**Improved One-hot $2^n$-modulo Multiplier in RNS**

Seyyed Mohammad Safi\(^1\), Hadi Toofani\(^1\), Mahdi Mahmoodi\(^1\), Misagh Mohammadizadeh\(^2\)

**Abstract:** Residue number system (RNS) is a mathematical method to execute computing processes. In this method, the large operands which need more time to be calculated, convert to small ones to consume less time. In RNS, to decrease the delay time, some transistors used as a matrix to select output of operations in a transistor delay time. In this paper, by using of one-hot system in RNS, the used hardware decreased to calculate the result of $2^n$-modulo multiplication.

**Keywords:** One-hot, RNS multiplier.

1. **Background**

Residue Number Systems (RNS) are parallel weight-less number systems which are based on modular arithmetic by using of relatively prime numbers called moduli. The product of these moduli is equal to the dynamic range of the RNS. One important advantage of the RNS is that each residue digit is computed independently [1].

Recent analytical results have indicated that the One-Hot Residue (OHR) number system can reduce the delay-power product (DPP) of mathematical operation circuits when compared with those based on the binary number system [2]. Delay reduction is a consequence of the fact that addition and multiplication are performed by barrel shifters. Power reduction is due to the decreased switching activity factors of digits, which are encoded in the one-hot representation.

The recently developed OHR Number System offers just such a possibility. It possesses delay-power products which are significantly improved over the binary number system, and is ideally suited for high-speed portable applications.

In next section some of the RNS multipliers will be introduced briefly, and then the one-hot adder will be analyzed. In fourth section, the new proposed multiplication will be explained and finally, in last section, the conclusion of this paper is illustrated and the results compared with previous methods.

2. **RNS Multipliers**

Many RNS multiplier has been designed so far which are pure table-look-up multipliers, quarter square multipliers, index transform multipliers, and array multipliers [4]. In its simplest form, each modulo $m_i$ multiplier can be implemented using ROMs or an array of full adder cells. Design using the ROM approach is based on a look-up table and is excellent for smaller moduli multipliers, however it takes up a lot of area as the magnitude...
of the number goes up. The ROM size increases exponentially with the number of bits in each operand. For a multiplier employing an array of full adder cells the delay is linear with the number of bits in each operand.

3. One-Hot Adder

Modulo $m_i$ addition is performed using an $m_i \cdot m_j$ barrel shifter because it is a cyclic permutation when the operands are one-hot encoded. The degradation is due to the threshold voltage drop across the pass transistors. Consequently, level restoration circuitry must be included on the output. This topic will be discussed below. Subtraction is implemented by transposing the subtrahend wires to generate its additive inverse modulo $m_i$. Note the simplicity with which the additive inverse is formed; no active circuitry is required. Figure 1 shows an OHR adder.

![Figure 1: Modulo 5 OHRNS adder model.](image)

4. Improved Adder

In this paper, a new way proposed to increase the speed of $2^n$-modulo multiplication. In this way, the product of multiplication is achieved by new algorithm and calculation will be done by OHR. In this method, the result of multiplication calculated by some of the small sum. For better result, we use one-hot adder to calculate sum results which have one-transistor delay.

It is asserted that in $2^n$-modulo multiply we need n bit of results least important bits. Figure 2 shows the result of $2^n$ -modulo multiplication in previous method which has n² delay.
In order to decrease the delay of the operation, we use OHR in multiplication steps. For this purpose, an OHR matrix which designed for $2^n$-modulo multiplication will be used. As shown in figure 3, digits of operands will be multiplied by OHR, and results gathered in columns and will be add by OHR adder. Because of parallel operations in this system the delay of this calculation will be increased to two transistors. First is for multiplication and second for addition.

For achieving of A multiplication by B, we consider each of them as two part: High order and low order as shown in equation 1.

On this base:

\[ m = r^n \Rightarrow 0 \leq A < m = r^n \]

\[ A = \left( a_{n-1} \cdots a_{(n/2)-1} a_{(n/2)} \cdots a_1 a_0 \right)_r = \alpha_r \alpha_0 \]

\[ (\alpha_1)_P = \left( a_{n-1} \cdots a_{(n/2)} \right)_r \]

\[ (\alpha_0)_P = \left( a_{(n/2)-1} \cdots a_1 a_0 \right)_r \]

\[ B = \left( b_{n-1} \cdots b_{(n/2)-1} b_{(n/2)} \cdots b_1 b_0 \right)_r = \beta_r \beta_0 \]

\[ (\beta_1)_P = \left( b_{n-1} \cdots b_{(n/2)} \right)_r \]

\[ (\beta_0)_P = \left( b_{(n/2)-1} \cdots b_1 b_0 \right)_r \]

\[ P_1 = P_2 = r^{(n/2)} \]

As ordinary multiplication, each part of operands is multiplied and results add as shown in figure 3.
In the proposed method, multiplication operands divide to two parts and become smaller to use OHR as a fast way to get results. As shown in figure 3, for achieving A and B multiplication, we need \( n \times n \) transistor OHR multiplier and in contrast, for \( \alpha \) and \( \beta \) multiplication, it is needed \( \frac{n}{2} \times \frac{n}{2} \) transistor which means four times less hardware.

Figure 3: ordinary multiplication steps

\[
\begin{array}{c}
\alpha_1 \alpha_0 \\
\times \quad \beta_1 \beta_0 \\
\frac{\beta_0 \alpha_1}{Z_2} \quad \frac{\beta_0 \alpha_0}{Z_1} \\
+ \quad \frac{\beta_1 \alpha_1}{Z_2} \quad \frac{\beta_1 \alpha_0}{Z_1}
\end{array}
\]

Figure 4: Improved One-hot \( 2^n \)-modulo Multiplier

After achieving of multiplication results, we add up the results to achieve final one. As mentioned in previous, some products most important bits are not needed and we can decrease the OHR adder size by using of needed bits calculation as shown in figure 4.

5. Conclusion and Comparison

In this paper, a new method is proposed to calculate the result of \( 2^n \)-modulo multiplication. In this method, the large circuits of multiplication are converted to smaller one-hot adders which have one transistor delay. The results show that the delay of calculation for \( 2^n \)-modulo multiplication is decreased to three transistor delay.

The One-Hot Residue Number System allows the implementation of high-speed simple adders and multipliers. The OHRNS is an effective way to mathematical operation but for
large moduli calculation, the hardware area will be grown exponentially. By proposed way, multiplication calculation will be done in smaller area up to four times [7]. The compared results illustrated in table 1.

Table 1: Proposed Multiplier compared with [7]

<table>
<thead>
<tr>
<th></th>
<th>Delay (transistor)</th>
<th>Hardware (transistor)</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[7]</td>
<td>1</td>
<td>22n</td>
<td>22n</td>
</tr>
<tr>
<td>Proposed OHRNS Multiplier</td>
<td>3</td>
<td>4*22n +2n</td>
<td>2n</td>
</tr>
</tbody>
</table>

Reference


