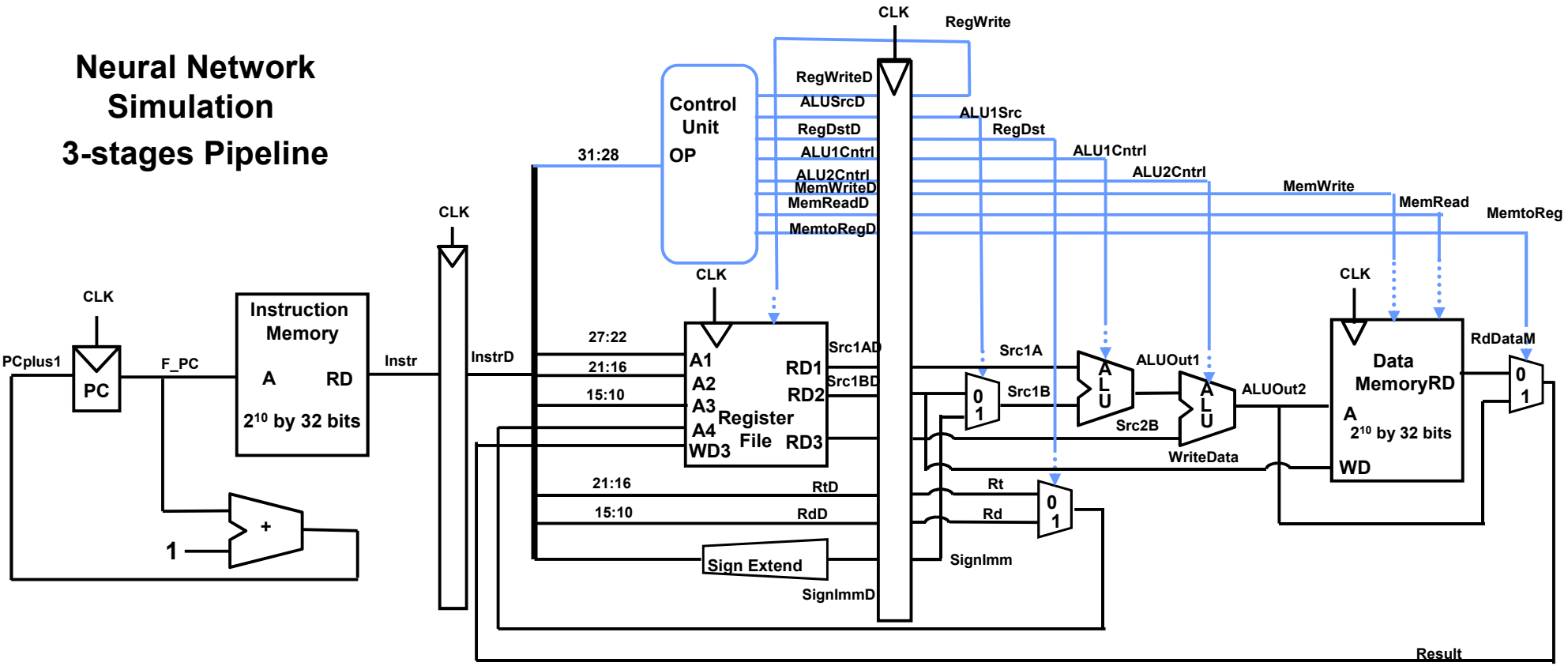
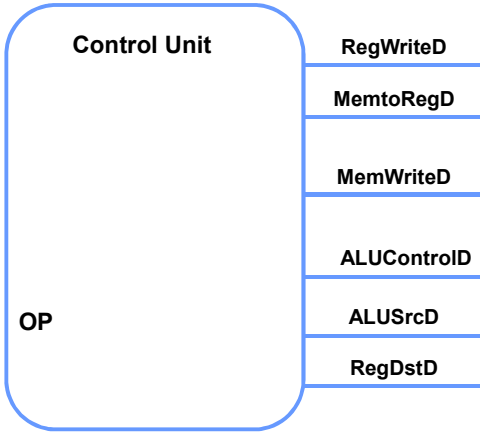
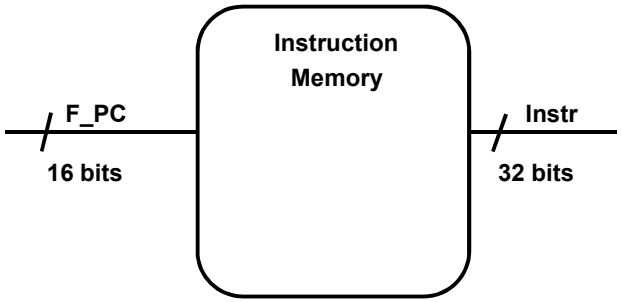
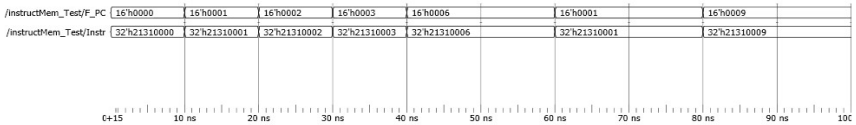
Neural Network Simulation
3-stages Pipeline



Saroj Bardewa & Conor O'Connell


Control Unit
OP
RegWriteD
MemtoRegD
MemWriteD
ALUControlD
ALUSrcD
RegDstD


Instruction
Memory
F_PC
16 bits
Instr
32 bits



Entity:instructMem_Test Architecture: Date: Wed May 04 01:36:38 PDT 2016 Row: 1 Page: 1

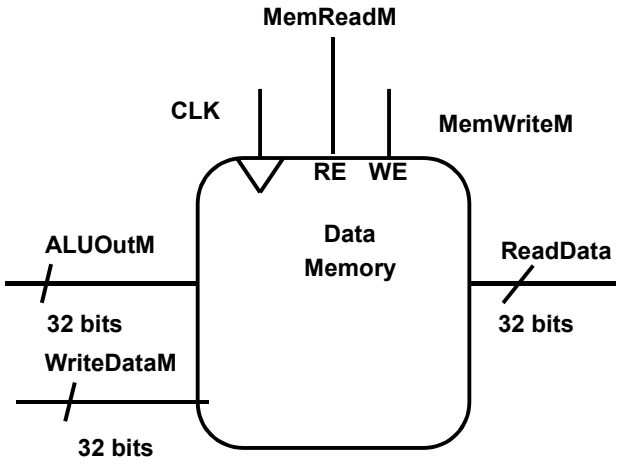
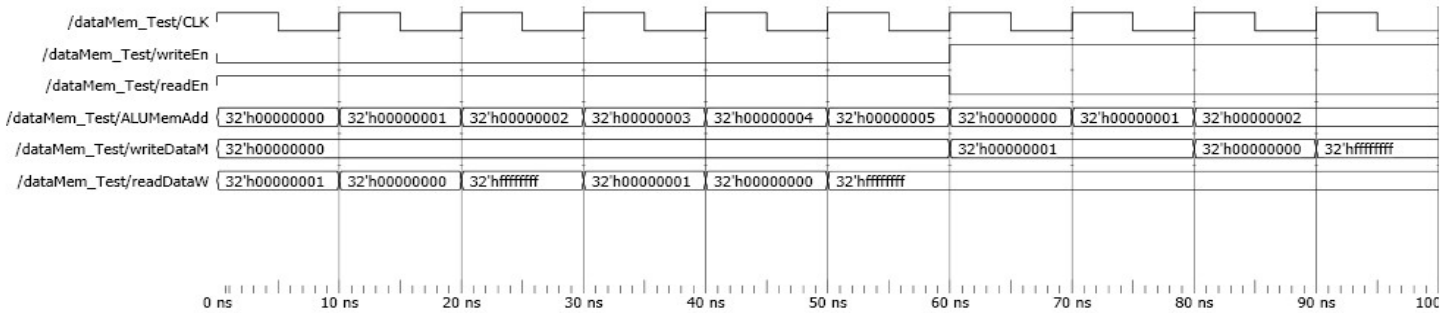


Figure: Data Memory Unit

```
/*******************************
 * PURPOSE: Load and store of data
 *          takes place on Data Memory
 *          It is used for I-type of instruction.
 *  INPUTS: Clock, Memory Write Enable, ALU Memory Address,
 *          Write Data
 *  OUTPUT: Stored Data
 *  AUTHOR: Saroj Bardewa
 *******************************/

module dataMemory(CLK,writeEn,readEn,ALUMemAdd,writeDataM,readDataW);
   parameter DATA_BASE_ADD = 5;        // Starting data base address
   parameter OUTPUT_FILE_SIZE = 4;         // Depends on the number of output to write
   parameter IN_BUS_WIDTH=32;
   parameter MEMORY_WIDTH=32;              // Bits of Memory accessed at a time
   parameter ADDRESS_SIZE=20;              // Size of memory bank
                                          // that can be referenced by a Address size
   integer file;
   input   CLK, writeEn,readEn;
   input   [IN_BUS_WIDTH-1:0] ALUMemAdd;
   input   [MEMORY_WIDTH-1:0] writeDataM;  // 32bit data value
   output reg signed [MEMORY_WIDTH-1:0] readDataW;


   /* At positive clock edge, if there is write enable, the module latches in the data
    * specified by the ALU memory address. If it is load, then the module reads out the
    * value stored at the particular register specified by the input address */

   reg signed [MEMORY_WIDTH-1:0] dataMemoryBank[0:ADDRESS_SIZE-1]; //#of locations = ADDRESS_SIZE each with MEMORY_WIDTH size
   reg signed [MEMORY_WIDTH-1:0] outputMemoryBank[0:OUTPUT_FILE_SIZE];

   initial $readmemh("dataMemoryFile.txt",dataMemoryBank);  //Read Memory Image
   initial file = $fopen("output.txt","w");    // Initially the file is empty

   //Write Data is clock synchronous
   always@(posedge CLK,ALUMemAdd,writeEn,writeDataM,readEn)
       begin
           if(writeEn)
               begin
                   outputMemoryBank[ALUMemAdd] = writeDataM ;  // Read F_PC = 0 --> first eight bits
                   $fdisplay(file,outputMemoryBank[ALUMemAdd]);  // Write the value to the file
               end
           else if(readEn)
               begin
                   readDataW = dataMemoryBank[ALUMemAdd+DATA_BASE_ADD];  //Read from an address and output the data
                   $display("Read %d from the Address: %d",readDataW,ALUMemAdd);
               end
           else
               $display("No Memory Access!");
           end

endmodule
```



Entity:dataMem_Test Architecture: Date: Sun May 08 16:29:08 PDT 2016 Row: 1 Page: 1