KONERU LAKSHMAIAH EDUCATION FOUNDATION
(Deemed to be University estd, u/s, 3 of the UGC Act, 1956)
(NAAC Accredited "A++" Grade University)
Green Fields, Guntur District, A.P., India – 522502
Department of Basic Engineering Science - II

I B.Tech. II Semester – CSE / AI & DS / ECE / EEE / CS & IT / IOT

A.Y.2024-25 - EVEN SEMESTER

Digital Design and Computer Architecture (23EC1202)

CO – 1: Combinational Digital Logic Circuits

Session No.: 1

1. Course Description (Description of the subject):

The course on "Digital Design and Computer Architecture" provides a comprehensive exploration of the foundational principles in digital design process and computer organization. Students explore the concepts of combinational and sequential circuits, memory circuits. The curriculum extends to the Basic computer architecture concepts, memory hierarchies, and input/output fundamentals, fostering a deep understanding of computer organization. Through practical projects and simulations, students develop the skills to design and implement digital circuits. Graduates emerge with a robust skill set, ready to embark on careers in hardware design, computer architecture, and related fields, equipped to contribute to the ever-evolving landscape of digital technology.

2. Aim of the Course:

The course aims to equip students with the knowledge and skills related to:

i. Proficiency in designing and optimizing Combinational and Sequential Circuits using Boolean algebra and programmable logic devices with a solid foundation in digital design.

ii. Skill development using hands-on experience in designing digital circuits which includes latches, flip-flops, and counters in combination with memory, registers, and timing and sequence control modules using hardware & modeling tools.

iii. Explore the architecture of modern computers, including the organization and structure of central processing units, memory systems, and input/output interfaces.

iv. Bridge theoretical concepts with real-world applications by examining case studies and examples of digital design and computer architecture in modern computing systems.

Overall, the aim of the course is to prepare the student well-equipped to apply their knowledge to the design and analysis of digital systems and computer architectures, preparing them for careers in areas such as hardware design, computer engineering, and embedded systems development.

3. Instructional Objectives (Course Objectives):

The course objectives for "Digital Design and Computer Architecture" typically include:

i. To Understand and apply foundational concepts in digital design which results in proficiency over designing and analyzing combinational and sequential logic circuits.

ii. To Gain hands-on experience with industry-standard simulation and modeling tools, for verifying and testing digital designs.

iii. To analyze the architecture of a computer system, including the organization and operation of the CPU, memory hierarchy, and input/output subsystems.

iv. To apply digital design and computer architecture principles to solve real-world engineering problems and challenges by reinforcing theoretical knowledge with hands-on experience.

4. Learning Outcomes (Course Outcomes):

i. Able to build the combinational and programmable digital logic circuits using logic gates and optimization methods.

ii. Able to construct the sequential and memory circuits using flip-flops, demonstrating a comprehensive understanding of the principles governing clocked sequential logic.

iii. Able to organize computer architecture and instructions sequence through a grasp of the foundational principles that govern the organization and functioning of a computer system.

iv. Capable of modeling Memory Architecture and I/O Organization modules proficiently.

v. Able to develop and analyze the computer architecture modules using basic combinational, sequential and memory logics.

5. Module Description (CO - 1 Description):

The module covers essential topics in digital electronics, starting with Boolean algebra and progressing to the representation and optimization techniques of digital logic using SOP/POS forms. Students will delve into the design of key components such as adders, subtractors, multiplexers, de-multiplexers, decoders, and encoders. The module introduces the concept of reversible gates, exploring their unique properties. Additionally, students will gain insights into Programmable Logic Devices (PLDs) like PROM, PAL, and PLA, understanding their design principles. The implementation of Complex Programmable Logic Devices (CPLDs) with macrocells and Field-Programmable Gate Arrays (FPGAs) featuring Configurable Logic Blocks (CLBs) and Look-Up Tables (LUTs) will be covered. Practical applications of these digital logic modules in various scenarios will be emphasized, providing students with a comprehensive understanding of digital electronics and its real-world applications.

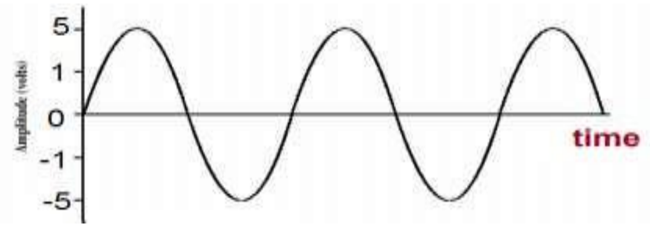
6. Introduction to Digital systems

System:

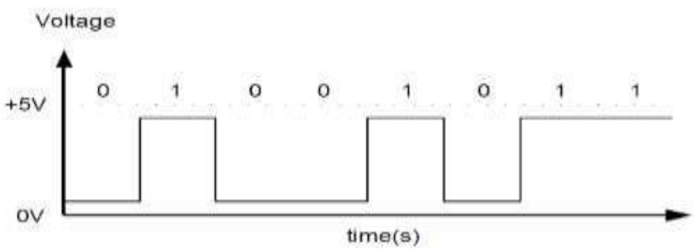
A system is a set of related components that work as a whole to achieve a goal. It contains inputs, behavior and outputs. Behavior is a function that translates inputs to output.

Components are electronics blocks – digital, analog, or mixed signal.

Analog system uses continuous signals –Infinite range of values



Digital system is a system in which signals have a finite number of discrete values (i.e binary value ‘1’ and ‘0’)



Mixed system has both analog and digital signals

Advantages of Digital system

1. Flexibility and functionality: easier to store, transmit and manipulate information
2. Economy: cheaper device and easier to design
3. High speed
4. High noise immunity

Applications of Digital system

1. Smart watch
2. ATM card
3. Computer
4. Laptop
5. Mobile phone
6. Digital clock
7. LED Display
8. Count down timer
9. Digital camera
10. Dish washers
11. Washing machine
12. Calculator

Digital systems are designed to store, process, and communicate information in digital form. They are found in a wide range of applications, including process control, communication systems, digital instruments, and consumer products. The digital computer, more commonly called the computer, is an example of a typical digital system. A computer manipulates information in digital, or more precisely, binary form. A binary number has only two discrete values — zero or one, low or high, true or false- 0 volts (is called "binary 0", or just "0") and 5 volts (is called "binary 1", or just "1")..Each of these discrete values is represented by the OFF and ON status of an electronic switch called a transistor. All computers, therefore, only understand binary numbers.

7. Review of Number Systems:

The study of number systems is important from the viewpoint of understanding how data are represented before they can be processed by any digital system including a digital computer. Different characteristics that define a number system include the number of independent digits used in the number system, the place values of the different digits constituting the number and the maximum numbers that can be written with the given number of digits. The base or radix of a number system is defined as the maximum number of digits or symbols that can be used in any position. The radix of the decimal number system is 10 as it has 10 independent digits, i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. Similarly, the binary number system with only two independent digits, 0 and 1, is a radix-2 number system. The octal and hexadecimal number systems have a radix (or base) of 8 and 16 respectively.

Decimal Number System:

The decimal number system is mainly suitable for human beings. As it uses 10 digits in any position of a given number, the base or radix is 10.

Example: $(152)_{10}$, $(82.45)_{10}$, $(98.79)_{10}$

It obeys positional notation rule. The value or magnitude of a given decimal number can be expressed as the sum of the various digits multiplied by their place values or weights.

Example: $(1245)_{10} = 1 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$

Binary Number System:

Most of the electronic components will have two state or binary operation ON and OFF. The decimal number system is not suitable to digital systems which contain two state operation. The number system used for digital systems is binary number system. The binary number system is a radix-2 number system with ‘0’ and ‘1’ as the two independent digits. All larger binary numbers are represented in terms of ‘0’ and ‘1’. These symbols are known as binary digits or simply bits. It also obeys positional notation rule.

Example: $(101001)_2$, $(11.01101)_2$

$$(1001.11)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

Octal Number System:

The octal number system has a radix of 8 and therefore has eight distinct digits. The independent digits are 0, 1, 2, 3, 4, 5, 6 and 7. It also obeys positional notation rule.

Example: $(754.16)_8$, $(557.63)_8$

$$(714.77)_8 = 7 \times 8^2 + 1 \times 8^1 + 4 \times 8^0 + 7 \times 8^{-1} + 7 \times 8^{-2}$$

Hexadecimal Number System:

To represent large numbers in digital systems like computers, hexadecimal number system is used. Here the base is 16 and hence 16 digits or symbols (0-9) and (A-F) can be placed in any given number. This number system obeys positional notation rule.

Example: $(1AC.BD)_{16}$, $(2BC)_H$

$(3CD.4E)_{16} = 3 \times 16^2 + C \times 16^1 + D \times 16^0 + 4 \times 16^{-1} + E \times 16^{-2}$

Number	Decimal Number System	Binary Number System	Octal Number System	Hexadecimal Number System
0	0	0	0	0
1	1	1	1	1
2	2	10	2	2
3	3	11	3	3
4	4	100	4	4
5	5	101	5	5
6	6	110	6	6
7	7	111	7	7
8	8	1000	10	8
9	9	1001	11	9
10	10	1010	12	A
11	11	1011	13	B
12	12	1100	14	C
13	13	1101	15	D
14	14	1110	16	E
15	15	1111	17	F
16	16	10000	20	10

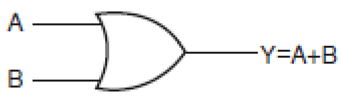
8. Logic Gates

Logic gates are electronic circuits that can be used to implement the most elementary logic expressions, also known as Boolean expressions. The logic gate is the most basic building block of combinational logic. There are three basic logic gates, namely the OR gate, the AND gate and the NOT gate. Other logic gates that are derived from these basic gates are the NAND gate, the NOR gate, the EXCLUSIVEOR gate and the EXCLUSIVE-NOR gate. This chapter deals with logic gates and implementations using NAND and NOR gates followed by simplification of Boolean functions using Boolean Laws and theorems and using K-maps.

OR Gate

A logic gate used to perform the operation of logical addition is called an OR gate. An OR gate performs an ORing operation on two or more than two logic variables. The OR operation on two independent logic variables A and B is written as Y = A+B and reads as Y equals A OR B. An OR gate is a logic circuit with two or more inputs and one output. The output of an OR gate is LOW only when all of its inputs are LOW. For all other possible input combinations, the output is HIGH. A truth table lists all possible combinations of input binary variables and the corresponding outputs of a logic system. Figure shows the circuit symbol and the truth table of a two-input OR gate. The operation of a two-input OR gate is explained by the logic expression

$$Y = A+B$$



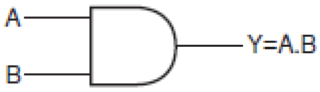
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Two input OR Gate

AND Gate

A logic gate used to perform logical multiplication is known as AND gate. An AND gate is a logic circuit having two or more inputs and one output. The output of an AND gate is HIGH only when all of its inputs are in the HIGH state. In all other cases, the output is LOW. The logic symbol and truth table of a two-input AND gate is shown in figure. The AND operation on two independent logic variables A and B is written as Y = A.B and reads as Y equals A AND B. The operation of a two-input AND gate is explained by the logic expression

$$Y = A.B$$



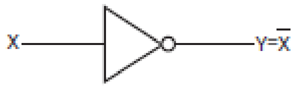
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Two input AND Gate

NOT Gate

A logic gate used to perform logical inversion is known as a NOT gate. A NOT gate is a one -input, one-output logic circuit whose output is always the complement of the input. That is, a LOW input produces a HIGH output, and vice versa. If X is the input to a NOT circuit, then its output Y is given by Y = $\bar{X}$ or X' and reads as Y equals NOT X. The logic symbol and truth table of a NOT gate is shown in figure. The operation of a NOT gate is explained by the logic expression

$$Y = \bar{X}$$



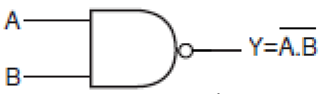
X	Y
0	1
1	0

NOT Gate

NAND Gate

NAND stands for NOT AND. An AND gate followed by a NOT circuit makes it a NAND gate. The output of a NAND gate is logic ‘0’ when all its inputs are logic ‘1’. For all other input combinations, the output is logic ‘1’. The symbol and truth table of a NAND gate is as shown. NAND gate operation is logically expressed as

$$Y = \overline{AB}$$



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

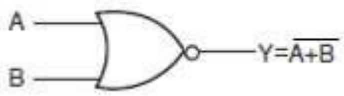
Two input NAND Gate

NAND Gate is known as Universal gate as it can be used alone to implement any gate operation. Hence it is said to be functionally complete.

NOR Gate

NOR stands for NOT OR. An OR gate followed by a NOT circuit makes it a NOR gate. The output of a NOR gate is logic ‘1’ when all its inputs are logic ‘0’. For all other input combinations, the output is logic ‘0’. The symbol and truth table of a NOR gate is as shown. The output of a two-input NOR gate is logically expressed as

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



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

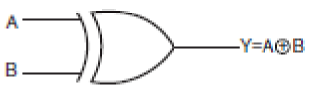
Two input NOR Gate

NOR gate is also known as Universal gate as it is used alone to implement any gate operation and hence it is also functionally complete.

EXCLUSIVE-OR Gate

The EXCLUSIVE-OR gate, commonly written as EX-OR gate, is a two-input, one-output gate. The output of an EX-OR gate is logic ‘1’ when the inputs are unlike and logic ‘0’ when the inputs are like. Although EX-OR gates are available in integrated circuit form only as two-input gates, unlike other gates which are available in multiple inputs also, multiple-input EX-OR logic functions can be implemented using more than one two-input gates. The output of a multiple-input EX-OR logic function is logic ‘1’ when the number of 1s in the input sequence is odd and logic ‘0’ when the number of 1s in the input sequence is even, including zero. The symbol and truth table of an EX-OR gate is shown in figure. The output of a two-input EX-OR gate is logically expressed as

$$Y = A \oplus B = A'B + AB'$$



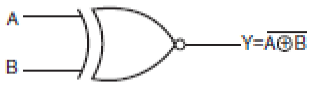
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Two input EX-OR Gate

EXCLUSIVE-NOR Gate

EXCLUSIVE-NOR (commonly written as EX-NOR) means NOT of EX-OR, i.e. the logic gate that we get by complementing the output of an EX-OR gate. The truth table of an EX-NOR gate is obtained from the truth table of an EX-OR gate by complementing the output entries as shown in figure. Logically,

$$Y = \overline{A \oplus B} = A'B' + AB$$



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

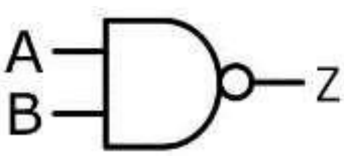
Two input EX-NOR Gate

The output of a two-input EX-NOR gate is logic '1' when the inputs are like and logic '0' when they are unlike. In general, the output of a multiple-input EX-NOR logic function is logic '0' when the number of 1s in the input sequence is odd and a logic '1' when the number of 1s in the input sequence is even including zero.

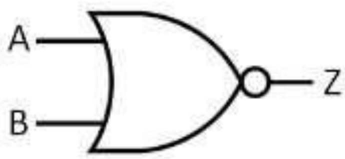
Universal Gates

OR, AND and NOT gates are the three basic logic gates as they together can be used to construct the logic circuit for any given Boolean expression. NOR and NAND gates have the property that they individually can be used to hardware-implement a logic circuit corresponding to any given Boolean expression. That is, it is possible to use either only NAND gates or only NOR gates to implement any Boolean expression. This is so because a combination of NAND gates or a combination of NOR gates can be used to perform functions of any of the basic logic gates. It is for this reason that NAND and NOR gates are universal gates.

NAND Gate



Nor Gate



8. Terminal Questions:

- Explain the basic operations of the AND, OR, and NOT gates in Boolean algebra.
- How to convert a decimal number to binary number?
- List the number system representations for decimal numbers 0 to1

9. References books:

- 1. Computer System Architecture by M. Moris Mano
- 2. Fundamentals of Digital Logic with Verilog HDL by Stephen Brown and ZvonkoVranesic

10. Sites and Web links:

- https://www.studyadda.com/notes/9th-class/computers-science/allchapter/introduction-to-logic-gates-and-number-system/13264