УДК 519.6

Hardware Architectures of the QR-Decomposition Based on a Givens Rotation Technique

Alexey V. Sokolovskiy*
Evgeny A. Veisov†
Valery N. Tyapkin‡
Dmitry D. Dmitriev§
Siberian Federal University
Svobodny, 79, Krasnoyarsk, 660041
Russia

Received 18.03.2019, received in revised form 11.05.2019, accepted 20.07.2019

The fixed-point hardware architecture of the QR decomposition is constrained by a several issues that leads to decrease of a compute accuracy depending on a matrix size. In this article described the hardware architectures based on CORDIC algorithm and approximation functions. As a basis technique is used a Givens rotation technique, because it is a most suitable technique for hardware implementation.

Keywords: QR-decomposition, FPGA, CORDIC algorithm, approximation function, fixed-point arithmetic.
DOI: 10.17516/1997-1397-2019-12-5-606-613.

QR decomposition (QRD) is a widely used in a telecommunication systems as a pre-processing algorithm to advance a characteristics of a processed signals. Known several techniques to implement QR decomposition, as a Gram–Schmidt orthogonalization, a Householder orthogonalization and a Givens rotation technique. Usually a Givens rotation technique is used for hardware implementation, because it may be effectively designed based on a CORDIC algorithm and others effective computing architectures based on a approximation functions. The Givens rotation technique uses an iterated rotation operations of an adjacent row elements of an input matrix to get an upper-triangular matrix. The performance of an implemented hardware architecture may be evaluated in a computing speed and a hardware cost field of views. To improve a computing speed of the hardware designs may be used a pipelined or parallel architectures [1,2]. A pipelined architectures as well as a parallel architectures leads to increase a hardware costs, but gives an ability to improve computation speed up to 100 MHz. In other hand to decrease requirements to the memory usage may be used coding techniques for an input data or an pre-computed data such as an approximation function coefficients [3]. Moreover the QR decomposition computation architecture based on a fixed-point arithmetic is may be evaluated in a computation accuracy field of view [4]. Usually to increase an computation accuracy of the QR decomposition by using a Givens rotation technique the input matrix sorting is used. The sorting process has an aim to maximize a norm of the adjacent row elements, as a result an rotation angle computation accuracy is increased. In the same time adding a sorting process leads to an adding computation delay that is proportional an input matrix size.

*sokolovskii_a@mail.ru
†eveisov@sfu-kras.ru
‡tyapkin58@mail.ru
§dmitriev121074@mail.ru

1. The QRD-based signal processing

In general the multichannel signal processing model may be exposed as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,m} \\ h_{2,1} & h_{2,2} & \dots & h_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{k,1} & h_{k,2} & \dots & h_{k,m} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_k \end{bmatrix}. \tag{1}$$

where Y is the processing result vector, S is the received digital signal vector, H is the transformation matrix, N is the transformation error vector, m is received digital signal index, k is processing channel index

For QRD-based signal processing algorithms the matrix H may be exposed in form

$$\begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,m} \\ h_{2,1} & h_{2,2} & \dots & h_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{k,1} & h_{k,2} & \dots & h_{k,m} \end{bmatrix} = \begin{bmatrix} q_{1,1} & q_{1,2} & \dots & q_{1,l} \\ q_{2,1} & q_{2,2} & \dots & q_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ q_{k,1} & q_{k,2} & \dots & q_{k,l} \end{bmatrix} \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,m} \\ 0 & r_{2,2} & \dots & r_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{l,m} \end{bmatrix}, \tag{2}$$

where Q is the ortogonal matrix, R is the upper-triangular matrix

As an useful example of the QRD-based signal processing algorithm is a direct matrix inversion [4], since a Q matrix is a orthogonal, the inverse matrix of R simply compute as

$$R^{-1}_{i,j} = \begin{cases} -\left(\sum_{k=1}^{j-1} R^{-1}_{i,k} r_{k,j}\right) / r_{j,j} & i < j, \\ 1/r_{j,j} & i = j, \\ 0 & i > j. \end{cases} \tag{3}$$

Matrix inversion in the form (3) may be more effective to hardware implement.

2. The hardware architectures of the QR decomposition

2.1. The QRD hardware architecture based on the Q, R matrices direct computing

The direct compute QR-decomposition by the Givens rotation techinique may be expressed as

$$\begin{bmatrix} r_{k-1,m} \\ r_{k,m} \end{bmatrix} = \begin{bmatrix} \Re(|R_{k-1,m}|e^{j(\alpha_m - \alpha_n)}) \\ \Im(|R_{k-1,m}|e^{j(\alpha_m - \alpha_n)}) \end{bmatrix}, \quad \begin{array}{c} m > n, |R_{k-1,m}| = \sqrt{h^2_{k-1,m} + h^2_{k,m}}, \\ \alpha_n = \arctan\left(\frac{h_{n+1,n}}{h_{n,n}}\right), \ \alpha_m = \arctan\left(\frac{h_{k,m}}{h_{k-1,m}}\right), \end{array} \tag{4}$$

$$\begin{bmatrix} q_{k-1,m} \\ q_{k,m} \end{bmatrix} = \begin{bmatrix} \Re(|Q_{k-1,m}|e^{j(\beta_m - \alpha_n)}) \\ \Im(|Q_{k-1,m}|e^{j(\beta_m - \alpha_n)}) \end{bmatrix}, \quad \begin{array}{c} m > n, |Q_{k-1,m}| = \sqrt{h^2_{k-1,m} + h^2_{k,m}}, \\ \alpha_n = \arctan\left(\frac{h_{n+1,n}}{h_{n,n}}\right), \beta_m = \arctan\left(\frac{q_{k,m}}{q_{k-1,m}}\right), \end{array} \tag{5}$$

where n is the diagonal element index of the H matrix, k is the current processing channel, m is the current received signal index, r is the term of the R matrix, q is the term of the Q matrix.

2.2. The QRD hardware architecture based on the CORDIC algorithm

The QR decomposition expressed in (4), (5) may be implemented using the Givens rotation matrix

$$G_n = \begin{bmatrix} c & s & \dots & 0 \\ -s & c & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad c = \cos(\alpha_n), \quad s = \sin(\alpha_n). \tag{6}$$

As an equation (6) is the rotation matrix, then is more suitable in hardware costs point of view to implement based on the CORDIC algorithm. The generalized CORDIC algorithm [5] is expressed as

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

To implement equations (4), (5) the CORDIC algorithm (7) is configured in a vectoring mode (8), wherein the angles computing is not required, because in back substitution may be use the same angles as is required for rotation a diagonal terms and adjacent column terms with it of H matrix.

$$\mu = 1, \; d_i = -sign(x^{(i)} y^{(i)}), \; e^{(i)} = \arctan\left(2^{-i}\right). \tag{8}$$

Then taking into account an equations (7), (8) the Givens rotation matrix (6) is expressed as pseudorotation matrix

$$G_n^{(i)} = \begin{bmatrix} 1 & sign\left(x^{(i)} y^{(i)}\right) 2^{-i} & \dots & 0 \\ -sign\left(x^{(i)} y^{(i)}\right) 2^{-i} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \tag{9}$$

When calculating the Givens rotation matrix according to equation (9) the results vectors is stretched [5] with factor $K = 1.646760258121$, to compensate this for an each adjacent row terms rotation iteration a multiplication it on the factor inversed to stretch factor $K^{-1} = 0.6072529350089$ is required.

2.3. The QRD hardware architecture based on the approximation polynomials

Another hardware architecture of the Givens rotation matrix (6) may be implemented based on the approximation polynomials for the trigonometric functions [5]

$$\sin(x) = x - \frac{1}{3!}x + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \ldots + (-1)^i \frac{1}{(2i+1)!}x^{2i+1} + \ldots, \tag{10}$$

$$\cos(x) = 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + \ldots + (-1)^i \frac{1}{(2i)!}x^{2i} + \ldots, \tag{11}$$

$$\arctan(x) = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \ldots + (-1)^i \frac{1}{2i+1}x^{2i+1} + \ldots, \; -1 < x < 1, \tag{12}$$

$$\sqrt{(}x) = 1 - \frac{1}{2}y - \frac{1}{2*4}y^2 - \frac{1*3}{2*4*6}y^3 - \ldots - \frac{1*3*5*\ldots*(2i-3)}{2*4*6*\ldots*2i}y^i - \ldots, \; y = 1 - x. \tag{13}$$

For more effective calculation of the equations (10)–(13) the functions argument x is divided by two terms according to

$$x = x_H + 2^{-i}x_L, \quad \begin{cases} 0 \leqslant x_H \leqslant 4 \\ t + 2(bits) \end{cases}, \begin{cases} 0 \leqslant x_L < 1 \\ l - t(bits) \end{cases}, \tag{14}$$

where x_L is the LSBs of the function argument, x_H is the MSBs of the function argument, t is the LSB shift coefficient, l is the function argument bit length.

Then the functions (10)–(13) may be expressed by the Taylor sequence

$$f(x) = \sum_{j=0}^{\infty} f^{(j)}(x_H) \frac{(2^{-t}x_L)^j}{j!}, \tag{15}$$

where $f^{(j)}$ is the j-nd order derivatives and $f^{(0)}(x_H) = f(x_H)$.

Since an equation (15) is rapidly decreasing with a rising j (16), an acceptable compute accuracy may be done based on linear approximation (17)

$$\frac{2^{-jt}}{j!} \approx 0, j \to \infty, \tag{16}$$

$$f(x) = f(x_H) + 2^{-t}x_L f'(x_H). \tag{17}$$

After Givens rotation matrix computation (6) or (9) is done the Q and R matrices is computed according to

$$\begin{aligned} R &= G_1 G_2 \dots G_n H, \\ Q &= [G_1 G_2 \dots G_n]^T. \end{aligned} \tag{18}$$

3. The fixed-point hardware architectures

The hardware implementation of the Givens rotation matrix computing (6), (9) based on the approximation polynomials (15), (17) may be done according to scheme in below (Fig. 1).

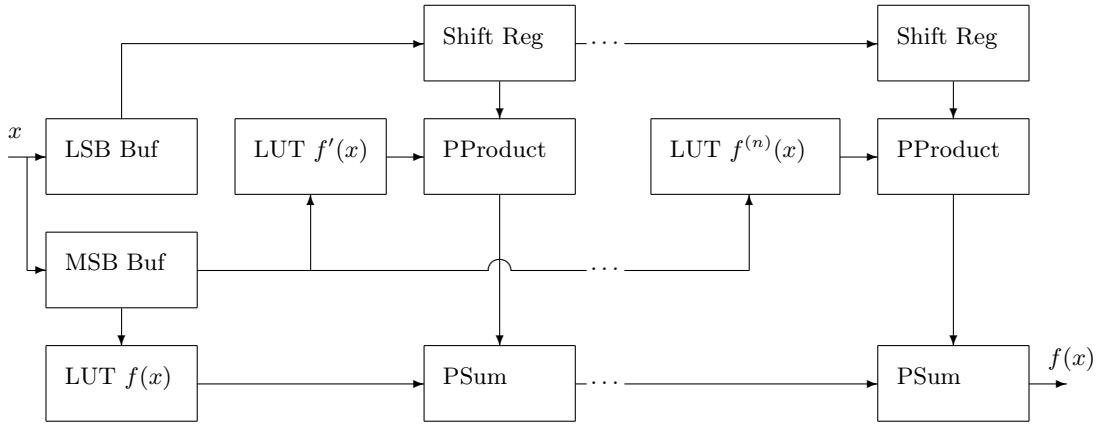

Fig. 1. The structural scheme of the approximation polynomials hardware architecture, where x is the function argument, LSB Buf is the LSB argument buffer, MSB Buf is the MSB argument buffer, LUT $f(x)$ is the function values lookup table, LUT $f'(x)$ is the function's 1st derivative lookup table, LUT $f^{(n)}(x)$ is the function's n-nd derivative lookup table, PProduct is the pipelined multiplier, PSum is the pipelined adder

To increase the speed of signal processing device based on this architecture using the pipelining techique and the hardware DSP-slices is required. This architecture also may be used to direct compute Q and R matrices according to equations (4), (5).

An another architecture based on CORDIC algorithm is exposed in scheme below (Fig. 2).

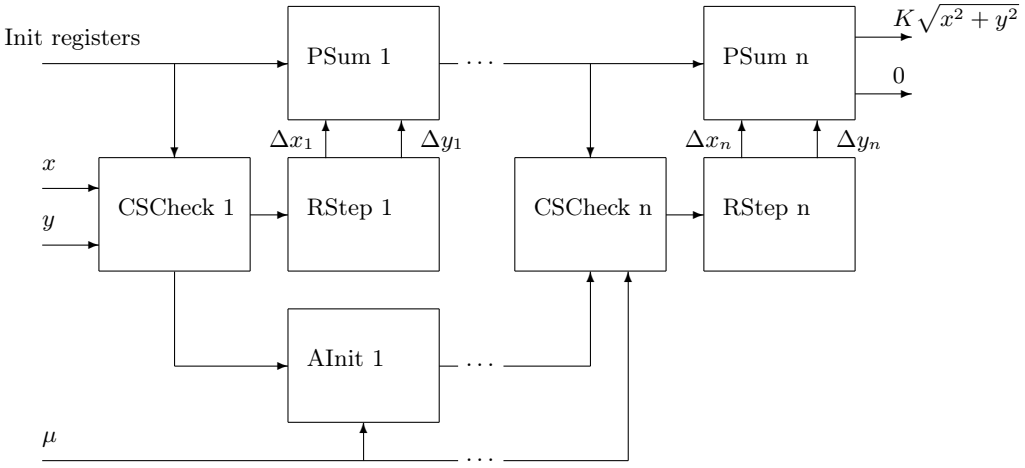

Fig. 2. The structural scheme of the CORDIC-based algorithm R matrix computation hardware architecture, where x and y are adjacent row terms of the H matrix that need to rotate, and y is the term that need to zeroing, CSCheck is the current sign checking unit, PSum is the pipelined adder, RStep is the rotation angle sampling unit, AInit is the current angle initialization unit, K is the CORDIC algorithm's stretching coefficient, n is the rotation sample index

This architecture doesn't require an FPGA DSP-slices and may be effectively implemented based on the pipelined adders and shift registers according to (7), (8).

4. The mean square error evaluation of the different QR decomposition fixed-point hardware architectures

To evaluate the mean square errors for several QR decomposition architectures the Q and R matrices is computed separately for matrix H then the product of Q and R matrices was taken to get as a result the matrix H in fixed-point representation. The mean square error is evaluated according to

$$MSE_H = \sqrt{\frac{\sum_{n=1}^{1000}\sum_{x=1,y=1}^{LL}\left(H_{x,y}^{fp} - H_{x,y}^{fx}\right)^2}{1000L^2}}, \tag{19}$$

where H^{fp} is the H matrix floating-point representation, H^{fx} is the H matrix fixed-point representation, L is the square matrix size, and x, y is the H matrix term indices (Figs. 3, 4, 5).

In the Fig. 4 there are nonlinearity of the mean square error H matrix fixed-point representation for 12-bit and 14-bit hardware arithmetic units based on approximating polynomials. This means that in a rising hardware arithmetic unit bit length the approximation polynomials (10)–(13) term count increasing is required.

The mean square error of the H matrix in a fixed-point representation is a smallest among other architectures.

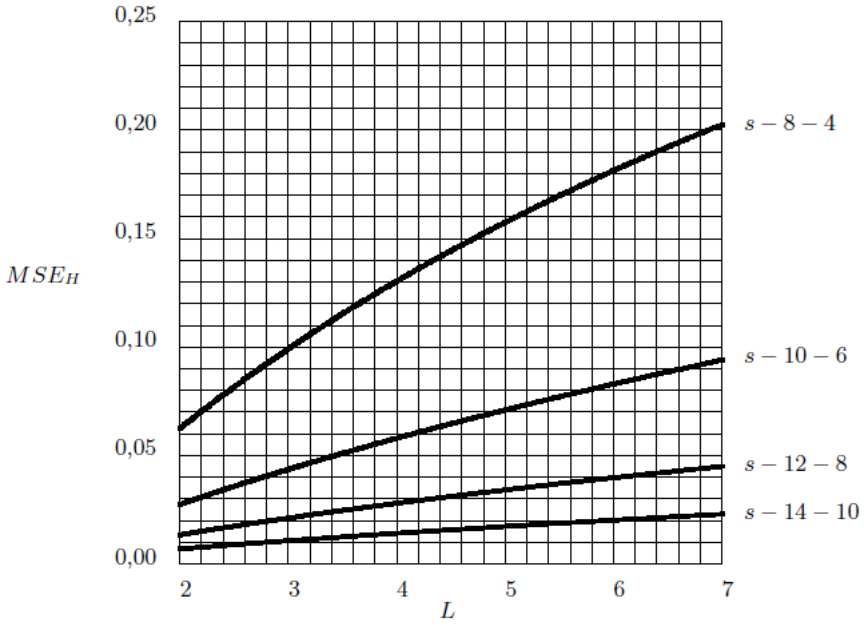

Fig. 3. The mean square error of the H matrix in a fixed-point representation taken based on the direct Q and R matrices computation, where $s-x-y$ is the fixed-point arithmetic uses a signed terms with x bits integer and y bits fractional parts

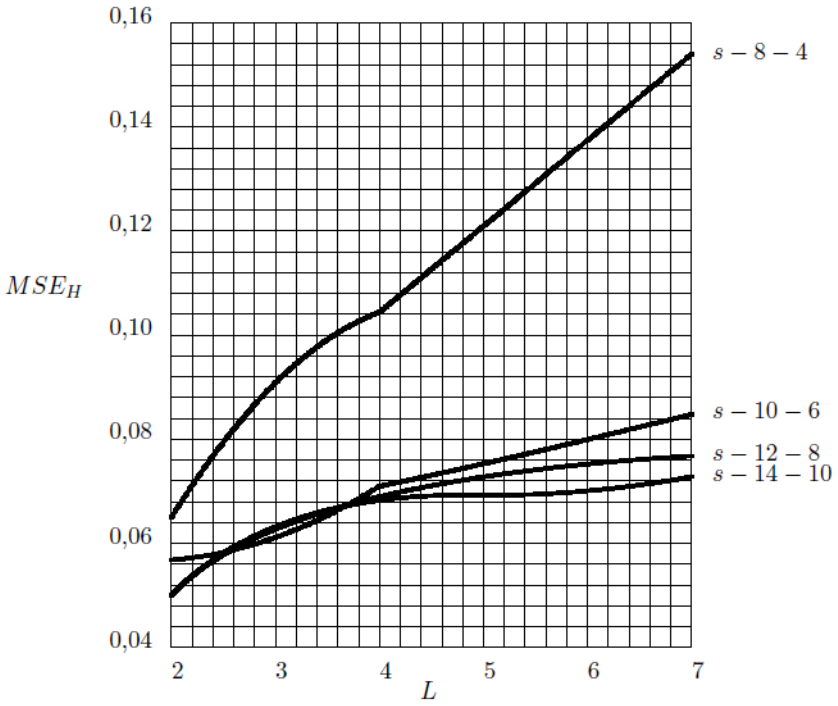

Fig. 4. The mean square error of the H matrix in a fixed-point representation taken based on the Q and R matrices computed by the approximating polynomials, where $s-x-y$ is the fixed-point arithmetic uses a signed terms with x bits integer and y bits fractional parts

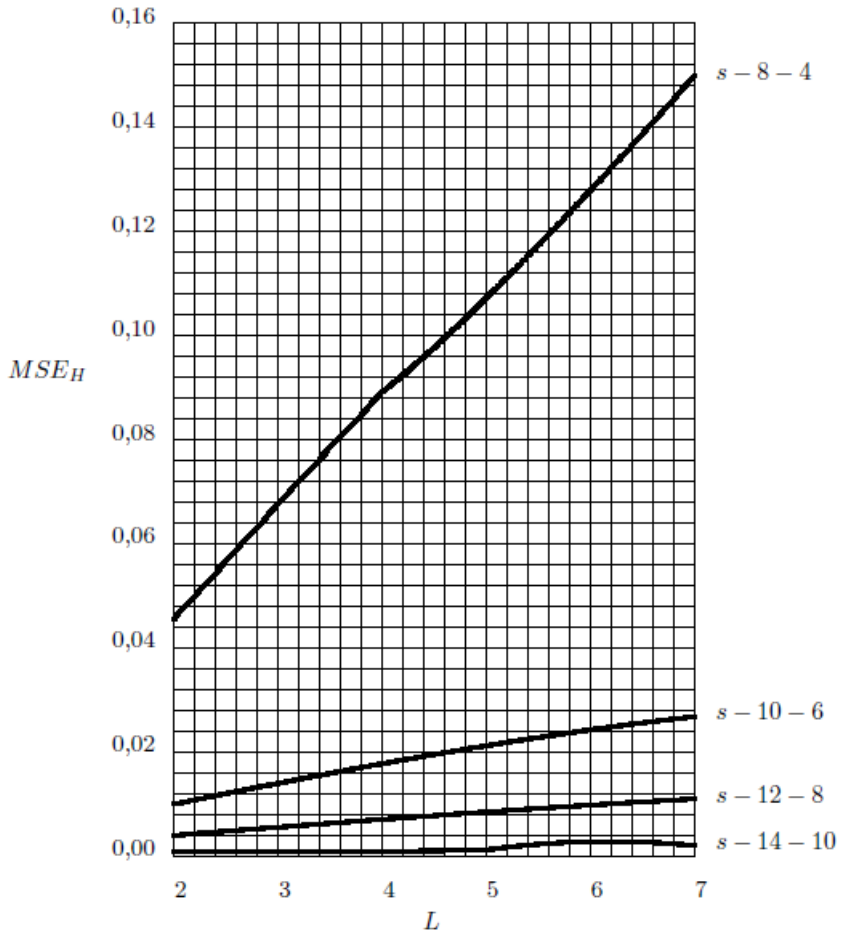

Fig. 5. The mean square error of the H matrix in a fixed-point representation taken based on the Q and R matrices computed by the CORDIC algorithm, where $s-x-y$ is the fixed-point arithmetic uses a signed terms with x bits integer and y bits fractional parts

As in view on graphs (Figs. 3, 4, 5) the largest mean square error is in hardware architecture based on the direct Q and R matrices computation.

The represented hardware architectures of the QR decomposition may be implemented using DSP-slices, or basic logic elements, pipelined adders and shift registers that in all gives an ability to take a mind the FPGA or SoC hardware features and constraints. For hardware devices with small amount of the DSP-slices that utilizes the QR decomposition recommends to use the hardware architecture based on the CORDIC algorithm. In other hand if high-speed DSP-slices is in hardware device then using direct Q and R matrices computation is more preferable for arithmetic units with 12-bit and wider terms. For terms that's smaller than 12-bit is recommends to implement hardware architecture based on the approximating polynomials.

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 no. 14.578.21.0247, unique ID project RFMEFI57817X0247).

References

[1] A.V.Sokolovskiy, A.B.Gladyshev, D.D.Dmitriev, V.N.Ratushniak, Hardware diagram computing devices navigation equipment consumers SRNS, Dynamics of Systems, Mechanisms and Machines (Dynamics), 2017, 1–4.

[2] H.Lee, H.Kim, M.Cao, J.Kim, Low-latency implementation of CORDIC-based sorted QR decomposition for high-speed MIMO-OFDM system, 28th International Conference Radioelektronika, 2018, 1–4.

[3] W.Fan, A.Alimohammad, Givens rotation-based QR decomposition for MIMO systems, *IET Communications*, **11**(2017), 1838–1845.

[4] D.Chen, M.Sima, Fixed-point CORDIC-based QR decomposition by Givens rotations on FPGA, Proceedings 2011 International Conference on Reconfigurable Computing and FPGAs, ReConFig 2011, 2011, 327–332.

[5] B.Parhami, Computer arithmetic: algorithms and hardware designs, 2000.

Аппаратная архитектура QR-разложения, основанного на методе вращения Гивенса

Алексей В. Соколовский
Евгений А. Вейсов
Валерий Н. Тяпкин
Дмитрий Д. Дмитриев
Сибирский федеральный университет
Свободный, 79, Красноярск, 660041
Россия

Аппаратная архитектура QR-разложения с фиксированной точкой ограничена некоторыми проблемами, которые приводят к потере точности вычислений в зависимости от размерности матрицы. В этой статье описаны аппаратные архитектуры на основе алгоритма CORDIC и функций аппроксимации. За основу взят метод вращения Гивенса, поскольку это наиболее эффективный метод для аппаратной реализации.

Ключевые слова: QR-разложение, ПЛИС, алгоритм CORDIC, функция аппроксимации, арифметика с фиксированной точкой.