



# DESIGN OF MODULO $2^n+1$ ADDER FOR APPLICATION IN CONVOLUTIONAL ENCODER

M. Jayasanthi, T. Karthik

Assistant Professor, Department of ECE, Karpagam College of Engineering, Coimbatore, India

**ABSTRACT:** The modulo  $2^n+1$  adders have been used extensively in the field of Wireless Communications and Signal Processing Applications. They find their application especially in Convolutional Encoders. The proposed system is a low complexity design of modulo  $2^n+1$  adder derived by decomposition of parallel prefix computation into several blocks of smaller input bit-widths. Our proposed adders can produce modulo sums within the range  $\{0, 2n\}$ , which is more than the range  $\{0, 2n - 1\}$  produced by existing diminished-1 modulo  $2^n + 1$  adders (with some delay constraints). In this paper, we use a novel enhanced circular carry generation to process the carry bit, to obtain the final modulo sum efficiently in terms of area delay product. The performance of the adders is evaluated and verified using Verilog structuring HDL.

**KEYWORDS:** Parallel Prefix, circular carry generation

## I. INTRODUCTION

### II.

Wireless communication plays a major role in today's world. They find their applications in battlefield intelligence, surveillance, reconnaissance, security monitoring, emergency response, disaster rescue, environmental tracking, telemedicine, and multimedia systems in consumer electronics. Modulo  $2^n + 1$  arithmetic has been used in a variety of applications, ranging from pseudorandom number generation, Wireless Communication and cryptography up to convolution computations without round-off errors. A channel performing arithmetic operations modulo  $2^n + 1$  is most commonly an integral part of almost every residue number system (RNS). The RNS is an arithmetic system well-suited to applications in which the operations are limited to addition, subtraction and multiplication. The adoption of an RNS can offer significant speedup over the binary system and has therefore been reported in the design of digital signal processors, FIR filters and communication components [1].

The designed modulo  $2^n+1$  adder finds its application in Convolutional Encoder. The Convolutional Encoder is the one which has 'n' modulo-2 adders, and 'n' generator polynomials one for each adder. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs 'n' bits. They are used mainly in digital communication.

Residue number systems (RNS), which have been studied since the early days of digital computers, have again come into the forefront in view of improved algorithms for some of the difficult operations and the desire to use plentiful transistors for performance improvement without an undue energy burden. Moduli of the form  $2n \pm 1$ , which greatly simplify certain RNS arithmetic operations have also been of longstanding interest. Over the years, many articles have been published on the design of modulo- $(2n \pm 1)$  adders (e.g.,) and other arithmetic circuits, with new proposals still appearing on a regular basis [2].

In Section-II, we describe the architecture of Diminished  $2^n+1$  Adder. In Section-III, we have reviewed the existing architectural design of the Modulo-Adders. In Section-IV, we have proposed our design of Weighed modulo

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

$2^n+1$  adder with correction scheme which can be implemented in Convolutional Encoder. The synthesis and analysis results are discussed in Section V. In Section VI, the conclusion and future prospects of the proposed work are discussed briefly.

## II. ARCHITECTURE OF DIMINISHED $2^N+1$ ADDER

In this paper, we show that a diminished-1 adder with minor modifications can be also used for the modulo  $2n+1$  addition of  $n$ -bit weighted operands, provided that it is driven by operands whose sum has been decreased by 1. The architecture of Diminished  $2^n+1$  Adder is shown in Fig 1. The required modifications do not increase the execution delay of the adder and can be implemented in low area. Since currently, the most efficient architectures for a diminished-1 adder outperform those for weighted operands in both area and delay terms, the augmented diminished-1 adder can be used very efficiently if the decreased sum of the input operands can be easily derived. We show that this is the case in both the residue generators and the weighted multioperand modulo adders cases [3].

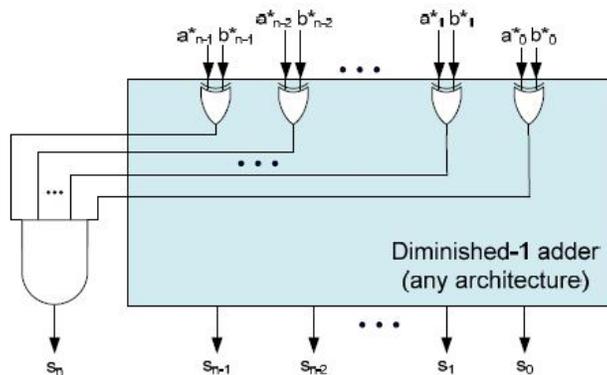


Fig. 1. Architecture of Diminished  $2^n+1$  adder

## III. Review and Analysis of previously proposed weighed modulo $2^n+1$ Adders

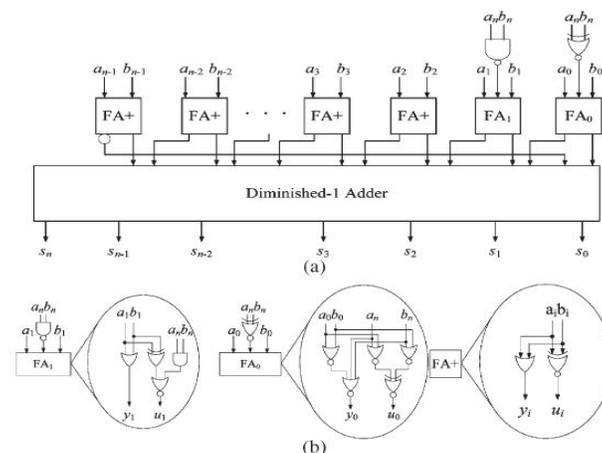


Fig. 2. (a) Architecture of weighed modulo  $2^n+1$  adder with Diminished-1 Adder. (b) Individual Architecture of  $FA_1$ ,  $FA_0$ , and  $FA+$

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

Given two  $(n + 1)$ -bit numbers  $A$  and  $B$ , where  $0 \leq A, B \leq 2n$ , the values of diminished-1 of  $A$  and  $B$  are denoted by  $A^* = A - 1$  and  $B^* = B - 1$ , respectively. The diminished-1 sum  $S^*$  can be computed by

$$S^* = |S - 1|_{2^{n+1}} = |A + B - 1|_{2^{n+1}} = |A^* + B^*|_{2^n} + \bar{c}_{out} \tag{1}$$

where  $|X/Z|$  is defined as modulo  $Z$  of  $X$ , and  $c_{out}$  is denoted as the inverted end-around carry of the diminished-1 modulo  $2^n$  sum of  $n$ -bit  $A^*$  and  $B^*$ .

### A. Vergos and Efstathiou [4]

In [4], the authors first compute the congruent modulo sum of  $A+B$  to produce  $Y$  and  $U$ , and then, the final modulo sum is performed by any diminished-1 modulo adder as follows. Suppose  $A$  and  $B$  are two  $(n + 1)$ -bit input numbers, i.e.,  $A = a_n a_{n-1} \dots a_0 = a_n \times 2^n + A_n$  and  $B = b_n b_{n-1} \dots b_0 = b_n \times 2^n + B_n$ , where  $0 \leq A, B \leq 2n$ , and  $A_n$  and  $B_n$  are two  $n$ -bit numbers; then

$$\begin{aligned} |A + B|_{2^{n+1}} &= ||A_n + B_n + D + 1|_{2^{n+1}} + 1|_{2^{n+1}} \\ &= |Y + U + 1|_{2^{n+1}}. \end{aligned} \tag{2}$$

In (2),  $D = 2n - 4 + 2c_{n+1} + s_n$ , which is equivalent to  $1111, \dots, c_{n+1}s_n$ , where  $c_{n+1} = a_n \cdot b_n$  ( $\cdot$  is denoted as the logic AND operation), and  $s_n = a_n \oplus b_n$  ( $\oplus$  is denoted as the logic EXCLUSIVE-OR operation) is the bit of  $D$  with binary weights  $2^1$  and  $2^0$ , respectively. The first step of (2) computes modulo  $2n + 1$  carry-save addition, giving the carry vector  $Y$  and the sum vector  $U$ , where  $Y = y_{n-2} y_{n-3} \dots y_0 y_{n-1}$  and  $U = u_{n-1} u_{n-2} \dots u_0$  are produced by adding  $A_n, B_n$ , and  $D$ , respectively. It can be seen that the values of  $D$  with binary weights of  $2^2$  through  $2^{n-1}$  are all 1, which can simplify the design of adders to produce the carries and sums using OR and XNOR gates for every bit position directly [denoted FA+ in Fig. 2(a)]. In the bits of  $D$  with binary weights  $2^1$  and  $2^0$ , the adders should be modified to accept the values  $s_n$  and  $c_{n+1}$ , respectively, as shown in Fig. 2(b).

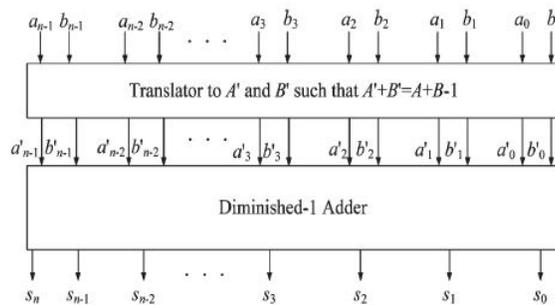


Fig. 3. Architecture proposed in [5].

### B. Vergos and Bakalis [5]

In [5], the authors subtract the sum of the two  $n$ -bit inputs  $A$  and  $B$  by 1 to produce the diminished-1 values  $A'$  and  $B'$ , and modulo  $2n$  sum of  $A'$  and  $B'$  can be performed by any diminished-1 architecture, as follows:

$$||A + B|_{2^{n+1}}|_{2^n} = |A' + B'|_{2^n} + \bar{c}_{out}. \tag{3}$$

The value  $c_{out}'$  is the inverted end-around carry produced by  $A' + B'$ , and the architecture of [5] is shown in Fig. 3.



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

The architecture proposed in [4] makes use of a constant-time operator, which is composed of a simplified carry-save adder stage, leading to efficient modulo  $2n + 1$  adders. The architecture proposed in [5] can be applied in the design of area-efficient residue generators and multioperand modulo adders. However, it should be noted that, in [4], the values that are subtracted by the inputs  $A$  and  $B$  are not constants.

In [5], the way to implement the translator for decreasing the sum of two inputs by 1 was not mentioned. Furthermore, in [5], the ranges of two inputs  $A$  and  $B$  are less than the one proposed in [4] (i.e.,  $\{0, 2^n - 1\}$  versus  $\{0, 2^n\}$ ). To resolve the above mentioned problems, we have proposed our design of area-efficient weighted modulo  $2^n + 1$  adder in the next section.

## IV. PROPOSED WEIGHED MODULO $2^N+1$ ADDER WITH CORRECTION SCHEME

Instead of subtracting the sum of  $A$  and  $B$  by  $D$ , which is not a constant as proposed in [4], we use the constant value  $-(2n + 1)$  to be added by the sum of  $A$  and  $B$ . In addition, we make the two inputs  $A$  and  $B$  to be in the range  $\{0, 2n\}$ , which is 1 more than  $\{0, 2n - 1\}$  as proposed in [5]. In the following, we present the designs of our proposed weighted modulo  $2n + 1$  adder.

Given two  $(n + 1)$ -bit inputs  $A = a_n a_{n-1}, \dots, a_0$  and  $B = b_n b_{n-1}, \dots, b_0$ , where  $0 \leq A, B \leq 2n$ . The weighted modulo  $2n + 1$  of  $A + B$  can be represented as follows:

$$|A + B|_{2n+1} = \begin{cases} A + B - (2^n + 1), & \text{if } (A + B) > 2^n \\ A + B, & \text{otherwise.} \end{cases} \quad (4)$$

Equation (4) can be stated as

$$\begin{aligned} & ||A + B|_{2n+1}|_{2^n} \\ &= \begin{cases} |A + B - (2^n + 1)|_{2^n}, & \text{if } (A + B) > 2^n \\ |A + B - (2^n + 1)|_{2^n} + |(2^n + 1)|_{2^n}, & \text{otherwise.} \end{cases} \end{aligned} \quad (5)$$

This can be expressed as

$$\begin{aligned} & ||A + B|_{2n+1}|_{2^n} \\ &= \begin{cases} |A + B - (2^n + 1)|_{2^n}, & \text{if } (A + B) > 2^n \\ |A + B - (2^n + 1)|_{2^n} + 1, & \text{otherwise.} \end{cases} \end{aligned} \quad (6)$$

From (6), it can easily be seen that the value of the weighted modulo  $2n + 1$  addition can be obtained by first subtracting the value of the sum of  $A$  and  $B$  by  $(2n + 1)$  (i.e., 0111,  $\dots$ , 1) and then using the diminished-1 adder to get the final modulo sum by making the inverted end-around carry as the carry-in.

Now, we present the method of weighted modulo  $2n + 1$  addition of  $A$  and  $B$  as follows.

Denoting  $Y'$  and  $U'$  as the carry and sum vectors of the summation of  $A, B$  and  $-(2n + 1)$ , where  $Y'_n = y_n - 2y_n - 3, \dots, y_0 y_{n-1}$  and  $U' = u_{n-1} u_{n-2}, \dots, u_0$ , the modulo addition can be expressed as follows:



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

$$\begin{aligned}
 &|A + B - (2^n + 1)|_{2^n} \\
 &= \left| \sum_{i=0}^{n-2} (2^i \times (a_i + b_i)) + 2^{n-1} \right. \\
 &\quad \left. \times (2a_n + 2b_n + a_{n-1} + b_{n-1}) + 0 \underbrace{11 \dots 11}_{n \text{ bits}} \right|_{2^n} \\
 &= \left| \sum_{i=0}^{n-2} (2^i \times (a_i + b_i + 1)) + 2^{n-1} \right. \\
 &\quad \left. \times (2a_n + 2b_n + a_{n-1} + b_{n-1} + 1) \right|_{2^n} \\
 &= \left| \sum_{i=0}^{n-2} (2^i \times (2y'_i + u'_i)) + 2^{n-1} \right. \\
 &\quad \left. \times (2a_n + 2b_n + a_{n-1} + b_{n-1} + 1) \right|_{2^n} .
 \end{aligned}$$

For  $i = 0$  to  $n - 2$ , the values of  $y_i$  and  $u_i$  can be expressed as  $y_i = a_i \vee b_i$  and  $u_i = a_i \oplus b_i$ , respectively ( $\vee$  is denoted as logic OR operation). Since the bit widths of  $Y'$  and  $U'$  are only  $n$  bits, the values of  $y_{n-1}$  and  $u_{n-1}$  are required to be computed taking the values of  $a_n, b_n, a_{n-1}$ , and  $b_{n-1}$  into consideration (i.e.,  $y_{n-1}$  and  $u_{n-1}$  are the values of the carry and the sum produced by  $2a_n + 2b_n + a_{n-1} + b_{n-1} + 1$ , respectively). It should be noted that  $0 \leq A, B \leq 2n$ , which means  $a_n = a_{n-1} = 1$  or  $b_n = b_{n-1} = 1$  will cause the value of  $A$  or  $B$  to exceed the range of  $\{0, 2n\}$ . Thus, these input combinations (i.e.,  $a_n = a_{n-1} = 1$  or  $b_n = b_{n-1} = 1$ ) are not allowed and can be viewed as don't care conditions, which can help us simplify the circuits for generating  $y_{n-1}$  and  $u_{n-1}$ . That is, the maximum value of  $2a_n + 2b_n + a_{n-1} + b_{n-1} + 1$  is 5, which occurs at  $a_n = b_n = 1$  (i.e., the maximum value of  $y_{n-1}$  is 2). The truth table for generating  $y_{n-1}, u_{n-1}$  and  $FIX$  is given in Table I, where  $\times$  is denoted as don't care.

The reason for  $FIX$  is that, under some conditions,  $y_{n-1} = 2$  (e.g.,  $a_n = b_n = 1$  and  $a_{n-1} = b_{n-1} = 0$ ), which cannot be represented by 1-bit line (marked as “\*” in Table I); therefore, the value of  $y_{n-1}$  is set to 1, and the remaining value of carry (i.e., 1) is set to  $FIX$ . Notice that  $FIX$  is wired-OR with the carry-out of  $Y'+U'$  (i.e.,  $c_{out}$ ) to be the inverted end-around carry (denoted by  $c_{out} \vee FIX$ ) as the carry-in for the diminished-1 addition stage later on. When  $y_{n-1} = 2$ ,  $FIX = 1$ ; otherwise,  $FIX = 0$ .

TABLE I  
TRUTH TABLE FOR GENERATING  $y'_{n-1}, u'_{n-1}$  AND  $FIX$   
(\*: CONDITIONS WHEN  $y'_{n-1} = 2$ )

$a_n$	$b_n$	$a_{n-1}$	$b_{n-1}$	$u'_{n-1}$	$y'_{n-1}$	$FIX$
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	1	1	0
0	1	0	0	1	1	0
0	1	0	1	$\times$	$\times$	$\times$
0	1	1	0	0	1*	1
0	1	1	1	$\times$	$\times$	$\times$
1	0	0	0	1	1	0
1	0	0	1	0	1*	1
1	0	1	0	$\times$	$\times$	$\times$
1	0	1	1	$\times$	$\times$	$\times$
1	1	0	0	1	1*	1
1	1	0	1	$\times$	$\times$	$\times$
1	1	1	0	$\times$	$\times$	$\times$
1	1	1	1	$\times$	$\times$	$\times$

According to Table I, we can have  $y_{n-1} = (a_n \vee b_n \vee a_{n-1} \vee b_{n-1})$ ,  $u_{n-1} = a_{n-1} \oplus b_{n-1}$ , and  $FIX = a_n b_n \vee b_n a_{n-1} \vee a_n b_{n-1}$ , respectively.

Based on the above mentioned equations, our proposed weighted modulo  $2n + 1$  addition of  $A$  and  $B$  is equivalent to

$$\begin{aligned}
 ||A + B|_{2^{n+1}}|_{2^n} &= |A + B - (2^n + 1)|_{2^n} \\
 &= |Y' + U'|_{2^n} + \overline{c_{out} \vee FIX}. \quad (7)
 \end{aligned}$$



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

Two examples for our proposed addition methods are given as follows.

*Example 1:* Suppose  $n = 4$ ,  $A = 1610 = 100002$ , and  $B = 1510 = 011112$ , respectively.

Step 1)  $(A + B) - (2n + 1) \Rightarrow Y' = 11102, U' = 00002, FIX = 1.$

Step 2)  $Y_- + U_- = 11102, cout = 0, \Rightarrow Y' + U' +$

$(C_{out} \vee FIX)' = 11102 = /16 + 15/17 = 1410.$

*Example 2:* Suppose  $n = 4$ ,  $A = 1110 = 010112$ , and  $B = 510 = 001012$ , respectively.

Step 1)  $(A + B) - (2n + 1) \Rightarrow Y' = 11102, U_- = 00012, FIX = 0.$

Step 2)  $Y_- + U_- = 11112, cout = 0, \Rightarrow Y' + U' +$

$(C_{out} \vee FIX)' = 100002 = /11 + 5/17 = 1610.$

The architecture for our proposed adder is given in Fig. 4. From Fig. 4, the signal of  $FIX$  can be computed in parallel with the translation to  $Y'+U'$ , leading to efficient correction [6]. In addition, the hardware cost for our correction scheme and FAF are less than the one proposed in [4], due to the fact that there are two inconstant numbers that should be processed [i.e., FA1 and FA0, as shown in Fig. 2(b)] in the translation stage.

## V. SYNTHESIS AND RESULT ANALYSIS

We use Verilog structuring Hardware Description Language to design our proposed adders and the existing work in [4] and [5] to compare delay/area performance. Table 2. shows area utilization summary for our proposed adder.

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

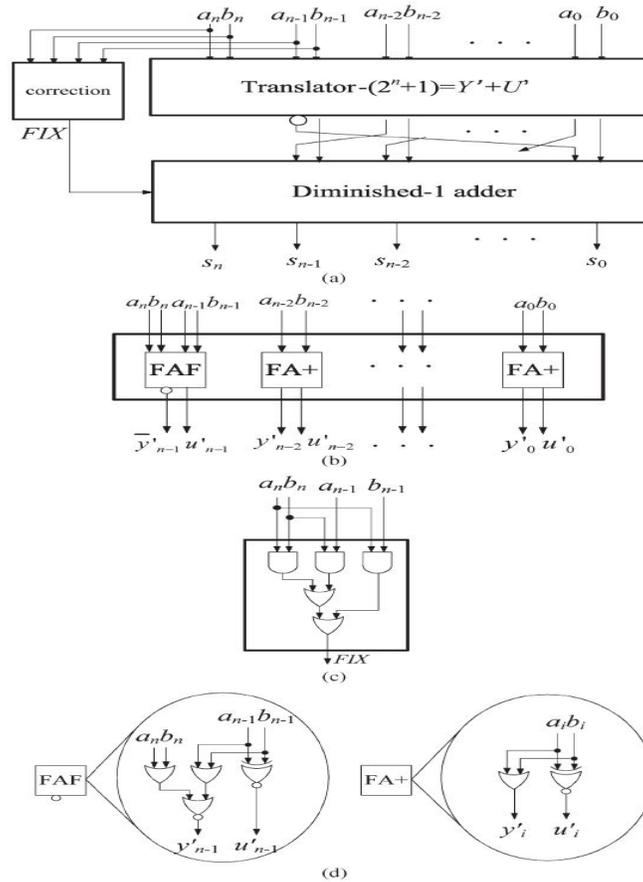


Fig. 4. (a) Architecture of our proposed weighted modulo  $2n + 1$  adder with the correction scheme (b) Architecture of the translator  $-(2n + 1)$  (c) Architecture of the correction scheme. (d) Individual Architecture of FAF and FA+

We adopt Sklansky-style [12], [19], [20] and Brent–Kungstyle [21] parallel-prefix structures for the diminished-1 adder implementations. The diminished-1 adder based on the Sklansky-style parallel-prefix structure with correction circuits for our proposed weighted modulo  $28 + 1$  adder is shown in Fig. 5(a). The square ( $\square$ ) and diamond ( $\diamond$ ) nodes denote the pre- and postprocessing stages of the operands, respectively. The black nodes ( $\bullet$ ) evaluate the prefix operator, and the white nodes ( $\circ$ ) pass the unchanged signals to the next prefix level.

Table 2.

Area Utilization Summary

	Area			
	Slices (out of 4656)	4 input LUTs (out of 9312)	IOs	Bonded IOBs (out of 190)
<b>Area</b>	27	48	43	43
<b>Utilization</b>	0%	0%	-	22%

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013

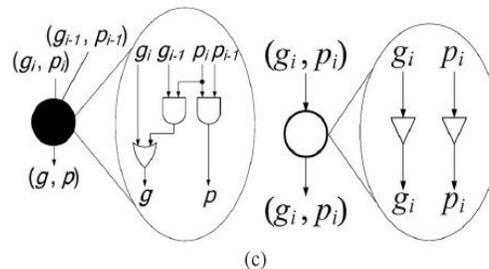
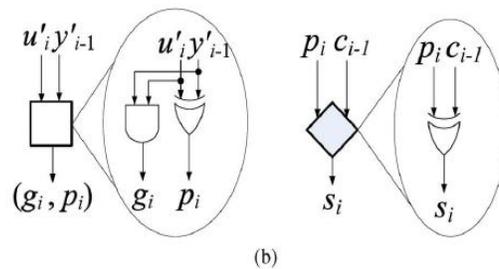
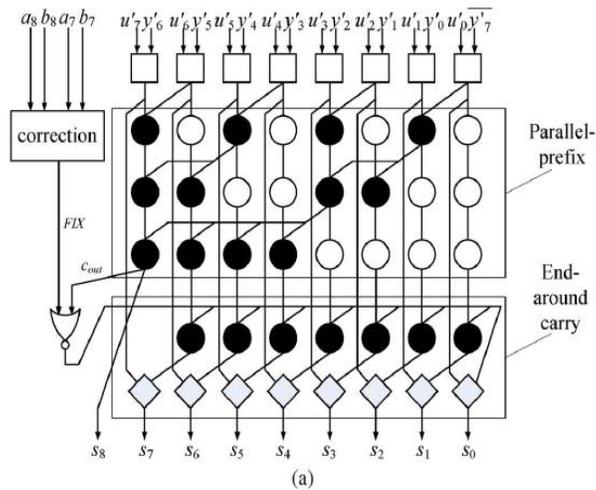


Fig. 5. (a) Diminished-1 adder based on the Sklansky-style parallel-prefix structure with the correction circuits for our proposed weighted modulo  $28 + 1$  adder (b) (□) Square and (♦) diamond nodes (c) (●) Black and (○) white nodes

The four nodes with detailed implementations are given in Fig. 5(b). It should be noted that the value of  $s_n$  (i.e.,  $c_{out}$ ) can be computed by wiring-AND all propagate signals (i.e.,  $s_n = \prod_{i=0}^{n-1} p_i$ )



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2013



Fig. 5. Simulated Output

## VI. CONCLUSION AND FUTURE PROSPECTS

Thus, an improved area-efficient weighted modulo  $2n + 1$  adder has been proposed. This has been achieved by modifying the existing diminished-1 modulo adders to incorporate simple correction schemes. The proposed adders can perform weighted modulo  $2n + 1$  addition and produce sums that are within the range  $\{0, 2n\}$ . Synthesis results show that our proposed adders can outperform previously reported weighted modulo adder in terms of area under the same delay constraints. The future prospects of our proposed work is to improve the area delay constraints of the proposed design.

## ACKNOWLEDGMENT

The authors gracefully acknowledge the professional help and extended support given by our project co-ordinator Mrs.Umma Habiba.

## REFERENCES

- [1] Lie-Liang Yang, Lajos Hanzo, "A Residue Number System Based Parallel Communication Scheme Using Orthogonal Signaling: Part II— Multipath Fading Channels" UK: IEEE Press, 2002.
- [2] Ghassem Jaberipur, Behrooz Parhami, " Unified Approach to the Design of Modulo- $(2n \pm 1)$  Adders Based on Signed-LSB Representation of Residues" Electron. Lett., vol. 38, no. 6, pp. 266– 268, Mar. 2009.
- [3] H. T. Vergos, C. Efstathiou, and D. Nikolos, "Diminished-one modulo  $2n + 1$  adder design," IEEE Trans. Comput., vol. 51, no. 12, pp. 1389– 1399, Dec. 2002.
- [4] H. T. Vergos and C. Efstathiou, "A unifying approach for weighted and diminished-1 modulo  $2n + 1$  addition," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 10, pp. 1041–1045, Oct. 2008.
- [5] H. T. Vergos and D. Bakalis, "On the use of diminished-1 adders for weighted modulo  $2n + 1$  arithmetic components," in Proc. 11<sup>th</sup> EUROMICRO Conf. Digit. Syst. Des. Archit., Methods Tools, Sep. 2008, pp. 752–759.
- [6] S.-H. Lin and M.-H. Sheu, "VLSI design of iminished-one modulo  $2n + 1$  adder using circular carry selection," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 9, pp. 897–901, Sep. 2008.
- [7] T.-B. Juang, M.-Y. Tsai, and C.-C. Chiu, "Corrections on 'VLSI design of diminished-one modulo  $2n + 1$  adder using circular carry selection'," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 3, pp. 260–261, Mar. 2009.