FPGA Technology at Crossroads

Vaughn Betz, Derek Chiou, Franz Franchetti,
James C. Hoe, David Z. Pan, Vyas Sekar, Justine Sherry

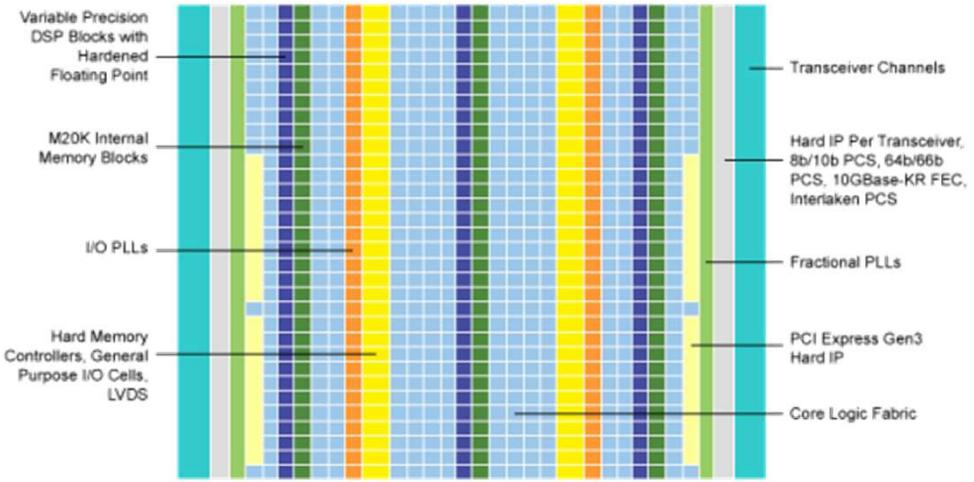
Intel/VMware Crossroads 3D-FPGA
Academic Research Center

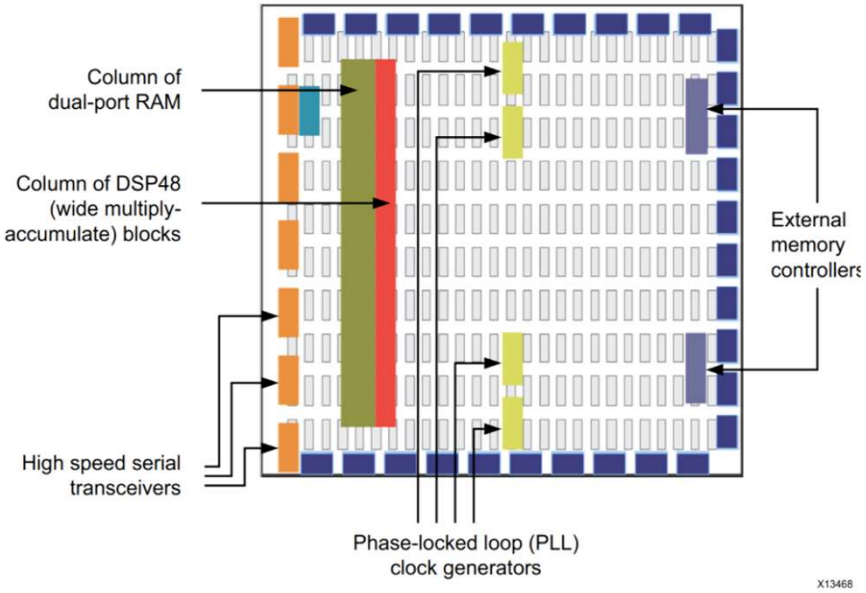
with generous support from Intel and VMware

Carnegie Mellon University
TEXAS The University of Texas at Austin
UNIVERSITY OF TORONTO

What is an FPGA?

- If you asked as recently as 2015 . . . .

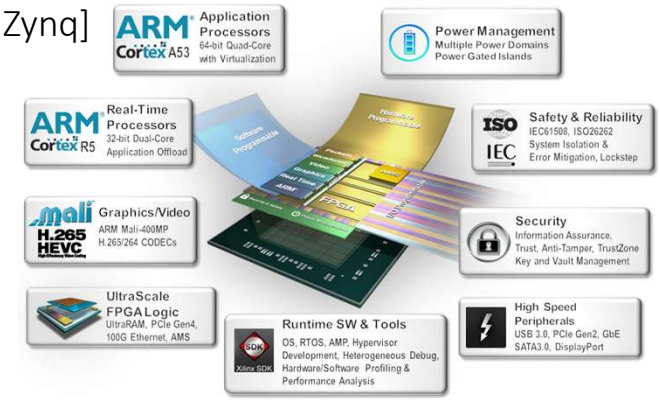




Screenshots from AMD/Xilinx and Intel/Altera's websites in 2015. Which is which?

Today's FPGAs not RTL targets

[Xilinx Zynq]

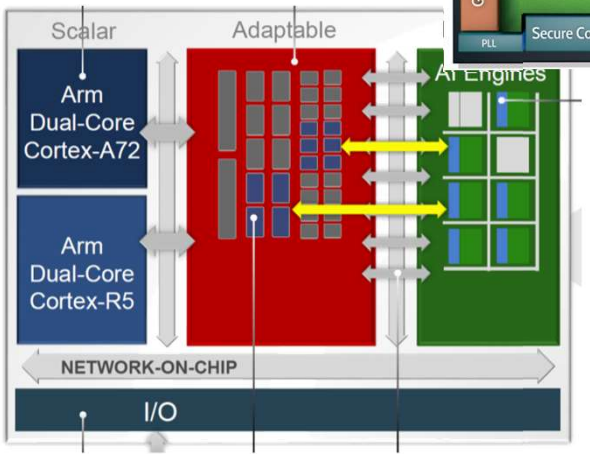


[Achronix Speedster]

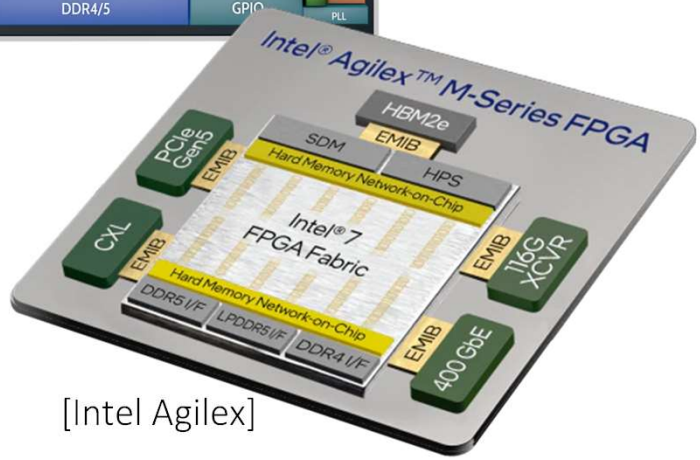
Heterogeneity

+

Adaptivity



[Xilinx Versal]



[Intel Agilex]

FPGA's Role in Heterogeneity

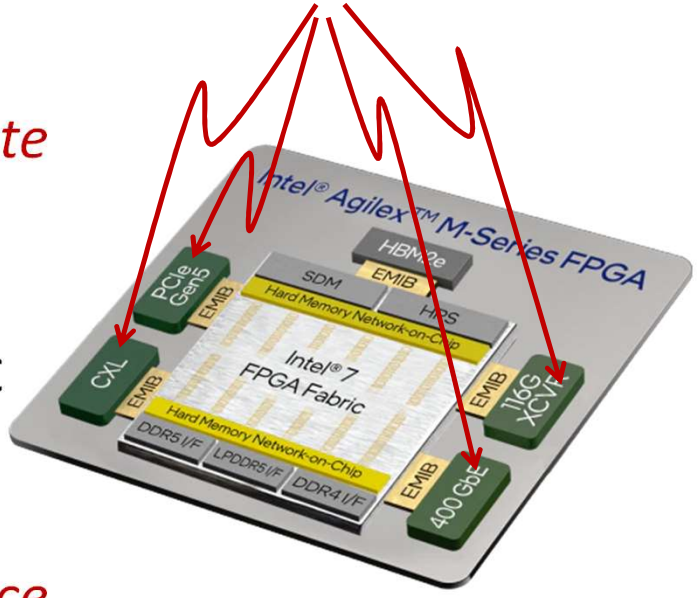
- Nothing beats ASICs in what it is designed to do
- **BUT,** ASICs horrible at what it is not designed to do
  - function and optimization locked at design time
  - limited applicability in the field
  - limited useful shelf life

ASIC designs deliberately de-tuned to compensate

- Programmable logic value proposition
  - allow ASIC parts to be common-case only
  - programmable upgrade/update around fixed ASIC functions
  - generality of compute, control, and interface

Programmable entity that augments and keeps pace

Intel EMIB: add your custom chips/chiplets here

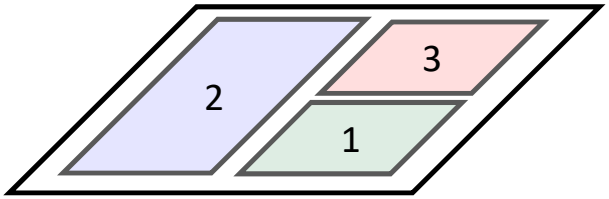


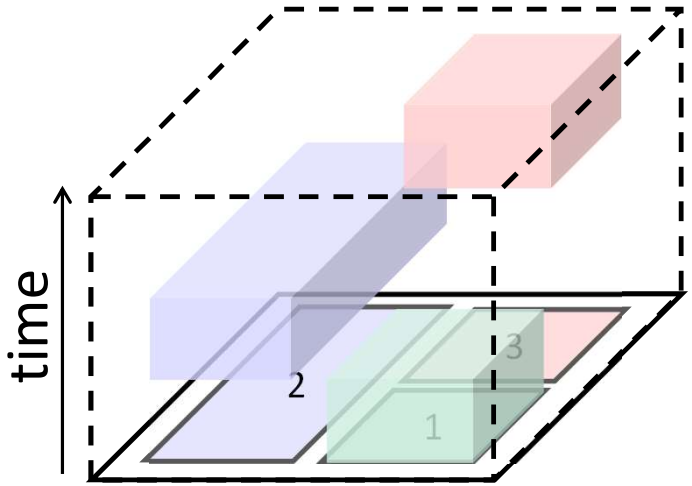
From "Field Programmable" to "Programmable"

- When FPGAs used as ASICs
  - field programmability avoided manufacturing NRE
  - **BUT** once power-on, FPGAs acted like fixed ASICs
- When FPGAs used for computing
  - "role-and-shell" partial reconfiguration allows repurposing
  - **BUT** roles individually still designed to act like fixed ASICs
- Truly use FPGA programmability to be more than ASICs?
  - multi-tenant dynamic allocation and sharing
  - dynamic reprogramming as design optimization

Static vs Dynamic Perspective to Logic Design

Basic questions in ASIC design:
(1) what to include, and
(2) how much "resource" to spend on each?

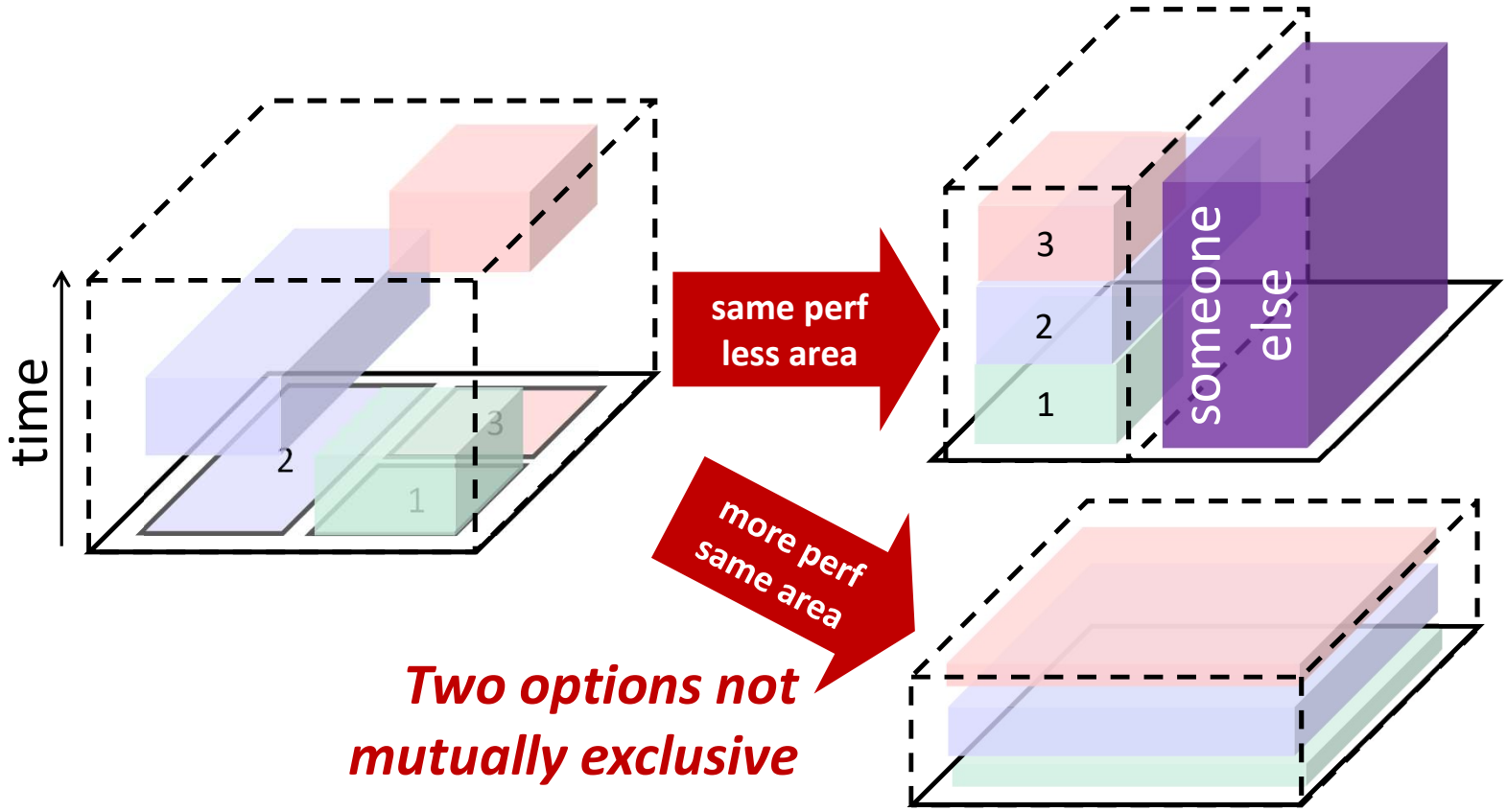




Fabric fully allocated in static thinking
$Area_{used} = A_1 + A_2 + A_3$, but . . .

Underutilization is possible in area-time volume
"Slack" = $Area_{used} \bullet t_{total} - (A_1 \bullet t_1 + A_2 \bullet t_2 + A_3 \bullet t_3)$

Reprogramming as Design Optimization



Two options not mutually exclusive

Outline

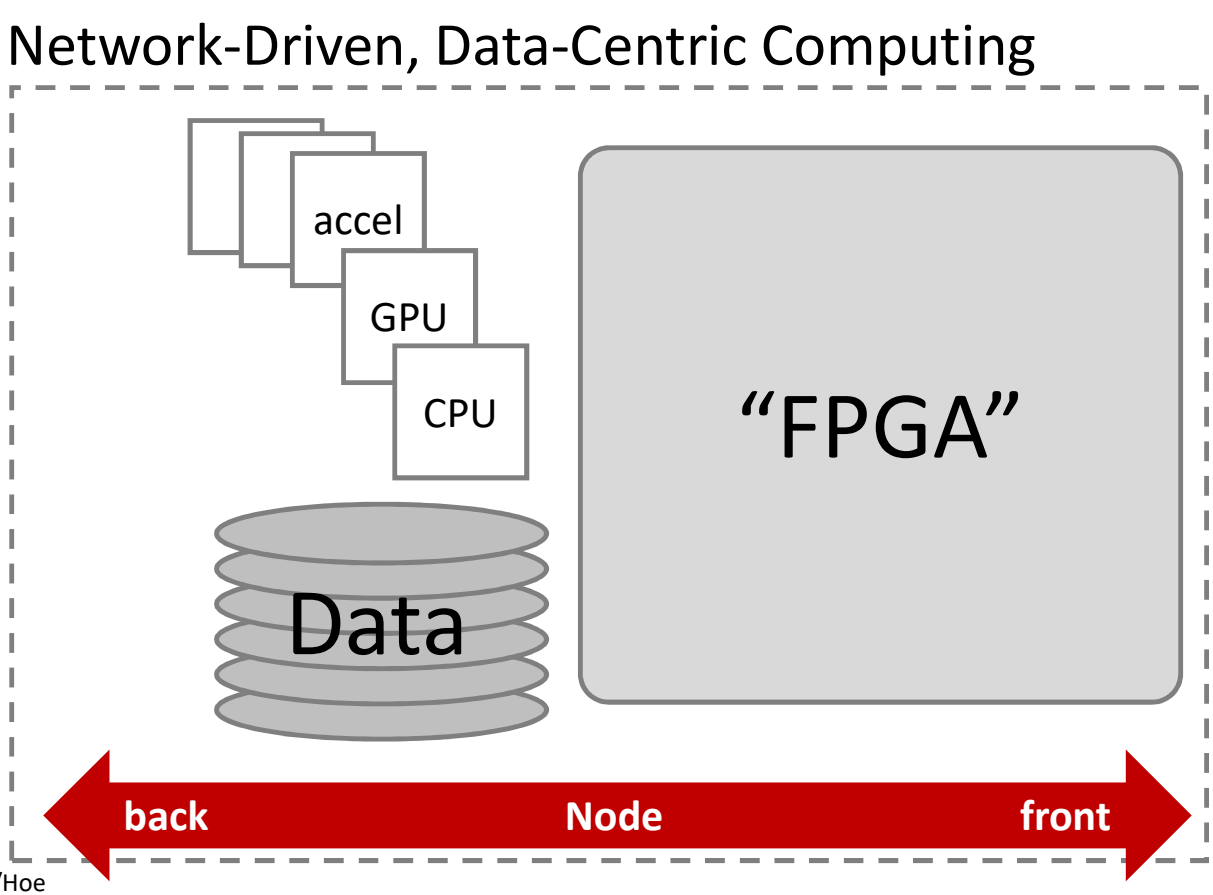
√ • 2023 FPGA and beyond?
  – distinct value proposition in a heterogenous system
  – programmability a feature not overhead

 • Crossroads 3D-FPGA for Datacenters

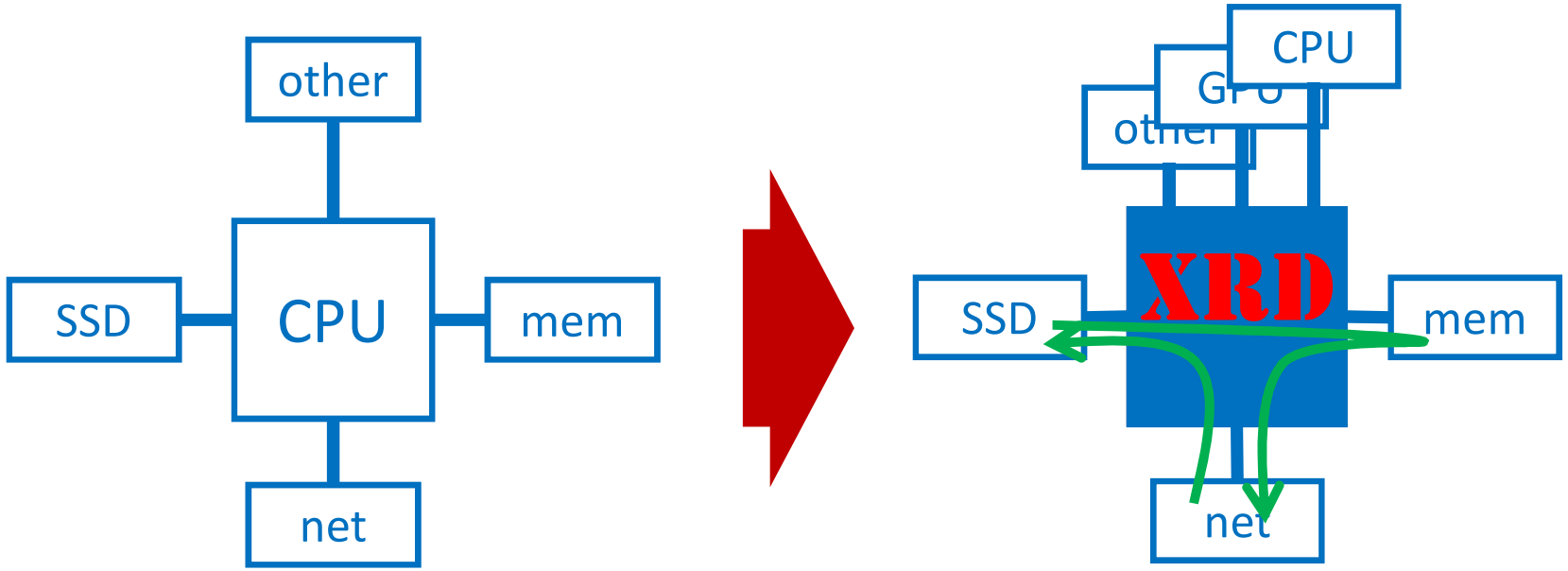
• Crossroads Research Center

What is the role for Programmable Logic in future datacenter servers?

- Network-Driven, Data-Centric Computing



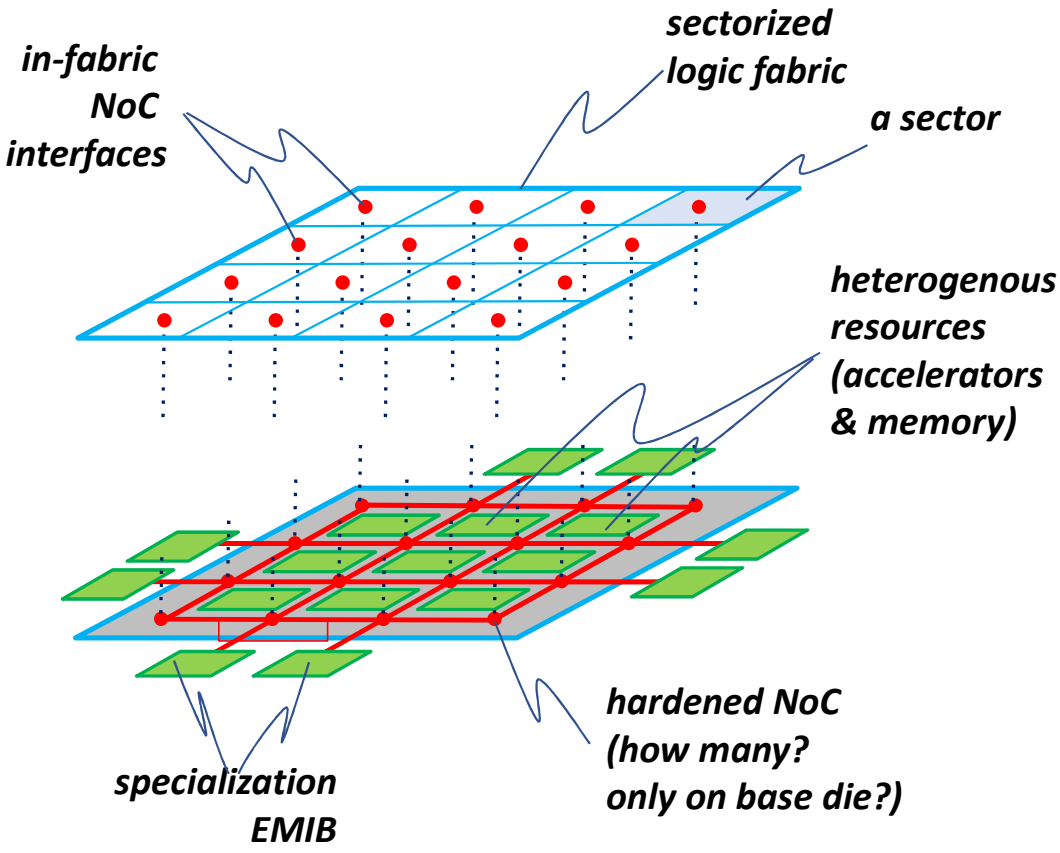
Crossroads FPGA: New Central Fixture



- Programmable active dataplane (switching and processing):
  1. data movement without CPU in the loop
  2. manipulate on-the-move data
  3. lend smarts to single-minded commodity HW

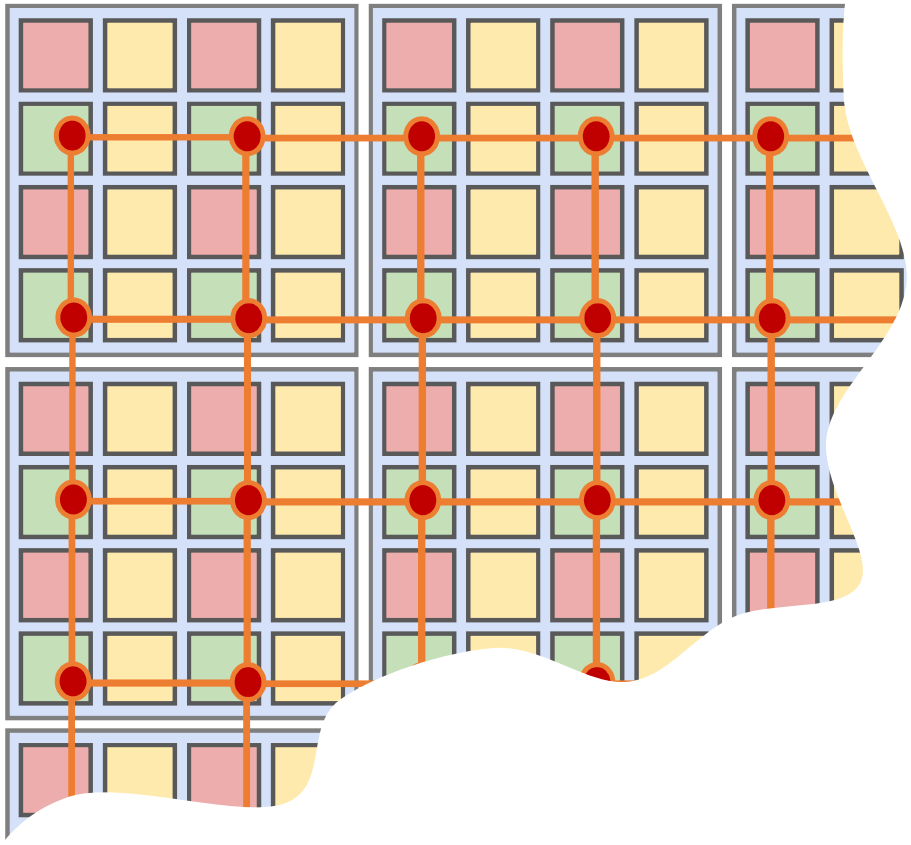
3D Hybrid Strategy

- General programmable logic fabric top die
  - retaining basic fabric and EDA know-how's
  - network-on-chip on-ramps
- Hardened specialized base die
  - datacenter IPs and NoC
  - single-minded primitives to hit perf/area/energy
  - leverage access to soft-logic for extensible specialization



Sectorized PR Logic Fabric with NoC

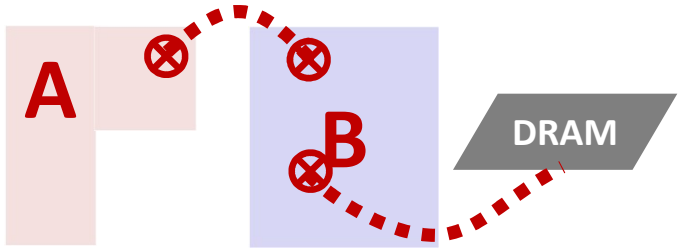
- Dynamic multi-tenant sharing by partial reconfiguration
  - "quantized" regular sectors for relocatable PR bitstream
  - whole sector as units of PR
  - *fabric remains seamless for multi-sector PR module*
- NoC for transport & virtualization
  - PR modules interact only through NoC
  - latency-insensitive message-passing or streaming

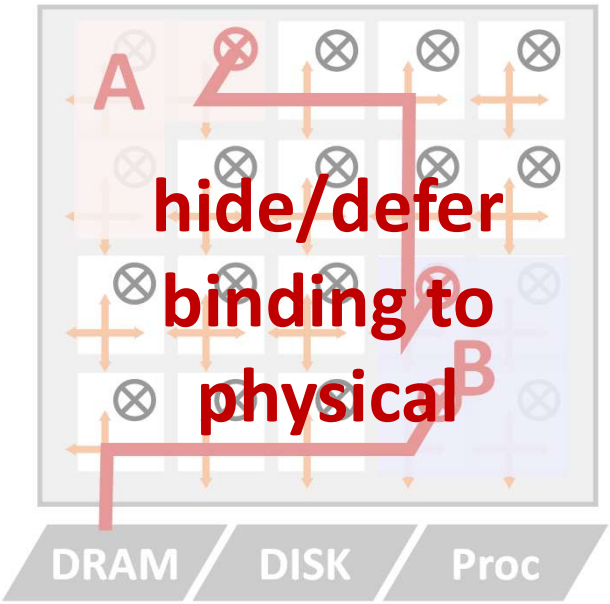


Think of managing sectors like managing DRAM pages

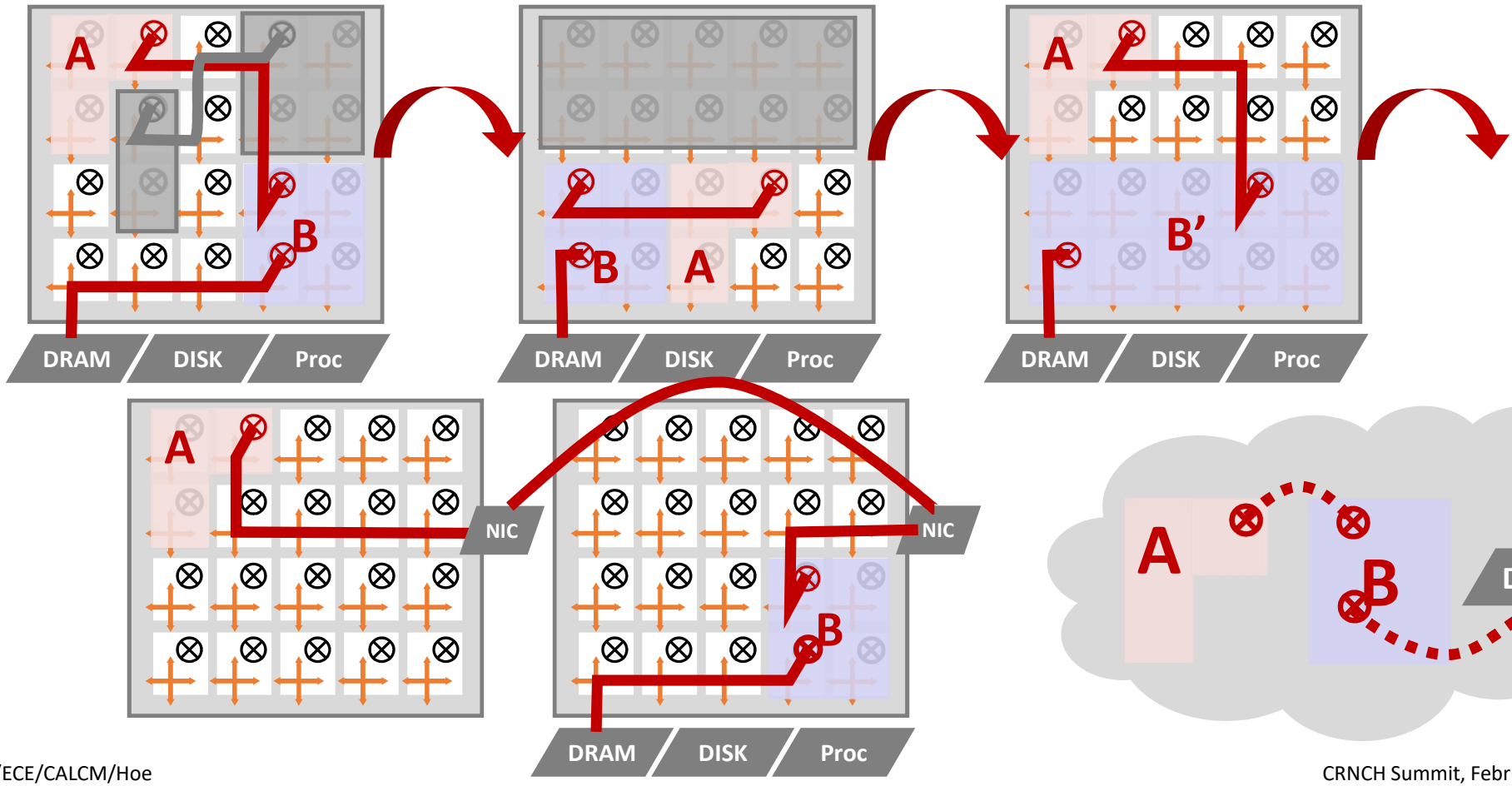
"Application Programming Interface"

- Application exists as one or more PR modules
  - user, library services, and infrastructure
  - each module exists in a whole-number tile context
  - interact through latency-insensitive, messages and streams
- Restricted global knowledge
  - defer binding decision
  - Compiler and runtime optimizations





Putting it Together:
Dynamic, Sharable, and Managed



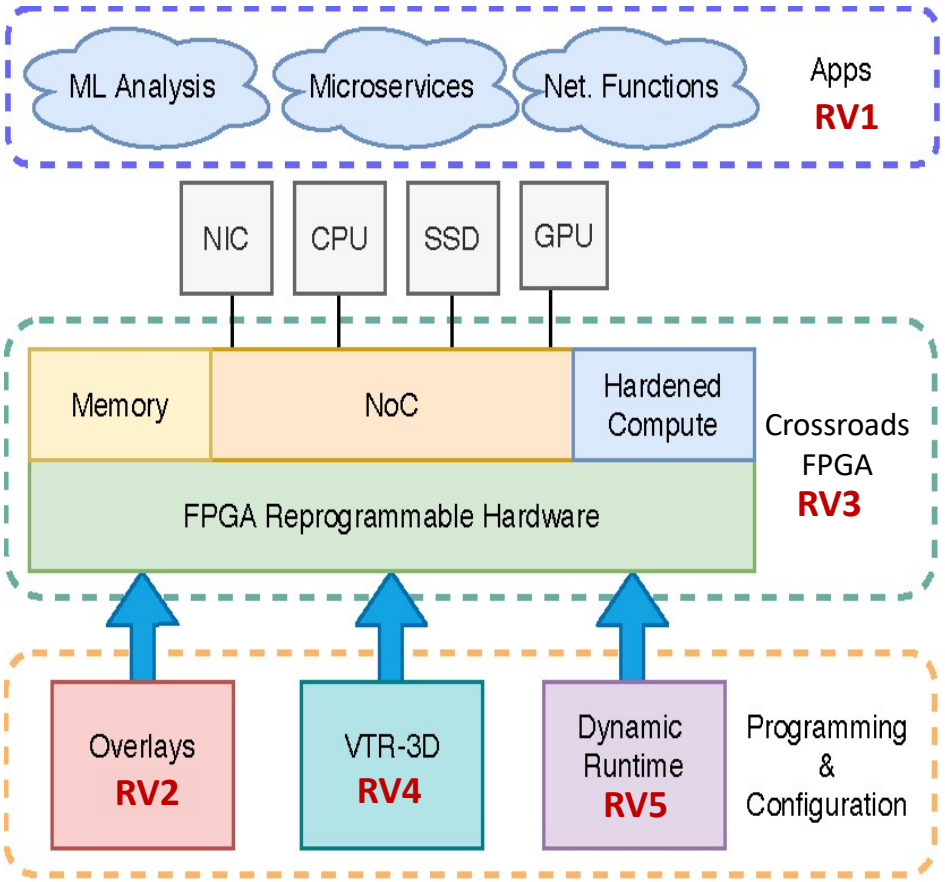
Outline

√ • 2023 FPGA and beyond?

√ • Crossroads 3D-FPGA for Datacenters
  – bit/cycle-granularity, data-on-the-move acceleration
  – programmable, dynamic, and adaptive processing resource
  – purposeful heterogenous architecture

 • Crossroads Research Center

Crossroads Research Vectors



What purposes served? What values added?

What capabilities needed? How to realize them?

How to support user in exercising programmability and dynamism?

Crossroads PIs

www.crossroadsfpga.org

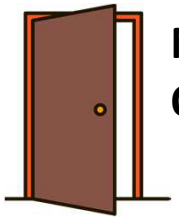
						
Betz	Chiou	Franchetti	Hoe	Pan	Sekar	Sherry
(U Toronto)	(UT Austin)	(CMU)	(CMU)	(UT Austin)	(CMU)	(CMU)
RV1/RV3/RV4	RV2/RV3	RV2	RV1/RV3/RV5	RV4	RV1	RV1
FPGA, EDA, VLSI	**architecture, FPGA, data center**	**compiler, algorithm, performance**	**architecture, reconfig computing, acceleration**	**EDA, VLSI, ML**	**networking, security, systems**	**networking, middle box, cloud**

RV1 Applications — Data-on-the-Move

- "Data on the Move" Apps:
  - high traffic I/O from NIC to CPU to disk to accelerators
  - processing performed on data "in flight" between platforms
  - standardized "boilerplate", and parallelizable code
- Driving examples:

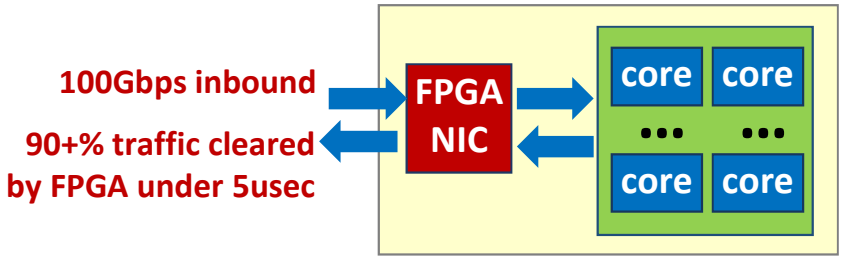
Pigasus Intrusion Detection

HPIPE CNN Inference

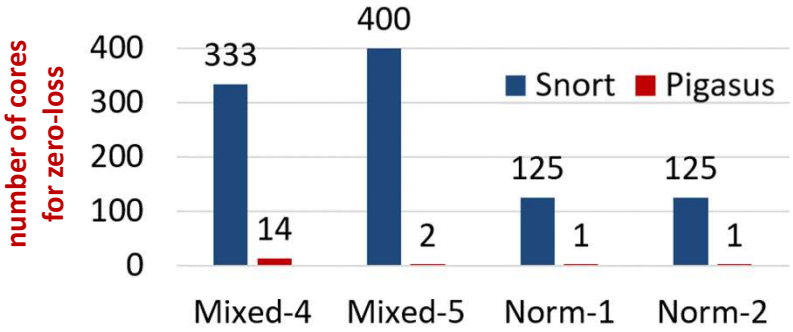


Norman OS

- **Showcase:** Pigasus 100Gbps IDS [CMU]
  - FPGA-first hybrid architecture



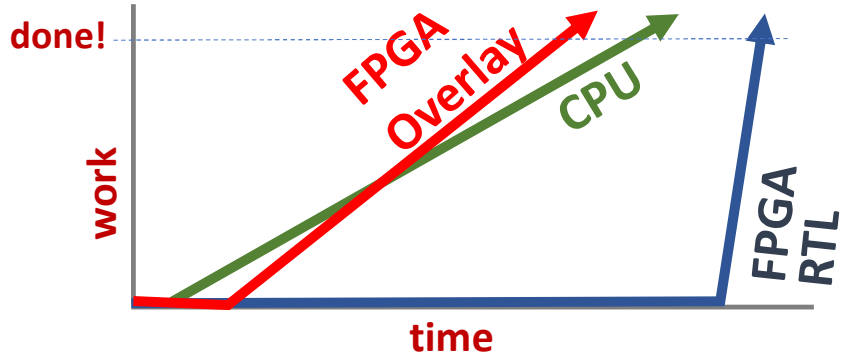
  - FPGA saves 100x cores from SW-only



github.com/crossroadsfpga/pigasus

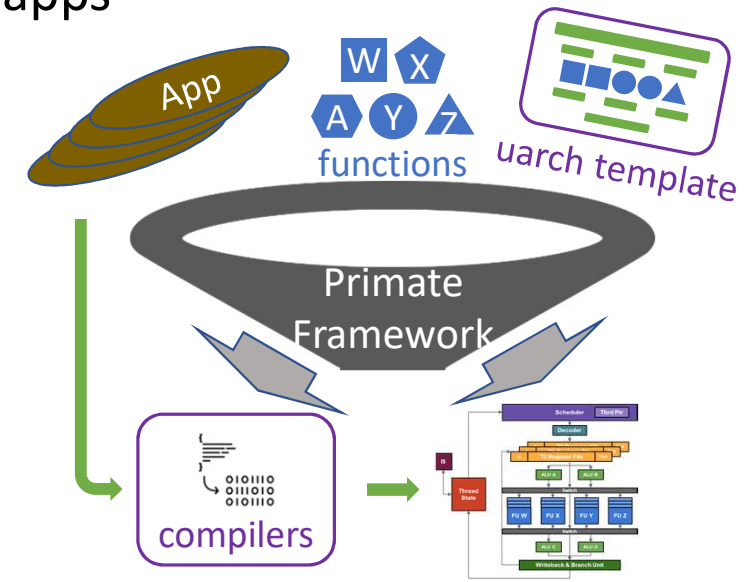
RV2 Overlays — Time-to-Solution

- Time-to-solution (dev/debug/deploy) important in datacenters
- FPGA fast to rollout compared to ASICs, but slow compared to CPU



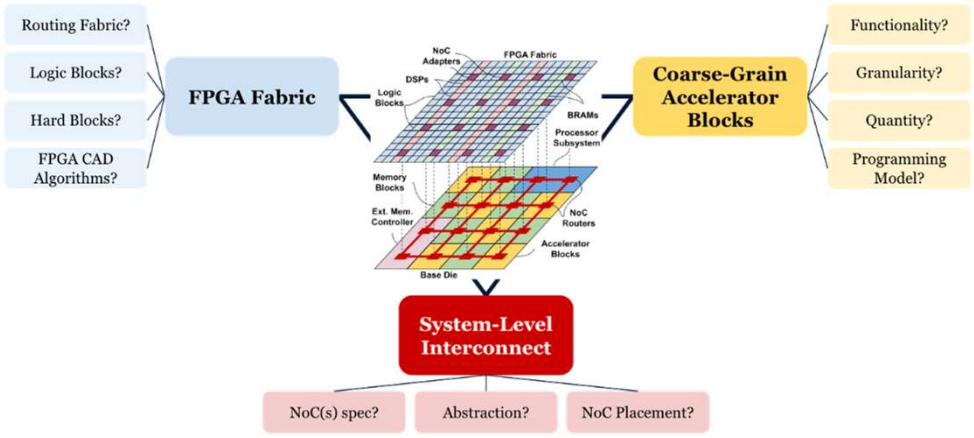
- SW programmable overlay
  - custom generated efficient datapath
  - rapid development/debug/deploy

- **Showcase:** Primate [UT Austin]
  - generate overlay processor tuned to core C++ app set
  - rapidly compile and deploy new apps



RV3 Architecture —
Heterogenous, Adaptive, Systems

- How to architect Crossroads?
  - large new design space
  - co-design with application
  - end-to-end quantitative evaluation: performance + silicon



- **Showcase:** RAD-Sim [U of Toronto]
  - specify RAD arch parameters (NoC specs, module freq, etc.)
  - prototype application modules in SystemC and NoC connections
  - cycle-accurate simulation
  - tweak **both** application and architecture ⇒ repeat
- **Also:** RAD-Gen to estimate silicon speed and cost

Boutros, et al. IEEE Access, 2022
github.com/andrewboutros/rad-flow

RV4 FPGA CAD — 3D and NoC-Aware

- Pathfind EDA approaches for novel Crossroads features: 3D, NoC, partial reconfiguration

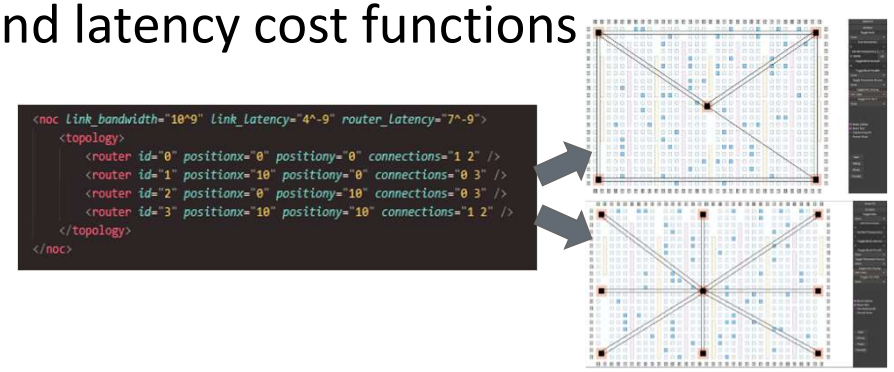
  You cannot use what EDA cannot target

- Map real designs onto *virtual* Crossroads FPGA fabric for research

  You cannot improve what you cannot measure

- Enable community with a flexible, scalable and re-usable EDA platform (verilogtorouting.org)

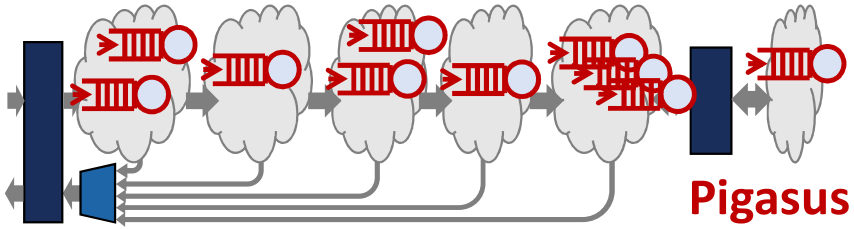
- **Showcase:** NoC driven placement in VPR [U of Toronto]
  - extend VTR's architecture description with NoC router
  - analyze application traffic flow over logical routers
  - place app module and virtual routers to minimize bandwidth and latency cost functions



RV5 Dynamic Reconfiguration —
Field Programmable to Programmable

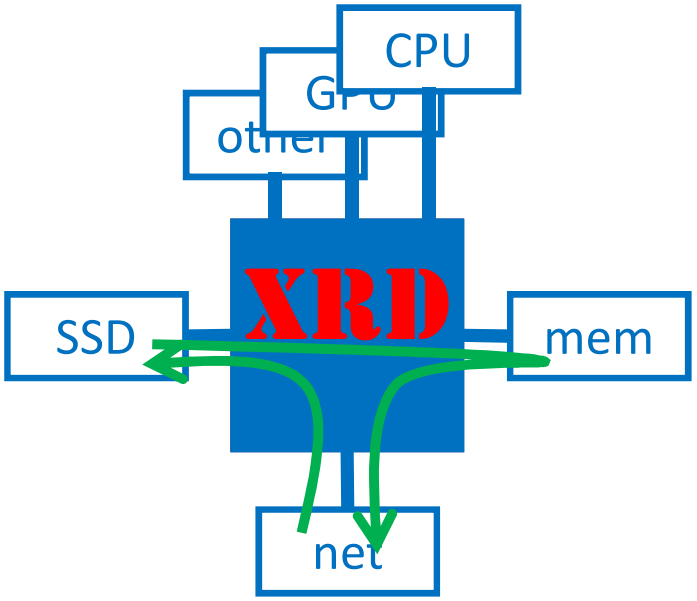
- PR as basic mode of operation
- New design and usage mindset
  - what is instantiated on fabric can change
  - resources one don't use, other can
- Research needs:
  - programming abstraction and tools
  - virtualized execution environment
  - needed hardware support

- **Showcase:** FPGA as a System [CMU]
  - new applications
    - irregular and data-dependent
    - dynamically varying tuning
    - service level performance objectives



  - systems methodology for FPGA design
    - services and buffers abstraction
    - queuing and stochastic modeling
    - SLO and TCO optimization metrics

Crossroads Mission Sum Up



www.crossroadsfpga.org

- Data-on-the-move acceleration as a new role for programmable logic in future data-centric computing
  - new fixture device architected to purpose
  - programmable logic as distinct dimension to HW vs SW heterogeneity
- Commitment to public dissemination and technology transfer
  - opensource and community building
  - Intel and VMware collaborations

Check out our seminar, GitHub and YouTube