Computer Aided Design (CAD)

Lecture 13

Introduction to FPGA

Dr.Eng. Basem ElHalawany

Schedule (2nd Update 2-12)

Topics	Estimated Duration (# Lectures)
Introduction	1
Introduction to Matlab Environment	1
Matlab Programing (m-files) (1)	5
Modeling using Matlab Simulink Tool (1)	3
Midterm	7th Week
Communication Systems Simulation (Applications) (2)	2
Introduction to FPGA	1
Matlab and Simulink Projects Delivery	1
VHDL Modeling Language	2
VHDL Application	1
Course Closeout / Feedback/ project (s) Delivery	1

Introduction

✓ Three basic kinds of devices exists in digital electronic systems:

1. Memory,
2. Microprocessor,
3. Logic devices

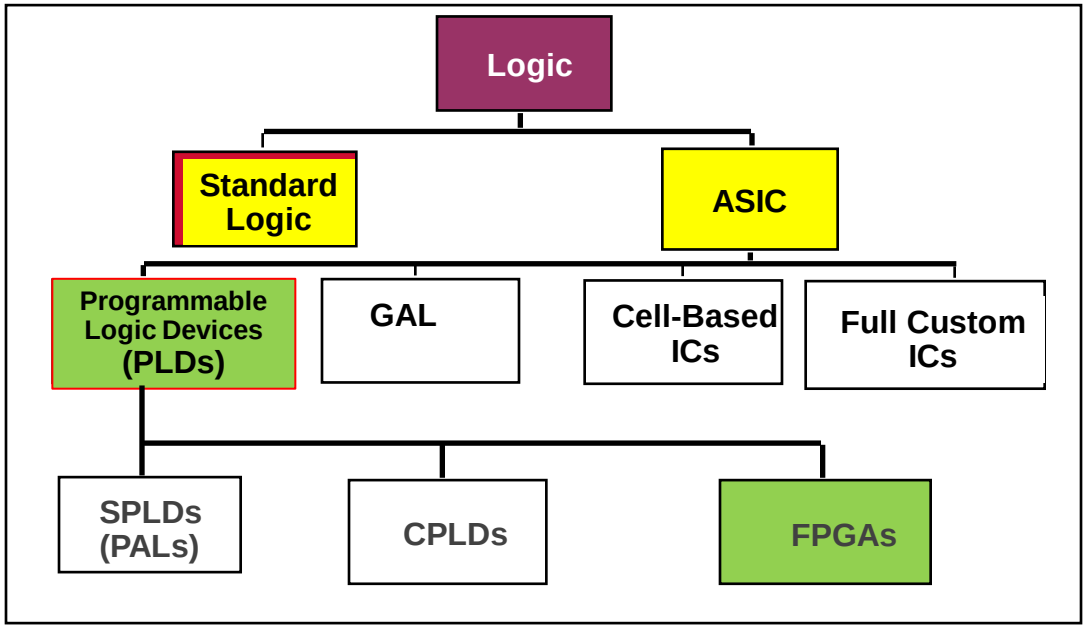
- Memory devices store information

- Microprocessors execute software instructions to perform a wide variety of tasks such as running a word processing Program or video game.

- Logic devices provide specific logic functions which forms the basis of many operations including:
  - device-to-device interfacing,
  - data communication,
  - signal processing,
  - data display,
  - timing and control operations,

Logic Devices

- A logic device is one which can perform any logic function
- Logic devices are broadly classified into:

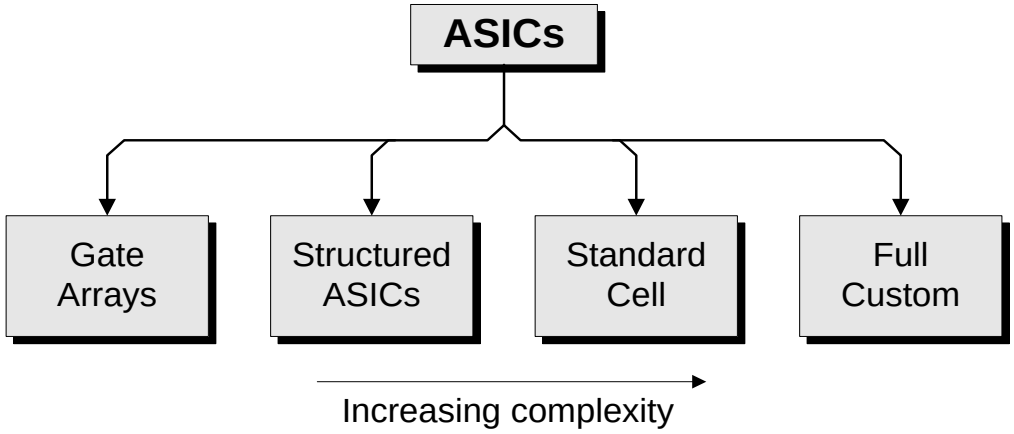


- ASIC = Application-Specific IC
- SPLD = Simple Programmable Logic Devices (PLD)
- PAL = Programmable Array of Logic
- CPLD = Complex PLD
- GAL = Gate Array logic
- FPGA = Field Programmable Gate Array

ASIC versus Programmable Logic

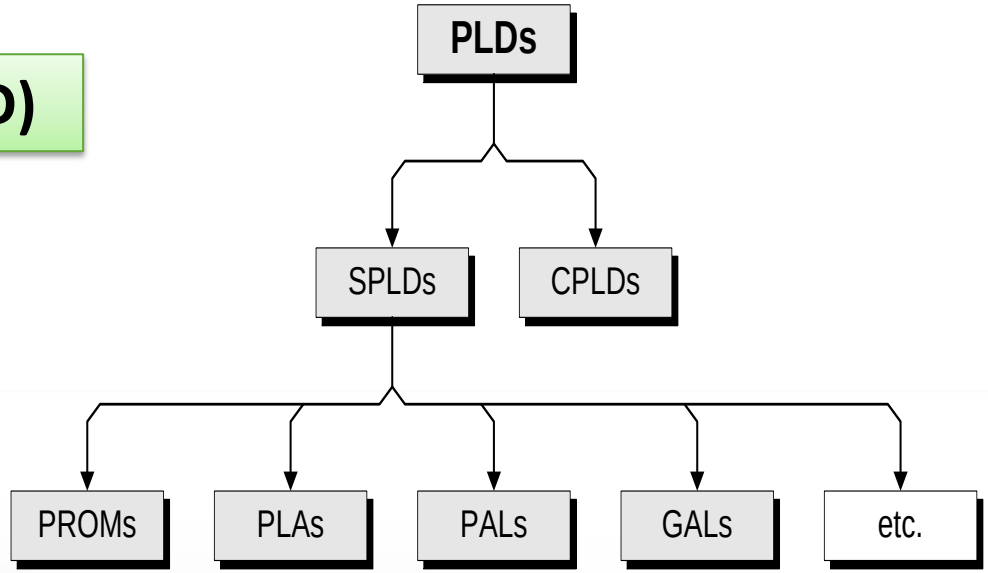
ASIC (Fixed Logic device)

- The circuits in a fixed logic device are permanent,
- They perform one function or set of functions - once manufactured, they cannot be changed.



Programmable Logic devices (PLD)

- Programmable devices offers a wide range of logic features and voltage characteristics .
- They can be changed at any time to perform various logic functions according to the application.

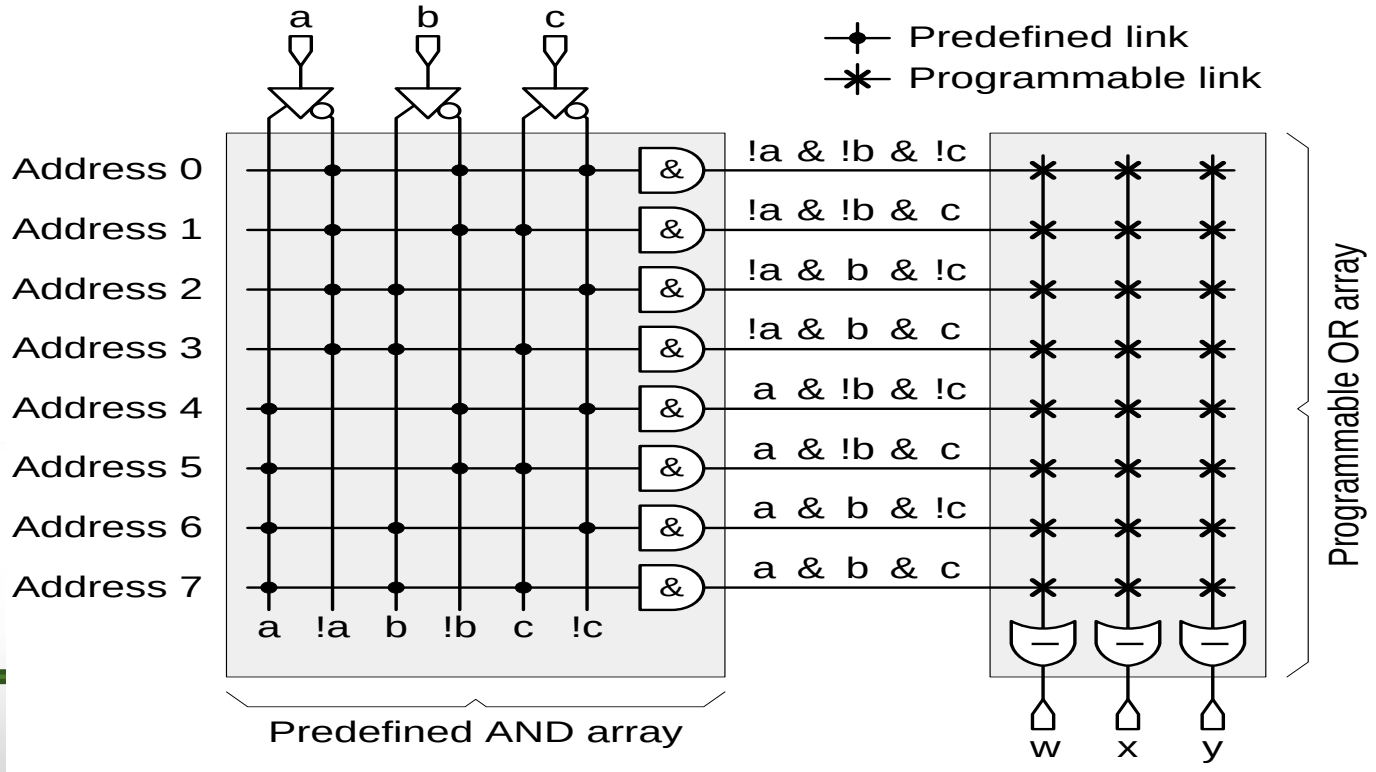


Programmable Logic devices (PLD)

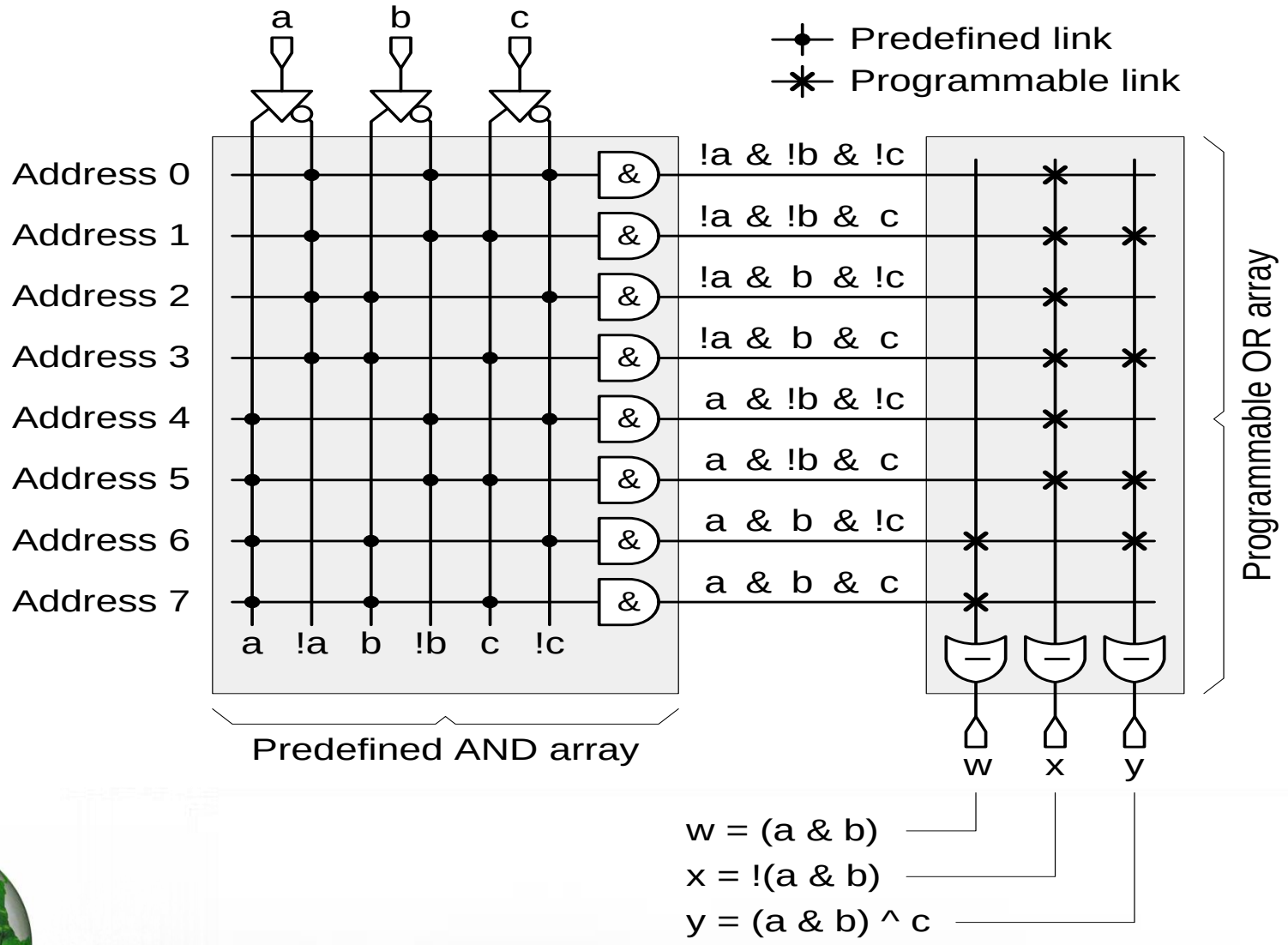
- The PLD consists of internal logic gates and interconnects (switches).
- These gates can be connected to obtain the required logic Configuration.
- The configuration of the internal logic is done by the user according to the required logic functions.
- PROM,EPROM,PAL, andPLA are examples of PLD

1-PROM (Programmable ROM)

- Pre-defined AND array,
- Programmable OR array

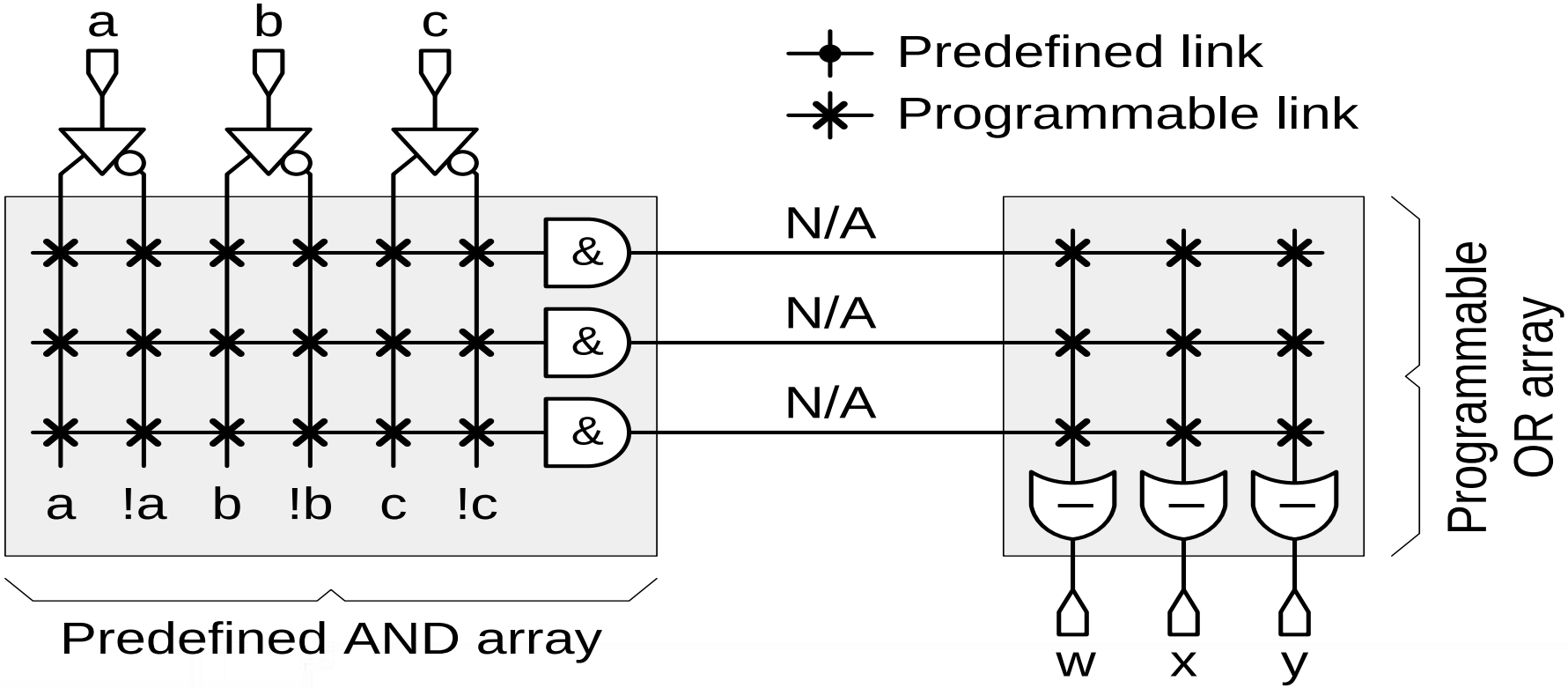


1-PROM (Programmable ROM)



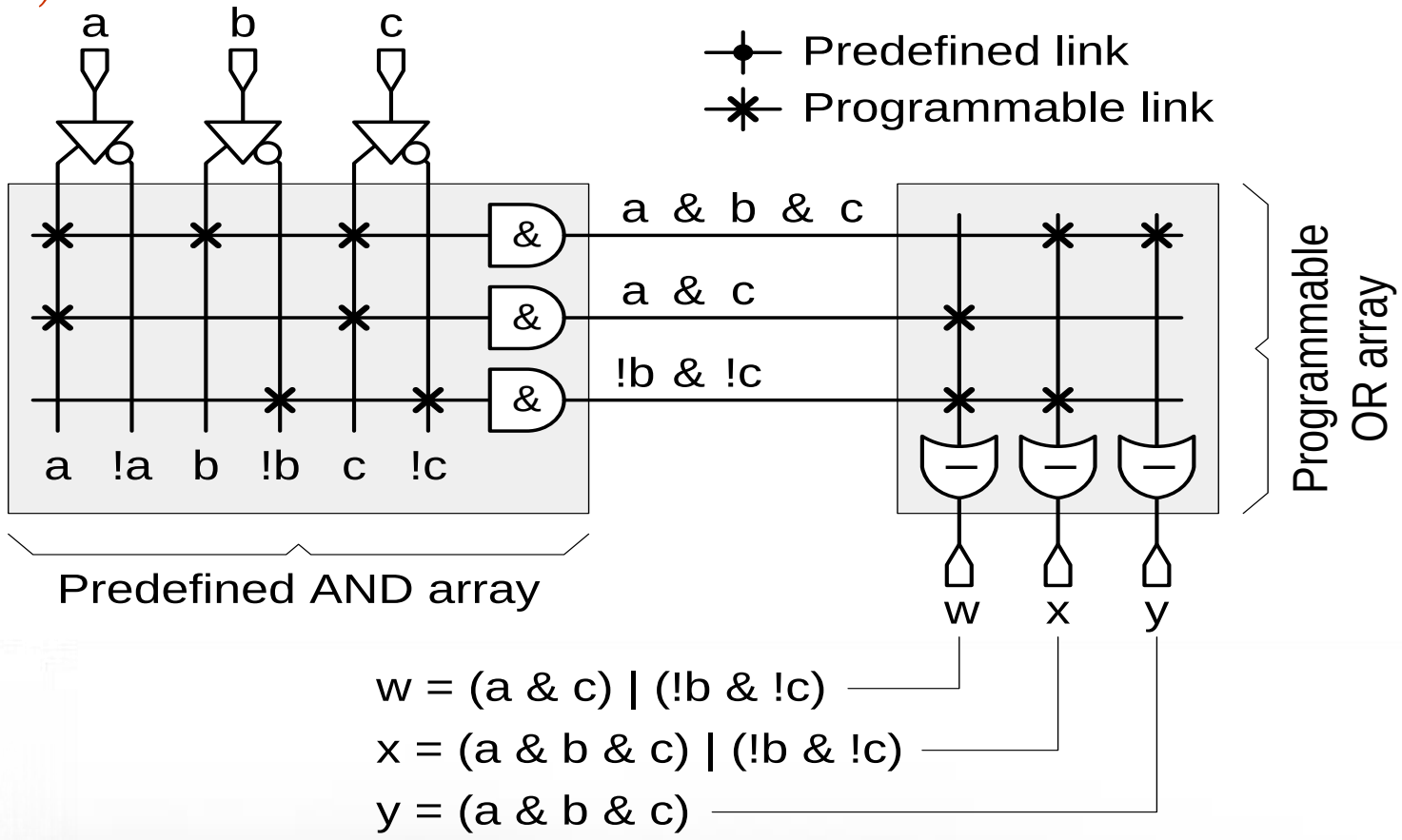
2-PLA (Programmable Logic Arrays)

- Programmable AND array, programmable OR array



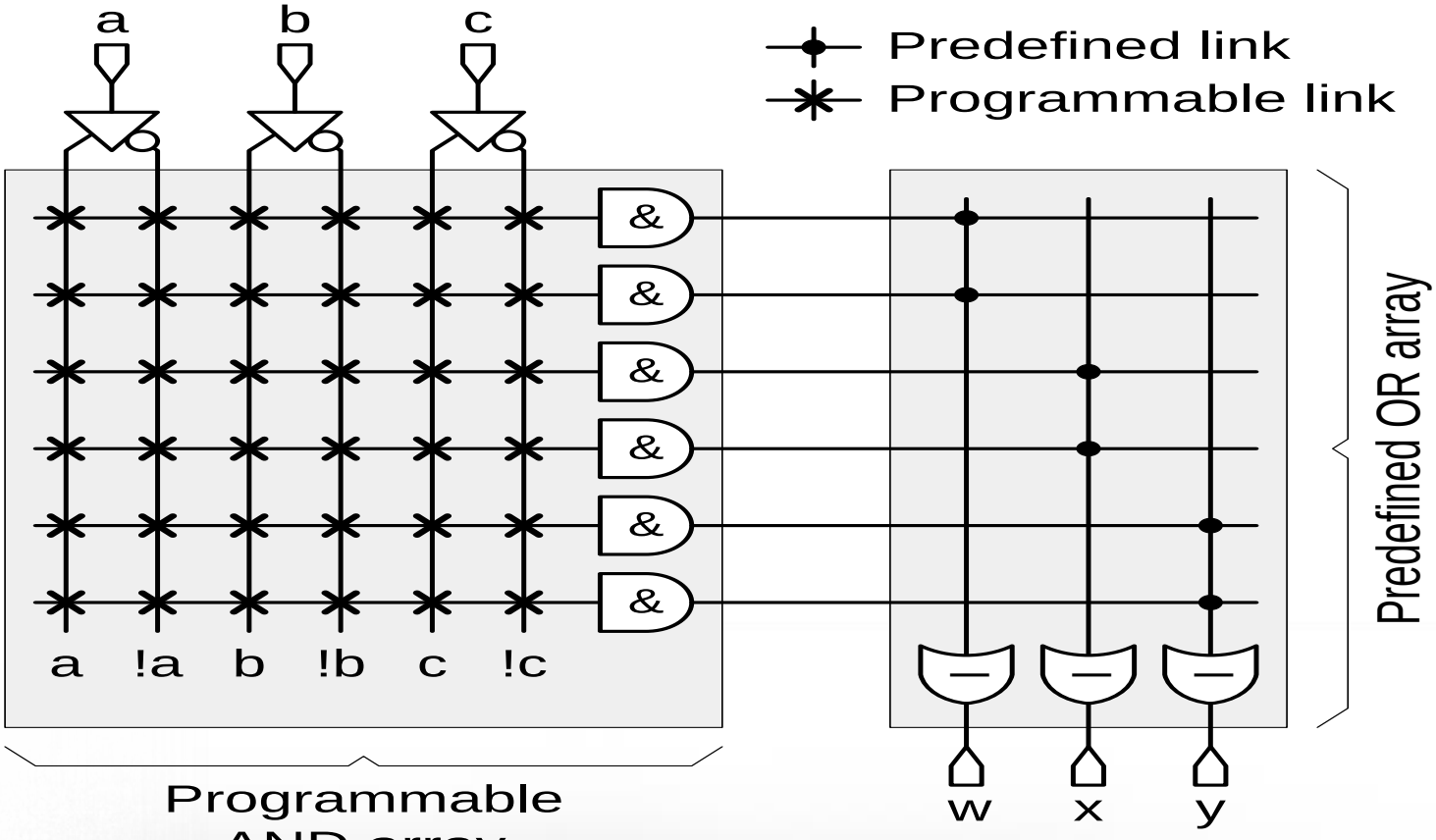
2- PLA (Programmable Logic Arrays)

- Programmable AND array, programmable OR array
- The architecture of PLA is more flexible.
- There is a speed problem due to the number of programmable switches in (AND, OR) Sections



3- PAL (Programmable Array Logic)

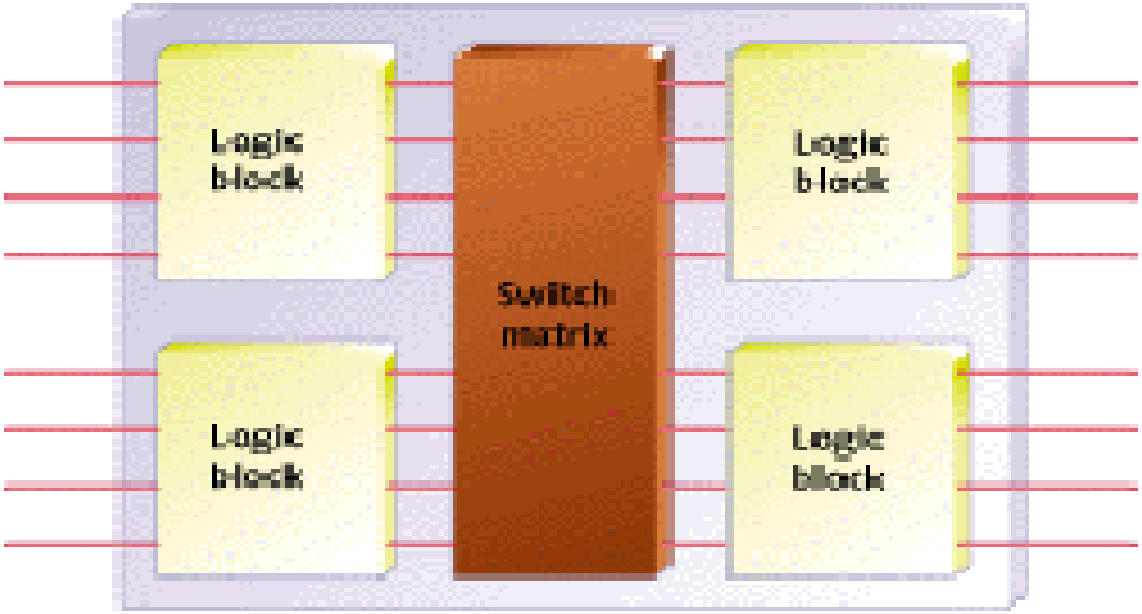
1. Exactly opposite to PROM
2. Programmable AND arrays, predefined OR arrays
3. Address speed issues in PLAs



4- CPLDs (Complex PLDs)

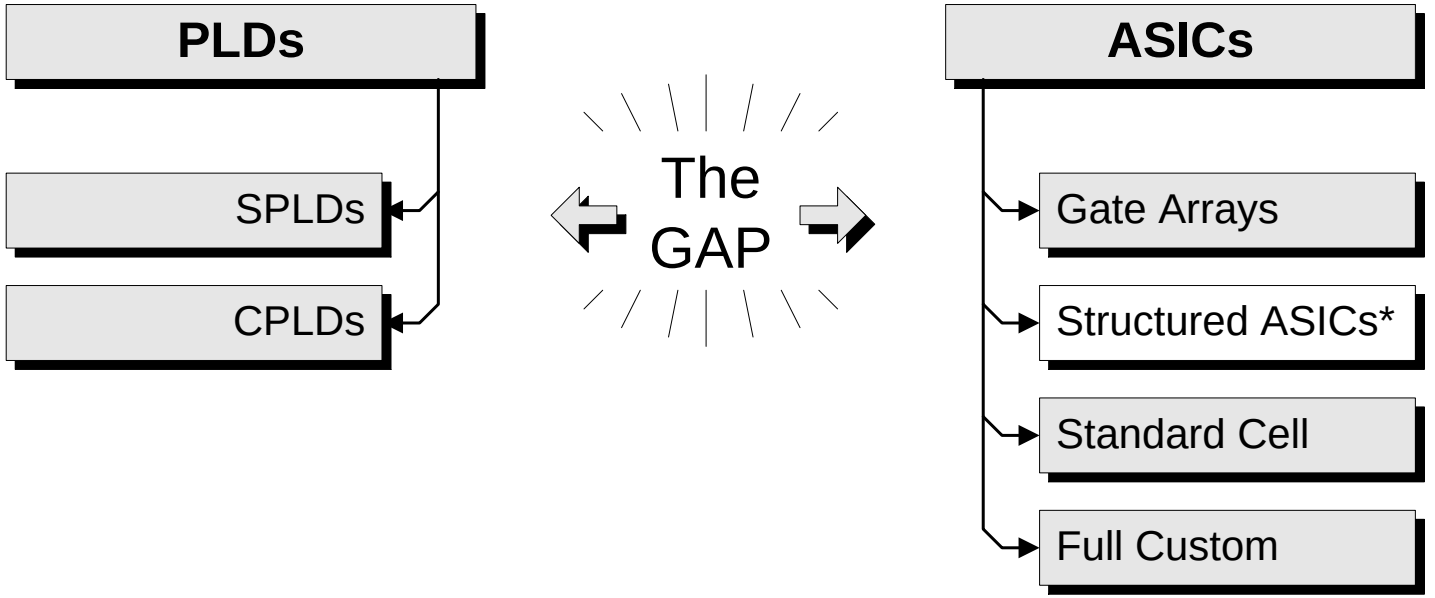
To retain the advantages and to overcome the disadvantages of PLAS and PALS , a new devices is introduced (known as CPLDs)

- CPLD contains multiple combination of PLAs and PALs
- A simple architecture of CPLD is shown below.



The Gap between PLDs & ASICs

1. PLDs : Programmable but less complexity
2. ASICs: High complexity but no programmability



*Not available circa early 1980s

- The gap has been filled with FPGA

ASICs versus FPGA)

- Manufacturing cycle for ASIC is very costly, lengthy and engages lots of manpower in fabrication.
- Mistakes not detected at design time have large impact on development time and cost of ASIC
- FPGAs are re-programmable chips, which is perfect for rapid prototyping of digital circuits.
- Easy upgrades like in case of software applications in FPGA

ASICs versus FPGA

ASIC

- Designs must be sent for expensive and time consuming fabrication in semiconductor foundry
- designed all the way from behavioral description to physical layout
- Higher performance
- Low Cost (only in mass-volume)
- Low power consumption

FPGA

- bought off the shelf and reconfigured by designers themselves
- No physical layout design;
- Design ends with a bit-stream used to configure a device
- Low development cost
- Short time to market
- Reconfigurable
- it is possible to implement a complex logic design in a manner which is easy to test, debug and even change using software the behavior of the design

CPLDs versus FPGA

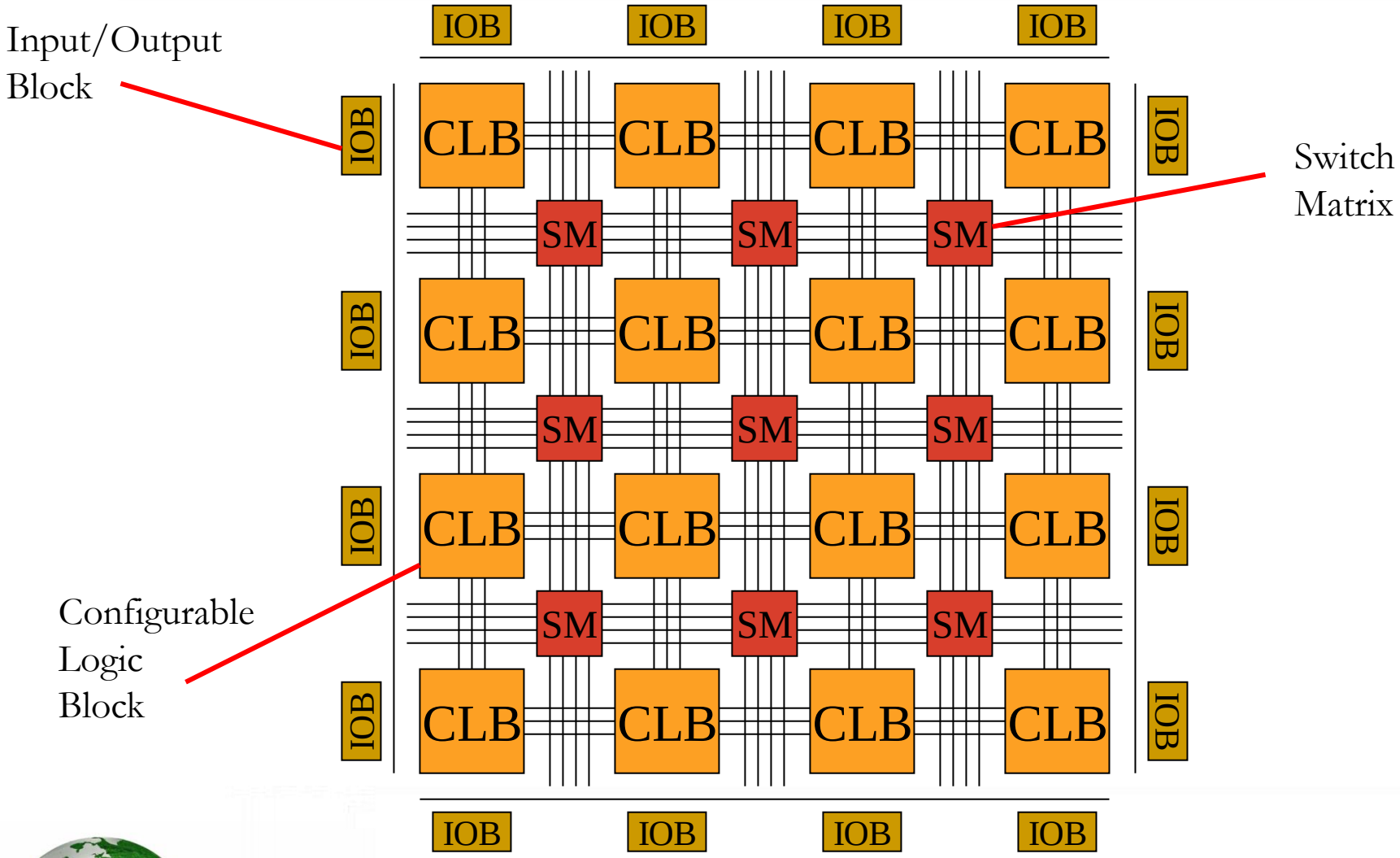
CPLDs

- CPLDs are "coarse-grain" devices. They contain relatively few (a few 100's max) large blocks of logic with flip-flops.
- CPLDs are EEPROM based. They are active at power-up (i.e. as long as they've been programmed at least once
- CPLDs can contain small designs only.

FPGA

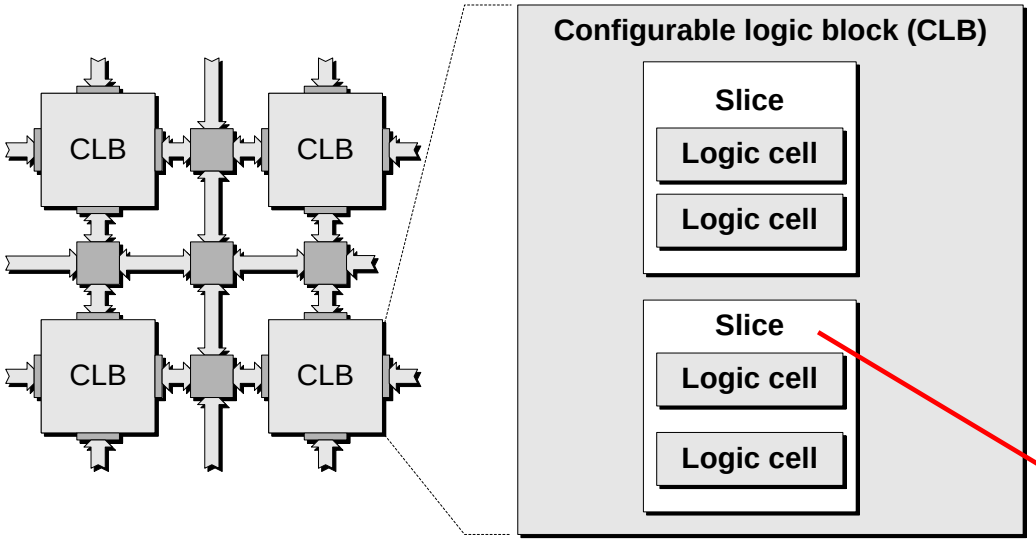
- FPGAs are "fine-grain" devices. That means that they contain a lot (up to 100000) of tiny blocks of logic.
- FPGAs are RAM based. They need to be "downloaded" (configured) at each power-up.
- FPGAs can contain very large digital designs
- FPGAs have special routing resources to implement efficiently binary counters and arithmetic functions (adders, comparators.).

Field Programmable Gate Array (FPGA) Structure

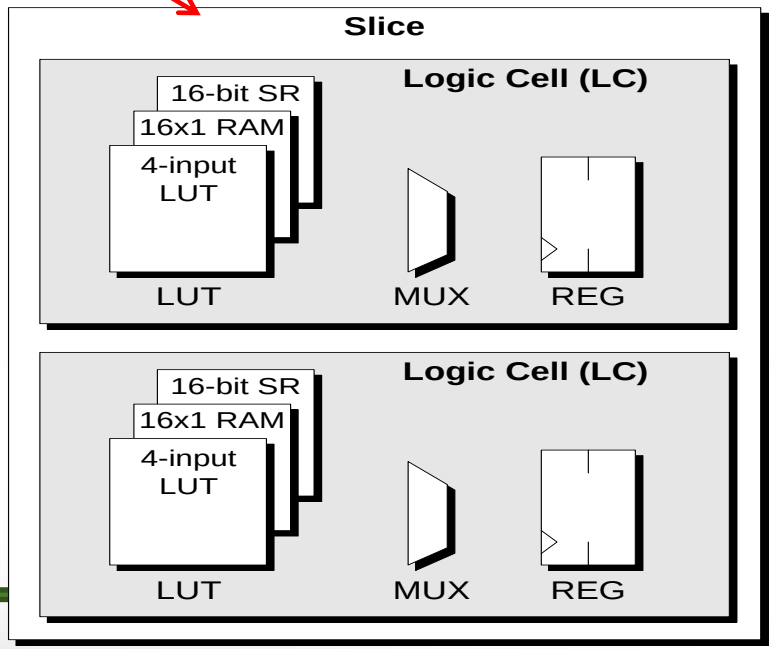


- User can configure :
  - Intersections between the logic blocks
  - The function of each block

Configurable Logic Block (CLB)



1. 2-D array of CLBs
2. Each CLB consists of 2 slices



1. Each slice has 2 logic Cell (LC)
2. Each Logic cell comprises a LUT, and some additional components (i-e Multiplexers, Registers)
3. The internal construction differs from vendor to vendor

Major FPGA Vendors

Company	General Architecture	Logic Block Type	Programming Technology
Xilinx	Symmetrical Array	Look-up Table	Static RAM
Actel	Row-based	Multiplexer-Based	Anti-Fuse
Altera	Hierarchical PLD	PLD Block	EPROM/SRAM
Plessey	Sea-of-gates	NAND-gate	Static RAM
Plus	Hierarchical PLD	PLD Block	EPROM
AMD	Hierarchial PLD	PLD Block	EEPROM
Quicklogic	Symmetrical Array	Multiplexer-Based	Anti-Fuse
Algotronix	Sea-of-gates	Multiplexers and Basic Gates	Static RAM
Concurrent	Sea-of-gates	Multiplexers and Basic Gates	Static RAM
Crosspoint	Row-based	Transistor Pairs and Multiplexers	Anti-Fuse

FPGA Design and Programming

- To define the behavior of the FPGA the user provides a hardware description language (HDL) or a schematic design.
- Then, using an electronic design automation tool, a technology-mapped net list is generated.
- The netlist can then be fitted to the actual FPGA architecture using a process called place-and-route.
- The user will validate the map, place and route results via timing analysis, simulation, and other verification methodologies.
- Once the design and validation process is complete, the binary file generated used to configure the FPGA.

HDL Languages

- Verilog
- VHDL