Instruction Memory, Register File, Data Memory

COEN 122 Lab 3

Grading

- Lab 1-4 (10% each) 40%
- Final Project 60%

Late Policy

- Late submissions submitted within 24 hours after the deadline receive 50% credit. After 24 hours, no credit is given.

Demo

- Labs submitted with no demo will receive no credit. Please demo to me before the lab is due.
- You can demo in either TAs Office Hours

Lab 3 Overview

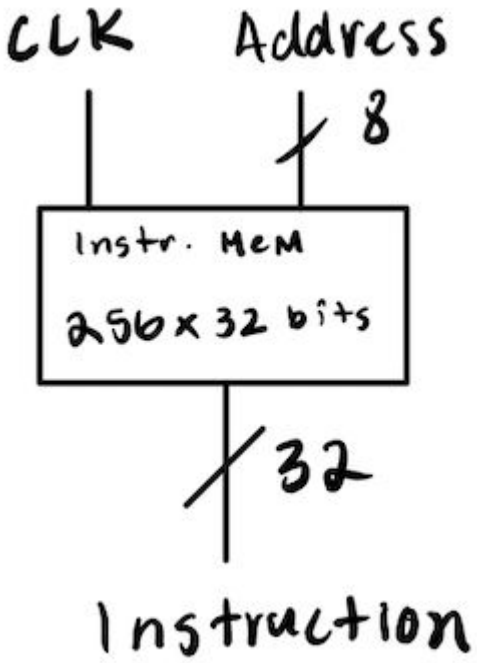
- Objective: Implement a Instruction Memory, Register File, and Data Memory in Verilog using Vivado
- For the first 4 labs, you will be working individually, but will be working in teams for the project

Helpful website: http://www.asic-world.com/verilog/veritut.html

Design Flow

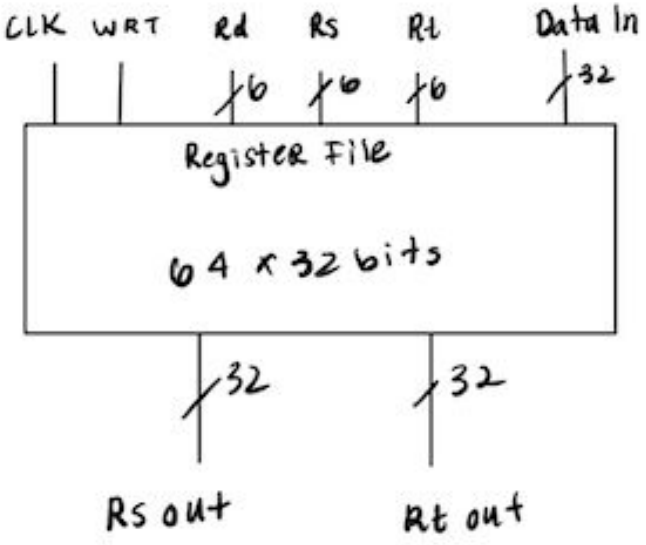
- Write code for logic block (e.g. Instruction Memory, Register File, Data Memory)
- Write test bench to verify proper function of logic block
- Run Synthesis
- Run Simulation
- View waveform, verify results

Instruction Memory



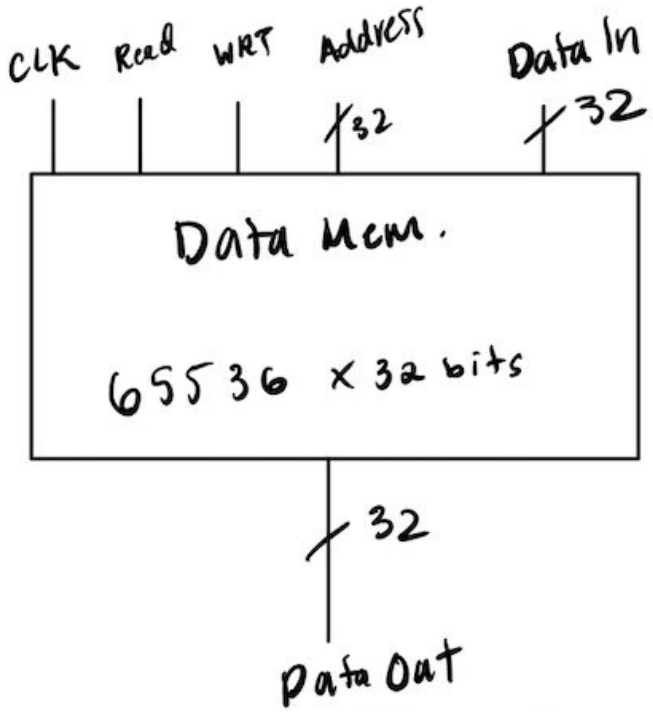
- Holds 32-bit values for pipeline instructions
- In design file, hardcode values and in testbench file, read those values

Register File



- If write signal is high, datain should be written to the address input, rd
- In testbench file, write values into memory in testbench, then read them back out

Data Memory



- 2^16 = 6536
  - Only use the lower 16 by writing address[15:0]

- Similar to the register file, if the write is high, data_in will be written to the data at the specific address input
- If read is high, then data_out will contain the data at the address

Important Constructs

- Test Bench:

```
reg clock;
initial
begin
        clk = 0;
        forever $5 = ~clk;
end
initial
begin
        //…
$finish;
end
```

- Design Source:

```
input clock;
always@(posedge clock)
begin
        //…
end
```

- 2d array:
  - reg [31:0] data [15:0]