World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:5, No:8, 2011

Efficient Power-Delay Product Modulo $2^n + 1$ Adder Design

Yavar Safaei Mehrabani and Mehdi Hosseinzadeh

***Abstract*—**As embedded and portable systems were emerged power consumption of circuits had been major challenge. On the other hand latency as determines frequency of circuits is also vital task. Therefore, trade off between both of them will be desirable. Modulo $\mathbf{2^n + 1}$ adders are important part of the residue number system (RNS) based arithmetic units with the interesting moduli set $\{\mathbf{2^n - 1}, \mathbf{2^n}, \mathbf{2^n + 1}\}$. In this manuscript we have introduced novel binary representation to the design of modulo $\mathbf{2^n + 1}$ adder. VLSI realization of proposed architecture under 180 nm full static CMOS technology reveals its superiority in terms of area, power consumption and power-delay product (PDP) against several peer existing structures.

***Keywords*—**Computer arithmetic, modulo $\mathbf{2^n + 1}$ adders, Residue Number System (RNS), VLSI

I. Introduction

THE Residue Number System (RNS) is a non weighted integer system that by decomposing the arithmetic operations into several independent sub operations implies carry free and thereby high speed operations [1]. RNS is useful in several applications including Digital Signal Processing [2], [3], Image Processing [4], and Fast Fourier Transform computation [5]. Moreover RNS is also inherently fault tolerant against faults and makes diagnosis and correction of errors easier [6], [7].

An RNS system is based on a set of n moduli $\{m_1, m_2, \dots, m_n\}$, that are pair-wise relative prime. The number of integers that can be uniquely coded in RNS called *dynamic range* is determined by the product of the moduli $D = \prod_{i=1}^{n} m_i$. Assume that $|X|_m$ is the least nonnegative remainder of the division of X by m therefore each integer X is represented by n-tuple $X = (x_1, x_2, \dots, x_n)$ residues, where $x_i = |X|_{m_i}$ if $X \geq 0$ and $x_i = |D + X|_{m_i}$ otherwise. An RNS operation $\diamond$ is defined as $(z_1, z_2, \dots, z_n) = (x_1, x_2, \dots, x_n) \diamond (y_1, y_2, \dots, y_n)$, where $z_i = |x_i \diamond y_i|_{m_i}$ and $\diamond$ indicates addition, subtraction or multiplication. Therefore an RNS operation is decomposed into several independent and parallel operations called *channel.*

Modulo $2^n + 1$ is used in the interesting triple moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ which has several advantages [8]. The modulus of the $2^n + 1$ form is also used in five moduli set proposed in [9]. This channel is also used as an adder in the last stage of RNS multipliers. Moreover modulo $2^n + 1$ is widely applied in pseudo-random number generation, cryptography [10], and Data Encryption Algorithm [11]. Therefore effective design of modulo $2^n + 1$ adder in terms of latency, power consumption, and power-delay product (PDP) is a vital task. The structure proposed in this manuscript is not competitive with regard to delay, but that it offers advantages in area, power, and PDP parameter.

Over the years, many papers have been addressed on the design of modulo $2^n + 1$ adders (e.g., [12], [13], [14], and [15]) that the adder proposed in [15] is the most efficient one. We have compared our approach against the corrected version of previously published design in [15] which is described in [16] and the classic modulo adder which is proposed in [12] but uses ripple adders in its structure.

The rest of manuscript is organized as follows. First we introduce new binary representation in section II and then design its corresponding modulo adder in section III. We will show simulation results in section IV. Finally there are conclusions in section V.

II. Novel Representation

The bit positions in our novel number representation system have weights similar to conventional binary number system except that two least significant bits (LSBs) have the same weights. As a matter of fact, the weights of the bit positions are $2^n, 2^{n-1}, \dots, 2^2, 2^1, 2^0, 2^0$. There is an example in table I which illustrates representation of residues in the modulus of $2^2 + 1$ in the novel binary system.

TABLE I
REPRESENTATION OF RESIDUES IN BOTH CONVENTIONAL AND PROPOSED NUMBER REPRESENTATIONS IN THE MODULO $2^2 + 1$

Number	Conventional	Propose Representation
0	000	000
1	001	001
2	010	011
3	011	101
4	100	111

Note that in cases that there are two ways for representation of numbers as number 2 which can be represented by either "0011" or "0100" the used convention in our approach is the using of least significant bits, therefore number 2 will be represented by "0011".

We will show that this new system will save area, power consumption, and PDP parameter with regard to earlier proposals.

Yavar Safaei Mehrabani is with the Department of Computer Engineering, Science and Research Branch, Islamic Azad university, Tehran, Iran (corresponding author e-mail: y.safaei@ srbiau.ac.ir).

Mehdi Hosseinzadeh is with the Department of Computer Engineering, Science and Research Branch, Islamic Azad university, Tehran, Iran as an assistant professor(e-mail: hosseinzadeh@srbiau.ac.ir).

III. MODULO $2^n + 1$ ADDER

Let $A = (a_n, a_{n-1}, \dots, a_1, a_0)$ and $B = (b_n, b_{n-1}, \dots, b_1, b_0)$ denote two nonnegative (n+1)-bit binary integers in the proposed number representation. Since two LSBs in each operand have the same weights therefore summation of these operands in the two LSB positions differ from conventional binary system. We have shown the summation of these positions in table II.

TABLE II
SUMMATION OF TWO LSBs IN THE OPERANDS A AND B

$b_1\ a_1\ b_0\ a_0$	$c\ \ s_1\ \ s_0$
0 0 0 0	0 0 0
0 0 0 1	0 0 1
0 0 1 0	0 0 1
0 0 1 1	0 1 1
0 1 0 0	x x x
0 1 0 1	0 1 1
0 1 1 0	x x x
0 1 1 1	1 0 1
1 0 0 0	x x x
1 0 0 1	x x x
1 0 1 0	0 1 1
1 0 1 1	1 0 1
1 1 0 0	x x x
1 1 0 1	x x x
1 1 1 0	x x x
1 1 1 1	1 1 1

The symbol 'x' in table II indicates do not care cases, because in the novel number representation there are no such representations. Considering table II, we have designed corresponding circuits for the outputs of c, s_1, and s_0 that they have suggested in Fig. 1.

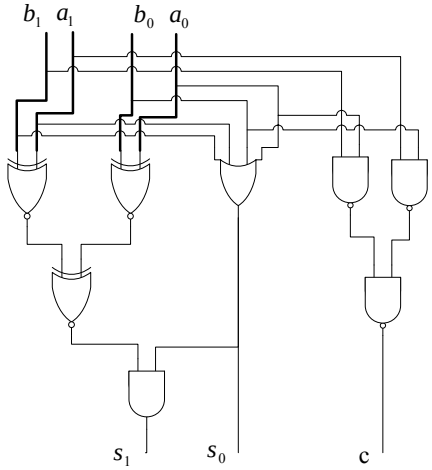

Fig. 1 Logic circuit of c, s_1, and s_0

Residue addition of two operands X and Y in modulo $2^n + 1$ and normal binary system is defined as:

$$Z = |X + Y|_{2^n+1} = \begin{cases} X + Y & (a) \\ X + Y + (2^n - 1) & (b) \end{cases} \quad (1)$$

In the case of (a) from (1), the result of addition is correct whereas in the case of (b) as result exceeds the module, therefore it should be corrected by subtracting of $2^n + 1$ from the result.

$$Z = X + Y - (2^n + 1) \quad (2)$$

By some modifications of (2) it can be rewritten as:

$$Z = X + Y + (2^n - 1) - 2^{n+1} \quad (3)$$

In order to implementation of (3) we can ignore output carry from position of 2^n and add constant value of $2^n - 1$ to the result of $X + Y$. In the proposed number representation system there are two representations for the constant value. Considering modulo $2^3 + 1$ addition, the constant value will be 7 which can be represented by either "1101" or "1110". Since existence of logical '0' in the LSB position of the latest one results adders at the LSB positions of the modulo adder are removed and width of multiplexer decreased from (n+1)-bit to n-bit therefore by choosing the constant value of "1110", modulo $2^n + 1$ adder will be realized more effectively in VLSI criteria. In general, for modulo $2^n + 1$ adder constant value is an (n+1)-bit binary string that has one '0' in its LSB position and logical '1's in the other positions.

We have illustrated our approach in Fig.2 by an example. Considering modulo $2^3 + 1$ addition in the proposed novel system with operands A and B we have shown addition operation for three cases. If both carry outputs from phases 1 and 2 are logical '1' then sum of phase 2 instead of phase 1 will be selected by the multiplexer. Considering table II, the LSB of the sum is always logical '1' except when sum is zero and this case as suggested in Fig. 1 can be detected with a four input OR gate.

$$\begin{array}{lll} \begin{array}{l} cout_1 = 0 \\ \\ \\ cout_2 = 1 \end{array} \begin{array}{r} 1111 \\ 0000 \\ \hline 11\,1 \\ 11\,0 \end{array} \begin{array}{l} \\ +\,phase\,1 \\ \\ +\,phase\,2 \end{array} & \begin{array}{l} cout_1 = 1 \\ \\ \\ cout_2 = 0 \end{array} \begin{array}{r} 1001 \\ 0111 \\ \hline 00\,0 \\ 11\,1 \end{array} + & \begin{array}{l} cout_1 = 1 \\ \\ \\ cout_2 = 1 \end{array} \begin{array}{r} 1111 \\ 1001 \\ \hline 10\,0 \\ 11\,1 \\ \hline 01\,1 \end{array} + \\ (a) & (b) & (c) \end{array}$$

Fig. 2 (a) $A + B < 2^3 + 1$ (b) $A + B = 2^3 + 1$
(c) $A + B > 2^3 + 1$

Finally, we have suggested the structure of our novel modulo $2^n + 1$ adder for n=4 in the Fig. 3. The multiplexer is controlled by SEL signal which is produced by a gate that ANDs carry outputs from two adders.

The H.A* in the Fig. 3 is a Full Adder which one of its inputs is always driven by logical '1' [14]. The structure of H.A* is suggested in Fig. 4.

Moreover, as input carry of H.A* in the position of x_1 from Fig. 3 is always logical '0', so we have replaced it with an equivalent NOT gate.

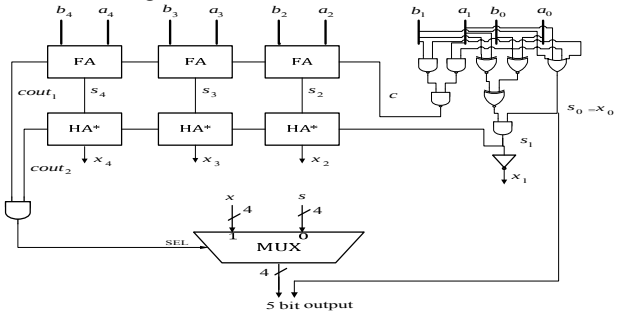

Fig.3 The structure of modulo $2^4 + 1$ adder

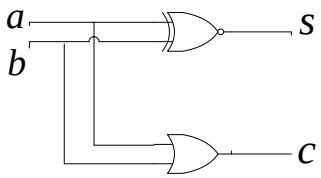

Fig. 4 The logic diagram of H. A*

IV. COMPARISONS

We have compared our novel modulo adder against those reported in [15] and [12]. Note that as suggested in Fig. 5 we have applied ripple adders in the structure of modulo adder proposed in [12].

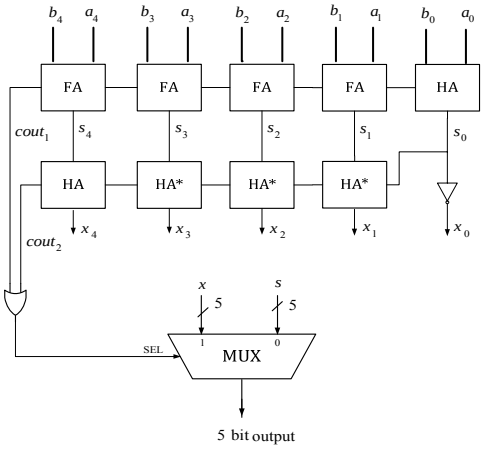

Fig. 5 The structure of modulo adder proposed in [12]

All structures are mapped into 180 nm full static CMOS technology with power supply of 1.8v and temperature of $25°C$. In order to quantitative comparisons we ran simulations, for n= 4, 6, and 8 using HSPICE tool. The results for different values of n in modulo $2^n + 1$ adder appear in tables III, IV, and V. Considering cases that n is not power of 2, since there is no formal way to design of TPP therefore it is not applicable (N.A.) whereas our described modulo $2^n + 1$ adder can be applied for any number of n (not necessarily power of 2).

TABLE III
COMPARISON RESULTS FOR MODULO $2^n + 1$ ADDER USING 180 nm CMOS TECHNOLOGY WITH N=4

Adder	Area (Number of transistors)	Power (μW)	Delay (ns)	PDP (fJ)
Bayoumi [12]	194	24.483	0.5757	14.094
TPP [15]	408	52.829	0.5259	27.782
Proposed	184	25.327	0.3704	9.381

TABLE IV
COMPARISON RESULTS FOR MODULO $2^n + 1$ ADDER USING 180 nm CMOS TECHNOLOGY WITH N=6

Adder	Area (Number of transistors)	Power (μW)	Delay (ns)	PDP (fJ)
Bayoumi [12]	278	41.210	0.5919	24.392
TPP [15]	N.A.	N.A.	N.A.	N.A.
Proposed	268	35.999	0.6144	22.117

TABLE V
COMPARISON RESULTS FOR MODULO $2^n + 1$ ADDER USING 180 nm CMOS TECHNOLOGY WITH N=8

Adder	Area (Number of transistors)	Power (μW)	Delay (ns)	PDP (fJ)
Bayoumi [12]	362	52.663	0.8197	43.167
TPP [15]	852	109.55	0.6275	68.742
Proposed	352	47.716	0.8391	40.038

Our approach is slower than compared methods for N=6 and 8 but tables VI and VII indicate that area, power consumption, and PDP parameter of our approach are more efficient than previous proposals.

TABLE VI
SAVINGS OFFERED BY PROPOSED METHOD AGAINST [15] (%)

	Proposed versus TPP [15]			
N	Number of transistors	Power (μW)	Delay (ns)	PDP (fJ)
4	54.9	52.05	29.56	66.23
6	N.A.	N.A.	N.A.	N.A.
8	58.68	56.44	-33.72	41.75

TABLE VII
SAVINGS OFFERED BY PROPOSED METHOD AGAINST [12] (%)

	Proposed versus Bayoumi [12]			
N	Number of transistors	Power (μW)	Delay (ns)	PDP (fJ)
4	5.15	3.44	35.66	33.43
6	3.59	12.64	-3.8	9.32
8	2.76	9.39	-2.36	7.24

V. CONCLUSIONS

Modulo $2^n + 1$ adder is a fundamental module in the most arithmetic units based on RNS. In this paper a novel binary representation has been proposed and utilized to the design of modulo $2^n + 1$ adder. The proposed architecture has been simulated and evaluated in 180 nm CMOS process technology with HSPICE software. Simulation results indicate that proposed modulo adder is more efficient in respect of area, power consumption, and PDP parameter than compared architectures.

REFERENCES

[1] G. Lakhani, "VLSI design of modulo adders/subtractors," IEEE Int. conf. on Computer Design, ICCD'92, October 1992, PP. 68-71.
[2] W. L. Freking, and K. K. Parhi, "Low-Power FIR digital filters using residue arithmetic," proc. of 31^{th} Asilomar Conference on Signals, Systems, and Computers, Vol. 1, November 1997, PP. 739–43.
[3] F. Taylor, "A Single Modulus ALU for Signal Processing," IEEE Trans. on Acoustics, Speech, Signal Processing, Vol. 33, 1985, PP. 1302-1315.
[4] M. Bhardwaj, and B. Ljusanin, "The Renaissance-A Residue Number System Based Vector Co-Processor for DSP Dominated Embedded

ASICs," Proc. of Asimolar conference on Signals, Systems, and computers, 1998, PP. 202-207.
[5] P. G. Fernandez, A. Garcia, J. Ramirez, L. Parrilla, and A. Lioris, "A RNS-Based Matrix-Vector-Multiply FCT architecture for DCT computation," Proc. 43rd IEEE Midwest Symposium On circuits and systems, 2000, PP. 350-353.
[6] E. Kinoshita, and K. Lee, "A Residue Arithmetic Extension for Reliable Scientific Computation," IEEE Trans. on Computers, Vol. 46, No. 2, 1997, PP. 129-138.
[7] V. Paliouras, and T. Stouraitis, "Novel High-Radix Residue Number System Architectures," IEEE Trans. circuits SYST. II, Vol. 47, No. 10, October 2000, PP. 1059-1073.
[8] A. S. Molahosseini, and K. Navi, "New arithmetic Residue to Binary converters," International Journal of Computer Sciences and Engineering Systems, Vol. 1, No. 4, October 2007, PP. 295-299.
[9] B. Cao, C. H. Chang, and T. Srikanthan, "A residue-to-binary converter for a new 5-moduli set," IEEE Trans. circuits SYST. I, Vol. 54, No. 5, 2007, PP. 1041-1049.
[10] A. Curiger, "VLSI Architectures for computations in finite rings and fields," ph. d. thesis, Swiss federal institute of technology, 1993.
[11] R. Zimmermann, and *et al.*, "A 177Mb/s VLSI implementation of the international data encryption algorithm," IEEE J. solid-state circuits, Vol. 29, No. 3, 1994, PP. 303-307.
[12] M. Bayoumi, and G. Jullien, "A VLSI Implementation of Residue Adders," IEEE Trans. Circuits and systems, Vol. 34, 1987, PP. 284-288.
[13] M. Dugdale, "VLSI Implementation of Residue Adders based on binary Adders," IEEE Trans. circuits SYST. II, Vol. 39, No. 5, 1992, PP. 325-329.
[14] A. A. Hiasat, "High-speed and Reduced Area Modular Adder structures for RNS," IEEE Trans. on Computers, Vol. 51, No. 1, 2002, PP. 84-89.
[15] C. Efstathiou, and H. T. Vergos, "Fast Parallel-Prefix modulo $2^n + 1$ Adder," IEEE Trans. on Computers, Vol. 53, No. 9, 2004, PP. 1211-1216.
[16] G. Jaberipur, and B. Parhami, "Unified Approach to the design of Modulo-$(2^n \pm 1)$ Adders Based on Signed-LSB Representation of Residues," in proc. of the 19^{th} IEEE Symposium on computer arithmetic, June 2009, PP. 57-64.