International Journal of LNCT
Volume-4 Issue-20 October -2020
ISSN: (Online):2456-9895

High Speed Post Processing Residue Module Arithmetic's Structure using Reversible Gate

Rajdeep Kaurav, Prof. Prashant Purohit
M. Tech. Scholar, Associate Professor
Dept. of Electronics and Communication
LNCT, Bhopal

***Abstract*— Due to the increasing spread of digital computing, investigation of various number representation systems in the digital field seems necessary. In general, the number representation systems regarding their applications can be classified in two areas: general-purpose and specific-purpose. Binary Digit Residue Number System (BD-RNS) is one of the specific-purpose and optimized number system.**

In Residue Number System-based systems, how to select moduli set and evaluate its performance is an important issue. The objective of this study is to propose a systemic performance evaluation method for RNS based on the properties of moduli set. By abstracting the inherent properties of moduli sets, such as the complexity of arithmetic units, utilization ratio of dynamic range, parallelism and balance between residue channels, this method can provide advices on moduli set selection and carry out performance estimation before circuit's implementation.

Keywords: - CMOS Technology, Reversible Gate, Feynman Gate, Peres Gate

I. INTRODUCTION

Residue Number System (RNS) is a non-weighted numerical system, in which a large integer is divided into several small integers. These small integers are computed independently and concurrently in multiplication and addition operations. There is no carry propagation between residue channels.

RNS is used in cryptography firstly, and then it is received intensive researches in digital signal processing (DSP) systems with intensive multiplication and addition operations.

For many RNS-based applications, the moduli set are the main factor for implementation complexity. This is because the complexity of modular multiplication and addition are related to the form of moduli set. Besides, the complexities of the operations, such as forward and backward conversion, scaling and sign detection, etc., also depend on the moduli set form. Many specific fonns of moduli set has been proposed. The commonly used three-channel muduli sets are proposed in [1-3].

In recent years, efficient schemes for modulo adder have been studied intensively [11]-[13]. Generally, modulo 2^n+1 adder can be divided into three categories, depending on the type of operands that they accept and output:

i. The result and both inputs use weighted representation;
ii. The result and both inputs use diminished-1 representation;
iii. The result and one input use weighted representation, while the other input uses diminished-1.

For the first category, used Booth encoding to realize, but depart from the diminished-1 arithmetic, which leads to a complex architecture with large area and delay requirements. For the second category, proposed diminished-1 adder with n-bit input operands. The adders use a non-Booth recoding and a zero partial-product counting circuit. The main drawback in this architecture was handling of zero inputs and results were not considered.

The proposed new modulo adder by using the third category. This architecture use ROM based look-up methods are competitive. The main drawback in this architecture increasing n-bit, they become infeasible due to excessive memory requirements.

The also proposed for the third category architecture and reduce the memory requirement and speed up. The new architecture is based on n-bit addition and radix-4 booth algorithm, which is efficient and regular. We are replaced diminished-1 modulo 2^n+1 adder by inverted n-bit adder.

Actually, the design in this letter is not the first reversible version ALU. Also gave an ALU design by generalizing the V-shape design which can achieve five basic arithmetic-logical operations on two n-bit operands, but the structure is so simple that some functions cannot even get the right results, for example, we cannot calculate the ADD operation only using the bitwise exclusive-or of the two n-bit arguments| A > and| B > without considering the carry bit or only by setting the carry bit to the TRUE. The goal of this letter is to build a multi-functional circuit reasonably that conditionally performs one of several possible arithmetic-logical operations on two operands | A > and | B > depending on control input data instructions. The following section is based on the assumption that the readers are familiar with the basics of reversible logic [5].

II. RESIDUE NUMBER

For conventional logic circuits there exists much research, even whole books, dedicated to the design and implementation of computer arithmetic. This is definitely not the case for reversible logic. The constraint that the circuits must be garbage-free is what makes it an interesting research problem, but most proposed designs (both hand-made and CAD generated) still implement the conventional algorithms with garbage. They use the reversible gates, but as their sole goal is to reduce logic size or number of garbage bits for a specific fixed-size circuit, very little knowledge is actually gained from this approach. However, arithmetic functions often have some inherent properties that can be exploited to make a very regular circuit design. A good example is the ripple-carry adder, where only a redesign gave the garbage-free V-shaped adder; a redesign that none of the automatic approaches can find. In many cases the arithmetic function itself must also be redefined, such that it can be expressed reversibly. Here our current work on multiplication is the obvious example. With this in mind, we need more design work on good garbage-free implementations of reversible circuits.

To convert the decimal number 29 to a residue number, we compute:

$$R_5\ 29 \bmod 5 = 4$$

$$R_3\ 29 \bmod 3 = 2$$

$$R_2\ 29 \bmod 2 = 1$$

The decimal number 29 is represented by in the above residue number system.

The main advantage of the residue number system is the absence of carriers between columns in addition and in multiplication.

Residue Addition:-
Addition can be accomplished by simply adding (or subtracting) the small integer values, modulo their specific moduli. That is,

$$C = A + B \bmod M$$

Can be calculated in RNS as

$$C_i = a_i + b_i$$

One does not have to check for overflow in these operations.

Residue Subtractor:-
Addition can be accomplished by simply adding (or subtracting) the small integer values, modulo their specific moduli. That is,

$$C = A - B \bmod M$$

Can be calculated in RNS as

$$C_i = a_i - b_i$$

One does not have to check for overflow in these operations.

As We know, the addition arithmetic operation [1] represent the basic operation that is used by all of us in a daily basis, and this basic operation is one that enable us to implement another more complex mathematical operations such as subtraction, multiplication and division. The literature of the mathematics proposed many types of mathematics; the most commonly used and known is the traditional mathematics where the arithmetic functions performed directly on integer or floating point operands. Another type that is known as a modular arithmetic has been proposed where the arithmetic functions are executed on the residues of the input operands with respect to a set of modulo. This simple definition of the modular arithmetic [2, 3] reveals to us that there should be some way to represent the conventional numbers in such a way that will allow us to perform this type of mathematics. This kind of representation is called a Residual Numbering System (RNS). We propose an RNS-Based two-operand digital adder [1, 2]; we abbreviate it as RNS-Adder. This adder is based on RNS (Residual Numbering System) in which an integer is represented by an ordered set of residues. This scheme is preferable in high speed computing systems since the high-independency among the set of

residues. Thus, instead of working on large operands, working on residues which are definitely smaller ones eases some arithmetic operations like addition, subtraction, and multiplication. Fig.1 depicts shortly the design of a complete RNS-adder module.

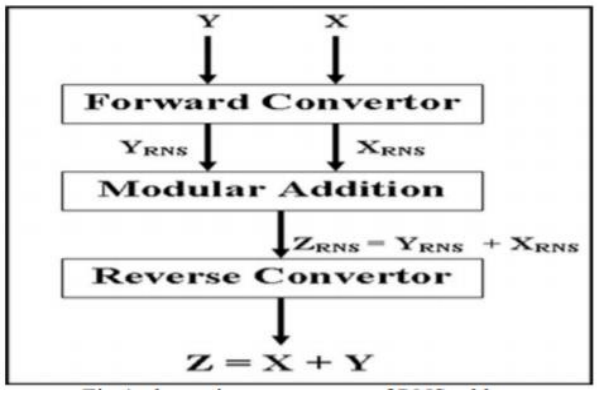

Figure1: The main components of RNS adder

An important issue that must be taken into consideration when designing RNS-based digital system is the representation of inputs and outputs. Unfortunately, we used to use decimal numbering system in our life, and no one can force us to use another one. Thus, despite the efficiency of RNS in many arithmetic operations like addition and multiplication it's useless unless there is a technique to convert the conventional represented inputs to RNS (forward conversion) and the RNS results to conventional output (reverse conversion). Another issue in designing RNS-based arithmetic systems is the set of moduli used. As we will show that the basic principle in the computation of residues is division, with the moduli as the divisors. But division is an expensive operation in hardware and so is rarely used in the computation of residues. Division can be avoided in the case of using special set of moduli. This also simplifies the implementation of the modules but with some mathematical tricks.

III. REVERSIBLE GATE

Reversible common sense is gaining importance in regions of CMOS layout because of its low energy dissipation. The traditional gates like AND, OR, XOR are all irreversible gates. Recall the case of conventional AND gate. It consists of inputs and one output. As an end result, one bit is misplaced every time a computation is completed. In line with the reality desk shown in Fig.1, there are three inputs (1, zero), (0, 1) and (zero, 0) that corresponds to an output zero. Subsequently it isn't feasible to determine a unique enter that resulted in the output 0. with a view to make a gate reversible additional input and output strains are brought in order that a one to at least one mapping exists between the enter and output. This prevents the loss of statistics that is predominant purpose of strength dissipation in irreversible circuits. The enter that is introduced to an m x n characteristic to make it reversible is known as consistent enter (CI). All the outputs of a reversible circuit want not are used within the circuit. The ones outputs that aren't used within the circuit are called as garbage output (cross). The wide variety of garbage output for a specific reversible gate is not constant. The 2 fundamental constraints of reversible common sense circuit is

- Fan out not allowed
- Feedbacks or loops not allowed.

IV. PROPOSED METHODOLOGY

The proposed algorithm can be implemented easily as shown in figure 6. This implementation is easy to build if we have a modules called modulo adders. Thus we have to implement these modules from reversible gate.

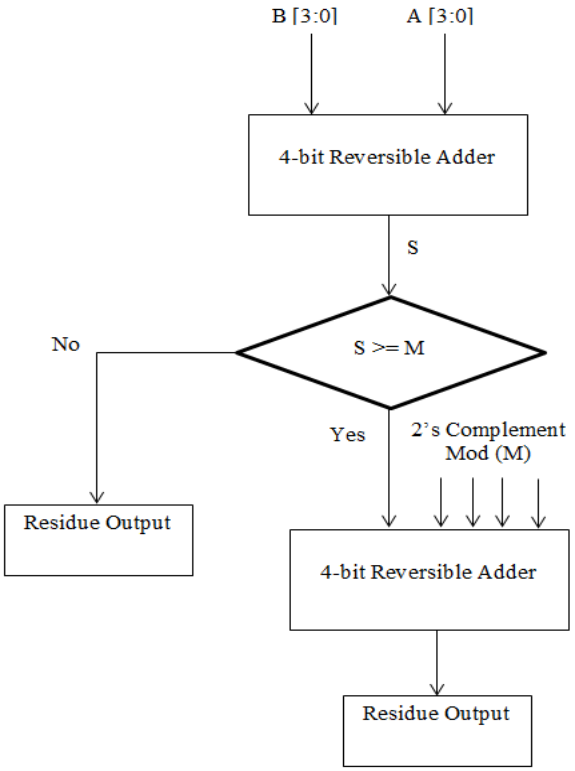

Figure 2: Flow Chart of 4-bit Reversible Residue Adder

The proposed implementation is programmed (Described) and implemented using VHDL language which is a Hardware Description Language that was developed by the Institute of Electrical and Electronic

Engineers (IEEE) as a standard language for describing the structure and behavior of digital electronic systems. It has many features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips. Features of VHDL allow electrical aspects of circuit behavior (such as rise and fall times of signals, delays through gates, and functional operation) to be precisely described. The resulting VHDL simulation models can then be used as building blocks in larger circuits (using schematics, block diagrams, or system-level VHDL descriptions) for the purpose of simulation. As a compiling and simulation tool for VHDL, we used the ModelSim XE III 6.2g which is known as a powerful tool developed by Mentor Graphics Company to offer an appropriate environment to validate the functional correctness of the design hardware.

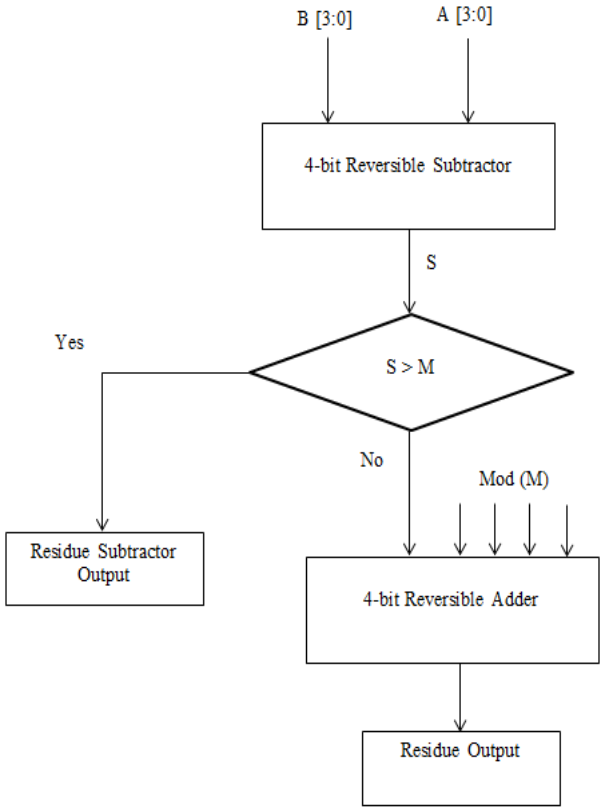

Figure 3: Flow Chart of 4-bit Reversible Residue Sub-tractor

V. SIMULATION RESULT

The proposed implementation is programmed (Described) and implemented using VHDL language which is a Hardware Description Language that was developed by the Institute of Electrical and Electronic Engineers (IEEE) as a standard language for describing the structure and behavior of digital electronic systems. It has many features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips. The resulting VHDL simulation models can then be used as building blocks in larger circuits (using schematics, block diagrams, or system-level VHDL descriptions) for the purpose of simulation.

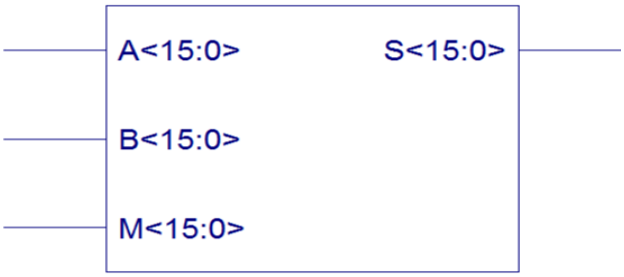

Figure 4: View Technology Schematic of 16-bit Reversible Residue sub-tractor using TSG Gate

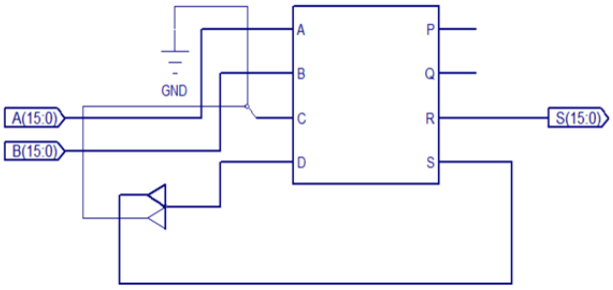

Figure 5: RTL View of 16-bit Reversible Residue sub-tractor using TSG Gate

```
Device utilization summary:
---------------------------

Selected Device : 2vp2fg256-7

 Number of Slices:                       20  out of   1408     1%
 Number of 4 input LUTs:                 35  out of   2816     1%
 Number of bonded IOBs:                  48  out of    140    34%

Timing Summary:
---------------
Speed Grade: -7

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 15.143ns
```

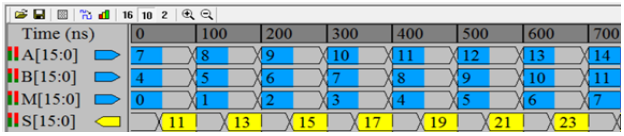

Figure 6: Output Waveform of 16-bit Reversible Residue sub-tractor using TSG Gate

VI. CONCLUSION

This paper has presented a high-speed residue binary number adder and converters between binary and residue numbers for moduli n. In the proposed method, only some fast binary additions are used for the arithmetic operations and the conversions. The design results show that the performance of proposed circuits will comparable with binary architectures and the schemes are high-speed architectures.

REFERENCES

[1] Vladimir Rožić and Ingrid Verbauwhede, “Hardware-Efficient Post-Processing Architectures for True Random Number Generators”, IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 66, Issue 7, July 2019.
[2] Uttam Narendra Thakur, Samsubhra Mallick, Rabindra Mahan Moitra and Mayukh Kotal, “FPGA Based Effecient Architecture for Conversion of Binay to Residue Number System”, IEEE 2017.
[3] Adib Armand and Somayeh Timarchi, “Low Power Design of Binary Signed Digit Residue Number System Adder”, 2016 24th Iranian Conference on Electrical Engineering (ICEE).
[4] Jian Wang, Shang Ma, Ze-guo Yang and Jianhao Hu, “A Systemic Performance Evaluation Method for Residue Number System”, 2016 2nd IEEE International Conference on Computer and Communications.
[5] Shugang Wei, “Fast Signed-Digit Arithmetic Circuits for Residue Number Systems”, 978-1-5090-0246-7/15/$31.00 ©2015 IEEE.
[6] I. B. K. Raju, P. Rajesh Kumar and P. Bhaskara Rao, “Residue Arithmetic’s using Reversible Logic Gates”, Devices, Circuits and Systems (ICDCS), 2014 2nd International Conference on 10.1109/ICDCSyst.2014.6926193.
[7] H.R. Bhagyalakshmi and M.K. Venkatesha, "Optimized reversible BCD adder using new reversible logic gates", *Journal of Computing*, vol. 2, no. 2, 2010.
[8] A. D'amora, "Reducing power dissipation in complex digital filters by using the quadratic residue number system", *Conf. Record 34th Asil. Conf. Signals SystComput. (ACSSC 2000)*, vol. 2, pp. 879-883, 2000.
[9] H. Thapliyal, N. Ran-Ganathan and S. Kotiyal, "Design of Testable Reversible Sequential Circuits", *IEEE Transactions on VLSI*, pp. 1-9, 2012.
[10] H. Thapliyal and N. Ranganathan, "Design of reversible latches optimized for quantum cost delay and garbage outputs", *Proceedings of the Twenty Third IEEE International Conference on VLSI Design*, pp. 235-240, 2010.
[11] M. Chuang and C. Wang, "Reversible sequential element designs", *Proceedings of the IEEE Asia and South Pacific Design Automation Conference*, pp. 420-425, 2007.
[12] S. Kumar Sastry Hari, S. Shroff, S. Noor Mahammad and V. Kamakoti, "Efficient building blocks for reversible sequential circuit design", *Proceedings of the Forty Ninth IEEE International Midwest Symposium on Circuits and Systems*, pp. 437-441, 2006.
[13] J.E. Rice, "A New Look at Reversible Memory Elements", Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 243-246, 2006.
[14] Amos Omondi and Benlamin Premkumar, Residue Number Susyem Theory and Implementation, Imperial College Press, 2007, pp. 1-134.
[15] Milos D. ERCEGOVAC, Tomas LANG, Digital Arithmetic, Morgan Kaufmann Publishers, by Elsevier Science (USA), Vol1, Ch2, pages (51-136), 2004.