Journal of VLSI circuits and systems

RESEARCH ARTICLE

Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 6, No. 2, 2024 (pp.122-129)

WWW.VLSIJOURNAL.COM

Thermometer Coding-Based Application-Specific Efficient Mod Adder for Residue Number Systems

[1,2]K. Vijaya Vardhan,[3]Sarada Musala

[1]*Department of ECE, Vignan's Foundation for Science, Technology and Research, Guntur, Andhra Pradesh, India 522213*
[2]*Department of ECE, Vignan's LARA Institute of Technology & Science, Guntur, Andhra Pradesh, India522213*
[3]*Department of ECE,Vignan's Foundation for Science, Technology and Research, Guntur, Andhra Pradesh, India 522213*

KEYWORDS:
RNS,
TCR,
Modular Adder,
Forward and
Reverse Converters.

ARTICLE HISTORY:
Submitted : 15.11.2021
Revised : 12.01.2023
Accepted : 26.11.2024

https://doi.org/10.31838/jvcs/06.02.14

Abstract

The interest of researchers in the implementation of power-efficient digital circuits has drastically increased in the last few decades. The reason for this is that most of the applications like Embedded Systems, Communications, Digital Signal Processing, etc., are battery-operated and in some applications, it is impossible to give a conventional power supply connection. In recent years, a substitution for the conventional number system is the Residue Number System (RNS). RNS systems are efficient in terms of power, area, speed, etc. The power consumption of the circuit depends on several factors, one factor is how fast the circuit can perform the operations. In most digital applications adders play a crucial role. A novel mod adder based on Thermometer coding has been proposed, which yields superior results in comparison to the current state-of-the-art methods. The suggested addition process was designed by using thermometer coding. The proposed adder was simulated by using the NC launch tool in the Cadence.

Author e-mail: kvvardhan405@gmail.com,Sarada.marasu@gmail.com

How to cite this article: Vardhan VK, Musala S. Thermometer Coding-Based Application-Specific Efficient Mod Adder for Residue Number Systems, Journal of VLSI Circuits and System Vol. 6, No. 2, 2024 (pp. 122-129).

Introduction

Modern technology Enhancing energy efficiency is necessary for Deep Learning,[1] DSP,[2] and the Internet of Things.[3] With the use of arithmetic and conventional number systems, efficient embedded systems (ES) have recently been created, and their performance has been enhanced.[2] Applications such as Digital Signal Processing (DSP),[4] Cryptography,[5] and Residue Number System (RNS)[6] are employed to perform high-speed, fault-tolerant, and low-power calculations. Add and multiply are commonly used in the majority of RNS operations, such as add, subtract, divide, scaling, forward and reverse conversions.[2]. The conventional data is converted into residue data this process is called Forward Conversion. Similarly, residue data is converted into conventional data is called Reverse conversion. The second one is more difficult than the first conversion.

The unary coding, in which the size of the number is determined by the number of 1s it contains, is commonly used in thermometer coding (TC). One application of Golomb coding is in data compacting in neural networks.[7-8] One of the subclasses of Golomb coding is TC. Thermometer coding (TC) is a straightforward technique for enhancing the performance of RNS computing components, such as modular multipliers, subtractors, and adders.[9]

Engineers and researchers are becoming more interested in the design of modular adders that use TC.[10] It is legal to translate equivalent residues in TC format from analog inputs using current ADCs.[11] By adding TC numbers in RNS, one can add compact moduli faster than in traditional processes because there is no carry propagation involved. Forward converters are not needed for TC number addition because ADCs are employed to encode the input TC numbers.[11] Effective RNS systems require the selection of modest moduli. Applications in the form of embedded low-power devices and Internet of Things devices Thermometer Coding-based mod adders are appropriate since adders need a limited dynamic range.[2] Among the first unweighted number systems was RNS. One intrinsic feature of RNS is its capacity to do simple arithmetic operations, where the outcome of each numerical position is determined by the number that exists at that location. The moduli set is denoted as $\{p_1, p_2, \ldots, p_n\}$, where '$p_i$' represents the value of the ith modulus. The RNS is typically depicted as a collection of suitably chosen prime numbers referred to as the moduli.[12]

Residue Number System (RNS) represents each 'K' number as a collection of smaller numbers called residues. The value of each residue, designated by the symbol 'r_i', corresponds to the ith position. The residue 'r_i' is defined as the remainder obtained when 'K' is divided by the modulus value 'p_i' such that the remainder is minimized. In technical terms, the relationship might be characterized as,

$$|K|pi=r_i \tag{1}$$

Every integer 'K' that falls under the 'Dynamic Range (DR)' of RNS and is expressly expressed in it. The Dynamic Range is represented as the set of moduli $\{p_1, p_2, \ldots, p_n\}$, denoted as 'L'.

$$L=\Pi_{i=1}^{n} m_i \tag{2}$$

All integers in the interval '0' to 'L-1' have explicit definitions in RNS. After that, a number may develop similarly.[13]

Mod addition between two residue numbers U and V[14] as follows

$$T = U + V = (t_n, t_{n-1}, \ldots, t_1) \tag{3}$$

With

$$t_i = \begin{cases} u_i + v_i & if\, u_i + v_i < m_i \\ u_i + v_i - m_i & if\, u_i + v_i \geq m_i \end{cases} \tag{4}$$

Consider the RNS numbers for the above addition

$$U=(u_n,u_{(n-1)},\ldots,u_1\ ) \tag{5}$$

$$V=(v_n,v_{(n-1)},\ldots,v_1\ ) \tag{6}$$

Thermometer Coding (TC) is used in RNS systems, it will be called TCR. By using TC in RNS systems are less power consumption and energy efficient. The TC representation of digits from 0 to 5 by using 6 bits is shown in below Table 1.

In the TC format, the magnitude of the number increases as the number of needed bits increases. Likewise, the numerical value is low, and the number of bits needed is minimal. The number of occurrences of the digit 1 in the number can be positioned in any direction, although for optimal use of TC numbers, it is advisable to place the 1's on the right-hand side.TCRs enable the modular addition of two operands by concatenating and bit-shifting the concatenated augend and addend, then taking the modulo of the intermediate sum to keep the result inside the original length of the modular thermometer residue.

The reverse conversion was carried out using a variety of methods, including the Chinese Remainder Theorem (CRT),[15-17] Mixed Radix Conversion (MRC),[18] and the New Chinese Remainder Theorem.[19] The design of reverse converters heavily relies on mod adders. The CRT methodology is more commonly used than MRC due to its concurrent nature. Adders are designed these days with a variety of methods0[20], [21]

Table 1. TC representation of digits from 0 to 5

Decimal Digit	TC Representation
0	000000
1	000001
2	000011
3	000111
4	001111
5	011111

This study suggests a brand-new modular adder construction based on TCRs. The suggested modular adder consumes less power than the current works,[11], [22] which utilize 2X1 MUXs and many logic gates. The addition process was accomplished by using XOR gates.[23-27]

The structure of this document is as follows. First discussed the current and TC modular adders. Next discussed the anticipated TC-based mod adder. Later comparison between the proposed design and the existing works are compared. At the end concluded with some findings are discussed.

Literature

By retaining the distributed arithmetic paradigm, the TC-based modular adders are essential to TC-based modular multiplication.[14] Fig. 1 illustrates the design of the Mod 7 thermometer coding-based adder, as per reference.[21] One integer is given in binary form as b[1], b[2], & b[3] out of two numbers in the proposed adder. In this case, the positional weights of bits b[1], b[2], and b[3] are 1, 2, and 4, respectively. The first number's 1s should be added to the second (a[6]a[5]a[4]a[3]a[2]a[1]). If the sum of ones in the result exceeds the value of 'm- 1', then the number of beginning ones is subtracted from the sum.

Now, let us analyze an illustration.where the first number (A) is expressed in TC form (a[6]a[5]a[4]a[3]a[2]a[1] =

000111), the second number (B) is stated in binary form 101 (b[1] = 1, b[2] = 0 & b[3] = 1). The least significant binary digit (b[1]) affects the MUXs in the opening stage, moving the first binary digit to the left and adding one to the end of the least significant binary digit (rightmost bit position). If b[2] is '1', the second stage MUXs then move the input two bits to the left before appending two 1s to the end of the least significant bit position. Alternatively, during this phase, the MUXs simply add or shift zero values; they merely move them from the input into the output. The third stage's MUXs transfer inputs at zero to outputs similarly to the second stage. The MUXs in this stage left shift four times and add four ones to the end of the least significant bit position when b[3] is '0' or equal to one.

Therefore, the values of 'D' (d[12]d[11]d[10]d[9]d[8] d[7]d[6]d[5]d[4]d[3]d[2]d[1]) indicate the output of third stage MUXs. These numbers are 000011111111.

Depending on d[7], the adder's output is expressed in a certain way. Bits d[8] to d[12] are obtained via the final MUXs stage if d[7] = '1', indicating that the 'sum ≥ m'. In the last stage, MUXs result in the bits d[1] to d[6] if d[7] = '0', indicating that the sum is less than 'm'. d[7] wasthe selection signal utilized by the final stage MUXs.

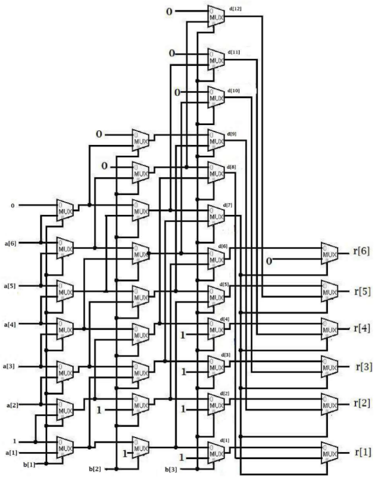


Fig.1: TCR-based Mod-7 Adder [21]

A novel TCR-based modular adder with reduced latency and circuit size was introduced in,[22] in contrast to the TC-based adder disclosed in.[21] This recommended adder takes TC-formatted inputs. Then, it is determined if 'P + Q ≥ m' or 'P + Q < m' by combining the two residues with moduli 'm'. Before calculating the sum, the bits that indicate the integer 'Q' are reversed. After that, the inputs are put via logical AND & NOR procedures. When an AND gate has at least one output that is 1, the outcome (P + Q ≥ m) occurs. To put it another way, if only one AND gate output is '1', total (P + Q = m) will result in '0'. This keeps on until a couple AND gates generate logic'1', which happens when the output also generates logic '1'. The function P + Q < m can be implemented using the logical gates NOR.

$$Sum = \begin{cases} \mathrm{SUM1}, & if\,k = 1\ (P + Q \geq m) \\ \mathrm{SUM0}, & if\,k = 0\ (P + Q < m) \end{cases} \tag{7}$$

where

$$k = \vee_{i=1}^{m-1} (p_i \wedge q_{m-i}) \tag{8}$$

$$SUM1 = (\textstyle\sum_{i=1}^{m-1} p_i \wedge q_{m-i}) - 1 \tag{9}$$

$$SUM0 = (m-1) - \left(\textstyle\sum_{i=1}^{m-1} (\overline{p_i \vee q_{m-i}})\right) \tag{10}$$

To use the values in arithmetic modulo 'm', P and Q are framed with 'm-1' - bits in TCR.

$$P = p_{m-1} \cdots p_2 p_1 = \underbrace{0 \ldots 0}_{p\ \text{zeros}} \underbrace{1 \ldots 1}_{p\ \text{ones}} \tag{11}$$

$$Q = q_{m-1} \cdots q_2 q_1 = \underbrace{0 \ldots 0}_{q\ \text{zeros}} \underbrace{1 \ldots 1}_{q\ \text{ones}} \tag{12}$$

The sum of the bits in P_{zeros} plus P_{ones} and Q_{zeros} plus Q_{ones}, respectively, in equations (11) and (12) is equal to the (m-1). when faced withindividual occurrences in the input numbers are added together, there's a chance the total will be greater than the mod value and require more storage space than is required. The second number in this case may be interpreted as flipped or reversed. To assess this scenario, the bitwise logical AND operation is performed between the variable 'P' and the complement of variable 'Q'.

$$P = \underbrace{0 \ldots 0}_{p\ \text{zeros}} \underbrace{1 \ldots 1}_{z} \underbrace{1 \ldots 1}_{p\ \text{ones-z}} \tag{13}$$

$$Q_{reverse} = \underbrace{1 \ldots 1}_{q\ \text{ones-z}} \underbrace{1 \ldots 1}_{z} \underbrace{0 \ldots 0}_{q_{\text{zeros}}} \tag{14}$$

$$P \wedge Q_{reverse} = \underbrace{0 \ldots 0}_{p\ \text{zeros}} \underbrace{1 \ldots 1}_{z} \underbrace{0 \ldots 0}_{q_{\text{zeros}}} \tag{15}$$

In this instance, P + Q ≥ m (16) because the ones of 'P' and the opposite version of 'Q' overlap. where the number of overlapping 1s is indicated by 'z'.

$$\begin{cases} P + Q < m, & if \quad z = 0 \\ P + Q = m, & if \quad z = 1 \\ P + Q > m, & if \quad z > 1 \end{cases} \tag{16}$$

Since the traditional addition of 'P' and 'Q' needs to be subtracted by mod ‘m’, the covering on bit short one can be utilized to get the desired result.

$$SUM1 = \underbrace{0\ldots0}_{q\text{ zeros}}\underbrace{0\ldots0}_{p\text{ zeros}}\underbrace{1\ldots1}_{\text{z-1}} \tag{17}$$

During the modular addition P+Q, the number of zeros is significant if the result is less than’m’. Like the previous example, the bitwise-OR operation between the flipped versions of ‘Q’ and ‘P’ is considered.

$$P = \underbrace{0\ldots0}_{p\text{ zeros-z}}\ \underbrace{0\ldots0}_{z}\underbrace{1\ldots1}_{p\text{ ones}} \tag{18}$$

$$Q_{reverse} = \underbrace{1\ldots1}_{q\text{ ones}}\underbrace{0\ldots0}_{z}\ \underbrace{0\ldots0}_{q_{\text{zeros-z}}} \tag{19}$$

$$P \vee Q_{reverse} = \underbrace{1\ldots1}_{Q\text{ ones}}\underbrace{0\ldots0}_{z}\underbrace{1\ldots1}_{p\text{ ones}} \tag{20}$$

The output contains (Pones + Qones) bits with z-zeros if P+Q < m.

$$SUM0 = \underbrace{0\ldots0}_{z}\underbrace{1\ldots1}_{q\text{ ones}}\underbrace{1\ldots1}_{p\text{ ones}} \tag{21}$$

The mod adder for the mod ‘7’ based on TCR is shown in Fig. 2.[22] Multiplexers (MUXs) and logic gates were the only components used in the construction of the adder. SUM1 stores the result of this adder when total ≥ mod value, while SUM0 stores the result when P+Q < mod value. Depending on the logic level on ‘Set’, MUX can produce either SUM0 or SUM1 as the adder’s output. The ‘Set’ was the NOR gate’s output, which received its inputs from AND gates’ outputs, which contained the inputs ‘P’ and ‘Q’, bit reversed.

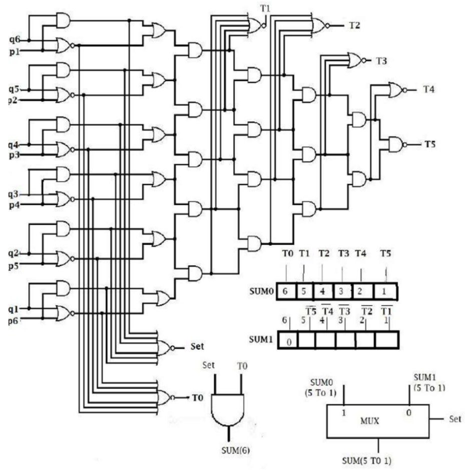


Fig. 2: Mod-7 Adder Based on TCR[22]

Another mod adder based on TCR was proposed in [27].In this how the adder performed the addition process was presented. Here, based on the Thermometer Coding idea, another mod adder that is more power-efficient than current adders is proposed. In this mod adder, input integers that participate in addition can be indicated in TC using m-bits. The following can be used to calculate the modular addition of two numbers using the flow chart shown in Fig. 3.

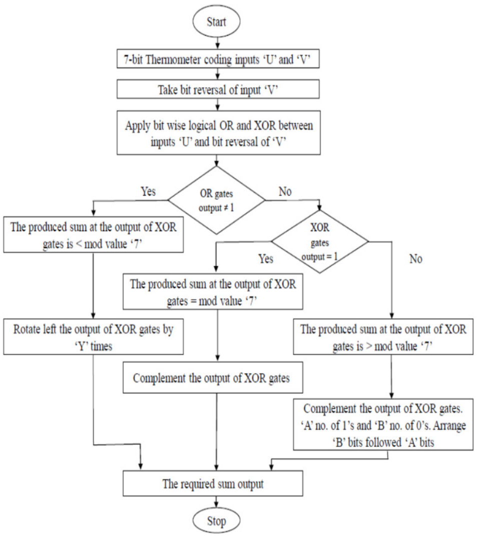
Once both inputs have been represented using Thermometer Coding (TC), the first number is regarded as ‘U’ and the second as ‘V’. Before applying the input, the bit inverted value for ‘V’ must be determined to conduct logical OR and bitwise modulo-2 addition (XOR) simultaneously. The following step process finds overlapping zeros, denoted by ‘p’.

The logic ‘p ≥ 1’ can be used to determine the sum (U + V < m). Then, rotate the output of the XOR gates to the left by ‘V’ times. As illustrated in the equation below, the outcome appears to be a sequence of overlapping zeros (p) followed by a total of 1s are present in both the input integers ‘U’ and ‘V’.

$$U + Vreverse = \underbrace{0\ldots0}_{p}\underbrace{1\ldots1}_{uones}\underbrace{1\ldots1}_{v\text{ ones}} \tag{22}$$

If the input integers do not have any overlapping zeros (p=0). This suggests that the sum is greater than the modulus’m’ when the output of any OR gate is non-zero. To be more precise, the output of all XOR gates is logic ‘1’ rather than the ‘sum=m’. The result is obtained by invertingthe XOR gates’ outputs. Assume for the moment the results are xn xn-1...x2 x1.

$$U + Vreverse = xn^{1}xn\text{-}1^{1}\ldots.x2^{1}x1^{1} \tag{23}$$



Start
7-bit Thermometer coding inputs ‘U’ and ‘V’
Take bit reversal of input ‘V’
Apply bit wise logical OR and XOR between inputs ‘U’ and bit reversal of ‘V’
Yes
OR gates output ≠ 1
No
The produced sum at the output of XOR gates is < mod value ‘7’
XOR gates output = 1
Yes
No
The produced sum at the output of XOR gates = mod value ‘7’
Rotate left the output of XOR gates by ‘Y’ times
The produced sum at the output of XOR gates is > mod value ‘7’
Complement the output of XOR gates
Complement the output of XOR gates. ‘A’ no. of 1’s and ‘B’ no. of 0’s. Arrange ‘B’ bits followed ‘A’ bits
The required sum output
Stop

Fig. 3: TCR-based Mod-7 Adder [27]

When zeros that intersect equal zero, that is a different case. This means that the result is made sum > modulusvalue 'm' if all OR gates give outputs that are not zero and all XOR gates give outputs that are not one. The outcome is added to the outputs of all XOR gates and sets up the 'A' number of 0's and 'B' number of 1's. This time, a certain number (B) of XOR gates makes zero, while the leftover number (A) of gates makes one.

$$U + Vreverse = \underbrace{0\ldots0}_{B\,zeros}\underbrace{1\ldots1}_{A\,ones} \qquad (24)$$

Proposed Model

One can review a new proposed TCR-based mod adder in this part. The suggested TCR-based mod adder resembles the TCR-based mod adder reported in [27]. In this proposed mod adder, the input numbers are represented with only 'm-1' bits, but in the case of [27], the inputs are represented with the 'm' number of bits. Because it uses a smaller number of bits, the proposed TCR-based mod produces better results than the mod adders mentioned in the previous section.

During the addition process, it is necessary to reverse the bits of the input 'V' before performing logical OR and XOR operations with the input 'U'. The logical OR operation is employed to validate the presence of overlapping zeros. If overlapping zeros exist, the resulting value is < 'm'. Otherwise, the resulting value is ≥ 'm'. In this scenario, the outcome can be achieved by left-rotating the output of the XOR gates by 'V' times. For a good understanding of the scenario, the following example is useful.

Example: Input Numbers U = 3 and V = 2 then the result of U + V.

Input U = 3 = 000111; Input V = 2 = 000011
The flipped version of V = 110000
-----------
Logical OR b/w U and Flipped version of V = 110111
Logical XOR b/w U and Flipped version of V = 110111
Rotate left XOR gates output by 'V' times = 011111 = 5

Another scenario in this addition process was, no overlapping zeros between the 'U' and bit reversal of 'V', at the same time the out of the XOR gates is one, for this scenario the result is straightway the outcome of the XOR gates which is (m-1).

Example: Input Numbers U = 4 and V = 2 then the result of U + V.

Input U = 4 = 001111; Input V = 2 = 000011
The flipped version of V = 110000
-----------
Logical OR b/w U and Flipped version of V = 111111
Logical XOR b/w U and Flipped version of V = 111111
The XOR gates output is the result = 111111 = 6

One more scenario in this addition process was, no overlapping zeros between 'U' and bit reversal of 'V', at the same time only one XOR gate output is equal to zero, which indicates the generated result = 'm'. To get the outcome, complement the XOR gates' output, which produces logic '1' at their output.

Example: Input Numbers U = 5 and V = 2 then the result of U + V.

Input U = 5 = 011111; Input V = 2 = 000011
The flipped version of V = 110000
-----------
Logical OR b/w U and Flipped version of V = 111111
Logical XOR b/w U and Flipped version of V = 101111
Complement the XOR gates output, whose output is '1', to get the result = 000000 = 0

The final scenario in this addition process was there were no intersecting zeros between 'U' and bit reversal of 'V'. In the addition process complement the XOR gates output at the time some 'M' no. of XOR gates generated logic '0', a 'N' no. of XOR gates generated logic '1'. For this case, get the result (M+1) number of zeros followed by (N-1) number of ones.

Example: Input Numbers U = 6 and V = 2 then the result of U + V.

Input U = 6 = 111111; Input V = 2 = 000011
The flipped version of V = 110000
-----------
Logical OR b/w U and Flipped version of V = 111111
Logical XOR b/w U and Flipped version of V = 001111
Complement the XOR gates output = 110000
The result of the addition = 000001 = 1

Fig. 4 displays the flow chart of the suggested TCR-based mod adder.

The pseudo system verilog code of the suggested TCR-based mod adder is as follows.

```
module model_1#(parameter N)(A,B,out);
input    [N-2:0] A;
input    [N-2:0] B;
output  reg [N-2:0] out;
----------
-----------
-----------
-----------
```

```
always@(*)
begin
  if(count_overlap_zeros > {(N/2){1'b0}})
  begin
----------
-----------
function [N-1:0] convert_input;
input  [N-1:0] in_1;
reg   [N-1:0] out_1;
integer l,m;
begin
----------
-----------
endfunction
function [(N/2)-1:0] count_zeros;
input   [N-1:0] in;
----------
-----------
end
endfunction
function [N-1:0] reverse_in;
input    [N-1:0] in1;
reg    [N-1:0] temp;
----------
-----------
endfunction
endmodule
```

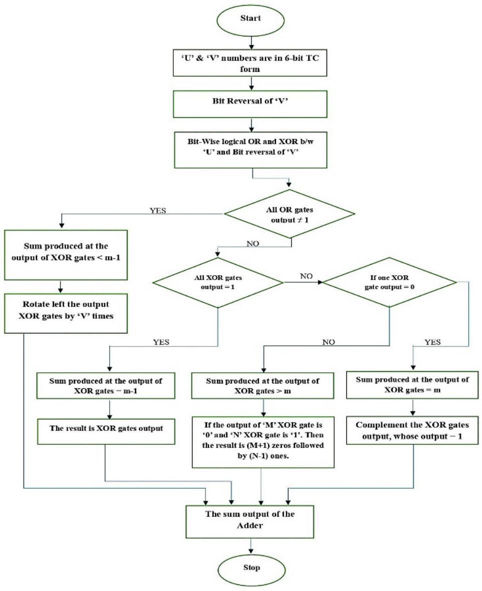

Start
'U' & 'V' numbers are in 6-bit TC form
Bit Reversal of 'V'
Bit-Wise logical OR and XOR b/w 'U' and Bit reversal of 'V'
All OR gates output ≠ 1
YES
NO
Sum produced at the output of XOR gates < m-1
Rotate left the output XOR gates by 'V' times
All XOR gates output = 1
NO
If one XOR gate output = 0
YES
NO
YES
Sum produced at the output of XOR gates = m-1
The result is XOR gates output
Sum produced at the output of XOR gates > m
If the output of 'M' XOR gate is '0' and 'N' XOR gate is '1'. Then the result is (M+1) zeros followed by (N-1) ones.
Sum produced at the output of XOR gates = m
Complement the XOR gates output, whose output = 1
The sum output of the Adder
Stop

Fig. 4: Proposed TCR-based Mod Adder

Experimental Results

The current mod adder and recommended one based on the TC model are simulated using the Cadence NC launch simulation analysis environment. The delay, total power consumption, area, power delay product (PDP), and area delay product (ADP) comparison of the TCR-based modulo adder, the proposed 90nm TCR-based mod adder design in the Encounter tool, and the existing TCR-based mod adder are shown in Tables 2, 3, 4,5, and 6. According to the Tables, the suggested design outperforms current solutions with acceptable areas and delays in terms of power consumption and power delay product.

The proposed modulo adder consumes less power compared to the existing system. One of the main reason for this is the number bits are used for the representation of the input numbers in thermometer coding. The number of bits used in the proposed modulo adder 'm' adder is (m^{-1})-bits. The operation of the proposed adder is simple.

Table 2: Comparison of Delay (ns)

Mod Value	Suggested	Mod Adder [27]	Mod Adder [22]	Mod Adder [21]
7	1.56	1.93	1.06	2.09
8	1.64	1.98	1.21	2.30
9	2.35	2.88	1.44	3.51
11	2.46	2.98	1.76	3.86
13	3.98	4.48	2.06	5.31

Table 3: Comparison of Total Power (μW)

Mod Value	Suggested	Mod Adder [27]	Mod Adder [22]	Mod Adder [21]
7	3.16	3.88	6.23	16.62
8	2.10	2.35	7.43	11.76
9	2.18	2.69	8.90	16.69
11	2.75	3.08	13.32	21.78
13	4.98	6.97	23.88	24.74

Table 4: Comparison of Area in (μm^2)

Mod Value	Suggested	Mod Adder [27]	Mod Adder [22]	Mod Adder [21]
7	301.32	375.42	255.08	265.26
8	395.65	448.84	326.22	348.25
9	410.36	498.04	406.46	425.21
11	581.23	600.97	602.49	623.54
13	687.43	763.71	829.56	840.96

Table 5: Comparison of PDP in (μW * ns)

Mod Value	Suggested	Mod Adder [27]	Mod Adder [22]	Mod Adder [21]
7	4.93	7.49	6.6	34.74
8	3.44	4.65	8.99	27.05
9	5.12	7.75	12.82	58.58
11	6.76	9.18	23.44	84.07
13	19.82	31.23	49.19	131.37

Table 6: Comparison of ADP in (μm² * ns)

Mod Value	Suggested	Mod Adder [27]	Mod Adder [22]	Mod Adder [21]
7	470.06	724.56	270.38	554.39
8	648.87	888.70	394.73	800.98
9	964.35	1434.35	585.30	1492.49
11	1429.83	1790.89	1060.38	2406.86
13	2735.97	3421.42	1708.89	4465.50

The RTL Schematic of the proposed adder is shown in below Fig. 5

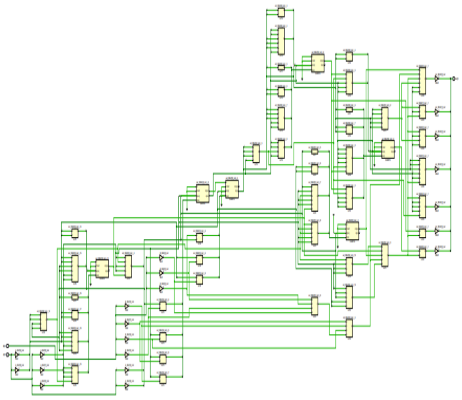


Fig. 5: RTL Schematic of the Proposed Mod - 7 Adder

The simulation results of the proposed TCR-based mod adder (mod value = 7) are shown in Fig. 6 and its layout is shown in Fig. 7.

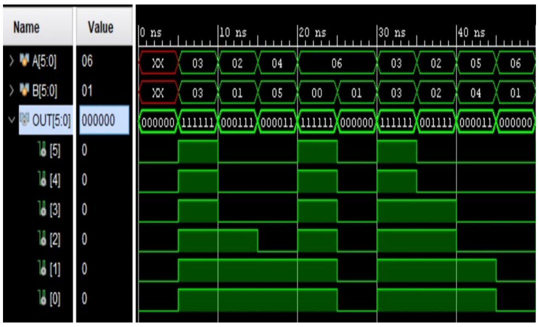


Fig. 6: Suggested Mod Adder Simulation Result

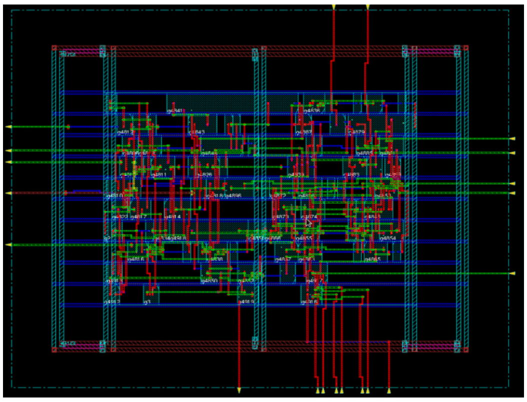


Fig. 7: Layout of the Proposed Mod Adder

Conclusion

The proposed TCR-based mod adder in this paper gives good results in contrast to the existing similar mod adders based on TCR in terms of Delay, Total Power Consumption, Area, Power Delay Product (PDP), and Area Delay Product (ADP). By considering all these results of the suggested mod adder based on TCR it was suitable for applications like Embedded Systems, Digital Signal Processing, Digital Image Processing, Cryptography, communications, etc., In these applications power consumption, area, and delay are crucial.

References

[1] Chen Y.H, Krishna T, Emer J. S, Sze V, Eyeriss. An energy-efficient reconfigurable accelerator for deep convolution neural networks.In:IEEE International Solid State Circuits conference2016; 53, 1: 127-138.

[2] Molahosseini A. S, Sousa L, Chang C. H (Eds.). Embedded Systems Design withSpecial Arithmetic and Number Systems, New York: Springer, 2017.

[3] Alioto M (Ed). Enabling the Internet of Things: from Integrated Circuits to Integrated Systems, New York: Springer, 2017.

[4] Chang C. H, Molahosseini A. S, Zarandi A. A. E, Tay T. F. Residue Number Systems: A new paradigm to data path organization for low power and high-performance digital signal processing applications. IEEE Circuits and Systems Magazine 2015; 15, 4: 26-44.

[5] Rahim, Robbi. "Review of Modern Robotics: From Industrial Automation to Service Applications." Innovative Reviews in Engineering and Science 1.1 (2024): 34-37.

[6] Ananda Mohan P. V. Residue Number Systems: Theory and Applications, New York: Springer, 2016.

[7] Golomb S. W. Run-length encodings (Correspond). IEEE Transactions Information Theory 1996; 12, 3: 399-401.

[8] Kumar, TM Sathish. "Low-Power Communication Protocols for IoT-Driven Wireless Sensor Networks." Journal of Wireless Sensor Networks and IoT 1.1 (2024): 24-27.

[9] Abdullah, Dahlan. "Leveraging FPGA-Based Design for High-Performance Embedded Computing." SCCTS Journal of Embedded Systems Design and Applications 1.1 (2024): 29-32.

[10] Vun C. H, Premkumar A. B. Thermometer code based modular arithmetic, In Spring Congress. Eng. Technology, May 2012, 534-38.

[11] Vun C. H, Premkumar A. B. RNS encoding based folding ADC, In IEEE Int. Symp. Circuits, May 2012, 814-17.

[12] K Vijaya Vardhan, M. Sarada, K. M. Santhoshini, A. Srinivasulu. A Critical Look at Modular Adders using Residue Number Systems, Solid State Electronic Letters 2019; 1, 2: 84-91.

[13] Premkumar B, Omondi A. Residue Number Systems: Theory and Implementation, New York: (vol 2) World Scientific, 2007.

[14] Patel R, Benaissa M, Boussakta S. Fast Modulo 2n – (2n-2 + 1) addition: a new class of adder for RNS, IEEE Transactions on Circuits and Systems I 2007; 56,6: 1279-92.

[15] Jaberipur G, Parhami B, Nejati S. On building general modular adders from standard binary arithmetic components, In 45thAsilomar Conf. Signals Systems and Computers, 2011, 6-9.

[16] Ma S, Hu J, Wang C. H. A novel modular adder for 2n – 2k – 1 residue number system, IEEE Transactions on Circuits and Systems 2013; 60, 11: 2962-72.

[17] K. VijayaVardhan, G Sailakshmi, M. Sarada, A. Srinivasulu. Analysis and Design of High-SpeedResidue to Binary Reverse Converter for the Moduli Set {2n + 1, 2n, 2n – 1}, In IEEE Recent Advances in Intelligent Computational Systems, December 2020,53-60.

[18] Hiasat A. General Modular Adder designs for Residue Number System Applications, IET Circuits Devices Systems 2018; 12, 4: 424-31.

[19] White S. A. Applications of distributed arithmetic to digital signal processing: A tutorial review, IEEEASSP Magazine 1989; 6, 3: 4-19.

[20] J. K. Saini, A. Srinivasulu, B. P. Singh. A new low-power full-adder cell for low voltage using CNTFETs, In IEEE International Conference on Electronics, Computers, and Artificial Intelligence 2017, 6-10.

[21] T. Nagateja, T. Venkata Rao, A. Srinivasulu. Low Voltage, High-Speed FinFET based 1-bit BBL-PT Full Adders, In IEEE International Conference on Communication and Signal Processing 2015, 1247-51.

[22] Jafarzadehpour F, A. S. Molahosseini, A. A. E. Zarandi, L. Sousa. Efficient Modular Adder Designs Based on Thermometer and One-Hot Coding, IEEE Transactions on Very Large Scale Integration Systems 2019; 27, 9: 2142-55.

[23] P. V. Lakshmi, M. Sarada, A. Srinivasulu, D. Pal. Three Novel Single-Stage Full-Swing 3-input XOR, International Journal of Electronics 2018; 105, 8: 1416-32.

[24] M. Sarada, A. Srinivasulu, D. Pal. Novel Low-Supply, Differential XOR/XNOR with Rail-to-Rail Swing for Hamming Code Generation, International Journal of Electronics Letters 2018; 6, 3: 272-87.

[25] M. Sarada, A. Srinivasulu. Two DCVSL XOR/XNOR Circuits Operated at Sub-Threshold Voltages Using MOSFETS and FinFETs, Journal of Advanced Research in Dynamical and Control Systems 2017; 9, 2: 123-134.

[26] Green, K., and R. Vrba. “Research on Nano Antennas for Telecommunication and Optical Sensing.” National Journal of Antennas and Propagation 6.2 (2024): 1-8.

[27] K Vijaya Vardhan, and Sarada Musala, “Ultra-Low-Power Modulo Adder with Thermometer Coding for Uncertain RNS Applications”, Journal of Uncertain Systems 2022, 15, 3, 12-20.