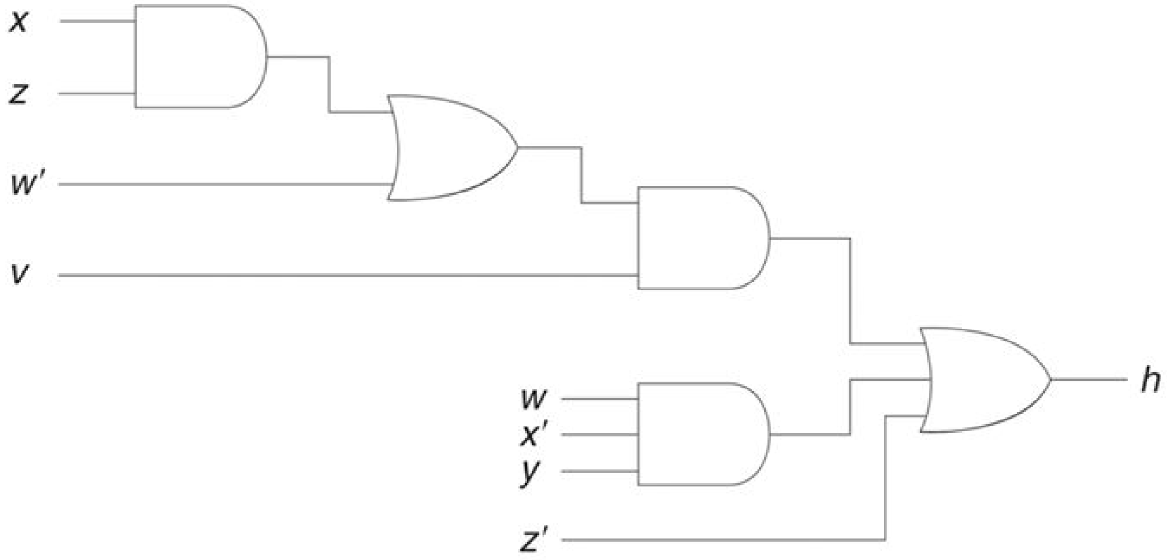


Implementations when not in SOP or POS form

$h = z' + wx'y + v(xz+w')$

Implementations when not in SOP or POS form

$h = z' + wx'y + v(xz+w')$



More types of gates

NAND, NOR, and Exclusive-OR

Why do we use these?

Sometimes needs less gates

Are functionally complete. They can be used to implement AND, OR, and NOT so we need less types of gates

NAND and NOR gates

Figure 2.12 NAND gates.

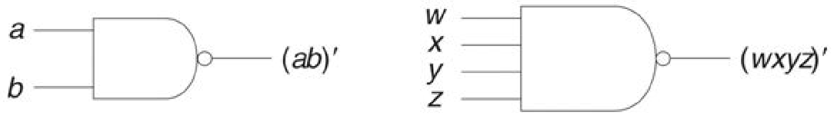


Figure 2.13 Alternative symbol for NAND.

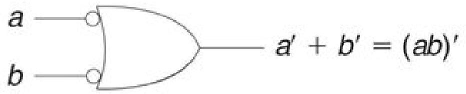
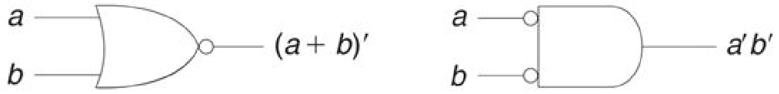
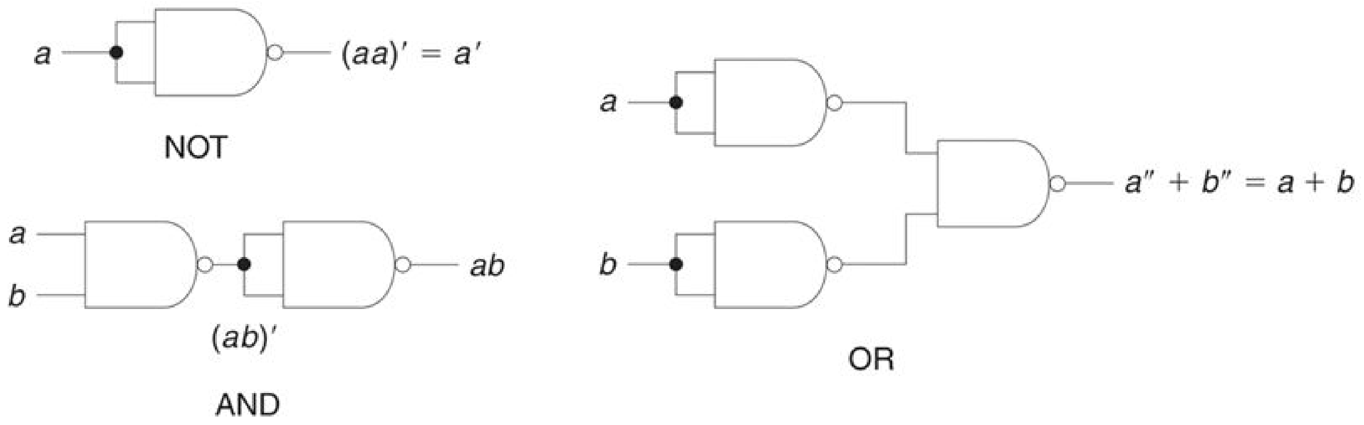


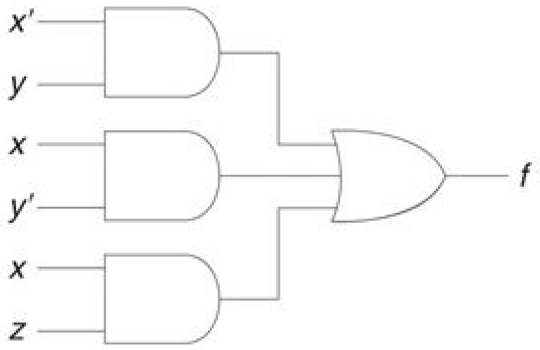
Figure 2.14 Symbols for NOR gate.



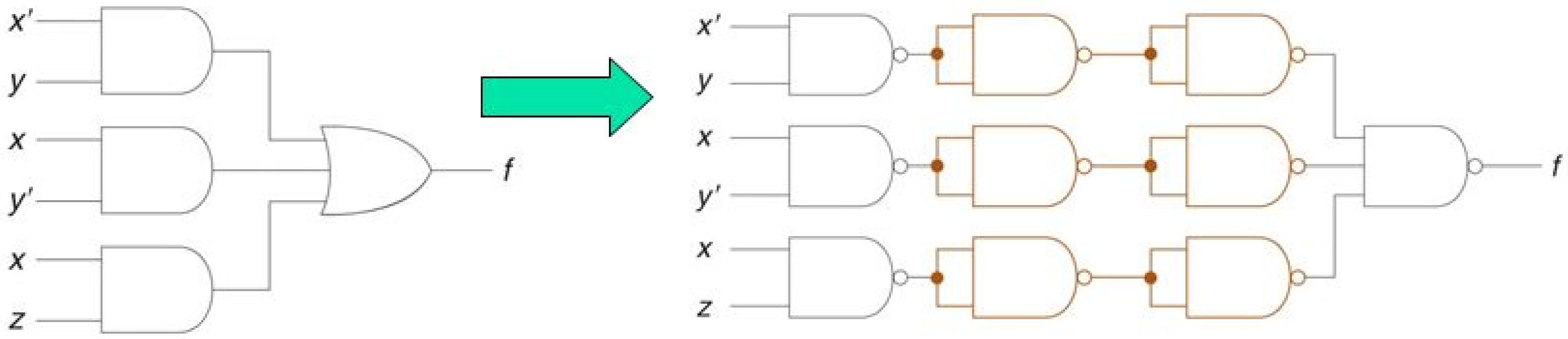
Use NAND to Implement AND, OR, and NOT



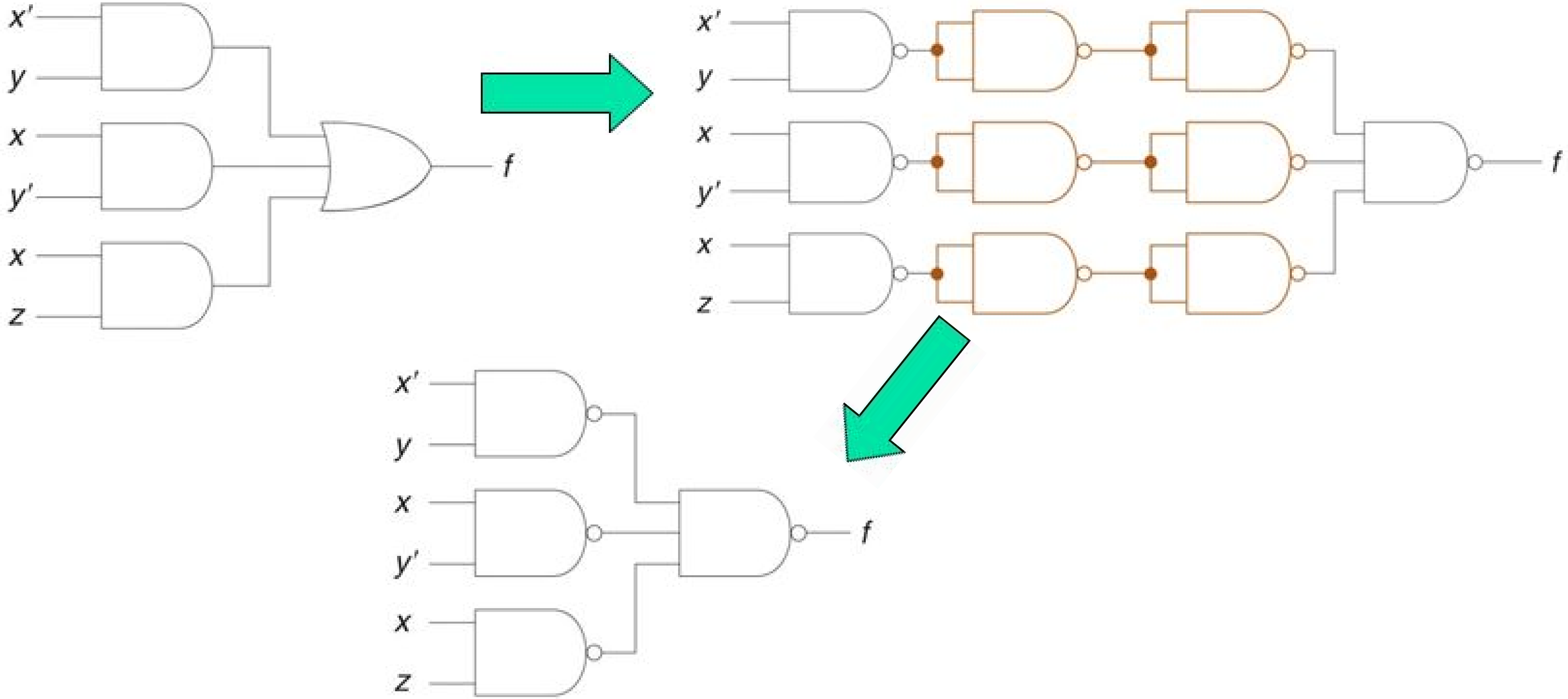
Implementation using NAND



Implementation using NAND

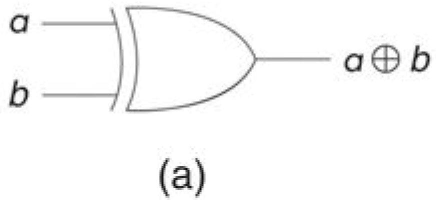


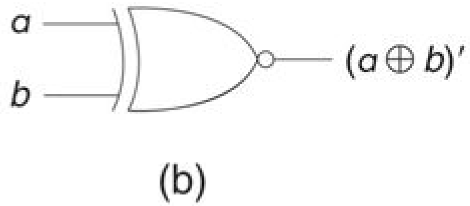
Implementation using NAND



Exclusive-OR and Exclusive-NOR Gates

Figure 2.20 (a) An Exclusive-OR gate. (b) An Exclusive-NOR gate.





Properties or Exclusive-OR

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

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

$$(a \oplus b') = (a \oplus b)'$$

$$a \oplus 0 = a = (a' \cdot 0 + a \cdot 1)$$

$$a \oplus 1 = a' = (a' \cdot 1 + a \cdot 0)$$

$$a \oplus b = b \oplus a$$

$$(a \oplus b) \oplus c = a \oplus (b \oplus c)$$

Simplification of Algebraic Expression

Primary tools:

P9a. $ab + ab' = a$ P9b. $(a + b)(a + b') = a$

P10a. $a + a'b = a + b$ P10b. $a(a' + b) = ab$

Simplification of Algebraic Expression

Other useful properties

P6a. $a + a = a$ P6b. $aa = a$

P8a. $a(b + c) = ab + ac$ P8b. $a+bc = (a+b)(a+c)$

When function isn't in SOP or POS form

Absorption

P12a. $a + ab = a$ P12b. $a(a+b) = a$