Combinational Logic Circuits

There are 2 types of quantities: analog and digital. Analog ones are those continuously varying. Digital ones are discrete. In conventional logic systems, there are only 2 states: TRUE and FALSE.

Axioms

Let $L_k = 1$ and $\overline{L_k} = 0$, then

$$\bigvee_i L_i = \bigvee_i L_i \bigvee_j \overline{L_j} = 1$$

Operator OR

$$\bigvee_i \overline{L_i} = 0$$

And

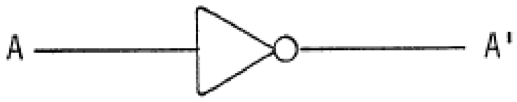
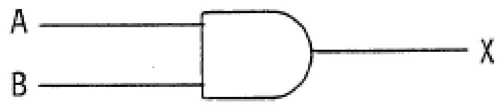
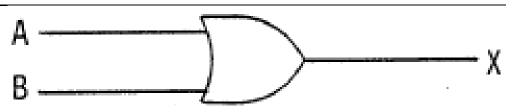
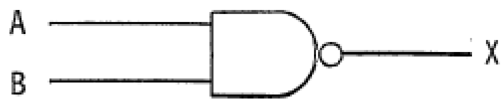
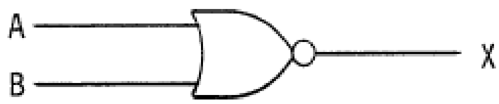
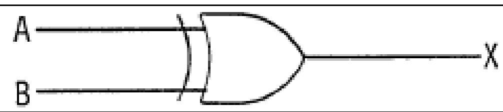
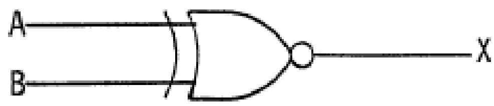
Operator AND

$$\bigwedge_i L_i = 1$$

$$\bigwedge_i \overline{L_i} = \bigwedge_i L_i \bigwedge_j \overline{L_j} = 0$$

Logic gates

In logic gates, we use Boolean algebra and we use $+, \Sigma$ to represent OR; and $*, \Pi$ to represent AND:

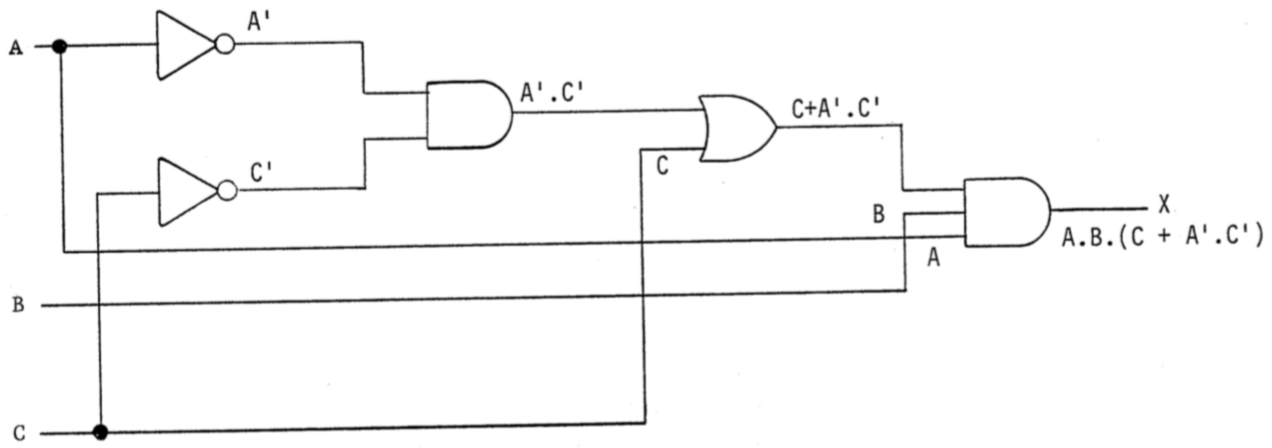
Name	Gate	Operation
NOT		$X = \bar{A}$
AND		$X = AB$
OR		$X = A + B$
NAND		$X = \overline{AB}$
NOR		$X = \overline{A + B}$
Exclusive OR (XOR)		$X = \bar{A}B + A\bar{B} = A \oplus B$
Exclusive NOR (XNOR)		$X = AB + \bar{A}\,\bar{B}$

Boolean algebra

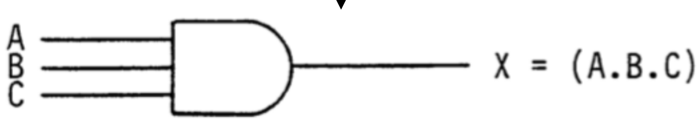
Theorems

Name	Operation
Commutative Laws	$AB = BA$ $A + B = B + A$
Associative Laws	$A(BC) = (AB)C$ $A + (B + C) = (A + B) + C$
Idempotent Laws	$AA = A$ $A + A = A$
Laws of Identities	$A(1) = A$ $A + 0 = A$
Laws of Null Elements	$A(0) = 0$ $A + 1 = 1$
Laws of Complements	$A\bar{A} = 0$ $A + \bar{A} = 1$
Laws of Absorption	$A + AB = A$ $A(A + B) = A$
Distributive Laws	$A(B + C) = AB + AC$ $(A + B)(A + C) = A + BC$
Law of Double Negation	$\bar{\bar{A}} = A$
De Morgan's Theorem	$\overline{\prod_{\iota \in I} A_\iota} = \sum_{i \in I} \overline{A_\iota}$ $\overline{\sum_{\iota \in I} A_\iota} = \prod_{i \in I} \overline{A_\iota}$

Example:
Simplify the following circuit:

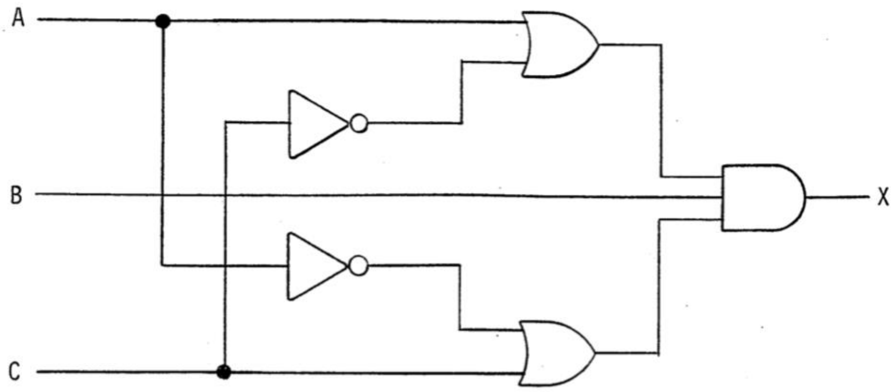


$$\begin{aligned} X &= AB(C + \bar{A}\bar{C}) \\ &= ABC + AB\bar{A}\bar{C} \\ &= ABC \end{aligned}$$



Therefore, only 1 component is required.

Circuit analysis and synthesis
For any logic circuit, for example:



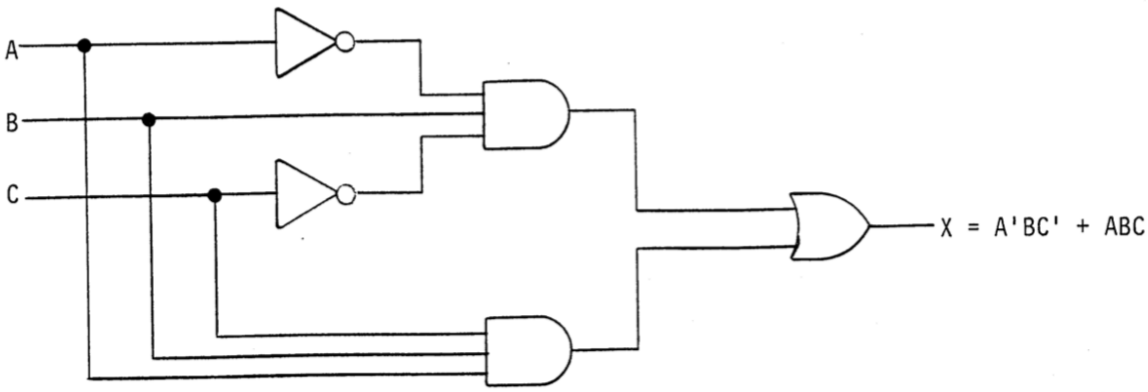
Analysis:
This circuit can be described by:

$$X = (A + \bar{C})B(\bar{A} + C)$$

Which can be simplified into:

$$\begin{aligned} X &= (A + \bar{C})B(\bar{A} + C) \\ &= (A\bar{A} + AC + \bar{A}\bar{C} + \bar{C}C)B \\ &= (\bar{A}\bar{C} + AC)B \\ &= \bar{A}B\bar{C} + ABC \end{aligned}$$

Synthesis:
This can be implemented by a circuit:



Combinational Logic Circuit

Standard forms

To minimize the number of logic ICs in a circuit, it is convenient to express logic functions in sum of products form (SOP) or product of sums form (POS):

SOP:

Example:

Find the function that can represent the following truth table:

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$\begin{aligned} X &= \bar{A}BC + A\bar{B}C + ABC \\ &= \bar{A}BC + A\bar{B}C + ABC + ABC \\ &= AC(B + \bar{B}) + BC(A + \bar{A}) \\ &= AC + BC \end{aligned}$$

This Minimized SOP form requires two 2-input AND gates and one 2-input OR gate.

Then, by De Morgan's Theorem:

$$\begin{aligned} X &= AC + BC \\ &= \overline{(\overline{AC})(\overline{BC})} \end{aligned}$$

This expression requires three 2-input NAND gates, which use one type of IC only.

POS:

The minimized SOP form above can be expressed as:

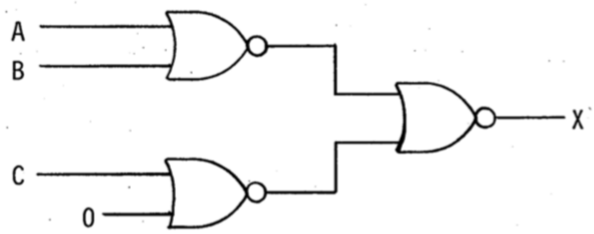
$$\begin{aligned} X &= AC + BC \\ &= (A + B)C \end{aligned}$$

This is called the Minimized POS form and requires 1 AND gate and 1 OR gate.

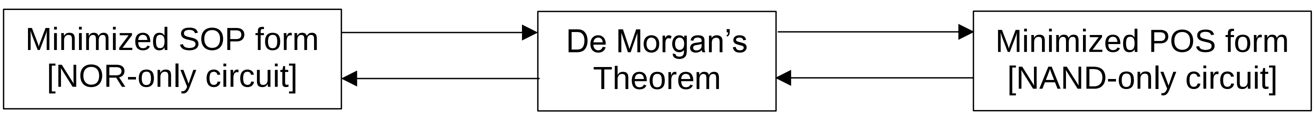
Then, if we apply De Morgan's Theorem:

$$\begin{aligned} X &= (A + B)C \\ &= \overline{\overline{A + B} + \bar{C}} \end{aligned}$$

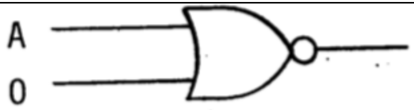
Which needs only 3 NOR gates to implement:



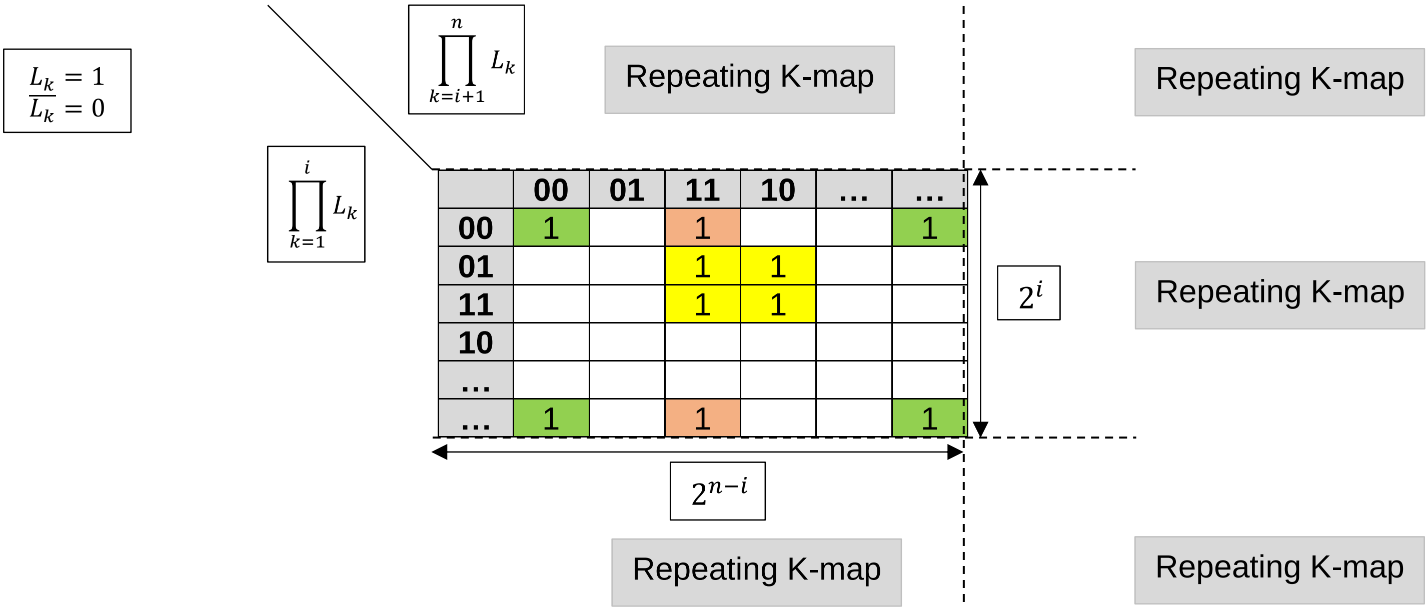
We can see that:



Also, a NOT function can be performed by NAND gate and NOR gate:

Configuration	Expression
	$\overline{A(1)} = \bar{A}$
	$\overline{A+0} = \bar{A}$

<u>Karnaugh map (K-map) and minimization technique</u>

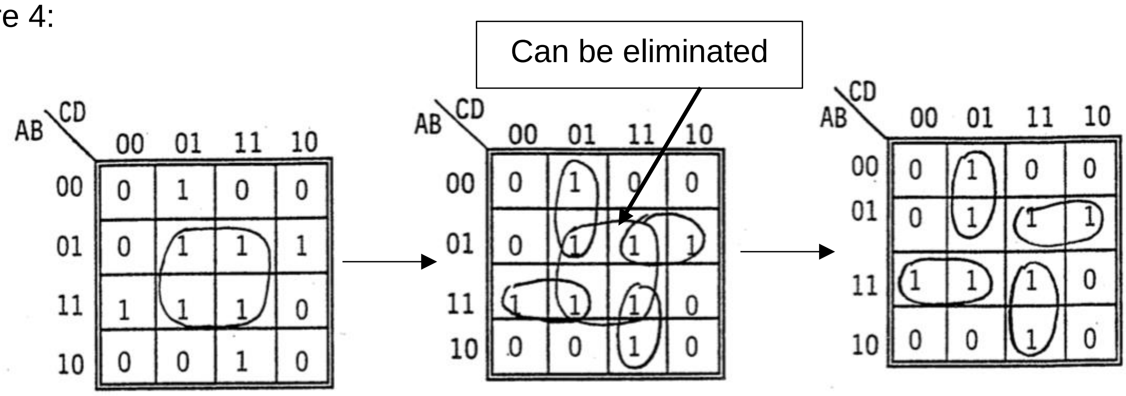


Procedures:

1. Express logical expressions in SOP form:
$$X = \sum\left(\prod L_k\right)$$
2. Mark [1] in boxes which describe the terms:
$$\prod L_k$$
in the expression $X = \sum(\prod L_k)$
3. Encircle adjacent [1]s start with largest size possible which obey $2^s, 1 < s < n$
4. Some circles may overlap, eliminate those with ALL [1]s encircled more than once by other circles.
5. Find the invariant L_ks of each circle. Then form the simplest SOP form with $L_k = 1, \overline{L_k} = 0$

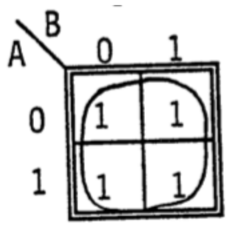
**IF we want to get the simplest POS form using K-map, we encircle [0]s instead, all other procedures are the same

For procedure 4:
Example:



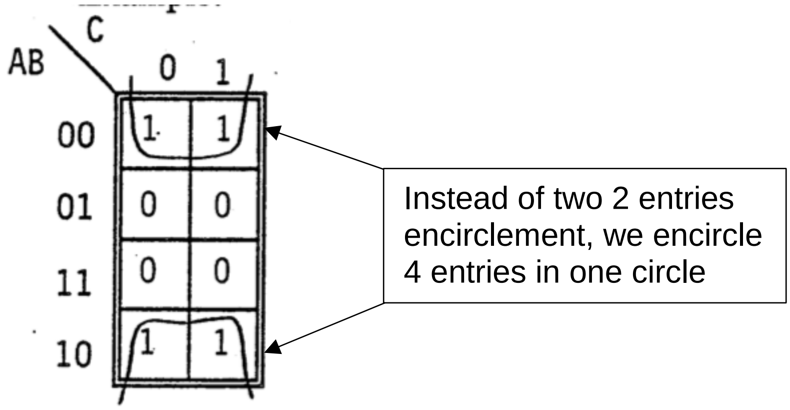
More examples:

1. If all the entries in the K-map is filled with [1], then the expression is: $X = 1$

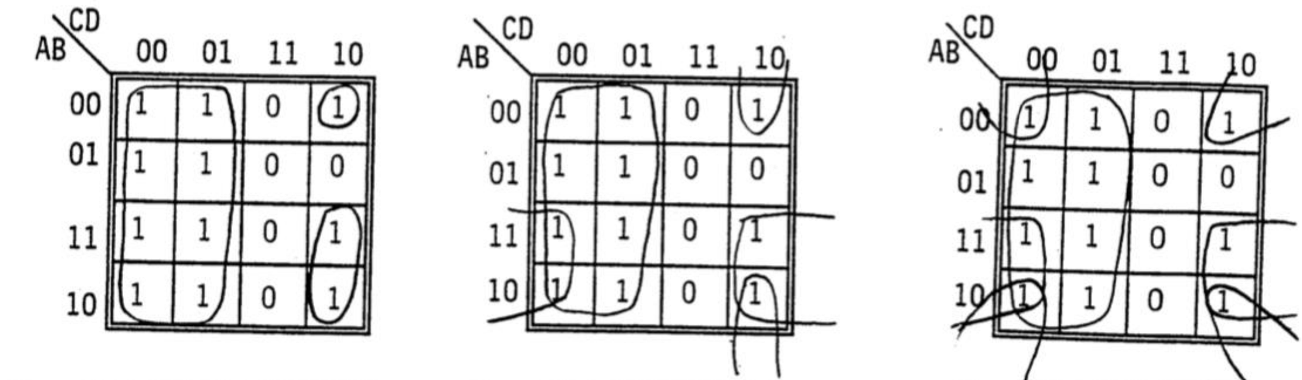


2. To get the simplest form of the expression we need to pretend there are 8 repeating K-maps around the K-map we are considering:

$$X = \bar{A}\bar{B}C + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$$



3. Another example to show the importance of the idea above:



X = C'+ACD'+A'B'CD' X = C'+AD'+B'CD' X = C'+AD'+B'D' (simplest)

Tutorial

Use K-map to obtain the Minimised SOP and POS form of the following:

1. $X = A\bar{B}C + AB\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC$ $[X = B + C]$
2. $X = \bar{A}BC + AB\bar{C} + \bar{A}\bar{B}\bar{C} + ABC + \bar{A}\bar{B}C + \bar{A}B\bar{C}$ $[X = \bar{A} + B]$
3. $X = \bar{A}B + \bar{A}D + ABD + BCD$ $[X = \bar{A}B + \bar{A}D + BD]$
4. $X = \bar{A}\bar{B} + \bar{A}\bar{C}D + AC + \bar{B}C$ $[X = \bar{A}\bar{B} + \bar{A}\bar{C}D + AC]$
5. $X = AB\bar{C} + B\bar{C}D + \bar{A}BD + \bar{A}CD + \bar{A}\bar{B}C$ $[X = \bar{A}\bar{B}C + \bar{A}BD + AB\bar{C}]$