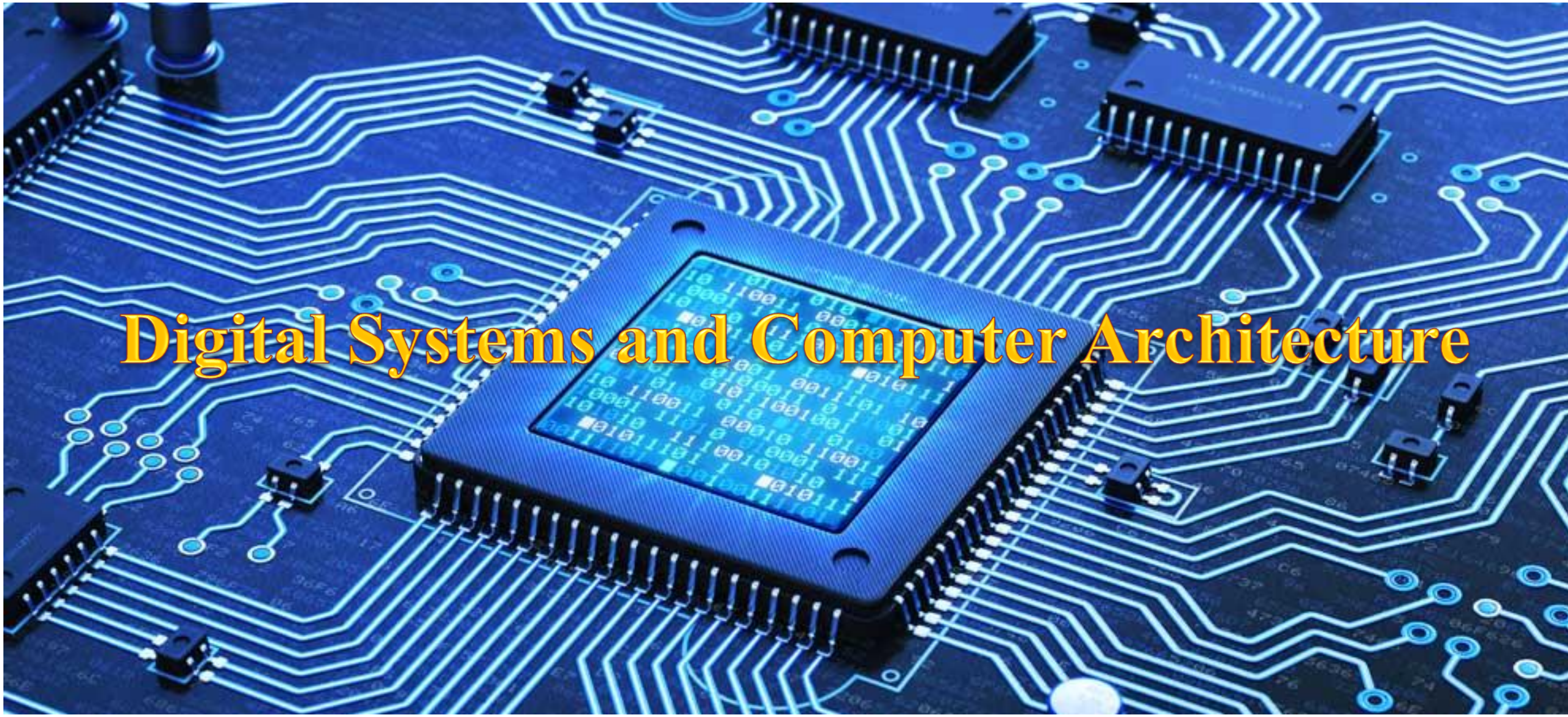


Digital Systems and Computer Architecture

Session 2.5

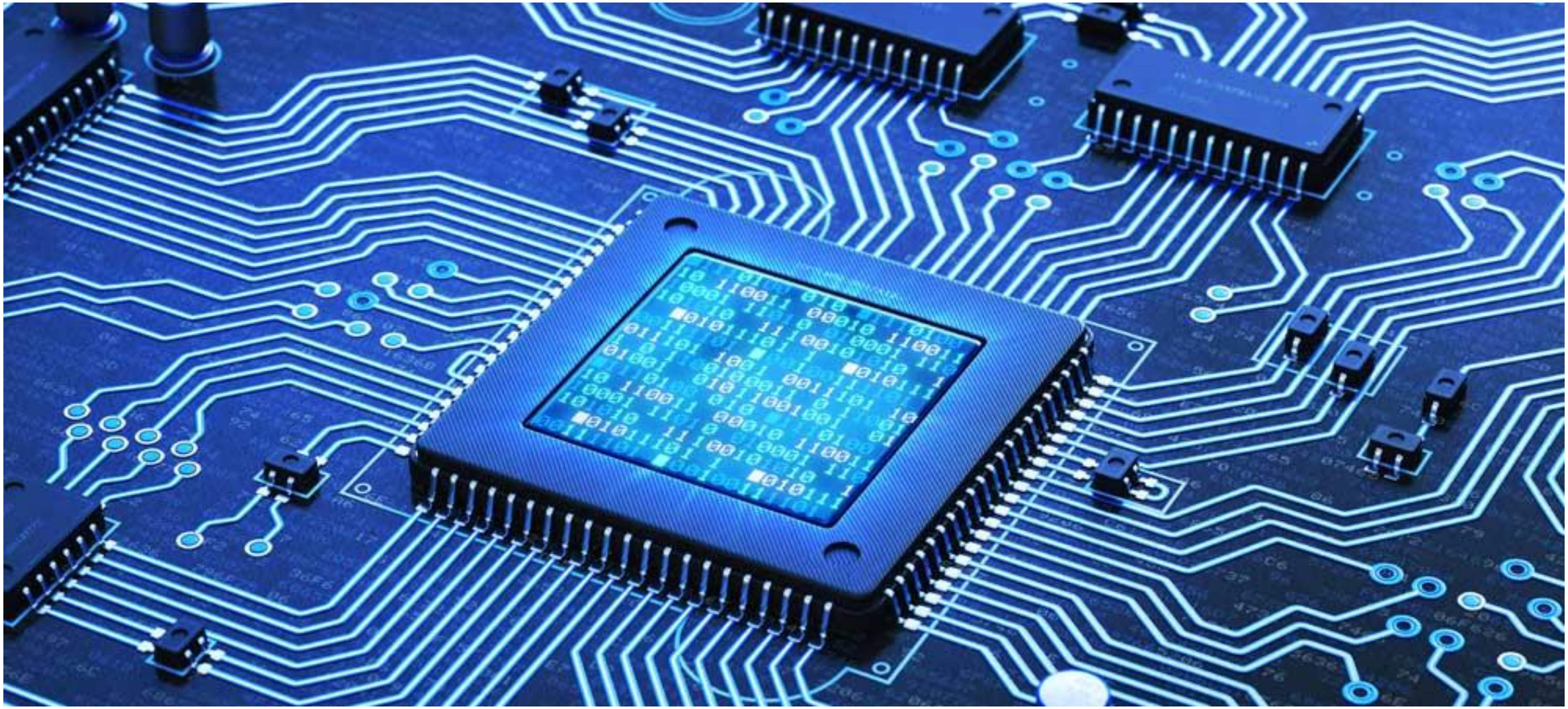
Module 2

Mouli Sankaran

SOP and POS forms of Boolean Expressions

Session 2.5: Focus

- Simplification Techniques
- Simplification using Boolean Algebra
  - Quiz 1 to 3
- Two forms of Boolean Expressions
  - Sum-of-Products (minterms)
  - Products-of-Sums (maxterms)
  - Canonical notation
  - Example
- Home Work



Introduction to Boolean Algebra

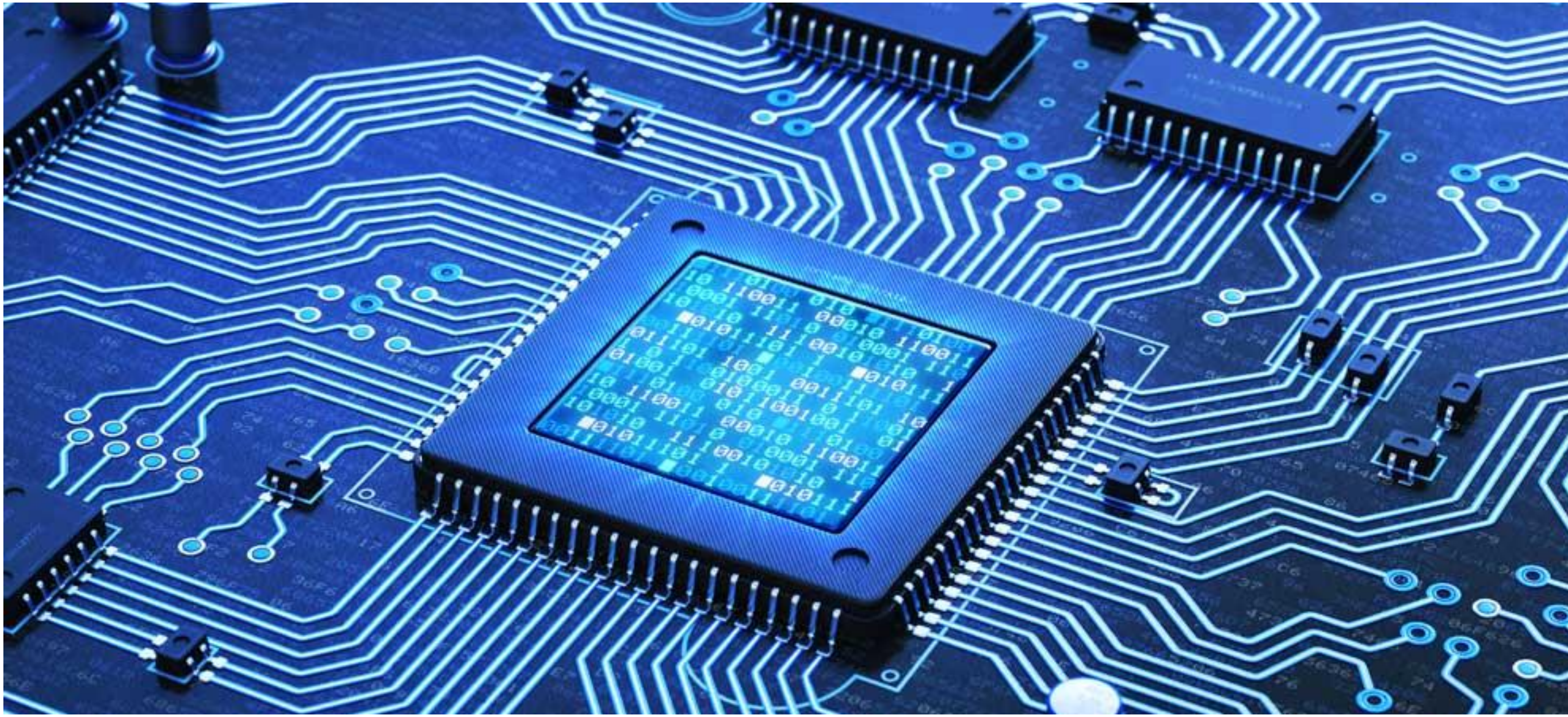
Recap: Variables, Literals and Terms

- **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**

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

Variables : 2
Literals : 3
Terms : 2



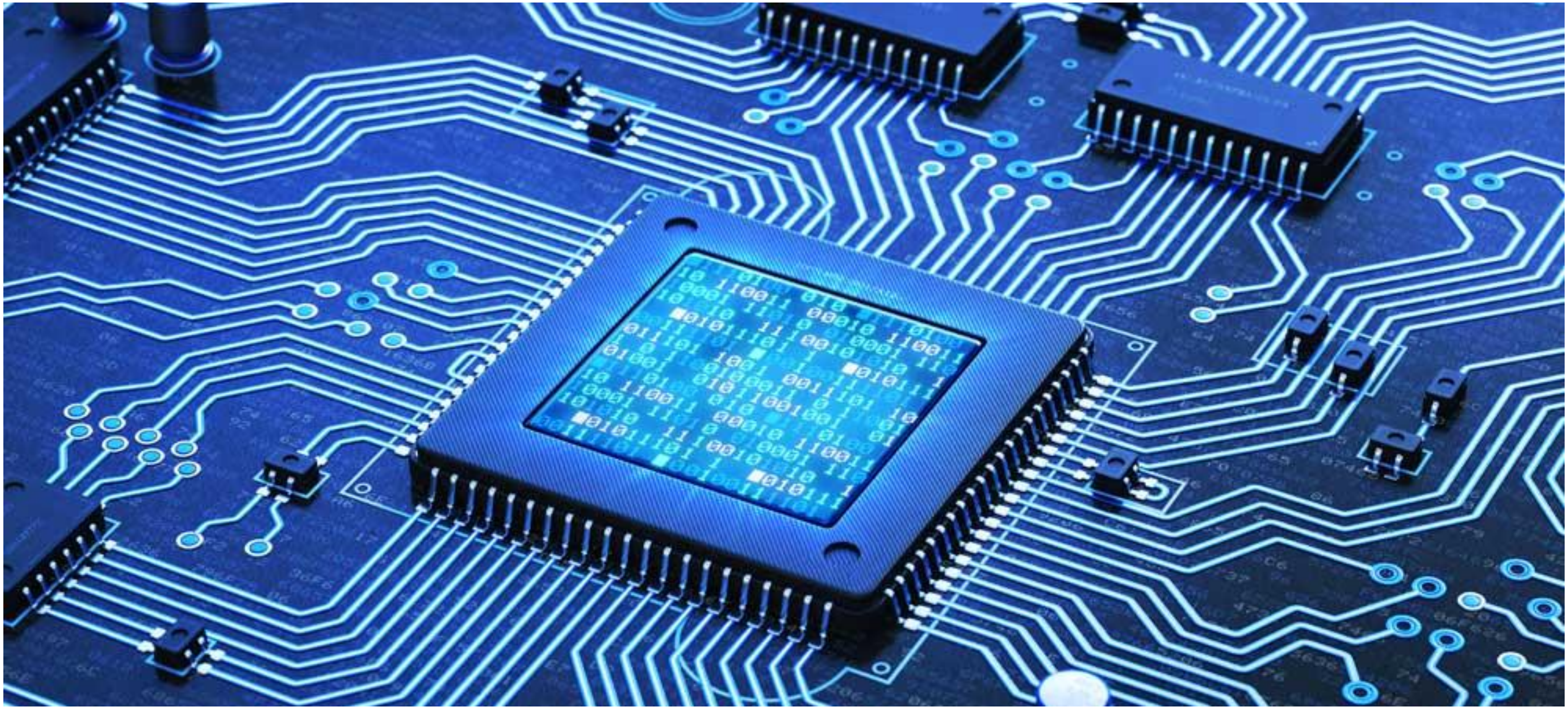
Simplification Techniques

Need for Simplification

- Simplification means simplifying or more precisely minimizing a given complex Boolean expression
- The **primary objective** of all simplification procedures is **to obtain** an expression that has the **minimum number of terms**
  - A simplified Boolean expression uses the fewest gates possible to implement a given expression
  - If gates are reduced, the complexity of the implementation as well as power consumption are also reduced
- Obtaining an expression with the **minimum** number of **literals** or **symbols** is usually the **secondary objective**

Simplification Techniques

- Simplification using Boolean algebra
- Other two popular methods are:
  - The Quine–McCluskey tabular method;
  - The Karnaugh map method



Simplification
Using
Boolean Algebra

Simplification Using Boolean Algebra

- The approach taken in this section is to
  - Use the basic laws, rules and theorems of Boolean Algebra
  - To manipulate and simplify an Expression
  - Which will result in less number of gates/literals without changing the function of the original Boolean Expression
- A **simplified** Boolean Expression **uses the fewest gates** possible **to implement** a given expression

Quiz 1: Simplify the Boolean Expression

- Simplify the expression: AB + A(B+C) + B(B+C)
- **Step 1**: Apply Distributive law to 2nd and 3rd items
  - AB + **AB + AC + BB + BC**
- **Step 2**: Apply **rule 7** (BB = B)
  - AB + AB + AC + **B** + BC
- **Step 3**: Apply **rule 5** (AB + AB = AB)
  - **AB** + AC + B + BC
- **Step 4**: Apply **rule 10** (B + BC = B)
  - AB + AC + **B**
- **Step 5**: Apply **rule 10** (AB + B = B)
  - AC + **B**

Note: The **terms in Bold** are the ones **affected** by the **rules** in **each step.**

Quiz 2: Simplify the Boolean Expression

- Simplify the expression: $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$
- **Step 1**: Apply Distributive law to the terms within brackets
  - $(A\overline{B}C + A\overline{B}BD + \overline{A}\,\overline{B})C$
- **Step 2**: Apply **rule 8 (B $\overline{\mathbf{B}}$ = 0)**
  - $(A\overline{B}C + A\cdot 0\cdot D + \overline{A}\,\overline{B})C$
- **Step 3**: Apply **rule 3** (A . 0 . D = 0)
  - $(A\overline{B}C + 0 + \overline{A}\,\overline{B})C$
- **Step 4**: Apply **rule 1** (drop the 0)
  - $(A\overline{B}C + \overline{A}\,\overline{B})C$
- **Step 5**: Apply Distributive law
  - $A\overline{B}CC + \overline{A}\,\overline{B}C$

Quiz 2 … contd

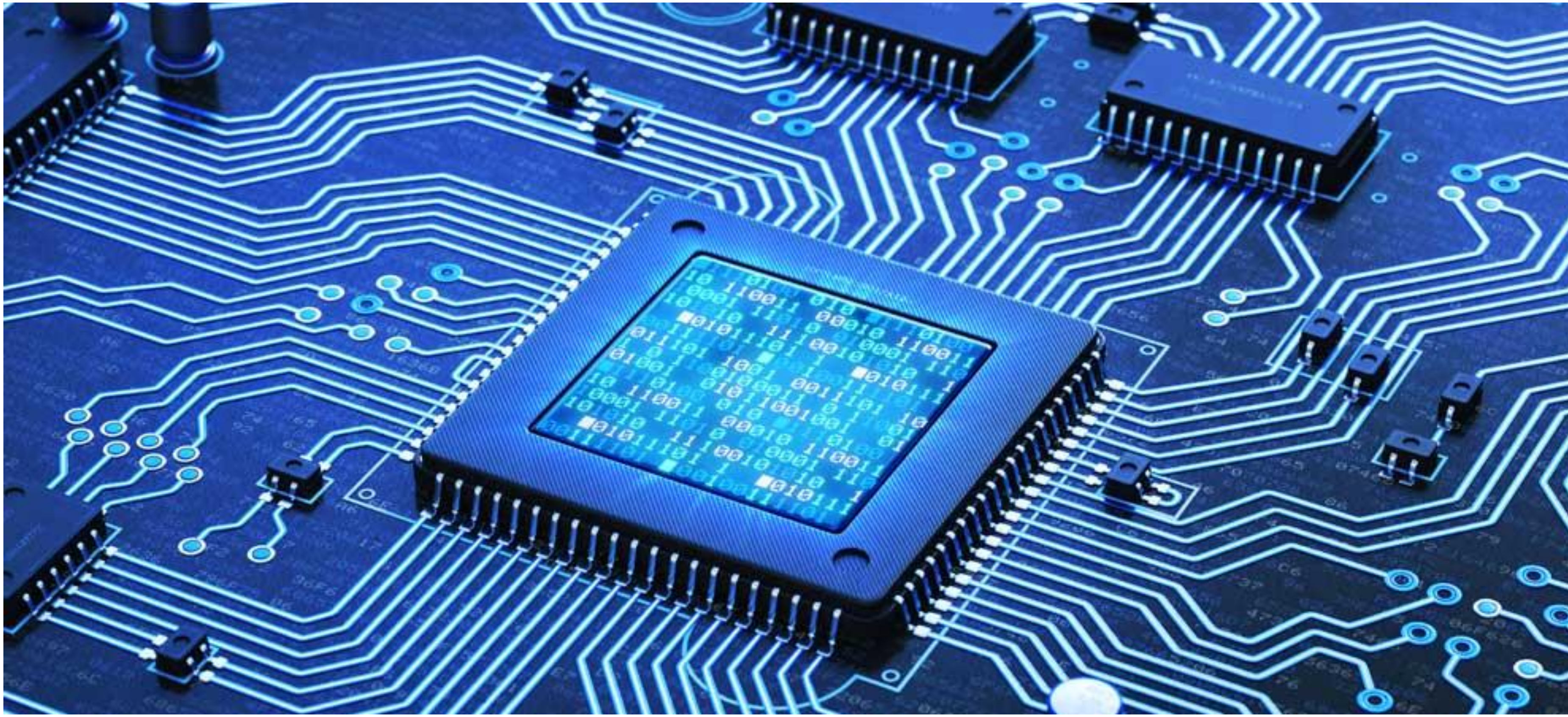
- Simplify the expression: $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$
- **From Step 5:** $A\overline{B}CC + \overline{A}\,\overline{B}C$
- **Step 6**: Apply **rule 7 (**CC = C) to the first term
  - $A\overline{B}C + \overline{A}\,\overline{B}C$
- **Step 7**: Factor out $\overline{B}$ **C**
  - $\overline{B}C(A + \overline{A})$
- **Step 8**: Apply **rule 6 (A + $\overline{A}$ = 1)**
  - $\overline{B}C \cdot 1$
- **Step 9**: Apply **rule 4** (drop the 1)
  - $\boxed{\overline{B}C}$

Quiz 3: Simplify the Boolean Expression

- Simplify the expression: $\overline{AB + AC} + \overline{A}\,\overline{B}C$
- **Step 1**: Apply DeMorgon's theorem to the first term
  - $(\overline{AB})(\overline{AC}) + \overline{A}\,\overline{B}C$
- **Step 2**: Apply DeMorgon's theorem to each term in parenthesis
  - $(\overline{A} + \overline{B})(\overline{A} + \overline{C}) + \overline{A}\,\overline{B}C$
- **Step 3**: Apply Distributive law to the terms within parenthesis
  - $\overline{A}\,\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$
- **Step 4**: Apply **rule 7** $(\overline{A}\,\overline{A} = \overline{A})$ to the first term
  - $\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$
- **Step 5**: Apply **rule 10 (A + AB = A)** to the 3rd and last terms
  - $\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$

Quiz 3 … contd.

- From **Step 5:** $\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$
- **Step 6**: Apply **rule 10 (A + AB = A)** to the 1st and 2nd terms
  - $\overline{A} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$
- **Step 7**: Apply **rule 10 (A + AB = A)** to the 1st and 2nd terms
  - $\boxed{\overline{A} + \overline{B}\,\overline{C}}$



Two Forms
of
Boolean Expressions

Two Forms of Boolean Expressions

- Boolean Expressions are normally expressed in one of these forms for Simplification
  - **Sum-of-Products (SOP)**
  - **Product-of-Sums (POS)**
- Let us see them before studying other Simplification techniques

Sum-of-Products

- **Sum-of-products** expression contains the **sum** of different **terms**, with each term being either a **single literal** or a **product of more than one literal**
- It can be obtained from the **truth table** directly by considering those **input combinations** that produce a **logic '1'** at the **output**.
- Each such input combination produces a term, which are called **minterms**
- A **minterm** is a combination of variables that produces a **1** in the function and then taking the **OR** of **all those terms**

Example: Sum-of-Products

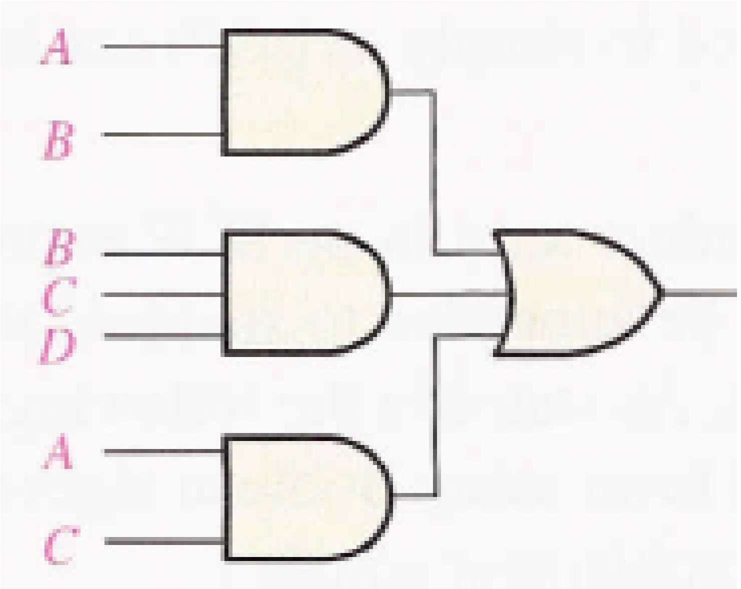
- For Example **Truth table** below can be expressed as:
  - Only those terms which produce an **output** of **1**

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$Y = \overline{A}\,.\overline{B}.\overline{C} + \overline{A}.B.C + A.B.\overline{C} + A.\overline{B}.C$$

Implementation of Sum-of-Products

- **Example: AB + BCD + AC** can be implemented as shown below:



AND-OR

Quiz 4: Is this in SOP or POS form?

- $\overline{ABC} + A\overline{B}C$: **Neither in SOP nor POS form!!**

For an expression to properly follow the **SOP** or **POS** canonical form, **no complementation bar** should **cover more than one variable**!

Minterms of Three Binary Variables

- **All** the **combinations** of **Sum-of-Products** of **three variables** and their **nomenclature** are given below

			Minterms	
x	**y**	**z**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0
0	0	1	$x'y'z$	m_1
0	1	0	$x'yz'$	m_2
0	1	1	$x'yz$	m_3
1	0	0	$xy'z'$	m_4
1	0	1	$xy'z$	m_5
1	1	0	xyz'	m_6
1	1	1	xyz	m_7

Product-of-Sums

- **Product-of-Sum**s expression contains the **product** of different **terms**, with each term being either a **single literal** or a **sum of more than one literal**
- It can be obtained from the **truth table** directly by considering those **input combinations** that produce a **logic '0'** at the **output**.
- Each such input combination produces a term, which are called **maxterms**
- A **maxterm** is a combination of variables that produces a **0** in the function and then taking the **AND** of **all those terms**
- Boolean functions expressed as a **sum of minterms** or **product of maxterms** are said to be in **canonical form**

Example: Product-of-Sums

- For Example **Truth table** below can be expressed as:

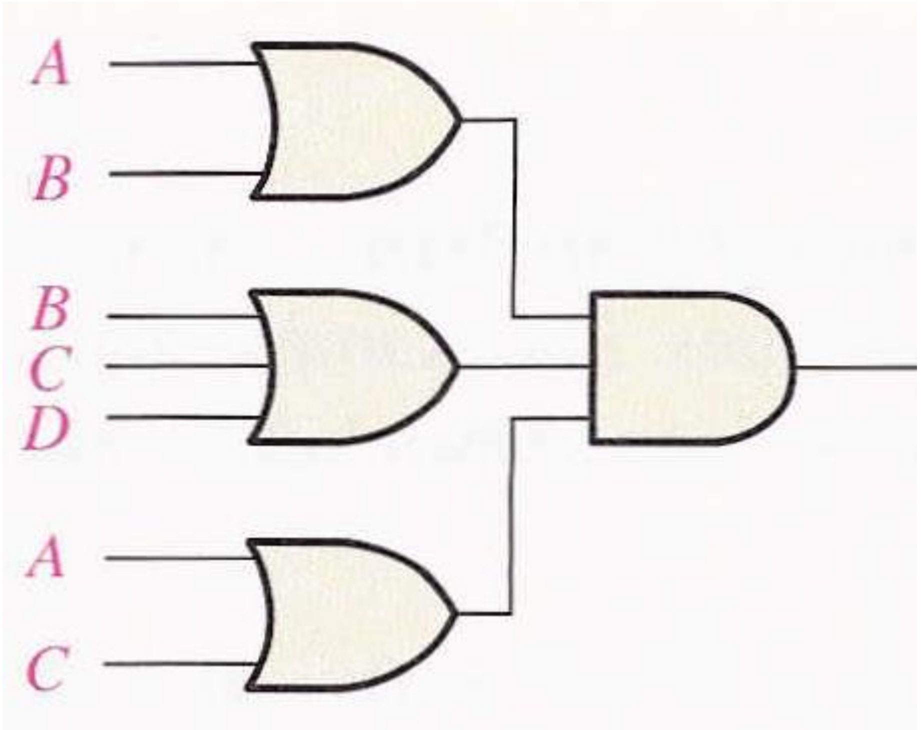
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$\mathbf{Y} = (A + B + \overline{C}).(A + \overline{B} + C).(\overline{A} + B + C).(\overline{A} + \overline{B} + \overline{C})$$

Implementation of Product-of-Sums

- **Example: (A+ B) . (B+C+D) . (A+C)**

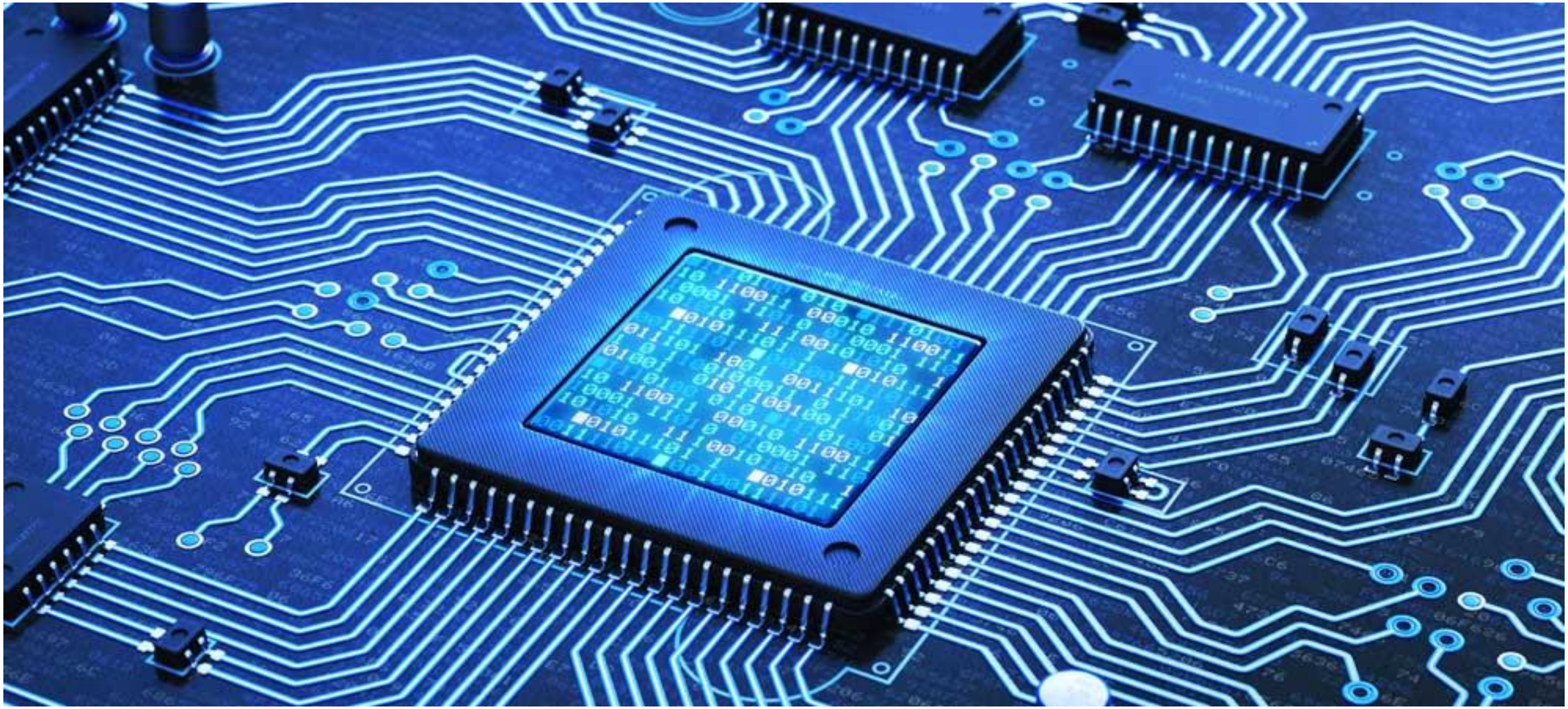
OR-AND



Maxterms of Three Binary Variables

- **All** the **combinations** of **Product-of-Sums** of **three variables** and their **nomenclature** are given below

Minterms					Maxterms	
Term	Designation	x	y	z	Term	Designation
$x'y'z'$	m_0	0	0	0	$x + y + z$	M_0
$x'y'z$	m_1	0	0	1	$x + y + z'$	M_1
$x'yz'$	m_2	0	1	0	$x + y' + z$	M_2
$x'yz$	m_3	0	1	1	$x + y' + z'$	M_3
$xy'z'$	m_4	1	0	0	$x' + y + z$	M_4
$xy'z$	m_5	1	0	1	$x' + y + z'$	M_5
xyz'	m_6	1	1	0	$x' + y' + z$	M_6
xyz	m_7	1	1	1	$x' + y' + z'$	M_7



Examples

Another Example

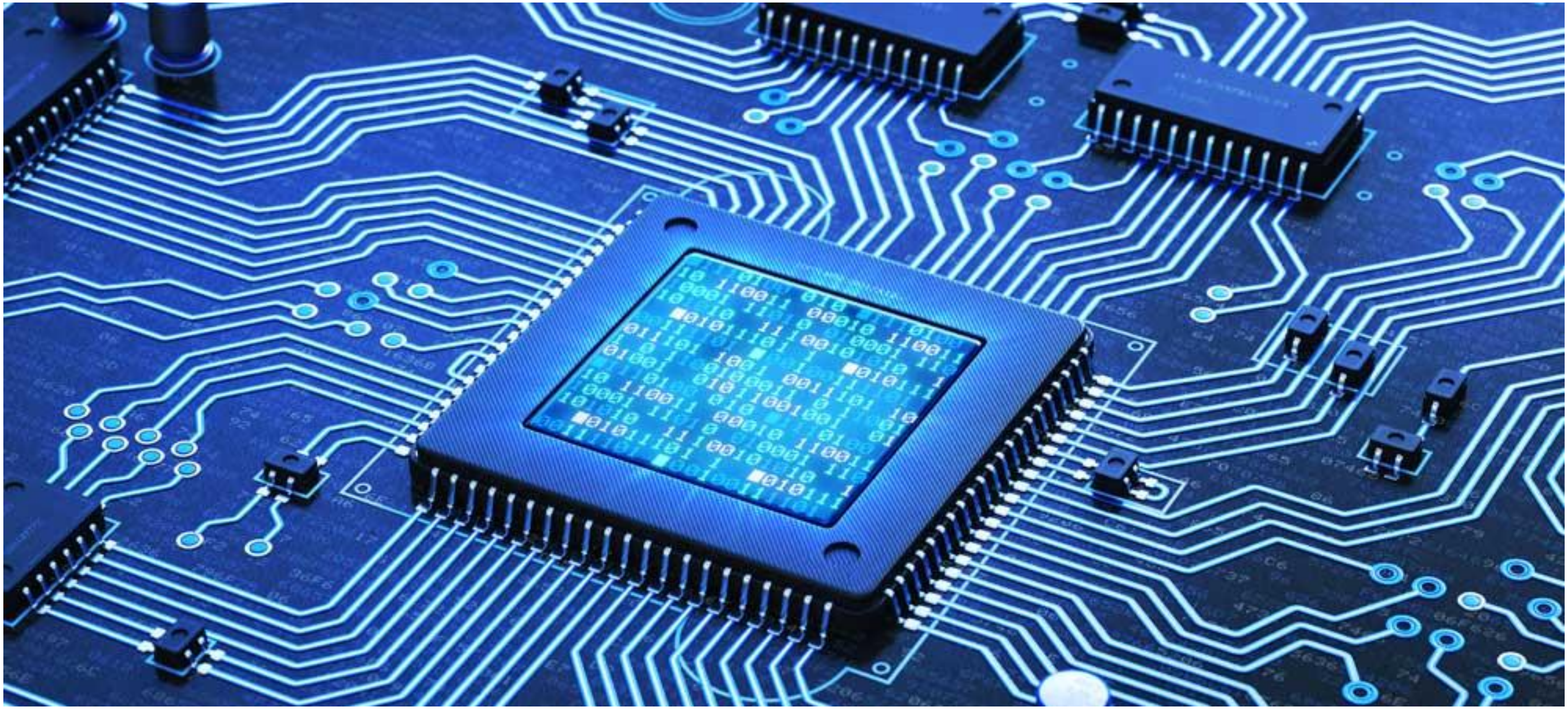
$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$\begin{aligned} f_1 &= (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z) \\ &= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 \end{aligned}$$

$$\begin{aligned} f_2 &= (x + y + z)(x + y + z')(x + y' + z)(x' + y + z) \\ &= M_0 M_1 M_2 M_4 \end{aligned}$$



Home Work

Home Work

- **Simplify** the following Boolean Expressions, using Boolean Algebra:
- 1. $\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$
  - Solution: $AC + BC + \overline{B}\,\overline{C}$ **or** $BC + A\overline{B} + \overline{B}\,\overline{C}$
- 2. $xy + x'z + yz$
  - Solution: $xy + x'z.$
- 3. $(x + y)(x + y')$
  - Solution: **x**

Note: For the problem 1, both the above solutions are correct.
You can verify this by constructing a truth table.
The reason for getting different simplified equations is the order we apply the rules and the terms that we take up first.

Session 2.5: Summary

- Simplification Techniques
- Simplification using Boolean Algebra
  - Quiz 1 to 3
- Two forms of Boolean Expressions
  - Sum-of-Products (minterms)
  - Products-of-Sums (maxterms)
  - Canonical notation
  - Example
- Home Work