T. HOEFLER

Will FPGAs make it this time?

Panel, SC18 – November 2018, Dallas, TX

FPGAs keep coming back …

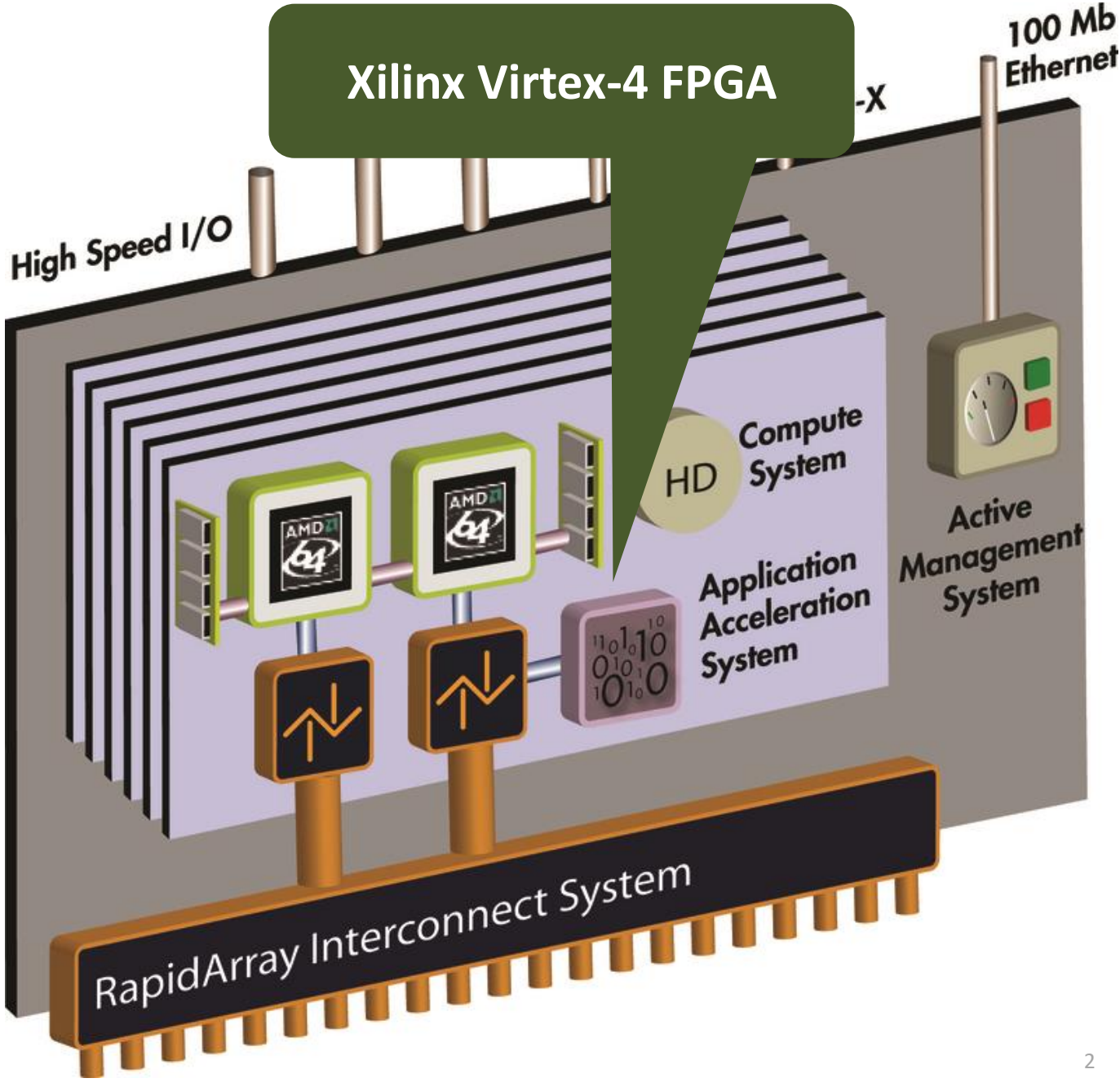
- **Every 10-15 years – e.g., Cray XD1**
  - Nearly 15 years ago!

- **I personally know no single application using the FPGA**
  - At that time, would require users to write RTL-level code

    I loved it! Of course, as a CS student ☺

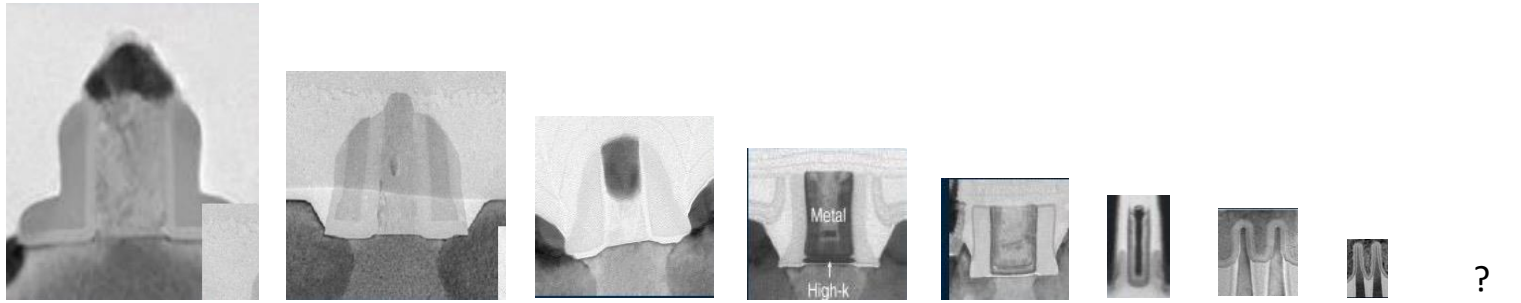
- **Now fast forward to 2018?**
  - Will FPGAs succeed?

    My answer: yes and no!



Why will FPGAs succeed?

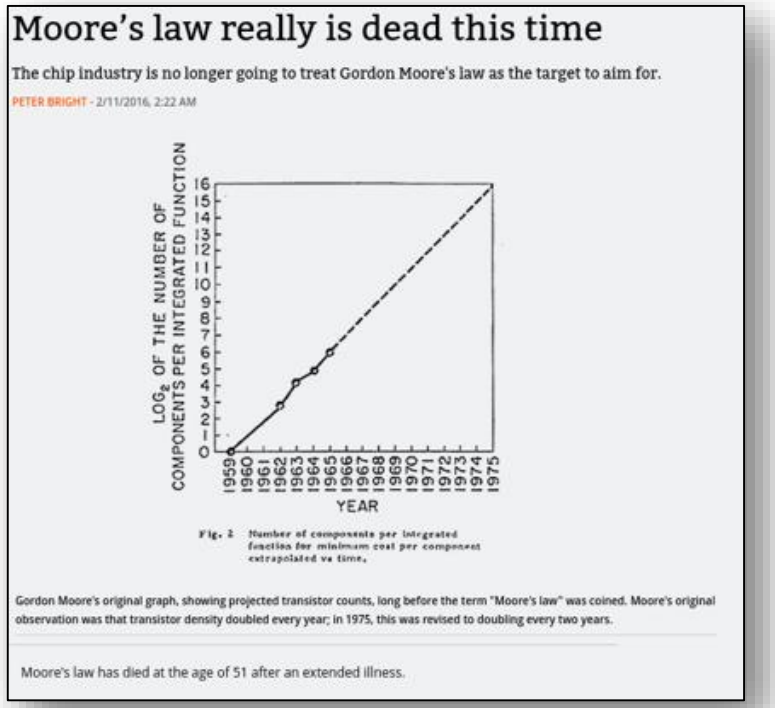
1. The HPC industry largely bases on process scaling



130nm 90nm 65nm 45nm 32nm 22nm 14nm 10nm 7nm

- The end of Moore's law will kill our industry

  reconfigurable technologies can delay it a slight bit ☺



Moore's law really is dead this time

The chip industry is no longer going to treat Gordon Moore's law as the target to aim for.

PETER BRIGHT - 2/11/2016, 2:22 AM

Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Gordon Moore's original graph, showing projected transistor counts, long before the term "Moore's law" was coined. Moore's original observation was that transistor density doubled every year; in 1975, this was revised to doubling every two years.

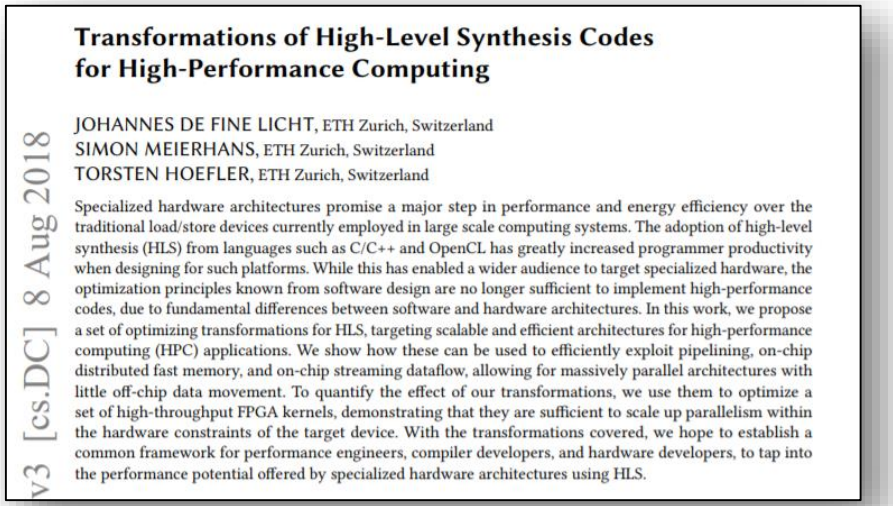
Moore's law has died at the age of 51 after an extended illness.

2. Because they can (now) and the interest is there!

- HLS is widely available now (much better than 2004 at least!)

OpenCL

SC18 Sunday Tutorial: Productive Parallel Programming for FPGA with High-Level Synthesis

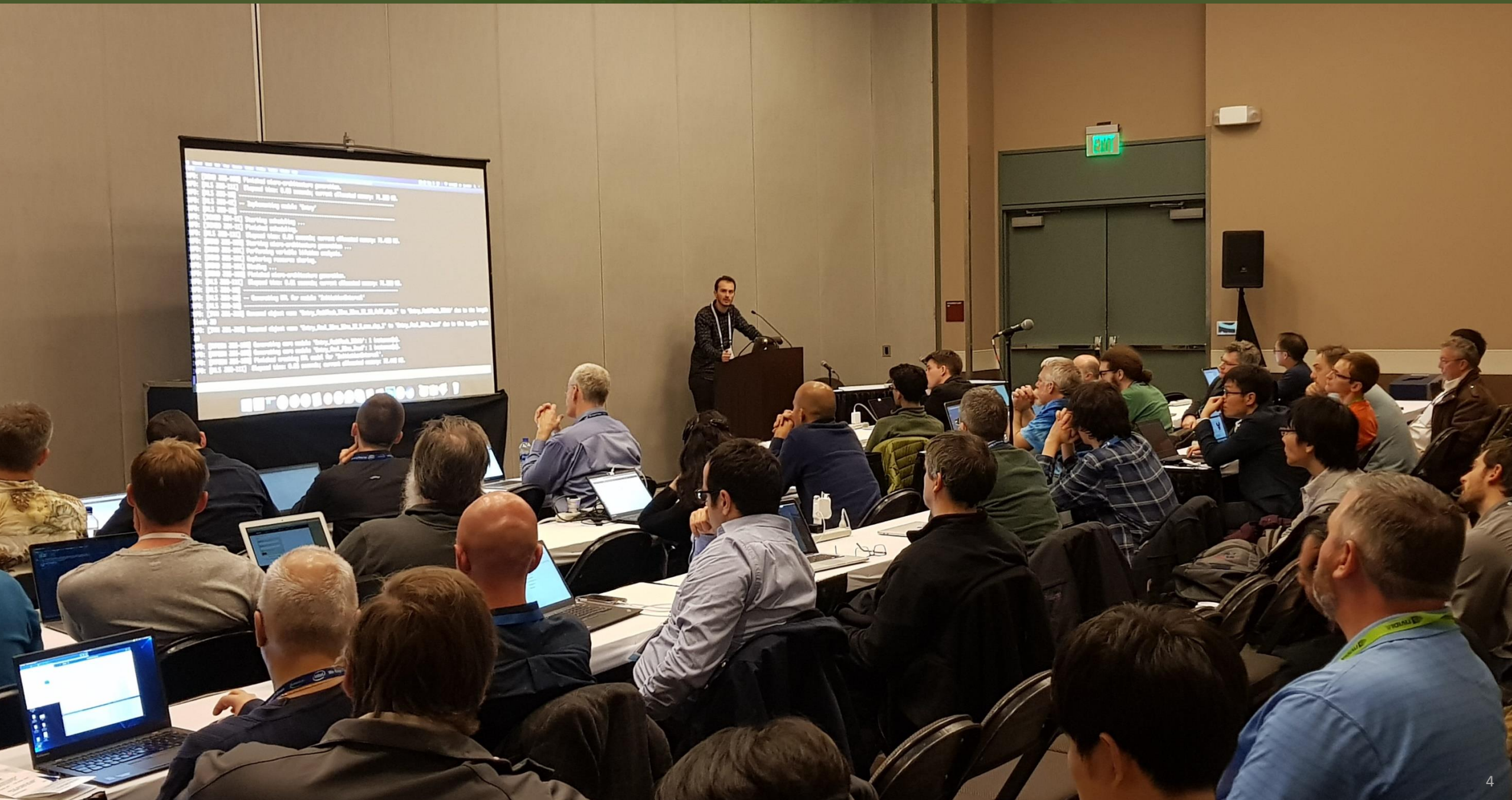


Transformations of High-Level Synthesis Codes for High-Performance Computing

JOHANNES DE FINE LICHT, ETH Zurich, Switzerland
SIMON MEIERHANS, ETH Zurich, Switzerland
TORSTEN HOEFLER, ETH Zurich, Switzerland

v3 [cs.DC] 8 Aug 2018

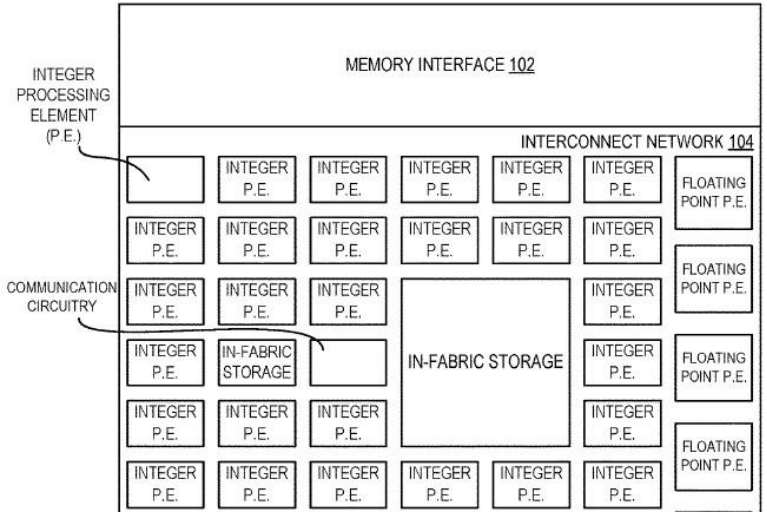
Specialized hardware architectures promise a major step in performance and energy efficiency over the traditional load/store devices currently employed in large scale computing systems. The adoption of high-level synthesis (HLS) from languages such as C/C++ and OpenCL has greatly increased programmer productivity when designing for such platforms. While this has enabled a wider audience to target specialized hardware, the optimization principles known from software design are no longer sufficient to implement high-performance codes, due to fundamental differences between software and hardware architectures. In this work, we propose a set of optimizing transformations for HLS, targeting scalable and efficient architectures for high-performance computing (HPC) applications. We show how these can be used to efficiently exploit pipelining, on-chip distributed fast memory, and on-chip streaming dataflow, allowing for massively parallel architectures with little off-chip data movement. To quantify the effect of our transformations, we use them to optimize a set of high-throughput FPGA kernels, demonstrating that they are sufficient to scale up parallelism within the hardware constraints of the target device. With the transformations covered, we hope to establish a common framework for performance engineers, compiler developers, and hardware developers, to tap into the performance potential offered by specialized hardware architectures using HLS.


EXIT

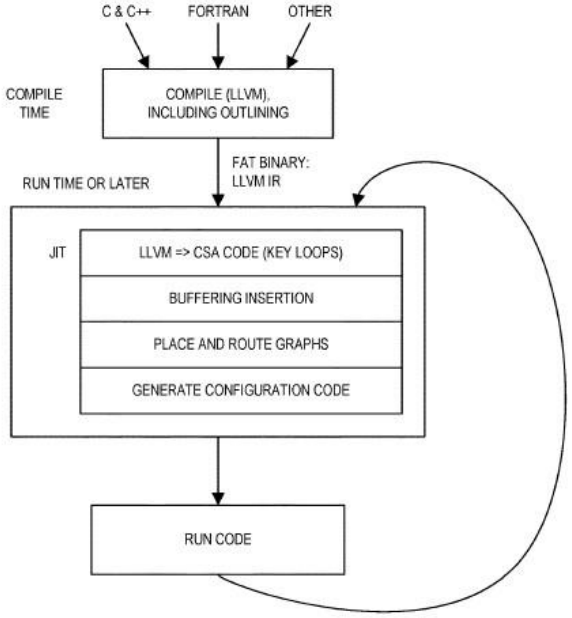
Why will FPGAs fail?

1. **FPGA vendors don't understand software! The toolchain is very sad … a little play in three acts**
   - **Act 1**: run 30 applications (polybench) through Vendor's HLS compiler
     *all **30 simulate correctly but only 14 produce correct results on FPGA*** ☹
   - **Act 2**: submit bug report: the compiler tool crashes due to the power report generated by the tool exceeding the maximum size for the Google Protocol Buffer.
   - **Act 3**: Company's Principal Engineer on a public forum: ***"If it doesn't result in a bitstream, on a shipped board, then there is no money. No money = no interest.*** *An academic exercise is of little import here ."*
2. **FPGAs may not be the right design for HPC**

Learn from the past failures? ☺



Source: nextplatform.com [1]



Source: nextplatform.com [1]

[1] https://www.nextplatform.com/2018/08/30/intels-exascale-dataflow-engine-drops-x86-and-von-neuman/