QR-Decomposition Architecture Based on Two-Variable Numeric Function Approximation

Jochen Rust, Frank Ludwig and Steffen Paul
Institute of Electrodynamics and Microelectronics (ITEM.me)
University of Bremen, Bremen, Germany, +49(0)421/218-62538
Email: {rust, ludwig, steffen.paul}@me.uni-bremen.de

***Abstract*—This paper presents a new approach for hardware-based QR-decomposition using an efficient computation scheme of the Givens-Rotation. In detail, the angle of rotation and its application to the Givens-Matrix are processed in a direct, straightforward manner. High-performance signal processing is achieved by piecewise approximation of the arctangent and sine function. In order to identify appropriate function approximations, several designs with varying constraints are automatically generated and analyzed. Physical and logical synthesis is performed in a 130 nm CMOS-technology. The application of our proposal in a multi-antenna mobile communication scenario highlights our work to be very efficient in terms of calculation accuracy and computation performance.**

***Index Terms*—QR-decomposition, Givens-Rotation, VLSI, two-variable, numeric function approximation**

I. Introduction

Efficient computation of the QR-decomposition (QRD) is a mandatory task in many application areas. In the scope of mobile communication, it can be used in various ways [1]. In the scope of channel precoding, it has become increasingly important in recent years, as it enables an evident reduction of signal processing effort at the receivers side. Additionally, it is a practical solution for the realization of multi-user multiple-input-multiple-output (MIMO) communication [1]. In this paper a QRD algorithm is introduced that performs all necessary computation steps by using two-variable (also called two-dimensional (2D)) numeric function approximations (NFA). In general, NFA-based computation enables a notable simplification of signal processing effort by allowing minor (computational) errors [2]. In order to achieve high performance benefits, the resulting NFAs must only consist of a multiplexer tree and an adder-based accumulation unit. For a quick redesign, an automated NFA design generator is mandatory, as this enables the application-specific evaluation of appropriate NFAs by experimental means, taking different errors into account. Thus, an automated design method is used that satisfies these challenging constraints (see Sec. III).

II. Givens-Rotation-based QR decomposition

For an efficient realization of the QRD in hardware, Givens-Rotation-based computing is proposed in this paper, as its feasibility has been proven in many different references, recently [3], [4]. Generally speaking, the Givens-Rotation is a repeated application of the Givens-Matrix

$$\mathbf{G}(\varphi) = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \tag{1}$$

to a given matrix $\mathbf{H}$ in order to zero all elements below the diagonal as well as the imaginary parts of the diagonal. Hence, for QRD processing, the generation of φ (GR1) and its application to both $\mathbf{H}$ and the identity matrix $\mathbf{I}$ (GR2) leads to $\mathbf{R}$ and $\mathbf{Q}$. A detailed description can be found in [5]

Considering a direct computation scheme, φ is achievable using the arctangent function

$$f_1(y_1, y_2) = \mathrm{atan}(y_1, y_2) = \left(\frac{y_1}{y_2}\right) \ , \tag{2}$$

with y_1, y_2 as the elements of the affected rows. For GR2, the Givens-Matrix must be applied to the elements that remain in the same row

$$\begin{pmatrix} y'_n \\ y'_{n+1} \end{pmatrix} = \begin{pmatrix} y_n \sin(\frac{\pi}{2} + \varphi) + y_{n+1}\sin(\varphi) \\ y_{n+1}\sin(\frac{\pi}{2} + \varphi) - y_n \sin(\varphi) \end{pmatrix} \ , \tag{3}$$

with $\cos(\varphi) = \sin(\frac{\pi}{2} + \varphi)$. Hence, GR2 computation can be realized by calculating the expression $f_2(y_n, \varphi) = y_n \sin(\varphi)$.

III. Approximation of 2D numeric functions

The main idea of 2D NFAs for Givens-Rotation-based QRD refers to the approximation of a given function $f(x_1, x_2)$ by a bilinear equation

$$\tilde{f}(x_1, x_2) = \tilde{\beta}_0 + \tilde{\beta}_1 x_1 + \tilde{\beta}_2 x_2 \ , \tag{4}$$

where $\tilde{\beta}_0, \tilde{\beta}_1, \tilde{\beta}_2$ denote the linear coefficients, x_d the input variables (with the dimension index $d \in \{1, 2\}$) and $\tilde{f}(x_1, x_2)$ is the NFA. For the generation of 2D NFAs from a given function, we propose the use of non-uniform, piecewise and multiplier-less function approximation. A general overview of the proposed method is presented in the following.

In order to achieve high approximation quality, a hardware-efficient segmentation of the function is used. In detail, $f(x_1, x_2)$ is split up into several segments possessing an approximated sub-function each. The selection of a segment is performed by regarding the most significant bits (MSB) of the input operands. For an automated NFA generation, a corresponding recursive and error-dependent heuristic is used.

978-3-9815370-4-8/DATE15/©2015 EDAA

In detail, for each segment a maximum error is taken into account which is defined by

$$\varepsilon_{\max} = \max|f(n_1X_1, n_2X_2) - \tilde{f}(n_1X_1, n_2X_2)| \ , \quad (5)$$

where $x_d = n_dX_d$ denote the quantization effects of the digital data path. If ε_{max} is higher than a specified error threshold, the function is split up into four segments by bisection of the input ranges; each segment contains a corresponding sub-function. As this can be performed several times and independently, a non-uniform segmentation scheme is established (see Fig. 1). Otherwise, if ε_{max} is below the specified error value, the remaining segments are processed in a clockwise manner.

For the NFA generation inside a segment, the reference coefficients β_1', β_2' are determined by building the arithmetic mean of partial gradients. Due to the already mentioned digital quantization effects, this can be realized by calculating the one-dimensional derivation of the (sub-)function values inside a segment ($\frac{\partial}{\partial n_d}\left(f(n_1X_1, n_2X_2)\right)$). Next, the resulting values are summed up and divided by the overall number of partial gradients. By the application of this method in both dimensions the reference coefficients β_1', β_2' can be obtained.

In order to enable multiplier-less signal processing, the desired coefficients $\tilde{\beta}_d$ have to fulfill further restrictions. In detail, only a small set of partial products is permitted, its number must be specified in advance by the so-called quantization factor (QF) (see [6]). Hence, the desired bilinear coefficients are achieved by minimizing the gap between β_d' and $\tilde{\beta}_d$ taking the specified QF into account.

The coefficient $\tilde{\beta}_0$ is calculated by solving the equation

$$\tilde{\beta}_0 = \arg\min_{\beta'_0} |f(x_1, x_2) - \left(\tilde{\beta}_{1,q}x_1 + \tilde{\beta}_{2,q}x_2 + \beta'_0\right)| \ , \quad (6)$$

with β'_0 as variable of the binary search bisection method, where the range of all possible values is reduced by half, iteratively [7].

Taking all mentioned aspects into account, the proposed NFA technique can be expressed

$$\tilde{f}(\mathbf{x}) = \left[\left(\tilde{\boldsymbol{\beta}}_1 \quad \tilde{\boldsymbol{\beta}}_2\right)\mathbf{x} + \tilde{\boldsymbol{\beta}}_0\right]^T \cdot \boldsymbol{\kappa}(\mathbf{x}) \ , \quad (7)$$

with m as the overall number of segments, $\boldsymbol{\kappa}(\mathbf{x})$ as fade-out function selecting the active segment and $\mathbf{x} = (x_1 \ x_2)^T$. In this paper, only functions with an equal function size and range are considered $0 \leq x_d < 1 - 2^{-15}$.

In a last step, the extracted parameters are mapped to hardware modules. Thus, corresponding VHDL-template strings are used that must be filled with the previously extracted parameters: Input/output ports and constants are realized by a direct assignment of data-types, the partial products refer to simple operand shifts, the segmentation is realized by encapsulated if-statements in VHDL and the accumulation can be performed by a tree-adder.

For the calculation of Givens-Rotation-based QRD, this NFA generation scheme is applied to the two functions $f_1(y_1, y_2)$ and $f_2(y_n, \varphi)$ introduced in Sec. II (see Fig. 1). Due to their function properties, only positive input values are considered, as the impact of operand signs can be handled separately. In addition, there are function-specific simplifications that can be used for further signal processing improvement. Thus, a detailed explanation of both functions is given in the following.

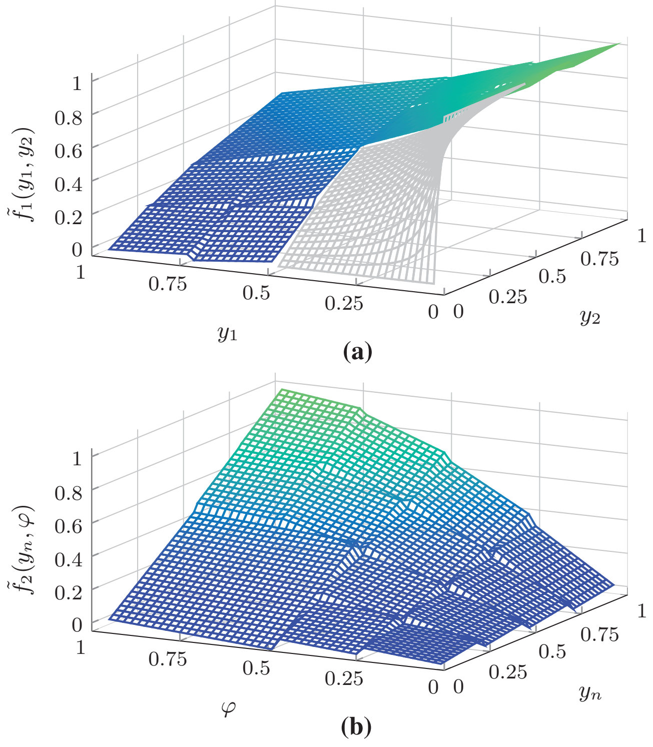

Fig. 1. Example of the (scaled) **(a)** arctangent $\tilde{f}_1(y_1, y_2) = \mathrm{atan}(\frac{y_1}{y_2})\frac{2}{\pi}$ and **(b)** quarter sine $\tilde{f}_2(y_n, \varphi) = y_n\sin(\frac{\pi}{2}\varphi)$ NFA. The gray pattern represent the neglected input range $0 \leq y_i < 0.5 - 2^{-15}$.

The arctangent function given in (2) is required in order to calculate the angle of rotation in GR1. As there are strong variations of the gradient slope, a simple approximation of the entire function will cause bad results for the proposed approximation scheme in terms of segmentation complexity. In detail, approximations that are performed close to the origin lead to a high number of sub-functions. In order to minimize this problem, the quotient calculation in (2) can be exploited: As every scaling of the numerator and the denominator with an equal constant will lead to exactly the same result, the overall input range can be reduced. An adapted shift operator is integrated into the data path as well as a comparator that detects the total length of zero MSBs. Hence, the input range $0 \leq y_i < 0.5 - 2^{-15}$ can be neglected (see Fig. 1). For computation effort reduction in GR2, a scale factor of $\frac{\pi}{2}$ is also considered for the arctangent NFA ($\tilde{f}_1(y_1, y_2) = \mathrm{atan}(\frac{y_1}{y_2})\frac{2}{\pi}$), limiting the output range to $0 \leq \tilde{f}_1(y_1, y_2) < 1 - 2^{-15}$.

For the NFA of $f_2(y_n, \varphi)$, only a quarter wave of the sine function is considered. The computation of the cosine function is achieved by adding a (scale factor corrected) offset to φ and a function flip back on the approximated quarter sine function.

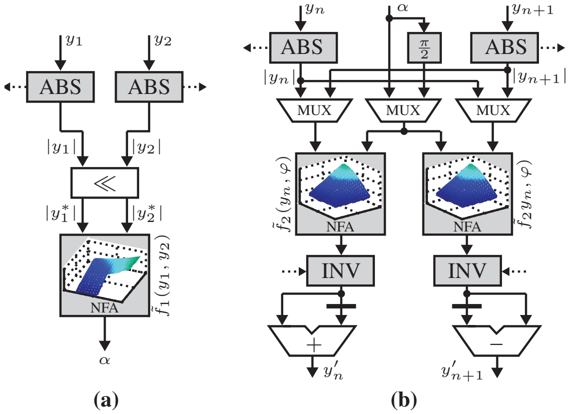

Fig. 2. Data path of 2D NFA-based QRD with **(a)** GR1 and **(b)** GR2. The solid and dotted lines depict the control signals. y_n and y_{n+1} denote the unrotated inputs as well as y'_n and y'_{n+1} denote the results. ABS and INV calculate the absolute value and the conditional result inversion, respectively.

This is done by a simple operand negation, causing only a negligible additional error. Considering negative input values, further additional signal processing is required. If $y_n < 0$, the result of $\tilde{f}_2(y_n, \varphi)$ must be negated. The appearance of negative quotients in GR1 requires an equal treatment for sine function calculation. In contrast to the sine function, the results of GR1 must not be taken into account for the cosine function ($\cos(\varphi) = \cos(-\varphi)$). Due to the scale factor that is used for the approximation of the arctangent, a rescale must be considered for the sine function approximation. Hence, the function $\tilde{f}_2(y_n, \varphi) = y_n \sin(\frac{\pi}{2}\varphi)$ is taken into account.

IV. Architecture

The hardware architecture consists of the control path that determines the sequence of the matrix computation and the data path which realizes all necessary arithmetic calculations (see Fig. 2). The calculation of the $\mathbf{R}$ and $\mathbf{Q}$ matrices requires a memory module with 64 slots that is realized by a register array. The data path width is set to 16 bit (Q.15 fixed-point format) which causes a NFA resolution of 15 bit as only absolute values must be considered. A hardwired hardware-accelerator is selected as global architecture.

The control path comprises a finite state machine (FSM) that is responsible for the global QRD computation sequence as well as for the assignment of operands and results to corresponding memory slots. In order to keep this decoding task as simple as possible, the operand addresses are derivable directly from the actual FSM state. Thus, the FSM is realized as counter with a variable increment. Further, the control unit possesses an input-output (IO) unit enabling the read/write access to $\mathbf{H}$ and $\mathbf{I}$ (containing the values of $\mathbf{Q}$ and $\mathbf{R}$ when QRD calculation is done).

In general, two different computation tasks must be considered for the data path: The angle rotation and the application. As the latter requires significantly higher signal processing effort, two sine-function NFAs are used (see Fig. 2). For the arctangent, only one NFA module is considered. As the calculation in GR1 only depends on a single vector rotation step, it can be executed in parallel to GR2 which causes performance benefits in terms of latency. A graphical overview of the data path is given in Fig. 2. In order to speed-up the QRD processing and avoid divisions by zero, the appearance of zeros in the data path are treated by special means:

1) If y_2 is zero at GR1 calculation, GR2 calculation can be neglected for the entire row.
2) If either y_1 or y_2 is zero at GR2 calculation, the computation of y_2 or y_1 can be neglected, respectively.
3) If both y_1 and y_2 are zero at GR2 calculation, the computation of both y_2 and y_1 can be neglected.

The resulting data path is realized by the use of the time-sharing implementation technique [8]. Thus, the sine function NFAs are used twice in each computation step of GR2. As a consequence, GR1 and GR2 require one and two cycles, respectively. For dynamic power reduction, operand isolation and clock gating are installed [9].

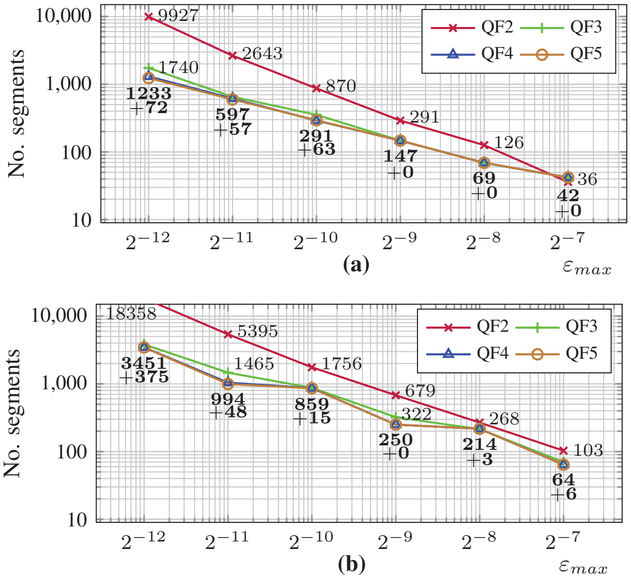

Fig. 3. Segments over specified error for **(a)** $\tilde{f}_1(y_1, y_2)$ and **(b)** $\tilde{f}_2(y_n, \varphi)$ with varying QFs. Bold expressions mark similar results with the added denoting range. Other values depict the number of segments for a single QF.

V. Results

For a meaningful analysis of the results, **1) algorithmic complexity**, **2) application-specific accuracy** and **3) IC implementation** are taken into account in the following.

1) The selection of appropriate NFAs is performed analyzing several NFAs with different specified errors and QFs. From the results in Fig. 3, three data path configurations with different ε_{max} are chosen for the application-specific evaluation. For the sine function approximation, QF3 with $\varepsilon_{max} = 2^{-8}$ (NFA$_1$), QF3 with $\varepsilon_{max} = 2^{-9}$ (NFA$_2$) and QF4 with $\varepsilon_{max} = 2^{-11}$ (NFA$_3$) are taken into account. The arctangent is realized as QF3 with $\varepsilon_{max} = 2^{-8}$ (NFA$_1$) and QF3 with $\varepsilon_{max} = 2^{-11}$ (NFA$_2$ and NFA$_3$).

2) The performance of the proposed QRD is evaluated by the bit error rate (BER) of a wireless multi-antenna communication system using Zero-Forcing Tomlinson-Harashima-

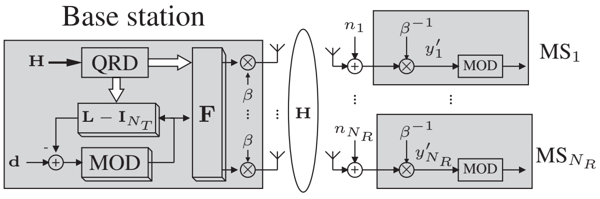

Fig. 4. Basic topology of the ZF-THP in a wireless multi-user MIMO system.

Precoder (ZF-THP) pre-equalization at the transmitter. A multi-user MIMO transmission from a single base station to N_R single-antenna mobile stations is considered. The base station is equipped with $N_T = N_R$ antennas (see Fig. 4). The transmitted data consists of data symbols with quadrature amplitude modulation (QAM) denoted by the vector $\mathbf{d} = [d_1, \ldots, d_{N_T}]^T$. Correspondingly, $\mathbf{y} = [y_1, \ldots, y_{N_R}]^T$ defines the received symbols at the mobile stations. In the baseband the transmission equation can be written as

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n} \ , \qquad (8)$$

where $\mathbf{H} \in \mathbb{C}^{N_R \times N_R}$ denotes the channel matrix and $\mathbf{n} \in \mathbb{C}^{N_R \times 1}$ the complex Gaussian noise with variance σ_n^2. A more detailed description of the used pre-equalizer setup can be found in [6]. The evaluation is performed for NFA_1, NFA_2 and NFA_3 with 16-QAM and 64-QAM modulation schemes as well as 4×4 multi-antenna scenarios. The resulting uncoded BER over E_b/N_0 are compared to ideal and CORDIC-based QRD results [2]. NFA_3 achieves very high accuracy, NFA_1 and NFA_2 still obtain better results than the CORDIC design with six iteration steps. Thus, for IC implementation NFA_1 and NFA_3 are considered, as they achieve highest accuracy and require lowest segmentation effort, respectively.

3) For IC implementation, logical and physical synthesis is applied to the proposed QRD designs NFA_1 and NFA_3 considering the 4×4 QAM-16 scenario. As target technology, the UMC-Faraday 130 nm process is chosen. Considering common hardware performance criteria, NFA_1 and NFA_3 achieve 192 MHz and 149 MHz, 71.75 kGE and 176.76 kGE as well as 273.69 μW/MHz and 960.34 μW/MHz in terms of maximum operating frequency, area and energy, respectively. Taking the computation of a single matrix element annihilation (computation of GR1 and GR2) into account, our approach obtains a maximum number of three cycles which is low compared to CORDIC-approaches with similar accuracy (see Fig. 5 and Tab. I). Considering the trade-off delay, latency and QRD calculation accuracy, the 2D NFA approach achieves very high performance. An overview taking actual references into account is given in Tab. I. In order to increase the throughput, common implementation techniques, for example parallelization [8], must be applied and, hence, must be considered for future work.

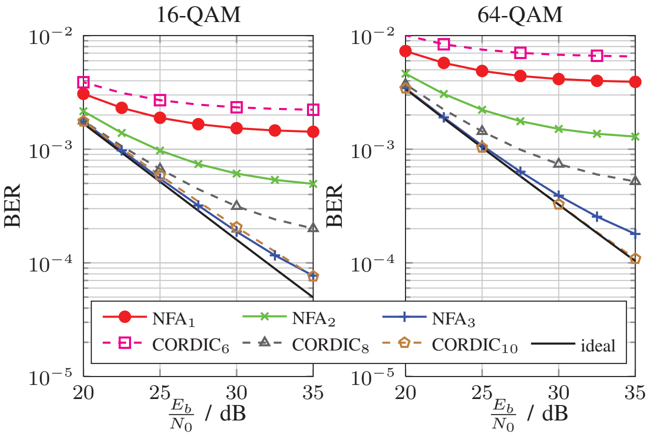

Fig. 5. Uncoded BER performance with proposed QRD realizations in 4×4 MIMO. The CORDIC_m is simulated in floating-point, m denotes the number of micro-rotations.

TABLE I
SYNTHESIS RESULTS OF NFA_1 AND NFA_3 FOR A 4×4 16-QAM MIMO SYSTEM COMPARED TO ACTUAL REFERENCES.

Reference	Frequency [MHz]	Latency[†] [cycles]	Energy [μW/MHz]	Throughput [MQRDs/s]
NFA_1	**192**	**3**	**273.69**	**0.87**
NFA_3	**149**	**3**	**960.34**	**0.68**
[3]	250	8	89.30	31.25
[6]	129	4	44.42	0.45
[4]	100	4	3,190	25

[†] Latency of one matrix element annihilation.

VI. Conclusion

In this paper, a QR-decomposition architecture based on numeric function approximation is presented. Its main advantage refers to the simplification of two-variable functions to a corresponding hardware architecture that performs approximative result computation. The proposed automated technique enables quick function approximation generation in terms of design time. Further, a Givens-Rotation-based approach is realizable allowing a direct processing of the angel of rotation and the Givens-Matrix. The evaluation highlights our work as very efficient in terms of calculation accuracy and the trade-off between delay and latency.

References

[1] R. Fischer, *Precoding and Signal Shaping for Digital Transmission*. John Wiley & Sons, Inc., 2005.

[2] J.-M. Muller, *Elementary Functions: Algorithms and Implementation*, 2nd ed. Birkhaeuser Bosten, 2005.

[3] J.-H. Yoon, D. Shin, and J. Park, "A low-complexity composite QR decomposition architecture for MIMO detector," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014.

[4] Z.-Y. Huang and P.-Y. Tsai, "Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems," *Regular Papers, IEEE Transactions on Circuits and Systems I*, vol. 58, no. 10, Oct. 2011.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins Univ. Press, 1996.

[6] J. Rust, F. Ludwig, and S. Paul, "Low Complexity QR-Decomposition Architecture Using the Logarithmic Number System," in *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2013.

[7] D. Knuth, *Sorting and Searching*, ser. The Art of Computer Programming 3. Addison-Wesley, 1968.

[8] H. Kaeslin, *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication*. Cambridge University Press, 2008.

[9] M. Keating, D. Flynn, R. Aitken, and A. Shi, *Low Power Methodology: For System-on-Chip Design*, ser. Series on Integrated Circuits and Systems. Springer-Verlag, 2008.