Making BRAMs Compute
Creating Scalable Computational Memory Fabric Overlays

MD Arafat Kabir, Joshua Hollis, Atiyehsadat Panahi, Jason Bakos, Miaoqing Huang, and David Andrews

UNIVERSITY OF ARKANSAS

Introduction

- PIM Architectures break memory bottleneck in ML Applications.
- FPGAs are ideal substrates for creating custom PIM accelerators.
- Proposed work presents a very efficient PIM overlay architecture.

PIM Architecture

- BRAM is used as the registerfile.
- A full adder/subtractor (FA/S) is implemented using a single LUT6.
- Op-Encoder provides a high-level interface to the FA/S module.
- Operand-Mux provides a zero-copy path for faster reduction.
- A binary-hopping data movement network between PIM blocks.
- Network module and OpMux overlap data movement with computation.

Analysis

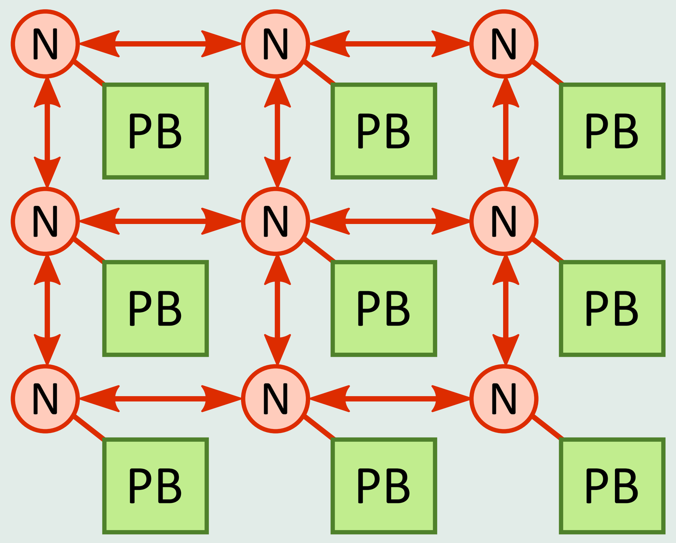
- A state-of-the-art PIM overlay is used as the benchmark.
- 2x smaller than the benchmark
- 2x faster than the benchmark
- Runs at BRAM max frequency
- Scales linearly with BRAM capacity with >95% BRAM utilization

Download This Poster

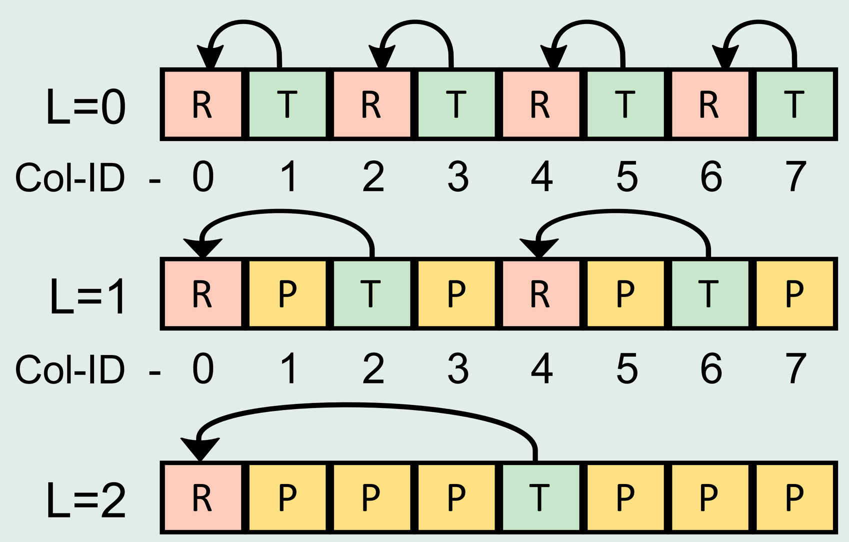
The **fastest** and **most scalable** architecture for **Processing-in-Memory** overlay fabric in FPGA!

Compared to state-of-the-art

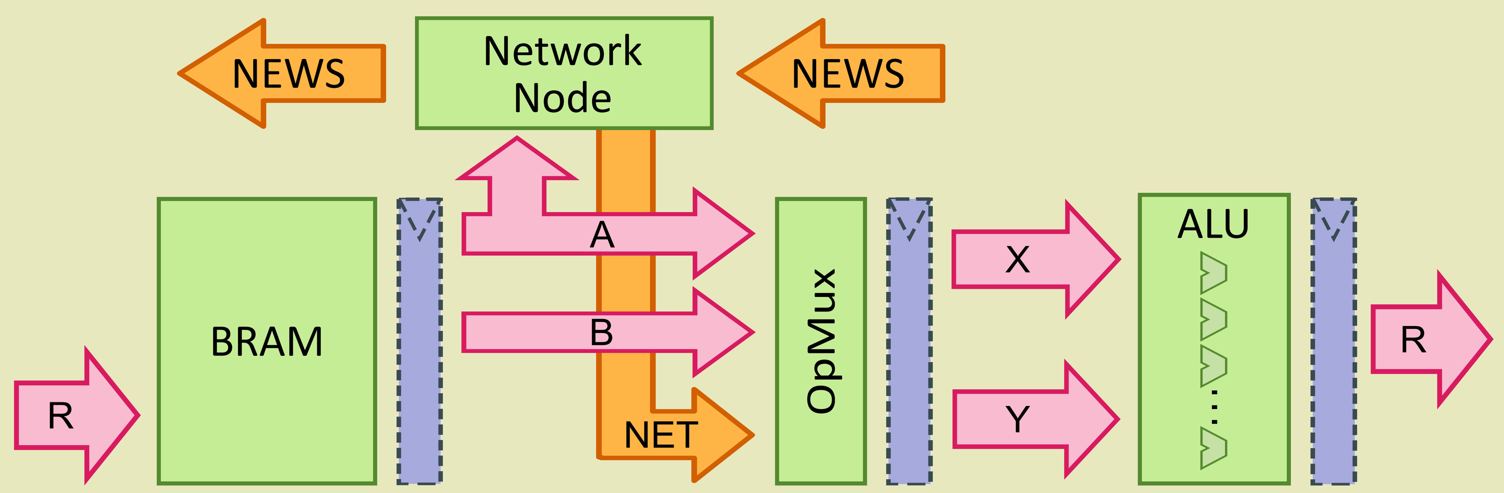
- System Freq : **2x faster**
- Accumulation : **17x faster**
- Utilization : **2x smaller**



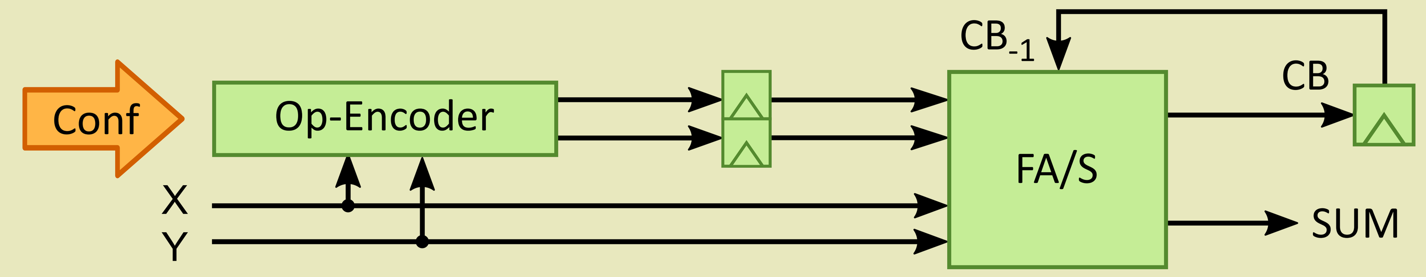
Network Architecture



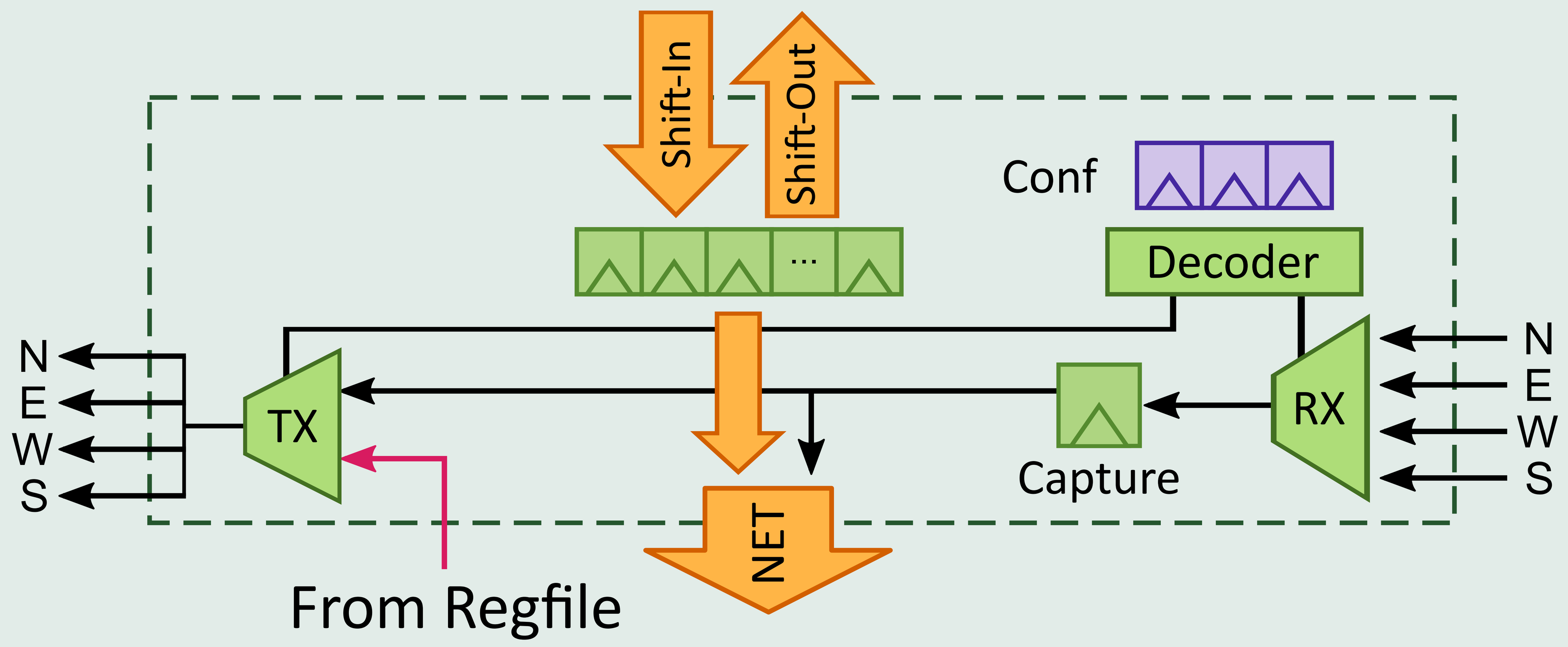
Jump over PE-Blocks



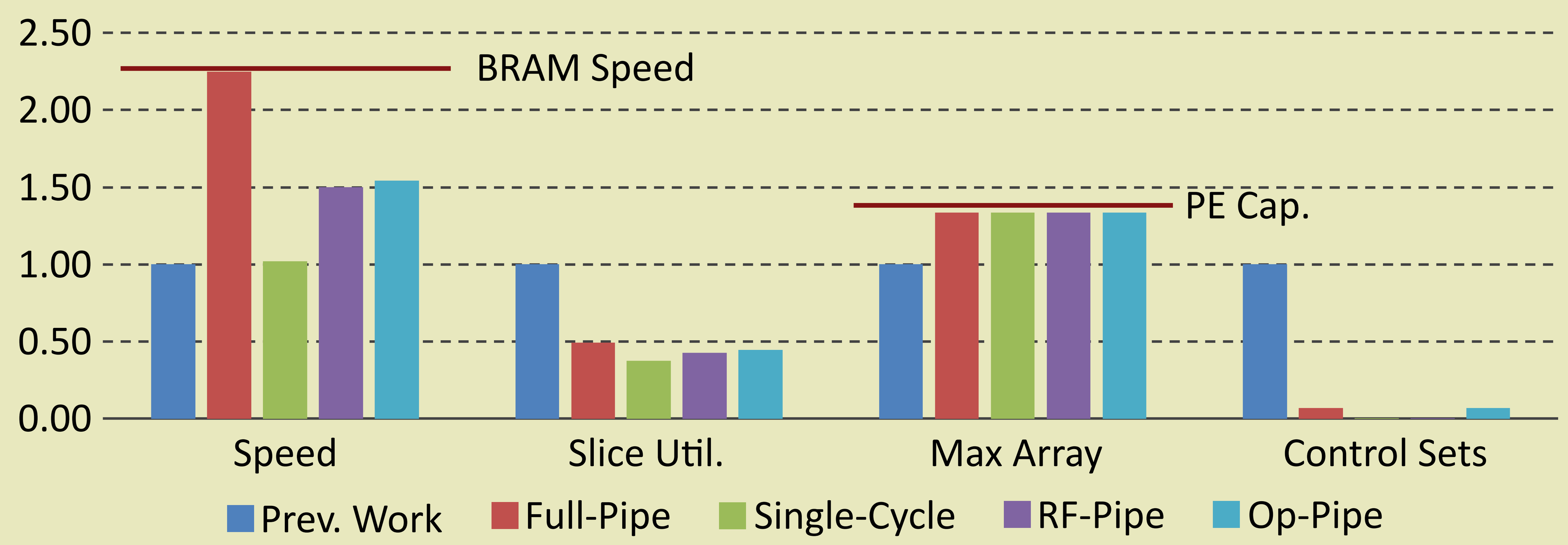
Processing-in-Memory architecture



ALU architecture



Network node (N) architecture for hopping



Relative improvements in the proposed Architecture

Tile Performance

	Previous Work [11]		Full-Pipe	
	Tile	Per-Block	Tile	Per-Block
LUT	3023	189	835	52
FF	1024	64	1799	112
Slice	1056	66	522	33
Max-Freq	240 MHz		540 MHz	

Scalability Study

	Prev. Work [11]	Full-Pipe	Single-Cycle	RF-Pipe	Op-Pipe
Max-Size	24K	32K	32K	32K	32K
LUT	74.6%	32.0%	35.8%	36.0%	31.7%
FF	16.0%	32.2%	16.3%	16.3%	26.8%
BRAM	73.8%	97.2%	97.2%	97.2%	97.2%
Uniq. Ctrl. Set	32.1%	2.2%	<0.01%	<0.01%	2.2%
Slice	86.0%	80.2%	58.0%	58.6%	64.7%

Accumulation Latency

Operation	Previous Work [11]	Full-Pipe
ADD/SUB	$2N$	$2N$
MULT[1]	$2N^2 + 2N$	$2N^2 + 2N$
Reduction[2]	$(q - 1 + 2\log_2 q)N$	$15 + \frac{q}{16} + 4N + (N+4)J$
q = 128, N = 32	4512	259

[1] Booth's Radix-2 multiplication

[2] q : Number of columns to be accumulated
N : Operand width
J : Number of network jumps needed = $\log_2(q/16)$

OpMux Configuration

Config Code	X	Y	Description
A-OP-B	A	B	Used in standard operations
A-FOLD-1	A	{0, A[H2]}	A[H2]: second half of A
A-FOLD-2	A	{0, A[Q2]}	A[Q2]: second quarter of A
A-FOLD-3	A	{0, A[HQ2]}	A[HQ2]: second half-quarter of A
A-FOLD-4	A	{0, A[HHQ2]}	A[HHQ2]: second half of A[HQ1][1]
A-OP-NET	A	NET	Operates on network stream
0-OP-B	0	B	Used in the first iteration of MULT

[1] A[HQ1] : first half-quarter of A

Block Performance

	Prev. Work [11]	Full-Pipe	Single-Cycle	RF-Pipe	Op-Pipe
LUT	187	53	82	74	53
FF	64	119	71	103	103
Slice	64	32	42	36	37
Max-Freq	270 MHz	540 MHz	265 MHz	400 MHz	390 MHz