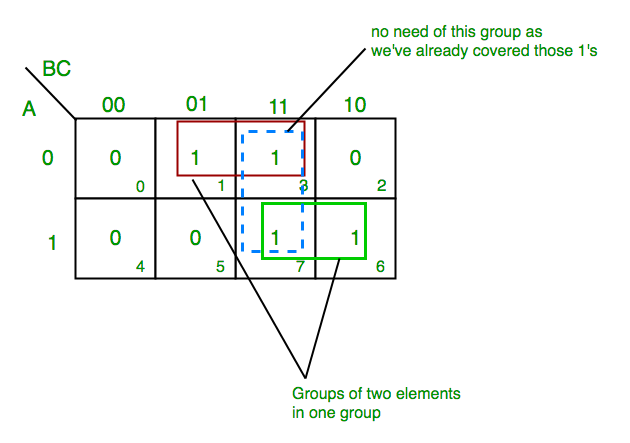
**GATE LEVEL MINIMIZATION**

**K-Map (Karnaugh’s Map)**

* Used to minimize expressions.
* No Boolean algebra is required here.
* Is **either** SOP or POS.
* Pairs of 1s are made for **minterms**.
* While pairs of 0s are made in **maxterms**.

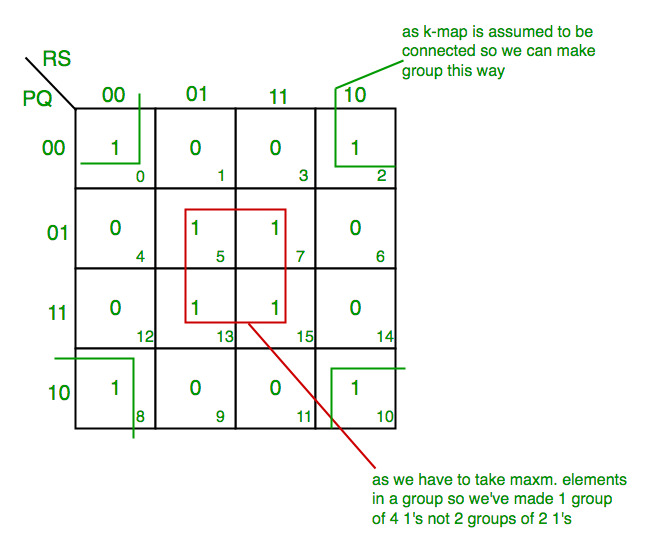
**K-Map for 3 Variables**

**F(A,B,C) = Sigma(1,3,4,5) etc…**



* **Equation formed:** A’C + AB
  + In red pair, A is 0 in both & C is 1 in both.
  + In green pair, A is 1 in both & B is 1 in both.

**K-Map for 4 Variables**



**K-Map Facts**

* In final equation,
  + For **minterms**, obtained variable equations are written as **SOP**.
  + For **maxterms**, obtained variable equations are written as **POS**.
* Variables to be paired, are paired **if possible**, else they are **freely marked**.
* Never forget about **4-cornered** match pair.
* POS != (SOP)’
* (POS of F) = (SOP of F’)’

**Implicants in K-Map**

* Implicant is **product/minterm** term in SOP.
* Or it is **sum/maxterm** term in POS.
* It is simply any term in SOP or POS.

**F = AB + ABC + BC**

* Here, implicant is AB, ABC & BC.

**Types of Implicants**

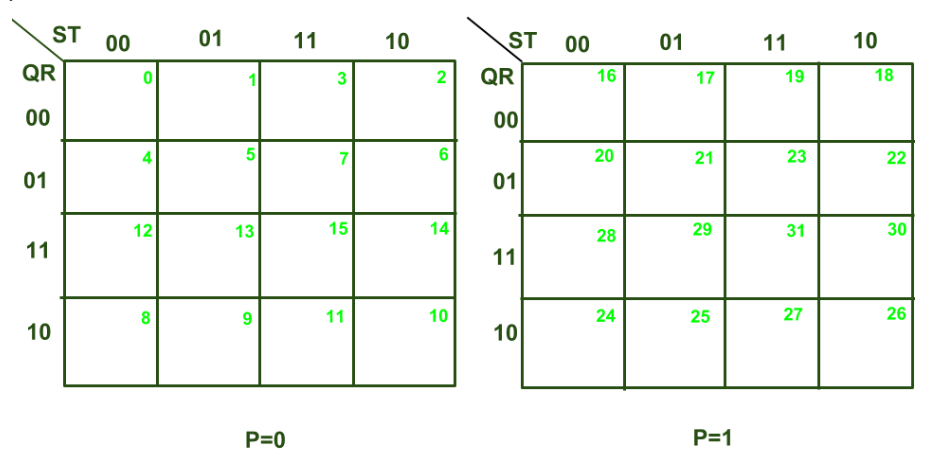
* **Implicant:** Squares, not groups. Squares to be considered.
* **Prime implication (PI):** All possible pair groups in K-Map.
* **Prime implicants:** All implicants involved in the final solution.
* **Essential prime implications (EPI):** Those prime implications that appear in the final solution.
* **Redundant prime implication (RPI):** Implicants with shared minterms/maxterms with EPI, not appearing in final solution.
* **Selective prime implication (SPI):** Neither EPI nor RPI.

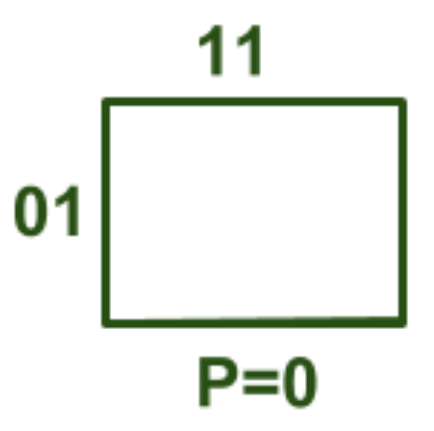
**Facts to Note**

* Group of tetra implicants **is possible**.
* There must be **no repetition** of minterms/maxterms among the same variety of implication, except SPI.
* Implicant = Pair/group reference

**5 Variable in K-Map**

* **Efficacy:** Effectiveness
* The **more** the number of variables, the **less** the efficacy of K-Map.
* **5 variable K-Map (P is MSB):-**

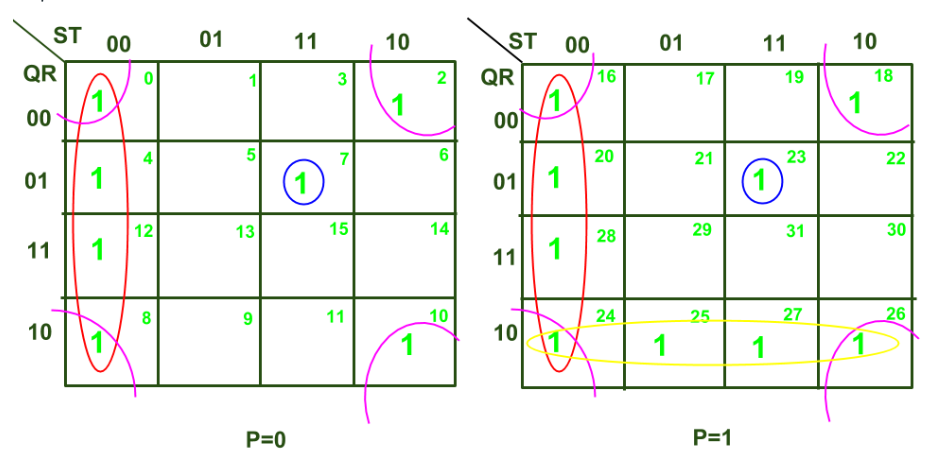




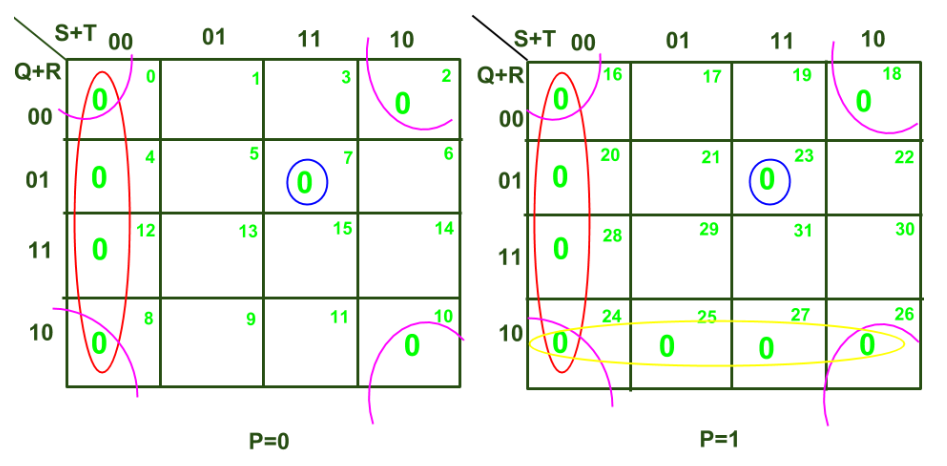
* P=0, 01 = QR, 11 = ST

**(PQRST) = (00111) = 7 in decimal**

* So, it described the position of cell no. 7.
* Write **1** corresponding to the cell numbers, for **minterms**.
* Write **0** corresponding to the cell numbers, for **maxterms**.
* **Sub-cube:** Row or column in K-Map.
* **Solving it:**

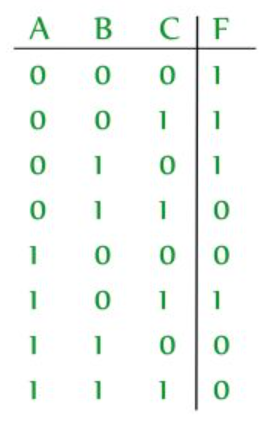
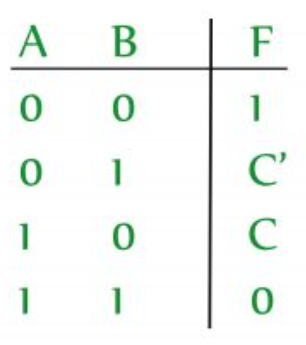


* The groups made (the pairs or tetra groups) are in **2n** form **always**.
* **Example for POS:-**

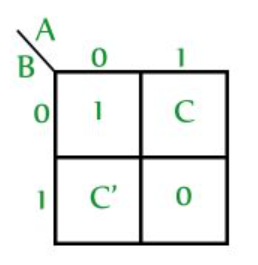


**Variable Entrant Map (VEM)**

* Used to handle large number of variables.
* 3 variable function can be written in form of 2, where the output is written in form of **3rd** variable.

* **As per what we did:**
  + 00 = 0
  + 01 = C
  + 10 = C’
  + 11 = 1



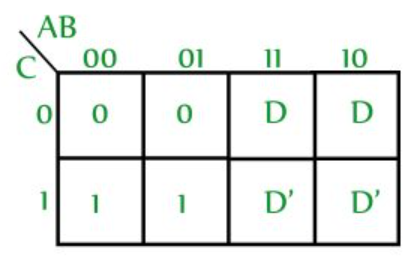
**Advantages of Using VEM**

* Can represent many variables, with less variables.
* Helps in solving **multiplexer** circuits.

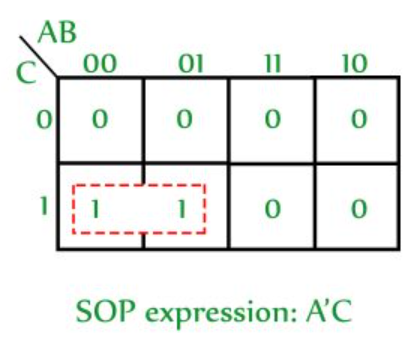
**Multiplexer:** Circuit taking multiple inputs and giving a single output.

**Minimization Procedure for VEM**

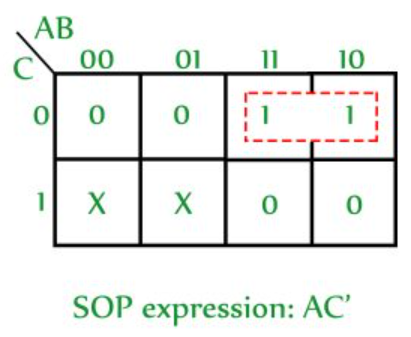
* **Step 1:** Write the minterm outputs.



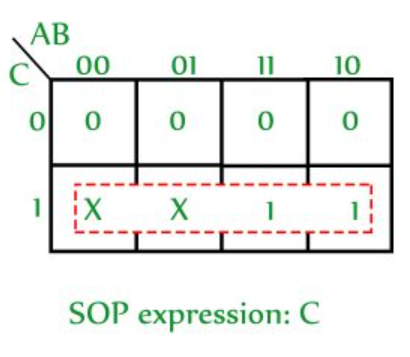
* **Step 2:** Write **D as 0** (whether complemented or not), and **leave don’t cares** as it is.



* **Step 3:** Replace **1s with don’t cares** and **D with 1s**, and again **leave don’t cares**.



* **Step 4:** Multiply the obtained SOP term with the variable replaced. So, the SOP expression becomes AC’D
* **Step 5:** **Repeat** same procedure (step 3 & 4) **for D’**



* **Step 6 (last):** **OR** all the obtained expressions.

**Minimization of Boolean Function**

* **Literal:** Mathematical representation of a Boolean function.
* **Minimization:** Simplifying algebraic Boolean expression.
* Minimization also **reduces** number of circuit connections/ logic gates.
* **It is done by using:**
  + algebraic manipulation
  + K-Map method.

**Minimization Using Algebraic Manipulation**

* Used for **medium** sized expression.
* **Medium sized expression:** 4 or 5 variables involved.
* Manual, so prone to human error.

**Laws Used in Algebraic Manipulation**

**A + A’ = 1**

**A + A’B = A + B**

**A + AB = A**

**Minimization Using K-Map**

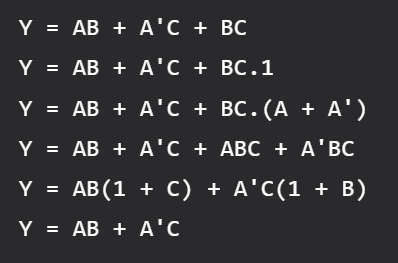
* **Faster** and **less** complex than algebraic manipulation.
* Used for equation with upto 5 variables.

**Consensus Theorem**

**AB + A’C + BC = AB + A’C**

* Also known as **redundancy theorem**.
* BC is **consensus/resolvent/redundancy term** of AB and A’C, i.e., **conjuction** of all 3 unique literals.
* Same goes with the **SOP** form of the same equation.

**Proof for Consensus Theorem**



* **For SOP:-**

