

ELEC 522, Advanced VLSI
Rice University, Fall 2022
10 November 2021
Due: Monday 21 November 2022

Design Project 5

QR Decomposition using CORDIC arithmetic for a 4 x 4 matrix

Goal: Explore rapid prototyping systems for FPGA/ASIC synthesis along with signal processing applications using Xilinx System Generator and Vitis HLS.

As in the handouts on the QR decomposition, assume that a matrix of fixed-point values is given to you from a host ARM processor in the Zynq FPGA. The 4 x 4 matrix is transferred to your array possibly row by row. Produce the orthogonal matrix of rotation factors Q and the upper triangular matrix R. There are several solution approaches and array structures. Please refer to the literature and describe the benefits of your approach. Use Xilinx Model Composer to create the overall array timing structure.

However, instead of using the built-in Model Composer CORDIC block, create the CORDIC functions using Vitis HLS (from your Project 4) with then a Vitis HLS block into Model Composer. The use of Vitis HLS for the CORDIC datapath will allow you to take better advantage of the angle generation and angle application capabilities. You would use the same fixed point data format as in Project 4 for the Vitis HLS CORDIC block (You may need to increase the integer part.)

Your CORDIC modules should be the inverse tangent type that produces an angle and the scale factor corrected "r" value and also the scale factor corrected rotation module that takes an arbitrary "x" and "y" and the angle from the inverse tangent CORDIC module to apply the rotation. Your CORDIC modules from Project 4 should do scale factor correction with extra shift and add rotations internally. Therefore, there should not be the need for any multipliers in your QRD array. This should be a multiplier free design.

Testing with Vitis: As we plan to use the QRD module in Project 6 and connect this as an IP block to the ARM core via the AXI Lite interface in Vivado, you would export your Model Composer design as an IP block and use the IP inclusion features in Vivado as we did in previous demos. This would allow the use of Vitis to create C++ code to test your QRD block. This will give you a head start on Project 6 instead of using the Model Composer Hardware in the loop Co-Sim approach for verification.

Project Report: Please describe the architecture format that you are using in terms of the systolic array structure. Describe the order of pairings of the matrix elements. Describe your use of CORDIC for inverse tangent and vector rotation. This architecture should use only CORDIC blocks and not require the use of any multipliers. Please include your device utilization report. From the implementation reports (For the HDL Netlist without the ARM core overhead in the ZedBoard target), describe the potential clock period and CLB/slices and IOB utilization. Please include your block diagrams and scope outputs in the report. Describe the hardware co-simulation tests and the accuracy of your results. Also, please prepare the report and model files and upload all of this information to Canvas.

Notes: A sample test matrix will be provided on Canvas in the Project 5 Assignment folder. Also we will revisit the word length and fixed point properties so that this array will work later in Project 6 for the linear system solver.

Extra Credit Architecture: We are considering the QRD of a 4×4 real matrix. In the Paper_Readings for Weed 12, there are two papers on the extension of the QRD to handle a matrix with complex data values as is common in many signal processing applications. There is an additional rotation angle to be handled for complex data elements. For extra credit, extend you QRD array to handle a 4×4 complex matrix.