EE4321 Digital Systems Design using HDL

Final Project

Mark W. Welker

Project: Simplistic Matrix engine

You will be creating the basic matrix unit that could be utilized in deep learning.

The matrix unit will perform matrix multiplication, scaler multiplication, subtraction, addition, and transposition.

A testbench will place a starting set of matrix in RAM. It will also place the opcodes for the execution unit to perform the matrix functions using the matrix modules.

Project Details

All math functions will have a clear input to set it to 0.

The Register value needs to be able to be written back into a memory location

All Matrices will be 4x4 16 bit deep.

Matrix multiplier will multiple two 4x4 matrix and return a 4x4 matrix

Scalar multiplication is multiplying a matrix by a single number

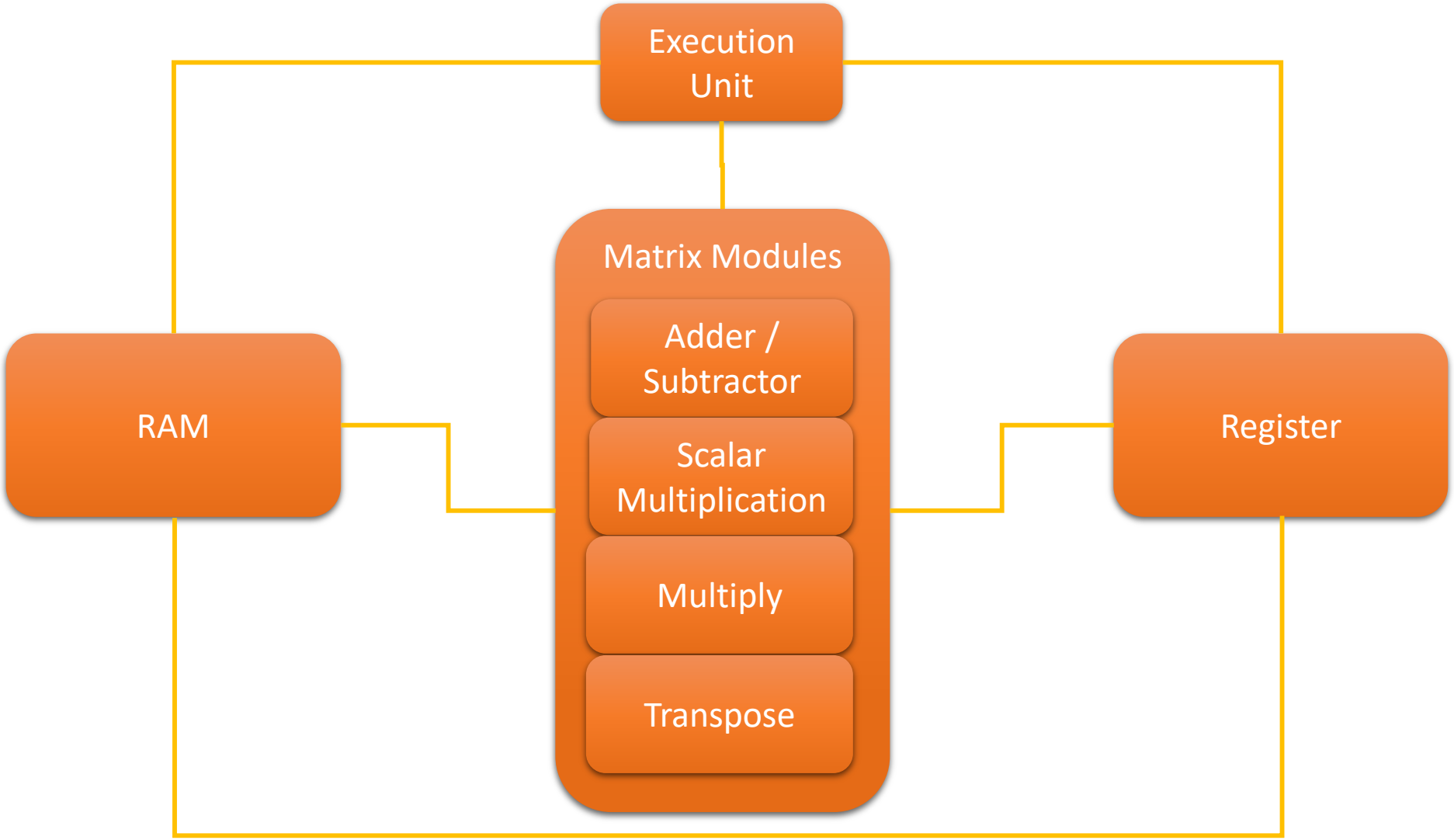
Add and Subtract will add and subtract 2 4x4 matrix

Transpose will flip a matrix along its diagonal.

If you want to brush up an Matrix Math

- https://www.mathsisfun.com/algebra/matrix-introduction.html

Project: Matrix MATH function and Accumulator



Project: Test Bench operation

Load 2 matrix's into RAM
Load opcodes into RAM for the execution unit.

Operations to be performed

1. Add the first matrix to the second matrix and store the result in memory.
2. Subtract the first matrix from the result in step 1 and store the result somewhere else in memory.
3. Transpose the result from step 1 store in memory
4. Scale the result in step 3 store in a register
5. Multiply the result from step 4 by the result in step 3, store in memory.

Project Input Data

The scalar is 7_{10}

Matrix 1

4	12	4	34
7	6	11	9
9	2	8	13
2	15	16	3

Matrix 2

23	45	31	22
7	6	4	1
18	12	13	12
13	5	7	19

Project Details

The test bench will Start the clock and toggle reset

Execution engine will fetch the first opcode from instruction memory and begin execution.

The execution engine will direct the transfer of data between the memory, the appropriate matrix modules and memory.

The execution engine will continue executing programs until is finds a STOP opcode.

The test bench will display an output waveform to determine correct operation.

Project Recommendations

Execution engine

- You will need to generate your own opcode set it could include opcodes for
  - STOP
  - Matrix Addition to/from memory, to/from register
  - Matrix Subtraction to/from memory, to/from register
  - Matrix Multiply to/from memory, to/from register
  - Matrix Transpose to/from memory, to/from register
  - Matrix scale to/from memory, to/from register
- You are free to designate the opcodes any way you wish

Registers can be implemented as a register file or as a registers in the modules.

You will need to determine how you wish to transfer the matrix between modules.

Extra credit

- During the semester I will be adding items into the extra credit category. (THIS PAGE and pages like it)
- Extra credit will be applied to your project if and ONLY if your project provides the proper matrix outputs.
- Possible extra credit each one could add up to 10 points to your project. Which will help your overall grade.
  - Create a more CPU like architecture, everything is memory mapped, simpler instructions
  - Handle overflow of matrix multiplications – new scalar to be used (42 instead of 7)
  - Handle standard integer math (+,-,*,/)