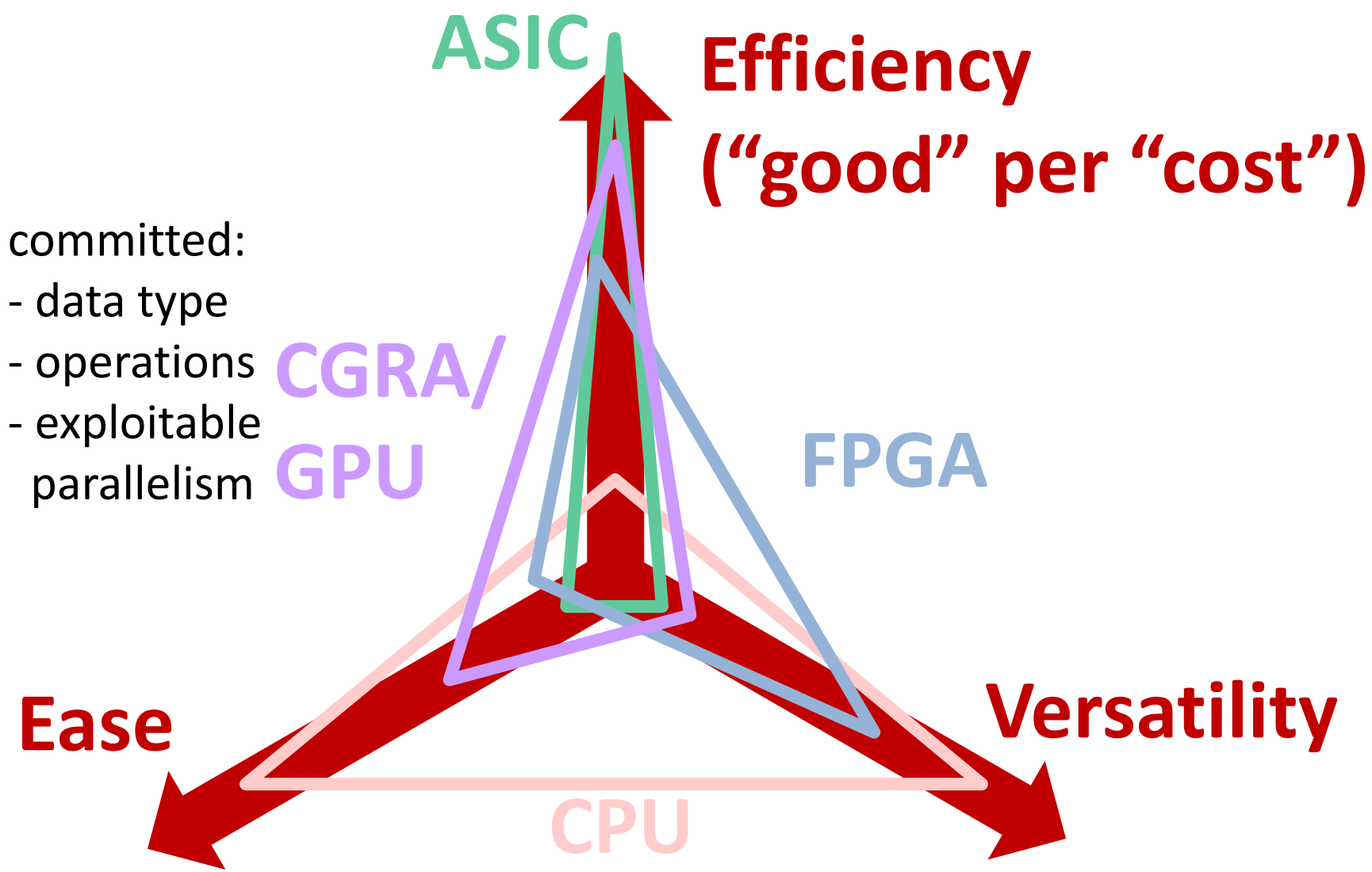
18-643 Lecture 4: FPGAs with Purpose

James C. Hoe
Department of ECE
Carnegie Mellon University

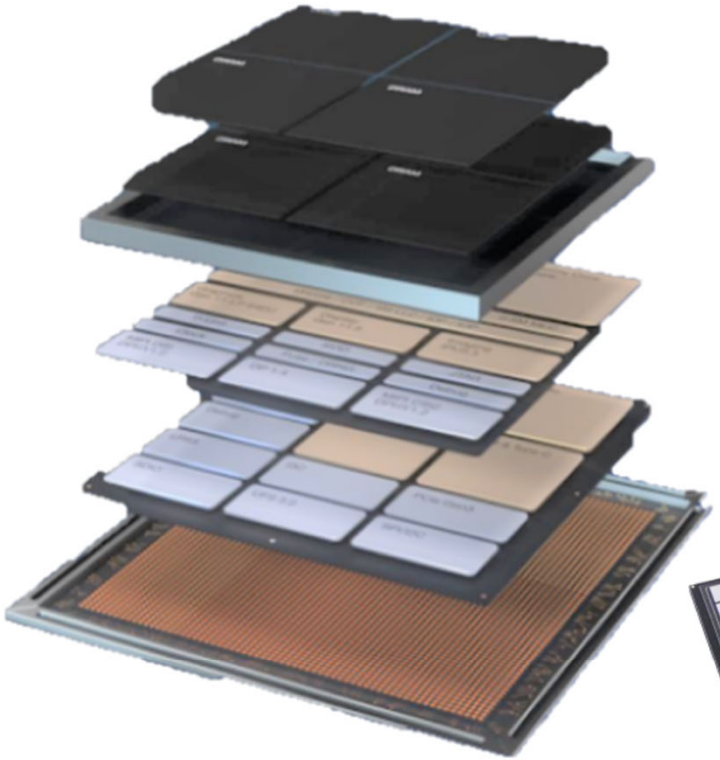
Housekeeping

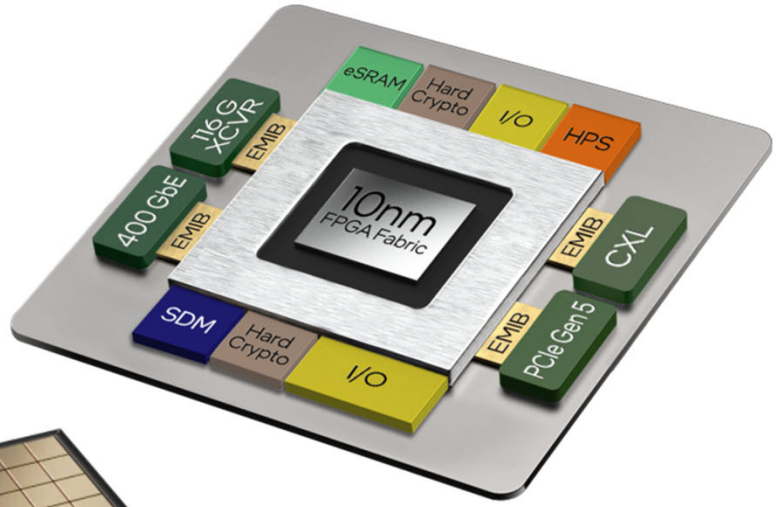
- Your goal today: appreciate modern "FPGAs" as heterogeneous and purposefully architected
- Notices
  - Lab 0, due noon, Mon 9/7
  - Complete survey on Canvas, past due
  - Ultra96 pick up in HH-1301, 9am~5pm, next week
  - Recitation starts next week; watch for time poll
- Readings (see lecture schedule online)
  - Skim [Chromczak20] and [Ahmed16]
  - Skim [Caulfield16]

Differing Tradeoff and Sweetspots



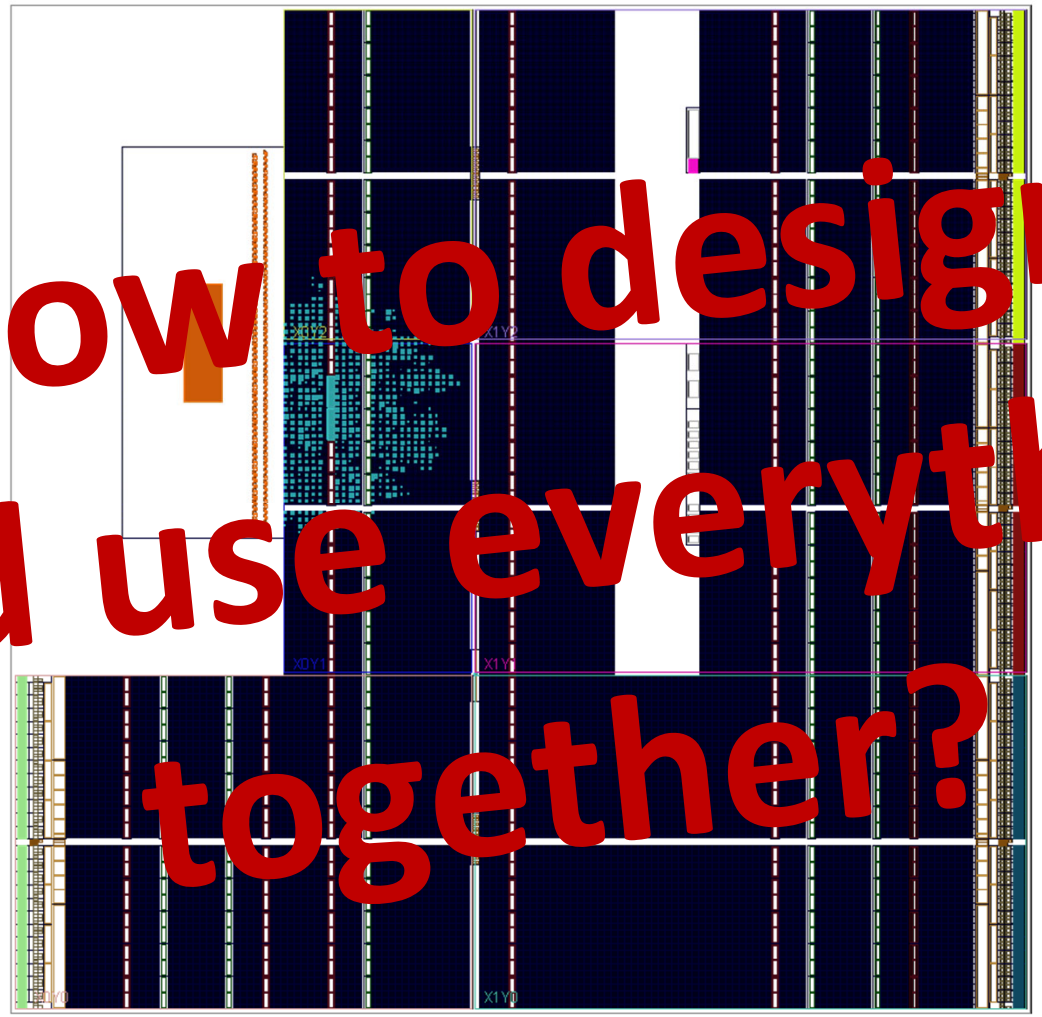
Energy/Power Bound ⇒ Heterogeneity





Wrong to think ASIC "most" efficient!!

Die Area "Return on Investment"



How to design and use everything together?

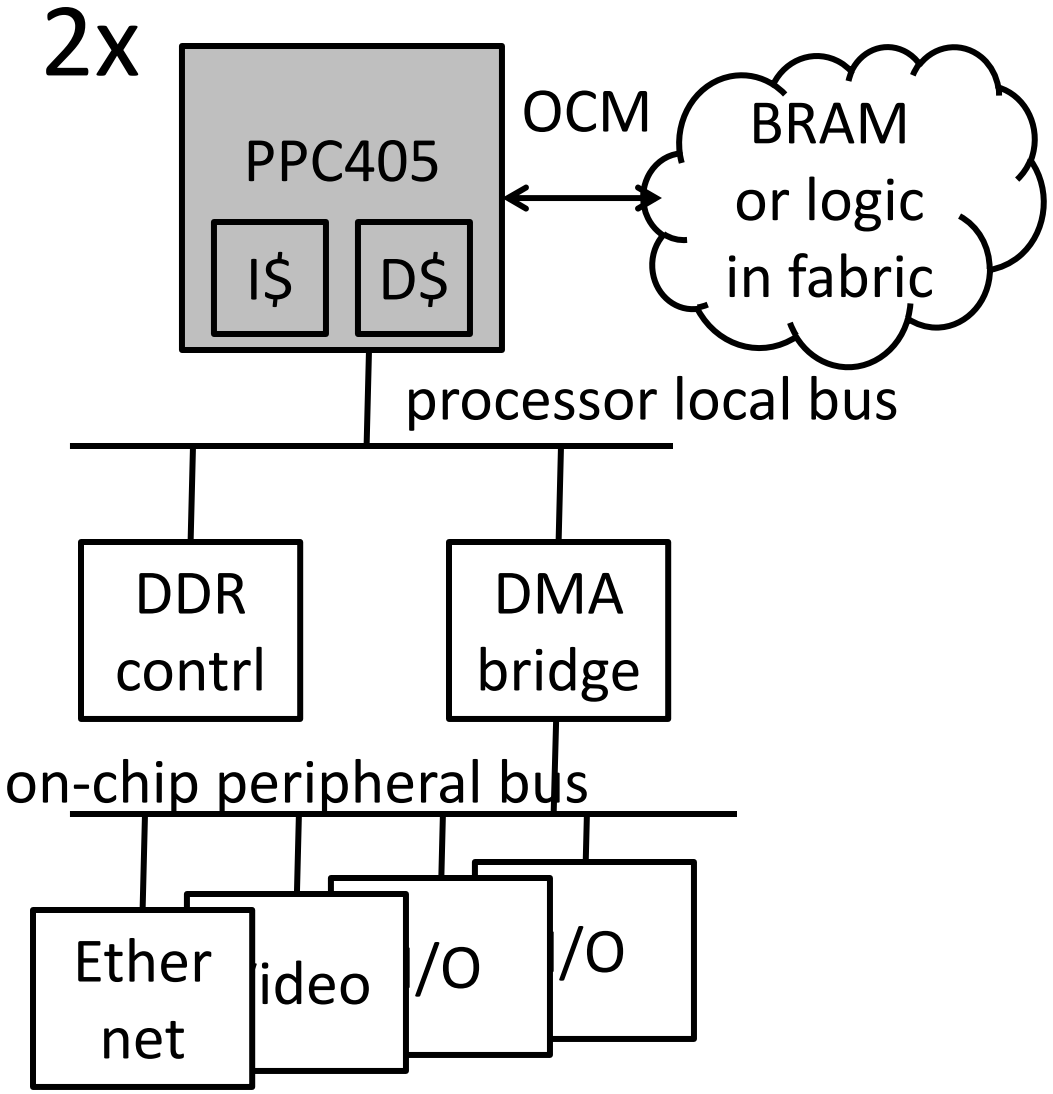
[Vivado screenshot XC7 2020]

Soft-logic dominates die area, but compute/storage concentrated in DSP and BRAM—consider what if 100% soft or 100% hard

Recall

2010: Xilinx Zynq SoC FPGA

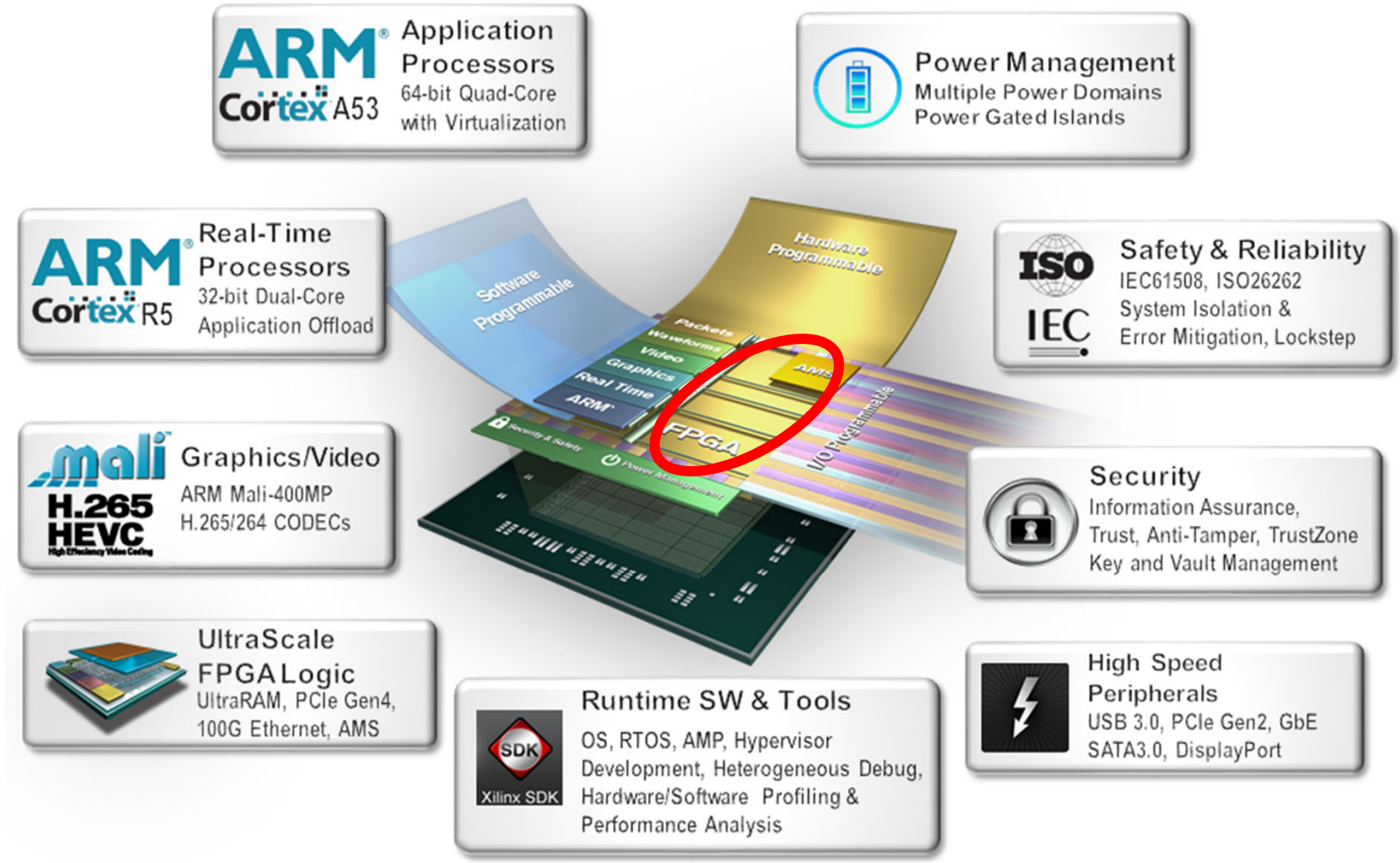
Precursor: Virtex II Pro and "EDK"



- everything else is soft
- two hierarchies of soft-logic busses
- special on-chip memory (OCM) port allows ld/st directly into fabric
- CoreGen Library of IPs to hang off the busses

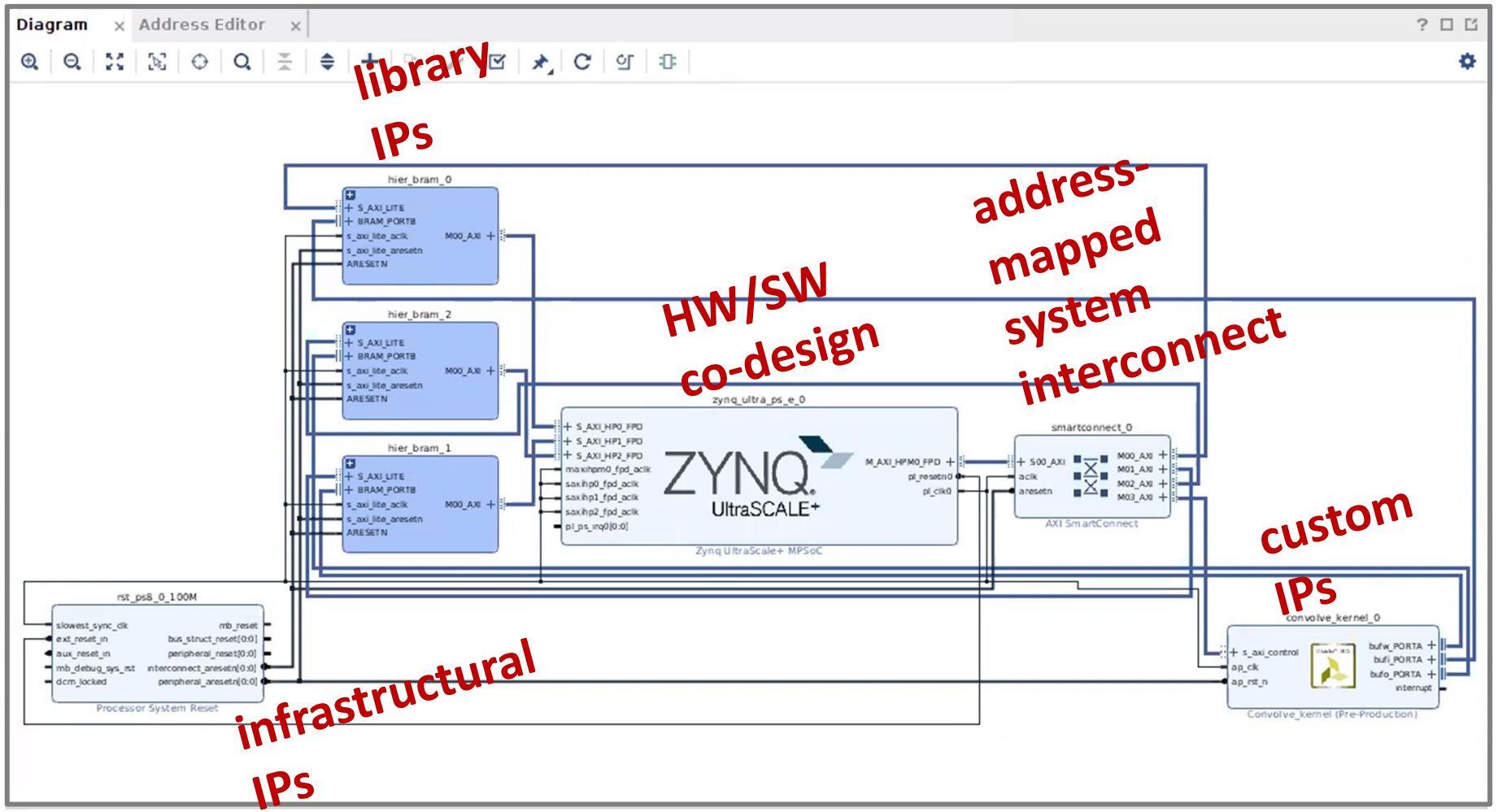
[Xilinx Virtex II, early 2000]

Xilinx Zynq SoC FPGA



[http://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html]

Zynq SoC-FPGA Designer Mindset



Vivado IP Integrator Screenshot

Theme 1: HW/SW Co-Design

- An application is partitioned for mapping to
  - HW: everything SW is not good enough for
  - SW: everything else + control
- SW is the heart and soul
  - in control of HW
  - enables product differentiation
- SW can be harder than HW **(Is this surprising?)**
  - embodies most of the complexity
  - often dominates actual development time/effort

Theme 2: IP-Based Design

- Complexity wall
  - designer productivity grows slower than Moore's Law on logic capacity
  - diminishing returns on scaling design team size

⇒ must stop designing individual gates

- Decompose design as a connection of IPs
  - each IP fits in a manageable design complexity

Bonus, IPs can be reused across projects

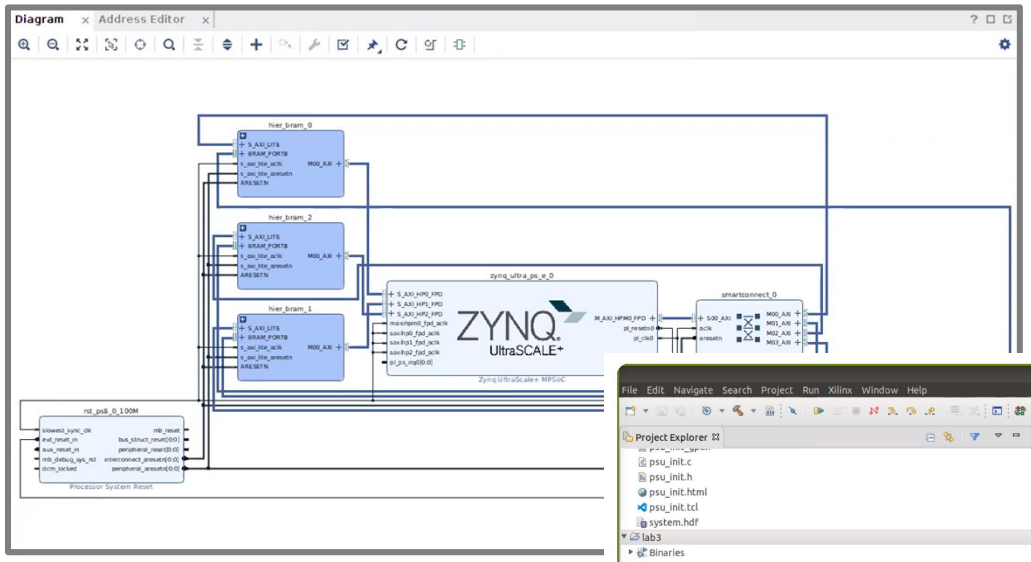
——————abstraction boundary——————

  - IP integration fits in a manageable design complexity

Theme 3: Systematic Interconnect

- More IPs, more elaborate IPs $\Rightarrow$ intractable to design wires at bit- and cycle-granularity
- On-chip interconnect standards (e.g. AXI) with ***address-mapped*** abstraction
  - each ***target*** IP assigned an ***address*** range
  - ***initiator*** IPs issue ***read*** (or ***write***) transactions to pull (or push) data from (or to) addressed target IP
  - even easier are IPs with streaming interfaces
- Plug&play integration of interface-compliant IPs
- Physical realization abstracted from IPs, e.g., network-on-chip ("route data not wires")

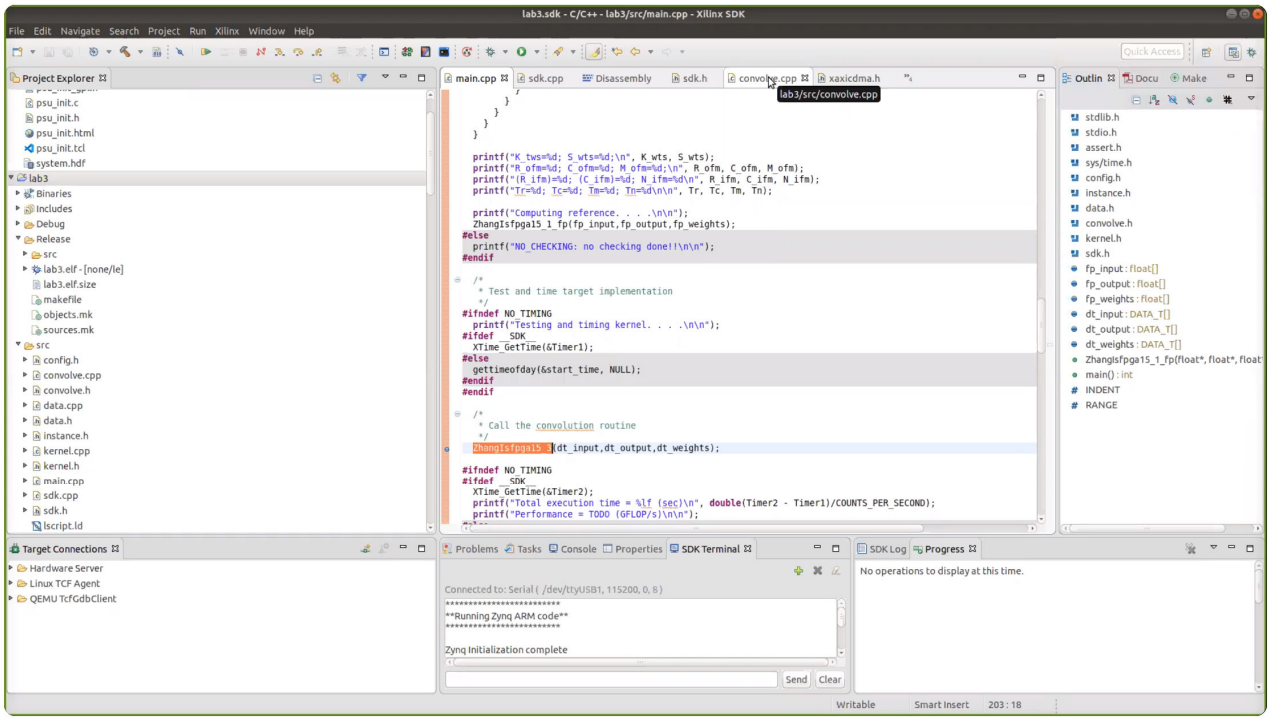
Explicit HW-SW Application Co-Design



Vivado IP Integrator

Two-step process

- **design SoC datapath**
- **program SoC behavior**



Xilinx Software Development Kit (SDK)

Vitis Software-Defined SoC

```
int main(int argc, char* argv[]) {
   ...
   cl::Program program(context, devices, bins);
   ...
   cl::Buffer buffer_a(context, CL_MEM_READ_ONLY,
                       size_in_bytes);
   ...
   q.enqueueMigrateMemObjects({buffer_a,buffer_b},0);
   ...
   q.enqueueTask(krnl_matrix_mult);
   ...
   q.enqueueMigrateMemObjects({buffer_result},
                  CL_MIGRATE_MEM_OBJECT_HOST);
   ...
}
```

load kernel IP to fabric

allocate data buf

send data to kernel

invoke kernel

get data from kernel

The result will be correct, but will it be good?

2015: FPGAs in Datacenters

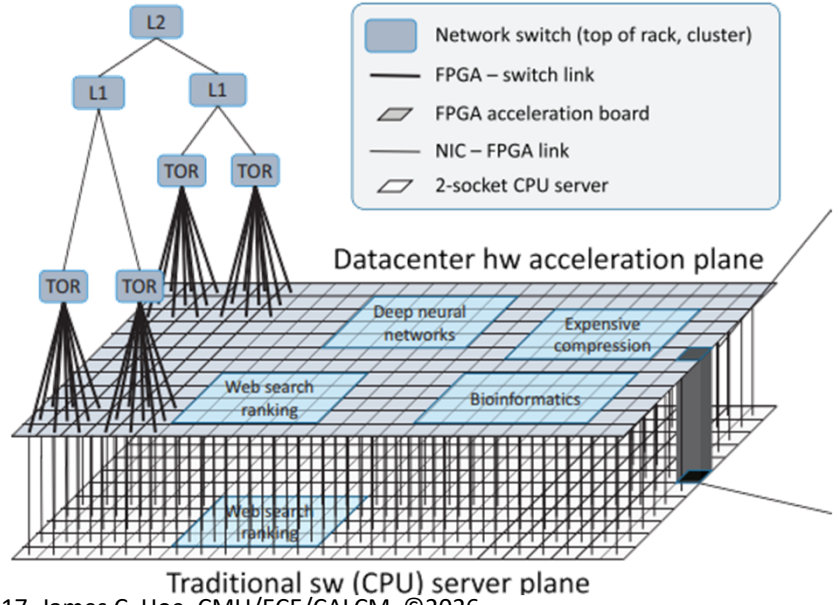
MSR Catapult Bing Experiment
[Putnam et al., 2014]

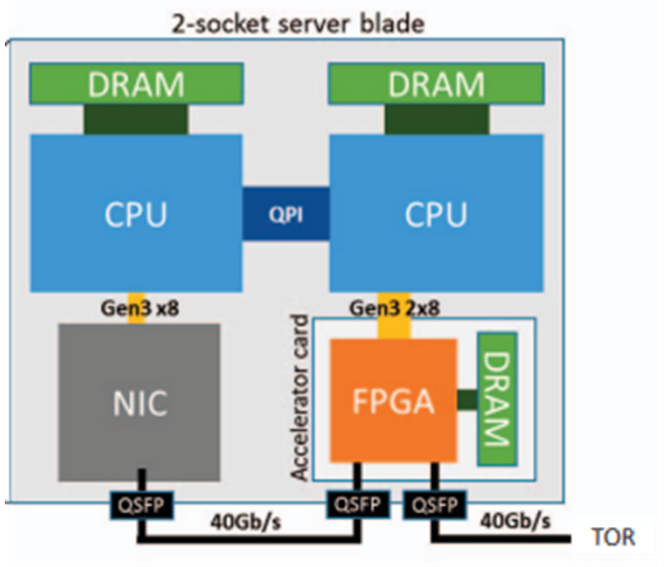
- "Small" scale test (1632 servers) to accelerate Bing ranking using FPGAs
  - *Key Result: 2x throughput at 95th percentile latency*
  - fit in 10% server cost and power budget
  - algorithm updates at intervals of weeks
  - datacenter Reliability/Availability/Serviceability
- Takeaway
  - existence proof of datacenter application
  - modern FPGAs large/capable enough
  - Microsoft desperate enough to pivot from SW-only

In every Microsoft datacenter server

[Caulfield, et al., 2016]

- Individually as SmartNIC (vlan, en/decrypt)
- Individually as CPU off-load accelerator
- Collectively as an FPGA super-accelerator
  - operate separately from host
  - microseconds any FPGA to any FPGA

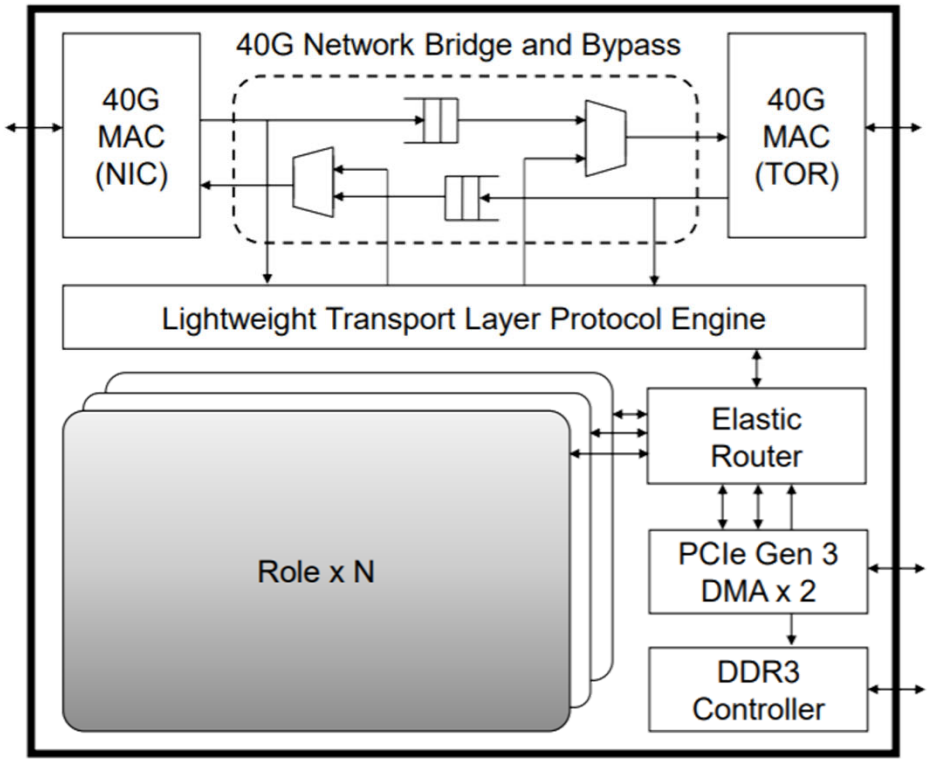




"bump-in-the-wire"

Role-and-Shell

- Fixed “shell”: base NIC fxn & infrastructure wrapper
- Reloadable “roles”: network acceleration, local and remote CPU offload, FPGA accelerator plane



	ALMs	MHz
Role	55340 (32%)	175
40G MAC/PHY (TOR)	9785 (6%)	313
40G MAC/PHY (NIC)	13122 (8%)	313
Network Bridge / Bypass	4685 (3%)	313
DDR3 Memory Controller	13225 (8%)	200
Elastic Router	3449 (2%)	156
LTL Protocol Engine	11839 (7%)	156
LTL Packet Switch	4815 (3%)	-
PCIe DMA Engine	6817 (4%)	250
Other	8273 (5%)	-
Total Area Used	**131350 (76%)**	-
Total Area Available	172600 **24% unused??**	

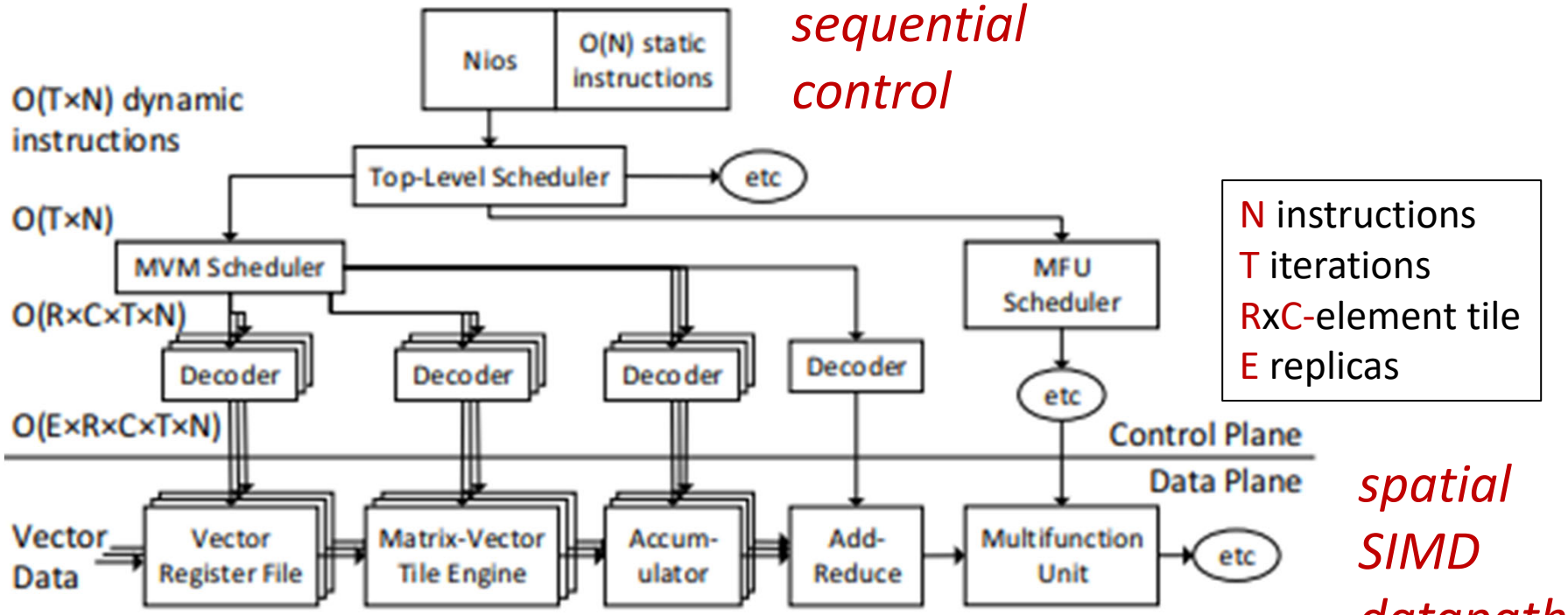
overhead?

1st-gen Stratix V Catapult

Overlay Programming (think μcode)

- ML programmers
  - don't have time to design hardware
  - won't wait 24-hrs to try a new algo
- HW designers bad at ML

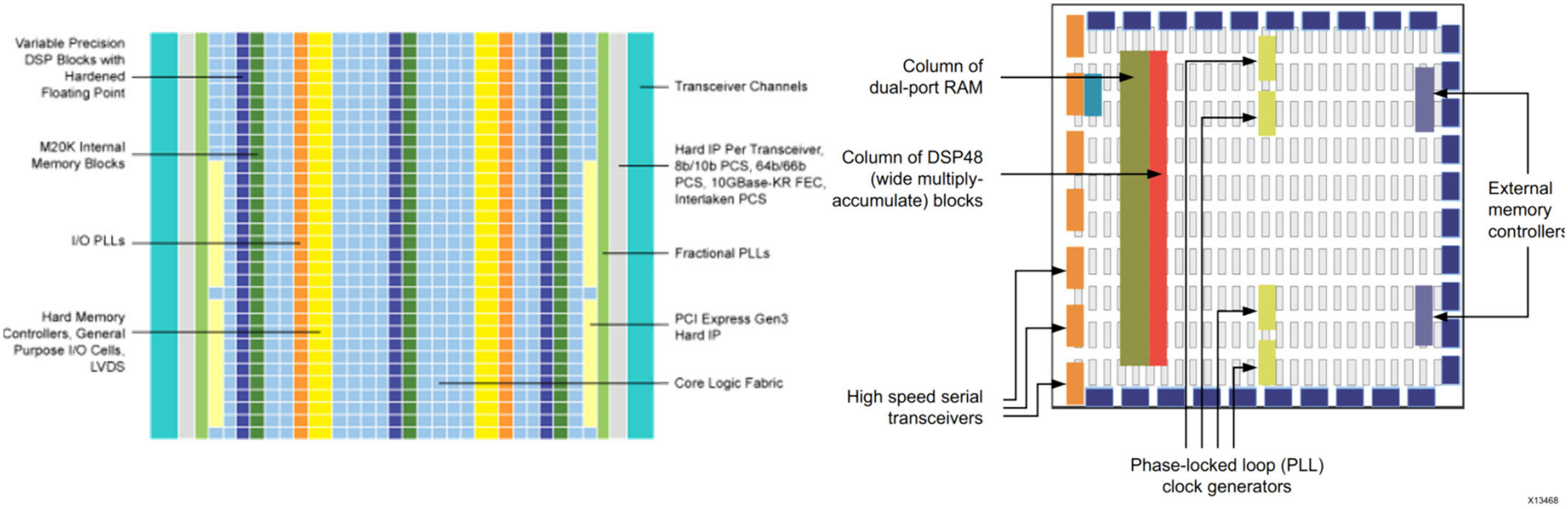
Pay doubly interpretation overhead, okay?



2020: Diverging FPGA Architectures

What is FPGA architecture?

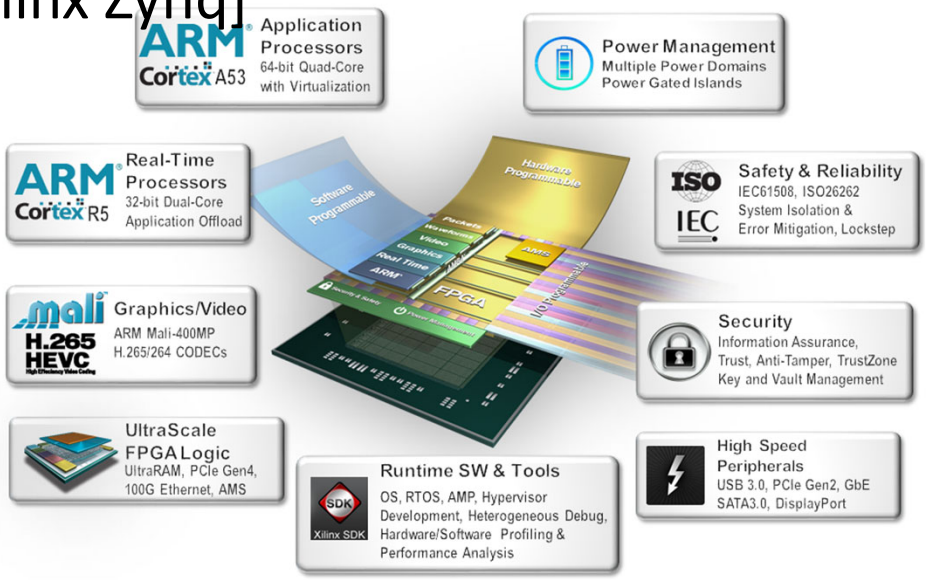
- If you asked in 2015

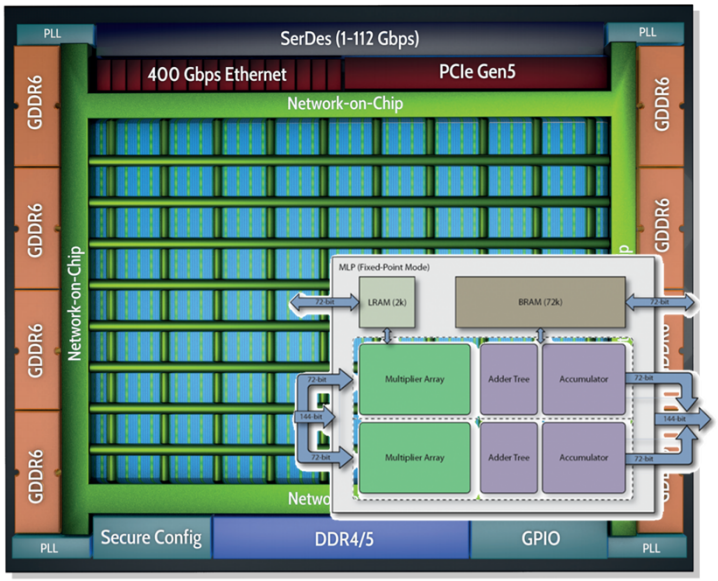


One is Xilinx, the other Altera. Which is which?

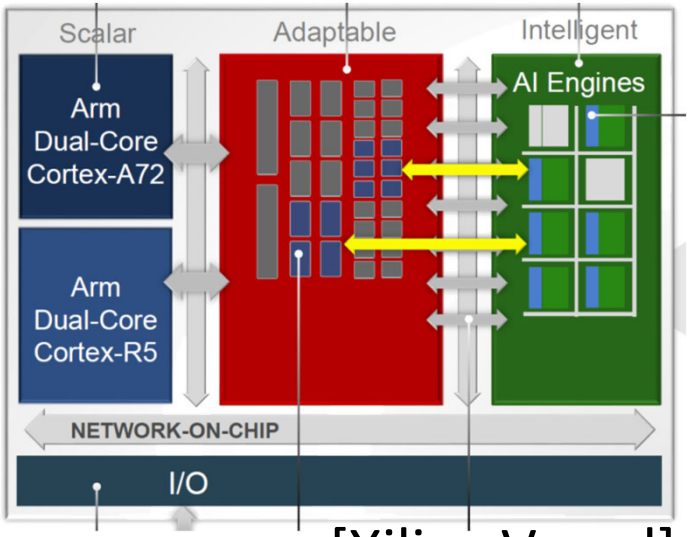
FPGAs not RTL targets

[Xilinx Zynq]





[Achronix Speedster]



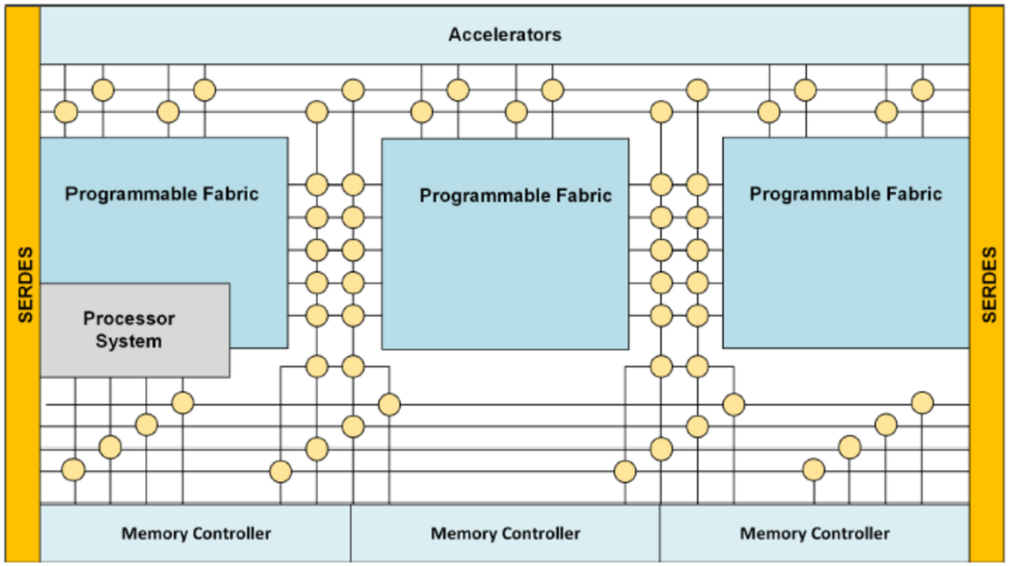
[Xilinx Versal]

[Altera Agilex]

Architecture follows Purpose

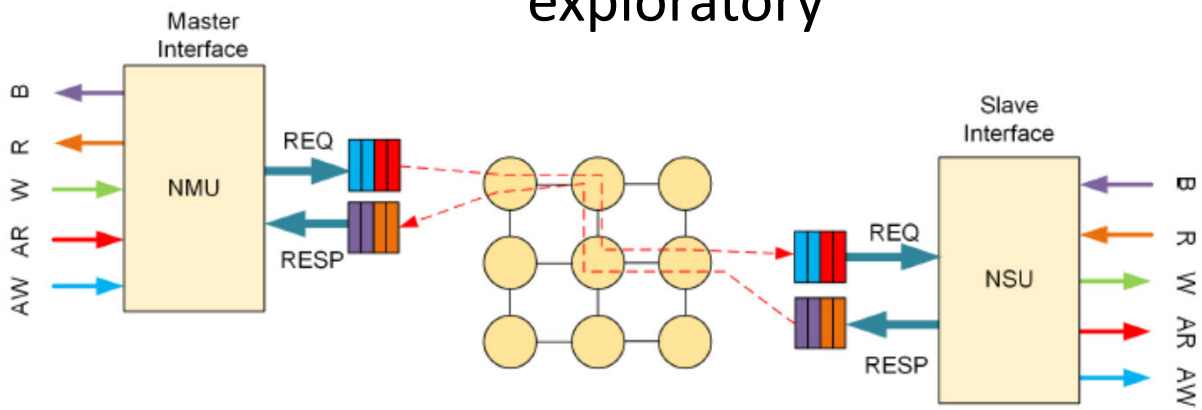
- FPGA vendors doing what markets want
  - future "FPGA" not sea-of-gates for RTL netlist
  - FPGAs wanted not because can't afford ASICs
- **Purposeful architectures for targeted use/app**
  - make select things easier/cheaper to do
  - be very good at what they are intended to do
- Coping with architectural divergence
  - soft-logic adds malleability to "architecture"
  - 2.5/3D integration allows specialization off a common denominator
  - push reconvergence of abstraction up the stack

Xilinx Versal Hardened NoC



Current NoC emphasis on bridge high bandwidth I/O (memory and network) with where compute happens

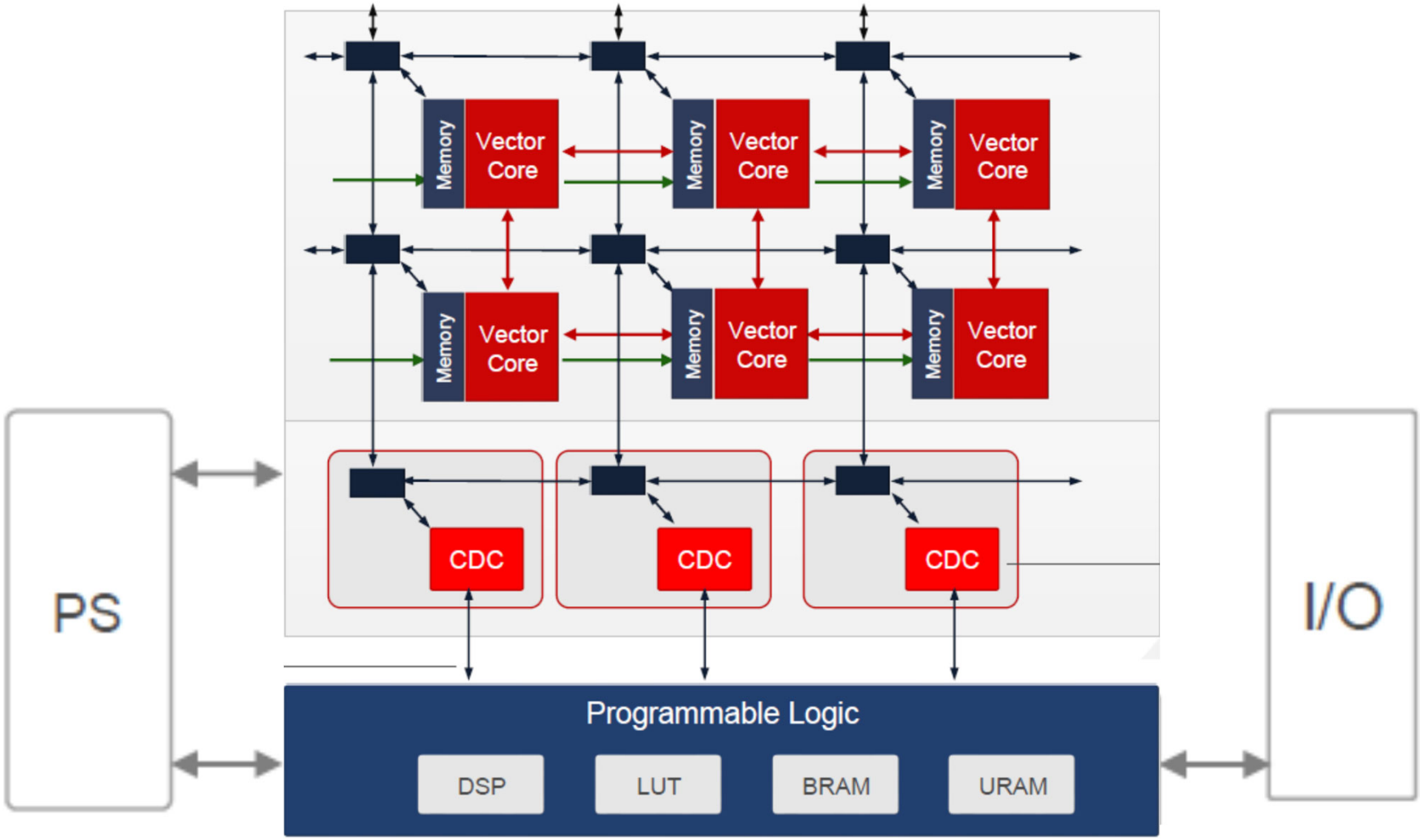
Peer-to-peer use still exploratory



Usage as AXI remains abstracted and automated

ISFPGA 2019: "Network-on-Chip Programmable Platform in Versal™ ACAP Architecture"

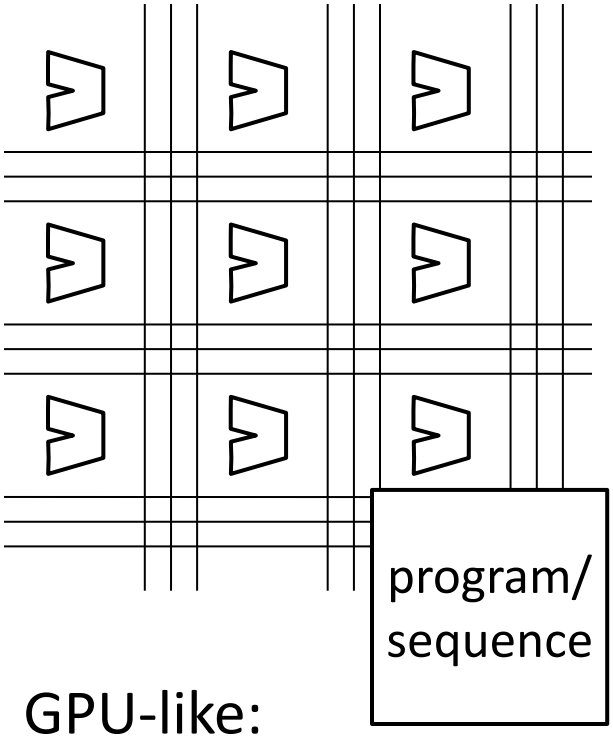
Xilinx Versal AI Engines



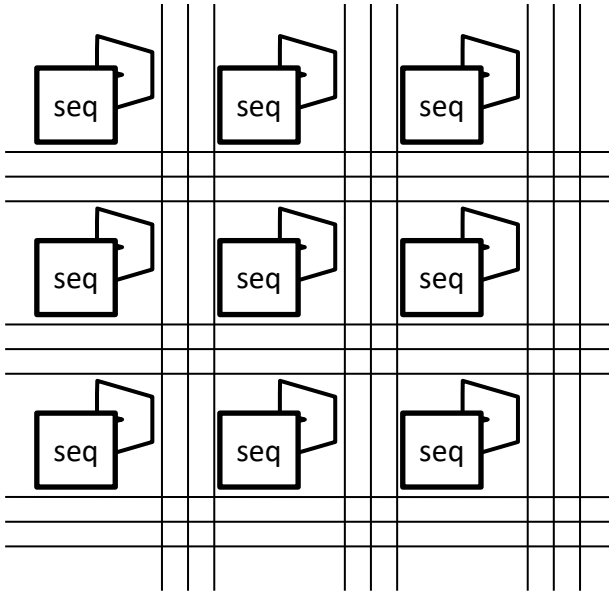
HotChips 2018, “HW/SW Programmable Engine”

If not RTL then what?

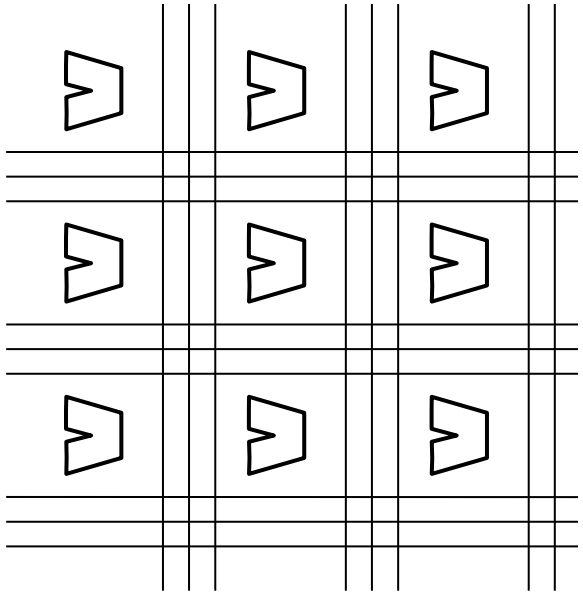
CGRA Combinations (coarse-grained reconfigurable array)



GPU-like:
- coarse operators
- collectively sequenced



many tiny "cores":
- coarse operators
- sequenced

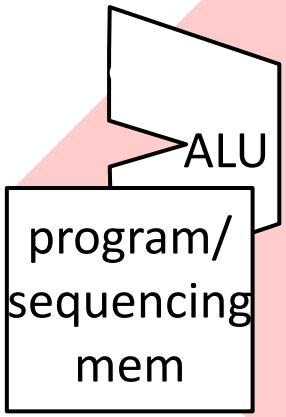


systolic array
static dataflow:
- coarse operators
- not sequenced

Different application (non)sweetspots

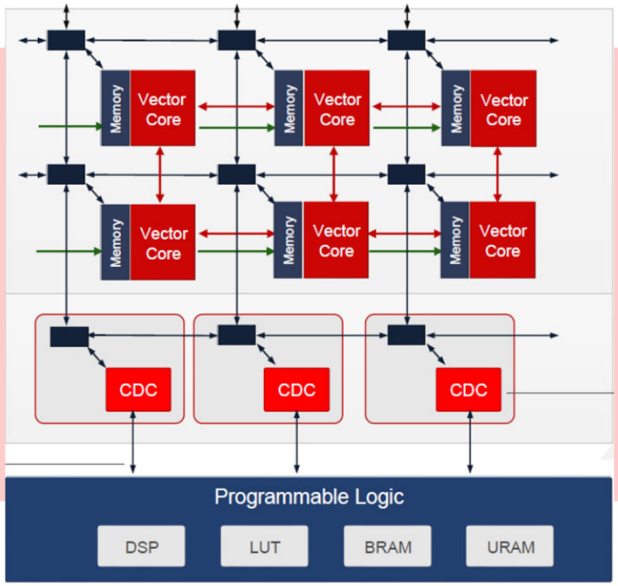
Why CGRAs now?
What is being traded off?

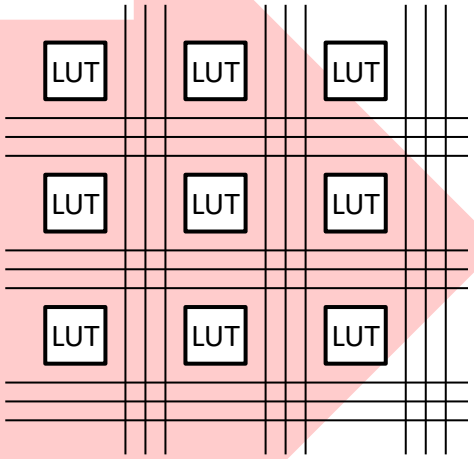
von Neumann → **Spatial**



multicore

manycore, GPU, etc

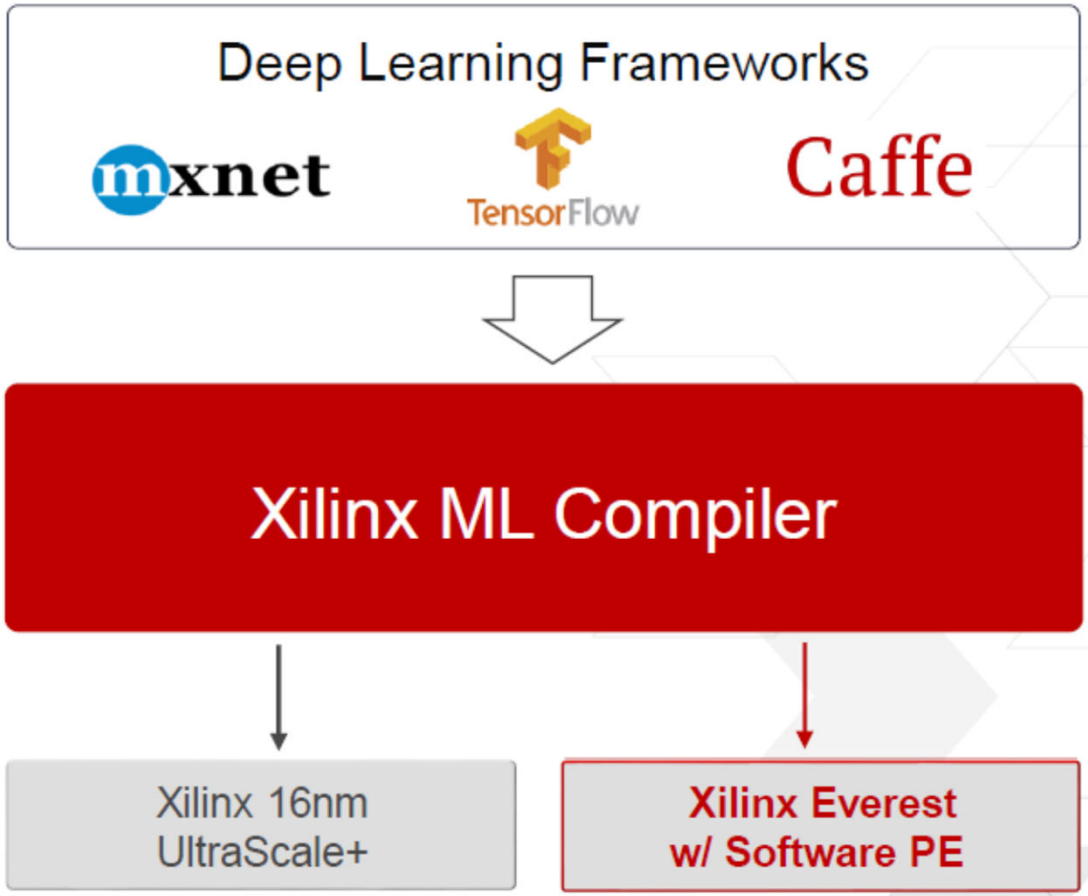




CPU-like:
- coarse operator
- programmed sequencing

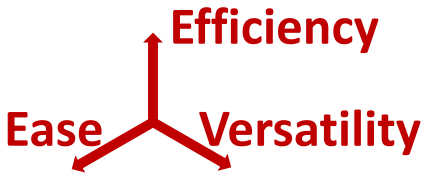
FPGA-like:
- fine operators
- logic netlist (no sequencing)

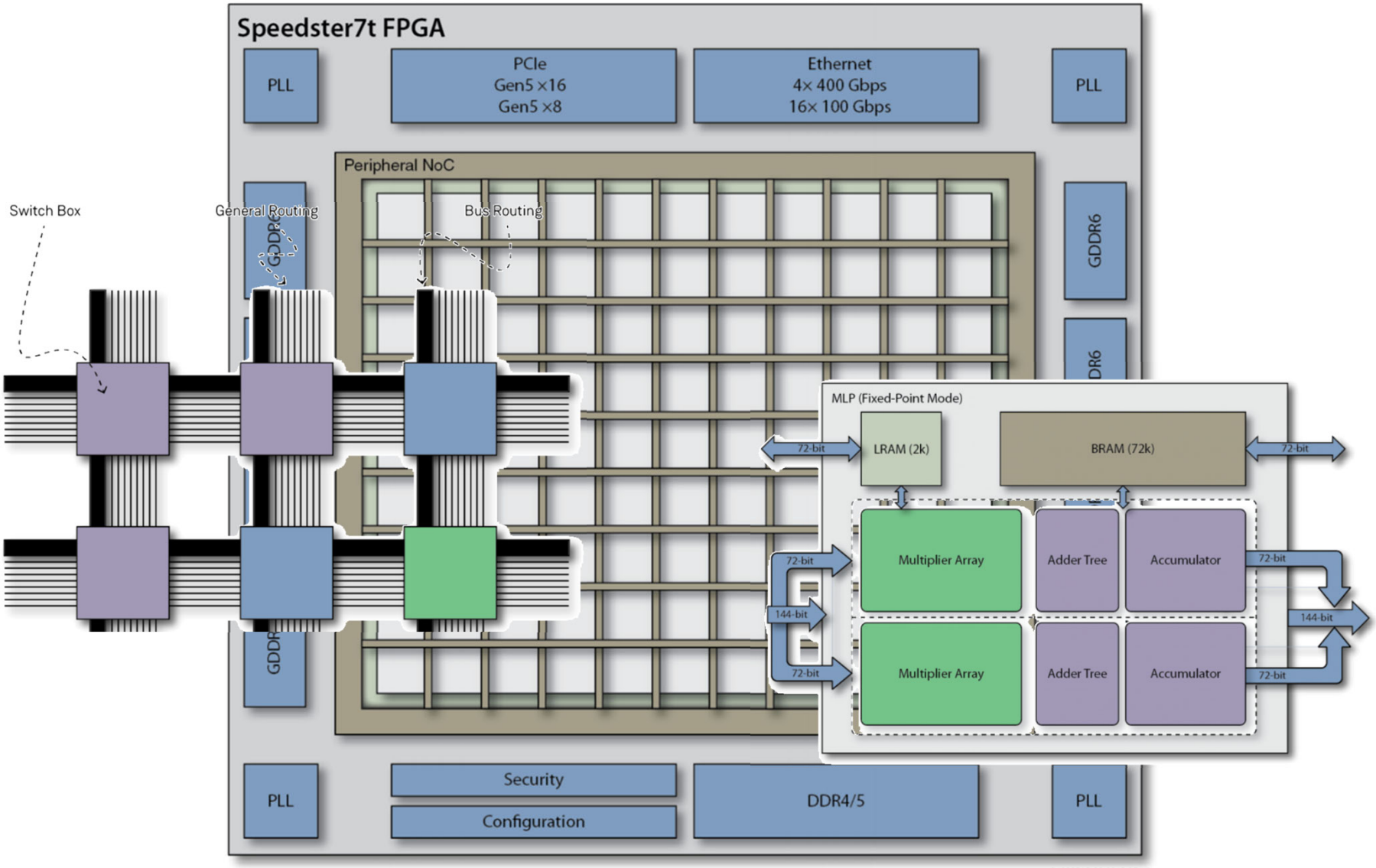
Domain Specialized Programming Support



HotChips 2018, “HW/SW Programmable Engine”

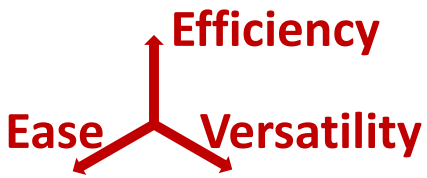
Achronix



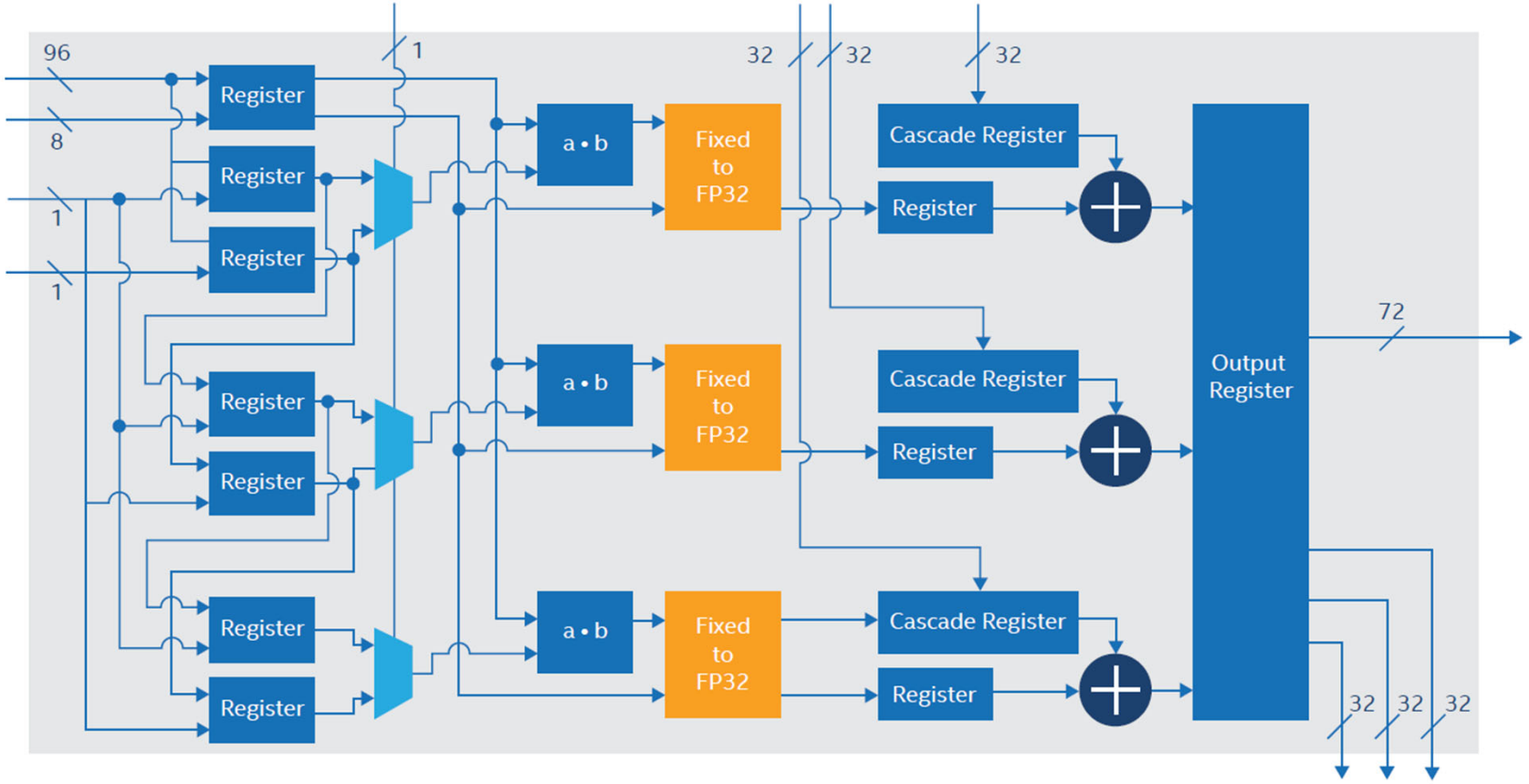


[www.achronix.com]

Intel/Altera AI Tensor Block



AI Tensor Block High-Level Diagram

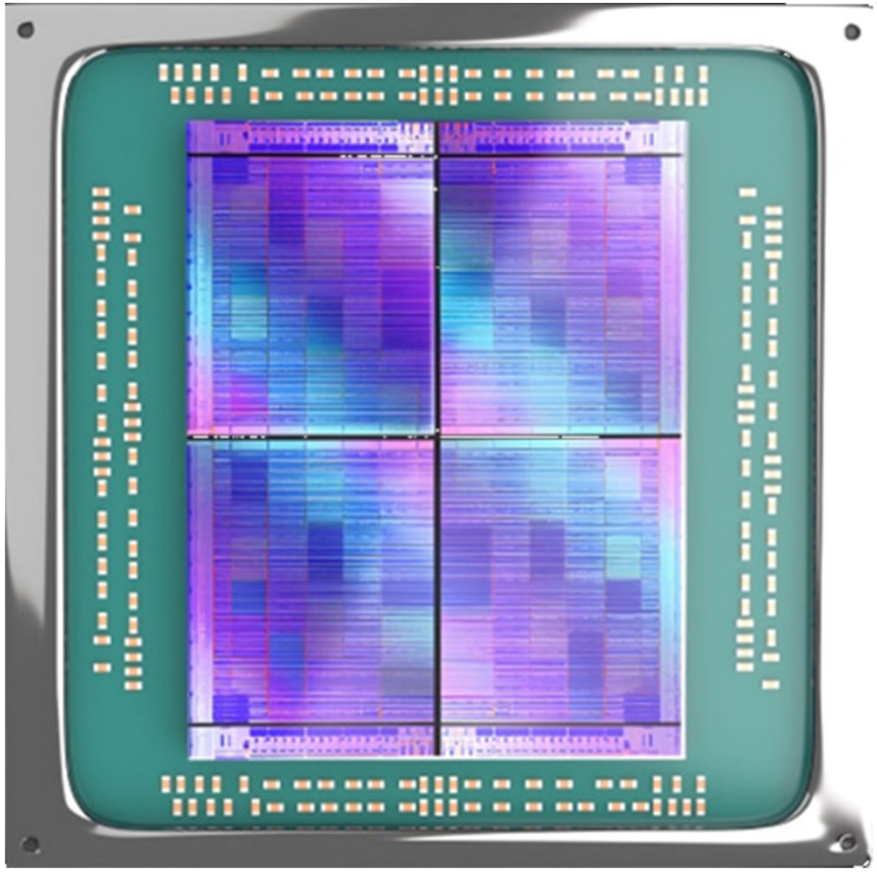


[Intel/Altera Stratix-10 NX FPGA, Technical Brief]

up to 143 INT8 TOPS at ~1 TOPS/W

FPGA for What Purpose Today?

Where we are in 2026



100+ billion transistors

- AMD-Xilinx VP1902 for **ASIC prototyping and emulation**
  - 18.5M logic cells
  - 12Tb/s total GTYP (board-level links)
- AMD-Xilinx VP2802 for **telecom signal processing**
  - 9Tb/s total GTM (for remote optical links)
  - 7.3M logic cells + “AI engines” (152 INT8 TOPS)

Recall

SmartNIC/DPU/IPU

- Microsoft launched AccelNet in 2015 because . . .
  - could not find just the right ASIC to buy
  - not ready to build their own ASIC
  - recognized programmability as value added
- In ensuing years
  - AccelNet expanded to offload more "virtualization tax" to FPGA (out of the CPU)
  - other hyperscalers followed later with similar virtualization offload but in custom ASIC
  - Microsoft also shifted to ASIC+FPGA hybrid in latest

- Programmable-->hardened a sign of maturity

- There are more uses if FPGAs easier to "program"

AI/ML

- FPGAs put up a serious fight pre-ChatGPT
- LLMs pivotally changed the AI landscape
  - FPGAs competitive in small-batch, low-latency inference, but not what datacenter LLMs crave
  - CUDA+GPU+HBM just the right starting point to set off all-hands-on-board virtuous development cycle
  - GenAI economics and scale favor ASICs over FPGAs
- New opportunities
  - sub-usec lightweight inference (physics triggers, fintech, robotics)
  - “LUT-native” neural nets (e.g., ISFPGA’26 KANELÉ)

Parting Thoughts

- SoC'ness complements FPGA'ness
  - hardware performance that is flexible
  - fast design turnaround (time-to-market)
  - low NRE investments
  - in-the-field update/upgrades
- FPGA "architecture" evolving rapidly
  - heterogeneity+cheap transistors --> perf/Watt
  - high-valued applications lead to specialization
  - different high-valued applications lead to "speciation"

Don't let what you see today limit your imagination

Looking Ahead

- Lab 1 (wk3/4): first design with Vitis
  - most important: know what is there
- Lab 2 (wk5/6): try out HLS
  - most important: decide if you like it
- Lab 3 (wk7/8): hands-on with acceleration
  - most important: have confidence it can work
- Project: we already started . . .