Improving Fixed-Point Implementation of QR Decomposition by Rounding-to-Nearest

Sergio D. Muñoz and Javier Hormigo
Dept. Computer Architecture
Universidad de Malaga
Malaga, Spain
fjhormigo@uma.es

***Abstract*—QR decomposition is a key operation in many current communication systems. This paper shows how to reduce the area of a fixed-point QR decomposition implementation based on Givens rotations by using a new number representation system. This new representation allows performing round-to-nearest at the same cost of truncation. Consequently, the rounding errors of the results are halved, which allows it to reduce the word-length by one bit. This reduction positively impacts on the area, delay and power consumption of the design.**

Keywords—QRD; CORDIC; Consumer communication; MIMO; adaptive filters; watermarking; fixed-point optimization

I. INTRODUCTION

QR Decomposition (QRD) is a well-known operation in linear algebra. It consists of decomposing a given matrix as a product of an orthogonal matrix and an upper triangular one. Nowadays, QRD is used in many digital signal processing applications, such as wireless communications [1][2] and image watermarking [3]. The hardware implementation of this matrix operation has been thoroughly studied in the literature. Many of these implementations are based on fixed-point numbers and truncation to maintain the bit width. Although, it is demonstrated that round-to-nearest produces less rounding errors, the cost of implementing this rounding mode prevents its use on these applications.

Recently, in [4] a new representation system which allows performing round-to-nearest simply by truncation has been presented. Thanks to the improvements in the rounding errors of arithmetic operations, a very significant area reduction on the implementation of FIR filters has been reported in [4].

In this paper, we study how this new representation system can improve the hardware implementation of QRD. Specifically, we focus on the implementation of QRD by using Givens rotations, which is the most used in hardware designs.

II. HARDWARE IMPLEMENTATION OF QRD

A. Givens Rotation Algorithm

The Givens method performs the QRD by using unitary transformations, called Givens rotations, which selectively introduces a zero element into the matrix [5]. A Givens rotation consists of two consecutive operations over two rows of the matrix. First, the first element of both rows is used to compute the rotation angle which zeroes one of them. Then, all elements of the two rows are accordingly rotated in pairs.

The Givens method zeroes the lower elements of the matrix, starting from the left column to the right one, and, on each column, from the bottommost element to the diagonal element. The upper triangular matrix R is obtained by accumulating the Givens rotations on the initial matrix. Similarly, Q is obtained when the same rotations are applied to the identity matrix.

B. CORDIC Algorithm and QRD Architecture

The CORDIC (COordinate Rotation DIgital Computer) is an iterative algorithm which calculates transcendental functions with a very simple hardware [6]. CORDIC circuits only utilize addition and shifting and may operate in vectoring or rotation mode. The former rotates a vector until one of its coordinates reaches zero. The latter rotates a vector with a determined angle. Therefore, a CORDIC unit could be used to compute the two operations of a Givens rotation.

Several hardware approaches based on CORDIC has been proposed to solve QRD in the literature. They go from serial to fully parallel 2D systolic architectures. Besides the CORDIC itself, only multipliers are sometimes used to compensate the scale factor introduced by CORDIC. Our study is based on the 2D systolic array architecture [7], but the same ideas are also applicable to the other ones.

III. MODIFICATIONS FOR THE NEW REPRESENTATION

To operate using the new representation system presented in [4], some modifications are required in the basic data-path of the CORDIC algorithm. First we show the main characteristics of this representation system and then, the new CORDIC architecture based on this format.

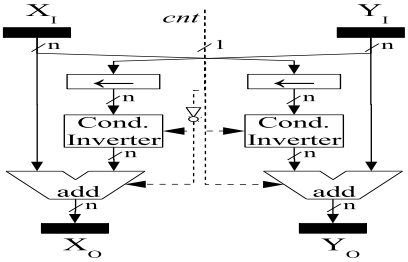

Fig. 1. Classic CORDIC circuit for one iteration.

This work was supported in part by the Ministry of Education and Science of Spain and Junta of Andalucía under contracts TIN2013-42253-P and TIC-1692, respectively, and Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

A. Half-Unit Biased Formats

The new representation system, called Half-Unit Biased (HUB), is based on shifting the values exactly represented under conventional formats by half of the weight of the Least Significant Bit (LSB). Thus, when a value is truncated to produce a HUB number, said HUB number is always the nearest to the initial value. In practice, the HUB numbers are like conventional ones but they have an implicit LSB, which always equals one. This hidden LSB does not have to be stored or transmitted. It only has to be taken into account when an operation is performed with that number. Thus, the new format has the same number of explicit bits and precision as a conventional one.

A side effect is that the two's complement of a HUB number is implemented simply by inverting all explicit bits (one's complement). Since the hidden LSB is always zero after the inversion, the final addition sets this hidden bit to one again, and no carry is propagated to the explicit bits.

B. CORDIC Architecture for HUB Numbers

The new CORDIC architecture is designed by considering the hidden LSB and it is shown in Fig. 2. The only actual change is that the carry input of the adders is not connected to the subtraction control signal (cnt). That input is connected to the Most Significant Bit (MSB) of the discarded bits after shifting (Px and Py). Since this bit has to be added to the hidden bit of the other coordinate, which is one, the carry output of this operation equals said MSB. In the first stage, this bit is one, since it corresponds to the hidden bit.

IV. Implementation Results

The two different approaches to perform QRD have been implemented on FPGA for 4x4 matrices using the pipeline architecture presented in [7]. First, using both circuits, the QR decomposition has been calculated for 50,000 random matrices whose results have been checked by computing the inverse QRD. Table I shows the statistical parameters of these errors. It is seen that the proposed approach practically halves the rounding errors for the tested word-lengths.

TABLE I. Rounding Error Parameters For QRD

Bit width	Conventional QRD		Proposed QRD	
	max	***mean***	***Max***	***mean***
16	692.829e-6	362.938-6	265.896e-6	153.388e-6
32	16.627e-9	7.542e-09	7.5785e-9	3.2057e-9

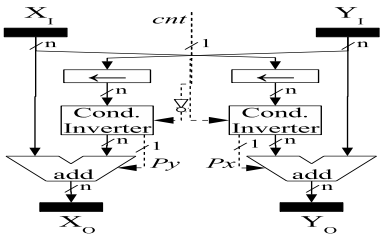

Fig. 2. Proposed CORDIC circuit for one iteration.

TABLE II. FPGA Implementation Results

	Slice Reg.	***Slice LUTs***	***Max. Freq.***	***Max. Error***
New-15bits	2,576	2,666	568 Mhz	687.6e-6
Classic-16 bits	2,810	2,960	421 Mhz	692.8e-6
New-31 bits	10,900	11,188	492 Mhz	15.33e-9
Classic-32 bits	11,240	11,792	378 Mhz	16.63e-9

Therefore, given a desired error bound, the QRD architecture using HUB numbers could use one bit less than the classic approach. Taking this into account, the implementation parameters of both architectures, has been compared targeting the same error bounds and device. The main results are provided in Table II, except the number of multipliers which is the same in both approaches. We should remember that although the proposed designs are 15 and 31 bit-widths, their rounding error bounds are equivalent to the ones of the conventional 16 and 32-bit architectures, respectively.

These implementation results show that the proposed architecture simultaneously improves area and delay by a significant amount. Specifically, the number of LUTs is reduced about a 10% and 5% for 16 and 32 bits, respectively, and a little less for the number of registers. Similarly, the maximum clock frequency is augmented practically by a 35% and 30% for 16 and 32 bits, respectively.

V. Conclusions

In this paper the use of a HUB format to optimize the fixed-point implementation of QRD is proposed. The improvement of the rounding errors allows reducing the word-length for the proposed approach. The simplification of the two's complement computation along with this bit-width reduction produces a FPGA architecture which simultaneously has up to 10% less area and up to 35% faster. A few patent applications have been filed for different issues regarding to the circuits to operate under the new format.

References

[1] Hun-Hee Lee; Myung-Sun Baek; Jee-Hoon Kim; Hyoung-Kyu Song, "Efficient detection scheme in MIMO-OFDM for high speed wireless home network system," Consumer Electronics, IEEE Transactions on, vol.55, no.2, pp.507,512, May 2009.

[2] Peng Xue; Kitaek Bae; Kyeongyeon Kim; Ho Yang, "Progressive equalizer matrix calculation using QR decomposition in MIMO-OFDM systems," Consumer Communications and Networking Conference (CCNC), 2013 IEEE , vol., no., pp.801,804, 11-14 Jan. 2013.

[3] Q. Su, Y. Niu, G. Wang, S. Jia, and J. Yue, "Color image blind watermarking scheme based on QR decomposition," Signal Processing, vol. 94, no. 0, pp. 219 – 235, 2014.

[4] J. Hormigo and J. Villalba, "Optimizing DSP circuits by a new family of arithmetic operators," in Signals, Systems and Computers, 2014 Asilomar Conference on, pp. 871–875, Nov 2014.

[5] G. H. Golub and C. F. Van Loan, Matrix Computations (3rd Ed.), Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[6] M. D. Ercegovac and T. Lang, Digital arithmetic. Elsevier, 2003.

[7] S. D. Muñoz and J. Hormigo, "High-Throughput FPGA Implementation of QR Decomposition", Circuits and Systems-II, IEEE Transactions on, in press.