Digital Electronics

Boolean Algebra & Logic Gates

Simplification Algorithm

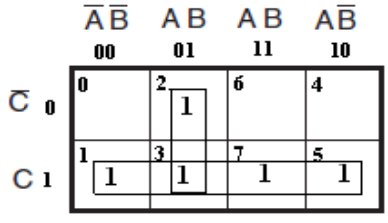
Simplification of logical functions using K-maps is based on the principle of combining terms in adjacent cells. Two cells are said to be adjacent if they differ in only one variable.

1. Identify the ones which cannot be combined with any other ones and encircle them. These are called essential prime implicants.
2. Identify the ones that can be combined in groups of two in only one way. Encircle them.
3. Identify the ones that can be combined with three other ones, to make a group of four adjacent ones, in only one way. Encircle such group of ones.
4. Identify the ones that can be combined with seven other ones, to make a group of eight adjacent ones, in only one way. Encircle them.
5. After identifying the essential groups of 2, 4, and 8 ones, if there still remain some ones which have not been encircled, then these are to be combined with each other or with other already encircled ones.

Examples:

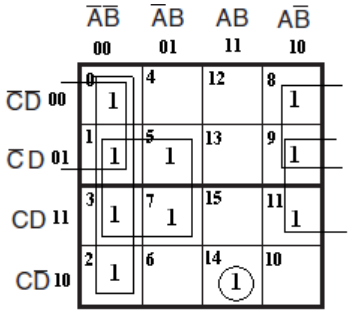
1. Simplify the Boolean function $F = A'C + A'B + AB'C + BC$

Sol: $F = A'C + A'B + AB'C + BC = A'B'C + A'BC' + A'BC + AB'C + AB'C$



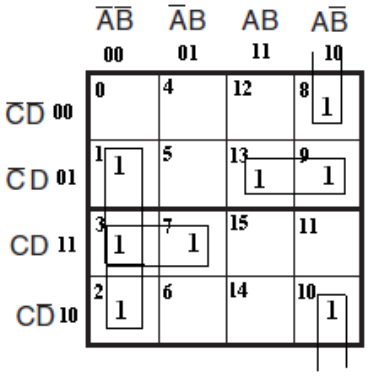
$F = C + A'B$

2. $F(A,B,C,D) = \sum m\,(0,1,2,3,5,7,8,9,11,14)$



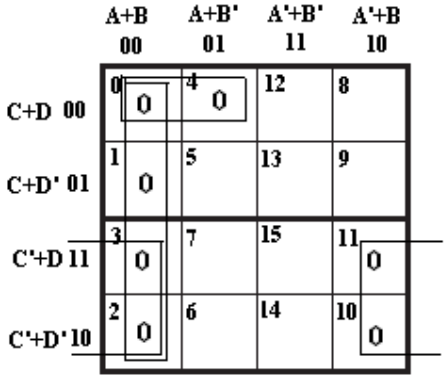
$$F = A'B' + A'D + ABCD' + B'C' + B'D$$

3. $F(A,B,C,D) = \sum m\,(1,2,3,5,7,8,9,10,13)$



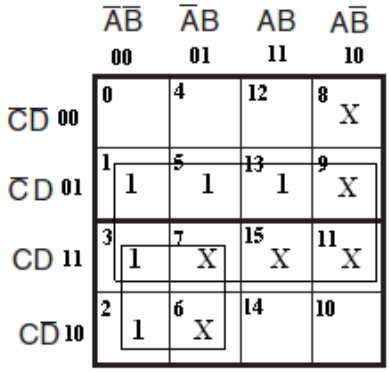
$$F = A'CD + AC'D + AB'D' + A'B'D + A'B'C$$

4. $F(A,B,C,D) = \prod M\,(0,1,2,3,4,10,11)$



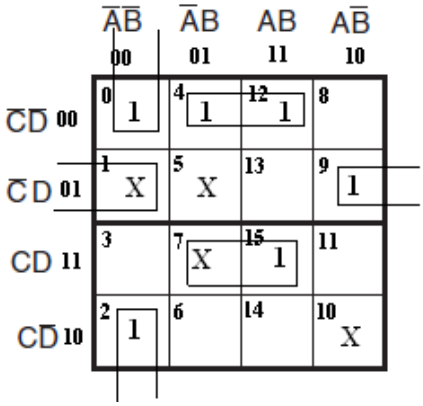
$$F = (A + B)(A + C + D)(B + C')$$

5. $F = \sum m\,(1,2,3,5,13) + \sum d\,(6,7,8,9,11,15)$



$$F = D + A'C$$

6. $F = \sum m(0,2,4,9,12,15) + \sum \emptyset(1,5,7,10)$



$$F = A'B'D' + BCD + BC'D' + B'C'D$$

Two level Implementation:

Any logic circuit can be implemented in two levels by representing the Boolean function either in SOP or POS form. Two level NAND and NOR circuits can be obtained by representing the expression in SOP and POS form respectively. Minimum propagation delay will be obtained by using two level implementation. But as the number of terms increases, the number of inputs increases for the second level gate.

NAND Implementation:

By expressing the given function in SOP form, the logic circuit can be implemented using two level NAND gates.

Example:

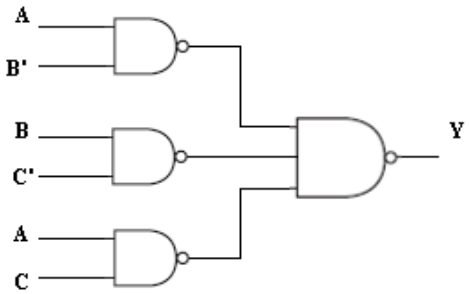
Implement $Y = A(A' + B') + BC' + AC$ using NAND gates.

Sol: $Y = A(A' + B') + BC' + AC$

$= AB' + BC' + AC$ ----- SOP form

By using double complements we get,

$$Y = (Y')' = \{(AB' + BC' + AC)'\}' = (AB')'(BC')'(AC)'$$



NOR Implementation:

By expressing the given function in POS form, the logic circuit can be implemented using two level NOR gates.

Example:

Implement $Y = A'B + BC'$ using NOR gates.

Sol: $Y = A'B + BC'$

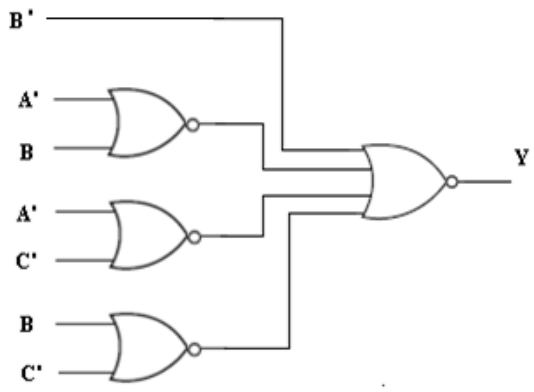
$= (A'B + B)(A'B + C')$ ……. Distributive law

$Y = B(A' + B)(A' + C')(B + C')$ ……. Distributive law

and the expression is in POS form.

$Y = (Y')' = [\{B(A' + B)(A' + C')(B + C')\}']'$

$= [(B)' + (A' + B)' + (A' + C')' + (B + C')']'$



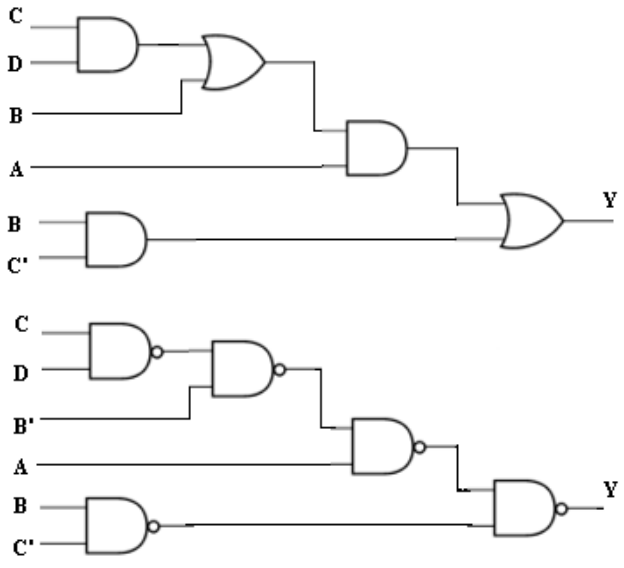
Multilevel Implementation:

Multi level NAND or NOR implementation of a Boolean circuit can be achieved by replacing each gate in the circuit with the NAND or NOR equivalent circuits respectively. With this type of implementation, very large propagation delay is achieved and only two input gates are required.

NAND Implementation:

By replacing each gate in the circuit with the NAND equivalent, multi level NAND circuit is achieved. When two single input NAND gates (inverters) are in series, they can be removed.

Example: $Y = A(B + CD) + BC'$



NOR Implementation:

By replacing each gate in the circuit with the NOR equivalent, multi level NOR circuit is achieved. When two single input NOR gates (inverters) are in series, they can be removed.

Example: $Y = A(B + CD) + BC'$