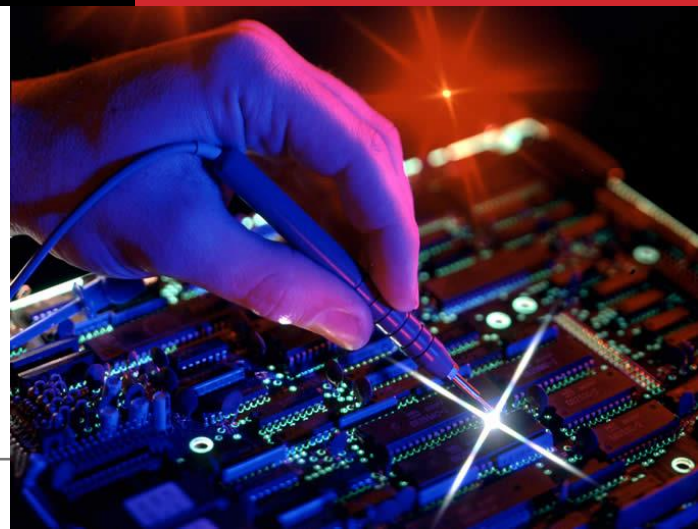
Advanced Computer Architecture & Advanced Microprocessor System

INTRODUCTION TO FPGAS AND LABS

Dennis A. N. Gookyi



CONTENTS

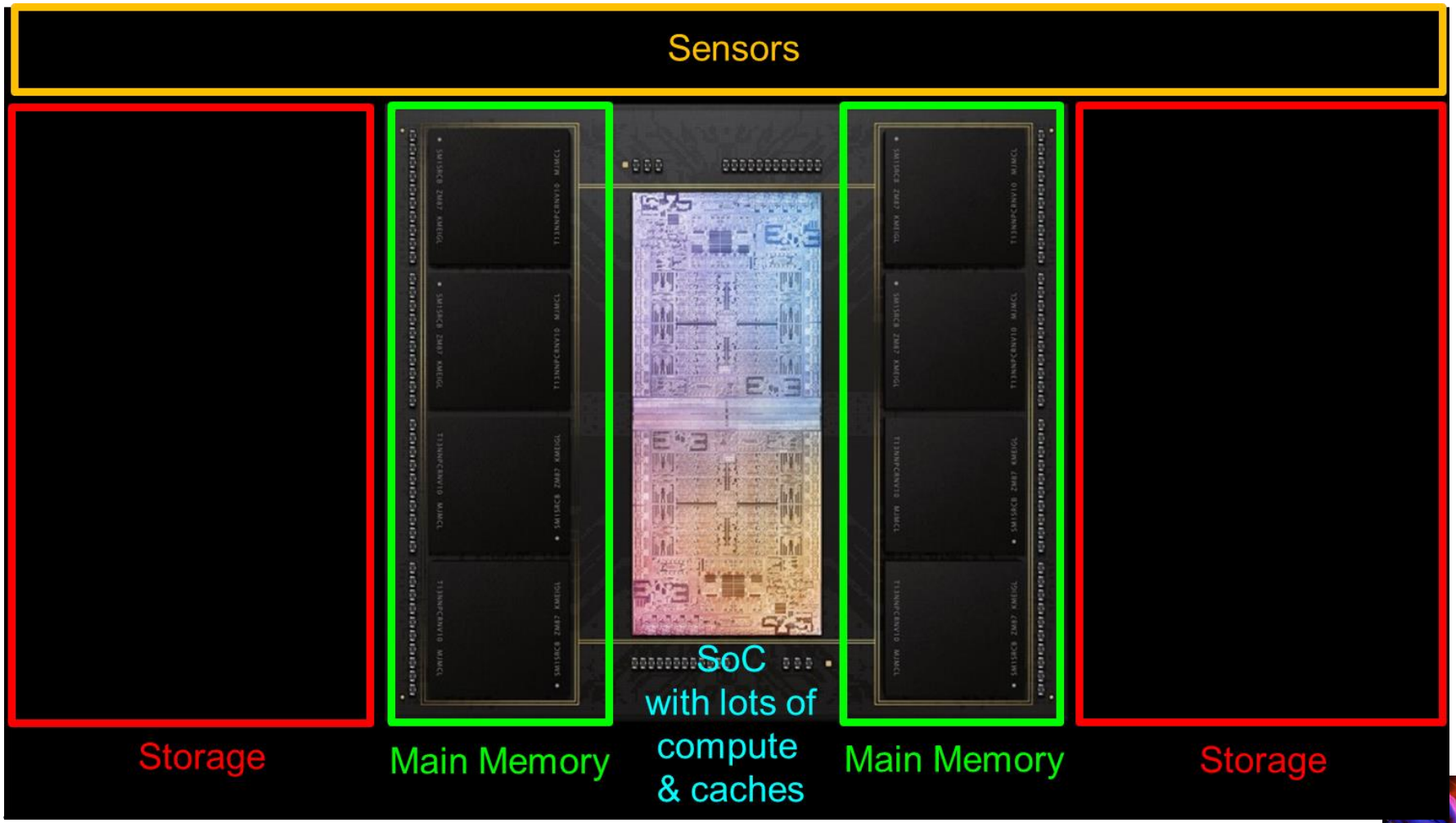
- Objectives
- FPGAs
- FPGAs in Today's Systems
- More about FPGAs
- Programming an FPGA
- Tutorial and Demo
- Overview of the Lab Exercises

OBJECTIVES

- High-level Goals of this course
  - In Microprocessors & Computer Architecture:
    - Understand the basics
    - Understand the principles of design
  - Based on such understanding:
    - Learn how a modern computer works underneath
    - Evaluate tradeoffs of different designs and ideas
    - Implement a principled design (a simple microprocessor)
    - Learn to systematically debug increasingly complex systems
    - Hopefully enable you to develop novel, out-of-the-box designs
  - The focus is on basics, principles, and how to use them to create/implement good designs

OBJECTIVES

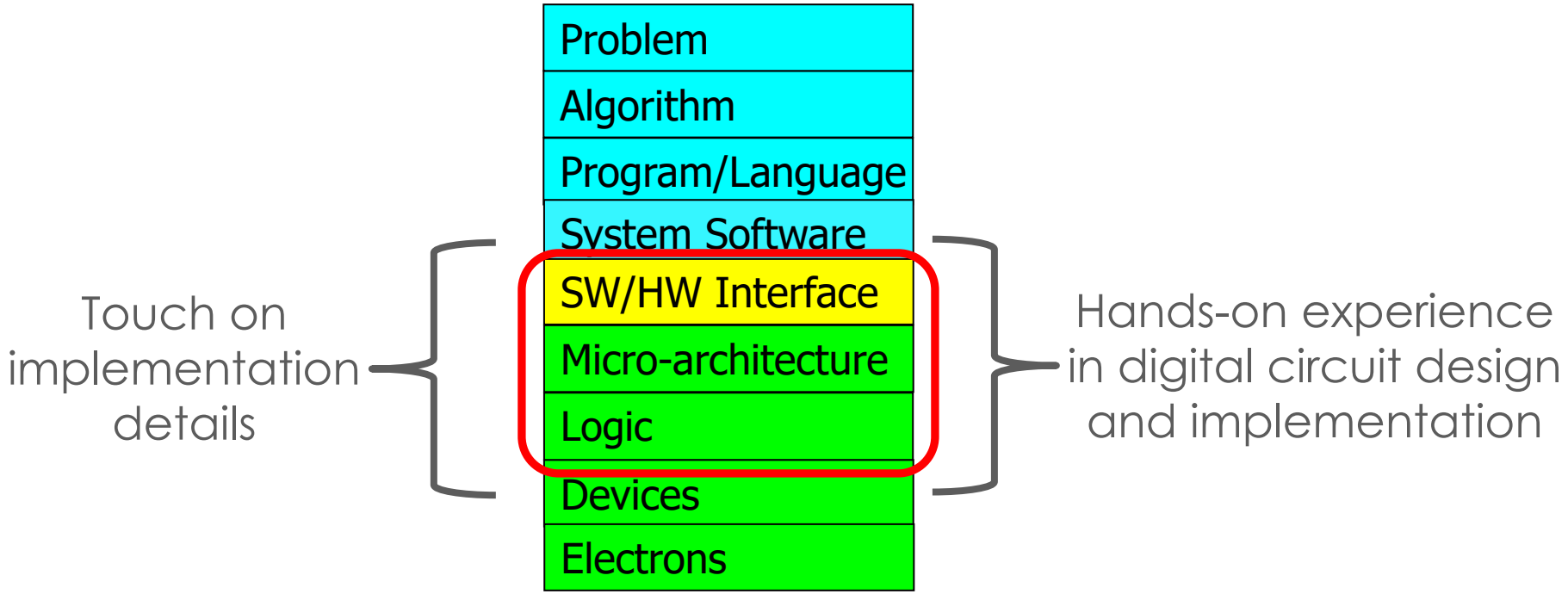
- We will study how something like this works



Apple M1 Ultra System (2022)

OBJECTIVES

- The Transformation Hierarchy



Understanding how a processor works underneath the software layer

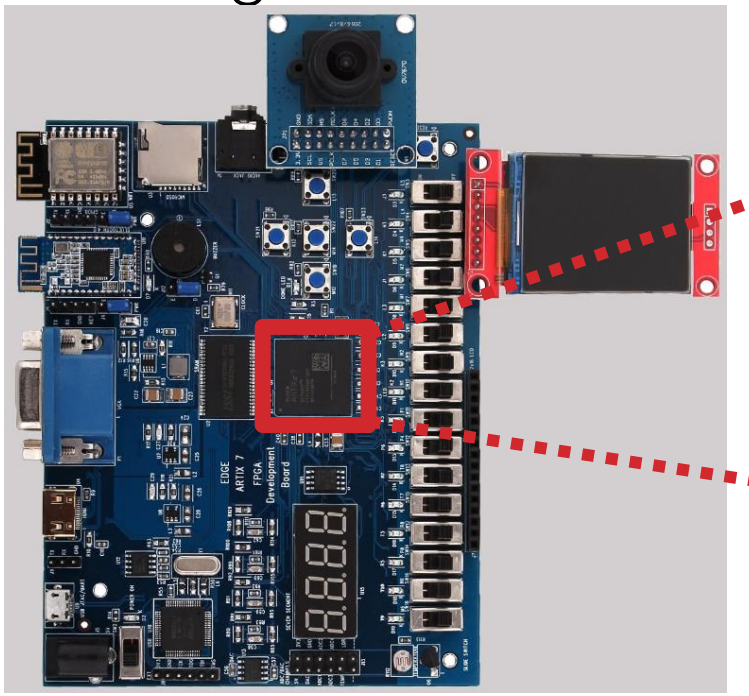
OBJECTIVES

- How to make trade-offs between performance and area/complexity in your hardware implementation

- Hands-on experience on:
  - Hardware Prototyping on Field Programmable Gate Arrays (FPGAs)
  - Debugging Your Hardware Implementation
  - Hardware Description Language (HDL)
  - Hardware Design Flow
  - Computer-Aided Design (CAD) Tools

FPGA

- Field Programmable Gate Array (FPGA)

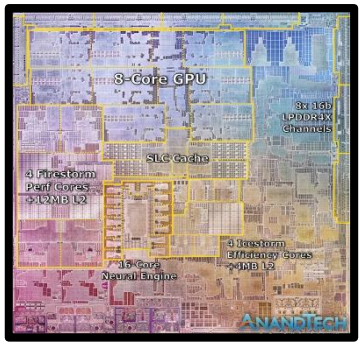


- FPGA is a software-reconfigurable hardware substrate
  - Reconfigurable functions
  - Reconfigurable interconnection of functions
  - Reconfigurable input/output (IO)

FPGA

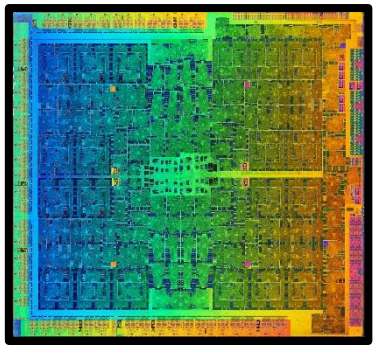
- FPGAs and Other Integrated Circuits

CPUs



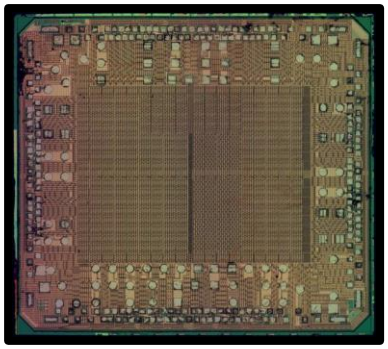
Apple M1

GPUs



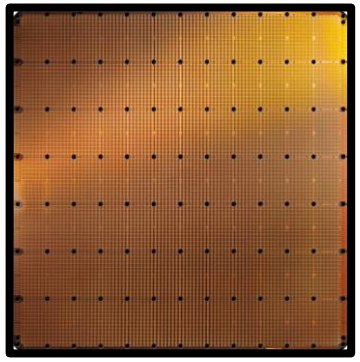
Nvidia GTX 1070

FPGAs



Xilinx Spartan

ASICs



Cerebras WSE-2

Flexibility
Programming Ease

Efficiency

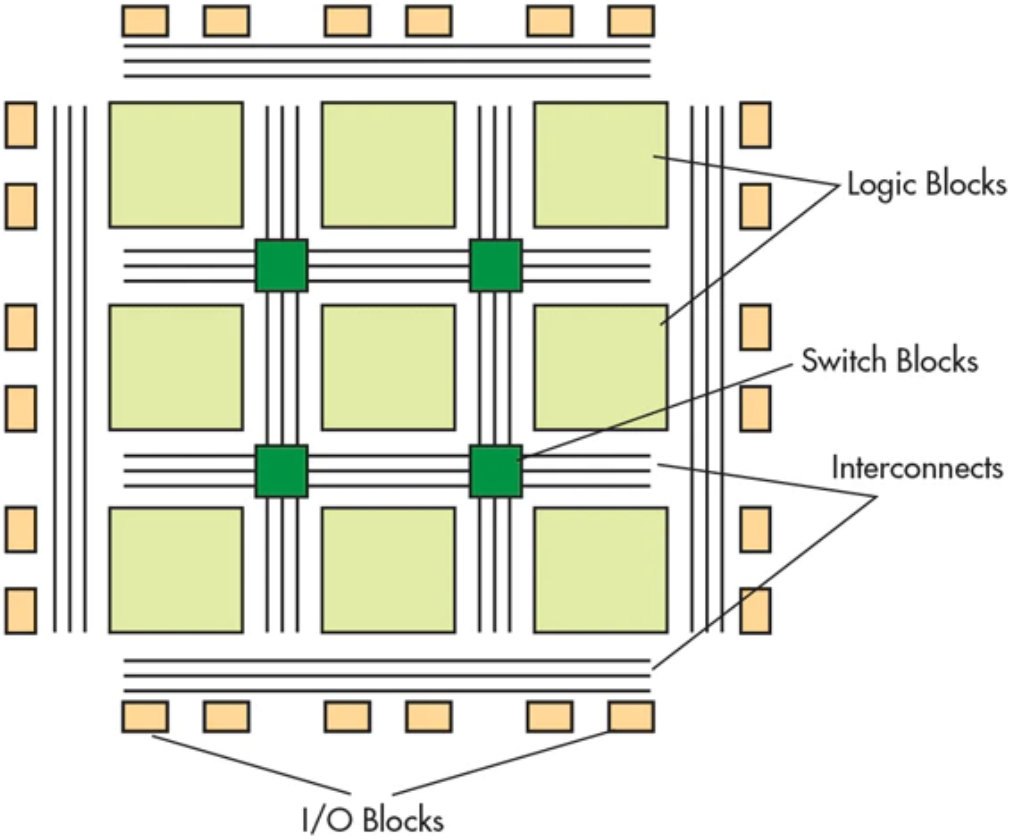
FPGAS

- They look the same

	Microprocessors	FPGAs	ASICs
In short:	Common building block of computers	Reconfigurable hardware, flexible	You customize everything
Program Development Time	minutes	days	months
Performance	o	+	++
Good for	Ubiquitous Simple to use	Prototyping Small volume	Mass production, Max performance
Programming	Executable file	Bit file	Design masks
Languages	C/C++/Java/…	Verilog/VHDL	Verilog/VHDL
Main Companies	Intel, ARM, AMD, Apple, NVIDIA	Xilinx, Altera	TSMC, Globalfoundries

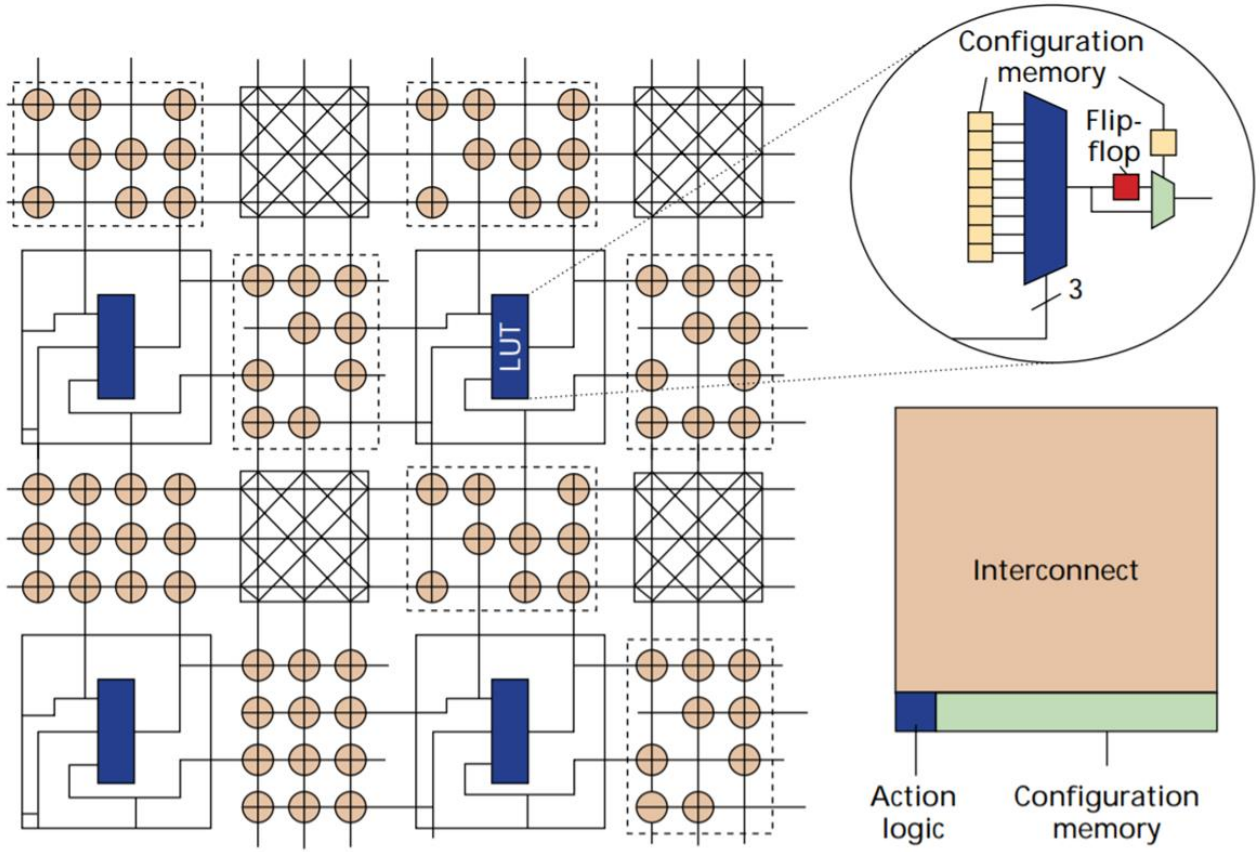
FPGAS

- High-level overview of FPGAs
  - We configure logic blocks, their connections, and IO blocks to create hardware circuits and map programs onto those circuits



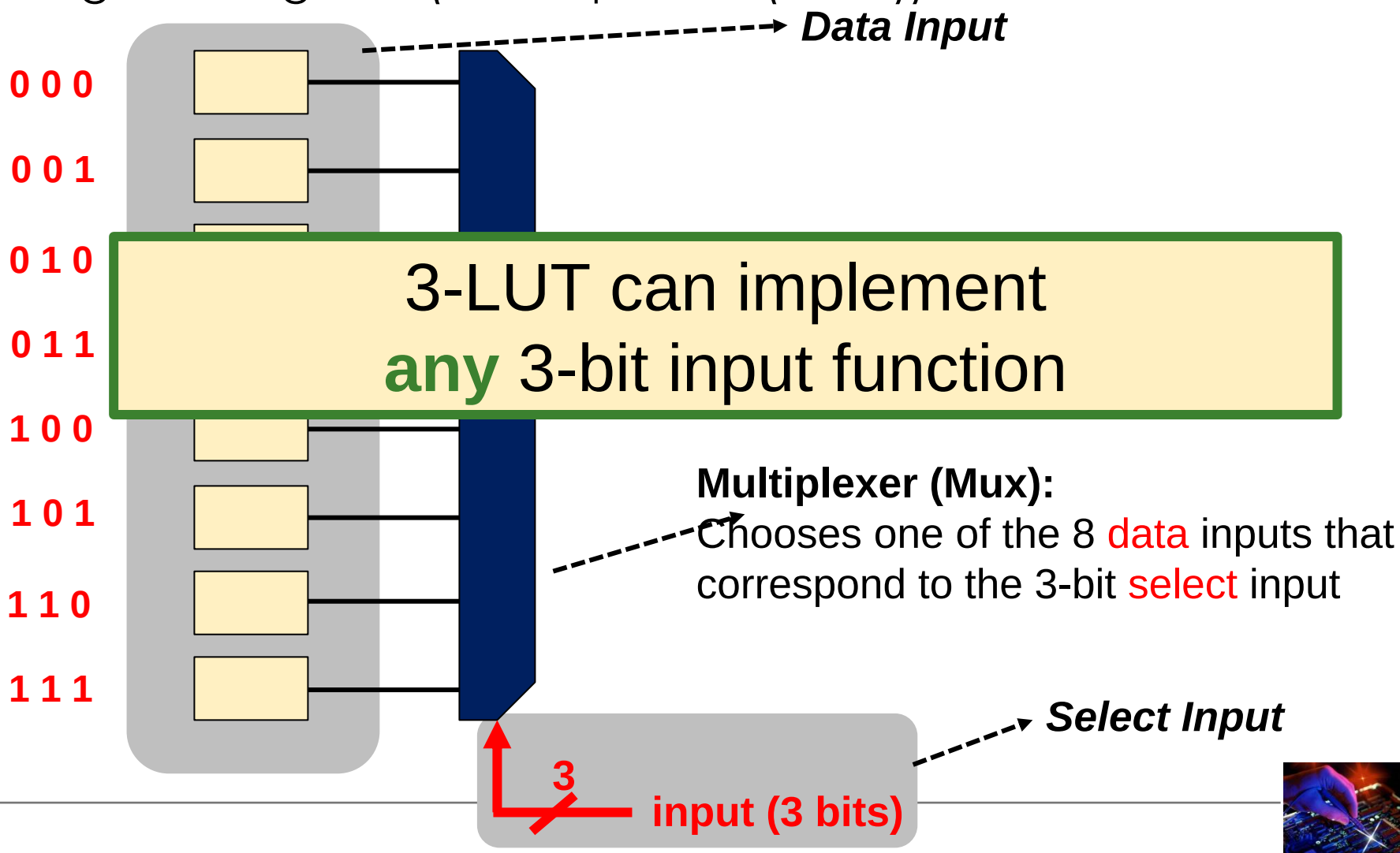
FPGAS

- Looking inside an FPGA
  - Two main building blocks
    - Look-Up Tables and Switches



FPGAS

- Programming LUTs (3-bit input LUT (3-LUT))



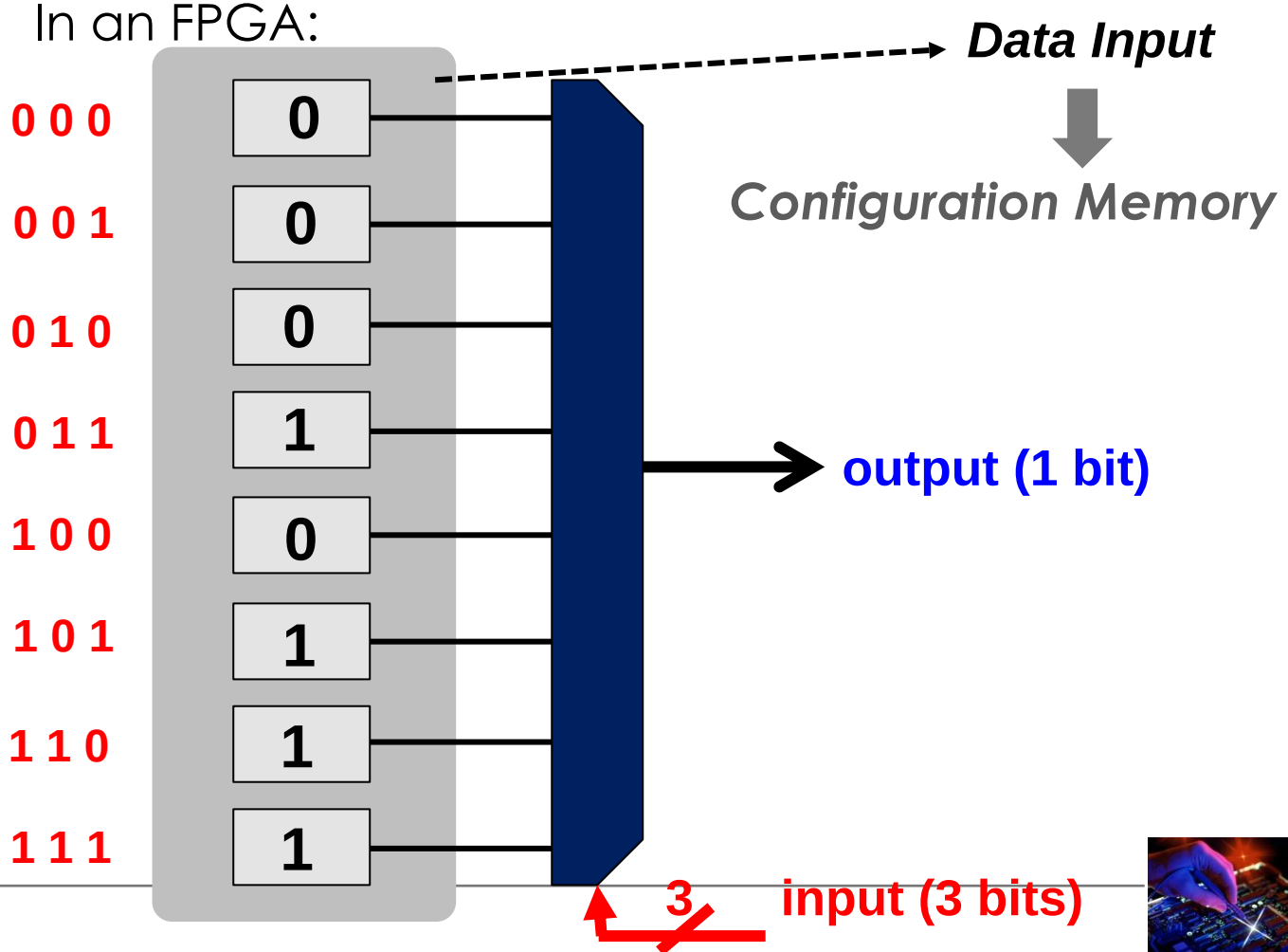
FPGAS

- Let's implement a function that outputs '1' when there are at least two '1's in a 3-bit input

In C:

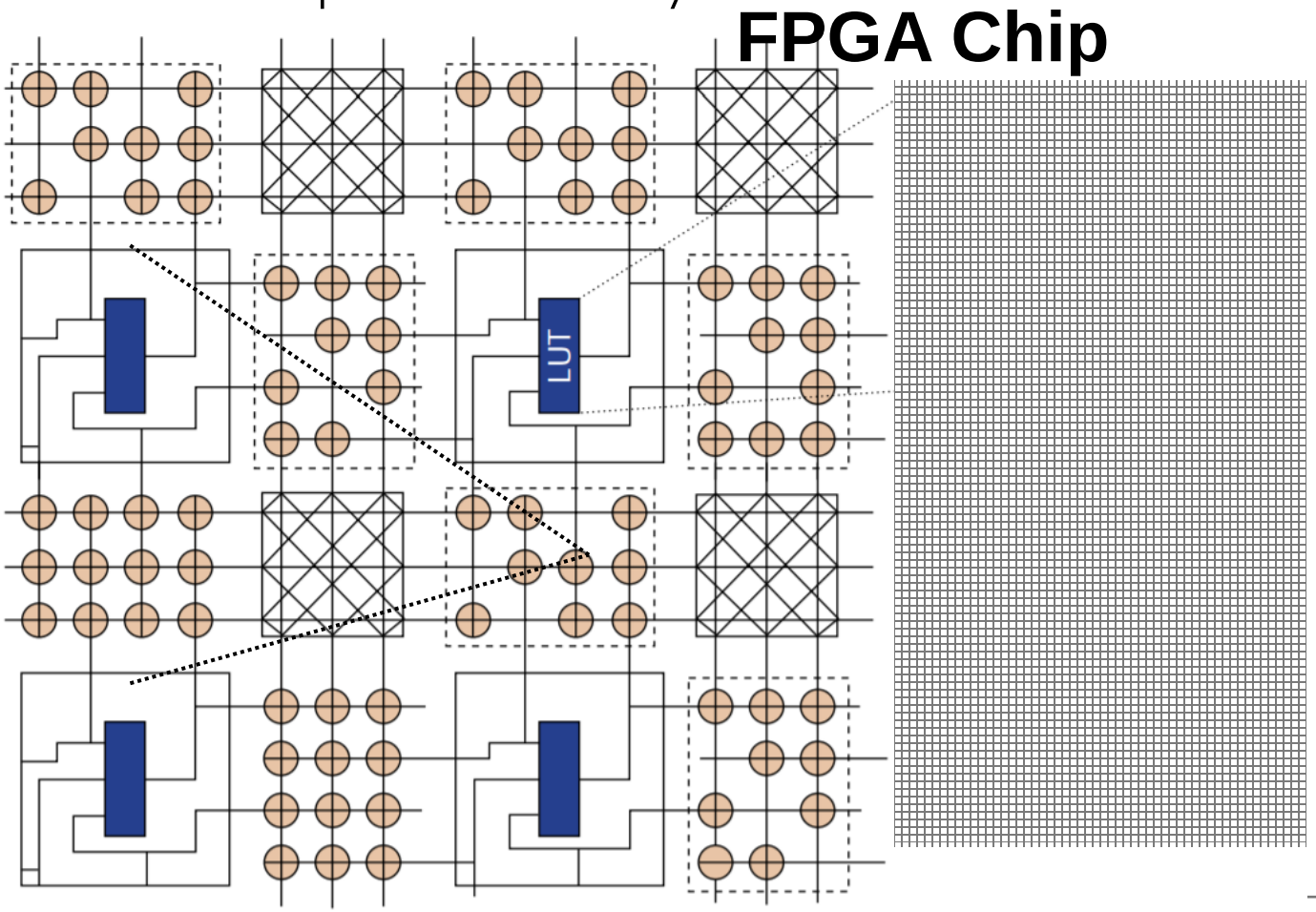
```
switch(input){
   case 0:
   case 1:
   case 2:
   case 4:
      return 0;
   default:
      return 1;}
```

In an FPGA:



FPGAS

- Implementing complex functions
  - FPGAs are composed of many LUTs and Switches

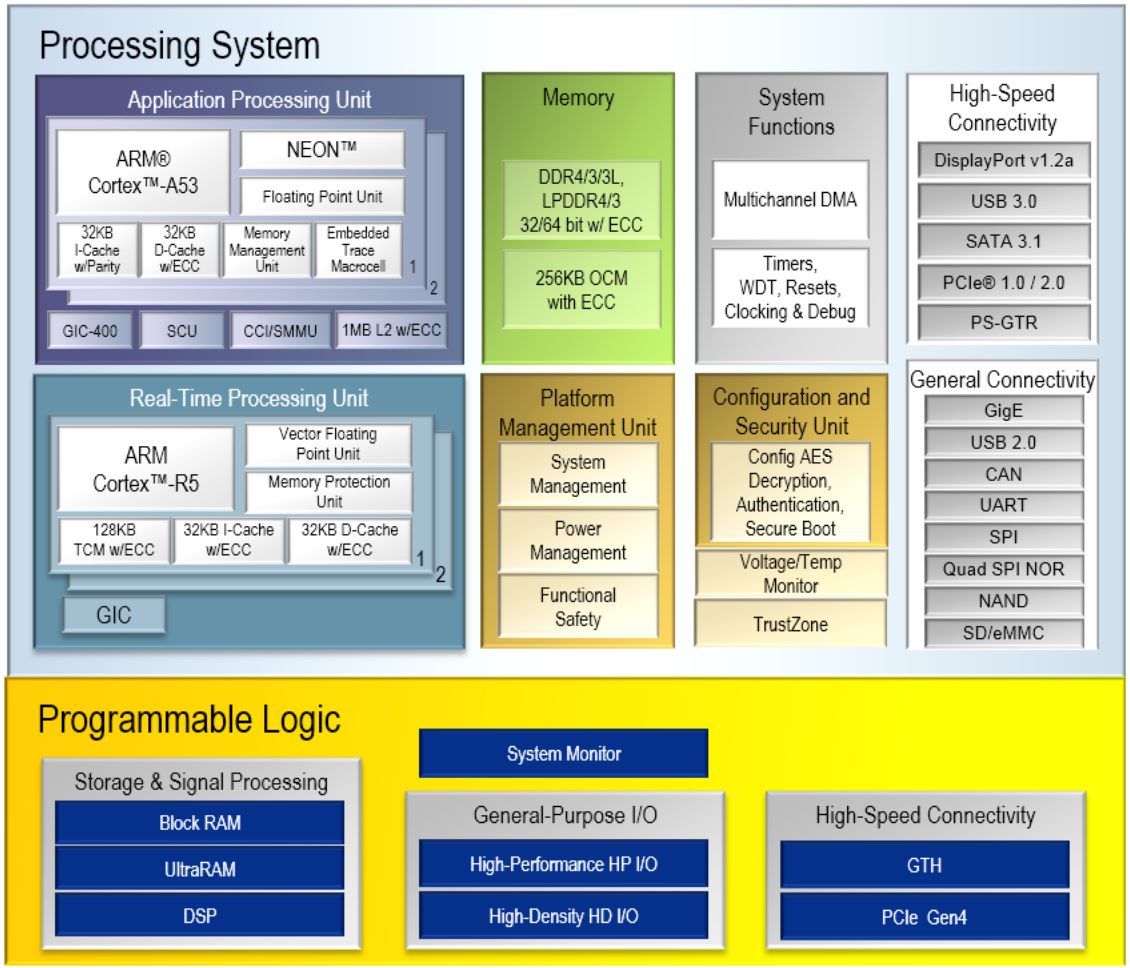


FPGAS

- Modern FPGA architecture
  - Typically use LUTs with 6-bit select input (6-LUT)
    - Thousands of them
  - MegaBytes of distributed on-chip memory
  - Hard-coded special-purpose hardware blocks for high-performance operations
    - Memory interface
    - Low latency and high bandwidth off-chip I/O
  - Even a general-purpose processor embedded within the FPGA chip

FPGAS

- Modern FPGA architecture
  - An Example Modern FPGA Platform: Xilinx Zynq Ultrascale+



FPGAS

- Advantages and Disadvantages of FPGAs
  - Advantages
    - An algorithm can be implemented directly in hardware
    - No ISA, high specialization -> high performance, high energy efficiency
    - Low development cost (vs. a custom hardware design)
    - Short time to market (vs. a custom hardware design)
    - Reconfigurable in the field
    - Usable and reusable for many purposes
    - Good for both prototyping and application acceleration
  - Disadvantages
    - Not as fast and power efficient as dedicated hardware customized for an algorithm
    - Reconfigurability comes at a cost: significant area and latency overhead

FPGA

- Programming the FPGA
- Computer-Aided Design (CAD) Tools
  - FPGAs have many resources (e.g., LUTs, switches)
  - They are hard to program manually
  - How can we
    - represent a high-level functional description of our hardware circuit using the FPGA resources?
    - select the resources to map our circuit to?
    - optimally configure the interconnect between the selected resources?
    - generate a final configuration file to properly configure an FPGA?

FPGAS

- FPGA design flow

Your task!

Problem Definition

↓

Hardware Description Language (HDL)

Verilog, *VHDL*

↓

Logic Synthesis

Automated *by Xilinx Vivado tools*

↓

Placement and Routing

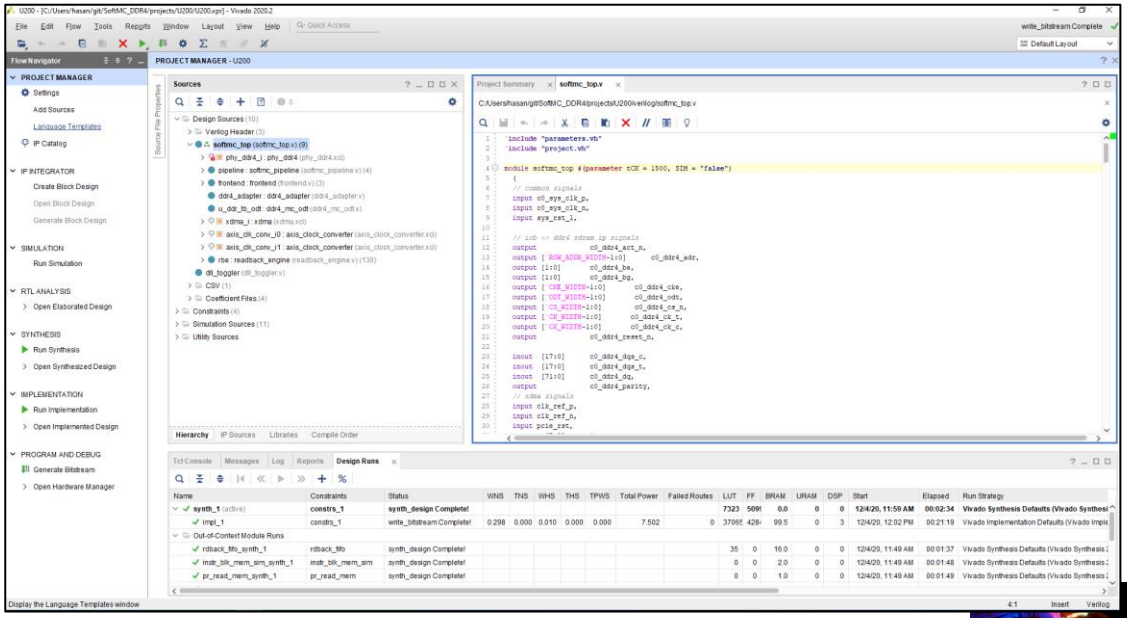
↓

Bitstream Generation

→

FPGAS

- Vivado
  - A software tool that helps us throughout the FPGA design flow
  - Provides tools to simulate our designs
    - Validate the correctness of the implementation
    - Debugging
  - Provides drivers and graphical interface to easily program the FPGA using a USB cable



LAB SUMMARY

- Each week we will have a new exercise
  - Not all exercises will require the FPGA board
- At the end of the exercises, we will have built a 32-bit micro processor running on the FPGA board
  - It will be a small processor, but it will be able to execute pretty much any program
- You are encouraged to experiment with the board on your own

LAB SUMMARY

- Research Paper Reading:
  - **RISC-V Hardware Synthesizable Processor Design Test and Verification Using User-Friendly Desktop Application**
    - https://www.webology.org/abstract.php?id=1069

Webology, Volume 19, Number 1, January, 2022

RISC-V Hardware Synthesizable Processor Design Test and Verification Using User-Friendly Desktop Application

Hyogeun An
Hanbat National University, Daejeon, South Korea. E-mail: ahnhyogean@gmail.com

Sudong Kang
Hanbat National University, Daejeon, South Korea. E-mail: dongdonge9555@gmail.com

Guard Kanda
Hanbat National University, Daejeon, South Korea. E-mail: guardkanda@gmail.com

Kwangki Ryoo*
Hanbat National University, Daejeon, South Korea. E-mail: kkryoo@gmail.com

Received September 26, 2021; Accepted December 19, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V19I1/WEB19305