MASR: A Modular Accelerator for Sparse RNNs

Udit Gupta, Brandon Reagen,
Lillian Pentecost, Marco Donato, Thierry Tambe
Alexander M. Rush, Gu-Yeon Wei, David Brooks

Harvard University

In this talk

Parallelism

Sparsity

In this talk

Parallelism

Sparsity

Custom sparse encoding

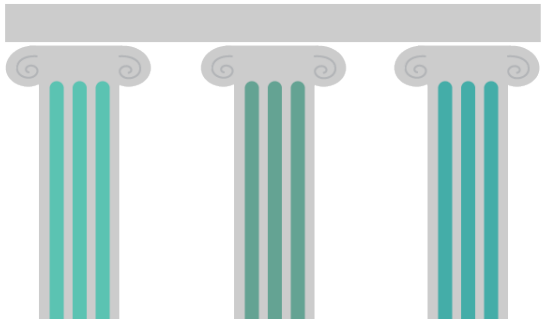
In this talk

Parallelism

Sparsity

Custom sparse encoding

Sparsity + Parallelism



Area Energy Perf.

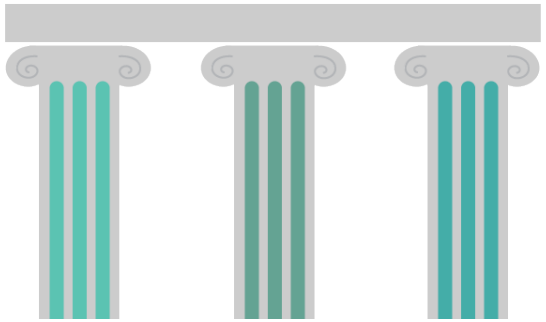
In this talk

Parallelism

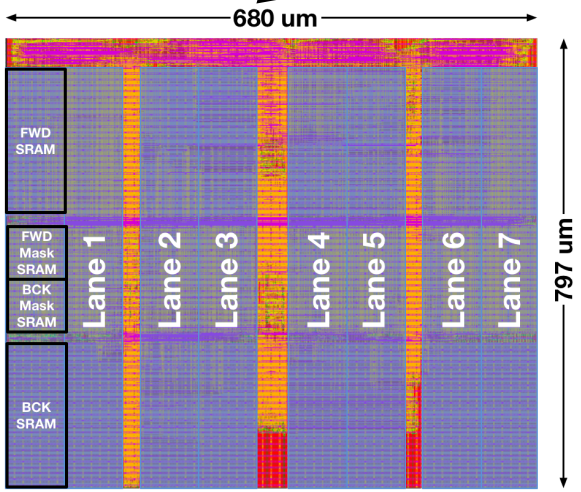
Sparsity

Custom sparse encoding

Sparsity + Parallelism

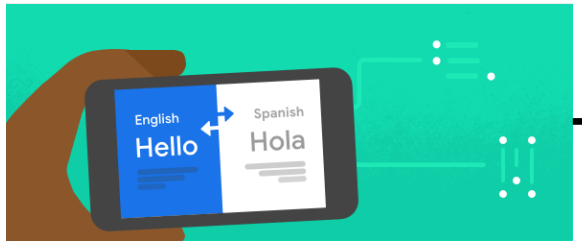


Area Energy Perf.



RNNs can revolutionize interactions with tech

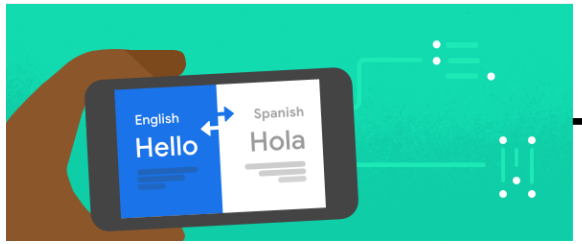
Applications → Model



Recurrent Neural Networks

Must deploy RNNs onto resource constrained HW

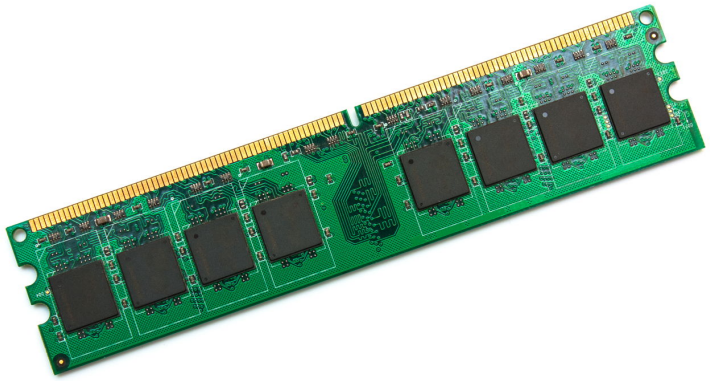
Applications → Model → Hardware Platforms



Recurrent Neural Networks

RNNs levy high inference cost

Large memory footprint



Tens of MBs for ASR

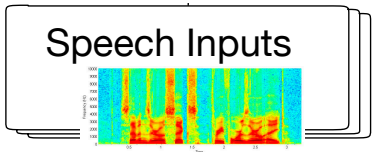
High compute footprint

Tens of GFLOPs for ASR

High energy footprint

Billions of memory accesses for ASR

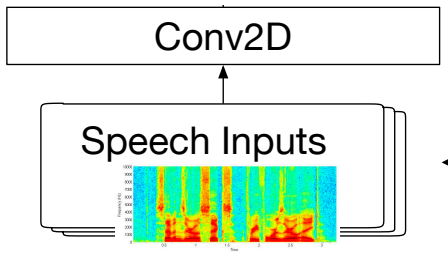
Inference cost of ASR RNNs



DeepSpeech 2 Topology

Inputs are represent as a sequence of waveforms

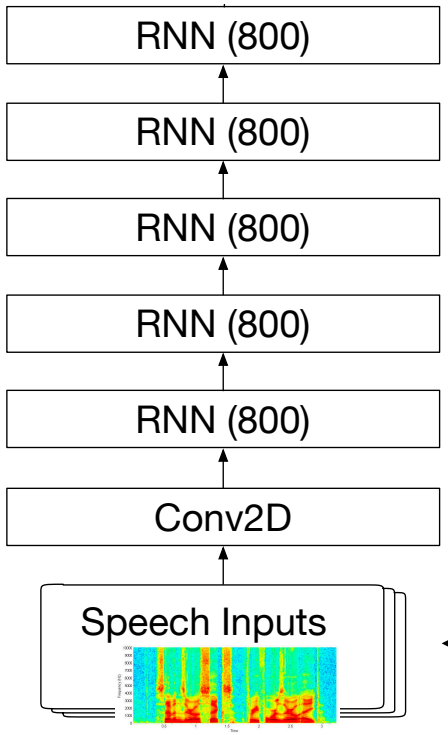
Inference cost of ASR RNNs



DeepSpeech 2 Topology

Inputs are represent as a sequence of waveforms

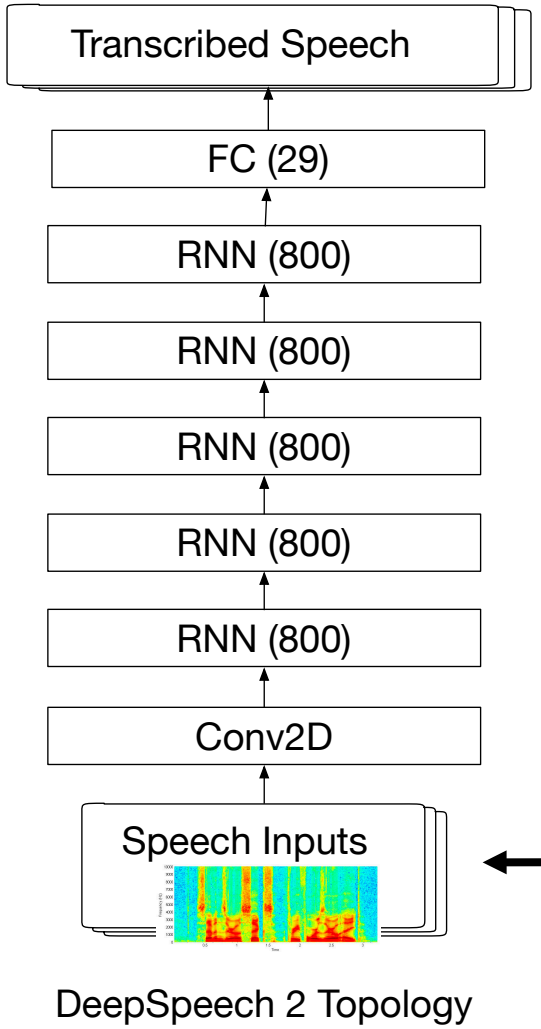
Inference cost of ASR RNNs



DeepSpeech 2 Topology

Inputs are represent as a sequence of waveforms

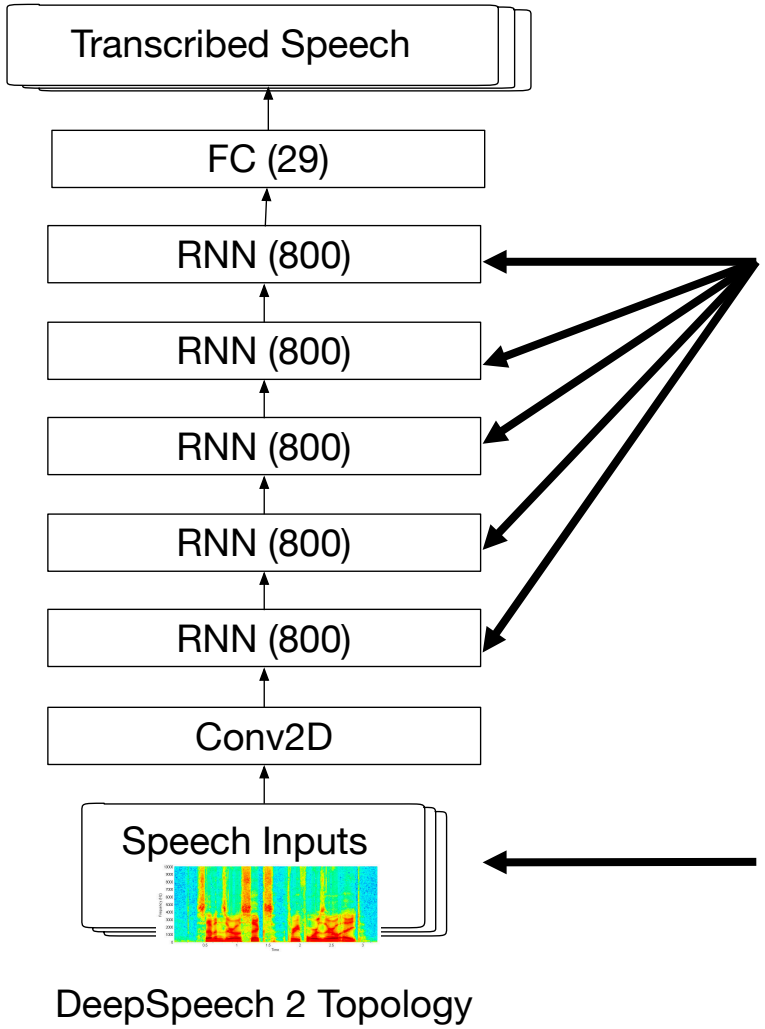
Inference cost of ASR RNNs



DeepSpeech 2 Topology

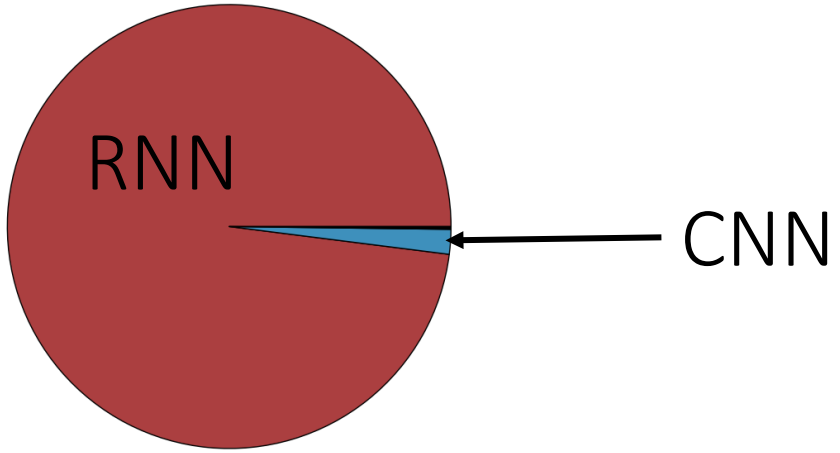
Inputs are represent as a sequence of waveforms

Inference cost of ASR RNNs



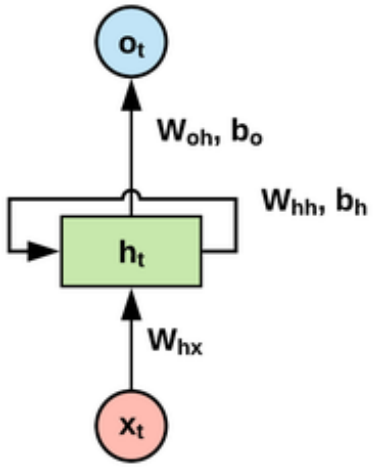
DeepSpeech 2 Topology

Over 98% of parameters found in recurrent layers
(over 30 million, 50MB)

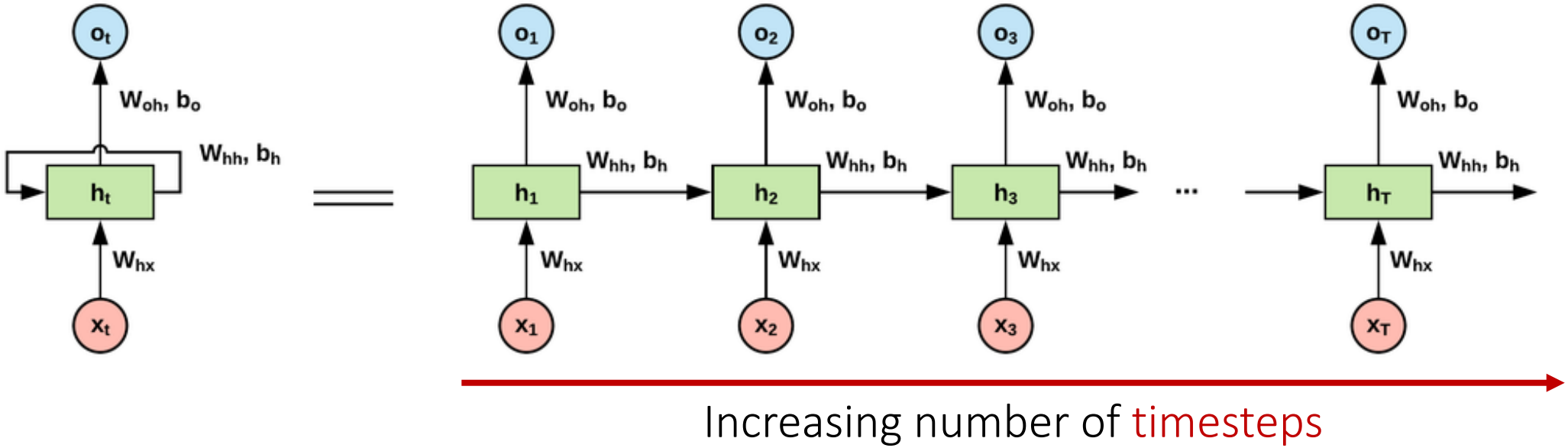


Inputs are represent as a sequence of waveforms

RNNs cost scales with input length (timesteps)



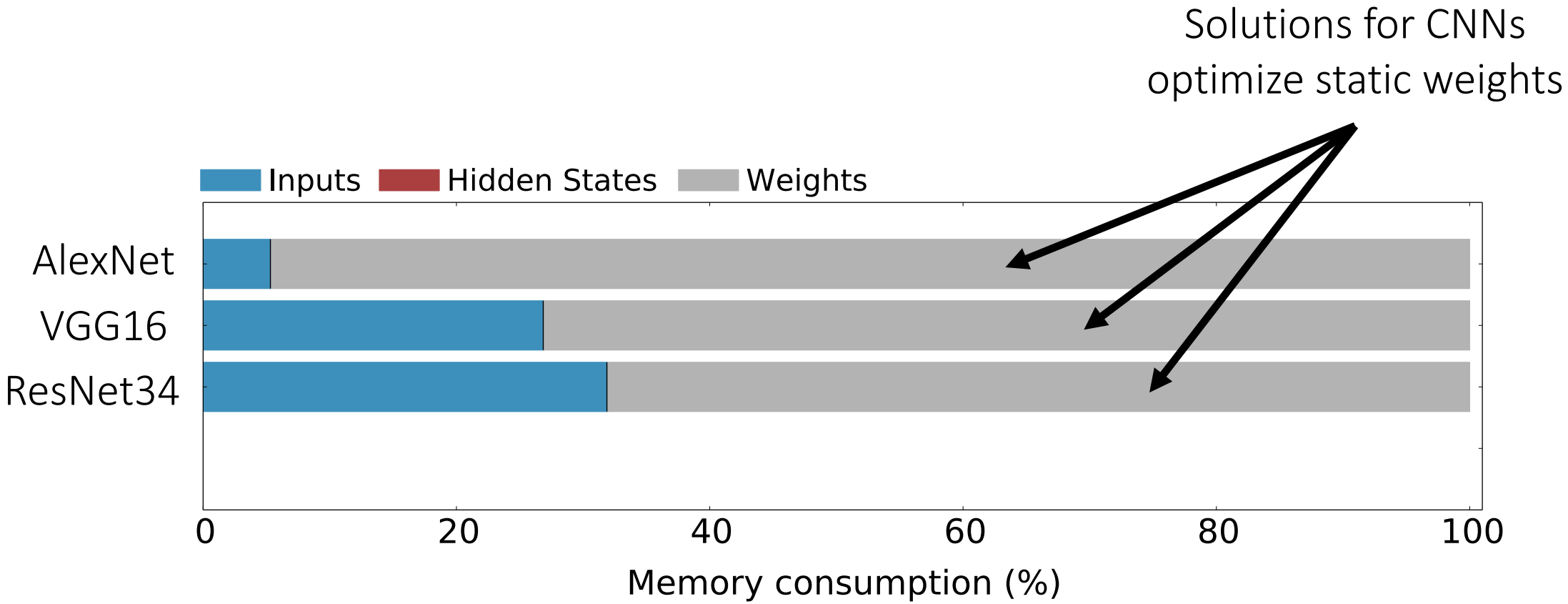
RNNs cost scales with input length (timesteps)



With increasing number of timesteps:

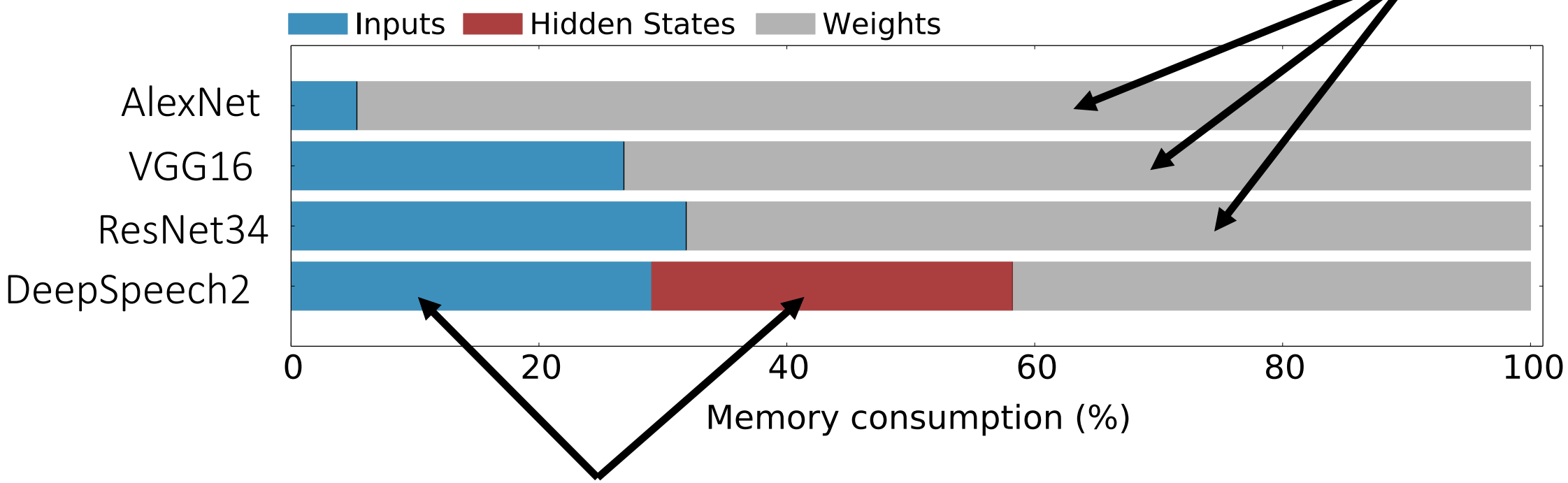
- Number of matrix-vector operations increases (FLOPs)
- Activation storage increases (area)

RNNs pose unique challenges



RNNs pose unique challenges

Solutions for CNNs
optimize static weights



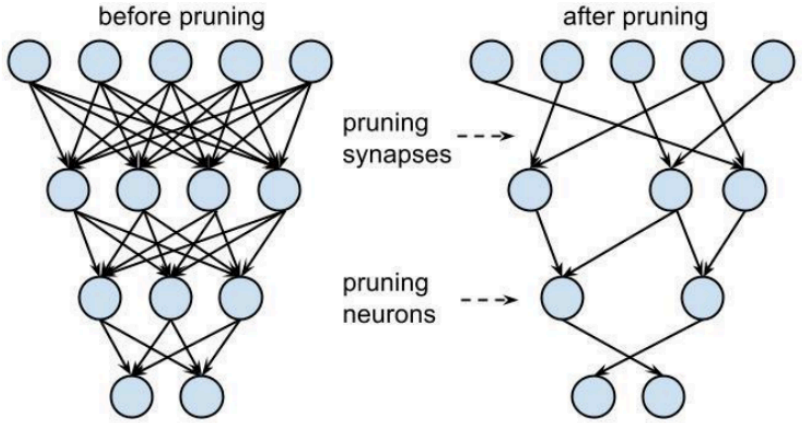
RNN memory consumption
dominated by dynamic activation
requiring unique solutions

Limitations of current sparse DNN accelerators

EIE: Efficient Inference Engine on Compressed DNNs

Song Han, et. al.

ISCA, 2016, citation count: 909



Reduces weight footprint by 3x

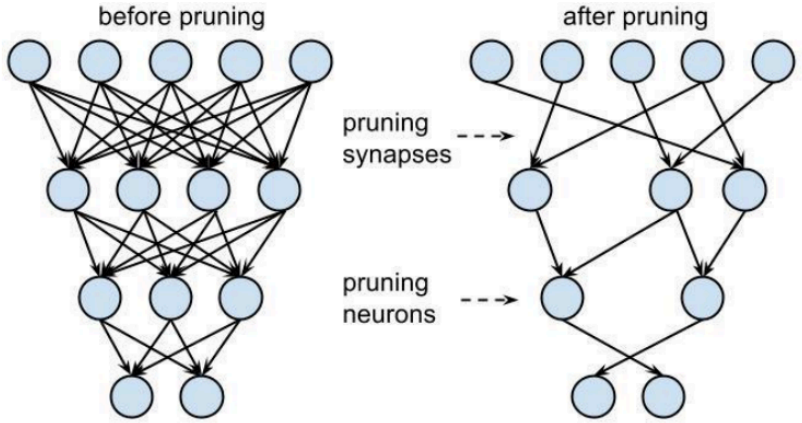
Does not compress activations (up to 3x savings)

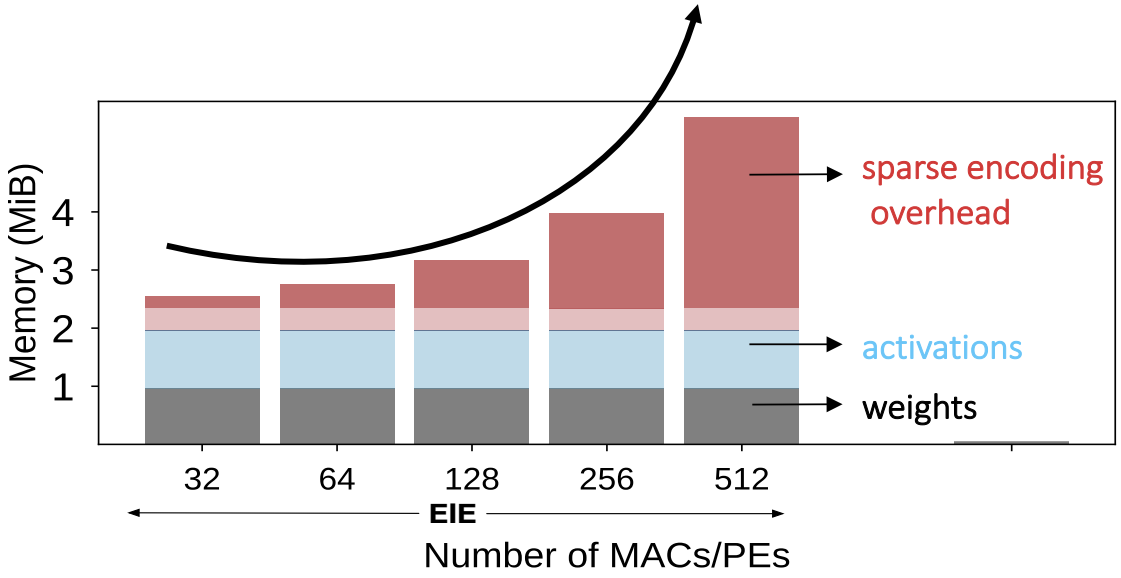
Limitations of current sparse DNN accelerators

EIE: Efficient Inference Engine on Compressed DNNs

Song Han, et. al.

ISCA, 2016, citation count: 909





Reduces weight footprint by 3x

Does not compress activations (up to 3x savings)

Does not scale (over 2x savings at high parallelism)

Proposed solution: MASR

Problem

Large memory footprint – static weights and dynamic activations

Solution

Logic centric sparse encoding

Proposed solution: MASR

Problem	Solution
 Large memory footprint – static weights and dynamic activations	Logic centric sparse encoding
 Irregular, sparse computation makes parallelism hard	Scalable sparse encoding architecture Accelerator to exploit parallelism

Proposed solution: MASR

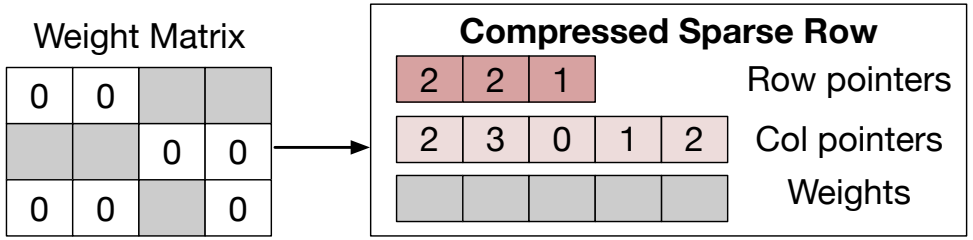
Problem	Solution
 Large memory footprint – static weights and dynamic activations	Logic centric sparse encoding
 Irregular, sparse computation makes parallelism hard	Scalable sparse encoding architecture Accelerator to exploit parallelism
 High performance with parallelism and irregularity is hard	Work stealing for load balancing

Proposed solution: MASR

Problem	Solution
 Large memory footprint – static weights and dynamic activations	Logic centric sparse encoding
Irregular, sparse computation makes parallelism hard	Scalable sparse encoding architecture Accelerator to exploit parallelism
High performance with parallelism and irregularity is hard	Work stealing for load balancing

Encoding techniques to exploit sparsity

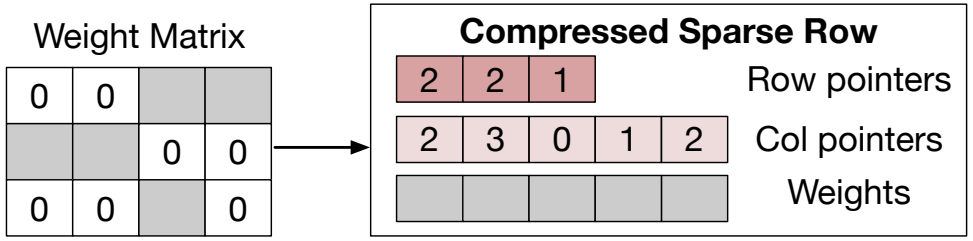
Memory centric



Current state-of-the-art
Song Han, et. al. ISCA, 2016

Encoding techniques to exploit sparsity

Memory centric



Pressures memory system (2 pointers/1 weight)

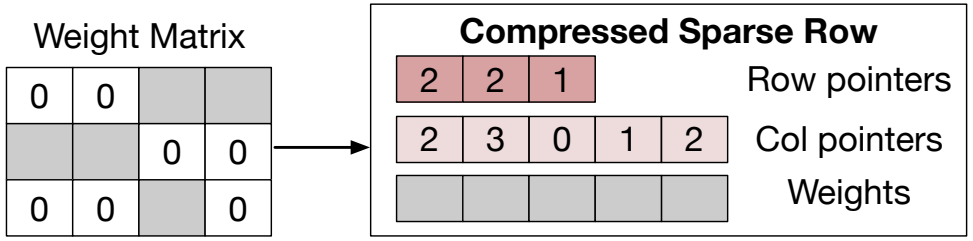
Static weight encoding computed offline

Activations generated at run-time; uncompressed

Current state-of-the-art
Song Han, et. al. ISCA, 2016

Encoding techniques to exploit sparsity

Memory centric



Pressures memory system (2 pointers/1 weight)

Static weight encoding computed offline

Activations generated at run-time; uncompressed

Current state-of-the-art
Song Han, et. al. ISCA, 2016

Logic centric

Sparse encoding binary mask

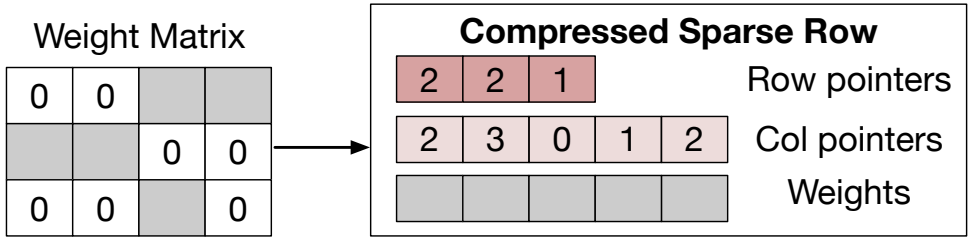
0	0	1	1
1	1	0	0
0	0	1	0

\+ Logic

Proposed solution

Encoding techniques to exploit sparsity

Memory centric



Pressures memory system (2 pointers/1 weight)

Static weight encoding computed offline

Activations generated at run-time; uncompressed

Current state-of-the-art
Song Han, et. al. ISCA, 2016

Logic centric

Sparse encoding binary mask

0	0	1	1
1	1	0	0
0	0	1	0

\+ Logic

Relieves memory pressure (single pointer)

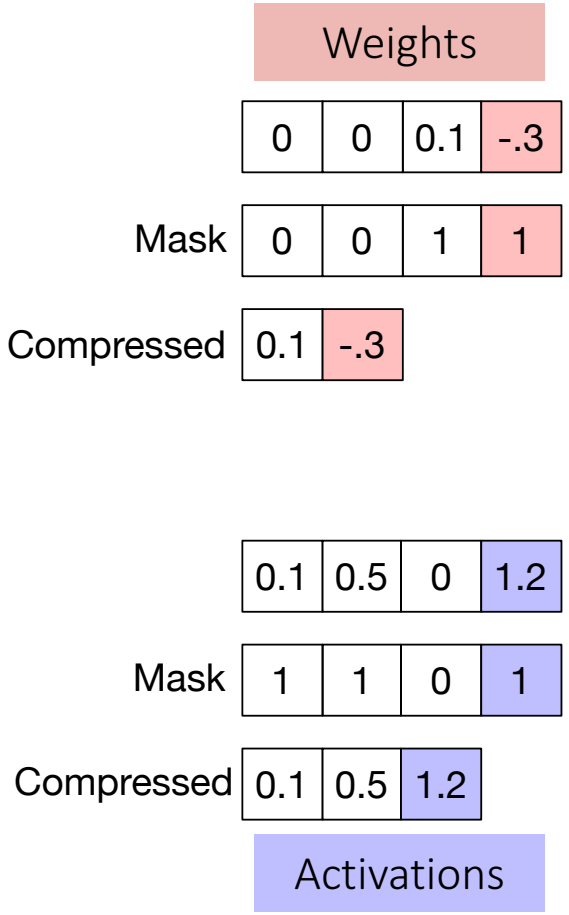
Compute sparse address at run-time!

Weights and activations are compressed

Proposed solution

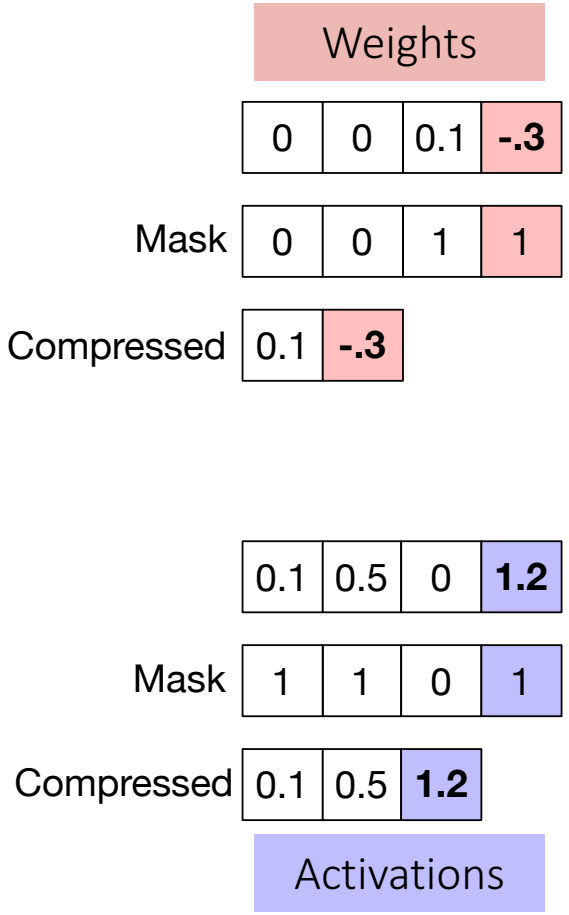
MASR's logic centric sparse encoding

Compute address of non-zero weight and activation stored in compressed format



MASR's logic centric sparse encoding

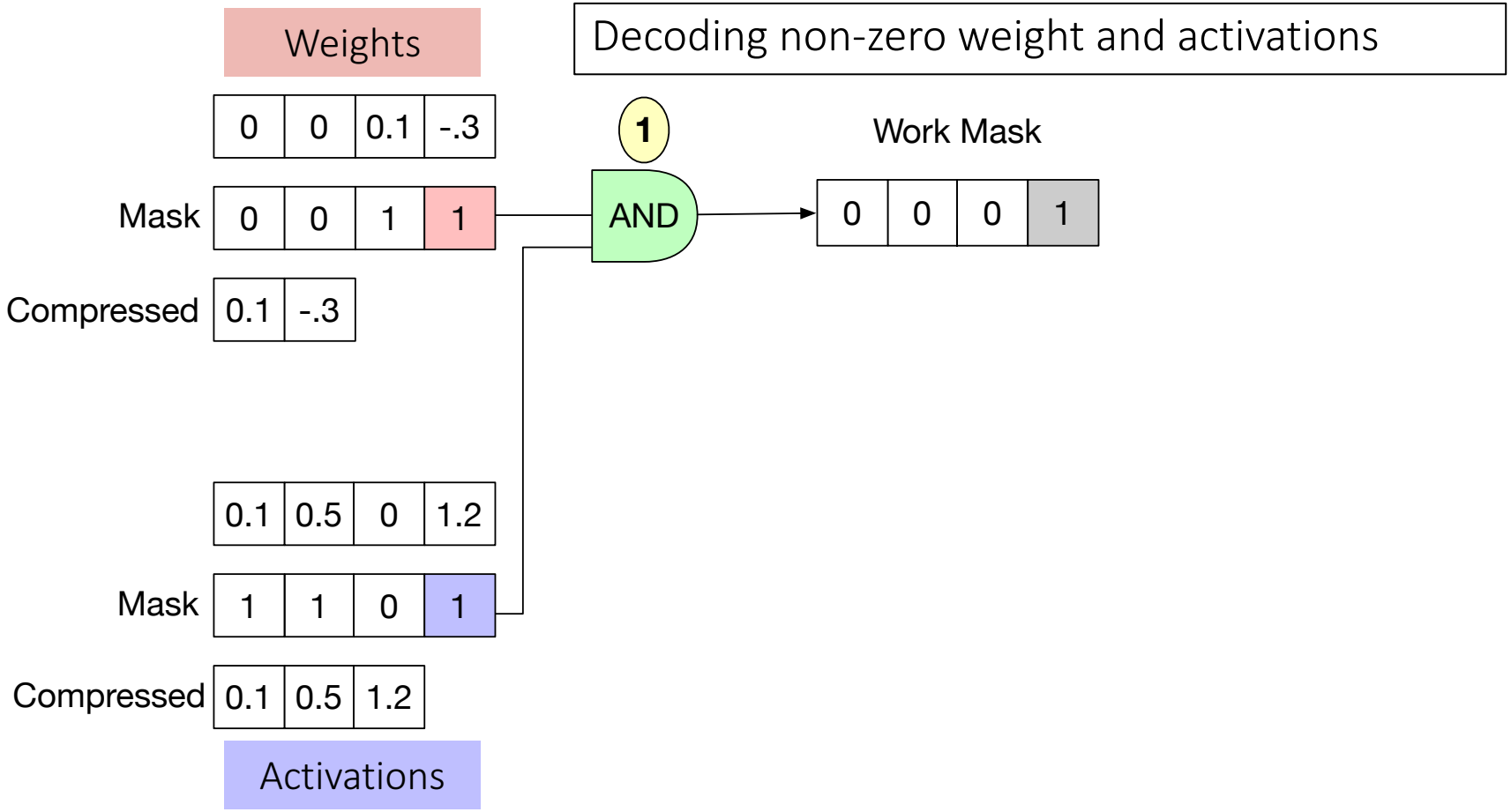
Compute address of non-zero weight and activation stored in compressed format



MASR's logic centric sparse encoding

Compute address of non-zero weight and activation stored in compressed format

1. Compute when weight and activation are both non-zero

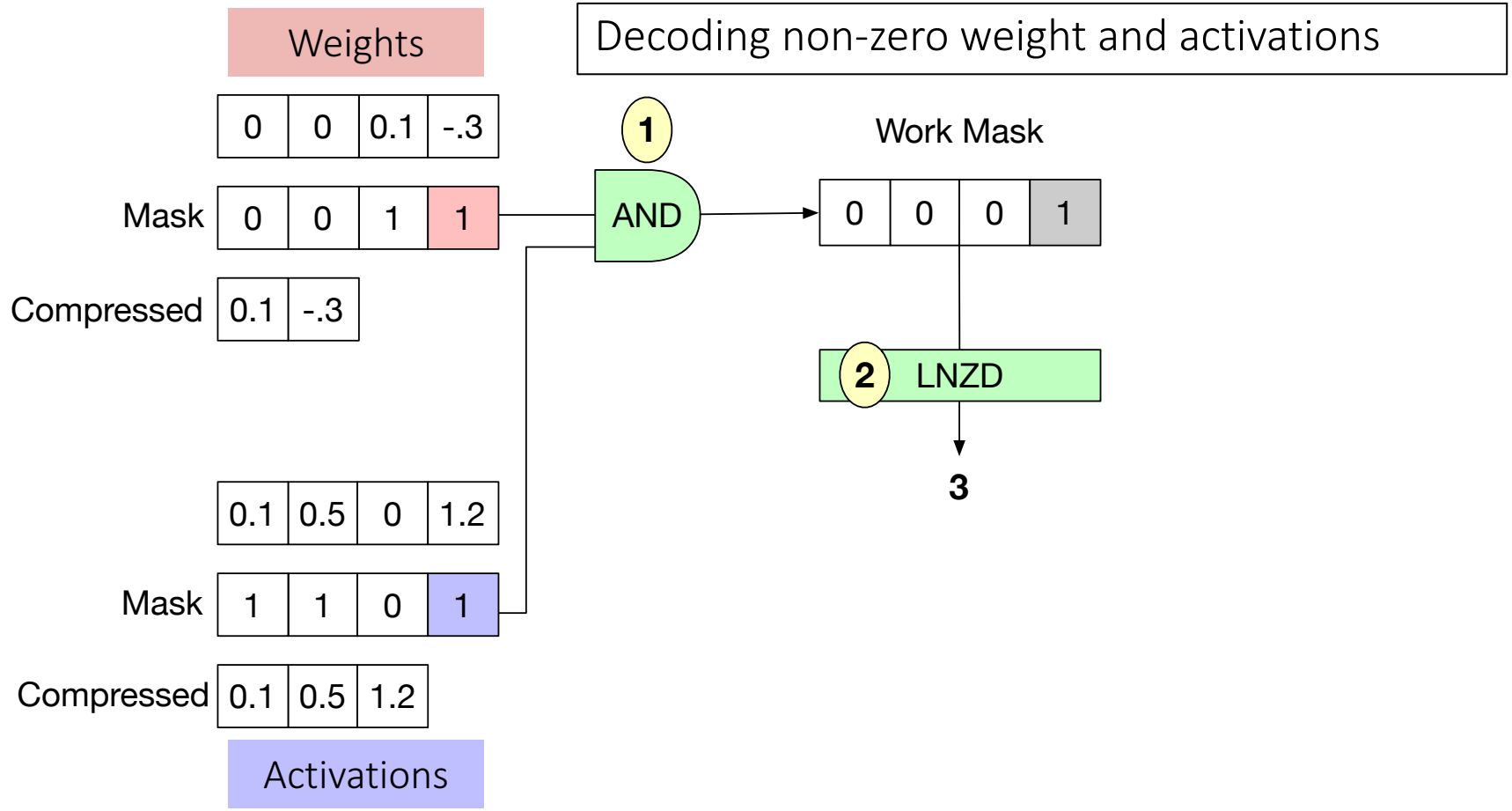


MASR's logic centric sparse encoding

Compute address of non-zero weight and activation stored in compressed format

1. Compute when weight and activation are both non-zero

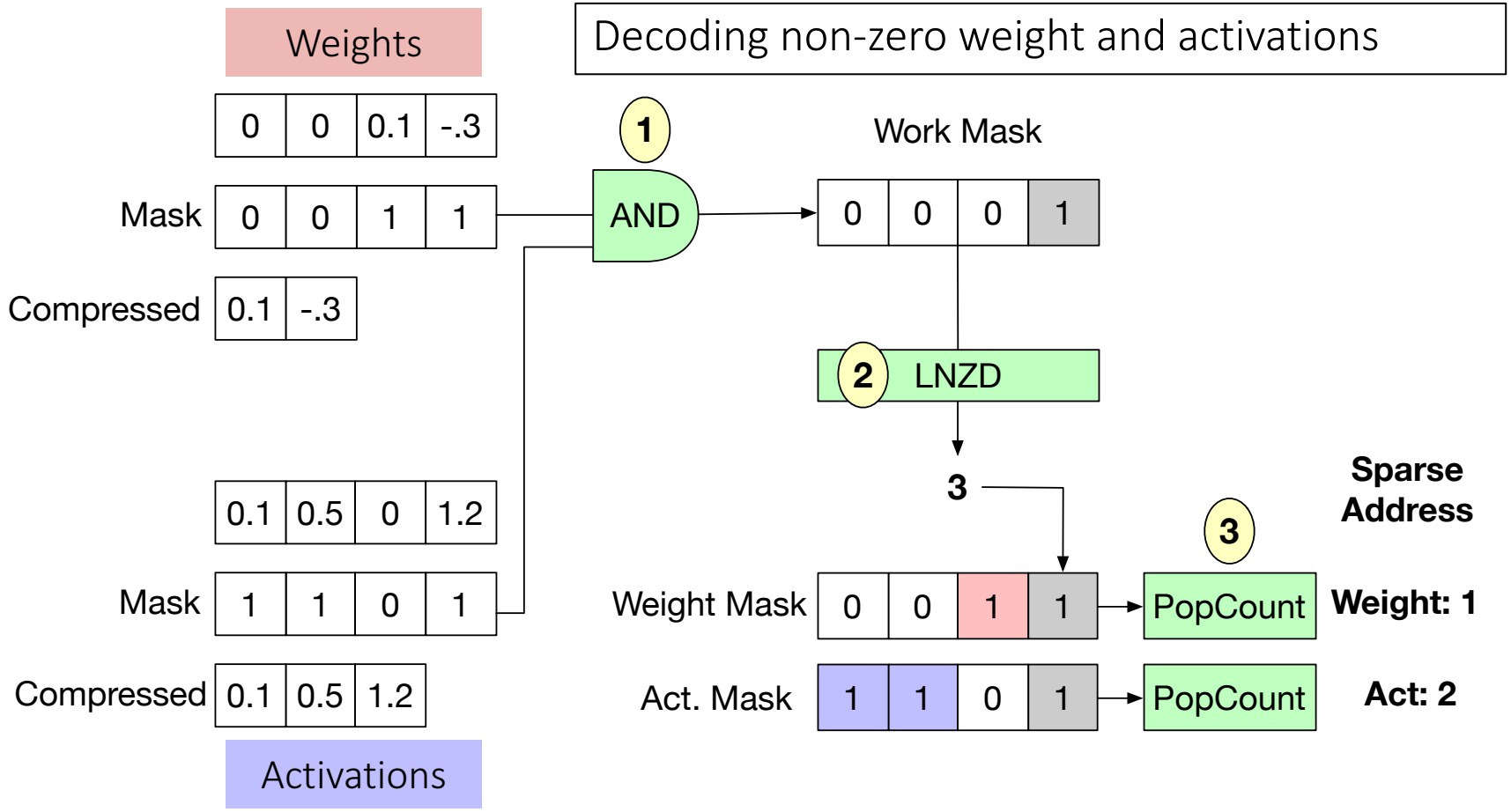
2. Find next non-zero pair (leading non-zero detect)



MASR's logic centric sparse encoding

Compute address of non-zero weight and activation stored in compressed format

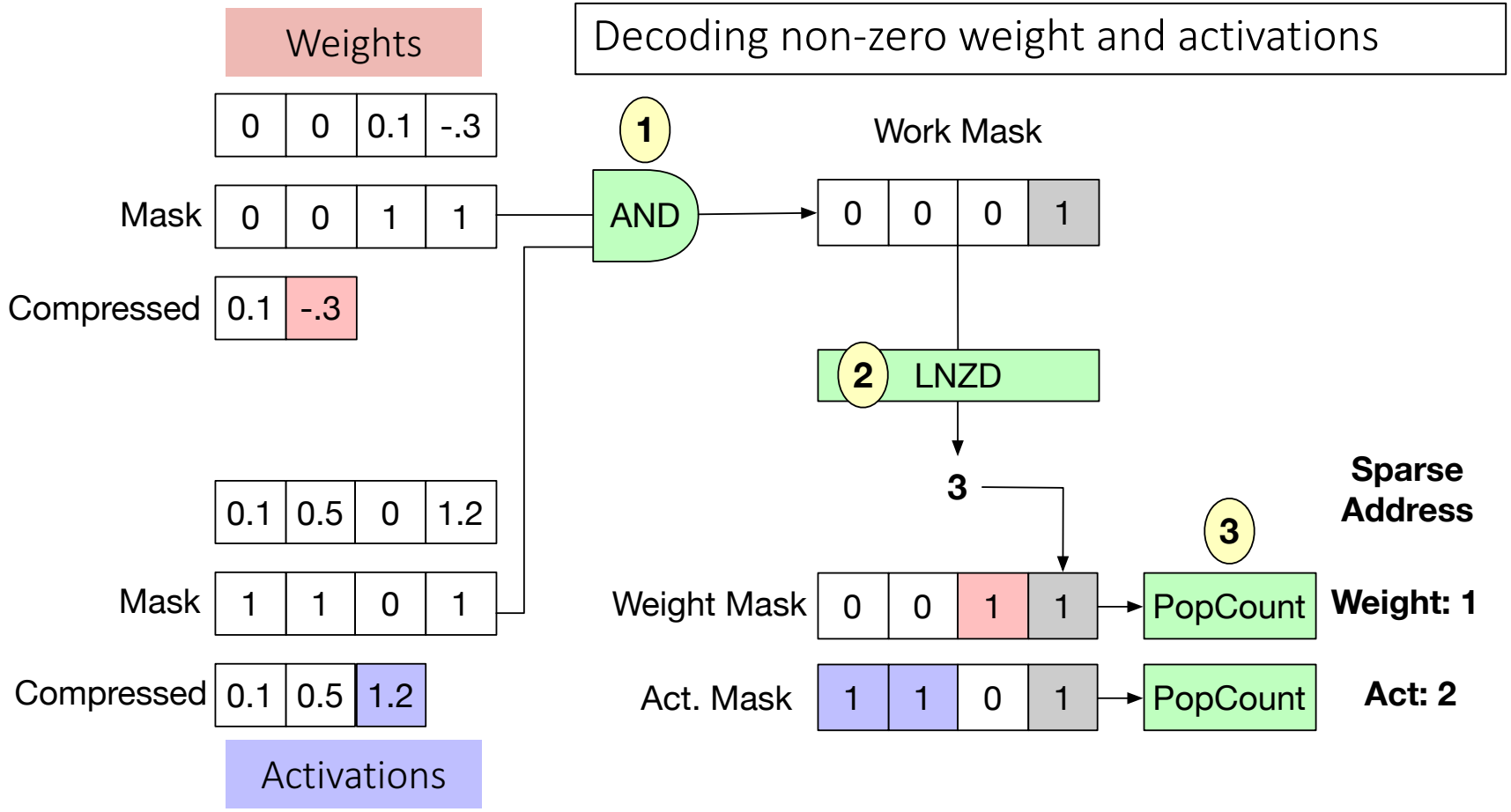
1. Compute when weight and activation are both non-zero
2. Find next non-zero pair (leading non-zero detect)
3. Evaluate address of non-zero weight and activation (population count)



MASR's logic centric sparse encoding

Compute address of non-zero weight and activation stored in compressed format

1. Compute when weight and activation are both non-zero
2. Find next non-zero pair (leading non-zero detect)
3. Evaluate address of non-zero weight and activation (population count)



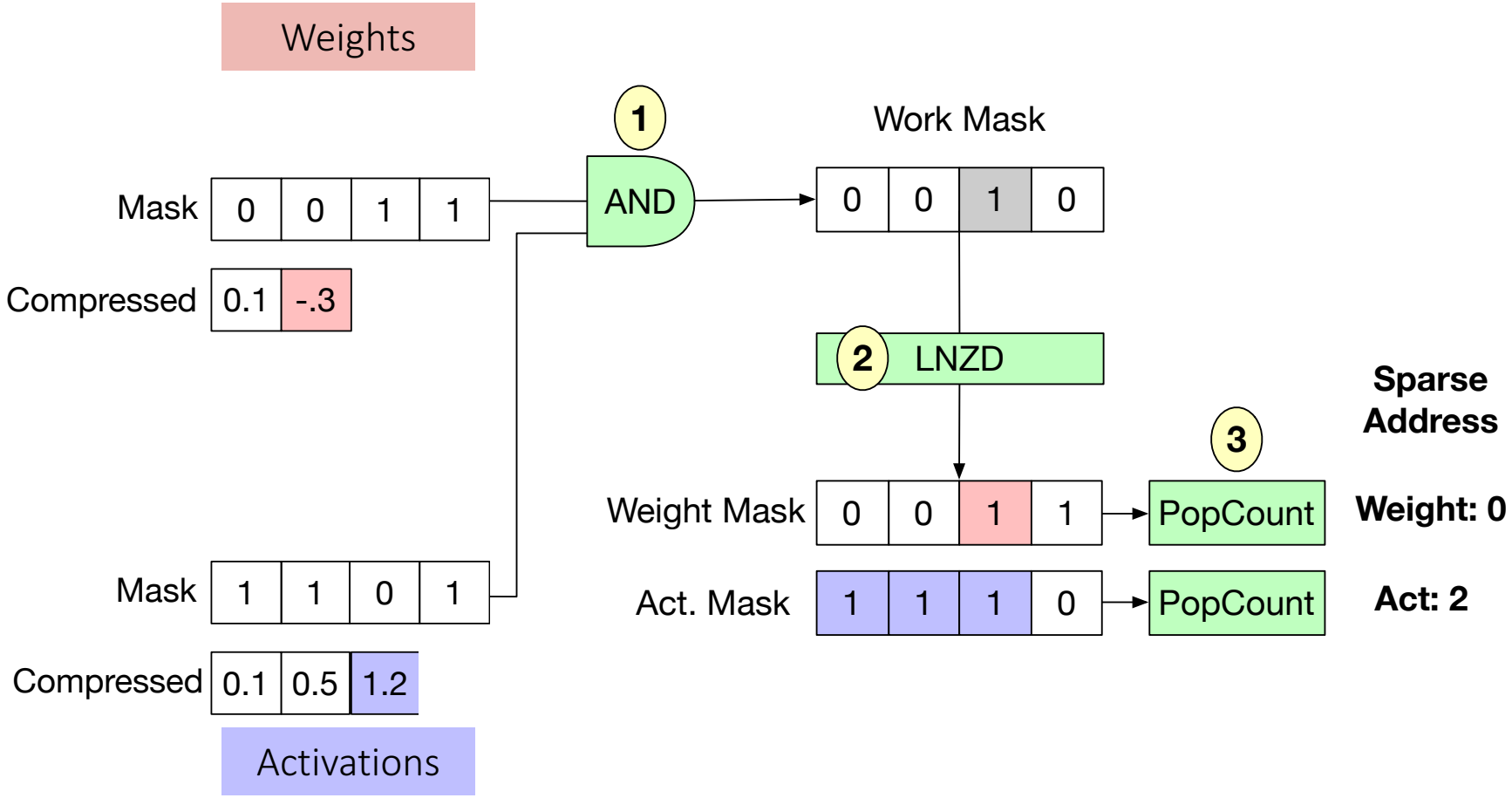
MASR's logic centric sparse encoding

Can compute address of sparse weight/activation stored compactly in memory!

Takeaways

3x memory savings from weight compression

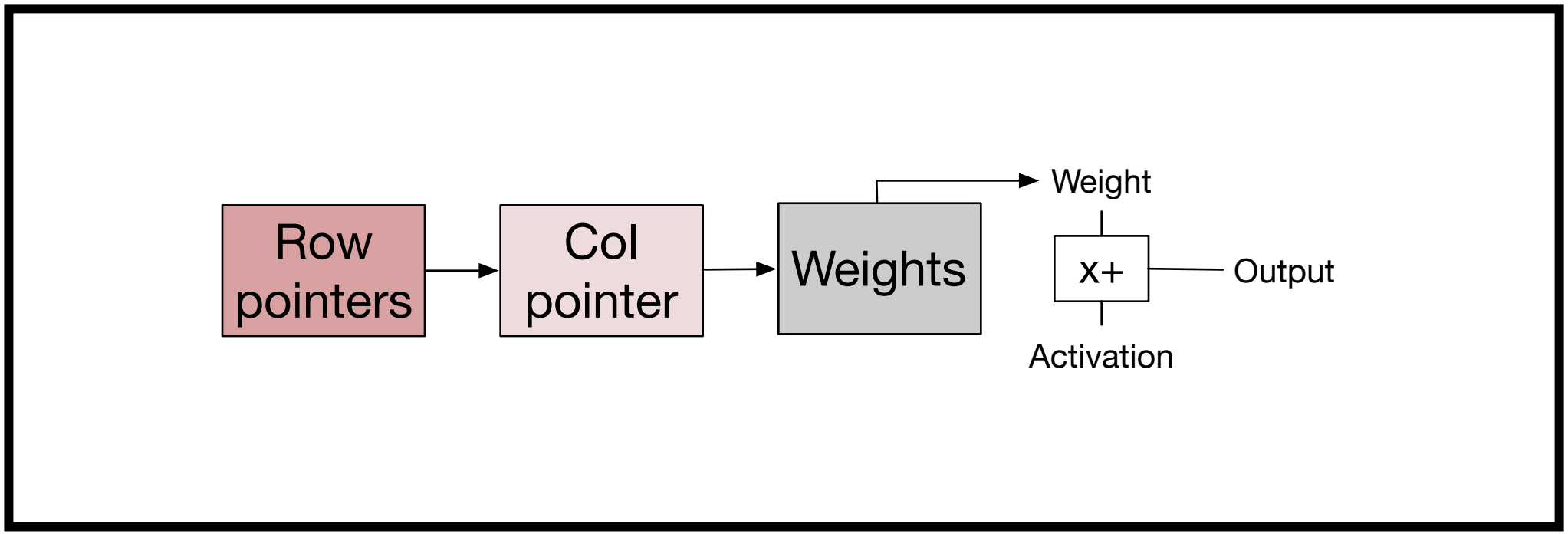
3x additional memory savings from activation compression



Proposed solution: MASR

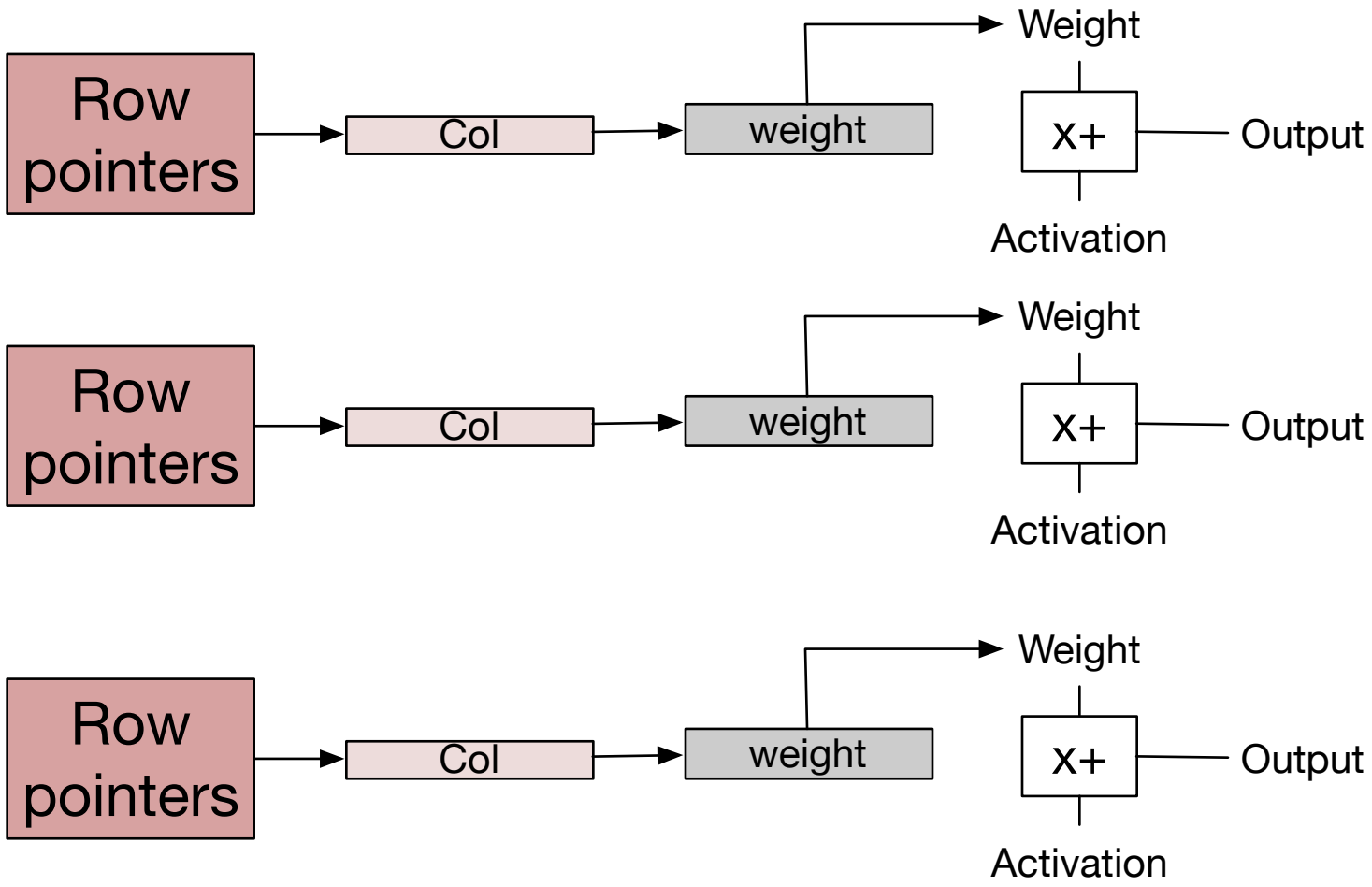
Problem	Solution	Optimizes
 Large memory footprint – static weights and dynamic activations	Logic centric sparse encoding	Area
 **Irregular, sparse computation makes parallelism hard**	**Scalable sparse encoding architecture** Accelerator to exploit parallelism	
High performance with parallelism and irregularity is hard	Work stealing for load balancing	

Memory centric encodings do not scale



Song Han, et. al. ISCA, 2016

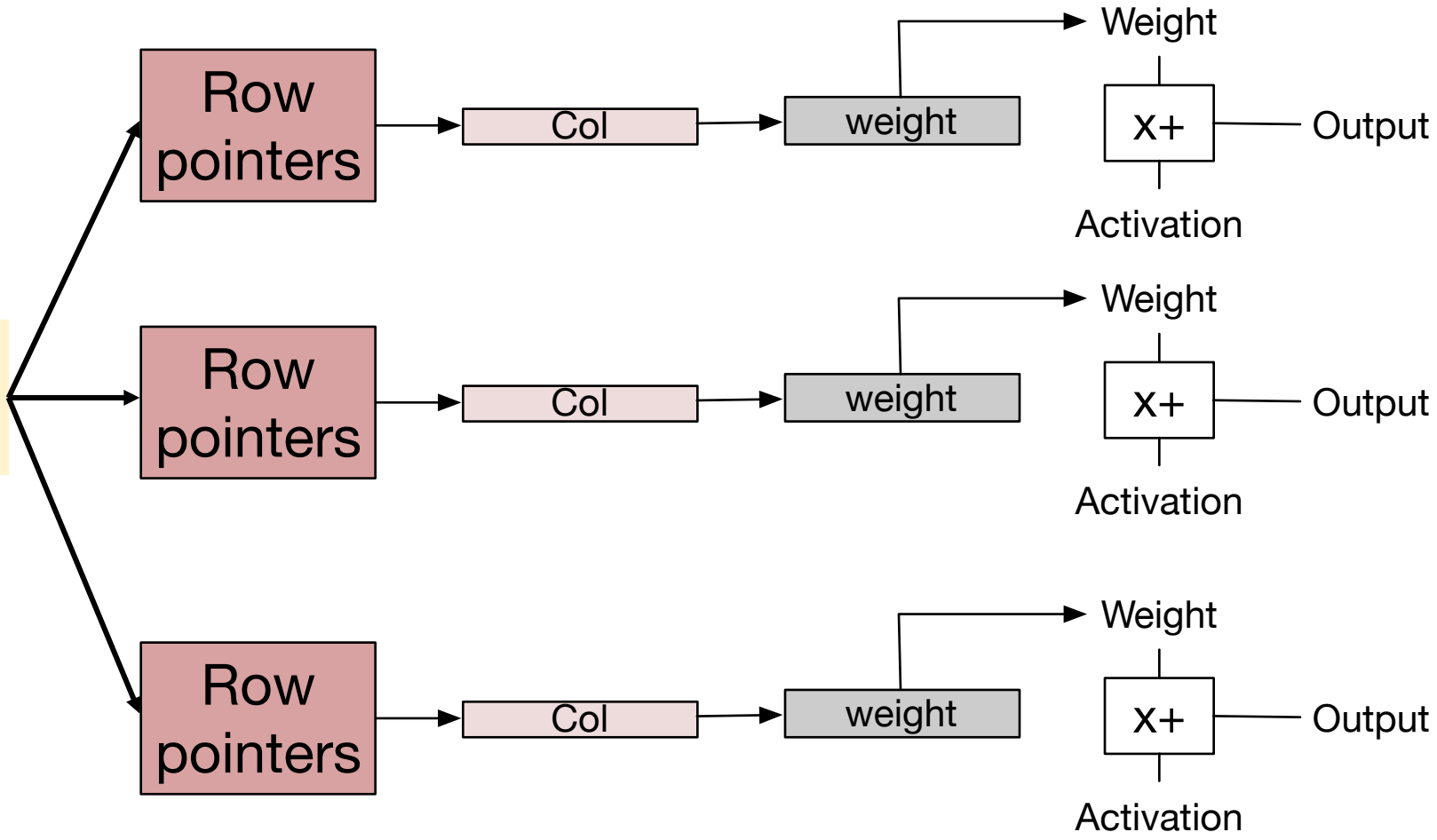
Memory centric encodings do not scale



Song Han, et. al. ISCA, 2016

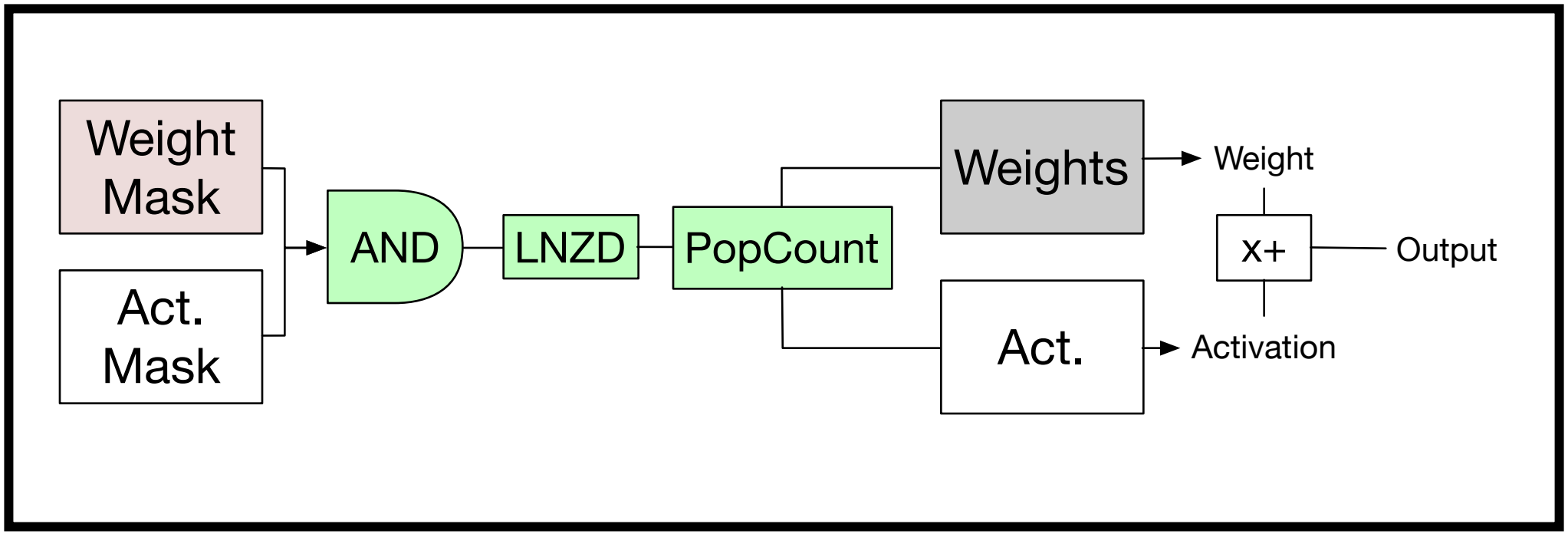
Memory centric encodings do not scale

Duplicating memory is expensive!



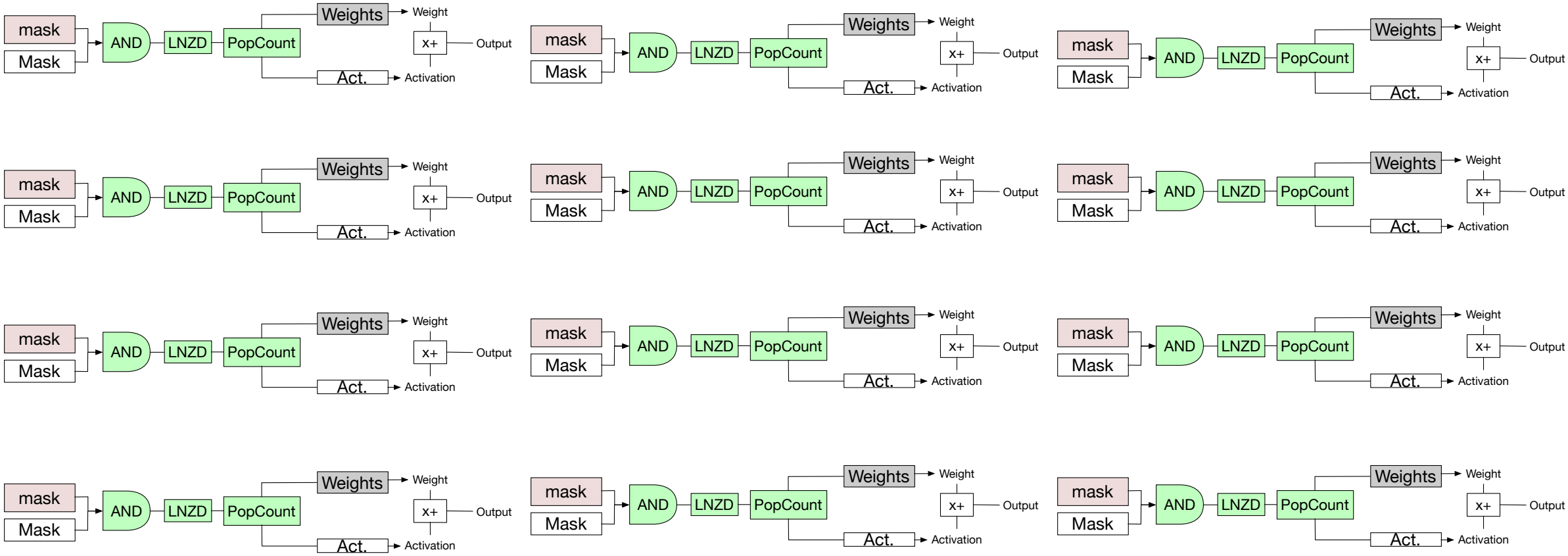
Song Han, et. al. ISCA, 2016

Proposed: parallelize logic centric encoding

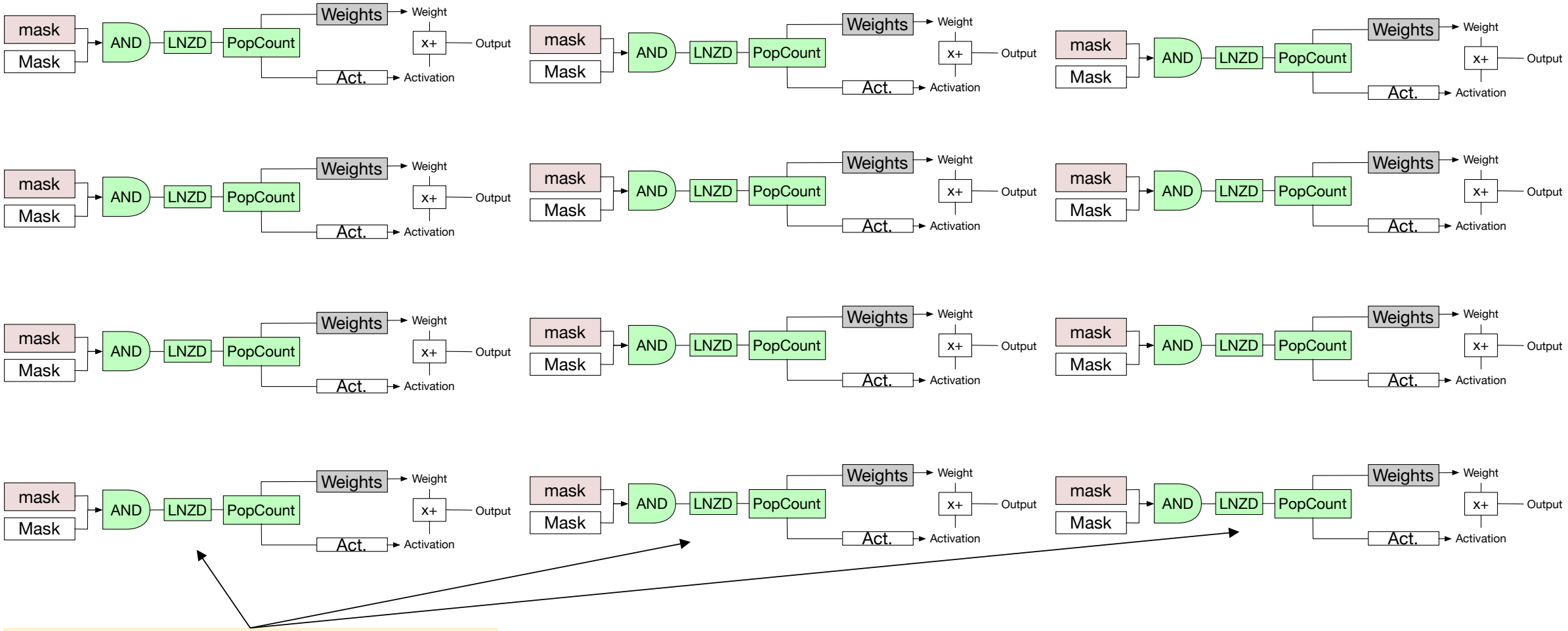


Single MASR Lane

Proposed: parallelize logic centric encoding

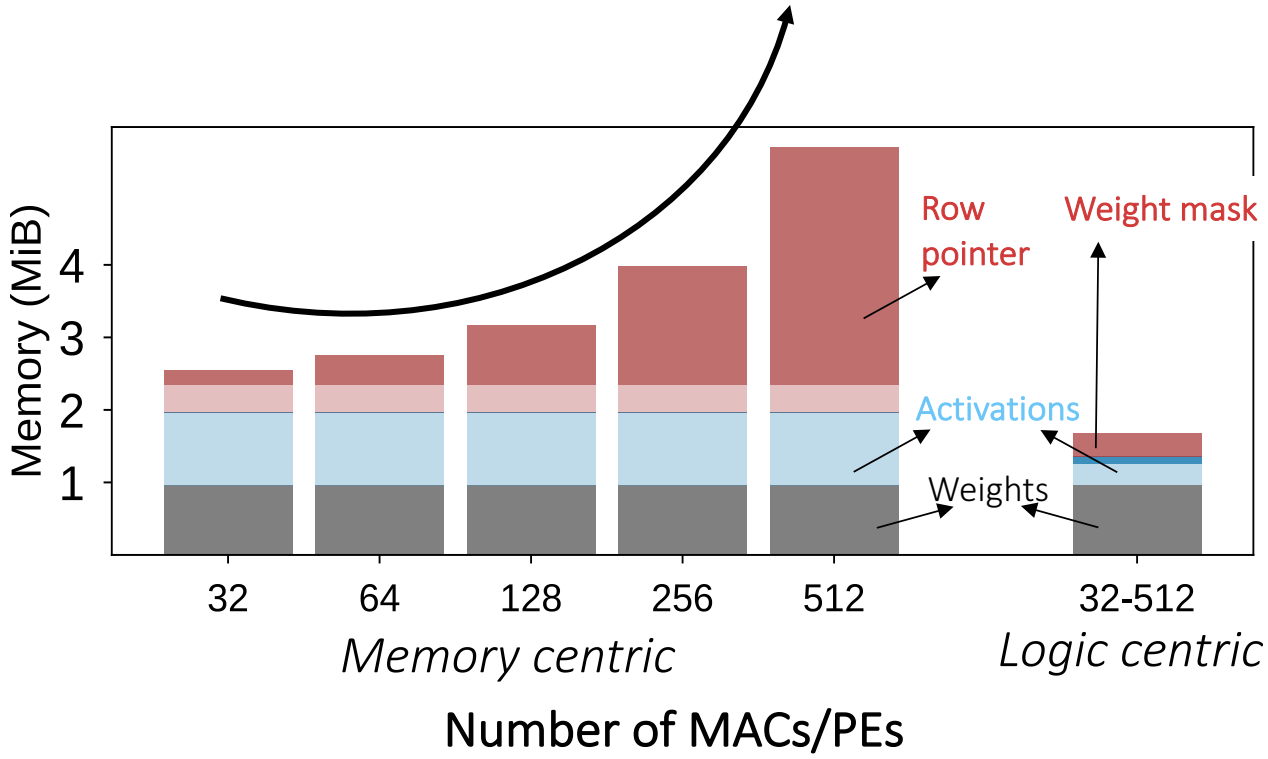


Proposed: parallelize logic centric encoding



Duplicating logic is cheap!

MASR's sparse encoding improves scalability



Takeaways

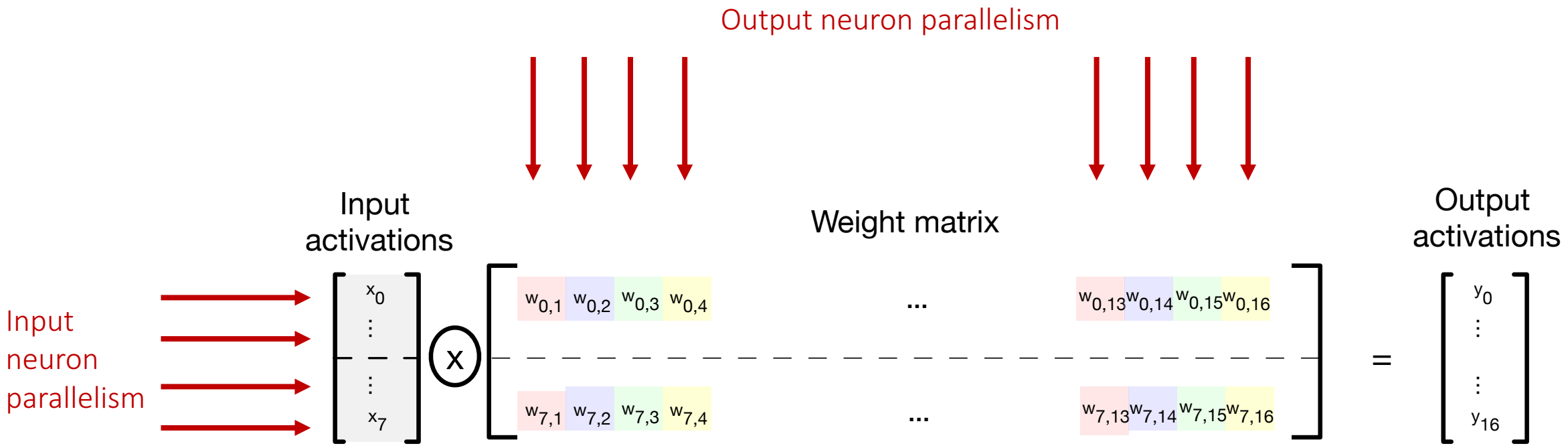
Scalable sparse encoding and architecture

- Enables highly parallel execution with varying number of MACs/PEs

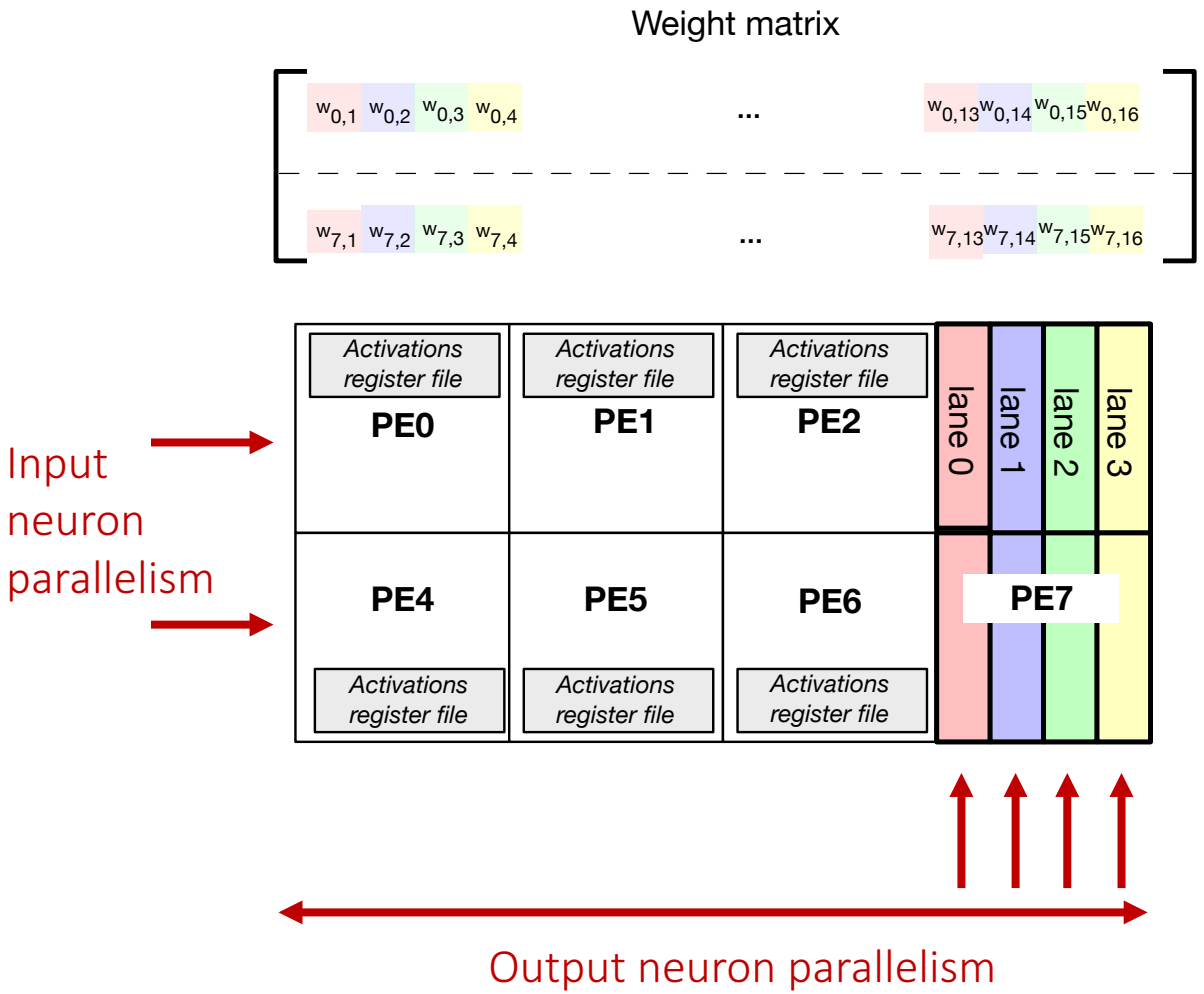
Proposed solution: MASR

Problem	Solution	Optimizes
 Large memory footprint – static weights and dynamic activations	Logic centric sparse encoding	Area
 **Irregular, sparse computation makes parallelism hard**	Scalable sparse encoding architecture **Accelerator to exploit parallelism**	
High performance with parallelism and irregularity is hard	Work stealing for load balancing	

Parallelism within matrix-vector multiplication



MASR micro-architecture: Parallelizing across input and output neurons



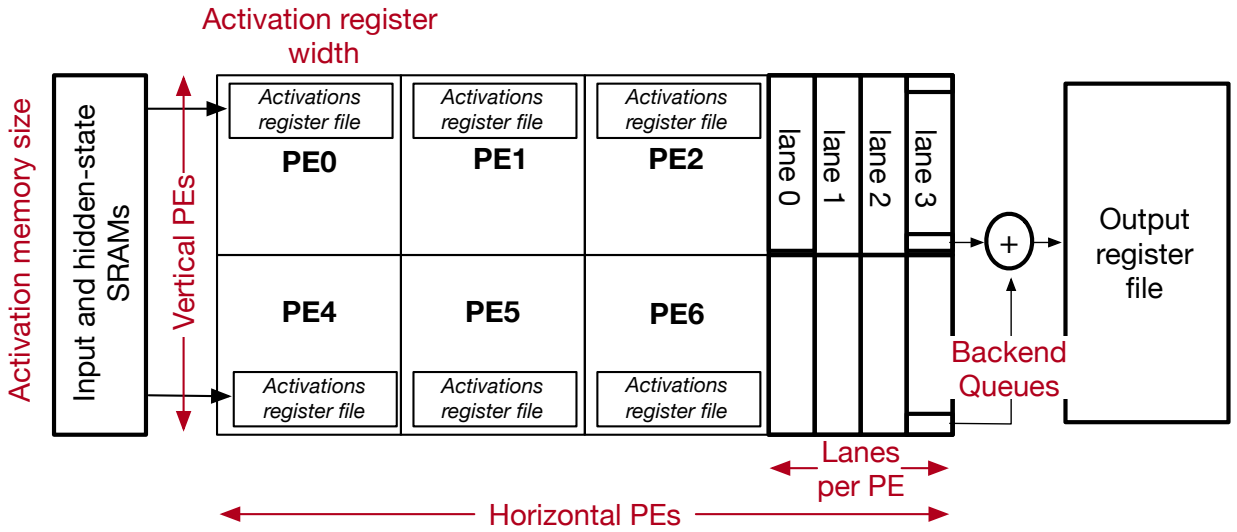
Composed of 2D array of lanes

- Horizontal lanes parallelize output neurons
- Vertical lanes parallelize input neurons

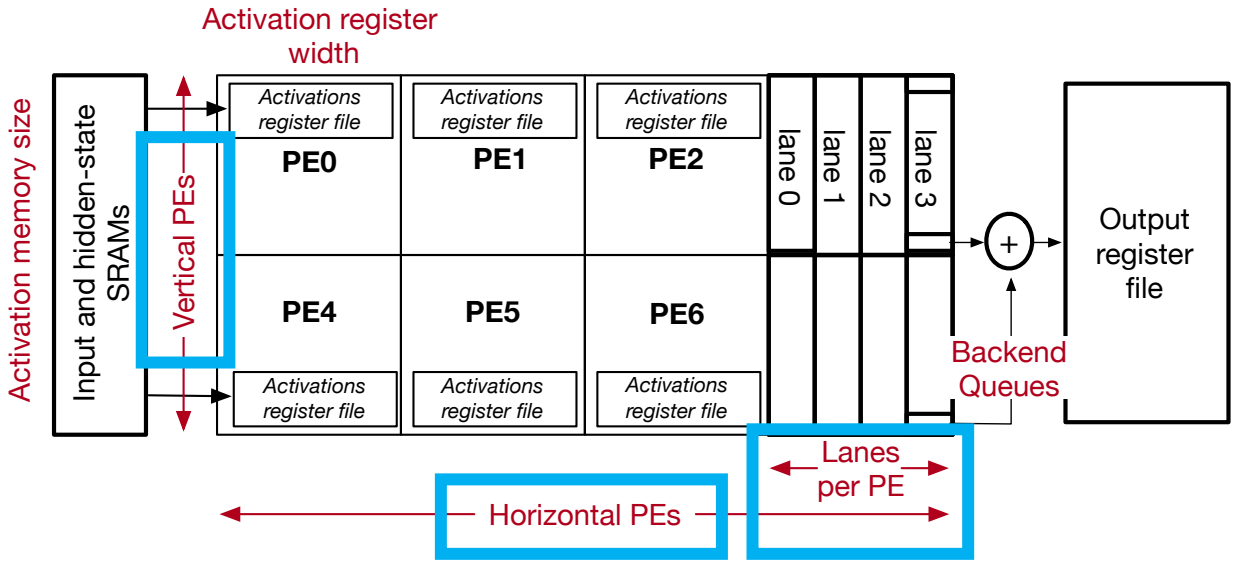
PEs composed of neighboring horizontal lanes

- Share activation register file (area, power, load time)
- Private weight and mask SRAM within lane (decoupled to enable high-bandwidth access)

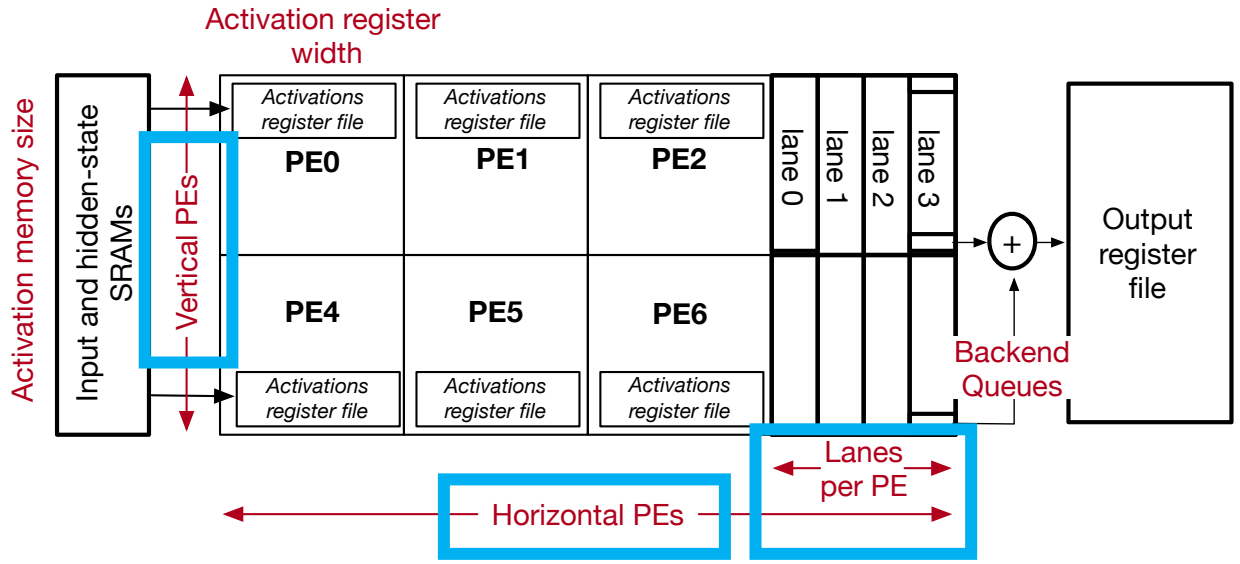
MASR: design space exploration

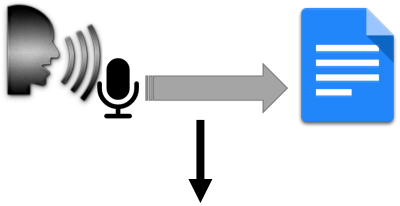


MASR: design space exploration



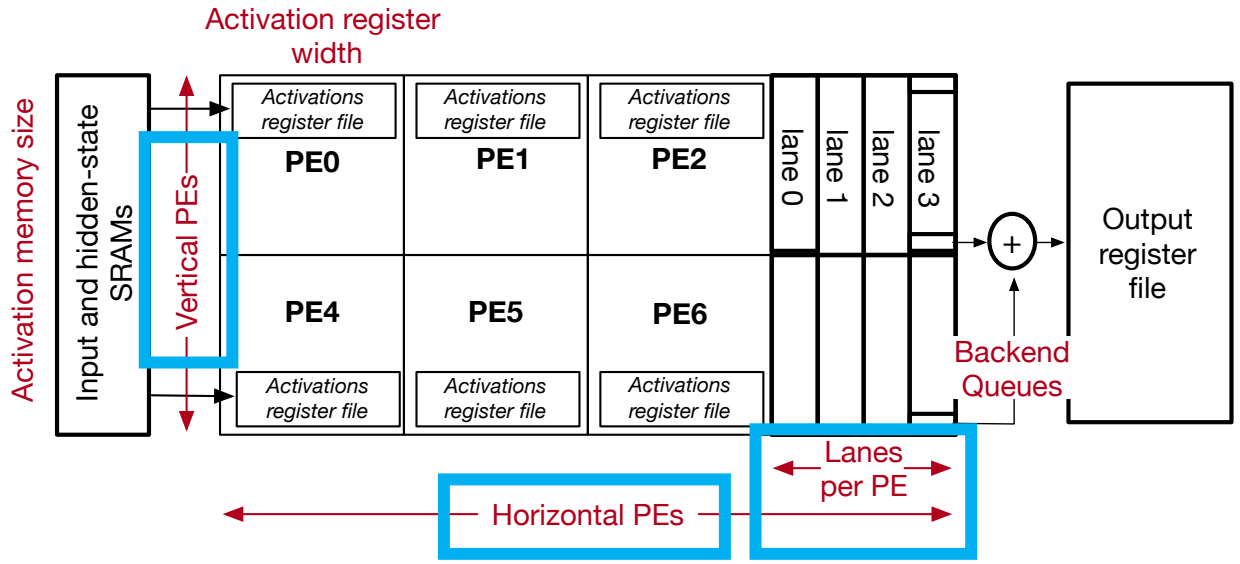
MASR: design space exploration

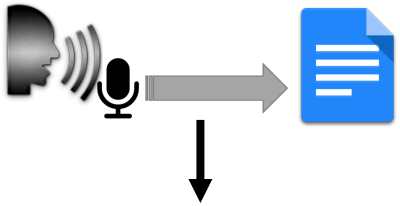




Recurrent Neural Network

MASR: design space exploration





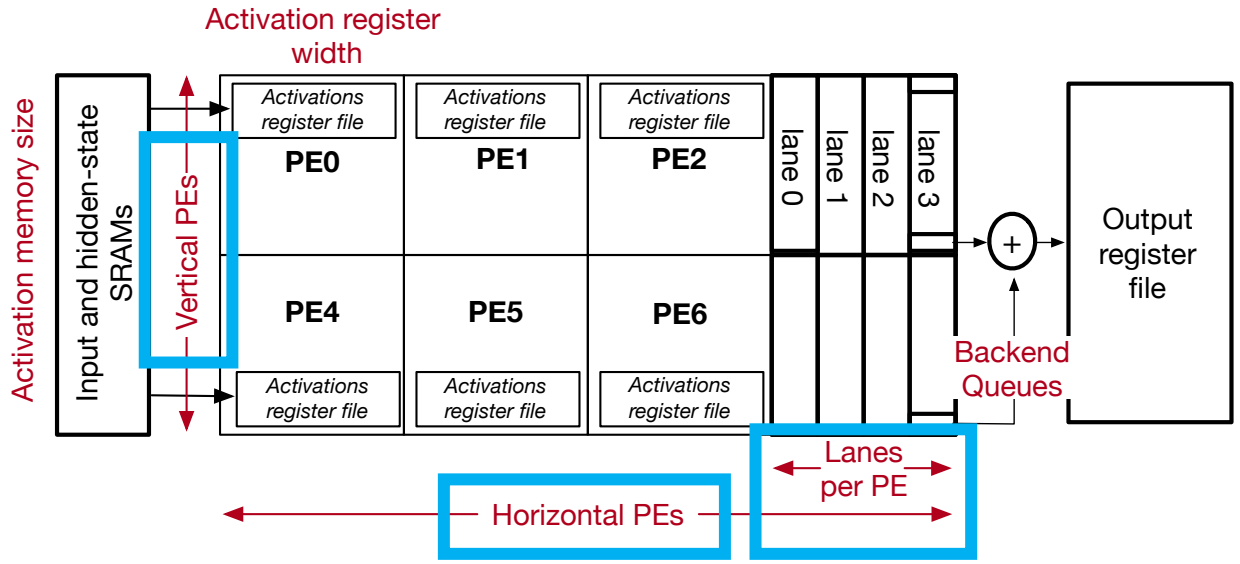
Recurrent Neural Network

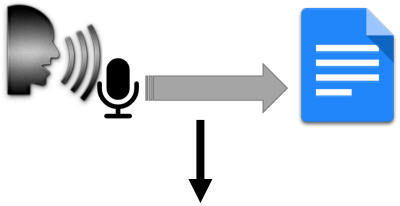
1GHz

16nmFinFet

Cycle accurate simulation

MASR: design space exploration

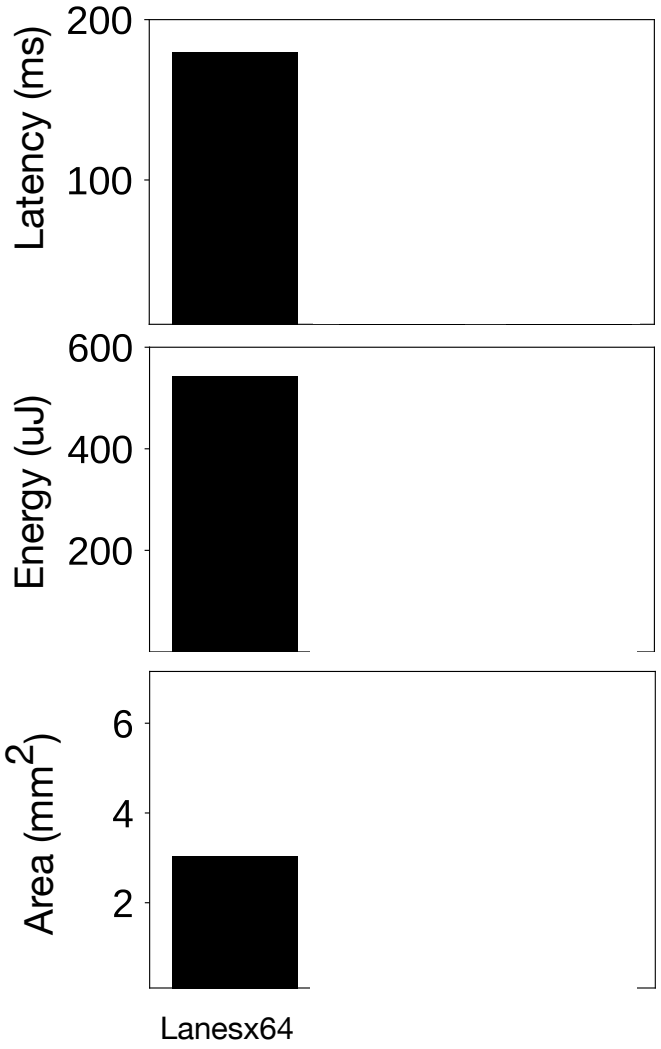




1GHz 16nmFinFet

Recurrent Neural Network → Cycle accurate simulation → Design space exploration

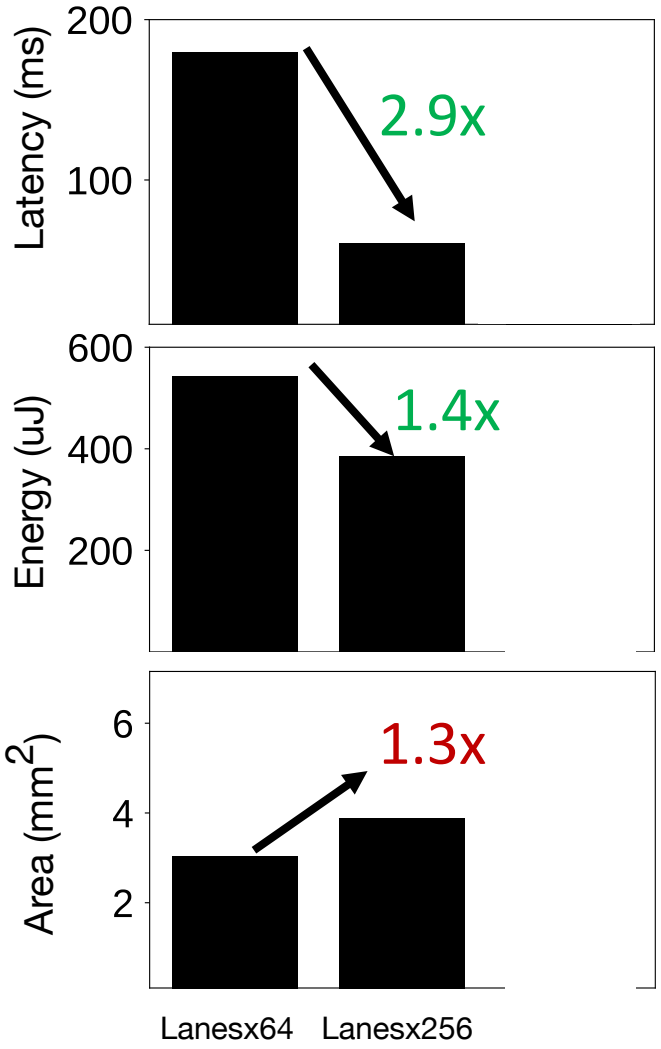
MASR: Performance, Energy, Area tradeoffs



MASR is *2 orders* of magnitude faster than CPU

Fits our *on-chip* area target for mobile devices

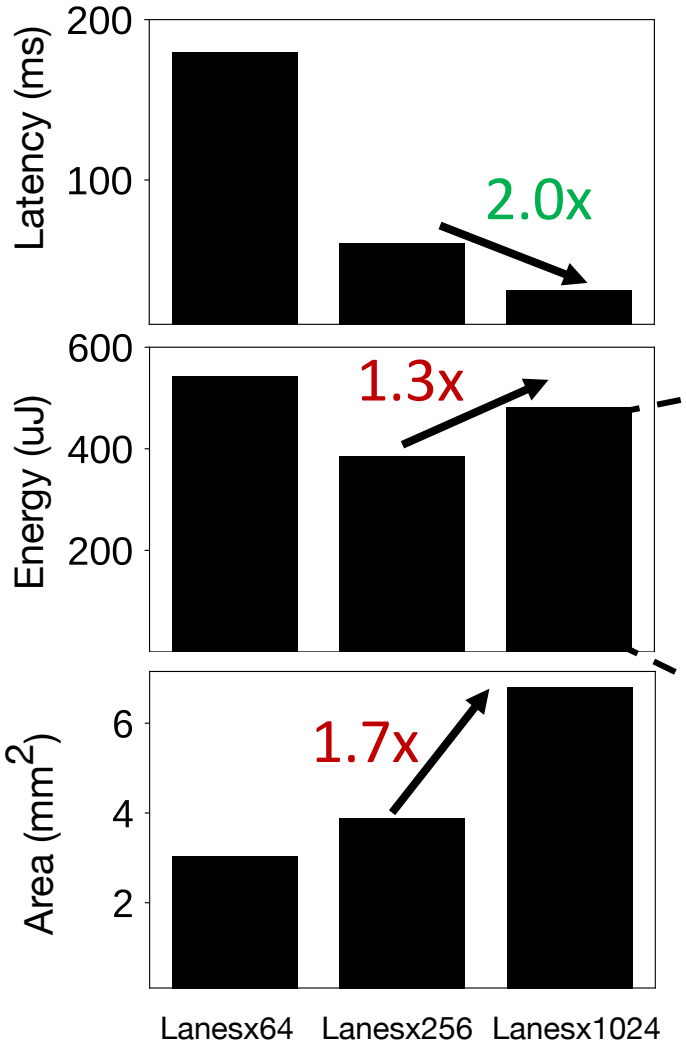
MASR: Performance, Energy, Area tradeoffs



MASR is *2 orders* of magnitude faster than CPU

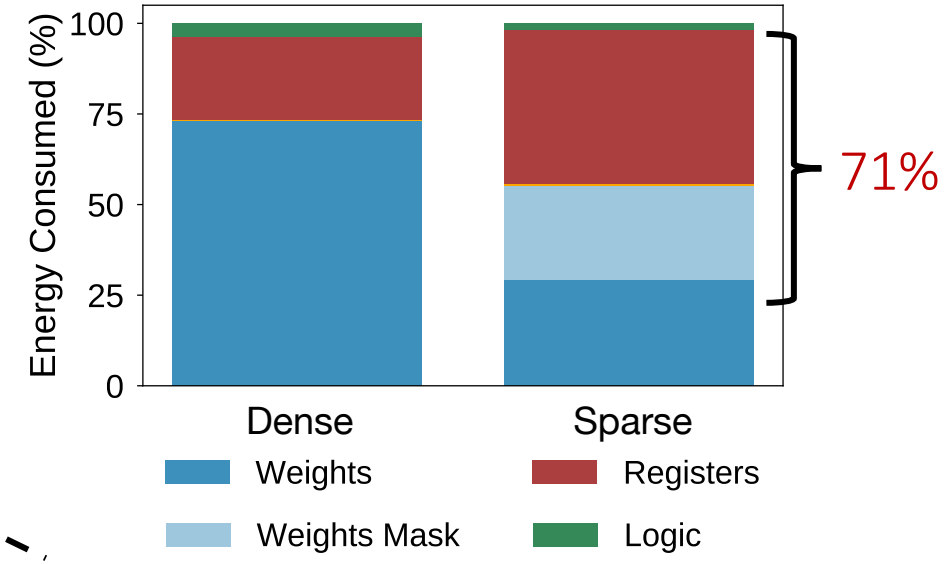
Fits our *on-chip* area target for mobile devices

MASR: Performance, Energy, Area tradeoffs

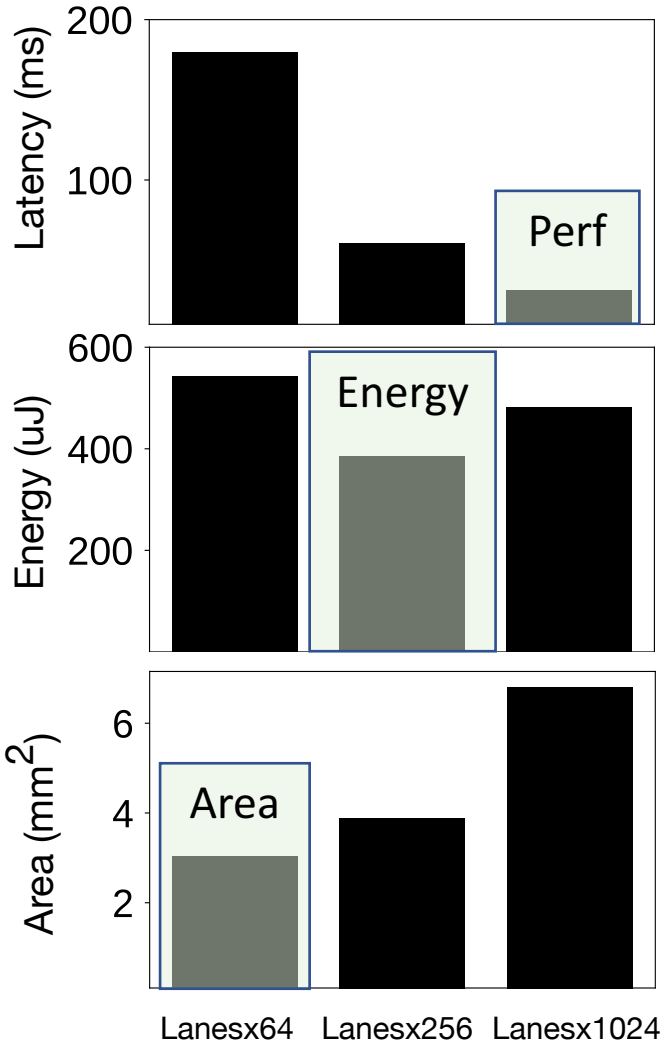


MASR is *2 orders* of magnitude faster than CPU

Fits our *on-chip* area target for mobile devices



MASR: Performance, Energy, Area tradeoffs



MASR is *2 orders* of magnitude faster than CPU

Fits our *on-chip* area target for mobile devices

Takeaways

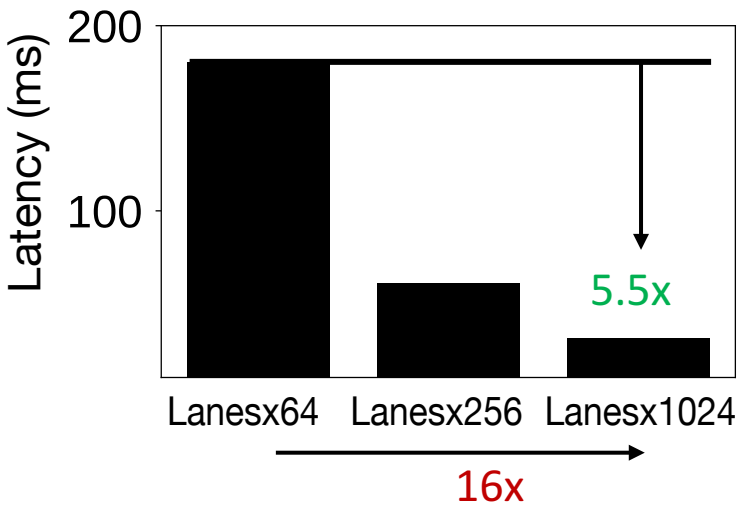
Parallelism can be configured to target:

- High-performance
- Energy-efficiency
- Area-efficiency

Proposed solution: MASR

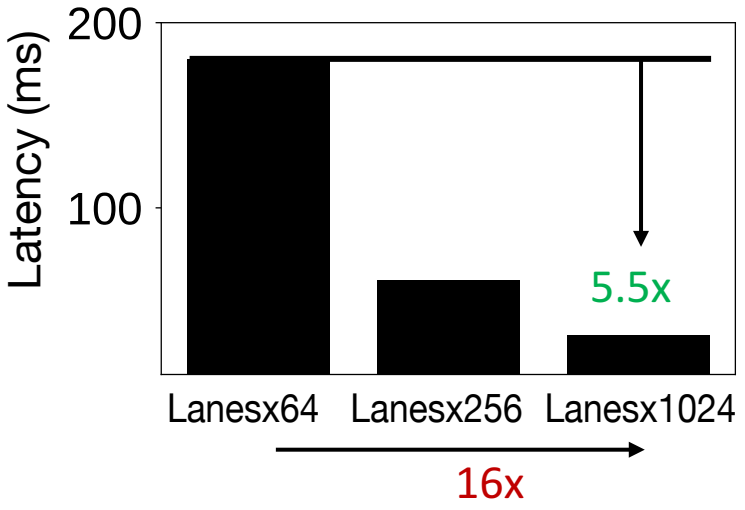
Problem	Solution	Optimizes
 Large memory footprint – static weights and dynamic activations	Logic centric sparse encoding	Area
 Irregular, sparse computation makes parallelism hard	Scalable sparse encoding architecture Accelerator to exploit parallelism	Performance Area Energy
 **High performance with parallelism and irregularity is hard**	**Work stealing for load balancing**	

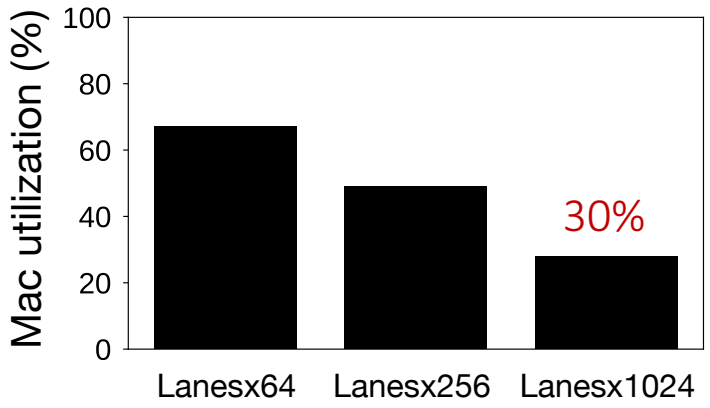
Further investigating sources of inefficiency



Increasing number of parallel MACs/lanes from 64 to 1024 (16x), improves performance by 5.5x

Further investigating sources of inefficiency



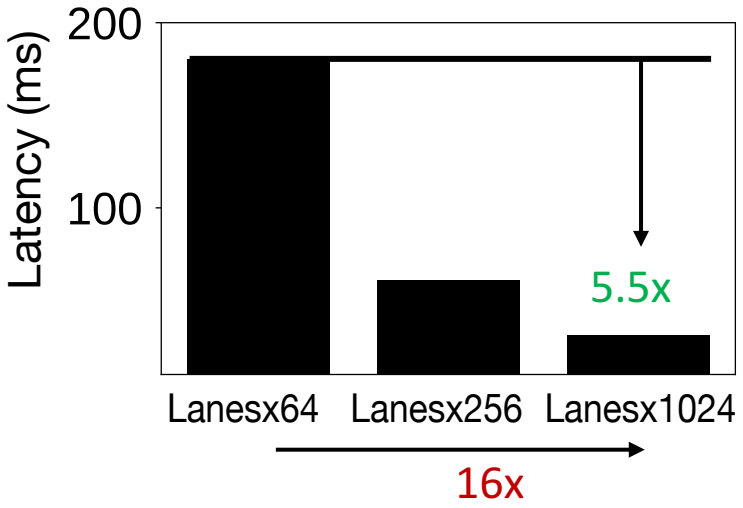


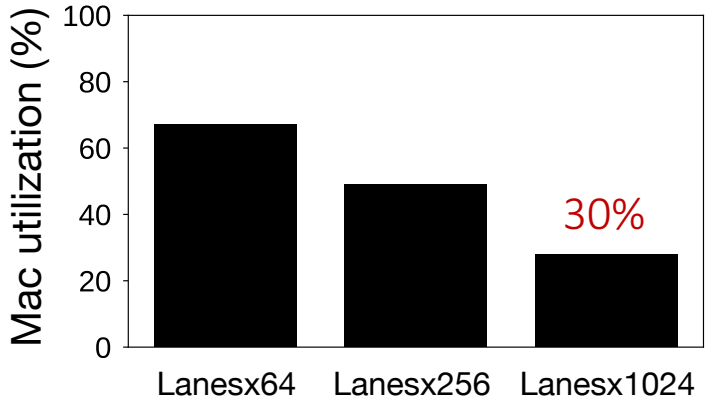
Increasing number of parallel MACs/lanes from 64 to 1024 (16x), improves performance by 5.5x

30% MAC utilization

Remainder spent on stalls/idles due to load imbalance

Further investigating sources of inefficiency





Increasing number of parallel MACs/lanes from 64 to 1024 (16x), improves performance by 5.5x

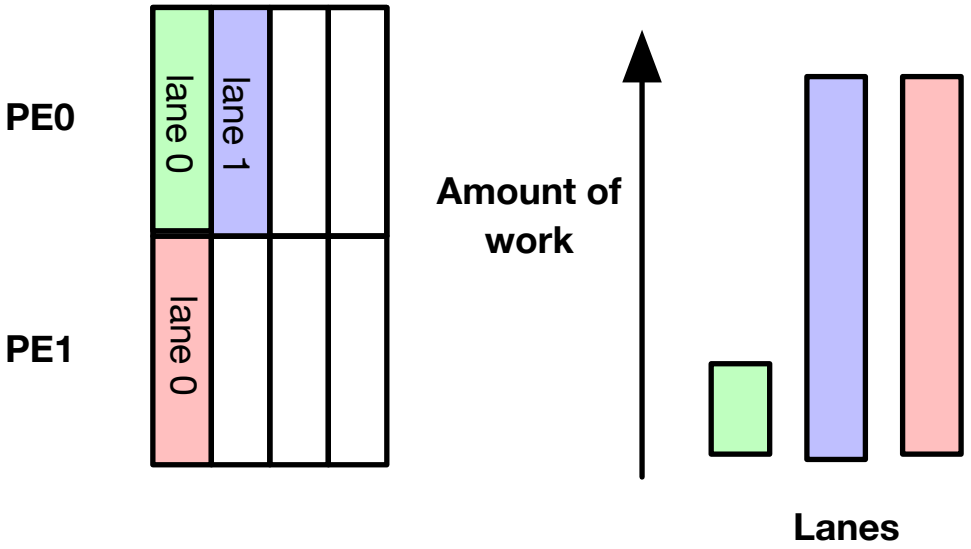
30% MAC utilization

Remainder spent on stalls/idles due to load imbalance

Some lanes get 1.5x more work (non-zeros) to process

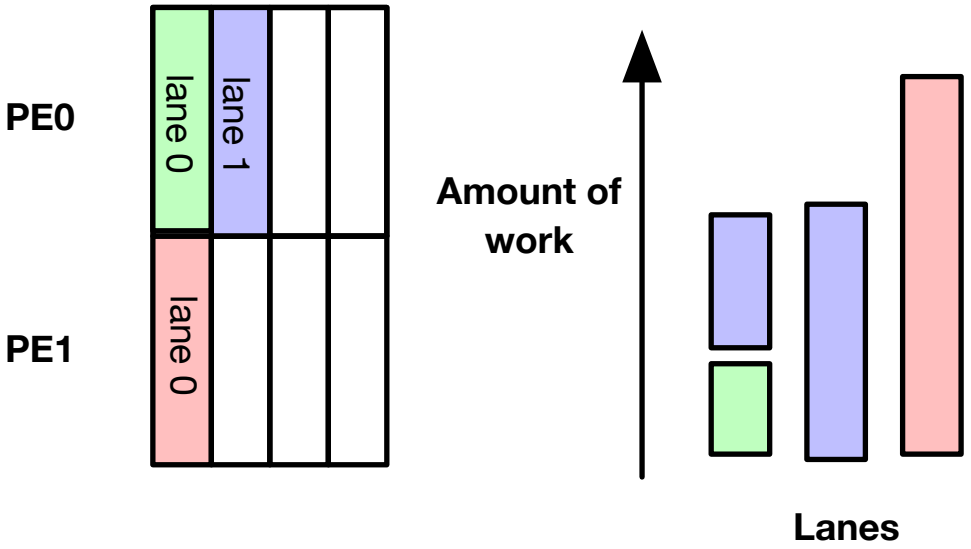
Dynamic load balancing

Lanes that finish early can steal work from neighboring lanes that are straggling behind



Dynamic load balancing

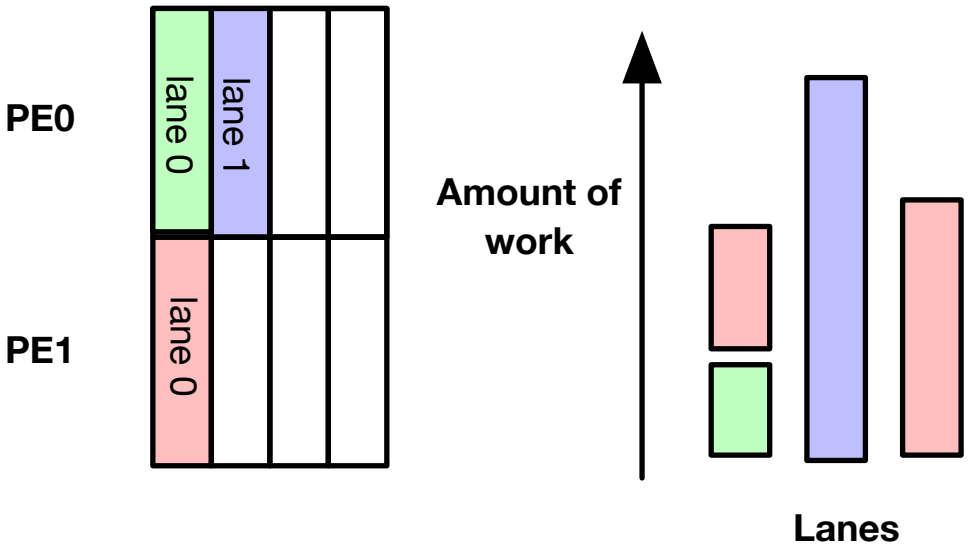
Lanes that finish early can steal work from neighboring lanes that are straggling behind



Horizontal load balancing requires duplicating weights

Dynamic load balancing

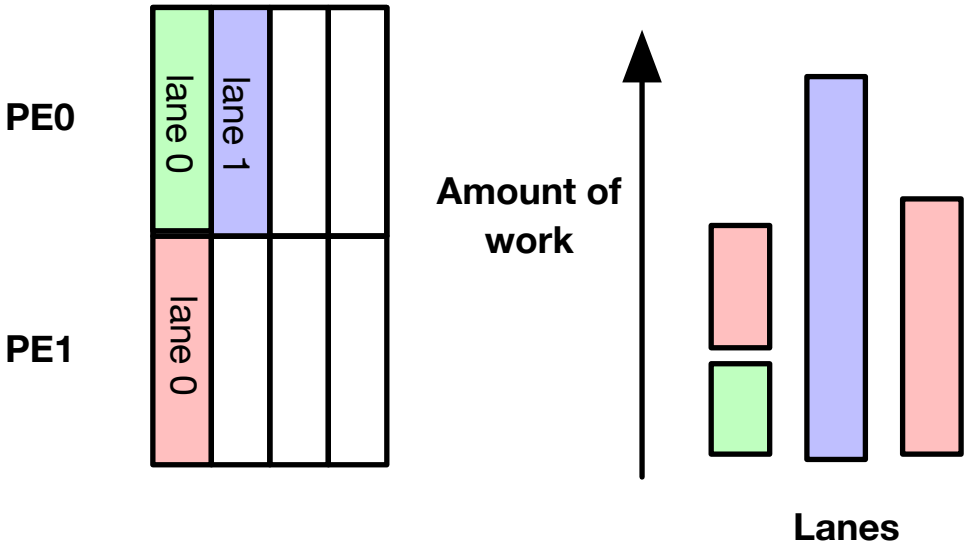
Lanes that finish early can steal work from neighboring lanes that are straggling behind

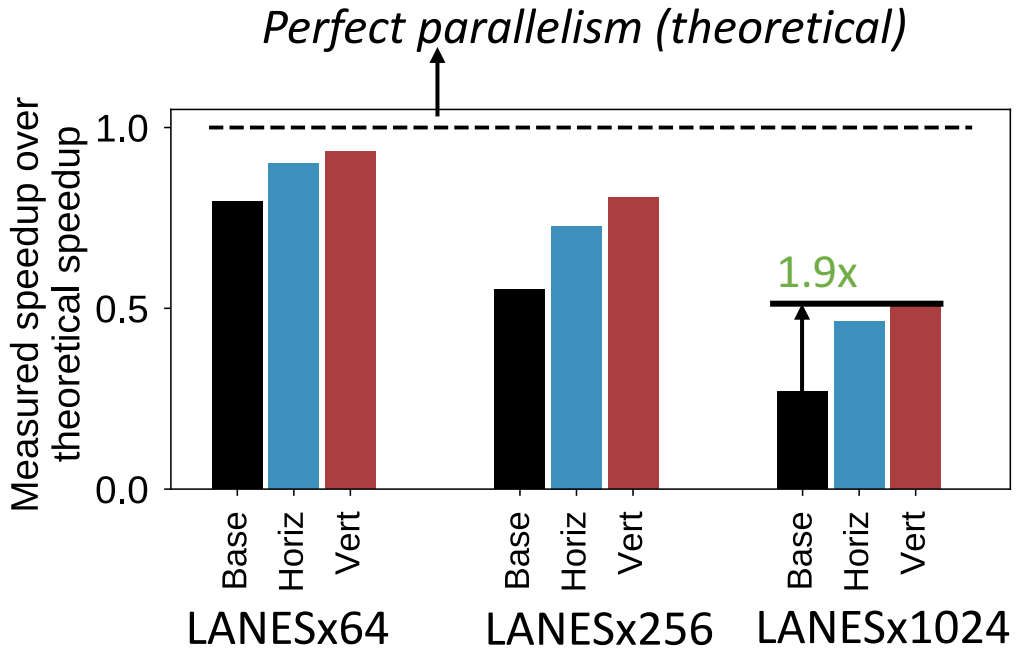


Horizontal load balancing requires duplicating weights

Vertical load balancing requires duplicating weights and activation register files

Dynamic load balancing





Vertical load balancing better targets load imbalance in dynamic activations

- Up to 1.9x speedup (LANESx1024)
- Requires duplicating 10% weight storage and activation register files

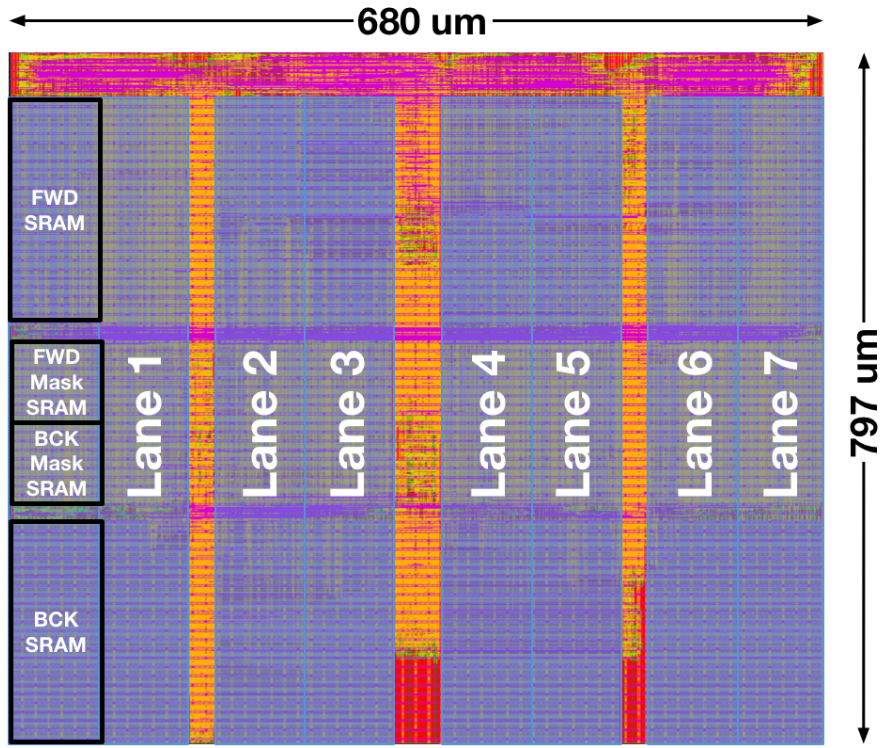
Proposed solution: MASR

	Problem	Solution	Optimizes
	Large memory footprint – static weights and dynamic activations	Logic centric sparse encoding	Area
	Irregular, sparse computation makes parallelism hard	Scalable sparse encoding architecture Accelerator to exploit parallelism	Performance Area Energy
	High performance with parallelism and irregularity is hard	Work stealing for load balancing	Performance

Over state-of-the-art, MASR provides:

3x area **3x** energy **2x** perf

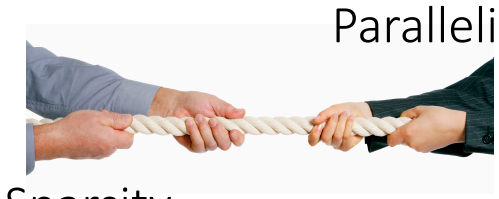
Scalable acceleration of sparse RNNs is possible!



Stay tuned…

MASR: A Modular Accelerator for Sparse RNNs

Udit Gupta, Brandon Reagen,
Lillian Pentecost, Marco Donato, Thierry Tambe
Alexander M. Rush, Gu-Yeon Wei, David Brooks



ugupta@g.harvard.edu

Thanks for listening!