2024 Spring VLSI DSP Homework Assignment 4

Due date: 2024/05/28

1. Introduction

In this assignment, you are asked to develop a hardware QR factorization module. Given a M×N matrix, the QR factorization convert it to the product of a unitary matrix Q and an upper triangular matrix R.

2. QR factorization scheme

Assume matrix $\mathbf{A}_{8\times4}$, you may apply a sequence of Givens rotation to convert it into an upper triangular matrix $\mathbf{R}_{8\times4}$ with the lower half as a zero matrix. However, to obtain the $\mathbf{Q}_{8\times8}$, you need extra computations to obtain the product of these Givens rotations. An easy way to accomplish it is augmenting matrix $\mathbf{A}_{8\times4}$ with an identity matrix $\mathbf{I}_{8\times8}$ and updating the identity matrix along with every Givens rotation applied to matrix $\mathbf{A}_{4\times4}$.

$$\mathbf{Q_n}\times\mathbf{Q_{n-1}}\times\cdots\times\mathbf{Q_2}\times\mathbf{Q_1}\times[\mathbf{A}\,|\,\mathbf{I}]=[\mathbf{R}\,|\,\mathbf{Q}]$$
$$\mathbf{Q}=\mathbf{Q_n}\times\mathbf{Q_{n-1}}\times\cdots\times\mathbf{Q_2}\times\mathbf{Q_1}$$

For the triangularization part (**R** matrix calculation), a triangular systolic array structure as shown in the lecture note is needed. For **Q** matrix calculation, an additional rectangular array is needed to serve the purpose. This makes the array structure of the entire design a trapezoidal one (Fig. 1). Not shown in the figure includes a controller module (or FSM) to control the computations.

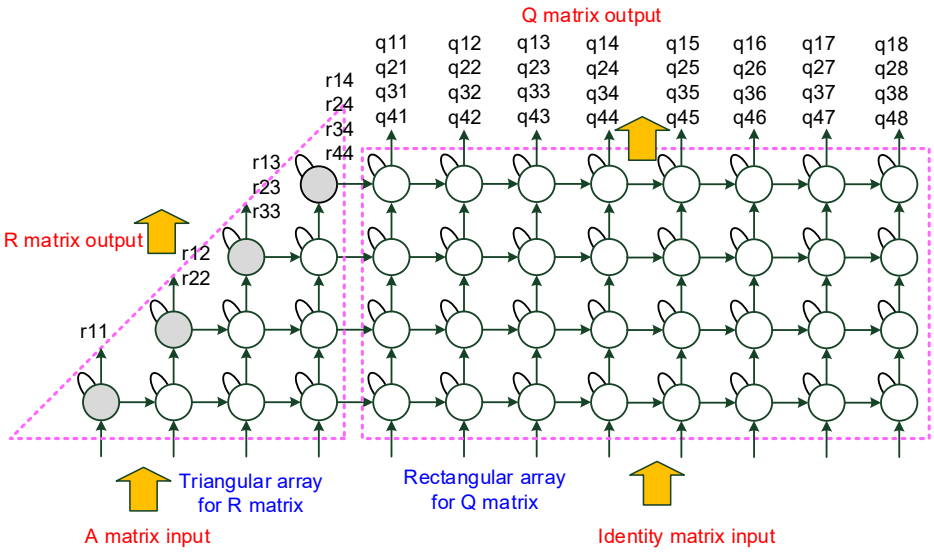

Figure 1. Trapezoidal array for QR factorization

3. Hardware design guidelines

- **CORDIC module**

  CORDIC module is employed to implement the Givens rotations. The word length is set

as 12 bits and you can determine how many bits are for the integral part and how many bits are for the fractional part so that no overflows occur. The iteration number is set as 12. To reduce the computing latency iterations, 4 iterations are performed in one clock cycle and it takes 3+1 (extra clock cycle for normalization) to complete one Givens rotation. An unfolded CORDIC architecture with two iterations per clock cycle is shown below.

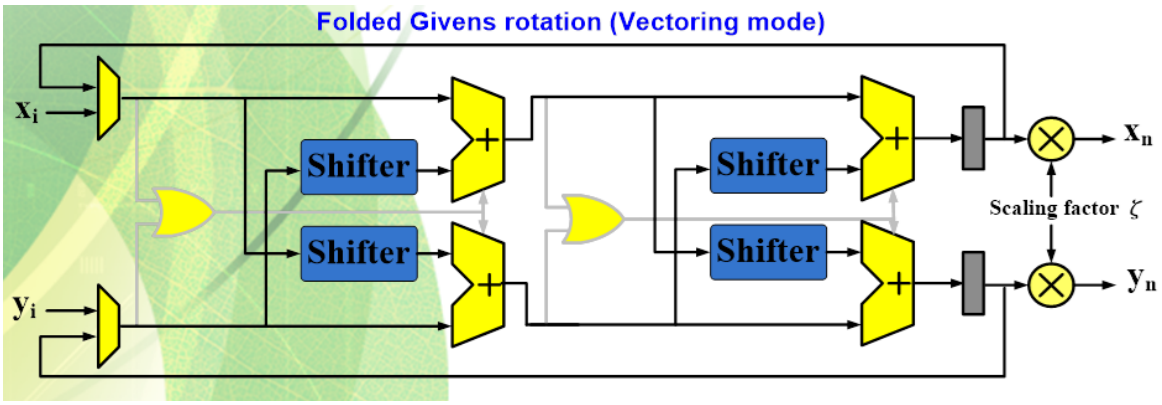

Figure 2. unfolded CORDIC processor design

Note that there is no need of computing the rotation angle θ explicitly. In each row of the CORDIC array, the leading one (performing in the vectoring mode) passes the rotation directions sequence, which consists of +1 or -1, to the trailing modules (performing in the rotation mode) in the same row. In other words, these trailing modules simply follow the sequence of micro-rotations performed in the leading one. Note that there are 4 rotation directions to be passed in each clock cycle because 4 iterations are performed.

- **I/O requirements**

As shown in Figure 1, matrix A is inputted from the bottom of the triangular array and an identity matrix is inputted from the bottom of the rectangular array. The resultant upper triangular matrix R can be obtained along the diagonal border of the triangular array. The Q matrix will reside on the rectangular array. Eight additional clock cycles are needed to propagate the results upward so that the Q matrix can be obtained from the top row of the rectangular array. Also assume the entries of input matrix A are all integral, signed and 8-bit wide. The entries of R and Q matrices are all 12-bit wide (you need to indicate how many bits are for integral part, and how many bits are for fractional part).

4. Design Merits

Please write the corresponding Verilog code and verify the correctness of the design through simulations. Prepare a fixed point MatLab code to generate the golden patterns of the result. Compare the Verilog simulation results with the golden patterns to prove the correctness of your design. For your verified design, please indicate

- Fixed point (finite precision) hardware implementation loss. This is obtained by calculating the Frobenius Distance (Euclidean Norm) of between a floating point

result matrix A and a fixed point result matrix B as

$\mathrm{F(A,B)} = \sqrt{trace((A-B)\cdot(A-B)^T}$

And then dividing the distance by the norm 2 value of matrix A

- Timing diagram of your design
- Clock cycles needed to complete one QR factorization
- The initiation interval of two successive QR factorizations

Hint: A suggested timing is as follows