DATE 16

DESIGN, AUTOMATION & TEST IN EUROPE

14 – 18 March, 2016 · ICC · Dresden · Germany

The European Event for Electronic
System Design & Test

Throughput Oriented FPGA Overlays Using DSP Blocks

NANYANG TECHNOLOGICAL UNIVERSITY

Abhishek K. Jain, Douglas L. Maskell
School of Computer Engineering
Nanyang Technological University, Singapore

WARWICK
THE UNIVERSITY OF WARWICK

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

FPGAs: ready for the mainstream?

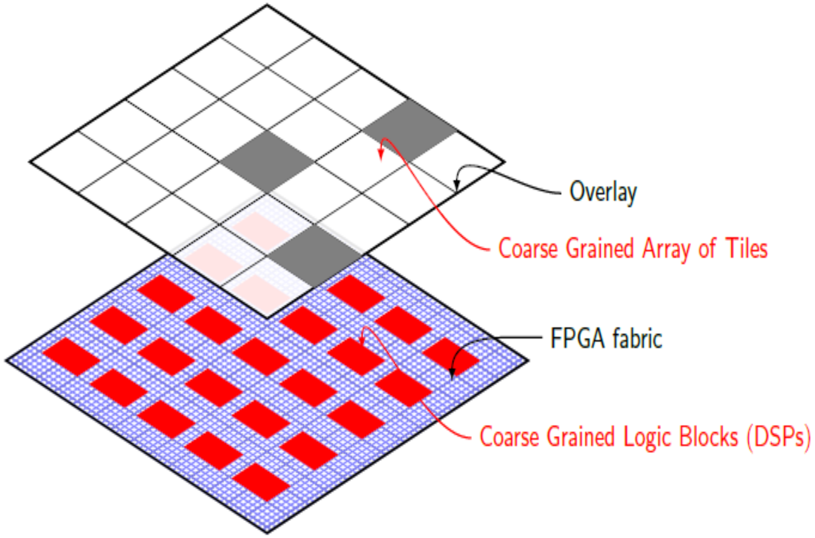
- **Established strength in hardware acceleration**
- **Only heavily used in niche applications**
- **Mainly due to poor design productivity:**
  - **Difficulty of hardware design at low level of abstraction (RTL)**
  - **Long compilation times (specifically place and route times)**
- **Need for software-like abstractions and fast development cycles**
- **Two approaches to address this:**
  - **High level synthesis (HLS)**
  - **Coarse-grained FPGA overlays**

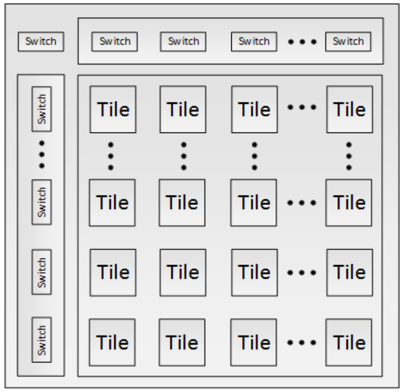
High level synthesis (HLS)

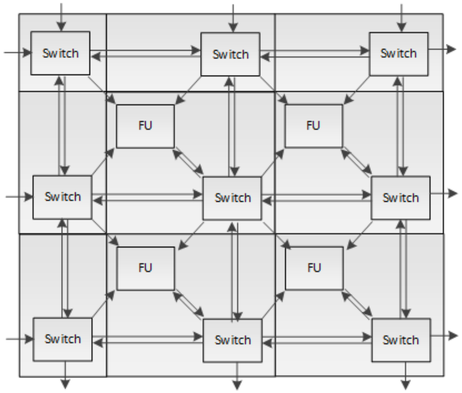
- **HLS tools allow designers to use more abstract languages**
  - **Less focus on low-level bits and clock cycles**
  - **More powerful expressiveness**
  - **A large back-catalogue of code**
- **However, they still generate RTL code**
  - **Must still go through complicated and time-consuming back-end flow**
  - **They can be inefficient since the architecture is one more step removed from the code**

Coarse grained FPGA overlays

- Array of coarse-grained tiles
- Programmable FU and interconnect resources
- Benefits:
  - Accelerator design at a higher level of abstraction
  - Fast compilation and development cycles
  - Improved design productivity
- Cost: area and performance overheads
- Example: DySER Architecture

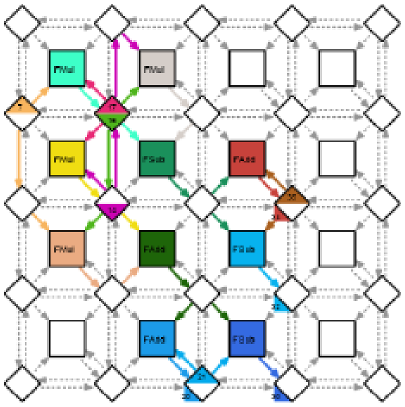

Overlay
Coarse Grained Array of Tiles
FPGA fabric
Coarse Grained Logic Blocks (DSPs)



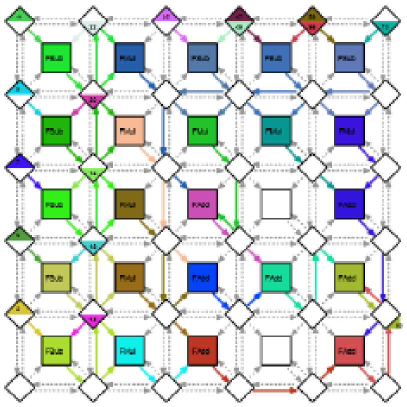


Coarse grained FPGA overlays

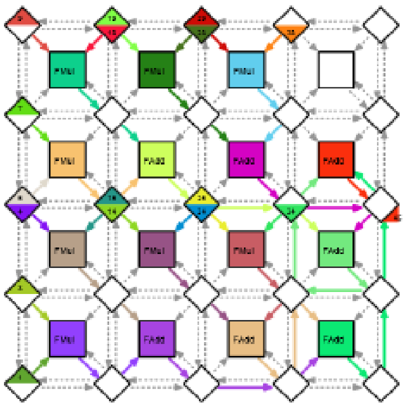
- **Kernel Mapping on DySER Architecture**



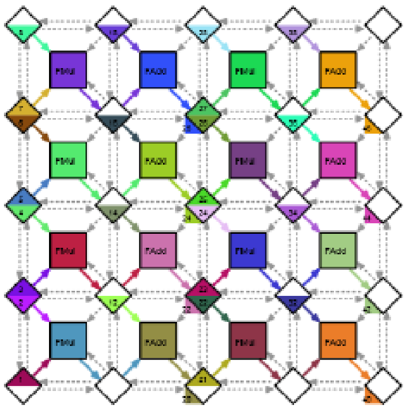
fft



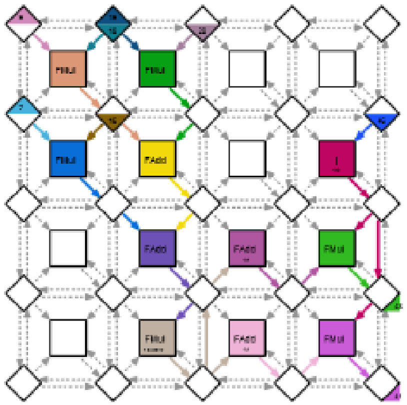
kmeans



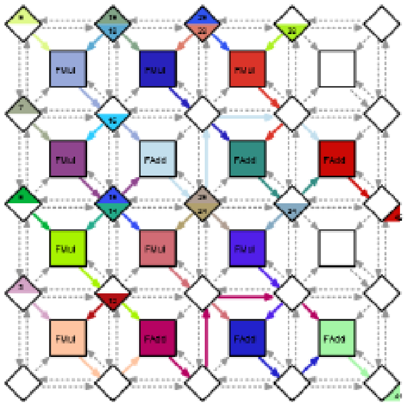
mm



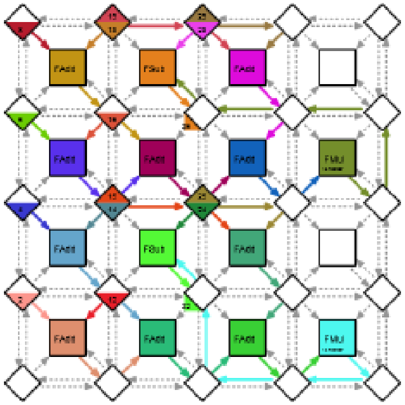
conv



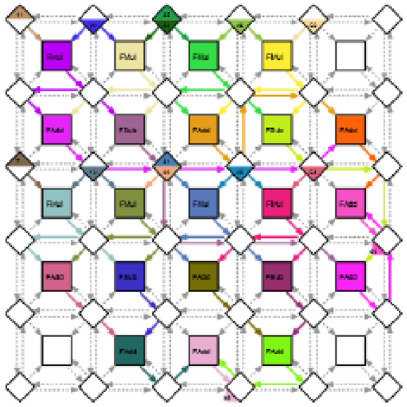
mri-q



spmv

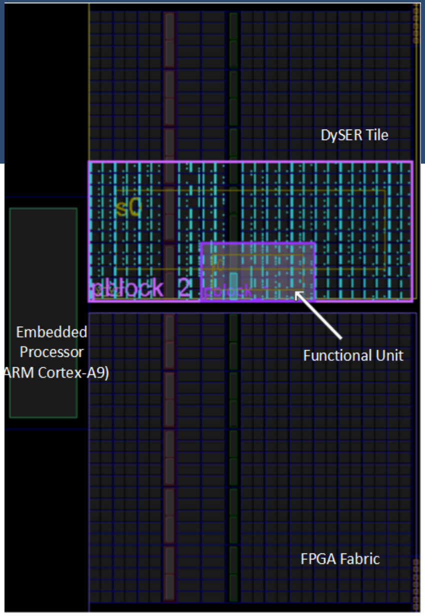


stencil

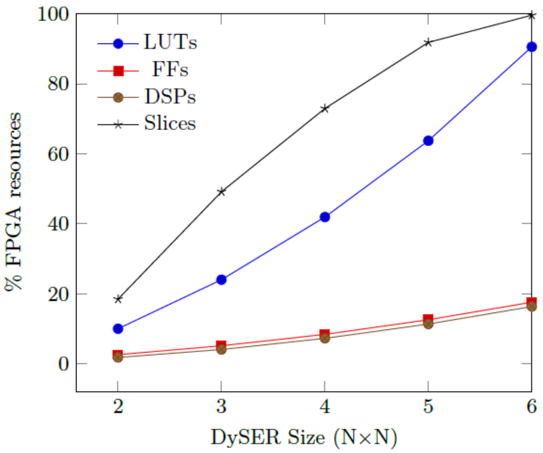


radar

Coarse grained FPGA overlays



- **DySER [1,2]**
  - Compiled a set of 8 kernels
  - RTL implementation[1] does not fit on V5 due to excessive LUT requirements
- **Adapt using DSP block as FU[2]**
  - Fit 36 homogenous flexible FUs on a Xilinx Zynq ZC7Z020
  - F_{max} = 175 MHz, peak throughput of 6 GOPS
  - Area overhead: 33K extra LUTs compared to direct FPGA implementation of kernels
  - Mainly due to resource heavy interconnect

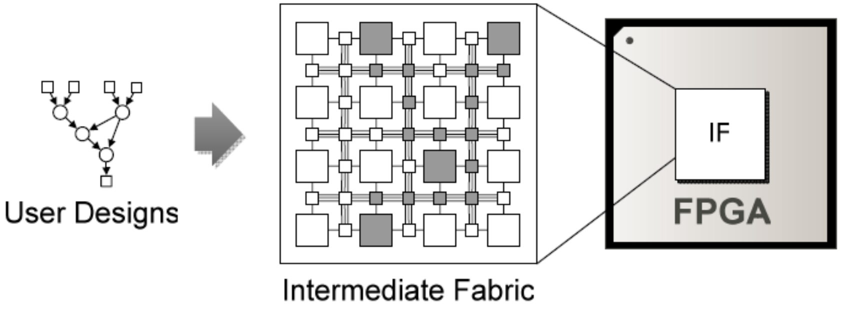


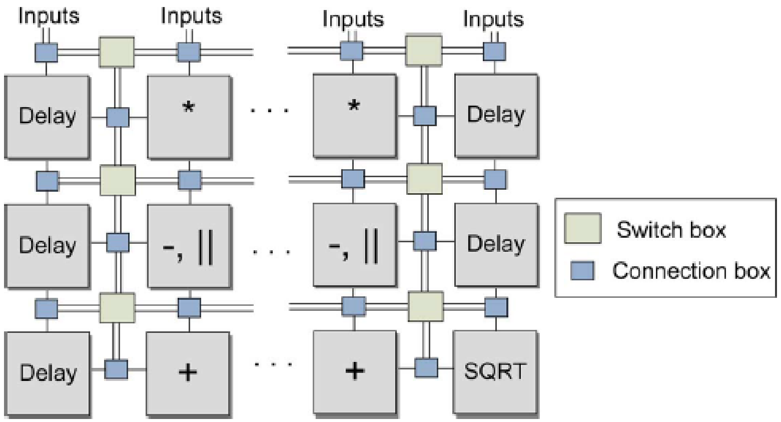
[1] Chen et. al., "Performance evaluation of a DySER FPGA prototype system spanning the compiler, microarchitecture, and hardware implementation," in ISPASS, 2015.

[2] 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 HEART, 2015.

Coarse grained FPGA overlays

- **Intermediate Fabrics [1]**
  - **192 heterogeneous FUs on Altera Stratix III FPGA for a set of 8 kernels**
  - **Configuration data size: 9K bits compared to 15M bits of FPGA**
  - **PAR speedup of 700× compared to FPGA (2.7 second)**
  - **F_{max} of only 124 MHz resulting in a peak throughput of 24 GOPS**
  - **Area overheads: 42K extra LUTs compared to direct FPGA implementation consuming 15K LUTs**

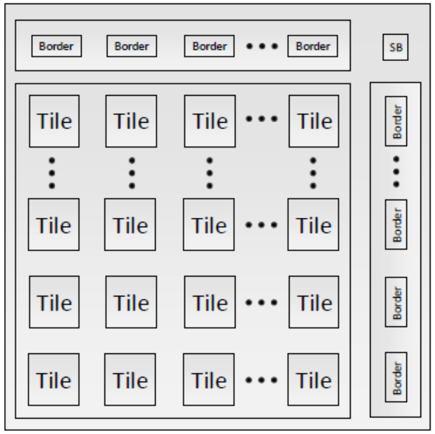


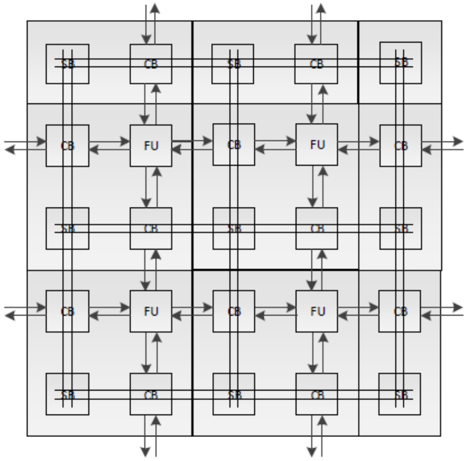


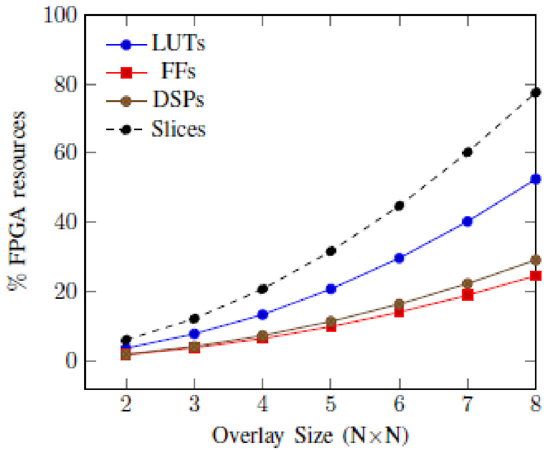
[1] G. Stitt and J. Coole, "Intermediate fabrics: Virtual architectures for near-instant FPGA compilation," IEEE ESL, vol. 3(3), 2011.

Coarse grained FPGA overlays

- **Fully pipelined DSP Block based overlay[1]**
  - **FU uses fully pipelined DSP block as a programmable ALU**
  - **FPGA like island-style interconnect network (full word)**
  - **Scalability Analysis: can fit an 8×8 array of FUs (64 DSPs) on Xilinx Zynq**
  - **For CW=2, consuming 28K LUTs with F_{max} = 338 MHz**
  - **LUTs/DSP = 438, LUTs/OP = 146 and Peak Throughput: 65 GOPS**
  - **LUT usage limits scalability and peak performance**







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

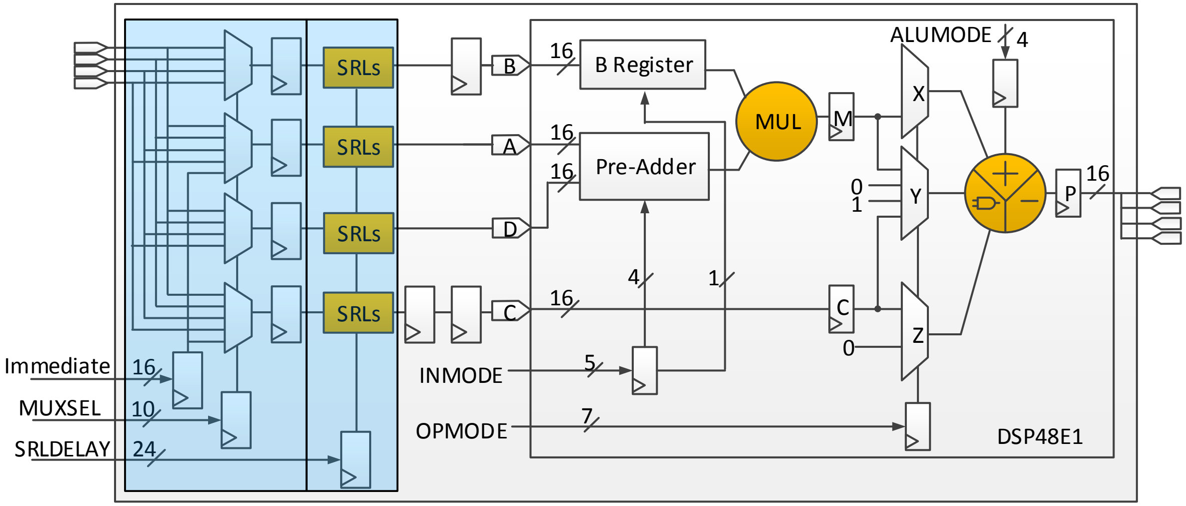
Coarse Grained FPGA Overlays

- **Fully pipelined DSP Block based overlay[1]**
  - Prototyped two overlays:
    - Overlay-I (5×5, CW=2, F_{max} = 370 MHz)
    - Compiled a set of 8 kernels: Average FU utilization: 30%
    - Overlay-II(7×7, CW=4, F_{max} = 300 MHz)
    - Compiled a set of 4 kernels: Average FU utilization: 60%
  - Up to 53% saving in required tiles (Using DSP block aware mapping)
  - An improvement of up-to 52% in throughput compared to Vivado HLS
  - Can we optimize the overlay further to reduce LUTs/DSP and to improve peak performance?
  - How do we improve the average FU utilization?
  - Can we map multiple instances of smaller kernels on the overlay fabric?

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

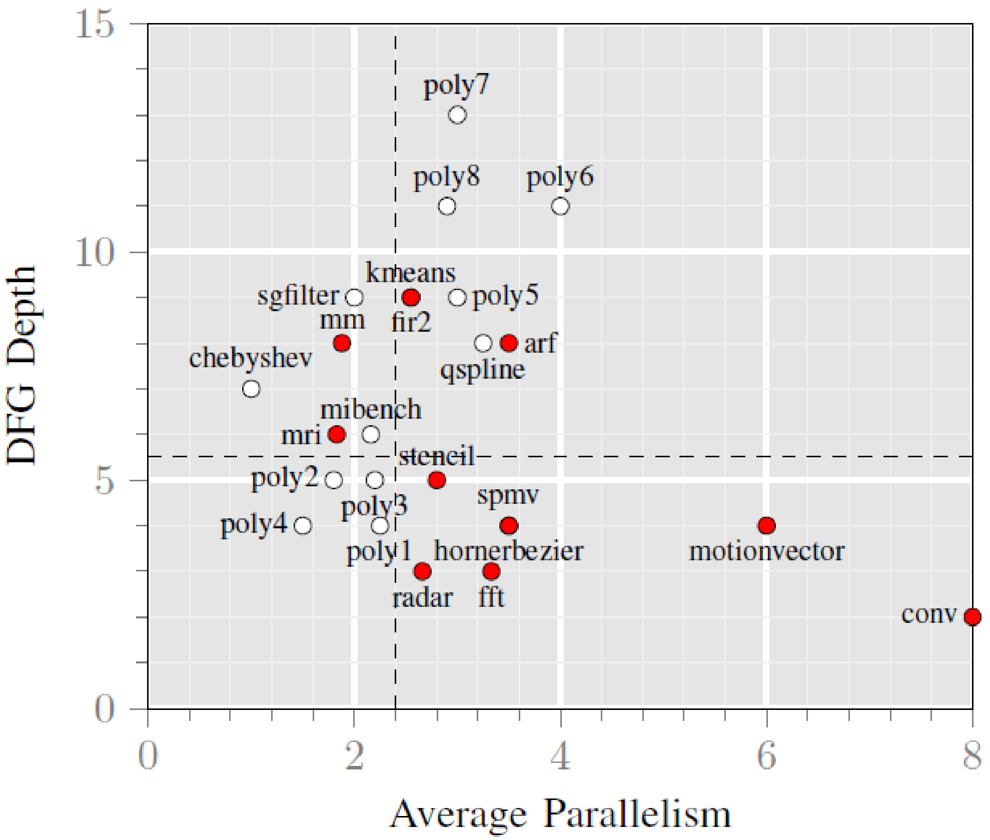
FU based on single DSP Block

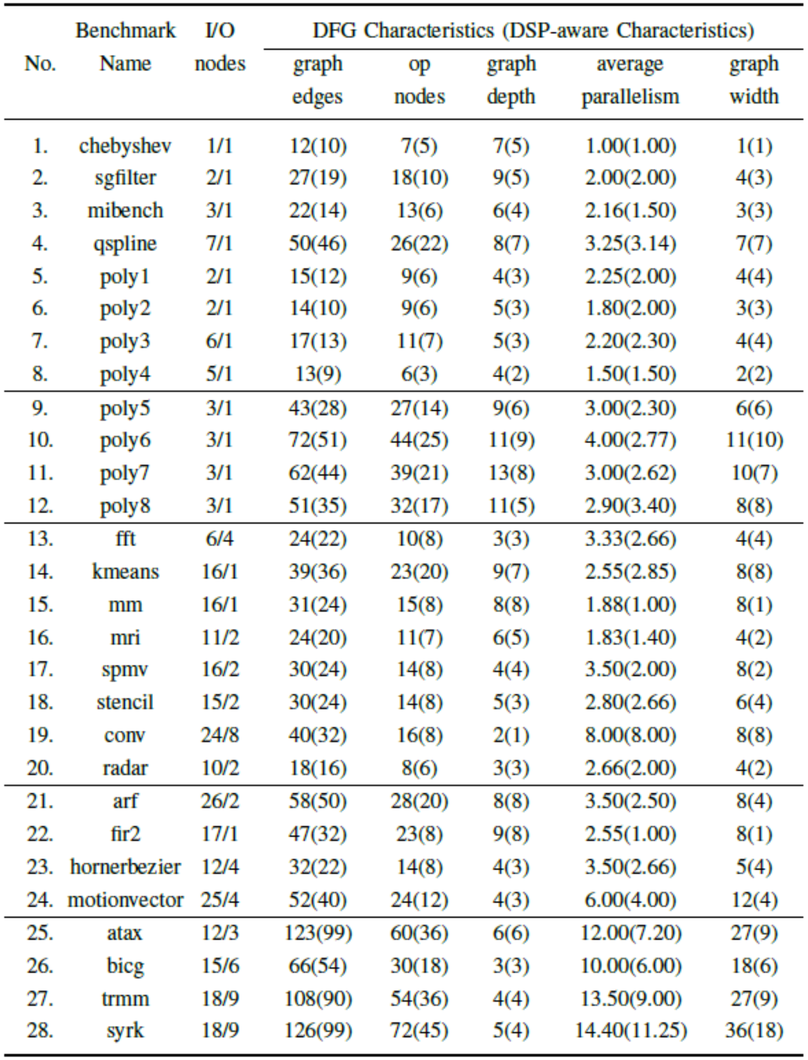
- **Achievable frequency near theoretical limits**
- **400 MHz on the Xilinx Zynq Device (XC7Z020)**
- **Three main blocks:**
  - **Fully pipelined DSP48E1 block as programmable PE**
  - **SRL based variable-length shift registers**
  - **MUX based reordering logic**



Analysis of compute kernels

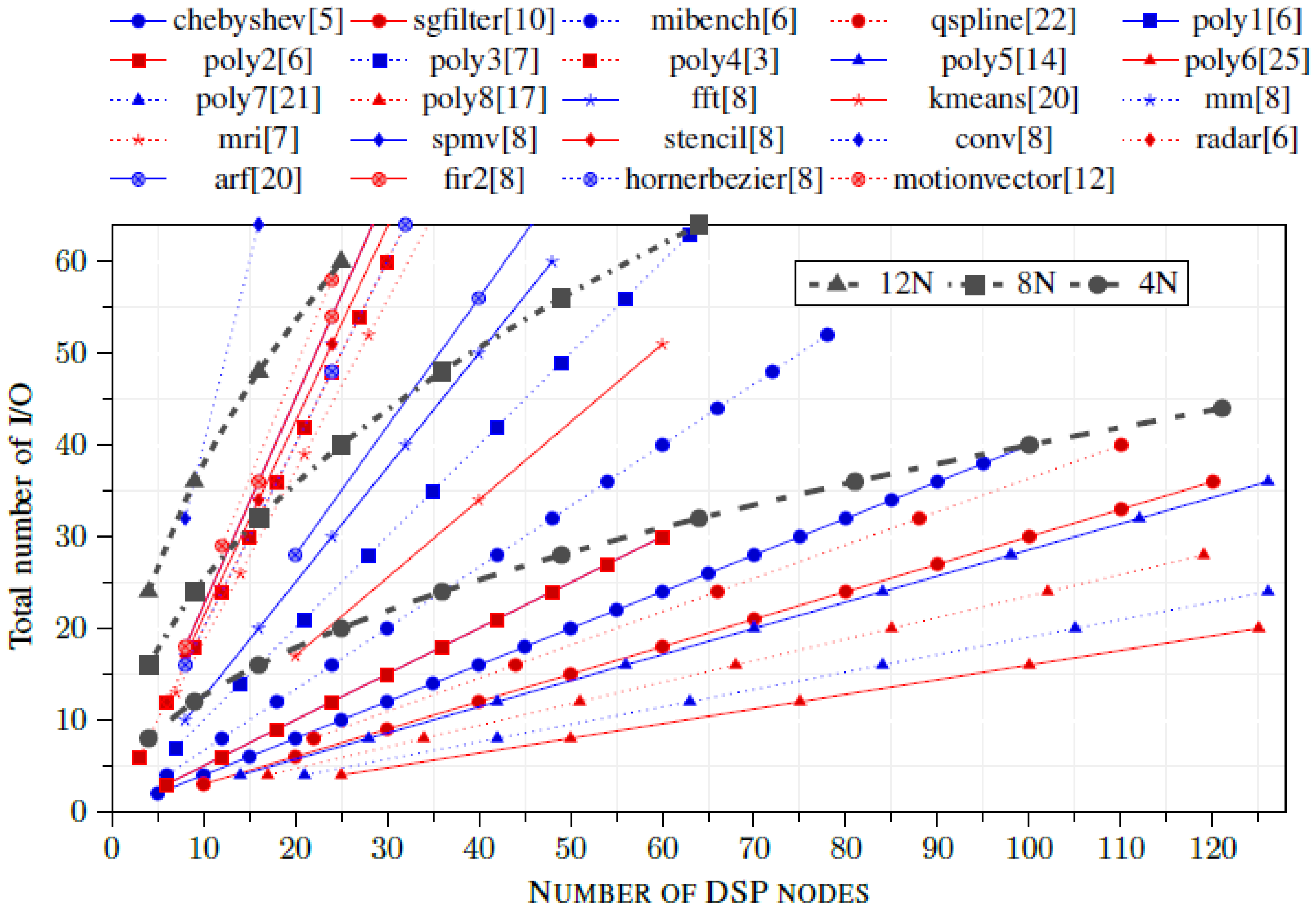
- **Benchmark set of 28 kernels**



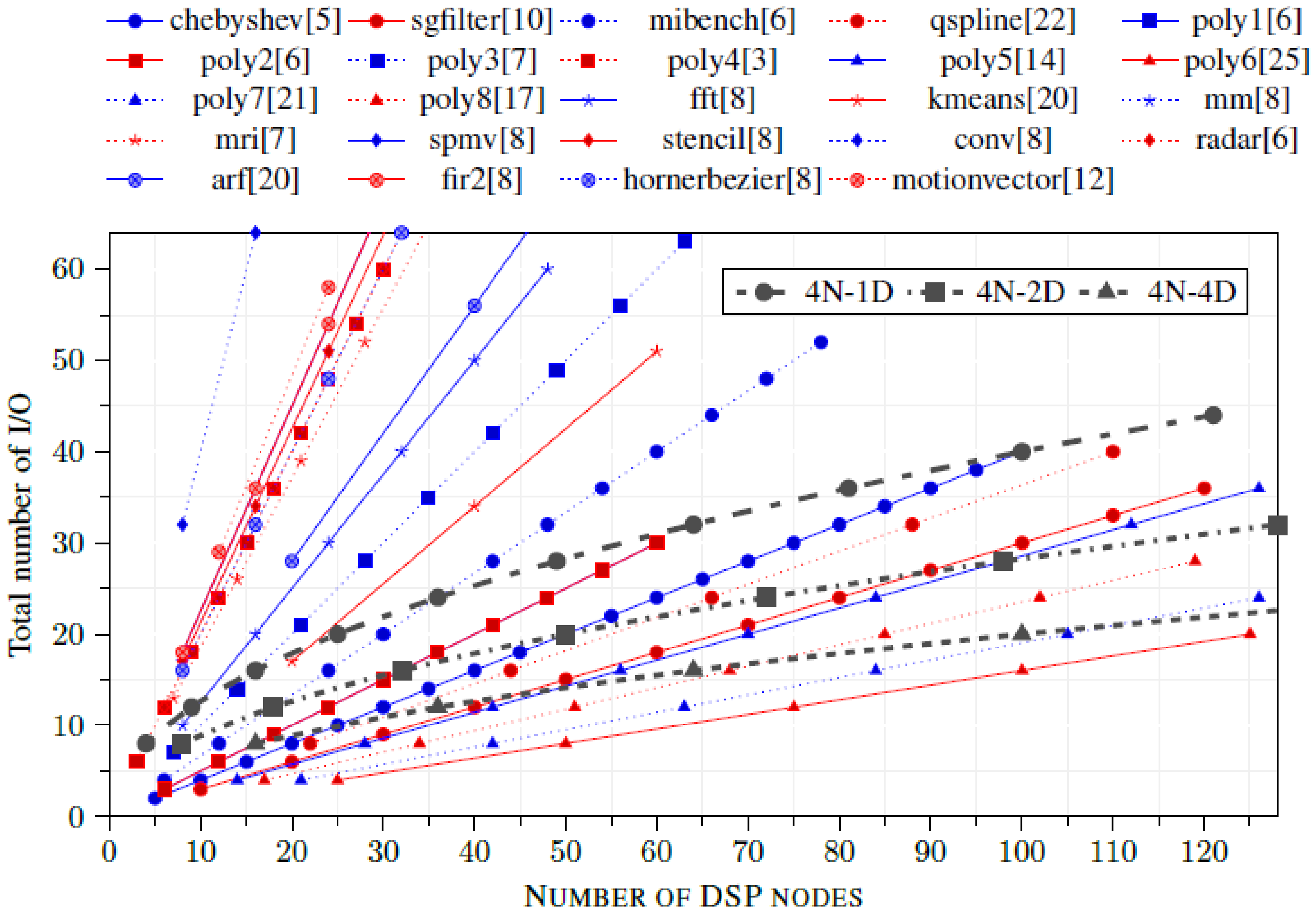


No.	Benchmark Name	I/O nodes	DFG Characteristics (DSP-aware Characteristics)				
			graph edges	op nodes	graph depth	average parallelism	graph width
1.	chebyshev	1/1	12(10)	7(5)	7(5)	1.00(1.00)	1(1)
2.	sgfilter	2/1	27(19)	18(10)	9(5)	2.00(2.00)	4(3)
3.	mibench	3/1	22(14)	13(6)	6(4)	2.16(1.50)	3(3)
4.	qspline	7/1	50(46)	26(22)	8(7)	3.25(3.14)	7(7)
5.	poly1	2/1	15(12)	9(6)	4(3)	2.25(2.00)	4(4)
6.	poly2	2/1	14(10)	9(6)	5(3)	1.80(2.00)	3(3)
7.	poly3	6/1	17(13)	11(7)	5(3)	2.20(2.30)	4(4)
8.	poly4	5/1	13(9)	6(3)	4(2)	1.50(1.50)	2(2)
9.	poly5	3/1	43(28)	27(14)	9(6)	3.00(2.30)	6(6)
10.	poly6	3/1	72(51)	44(25)	11(9)	4.00(2.77)	11(10)
11.	poly7	3/1	62(44)	39(21)	13(8)	3.00(2.62)	10(7)
12.	poly8	3/1	51(35)	32(17)	11(5)	2.90(3.40)	8(8)
13.	fft	6/4	24(22)	10(8)	3(3)	3.33(2.66)	4(4)
14.	kmeans	16/1	39(36)	23(20)	9(7)	2.55(2.85)	8(8)
15.	mm	16/1	31(24)	15(8)	8(8)	1.88(1.00)	8(1)
16.	mri	11/2	24(20)	11(7)	6(5)	1.83(1.40)	4(2)
17.	spmv	16/2	30(24)	14(8)	4(4)	3.50(2.00)	8(2)
18.	stencil	15/2	30(24)	14(8)	5(3)	2.80(2.66)	6(4)
19.	conv	24/8	40(32)	16(8)	2(1)	8.00(8.00)	8(8)
20.	radar	10/2	18(16)	8(6)	3(3)	2.66(2.00)	4(2)
21.	arf	26/2	58(50)	28(20)	8(8)	3.50(2.50)	8(4)
22.	fir2	17/1	47(32)	23(8)	9(8)	2.55(1.00)	8(1)
23.	hornerbezier	12/4	32(22)	14(8)	4(3)	3.50(2.66)	5(4)
24.	motionvector	25/4	52(40)	24(12)	4(3)	6.00(4.00)	12(4)
25.	atax	12/3	123(99)	60(36)	6(6)	12.00(7.20)	27(9)
26.	bicg	15/6	66(54)	30(18)	3(3)	10.00(6.00)	18(6)
27.	trmm	18/9	108(90)	54(36)	4(4)	13.50(9.00)	27(9)
28.	syrk	18/9	126(99)	72(45)	5(4)	14.40(11.25)	36(18)

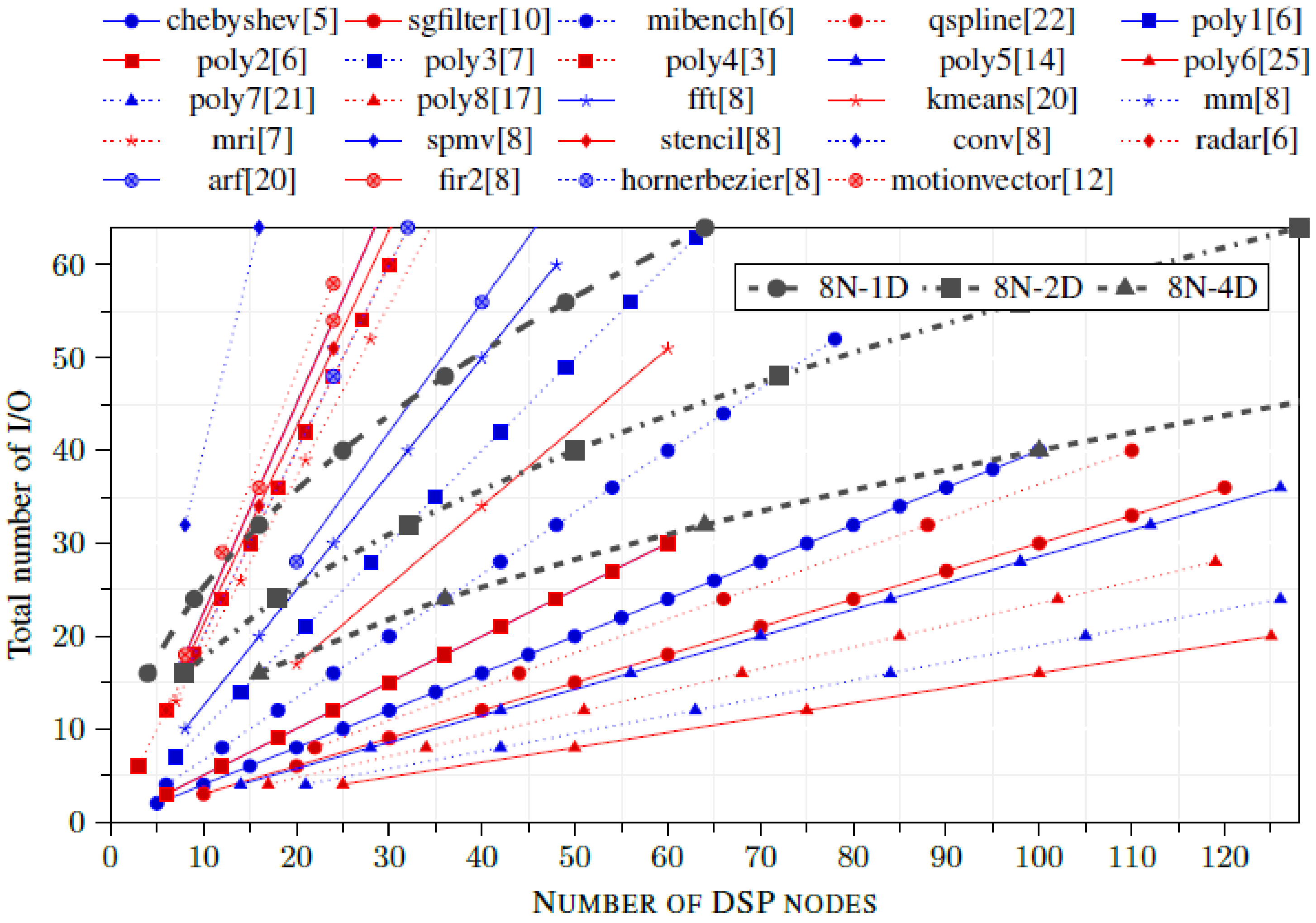
Kernel & overlay I/O scalability analysis



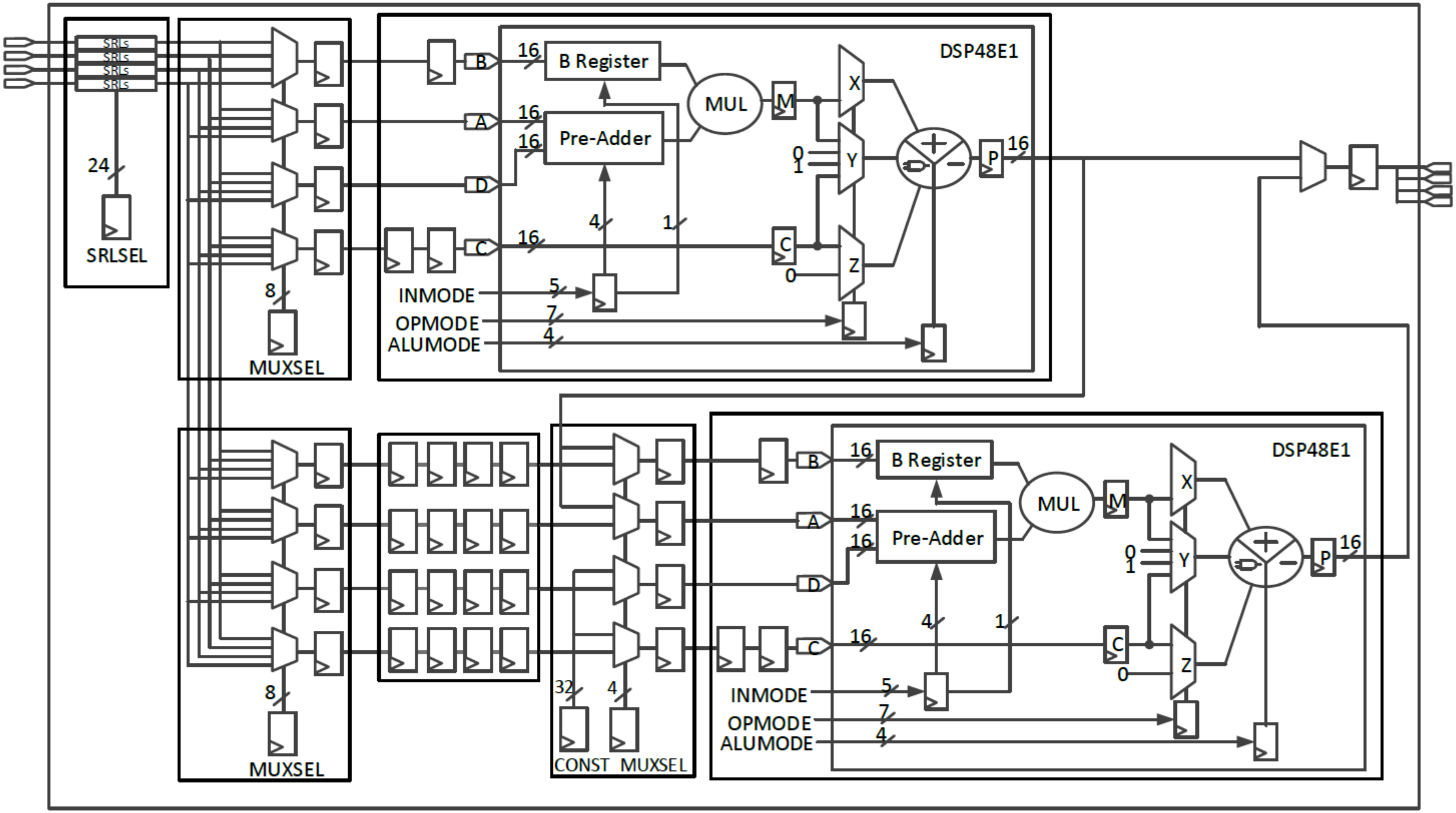
DSP scalability analysis (4N architecture)



DSP scalability analysis (8N architecture)



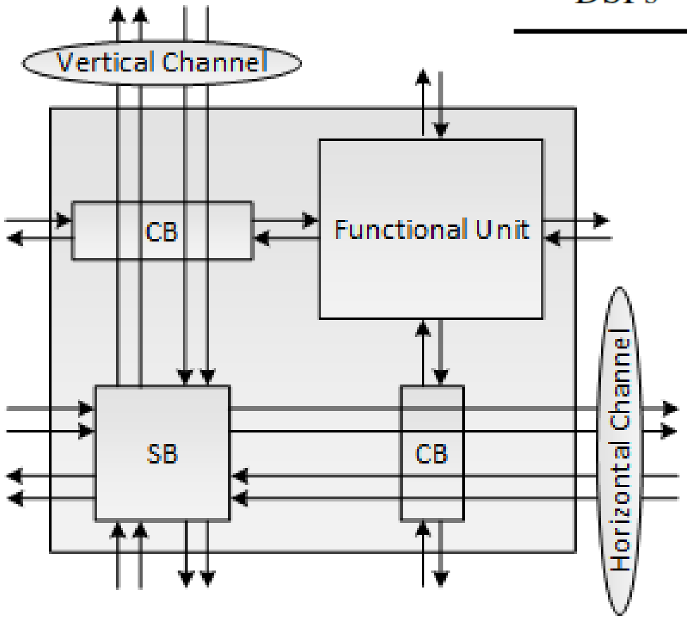
Enhanced FU using 2 DSP Blocks



Mapping overlay components on FPGA

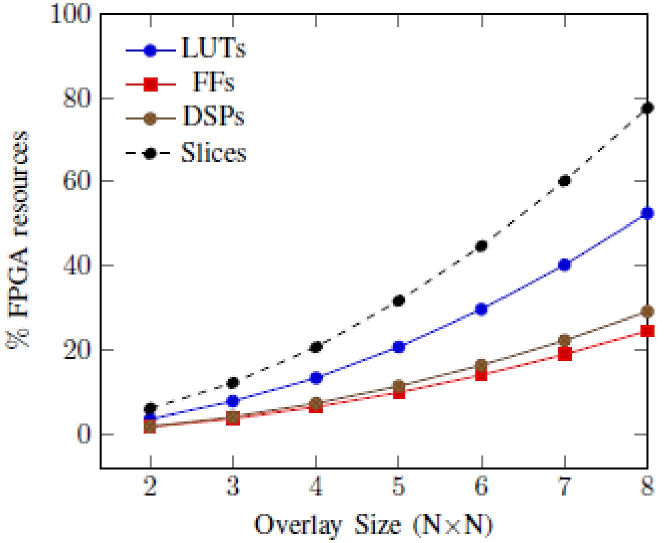
- **Resource usage of FU and interconnect blocks**

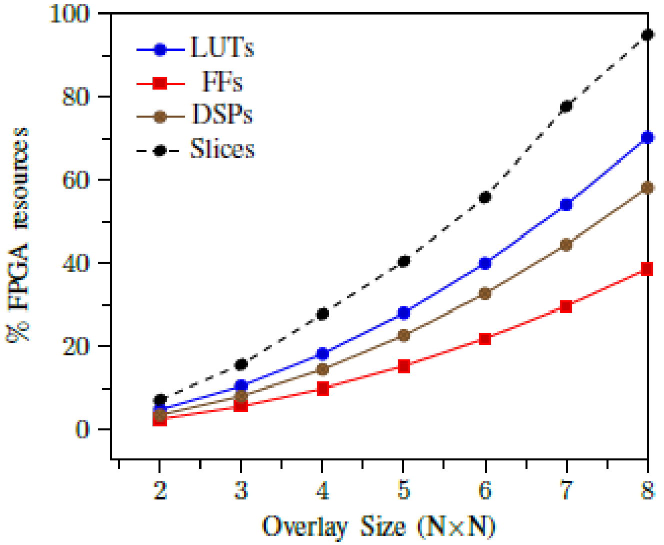
Resource	FU	FUCR	SB	SBCR	CB	CBCR
LUTs	360	0	64 (128)	0	48 (96)	0
FFs	432	109	0	8 (16)	64 (128)	6 (12)
DSPs	2	0	0	0	0	0



Single vs dual DSP block FU

- **Underutilization of DSP blocks (Only 30%) for 1 DSP-FU**
- **Using 2 DSP blocks in an FU:**
  - **Allows better use of remaining DSP blocks**
  - **Scalable: can fit an 8×8 array of FUs (128 DSPs) on Xilinx Zynq**
  - **For CW=2, consuming 37K LUTs with F_{max} = 300 MHz**
  - **LUTs/DSP = 289, LUTs/OP = 96 and Peak Throughput: 115 GOPS**

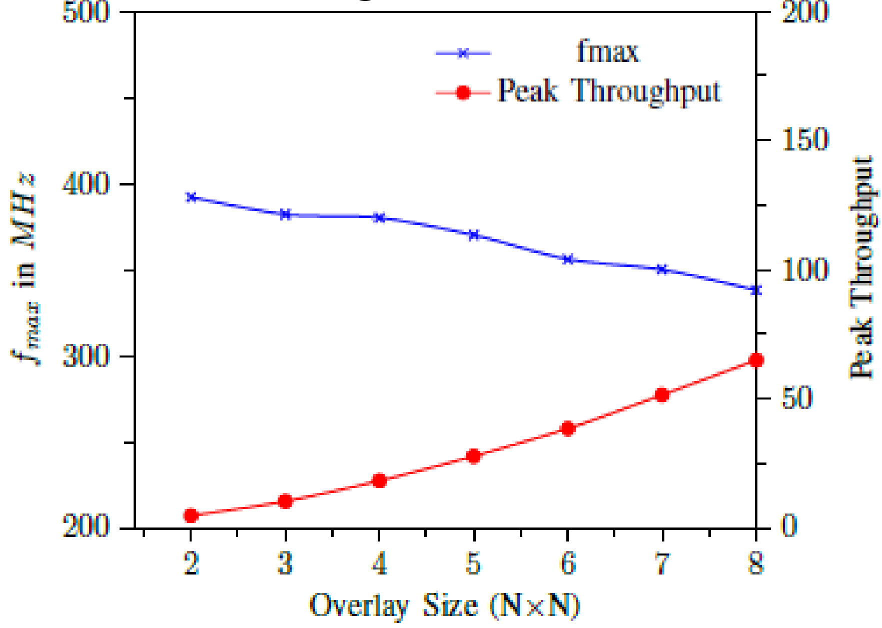




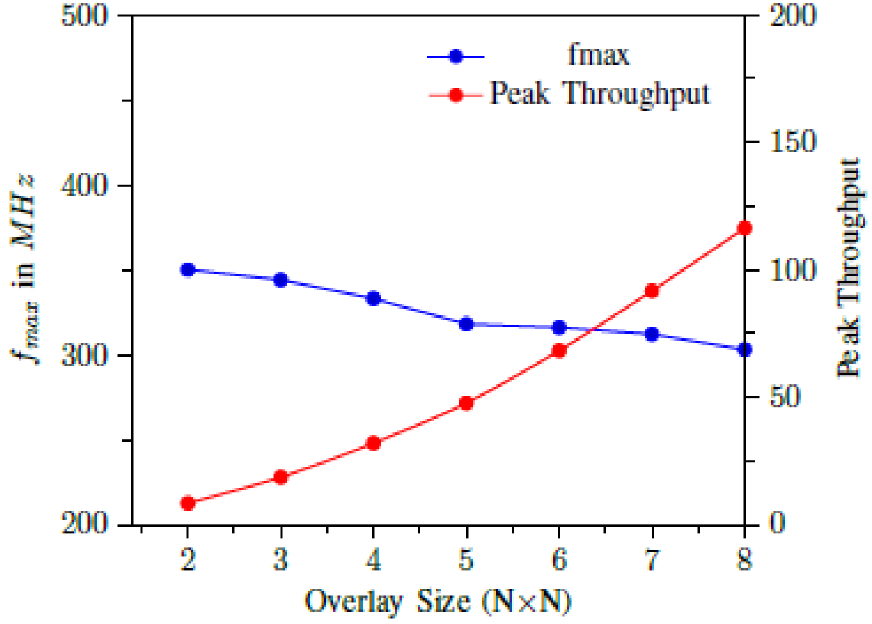
Single vs dual DSP block FU

- **Modest drop in frequency on scaling in both cases**
- **Using 2 DSP blocks in an FU allows:**
  - **doubling the peak throughput of the overlay**

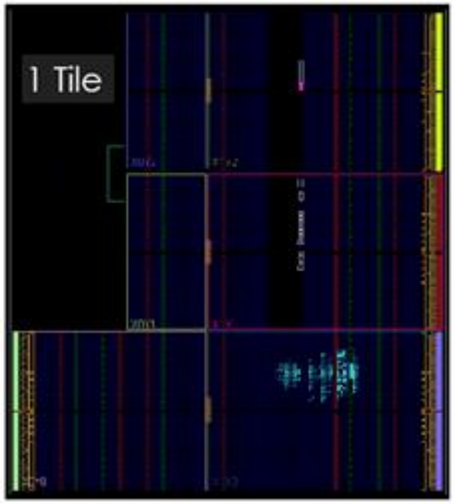
Single DSP block

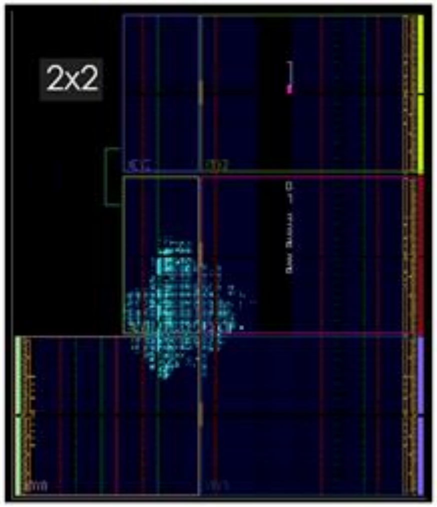


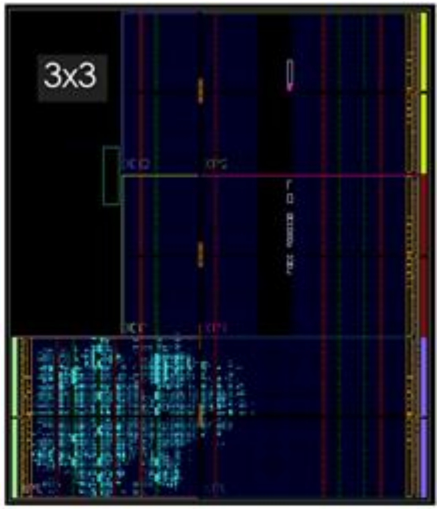
Two DSP blocks

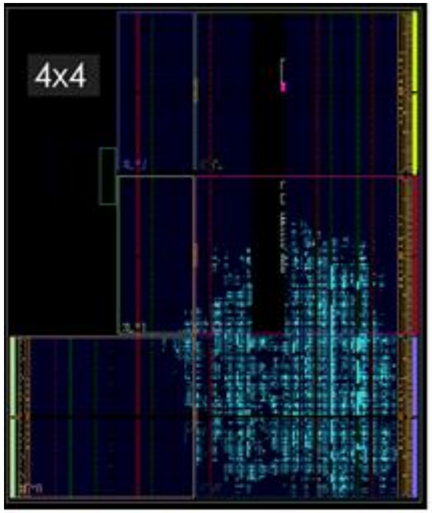


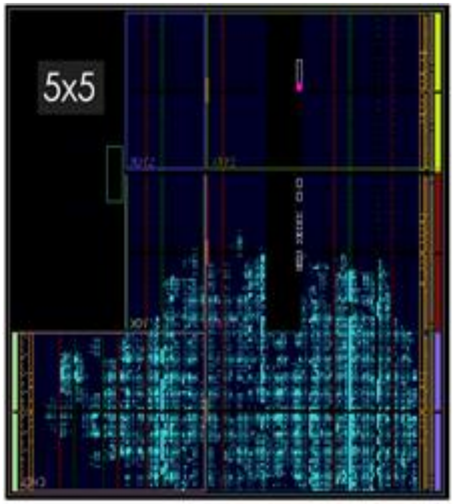
Mapping the overlay on an FPGA

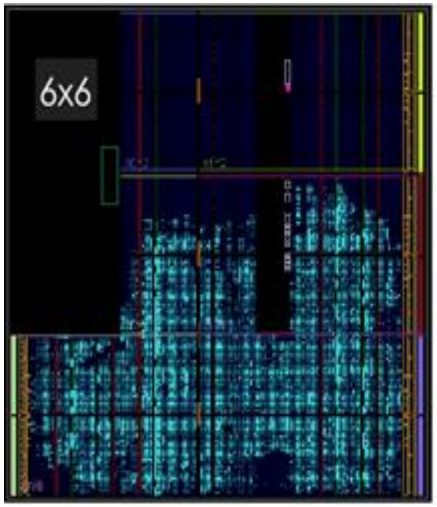


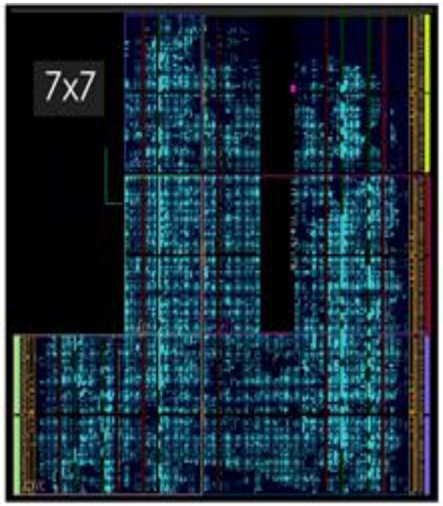


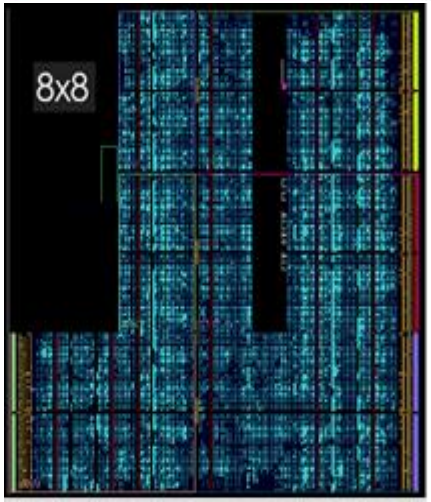






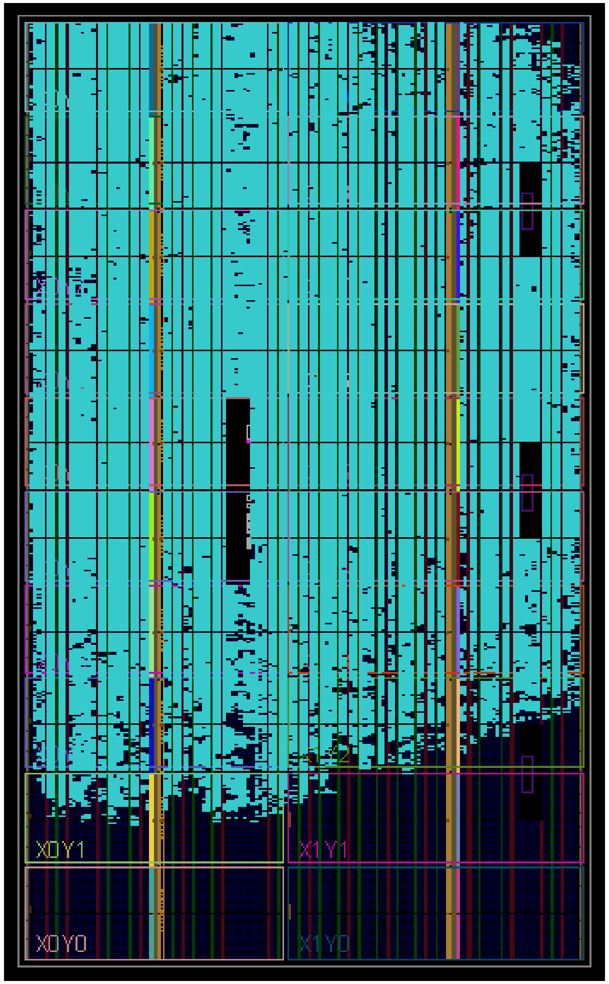






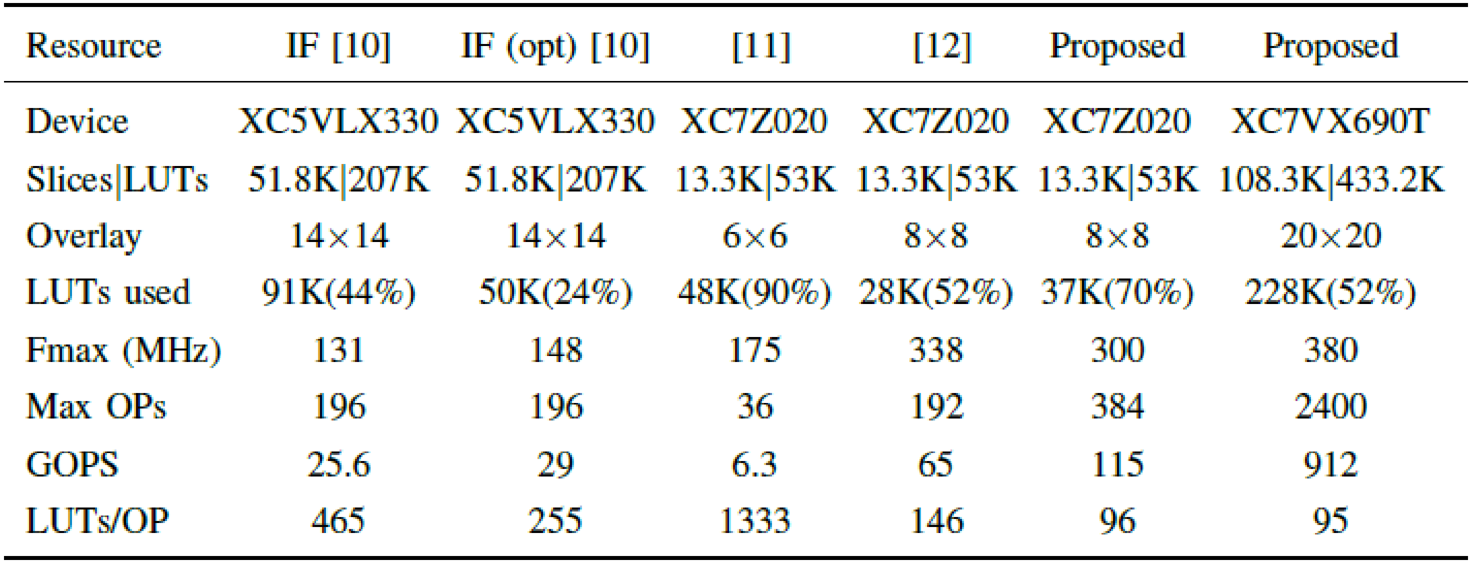
Scalability analysis

- 20×20 Overlay (800 DSP Blocks)
- Mapped on Virtex-7 (XC7VX690T)
- F_{max} = 380 MHz
- Can support up-to 2400 nodes
- Peak throughput of 912 GOPS
- Consuming 228K LUTs
- LUTs/DSP = 285
- LUTs/OP = 95



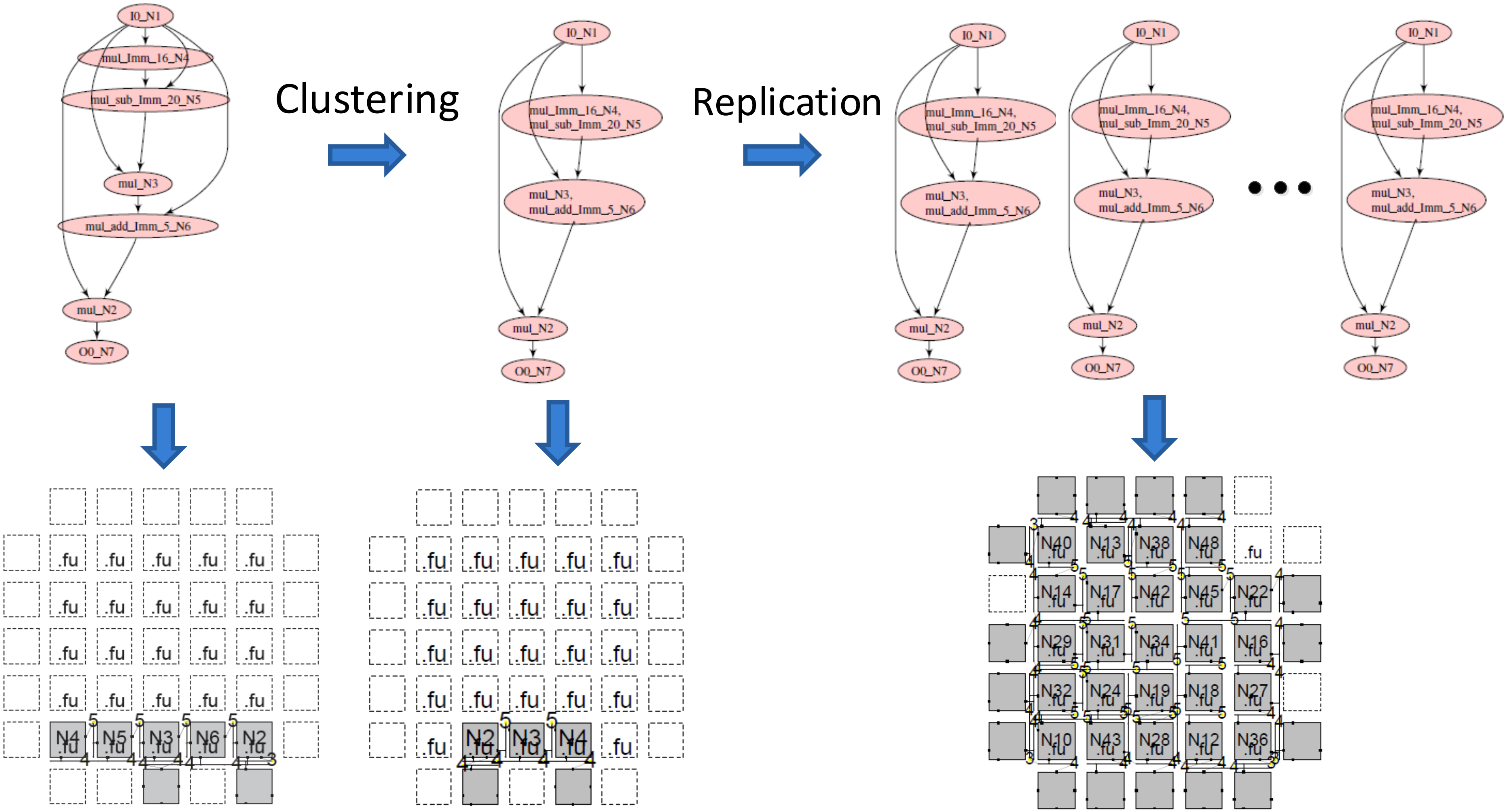
Quantitative comparison of overlays

- **Using proposed overlay**
  - **Significant reduction in LUTs/OP compared to others (Only 95 LUTs/OP)**
  - **Can support a significantly higher number of operations**
  - **Can fit an array of 128 DSP blocks on a Xilinx Zynq**
  - **Can fit an array of 800 DSP blocks on Virtex-7 device**
  - **Significant improvement in peak throughput (up to 912 GOPS)**



Resource	IF [10]	IF (opt) [10]	[11]	[12]	Proposed	Proposed
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
GOPS	25.6	29	6.3	65	115	912
LUTs/OP	465	255	1333	146	96	95

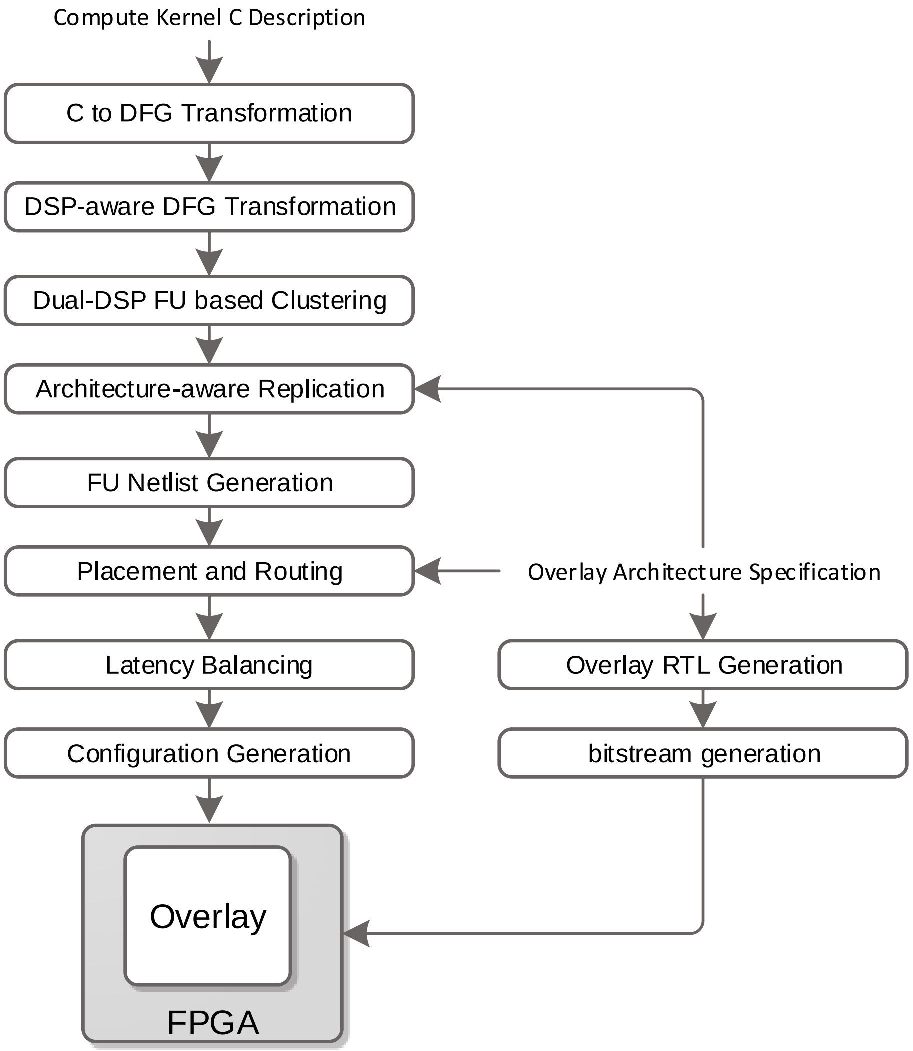
Compiling to the Overlay



Clustering

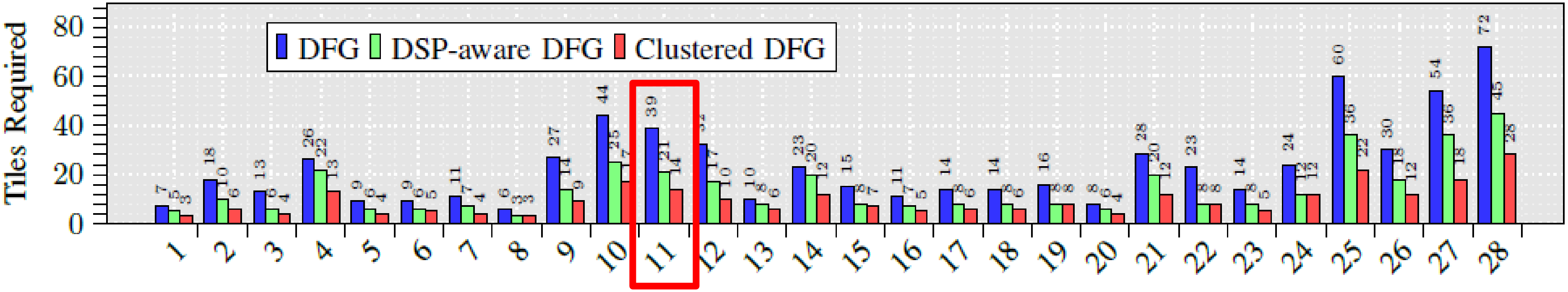
Replication

Automated tool-flow for kernel compilation



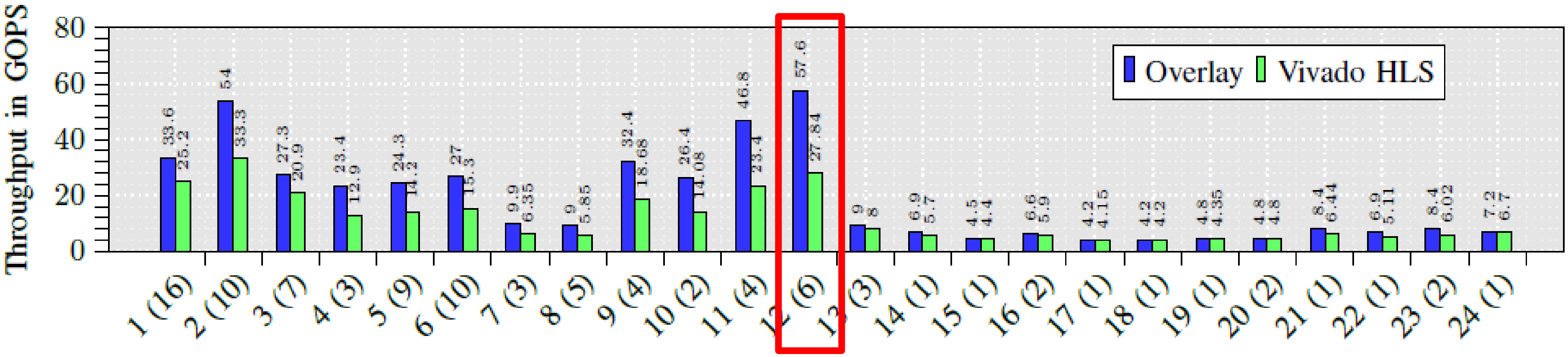
Mapping results

- Using proposed overlay and compilation method:
  - Reduction of up to 50% in required tiles for DSP-aware DFGs
  - Reduction of up to 69% in required tiles for clustered DFGs
- For example, Benchmark 11 (poly7)
  - A single DFG instance can fit onto an 8×8 overlay
  - Using DSP-aware clustered DFGs, 4 instances can fit
  - Utilises 56 of the 64 tiles



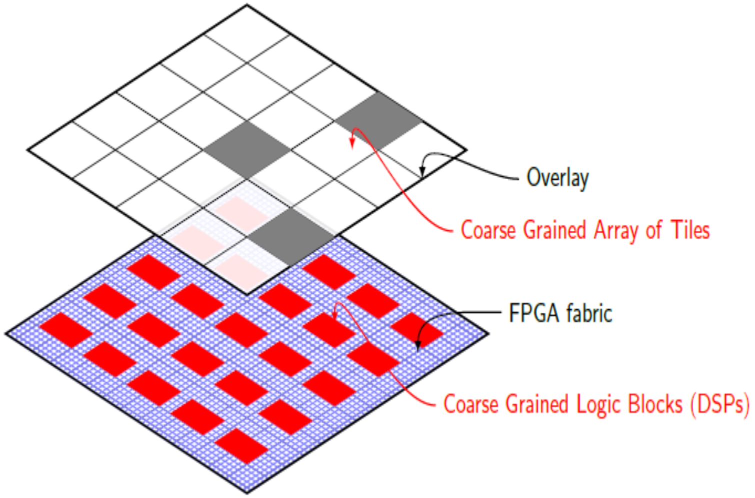
Kernel replication results

- **Using proposed overlay and compilation method:**
  - **Able to replicate kernel instances on the overlay**
  - **Achieve application throughput of up to 57.6 GOPS**
  - **Kernels with modest or low I/O requirements benefit from replication**
  - **A throughput of 9.6 GOPS for one instance of Benchmark 12 (poly6)**
  - **Can map 6 instances of Benchmark 12 to achieve 6 times throughput.**
  - **Average throughput improvement of 40% over Vivado HLS**
  - **Due to highly pipelined architecture of the overlay**



Overlay reconfiguration latency

- **1137 Bytes of configuration data**
- **Compared to 4 MB for the entire Zynq Fabric**
- **Zynq fabric can be reconfigured in 31.6 ms.**
- **Entire overlay can be reconfigured in 45 us.**
- **1000x faster reconfiguration**


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

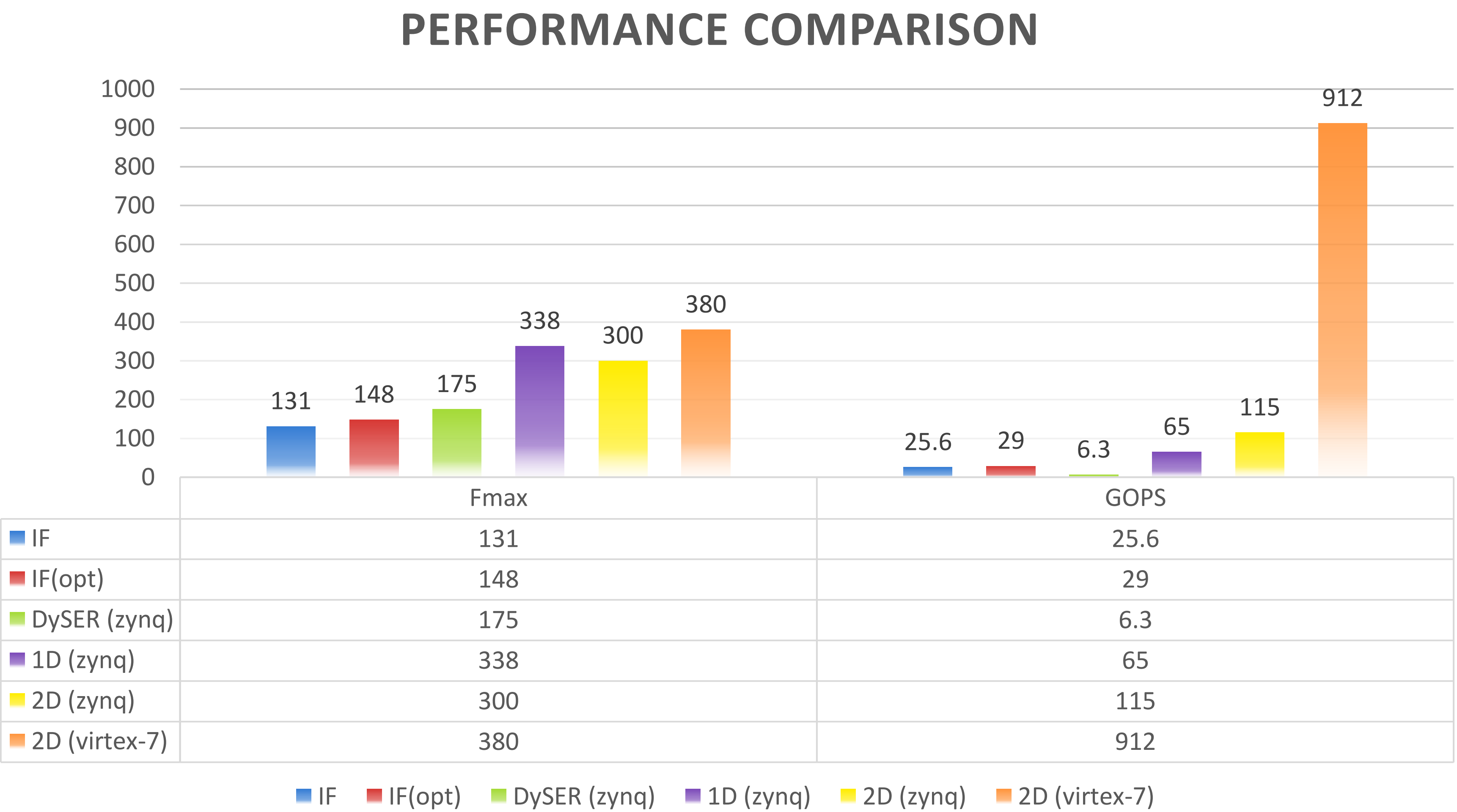
Conclusions

- Presented a throughput oriented FPGA Overlay
- Built using fully pipelined DSP blocks
- With significantly improved performance
  - An 8×8 array of FUs (128 DSPs) with an F_{max} 300 MHz on Xilinx Zynq
  - Can support up-to 384 operations
  - A peak throughput of 115 GOPS (2× than single DSP-FU)
  - LUTs/ OP of 96 (33% less than single DSP-FU)
- Analysis of compute kernels
  - To justify the use of 2 DSP blocks in an FU for island-style overlays
- An automated tool-flow for kernel compilation
- Average throughput improvement of 40% over Vivado HLS

Future Work

- **Lower overhead interconnect architecture**
- **Integration of Overlay with general purpose processor**
  - **Memory subsystem for communication**
  - **Embedded applications on Zynq**
  - **Cloud applications using DyRACT partial reconfiguration**
- **OpenCL support in the toolflow**
  - **Comparison with OpenCL synthesis**
- **Comparison against time-multiplexed overlays**

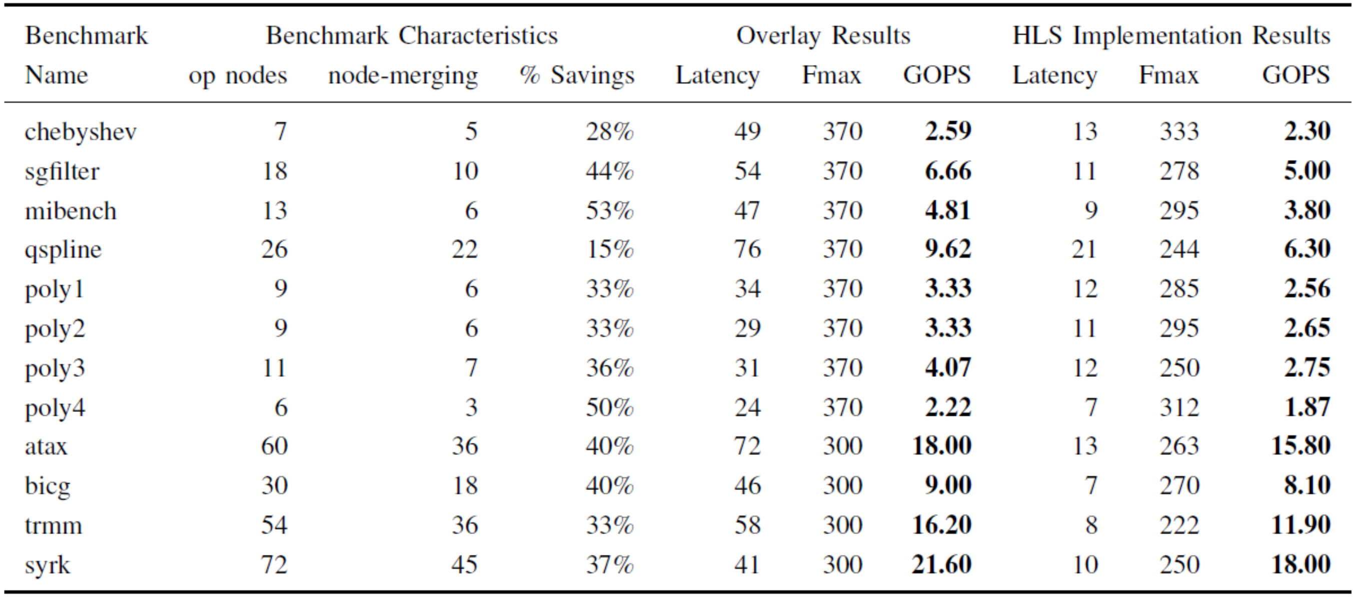
Back-up Slides



PERFORMANCE COMPARISON

	Fmax	GOPS
🟦 IF	131	25.6
🟥 IF(opt)	148	29
🟩 DySER (zynq)	175	6.3
🟪 1D (zynq)	338	65
🟨 2D (zynq)	300	115
🟧 2D (virtex-7)	380	912

Back-up Slides



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		