Residue number system arithmetic based on integrated nanophotonics

Jiaxin Peng,¹ Shuai Sun,¹ Vikram K. Narayana,² Volker J. Sorge,¹,* and Tarek El-Ghazawi¹

¹Department of Electrical and Computer Engineering, The George Washington University, 800 22nd St. NW, Washington, DC 20052, USA
²Intel Corporation, 3600 Juliette Ln., Santa Clara, California 95054, USA
*Corresponding author: sorger@gwu.edu

Received 29 January 2018; revised 21 March 2018; accepted 24 March 2018; posted 26 March 2018 (Doc. ID 319723); published 20 April 2018

The residue number system (RNS) enables dimensionality reduction of an arithmetic problem by representing a large number as a set of smaller integers, where the number is decomposed by prime number factorization. These reduced problem sets can then be processed independently and in parallel, thus improving computational efficiency and speed. Here, we show an optical RNS hardware representation based on integrated nanophotonics. The digit-wise shifting in RNS arithmetic is expressed as spatial routing of an optical signal in 2 × 2 hybrid photonic-plasmonic switches. Here, the residue is represented by spatially shifting the input waveguides relative to the routers’ outputs, where the moduli are represented by the number of waveguides. By cascading the photonic 2 × 2 switches, we design a photonic RNS adder and a multiplier forming an all-to-all sparse directional network. The advantage of this photonic arithmetic processor is the short (10’s ps) computational execution time given by the optical propagation delay through the integrated nanophotonic router. Furthermore, we show how photonic processing in-the-network leverages the natural parallelism of optics such as wavelength-division-multiplexing in this RNS processor. A key application for such a photonic RNS engine is the functional analysis of convolutional neural networks. © 2018 Optical Society of America

OCIS codes: (200.0200) Optics in computing; (200.6715) Switching; (250.5403) Plasmonics; (230.0250) Optoelectronics.

https://doi.org/10.1364/OL.43.002026

In the field of digital computer arithmetic, the residue number system (RNS) has shown advantages in decomposing larger integers into a set of smaller integers via independent and parallel calculations. Adopting photonics into RNS arithmetic for signal processing could benefit from: (1) the fundamental fact that a photon always has to propagate because it has a non-zero momentum, which allows for operation execution and computing while the photon propagates through a network; (2) the fast execution time which is given by the photon’s time-of-flight through the network; and (3) wavelength division multiplexing (WDM) enabling a high degree of parallel instruction execution, which is further supported by the spectrally broadband nanophotonic devices used in this work. The RNS is a fast and highly scalable technique, and as such, well suitable for high-performance computing [1]. An integer number is represented as a set of residues obtained from a set of moduli, resulting in high compute-efficiency not requiring digit-carry propagation [2]. In addition, the data movement of RNS arithmetic is synergistic to the use of optical switching (e.g., spatial routing) with high performance, such as low latency and energy overheads. Indeed, optical signaling distance communication scales more favorably than electronics in terms of data bandwidth and delay, due to the lack of capacitive parasitic [3,4]. However, the promise of optical computing has not delivered engines that outperform electronics when considering bandwidth, energy-per-bit, system size, and cost [5]. Fundamental physics, however, point towards a higher information density of photonics since multiple properties, such as amplitude, phase, orbital angular momentum, polarization, and long-range entanglement, can be utilized simultaneously [6,7]. Here, we show a hybrid photonic-plasmonic (HPP) RNS adder with an all-to-all sparse directional (ASD) structure, based on cascaded 2 × 2 switches forming a crossbar. This modulo-5 ASD RNS adder shows short (5 ps) response times, requires only a 0.1-ps propagation time per switch, has a footprint of about 200 μm², and consumes about 7 fJ/bit. Such a high-speed, low-power consumption RNS adder is a novel building block for computing-in-the-network paradigms.

The RNS uses remainders of the different moduli to describe a given number. For a single modulo Mi, the number X has a specific residue, r. If X = 96, and M = 11, then r = [96]11 = 8. Assuming there are a set of moduli {M1, M2, ..., Mn} then the corresponding residues of X are {r1, r2, ..., rm}. The moduli {M1, M2, ..., Mn} should not have a common factor with each other, and prime numbers are therefore typically chosen as moduli. In general, the RNS can represent numbers ranging from 0 to (∏i=1M Mi). It also supports basic arithmetic operations. The general representation forms in the RNS shows integer X as {r1, r2, ..., rm} [M1, M2, ..., Mn]. For example, given the moduli {11, 19, 23}, X = 96 and Y = 32 are then represented as {8,1,4}[11,19,23] and {10,13,9}[11,19,23], respectively. Within the chosen representation range, each number is unique. Since a RNS...
calculation does not require carry propagation, operations on larger numbers can be decomposed into smaller number calculations making the RNS fast and straightforward to parallelize.

The main difficulty of the RNS, relative to arithmetic operations, is the determination of the relative magnitude of the two numbers expressed. The residue code is therefore of little utility for general purpose computation, but this code has many characteristics that recommend its use for special purpose computations. In this Letter, we focus on single-residue arithmetic, but note that it can be expanded easily since RNS is a digit-wise operation.

RNS computing modules based on $2 \times 2$ optical switches have been proposed on a mesh grid network topology deploying one-hot-end coding [8]. Here, each location represents a specific number in a modulo-$M$ system, and thus such a system requires $M$ bits. Both the inputs and the outputs are represented by a single active signal. By converting the two input numbers of a RNS operation into optical signals and electrical signals separately, the results are obtained via bit-shifting. Following such a mesh grid design, earlier work presented an RNS adder and an RNS multiplier based on an all-optical-switch (AOS) [9] and micro-ring resonators (MRR) [10], respectively. However, an AOS is unfeasible due to the high energy consumption to switch a signal using the weak nonlinear optic transfer functions (e.g., bistability), and the MRR are not only bulky, but their long photon lifetime limit switching and propagation delay. As a result, mesh RNS computing units may be unfeasible for high-performance applications since they require high power and large footprints, and may be challenged in down-scaling. Therefore, we proposed an ASD design based on our previously explored $2 \times 2$ HPP ITO switches [11,12]. The essential component of the RNS adder is the $2 \times 2$ optical two-state switch featuring a bar state and a cross state. In the bar state, the light will reside in the bar arm (e.g., propagate straight through the switch), whereas in the cross state, lights are routed to the opposite (cross) arm output [Figs. 1(a) and 1(b)]. A voltage control signal applied to the switch, and directs the input light to the desired output port. Our proposed ASD RNS adder of the modulo-5 system is based on a HPP switch [Fig. 2(a)]. Each switch in our design has its own control signal. While this requires more control circuits, it enables a fewer number of required optical switches, yet a look-up table (LUT) is necessary to store the states of each optical switches. In terms of device performance, these HPP switches are beneficial over the MRR, the Mach Zehnder interferometer (MZI), and the AOS in terms of energy consumption and propagation time. The RNS addition acts like all-to-all routing. Previously, it is proved that a $M \times M$ router requires $(M - 1)^2/2$ optical switches to implement a wide variety of communication options in the router [11,13]. However, this design is still limited without self-communication (blocking). Our proposed ASD RNS adder requires two more ($(M - 1)^2/2 + 2$) optical switches ($M > 3$, and $M$ is odd) to ensure strict non-blocking communication. Several $2 \times 2$ switches are designed on a set of parallel waveguides. Here, light could propagate any possible path in the router by changing the control signal after searching the LUT. Once the states of all of the switches are set, the input light beam propagates through the desired path. Finally, photodetectors at the output ports translate the result, thus capturing the spatial position.

Based on the RNS adder, we developed a similar structure for an RNS-based multiplier [Fig. 2(b)]. When one of the factors, the multiplier, is zero, the result equals zero (i.e., no spatial shifting). Thus, there are no switches along the path for input 0. Furthermore, considering the situation that the other factor (the multiplicand) can be 0, each non-zero output has a corresponding zero output to cover this scenario.

The fundamental challenge of the aforementioned photonic devices is the weak light matter-interaction (LMI) originating from the three orders of magnitude large size discrepancy between the optical wavelength and the atomic wavefunction, leading to miniscule dipole moments. This results in 0.1-1-mm-long interaction lengths of optoelectronic devices, thus lowering the integration density [14]. However, making the photon more polaritonic (matter-like), such as in plasmonics, enables strong LMIs, hence short devices, which has positive effects on the device performance [5,15]. These positive effects of wavelength-scale active optoelectronics are
(a) low electrical capacitance, (b) short photon lifetimes, allowing rapid re-excitation of the device, and (c) high energy efficiency due to the small capacitance and driving voltage. Nevertheless, the high intrinsic ohmic loss limits the plasmonic devices for large-scale integration [3,16].

Hence, we combine the low propagation loss such as given by the silicon photonic waveguides with ultra-fast tunable material-enabled plasmonic components to alleviate the fundamental drawbacks from both sides. For instance, our hybrid photonic-plasmonic (HPP) switch can combine high LMI active optoelectronics with low-loss passive photonic elements, as we demonstrated previously [11,12]. With an index-tunable in-
titive optoelectronics with low-loss passive photonic elements, as we demonstrated previously [11,12]. With a index-tunable in-
titive optoelectronics with low-loss passive photonic elements, as we demonstrated previously [11,12]. With a index-tunable in-

Moreover, a

200 GHz of switching time and requires only 13.1 fJ per switching. They are critical for high efficient data processing.

Next, we are interested in exploring a variety of switching technologies to study performance of a RNS adder, including electrical ones and optical ones. Transistors are chosen as an electrical switching strategy. Optical switching includes an MRR, an MZI, and an AOS, in addition to our HPP switch (Table 1). Of interest is the required energy per operation, the footprint area, and the delay. The latter has two parts for optical switchings, namely optical propagation delay through the RNS engine, and the individual switch setup response time. The latter, however, is a one-time event since all switches are set simultaneously, similarly to the photodetector read-outs, thus forming the root for the known one-shot execution of RNS. Though transistors perform well in footprint and speed, the energy consumption of this model increases dramatically as the modulo size \( M \) increases, due to increasing capacitance and resistance; however, our design is controlled locally instead of row-by-row without causing this problem. The other limitation of the electrical switches is the clock speed, which is around 2–3 GHz for current CPUs, constrained by power dissipation [4], whereas our optical design could keep a high frequency without extra power dissipation. It is also worth mentioning that the optical-electrical-optical (O-E-O) signal conversion is another drawback that needs to be considered for a transistor-based mesh RNS model, if it is to be added into an optical communication network. Although our proposed ASD schematic requires electrical control circuits with LUT, the actual area needed is relatively small due to the compact transistor size (0.01 \( \mu m^2 \)) [20], comparing to that of an optical switch (0.2–20 \( \mu m^2 \)). The area of the control circuit of our exemplary modulo-5 adder is about 2.5-fold the area of the mesh grid RNS adder design. The AOS is fundamentally unfeasible given its high switching energy (Table 1). The MRR consumes less energy due to its sharp resonance (high cavity quality \( Q \)-factor), but it requires large area and has a response time scales inversely with \( Q \)-factor. The MZI switch performance falls in between those of the MRR and the AOS. The HPP switch introduced has a comparable footprint relative to the mesh grid-based MZI switch, thus resulting in a similar propagation delay and about the same setup response time. However, it only requires about 10% of the MZI’s energy consumption, which is a direct result of utilizing a stronger LMI material and optical mode comb. Using a comparison figure of merit Speed-Energy-Area Product (SEAP), the ASD RNS outperforms the MRR and the MZI by about 100 fold. As the modulo size increases, the overall performance decreases as expected, since system complexity grows.

By combining both the RNS adder and multiplier, several applications are enabled such as convolution for signal processing in neural networks, because here 90% of the calculations are the multiplication-accumulation (MAC) operation in the convolutional neural network [25]. Thus, the same weight matrix becomes one of the multiplicands corresponding to a state of 2 \( \times \) 2 switches stored in the LUT. Since MAC operations use one summand/multiplicand repetitively millions of times, a photonic RNS engine is an ideal processor for this specific calculation. Once the switch-state is set, operations execute rapidly by taking advantage of the short optical propagation delay,

\[ \text{Table 1. Comparison of Mesh Grid RNS Model}^{*} \]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>mesh RNS Model [8]</th>
<th>Scale with ( M )</th>
<th>HPP [11]</th>
<th>Scale with ( M )</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Switches</td>
<td>Trans. MRR [21]</td>
<td>M(M - 1)</td>
<td>10 ((M - 1)^2/2 + 2)</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td># of Ctrl. Circuit</td>
<td>MZI [22]</td>
<td>( M - 1 )</td>
<td>10 ((M - 1)^2/2 + 2)</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td># of Look-up Table</td>
<td>AOS [23]</td>
<td>( M(M - 1) )</td>
<td>5 ((M - 1)^2/2 + 2)</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td>Energy/op. (Jf/bit)</td>
<td>Thermal Switching</td>
<td>( 16^2/4 )</td>
<td>2 ((M - 1)^2/2 + 2)</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td>Control</td>
<td>Control</td>
<td>&lt;1</td>
<td>200 ((M - 1)^2/2 + 2)</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td>Area (( \mu m^2 ))</td>
<td>Switch Control</td>
<td>0.2 \times 3200</td>
<td>((M - 1)^2/2 + 2) ( A_{\text{device}} )</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td>Control</td>
<td>Control</td>
<td>&lt;1</td>
<td>((M - 1)^2/2 + 2) ( A_{\text{control}} )</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td>Response Time (ps)</td>
<td>5.1 [24]</td>
<td>40</td>
<td>5.1</td>
<td>Scale with ( M )</td>
</tr>
<tr>
<td>Prop. Time/Dev. (ps)</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1 ( A_{\text{LUT}} )</td>
<td>Scale with ( M )</td>
</tr>
</tbody>
</table>

*All-to-all sparse directional (ASD) RNS model (proposed) in modulo-5 adder with transistor-based electrical switch (Trans.), micro ring resonator (MRR), Mach Zehnder interferometer (MZI), all-optical switch (AOS), and hybrid photonic-plasmonic (HPP) ITO switch and their scale with modulo-M system. The energy consumption and the response time of our design are calculated based on the equivalent capacitance and resistance of HPP switch with worst-case scenario when driving the adder at maximum speed without any technologies to improve the data rate.

*Transistor switching requires additional optical-electrical-optical (O-E-O) signal conversion to satisfy full-optical communication system. With 20% quantum efficiency assumed for each conversion, the final energy consumption of a transistor-based modulo-5 RNS model is around 395 Jf/bit.
and 10's of GHz-fast photodetector and HPP switch response. However, to the best of our knowledge, the existing optical RNS adder design can calculate only one operation at a time. In optical RNS addition and multiplication, they scale with (1/M) for a single operation [9,10]. This is under-utilizing the RNS system, since most of the resources are idle. Next, we show that utilizing multiple wavelengths simultaneously (WDM) significantly increases the performance. Our HPP switch is spectrally broadband and hence non-spectrally blocking. Spectral selectivity is ensured by ring-drop filters back-end (Fig. 3). This module allows multiple operations simultaneously by allocating one modulus to one wavelength, thus increasing the system efficiency. For example, if one of the summand is 4, the other summands are (1) the same input with different wavelengths—{\( \lambda_1 \) and \( \lambda_2 \). The MRR with a photodetector recognizes that the result of both operation 1 (green) and operation 2 (blue) are 0; (2) different input summand—\( \lambda_2 \). Operation 3 “0 + 4” (purple) finally obtains a result of 4. Multiple operations can be executed at the same time given by the number of available wavelengths. WDM RNS computing units are ideal for a convolutional neural network due to the feature of MAC operations requiring one constant input (weight matrix).

Here we show an RNS adder and multiplier based on an all-to-all, non-blocking, sparse directional crossbar. The RNS arithmetic is synergistically implemented by spatial routing of light using nanophotonic 2 \( \times \) 2 switching building blocks, thus enabling a highly parallel compute engine. This one-shot programmable photonic processor utilized a extremely short execution time, only limited by the picosecond short time-of-flight through the 10's of micrometer compact optical router. Since MAC operations use one summand/multiplicand repetitively millions of times, a photonic RNS engine is well suited for convolutional neural networks by transferring the weighted matrix into the switch states.


**REFERENCES**