CS250B: Modern Computer Systems

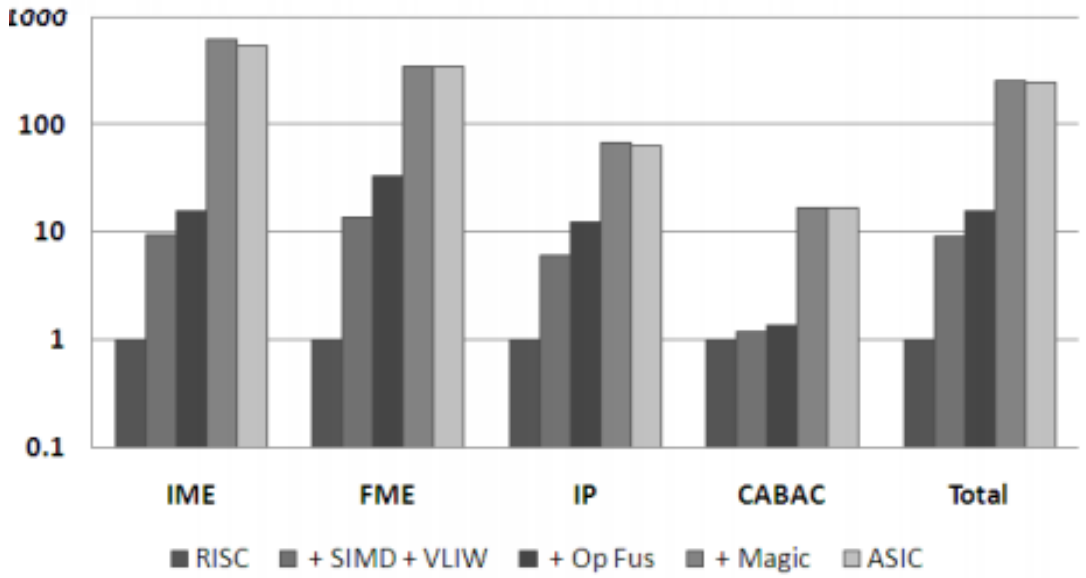
What Are FPGAs
And Why Should You Care

Sang-Woo Jun

UCI

Remember: Magic Instructions For Performance and Power Efficiency

- Need to break free from regularly structured ISAs
- Application-specific instructions, memory, interface necessary
- ASICs are expensive (Time, effort, cost, peripherals, …)
  - Makes sense for common apps



Hameed Rehan et. al., "Understanding sources of inefficiency in general-purpose chips," ISCA 2010

Scaling From CPU To ASICs

General-Purpose CPU

SIMD Extensions

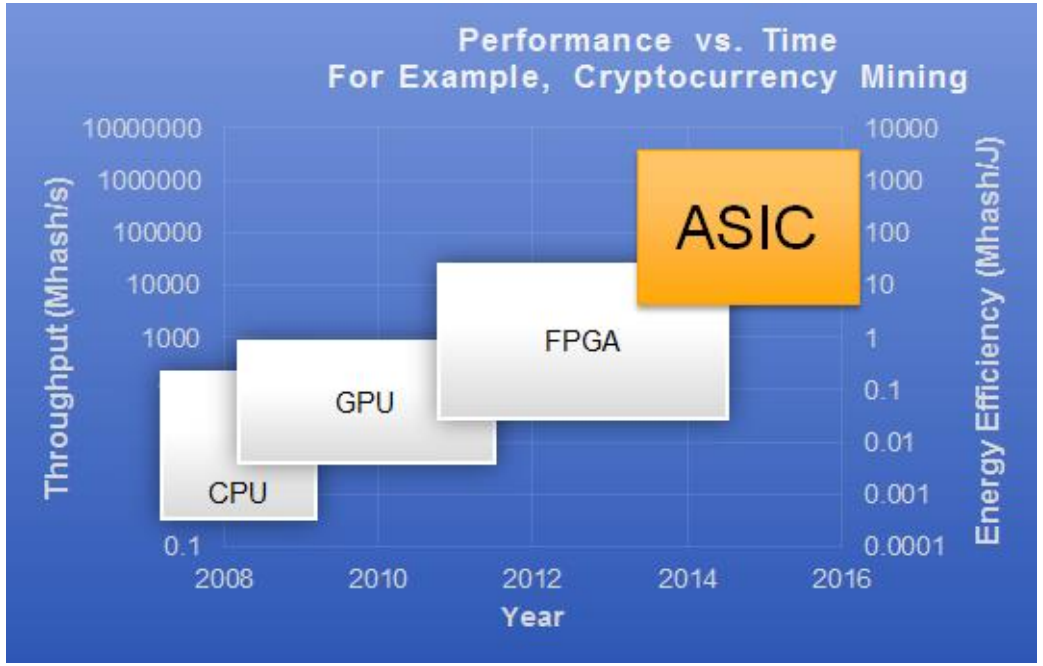
GPGPU

Field-Programmable Gate Arrays (FPGA)

Application-Specific Integrated Circuits (ASIC)

Flexibility, Programmability

Efficiency, Performance



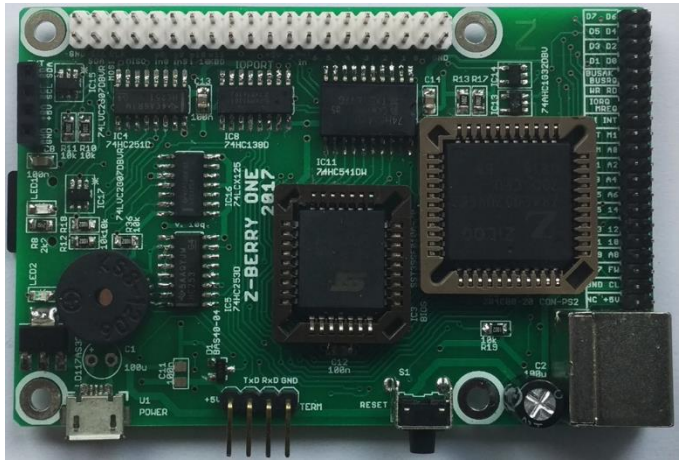
"The Future of Machine Learning Hardware", Phillip Jama, Sept. 2016

What Are FPGAs

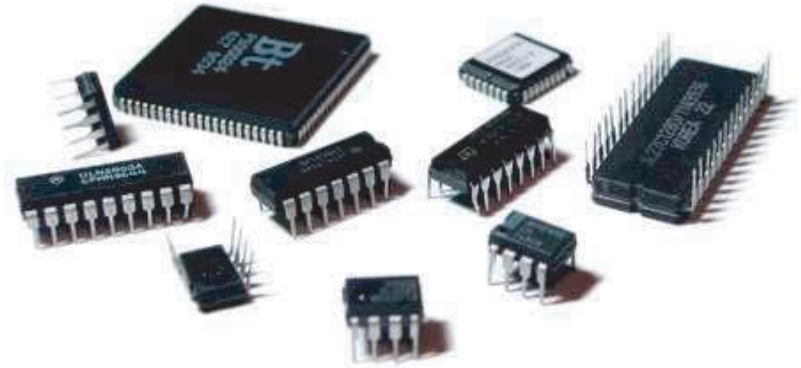
- ***Field-Programmable*** Gate Array
- Can be configured to act like any ASIC – More later!
- Can do many things, but we focus on computation acceleration

Analogy

CPU/GPU comes with fixed circuits

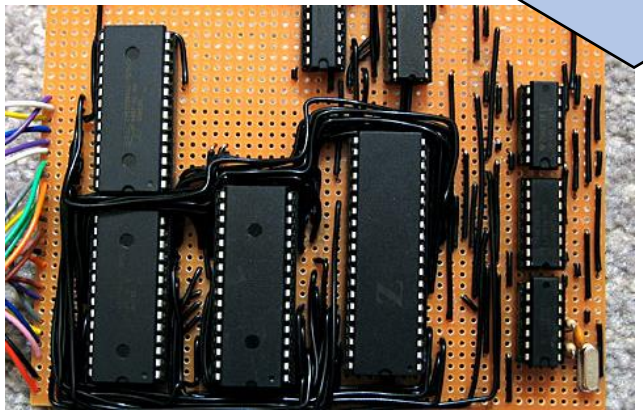


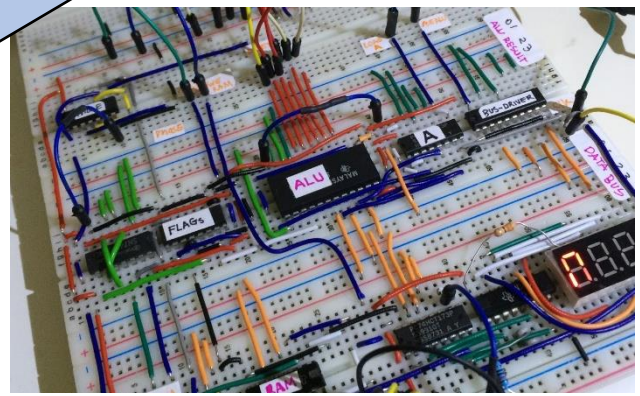
FPGA gives you a big bag of components



To build whatever

Could be a CPU/GPU!

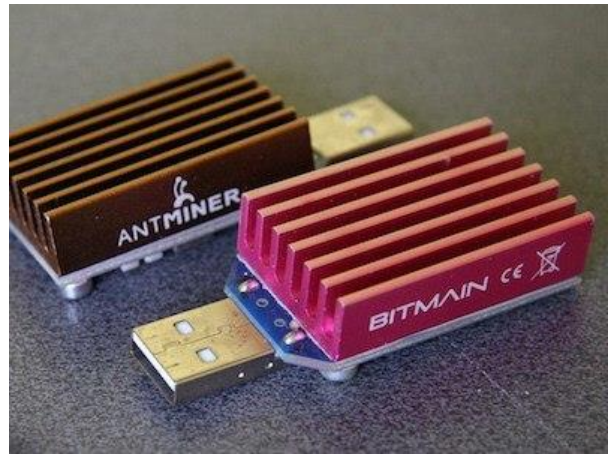




"The Z-Berry"
"Experimental Investigations on Radiation Characteristics of IC Chips"
benryves.com "Z80 Computer"
Shadi Soundation: Homebrew 4 bit CPU

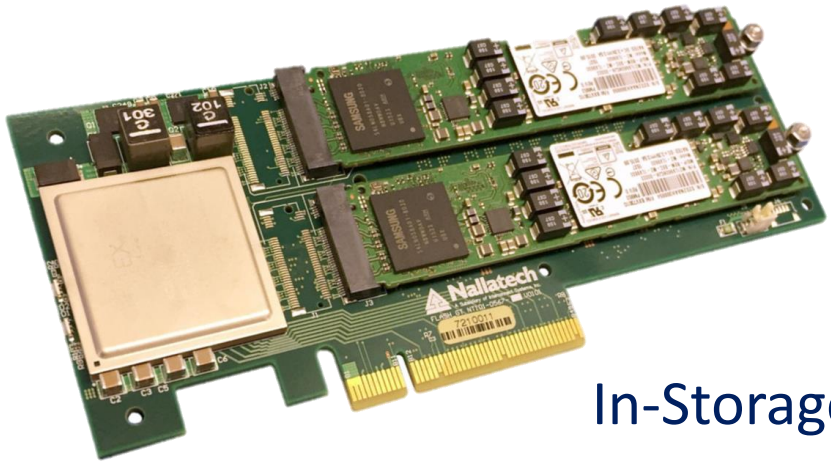
How Is It Different From ASICs

- ASIC (Application-Specific Integrated Circuit)
  - Special chip purpose-built for an application
  - E.g., ASIC bitcoin miner, Intel neural network accelerator
  - Function cannot be changed once expensively built
- \+ FPGAs can be ***field-programmed***
  - Function can be changed completely whenever
  - FPGA fabric ***emulates*** custom circuits
- \- Emulated circuits are not as efficient as bare-metal
  - ~10x performance (larger circuits, faster clock)
  - ~10x power efficiency

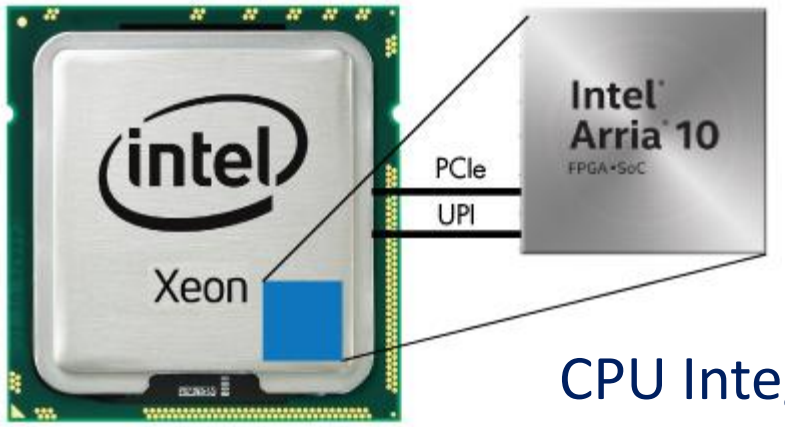


FPGAs Come In Many Forms

PCIe-Attached



In-Storage

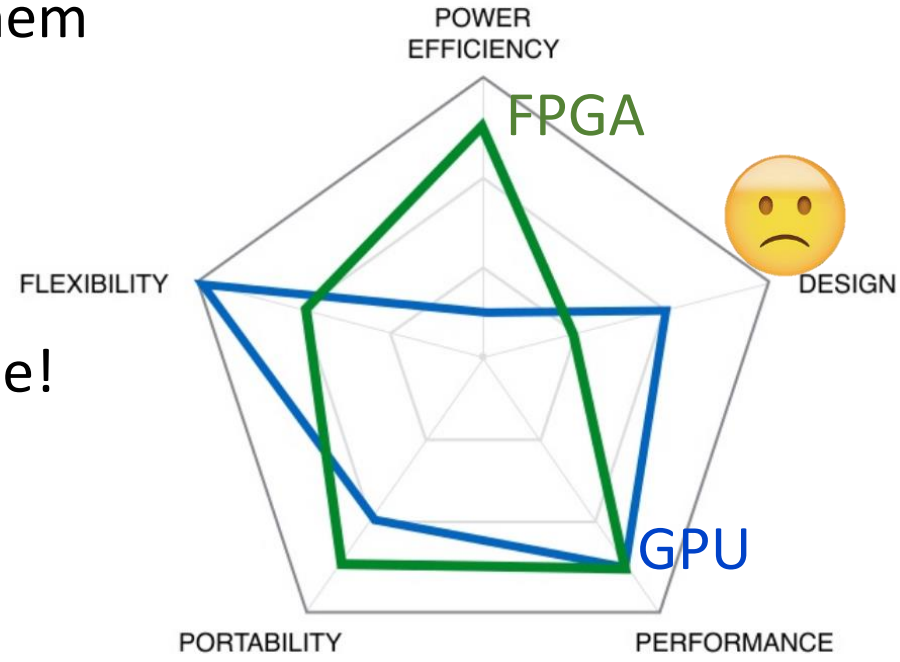


CPU Integrated

In-Network

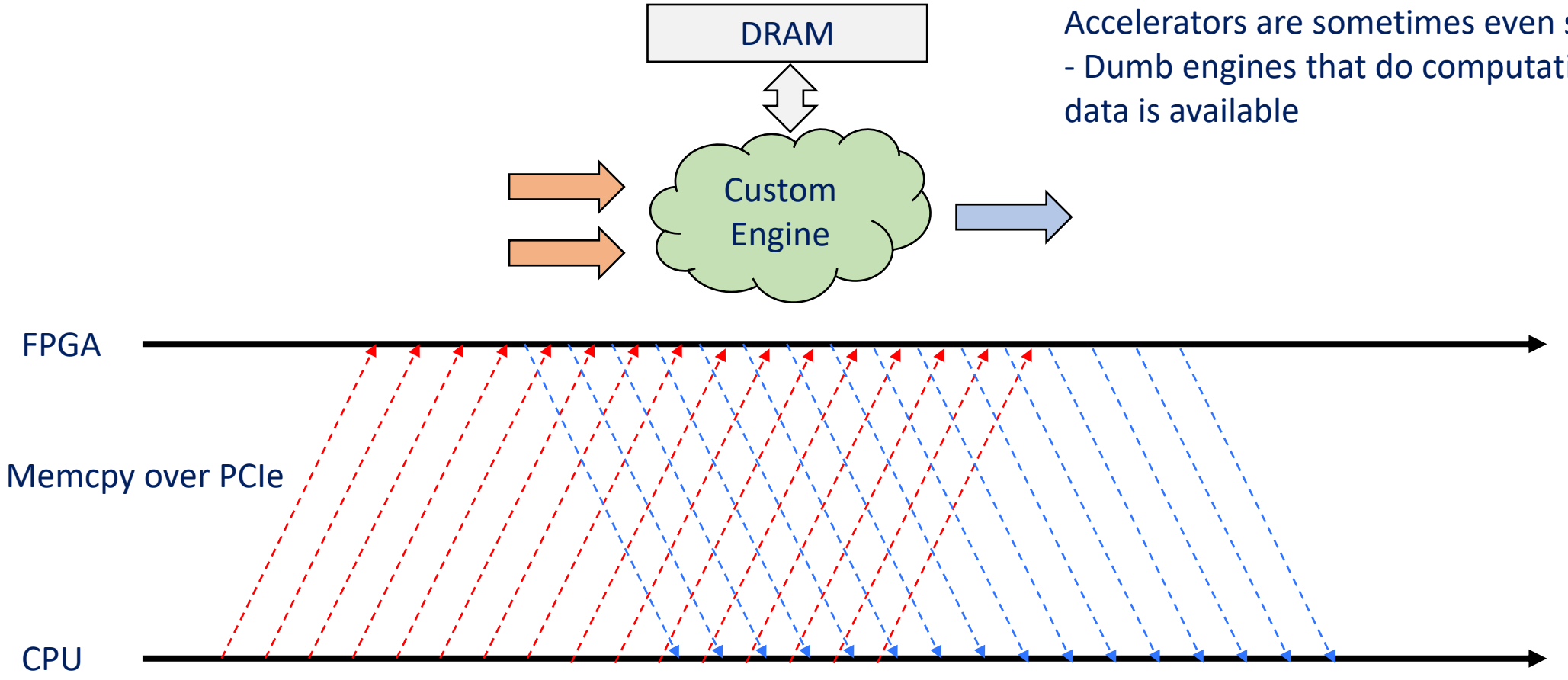
How Is It Different From CPU/GPUs

- GPU – The other major accelerator
- CPU/GPU hardware is fixed
  - "General purpose"
  - we write programs (sequence of instructions) for them
- FPGA hardware is not fixed
  - "Special purpose"
  - Hardware can be whatever we want
  - Will our hardware require/support software? Maybe!
- Optimized hardware is very efficient
  - GPU-level performance**
  - 10x power efficiency (300 W vs 30 W)



Bracco Filippo, "Rationale behind FPGA", 2017

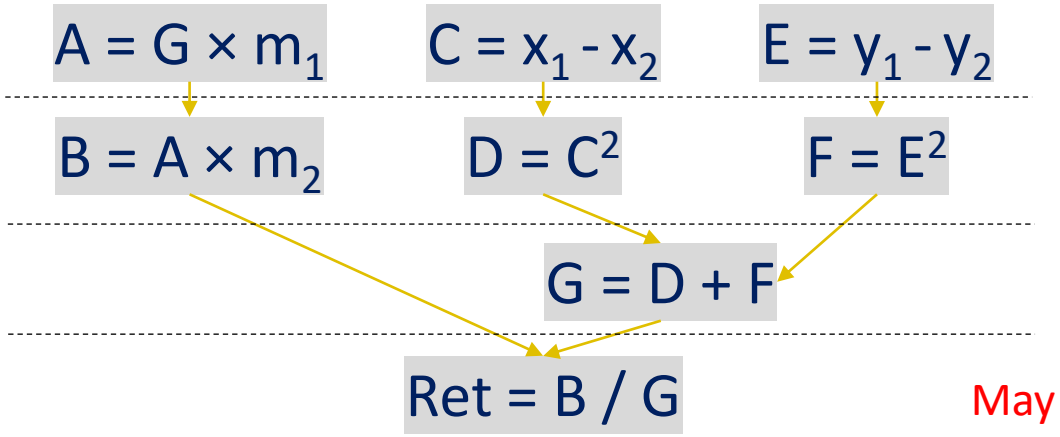
Example Use of an FPGA Accelerator



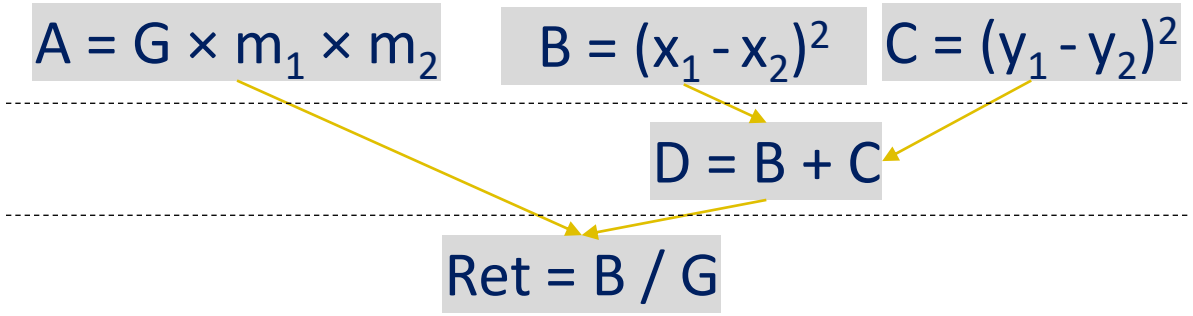
Accelerators are sometimes even stateless:
- Dumb engines that do computation when data is available

Fine-Grained Parallelism of Special-Purpose Circuits

- Example -- Calculating gravitational force: $\frac{G \times m_1 \times m_2}{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- 8 instructions on a CPU → 8 cycles**
- Special-purpose units can spawn many ALUs for parallelism



4 cycles with basic operations



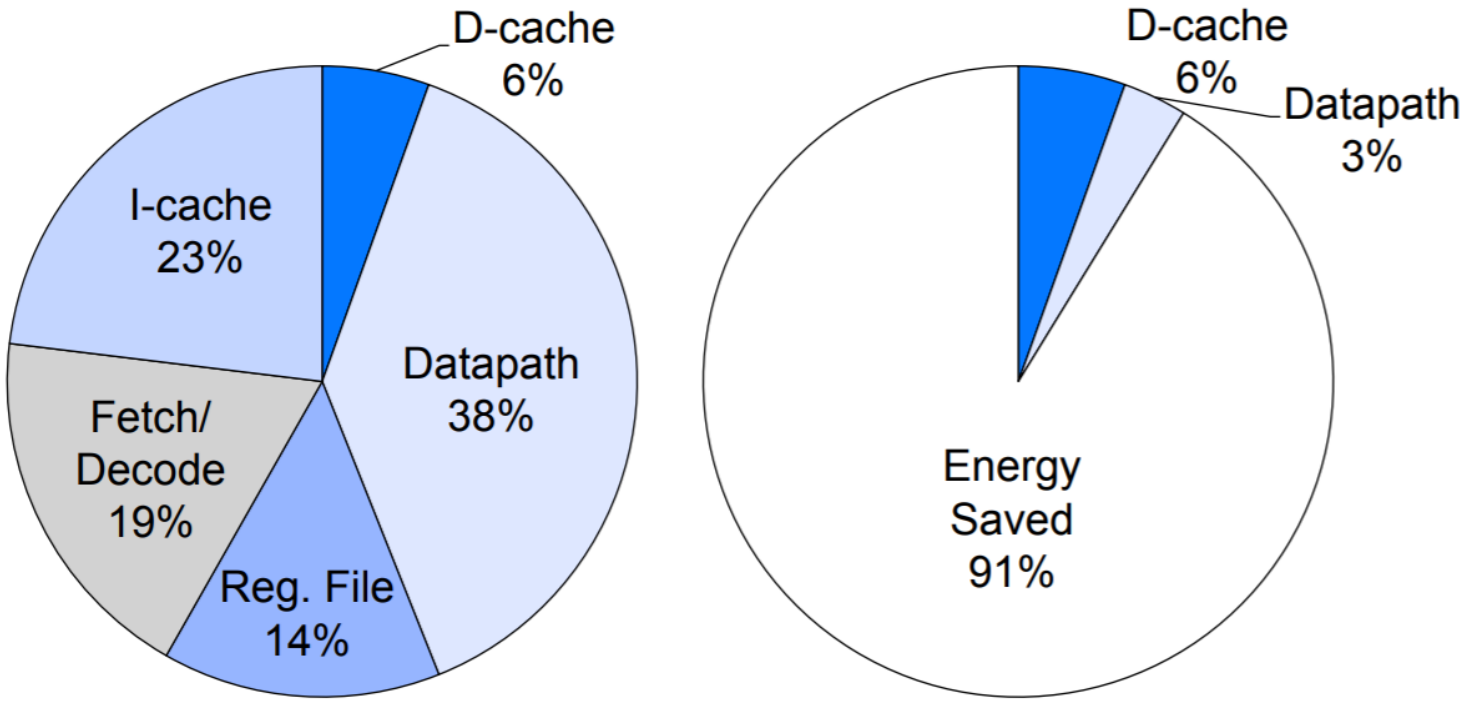
3 cycles with compound operations

May slow down clock

$$Ret = (G \times m_1 \times m_2) / ((x_1 - x_2)^2 + (y_1 - y_2)^2)$$

1 cycle with even further compound operations

Also, Remember:



RISC baseline
91 pJ/instr.

← ~11x →

C-cores
8 pJ/instr.

Coarse-Grained Parallelism of Special-Purpose Circuits

- Typical unit of parallelism for general-purpose units are threads ~= cores
- Special-purpose processing units can also be replicated for parallelism
  - Large, complex processing units: Few can fit in chip
  - Small, simple processing units: Many can fit in chip
- Only generates hardware useful for the application
  - Instruction? Decoding? Cache? Coherence?

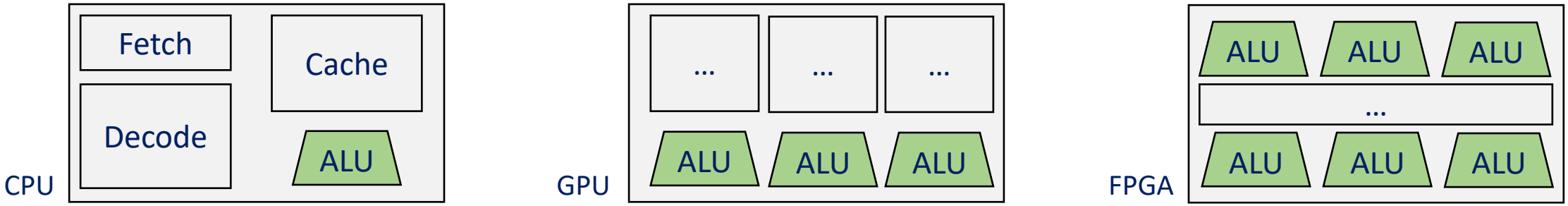
Hurdles of FPGA Programming

- Very close to actually designing a chip, with all associated difficulties
  - Signal propagation delays, clock speeds
  - Circuit design with acceptable propagation delay
  - Explicit pipeline design for performance
  - On-chip resource management (Number of available transistors, SRAM, …)

- High-level tools exist, but not yet very effective
  - Often trade-off between ease of programming and performance/utilization
  - More on this later!

Prominent Technology For Performance Scaling

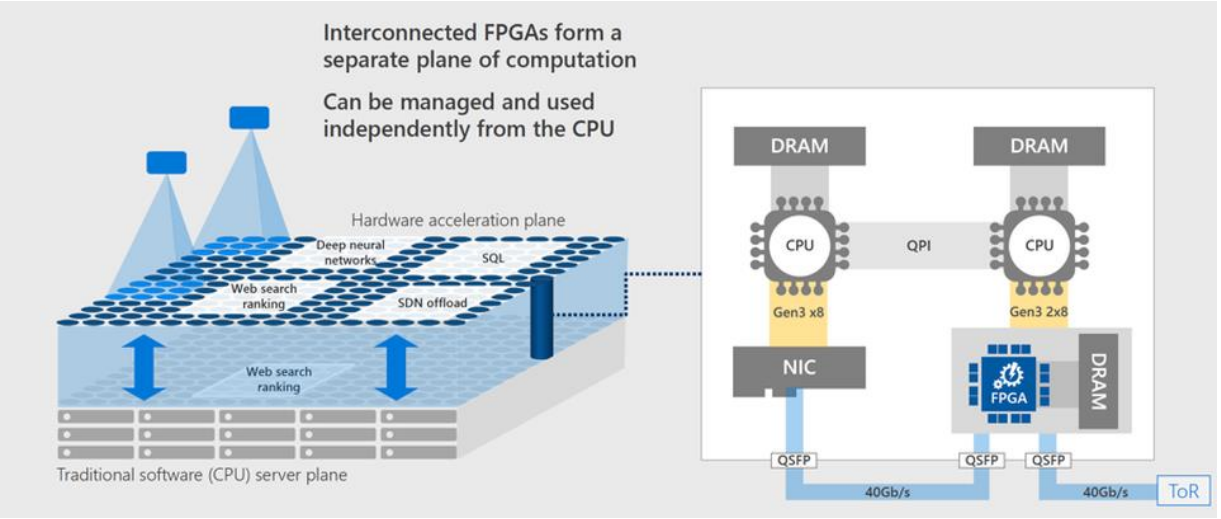
- Efficient use of increasing transistor budget
  - Platform for specialized architectures
- Reconfigurable based on application/algorithm changes
  - Kind of like software running on a hardware processor
  - Single FPGA hardware can be shared between different applications
- Common operations often embedded using ASIC blocks
  - High-performance, low-power, small size
  - Arithmetic, shifters, neural network operations, high-speed communications, etc



Prominent Technology For Performance Scaling

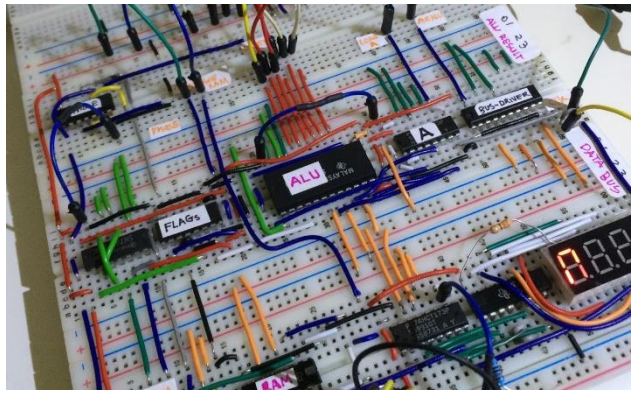
- Intel acquires Altera (no. 2 in FPGA industry) for $16.7 B (2015)
  - Largest acquisition in Intel history
- AMD acquiring Xilinx (no. 1 in FPGA industry) (2021)
  - Stock-swap valued at $35 B

Unfortunately, learning curve is still quite high…



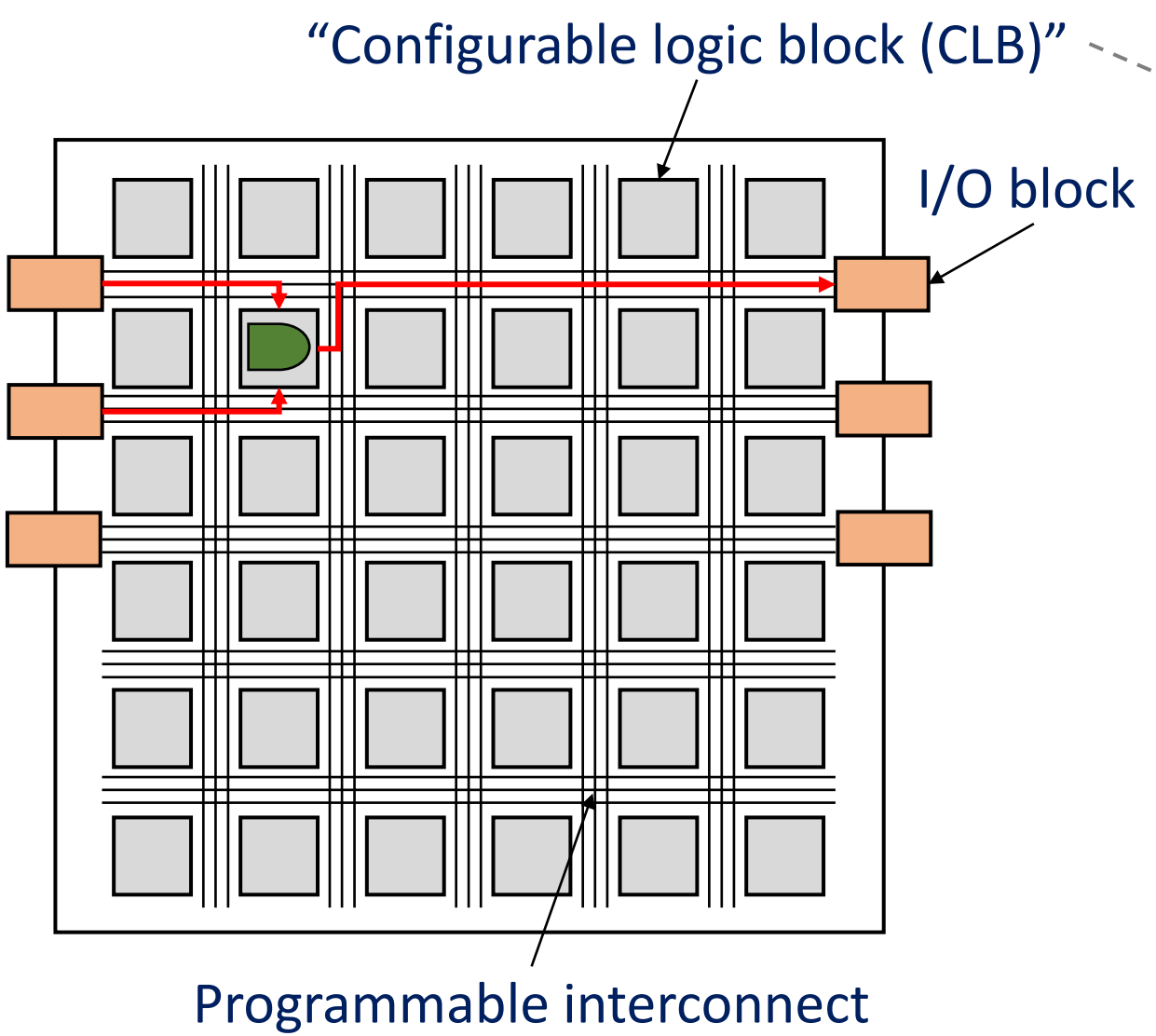
FPGA Architecture Goals

- Reconfigure a physical chip to act like a different circuit
- Transistors within an integrated circuit are fixed during fabrication!
  - How can FPGAs change themselves?

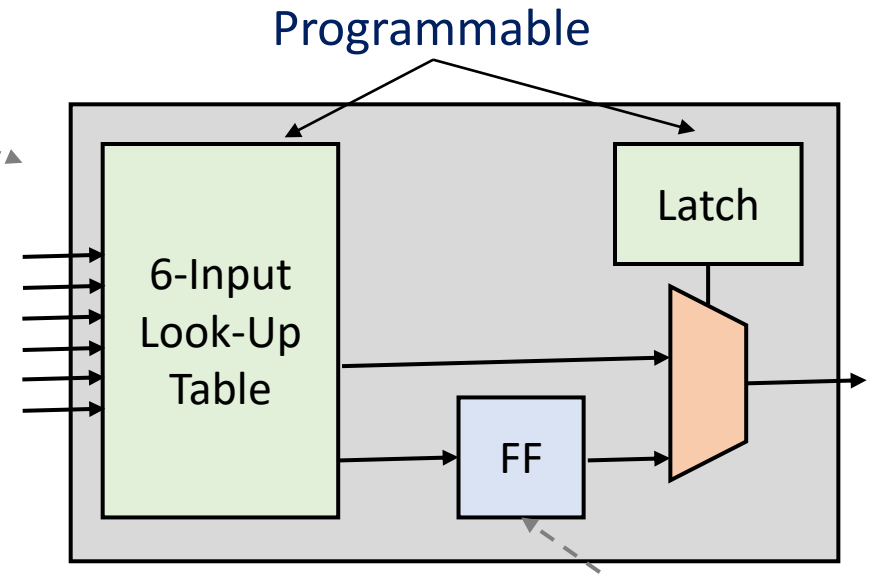


Shadi Soundation: Homebrew 4 bit CPU

Basic FPGA Architecture



≈

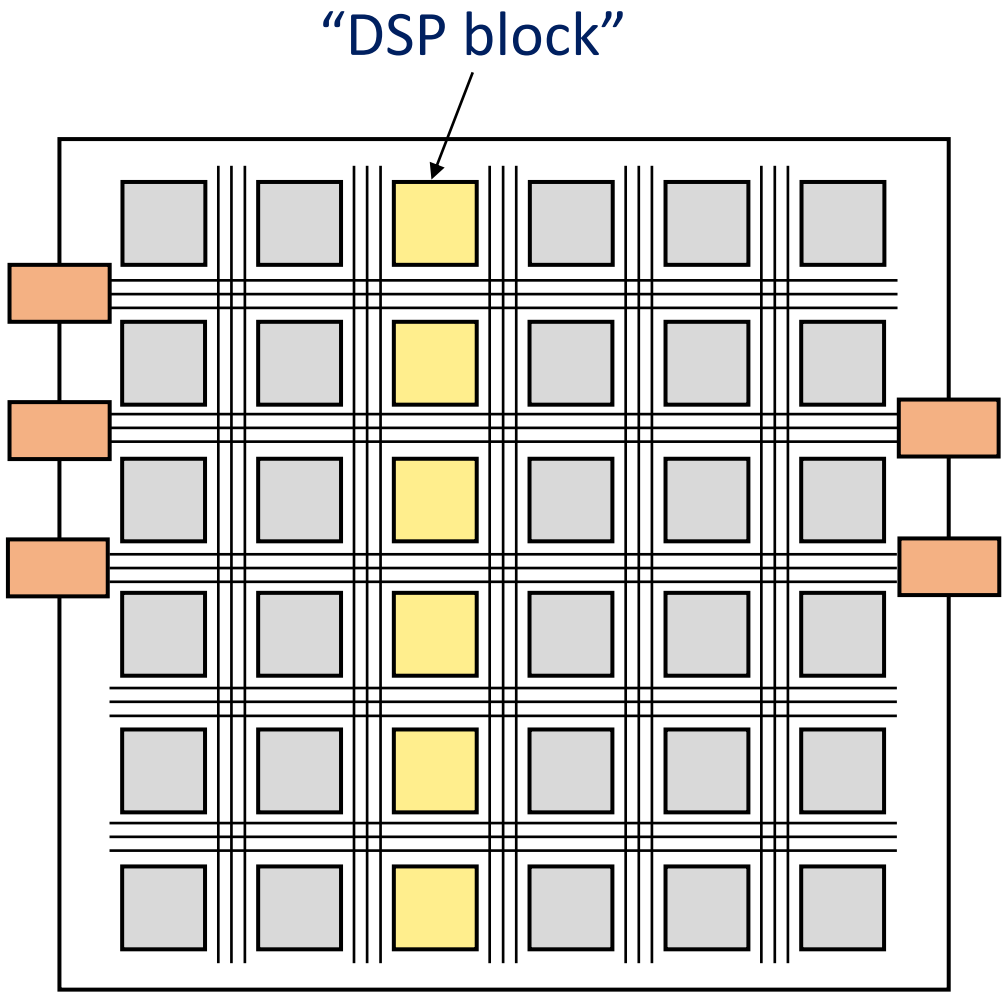


Ex) 2-LUT for “AND”

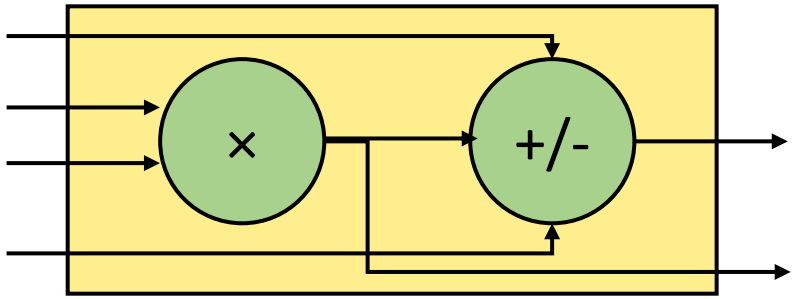
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Stores state for sequential circuit construction

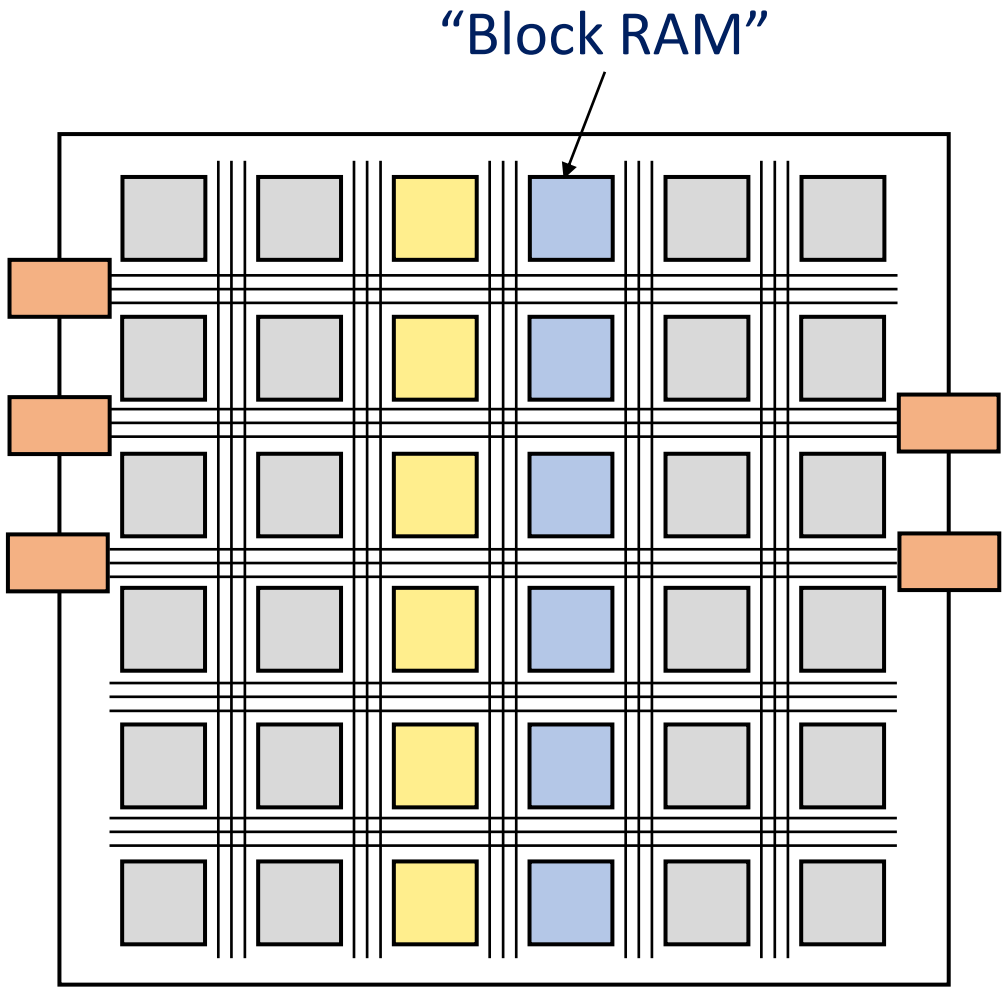
Basic FPGA Architecture – DSP Blocks



- CLBs act as gates – Many needed to implement high-level logic
- Arithmetic operation provided as efficient ALU blocks
  - “Digital Signal Processing (DSP) blocks”
  - Each block provides an adder + multiplier

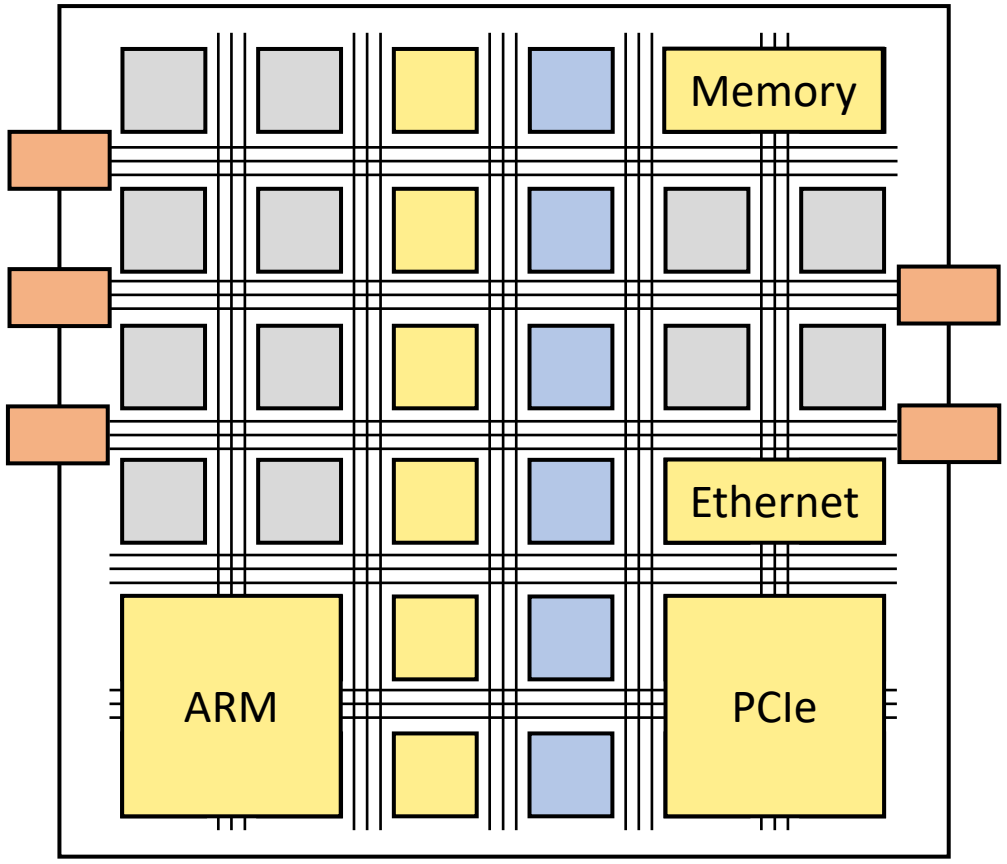


Basic FPGA Architecture – Block RAM



- CLB can act as flip-flops
  - (~N bit/block) – tiny!
- Some on-chip SRAM provided as blocks
  - ~18/36 Kbit/block, MBs per chip
  - Massively parallel access to data → multi-TB/s bandwidth

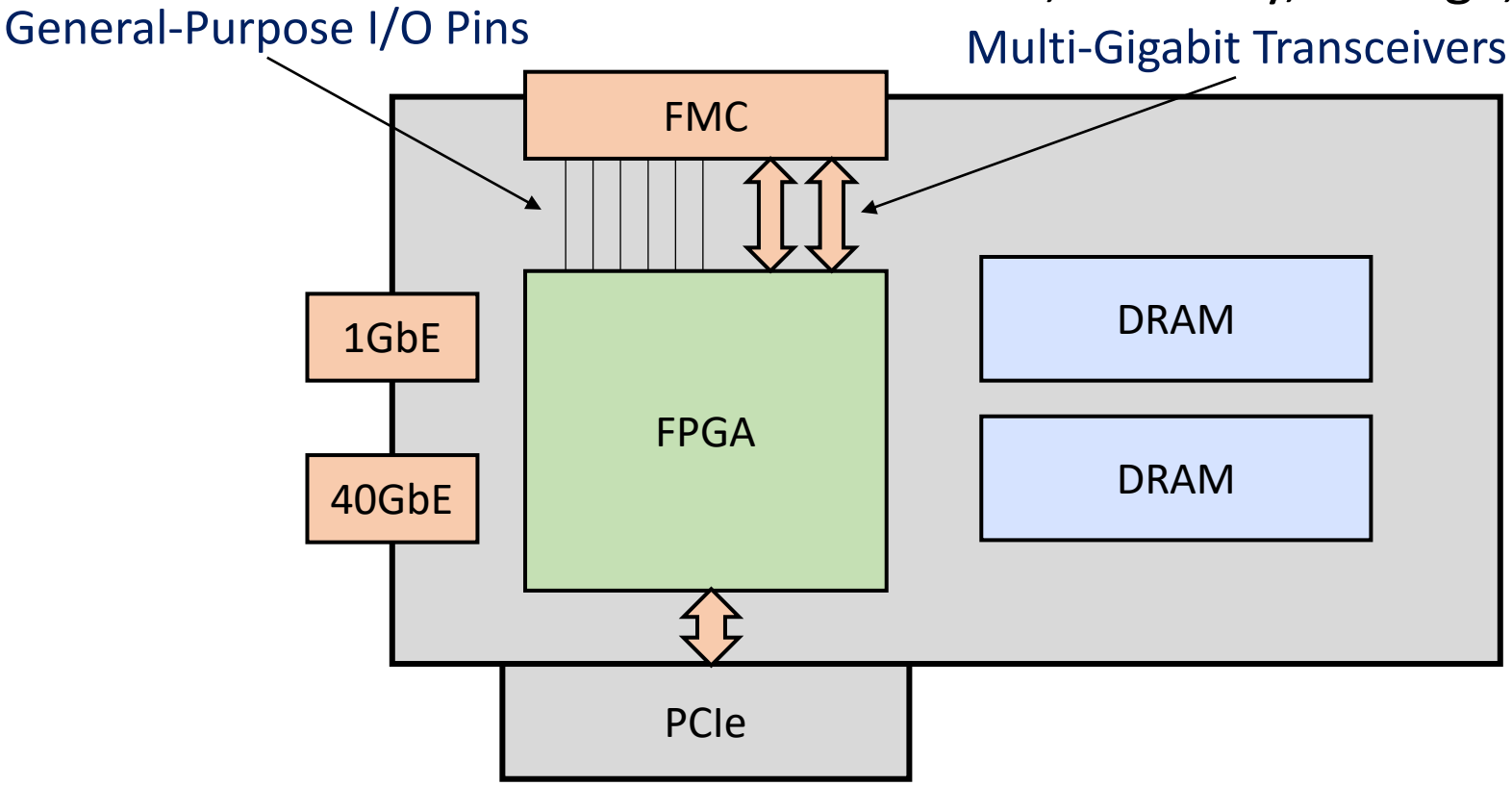
Basic FPGA Architecture – Hard Cores



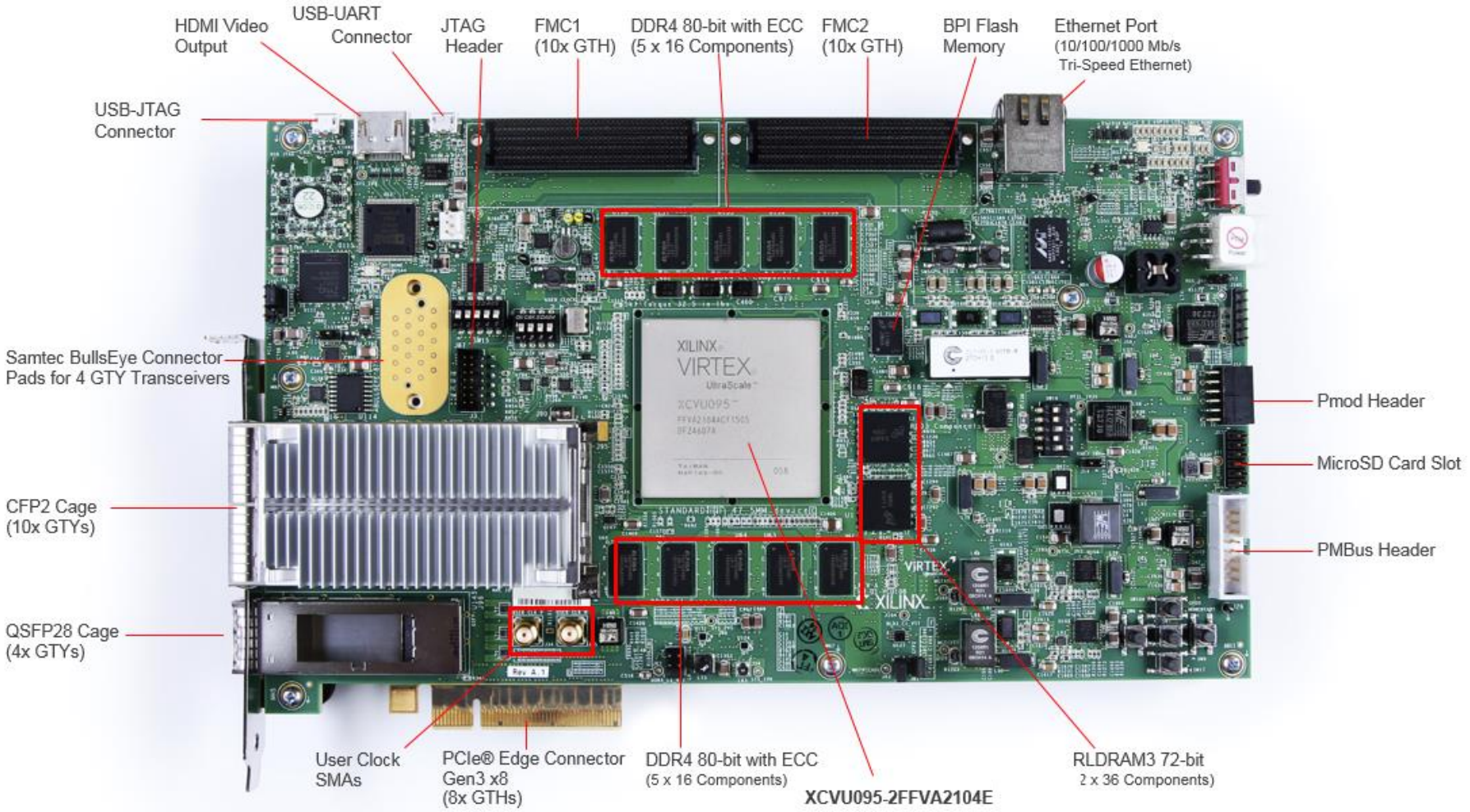
- Some functions are provided as efficient, non-configurable “hard cores”
  - Multi-core ARM cores (“Zynq” series)
  - Multi-Gigabit Transceivers
  - PCIe/Ethernet PHY
  - Memory controllers
  - …

Example Accelerator Card Architecture

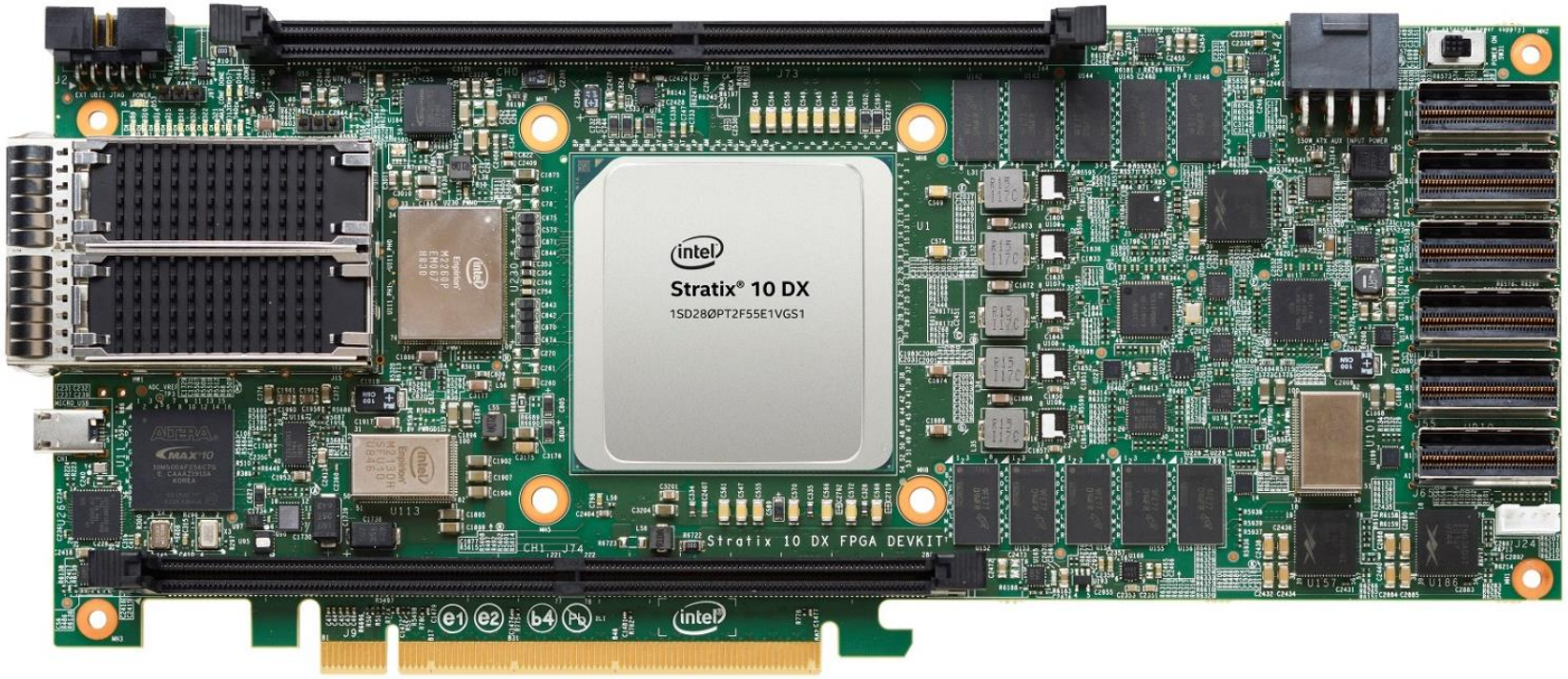
- "FPGA Mezzanine Card" Expansion
  - Network Ports, Memory, Storage, PCIe, …



Example Development Board (Xilinx VCU108)



Example Development Board (Intel Stratix 10)



Example Accelerator Card (Xilinx Alveo U250)