SIT-111

COMPUTER SYSTEMS jasveena-224001588

DIGITAL LOGIC

it is basically the basis of digital systems, like computers and other electronic devices, there are tiny components called logic gates which make up the integrated circuits which are the heart of these devices.

WHAT ARE BINARY VALUES?

Binary values are nothing but 0 and 1. It is the language of digital electronics.

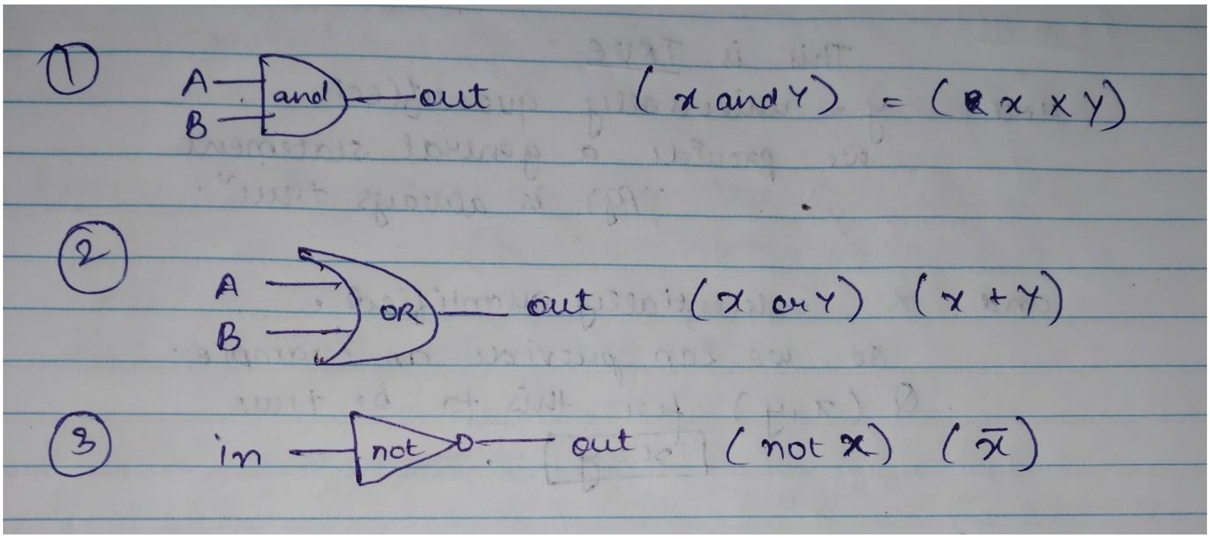
THE ROLE OF BOOLEAN ALGEBRA

Boolean algebra is a branch of mathematics dealing with binary values 0 and 1.

A logic gate is an electronic component that can be used to conduct Boolean operations.

And its 3 basic types are:

1. **AND Gate**: Outputs 1 only if both inputs are 1.
2. **OR Gate**: Outputs 1 if at least one of the inputs is 1.
3. **NOT Gate**: Outputs the opposite of the input - if the input is 0, it outputs 1, and vice versa.



there are some more advanced Boolean operations that are performed on binary values,

- **NOR (⊽ or ↓)**: The NOR gate is a logical operation that is the negation or not of the OR operation. It outputs the opposite of what the result of OR will be.
- **NAND (⊼ or ↑)**: The NAND gate is the inverse of the AND operation. It outputs true if either of the inputs is false, which is again what the opposite of AND gate is. NAND gate is also called the universal gate.
- **XOR (⊻ or ≠)**: The XOR gate, or Exclusive OR, provides a true output only when the number of true inputs is odd. When we add two inputs, it outputs true if either A or B is true, but not both. This gate is used for more practical problem solving.

TRUTH TABLE

A truth table is a table we fill when we are given a logic operation or a figure of it . generally, a truth table can be filled by using the pre given inputs and sometimes we have to take all possible combinations to determine which gate will output what value accordingly.

BOOLEAN IDENTITIES

1. Commutative law

   AND: (A ∧ B = B ∧ A)

   OR: (A ∨ B = B ∨ A)

2. Associative law

   AND: ((A ∧ B) ∧ C = A ∧ (B ∧ C))

OR: ((A ∨ B) ∨ C = A ∨ (B ∨ C))

3. Distributive law

   (A ∧ (B ∨ C) = (A ∧ B) ∨ (A ∧ C))

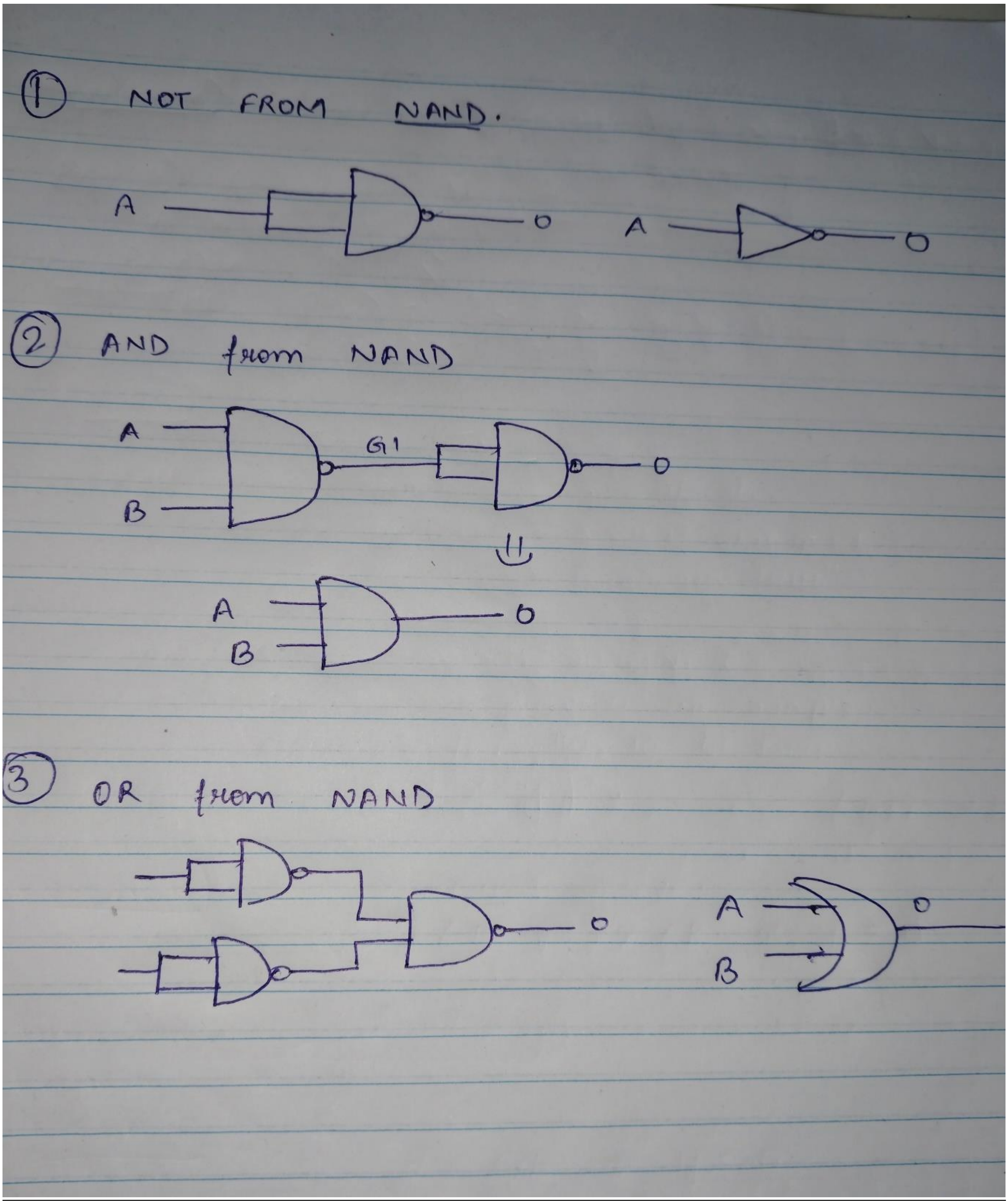
   (A ∨ (B ∧ C) = (A ∨ B) ∧ (A ∨ C))

4. De morgan’s law

   (¬(A ∧ B) = ¬A ∨ ¬B)

   (¬(A ∨ B) = ¬A ∧ ¬B)

CONSTRUCTING DIFFERENT GATES FROM NAND GATE:



DIGITAL LOGIC

CREATING AN XOR GATE

An XOR (Exclusive OR) gate is a digital logic gate that gives a true (1) output when the number of true inputs is odd. An XOR gate can be created using AND, OR, and NOT gates.

A XOR B = (A AND NOT B) OR (NOT A AND B)

From Logic Gates to Hardware in the Real World

The Role of Hardware Description Language (HDL)

HDL is a tool used by hardware designers and its function is to program the tool not by writing the code but by creating something physical for it which can contain the program. We write an HDL program to specify the structure and function of the chip and the testing is done virtually.

A hardware simulator is a special software and it takes the HDL program and creates a virtual model of the chip. This virtual chip runs through various scenarios and feeds different inputs and observes the outputs. Simulation has benefits like checking the accuracy of the chip working and optimization. Once the simulated chip meets all the requirements, the HDL gets finalised.

TWO'S COMPLEMENT NUMBER REPRESENTATION

Two's complement is a method used for encoding signed number in binary form. It has simplicity and efficiency in arithmetic operations.

BINARY ARITHMETIC

Binary addition

- 0 + 0 = 0
- 1 + 0 = 1
- 1 + 1 = 10 (where 0 is in the current digit and 1 is carried to the next higher digit)

Binary subtraction

similar to binary addition the binary subtraction is also the same as normal subtraction
* 1-1 = 0

*1-0 = 1

* 0-0 = 0

SIGNED AND UNSIGNED NUMBERS

- **Unsigned numbers:** These are always positive or zero and are straightforward to represent in binary.
- **Signed Numbers**: These can be positive, negative, or zero. Their representation is more complex as we need to encode the sign (positive or negative) along with the magnitude.

  TWO'S COMPLEMENT - Two's complement is a system for representing signed numbers where negative numbers are written as the complement of their absolute value.

Practical Applications

Computer Arithmetic

- Two's complement is extensively used in computer arithmetic to simplify the design of arithmetic logic units (ALUs).

Significance in Programming

- Understanding Two's complement is crucial for programmers, especially when dealing with low-level programming or languages like C and C++.

ARITHMETIC LOGIC UNIT (ALU)

The ALU is a critical component of the CPU in a computer. And is responsible for performing various operations on binary data. ALU can carry out operations like arithmetic operations (+- x /) and logical operations AND, OR, NOT, XOR. Arithmetic operations by the ALU are used for calculations in mathematical problems and data processing. Logic operations involve comparison and decision-making tasks, such as comparing numbers or evaluating conditions. Modern ALUs are capable of more complex operations, like shifting data bits to the left or right which can be used for rapid multiplication or division by powers of two, among other applications.

COMPUTER MEMORY

Computer memory is a way to store information. We need a place in the computer to store information which are called ‘registers’. for a computer to perform tasks like adding a series of numbers, it needs the ability to remember past operations. That’s when we use ‘sequential logic’. it uses time-dependent elements to hold onto past information

CLOCK

the lapse of time in a computer is represented by a master clock. This master clock consists of a continuous stream of alternating bits of 1s and 0s.

FLIP FLOPS

Flip-flops are the backbone of memory in computers, capable of holding a single bit of data. They are called “sequential” circuits because they are time-dependent, changing their state with the progression of clock signals.A flip-flop is a bistable device, meaning it has two stable states. Once it’s set to one state, it stays there until an input triggers a change. This ability to “remember” a state makes the flip-flop an essential component for memory storage. It’s the simplest form of sequential circuit we have, and with it, we can start to build more complex memory systems.

DATA FLIP FLOPS

At the heart of these systems is the Data Flip-Flop (DFF) gate, which is used to create a 1-bit memory cell. It has a single input and output, and its state is controlled by the input it receives and the timing of the clock. DFF gates are deeply integrated into computer architecture. They are used to create registers, which store the data that the computer’s processor is currently using, and counters, which keep track of occurrences or timings