A 28nm 83.23TFLOPS/W POSIT-Based Compute-in-Memory Macro for High-Accuracy AI Applications

Yang Wang[1], Xiaolong Yang[1], **Yubin Qin[1]**, Zhiren Zhao[1],
Ruiqi Guo[1], Zhiheng Yue[1], Huiming Han[1], Shaojun Wei[1], Yang Hu[1], Shouyi Yin[1,2]

[1]Tsinghua University, Beijing, China
[2]Shanghai AI Laboratory, Shanghai, China

Outline

- Background and Motivation
- Challenges of POSIT-Based CIM Macro
- Proposed POSIT@CIM Macro Features
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- Measurement and Comparison
- Conclusion

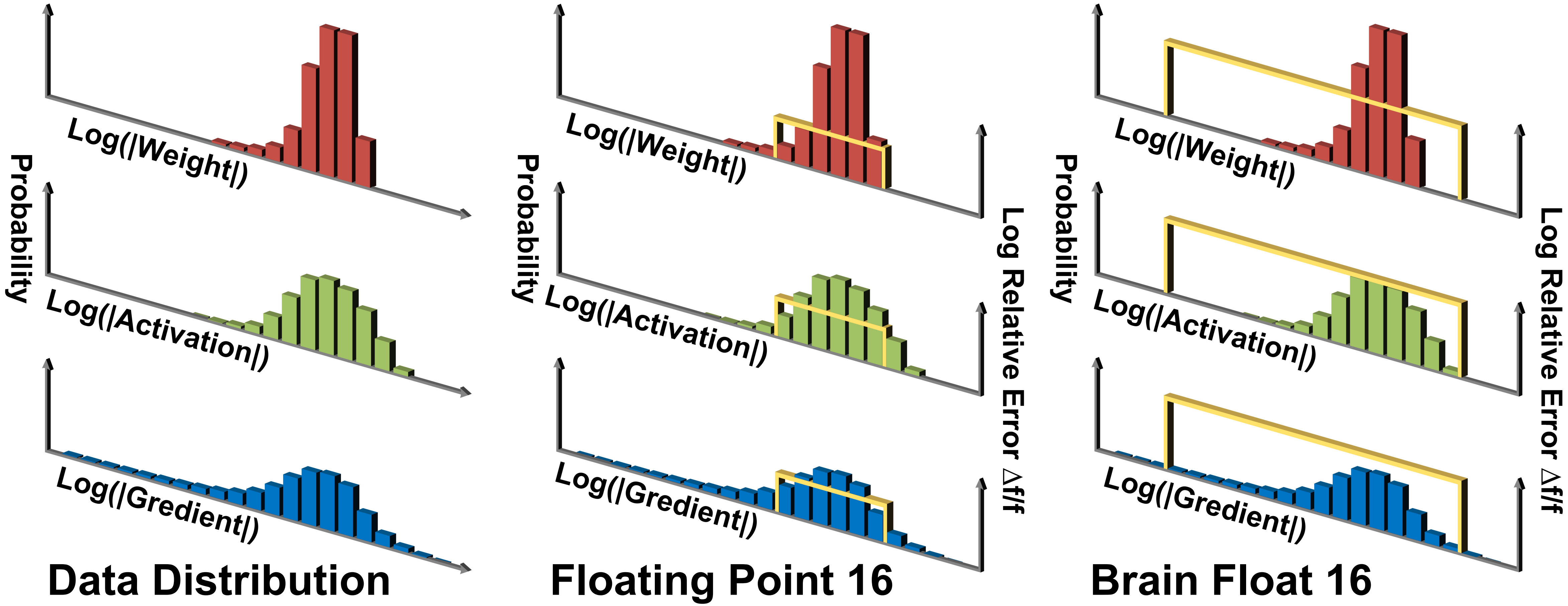
FP-CIM for High-accuracy AI Applications

Keyword Spotting → Face Recogn. → AI Assistant → Automatic Drive → AIGC

RNN CNN Transformer BEV Diffusion Model

- Recent AI tasks are becoming **increasingly complex**.
- Complex AI application requires **FP-CIM for high accuracy**.

Limitation of Conventional FP Data Format



- Conventional FP cannot achieve **high accuracy with low power**.

Principle of POSIT Data Format

Posit Data Format

S	Regime*(R)*	Exponent*(E)*	Mantissa*(M)*
	(r bits) Dynamic	*(es bits)* Pre-set	*(n-r-es-1 bits)* Dynamic

$$\text{value} = (-1)^{\text{sign}} \times (2^K)^R \times 2^E \times 1.M,\ K=2^{es}$$

Regime(R): unary (thermometer) code with successive 0s and 1s

Regime*(R)* $\begin{cases} \text{1111X} & \text{positive R,1's count - 1} \\ \text{0000X} & \text{negative R,0's count} \end{cases}$

Example: POSIT(8,2), where es=2

1	1110	01	1

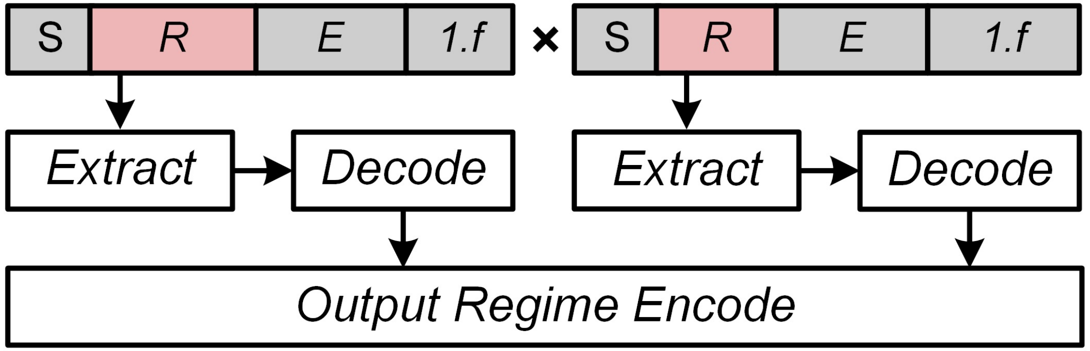
$K=2^{es}=2^2=4$

R=Regime(1110)=(3-1)=2

E=Exponent(10)=1

M=Mantissa(1.1)=1.5

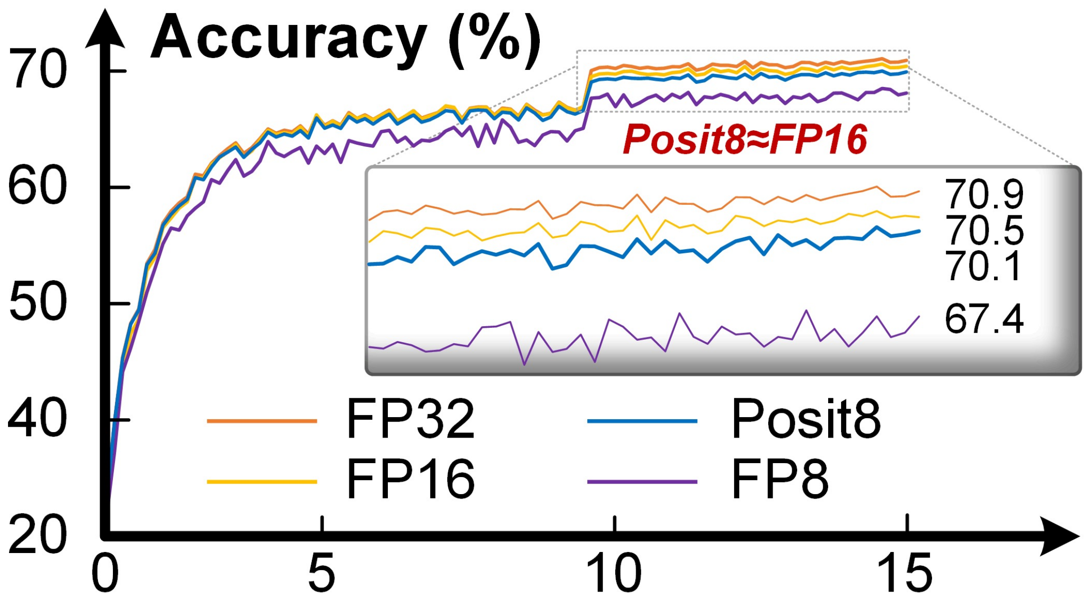
$$\text{value} = -1 \times (2^4)^2 \times 2^1 \times 1.5 = -1.5 \times 2^9$$



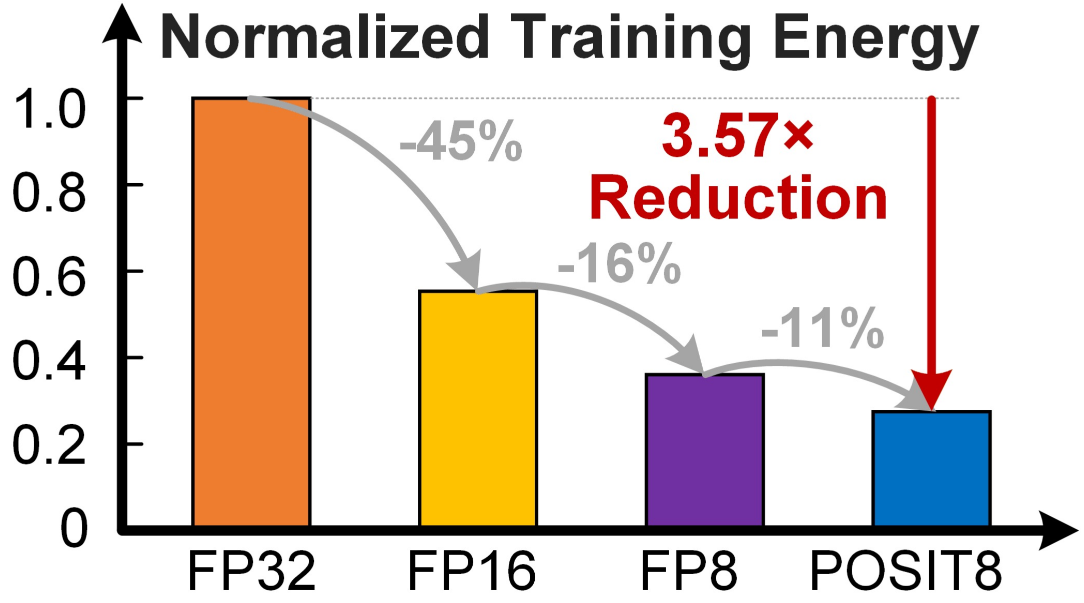
■ **POSIT exploits dynamic bit to adapts to varied distributions.**

Conventional FP VS. POSIT

Training with Different Data Formats
(ImageNet on ResNet18)



Training Energy Comparison
(Achieving Same Accuracy)

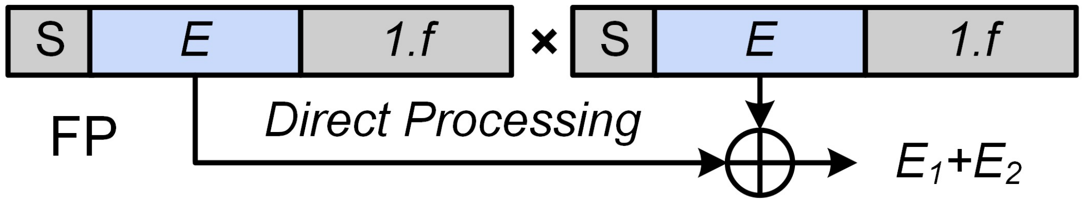


- POSIT8 saves **27% energy** with **0.4% accuracy loss** than FP16.

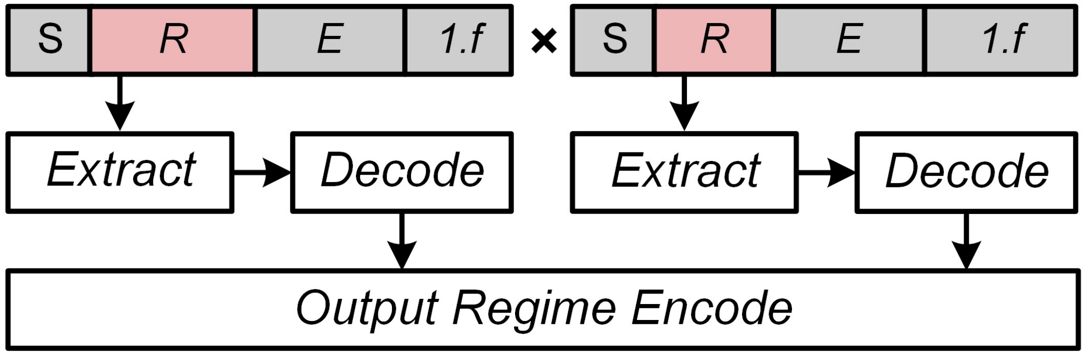
Outline

- Background and Motivation
- **Challenges of POSIT-Based CIM Macro**
- Proposed POSIT@CIM Macro Features
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- Measurement and Comparison
- Conclusion

Challenge 1: Large Power in Regime Processing

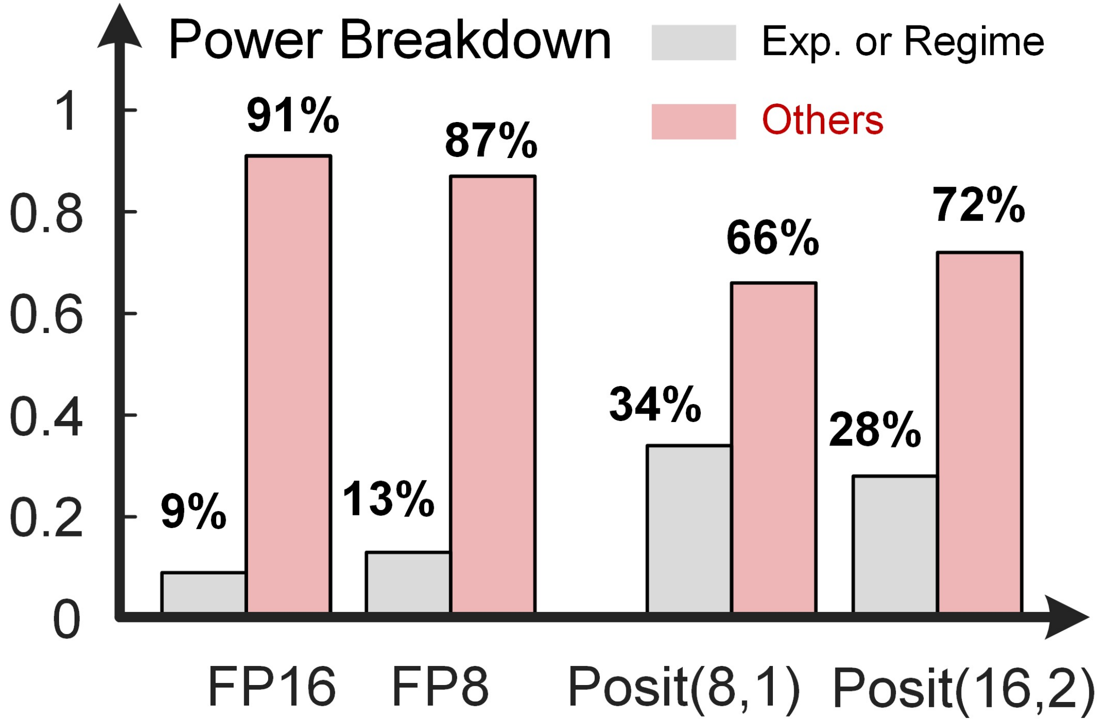


Exponent Processing of FP Data



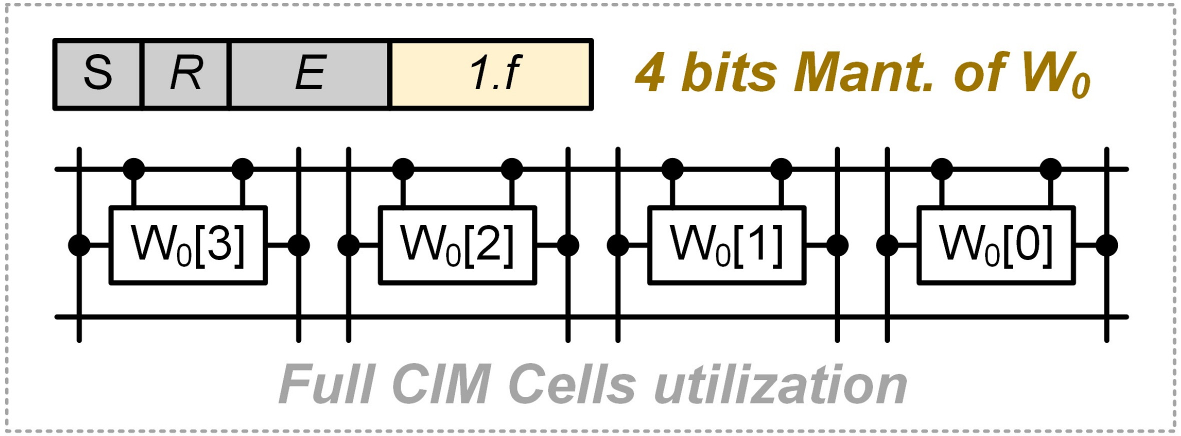
Regime Processing of FP Data

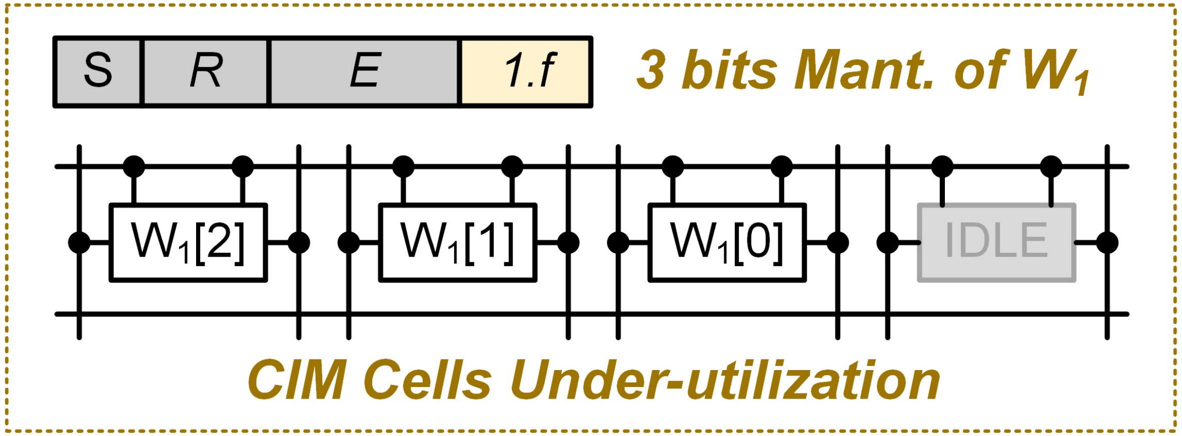
Power Breakdown of FP and POSIT

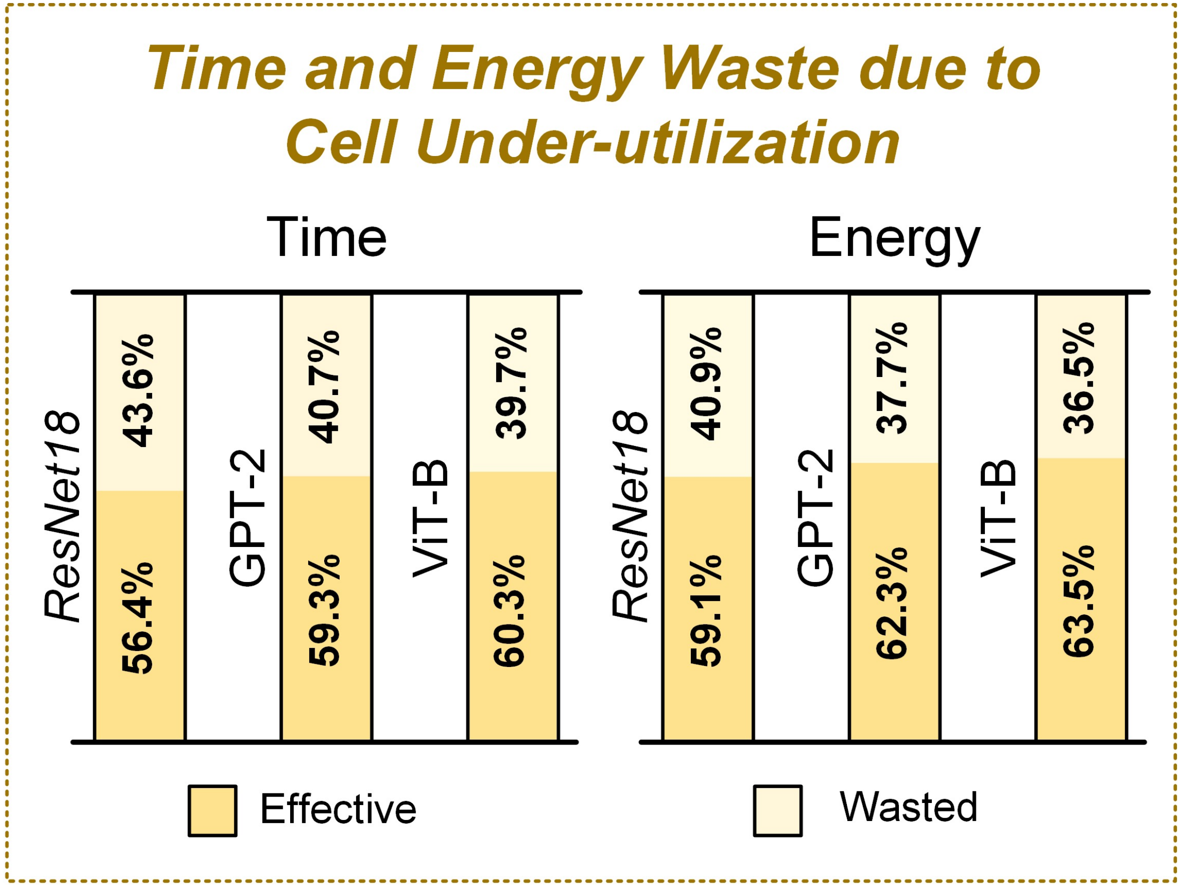


- **Dynamic regime increases 2.62× pre-processing energy.**

Challenge 2: Cell Under-utilization in CIM Array



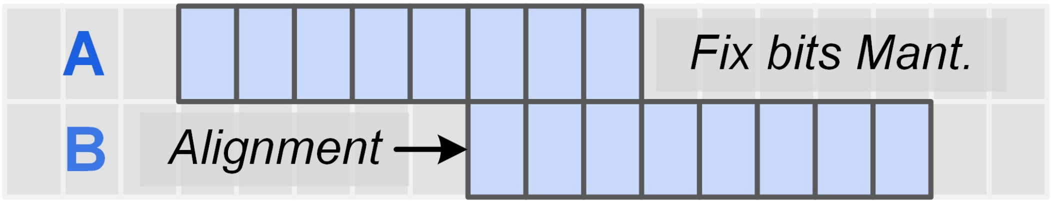




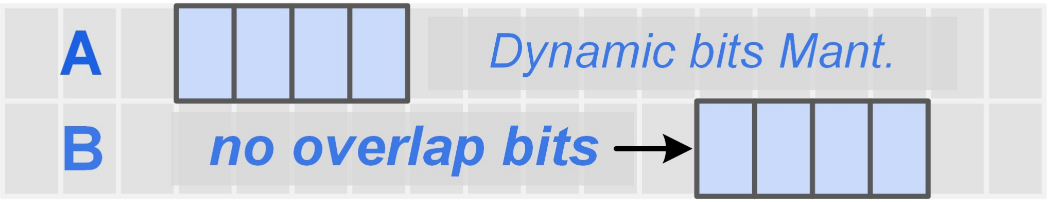
- Dynamic mantissa introduces **41.3% CIM cell underutilization**.

Challenge 3: Redundant Toggle in Adder Tree

FP: Dynamic Alig. + Fixed Bit-width



Posit: Dynamic Alig. + Dynamic Bits



A + B = A | B

Redundant Logic Toggle Power Consumption in Adder Tree

174.38nW **A + B = A | B** **29.75nW**

(16b Simulation@400HMz,0.9V)

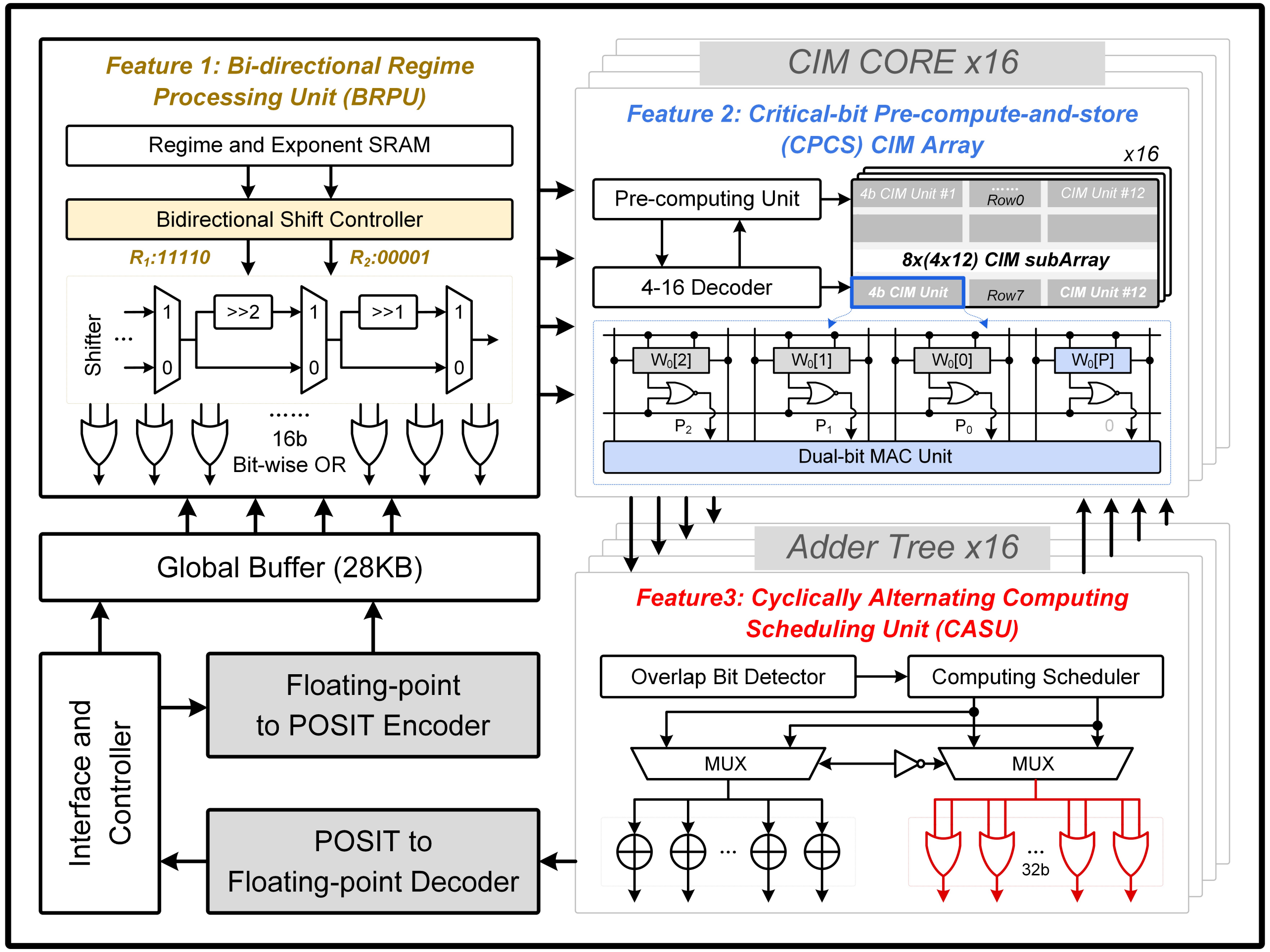
Models	Adder Tree Energy		**Ratio**
	Total	Redun.	
ResNet18	0.11mJ	0.083mJ	***76.3%***
GPT-2	5.8mJ	3.1mJ	***53.5%***
ViT-B	1.1mJ	0.63mJ	***57.6%***

- **Dynamic aligned accumulation incurs 66.8% power waste.**

Outline

- Background and Motivation
- Challenges of POSIT-Based CIM Macro
- **Proposed POSIT@CIM Macro Features**
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- Measurement and Comparison
- Conclusion

Overall Architecture of POSIT CIM Macro



- **BRPU** replaces regime codec with Shift and OR logic to save regime pre-processing energy.
- **CPCS CIM Array** exploits spare bits to perform dual-bit MAC to increase CIM utilization.
- **CASU** simplifies addition logic to bit-wise OR operations to reduce accumulation power.

Outline

- Background and Motivation
- Challenges of POSIT-Based CIM Macro
- **Proposed POSIT@CIM Macro Features**
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- Measurement and Comparison
- Conclusion

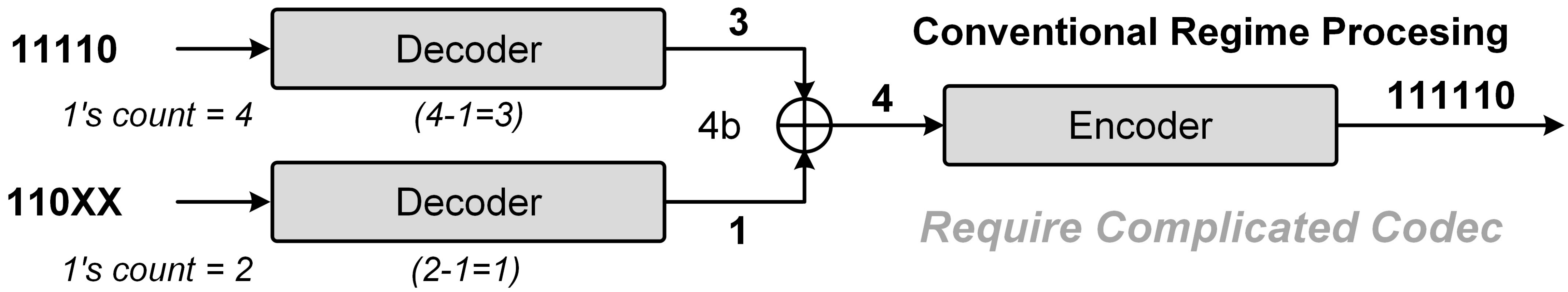
Feature 1: Bi-directional Regime Processing

Binary	00001	**0001x**	001xx	……	110xx	**1110x**	11110
Regime	-4	**-3**	-2	……	1	**2**	3

0's count for neg. R (0001 for -3)

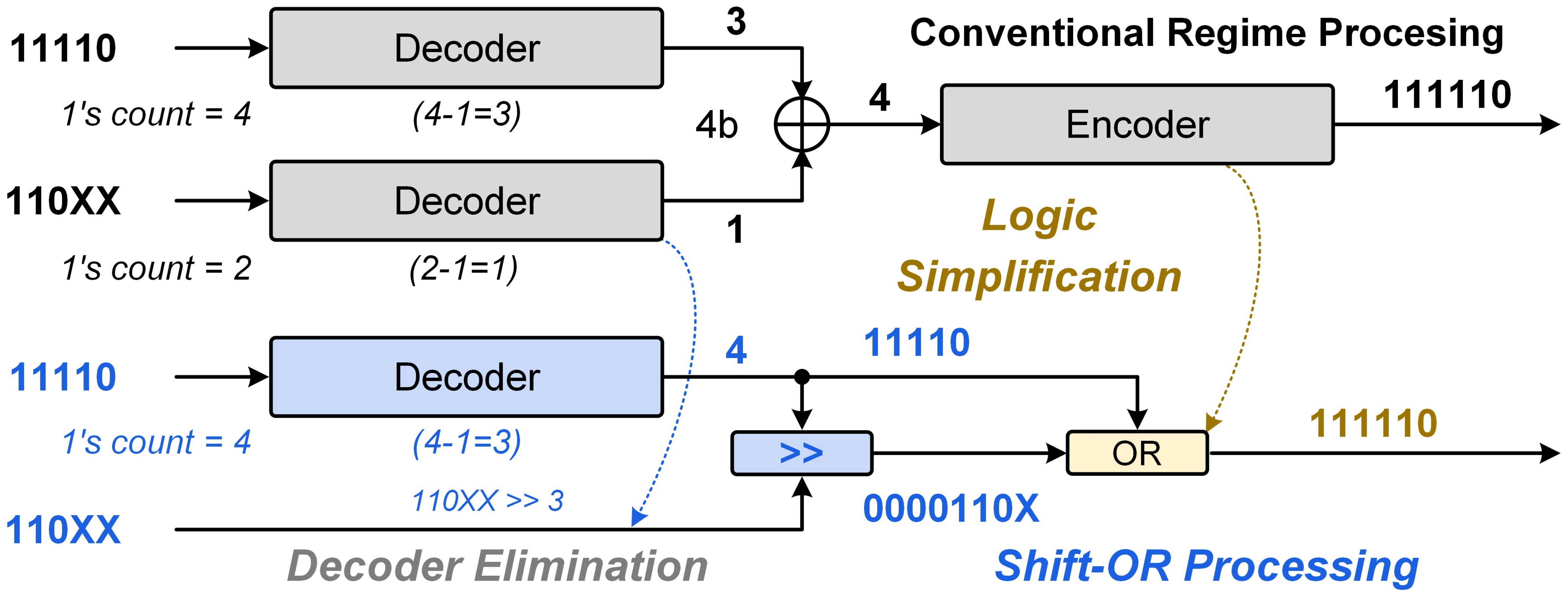
1's count sub 1 for pos. R (1110 for 2)

$$\mathbf{A \times B = (S_A \times S_B) \times (2^K)^{(R_A+R_B)} \times 2^{(E_A+E_B)} \times (1.f_A \times 1.f_B)}$$



- **Step1: Regime extracting with leading 1/0 detector.**
- **Step2: Regime processing with codec and addition.**

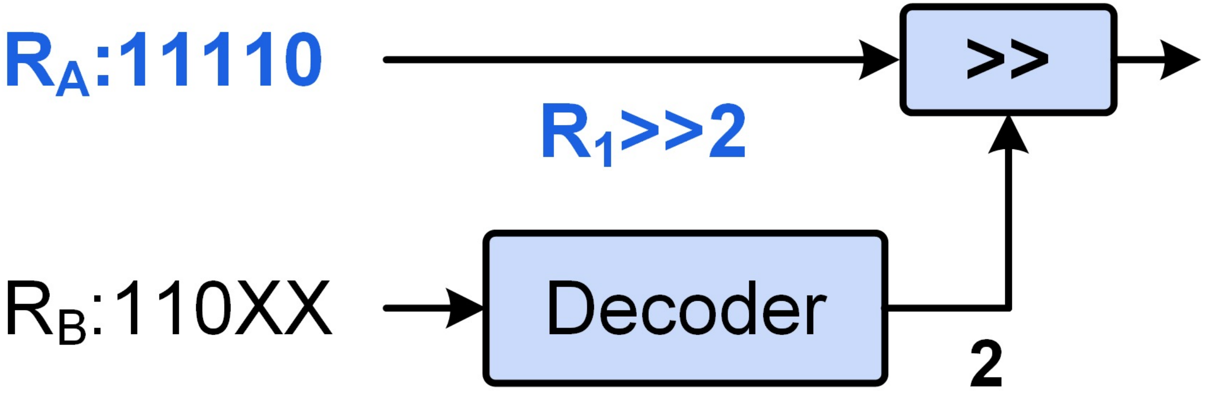
Feature 1: Bi-directional Regime Processing



- BRPU replaces codec-addition with shift-or processing.

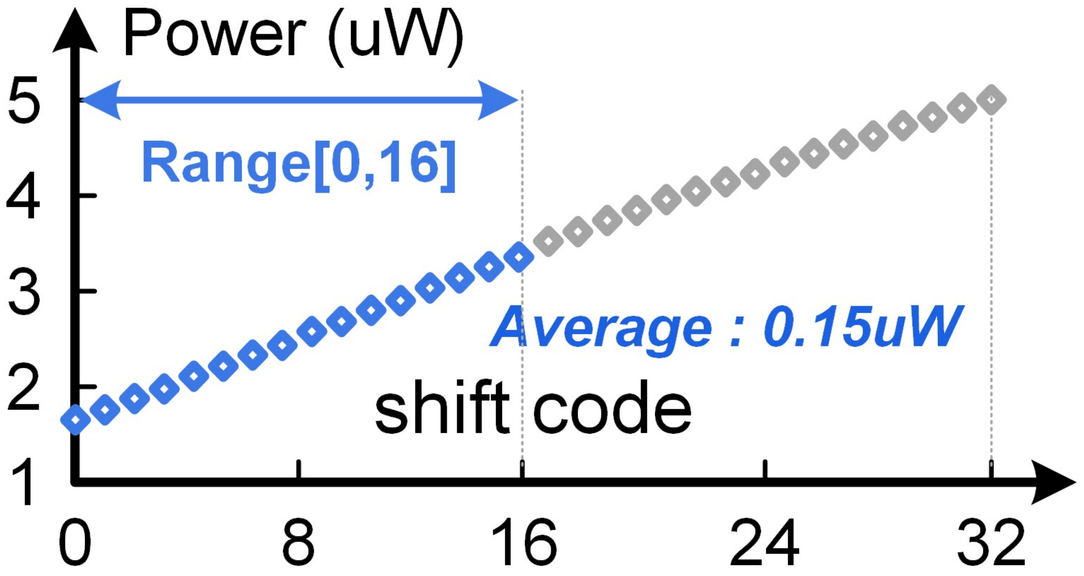
Feature 1: Bi-directional Regime Processing

Same Sign $R_A + R_B$



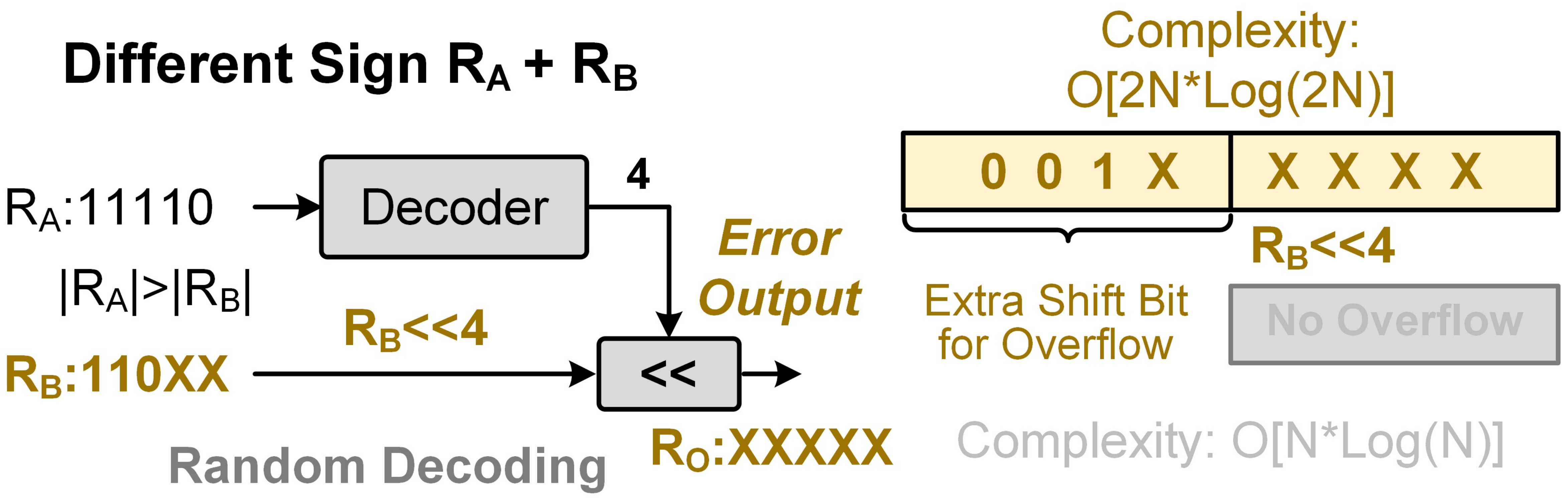
Small |R| Static Decoding

Simulation with TSMC 28nm Technology at 400MHz



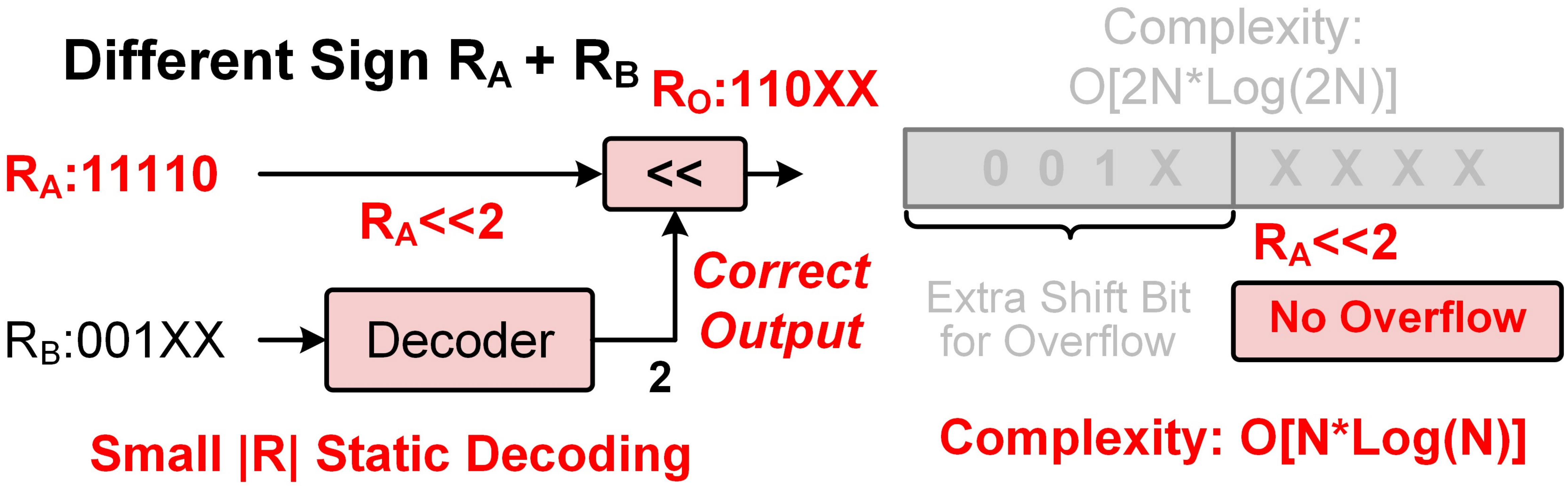
- **BRPU dynamically decodes small $|R_B|$ to shifts large $|R_A|$.**
- **BRPU minimizes shift code to saves 40% of shift energy.**

Feature 1: Bi-directional Regime Processing



- Different sign addition: logic shift to **decrease 1's/0's counts**.
- If shift code ≥ **R's effective bit-width**, it introduces shift error.

Feature 1: Bi-directional Regime Processing



- BRPU dynamically **decodes small $|R_B|$** to **shifts large $|R_A|$**.
- BRPU avoids shift overflow to **reduce 50% of shift logic**.

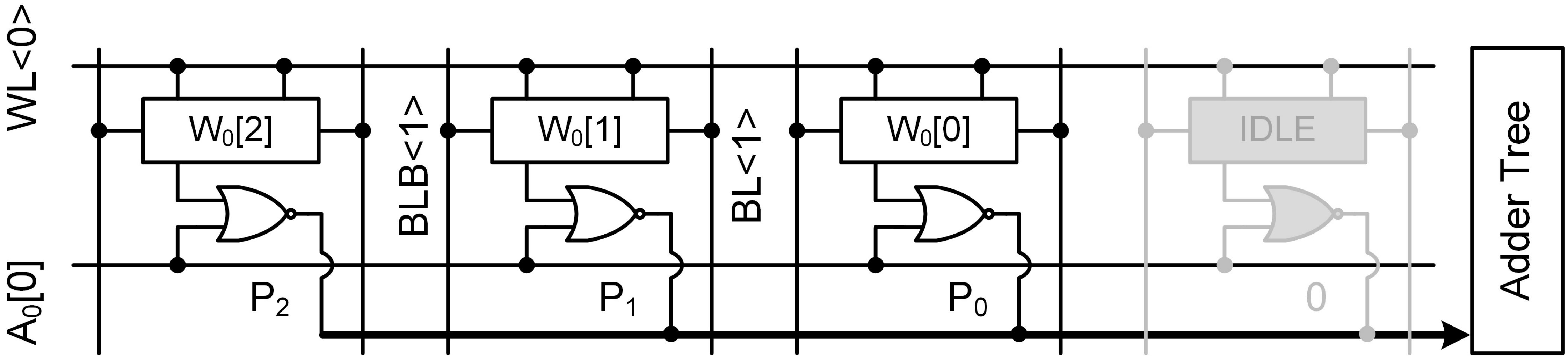
Outline

- Background and Motivation
- Challenges of POSIT-Based CIM Macro
- **Proposed POSIT@CIM Macro Features**
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- Measurement and Comparison
- Conclusion

Feature 2: Critical-bit Pre-compute-and-store

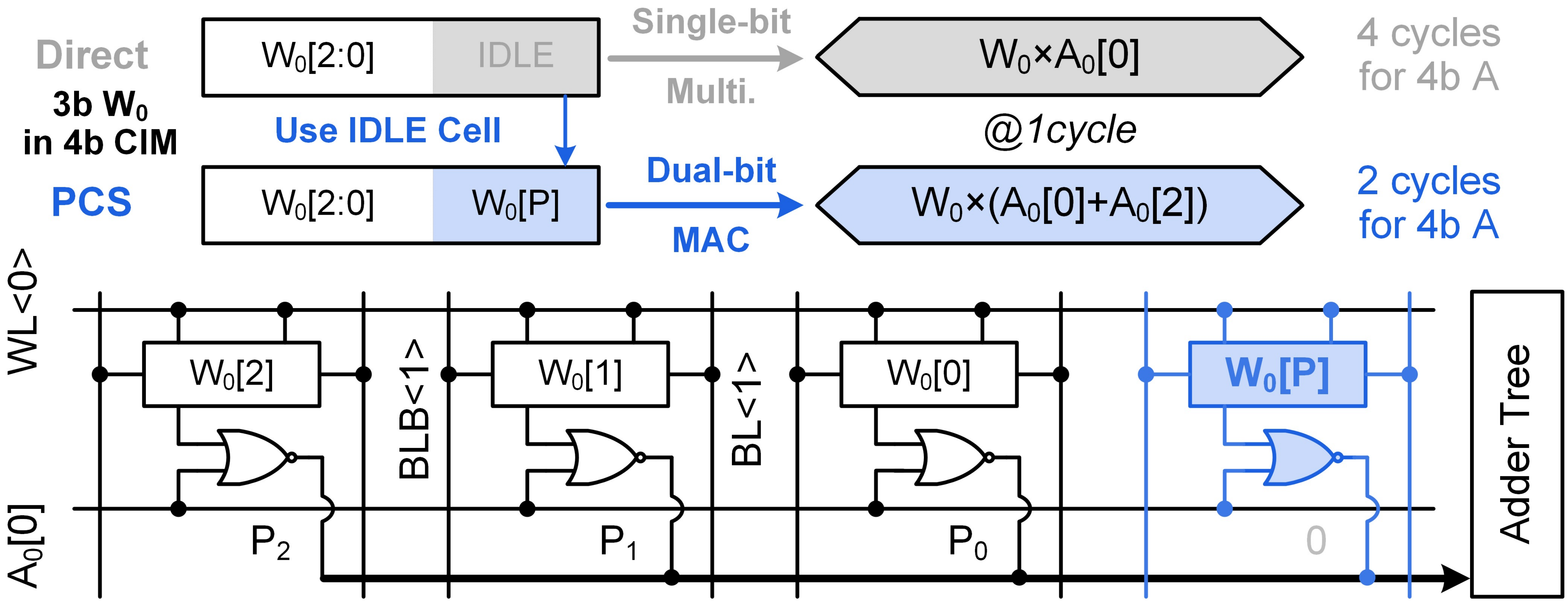
Mantissa Distribution of Posit Format Weight for ResNet18 Training

Posit(8,1)	2b(15.8%)	**3b(46.1%)**	**4b(31.6%)**	others
Posit(8,2)	2b(10.6%)	**3b(51.6%)**	**4b(34.2%)**	others



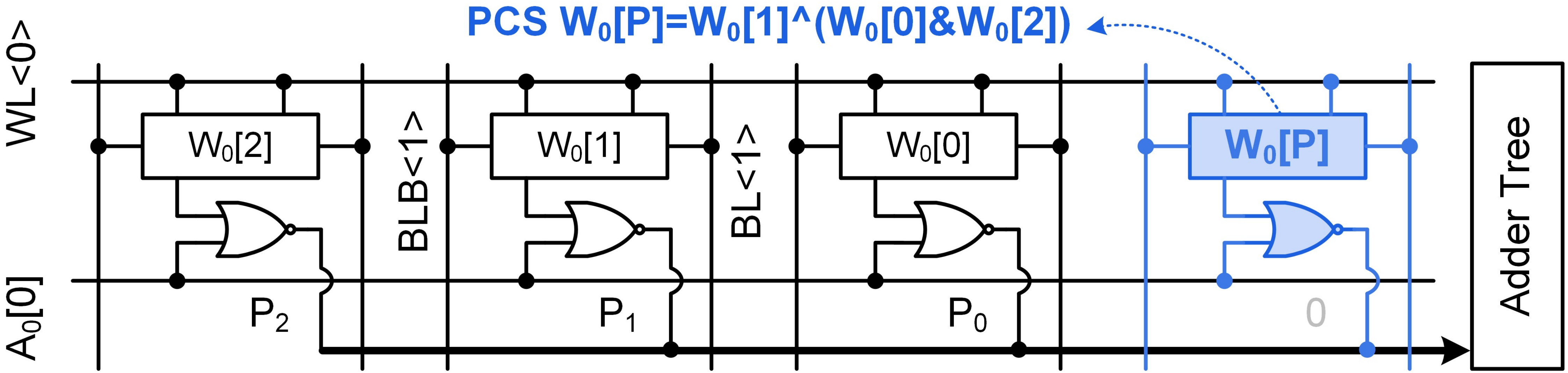
- **Dynamic mantissa bit-width introduces 48.9% cell waste.**

Feature 2: Critical-bit Pre-compute-and-store



- CPCS uses spare bits to achieve **dual-bit MAC** in each cycle.

Feature 2: Critical-bit Pre-compute-and-store



$$\text{Output} = W[2:0] \times A[0] + W[2:0] \times A[2]$$

$$\begin{array}{cccccc} & & P_2 & P_1 & P_0 \\ + & P_2 & P_1 & P_0 & & \\ \hline S_3 & S_2 & S_1 & S_0 & P_1 & P_0 \end{array}$$

$$O = \begin{cases} \{S[3:0],P[1:0]\}, & A[0],A[2]=11 \\ P[2:0], & A[0],A[2]\neq 11 \end{cases}$$

Feature 2: Critical-bit Pre-compute-and-store

Dual-bit MAC for 2b $W_m[1:0]$

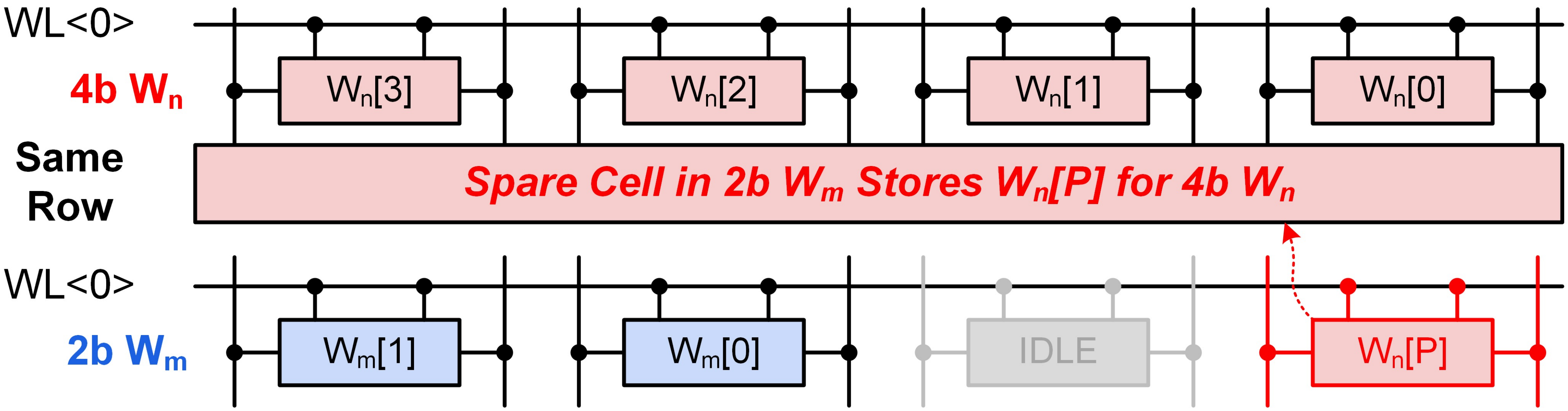
$$
\begin{array}{rcccccc}
 & & & P_2 & P_1 & P_0 \\
+ & P_2 & P_1 & P_0 & & \\
\hline
S_3 & P_2 & P_1 & P_2 & P_1 & P_0
\end{array}
$$

Do not Need Extra $W_m[P]$

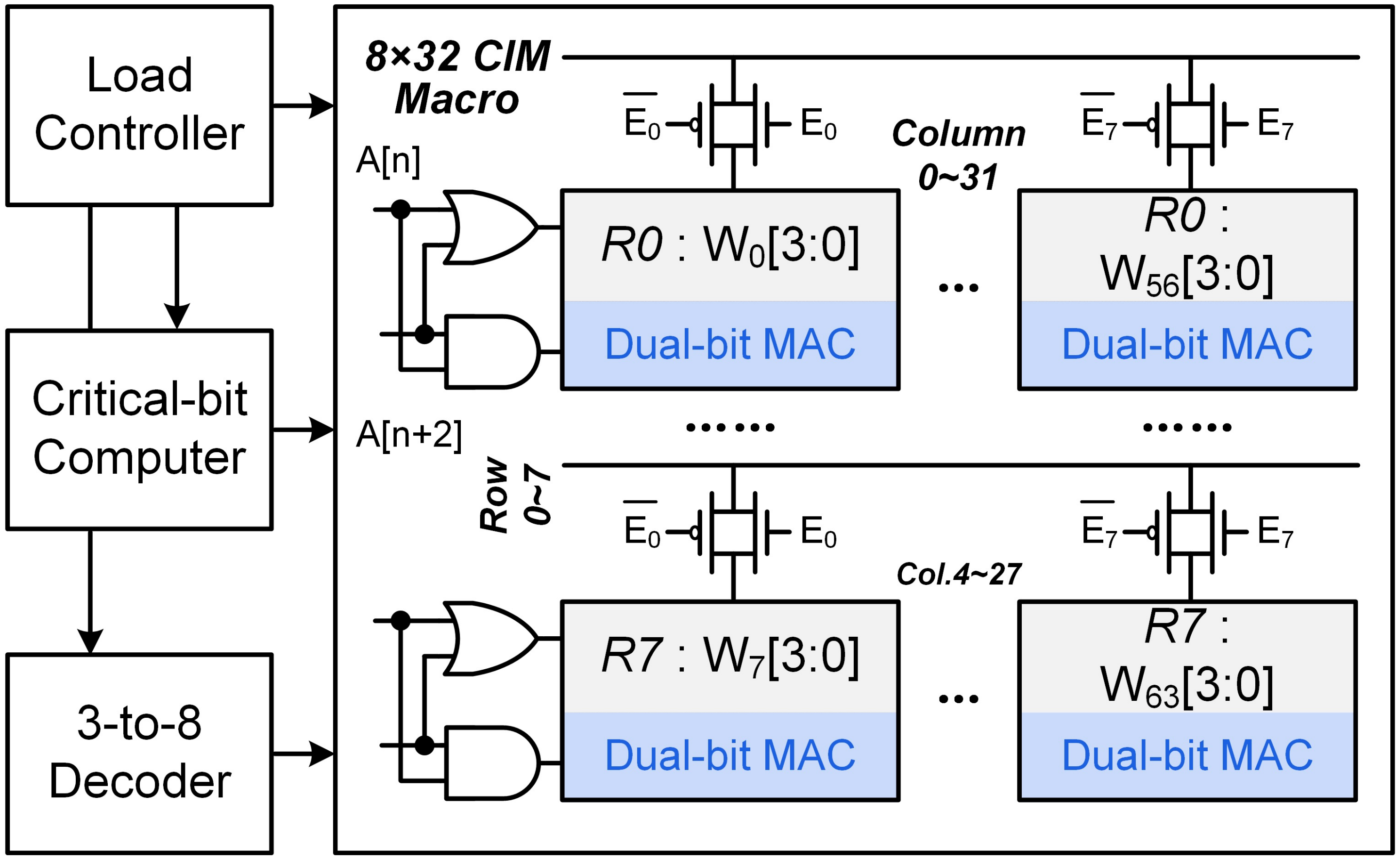
Dual-bit MAC for 4b $W_n[3:0]$

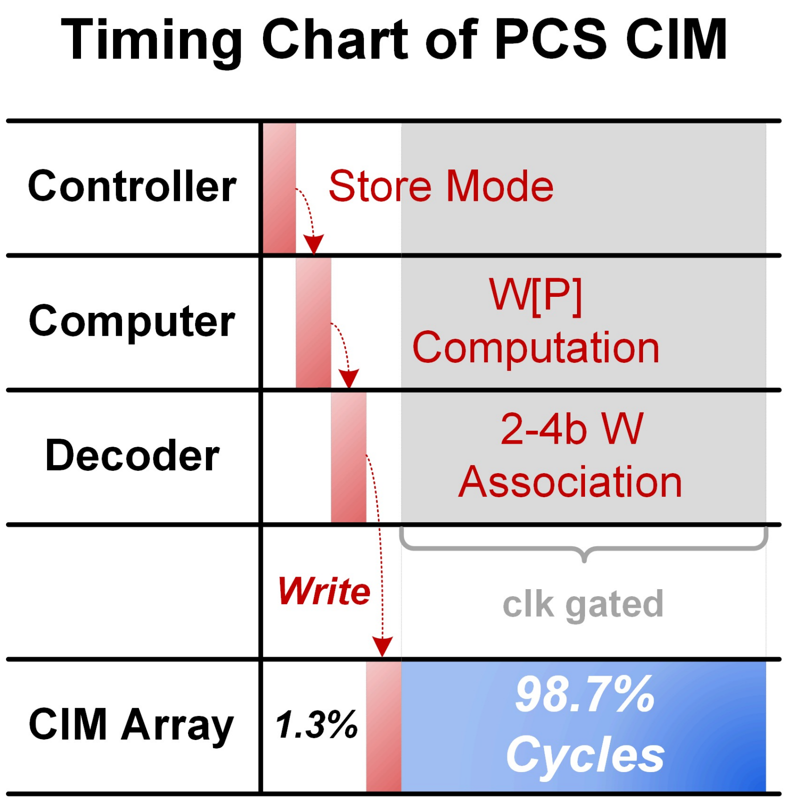
$$
\begin{array}{rccccccc}
 & & & & P_3 & P_2 & P_1 & P_0 \\
+ & P_3 & P_2 & P_1 & P_0 & P_0{=}0 & & \\
\hline
S_4 & S_3 & S_2 & S_1 & S_0 & P_1 & P_0 &
\end{array}
$$

Equivalent to 3b W_n if $W_n[0]=0$



Feature 2: Critical-bit Pre-compute-and-store





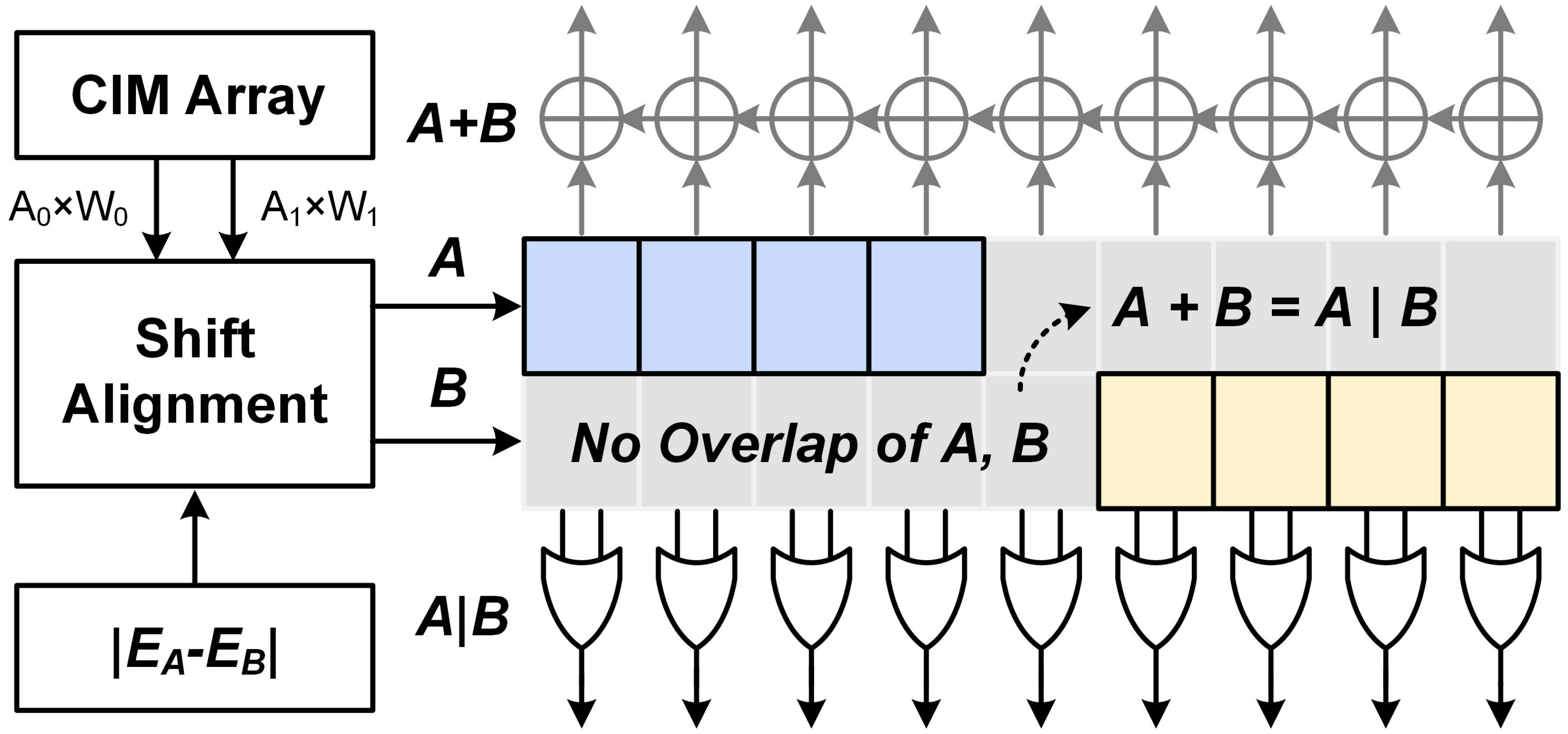
■ **Pre-computer only works one time before storing weight.**

Outline

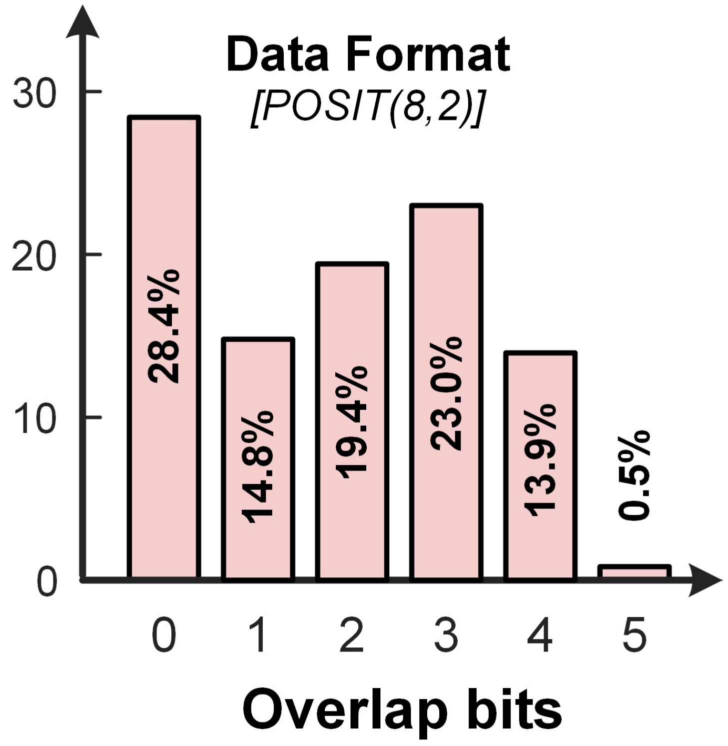
- Background and Motivation
- Challenges of POSIT-Based CIM Macro
- **Proposed POSIT@CIM Macro Features**
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- Measurement and Comparison
- Conclusion

Feature 3: Cyclically-alternating Scheduling

Bit-wise OR-based Accumulation

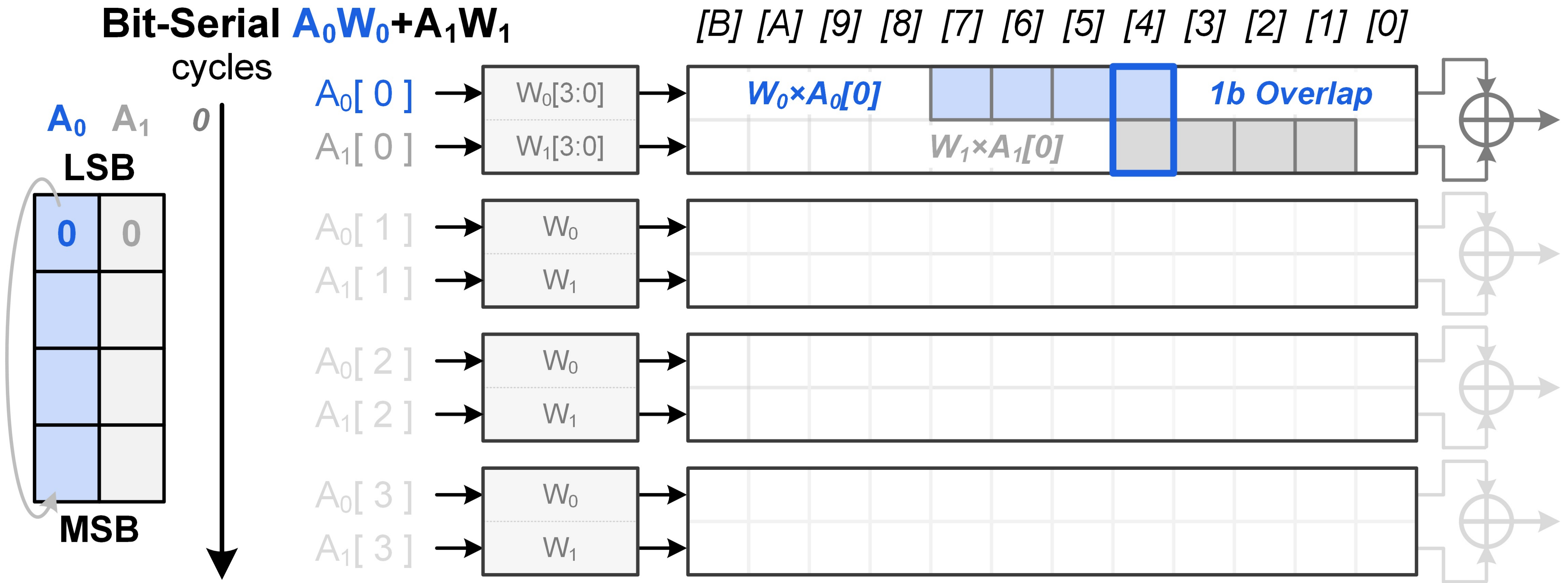


Overlap Ratio



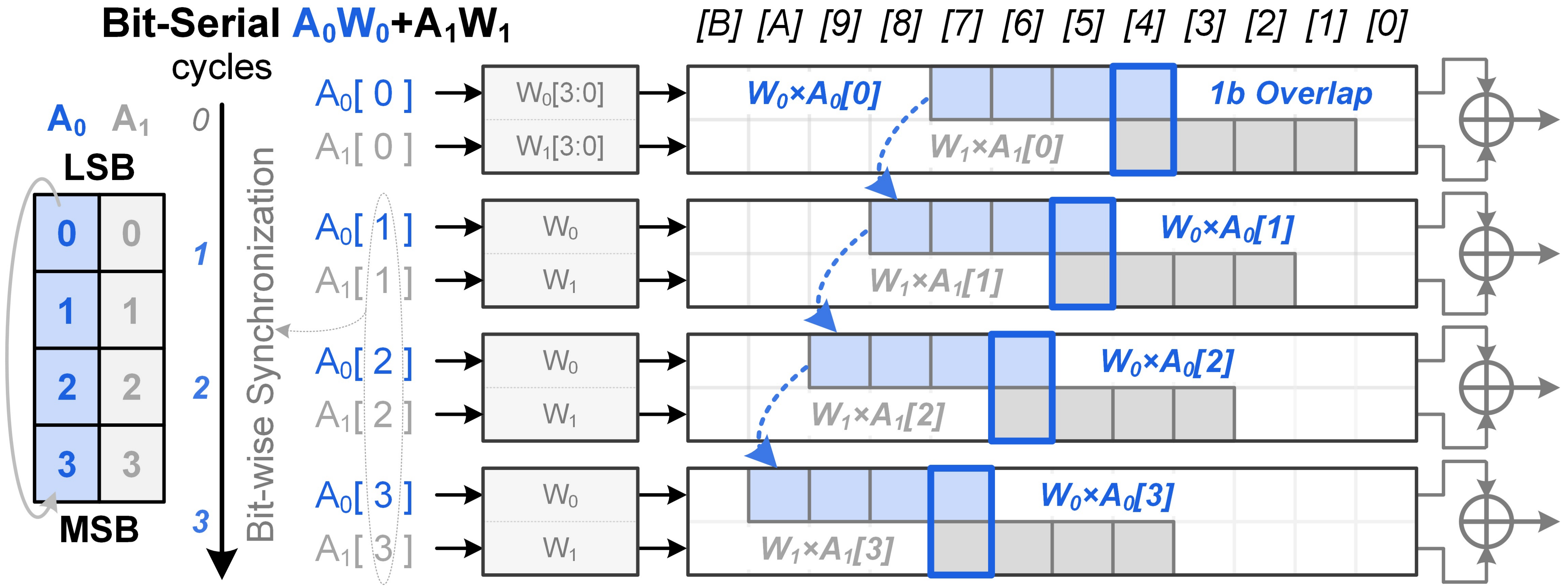
- If A and B have **no overlap bits**, A + B is equal to A | B.

Feature 3: Cyclically-alternating Scheduling



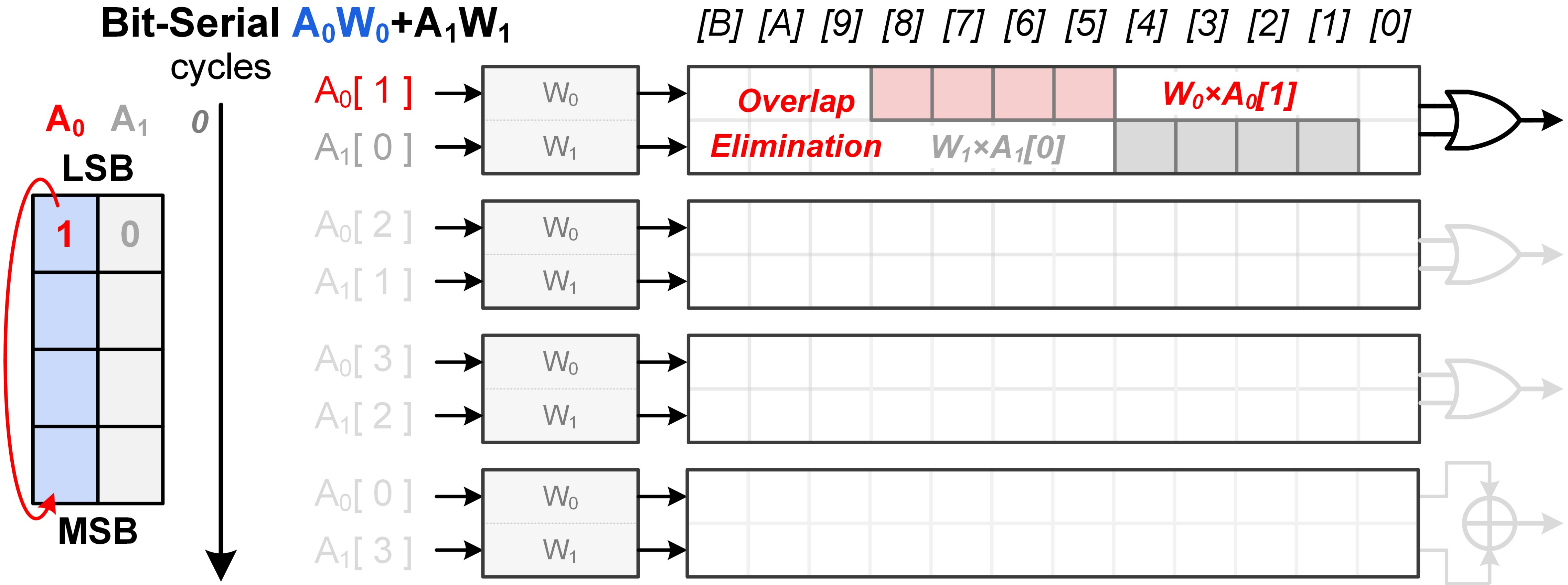
- Even A_0/A_1 have 1 overlap bit, $A_0W_0+A_1W_1$ has to use adder.

Feature 3: Cyclically-alternating Scheduling



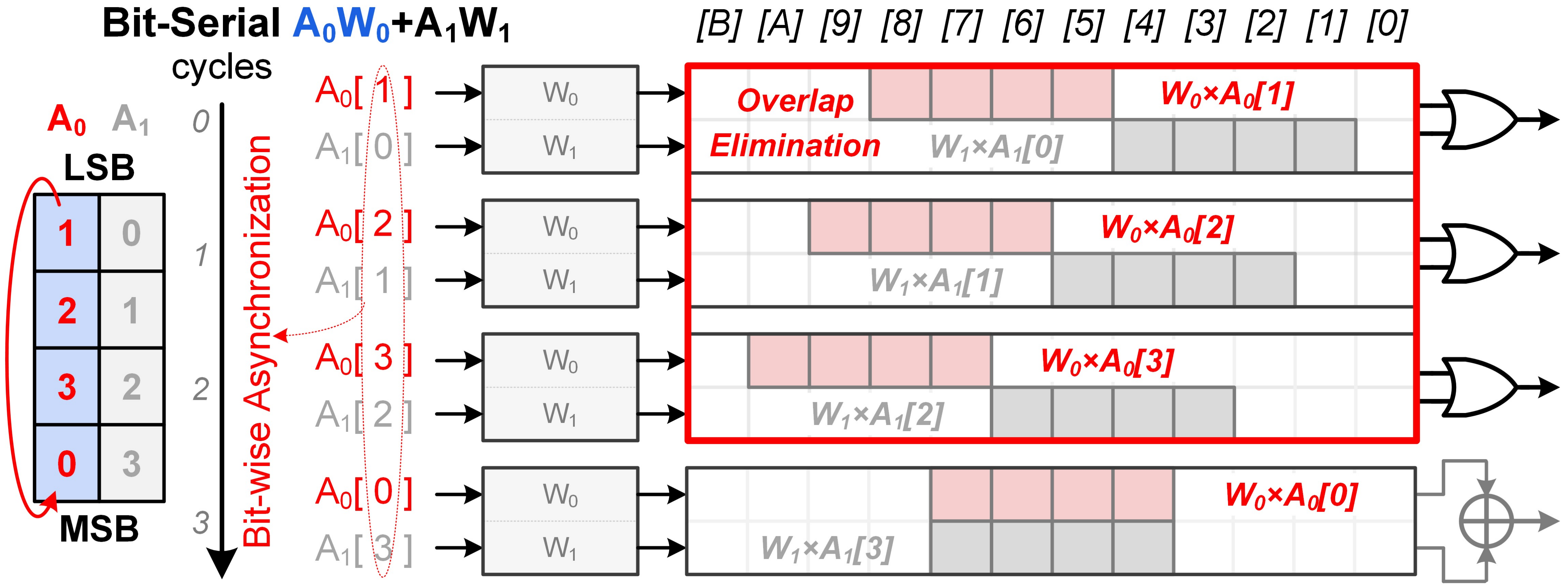
- All cycles need adders for synchronous bit-serial computing.

Feature 3: Cyclically-alternating Scheduling



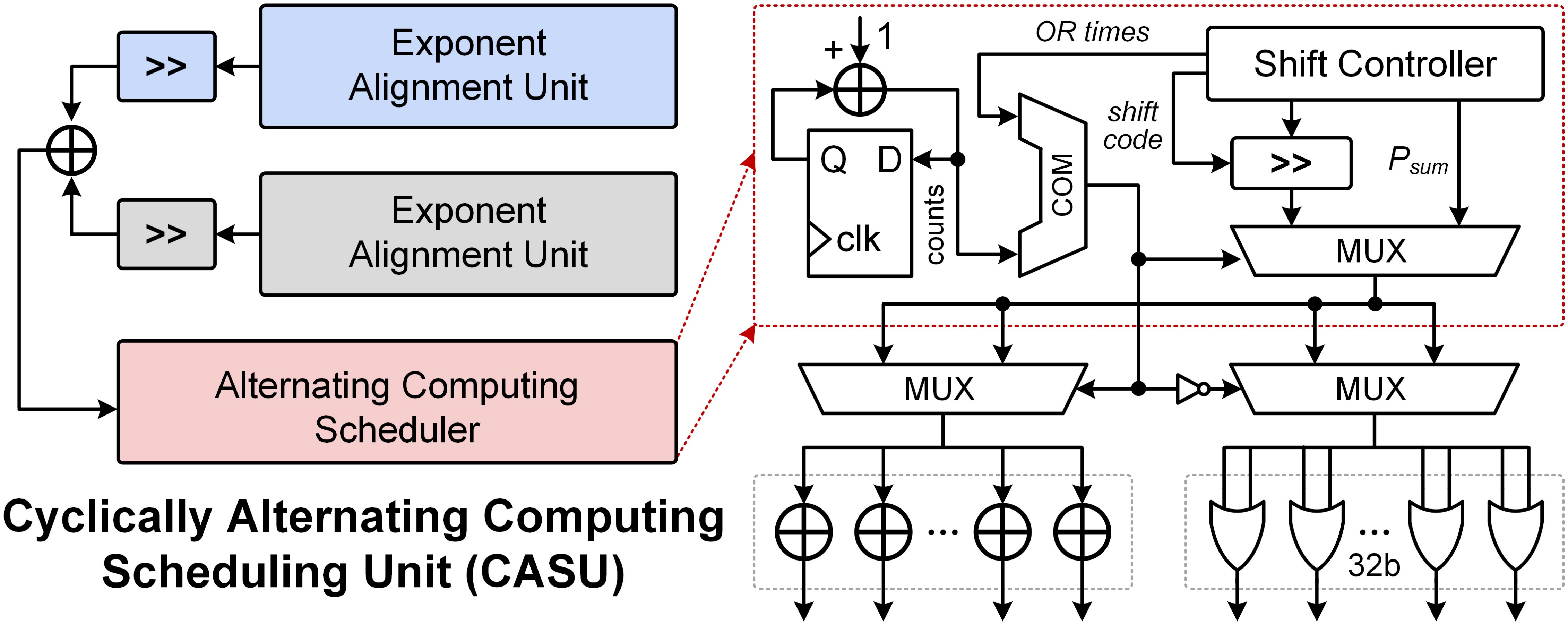
- CASU cyclically shifts A_0 for **asynchronous computing with A_1.**

Feature 3: Cyclically-alternating Scheduling



- CASU **eliminates overlap bits** in former cycles of $A_0W_0+A_1W_1$.

Feature 3: Cyclically-alternating Scheduling



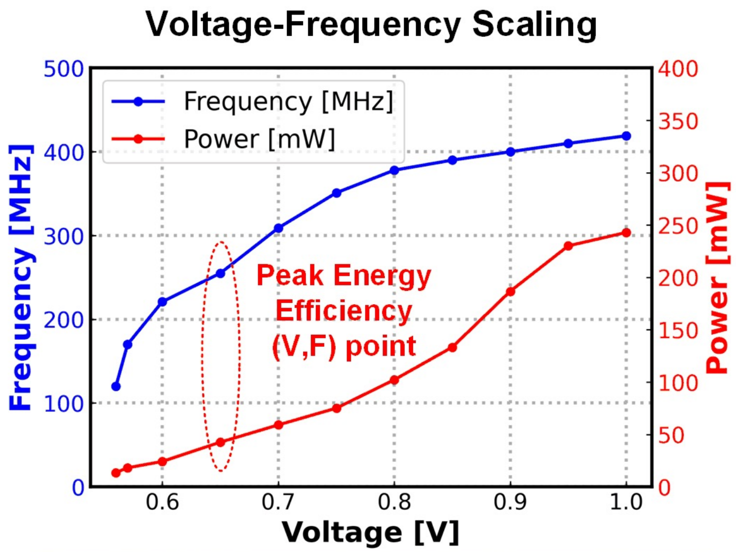
Cyclically Alternating Computing Scheduling Unit (CASU)

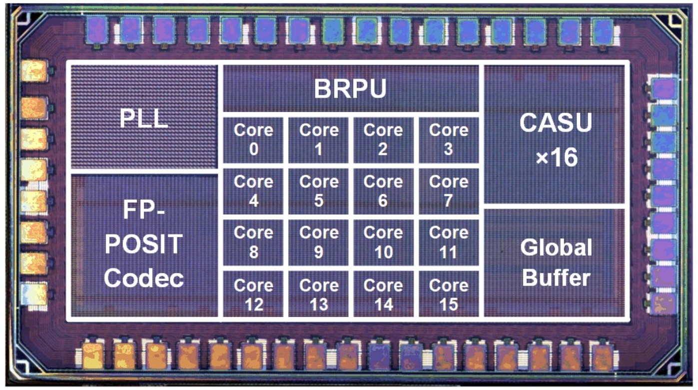
- **CASU saves 56.9% of accumulation energy for adder tree.**

Outline

- Background and Motivation
- Challenges of POSIT-Based CIM Macro
- Proposed POSIT@CIM Macro Features
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- **Measurement and Comparison**
- Conclusion

Chip Photograph and Summary





		Specifications	
Technology		28nm CMOS	
Die Area		1.41 mm^2	
CIM Size		12 KB	
Buffer Size		28 KB	
Voltage		0.55V-1.0V	
Frequency		78MHz-419MHz	
Data Precision		Posit(8,1)	Posit(16,2)
Peak Performance[1)]		3.86TOPS	1.91TOPS
Area Efficiency		2.74TOPS/mm^2	1.35TOPS/mm^2
Cim Micro Energy Efficiency[2)]		16.34-83.23 TOPS/W	7.47-38.37 TOPS/W
System Energy Efficiency[2)]		10.90-55.60 TOPS/W	5.35-27.61 TOPS/W
Differerent AI Models	ResNet18[3)] @Imagenet1k	69.64% (Top1 Acc↑)	69.71% (Top1 Acc↑)
	GPT-2[4)] @Wikitext-2	21.31 (Perplexity↓)	21.45 (Perplexity↓)
	VIT-B[5)] @Imagenet1k	80.17% (Top1 Acc↑)	80.05% (Top1 Acc↑)

One operation (OP) represents one multiplication or addition.

1) Highest performance (lowest effiency) point, 1.0V, 419MHz

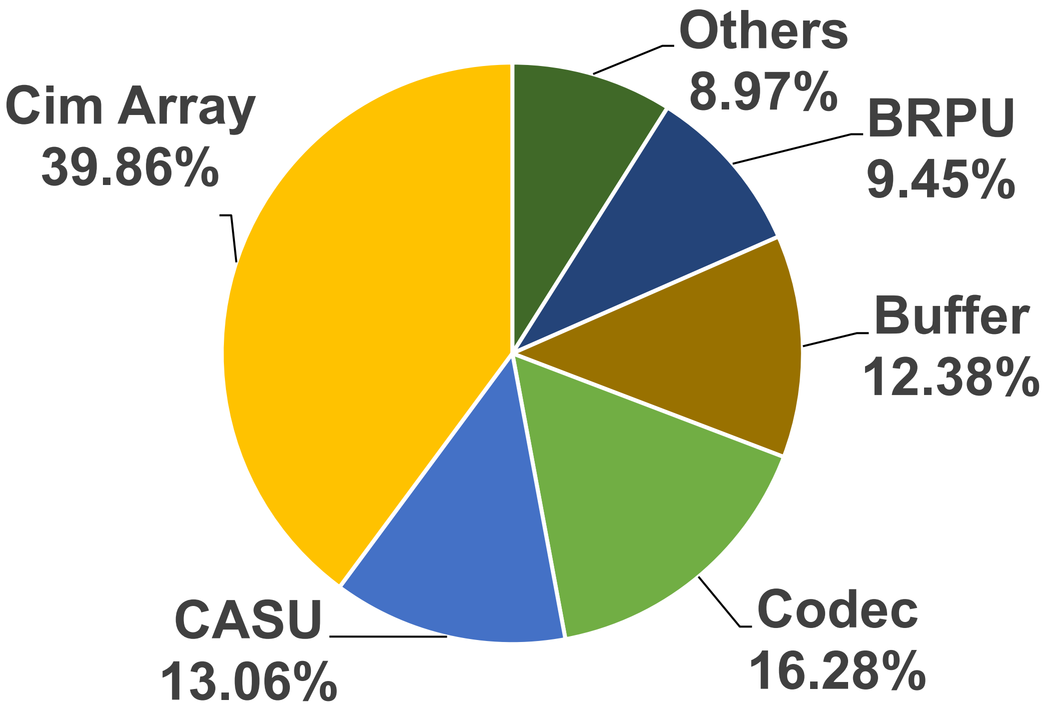
2) Highest efficiency point, 0.65V, 78MHz, 50% input sparsity

3) The baseline is 69.76%

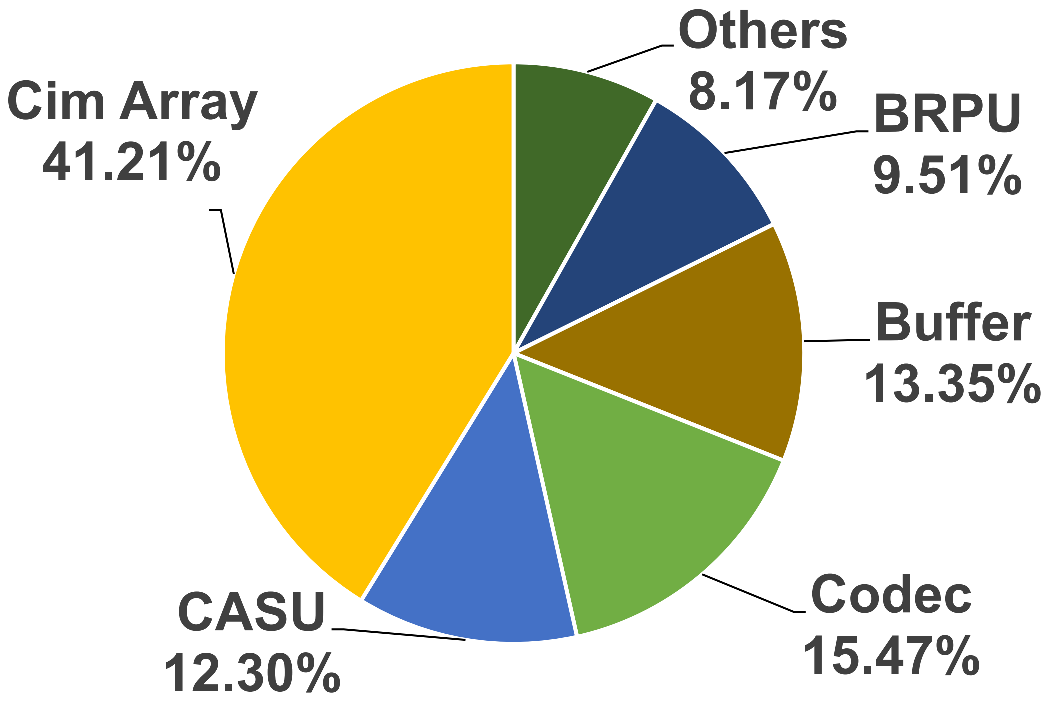
4) The baseline is 21.30

5) The baseline is 80.31%

Area and Power Breakdown



Area Breakdown



Power Breakdown

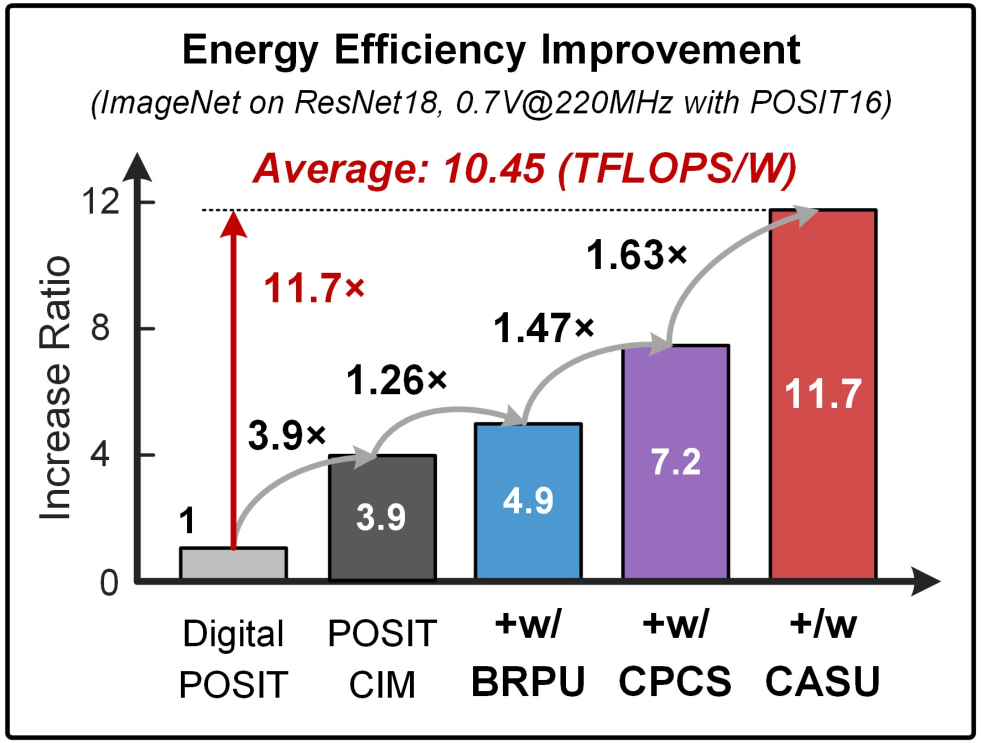
- BRPU and CASU take limited area (22.5%) and power (21.8%)
- CIM Array takes most area (39.9%) and power (41.2%)

Training and Inference Performance

Evaluation on Different Models[1)]

Model	ResNet18	GPT-2	VIT-B
Dataset	Imagenet-1k	Wikitext-2	Imagenet-1k
Task	Training	Inference	Inference
Data Precision	Posit(16,2)	Posit(8,1)	Posit(8,1)
Accuracy	69.71% (Top1 Acc)	21.31 (Perplexity)	80.17% (Top1 Acc)
Accuracy Loss[2)]	-0.04%	-0.05%	-0.14%
Performance(TOPS)[3)]	1.73	2.95	2.91
Cim Micro Energy Efficiency (TOPS/W)	16.27	34.30	33.81
System Energy Efficiency (TOPS/W)	10.45	21.79	21.48
Energy Saving[5)]	8.81x	7.90x	7.12x

1) Measured at 1.0V, 419 MHz for high-performance evaluation. 2) Compared with models training in FP32.
3) Include all on-chip components,. Off-chip memory is not included.



- POSIT-CIM only incurs 0.14% of accuracy loss than FP32.
- It achieves 10.45TFLOPS/W of average energy efficiency.

Performance Comparison

Comparison with SOTA FP CIM Macros

	VLSI'21[1]	ISSCC'22[2]	ISSCC'23[3]	ISSCC'23[4]	ISSCC'21[5]	ISSCC'21[6]	This Work
Dynamic Format	NO	NO	NO	NO	NO	YES	**YES**
Technique (nm)	28	28	22	28	28	28	**28**
Die Area (mm^2)	5.83	6.69	18	0.146	4.54	3.8	**1.41**
Supply Voltage (V)	0.76-1.1	0.6-1.0	0.6-0.8	0.6-0.9	0.397-0.90	0.6-0.9	**0.55-1.0**
Frequency (MHz)	250	50-220	NA	NA	10-400	104-288	**78-419**
Precision	BF16	FP32/BF16 INT16/INT8	BF16	BF16 INT8	FP16/BF16 INT8/4	32b/16b/8b CUSTOM POSIT	**POSIT16 POSIT8**
Power (mw)	1.2-156.1[1)]	12.5-69.4	NA	NA	0.87-74.9	50-230	**5.5-237**
Performance (TOPS)	0.12-0.66[1)]	0.14@FP32 1.35@INT8	1.24-1.28	NA	1.64-9.63[3)]@INT4	0.0163@POSIT16 0.0337@POSIT8	**1.91@POSIT16[2)] 3.86@POSIT8[2)]**
Energy Efficiency (TOPS/W)	1.43-13.7[1)]	3.7@FP32 36.5@INT8	16.2-70.2[1)]	14-31.6@BF16[2)] 19.5-44@INT8[2)]	3.2-16.9[3)]@FP16 51-300[3)]@INT4	0.121@POSIT16 0.248@POSIT8	**38.37@POSIT16[2)] 83.23@POSIT8[2)]**
Area Efficiency (TOPS/ mm^2)	0.021-1.1[1)]	0.02@FP32 0.20@INT8	0.069-0.071	NA	0.36-2.12[3)]@INT4	0.0043@POSIT16 0.0089@POSIT8	**1.35@ POSIT16[2)] 2.74@ POSIT8[2)]**

1) Evaluated with 90% input sparsity. 2) Evaluated with 50% input sparsity. 3) From dense models to average of test sparse NN models.

Outline

- Background and Motivation
- Challenges of POSIT-Based CIM Macro
- Proposed POSIT@CIM Macro Features
  - Bi-directional Regime Processing Codec
  - Critical-bit Pre-compute-and-store CIM Array
  - Cyclically-alternating Scheduling Adder Tree
- Measurement and Comparison
- **Conclusion**

Conclusion

- **An Energy Efficient POSIT-Based CIM Macro**
  - Bi-directional Regime Processing Codec
    - ✓ Save Pre-processing Energy by Replacing Codec to Shift-OR
  - Critical-bit Pre-compute-and-store CIM Array
    - ✓ Improve CIM Utilization by Using Spare Bit for Dual-bit MAC
  - Cyclically-alternating Scheduling Adder Tree
    - ✓ Reduce Accumulation Power by Simplifying Addition to OR

A POSIT-Based CIM Macro with Bi-directional Regime Codec, Critical-bit Pre-computing-Storing and Cyclically-alternating Scheduling Achieving 83.23TFOPS/W Energy Efficiency