Homework Assignment 04

Matrix Multiplication with Systolic Array

Design a systolic array to realize the matrix multiplication $\boldsymbol{C} = \boldsymbol{A} \times \boldsymbol{B}$. You have the freedom to decide the architecture and design parameters that are not specifically defined. You are also encouraged to raise the discussion when in doubt.

Problem Description and Design Specification

We are going to design the systolic engine to calculate the multiplication of two 4x4 matrices.

1. Assume that the two input matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, are stored in SRAM; the output matrix, $\boldsymbol{C}$, has to be stored to SRAM as well.
2. The testbench of 128 matrix multiplications is given by using 3 separate files:
   a. The **`input_A.txt`** contains the $\boldsymbol{A}$ matrices. Each line represents the 16 elements of one matrix, row by row. E.g.,
   `10 7 5 6 4 7 6 1 7 4 6 3 4 11 9 4`
   means
   ```
   [[10, 7, 5, 6],
    [4, 7, 6, 1],
    [7, 4, 6, 3],
    [4, 11, 9, 4]]
   ```
   b. The **`input_B.txt`** contains the $\boldsymbol{B}$ matrices. Each line represents the 16 elements of one matrix, row by row.
   c. The **`response_C.txt`** contains the golden output $\boldsymbol{C}$ matrices for you to compare with.
3. The input patterns of A and B have to be loaded to SRAM in the linear (given) order; the output responses have to be compared with the golden ones and stored into SRAM in the linear order. You should create the behavior SRAM model of $\mathbf{3} \times \mathbf{16} \times \mathbf{128}$ bytes for $\boldsymbol{A}$ and $\boldsymbol{B}$ inputs, and $\boldsymbol{C}$ outputs.
   a. You should preload the inputs into the SRAM core(s).
   b. Assume each number is represented by one byte (i.e., 8 bits).
   c. Assume it is unsigned-integer multiplication and accumulation. The result is between 0 and 255.
   d. You decide to use one large memory or three smaller memories, or something else.
4. You may use additional buffers (either using SRAM or DFF) to shuffle (or add extra delays to) the inputs before entering the computation engine.

5. The computation engine should be a systolic design. You have the freedom to define the architecture (e.g., 2D or 1D, the projection vector, the scheduling vector, etc.).
6. You may use additional buffer (either using SRAM or DFF) to shuffle the outputs before storing back to the SRAM.
7. Compare the results with the golden responses in your Verilog testbench.
8. Measure the latency between reading the first input from the memory and writing the last result to the memory for the all 128 patterns. You may give the analytical model for the latency. What if we have different number of patterns (e.g., 2048)?
9. Analyze the area based on the synthesis result and estimated SRAM area.
   I. Please leave the SRAM instances un-synthesized.
   II. Instead, estimate the area of SRAM:
      a. The area of a single-port 8192x8 SRAM is 347,200 um^2.
      b. The area of a dual-port 8192x8 SRAM is 766,600 um^2.
      c. As the reference, the smallest NAND2 is 5.09 um^2; a DFF with asynchronous active-low reset is 32.25 um^2, in our cell library.
10. Write the report to summarize all the discussions.

Note

➔ This is a graduate-school-level homework assignment. Make reasonable assumptions or raise a discussion if there is any detail needed to be more specific.

➔ For each assignment, you are requested to write a report with your name and student ID. The topics should include, but are not limited to, the following items:
   a. The design concept with figure and description;
   b. Simulation result with explanation, including the discussion about problems you encounter, and the way you solve them (or not);
   c. A brief summary, including suggestions for us (or this course).

   DO NOT put the entire source code or boring waveform screenshots into the report without proper explanation!

➔ Submit the source files and the electrical report based on TA's instructions.

➔ The source files may include the followings (I assume that you can know what these files are for from their naming) for **HW=hw04**:
   - Testbench: ${HW}_t.v
   - RTL designs: ${HW}.v
   - Non-synthesizable (e.g., SRAM part) designs: ${HW}_nonsyn.v
   - Gate-level implementation: ${HW}_syn.v
   - SDF file from the Design Compiler: ${HW}_syn.sdf
   - FSDB waveforms: ${HW}.fsdb, ${HW}_syn.fsdb
   - Document: README.txt to briefly describe how to perform all the simulations
   - (Optional) Makefile (if you use your own Makefile)

- (Optional) ${HW}.f (if you have your own one)
- (Optional) Other files or documents for your designs and simulations to fulfill the requirement of this assignment. Put the details in your report and README.txt
- The PDF report: ${HW}_YourStudentID.pdf

DO NOT hand in compressed files.

Assignment due on Sunday 5/05/2019, 23:30

1. Submit your source designs and the report.
   - No overdue is allowed.
   - Submission rules will be strictly enforced.