ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

A New Class Of Modulo $2^n - 2^k - 1$ Adder for Multi-Channel RNS

Karunamoorthi T[1], Babykala M[2]

Department of Electronics and Communication Engineering[1],K. S. Rangasamy College of technology, Tiruchengode, Namakkal, India.

Department of Electronics and Communication Engineering [2],K. S. Rangasamy College of technology, Tiruchengode, Namakkal, India.

ABSTRACT –In Residue Number System (RNS) the modular adder is one of the major key component of RNS applications. For multichannel RNS processing the moduli set with the form 2^N-2^K-1 can offer excellent balance. In this algorithm, parallel prefix operation and carry correction techniques are used to eliminate the recomputation of carries. Moreover, any existing parallel prefix structure can be used in the proposed structure and it gives flexible tradeoff between area and delay. Compared with same type modular adder, the proposed modulo2^N-2^K-1 adder gives better performance in delay and area.

INDEX TERMS–Carry correction, modular adder, parallel prefix, residue number system (RNS),VLSI.

I. INTRODUCTION

RNS is an ancient numerical representation system. It is recorded in one of Chinese arithmetical masterpieces, the Sun Tzu Suan Jing, in the 4th century and transferred to European known as Chinese Remainder Theorem (CRT) in the 12th century. RNS is a non-weighted numerical representation system and has carry-free property in multiplication and addition operations. In recent years, it has been received intensive study in the very large scale integration circuits (VLSI) design for digital signal processing (DSP) systems with high speed and low power consumption [1]-[4]. For integers A and B with n-bit width, the modular addition can be performed if A and B is less than the modulus m.

$$C = \begin{cases} A+B & A+B+T < 2^n \\ (A+B+T)_{2^n} & A+B+T \geq 2^n \end{cases} \quad (1)$$

In the general modular adder design, the two values A+B, and A+B+T, should be computed firstly [5] [8]. Then, one of them is selected as the final output. According to the form of the modulus, modular adders can be classified into two types: the general modular adder and the special modular adder.Bayoumi proposed a scheme for arbitrary modulus by using two cascaded binary adders [5]. However, the delay is the sum of the two binary adders.Dugdale proposed a method to construct a type of general modular adders with a reused binary adder [9]. The shortage of this structure is that it will use two operation cycles to perform one modular addition.Hiasat proposed a class ofmodular adders in which any regular Carry Look-Ahead (CLA)based binary adder can be used in the final stage [10]. However, it needs an extra CLA unit to get the carry-out bit of A+B+T before the final CLA addition. As a result, the structure does not reduce the delay significantly.Patel *et al.* [12] also proposed several algorithms which can generate carriesfastly. A new number representation for modulo additionis proposed in [8].However, its outputs are represented inspecialformat. Thus, the extra area and delay are needed to performthe conversion from the special representation to binarynumber representation or all operations should be performed inthis number representation format in RNS-based systems. In the proposed scheme, the carry information of $A+B+T$ computed by prefix computation unit is modified twice to obtain the final carries required in the sum computation module.

Meanwhile, any existing fast prefix structure of binary adders can be used in the proposed modular adder structure, which offers superior flexibility in design.One of the important issues is the selection of moduli sets inRNS-based application. In addition and multiplication intensive systems, residue channels are always expected as many as possible when the dynamic range is fixed, that is, the word length of individual residue can be reduced to achieve better speed performance. Meanwhile, the width of eachchannel is also expected as close as possible to get similar critical path delay. That is the balance between each residue channel.

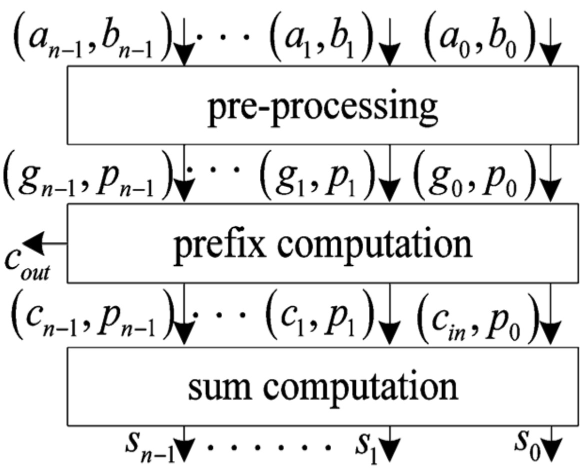
(a_{n-1}, b_{n-1}) · · · (a_1, b_1) (a_0, b_0)
pre-processing
(g_{n-1}, p_{n-1}) · · · (g_1, p_1) (g_0, p_0)
c_{out}
prefix computation
(c_{n-1}, p_{n-1}) · · · (c_1, p_1) (c_{in}, p_0)
sum computation
s_{n-1} · · · · · · s_1 s_0

Fig 1. Prefix Computation-Based Adder Structure.

In the rest of the paper, the brief introduction of RNS and modular addition are presented in Section II. Section III introduces the algorithm and hardware architecture of the proposed modulo 2^N-2^K-1adder.

II. BACKGROUND

A. *RNS and Modular Addition*

RNS is defined as a group of co-prime modular radixes{m1,m2…..m_n}, where N>1,GCD(m_i,m_j)=1,i=j,i,j=1,2…N and GCD (m_i,m_j) and is the greatest common divisor of m_i and m_j and . The integer X can be represented uniquely by its residues respect to the modulus m_i,ie., (x_1,x_2,…xn),

$$X_i = (X)m_i, M = \prod_{i=1}^{N} m_i (2)$$

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 A,B and C in the range of (0,M). For integers A and B in the range of (0,M) , modulo addition is defined as

$$C = (A + B)_m = \begin{cases} A + B & A + B < m \\ A + B - m & A + B \geq m \end{cases} (3)$$

If C= $(A+B)_m$and the bit width of the modular adder is n-bit, where n=($\log_2$m),(that is, is the smallest integer less than $\log_2$m).Equation [3] can be represented as

$$C = \begin{cases} A + B & A + B + T < 2^n \\ (A + B + T)_{2^n} & A + B + T \geq 2^n \end{cases} (4)$$

where the correction T=2^{n-m}[7] [8] [20]. That is, if the carry-out bit of A+B+T is "1", the result of modular addition is the least significant bits of A+B+T, otherwise, the result is A+B. This is the basic rule in most modular adders design.

B. *Prefix Parallel Addition*

Parallel prefix operation is widely adopted in binary adder design. Each sum bit s_i and carry bit c_i can be calculated with the previous carries and inputs [22]. Prefix based binary adders can be divided into three units, the pre-processing unit, the prefix computation unit, and the sum computation unit. In the pre-processing unit, prefix computation is calculated as

$$(g_i, p_i) = (a_i b_i, a_i \oplus b_i) (5)$$

where$g_i$and p_i (i=0, 1, n-1) represent the carry generation bit and carry propagation bit respectively. The prefix computation unit is used to compute the carry information used in the sum computation unit. Prefix computation can be performed by

$$\begin{aligned} (G_{i:i}^0, P_{i:i}^0) &= (g_i, p_i) \\ (G_{i:k}^l, P_{i:k}^l) &= (G_{i:j+1}^{l-1}, P_{i:j+1}^{l-1}) \bullet (G_{j:k}^{l-1}, P_{j:k}^{l-1}) \\ &= (G_{i:j+1}^{l-1} + P_{i:j+1}^{l-1} G_{j:k}^{l-1}, P_{i:j+1}^{l-1} P_{j:k}^{l-1}) \end{aligned}$$

where (i=0,1,…,n-1), *l* =1,2,…m and *l* represents the *l*th stage. The smaller *l* means the shorter delay of the carry chain. The operator "." [5]is the prefix operator and the prefix computation result of the *l*thstage from the kth bit to the ith bit, which is also called group prefix computation.

There are several well-known binary prefix addition structures such as Sklansky (SK), Brunt-Kung (BK), Kogge-Stone (KS), Han-Carlson (HC), ELM and so forth [22]. These prefix structures are usually called prefix trees.After prefix computation, carries c_i (i=0, 1, 2….n) for the ith bit can be obtained can be computed as

$$c_0 = c_{in}$$
$$c_i = G_{i-1:0}^l + P_{I-1:0}^l + c_{in}$$
$$c_{out} = c_n (7)$$

In the sum computation unit, the carries c_i from the prefix computation unit and the partial sum p_i from the pre-processing unit are used together to compute the final sum bits s_i.

$$s_i = p_i \oplus c_i (8)$$

III. PROPOSED MODULO 2^N-2^K-1 ADDER

Modulo 2^N-2^K-1 adder is composed of four modules, pre-processing unit, carry generation unit, carry correction and sum computation unit. In Fig. 2, different shade represents different processing units. The proposed modular adder can be divided into two general binary adders, A1and A2 in Fig .2, with carry correction and sum computation module according to the characteristics of T for modulus 2^N-2^K-1.

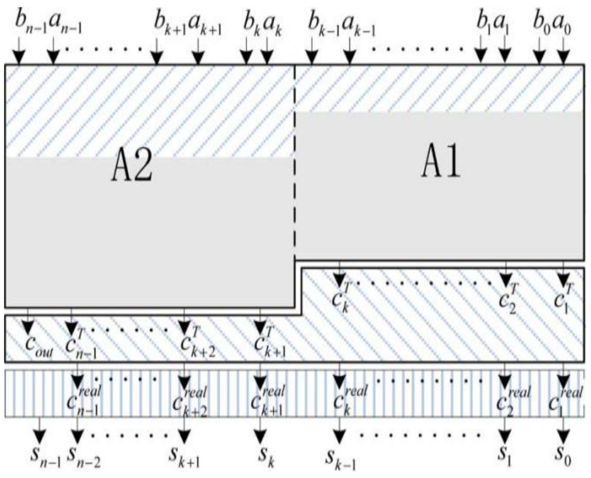

Fig.2. The Proposed Modulo 2^N-2^K-1 Adder

At last, the final modular addition result from C_i^{real} and partial sum information. The architecture shown in Fig. 2 can avoid the calculation of carries information for A+B+T and A+B separately.

A. Preprocessing Unit

The pre-processing unit is used to generate the carry generation and carry propagation bits (g_i,p_i) of A+B+T.From (3) when m = 2^N-2^K-1.

$$T = 2^{[\log_2 2^n - 2^k - 1]} - m = 2^k + 1 \qquad (9)$$

The difference is that the lowest carry-in bit should be considered. Therefore, carry generation and carry propagation bits are

$$\begin{cases} (g_o.p_o) = (a_o + b_0.\overline{a_o \oplus b_i}) & i = 0 \\ (g_i.p_i) = (a_i b_i.a_i \oplus b_i) & i = 1,2 \ldots\ldots k-1 \end{cases}$$

(10)

B. Carry Generation Unit

In carry generation unit, the carries C_i^T(i= 1, 2…n) of A+B+T can be obtained with the carry generation and carry propagation bits from the pre-processing unit. Any existing prefix structure can be used to get the carries C_i^T. It is worth pointing out that the carry-out bit of SCSA in the pre-processing unit is not involved in the prefix computation. Instead, c_{SCSA} combined with the carry-out bit of the prefix tree is required to determine the carry-out bit of A+B+T (denoted as C_{out}).

$$c_{out} = c_{SCSA} + c_n^T = c_{SCSA} + G_{l-1:0}$$

$$= c_{SCSA} + G_{n-1:l} + P_{n-1:l}G_{l-1:0}$$

$$= c_{SCSA} + G_{n-1:l} + P_{n-1:l}C_l^T \,(11)$$

C. Carry Correction Unit

The carry correction unit is used to get the real carries $c_i^{real}$for each bit needed in the final sum computation stage. In order to reduce the area and get the carries of A + B by correcting the carries of in the carry correction unit.Meanwhile, z1 and z2 can be pre-computed and used as the inputs of the MUX. Similarly, let Z_1^1 and Z_2^1be the value of z_1 and z2 when c_k^T = 1 respectively.

$$z_1 = \overline{c_k^T} z_1^0 + c_k^T z_1^1$$
$$= \overline{c_k^T}(P_{k-1:0}$$
$$+ \overline{p_k})$$
$$+\overline{c_k^T}(P_{k-1:0} + p_k)$$

$$z_2 = \overline{c_k^T} z_2^0 + c_k^T z_2^1 = \overline{c_k^T}(P_{i:k+1})(P_{k-1:0} +$$
$$pk+ckTPi:k+1Pk-1:0+pk) \quad (12)$$

Thus, we can get the carry information that will be used in the sum computation unit of the proposed modular adder.

D. The sum computation

Generally, the sum computation is as same as that in prefixbasedbinary adder. However, $c_i^{real}$is the correction result whenis taken into account. That isC_{out}= 0, if, is the carrybit of A + B. Otherwise, it is the carry bit ofA+B+T. Thus the partial sum bits of A+B and A+B+T are both requiredin the final sum computation. Let P_i^oandP_i^1 (i = 0, 1, 2…n - 1) be the partial sum bits of A+B and A+B+T respectively.

$$s_i = \begin{cases} c_{out} \oplus \overline{p_o} & i = 0 \\ c_k^{real} \oplus c_{out} \oplus \overline{p_k} & i = 0 \\ c_i^{real} \oplus p_i \quad i = 1..k-1, k+1 \ldots n-1. \end{cases}$$

(13)

In (13)$c_{out} \oplus \overline{p_k}$and C_k^{real} can be obtained at the same time. Therefore, there is no extra delay compared with other sum computation units.

IV. MODULO 2^N-2^K-1 ADDER STRUCTURE

Modulo adder is composed of four modules, pre-processing unit, carry generation unit, carry correction unit, and sum computation unit. Modular adder can be divided into two general binary adders. A1 and A2 with carry correction and sum computation module according to the characteristics of correction T for modulus 2^N-2^K-1.

This algorithm can construct a new class of general modular adder with better performance in area and delay. A new class of RNS with larger dynamic and more balanced complexity among each residue channel.

A. Preprocessing Unit

Pre-processing unit is used to generate the carry generation and carry propagation bits (gi,pi). The pattern “0” is the pre-processing unit and used to generate carry generation and carry propagation (10), (11), (12) bits for the following prefix computation.

B. Prefix Computation Unit

The pattern "●" is the prefix computation unit. The Sklansky prefix tree is used and there are eleven prefix computation units. The delay of "●" is determined by its carry generation path which is one OR gate and one AND gate. However, the pattern "●"in the final stage of prefix tree is not needed to compute propagation bits (4).This unit can be divided into two sections namely

- Carry Generation Unit
- Carry Correction Unit

C. Carry Generation Unit

In carry generation unit, the carries C_i^T(i=1, 2….n)of (X+Y+T) can be obtained with the carry generation andcarry propagation bits from the pre-processing unit. Instead, C_{csca} combined with the carry-out bit of the prefix tree is required to determine the carry-out bit of X+Y+T (denoted as C_{out}).

D. Carry Correction Unit

In this seven correction operators are used for three different situations, that is i=0,1,….,k-1,i=k andi=k+1,…,n-2. The $P_{i:0,}$ z1 and z2 can be computed by independent modules. C_k^t is computed out before c_i^t(i=k+1, n-1) with two prefix computation stages.In the worst case, the group propagation bits required are needed to be computed one by one from p_i(i=0,.. n-2). However, the extra components for computing these group propagation bits exist in prefix structure.

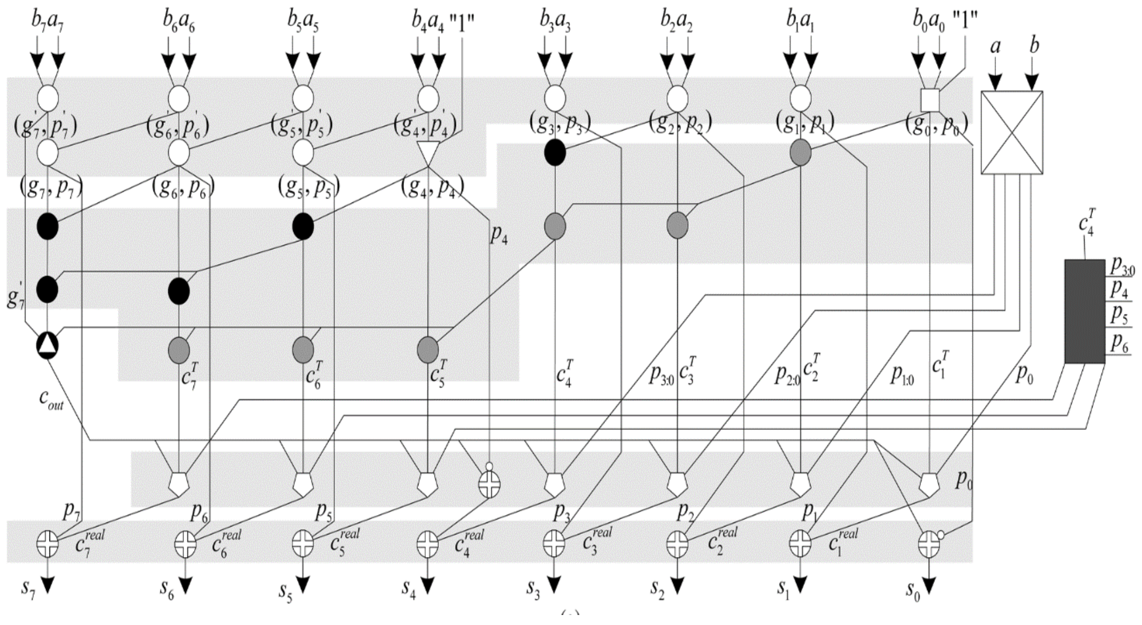

Fig.3.Modulo2^N-2^K-1 Adder Structure

E. um Computation Unit

This unit is used for performing the sum computation. The computation can be performed with carry correction simultaneously, only one XOR operations are required to perform the sum computation and no extra delay is introduced.

COMPARISON RESULTS

TABLE 1

AREA REPORT

Modulo 2^N-2^K-1 Adder			
Logic Utilization	**Used**	**Available**	**Utilization**
Number Of Slices	18	768	2%
Number Of 4input LUTs	32	1536	2%
Number Of Bounded IOBs	24	182	13%
Modulo 2^N-1 Adder			
Logic Utilization	**Used**	**Available**	**Utilization**
Number Of Slices	69	768	8%
Number Of 4input LUTs	120	1536	7%
Number Of Bounded IOBs	26	182	14%

TABLE 2
DELAY REPORT

Adder	**Modulo 2^N-2^K-1 Adder**	**Modulo 2^N-1 Adder**
Delay(ns)	6.514	6.613

V. CONCLUSION

A new class of modulo 2^N-2^K-1 adder is proposed. This structure is consisting of four units, the pre-processing, the carry computation, the carry correction and the sum computation unit. A comparison shows that the algorithm can construct a new class of general modular adder with better performance in "area and delay". It has some main features as following: The way using twice carry corrections improves the performance of area and reduces the redundant units for parallel computation of A+B+T and A+B in the traditional modular adders. Furthermore, the modulus with the form of 2^N-2^K-1 facilitates the construction of a new class of RNS with larger dynamic and more balanced complexity among each residue channels.

REFERENCES

[1] Shang Ma, Jain-Hao Hu, and Chen-Hao Wang, "A novel modulo 2^N-2^K-1adder for residue number system," in Proc. IEEE Transactions on circuits and systems i:regular papers.2013.

[2] S. Ma, J. H. Hu, L. Zhang, and L. Xiang, "An efficient RNS parity checker for moduli set {2^n-1.2^n+1.2^{2n}+1} and its applications," Sci. in China, Ser. F: Inform. Sci., vol. 51, no. 10, pp. 1563–1571, Oct. 2008.

[3] Y.Liu and E.M.-K. Lai, "Design and implementation of an RNS-based 2-D DWT processor," IEEE Trans. Consum, Electron., vol. 50, no. 1, pp. 376–385, Feb. 2004.

[4] P. Patronik, K. Berezowski, S. J. Piestrak, J. Biernat, and A. Shrivastava, "Fast and energy-efficient constant-coefficient FIR filters using residue number system," in Proc. Int. Symp. Low Power Electronicsand Design (ISLPED), pp. 385–390, 2011.

[5] J. C. Bajard, L. S. Didier, and T. Hilaire, "Direct form transposed and residue number systems for filter implementations," in Proc. IEEE54th Int. Midwest Symp. Circuits and Systems (MWSCAS), pp.1–4, 2011.

[6] S. J. Piestrak, "Design of residue generators and multioperand modular adders using carry-save adders," IEEE Trans. Comput,, vol. 43, no. 1, pp. 68–77, Jan. 1994.

[7] H. Vergos, "On the design of efficient modular adders," J. Circuits, Syst., and Comput., vol. 14, no. 5, pp. 965–972, Oct. 2005.

[8] G. Jaberipur, B. Parhami, and S. Nejati, "On building general modular adders from standard binary arithmetic components," in Proc. 45thAsilomar Conf. Signals, Systems, and Computers, pp. 6–9, 2011.

[9] A. Hiasat, "High-speed and reduced-area modular adder structures for RNS," IEEE Trans. Comput,, vol. 51, no. 1, pp. 84–89, Jan. 2002.

[10] R. A. Patel, M. Benaissa, N. Powell, and S. Boussakta, "ELMMA: A new low power high-speed adder for RNS," in Proc. IEEE Workshopon Signal Processing Systems, pp. 95–100, Oct. 2004.

[11] R. A. Patel, M. Benaissa, N. Powell, and S. Boussakta, "Novel power-delay-area-efficient approach to generic modular addition,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1279–1292, Jun. 2007.

[12] Vassalos, D. Bakalis, and H. T. Vergos, "Modulo 2^n+1arithmetic units with embedded diminished-to-normal conversion," in Proc. 14thEuromicro Conf. Digital System Design (DSD), pp. 468–475, 2011.

[13] Jaberipur and S. Nejati, "Balanced minimal latency RNS addition for moduli set {2^n-1.2^n+1.2^n+1} „" in Proc. 18th Int. Conf. Systems,Signals and Image Processing (IWSSIP), pp. 1–7, 2011.

[14] T. Vergos and C. Efstathiou, "A unifying approach for weighted and diminished-1 modulo 2^n+1 addition," IEEE Trans. Circuits Syst. II,Exp. Briefs, vol. 55, no. 10, pp. 1041–1045, Oct. 2008.

[15] S. H. Lin and M. H. Sheu, "VLSI design of diminished-one modulo 2^n+1 adder using circular carry selection," IEEE Trans. Circuits Syst.II, Exp. Briefs, vol. 55, no. 9, pp. 897–901, Sep. 2008.

[16] Efstathiou, H. T. Vergos, and D. Nikolos, "Fast parallel-prefix modulo 2^n + 1 adders," IEEE Trans. Comput., vol. 53, no. 9, pp. 1211–1216, Sep. 2004.

[17] [18] R. A. Patel and S. Boussakta, "Fast parallel-prefix architectures for Modulo 2^n - 1 addition with a single representation of zero," IEEETrans. Comput., vol. 56, no. 11, pp. 1484–1492, Nov. 2007.

[18] P. M. Matutino, R. Chaves, and L. Sousa, "Arithmetic units for RNS moduli 2^n - 3 and 2^n + 3 operations," in Proc. 13th Euromicro Conf.Digital System Design: Architecture, Methods and Tools (DSD), 2010, pp. 243–246.

[19] R. A. Patel, M. Benaissa, and S. Boussakta, "Fast modulo 2^n - (2^n - 2 + 1) addition: A new class of adder for RNS," IEEE Trans. Comput., vol. 56, no. 4, pp. 572–576, Apr. 2007.

[20] L. Li, J. Hu, and Y. Chen, "An universal architecture for designing modulo 2^n– 2^p - 1multipliers," IEICE Electron. Expr., vol. 9, no. 3, pp. 193–199, Feb. 2012.

[21] R. Zimmermann, "Binary Adder Architectures for Cell-Based VLSI and their Synthesis," Ph.D. dissertation, Integrated Syst. Lab., SwissFederal Inst. of Technol., Zurich, 1997.