Introduction to FPGA

And FPGA-related papers in recent systems conferences

Jiacheng Ma

FPGAs are getting increasingly popular.

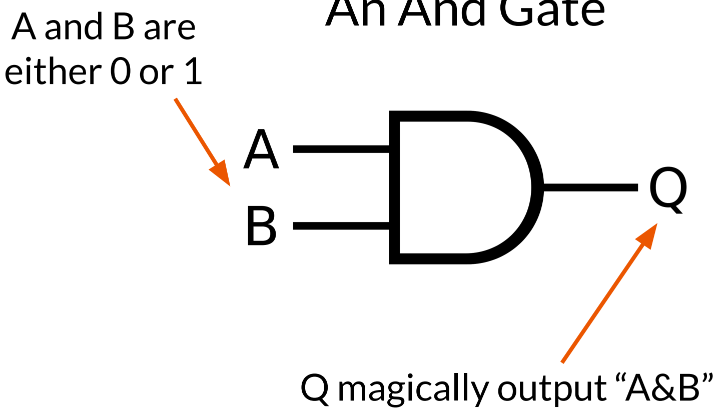
- Accelerate neural networks, graph processing, image processing, etc
- Deployed by Azure, AWS, Alibaba, etc

Programing FPGAs is designing hardware.

- Unlike talks on W1 and W2
- Fundamentally different than software
  - No instruction
  - No program counter
  - No thread
  - No ...

Let's review some elementary school physics

An And Gate



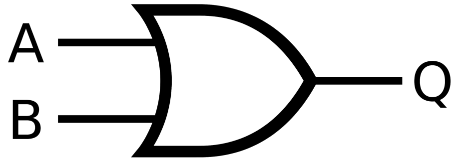
The Truth Table

A	B	Q
1	0	0
1	1	1
0	1	0
0	0	0

Figure from https://en.wikipedia.org/wiki/AND_gate

Let's review some elementary school physics

An Or Gate



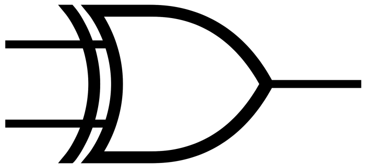
The Truth Table

A	B	Q
1	0	1
1	1	1
0	1	1
0	0	0

Figure from https://en.wikipedia.org/wiki/OR_gate

Let's review some elementary school physics

An Xor Gate



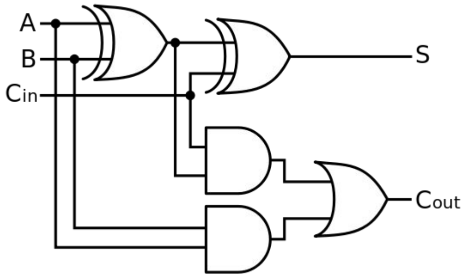
The Truth Table

A	B	Q
1	0	1
1	1	0
0	1	1
0	0	0

Figure from https://en.wikipedia.org/wiki/XOR_gate

Let's review some elementary school physics

A 1-Bit Adder



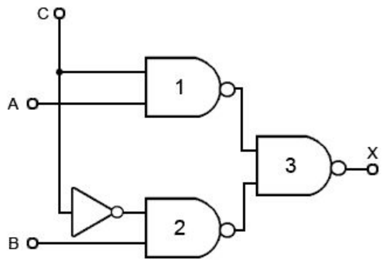
The Truth Table

Cin	A	B	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure and table from https://microcontrollerslab.com/cd4008-4-bit-full-adder-ic-pinout-features-example-datasheet/

Let's review some elementary school physics

A Selector (MUX)



The Truth Table

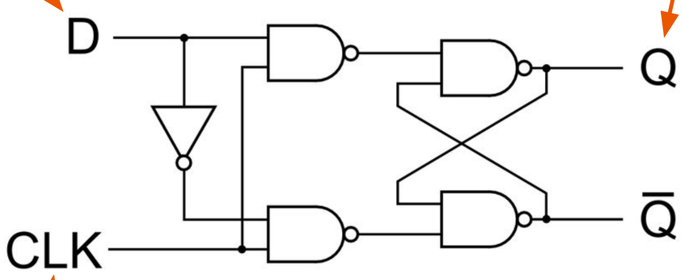
C	B	A	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Figure and table from https://learnabout-electronics.org/Digital/dig42.php

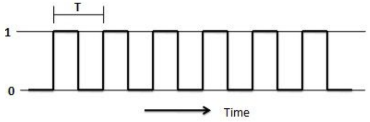
A D-Flip-Flop

The data input

Q saves the data of D for a cycle



The clock input



A DFF, also called a “register”

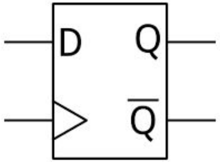
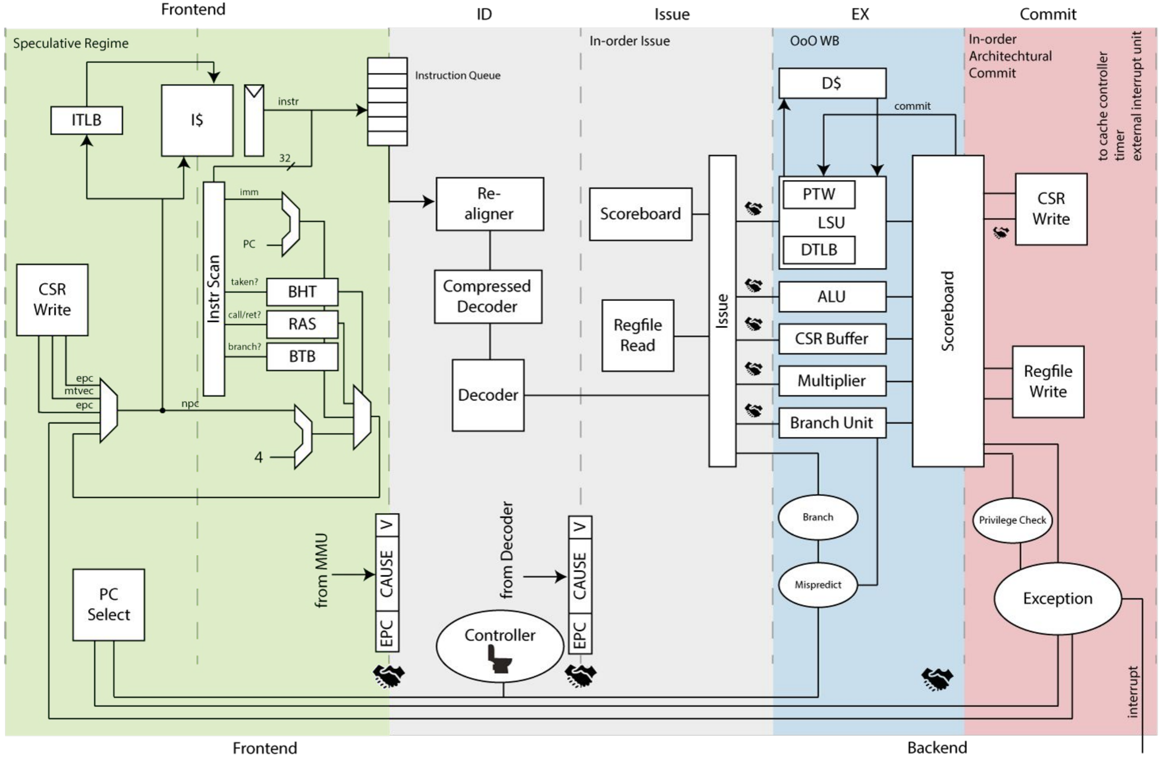


Figure from https://www.physicsforums.com/threads/need-help-with-d-flip-flop.534311/

DFFs as Pipeline Registers



The pipeline design of CVA6 RISC-V processor
https://github.com/openhwgroup/cva6

DFFs are used between different stages

But wait, where's the FPGA?

Field Programmable Gate Array

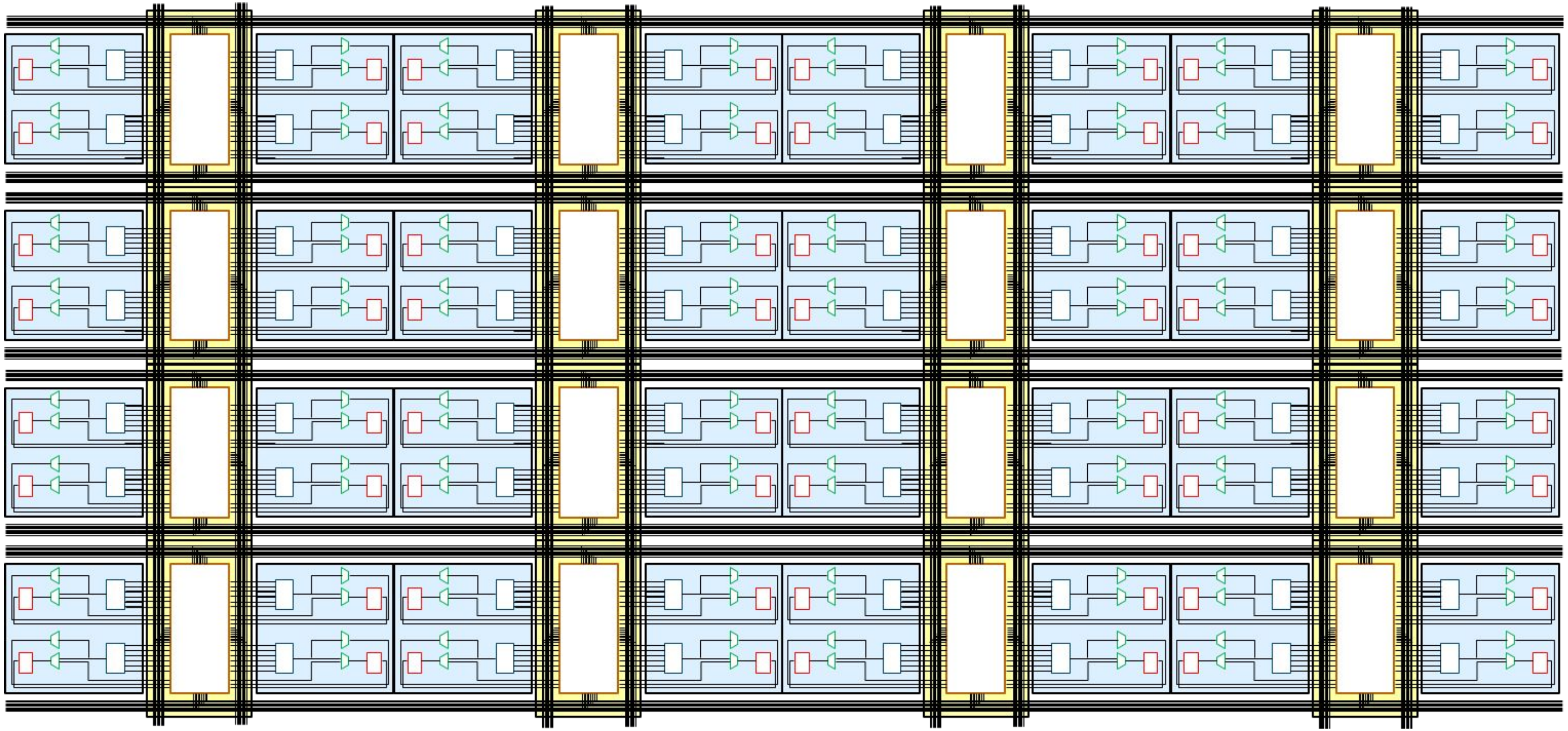


Figure and table from https://www.rapidwright.io/docs/FPGA_Architecture.html

Lock Up Table: something like a truth table

Flip-Flop (register)

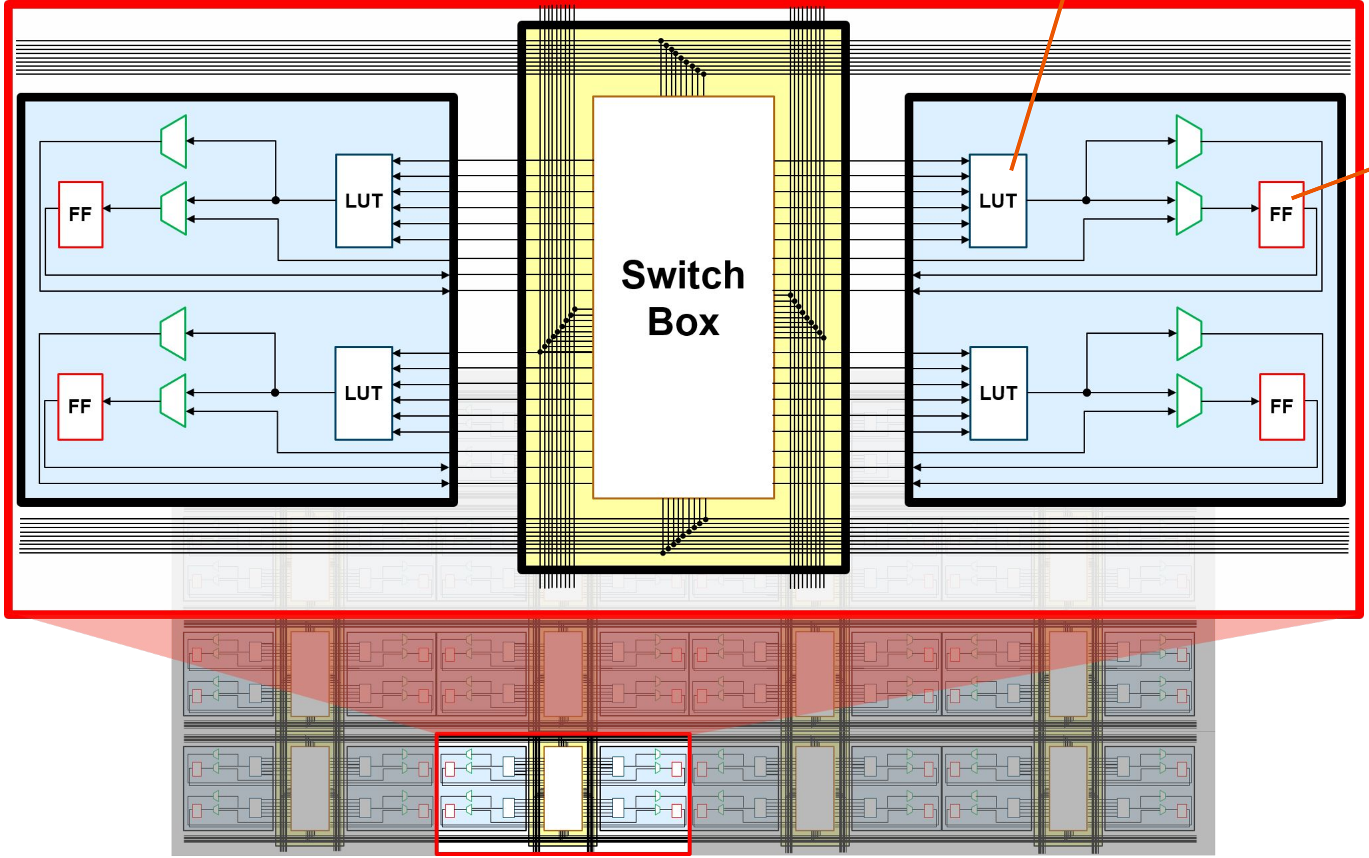
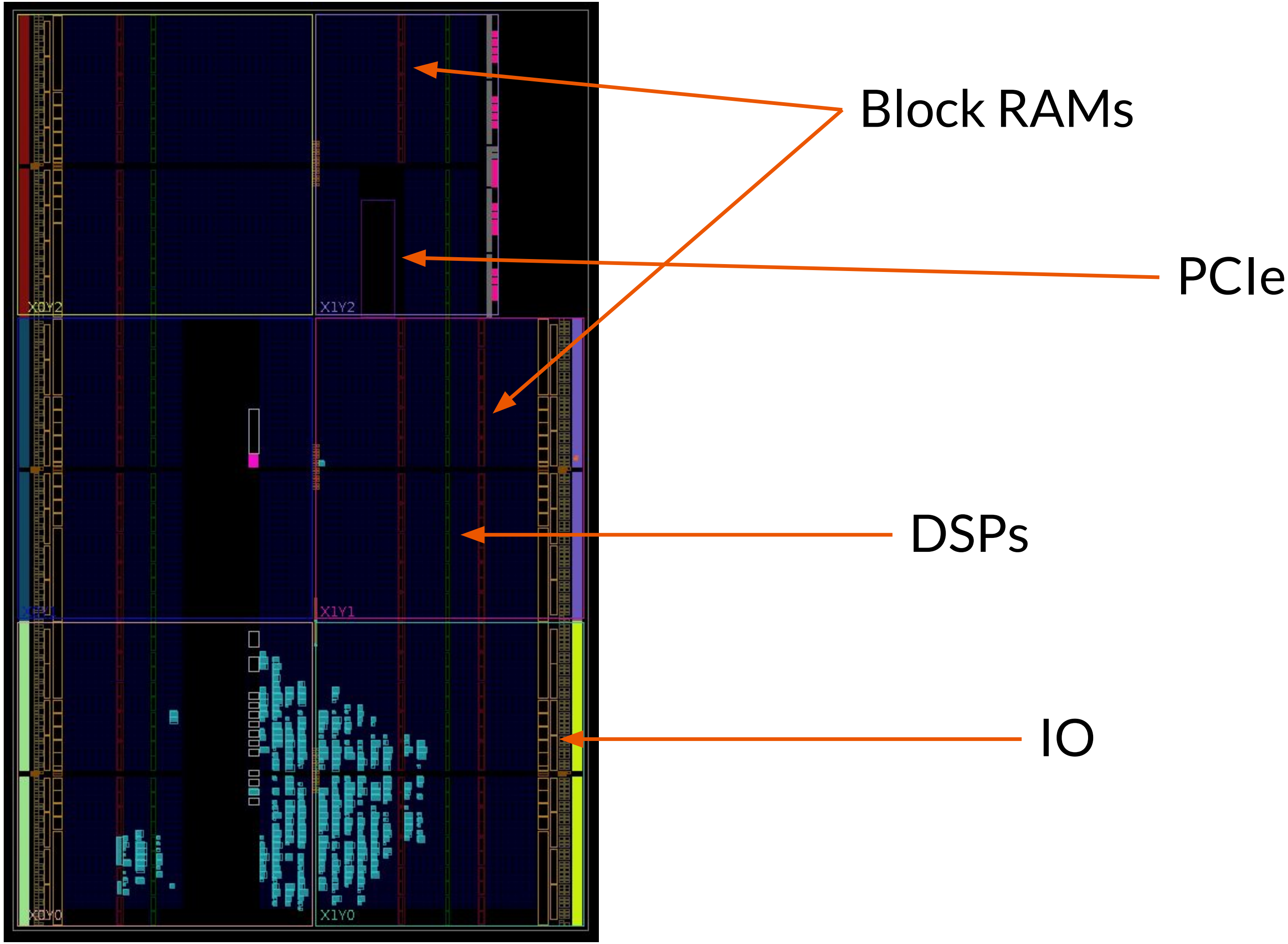
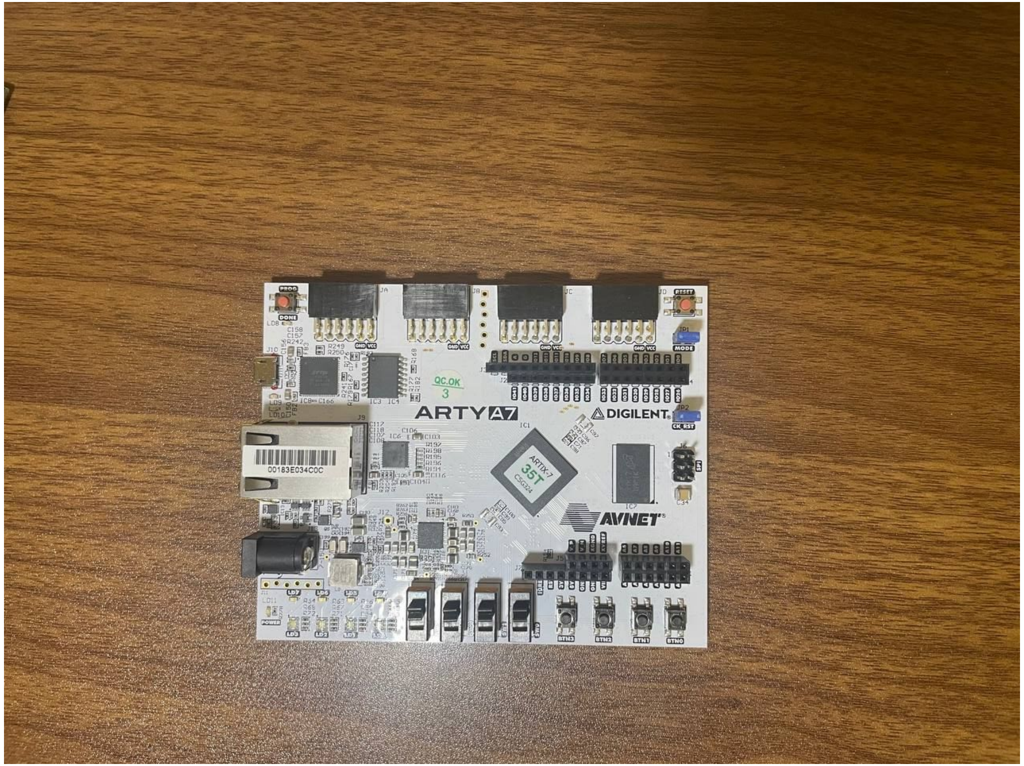
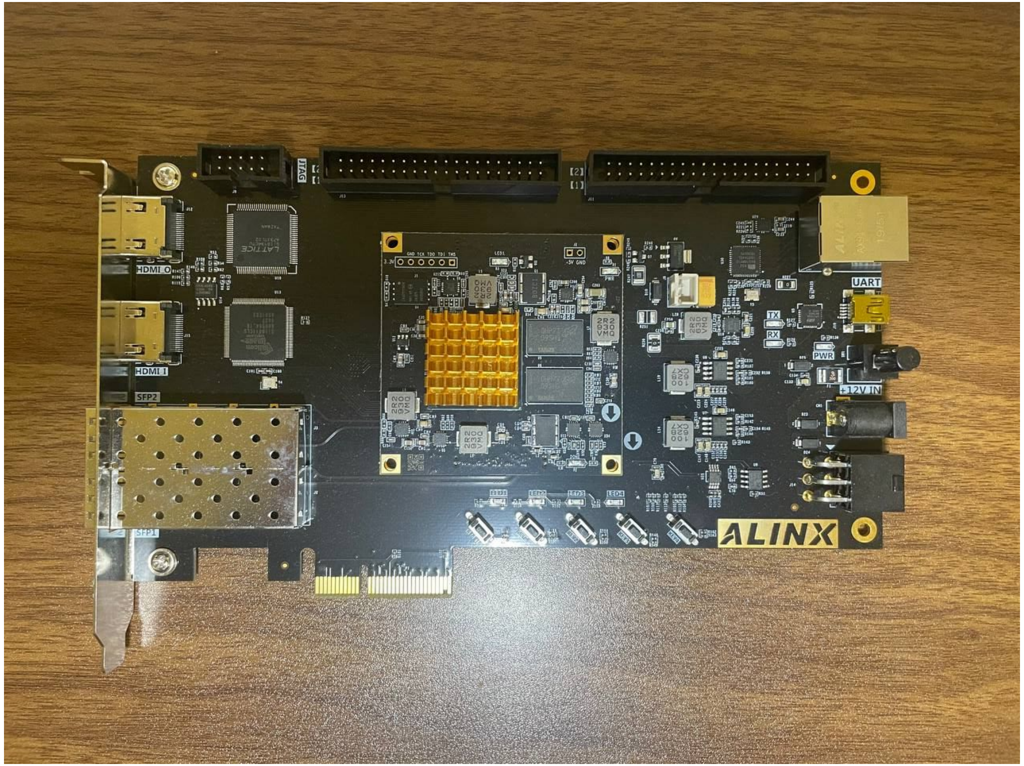


Figure and table from https://www.rapidwright.io/docs/FPGA_Architecture.html


X0Y2
X1Y2
X1Y1
X0Y0
X1Y0
Block RAMs
PCIe
DSPs
IO


ARTY A7
DIGILENT
AVNET
ARTIX-7
35T
QC.OK
3
00183E034C0C


JTAG
HDMI_O
HDMI_I
SFP2
SFP1
UART
TX
RX
PWR
+12V IN
ALINX

Sounds like a weird device.
How do you program it?

Hardware Description Languages

Verilog, SystemVerilog, VHDL, …
They are languages not only for FPGA, but also silicon.

Let's use Verilog

```
module example(clk, valid, a, b, o1, o2);
  input wire clk, valid;
  input wire [1:0] a, b;
  output reg [1:0] o1, o2;
  reg [1:0] tmp;
  always @(posedge clk) begin
    if (valid) begin
      o1 <= tmp;
      o2 <= a + b;
    end
    else begin
      o1 <= 0;
      o2 <= 0;
    end
  end
  always @(*) begin
    tmp = a - b;
  end
endmodule
```

- Module
- Always
  - With clock
  - Without clock
- Type
  - "wire": outside always
  - "reg": inside always
  - "logic": everything
- Assignments
  - Blocking "="
  - Nonblocking "<="
    - Nonblocking → DFF
- If-statements
  - Like C
  - If → MUX
- Parallelism

IPs

- Close-source libraries
  - Like a configurable module
  - May use on-chip non-programmable resources
    - E.g., memory controller, block ram, PCIe

Synthesis

- A "synthesizer"
  - Like a software compiler
  - Verilog → a bunch of gates and wires
    - Mapping to the LUTs, FFs, and MUXs
    - A "bitstream" which configures the FPGA

Reconfiguration

- FPGA is reconfigurable
  - Program with different bitstream
  - Only one bitstream at a time
  - No context save/restore
    - Kill and relaunch?

I’m a software guy and I’m lazy.
Verilog is too complex for me.

High Level Synthesis

Something that translates C/C++/OpenCL/LLVMIR to Verilog.
So a software programmer can write hardware.

Cons: slow, resource consuming

Why do we want to use FPGA? Why not use GPU?

Why not use GPU?

GPU works well in some tasks but not all.
With FPGA, you can design very specialized hardware.
Also, power consumption matters.

If you need very specialized hardware, why not just build a silicon chip?

Why not build a chip?

People are not that rich.
Also, silicon fabrication is not fast.

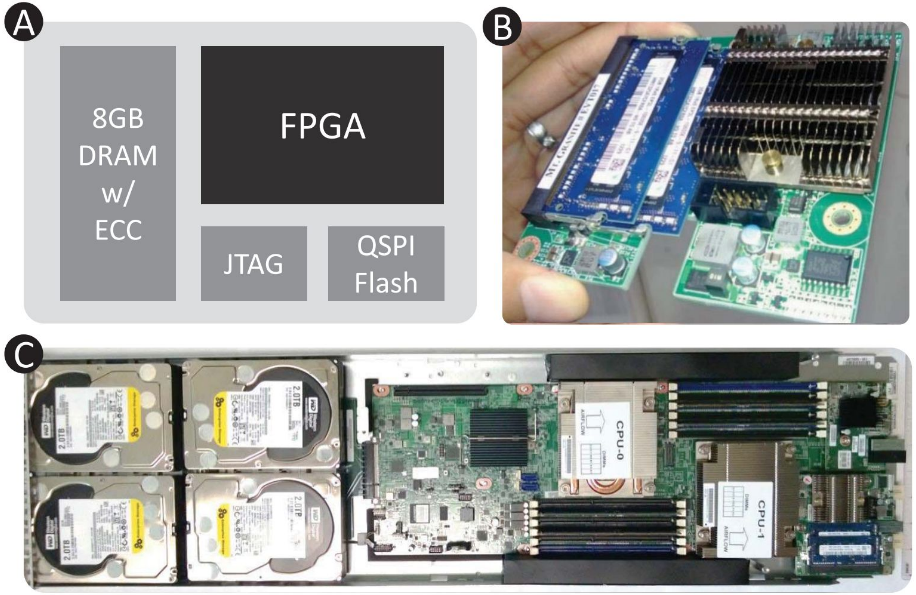
Research on FPGA?

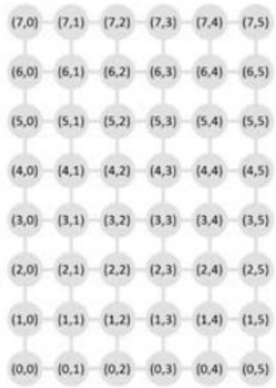
Catapult

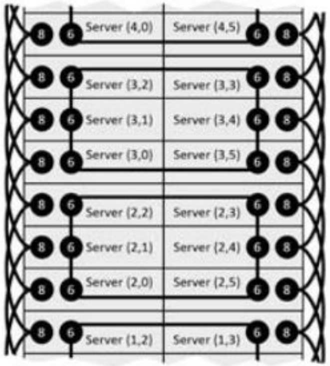
How does Microsoft use FPGA to make money?

The Platform

➔ M$ is rich, they build their own boards for the FPGA



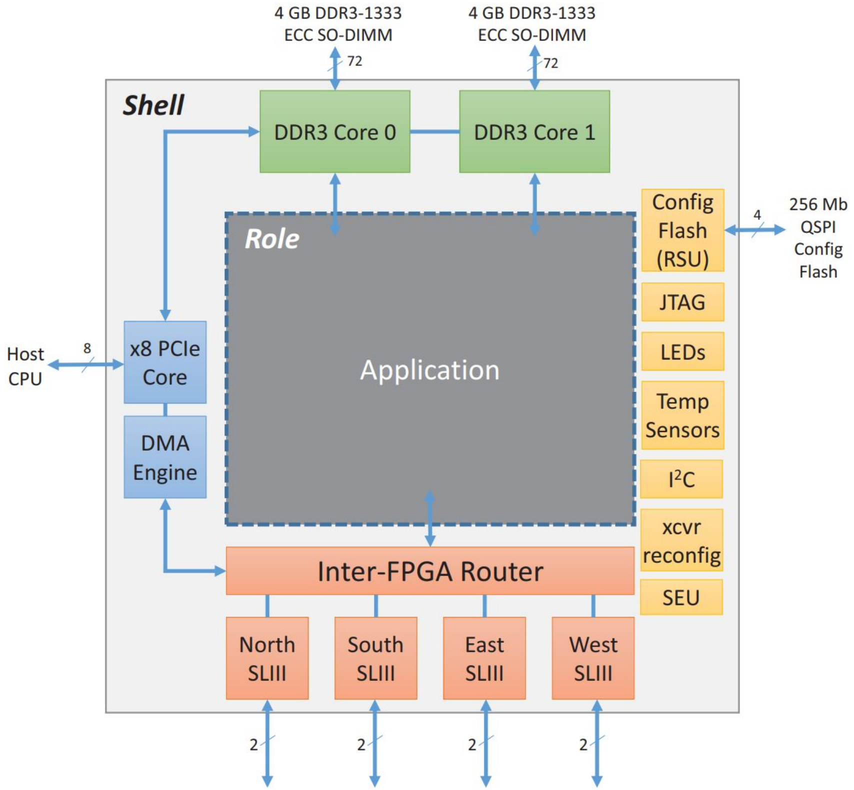




[1] A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services

The Shell and the Role

- Shell
  - Standard logic
  - same for all applications
- Role (~23% of resource)
  - Application logic
  - Partial reconfiguration



[1] A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services

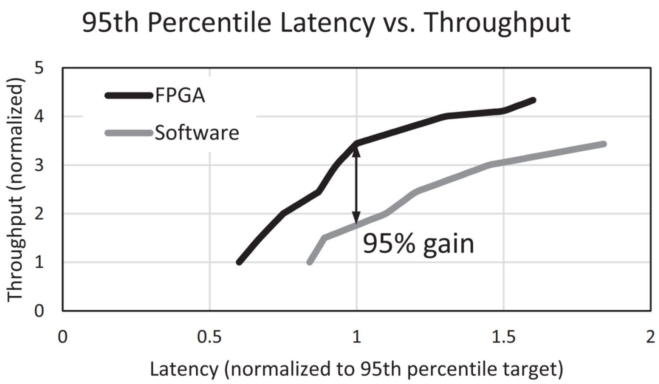
Software Interface

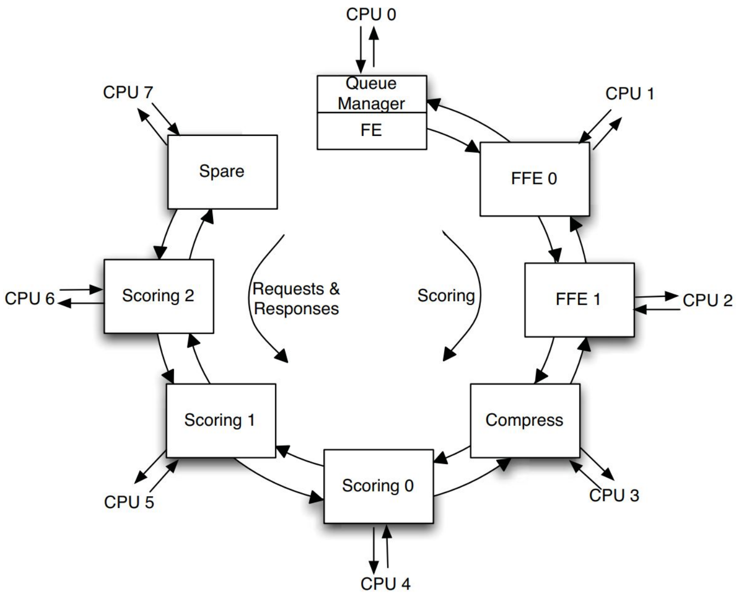
- User space driver
- Per-thread buffer
  - FPGA selects the buffer to read
- Data-center software support
  - Job scheduling
  - Fault detection and recovery

[1] A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services

Accelerating Bing's Ranking Algorithm

- ➔ 7-FPGA pipeline for one task
  - ◆ 1 for backup, used when error occurs
- ➔ 95% speedup



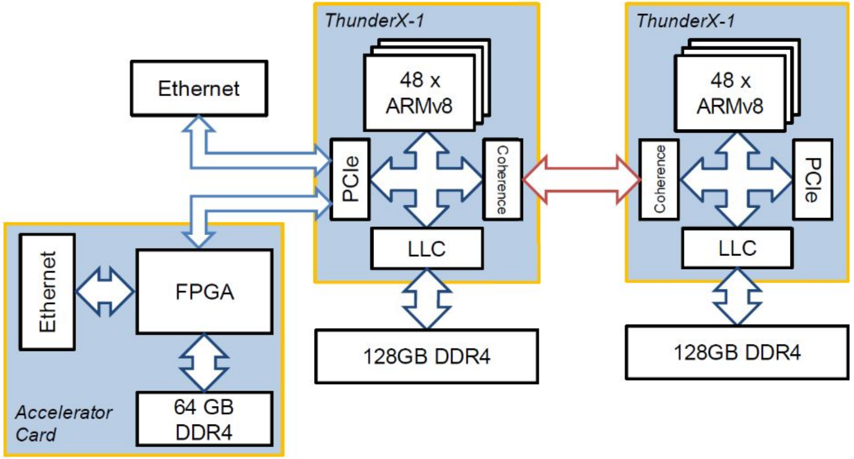


Enzian

The future of CPU-FPGA interconnection

A Traditional FPGA...

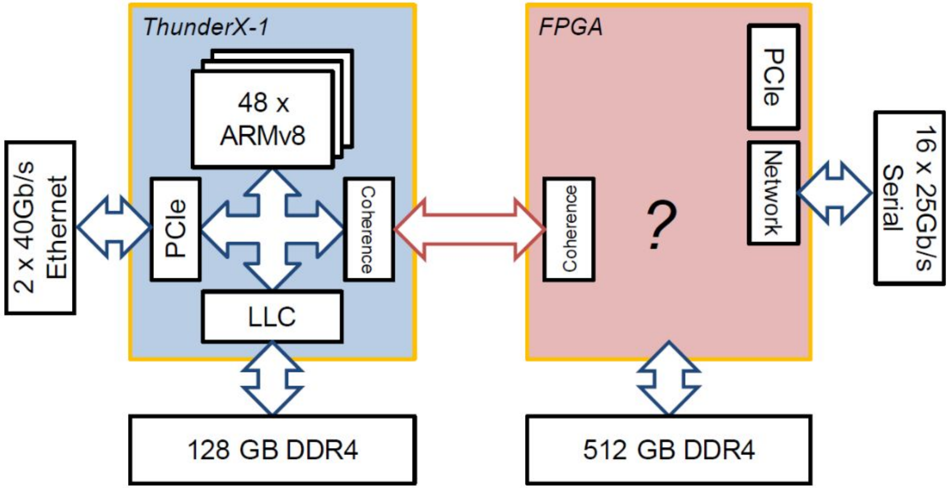
- ➔ Access CPU-side memory via PCIe
  - ◆ Relatively high latency for small transactions (while accessing memory)



[1] http://enzian.systems

Cache Coherent FPGA!

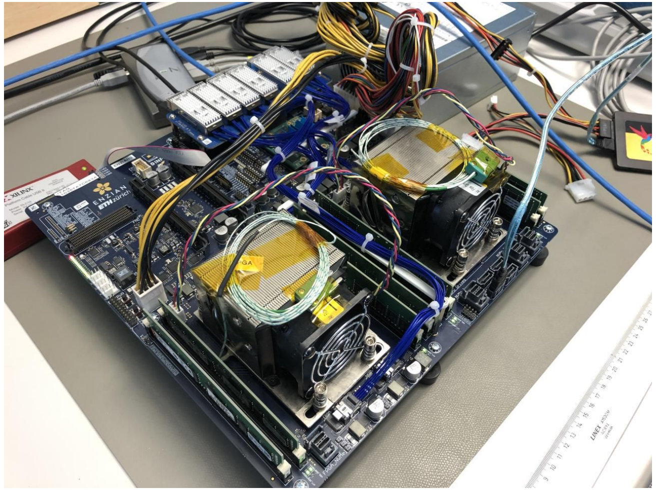
- Implementing cache coherent protocols on the FPGA
  - Like a NUMA node; cacheline-grained access with low latency



[1] http://enzian.systems

Implementation of Enzian

➔ 1 postdoc, a bunch of PhD students, 6 years



[1] http://enzian.systems

AmorphOS

Now we have an OS for the FPGA...

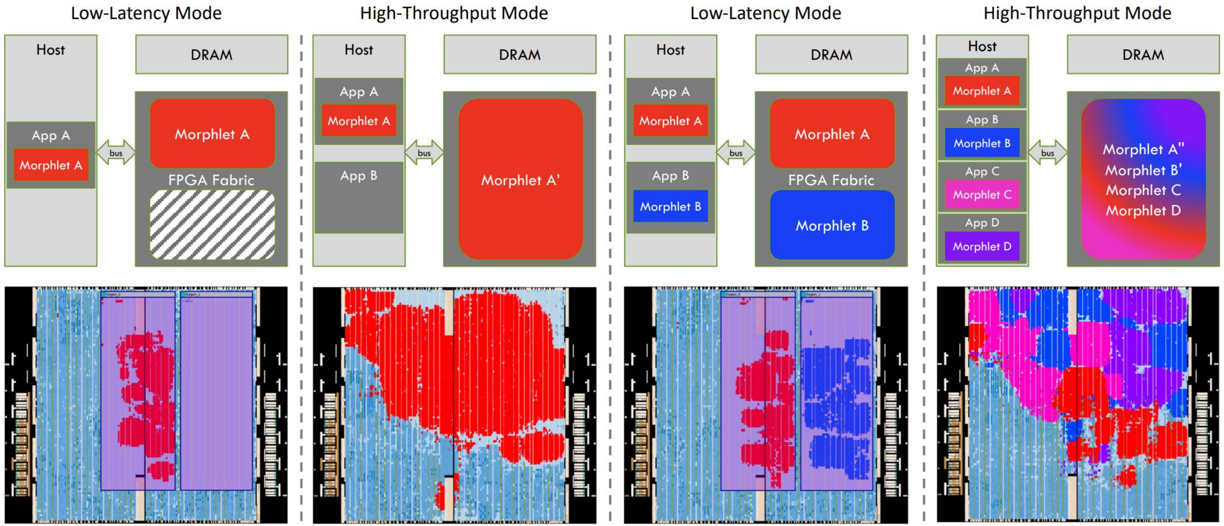
The Problem

- User may not fully utilize the FPGA
  - Low resource utilization in data centers
- Different sized accelerator at different time
  - E.g., at midnight, you may want to use a small one
- Different number of instances at different time
  - E.g., Tencent's server may be more crowded during the summer break

[1] Sharing, Protection and Compatibility for Reconfigurable Fabric with AmorphOS

The solution

➔ Use multiple partial-reconfigurable regions on the FPGA



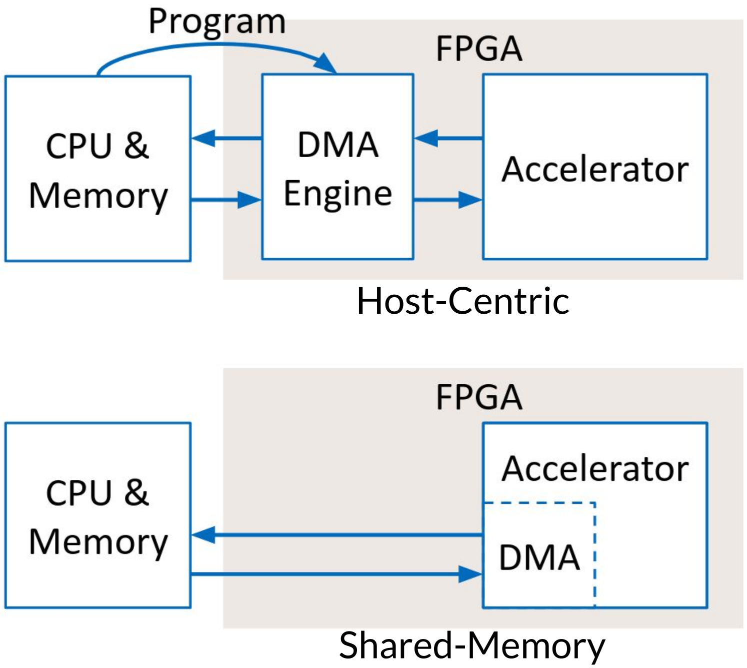
[1] Sharing, Protection and Compatibility for Reconfigurable Fabric with AmorphOS

Optimus

Exposing multiple accelerators to different VMs

Host-Centric vs. Shared-Memory

- ➔ Whether there's a DMA engine
- ➔ Catapult/AmorphOS are for host-centric FPGAs
- ➔ **How to virtualize an FPGA if it's shared-memory?**



[1] A Hypervisor for Shared-Memory FPGA Platforms

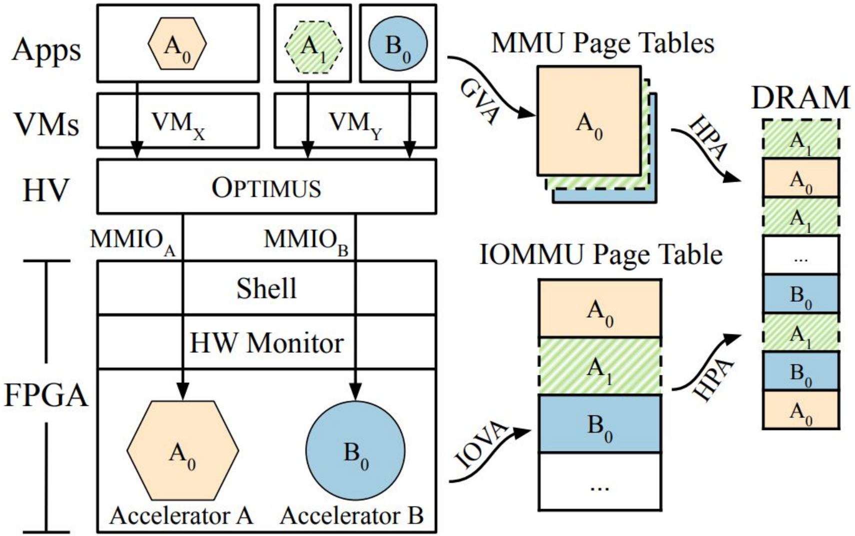
The Problem

- → No SR-IOV support
- → VMs share one IOMMU page table

[1] A Hypervisor for Shared-Memory FPGA Platforms

The Solution

➔ Divide the IOMMU page table into multiple slices



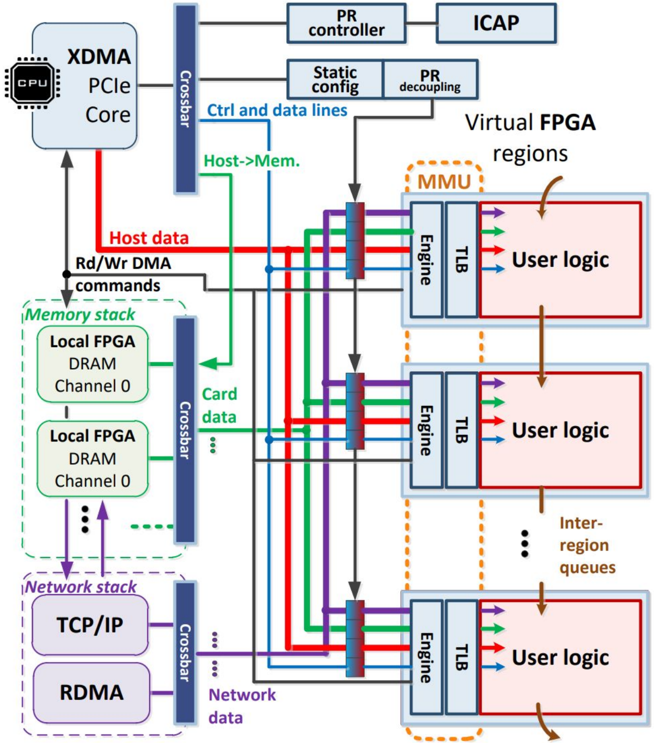
[1] A Hypervisor for Shared-Memory FPGA Platforms

Coyote

A more concrete implementation of FPGA OS

The Architecture

- ➔ Per-accelerator TLB
  - ◆ Software-controlled
  - ◆ No page table
- ➔ Software-managed memory allocation



[1] Do OS abstractions make sense on FPGAs?

Questions?