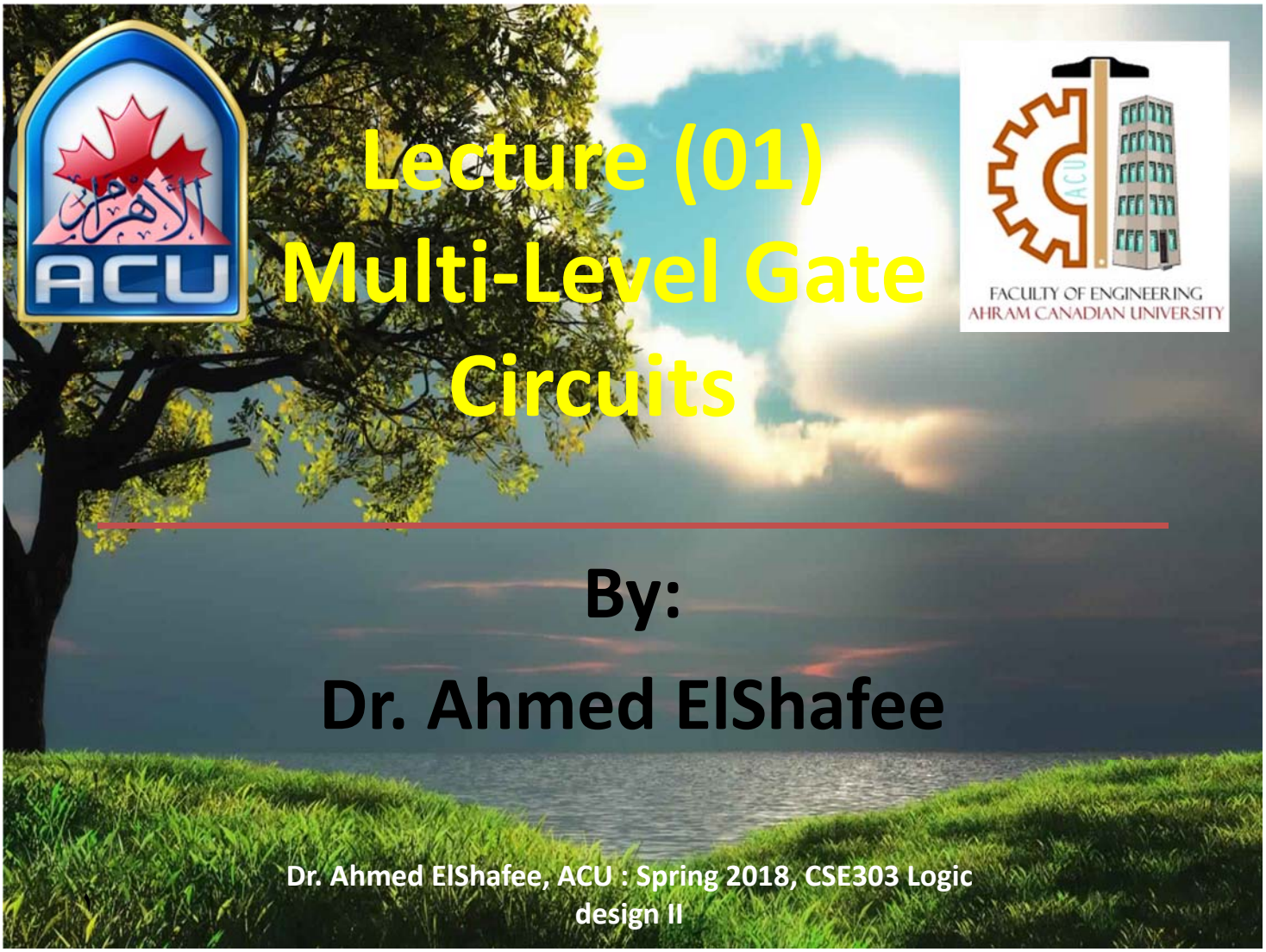


Multi-Level Gate Circuits

- In this lecture, we will use the following terminology:

1. *AND-OR circuit* means a two-level circuit composed of a level of AND gates followed by an OR gate at the output.

- Sum-of-products(SOP): AB'+ACD+ABC'……

2. *OR-AND circuit* means a two-level circuit composed of a level of OR gates followed by an AND gate at the output.

- Product-of-sums(POS): (A+B')(A+C+D)(A+B+C')……

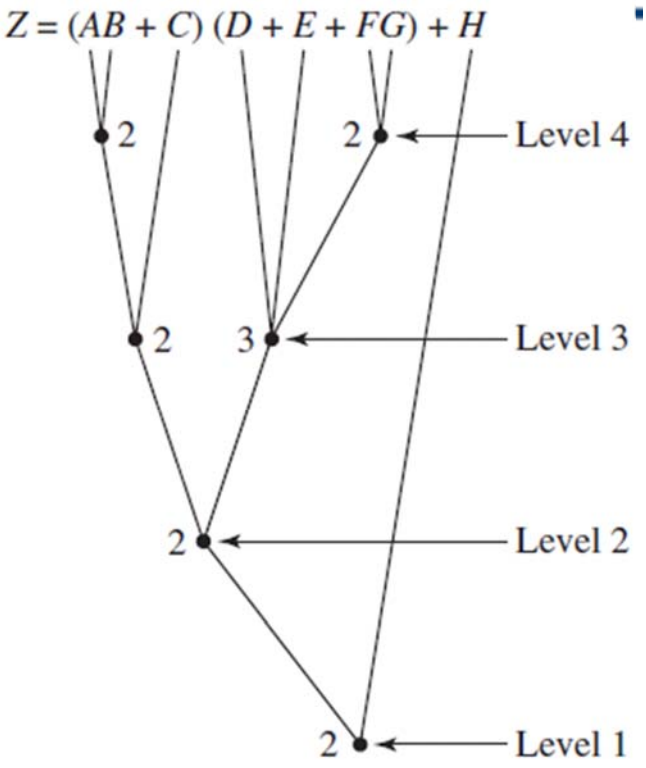
3. *OR-AND-OR circuit* means a three-level circuit composed of a level of OR gates followed by a level of AND gates followed by an OR gate at the output.
4. Circuit of AND and OR gates implies no particular ordering of the gates; the output gate may be either AND or OR.

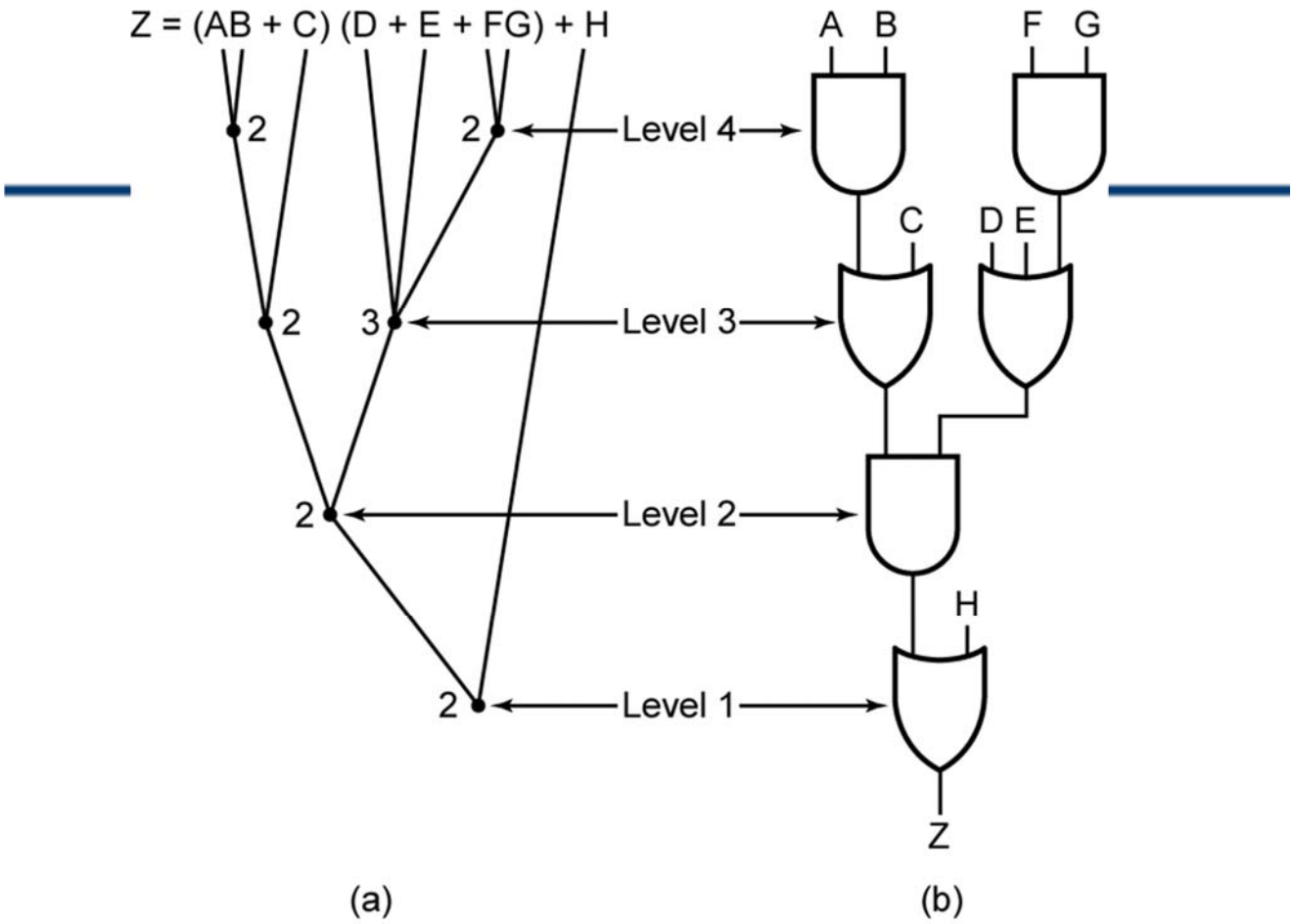
- What's the concerned issues from a logic designers:

1. Number of gates
2. Gate inputs
3. Level of a circuit
   - a. Cascaded circuit will increase the gate-delay
   - b. and also the cost.
   - c. gate-delay will slow down the operation of a digital system

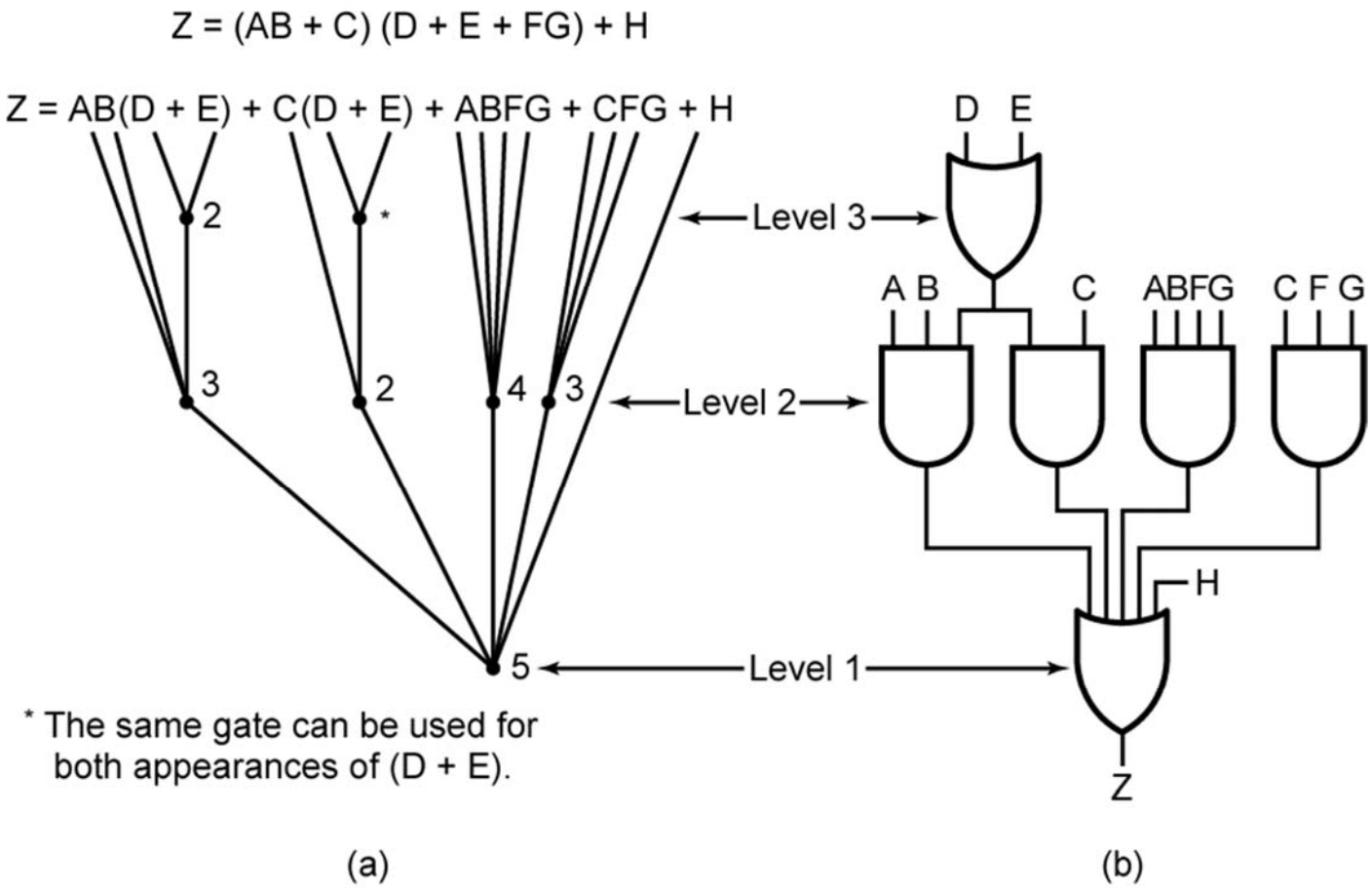
Tree Diagrams

- Each node on a tree diagram represents a gate, and the number of gate inputs is written beside each node.





4 level, 6 gates, 13 gate inputs



3 level, 6 gates, 19 gate inputs

Example

Find a circuit of AND and OR gates to realize

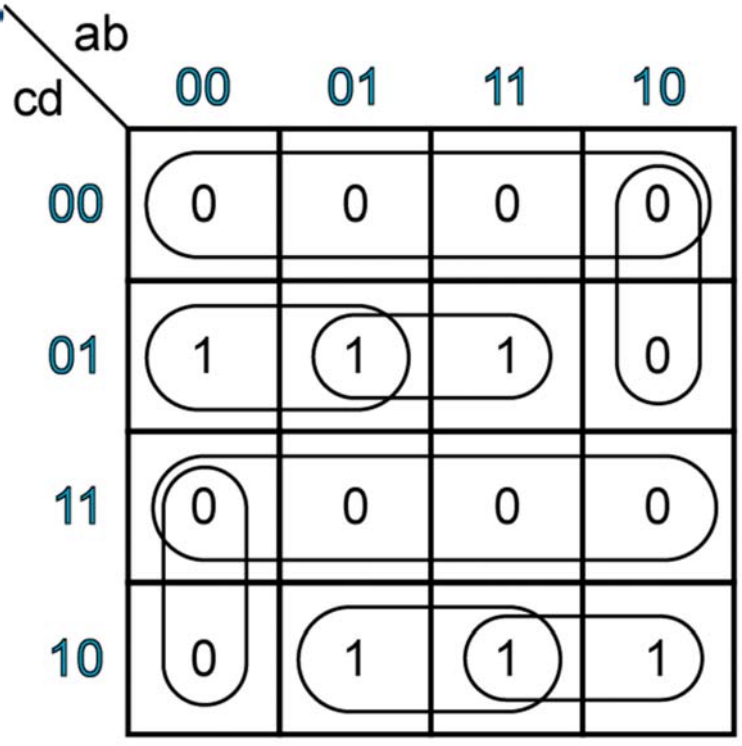
$f(a, b, c, d) = \Sigma\, m(1, 5, 6, 10, 13, 14)$

Consider solutions with two levels of gates and three levels of gates. Try to minimize the number of gates and the total number of gate inputs. Assume that all variables and their complements are available as inputs.

OR gate at the output

$f(a, b, c, d) = \Sigma\, m(1, 5, 6, 10, 13, 14)$

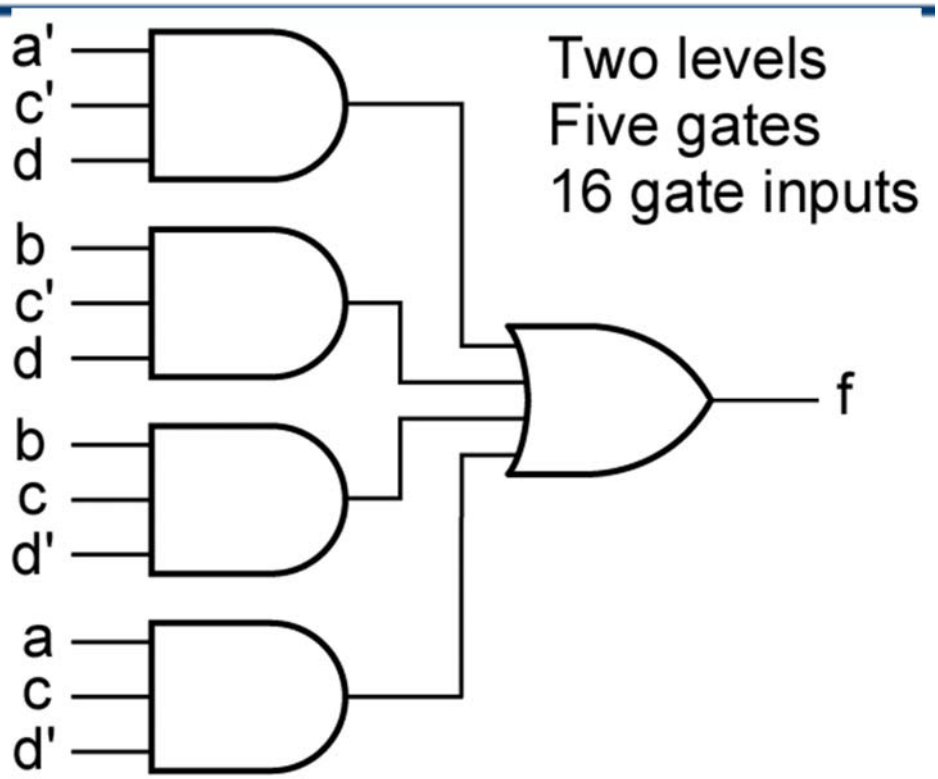
- First, simplify *f* by using a Karnaugh map.



f = a'c'd + bc'd + bcd' + acd'

$$f = a'c'd + bc'd + bcd' + acd'$$

- This leads directly to a two-level AND-OR gate circuit.

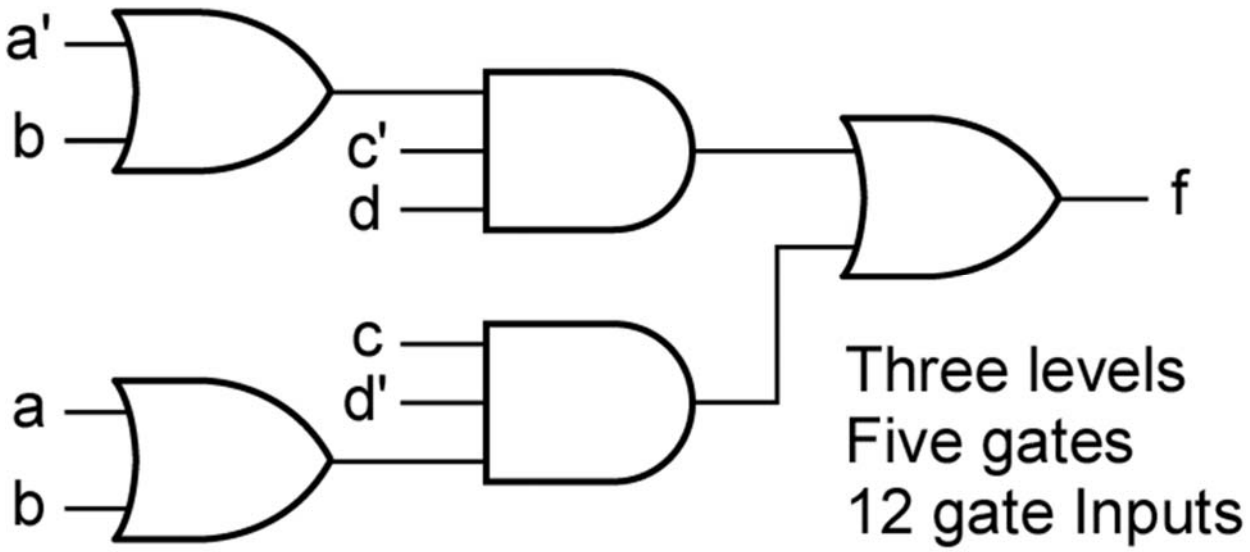


$$f = a'c'd + bc'd + bcd' + acd'$$

Factoring yields

$$f = c'd(a' + b) + cd'(a + b)$$

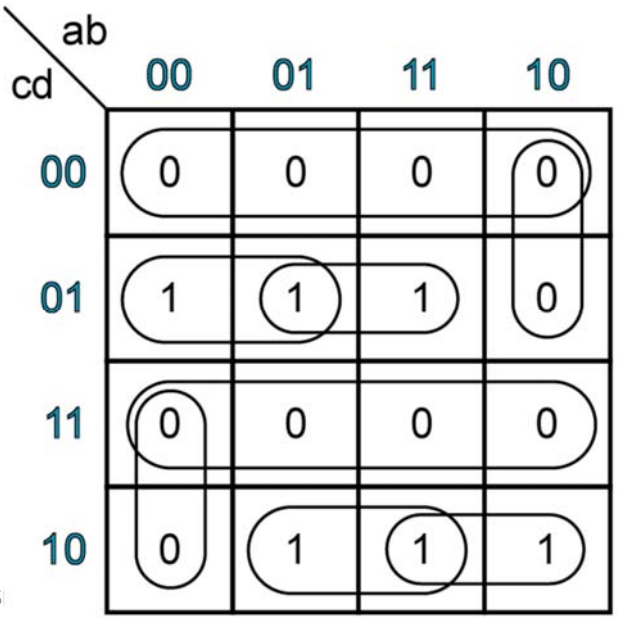
-



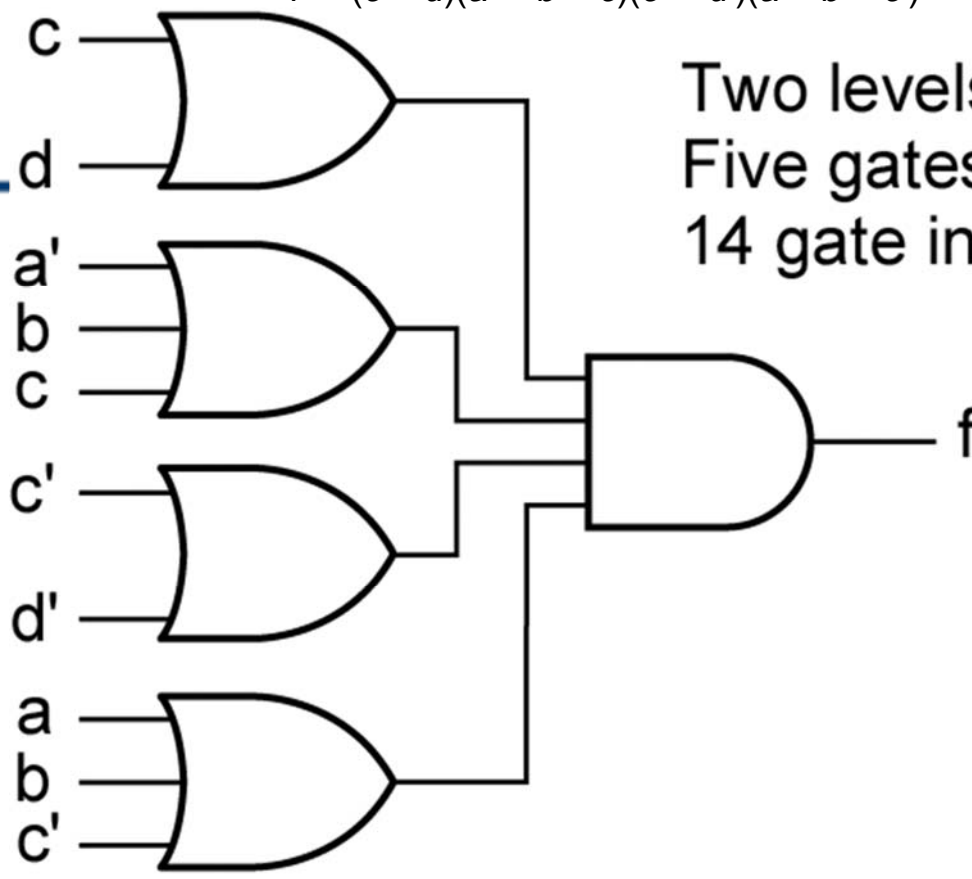
- Both of these solutions have an OR gate at the output. A solution with an AND gate at the output might have fewer gates or gate inputs. A two-level OR-AND circuit corresponds to a product-of-sums expression for the function. This can be obtained from the 0′s on the Karnaugh map as follows:

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

leads directly to a two-level OR-AND circuit



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



Two levels
Five gates
14 gate inputs

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

To get a three-level circuit with an AND gate output, we partially multiply out Equation using $(X + Y)(X + Z) = X + YZ$:

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

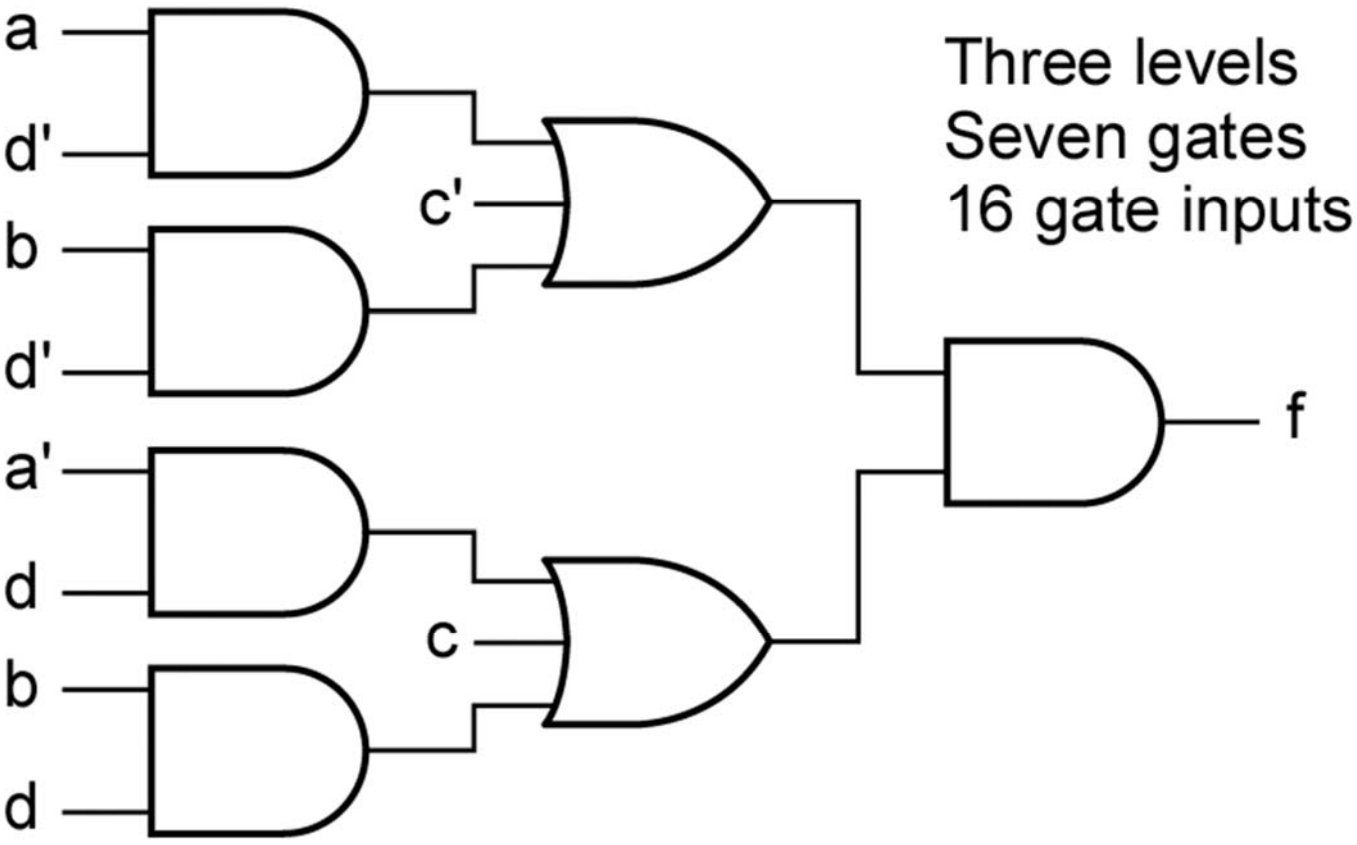
$f = [c + d(a' + b)][c' + d'(a + b)]$

Equation would require four levels of gates to realize; however, if we multiply out $d'(a + b)$ and $d(a' + b)$, we get

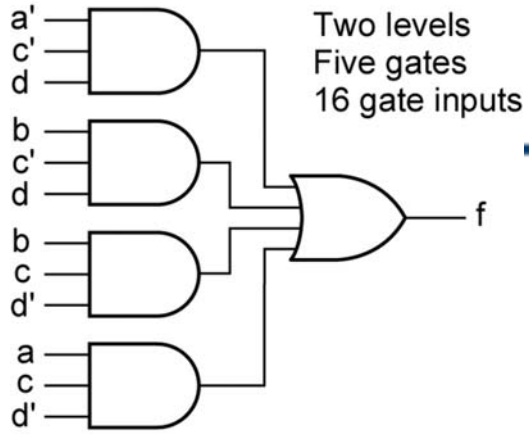
$f = (c + a'd + bd)(c' + ad' + bd')$

which leads directly to a three-level AND-OR-AND circuit.

$f = (c + a'd + bd)(c' + ad' + bd')$

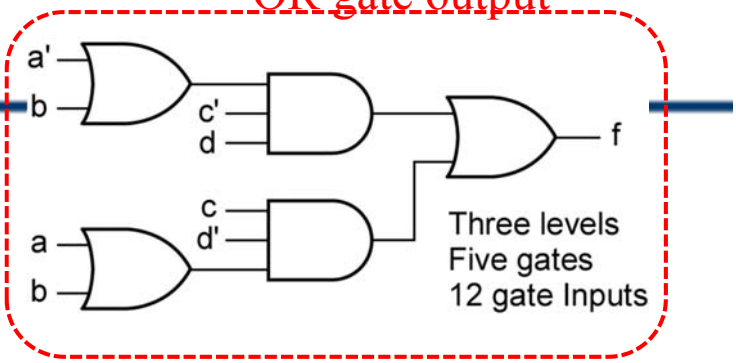


f = a'c'd + bc'd + bcd' + acd'



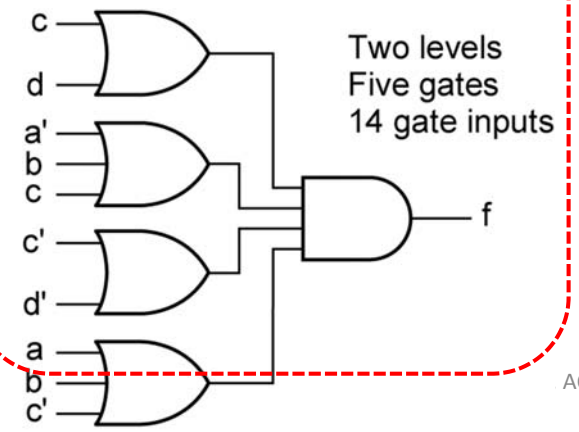
$f = c'd(a' + b) + cd'(a + b)$

OR gate output

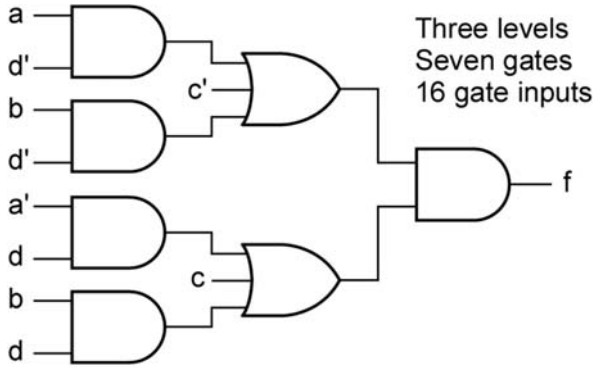


$f = (c + d)(a' + b + c)(c' + d')(a + b + c')$

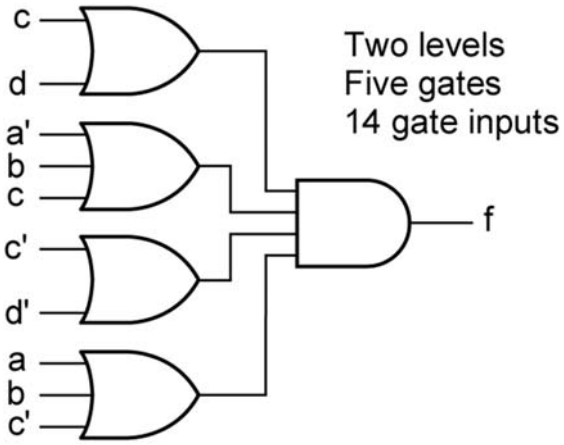
AND gate output

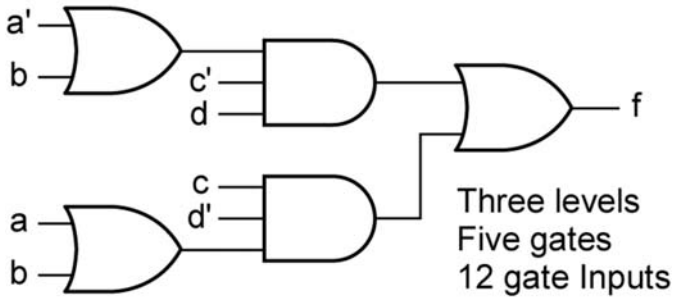


$f = (c + a'd + bd)(c' + ad' + bd')$



‘ For this particular example, the best two-level solution had an AND gate at the output, and the best three-level solution had an OR gate at the output. In general, to be sure of obtaining a minimum solution, one must find *both* the circuit with the AND-gate output and the one with the OR-gate output.






Thanks,..