Exploiting Special-Purpose Function Approximation For Hardware-Efficient QR-Decomposition

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

***Abstract*—Efficient signal processing takes a key role in application-specific circuit design. For instance, future mobile communication standards, e.g., high-performance industrial mobile communication, require high data rates, low latency and/or high energy-efficiency. Hence, sophisticated algorithms and computing schemes must be explored to satisfy these challenging constraints. In this paper we leverage the paradigm of approximate computing to enable hardware-efficient QR-decomposition for channel pre-coding. For an efficient computation of the Givens-Rotation, bivariate, non-linear numeric functions are taken into account. An effective design method is introduced leading to highly adapted (special-purpose) functions. For evaluation, our work is tested with different configurations in a Tomlinson-Harashima pre-coding downlink environment. In addition, a corresponding HDL implementation is set up and logic and physical CMOS synthesis is performed. The comparison to actual references prove our work to be a powerful approach for future mobile communication systems.**

***Index Terms*—Special-purpose function approximation, QR-decomposition, multi-antenna communication systems**

I. Introduction and Related Work

High-performance wireless communication is a paramount challenge, as there is still an enormous growth of activity fields. Trends like 5G and industrial radio force a significant increase of the data volume that has to be processed without any impairments in terms of energy consumption, latency and data rate [1]. Hence, the employment of todays hardware solutions, e.g., DSPs or GPUs, will not be able to solve this issue sufficiently. Rather, new hardware architectures have to be designed exploiting novel concepts and/or measures for high-performance digital computing.

In the last years, multi-antenna wireless communication, such as multiple-input-multiple-output (MIMO) has turned out to be a promising technology to satisfy these challenging objectives [2]. In this scope, matrix decomposition algorithms are of major significance, e.g., for signal restoration or channel estimation [3]. Another interesting task is channel matrix pre-coding, that reduces the signal processing effort at the receiver as well as it enables multi-user multiple-input-single-output (MU-MISO) wireless communication [2]. Thus, the hardware-efficient realization of matrix decomposition computation schemes is essential to these demanded benefits.

In the last years, a lot of research has been carried out for hardware-efficient matrix decomposition, mainly concentrating on the QR-decomposition (QRD) as underlying computational approach. To achieve high data rates, a large number of CORDIC-based approaches have been proposed, for instance in [4], [5] or [6]. Though this method permits signal processing with only small complexity (area) requirements, parallelization or look-ahead-mechanisms cannot be considered in CORDIC architectures by reasonable effort [7]. Consequently, low-latency architectures, which are of major interest, e.g., in closed-loop industrial radio scenarios [1], are difficult to realize. Other approaches concentrate on high-performance iterative approximation, e.g., an estimation of elementary functions [8] or an approximate extraction of singular vectors [9]. In [10] a multiplier-less and piecewise approximation of non-linear converter functions is introduced, that performs efficient calculations in the logarithmic number system. In the scope of channel pre-coding, also bivariate numeric function approximation (NFA) has proven to be a feasible approach, especially in terms of latency [11]. However, for high-accuracy signal processing comparatively high energy consumption and area requirements have to be accepted mitigating the overall hardware performance.

In this paper we will improve State-of-the-Art bivariate NFA generation techniques for QRD computation by the use of highly adapted signal processing techniques. A customized (which we call special-purpose) bivariate function approximation scheme is introduced taking the application-specific utilization of a single function into account. Out of this, the hardware requirements can be reduced, leading to better performance in terms of transmission quality. To value our work, the results are both tested in a realistic wireless communication scenario and compared to actual references from the literature.

II. Givens-Rotation-based QR-Decomposition

The main idea behind the QRD is to decompose a given system matrix $\mathbf{A}$ into a orthogonal matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$, such that $\mathbf{A} = \mathbf{QR}$. Considering hardware-based QRD, three different computational approaches are generally distributed: Householder reflections, (modified) Gram-Schmidt orthogonalization and the Givens-Rotation [3]. As the latter provides numerical stability and parallel signal processing, it is commonly and frequently employed. On the other hand, the Givens-Rotation requires the calculation of non-linear numeric functions, which appears to be a highly tedious task for hardware-based signal processing.

978-3-9815370-8-6/17/$31.00

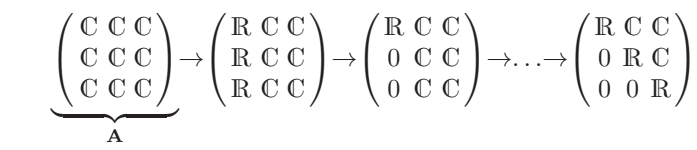

Fig. 1. Possible QRD computation scheme for a 3×3 complex valued matrix. In the first step the imaginary values are rotated to zero. Next, the resulting (co-)sine values are applied to both $\mathbf{A}$ and the unitary matrix $\mathbf{I}$ (not shown in this figure). Next, this procedure restarts taking the real values into account.

The Givens-Rotation is realized by the application of the Givens-Matrix

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

to the sub-vectors $\mathbf{a}_{\mathrm{sub}}$ and $\mathbf{i}_{\mathrm{sub}}$ located within $\mathbf{A}$ and the identity matrix $\mathbf{I}$, respectively. The determination of the (co-) sine values can be realized by

$$\cos(\varphi) = \begin{cases} \left(1 + \left(\frac{x_2}{x_1}\right)^2\right)^{-\frac{1}{2}} & ;\ x_1 \in \mathbb{R} \setminus \{0\} \\ 0 & ;\ \text{else} \end{cases} \quad \text{and} \tag{2}$$

$$\sin(\varphi) = \begin{cases} \left(1 + \left(\frac{x_1}{x_2}\right)^2\right)^{-\frac{1}{2}} & ;\ x_2 \in \mathbb{R} \setminus \{0\} \\ 0 & ;\ \text{else} \end{cases} , \tag{3}$$

and with x_1, x_2 as two elements of $\mathbf{A}$ or $\mathbf{I}$ in the same column. Thus, a single rotation step can be expressed

$$\begin{pmatrix} a_i' \\ a_{i+1}' \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} a_i \\ a_{i+1} \end{pmatrix} ,$$

where $\mathbf{a}' = (a_i', a_{i+1}')^T$ denotes the resulting (rotated) sub-vector. Out of this, Givens-Rotation-based QRD calculates $\mathbf{R}$ by rotating the matrix elements below the main diagonal of $\mathbf{A}$ to zero and applying $\mathbf{G}(\varphi)$ to $\mathbf{A}$ and $\mathbf{I}$. A possible computation scheme is given in Fig. 1. A comprehensive explanation of the Givens-Rotation-based QRD is given in [3].

III. Special-purpose NFA generation

In general, generating special-purpose NFA is an extension to existing function approximation design methods. As it has no dependencies or additional conditions with respect to the other design steps, its application in a vast range of different NFA generation tools is generally possible. In this paper, we consider multiplier-less and piecewise polynomial approximation, as this has proven its feasibility for bivariate numeric functions. Its basic design flow and setup is explained briefly in Sec. III-B. A more sophisticated description can be found in [12].

A. Special-purpose function adaption

The main idea of special-purpose NFA generation refers to a comprehensive reduction of the hardware-based signal processing task by exploiting application-specific function calls. In detail, the distribution of the input operands is examined to allow a relaxation of the computational accuracy. Hence, instead of a static error constraint that is commonly used, a variable error function $\varepsilon(\mathbf{x}) = \varepsilon_{\max}(\mathbf{x})$[1] is considered. To achieve a reasonable input operand distribution, a sufficient number of input stimuli has to be taken into account. In this work, this is realized by a (Matlab-based) simulation environment with a equivalent model of the Givens-Rotation-based QRD.

In order to qualify the distribution of the input operands for bivariate functions, a density function $d(x_1, x_2)$ of a given original function $f(x_1, x_2)$ is set up representing a probability measure of the input operand distribution. Each time $f(x_1, x_2)$ is calculated, the operand location is weighted by $'1'$ and stored to a (surface) matrix $\mathbf{D}$. To keep the required memory amount reasonable, data path quantization effects from the hardware architecture are already considered at this (see Sec. IV). For instance, for a data path size of twelve bits, $2^{12} = 4096$ rows and columns are necessary, leading to $2^{24} \approx 16.78 \times 10^6$ elements of $\mathbf{D}$. Whenever a function call with a pair of (quantized) input operands occurs two or more times, the resulting matrix entry is accumulated. Hence, if this mechanism is applied to a bivariate function used in a certain algorithm of application with realistic stimuli, an application-dependent distribution of the input operands is achieved.

After the accumulation of the input operand distribution has finished, the density function is set up. To this end, locally related weights within the matrix $\mathbf{D}$ are clustered (summarized) to accuracy domains and normalized, scaling the cluster with the highest and lowest number of weights to one and zero, respectively. For an automatic generation, a fixed and uniformly distributed number of those density domains has to be specified in advance. Out of this, the density function $d(x_1, x_2)$ can be set up, returning the normalized clustered weights for a given pair of input operands. A graphical example considering the (co-)sine function from Eq. (2) and Eq. (3) is depicted in Fig. 5.

In order to exploit the density functions in terms of NFA generation, its integration into a corresponding design flow is mandatory. In detail, the static error constant ε is extended to the bivariate function $\varepsilon(x_1, x_2) \sim d(x_1, x_2)$. The relation between the error function and the density function can be realized in many different ways. In this work, we consider the the equation

$$\varepsilon(x_1, x_2) = \varepsilon_{\mathrm{u}} + (-\varepsilon_{\mathrm{u}} + \varepsilon_{\mathrm{l}}) \times \hat{d}(x_1, x_2) \tag{4}$$

with ε_{l} and ε_{u} as the lower and upper (relaxed) border of the error function, respectively. $\hat{d}(x_1, x_2)$ represents a modified density function considering different severities of the function course, defined by

$$\hat{d}(x_1, x_2) := \begin{cases} (1 - d(x_1, x_2)^k) & ;k \geq 1 \\ (1 - d(x_1, x_2))^k & ;k < 1 \end{cases} ;k > 0 \ . \tag{5}$$

Hence, Eq. (4) enables the relaxation of the error constraint between areas of high (ε_{l}) and low (ε_{u}) density and, conse-

[1]The postfix max denotes the maximum error as underlying evaluation criterion.

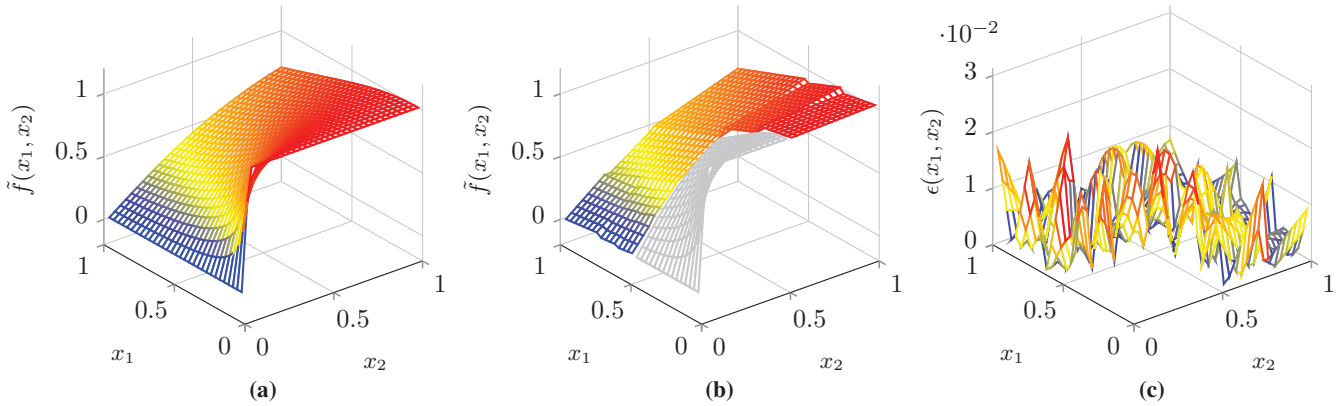

Fig. 2. Exemplary cosine function approximation from (2) with **(a)** the original function $f(x_1, x_2) = \cos(\varphi)$, **(b)** the NFA and **(c)** the resulting NFA derivation $\epsilon(x_1, x_2) = |\tilde{f}(x_1, x_2) - f(x_1, x_2)|$. Note, that the gray pattern does not have to be approximated, due to the operand shift explained in Sec. III-A.

quently, realizes application-specific NFA generation leading to special-purpose functions.

B. Bivariate function approximation

For the proposed special-purpose function approximation technique, a piecewise and multiplier-less NFA design method is considered in this work. Hence, the fundamental approximation can be formulated

$$\tilde{f}(x_1, x_2) = \mathbf{x}\boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ , \tag{6}$$

where x_n denotes the input operands, $n \in \{1, 2\}$ as the dimension index and $\boldsymbol{\beta}$ as the function coefficients. In order to achieve high-accuracy NFAs the original function is split up into several sub-function ranges (segments) each possessing its own set of coefficients.

In general, the automated translation of a bivariate original function to a corresponding HDL-description is fourfold: *1)* The global segmentation, *2)* the multiplier-less approximation, *3)* the pre- and post-optimization and *4)* the hardware mapping step, which will be concisely explained in the following. *1)* In the global segmentation, a first estimation of the resulting segments is set up, considering a non-uniform and accuracy-driven segmentation scheme. In order to qualify the approximation quality, a global function approximation is performed within each segment. In this work this is done taking the well-established Least-Squares (LS) methodology into account. To enable a rough evaluation of the resulting approximation quality, a linear LS-based approximation is used. Therefore, all input operands and function values within the actual function range are taken into account. In addition, data path quantization effects are considered, too.

The global segmentation starts with an LS-estimation, taking the entire function range of the original function into account. If the actual approximation (which is a first simple NFA) does not satisfy the error constraint, the function range is split up into four sub-segments by bisection. Note, that in this work the error function from Sec. III-A is used to enable special-purpose NFA generation. Next, the segmentation restarts by setting up an LS-based approximation for the lower left segment, until the error constraint is met. When an entire set of four segments has been successfully approximated, the remaining segments on the next higher level are processed. The segmentation is completed, when there are no remaining segments. An example is given in Fig. 2. To keep the access to a single sub-segment as simple as possible, the size of each segment as well as the overall function range must be always a power of two value. Consequently, the coefficients of each sub-function are simply accessible by regarding the most significant bits (MSBs) of the input operands.
2) In the approximation step, the global NFA is refined, considering a sophisticated approximation scheme with minimized arithmetic signal processing effort. The linear gradients are replaced by multiplier-less coefficients that possess only a reduced number of nonzero digits. These multiplier-less gradients can be formulated

$$\beta_{n,q} = \sum_{j=1}^{q} \pm \left(2^{\lambda_{n,j}}\right) \ , \tag{7}$$

with q as the total number of nonzero digits and λ as the exponent of the jth partial product. Out of this, the calculation of the gradients can be realized considering shift-and-add techniques [7]. In order to set up multiplier-less gradients, q has to be specified in advance. In addition, the approximation technique that is described in the segmentation step is performed again considering the quantized coefficients. To keep the derivation to the original function as small as possible, β_1, β_2 are refined iteratively. Last, the offset β_0 is adapted by minimizing the total resulting error. The resulting NFA can be formulated

$$\tilde{f}(x_1, x_2) = \mathbf{x} \begin{pmatrix} \beta_{01} & & \beta_{0l} \\ \beta_{11,q} & \dots & \beta_{1l,q} \\ \beta_{21,q} & & \beta_{2l,q} \end{pmatrix} \cdot \boldsymbol{\kappa}(\mathbf{x}) \ , \tag{8}$$

with $\mathbf{x} = (1, x_1, x_2)$, l as the number of segments in total and $\boldsymbol{\kappa}(\mathbf{x})$ realizing the selection of a single equation at runtime.

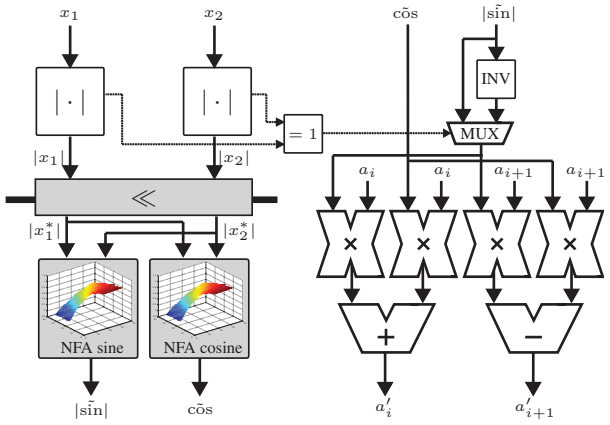

Fig. 3. Proposed hardware architecture for the special-purpose QRD signal processing. The x_n^* represents shifted values preventing the case $x_n < 0.5$.

3) The pre-optimization step exploits the occurrence of function-specific properties in order to reduce the approximation effort at the design time and the runtime. A range reduction can be performed for either symmetric functions or quotients, leading to a significant decrease of the function range. In the latter case, if $x_n < 0.5$, the operands are shifted, until either $x_1 \geq 0.5$ or $x_2 \geq 0.5$ is fulfilled. By this measure, all function values in the range $0 \leq x_n < 0.5$ can be neglected during the NFA generation (see Fig. 2). Additional control logic is inserted into the resulting hardware architecture that realizes this operand scaling by shift operations. In the post-optimization stage further complexity reduction is achieved by the merge of adjacent segments. In order to maintain the specified accuracy, the coefficients of $\tilde{f}(x_1, x_2)$ have to be updated which is done by revoking of the gradient approximation step. If the error constraint is satisfied again, the existing segments are replaced by one non-quadratic segment. In order to guarantee high benefits, the entire set of segments is taken into account in this step.

4) After all necessary parameters have been extracted from the original function, they are mapped to a corresponding hardware description. A (technology-independent) VHDL-based file of the bivariate NFA is created. Template files providing a generic structure of the resulting hardware architecture are utilized that base on the StringTemplate code generator [13]. The input and output ports as well as constants and the result accumulation are directly mapped to corresponding VHDL-expressions and -operands. For the quantized gradients, the input operands are shifted accordingly, which is realized by common VHDL signal reassignments. The segmentation is translated using multiplexers that refer to encapsulated if-statements. In order to enable a straightforward StringTemplate processing, the MSB data is transferred to a binary tree that is set up in a depth-first manner for all items in the set of segments; the input operands are be stored in an alternating sequence. To reduce the complexity of the multiplexer, an gradient encoding scheme is exploited. Each partial product is mapped to a related constant that has to traverse though the multiplexer tree. For the decoding, an additional multiplexer has to be inserted at the output of the multiplexers, re-mapping the encoded values back to the partial gradients.

IV. Architecture

For the hardware-based realization of the QRD, the NFA HDL-implementation has to be embedded into an adapted hardware architecture. Because of the very simple processing scheme, this is realized by an hard-wired hardware accelerator. Further, a control path is installed which determines the sequence of the matrix computation as well as a data path that handles for the arithmetic calculations (see Fig. 3). Due to the selected architecture, a register array with 64 slots is required in which the $\mathbf{R}$ and $\mathbf{Q}$ matrices are stored. The data path width is set to 14 bit (Q.13 fixed-point format).

The control path consists of a finite state machine (FSM) realizing both the global QRD computation and the assignment of operands and results to the register array. To keep the control overhead reasonable, the FSM is realized as counter with a variable increment. For the communication to external modules, the control unit further possesses an input-output (IO) unit.

In the data path, all arithmetic signal processing units are located. For the proposed QRD architecture, two different tasks can be identified: The calculation of the (co-)sine values and its application to the remaining matrix elements in the row. For the former Eq. (2) and Eq. (3) are taken for special-purpose NFA generation. The latter is realized by generic addition an multiplication hardware structures. An overview of both tasks is given in Fig. 3. To improve the performance of the QRD processing, the occurrence of zeros within the data path are supported by special means. Hence, if x_{i+1} is zero at the NFA input, the remaining elements in the row remain untouched. Further, ff either x_1 or x_2 is zero, the computation of x_1 or x_2 is neglected, respectively.

V. Results

In order to evaluate the performance of the designed QRD, a wireless MU-MISO communication system is simulated. It consists of a single base station with N_B antennas and $N_M = N_B = N$ decentralized non-cooperative single-antenna mobile stations. The vector of the M-QAM data symbols is given by $\mathbf{d} = [d_1, \ldots, d_{N_B}]^T$, accordingly the received symbols are defined by $\mathbf{y} = [y_1, \ldots, y_{N_M}]^T$. For the considered downlink transmission scenario, the base station sends data to the mobiles. For a fair comparison of the simulation results, a perfect knowledge of the downlink channel matrix $\mathbf{H} \in \mathbb{C}^{N_M \times N_B}$ in the base station is assumed. In this contribution, a Rayleigh multipath channel model is applied, where the channel entries of $\mathbf{H}$ are normalized independent and identically distributed zero-mean complex Gaussian random variables. At the mobile stations the received signal is added to the noise vector $\mathbf{w} \in \mathbb{C}^{N_M \times 1}$ of complex Gaussian independent and identically distributed samples with variance σ_w^2. Generally, for this system the transmission equation reads as

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{w}\,. \tag{9}$$

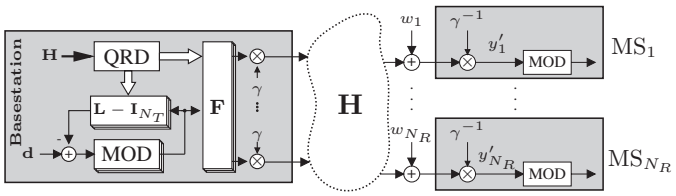

Fig. 4. ZF-THP structure in the base station for downlink transmission with MS as a set of decentralized mobile receiver units.

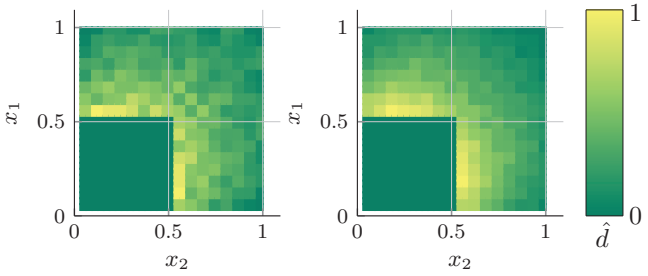

Fig. 5. Density functions for the **(a)** 4×4 and **(b)** 8×8 MU-MISO ZF-THP system.

In this contribution the ZF-THP is used. The system setup for the downlink THP is shown in Fig. 4. The data $\mathbf{d}$ is precoded in the base station so that the transmitted data $\mathbf{s}$ is produced. The unitary feedforward filter matrix $\mathbf{F}$ obtains spatial causality and the feedback filter matrix $(\mathbf{L} - \mathbf{I}_{N_B})$ eliminates the interferences. A modulo device (MOD) is used to map the values into fundamental Voronoi region which depends on the order of the chosen QAM. This will be revoked by another MOD device in the mobiles. The required filter matrices in Fig. 4 can be calculated by a QRD of $\mathbf{H}^{\mathrm{H}}$, which is the conjugate transpose of the channel matrix $\mathbf{H}$. For a system with $N_M = N_B$ antennas this results in $\mathbf{H}^{\mathrm{H}} = \mathbf{QR}$. Then, the feedforward filter matrix is defined by $\mathbf{F} = \mathbf{QV}$, where $\mathbf{V} = \mathrm{diag}\{\mathrm{diag}^{-1}\{\mathbf{R}\}\}$ consists of the diagonal matrix elements of $\mathbf{R}$. The lower left feedback filter matrix is obtained by $\mathbf{L} = \mathbf{HQV}$. The scalar γ is used to fulfill the power constraint and is defined as

$$\gamma = \sqrt{\frac{N_B}{\mathrm{tr}\{\mathbf{QVV}^{\mathrm{H}}\mathbf{Q}^{\mathrm{H}}\}}}\,. \tag{10}$$

The evaluation of the special-purpose QRD is threefold: First, an appropriate error function is set up and its results are analyzed. Next, Application-specific analysis is carried out, considering a sophisticated Matlab model of the NFA and the MU-MISO wireless communication system. In a last step, the HDL-implementation of a corresponding QRD hardware architecture is synthesized and its performance values are compared to actual references.

A. Analysis of special-purpose functions

In order to specify an appropriate error function, two density functions for $N = 4$ and $N = 8$ are set up. For the generation of the density function, a set of $2^4 \times 2^4 = 256$ uniform clusters is taken into account in which locally related weights are summarized to independent accuracy domains (see Sec. III-A). A Q.13 fixed-point number format is used, leading to an data path width of 14 bits. To avoid overflows, the input data of the channel matrix is scaled to $0 \leq x < 0.125$. Exhaustive simulation of the ZF-THP is carried out with ideal, quantized (co-)sine functions.

The results in Fig. 5 indicate significant density variations Because of the operand shifting of post-optimization step (see Sec. III-B) no input operands occur in the range $x_n < 0.5$. Thus, the value $x_n = 0.5$ denotes a sharp edge. In the range $0.5 \leq x_n < 1$ the input operand density continuously decreases. Note, that in this application the density function is a symmetric function, as the sine and cosine function are always calculated with the same values for Givens-Rotation.

To set up an appropriate error function $\varepsilon(x_1, x_2)$, the scale factor k must be adapted to the actual application. Hence, for the elements $k \in \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1, 2, 3, 4\}$ a rough simulation is performed, taking special-purpose NFAs into account. Therefore, ε_{l} and ε_{u} are set to 2^{-12} and 2^{-10}, respectively. About $10,000$ QRDs are calculated within the proposed ZF-THP system with both $N = 4$ and $N = 8$ as well as a 64-QAM. The number of non-zero digits is set up $q = 3$. As a result, $k = \frac{1}{3}$ achieves best results with a total amount of 858 and 681 segments for $N = 4$ and $N = 8$, respectively. Hence, these configurations are used for further evaluation.

B. Application-specific accuracy

In the next step, a more sophisticated evaluation of the application-specific performance is carried out, considering the uncoded BER-performance of the ZF-THP system. A 4×4 ($N = 4$) and 8×8 ($N = 8$) antenna scenario is considered with a 64-QAM. In addition the special-purpose NFA from Sec. III-A is taken into account and compared to actual references. The results in Fig. 6 highlight our approach to be highly suitable in terms of BER performance. Thus, special-purpose NFA generation is a powerful extension in the scope of Givens-Rotation-based QRD in ZF-THP design. An overview over the uncoded BER results is given in Fig. 6.

C. IC implementation

In a last step, a HDL-description for the presented QRD is implemented for the 4×4 multi-antenna system and evaluated. Logic and physical synthesis is carried out using the Cadence RTL-compiler and Encounter design tools, considering timing, power and parasitic noise effects. As target technology the TSMC 90 nm general purpose process (in a typical case configuration) is taken into account. In order to enable a fair complexity comparison, kilo gate equivalents (kGE) are regarded [14].

According to the results in Tab. I, our design reaches a maximum operating frequency of 151 MHz. For the computation of the rotation and application in the QRD algorithm values and one cycles are required, respectively, leading to a total

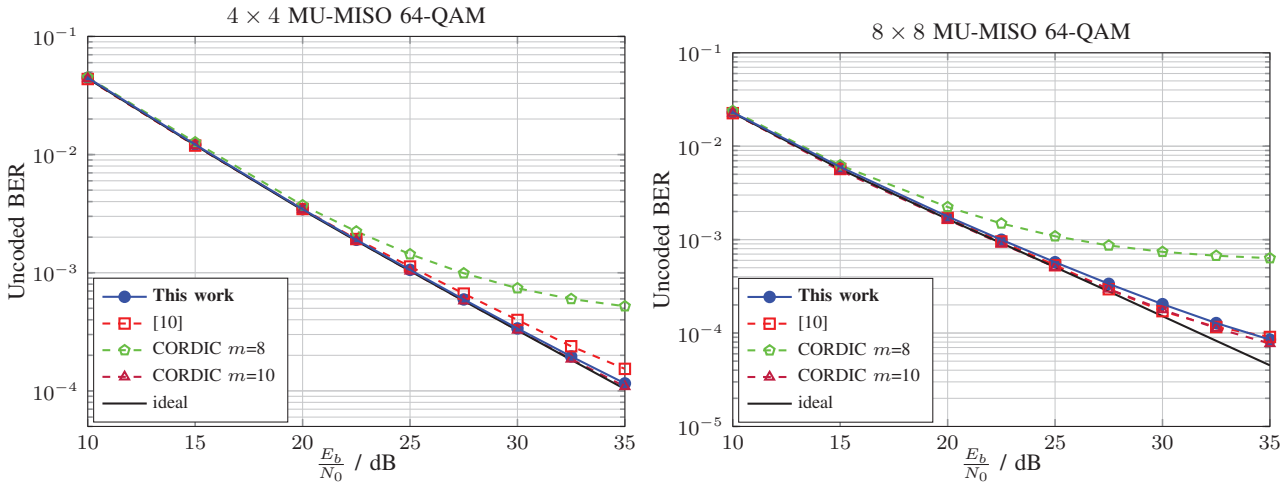

Fig. 6. Uncoded BER performance at the receivers side of the proposed special-purpose NFA QRD realization compared to actual references. The CORDIC is simulated in double precision floating-point with m as the number of micro-rotations.

number of 133 cycles and, consequently, to a throughput of 1.35×10^6 QRDs. Further, a total area of 50.01 kGE and an energy consumption of 24.67 pJ is required.

In summary, the exploitation of special-purpose NFA generation is a powerful first step for the hardware-efficient signal processing of the QRD. For future work, high-throughput realizations of our work must be considered in order to satisfy the challenging constraints of next generation mobile communication standards. For instance, a more sophisticated hardware architecture must be exploited, e.g., systolic arrays or dynamically reconfigurable systems.

VI. Conclusion

In this paper, a novel Givens-Rotation-based QR-decomposition architecture based on special-purpose numeric function approximation is presented. The bivariate (co-)sine function is highly adapted to a pre-coding system by exploiting the distribution of the input operands. Out of this, a variable error function is set up that enables accuracy relaxations for function ranges with only a low utilization. The evaluation highlights our work as very hardware-efficient approach, especially in terms of calculation accuracy and energy-efficiency.

TABLE I
Comparison of the synthesis results considering our proposed special-purpose NFA as well as actual references in a 4×4 64-QAM MU-MISO system.

Reference	Frequency [MHz]	Complexity [kGE]	Energy [μW/MHz]	Throughput [MQRDs/s]
This work	**151**	**50.01**	**24.67**	**1.35**
[11]†	192	n/a	273.69	0.87
[10]†	129	22.38	44.42	0.45
[4]	143	451.53	653.85	35.75
[9]	182	120	98.90	7

† *Designed in a* 130 *nm CMOS technology.*

References

[1] A. Varghese and D. Tandur, "Wireless requirements and challenges in industry 4.0," in *International Conference on in Contemporary Computing and Informatics (IC3I)*, Nov 2014, pp. 634–638.

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

[3] A. Burg, "VLSI Circuits for MIMO Communication Systems," Ph.D. dissertation, Swiss Federal Institute of Technology Zurich, 2006.

[4] Y. T. Hwang, K. T. Chen, and C. K. Wu, "A high throughput unified SVD/QRD precoder design for MIMO OFDM systems," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, July 2015, pp. 1148–1151.

[5] B. Han, Z. Yang, and Y. R. Zheng, "Efficient implementation of iterative multi-input-multi-output orthogonal frequency-division multiplexing receiver using minimum-mean-square error interference cancellation," *IET Communications*, vol. 8, no. 7, pp. 990–999, May 2014.

[6] ——, "FPGA implementation of QR decomposition for MIMO-OFDM using four CORDIC cores," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 4556–4560.

[7] J.-M. Muller, *Elementary Functions: Algorithms and Implementation*, 2nd ed. Birkhäuser Bosten, 2006.

[8] G. R. Prabhu, B. Johnson, and J. S. Rani, "FPGA Based Scalable Fixed Point QRD Core Using Dynamic Partial Reconfiguration," in *2015 28th International Conference on VLSI Design*, Jan 2015, pp. 345–350.

[9] C. Z. Zhan, Y. L. Chen, and A. Y. Wu, "Iterative Superlinear-Convergence SVD Beamforming Algorithm and VLSI Architecture for MIMO-OFDM Systems," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 3264–3277, June 2012.

[10] J. Rust, F. Ludwig, and S. Paul, "Low Complexity QR-Decomposition Architecture Using the Logarithmic Number System," in *Design, Automation Test in Europe Conference Exhibition*, 2013, pp. 97–102.

[11] ——, "QR-Decomposition Architecture Based on Two-Variable Numeric Function Approximation," in *Design, Automation Test in Europe Conference Exhibition*, 2015.

[12] J. Rust, N. Heidmann, and S. Paul, "Two-Variable Numeric Function Approximation Using Least-Squares-Based Regression," in *Nordic Circuits and Systems Conference (NORCAS): NORCHIP International Symposium on System-on-Chip (SoC), 2015*, Oct 2015, pp. 1–4.

[13] T. Parr, "Enforcing Strict Model-view Separation in Template Engines," in *Proceedings of the 13th International Conference on World Wide Web*, New York, NY, USA, 2004.

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