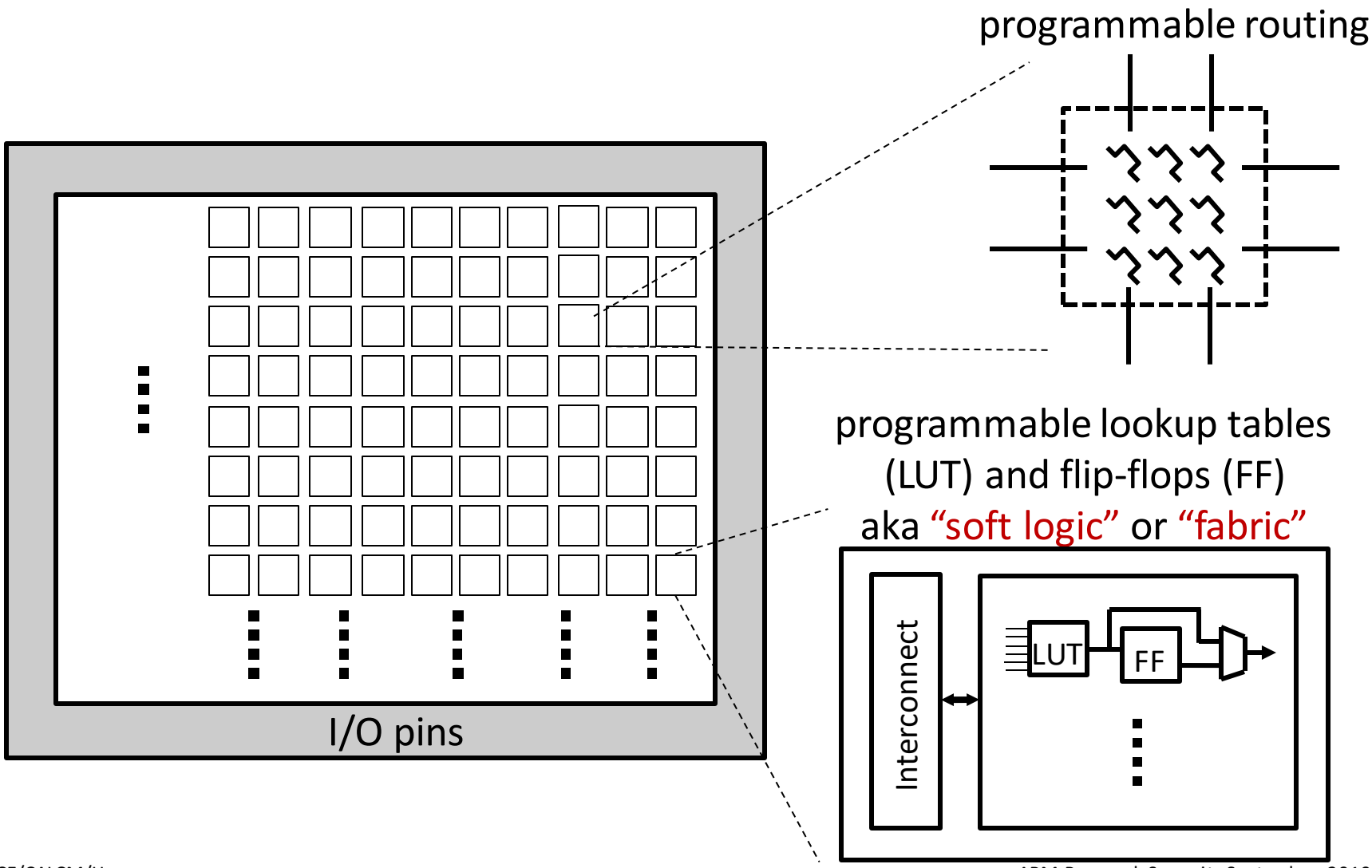
From "Field-Programmable" to "Programmable"

James C. Hoe
Department of ECE
Carnegie Mellon University

Classic FPGA in a Nutshell



FPGAs as we knew it

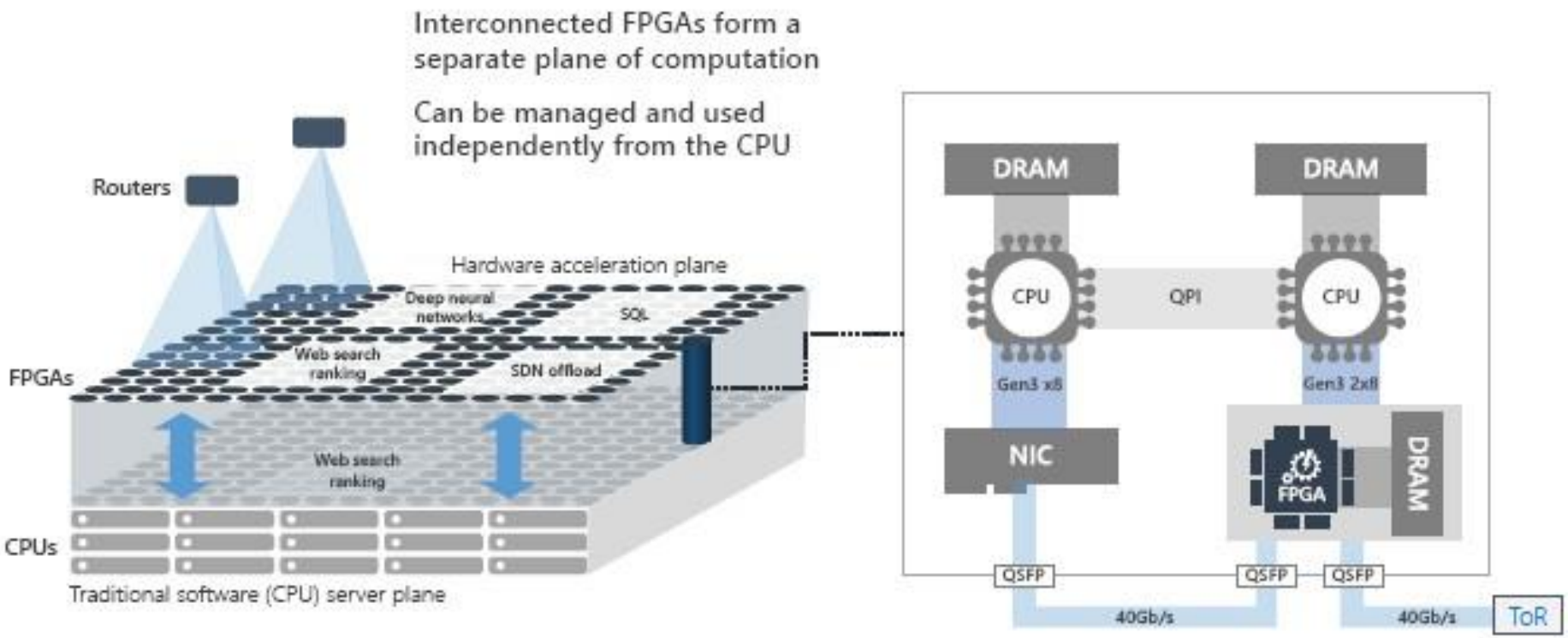
Traditionally, FPGAs have been the bastard step-brother of ASICs. They have been forced to act like ASICs and fit themselves into the ASIC development model. . . . . .

. . . . . This has meant ignoring their unique strengths: reprogrammability, late binding and run-time reconfiguration.

Andre DeHon, ISFPGA 2004

Perspective on FPGAs changed when

- Microsoft (and others) got desperate enough to do this



[www.microsoft.com/en-us/research/project/project-catapult]

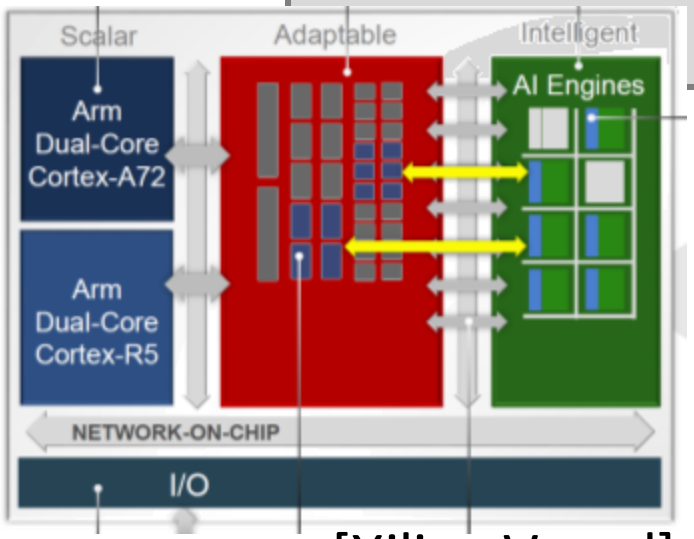
New FPGAs are not RTL targets

"*"

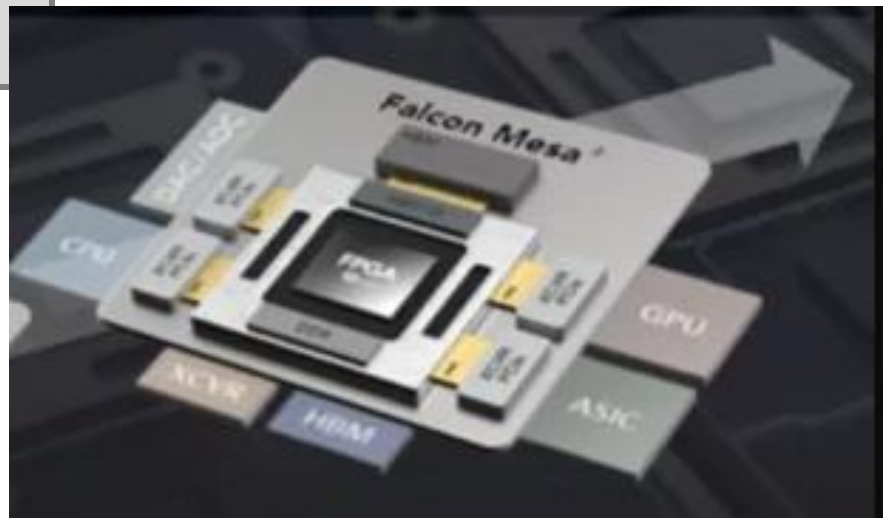
- spatial data/compute
  - highly concurrent
  - finely controllable
  - reprogrammable

Immediate Challenges

- ***killer apps***
- ***ease of development***



[Xilinx Versal]



[Intel Agilex]

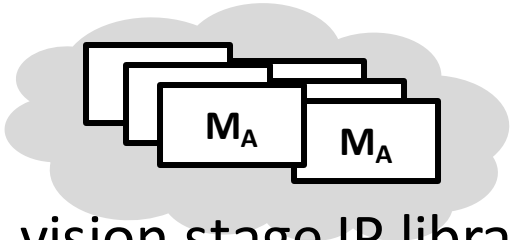
Greater break from ASIC mentality

- **Dynamism** — actually use the programmability
  - support more functionality on same parts cost
  - achieve better performance by specializing
- **Shareability** — multitenancy to consume "slack"
  - too much logic: partition fabric spatially
  - too much throughput: repurpose fabric temporally
- **Manageability** — bring FPGA under OS purview
  - part of compute resource pool (CPU cycles, DRAM)
  - seamless interface, virtualization and isolation (security and QoS)

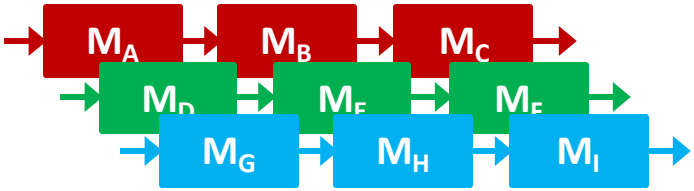
Dynamic Partial Reconfiguration is a key capability

DPR: what is feasible today?

Dynamic Execution Framework for Interactive Vision

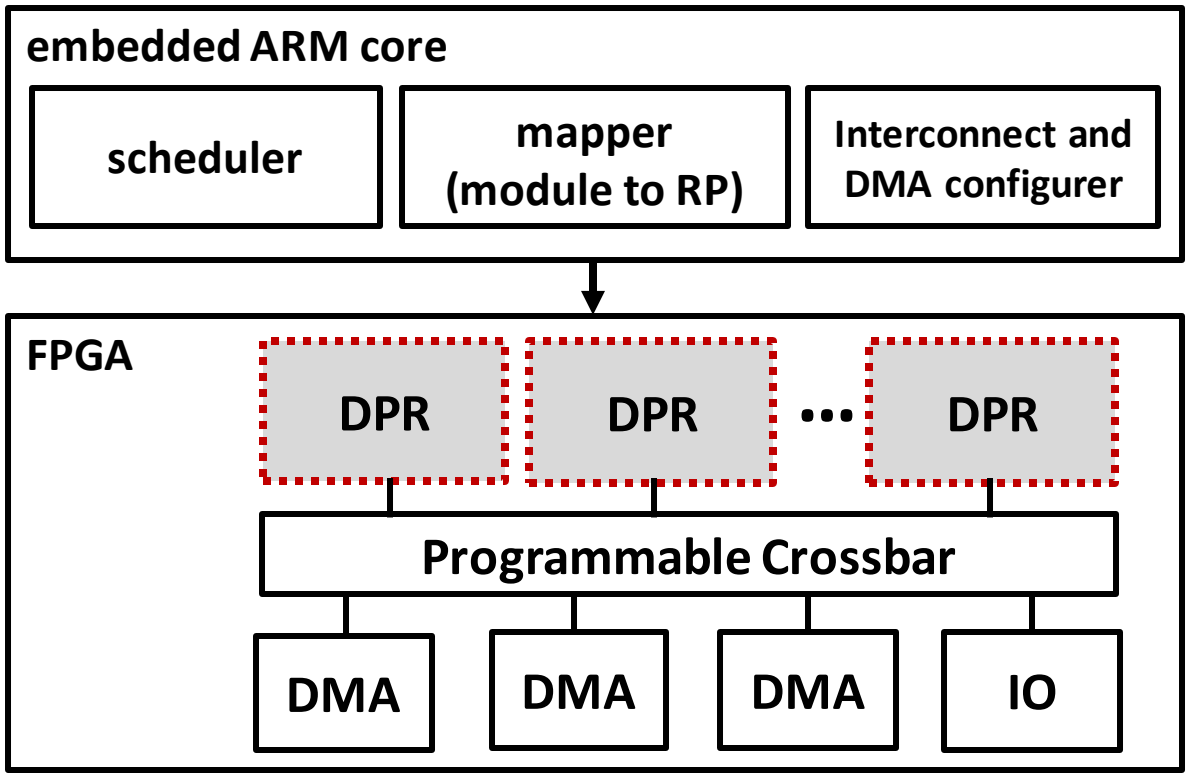


vision stage IP library + pipeline specifications



runtime

plug-and-play architecture



Spatial and Temporal Multitenancy

Camera → M_A → M_B → M_C → Display

Camera → M_D → M_E → M_F → Display

Camera → M_G → M_H → M_I → Display

PS

Flash

DRAM

ARM core
(user code + SW management)

AXI-PCAP Bridge

AXI Master Interface

FIFO

PCAP Interface

FPGA

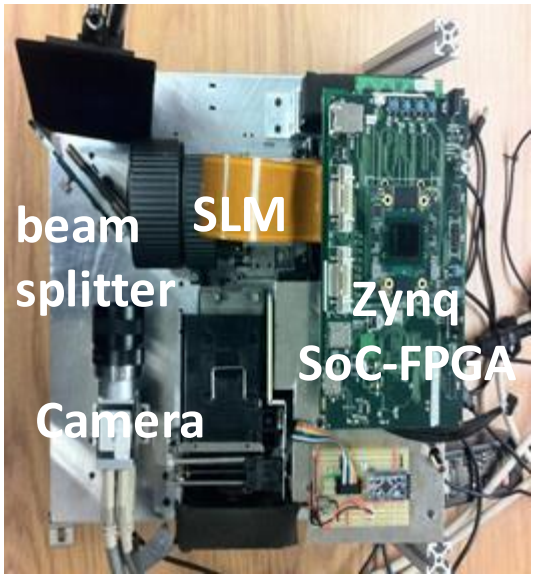
M_A M_B M_C M_G M_H M_I

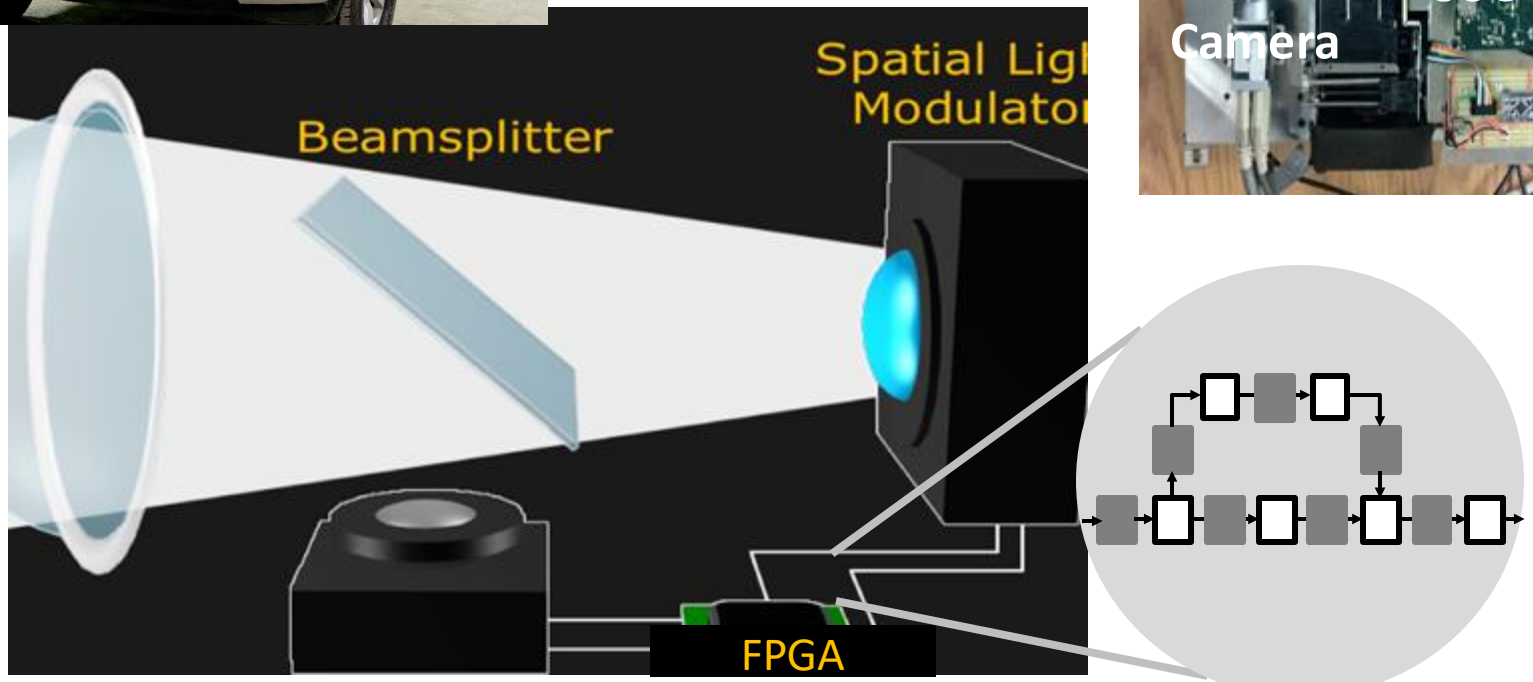
Overlay Crossbar

DMA DMA DMA DMA DMA Camera Display

We Actually Use This

https://www.cs.cmu.edu/smartheadlight



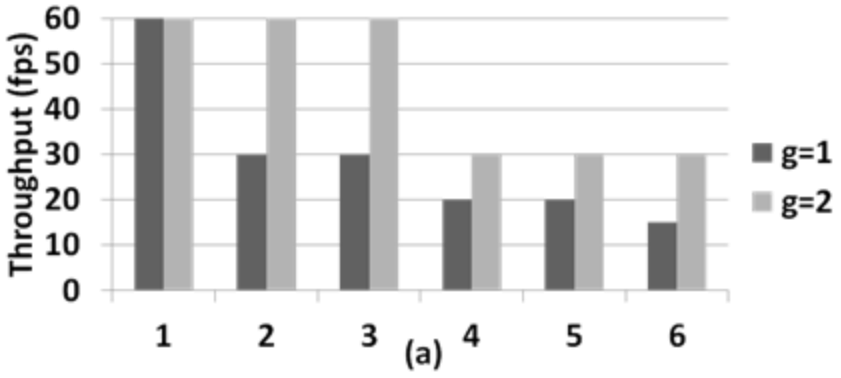


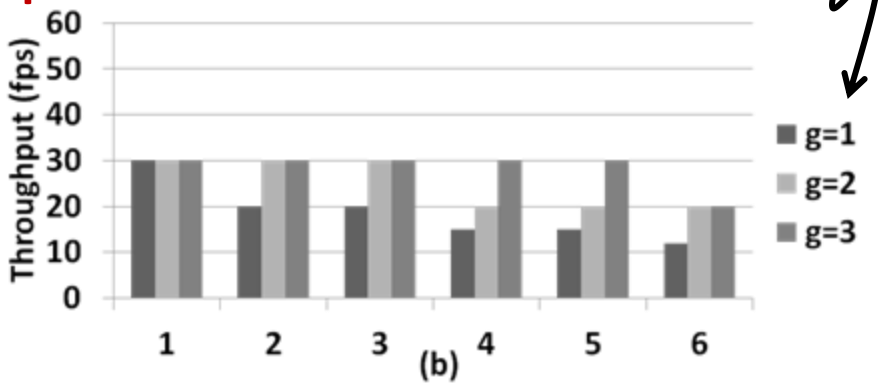
Time-Multiplexing Feasibility [FPL2018]

- Interleaving (a) 2 pipelines and (b) 3 pipelines
- Pipelines differ between 1 and 6 PR partitions

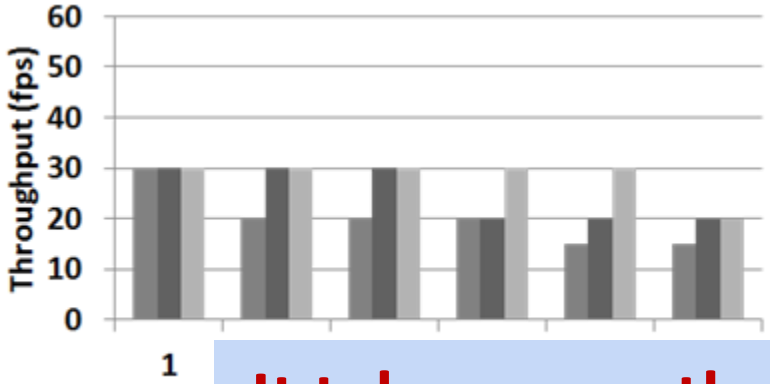
batch size

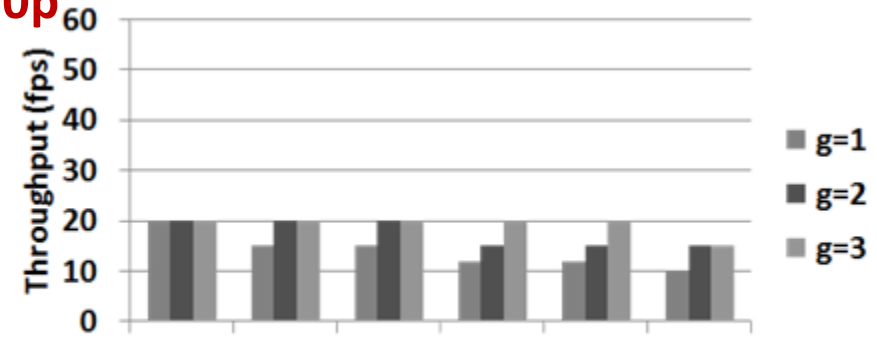
720p





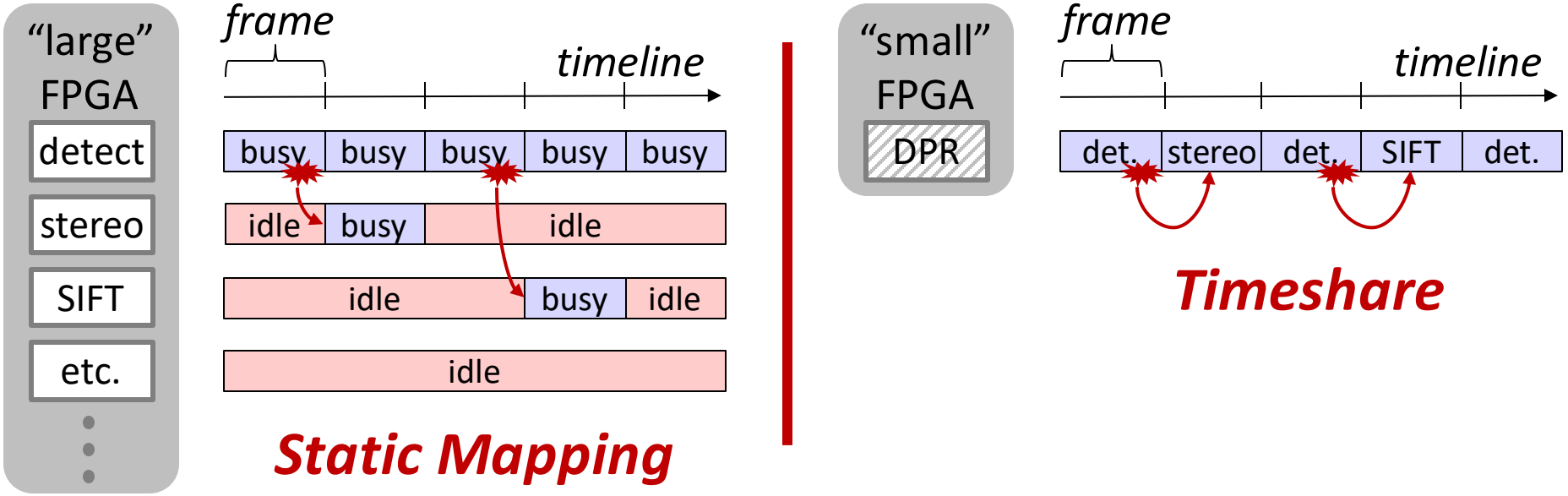
1080p





It takes more than doing PR in quick successions!!!

Cost and Energy/Power Benefits [FPL2019]



- Application casestudy with 6 modules
  - color-based detector triggers follow-up processing
  - meets throughput and latency requirements
  - ~**3x** logic saving (7x **$** saving in parts cost)
  - ~**30%** power/energy saving in worstcase

Today's Practical Constraints

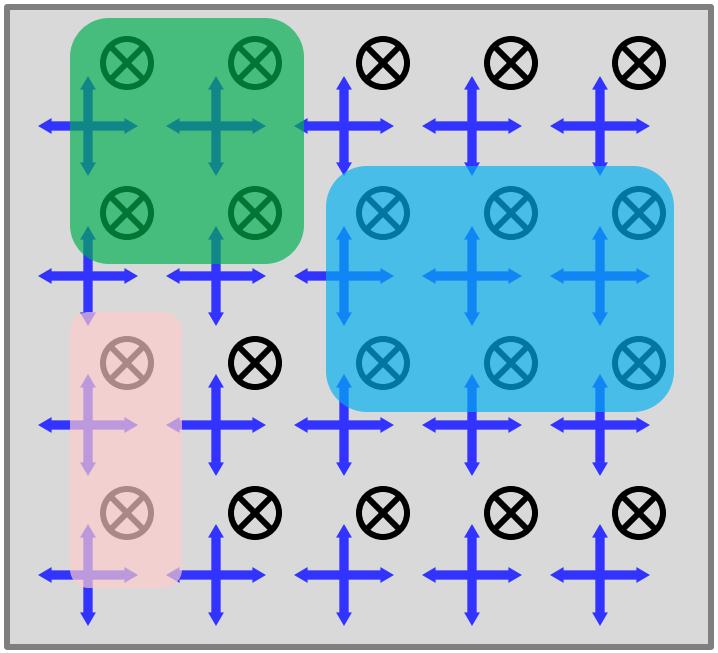
- Number and size of PR partitions fixed apriori
  - too few/too large: internal fragmentation
  - too many/too small: external fragmentation
- Not all PR partitions are equal—even if same interface and shape
  - a module needs a different bitstream for each partition it goes into
  - build and store upto MxN bitstreams for N partitions and M modules
- PR is not all that fast . . .

PR could be a lot better if need be

- Reconfiguration time could be a lot faster
  - increase raw speeds
  - increase concurrency
- Tools could be more powerful and friendly
  - more push-button GUI, less manual scripting
  - higher level interfaces (incorporating management and scheduling) in-line with use models (e.g., AOC)
- More flexibility, e.g. soft partition boundaries, relocatable bitstreams
- Time-multiplex contextful modules?

Hopes and Dreams: Spatial/Temporal Multitasking Fabric?

- Manage and schedule FPGA fabric like one would with CPU cycles and memory
- First-class support of PR compute modules
  - hard transport with standard virtually, private interfaces to memory, I/O & resources
  - loadable modules designed independently of placement and surroundings
  - security and QoS provisions for multitenant sharing



Sponsored by the CONIX Research Center, one of six centers administered by the JUMP phase of the Focused Center Research Program (FCRP), a Semiconductor Research Corporation program sponsored by MARCO and DARPA.