Efficient VLSI Architecture for Modulo 2^n+1 Multiplier using n-bit Inverted Adder

Komal Gupta[1], **Prof. Amrita Pahadia**[2], **Prof. Satyarth Tiwari**[3]
M. Tech. Scholar, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[1]
Guide, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[2]
Co-guide, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[3]

***Abstract*—Efficient modulo 2^n+1 multipliers is proposed. According to our algorithm, the resulting partial products are reduced by an inverted and carry save adder to two operands, which are finally adder by a 2-stage inverted n-bit adder. By using the 2-stage inverted n-bit adder, the new multipliers reduce the number of the partial product to n/2 for even and (n+1)/2 for odd expect for one correction term. The analytical and experimental result indicates that the new modulo 2^n+1 multipliers, offer enhanced operation among all the efficient existing solutions.**

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

I. INTRODUCTION

Residue number systems (RNS) [1]-[2] reduces 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) [3]-[4], Finite Impulse Response (FIR) filters [5], image processing units [6], Discrete Cosine Transform (DCT) processors [7], communication components [8], cryptography [9], and other DSP applications [10].

In recent years, efficient schemes for modulo multipliers have been studied intensively [11]-[13]. 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. [11] 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.* [12] 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. [13] 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. [14] 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 inverted 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

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

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,

$$A = 0, d[A] = d[0] = |-1|_{2^n+1} = 2^n \tag{2}$$

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

$$d[-A] = \overline{d[A]}, if\ d[A] \in [0, 2^n - 1] \tag{3}$$

$$d[A + B] = |d[A] + d[B] + 1|_{2^n+1} \tag{4}$$

$$d[A - B] = \left|d[A] + \overline{d[B]} + 1\right|_{2^n+1} \tag{5}$$

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

$$= |d[A] \times B + B - 1|_{2^n+1} \quad (6)$$

$$d[2^k, A] = iCLS(d[A], k) \quad (7)$$

$$d[-2^k, A] = iCLS(\overline{d[A]}, k) \quad (8)$$

where $\overline{d[A]}$ represents the one's complement of d [A]. In (8) 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

A proposed 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 inverted 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 [15], 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 [15].

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 $(......0x_{i+1}0x_i......0x_1 0x_0)$ 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 inverted n-bit adder. The 2-stage inverted n-bit adder is consisting of two rows of adders. First row consist of n-bit ripple carry adder of one half adder and (n-1) full adders and the second row consist of n-bit ripple carry adder of n half adders, as shown in fig.(3).

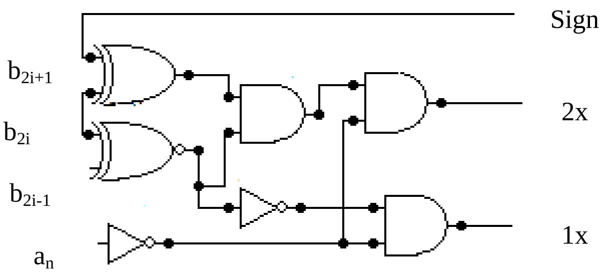


Fig. 2: Booth encoder

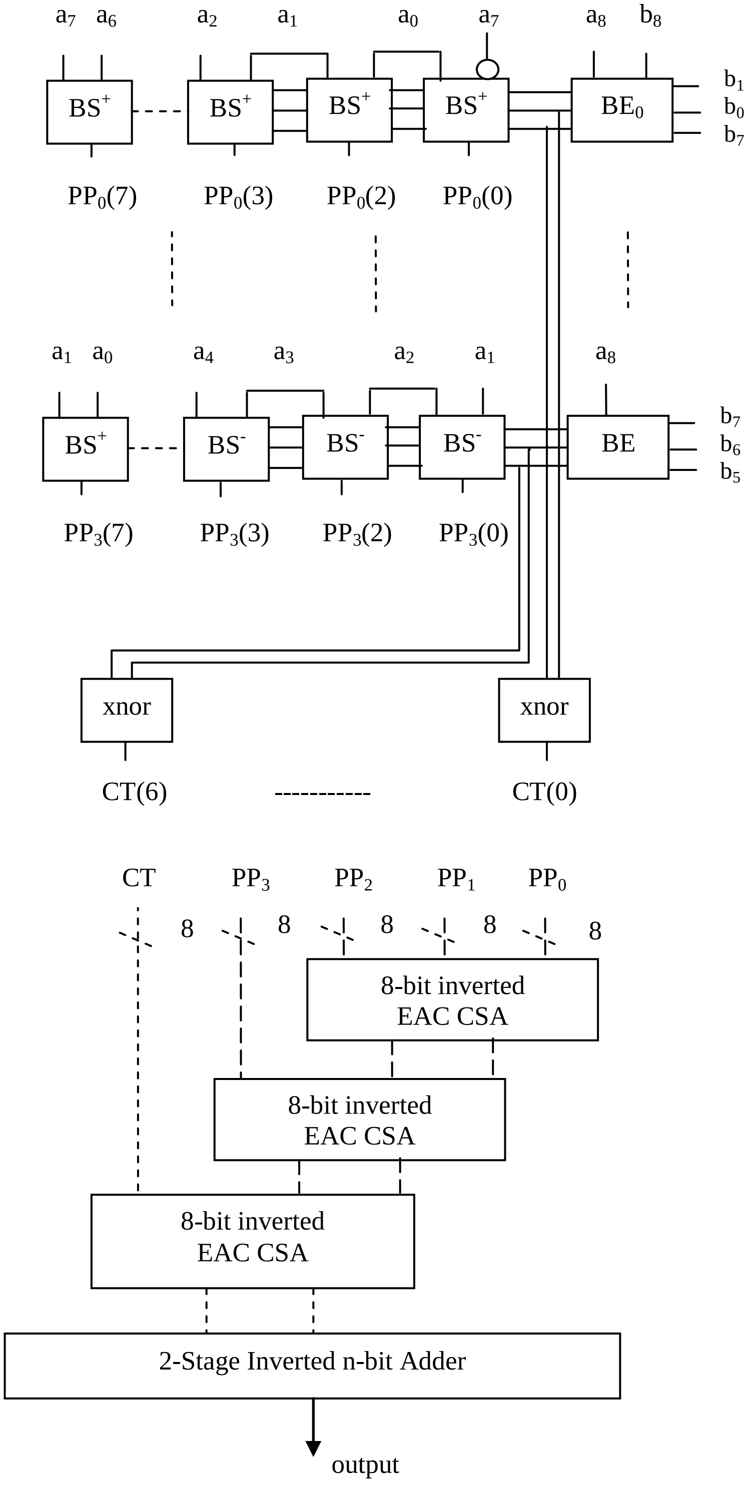


Fig.1: Architecture of the new modulo 2^n+1 multiplier

The proposed architecture in fig.1 take two input of n bit as d[A] and B and gives result as $P = |A \times B|_{2^n+1}$.Where d[A] is the diminished-1 representation of A.

Table I: True Table

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

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

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

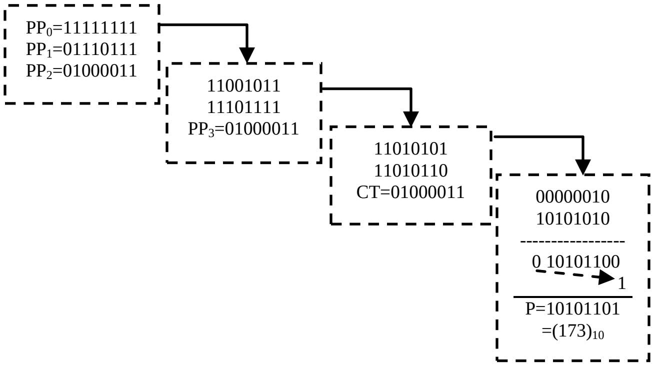
b_3 b_2 $b_1 \cdot \overline{b_7}$ ---110---PP_1---01110111

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

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

CT=00000001

Calculation



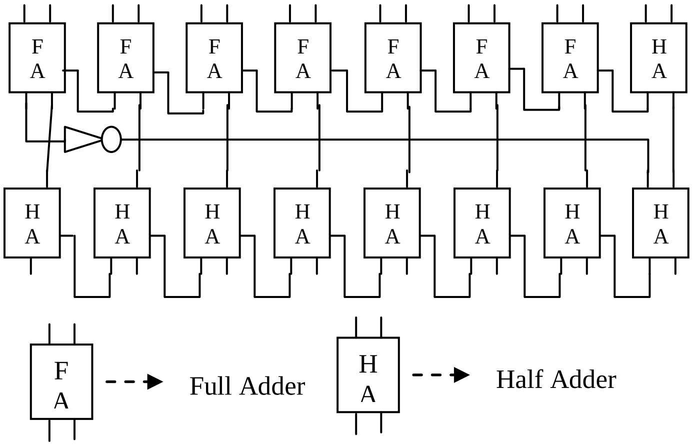


Fig. 3: 2-Stage Inverted n-bit Adder

IV. RESULT AND SIMULATION

The proposed architecture has very low hardware complexity compared to [15], 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-bit.

For a more realistic comparison, we implemented modulo multipliers for the new, [11]–[14] and [15]. At first, we used VHDL language to generate hardware models for the new and the multipliers in [11]–[14] and [15] with operand sizes of 8 and 12-bits.

Comparison of Synopsys result in the proposed architecture and diminished-1 modulo 2^n+1 architecture is given in Table II and Table III 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 efficient inverted n-bit adders.

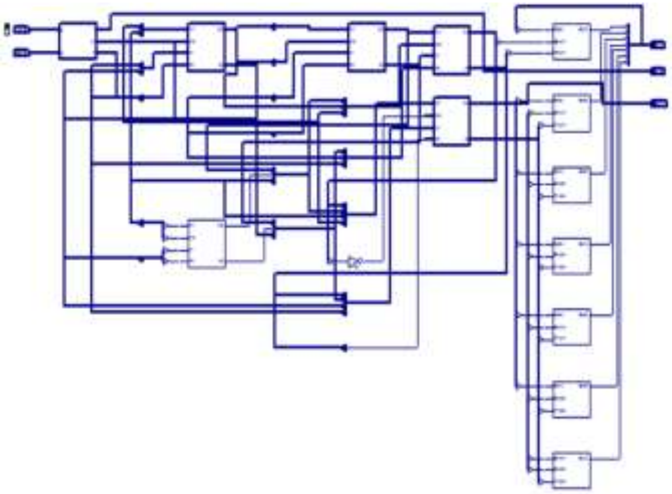


Fig. 4: RTL View of modulo 2^N+1 8-bit multiplier

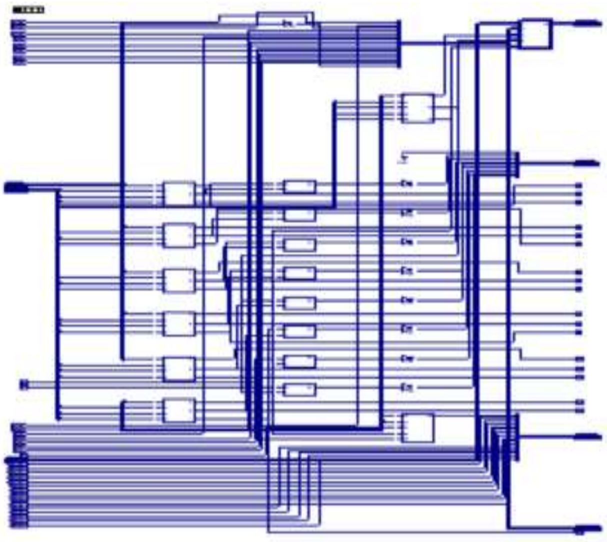


Fig. 5: RTL View of modulo 2^n+1 16-bit multiplier

Table II: Result for 8-bit

Multiplier	Previous Technique	Proposed Technique
No. of slices	2 out of 192	1 out of 192
No. Of 4 input LUTS	2 out of 384	3 out of 384
Delay	8.23 ns	8.79 ns

Table III: Result for 12 bit

Multiplier	Previous Technique	Proposed Technique
No. of slices	22 out of 192	28 out of 192
No. Of 4 input LUTS	39 out of 384	46 out of 384
Delay	19.80ns	21.24 ns

Table IV: Comparison Result of Previous and Proposed Algorithm

Bit	Proposed Algorithm	Previous Algorithm
8-bit	16.025 ns	23.02 ns
12-bit	28.062 ns	30.77 ns
16-bit	32.162 ns	34.21 ns

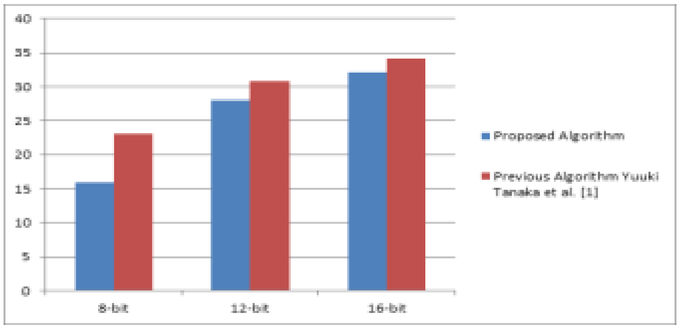


Fig. 6: Show the Bar Graph of the different bit Modulo Multiplier

V. CONCLUSION

In this paper, we proposed efficient architecture for modulo 2^n+1 multiplier. This architecture uses inverted n-bit adder Booth recoding and 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] Beerendra Kumar Patel and Jitendra Kanungo, “Efficient Tree Multiplier Design by using Modulo 2^n + 1 Adder”, Emerging Trends in Industry 4.0 (ETI 4.0), IEEE 2021.

[2] S. Janwadkar and R. Dhavse, “Qualitative and quantitative analysis of parallel-prefix adders,” in Advances in VLSI and Embedded Systems, Z. Patel, S. Gupta, and N. Kumar Y. B., Eds. Singapore: Springer Singapore, 2021, pp. 71–88.

[1] Sudhanshu Janwadkar and Rasika Dhavse, “Implementation and Performance Evaluation of Novel Line Adder Architecture for Portable Systems”, IEEE Region 10 Conference (TENCON), IEEE 2020.

[2] N. I. Chervyakov P. A. Lyakhov M. A. Deryabin N. N. Nagornov M. V. Valueva and G. V. Valuev "Residue Number System-Based Solution for Reducing the Hardware Cost of a Convolutional Neural Network" Neurocomputing vol. 407 pp. 439-453 2020.

[3] Elango Sekar and Sampath Palaniswami "Hardware Implementation of Residue Multipliers based Signed RNS Processor for Cryptosystems" J. Microelectron. Electron. Compon. Mater. vol. 50 no. 2 pp. 71-86 2020.

[4] S. Elango and P. Sampath "Implementation of High Performance Hierarchy Based Parallel Signed Multiplier for Cryptosystems" J. Circuits Syst. Comput. vol. 29 no. 13 pp. 2050214-1-2050214-25 2020.

[5] Ghassem Jaberipur Armin Belghadr and Saeed Nejati "Impact of diminished-1 encoding on residue number systems arithmetic units and converters" Computers and Electrical Engineering Elsevier vol. 75 pp. 61-76 2019.

[6] M. Sumalatha, P. Naganjaneyulu, and K. S. Prasad, “Low power and low area vlsi implementation of vedic design fir filter for ecg signal de-noising,” Microprocess Microsy, vol. 71, p. 102883, 2019.

[7] K. Desai, A. D. Darji, and H. M. Singapuri, “Implementation of high speed, low power modified vedic multiplier and its application in lifting based discrete wavelet transform,” in IEEE Region 10 Conference (TENCON). IEEE, 2019, pp. 2387–2391.

[8] R. Turaka and S. Sai, “Low power vlsi implementation of real fast fourier transform with dram-vm-cla,” Microprocess Microsy, vol. 69, pp. 92–100, 2019.

[9] K. Sivanandam and P. Kumar, “Design and performance analysis of reconfigurable modified vedic multiplier with 3-1-1-2 compressor,” Microprocess Microsy, vol. 65, pp. 97 – 106, 2019.

[10] J. Peng S. Sun Vikram K. Narayana Volker J. Sorger and Tarek el-Ghazawi "Residue number system arithmetic based on integrated nanophotonics" Optical Society of America(Optics Letters) vol. 43 no. 9 May 2018.

[11] Konstantin Isupov and Vladimir Knyazkov "Interval Estimation of Relative Values in Residue Number System" Journal of Circuits Systems and Computers vol. 27 no. 1 June 2018.

[12] Beerendra K. Patel and J. Kanungo "Diminished-1 multiplier using modulo 2n+1 adder" International journal of engineering and Technology vol. 4 no. 4 2018.

[13] B. Koziel R. Azarderakhsh and M. M. Kermani "A high-performance and scalable hardware architecture for isogeny-based cryptography" IEEE Transactions on Computers vol. 67 no. 11 pp. 1594-1609 Nov 2018.

[14] A. Jalali R. Azarderakhsh and M. M. Kermani "Neon sike: Supersingular isogeny key encapsulation on armv7" in Security Privacy and Applied Cryptography Engineering Cham:Springer International Publishing pp. 37-51 2018.

[15] P. L. H. Seo Z. Liu and Z. Hu "Sidh on arm: faster modular multiplications for faster post-quantum supersingular isogeny key exchange" IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 1-20 2018.