Traineeships in Advanced Computing for High Energy Physics (TAC-HEP)

GPU & FPGA module training: Part-2

Week-1: Introduction to FPGA and its architecture

Lecture-1: March 21st 2023

Varun Sharma

University of Wisconsin – Madison, USA

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Welcome!

- Welcome to the first lecture of **FPGA part** of the "*GPU & FPGA training module*"

- We will continue to meet twice a week for an hour:
  - 1 hr = 45min lecture + 15min Q&A.
  - Tuesdays: 9:00-10:00 CT / 10:00-11:00 ET / 16:00-17:00 CET
  - Wednesday: 11:00-12:00 CT / 12:00-13:00 ET / 18:00-19:00 CET
  - **Exceptional this week due to US-Europe 6 hr difference**

- Feel free to interrupt during the lecture in case of any clarifications/doubts

- Outside lecture hours:
  - Post your queries on slack channel, or;
  - Via email: varun.sharma@cern.ch

Content

- Motivation to know about FPGAs
- A bit of history
- Overview of FPGAs and comparison with others
- FPGA architecture
- Parallelism in FPGA

What these lectures are about?

You may get better at following things after this module:

- Understanding of FPGA, its architecture and usage
- Overview of a programmable logic
- Your interactions with electrical engineers taking about FPGA/programmable logic
- Write your own firmware for a physics algorithm
- How to read/understand/debug some of the issues

You may NOT:

- Become an electrical engineer, electonics is more than FPGAs...
- Be a FPGA "expert" it needs much more time..
- Improve your soldering skills, its just software/firmware ☺

Why do we need to learn about FPGA?

- We are experimental physicist (CMS, ATLAS, Neutrino, etc.)
- All experiments collect physics data via optical/electrical links
  - Readout and processing in almost all cases is based on FPGAs
- In general, distribution of work in experiments:
  - Physicist:
    - Algorithm, Firmware, tests, commissioning, etc…
  - Engineers:
    - Design, layouts, production, etc…
- To better understand and troubleshooting relevant components of the experiments
  - Physicists need FPGA knowledge
  - Understand the processing happening inside the FPGA
  - To talk to engineers and explain the needs

Are we enough motivated?

- Progammable Logic is state of the art:
  - Designing prototype electronics
  - Allow for easy reconfiguration

- Progammable Integrated Chips (ICs):
  - Producing chips is costlier than producing PCBs
  - One chip on many PCBs can reduce cost/chip
  - Programmable chips can reduce even further

Some history

- **1960:** Fuse Configurable Diode Matrix
- **1962:** Transistor-Transistor Logic (TTL)
- **1971:** Erasable Programmable ROM
- **1975:** Programmable Logic Array
- **1978:** Progammable Array Logic (PAL) (with EPROM)
- **1983**
  - Generic Array Logic (GAL)
  - Complex Programmable Logic Device (**CPLD)**
- **1985**
  - First Field Programmable Gate Array (FPGA)
  - ~6M logical gates

Fig. 1: Hand wired

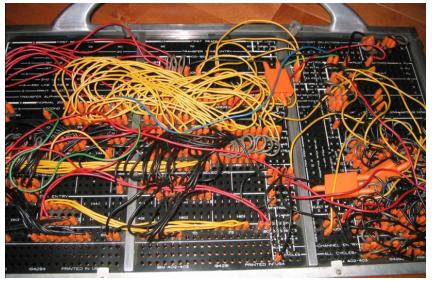
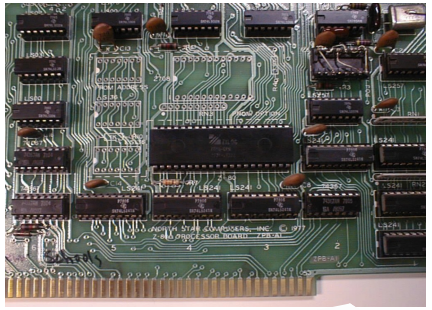


Fig. 2: Digital Electronics



FPGA

Fig. 3

What is a FPGA?

FPGA: Field **P**rogrammable **G**ate **A**rray

- Programmable hardware whose sub-component configuration can be changed even after fabrication: *"field-programmable"*
- Has 2D array of logic gates in its architecture: *"Gate Array"*

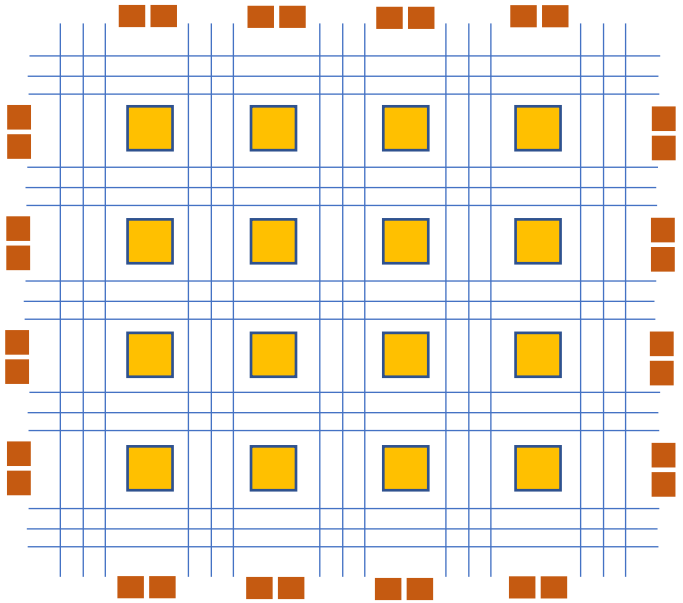


Fig. 4

ASIC: Application **S**pecific **I**ntegrated **C**ircuit

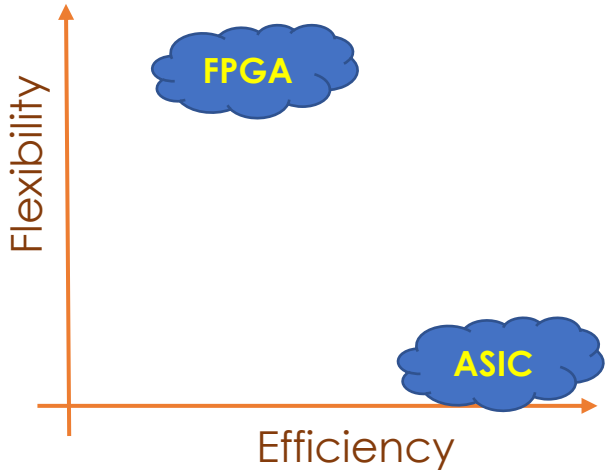


Fig. 5

FPGA originally popular for prototyping ASICs, but now also for high performance computing

A comparison

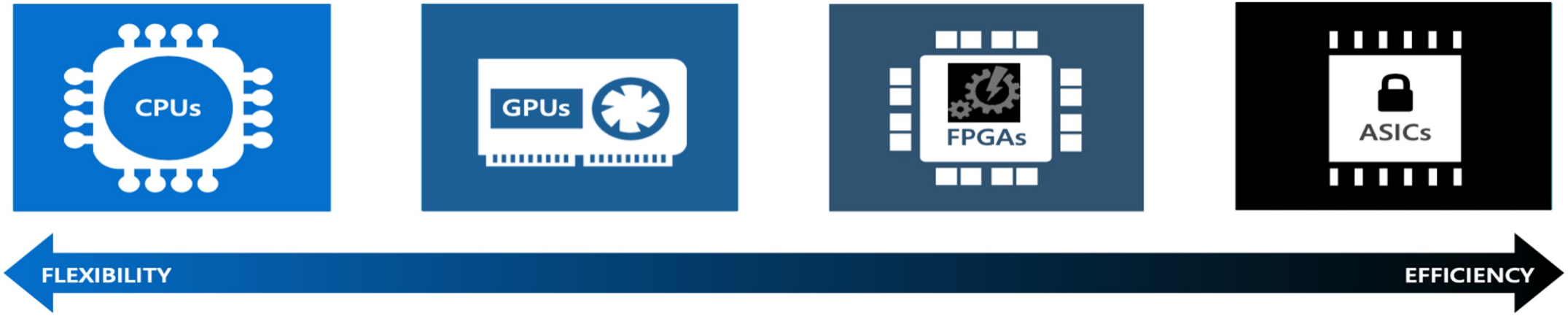


Fig. 6

General-purpose processors, the performance of which isn't ideal for graphics and video processing

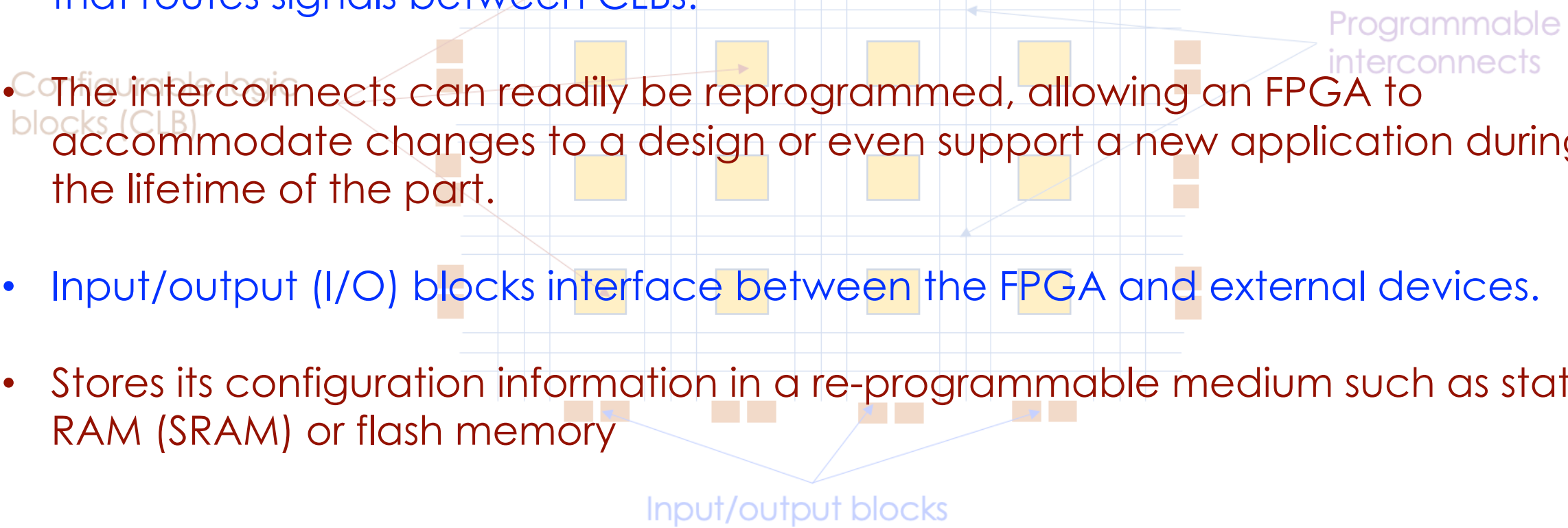
A popular choice for AI computations. GPUs offer parallel processing capabilities, making it faster at image rendering than CPUs.

FPGAs, such as those available on Azure, provide performance close to ASICs. They're also flexible and reconfigurable over time, to implement new logic.

Custom circuits, such as Google's Tensor Processor Units (TPU), provide the highest efficiency. They can't be reconfigured as your needs change.

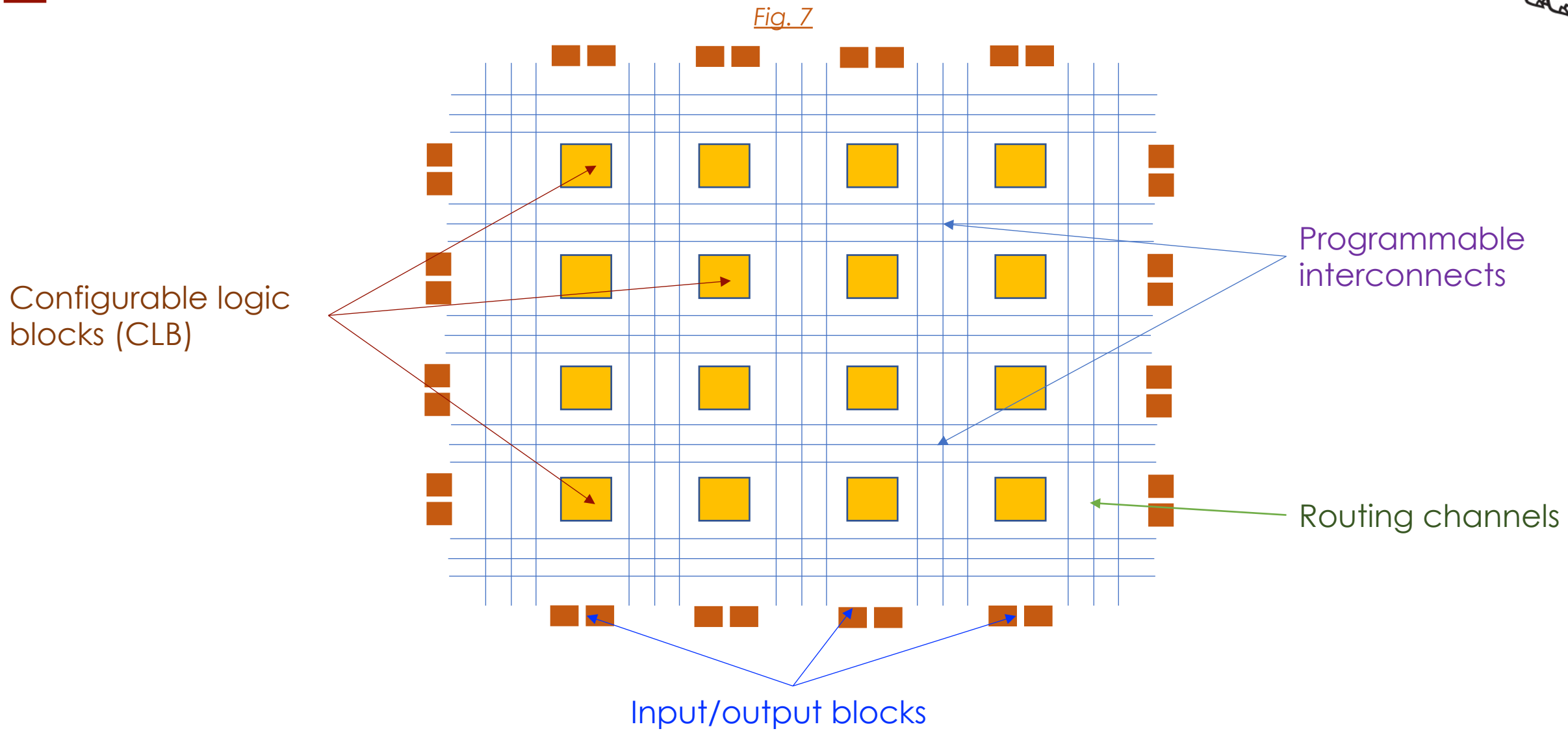
FPGA

- Contains thousands of fundamental elements called **configurable logic blocks (CLBs)** surrounded by a system of programmable interconnects, called **a fabric**, that routes signals between CLBs.



- The interconnects can readily be reprogrammed, allowing an FPGA to accommodate changes to a design or even support a new application during the lifetime of the part.
- Input/output (I/O) blocks interface between the FPGA and external devices.
- Stores its configuration information in a re-programmable medium such as static RAM (SRAM) or flash memory

FPGA Architecture



FPGA Architecture

The basic structure of an FPGA is composed of:

- Look-up table (LUT): This element performs logic operations
- Flip-Flop (FF): This register element stores the result of the LUT
- Wires: These elements connect elements to one another
- Input/Output (I/O) pads: These physically available ports get data in and out of the FPGA

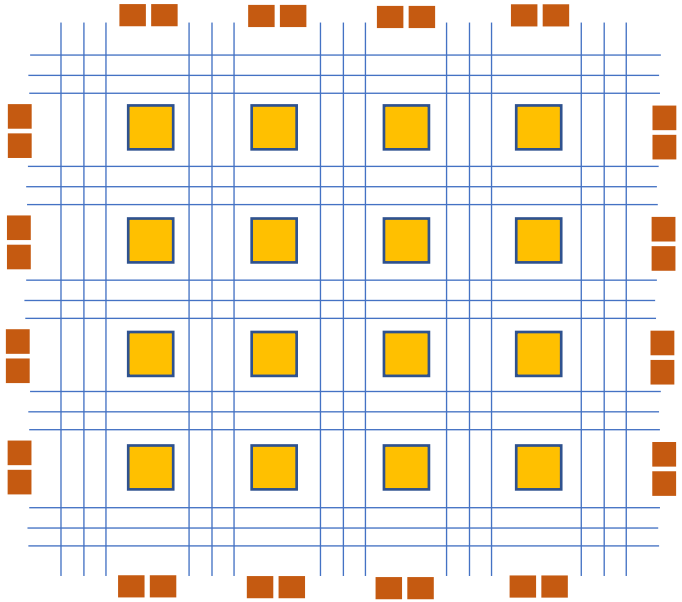
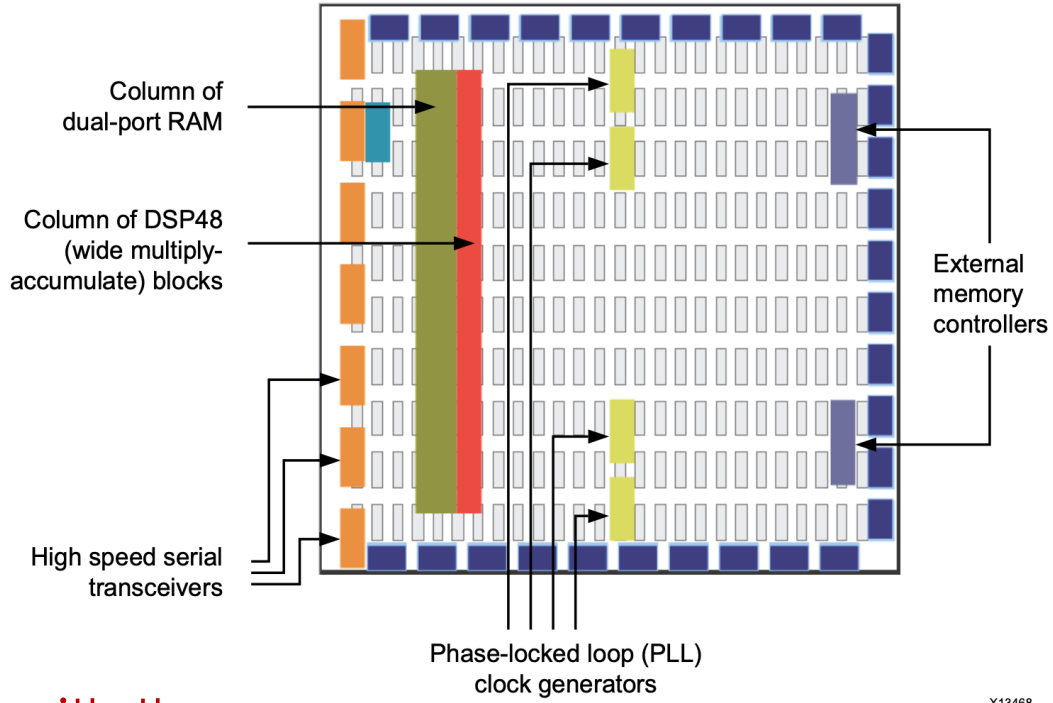


Fig. 8

FPGA Architecture

Contemporary FPGA architectures incorporate the basic elements along with additional computational and data storage blocks that increase the computational density and efficiency of the device:

- Embedded memories for distributed data storage
- Phase-locked loops (PLLs) for driving the FPGA fabric at different clock rates
- High-speed serial transceivers
- Off-chip memory controllers
- Multiply-accumulate blocks



The combination of these elements provides the FPGA with the flexibility to implement any software algorithm running on a processor

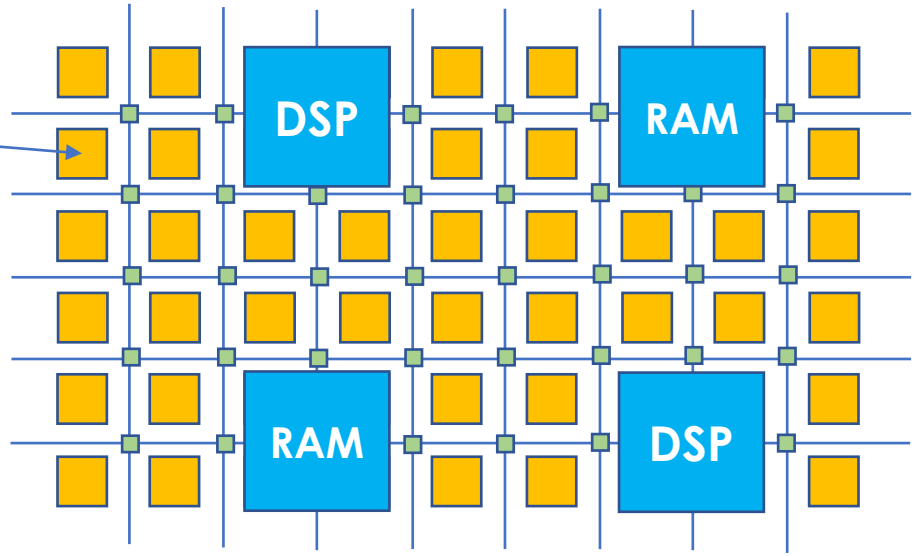
Fig. 9

FPGA Components: LUT

LUTs or logic cells:

- Basic building block of FPGA
- Capable of performing any arbitrary functions on small bitwidth inputs (N), generally N ≤ 6
- Memory location accessed by LUTs: 2^N
- LUT can be thought of as a collection of memory cells connected to a set of multiplexers.
- It can be used as both a function compute engine and a data storage element

Fig. 10



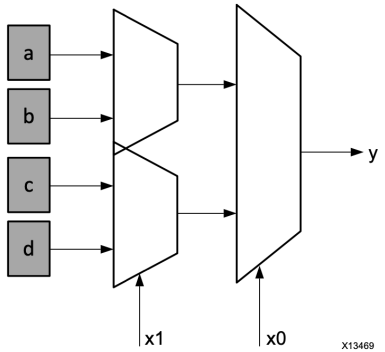


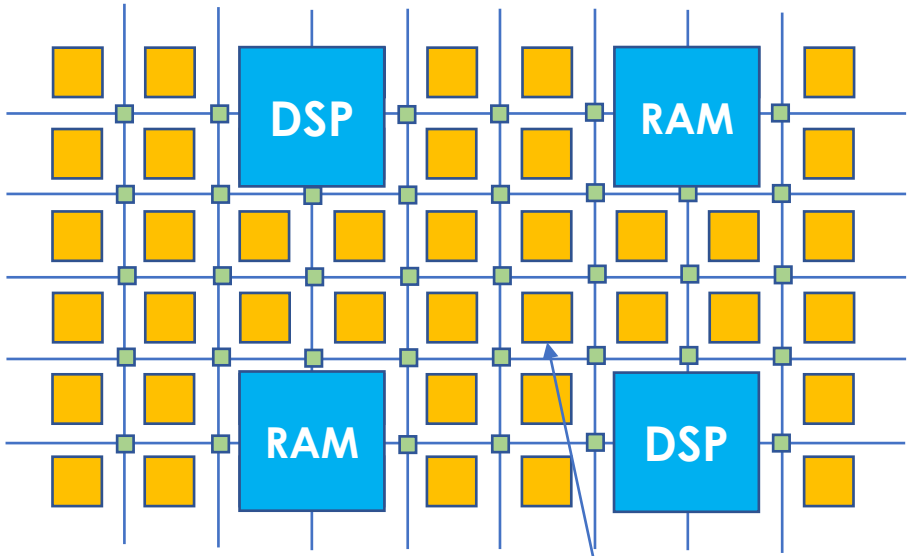
Fig. 11: Functional Representation of a LUT as Collection of Memory Cells

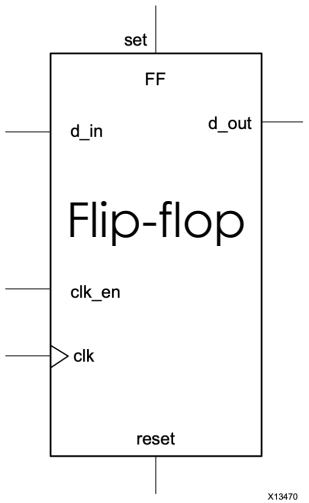
FPGA Components: FF

Flip-Flops:

- Basic storage unit within the FPGA fabric
- Always paired with a LUT to assist in logic pipelining and data storage
- Operation: value at the data input port is latched and passed to the output on every pulse of the clock
- Data is passed only when clock and clock enable = 1

Fig. 12





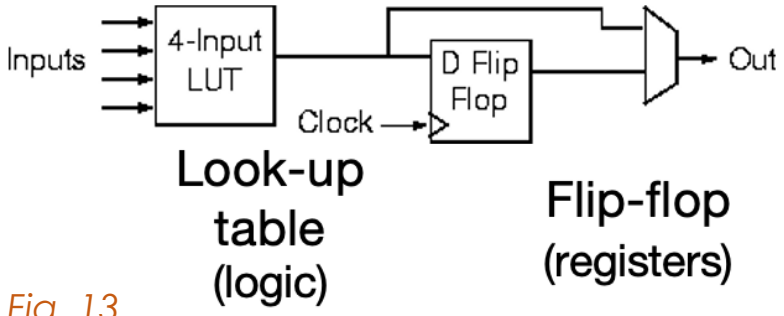


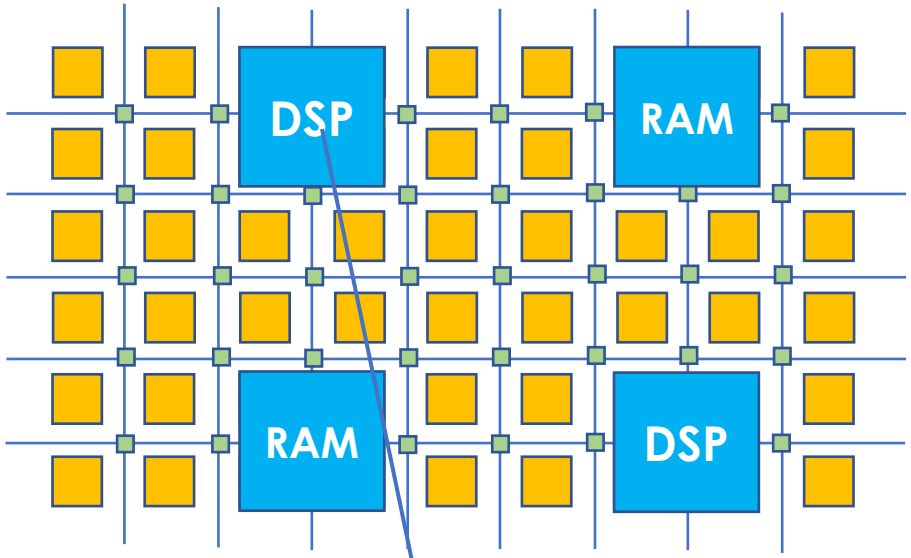
Fig. 13

FPGA Components: DSP

DSP (Digital Signal Processor) Block:

- Most complex computational block available in a FPGA
- Arithmetic Logic unit: specialized unit for multiplication and arithmetic
  - Eg: p = a x (b + d) + c
- Faster and more efficient than using LUTs for these types of operations
- Often most scarce in available resources

Fig. 14



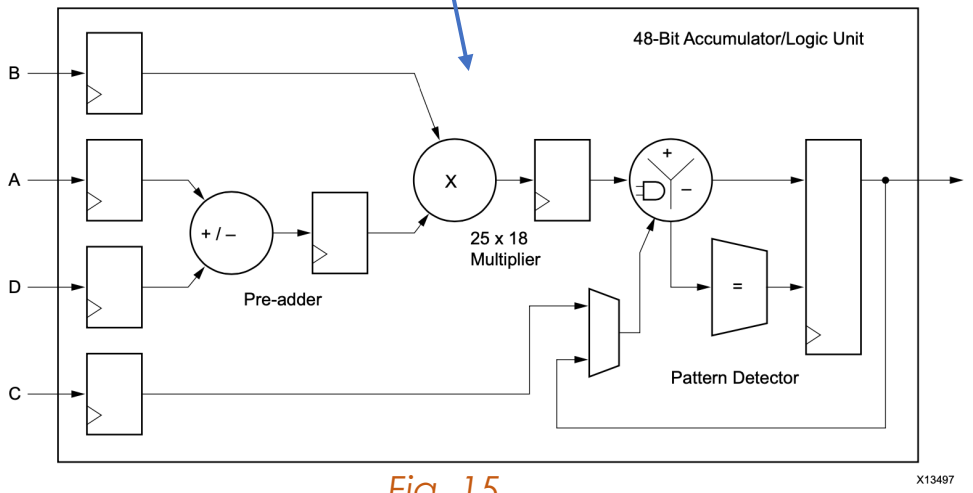
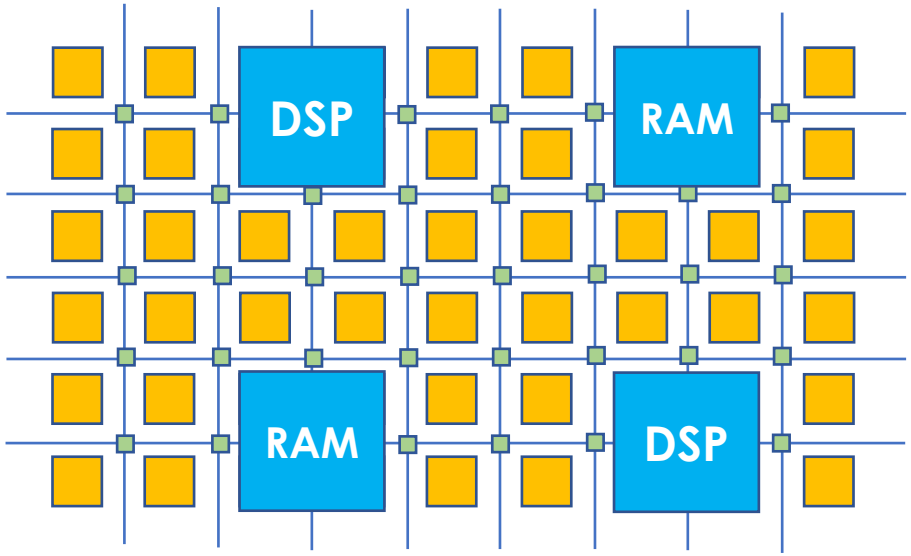


Fig. 15

FPGA Components: Storage elements

Fig. 16

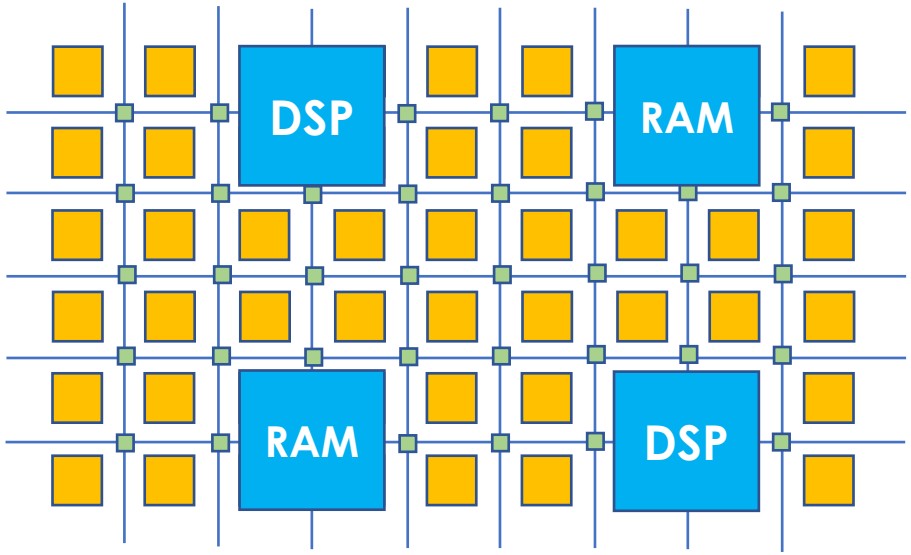


BRAMs (Block RAM)

- Embedded memory elements that can be used as Random-access-memory
- BRAM is a dual-port RAM module instantiated to provide on-chip storage for a relatively large set of data
  - can hold either 18 k or 36 k bits
- The dual-port nature of these memories allows for parallel, same-clock-cycle access to different locations
- BRAMs can implement either a RAM or a ROM. The only difference is when the data is written to the storage element.

FPGA Components: Storage elements

Fig. 17



LUTs as storage element:

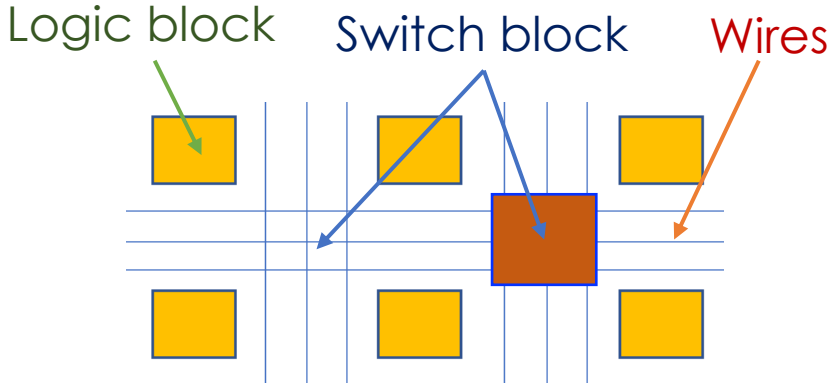
- They can be used as 64-b memories due to its structural flexibity
- Commonly referred to as distributed memories

- Fastest kind of memory available on the FPGA device, because it can be instantiated in any part of the fabric that improves the performance of the implemented circuit
- Memories using BRAMs more efficient than using LUTs

FPGA Components: Routing

- Between rows and columns of logic blocks are wiring channels
- These are programmable – a logic block pin can be connected to one of many wiring tracks through programmable switch
- Xilinx FPGA have dedicated switch block circuits for routing (flexible)
- Each wiring segment can be connected in one of many ways

Fig. 18



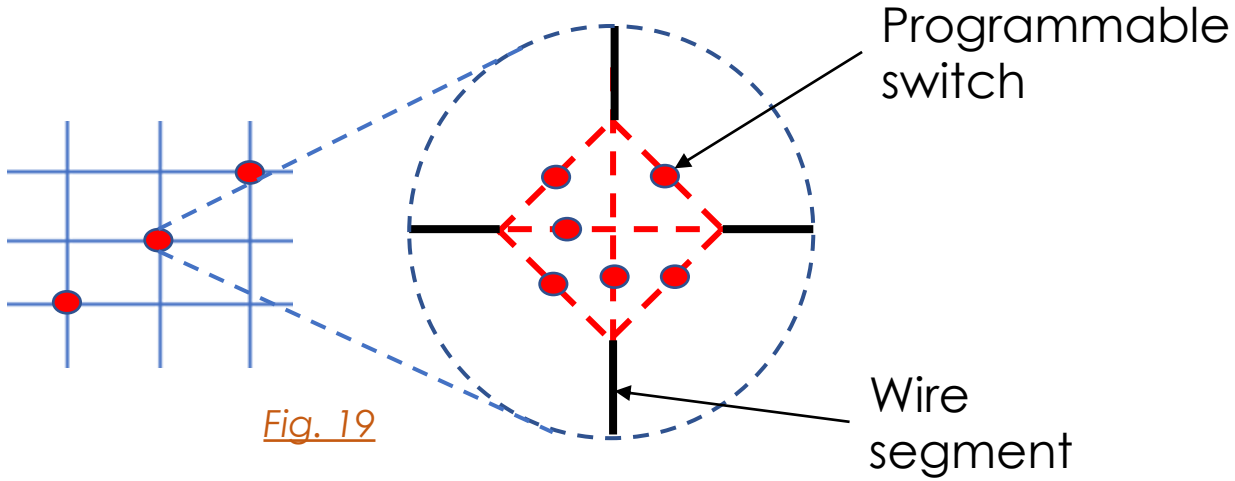


Fig. 19

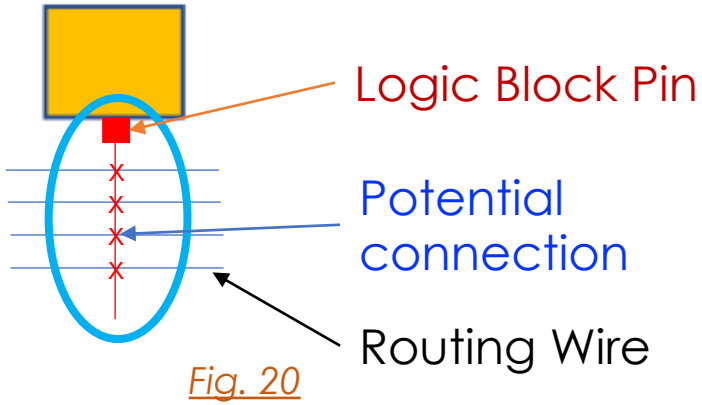


Fig. 20

The main advantage and attraction of FPGA comes from the programmable interconnect – more so than the programmable logic.

FPGA Components: I/O

There are specialised blocks for I/O

- Making FPGAs popular in embedded systems and HEP triggers

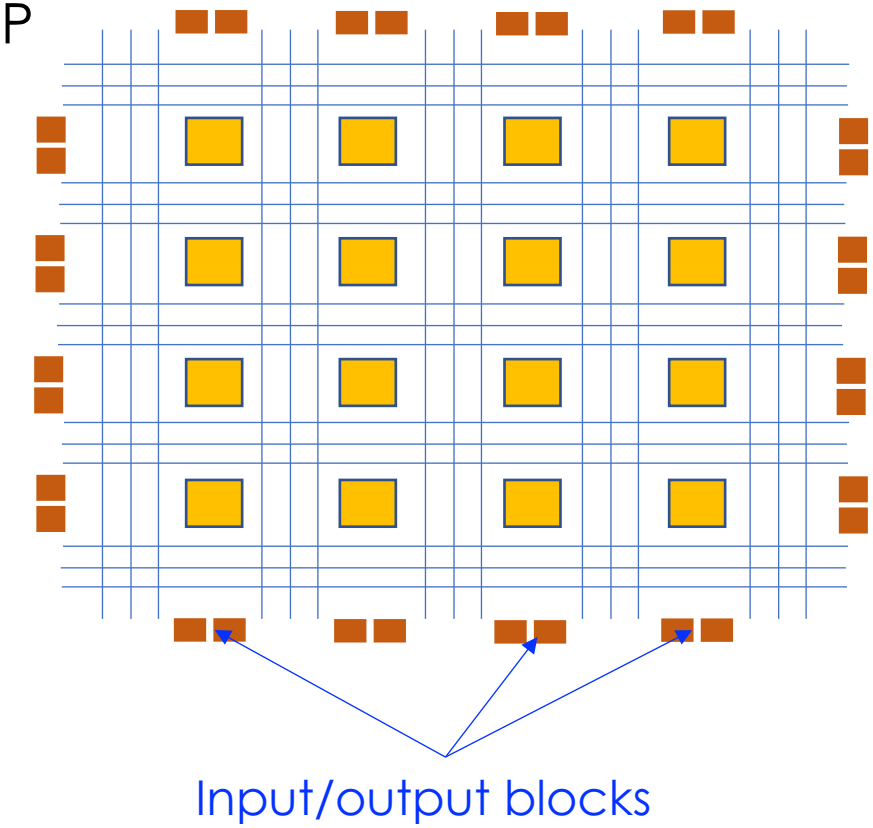
High speed transceivers

- with Tb/s total bandwidth PCIe
- (Multi) Gigabit Ethernet
- Infiniband

Support highly parallel algorithm implementations

Low power per Operation (relative to CPU/GPU)

Fig. 21



Xilinx Virtex Ultra Scale+ Product Table

COMPARE | Reset

	XCVU3P	XCVU5P	XCVU7P	XCVU9P	XCVU11P
System Logic Cells (K)	862	1,314	1,724	2,586	2,835
DSP Slices	2,280	3,474	4,560	6,840	9,216
Memory (Mb)	115.3	168.2	230.6	345.9	341
GTY/GTM Transceivers (32.75/58 Gb/s)	40/0	80/0	80/0	120/0	96/0
I/O	520	832	832	832	624

Source: https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale-plus.html#productTable

Decide wisely which FPGA to use as per your needs

Why are FPGAs fast

Fine-grained/resource parallelism

- Use the many resources to work on different parts of the problem simultaneously
- Allows us to achieve low latency

Most problems have at least some sequential aspect, limiting how low latency we can go

- But we can still take advantage of it with…

Fig. 22: Like a production line for data…

Pipeline parallelism

- Use the register pipeline to work on different data simultaneously
- Allows us to achieve high throughput

TAC-HEP 2023

FPGA WorkFlow

Workflow during FPGA development

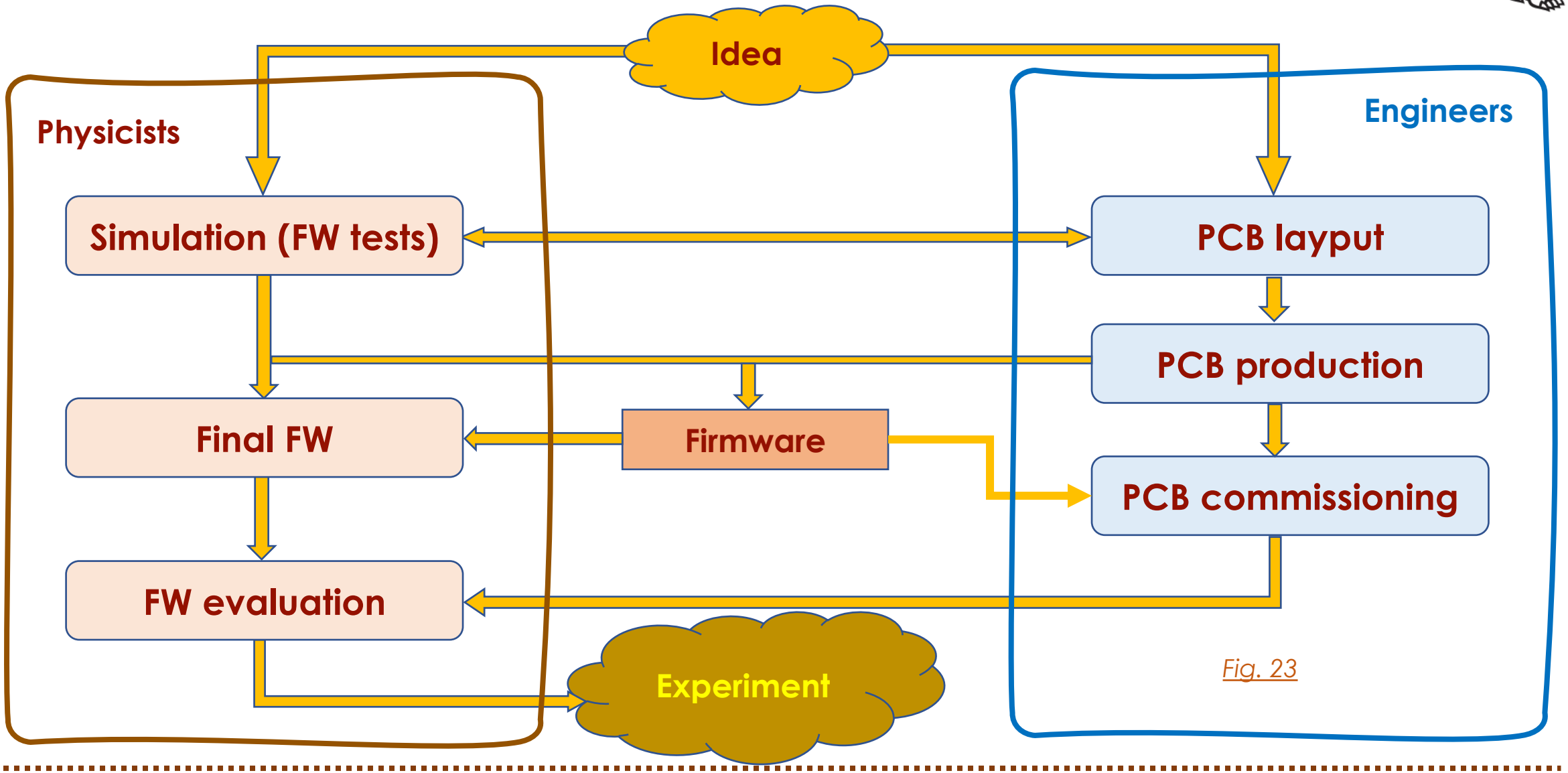


Fig. 23

Workflow during FPGA development

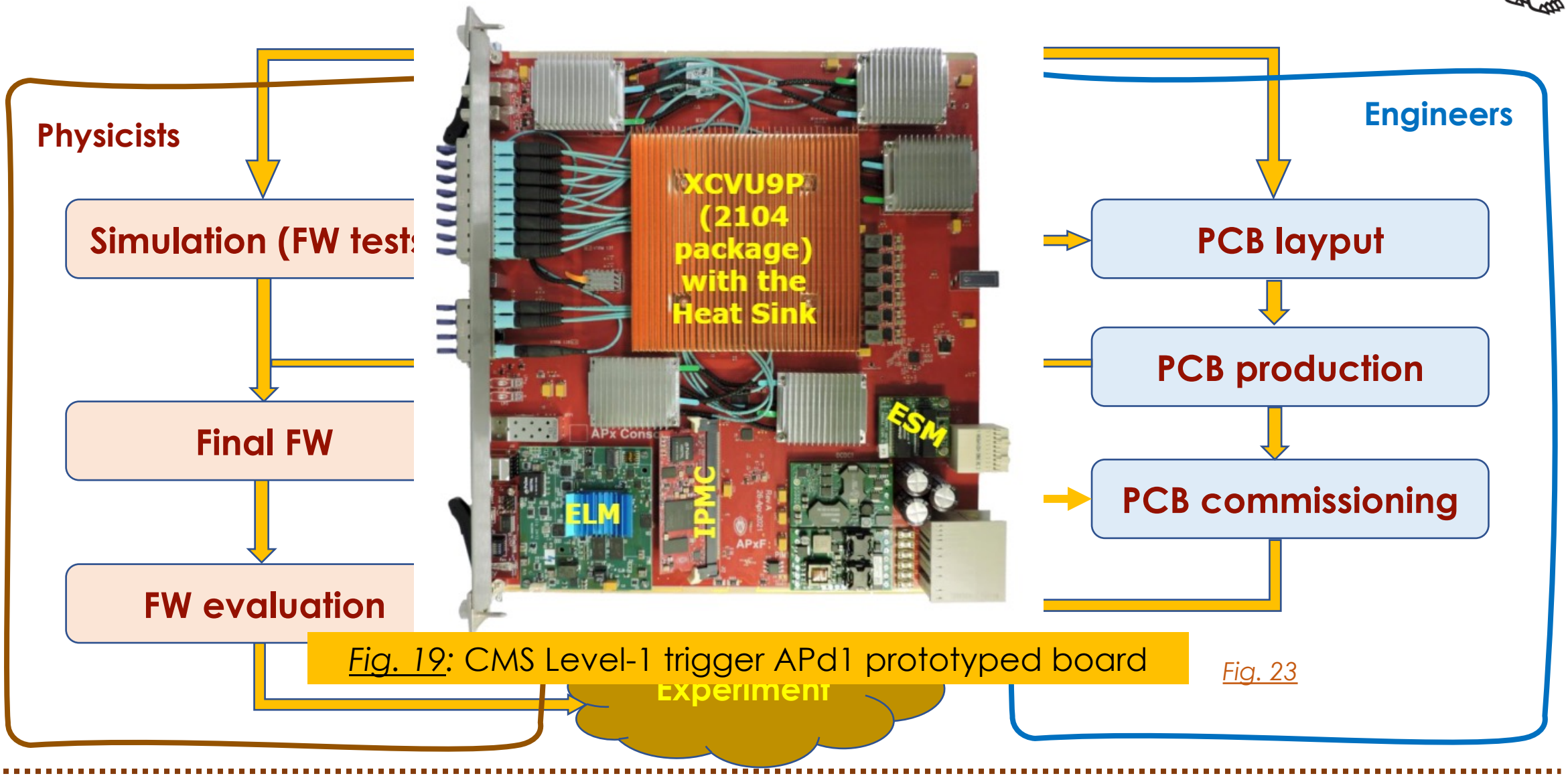
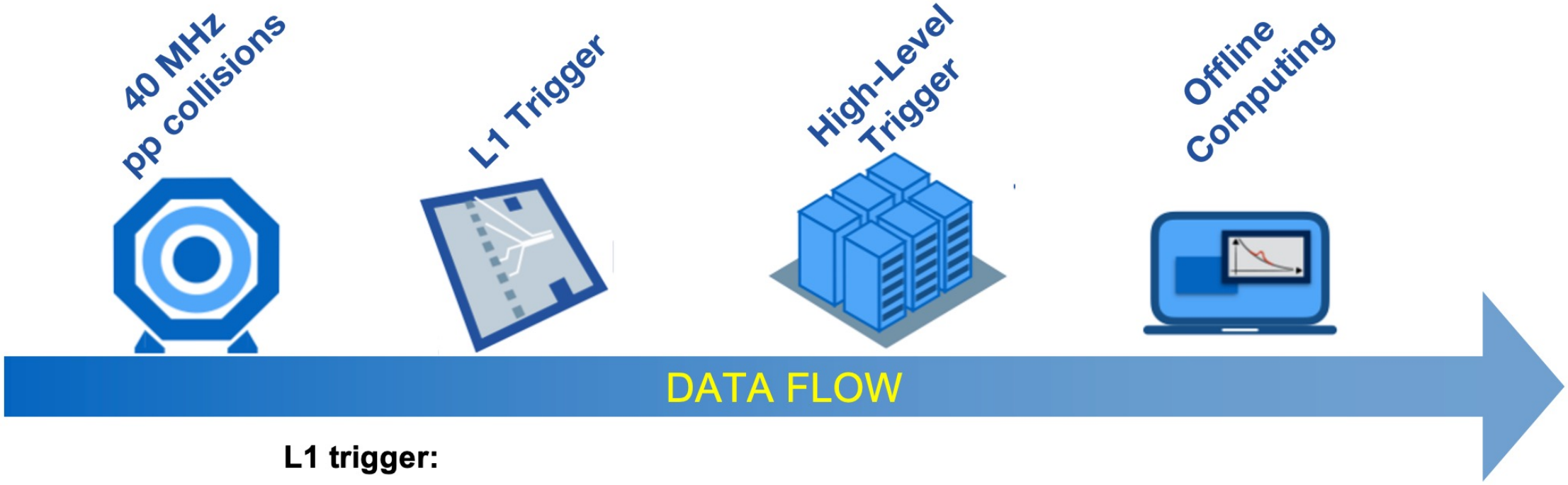


Fig. 19: CMS Level-1 trigger APd1 prototyped board

Fig. 23

LHC Experiment data Flow



L1 trigger:

- 40 MHz in / 100 KHz out
- Process 100s TB/s
- Trigger decision to be made in ≈ **10 μs**
- Coarse local reconstruction
- FPGAs / Hardware implemented

Fig. 25

TAC-HEP 2023

Questions?

TAC-HEP 2023

Additional material

Jargons

- **ICs - Integrated chip:** assembly of hundreds of millions of transistors on a minor chip
- **PCB:** Printed Circuit Board
- **LUT - Look Up Table aka 'logic'** - generic functions on small bitwidth inputs. Combine many to build the algorithm
- **FF - Flip Flops** - control the flow of data with the clock pulse. Used to build the pipeline and achieve high throughput
- **DSP - Digital Signal Processor** - performs multiplication and other arithmetic in the FPGA
- **BRAM - Block RAM** - hardened RAM resource. More efficient memories than using LUTs for more than a few elements
- **PCIe or PCI-E - Peripheral Component Interconnect Express**: is a serial expansion bus standard for connecting a computer to one or more peripheral devices
- **InfiniBand** is a computer networking communications standard used in high-performance computing that features very high throughput and very low latency
- **HLS** - High Level Synthesis - compiler for C, C++, SystemC into FPGA IP cores
- **HDL** - Hardware Description Language - low level language for describing circuits
- **RTL** - Register Transfer Level - the very low level description of the function and connection of logic gates
- **Latency** - time between starting processing and receiving the result
  - Measured in clock cycles or seconds
- **II - Initiation Interval** - time from accepting first input to accepting next input