Team Members: Saroj Bardewa and Conor O'Connell

Week #6

THIS WEEK:

1. Designed Instruction Memory Unit
2. Designed Data Memory Unit

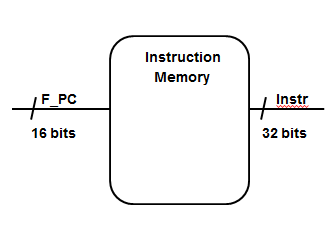
**NEXT WEEK:**

1. Work further on designing a basic pipeline NN processor.
2. Extend capability of our simulation to load in a hex textfile and save the results to a hex file
3. Complete the assembler and produce the instructions in hex format

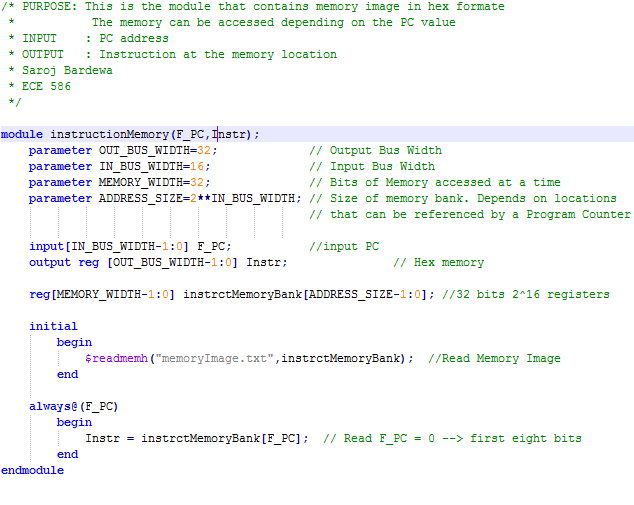
**# Detail of this week:**

1. Designed Instruction Memory Unit

The first step in pipeline is to read in instruction from the memory\_image. We designed an instruction memory unit that reads in and saves 32-bit of data in a memory bank. And the program can access the instruction in any order –it can process out of order execution.

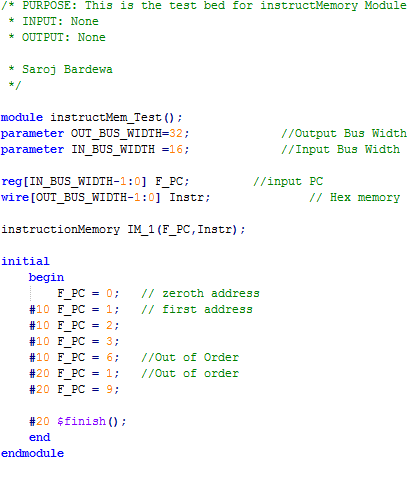


***Instruction Memory Module Program***



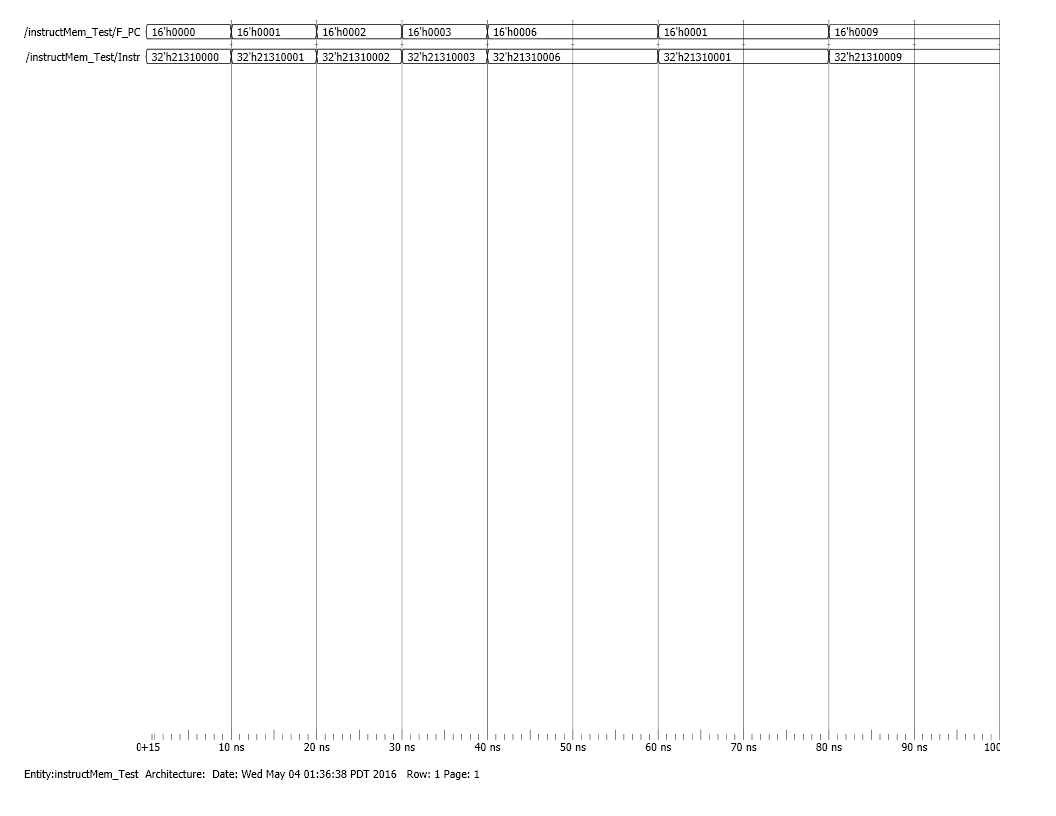
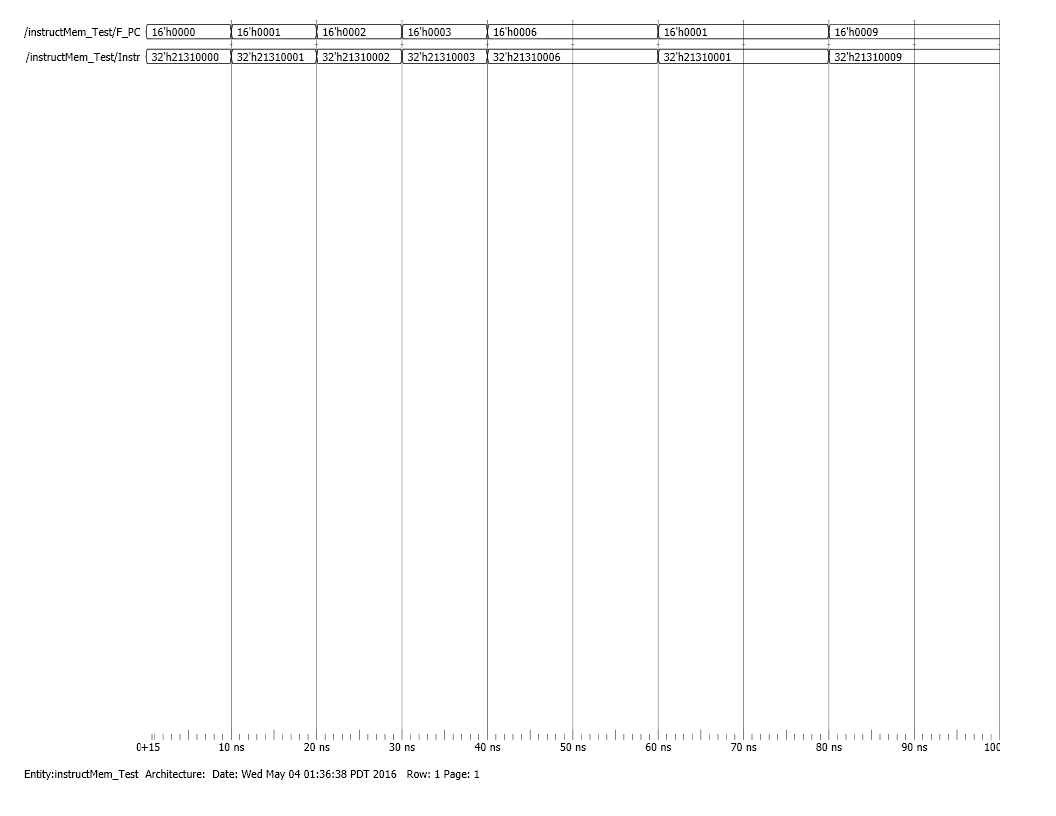
Test bench for the instructionMemory Module

This program tests the InstructionMemory Module to verify the output of the module for given input combination. It is shown in figure 2.



**Result of the simulation:**

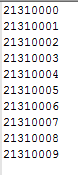
The result of the simulation gives the memory instruction accessed for a given PC value. The out of order execution makes the program very efficient. It is shown in figure 2.



1. **Conor??**

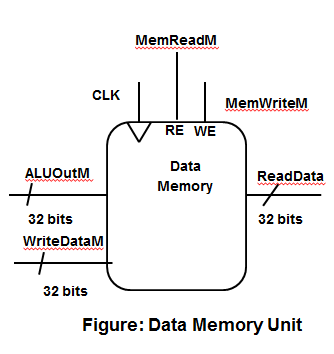
**Test File:**

This test file contains the memory image of 32-bit address in hex format.

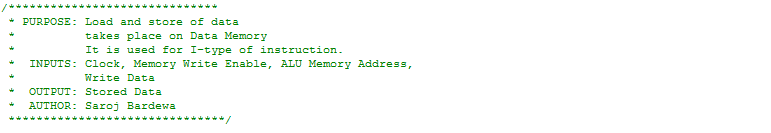
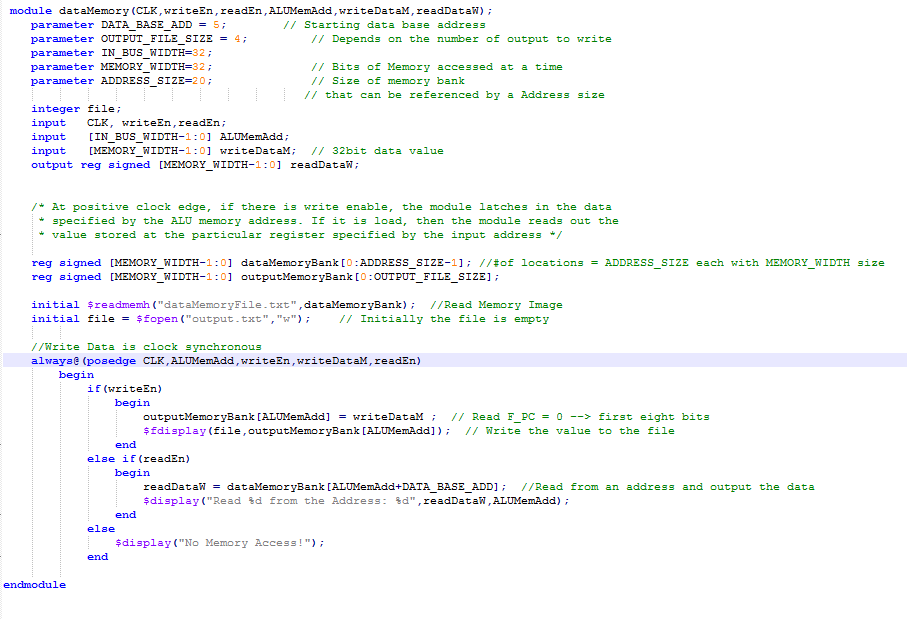
****

1. **Designed Data memory Unit:**

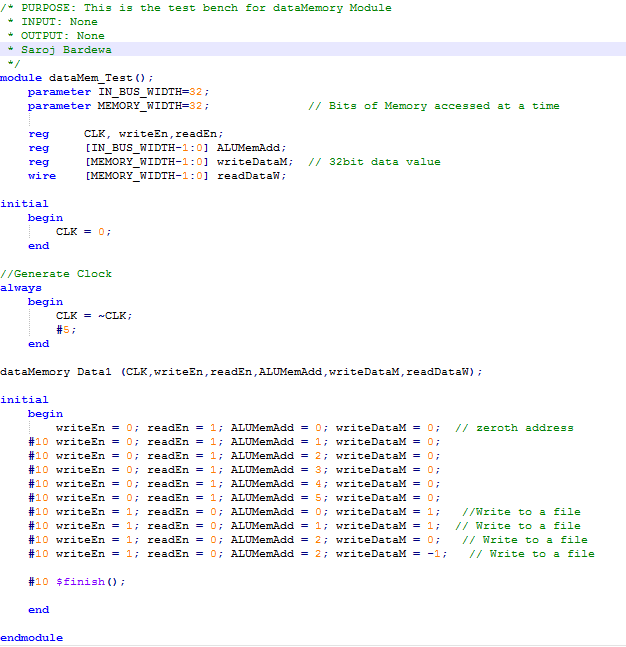
Data memory unit is used to load and store data. The input values and internal weights are loaded into this module which is accessed for calculation. Also, the output of the neural network is stored in this unit, and which is further written into a file.



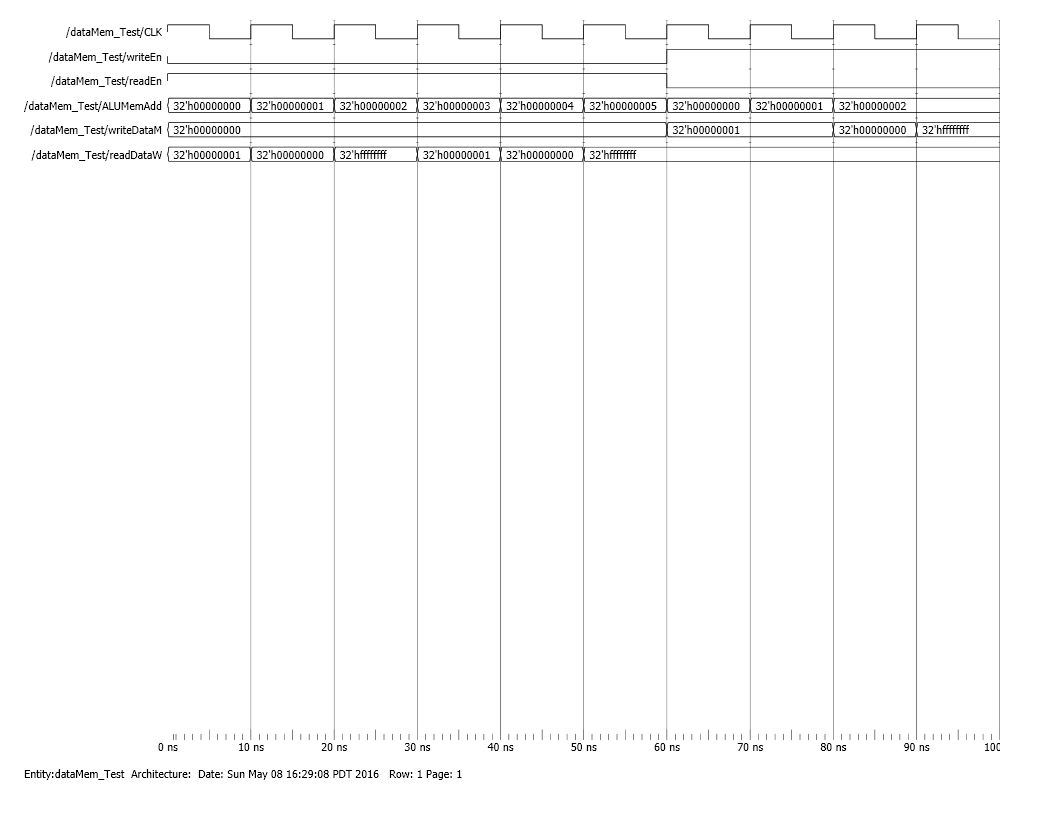
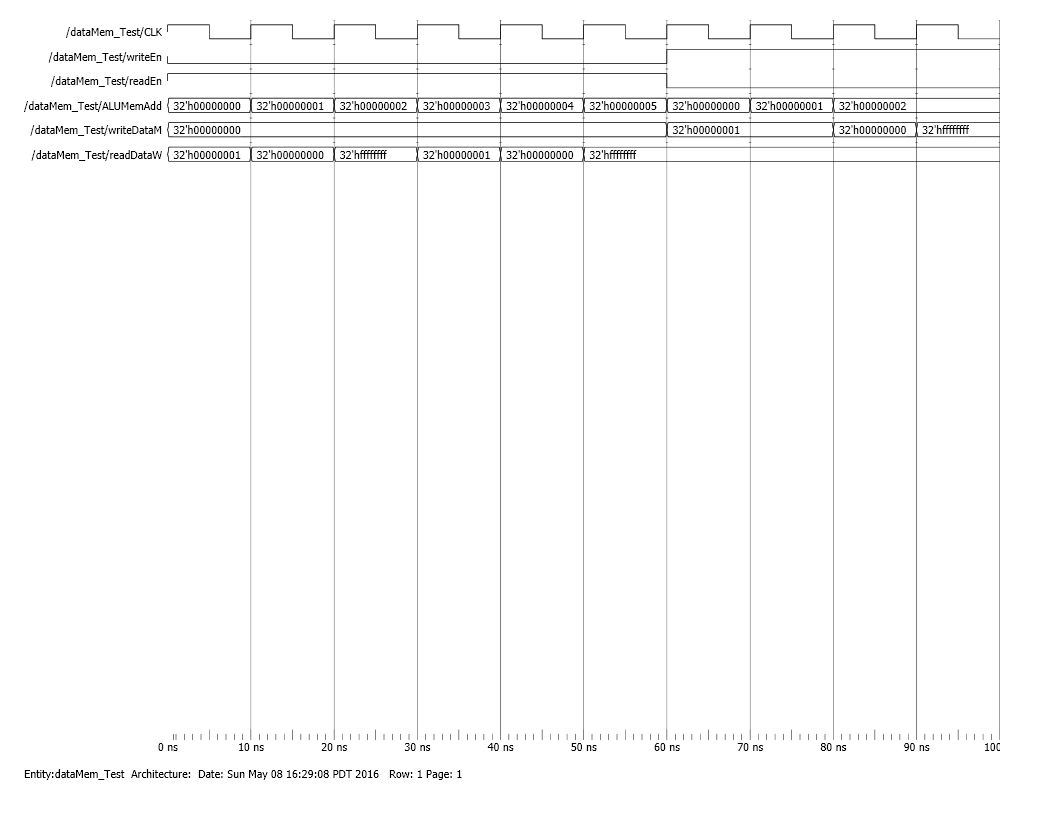
*Data Memory Module:*



*TestBench of Data Memory module:*

**

*Output of the module:*



*Test Input File:*

**

*Test Output File Generated from the module:*

**