Aleksandr CARIOW, Galina CARIOWA
WEST POMIERANIAN UNIVERSITY OF TECHNOLOGY, SZCZECIN
49 Żołnierska St., 71-210, Szczecin, Poland

A Hardware-Efficient Structure of Complex Numbers Divider

Abstract

In this correspondence an efficient approach to structure of hardware accelerator for calculating the quotient of two complex-numbers with reduced number of underlying binary multipliers is presented. The fully parallel implementation of a complex-number division using the conventional approach to structure organization requires 4 multipliers, 3 adders, 2 squarers and 2 divider while the proposed structure requires only 3 multipliers, 6 adders, 2 squarers and 2 divider. Because the hardware complexity of a binary multiplier grows quadratically with operand size, and the hardware complexity of an binary adder increases linearly with operand size, then the complex-number divider structure containing as little as possible embedded multipliers is preferable.

Keywords: complex-number divider, hardware complexity reduction, VLSI implementation.

1. Introduction

Mathematical calculations with complex numbers are widely used today for data processing in numerous fields of science and engineering such as modern digital signal and image processing, computer games and 3D graphics applications, control of power systems, wireless communication, and optical systems. The use of complex number mathematics greatly enhances the power of data processing, offering techniques which cannot be implemented with the help of real number operations. In comparison with real-valued data processing, the complex-valued data processing is more abstract and theoretical, but also more universal and comprehensive [1-3]. Perhaps it is easier to list the fields where these numbers are not applicable than those where they are applied.

In complex-valued arithmetic the most time and area consuming operations are multiplication and division of two complex numbers. What is more, the division is even more complicated and expensive than multiplication [3-7].

The schoolbook multiplication of complex numbers requires performing 4 real multiplications and 2 real additions, and the schoolbook division of complex numbers requires performing 4 real multiplications, 2 real squarings, 3 real additions and 2 real divisions. In turn, multiplication and division of two ordinary real numbers are also more time consuming operations than addition or subtraction of ordinary real numbers. Since these operations are carried out repeatedly, the total time to the correct implementation of the final algorithm may be unacceptable. This problem it becomes extremely challenging for applications requiring real-time processing at high throughput especially in digital signal and image processing. Hence, to meet the stringent requirements to throughput, latency, and power-consumption constraints of real-time DSP systems, developing dedicated hardware implementations, such as application specific integrated circuits (ASICs) or field-programmable gate arrays (FPGAs), is of paramount importance. It is therefore evident that finding ways, which reduce the number of real multiplications required for performing multiplication and division of complex numbers, is very important task.

In 1805 Gauss had discovered a way (so called Gauss's complex multiplication trick) of reducing the number of multiplications to three [8, 9]. The Gauss's optimization saves one real multiplication out of four. In [10] a Gauss-like trick that can save one multiplication in the case of division of complex numbers was offered. The aim of the present communication is to suggest an efficient hardware accelerator structure for dividing complex numbers.

2. Statement of the problem

Let us suppose that, we need to divide two complex numbers

$$y = z_1 / z_2 \tag{1}$$

where $z_1 = a + ib$, $z_2 = c + id$, and $y = e + if$ are complex numbers; a, b, c, d are real variables, and i is the imaginary unit, satisfying $i^2 = -1$.

The task is to calculate the quotient defined by the expression (1) with the minimal multiplicative complexity.

A schoolbook method of finding the quotient of two complex numbers can be represented by the following equations:

$$e = \frac{ac + bd}{c^2 + d^2}, \quad f = \frac{ad - bc}{c^2 + d^2} \tag{2}$$

Direct implementation of the calculations in accordance with these equations requires 4 multiplications, 3 additions, 2 squarings and 2 divisions of real numbers. Below we will show how you can reduce the multiplicative complexity of the operation of calculating the quotient of two complex numbers.

3. The structure

Let us introduce the two column vectors: $\mathbf{X}_{2\times1} = [a, b]^{\mathrm{T}}$ and $\mathbf{Y}_{2\times1} = [e, f]^{\mathrm{T}}$. Let also $R = c^2 + d^2$.

Now the division of complex numbers can be presented in the matrix–vector multiplication form as

$$\mathbf{Y}_{2\times1} = \frac{1}{R}\mathbf{A}_2\mathbf{X}_{2\times1} \tag{3}$$

where

$$\mathbf{A}_2 = \begin{bmatrix} c & d \\ d & -c \end{bmatrix}$$

The direct multiplication of the vector–matrix product in Eq. (3) requires 4 real multiplications and 2 real additions. We propose a simple way to reduce multiplicative complexity of this operation to 3 real multiplications at the price of 3 more real additions. Proposed way will be quite similar to the Gauss's trick for complex number multiplication.

It is easily to verify [11] that the matrix with this structure can be factorized, than the computational procedure for calculate the product of $\mathbf{A}_2\mathbf{X}_{2\times1}$ can be represented as follows:

$$\mathbf{Y}_{2\times1} = \mathbf{D}_2\mathbf{T}_{2\times3}\mathbf{D}_3\mathbf{T}_{3\times2}\mathbf{X}_{2\times1}$$

where

$$\mathbf{D}_3 = diag(d_0, d_1, d_2) = diag\{(c-d), d, -(c+d)\},$$

$$\mathbf{D}_2 = \begin{bmatrix} R^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix},\ \mathbf{T}_{2\times 3} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix},\ \mathbf{T}_{3\times 2} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix},$$

Fig. 1 shows a data flow diagram of the rationalized way for computation of the quotient of two complex numbers. Data flow diagram is oriented from left to right. Straight lines in the figure denote the operations of data transfer. We use the usual lines without arrows on purpose, so as not to clutter the picture. Points where lines converge denote summation. The circles in these figures show the operation of multiplication by a real number inscribed inside a circle.

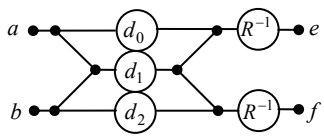

Fig. 1. The data flow diagram of proposed complex-number divider

Fig. 2 shows a hardware efficient structure of the dedicated circuit for computation of the quotient of two complex numbers. The rectangles in the figure denote computation blocks that implement arithmetic operations on binary data, the essence of which is clear from the context.

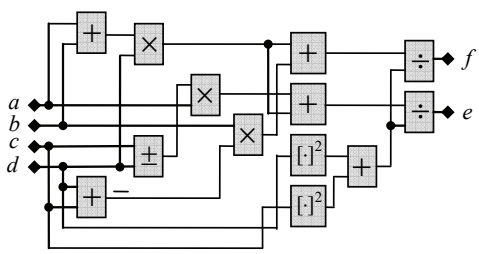

Fig. 2. Hardware efficient scheme of the accelerating circuit for computation of the quotient of two complex numbers

4. Conclusions

We presented a hardware oriented structure for calculating the quotient of two complex numbers. The fully parallel implementation of a complex number divider on the base of the naive method takes 4 multipliers, 3 adders, 2 squaring units and 2 dividers of binary numbers while the fully parallel implementation of proposed structure requires only 3 multipliers, 6 adders, 2 squaring units and 2 dividing units of binary numbers. So, the use of proposed algorithmic solution reduces the multiplicative complexity of complex number dividing, thus reducing hardware implementation complexity and leading to a high-speed circuit suitable for VLSI implementation. In low power VLSI design, optimization must be primarily done at the level of logic gates amount. From this point of view a multiplication requires much more intensive hardware resources than an addition. Moreover, a multiplier occupies much more area and consumes much more power than an adder. This is because the hardware complexity of a multiplier grows quadratically with operand size, while the hardware complexity of an adder increases linearly with operand size. Therefore, the algorithm for division of complex numbers containing as little as possible of real multiplications is preferable.

5. References

[1] Brown J. A., Crowder H.: Graphics Application Using Complex Numbers in APL2 Technical Report: TR 03.265, Santa Teresa Laboratory San Jose, CA. March 1985.

[2] Martin K.: Complex signal processing is not – complex. Proc. of the 29th European Conf. on Solid-State Circuits (ESSCIRC'03), 2003, pp. 3-14, Estoril, Portugal, 16-18 Sept. 2003.

[3] Nikolova Z., Iliev G., Ovtcharov M. and Poulkov V.: Complex Digital Signal Processing in Telecommunications. In: Applications of Digital Signal Processing, edited by Christian Cuadrado-Laborde. InTech Open Access Publisher, 2011, pp. 3-24.

[4] Cariow A., Cariowa G.: A rationalized algorithm for complex-valued inner product calculation. Pomiary Automatyka Kontrola, 2012, 58, 674-676.

[5] Stewart G.W.: A note on complex division. ACM Transactions on Mathematical Software, 1985, vol. 11, no 3, pp. 238–324, doi:10.1145/214408.214414.

[6] Varghese A. A. Pradeep, C., Eapen, M. E., Radhakrishnan, R.: FPGA Implementation of Area-Efficient IEEE 754 Complex Divider, International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015): Procedia Technology, vol. 24, 2016, pp. 1120–1126, doi:10.1016/j.protcy.2016.05.245.

[7] López-Martínez, F. J., del Castillo-Sánchez, E, Entrambasaguas J. T., Martos-Naya E.: Iterative-Gradient Based Complex Divider FPGA Core with Dynamic Configurability of Accuracy and Throughput. Journal of Signal Processing Systems, 2011, vol. 62, no 3, pp. 319-324, doi 10.1007/s11265-010-0464-y.

[8] Knuth D.E.: The Art Of Computing Programming. Volume 2, Seminumerical Algorithms, Addison-Wesley, Reading, MA, USA, Second Ed., 1981.

[9] Blahut R.E.: Fast algorithms for digital signal processing. Addison-Wesley Publishing company, Inc. 1985.

[10] Cariow A.:An algorithm for dividing two complex numbers. CoRR abs/1608.07596 2016, pp. 1-4.

[11] Cariow A.: Strategies for the Synthesis of Fast Algorithms for the Computation of the Matrix-vector Products, Journal of Signal Processing Theory and Applications, 2014, vol. 3 No. 1 pp. 1-19. doi:10.7726/jspta.20141001.

Received: 21.03.2017 *Paper reviewed* *Accepted: 03.05.2017*

Prof. Aleksandr CARIOW, DSc.

He received the Candidate of Sciences (PhD) and Doctor of Sciences degree (DSc or Habilitation) in Computer Sciences from LITMO of St. Petersburg, Russia in 1984 and 2001, respectively. In September 1999, he joined the faculty of Computer Sciences and Information Technology at the West Pomeranian University of Technology, Szczecin, Poland, where he is currently a professor and chair of the Department of Computer Architectures and Telecommunications. His research interests include digital signal processing algorithms, VLSI architectures, data processing parallelization.

e-mail: acariow@wi.zut.edu.pl

Galina CARIOWA, PhD

She received the MSc degree in Mathematics from Moldavian State University, Chişinău in 1978, and PhD degree in computer science from West Pomeranian University of Technology, Szczecin, Poland in 2007. She is currently working as an assistant professor of the Department of Multimedia Systems. She is also an Associate-Editor of World Research Journal of Transactions on Algorithms. Her scientific interests include numerical linear algebra and digital signal processing algorithms, VLSI architectures, and data processing parallelization.

e-mail: gcariowa@wi.zut.edu.pl