High-Parallel In-Memory NTT Engine with Hierarchical Structure and Even-Odd Data Mapping

Bing Li[1], Huaijun Liu[2], Yibo Du[3,4], Ying Wang[3,4]

Institute of Microelectronics, Chinese Academy of Sciences[1]
Capital Normal University[2]
Institute of Computing Technology, Chinese Academy of Sciences[3],
University of Chinese Academy Sciences[4]

Outline

- **Background and Motivation**
- **Proposed Method**
  - **Overview**
  - **Architecture & Data Mapping**
- **Evaluation and Results**
- **Conclusion**

Fully Homomorphic Encryption

Medical Treatment

Machine Learning

Cloud Computing

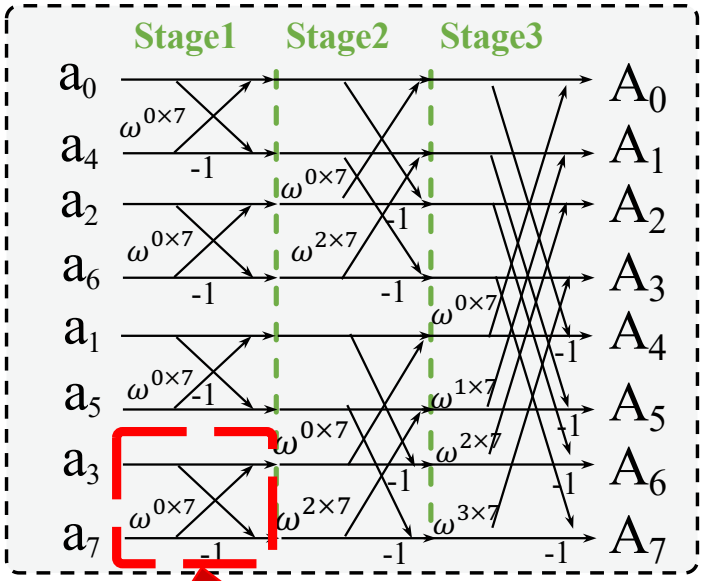
Fitness App

- Data Security
- Powerful Functionality
- High Computational Overhead

➢ **FHE Review**

[Viand A, et al., S&P 2021]

Classic NTT Challenges & Advantages



Butterfly Operation

Algorithm In-Place Cooley-Tukey-based NTT

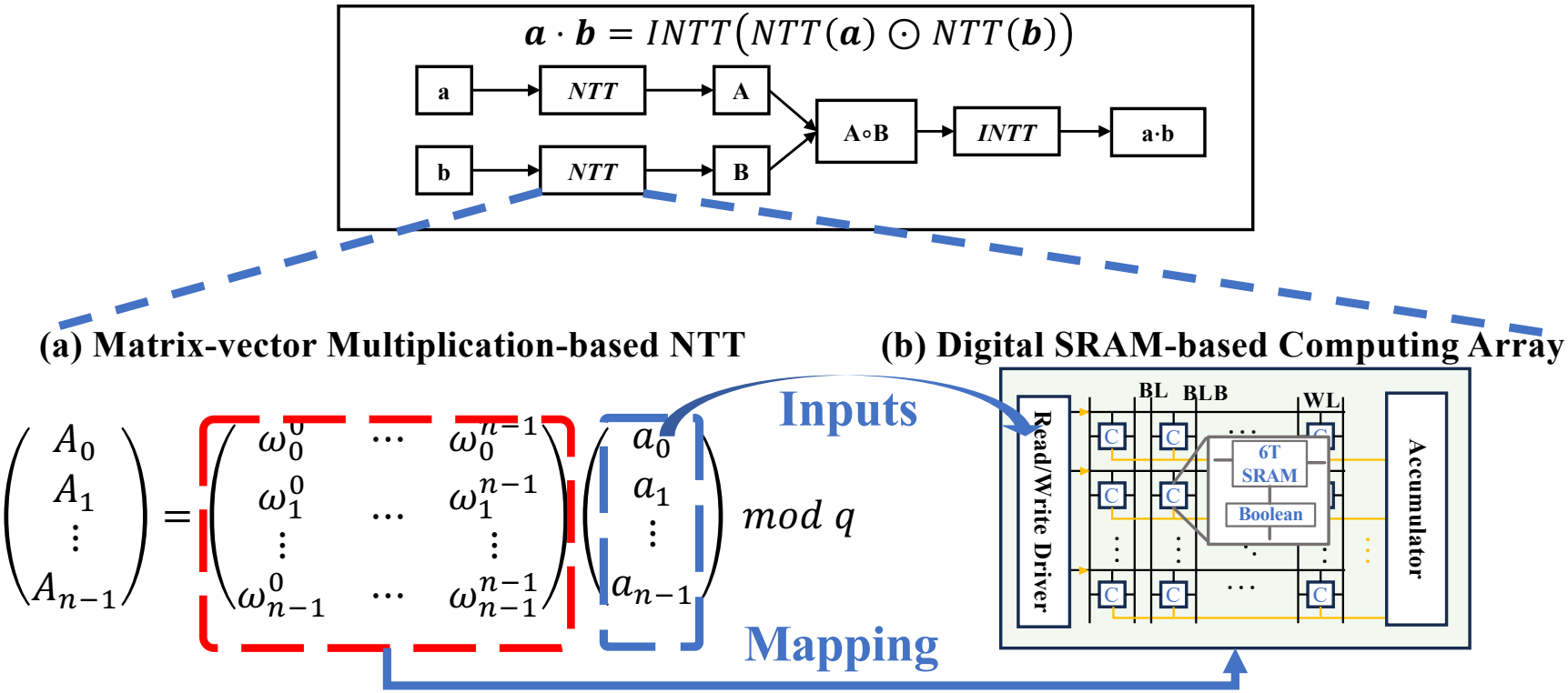
Input: $\mathbf{a} = (a_{n-1}, ..., a_0) \in R$, n-th root of unity ω in $\mathbb{Z}_q$ with bit-reversed order

Output: $A = NTT(\mathbf{a})$ in bit-reversed order

1: $t = n$
2: **for** $(m = 1; m < n; m = 2m)$ **do**
3: $t = t/2$
4: **for** $(i = 0; i < m; i++)$ **do**
5: **for** $(j = 2i \cdot t; j \leq 2i \cdot t + t - 1; j++)$ **do**
6: $A_j = a_j + a_{j+t} \cdot \omega_n^{m+i} \bmod q$
7: $A_{j+t} = a_j - a_{j+t} \cdot \omega_n^{m+i} \bmod q$
8: **end for**
9: **end for**
10: **end for**
11: **return** A

👎 Serially stage computation 👎 Complex data transfer 👍 ω with n/2 size

MVM-Based NTT Challenges & Advantages



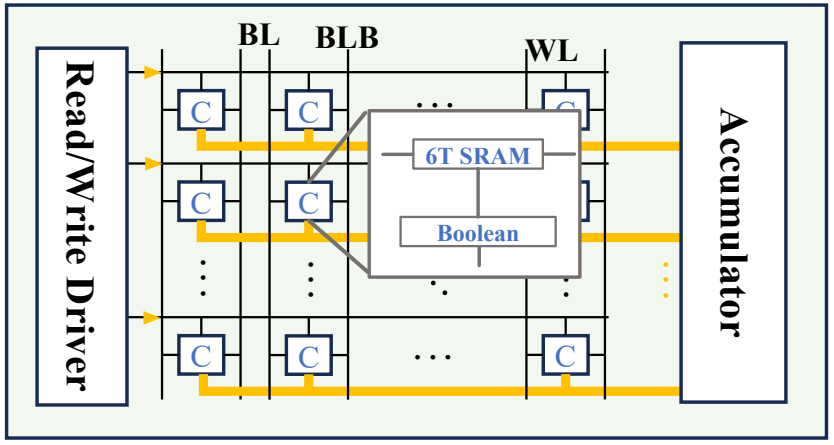
👎 ω with n^2 size

👍 High Parallelism and Efficiency

Digital in-SRAM Computing

- **High-Efficiency Parallel Computation**
- **Computing-in-Memory Capacity**
- **Digital Matrix-Vector Multiplication**

Multiple
Bit-serial inputs



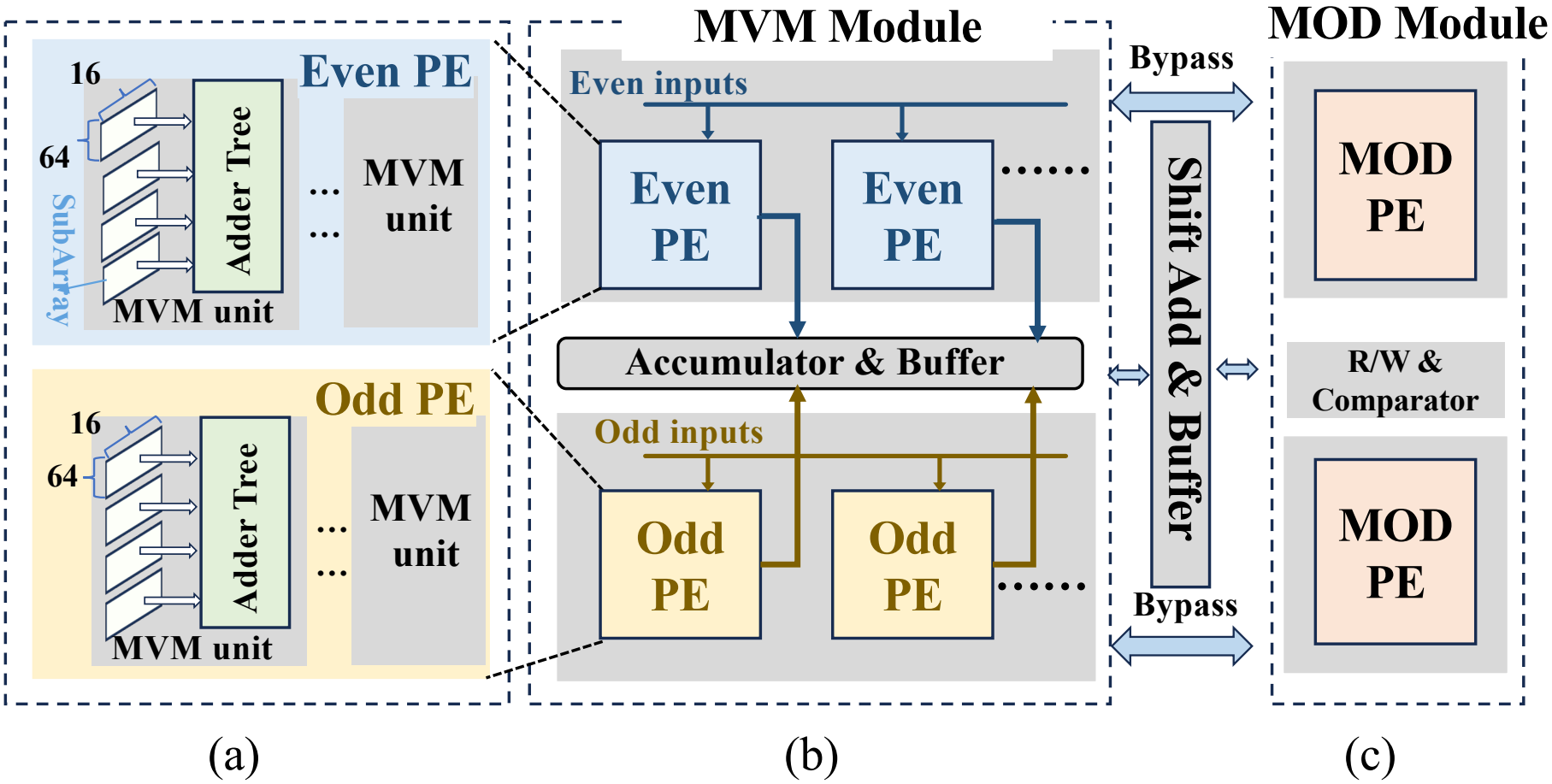
➢ **All-Digital SRAM-Based ML Edge Applications**

[Yu-Der Chih, et al., ISSCC 2021]

Outline

- Background & Motivation
- Proposed Method
  - Overview
  - Architecture & Data Mapping
- Evaluation and Results
- Conclusion

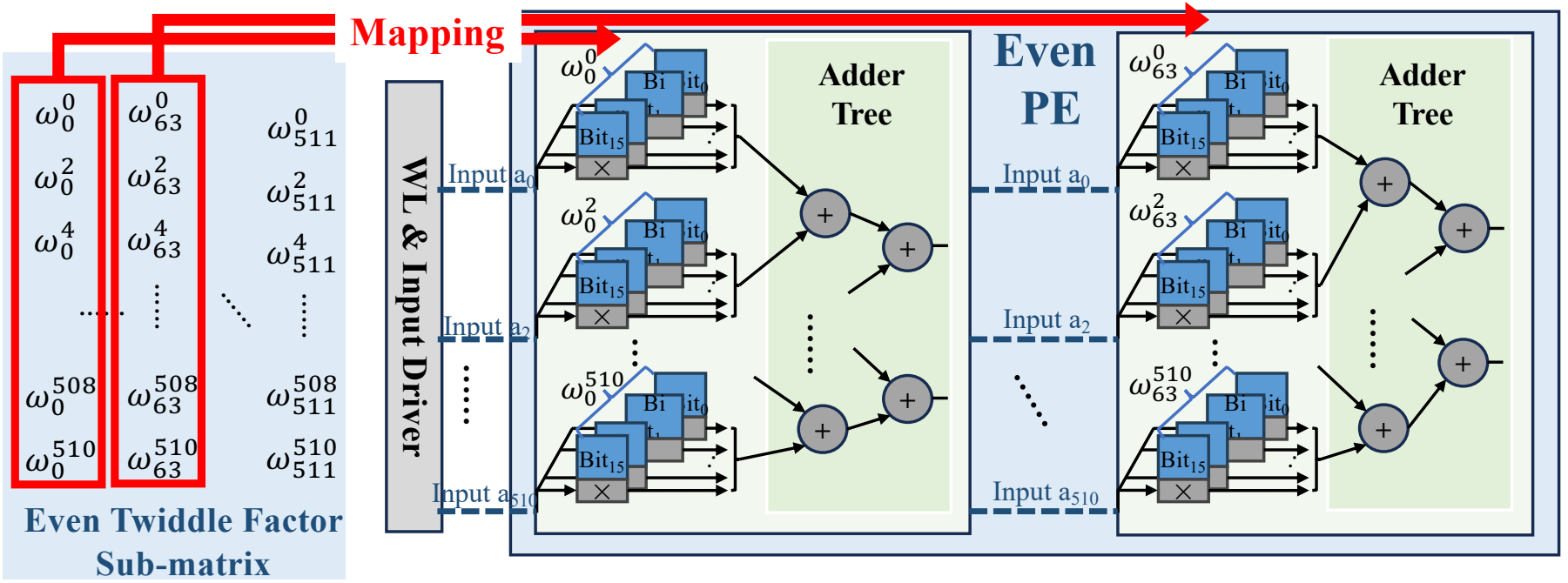
Overview


Even PE
16
64
SubArray
Adder Tree
MVM unit
MVM unit
Odd PE
16
64
Adder Tree
MVM unit
MVM unit
MVM Module
Even inputs
Even PE
Even PE
Accumulator & Buffer
Odd inputs
Odd PE
Odd PE
Bypass
Shift Add & Buffer
Bypass
MOD Module
MOD PE
R/W & Comparator
MOD PE
(a)
(b)
(c)

Outline

- Background & Motivation
- Proposed Method
  - Overview
  - Architecture & Data Mapping
- Evaluation and Results
- Conclusion

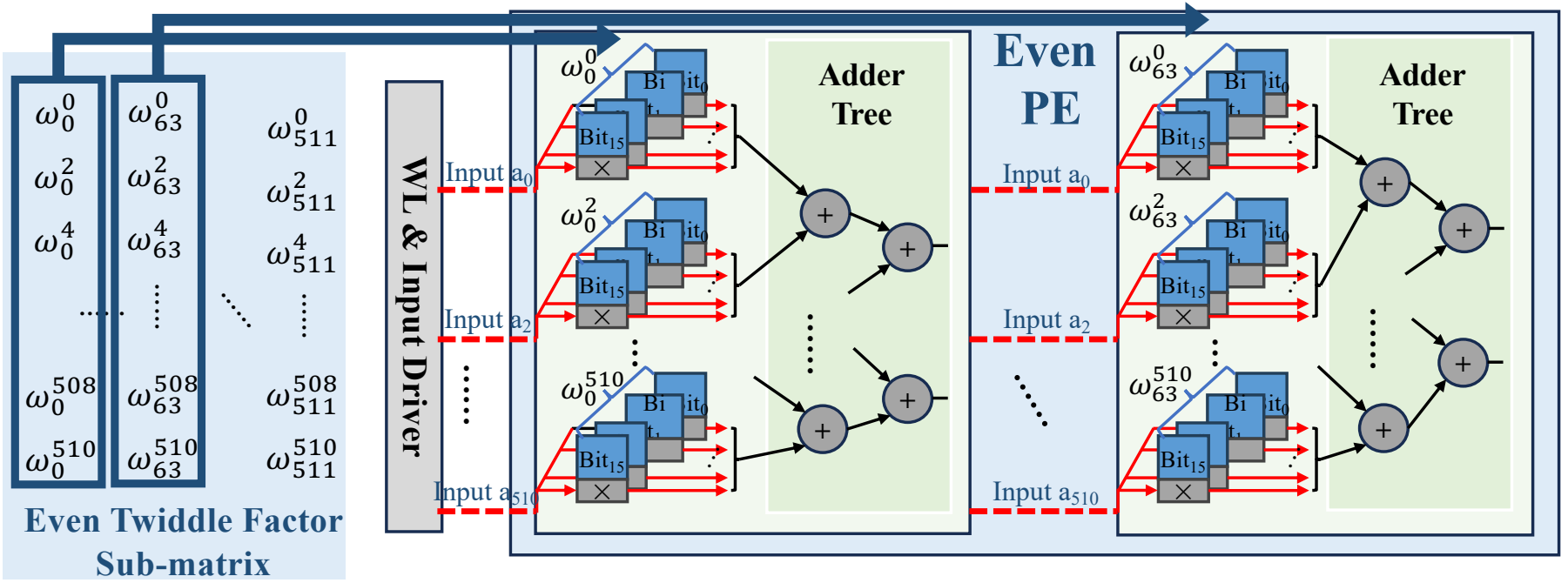
MVM Module-Data Mapping



$$A_i = \sum_{j=0}^{n-1} \omega_i^j a_j \bmod q$$

$$\begin{cases} A_k = [\sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i)} (a_{2i} + a_{2i+\frac{n}{2}}) + \sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i+1)} (a_{2i+1} + a_{2i+1+\frac{n}{2}})] \bmod q \\ A_{k+\frac{n}{2}} = [\sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i)} (a_{2i} + a_{2i+\frac{n}{2}}) - \sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i+1)} (a_{2i+1} + a_{2i+1+\frac{n}{2}})] \bmod q \end{cases}$$

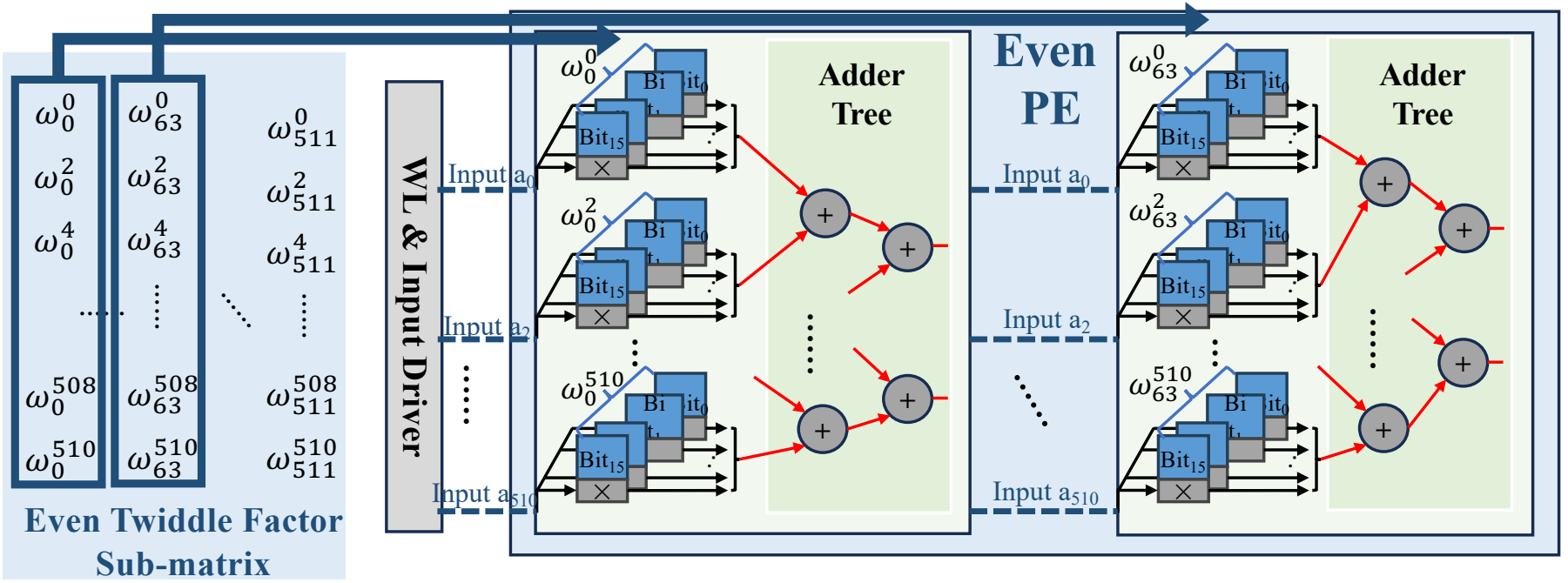
MVM Module-Data Mapping



$$A_i = \sum_{j=0}^{n-1} \omega_i^j a_j \bmod q$$

$$\begin{cases} A_k = [\sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i)}(a_{2i} + a_{2i+\frac{n}{2}}) + \sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i+1)}(a_{2i+1} + a_{2i+1+\frac{n}{2}})] \bmod q \\ A_{k+\frac{n}{2}} = [\sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i)}(a_{2i} + a_{2i+\frac{n}{2}}) - \sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i+1)}(a_{2i+1} + a_{2i+1+\frac{n}{2}})] \bmod q \end{cases}$$

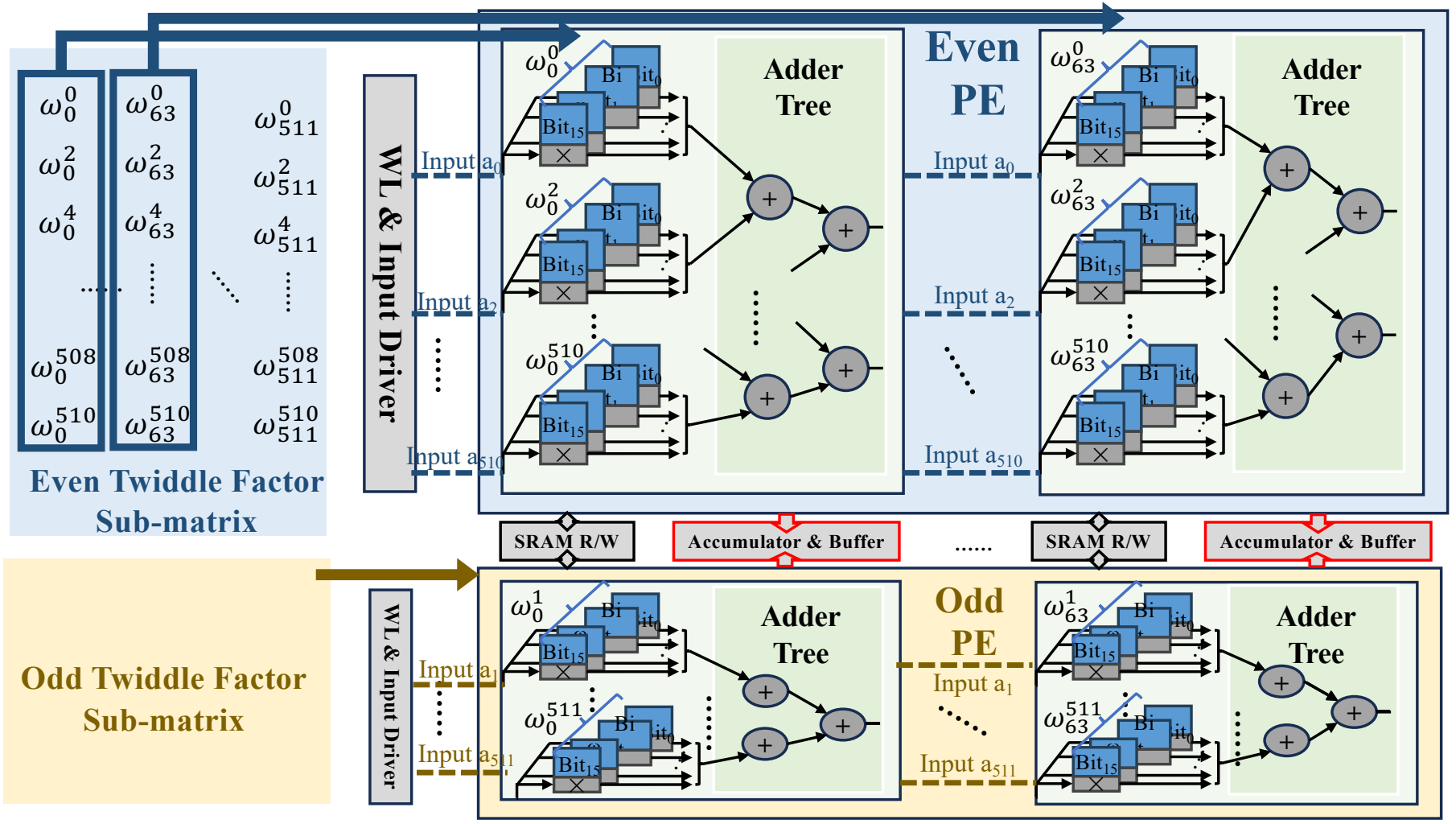
MVM Module-Computation



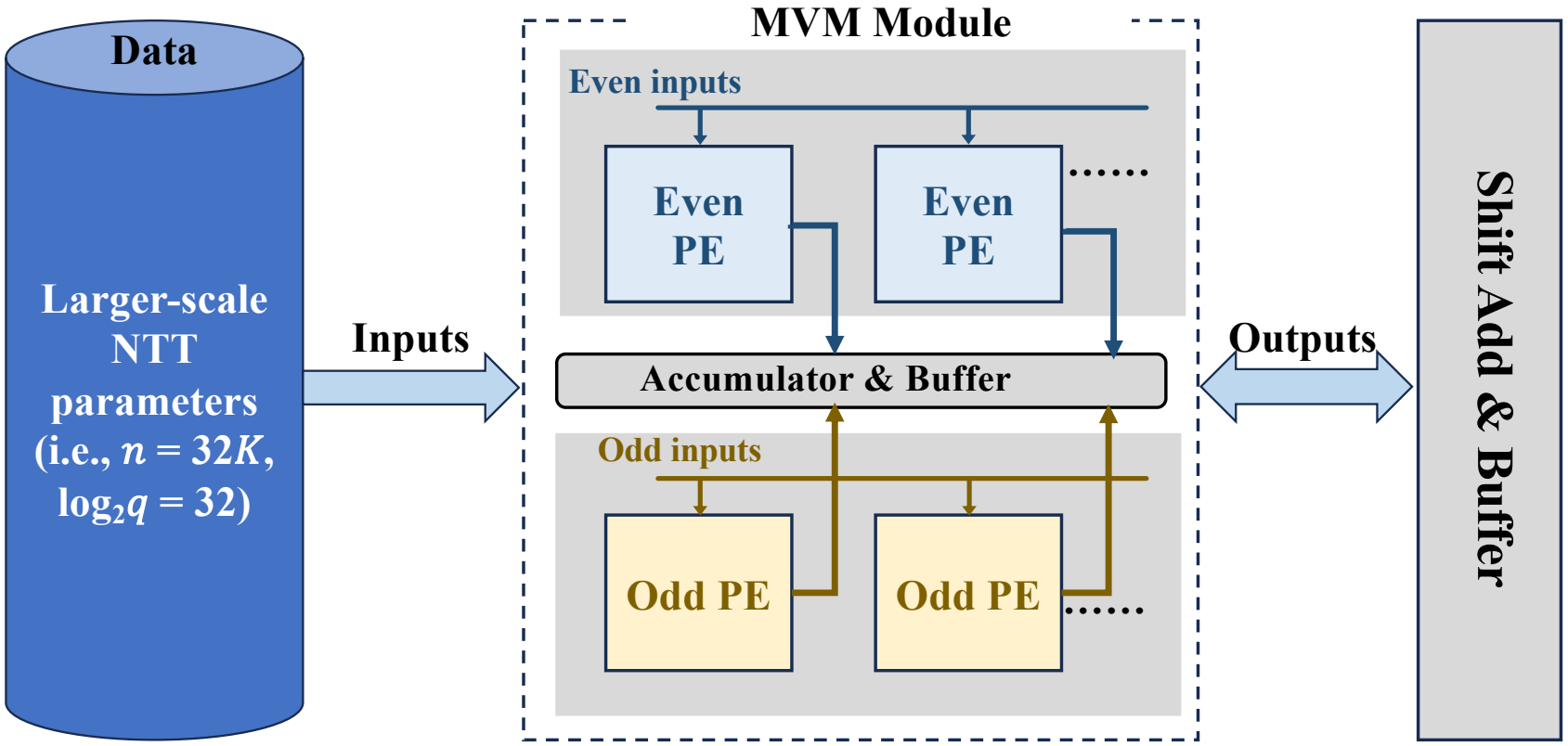
$$A_i = \sum_{j=0}^{n-1} \omega_i^j a_j \bmod q$$

$$\begin{cases} A_k = [\sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i)}(a_{2i} + a_{2i+\frac{n}{2}}) + \sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i+1)}(a_{2i+1} + a_{2i+1+\frac{n}{2}})] \bmod q \\ A_{k+\frac{n}{2}} = [\sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i)}(a_{2i} + a_{2i+\frac{n}{2}}) - \sum_{i=0}^{\frac{n}{4}-1} \omega_k^{(2i+1)}(a_{2i+1} + a_{2i+1+\frac{n}{2}})] \bmod q \end{cases}$$

MVM Module-Reduction



Large-scale NTT Operations



- **Karatsuba multiplication algorithm**

$$a \cdot \omega = a_h \cdot \omega_h \ll 2^{log_2 q} + (a_h \cdot \omega_l + a_l \cdot \omega_h) \ll 2^{(log_2 q)/2} + a_l \cdot \omega_l$$

Mod Algorithm Optimization

- **Adapt the original Barrett algorithm to the efficient implementation on CIM**

Barrett's Mod Algorithm

Calculation: $r = c \bmod q$ (q : n bit, 2^n)
Preparation: modulus q
Standby arithmetic c

1. $mu = 2^{2n}/q$
2. $t_1 = c >> n-1$
3. $t_2 = t_1 \times mu$
4. $t_3 = t_2 >> n+1$
5. $r_1 = c \% 2^{n+1}$
6. $r_2 = (t_3 \times q) \% 2^{n+1}$
7. $r = r_1 - r_2$

Condition: r > q/2 ? (r-q) : r
Return r

Optimization

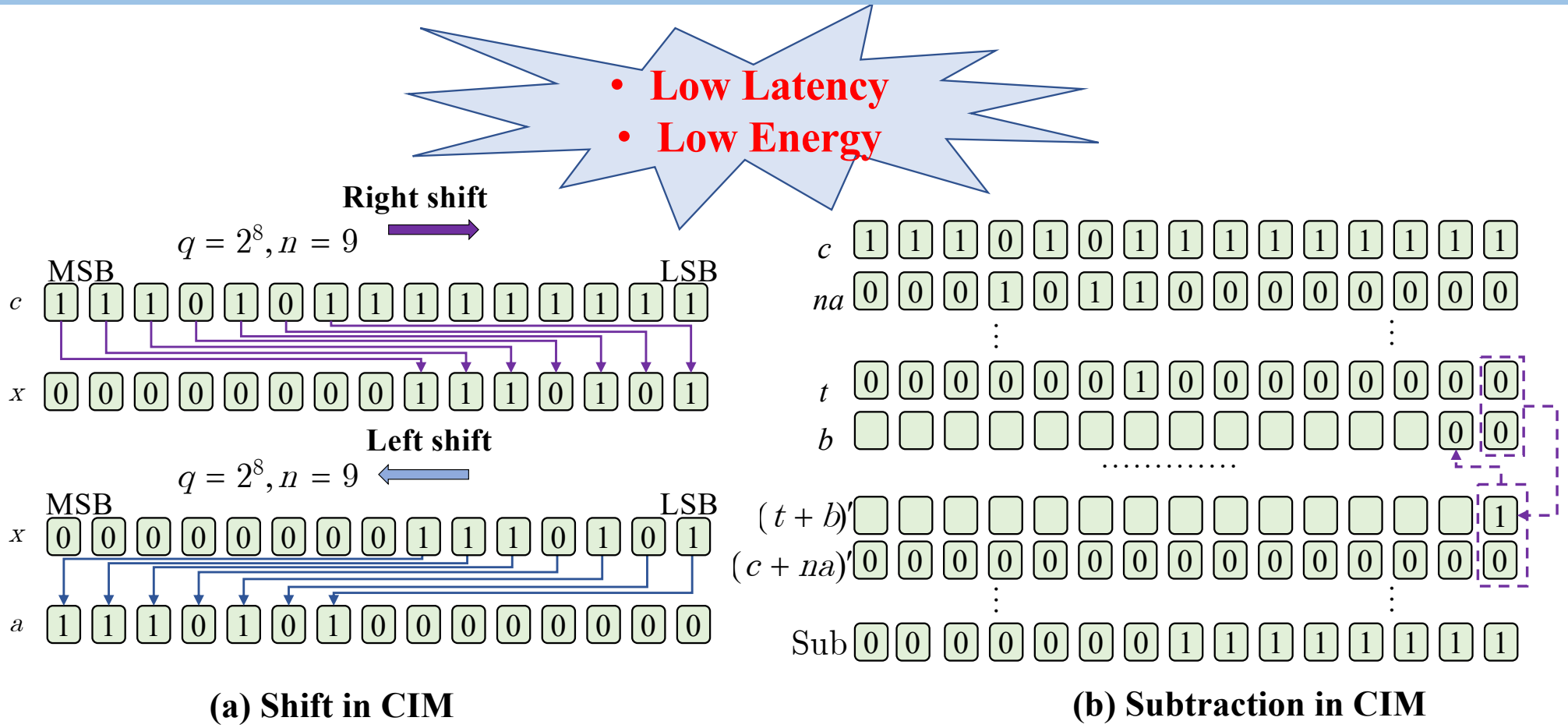
Implementing in CIM

Calculation: $r = c \bmod q$ (q : n bit)

1. x = c>>n-1;
2. a = x<<n-1;
3. r = c – a;

Condition: r > q/2 ? (r-q) : r
Return r

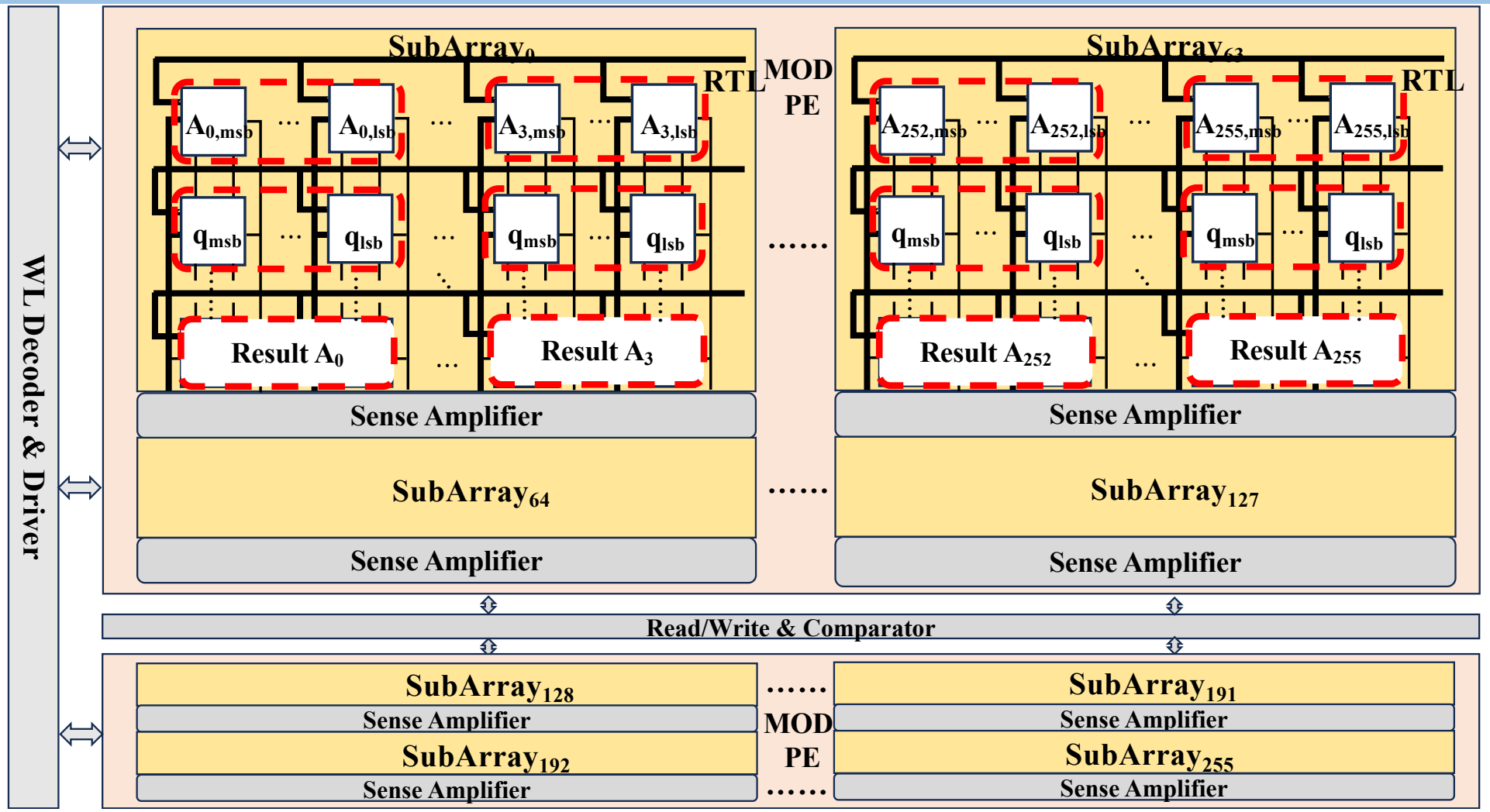
Mod Algorithm Optimization



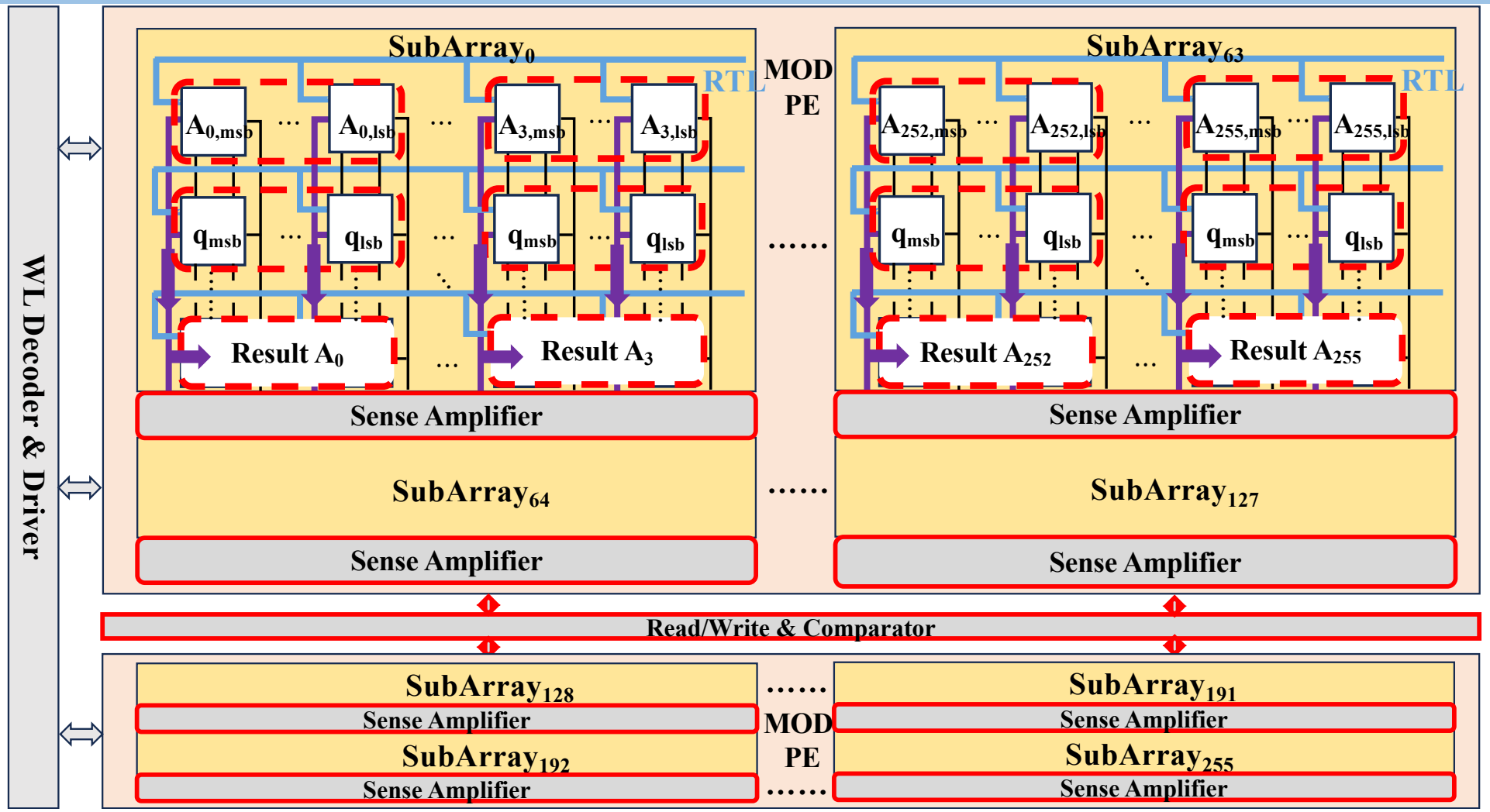
(a) Shift in CIM

(b) Subtraction in CIM

Mod Module-Data Mapping



Mod Module-Computation



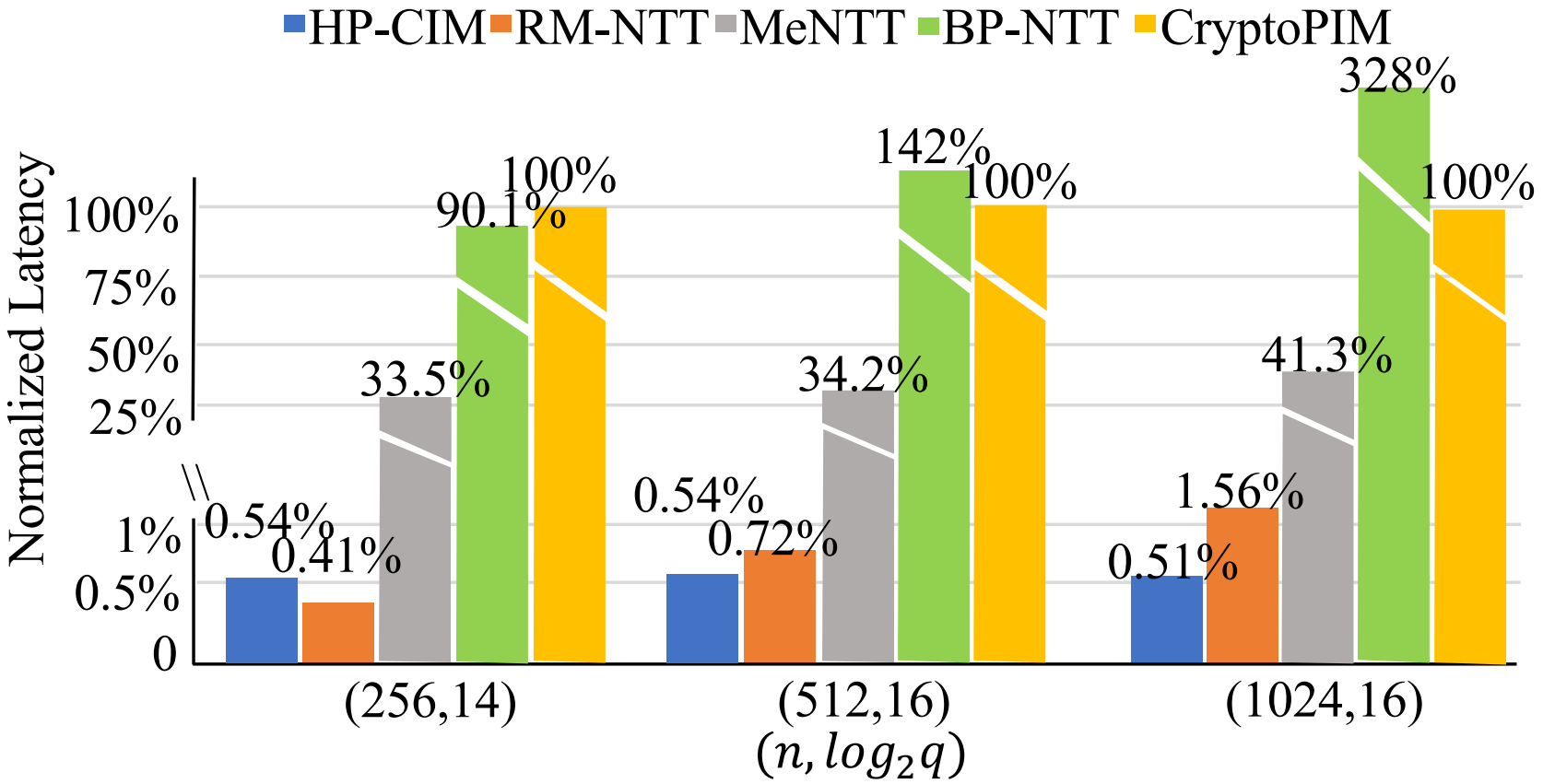
Outline

- Background & Motivation
- Proposed Method
  - Overview
  - Architecture & Data Mapping
- Evaluation and Results
- Conclusion

Evaluation Setup

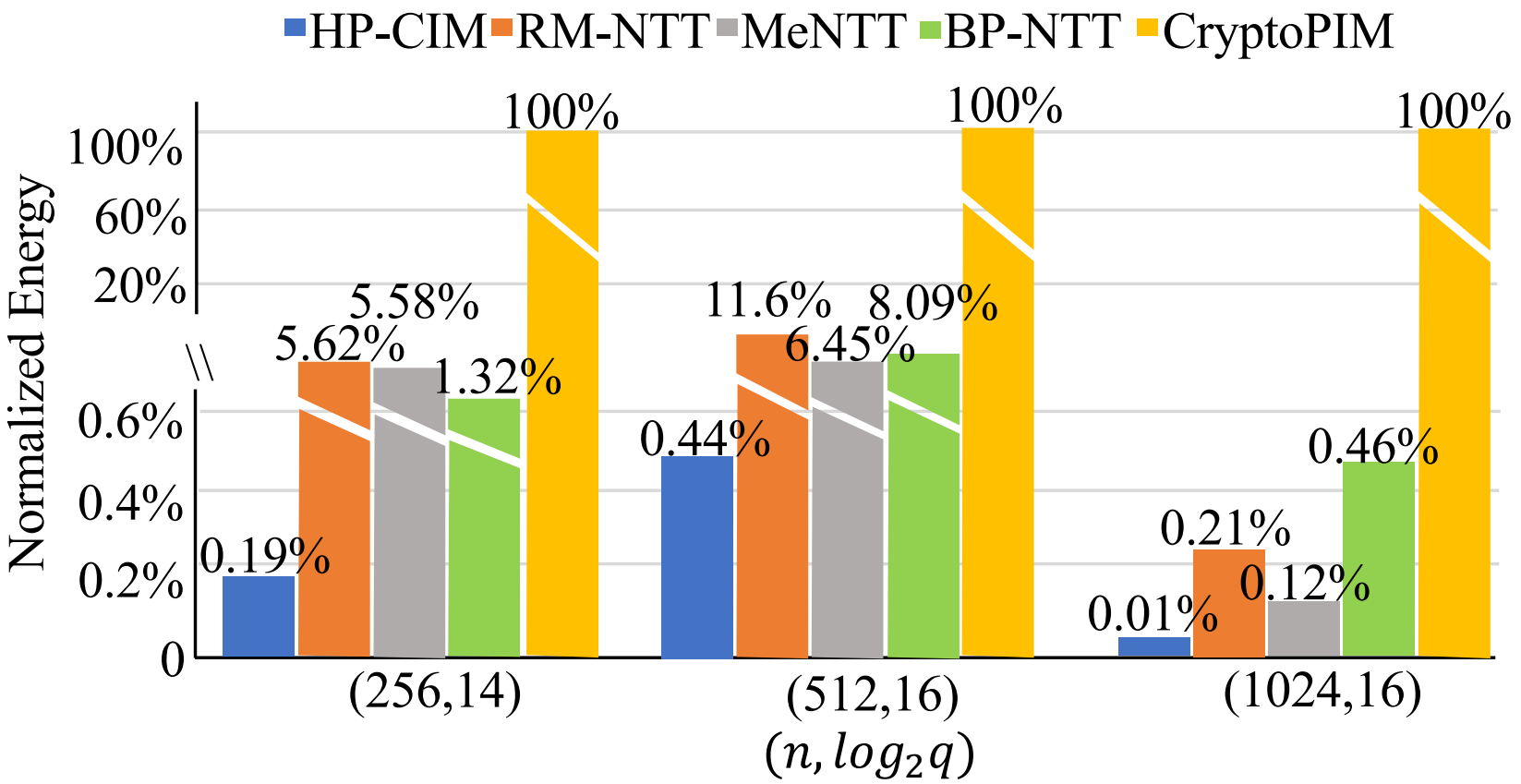
Design	Platform	Algorithm	NTT Parameters (n, $\log_2 q$)
HP-CIM(Ours)	6T SRAM	MVM	(32K,32)
BP-NTT	6T SRAM	CT Butterfly	(1024,16)
MeNTT	6T SRAM	CT Butterfly	(32K,32)
RM-NTT	ReRAM	MVM	(1024,16)
CryptoPIM (Baseline)	ReRAM	Butterfly	(32K,32)
HP-CIM Settings			
MVM Module	16 PEs, 32 KB/PE 256 SubArrays/PE, 64×16 SubArray		
MOD Module	2 PEs, 8 KB/PE 128 SubArrays/PE, 8×64 SubArray		

Evaluation Result



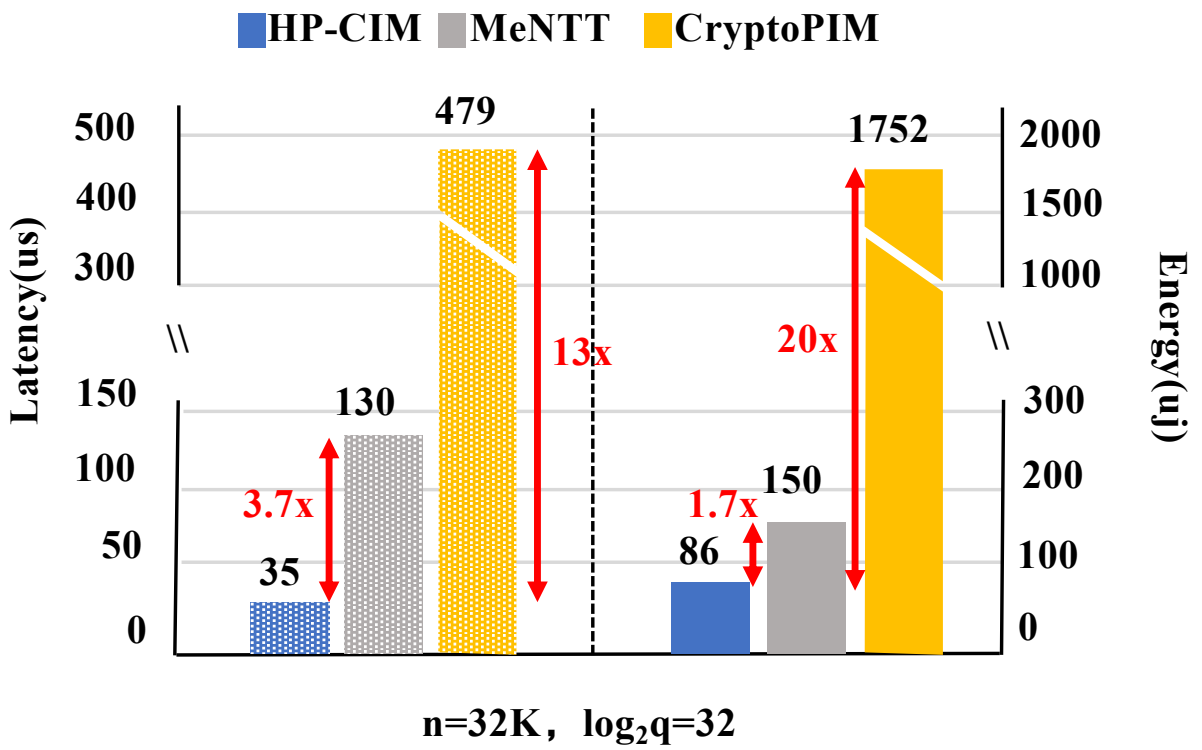
- **HP-CIM achieves a latency reduction of up to 3.08× compared to the fastest existing CIM-based NTT accelerator, RM-NTT**

Evaluation Result



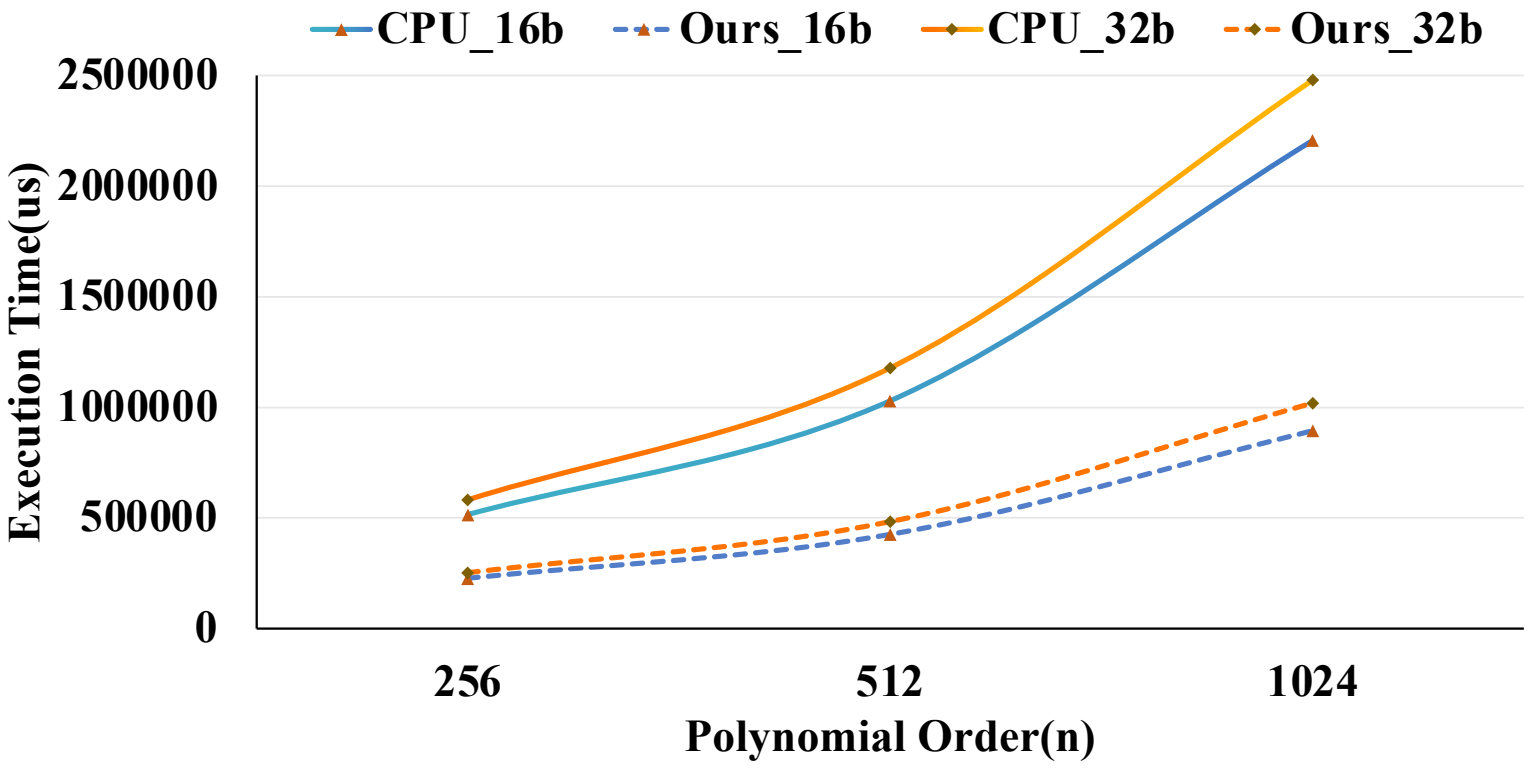
- **HP-CIM provides significant energy savings of up to 4.96× over the most energy-efficient prior solution, MeNTT**

Evaluation Result



- Under large-scale NTT parameter settings, HP-CIM outperforms other designs in terms of latency and energy

Evaluation Result



- HP-CIM reduces execution time by over 2.4× compared to CPU

Conclusion

1. High Parallelism with Hierarchical SRAM Architecture

- **Introduced a digital SRAM-based CIM NTT engine, utilizing a hierarchical structure to achieve high parallelism and scalability for large-scale NTT operations.**

2. Novel Even-Odd Data Mapping Strategy

- **Proposed an even-odd data mapping approach to optimize memory utilization, enabling efficient reuse of intermediate computation results for better scalability.**

3. Integrated Mod Computation within CIM Arrays

- **Developed efficient mod operations directly within CIM arrays using SRAM read-write capabilities, eliminating the need for extra peripheral circuits and enhancing area and energy efficiency.**

4. Significant Performance and Energy Improvements

- **Achieved up to 3.08× faster execution and 4.96× energy savings compared to prior CIM-based designs, validated through extensive comparisons with state-of-the-art methods.**

THANK YOU

High-Parallel In-Memory NTT Engine with Hierarchical Structure and Even-Odd Data Mapping

Bing Li[1], Huaijun Liu[2], Yibo Du[3,4], Ying Wang[3,4]

Institute of Microelectronics, Chinese Academy of Sciences[1]
Capital Normal University[2]
Institute of Computing Technology, Chinese Academy of Sciences[3],
University of Chinese Academy Sciences[4]

Reference

[1] Gentry C. Fully homomorphic encryption using ideal lattices[C]. Proceedings of the forty-first annual ACM symposium on Theory of computing, Bethesda, Maryland, 2009: 169-178.
[2] Fan J, Vercauteren F. Somewhat Practical Fully Homomorphic Encryption[J]. IACR Cryptology ePrint Archive,2012,2012(2012):144-162.
[3] Kim S, Kim J, Kim M J, et al. Bts: An accelerator for bootstrappable fully homomorphic encryption[C]. Proceedings of the 49th Annual International Symposium on Computer Architecture, New York, 2022: 711-725.
[4] Samardzic N, Feldmann A, Krastev A, et al. F1: A fast and programmable accelerator for fully homomorphic encryption[C]. MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture, Greece, 2021: 238-252.
[5] He Y, Qu S, Lin G, et al. Processing-in-SRAM acceleration for ultra-low power visual 3D perception[C]. Proceedings of the 59th ACM/IEEE Design Automation Conference, San Francisco California, 2022: 295-300.
[6] Li D, Pakala A, Yang K. MeNTT: A compact and efficient processing-in-memory number theoretic transform (NTT) accelerator[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2022, 30(5): 579-588.
[7] Albrecht M, Chase M, Chen H, et al. Homomorphic encryption standard[J]. Protecting privacy through homomorphic encryption, 2021: 31-62.