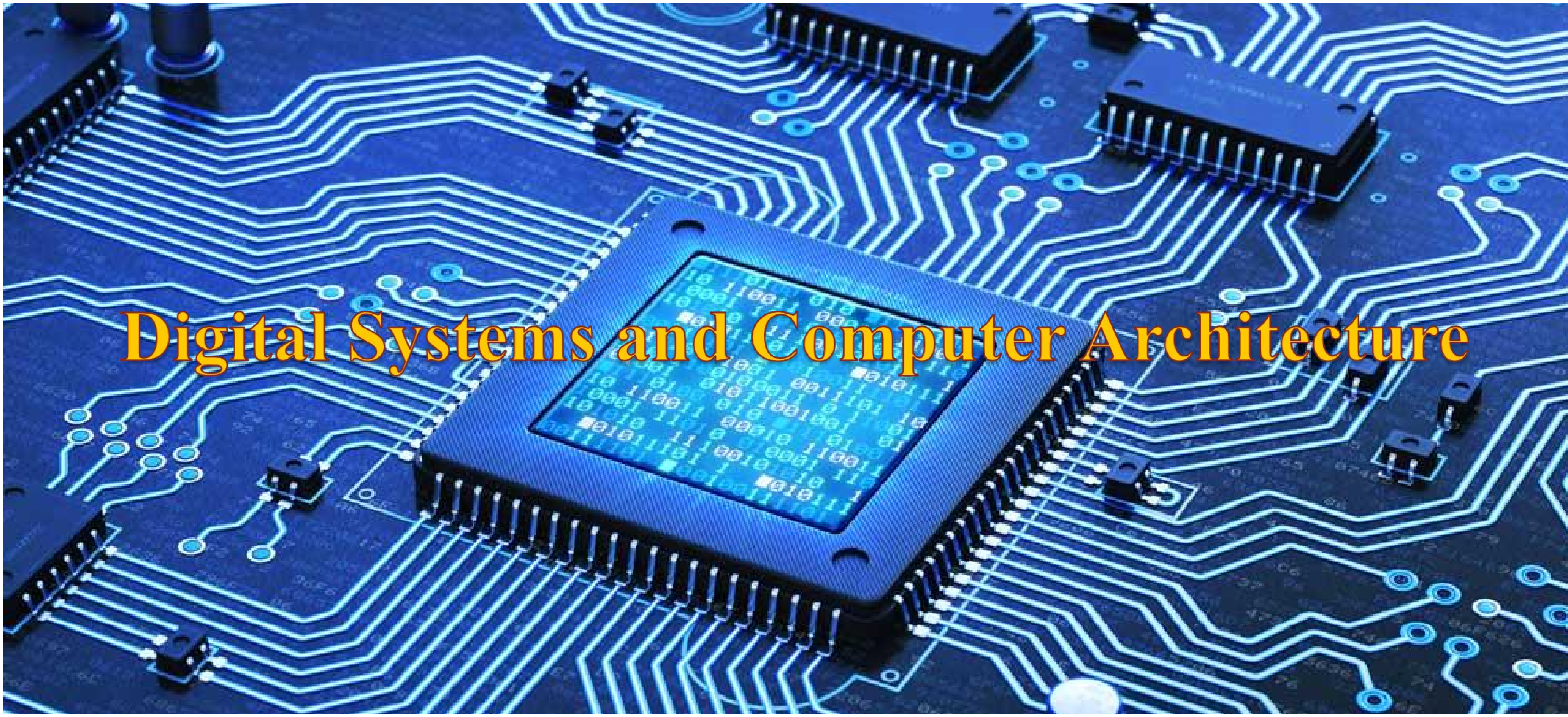


Digital Systems and Computer Architecture

Session 2.4

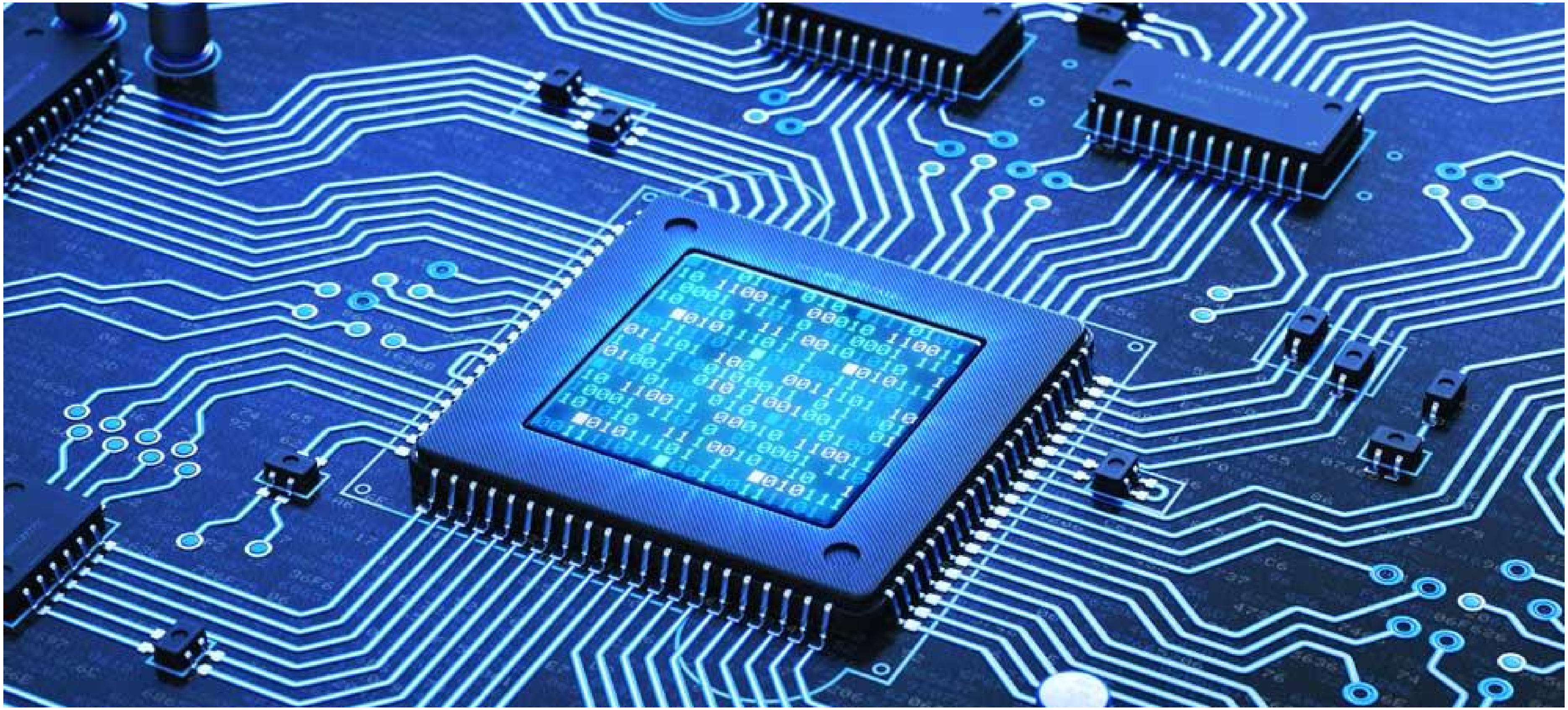
Module 2

Mouli Sankaran

Theorems in Boolean Algebra

Session 2.4: Focus

- Introduction to Boolean Algebra
- Ordinary Algebra Vs Boolean Algebra
- Boolean Expressions
  - Variables, literals and terms
  - Complement and Dual
- DeMargon’s Theorems



Introduction to Boolean Algebra

George Boole

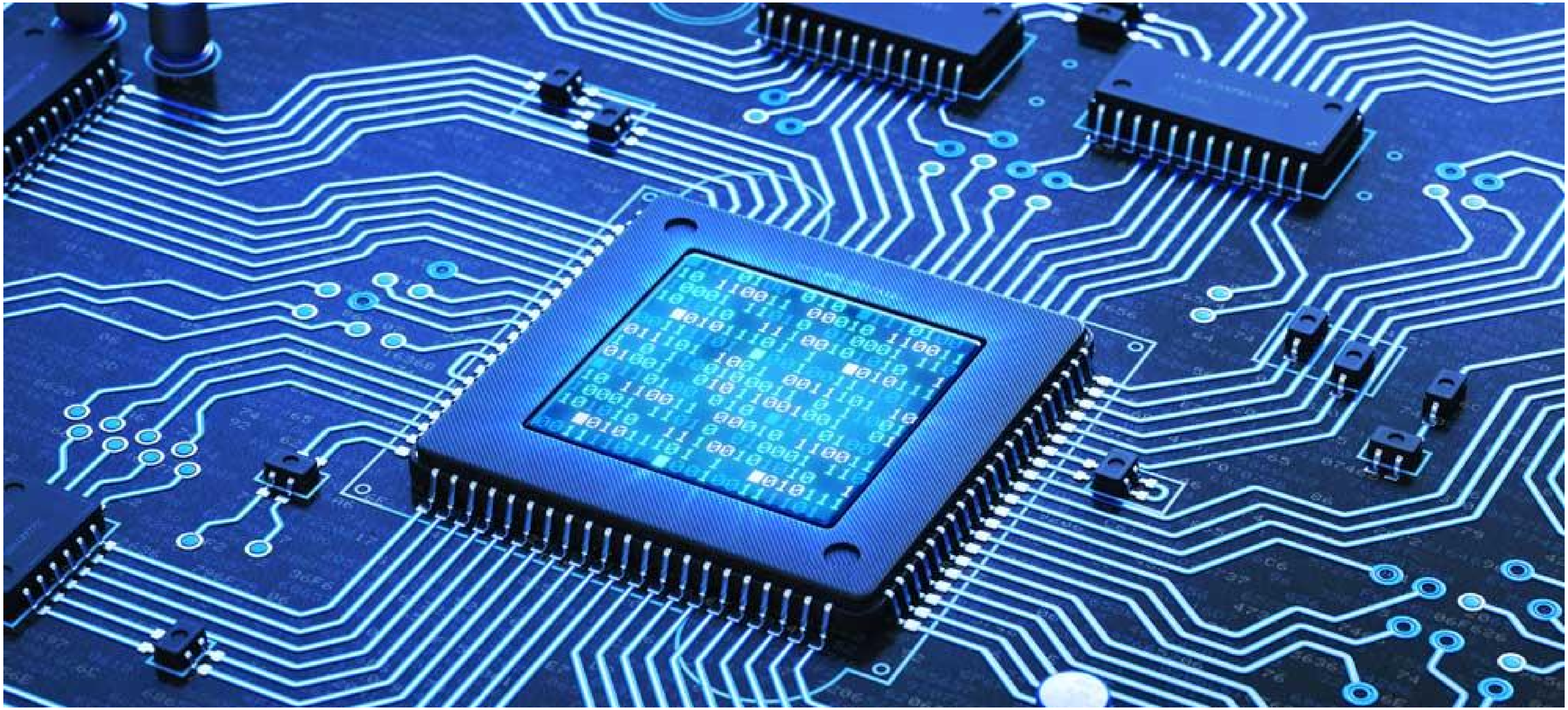
- **George Boole** (2 November 1815 – 8 December 1864)
- He was an English mathematician, educator, philosopher and logician.
- He worked in the fields of differential equations and algebraic logic, and is best known as the author of **The Laws of Thought (1854)** which contains **Boolean algebra**.
- Boolean logic is credited with laying the foundations for the information age.

Boolean Algebra

- Founded by **George Boole**, in **1854**.
- Boolean algebra is **mathematics of logic**, a systematic way of expressing and analyzing the operations of **logic circuits**
- It is one of the **most basic tools** available to **the logic designer**
  - It can be effectively **used** for **simplification** of **complex logic expressions**
- Now, let us have a closer look at the different **postulates** and **theorems** of **Boolean algebra**
- Their applications in **minimizing Boolean expressions**

Introduction to Boolean Algebra

- **Boolean algebra**, quite interestingly, is **simpler** than **ordinary algebra**.
- It is also composed of a set of **symbols** and a set of **rules to manipulate** these symbols



Ordinary Algebra
Vs
Boolean Algebra

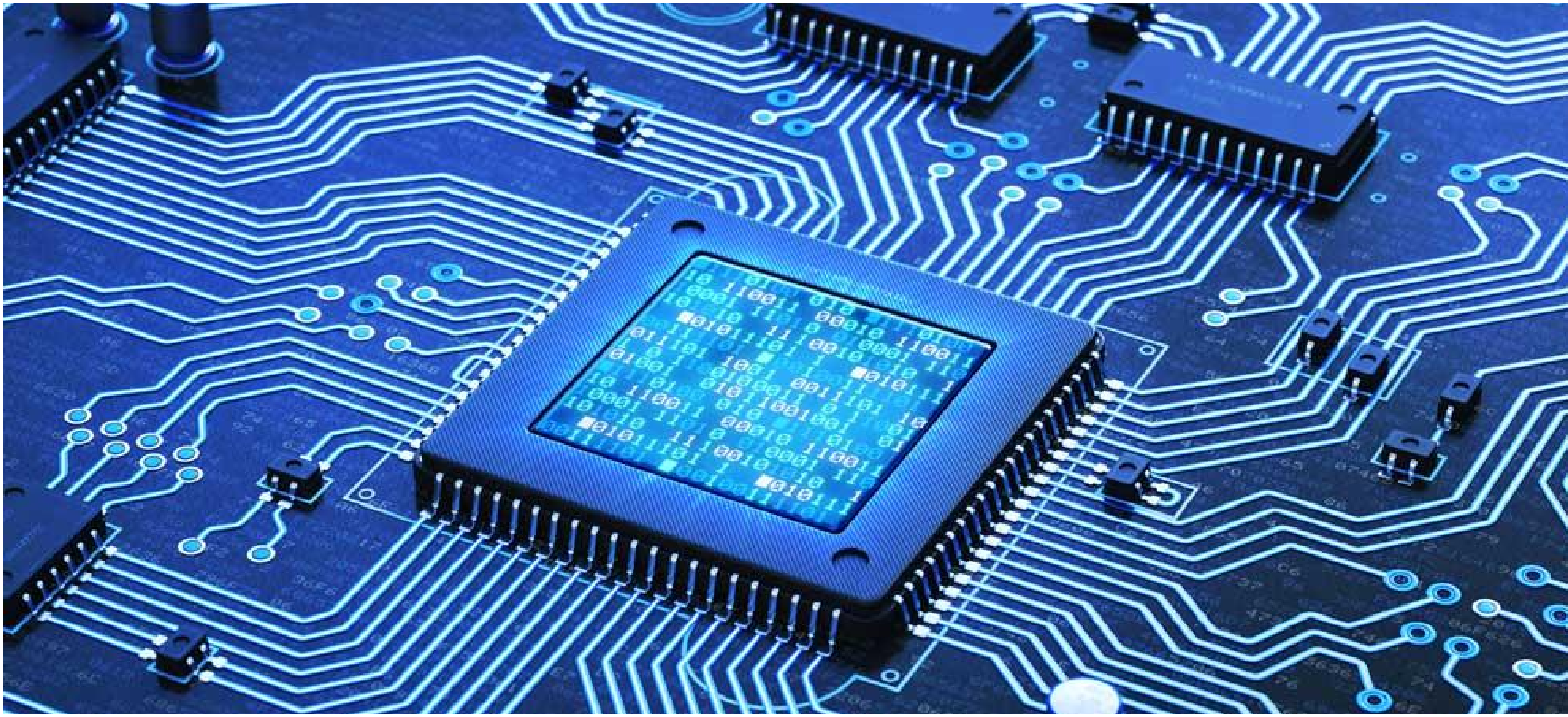
Ordinary Algebra Vs Boolean Algebra

Ordinary Algebra	Boolean Algebra
Letter symbols can take on **any** number of **values** including **infinity**	Letter symbols can take **on either of two values**, that is, **0 and 1**

Ordinary Algebra Vs Boolean Algebra

Repeated again for the pdf copy

Ordinary Algebra	Boolean Algebra
Letter symbols can take on **any** number of **values** including **infinity**	Letter symbols can take **on either of two values**, that is, **0 and 1**
Values assigned to a variable have a **numerical significance**	**Values** assigned to a variable have a **logical significance**
'.' and **'+'** are respectively the signs of **multiplication** and **addition**	**'.'** means an **AND** operation and **'+'** means an **OR** operation
A+B is **read** as **A plus B**	**A+B** is **read** as **A OR B** **Basic logic** operations are **AND**, **OR** and **NOT**
	Captures both **logic operations** and **set operations** such as **intersection**, **union** and **complement**



Boolean Expressions

Variables, Literals and Terms

- **Variables** are different **symbols** in a **Boolean expression**
- **They** may **take** on the **values '0' or '1'**
- **Complement** of a variable is **not** considered as a **separate variable**
- **Each occurrence** of a **variable** or **its complement** is called a **literal**

- A **term** is the **expression formed** by **literals and operations** at **one level**.
- Each **term requires** a **gate** and **each variable within the term** designates an **input to the gate**

$$\overline{A} + A.B$$

Quiz:

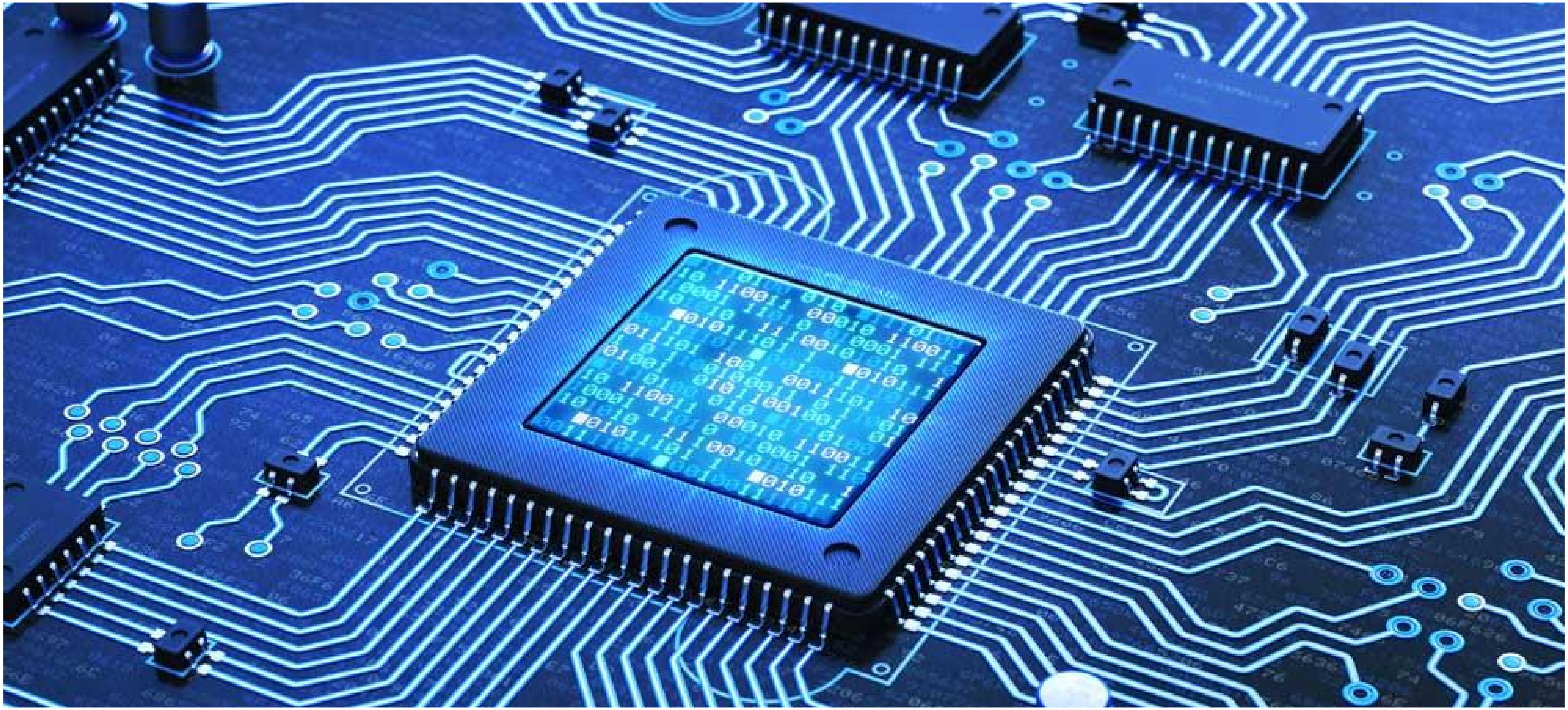
Variables : 2
Literals : 3
Terms : 2

Quiz 1: Variables, Literals and Terms

- How many **Variables**, **literals** and **Terms** are there in these **two Boolean expressions**?
- How many **gates** are needed to construct them?

Boolean Expressions	Variables	Literals	Terms	Gates
$F_2 = x'y'z + x'yz + xy'$	**3**	**8**	**3**	**NOT : 2** **AND : 3** **OR : 1**
$F_2 = xy' + x'z$	**3**	**4**	**2**	**NOT : 2** **AND : 2** **OR : 1**

Note: Assume both 2-input and 3-input gates are available to construct these expressions, without minimization.



Equivalent and Complement of Boolean Expressions

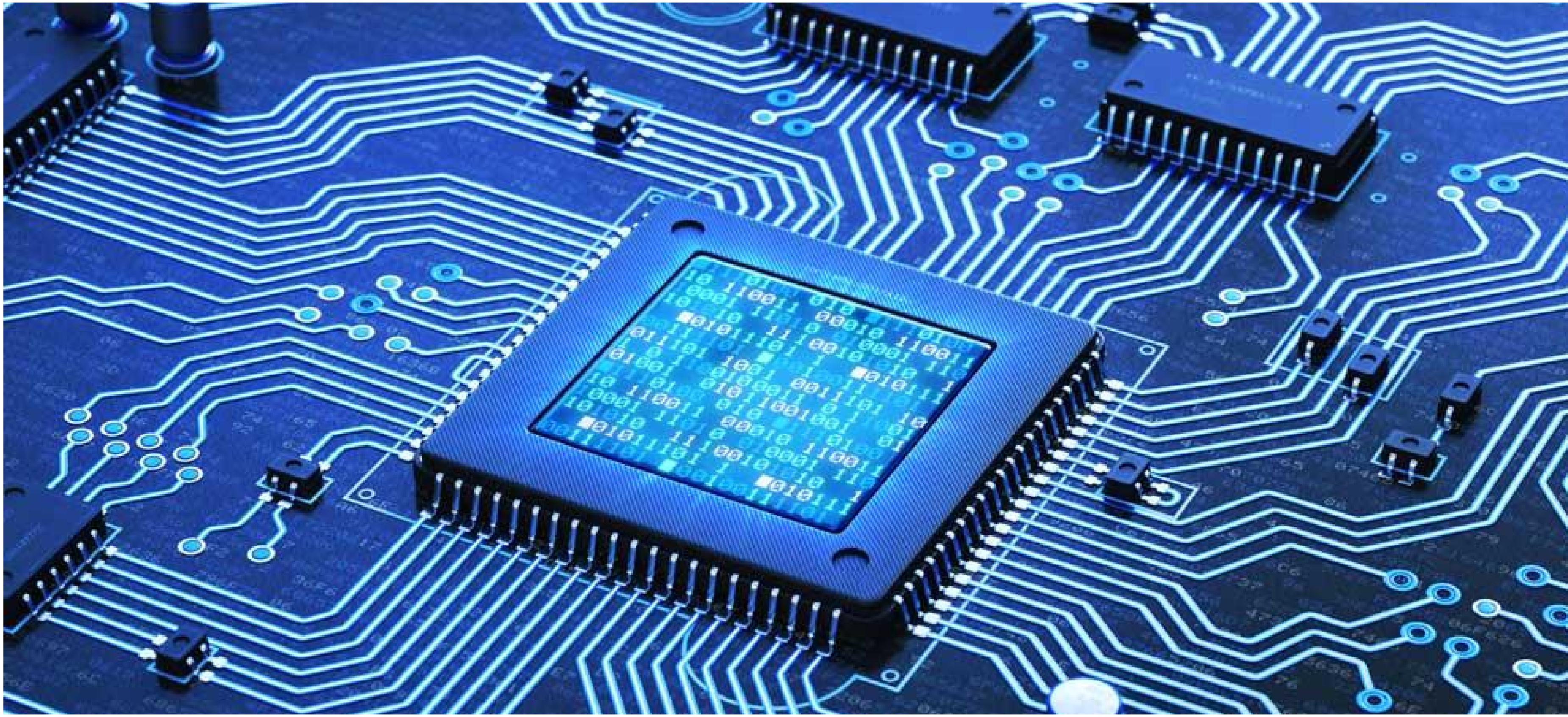
Equivalent and Complement of Boolean Expressions

- **Two** given **Boolean expressions** are said to be **equivalent**
  - If **one** of them **equals '1'** only when the **other equals '1'** and also **one equals '0'** only when the **other equals '0'**
- They are said to be the **complement** of each other
  - If **one** expression **equals '1'** only when the **other equals '0'**, and **vice versa**
- The **complement** of a given **Boolean expression** is **obtained** by
  - **Complementing each literal**,
  - **Changing** all **'.' to '+'** and all **'+' to '.'**, all **0s to 1s** and all **1s to 0s**.

Quiz 2: Find Complements

- Find the **complement** of below **Boolean expressions**:

Boolean Expression	Complement
$\overline{A}.B+A.\overline{B}$	$(A+\overline{B}).(\overline{A}+B)$
$(A+B).(\overline{A}+\overline{B})$	$\overline{A}.\overline{B}+A.B$
$[(A.\overline{B}+\overline{C}).D+\overline{E}].F$	$[(\overline{A}+B).C+\overline{D}].E+\overline{F}$



Dual of Boolean Expressions

Dual of Boolean Expressions

The **dual** of a Boolean expression is obtained by **replacing**

- All **'.' operations** with **'+' operations**
- All **'+' operations** with **'.' operations**,
- All **0s** with **1s** and all **1s** with **0s** and
- Leaving all **literals unchanged**

Quiz 3: Find Dual

- Find the **dual** of below **Boolean expressions**:

Boolean Expression	**Dual**
$\overline{A}.B + A.\overline{B}$	$(\overline{A} + B).(A + \overline{B})$
$(A + B).(\overline{A} + \overline{B})$	$A.B + \overline{A}.\overline{B}$
$A.\overline{B} + B.\overline{C} + C.\overline{D}$	$(A + \overline{B}).(B + \overline{C}).(C + \overline{D})$

Quiz 4: Find Dual and Complement

- Find the **dual** and **Complement** of below **Boolean expression**:

Boolean Expression	$[(A.\overline{B}+\overline{C}).D+\overline{E}].F$
Dual	$[(A+\overline{B}).\overline{C}+D].\overline{E}+F$
Complement	$[(\overline{A}+B).C+\overline{D}].E+\overline{F}$

What is Dual of a Boolean Expression?

The principle of **duality** pronounces that given an **expression** which is always **valid** in boolean algebra, the **dual expression** is **also** always **valid**

- For example:
  - **A (B + C) = A . B + A . C**
  - It's **dual** is also **valid**
  - **A + (B . C) = (A + B) . (A + C)**

Rules of Boolean Algebra

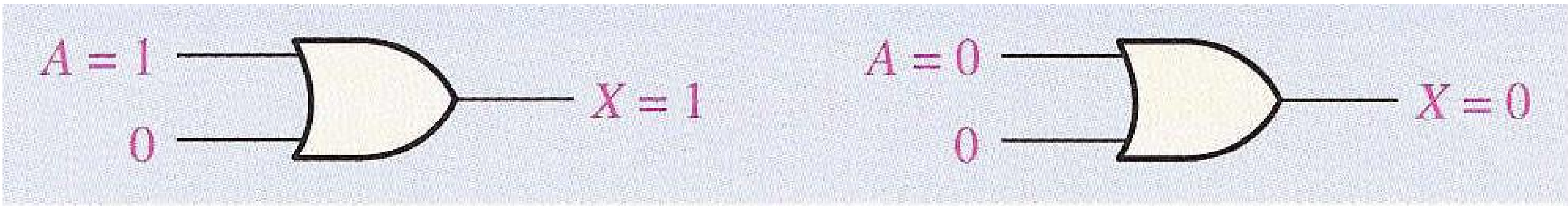
- **A, B or C** can represent a single variable or a combination of variables

1. $A + 0 = A$	7. $A \cdot A = A$
2. $A + 1 = 1$	8. $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	9. $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	10. $A + AB = A$
5. $A + A = A$	11. $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	12. $(A + B)(A + C) = A + BC$

- These **12 rules** are useful in **simplifying** Boolean expressions

Rule 1

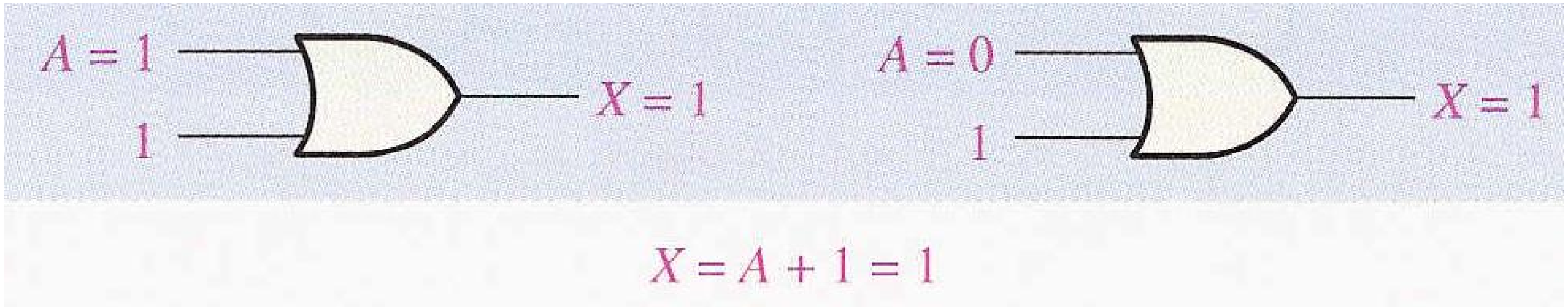
- **$A + 0 = A$**



$X = A + 0 = A$

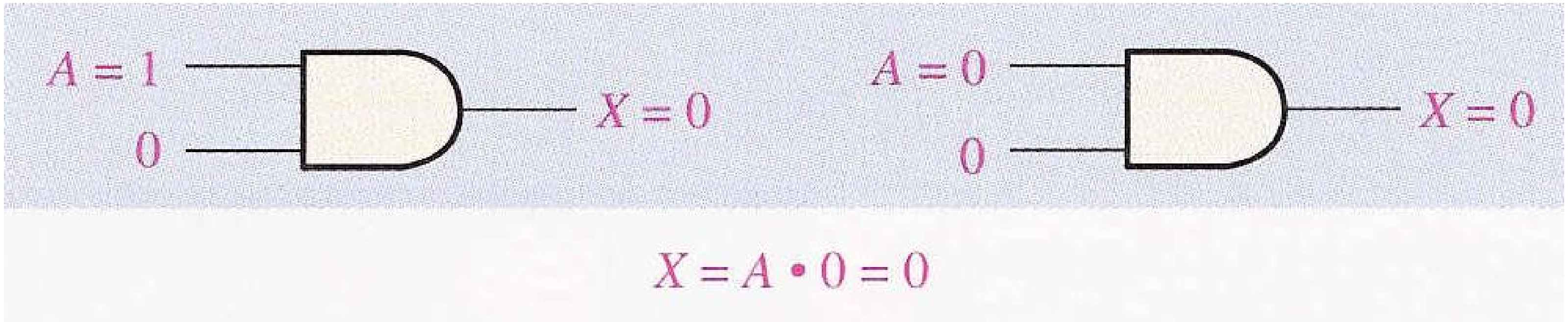
Rule 2

- **A + 1 = 1**



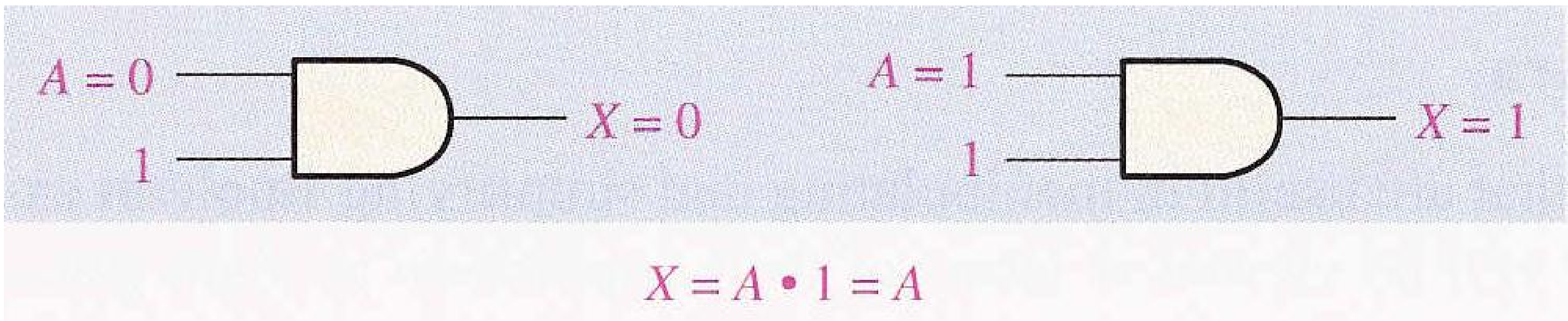
Rule 3

- **A . 0 = 0**



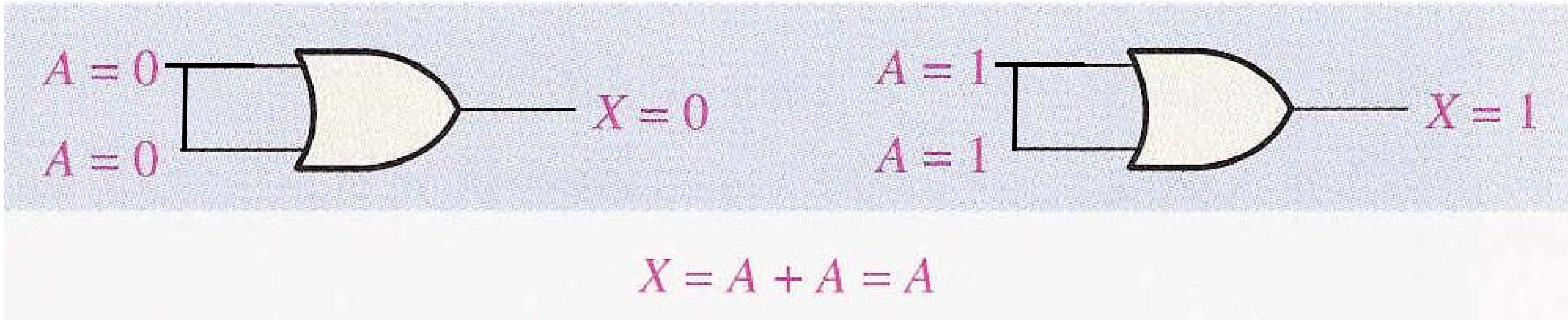
Rule 4

- **A . 1 = A**



Rule 5

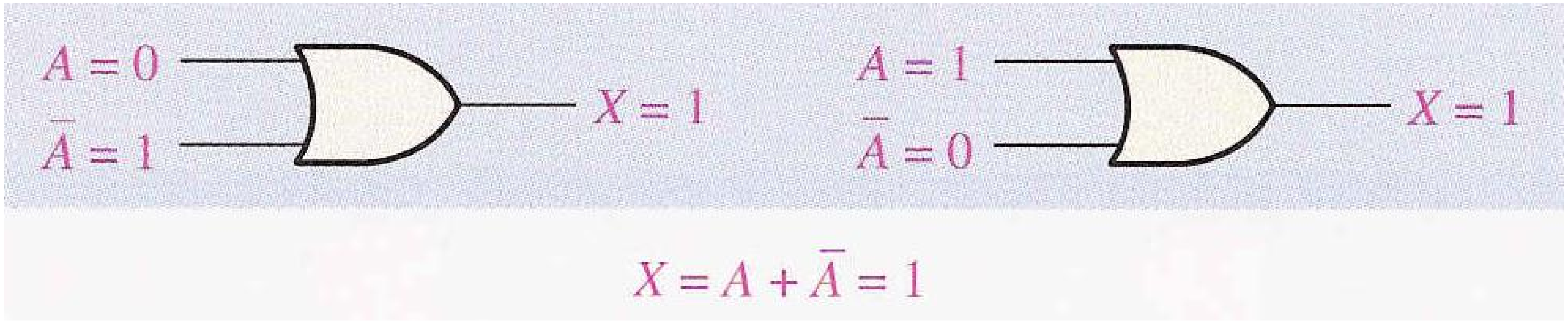
- **A + A = A (Idempotent or Identity Law)**



- Valid for any number of the same input:
- **A + A + A + + + + A = A**

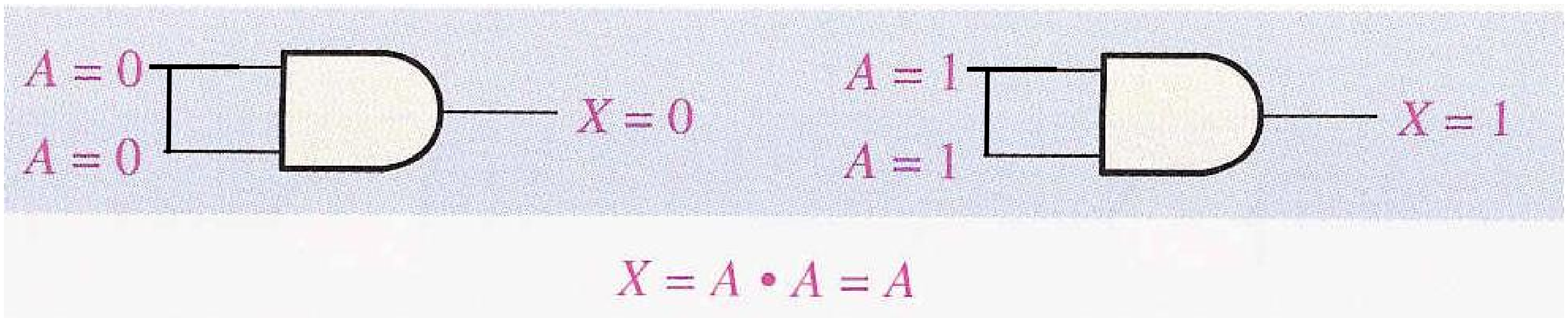
Rule 6

- $A + \overline{A} = 1$



Rule 7

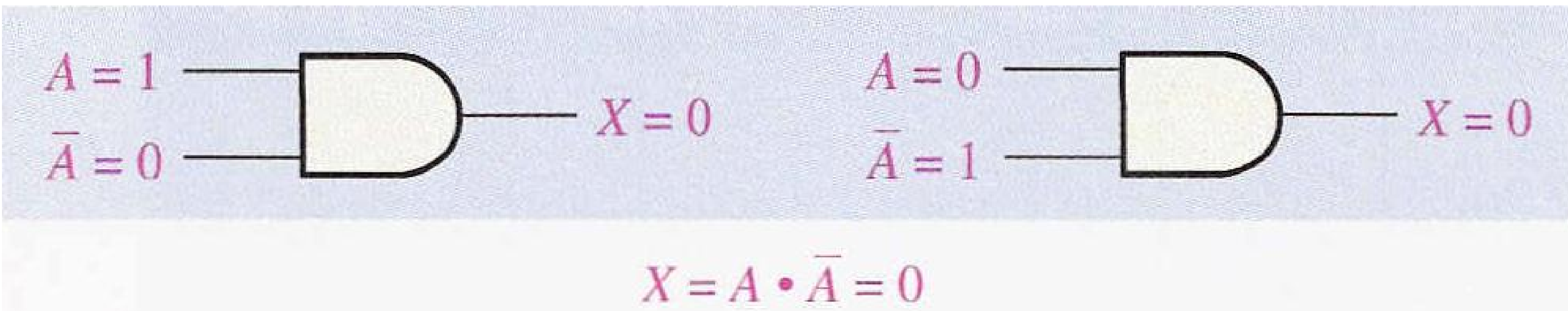
- **A . A = A (Idempotent or Identity Law)**



- Valid for **any number** of the **same input**:
- **A . A . A ….. A = A**

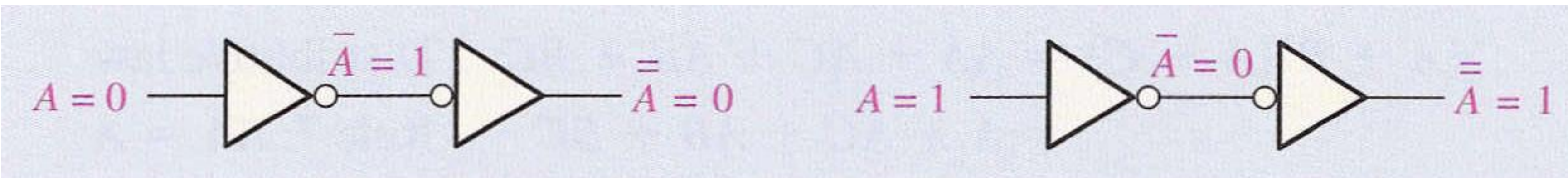
Rule 8

- **$A \cdot \overline{A} = 0$**



Rule 9

- $\overline{\overline{A}} = A$ **(Involution law)**



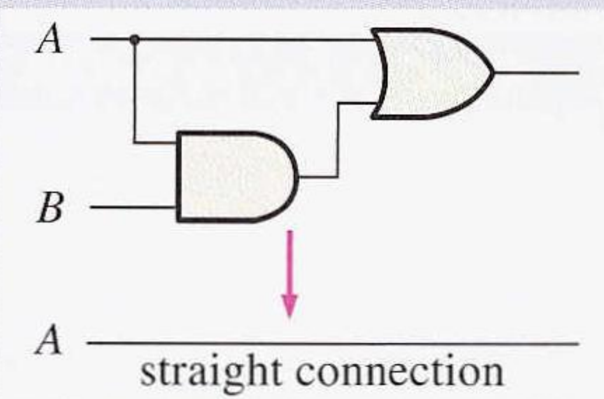
$$\overline{\overline{A}} = A$$

Rule 10

- A + AB = A   (Absorption or Redundancy Law)
- A + AB = A (1 + B) : Factoring (Distributive law)
-             = A . 1   : By Rule 2, (1 + B) = 1
-             = A   : By Rule 4, A . 1 = A
- A + AB = A

A	*B*	*AB*	*A + AB*
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

equal



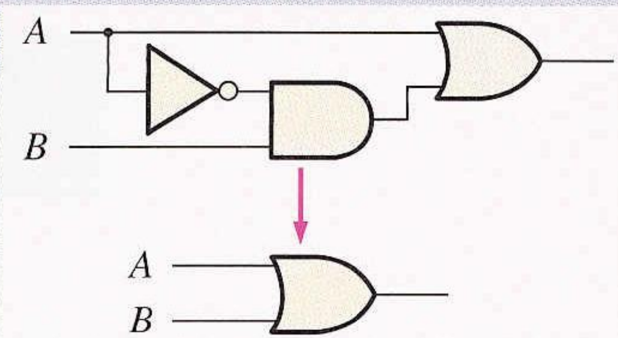
Rule 11

- $\mathbf{A + \overline{A}B = A + B}$

$A + \overline{A}B = (A + AB) + \overline{A}B$ Rule 10: $A = A + AB$

A	B	$\overline{A}B$	$A + \overline{A}B$	$A + B$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

equal



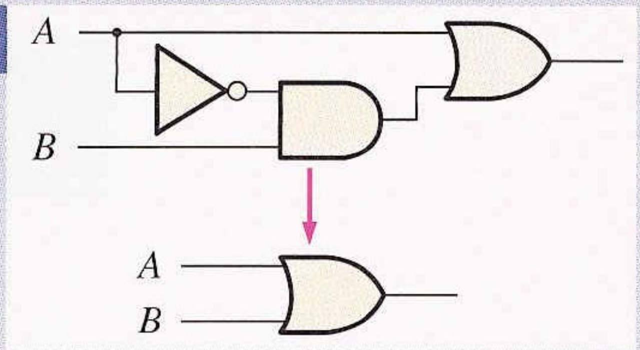
Rule 11

- $\mathbf{A + \overline{A}\,B = A + B}$

$$
\begin{aligned}
A + \overline{A}B &= (A + AB) + \overline{A}B && \text{Rule 10: } A = A + AB \\
&= (AA + AB) + \overline{A}B && \text{Rule 7: } A = AA \\
&= AA + AB + A\overline{A} + \overline{A}B && \text{Rule 8: adding } A\overline{A} = 0 \\
&= (A + \overline{A})(A + B) && \text{Factoring} \\
&= 1 \cdot (A + B) && \text{Rule 6: } A + \overline{A} = 1 \\
&= A + B && \text{Rule 4: drop the 1}
\end{aligned}
$$

A	B	$\overline{A}B$	$A + \overline{A}B$	$A + B$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

equal



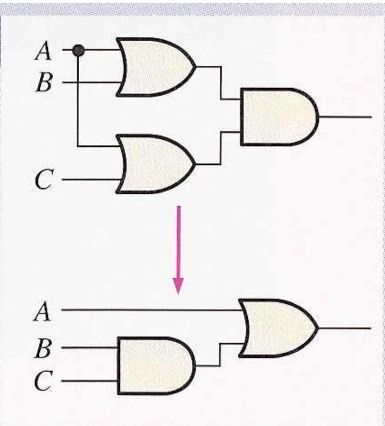
Rule 12

- **(A + B) . (A + C) = A + BC**

$$(A + B)(A + C) = AA + AC + AB + BC \quad \text{Distributive law}$$

A	B	C	A + B	A + C	(A + B)(A + C)	BC	A + BC
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

equal



Rule 12

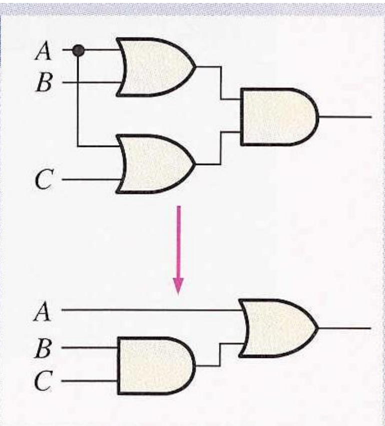
Repeated again for the pdf copy

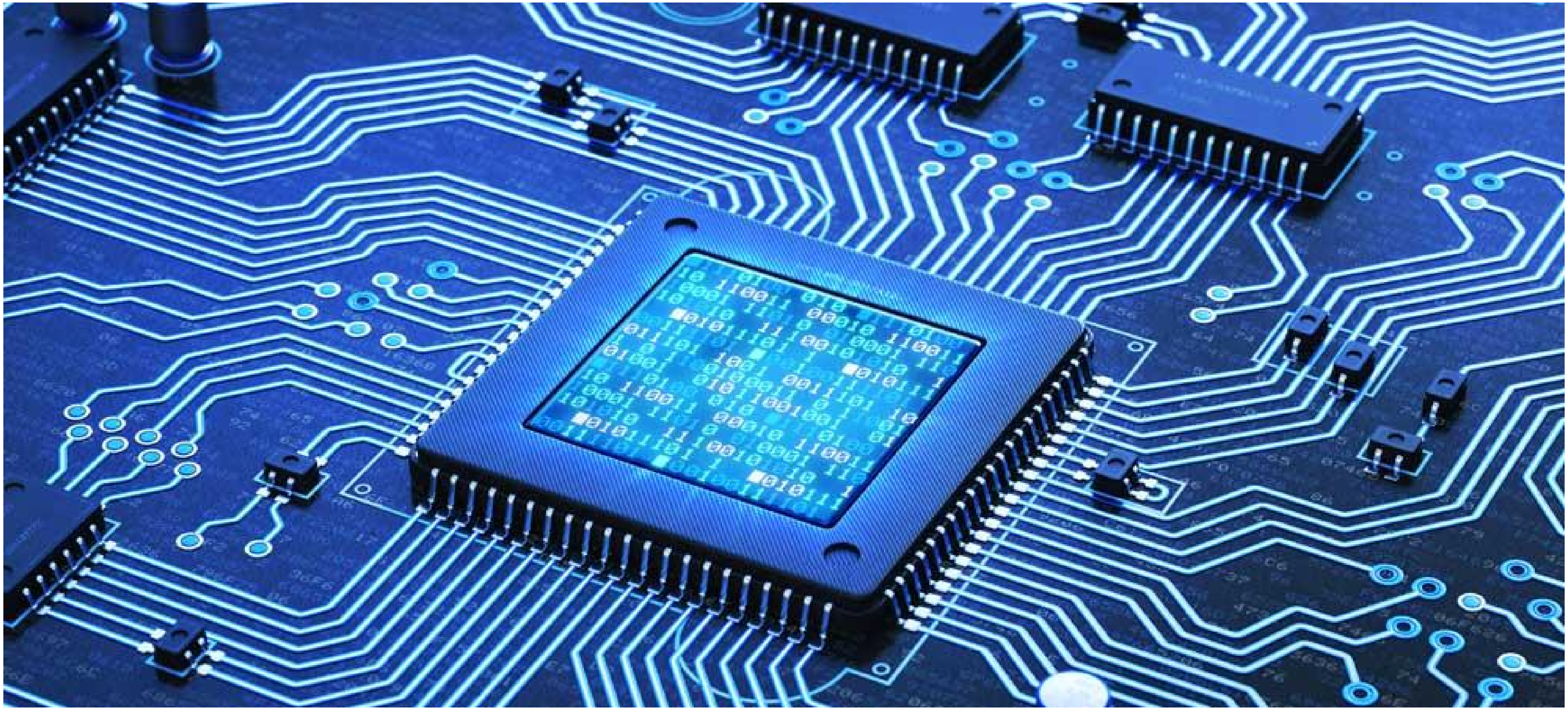
- **(A + B) . (A + C) = A + BC**

$(A+B)(A+C)$	$= AA + AC + AB + BC$	Distributive law
	$= A + AC + AB + BC$	Rule 7: $AA = A$
	$= A(1 + C) + AB + BC$	Factoring (distributive law)
	$= A \cdot 1 + AB + BC$	Rule 2: $1 + C = 1$
	$= A(1 + B) + BC$	Factoring (distributive law)
	$= A \cdot 1 + BC$	Rule 2: $1 + B = 1$
	$= A + BC$	Rule 4: $A \cdot 1 = A$

A	B	C	A + B	A + C	(A + B)(A + C)	BC	A + BC
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

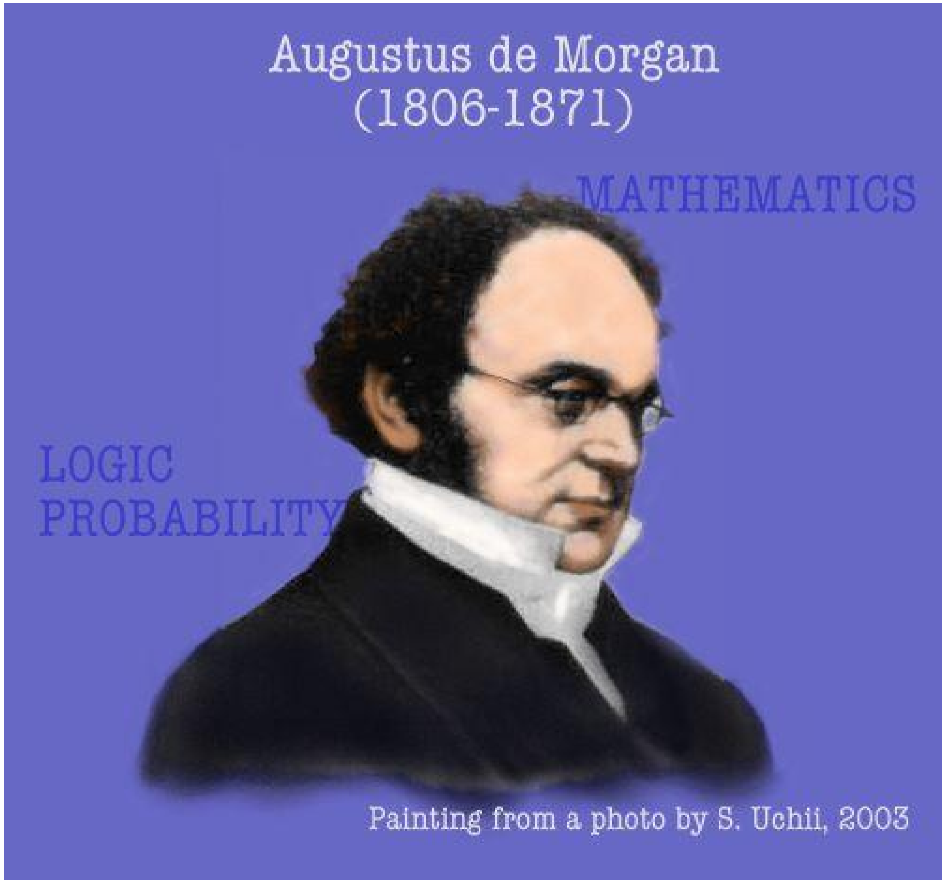
equal





DeMorgon’s Theorems

Augustus De Morgan



Augustus De Morgan,
(born June 27, 1806, Madura, India)
English mathematician and logician whose major **contributions** to the study of logic include the formulation of **De Morgan's** laws

DeMorgan, a mathematician who knew Boole, proposed two important theorems of Boolean Algebra

37

DeMorgon's Theorem 1

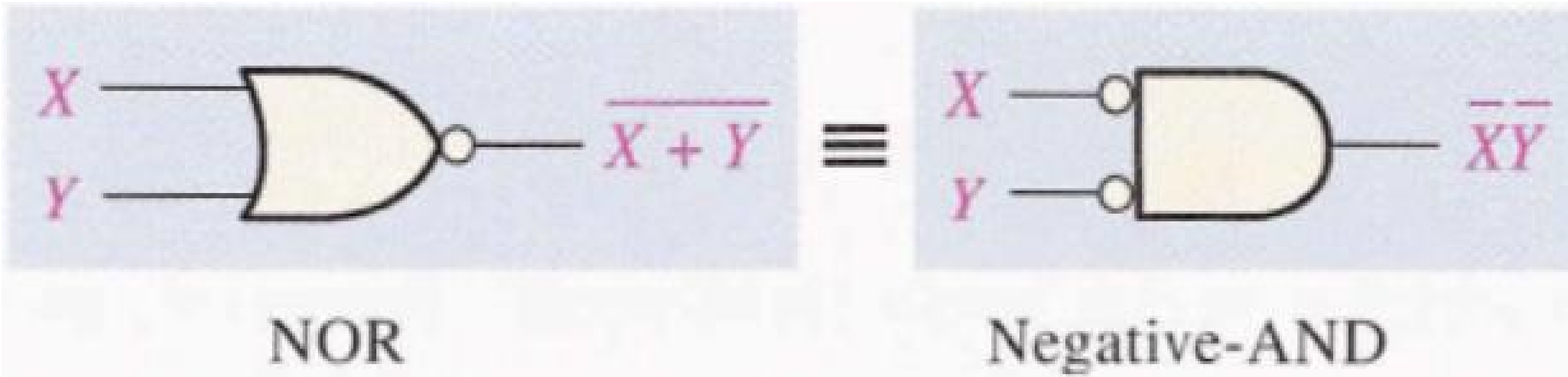
- **Theorem 1:**
- The **complement** of a **sum of variables** is **equal** to the **product of the complements** of the variables

$$\overline{X + Y} = \overline{X}\,\overline{Y}$$

- It can also be stated as:
- The **complement** of two or more **ORed variables** is **equivalent** to the **AND** of the **complements** of the **individual variables**.
- Valid for any number of variables:

$$[\overline{X_1 + X_2 + X_3 + \ldots + X_n}] = \overline{X_1}.\overline{X_2}.\overline{X_3}.\ \ldots\ .\overline{X_n}$$

DeMorgon’s Theorem 1



Inputs		Output	
X	Y	$\overline{X+Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

39

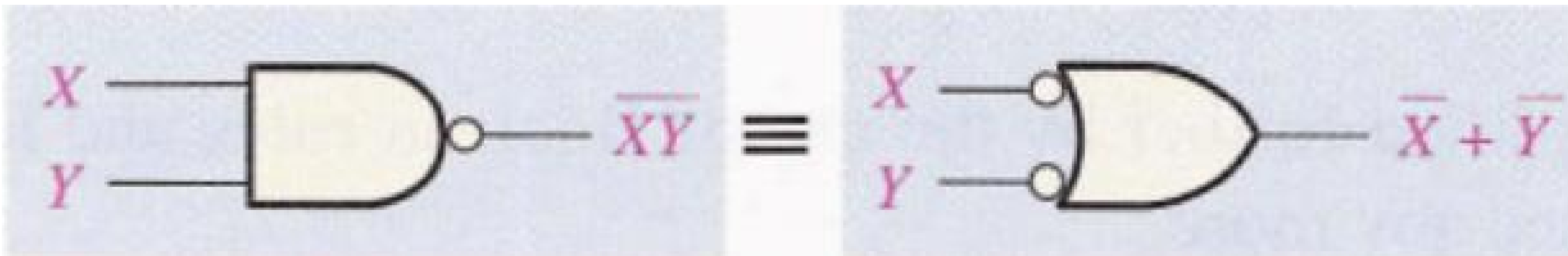
DeMorgon's Theorem 2

- **Theorem 2:**
- The **complement** of a **product of variables** is **equal** to the **sum of the complements** of the variables $\overline{XY} = \overline{X} + \overline{Y}$
- It can also be stated as:
- The **complement** of two or more **ANDed variables** is **equivalent** to the **OR** of the **complements** of the **individual variables**.
- Valid for any number of variables:

$$[\overline{X_1.X_2.X_3. \ . \ . \ . \ .X_n}] = [\overline{X_1} + \overline{X_2} + \overline{X_3} + . \ . \ . \ + \overline{X_n}]$$

40

DeMorgon's Theorem 2



NAND   Negative-OR

Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Home work:
Proof for Theorem 2

41

Session 2.4: Summary

- Introduction to Boolean Algebra
- Ordinary Algebra Vs Boolean Algebra
- Boolean Expressions
  - Variables, literals and terms
  - Complement and Dual
- DeMargon’s Theorems