PROGRAMMABLE LOGIC DEVICES (PLD)

DR. FATMA ELFOULY

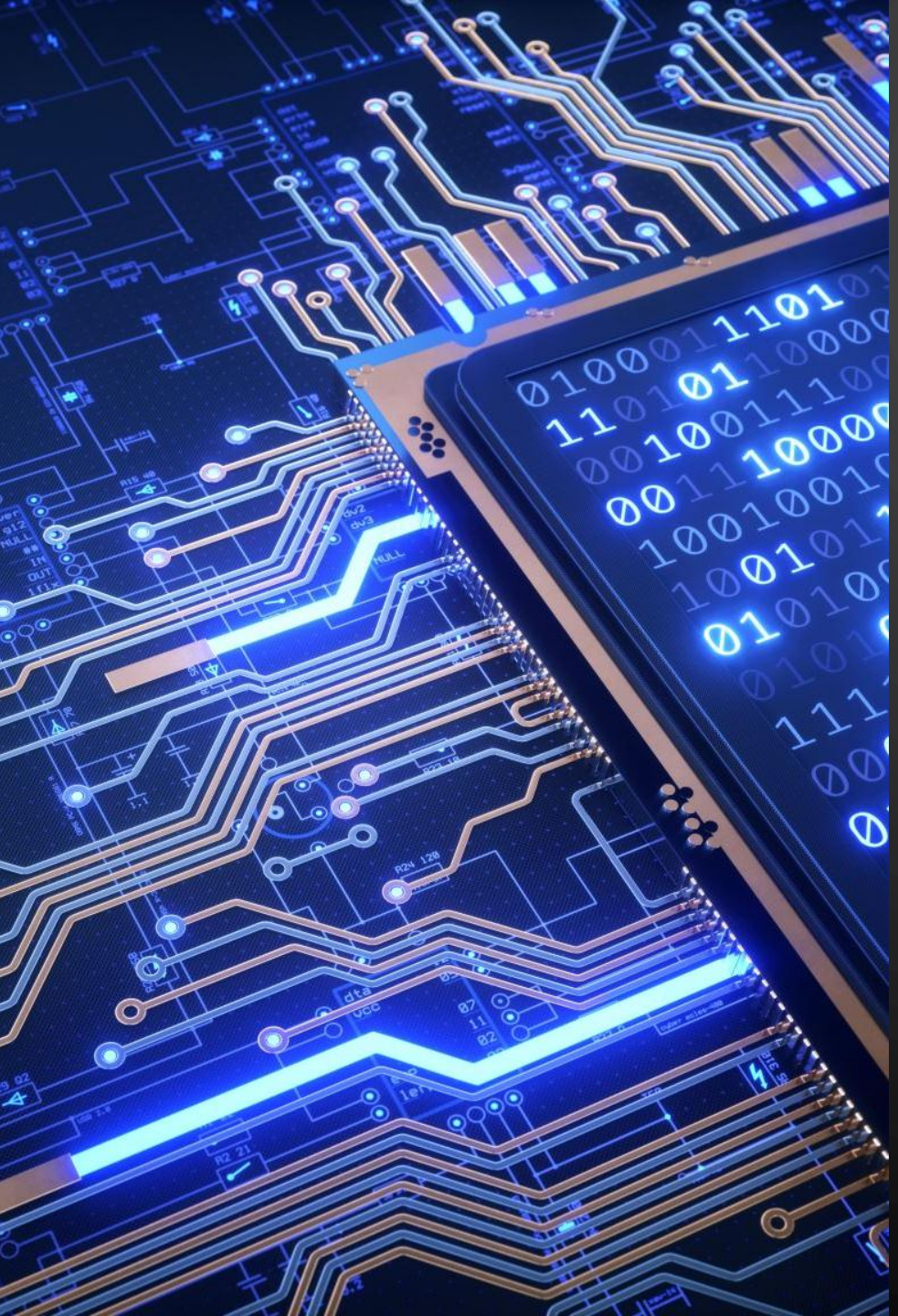
PROBLEMS BY USING BASIC GATES

Many components on PCB:

- As no. of components rise, nodes interconnection complexity grow exponentially
- Growth in interconnection will cause increase in interference, PCB size, PCB design cost, and manufacturing time

PLDC

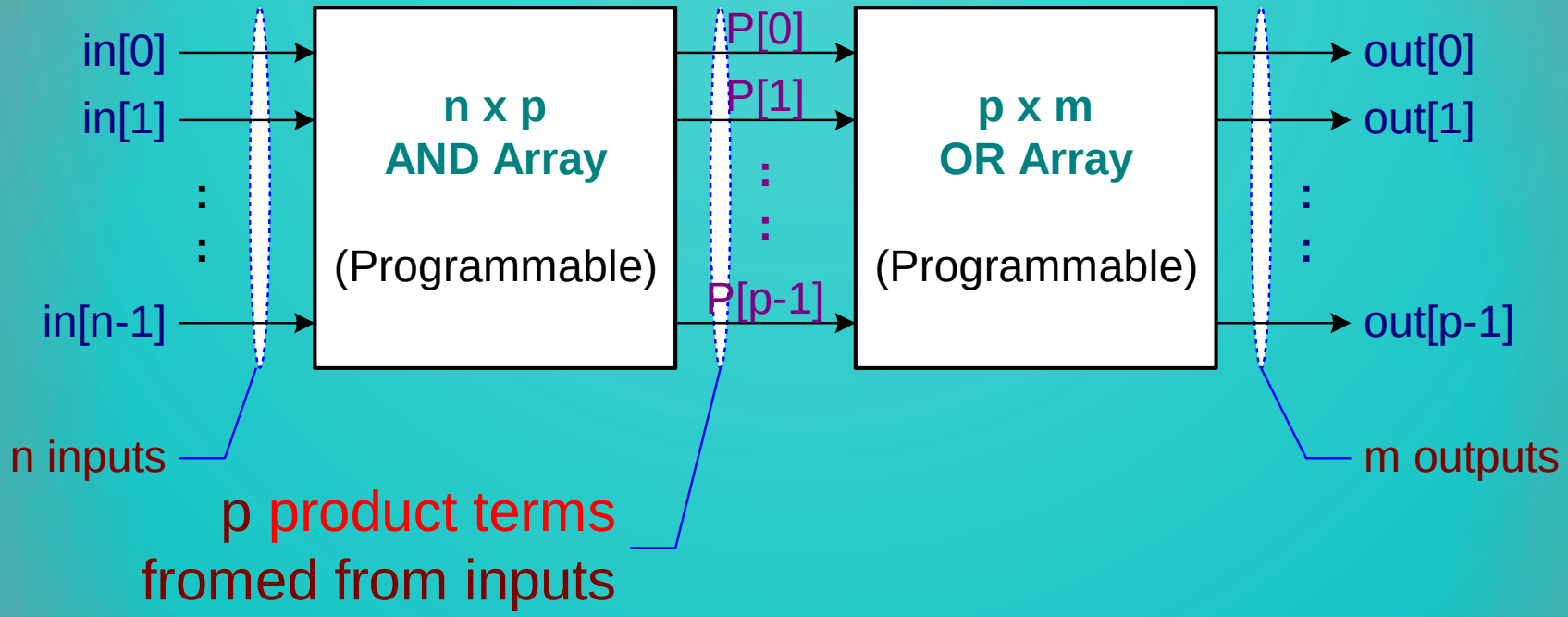
- The purpose of a PLD device is to permit elaborate digital logic designs to be implemented by the user in a single device.
- Can be erased electrically and reprogrammed with a new design, making them very well suited for academic and prototyping
- ***Types of Programmable Logic Devices***
  - **SPLDs (Simple Programmable Logic Devices)**
    - **PLA (Programmable Logic Array)**
    - **PAL (Programmable Array Logic)**
    - **GAL (Generic Array Logic)**
  - **CPLD (Complex Programmable Logic Device)**
  - **FPGA (Field-Programmable Gate Array)**



PLD

- 3 categories of PLDs:
  1. SPLD (Simple Programmable Logic Device)
     - PLA (Programmable Logic Array)
     - PAL (Programmable Array Logic)
     - Registered PAL
  2. CPLD (Complex Programmable Logic Device)
  3. FPGA (Field Programmable Gate Array)

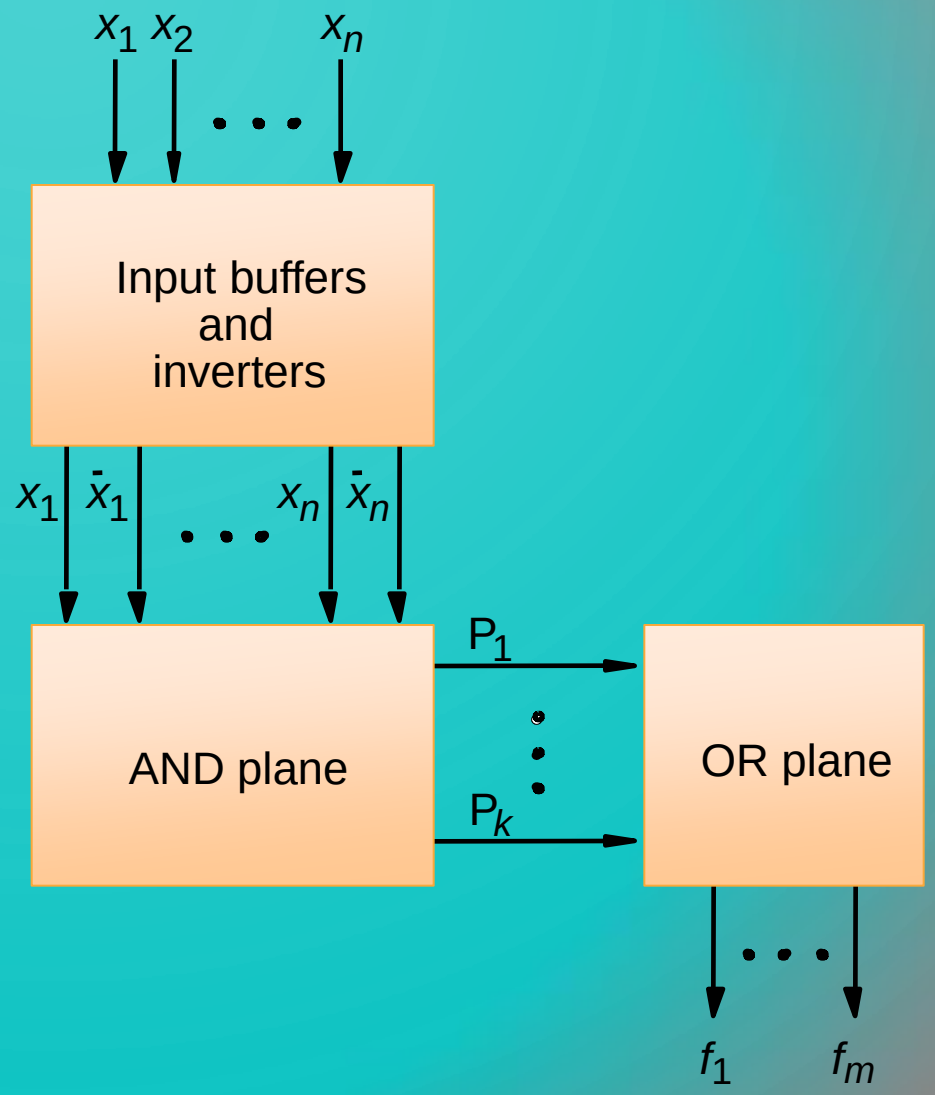
ARCHITECTURE OF PLDS



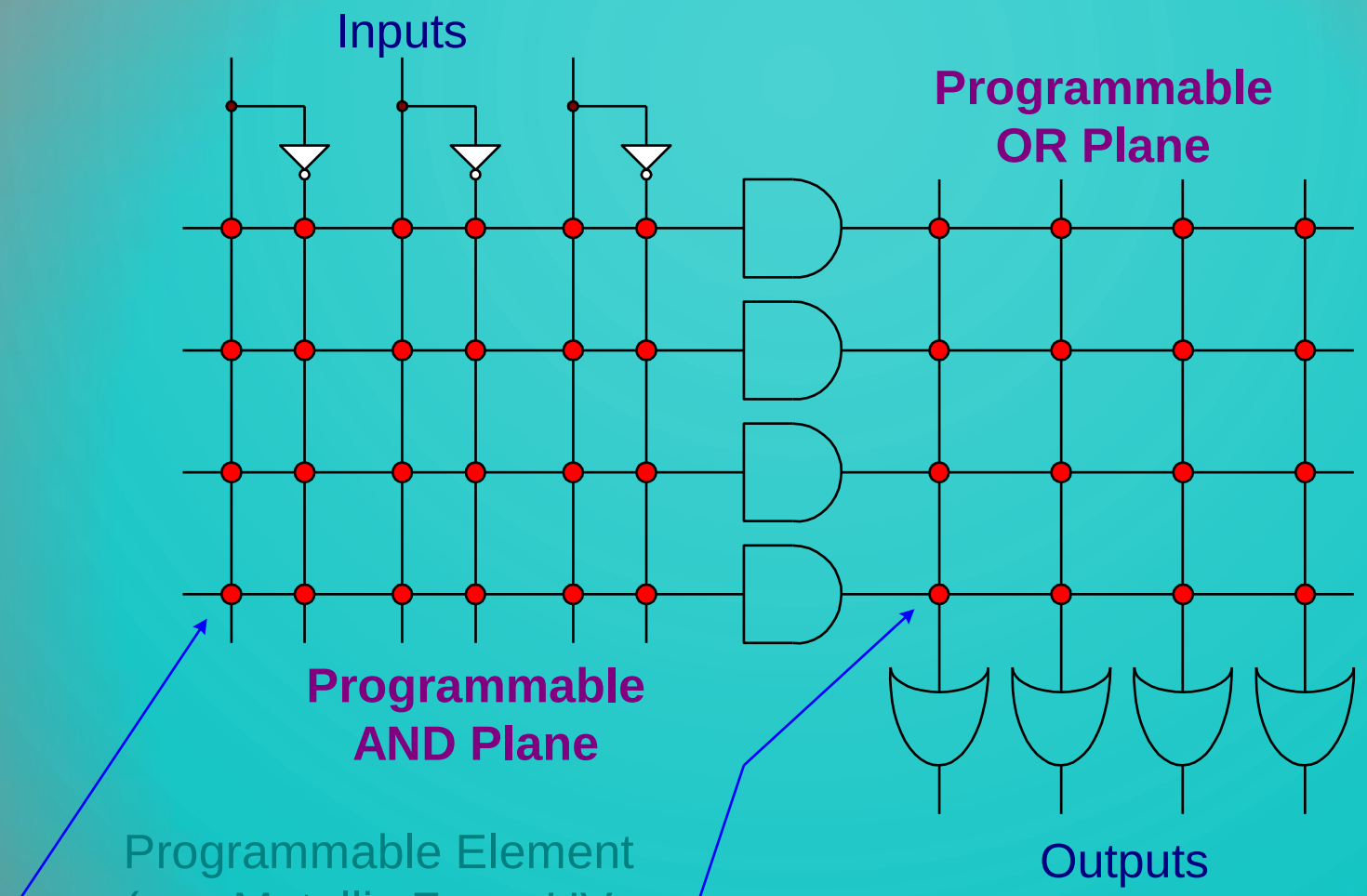
Generic Architecture of PLDs

PROGRAMMABLE LOGIC ARRAY (PLA)

- The connections in the AND plane are programmable
- The connections in the OR plane are programmable



PLA

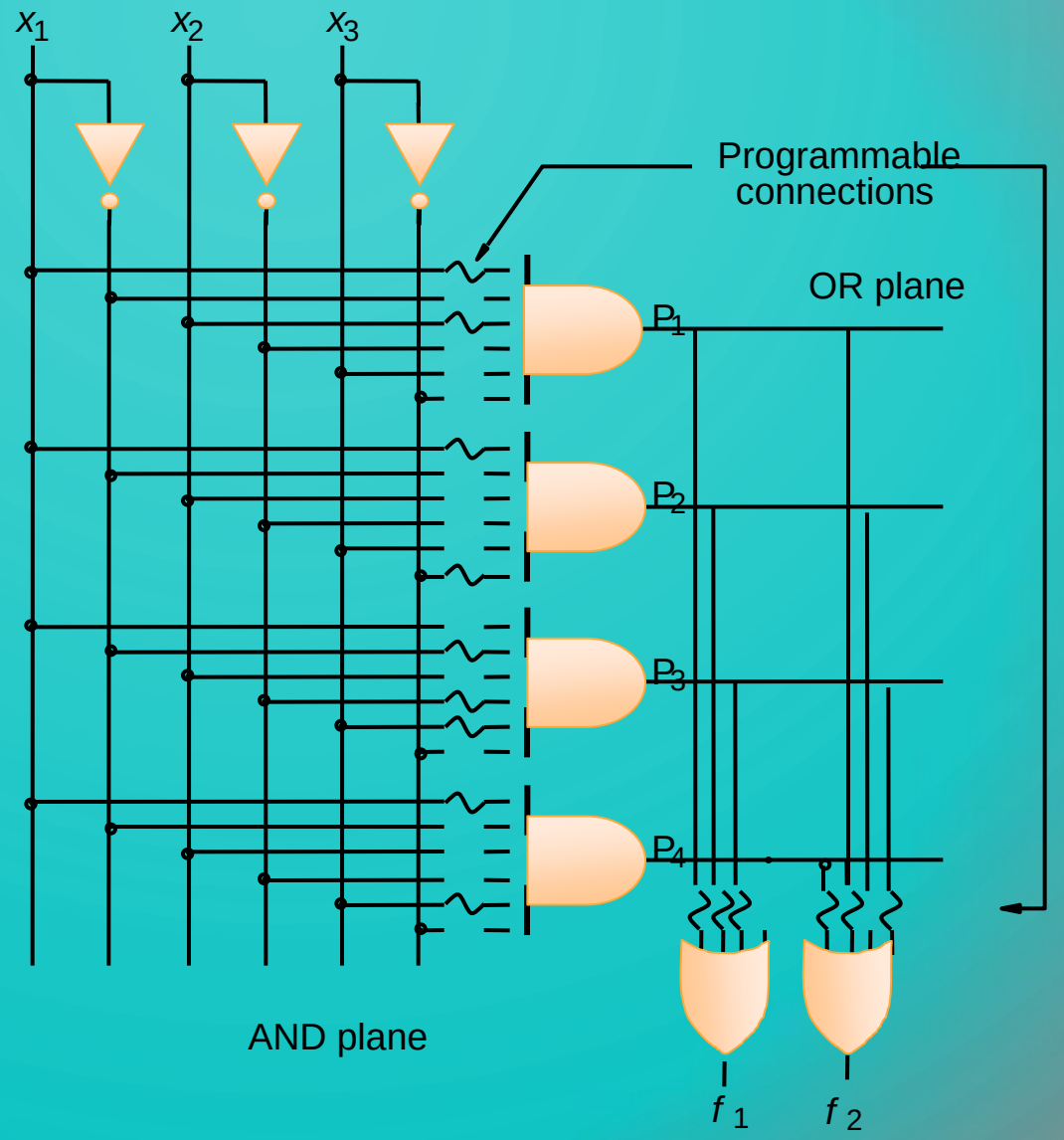


The PLA Architecture

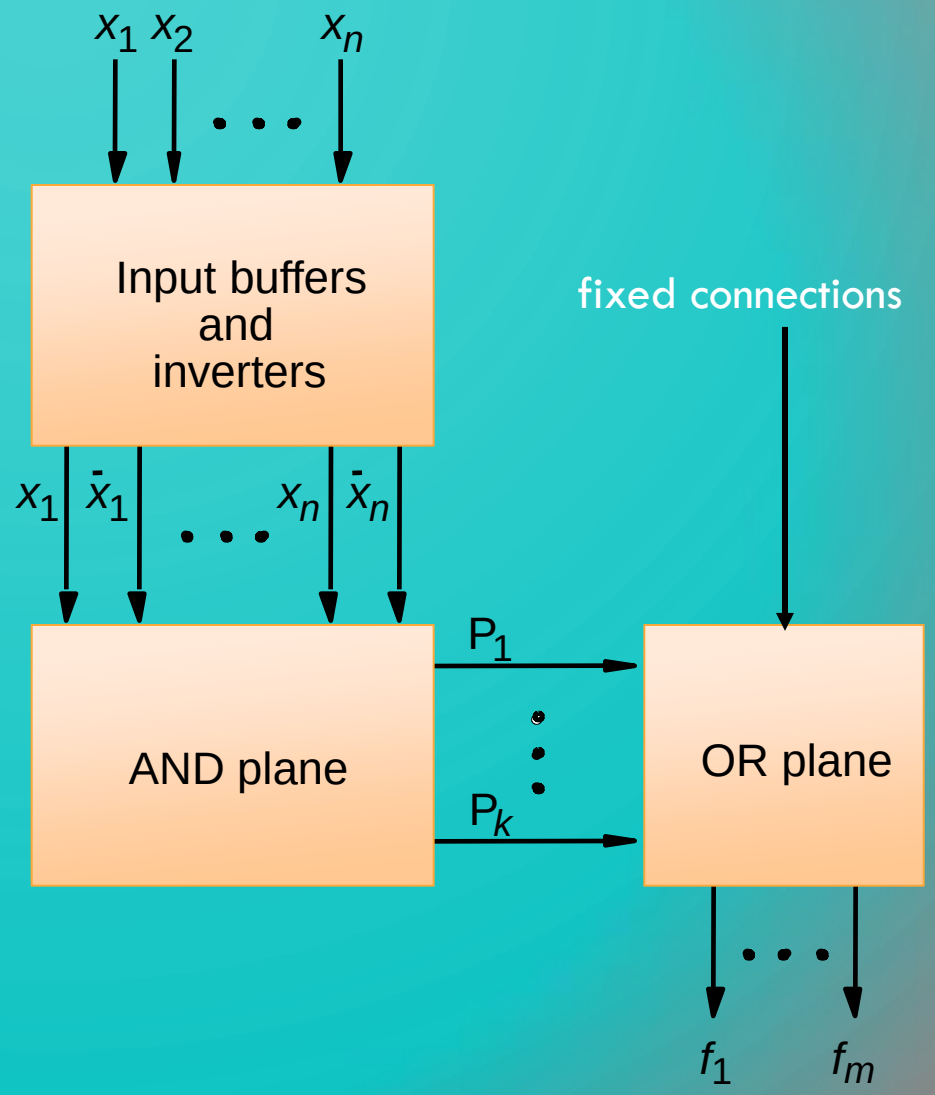
- Gate Level Version of PLA

$f_1 = x_1x_2+x_1x_3'+x_1'x_2'x_3$

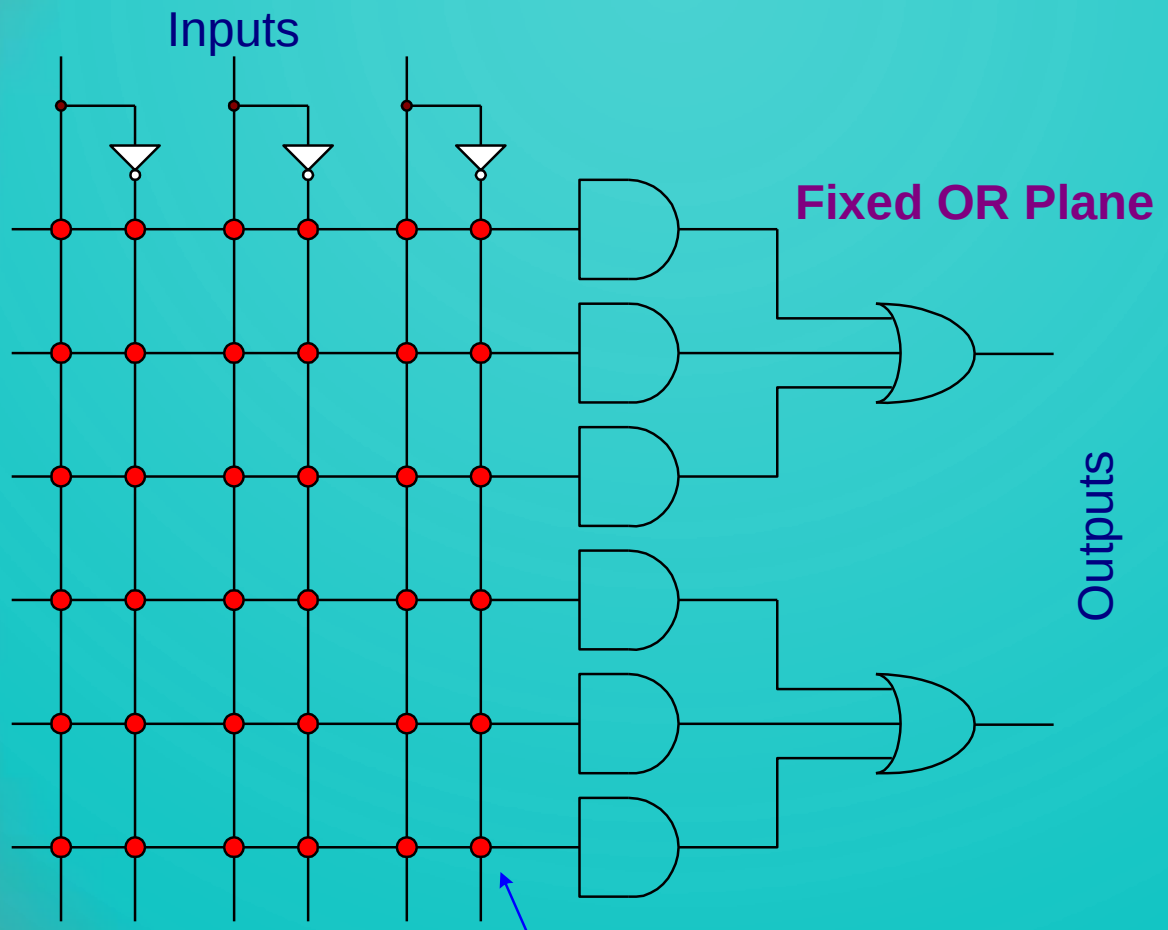
$f_2 = x_1x_2+x_1'x_2'x_3+x_1x_3$



- Programmable Array Logic (PAL)
  - The connections in the AND plane are programmable
  - The connections in the OR plane are NOT programmable

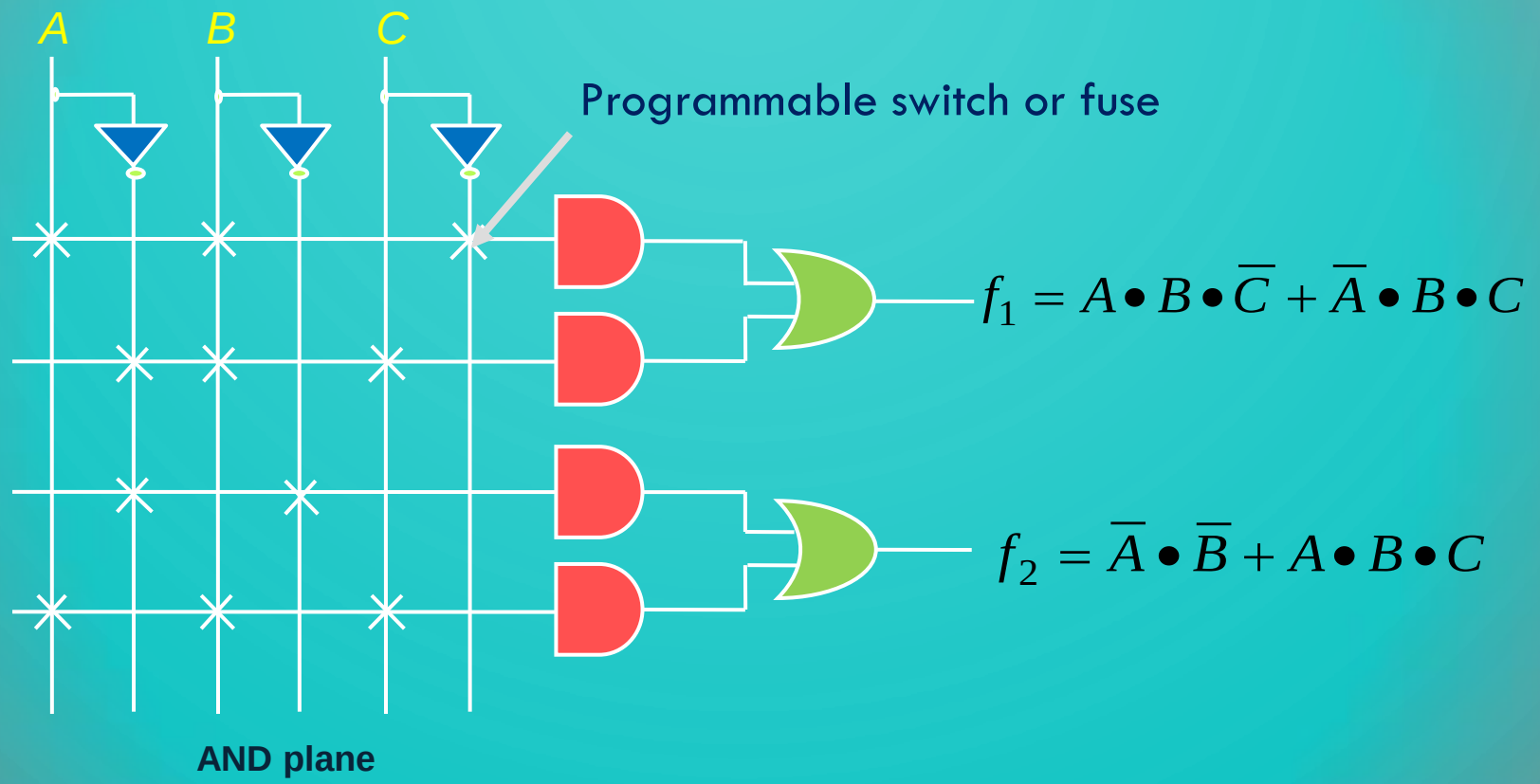


PAL

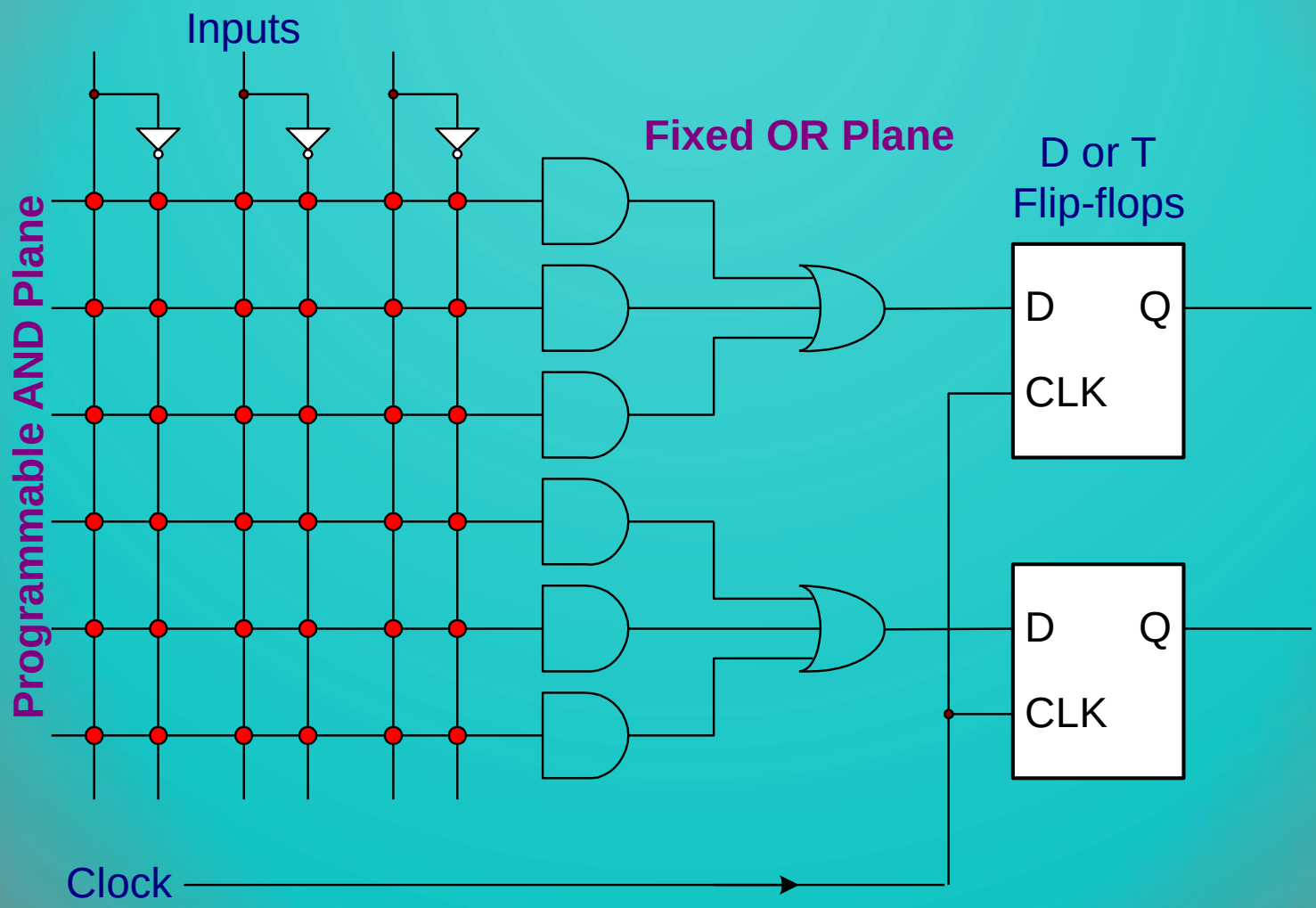


The PAL Architecture

PAL


A
B
C
Programmable switch or fuse
$f_1 = A \bullet B \bullet \overline{C} + \overline{A} \bullet B \bullet C$
$f_2 = \overline{A} \bullet \overline{B} + A \bullet B \bullet C$
AND plane

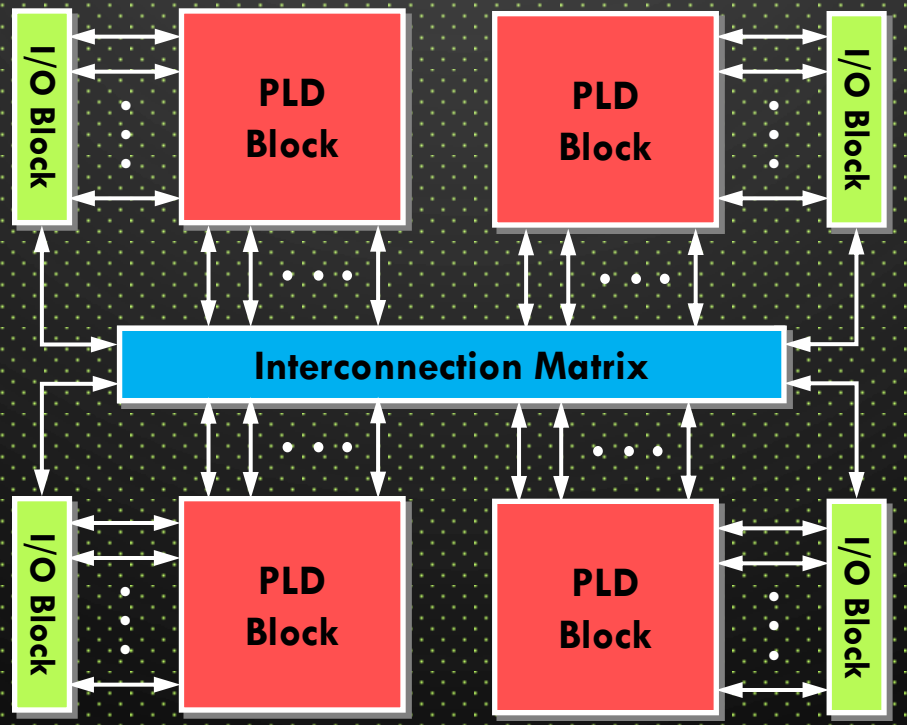
PAL



The Registered PAL Architecture

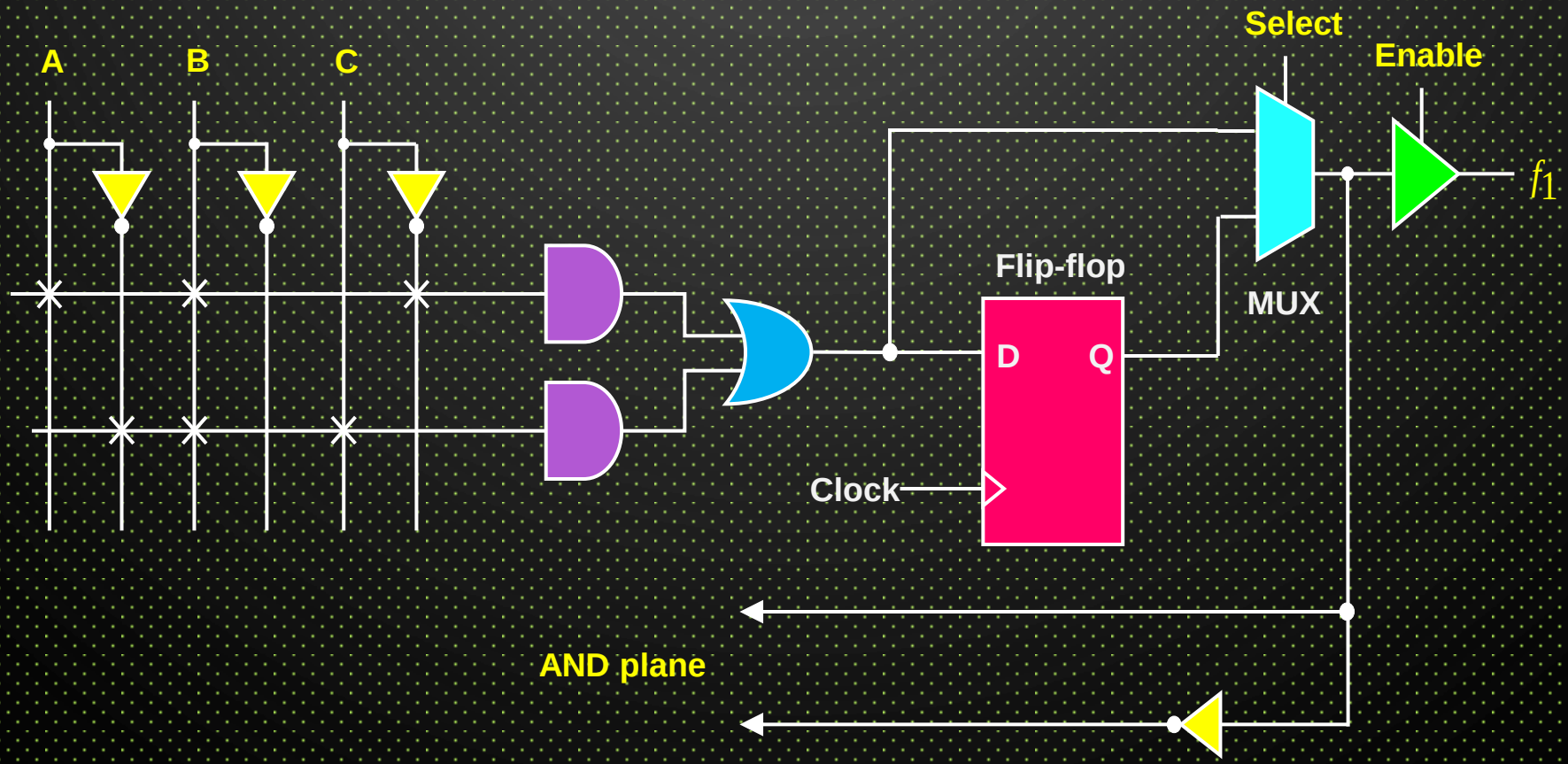
CPLD STRUCTURE

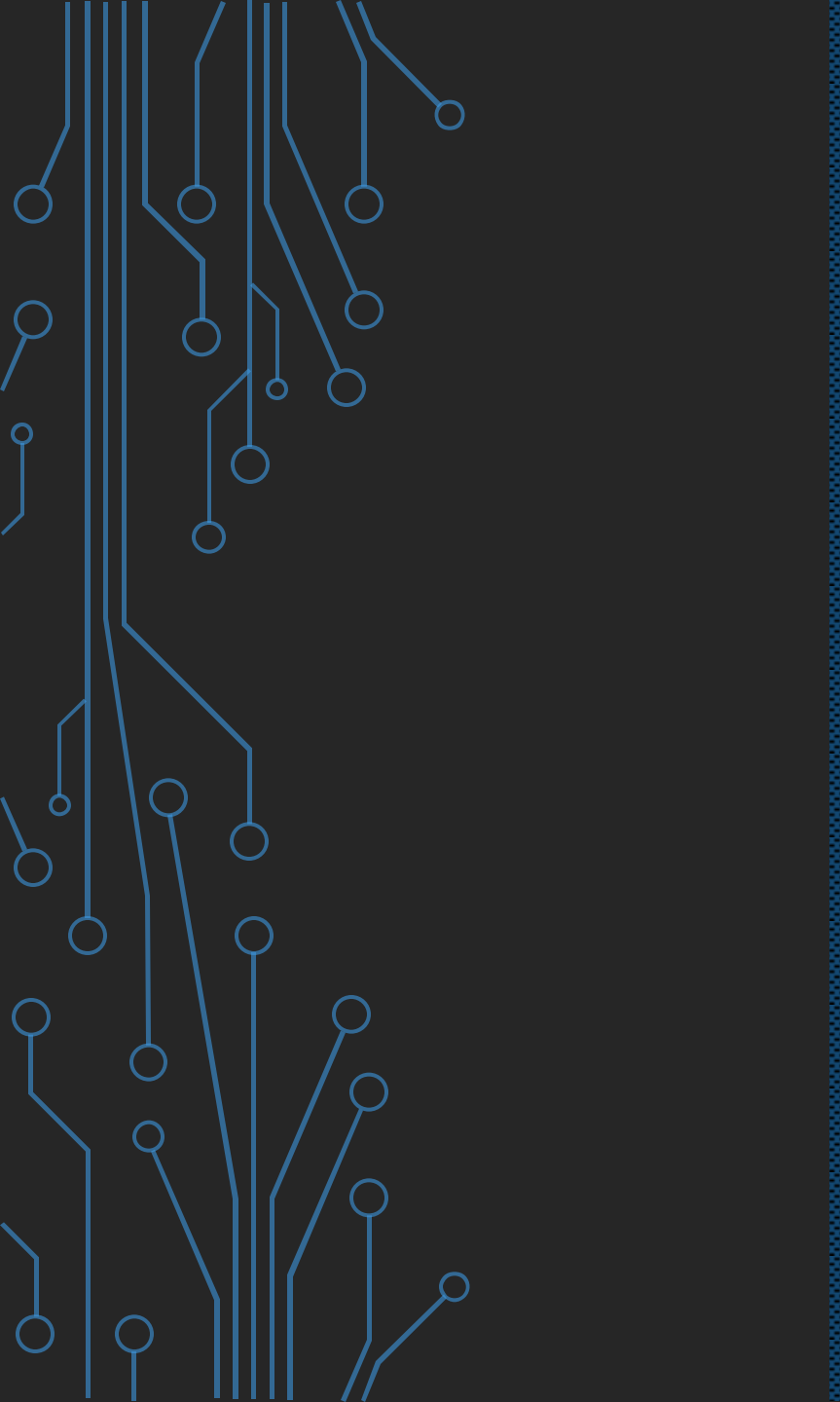
Integration of several PLD blocks with a programmable interconnect on a single chip



PLD - MACROCELL

Can implement combinational or sequential logic





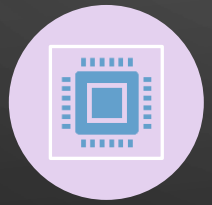
FPGA ARCHITECTURE

FPGA - GENERIC STRUCTURE

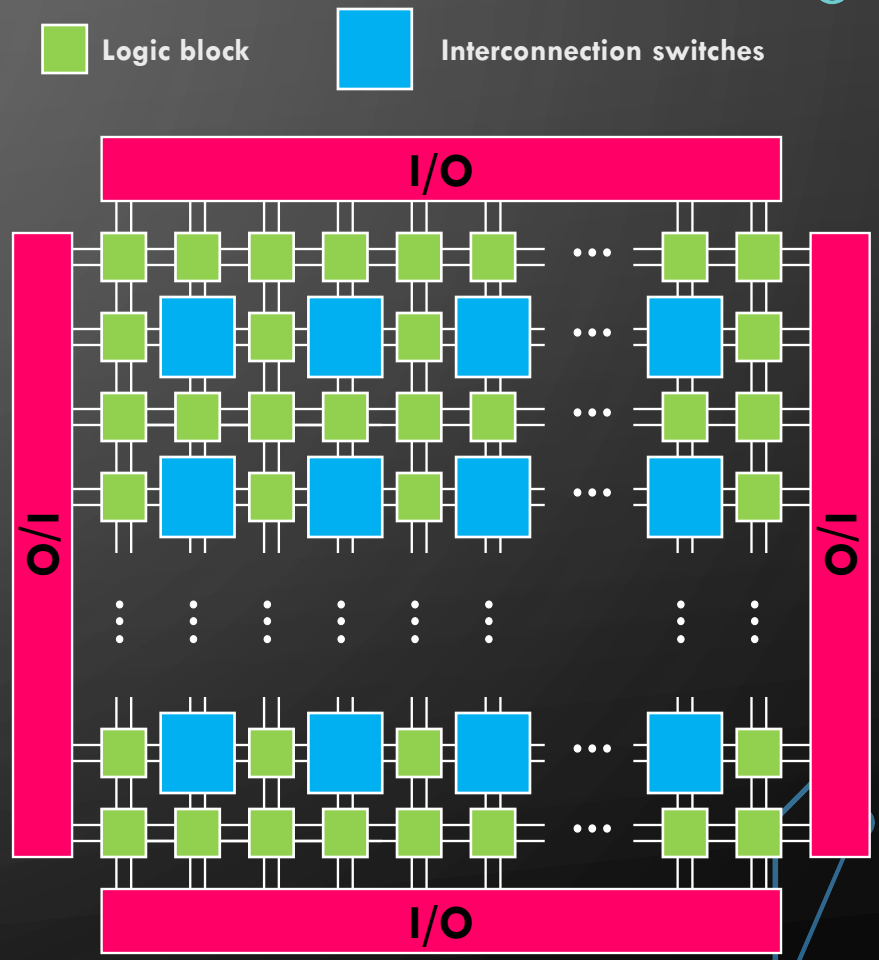
FPGA BUILDING BLOCKS:

PROGRAMMABLE LOGIC BLOCKS IMPLEMENT COMBINATORIAL AND SEQUENTIAL LOGIC

PROGRAMMABLE INTERCONNECT WIRES TO CONNECT INPUTS AND OUTPUTS TO LOGIC BLOCKS



PROGRAMMABLE I/O BLOCKS SPECIAL LOGIC BLOCKS AT THE PERIPHERY OF DEVICE FOR EXTERNAL CONNECTIONS



OTHER FPGA BUILDING BLOCKS

Clock distribution

Embedded memory blocks

Special purpose blocks:

DSP blocks:

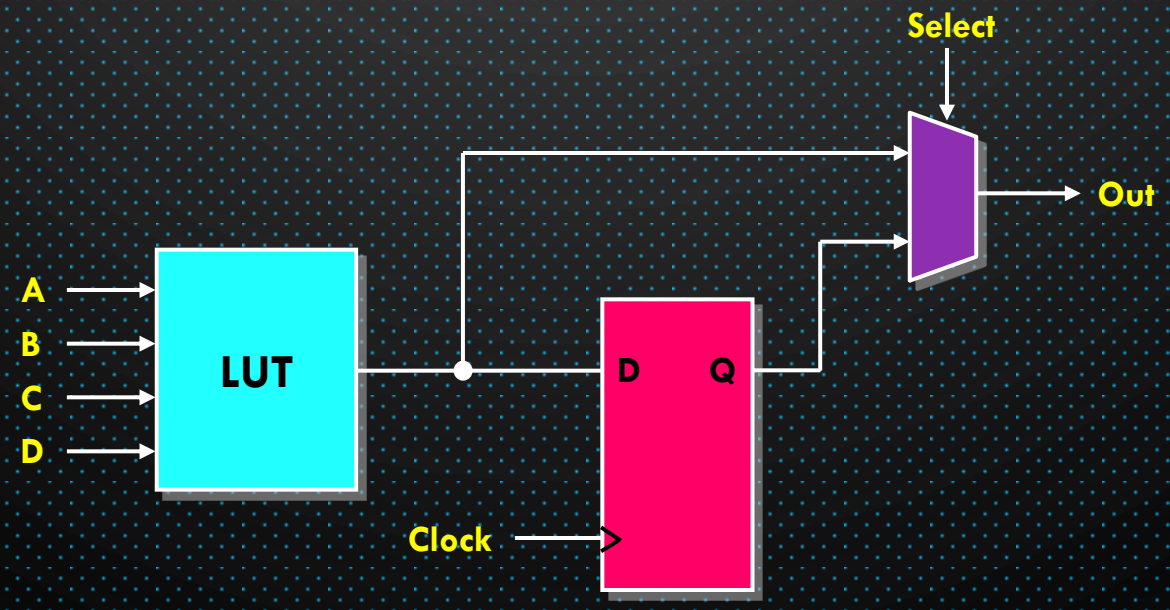
- Hardware multipliers, adders and registers

Embedded microprocessors/microcontrollers

High-speed serial transceivers

FPGA - BASIC LOGIC ELEMENT

- LUT to implement combinatorial logic
- Register for sequential circuits
- Additional logic (not shown):
  - Carry logic for arithmetic functions
  - Expansion logic for functions requiring more than 4 inputs

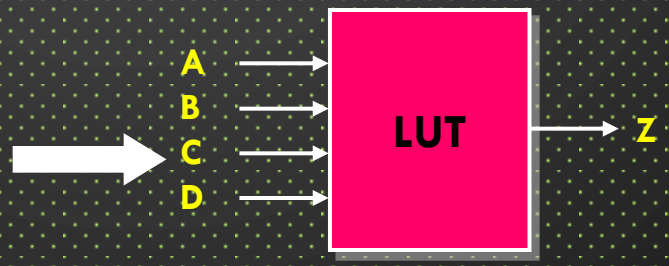


LOOK-UP TABLES (LUT)

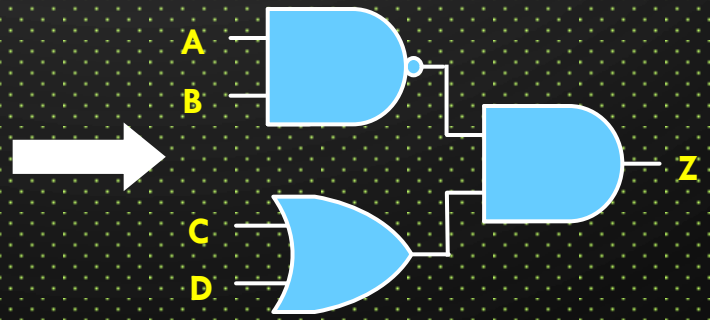
- Look-up table with N-inputs can be used to implement any combinatorial function of N inputs
- LUT is programmed with the truth-table

A	B	C	D	Z
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**1**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**1**
0	1	1	0	**1**
0	1	1	1	**1**
1	0	0	0	**0**
1	0	0	1	**1**
1	0	1	0	**1**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**0**
1	1	1	0	**0**

Truth-table



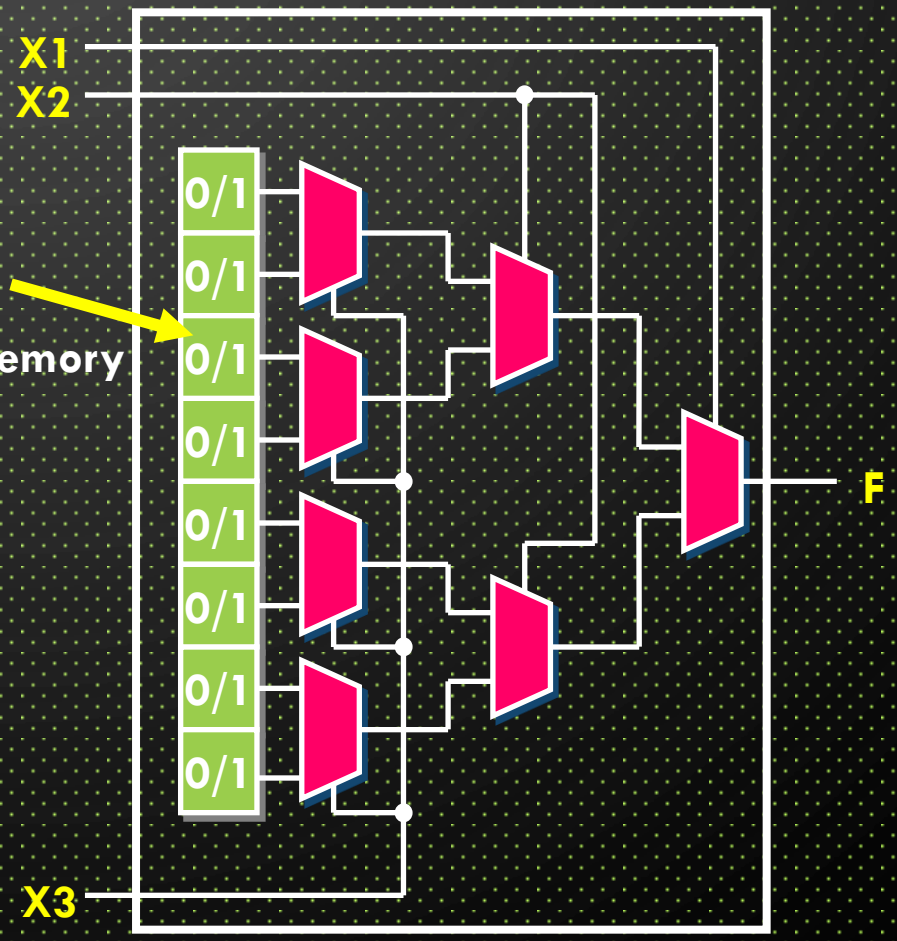
LUT implementation



Gate implementation

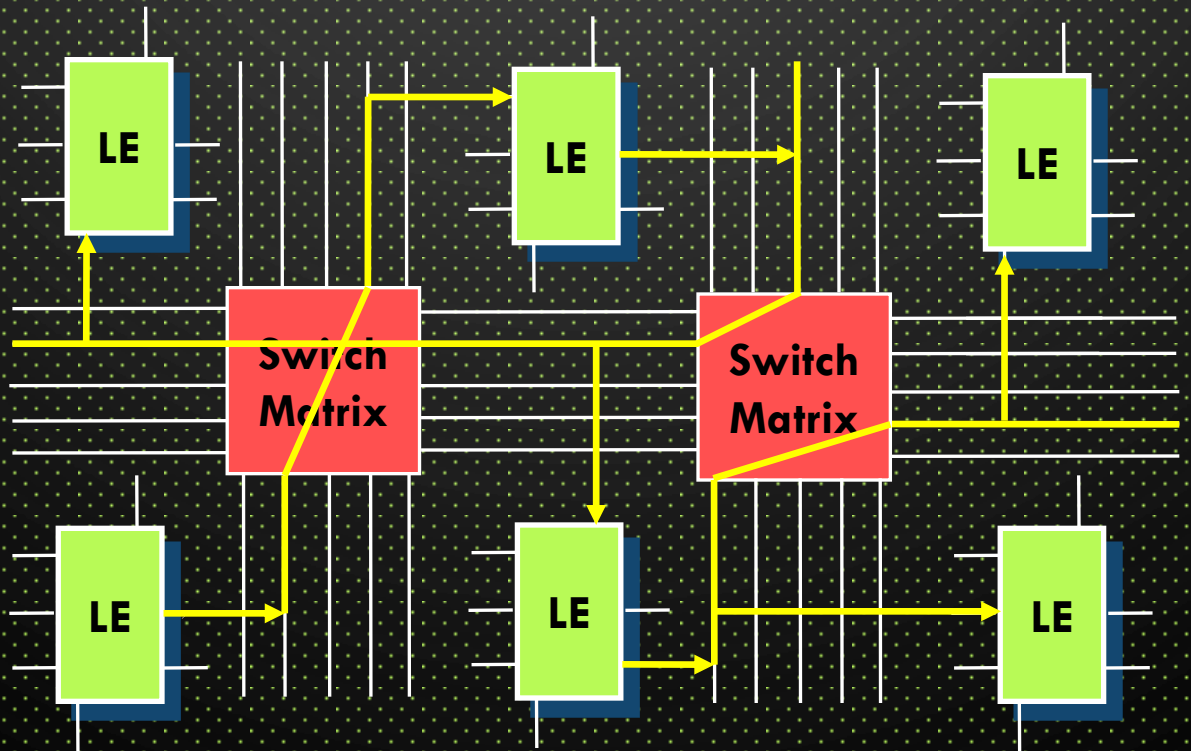
LUT IMPLEMENTATION

- Example: 3-input LUT
- Based on multiplexers (pass transistors)
- LUT entries stored in configuration memory cells



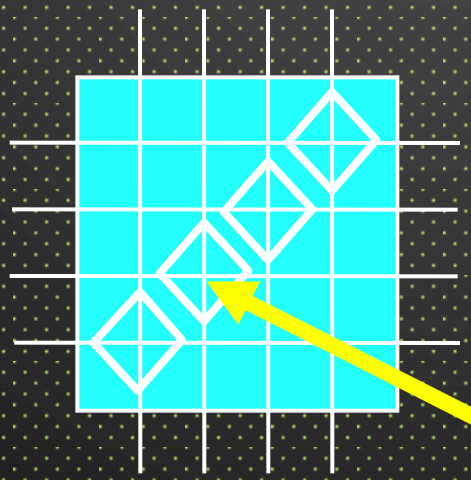
PROGRAMMABLE INTERCONNECT

- Interconnect hierarchy (not shown)
  - Fast local interconnect
  - Horizontal and vertical lines of various lengths

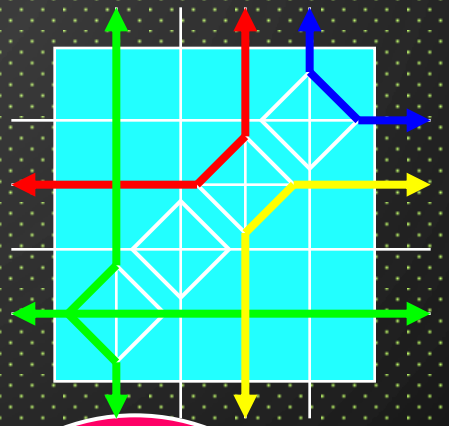


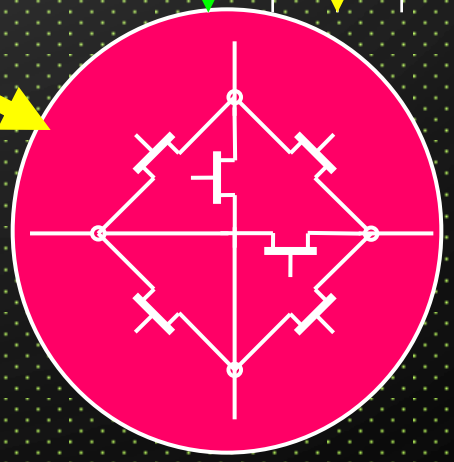
SWITCH MATRIX OPERATION

Before Programming



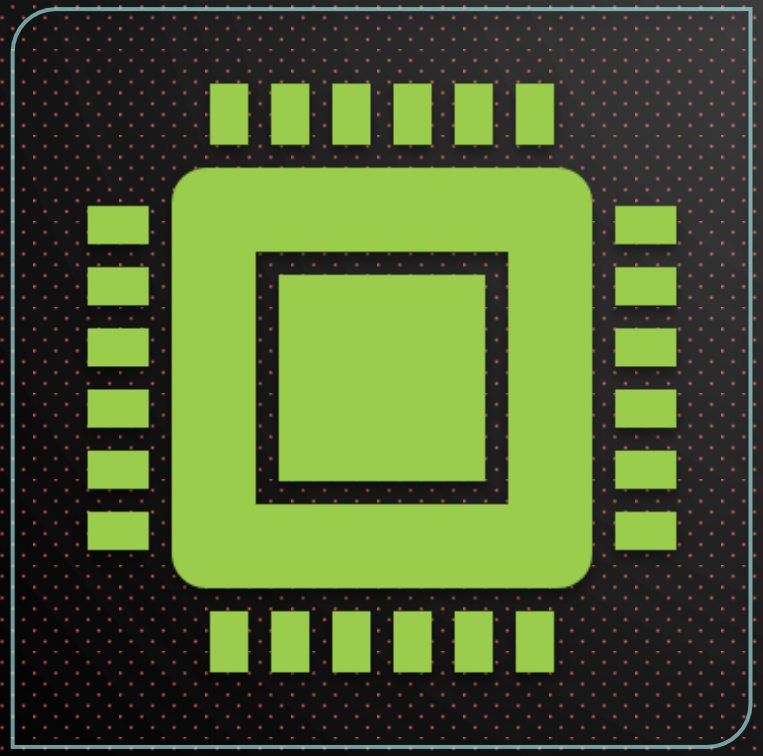
After Programming





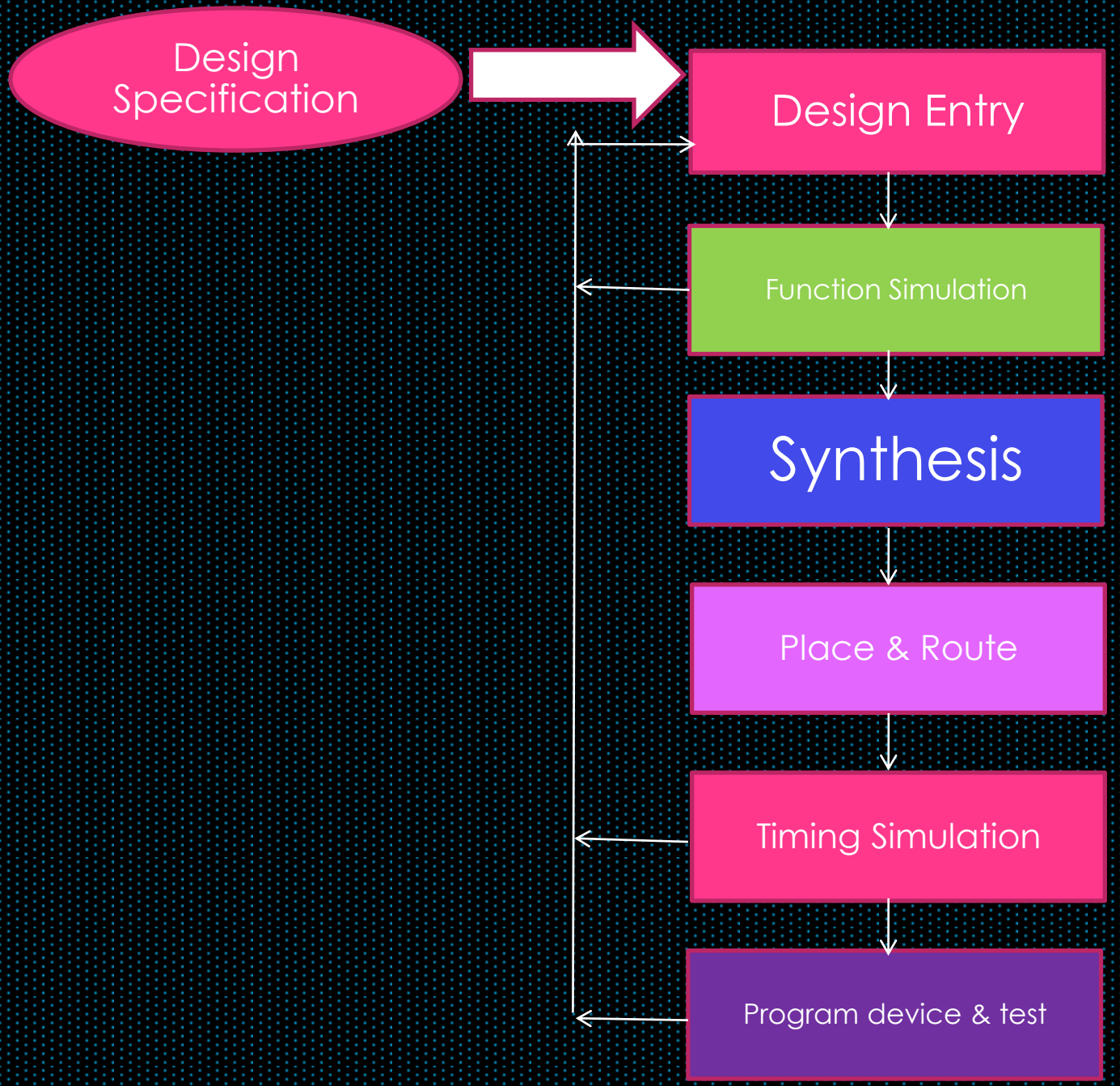
- 6 pass transistors per switch matrix interconnect point
- Pass transistors act as programmable switches
- Pass transistor gates are driven by configuration memory cells

FPGA VENDORS



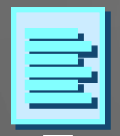
- Altera
- Xilinx
- Actel
- Lattice
- QuickLogic

FPGA DESIGN FLOW



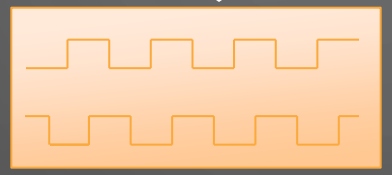
FPGA DESIGN FLOW

Design Specification



Design Entry/RTL Coding

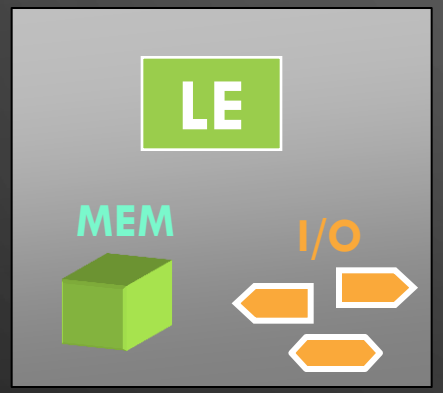
Behavioral or Structural Description of Design



RTL Simulation

- Functional Simulation
- Verify Logic Model & Data Flow

(No Timing Delays)



Synthesis

- Translate Design into Device Specific Primitives
- Optimization to Meet Required Area & Performance Constraints

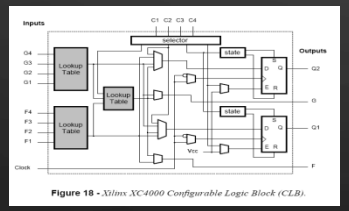
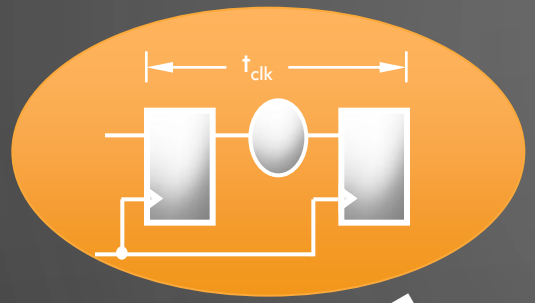


Figure 18 - *Xilinx XC4000 Configurable Logic Block (CLB).*

Place & Route

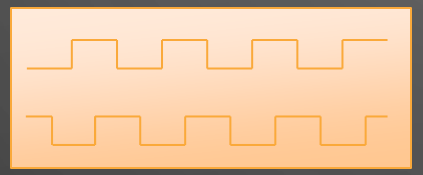
- Map Primitives to Specific Locations inside Target Technology with Reference to Area &
- Performance Constraints
- Specify Routing Resources to Be Used

FPGA DESIGN FLOW



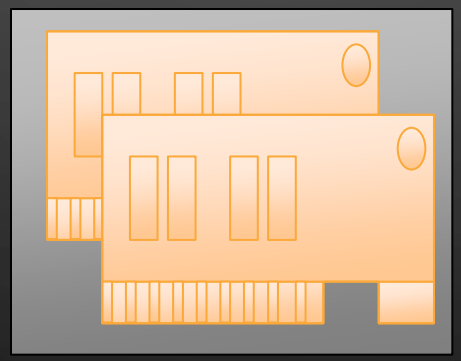
Timing Analysis

- Verify Performance Specifications Were Met
- Static Timing Analysis



Gate Level Simulation

- Timing Simulation
- Verify Design Will Work in Target Technology



Program & Test

- Program & Test Device on Board