Boolean Algebra

- Digital circuits
- Boolean Algebra
- Two-Valued Boolean Algebra
- Boolean Algebra Postulates
- Precedence of Operators
- Truth Table & Proofs
- Duality

- Basic Theorems of Boolean Algebra
- Boolean Functions
- Complement of Functions
- Standard Forms
- Minterm & Maxterm
- Canonical Forms
- Conversion of Canonical Forms
- Binary Functions

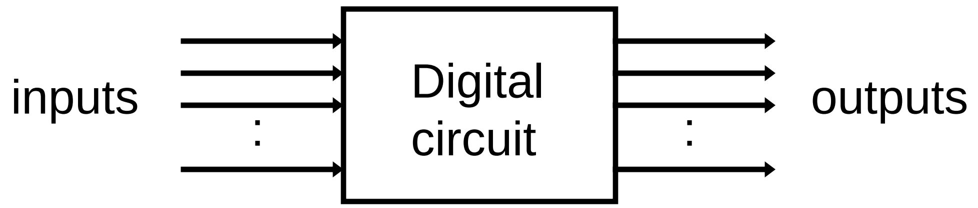
Introduction

Boolean algebra forms the basis of logic circuit design. Consider very simple but common example: **if (A is true) and (B is false) then print "the solution is found".** In this case, two Boolean expressions (A is true) and (B is false) are related by a connective '**and**'. How do we define these?

In typical circuit design, there are many conditions to be taken care of (for example, when the 'second counter' = 60, the 'minute counter' is incremented and 'second counter' is made 0. Thus it is quite important to understand Boolean algebra. In subsequent chapters, we are going to further study how to minimize the circuit using laws of Boolean algebra

Digital Circuits

- **Digital circuit** can be represented by a black-box with inputs on one side, and outputs on the other.



The input/output signals are discrete/digital in nature, typically with two distinct voltages (a high voltage and a low voltage).

In contrast, analog circuits use continuous signals.

Digital Circuits

- Advantages of Digital Circuits over Analog Circuits:
  - more reliable (simpler circuits, less noise-prone)
  - specified accuracy (determinable)
  - but slower response time (sampling rate)

- Important advantages for two-valued Digital Circuit:
  - Mathematical Model – Boolean Algebra
  - Can help *design*, *analyse*, *simplify* Digital Circuits.

Boolean Algebra

What is an Algebra? (e.g. algebra of integers)
set of elements (e.g. 0,1,2,..)
set of operations (e.g. +, -, *,..)
postulates/axioms (e.g. 0 + x = x,..)

- In 1854, George Boole invented a new kind of algebra--- the algebra of logic, Popularly known as Boolean Algebra.
- Events : *true* or *false*
- Connectives : a *OR* b; a *AND* b, *NOT* a
- Example: Either “it has rained” *OR* “someone splashed water”, “must be tall” *AND* “good vision”.

Boolean Algebra

a	b	a AND b
F	F	F
F	T	F
T	F	F
T	T	T

a	b	a OR b
F	F	F
F	T	T
T	F	T
T	T	T

a	NOT a
F	T
T	F

Later, Shannon introduced switching algebra (two-valued Boolean algebra) to represent bi-stable switching circuit.

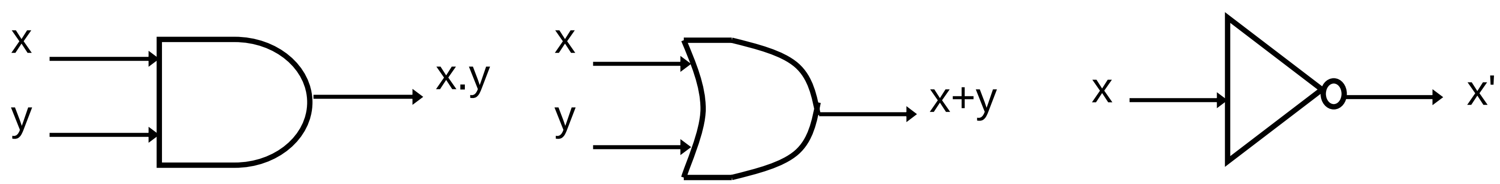
Two-valued Boolean Algebra

- Set of elements: {0,1}
- Set of operations: { ., + , ‘ }

x	y	x . y
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0



Signals: High = 5V = 1; Low = 0V = 0

Boolean Algebra Postulates

A **Boolean algebra** consists of a set of elements *B*, with two binary operations {+} and {.} and a unary operation {'}, such that the following axioms hold:

- The set *B* contains at least two distinct elements x and y.
- **Closure**: For every x, y in *B*,
  - x + y is in *B*
  - x . y is in *B*
- **Commutative laws**: For every x, y in *B*,
  - x + y = y + x
  - x . y = y . x

- **Associative laws**: For every x, y, z in *B*,
  - $(x + y) + z = x + (y + z) = x + y + z$
  - $(x \cdot y) \cdot z = x \cdot (y \cdot z) = x \cdot y \cdot z$
- **Identities** (0 and 1):
  - $0 + x = x + 0 = x$ for every x in *B*
  - $1 \cdot x = x \cdot 1 = x$ for every x in *B*
- **Distributive laws**: For every x, y, z in *B*,
  - $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
  - $x + (y \cdot z) = (x + y) \cdot (x + z)$

- **Complement**: For every x in *B*, there exists an element x' in *B* such that
  - $x + x' = 1$
  - $x \cdot x' = 0$

The set *B* = {0, 1} and the logical operations OR, AND and NOT satisfy all the axioms of a Boolean algebra.

A **Boolean function** maps some inputs over {0,1} into {0,1}

A **Boolean expression** is an algebraic statement containing Boolean variables and operators.

Precedence of Operators

- To lessen the brackets used in writing Boolean expressions, operator precedence can be used.
- Precedence (highest to lowest): **'   .   +**
- Examples:

  a . b + c = (a . b) + c

  b' + c = (b') + c

  a + b' . c = a + ((b') . c)

Precedence of Operators

- Use brackets to overwrite precedence.
- Examples:

$$a \cdot (b + c)$$

$$(a + b)' \cdot c$$

Truth Table

- Provides a listing of every possible combination of inputs and its corresponding outputs.

INPUTS	OUTPUTS
…	…
…	…

- Example (2 inputs, 2 outputs):

x	y	x . y	x + y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Truth Table

- Example (3 inputs, 2 outputs):

x	**y**	**z**	**y + z**	**x.(y + z)**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Proof using Truth Table

- Can use truth table to prove by perfect induction.
- Prove that: x . (y + z) = (x . y) + (x . z)

(i) Construct truth table for LHS & RHS of above equality.

x	y	z	y + z	x.(y + z)	x.y	x.z	(x.y)+(x.z)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

(ii) Check that LHS = RHS

Postulate is SATISFIED because output column 2 & 5 (for LHS & RHS expressions) are equal for all cases.

Duality

- **Duality Principle** – every valid Boolean expression (equality) remains valid if the operators and identity elements are interchanged, as follows:

  $+ \leftrightarrow \mathbf{.}$

  $1 \leftrightarrow 0$

- Example: Given the expression

  $$a + (b\mathbf{.}c) = (a+b)\mathbf{.}(a+c)$$

  then its dual expression is

  $$a \mathbf{.} (b+c) = (a\mathbf{.}b) + (a\mathbf{.}c)$$

Duality

- Duality gives free theorems – “two for the price of one”. You prove one theorem and the other comes for free!
- If $(x + y + z) = x \cdot y \cdot z$ is valid, then its dual is also valid:

  $$(x \cdot y \cdot z) = x + y + z$$

- If $x + 1 = 1$ is valid, then its dual is also valid:

  $$x \cdot 0 = 0$$

Basic Theorems of Boolean Algebra

- Apart from the axioms/postulates, there are other useful theorems.

1. **Idempotency**.

$$\text{(a) } x + x = x \qquad \text{(b) } x \,.\, x = x$$

Proof of (a):

$$\begin{aligned} x + x &= (x + x).1 && \text{(identity)} \\ &= (x + x).(x + x') && \text{(complementarity)} \\ &= x + x.x' && \text{(distributivity)} \\ &= x + 0 && \text{(complementarity)} \\ &= x && \text{(identity)} \end{aligned}$$

Basic Theorems of Boolean Algebra

2. **Null elements** for + and . operators.

    (a) $x + 1 = 1$ (b) $x \,.\, 0 = 0$

3. **Involution**. $(x')' = x$

4. **Absorption**.

    (a) $x + x\mathbf{.}y = x$ (b) $x\mathbf{.}(x + y) = x$

5. **Absorption** (variant).

    (a) $x + x'\mathbf{.}y = x+y$ (b) $x\mathbf{.}(x' + y) = x\mathbf{.}y$

Basic Theorems of Boolean Algebra

6. **DeMorgan**.

   (a) $(x + y)' = x' \cdot y'$

   (b) $(x \cdot y)' = x' + y'$

7. **Consensus**.

   (a) $x \cdot y + x' \cdot z + y \cdot z = x \cdot y + x' \cdot z$

   (b) $(x+y) \cdot (x'+z) \cdot (y+z) = (x+y) \cdot (x'+z)$

Basic Theorems of Boolean Algebra

- Theorems can be proved using the truth table method. (Exercise: Prove De-Morgan's theorem using the truth table.)
- They can also be proved by algebraic manipulation using axioms/postulates or other basic theorems.

Basic Theorems of Boolean Algebra

- Theorem 4a (absorption) can be proved by:

$$
\begin{aligned}
x + x.y &= x.1 + x.y && \text{(identity)}\\
&= x.(1 + y) && \text{(distributivity)}\\
&= x.(y + 1) && \text{(commutativity)}\\
&= x.1 && \text{(Theorem 2a)}\\
&= x && \text{(identity)}
\end{aligned}
$$

- By duality, theorem 4b:

$$x.(x+y) = x$$

- Try prove this by algebraic manipulation.

Boolean Functions

- **Boolean function** is an expression formed with binary variables, the two binary operators, OR and AND, and the unary operator, NOT, parenthesis and the equal sign.
- Its result is also a binary value.
- We usually use **.** for AND, **+** for OR, and **'** for NOT. Sometimes, we may omit the **.** if there is no ambiguity.

Boolean Functions

- Examples:

  F1= x.y.z'

  F2= x + y'.z

  F3=(x'.y'.z)+(x'.y.z)+(x.y')

  F4=x.y'+ x'.z

x	y	z	F1	F2	F3	F4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

From the truth table, F3=F4.
Can you also prove by algebraic manipulation that F3=F4?