Abstract— Diminished-one modulo $2^{n+1}$ Adder using Circular carry selection (CCS) is an important building block of RNS based DSP system. In this paper, we have presented a Diminished-one modulo $2^{n+1}$ adder using CCS scheme. The architecture design of Diminished-one modulo $2^{n+1}$ adder using CCS consists of a Dual Sum-Carry Look Ahead Adder (DS-CLA), a Circular Carry Generator (CCG) and a Multiplexer (MUX). The adder has been simulated using verilog HDL codes and mapped this design to the TSMC (180nm) implementation technology using the Synopsys Design Compiler and calculated the area, power Dissipation and Time delay for $n = 8, 12, 16, 24, 32, 48, 64$. We have compared our results with Select-Prefix method for $n = 8, 16, 32$ and 64 and found that the area occupied is lesser than Select-Prefix Method.

Index Terms— VLSI Design, Residues number system (RNS), Modulo $2^{n+1}$ adder, Diminished-one, Circular carry selection (CCS)

I. INTRODUCTION

RESIDUE number system (RNS) is a non-weighted number system and it works well in the area where key operation required are addition, subtraction and multiplication such as digital filter, FFT, Image processing and digital communication [1], [2]. In RNS, addition, subtraction and multiplication are inherently carry free, i.e. each digits of the result is a function of only one digit of each operand, hence independent of all the other digits so complexity is reduced and operation can be executed in less time [1], [2]. In a standard Residue number system, RNS is defined by a set of pair-wise relative prime integer, $\{p_1, p_2, \ldots, p_n\}$ called moduli. A number $R$ can be converted into residue representation $\{r_1, r_2, \ldots, r_n\}$, where $r_1, r_2, \ldots, r_n$ are the least positive remainders when dividing $R$ by the $p_1, p_2, \ldots, p_n$. It is denoted by $r_i = R \mod p_i$ for $i = 1, \ldots, n$ [1], [2]. Residues arithmetic operation of two numbers $A$ and $B$ is defined by $(C_1, C_2, \ldots, C_n) = (A_1, A_2, \ldots, A_n) \# (B_1, B_2, \ldots, B_n)$, where $\#$ is either ‘addition’, ‘subtraction’ or ‘multiplication’ and $A$ and $B$ are residue representation of $A$ and $B$.

There are many moduli sets like $(2^1, 2^2, 2^3)$ [3], $(2^1, 2^2, 2^3+1, 2^{2n+1})$ [4], $(2^1, 2^2, 2^3+1, 2^{2n+1}+1)$ [5] etc. The most well-known three-moduli RNS that uses a base of the form $(2^1, 2^2, 2^3+1)$ has received significant attention, mainly due to the existence of very efficient combinational converters from/to the binary system. A $2^{n+1}$ channel is also an integral part of the 5-moduli RNS proposed in [6]. From the above it can be said that the design of efficient modulo $2^{n+1}$ adder is vital in RNS-based applications.

Modulo $2^{n+1}$ channel handles $n+1$ bit input where as modulo $2^2$ and $2^n$ type can handle only $n$ bit input operands. So, the implementation of modulo $2^{n+1}$ channel is more complicated than $2^2$ or $2^n$ type channel [7]. To remove this problem, we use Diminished-one arithmetic proposed by Leibowitz [8]. In Diminished-one arithmetic, the inputs $A \& B$ are represented by $A^*$ and $B^*$ where $A^* = A-1$ and $B^* = B-1$ so the concept of diminished-one makes $n+1$ bits to $n$ bits wide.

Many papers have been presented on Diminished-one modulo $2^{n+1}$ addition [7], [9]-[11].

Here, we have presented a paper on Diminished-One modulo $2^{n+1}$ adder Using CCS scheme which is based on paper [7]. This adder consists of a DS-CLA, a CCG and a MUX. The DS-CLA adder gives two modulo results in parallel and the CCG computes the carry out bit and circularly controls the MUX to find the correct modulo sum from DS-CLA adder.

The paper has been organized in four sections. In section II, the details of the Diminished-one modulo $2^{n+1}$ adder using CCS is presented. Section III gives the performance parameters comparison and the conclusion part is given in section IV.

II. DIMINISHED-ONE MODULO $2^{n+1}$ ADDER USING CIRCULAR CARRY SELECTION SCHEME

We have analyzed the Modulo adder using two methods, these methods are:

A. Direct equation Implementation Method

Let $A' = a_{n-1}, \ldots, a_0^*$ and $B' = b_{n-1}, \ldots, b_0^*$ are two numbers represented in Diminished-one format. The modulo $2^{n+1}$ addition of $A'$ and $B'$ is given by $S' = s_{n-1}, \ldots, s_0^*$ where

$$S' = | A' + B' + c_{n-1}^- |$$  \hspace{1cm} (1)
Here $c_{n-1}$ is carry out bit of $(A^* + B^*)$ and given by equation as proposed by [7].

$$c_{n-1} = g_{n-1}^* + \sum_{j=0}^{n-2}(\prod_{k=j+1}^{n-1} p_k) g_j^*$$ (2)

According to CLA function we take two terms.

**Carry Generate term** $g_i^* = a_i^* \cdot b_i^*$

**Carry Propagate term** $p_i^* = a_i^* \oplus b_i^*$

Where $i = 0$ to $n-1$.

The diminished-one operators are

- $S^* = S - 1$
- $g_i^* = g_i - 1$
- $p_i^* = p_i - 1$

The block diagram of CCS Diminished-one modulo $2^n+1$ adder is shown in figure (1). The input to this block is $A^*$ and $B^*$ of $n$ bit each. These inputs are applied to DS-CLA adder circuit. The DS-CLA adder produces two generate ($g^*$) and propagate ($p^*$) terms and gives three outputs, one is applied to CCG which is used to produce the carry-out bit $c_{n-1}$, and the other two outputs are $S^*_{i,1}, S^*_{i,0}$. The carry-out bit $c_{n-1}$ is connected to an inverter and applied as a control signal to the MUX. If the $c_{n-1}$ is '0' the output of MUX is $S^*_{i,0}$ otherwise the output is $S^*_{i,1}$. The output of MUX is connected to XOR gate. The other input to XOR gate is propagate term $p_i^*$ and finally we get $S^* = s^*_{n-1}, \ldots, s^*_{1}, s^*_{0}$ which is given by the following equation as proposed by [9]:

$$S^*_{i,1} = \begin{cases} g_{i-1}^* + \left( \sum_{j=0}^{i-2} \left( \prod_{k=j+1}^{i-1} p_k^* \right) g_j^* \right) \oplus p_i^* & \text{if } c_{n-1} = 0 \\ g_{i-1}^* + \left( \sum_{j=0}^{i-2} \left( \prod_{k=j+1}^{i-1} p_k^* \right) g_j^* \right) \oplus p_i^* & \text{if } c_{n-1} = 1 \end{cases}$$ (3)

For $n = 4$, the equations are

$$S^*_{2,1} = (g_2^* + (p_2^* \cdot p_1^* \cdot g_0^*) + (p_2^* \cdot g_1^*)) + (p_2^* \cdot p_1^* \cdot p_0^*) \oplus p_3^*$$
$$S^*_{2,0} = (g_2^* + (p_1^* \cdot g_0^*) + (p_1^* \cdot p_0^*)) \oplus p_2^*$$
$$S^*_{1,1} = (g_1^* + p_0^*) \oplus p_1^*$$
$$S^*_{1,0} = p_0^*$$
$$S^*_{0,1} = (g_0^* + (p_2^* \cdot p_1^* \cdot g_0^*) + (p_2^* \cdot g_1^*)) \oplus p_3^*$$
$$S^*_{0,0} = (g_0^* + p_1^* \cdot p_0^*) \oplus p_2^*$$

The equations for $S^*_{1,0}$ and $S^*_{1,1}$ are same except the AND operation of propagate terms $p_i^*$ in $S^*_{1,1}$ and $S^*_{0,1}$ is the compliment of $S^*_{0,0}$.

The complete circuit of CCS Diminished-one modulo $2^n+1$ adder for $n=4$ is shown in fig. (2) as proposed by Tso-Bing Juang, Ming-Yu Tsai and Chin-Chieh Chiu [9].

**B. Partitioning circuit for large dimension**

For the large dimension of $n$, we partition the $n$ bit CCS modular adder into $m, r$ bits, which are given in paper [7].

Here $n = m \times r$

Where $m$ is no. of DS-CLA blocks
r = no. of inputs in a block
Both input data are divided into m block inputs

\[ A^* = a_{m-1}, \ldots, a_{0} \]
\[ B^* = b_{m-1}, \ldots, b_{0} \]
And Sum
\[ S^* = s_{n+1}, s_{n}^* \]
\[ = A^* + B^* + k^*_{r,i} \]

Where
\[ k^*_{i,j} \] is carry-out bit
\[ A^* = a_{m-1}, \ldots, a_{0}, a^*_p \]
\[ B^* = b_{m-1}, \ldots, b_{0}, b^*_p \]
for \( t = 0, \ldots, m-1 \)

Where \( k^*_{r,i} \) represents the carry-out bit of the \((i-1)^{th}\) addition block. In each r-bit CCS addition block, the DSCLA Adder generates two block sums \( S^*_r = S^*_r \) for \( k^*_{r-1} = 0 \) and for \( k^*_{r-1} = 1, S^*_r = S^*_r \). So we can say that the carry-out bit \( k^*_{r,i} \) is used to select the correct block sum. Each carry-out bit \( k^*_{r-1} \) generated by CCG can be understood by these equations proposed by [7] as follows:

\[ k^*_{r-1} = \left[ G^*_i + \left\{ \sum_{j=0}^{\lceil \frac{r}{2} \rceil} \left( \prod_{l=j+1}^{t-1} p_l \right) G^*_j + c_{n-1} \prod_{l=j+1}^{t-1} p_l \right\} \right] \]
\[ = \begin{cases} k^*_{r-1} \, & \text{if} \ c_{n-1} = 0 \\
\left( G^*_i + \sum_{j=0}^{\lceil \frac{r}{2} \rceil} \left( \prod_{l=j+1}^{t-1} p_l \right) G^*_j \right) \, & \text{if} \ c_{n-1} = 1 \end{cases} \]

The direct equation method is expressed as CCS and Partitioning methods are expressed as CCG is lesser than Select-Prefix method. In paper, proposed by S.-H. Lin and M.-H. Sheu[7] they use the UMC 180nm technology on the Cadence PKS and Silicon Ensemble tool. They have used only Partitioning method for the same value of n. However, there are some corrections in paper [7] which is proposed in paper [9].

III. PERFORMANCE PARAMETERS EVALUATION

The CCG Diminished-one modulo 2\(^{r+1}\)+1 adder has been simulated using Verilog code and mapped this design to the TSMC 180nm implementation technology using the Synopsys Design Compiler for \( n = 8, 12, 16, 24, 32, 48, 64 \). For this, we have written Verilog HDL programs for Direct equation implementation method and for Partitioning circuit. Then calculated area, power dissipation and time delay using the Synopsys design compiler tool and shows the AT (Area \( \times \) Delay) and TP (Delay \( \times \) Power) products. All the area results are expressed in \( \mu m^2 \), delay results are based on the assumption of worst case commercial model which is given in ns and power dissipation results are expressed in mW.

The performance parameters and the AT and TP products in tabular form are shown in Table (1). The Direct equation methods are expressed as CCS and Partitioning methods are expressed as CCG \( m \times n \). In the paper, proposed by S.-H. Lin and M.-H. Sheu[7] they use the UMC 180nm technology on the Cadence PKS and Silicon Ensemble tool. They have used only Partitioning method for the same value of n. However, there are some corrections in paper [7] which is proposed in paper [9]. For comparison of the results there is no data in paper [9], so we have compared our results with the work proposed by C. Efstatthiou, H. T. Vergos and D. Nikolos [10]. They have verified their result using Hardware Description language and mapped this design to the VST Diplomat technology 250nm implementation technology using the Synopsys Design Compiler. Table (2) shows the comparison of area and time delay of CCG Diminished-one modulo 2\(^{r+1}\)+1 adder using Equation method, Partitioning method and Select-Prefix method as proposed in paper [10] for \( n = 8, 16, 32, 64 \). Area of CCG Diminished-one modulo adder using CCG is lesser than Select-Prefix method.
dissipation and time delay are increasing.

\[ n = 8, 16, 32 \text{ and } 64. \]

Partitioning method is lesser than Select-Prefix method for diminished-one modulo adder using Equation method and the same value of \( n \). From Table (1), it is clear that AT and TP for \( n = 8 \) and 12, Equation method gives a lesser AT and TP method. From Table (2) it is clear that area of \( n = 8 \) and 12, Equation method gives a lesser AT and TP method.

**Table I**

<table>
<thead>
<tr>
<th>( n )</th>
<th>Architecture</th>
<th>Area(( \mu m^2 ))</th>
<th>Power(mW)</th>
<th>Delay(ns)</th>
<th>AT</th>
<th>TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CCS 2×4</td>
<td>1124.32</td>
<td>1.87</td>
<td>2.29</td>
<td>2574.69</td>
<td>4.28</td>
</tr>
<tr>
<td></td>
<td>CCS</td>
<td>771.73</td>
<td>1.38</td>
<td>2.46</td>
<td>1898.46</td>
<td>3.40</td>
</tr>
<tr>
<td>12</td>
<td>CCS 3×4</td>
<td>1792.93</td>
<td>2.99</td>
<td>2.53</td>
<td>4536.11</td>
<td>7.57</td>
</tr>
<tr>
<td></td>
<td>CCS</td>
<td>1437</td>
<td>2.36</td>
<td>2.66</td>
<td>3822.42</td>
<td>6.28</td>
</tr>
<tr>
<td>16</td>
<td>CCS 4×4</td>
<td>2408.31</td>
<td>4.01</td>
<td>2.54</td>
<td>6117.11</td>
<td>10.19</td>
</tr>
<tr>
<td></td>
<td>CCS</td>
<td>2142.2</td>
<td>3.52</td>
<td>3.05</td>
<td>6533.71</td>
<td>10.74</td>
</tr>
<tr>
<td>24</td>
<td>CCS 2×12</td>
<td>4031.6</td>
<td>6.58</td>
<td>2.63</td>
<td>10603.11</td>
<td>17.31</td>
</tr>
<tr>
<td></td>
<td>CCS 6×4</td>
<td>3702.28</td>
<td>6.54</td>
<td>2.97</td>
<td>10995.77</td>
<td>19.42</td>
</tr>
<tr>
<td></td>
<td>CCS</td>
<td>3688.98</td>
<td>5.95</td>
<td>3.95</td>
<td>14571.47</td>
<td>23.50</td>
</tr>
<tr>
<td>32</td>
<td>CCS 2×16</td>
<td>5102.70</td>
<td>8.98</td>
<td>2.74</td>
<td>19084.01</td>
<td>33.59</td>
</tr>
<tr>
<td></td>
<td>CCS 8×4</td>
<td>3702.28</td>
<td>9.01</td>
<td>2.96</td>
<td>10958.75</td>
<td>26.67</td>
</tr>
<tr>
<td></td>
<td>CCS</td>
<td>5305.61</td>
<td>8.51</td>
<td>4.24</td>
<td>22495.79</td>
<td>56.08</td>
</tr>
<tr>
<td>48</td>
<td>CCS 4×12</td>
<td>7324.73</td>
<td>12.91</td>
<td>4.53</td>
<td>33181.03</td>
<td>58.48</td>
</tr>
<tr>
<td></td>
<td>CCS 12×4</td>
<td>8259.45</td>
<td>13.46</td>
<td>2.97</td>
<td>24530.57</td>
<td>39.98</td>
</tr>
<tr>
<td></td>
<td>CCS</td>
<td>9244.07</td>
<td>14.24</td>
<td>4.89</td>
<td>45203.5</td>
<td>69.63</td>
</tr>
<tr>
<td>64</td>
<td>CCS 4×16</td>
<td>10185.44</td>
<td>17.99</td>
<td>3.37</td>
<td>34324.93</td>
<td>60.63</td>
</tr>
<tr>
<td></td>
<td>CCS 16×4</td>
<td>12088.14</td>
<td>19.21</td>
<td>3.05</td>
<td>36688.83</td>
<td>58.59</td>
</tr>
<tr>
<td></td>
<td>CCS</td>
<td>14812.46</td>
<td>21.5</td>
<td>5.86</td>
<td>86801.02</td>
<td>125.99</td>
</tr>
</tbody>
</table>

**IV. CONCLUSION**

In this paper, we have implemented the CCS Diminished-one modulo 2⁴+1 adder using two methods: first using Direct equation method and the second using Partitioning method. From the result shown in Table (1), it is clear that for \( n = 8 \) and 12, Equation method gives a lesser AT and TP product and for larger value of \( n \), Partitioning method gives a better result. The shaded portion shows the best result for the same value of \( n \). From Table (2) it is clear that area of Diminished-one modulo adder using Equation method and Partitioning method is lesser than Select-Prefix method for \( n = 8, 16, 32 \) and 64. By increasing value of \( n \), area, power dissipation and time delay are increasing.

**REFERENCES**