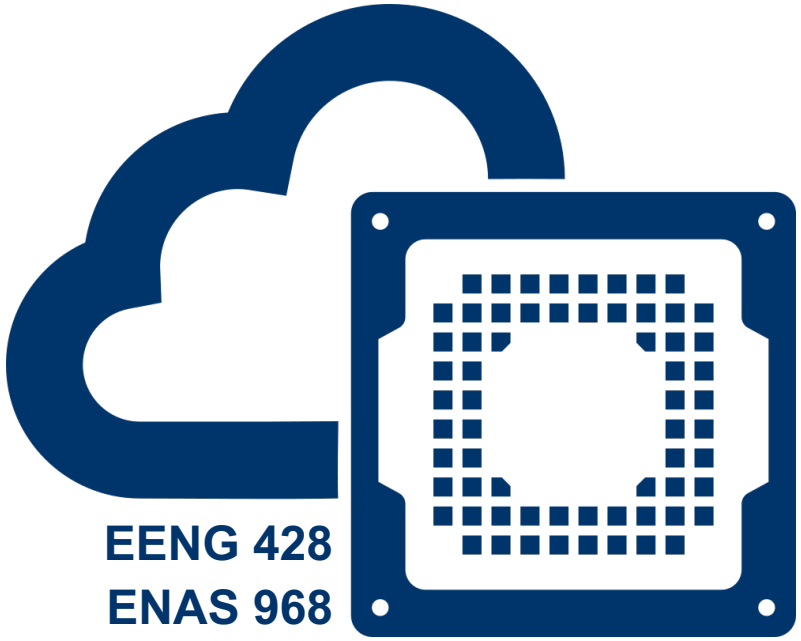
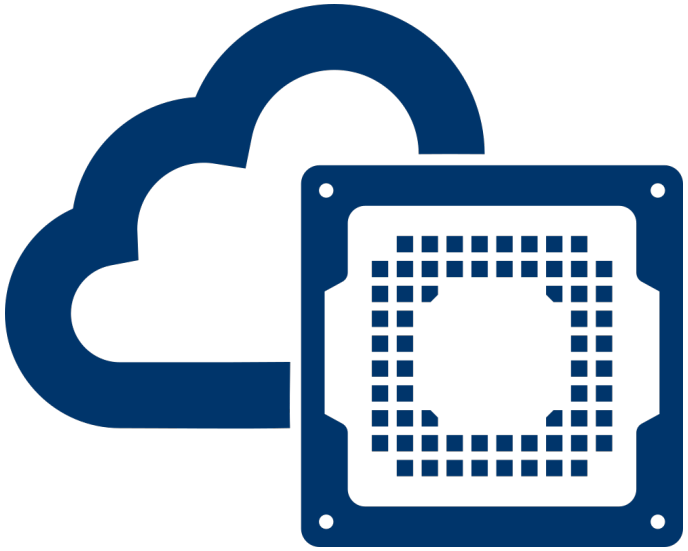
Cloud FPGA



bit.ly/cloudfpga



Lecture: Cloud FPGA Infrastructures

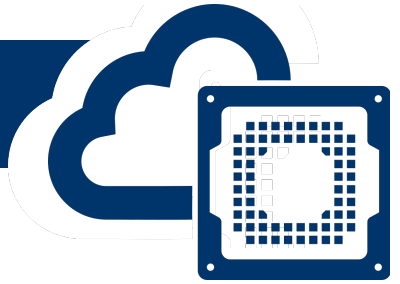
Prof. Jakub Szefer
Dept. of Electrical Engineering, Yale University

EENG 428 / ENAS 968
Cloud FPGA

Public Cloud FPGA Infrastructures

- **In recent 2~3 years, there has been an emergence of public cloud providers offering FPGAs for customer use in their data centers** (as of 2019):
  - Xilinx Virtex UltraScale+: Amazon AWS, Huawei Cloud, and Alibaba Cloud
  - Xilinx Kintex UltraScale: Baidu Cloud and Tencent Cloud
  - Xilinx Alveo Accelerator: Nimbix
  - Intel Arria 10: Alibaba Cloud and OVH
  - Intel Stratix V: Texas Advanced Computing Center (TACC)
  - Intel Stratix 10: Microsoft Azure (for AI applications)
- Most infrastructures let users load any hardware design (with limitations imposed by the underlying FPGA and design rule checks)
- Some infrastructures only give indirect access to FPGA, e.g., via HLS
- Most use pay-as-you-go model, or require users to rent compute resources up-front for a specified amount of time (e.g. per month)

Amazon AWS as a Public Cloud Example

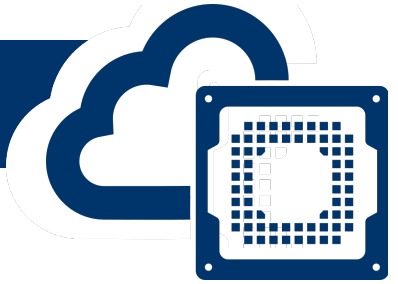


- **Amazon Web Services (AWS)** is a major public cloud infrastructure provider
  - A subsidiary of Amazon
  - Provides on-demand cloud computing platforms to individuals, companies, and governments
  - AWS operating income was $7.2 billion in 2018
  - Amazon as a whole, had operating income of $12.4 billion in 2018 (AWS contributed then 58%)
  - AWS page is at https://aws.amazon.com

AWS services relevant to Cloud FPGAs include:

- Amazon Elastic Compute Cloud (EC2) – on-demand access to virtual servers in the cloud
- Amazon Elastic Block Storage (EBS) – block storage devices (i.e. “hard drives” for EC2)
- Amazon Simple Storage Service (S3) – data object storage service for sharing files (think of it like Dropbox)

Running Virtual Machines in AWS



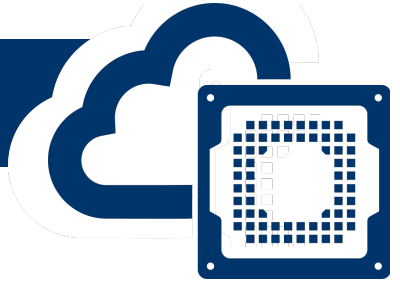
Abstracting away the details, users wishing to run virtual machines (VMs) in AWS EC2 need to decide the type of virtual machine image and the type of hardware to run the VM on

- **Virtual Machine Image** (OS, libraries, applications, etc.)
  - Amazon calls these **Amazon Machine Image (AMI)**
  - Basically disk image of a pre-installed operating system (OS), with some libraries and applications pre-installed as well
- **Type of Hardware** (CPU, memory, disk, optionally GPU, optionally FPGA)
  - Amazon calls these **Instance Types**
  - Physical hardware is shared by many users, e.g. 8 core CPU may be used to give 2 cores to 4 users (without oversubscription)
  - Three instance types exist for FPGAs, they are called F1 in Amazon

Model	FPGAs	vCPU	Mem (GiB)	SSD Storage (GB)	Networking Performance
f1.2xlarge	1	8	122	470	Up to 10 Gigabit
f1.4xlarge	2	16	244	940	Up to 10 Gigabit
f1.16xlarge	8	64	976	4 x 940	25 Gigabit

F1 instance type info from [11]

Virtual Machine Costs in EC2

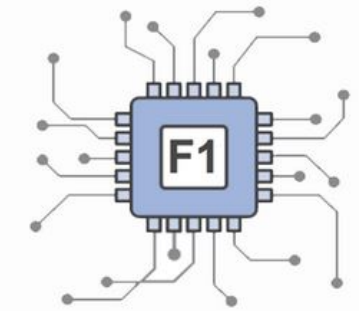


- **Common AMIs are free**, but there is marketplace for purchasing AMIs with may come with some proprietary software or features
- **Major costs are associated with the instance types**, additional storage, networking, accelerators, etc.
  - vCPU, Memory, and Storage: https://aws.amazon.com/ec2/pricing/on-demand/
  - Additional storage: https://aws.amazon.com/s3/pricing/#Storage_pricing
  - Data transfer: https://aws.amazon.com/s3/pricing/#Data_Transfer_pricing
- **Payment options**
  - Free Instances – t2.micro instances
  - Pay-as-you-go Instances – pay as you use the resources
    - On-Demand instances, pay for EC2 resources by per hour or per second depending on instance type
    - **Spot instances**, bid for EC2 resources, can be kicked off when others pay more, but can save up to 90%
  - Reserve Instances – pre-pay for the resources up front, about 40% savings
  - Dedicated Hosts – get whole EC2 server to your self, up to 70% cheaper over pay-as-you-go, but need get whole server

Can use Spot instances for F1 FPGAs

Amazon F1 Instances

- Amazon EC2 F1 instances use FPGAs to enable users to deploy custom hardware accelerators
- Amazon EC2 F1 instance types introduced in beta in 2016, and general availability in 2017
  - Amazon EC2 itself was introduced in 2006
- Three different types of instances are available (as of 2019):



Name	FPGAs	vCPUs	Instance Memory (GiB)	SSD Storage (GB)	Enhanced Networking	EBS Optimized	On-Demand Price/hr*	1-yr Reserved Instance Effective Hourly*	3-yr Reserved Instance Effective Hourly*
f1.2xlarge	1	8	122	470	Yes	Yes	$1.65	$1.06	$0.76
f1.4xlarge	2	16	244	940	Yes	Yes	$3.30	$2.12	$1.52
f1.16xlarge	8	64	976	4 x 940	Yes	Yes	$13.20	$8.50	$6.10

- Spot pricing can be seen at https://aws.amazon.com/ec2/spot/pricing/
  - During off-peak, e.g. weekends, there is 5% additional discount

Amazon Datacenters with F1 Instances

- Amazon data centers are distributed throughout the world
- F1 instances are available to general public in limited regions, as of Oct. 2018:
  - US East (N. Virginia)
  - US West (Oregon)
  - Europe (Ireland)
  - AWS GovCloud (US) regions
- Previews are available in:
  - Europe (Frankfurt and London)
  - Asia Pacific (Sydney)
- Benefits of different regions:
  - Physical location (legal requirements for data storage
  - Network latency

AWS Inter-Region Latencies

- Data is available for latencies between AWS datacenters: https://www.cloudping.co/
- Inter-region latencies from ~20ms to ~250ms
- Random data about US broadband [17]:

Technology	Internet Provider	Bandwidth	Latency
Cable	XFINITY by Comcast, Charter Spectrum, Cox Communications	41 Mbps	28ms
DSL	AT&T, Frontier, CenturyLink	16 Mbps	44ms
Fiber	AT&T, Frontier, CenturyLink	73 Mbps	17ms
Satellite	Hughesnet	7 Mbps	603ms

  - Federal Communications Commission (FCC) 2016 report provides similar data [18]
- Advertised 4G wireless: ~100Mbps, ~20ms latency
- Latency is finally limited by the physical distance

AWS Inter-Region Latency

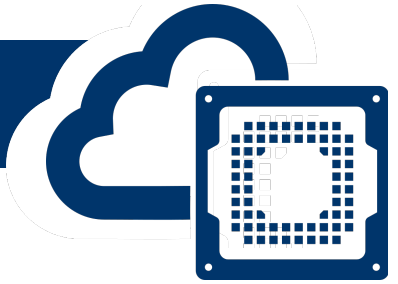
The data in the table below represents the averages of the previous 24 hours of data collected. Hover over a cell for the weekly average.

Destination Region / Source Region	ap-northeast-1	ap-northeast-2	ap-south-1	ap-southeast-1	ap-southeast-2	ca-central-1	eu-central-1	eu-north-1	eu-west-1	eu-west-2	eu-west-3	sa-east-1	us-east-1	us-east-2	us-west-1	us-west-2
ap-northeast-1		43.79	132.18	75.89	112.34	164.64	247.83	266.04	219.46	229.79	256.17	289.57	161.38	159.85	122.35	103.09
ap-northeast-2	41.10		160.83	101.86	150.45	195.05	275.26	290.85	248.91	256.57	287.53	317.59	192.66	190.76	142.42	131.62
ap-south-1	127.11	156.16		58.09	235.40	202.01	119.27	146.48	126.52	115.52	113.72	311.45	188.19	196.52	235.04	222.17
ap-southeast-1	72.70	98.70	60.27		182.56	224.32	180.77	202.78	181.23	173.17	168.97	351.97	231.91	229.09	177.54	164.80
ap-southeast-2	119.73	164.92	237.41	197.35		234.12	293.70	320.98	277.93	276.73	294.46	350.73	206.72	208.77	169.17	146.95
ca-central-1	159.36	190.81	203.66	225.74	220.30		105.02	128.76	81.87	91.96	99.47	131.38	21.09	26.39	91.70	73.85
eu-central-1	245.49	272.37	120.51	178.13	286.67	103.36		29.12	26.89	16.95	12.94	211.65	88.53	99.17	148.70	166.28
eu-north-1	257.35	282.98	143.89	200.26	311.16	125.19	29.55		47.30	39.65	34.58	237.45	113.75	124.50	173.23	178.75
eu-west-1	220.79	244.19	124.47	180.89	274.53	84.41	26.67	48.09		16.61	20.89	191.60	77.55	101.00	158.18	138.27
eu-west-2	223.74	245.34	114.98	169.14	275.91	91.08	17.93	43.12	13.12		11.00	199.49	79.58	90.05	140.74	144.89
eu-west-3	249.56	276.98	110.01	163.59	281.24	97.60	12.17	34.62	21.35	10.18		202.83	83.08	92.99	142.37	156.44
sa-east-1	298.11	337.87	344.06	378.21	335.63	146.10	211.44	263.57	190.80	226.13	236.31		125.59	140.95	224.33	199.27
us-east-1	163.36	187.92	187.24	230.32	202.18	18.59	90.02	116.28	76.25	79.84	82.35	126.10		16.27	64.47	77.64
us-east-2	166.52	192.21	202.00	231.56	199.80	32.70	101.79	126.09	100.11	92.58	94.34	141.34	18.60		56.15	77.10
us-west-1	114.06	139.00	233.19	179.13	154.99	87.32	151.27	175.93	150.53	141.43	141.73	204.08	63.39	54.93		23.85
us-west-2	104.00	138.80	223.81	166.69	141.81	74.08	165.14	181.14	137.77	147.35	159.19	193.76	77.08	72.87	24.43	

Latency: < 100ms 100-180ms > 180ms

All times are in milliseconds.

Potential for Cloud FPGAs Revisited



- 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
- Other high-performance components in the cloud (fast networking, fast storage, etc.)

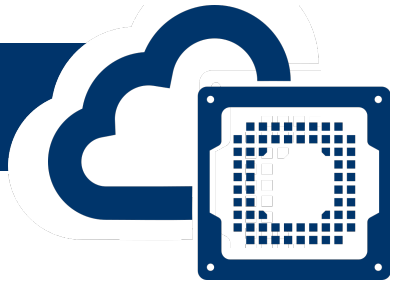
Challenges:

- Need to write the hardware design, e.g. Verilog, test, and deploy
- Need to understand FPGAs and whole system to get best results
- Getting data to and from Cloud FPGAs (via the internet)

Limitations:

- Homogenous deployment, can only get one type of FPGA typically
- Cloud FPGAs are pay-as-you-go, with enough usage own FPGAs may be better
- Less control over data, cloud provider (today) can often access data

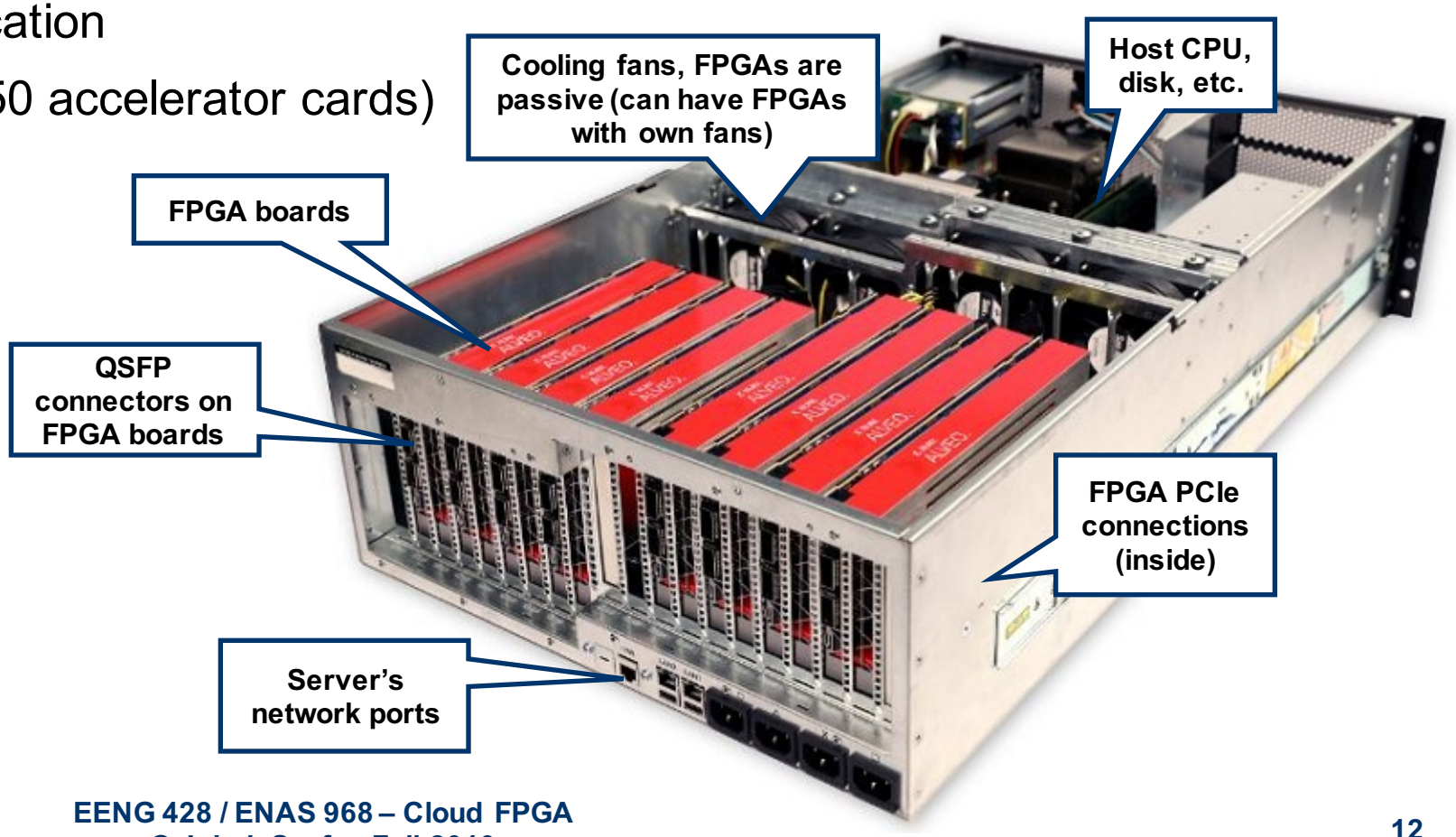
Cloud FPGA Servers and FPGA Hardware



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)

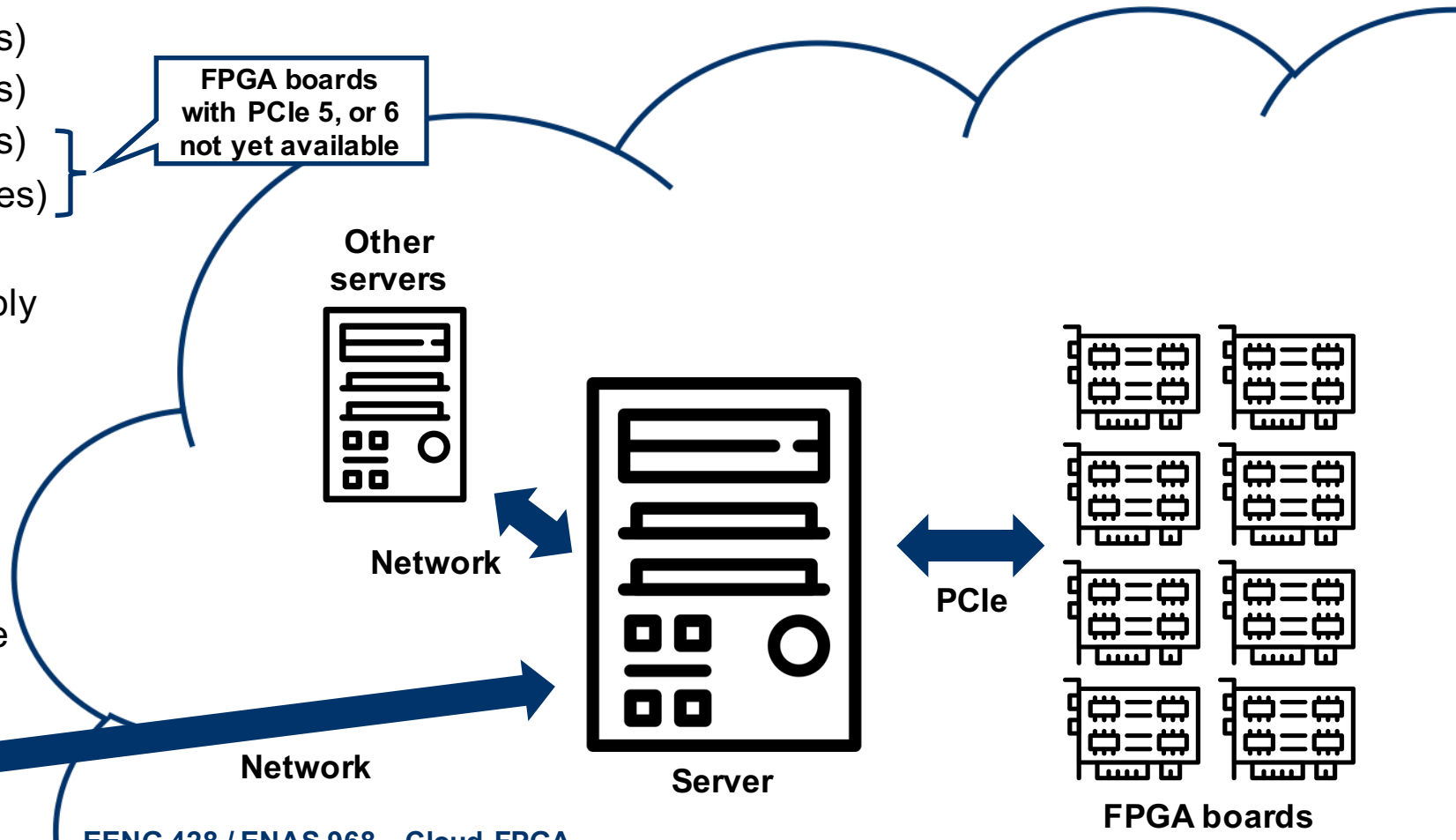


Share: bit.ly/cloudfpga

Image and server information from [1]

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]

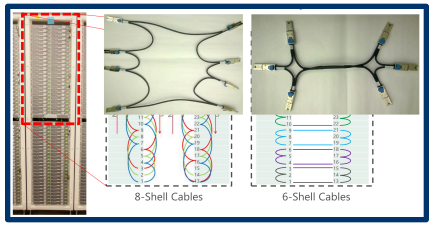


Cloud FPGA Servers – FPGA Interconnections

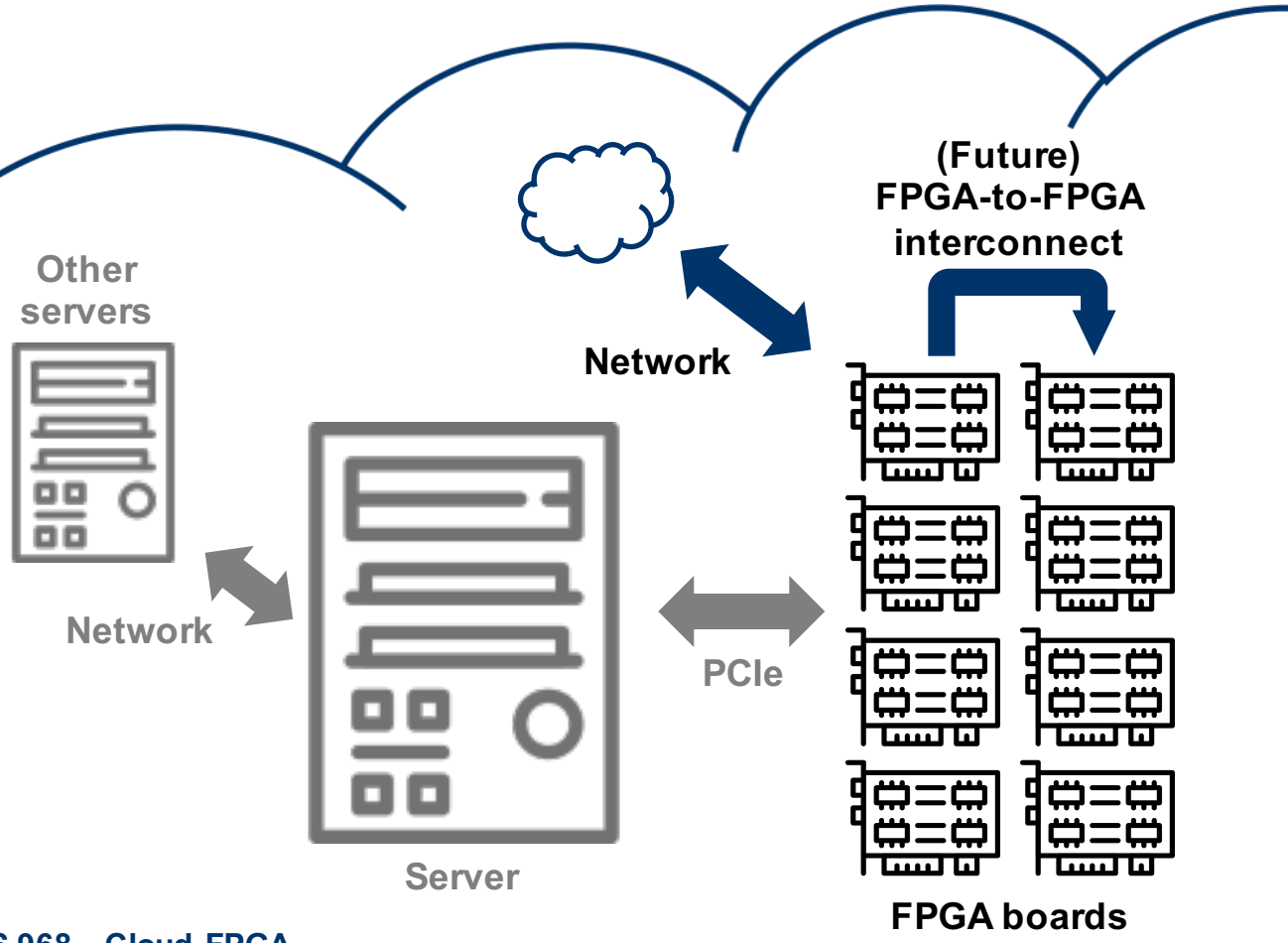
- Future options for interconnecting Cloud FPGAs
  - Many FPGA boards have QSFP connectors
    - QSFP: 4 Gbit/s
    - QSFP+: 40 Gbit/s
    - QSFP14: 50 Gbit/s
    - QSFP28: 100 Gbit/s

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

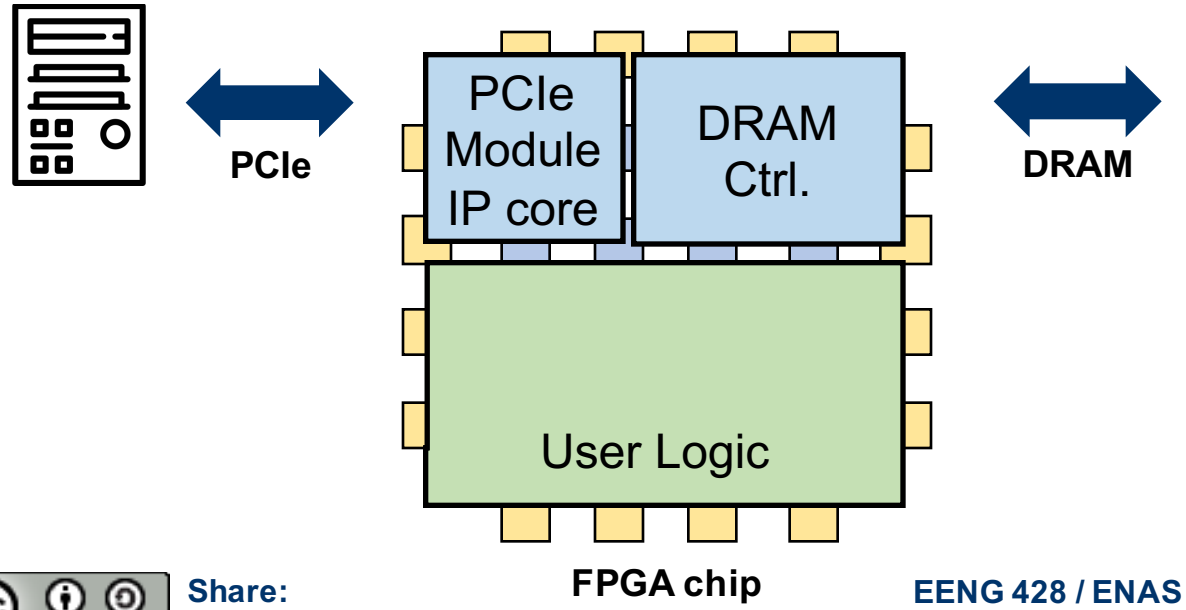


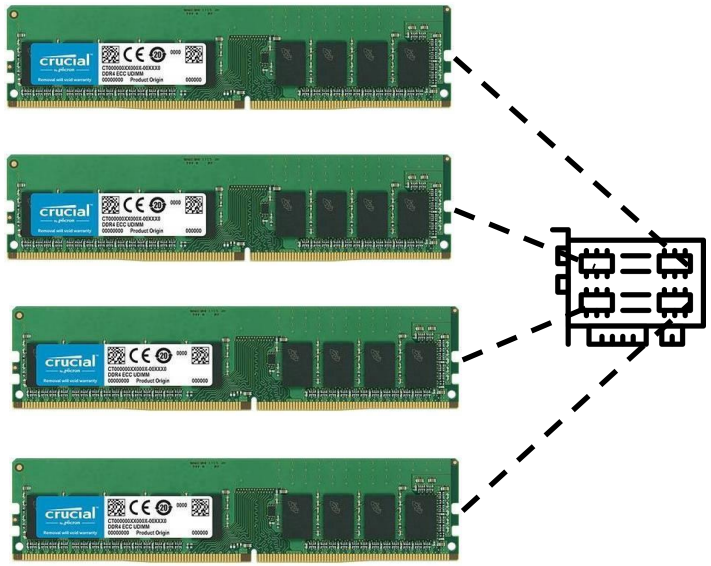
Share: bit.ly/cloudfpga

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





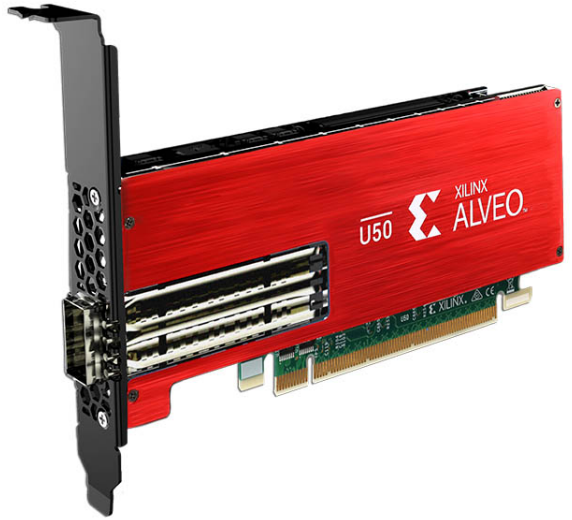
DRAM images from https://www.newegg.com/p/0RN-003K-00071

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

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



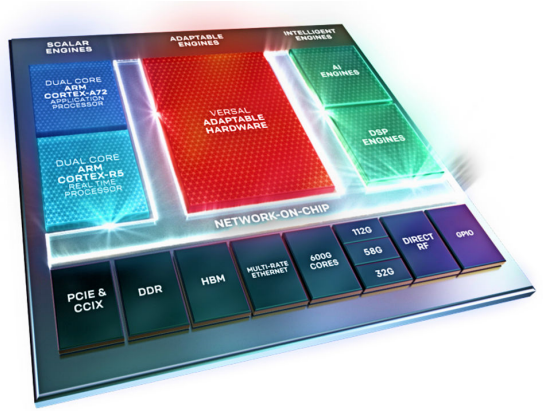
- Intel D5005 Programmable Acceleration Card (2019)
  - Uses Intel's 14nm Stratix 10 SX FPGA architecture
  - 2,800,000 logic elements
  - Thermal Design Power (TDP) of 215W
  - PCIe 3.0 and DDR4
  - QSFP networking

Share: bit.ly/cloudfpga

Images and information from [2] and [3]

Upcoming 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

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

Share: bit.ly/cloudfpga

Images and information from [4] and [5]

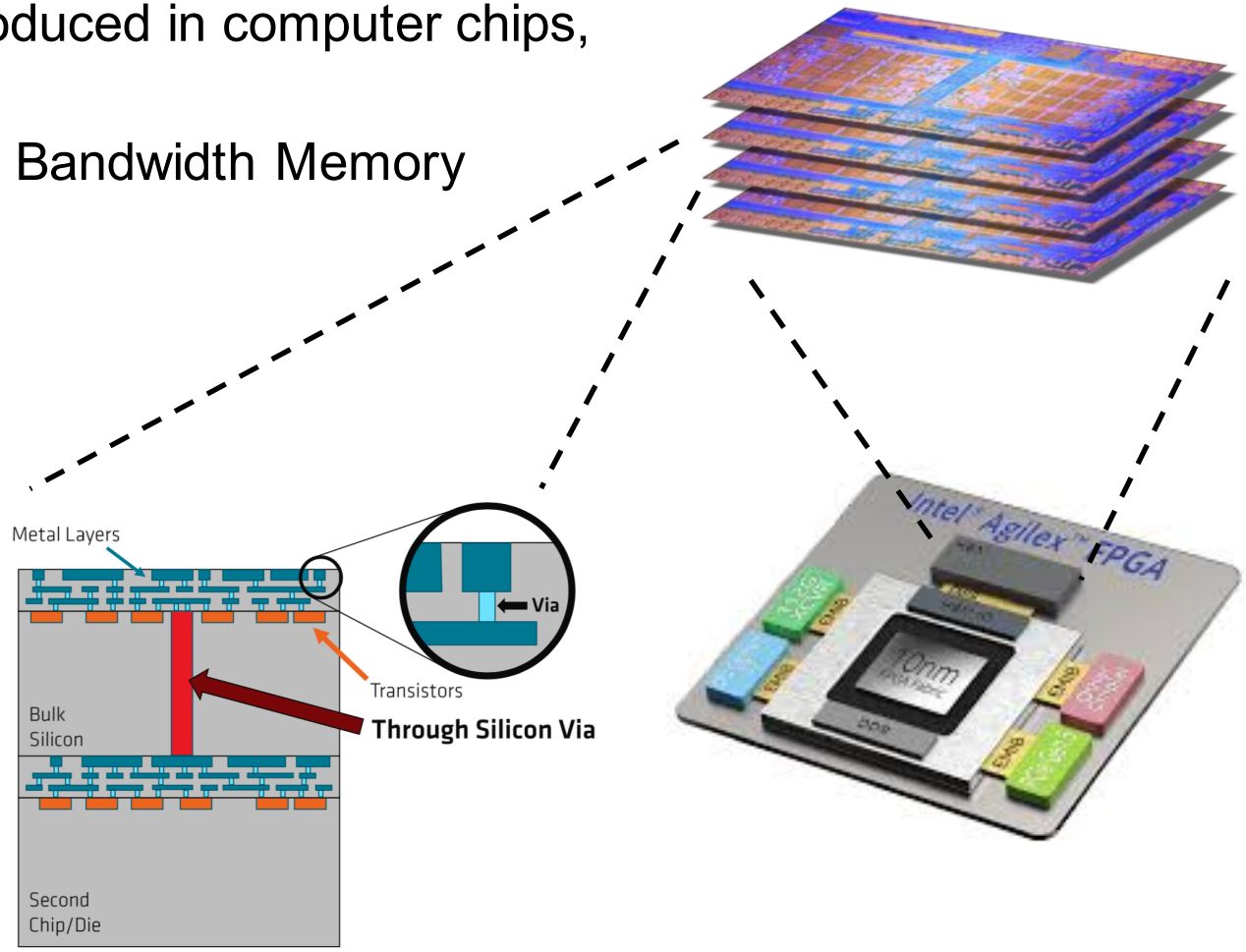
Vertically Integrating More Components onto a Chip

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)



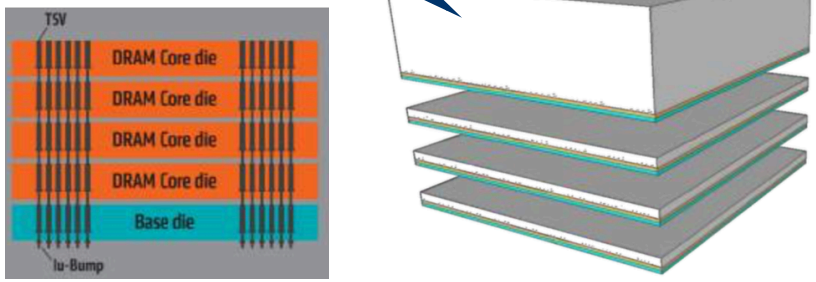
Share: bit.ly/cloudfpga

Agilex image from [5]

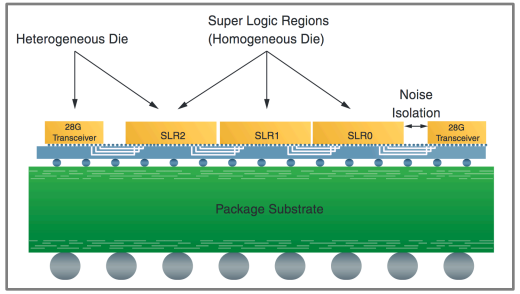
Die Stacking: 2.5D, 3D, and Beyond

Die stacking images and information from Gabe Loh [16]

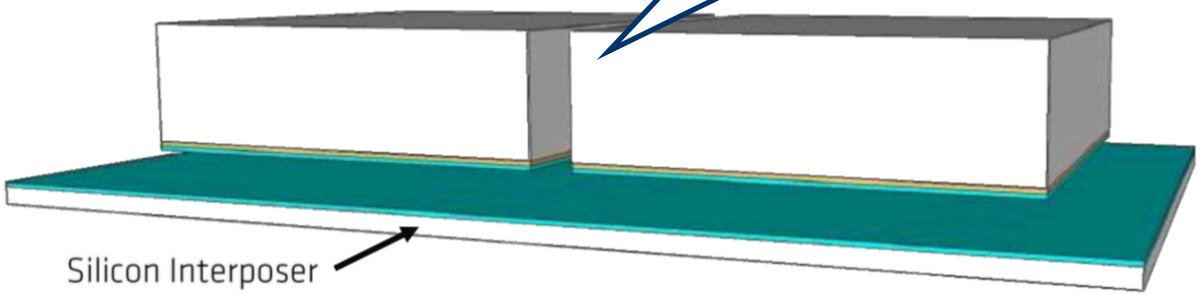
Currently used in HBM DRAM memories



3D Stacking



2.5D stacking is already used in UltraScale chips from Xilinx



2.5D Stacking

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

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

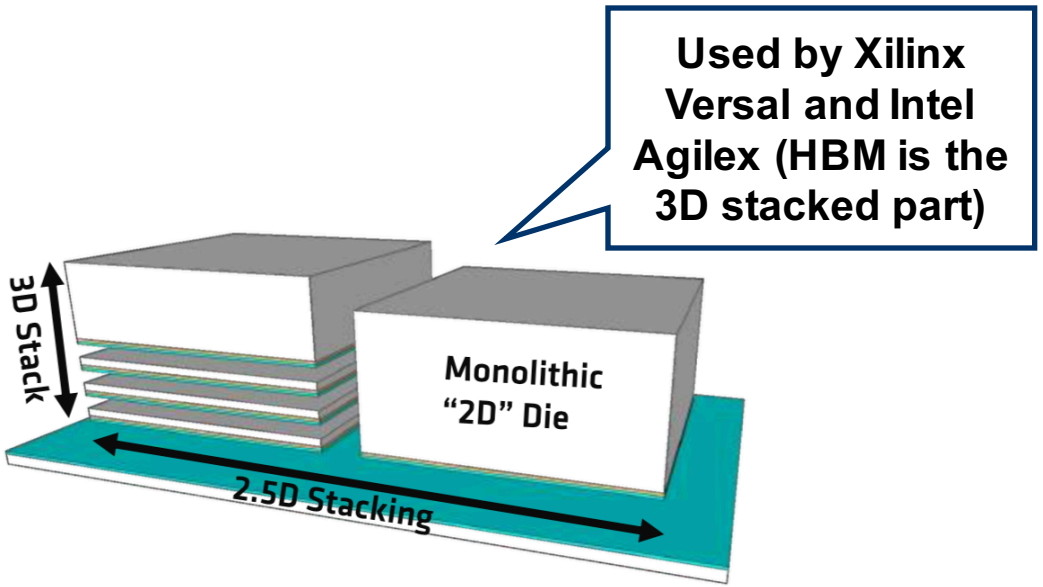
- Only one layer high
- Variable die sizes
- Cooling is easier
- More pins, but bigger area
- Power delivery is easier

Share: bit.ly/cloudfpga

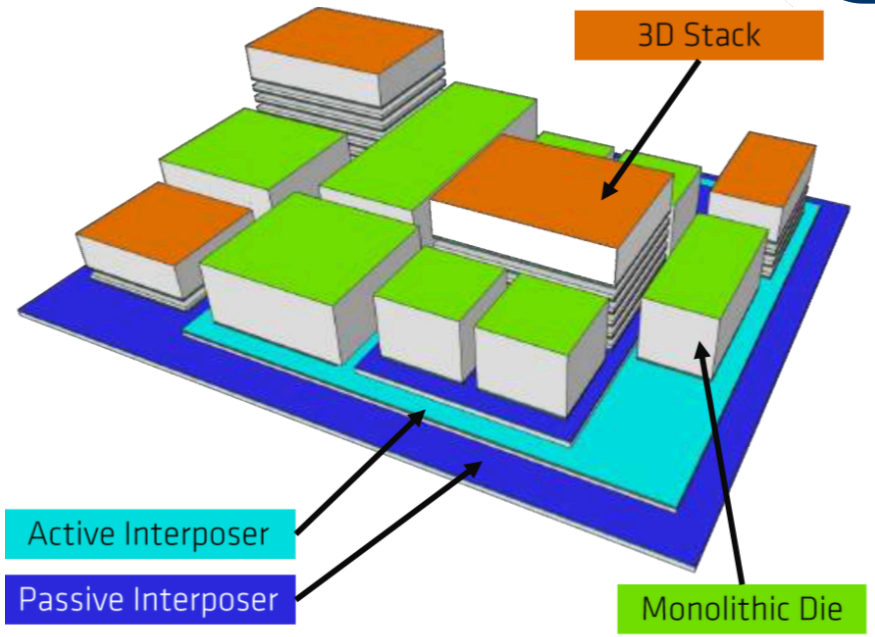
Xilinx die diagram from [15]

Die Stacking: 2.5D, 3D, and Beyond

Die stacking images and information from Gabe Loh [16]



5.5D Stacking



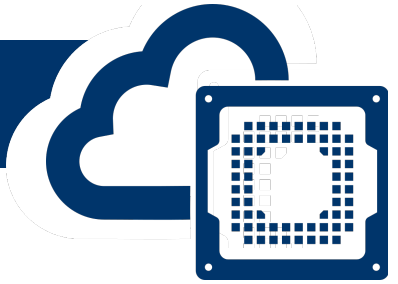
Infinite Stacking Options

- 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

Share: bit.ly/cloudfpga

FPGA "Free Lunch" with New Technologies



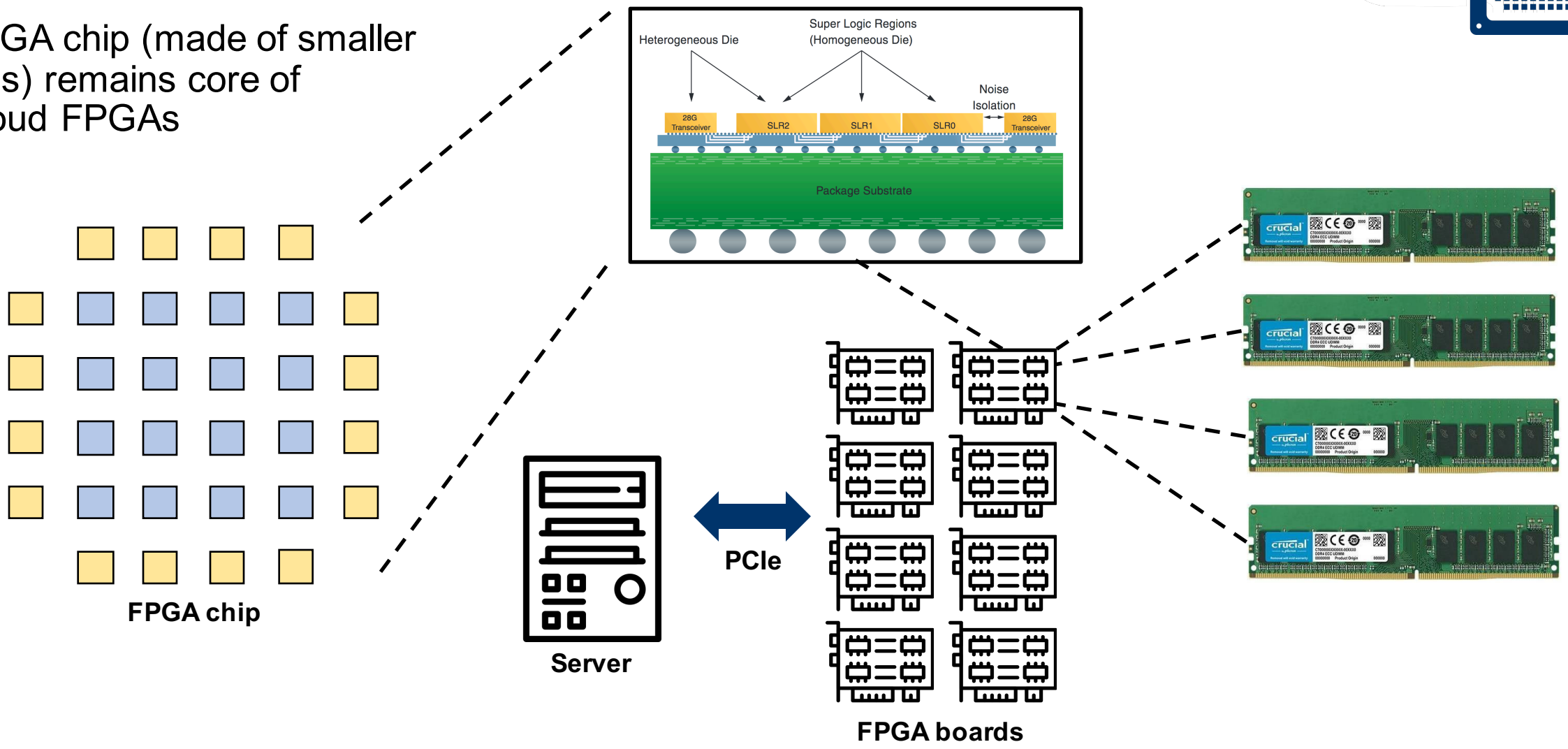
Future hardware allowing for bigger FPGAs, with more features

- 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

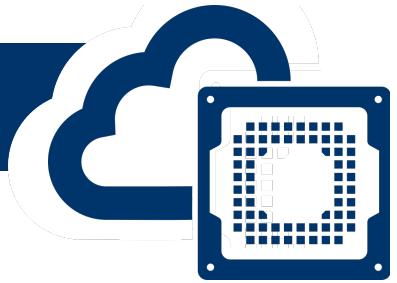
Summary of Typical Cloud FPGA Hardware

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

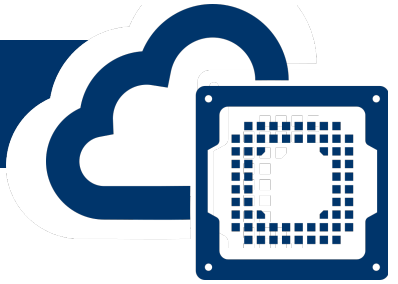


Share: bit.ly/cloudfpga

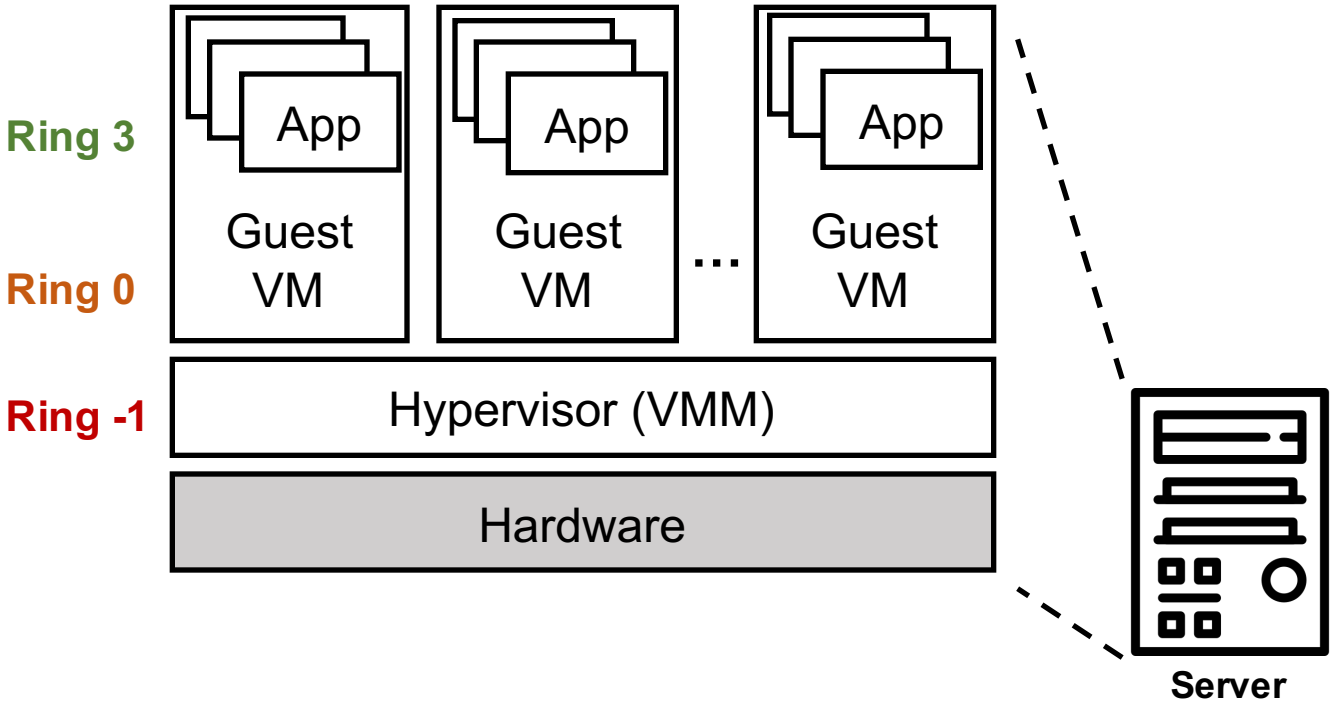
Cloud FPGA Server Software



Cloud FPGA Server Software Stack

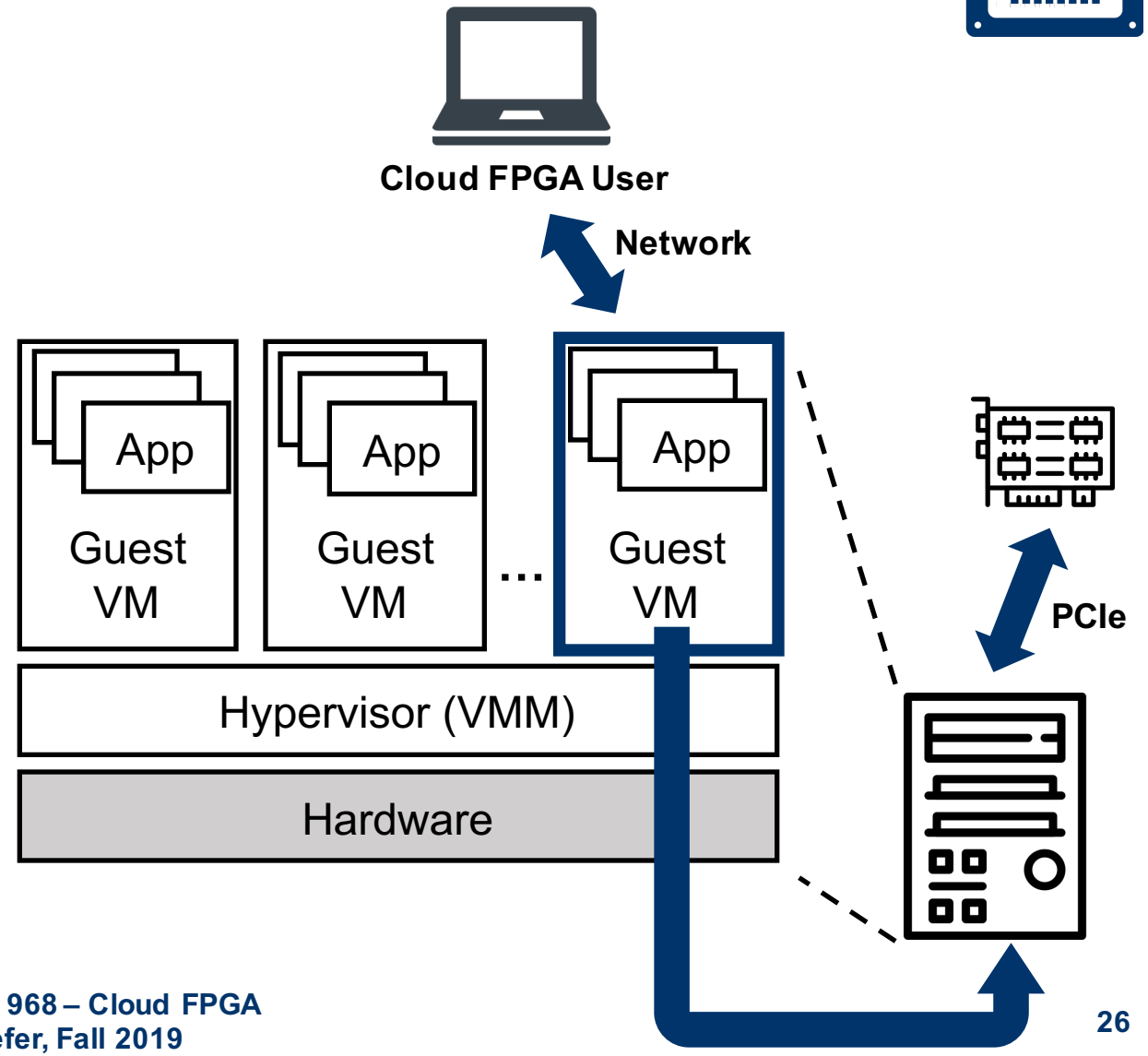


- Server software stack includes hypervisor, guest VMs, and user applications running on the VMs
- In IaaS setting:
  - Hypervisor is controlled by the cloud provider
  - Hypervisor controls all hardware
  - Users provide guest VM images (AMIs in Amazon's terminology)
  - Users provide applications and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



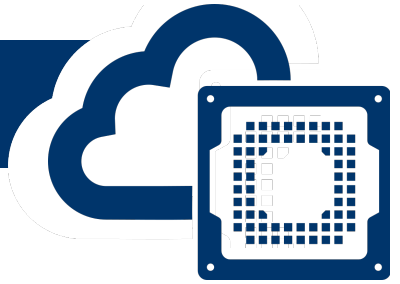
Leveraging Hypervisor for Server Management

- 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



Share: bit.ly/cloudfpga

Guest VM Software and Libraries

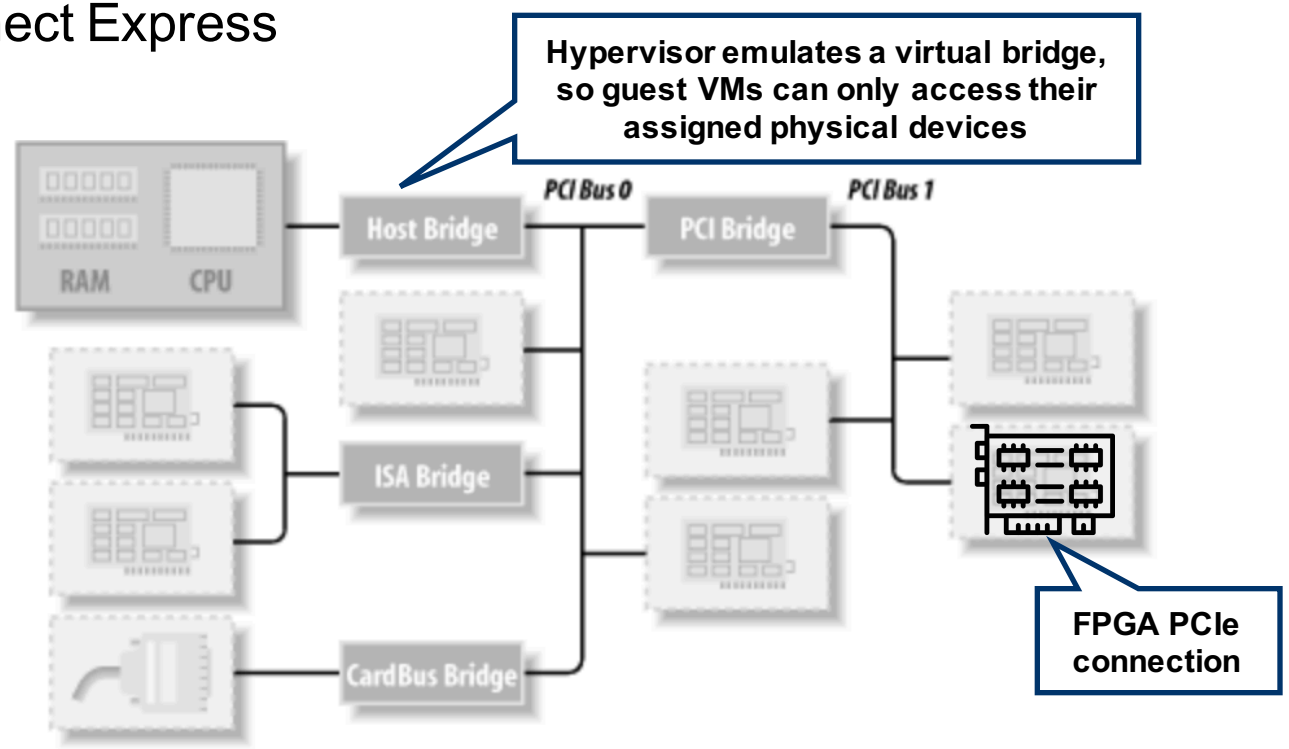


- FPGA programming tools
  - Verilog, VHLD, SystemVerilog, etc.
  - High-Level Synthesis
  - E.g. Xilinx Vivado or Intel Quartus
- FPGA programming tools can be run: locally by user, on VM (without FPGA), on VM (with FPGA)

- 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

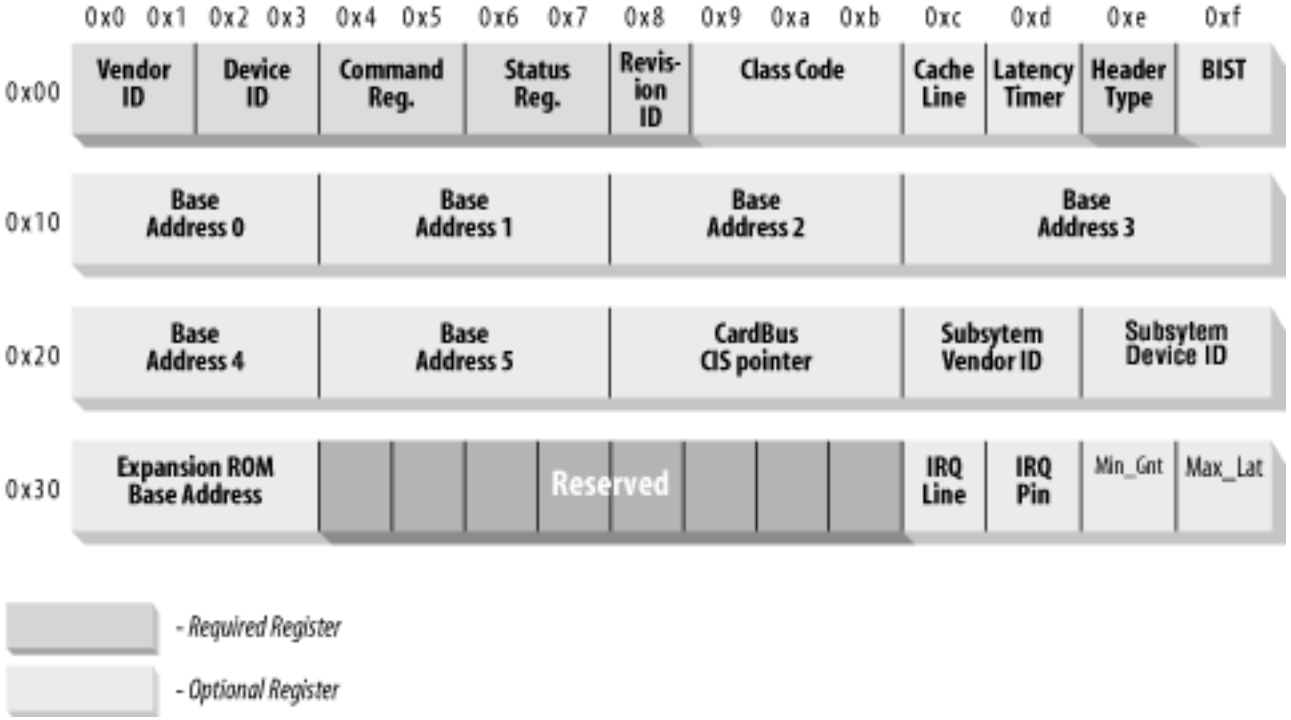


Text and images adapted from:
Linux Device Drivers, 3rd Edition, Chapter 12
by Greg Kroah-Hartman, Alessandro Rubini, Jonathan Corbet

Share: bit.ly/cloudfpga

Interacting with PCIe Devices

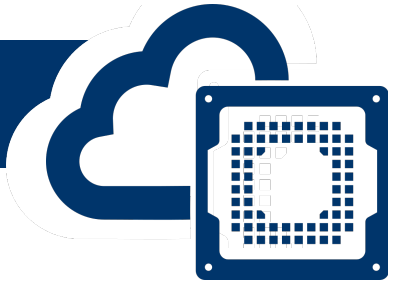
- PCI and PCIe devices are access through a set of memory-mapped registers
- All PCI devices feature at least a 256-byte address space; the first 64 bytes are standardized, while the rest are device dependent:



- 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

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



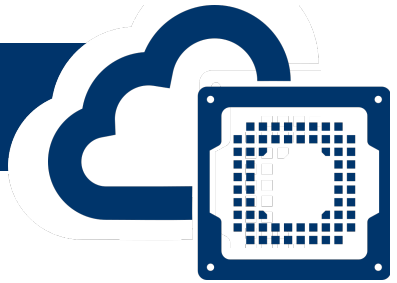
- Linux (or in general OS) drivers are responsible to configure the devices
- Users can write or read data via the BARs to get or put data to or from device

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…

Run Time of a Hardware Design in Amazon F1

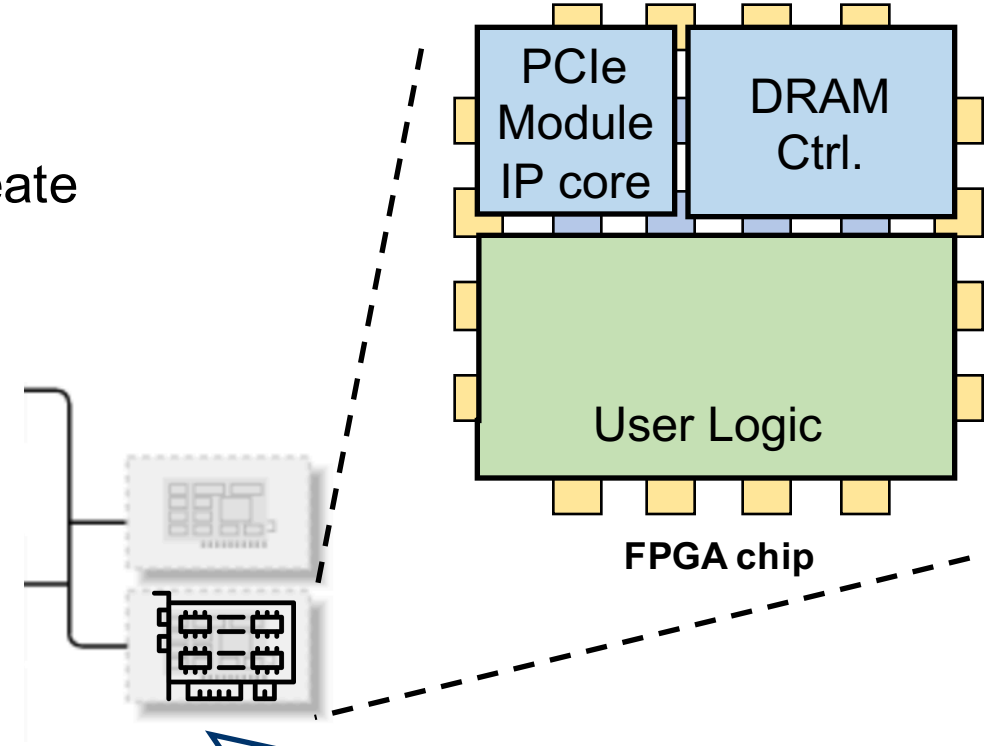


Share:
bit.ly/cloudfpga

Compile and Load Designs

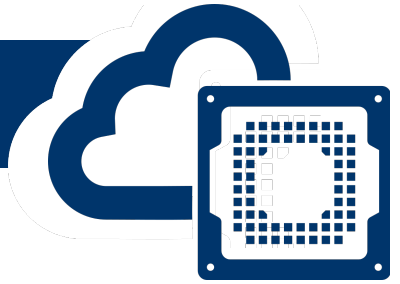
Details of working with Amazon F1 Cloud FPGAs will be given in following lectures throughout the semester, this is a high-level summary…

- 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

Using F1 Cloud FPGAs: Command Line Interface



- A command line interface (CLI) can be used to interact with the FPGAs:
  - Generate AFIs (bistreams)
  - Get status of AFIs
  - Get status of FPGA board
  - Load design onto FPGA
  - Use virtual LEDs, virtual dipswitches, or virtual JTAG
  - Interact with software (C or Python libraries)
  - Clear the FPGA board

```
$ sudo fpga-clear-local-image  -S 0
```

```
$ sudo fpga-describe-local-image -S 0 -H

Type  FpgaImageSlot  FpgaImageId             StatusName    StatusCode   ErrorName    ErrorCode
AFI          0       none                    cleared           1        ok               0
Type  FpgaImageSlot  VendorId    DeviceId    DBDF
AFIDEVICE    0       0x1d0f      0x1042      0000:00:0f.0
```

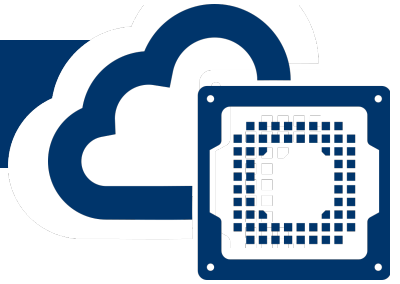
```
$ sudo fpga-load-local-image -S 0 -I agfi-0f0e045f919413242
```

```
$ sudo fpga-describe-local-image -S 0 -R -H

Type  FpgaImageSlot  FpgaImageId             StatusName    StatusCode   ErrorName    ErrorCode
AFI          0       agfi-0f0e045f919413242  loaded            0        ok               0
Type  FpgaImageSlot  VendorId    DeviceId    DBDF
AFIDEVICE    0       0x6789      0x1d50      0000:00:0f.0
```


Share: bit.ly/cloudfpga

C and Python Libraries



Amazon provides libraries for both C and Python for programs to interact with the F1 instances running in the cloud:

`fpga_mgmt`

- Get FPGA status, load image, clear image, etc.
- Read virtual LEDs
- Set virtual dip switches

`fpga_pcie`

- PCIe setup related
- **`peek()`** and **`poke()`** implementations

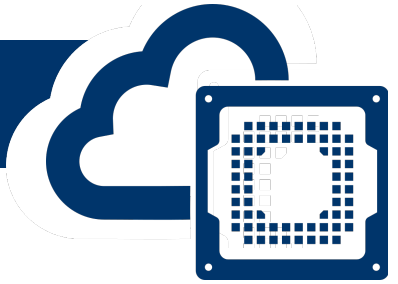
`fpga_dma` – functions to control Direct Memory Access

- Setup DMA
- Copy data from device
- Copy data to device

C header files in aws-fpga git:
aws-fpga/sdk/userspace/include/

Python binding in aws-fpga git:
aws-fpga/sdk/userspace/python_bindings/

Some Performance Improving Features



DMA (Direct Memory Access)

- Allow for direct memory copying between server's DRAM and FPGA (registers or DRAM)
- Bandwidth approaching 8GB/s (depends on HDK version, PCIe driver, etc.)

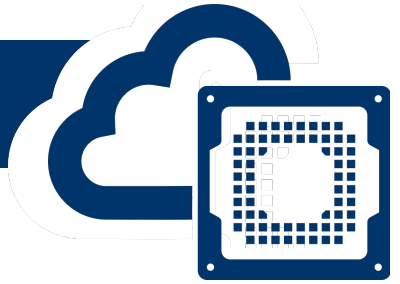
Data Retention in DRAM

- FPGA's DRAM is cleared when new user is assigned an FPGA
- While user controls FPGA, and loads new designs, DRAM data is maintained
  - Use one design to load or generate data, store in DRAM
  - Use another design to process data from DRAM
  - Avoid costly (taking time) data movement to and from FPGA when switching designs

Inter-FPGA high-speed links (future)

- High-speed communication between FPGAs
  - QSFP+: 40Gb/s for 4 channels fully used by FPGAs
  - PCIe 4.0: 31.51 GB/s for x16, but shared with server and other FPGAs

Clear FPGA and Save Data

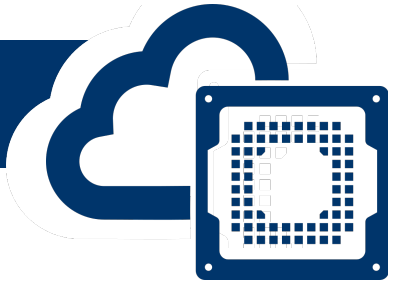


- End of hardware design's run time in Cloud FPGAs:
  1. Compile and load design
  2. Use Cloud FPGAs
  3. Clear FPGA and save data

- Data generated by FPGA needs to be moved from FPGA to the server, then usually to Amazon storage (such as S3) or downloaded by the user

- The bitstream (referenced by AFI or GAFI) can remain in Amazon, and be re-used later
  - Can share GAFI with other users so they don't have to compile the design

Need for Safety Checks



- Ability to load arbitrary bitstreams (AFIs) can create danger for the Cloud FPGA infrastructure and other users of Cloud FPGAs

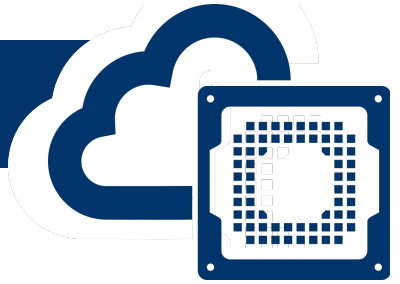
- Cloud FPGA Security is new and active research field

E.g. recent research by:
Prof. Tessier (web page), Prof. Holcomb (web page), and Prof. Szefer (web page)

Potential dangers of malicious hardware:

- Designs that dissipate a lot of power
  - Damage FPGA
  - Damage servers
  - Damage other infrastructure
- Designs that create "sensors", to learn otherwise forbidden information (e.g. data center temperature, FPGA voltage, etc.)
- Designs that leak information through side-channels
- Unknown dangers to future multi-tenant Cloud FPGAs (if they become reality)

Safety Checks in Cloud FPGAs



- Different cloud providers deploy different levels of safety checks
  1. None – let users do whatever they want, mostly in academic clouds
  2. Check some features of design – use design rules and check for FPGA software warnings or errors
  3. Synthesize hardware designs fully by cloud provider – e.g. user provide high-level design in C, and cloud provider runs HLS tools to generate hardware

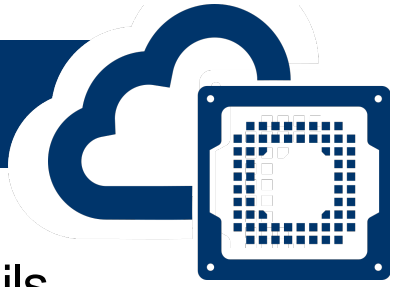
Approximate safety checks done by Amazon F1:

- Check proper tool's version, HDK version, *shell* version, etc.
- Users submit digital checkpoint files (DCPs)
  - Users generate DCPs, submit to Amazon
  - Amazon runs partial synthesis on DCPs
  - Checks for problems, errors, warnings (e.g. no ring oscillators are allowed)

> ***.dcp** files generated by Xilinx (Vivado) tools are the design check points, which are created after each step of the compilation: synthesis, optimization, place & route, etc.

- Run-time monitoring of power
  - Throttle FPGA clocks, or even shut down if too much power is used (more information is here)

Privacy Considerations



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

References

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

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.