Novel Modulo 2^n+1 Subtractor and Multiplier

Dina Younes
Department of Microelectronics
Brno University of Technology
Brno, Czech Republic
e-mail: xyoune00@stud.feec.vutbr.cz

Pavel Steffan
Department of Microelectronics
Brno University of Technology
Brno, Czech Republic
e-mail: steffan@feec.vutbr.cz

***Abstract*— This paper introduces a novel design of modulo 2^n+1 subtractor in Residue Number System (RNS). A novel design of modulo 2^n+1 multiplier has also been introduced by utilizing the presented subtractor. The two designs are suitable for small values of n, as they depend on the normal (binary) representation of RNS numbers, instead of diminished-one representation which has difficulties in representing zero operands and results, and need to deal with them separately. The presented circuits were implemented and simulated using VHDL to prove the theoretical consideration.**

Keywords-Residue number system; modulo 2^n+1; subtractor; multiplier

I. Introduction

The residue number system (RNS) is a non-weighted integer number system [1], which decomposes binary large numbers to smaller residues. Obviously, there is no carry propagation problem between residues [2]. RNS offers the potential for high-speed and parallel arithmetic. The RNS based on the moduli set (2^n-1, 2^n, 2^n+1) is most frequently utilized to achieve a high-performance RNS application since the resulting RNS architecture performs fast residue arithmetic.

The 2^n+1 modulo arithmetic gains the most significant importance because modulo 2^n+1 channel is critical for whole system in terms of area and delay. It is considerably more difficult than 2^n-1 modulo, in the sense that it cannot be realized with the same speed or efficiency [3]. The representation of a number in modulo 2^n+1 arithmetic has to be (n+1) bit long instead of n-bit, so that the whole system will require blocks that are (n+1) bits long. Therefore 2^n+1 adders and multipliers became of interest to many researches.

In many publications about modulo 2^n+1 adders and multipliers [4][5][6], the diminished-one representation of residues has been used in order to solve the problem of (n+1) bit long operands and use only n-bit long operands. As a result, only n-bits are used in the computation units. But this representation has some difficulties in representing zero operands and results as it has to be treated separately. However, by using small values of n, the normal representation of residues can be used without causing a considerable effect on the overall delay of the system.

Small values of n will produce small dynamic ranges. Small dynamic ranges are usually less than 15 bits. The presented design is very attractive for many digital signal processing applications that use small dynamic ranges. An example of such application is the [0,255] range grayscale image data [7]. Or even RGB images with 8 bit format. For this application, a small moduli set {7,8,9} will be enough for the encoding. The dynamic range of this set is small (9 bits).

RNS multipliers are widely used in many DSP applications, such as image processing, FIR filters and Fourier transform.

In this paper, a novel design of modulo 2^n+1 subtractor is presented, and it has been used in designing modulo 2^n+1 multiplier. These designs are suitable for small values of n, which is favorable from a practical point of view because the overall speed of system will be increased [8].

The proposed circuits were implemented and simulated using VHDL.

The organization of this paper is as below:

In Section 2, an overview of RNS arithmetic is given. The design of the proposed subtractor is shown in Section 3. In Section 4, the proposed multiplier and a description of its circuit are presented. The results and a comparison with an existing 2^n+1 multiplier are stated in Section5. Finally, the conclusion is discussed in Section 6.

II. RNS Overview Arithmetic

The primary advantage of RNS is that addition, subtraction, and multiplication can be performed independently and in parallel on the various residues.

The residual number system (RNS) is defined by a set of numbers $m_1, m_2, \ldots, m_n$ called the moduli. Where the great common divider (GCD) for m_i , m_j = 1. In this system, an integer X is represented by an ordered set of residues, $\{x_1, x_2, \ldots, x_n\}$ where:

$$x_i = X \bmod m_i \qquad (1)$$

Where X: $$0 \le X < \prod_{i=1}^{N} m_i$$

Arithmetic operations (addition, subtraction and multiplication) are performed totally parallel on those residues.

Assuming that A and B are two RNS numbers; the addition (subtraction) of these two numbers is given by:

ISBN:978-1-61208-114-4

$$A \pm B = \left\{ \left|a_1 \pm b_1\right|_{m_1}, \left|a_2 \pm b_2\right|_{m_2}, \ldots, \left|a_N \pm b_N\right|_{m_N} \right\} \quad (2)$$

Also the multiplication of A and B is given by:

$$A \times B = \left\{ \left|a_1 \times b_1\right|_{m_1}, \left|a_2 \times b_2\right|_{m_2}, \ldots, \left|a_N \times b_N\right|_{m_N} \right\} \quad (3)$$

The strongest points in this system are the independency, and carry free among the residues. In other words, each residue can be treated as a separated integer.

III. Proposed RNS Subtractor

Subtraction is an operation widely met in digital signal processing applications [9] [10] for operations such as mean error estimation, mean square error estimation and calculation of sum of absolute differences. Since modulo arithmetic is also frequently used in these types of applications, efficient modulo subtraction circuits are welcome. However, very little work [11] has been focused on designing modulo 2^n+1 subtractors.

To design modulo 2^n+1 subtractor, a binary subtractor, a binary adder and a multiplexer have been used, as shown in Fig. 1. This simple structure and small number of elements used in the circuit provide less delay and more efficiency to accomplish subtractive operation.

The operands used are (n+1) bit long, because they are residues resulted from binary to RNS conversion with respect to modulo 2^n+1. The output of the proposed subtractor is also (n+1) bit long.

For modulo $m=2^n+1$, the subtraction of two residues is defined as:

$$C = \left|A - B\right|_{2^n+1} = \begin{cases} \left|A - B\right|_{2^n+1} & if \;\; A - B \geq 0 \\ \left|A - B + \left(2^n + 1\right)\right|_{2^n+1} & if \;\; A - B < 0 \end{cases} \quad (4)$$

The second case is implemented by adding 1 to the result of subtraction [12].

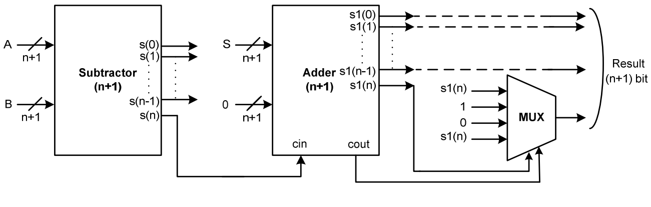

Figure 1. Proposed modulo 2^n+1 subtractor

However dealing with operands of (n+1) bit long is not same as dealing with just n-bit, because the MSB (2^n bit) will create some confusion, especially when $A<B$. To overcome that confusion, a multiplexer has been added to correct the final output of the proposed subtractor.

For example: n=4, $m=2^n+1=17$, A=0, B=1
$|A-B|_{2n+1} = |0-1|_{17} < 0$

```
 0:     0 0000
-1:     1 1111 –
        1 1111
             1 +
        0 0000
```

By adding '1' to correct the result, we got "0 0000", instead of the correct result "1 0000". Therefore the multiplexer was added to overcome cases like this one. The carry out after the addition is '1' and the MSB is '0', so according to the multiplexer, the MSB will become '1', and the final output will become "1 0000".

IV. Proposed RNS Multiplier

Let $A=a_n \ldots a_0$ and $B=b_n \ldots b_0$ refer to two (n+1) bit modulo 2^n+1 operands, such that $0 \leq A,B \leq 2^n$. The multiplication of A and B is given by:

$$R = A \times B \quad (5)$$

Assuming that A, B are expressed using (n+1) bits:

$$R = \sum_{i=0}^{2n} r_i 2^i \quad (6)$$

Where r_i is the ith bit resulting from AB:

$$R = \left| \sum_{i=0}^{n-1} r_i 2^i + \left|2^n\right|_{2^n+1} \sum_{i=n}^{2n} r_i 2^{i-n} \right|_{2^n+1} \quad (7)$$

$$R = \left|X + 2^n Y\right|_{2^n+1} \quad (8)$$

An important aspect in RNS is:

$$\left|2^n\right|_{2^n+1} = \left|2^n + 1 - 1\right|_{2^n+1} = \left|-1\right|_{2^n+1} \quad (9)$$

Substituting equation (9) in (7):

$$R = \left| \sum_{i=0}^{n-1} r_i 2^i - \sum_{i=n}^{2n} r_i 2^{i-n} \right|_{2^n+1} \quad (10)$$

$$R = \left|X - Y\right|_{2^n+1} \quad (11)$$

The presented modulo 2^n+1 multiplier circuit is shown in Fig. 2. It consists of a binary multiplier (n+1)×(n+1), and a modulo 2^n+1 subtractor. The output of the multiplier is separated into two (n+1) bit operands, and fed into modulo 2^n+1 subtractor.

As noticed from equation (11), to multiply two modulo 2^n+1 numbers, a modulo 2^n+1 subtractor is needed.

The presented modulo 2^n+1 subtractor is used, but the operands of that subtractor are (n+1) bit long, therefore we just made the MSB of X = '0'; so both operands become (n+1) bit long, and can be applied to the subtractor to acquire the correct result.

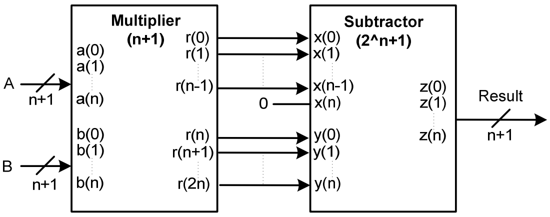

Figure 2. Proposed modulo 2^n+1 multiplier

V. Implementations and Results

The presented modulo 2^n+1 multiplier has been implemented and simulated using VHDL. A comparison of this multiplier with an existing one [13] has been done. Table 1 shows the delay comparison between the two designs. As can be noticed that for small values of n, the delay in the presented multiplier is considerably smaller than the one stated in [13], and that makes it more efficient for applications requiring small dynamic ranges, such as image processing of images with 8 bit format. But as the value of n increases, the delay also increases. We have noticed that at the value of n=12 the delay in the multiplier stated in [13] becomes less.

Table 1. The Delay Comparison between the Presented Multiplier and the Multiplier Stated in [13]

n	[13]	The presented multiplier	Delay reduced
3	20.2 ns	15.2 ns	24.75 %
4	25 ns	18.3 ns	26.8 %
8	31 ns	27.5 ns	11.3 %
10	35.6 ns	32.6 ns	8.4 %
11	36 ns	34.9 ns	3.1 %
12	36.2 ns	36.8 ns	-
14	40.6 ns	43.7 ns	-

VI. Conclusion

Novel simplified architectures of 2^n+1 modulo subtractor and multiplier have been presented and detailed in this paper. To realize modulo 2^n+1 subtractor, a binary subtractor, a binary adder and a multiplexer have been used. The proposed subtractor has been used to realize modulo 2^n+1 multiplier. RNS residues were presented using binary representation instead of diminished-one representation that has been used recently. This representation acquires additional components to solve the difficulties resulted in zero representing. The main advantages of the proposed circuits are design simplicity, reduced computation complexity and reduced delay in the system when using small values of n. These circuits can be effectively used in digital signal processing applications that require a small dynamic range, such as image processing of images with 8 bit format, where the range of image data is [0,255]. The effectiveness of these architectures has been proved by presenting a comparison with an existing 2^n+1 modulo multiplier. This comparison has been done using VHDL implementation and simulation of the design.

Acknowledgment

Funding for this work was provided by Czech Ministry of Education in the frame of Research Program MSM0021630503 MIKROSYN New Trends in Microelectronic Systems and Nanotechnologies and ENIAC JU Project No. 120001 Nanoelectronics for an Energy Efficient Electrical Car and ARTEMIS JU Project No. 100205 Process Oriented Electronic Control Units for Electric Vehicles Developed on a multi-system real-time embedded platform.

References

[1] M. A. Sonderstrand , "Residue Number System Arithmetic". Modern Applications in Digital Signal Processing, New York: IEEE Press, 1986.

[2] Szabo and R. Tanaka, "Residue arithmetic and its applications to computer technology". New York, McGraw-Hill, 1967.

[3] A. Omondi and B. Premkumar, "Residue Number Sustem Theory and Implementation". Imperial College Press, 2007.

[4] H. T. Vergos, C. Efstathiou, and D. Nikolos, "Diminished-One Modulo 2^n+1 Adder Design," *IEEE Transactions on Computers*, vol. 51, no. 12, pp 1389-1399, December 2002.

[5] H.T. Vergos, and D. Bakalis, "On the Use of Diminished-1 Adders for Weighted Modulo 2^n+1 Arithmetic Components," In *Proc. Euromicro Conference on Digital System Design*, pp. 752-759, 2008.

[6] C. Efstathiou, H. T. Vergos, G. Dimitrakopoulos and D. Nikolos, "Efficient diminished-1 modulo 2^n+1 multipliers," *IEEE Transactions on Computers*, vol. 54, no. 4, pp. 491–496, April 2005.

[7] Wei Wang, M.N.S. Swamy, and M.O. Ahmad, "RNS Application for Digital Image Processing", In *System-on-Chip for Real-Time Applications, 2004.Proceedings. 4th IEEE International Workshop on*, pp. 77-80, July 2004.

[8] B. Parhami, "Computer arithmetic: algorithms and hardware designs", Oxford, 2002.

[9] P. G. Fernandez, and A. Lloris, "RNS-based implementation of 8x8 point 2D-DCT over field-programmable devices," *Electronics Letters*, vol. 39, no. 1, pp. 21-23, January 2003.

[10] P.M. Matutino, and L. Sousa, "An RNS based Specific Processor for Computing the Minimum Sum-of-Absolute-Differences," In *Proc. Euromicro Conference on Digital System Design*, pp. 768-775, 2008.

[11] S. Timarchi, K. Navi, and M. Hosseinzade, "New Design of RNS Subtractor for modulo 2^n+1," In *Proc. Int. Conference on Information and Communication Technologies*, pp 2803-2808, 2006.

[12] A. Hiasat, "New Memoryless, Mod *(2^n - 1)* Residue Multiplier", *Electronics Letters,* vol. 28, no. 3, pp. 314-315, January 1992.

[13] R. Zimmermann, "Efficient VLSI implementation of modulo (2^n±1) addition and multiplication", In *Computer Arithmetic, 1999. Proceedings. 14th IEEE Symposium on*, pp. 158-167, 1999