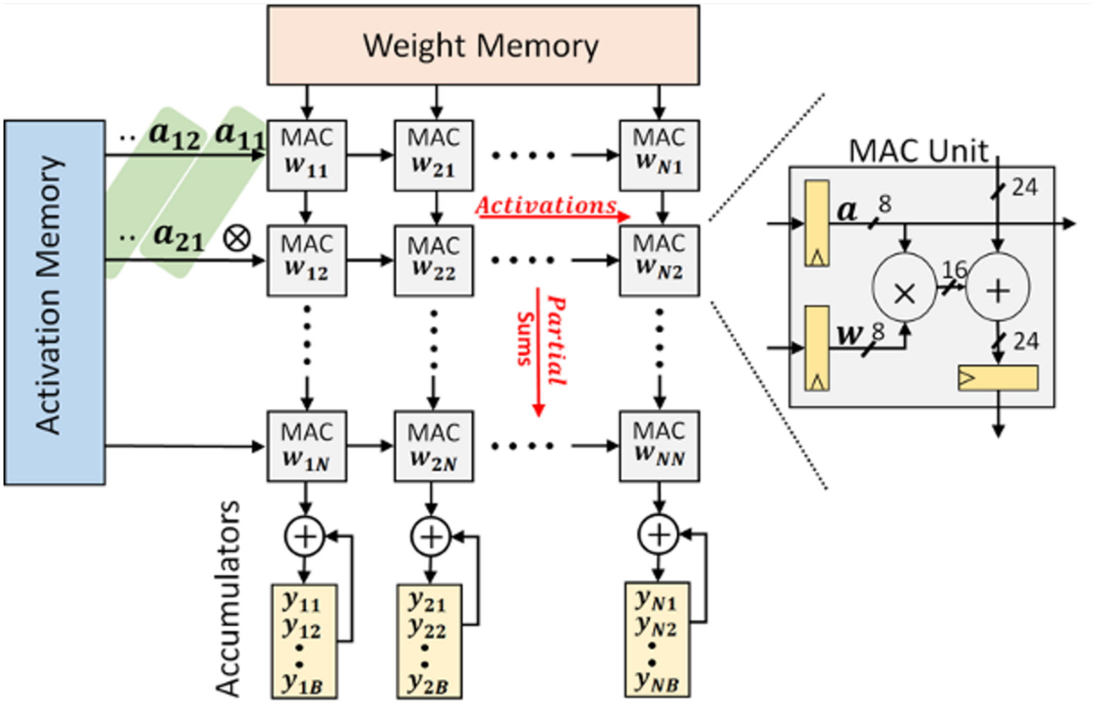
EEE598 Hardware and Systems for Machine Learning - Fall 2023

HW #2

Due Oct/16/23 (Monday) by NOON



Submit a Zip folder named with your ASURite ID, for example 'jeffzhang.zip' which contains:

a. **The Verilog source code, your testbench**

b. **a detailed assignment writeup (PDF report)**
   i. **Describing your design (each module for example)**
   ii. **Screenshots of RTL simulation results**
   iii. **DC synthesis report on timing, power, and area**
   iv. **Post-synthesis gate level simulation results**

In this assignment, we will implement a systolic array from scratch. Please review our lecture notes and understand how a systolic array (SA) works. Specifically, *we will implement a weight-stationary SA, which has a size of 4X4 MAC units.*

Note that we only need to implement the core array (i.e., the 2D MAC array) in this assignment.

1. To help you start, we provide some RTL templates which has the following:
   ○ MAC.v - Please implement your MAC unit inside this file

- MMU.v - This should be your top level module, i.e., the 2-D systolic array. Once you have the MAC unit, you should use your MAC to construct the systolic array. Feel free to create your own modules based on MAC.v but make sure MMU.v is the top module.
- MMU_test.v - This is a sample testbench for you to validate your design. We may create other testbenches to test your submission.

2. Once you finish the RTL, please verify your design in RTL simulation. This can be done on our EECAD servers. Observe the waveform and make sure the SA can produce corrected matrix multiplication results.
   [We can verify this by looking at a single MAC operation, and then a row/column of MAC units in the SA.]
3. If your design passed the RTL simulation. On EECAD servers, please do logic synthesis (DC compiler) on your design with ASAP7 PDK. Please make sure that there are no error messages during the synthesis and report the best timing (fastest clock with no violation) you can achieve, and the power, area of your design.

[Tips: There are documents on ASAP7 setup, DC compiler from EEE525.]

4. Run gate-level (netlist) simulation with the same testbench and verify the results of your design.

5. [Optional Bonus Points] Rethink your RTL implementation. We just implemented a 4X4 SA. What if we wanted to do a 256X256 SA like TPUv1? Can your RTL easily scale to other sizes of SA? [Hints: Verilog generate block can be quite useful!]