Project Plan report format – Your first page MUST match this format

/10

Name: Sean Daniels

Unityid: sdaniel5

StudentID:. 200175286

Summary Risk Plan:

Interfacing with the SRAM

Connecting all modules

Time

Schedule:

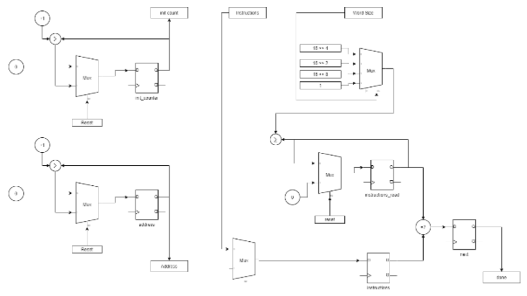
Week 1: Write individual modules, link entire design, get system working

Week 2: Test and optimize

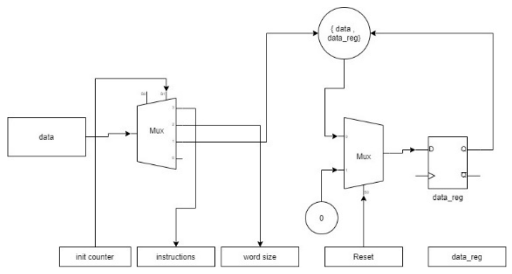
Brief Description of Mode of operation, including selected algorithms

The system follows the class outline of separate data path and control algorithms. Both weights and inputs will have there own instantiations of data path and control modules. When both control modules register the completion of their respective tasks, the system will execute the matrix multiplication. When the matrix multiplication is complete, an independent control module will begin the process of writing the data to the appropriate address in the output SRAM

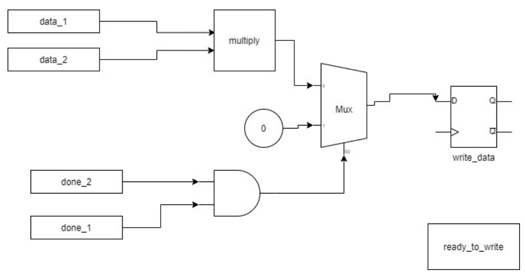
Control:



Data Path:



Executor:



Verification Plan:

In order to verify the correct functionality of the entire project, I plan to employ modular verification checks. The checks will be as follows:

Check 1: Verify synchronized input of SRAM data, i.e. can I pull data in from SRAM

Check 2: Verify synchronized control scheme for SRAM data, i.e can I pull data end and send it to the appropriate registers at the appropriate time

Check 3: Verify correct data output from data path 1, i.e is the data being passed to execution function correct

Check 4: Verify correct data output from data path 2, i.e is the data being passed to the execution function correct

Check 6: Verify synchronization between control instance 1 and control instance 2, i.e do control 1 and control 2 arrive at the execution function at the right time

Check 7: Verify correct output from the execution function, i.e is the matrix multiplication function working correctly

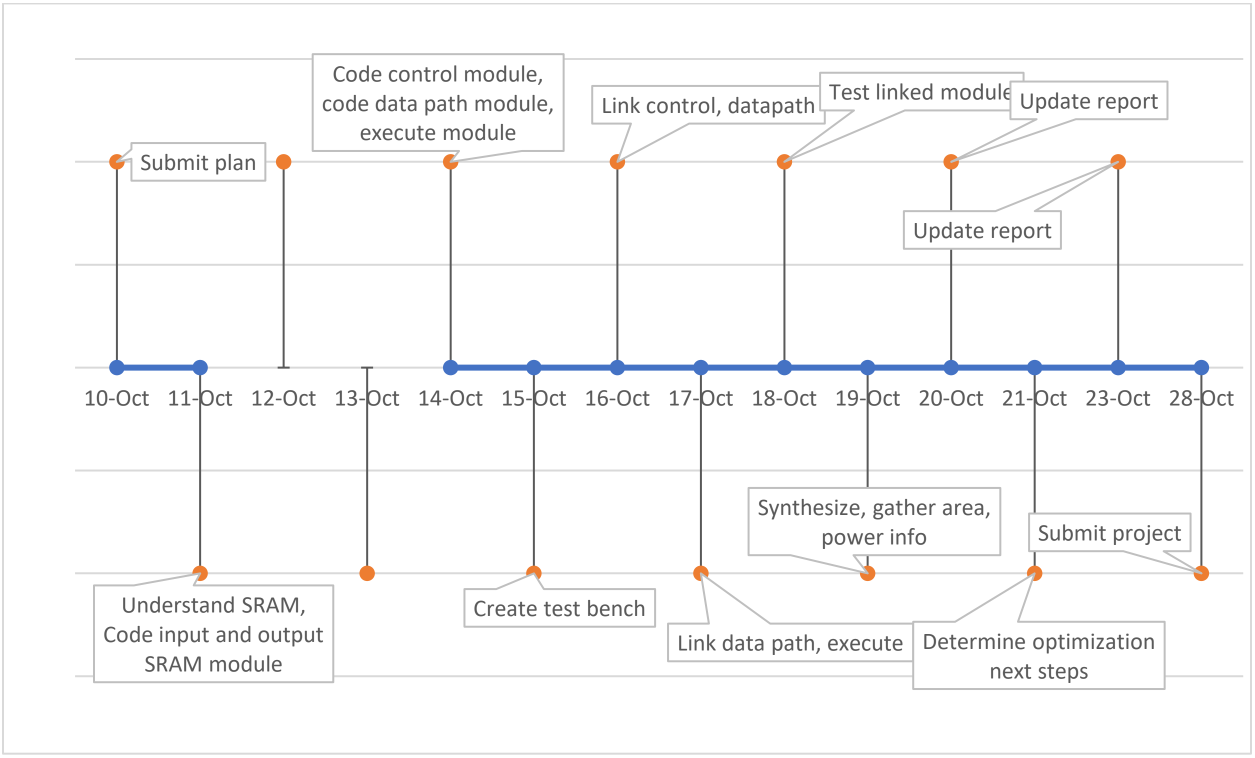
Check 8: Verify correct output to the SRAM, ie is the correct data being placed at the correct address

Risk Assessment:

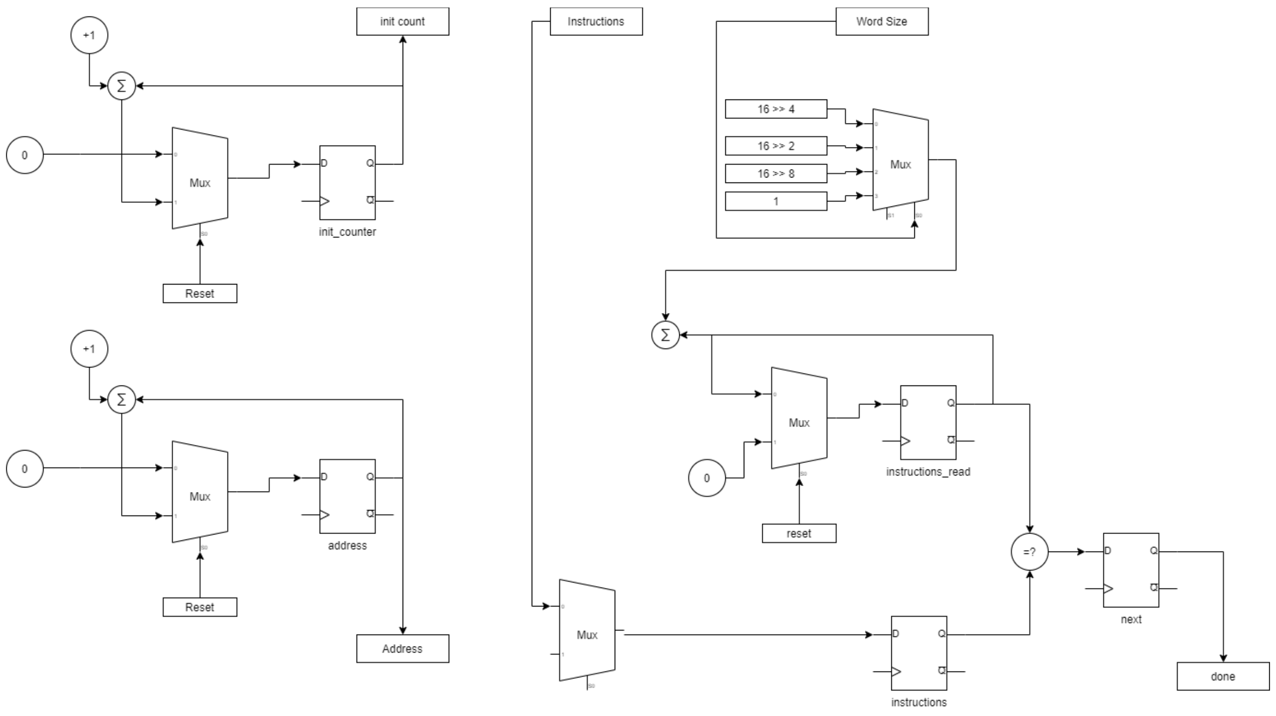
At this point, I am primarily concerned with interfacing with the SRAM. I'm not totally sure the timing scheme for them, and I am also confused by the 12 bit addresses being passed to them (why not 16-bit addresses?). I am also confused about 'read_write_select' toggles for the SRAM. I plan on mitigating this in concert with my other benchmark goals.

A secondary concern is organization relative to instantiating a multi-level design. I really want to avoid getting to the end of this project, the output being incorrect, and having a mess of nets and registers which have connections that I don't understand. I plan on mitigating this with the modular synchronization plan.

Milestone Plan:

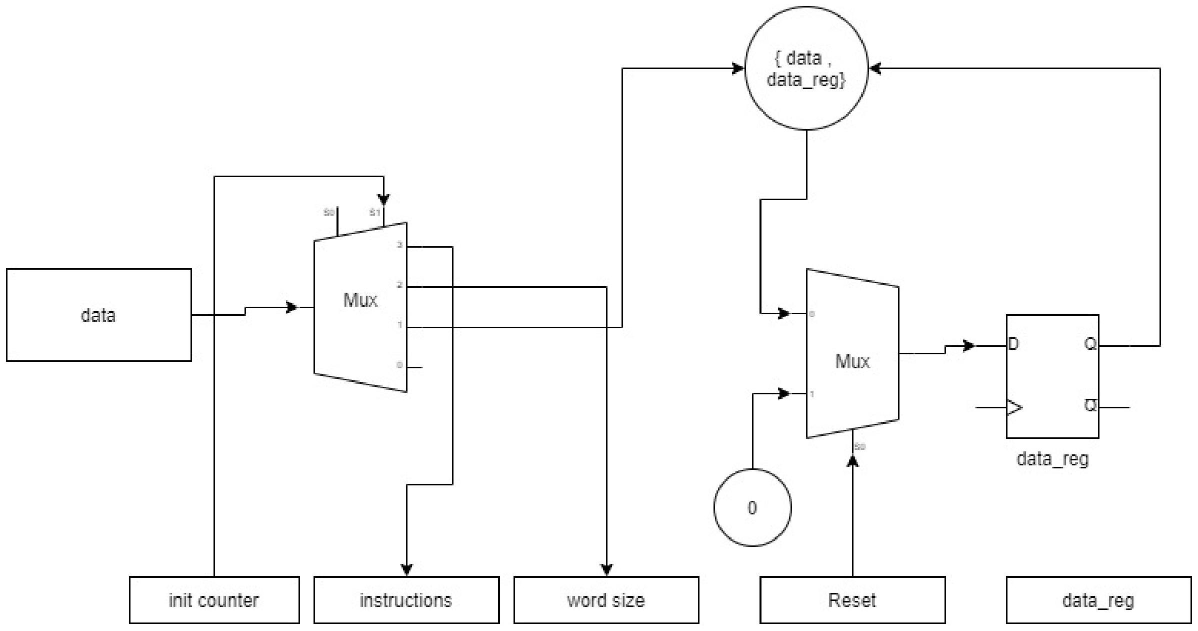


Control Module (x2):



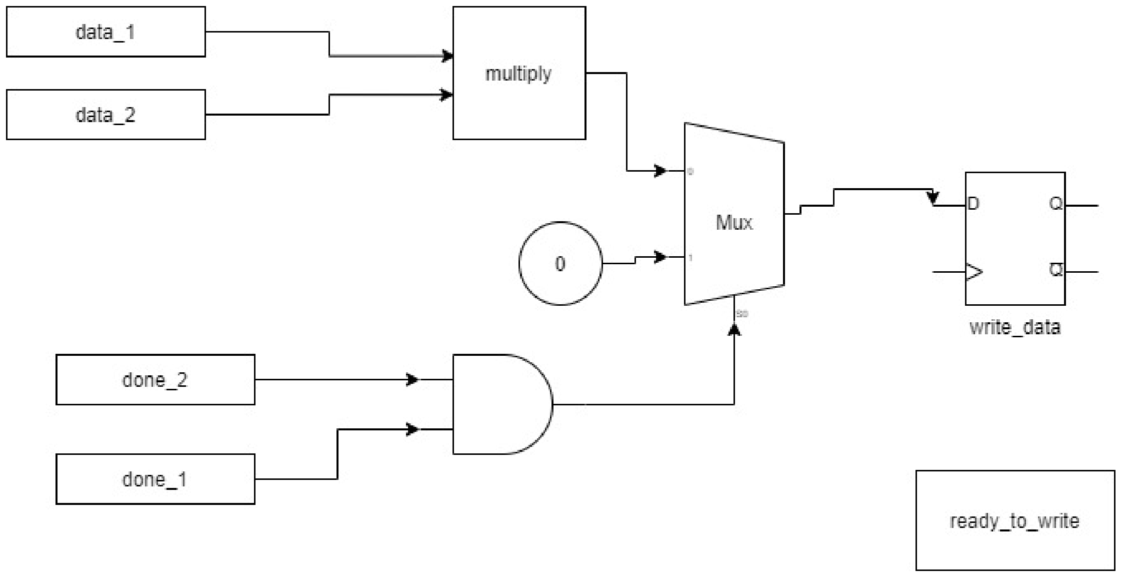
The control module handles iterating through the addresses in the SRAM. The SRAM data is passed to the data path, and some of those values are passed back to control path, such as number of instructions and word size. The control module needs these values to know when to stop iterating relative to a particular input. After all the values for the input are received, a signal is sent to the execution module, indicating that this module is ready to be used as an argument in the execution module.

Data Path (x2):



The data path reads values from the SRAM and either sends them back to the control unit, to be used as values for the control of the instruction read, or concatenated and used as an output for the execution module

Execute:



The execution module receives the outputs from the two data path modules, and multiplies them together. After multiplying the arguments, the executor module sends a ready to write signal, along with the result of the multiplication to the SRAM output module.