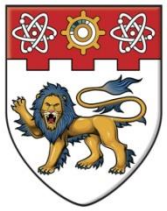


NANYANG TECHNOLOGICAL UNIVERSITY

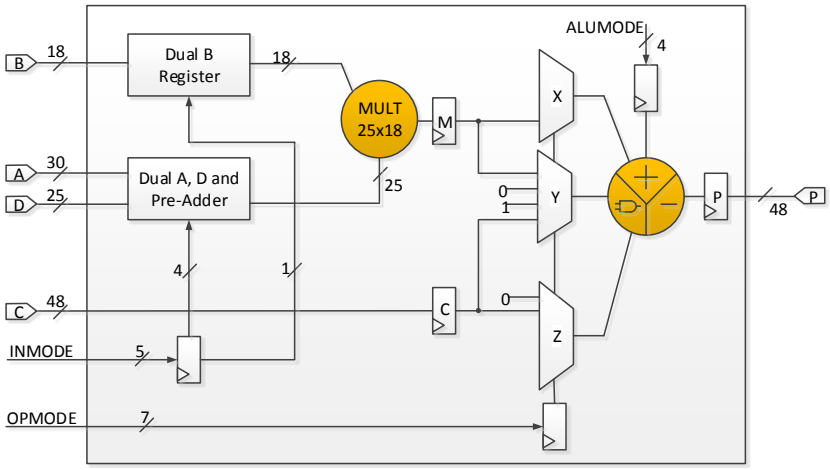
Efficient Overlay Architecture Based on DSP Blocks

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

International Symposium On Field-Programmable Custom Computing Machines (FCCM)
4th May 2015, Vancouver, Canada

Coarse Grained Overlay Architectures

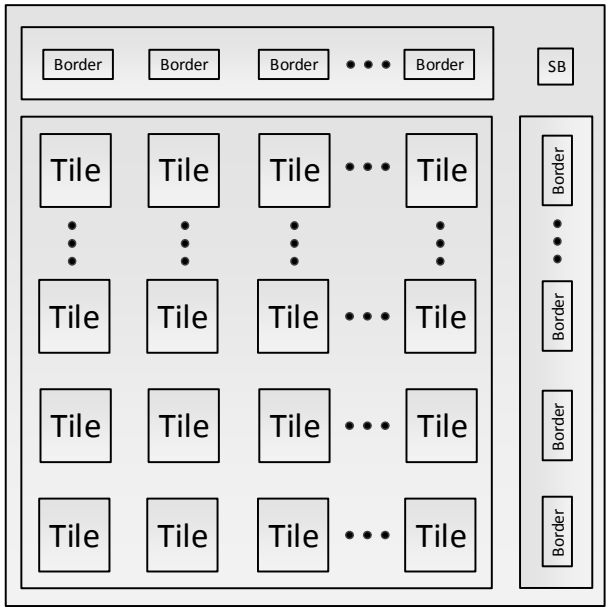
- Accelerator Design at a higher level of abstraction
- Fast compilation and development cycles
- Improved design productivity
- Cost: area and performance overheads
- High performance hard macros in the underlying FPGA architecture
- Exploit programmability feature of DSP Blocks
  - fully pipelined processing elements
  - maximize frequency and throughput

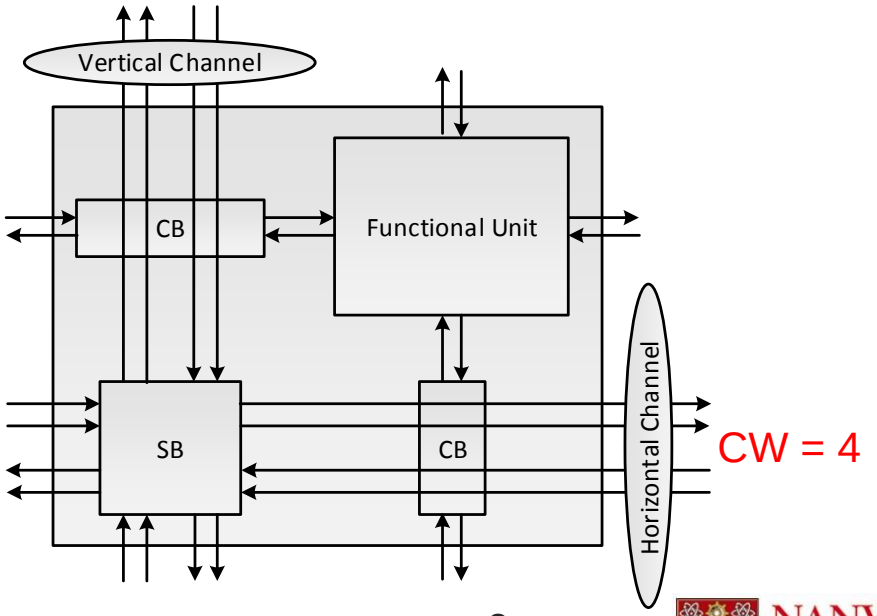


NANYANG TECHNOLOGICAL UNIVERSITY

Proposed Overlay

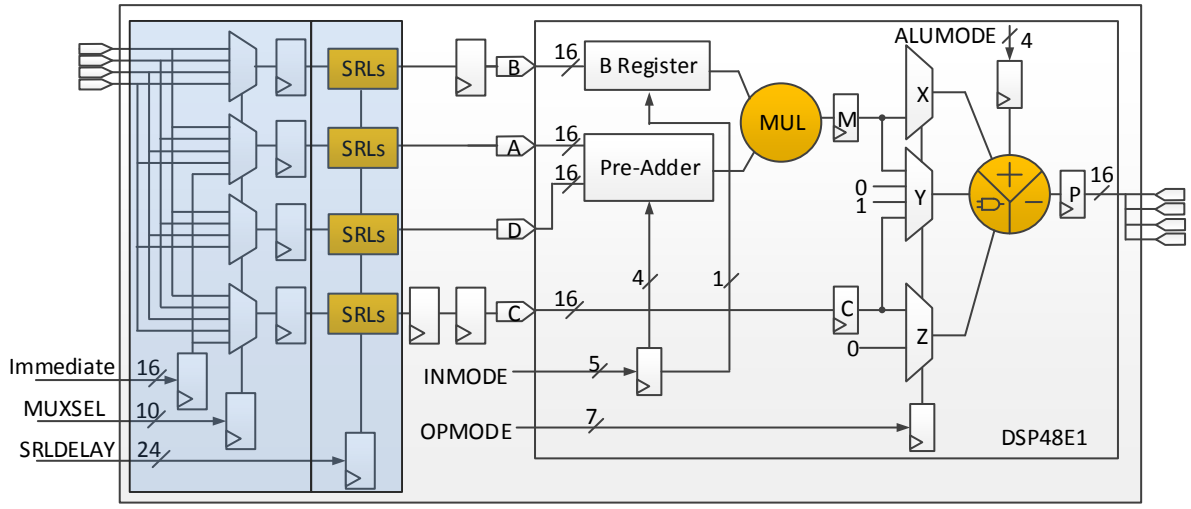
- 2D array of Tiles
  - Programmable functional unit and routing resources in each Tile
  - Functional units interconnected via island-style routing network
- Coarse grained programmable routing resources





DSP Block based Functional Unit (FU)

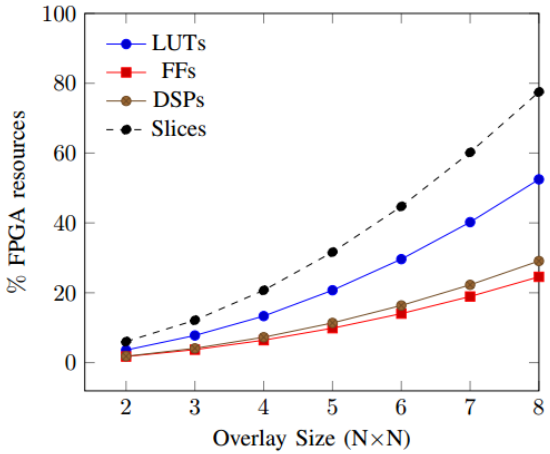
- Fully pipelined DSP48E1 as a programmable PE
- Add/sub, a multiplier and an ALU inside the PE
- Achievable frequency near theoretical limits
- 400 MHz on the Xilinx Zynq device (XC7Z020 CLG484-1)
- MUX based reordering logic
- SRL based variable-length shift registers



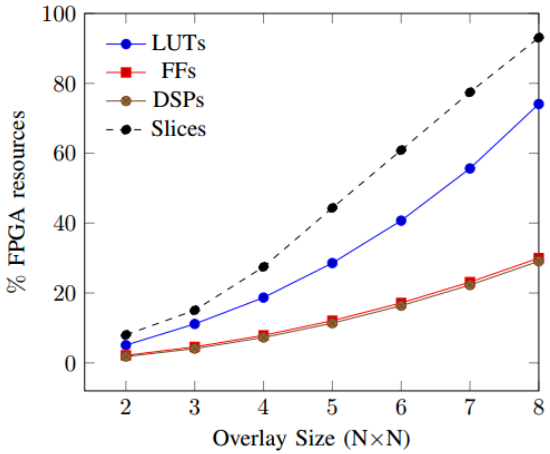
NANYANG TECHNOLOGICAL UNIVERSITY

Mapping to the Xilinx Zynq device

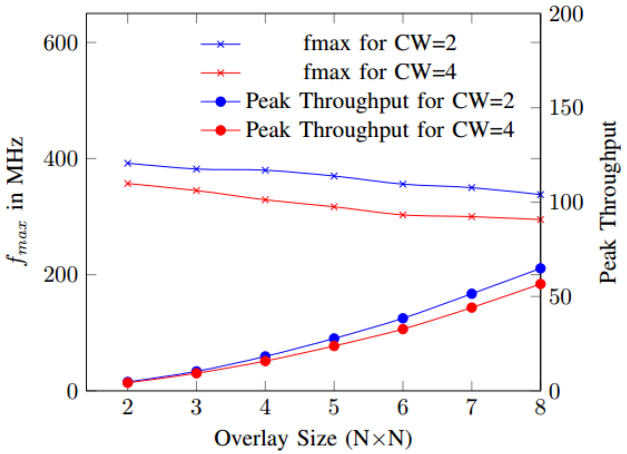
- Underutilization of DSP blocks (Only 30% were used)
- Modest drop in frequency on scaling
- A frequency of 300 MHz for an 8x8 overlay (CW=4)



(a) Resources usage for CW=2.



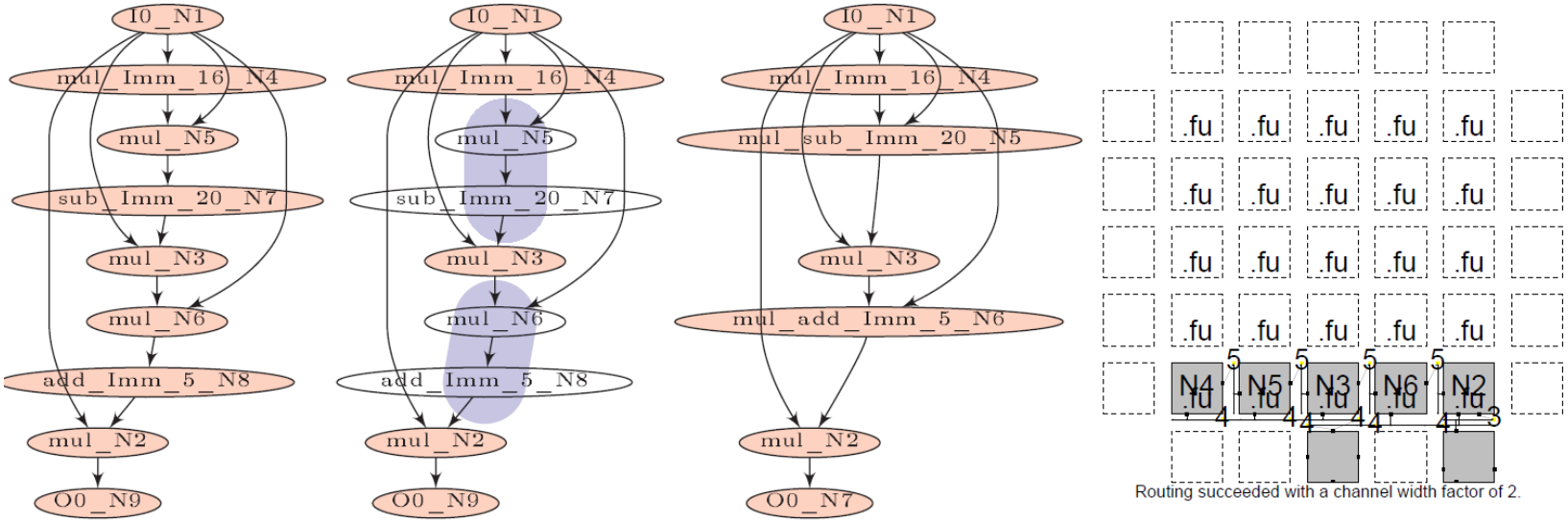
(b) Resources usage for CW=4.



(c) Frequency trend.

NANYANG TECHNOLOGICAL UNIVERSITY

Automated Mapping of Compute kernels



- Step 1: C to data flow graph (DFG) transformation
- Step 2: DSP48E1 aware mapping
- Step 3: Placement and routing onto the overlay
- Step 4: Latency balancing and configuration generation

Experimental Evaluation

Benchmark	Benchmark Characteristics				Routability		Overlay Results			HLS Implementation Results				
	i/o nodes	op nodes	merged nodes	savings	CW=2	CW=4	Latency	MLI	GOPS	Latency	Fmax	GOPS	Slices	DSPs
chebyshev	1/1	7	5	28%	3×3	3×3	49	36	2.59	13	333	2.3	24	3
sgfilter	2/1	18	10	44%	4×4	4×4	54	31	6.66	11	278	5.0	40	8
mibench	3/1	13	6	53%	3×3	3×3	47	35	4.81	9	295	3.8	81	3
qspline	7/1	26	22	15%	5×5	5×5	76	64	9.62	21	244	6.3	126	14
poly1	2/1	9	6	33%	3×3	3×3	34	22	3.33	12	285	2.56	62	4
poly2	2/1	9	6	33%	3×3	3×3	29	7	3.33	11	295	2.65	45	4
poly3	6/1	11	7	36%	3×3	3×3	31	11	4.07	12	250	2.75	52	6
poly4	5/1	6	3	50%	2×2	2×2	24	12	2.22	7	312	1.87	36	3
atax	12/3	60	36	40%	—	6×6	72	58	18.0	13	263	15.8	78	18
bicg	15/6	30	18	40%	—	6×6	46	32	9.0	7	270	8.1	91	18
trmm	18/9	54	36	33%	—	7×7	58	30	16.2	8	222	11.9	105	36
syrk	18/9	72	45	37%	—	7×7	41	19	21.6	10	250	18	237	24

	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		

Future Work

- Area reduction of the overlay through careful optimizations of:
  - routing architecture
  - and synchronization logic.
- Alternative interconnect architectures for a low overlay routing network