ELEC 522, Advanced VLSI
Rice University, Fall 2022
15 September 2022

Project 2: Xilinx FPGA Model Composer Systolic for Matrix Multiplication

Due: 29 September 2022

Class Presentation (3 slides each or 5 minutes): 29 September 2022

Please recall the Honor Code policy which is repeated here: Complete this Project 2 assignment **individually**. Solutions should be submitted via Canvas by 11:59pm on the day due to receive potentially full credit. You may freely use the 2022 course notes, the course Canvas resources, or course handouts for the assignments. Students may discuss and compare ideas on the homework (project) assignments, but each student must write up (code and block diagrams, etc.) solutions individually without resort to copying. That is, students must not create submissions together; students may discuss problems but may not actually write up their assignments together. You should also not refer to solutions posted on the web or from previous years' classes. Clearly state any assumptions that you make in order to solve the problems and show all requested work.

Design Problem:

For FPGA, based on the references on VLSI Array Processors, design in Xilinx Model Composer, a hardware accelerator block to perform 4 x 4 matrix multiplication.

FPGA Architecture: This should be a two-dimensional square structure with 16 processor elements that essentially do multiply and accumulate. You should design this as a square array, with some control signals to load, compute, and unload the array.

FPGA Data Format: You can assume simple data types, such as 16-bit two's complement fixed-point numbers. The data generation on Matlab or Simulink can be a simple function to either build the matrix or build individual vectors. You can assume a simple all integer data format, that is no fractional part, to make data generation easier.

FPGA Control: The goals are to handle the array structures and to include the associated delay elements for proper dataflow and timing. This array should interface with Simulink to provide data to the array and then unload it. The array should be able to be triggered multiple times and the array should be able to compute more than one 4 x 4 matrix computation. That is, you should be able to pipeline multiple matrix multiplications with as little delay as possible to load data and unload results. The array should also have a **Matrix-Vector mode** that can be controlled by an external signal. This will allow us to use

the array for both matrix-matrix multiplication and then later in the semester for matrix-vector multiplication in the linear system solver project.

FPGA Data Loading/Unloading: The input data should be loaded in a systolic manner into the rows and columns of the array. The Model Composer Gateways would be at the boundary of the matrix multiplication array. The data generation should not be part of the Xilinx block of the Model Composer model. Instead, the data should be generated in Matlab into the workspace or with basic Simulink blocks running on the host PC. Several unloading schemes were suggested. Choose a method that you believe will be most efficient. Avoid using the "RAM" addressing model that was mentioned in lecture as not easily scalable. Your design should not use large multiplexors for data loading and unloading, since multiplexors can be slow and are not part of the systolic design flow. Please note that the Zynq chip on the ZedBoard only has 200 available I/O pins for gateways and that a few will be used for timing and control by Model Composer. Each I/O bit uses a gateway which is then one physical I/O pin on the Zynq chip. Remember that a 16 bit input data element will use 16 I/O pins. So, plan your I/O with this constraint in mind.

FPGA Testing: Please create the Model Composer block diagram using adder, multiplier, and delay blocks and gateways. You may use other blocks for data generation in the purely Simulink/Matlab non-Model Composer part. Then simulate, implement and describe the resource utilization and the maximum clock rate and slack time for performing this matrix multiplication accelerator kernel. You would need to first use the HDL Netlist generation for our Zynq chip and Vivado to have the place and route implementation reports for only your core array modules excluding the ARM core support logic as in the Project 1 timing and resource utilization tutorial steps. After collecting resource utilization, then target for JTAG HW-Cosim on the ZedBoard for generation, download and run the resulting bit file, to verify hardware in the loop simulation on the ZedBoard using the hardware co-sim feature.

FPGA Files and Documentation: Please upload your design and report to Canvas. This should be a single zip file that contains your design files, the .slx Model Composer and any .m Matlab files along with a report. Include the Powerpoint file that you presented in class. The report should contain a description of your design approach, your tradeoff on input and output architecture, screen shots of your model files, and simulation and hardware co-simulation screen shot results. You should include your resource utilization and timing information from the testing phase. Also, you should describe the scalability of your design, the ability to pipeline a series of matrix multiplications on your array and the latency and delay characteristics. You should also describe your testing methodology so that we can build and run your design in simulation and on hardware.

Presentation: The purpose of the presentation is to describe your architecture decisions and to present resource utilization and maximum clock frequency and pipelining potential. There should be a maximum of 3 Powerpoint slides and 5 minutes with questions. Please email the Powerpoint to cavallar@rice.edu by noon on **29 September** at the latest, so that they can be pre-loaded on the lab computer to save time switching files.