*REDVET - Revista electrónica de Veterinaria* - ISSN 1695-7504
Vol 25, No. 2 (2024)
http://www.veterinaria.org
Article Received: 3 September 2024 Revised:4 October 2024 Accepted:4 November 2024

# FPGA Implementation of Modulo 2n ±1 Adder-Subtractor

**Pratik M[1*], Dinesh Sethi[2]**

[1*,2]Department of Electronics and Communication Engineering, JECRC University, Jaipur, India

***Corresponding author:**
*Email: pratik.m@jecrcu.edu.in

**Abstract**
Arithmetic modulo operators, especially those involving $2^n + 1$, have wide-ranging applications in fields like pseudorandom number generation, cryptography, and digital signal processing (DSP). These modulo operations are particularly useful in the Residue Number System (RNS). This work presents the design of a modulo $2^n \pm 1$ adder-subtractor using a parallel prefix adder. The design leverages two numerical representations: weighted-one and diminished-one. After developing the adder-subtractor, it is implemented on an FPGA to evaluate its performance. A detailed comparison has been conducted between the two representations, focusing on area utilization and execution time.

**Keywords-** Modulo Arithmetic, Modulo 2n+1, modulo 2n-1, Redundant Number Systems, Diminished-One Representation, LUTs, Timing Analysis, Area Optimization, Delay Optimization, Prefix Adder, CSA, Parallel Prefix Structures, Logic Slices.

## INTRODUCTION

Various domains such as pseudorandom number generation, cryptography, and convolution computations employ arithmetic modulo operators. Modulo $2^n+1$ operators are frequently used in Residue Number System (RNS) applications. RNS is an arithmetic system that decomposes numbers into parts and performs arithmetic operations in parallel, eliminating the need for carry computations for each residue. RNS is utilized for operations like addition, subtraction, and multiplication, and plays a key role in the design of digital signal processors.
The moduli set $\{2^n-1, 2^n, 2^n+1\}$ is particularly effective in high-performance RNS applications because it allows for fast residue arithmetic. Special moduli sets have been adopted to reduce hardware complexity in converters and arithmetic operations, with the triple moduli set $\{2^n-1, 2^n, 2^n+1\}$ offering notable benefits. Due to the operand lengths of these moduli, the operation delay is primarily determined by the modulo $2^n-1$ channel. Reducing the time required for modulo $2^n-1$ addition directly speeds up the overall RNS addition process. To accelerate modulo $2^n\pm1$ arithmetic operations, the diminished-1 representation of binary numbers has been introduced.
Modulo $2^n\pm1$ operators have long been of interest because they simplify certain RNS arithmetic operations.
The complexity of a modulo $2^n+1$ arithmetic unit is influenced by the representation of the input operands. Three main representations have been considered: the normal weighted representation, the diminished-1 representation, and the signed-LSB representation. In this context, we focus on the first two, as the signed-LSB representation is less efficient in terms of delay and area. For modulo $2^n+1$ arithmetic, the operands and results must be between 0 and $2^n$. In the normal weighted representation, each operand requires n + 1 bits. The diminished-1 representation, however, offers a more compact encoding of the input operands and simplifies modulo $2^n+1$ arithmetic operation. In diminished-1, a number A is represented as $a_zA^*$, where $a_z$ is a single bit called the zero indication bit, and A* is an n-bit vector. If A > 0, then $a_z$ = 0 and A* = A - 1; if A = 0, then $a_z$ = 1 and A* = 0. In the design of efficient diminished adders, the most commonly used adder is the inverted end-around-carry n-bit adder. This adder accepts two n-bit operands and produces a sum that is incremented by one compared to their integer sum, provided that the integer addition does not generate a carry output. An inverted end-around-carry adder can be implemented using an integer adder, with its carry output connected back to its carry input via an inverter. However, since the carry output depends on the carry input, directly connecting them may lead to unwanted race conditions.
This work presents the design of a modulo $(2^n \pm 1)$ adder-subtractor using a parallel prefix adder. The design leverages two numerical representations: weighted-one and diminished-one. After developing the adder-subtractor, it is implemented on an FPGA to evaluate its performance. A detailed comparison has been conducted between the two representations, focusing on area utilization and execution time.

## MODULAR ARITHMATIC

Modular arithmetic is a system of arithmetic for integers, where numbers "wrap around" upon reaching a certain value the modulus. Modular arithmetic defined mathematically by introducing a congruence relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication. For a positive integer n, two integers a and b are said to be congruent modulo n

$$a \equiv b \pmod{n},$$

*REDVET - Revista electrónica de Veterinaria* - ISSN 1695-7504
Vol 25, No. 2 (2024)
http://www.veterinaria.org
Article Received: 3 September 2024 Revised:4 October 2024 Accepted:4 November 2024

Their difference a − b is an integer multiple of n (or n divides a − b). The number n is called the modulus of the congruence.
Modular arithmetic is referenced in number theory, group theory, ring theory, knot theory, abstract algebra, computer algebra, cryptography, computer science, chemistry and the visual and musical arts.
Modular arithmetic is used to calculate checksums that are used within identifiers International Bank Account Numbers (IBANs) for example make use of modulo 97 arithmetic to trap user input errors in bank account numbers.
In cryptography, modular arithmetic directly underpins public key systems such as RSA and Diffie-Hellman, as well as providing finite fields which underlie elliptic curves.
In computer algebra, modular arithmetics is commonly used to limit the size of integer coefficients in intermediate calculations and data. It is used in polynomial factorization, a problem for which all known efficient algorithms use modular arithmetic.
In computer science, modular arithmetic is often applied in bitwise operations and other operations involving fixed-width, cyclic data structures.
The modulo operation, as implemented in many programming languages and calculators XOR is the sum of 2 bits, modulo 2.
The method of casting out nines offers a quick check of decimal arithmetic computations performed by hand. It is based on modular arithmetic modulo 9, and specifically on the crucial property that $10 \equiv 1 \pmod 9$.
Arithmetic modulo 7 is used in algorithms that determine the day of the week for a given date. In particular, Zeller's congruence and the doomsday algorithm make heavy use of modulo-7 arithmetic[2].

***Modulo adder***

Arithmetic modulo $(2^n - 1)$ (Mersenne numbers) and modulo $(2^n + 1)$ (Fermat numbers) is used in residue number systems and cryptography. Efficient and fast modulo adders and Subtractor are a need for corresponding high performance integrated circuits.
Binary numbers with n bits are denoted as $A = a_{n-1}a_{n-2}\ldots\ldots a_0$ then

$$A = \sum_{i=0}^{n-1} (2^i a_i)$$

Reduction of a number A modulo a number M ("A mod M") can be represented by a division (with the remainder as result) or by subtracting the modulus until A <M. For the moduli $(2^n - 1)$ and $(2^n + 1)$, the modulo reduction of a number A with at most $2^n$ bits can be computed by an addition or subtraction[3].
$2^n \bmod (2^n - 1) = 2^n - (2^n - 1) = 1$
the reduction modulo $(2^n - 1)$ can be represented as
$A \bmod (2^n - 1) = (A \bmod 2^n + A \operatorname{div} 2^n) \bmod (2^n - 1)$
where the modulo operation on the right hand side is used for final correction if the addition yields a result $> 2^n - 1$ ($2^n$-1 has to be subtracted once). Then the modulo ($2^n$-1) reduction is computed by adding the high n bit word (A div $2^n$) to the low n - bit word (A mod $2^n$) and then subtracting $2^n - 1$.
Similarly,
$2^n \bmod (2^n + 1) = 2^n - (2^n + 1) = -1$
the reduction modulo (2n + 1) can be represented as
$A \bmod (2^n + 1) = (A \bmod 2^n - A \operatorname{div} 2^n) \bmod (2^n + 1)$
where the modulo operation on the right hand side is used for final correction if the subtraction yields a negative result ($2^n + 1$ has to be added once). Then the modulo ($2^n$+1) reduction is computed by subtracting the high n bit word from the low n bit word and then adding $2^n + 1$ Also the modulo operator has the property that a sum modulo M is equivalent to the sum of its operands modulo M:
$(A + B) \bmod M = (A \bmod M + B \bmod M) \bmod M$
*Modulo $(2^n \pm 1)$ Adder*
Modulo-m addition of mod-m residues A and B ($0 \le A, B < m$) is defined as:

$$S = |A+B|_m = \begin{cases} A + B - m & \text{if } A + B > m \\ A + B & \text{otherwise} \end{cases}$$

Replacing m in this equation with $2^n - 1$ or $2^n + 1$ shows the corresponding equations for mod-($2^n$–1) or mod-($2^n$+1) addition, respectively. Because comparing A + B with $2^n$-1 or $2^n + 1$ is nontrivial, so this equation is modified into new equations which use much simple comparisons with $2^n$. Here, $W = (w_n\, w_{n-1} \ldots w_1\, w_0) = A + B$ is the true sum of A and B, which can be decomposed into a single bit wn and an n-bit number $|W|2^n$ . Similarly, $W' = (w'_n w'_{n-1} \ldots w'_1 w'_0) = A + B - 1$ is the diminished sum of A and B, with its associated decomposition into a single bit w′n and an n-bit number $|W'|2^n$.[4]

$$S^- = |A+B|_{2^n-1} = \begin{cases} W - 2^n + 1 & \text{if } W > 2^n \\ W & \text{otherwise} \end{cases}$$

$$S^+ = |A+B|_{2^n+1} = \begin{cases} W' - 2^n & \text{if } W' > 2^n \end{cases}$$

*REDVET - Revista electrónica de Veterinaria* - ISSN 1695-7504
Vol 25, No. 2 (2024)
http://www.veterinaria.org
Article Received: 3 September 2024 Revised:4 October 2024 Accepted:4 November 2024

W` + 1 otherwise

*Modulo ($2^n$ -1) addition*

Modulo ($2^n$ -1) addition can be formulated as

$(A+B) \mod (2^n-1) = \{ A + B - (2^n-1) = (A + B + 1) \mod 2^n$ if $A + B > 2^n-1 \}$
$A + B$ otherwise

This equation can also be written as:

$(A+B) \mod (2^n-1) = \{ A + B - (2^n-1) = (A + B + 1) \mod 2^n$ if $A + B > 2^n \}$
$A + B$ otherwise

***Modulo $2^n$+1 Subtractor***

Subtraction is an operation which is widely used in digital signal processing applications for calculating mean error estimation, mean square error estimation and calculation of sum of absolute differences. Modulo arithmetic is also used in these types of applications like efficient modulo subtraction circuits are mostly used.

**Weighted – one representation**

This representation is also called integer representation[1]. In this each operand requires n+ 1 bits for its representation but only utilizes $2^n + 1$ representations out of the $2^{n+1}$.

**Diminished-1 representation**

In diminished-1 representation each operand is represented decreased by one compared to its weighted representation[1].Zero operands are not used in the computation. The results are derived alternatively when any operand or the result is zero. Therefore only n-bit operands are used in a diminished-1 channel leading to smaller and faster components. In the diminished-1 representation A is represented as $a_zA^*$, where $a_z$ is a single bit, often called the zero indication bit, and A* is an n-bit vector, often called the number part. If A > 0, then $a_z = 0$ and $A^* = A - 1$. whereas for A =0; $a_z = 1$, and $A^* = 0$. For example, the diminished-1 representation of A = 5 modulo $17_{10}$ is $00100_2$.

**RESULTS**

Timing and area analysis is shown in table below for all adder-subtractor which have analyzed. Different values of n is used to calculate delay like n=5,8,15.

| S.No | Device name | No. of input | No.of SLICE | LUT | Gate density | Delays | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Total | Logic | nets |
| 1 | Weighted one plus one adder | 5 | 12 | 21 | 153 | 8.774ns | 4.607ns | 4.167ns |
| | | 8 | 19 | 35 | 261 | 10.922ns | 5.011ns | 5.910ns |
| | | 15 | 39 | 66 | 489 | 15.984ns | 6.028ns | 9.956ns |
| 2 | Weighted one minus one adder | 5 | 9 | 17 | 126 | 6.494ns | 4.199ns | 2.295ns |
| | | 8 | 18 | 35 | 261 | 9.805ns | 4.808ns | 4.997ns |
| | | 15 | 39 | 66 | 489 | 15.326ns | 5.823ns | 9.503ns |
| 3 | Weighted one plus one sub | 5 | 14 | 23 | 177 | 7.935ns | 4.404ns | 3.531ns |
| | | 8 | 24 | 40 | 333 | 10.821ns | 5.013ns | 5.808ns |
| | | 15 | 46 | 78 | 645 | 16.446ns | 6.382ns | 10.064ns |
| 4 | Weighted one minus one sub | 5 | 10 | 20 | 156 | 7.762ns | 4.204ns | 3.558ns |
| | | 8 | 22 | 39 | 324 | 10.007ns | 4.810ns | 5.197ns |
| | | 15 | 46 | 78 | 642 | 16.446ns | 6.382ns | 10.064ns |
| 5 | Diminished one plus one adder | 5 | 16 | 28 | 225 | 6.755ns | 4.199ns | 2.556ns |
| | | 8 | 28 | 50 | 435 | 10.800ns | 5.019ns | 5.781ns |
| | | 15 | 55 | 94 | 831 | 15.707ns | 6.198ns | 9.509ns |
| 6 | Diminished one minus one adder | 5 | 13 | 24 | 198 | 7.864ns | 4.402ns | 3.462ns |
| | | 8 | 28 | 50 | 435 | 10.019ns | 4.808ns | 5.211ns |
| | | 15 | 55 | 94 | 831 | 15.707ns | 6.198ns | 9.509ns |
| 7 | Diminished one plus one sub | 5 | 18 | 32 | 261 | 6.911ns | 4.199ns | 2.712ns |
| | | 8 | 32 | 56 | 519 | 10.778ns | 5.015ns | 5.763ns |
| | | 15 | 63 | 108 | 999 | 16.405ns | 6.437ns | 10.068ns |
| 9 | Diminished one minus one sub | 5 | 17 | 29 | 247 | 7.798ns | 4.402ns | 3.396ns |
| | | 8 | 31 | 54 | 491 | 10.123ns | 4.810ns | 5.313ns |
| | | 15 | 62 | 106 | 967 | 16.805ns | 6.737ns | 10.068ns |

*REDVET - Revista electrónica de Veterinaria* - ISSN 1695-7504
Vol 25, No. 2 (2024)
http://www.veterinaria.org
Article Received: 3 September 2024 Revised:4 October 2024 Accepted:4 November 2024

From the above table we can analyze that weighed-one approach is more area efficient for all adder subtractor in comparison to diminished-one approach. In terms of delay, weighed-one approach has less delay for $2^n - 1$ adder-subtractor and diminished-one has less delay for $2^n + 1$ adder-subtractor.

## CONCLUSION AND FUTURE WORK

Efficient modulo $2^n+1$ and $2^n-1$ adders and subtractors are valuable in various computer applications, including all implementations of Residue Number Systems (RNS). This work presents two significant contributions to the problem of modulo $2^n+1$ and $2^n-1$ addition and subtraction: weighted and diminished approaches.

A novel architecture has been developed using a sparse, fully regular parallel-prefix carry computation unit. This architecture leverages the inverted circular idempotency property of the parallel-prefix carry operator in modulo $2^n+1$ and $2^n-1$ arithmetic. Additionally, it introduces a new prefix operator, eliminating the need for the double computation tree required in previous high-speed designs.

The same parallel-prefix methodology was extended to support modulo $2^n+1$ and $2^n-1$ subtraction, incorporating the 2's complement technique. Detailed experimental evaluations demonstrate that the proposed architecture significantly outperforms earlier designs in terms of implementation efficiency, area usage, and timing, while maintaining a high execution rate.

In this design, the subtractor result has been optimized with respect to adder although it can be even more optimized by merging of parallel prefix subtractor or different approach for 2's complement.

## REFERENCES

[1] Bhaskar Gaur, and Himanshu Thapliyal**,** "Novel Optimized Designs of Modulo $2^n+1$ Adder for Quantum Computing", IEEE Transactions on Very Large Scale Integration (VLSI) Systems ( Volume: 32, Issue: 9, pp. 1759 – 1763, September 2024)

[2] Yamani, Sudha & Kudulla, H K Raghu & Bhavani, "Area Efficient Sparse-4 Diminished-1 Modulo $2^n+1$ Adder", IEEE Wireless Antenna and Microwave Symposium, (2024)

[3] Prabir Saha, Rekib Uddin Ahmed, and Sheba Thabah, "Design and Implementation of Multi-operand $2^n-1$, $2^n$, and $2^n+1$ Modulo Set Adder", Advances in Communication, Devices and Networking , pp.1-8, August (2021)

[4] Subodh Kumar Singhal, B. K. Mohanty, Sujit Kumar Patel, and Gaurav Saxena, "Efficient Diminished-1 Modulo $(2^n+1)$ Adder Using Parallel Prefix Adder", Journal of Circuits, Systems and Computers, vol. 29, No. 12, July (2020)

[5] Constantinos Efstathiou, Kiamal Pekmestzi, and Nikolaos Moshopoulos, "On the Diminished-1 Modulo $2^n+1$ Addition and Subtraction", Journal of Circuits, Systems and Computers, vol. 29, No. 05, July (2019)

[6] G H Asha, and J Uday, "Implementation of 32-bit area-efficient hybrid modulo $2^n+1$ adder and multiplier", International Conference on Control, Instrumentation, Communication and Computational Technologies, July (2014)

[7] Haridimos T. Vergos, and Giorgos Dimitrakopoulos, "On Modulo $2^n + 1$ Adder Design" Feb.(2012)

[8] H.T. Vergos and C. Efstathiou, "Efficient Modulo $2^n+1$ Adder Architectures," Integration, the VLSI J., vol. 42, no. 2, pp. 149-157,Feb. (2009).

[9] G. Jaberipur and B. Parhami, "Unified Approach to the Design of Modulo-$(2^n - 1)$ Adders Based on Signed-LSB Representation of Residues," Proc. 19th IEEE Symp. Computer Arithmetic, pp. 57-64,(2009).

[10] H. T. Vergos and D. Bakalis, "On the Use of Diminished-1 Adders for Weighted Modulo $2^n + 1$ Arithmetic Components"(2008)

[11] R. Zimmerman, "Efficient VLSI Implementation of Modulo $2^n \pm 1$ Addition and Multiplication," Proc. 14th IEEE Symp. ComputerArithmetic, pp. 158-167, Apr. (1999).

[12] Singh, A., Anand, V., Singh, S., & Sharma, A. Examining GIS Methodologies and Their Diverse Applications in Solid Waste Management. *Journal of Survey in Fisheries Sciences, 10*(1), 1442-1447. (2023)