DESIGN OF MODULO 2^N+1 FIR FILTER ARCHITECTURE

Mrs.M. Thiruveni*

D.Shanthi**

Abstract*—Filtering is one of the main operations in signal processing. The efficiency of the filter mainly depends on multiplier and adder. The modulo 2^n + 1 multiplier and adder are used to design modulo 2^n + 1 FIR filter architecture which is useful in applications like Residue Number System, Digital Signal Processing applications and cryptographic algorithms. In this multiplier [1], one operand uses weighted representation and another operand use diminished-1 representation. The new multiplier reduces the number of partial products which in turn reduces the operational time and power. Modulo 2^n + 1 adder [2] can produce modulo sums within the range {0, 2^n}, which is more than the range {0, 2^n − 1} produced by existing diminished-1 modulo 2^n + 1 adders. Since both units are designed effectively, the proposed FIR filter will be efficient.*

***Keywords*—** *FIR filter, dimnished-1 representation, residue number system, modular arithmetic, modular multiplier.*

* Associate Professor, Department of ECE, PSNA college of Engineering and Technology, Dindigul, Tamilnadu, India

** Professor, Department of CSE, PSNA college of Engineering and Technology, Dindigul, Tamilnadu, India

I. INTRODUCTION

The RESIDUE number system (RNS) has been employed for efficient parallel carry-free arithmetic computations suitable for high speed DSP applications due to their inherited parallelism, modularity, fault tolerance and localized carry propagation properties .Since modulo computations can achieve significant speedup over the binary-system-based computation, they are widely used in DSP processors, FIR filters, and communication components. Some arithmetic operations, such as addition and multiplication, can be carried out more efficiently in RNS than in conventional two's complement systems. The modulo $2^n + 1$ addition is the most crucial step among the commonly used moduli sets.

There are many previously reported methods to speed up the modulo $2^n + 1$ addition. Depending on the input/output data representations, these methods can be classified into two categories, namely, diminished-1 and weighted respectively.

In the diminished-1 representation, each input and output operand is decreased by 1 compared with its weighted representation. Therefore, only *n*-bit operands are needed in diminished-1 modulo $2^n + 1$ addition, leading to smaller and faster components.

However, this incurs an overhead due to the translators from/to the binary weighted system. On the other hand, the weighted-1 representation uses $(n + 1)$-bit operands for computations, avoiding the overhead of translators, but requires larger area compared with the diminished-1 representations.

Modulo multipliers can be divided into three categories, depending on the type of operands that they accept and output:

1) the result and both inputs use weighted representation;

2) the result and both inputs use diminished-1 representation;

3) the result and one input use weighted representation,while the other input uses diminished-1.

For the first category, Zimmermann [10] used Booth encoding to realize, but depart from the diminished-1 arithmetic, which leads to a complex architecture with large area and delay requirements. Sousa *et al* [11] modified the radix-4 Booth recoding in order to take advantage of the diminished-1 arithmetic. The number of the partial products is reduced to (n/2+1) but the area for the partial products generator and the correction term generator are large, and the constant "2" has to be added for the final modular adder. Vergos [13] proposed new modulo

multipliers using non-Booth recoding. The number of the partial products is n+1 and the word-length of each partial product is n-bit wide.

For the second category, Wang *et al.* [8] proposed diminished-1 multipliers with n-bit input operands. The multipliers use a non-Booth recoding and a zero partial-product counting circuit. Handling of zero inputs and results was not considered. Sousa *et al* [11] proposed modulo multipliers for diminished-1 representation with treatment of zero operands. The multipliers use a modified radix-4 Booth recoding and a Wallace tree addition [3]. The number of the partial products is approximately halved without counting in the correction term and the constant. The correction term generator is a complex combinational circuit. Furthermore, the modification of the radix-4 Booth recoding leads to complexities in the partial product generator. Efstathiou [12] designed a diminished-1 multiplier by using non-Booth recoding. Treatment of zero operands or results was not discussed. The number of the partial products is that leads to a large overhead for area and delay.

The third category [15] applies to some special applications, such as encryption algorithm and FIR filters. Due to one input using diminished-1 representation, the new architecture can be based on n-bit additions and radix-4 Booth recoding scheme, which is efficient and regular.

The coefficients of RNS FIR filters are constant, the diminished-1 representations of the coefficients can be pre-computed during design process, and its conversion does not belong to the critical path.

Improved weighted modulo $2^n + 1$ adder design using diminished-1 adders with simple correction schemes is achieved by subtracting the sum of two $(n + 1)$-bit input numbers by the constant $2^n + 1$ and producing carry and sum vectors. The modulo $2^n + 1$ addition can then be performed using parallel-prefix structure diminished-1 adders by taking in the sum and carry vectors plus the inverted end-around carry with simple correction schemes. The modulo $2^n + 1$ adder used do not require the hardware for zero detection that is needed in diminished-1 modulo $2^n + 1$ addition.

II.FIR FILTER

One of the most widely used operations in DSP applications is FIR filter [4-6]. A variety of approaches to implement FIR filters have been pursued. Low-power architecture for linear phase

FIR including retimed structure, balanced modular architecture, separated signed processing data flow and modification of the CSD representations are there.

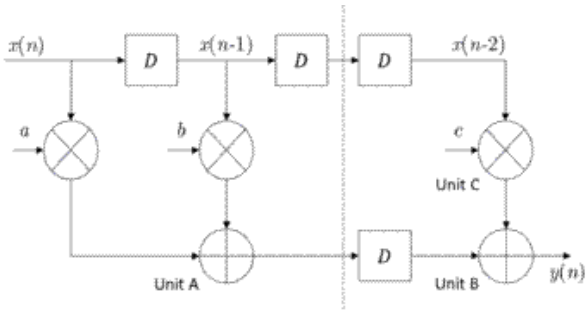


Fig.1. Linear phase FIR filter structure

The 8-tap modulo 2^n+1 FIR filter structure is constructed using modulo 2^n+1 multipliers [1] and adders [2].The proposed schemes not only address the linear-phase FIR filter, but also can improve the non linear-phase FIR filer.

III. MODULO $2^n + 1$ MULTIPLIER ARCHITECTURE

Multiplication can be done more efficiently in RNS [7-9] than in conventional two's complement systems. Efficient schemes for modulo multipliers have been studied intensively modulo $2^n + 1$ multiplier is proposed and it uses non booth recoding.

Modulo $2^n + 1$ *multiplier*

The modulo $2^n + 1$ arithmetic operations require (n+1)-bit operands. To avoid (n+1)-bit circuits, the diminished-1 number system [19] has been adopted. d [A] be the diminished-1 representation of the normal binary number, namely

A € $[0,2^n]$, that is

$$d[A] = |A - 1|_{2^n+1} \quad (1)$$

When A$\neq$ 0, d [A] € $[0,2^n-1]$, is an n-bit number, therefore (n+1)-bit circuits avoided in this case. In new modulo $2^n + 1$ multiplication, the result and one input use weighted representations, while the other input uses diminished-1 representation.

Let [A]= $[(a]_n a_{n-1} a_{n-2} \dots\dots\dots\dots . a_0)$

be the diminished-1 representation of the weighted binary number, and A,B=$[(b]_n b_{n-1} b_{n-2} \dots\dots\dots\dots . b_0)$ and the output

P=$|A \times B|_{2^n+1}$ =]= $[(P]_n P_{n-1} P_{n-2} \dots\dots\dots\dots P_0)$ all be weighted binary numbers. Although one operand using diminished-1 representation, the new modulo $2^n + 1$ multipliers avoid conversion circuits between weighted and diminished-1 representation for some special applications, such as encryption algorithm and FIR filters.

Modulo $2^n + 1$ multiplier architecture

In accordant with the radix-4 Booth recoding [14], the partial product generator (PPG) can be constructed with the Booth encoder (BE) and Booth selector (BS). For BE block and BS block many implementations [15-18] were published, but they can be reduced to two categories: one having 4-bit bus and the other having 3-bit bus [20]. The proposed multiplier [1] uses a 3-bit bus approach. The BE block examines successive overlapping triplets $b_{2i+1} b_{2i} b_{2i-1}$ and encodes for each as an element of the set {-2,-1, 0, 1, 2}. Each BE block produces 3 bits: 1x, 2x and Sign. The 3 bits along with the multiplicand are used to form partial products. The BS blocks produced the partial products. Each BS block takes as inputs two successive bits of d [A]. There are two types of BS blocks in the proposed multipliers, BS^+ andBS^-, since the inverted multi-bit left-circular shifts of d [A] and d [-A] are different.

For the i-th partial product, $0 \leq i \leq k$, there are 2i BS^- and (n-2i) BS^+ blocks Fig. 3 presents the BE block and its truth table. Fig. 4 presents the BE_i block used by the new multipliers. Fig. 5 presents BS^+ and blocks BS^- and their truth tables. The CTG produces C which has the form $[(0x]_{i+1} 0x_i \dots\dots\dots\dots x_0)$

TABLE-I

Input			Output			Code
b_{2i+1}	b_{2i}	b_{2i-1}	Sign	2x	1x	
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	0	1	1
0	1	1	0	1	0	2
1	0	0	1	1	0	-2

Truth table for BE block

The 2i-th bit x_i is 1 when the BE_i block encodes 0, otherwise x_i is 0, one XNOR gate accepting the 1x and 2x bits of the BE_i block can generate the 2i-th bit x_i .

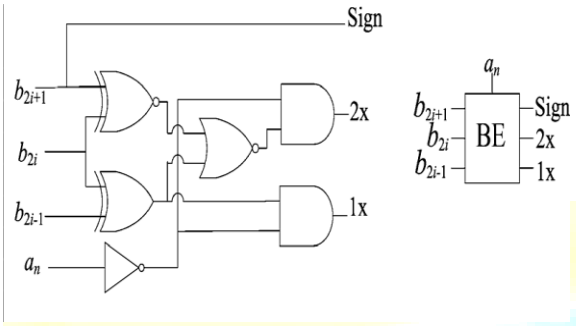
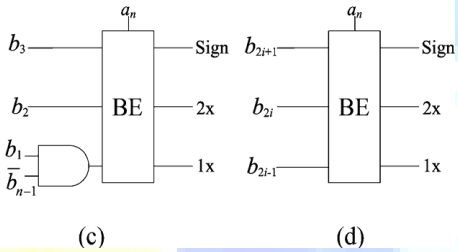


Fig.2. Block diagram of Booth encoder

(a) BE_0 for n Even (b) BE_0 for n odd



(c) (d)

(c) BE_1 for n even (d) BE_i 1<i<k for n even and 0<i<k for n odd

Fig.3.Logic diagram for BE blocks

The inverted EAC CSA tree reduces operands to two numbers. The CSA tree is usually constructed with full adders (FA). But in our multipliers, one CSA stage which takes the term can be further simplified,

Sign	2x	1x	p_i
0	0	0	1
0	0	1	a_i
0	1	0	a_{i-1}
1	1	0	$\bar{a}_{i-1}$
1	0	1	$\bar{a}_i$
1	0	0	1

Sign	2x	1x	p_i
0	0	0	0
0	0	1	$\bar{a}_i$
0	1	0	$\bar{a}_{i-1}$
1	1	0	a_{i-1}
1	0	1	a_i
1	0	0	0

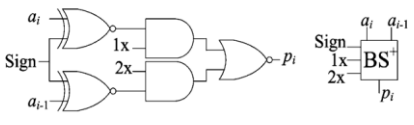


Fig.4. Truth table and BS^+ and BS^- blocks

since C has the form $[(\ldots.0x]_{i+1}0x_i \ldots\ldots\ldots\ldots x_0)$

with $x_i \in \{0\ 1\}$, every other full adder in this stage can be simplified as a half adder (HA). This CSA stage can be constructed by using n/2 HAs and n/2 FAs for even, and (n+1)/2 FAs and (n-1)/2 HAs for odd. The final adder is a diminished-1 modulo $2^n + 1$ adder. It is known that, the diminished-1 modulo $2^n + 1$ adder outperforms the normal binary modulo $2^n + 1$ adder in delay and area [21]. In this work, the fastest diminished-1 modulo $2^n + 1$ adder proposed in [22] is adopted.

IV. MODULO $2^n + 1$ ADDER

ARCHITECTURE

Instead of subtracting the sum of A and B by D, which is not a constant as proposed in [17], we use the constant value $-(2^n + 1)$ 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 $\{0, 2^n\}$, which is 1 more than $\{0, 2^n - 1\}$ as proposed in [18]. In the following, we present the designs of our proposed weighted modulo $2^n + 1$ adder.

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

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

This can be expressed as

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

It can easily be seen that the value of the weighted modulo $2^n + 1$ addition can be obtained by first subtracting the value of the sum of A and B by $2^n + 1$ (i.e., 0111, . . . , 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.

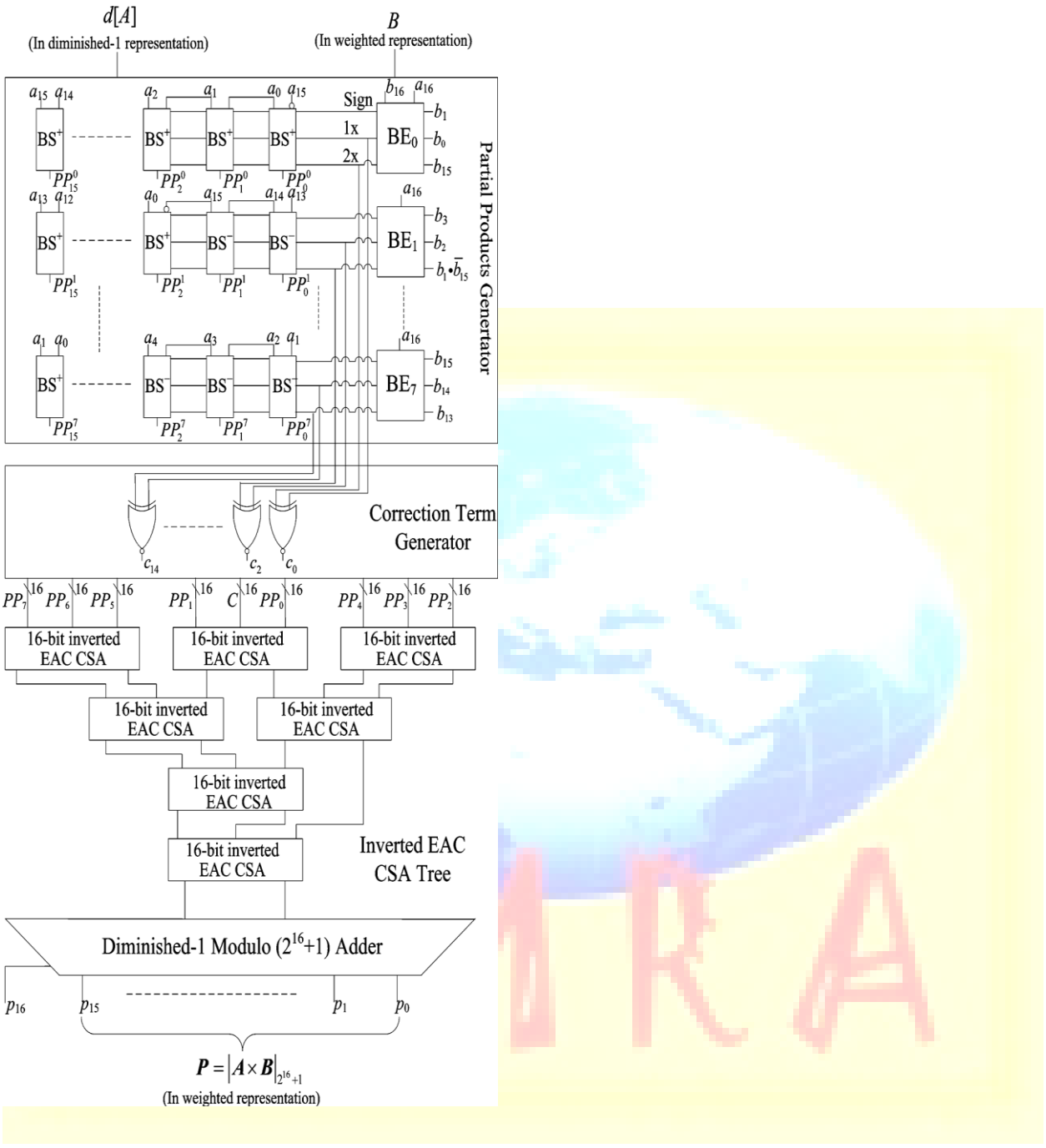


Fig.5. Architecture of proposed modulo $2^n + 1$ multiplier

The method of weighted modulo $2^n + 1$ addition of A and B as follows.

Denoting $Y\_$ and $U\_$ as the carry and sum vectors of the summation of A,B and $-(2^n + 1)$, where $Y' = y'_{n-2}\ y'_{n-3}, ..., y'_0\ y'_{n-1}$ and $U' = u'_{n-1}\ u'_{n-2}, ..., u'_0$ the modulo addition can be expressed as follows:

$$\left|A+B-(2^n+1)\right|_{2^n}$$

$$=|\sum_{i=0}^{n-2}(2^i*(a_i+b_i))+2^{n-1} \quad *(2a_n+2b_n+a_{n-1}+b_{n-1})+011...11|_2{}^n$$

$$=|\sum_{i=0}^{n-2}(2^i*(a_i+b_i))+2^{n-1}*(2a_n+2b_n+a_{n-1}+b_{n-1}+1)|_2{}^n$$

$$=|\sum_{i=0}^{n-2}(2^i*(2y'_i+u'_i))+2^{n-1} \quad *(2a_n+2b_n+a_{n-1}+b_{n-1}+1)|_2{}^n$$

For $i = 0$ to $n - 2$, the values of can be expressed as 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 (v *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. It should be noted that $0 \leq A, B \leq 2^n$, 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 $\{0, 2^n\}$. Thus, these input combinations 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' $_{n-1}$. The maximum value of $2a_n + 2b_n + a_{n-1} + b_{n-1} + 1$ is 5,which occurs at $a_n = b_n = 1$.The truth table for generating y'_{n-1} and u'_{n-1} is given in the table.

TABLE II

a_n	b_n	a_{n-1}	b_{n-1}	u'_{n-1}	y'_{n-1}
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	X	X
0	1	1	0	0	1
0	1	1	1	X	X

1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	X	X
1	0	1	1	X	X
1	1	0	0	1	1
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

Truth Table for generating y'_{n-1} , u'_{n-1}

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) $(A + B) - (2^n + 1) \Rightarrow Y' = 11102$,

$U' = 00002$, $FIX = 1$.

Step 2) $Y' + U' = 11102$, Cout = 0, $\Rightarrow Y' + U' + \text{Cout} \vee FIX = 11102 = \backslash 16 + 15|17 = 1410$.

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

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

Step 2) $Y' + U' = 11112$, Cout = 0, $\Rightarrow Y' + U' +$

cout $\vee FIX = 100002 = |11 + 5|17 = 1610$.

The architecture for our proposed adder is given in Fig. 6.

From Fig. 6, the signal of *FIX* can be computed in parallel

with the translation to $Y' + U'$ leading to efficient correction.

In addition, the hardware cost for our correction scheme and FAF are less than the one proposed in [21], due to the fact that there are two inconstant numbers that should be processed in the translation stage.

V RESULTS

The results of modulo $2^n + 1$ adder, modulo $2^n + 1$ multiplier and modulo FIR filter are obtained by simulation and synthesis using modelsim and Xilinx ISE.

Module	No of Slices	Delay (ns)
Modulo 2^{n+1} Adder	20	23.018
Modulo 2^{n+1} Multiplier	83	30.903
Modulo 2^{n+1} FIR Filter	629	143.294

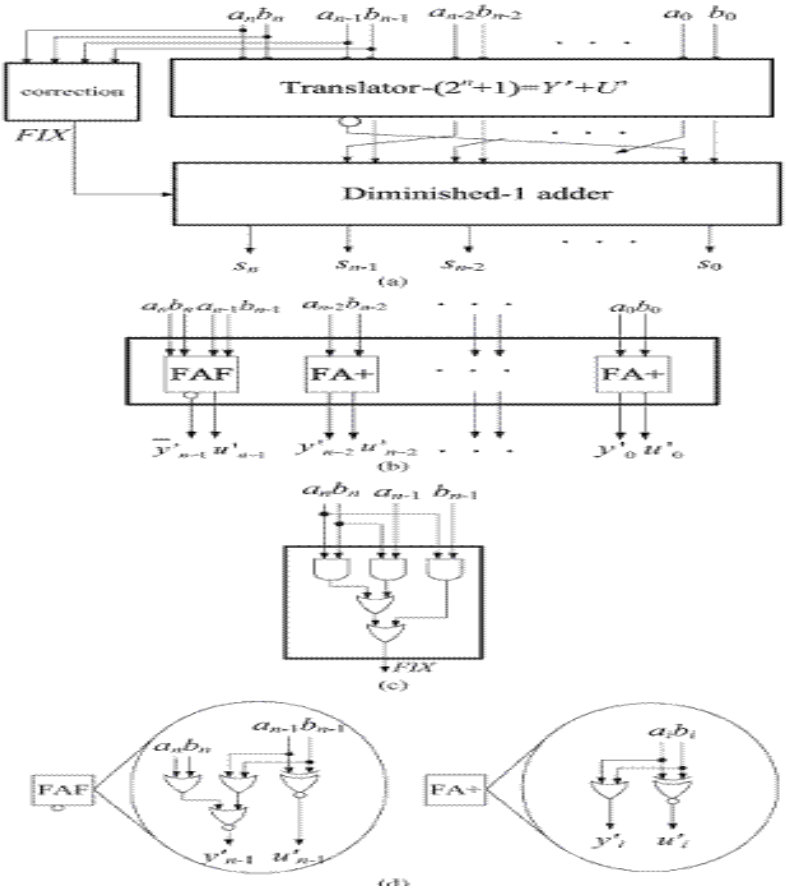


Fig. 6. (a) Architecture of our proposed weighted modulo 2*n* + 1 adder with the correction scheme.

(b) Architecture of the translator −(2*n* + 1).

(c) Architecture of the correction scheme.

(d) Architecture of FAF and FA+,respectively.

VI CONCLUSION

The efficiency of the filter mainly depends on multiplier and adder. In this paper, modulo $2^n + 1$ FIR filter architecture is built by using modulo $2^n + 1$ adder and modulo $2^n + 1$ multiplier. The multiplier uses weighted representation and diminished-1 representation. The multiplier reduces the number of partial products modulo $2^n + 1$ adder can produce modulo sums within the range $\{0, 2^n\}$, which is more than the range $\{0, 2^n - 1\}$ produced by existing diminished-1 modulo $2^n + 1$ adders.

It can be used in any Digital Signal Processing applications, Residue Number System, cryptographic algorithms,etc

REFERENCE

[1] Tso-Bing Juang, , Chin-Chieh Chiu, and Ming-Yu Tsai, "Improved Area-Efficient Weighted Modulo 2*n* + 1Adder Design With Simple Correction Schemes", IEEE transactions on CIRCUITS AND SYSTEMS—II: Express Briefs, Vol. 57, no. 3, march 2010.

[2] Jian Wen Chen, Ruo He Yao, and Wei Jing Wu," Efficient Modulo $2^n + 1$ Multipliers", IEEE transactions on very large scale integration (VLSI) systems, vol. 19, No. 12, December 2011

[3]N.Weste and K.Eshraghian, "Principles of CMOS VLSI design A System Perspective Reading", MA Addison-Wesley, 1988, ch.5.

[4]Vijaya Prakash.A.M,K.S.Gurumurthy,

"A Novel VLSI Architecture for low power FIR Filter",International Journal of Advanced Engineering and Application,, Jan uary'2011.

[5]S.Karunakaran ,N.Kasthuri,"high Performance VLSI Architecture for FIR Filter using on the FLY conversion Multiplier",European Journal of Scientific Research,Vol.67, No. 4, 2012.

[6]Evangelos F.Stefatos, Han Wei, Tughrul Arsla, Robert Thomson, "Low-Power reconfigurable VLSI Architecture for the implementation of FIR filters",proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium(IPDPS'05),2005.

[9]A. Wrzyszcz and D. Milford, "A new modulo multiplier," in *Proc. Int. Conf. Computer Design (ICCD'93)*, 1995, pp. 614–617.

[10] Z.Wang, G. A. Jullien, andW. C.Miller, "An efficient tree architecture for modulo multiplication," *J. VLSI Signal Process. Syst.*, vol. 14, no. 3, pp. 241–248, Dec. 1996.

[11] Y. Ma, "A simplified architecture for modulo multiplication,"*IEEE Trans. Comput.*, vol. 47, no. 3, pp. 333–337, Mar. 1998.

[12] R. Zimmermann, "Efficient VLSI implementation of modulo addition and multiplication," in *Proc. 14th IEEE Symp. Comput. Arithm.*, Adelaide, Australia, Apr. 1999, pp. 158–167.

[13] L. Sousa and R. Chaves, "A universal architecture for designing efficient modulo multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1166–1178, Jun. 2005.

[14] C. Efstathiou,H. T.Vergos,G. Dimitrakopoulos, and D.Nikolos, "Efficient diminished-1 modulo multipliers," *IEEE Trans. Comput.*, vol. 54, no. 4, pp. 491–496, Apr. 2005.

[15] H. T. Vergos and C. Efstathiou, "Design of efficient modulo multipliers," *IET Comput. Digit. Tech.*, vol. 1, no. 1, pp. 49–57, 2007.

[16] R. Chaves and L. Sousa, "Faster modulo multipliers without booth recoding," in *Proc. XX Conf. Design Circuits Integr.Systems (DCIS'05)*, 2005, pp. 400–404.

[17] A. Curiger, H. Bonnenberg, and H. Kaeslin, "Regular VLSI architectures for multiplication modulo," *IEEE J. Solid-State Circuits*, vol. 26, no. 7, pp. 990–994, Jul. 1991.

[18] W. C. Yeh and C. W. Jen, "High-speed booth encoded parallel multiplier design," *IEEE Trans. Comput.*, vol. 7, pp. 692–701, 2000.

[19] L. Leibowitz, "A simplified binary arithmetic for the fermat number transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, pp. 356–359, May 1976

[20] W. C. Yeh and C. W. Jen, "High-speed booth encoded parallel multiplier design," *IEEE Trans. Comput.*, vol. 7, pp. 692–701, 2000.

[21]H. T. Vergos and C. Efstathiou, "A unifying approach for weighted and diminished-1 modulo addition," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 10, pp. 1041–1045, Oct. 2008.

[22] H. T. Vergos, C. Efstathiou, and D. Nikolos, "Diminished-one modulo adder design," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1389–1399, Dec. 2002.

[23] A. Tyagi, "A reduced-area scheme for carry-select adders," *IEEE Trans. Comp.*, vol. 42, no. 10, pp. 1163–1170, Oct. 1993.

[24] M.A. Soderstand,W.K. Jenkins,G.A. Jullien,and F.J. Taylor Residue Number system arithmetic:Modern application in signal processing.