2024 Design, Automation & Test in Europe Conference (DATE 2024)

# PhotonNTT: Energy-efficient Parallel Photonic Number Theoretic Transform Accelerator

Xianbin Li[1,†], Jiaqi Liu[1,†], Yuying Zhang[1], Yinyi Liu[1], Jiaxu Zhang[1], Chengeng Li[1],
Shixi Chen[1], Yuxiang Fu[1], Fengshi Tian[1], Wei Zhang[1], Jiang Xu[1,2,*]
[1] *Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology*
[2] *Microelectronics Thrust, The Hong Kong University of Science and Technology (GZ)*

***Abstract*—Fully homomorphic encryption (FHE) presents a promising opportunity to remove privacy barriers in various scenarios including cloud computing and secure database search, by enabling computation on encrypted data. However, integrating FHE with real-world applications remains challenging due to its significant computational overhead. In the FHE scheme, Number Theoretic Transform (NTT) consumes the primary computing resources and has great potential for acceleration.**

***For the first time*, we present a photonic NTT accelerator, PhotonNTT, with high energy efficiency and parallelism to address the above challenge. Our approach involves formulating the NTT into matrix-vector multiplication (MVM) operations and mapping the data flow into parallel photonic MVM units. A dedicated data mapping scheme is proposed to introduce free spectral range (FSR) and distributed RAM design into the system, which enables a high bit-wise parallelism level. The system's reliability is validated through the Monte-Carlo BER analysis. The experimental evaluation shows that the proposed architecture outperforms SOTA CiM-based NTT accelerators with an improvement of 50x in throughput and 63x improvement in energy efficiency.**

## I. Introduction

With the growing awareness of the importance attributed to data privacy and integrity, there has been an increasing focus on privacy-preserving computing paradigms. Fully homomorphic encryption (FHE) is a form of encryption that allows computations to be performed in an encrypted form without access to the secret key. This ensures the confidentiality of sensitive information, even when processed on an untrusted server, thereby paving the way for its adoption in domains such as finance, healthcare, and national security scenarios [1], [2].

Nevertheless, the practicality of FHE remains limited for most applications due to its substantial overhead, encompassing computation time and memory usage, particularly in instances where low-latency and real-time processing are crucial. Even with a highly optimized FHE library, applications incur a computation time increase of approximately 5-6 orders of magnitude when operated encryptedly [3]. Moreover, employing an FHE-encrypted scheme for a simple neural network application like MNIST would necessitate approximately 120× more memory [4]. Consequently, there is an urgent need for performance enhancements and data movement optimizations to enable the practical implementation of FHE schemes.

Within FHE schemes, the Number Theoretic Transform (NTT) assumes a crucial role in polynomial multiplication and accounts for a significant portion of computing resources throughout the entire FHE process. For instance, it constitutes 51% of the execution time of ciphertext multiplication and 55% of the HE ResNet-50 inference time [5], [6]. Multiple hardware acceleration platforms, including FPGA, ASIC and Compute-in-Memory (CiM), have been proposed to address this challenge. However, the acceleration ratio is still limited.

Optical domain-specific accelerators have attracted significant attention due to their superb high bandwidth, high parallelism, and low latency. Several optical accelerators have been proposed for neural network applications, including CNNs [7], [8], SNN [9] and transformer-based models [10], which have been meticulously optimized. In fact, they exhibit remarkable performance for a wide range of applications that crave high throughput, bringing significant potential to NTT acceleration.

*For the first time*, we present a photonic accelerator for NTT, namely PhotonNTT, which possesses both energy efficiency and real-time capabilities. In essence, we employ a microring (MR) crossbar array as a fundamental functional unit for large-scale MVM operations. To elevate the system's performance, we incorporate a bit-slicing scheme that exploits free spectral range (FSR) parallelism. Furthermore, a distributed RAM design is integrated to mitigate the substantial overhead associated with data movement. We conducted a comprehensive evaluation of the system's throughput and energy efficiency, revealing a remarkable improvement of $50\times$ and $63\times$, respectively, compared to SOTA CiM-based NTT accelerators. Additionally, we conducted an exploration of the bit-error rate (BER) to validate the system's reliability.

## II. Background and Related Works

### A. Number Theoretic Transform (NTT)

As a generalization of FFT, NTT is defined on integer operation in a finite field by modulo $q$. For a polynomial $a = \sum_{i=0}^{n-1} a_i X^i$, the n-point NTT is defined as $\tilde{a}_i = \sum_{j=0}^{n-1} w^{i\times j} a_j$, where $w$ is a primitive $n$-th root of unity $w^n = 1 \bmod q$, and the term $w^{i\times j}$ represents twiddle factors. To multiply two polynomials $a$ and $s$, we first convert them into the NTT form, and then only point-wise multiplications and inverse NTT (INTT) are required, i.e. $b = \mathrm{INTT}(\tilde{a}\cdot\tilde{s})$. This significantly reduces the overall time complexity.

A common way to implement NTT is to utilize the Divide-and-Conquer principle as FFT, which is widely known as

[†] Xianbin Li and Jiaqi Liu are co-first authors.
[*] Corresponding author: jiang.xu@ust.hk.

979-8-3503-4859-0/DATE24/© 2024 EDAA

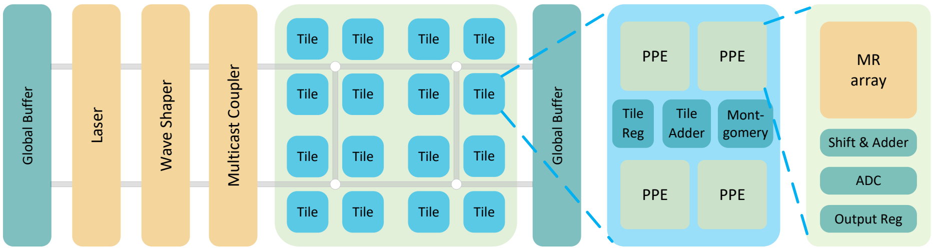


Fig. 1: System schematic

butterfly optimizations. The Cooley-Turkey and Gentleman-Sande algorithms reduce the complexity of NTT from $O(N^2)$ to $O(N \log N)$. Another approach is to map the twiddle factors into a matrix and perform Matrix-Vector Multiplication (MVM) operations, which is adopted in PhotoNTT, as the inherent parallelism of optics can enhance system performance while minimizing data movement overhead in this context.

### B. NTT Accelerators

Nejatollahi *et al.* [11] employs in-memory bit-wise operations in RRAM to implement Gentleman-Sande butterfly and Shift-Add-based reduction algorithms, which is the first RRAM-based NTT accelerator. Though their utilization of an unfolded dataflow structure achieves substantial throughput with a high density of RRAM cells, the acceleration ratios remain to be polished. Park *et al.* [12] propose a VMM-based RRAM NTT accelerator that employs a modified Montgomery reduction algorithm to convert the results instead of using FFT-like algorithms to improve the execution speed. Unfortunately, their design only supports polynomial orders up to 1k because they cannot reprogram the RRAM arrays to load larger twiddle factor matrices, which is expensive and time-consuming. Li *et al.* [13] propose MeNTT with bit-serial modular addition, subtraction, and multiplication using 6T-SRAM arrays. Although this approach can operate $n/2$ butterflies in parallel with the SRAM arrays, the FFT-like algorithm still requires $log_2 n$ serial execution stages that impede the speed of NTT.

In addition to CiM accelerators, there exist NTT implementations based on ASIC and FPGA. Banerjee *et al.* [14] propose a reconfigurable cryptographic processor featuring a modular arithmetic core where each polynomial is split among 4 single-port RAMs to reduce the area overhead of storing polynomials. Song *et al.* [15] construct a three-stage configurable NTT core that balances local storage and global routing to achieve a nearly optimal trade-off between latency and size. Except for ASIC-based design, Zhang *et al.* [16] propose a five-stage pipeline butterfly arithmetic unit to reduce critical path delay and employs a ping-pong memory access scheme that enables reading and writing two coefficients in 1 cycle by using two BRAMs in an FPGA platform. Other works [**?**], [17] propose customized data flow and optimized memory access strategy to accelerate NTT applications. However, these approaches suffer from frequent data movement between processing elements and memory, thereby impeding performance and energy efficiency. Additionally, their applicability is confined to small polynomial orders and limited parameter configurations, preventing them from FHE application.

### C. Domain-specific Photonic Accelerator

Given the current state of general computation cores, i.e. CPUs and GPUs, nearing the limits of Moore's Law, there is a promising opportunity to develop domain-specific photonic accelerators that harness the inherent advantages of high bandwidth, massive parallelism, and low latency offered by optics. Optical modulators, for example, exhibit operation speeds reaching tens of GHz while maintaining high energy efficiency. Furthermore, by leveraging Dense Wavelength Division Multiplexing (DWDM), the typical bandwidth of optical telecommunication C-band can extend to several THz, enabling higher throughput and increased computational capacity in accelerators. Optical matrix multiplication accelerators, predominantly in the form of optical neural network accelerators [18], have garnered significant attention. Foe example, Shen *et al.* proposed to accelerate neural networks [19] with MZI-arrays. Subsequently, ONNs using silicon weight banks are proposed to implement CNNs [20], [21]. To broaden the scope of acceleration scenarios, optical General Matrix Multiplication (GEMM) accelerators also emerge as a viable choice [22], [23].

In particular, FHE schemes requires NTT with immense parameters, resulting in a substantial computational intensity. In line with the growing trend of privacy-preserving scenarios, the development of real-time photonic NTT accelerators holds great promise.

## III. Architecture

The overall architecture schematic of PhotonNTT, depicted in Figure 1, consists of a global buffer and multiple photonic acceleration tiles. The global buffer serves as the storage for both input and output from different tiles. Each tile comprises four Photonic Processing Elements (PPEs) that perform (I)NTT operations in parallel. The intermediate MVM results obtained from PPEs will be post-processed by digital circuits and then buffered in the tile register for next-step operation.

Each PPE comprises a 16×16 MR crossbar array for MVM execution. A dedicated FSR-based parallelism, as depicted in Fig 2b, is adopted to patch several bits into one batch as input. This bit-wise parallelism, together with the original row-wise and column-wise parallelism, contributes to a significant performance enhancement.

### A. Dot-product in MR

In an MR modulator, only the light with the proper wavelength could transmit to the drop port. The resonant wavelength is determined by the radius of the ring and could be further modulated using thermal or electrical tuning. This provides a way to conduct optical dot products at a low cost. By tuning the resonant wavelength finely, the output in the drop port could be

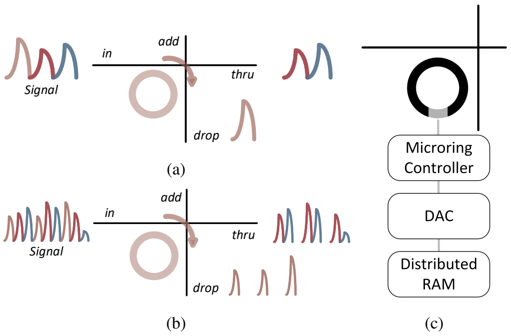


Fig. 2: (a,b) MR modulator as dot-product unit (with FSR-parallelism); (c) distributed RAM as local buffer

modulated at 4 or even 8-bit. In this case, we strike a balance between precision, speed, and device variation, opting for a 4-bit configuration. This is yet to be enough for NTT algorithms which typically require 16-bit or higher numbers. A bit-slicing algorithm is induced to address the issue in Sec III-D.

To leverage the expansive bandwidth of MR to its full potential, we incorporate the free spectral range (FSR) parallelism. As shown in the transmission pattern of MR in Fig.2b, light at various wavelengths could transmit through the drop port, whose intervals are identical, namely FSR, decided by $\Delta\lambda = -\frac{\lambda^2}{2\pi n_g R}$. Consequently, parallel operations of multiple input bits within a single MR become possible, constituting a cohesive unit referred to as a patch.

FSR parallelism suffers from the fact that the weight of the multiplications should remain identical, which hinders its adoption in other accelerators. However, within the mapping scheme of NTT, wherein twiddle factors stay fixed, FSR parallelism operates effectively.

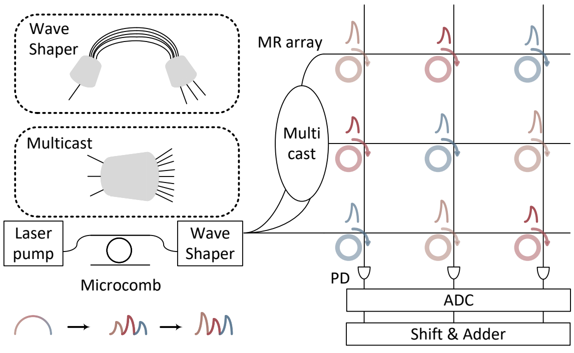


Fig. 3: Photonic Processing Element (PPE)

### B. Photonic MVM Unit

Fig 3 illustrates the operational principle of PPE. The Laser pump generates a broadband primitive signal, which is subsequently modulated by the Microcomb. The modulated signal is then directed to the wave shaper, where the intensity of each individual wavelength is fine-tuned. Then this signal would be multicast to various rows of the MR array, which serves as the dot-product unit. Addition naturally occurs when the multitude of multiplication outcomes gather in PDs, thereby constituting the complete MVM function unit.

Efforts are made to facilitate the incorporation of FSR parallelism. Extra photodiodes (PDs) are required at each column for wavelength de-multiplexing. To reduce the energy and area overhead of ADC usage, different columns use a MUX for ADC sharing, which transfers data into the digital domain.

### C. Data Flow, Mapping and the distributed RAM

In PhotonNTT, we devote to the full utilization of parallelism for NTT operation in the photonic platform. Hence, the MVM NTT mapping is adopted rather than the butterfly optimization, which prevents exclusive data movement and complex routing in a photonic implementation.

The NTT operation could be identified as

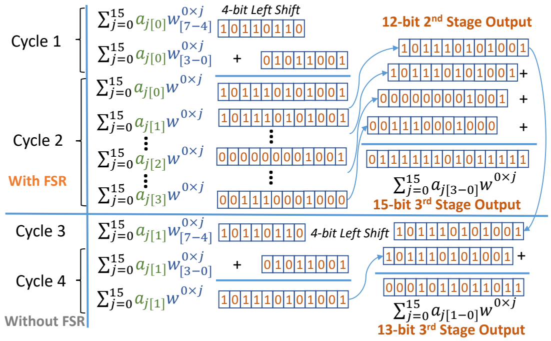
$$\begin{bmatrix} \tilde{a}_0 \\ \tilde{a}_1 \\ \vdots \\ \tilde{a}_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \cdots & \omega^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \cdots & \omega^{(n-1)^2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

For twiddle factors, which remain stationary throughout the operation, pre-calculation makes it available for storage in the distributed RAM. To enable the large-polynomial-order NTT, it is necessary to partition the workload into smaller tiles. Here, the $n \times n$ twiddle factor matrix is much larger than the total size of PPE at $p \times p$. Therefore, we tile this matrix into $[\frac{n}{p}]^2$ pieces and map one tile to the PPE each time.

To alleviate the pressure on memory access brought by distinguished twiddle factors, a distributed RAM is designed for each MR to store temporary twiddle factors as illustrated in Fig 2c. This allows each MR to access one entry of the distributed RAM to retrieve the value of the twiddle factor, rather than accessing the global buffer.

On the other hand, the large polynomial coefficients are multicast to the MR crossbar array bit by bit, due to a reliability concern. With the assistance of the bit-slicing scheme mentioned later in Sec III-D, intermediate results get aggregated and prepared for next-step computation. Specifically, with the dedicated FSR-parallelism, a reduced tiling scheme can fulfill the requirement of multi-bit input.

### D. Bit-slicing FSR-based Parallelism



Cycle 1
Cycle 2
With FSR
Cycle 3
Cycle 4
Without FSR
$\sum_{j=0}^{15} a_{j[0]} w_{[7-4]}^{0\times j}$
4-bit Left Shift
10110110
$\sum_{j=0}^{15} a_{j[0]} w_{[3-0]}^{0\times j}$
+ 01011001
$\sum_{j=0}^{15} a_{j[0]} w^{0\times j}$
101110101001
$\sum_{j=0}^{15} a_{j[1]} w^{0\times j}$
101110101001
$\sum_{j=0}^{15} a_{j[2]} w^{0\times j}$
000000001001
$\sum_{j=0}^{15} a_{j[3]} w^{0\times j}$
001110001000
12-bit 2nd Stage Output
101110101001
101110101001 +
000000001001 +
001110001000 +
011111101011111
$\sum_{j=0}^{15} a_{j[3-0]} w^{0\times j}$
15-bit 3rd Stage Output
$\sum_{j=0}^{15} a_{j[1]} w_{[7-4]}^{0\times j}$
10110110
4-bit Left Shift
101110101001
$\sum_{j=0}^{15} a_{j[1]} w_{[3-0]}^{0\times j}$
+ 01011001
$\sum_{j=0}^{15} a_{j[1]} w^{0\times j}$
101110101001
101110101001 +
0001011011011
$\sum_{j=0}^{15} a_{j[1-0]} w^{0\times j}$
13-bit 3rd Stage Output

Fig. 4: FSR-enabled Bit-slicing Scheme

As described in Fig 4, with the FSR-parallelism level at 4, the 16-bit input is sliced to 4-bit input batches and fed into the array simultaneously, where multiplications with twiddle factor occur. Subsequently, after results get demultiplexed and reach the PD, the FSR-enabled bit-slicing scheme significantly reduces system complexity and latency, which is detailed in Algorithm 1.

In the first stage, every 8-bit partial product $\sum_{j=0}^{15} a_{j[0]} w_{[7-4]}^{0\times j}$ from each column's PD is sampled by the 8-bit ADC. Then the output of different channels in one column is shifted by 1, 2, and 3 respectively, and gets summed, which gets

$\sum_{j=0}^{15} a_{j[3-0]} w_{[7-4]}^{0\times j}$ in each column. Then the result is left shifted by 4 to add with $\sum_{j=0}^{15} a_{j[3-0]} w_{[3-0]}^{0\times j}$ in another column. After iteration across all the columns, the result of $\sum_{j=0}^{15} a_{j[3-0]} w^{0\times j}$ is aggregated. Then the input changes to $a_{j[7-4]}$ as the next input batch. After another 3 input batches, the output is aggregated to $\sum_{j=0}^{15} a_j w^{0\times j}$.

**Algorithm 1** Bit-slicing Shift-Add Method

```
Input: An integer $t_0 = \sum_{j=0}^{15} a_{j[4k+m]} \bar{w}_{i[4n+3:4n]}$, where $t_0$ is from ADC output, and k,n $\in${0,1,...7}.
Output: An integer $t_3 = \sum_{j=0}^{15} a_{j[31:0]} \bar{w}_{i[31:0]}$.
$t_1 \leftarrow 0$, $t_2 \leftarrow 0$, $t_3 \leftarrow 0$, $FSR \leftarrow 4$
for k=0,k<8,k++ do
    for n=0,n<8,n++ do
        $t_1 \leftarrow \sum_{m=0}^{FSR-1}((\sum_{j=0}^{15} a_{j[4k+m]} \bar{w}_{i[4n+3:4n]}) << m)$
        $t_1 \leftarrow \sum_{j=0}^{15} a_{j[4k+FSR-1:4k]} \bar{w}_{i[4n+3:4n]}$
        $t_2 \leftarrow t_2 + (t_1 << 4n)$
    end for
    $t_3 \leftarrow t_3 + (t_2 << 4k)$
end for
Return $t_3$
```

### E. Montgomery Reduction Unit

After aggregation, the result may end with its bit-width larger than the original size. In this case, modular reduction is required to reduce the number to a proper size for next-step computation.

In order to use the MVM result as an operand, modification of the general modular reduction algorithm is necessary. This involves adapting the Montgomery reduction algorithm, which is capable of handling operands larger than two-integer multiplication. The resulting modified Montgomery reduction algorithm [12] is presented in Algorithm 2. To transfer the twiddle factors to the Montgomery space, we pre-multiply them by $n^{-1}$ and $r$, where $\bar{w}^{i\times j} = n^{-1} w^{i\times j} r \bmod q$. To simplify the logic computations, we chose $r = n^2 2^k$, where $k$ is a power of 2. This choice allows modulo $r$ to be selected by the lower bits of the integer, and division by $r$ to be performed using right shift operations.

**Algorithm 2** Montgomery Reduction Algorithm for (I)NTT

```
Input: An integer $T = \sum_{i=0}^{n-1} a_i \bar{w}_i$, where $a_i$ and $\bar{w}_i \in Z_q, \bar{w} = wn^{-1} r \bmod q$
Output: An integer $t = Tnr^{-1} \bmod q$ for NTT, or $t = Tr^{-1} \bmod q$ for INTT.
$r = n^2 2^k$, where $2^{k-1} < q < 2^k$
$q' = (rr^{-1} - 1)/q$, where $rr^{-1} \bmod q = 1$
if NTT then
    $T \leftarrow T << \log_2 n$
end if
$m \leftarrow q'(T \bmod r) \bmod r; z \leftarrow (T + mq)/r$
if $z \geq q$ then $t \leftarrow z - q$
else $t \leftarrow z$
end if
Return t
```

## IV. Experimental Evaluation

In this section, we conduct comparisons between PhotonNTT and other SOTA accelerators, as well as design space exploration on different device and hardware parameter settings. Moreover, the BER analysis ensures system reliability.

### A. Evaluation Setup

As depicted in Table I, the MR array has a default size of $16\times16$ with 4-bit modulation precision. The functionality of the MR array is evaluated using the photonic device modeling and simulation environment BOSIM [24]. Montgomery reduction unit, shifter, and adder are simulated and verified by verilog code and synthesized using Synopsys Design Compiler with 40nm FreePDK. Memory subsystems were simulated using the FN-CACTI tool [25], which is the latest extension of the CACTI [26] cache modeling tool for FinFET and recent CMOS devices.

We perform the evaluation of the latency, energy consumption, throughput, and area efficiency of PhotonNTT under polynomial orders at 256 and 1024, and compare the result with SOTA CiM, ASIC, and FPGA NTT accelerators. As proof of scalability, we evaluate the energy and latency of PhotonNTT under larger polynomial orders and compare them with CPU and RRAM-based accelerators.

TABLE I: Accelerator Configurations for n=256

| Component | Parameter | Spec | Power (W) | Area (mm$^2$) |
|---|---|---|---|---|
| MR Array [27] | Number<br>Size<br>Frequency | 256<br>16×16<br>200 MHz | 0.066 | 26.21 |
| Laser [28] | Number | 16 | 0.16 | 1.92 |
| Array Waveguide | Number | 1 per chip | - | 2 |
| SOA | Number | 256 | 1.28 | $4.35\times10^{-5}$ |
| PD [29] | Frequency<br>Number | 10GHz<br>64 per array | 0.16 | 0.16 |
| ADC [30] | Resolution<br>Number<br>Frequency | 8 bits<br>1024<br>10GHz | 15.16 | 2.92 |
| DAC [31] | Resolution<br>Number<br>Frequency | 4 bits<br>256×256<br>200MHz | 7.86 | 4.59 |
| Mont- Unit | Number | 256 | 0.43 | 0.36 |
| Shifter & Adder | Number | 256 | 2.18 | 0.65 |
| **Total** | | | **27.3** | **38.81** |

### B. Latency Comparison

Our proposed design exhibits significantly lower latency compared to CiM NTT accelerators. This is because the in-memory computing paradigm typically involves bit-serial multiplication, which is inefficient and time-consuming. In contrast, our design incorporates a direct analog matrix-vector multiplication method that reduces the latency of the multiplication of coefficients and twiddle factors. Moreover, the photonic MVM unit enables operation at an ultra-high frequency of 10GHz, outperforming all existing electronic-based accelerators. This reduces the latency by at least one order of magnitude. Additionally, our MVM-based dataflow obviates the latency of $log_2 n$ serial execution stages in FFT-like NTT algorithms [11], [13], enabling fully parallel exploitation of NTT operations. As a result, our design surpasses other SOTA accelerators by 2 to 4 orders of magnitude in terms of overall latency and throughput.

### C. Energy & Area Efficiency

As presented in Table II, PhotonNTT reaches the highest area efficiency (throughput-per-area-per-energy) compared to

TABLE II: Comparison with other NTT accelerators

| Design | Platform | n | Bitwidth | Frequency (MHz) | Latency (ns) | Energy (nJ) | Throughput (kNTT/s) | Energy Efficiency (kNTT/s/nJ) | Area Efficiency (kNTT/s/mm$^2$/nJ) |
|---|---|---|---|---|---|---|---|---|---|
| Our Work | Photonic | 256<br>1024 | 14 | 10K | **5.6** | 74.6<br>1193.3 | **178571.4** | **2393.7**<br>**149** | 72<br>0.366 |
| RM-NTT [12] | RRAM | 256<br>1024 | 14 | 400 | 280<br>1100 | 145<br>1160 | 3570<br>909 | 24.6<br>0.783 | 85.1<br>2.7 |
| MeNTT [13] | SRAM | 256<br>1024 | 14 | 151 | 23K<br>29K | 144<br>720 | 42<br>35 | 0.291<br>0.0486 | 0.81<br>0.135 |
| BP-NTT [32] | SRAM | 256 | 16 | 3.8K | 61.9K | 34 | 258.6 | 7.6 | 59.4 |
| CryptoPIM [11] | RRAM | 256<br>1024 | 16 | 909 | 68.7K<br>83.1 | 2.6K<br>5.0K | 553.3 | 0.212<br>0.111 | 1.39<br>0.73 |
| Sapphire [33] | ASIC | 256 | 14 | 64 | 20.1K | 236.3 | 49.7 | 0.21 | 0.593 |
| LEIA [15] | ASIC | 256 | 14 | 267 | 600 | 44.1 | 1667 | 37.87 | 21.3 |

other existing designs. Thanks to the highly parallel photonic MVM unit, a favorable trade-off between performance and energy efficiency is achieved. Albeit RM-NTT [12] exhibits higher throughput compared to SRAM-based designs due to the use of MVM-based dataflow instead of FFT-like dataflow, the working frequency of RRAM-based accelerators is typically at tens of megahertz, which is much lower than that of photonic accelerators, resulting in a significantly lower energy area efficiency correspondingly.

### D. Scalability

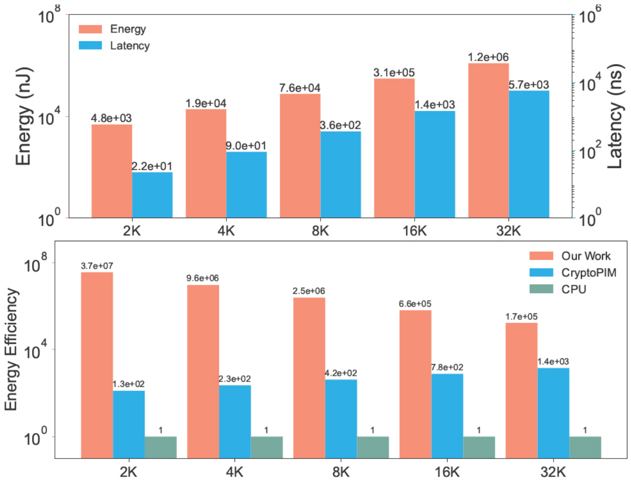


Fig. 5: Exploration of PhotonNTT's scalability: (a) Energy and latency analysis; (b) Energy efficiency comparison

For polynomial orders greater than 1K, the workload is partitioned with proper PPE reuse. The twiddle factor matrix is partitioned and reloaded from the distributed RAM for NTT implementation with larger polynomial orders. This enables a considerable scalability of the PhotonNTT, which supports up to 128K polynomials.

In comparison with CPU and RRAM-based accelerators, PhotoNTT achieves significant improvements in energy efficiency with 3 to 7 orders of magnitude for different polynomial orders. Although this advantage slightly degrades with the increase of polynomial order due to the latency and energy overhead associated with reloading and partitioning the twiddle factor matrix, there is still a considerable superiority when the polynomial order continues growing.

### E. Area and Energy Breakdown

The power consumption of the system is primarily dominated by AD conversions (ADC and DAC), which suggests the great potential of the photonic system. With an optimized dataflow mapping, chances are that energy efficiency and footprint could be further enhanced.

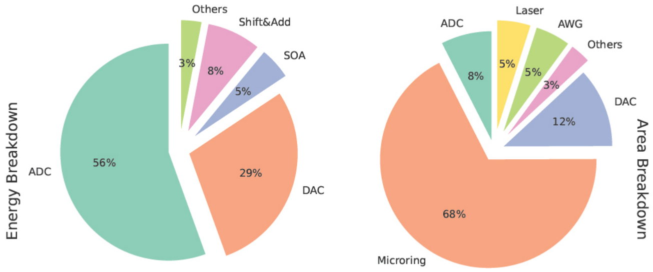


Fig. 6: Area and energy breakdown

On the other hand, the MR crossbar occupies the primary footprint. Considering the outstanding area efficiency of the PhotonNTT, we argue that it is worth reaching a higher speedup with a relatively larger area of the system.

### F. DSE on Cell Precision and Array Size

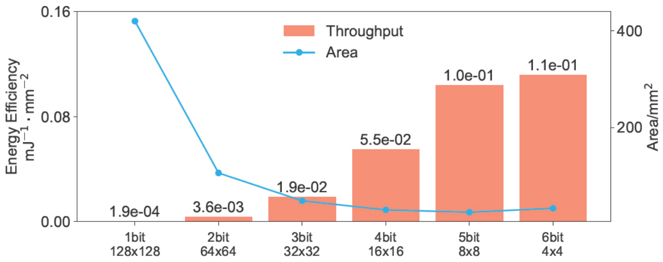


Fig. 7: Throughput efficiency and area variation with cell precision and array size

In this study, we investigate the variation of energy efficiency at different cell precision and array sizes. As illustrated in Figure 7, the trend suggests that the performance could benefit from the utilization of high-precision MRs. However, this benefit becomes marginal when it is greater than 4-bit. Additionally, as the bit precision increases and the array size decreases, the total area may increase due to the need for more arrays and ADCs to accommodate the twiddle factor matrix. Besides, a higher precision MR array is usually harder to manufacture and fabricate. Therefore, we select a 4-bit 16x16 MR array in our design to reach a good trade-off between fabrication feasibility and energy efficiency.

### G. Bit Error Rate (BER)

In NTT applications, accuracy is rather important since there is no error-tolerating mechanism. Therefore, it is necessary to validate the precision of the accelerator.

In photonNTT, several factors that might affect the BER are taken into consideration. Environmental temperature fluctuation as well as the thermal crosstalk is abstractly concluded as the thermal variation $v_t$. The difference between on and off states can also be indicated by the quality factor (Q-factor $Q$) of the MR, which measures both the sharpness of resonance and the dissipation of energy. After MAC is finished within each MR unit, the results are accumulated in the PD, where errors also accumulate size-wise. The dark current of the PD $I_d$ may further increase the chance of errors. By the way, the MR array size $s$ is also considered. In table III, we conducted a detailed survey on how these parameters would influence the BER of the PhotonNTT, and the robustness is found guaranteed.

TABLE III: BER exploration

| | $Q$ | 3500 | | | 6000 | | |
|---|---|---|---|---|---|---|---|
| $s$ | $I_d$ / $v_t$ | 1 | 10 | 100 | 1 | 10 | 100 |
| 16 × 16 | 0.01 | / | / | / | / | / | / |
| | 0.02 | / | / | / | / | / | / |
| | 0.03 | / | / | / | / | / | / |
| 32 × 32 | 0.01 | / | / | / | / | / | / |
| | 0.02 | / | / | / | / | / | / |
| | 0.03 | 1.11e-16 | 1.11e-16 | 1.33e-15 | 1.11e-16 | 1.11e-16 | 1.33e-15 |
| 64 × 64 | 0.01 | / | / | / | / | / | / |
| | 0.02 | / | / | / | / | / | / |
| | 0.03 | 3.81e-09 | 3.84e-09 | 8.00e-09 | 3.81e-09 | 3.83e-09 | 7.99e-09 |
| 128 × 128 | 0.01 | / | / | / | / | / | / |
| | 0.02 | 4.12e-10 | 4.16e-10 | 1.05e-09 | 4.11e-10 | 4.15e-10 | 1.04e-09 |
| | 0.03 | 3.09e-05 | 3.10e-05 | 3.76e-05 | 3.09e-05 | 3.10e-05 | 3.75e-05 |

It is worth noting that under the default setting, there is only neglectable BER, validating the robustness and error tolerance of the system. Even if the system scales up, chances are that the BER is confined to a reasonable range within the photonic system. Due to the overhead of OE conversion, the scale of crossbar is limited, which promises practicability even regarding fabrication variation. Furthermore, the BER result of photonNTT is much more reasonable compared to that of CiM-based NTT accelerators, mainly because the MR operates at a higher bit-width, contributing to a smaller device scale.

## V. Conclusion

*For the first time*, a photonic NTT accelerator PhotonNTT is proposed with ultra-high energy efficiency and considerable reliability. By mapping the NTT workload parallel into the MR crossbar, the inherent massive parallelism and high operating frequency of PhotoNTT enable an ultra-high throughput and energy efficiency. Additionally, the distributed RAM design facilitates scalable NTT operations. The practicality and reliability of our design are verified by a detailed BER analysis. The evaluation shows that the proposed architecture outperforms existing NTT accelerators based on ASIC and CiM, achieving an improvement of at least three orders of magnitude in area efficiency. PhotonNTT proves the outstanding potential of photonic accelerator in terms of NTT applications, which may contribute to real-time FHE hardware implementations.

## Acknowledgement

This work is supported by the Guangzhou-HKUST(GZ) Joint Funding Program (No. 2023A03J0013).

## References

[1] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *Cryptology ePrint Archive*, 2012.

[2] J. H. Cheon *et al.*, "Homomorphic encryption for arithmetic of approximate numbers," in *ASIACRYPT 2017*. Springer, 2017, pp. 409–437.

[3] J.-W. Lee *et al.*, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30 039–30 054, 2022.

[4] R. Gilad-Bachrach *et al.*, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *ICML*, 2016.

[5] S. S. Roy *et al.*, "Fpga-based high-performance parallel architecture for homomorphic computing on encrypted data," in *HPCA*, 2019.

[6] B. Reagen *et al.*, "Cheetah: Optimizing and accelerating homomorphic encryption for private inference," in *HPCA*. IEEE, 2021, pp. 26–39.

[7] X. Xu *et al.*, "11 TOPS photonic convolutional accelerator for optical neural networks," vol. 589, no. 7840, pp. 44–51.

[8] H. Bagherian *et al.*, "On-chip optical convolutional neural networks," *arXiv preprint arXiv:1808.03303*, 2018.

[9] J. Feldmann *et al.*, "All-optical spiking neurosynaptic networks with self-learning capabilities," vol. 569, no. 7755, pp. 208–214.

[10] S. Afifi *et al.*, "Tron: Transformer neural network acceleration with non-coherent silicon photonics," in *GLSVLSI*, 2023, p. 15–21.

[11] H. Nejatollahi *et al.*, "CryptoPIM: In-memory acceleration for lattice-based cryptographic hardware," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6.

[12] Y. Park *et al.*, "RM-NTT: An RRAM-based compute-in-memory number theoretic transform accelerator," vol. 8, no. 2, pp. 93–101.

[13] D. Li *et al.*, "MeNTT: A compact and efficient processing-in-memory number theoretic transform (NTT) accelerator," vol. 30, no. 5.

[14] U. Banerjee *et al.*, "2.3 an energy-efficient configurable lattice cryptography processor for the quantum-secure internet of things," in *ISSCC*.

[15] S. Song *et al.*, "LEIA: A 2.05mm2 140mw lattice encryption instruction accelerator in 40nm CMOS," in *CICC*, pp. 1–4.

[16] C. Zhang *et al.*, "Towards efficient hardware implementation of NTT for kyber on FPGAs," in *2021 ISCAS*, pp. 1–5.

[17] R. Agrawal *et al.*, "Fab: An fpga-based accelerator for bootstrappable fully homomorphic encryption," in *HPCA'23*, pp. 882–895.

[18] Y. Liu *et al.*, "PHANES: ReRAM-based photonic accelerator for deep neural networks," in *DAC '22*, pp. 103–108.

[19] Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature photonics*, vol. 11, no. 7, pp. 441–446, 2017.

[20] A. N. Tait *et al.*, "Neuromorphic photonic networks using silicon photonic weight banks," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[21] V. Bangari *et al.*, "Digital electronics and analog photonics for convolutional neural networks (deap-cnns)," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–13, 2019.

[22] N. Youngblood, "Coherent photonic crossbar arrays for large-scale matrix-matrix multiplication," *IEEE Journal of Selected Topics in Quantum Electronics*, 2022.

[23] L. Yang *et al.*, "On-chip optical matrix-vector multiplier," in *Optics and Photonics for Information Processing Vii*, vol. 8855. SPIE, 2013, pp. 100–104.

[24] X. Chen *et al.*, "Modeling and analysis of optical modulators based on free-carrier plasma dispersion effect," *TCAD*, pp. 977–990, 2019.

[25] D. P. Ravipati *et al.*, "FN-CACTI: Advanced CACTI for FinFET and NC-FinFET technologies," vol. 30, no. 3, pp. 339–352.

[26] R. Balasubramonian *et al.*, "CACTI 7: New tools for interconnect exploration in innovative off-chip memories," vol. 14, no. 2.

[27] R. Dubé-Demers *et al.*, "Ultrafast pulse-amplitude modulation with a femtojoule silicon photonic modulator," vol. 3, no. 6, pp. 622–627.

[28] A. Descos *et al.*, "Heterogeneously integrated III-v/si distributed bragg reflector laser with adiabatic coupling," in *ECOC*, pp. 1–3.

[29] P. Ma *et al.*, "Plasmonically enhanced graphene photodetector featuring 100 gbit/s data reception, high responsivity, and compact size," vol. 6, no. 1, pp. 154–161.

[30] J. Liu *et al.*, "A 10gs/s 8b 25fj/c-s 2850um2 two-step time-domain ADC using delay-tracking pipelined-SAR TDC with 500fs time step in 14nm CMOS technology," in *ISSCC*, vol. 65, pp. 160–162.

[31] H. Eslahi *et al.*, "Ultra compact and linear 4-bit digital-to-analog converter in 22nm FDSOI technology," in *ISCAS*, pp. 2778–2781.

[32] J. Zhang *et al.*, "BP-NTT: Fast and compact in-SRAM number theoretic transform with bit-parallel modular multiplication."

[33] U. Banerjee *et al.*, "Sapphire: A configurable crypto-processor for post-quantum lattice-based protocols," pp. 17–61.