Fixed-Point CORDIC-Based QR Decomposition by Givens Rotations on FPGA

Dongdong Chen and Mihai SIMA
Department of Electrical and Computer Engineering, University of Victoria
EOW 448, 3800 Finnerty Rd., Victoria, B.C., V8P 5C2, Canada
Email:{ddchen, msima@ece.uvic.ca}

***Abstract*— This paper presents a parallel architecture of an QR decomposition systolic array based on the Givens rotations algorithm on FPGA. The proposed architecture adopts a direct mapping by 21 fixed-point CORDIC-based process units that can compute the QR decomposition for an 4×4 real matrix. In order to achieve a comprehensive resource and performance evaluation, the computational error analysis, the resource utilized, and speed achieved on Virtex5 XC5VTX150T FPGA, are evaluated with the different precision of the intermediate word lengthes. The evaluation results show that 1) the proposed systolic array satisfies 99.9% correct 4×4 QR decomposition for the 2^{-13} accuracy requirement when the word length of the data path is lager than 25-bit; 2) occupies about $2,810$ (13%) slices, and achieves about 2.06 *M/sec* updates by running at the maximum frequency 111 *MHz*.**

I. Introduction

Nowadays, many signal processing applications, such as the adaptive beamforming, the 3G wireless communication, and the software defined radio, etc., apply the QR decomposition to factorize the matrix A into a product $A = QR$, where Q is an orthogonal matrix and R is an upper triangular matrix. In order to solve the QR decomposition, there are three different methods including the Gram-Schmidt algorithm [1], the Householder transformations algorithm [2], and the Givens rotations algorithm [3]. The QR decomposition using Givens rotations is a well-established technique that can be implemented using a efficient parallel and pipelined triangular systolic array architecture [4], [5], [6]. The standard Givens rotations requires square root, divisions and some trigonometric operations. However, those computer arithmetic units in hardware is not efficient to be applied for the QR decomposition. In [7], [8], [9], the square-root and division free Givens rotations have been proposed. However, by this kind of method, Givens rotations still need to perform additional multiplication operation.

The coordinate rotation by digital computer (CORDIC) algorithm [10], [11] is a hardware-friendly algorithm that can compute various transcendental functions by only using the shift and add operations.The QR decomposition systolic array based on Givens rotations need one operation to calculate the Givens rotations angle, and then perform another operation to apply this angle to the elements of input matrix. Therefore, the CORDIC algorithm can be applied to perform the angle calculation and rotation operations using vectoring and rotation modes of the CORDIC operations. The CORDIC-based processors can be efficiently mapped into the QR decomposition systolic array, which have been presented in [12], [13], [14], [15], [16], [17].

In this paper, we present a CORDIC-based QR decomposition systolic array by Givens rotations algorithm on FPGA. The main aspects of this work include:

- A parallel architecture of a QR decomposition systolic array that allows a direct mapping by 21 CORDIC-based process units with the fixed-point data-path for an 4×4 real matrix.
- A process unit based on a modified CORDIC algorithm, by which the multiplication of the inverse of the scale factor can be compensated by merging scaling operations into CORDIC iterations.
- A comprehensive resource and performance evaluation, including the computational error analysis, the resource utilized, and speed achieved on FPGA, with the different precision of the fixed-point intermediate word lengthes.
- A function verification platform for verify the hardware function of the QR decomposition systolic array on FPGA.

This paper is organized as follows: Section II reviews the most important aspects of the CORDIC-based Givens rotations algorithm for QR decomposition. In Section III, we present a parallel architecture of the QR decomposition systolic array, and then analyze the computational errors with the different word lengthes. Section IV analyzes the implementation results, and then compare the hardware performance of the proposed architecture with the design [17] on FPGA. Section V gives conclusions.

II. Algorithm

To make the presentation self-contained, the most important aspects of the CORDIC-based Givens rotations algorithm for QR decomposition are given as follows:

A. Overview of Givens Rotations

In order to achieve the upper triangular matrix R for an 4×4 real matrix A, a series of Givens rotations are applied to zero the elements of the lower triangular matrix by the rotation matrices G. In each rotation, two adjacent rows $A_{i-1,k}$ and $A_{i,k}$ are rotated by the rotation matrix $G_{i,k}$ in order to zero the element $a_{i,k}$, where $i \in \{2, ..., 4\}$ and $k \in \{1, ..., 4\}$. This

978-0-7695-4551-6/11 $26.00 © 2011 IEEE
DOI 10.1109/ReConFig.2011.38

can be expressed by:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \times \begin{bmatrix} A_{i-1,k} \\ A_{i,k} \end{bmatrix} = \begin{bmatrix} a'_{i-1,1} & ... & a'_{i-1,k} & ... & a'_{i-1,4} \\ 0 & ... & 0 & ... & a'_{i,4} \end{bmatrix} \quad (1)$$

In (1), the sine s and cosine c parameters consist of the rotation matrix $G_{i,k}$, where s and c can be achieved by (2) and (3) respectively:

$$c = \frac{a_{i-1,k}}{\sqrt{a_{i-1,k}^2 + a_{i,k}^2}} = \cos(\theta_{i,k}) \quad (2)$$

$$s = \frac{a_{i,k}}{\sqrt{a_{i-1,k}^2 + a_{i,k}^2}} = \sin(\theta_{i,k}) \quad (3)$$

where $\theta_{i,k} = \tan^{-1}(\frac{a_{i,k}}{a_{i-1,k}})$. Therefore, updated elements of two adjacent rows $a'_{i-1,n}$ and $a'_{i,n}$ can be expressed by:

$$a'_{i-1,n} = \cos(\theta_{i,k}) \times a_{i-1,n} + \sin(\theta_{i,k}) \times a_{i,n} \quad (4)$$

$$a'_{i,n} = \cos(\theta_{i,k}) \times a_{i,n} - \sin(\theta_{i,k}) \times a_{i-1,n} \quad (5)$$

Where, $n \in \{1, ..., 4\}$ and $n \neq k$. The sequential process of the Givens rotations starts from zeroing the bottom elements to the upper elements of the first column, and then move to the second column, and so on. In doing so, the upper triangular matrix R is achieved.

In order to achieve the orthogonal matrix Q, since the equation (6) is satisfied:

$$I = Q^T \times Q \quad (6)$$

The transpose of matrix Q can be computed by the successive multiplication of the rotation matrix $G_{i,k}$:

$$Q^T = \prod G_{i,k} \times I \quad (7)$$

B. QR Decomposition by CORDIC

The CORDIC algorithm is an iterative method for performing vector rotations by arbitrary angles using only shifts and additions, as shown in (8), (9) and (10):

$$x[j+1] = x[j] - \sigma[j] 2^{-j} y[j] \quad (8)$$

$$y[j+1] = y[j] + \sigma[j] 2^{-j} x[j] \quad (9)$$

$$z[j+1] = z[j] - \sigma[j] arctan(2^{-j}) \quad (10)$$

In (8), (9) and (10), $x[0]$ and $y[0]$ are the initial coordinate values, $z[0]$ is the initial angles, and $\sigma[j] \in \{1, -1\}$ is the direction of rotation for the iteration j. Since CORDIC algorithm is not perfect rotation, there is a scale factor $k_j = \sqrt{1+\sigma[j]^2 2^{-2j}}$ in each iteration. After $j+1$ iterations, the final results of $x[j+1]$ and $y[j+1]$ are scaled up by a scale factor K:

$$K = \prod_{j=0}^{\infty} \sqrt{1+\sigma[j]^2 2^{-2j}} \approx 1.64676025812 \quad (11)$$

In order to avoid the complex square-root and division operations in (2) and (3), the CORDIC algorithm operated in the vectoring mode can be applied to compute $\theta_{i,k}$. In the vectoring mode, if $x[j]y[j] \geq 0$, $\sigma[j] = -1$; otherwise $\sigma[j] = 1$,

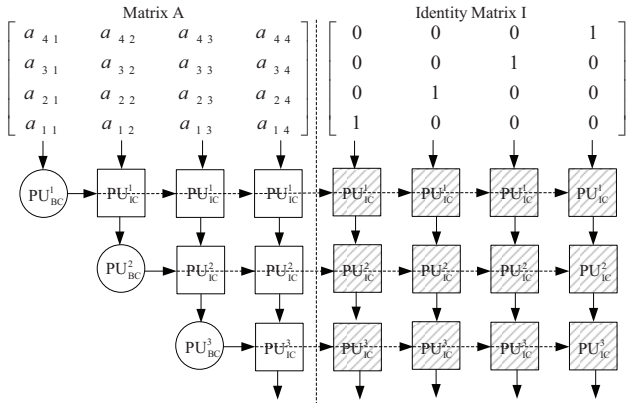

Fig. 1. QR decomposition systolic array architecture for an 4×4 matrix.

where $x[0] = a_{i-1,k}$ and $y[0] = a_{i,k}$. After performing the last iteration, the results are:

$$x[j+1] \approx K \sqrt{a_{i-1,k}^2 + a_{i,k}^2} \quad (12)$$

$$y[j+1] \approx 0 \quad (13)$$

$$\theta_{i,k} \approx \{\sigma[1], ..., \sigma[j]\} \quad (14)$$

To substitute the sine s and cosine c parameters into (4) and (5), it is obtained in (12) that $\frac{1}{K} x[j+1]$ represents the updated elements $a'_{i-1,k}$ of the updated matrix A. At the same time, $a_{i,k}$ is zeroed to $y[j+1]$ by the CORDIC iterations.

In (14), $\theta_{i,k}$ is represented by the digit sequence $\{\sigma[1], ..., \sigma[j]\}$ that can be applied to compute the updated elements based on the CORDIC algorithm in the rotation mode. After the last iteration, we achieve:

$$x[j+1] \approx K(\cos(\theta_{i,k}) \times a_{i-1,n} + \sin(\theta_{i,k}) \times a_{i,n}) \quad (15)$$

$$y[j+1] \approx K(\cos(\theta_{i,k}) \times a_{i,n} - \sin(\theta_{i,k}) \times a_{i-1,n}) \quad (16)$$

Therefore, the values of $\frac{1}{K} x[j+1]$ and $\frac{1}{K} y[j+1]$ are achieved in order to represent the updated elements $a'_{i-1,n}$ and $a'_{i,n}$.

C. Approximation of Scale Factor

A feature of the CORDIC algorithm is an increase of the magnitude of the vector by a scale factor $K \approx 1.64676025812$. In order to achieve correct Givens rotations results based on the CORDIC algorithm, the results should be compensated by a final multiplication with the inverse of the scale factor $\frac{1}{K} \approx 0.607252935$. However, this approach is based on the multiplication operation that lead to the increase of the latency and the reduction of the throughput of the QR decomposition in hardware. Therefore, there is a need for a method that can simplify the scaling operation and achieve a significant reduction in the overall hardware requirement.

According to [18], the inverse of the scale factor $\frac{1}{K}$ can be expanded into the product of the elementary factors $1+\sigma_n 2^{-n}$, where n is the shift length of a scaling operation, and $\sigma_n = \pm 1$. Therefore, the multiplication of the inverse of the scale factor can be compensated by merging the scaling operation into the

TABLE I

TIMING TABLE OF QR DECOMPOSITION SYSTOLIC ARRAY ARCHITECTURE FOR THE MATRIX R.

Time	Boundary Cells $\rightarrow \{\theta_{i,k}, r^*_{k,k}\}$	Internal Cells $\rightarrow \{r^*_{i-1,n}, r^*_{i,n}\}$
t_0	$PU^1_{BC}\{a_{1,1}, a_{2,1}\} \rightarrow \{\theta_{2,1}, r^{t_0}_{1,1}\}$	$PU^1_{IC}\{a_{1,2}, a_{2,2}; a_{1,3}, a_{2,3}; a_{1,4}, a_{2,4}\} \rightarrow \{r^{t_0}_{1,2}, r^{t_0}_{2,2}; r^{t_0}_{1,3}, r^{t_0}_{2,3}; r^{t_0}_{1,4}, r^{t_0}_{2,4}\}$
t_1	$PU^1_{BC}\{r^{t_0}_{1,1}, a_{3,1}\} \rightarrow \{\theta_{3,1}, r^{t_1}_{1,1}\}$	$PU^1_{IC}\{r^{t_0}_{1,2}, a_{3,2}; r^{t_0}_{1,3}, a_{3,3}; r^{t_0}_{1,4}, a_{3,4}\} \rightarrow \{r^{t_1}_{1,2}, r^{t_1}_{3,2}; r^{t_1}_{1,3}, r^{t_1}_{3,3}; r^{t_1}_{1,4}, r^{t_1}_{3,4}\}$
t_2	$PU^1_{BC}\{r^{t_1}_{1,1}, a_{4,1}\} \rightarrow \{\theta_{4,1}, r^*_{1,1}\}$ $PU^2_{BC}\{r^{t_0}_{2,2}, r^{t_1}_{3,2}\} \rightarrow \{\theta_{3,2}, r^{t_2}_{2,2}\}$	$PU^1_{IC}\{r^{t_1}_{1,2}, a_{4,2}; r^{t_1}_{1,3}, a_{4,3}; r^{t_1}_{1,4}, a_{4,4}\} \rightarrow \{r^*_{1,2}, r^{t_2}_{4,2}; r^*_{1,3}, r^{t_2}_{4,3}; r^*_{1,4}, r^{t_2}_{4,4}\}$ $PU^2_{IC}\{r^{t_0}_{2,3}, r^{t_1}_{3,3}; r^{t_0}_{2,4}, r^{t_1}_{3,4}\} \rightarrow \{r^{t_2}_{2,3}, r^{t_2}_{3,3}; r^{t_2}_{2,4}, r^{t_2}_{3,4}\}$
t_3	$PU^2_{BC}\{r^{t_2}_{2,2}, r^{t_2}_{4,2}\} \rightarrow \{\theta_{4,2}, r^*_{2,2}\}$	$PU^2_{IC}\{r^{t_2}_{2,3}, r^{t_2}_{4,3}; r^{t_2}_{2,4}, r^{t_2}_{4,4}\} \rightarrow \{r^*_{2,3}, r^{t_3}_{4,3}; r^*_{2,4}, r^{t_3}_{4,4}\}$
t_4	$PU^3_{BC}\{r^{t_2}_{3,3}, r^{t_3}_{4,3}\} \rightarrow \{\theta_{4,3}, r^*_{3,3}\}$	$PU^3_{IC}\{r^{t_2}_{3,4}, r^{t_3}_{4,4}\} \rightarrow \{r^*_{3,4}, r^*_{4,4}\}$

each CORDIC iteration as shown in:

$$x[j+1] = x[j](1+\sigma_n 2^{-n}) - \sigma[j]2^{-j}y[j](1+\sigma_n 2^{-n}) \quad (17)$$

$$y[j+1] = y[j](1+\sigma_n 2^{-n}) + \sigma[j]2^{-j}x[j](1+\sigma_n 2^{-n}) \quad (18)$$

The results given in (19) provide a precision of 20-bit scale factor [18] that can be expanded into a product of the elementary factors and merged by 11 scaling iterations:

$$\frac{1}{K} \approx (1-2^{-2})(1-2^{-3})(1-2^{-4})(1-2^{-5})^2 (1+2^{-6})^2(1+2^{-7})^3(1-2^{-8}) \approx 0.607252935 \quad (19)$$

In order to keep the delay over interconnections as low as possible, lower values for $n+j$ can be achieved by merging the short-shift CORDIC iterations with long-shift scaling iterations. For example, the first CORDIC iteration is merged with the largest-shift scaling iteration $1-2^{-8}$, and the second CORDIC iteration with the next largest-shift scaling iteration $1+2^{-7}$, and so on.

III. QR DECOMPOSITION SYSTOLIC ARRAY

Fig. 1 shows the proposed systolic array architecture based on the fixed-point CORDIC algorithm for an 4×4 matrix QR decomposition. The architecture consists of two kinds of CORDIC process units, the boundary cells (PU_{BC}) and the internal cells (PU_{IC}). Each row of cells in the systolic array performs a Givens rotation between the row of the updated upper triangular matrix R and the elements received by the input data stream in order to eliminate the lower triangular elements of the updated matrix. In each Given rotation, the boundary cell (PU_{BC}) computes the appropriate rotation angle $\theta_{i,k}$ and the diagonal element of the updated upper triangular matrix R. At the same time, the rotation angle $\theta_{i,k}$ that is represented by the digit sequence $\sigma[j]$ is sent to the internal cells (PU_{IC}) to compute the rest elements of the updated upper triangular matrix R and the updated transpose of orthogonal matrix Q^T. In Fig. 1, the right part of the systolic array (highlighted in dashed *PUs*) is applied to compute the transpose of orthogonal matrix Q^T with the input identity matrix I, while the left part is applied to compute the upper triangular matrix R with the input matrix A. In the data-path of the proposed systolic array architecture, in order to represent the signed binary intermediate value, all variables in the architecture are represented by the binary fixed-point numbers with 2's complement number system. Elements of the real matrix A are represented by 16-bit fixed-point binary fractional numbers in the range of $(-1, +1)$.

A. Timing Table

Table I gives the timing table of the proposed systolic array architecture for achieving the upper triangular matrix R. The timing table of the computation of the transpose of orthogonal matrix Q can be implemented in a similar manner. Since we set 18 as the number of iterations of each CORDIC-based process unit, the each time t is equal to 18 iterations in Table I.

At t_0, the elements of the first row and the second row of the input matrix A and the identity matrix I are processed in the first row of process units (PU^1_{BC} and PU^1_{IC}) in order to eliminate the element $a_{2,1}$. At t_1, the elements of the first row of R_{t_0} and $Q^T_{t_0}$, and the elements of the third row of A and I are processed in PU^1_{BC} and PU^1_{IC} in order to eliminate the element $a_{3,1}$. At t_2, the elements of the first row of R_{t_1} and $Q^T_{t_1}$, and the elements of the fourth row of A and I are processed in PU^1_{BC} and PU^1_{IC} in order to eliminate the element $a_{4,1}$. Therefore the final results of the first row of the upper triangular matrix R and the transpose of orthogonal matrix Q^T ($r^*_{1,n}$ and $q^*_{1,n}$) are achieved. At the same time, the elements of the second row of R_{t_0} and $Q^T_{t_0}$, read out from a random-access memory (RAM), and the elements of the third row of R_{t_1} and $Q^T_{t_1}$ are processed in the second row of process units (PU^2_{BC} and PU^2_{IC}) in order to eliminate the element $r^{t_0}_{3,2}$. At t_3, the elements of the second row of R_{t_2} and $Q^T_{t_2}$, and the elements of the fourth row of R_{t_2} and $Q^T_{t_2}$ are processed in the third row of process units (PU^3_{BC} and PU^3_{IC}) in order to eliminate the element $r^{t_2}_{4,2}$. Therefore the final results of the second row of R and Q^T ($r^*_{2,n}$ and $q^*_{2,n}$) are obtained. At t_4, the elements of the third row of R_{t_2} and $Q^T_{t_2}$, read out from a RAM, and the elements of the fourth row of R_{t_3} and $Q^T_{t_3}$ are processed in PU^3_{BC} and PU^3_{IC} in order to eliminate the element $r^{t_3}_{4,3}$. Therefore the final results of the third row and the forth row of R and Q^T ($r^*_{2,n}$, $r^*_{4,n}$, $q^*_{2,n}$ and $q^*_{4,n}$) are achieved. After 80 CORDIC iterations, the first result of the upper triangular matrix R and the orthogonal matrix Q are achieved for a 4×4 input matrix A. The second input matrix A can be sent to the proposed systolic array after t_3, such that the second result of the QR decomposition can be achieved in next 54 iterations.

B. Error Evaluation by Different Word Lengthes

In order to achieve a specific accuracy for the QR decomposition, the precision of the intermediate word length and final computational errors need to be evaluated. In this work, first,

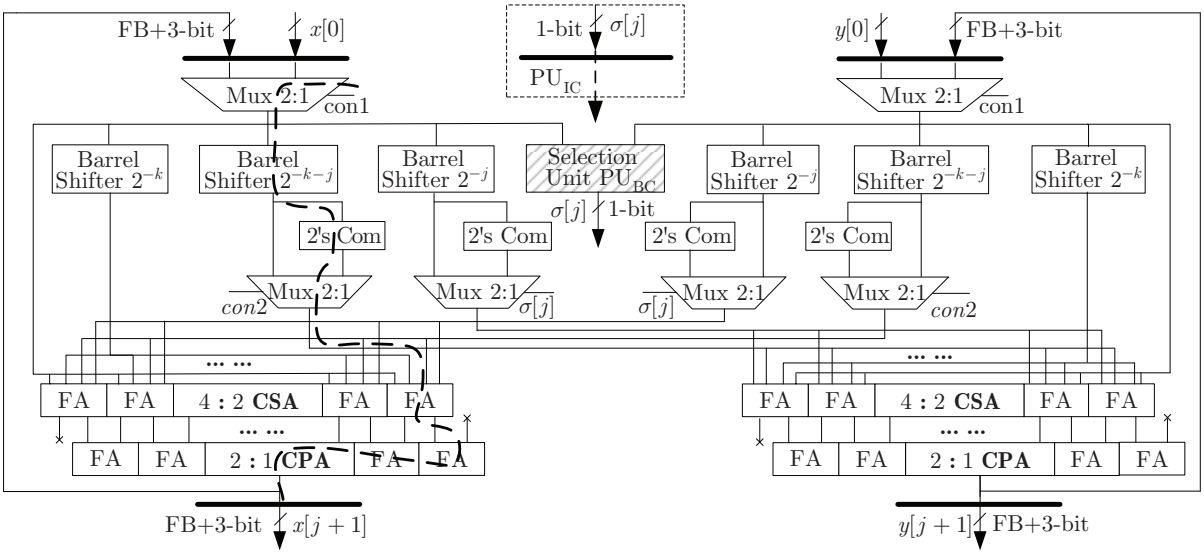

Fig. 2. Architecture of the proposed CORDIC-based process unit.

we set 2^{-13} as the required accuracy because it is sufficiently accurate for many signal processing applications; second, the word length of intermediate values are defined as *FB*+3-bit fixed-point numbers including 1-bit sign, 2-bit integer part and *FB*-bit fractional part; third, the final results of the QR decomposition are rounded to the 19-bit fixed-point number including 3-bit integer part and 16-bit fractional part.

Based on the proposed CORDIC-based Given rotations algorithm, a MATLAB simulation model is established, which is completely consistent with the hardware implementation of the proposed QR decomposition systolic array architecture. Furthermore, $100,000$ random input matrices A are simulated as test vectors in the MATLAB simulation model. In order to determine the appropriate fractional precision of the intermediate word length, *FB*-bit, we increase the value of *FB*-bit from 18-bit to 28-bit with the 2-bit step, and then evaluate the corresponded maximum absolute error $\varepsilon_{\max}$, the mean absolute error ε_{mean}, and the standard deviation of the absolute error ε_{std}, and the number of worst cases of the QR decomposition results under the each condition. Notes that the worst cases represent the number of test matrices that can not achieve the required accuracy 2^{-13}. From the error evaluation results given in Table II, it is evident that for the results of matrix R and Q, when *FB*-bit are equal to 18-bit and 20-bit, there are average 97.4% and 99.7% out of $100,000$ random test cases can achieve the required accuracy 2^{-13} respectively. The maximum errors of worst cases have a similar dynamic range of $2^{-7} < \varepsilon_{\max} < 2^{-8}$. When *FB*-bit increases from 22-bit to 28-bit, there are 99.9% out of $100,000$ random test cases can achieve the required accuracy. The maximum errors of worst cases have a similar dynamic range of $2^{-9} < \varepsilon_{\max} < 2^{-10}$. Both the mean absolute error and the standard deviation of the absolute error are in the range of $2^{-14} < \varepsilon_{mean/std} < 2^{-15}$, which decrease about 5 times when *FB*-bit increases from 18-bit to 28-bit. The error evaluation results indicate that the proposed systolic array can satisfy 99.9% correct 4×4 QR decomposition for the 2^{-13} accuracy requirement when the word length of the data path (*FB*+3-bit) is lager than 25-bit.

C. CORDIC-based Process Unit

Fig. 2 shows the implementation of the proposed process units (PU_{BC} and PU_{IC}) in the proposed systolic array architecture. The process units are implemented based on the proposed CORDIC algorithm as shown in equations (17) and (18), by which the multiplication of the inverse of the scale factor $\frac{1}{K}$ can be compensated by merging the 11 scaling operations into the CORDIC iterations. The only different part between the implementation of the PU_{BC} and PU_{IC} is that PU_{BC} needs a selection unit (highlighted in dashed line in Fig. 2) to generate the digit sequence $\{\sigma[1], ..., \sigma[j]\}$, which are sent to the PU_{IC} as the inputs.

In Fig. 2, each CORDIC iteration can be performed by one clock cycle. In each clock cycle, first, $x[j]$ and $y[j]$ are selected by 2-to-1 multiplexors with the selection signal (*con1*). Second, the values of $x[j]2^{-k}$, $x[j]2^{-k-j}$, $x[j]2^{-j}$, $y[j]2^{-k}$, $y[j]2^{-k-j}$ and $y[j]2^{-j}$ are computed in parallel by the barrel shifters. At the same time, the digit $\sigma[j]$ is achieved based on the logical-XOR operation between the sign bits of $x[j]$ and $y[j]$. Third, since the value of $\sigma[j]$ can be 1 or -1, $\sigma[j]$ acts as the selection signal of a 2-to-1 multiplexer to choose the positive or negative value (achieved by 2's complement converter) of $\sigma[j]2^{-j}y[j]$. Meanwhile, the positive or negative value of $\sigma[j]\sigma_k 2^{-j-k}$ is selected in a 2-to-1 multiplexer by the selection signal (*con2*) that is generated based on the logical-XOR operation between $\sigma[j]$ and σ_k. Forth, in order to achieve $x[j+1]$ and $y[j+1]$, the *FB*+3-bit 4-to-2 carry-save adders (CSA) are applied to compute the

TABLE II

COMPUTATIONAL ERROR AND HARDWARE PERFORMANCE EVALUATIONS OF QR DECOMPOSITION WITH DIFFERENT WORD LENGTHES.

Error evaluation	**Matrix R**						**Matrix Q**					
Word length (*FB*-bit)	18	20	22	24	26	28	18	20	22	24	26	28
Max. absolute error (10^{-3})	6.32	5.28	1.42	1.32	1.32	0.92	6.77	5.80	1.76	1.60	1.47	1.58
Mean absolute error (10^{-5})	5.39	1.99	1.41	1.34	1.32	1.33	5.09	1.69	1.12	1.06	1.06	1.06
Stand. deviation (10^{-5})	5.47	2.87	1.16	1.13	1.11	0.95	5.85	2.95	1.30	1.26	1.24	1.24
Worst cases (*No.*)	2910	312	109	95	94	93	3424	266	134	113	111	109
Device	**Xilinx Virtex5 XC5VTX150T**											
Word length (*FB*-bit)	18		20		22		24		26		28	
No. of occupied slices LUTs (%)	7,811 (8%)		8,593 (9%)		9,122 (9%)		9,758 (10%)		10,512 (11%)		11,513 (12%)	
No. of occupied slices (%)	2,609 (11%)		2,684 (11%)		2,810 (13%)		3,070 (13%)		3,379 (14%)		3,617 (15%)	
Critical path delay (*ns*)	8.70		8.78		8.82		8.97		9.21		9.27	
Processing time (*us/update*)	0.469		0.474		0.476		0.484		0.497		0.500	
Throughput (*M/sec*)	2.13		2.11		2.10		2.06		2.01		2.00	

sum and carry of $x[j+1]$ and $y[j+1]$, and then they are added together in the carry propagate adders in parallel to achieve the final results. Fifth, the $x[j+1]$ and $y[j+1]$ are sent back to registers for the next CORDIC iteration. After number of 18 clock cycles, the final results of the CORDIC-based process units are achieved.

IV. IMPLEMENTATION AND COMPARISON RESULTS

The proposed systolic array architecture for the 4×4 QR decomposition is modeled with Verilog and implemented on Xlinix Virtex5 XC5VTX150T FPGA configuration. The proposed architecture is synthesized with XST and placed and routed by Xilinx ISE 11.1. The clock, input signals, and output signals are assumed to be ideal, which occupies 1 out of 16 GCLK I/O blocks, 154 out of 680 I/O blocks. Inputs and outputs of the proposed architecture are registered and the design is optimized for delay.

A. Hardware Performance Evaluation

In order to achieve a comprehensive hardware performance evaluation, the resource utilized and speed achieved on FPGA are evaluated with the different precision of the intermediate word lengthes. The hardware implementation results as shown in Table II indicate that 1) the number of occupied slices LUTs by the proposed architecture linearly increases from $7,811$ to $11,513$ out of 92,800 which occupies from $2,609$ to $3,617$ out of 23,200 slices; 2) the critical path delay linearly increases from 8.70 *ns* to 9.27 *ns* respectively, when *FB*-bit increases from 18-bit to 28-bit. The critical path of the proposed architecture is located in the CORDIC-based process unit, which is highlighted in Fig. 2 (dotted line), and the details of the architecture with $FB = 22$-bit (including the net delay) are reported in Table III. It is evident that most two timing consumed blocks are the barrel shifter and the binary adder in the proposed architecture. By the proposed architecture, except that the first result needs to be computed after 80 clock cycles, the other QR decomposition results can be achieved in every 54 clock cycles. Moreover, since the average maximum frequency is about 111 *MHz*, therefore, considering the throughput of the proposed architecture, it can achieve 2.06 million updates in every second.

TABLE III

DETAILS OF CRITICAL PATH DELAY ON FPGA.

Blocks in the critical path							**Total**
Reg	FSM	Mux	Shift	2'sCom	Mux	Add	(*ns*)
1.01	0.90	0.82	1.82	0.81	0.82	2.64	8.82

B. Function Verification Platform

The function verification platform for verifying the proposed QR decomposition systolic array is implemented in Xilinx University Program Virtex5 XUPV5LX110T Development System [19] with Embedded Development Kit (EDK). Fig. 3 shows the architecture of this function verification platform.

The proposed verification methodology is created in *MicroBlaze* with *C* language. First, the random input matrix as the test vectors are sent to the proposed QR decomposition systolic array by *PLB* bus. The QR decomposition results achieved by the proposed architecture are sent back to *MicroBlaze* by *PLB BRAM* and *PLB* bus. Meanwhile, the same test vectors are computed by *MicroBlaze* to achieve the accurate single precision binary floating-point precision results as the benchmarks. Finally, the QR decomposition results achieved by the proposed systolic array are compared with these accurate results obtained by *MicroBlaze*. If they are not identical, the corresponding test vector will be sent to the personal computer through *RS232 UART* for analyzing the the computational error. It is difficult to verify all the test vectors due to the infinite processing time in this verification platform, therefore, 100,000 random test vectors are chosen and sent to this verification platform. The verification results show that the hardware implementation of the proposed systolic array can guarantee a faithful computation for the QR decomposition.

C. Comparison Results

The implementation results of the proposed systolic array architecture are compared with the design [17], because 1) the same CORDIC-based Givens Rotations algorithm is used

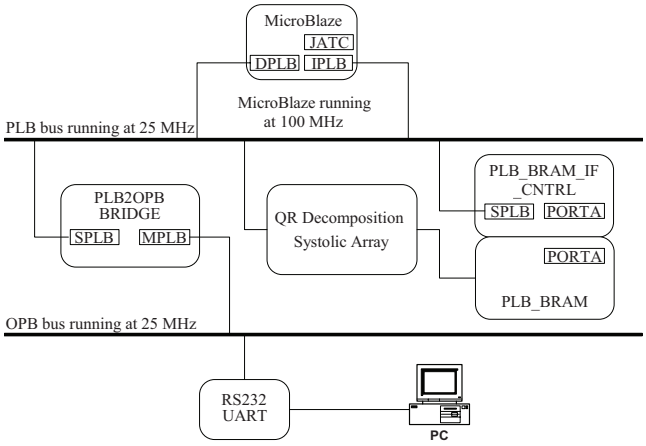

Fig. 3. Function verification platform for the proposed systolic array.

TABLE IV
COMPARISON OF THE IMPLEMENTATION RESULTS.

Works	**Ref. [17]**	**This work**
No. occupied logic elements (%)	$22,438$ (54%)	$28,458$ (69%)
Max. frequency(*MHz*)	75	60
No. clock cycles	73	54
Processing time (*us/update*)	0.97	0.90
Throughput (*M/sec*)	1.03	1.11

in both designs; 2) the direct mapping technology and fixed-point data-path are adopted in both designs. Since the QR decomposition architecture in [17] is implemented on Altera Stratix EP1S40 FPGA, in order to conduct a fair comparison, the proposed architecture is synthesized using the same devices. The implementation results are shown in Table IV which indicate that the proposed architecture occupies about $28,458$ (69%) logic elements, computes every update in about 0.90 *us*, and achieves about 1.11 *M/sec* updates by running at the maximum frequency 60 *MHz*. Compared with the design [17], the proposed architecture can achieve 1.08 times higher throughput than that in [17], with the expense of 1.26 times more area. The more logic elements occupied by the proposed architecture is because that the proposed architecture approximate the 20-bit inverse of the scale factor ($\frac{1}{K} \approx 0.607252935$) instead of the 3-bit factor ($\frac{1}{K} \approx 0.875$) adopted in [17], so that the proposed design can achieve the more accurate results than that by [17].

V. CONCLUSIONS

In this paper, first, we review the CORDIC-based Givens rotations algorithm. Second, we modify the CORDIC algorithm to compensate the inverse of the 20-bit scale factor by merging scaling operations into CORDIC iterations. Third, we design a parallel architecture of the QR decomposition systolic array that allows a direct mapping by 21 CORDIC-based process units with the fixed-point data-path. Forth, the implementation results of the proposed architecture are analyzed and verified on FPGA. In order to achieve a comprehensive resource and performance evaluation, the error analysis, the resource utilized, and speed achieved on FPGA, are evaluated with the different precision of the intermediate word lengthes. The results indicate that 1) the proposed architecture can satisfy 99.9% correct 4×4 QR decomposition for the 2^{-13} accuracy requirement when the word length of the data path ($FB+3$-bit) is lager than 25-bit; 2) the resource utilized, and speed achieved on FPGA linearly increase with the intermediate word length. Compared with the design [17], the proposed design can achieve a higher throughput and more accurate results for the QR decomposition.

ACKNOWLEDGMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] J. R. Rice, Experiments on Gram-Schmidt orthogonalization, J. Mathematics of Computation 20 (94) (1966) 325–328.
[2] A. S. Householder, Unitary triangularization of a nonsymmetric matrix, J. the ACM 5 (4) (1958) 339–342.
[3] W. Givens, Computation of plane unitary rotations transforming a general matrix to triangular form, J. the Society for Industrial and Applied Mathematics 6 (1) (1958) 26–50.
[4] W. M. Gentleman, H. T. Kung, Matrix triangularization by systolic arrays, In Proc. SPIE Real-Time Signal Processing IV 298 19–26.
[5] S. Y. Kung, VLSI array processors, IEEE ASSP Mag. 2 (3) (1985) 4–22.
[6] C. M. Rader, VLSI systolic arrays for adaptive nulling, IEEE Signal Processing Mag. 13 (4) (1996) 29–49.
[7] W. M. Gentleman, Least squares computations by givens transformations without square roots, J. Institute of Mathematics and its Applications 12 (3) (1973) 329–336.
[8] S. F. Hsieh, K. Liu, K. Yao, A unified square-root-free approach for qrd-based recursive-least-squares estimation, IEEE Trans. on Signal Processing 41 (3) (1993) 1405–1409.
[9] J. Gotze, U. Schwiegelshohn, A square root and division free givens rotation for solving least squares problems on systolic arrays, SIAM J. Scientific and Statistical Computing 12 (4) (1991) 800–807.
[10] J. E. Volder, The CORDIC trigonometric computing technique, IRE Trans. Electron. Comput. EC-8 (1959) 330–334.
[11] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, K. Maharatna, 50 years of CORDIC: algorithms, architectures, and applications, IEEE Trans. on Circuits and Systems-I 56 (9) (2009) 1893–1907.
[12] R. L. Walke, R. W. M. Smith, Architectures for adaptive weight calculation on ASIC and FPGA, in: In Proc. 33rd Asilomar Conf. on Signals, Systems and Computers, 1999, pp. 1375–1380.
[13] A. Sergyienko, O. Maslennikov, Implementation of Givens QR-decomposition in FPGA, Springer, Berlin, 2002, lecture Notes in Computer Science.
[14] D. Boppana, K. Dhanoa, J. Kempa, FPGA based embedded processing architecture for the QRD-RLS algorithm, in: In Proc. 12th IEEE Symp. on Field-Programmable Custom Computing Machines (FCCM'04), 2004, pp. 330–331.
[15] M. P. Fitton, S. Perry, R. Jackson, Reconfigurable antenna processing with matrix decomposition using FPGA-based application specific integrated processors, in: In Proc. ACM Software Defined Radio Technical Conf., 2004.
[16] F. Sobhanmanesh, S. Nooshabadi, Parametric minimum hardware QR-factoriser architecture for V-BLAST detection, IEE Proc. Circuits, Devices and Systems 153 (5) (2006) 433–441.
[17] Y. Y. K. M. A. Hiroyuki, Resource and performance evaluations of fixed point QRD-RLS systolic array through fpga implementation, IEICE Trans. on Communications (4) (2008) 1068–1075.
[18] M. Sima, M. Mcguire, Embedded reconfigurable solution for OFDM detection over fast fading radio channels, in: In Proc. IEEE Workshop on Signal Processing Systems (SiPS'07), 2007, pp. 13–18.
[19] Xilinx Inc., Xilinx University Program XUPV5-LX110T Development System, Hardware Reference Manual, 2009.