Design and implementation of 32-Bits MIPS processor to Perform QRD Based on FPGA

[1] Safaa S. Omran , [2] Ahmed K. Abdul-abbas
[1,2] Electrical Engineering Technical College Baghdad/Iraq
EMAIL : [1] omran_safaa@ymail.com , [2] ahmed89kareem@alkafeeluc.edu.iq

Abstract—**The QR decomposition (QRD) is an important prerequisite for many different applications such as multiple input multiple output (MIMO) detection in wireless communication system, matrix inversion, radar application and so on. This paper presents the design and implementation of a 32-bit MIPS (Microprocessor without Interlocked Pipeline Stage) processor based on QRD using Givens Rotation algorithm. Moreover, this design contains a developed hardware in order to reduce the latency of QRD process. The implementation results of this QRD processor are compared with another work. Finally, design, synthesis of this QRD processor has been achieved using Xilinx 14.7 ISE simulator and coding is written in Verilog HDL language and implemented on Virtex – 7 FPGA (Field Programmable Gate Array) kit.**

Keywords—QRD; givens rotation; RISC processor.

I. INTRODUCTION

Matrix processing and factorization problems are common to a wide set of communication system, image processing and signal processing applications such as filtering, adaptive beam forming, system identification, MIMO system and etc. [1] [2] [3]. In addition, QR decomposition is used in space-time adaptive processing (STAP) which can be used to extract signals well below the noise floor [4].

Let A be an m × n matrix with full column rank. The QR factorization of A is A = QR, where Q is an m×m orthogonal matrix and R is an m × n upper triangular matrix. There are different algorithms to compute this decomposition:

1 - Gram-Schmidt orthogonalization.

2 – Householder transformation.

3 - Givens rotation.

Gram Schmidt algorithm (GS) offers reduced stability and accuracy in fixed point implementations. Householder transformation algorithm (HH) suits well for dense matrices but it is difficult to carry out parallel implementations since it is working on all elements of column each time. Givens rotation algorithm (GR) has capability of selectively annihilating individual matrix element [2].

Many researchers were interested in QR decomposition based on FPGA. Dongdong Chen and Mihai Sima proposed a parallel architecture of an QR decomposition systolic array based on the GR algorithm (Givens rotation algorithm) on field programmable gate array, takes on the direct mapping by 21-fixed-point CORDIC based process units (PU) that can compute the QRD for a 4 × 4 real matrices [5]. Semih Aslan, et.al. proposed another work to compute QRD, they developed the architecture for QRD using the GR algorithm, based on Coordinate Rotation Digital Computer algorithm (CORDIC) and fixed point calculations, is optimized for FPGA platforms and the design can run at 246MHz as maximum speed [6]. High Throughput hardware design has been proposed by Sergio D. Munoz and Javier Hormigo, utilizing GR algorithm. They used a new two dimensional systolic-array architecture with pipelined-processing components. which are based on the CORDIC algorithm where they propose to use two dimensional array architecture, where each processing elements works with all the elements of the same row and these processing elements are pipelining [7]. while Ali Umut Irturk proposed a different design, a processor architecture has been proposed to perform QRD process. This processor architecture consists of two units to perform addition and subtraction operations, two units to perform multiplication and division operations and one unit to perform square root operation by using newton iteration method [8].

The main focus in this paper is on building Givens rotation algorithm using 32-bits MIPS processor which is designed especially for this purpose.

And a design of special circuit used for matrix multiplication, in which to reduce the number of clock cycles required for matrix multiplication. The 32-bits RISC processor designed and implemented using Verilog HDL language based on virtex – 7 FPGA kit.

The organization of this paper is as follows: Section two provides an overview discussion on QRD by using Givens rotation algorithm and a brief for mathematic procedure. Section three contains an explanation for the design of 32-bits MIPS processor. Section four presents how to address the problem of matrix multiplication by addition hardware. Section five, results and timing are discussed. finally, the conclusion given in the last section.

II. GIVENS ROTATION ALGORITHM

The QR decomposition can be computed using Givens rotation [9]. Givens Rotation algorithm eliminates one element in a matrix one at a time.

If matrix A consists of, m x n elements, to compute the QRD by Givens Rotation, the first step, is to use $a_{31}$to eliminate

978-1-5386-7858-9/18/$31.00 ©2018 IEEE

a_{41} and by using equation (1) then, computing Cosθ and Sinθ using equation (2) and (3) respectively.

$$r_{i,j} = \sqrt{a_{ik}^2 + a_{jk}^2} \text{ ………………………..……… (1)}$$

$$\text{Cos}\theta = \frac{a_{ik}}{r_{i,j}} \text{……………………………………..…….. (2)}$$

$$\text{Sin}\theta = \frac{a_{jk}}{r_{i,j}} \text{………….………………….…....…….. (3)}$$

Where: a_{jk}, a_{ik} are elements of matrix A, i = 3, j = 4, k = 1.

Step two, generating G^i matrix (Givens rotation matrix) by I matrix with addition values of Cos and Sin into I matrix. Where $G_{33}^1 = Cos\theta,\ G_{34}^1 = Sin\theta, G_{43}^1 = -Sin\theta$ and $G_{44}^1 = Cos\theta.$ As shown on fig.- 1.

$$\mathbf{G}_{3,4}^{(1)} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \cos\theta_{3,4} & \sin\theta_{3,4} \\ & & -\sin\theta_{3,4} & \cos\theta_{3,4} \end{pmatrix}$$

Fig. 1. $G_{3,4}^1$ matrix.

Step three, computing first rotation using equation (4), when multiplying G^1 and A^0, produce A^1.

$$A^i = G^i * A^{i-1} \text{ …………………………………….. (4)}$$

Then repeating this process five times to compute R matrix. Fig. 2 shows these process, and fig. 3 shows the givens rotation algorithm [9].

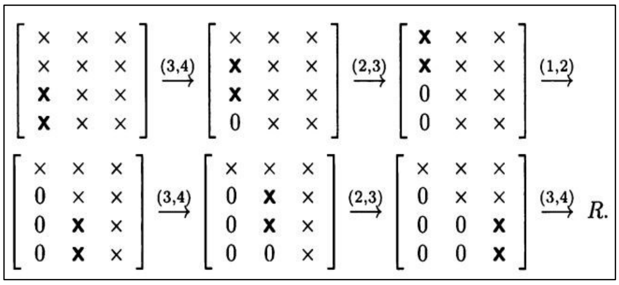

Fig. 2. steps to compute R matrix from A (4x3).

```
for $j = 1{:}n$
    for $i = m{:} - 1{:}j+1$
        $[c, s]$ = givens$(A(i-1, j), A(i, j))$
        $A(i-1{:}i, j{:}n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A(i-1{:}i, j{:}n)$
    end
end
```

Fig. 3. Algorithm to find R matrix.

After finding R matrix, equation (5) is used to compute Q (orthogonal matrix) from G matrices. In this step, the problem of matrix multiplication is appearing and this will be explained clearly in section four.

$$Q = (G^i G^{i+1} G^{i+2} \ldots\ldots G^m)^{-1} \text{ ………………………… (5)}$$

III. Designed Processor

Specific architecture design has been proposed to compute QR decomposition by Givens rotation algorithm. A 32-bit MIPS processor single cycle has been designed, based on FPGA processor [10] [11], fig. 4 shows the internal architecture of this design. It is consisting from the following units:

1. Program counter unit.
2. Instruction memory unit.
3. Data memory unit.
4. Register file unit.
5. Arithmetic unit.
6. Bus interface unit.

Program counter unit (PC), It is a 32-bit register used to store 32-bits address of memory location from which an instruction to be fetched. After every instruction fetched (every clock cycle) it is automatically incremented by 4 (depending on instruction size, which is four byte) so as to read the next instruction.

Instruction and data memory units, in single cycle processor the memory divided into two parts, first one to store the instructions and the second part to store the data, this arrangement enables programmer to read instruction and data at the same time or in the same cycle, this type of design is known as Harvard memory architecture [12]. In this processor the memory has been designed as four way interleaving and the size of instruction memory is 1K byte while the size of data memory is 256 bytes.

Register file unit (Regfile), this unit used for load and store operations, the register file unit consists of 32 registers and each register consists of 32 bits. As shown in figure-4, this unit also contains four input ports, the length of three of these four ports are 5 bits, these ports assigned as A1, A2 and A3. While the other port is 32-bits length which is used to move the data inside it. In addition to these ports, there are another two output ports, the length of each one is 32 bits (RD1, RD2). Table -I shows the details for these ports.

Arithmetic unit (AU), this unit performs all arithmetic operations on maximum 32 bits at a time. The arithmetic unit consists of different sub units, addition / subtraction subunit (add/sub), multiplication subunit (mul), division subunit (div) and square root subunit (sqrt). Each sub unit can perform its operation during one clock cycle only. Following are details about these subunits:

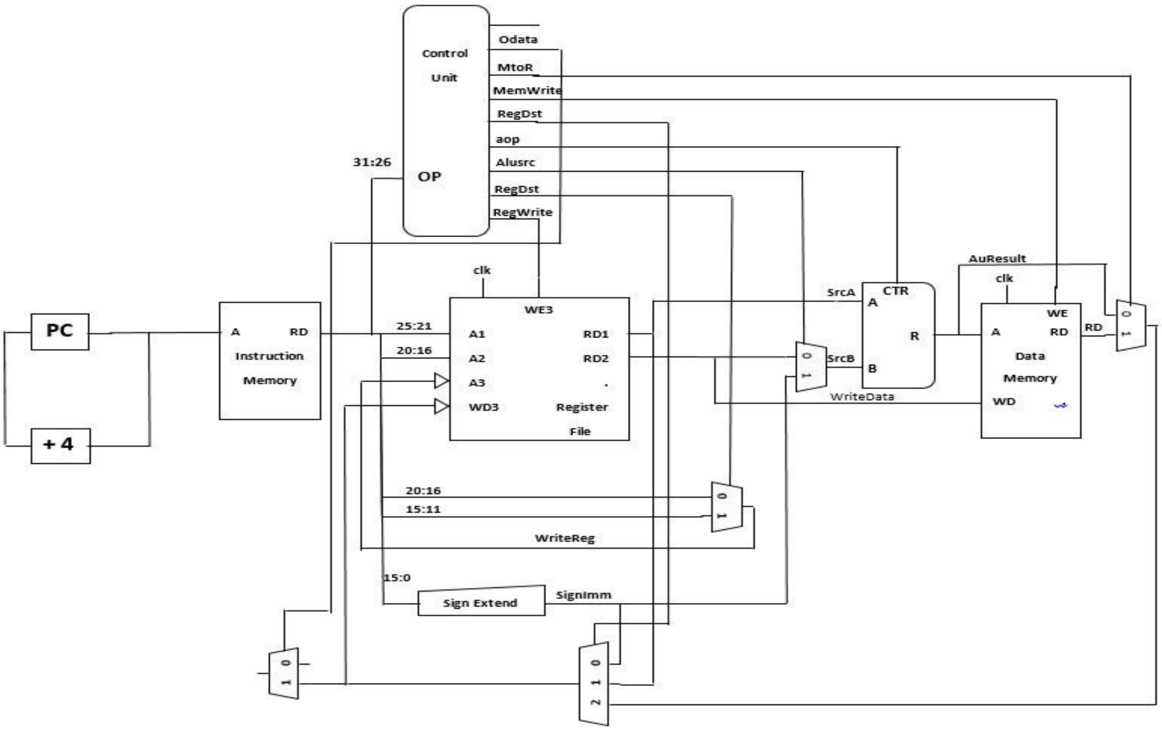

Fig. 4. Internal architecture of the designed MIPS processor.

TABLE I. PORTS OF REGFILE.

port	size	type	description
A1	5 bits	input	To select a source reg.
A2	5 bits	input	To select a destination reg.
A3	5 bits	input	To select a destination reg.
WD3	32 bits	input	to move a data to it
RD1	32 bits	output	to move a source1 data from it
RD2	32 bits	output	to move a source2 data from it

A) Addition and subtraction sub unit

Parallel full adder and parallel full sub tractor circuits have been used to design this sub unit. Building this subunit is easier than other subunits in arithmetic unit. Fig.5 shows the block diagram for this subunit.

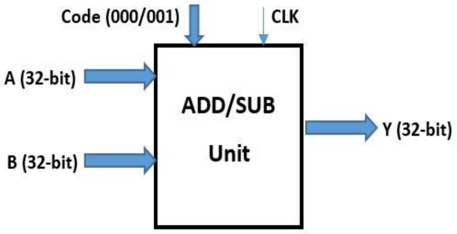

Fig. 5. Addition and subtraction sub unit.

B) Multiplication sub unit

There are several types of FPGA kits, some of them include an embedded hardware multiplication which is well suited for DSP application and their kinds of FPGA kits like (virtex 4) are able to perform a multiplication operation in one clock. The other type of FPGA kits which not supported with this hardware, needs to build a binary multiplier, which needs more than one clock cycle to do the multiplication process. In this paper the FPGA kit virtex-7 is used which supports a hardware unit for multiplication process.

C) Division sub unit

The “Xilinx LogiCORE™ IP Divider” Generator core creates a circuit for integer division based on Radix 2 non - restoring division, or High Radix division with pre-scaling. The Radix 2 algorithm exploits FPGA logic to achieve a range of throughput options that includes single cycle, and the High Radix algorithm exploits DSP slices at lower throughput, but with reuse to reduce resources. The virtex 7 device family support this option, it consumes 593 number of slice units.

D) Square root sub unit

The mat lab program is used to solve square root problem, the mat lab uses exact method to find the result of square root, and convert it to combinational code. It supports the HDL coder since version 2014A of mat lab. In using this method, the first step is to select the input port from sources tab and sets the properties (set a type of input port to fixed point (1,32,20), where 1 refers to sign bit, 32 refers to length of number (integer part = 11 = 32 – 20 – 1) and 20 refers to length of fraction part), the second step is, to select the output port from sink tab and then set its properties (set a type of input port to fixed point (1,32,20)). The third step, to select the desired function from math operation, sort function and then also set the properties of function (fixed point (1,32,20)). Last step is, to generate HDL code from Manu-par "code." The negative side in this way is the high consumption of the resources of FPGA kit.

Finally, there are two different types of representation to the real numbers: fixed point and floating point numeration systems. The fixed point arithmetic is an extension of the integer representation which allows us to represent relatively reduced range of numbers with a constant absolute precision as well as a low complexity in implementation. So, it has been used in this design with a word length equal to 32 bits divided to 20 bits as fraction point and 11 bits as integer part and last bit as sign number.

IV. Direct multiplication of matrix sub unit

If (A) and (B), two dimensional matrices (4x4 elements), to find the result of matrix multiplication it needs 64 multiplication operations and 48 addition operations, i.e., each element needs 4 multiplication operations and 3 addition operations, so, the total number of operations equal to 112 (7 * 16). This design of 32-bits MIPS processor can be performing any arithmetic operation during one clock cycle. So 112 clock cycle have been needed to perform multiplication of matrix. To compute QR decomposition by using Givens Rotation, five times of multiplication of matrix have been needed to find the orthogonal Q matrix, where the total clock cycles are equal to 560 (5 * 112).

This proposed design for 32-bits MIPS processor containing additional hardware (direct multiplication of matrix) to compute the multiplication of two matrices during one clock cycle, this hardware is consisting of cells to perform this operation, each cell performs the multiplication operation of one element. Each cell consists of four multipliers and three adders, and these cells are connected in parallel; fig. 6 shows the block diagram of these cells. That means, the total number of multipliers for this hardware are 64, and the total number of adders are 48.

Since the G matrix is identity matrix with values of sin and cos, as shows in fig. 1, that means, in each operation of matrix multiplication there is two rows without change, so, the number of multipliers and adders can be reduced to two multipliers and one adder in each cell, fig.7 shows the final design for cells.

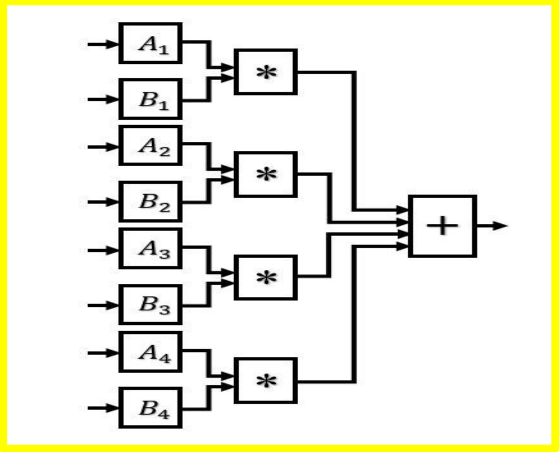

Fig. 6. Cell multiplication.

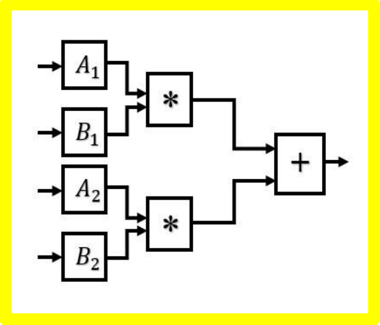

Fig. 7. Modified cell multiplication.

The challenge of our additional hardware is a data transfer from the register file unit to direct multiplication of matrix. In this case, it is required to transfer 32 elements (16 elements for each matrix), the transfer of data operation for one element needs one clock cycle and then needs 16 clock cycle to transfer the result of multiplication (16 elements) and one clock to start the operation. So, the total clock cycles for one operation equal to 49 clock cycle (32 + 16 + 1). But, if this hardware is built inside the register unit, in this case there is no need to transfer data between them, and one clock cycle is sufficient the operation of matrix multiplication. Fig.8 shows the block diagram of normal register unit.

The design of the direct multiplication of matrix to be reside inside the register file unit requires an addition of a new control signals to do this operation. First one, the position control signal (pos) to determine the position of sin and cos in G matrix, there are three positions, so, needs two bits for this signal. The second signal is rom (register or matrix) to determine the output data from registers or direct multiplication of matrix. And the last signal is smm (start matrix multiplication) to start the operation of matrix multiplication, fig. 9 shows the block diagram of modified register unit.

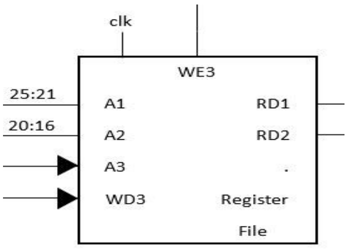

Fig. 8. Register File unit.

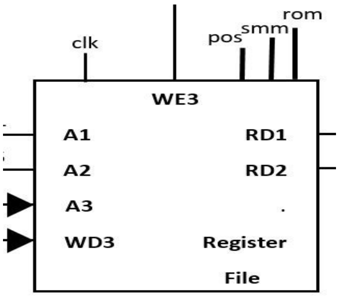

Fig. 9. Modified Register File unit.

V. Results

The designed processor is implemented using FPGA development board virtex-7 running with frequency 503MHz. An input matrix A is used for testing the designed processor as shown below:

$$A = \begin{bmatrix} 1 & -1 & 4 \\ 1 & 4 & -2 \\ 1 & 4 & 2 \\ 1 & -1 & 0 \end{bmatrix}$$

The factorization analysis of A for orthogonal matrix Q and upper triangular matrix R were given as below:

$$Q = \begin{bmatrix} 0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & -0.5 & -0.5 \end{bmatrix}$$

$$R = \begin{bmatrix} 2 & 3 & 2 \\ 0 & 5 & -2 \\ 0 & 0 & 4 \end{bmatrix}$$

Fig.11 shows the test bench simulation for orthogonal matrix Q and upper triangular matrix R for input matrix A. As shown from the figure it is clear that the time required for finding the decomposition of matrix A is 420 ns (0.42 µs). That means, the throughput of this processor was 2.38M matrix per second.

The required instructions for the different mathematical operations (add/sub, div/mul, sqrt) are 175 instructions, 169 instructions to calculate the upper triangular matrix and 5 instructions to calculate the orthogonal matrix.

Table II shows a comparison between this work and other works [6] [7] [8] according to size of matrix, word length, latency, type of design and FPGA board. From table II the latency of [7] is less than the latency of this work, this is because ref [7] used a pipelining architecture, in our case as a future work to design a pipelining processor for computing QR of A which will reduced the latency to expected to less than 100 clock cycle.

TABLE II. Final comparison.

Work	[6]	[7]	[8]	This Work
Matrix	4x4	4x4	4x4	4x3
Word	16	32	20	32
Pipelined	Non	Pip	Non	Non
Type	CORDIC	CORDIC	Processor	Processor
Latency	180	140	-	175
FPGA board	Virtex 5	Virtex 5	Virtex 4	Virtex 7

VI. Conclusion

This paper presents a fixed-point MIPS processor with additional hardware to achieve high speed QR Decomposition. The additional hardware is consuming 18425 number of slices, 4 number of bonded IOBs, 3 number of GCKs and 32 from DSP48s. In the other side, this hardware can reduce the number of clock cycles to perform a multiplication of matrix from 560 to 6 clock cycle only. And this design capable to decompose a matrix with any dimension by editing the software program for the processor.

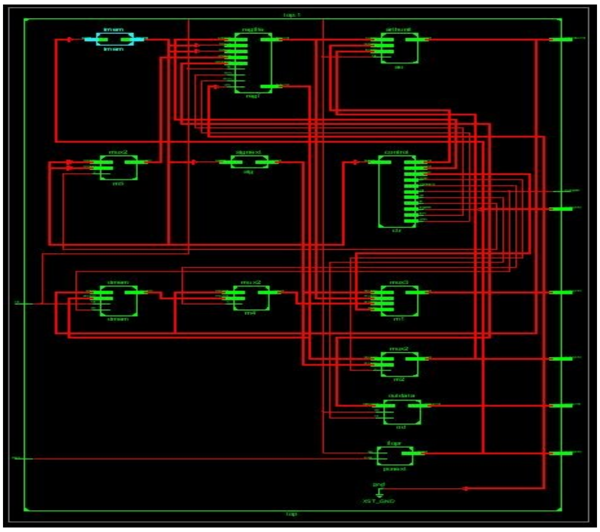

Fig. 10. RTL schematic for GRP 32-bit.

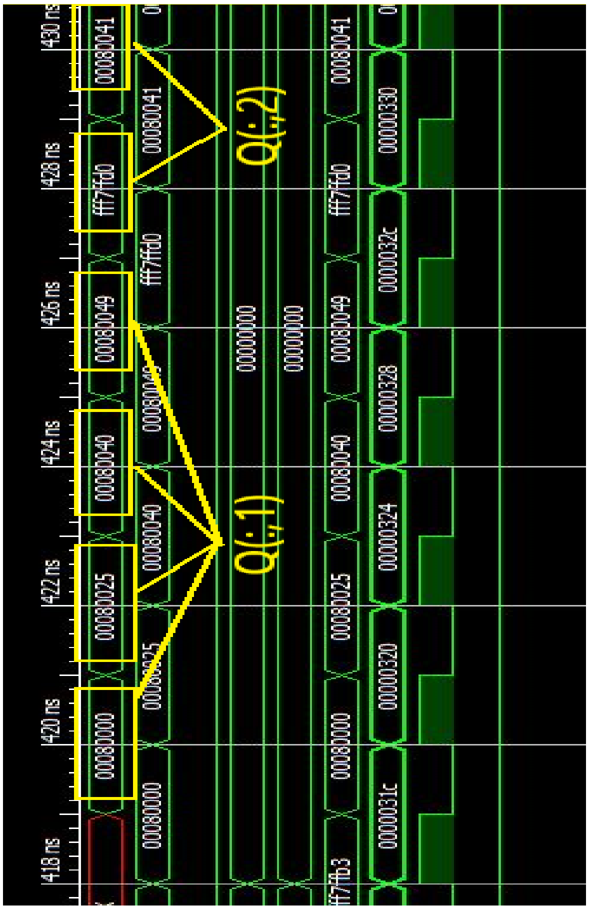

Fig. 11. simulation of result.

References

[1] P. Luethi, "Gram-schmidt-based QR decomposition for MIMO detection: VLSI Implementation and comparison," in IEEE Asia Pacific Conference on Circuits and Systems, Macao, China, 2008.

[2] G. R. Prabhu, "FPGA based scalable fixed point QRD core using dynamic partial reconfiguration," in 2015 28t International Conference on VLSI Design, IEEE, Bangalore, India, 2015.

[3] R. Gangarajaiah, O. Edfors and L. Liu, "An adaptive QR decomposition processor for carrier-aggregated LTE-A in 28-nm FD-SOI," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 7, pp. 1914 - 1926, 2017.

[4] M. Parker, V. Mauer and D. Pritsker, "QR decomposition using FPGAs," in EEE National Aerospace and Electronics Conference and Ohio Innovation Summit, Dayton, OH, USA, 2016.

[5] D. C. a. M. SIMA, "Fixed-point CORDIC-based QR decomposition by givens rotations on FPGA," in Reconfigurable Computing and FPGAs, Canada, 2011.

[6] S. Aslan, E. Oruklu, and J. Saniie, "Realization of area efficient QR factorization using unified division, square root, and inverse square root hardware," in IEEE International Conference on Electro/Information Technology, Canada, 2009.

[7] S. D. Munoz and J. Hormigo, "High-throughput FPGA implementation of QR decomposition," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 6, no. 9, p. 1, 2015.

[8] A. U. Irturk, "Implementation of QR decomposition algorithm using FPGAs," Univesity of california, California/santa barbara, 2007.

[9] G. H. Golub and C. F. V. Loan, Matrix comutations, Baltimore, Maryland: John Hopkins Univ. Press, 2013.

[10] S. S. Omran and A. J. Ibada, "FPGA implementation of MIPS RISC processor for educational purposes," Journal of Babylon University, vol. 24, no. Pure and applied sciences, pp. 1745-1761, 2016.

[11] S. S. Omran and H. S. Mahmood, "Hardware modelling of a 32-bit, single cycle RISC processor using VHDL," in ICIT 13 The IEEE 6th International Conference on Information Technology, Jordan, 2013

[12] S. S. Omran and H. S. Mahmood, "Pipelined MIPS processor with cache controller using VHDL implementation for educational purposes," in IEEE International Conference on Electrical Communication, Computer, Power, and Control Engineering, Mosul, Iraq, 2014.