AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-4 Lec-4

Gateway Classes

Today's target

- Interfacing of Logic families
- PYQ



Avinash Sir

AKTU : ***DIGITAL SYSTEM DESIGN (BEC-302)*** **Syllabus**

UNIT- 4

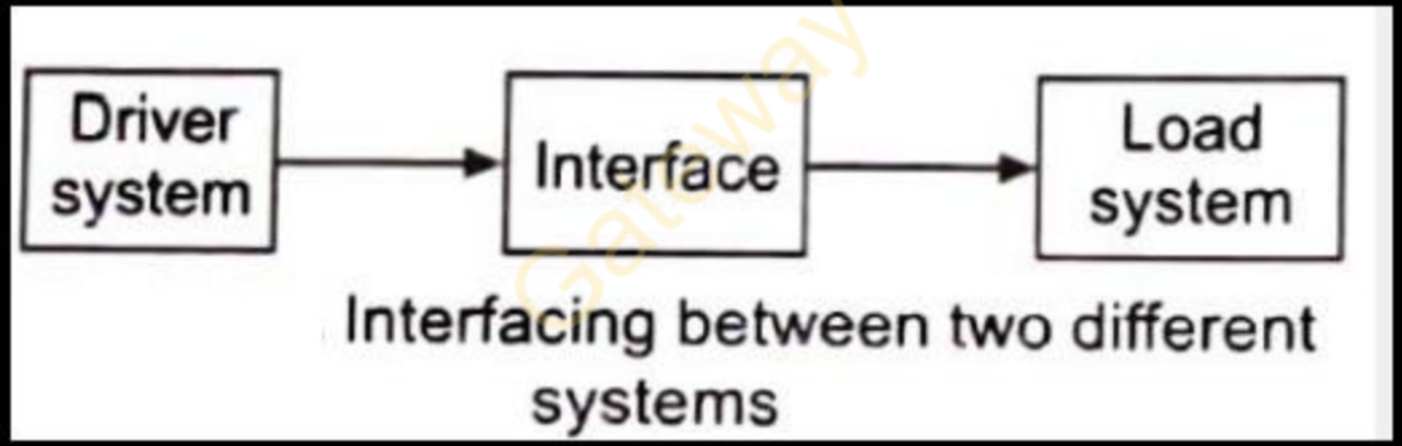
- Logic families and semiconductor memories: TTL NAND gate, specifications, noise margin, propagation delay, fan-in, fan-out, tristate TTL, ECL, CMOS families and their interfacing, memory elements, concept of programmable logic devices like FPGA, logic implementation using programmable devices.

Gateway Classes

Interface of logic Families

What is Interfacing?

- Interfacing = connecting the output of one logic family to the input of another.
- Needed when electrical characteristics differ (voltage levels, current levels).
- An interface circuit adjusts the signal so that the driver output becomes compatible with the load input.



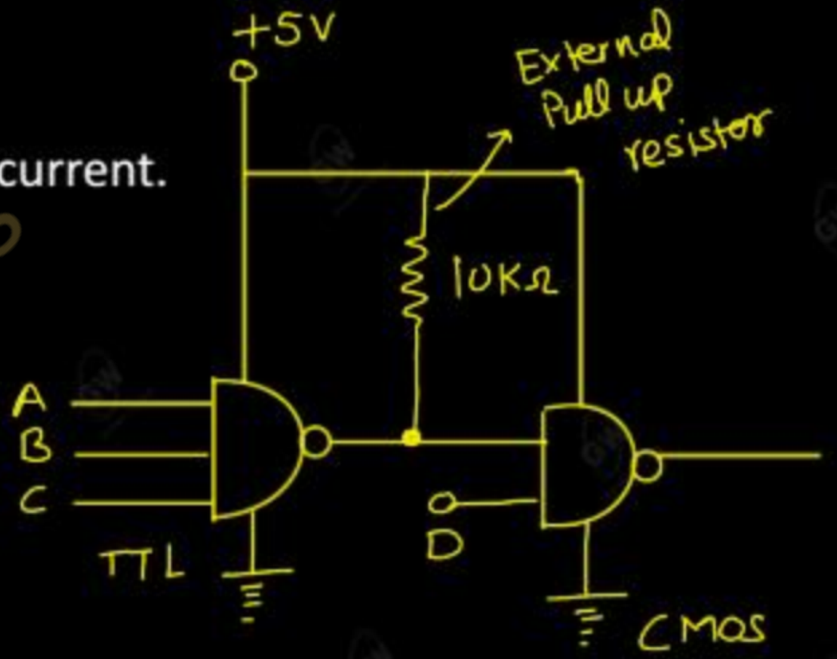
Interfacing between two different systems

Gateway Classes

TTL Driving CMOS

Basic Issue

- TTL and CMOS have different voltage levels.
- Current is not a problem because TTL output current > CMOS input current.
- Voltage mismatch is the major issue:
  - TTL VOH(min) is lower than CMOS VIH(min) requirement.
  - TTL HIGH output ≈ 2.4 V (min).
  - CMOS HIGH input requires ≥ 3.5 V (for 4000B).



Solution — Pull-up Resistor

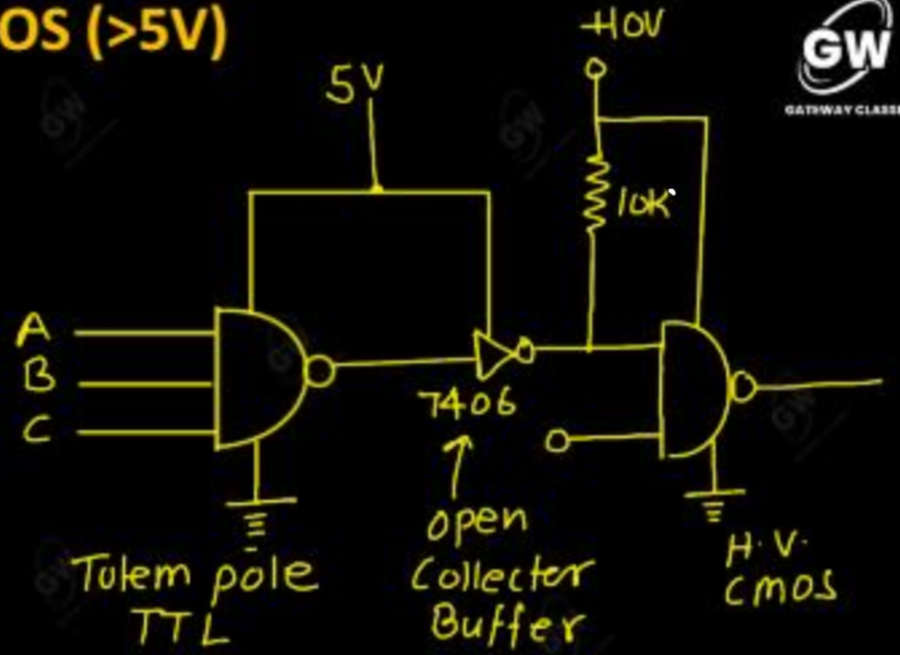
- A pull-up resistor (≈10 kΩ) is connected from TTL output to +5V.
- This raises TTL HIGH level to approx. 5 V → suitable for CMOS.
- Not required for:

  74HCT, 74ACT, These CMOS families accept TTL levels directly.

Gateway Classes

1.2. TTL Driving High-Voltage CMOS (>5V)

Problem:-High-voltage CMOS (e.g., VDD = 10–15V) requires HIGH input ≈ 7V or more. TTL cannot output >5V, so pull-up method cannot be used.

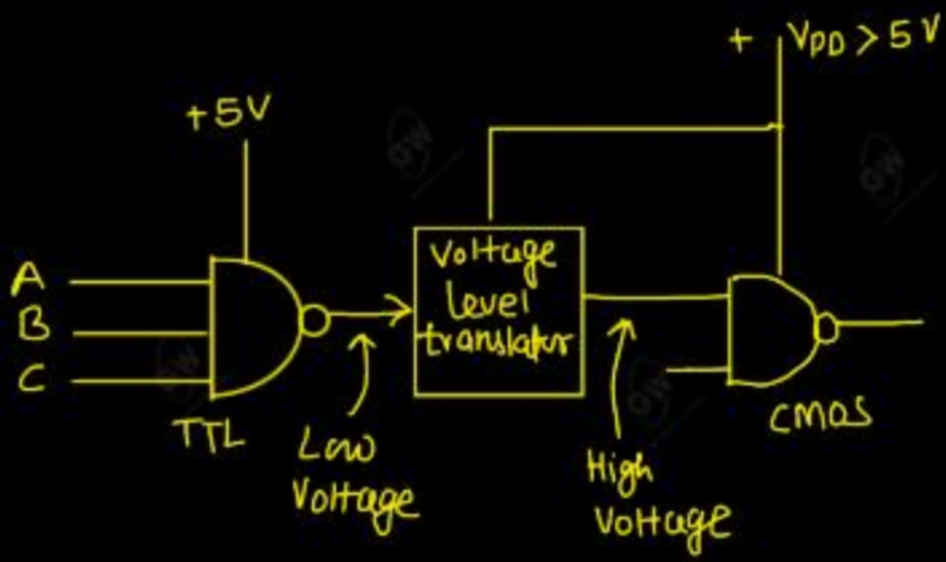


Solution —

- Open Collector Buffer (7406)Use 7406 (an inverting open-collector buffer) + pull-up resistor to high voltage.
- 7406 supports output up to 30V.
- Pull-up resistor raises the output to CMOS supply.

1.3 Interfacing Using Level Shifters

- A level shifter IC (e.g., 4504B) converts TTL 5V levels → CMOS higher voltage.
- Used when: Direct pull-up is not possible.
- Output sequencing must be clean. Voltage difference is large.



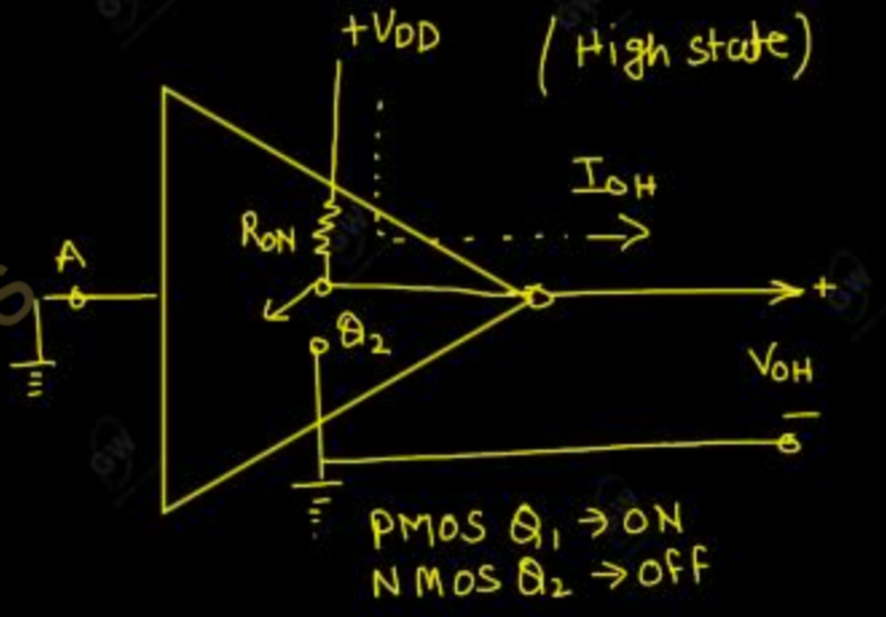
Gateway Classes

CMOS Driving TTL

To analyze, understand CMOS inverter outputs:

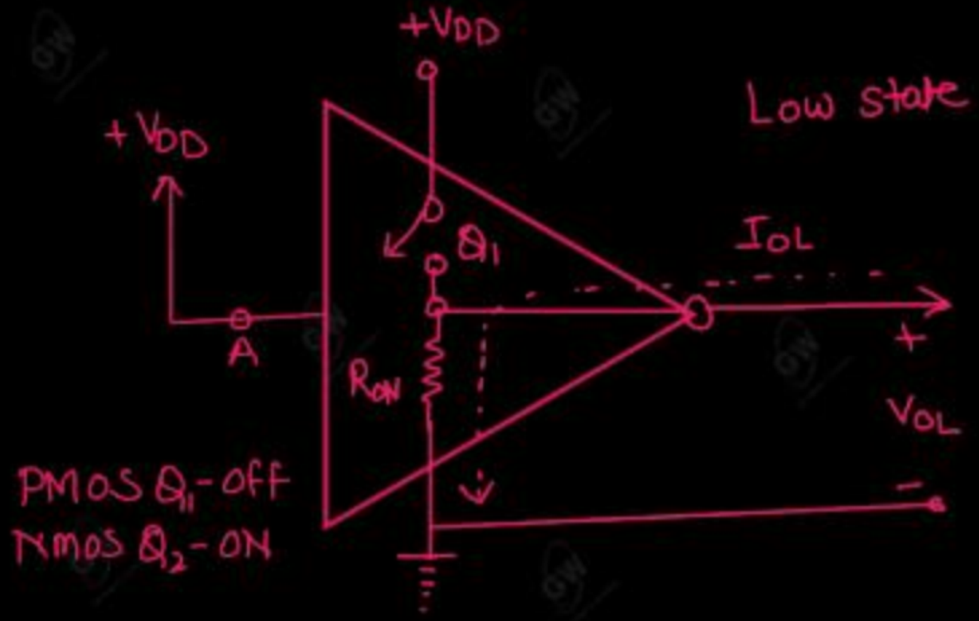
CMOS Output — HIGH State (equivalent circuit)

- PMOS ON → represented as RON (100 Ω – 1000 Ω).
- NMOS OFF → open switch.
- Acts like VDD source with series resistance RON.



CMOS Output — LOW State (equivalent circuit)

- PMOS OFF.
- NMOS ON → represented as RON to ground.
- Acts as low-resistance to ground.



CMOS driving TTL in HIGH State

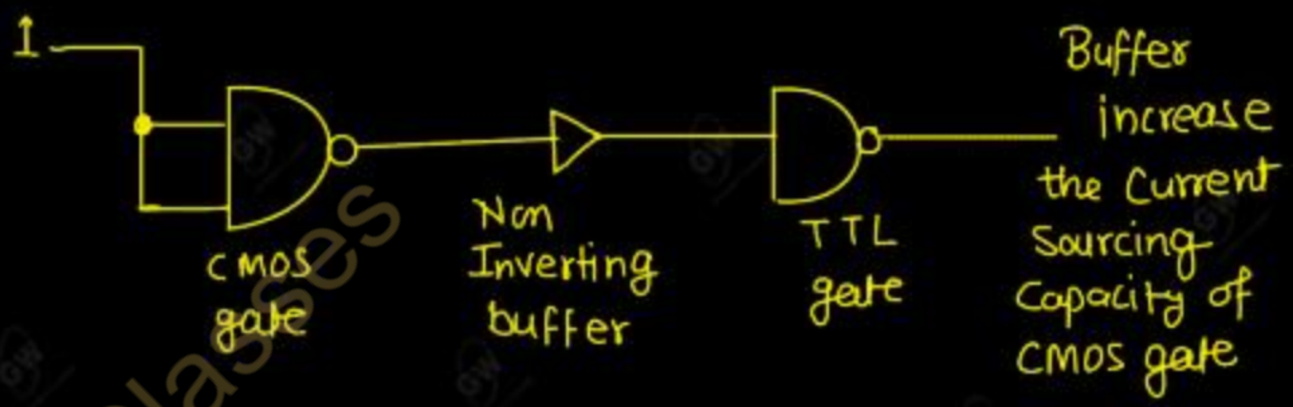
- CMOS HIGH output (≈4.9–5V) > TTL VIH(min) (≈2V).
- CMOS I_{oH} can supply TTL input current easily.
- No interface needed.

Gateway Classes

CMOS driving TTL in LOW State

Problem

- CMOS LOW output current IOL ≈ 0.4 mA.
- TTL input low current requires 1.6 mA.
- Thus, CMOS cannot sink enough current.



Solution

- Use a non-inverting buffer between CMOS and TTL.
- Buffer increases the current sinking capacity.

Gateway Classes

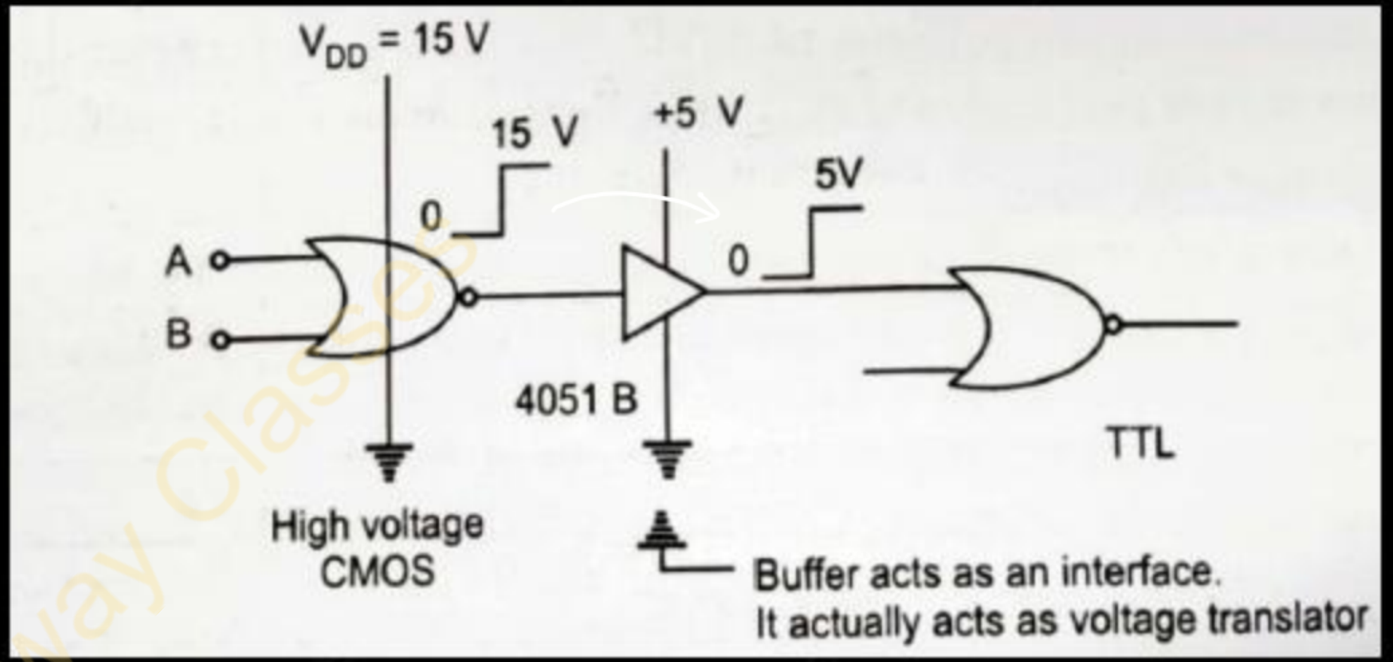
High-Voltage CMOS Driving TTL

Problem

- HV CMOS outputs ≈ 10–15V.
- TTL max input voltage allowed = 7V.
- Applying 15V would damage TTL.

Solution

- Use 4051B buffer (operates at +5V).
- Converts 15V CMOS HIGH → 5V TTL HIGH.
- Works as a voltage level translator.



Gateway Classes

Interfacing CMOS/TTL with ECL

Why Needed?

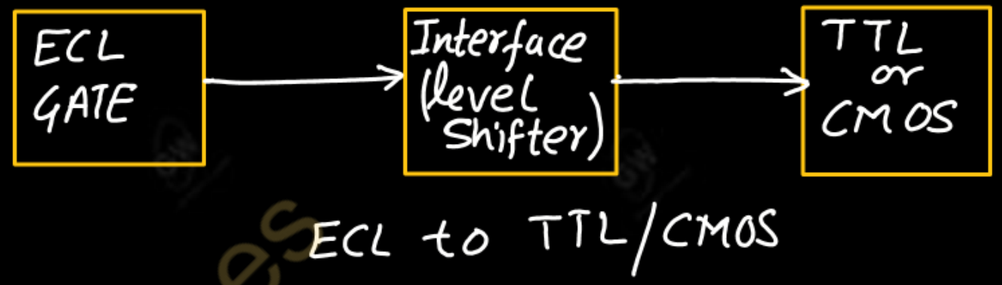
ECL has very different voltage levels:

Operates around –5.2V

Logic HIGH ≈ –0.9V

Logic LOW ≈ –1.7V

TTL/CMOS cannot directly interface.



ECL to TTL/CMOS

Solution

- Use voltage translators (in ECL families):
- TTL/CMOS → ECL converter
- ECL → TTL/CMOS converter

These circuits shift voltage levels safely.

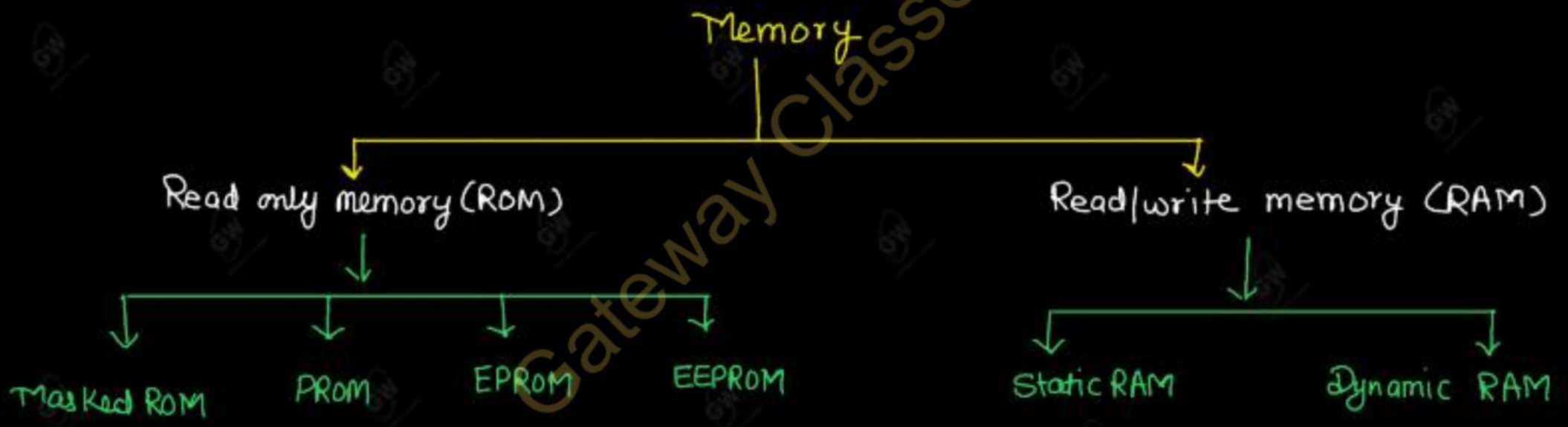


TTL/CMOS to ECL

Gateway Classes

Memory?

In the field of digital electronics, the memory is a device that is used to store data and instruction in the digital systems like computers and other microprocessor-based systems.



Gateway Classes

Types of ROMs

ROM : Read only memory: Its non volatile memory, ié, the information stored in it, is not lost even if the power supply goes off. It's used for the permanent storage of information.

The following types of ROMs an listed below :

(i) **PROM :** It's programmable ROM. Its contents are decided by the user. The user can store permanent programs, data etc in a PROM.

(ii) **EPROM :** An EPROM is an erasable PROM. The stored data in EPROM's can be erased by exposing it to UV light for about 20 min. It's not easy to erase it because the EPROM IC has to be removed from the computer and exposed to UV light.

(iii) **EEPROM (Electrically Erasable PROM) :** The chip can be erased & reprogrammed on the board easily byte by byte. It can be erased with in a few milliseconds.

Gateway Classes

COMBINATIONAL PLDs

Basic configuration of three PLDs are

1. PROM
2. programmable array logic (PAL)
3. Programmable logic array (PLA)



(a) Programmable read-only memory (PROM)



(b) Programmable array logic (PAL)



(c) Programmable logay aray (PLA)

Gateway Classes

COMBINATIONAL PLDs

A Programmable Logic Device (PLD) is an integrated circuit that can be configured or "programmed" by the user to implement specific logic functions .

Basic Structure: PLDs are essentially a combination of AND gates and OR gates .

Programmability: The connections within the AND array and OR array can be programmed (made or broken) by the user according to the required function

Categories of PLDs (Based on Programmability)

PLA (Programmable Logic Array): Both the AND array and the OR array are programmable.

PAL (Programmable Array Logic): The AND array is programmable, but the OR array is fixed (hardwired)

Gateway Classes

$(A + AB)$
$A(1+B) = A$

1. Logic Function Implementation using PLA

The primary goal is to implement a logic function using the PLA architecture .

Requirement: Minimized Boolean Expression To implement any function using a PLA. If the expression is not minimized (e.g., A + AB), it must be minimized first (e.g., to A) before implementation .

If the question provides minterms or a truth table, the function must first be calculated and minimized using a K-Map.

Calculating Required Gates (Size of the PLA)For implementing multiple output functions using a single PLA:

Number of OR Gates: This is equal to the Number of Output Functions .Example: If you have two functions (F1 and F2), you need two OR gates.

Number of AND Gates: This is equal to the total number of unique product terms found by combining all the functions.

Crucial Rule: Repeated product terms are NOT considered again (they are counted only once) .

Gateway Classes

Ex-1 A Combinational Circuit is defined by function

$F_1 = \Sigma m(3,5,7)$, $f_2 = \Sigma m(4,5,7)$

Implement the Circuit with PLA

Soln

F_1

A \ BC	00	01	11	10
0	0	1	1 (3)	2
1	1 (4)	1 (5)	1 (7)	6

F_2

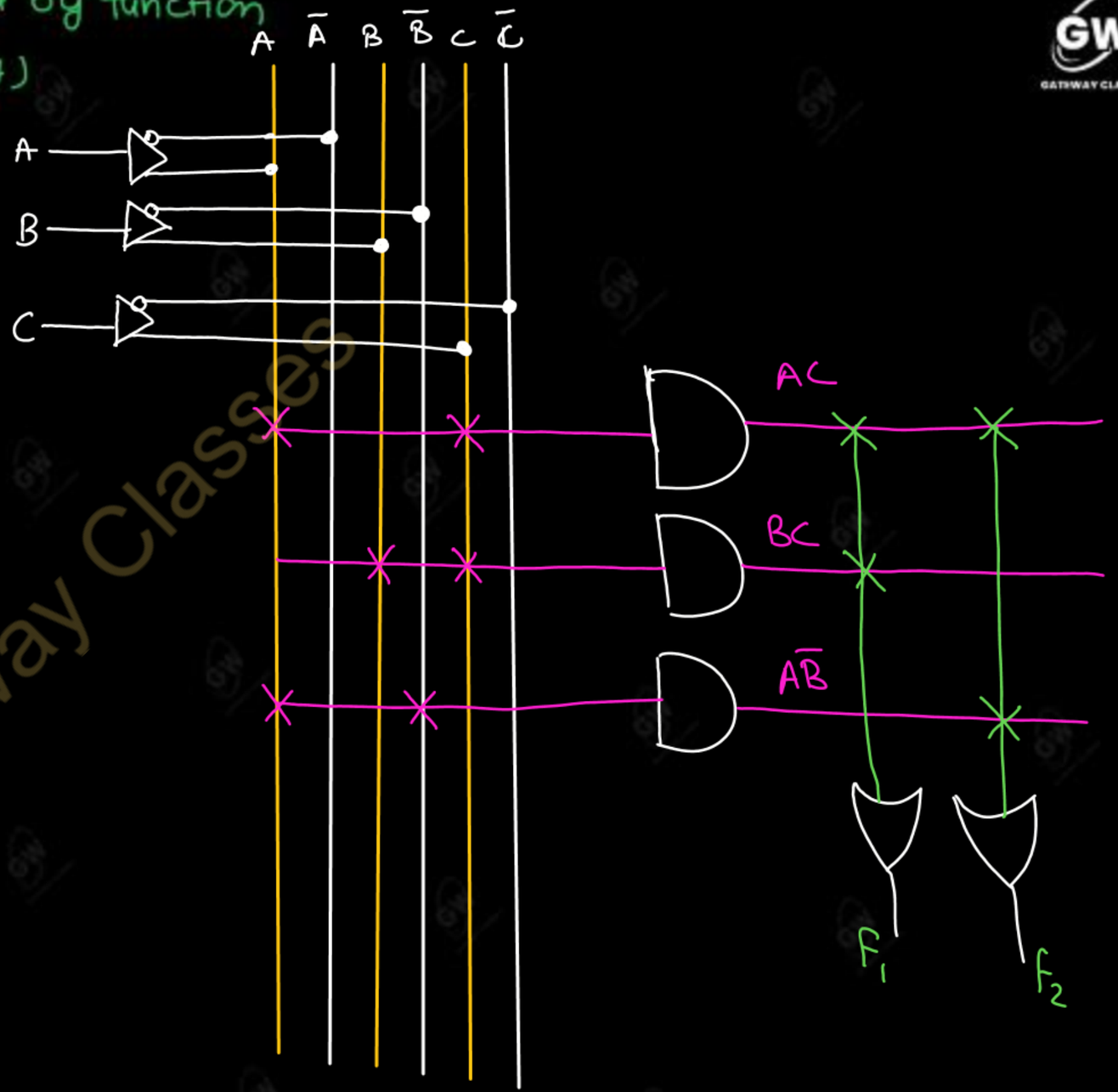
A \ BC	00	01	11	10
0	0	1	2	3
1	1 (4)	1 (5)	1 (7)	6

$F_1 = \underline{BC} + \underline{AC}$

$F_2 = \underline{A\bar{B}} + \underline{AC}$

No. of OR GATE = No. of function = 02

No. of AND GATE = No. of unique Product term = 03



Gateway Classes

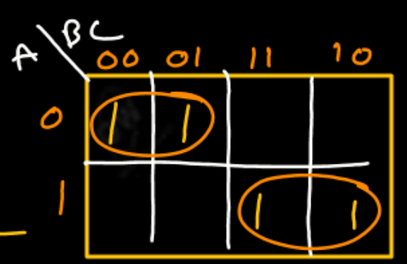
Ex-2 Implement the following function using PLA

$F_1(A,B,C) = \Sigma m(1,2,4,6)$, $F_2(A,B,C) = \Sigma m(0,1,6,7)$, $F_3(A,B,C) = \Sigma m(2,6)$

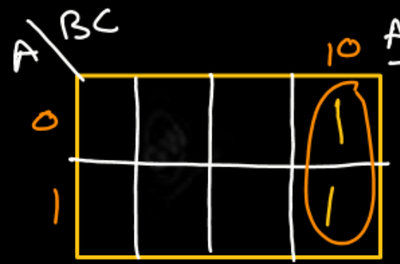
Sol :-



$F_1 = B\bar{C} + \bar{A}\bar{B}C + A\bar{C}$



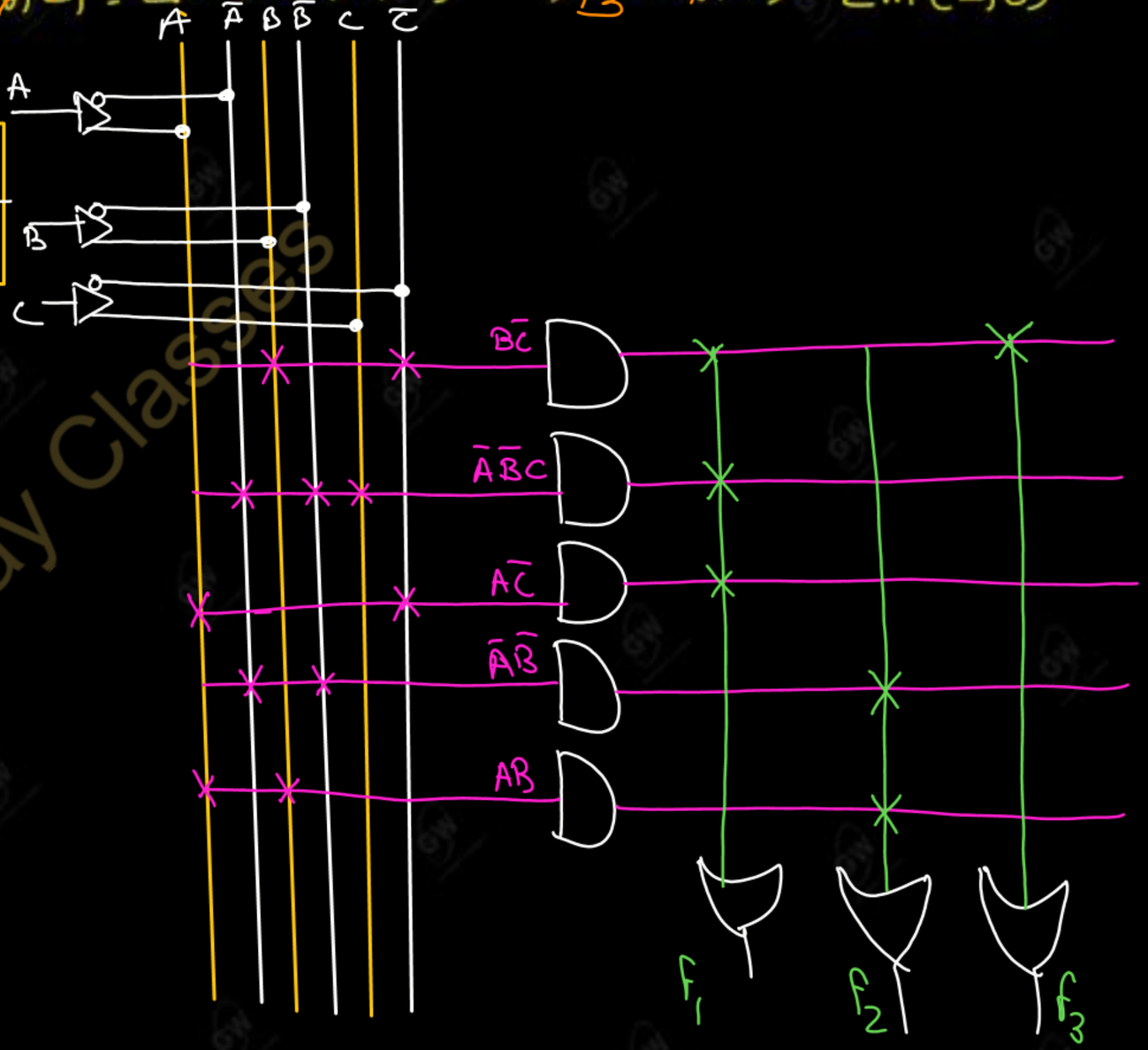
$f_2 = \bar{A}\bar{B} + AB$



$f_3 = B\bar{C}$

No. of OR GATE = 3 function = 3

No of AND GATE = 5



Gateway Classes

2. PAL (Programmable Array Logic)

PAL is another type of Programmable Logic Device (PLD).

Key Difference from PLA: In PAL, the AND array is programmable (like in PLA), but the OR logic (array) is fixed (hardwired). The fixed OR logic is the defining characteristic of the PAL structure.

Sizing the PAL (Calculating Required Gates)

To implement a function using PAL, the calculation for the number of gates reflects the fixed nature of the OR array.

Number of OR Gates: Equal to the Number of Output Functions (same as PLA).

Number of AND Gates: Calculated by finding the Maximum number of product terms in any single function, then multiplying by the Total number of functions.

Total AND Gates = Max Product Terms per Function x Number of Functions

Contrast with PLA: Unlike PLA, there is no concept of not considering repeated product terms. In PAL, an AND gate combination is fixed for each output function based on the maximum required.

Gateway Classes

Q: Implement the following Boolean function using suitable PAL

$W(A,B,C,D) = \Sigma m(1,3,4,6,9,11,12,14)$

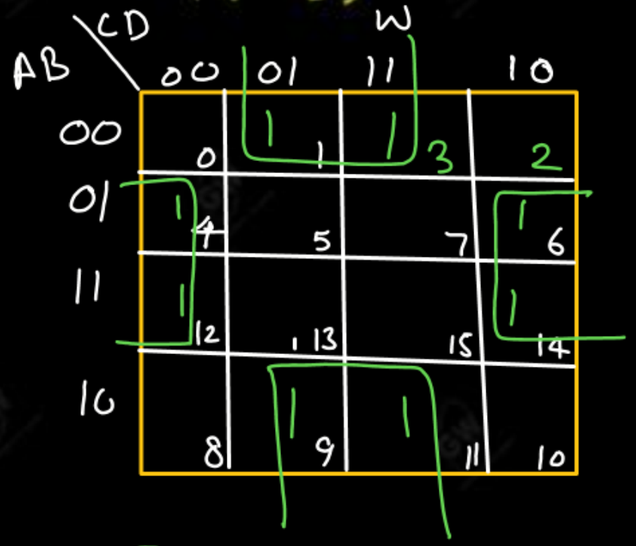
$x(A,B,C,D) = \Sigma m(1,3,4,6,9,11,12,14,15)$

$y(A,B,C,D) = \Sigma m(0,2,4,6,8,12)$

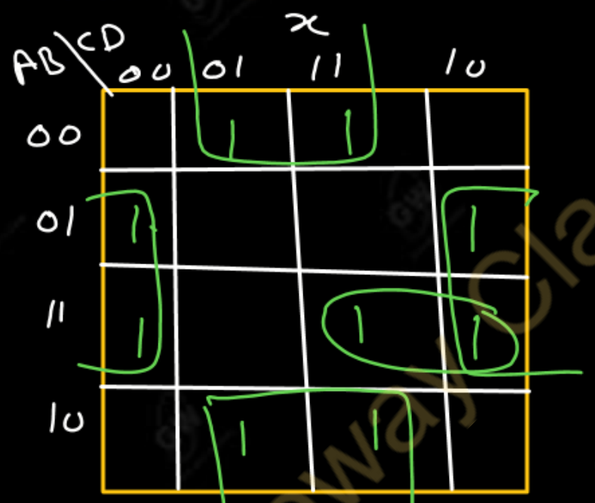
$z(A,B,C,D) = \Sigma m(2,3,8,9,12,13)$

No. of OR GATE = No. of fun = 4

No. of AND GATE = 2 × 4 = 8

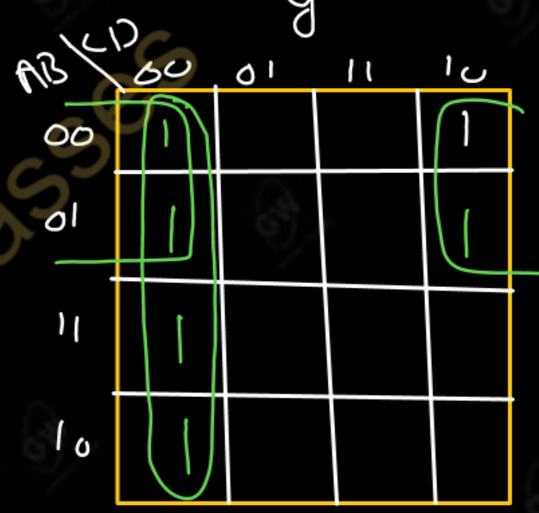


$W = \bar{B}D + B\bar{D}$

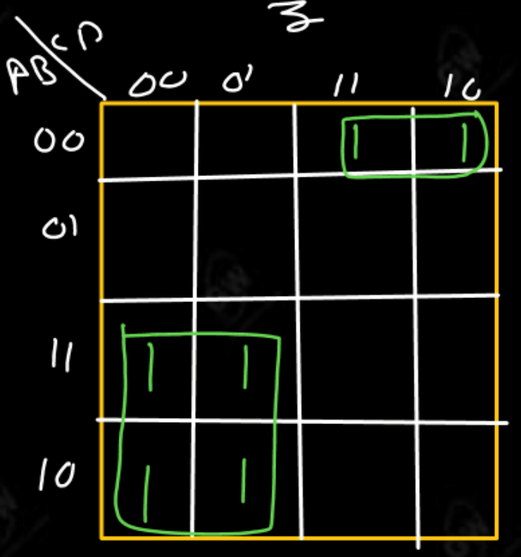


$x = B\bar{D} + \bar{B}D + ABC$

$x = W + ABC$

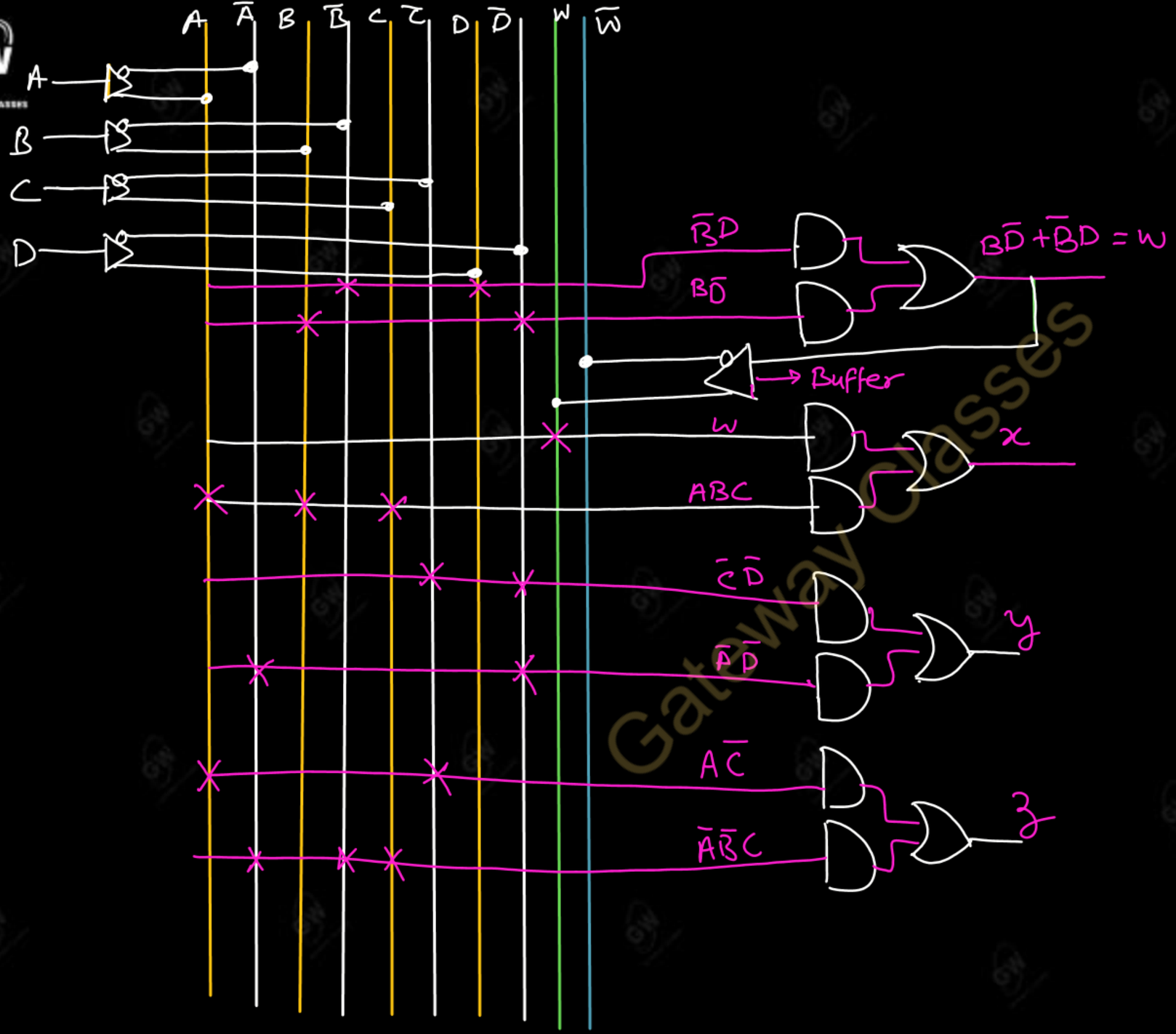


$y = \bar{C}\bar{D} + \bar{A}\bar{D}$



$z = A\bar{C} + \bar{A}\bar{B}C$

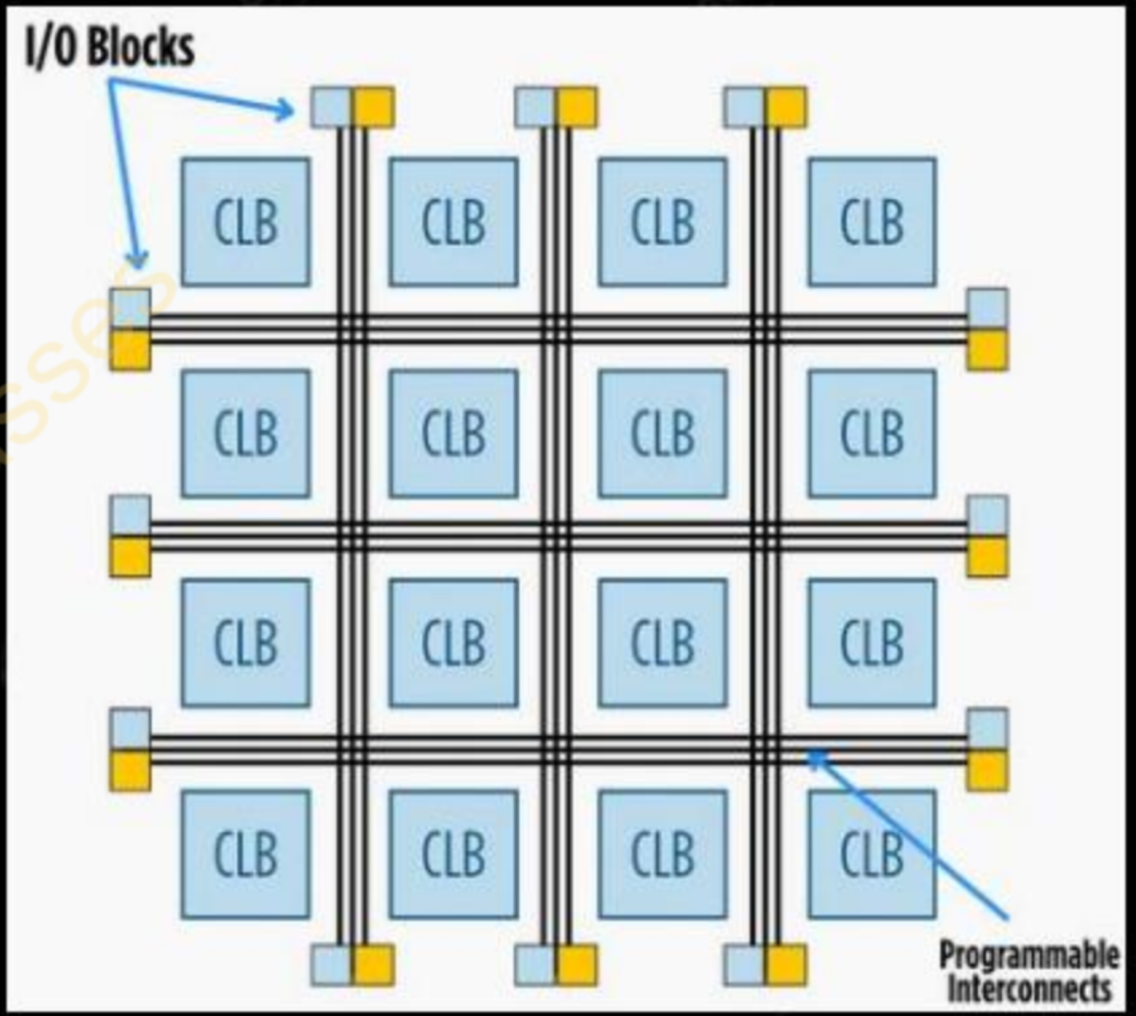
GW GATEWAY CLASSES
Gatewayclasses



Gateway Classes

Field Programmable Gate Array (FPGA)

- An FPGA (Field Programmable Gate Array) is a type of integrated circuit in digital electronics that can be programmed and reprogrammed after manufacturing to perform custom digital logic functions.
- They are built with a grid of configurable logic blocks (CLBs) and programmable interconnects, allowing designers to create custom circuits for tasks like signal processing, embedded systems, and high-performance computing.
- The configuration of the FPGA architecture is generally specified using a language, i.e., HDL (Hardware Description Language), which is similar to the one used for an ASIC (Application Specific Integrated Circuit).



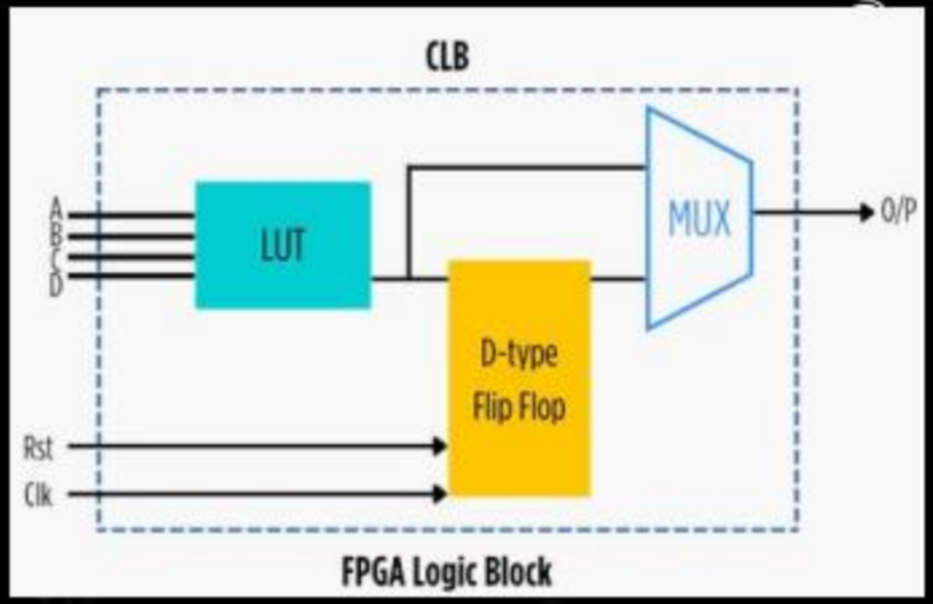
Gateway Classes

FPGA Architecture

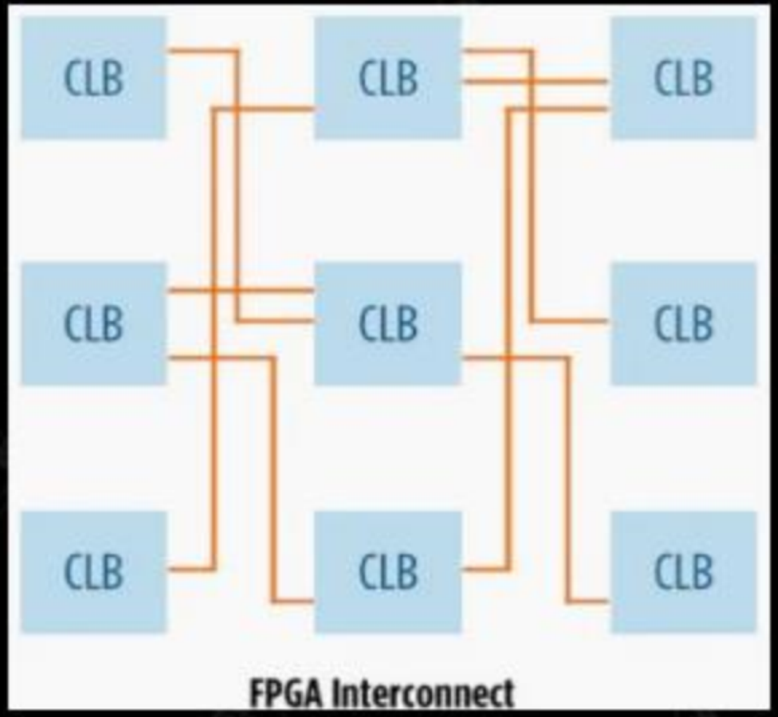
- **Configurable Logic Blocks (CLBs):** The core of FPGA functionality, these blocks contain logic gates and a small amount of memory that can be programmed to perform various logical functions. They are the building blocks for creating more complex digital circuits.

  LUT (Look Up Table) implements the combinational logical functions; the MUX (Multiplexer) is used for selection logic, and DFF (D type flip flop) stores the output of the LUT.

- **Programmable Interconnects:** A network of programmable wiring that connects the logic blocks. These interconnects can be configured to route signals between different logic blocks, allowing the creation of complex digital circuits by defining how data flows through the FPGA.



FPGA Logic Block

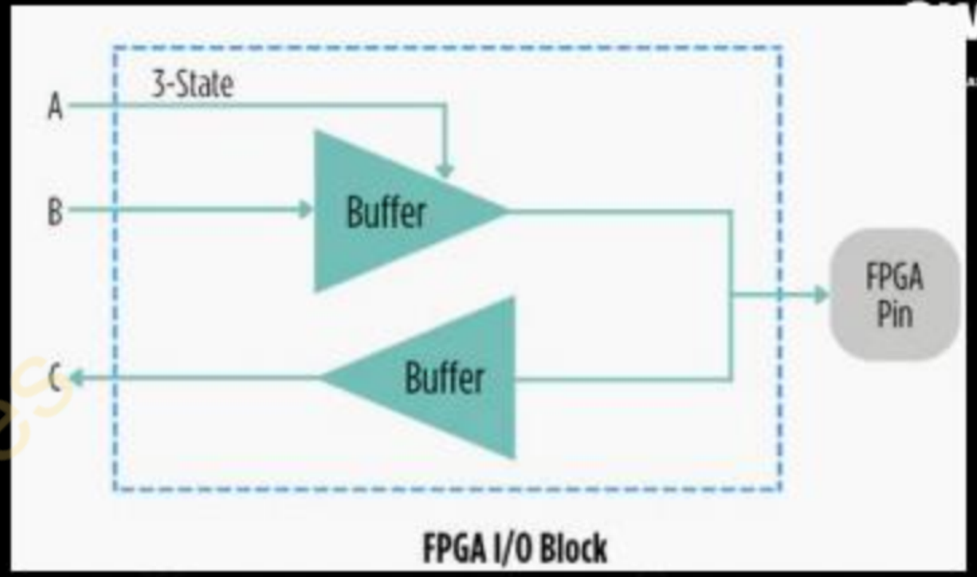


FPGA Interconnect

Gateway Classes

FPGA Architecture

I/O Blocks (Input/Output Blocks): Located next to every physical input or output pin, these blocks connect the internal logic of the FPGA to the external environment. They can be programmed to act as inputs, outputs or tri-states, enabling the FPGA to communicate with external devices and systems.



FPGA I/O Block

Advantages of FPGAs

- Flexibility: FPGAs can be reprogrammed to perform different tasks or update the functionality of a device after it has been deployed. This allows for modifications and improvements without needing to redesign the hardware.
- Rapid Prototyping: Designers can quickly implement and test their designs on FPGAs, making them ideal for prototyping and iterative development processes.
- Lower Initial Costs: For low to medium-volume productions, FPGAs generally have lower upfront costs compared to developing an ASIC, as there are no expensive fabrication costs involved.

Gateway Classes

Application of FPGA

- Telecommunications: FPGAs are essential for signal processing and updating network protocols without replacing hardware.
- Data Centers: Accelerate data processing, encryption, and network management, enhancing server efficiency.
- Aerospace and Defense: FPGAs provide reliability in harsh environments, handling radar, communications, and encryption.
- Consumer Electronics: Utilized for processing in cameras, TVs, and home systems, improving digital experiences.
- Industrial Control: Streamline control systems, automation, and machine vision, facilitating real-time processing.
- Medical & Scientific Instruments: Essential in imaging equipment, FPGAs process complex algorithms quickly, ensuring rapid diagnostics. Support research with precision and speed in data acquisition and analysis.
- Artificial Intelligence and Machine Learning: Accelerate AI/ML training and inference, supporting rapid development and deployment.

Gateway Classes

Gateway Classes

Empowering Learners, Transforming Futures

GATEWAY CLASSES

Thank You

Gateway Classes

GET IT ON Google Play

Download on the App Store

Helpline No- 7819 0058 53