VOL. 9, NO. 12, DECEMBER 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

DESIGN AND IMPLEMENTATION OF IMPROVED AREA EFFICIENT WEIGHTED MODULO 2N+1 ADDER DESIGN

Dhanabal R.[1], Roshni Gunerkar[1] and Bharathi V.[2]
[1]School of Electronics Engineering, VIT University, Vellore, Tamil Nadu, India
[2]GGR College of Engineering and Technology, Anna University, Vellore, Tamil Nadu, India
E-Mail: rdhanabal@vit.ac.in

ABSTRACT

In this, we proposed improved area - efficient weighted modulo 2^n+1 adder. This is achieved by modifying existing diminished-1, weighted modulo $2^n + 1$ adder to incorporate simple correction schemes .Proposed adder is designed using area efficient parallel prefix structure and carry select adder. Proposed adder can produce modulo sums within the range $(0, 2^n)$ that is more than the range (0, 2n-1) produced in existing diminished-1 modulo 2^n+1 adders. Modular adder is designed using verilog HDL and implemented using 45 nm technology and the area required by the proposed adder is lesser than the existing diminished-1,weighted modulo 2^n+1 adder.

Keywords: modulo 2^n+1 adder, residue number system, VLSI design.

INTRODUCTION

For efficient carry free arithmetic computation, Residue number system has been employed in DSP applications as the computation for each residue channel can be independently done without carry propagation. As compared to binary based system, RNS based system can achieve higher speed-up. Thats why RNS based system are widely used in DSP application, FIR filter and communication components.

Arithmetic modulo 2^n+1 adder is the common RNS operation in pseudorandom number generation and cryptography. The moduli 2^n+1 is the commonly used moduli. There are many existing methods to speed up the weighted modulo 2^n+1 addition. Depending on input/output data representation this methods are classified into two categories namely dimished-1 and weighted.

In Diminished-1, each input and output operand is decreased by 1 as compared to weighted one. Therefore, in diminished-1 modulo adder only n-bit operands are needed that leads to faster and smaller component. However this incurs an overhead due to translator from/to binary weighted system. On the other hand in weighed modulo adder, (n+1) bit operands are required that avoids the use of translator but leads to more area as compared to diminished-1 modulo adder.

The general operations for modulo 2^n+1 addition are discussed in [13]. The authors proposed efficient parallel prefix adders for diminished-1 modulo $2^n +1$ addition in [3] and [4]. For improvement of area-time and time-power products, circular carry selection scheme was used for final modulo addition [3, 4]. However the hardware for decreasing or increasing the inputs/outputs by 1 is omitted in the literature. There is the need of zero detection circuit also as the value zero is not allowed in the diminished-1 modulo 2^n+1 addition. This leads to the increase in the hardware cost.

In [1], a unified approach for weighted and diminished-1 modulo 2^n+1 addition is proposed. This approach is based on making the modulo 2^n+1 addition of two (n+1)-bit input numbers A and B congruent to Y+U+1, where Y and U are two n-bit numbers. Thus to perform weighted modulo 2^n+1 addition of Y and U any diminished -1 modulo adder can be used. In [2], the translators are used to decrease the sum of two n-bit inputs A and B by 1and then weighted modulo 2^n+1 addition is performed using diminished-1 adders.

In this brief ,Improved area efficient weighted modulo 2^n+1 adder design using diminished-1 adder with simple correction schemes is proposed and is modified for more area efficient using Sklansky carry select adder (SK CSLA).

The rest of the brief is organised as follows.In section II Background of adders is given. In section III, review of design of two previous weighted modulo 2^n+1 adder is presented. Proposed area efficient weighted modulo 2^n+1 adder is presented in section IV. In section V synthesis result and comparisons are given and section VI concludes this brief.

Review of two weighted modulo 2^n+1 adder

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

$$S^*=|S-1|_{2^n+1}=|A+B-1|_{2^n+1}=|A^*+B^*|_{2^n}+ (\sim cout) \quad (1)$$

Where $|X|_z$ is defined as modulo Z of X and (~ cout) is defined as the inverted end around carry of diminished-1 modulo 2^n+1 adder.

Vergos and efstathiou [1]

In [1], first computation of the congruent modulo sum of A+B is done to produce Y and U and then final modulo sum is computed by any diminished-1 modulo adder as follows. Suppose A and B are two (n+1)-bit input numbers. i.e.

$A = a_n\, a_{n-1}, \ldots, a_0 = a_n \times 2^n + A_n$ and $B = b_n\, b_{n-1}, \ldots, b_0 = b_n \times 2^n + B_n$, where $0 \le A,B \le 2^n$, and A_n and B_n are two *n*-bit numbers; then

$|A + B|_{2^n+1} = ||A_n + B_n + D + 1|_{2^n+1} + 1|_{2^n+1}$
$=|Y + U + 1|_{2^n+1}$ (2)

In (2), $D = 2^n - 4 + 2(\sim c_{n+1}) + (\sim s_n)$, which is equal to $1111\ldots\sim c_{n+1} \sim s_n$,

where
$c_{n+1} = a_n \bullet b_n$ and $s_n = a_n \wedge b_n$ is the bit of *D* with binary weights 2^1 and 2^0, respectively. The first step of (2) calculates modulo 2n+1 carry save addition, giving the sum vector U and carry vector where $Y = y_{n-2}\ y_{n-3}\ldots y_0$ $(\sim y_{n-1})$ and $U = u_{n-1}\ u_{n-2}\ldots u_0$. Y and U are produced by adding A_n, B_n and D respectively. The values of *D* with binary weights of 2^2 through 2^n-1 are all 1; therefore, adders to produce the carries and sums are designed using OR and XNOR gates for every bit position directly. In the bits of *D* with binary weights 2^1 and 2^0, the adders should be modified to accept the values $(\sim s_n)$ and $(\sim c_{n+1})$, respectively.

Vergos and bakalis [2]

In [2] the sum of the two *n*-bit inputs *A* and *B* is subtracted by 1 to produce the diminished-1 values A' and B' and modulo 2^n sum of A and B can be computed by any diminished-1 architecture, as follows:

$||A + B|_{2^n+1}|_{2^n} = |A' + B'|_{2^n} + \sim cout.$ (3)

The value (~*c*out) is the inverted end-around carry produced by A'+ B'. The architecture proposed in [1] makes use of a constant time operator, which is formed by the use of simplified carry-save adder stage, leading to efficient modulo $2^n + 1$ adders. The architecture proposed in [2] can be applied in the design of area-efficient residue generators and multioperand modulo adders.

However in [1], the values that are subtracted by the inputs *A* and *B* are not constants. In [2], the way to implement the translator for decreasing the sum of two inputs A and B by 1 was not mentioned. Further, in [2], the ranges of two inputs A and B are less than the one proposed in [1] (i.e., $\{0, 2^n - 1\}$ versus $\{0, 2^n\}$). To provide remedy on these problems, area-efficient weighted modulo 2^n+1 adder design using parallel prefix structure and sklansky carry select adder is proposed in the section 4.

Background

RNS and modular addition

RNS is defined as group of co-prime modular radixes $\{m_1, m2...m_N\}$ where N>1GCD $(m_i, m_j) = 1$ i≠j i, j=1, 2...N. The integer X in {0, M} can be represented with respect to the modulus m_i that is $\{x_1, x_2.....x_N\}$. Let $a_1\ a_2.....a_N$, $b_1\ b_2.....b_n$ and $c_1\ c_2.....c_n$ be the RNS representation of integers.According to Gaussian modular algorithms, if $c_i=|a_i \Delta b_i|_m$ we can get $C=|A \Delta B|_M$ where "Δ" represent subtraction, addition and multiplication. A, B and C are in the range [0, M]. For integers A and B in the range of [0, m] modulo m addition is defined as
$C=|A+B|_m = A+B$ A+B<m
A+B-m A+B>m

If $C=|A+B|_m$ and the bit width of the modular adder is n-bit, where $n= [\log_2 m]$. [1]

Parallel prefix adders

Parallel prefix adders are obtained from carry look ahead structure. Tree Structure form is used to increase the speed of arithmetic operation. Parallel prefix adders are fastest adders and are used for high performance.
Parallel prefix adder involves three stages:
a) Pre processing
b) Carry generation
c) Post processing

Pre- processing stage

In this, the signals are computed, generated and propagate to each pair of input U and Y. This signals are given by
$P_i=U_i \wedge Yi$
$G_i= U_i$ and Yi

Carry generation network

In this stage carries are computed corresponding to each bit. Execution of these operations is carried out in parallel. After the carries are computed in parallel they are segmented into smaller groups. Carry generate and propagate are used as intermediate signals which are given by

$P_{i:j}=P_{i:k+1}$ and $P_{k:j}$
$G_{i:j}=G_{i:k}+1$ or $(P_{ik+1}$ and $G_{k:j})$

Post processing

This is the final step to calculate summation of input bits.sum bits are calculated as:
$C_{i-1} = (P_i$ and $C_{in})$ or G_i
$S_i= P_i$ xor C_{i-1} [5]

Proposed improved area - efficient weighted modulo 2^n+1 adder

Instead of subtracting the sum of A and B by D, which is not a constant as proposed in [1], the constant value - $(2^n + 1)$ is used which is to be added by the sum of A and B. The two inputs A and B are in the range $\{0, 2^n\}$, which is 1 more than $\{0, 2^n - 1\}$ as proposed in [2]. In the following, the designs of proposed weighted modulo $2^n + 1$ adder is presented.

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

$|A + B|_{2^n+1} = A + B - (2^n + 1)$, if $(A + B) > 2^n$
A + B, otherwise. (4)

Equation (4) can be stated as
$|||A+B|_{2^n+1}|_{2^n} = |A+B-(2^n+1)|_{2^n}$, if $(A+B)>2^n$
$|A+B-(2^n+1)|_{2^n}+|(2^n+1)|_{2^n}$, otherwise (5)
This can be expressed as

$||A+B|_{2^n+1}|_{2^n} = |A+B - (2^n+1)|_{2^n}$ if $(A+B)>2^n$
$|A+B - (2^n+1)|_{2^n} + 1$, otherwise (6)

From (6), it can be noted that the value of weighted modulo 2^n+1 addition can be obtained by first subtracting the sum of A and B by 2^n+1 and then using diminished-1 adder to get the final modulo sum by making the inverted end around carry as carry-in.
The weighted modulo 2^n+1 addition of A and B can be calculated as follows:

Denoting U and Y as the sum and carry vector of the summation of A, B and - (2^n+1) where Y= y_{n-2} $y_{n-3....}$ y_0 $(\sim y_{n-1})$ and U=u_{n-1} $u_{n-2....}$ u_0, the modulo addition can be expressed as:

$$|A+B-(2^n+1)|_{2^n} = |\sum (2^i * (a_i + b_i)) + 2^{n-1} * (2a_n + 2b_n + 2b_n + a_{n-1} + b_{n-1}) + 011....11|_{2^n}$$
$$= |\sum (2^i * (a_i + b_i + 1)) + 2^{n-1} * (2a_n + 2b_n + a_{n-1} + b_{n-1} + 1)|_{2^n}$$
$$= |\sum (2^i * (2y+u)) + 2^{n-1} * (2a_n + 2b_n + a_{n-1} + b_{n-1} + 1)|_{2^n}$$

For i=0 to n-2, the values of U_i and Y_i can be expressed as $U_i = \sim (a_i \wedge b_i)$ and $Y_i = a_i | bi$ respectively. Since Y and U are n bit, the value of y_{n-1} and u_{n-1} are calculated using a_n, b_n, a_{n-1} and b_{n-1}. It should be noted that $0 \leq A, B \leq 2^n$, which means if $a_n=a_{n-1}=1$ or $b_n=b_{n-1}=1$, the value of A or B will exceed the range of $\{0, 2^n\}$. Thus this input combinations are not allowed and can be viewed as don't care condition.The truth table for generating y_{n-1}, u_{n-1} and FIX is given in Table-1.

The reason for FIX is that the value of y_{n-1} can be equal to 2(e. g $a_n= b_n =1$ and $a_{n-1}=b_{n-1}=0$) which cannot be represented by 1 bit line. Therefore value of y_{n-1} is set to 1 and remaining value of carry i.e. 1 is set to FIX. FIX is wired -or with the carry out of Y+U (i.e. cout) to be inverted end around carry as carry in for diminished -1 addition. When $y_{n-1}=2$, FIX=1 otherwise FIX=0. [11]

According to Table I $y_{n-1}= (a_n|b_n|a_{n-1}|b_{n-1})$, $u_{n-1} = \sim (a_{n-1} \wedge b_{n-1})$ and FIX = a_n $b_n|b_n a_{n-1}|a_n b_{n-1}$ respectively. Proposed weighted modulo 2^n+1 addition of A and B is equivalent to

$$||A+B|_{2^n+1}|_{2^n} = |A+B-(2^n+1)|_{2^n} = |Y+U|_{2^n} + \sim (cout \mid FIX). \quad (7)$$

Examples for proposed weighted modulo 2^n+1 addition are given as follows:

Example-1: suppose n=4, $A=16_{10}=10000_2$ and $B=15_{10}=01111_2$, respectively.
Step 1) $(A+B)-(2^n+1) \Rightarrow Y=1110_2$ $U=0000_2$, FIX=1.
Step 2) $Y+U=1110_2$, Cout = 0, $\Rightarrow Y+U+\sim(Cout|FIX) = 1110_2 = |16+15|_{17}=14_{10}$

The architecture for weighted modulo 2^n+1 adder based on sklansky parallel prefix structure is shown in Figure-1. The signal of FIX can be computed in parallel with the translation to Y+U, leading to efficient correction.

In Figure-2 Diminished-1 adder based on sklanky parallel prefix structure with correction circuits for weighted modulo 2^n+1 adder is shown. The weighted modulo 2^n+1 adder is enhanced for more area efficiency by designing the diminished -1 adder using sklansky carry select adder (SK CSLA)as shown in Figure-3. In 16- bit sk csla shown in Figures 3 2, 3, 4, 5 bit sklansky parallel prefix structure are used for carry in 0 and output of this is fed to add one circuit which is used to increment the output of 2, 3, 4, 5 bit sklansky structure by 1. The final carry out is computed by selecting the carry out of the 2, 3, 4, 5sklansky and add one circuits independently using Mux. [12].

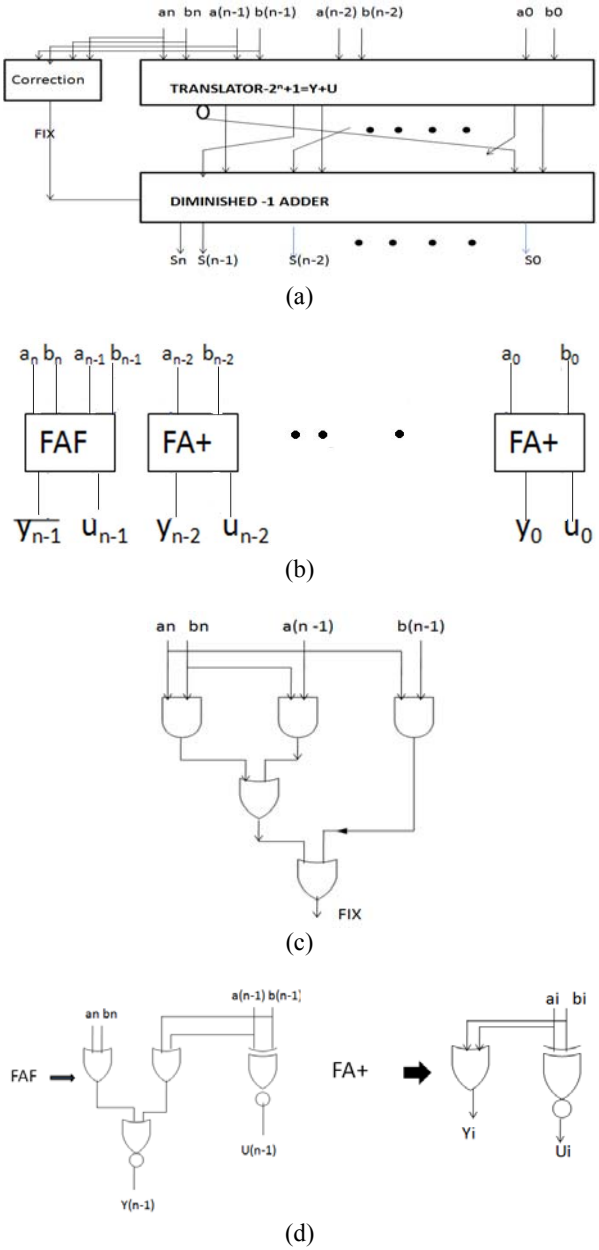

Figure-1. (a) Architecture of weighted modulo 2^n+1 adder with the correction scheme. (b) Architecture of the

translator-(2^n+1). (c)Architecture of the correction scheme. (d) Architecture of FAF and FA+, respectively. [11]

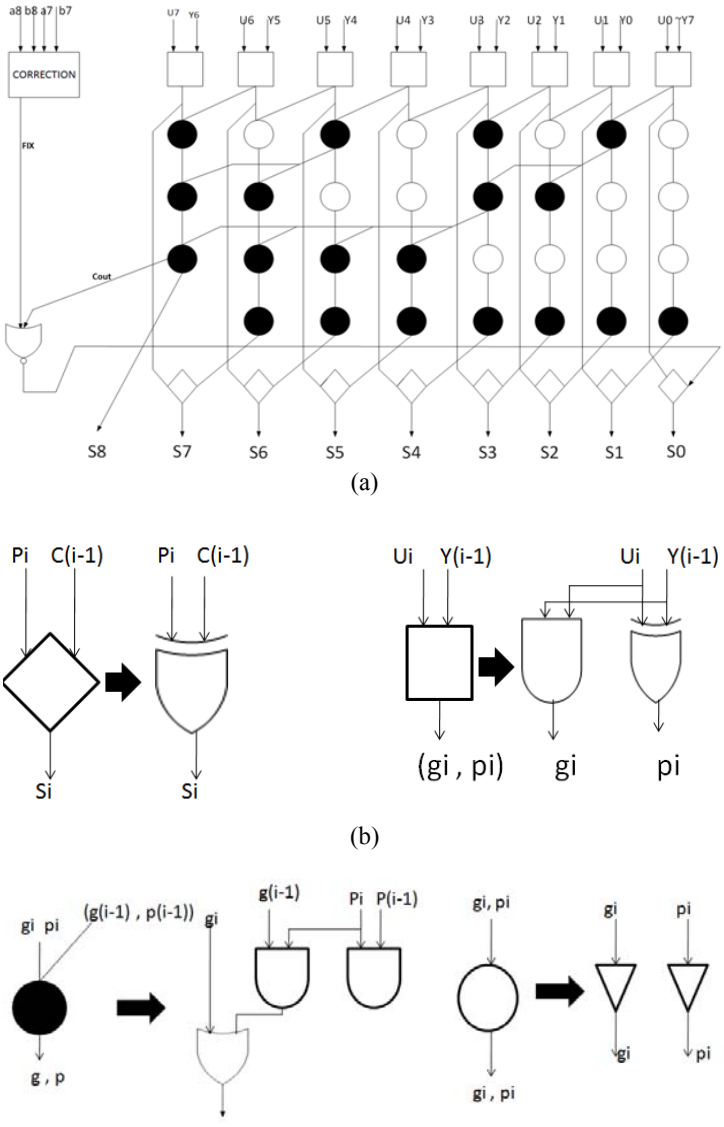

Figure-2. (a) Diminished-1 adder based on the sklansky-style parallel prefix structure for weighted modulo 2^n+1 adder (b) diamond nodes and square node respectively. (c) black operator and white node respectively. [11]

Simulation, synthesis results and comparison

Verilog structuring hardware description language is used to design proposed adder and the existing work and are implemented using 45 nm technology. The simulated output of 16 bit SK CSLA is shown in Figure-4. The values of the area and power for the diminished-1 adder designed using sklansky and sklansky carry select adder are shown in Table-2 and Table-3 respectively. The comparison for the various modulo 2^n+1 adder is shown in Figure-5. The diminished adder based on Sklansky style parallel prefix structure and SK CSLA with correction circuits for weighted modulo 2^n+1 adder are shown in Figure-2(a) and Figure-3 respectively. The square (□) and the diamond (◇) nodes denote the pre-processing and post-processing stages of the operands, respectively. The black nodes evaluate the prefix operator and the white nodes pass the unchanged signal to the next level. The detailed information of four nodes is given in Figure-2. (b), (c).

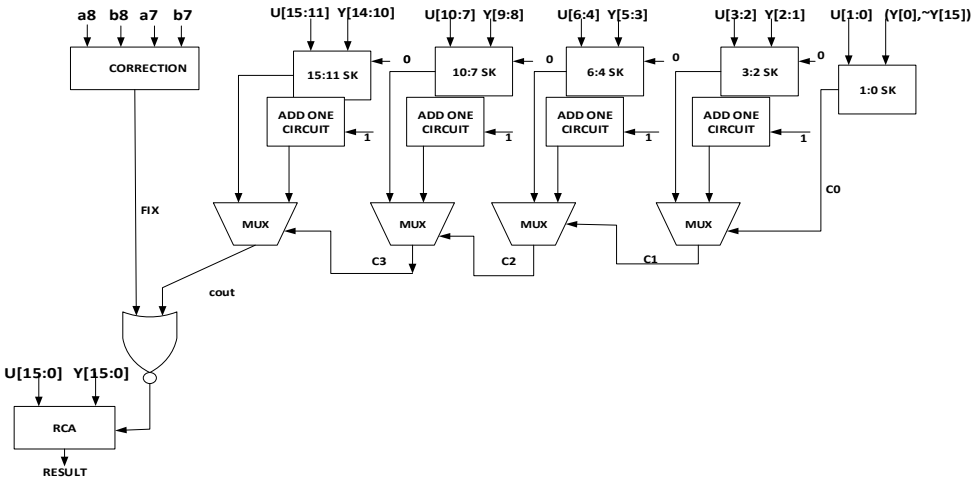

Figure-3. 16-bit sklansky carry select adder for proposed weighted modulo 2^n+1 adder where RESULT represents the final modulo addition.

Table-1. Truth table for generating y_{n-1}, u_{n-1} and FIX (*conditions when y_{n-1}=2).

An	bn	an-1	bn-1	un-1	yn-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	X	X	X
0	1	1	0	0	1*	1
0	1	1	1	X	X	X
1	0	0	0	1	1	0
1	0	0	1	0	1*	1
1	0	1	0	X	X	X
1	0	1	1	X	X	X
1	1	0	0	1	1*	1
1	1	0	1	X	X	X
1	1	1	0	X	X	X
1	1	1	1	X	X	X

Table-2. Area values for various modulo 2^n+1 adder (UNITS: um.sq).

Bit	Sklansky parallel prefix structure based modulo 2^n+1 adder	Sklansky carry select adder based modulo 2^n+1 adder
8	535.19	521.85
16	1002.32	707
32	2001.97	1957

Table-3. Power values for various modulo 2^n+1 adder (UNITS: nW).

Bit	Sklansky parallel prefix structure based modulo 2^n+1 adder	Sklansky carry select adder based modulo 2^n+1 adder
8	242126.280	228460.334
16	421658.326	189044.352
32	857017.366	596165.270

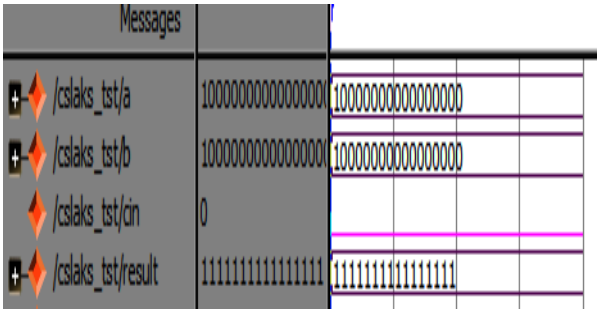

Figure-4. Simulated output for 16 bit SK CSLA.

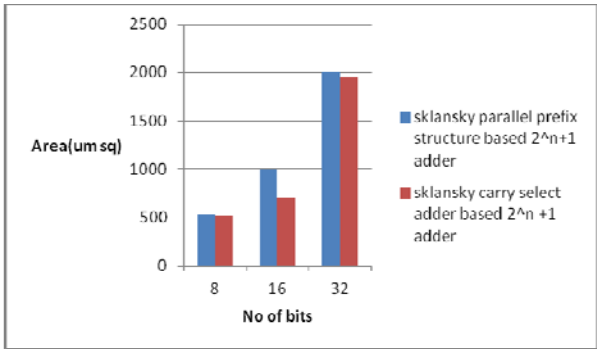

Figure-5. Area comparison of modular adder based on different adder design.

CONCLUSIONS

In this brief an improved area efficient weighted modulo 2^n+1 adder is proposed. This is achieved by modifying existing weighted modulo 2^n+1 adder by designing the diminished-1 adder used with more area efficient sklansky carry select adder. The proposed adders can perform weighted modulo 2^n+1 addition and produce sums within the range $\{0, 2^n\}$. Synthesis result shows that the weighted modulo 2^n+1 adder designed using SK CSLA can outperform weighted modulo 2^n+1 adder designed using sklansky adder in terms of area.

ACKNOWLEDGEMENT

Roshni Gunerkar would like to thank Mr. R. Dhanabal, Assistant Professor (Senior Grade), SENSE who had been guiding me throughout the project and helped me in technical issues about the paper.

REFERENCES

[1] H.T Vergos. and C.Efstathiou. 2008. A unifying approach for weighted and diminished-1 modulo 2^n+1 addition. IEEE Trans. Circuit's syst. II, exp briefs. 55(10): 1041-1045.

[2] H.T. Vergos and D. Bakalis. 2008. On the use of diminished -1 adders for weighted modulo 2^n+1 arithmetic components. In: proc.11th EUROMICRO Conf. Digit. Syst. Des. Archit, Methods tools. pp. 752-759.

[3] S-H. Lin and M.-H. Sheu. 2008. VLSI design of diminished-one modulo 2^n+1 adder Using circular carry selection,"IEEE trans. circuits syst II expbriefs. 55: 897-901.

[4] T.-B juang M.-Y. Tsai and C.-C. Chiu. 2009. Corrections on VLSI design of diminished-one modulo 2^n+1 adder using circular carry selection. IEEE trans. circuits syst II exp briefs. 56(3): 260-261.

[5] M. snir. 1986. Depth-size trade - offs for parallel prefix computation. In: Journal of algorithms. 7, pp. 185-201.

[6] H.T Vergos, C. Efstathiou and D. Nikolos. 2002. Diminished-one modulo 2^n+1 adder design. IEEE trans. Comput. 51(12): 1389-1399.

[7] C. Efstathiou, H.T. vergos and D. Nikolos. 2003. Modulo 2^n+1 adder design using select prefix blocks. IEEE Trans. Comput. 52(11): 1399-1406.

[8] A. A. Hiasat. 2002. High-speed and reduced-area modular adder structures for RNS. IEEE Trans. Comput. 51(1): 84-89.

[9] G. Jaberipur and S. Nejati. 2011. Balanced minimal latency RNS addition for moduli set, (2^n+1, 2^n-1) in Proc. 18th Int. Conf. Systems, Signals and Image Processing (IWSSIP). pp. 1-7.

[10] 2008. A unifying approach for weighted and diminished-1 modulo 2^n+1 addition. IEEE Trans. Circuits Syst. II, Exp. Briefs. 55(10): 1041-1045.

[11] 2010. Improved area efficient weighted modulo 2^n+1 adder design with simple correction schemes. IEEE transactions on circuits and systems-II: express briefs. 57(3).

[12] B. Ramkumar, Harish M kittur. 2012. Low-power and area-efficient carry select adder. IEEE transaction on very large scale integration (VLSI) system. 20(2): 371-375.

[13] R. Zimmermann. 1999. Efficient VLSI implementation of modulo 2^n+1 addition and multiplication. In: proc 14^{th} IEEE symp. comput. Arithmetic, apr. pp. 158-167.

[14] Ushasree G, Dhanabal r, Sarat kumar sahoo. 2013. Implementation of a High Speed Single Precision Floating Point Unit using Verilog. International Journal of Computer Applications (0975 - 8887) National conference on VSLI and Embedded systems.

[15] Dhanabal R, Bharathi V, Athmakuri Vivek. 2014. Design and Implementation of Low Power Floating Point Arithmetic Unit. International Journal of Applied Engineering Research. ISSN 0973-4562. 9(3): 339-346.

[16] Ushasree G, Dhanabal r, Sarat kumar sahoo. 2013. VLSI Implementation of a High Speed Single Precision Floating Point Unit Using Verilog. Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT). pp. 803-808.