J. de Fine Licht and T. Hoefler

Productive parallel programming on FPGA using High-level Synthesis

Based on material from:

Transformations of High-Level Synthesis Codes for High-Performance Computing
https://arxiv.org/abs/1805.08288

Code examples found at:

https://github.com/spcl/hls_tutorial_examples

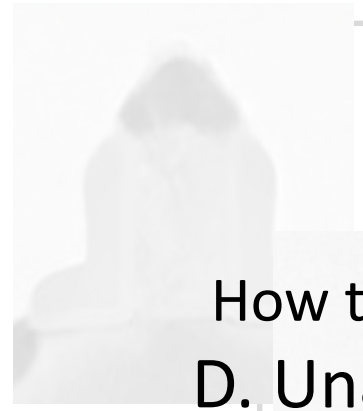
Virtual machine for emulation

http://spcl.inf.ethz.ch/~definelj/HLS_Tutorial.tar.gz

Nimbix Alveo Trial

https://www.nimbix.net/alveotrial

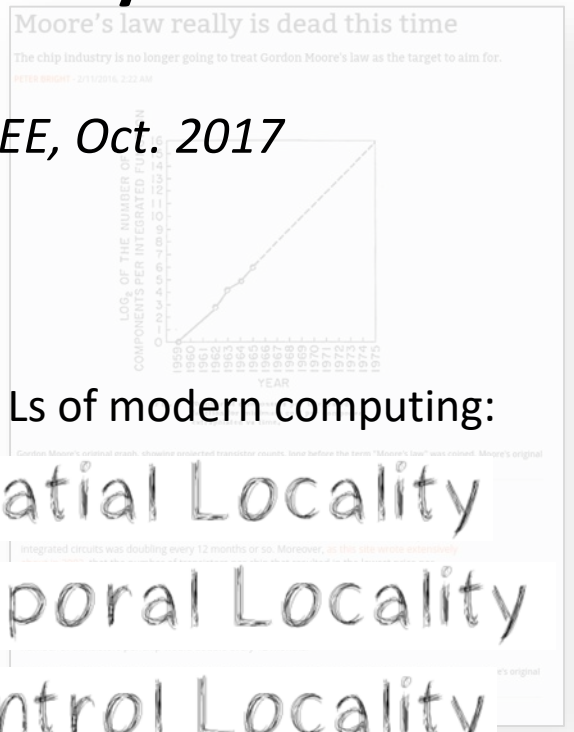
Changing hardware constraints and the physics of computing



How to address locality challenges on standard architectures and programming?

D. Unat et al.: “**Trends in Data Locality Abstractions for HPC Systems**”

IEEE Transactions on Parallel and Distributed Systems (TPDS). Vol 28, Nr. 10, IEEE, Oct. 2017



Three Ls of modern computing:

Spatial Locality

Temporal Locality

Control Locality

[1]: Marc Horowitz, Computing’s Energy Problem (and what we can do about it), ISSC 2014, plenary

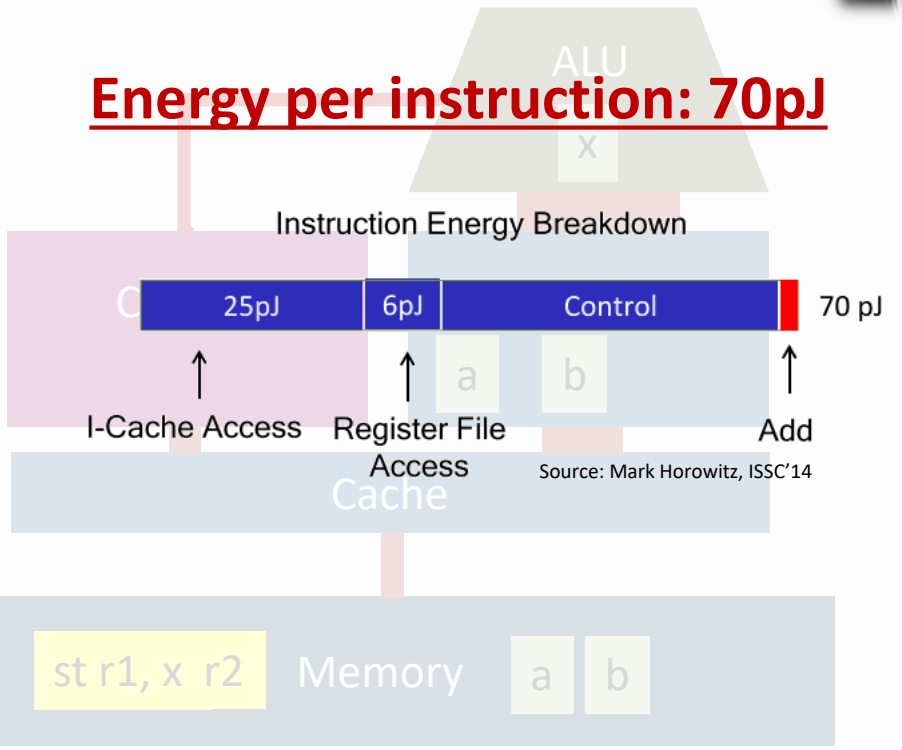
[2]: Moore: Landauer Limit Demonstrated, IEEE Spectrum 2012

Load-store vs. Dataflow architectures

Turing Award 1977 (Backus): "Surely there must be a less primitive way of making big changes in the store than pushing vast numbers of words back and forth through the von Neumann bottleneck."

Load-store ("von Neumann")

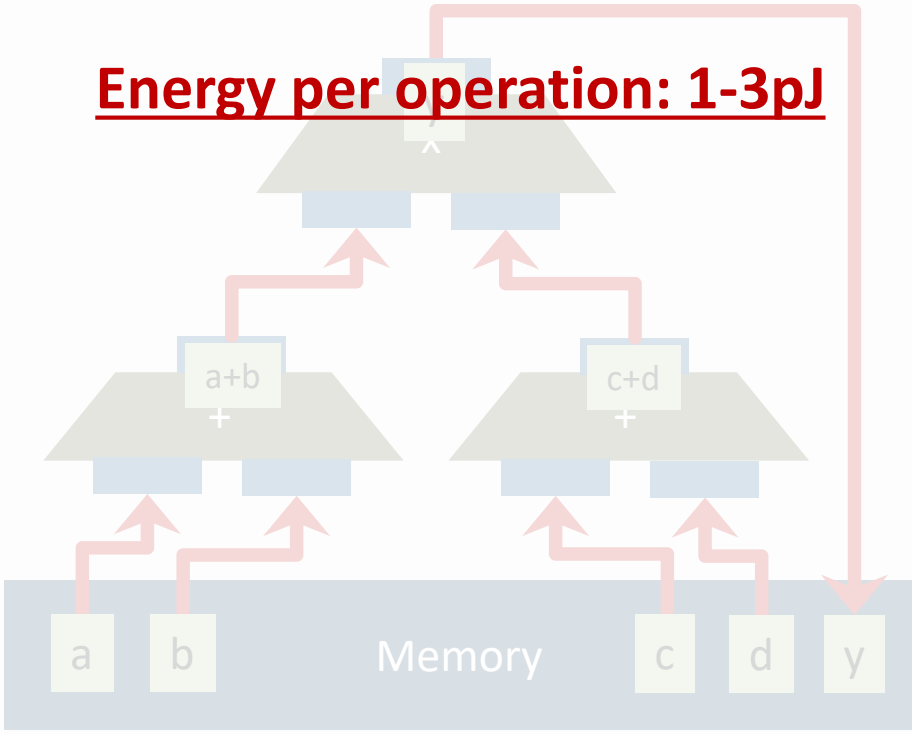
x=a+b



Energy per instruction: 70pJ

Static Dataflow ("non von Neumann")

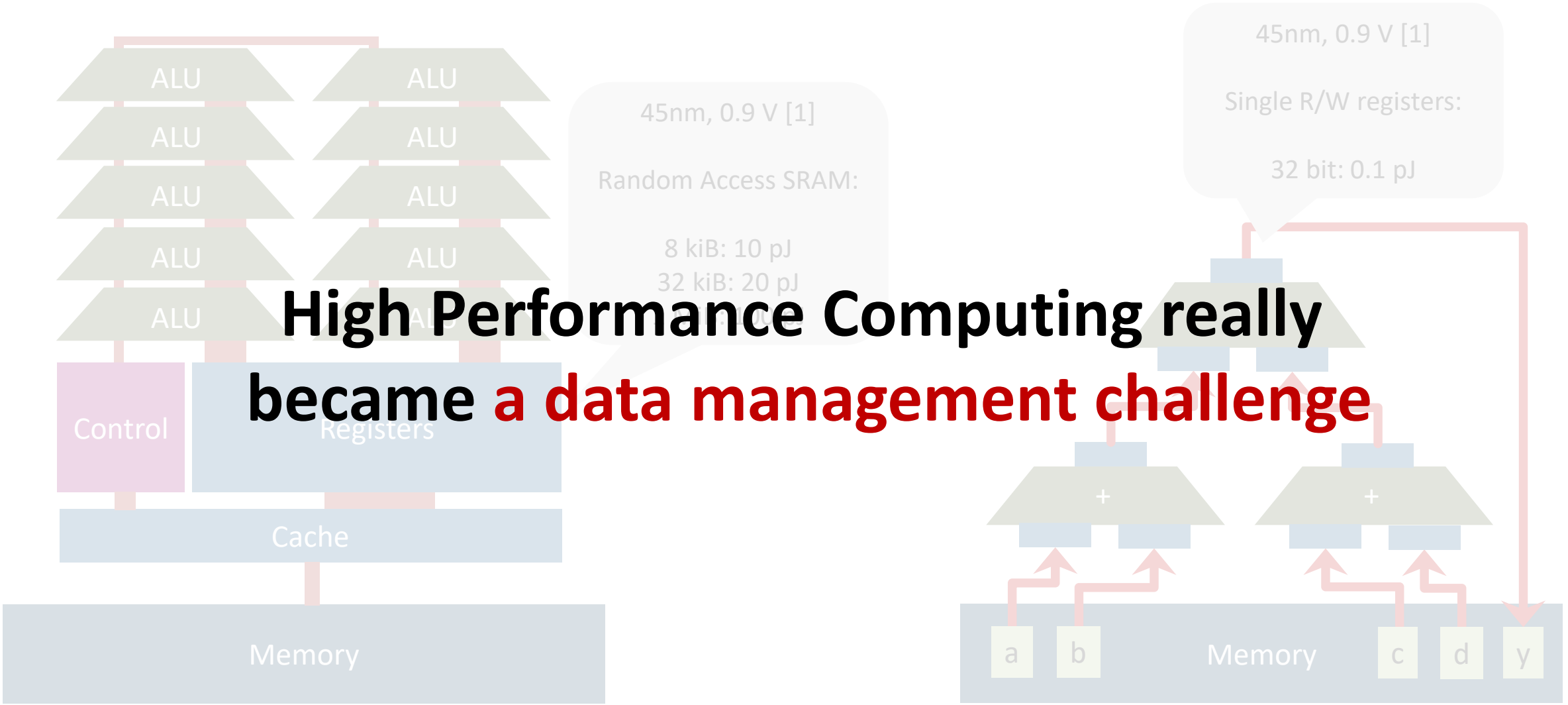
y=(a+b)*(c+d)



Energy per operation: 1-3pJ

Control Locality

Single Instruction Multiple Data/Threads (SIMD - Vector CPU, SIMT - GPU)

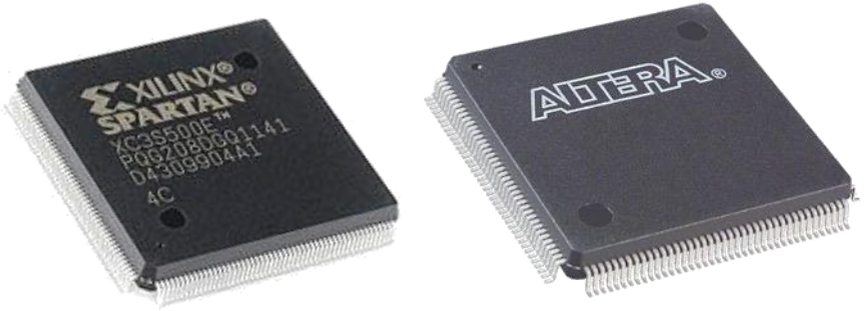


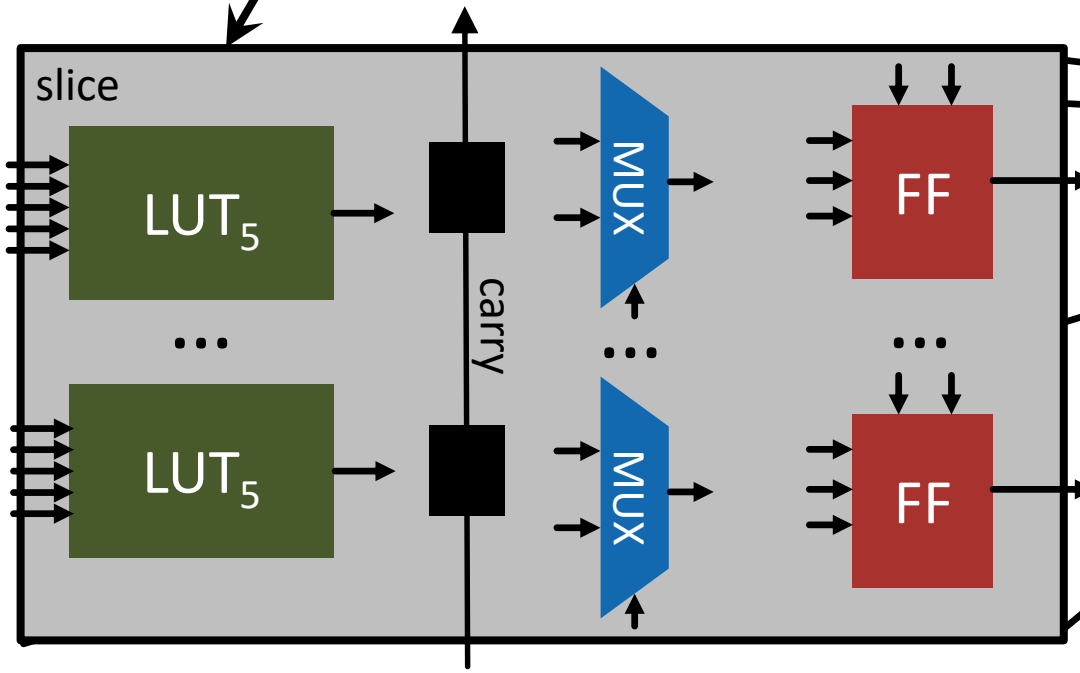
High Performance Computing really became a data management challenge

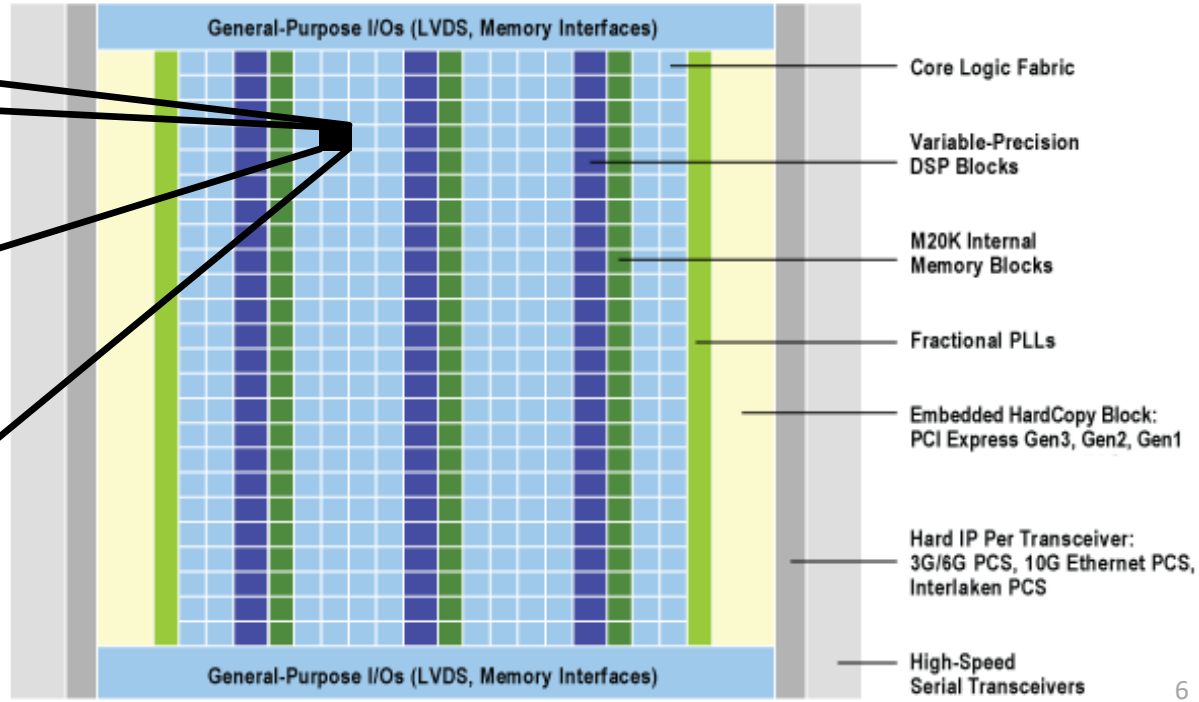
[1]: Marc Horowitz, Computing's Energy Problem (and what we can do about it), ISSC 2014, plenary

Field Programmable Gate Arrays (FPGA)

Address (in[0-4])	Out
00000	0
00001	1
…	…
11111	0







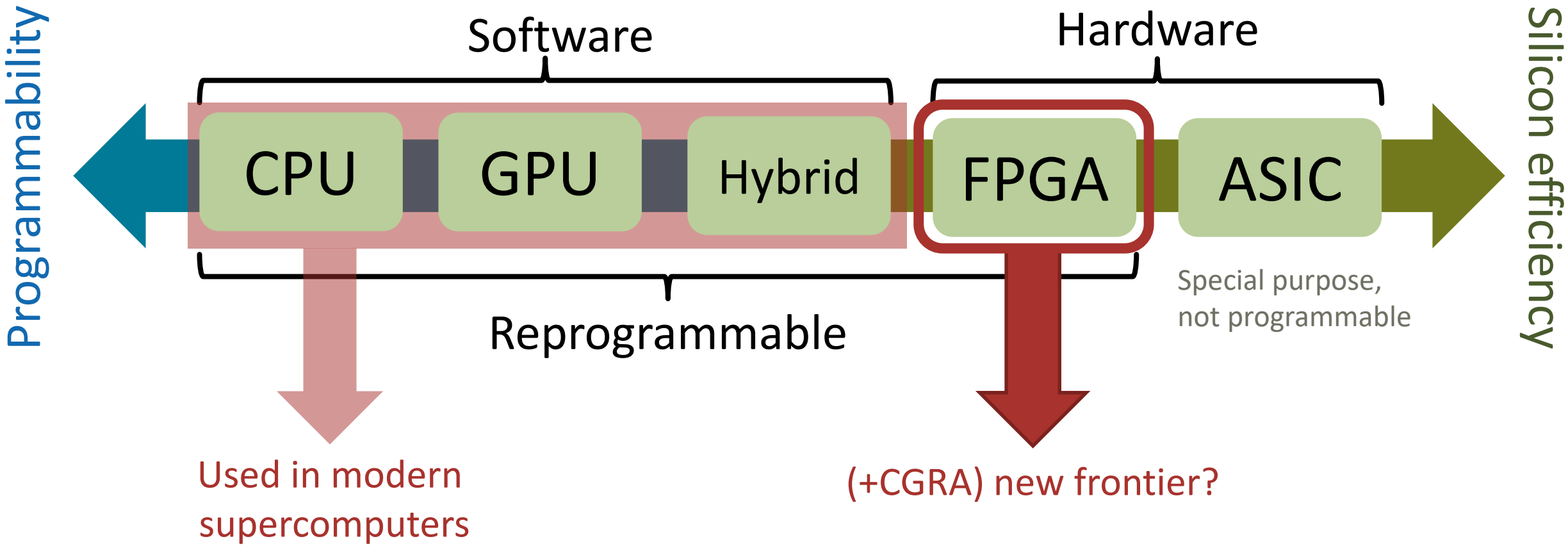
High-Performance FPGA Hardware Overview

- **14 nm Intel Tri-Gate**
  - 1 GHz
- **10 TF single precision**
- **5.5M Logic Elements**
  - 4-input LUT, register, carry, etc.
- **Block RAM: 28.6 MiB**
- **Hardened DRAM controller DDR 4**
  - Various options for memory
- **Hyper Flex Interconnect with Regs.**
- **TDP: 125W (estimated)**

- **14 nm**
  - ~600 MHz
- **8,490 DSPs** (3.8 TF single prec.)
- **2.5M Logic Elements**
  - 1,082,000 5-input LUTs
  - 2,164,000 FFs
- **Block RAM: 39.4 MiB**
- **TDP: 225 W**

- **12 nm**
  - 1455 MHz
- **5,120 cores** (15.7 TF single prec.)
  - CUDA programming
- **On-chip memory:**
  - Registers: 20.8 MiB
  - L1/SM: 7.7 MiB
  - L2 Cache: 6.1 MiB
- **TDP: 300W**

Ease of programmability vs. efficiency



FPGAs are already in the cloud since a while

Microsoft Goes All in for FPGAs to Build Out AI Cloud

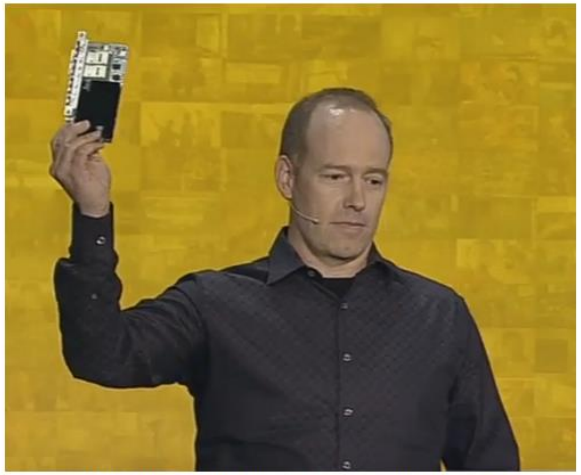
Michael Feldman | September 27, 2016 08:42 CEST

@ E-mail | Tweet | Like | G +1 | in Share

Software giant bets the (server) farm on reconfigurable computing

Microsoft has revealed that Altera FPGAs have been installed across every Azure cloud server, creating what the company is calling "the world's first AI supercomputer." The deployment spans 15 countries and represents an aggregate performance of more than one exa-op. The announcement was made by Microsoft CEO Satya Nadella and engineer Doug Burger during the opening keynote at the Ignite Conference in Atlanta.

The FPGA build-out was the culmination of more than five years of work at Microsoft to find a way to accelerate machine learning and other throughput-demanding applications and services in its Azure cloud. The effort began in earnest in 2011, when the company launched Project Catapult, the R&D initiative to design an acceleration fabric for AI services and applications. The rationale was that CPU evolution, a la Moore's Law, was woefully inadequate in keeping up with the demands of these new hyperscale applications. Just as in traditional high performance computing, multicore CPUs weren't keeping up with demand.



Doug Burger with Microsoft-designed FPGA card

Amazon Adds FPGA Instance to Public Cloud

Michael Feldman | December 7, 2016 07:17 CET

@ E-mail | Tweet | Like | G +1 | in Share

In a blog post published last week, Amazon Web Services Chief Evangelist Jeff Barr announced a new Elastic Compute Cloud (EC2) instance, known as the F1, which incorporates Xilinx FPGAs. The web giant says users will not only be able to build FPGA-accelerated applications for their own purposes with the new instance, but also resell them in the AWS Marketplace to third parties.

Barr makes the conventional pitch for FPGA acceleration, noting its inherent advantage in implementing parallel computation much more efficiently than general-purpose chips by being able to implement an application's dataflow at the level of the logic elements. Writes Barr:

"This highly parallelized model is ideal for building custom accelerators to process compute-intensive problems. Properly programmed, an FPGA has the potential to provide a 30x speedup to many types of genomics, seismic analysis, financial risk analysis, big data search, and encryption algorithms and applications."

The F1 instance is equipped with Xilinx's Virtex Ultrascale+ VU9P, a 16nm device that contains about 2.5 million logic elements and over 6,800 DSP engines. The F1's host CPU is Intel's 18-core Broadwell E5 2686 v4 processor, which runs at 2.3 GHz. The instance may be configured with up to 8 FPGAs, 967 GiB of memory and 4 TB of NVMe flash storage. The PCIe fabric allows multiple FPGAs to communicate with one another directly, as well as share the same memory space.

Application developers will have access to the Xilinx Vivado Design Suite free of charge, but that will require that the application code be implemented in VHDL or Verilog. Third-party tools can also be used, including higher level frameworks like OpenCL, but that means users will be responsible for their own development environments.

At this point, FPGA application development is geared toward a rather niche audience, so this will hardly be a volume business for Amazon in the near-term. HPC cloud specialist Nimbix recently added Xilinx FPGAs to its own infrastructure in the hopes of building a customer base of FPGA computing enthusiasts. Much of the initial customer base will be Xilinx engineers themselves, who will be developing and testing software on their own product.

Vendors trying to push FPGAs into HPC

Intel Launches FPGA Accelerator Aimed at HPC and HPDA Applications

Michael Feldman | December 20, 2017 12:44 CET

Intel has released the Stratix 10 MX, the company's first FPGA family that includes integrated High Bandwidth Memory (HBM2).

HBM2 is comprised of a stack of DRAM chips linked to one another in parallel using thru silicon via (TSV) interconnects. The 3D design enables much faster data communication rates compared to normal 2D setups, which are limited by the number of pins that can be connected to the edge if the DRAM chips. According to Intel, HBM2 will deliver memory speeds as much as 10 times faster – up to 256 GB/second per HBM2 stack – than what would be possible with a conventional DDR4 solution. The technology also has the advantage of delivering about 35 percent better performance per watt.

Depending on the specific model, and Stratix 10 MX devices will be equipped with 3.25, 8, of 16 gigabytes of HBM2 memory.

But HPC uses GPUs – why?

- **Productivity:** CUDA enabled GPGPU without hacking the graphics pipeline
- **Hardware support:** Tesla line with ECC, double/half precision

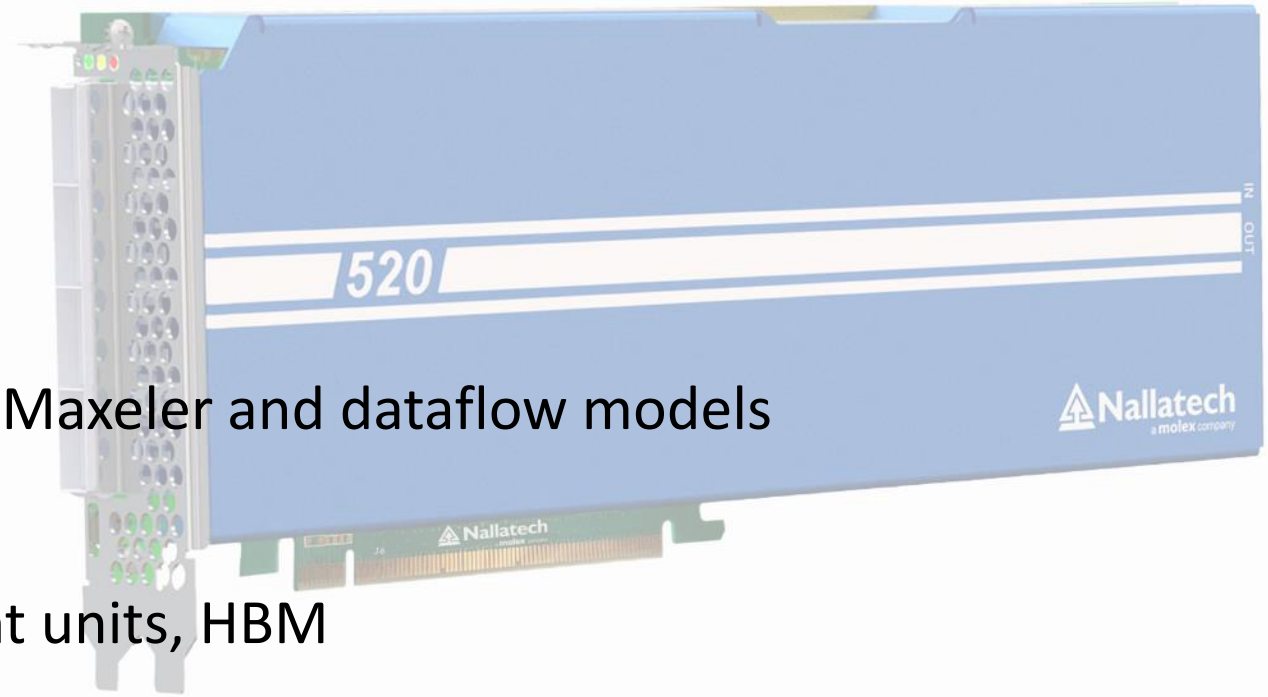
```
template <typename T>
__global__ void integrateBodies(typename vec4<T>::Type *__restrict__ newPo
                                typename vec4<T>::Type *__restrict__ ol
                                typename vec4<T>::Type *vel,
                                unsigned int deviceOffset,
                                unsigned int deviceNumBodies,
                                float damping, int numTile
  int index = blockIdx.x * blockDim.x + threadIdx.x;


  if (index >= deviceNumBodies) {
    return;
  }


  typename vec4<T>::Type position = oldPos[deviceOffset + index];
```

FPGA are not used in HPC - why?

- **Productivity:** steep learning curve of hardware design, unpolished tools
- **Hardware support:** low bandwidth, no native floating point units



Recent developments

- **Productivity:** OpenCL, HLS, (not so recent) Maxeler and dataflow models
  - Major focus on portability now!
- **Hardware support:** Hardened floating point units, HBM

High-level synthesis

- **Both major vendors, as well as third parties, now offer HLS tools**
  - Input C/C++/OpenCL is transformed to the spatial paradigm
  - Lift programming from the bit level to the word/datatype level

- **Xilinx and Intel both offer a C/C++ and an OpenCL tool**
  - Xilinx focused on the former, and Altera on the latter
  - Many other HLS tools …

Bambu

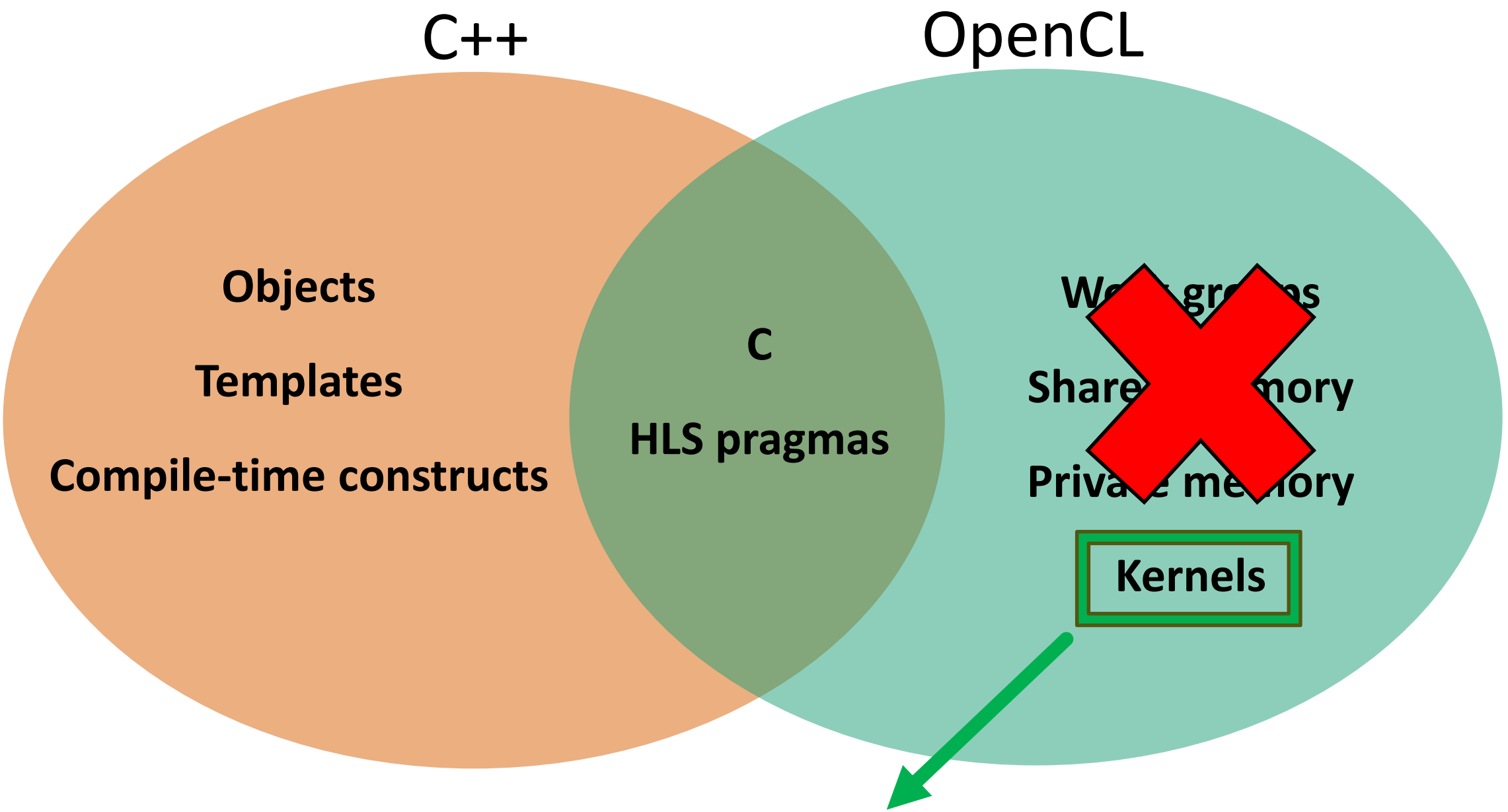
LegUp High-Level Synthesis

impulse accelerated technologies

bluespec

Microsoft Catapult

… many many more (cf. Nane et al.: “A Survey and Evaluation of FPGA High-Level Synthesis Tools”, Oct. 2016)



Primarily concerns interaction between host and device