Are Coarse-Grained Overlays Ready for General Purpose Application Acceleration on FPGAs?

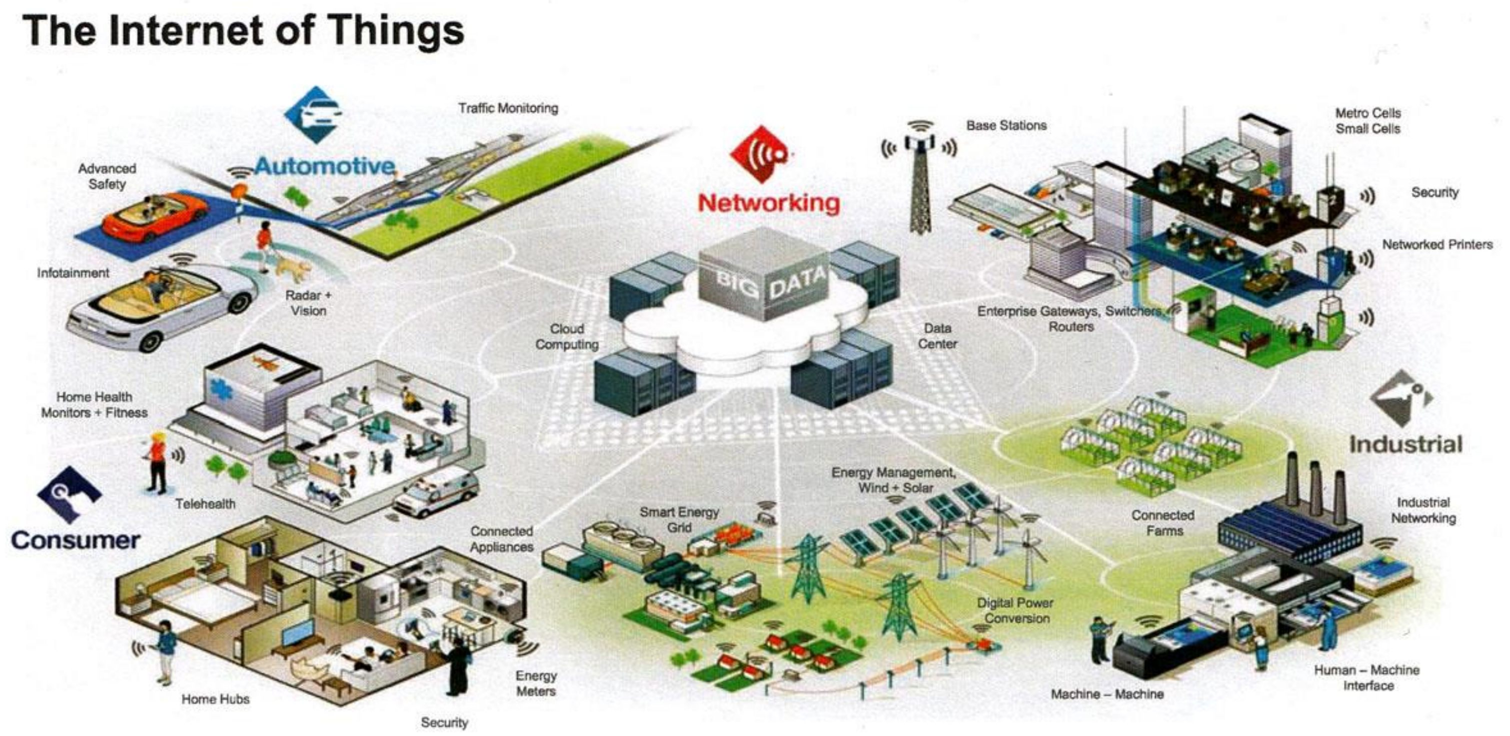
Abhishek Kumar Jain, **Douglas L. Maskell**
School of Computer Science and Engineering
Nanyang Technological University (NTU), Singapore

Suhaib A. Fahmy
School of Engineering
University of Warwick, UK

International Conference on Pervasive Intelligence and Computing, 8th August 2016, Auckland, New Zealand

Internet of Things (IoT)

- Diverse range of computing requirements
- Central Focus: Adaptability and low power computing
- Possible Solution: FPGAs coupled with high performance CPUs



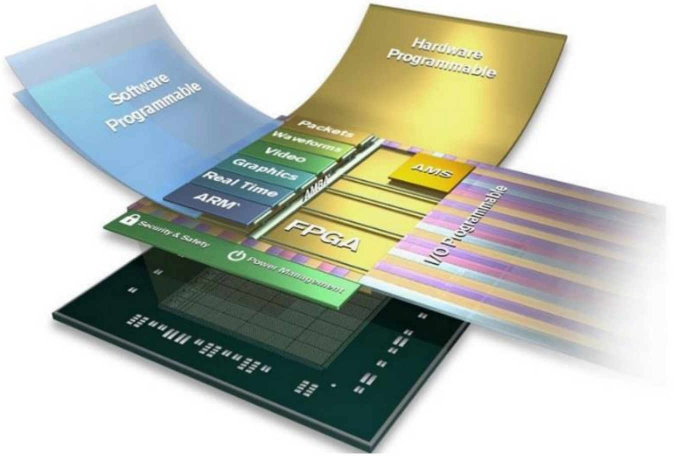
https://opentechdiary.wordpress.com/tag/internet-of-things

FPGAs in Heterogeneous Computing Platforms

- Integration of FPGAs in Heterogeneous computing platforms
  - Massive parallelism
  - Low power consumption
  - Customizability

- Example Platforms from Intel and Xilinx
  - Xilinx UltraScale+ MPSoC, Intel Broadwell + Arria 10

- Xilinx UltraScale+ MPSoC
  - Quad-core ARM Cortex-A53
  - Dual-core ARM Cortex-R5
  - ARM Mali-400 GPU
  - High performance FPGA fabric



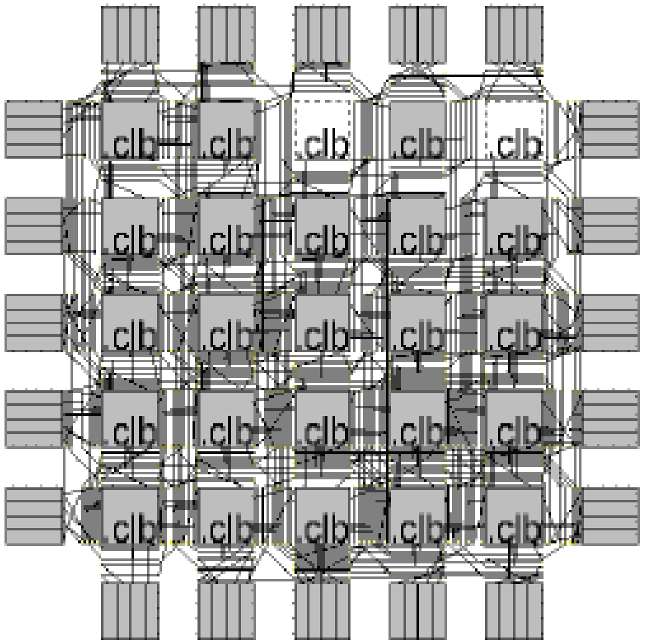
Are FPGAs in these platforms ready for the mainstream?

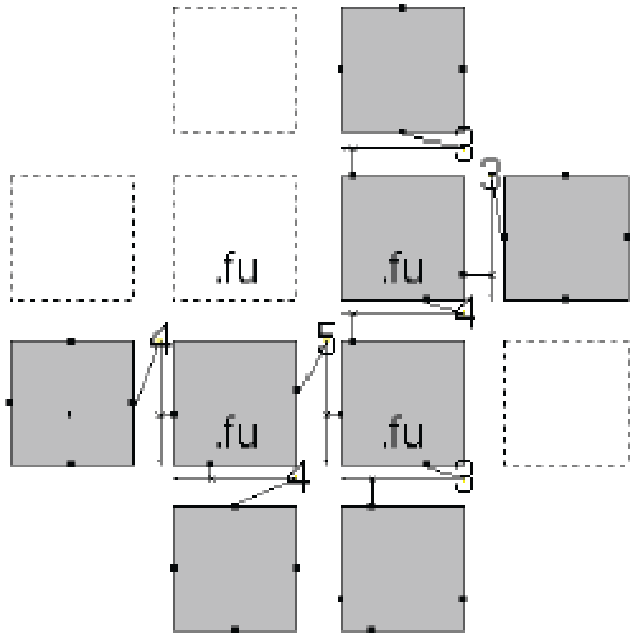
- No, Mainly due to poor design productivity issues
  - Accelerator design at RTL level
  - Hardware design expertise
  - Long compilation times of RTL design

```
module kernel(a,b,c,d,out);
input[15:0] a,b,c,d;
output[15:0] out;

assign out = a + b + c + d;

endmodule
```

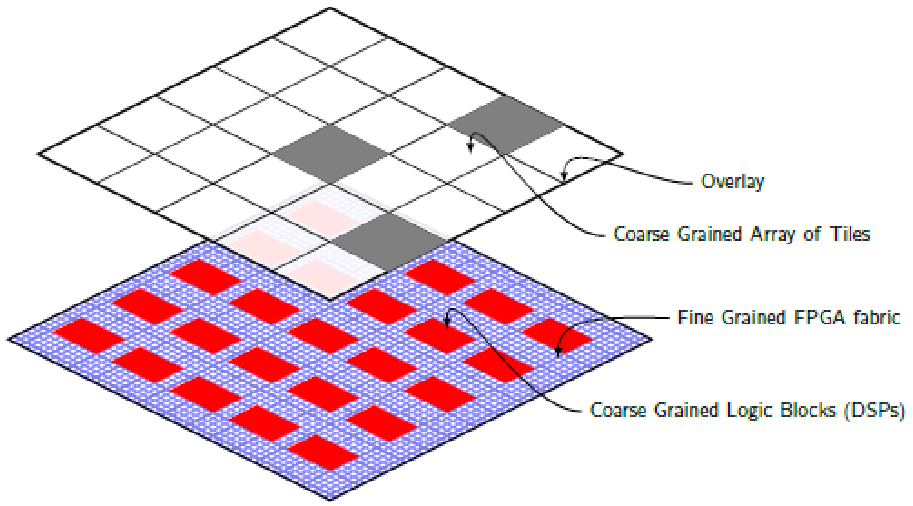




Coarse grained FPGA overlays

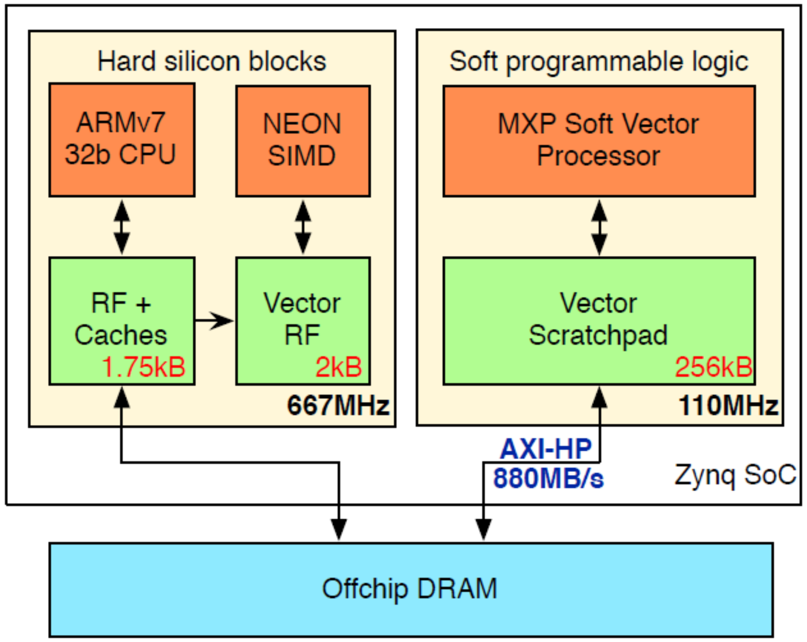
- Array of coarse-grained tiles
- Programmable functional unit and interconnect resources

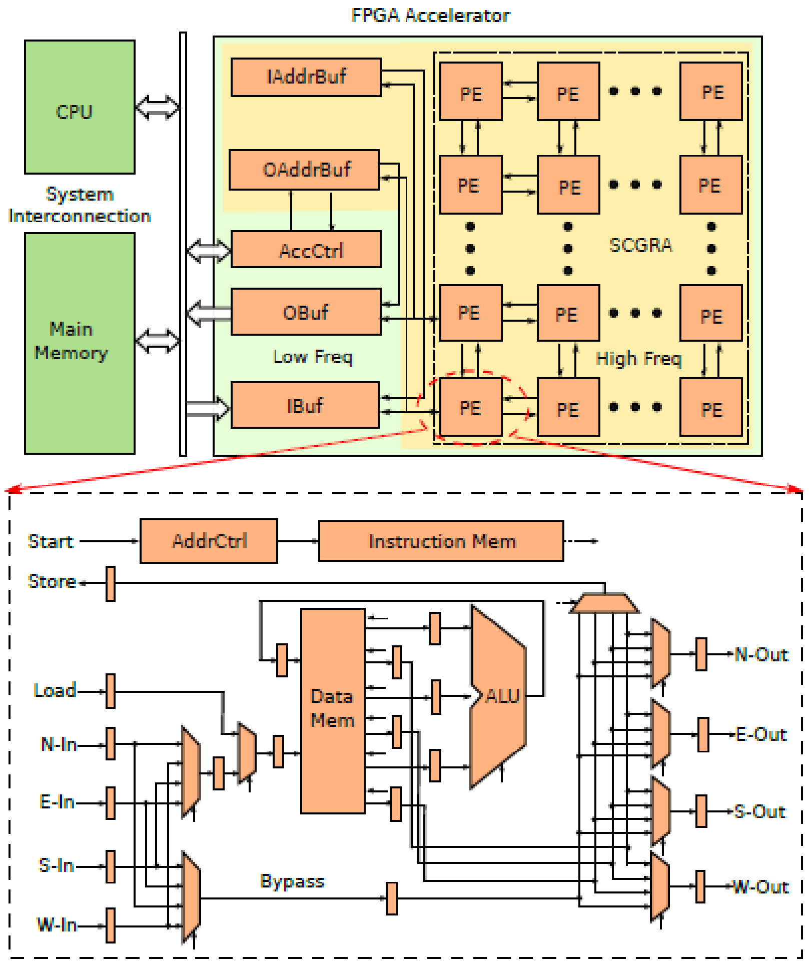
- Benefits:
  - Accelerator design at a higher level of abstraction
  - Fast compilation
  - Fast reconfiguration
  - Improved design productivity


Overlay
Coarse Grained Array of Tiles
Fine Grained FPGA fabric
Coarse Grained Logic Blocks (DSPs)

Time-multiplexed Overlays

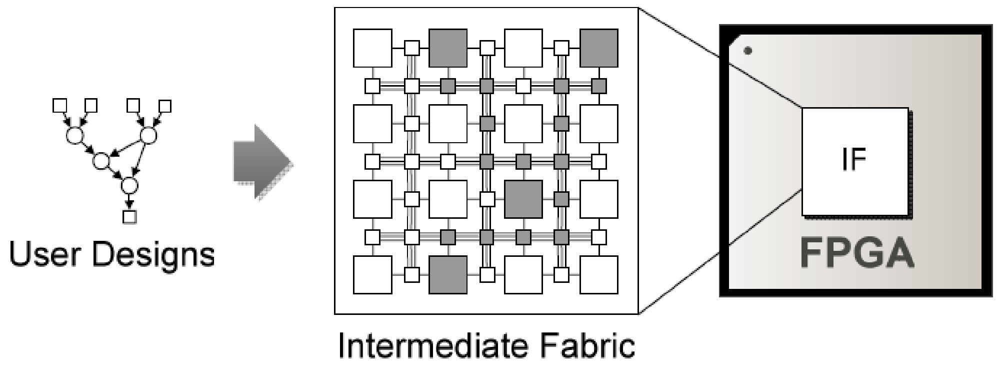
- Less area requirement at the cost of reduced throughput
- Examples:
    - VectorBlox MXP
    - SCGRA

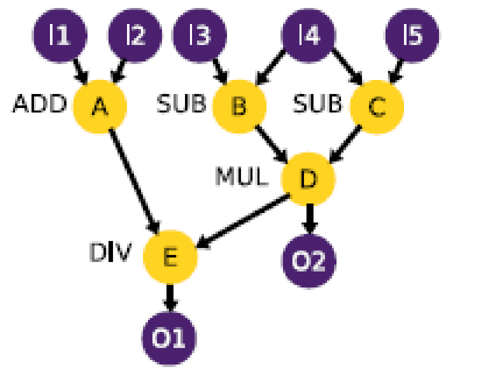


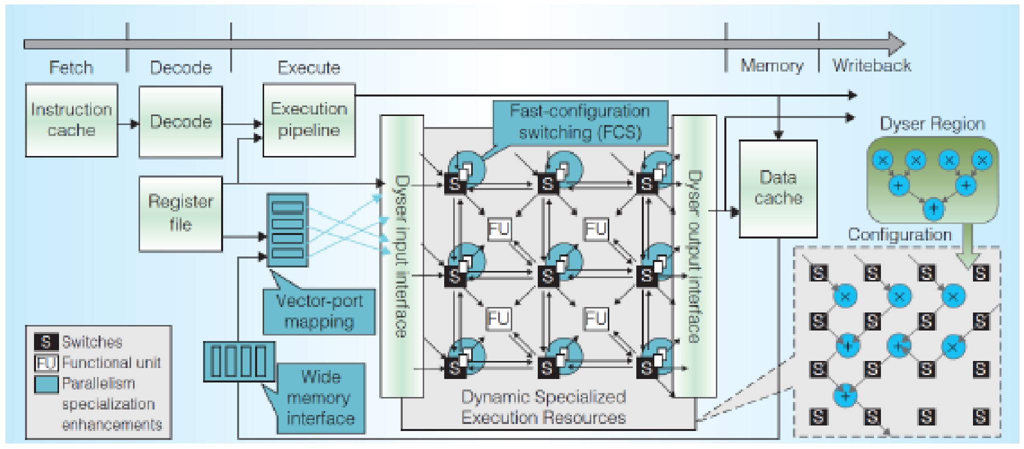


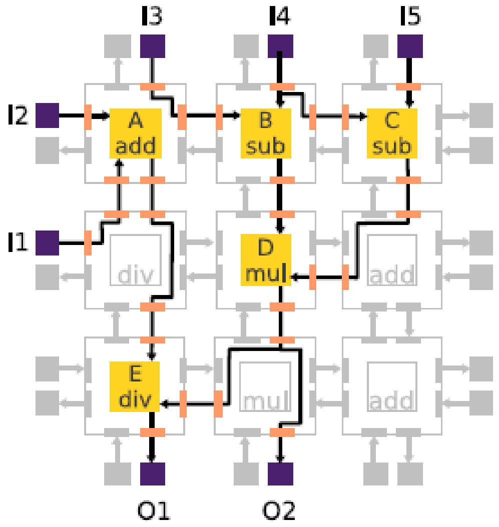
Spatially-configured Overlays

- Maximum performance (throughput) at the cost of area overheads
- Examples: Intermediate Fabrics, DySER, Mesh-of-FU Overlay



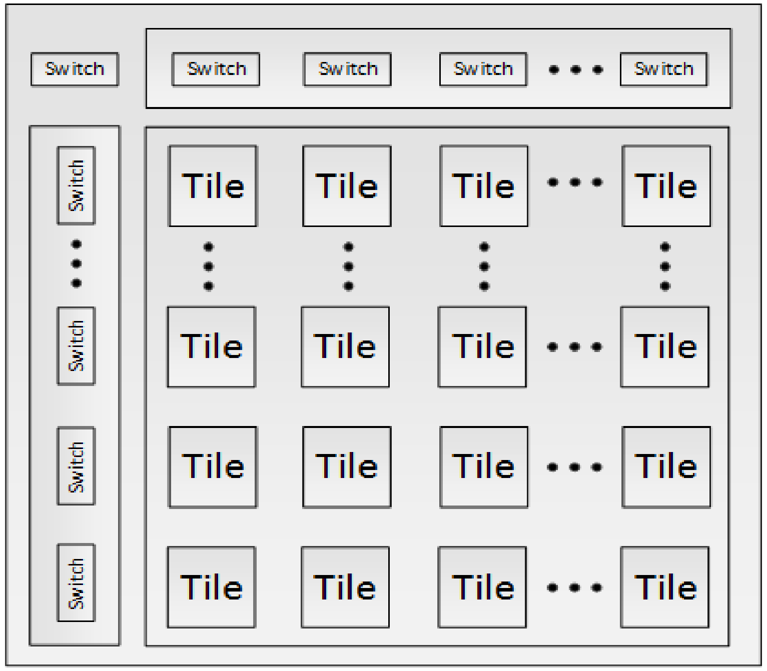


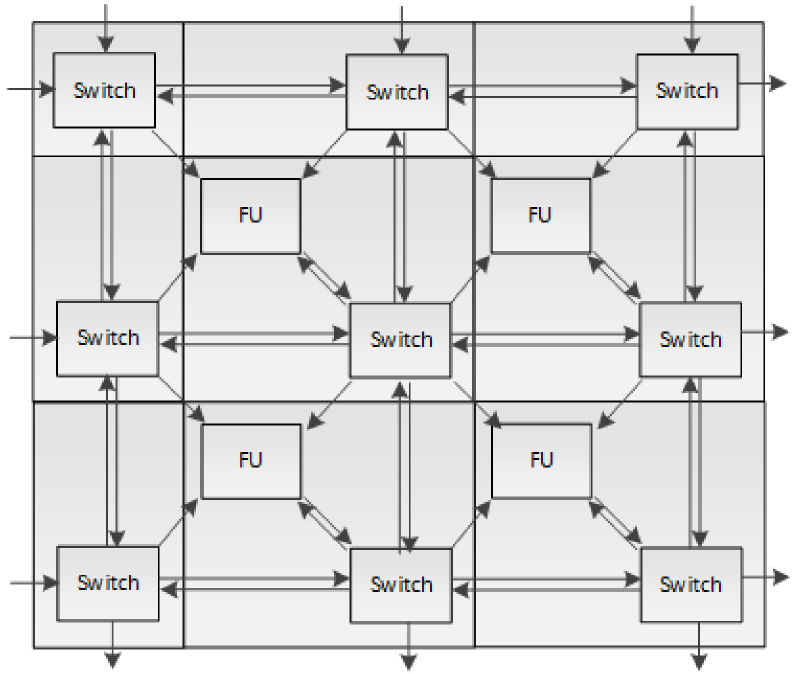




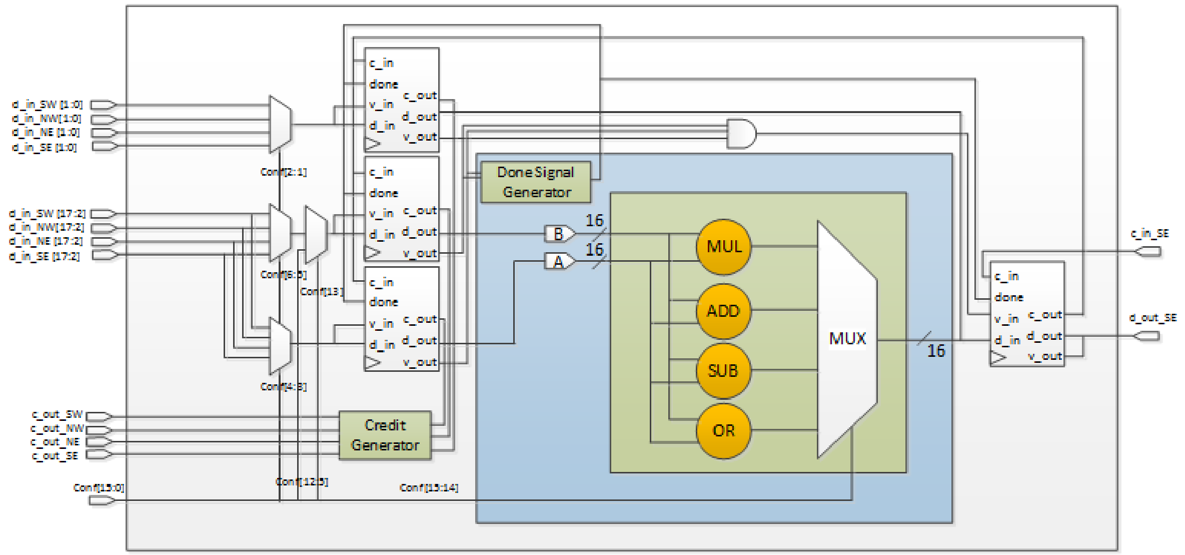
DySER Architecture

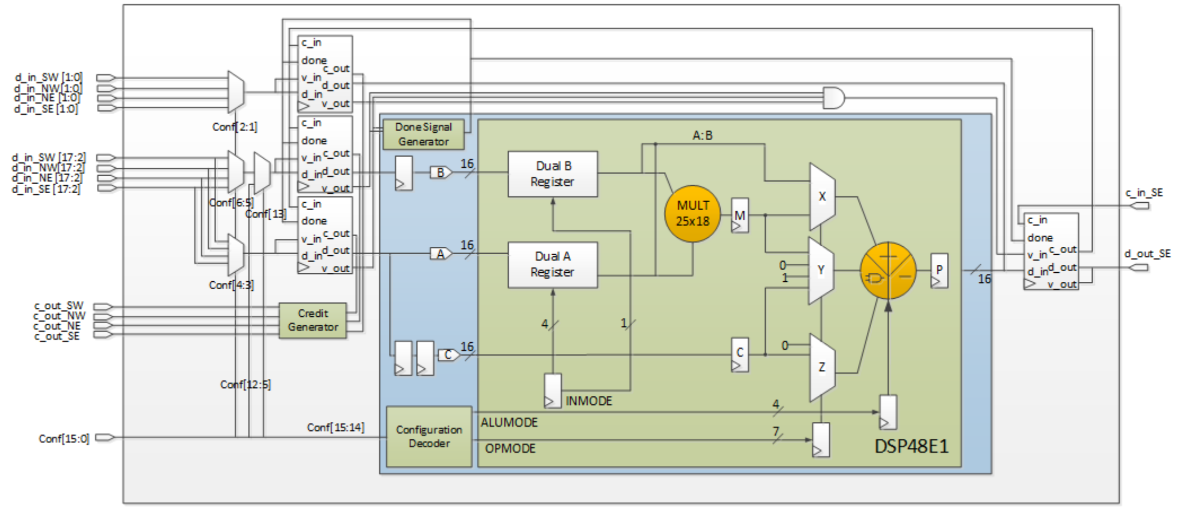
- Array of coarse-grained tiles
- Programmable functional unit and interconnect resources
- Highly flexible interconnect architecture + FU un-optimized for FPGA implementation -> Area and Performance overheads

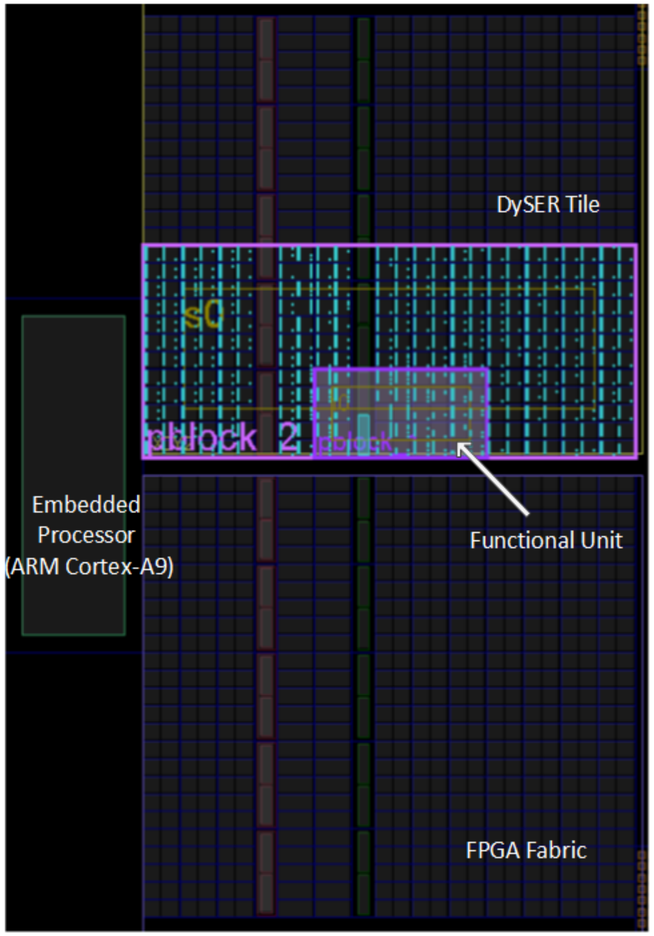




FPGA Architecture-focused FU

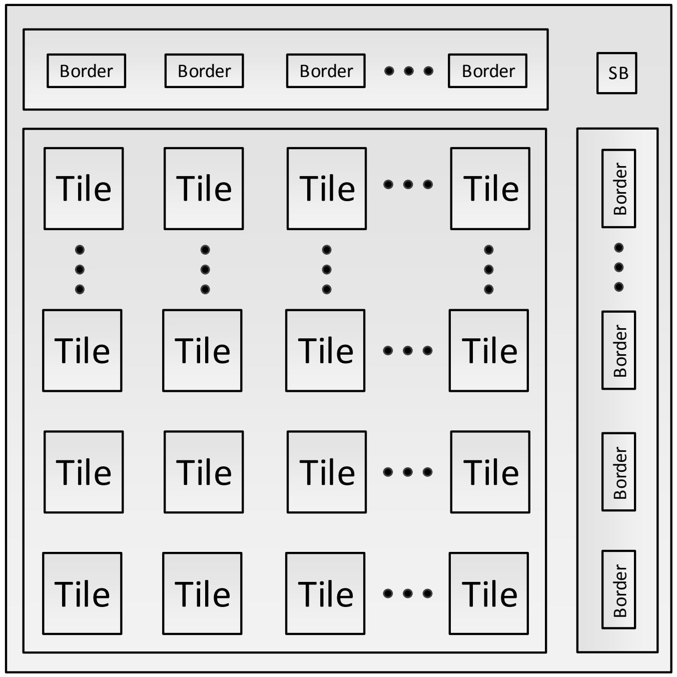


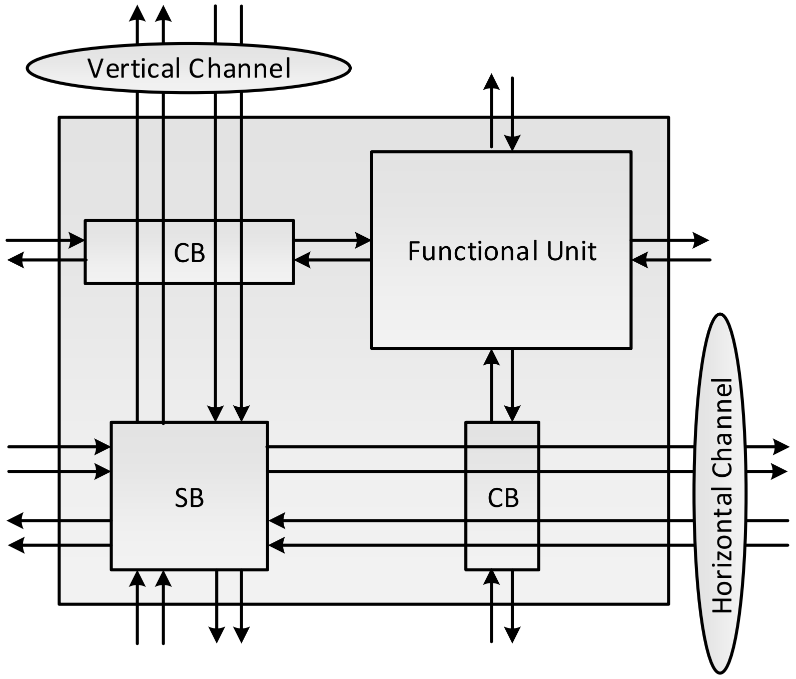




DSP Block based Island-Style Overlay (DISO)

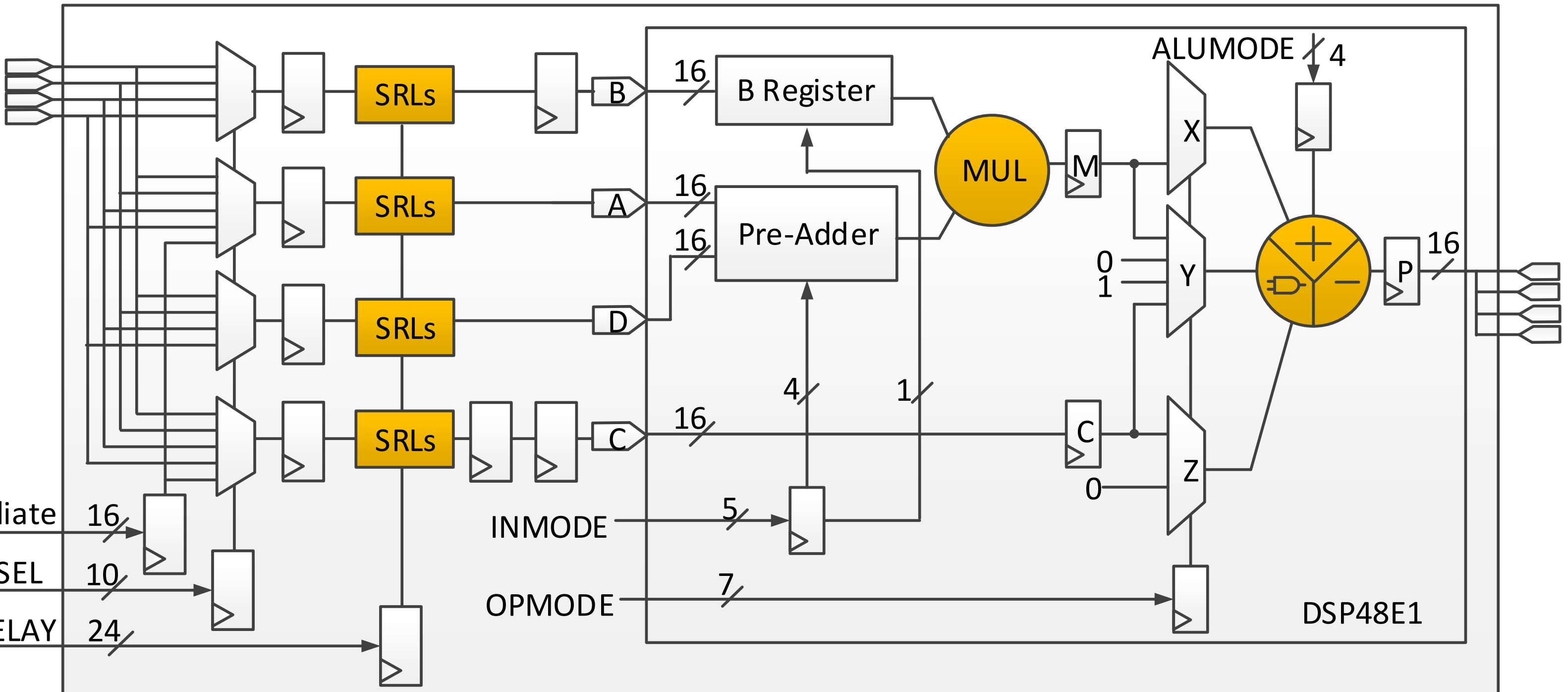
- Less flexible interconnect architecture compared to DySER
- Fully pipelined DSP block based FU
- Optimized for FPGA implementation
- Achievable frequency near theoretical limits



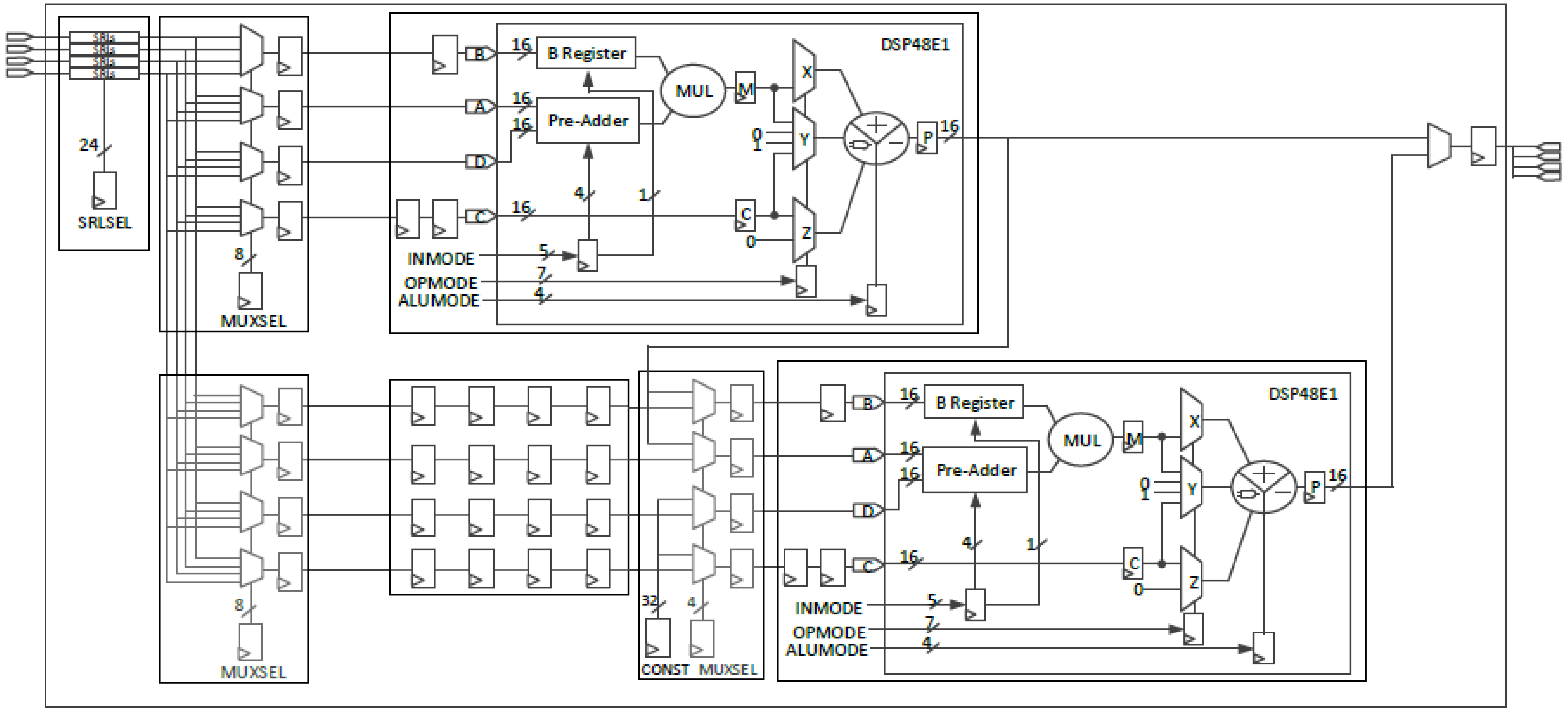


DSP Block based Island-Style Overlay (DISO)

- Add/Sub, a multiplier and an ALU within FU
- 400 MHz on the Xilinx Zynq device
- MUX based reordering logic
- SRL based variable-length shift registers

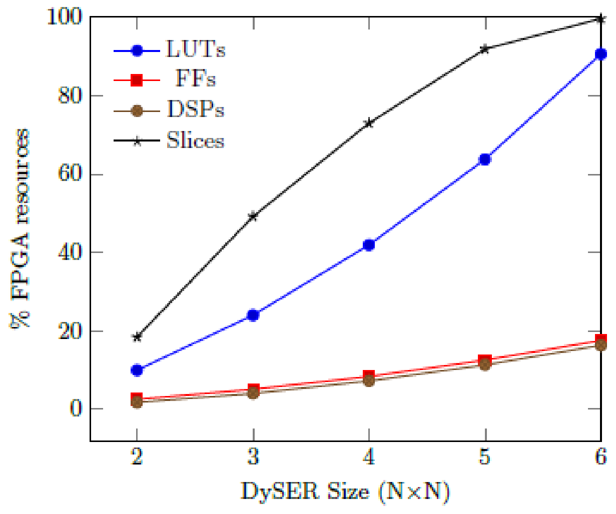


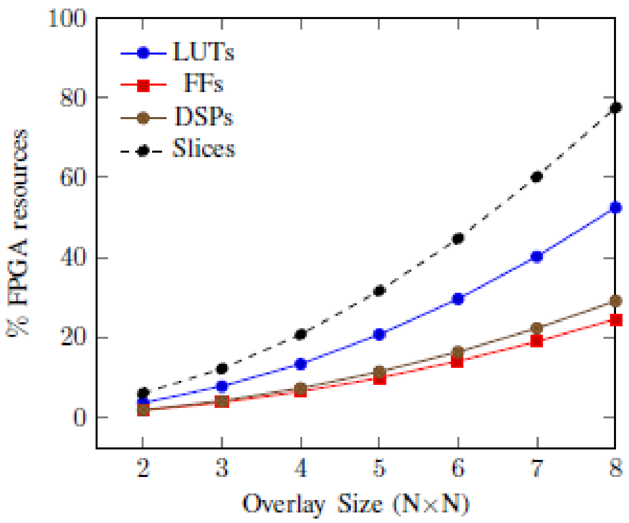
Dual-DSP Block based Overlay (Dual-DISO)

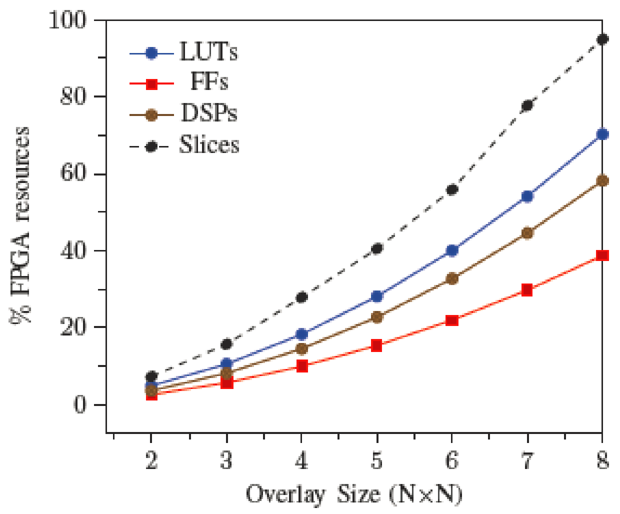


Mapping Overlays onto Zynq

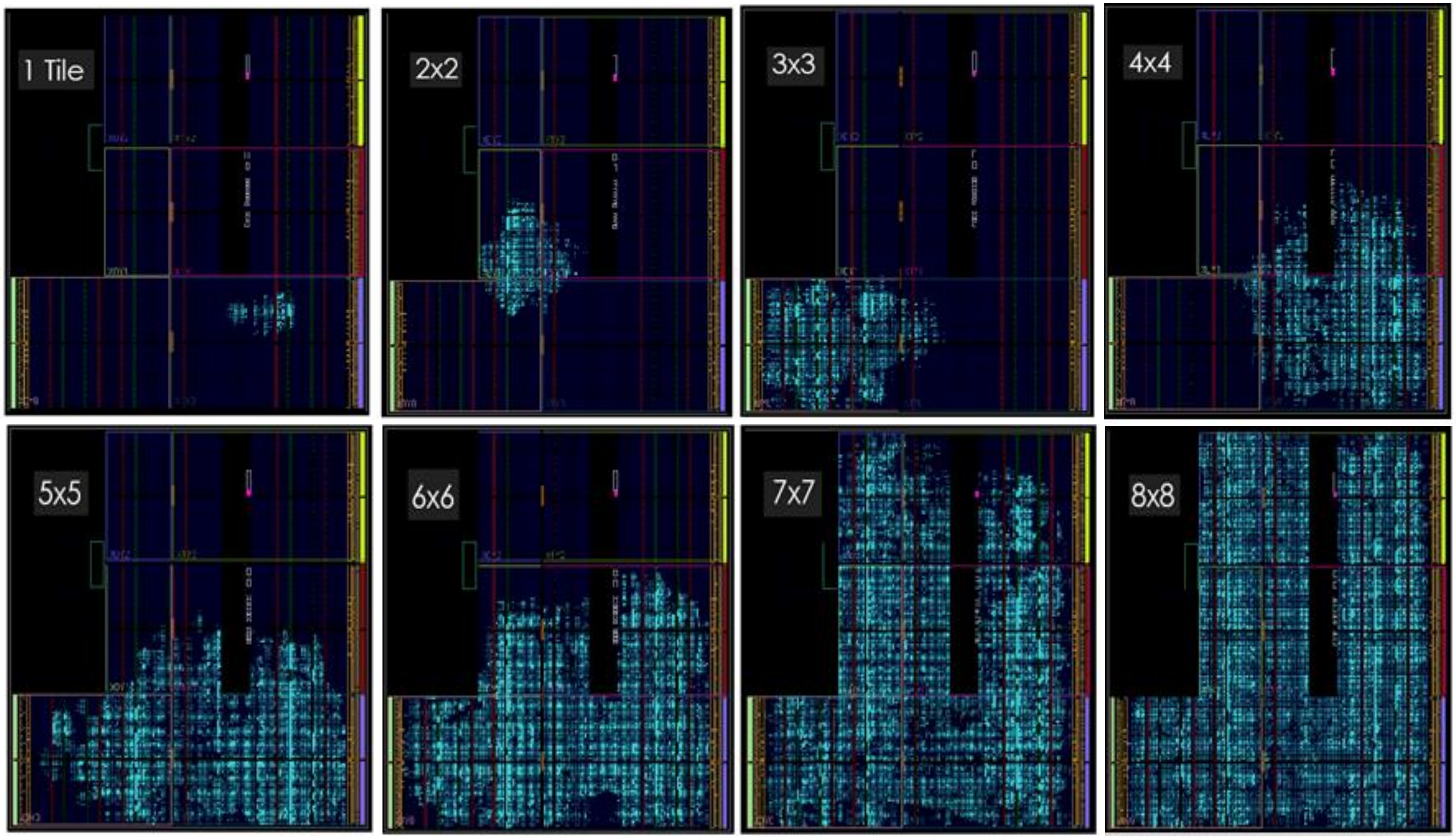
- Xilinx Zynq device can accommodate
  - 6x6 DSP-DySER overlay (36 DSP blocks)
  - 8x8 DISO overlay (64 DSP blocks)
  - 8x8 Dual-DISO overlay (128 DSP blocks)



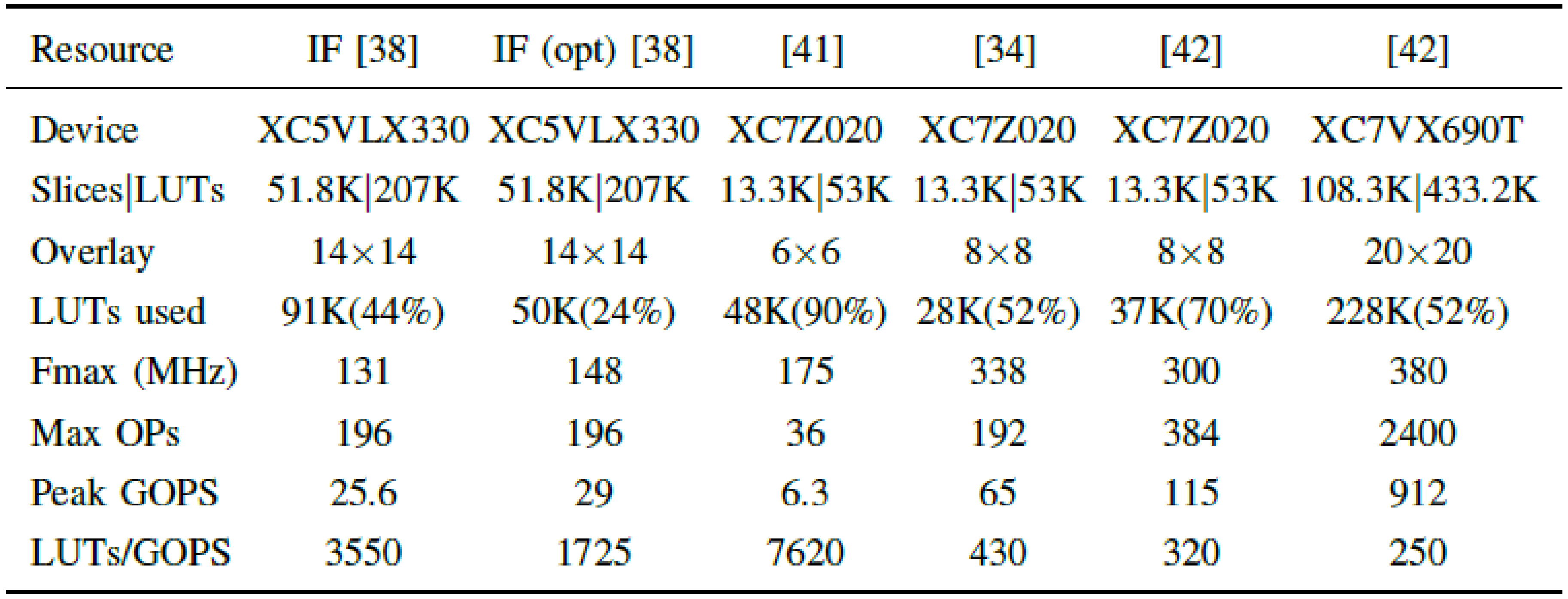




Mapping Overlays onto Zynq

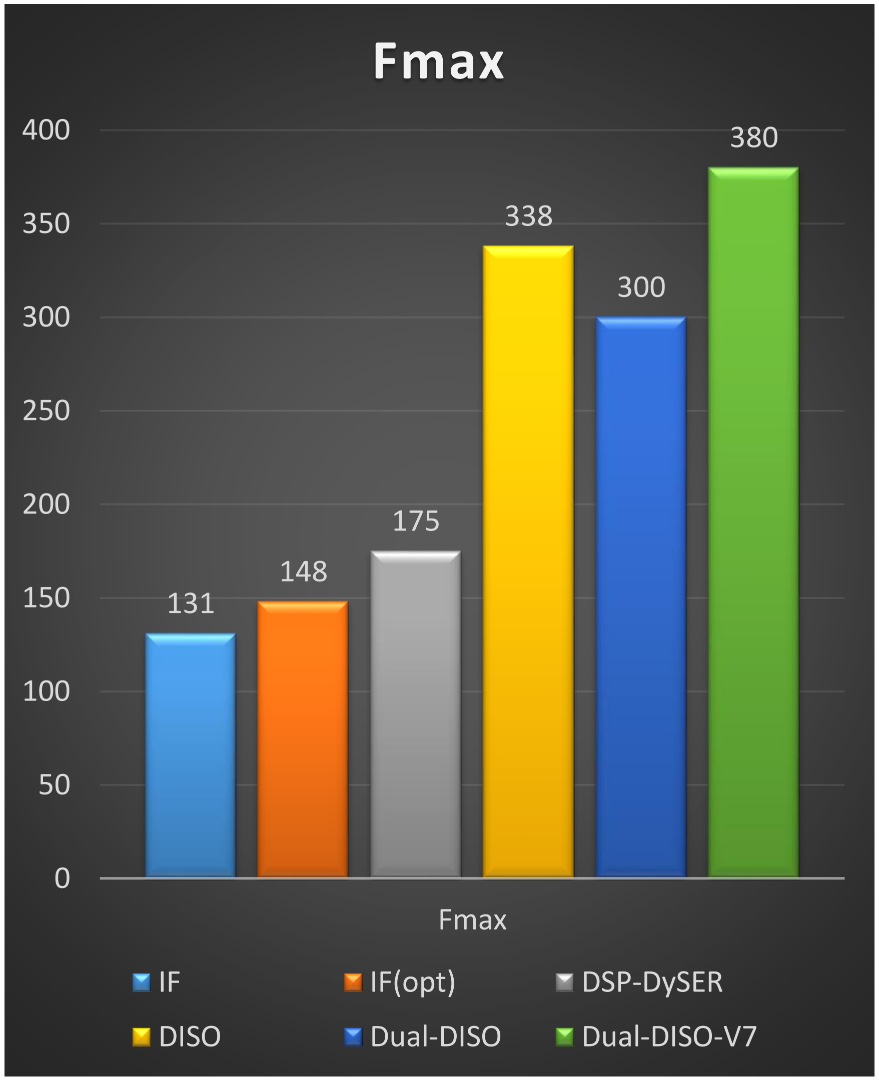


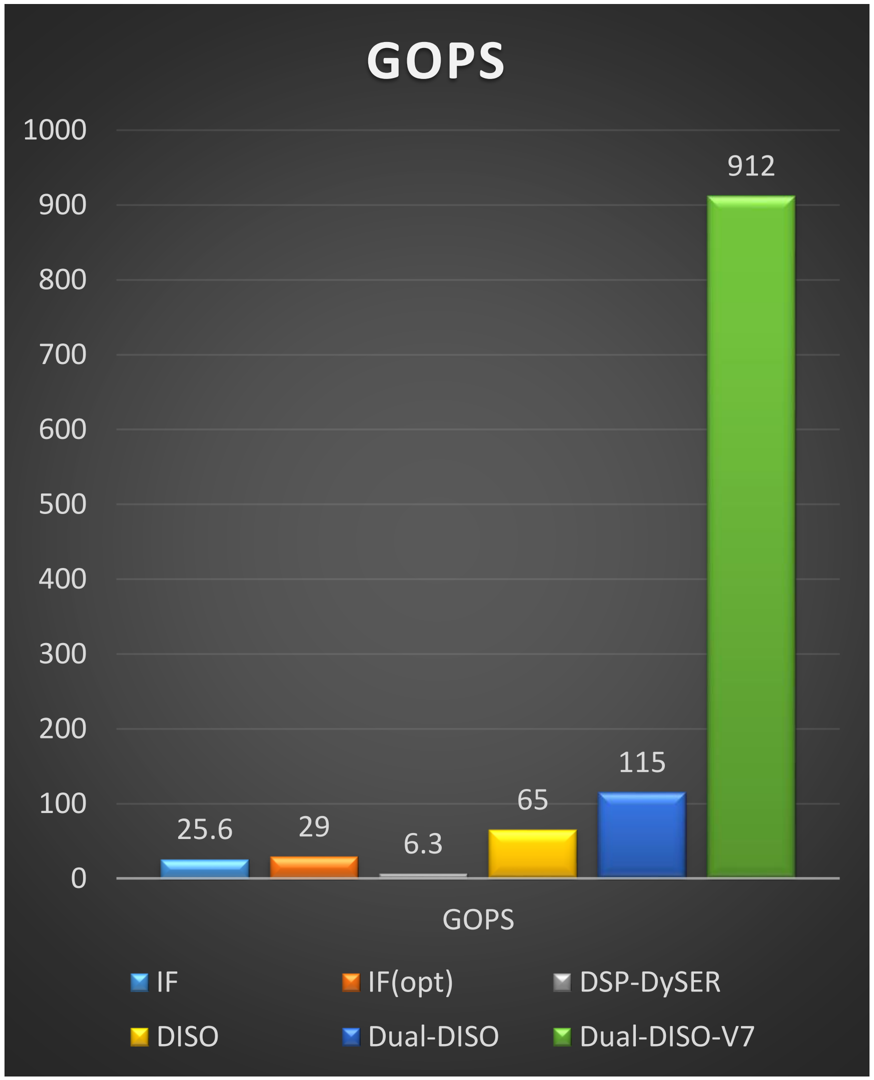
Quantitative Comparison of Overlays



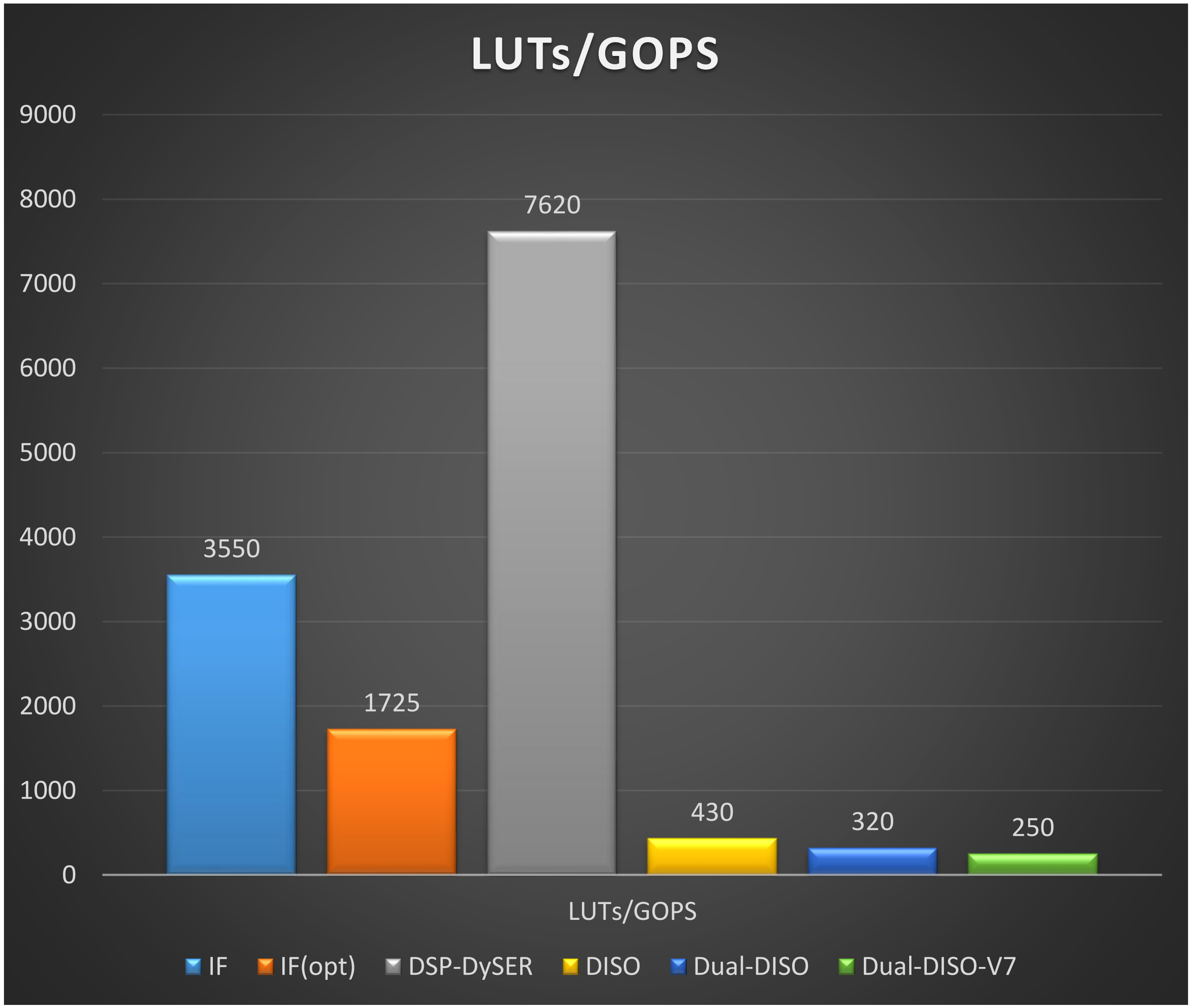
Resource	IF [38]	IF (opt) [38]	[41]	[34]	[42]	[42]
Device	XC5VLX330	XC5VLX330	XC7Z020	XC7Z020	XC7Z020	XC7VX690T
Slices|LUTs	51.8K|207K	51.8K|207K	13.3K|53K	13.3K|53K	13.3K|53K	108.3K|433.2K
Overlay	14×14	14×14	6×6	8×8	8×8	20×20
LUTs used	91K(44%)	50K(24%)	48K(90%)	28K(52%)	37K(70%)	228K(52%)
Fmax (MHz)	131	148	175	338	300	380
Max OPs	196	196	36	192	384	2400
Peak GOPS	25.6	29	6.3	65	115	912
LUTs/GOPS	3550	1725	7620	430	320	250

Quantitative Comparison of Frequency and GOPS

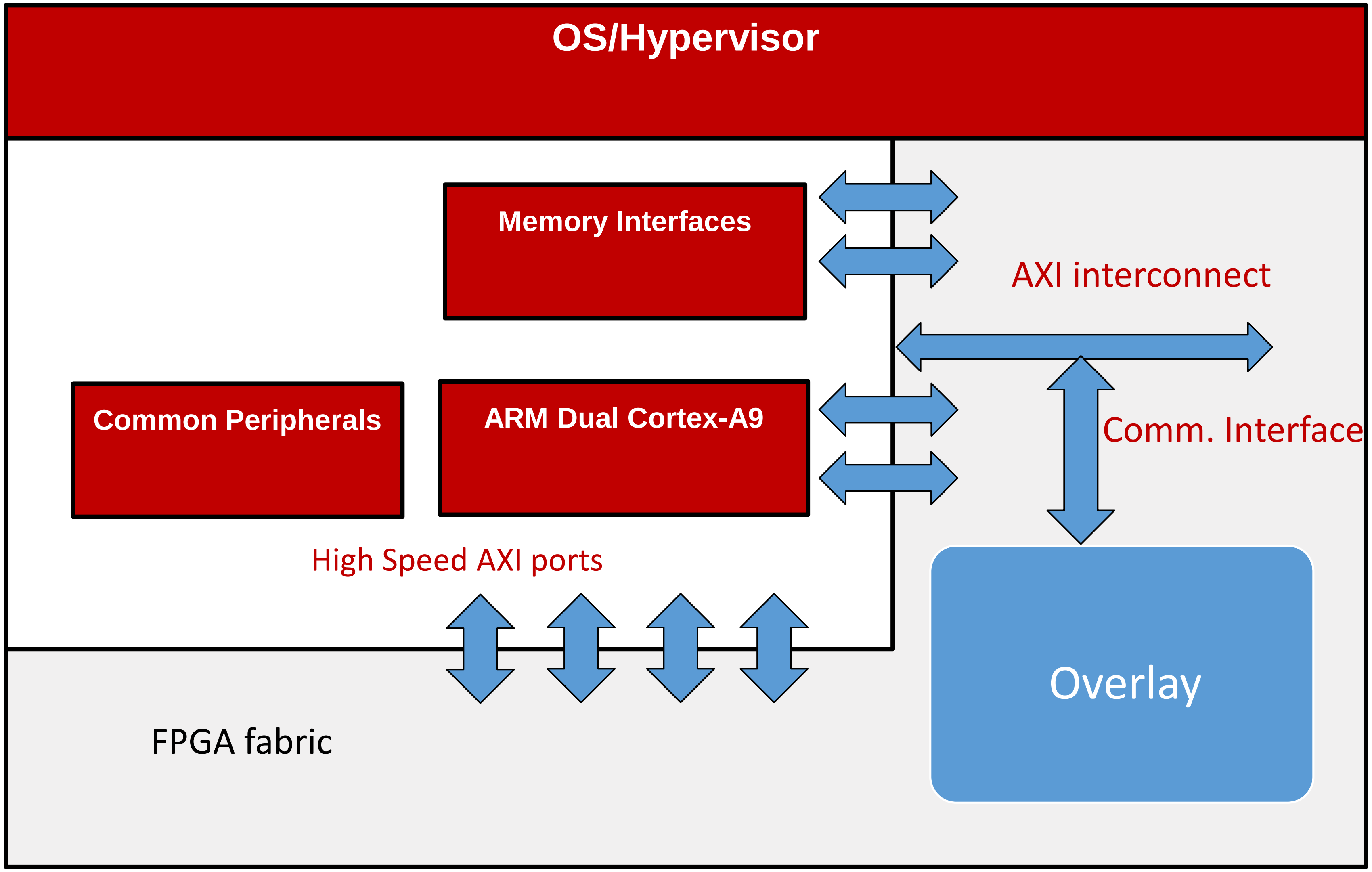




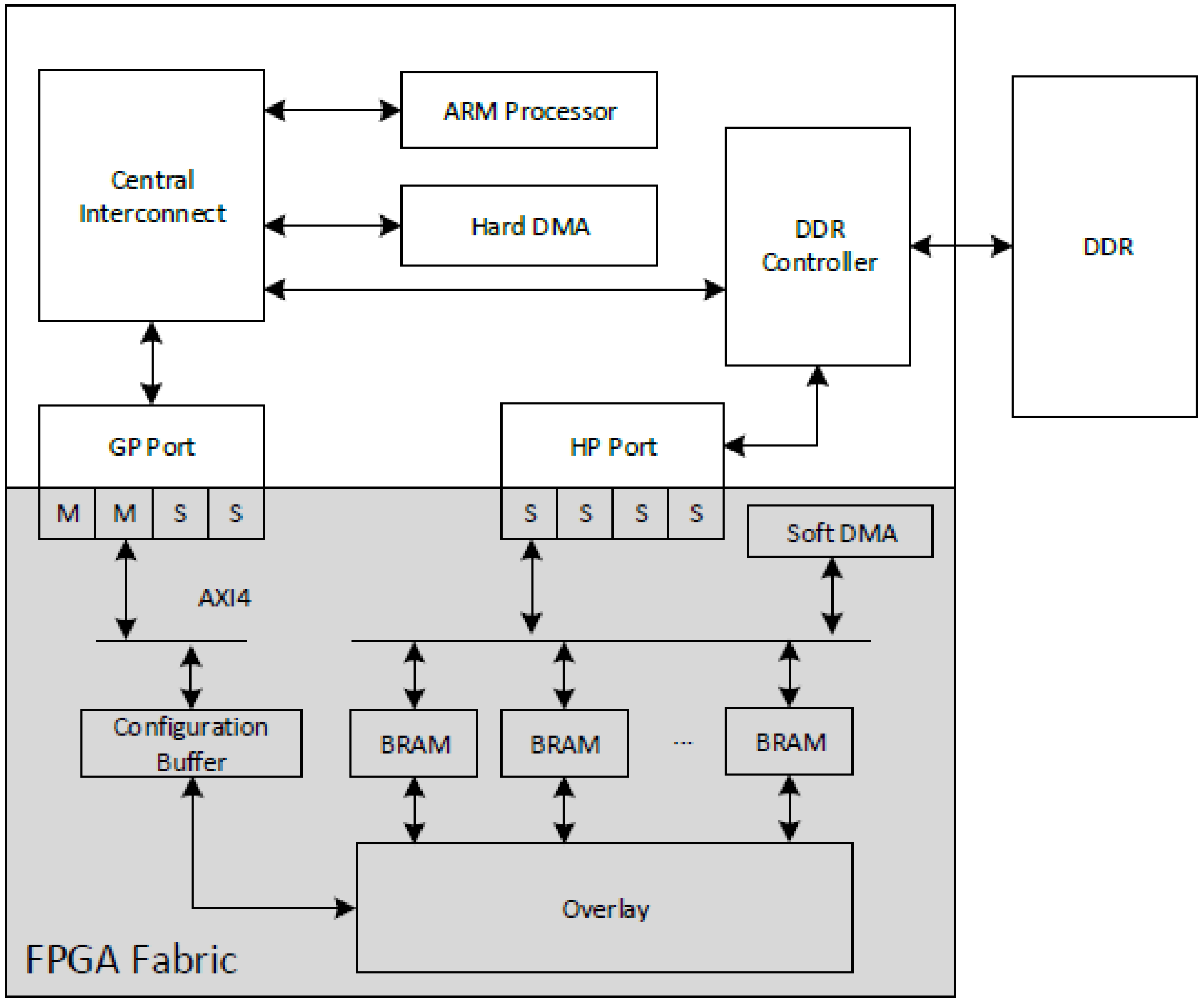
Comparison of Interconnect area overhead



Overlay for FPGA Virtualization in Xilinx Zynq SoC

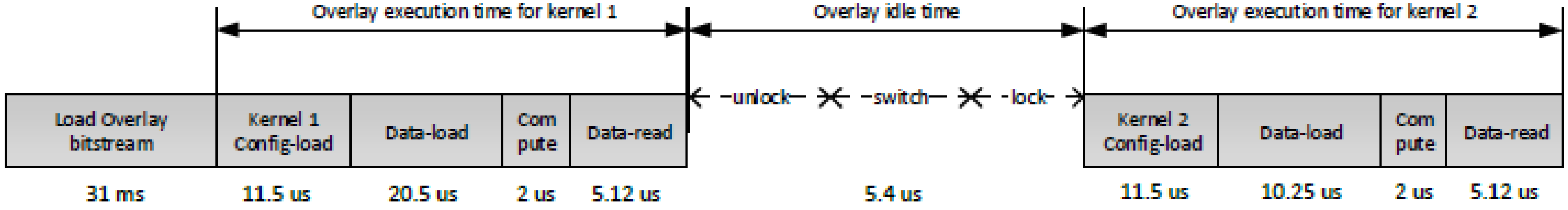


Overlay for FPGA Virtualization in Xilinx Zynq SoC



Context Switching using Hypervisor

- Two kernels using one overlay architecture for execution
- Hypervisor responsible for context switching
- Data transfer is the major bottleneck



Conclusion

- Examined the use of Overlay for
  - General purpose on-demand application acceleration
  - Fast compilation and fast configuration
  - 1200 times faster place and route
  - 1000 times faster context switching
  - 4.5 times hardware performance penalty

- Explored the embedding of an overlay within Xilinx Zynq

- Data communication, major bottleneck in performance

Future Work

- OpenCL compiler for Overlays
- Efficient data communication framework