Cloud FPGA

EENG 428
ENAS 968

bit.ly/cloudfpga

Overview of Cloud FPGA Systems

Prof. Jakub Szefer
Dept. of Electrical Engineering
Yale University

Cloud FPGA Review

- On-demand access to FPGAs to deploy custom hardware accelerators
- No up-front costs (no purchasing hardware, licenses, servers, etc.)
- Easy to scale out, just rent more instances

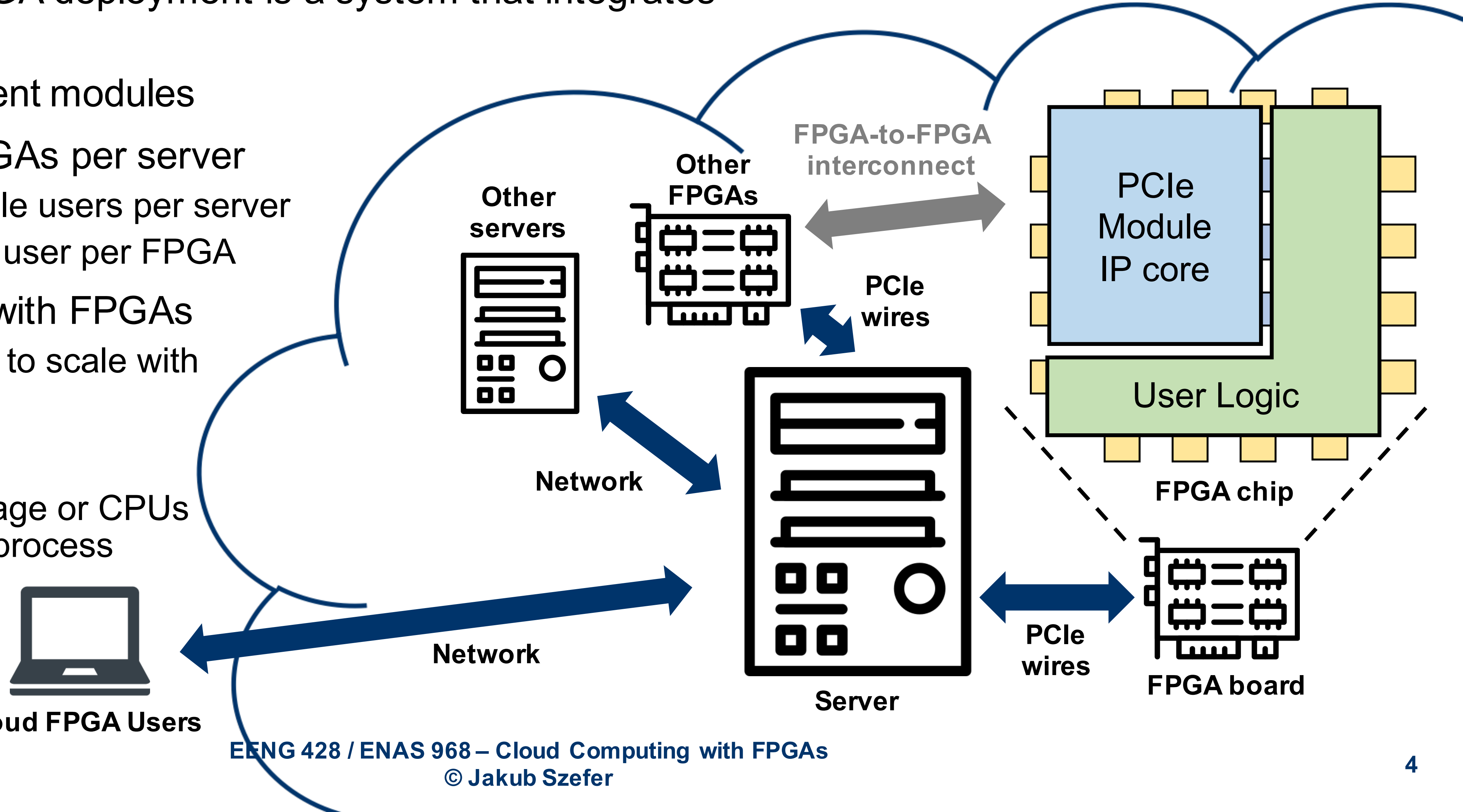
Challenges (and Opportunities):

- Need to write the hardware design from scratch, e.g. Verilog, then test, and deploy
  - Motivation for EENG 428 / ENAS 968
  - Also, recent move to High-Level Synthesis
  - Also, pre-made FPGA designs sold on marketplaces
- Need to understand FPGAs and whole system to get best results
  - More motivation for EENG 428 / ENAS 968
- Less control over data, and resulting security issues due to cloud provider or other users
  - Even more motivation for EENG 428 / ENAS 968

End-to-End Cloud FPGA System

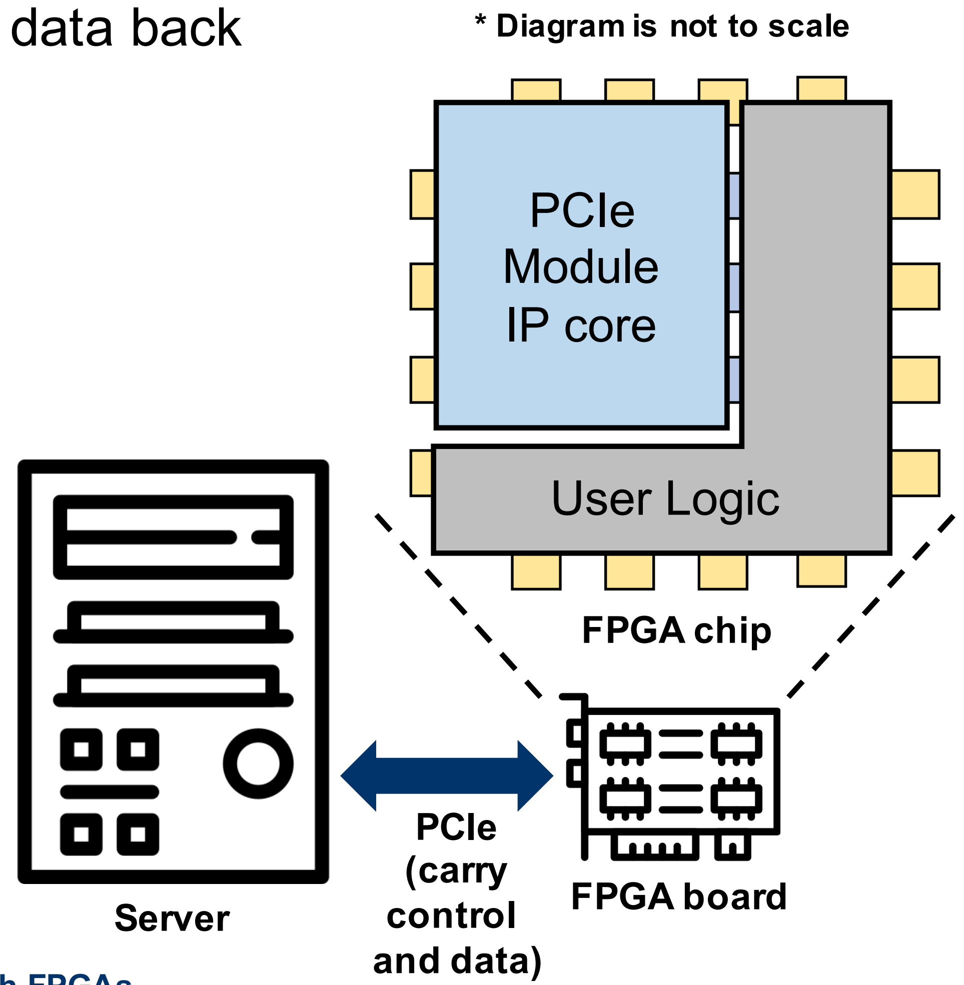
A typical Cloud FPGA deployment is a system that integrates many components:

- FPGA with different modules
- One or more FPGAs per server
  - Support multiple users per server
  - Currently, one user per FPGA
- Multiple servers with FPGAs
  - Support ability to scale with more servers
- Other servers
  - Leverage storage or CPUs to manage or process FPGA inputs or outputs



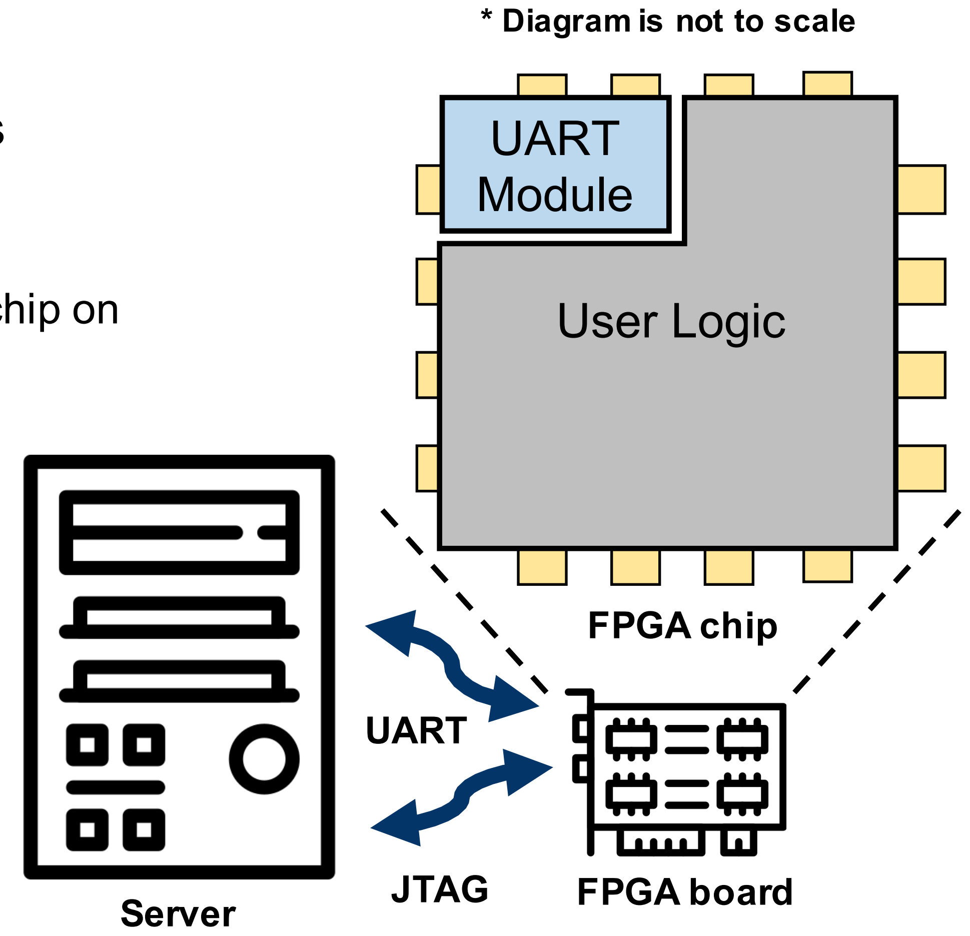
Interfacing FPGAs with Servers using PCIe

- Part of the FPGA's logic is used for communication with the outside world
- Input and output (I/O) are need to send data to FPGA or get data back from FPGA after processing is done
- Many interface standards exist:
  - Serial port, USB
  - Ethernet, QSFP, QSFP+, SATA
  - PCI Express (PCIe)
  - Custom
- In commercial Cloud FPGAs, PCIe is the de-facto standard for communication between the host server and the FPGA board
  - Send or receive data byte by byte
  - Send or receive data using Direct Memory Access (DMA)
- Newest protocol: Compute Express Link (CXL) using PCIe



Interfacing FPGAs with Servers using Serial Port & JTAG

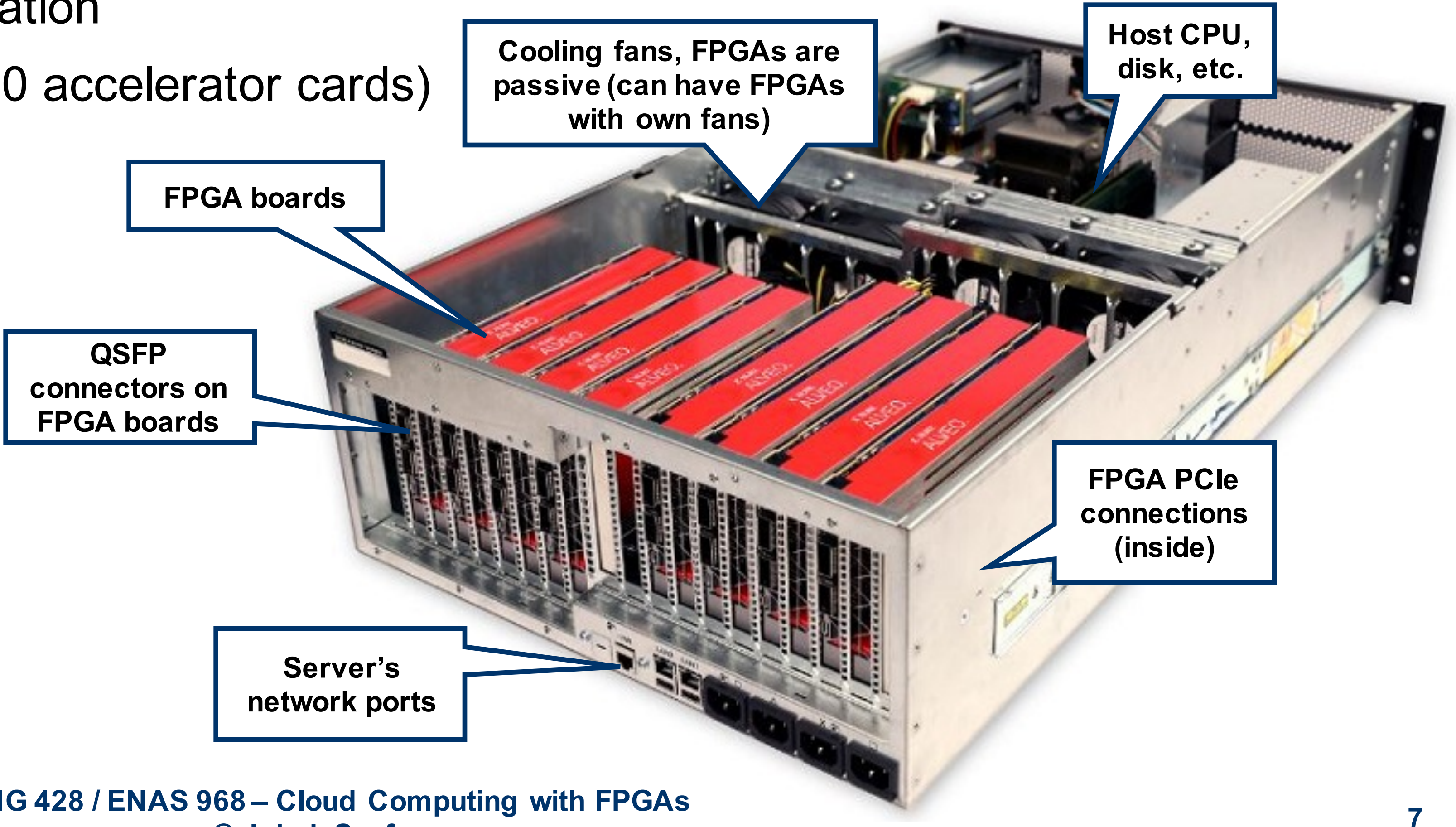
- In research or prototype deployments the communication between the FPGA and server can be done via UART, i.e. serial port
  - Much slower than PCIe
  - Simpler to setup, does not require special PCIe device drivers
  - Most often UART-over-USB is used, serial port data is accessed via `/dev/ttyUSB` on Linux for example,
    - Computer encodes UART packets into USB, and there is extra chip on the FPGA board that decodes UART from USB before sending data to/from the FPGA chip itself.
- Also need to configure the FPGA, which is done using JTAG port
  - Most often JTAG-over-USB is used
    - Computer encodes JTAG packets into USB, and there is extra chip on the FPGA board that decodes JTAG from USB before sending commands to the FPGA chip itself.



Example AMD Server with Xilinx FPGAs

Not actual server used in Cloud FPGAs, but has all the key hardware:

- Main CPU for controlling the server (32-core EPYC 7551 CPU)
- Network interfaces for communication
- 8 FPGA boards (Xilinx Alveo U250 accelerator cards)

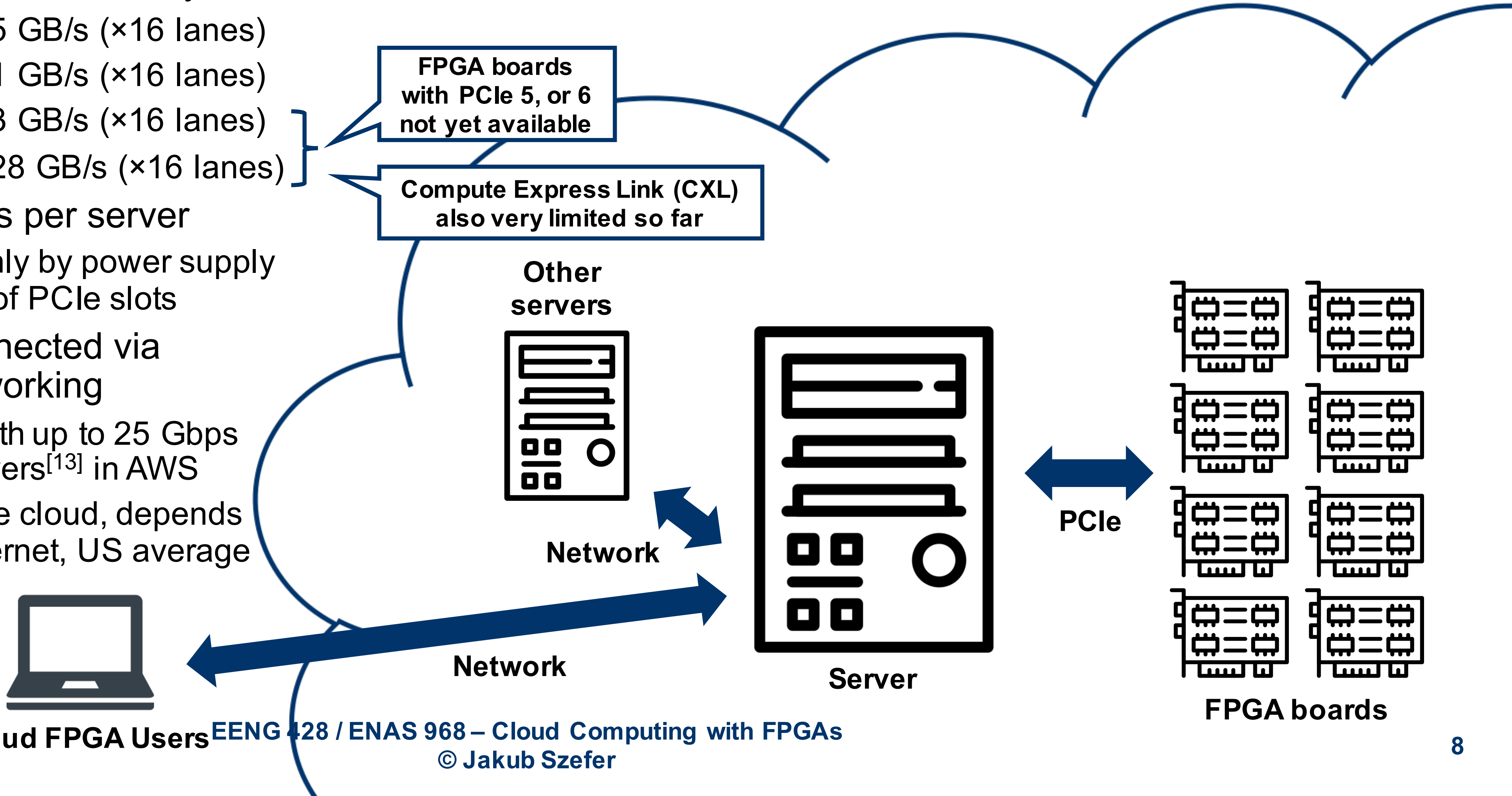

Cooling fans, FPGAs are passive (can have FPGAs with own fans)
Host CPU, disk, etc.
FPGA boards
QSFP connectors on FPGA boards
FPGA PCIe connections (inside)
Server's network ports

Cloud FPGA Servers – PCIe and Server Networking

- FPGA board and networking connections today in Cloud FPGAs
  - Today typically connected by PCIe to the server
    - PCIe 3.0: ~15 GB/s (×16 lanes)
    - PCIe 4.0: ~31 GB/s (×16 lanes)
    - PCIe 5.0: ~63 GB/s (×16 lanes)
    - PCIe 6.0: ~128 GB/s (×16 lanes)
  - Usually 8 FPGAs per server
    - But limited only by power supply and number of PCIe slots
  - Servers are connected via high-speed networking
    - E.g. bandwidth up to 25 Gbps between servers[13] in AWS
    - Outside of the cloud, depends on user's internet, US average is 64 Mbps[14]

FPGA boards with PCIe 5, or 6 not yet available

Compute Express Link (CXL) also very limited so far

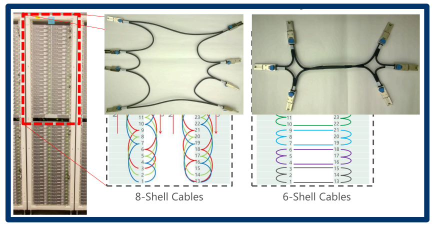


Cloud FPGA Servers – FPGA Interconnections

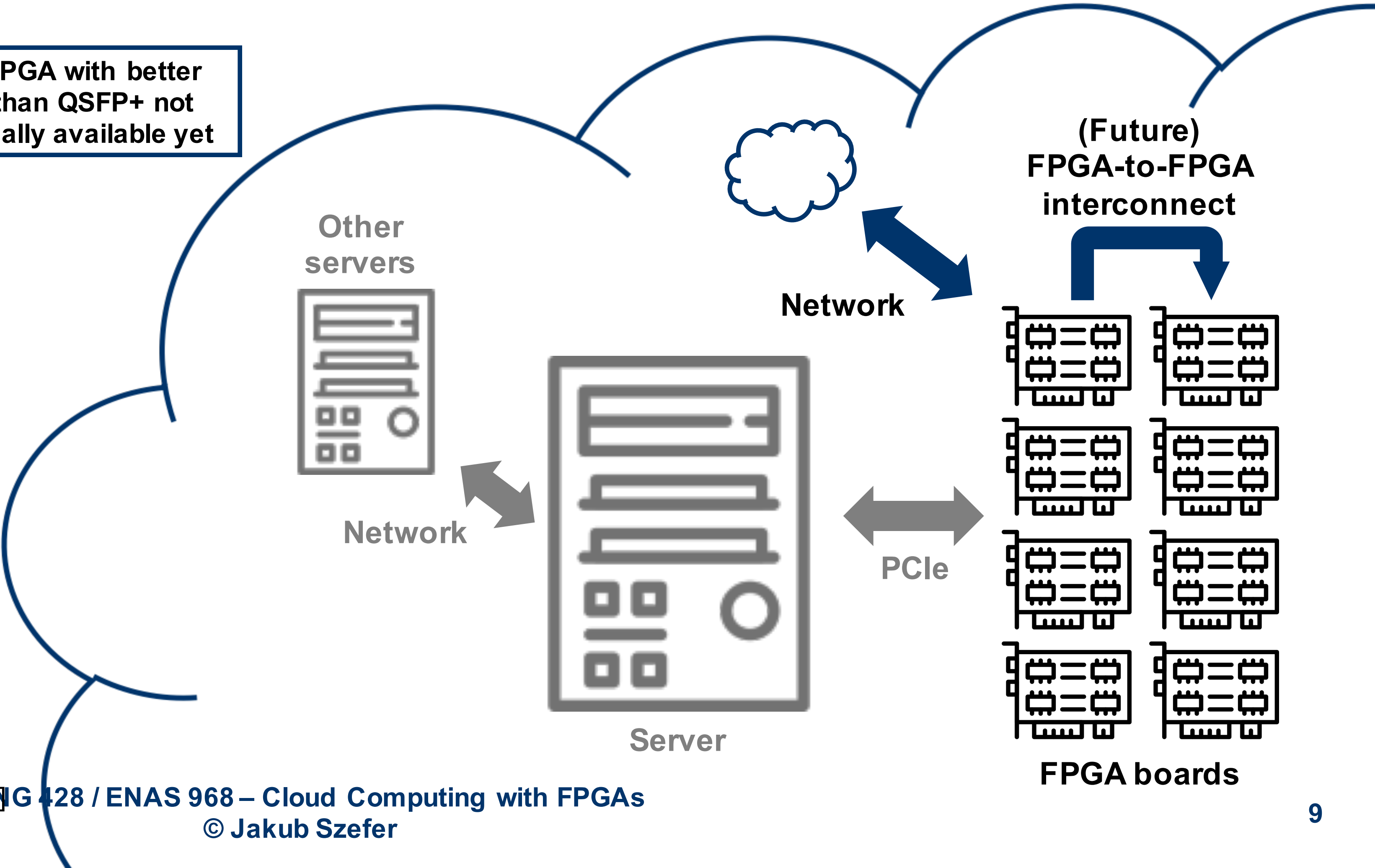
- Future options for interconnecting Cloud FPGAs
  - Many FPGA boards have QSFP connectors
    - QSFP: 4 Gbit/s
    - QSFP+: 40 Gbit/s
    - QSFP14: 50 Gbit/s
    - QSFP28: 100 Gbit/s

FPGA with better than QSFP+ not really available yet

  - Many plan to add FPGA-to-FPGA connections using direct links, and likely the QSFP connectors
    - E.g. work by Microsoft[5]



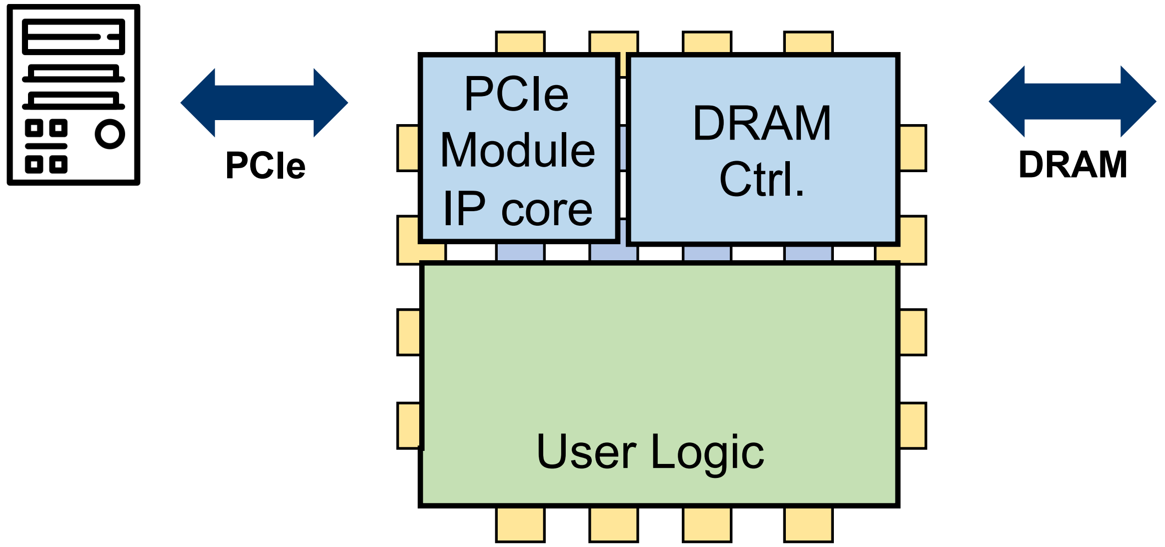
  - Could use Ethernet or QSFP to connect to internet directly



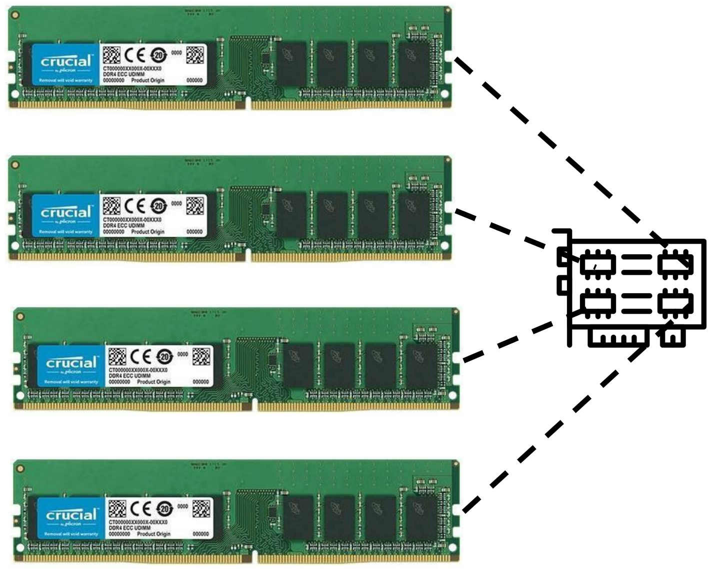
Microsoft image from [5]

Cloud FPGA Servers – FPGA DRAM

- FPGA boards used in Cloud FPGAs typically come with DRAM chips installed on the boards itself, for direct use by the FPGA
  - Today usually DDR4: about 18 GB/s
  - All memories work in parallel: about 150 GB/s
  - Typically 4 x 16GB = 64GB of memory
- DRAM controller needs to be instantiated on the FPGA
  - In addition to other controllers, such as PCIe



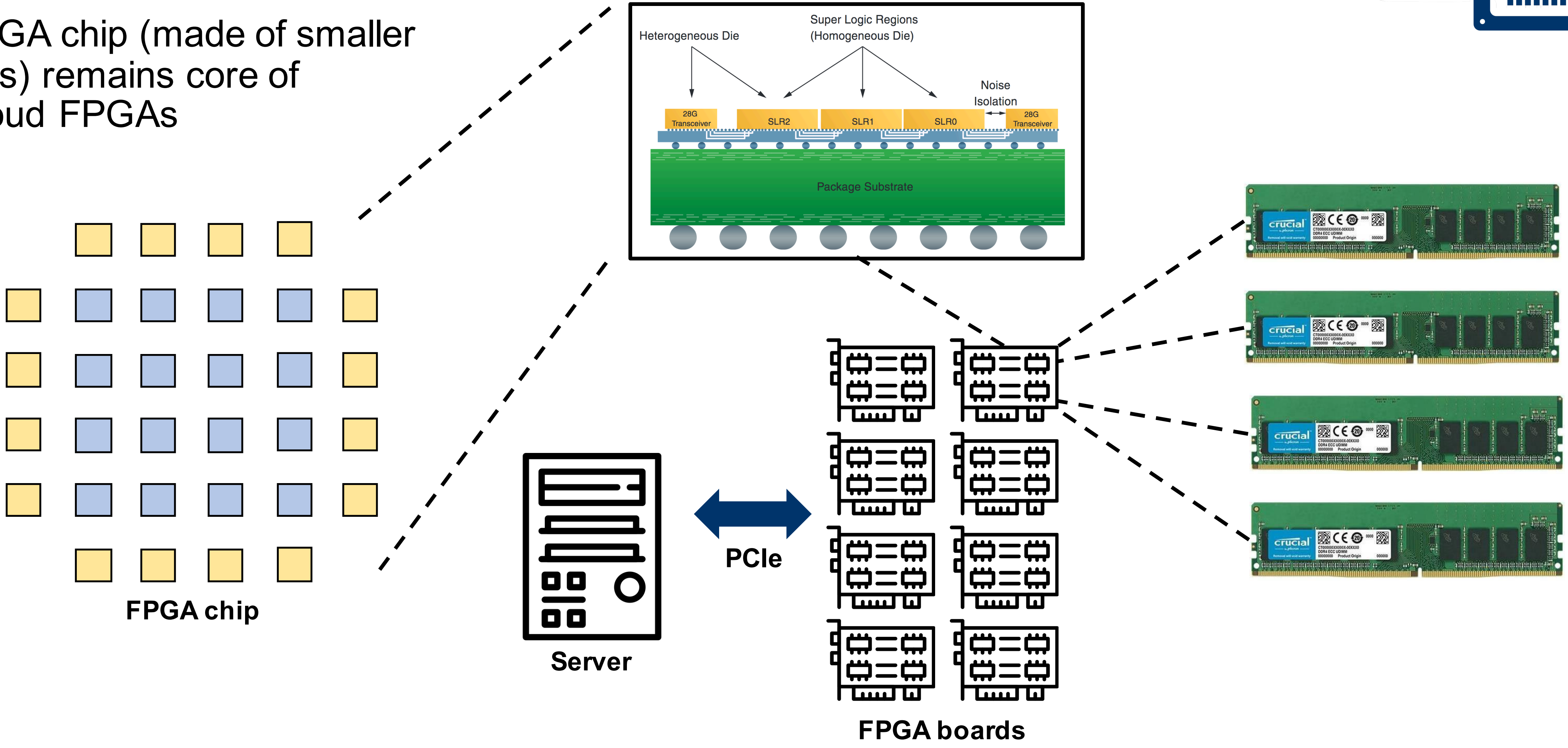
FPGA chip



DRAM images from
https://www.newegg.com/p/0RN-003K-00071

Summary of Typical Cloud FPGA Hardware

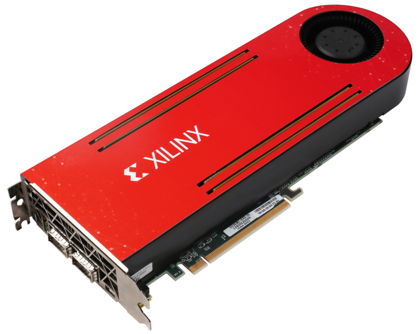
- FPGA chip (made of smaller dies) remains core of Cloud FPGAs



Example FPGA Accelerator Cards

FPGAs used in Cloud FPGA Infrastructures

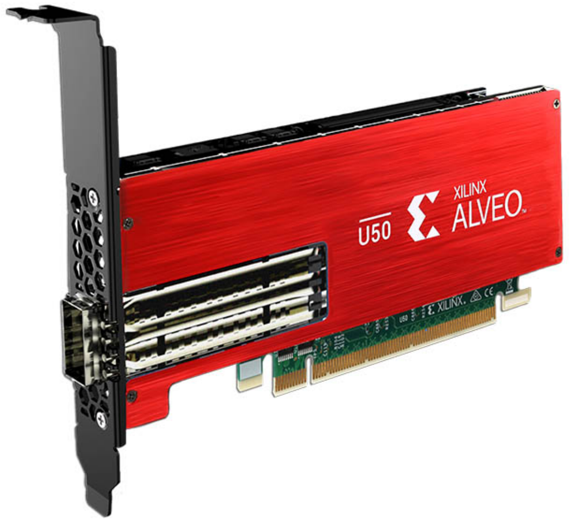
- Xilinx Virtex UltraScale+ FPGA VCU1525
  - Very similar to card used in **Amazon F1**
  - 2,500,000 logic cells
  - Thermal Design Power (TDP) of 225W
  - Up to PCIe 4.0 and DDR4 and QSFP networking
- Xilinx Alveo U200/U250/U280 Accelerator Cards
  - Likely cards for **Amazon F1 SDAccel**
  - 800,000 to 1,000,000 LUTs
  - Thermal Design Power (TDP) of 225W
  - Up to PCIe 4.0 and DDR4 and QSFP networking
- Catapult FPGA Accelerator Card (Microsoft + Intel FPGAs)
  - Altera Stratix V GS D5
  - 172,000 ALMs
  - PCIe 3.0 and DDR3



Images and information from [6] and [7] and [8]

Other Recent FPGA Cards

- Xilinx Alveo U50 Accelerator Card (2019)
  - Uses Xilinx's 16nm UltraScale+ FPGA architecture
  - 800,000 LUTs
  - Thermal Design Power (TDP) of 75W
  - PCIe 4.0 and HBM2
  - QSFP networking



- Intel D5005 Programmable Acceleration Card (2019)
  - Uses Intel's 14nm Stratix 10 SX FPGA architecture
  - 2,800,000 logic elements
  - Thermal Design Power (TDP) of 215W
  - PCIe 3.0 and DDR4
  - QSFP networking

Images and information from [2] and [3]

Other Recent FPGA Cards

- SmartSSD Computational Storage Drive (approx. 2021)
  - 3.85 TB SSD NVMe drive
  - Uses Xilinx Kintex™ Ultrascale+ KU15P FPGA
  - 300,000 LUTs
  - Thermal Design Power (TDP) of 25W (FPGA)
  - PCIe 3.0 and 4GB DRAM on FPGA

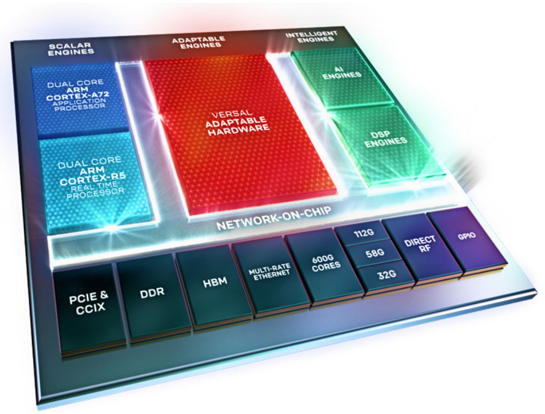
- Achronix VectorPath Accelerator Card (approx. 2021)
  - Uses Achronix Speedster7t FPGAs
  - 380,000 to 2,600,000 logic elements depending on part number
  - Thermal Design Power (TDP) of not specified
  - PCIe 4, may be updated to PCIe5, and DDR5

Images and information from [19] and [20]

New FPGA Architectures and Other Advances

New FPGA Architectures

- **Xilinx Versal architecture**
  - 7nm technology
  - 800,000 LUTs + AI engines + Arm cores
  - PCIe 4.0, DDR4, HBM, and memory coherent with CPUs via CCIX
  - Heterogeneous system, with Arm cores, real-time processors, AI and DSP engines, DDR and HBM, etc.



- **Intel Agilex architecture**
  - 10nm technology
  - 900,000 ALMs + Arm cores
  - PCIe 4.0, DDR4, HBM, and memory coherency with Xeon CPUs
  - 3D heterogeneous system-in-package (SiP) using Embedded Multi-Die Interconnect Bridge (EMIB), and chiplet-based architecture

Intel® Agilex™ FPGA
10nm
EMIB
DDR

- **Future trend from pure FPGAs to heterogeneous systems built around FPGAs**

Images and information from [4] and [5]

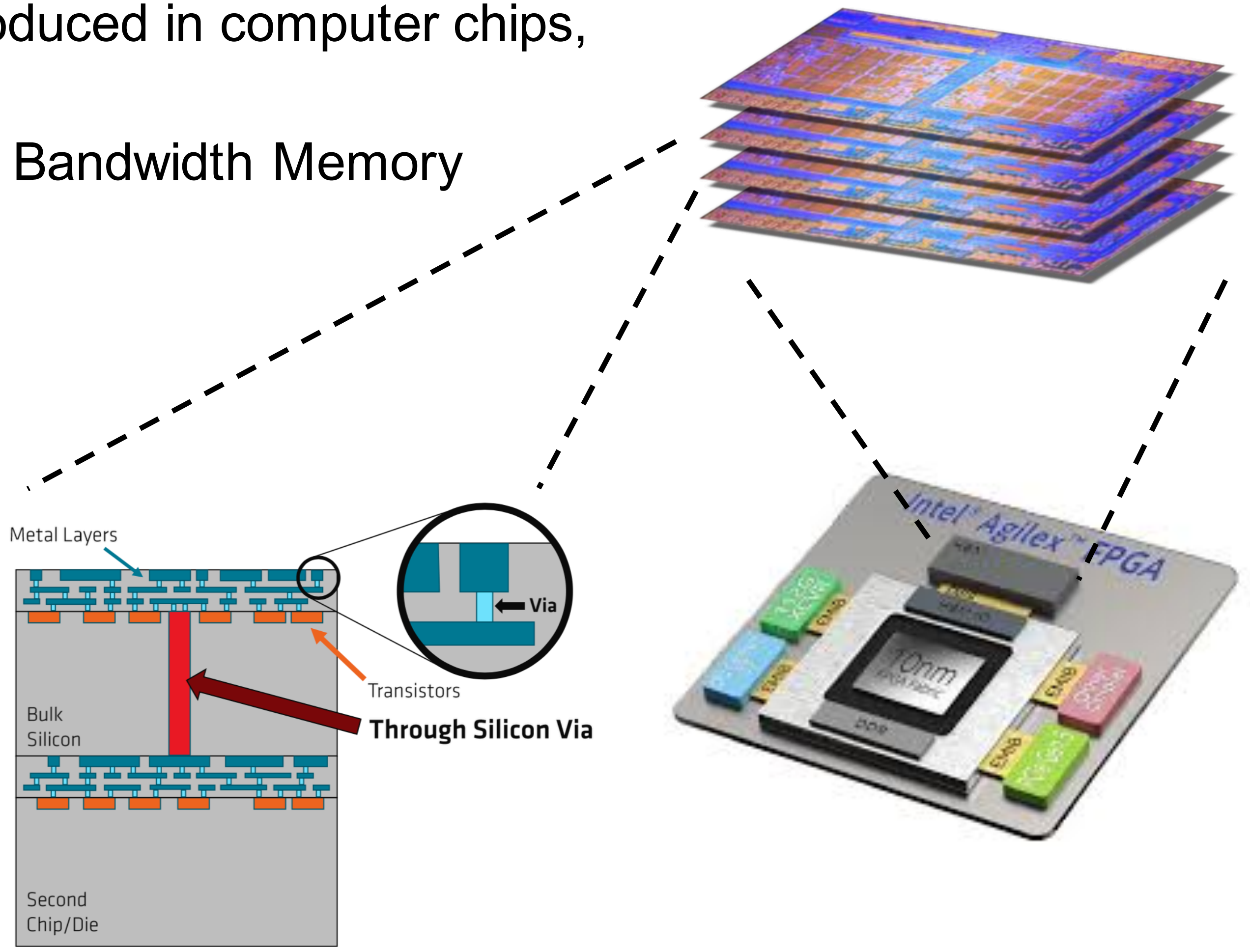
Vertically Integrating More Components onto a Chip

Die stacking images and information from Gabe Loh [16]

- Die stacking is recent technology introduced in computer chips, coming to FPGAs as well
- First major application in DRAM: High Bandwidth Memory (HBM) made of multiple DRAM dies

Die stacking:

- Multiple chips are stacked vertically on top of each other
- Increase density of chips
- Reduce interconnect distance
  - Faster connection
  - Less power
- Enabled by technologies such as Through Silicon Vias (TSVs)

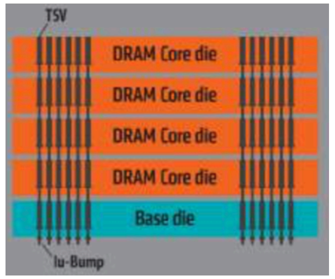


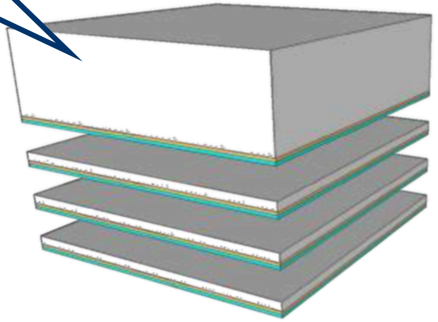
Agilex image from [5]

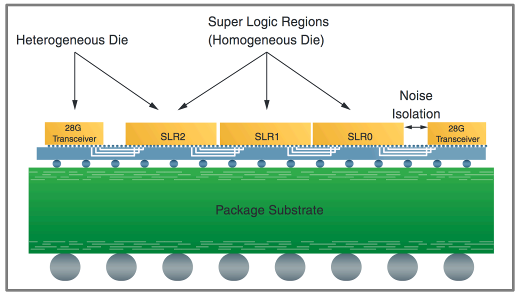
Die Stacking: 2.5D, 3D, and Beyond

Die stacking images and information from Gabe Loh [16]

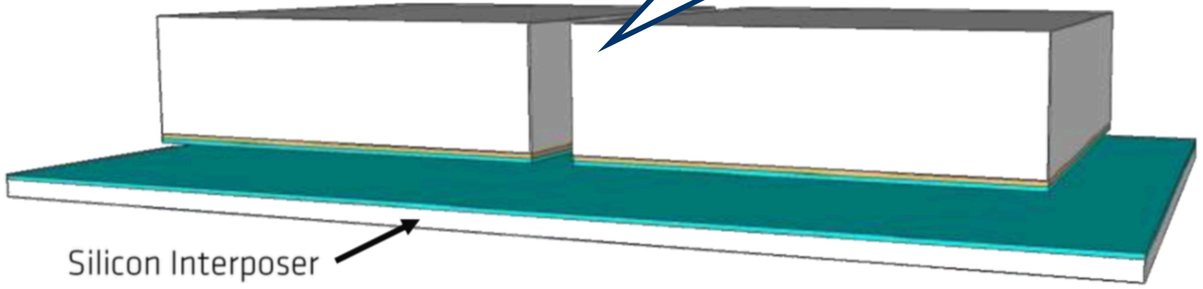
Currently used in HBM DRAM memories







2.5D stacking is already used in UltraScale chips from Xilinx



3D Stacking

- Can stack many layers
- Each layer is (about) the same size
- Cooling is difficult
- Limited number of pins
- Power delivery is difficult

2.5D Stacking
(sometimes "3D" is used for this configuration as well)

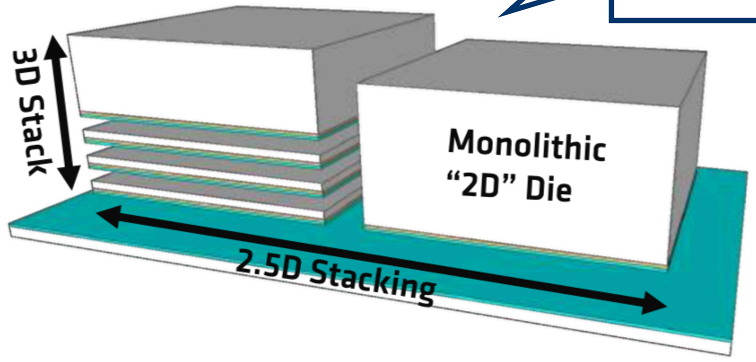
- Only one layer high
- Variable die sizes
- Cooling is easier
- More pins, but bigger area
- Power delivery is easier

Xilinx die diagram from [15]

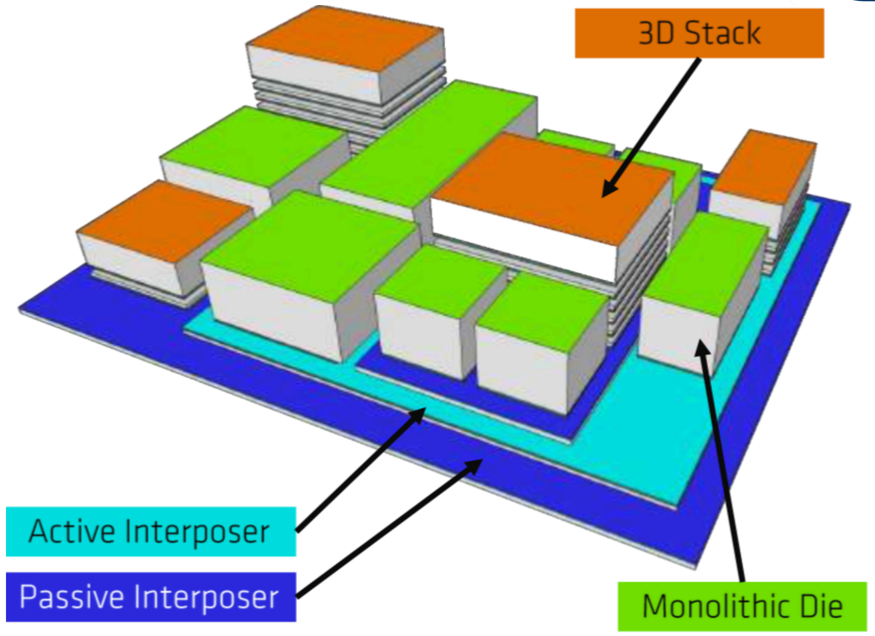
Die Stacking: 2.5D, 3D, and Beyond

Die stacking images and information
from Gabe Loh [16]

Used by Xilinx Versal and Intel Agilex (HBM is the 3D stacked part)



5.5D Stacking



Infinite Stacking Options

- Combine 2.5D with 3D stacking
- Some components are 3D stacked
- Some components are regular 2D
- All are stacked on the interposer in 2.5D manner

- Mix 2.5D, 3D, and 5.5D

FPGA “Free Lunch” with New Technologies

Future hardware allowing for bigger FPGAs, with more features

- Bigger FPGAs – 2.5D stacking allows smaller dies to be made into bigger FPGAs
- Other improvements:
  - Better PCIe – for faster connection to host server
  - FPGA-to-FPGA communication – for faster data transfer between FPGAs
  - Faster DDR – quicker access to large DRAM
  - HBM – faster access to memory
  - Other accelerators – use 3D integration and chiplets to add AI, GPU, and other accelerators

- Programming FPGA will not change, still need hardware design skills
- But need to understand whole system, not just FPGA chip itself

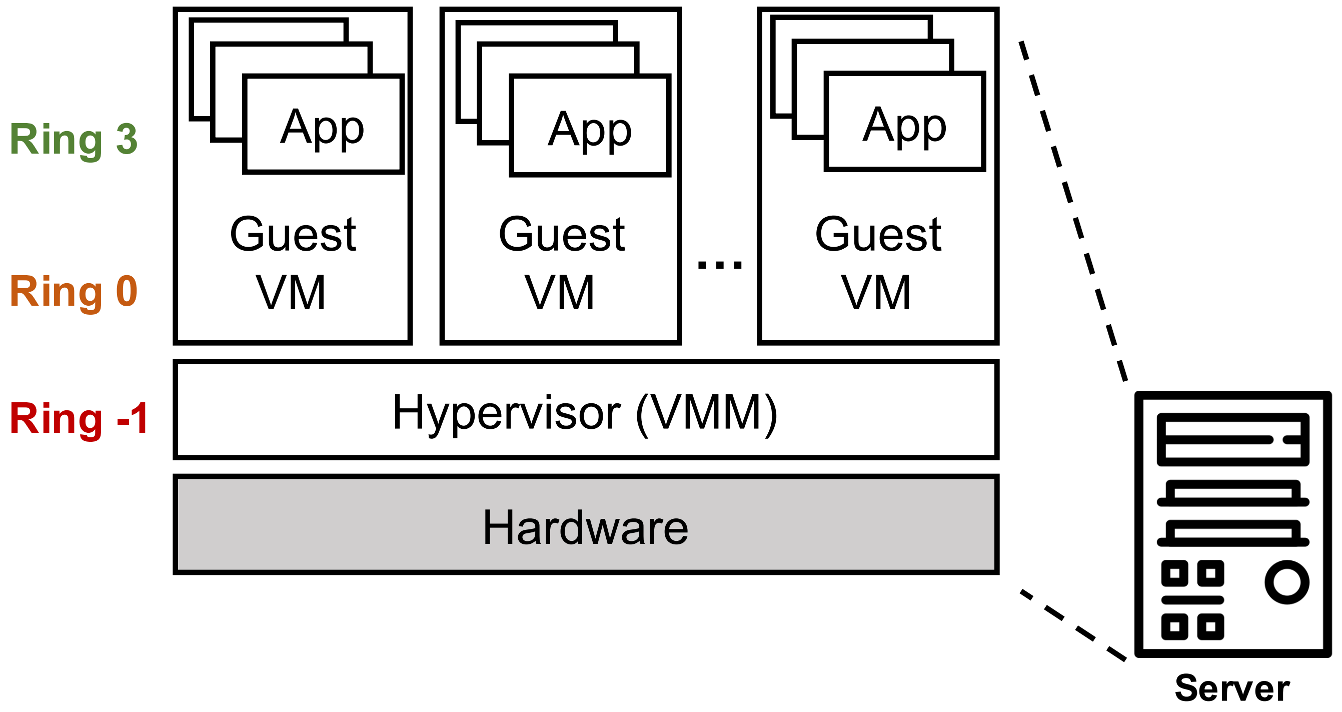
Questions?

Cloud FPGA Software

Prof. Jakub Szefer
Dept. of Electrical Engineering
Yale University

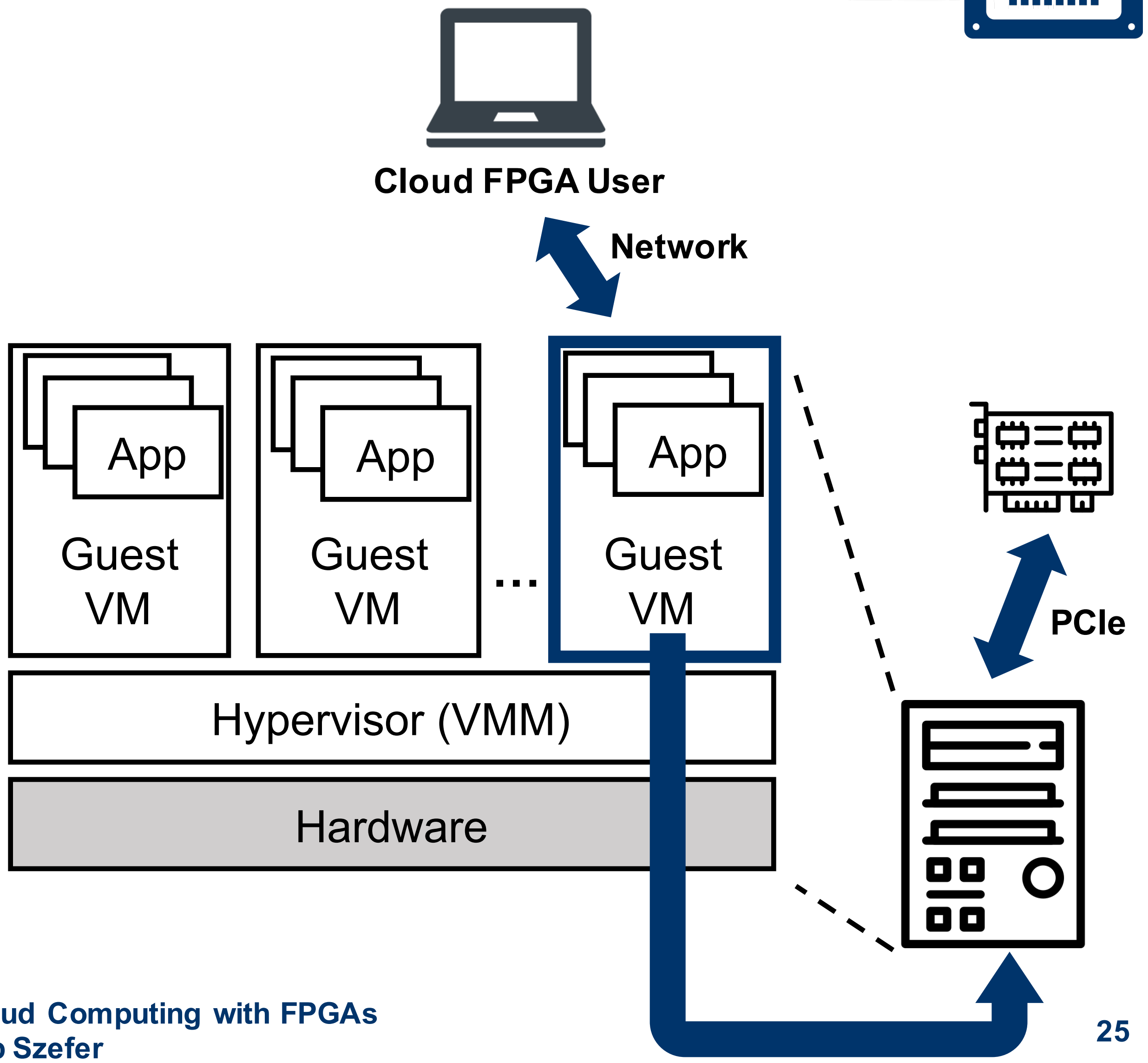
Cloud FPGA Server Software Stack

- Server software stack includes hypervisor, guest VMs, and user applications running on the VMs
- In IaaS setting:
  - Hypervisor is controlled by the cloud provider
  - Hypervisor controls all hardware
  - Users provide guest VM images (AMIs in Amazon's terminology)
  - Users provide applications and software that run in VMs
  - Cloud providers can give VM images with pre-installed tools, e.g., for FPGA development and programming



Leveraging Hypervisor for Server Management

- Cloud provider runs management software (e.g., open-source OpenStack or proprietary software) that allocates instances to users
- Each server runs hypervisor that actually controls the VMs on each server and manages resources assigned to that VM (instance)
- Hypervisor gives access to assigned hardware, such as FPGAs
- User's VMs use built-in libraries to communicate with the FPGAs via PCIe drivers



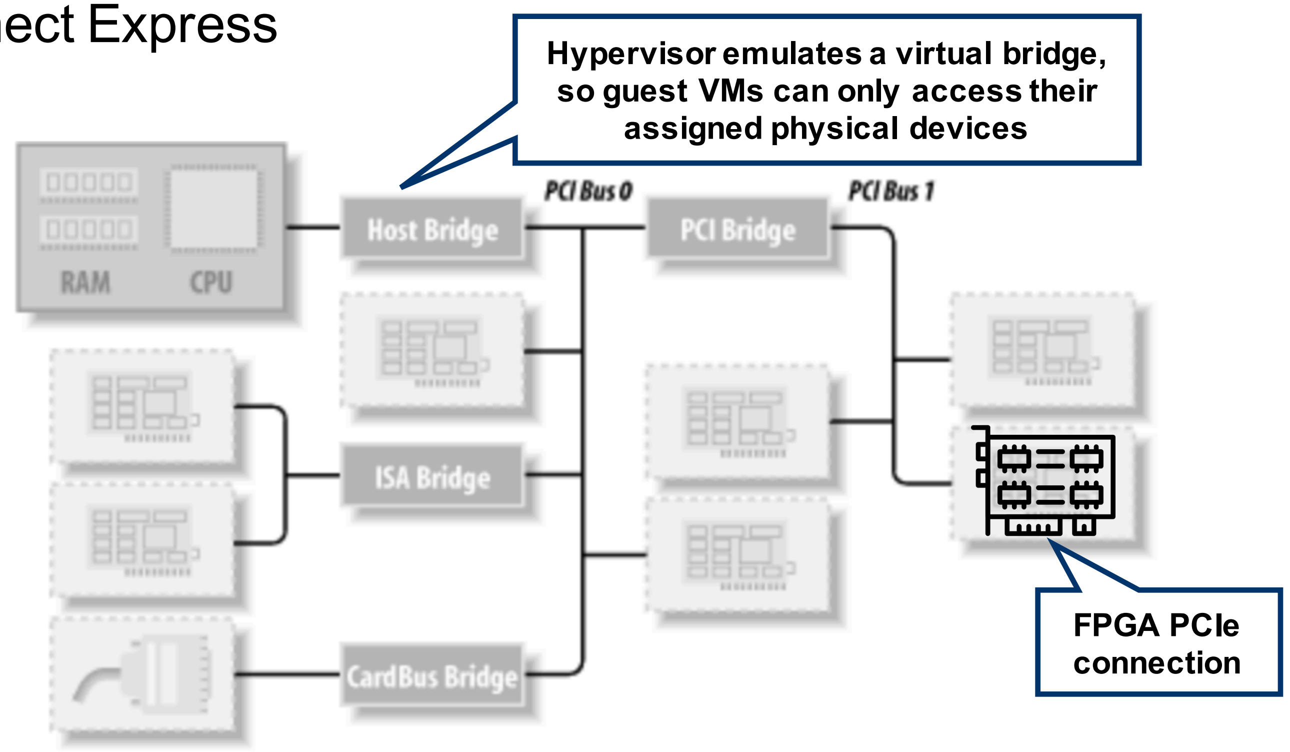
Guest VM Software and Libraries

- FPGA programming tools
  - Verilog, VHLD, SystemVerilog, etc.
  - High-Level Synthesis
  - E.g. Xilinx Vivado or Intel Quartus
- FPGA programming tools can be run: locally by user, on VM (without FPGA), on VM (with FPGA)

- PCIe drivers for FPGA board
- Tools (often command line) for checking status, programming, clean up, etc.
- Libraries for programming languages (e.g., C or Python) for sending and receiving data from the FPGA
  - (slow) Read or write data word by word – user initiates each read or write
  - (faster) Bulk copy of data – copy data word by word, but under control of a library function
  - (fastest) Direct Memory Access – copy data in large chunks between DRAM and FPGA

FPGAs as End PCIe Devices

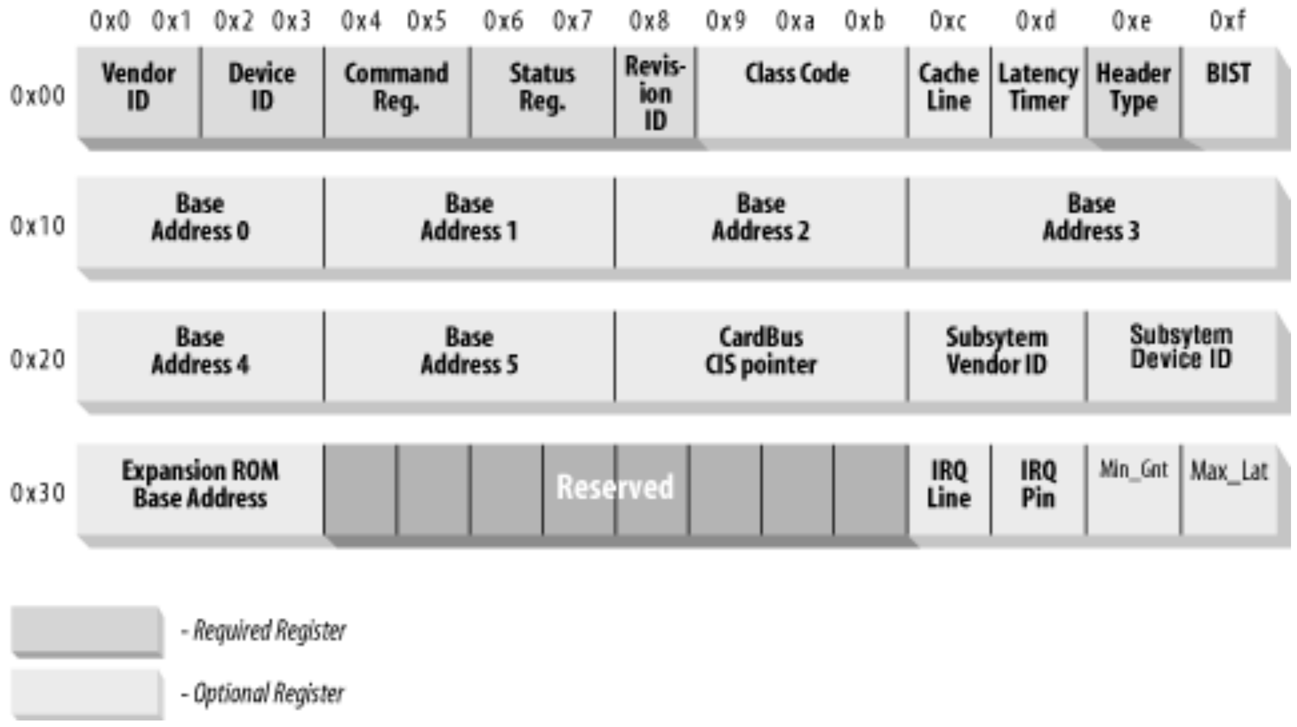
- Peripheral Component Interconnect (PCI) is a standard computer bus for connecting devices to the CPU
- PCIe is the Peripheral Component Interconnect Express bus introduced in 2003, de-facto standard for all peripherals
  - There is also SATA for disks, and new standards for SSD drives
- Each PCI peripheral is identified by a **bus** number, a **device** number, and a **function** number: BDF triplet
- Each FPGA will have assigned BDF number in the system
  - Virtual numbers, managed by the hypervisor


Hypervisor emulates a virtual bridge, so guest VMs can only access their assigned physical devices
PCI Bus 0
PCI Bus 1
RAM
CPU
Host Bridge
PCI Bridge
ISA Bridge
CardBus Bridge
FPGA PCIe connection

Text and images adapted from:
Linux Device Drivers, 3rd Edition, Chapter 12
by Greg Kroah-Hartman, Alessandro Rubini, Jonathan Corbet

Interacting with PCIe Devices

- PCI and PCIe devices are access through a set of memory-mapped registers
- All PCI devices feature at least a 256-byte address space; the first 64 bytes are standardized, while the rest are device dependent:
- Further configuration and interaction is done through memory regions mapped at the base address registers (BAR)
  - PCIe example on right shows 6 BARs
  - Each BAR is a memory-mapped region
  - Side and location is specified by the corresponding BAR in the configuration space



	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x00	Vendor ID		Device ID		Command Reg.		Status Reg.		Revis-ion ID	Class Code			Cache Line	Latency Timer	Header Type	BIST
0x10	Base Address 0				Base Address 1				Base Address 2				Base Address 3			
0x20	Base Address 4				Base Address 5				CardBus CIS pointer				Subsystem Vendor ID		Subsystem Device ID	
0x30	Expansion ROM Base Address				Reserved								IRQ Line	IRQ Pin	Min_Gnt	Max_Lat

▨ - *Required Register*

□ - *Optional Register*

Text and images adapted from:
Linux Device Drivers, 3rd Edition, Chapter 12
by Greg Kroah-Hartman, Alessandro Rubini, Jonathan Corbet

Interacting with PCIe Devices from Applications

- Linux (or in general OS) drivers are responsible to configure the devices
- Users can write or read data via the BARs to get or put data to or from device

Different BARs can exist for:

- Directly read or write data
  - Writing data causes it to be written to a register or buffer in FPGA
  - Writing data is done by **`poke()`** commands in Amazon F1
  - Reading data causes data from FPGA register or buffer to be sent to user
  - Reading data is done by **`peek()`** command in Amazon F1
- Setup information about DMA transaction
  - Write data such as: amount of data to copy, DRAM address, copy to or from FPGA
  - Write anther configuration register to trigger the DMA

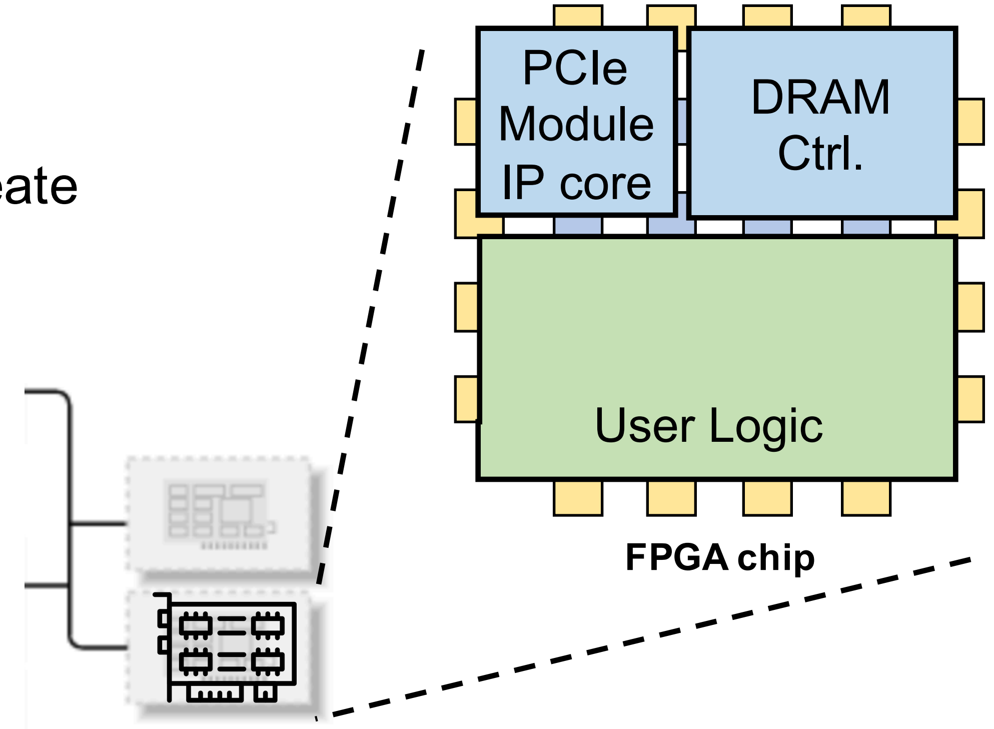
More later in the course on PCIe and AXI bus…

Executing Designs and Data & IP Privacy Issues

Compile and Load Designs

Details of working with Amazon F1 Cloud FPGAs will be given in following lectures throughout the semester, this is a high-level summary…

- A hardware development kit (HDK) needs to be used to create and compile a design for the FPGA
  - Design must include required parts such as PCIe, which is part of the *shell* in AWS terminology
  - Other parts are optional and up to the user, e.g. DRAM and user logic; all must pass design check rules
- The FPGAs are loaded with Amazon FPGA Images (AFIs), which are pretty much just *bitstreams* generated by the FPGA tools
  - Once loaded, user can interact with the hardware



Even idle or "unused" FPGA needs some minimal bitstream loaded, PCIe is used to program the FPGA; FPGA may be flushed with default bitstream from Flash memory when users are switched

Data & IP Privacy Issues

Using Amazon F1 as an example, the cloud provider has access to all the design details in the digital check points

- Need to read the license agreement and privacy policies
- Effectively cloud provider has access to whole hardware (and software) design

Data protection

- Data transfer between server and FPGA is not encrypted
- Data in FPGA's DRAM is not encrypted
  - May be scrambled by DRAM modules
- Possible to physically probe buses, debug via JTAG, etc.
  - Cloud provider controls *shell*, PCIe code, DRAM code, etc.

Certification of cloud providers

- For use with sensitive (government or medical data) cloud providers need to get certified
- Check procedures, maybe code, and form promise by the cloud provider to follow certain rules

Questions?

A Step Back: Mapping from HDLs to FPGAs

Prof. Jakub Szefer
Dept. of Electrical Engineering
Yale University

From HDL to FPGA Configuration Bitstream

- FPGA needs to be programmed to actually perform useful operation
- Usually start with hardware designs written or described in a Hardware Description Language (HDL)
  - Verilog, SystemVerilog, VHDL, or generated from HLS

- **Synthesize** to logic blocks
- **Place or map** logic blocks in FPGA
- **Route** connections between logic blocks
- **Generate** FPGA programming file

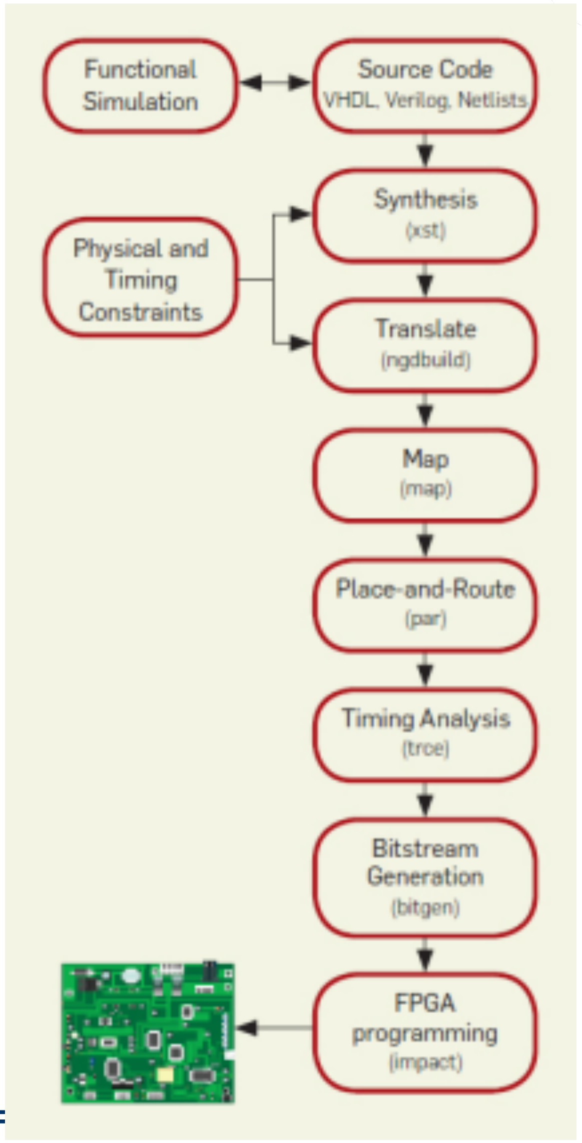
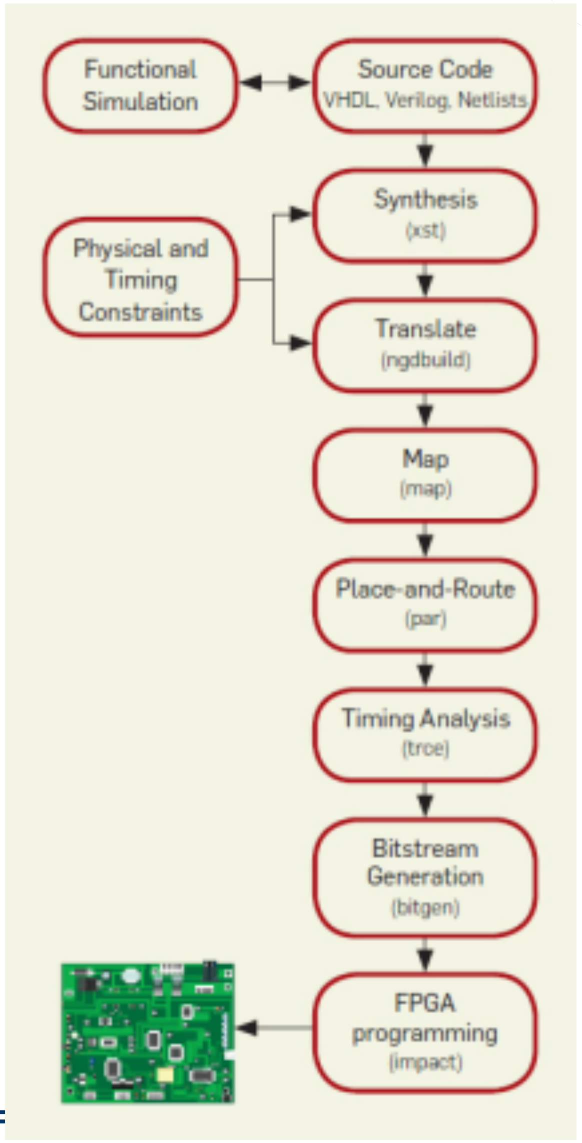

Functional Simulation
Source Code VHDL, Verilog, Netlists
Physical and Timing Constraints
Synthesis (xst)
Translate (ngdbuild)
Map (map)
Place-and-Route (par)
Timing Analysis (trce)
Bitstream Generation (bitgen)
FPGA programming (impact)

Figure from Bacon, et al. [4]

Routing Logic and Generate Bitstream

- Once the locations for all the logic gates in a circuit have been selected, routing needs to be performed to connect the CLBs together via the available wires
- Once routing is selected, then it can be determined which connections in the switchboxes should be turned on to connect the required wires and CLBs
  - Two-level routing:
    - First select routing channels, but not specific wires (global routing)
    - Second, select wires within routing channels (local routing)
- Once the CLB configurations are selected and the switchbox configurations are selected, generate a bitstream according to the FPGA vendor's format



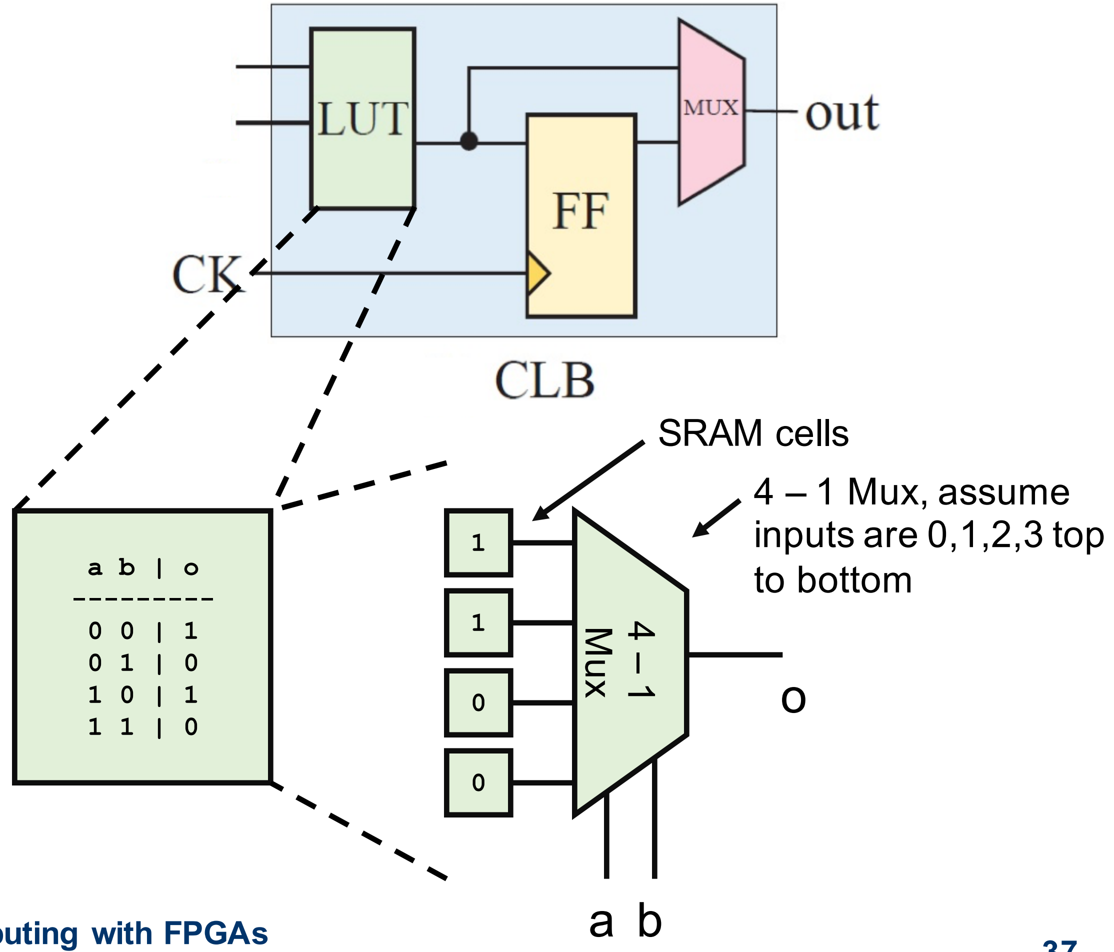
Functional Simulation
Source Code VHDL, Verilog, Netlists
Synthesis (xst)
Physical and Timing Constraints
Translate (ngdbuild)
Map (map)
Place-and-Route (par)
Timing Analysis (trce)
Bitstream Generation (bitgen)
FPGA programming (impact)

Figure from Bacon, et al. [4]

Look-Up Tables (LUTs) Review

- Look-Up Tables (LUTs) are part of Configurable Logic Blocks (CLBs)
  - They are used to realize combinatorial logic
  - Need to define contents of each LUT

- Example, 2-input LUT can realize
  - Any function of 2 inputs
  - Any function of 1 input

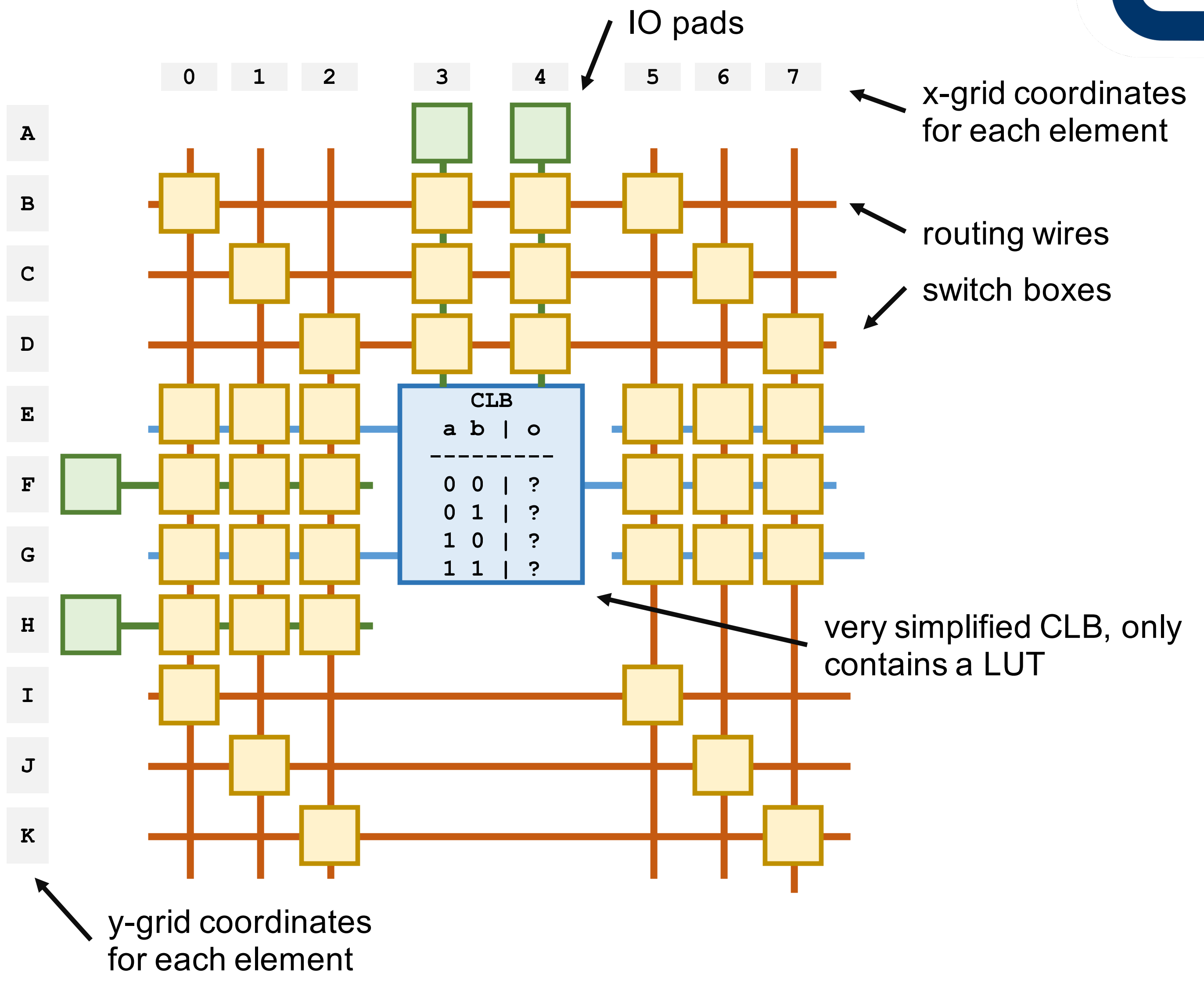


```
a b | o
---------
0 0 | 1
0 1 | 0
1 0 | 1
1 1 | 0
```

A Simplified FPGA: Floorplan

- Components of a very simplified FPGA:
  - CLBs with LUTs
  - Routing wires
  - Switchboxes
  - IO pads

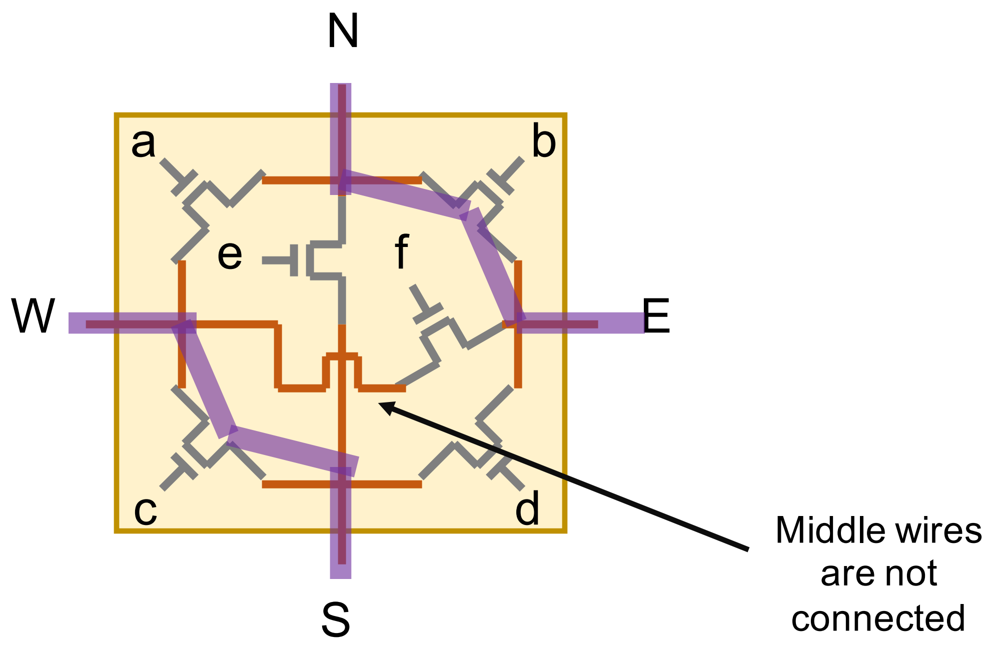
- Can reference the elements by their (x,y) coordinates


IO pads
0
1
2
3
4
5
6
7
x-grid coordinates for each element
routing wires
switch boxes
A
B
C
D
E
F
G
H
I
J
K
CLB
a b | o
----
0 0 | ?
0 1 | ?
1 0 | ?
1 1 | ?
very simplified CLB, only contains a LUT
y-grid coordinates for each element

A Simplified FPGA: Switchbox

- A switchbox is used to connect routing wires
- An example switchbox with 6 possible connections
  a) N to W
  b) N to E
  c) W to S
  d) S to E
  e) N to S
  f) W to E
- Set control bit to 1 to connect, set control bit to 0 to keep connection open
- Example configuration

a	b	c	d	e	f
0	1	1	0	0	0



Example: 4-input AND Gate Circuit

- A 4-input AND gate is a logic gate that performs logical AND of four 1-bit inputs
  - Truth table for the 4-input AND

a	b	c	d	y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

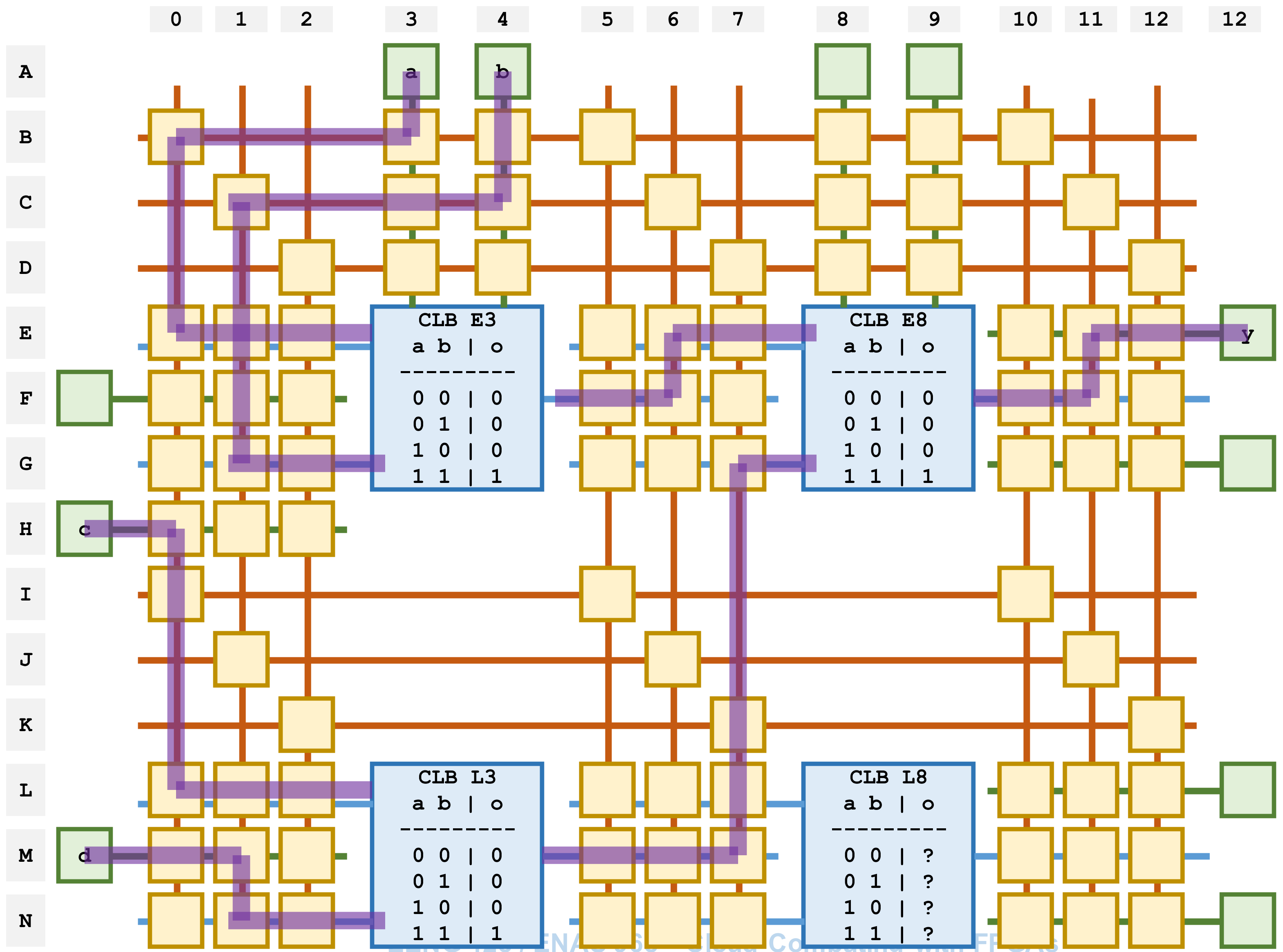
- Mapping 4-input AND to 2-input, 1-output LUTs requires 3 LUTs
  - AND(a,b,c,d) = AND(AND(a,b), AND(c,d))

a	b	y′
0	0	0
0	1	0
1	0	0
1	1	1

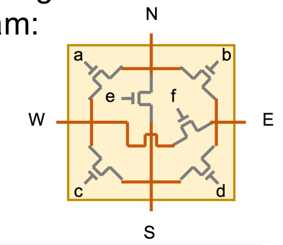
c	d	y″
0	0	0
0	1	0
1	0	0
1	1	1

y′	y″	y
0	0	0
0	1	0
1	0	0
1	1	1

Mapping 4-input AND to a Simplified FPGA



Switchbox diagram and bitstream:



	a	b	c	d	e	f
B0	0	0	0	1	0	0
B3	1	0	0	0	0	0
B4	0	0	0	0	1	0
B5	0	0	0	0	0	0
...						
C1	0	0	0	1	0	0
C3	0	0	0	0	0	1
C4	1	0	0	0	0	0
...						

Example: Half-Adder Circuit

- Half-Adder is a simple circuit that performs addition of two 1-bit numbers **a** and **b**, and generates output of sum, **s**, and carry out, **c**
  - Truth table for the half-adder:

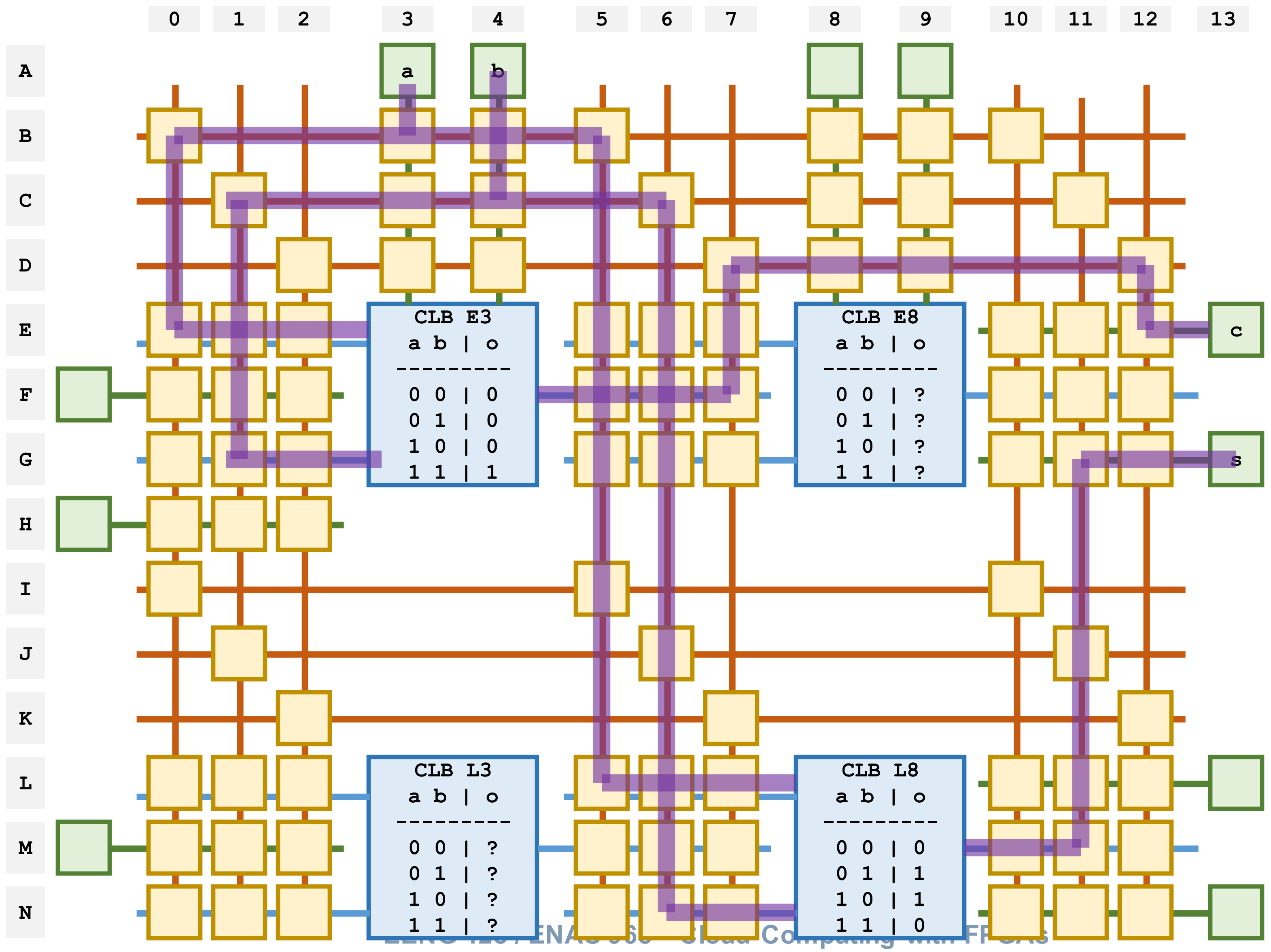
a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- Mapping half-adder to 2-input, 1-output LUTs requires 2 LUTs
  - One LUT for c output, and one LUT for s output

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

Mapping Half-Adder to a Simplified FPGA



CLB E3

a	b	o
0	0	0
0	1	0
1	0	0
1	1	1

CLB E8

a	b	o
0	0	?
0	1	?
1	0	?
1	1	?

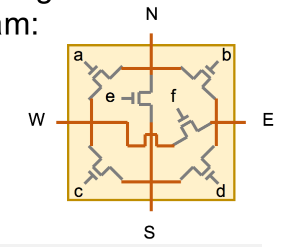
CLB L3

a	b	o
0	0	?
0	1	?
1	0	?
1	1	?

CLB L8

a	b	o
0	0	0
0	1	1
1	0	1
1	1	0

Switchbox diagram and bitstream:

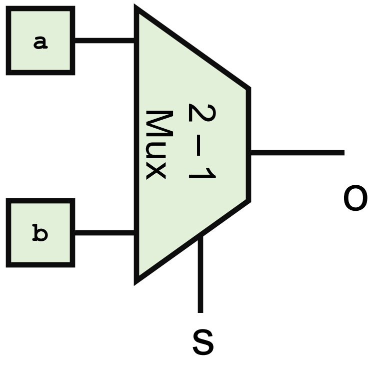


	a	b	c	d	e	f
B0	0	0	0	1	0	0
B3	1	1	0	0	0	0
B4	0	0	0	0	1	1
B5	0	0	1	0	0	0
...						
C1	0	0	0	1	0	0
C3	0	0	0	0	0	1
C4	1	1	0	0	0	0
...						

Share: bit.ly/cloudfpga

Practice Examples

- A 2-1 Mux has 3 inputs: 2 data inputs and 1 select signal and 1-output gate
  - Select s = 0 select input a, else select input b

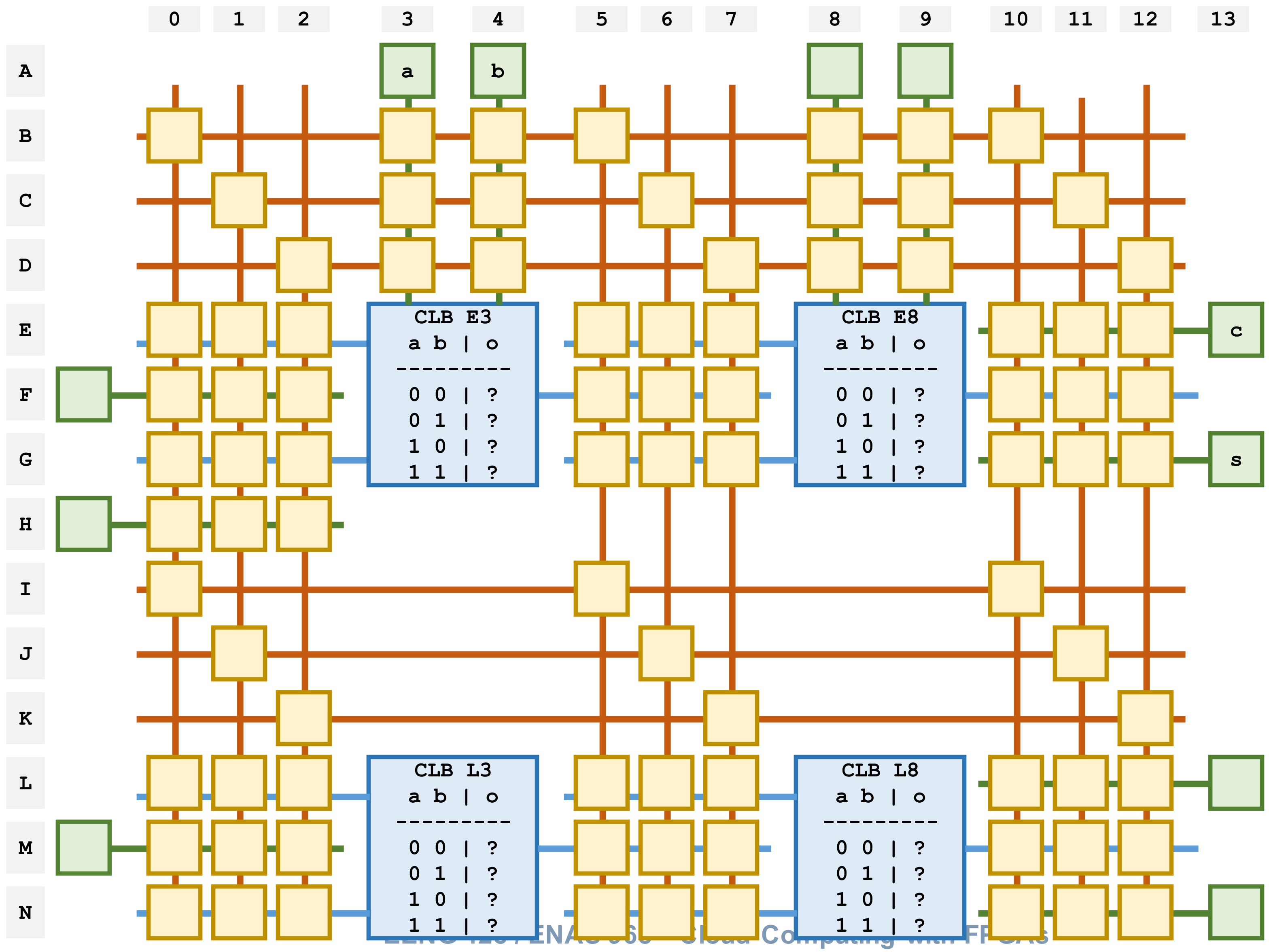


a	b	s	o
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

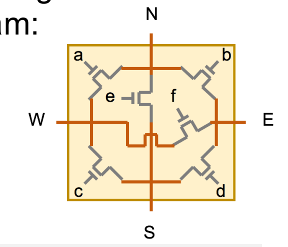
- A Full-Adder is a circuit that adds two 1-bit inputs **a** and **b** as well as a carry-in signal **ci**, and outputs sum **s** and carry out **co**

a	b	ci	co	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Empty Floorplan for Practice Examples



Switchbox diagram and bitstream:



	a	b	c	d	e	f
B0	?	?	?	?	?	?
...						

Questions?

References Used in Slides

1. Lucian Armasu. “AMD, Xilinx Claim World Record for Machine Learning Inference.” Tom’s Hardware, Oct. 2018. Available at: https://www.tomshardware.com/news/amd-xilinx-machine-learning-inference-record,37885.html.
2. Arne Verheyde. “Xilinx One-Ups Intel With PCIe 4.0 Alveo U50 Data Center Card.” Tom’s Hardware, Aug. 2019. Available at: https://www.tomshardware.com/news/xilinx-alveo-u50-accelerator-card-pcie-4-hbm,40111.html.
3. Intel® FPGA Programmable Acceleration Card D5005. Available at: https://www.intel.com/content/www/us/en/programmable/products/boards_and_kits/dev-kits/altera/intel-fpga-pac-d5005/overview.html.
4. Xilinx Starts Sampling 7nm Versal FPGA. Available at: https://www.tomshardware.com/news/xilinx-shipping-versal-acap-ai-prime,39661.html.
5. Intel® Agilex™ FPGA Product Brief. Available at: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/solution-sheets/intel-agilex-fpga-product-brief.pdf.
6. Xilinx Alveo, Adaptable Accelerator Cards for Data Center Workloads. Available at: https://www.xilinx.com/products/boards-and-kits/alveo.html.
7. Xilinx Virtex UltraScale+ FPGA VCU1525 Acceleration Development Kit. Available at: https://www.xilinx.com/products/boards-and-kits/vcu1525-a.html.
8. Large-Scale Reconfigurable Computing in a Microsoft Datacenter. Available at: https://www.microsoft.com/en-us/research/uploads/prod/2014/06/HC26.12.520-Recon-Fabric-Pulnam-Microsoft-Catapult.pdf.

References Used in Slides

9. "Amazon (company)." Wikipedia, The Free Encyclopedia. Available at: https://en.wikipedia.org/w/index.php?title=Amazon_(company)&oldid=910179769.
10. "Amazon Web Services." Wikipedia, The Free Encyclopedia. Available at: https://en.wikipedia.org/w/index.php?title=Amazon_Web_Services&oldid=909747505.
11. Amazon EC2 Instance Types. Available at: https://aws.amazon.com/ec2/instance-types/.
12. Amazon EC2 F1 Instances. Available at: https://aws.amazon.com/ec2/instance-types/f1/.
13. Announcing improved networking performance for Amazon EC2 instances. Available at: https://aws.amazon.com/about-aws/whats-new/2017/09/announcing-improved-networking-performance-for-amazon-ec2-instances/.
14. Fixed broadband speeds are getting faster — what's fastest in your city? Vox. Sep 7, 2017. Available at: https://www.vox.com/2017/9/7/16264430/fastest-broadband-speeds-ookla-city-internet-service-provider.
15. "Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency". Available at: https://www.xilinx.com/support/documentation/white_papers/wp380_Stacked_Silicon_Interconnect_Technology.pdf.
16. Gabe Loh. "Die Stacking Integration: Technologies and Applications". ACACES Summer School 2019.
17. "Bandwidth vs. Latency: What is the Difference?". Available at: https://www.highspeedinternet.com/resources/bandwidth-vs-latency-what-is-the-difference.
18. FCC. "2016 Measuring Broadband America Fixed Broadband Report". Available at: https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-fixed-broadband-report-2016.

References Used in Slides

19. “Smart SSD, Computational Storage Drive”, https://www.xilinx.com/content/dam/xilinx/publications/product-briefs/xilinx-smartssd-computational-storage-drive-product-brief.pdf
20. “VectorPath Accelerator Card”, https://www.achronix.com/product/speedster7t-fpgas