International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.3, Issue.3, May-June. 2013 pp-1372-1376 ISSN: 2249-6645

Area and Power Efficient Modulo 2^n+1 Multiplier

K. Pitambar Patra, [1] Saket Shrivastava, [2] Snehlata Sahu, [3] Sujit Kumar Patel[4]
[1234]*Department of Electronics & Communication Engineering Jaypee University of Engineering & Technology, Guna*

Abstract: *In this paper area-power efficient modulo 2^n+1 multiplier is proposed. The result and one operand for the new modulo multipliers use weighted representation, while the other uses the diminished- 1. By using the radix-4 Booth recoding, the new multipliers reduce the number of the partial products to n/2 for even and (n+1)/2 for odd except for one correction term. According to our algorithm, the resulting partial products are added through inverted end around carry save adder into two operands, which are finally adder by a 2-stage n-bit adder containing 2:1 multiplexer. By using the purposed adder, the new multipliers reduce the area and power. The analytical and experimental result indicates that the new modulo 2^n+1 multipliers, offer reduced power and more compact area among all the existing structures.*

Keywords: *2-Stage n-Bit Adder, Modulo Multiplier, Residue Number System (RNS).*

I. INTRODUCTION

Residue number systems (RNS) reduce the delay of carries propagation, thus suitable for the implementation of high-speed digital signal processing devices. Some arithmetic operations, such as addition and multiplication, can be carried out more efficiently in RNS than in conventional two's complement systems. RNS has been adopted in the design of Digital Signal Processors (DSP), Finite Impulse Response (FIR) filters], image processing units, Discrete Cosine Transform (DCT) processors, communication components, cryptography, and other DSP applications . In recent years, efficient schemes for modulo multipliers have been studied intensively. Generally, modulo 2^n+1 multipliers can be divided into three categories, depending on the type of operands that they accept and output:

i. the result and both inputs use weighted representation;
ii. the result and both inputs use diminished-1 representation;
iii. The result and one input use weighted representation, while the other input uses diminished-1.

For the first category, Zimmermann et al. [1] used Booth encoding to realize, but depart from the diminished-1 arithmetic, which leads to a complex architecture with large area and delay requirements. For the second category, Wang *et al.* [2] proposed diminished-1 multipliers with -bit input operands. The multipliers use a non-Booth recoding and a zero partial-product counting circuit. The main drawback in this architecture was handling of zero inputs and results were not considered.

Curiger et al. [3] proposed new modulo multipliers by using the third category. This architecture use ROM based look-up methods are competitive. The main drawback in this architecture increasing n-bit, they become infeasible due to excessive memory requirements.

Jian et al. [4] also proposed for the third category architecture and reduce the memory requirement and speed up. The new architecture is based on n-bit addition and radix-4 booth algorithm, which is efficient and regular. We are replaced diminished-1 modulo 2^n+1 adder by 2-stage n-bit adder.

The remainder of the paper is organized as follows: mathematical formulation of Diminished-1 number representation computation of modulo multiplier is presented in Section II. The proposed structures are presented in Section III. Hardware and time complexity of the proposed structures are discussed and compared with the existing structures in Section IV. Conclusion is presented in Section V.

II. DIMINISHED -1 NUMBER REPRESENTATION

The modulo 2^n+1 arithmetic operations require (n+1) bit operands. To avoid (n+1)-bit circuits, the diminished-1 number system [15] has been adopted. Let d[A] be the diminished-1 representation of the normal binary number $A \in [0,2^n]$, namely

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

In (i), when $A \neq 0, d[A] \in [0,2^n-1]$,is an n -bit number, therefore (n+1) -bit circuits can be avoided in this case. However,

$$A = 0, d[A] = d[0] = |-1|_{2^n+1} = 2^n \qquad \text{(ii)}$$

Is an (n+1) -bit number. This leads to special treatment for d [0]. The diminished-1 arithmetic operations [15] are defined as

$$d[-A] = \overline{d[A]}, if d[A] \in [0,2^n-1] \qquad \text{(iii)}$$

$$d[A+B] = \left| d[A] + d[B] + 1 \right|_{2^n+1} \quad \text{(iv)}$$

$$d[A-B] = \left| d[A] + \overline{d[B]} + 1 \right|_{2^n+1} \quad \text{(v)}$$

$$d[AB] = \left| d[A] \times d[B] + d[A] + d[B] \right|_{2^n+1}$$

$$= \left| d[A] \times B + B - 1 \right|_{2^n+1} \quad \text{(vi)}$$

$$d[2^k, A] = iCLS(d[A], k) \quad \text{(vii)}$$

$$d[-2^k, A] = iCLS(\overline{d[A]}, k) \quad \text{(viii)}$$

Where $\overline{d[A]}$ represents the one's complement of d[A]. In (vii) and (viii) iCLS (d[a], k) is the k -bit left-circular shift of in which the bits circulated into the LSB are complemented.

III. PROPOSED ARCHITECTURE

In the new modulo 2^n+1 multiplication, the result and one input use weighted representations, while the other input uses diminished-1 representation. Let $d[A]=(a_n a_{n-1} \ldots a_1 a_0)_2$ be the diminished-1 representation of weighted A , $B=(b_n b_{n-1} \ldots b_1 b_0)_2$ and $P = \left| A \times B \right|_{2^n+1} = (p_{n-1} p_{n-2} \ldots . p_0)_2$ all be weighted one. According toradix-4 booth recording [15] the product can be written as

$$P = \left| A \times B \right|_{2^n+1} = \left| \sum_{i=0}^{K-1} PP_i + C + K \right|_{2^n+1} \quad \text{(ix)}$$

Where

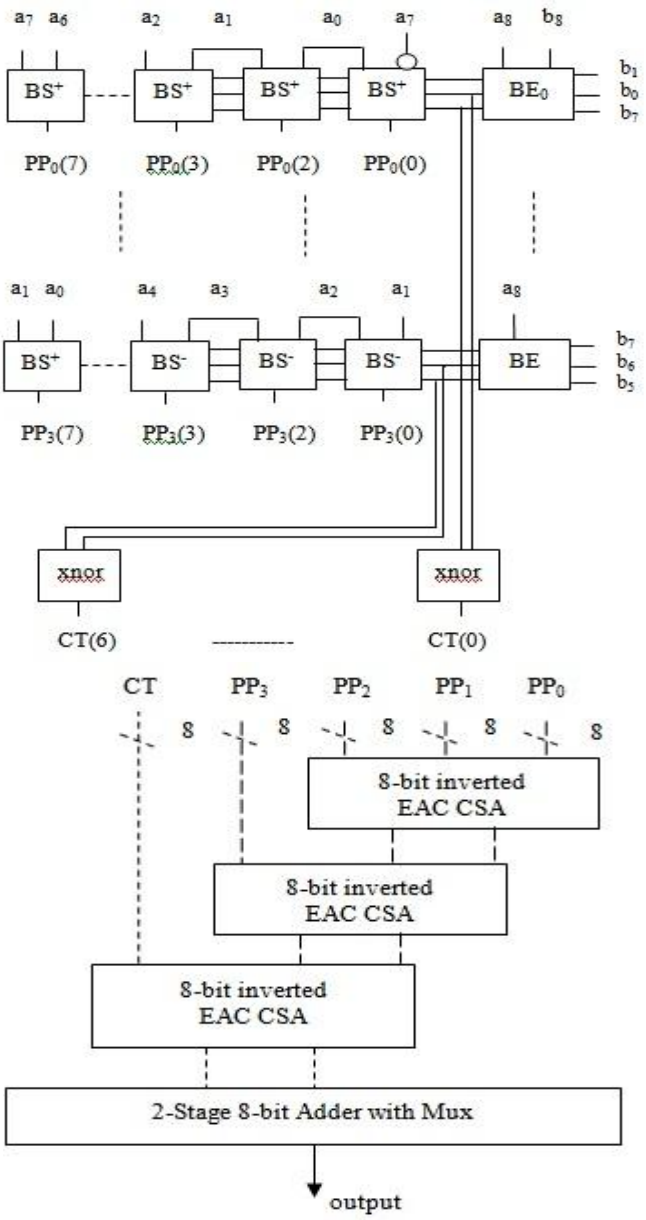
$$C = \sum_{i=0}^{K-1} c_i$$

And $$K = \begin{cases} n/2, even \\ (n+1)/2, odd \end{cases}$$

From (ix) it is clear that the architecture consists of the partial products generator (PPG), the correction tern generator (CTG), the inverted end-around-carry carry save adder (EAC CSA) and 2-stage n-bit adder. Based on this architecture, a solution which is more effective is proposed.

The encoding scheme accordant with the radix-4 Booth recoding [4], the partial product generator (PPG) can be constructed with the well-known Booth encoder (BE) and Booth selector (BS). The different blocks used in PPG and EAC CSA are taken from [4].

In this paper, we modified BE block which take successive overlapping triplets ($b_{2i+1} b_{2i} b_{2i+1}$) and encodes 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 CTG produces which has the form $(.....0x_{i+1}0x_i.....0x_10x_0)$ with $x_i \in \{0,1\}$. Since 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 block can generate the 2i-th bit x_i.

The inverted EAC CSA tree can reduce the Partial Products to two numbers. The CSA tree is usually constructed with full adders (FA).Then the final two numbers from the tree is passed through the 2-stage n-bit adder. The 2-stage n-bit adder is consisting of two ripple carry adder with C_{in}=0 and C_{in}=1and one 2:1 multiplexer. The C_{out} of first n-bit ripple carry adder is act as control signal to the multiplexer. The two n-bit sum of the ripple carry adder is given to the multiplexer. If C_{out}=0 then the final sum is the sum where the C_{in} =1 as shown in fig.(3).

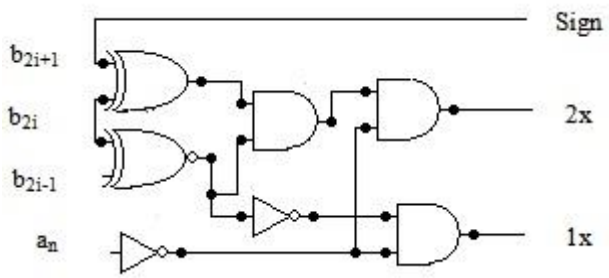


Figure 2(a): Booth encoder

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
1	0	1	1	0	1	-1
1	1	0	1	0	1	-1
1	1	1	1	0	0	-0

Figure 2(b): Truth table

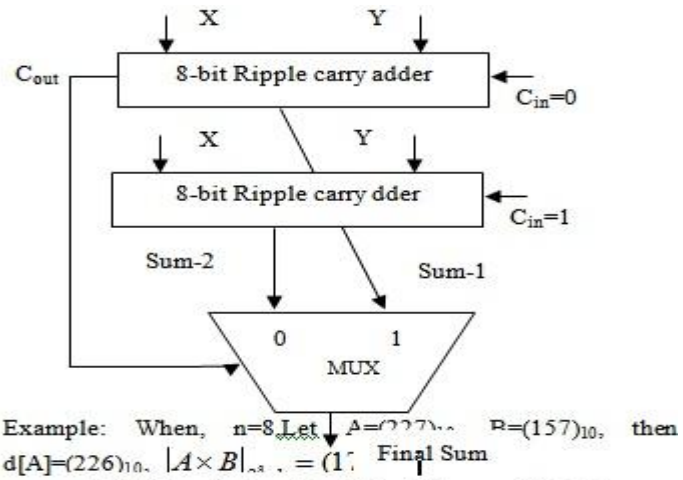


Figure 3: 2-Stage8-bit adder with Multiplexer

Example: When, n=8 ,Let A=$(227)_{10}$, B=$(157)_{10}$, then d[A]=$(226)_{10}$, $|A\times B|_{2^8+1}=(173)_{10}$.

Example

n=8,d[A]=$(11100010)_2$,B=$(10011101)_2$,$a_8=0$,$b_8=0$

Encode Partial Products

$(b_8 \vee (b_7 \oplus b_1))b_0(b_8 \vee b_7)$ --- 011---PP_0---11111111

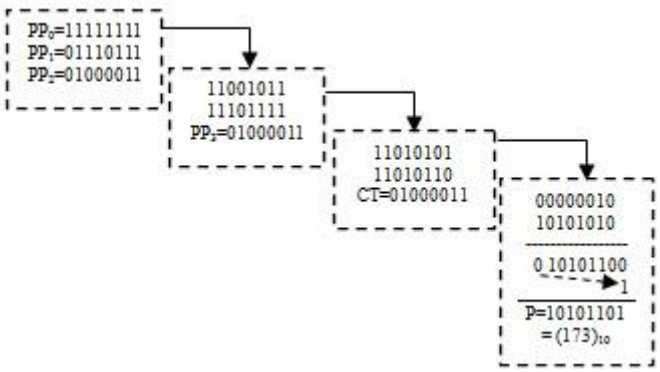
b_3 b_2 $b_1 \cdot \overline{b_7}$ ---110---PP_1---01110111

b_5 b_4 b_3 ---011---PP_2---01000011

b_7 b_6 b_5 ---100---PP_3---11110001

CT=00000001

Calculation



IV. RESULT AND SIMULATION

The proposed architecture has very low hardware complexity compared to [4], which consist of modulo 2^n+1 adder. In the proposed architecture, we use the 2-stage inverted n-bit adder. And calculate the output for 8, 12, 16-bit. To estimate the timing, area and power information for ASIC design, we have used Synopsys Design Compiler to synthesize the design into gate Level.

Comparison of Synopsys result in the proposed architecture and diminished-1 modulo 2^n+1 architecture is given in Table 1 and Table 2 respectively.

These improvements are reasonable. When compared with Diminished-1 modulo 2^n+1 multipliers for weighted representation; the blocks of the new multipliers are based on inverted n-bit adder architecture and use area-power efficient in n-bit adders.

Table 1: Synopsys Result for Area

Area(μm^2)			
Multiplier	8 bit	12 bit	16 bit
Proposed	4755.2651	8984.3446	15124.7143
Jian *et al*[4]	4901.5240	9127.5707	15370.098

Table 2: Synopsys Result for Power

Power at 50Hz(μW)			
Multiplier	8 bit	12 bit	16 bit
Proposed	13.6532	15.6768	29.0434
Jian *et al*[4]	14.2816	16.2569	30.0773

V. CONCLUSION

In this paper, we proposed the area-power efficient a modulo 2^n+1 multiplier. This architecture uses 2-stage n-bit adder, Booth recoding which reduces the number of the partial products to n/2 for even and (n+1)/2 for odd, this is the least number of the partial products among all modulo multipliers published. The reduction scheme uses the well-known inverted EAC CSA tree and the final 2-stage inverted n-bit adder generates the result. The circuit to handle the zero-input case is merged into the first Booth encoder and there is no extra delay to be added. The new multipliers, compared to existing implementations, offer better power while being more compact and their regular structure allows efficient VLSI implementations.

References

[1] R. Zimmermann," Efficient VLSI implementation of modulo $(2^n \pm 1)$ addition and multiplication," in Proc. 14th IEEE Symp. Comput. Arithm., Adelaide, Australia, Apr. 1999, pp. 158–167.

[2] Z.Wang, G. A. Jullien, andW. C.Miller, "An efficient tree architecture for modulo (2^n+1) multiplication," J. VLSI Signal Process. Syst., vol.14, no. 3, pp. 241–248, Dec. 1996.

[3] A. Curiger, H. Bonne berg, and H. Kaeslin, "Regular VLSI architectures for multiplication modulo (2^n+1) ," IEEE J. Solid-State Circuits,vol. 26, no. 7, pp. 990–994, Jul. 1991.

[4] J.W.Chen, R.H.Yao and W.J.Wu, Efficient "modulo (2^n+1) multipliers," IEEE Trans. VLSI systems, vol. 19, no 12, pp. 2149–2157, Dec. 2011.