Modelling of Parallel Unsigned $2^n-1$ Modular Arithmetic Multiplier for RNS

SElango$^1$, P Sampath$^2$, Sajan P Philip$^3$ and S Raja Sekar$^4$

$^1,^2,^3,^4$Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam, Erode, 638 401, Tamil Nadu, India.

elangos@bitsathy.ac.in, sampathece@bitsathy.ac.in, sajanphilip@bitsathy.ac.in, rajasekar@bitsathy.ac.in

Abstract. Modular Multiplication operations are widely used in Digital crypto processors. Modulo multipliers is an essential block for Residue Number System (RNS) computation. Pointing to increase the performance of the RNS computation, the parallel unsigned modulo multiplier for $2^n-1$ moduli is designed. A mathematical modelling, VLSI architecture and real-time verification are done in this work. Further, the modulo multipliers are described using Verilog HDL, and the synthesize results for both FPGA and ASIC technologies are presented. Comparison is made based on the parameters such as Area, Power, Delay, PDP& ADP using Cadence RTL Compiler with 180 nm, 90 nm and 45 nm TSMC CMOS Technologies. From the analysis indicate that the proposed multiplier provides a 16% area reduction and 40% speed improvement with a better PDP and ADP performance compared to the existing modulo multipliers. Finally, the usefulness of $2^n-1$ modulo multiplier in RNS environment is discussed.

Keywords: Modular arithmetic, modulo multiplication, Residue Number System (RNS).

1. INTRODUCTION

The multiplier is one of the fundamental elements in high-performance computing applications such as cryptography, Digital Signal Processing (DSP) [1]. Residue arithmetic is a particular type of arithmetic that processes only the remainder of the integers. The features of RNS are applied in many applications such as cryptographic systems, Digital Signal Processing (DSP), Wireless Sensor Networks (WSN), file storage overcloud, and privacy protection in cloud computing [2]. Hardware implementation of the RNS based application is dependent on the chosen moduli set and their arithmetic operation. In [3] the techniques used in residue arithmetic were examined and RNS in a cloud environment. The author surveyed the future research and open problems that RNS could be addressed. Modulo multiplication is an essential arithmetic operation in applications like Crypto processing and filtering in Digital Signal Processing (DSP) [2] [4]. Modular multiplication is the crucial element in RNS and the core element of many complex applications [5][6].

The multiplication operation is divided into two type’s array and booth encoder. The array-based architecture requires less area at the same time speed of the process is also less. The other kind of multiplication is booth encoder based architecture operating at high speed, but it requires more area. To overcome this trade-off, in this paper, array-based architecture is considered an attempt to increase the operation speed of unsigned array-based modular multiplication. Hence low area high-speed architecture can be achieved. This architecture can be an alternative to Booth based architectural design where the application like RSA used only unsigned integers. The significant contributions of this paper are (i) a new unsigned RNS modular multiplication algorithm for $2^n-1$ moduli is proposed. (ii) The mathematical modelling of $2^n-1$ RNS multiplication module is presented (iii) VLSI architecture is designed (v) Functional verification of the proposed module is performed in Field Programmable Gate Array (FPGA) (vi) Compare the synthesis results of both ASIC and FPGA platform are represented in terms of area,
power, delay, Power Delay Product (PDP) & Area Delay Product (ADP) (vii) Application of $2^n-1$ in RNS is discussed.

The rest of the paper is as follows: In Section 2, the related works of $2^n-1$ modular multipliers. In Section 3, proposed characteristic equation, algorithm and VLSI architecture is discussed. The synthesis results and analysis are presented in section 4. The conclusion is drawn in Section 5.

2. BACKGROUND $2^n-1$ MODULO MULTIPLIER

In [7], the author proposed an array-based and Booth Encoded based residue multiplication modules applied for the International Data Encryption Algorithm (IDEA) cryptography applications. The addition process is carried out using CSA and Parallel prefix adders. The analysis shows that the array-based approach results in a regular structure with less speed, and the booth scheme achieves high-speed operation with the increased required area. An MBE based scheme for the $2^n-1$ multiplication module to reduce the number of partial products, corresponding mathematical modelling and detailed architecture of the multiplication module are presented in [8]. The results show a significant improvement in the area and delay compared to [7].

In [9] author presents a radix-8 booth encoded RNS $2^n-1$ multiplier using an unbalanced word length of moduli. It supports a broad dynamic range with adaptable delay, achieves less area, and power consumption. In [10] the author have presented various modulo properties and designed a radix-8 $2^n-1$ & $2^n+1$ multiplier with a balanced word-length of moduli that uses CSA in [9] and parallel prefix adders in [10] for efficient addition operations. This achieves less area and power with an increase in operational delay. In [11] the authors have designed new multiplexer based booth encoder and selector architecture for MBE multiplication. This improved the speed performance of the $2^n-1$ multiplier with the area overhead because of the usage of a multiplexer in the design.

In [1], the author has designed a signed residue multiplication algorithm for three moduli, namely $2^n-1$, $2^n$ and $2^n+1$ achieved better speed performance and less area. In [1] used a hierarchy multiplication technique in [14]. In [12] the author made a critical review of the methods used to design $2^n-1$ modulo multiplier. The author identified that Booth based modulo multiplication is suitable for the high-speed application also provide further scope and Opportunities in modulo $2^n-1$ for RNS.

3. PROPOSED UNSIGNED $2^n-1$ MODULO MULTIPLICATION

The primary concept of the design is considered in [1] [13–14]. The steps followed in this method are dividing the input operands into two halves. Perform the sub multiplication among the inputs. Arrange the output of the sub multiplication then rearrange and finally perform addition operation to obtain the modulo $2^n-1$ output.

3.1. Mathematical Modelling

Unsigned integer operands $A$ and $B$ can be expressed by

\[ A = \sum_{i=0}^{n-1} a_i 2^i \]  

(1)

\[ B = \sum_{i=0}^{n-1} b_i 2^i \]

(2)

Then Modulo Product $P$ is given by the following equation

\[ P = \left( \sum_{i=0}^{n-1} a_i 2^i \right) \times \left( \sum_{i=0}^{n-1} b_i 2^i \right) \mod 2^n-1 \]

(3)

Instead of performing direct modulo multiplication given equation (3), the following steps are considered for high speed of operation.

**Step 1: Dividing the Multiplier and Multiplicand:**

In order to parallel processing, the input operands $A$ and $B$ are divided into two equal halves as per (4) – (7).
\[ A_L = \sum_{j=0}^{m-1} a_j 2^j \]  
\[ A_H = \sum_{j=0}^{m-1} a_j 2^{j+m} \]  
\[ B_L = \sum_{j=0}^{m-1} b_j 2^j \]  
\[ B_H = \sum_{j=0}^{m-1} b_j 2^{j+m} \]  

Let \( m = \frac{n}{2} \) be the half bit width of either multiplier or multiplicand.

**Step 2: Sub Multiplication:**
The sub-multiplication combination is given in table 1.

| Sub-Multiplication | Operation | Output |
|--------------------|-----------|--------|
| 1                  | \( A_L \times B_L \) | W      |
| 2                  | \( A_H \times B_L \) | X      |
| 3                  | \( A_L \times B_H \) | Y      |
| 4                  | \( A_H \times B_H \) | Z      |

**Sub-Multiplication 1**
From equation (4) & (6)
\[ W = \left( \sum_{j=0}^{m-1} a_j 2^j \right) \left( \sum_{j=0}^{m-1} b_j 2^j \right) = \sum_{j=0}^{m-1} \sum_{j=0}^{m-1} (a_j \cdot b_j)2^{j+j} \]  
(8)

Where \( \cdot \) represents AND operation

**Sub-Multiplication 2**
From equation (5) & (6)
\[ X = \left( \sum_{j=m}^{m} a_j 2^{j+m} \right) \left( \sum_{j=0}^{m} b_j 2^j \right) = \sum_{j=m}^{m} \sum_{j=0}^{m} (a_j \cdot b_j)2^{j+j-m} \]  
(9)

**Sub-Multiplication 3**
From equation (4) & (7)
\[ Y = \left( \sum_{j=m}^{m} b_j 2^{j+m} \right) \left( \sum_{j=0}^{m} a_j 2^j \right) = \sum_{j=m}^{m} \sum_{j=0}^{m} (b_j \cdot a_j)2^{j+j-m} \]  
(10)

**Sub-Multiplication 4**
From equation (5) & (7)
\[ Z = \left( \sum_{j=m}^{m} a_j 2^{j+m} \right) \left( \sum_{j=0}^{m} b_j 2^{j+m} \right) = \sum_{j=m}^{m} \sum_{j=0}^{m} (a_j \cdot b_j)2^{j+j-2m} \]  
(11)
3.2 Algorithm unsigned 2^n-1 Modulo Multiplication

Step 1: Consider the two unsigned integers input A and B
Step 2: Divide the A and B into two equal halves
Step 3: Perform sub-multiplication as given in table 1
Step 4: Arrange the sub-multiplication output as presented in Figure 1
Step 5: Fold the Right half portion towards the Left half as shown in figure 2
Step 6: Adding the bits in figure 2 produce Sum and Carry
Step 7: Final 2^n-1 Modulo Product P= Sum + EAC (End Around Carry)

3.3 Proposed 2^n-1 Modulo Multiplier architecture

Figure 3 depicts the VLSI architecture of the proposed modulo multiplier. The architecture majorly consists of Sub-multiplication and adder block. It is evident from figure 3 the addition process is carried out parallels as well as sub-multiplication is performed independently. Hence it is a viable alternative to perform unsigned modulo multiplication at high speed.
Fig. 3. Unsigned Architecture $2^n$-1 modulo multiplier

4. RESULTS AND DISCUSSIONS

4.1. FPGA Synthesis
The functionality verification of the proposed architecture is done by expressing the connectivity of the modules using Verilog RTL coding and simulated using Xilinx Integrated Simulator (ISIM) by writing test benches for feeding random inputs. Table 3 shows the FPGA synthesis results of proposed modulo multiplier in terms of Look Up-Table (LUT) resource utilization and delay of the proposed model in reconfigurable hardware architectures.

| Multiplier          | n  | ZED Board (XC7Z020CLG484-1) |
|---------------------|----|-----------------------------|
|                     |    | LUTs | Delay (ns)     |
| Proposed $2^n$-1    | 8  | 129.6 | 15.93         |
| Unsigned Modulo     | 16 | 613.36 | 24.969       |
| Multiplier          | 32 | 2449.28 | 45.05       |

4.2. ASIC Synthesis
Area, power and timing performance of the proposed architecture is evaluated by synthesizing the HDL Code using Cadence RTL Compiler genus synthesizer. The design environment is set in such a way to point the various TSMCs standard cell typical libraries 180 nm, 90 nm and 45 nm. The procedure is repeated for existing multipliers and the parameters such as area, power and timing results are given in Table 3. PDP and ADP results for modulo multiplication are shown in Table 4.
Table 3. ASIC synthesis results of $2^n$-1 modulo multiplication

| Multiplier | 180 nm | 90 nm | 45 nm |
|------------|--------|--------|--------|
|            | Area (µm²) | Power (mW) | Delay (ns) | Area (µm²) | Power (mW) | Delay (ns) | Area (µm²) | Power (mW) | Delay (ns) |
| 8          | 8668    | 0.85    | 7        | 2733      | 0.17      | 4        | 1477      | 0.11      | 4         |
| 16         | 34382   | 4.33    | 29       | 9770      | 0.80      | 16       | 5281      | 0.51      | 14        |
| 32         | 125346  | 19.18   | 87       | 39270     | 3.84      | 49       | 20981     | 2.32      | 41        |
| 8          | 8148    | 0.78    | 6        | 2569      | 0.16      | 4        | 1389      | 0.10      | 4         |
| 16         | 32663   | 3.98    | 27       | 9282      | 0.73      | 15       | 5017      | 0.47      | 13        |
| 32         | 119079  | 17.26   | 79       | 37306     | 3.45      | 45       | 19932     | 2.09      | 38        |
| 8          | 8235    | 0.80    | 6        | 2597      | 0.16      | 4        | 1404      | 0.10      | 4         |
| 16         | 32663   | 4.07    | 27       | 9282      | 0.75      | 15       | 5017      | 0.48      | 13        |
| 32         | 117825  | 18.03   | 80       | 36913     | 3.61      | 46       | 19722     | 2.19      | 38        |

Proposed $2^n$-1 Multiplier

| Multiplier | 180 nm | 90 nm | 45 nm |
|------------|--------|--------|--------|
|            | Area (µm²) | Power (mW) | Delay (ns) | Area (µm²) | Power (mW) | Delay (ns) | Area (µm²) | Power (mW) | Delay (ns) |
| 8          | 8603    | 0.08    | 4        | 1873      | 0.02      | 3        | 1027      | 0.01      | 3         |
| 16         | 25465   | 3.65    | 17       | 7146      | 0.75      | 10       | 3862      | 0.48      | 8         |
| 32         | 98435   | 15.73   | 50       | 30839     | 3.23      | 29       | 16477     | 1.90      | 23        |

Table 4. PDP & ADP results of $2^n$-1 modulo multiplication

| Multiplier | 180 nm | 90 nm | 45 nm |
|------------|--------|--------|--------|
|            | PDP (nJ) | ADP (µm²-ns) | PDP (nJ) | ADP (µm²-ns) | PDP (nJ) | ADP (µm²-ns) |
| 8          | 6       | 7368    | 0.7     | 10932     | 0.4     | 5908      |
| 16         | 126     | 148874  | 12.8    | 156320    | 7.1     | 73934     |
| 32         | 1669    | 2404136 | 188.2   | 1924230   | 95.1    | 860221    |
| 8          | 5       | 6355    | 0.6     | 10276     | 0.4     | 5556      |
| 16         | 107     | 129999  | 11.0    | 139230    | 6.1     | 65221     |
| 32         | 1364    | 2055304 | 155.3   | 1678770   | 79.4    | 757416    |
| 8          | 5       | 6588    | 0.6     | 10388     | 0.4     | 5616      |
| 16         | 110     | 132938  | 11.3    | 139230    | 6.2     | 65221     |
| 32         | 1442    | 2124385 | 95.0    | 1697998   | 83.2    | 749436    |
| Proposed   | 8       | 0.3     | 483     | 0.1       | 5619    | 0.0      | 3081      |
| $2^n$-1    | 16      | 62      | 92947   | 7.5       | 71460   | 3.8      | 30896     |
| Multiplier | 32      | 787     | 1548383 | 93.7      | 894331  | 43.7     | 378971    |

4.3. Analysis

From the synthesis results given in Table 4 indicates that the proposed work has shown a area reduction ranging from 16 % to 21% when compared with existing modulo multipliers [8][9][11]. The area reduction is due to using of array architecture rather than Booth Encoder. The power reduction in the range of 9 to 18% and achieved a maximum of 40% speed improvement compared with existing modulo multipliers. The speed improvement is due to the addition process is carried out parallels as well as sub-multiplication is performed independently. Hence the proposed work achieves PDP and ADP improvement of 45 % and 27 % respectively. The detailed performance improvement of proposed $2^n$-1 modulo multiplier is given in figure 4. The chart represents the percentage of performance improvement of various parameters of proposed work is presented in Y-axis with respect to the existing multipliers(X-axis).
4.4. **RNS Computation**

In this section, a sample example is discussed to explain the usefulness of $2^n-1$ modulo multiplication for RNS computation [1]. Suppose, to perform multiplication operation between 31 and 7 using RNS environment with respect to $\{2^n-1, 2^n\}$ moduli. RNS environment consists of three major blocks, namely, forward converter, modulo arithmetic channel, and Reverse converter. In Forward converter, converts an integer into modulus value concerning 15 and 16 here $n=4$. Hence 31 will be converted into 1 and 15 for the moduli 15 and 16 respectively. Since seven is less than 15 and 16, there is no change in integers. In modulo arithmetic channel, multiplies the two modulo integers using $2^n$ multiplier and proposed $2^n-1$ multipliers. The resultant value is $\lfloor \frac{7 \times 31}{15} \rfloor_{15} = 7$ and $\lfloor \frac{5 \times 7}{16} \rfloor_{16} = 9$. The reverse converter converts the residue numbers to full integers by using the MRC technique, so the output of this block would be 217.

5. **CONCLUSION**

In this paper, mathematical modelling and VLSI architecture of parallel unsigned $2^n-1$ modulo multiplier are presented. The proposed and existing multipliers are coded, synthesized for various CMOS Technology and verified using Zynq (XC7Z020CLG484-1) FPGA board. From the synthesized results, it is observed that the proposed multiplier provides a 16% area reduction and 40% speed improvement with a better PDP and ADP performance when compared with the state of the art modulo multipliers. Finally, using proposed modulo $2^n-1$ arithmetic modules used for RNS computing platform which may be used for cryptographic algorithm hardware implementation in future.

**References**

[1] Elango, S., & Sampath, P. (2020). Hardware Implementation of Residue Multipliers based Signed RNS Processor for Cryptosystems. Accepted for publishing in Journal of Microelectronics, Electronic Components and Materials, Informacije, MIDEM, 50 (2), 71 – 86.

[2] Chang, C. H., Molahosseini, A. S., Zarandi, A. A. E., & Tay, T. F. (2015). Residue number systems: A new paradigm to datapath optimization for low-power and high-performance digital signal processing applications. *IEEE Circuits and Systems Magazine, Institute of Electrical and Electronics Engineers Inc.* https://doi.org/10.1109/MCAS.2015.2484118

[3] Schoinianakis, D. (2020). Residue arithmetic systems in cryptography: a survey on modern security applications. *Journal of Cryptographic Engineering, 10*(3), 249–267. https://doi.org/10.1007/s13389-020-00231-w

[4] Antão, S., & Sousa, L. (2013). The CRNS framework and its application to programmable and reconfigurable cryptography. *Transactions on Architecture and Code Optimization, 9*(4), 1–25. https://doi.org/10.1145/2400682.2400692

[5] Sousa, L., Antão, S., & Martins, P. (2016). Combining Residue Arithmetic to Design Efficient
Cryptographic Circuits and Systems. *IEEE Circuits and Systems Magazine*, 16(4), 6–32. https://doi.org/10.1109/MCAS.2016.2614714

[6] Ananda Mohan, P. V. (2016). *Residue number systems: Theory and applications. Residue Number Systems: Theory and Applications*. Springer International Publishing. https://doi.org/10.1007/978-3-319-41385-3

[7] Zimmermann, R. (1999). Efficient VLSI implementation of modulo \(2^n \pm 1\) addition and multiplication. *Proceedings - Symposium on Computer Arithmetic*, 158–167. https://doi.org/10.1109/arith.1999.762841

[8] Efstatiiou, C., Vergos, H. T., & Nikolos, D. (2004). Modified booth modulo \(2^n - 1\) multipliers. *IEEE Transactions on Computers*, 53(3), 370–374. https://doi.org/10.1109/TC.2004.1261842

[9] Muralidharan, R., & Chang, C. H. (2011). Radix-8 booth encoded modulo \(2^n-1\) multipliers with adaptive delay for high dynamic range residue number system. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(5), 982–993. https://doi.org/10.1109/TCSI.2010.2092133

[10] Muralidharan, R., & Chang, C. H. (2012). Area-power efficient modulo \(2^n-1\) and modulo \(2^n+1\) multipliers for \(\{2^n-1, 2^n, 2^n+1\}\) based RNS. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(10), 2263–2274. https://doi.org/10.1109/TCSI.2012.2185334

[11] Li, L., Hu, J., & Chen, Y. (2012). Modified booth encoding modulo \(2^n-1\) multipliers. *IEICE Electronics Express*, 9(5), 352–358. https://doi.org/10.1587/elex.9.352

[12] Kumar, R., Jaiswal, R. K., & Mishra, R. A. (2020, September 15). Perspective and Opportunities of Modulo \(2^n - 1\) Multipliers in Residue Number System: A Review. *Journal of Circuits, Systems and Computers*. World Scientific. https://doi.org/10.1142/S0218126620300081

[13] Shoba, M., &Nakkeeran, R. (2017). Energy and area efficient hierarchy multiplier architecture based on Vedic mathematics and GDI logic. *Engineering Science and Technology, an International Journal*, 20(1), 321–331. https://doi.org/10.1016/j.jestch.2016.06.007

[14] Elango, S., &Sampath, P. (2020). Implementation of High Performance Hierarchy-Based Parallel Signed Multiplier for Cryptosystems. *Journal of Circuits, Systems and Computers*, 29(13). https://doi.org/10.1142/S021812662050214X