Coarse-Grained FPGA Overlays for On-demand Acceleration of Data Center Workloads

Abhishek Kumar Jain, Douglas L. Maskell, Suhaib A. Fahmy
abhishek013@ntu.edu.sg, asdouglas@ntu.edu.sg, s.fahmy@warwick.ac.uk

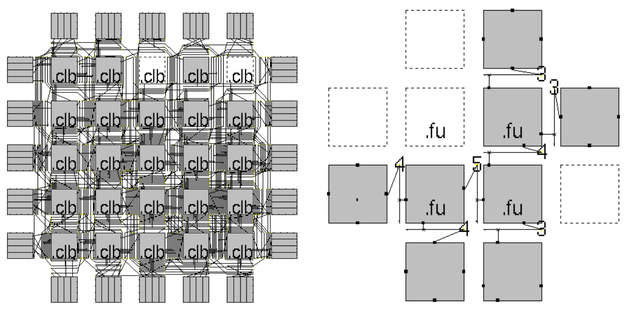
Background and Motivation

Data center workloads such as deep learning kernels and Database query processing:

- Hardware accelerated portions suitable for FPGA resources
- Requirement for dynamic compilation and loading of accelerators

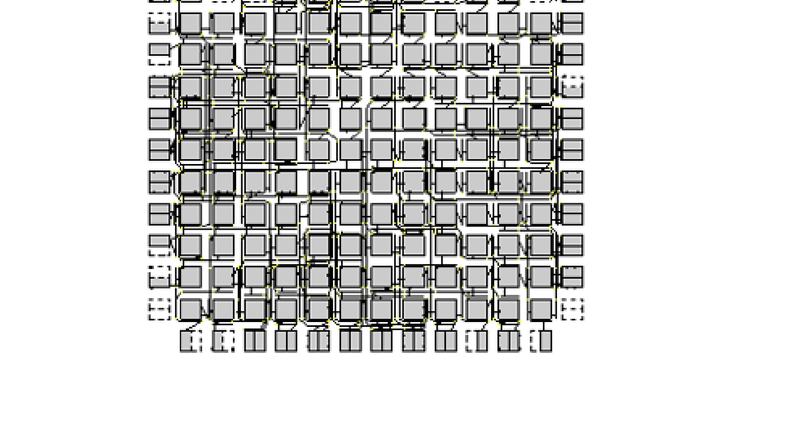
Issues in on-demand acceleration using FPGAs:

- Long compilation times (place and route)
- Long reconfiguration times
- Poor design productivity
- fine-granularity: one of the key issue
- Example: 4 input accumulator



One solution is to use coarse-grained overlays:

- Fast compilation within seconds allowing on-the-fly generation of accelerators on server node
- Dynamic reuse of FPGA resources for multiple workloads
- Fast reconfiguration within microseconds
- Software-like programmability and improved design productivity
- Support for OpenCL programming model, emerging standard for programming heterogeneous computing platforms



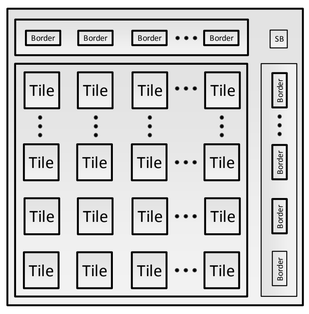
Convolution Engine on Overlay

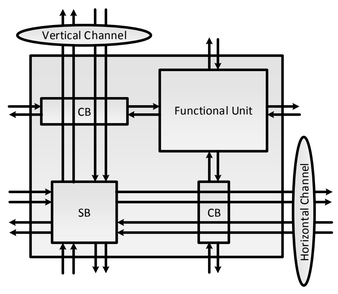
- Mask Size: 3×3
- Evaluating 16 pixels of output image by performing 144 multiplication and 128 addition operations every clock cycle

Coarse-Grained FPGA Overlay

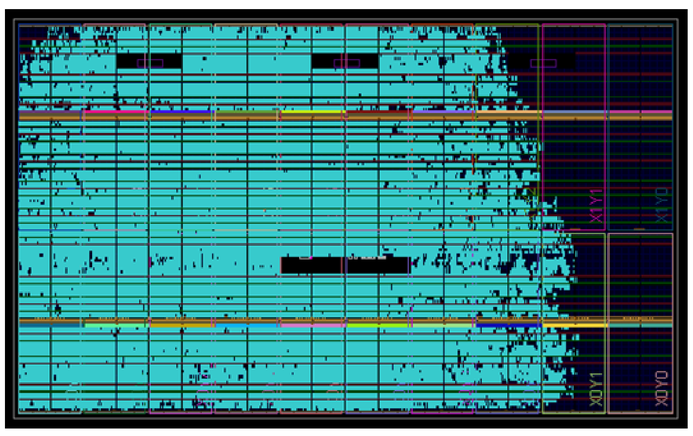
2D array of tiles [1, 2]:

- Programmable functional unit (FU) and routing resources in each tile
- Functional units interconnected via an island-style routing network
- Coarse grained switch boxes, connection boxes and routing channels as programmable routing resources
- Customizable channel width (CW), number of tracks in a routing channel
- A cluster of fully pipelined DSP blocks as a programmable FU
- DSP block based Island-Style Overlay (DISO) presented in FCCM'15 [1]
- Dual-DSP block based Island-Style Overlay (Dual-DISO) presented in DATE'16 [2]





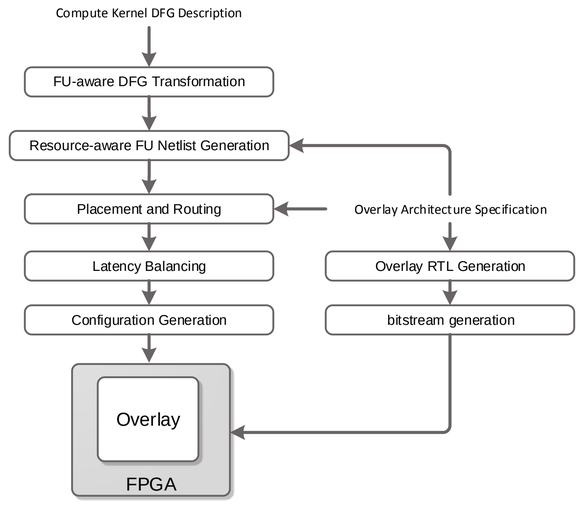
Dual-DISO architecture consisting of 800 general purpose FUs on Virtex-7 FPGA device with a peak performance of 912 GOPS:



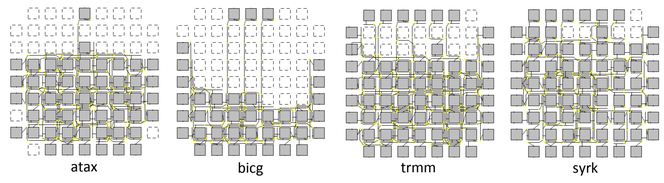
8 KB of configuration data is needed for switching between kernels Can be loaded over PCIe interface, using OpenCL API, taking 50–100μs.

JIT Compilation on Overlay

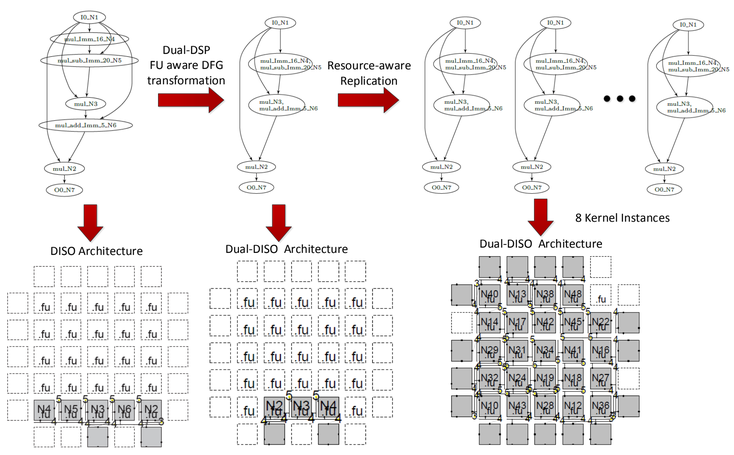
- Uses LLVM front-end, clang, for generating LLVM IR from OpenCL kernels
- Uses LLVM optimization passes for generating optimized LLVM IR
- Generation of data flow graph (DFG) from optimized LLVM IR
- Rest of the process is shown below:



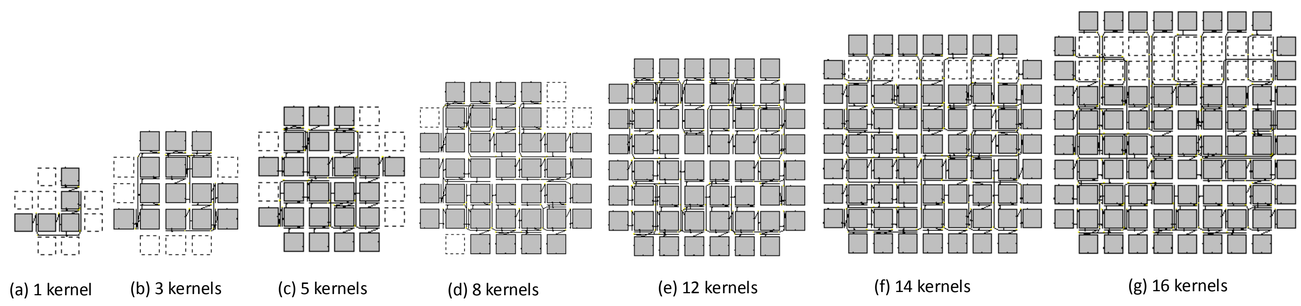
Compilation of linear algebra kernels onto the overlay within a second:



Capabilites of JIT Compiler



Resource-aware Performance Scaling



Future Work

- Development of efficient communication interfaces between host processor and the overlay
- Development of OpenCL drivers and runtime for the overlay

References

[1] A. K. Jain, S. A. Fahmy, and D. L. Maskell, "Efficient Overlay architecture based on DSP blocks," in *FCCM*, 2015.

[2] A. K. Jain, D. L. Maskell, and S. A. Fahmy, "Throughput oriented FPGA overlays using DSP blocks," in *DATE*, 2016.