Combinational Logic Design

Dr. Chandan Karfa
CSE IIT Guwahati

- NAND and NOR Implementation
    - Chapter 3, Mano’s book

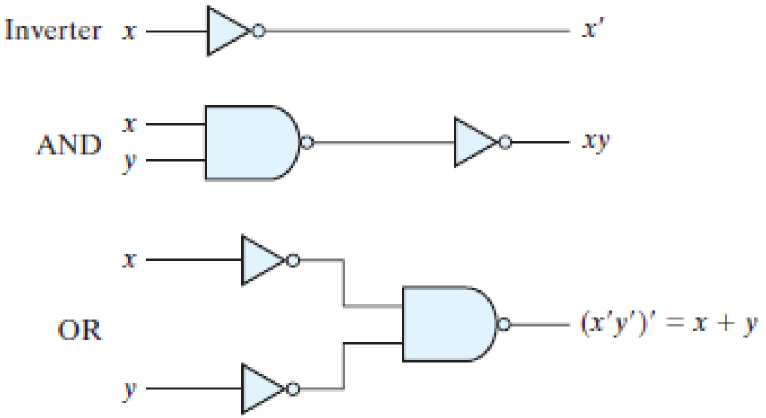
NAND and NOR Implementation

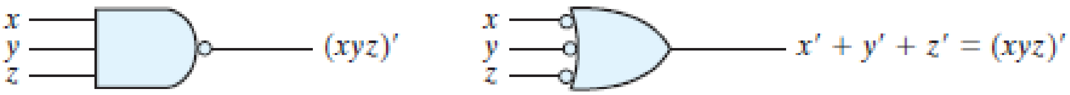
- Digital circuits are frequently constructed with NAND or NOR gates rather than with AND and OR gates
- Both NAND and NOR gates are universal gate.
- NAND and NOR gates are easier to fabricate with electronic components and are the basic gates used in all IC digital logic families
- Conversion from Boolean functions given in terms of AND, OR, and NOT into equivalent NAND and NOR logic diagrams

NAND Implementation

- NAND is a universal gate/functionally complete
- {AND, OR, NOT} is functionally complete
- A convenient way to implement a Boolean function with NAND gates is to obtain the simplified Boolean function (SoP/AND-OR) in terms of Boolean operators and then convert the function to NAND logic.

NAND Implementation





(a) AND-invert (b) Invert-OR

FIGURE 3.17
Two graphic symbols for a three-input NAND gate

Two-Level Implementation

- **The implementation of Boolean functions with NAND gates requires that the functions be in sum-of-products form**

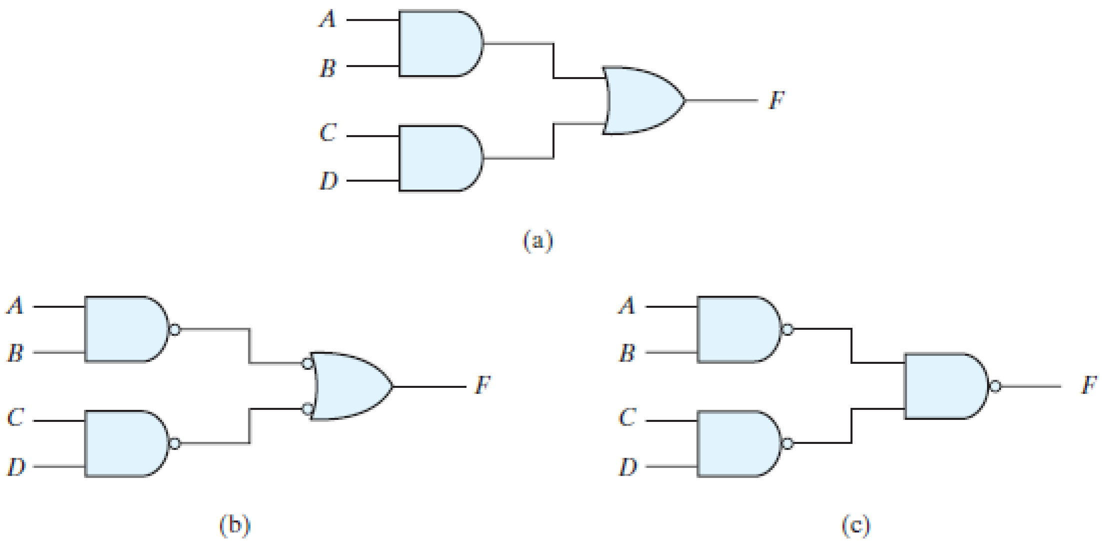


FIGURE 3.18
Three ways to implement $F = AB + CD$

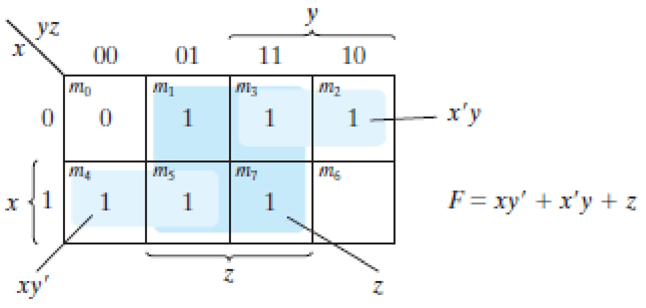
$$F = ((AB)'(CD)')' = AB + CD$$

Two-Level Implementation

- Simplify the function and express it in sum-of-products form.
- Draw a NAND gate for each product term of the expression that has at least two literals. The inputs to each NAND gate are the literals of the term. This procedure produces a group of first-level gates.
- Draw a single gate using the AND-invert or the invert-OR graphic symbol in the second level, with inputs coming from outputs of first-level gates.
- A term with a single literal requires an inverter in the first level. However, if the single literal is complemented, it can be connected directly to an input of the second level NAND gate.

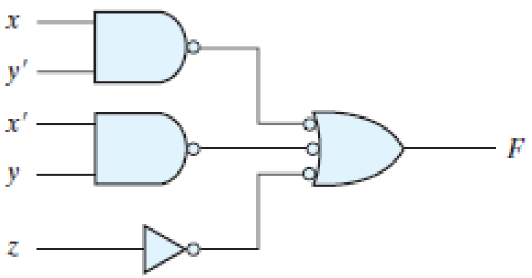
Two-Level Implementation

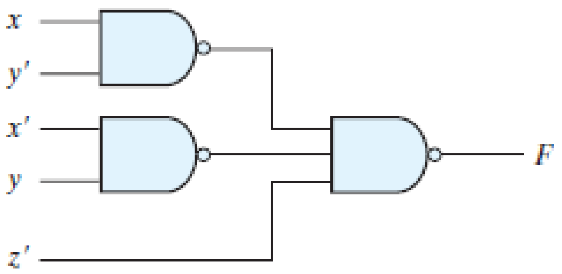
$F(x, y, z) = (1, 2, 3, 4, 5, 7)$



(a)

$$F = xy' + x'y + z$$



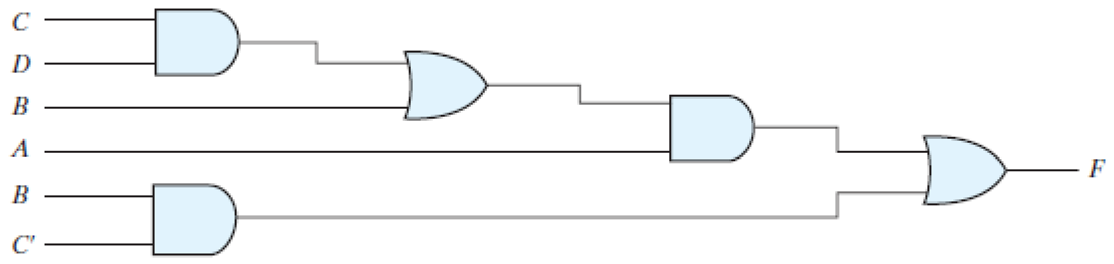


Multilevel NAND Circuits

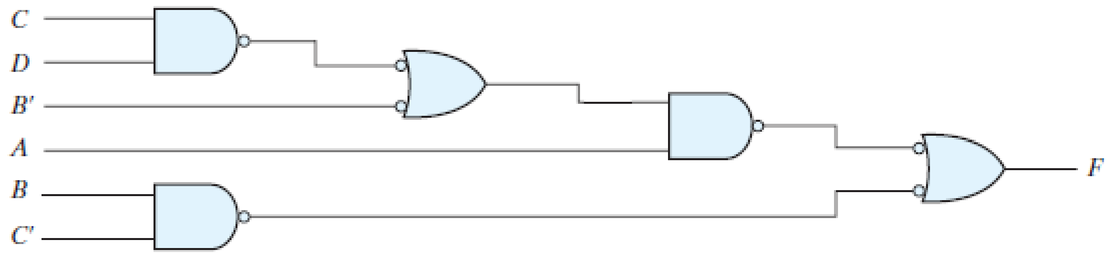
- The design of multilevel circuits is to express the Boolean function in terms of AND, OR, and complement operations.
- The function can then be implemented with AND and OR gates. After that, convert into an all-NAND circuit

Multilevel NAND Circuits

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



(a) AND–OR gates



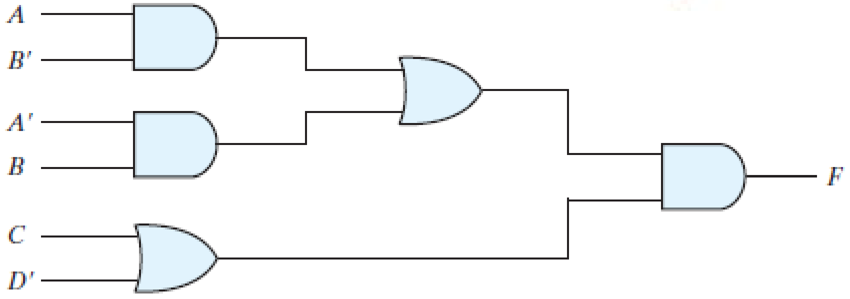
(b) NAND gates

Multilevel AND–OR diagram into an all-NAND

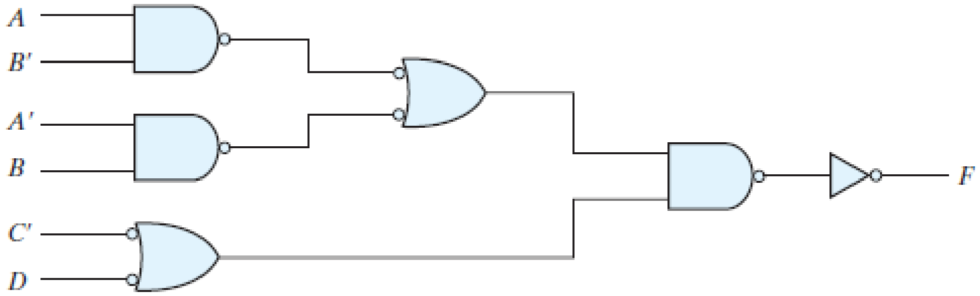
- Convert all AND gates to NAND gates with AND-invert graphic symbols.
- Convert all OR gates to NAND gates with invert-OR graphic symbols.
- Check all the bubbles in the diagram. For every bubble that is not compensated by another small circle along the same line, insert an inverter (a one-input NAND gate) or complement the input literal.

Multilevel AND–OR diagram into an all-NAND

Implementing $F = (AB' + A'B)(C + D')$



(a) AND–OR gates

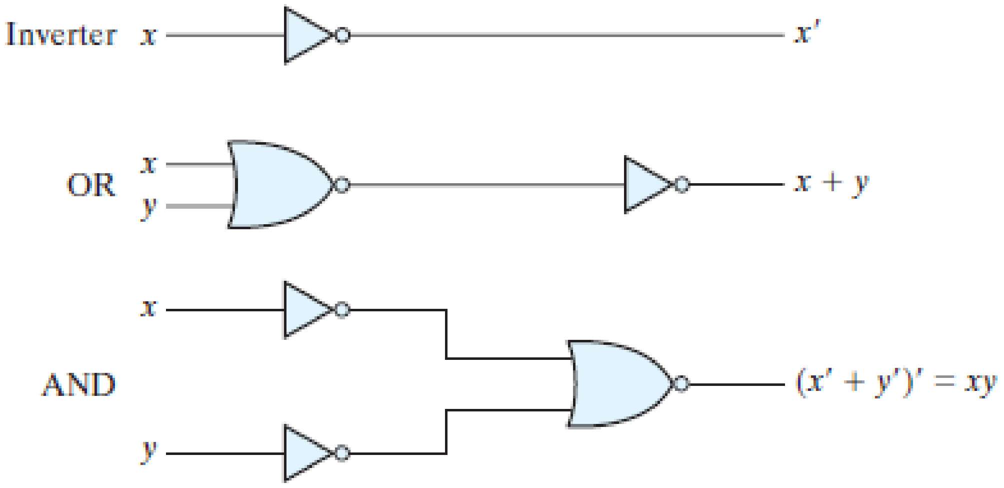


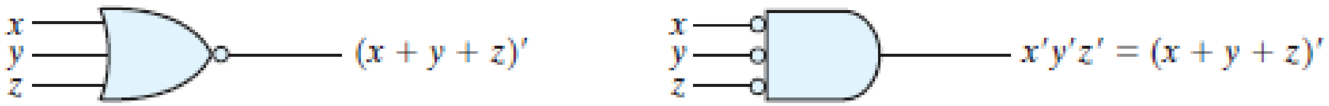
(b) NAND gates

NOR Implementation

- The NOR gate is another universal gate that can be used to implement any Boolean function.
- The NOR operation is the dual of the NAND operation.
- Therefore, all procedures and rules for NOR logic are the duals of the corresponding procedures and rules developed for NAND logic.

NOR Implementation





(a) OR-invert (b) Invert-AND

Two-level implementation with NOR

- A two-level implementation with NOR gates requires that the function be simplified into product-of-sums form

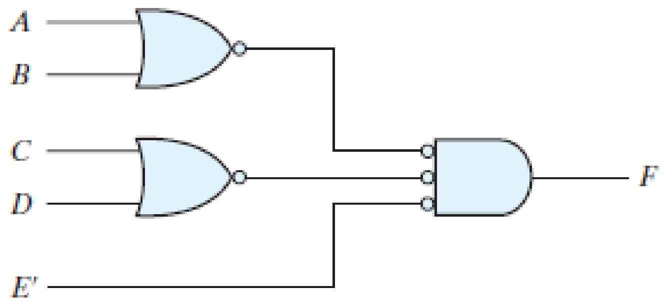


FIGURE 3.24
Implementing $F = (A + B)(C + D)E$

Multilevel Logic Implementation with NOR

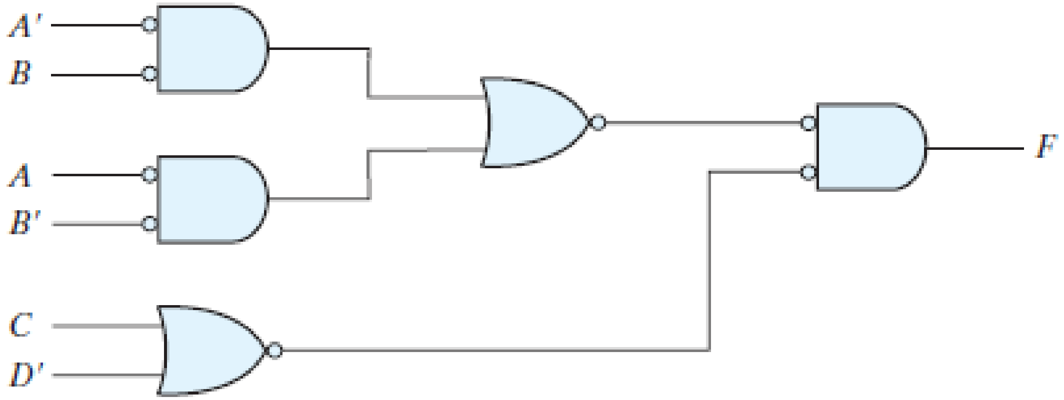


FIGURE 3.25
Implementing $F = (AB' + A'B)(C + D')$ with NOR gates