Introduction to FPGAs

IAS0600 Digital Systems Design with VHDL

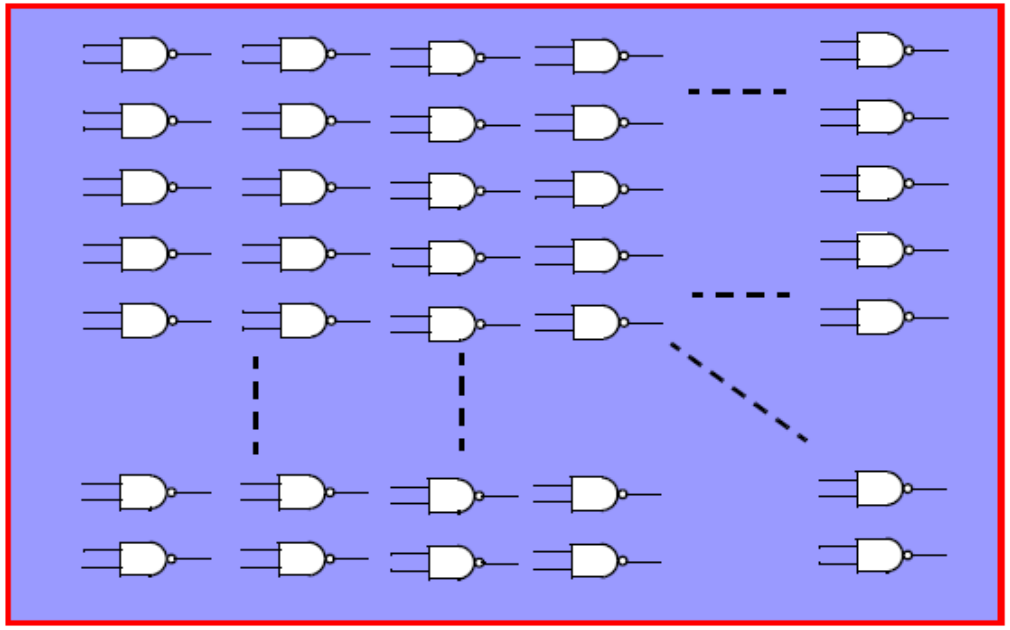
Jaan Raik, Natalia Cherezova

Introduction to FPGAs

- History of FPGAs
- Reconfigurable hardware vs integrated circuits
- FPGA applications
- Inner architecture of FPGAs
- Design flow for FPGA

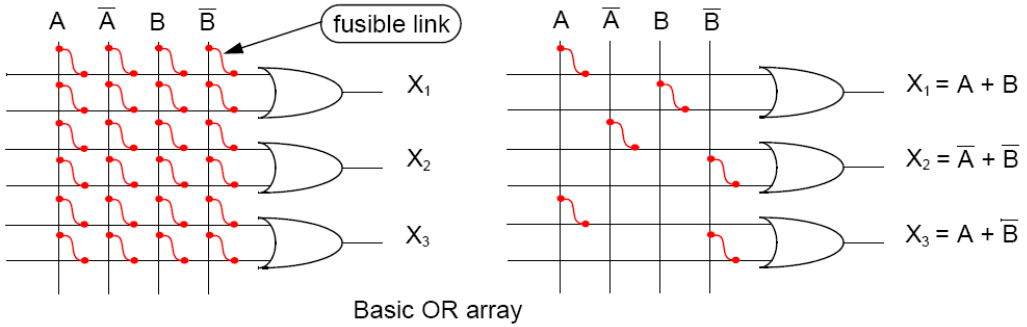
History. Gate Arrays

- In the beginning, digital design was done with the ’74 series of chips.
- Some people would design their own chips based on Gate Arrays, which were nothing else than an array of NAND gates:

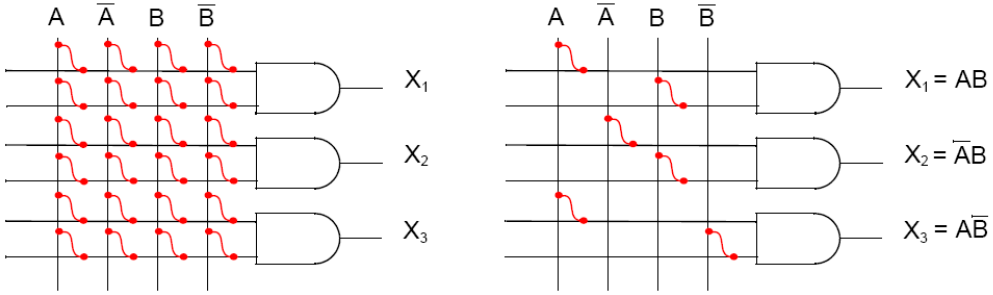


History. PLA/PLD

- The first programmable chips were PLAs (Programmable Logic Arrays): two level structures of AND and OR gates with user programmable connections.
- Programmable Array Logic devices were an improvement in structure and cost over PLAs. Today such devices are generically called Programmable Logic Devices (PLDs).



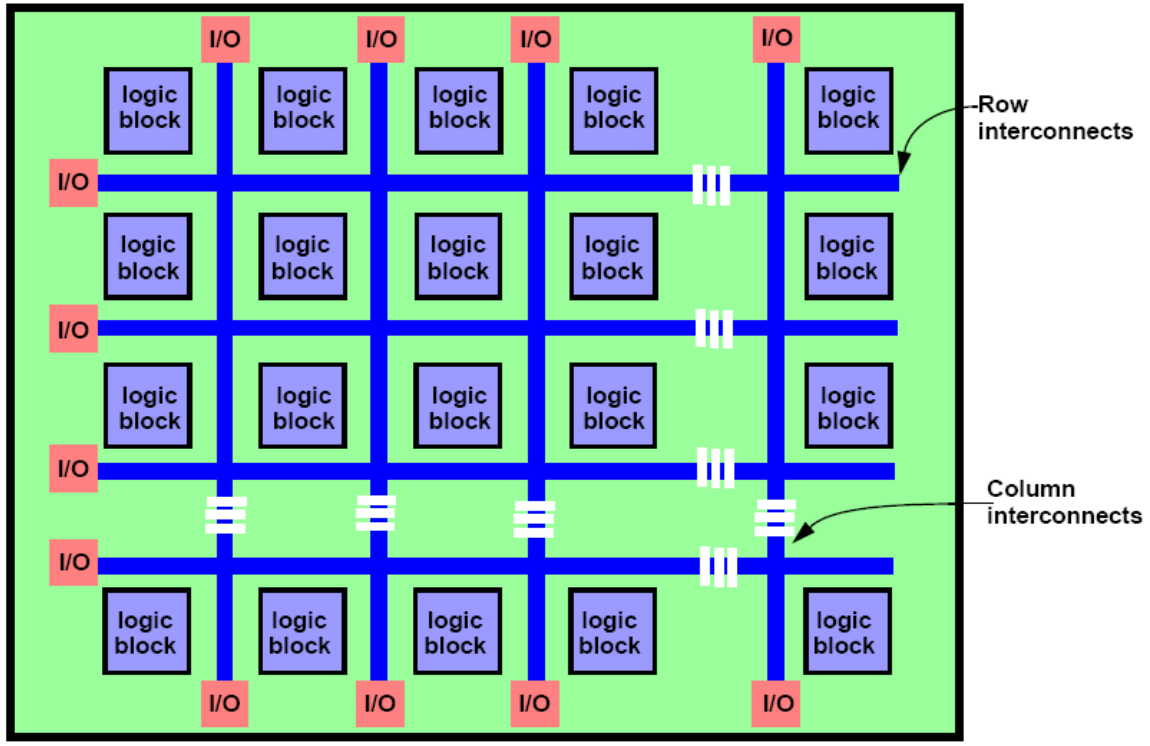
Basic OR array



Basic AND array

History. CPLD/FPGA

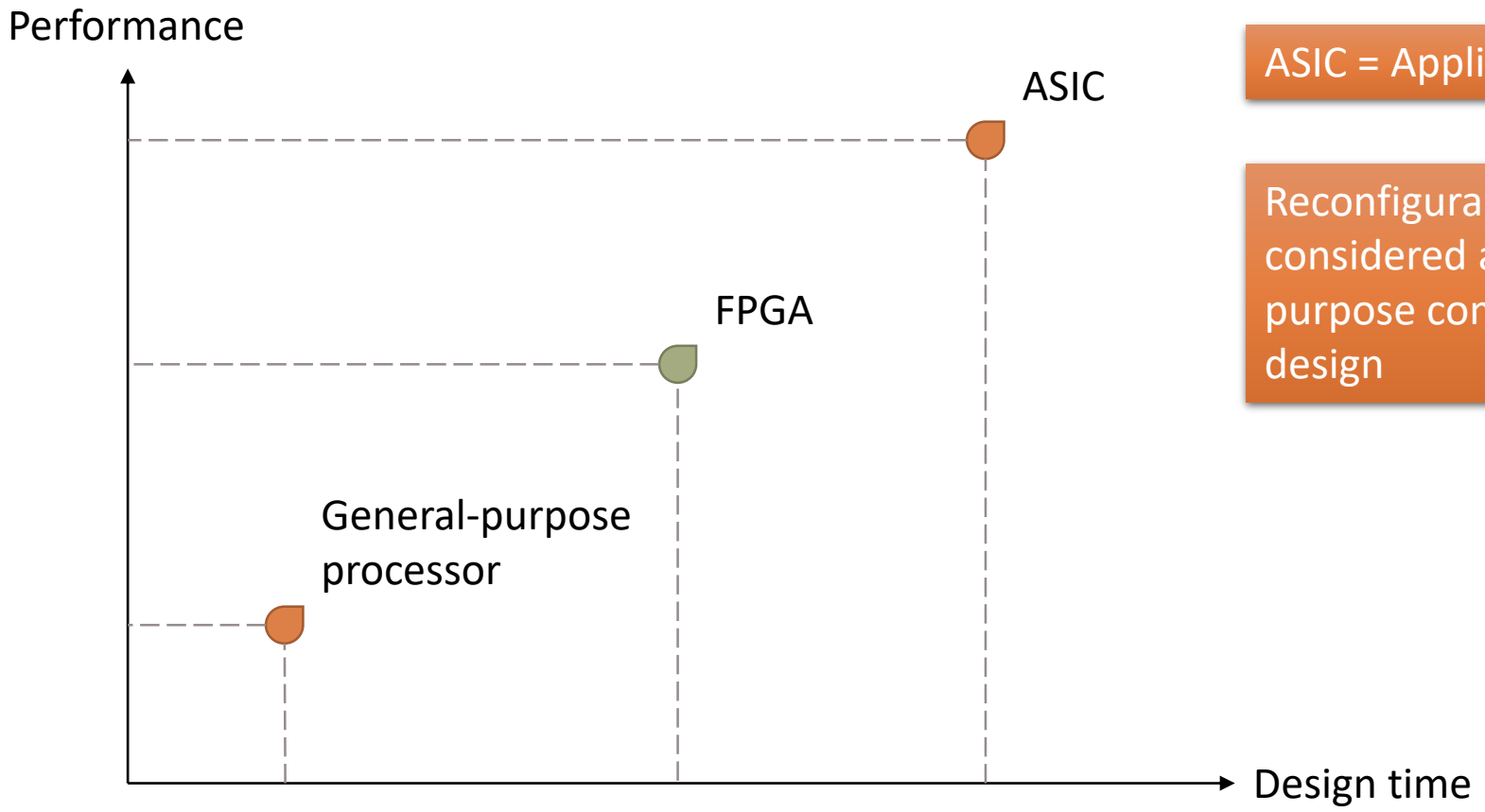
- A complex PLD (CPLD) is nothing else than a collection of multiple PLDs and an interconnection structure.
- Compared to a CPLD, a Field Programmable Gate Array (FPGA) contains a much larger number of smaller individual blocks + large interconnection structure that dominates the entire chip.



Reconfigurable hardware vs ICs

- Traditional IC
  - Once fabricated cannot be changed
  - Examples: CPU, GPU, TPU
- Reconfigurable hardware
  - Does not implement a specific circuit functionality
  - Can be reconfigured to implement any desired functionality/circuit
  - Can be reconfigured many times during the lifetime
  - Offer a flexible platform for increasingly complex systems
  - Example: FPGA (Field-Programmable Gate Array)

Reconfigurable hardware vs ICs

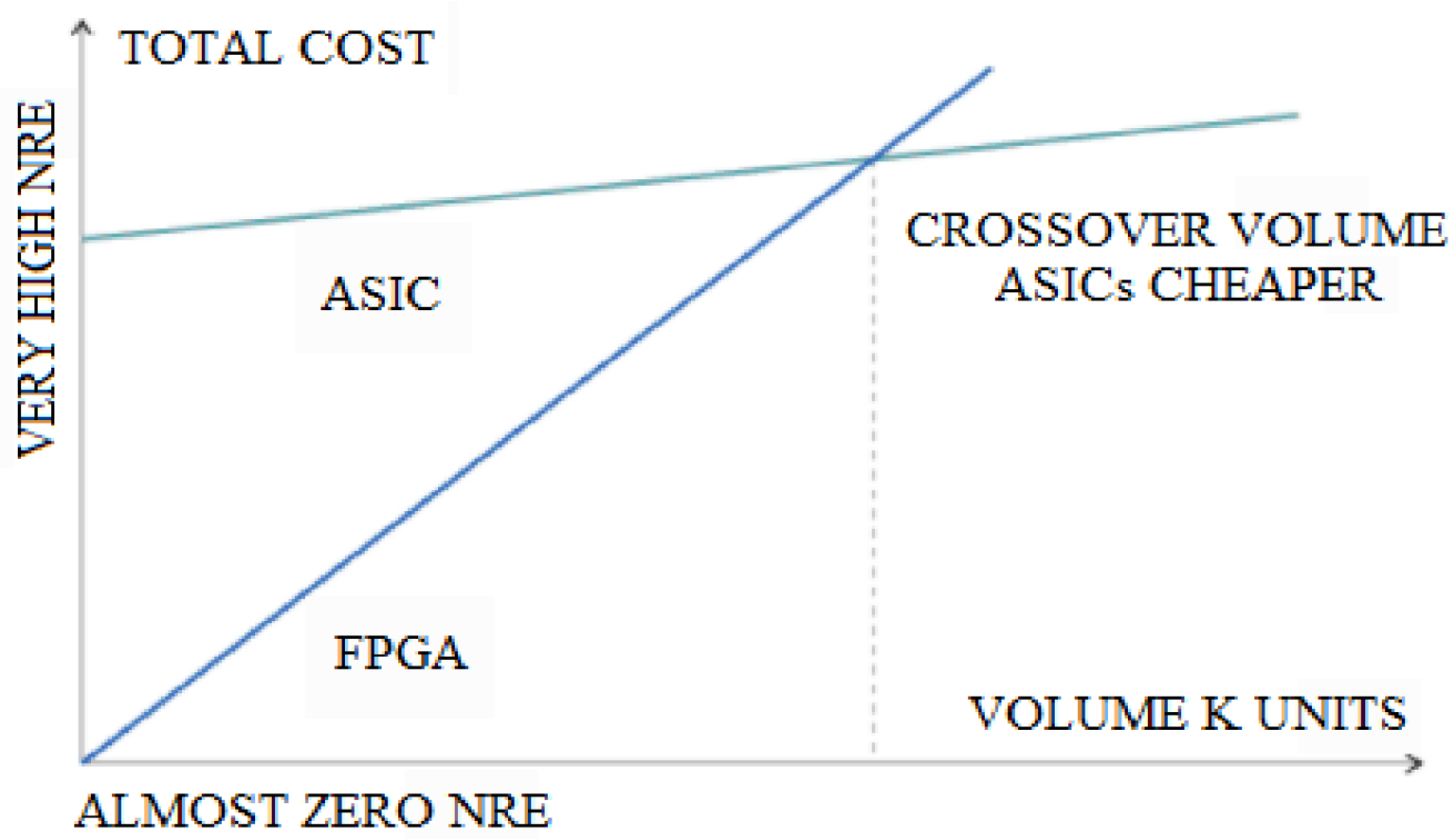


ASIC = Application Specific Integrated Circuit

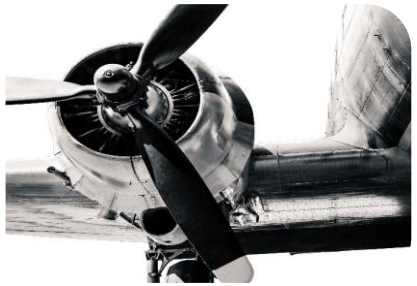
Reconfigurable computing can be considered as a trade-off between general-purpose computing and application-specific design

Reconfigurable hardware vs ICs

- FPGAs are less costly with smaller volumes due to lower Non-Recurring Engineering (NRE) costs:



FPGA applications



Aerospace
(satellites, planes)

Military equipment

Medical devices

Automotive
industry

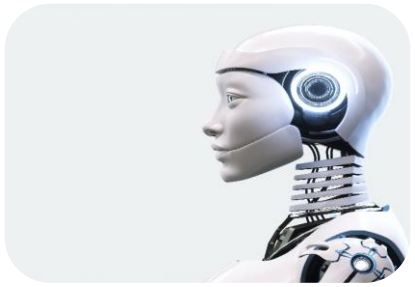
High-end
broadcast systems

Wired and wireless
communication

Production line
robots

Video processing
and surveillance

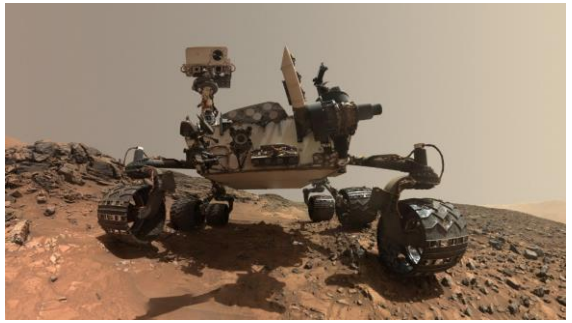
Data centers



AI/ML applications

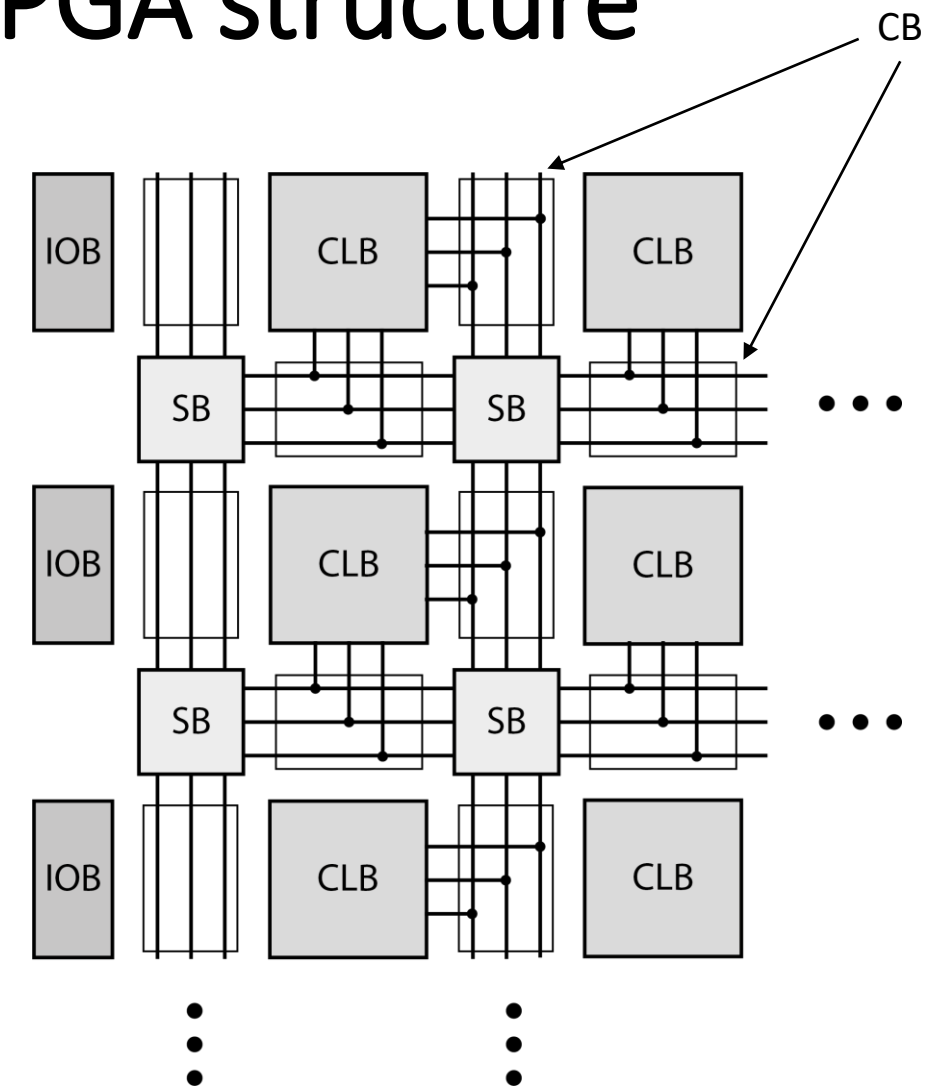
Examples of FPGA applications

- NASA
  - Mars Science Lab mission (Curiosity)
  - Mars Exploration Rover mission (Spirit, Opportunity)
  - Mars 2020 Perseverance rover
- Microsoft
  - Bing indexing acceleration
  - Deep learning acceleration in Azure cloud
- Amazon
  - Amazon Elastic Compute Cloud (EC2) F1 instances



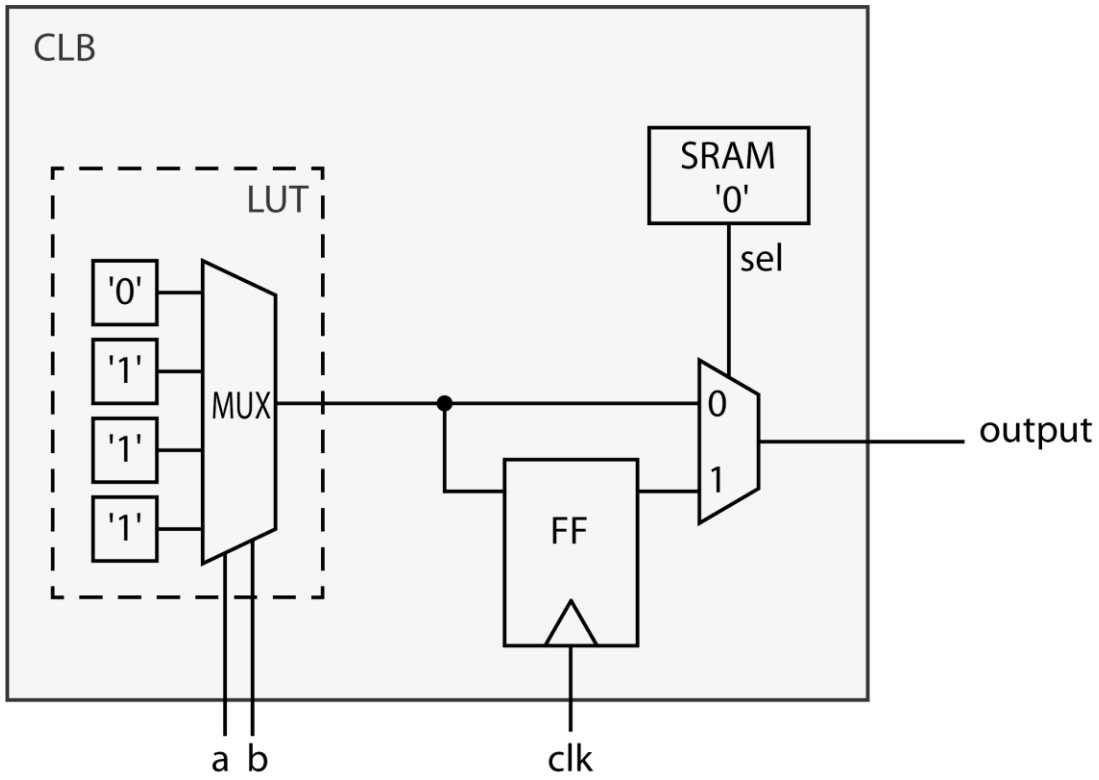
https://www.xilinx.com/about/blogs/xilinx-xclusive-blog/2021/rover-lands-on-mars-with-xilinx-fpgas-on-board.html

FPGA structure



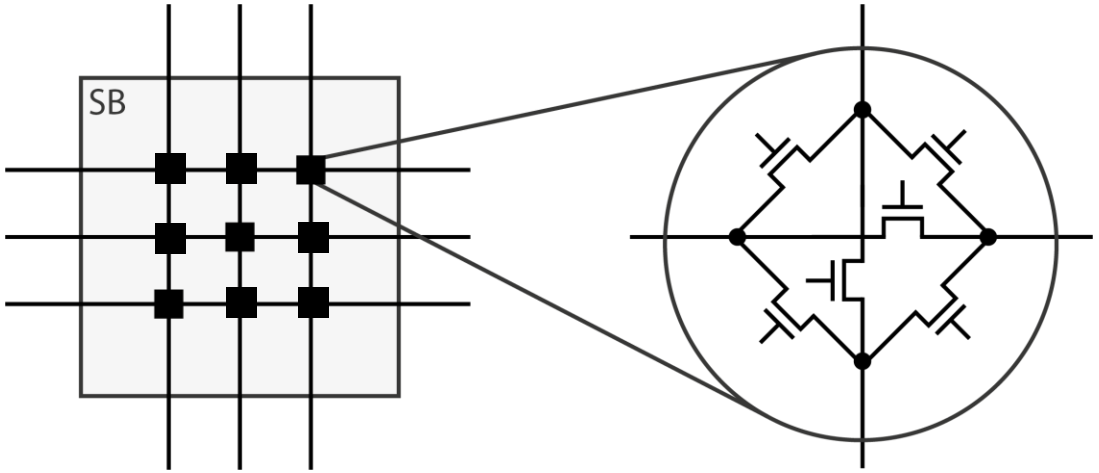
- Basic blocks:
  - Configurable Logic Block (CLB)
    - Implement logic functions
  - Connection Box (CB)
    - Connect inputs and outputs of CLBs to the connection wires
  - Switch Box (SB)
    - Connect vertical and horizontal wires
  - Input/Output Block (IOB)
    - Connect IO pins with the internal resources

Configurable Logic Block



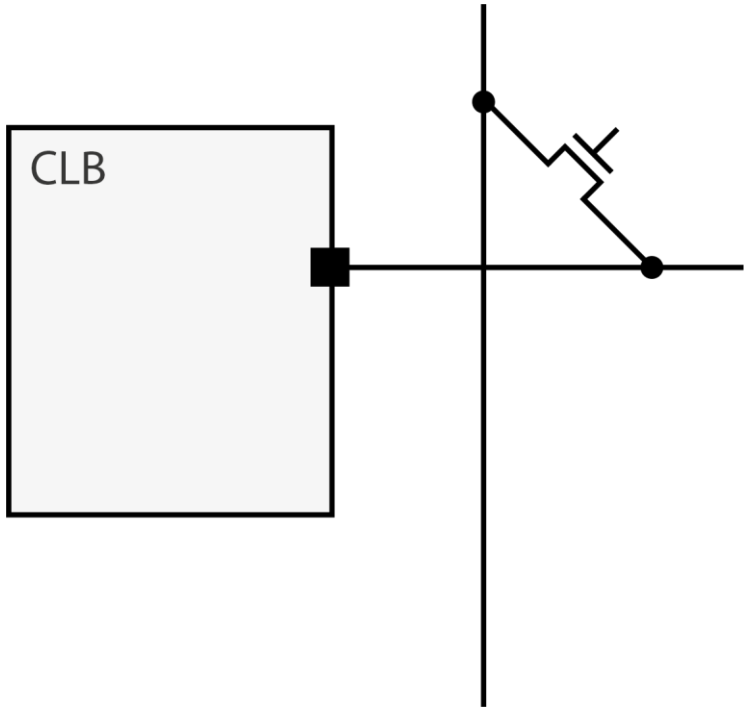
- Look-up Table (LUT) implements a combinational logic function
  - Stores a truth table of the function
  - Inputs act as an address or index of the table cell
- Register (flip-flop FF) optionally stores the output of LUT
- MUX selects the output of the CLB

Switch Box

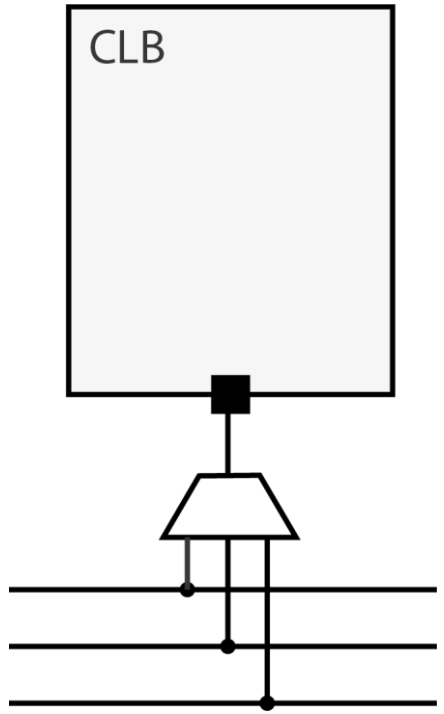


- Each switch point consists of six programmable switches
- Switches are controlled by configuration memory cells

Connection Box

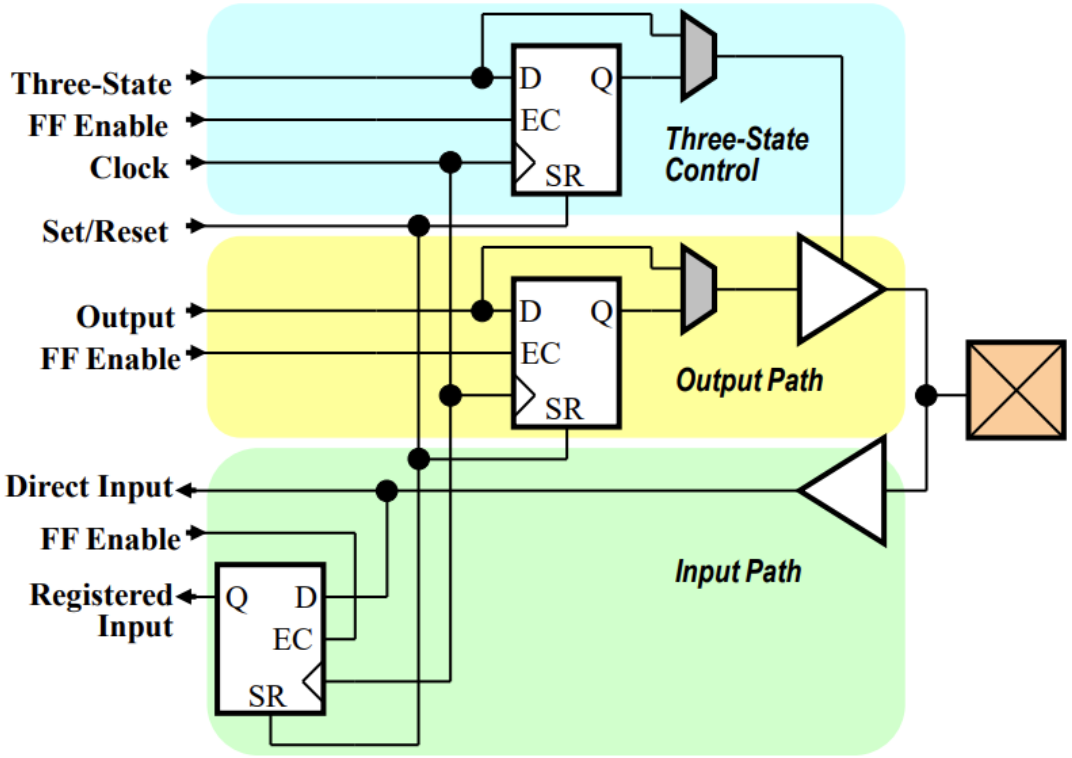


Connecting to output of a CLB



Connecting to input of a CLB

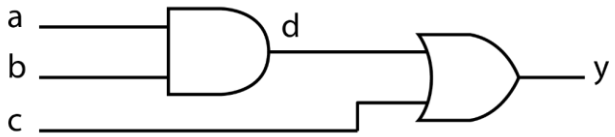
Input/Output Block

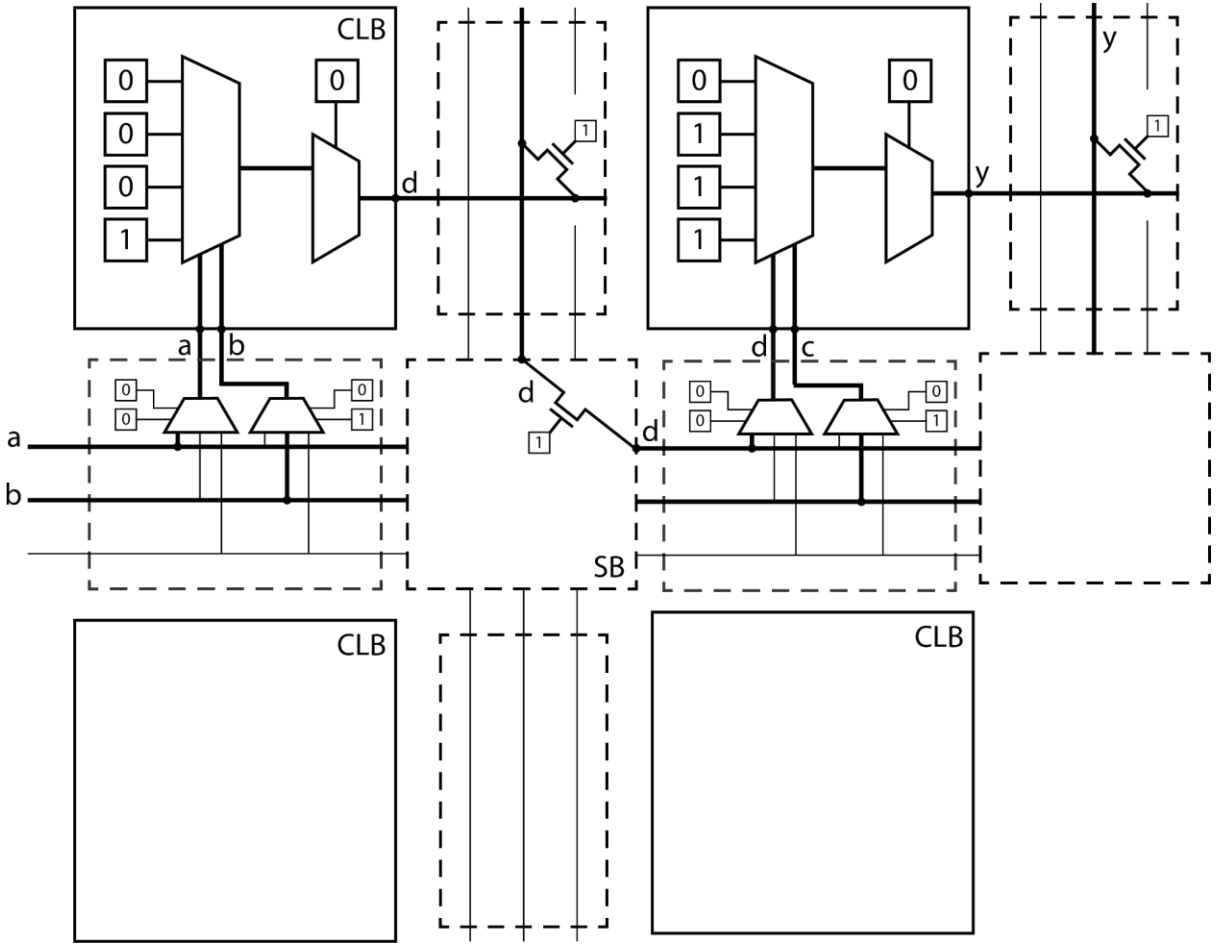


Cristinel Ababei, *EECE-4740 Advanced VHDL and FPGA Design* [lecture slides]

- Each IOB can work as a uni- or bi-directional IO
- Outputs can be forced into high-impedance
- Inputs and outputs can be latched to registers

Example





FPGA technology

- Antifuse
  - One-time programmable
- Flash
  - Non-volatile memory → configuration is not lost when the power is down
  - Limited re-programmability
- SRAM
  - Volatile memory
  - Easily re-programmable

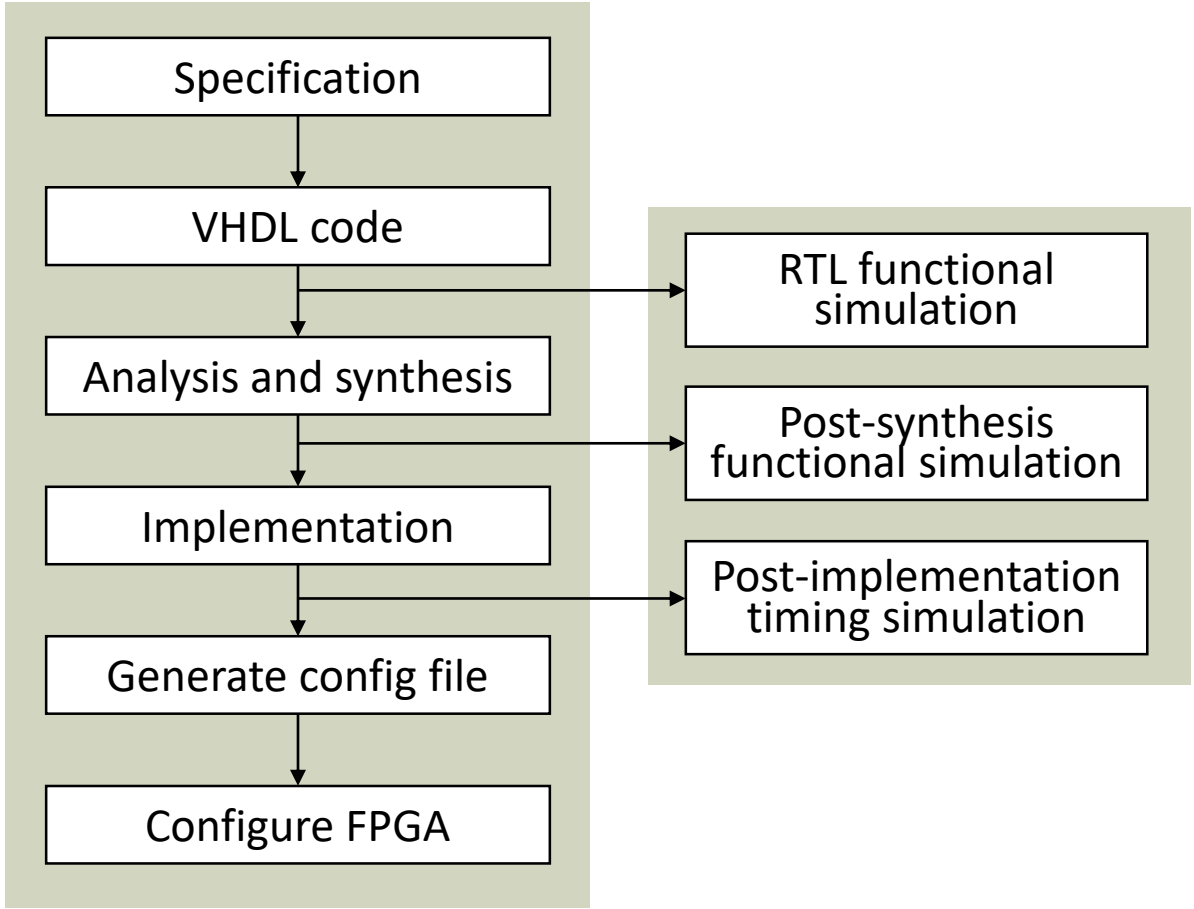
FPGA vendors

- AMD-Xilinx Inc.
- Intel (Altera)

~90% of the market

- Microchip
- Lattice Semiconductor
- QuickLogic Corporation
- Achronix
- Efinix Inc.

Design flow for FPGA

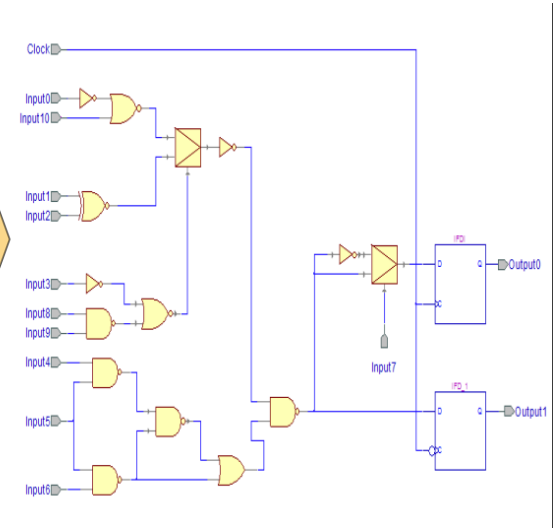


- Synthesis is the transformation of the HDL description into HW components
- Implementation combines placement and routing
- Configuration file (bitstream) is loaded into configuration memory

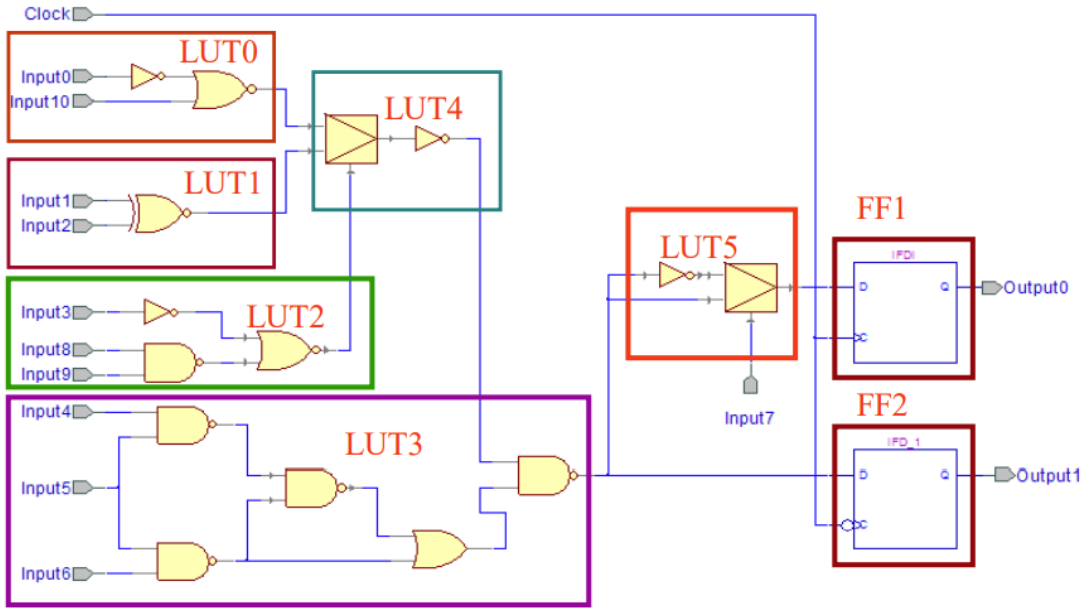
Synthesis

HDL description → HW components

```
architecture MLU_DATAFLOW of MLU is

signal A1:STD_LOGIC;
signal B1:STD_LOGIC;
signal Y1:STD_LOGIC;
signal MUX_0, MUX_1, MUX_2, MUX_3: STD_LOGIC;

begin
        A1<=A when (NEG_A='0') else
                not A;
        B1<=B when (NEG_B='0') else
                not B;
        Y<=Y1 when (NEG_Y='0') else
                not Y1;

        MUX_0<=A1 and B1;
        MUX_1<=A1 or B1;
        MUX_2<=A1 xor B1;
        MUX_3<=A1 xnor B1;

        with (L1 & L0) select
                Y1<=MUX_0 when "00",
                        MUX_1 when "01",
                        MUX_2 when "10",
                        MUX_3 when others;

end MLU_DATAFLOW;
```

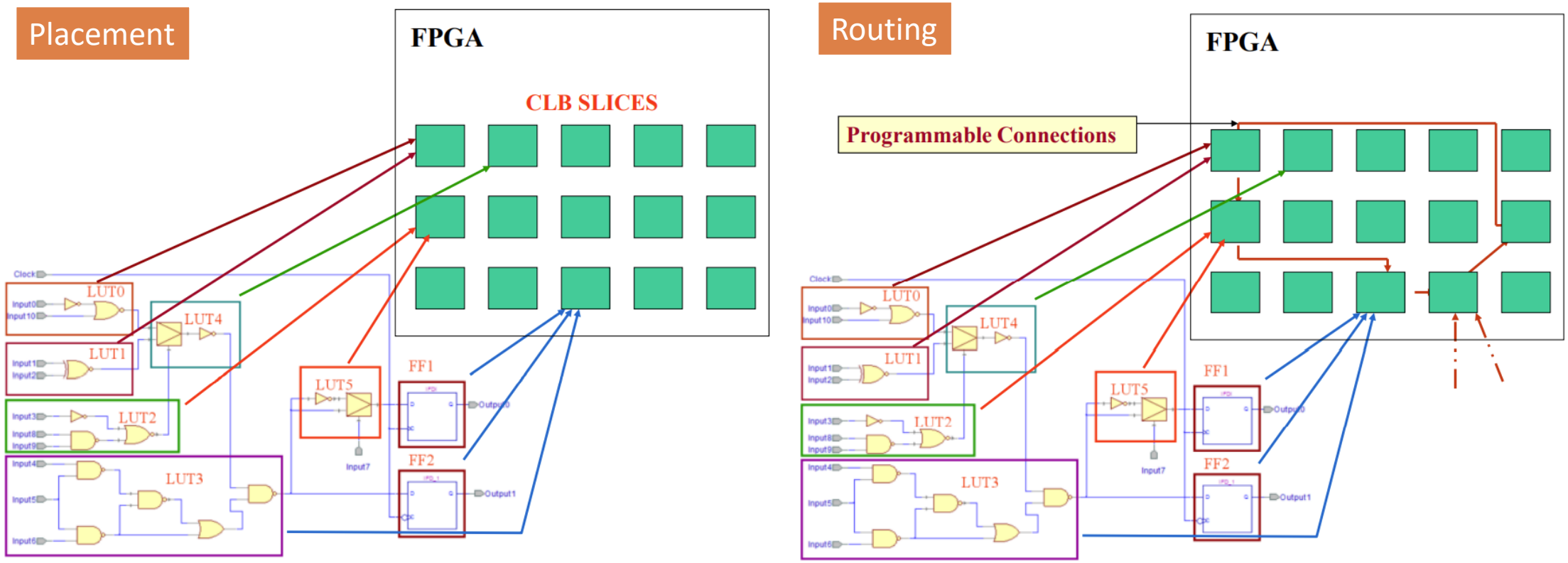


Technology mapping



Cristinel Ababei, *EECE-4740 Advanced VHDL and FPGA Design* [lecture slides]

Implementation



Cristinel Ababei, *EECE-4740 Advanced VHDL and FPGA Design* [lecture slides]

Configuration

- Once a design is implemented, you must create a file that the FPGA can understand
- This file is called a bitstream: a BIT file
- Bitstream holds values that will be loaded into configuration memory
  - LUT values
  - Routing configuration
  - Control signals for CLBs and IOBs