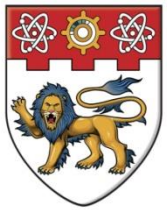


NANYANG TECHNOLOGICAL UNIVERSITY

Adapting the DySER Architecture with DSP Blocks as an Overlay for the Xilinx Zynq

Abhishek K. Jain, Xiangwei Li, Suhaib A. Fahmy, **Douglas L. Maskell**
School of Computer Engineering
Nanyang Technological University (NTU), Singapore

International Symposium On Highly-Efficient Accelerators and Reconfigurable Technologies (HEART)
1st June 2015, Boston MA, USA

Motivation

- Coarse grained overlay architectures
  - Accelerator Design at a higher level of abstraction
  - Software like programmability and fast compilation
  - Improved design productivity

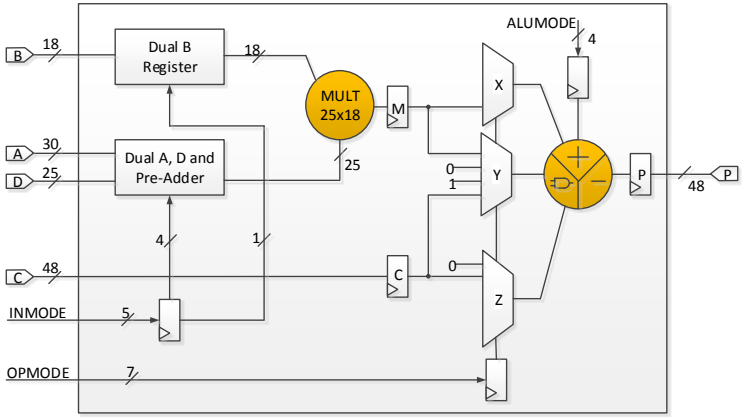
- Cost: Area and performance overheads

- Example: DySER Architecture on Xilinx Virtex-5 FPGA
  - Fine grained resources to implement functional unit and switches
  - Possibility to use embedded DSP blocks as high performance functional units

Idea

- DSP Blocks as hard macros in underlying FPGA architecture

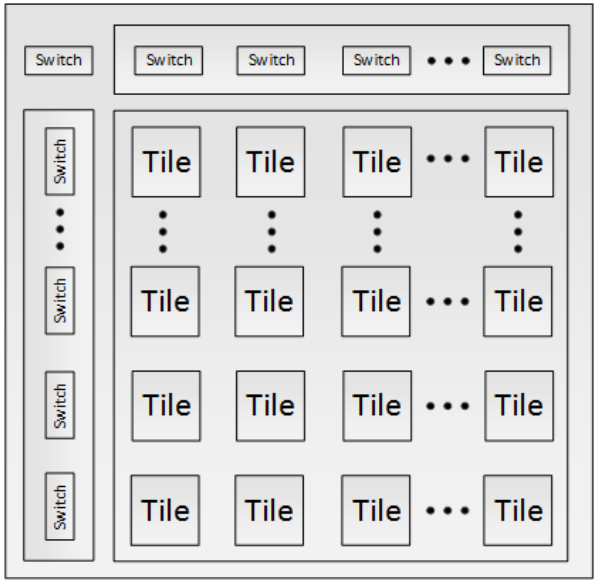
- Fundamental idea: exploiting DSP Block as programmable FU

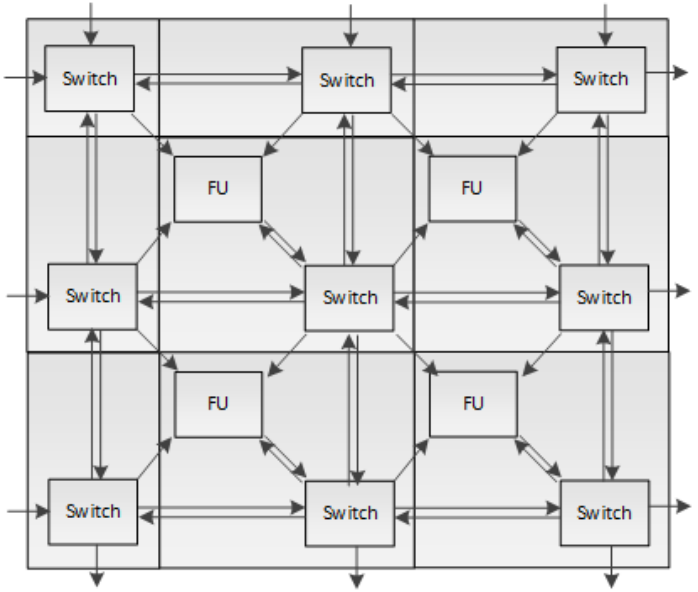


- Exploit programmability feature of DSP Blocks
  - fully pipelined processing elements
  - maximize frequency and throughput

DySER Architecture

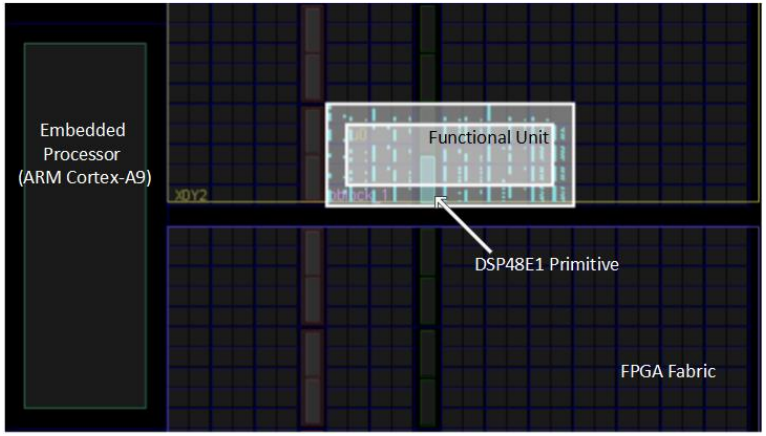
- 2D array of Tiles
- Programmable functional unit and switch in each Tile
- Functional units interconnected via island-style routing network

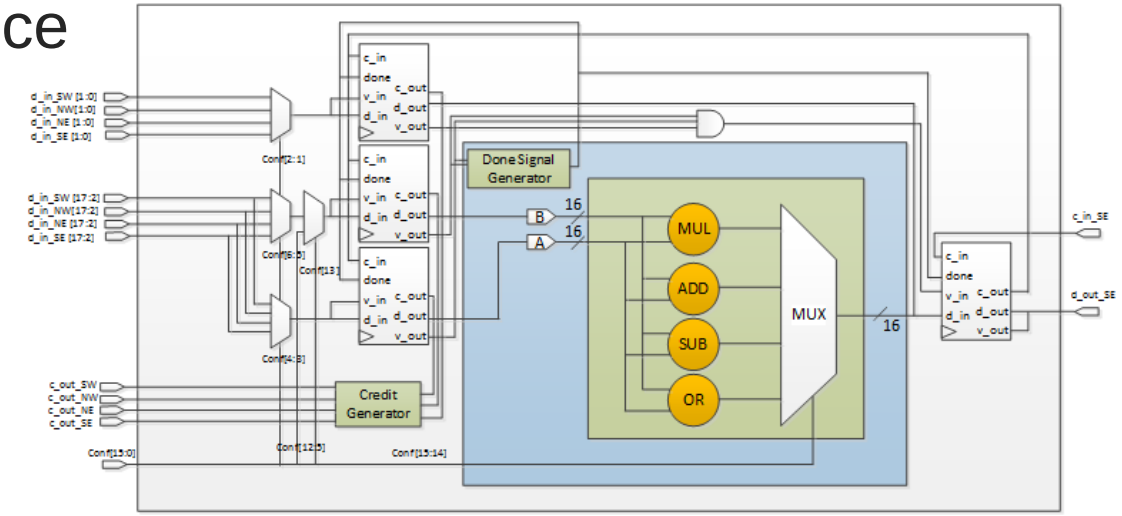




Original Functional Unit

- Implemented on Xilinx Zynq (XC7Z020 CLG484-1)
- Tool infers the DSP block only for the multiplication
- Critical path of 6.7 ns hence limits the performance

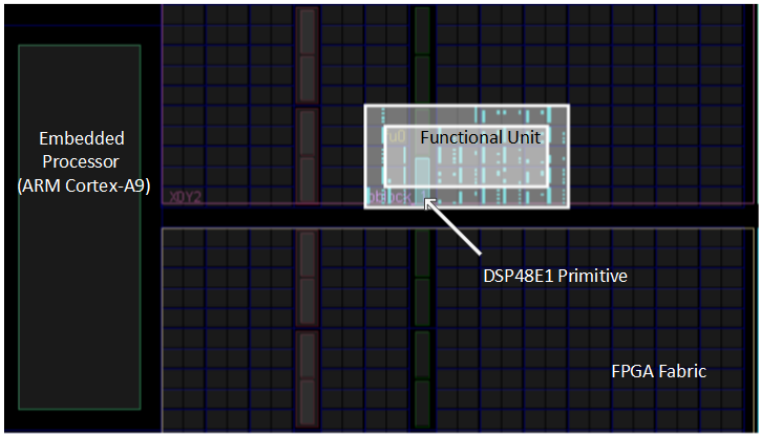


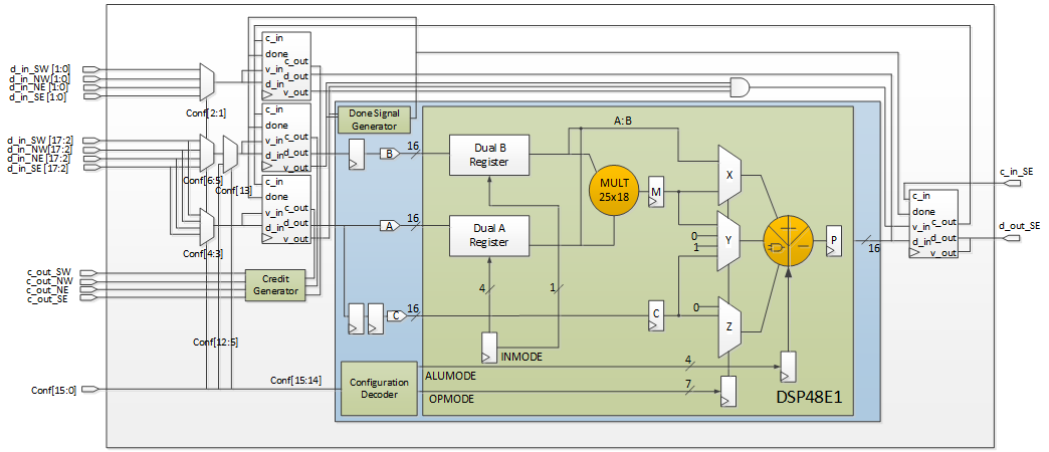


NANYANG TECHNOLOGICAL UNIVERSITY

DSP Block based Functional Unit

- Fully pipelined DSP48E1 as a programmable PE
- Achievable frequency near theoretical limits
- Critical path of 2.7 ns
- Frequency of 370 MHz

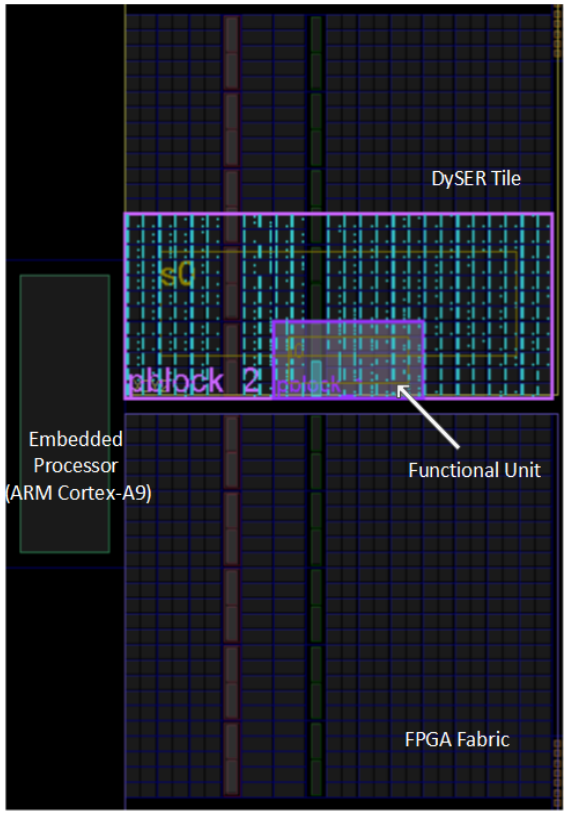




NANYANG TECHNOLOGICAL UNIVERSITY

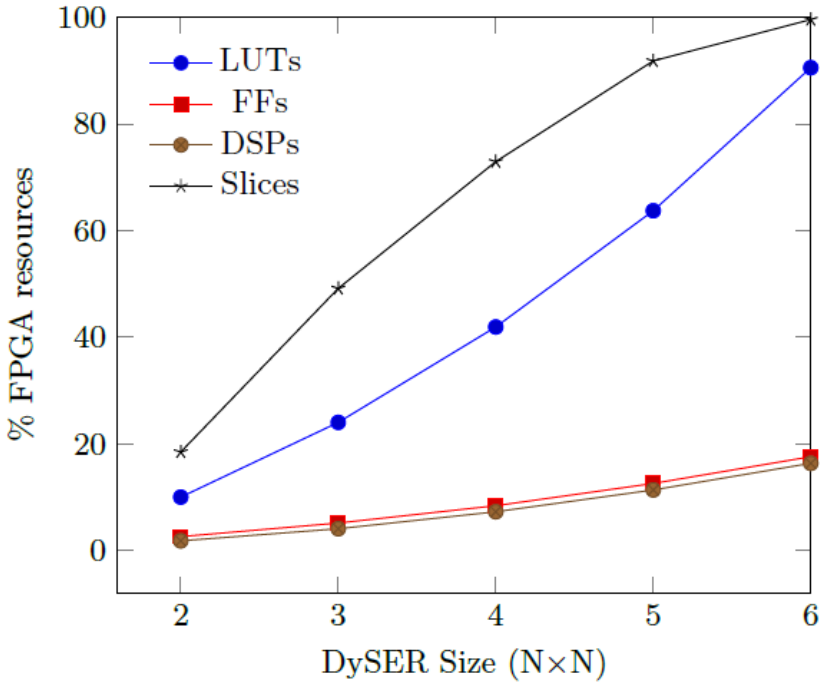
Physical Mapping of DySER Tile

- Now the switch limits the performance of the tile with a critical path of 5.3 ns
- Tile resource consumption: 288 Slices
- Switch: 251 Slices (87% of the tile)
- Small and fast FU using architecture-oriented approach
- Routing resources becomes the limiting factor for the scalability



Mapping to the Xilinx Zynq device

- 6x6 DySER is the largest that can fit on the Zynq
- Slice usage becomes a limiting factor
- Underutilization of DSP blocks (Only 16% were used)



Area Overhead Quantification

- 5x5 DySER can be used to implement the set of compute kernels with a programmability overhead of 13x slices

Benchmark	LUTs	FFs	Slices	DSPs	Frequency (MHz)
fft	218 (0.4%)	485 (0.4%)	117 (0.9%)	4 (1.8%)	324
kmeans	613 (1.1%)	1252(1.2%)	215 (1.6%)	8 (3.6%)	249
mm	315 (0.6%)	920 (0.8%)	205 (1.5%)	8 (3.6%)	295
mri-q	243 (0.4%)	588 (0.5%)	147 (1.1%)	6 (2.7%)	268
spmv	292 (0.5%)	842 (0.8%)	180 (1.3%)	8 (3.6%)	297
stencil	460 (0.8%)	870 (0.8%)	200 (1.5%)	2 (0.9%)	303
conv	353 (0.6%)	918 (0.8%)	222 (1.6%)	8 (3.6%)	272
radar	163 (0.3%)	457 (0.4%)	92 (0.7%)	6 (2.7%)	304
5×5 FU array	2900 (5.5%)	2925 (2.7%)	925 (6.9%)	25 (11.4%)	370
5×5 DySER	33875 (63.7%)	13390 (12.6%)	12284 (92.4%)	25 (11.4%)	175

Conclusion

- Area and performance efficient FU can be built by making use of DSP block as an ALU, instead of just as a multiplier, and enabling the internal pipeline registers
- Enhancement to the DySER FU
  - An improvement of 2.5 times in the frequency
  - A reduction of 25% in the area
- Quantification of area overheads by mapping benchmark set on DySER and to the FPGA Fabric using Vivado-HLS
- 5x5 DySER can be used to implement the set of compute kernels with a programmability overhead of 13x slices

Future Work

- Area overhead reduction by optimizing routing network
- DySER integration with the ARM processor
- Cycle by cycle reconfiguration of the DSP block to support arbitrary size kernels

Experimental Evaluation

Benchmark	Benchmark Characteristics			Overlay Results			HLS Implementation Results		
Name	op nodes	node-merging	% Savings	Latency	Fmax	GOPS	Latency	Fmax	GOPS
chebyshev	7	5	28%	49	370	**2.59**	13	333	**2.30**
sgfilter	18	10	44%	54	370	**6.66**	11	278	**5.00**
mibench	13	6	53%	47	370	**4.81**	9	295	**3.80**
qspline	26	22	15%	76	370	**9.62**	21	244	**6.30**
poly1	9	6	33%	34	370	**3.33**	12	285	**2.56**
poly2	9	6	33%	29	370	**3.33**	11	295	**2.65**
poly3	11	7	36%	31	370	**4.07**	12	250	**2.75**
poly4	6	3	50%	24	370	**2.22**	7	312	**1.87**
atax	60	36	40%	72	300	**18.00**	13	263	**15.80**
bicg	30	18	40%	46	300	**9.00**	7	270	**8.10**
trmm	54	36	33%	58	300	**16.20**	8	222	**11.90**
syrk	72	45	37%	41	300	**21.60**	10	250	**18.00**

	Benchmark set-I 8 compute kernels (up-to 26 operations)	**Benchmark set-II 4 compute kernels (up-to 72 operations)**
Benchmark set Mapped on	Overlay-I (5x5, CW=2)	Overlay-II (7x7, CW=4),
Operating frequency	370 MHz	300 MHz
Overlay reconfiguration time	11.5 us	28 us
11-52% higher throughput compared to Vivado HLS implementations		