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 9 & 10: CPLD and FPGA

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.

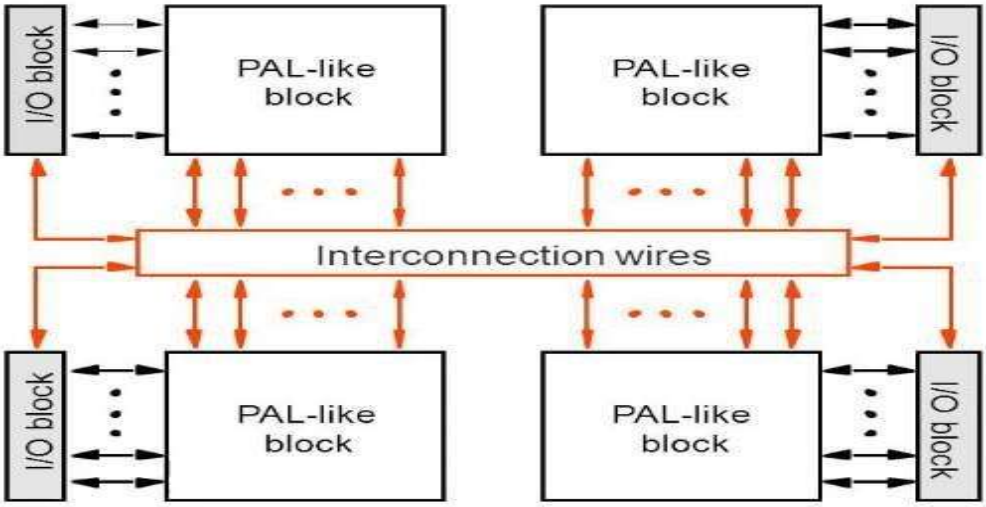
6. Session Introduction:

CPLD (Complex Programmable Logic Device) and FPGA (Field-Programmable Gate Array) are both types of programmable logic devices used in electronic circuits for implementing digital logic functions. They allow designers to create complex digital logic circuits without the need for custom fabrication of a physical circuit for each new design. Despite their similarities, CPLDs and FPGAs differ significantly in terms of architecture, capabilities, and typical applications.

6.1 Complex Programmable Logic Device (CPLD):

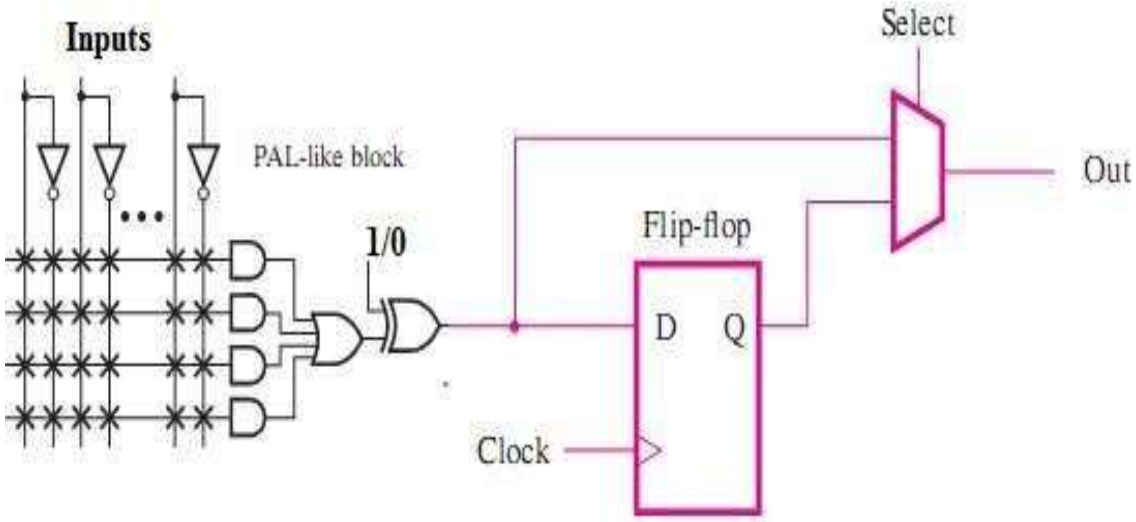
CPLD is an abbreviation for Complex Programmable Logic Device. It is a digital programmable logic device with a programmable logic array (PLA) and a programmable interconnect. CPLDs have been developed to perform a wide range of logic functions and are often used in digital circuits to handle data processing, control, and communication.

In a CPLD, the programmable interconnect connects the output of one logic cell to the input of another. The interconnect can be designed to generate a wide range of logic connections, allowing complicated logic functions to be created. The interconnect is often made up of a matrix of programmable switches that connect the logic cells' inputs and outputs.



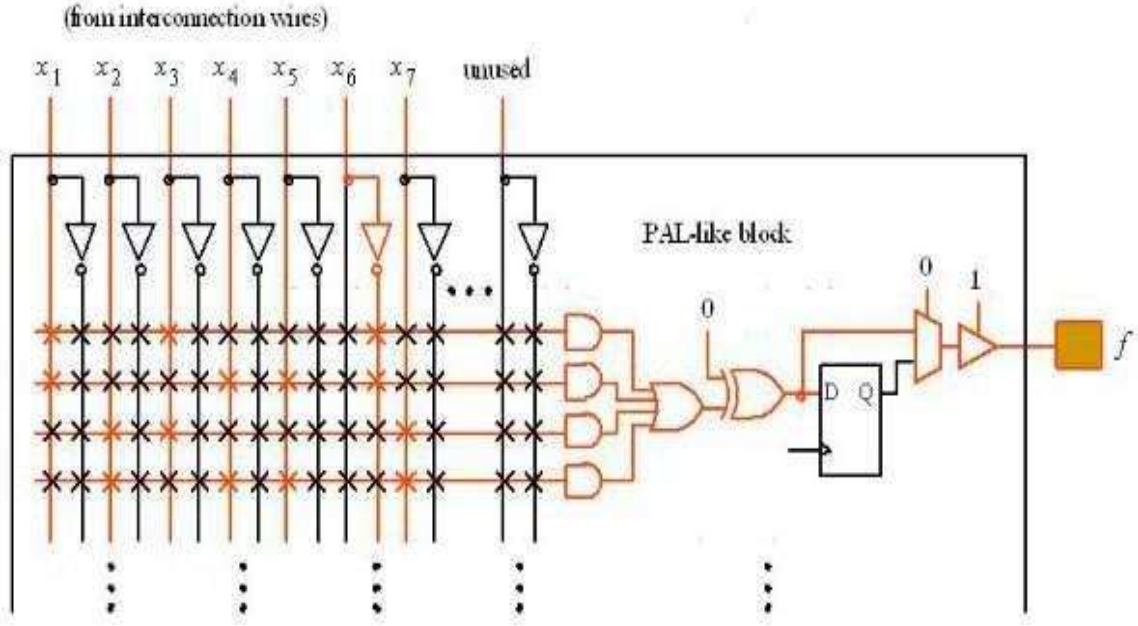
One of the benefits of CPLDs is their ease of use. Because CPLDs typically have a specified link structure, they can be designed and implemented faster than FPGAs. (Field Programmable Gate Arrays). They are also less expensive than FPGAs, making them an attractive option for low- to medium-complexity systems.

6.1.1 Internal Structure of a PAL Block (Macro Cell)



6.1.2 Logic implementation using macrocell / CPLD:

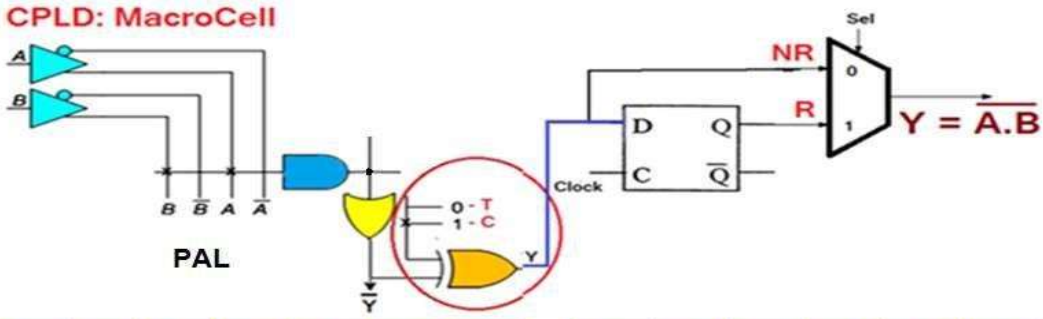
Problem 1: Implement the given Boolean function using CPLD. F = x1 x3 x6'+ x1 x4 x5 x6' + x2 x3 x7 + x2 x4 x5 x7



Problem 2: Implement the Boolean function Y = (A.B) ' = ∑ m (0,1,2) using macro-cell.

	T/C	P	Y
T	0	0	0
	0	1	1
C	1	0	1
	1	1	0

X-OR Gate



A	B	CLK	NR	R	SEL	Y
0	0	0	1	Z	0	1
0	1	0	1	Z	0	1
1	0	0	1	Z	0	1
1	1	0	0	Z	0	0
1	0	0	1	Z	1	Z

A	B	CLK	NR	R	SEL	Y
0	0	1	1	1	1	1
0	1	1	1	1	1	1
1	0	1	1	1	1	1
1	1	1	0	0	1	0
1	0	0	1	Z	0	1

6.1.3 Applications of CPLD:

- CPLDs are ideal for high performance, critical control applications.
- CPLDs are used to implement control logic in embedded systems.
- CPLDs are employed in DSP applications to enable parallel processing, optimizing the execution of algorithms that require simultaneous data processing.
- CPLDs are utilized to implement complex state machines, enabling efficient control and sequencing of operations in applications such as communication protocols and digital controllers.

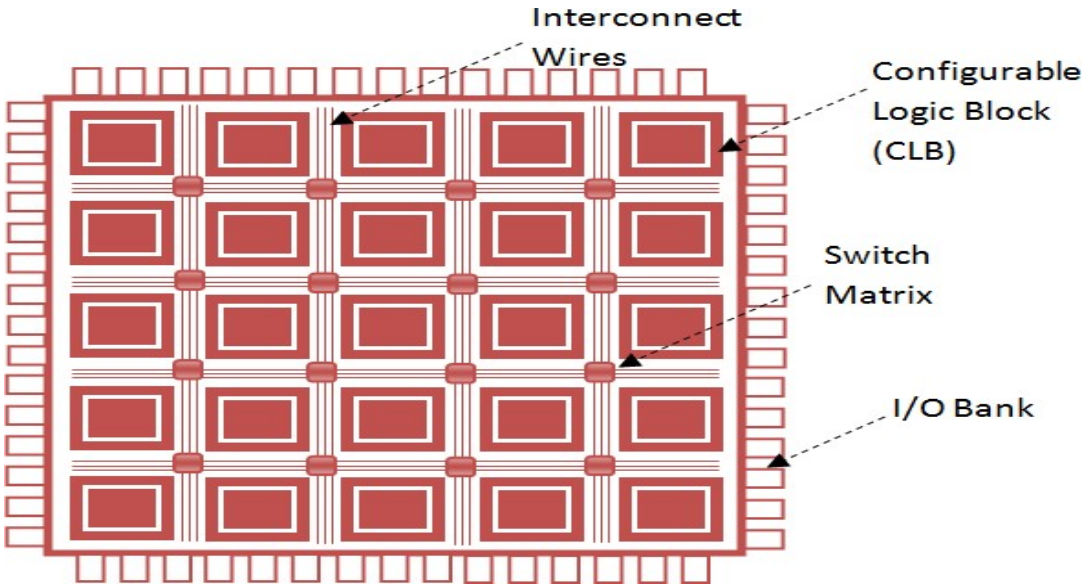
6.2 Field-Programmable Gate Array (FPGA):

FPGA is an abbreviation for Field Programmable Gate Array, which is a type of programmable logic device used to build digital circuits. FPGAs are constructed from a grid of programmable logic blocks that can be coupled to perform sophisticated logic tasks. FPGAs feature more logic blocks than CPLDs and are therefore better suited for implementing larger and more complicated logic functions.

For implementing simpler logic operations, CPLDs are generally faster and more power-efficient than FPGAs. FPGAs are more flexible and provide greater performance and scalability when it comes to implementing larger and more complicated logic operations. Furthermore, because of their bigger size and better level of integration, FPGAs are often more expensive than CPLDs.

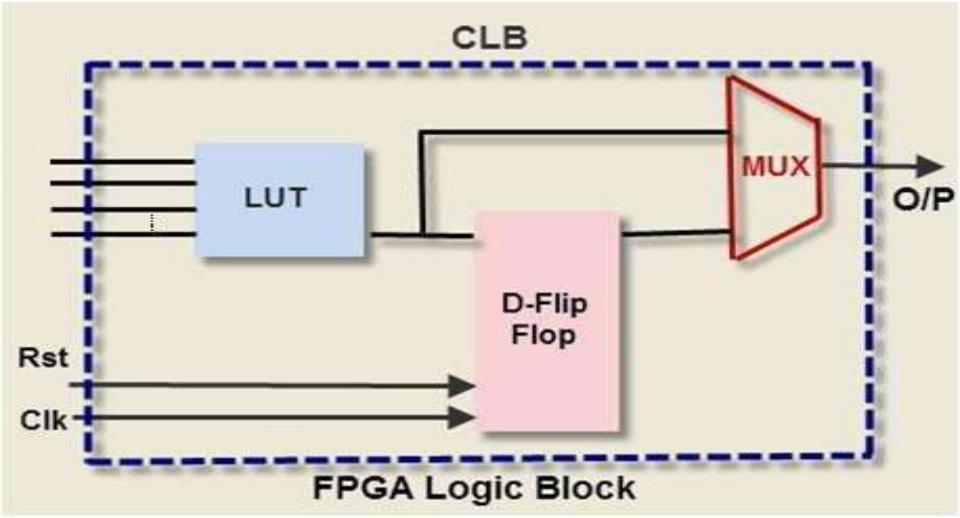
The logic cell, also known as a configurable logic block, is the fundamental building block of an FPGA. A CLB is made up of a look-up table (LUT), a flip-flop, and a programmable interconnect. The LUT is a memory block that can be designed to perform a specific logic function. The flip-flop stores the logic function's output, and the programmable interconnect connects the output of one CLB to the input of another.

FPGAs often have a large number of CLBs that can be configured in a two-dimensional array or matrix. An FPGA's connection topology is far more versatile than that of a CPLD, allowing for more complicated designs. The interconnect is often made up of programmable switches that may be set to connect the CLBs' inputs and outputs.



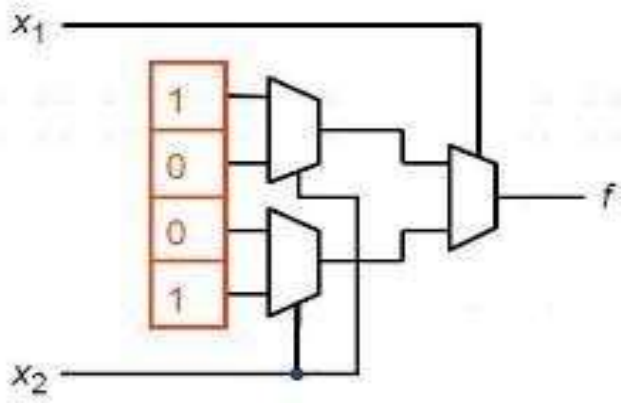
6.2.1 CLB and LUT in FPGA:

A CLB, or Configurable Logic Block, is a fundamental building block of FPGA (Field-Programmable Gate Array) devices. FPGAs are highly versatile and reconfigurable silicon chips used in a wide range of applications, from digital signal processing to complex system-on-chip (SoC) designs. The flexibility of FPGAs largely stems from their architecture, which consists of an array of CLBs interconnected by programmable routing resources. Understanding the structure and function of CLBs is key to grasping how FPGAs can be programmed to perform a vast array of digital functions.



The exact structure of a CLB can vary between different FPGA manufacturers and even between different FPGA families from the same manufacturer. However, most CLBs share some common features:

Look-Up Tables (LUTs): A LUT is a programmable truth table that can implement any logic function of its inputs. The number of inputs to a LUT (and thus the complexity of the logic function it can represent) can vary, but 4-input and 6-input LUTs are common. By programming the LUT, any logic operation can be implemented.



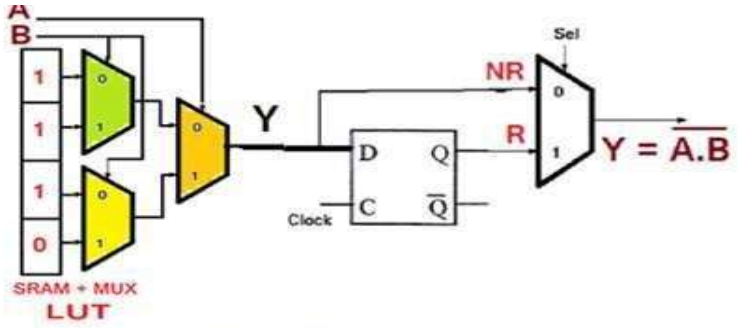
- LUTs comprise of 1-bit SRAM cells to hold either '0' or '1" and a set of multiplexers.
- The number of inputs available for a LUT determines its size.
- In general, a LUT with *n* inputs comprise of 2^n single-bit SRAM memory cells followed by a 2^n -1 multiplexers.
- Example of an LUT with 2 inputs

***Problem 1**:* Implement the CLB for the given Boolean function Y=(A.B)‘ or Y(A,B)= ∑m(0,1,2)

- ***Solution:***

FPGA: CLB

A	B	Σm	Y
0	0	0	1
0	1	1	1
1	0	2	1
1	1	3	0
NAND Gate			



A	B	CLK	NR	R	SEL	Y
0	0	0	1	Z	0	1
0	1	0	1	Z	0	1
1	0	0	1	Z	0	1
1	1	0	0	Z	0	0
1	0	0	1	Z	1	Z

A	B	CLK	NR	R	SEL	Y
0	0	1	1	1	1	1
0	1	1	1	1	1	1
1	0	1	1	1	1	1
1	1	1	0	0	1	0
1	0	0	1	Z	0	1

6.2 CPLD Vs FPGA:

Characteristics	CPLD	FPGA
Logic Cells	It has a small number of logic cells.	It has a large number of logic cells.
Interconnect Structure	It has a fixed interconnect structure.	It has a flexible interconnect structure.
Flexibility	Less Flexible	More Flexible
Cost	Low Cost	High Cost
Power Consumption	Less power consumption	Higher power consumption
Reconfigurability	Less reconfigurability	More reconfigurability
Density	Low to medium	Medium to high
Flip-flop ratios	Less flip-flop ratio	More flip-flop ratio
Applications	Best for simple applications	Best for complex applications

- In conclusion, FPGAs contain a far higher number of logic cells and a more flexible connection topology, making them better suited for larger, more complicated designs. They provide greater reconfigurability and can serve a broader range of applications. Their more complex architecture can result in longer design times, higher prices, and higher power consumption.
- CPLDs feature fewer logic cells and are best suited for simpler logic functions and smaller designs. They have a simpler architecture with a set connecting structure, which can result in quicker design times and reduced costs. They also consume less electricity and are better suited for low-volume production runs.

7. SAQ's – Self Assessment Questions:

1. The complex programmable logic device contains several PLD blocks and _____

a) A language compiler

b) AND/OR arrays

c) Global interconnection matrix

d) Field-programmable switches

2. Which type of device FPGA are?

a) SLD

b) SROM

c) EPROM

d) PLD

3. In FPGA, vertical and horizontal directions are separated by _____

a) A line

b) A channel

c) A strobe

d) A flip-flop

4. SPLDs, CPLDs, and FPGAs are all which type of device?

a) PAL

b) PLD

c) EPROM

d) SRAM

8. Summary:

- In conclusion, FPGAs contain a far higher number of logic cells and a more flexible connection topology, making them better suited for larger, more complicated designs. They provide greater reconfigurability and can serve a broader range of applications. Their more complex architecture can result in longer design times, higher prices, and higher power consumption.
- CPLDs feature fewer logic cells and are best suited for simpler logic functions and smaller designs. They have a simpler architecture with a set connecting structure, which can result in quicker design times and reduced costs. They also consume less electricity and are better suited for low-volume production runs.

9. Terminal Questions:

- Draw the architecture of a Complex Programmable Logic Device (CPLD) and its key components.
- Discuss the role of macro cells in CPLD architecture.
- Sketch the block diagram of a CPLD, highlighting main blocks.
- Implement the CLB for the given Boolean function $Y(A,B)= \sum m(0,1,2)$.
- Illustrate the flexibility and programmability enabled by configuring Look-Up Tables (LUTs) in FPGAs.
- Compare and contrast CPLD and FPGA architectures.
- Illustrate the internal structure of Macrocell in CPLD using D Flip-flop and give the insights.
- Design the function $F = x1\ x3\ x6' + x1\ x4\ x5\ x6' + x2\ x3'\ x7 + x2'\ x4\ x5\ x7$ using CPLD and flip-flops.
- Implement the CLB for the given Boolean function $Y(A,B)= \sum m(1,2,3)$.
- Draw & present insights of FPGA architecture, focusing on Look-Up Tables (LUT) & Configurable Logic Blocks (CLB).

10. Glossary:

CPLDs (Complex Programmable Logic Devices) and FPGAs (Field-Programmable Gate Arrays) are integral components in the field of digital electronics, enabling the design and implementation of reconfigurable logic circuits. CPLDs, characterized by their fixed interconnect matrix and macrocells, excel in implementing straightforward, deterministic logic functions efficiently, making them ideal for control applications and device initialization tasks. On the other hand, FPGAs offer a more flexible architecture composed of Configurable Logic Blocks (CLBs) that include Look-Up Tables (LUTs), flip-flops, and a versatile network of programmable routing resources, allowing for the realization of complex, high-density logic functions. Both devices leverage Hardware Description Languages (HDLs) like VHDL and Verilog for design and configuration, making them adaptable to a wide range of applications—from simple logic control to sophisticated signal processing and system-on-chip (SoC) designs. Despite their differences, CPLDs and FPGAs share the fundamental goal of providing designers with a platform to develop and iterate digital circuits with unprecedented speed and flexibility.

11. References books:

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

12. Sites and Web links:

- https://www.tutorialspoint.com/difference-between-cpld-and-fpga
- https://www.geeksforgeeks.org/xilinx-fpga-architecture/
- https://www.scribd.com/document/191760393/UNIT-I-CPLD-FPGA-Architectures