uGEMM: Unary Computing Architecture for GEMM Applications

Di Wu, Jingjie Li, Ruokai Yin, Hsuan Hsiao (University of Toronto), Younghyun Kim and Joshua San Miguel

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Executive Summary

- Review the demand of energy-efficient GEMM implementations, and motivate uGEMM over other unary approaches.
- Demonstrate uGEMM's compatibility for arbitrarily encoded inputs for multiplication and addition mathematically.
- Prove the knob of uGEMM's high energy efficiency to be early termination enabled by high accuracy and stability.

Outline

- ❑ **Background**
- ❑ Motivation
- ❑ Architecture
- ❑ Evaluation

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

GEneral Matrix Multiply (GEMM)

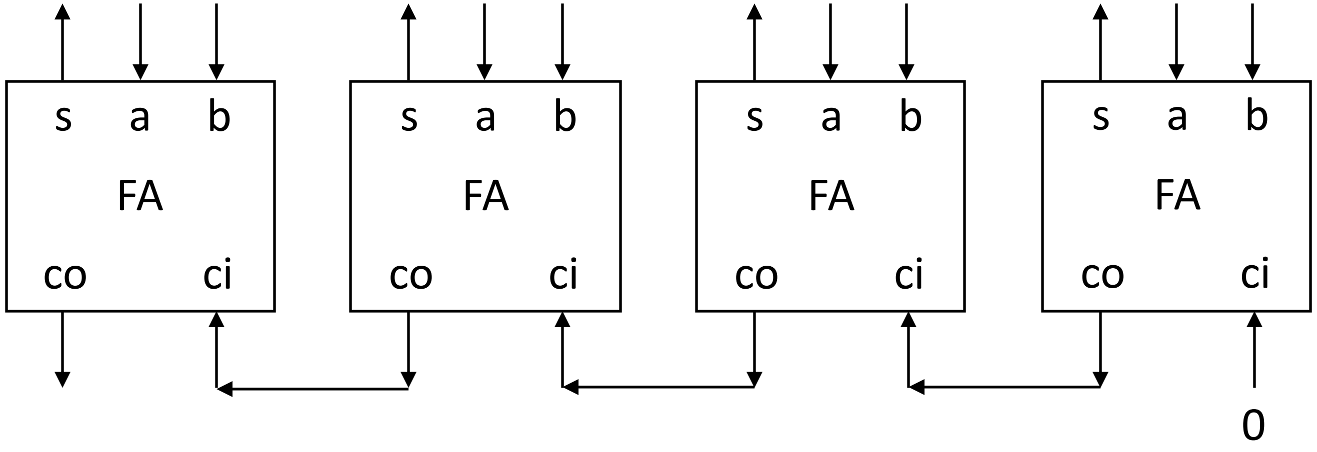
- Ubiquitous in applications
  - Computer vision
  - Signal processing
  - Machine learning
- "At the center of Deep Learning"
  - 95% of GPU runtime
  - 89% of CPU runtime
- Energy efficiency
  - Unary computing as salvation

Y. Jia, "Learning Semantic Image Representations at a Large Scale," PhD thesis.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

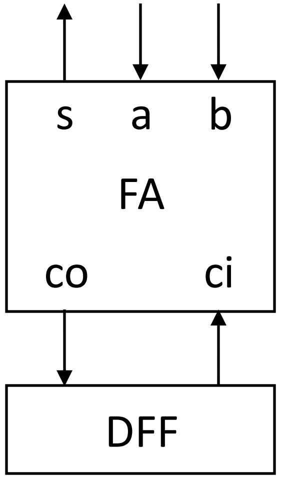
Computing paradigm

Paradigm		Data	Bit significance	Computing domain
Binary computing	Bit parallel	Parallel	Varying	Spatial
	Bit serial			
Unary computing				



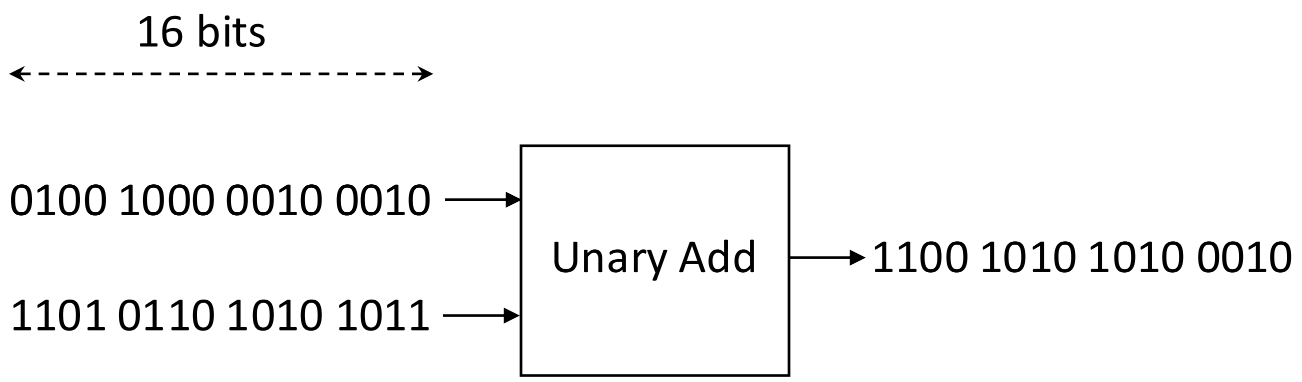
Computing paradigm

Paradigm		Data	Bit significance	Computing domain
Binary computing	Bit parallel	Parallel	Varying	Spatial
	Bit serial	Serial	Varying	Temporal
Unary computing				



Computing paradigm

Paradigm		Data	Bit significance	Computing domain
Binary computing	Bit parallel	Parallel	Varying	Spatial
	Bit serial	Serial	Varying	Temporal
Unary computing		Serial	Equal	Temporal

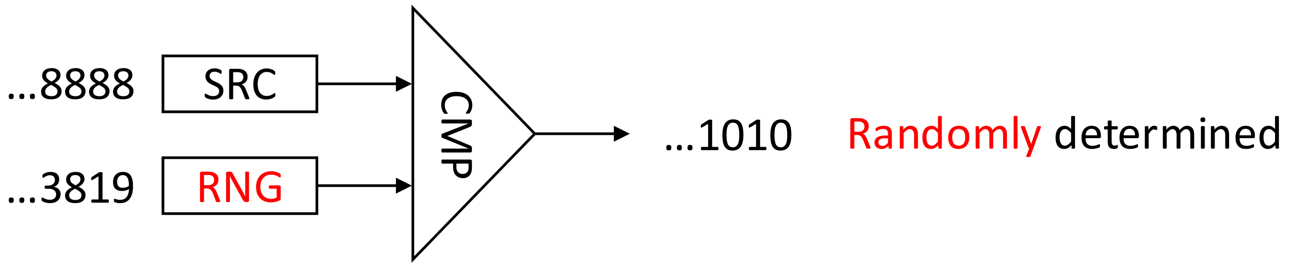


Unary computing scheme

Scheme	Bit stream	Application
Stochastic computing	Rate coding	LDPC, image processing, machine learning
Race logic		

Bit stream as data.
Probability of 1s matters only.

1101 1000 1010 1010 8 1s

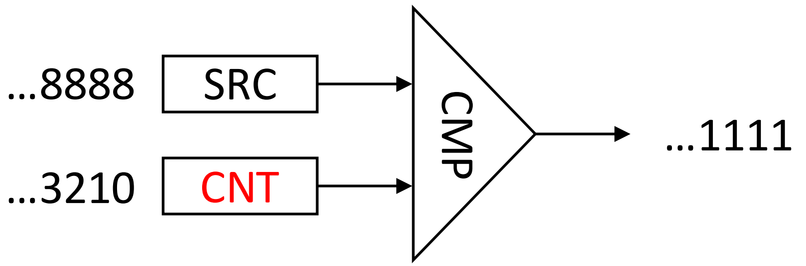


Unary computing scheme

Scheme	Bit stream	Application
Stochastic computing	Rate coding	LDPC, image processing, machine learning
Race logic	Temporal coding	DNA sequencing, decision tree, sorting

Bit stream as data.
Delay of 1s matters only.

0000 0000 1111 1111 8 1s



Deterministically determined

Unary computing data – bit stream

Polarity	Binary width	Unary length	Probability of 1s	Value	Range
Unipolar	N	2^N	P(1)	P(1)	Unsigned, [0, 1]
Bipolar					

1101 1000 1010 1010 8 1s Unipolar value=0.5

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Unary computing data – bit stream

Polarity	Binary width	Unary length	Probability of 1s	Value	Range
Unipolar	N	2^N	P(1)	P(1)	Unsigned, [0, 1]
Bipolar				2*P(1)-1	Signed, [-1, 1]

1101 1000 1010 1010 8 1s Bipolar value=0.0

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Measure of bit streams

Metric	Bit stream	Goal
Correlation	Two	How similar two bit streams are.
Stability		

+1 correlation: count of aligned 1s is maximized.

1101 1000 1010 1010   8 1s
1101 1000 1010 1010   8 1s

Aligned 1 count: 8

Expected in temporal coding

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Measure of bit streams

Metric	Bit stream	Goal
Correlation	Two	How similar two bit streams are.
Stability		

+1 correlation: count of aligned 1s is maximized.

1101 1000 1010 1010	8 1s	Aligned 1 count: 8
1101 1000 1010 1010	8 1s	

0 correlation: count of aligned 1s is balanced.

1101 1000 1010 1010	8 1s	Aligned 1 count: 4
1110 0100 0101 1010	8 1s	

Expected in rate coding

Measure of bit streams

Metric	Bit stream	Goal
Correlation	Two	How similar two bit streams are.
Stability		

+1 correlation: count of aligned 1s is maximized.
- 1101 1000 1010 1010 — 8 1s
- 1101 1000 1010 1010 — 8 1s

Aligned 1 count: 8

0 correlation: count of aligned 1s is balanced.
- 1101 1000 1010 1010 — 8 1s
- 1110 0100 0101 1010 — 8 1s

Aligned 1 count: 4

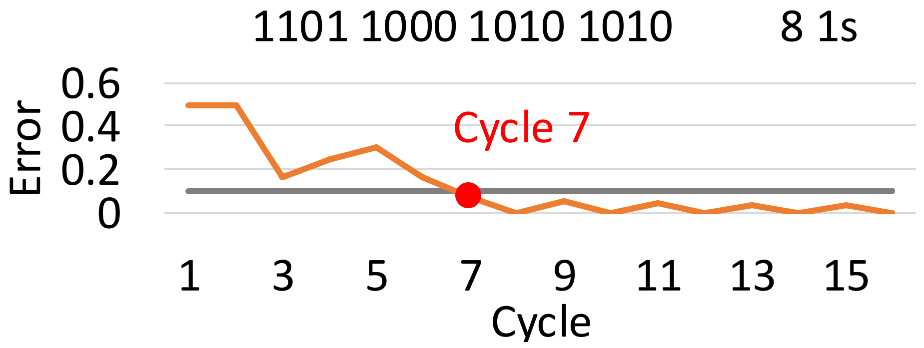
-1 correlation: count of aligned 1s is minimized.
- 1101 1000 1010 1010 — 8 1s
- 0010 0111 0101 0101 — 8 1s

Aligned 1 count: 0

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Measure of bit streams

Metric	Bit stream	Goal
Correlation	Two	How similar two bit streams are.
Stability	One	How fast a bit stream converges to its desired value.



Stability = 1-(7-1)/16 = 0.625

Stable point:

Error never exceeds a given threshold from now on.

Stability:

Ratio of stable cycle count to all.

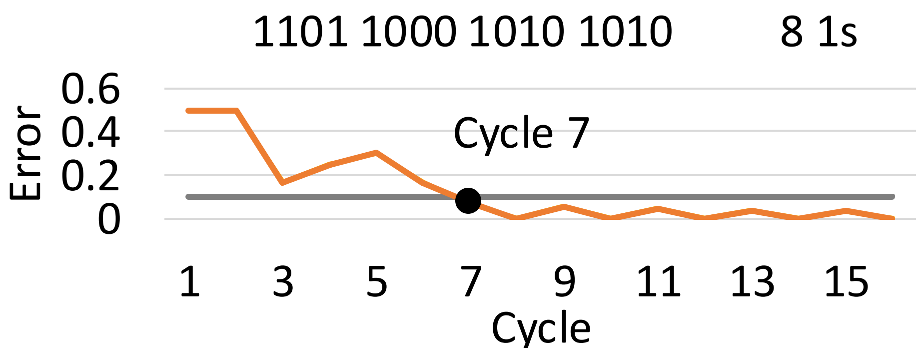
Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Measure of bit streams

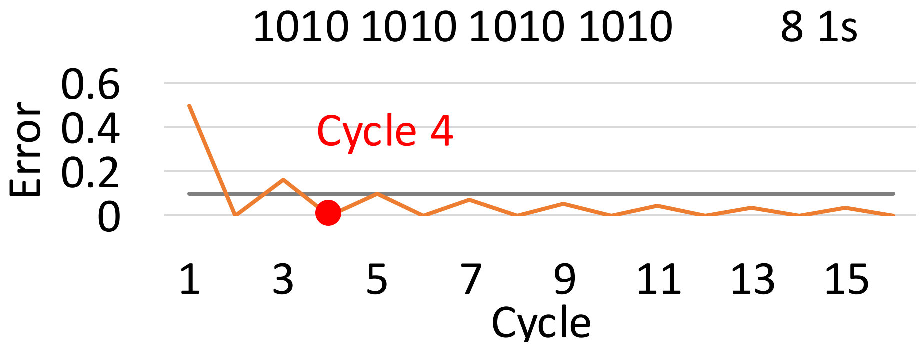
Metric	Bit stream	Goal
Correlation	Two	How similar two bit streams are.
Stability	One	How fast a bit stream converges to its desired value.

Stable point:
Error never exceeds a given threshold from now on.

Stability:
Ratio of stable cycle count to all.



Stability = 1-(7-1)/16 = 0.625



Stability = 1-(4-1)/16 = 0.8125

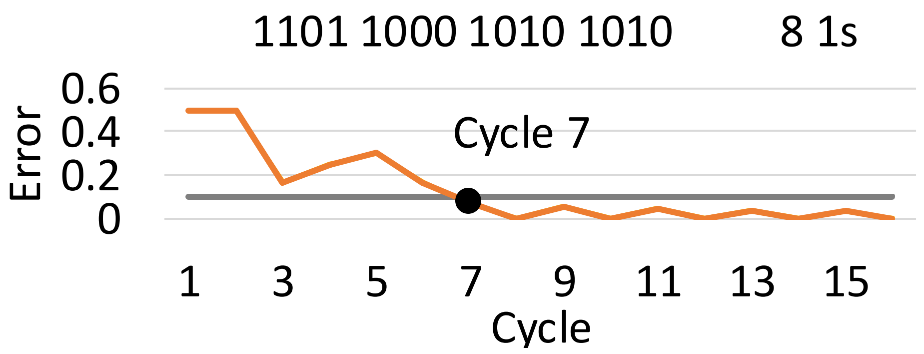
Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Measure of bit streams

Metric	Bit stream	Goal
Correlation	Two	How similar two bit streams are.
Stability	One	How fast a bit stream converges to its desired value.

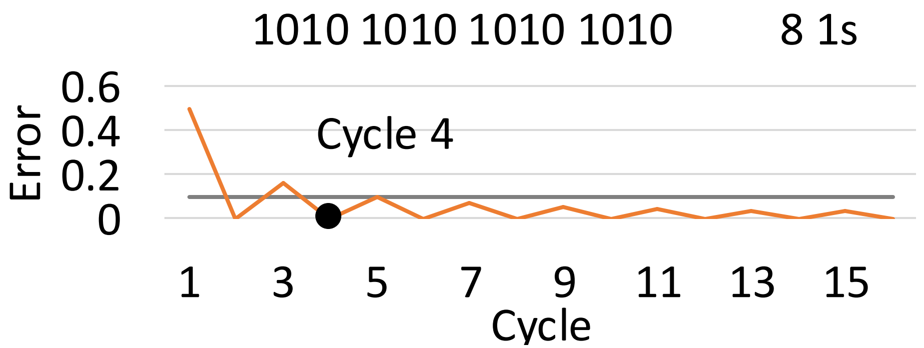
Stable point:
Error never exceeds a given threshold from now on.

Stability:
Ratio of stable cycle count to all.



Stability = 1-(7-1)/16 = 0.625

High stability enables early termination.



Stability = 1-(4-1)/16 = 0.8125

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Outline

- ❑ Background
- ❑ **Motivation**
- ❑ Architecture
- ❑ Evaluation

Unified uGEMM

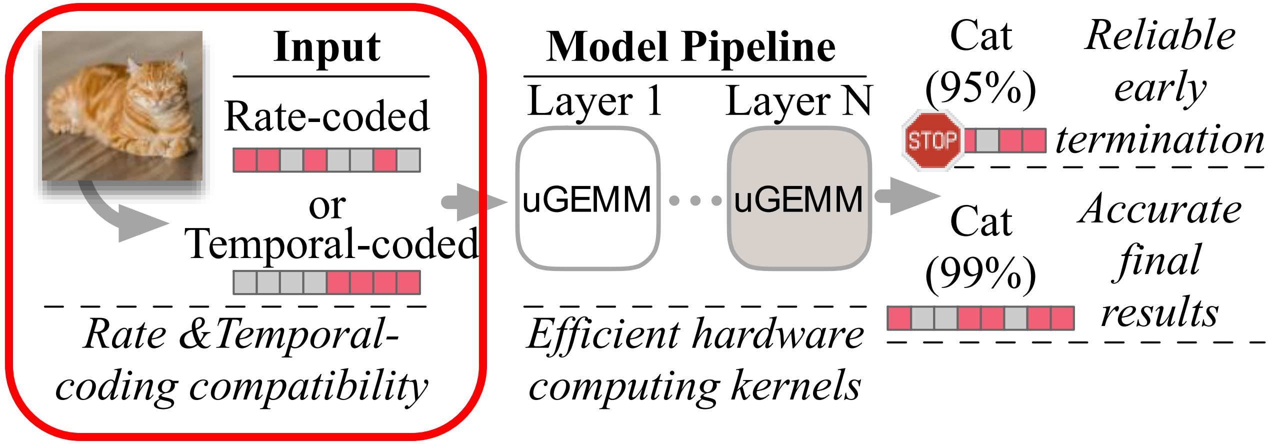
- Unified unary General Matrix Multiplication
  - First to support Add/Mul in temporal coding

Function	Rate coding	Temporal coding
Add	[14, 16, 19, 25], uGEMM	uGEMM
Multiply	[14, 16, 19, 57], uGEMM	uGEMM
Add constant		[39, 63]
Minimum	[4, 32]	[38, 63]
Maximum	[4, 32]	[38, 63]
Inhibit		[59, 63]

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Unified uGEMM

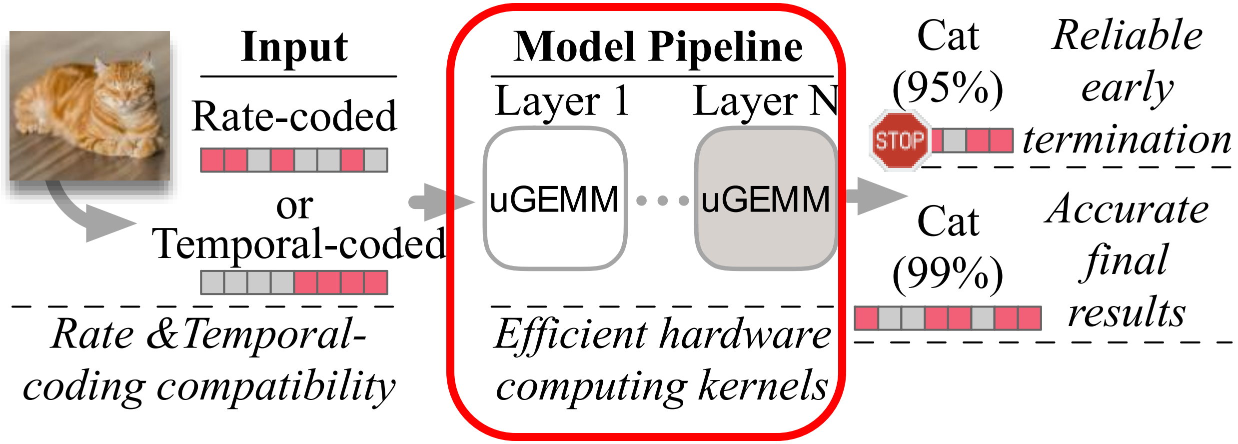
- Unified unary General Matrix Multiplication
  - First to support Add/Mul in temporal coding
  - Compatibility for varying coding and polarity



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Unified uGEMM

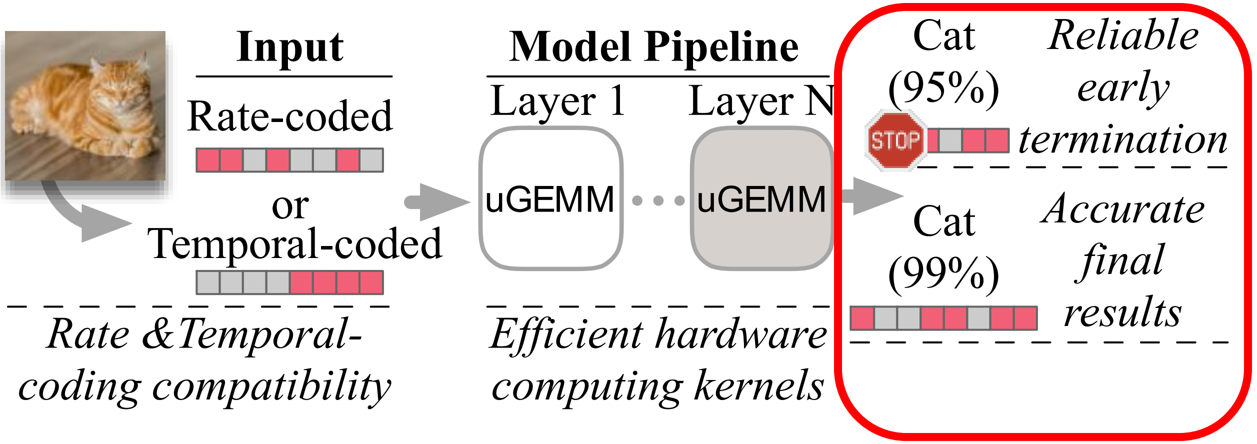
- Unified unary General Matrix Multiplication
  - First to support Add/Mul in temporal coding
  - Compatibility for varying coding and polarity
  - Early termination
    - Fully streaming computation



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Unified uGEMM

- Unified unary General Matrix Multiplication
  - First to support Add/Mul in temporal coding
  - Compatibility for varying coding and polarity
  - Early termination
    - Fully streaming computation
    - High accuracy and stability by solving the correlation problem



Outline

- ❑ Background
- ❑ Motivation
- ❑ **Architecture**
- ❑ Evaluation

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM overview

- Multiplication
  - uMUL
    - Unipolar and Bipolar

- Addition
  - Scaled (uSADD)
    - Unipolar and Bipolar
  - Non-Scaled (uNSADD)
    - Unipolar and Bipolar (first support)

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Multiplication

- uMUL: unipolar
  - Expected function

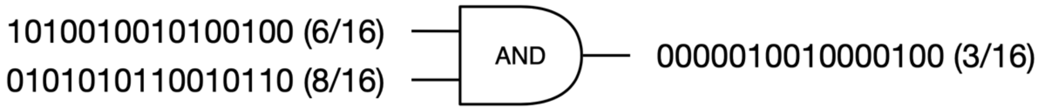
$$V_{out} = V_{in,0} \cdot V_{in,1},$$

$$P(S_{out} = 1) = P(S_{in,0} = 1) \cdot P(S_{in,1} = 1),$$

uGEMM – Multiplication

- uMUL: unipolar
  - Expected function

$$V_{out} = V_{in,0} \cdot V_{in,1},$$
$$P(S_{out} = 1) = P(S_{in,0} = 1) \cdot \boxed{P(S_{in,1} = 1)},$$



  - Actual AND gate function

$$P(S_{out} = 1) = P(S_{in,0} = 1, S_{in,1} = 1)$$
$$= P(S_{in,0} = 1) \cdot \boxed{P(S_{in,1} = 1 | S_{in,0} = 1)}$$

uGEMM – Multiplication

- uMUL: unipolar
  - Expected function

$$V_{out} = V_{in,0} \cdot V_{in,1},$$

$$P(S_{out} = 1) = P(S_{in,0} = 1) \cdot P(S_{in,1} = 1),$$

3/16 6/16 8/16=0.5

  - Actual AND gate function

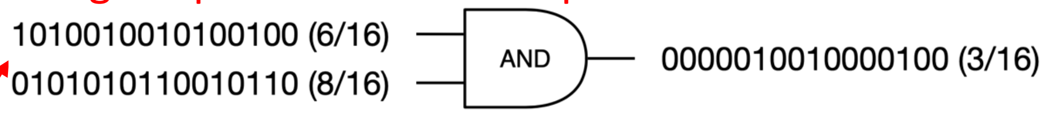
$$P(S_{out} = 1) = P(S_{in,0} = 1, S_{in,1} = 1)$$

$$= P(S_{in,0} = 1) \cdot P(S_{in,1} = 1 | S_{in,0} = 1)$$

3/6 = 0.5

Correct with 0 correlation:

Marginal prob = conditional prob



uGEMM – Multiplication

- uMUL: unipolar
  - Expected function

$$V_{out} = V_{in,0} \cdot V_{in,1},$$

$$P(S_{out} = 1) = P(S_{in,0} = 1) \cdot P(S_{in,1} = 1),$$

3/16 6/16 8/16=0.5

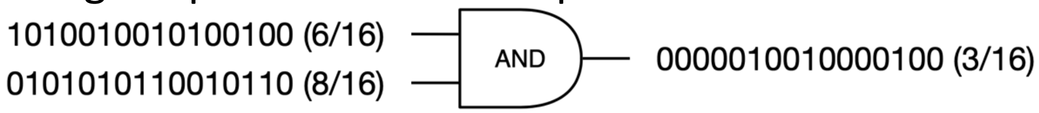
  - Actual AND gate function

$$P(S_{out} = 1) = P(S_{in,0} = 1, S_{in,1} = 1)$$
$$= P(S_{in,0} = 1) \cdot P(S_{in,1} = 1 | S_{in,0} = 1)$$

6/6 = 1.0

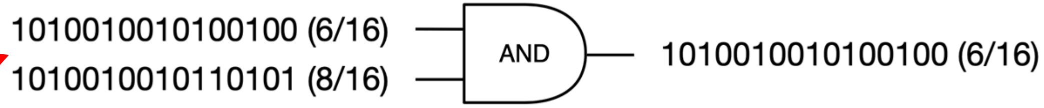
Correct with 0 correlation:

Marginal prob = conditional prob



Wrong with non-0 correlation:

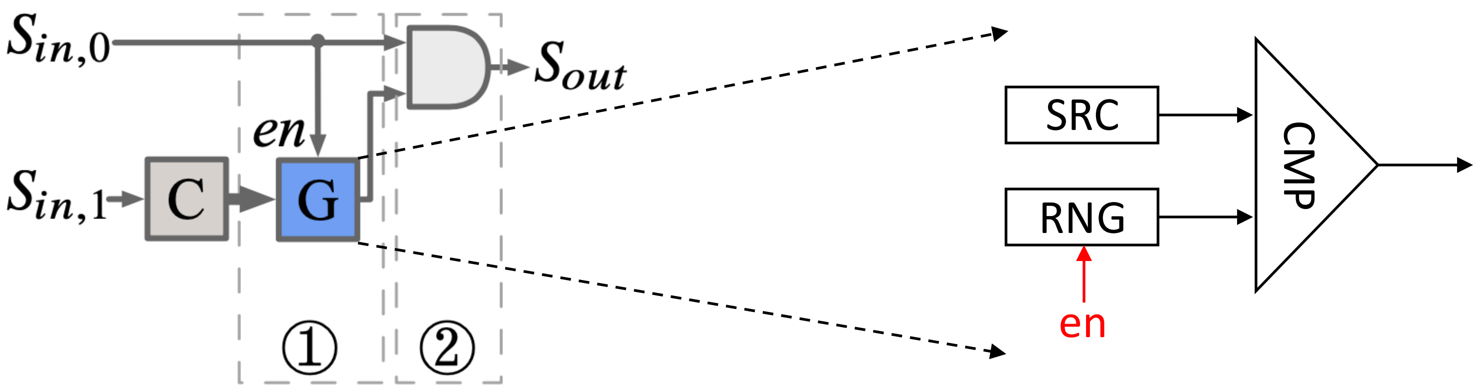
Marginal prob != conditional prob



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Multiplication

- uMUL: unipolar
  - Conditional bit stream generation
    - Enforce $P(S_{in,1}=1) = P(S_{in,1}=1 | S_{in,0}=1)$



C: Counter
G: Bit stream generator

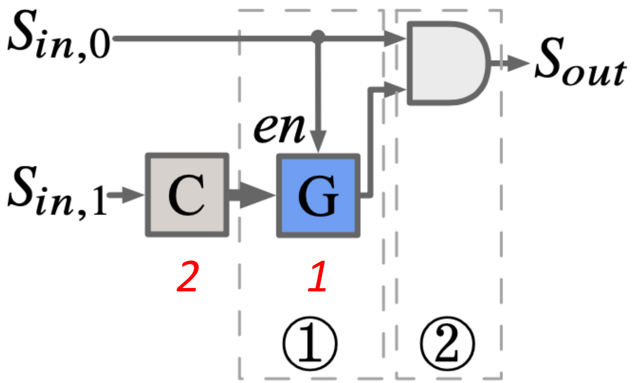
Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Multiplication

- uMUL: unipolar

2/4: 1 0 0 1

$S_{in,1}$ is static in counter. 2/4



0 0 0 1

2/4*2/4=1/4

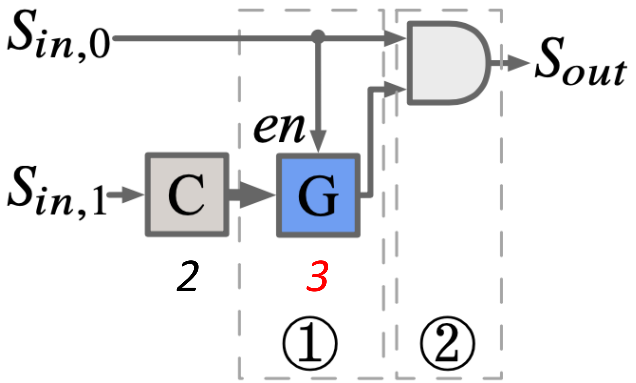
Cycle	Counter (C)	RNG (G)	AND 0 (In 0)	AND 1	Output
1	2 >	1	1	1	1
2					
3					
4					

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Multiplication

➢ uMUL: unipolar

2/4: 1 0 0 1



0 0 0 1

$S_{in,1}$ is static in counter. 2/4

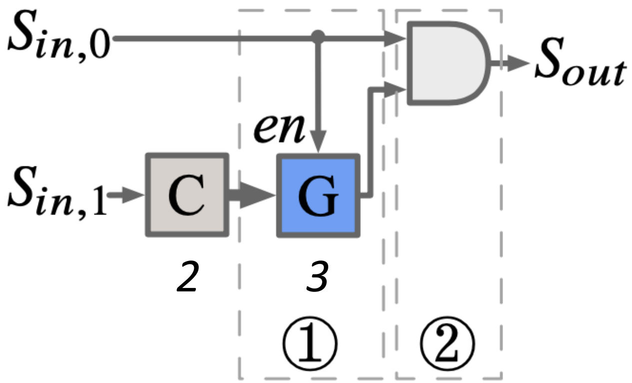
2/4*2/4=1/4

Cycle	Counter (C)	RNG (G)	AND 0 (In 0)	AND 1	Output
1	2	1 en	1	1	1
2	2 <	3	0	0	0
3					
4					

uGEMM – Multiplication

- uMUL: unipolar

2/4: *1 0 0 1*



0 0 0 1

$S_{in,1}$ is static in counter. 2/4

2/4*2/4=1/4

Cycle	Counter (C)	RNG (G)	AND 0 (In 0)	AND 1	Output
1	2	1	1	1	1
2	2	3	0	0	0
3	2 <	3	0	0	0
4					

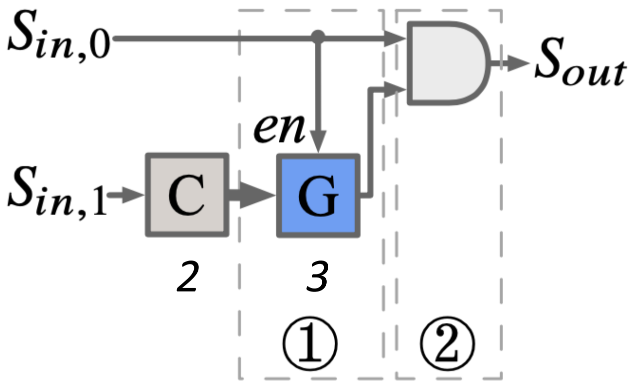
Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Multiplication

- uMUL: unipolar

2/4: 1 0 0 1

$S_{in,1}$ is static in counter. 2/4



0 0 0 1

2/4*2/4=1/4

Cycle	Counter (C)	RNG (G)	AND 0 (In 0)	AND 1	Output
1	2	1	1	1	1
2	2	3	0	0	0
3	2	3	0	0	0
4	2 <	3	1	0	0

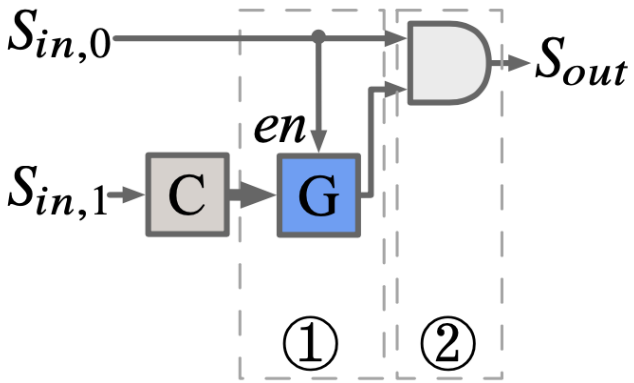
Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Multiplication

- uMUL: unipolar

2/4: *1 0 0 1*

$S_{in,1}$ is static in counter. 2/4



0 0 0 1

2/4*2/4=1/4

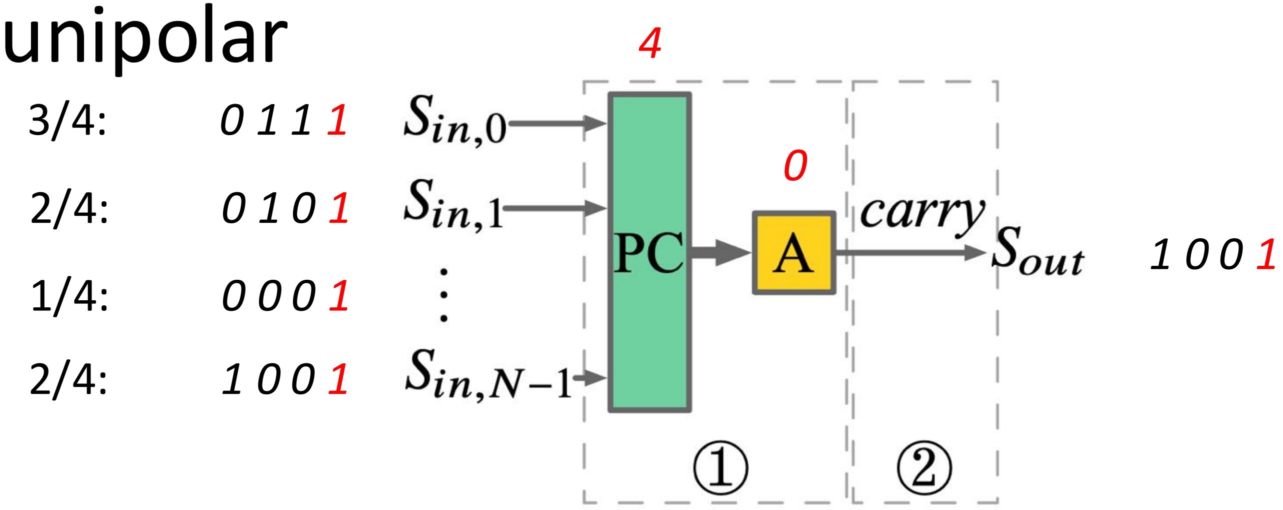
Cycle	Counter (C)	RNG (G)	AND 0 (In 0)	AND 1	Output
1	2	1	1	1	1
2	2	3	0	0	0
3	2	3	0	0	0
4	2	3	1	0	0

1/4

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Scaled addition

- uSADD: unipolar

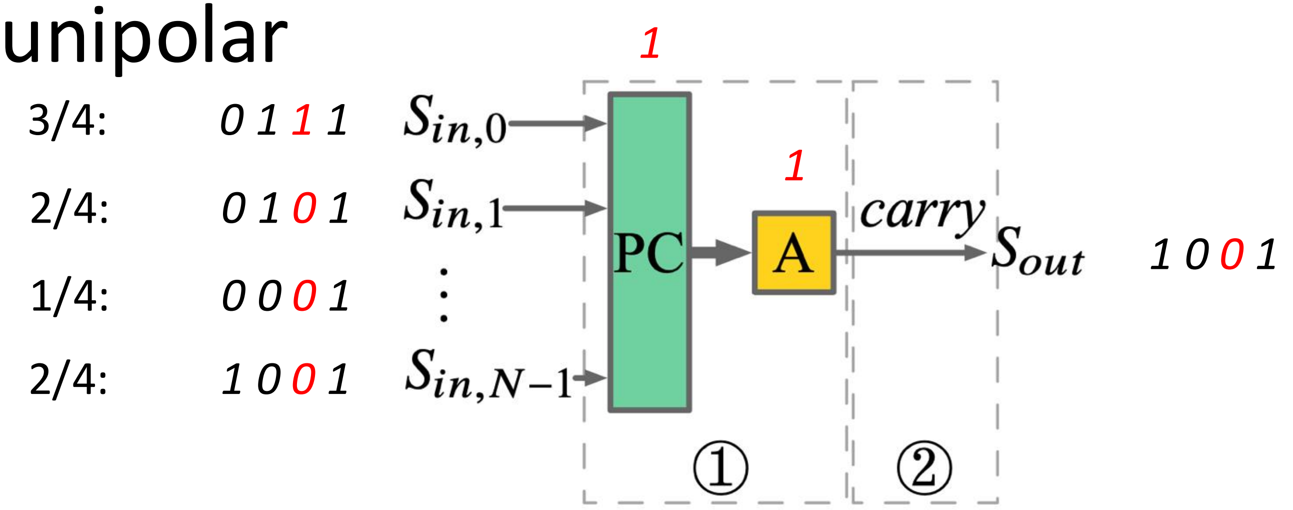


(3/4+2/4+1/4+2/4)/4=2/4

Cycle	Input count (PC)	Accumulator (A)	Carry (output)
1	4	0	1
2			
3			
4			

uGEMM – Scaled addition

- uSADD: unipolar



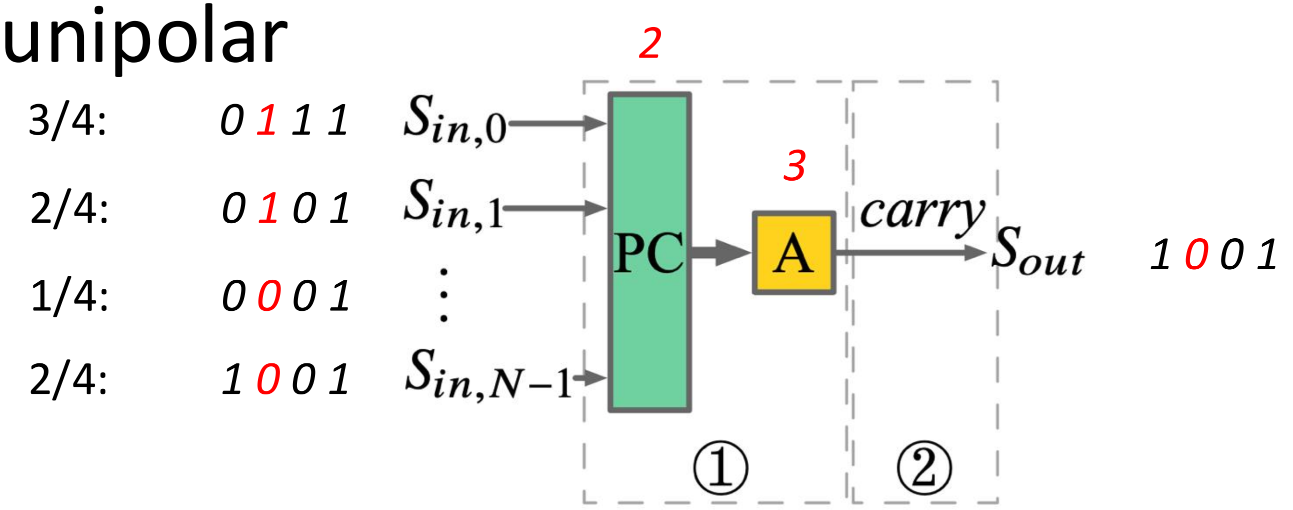
(3/4+2/4+1/4+2/4)/4=2/4

Cycle	Input count (PC)	Accumulator (A)	Carry (output)
1	4	0	1
2	1	1	0
3			
4			

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Scaled addition

- uSADD: unipolar



(3/4+2/4+1/4+2/4)/4=2/4

Cycle	Input count (PC)	Accumulator (A)	Carry (output)
1	4	0	1
2	1	1	0
3	2	3	0
4			

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Scaled addition

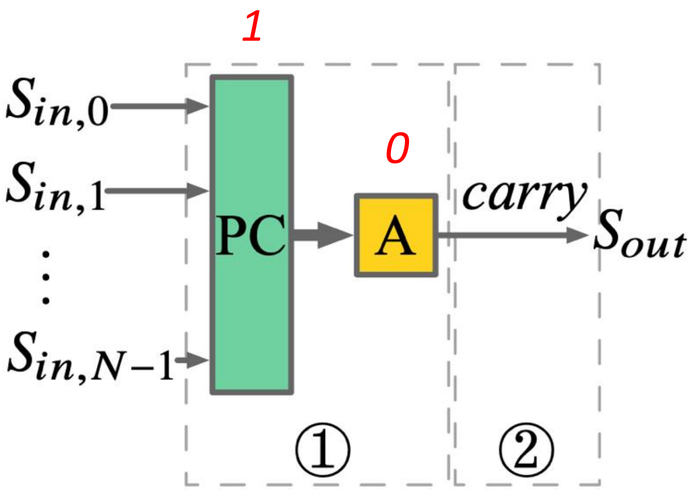
- uSADD: unipolar

3/4: 0 1 1 1

2/4: 0 1 0 1

1/4: 0 0 0 1

2/4: 1 0 0 1



1 0 0 1

(3/4+2/4+1/4+2/4)/4=2/4

Cycle	Input count (PC)	Accumulator (A)	Carry (output)
1	4	0	1
2	1	1	0
3	2	3	0
4	1	0	1

uGEMM – Scaled addition

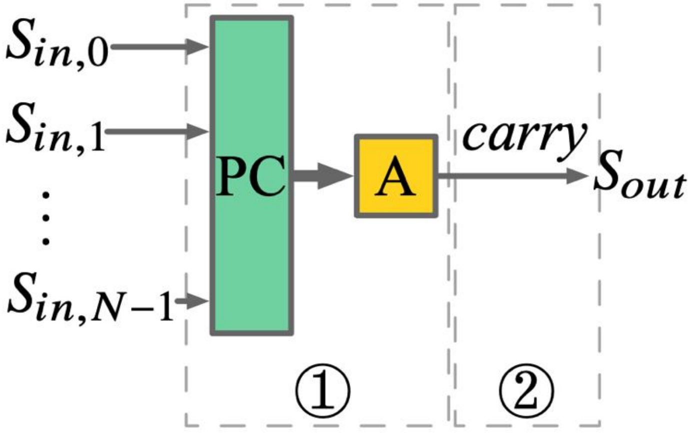
- uSADD: unipolar

3/4: *0 1 1 1*

2/4: *0 1 0 1*

1/4: *0 0 0 1*

2/4: *1 0 0 1*



1 0 0 1

(3/4+2/4+1/4+2/4)/4=2/4

Cycle	Input count (PC)	Accumulator (A)	Carry (output)
1	4	0	1
2	1	1	0
3	2	3	0
4	1	0	1

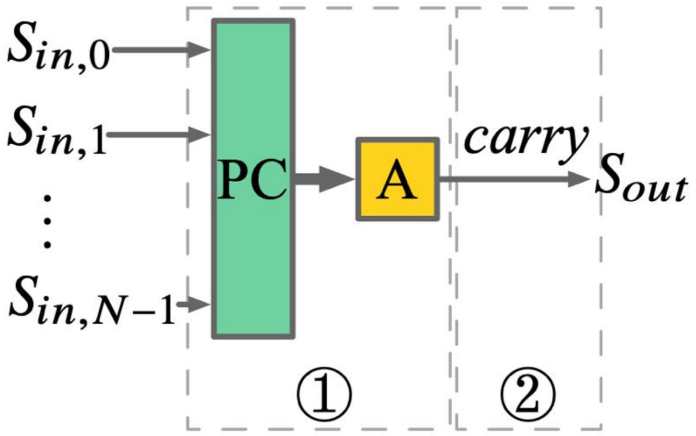
2/4

uGEMM – Scaled addition

- uSADD
  - Expected function

$$V_{out} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} V_{in,n}$$

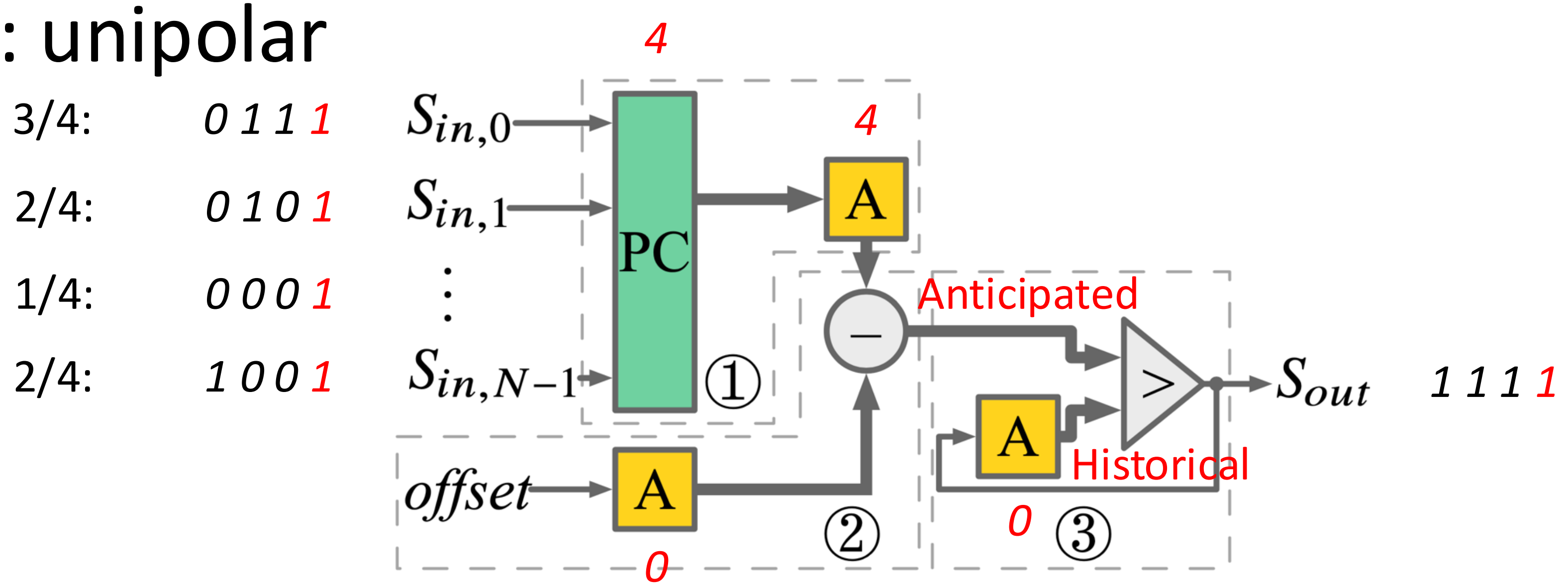
$$\mathbb{C}_{out} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} \mathbb{C}_{in,n}$$



  - Carry overflow mechanism
    - Parallel counter (PC) records current input.
    - When accumulation (A) overflows, output logic 1.

uGEMM – Non-scaled addition

- uNSADD: unipolar



Clip(3/4+2/4+1/4+2/4)=1

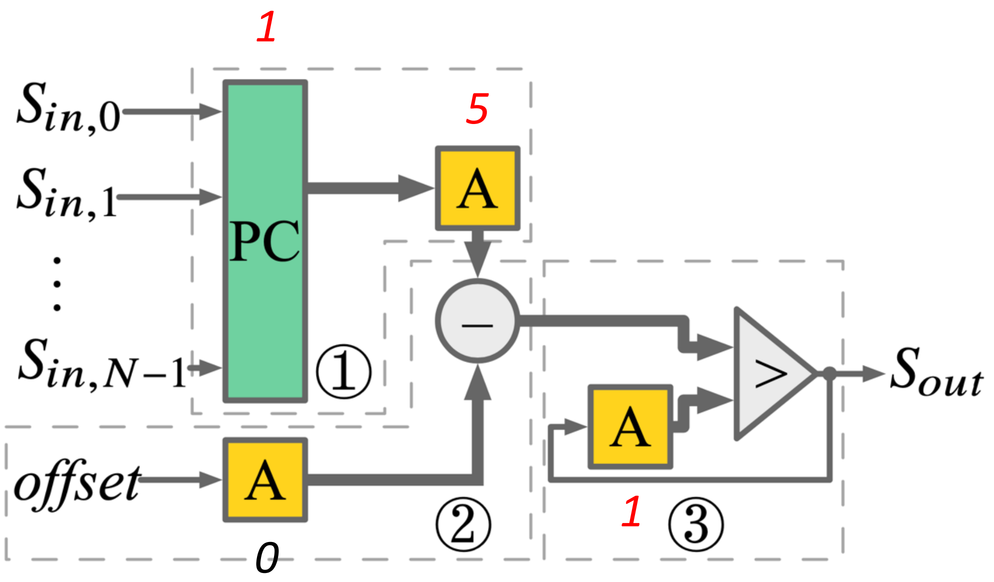
Cycle	Input count (PC)	Acc (1)	Acc (3)	Acc (1>3)	Output
1	4	4	0	True →	1
2					
3					
4					

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Non-scaled addition

- uNSADD: unipolar

3/4:	0 1 1 1
2/4:	0 1 0 1
1/4:	0 0 0 1
2/4:	1 0 0 1



Output: 1 1 1 1

Clip(3/4+2/4+1/4+2/4)=1

Cycle	Input count (PC)	Acc (1)	Acc (3)	Acc (1>3)	Output
1	4	4	0	True	1
2	1	5	1	True	1
3					
4					

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Non-scaled addition

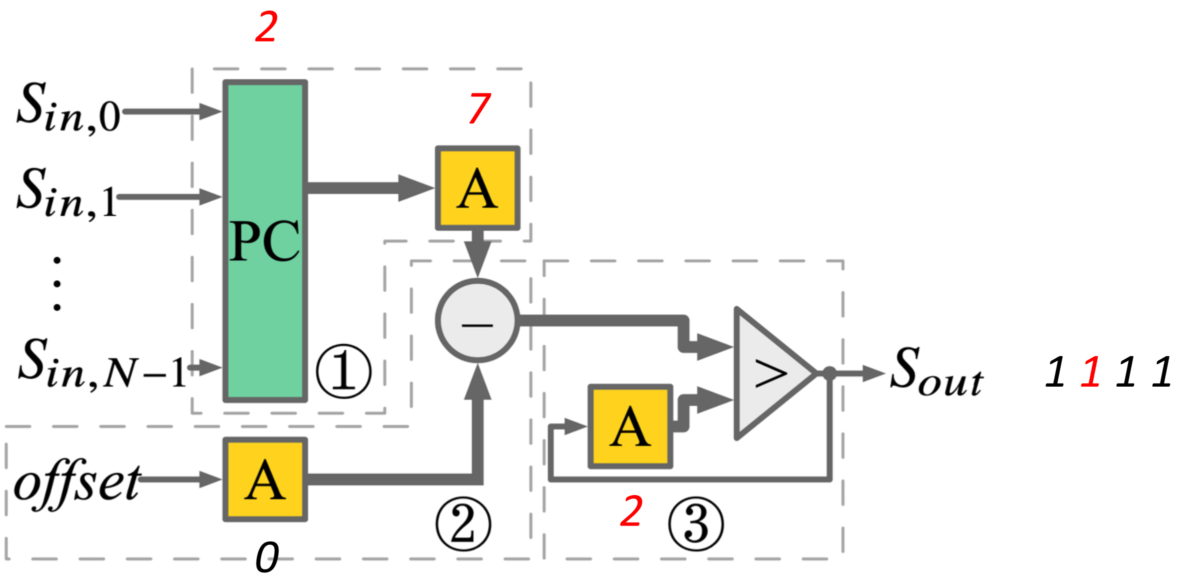
- uNSADD: unipolar

3/4: 0 1 1 1

2/4: 0 1 0 1

1/4: 0 0 0 1

2/4: 1 0 0 1

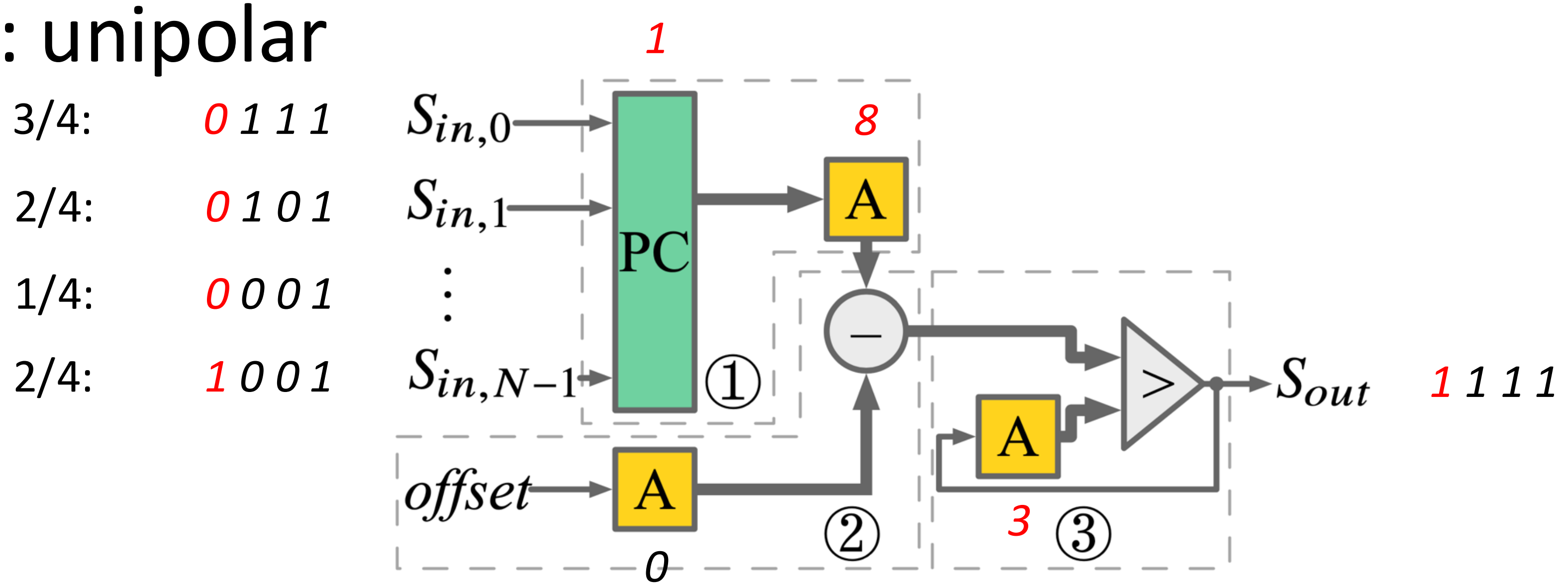


Clip(3/4+2/4+1/4+2/4)=1

Cycle	Input count (PC)	Acc (1)	Acc (3)	Acc (1>3)	Output
1	4	4	0	True	1
2	1	5	1	True	1
3	2	7	2	True	1
4					

uGEMM – Non-scaled addition

- uNSADD: unipolar



Clip(3/4+2/4+1/4+2/4)=1

Cycle	Input count (PC)	Acc (1)	Acc (3)	Acc (1>3)	Output
1	4	4	0	True	1
2	1	5	1	True	1
3	2	7	2	True	1
4	1	8	3	True	1

uGEMM – Non-scaled addition

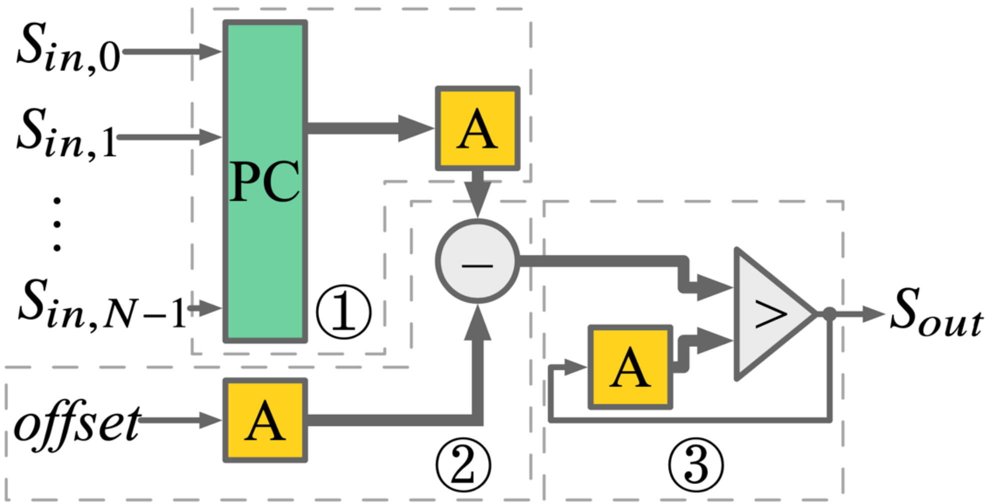
- uNSADD: unipolar

3/4: *0 1 1 1*

2/4: *0 1 0 1*

1/4: *0 0 0 1*

2/4: *1 0 0 1*



1 1 1 1

Clip(3/4+2/4+1/4+2/4)=1

Cycle	Input count (PC)	Acc (1)	Acc (3)	Acc (1>3)	Output
1	4	4	0	True	1
2	1	5	1	True	1
3	2	7	2	True	1
4	1	8	3	True	1

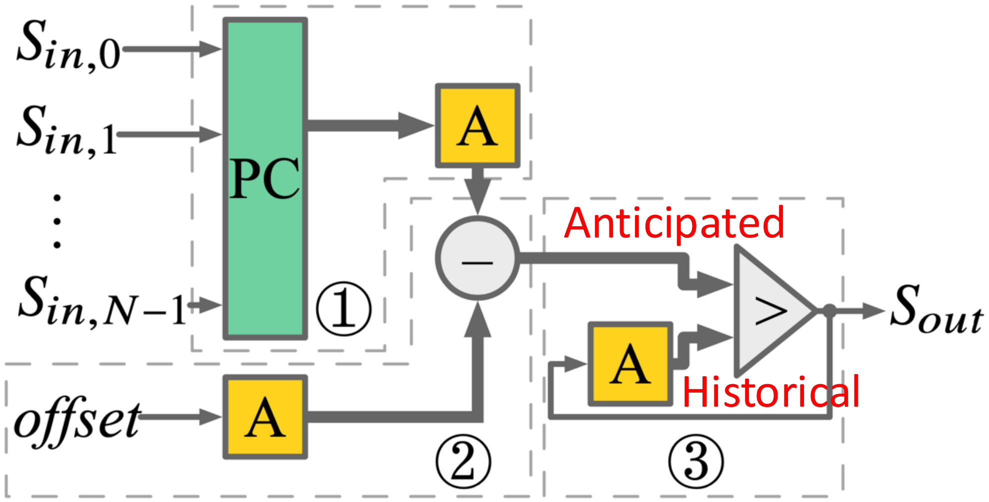
4/4

uGEMM – Non-scaled addition

- uNSADD
  - Expected function

$$V_{out} = \mathrm{clip}\left(\sum_{n=0}^{N-1} V_{in,n}, -k, 1\right)$$

$$\mathbb{C}_{out} = \min\left(\sum_{n=0}^{N-1} \mathbb{C}_{in,n} - L \cdot \mathbb{C}_{offset}, L\right)$$

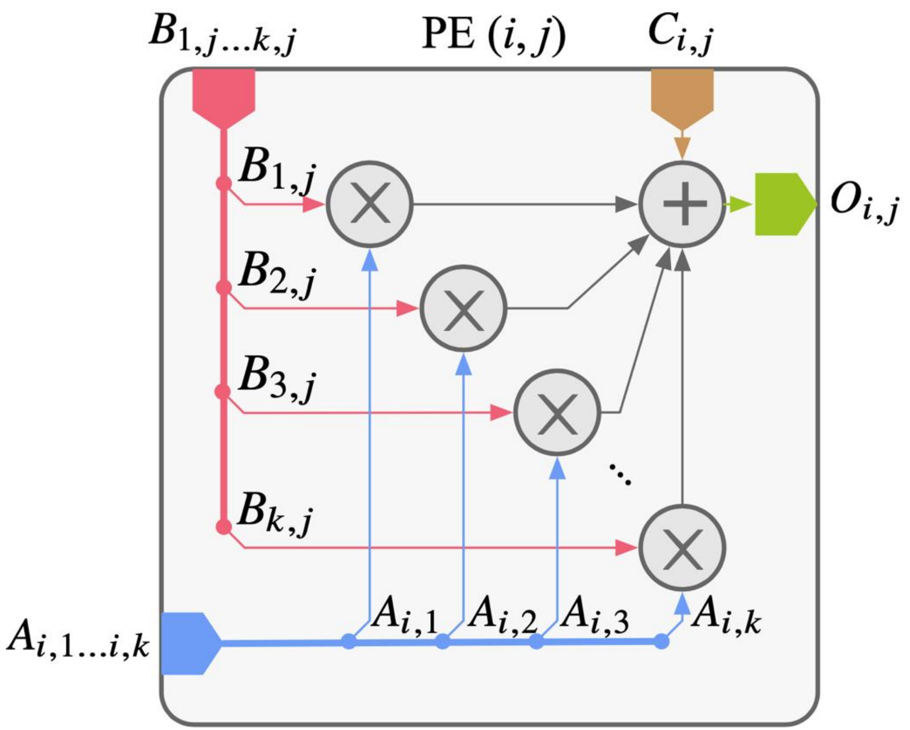


  - Output tracking mechanism
    - Difference between anticipated and historical output 1s indicates what to output.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Processing element

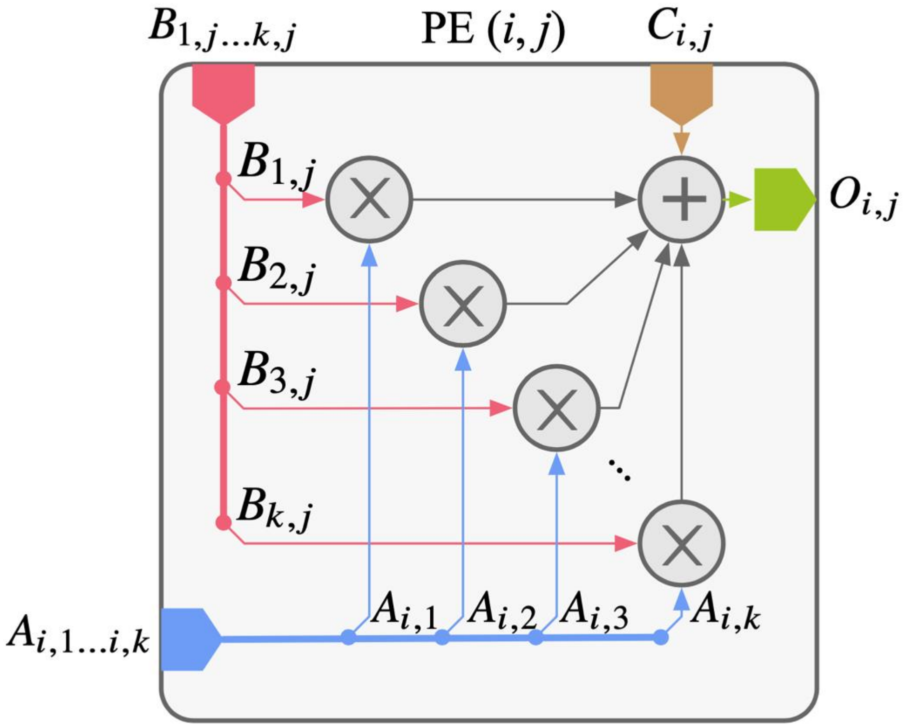
- Input insensitivity
  - Multiplication
    - Generating operand conditionally
    - Ignoring correlation



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – Processing element

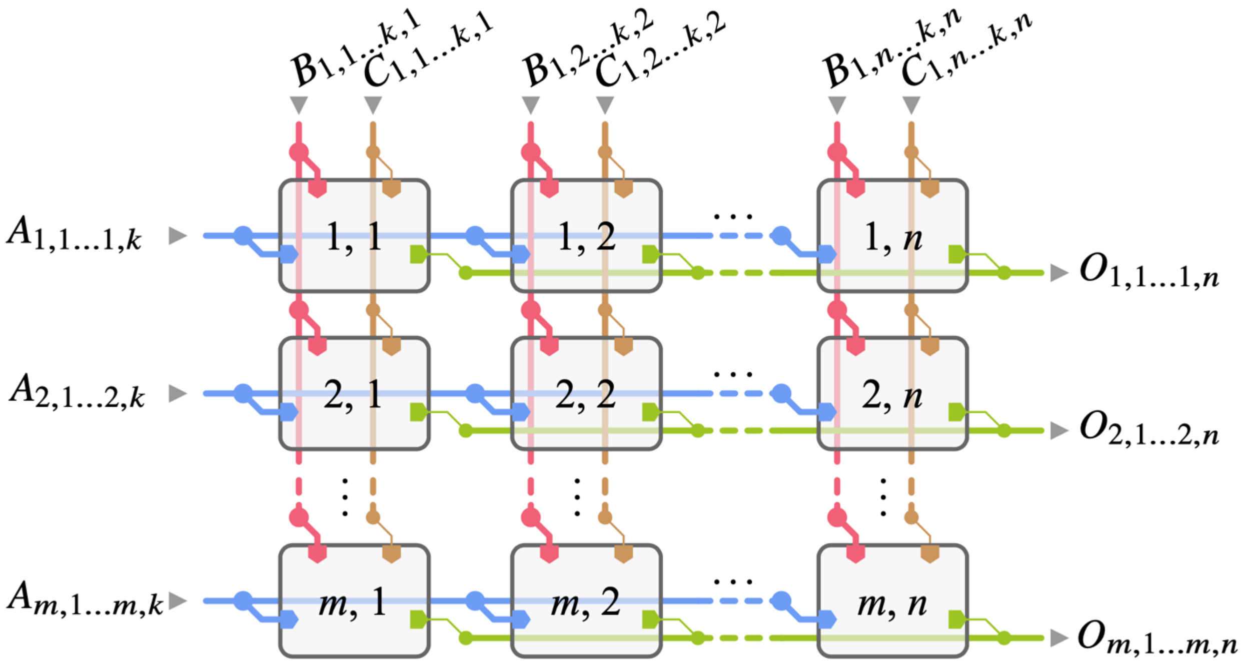
- Input insensitivity
  - Multiplication
    - Generating operand conditionally
    - Ignoring correlation
  - Addition
    - Caring about bit count
    - Ignoring bit distribution



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM – PE array

- High parallelism via broadcasting
- Input insensitivity due to PE
- Reliable early termination
  - Fully streaming computation
  - High accuracy and stability



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

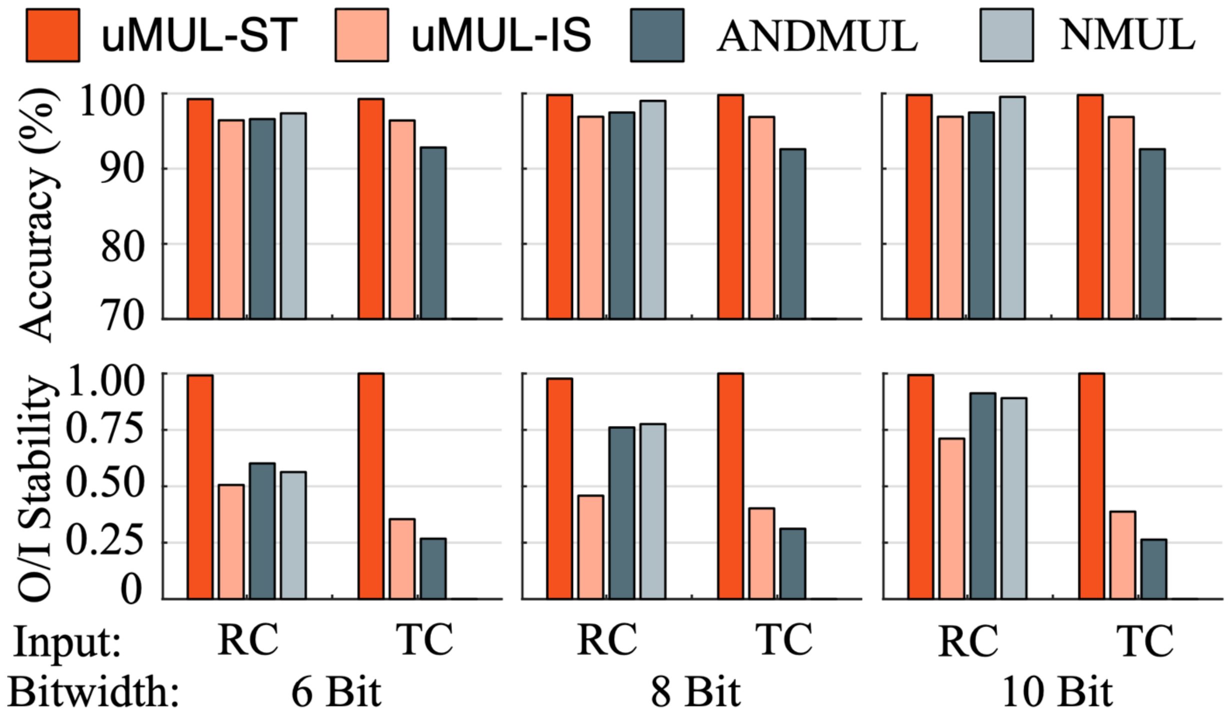
Outline

- ❑ Background
- ❑ Motivation
- ❑ Architecture
- ❑ **Evaluation**

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uMUL performance: Unipolar

- ➢ Accurate final result
  - Static input (ST)
  - In-stream input (IS)
- ➢ Faster stabilization
  - Output/Input stability
- ➢ Input insensitivity
  - Rate coding (RC)
  - Temporal coding (TC)

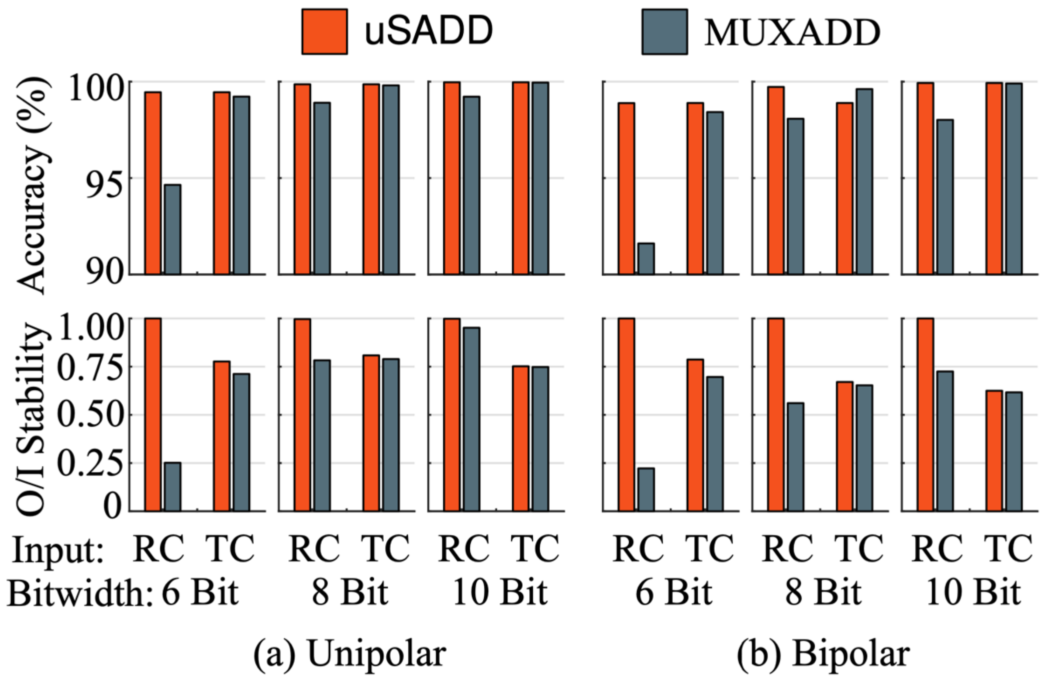


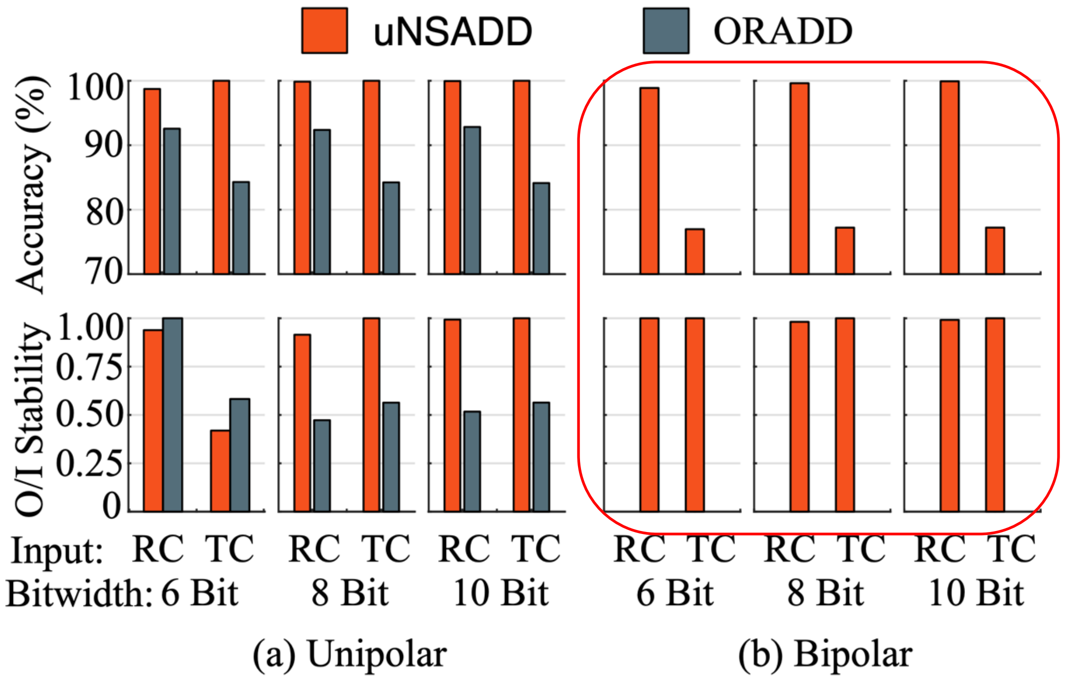
Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

ADD performance: uSADD and uNSADD

- ➢ Accurate final result
- ➢ Faster stabilization
- ➢ Input insensitivity

First to support bipolar non-scaled addition

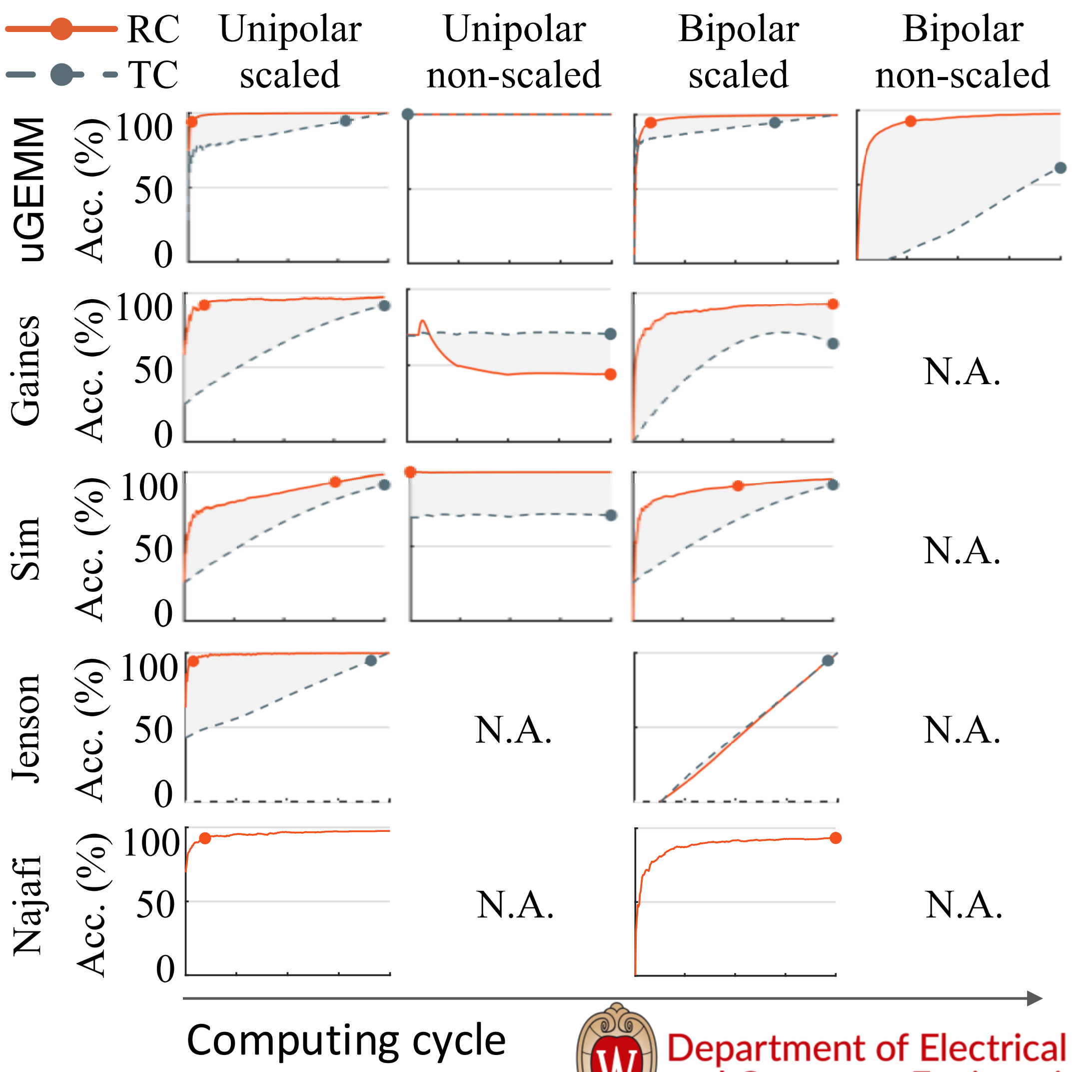




Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM performance

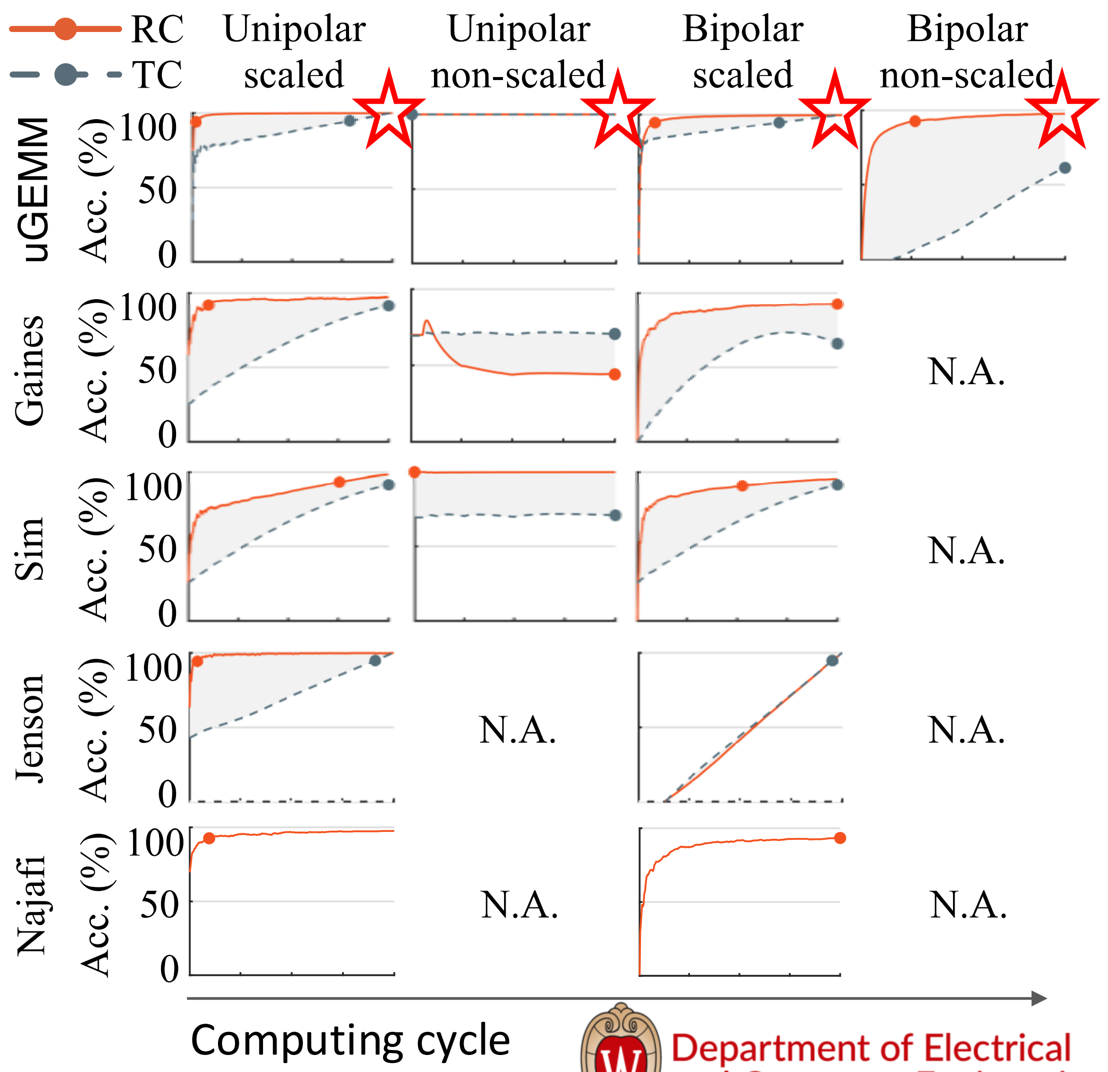
- Accurate final result
  - Cross out all configurations



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM performance

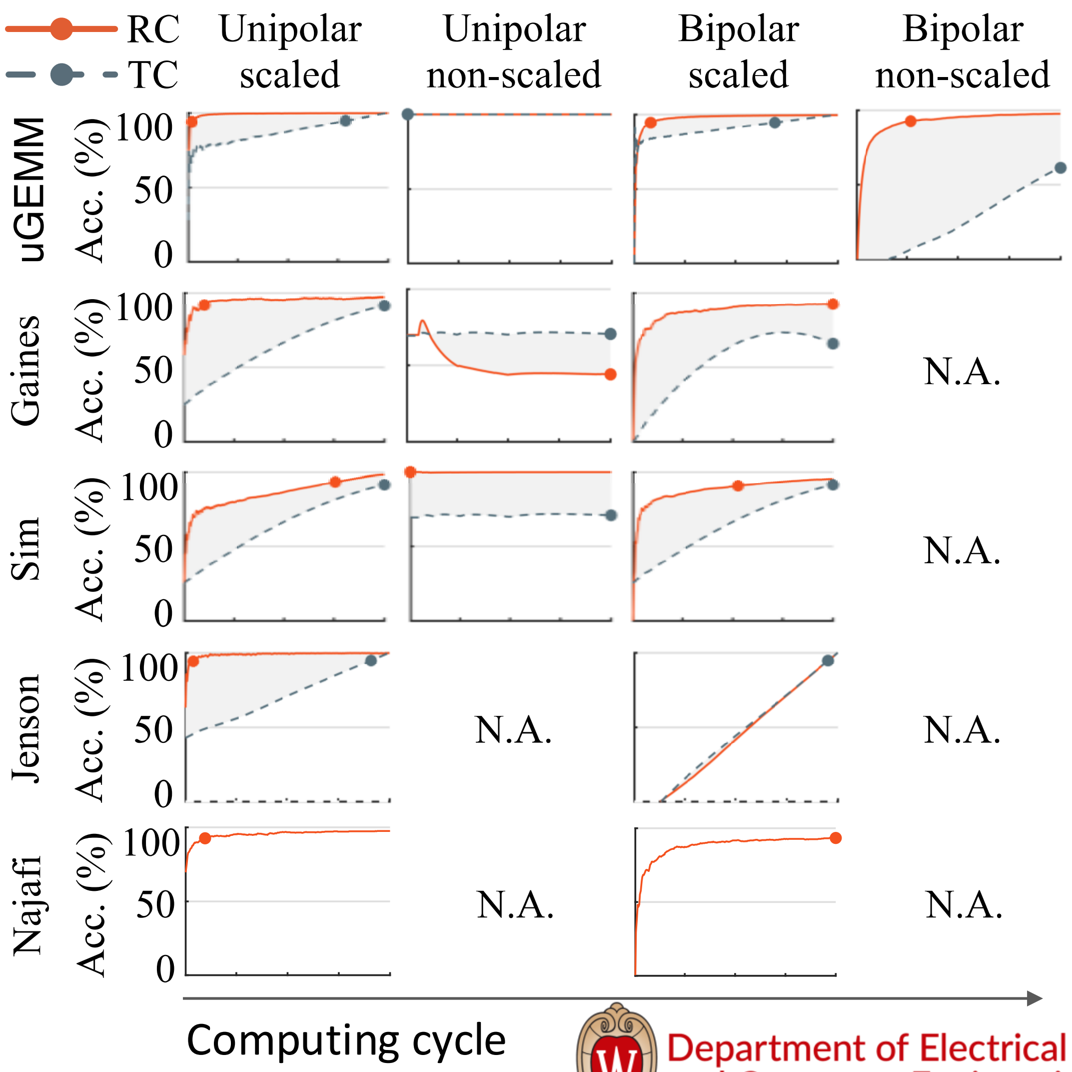
- Accurate final result
  - Cross out all configurations



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM performance

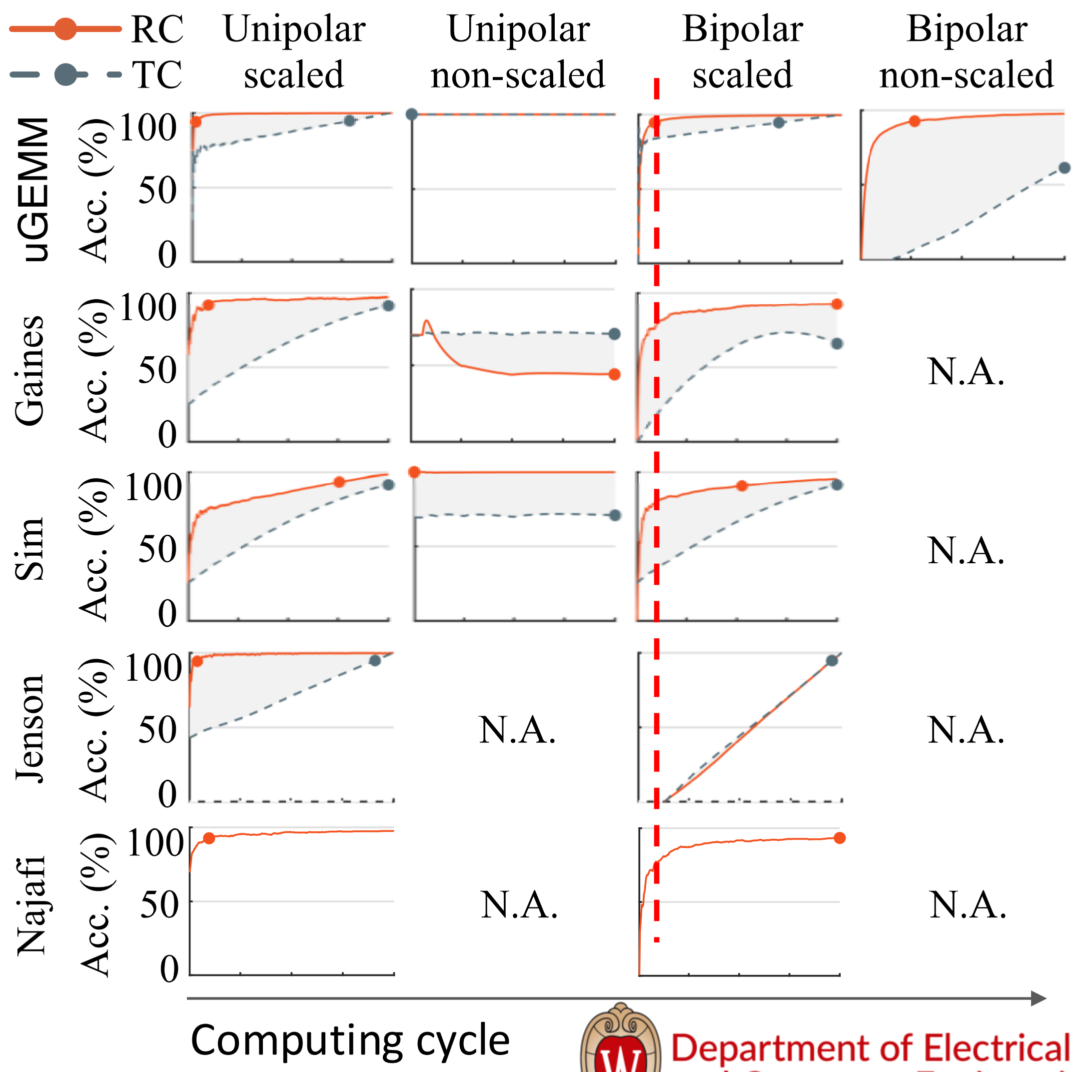
- Accurate final result
    - Cross out all configurations
- Reliable early termination
    - Earlier stable point



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM performance

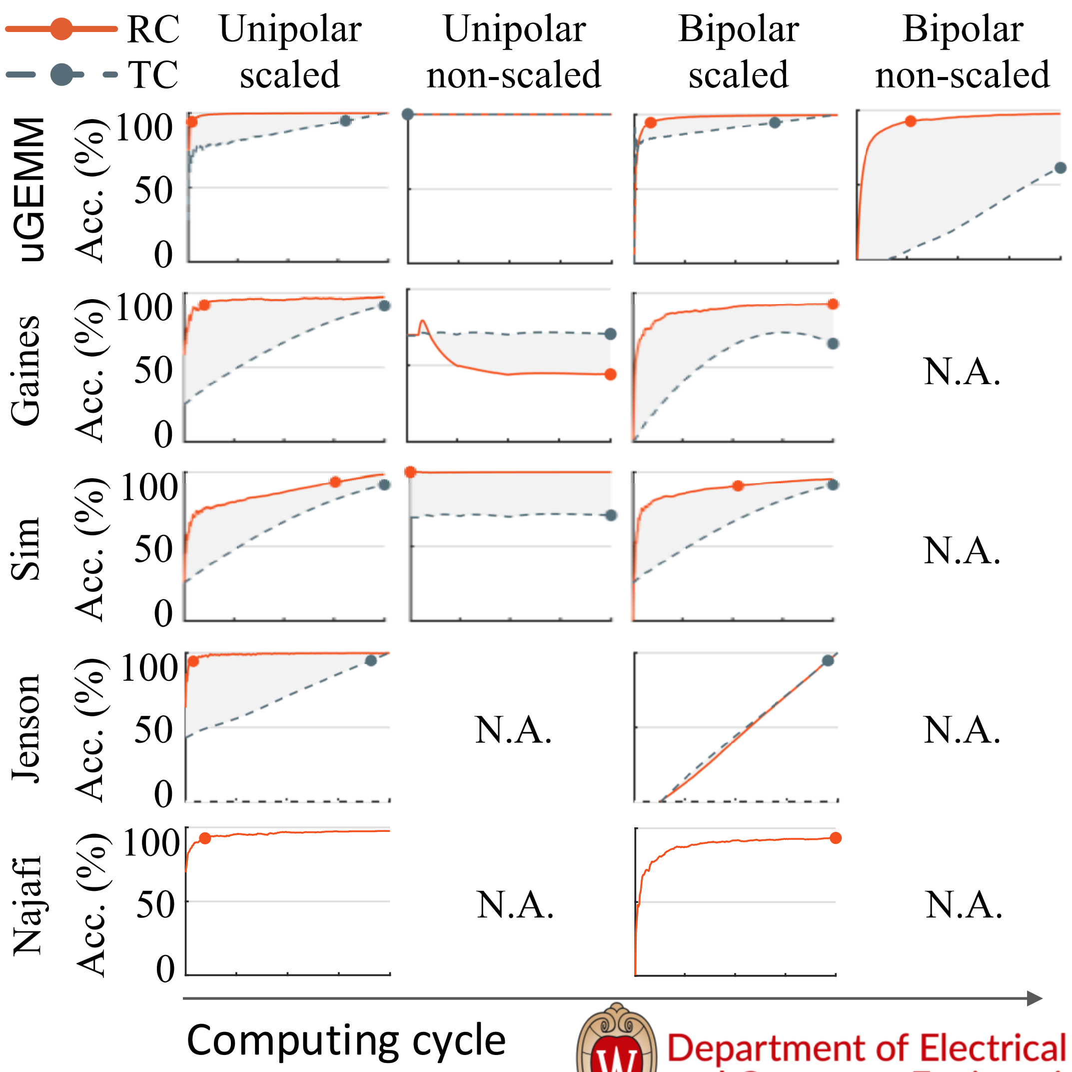
- Accurate final result
  - Cross out all configurations
- Reliable early termination
  - Earlier stable point



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM performance

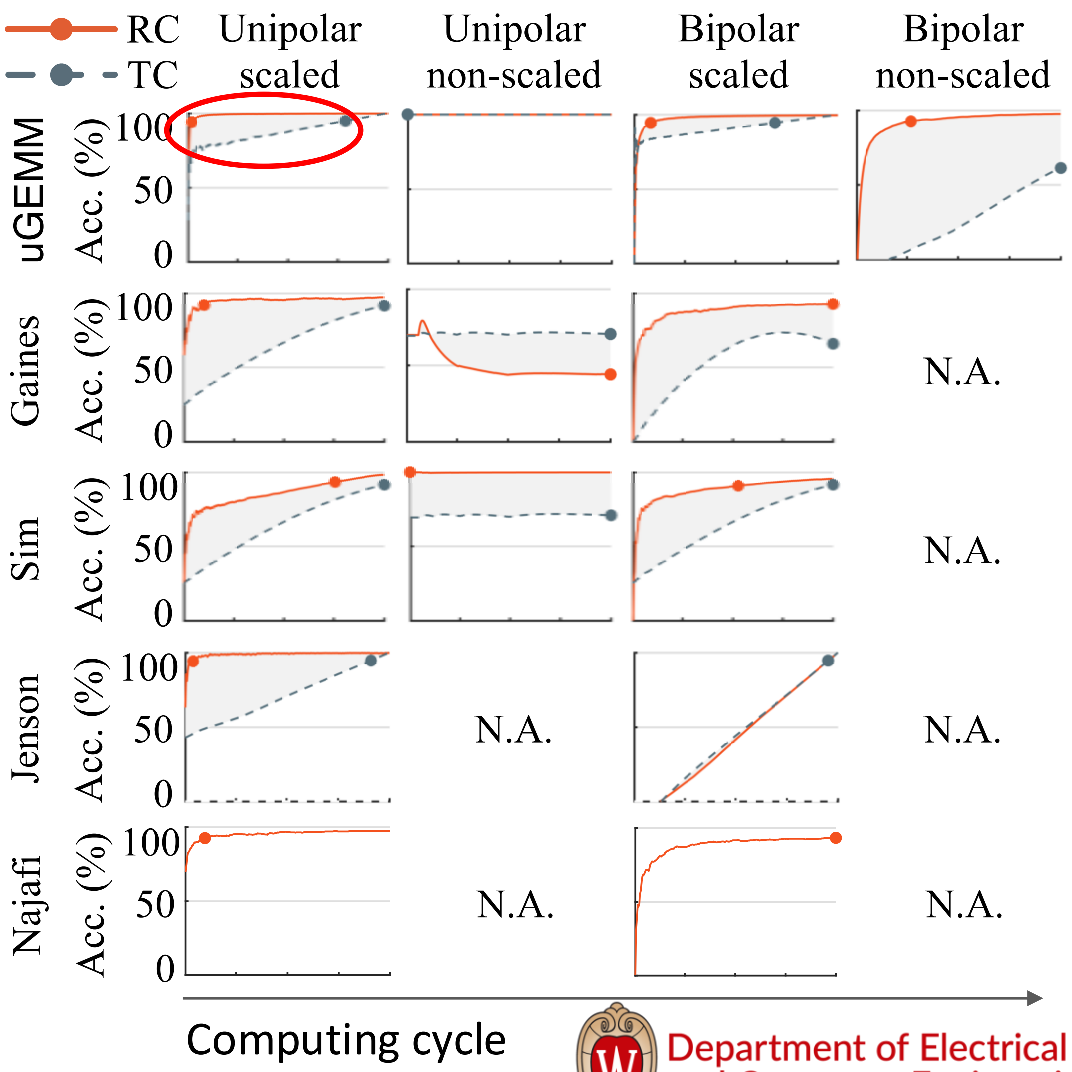
- Accurate final result
  - Cross out all configurations
- Reliable early termination
  - Earlier stable point
- Input insensitivity
  - Minimal RC/TC difference



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

uGEMM performance

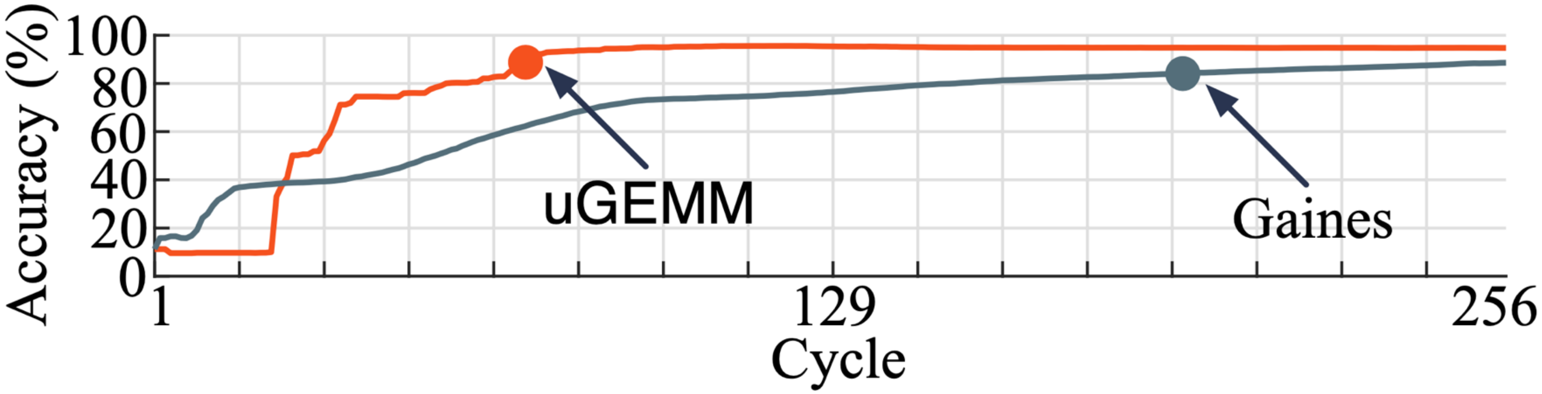
- Accurate final result
  - Cross out all configurations
- Reliable early termination
  - Earlier stable point
- Input insensitivity
  - Minimal RC/TC difference



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

MLP performance

- Final accuracy
  - FP-32bit : 96.87%
  - FXP-8bit: 96.08%
  - uGEMM: 94.7%
  - Gaines: 88.58%
- Cycle count for 95% final accuracy
  - uGEMM: 71
  - Gaines: 195



uGEMM Energy efficiency

- Compared to other unary designs
  - Higher energy efficiency, though slightly higher area
- Compared to binary bit-parallel designs
  - Comparable for matrix multiplication
  - 10X higher for matrix convolution
- Compared to binary bit-serial designs
  - 5X higher for matrix convolution

Thank you!
Q & A