Architecture Centric Overlays for Abstraction and Performance

Suhaib A Fahmy

School of Engineering
University of Warwick

s.fahmy@warwick.ac.uk

Overlays Today

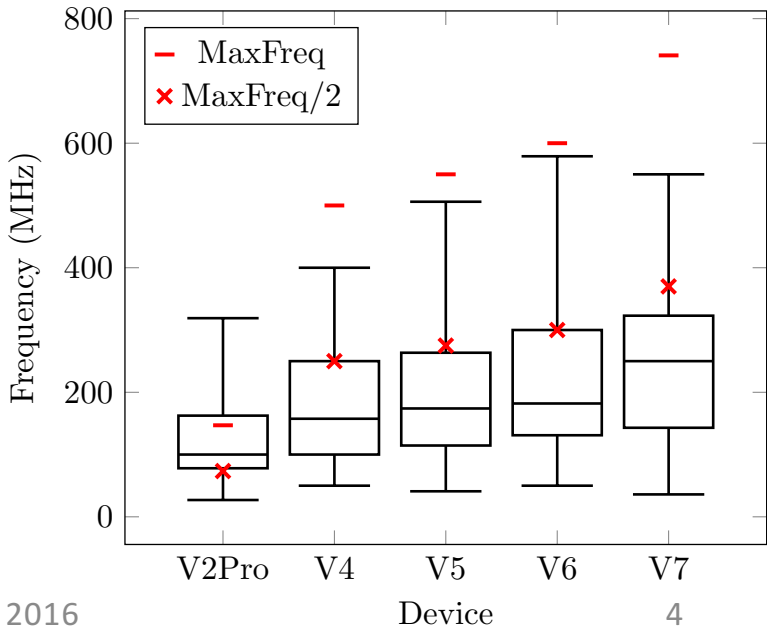
- Overlays can address key FPGA challenges:
  - High level design – *not through RTL*
  - Simplified toolflow – *working at a higher abstraction*
  - Virtualised view of resources
  - Simplified reconfiguration – *much smaller bitstreams*
- However they entail costs
  - Generality entails extra area cost for compute
  - Routing signals flexibly is expensive
  - Significant margin below fabric frequency capability

Why not ASIC CGRAs?

- Significant previous work on coarse grained arrays
- General purpose word-wide arithmetic processing elements and routing
- Some commercial hints (Samsung)
- ASIC entails
  - Fixed functional unit design – lack of flexibility for new applications/changes in requirements
  - Fixed routing infrastructure – leads to over-provision and high routing overhead
  - How do you scale an ASIC to many different deployment scenarios? – cost!

FPGA Architecture Evolution

- More hardened blocks on modern FPGAs: DSP block, block memory, floating point blocks
- Non-standard functions – synthesis limitations
- Our designs not keeping up with architectural improvements
- We are still breaking things down and hoping the synthesis tools can work out how to best use these resources



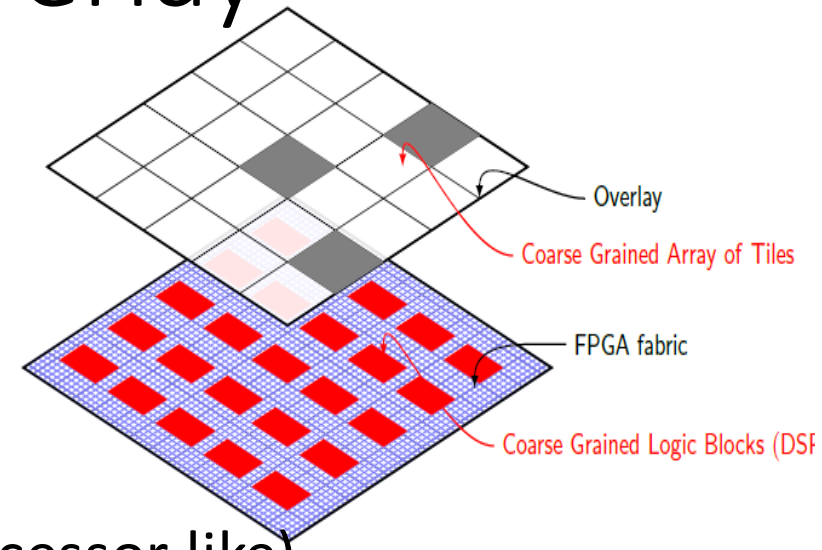
Some Waypoints

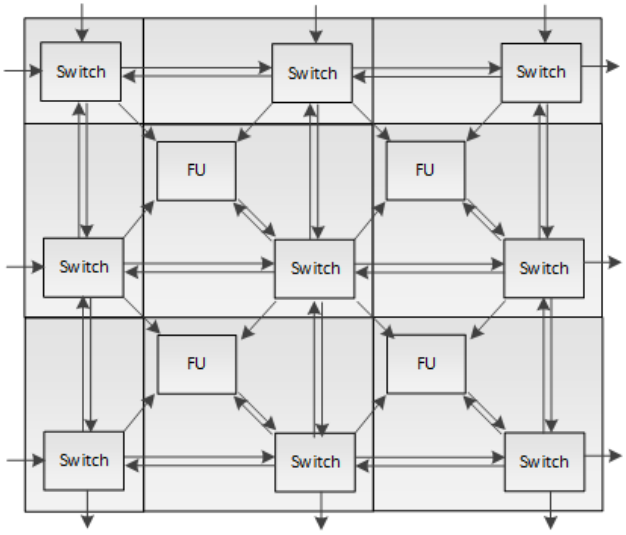
- Mapping to DSP Blocks
  - Template of full speed configurations, and a high level mapping tool to stitch them together (TCAD 2016)
- iDEA
  - A fully functional soft processor using a DSP block and a few LUTs/FFs (FPT 2012, TRETS 2014)
- DySER Adaptation
  - Swap the FU for a flexible DSP block: 25% area improvement, 2.5× frequency! (HEART 2015)
- Hoplite and GRVI Phalanx
  - Exploit low-level logic block features to build a tiny, fast NoC (and soft processor) (FPL 2015, FCCM 2016)

How do we reconcile low-level optimisation with high-level design?

Components of an Overlay

- Functional Units (FUs/PEs)
  - Homogenous or heterogeneous
  - Single or multiple-function
  - Typically ALU-like
  - Can have their own memory (processor like)
- Routing Fabric/Interconnect
  - Word-wide flexible routing
  - Circuit switched or NoC
  - Offers the generality in connecting PEs



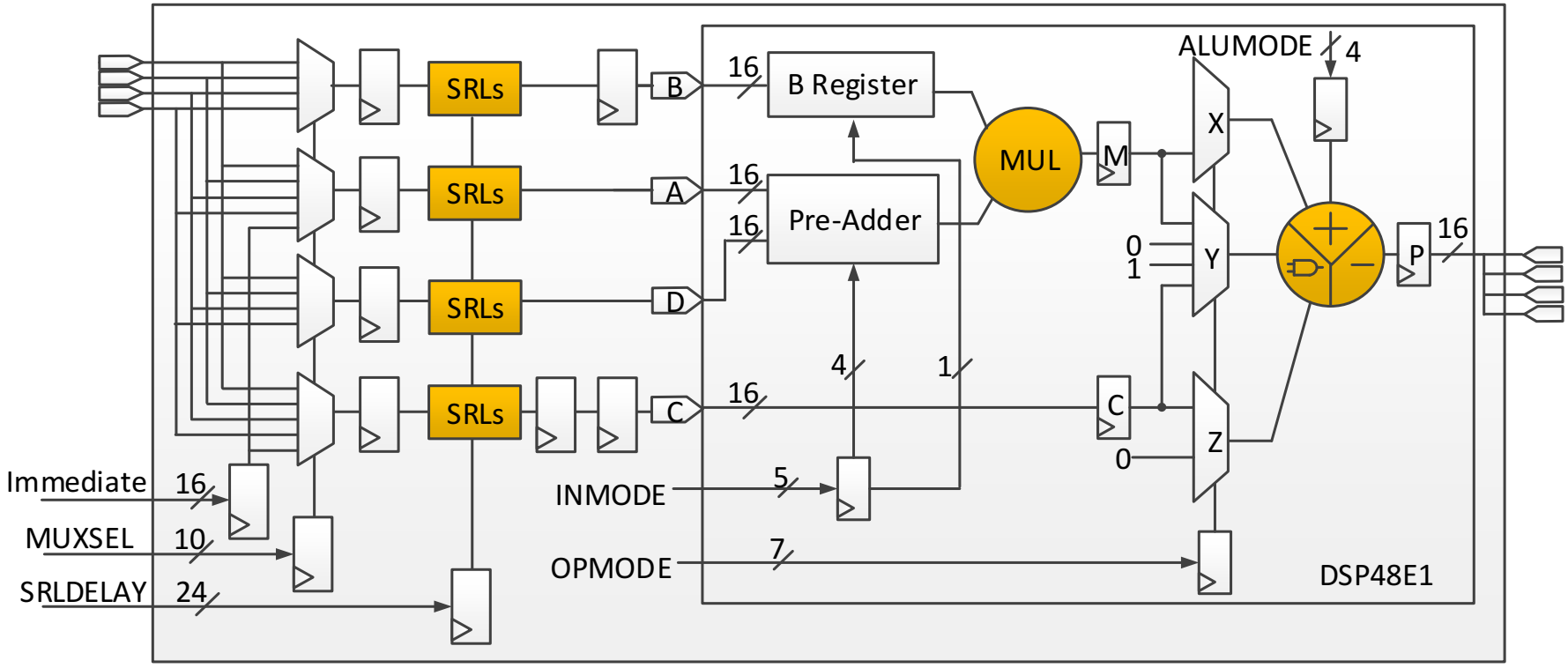


Overlays

- FUs/PEs designed by hand to extract maximum performance from minimal area
- Routing fabric/interconnect designed to offer flexible routing of words, with consideration for architecture
- On FPGAs, the ability to modify overlay design (even at runtime) to suit application requirements

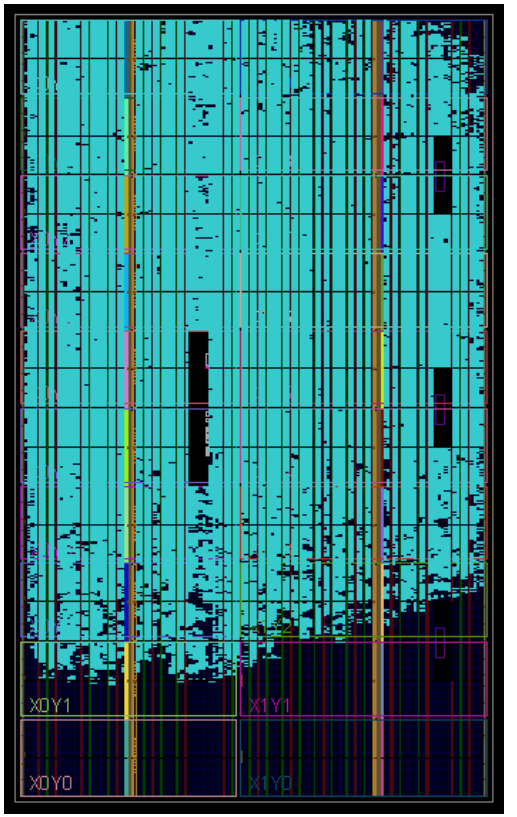
Overlay Functional Units

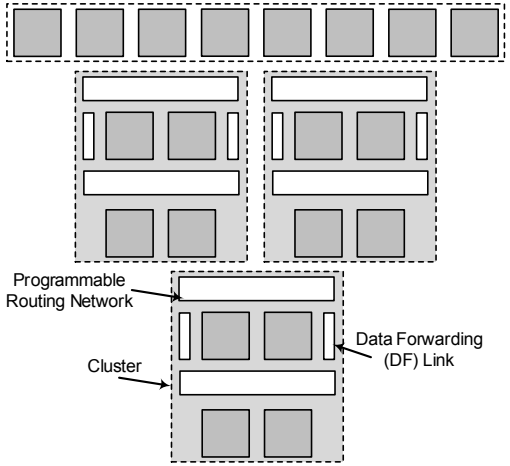
- DSP Block overlay using Xilinx DSP block dynamic programmability



More Waypoints

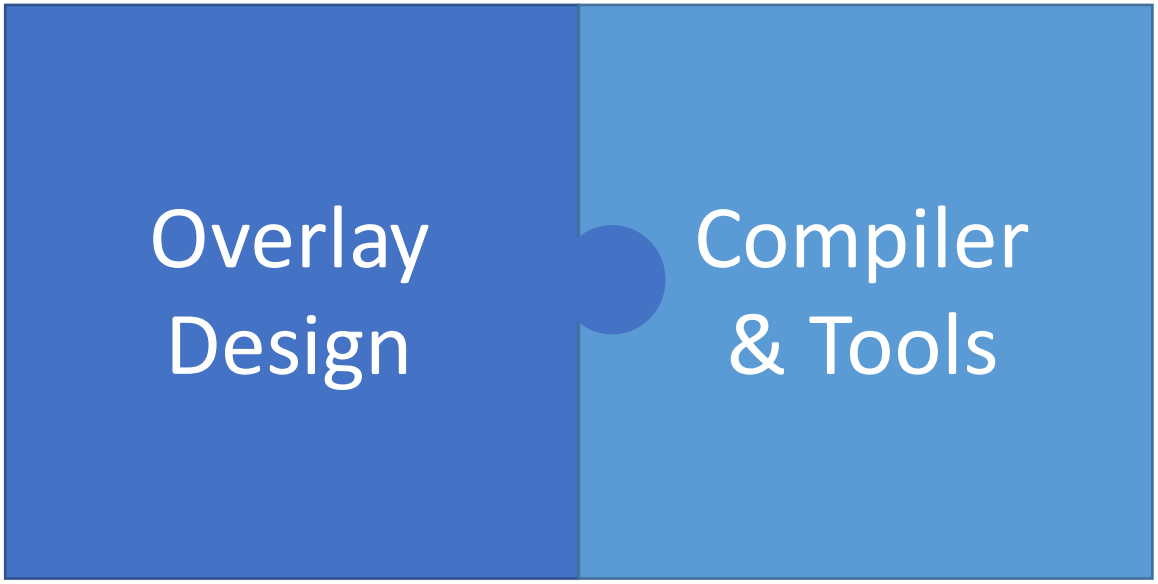
- Large overlay: 800 DSP blocks at 380MHz
- DeCO datapath overlay:
  - Optimised for feed-forward dataflow graphs
  - Simpler interconnect for lower routing overhead
  - While optimised for a set of algorithms, supports others





The Two Pieces

- A good overlay must target a specific type of problem, and the mapping tools must be able to operate at that functional level.



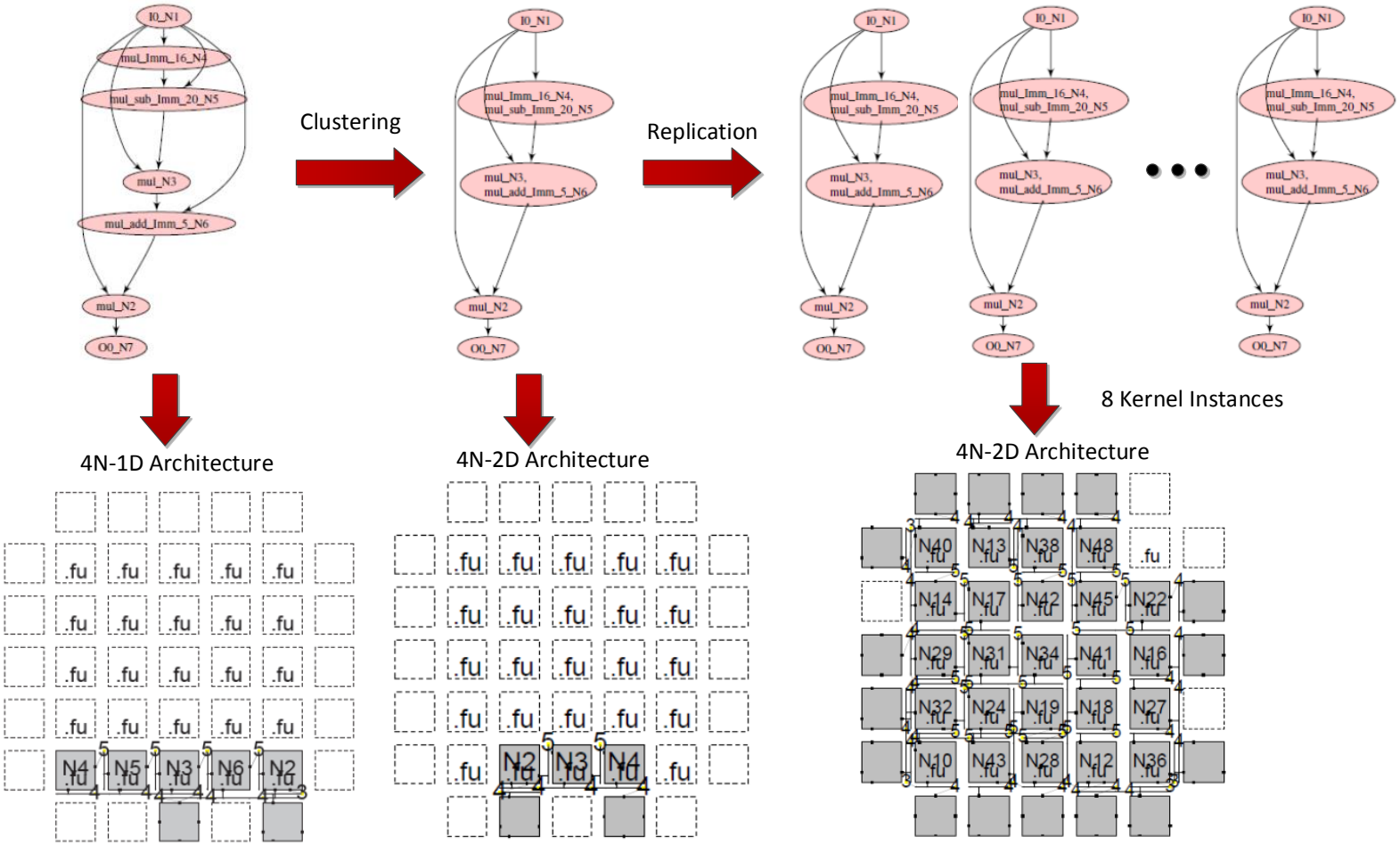
Domain Specific Overlays

- Tap into DSL compiler infrastructure to extract library of basic blocks
- Map individually to FPGA components with manual optimisation
- Implementation tools now chain these together
  - Recall our DSP block template mapping tool
- Can build an overlay using these blocks with flexible enough routing to support range of applications

On-Chip Just-In-Time

- Simpler backend flow can run on embedded processor, e.g. ARM in Zynq in under a second
- Processor is aware of the state of the overlay at any point in time
- Dynamic allocation of resources to tasks, including placement and routing on the overlay
- Virtualised in terms of locations and specific bitstream
- Even new kernels can be compiled and mapped at runtime

On-Chip Just-In-Time



Virtualised Cloud Accelerators

- Multi-user virtualised view of FPGAs is cumbersome:
  - Require partial reconfiguration to fix interfaces
  - A priori partitioning is inefficient
  - Relocatable bitstreams unsupported
  - Slow reconfiguration, even with optimised PR (milliseconds)
- Overlay manageable at finer granularity, faster reconfiguration
- Generally targetting a specific domain through abstracted API
  - "Specialised" overlays ideal – Google TPU

Related Publications

- A. K. Jain, D. L. Maskell, and S. A. Fahmy, "DeCO: A DSP Block Based FPGA Accelerator Overlay With Low Overhead Interconnect", in Proceedings FCCM 2016.
- B. Ronak and S. A. Fahmy, "Mapping for Maximum Performance on FPGA DSP Blocks", in IEEE TCAD 2016.
- A. K. Jain, D. L. Maskell, and S. A. Fahmy, "Throughput Oriented FPGA Overlays Using DSP Blocks", in Proceedings DATE 2016.
- S. A. Fahmy, K. Vipin, and S. Shreejith, "Virtualized FPGA Accelerators for Efficient Cloud Computing" in Proceedings CloudCom 2015.
- A. K. Jain, X. Li, S. A. Fahmy, and D. L. Maskell, "Adapting the DySER Architecture with DSP Blocks as an Overlay for the Xilinx Zynq", in Proceedings HEART 2015.
- A. K. Jain, K. D. Pham, J. Cui, S. A. Fahmy, and D. L. Maskell, "Virtualized Execution and Management of Hardware Tasks on a Hybrid ARM-FPGA Platform", in JSPS 2014.
- H. Y. Cheah, F. Brosser, S. A. Fahmy, and D. L. Maskell, "The iDEA DSP Block Based Soft Processor for FPGAs", in ACM TRETS 2014.