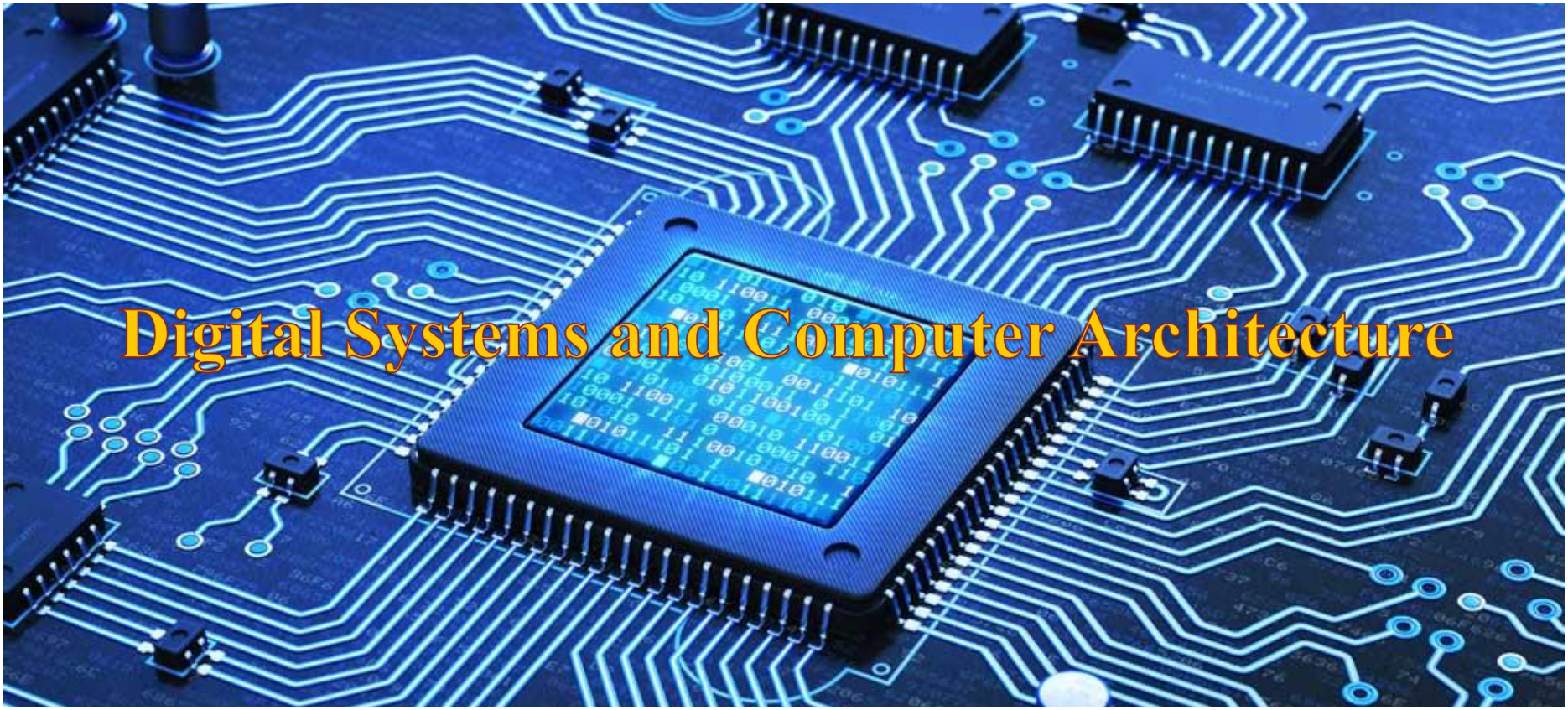


Session 2.2

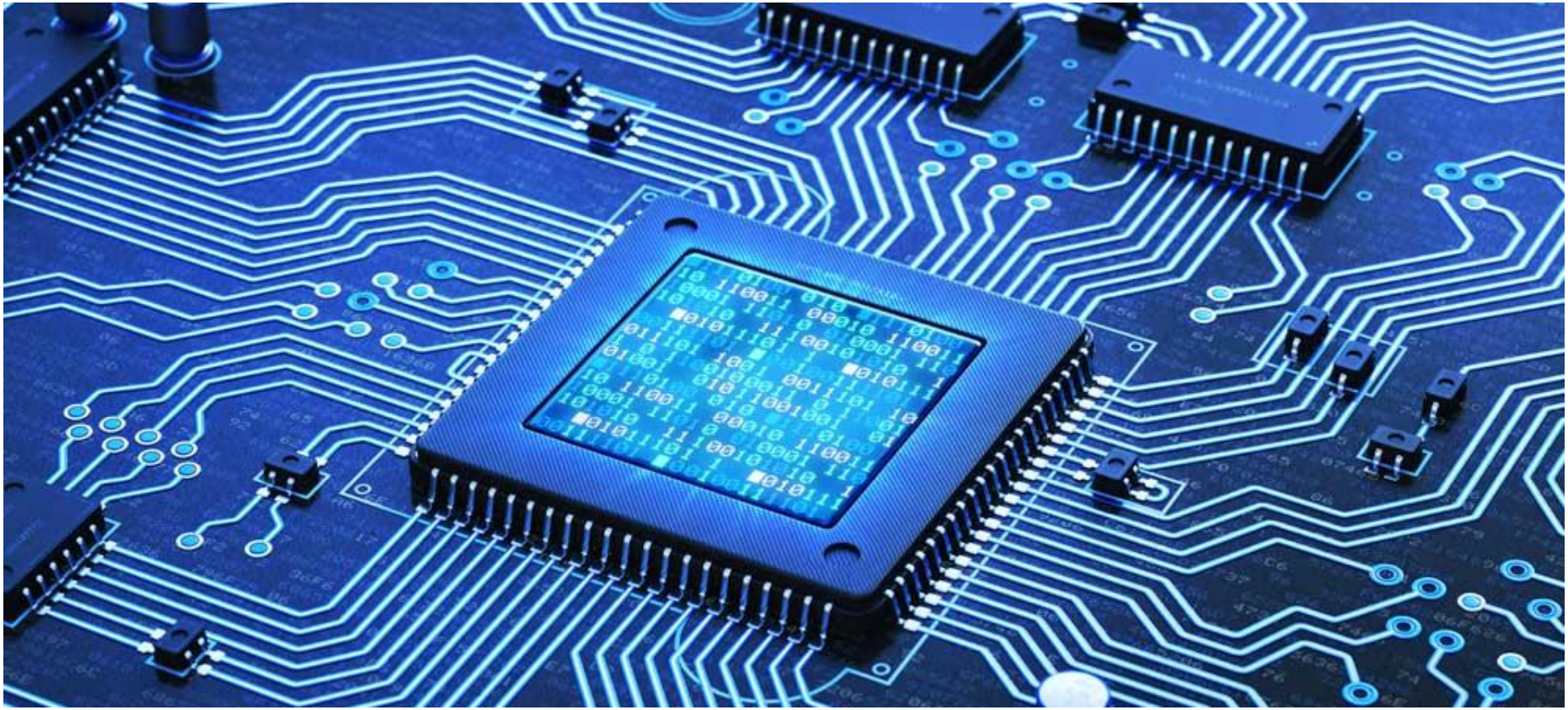
Module 2

Mouli Sankaran

Logic Gates and Binary Adders and Subtractors

Session 2.2: Focus

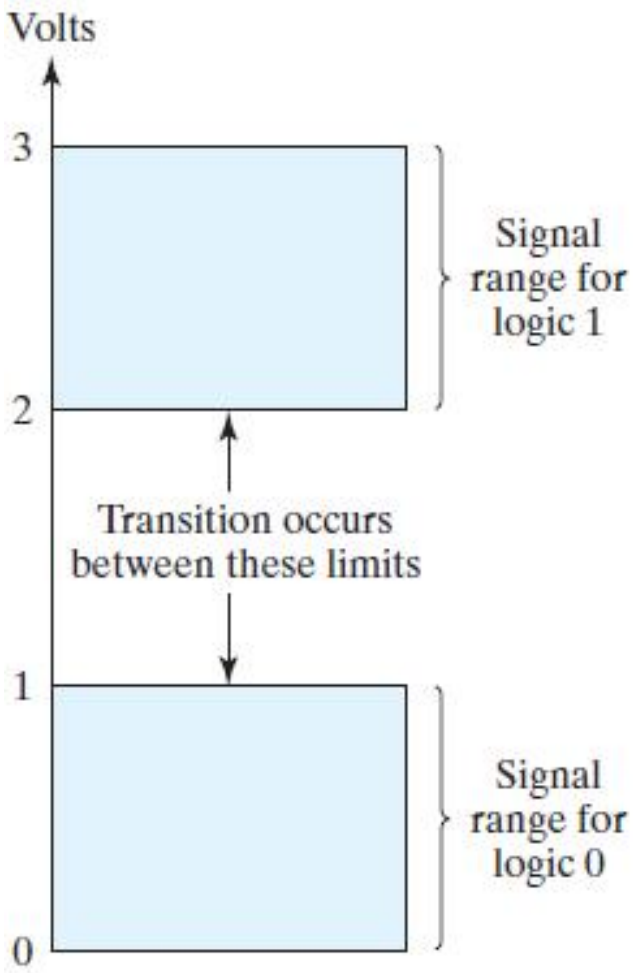
- Logic Gates
- AND, OR, NOT
- NAND and NOR
- XOR and Exclusive-NOR
- Logic Gates - ICs
- Binary Addition
  - Half and Full Adder Circuits
- Binary Subtraction
  - Half and Full Subtractor Circuits
- 7-Segment Display



Logic Gates

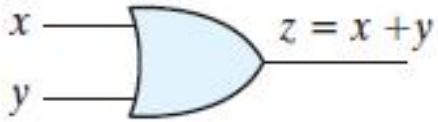
Logic Signals

- The logic signals (**0 and 1**) with which logic gates are driven are shown here
- The voltage levels have consistently come down due to **low power** requirements from **3V** to **less than 1V**

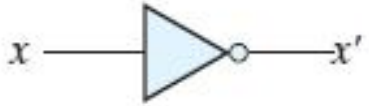


AND, OR, NOT Gates

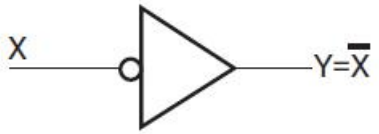
(a) Two-input AND gate



(b) Two-input OR gate



(c) NOT gate or inverter



Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Quiz 1: Draw the Output signals

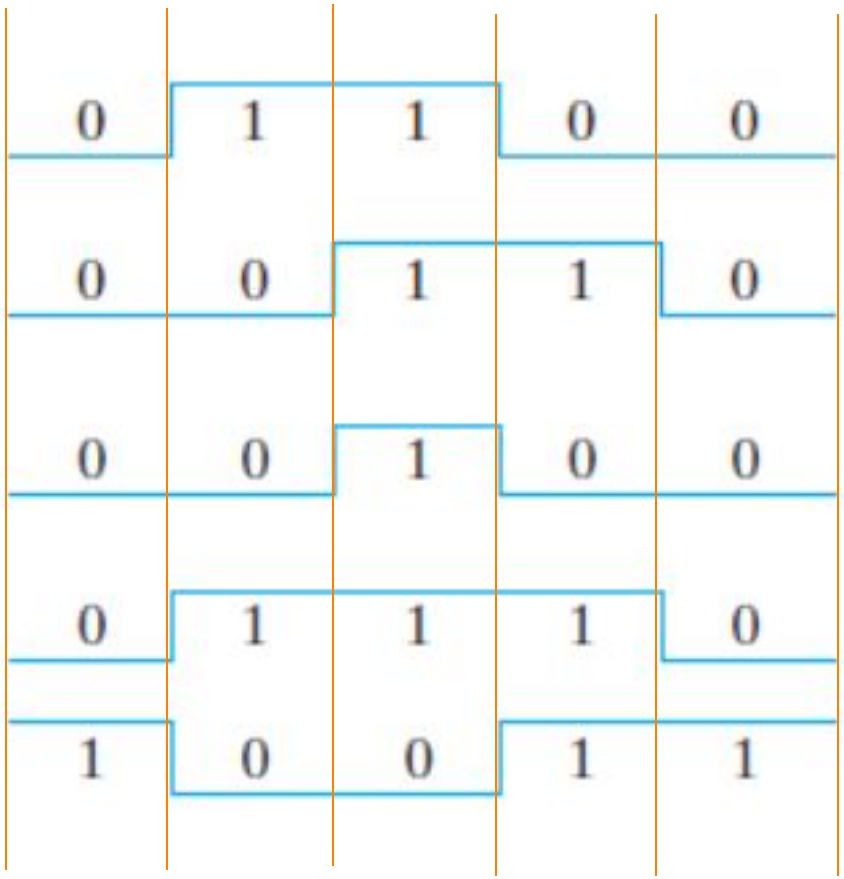
Inputs to the gates

Outputs of the gates

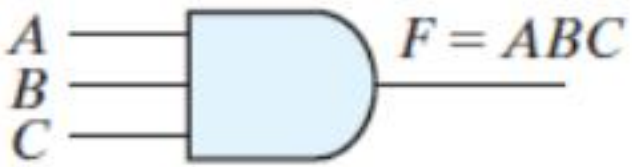
x AND y

x OR y

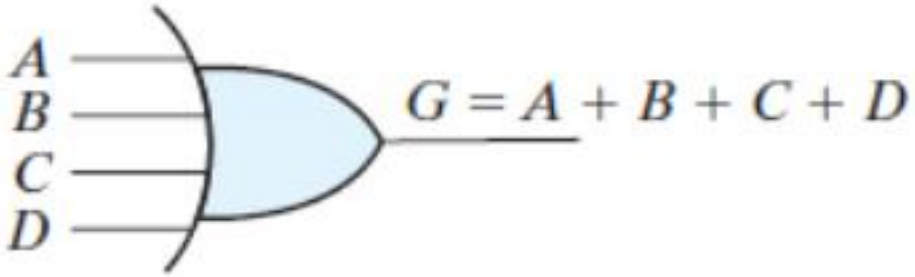
NOT x



Quiz 2: Give Truth Table



(a) Three-input AND gate



(b) Four-input OR gate

Answer for (a):

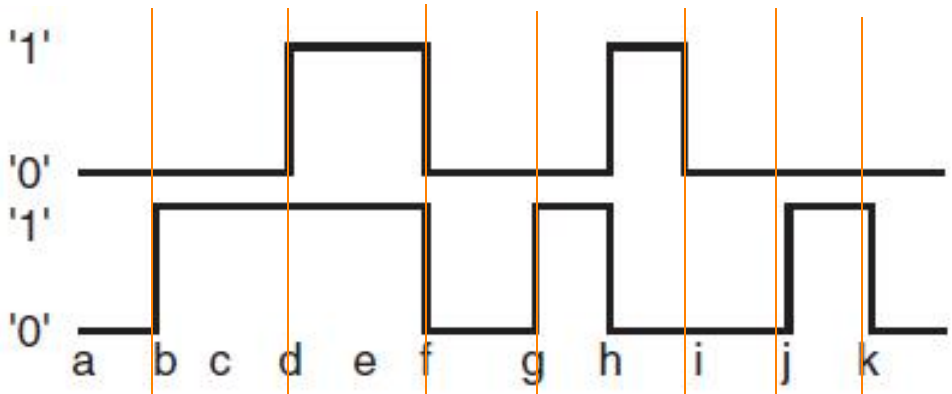
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

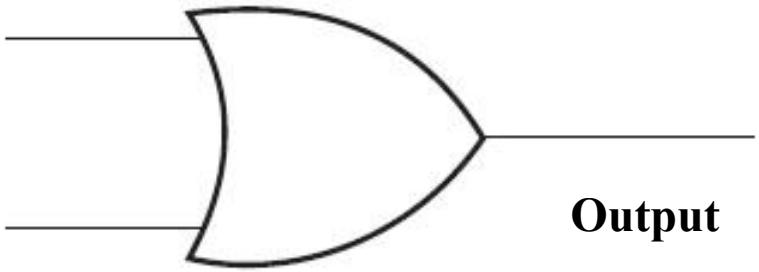
Answer for (b):

If any one input is 1,
the output (**G**) will be one

G is zero only when all
A, B, C, D are zeros

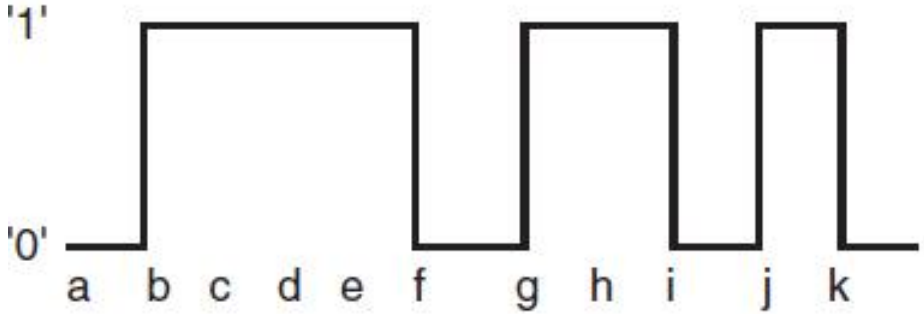
Quiz 3: Draw the Output waveform





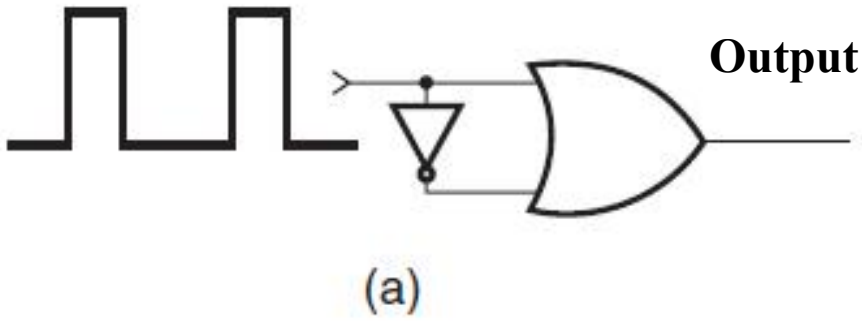
(a)

Answer:



Output

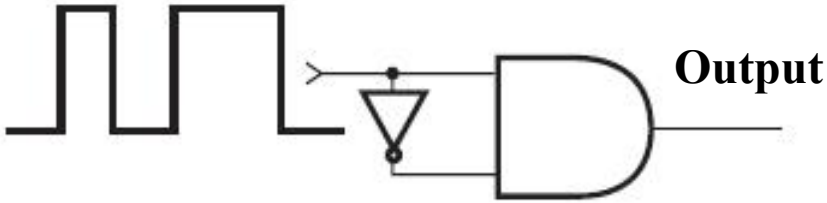
Quiz 4: What are the Outputs?



(a)

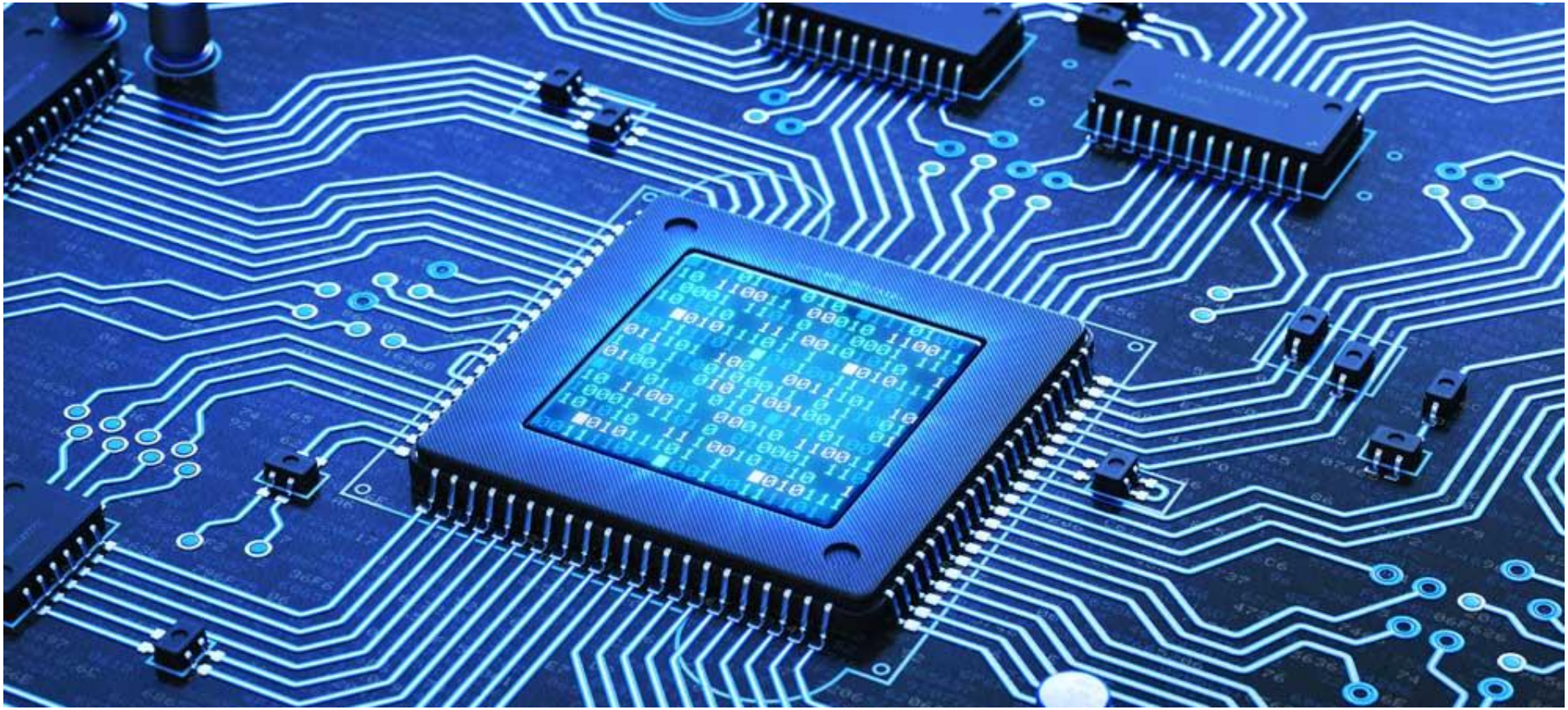
Answers:

Always One



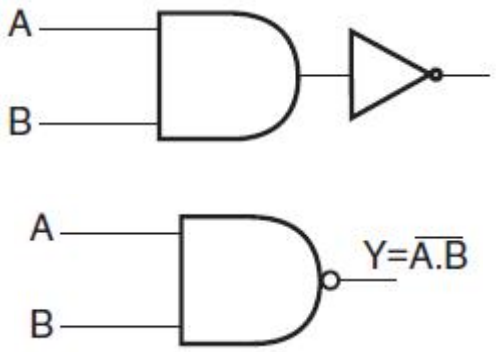
(b)

Always Zero



NAND and NOR Gates

NAND Gates



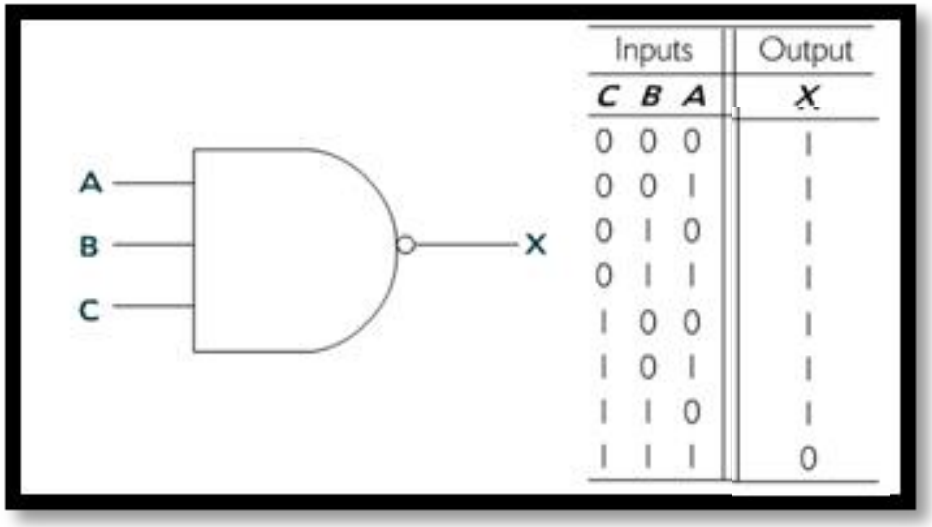
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

$$Y = \overline{A.B}$$

Can also be written as:

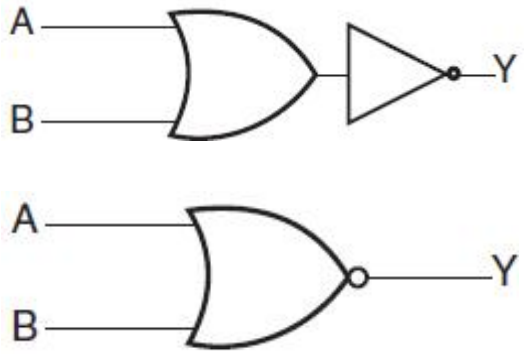
$$\mathbf{Y = \overline{AB}}$$

NAND Gate with **more than two inputs**:



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

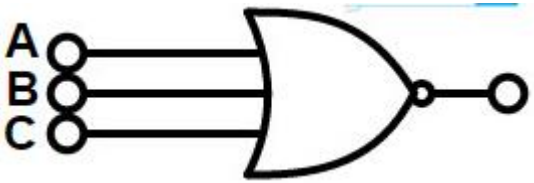
NOR Gates



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

$$Y=\overline{A+B}$$

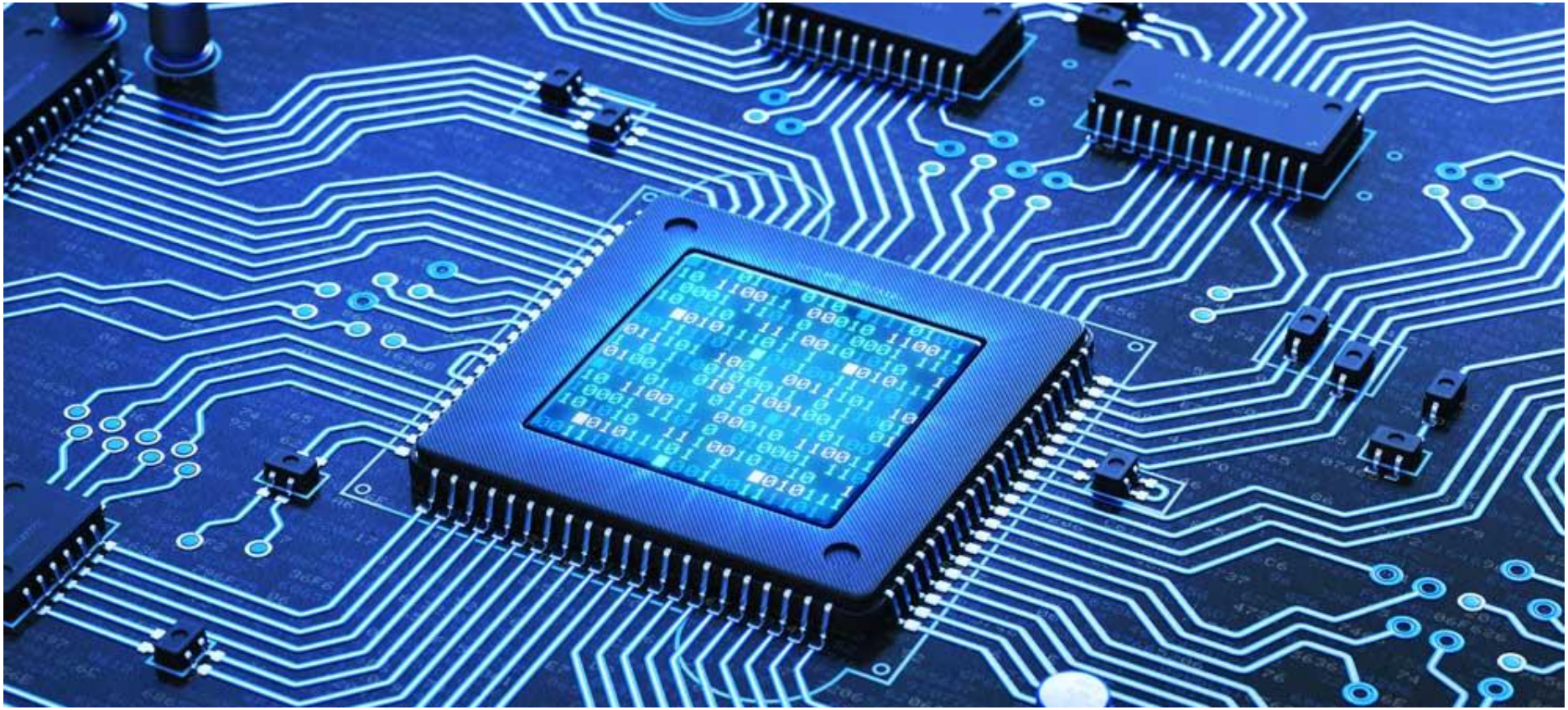
3-input NOR Gate :



TRUTH TABLE

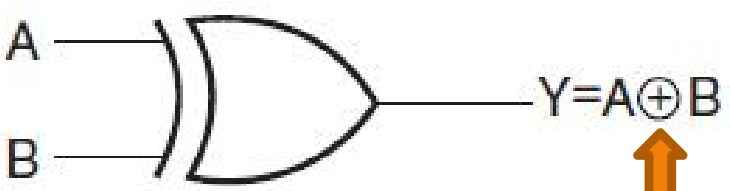
INPUTS			OUTPUT
W	X	Y	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

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



XOR Gate

XOR Gate

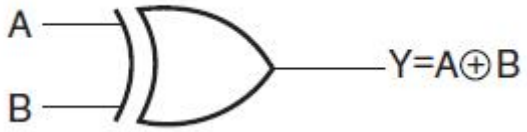


XOR Operator

Truth Table of XOR Gate:

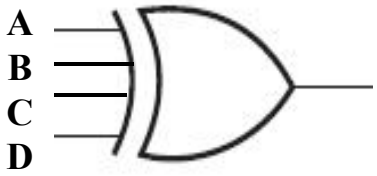
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

XOR Gates

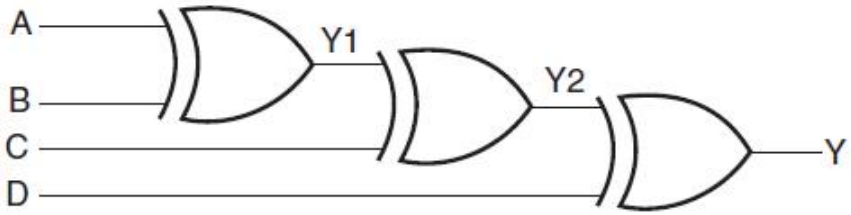


- **XOR** gate is sometimes referred to as "**anything but not all**"
- XOR gate is enabled only when there are **odd number** of digital **1s**
- Therefore, an XOR gate can be viewed as an **odd bits check circuit**

Multiple input XOR Gate

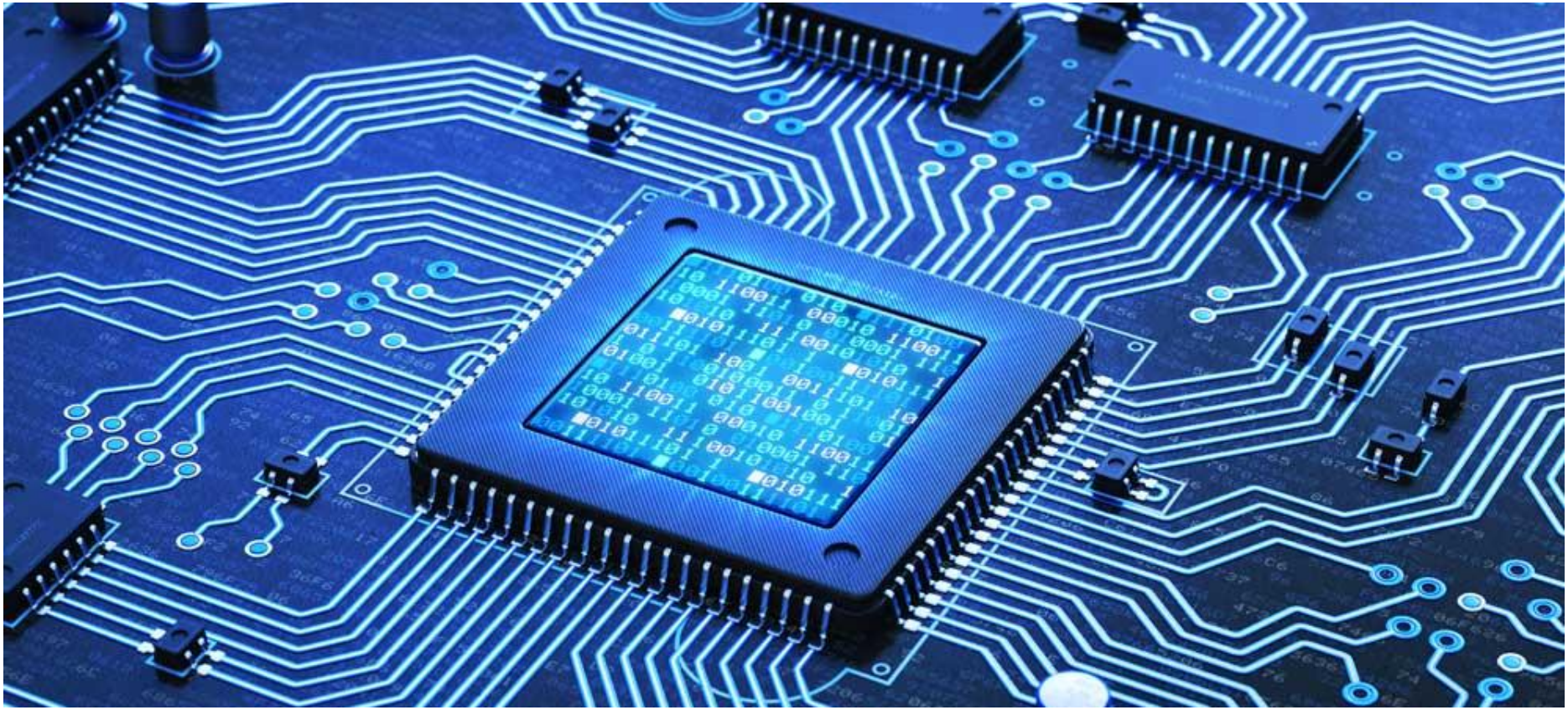


$$Y = A \oplus B \oplus C \oplus D$$



1. **Output** is **1** when **only when** either there are **one** or **three 1s** as **inputs**
2. **Output** is **one when** there are **odd 1s** as **inputs**

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



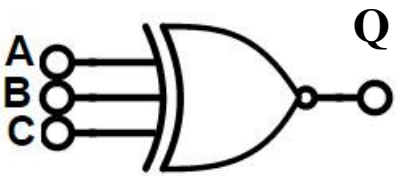
Exclusive-NOR Gates

Exclusive-NOR Gates

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

$$Y = \overline{A \oplus B}$$

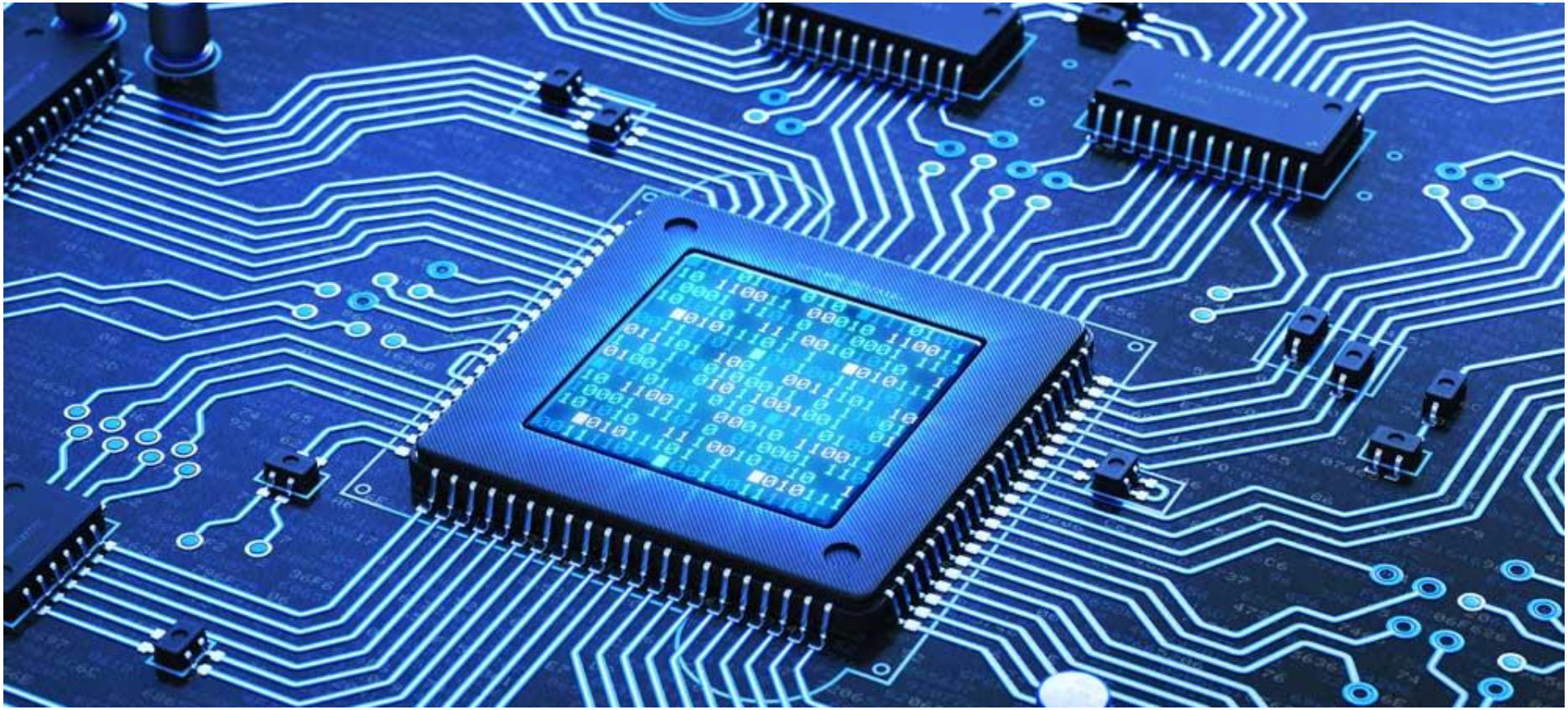
3-input Exclusive-NOR Gate :



Inputs			outputs
W	X	Y	$Q = \overline{A \oplus B \oplus C}$
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

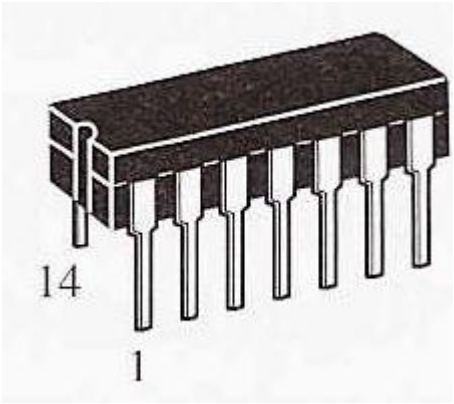
$$Q = \overline{A \oplus B \oplus C}.$$

This is an **Even function**. **Output** is **1** when there are **even** number of **1s** as i**nputs**

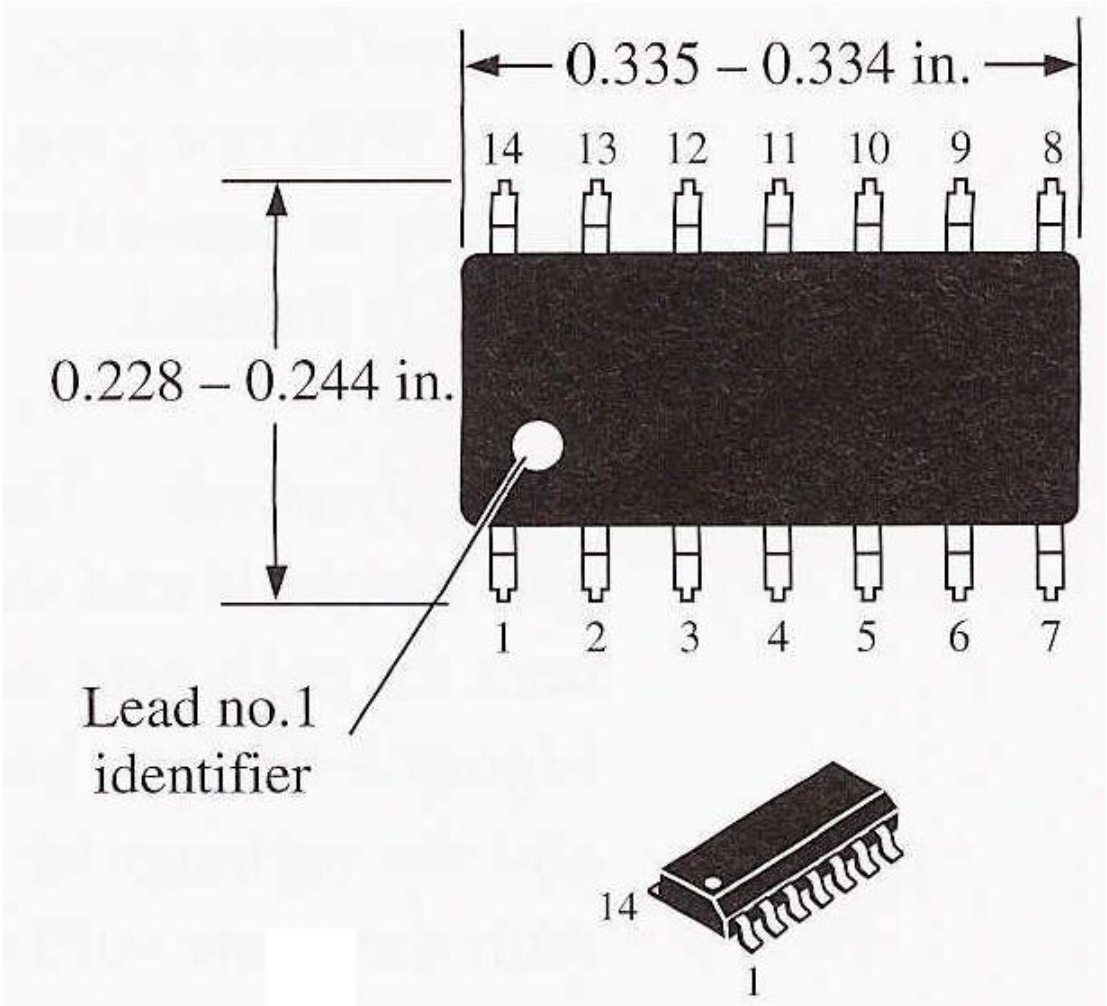


Logic Gates ICs

74LS Series ICs

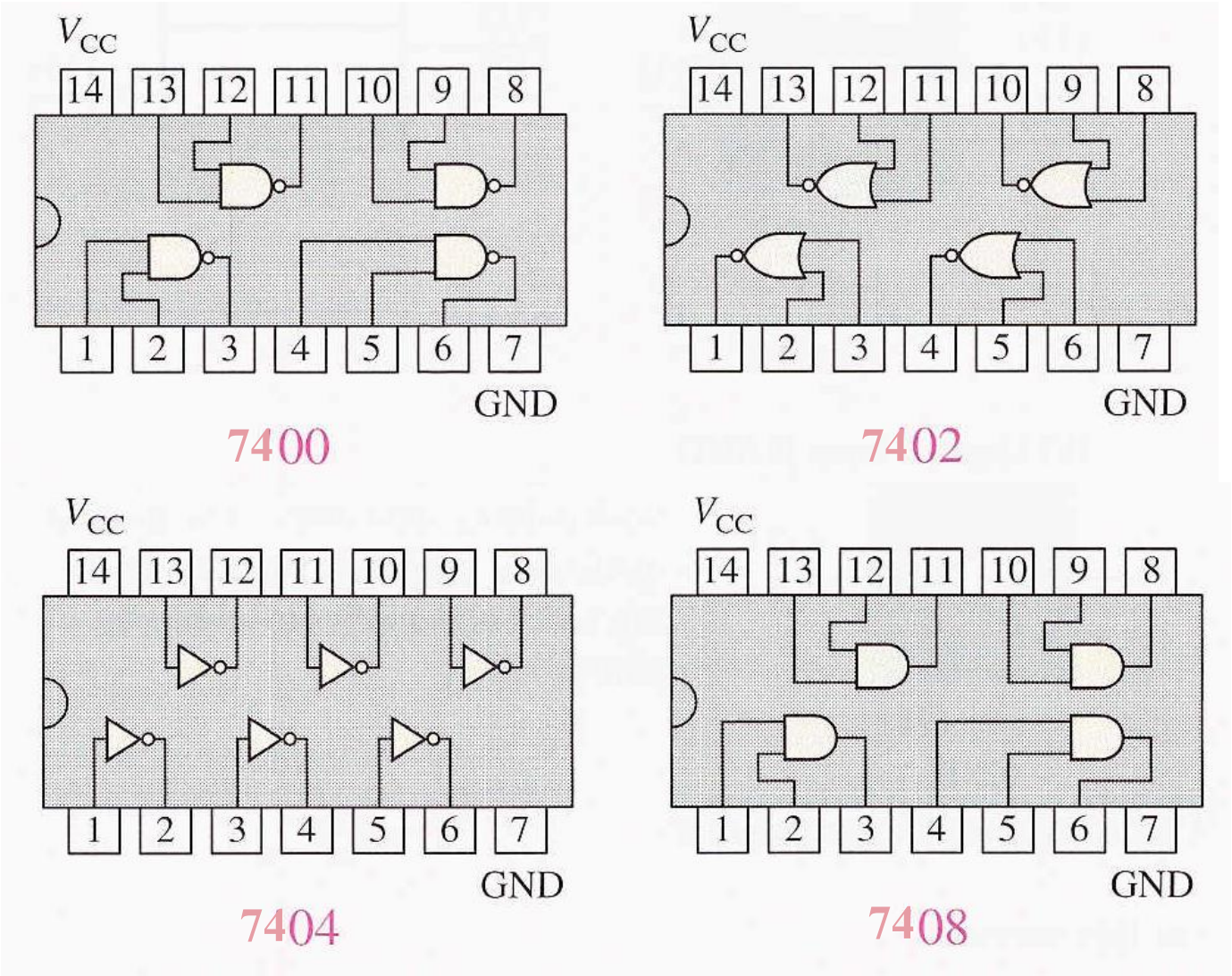


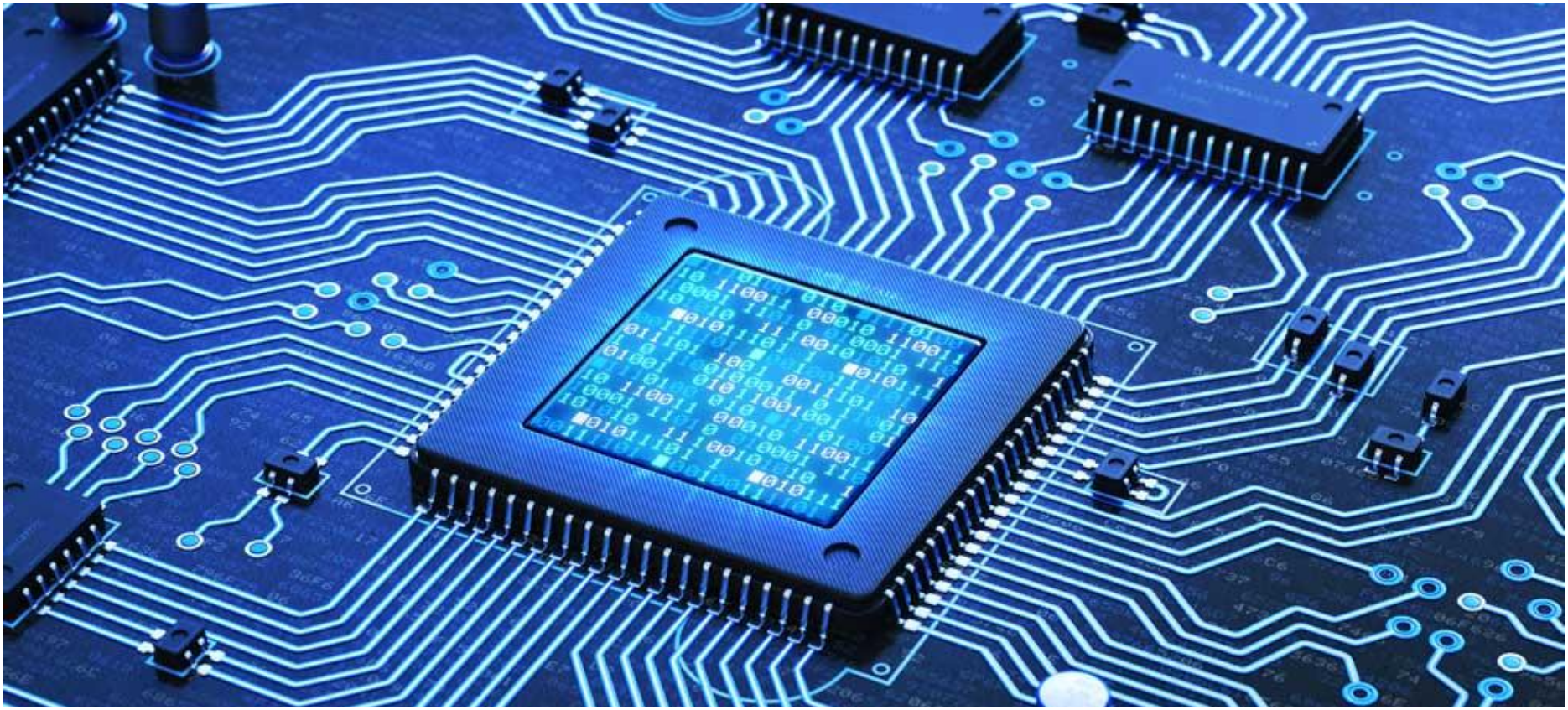
Pin Numbers



LS: Low-power Schottky family

Some 74LS Series ICs





Binary Addition

Basic Rules of Binary Addition (Half Adder)

A + B = Sum (S)

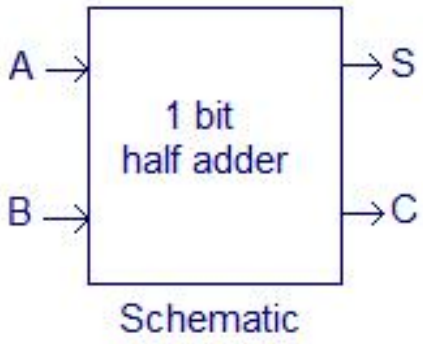
1. $0 + 0 = 0.$
2. $0 + 1 = 1.$
3. $1 + 0 = 1.$

Carry is zero

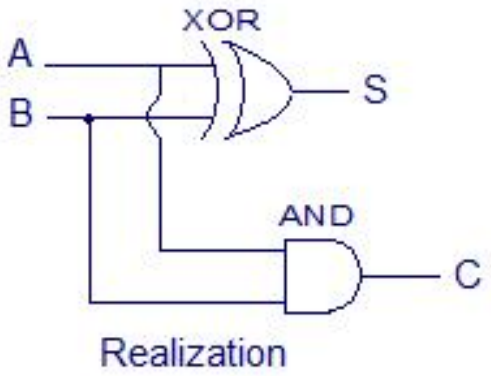
4. $1 + 1 = 0$ with a carry of '1' to the next more significant bit.

Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Truth table



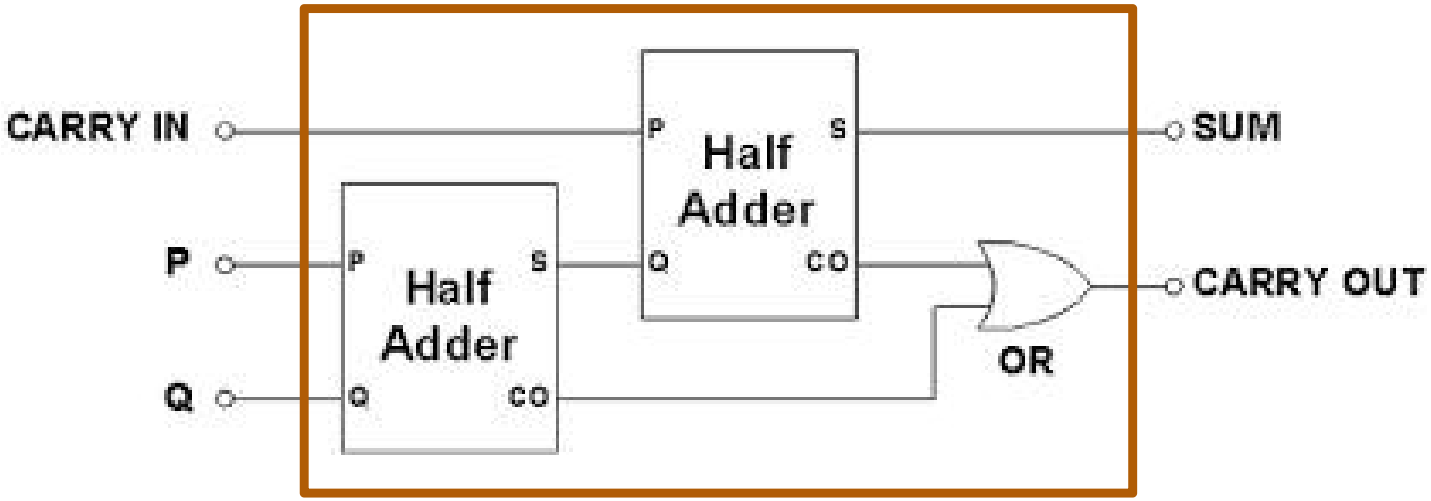
Schematic



Realization

3-bit Binary Addition (Full Adder)

A	B	Carry-in (C_{in})	Sum	Carry-out (C_o)	A	B	Carry-in (C_{in})	Sum	Carry-out (C_o)
0	0	0	0	0	1	0	0	1	0
0	0	1	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1
0	1	1	0	1	1	1	1	1	1



Larger-bit Binary Addition

- Consider **32 bit adders** are used to add **two 128 bit** numbers $(A_3\ A_2\ A_1\ A_0)$ and $(B_3\ B_2\ B_1\ B_0)$ **Note**: Consider each of them are 32-bit numbers

1.

			(C_0)	
	A_3	A_2	A_1	A_0
	B_3	B_2	B_1	B_0
				S_0

2.

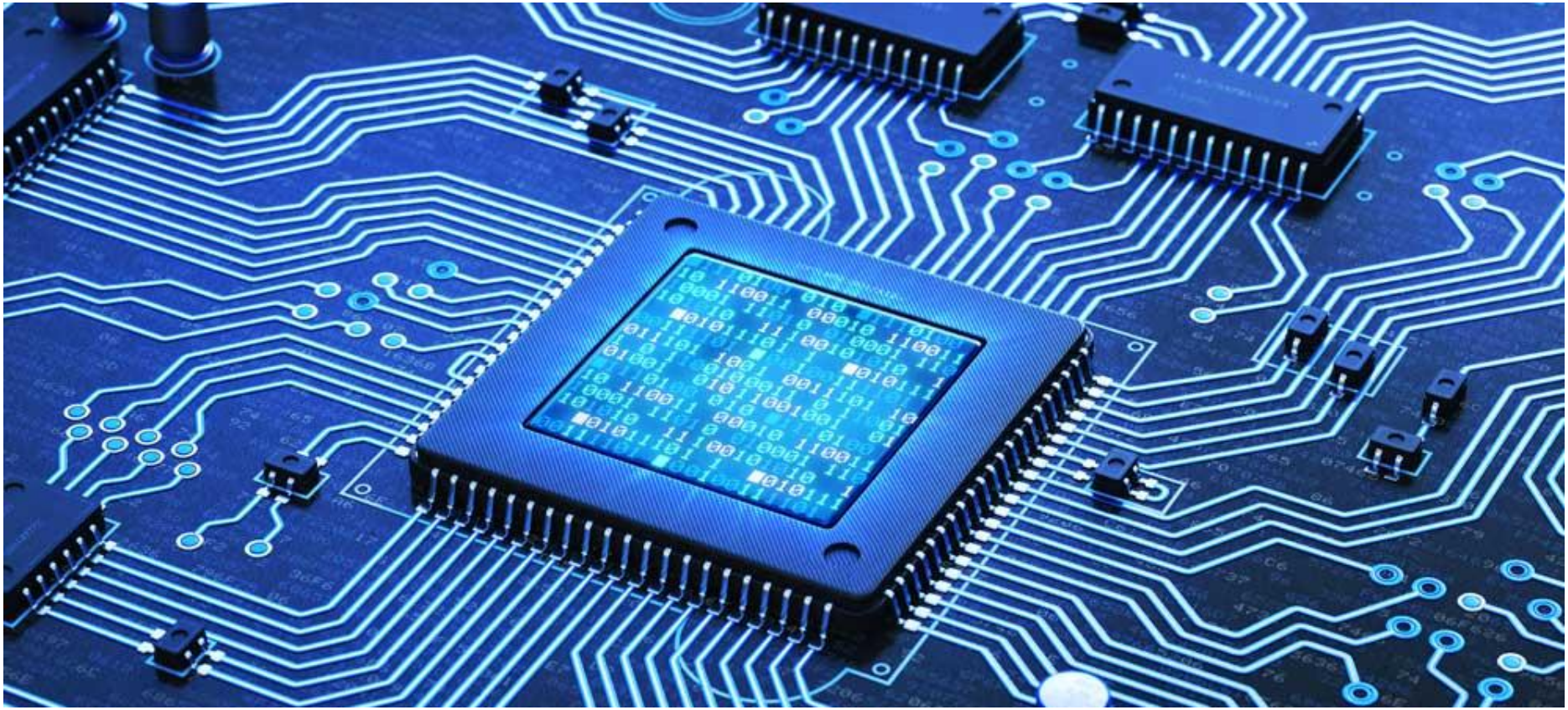
		(C_1)	(C_0)	
	A_3	A_2	A_1	A_0
	B_3	B_2	B_1	B_0
			S_1	S_0

3.

(C_2)	(C_1)	(C_0)	
A_3	A_2	A_1	A_0
B_3	B_2	B_1	B_0
	S_2	S_1	S_0

4.

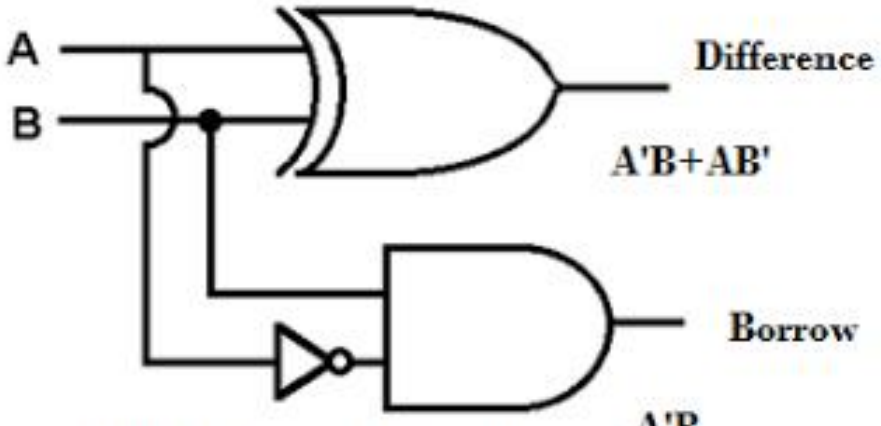
	(C_2)	(C_1)	(C_0)	
	A_3	A_2	A_1	A_0
	B_3	B_2	B_1	B_0
C_3	S_3	S_2	S_1	S_0



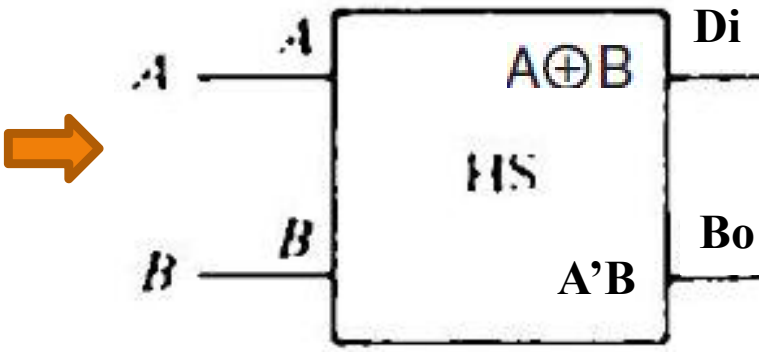
Binary Subtraction

Basic Rules of Binary Subtraction (Half Subtractor)

	A Minuend		B Subtrahend		(A-B) Difference	Borrow out
Rule 1	0	–	0	=	0	
Rule 2	0	–	1	=	1	and borrow 1
Rule 3	1	–	0	=	1	
Rule 4	1	–	1	=	0	



Half Subtractor Logic Diagram

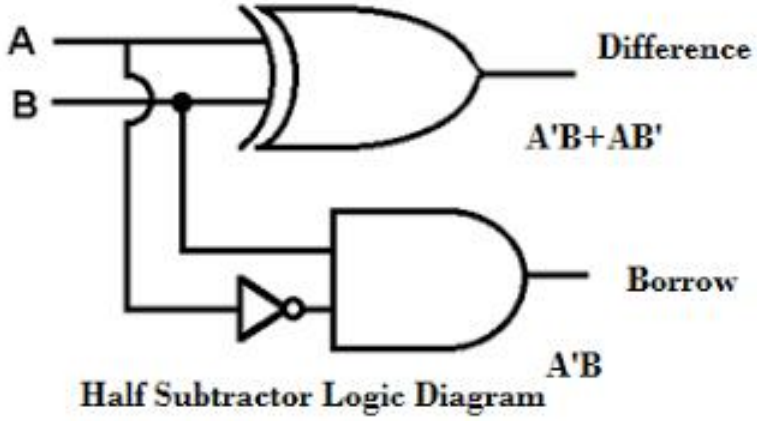


Subtraction: Borrow Bit

Inputs		Outputs	
Minuend A	Subtrahend B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0
A − B		Di	Bo

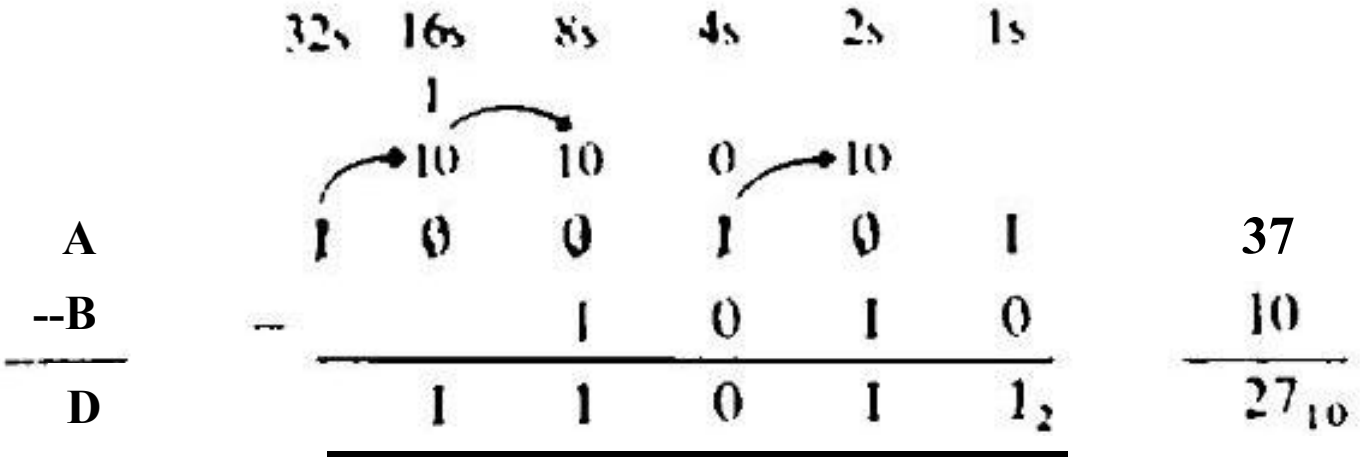
When 1 is subtracted from 0
A borrow is taken from the next most significant bit, by making it zero

Binary	Decimal
borrow	
Minuend 10	2
Subtrahend − 0 1	− 1
Difference 0 1	1



Half Subtractor Logic Diagram

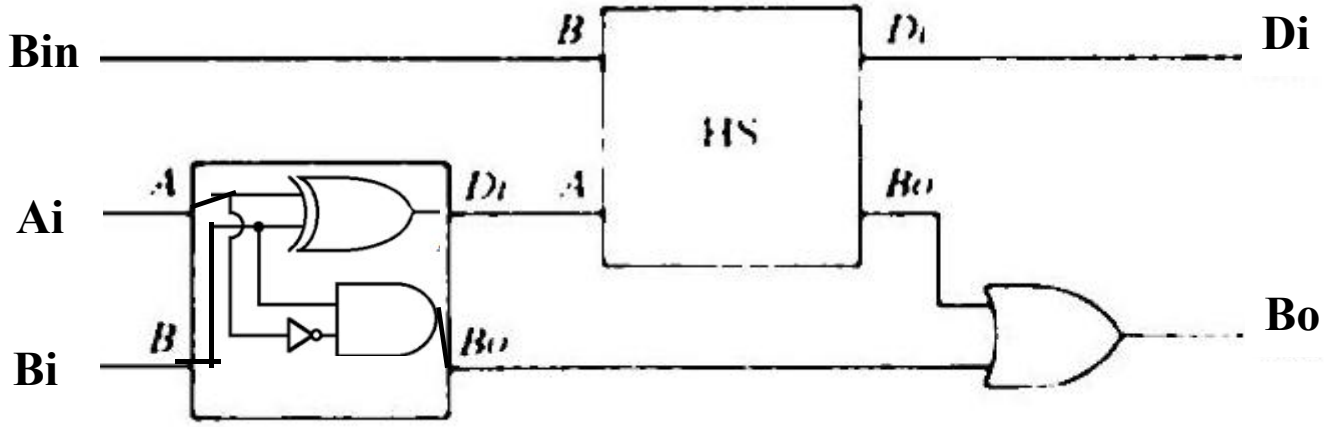
Subtraction: Example



Overall, the result will the Borrow bit will be cleared (zero).

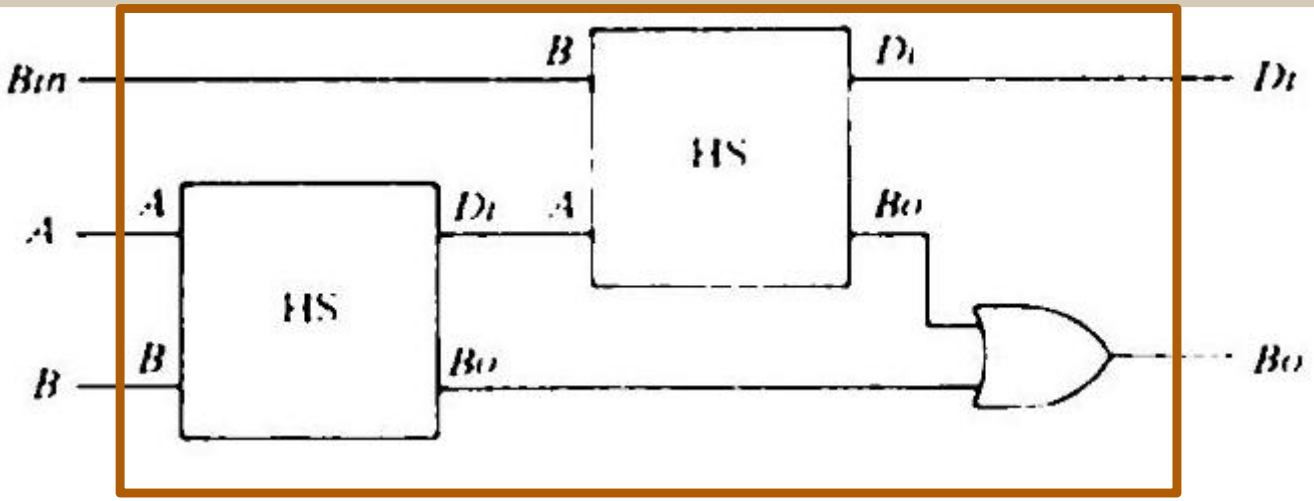
Subtraction: Digital Implementation

A		1	0	0	1	0	1	37
--B	−			1	0	1	0	10
D			1	1	0	1	1_2	27_{10}



HS

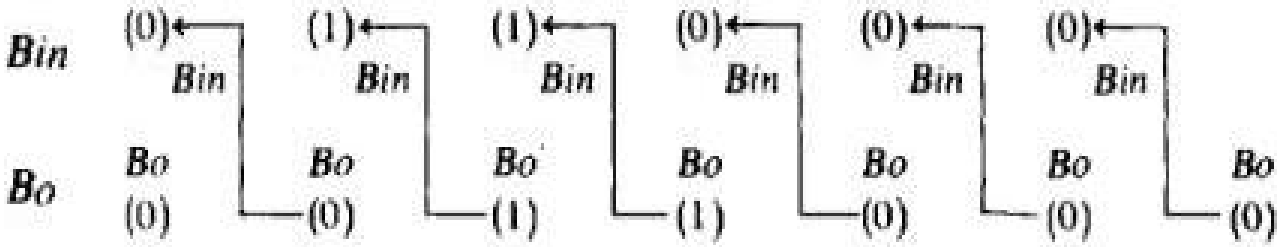
3-bit Binary Subtraction (Full Subtractor)



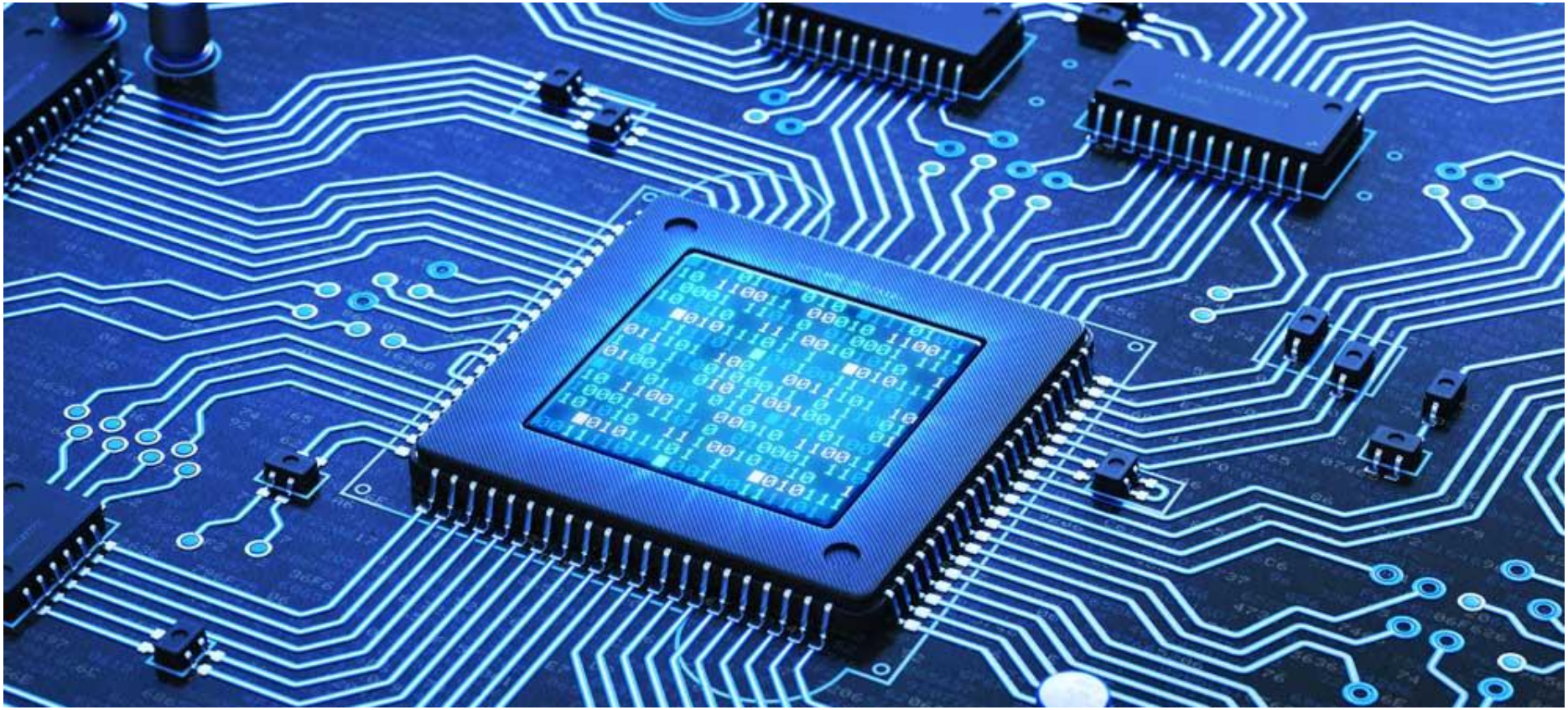
Inputs			Outputs	
Minuend (A)	Subtrahend (B)	Borrow-in (B_{in})	Difference (D)	Borrow-out (B_o)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Subtraction: Example

	64s	32s	16s	8s	4s	2s	1s	
A	1	1	1	0	1	0	1	117
− B	− 0	0	1	1	1	0	0	− 28
D1	1	0	1	1	0	0	1	89



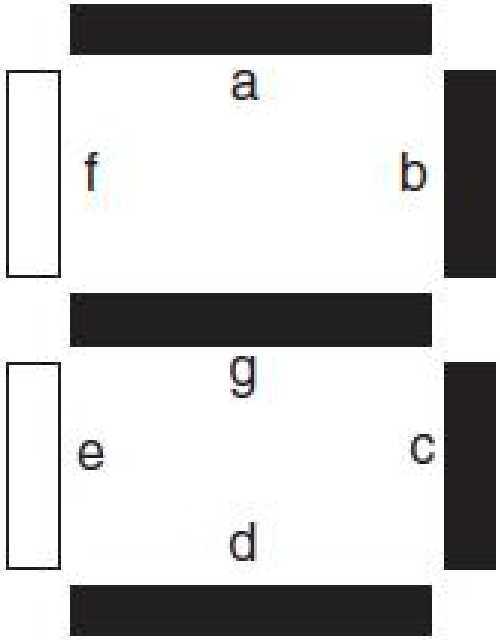
- **Borrow (in)** and **Borrow (out)** are shown above



7-Segment Display Code

7-Segment Display Code

Number displayed is 3 here



- **1 means the segment is ON**

	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	0	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	0	0	1	1
a	1	1	1	1	1	0	1
b	0	0	1	1	1	1	1
c	0	0	0	1	1	0	1
d	0	1	1	1	1	0	1
e	1	1	0	1	1	1	1
f	1	0	0	0	1	1	1

Session 2.2: Summary

- 7-Segment Display
- Logic Gates
- AND, OR, NOT
- NAND and NOR
- XOR and Exclusive-NOR
- Logic Gates - ICs
- Binary Addition
  - Half and Full Adder Circuits
- Binary Subtraction
  - Half and Full Subtractor Circuits