“Different” doesn’t mean “Difficult”: FPGA Programming Demystified

Sergei Storojev
Staff Tools and Design Methodology Application Engineer

XILINX.

The Case for FPGAs in Financial Computing

Standard Initial Margin Model Calculation

https://www.xilinx.com/products/boards-and-kits/alveo/applications/standard-initial-margin-model.html

10x - 50x

Real-Time Risk Dashboard

https://www.xilinx.com/products/boards-and-kits/alveo/applications/real-time-risk.html

89x

High Performance Monte Carlo Option Pricing Simulation

https://github.com/KitAway/FinancialModels_AmazonF1

42x - 540x

NOT STAC BENCHMARKS

XILINX.

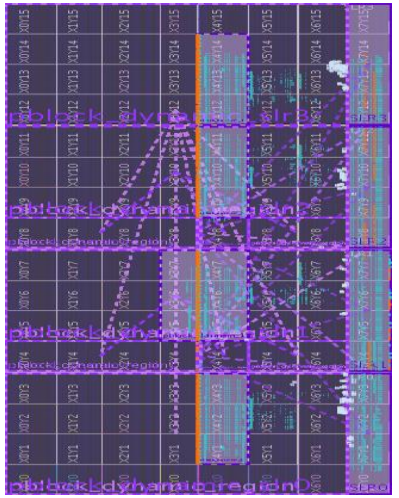
Customizable Architectures – The FPGA Advantage

CPUs and GPUs

- High speed, high power, low efficiency
- Fixed instruction set and rigid memory hierarchy
- Developer adapts the program to the architecture

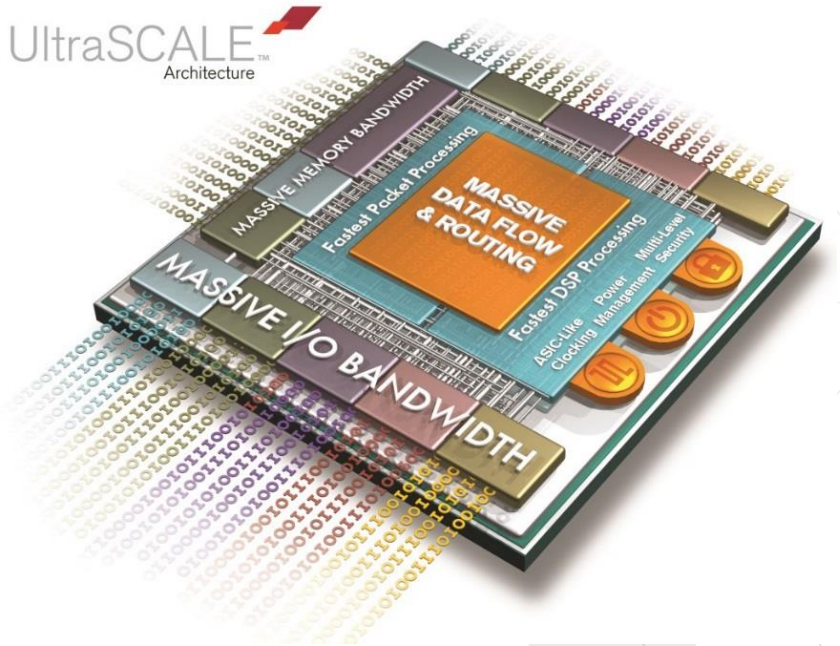
FPGAs

- High throughput, low power, high efficiency
- Flexible, fully customizable architecture
- Developer adapts the architecture to the program



FPGAs – The Ultimate Parallel Processing Device

- **No predefined instruction set or underlying architecture**
- **Developer customizes the architecture to his needs**
  - Custom datapaths
  - Custom bit-width
  - Custom memory hierarchies
- **Excels at all types of parallelism**
  - Deeply pipelined (e.g. Video codecs)
  - Bit manipulations (e.g. AES, SHA)
  - Wide datapath (e.g. DNN)
  - Custom memory hierarchy (e.g: Data analytics)
- **Adapts to evolving algorithms and workload needs**



XILINX.

FPGA Accelerator Cards That Fit Your Performance Needs

- **Alveo U200**
  - 18.6 Peak INT8 TOPs
  - 77GB/s DDR Memory Bandwidth
  - 31TB/s Internal SRAM Bandwidth
  - 892,000 LUTs
- **Alveo U250**
  - 33.3 Peak INT8 TOPs
  - 77GB/s DDR Memory Bandwidth
  - 38TB/s Internal SRAM Bandwidth
  - 1,341,000 LUTs
- **Deploy in the Cloud or On-Premises**

Sounds Great! But…

HW languages are low-level and very difficult

I know nothing of hardware programming concepts

How can I even get software applications to interact with FPGAs?

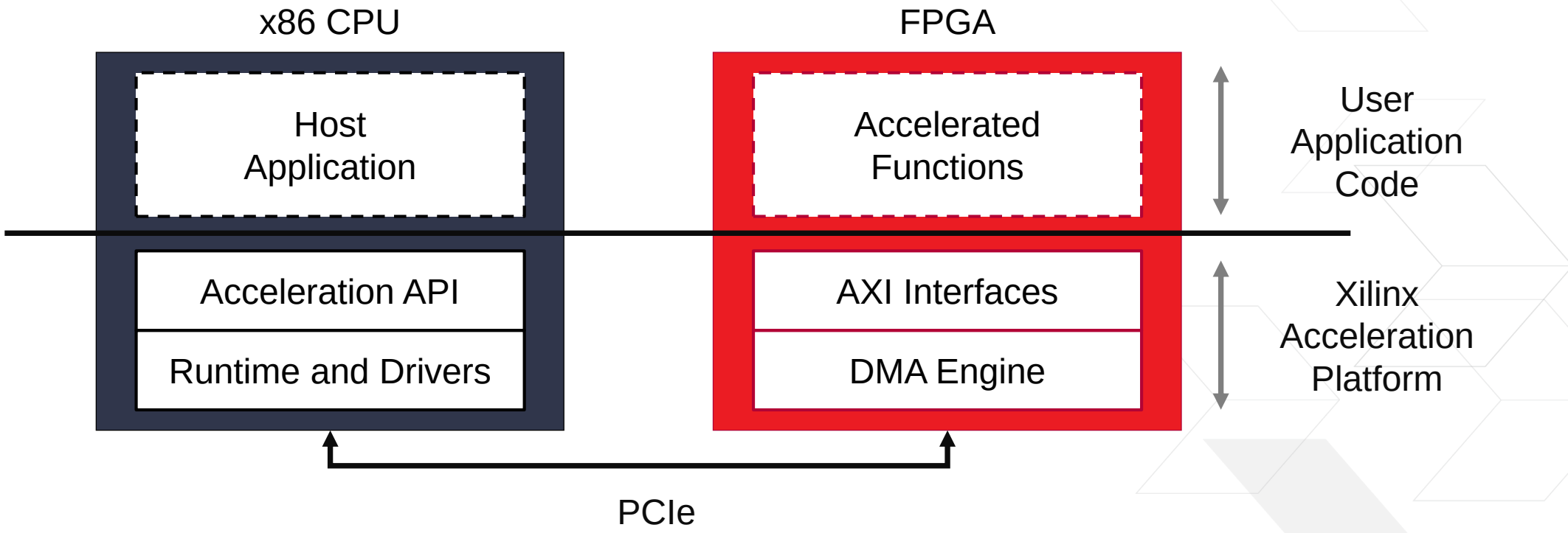
FPGA Solutions Have Come a Long Way

You can actually use C, C++ or OpenCL to program FPGAs

Familiar parallel programming concepts apply to FPGA development

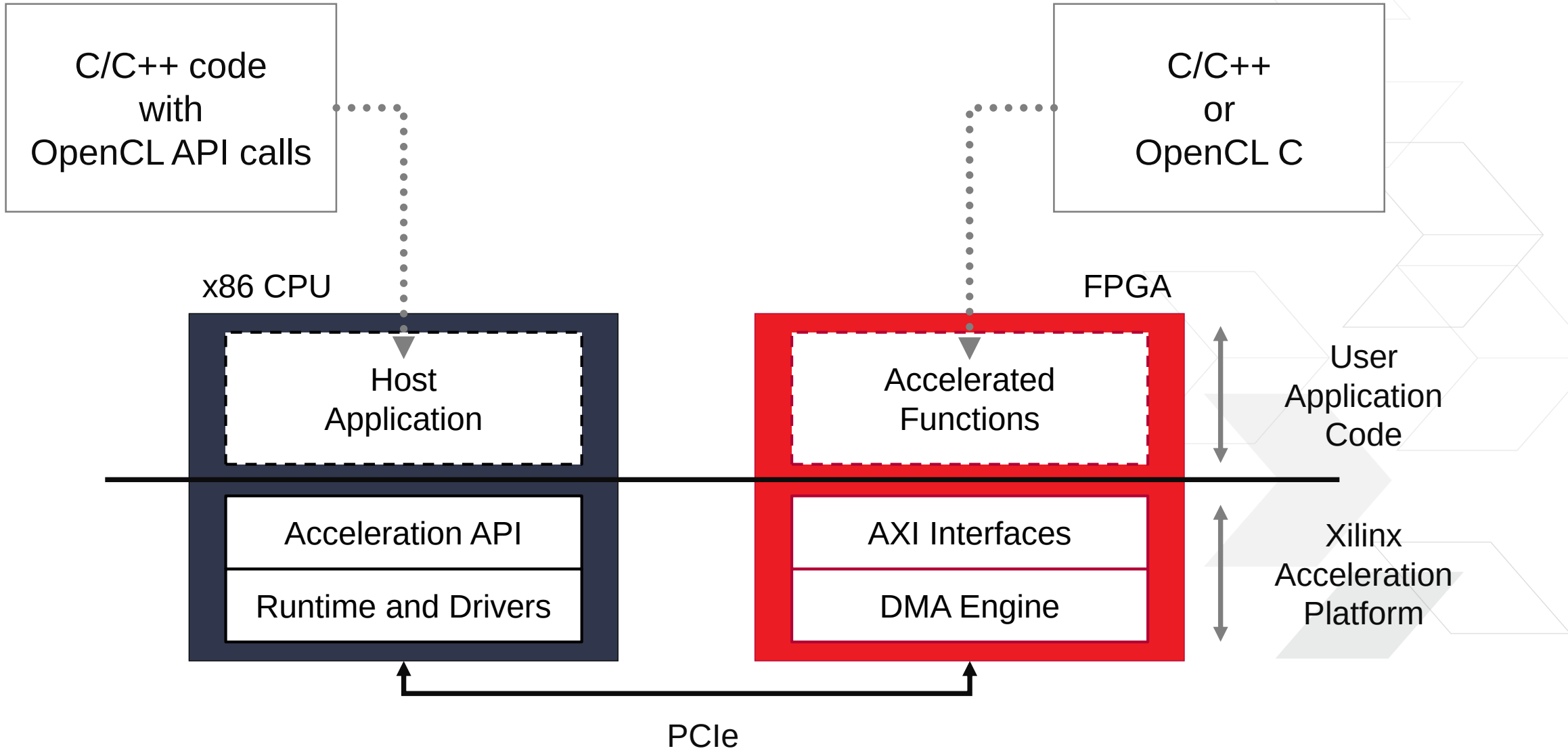
Off-the-shelf platforms make SW acceleration with FPGAs very easy

Off-the-shelf Platforms Simplify FPGA Acceleration



- **No need to develop cards, HW interfaces or SW stacks**
- **Acceleration platforms provides HW and SW APIs to ingrate your code**
- **Focus on the application, not on the implementation details**

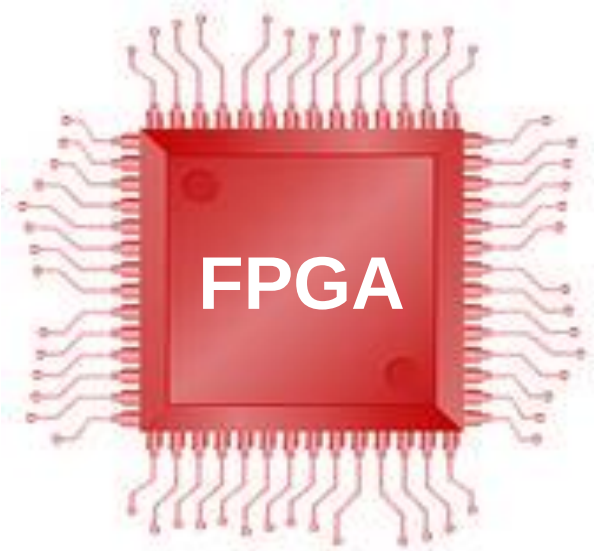
Software Programmability: FPGA Development in C/C++



Using C, C++ or OpenCL to Program FPGAs

```
loop_main:for(int j=0;j<NUM_SIMGROUPS;j+=2) {
  loop_share:for(uint k=0;k<NUM_SIMS;k++) {
    loop_parallel:for(int i=0;i<NUM_RNGS;i++) {
      mt_rng[i].BOX_MULLER(&num1[i][k],&num2[i][k],ratio4,ratio3);
      float payoff1 = expf(num1[i][k])-1.0f;
      float payoff2 = expf(num2[i][k])-1.0f;
      if(num1[i][k]>0.0f)
        pCall1[i][k]+= payoff1;
      else
        pPut1[i][k]-=payoff1;
      if(num2[i][k]>0.0f)
        pCall2[i][k]+=payoff2;
      else
        pPut2[i][k]-=payoff2;
    }
  }
}
```

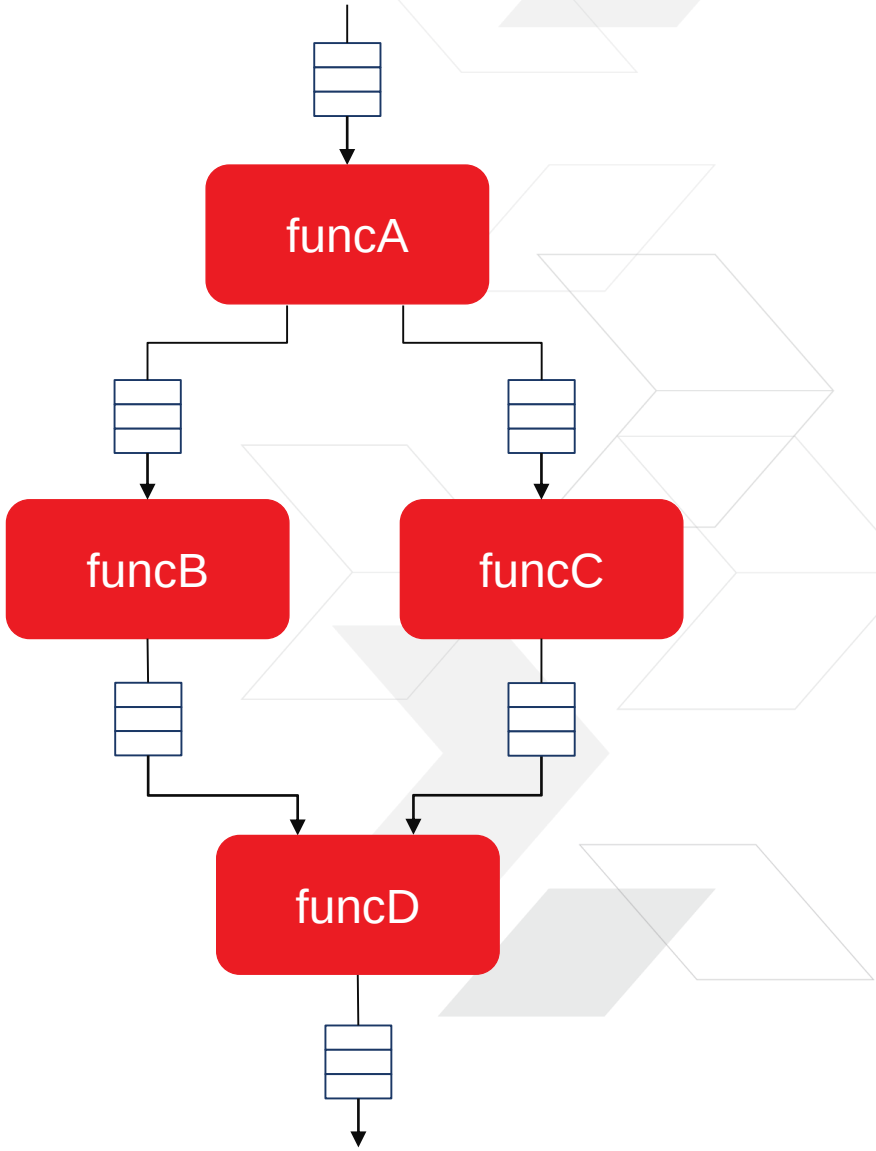
Compile



- Xilinx pioneered C to FPGA compilation technology (aka “HLS”) in 2011
- No need for low-level hardware description languages
- Enables “Software Programmability” of FPGAs

High-Performance FPGA Applications: Think “Parallel”

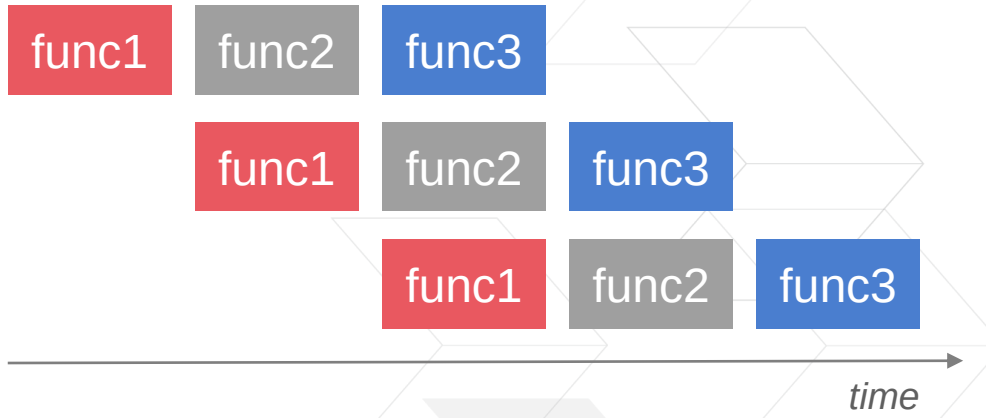
- **Data-level parallelism**
  - Processing different blocks of a data set in parallel
- **Task-level parallelism**
  - Executing different tasks in parallel
  - Executing different tasks in a pipelined fashion
- **Instruction-level parallelism**
  - Parallel instructions (superscalar)
  - Pipelined instructions
- **Bit-level parallelism**
  - Custom word width



Task-Level Parallelism

```
for (int i=0; i<N; i++)
{
  #pragma HLS DATAFLOW
  func1();
  func2();
  func3();
}
```

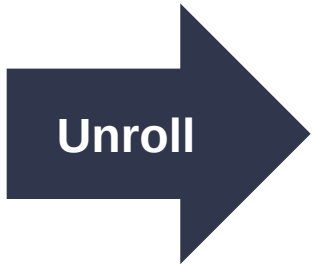
Dataflow

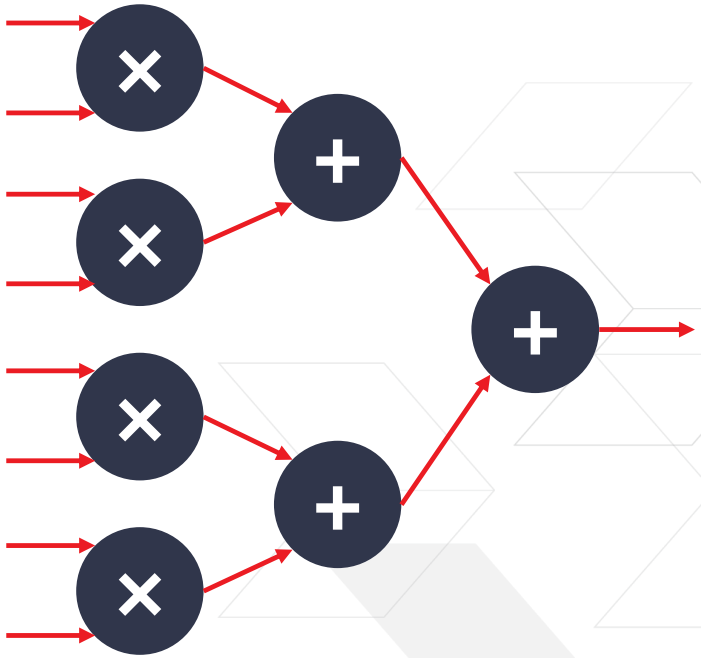


- Create custom dataflow pipelines
- Multiple tasks executing simultaneously
- Streaming programming paradigm

Instruction-Level Parallelism

```
for (int i=0; i<N; i++)
{
  acc += A[i] * B[i];
}
```

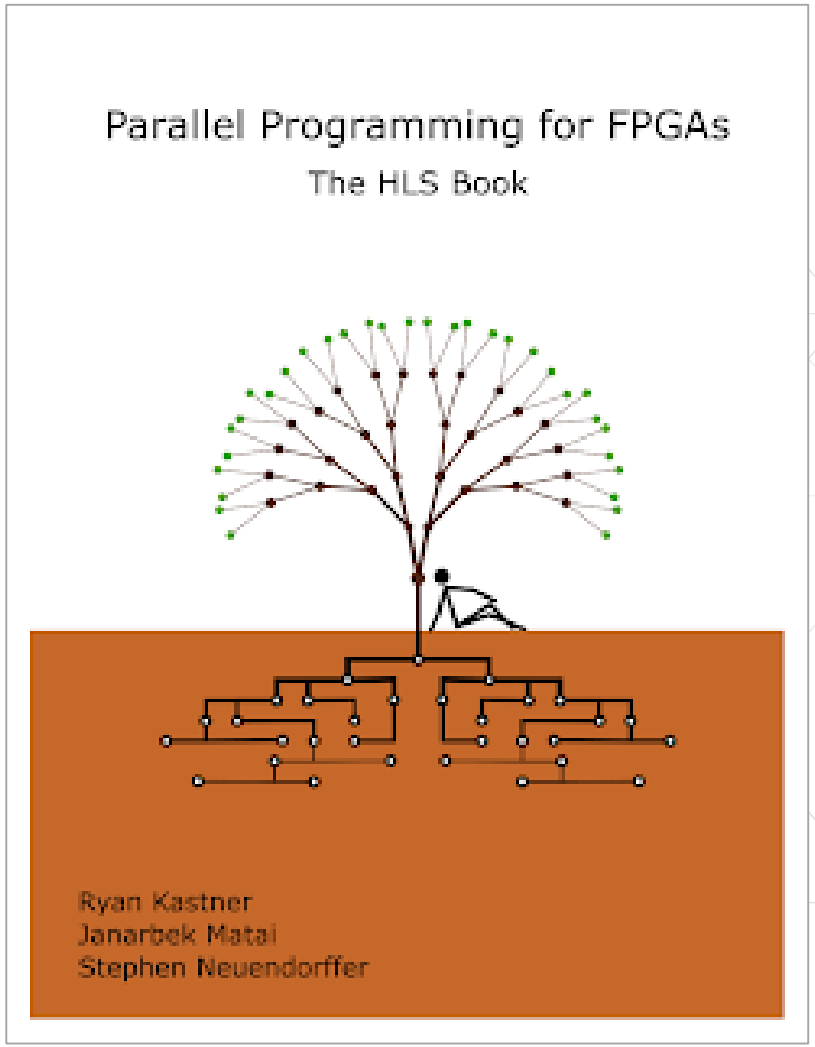




- Create custom datapaths within each task
- Execute hundreds of operations in parallel
- Automatic or user guided compiler optimization

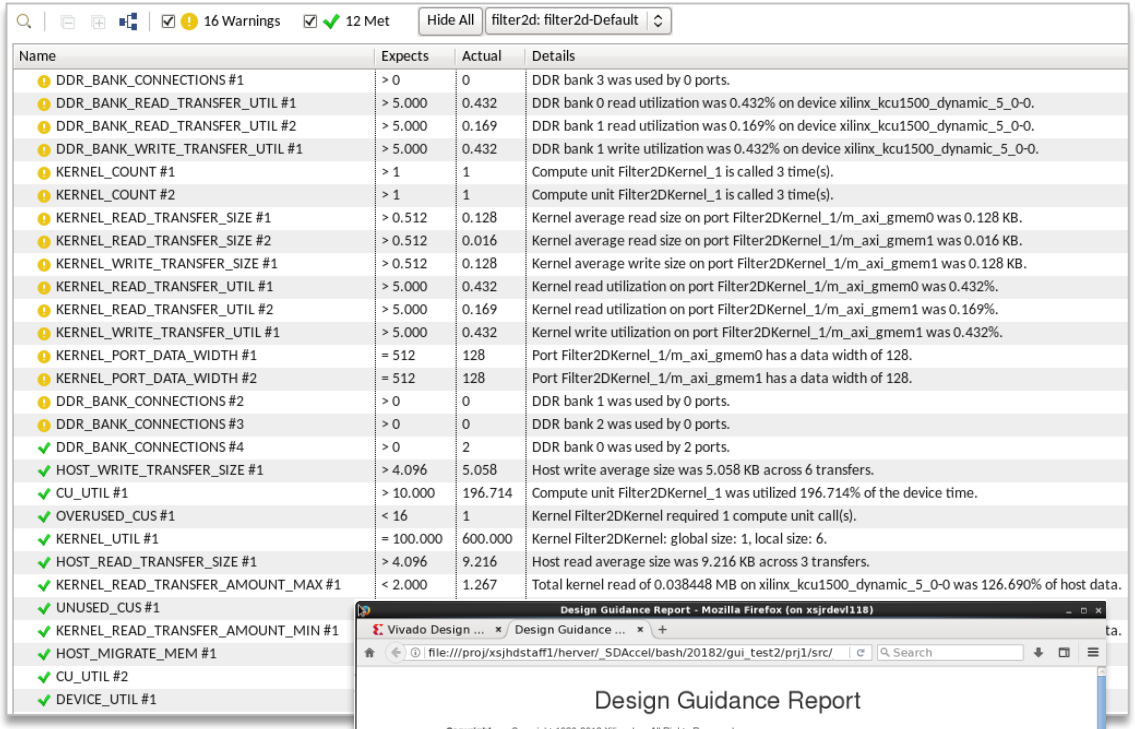
Learn More about Parallel Programming for FPGAs

- **Parallel Programming for FPGAs**
- **Open-source book**
- **Teaches HW and SW developers how to efficiently program FPGAs using high-level synthesis (HLS)**
- **Freely available from: http://hlsbook.ucsd.edu**



Follow the Guide!

- Expert system built-in the FPGA compiler
- Analyzes results; provides actionable feedback on how to improve the design
- Explicit messages, links to detailed explanation and solutions
- Helps developers achieve better results

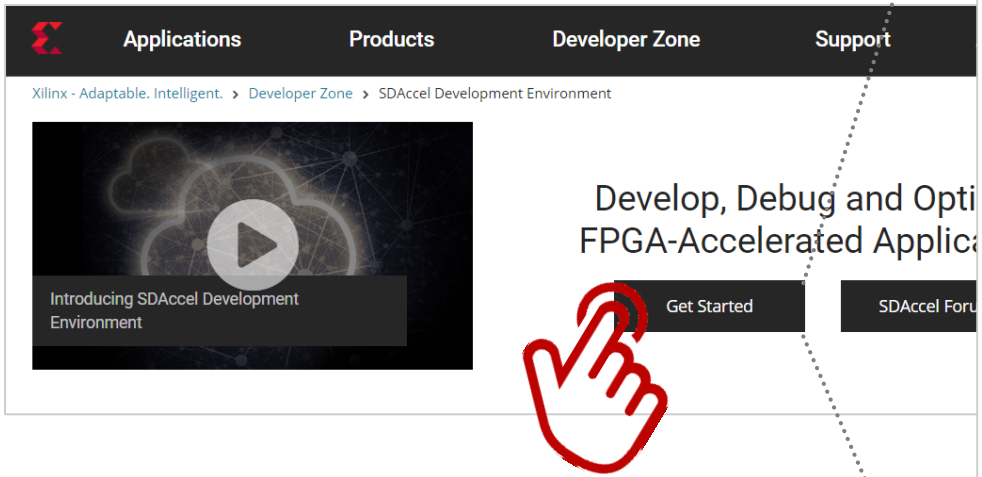


Design Feedback

HTML Report

XILINX.

Visit www.xilinx.com/sdaccel to Get Started !



Getting Started Essentials

- Download SDAccel
- Try a tutorial (Github)
- Browse the collection of SDAccel examples (Github)
- Browse the documentation
- Ask a question, find an answer (Forums)

Get Started in the Cloud with AWS F1

- Run Developer Labs, on-demand
- 5 steps guide to start with C/C++ kernels (Github)
- 5 steps guide to start with RTL kernels (Github)

Get Started in the Cloud with Nimbix

- Try the FPGA Developer Program

Learn and practice how to accelerate applications with FPGAs

XILINX.

Summary - “Different” doesn’t mean “Difficult”

- **FPGA “Software Programmability” is now a reality**
  - Develop FPGA-accelerated applications entirely in C/C++
- **FPGAs excels at high-throughput streaming applications**
  - Think “parallel”, adopt “dataflow” programming model
- **Off-the-shelf platforms make it easy to start and deploy at scale**
  - No need to develop custom card, hardware interfaces or software stacks

Adaptable.
Intelligent.

XILINX