Boolean Algebra – Defined with a set of elements, a set of operators and a number of unproved axioms or postulates

A set of elements – any collections of objects with a common property

A set of operators – a set of rules that act on a set of elements

Postulates – Form the basic assumptions from which rules, theorems and properties of the system are deduced

Common Postulates of a Mathematical System:

* Closure – A set is closed with respect to a binary operator if the binary operator specifies a rule for obtaining a unique element of that set. The set of natural numbers is closed with respect to the + operator since the result of adding any two members will be another member. However, they are not closed with respect to the – operator since there is a possibility of obtaining a negative number, which would not be a member of the set.
* Associative Law – A binary operator \*(any operator) on a set is said to be associative whenever , given all members are part of the same set.
* Commutative Law – A binary operator \* on a set is said to be commutative whenever , given all members are part of the same set.
* Identity Element – A set is said to have an identity element with respect to a binary operator \* on the set if there exists an element such that

for every value of in the set.

* Inverse – A set having the identity element with respect to a binary operator \* is said to have an inverse if for every element there also exists an element such that
* Distributive Law – If \* and + are two binary operators on a set, \* is said to be distributive over + if

Field – This is an algebraic structure that has a set of elements, together with two binary operators. For example:

1854 – George Boole introduces systematic treatment of logic and develops Boolean algebra

1938 – C.E. Shannon introduces two level Boolean algebra called switching circuit

1904 – E.V. Huntington’s postulates. Now used for formal definitions of Boolean algebra

Huntington’s Postulates:

Boolean algebra is an algebraic structure defined on a set of elements together with two binary operators \* and + provided they satisfy the following conditions:

* Closure with respect to +
* Closure with respect to \*
* An identity element with respect to + is designated by 0 ()
* An identity element with respect to \* is designated by 1 ()
* Commutative with respect to +
* Commutative with respect to \*
* \* is distributive over +
* + is distributive over \*
* For every element , an element exists such that and
* There exist at least two elements and such that

----------------------------- SKIPPED OVER PARTS---------------------------------

Positive Logic assigns a 0 value to 0 and a 1 value to anything other.

Negative logic does the opposite.

Negative logic is the opposite of positive logic in every way.

Positive logic can be converted to negative logic by switching AND gates to OR gates and vice versa and 1’s to 0’s and vice versa.

Venn Diagrams can be read with digital logic.

Minterm: A product that contains all variables of a particular function in either complemented (0) or non-complemented (1) form (standard product). variables can be combined with AND to give minterms. The symbol of a minterm is where is the decimal equivalent.

Maxterm: A sum that contains all variables of a particular function in either non-complemented (0) or complemented (1) form (standard sum). variables can be combined with OR to give maxterms. The symbol of a maxterm is where is the decimal equivalent.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Minterms | | Maxterms | |
| Term | Designation | Term | Designation |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Minterms and Maxterms can be used to express a function.

For example, if a function gives a result of 1 for the values 010, 100, 101, 110 and 111, then using minterms

Similarly, the complement of the function can be expressed as

This is called the standard sum of products (SOP) form of the function.

The function can also be expressed as

This is known as the minimal SOP form of the function.

Converting backwards from minimal SOP from to canonical/standard SOP form, each term of the minimal SOP form must be ANDed with where is each of the missing terms.

So,

Minterms are used to express a function in terms of its high outputs. Minterms are generally expressed in SOP form.

For low outputs, the maxterms are used. Maxterms are generally expressed in POS form.

If a function gives a low output for 000, 001 and 011,

This is known as the standard POS form.

As previous show, the SOP form of the function ’ was given by

Complementing this using De Morgan’s law will give the POS form of , i.e. the one obtained from maxterms.

The standard POS form of can also be expressed as

This is called the minimal POS form.

Converting backwards from minimal POS form to canonical/standard POS form, each term of the minimal POS form must be ORed with where is each of the missing terms.

So, if