PiCaSO: A Scalable and Fast Processor-in-Memory Overlay

MD Arafat Kabir

UNIVERSITY OF ARKANSAS

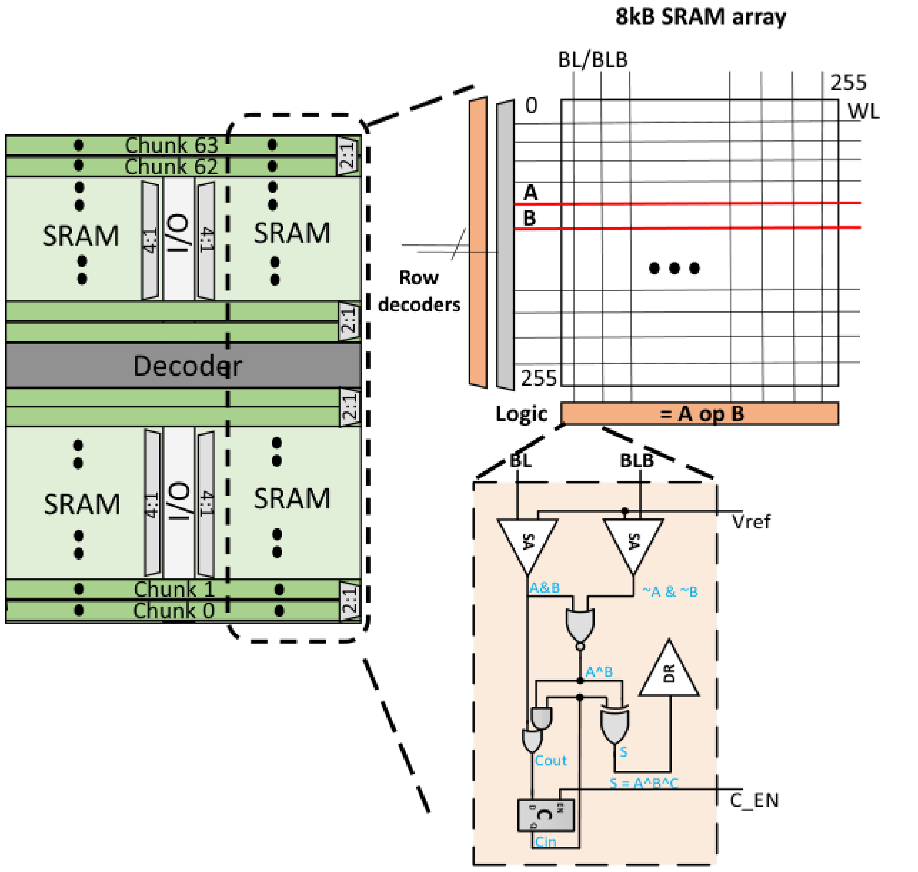
FCCM 2023 (Poster)
Making BRAMs Compute: Creating Scalable Computational Memory Fabric Overlays

FPL 2023 (Under Review)
FPGA Processor In Memory Architectures (PIMs): Overlay or Overhaul ?

Background

Processor-in-Memory (PIM) Architecture

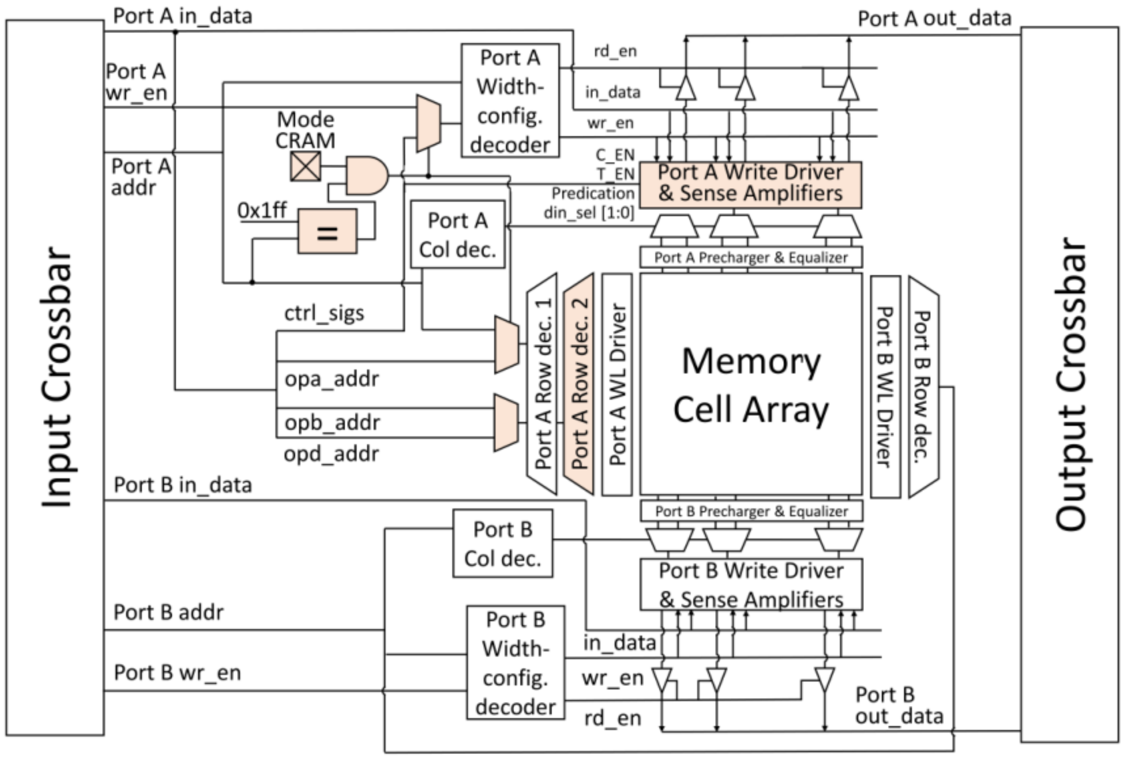
- Processing Elements (PEs) close to memory
- Breaking von Neumann memory bottleneck
- BRAMs distributed throughout FPGAs
- Improve memory intensive application performance
- Proposals modifying BRAM tile to PIM tile
    - CCB
    - CoMeFa

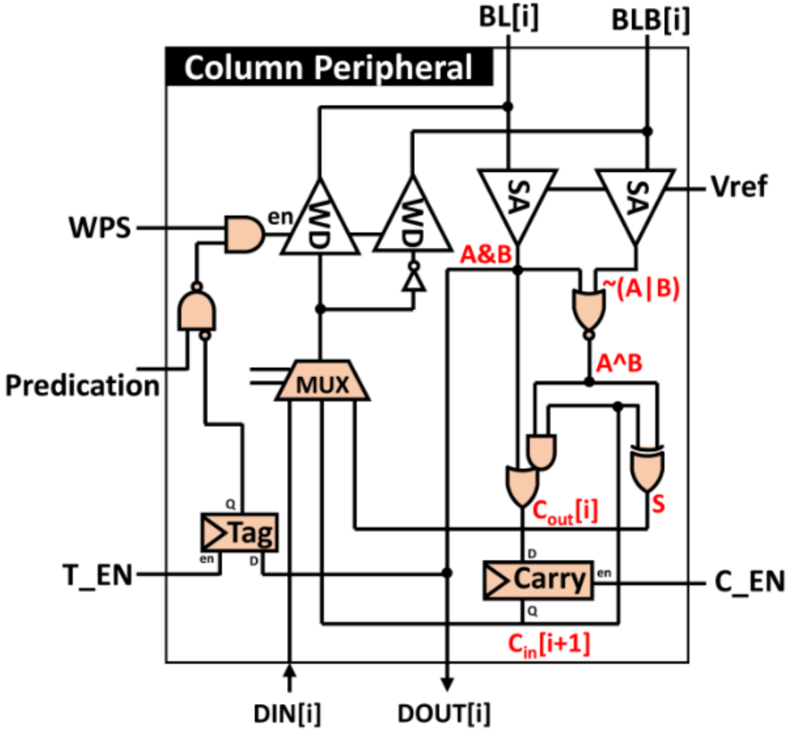


Neural Cache, ISCA 2018

Compute Capable BRAM (CCB)

- Based on SRAM Neural Cache (ISCA’18)
- Activate 2 wordlines at a time (requires modified voltage source)
- PE is equivalent of a Full-Adder

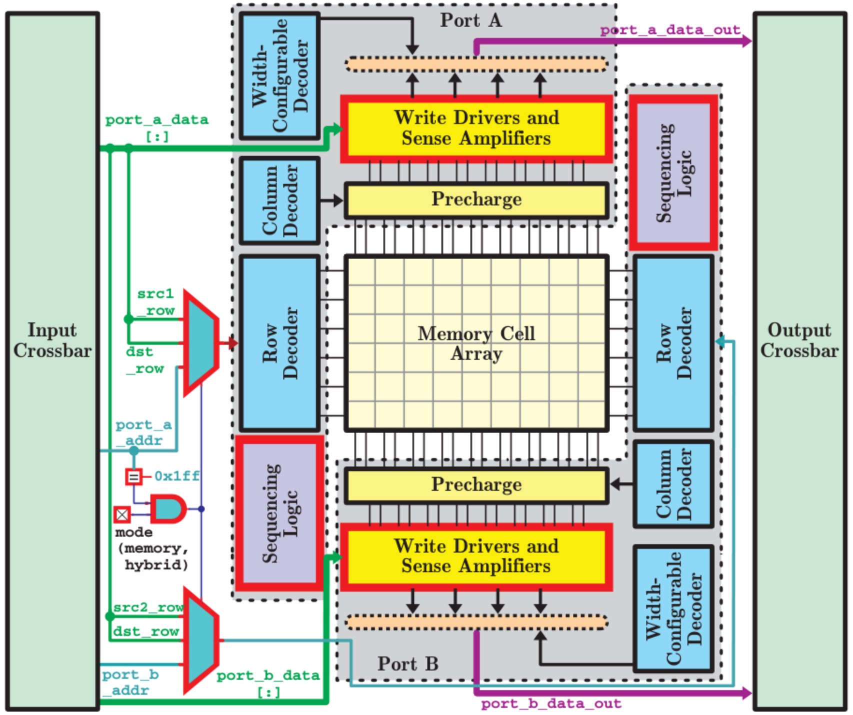


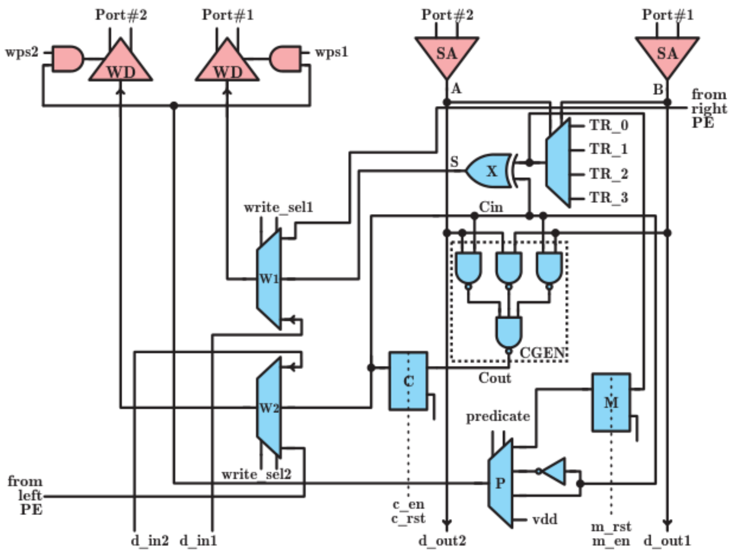


- Wang, Xiaowei, Vidushi Goyal, Jiecao Yu, Valeria Bertacco, Andrew Boutros, Eriko Nurvitadhi, Charles Augustine, Ravi Iyer, and Reetuparna Das. "Compute-Capable Block RAMs for Efficient Deep Learning Acceleration on FPGAs." In 2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), pp. 88-96. IEEE, **2021**.

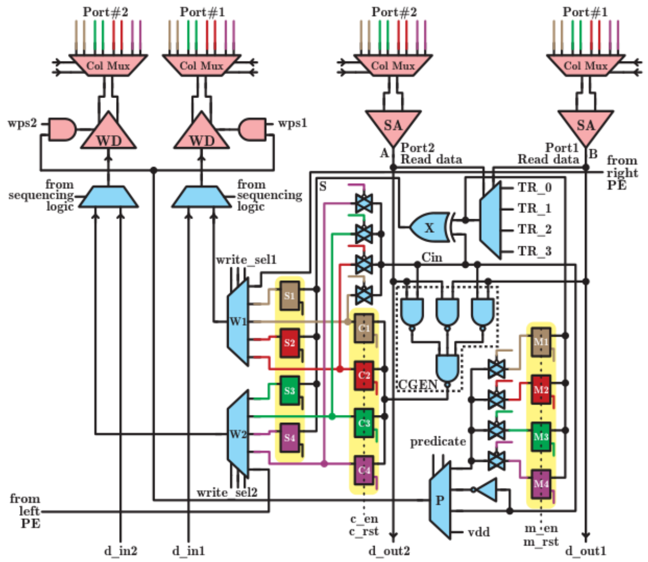
Compute-in-**Me**mory Blocks for **FPGA**s (CoMeFa)

- Improved upon CCB, using dual-port nature of BRAMs
- Does not require voltage source modification, requires SA cycling (CoMeFa-A)
- PE can be configured to implement any bit-wise operation (AND, OR, XOR, etc.)





CoMeFa-D: Delay Optimized

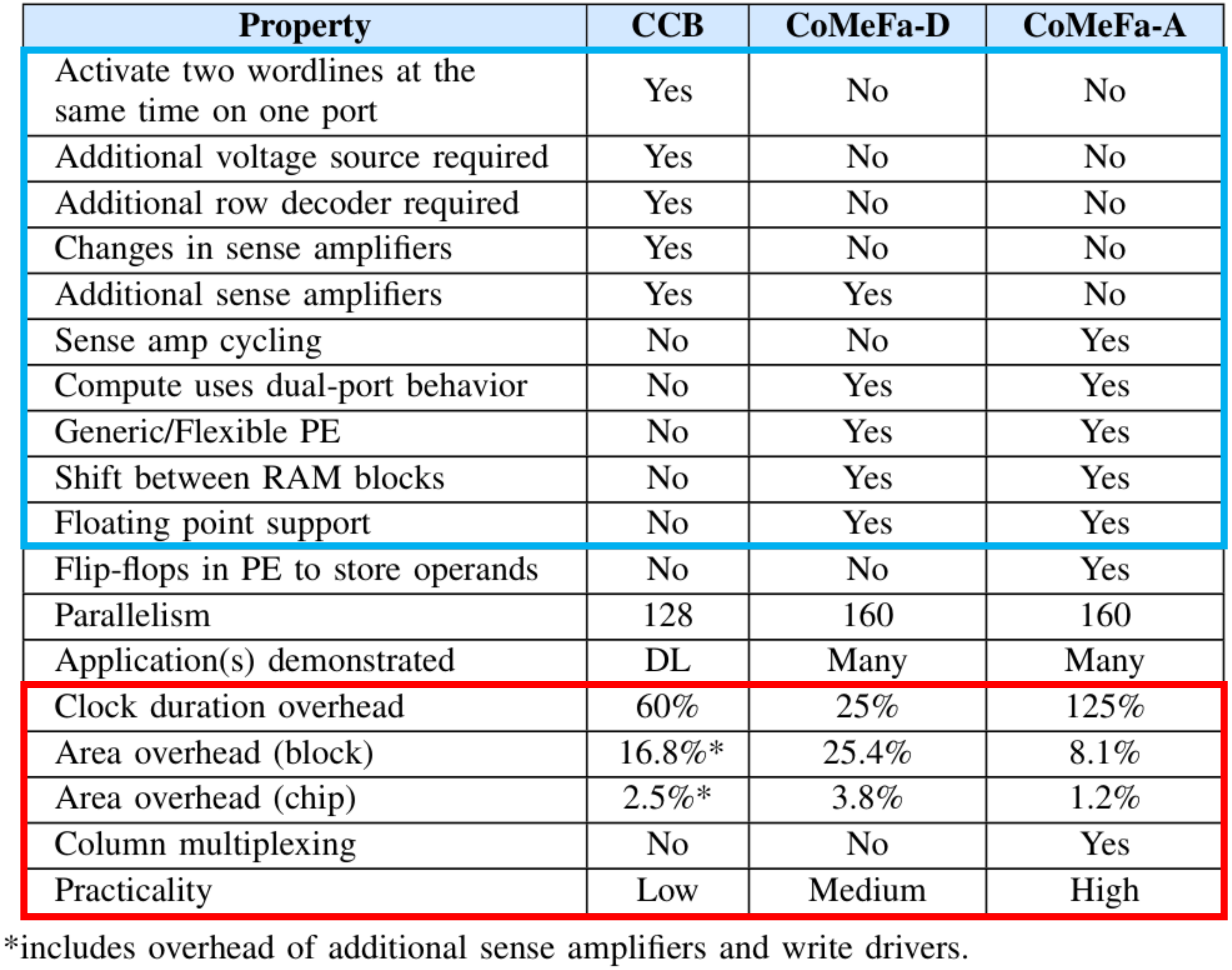


CoMeFa-A: Area Optimized

- Arora, Aman, Tanmay Anand, Aatman Borda, Rishabh Sehgal, Bagus Hanindhito, Jaydeep Kulkarni, and Lizy K. John. "CoMeFa: Compute-in-Memory Blocks for FPGAs." In *__2022__ IEEE 30th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 1-9. IEEE, 2022.

Comparison Summary: CCB vs CoMeFa

- 1.25x – 2.25x slower
- 8.1% – 25.4% bigger



Property	CCB	CoMeFa-D	CoMeFa-A
Activate two wordlines at the same time on one port	Yes	No	No
Additional voltage source required	Yes	No	No
Additional row decoder required	Yes	No	No
Changes in sense amplifiers	Yes	No	No
Additional sense amplifiers	Yes	Yes	No
Sense amp cycling	No	No	Yes
Compute uses dual-port behavior	No	Yes	Yes
Generic/Flexible PE	No	Yes	Yes
Shift between RAM blocks	No	Yes	Yes
Floating point support	No	Yes	Yes
Flip-flops in PE to store operands	No	No	Yes
Parallelism	128	160	160
Application(s) demonstrated	DL	Many	Many
Clock duration overhead	60%	25%	125%
Area overhead (block)	16.8%*	25.4%	8.1%
Area overhead (chip)	2.5%*	3.8%	1.2%
Column multiplexing	No	No	Yes
Practicality	Low	Medium	High

*includes overhead of additional sense amplifiers and write drivers.

PIM Overlay Motivation

- Custom PIM tile not readily available
- Custom PIMs significantly degrades clock frequency
- Overlays are portable between devices
- Overlays are easily reconfigurable

Questions

1. Can PIM overlays provide competitive performance?
2. If yes, what is the cost in terms of utilization?
3. Does overlays scale well at the device level?

PiCaSO Design Goals

- PiCaSO aims to be a memory-centric design
- System performance determined mainly by memory resources
  - BRAM is the bottleneck

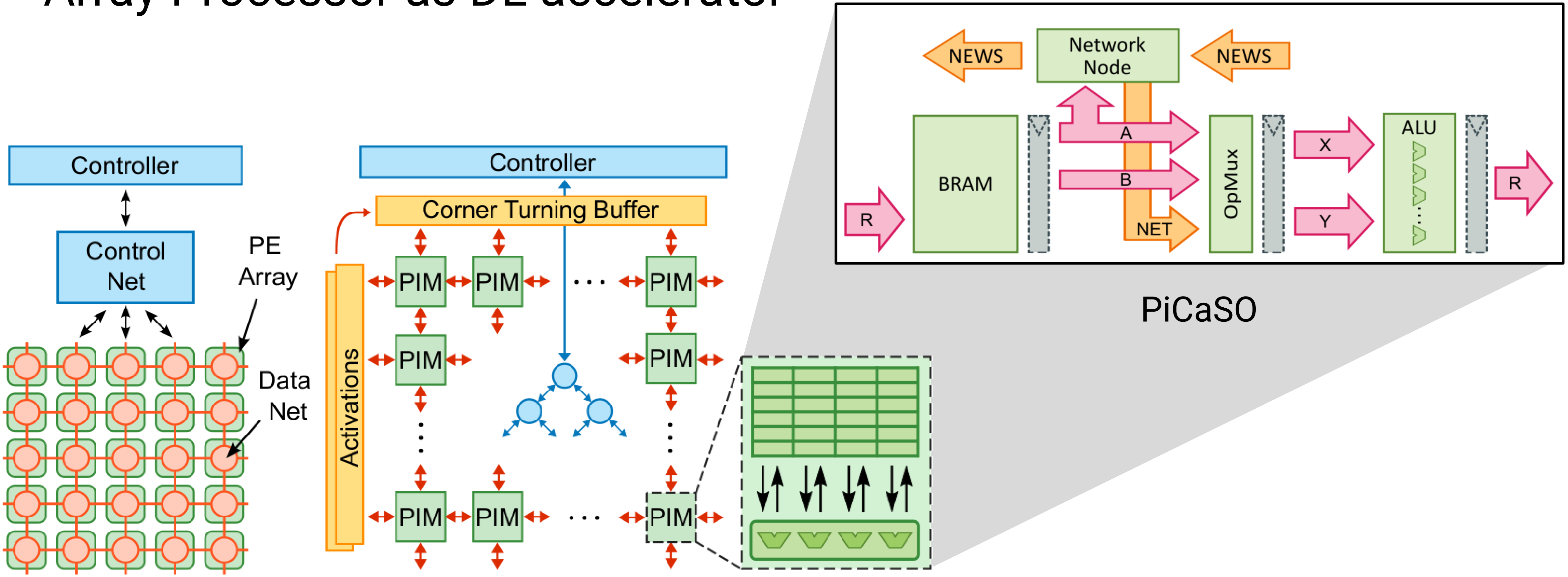
Design Goals

1. Run as fast as the maximum BRAM frequency
2. Scale linearly with BRAM capacity of target device
3. Use minimum logic resources from FPGA fabric

Architecture

PiCaSO Architecture

- Array Processor as DL accelerator

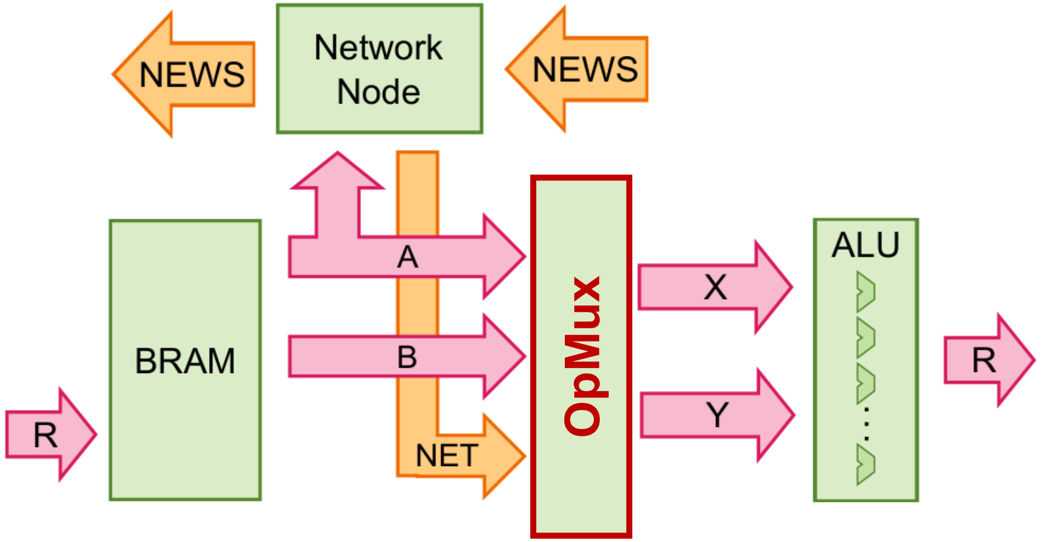


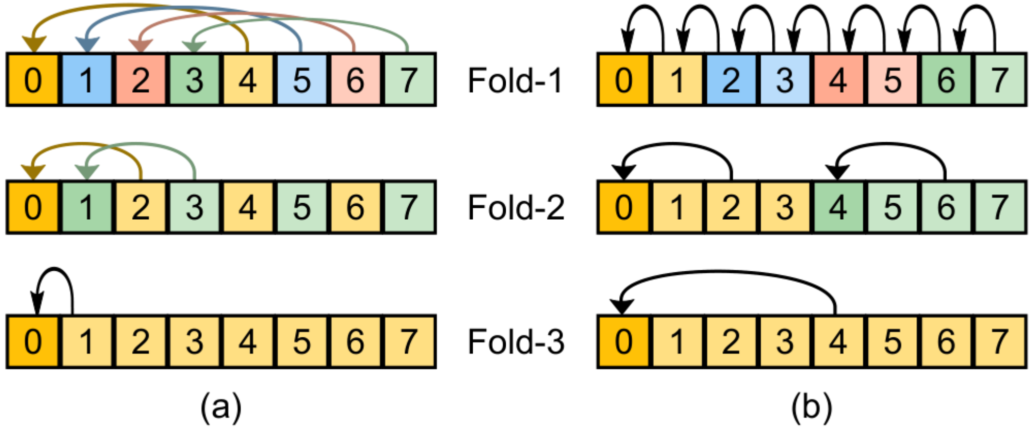
PiCaSO

PIM Overlay-based Array Processor

Operand Multiplexer (OpMux)

- Converts A/B/NET to X/Y
- Eliminates copy between bitlines
- Overlaps data move and comp.
- Employs “Folding Patterns”

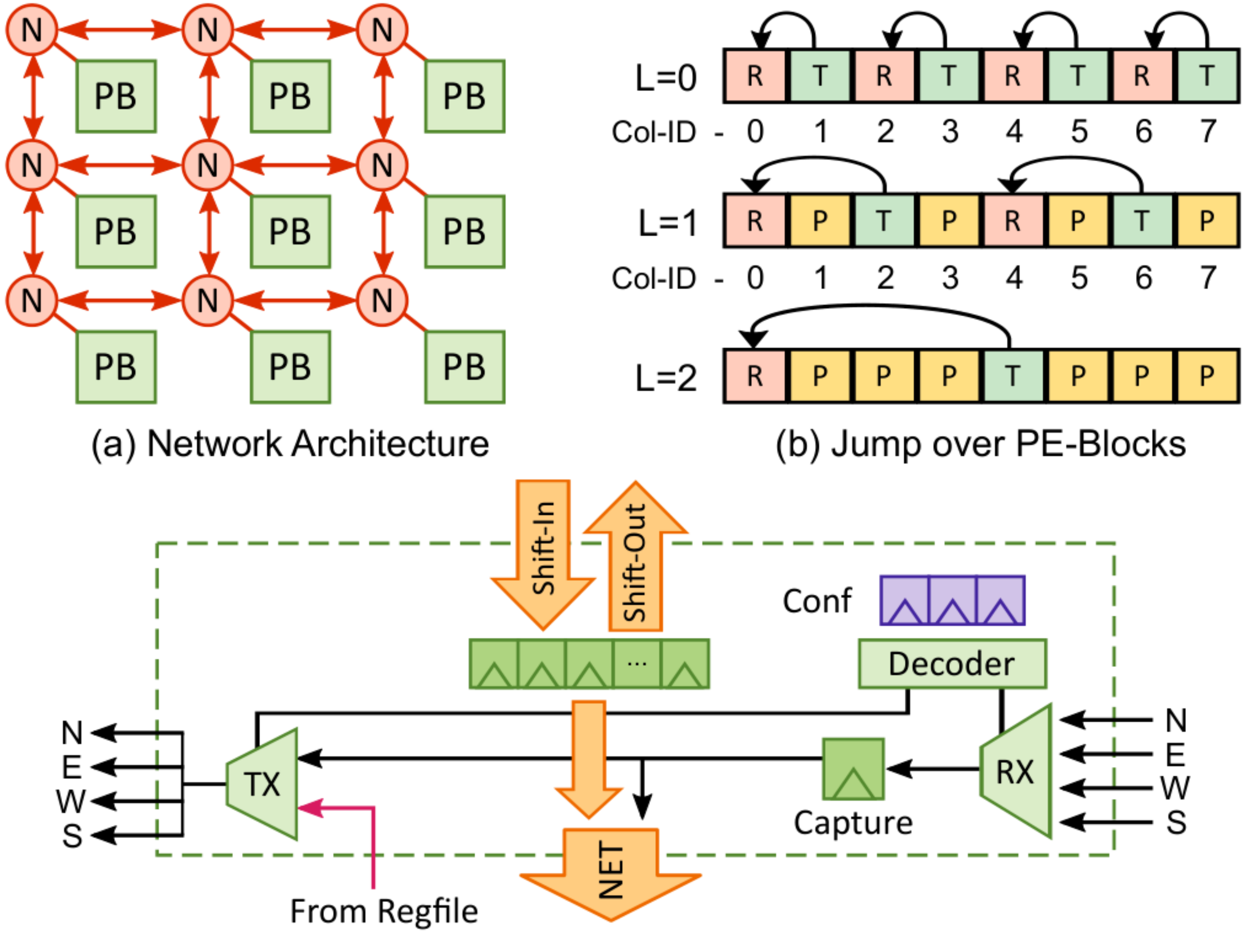




Two possible folding patterns in OpMux

Data Movement Network

- A binary hopping network
- Specialized network node
- 8-bit Configuration
  - Can handle 1 million PEs
- 1 node per BRAM (16 PEs)
  - Total 2 slices per node



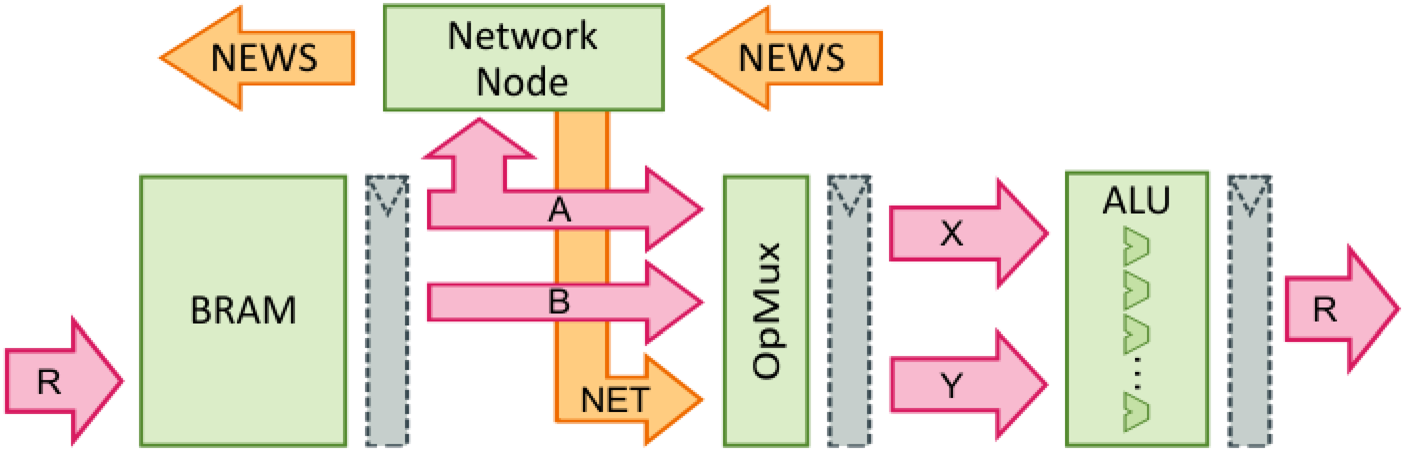
(a) Network Architecture

(b) Jump over PE-Blocks

(c) Network node (N) architecture for hopping

Pipeline Configurations

- Three potential points for pipelining
- We explored four configurations
  - Single Cycle: none of the stages are enabled
  - RF-Pipe: BRAM stage enabled
  - Op-Pipe: OpMux stage enabled
  - Full-Pipe: All stages enabled

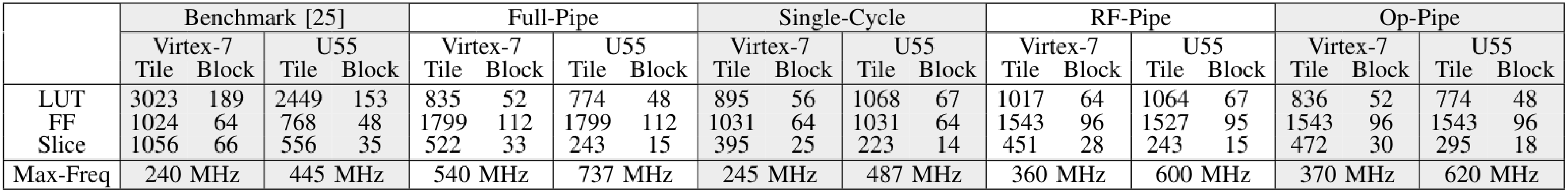


Analysis

Utilization and Performance

- Utilization numbers for 4x4 array of PIM (Tile) on Virtex-7 and Alveo U55 (US+)
- Full-Pipe: 2.24x, 1.67x faster, 2x smaller
- Single-Cycle: Similar speed, 2.6x and 2.5x smaller
- RF/Op-Pipe: Better than Single-Cycle, Op-Pipe minimizes network latency
- Full-Pipe meets one of the design goals
  - Runs as fast as the BRAM max frequency (543.77 for Virtex-7, 737 MHz for US+)

Comparison between tiles of 4 × 4 PE-Blocks of different overlay configurations

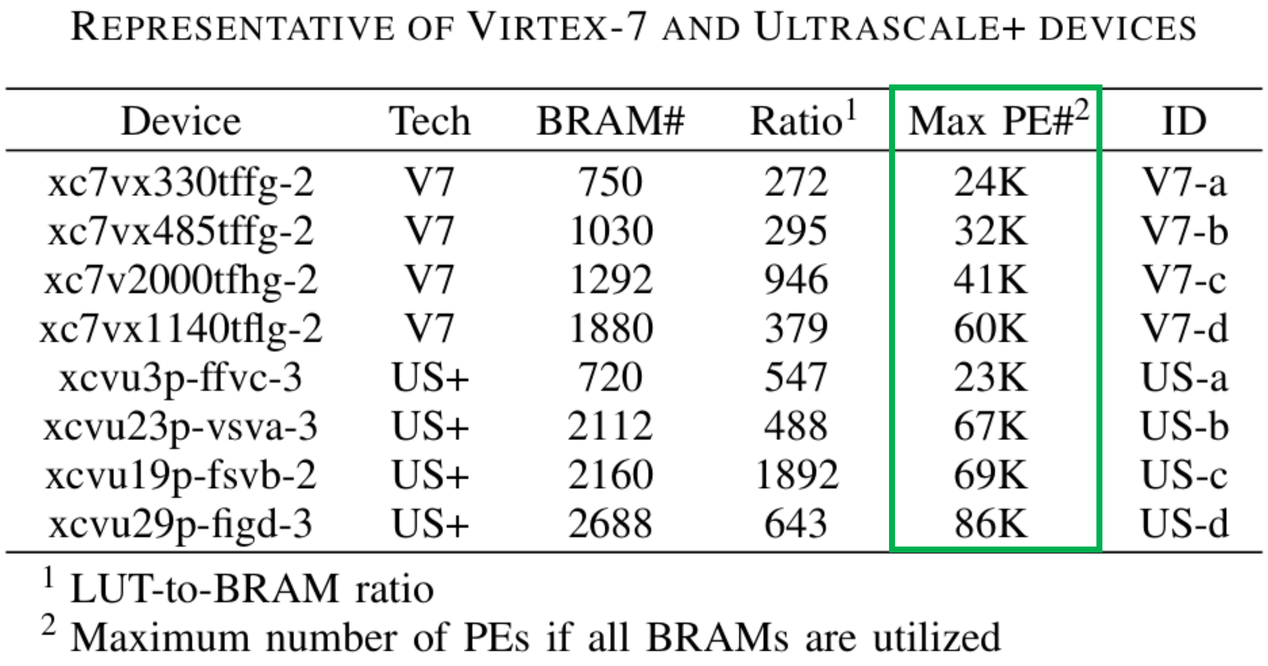


	Benchmark [25]				Full-Pipe				Single-Cycle				RF-Pipe				Op-Pipe			
	Virtex-7		U55		Virtex-7		U55		Virtex-7		U55		Virtex-7		U55		Virtex-7		U55	
	Tile	Block	Tile	Block	Tile	Block	Tile	Block	Tile	Block	Tile	Block	Tile	Block	Tile	Block	Tile	Block	Tile	Block
LUT	3023	189	2449	153	835	52	774	48	895	56	1068	67	1017	64	1064	67	836	52	774	48
FF	1024	64	768	48	1799	112	1799	112	1031	64	1031	64	1543	96	1527	95	1543	96	1543	96
Slice	1056	66	556	35	522	33	243	15	395	25	223	14	451	28	243	15	472	30	295	18
Max-Freq	240 MHz		445 MHz		540 MHz		737 MHz		245 MHz		487 MHz		360 MHz		600 MHz		370 MHz		620 MHz	

Scalability

- Virtex-7 and US+ representatives
  - BRAM count: largest/smallest
  - LUT-to-BRAM ratio: highest/lowest
- 100% BRAM on all devices
- Minimal logic utilization
  - <40% on the smallest/lowest
  - <5% on the largest/highest
- Reached maximum PE count
  - Min: 23K
  - Max: 86K
- Met the second goal

Scales linearly with BRAM capacity

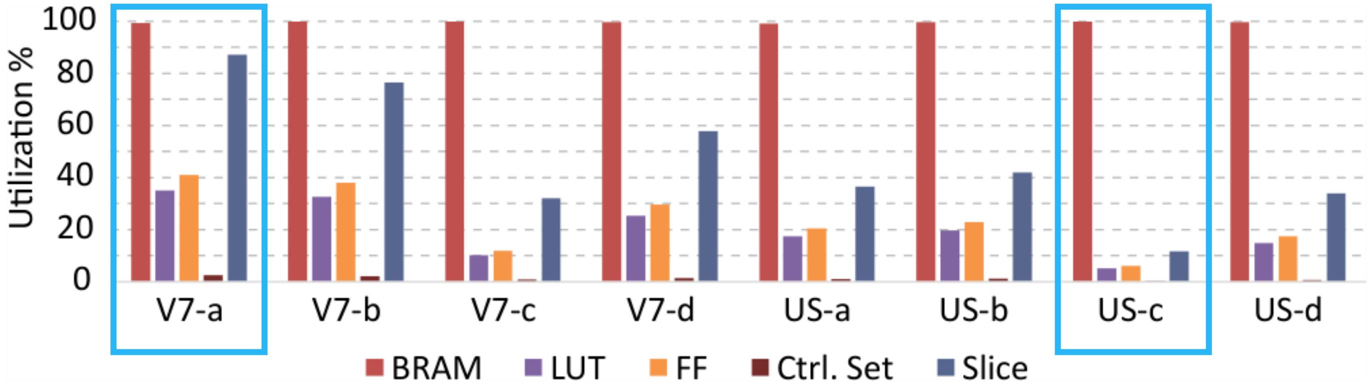


REPRESENTATIVE OF VIRTEX-7 AND ULTRASCALE+ DEVICES

Device	Tech	BRAM#	Ratio[1]	Max PE#[2]	ID
xc7vx330tffg-2	V7	750	272	24K	V7-a
xc7vx485tffg-2	V7	1030	295	32K	V7-b
xc7v2000tfhg-2	V7	1292	946	41K	V7-c
xc7vx1140tflg-2	V7	1880	379	60K	V7-d
xcvu3p-ffvc-3	US+	720	547	23K	US-a
xcvu23p-vsva-3	US+	2112	488	67K	US-b
xcvu19p-fsvb-2	US+	2160	1892	69K	US-c
xcvu29p-figd-3	US+	2688	643	86K	US-d

[1] LUT-to-BRAM ratio

[2] Maximum number of PEs if all BRAMs are utilized

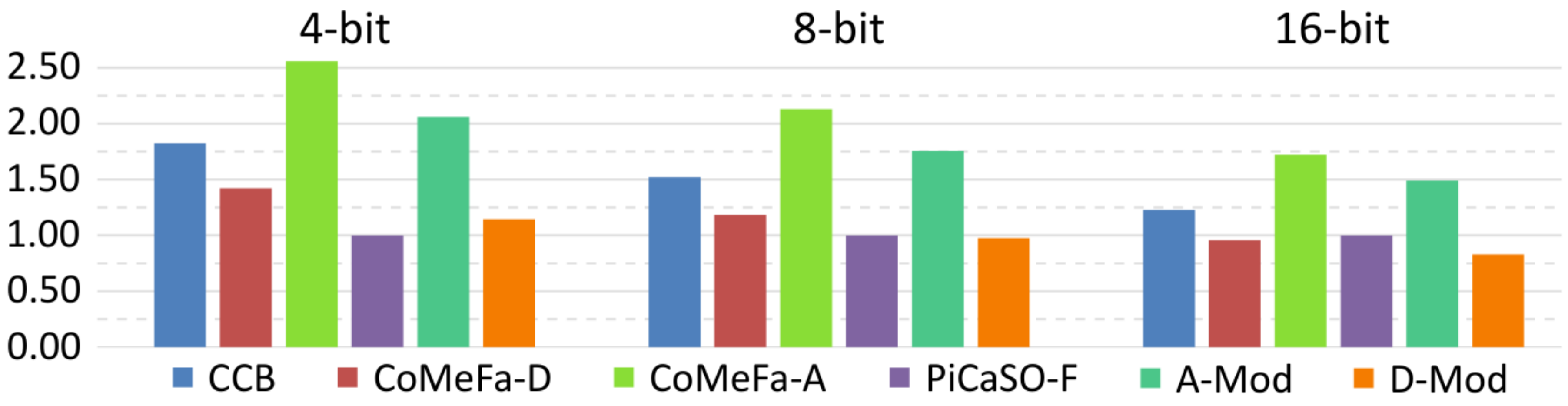


Smallest

Highest ratio

Latency Comparison with Custom Designs

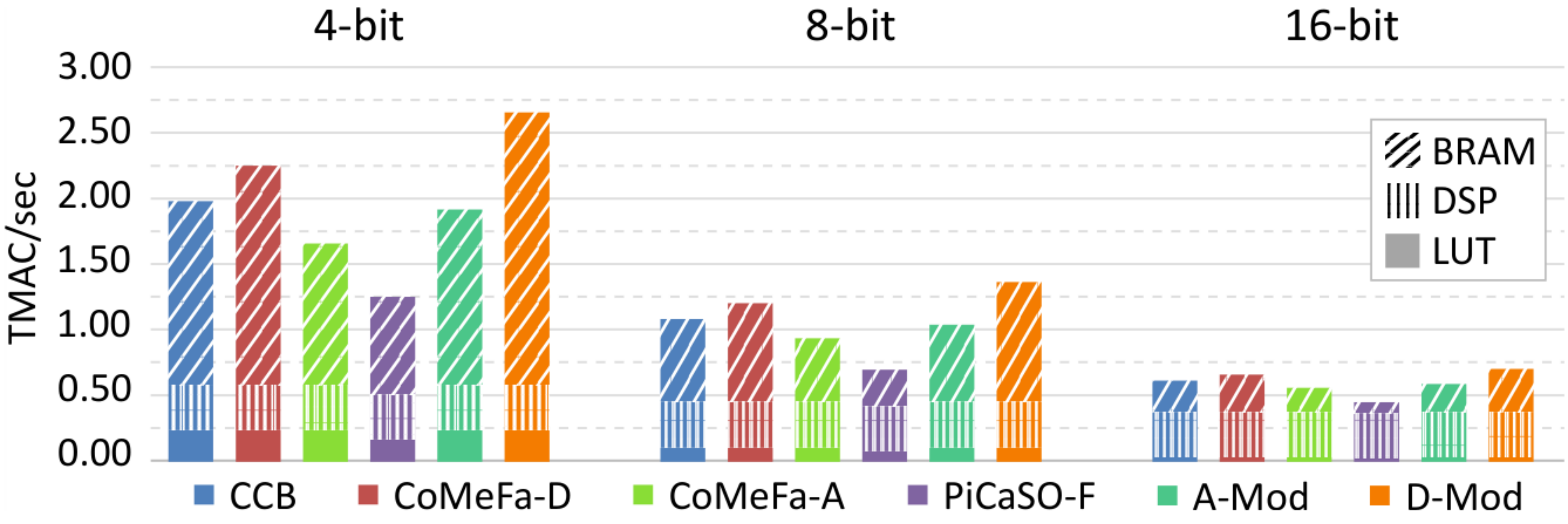
- A-Mod and D-Mod are modified CoMeFa-A and D
- PiCaSO is the fastest, except 16-bit CoMeFa-A
- Custom designs have single cycle read-modify-write
- Overlays need at least 2 cycles to read-modify then write
- PiCaSO adoption improves latency by 13.4% - 19.5%



Relative MAC latency w.r.t PiCaSO

Throughput Comparison

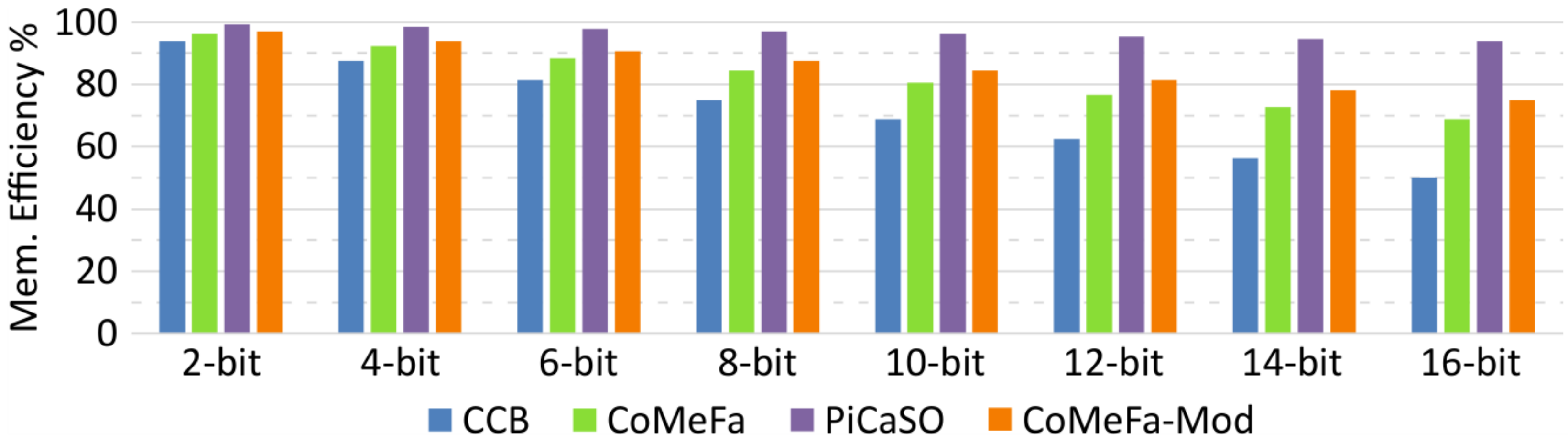
- Computed for 16 Mult then accumulation of products
- CCB/CoMeFa has 4x more PEs (1 PE per bitline)
- PiCaSO achieves 75% - 80% of peak throughput of CoMeFa-A (most practical)
- PiCaSO adoption improves throughput by 5% - 18%



Peak MAC throughput on Alveo U55

Memory Efficiency Comparison

- Memory utilization efficiency: Memory available for weights / Total Memory
- All require scratchpad memory:
  - CCB: 8N, CoMeFa: 5N, PiCaSO: 4N
- At 16-bit precision: CCB: 50%, CoMeFa: 68.8%, PiCaSO: 93.8%
- PiCaSO adoption improves memory utilization efficiency by 6.2%
  - 1.6 million more weights at 4-bit precision on a 100 Mb devices
- Widest mode is not good for memory efficiency

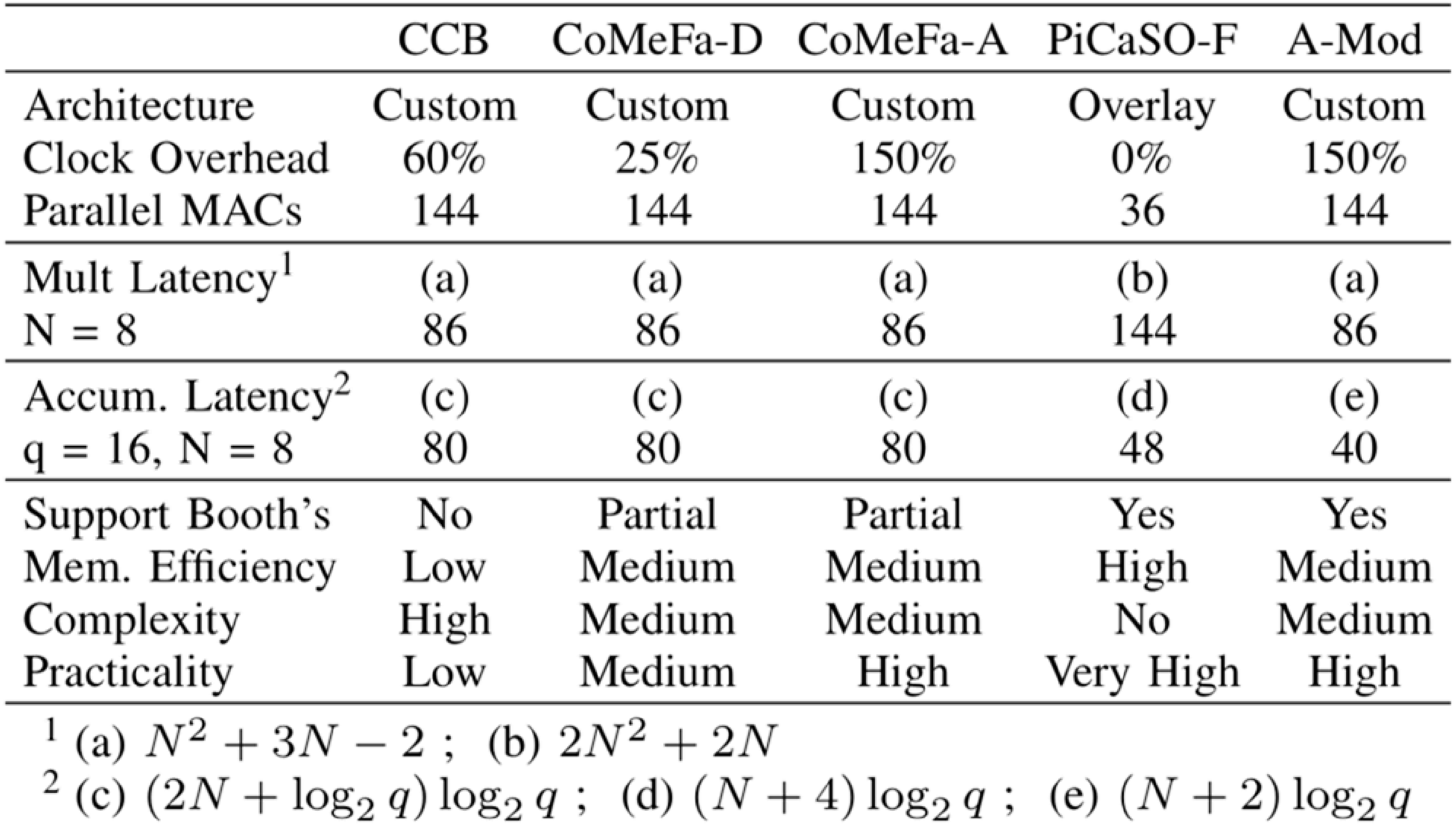


BRAM memory utilization efficiency on Virtex devices

Comparison Summary

PiCaSO

- is an overlay, readily available
- 0% clock overhead
- less parallel MAC units
- 2x slower mult
- 2x faster accumulation
- supports Booth's
- has high memory efficiency
- 0 design complexity
- Practicality: **Very High**



	CCB	CoMeFa-D	CoMeFa-A	PiCaSO-F	A-Mod
Architecture	Custom	Custom	Custom	Overlay	Custom
Clock Overhead	60%	25%	150%	0%	150%
Parallel MACs	144	144	144	36	144
Mult Latency[1]	(a)	(a)	(a)	(b)	(a)
N = 8	86	86	86	144	86
Accum. Latency[2]	(c)	(c)	(c)	(d)	(e)
q = 16, N = 8	80	80	80	48	40
Support Booth's	No	Partial	Partial	Yes	Yes
Mem. Efficiency	Low	Medium	Medium	High	Medium
Complexity	High	Medium	Medium	No	Medium
Practicality	Low	Medium	High	Very High	High

[1] (a) $N^2 + 3N - 2$; (b) $2N^2 + 2N$

[2] (c) $(2N + \log_2 q)\log_2 q$; (d) $(N + 4)\log_2 q$; (e) $(N + 2)\log_2 q$

Conclusions

- PIM Overlay fabric is readily available on off-the-shelf devices
- PiCaSO achieved memory-centric design goals
  - PIM block as fast as BRAM max speed and as many as the BRAM
- Provides 2x speed and 2x utilization improvement over benchmark design
- Highly scalable on Virtex-7 and US+ devices
- PiCaSO features can improve CCB and CoMeFa performance
- Overlays can be good enough/better for low latency and high mem. efficiency

Questions?