2019 International Siberian Conference on Control and Communications (SIBCON)

The Pipelined QR Decomposition Hardware Architecture Based On Givens Rotation CORDIC Algorithm

A.V. Sokolovskiy
Scientific and Educational Laboratory "Systems of Navigation, Control and Communication"
Siberian Federal University
Krasnoyarsk, Russia
sokolovskii_a@mail.ru

V.N. Tyapkin
Scientific and Educational Laboratory "Systems of Navigation, Control and Communication"
Siberian Federal University
Krasnoyarsk, Russia
tyapkin58@mail.ru

E.A. Veisov
Scientific and Educational Laboratory "Systems of Navigation, Control and Communication"
Siberian Federal University
Krasnoyarsk, Russia
eveisov@sfu-kras.ru

Yu.L. Fateev
Scientific and Educational Laboratory "Systems of Navigation, Control and Communication"
Siberian Federal University
Krasnoyarsk, Russia
fateev_yury@inbox.ru

***Abstract*— QR decomposition plays a huge role in the adaptive filtering, control systems and a computation modeling of the physical processes. This paper concerns the issue of a QR decomposition hardware implementation features based on Givens rotation technique. For speed-up of the computation purposes used a pipelined architecture and CORDIC algorithm. The hardware costs and speed of a computation is evaluated for a different adders architectures in the CORDIC algorithm. The Givens rotation technique is used because it have a simple architecture and optimized hardware implementation.**

***Keywords*— hardware algorithm, QR decomposition, CORDIC, pipelined architecture, Givens rotation, FPGA, adders architecture.**

I. Introduction

QR decomposition is the linear equation solving technique and it's having a huge role in a such issues as a adaptive signal filtering, control systems, robotics and computation modeling of a physical processes. The Givens rotation technique is used for hardware implementation of the QR decomposition because it's having an effective design based on CORDIC algorithm.

The Givens rotation technique based on upper-triangular matrix computation by a sequential rotating an adjacent components in a rows of the input matrix.

The CORDIC algorithm use an simple add/subtract operations with a shift register to rotate input vector. It's widely used in a digital signal processing systems that constrained by a low available dimensions for embedding, especially in a multiple-input and multiple-output (MIMO) systems [7]. In other hand, the pipelined architecture gives an opportunity to use this algorithm in a high-speed digital communication systems.

Nevertheless, that by default the CORDIC algorithm was proposed to compute a trigonometric functions besides implementing a multiplier/divider operator it's allowing to compute a several complex functions by applying the appropriate pre-processing to input parameters, configuring compute mode and post-processing to results. Pre-processing and post-processing to a matrix components also implemented based on CORDIC algorithm. This allows to simplify the design process.

II. QR Decomposition by Givens Rotation Technique

Any non-singular matrix *A*, sized of *M*x*N* is allowing to decompose to a multiplication of the orthogonal matrix *Q* to the upper-triangular matrix *R*. Given a matrix *A* (1).

$$A = QR = \begin{pmatrix} Q_{11} & \cdots & Q_{1k} \\ \vdots & \ddots & \vdots \\ Q_{m1} & \cdots & Q_{mk} \end{pmatrix} \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{kn} \end{pmatrix}. \qquad (1)$$

In this case, matrix *Q* is allowing equation (2)

$$Q^T Q = I. \qquad (2)$$

Thus, taking into account the expression (1) and expression (2) the matrix *R* is calculating by the equation (3).

$$R = Q^T A = \begin{pmatrix} q_{11} & \cdots & q_{1k} \\ \vdots & \ddots & \vdots \\ q_{m1} & \cdots & q_{mk} \end{pmatrix}^T \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kn} \end{pmatrix}. \qquad (3)$$

In the issues [1, 2] described a technique to calculate the upper-triangular matrix based on Givens rotation matrix *Q* (4, 5).

$$Q = Q_1 Q_2 \ldots Q_k = 1, \qquad (4)$$

$$Q_k = \begin{pmatrix} c_{mn,k} & s_{mn,k} & \cdots & 0 \\ -s_{mn,k} & c_{mn,k} & & \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & 1 \end{pmatrix}, \qquad (5)$$

where

$$c_{mn} = \frac{a_{mn}}{\sqrt{a_{mn}^2 + a_{(m+1)n}^2}}, \quad s_{mn} = \frac{a_{(m+1)n}}{\sqrt{a_{mn}^2 + a_{(m+1)n}^2}} .$$

After than the equation (3) is calculated, the matrix *R* can be expressed by the equation (6).

$$R = \begin{pmatrix} \sqrt{a_{11}^2 + a_{21}^2} & X & \cdots & X \\ 0 & \sqrt{a_{22}^2 + a_{32}^2} & & \\ \vdots & & \ddots & \vdots \\ 0 \quad 0 & & \cdots & a_{mn} \end{pmatrix}, \qquad (6)$$

where *X* – is a member of the upper-triangular matrix set at above a main diagonal.

The calculation costs of an member values iterative calculation of the upper-triangular matrix by taking into account the expression (5) primary relates to a multiply operator and a square root operator calculations. The QR decompose design throughput based on an iterative calculation technique is a matrix size dependent. This approach is not acceptable in the multi-input and multi-output (MIMO) high-speed systems.

In this paper proposed the calculation blocks pipelined architecture based on the CORDIC algorithm.

III. Hardware Architecture Based on the CORDIC Algorithm

The CORDIC algorithm is widely used thanks to its efficiency and a simple implementation capabilities for a trigonometric, a hyperbolic and other complex functions calculation based on add/subtract operation and shift registers. It is a valuable reason for those digital signal processing systems that's a multi-channel and a high-speed of processing with a limited hardware and dimension resources is a crucial.

The generalized CORDIC algorithm is described by an equation (7).

$$\begin{aligned} x^{(i+1)} &= x^{(i)} - \mu d_i y^{(i)} 2^{-i}, \\ y^{(i+1)} &= y^{(i)} + d_i x^{(i)} 2^{-i}, \quad , \\ z^{(i+1)} &= z^{(i)} - d_i e^{(i)}. \end{aligned} \qquad (7)$$

Mode of the CORDIC algorithm is determined by the μ parameter and by a selecting of $e^{(i)}$ function accordingly to an equation (8).

$$\begin{aligned} &\mu = 1,\ e^{(i)} = \tan^{-1} 2^{-i} && (\textit{Circular Rotation}) \\ &\mu = 0,\ e^{(i)} = 2^{-i} && (\textit{Linear Rotation}) \\ &\mu = -1,\ e^{(i)} = \tanh^{-1} 2^{-i} && (\textit{Hyperbolic Rotation}) \end{aligned} \qquad (8)$$

Since the CORDIC algorithm actually implements a pseudo-rotation instead of a really rotation according to the chosen function, then a result is stretched by the factor $K \approx 1{,}646760258121$. Thus a pre-processing and a post-processing must be applied to the input and output data respectively.

The Givens matrix calculation for an every matrix *A* member is implements according to the scheme in the Fig. 1.

On the scheme 1 depicted that CORDIC algorithms is used in two modes sequentially [3]. In a step one the input data pre-processing is implements in the rotation mode with a configuration (9).

$$\mu = 0,\ d_i = \mathrm{sign}(z^{(i)}),\ e^{(i)} = 2^{-i}. \qquad (9)$$

For align an matrix rows amplitude ratios in the upper-triangular matrix the pre-processing is used for scaling a matrix *A* members by the using a scaling diagonal matrix *S* (10).

$$S = \begin{pmatrix} K^{-M} & 0 & & \cdots & 0 \\ 0 & K^{-M} & & & \\ \vdots & & K^{-(M-1)} & 0 & \vdots \\ & & 0 & \ddots & \\ 0 & & \cdots & & K^{-1} \end{pmatrix}, \qquad (10)$$

This operation excludes an iterated rotate calculation results delay for each matrix *A* rows.

In a step two calculation of the matrix *R* (11) is implements in vectoring mode (12).

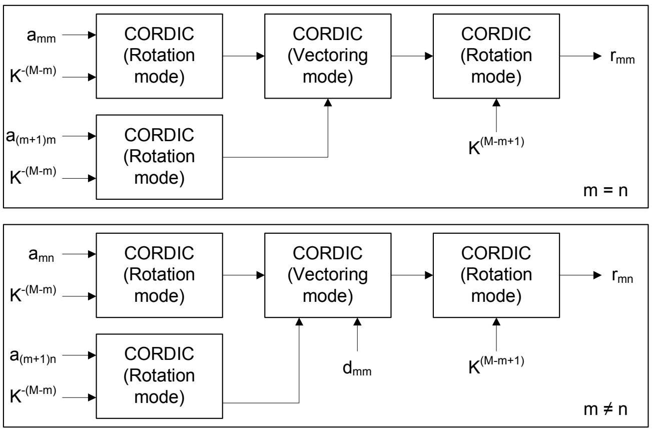

Fig. 1. The Givens rotation structural scheme.

$$R = \begin{pmatrix} K^{-(M-1)}\sqrt{a_{11}^2 + a_{21}^2} & X & \cdots & X \\ 0 & \cdots & & \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & a_{mm} \end{pmatrix}, \qquad (11)$$

$$\mu = 1,\ d_i = -\mathrm{sign}(x^{(i)}y^{(i)}),\ e^{(i)} = \tan^{-1}2^{-i}. \qquad (12)$$

The calculations is ending by the post-processing in the rotation mode with a configuration (9). The post-processing also used for scaling a R (11) matrix members by using a diagonal matrix S' (13), then matrix (6) is a result of this operation.

$$S' = \begin{pmatrix} K^{(M-1)} & 0 & \cdots & 0 \\ 0 & K^{(M-2)} & & \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & 1 \end{pmatrix}, \qquad (13)$$

where K – the stretching coefficient, m – the row number, that a diagonal matrix R member is located.

The Givens rotation compute pipeline (Fig. 2) for each a matrix A row consist of a delay lines (Delay), a pre-processing block (PreProc), a Givens rotation processing block (Proc) and a post-processing block (PostProc).

IV. Comparison of the QR Decomposition Hardware Costs with Different Adders Architecture

There are many hardware architectures of the adders, that are different from each other by a compute speed and a hardware costs.

The simplest in implementation is a ripple-carry adder architecture, with a sequential bit summing of a binary numbers [3]. The base compute cell of this architecture is a full adder (FA). The full adder consist of input signals – a carry in signal, a two terms signals and a output signals – a carry-out signal, a sum signals. The delay of a sum calculation for this architecture is relates with a composition of bit sequences. The maximum computation delay is relates with a length of terms, because carry-out signal must be evaluated on the each compute iteration.

The most compute speed effective architecture is a carry look-ahead architecture. In this architecture the carry-out bit propagation and generation for each bit position is evaluated in the special compute unit. Based on values of propagation and generation bits the pre-computing may be done for several group of bits, that is proportional reduces the sum computing time. The sum computing delay also relates with a composition of bit sequences. But in difference with a ripple-carry sum architecture for that architecture the maximum computation speed is relates with a length of terms and with length of a carry look-ahead computation block.

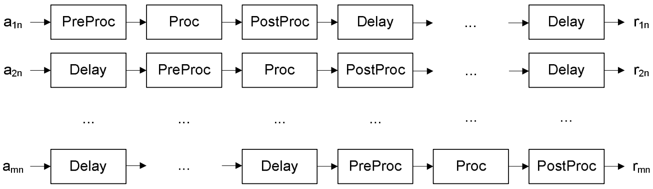

Fig. 2. The Givens rotation compute pipeline.

Another one of the most effective architecture is based on a sum of terms pre-compute for two different input carry bits. The true result of the sum computation is selected depending on a result computed carry bit. The main difference of this architecture from a carry look-ahead architecture is a group of bits sum computation is implemented base on a sequential sum computation of an each bit terms. The computation delay relates with a composition of bit terms. The maximum computation delay relates with a length of terms and with a carry look-ahead block length.

Another kind of an adder architectures relates with a probability of a bit-group occurrence with a generated or propagated bit group. The computation for this group may be dropped. This architecture includes capabilities of a carry look-ahead architecture and a carry-select architecture. The delay for this architecture relates with a length and a composition of bit sequences.

For minimizing a CORDIC algorithm computation delay is used a computing pipeline. As a result the computation delay of an algorithms becomes as computation delay of a one adder component with selected architecture.

The characteristics of the above mentioned architectures is presented in a table 1.

The necessary hardware resources for an implementation for a matrix with size 4×4 pipelined QR decomposition is presented in a table 2.

TABLE I. THE ADDITION COMPUTATION DELAY FOR 14-BIT LENGTH TERMS.

Architecture	Min. delay, clock cycles	Max. delay, clock cycles
Ripple-Carry adder	3	15
Carry-Lookahead adder	2	4
Carry-Select adder	4	5
Carry-Skip adder	4	15

TABLE II. NECESSARY HARDWARE RESOURCES PROCESSING BLOCKS

Based on the following architecture	Flip-flops	Processes
Ripple-Carry adder	1420335	232875
Carry-Lookahead adder	1666170	333855
Carry-Select adder	1843695	403650
Carry-Skip adder	1352025	302805

The main advantage of the hardware implementation of a pipelined QR decomposition architecture may be noted a high compute speed, that allow to use QR decomposition in a high speed telecommunication MIMO systems. The disadvantage of the above mentioned implementation in a fixed-point terms is a decreasing computation accuracy after a scaling pre-processing step, that is proportional relates with stretching coefficient K of the CORDIC algorithm.

Choosing an more effective solution for a implementation algorithm taking into account the hardware features of the implementation device is allowing to increase bandwidth of the digital signal processing device based on QR decomposition. The fastest QR decomposition may be implemented based on the carry look-ahead adders. At the same time the using of a hardware DSP-slices allow to valuable increase speed of QR decomposition and to optimize compute architecture, taking into account a hardware capabilities.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education and Science of the Russian Federation in the framework of the Federal target program «Research and development on priority directions of development of the scientific-technological complex of Russia for 2014-2020» (agreement № 14.578.21.0247, unique ID project RFMEFI57817X0247).

REFERENCES

[1] V.I. Zhigan, “Adaptive signal filtering: theory and algorithms,” Moscow: Technosphere, 2013.

[2] Semih Aslan, Sufeng Niu, Jafar Saniie, “FPGA implementation of fast QR decomposition based on Givens rotation,” IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 2012, pp. 470-473. doi: 10.1109/MWSCAS.2012.6292059

[3] Behrooz Parhami, “Computer arithmetic: algorithms and hardware designs,” NY, Oxford: Oxford University Press, 2000.

[4] A.V. Sokolovskiy, A.B. Gladyshev, D.D. Dmitriev, and V.N. Ratushniak, “Hardware diagram computing devices navigation equipment consumers SRNS,” 2017 Dynamics of Systems, Mechanisms and Machines (Dynamics), 2017, pp 1-4. doi: 10.1109/Dynamics.2017.8239510

[5] 1076 IEEE Standard VHDL Langeage Reference Manual, 2002, p. 300.

[6] Xilinx UG1046, UlatraFast embedded Design Methodology Guide, 2015, p 231.

[7] Hyukyeon Lee, Kyungmook Oh, Minjeong Cho, Yunseok Jang, and Jaeseok Kim, “Efficient low-latency implementation of CORDIC-based sorted QR decomposition for multi-Gbps MIMO systems,” IEEE transactions on circuits and systems – II: express briefs, vol. 65, no. 10, 2018, pp. 1375-1379. doi: 10.1109/TCSII.2018.2853099