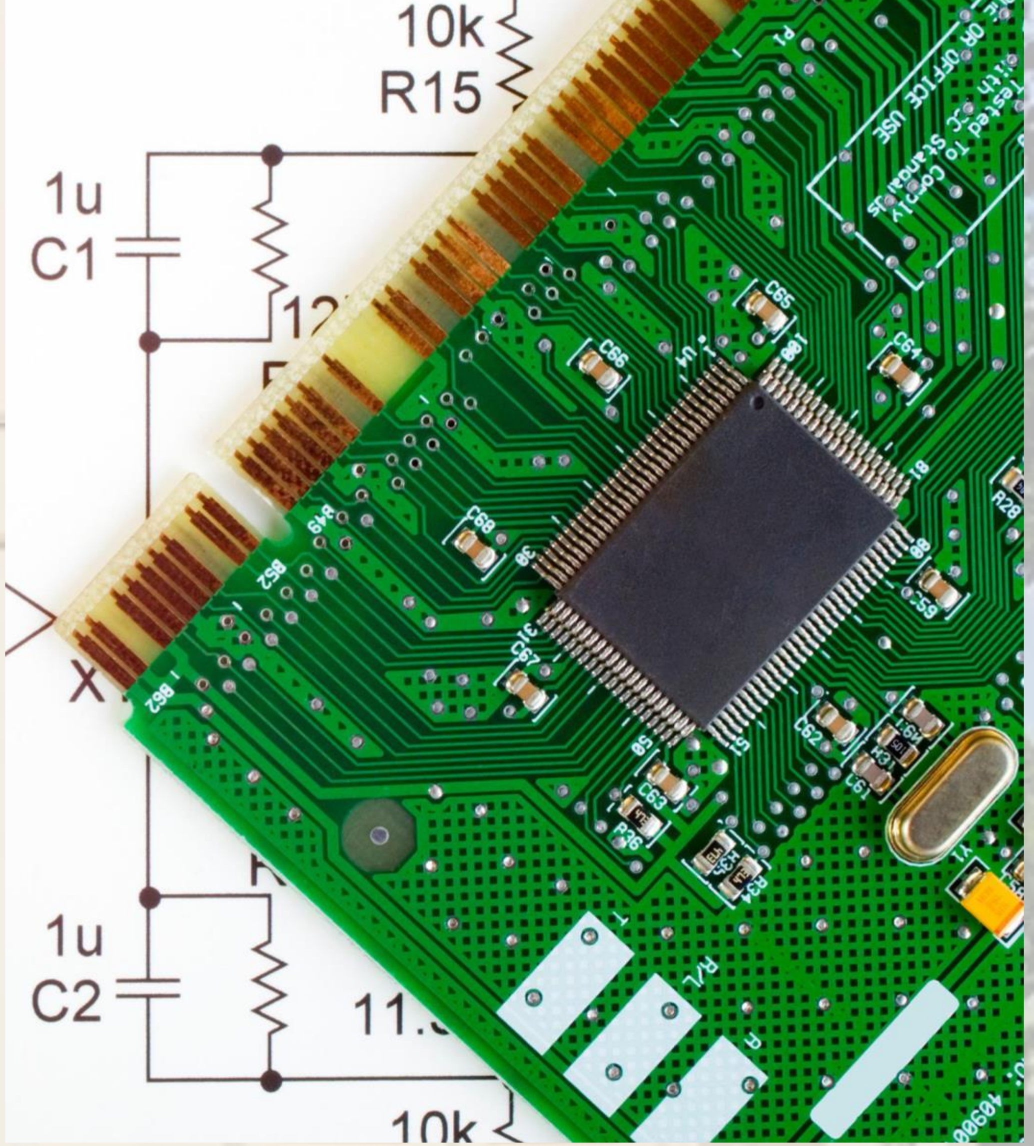
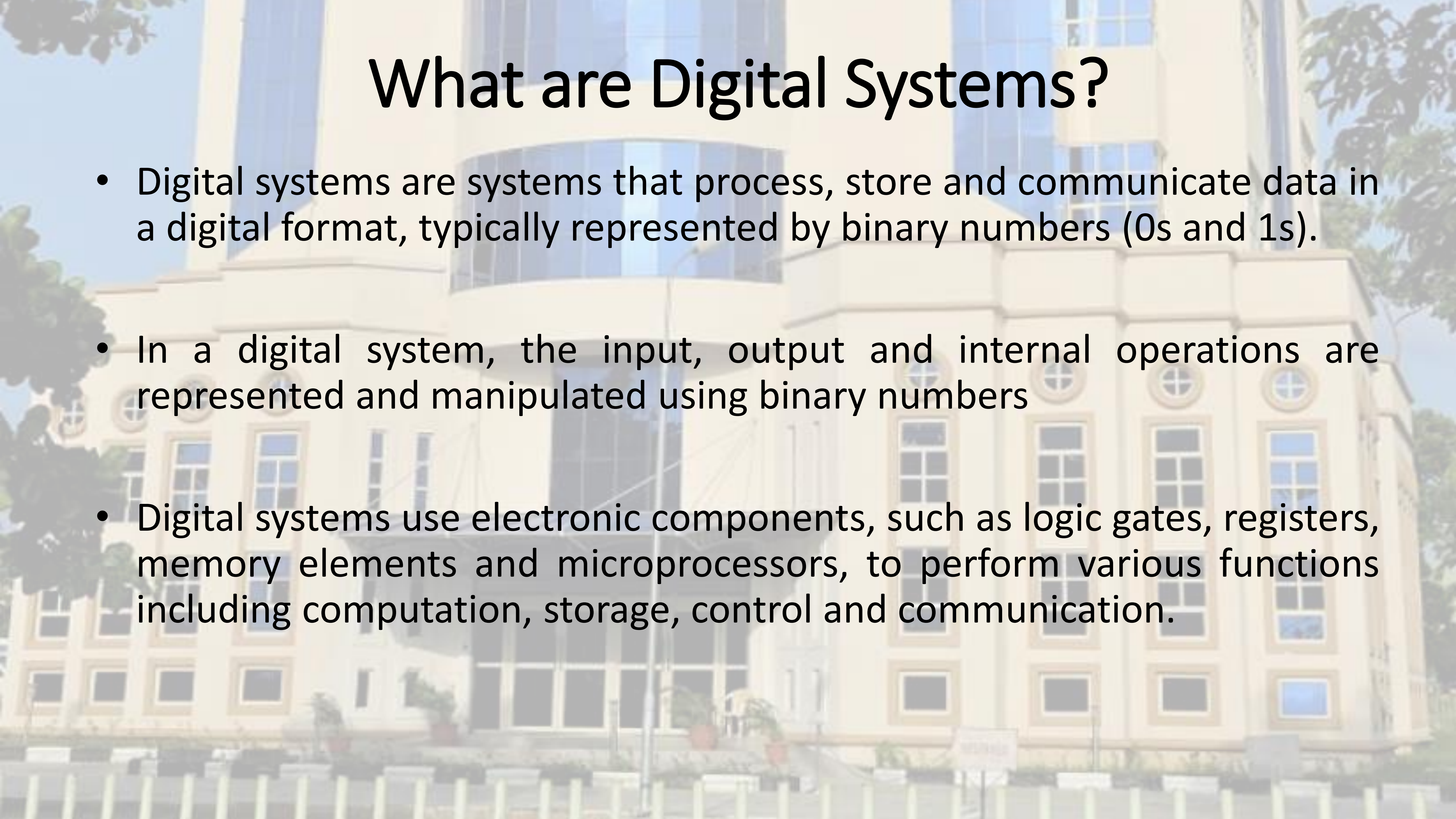
CEN 510:

Digital System Design with VHDL



Course Lecturers: Mr. Adekunle OJEWANDE & Mr. Chibueze UBOCHI

Department of Electrical and Information Engineering
Covenant University, Nigeria

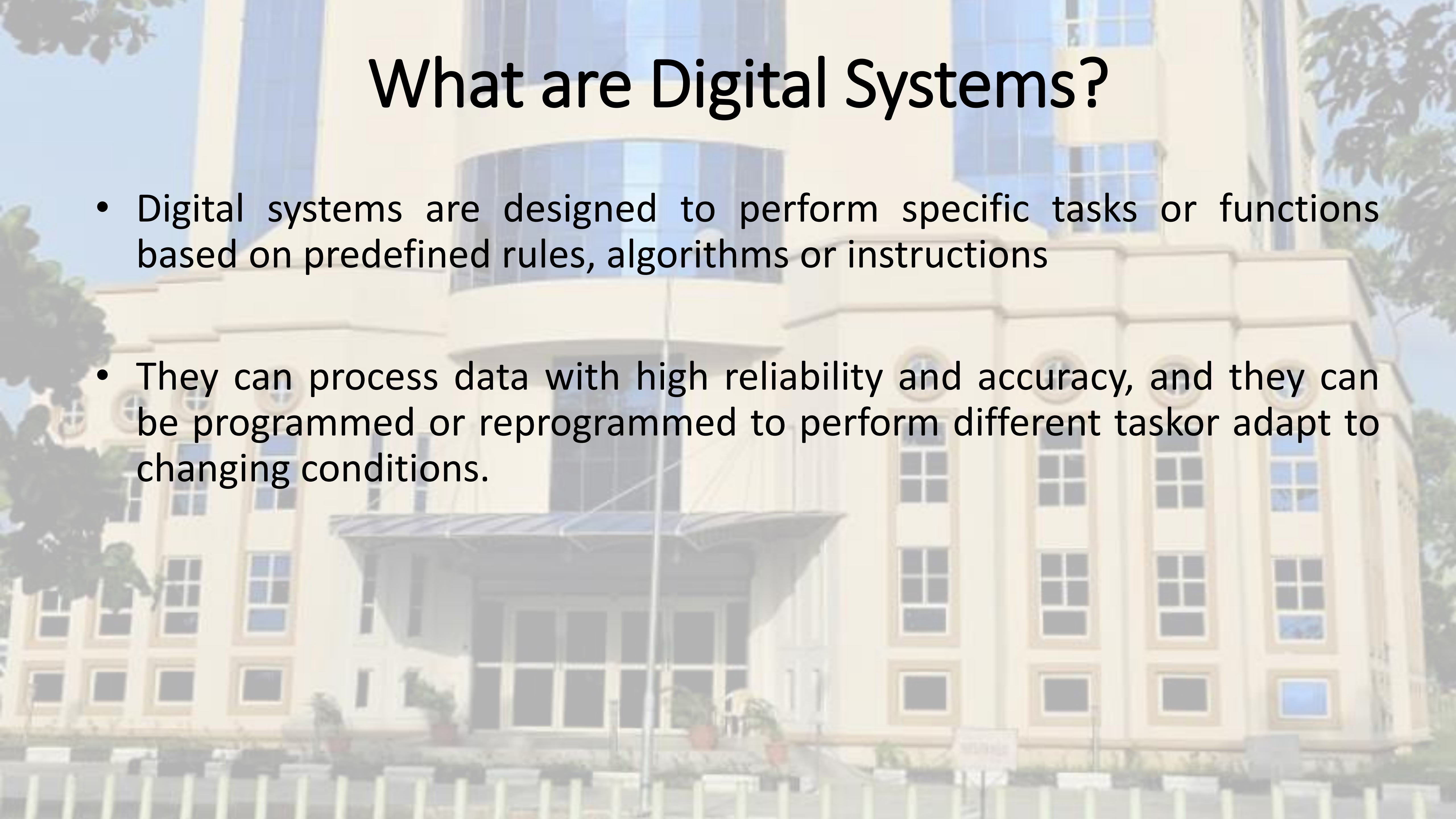


What are Digital Systems?

- Digital systems are systems that process, store and communicate data in a digital format, typically represented by binary numbers (0s and 1s).

- In a digital system, the input, output and internal operations are represented and manipulated using binary numbers

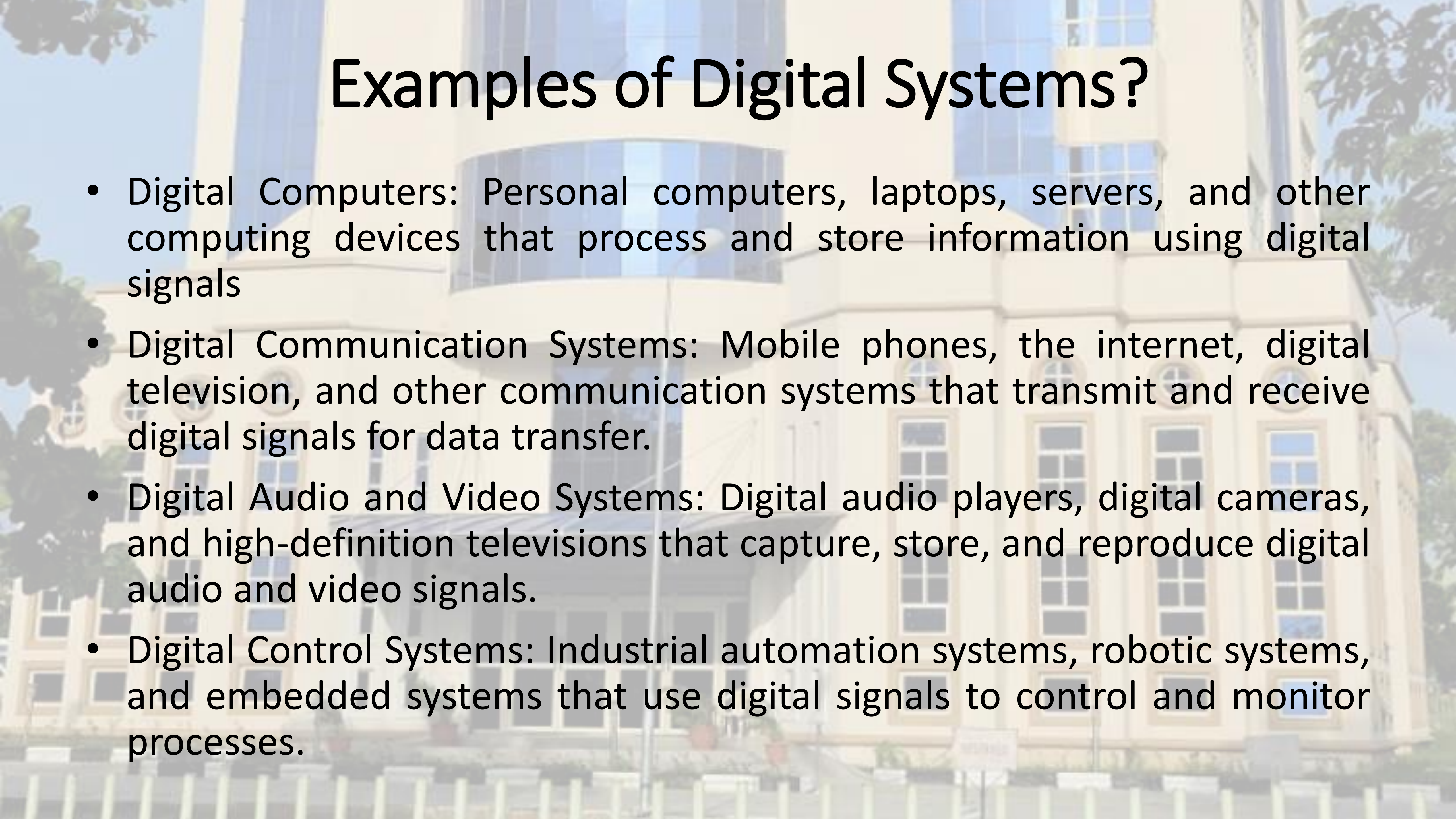
- Digital systems use electronic components, such as logic gates, registers, memory elements and microprocessors, to perform various functions including computation, storage, control and communication.



What are Digital Systems?

- Digital systems are designed to perform specific tasks or functions based on predefined rules, algorithms or instructions

- They can process data with high reliability and accuracy, and they can be programmed or reprogrammed to perform different taskor adapt to changing conditions.



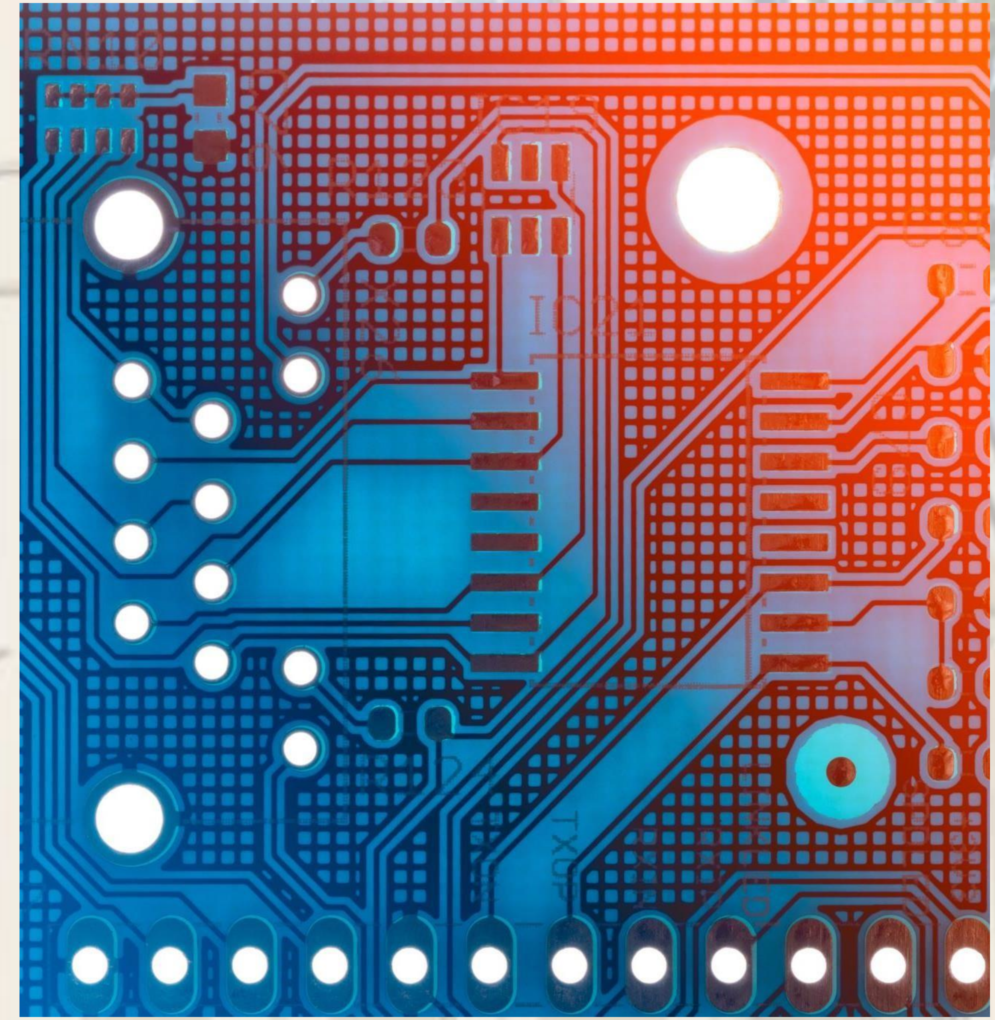
Examples of Digital Systems?

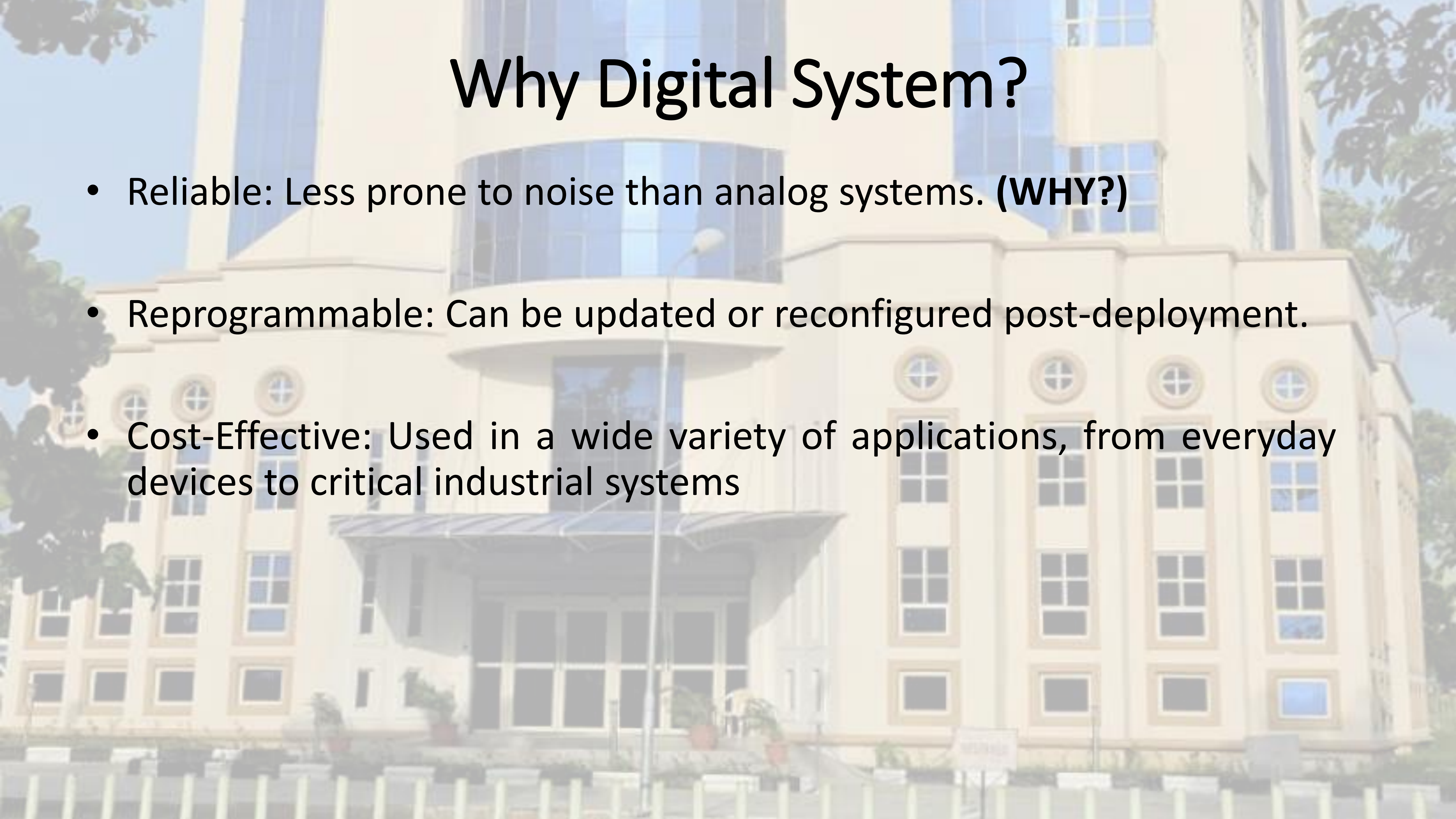
- Digital Computers: Personal computers, laptops, servers, and other computing devices that process and store information using digital signals
- Digital Communication Systems: Mobile phones, the internet, digital television, and other communication systems that transmit and receive digital signals for data transfer.
- Digital Audio and Video Systems: Digital audio players, digital cameras, and high-definition televisions that capture, store, and reproduce digital audio and video signals.
- Digital Control Systems: Industrial automation systems, robotic systems, and embedded systems that use digital signals to control and monitor processes.

What is Digital System Design?

- Digital system design is the process of creating electronic circuits and systems that operate on digital signals.

- VHDL is a hardware description language used to design digital chips, circuits and systems.



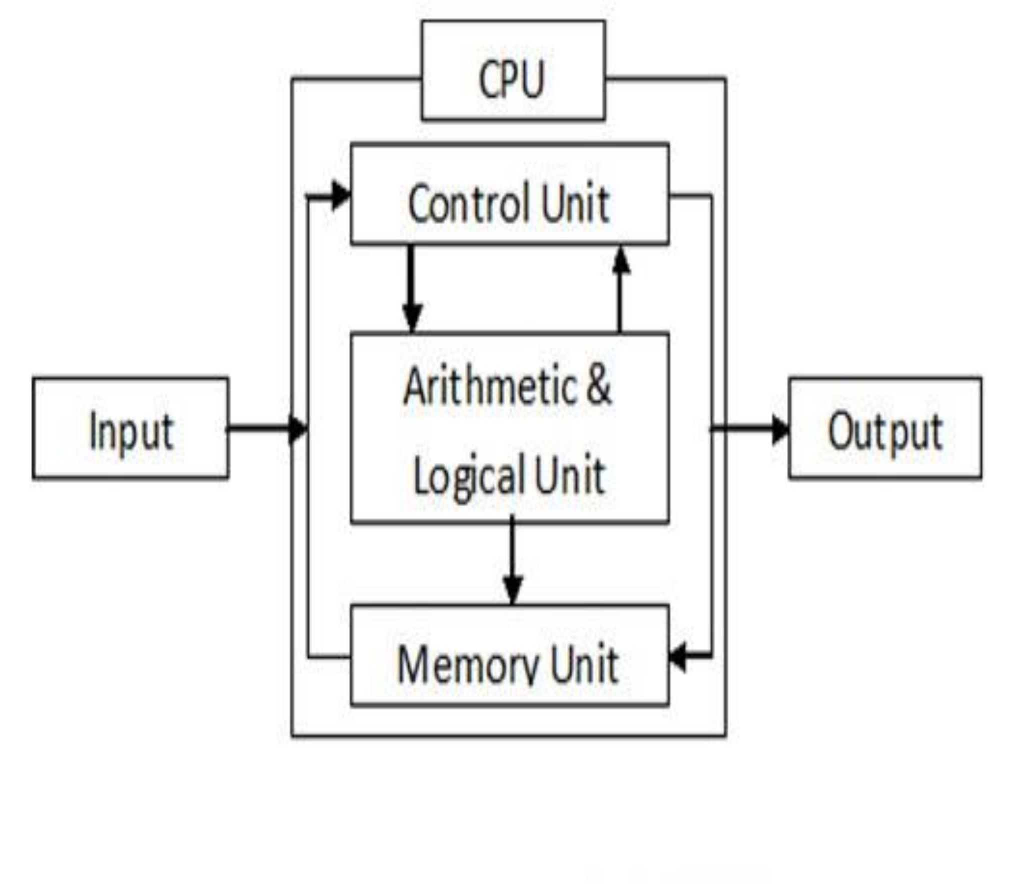


Why Digital System?

- Reliable: Less prone to noise than analog systems. **(WHY?)**
- Reprogrammable: Can be updated or reconfigured post-deployment.
- Cost-Effective: Used in a wide variety of applications, from everyday devices to critical industrial systems

Key Elements of Digital System

- Inputs: Sensors, user interfaces.
- Processing Units: Arithmetic Logic Units (ALU), control logic, memory.
- Outputs: Displays, actuators, communication modules.



Digital System Design

- Digital system design is the process of designing circuits to meet desired system requirements or perform specific tasks.

- They are built using digital components such as logic gates, flip-flops, and microprocessors.

Digital System Design Steps:

- Specification: Define the system’s goals and requirements.
- Design: Create the architecture and components.
- Implementation: Use hardware description languages (HDL).
- Testing: Verify and debug the system.

Digital System Design Approaches

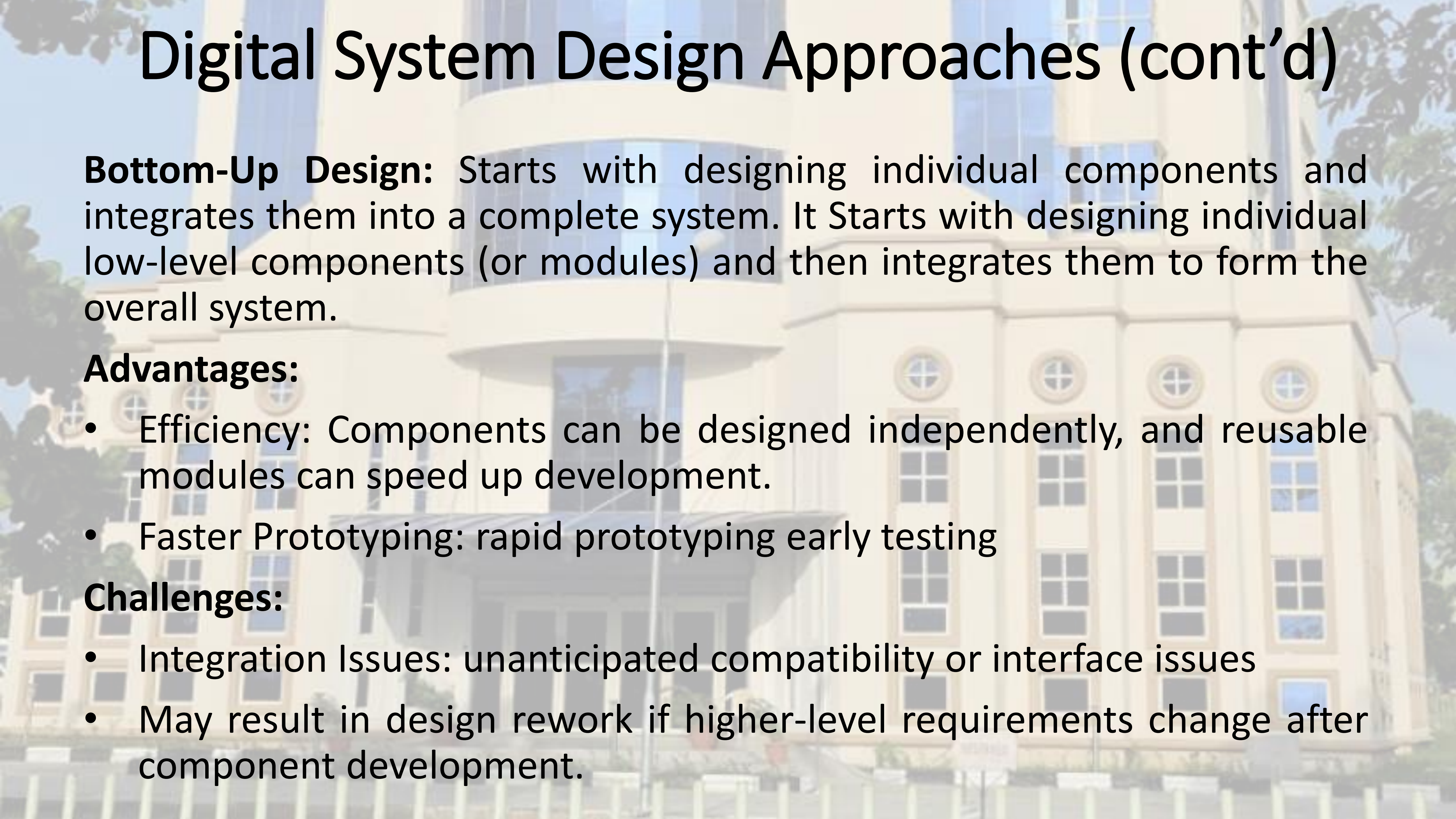
Top-Down Design: Starts with the most abstract specification and moves toward detailed implementations. It begins with the high-level functionality or system specification and breaks it down into smaller components until each part can be implemented.

Advantages

- Modularity: easy to debug and test subsystems separately
- Scalability: new features or functionalities can be added incrementally

Challenges

- More time-consuming, requiring careful planning and reworking
- Requires an accurate understanding of the entire system upfront



Digital System Design Approaches (cont’d)

Bottom-Up Design: Starts with designing individual components and integrates them into a complete system. It Starts with designing individual low-level components (or modules) and then integrates them to form the overall system.

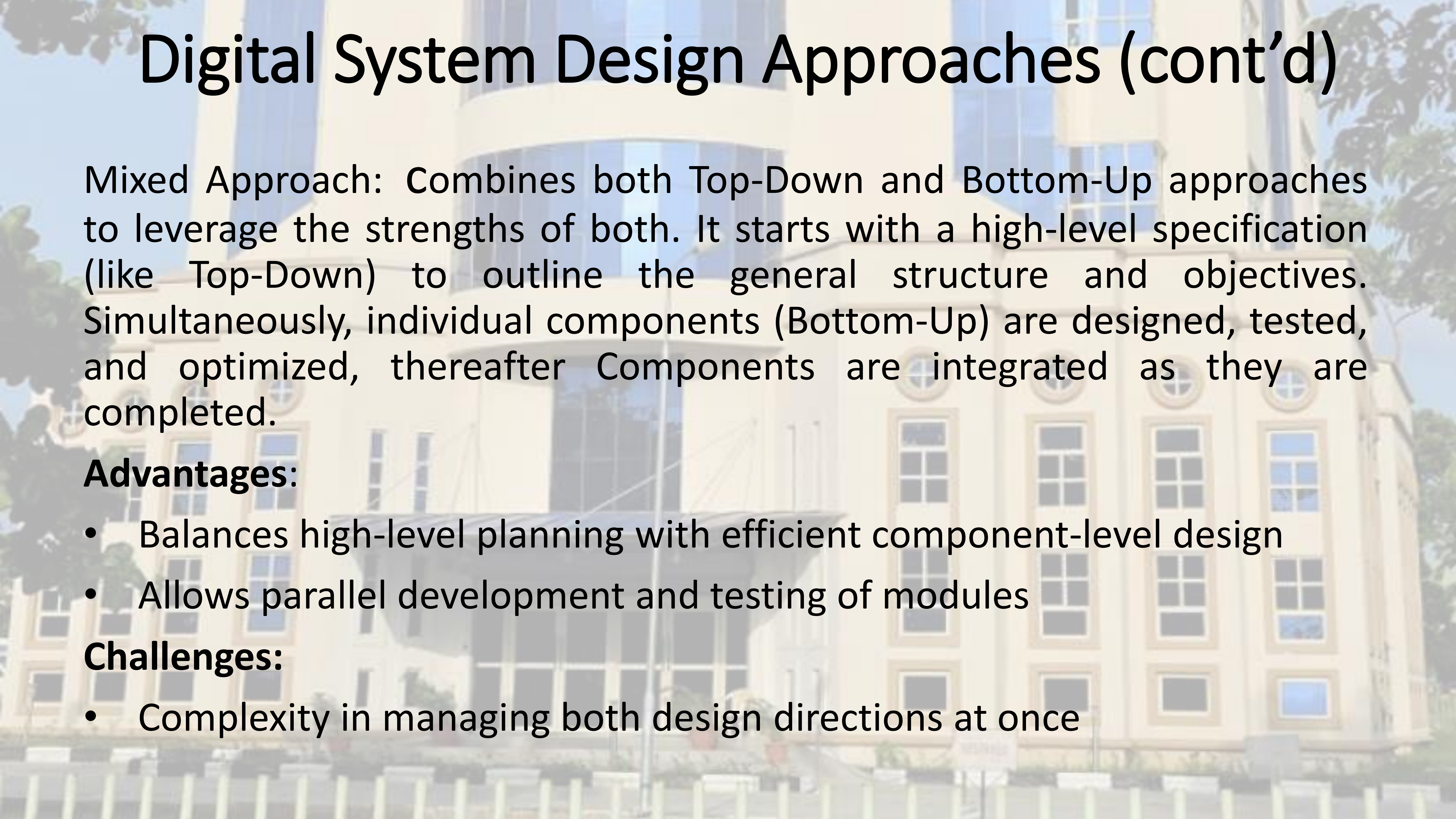
Advantages:

- Efficiency: Components can be designed independently, and reusable modules can speed up development.
- Faster Prototyping: rapid prototyping early testing

Challenges:

- Integration Issues: unanticipated compatibility or interface issues
- May result in design rework if higher-level requirements change after component development.

Digital System Design Approaches (cont’d)



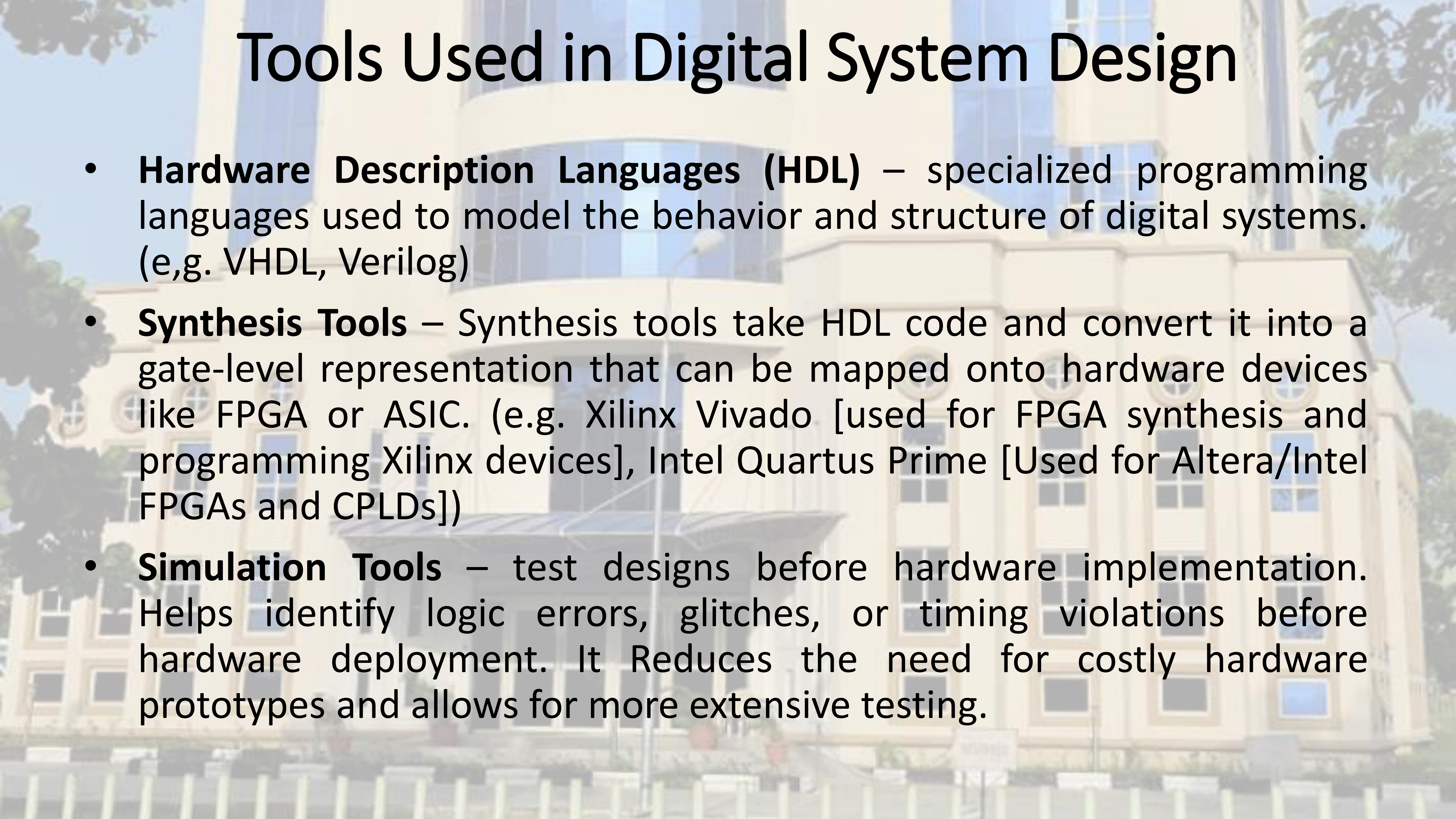
Mixed Approach: Combines both Top-Down and Bottom-Up approaches to leverage the strengths of both. It starts with a high-level specification (like Top-Down) to outline the general structure and objectives. Simultaneously, individual components (Bottom-Up) are designed, tested, and optimized, thereafter Components are integrated as they are completed.

Advantages:

- Balances high-level planning with efficient component-level design
- Allows parallel development and testing of modules

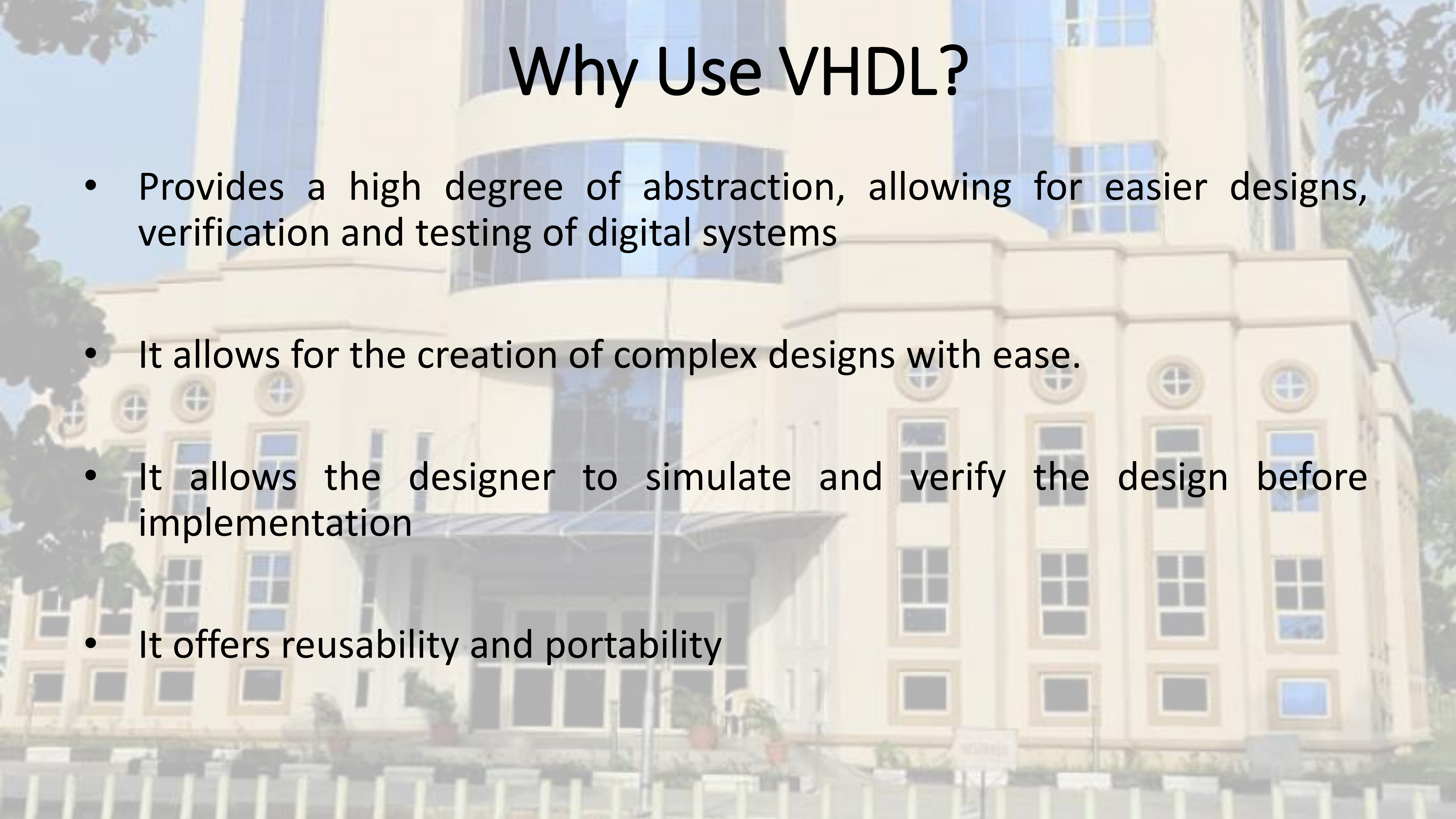
Challenges:

- Complexity in managing both design directions at once



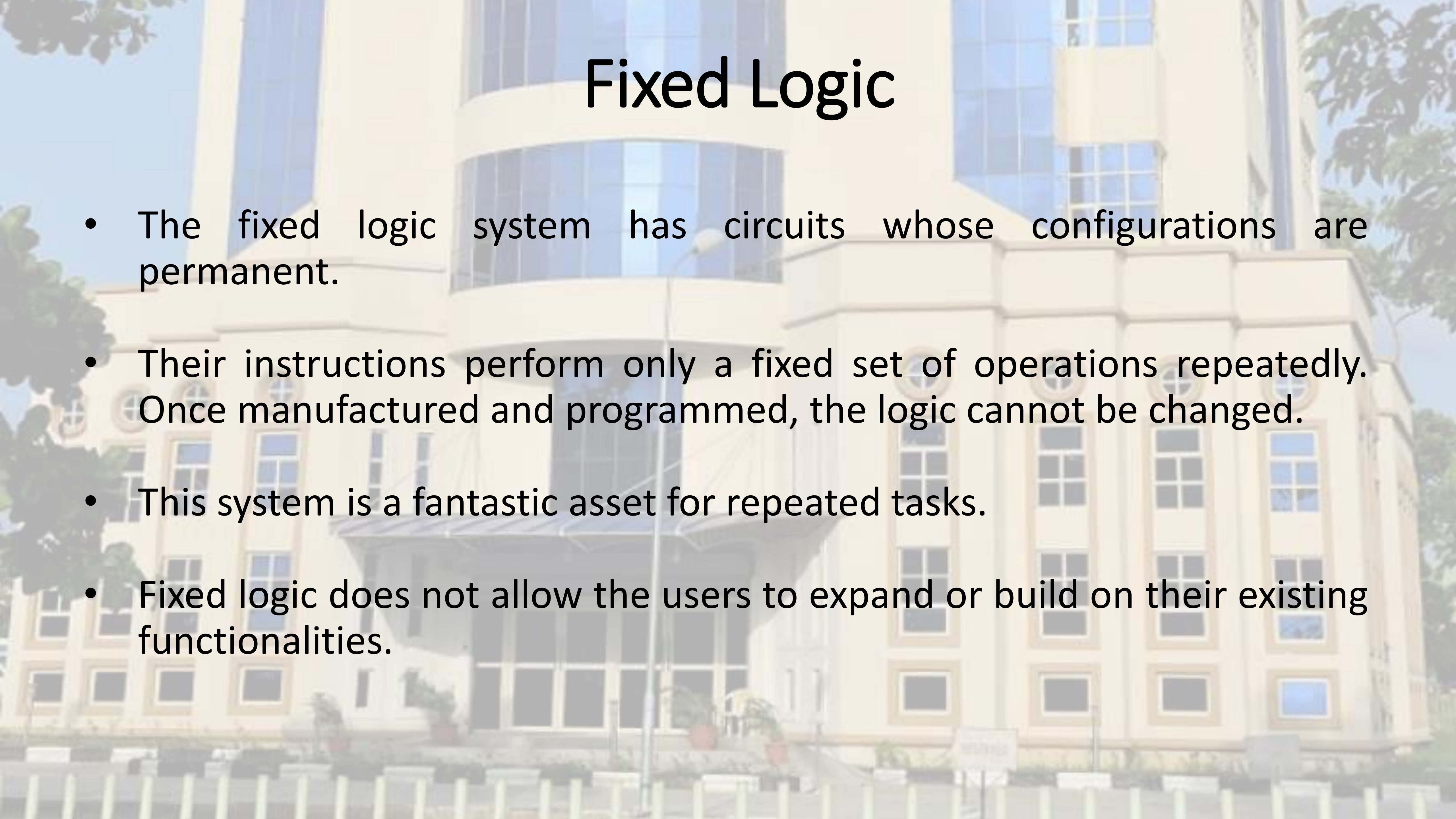
Tools Used in Digital System Design

- **Hardware Description Languages (HDL)** – specialized programming languages used to model the behavior and structure of digital systems. (e,g. VHDL, Verilog)
- **Synthesis Tools** – Synthesis tools take HDL code and convert it into a gate-level representation that can be mapped onto hardware devices like FPGA or ASIC. (e.g. Xilinx Vivado [used for FPGA synthesis and programming Xilinx devices], Intel Quartus Prime [Used for Altera/Intel FPGAs and CPLDs])
- **Simulation Tools** – test designs before hardware implementation. Helps identify logic errors, glitches, or timing violations before hardware deployment. It Reduces the need for costly hardware prototypes and allows for more extensive testing.



Why Use VHDL?

- Provides a high degree of abstraction, allowing for easier designs, verification and testing of digital systems
- It allows for the creation of complex designs with ease.
- It allows the designer to simulate and verify the design before implementation
- It offers reusability and portability



Fixed Logic

- The fixed logic system has circuits whose configurations are permanent.

- Their instructions perform only a fixed set of operations repeatedly. Once manufactured and programmed, the logic cannot be changed.

- This system is a fantastic asset for repeated tasks.

- Fixed logic does not allow the users to expand or build on their existing functionalities.

Reconfigurable Logic

- Reconfigurable logic – Logic that can be configured to perform different functions even after it is manufactured

Types

- Programmable Array Logic (PAL)
- Field-Programmable Gate Arrays (FPGA)

Benefits of Reconfigurable Logic

- Flexibility: Can be reprogrammed to meet new requirements.
- Prototyping: Ideal for testing and modifying designs.
- Cost-Effective: Reduces the need for multiple devices.

Embedded System

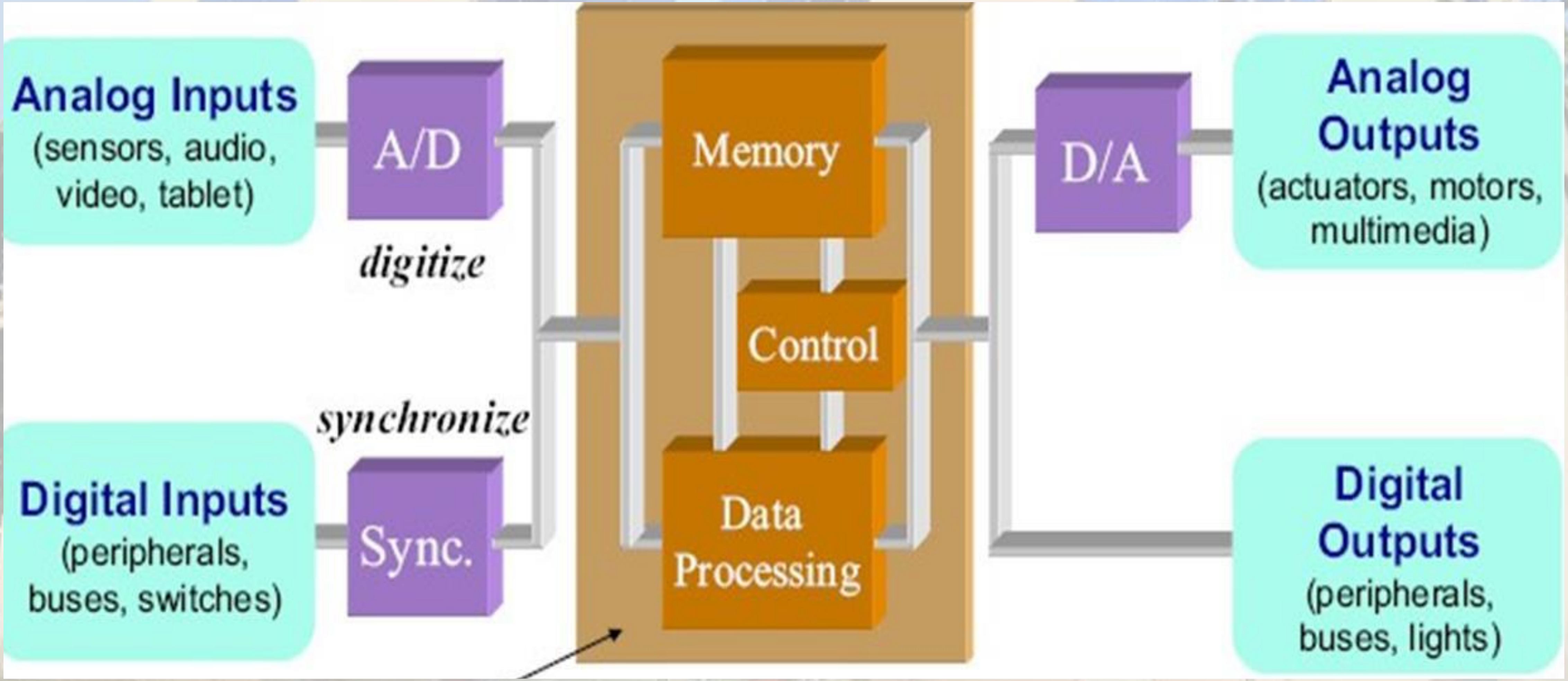
- Any digital system that manipulates data using some dedicated, application-specific software is an *embedded system.*

- An embedded system is a combination of computer hardware and software designed for a specific function.

- Embedded systems may also function within a larger system. The systems can be programmable or have a fixed functionality.

- Examples: Industrial machines, consumer electronics, agricultural and processing industry devices, automobiles, medical equipment, cameras, digital watches, household appliances, airplanes, vending machines and toys, as well as mobile devices, are possible locations for an embedded system

Embedded System (cont’d)


Analog Inputs
(sensors, audio, video, tablet)
A/D
digitize
synchronize
Digital Inputs
(peripherals, buses, switches)
Sync.
Memory
Control
Data Processing
D/A
Analog Outputs
(actuators, motors, multimedia)
Digital Outputs
(peripherals, buses, lights)

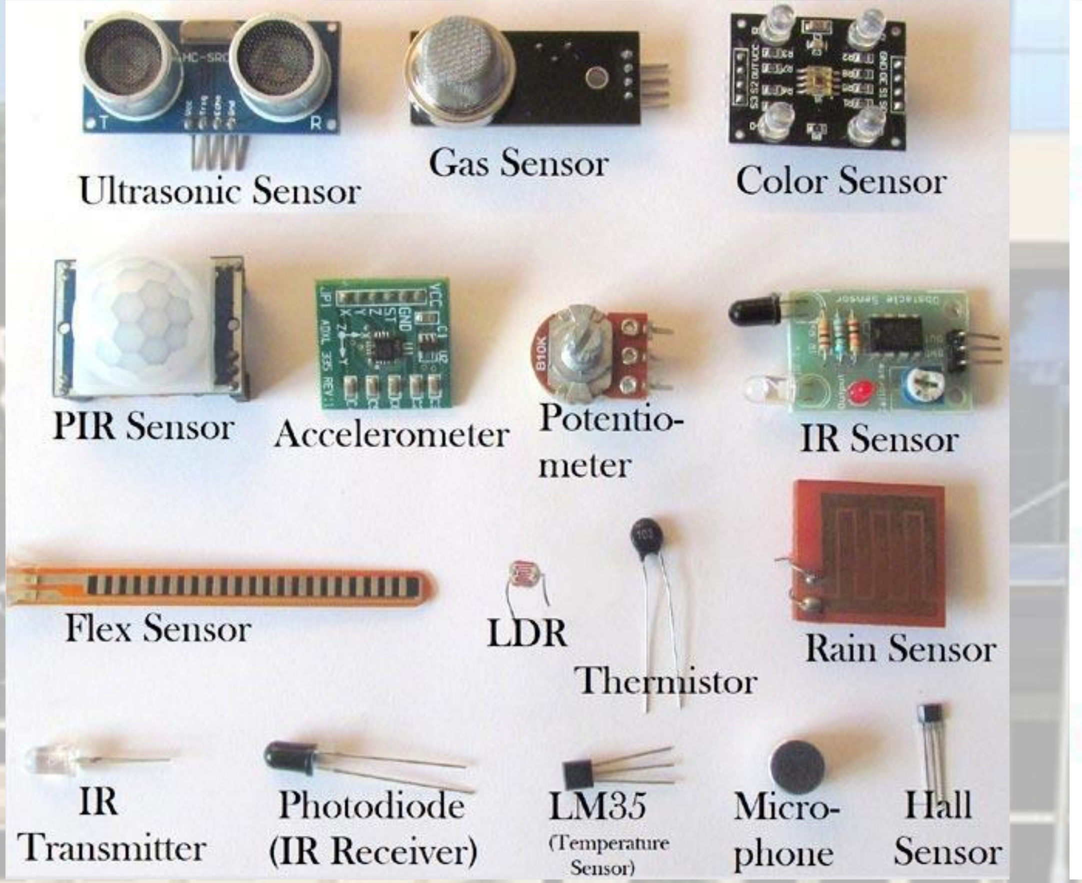
Embedded System (cont’d)

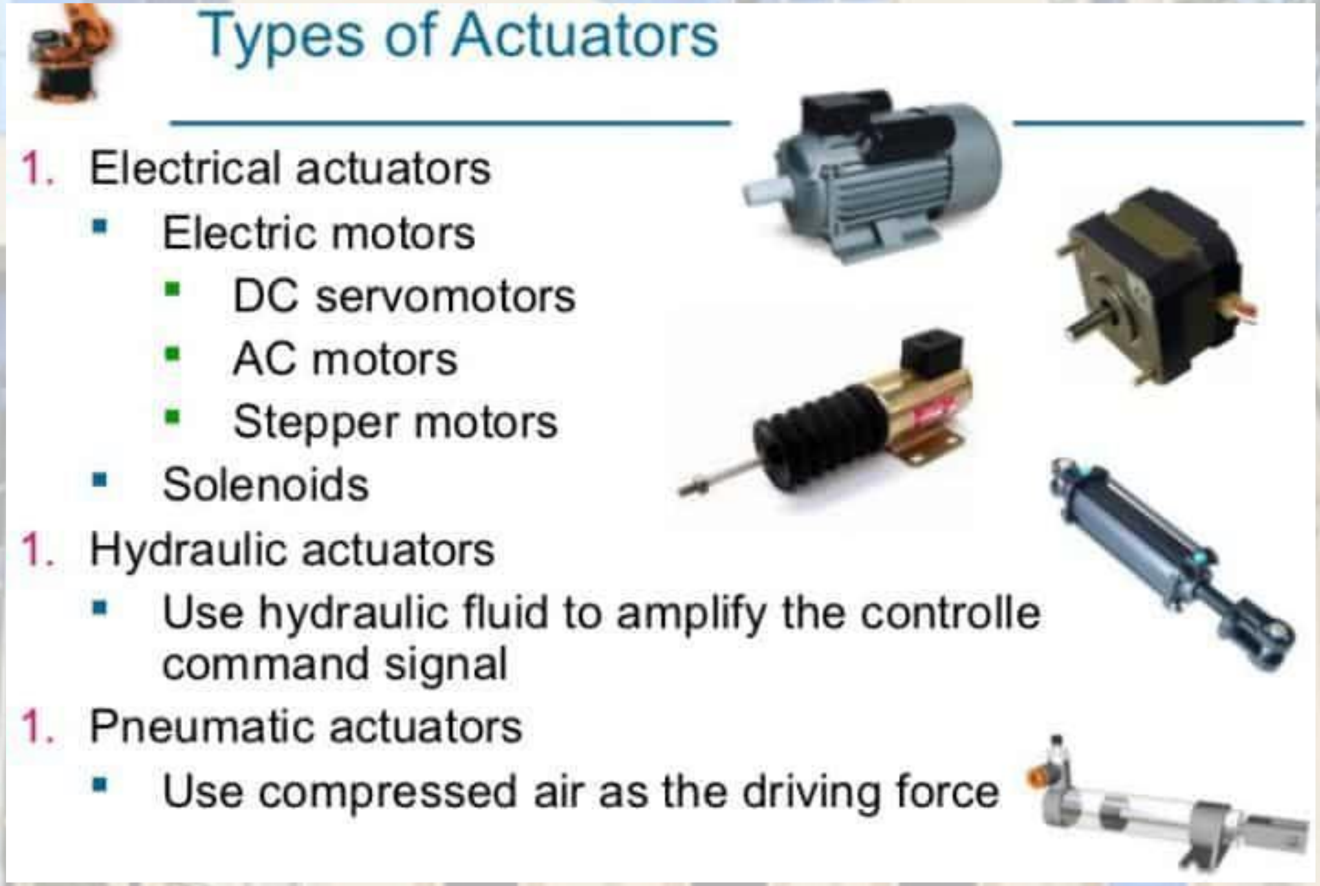
- Sensors and actuators are critical components of several embedded systems.
- A sensor is a device that detects and responds to some type of input from the physical environment. There are a broad range of sensors used in everyday life, which are classified based on the quantities and quality they detect.
- An actuator is a part of a device or machine that helps it to achieve physical movements by converting energy, often electrical, air, or hydraulic, into mechanical force. Simply put, it is the component in any machine that enables movement.

Examples of Embedded Systems

Many Different Products Depend on Embedded Systems

Embedded System (cont'd)

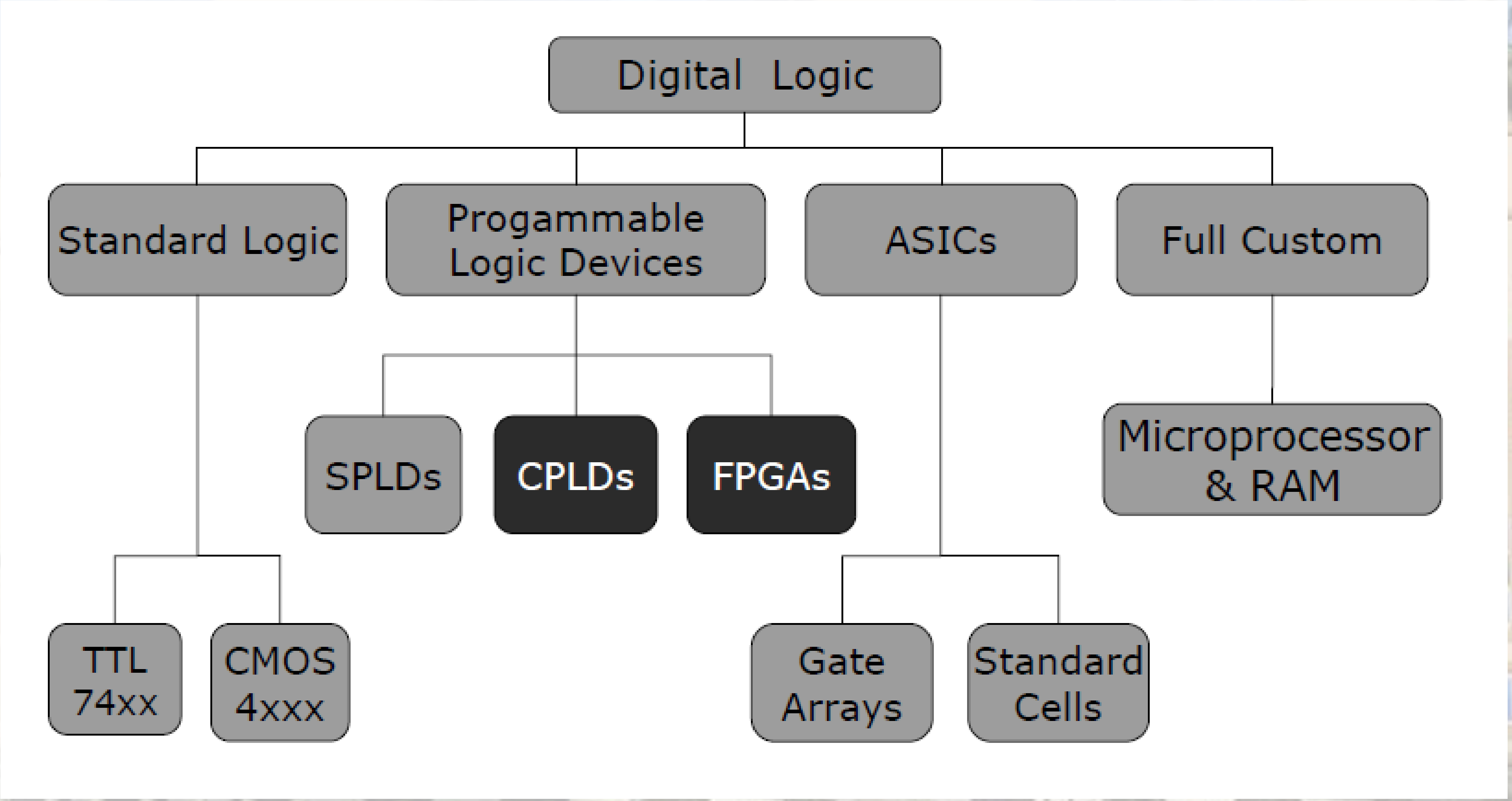




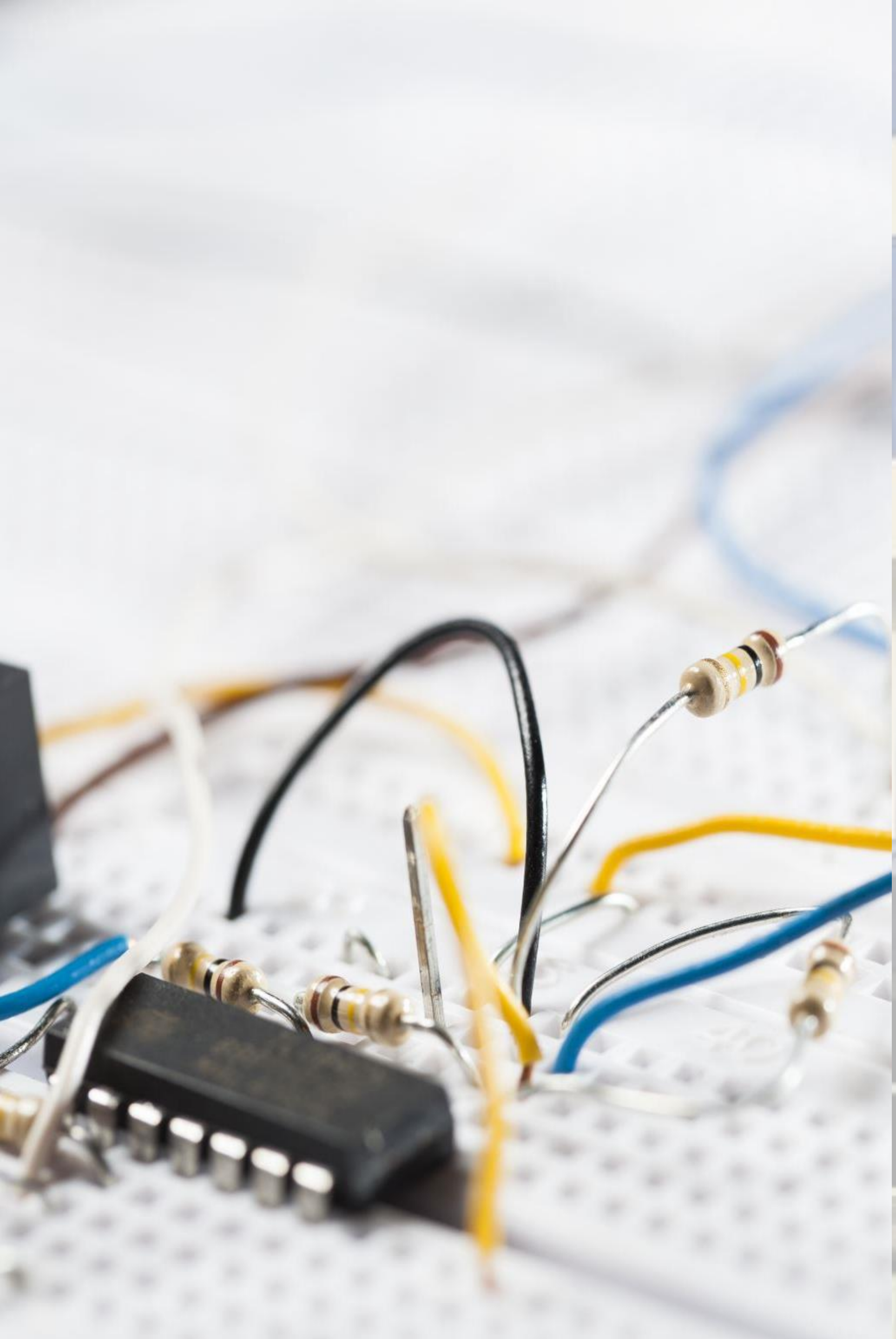
Types of Actuators

1. Electrical actuators
   - Electric motors
     - DC servomotors
     - AC motors
     - Stepper motors
   - Solenoids
1. Hydraulic actuators
   - Use hydraulic fluid to amplify the controlle command signal
1. Pneumatic actuators
   - Use compressed air as the driving force

Digital Logic Technologies



Programmable Logic Devices



- A programmable logic device (PLD) is an electronic component used to build reconfigurable digital circuits

- Unlike integrated circuits (IC) which consist of logic gates and have a fixed function, a PLD has an undefined function at the time of manufacture.

- Before the PLD can be used in a circuit it must be programmed (reconfigured) by using a specialized program.

In the beginning, there was...

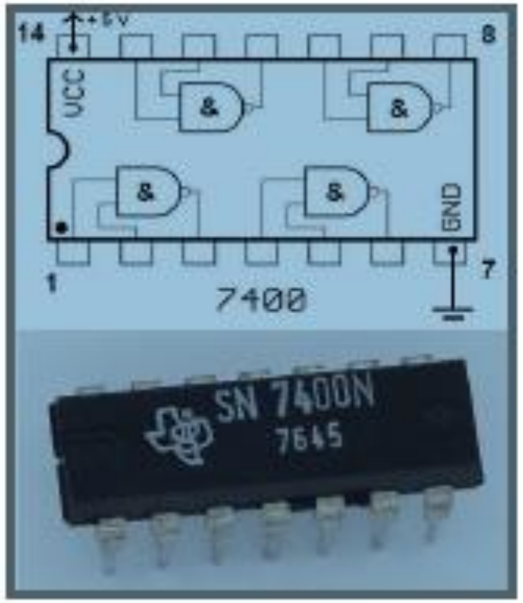
Standard 74`series of transistor`transistor logic (TTL) integrated circuits. (mid 1960s)

Hundreds of devices, ready for use, that provide all from basic logic gates (and,nand etc.), flip`flops, counters to arithme5c logic units (ALU).

“Programmability”: using wires to connect different devices (e.g. PCB, but not reprogrammable)

Content of device can not be changed

Today typically used as glue logic in computers and industrial electronics.



Programmable Logic Devices (cont’d)

- Comes with an undefined function but with the potential to be programmed
- Wanted function typically obtained by breaking or closing connections (e.g. fuse / anti-fuse)
- Various technologies available (1970s)

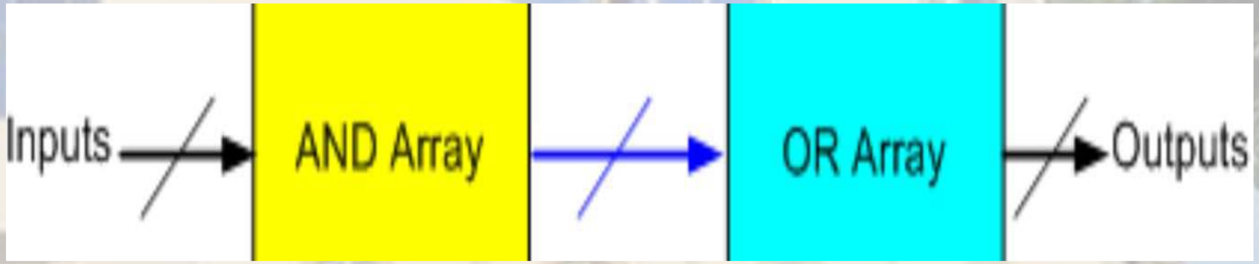
– PLA: Programmable Logic Array (Programmable AND & OR gate array)

– PAL: Programmable Array Logic (Programmable AND but fixed OR => Faster & cheaper than PLA)

– GAL: Generic Array Logic (PAL + output macrocell w/ e.g. flip flop and mux)

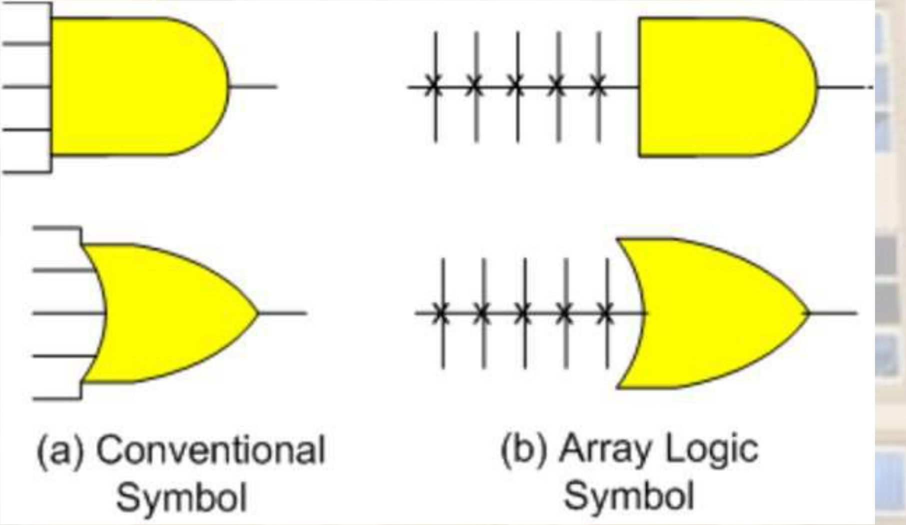
Programmable Logic Devices (cont'd)

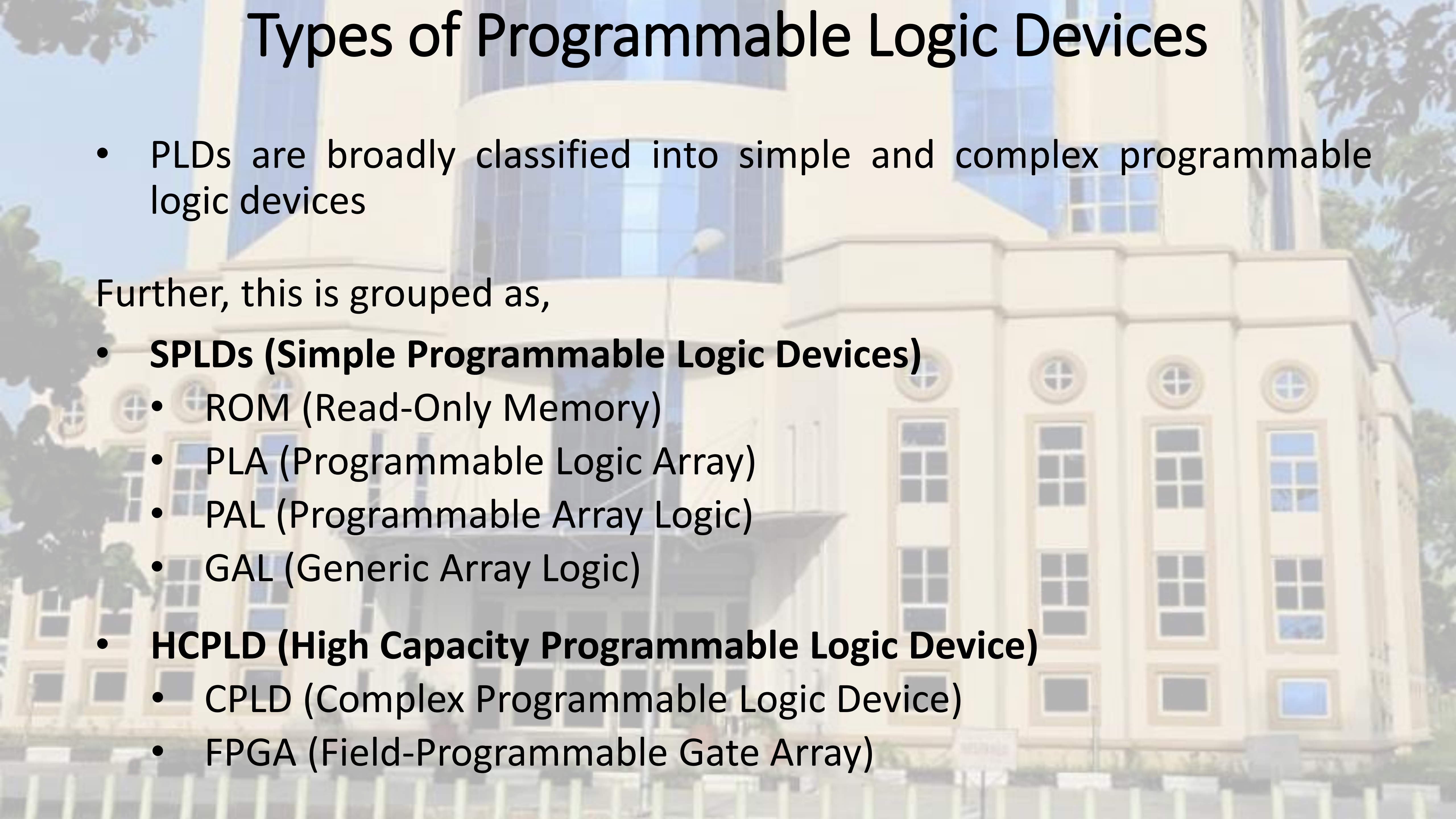
- PLDs are typically built with an array of AND gates (AND-array) and an array of OR gates (OR-array)



Advantages of using PLDs

- Less board space, faster, lower power requirements (i.e., smaller power supplies), less costly assembly processes, higher reliability (fewer ICs and circuit connections means easier troubleshooting), and availability of design software





Types of Programmable Logic Devices

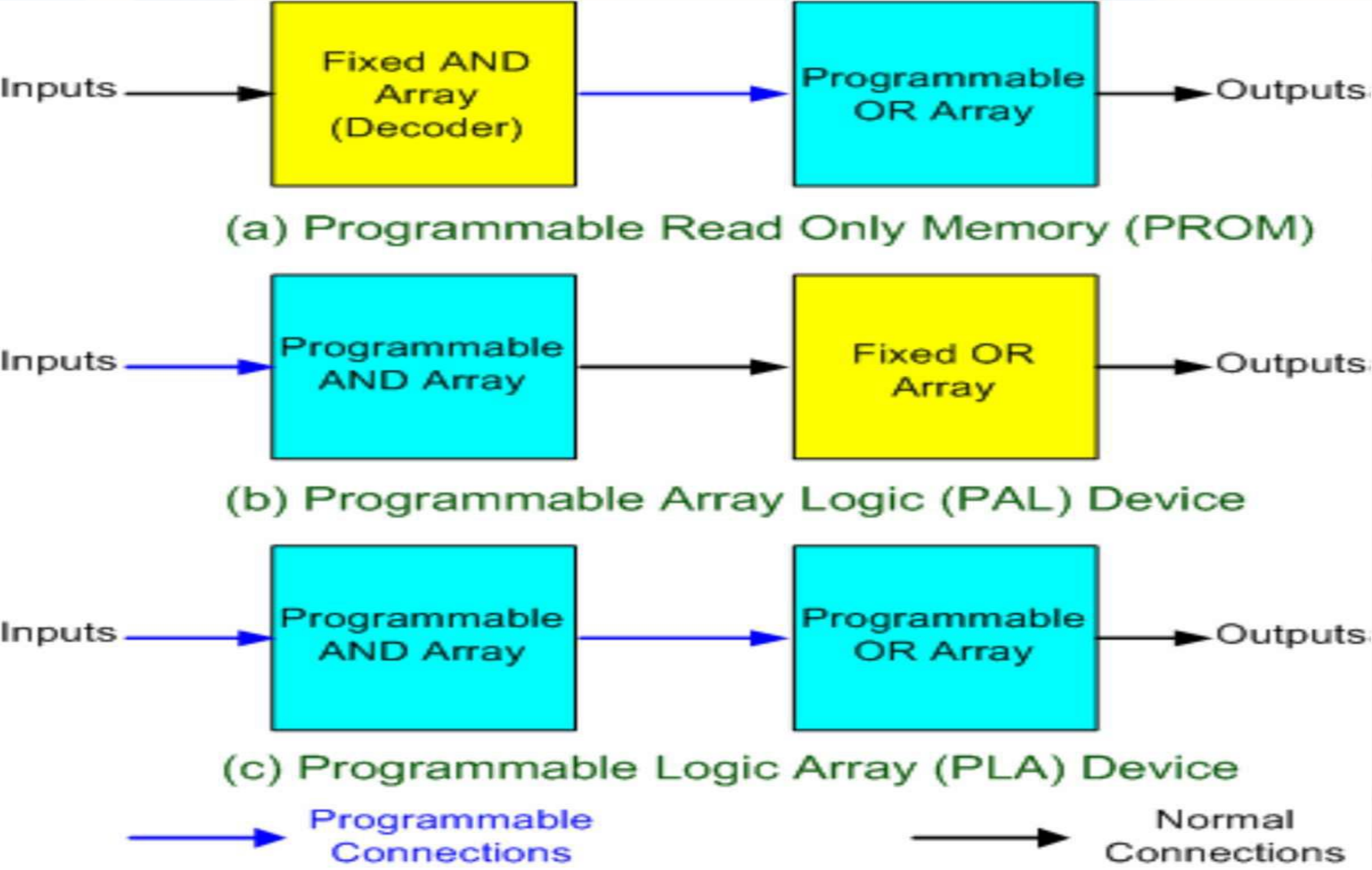
- PLDs are broadly classified into simple and complex programmable logic devices

Further, this is grouped as,

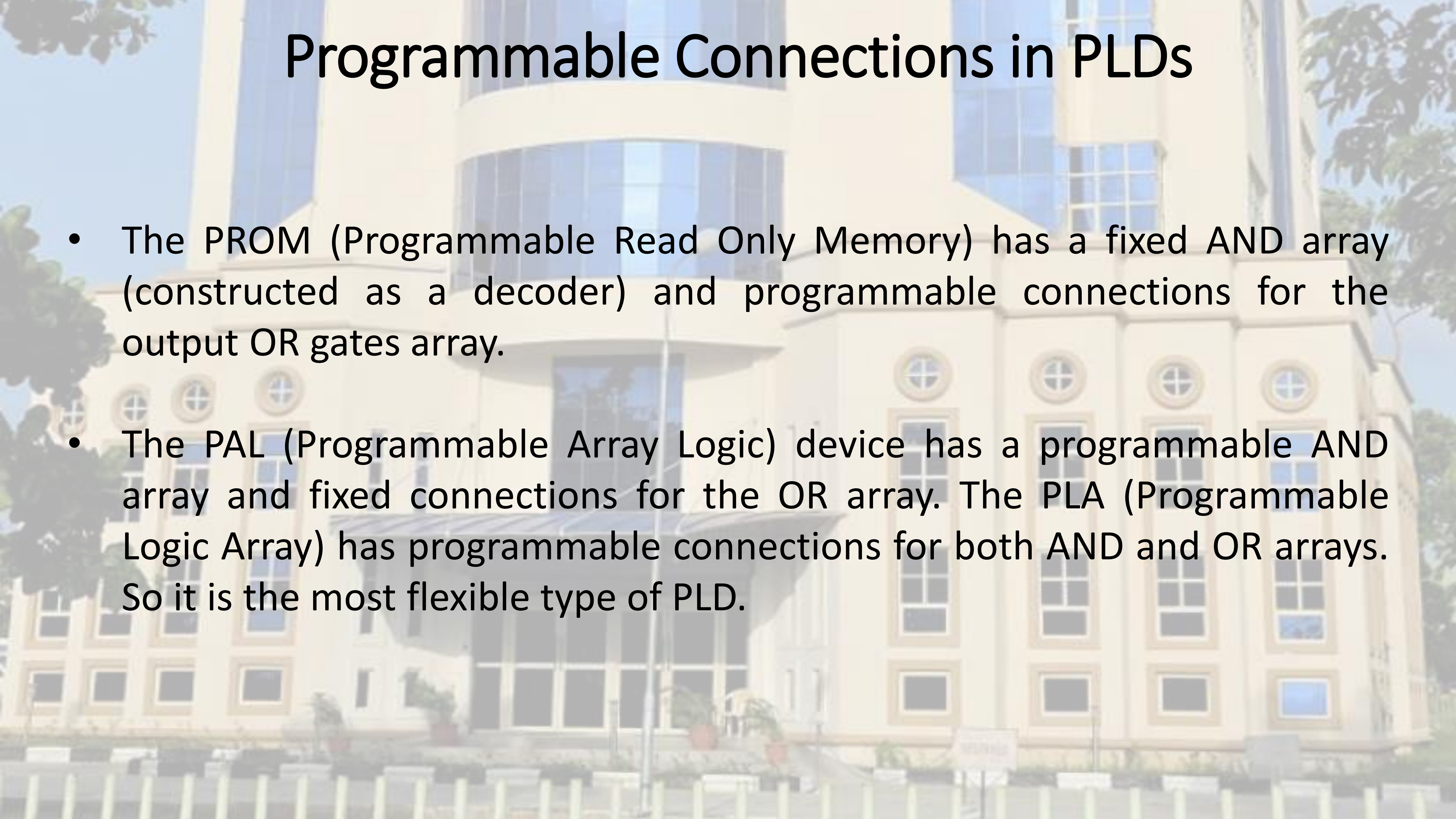
- **SPLDs (Simple Programmable Logic Devices)**
  - ROM (Read-Only Memory)
  - PLA (Programmable Logic Array)
  - PAL (Programmable Array Logic)
  - GAL (Generic Array Logic)

- **HCPLD (High Capacity Programmable Logic Device)**
  - CPLD (Complex Programmable Logic Device)
  - FPGA (Field-Programmable Gate Array)

Programmable Connections in PLDs



Programmable Connections in PLDs



- The PROM (Programmable Read Only Memory) has a fixed AND array (constructed as a decoder) and programmable connections for the output OR gates array.

- The PAL (Programmable Array Logic) device has a programmable AND array and fixed connections for the OR array. The PLA (Programmable Logic Array) has programmable connections for both AND and OR arrays. So it is the most flexible type of PLD.

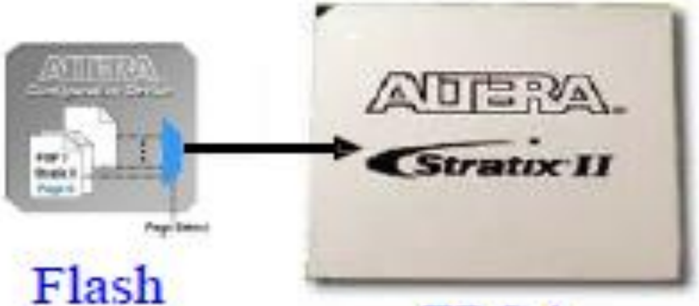
Applications of Programmable Logic Devices

- **Glue Logic** – to integrate and interconnect different digital components
- **State Machines** – for precise control of operations, based on input
- **Counters** – implement counters, which track and count specific events or clock cycles in digital systems
- **Synchronization** – assist in synchronization by aligning signals from different clock domains
- **Decoders** – converts coded inputs into specific output signals
- **Bus Interfaces** – used in bus interfaces to manage data communication between multiple components
- **Parallel-to-Serial** – converts parallel data streams into serial formats, enabling efficient transmission of data over communication links with fewer wire
- **Serial-to-Parallel** – handles serial-to-parallel data conversion

Programmable logic

- FPGA
- CPLD

CPLD



Flash

FPGA

- Circuit manufacturers:
  - **Altera** (www.altera.com)
  - **Xilinx** (www.xilinx.com)

ALTERA

XILINX

ATMEL

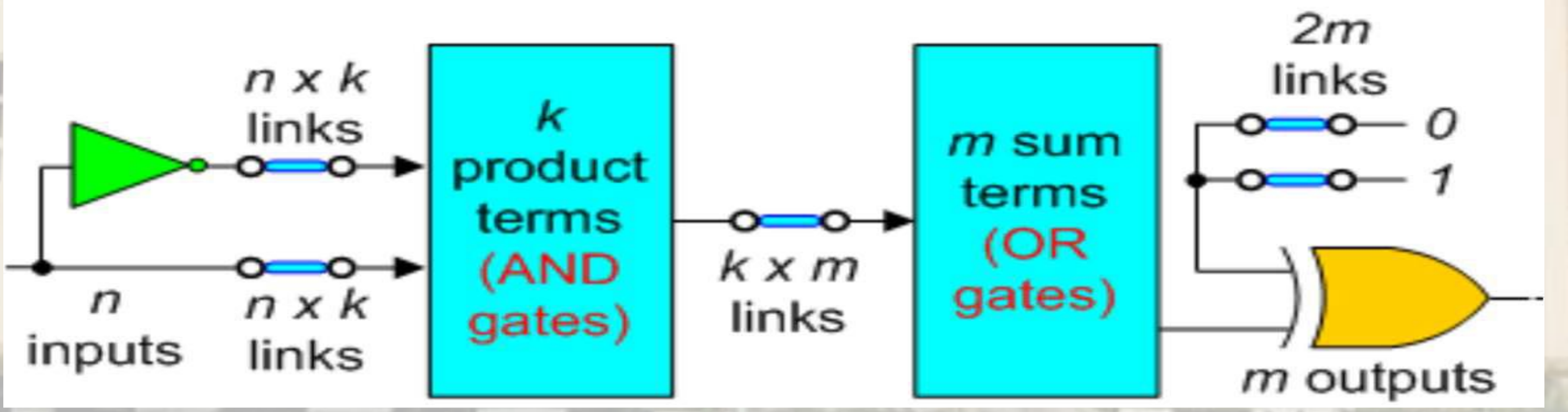
Actel
POWER MATTERS

Lattice
Semiconductor Corporation
Bringing the Best Together

- Tools:
  - **QUARTUS II (Altera)**
  - **Modelsim Altera (Mentor VHDL simulator)**
  - **NIOS II Embedded design suite** (+ C2H compiler)
  - DSP builder (Altera)
  - ISE (Xilinx)

Programmable Logic Array (PLA)

- Each of the AND gates can be programmed to generate a product term of the input variables
- It consists of n inputs, m outputs, and k product terms. The product terms constitute a group of k AND gates each of 2n inputs. Links are inserted between all n inputs and their complement values to each of the AND gates. Links are also provided between the outputs of the AND gates and the inputs of the OR gates *(see block diagram)*



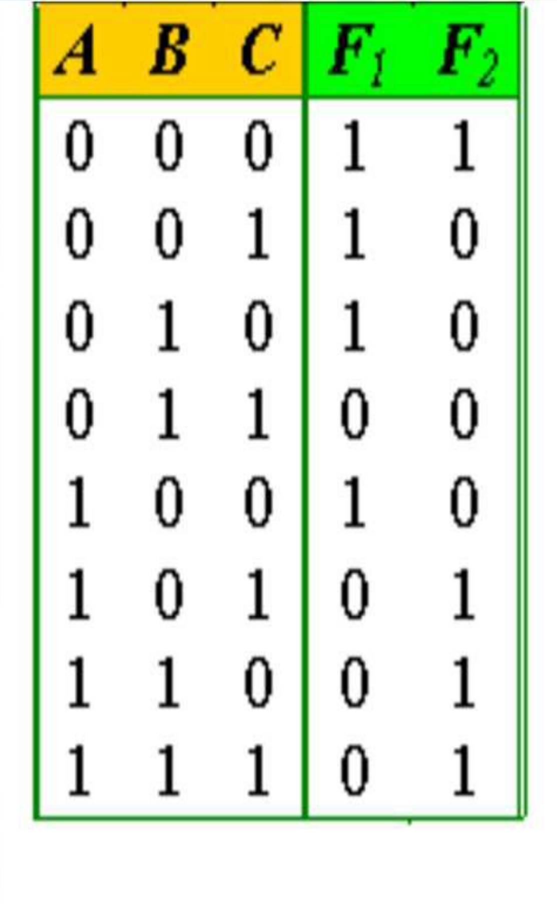
Programmable Logic Array (PLA)

- Since PLA has m-outputs, the number of OR gates is m. The output of each OR gate goes to an XOR gate, where the other input has two sets of links, one connected to logic 0 and the other to logic 1. It allows the output function to be generated either in the true form or in the complement form.
- The output is inverted when the XOR input is connected to 1 (since $X \oplus 1 = X$). The output does not change when the XOR input is connected to 0 (since $X \oplus 0 = X$).
- The size of the PLA is specified by the number of inputs (n), the number of product terms (k), and the number of outputs (m), (the number of sum terms is equal to the number of outputs).

Programmable Logic Array (PLA) (cont'd)

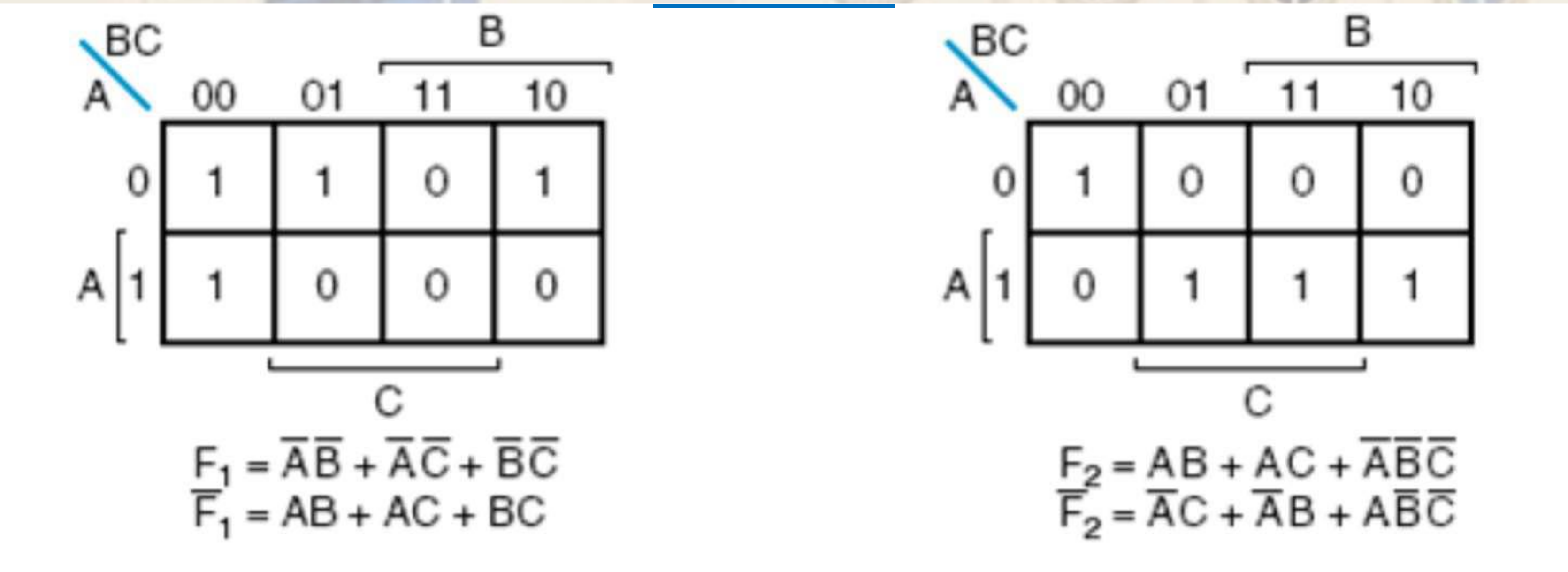
- Example: Implement the combinational circuit having the shown truth table, using PLA.

***Hint:** Each product term in the expression requires an AND gate. To minimize the cost, it is necessary to simplify the function to a minimum number of product terms.*



A	B	C	F_1	F_2
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Solution



A \ BC	00	01	11	10
0	1	1	0	1
1	1	0	0	0

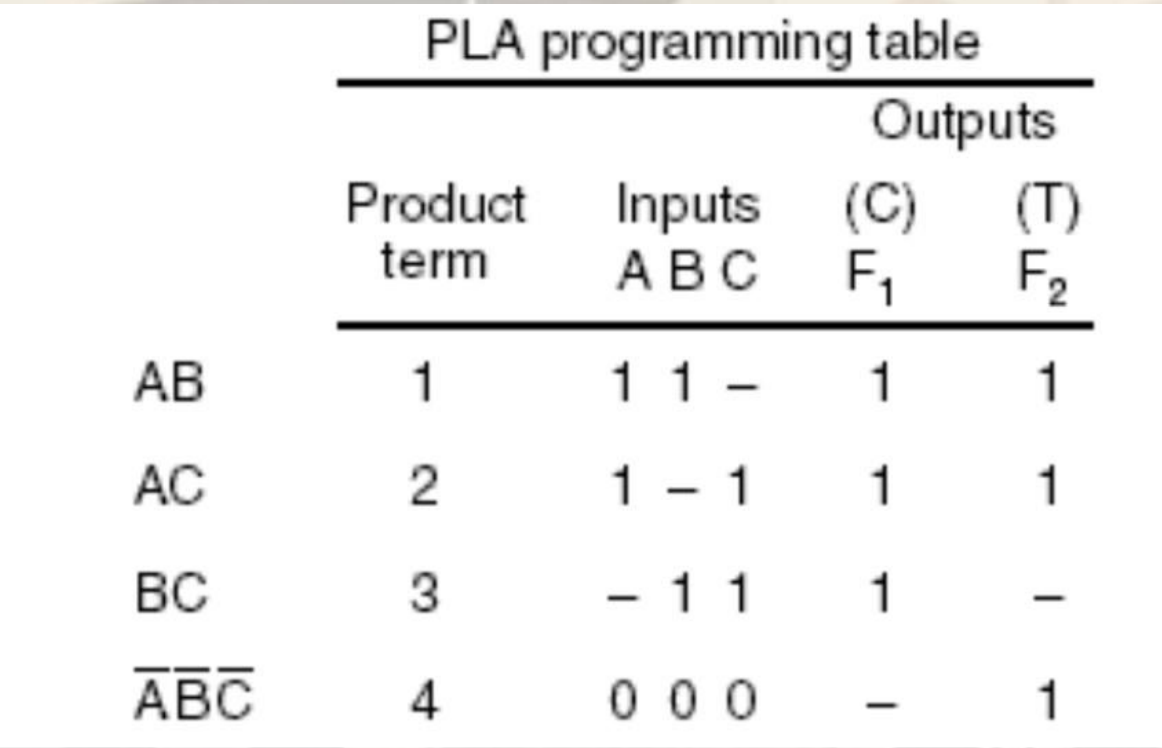
$$F_1 = \bar{A}\bar{B} + \bar{A}\bar{C} + \bar{B}\bar{C}$$
$$\overline{F_1} = AB + AC + BC$$

A \ BC	00	01	11	10
0	1	0	0	0
1	0	1	1	1

$$F_2 = AB + AC + \bar{A}\bar{B}\bar{C}$$
$$\overline{F_2} = \bar{A}C + \bar{A}B + A\bar{B}\bar{C}$$

Programmable Logic Array (PLA) (cont’d)

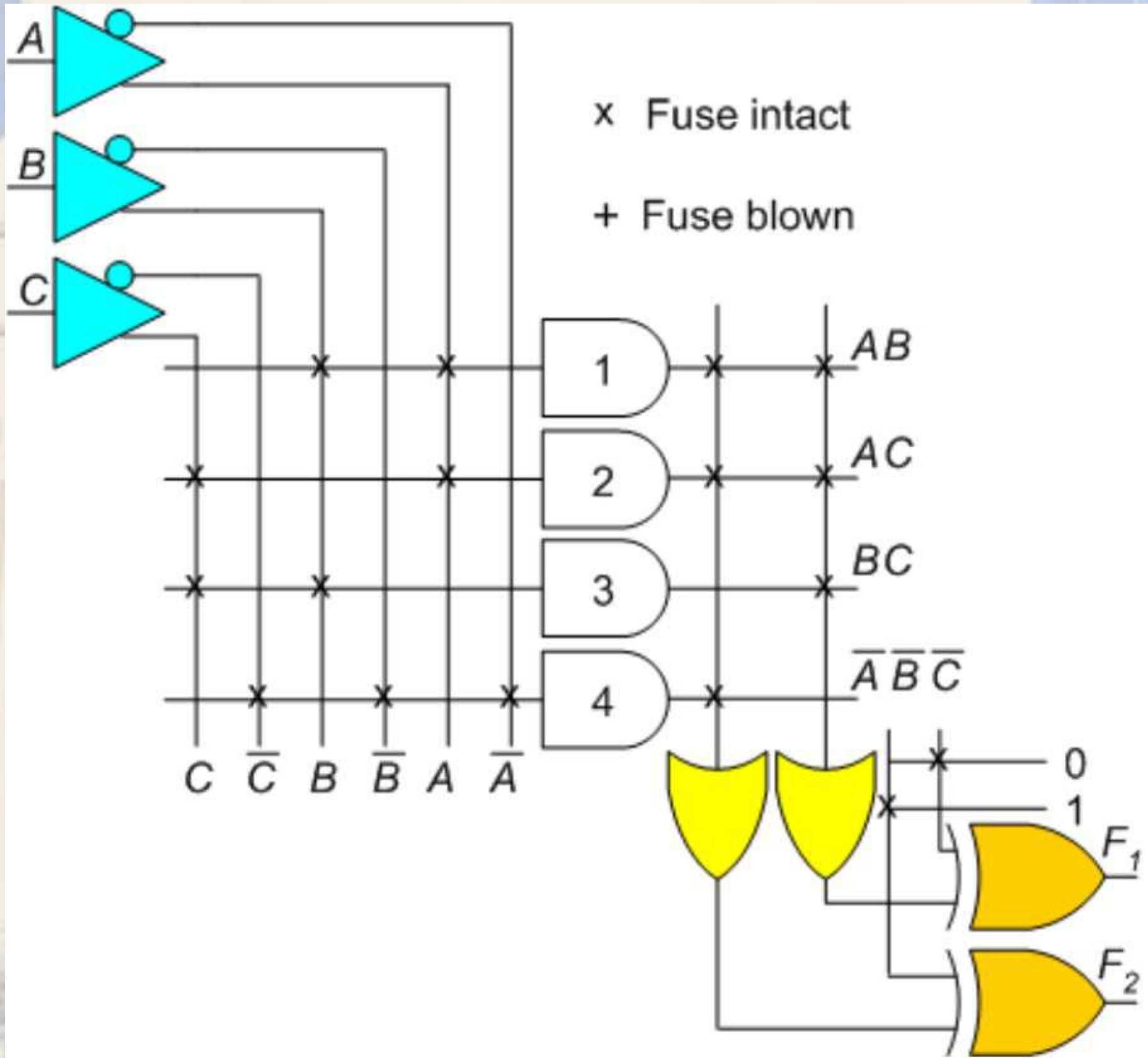
- The combination that gives a minimum number of product terms is,
- **F1 = AB + AC + BC or F1 = (AB + AC + BC)’ ; F2 = AB + AC + A’B’C’**
- This gives only 4 distinct product terms: AB, AC, BC, and A’B’C’. So the PLA table will be as follows,



PLA programming table

	Product term	Inputs			Outputs	
		A	B	C	(C) F_1	(T) F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$\overline{A}\overline{B}\overline{C}$	4	0	0	0	–	1

Programmable Logic Array (PLA) (cont’d)

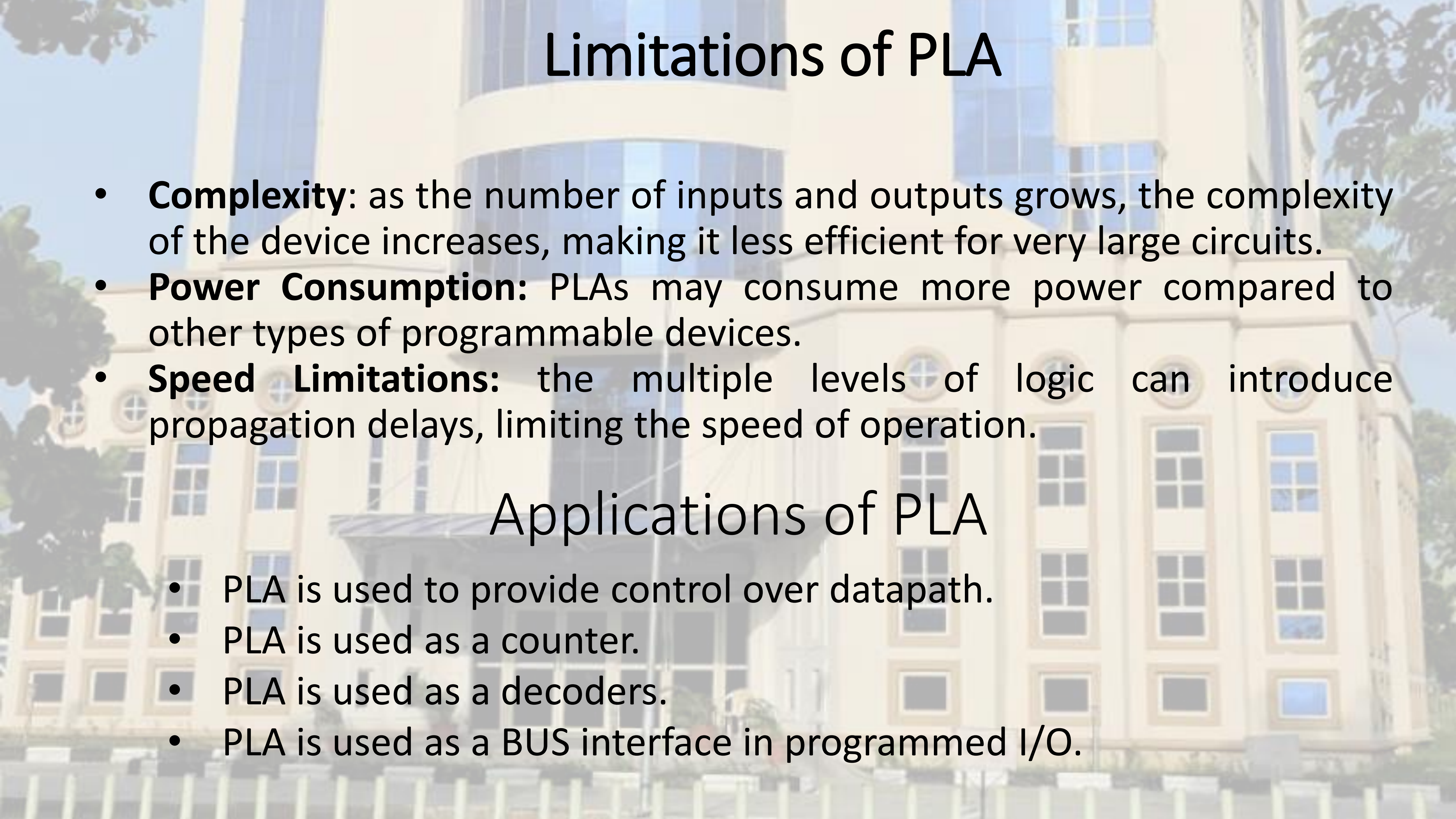

A
B
C
× Fuse intact
+ Fuse blown
1
2
3
4
AB
AC
BC
$\overline{A}\,\overline{B}\,\overline{C}$
C $\overline{C}$ B $\overline{B}$ A $\overline{A}$
0
1
F_1
F_2

Advantages of PLA

- Flexibility: Both the programmable AND gates and also the programmable OR gates offer the beneficial benefits of being used in a number of combinational circuits.

- Efficient Minterm Usage: It will only implement the necessary minterms thus making the design compact and more efficient.

- Reconfigurability: The PLAs can be reconfigured to perform different logic functions depending with the requirements of an application.

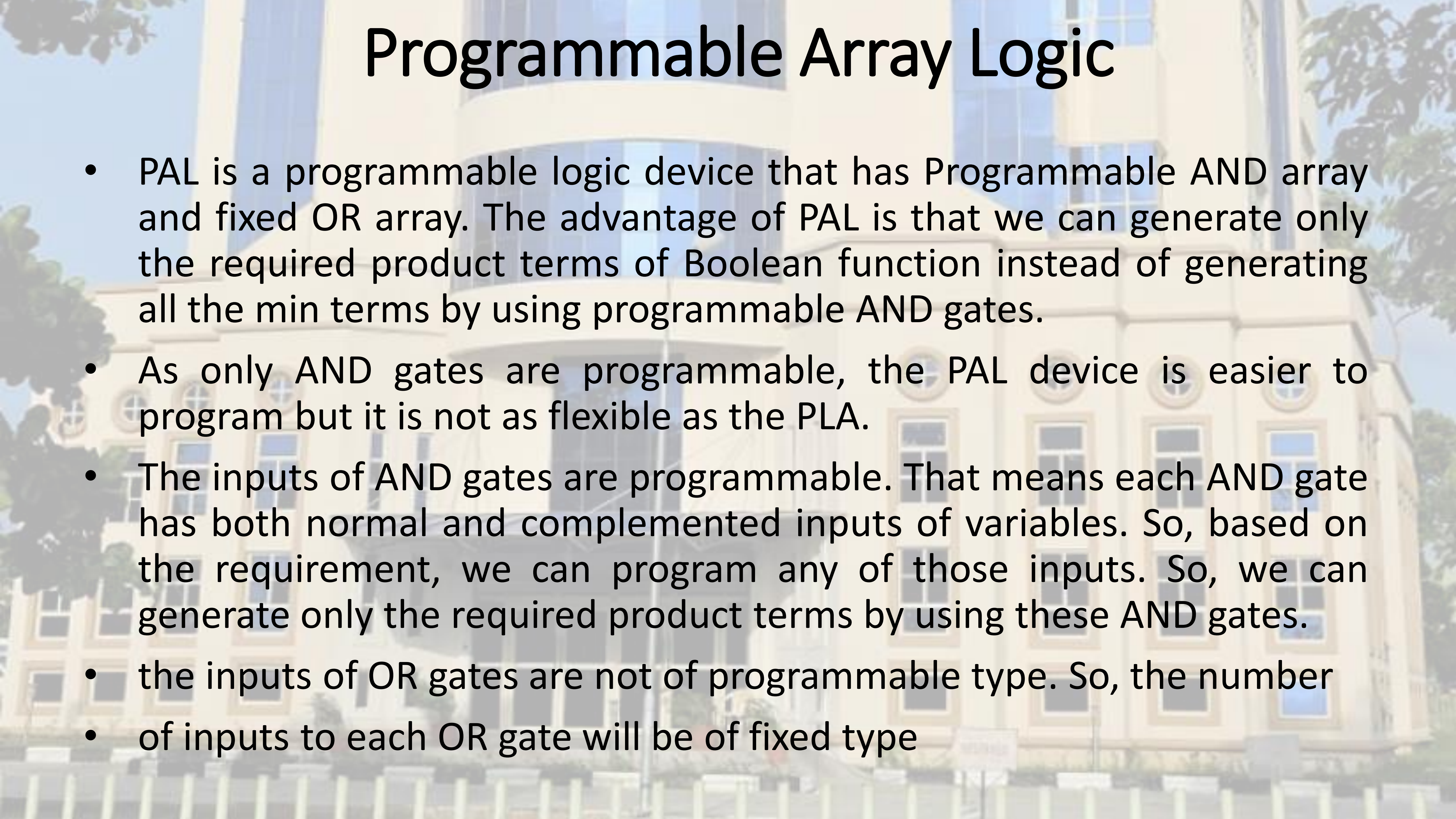


Limitations of PLA

- **Complexity**: as the number of inputs and outputs grows, the complexity of the device increases, making it less efficient for very large circuits.
- **Power Consumption:** PLAs may consume more power compared to other types of programmable devices.
- **Speed Limitations:** the multiple levels of logic can introduce propagation delays, limiting the speed of operation.

Applications of PLA

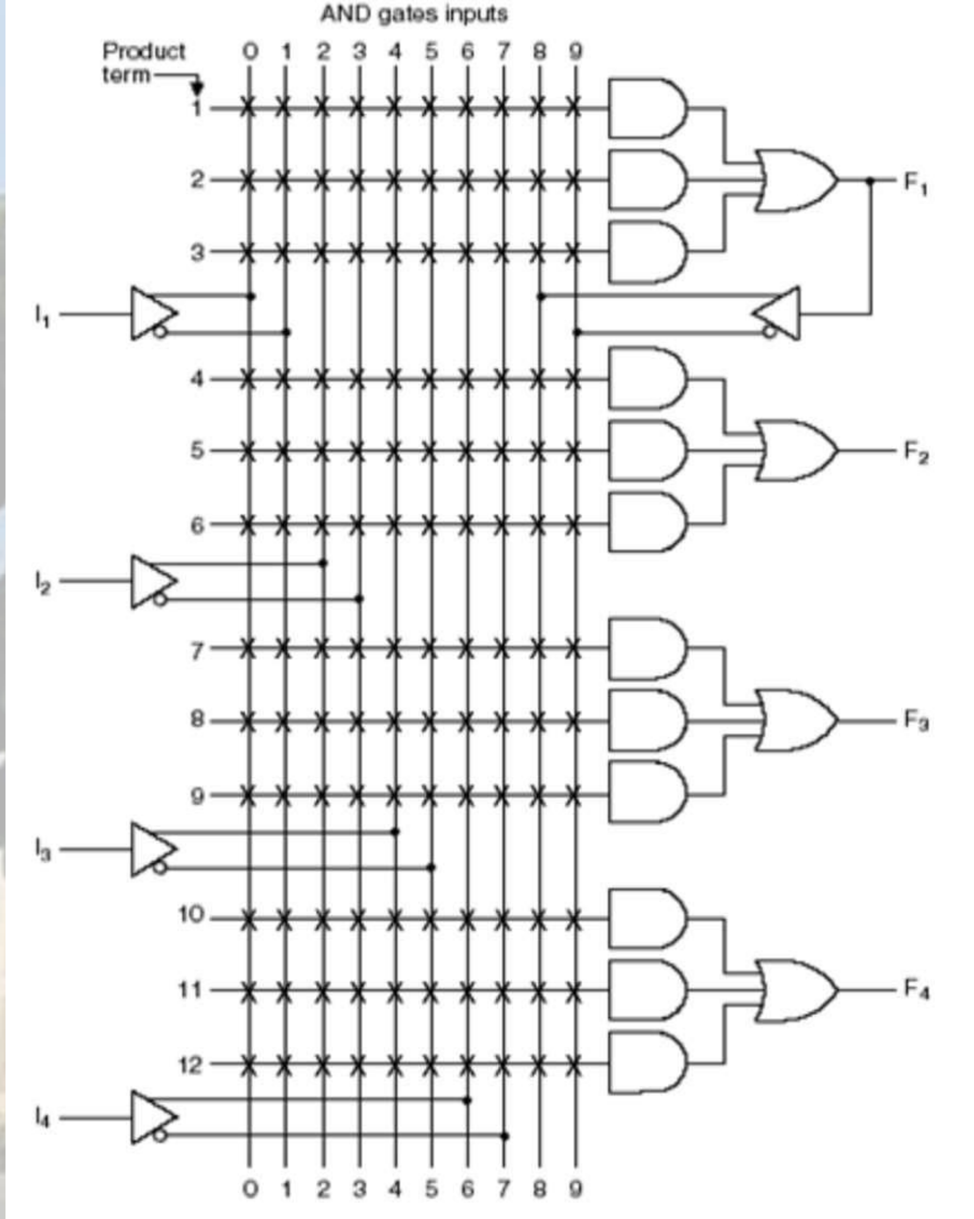
- PLA is used to provide control over datapath.
- PLA is used as a counter.
- PLA is used as a decoders.
- PLA is used as a BUS interface in programmed I/O.



Programmable Array Logic

- PAL is a programmable logic device that has Programmable AND array and fixed OR array. The advantage of PAL is that we can generate only the required product terms of Boolean function instead of generating all the min terms by using programmable AND gates.
- As only AND gates are programmable, the PAL device is easier to program but it is not as flexible as the PLA.
- The inputs of AND gates are programmable. That means each AND gate has both normal and complemented inputs of variables. So, based on the requirement, we can program any of those inputs. So, we can generate only the required product terms by using these AND gates.
- the inputs of OR gates are not of programmable type. So, the number
- of inputs to each OR gate will be of fixed type

Programmable Array Logic (cont'd)



The device shown in the figure has 4 inputs and 4 outputs. Each input has a buffer-inverter gate, and each output is generated by a fixed OR gate.

The device has 4 sections, each composed of a 3-wide AND-OR array, meaning that there are 3 programmable AND gates in each section.

Each AND gate has 10 programmable input connections indicating by 10 vertical lines intersecting each horizontal line. The horizontal line symbolizes the multiple input configuration of an AND gate. One of the outputs F1 is connected to a buffer-inverter gate and is fed back into the inputs of the AND gates through programmed connections.

Programmable Array Logic (cont’d)

- Example: Implement the following Boolean functions using the PAL device as shown above,

$$W(A, B, C, D) = \Sigma m(2, 12, 13)$$

$$X(A, B, C, D) = \Sigma m(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

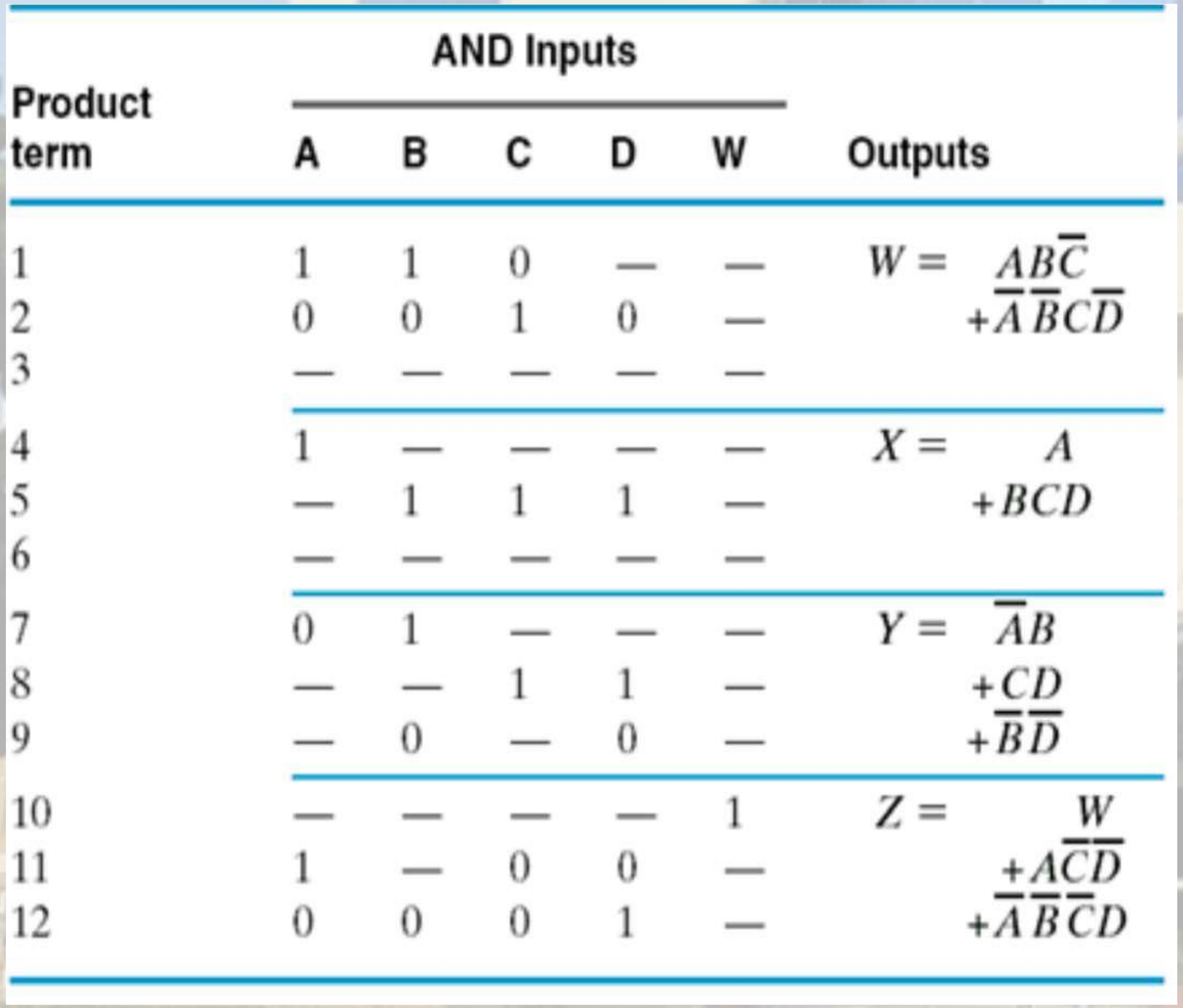
$$Y(A, B, C, D) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$Z(A, B, C, D) = \Sigma m(1, 2, 8, 12, 13)$$

SOLUTION

- *Simplifying the 4 functions to a minimum number of terms results in the following Boolean functions*:
- $W = ABC' + A'B'CD'$ $X = A + BCD$ $Y = A'B + CD + B'D'$
- $Z = ABC' + A'B'CD + AC'D' + A'B'C'D = W + AC'D' + A'B'C'D$

Programmable Array Logic (cont'd)

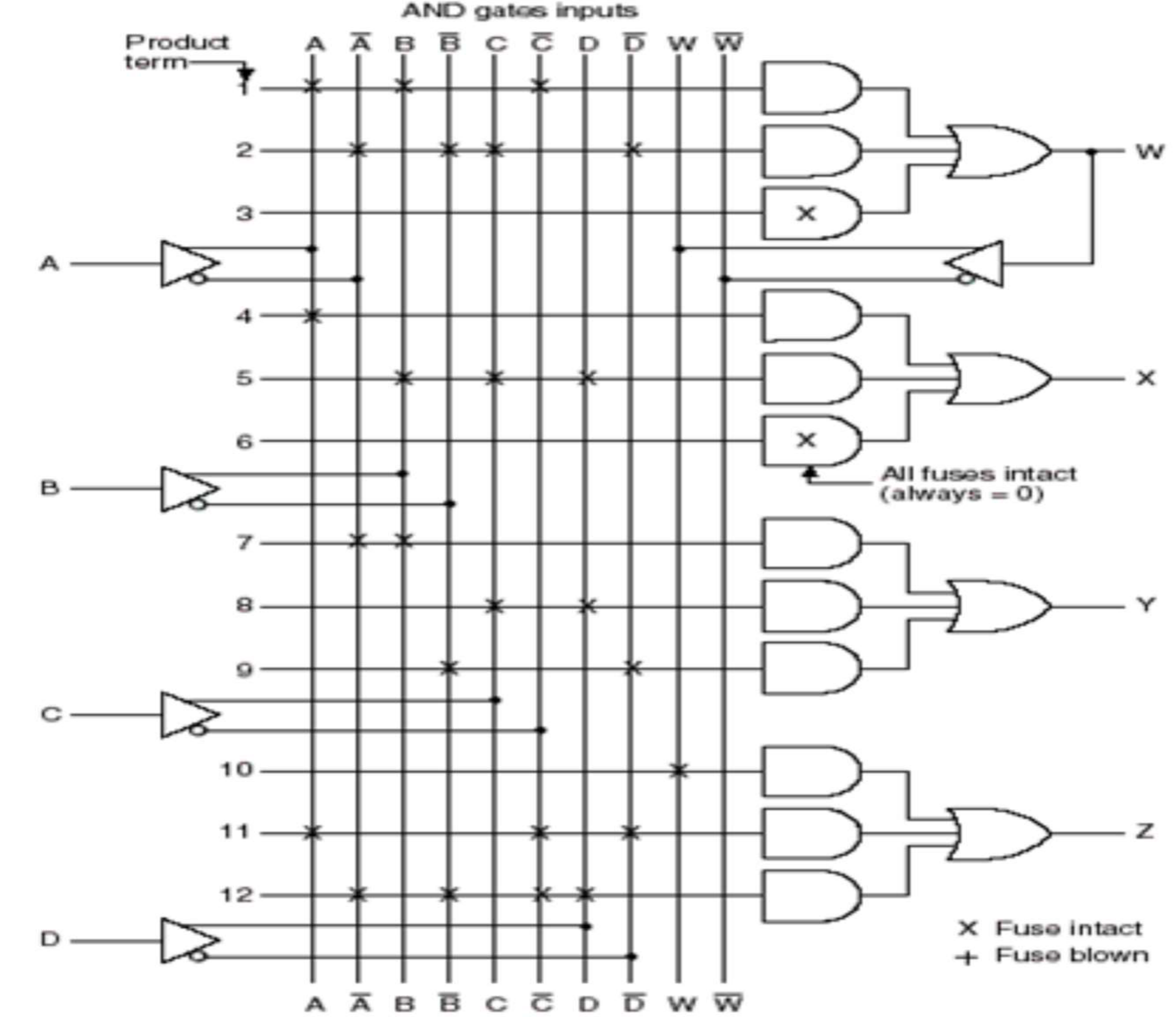


	AND Inputs					
Product term	A	B	C	D	W	Outputs
1	1	1	0	—	—	$W = AB\bar{C}$
2	0	0	1	0	—	$+\bar{A}\bar{B}C\bar{D}$
3	—	—	—	—	—	
4	1	—	—	—	—	$X = A$
5	—	1	1	1	—	$+BCD$
6	—	—	—	—	—	
7	0	1	—	—	—	$Y = \bar{A}B$
8	—	—	1	1	—	$+CD$
9	—	0	—	0	—	$+\bar{B}\bar{D}$
10	—	—	—	—	1	$Z = W$
11	1	—	0	0	—	$+A\bar{C}\bar{D}$
12	0	0	0	1	—	$+\bar{A}\bar{B}\bar{C}D$

Note that the function for Z has four product terms. The logical sum of two of these terms is equal to W. Thus, by using W, it is possible to reduce the number of terms for Z from four to three, so that the function can fit into the given PAL device

The PAL programming table is similar to the table used for the PLA, except that only the inputs of the AND gates need to be programmed.

Programmable Array Logic



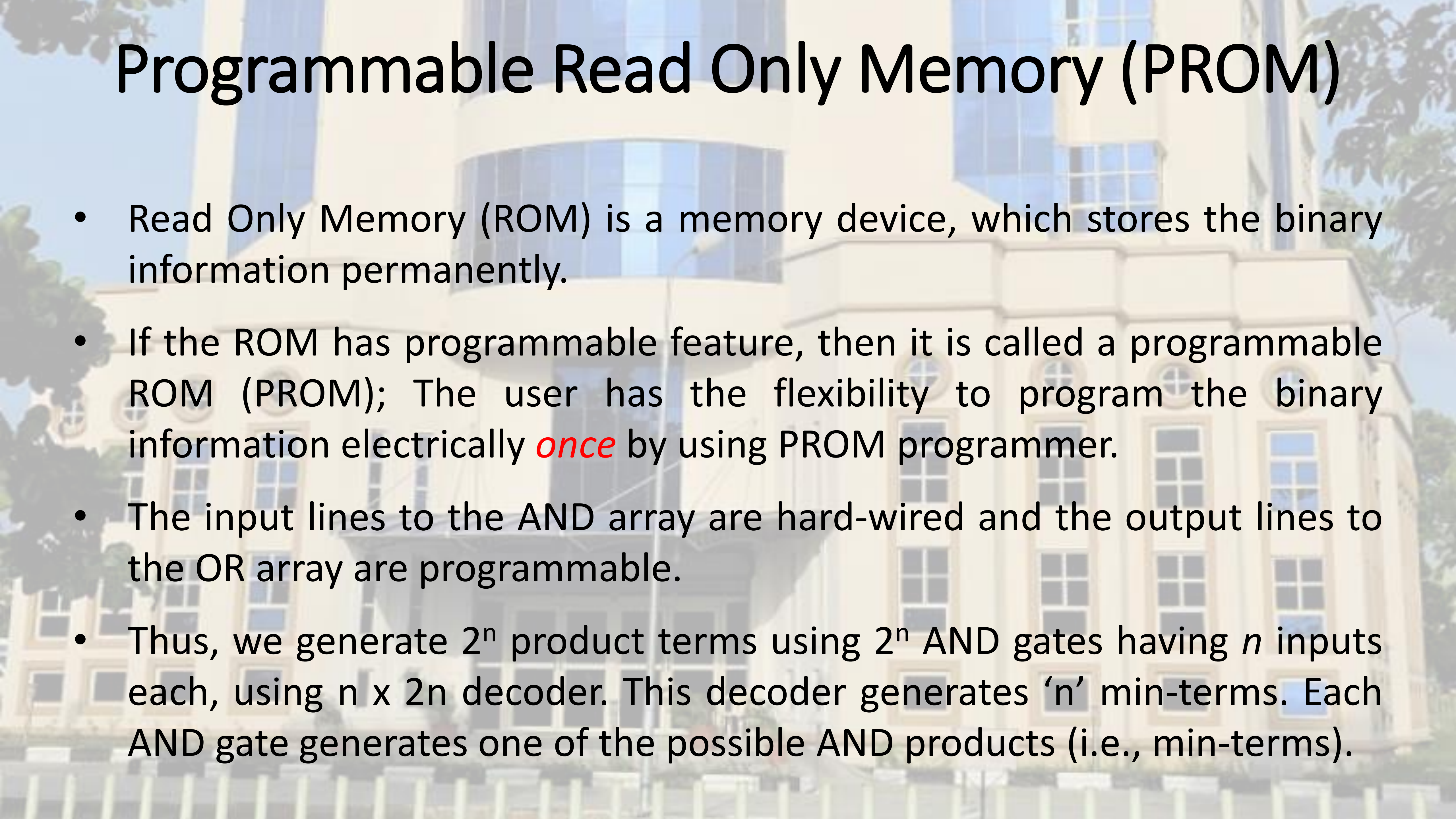
PLAs Vs PALs

- PALs have the same limitations as PLAs (small number of allowed AND terms) plus they have a fixed OR plane i.e., less flexibility than PLAs

- PALs are simpler to manufacture, cheaper, and faster (better performance)

- PALs also often have extra circuitry connected to the output of each OR gate

>>>> The OR gate plus this circuitry is called a macro-cell



Programmable Read Only Memory (PROM)

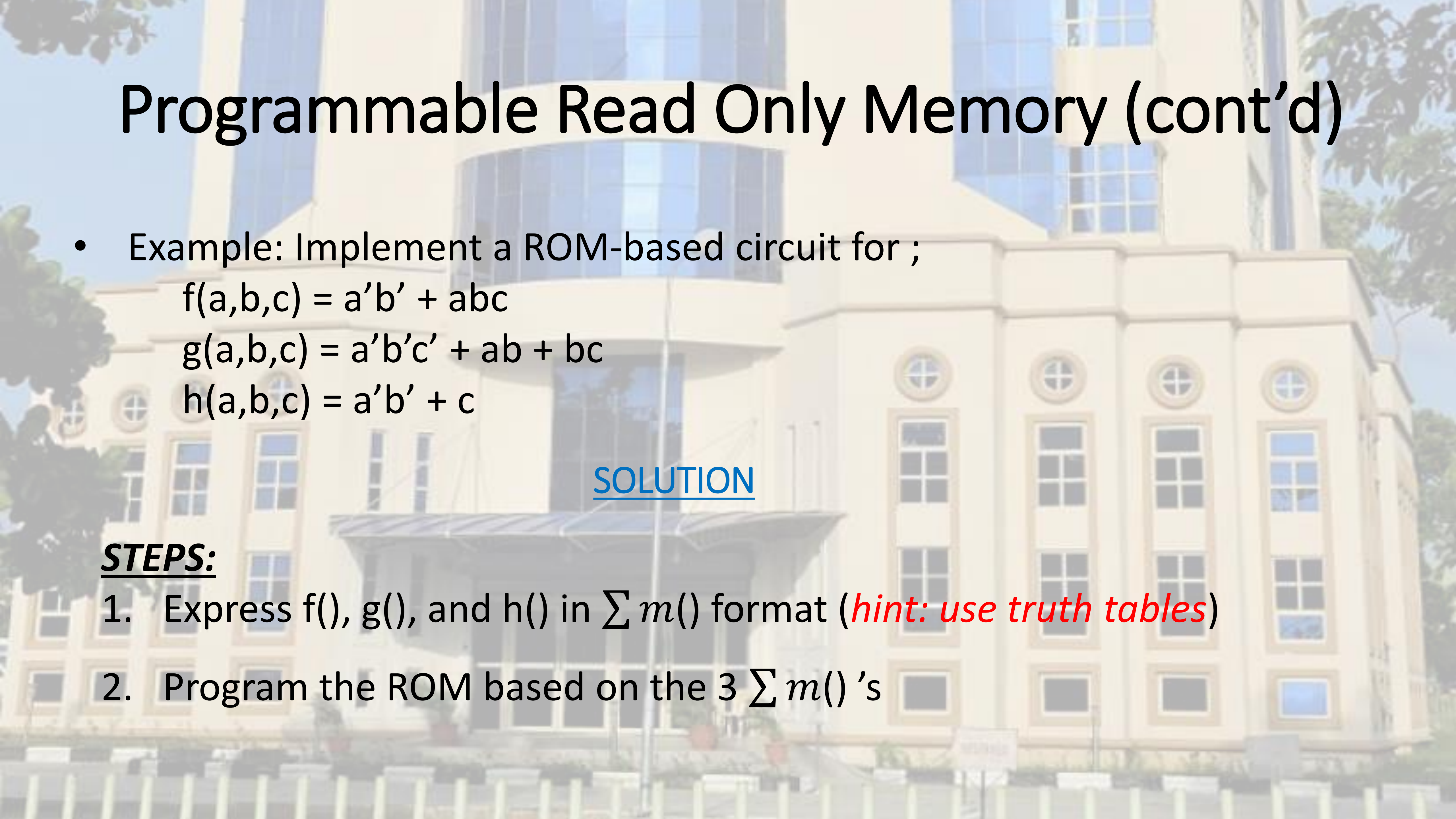
- Read Only Memory (ROM) is a memory device, which stores the binary information permanently.
- If the ROM has programmable feature, then it is called a programmable ROM (PROM); The user has the flexibility to program the binary information electrically *once* by using PROM programmer.
- The input lines to the AND array are hard-wired and the output lines to the OR array are programmable.
- Thus, we generate 2^n product terms using 2^n AND gates having *n* inputs each, using n x 2n decoder. This decoder generates ‘n’ min-terms. Each AND gate generates one of the possible AND products (i.e., min-terms).

Programmable Read Only Memory (cont’d)

- Given a 2^k x n ROM, we can implement ANY combinational circuit with at most *k* inputs and at most *n* outputs. Because;
  - ➢ k-to-2^k decoder will generate all 2^k possible min-terms
  - ➢ Each of the OR gates must implement a $\sum m()$
  - ➢ Each $\sum m()$ can be programmed

Programmable Read Only Memory (cont’d)

- Given a 2^k x n ROM, we can implement ANY combinational circuit with at most *k* inputs and at most *n* outputs. Because;
  - k-to-2^k decoder will generate all 2^k possible min-terms
  - Each of the OR gates must implement a $\sum m()$
  - Each $\sum m()$ can be programmed



Programmable Read Only Memory (cont’d)

- Example: Implement a ROM-based circuit for ;

  f(a,b,c) = a’b’ + abc

  g(a,b,c) = a’b’c’ + ab + bc

  h(a,b,c) = a’b’ + c

SOLUTION

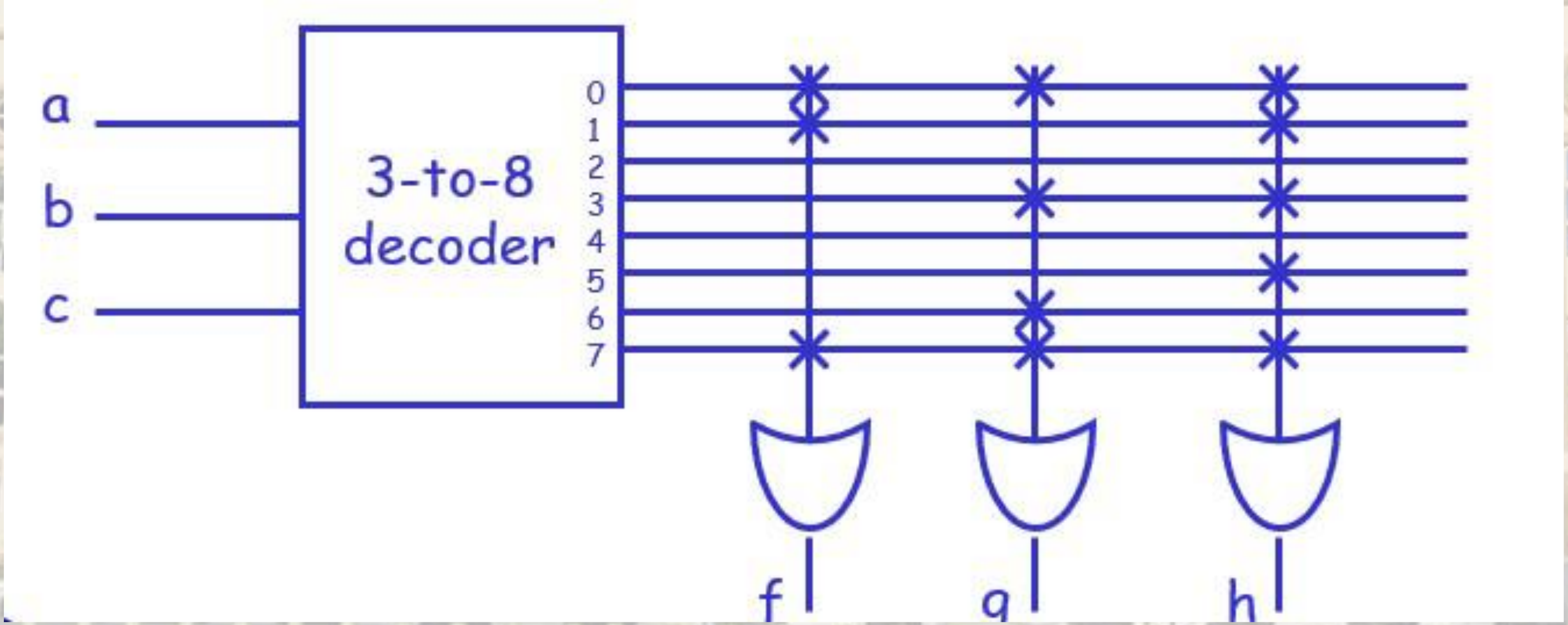
STEPS:

1. Express f(), g(), and h() in $\sum m()$ format (*hint: use truth tables*)
2. Program the ROM based on the 3 $\sum m()$ ’s

Programmable Read Only Memory (cont’d)

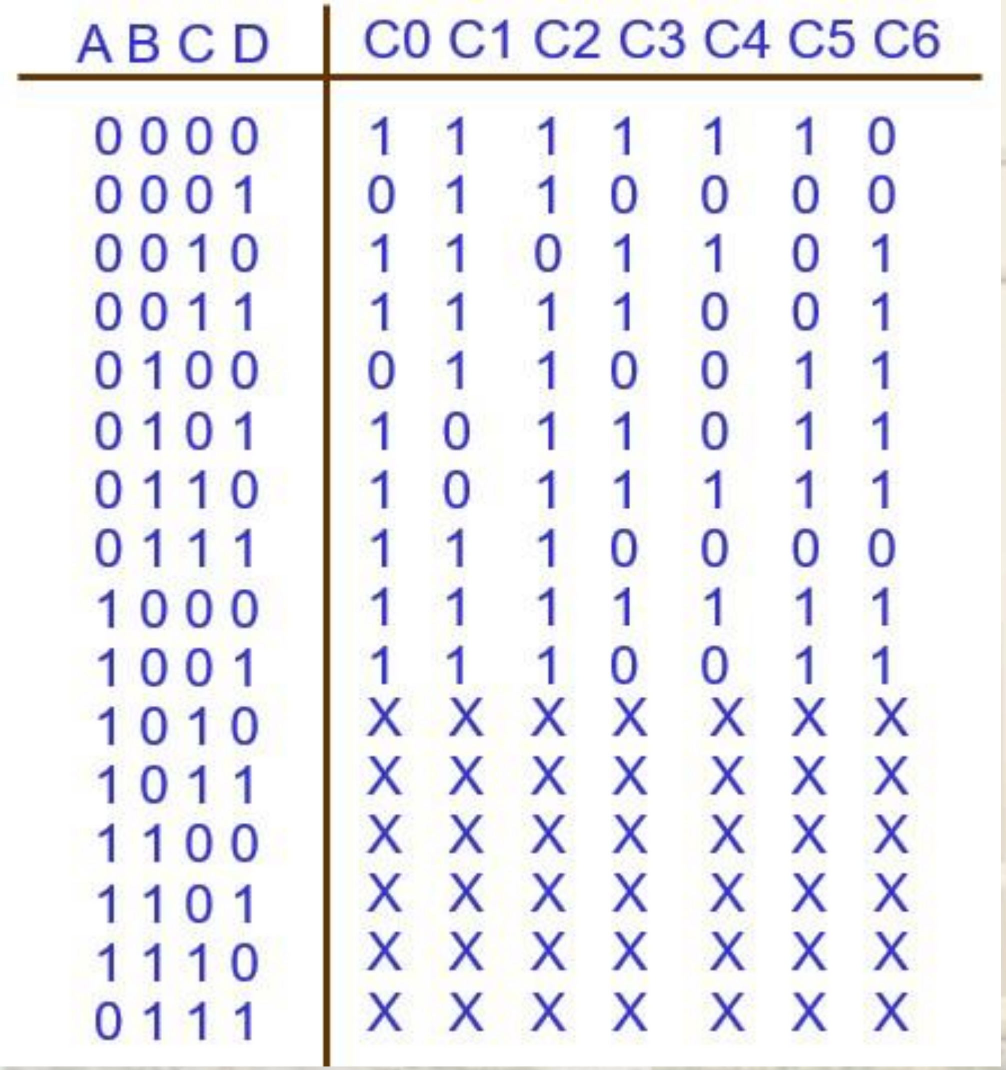
>>>> There are 3 inputs and 3 outputs, thus we need a 8x3 ROM block

$f = \sum m(0, 1, 7)$ $g = \sum m(0, 3, 6, 7)$ and $= \sum m(0, 1, 3, 5, 7)$

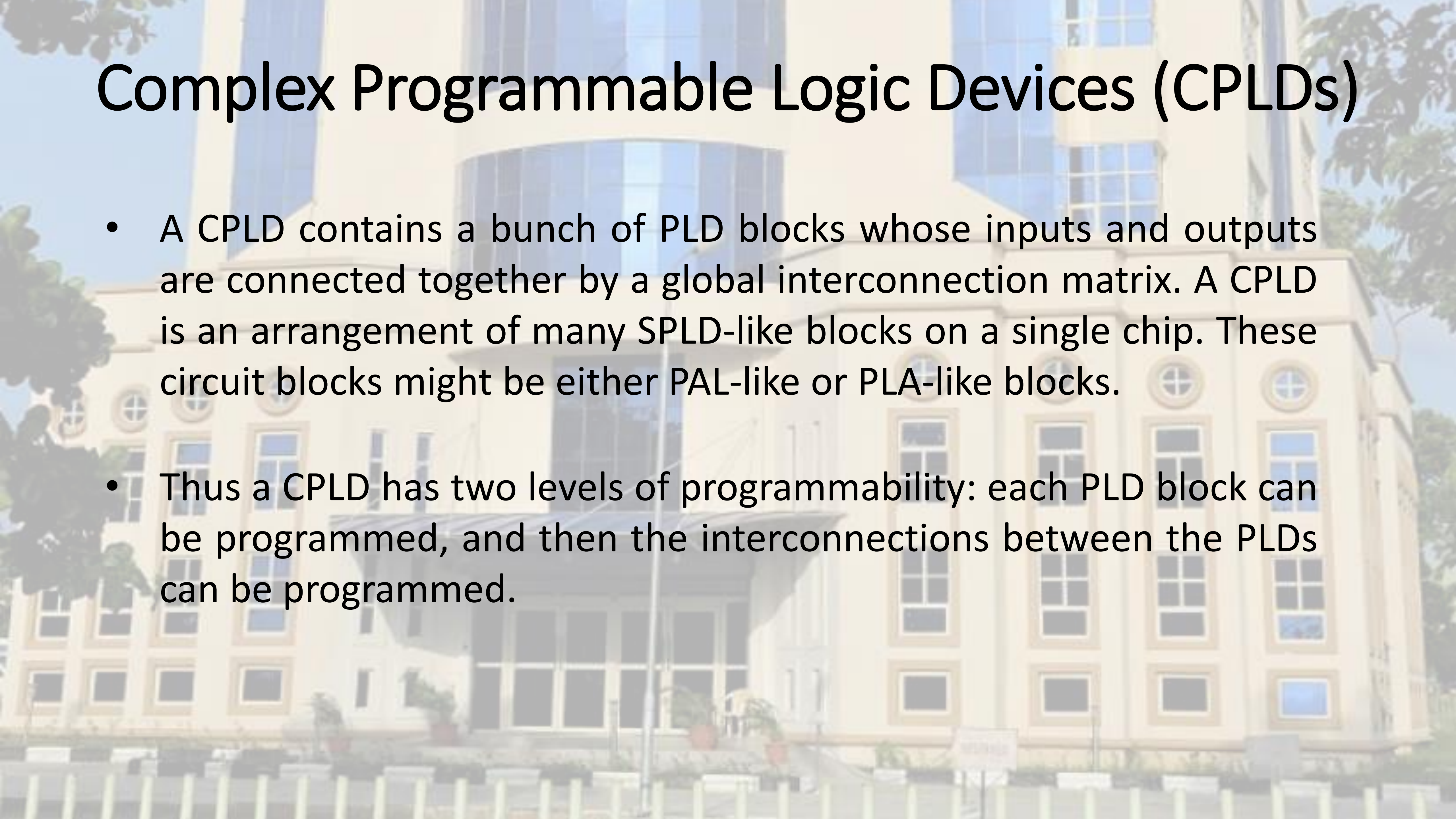


Programmable Read Only Memory (cont’d)

- Another practical application of PROM device is BCD to 7 Segment Display Controller and the corresponding input and output relationship are shown in the following table.



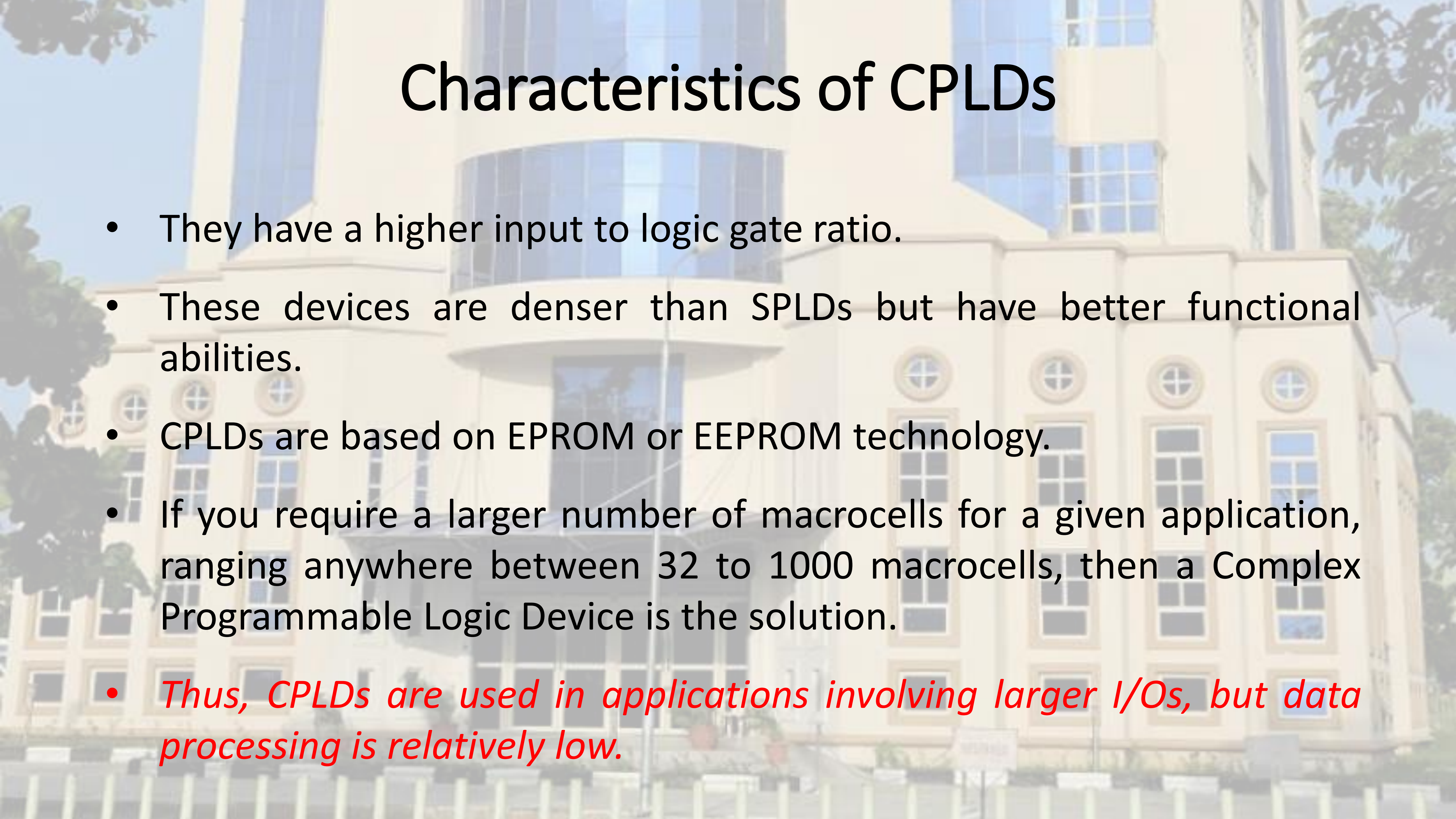
A B C D	C0	C1	C2	C3	C4	C5	C6
0 0 0 0	1	1	1	1	1	1	0
0 0 0 1	0	1	1	0	0	0	0
0 0 1 0	1	1	0	1	1	0	1
0 0 1 1	1	1	1	1	0	0	1
0 1 0 0	0	1	1	0	0	1	1
0 1 0 1	1	0	1	1	0	1	1
0 1 1 0	1	0	1	1	1	1	1
0 1 1 1	1	1	1	0	0	0	0
1 0 0 0	1	1	1	1	1	1	1
1 0 0 1	1	1	1	0	0	1	1
1 0 1 0	X	X	X	X	X	X	X
1 0 1 1	X	X	X	X	X	X	X
1 1 0 0	X	X	X	X	X	X	X
1 1 0 1	X	X	X	X	X	X	X
1 1 1 0	X	X	X	X	X	X	X
0 1 1 1	X	X	X	X	X	X	X



Complex Programmable Logic Devices (CPLDs)

- A CPLD contains a bunch of PLD blocks whose inputs and outputs are connected together by a global interconnection matrix. A CPLD is an arrangement of many SPLD-like blocks on a single chip. These circuit blocks might be either PAL-like or PLA-like blocks.

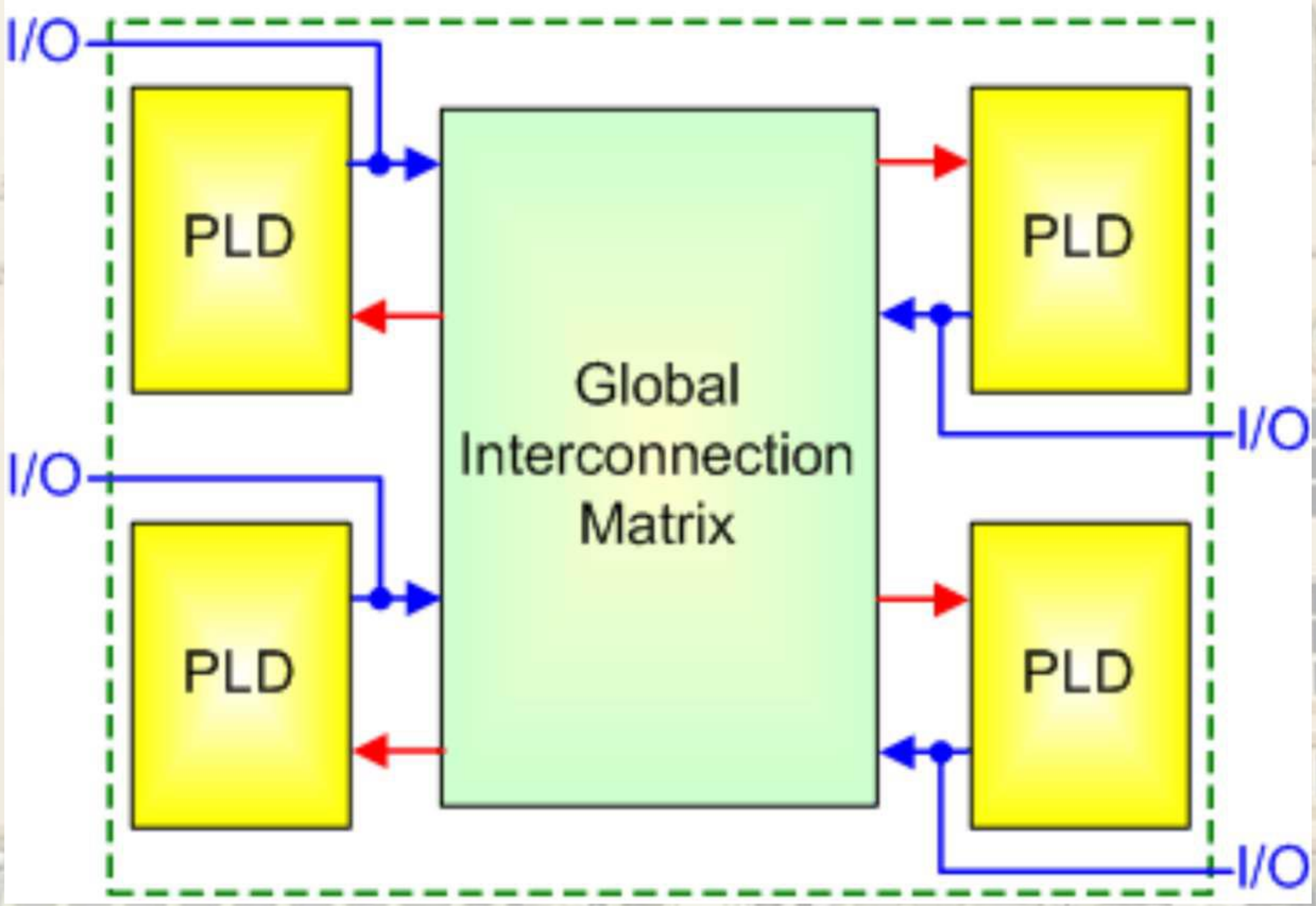
- Thus a CPLD has two levels of programmability: each PLD block can be programmed, and then the interconnections between the PLDs can be programmed.



Characteristics of CPLDs

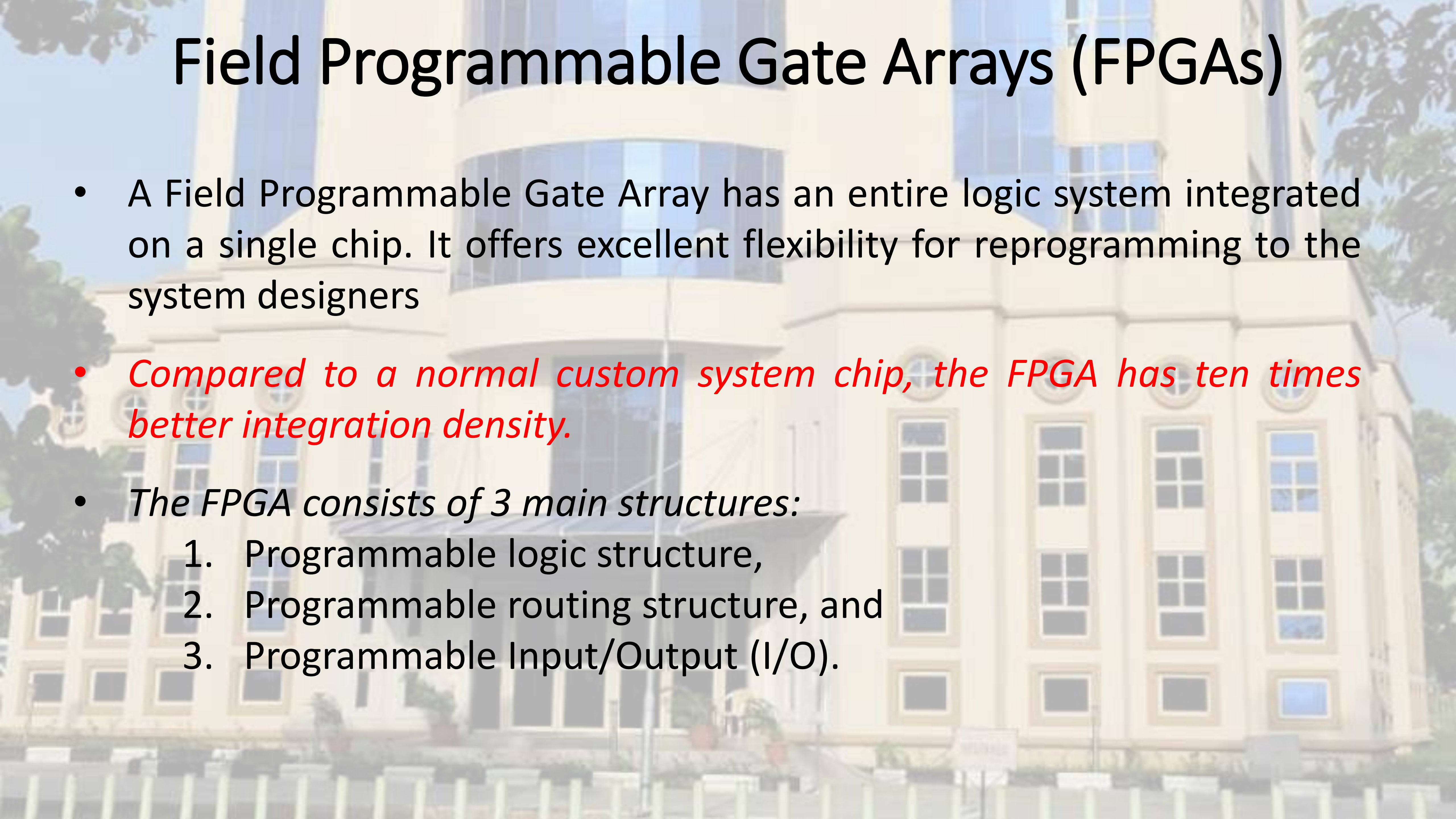
- They have a higher input to logic gate ratio.
- These devices are denser than SPLDs but have better functional abilities.
- CPLDs are based on EPROM or EEPROM technology.
- If you require a larger number of macrocells for a given application, ranging anywhere between 32 to 1000 macrocells, then a Complex Programmable Logic Device is the solution.
- *Thus, CPLDs are used in applications involving larger I/Os, but data processing is relatively low.*

Architecture of CPLDs



Applications of CPLDs

- Complex programmable logic devices are ideal for high performance, critical control applications.
- CPLD can be used in digital designs to perform the functions of boot loader
- CPLD is used for loading the configuration data of a field programmable gate array from non-volatile memory.
- Generally, CPLDs are used in small design applications like address decoding
- CPLDs are frequently used in many applications like cost-sensitive, battery operated portable devices due to its low size and usage of low power.



Field Programmable Gate Arrays (FPGAs)

- A Field Programmable Gate Array has an entire logic system integrated on a single chip. It offers excellent flexibility for reprogramming to the system designers

- *Compared to a normal custom system chip, the FPGA has ten times better integration density.*

- *The FPGA consists of 3 main structures:*
    1. Programmable logic structure,
    2. Programmable routing structure, and
    3. Programmable Input/Output (I/O).

Applications of FPGAs

- **Implementation of random logic**
  - ➢ easier changes at system-level (one device is modified)
  - ➢ can eliminate need for full-custom chips

- **Prototyping**
  - ➢ ensemble of gate arrays used to emulate a circuit to be manufactured
  - ➢ get more/better/faster debugging done than possible with simulation accelerators attached to general-purpose computers

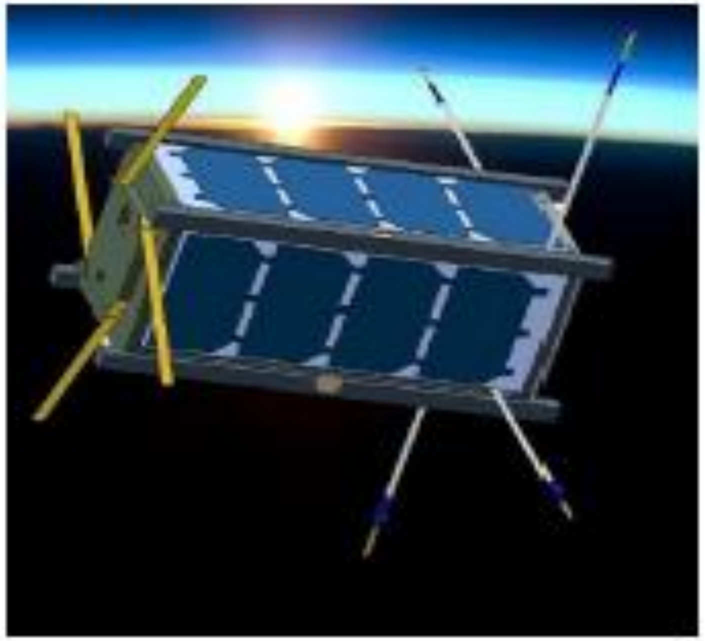
Applications of FPGAs (con’d)

- **Reconfigurable hardware**
  - one hardware block used to implement more than one function
  - functions must be mutually-exclusive in time
  - can greatly reduce cost while enhancing flexibility
  - RAM-based only option

- **Special-purpose computation engines**
  - hardware dedicated to solving one problem (or class of problems)
  - accelerators attached to general-purpose computers

Examples of FPGA use

- Aerospace
- Defence
- Medicine
- Imaging / Image processing
- Digital signal processing
- ASIC prototyping
- Software defined Radio
- Software defined GPS



Pico satellite (OBC)

NSM

Air-to-Air missile

PAL Vs FPGA

Feature	PAL	FPGA
Flexibility	Limited	High
Complexity	Low to moderate	High
Reconfigurability	Usually single-use	Multiple reconfigurations
Power Consumption	Low	Higher
Applications	Simple logic, address decoding	Complex systems, DSP, AI

Practice Questions

A	B	C	Output (Y)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Implement the combinational circuit having the shown truth table, using PLA

Thank You