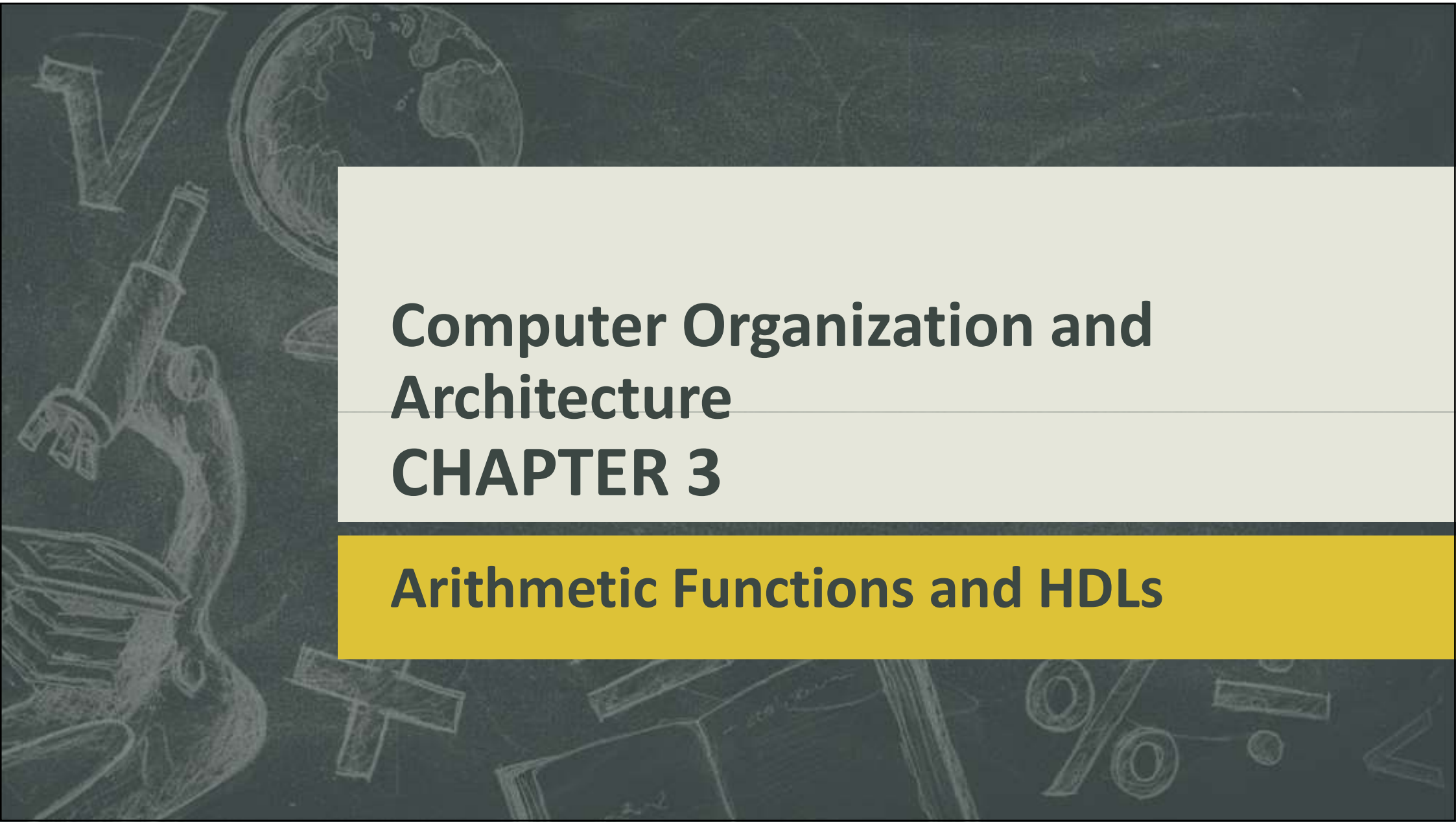


Computer Organization and Architecture

CHAPTER 3

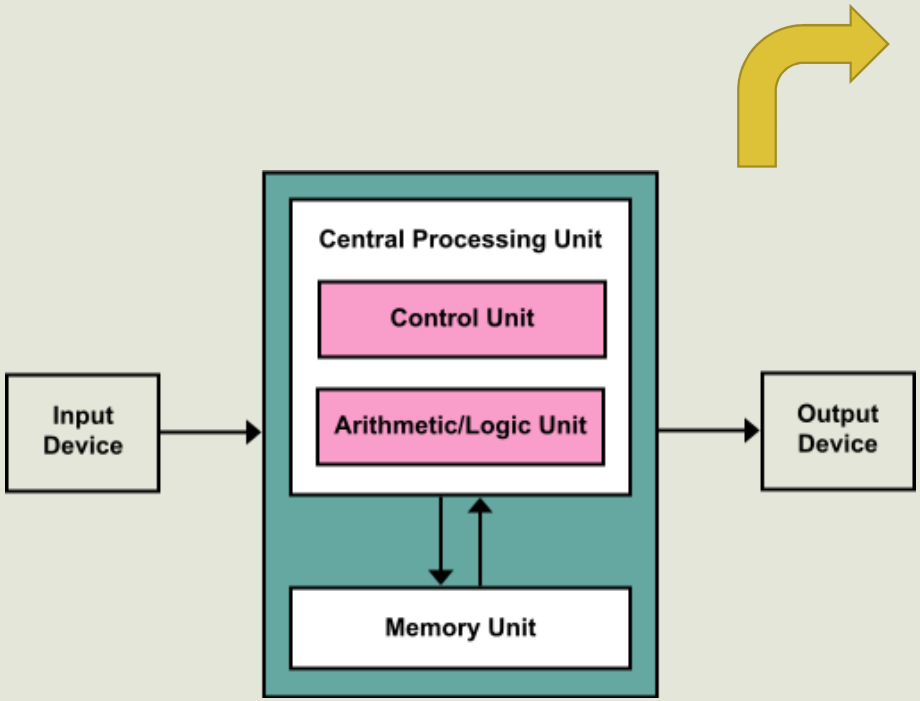
Arithmetic Functions and HDLs

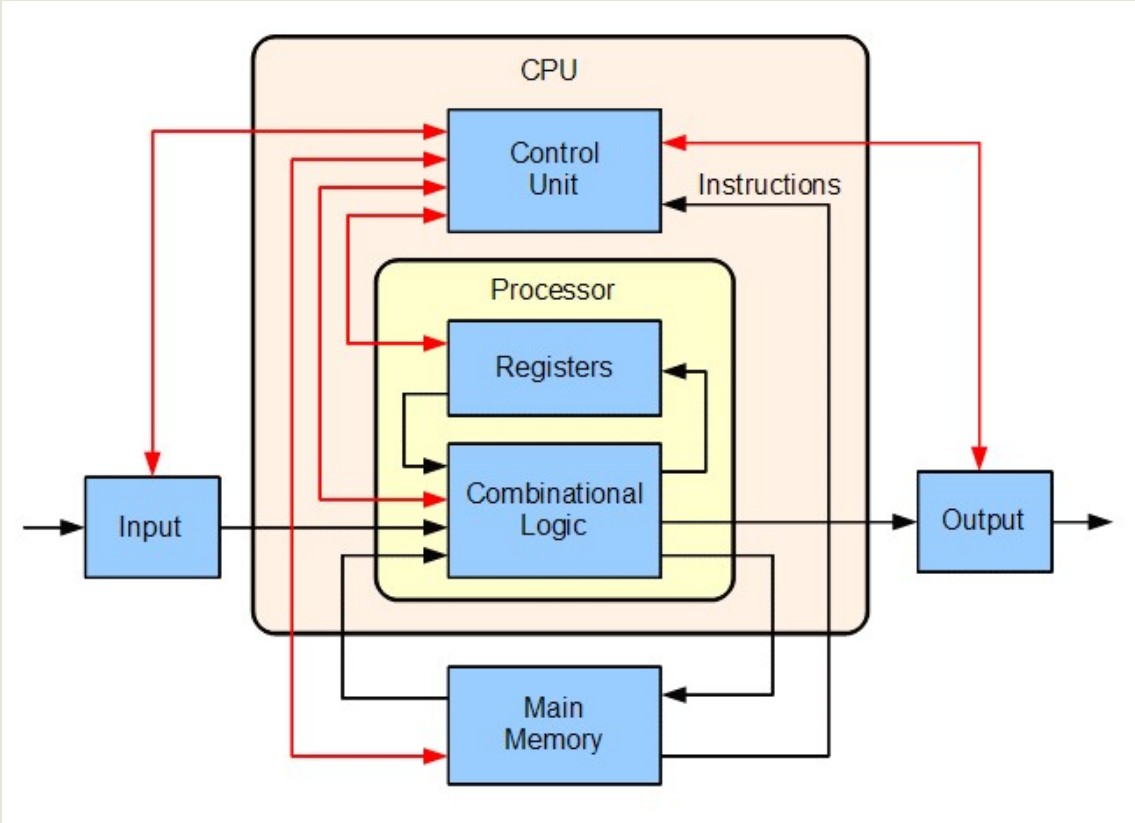
Agenda

- Block Diagram of ALU
- Binary Half and Full Adder
- Decimal Adder
- Binary Parallel Adder
- BCD Adder
- Half and Full Subtractor.

Where is ALU?

CPU Architecture Diagram



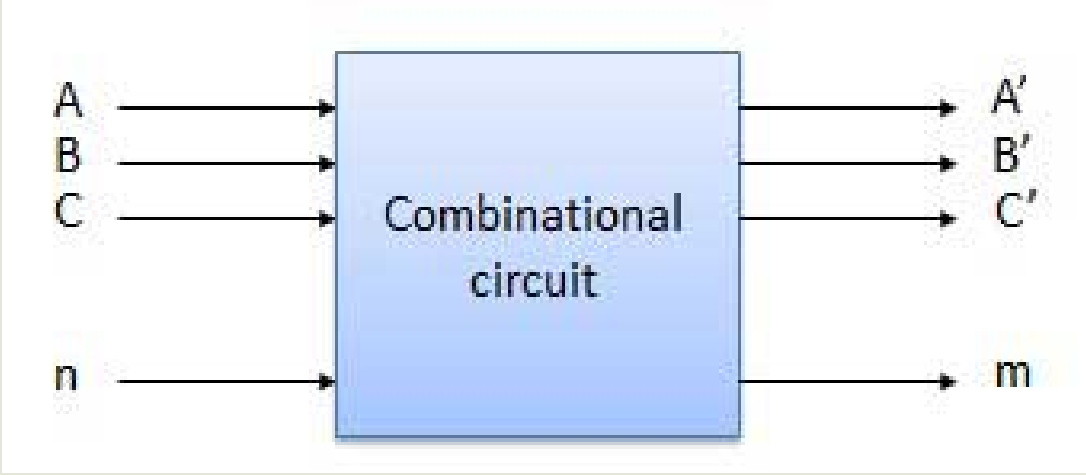


1. ARITHMETIC LOGIC UNIT

- An **arithmetic logic unit (ALU)** is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the **central processing unit (CPU)** of a computer.
- Modern CPUs contain very powerful and complex ALUs. In addition to ALUs, modern CPUs contain a control unit (CU).
- Most of the operations of a CPU are performed by one or more ALUs, which load data from input registers. A **register** is a small amount of storage available as part of a CPU.
- The control unit tells the ALU what operation to perform on that data, and the ALU stores the result in an output register. The control unit moves the data between these registers, the ALU, and memory.

1. ARITHMETIC LOGIC UNIT

- An ALU performs basic arithmetic and logic operations. Examples of arithmetic operations are addition, subtraction, multiplication, and division. Examples of logic operations are comparisons of values such as NOT, AND, and OR.

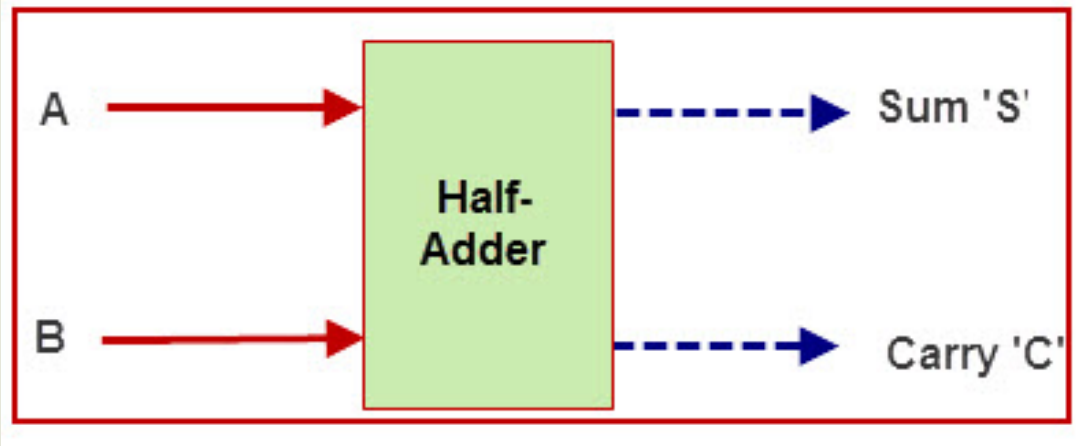


2. BINARY ADDERS

- Combinational circuit that performs the addition of two bits is called a *half adder*.
- One that performs the addition of three bits is called a *full adder*.

❑ Half Adder

- By using half adder, you can design simple addition with the help of logic gates.
- Let's see an addition of single bits.



❑ Half Adder

0+0 = 0

0+1 = 1

1+0 = 1

1+1 = 10

These are the least possible single-bit combinations. But the result for 1+1 is 10, the sum result must be re-written as a 2-bit output. Thus, the equations can be written as

0+0 = 00

0+1 = 01

1+0 = 01

1+1 = 10

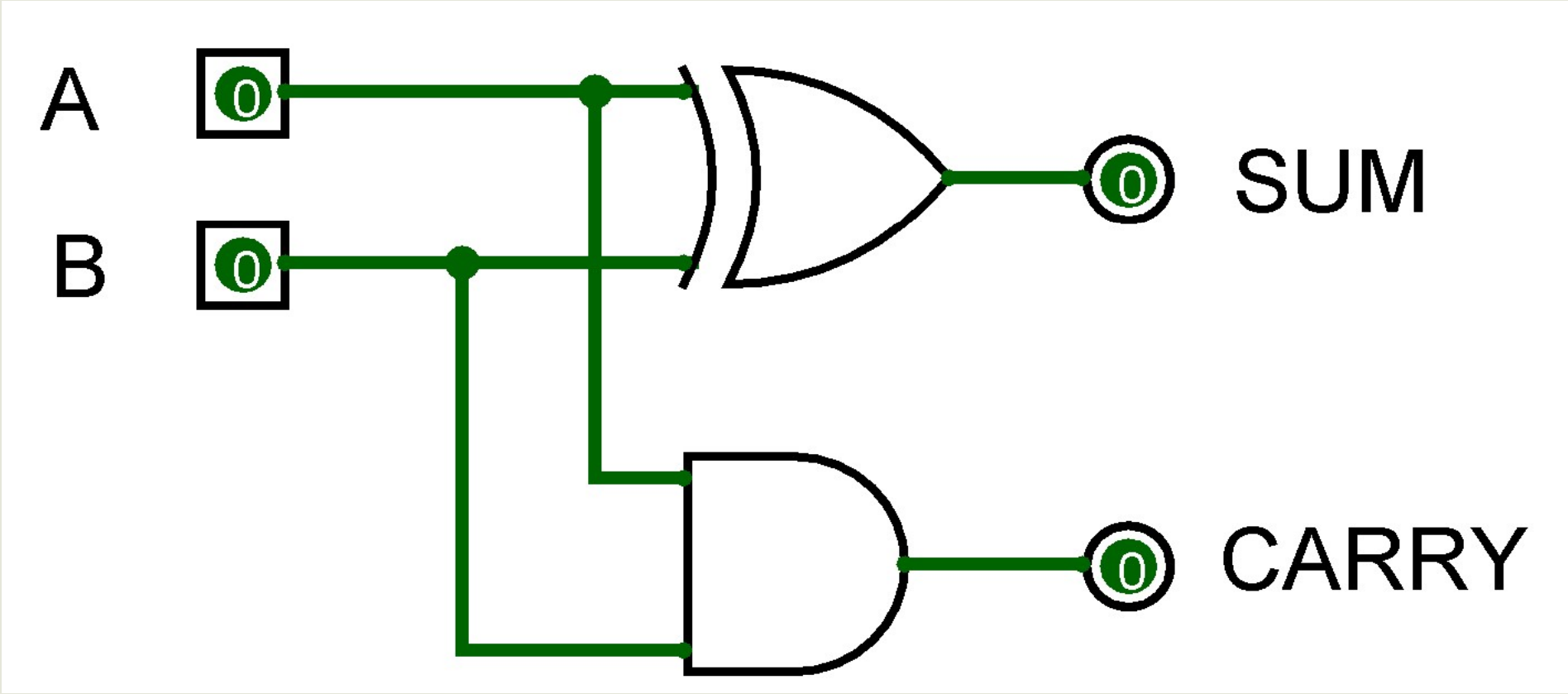
❑ Half Adder

- Half Adder Truth Table

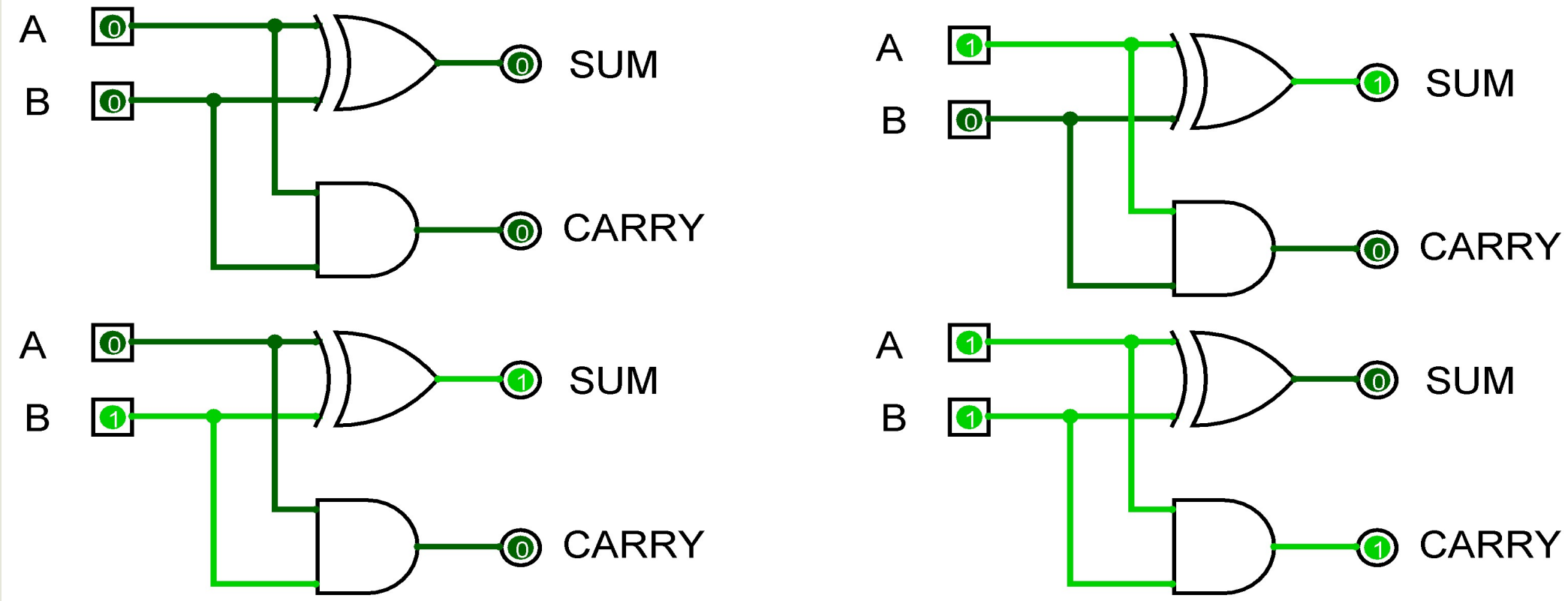
TURTH TABLE			
INPUT		OUTPUT	
A	B	$S = A \oplus B$	$C = AB$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Now it has been cleared that 1-bit adder can be easily implemented with the help of the XOR Gate for the output 'SUM' and an AND Gate for the 'Carry'.

❑ Half Adder

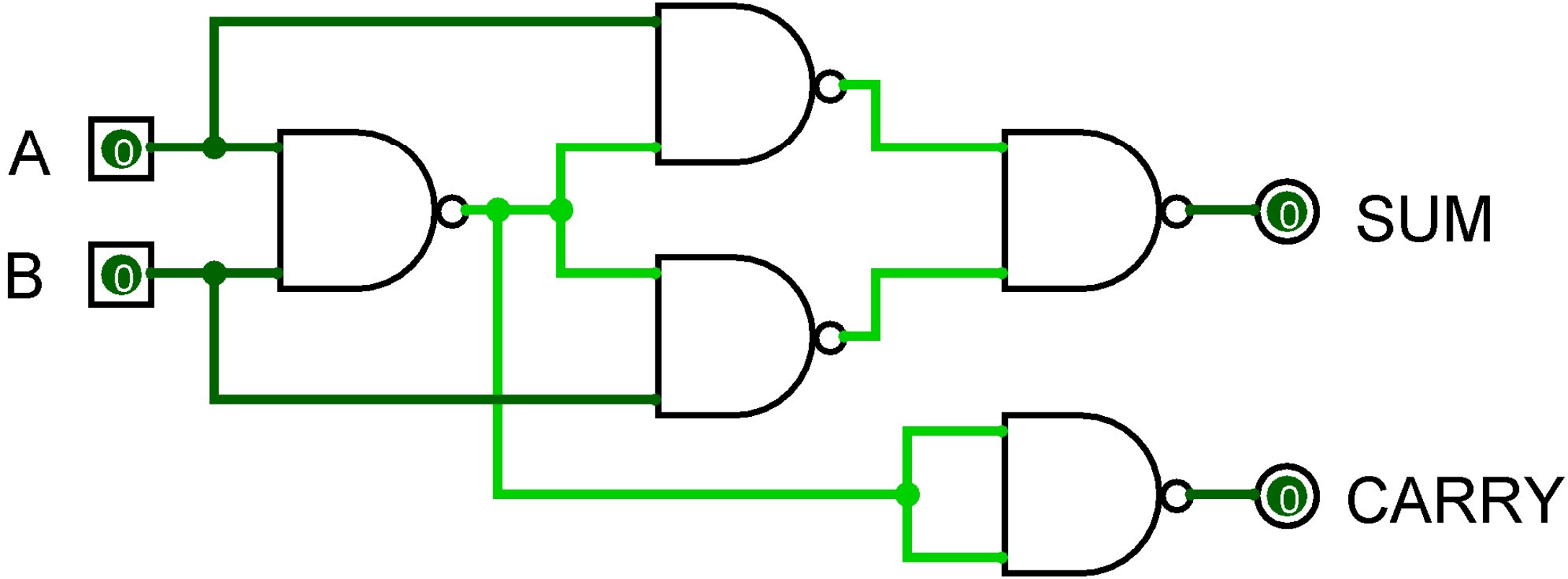


❑ Half Adder


A
B
0
0
SUM
0
CARRY
0
A
B
1
0
SUM
1
CARRY
0
A
B
0
1
SUM
1
CARRY
0
A
B
1
1
SUM
0
CARRY
1

❑ Half Adder

- **Half Adder Using NAND Gate**



❑ Half Adder

- **Half Adder Using NOR Gate**