

Lab 2 preview

Hung-Wei Tseng

Announcement

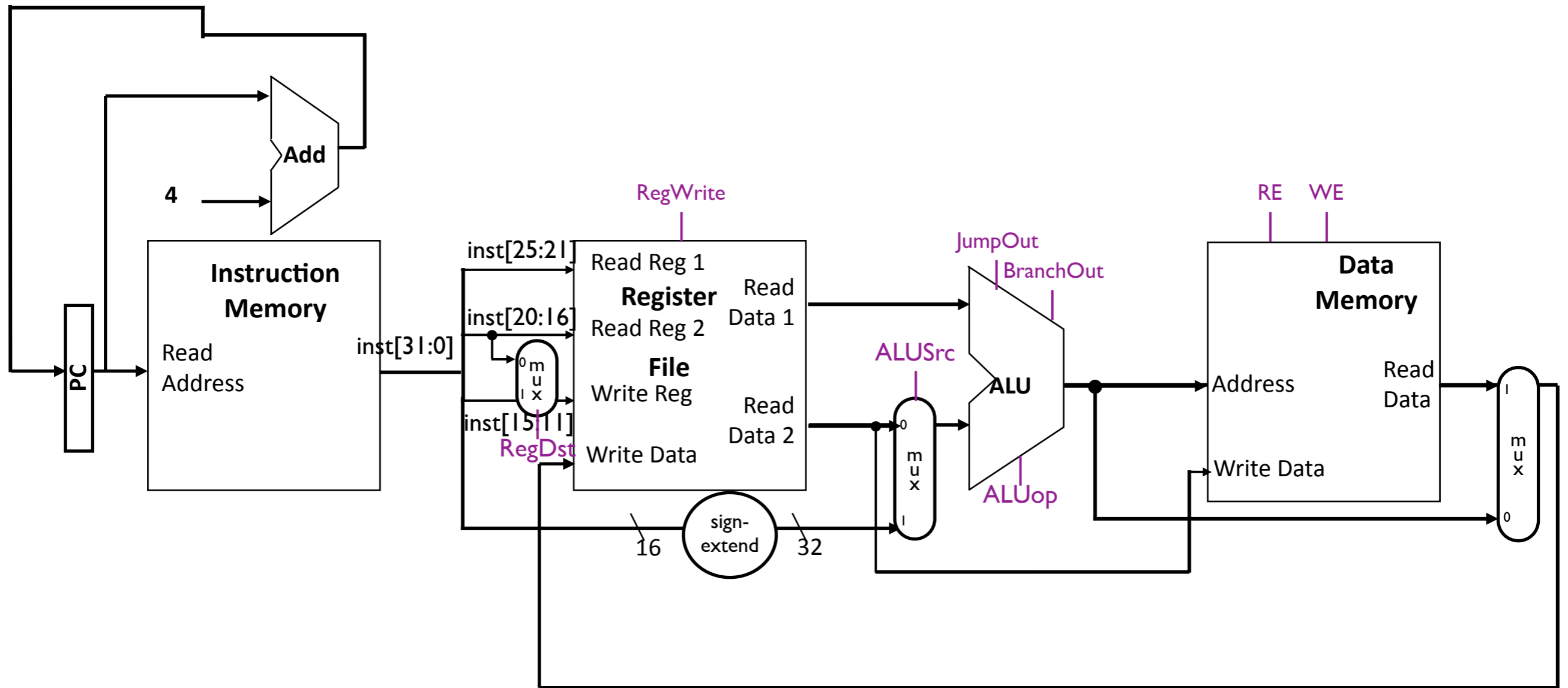
- Lab 1 due tomorrow
 - Mail it to h1tseng @ cs.ucsd.edu, not to the TAs
 - name your pdf as cse141L-lab1-LastName-FirstName.pdf
 - name your e-mail title as [CSE141LSU14] cse141L-lab1-LastName-FirstName
 - Your report will not be graded if any of the above goes wrong
 - No late reports, no group work
- Form your group from now!
 - We need 2 or 3 persons to make a processor
 - 4 or 5 are too more, we are not making burgers



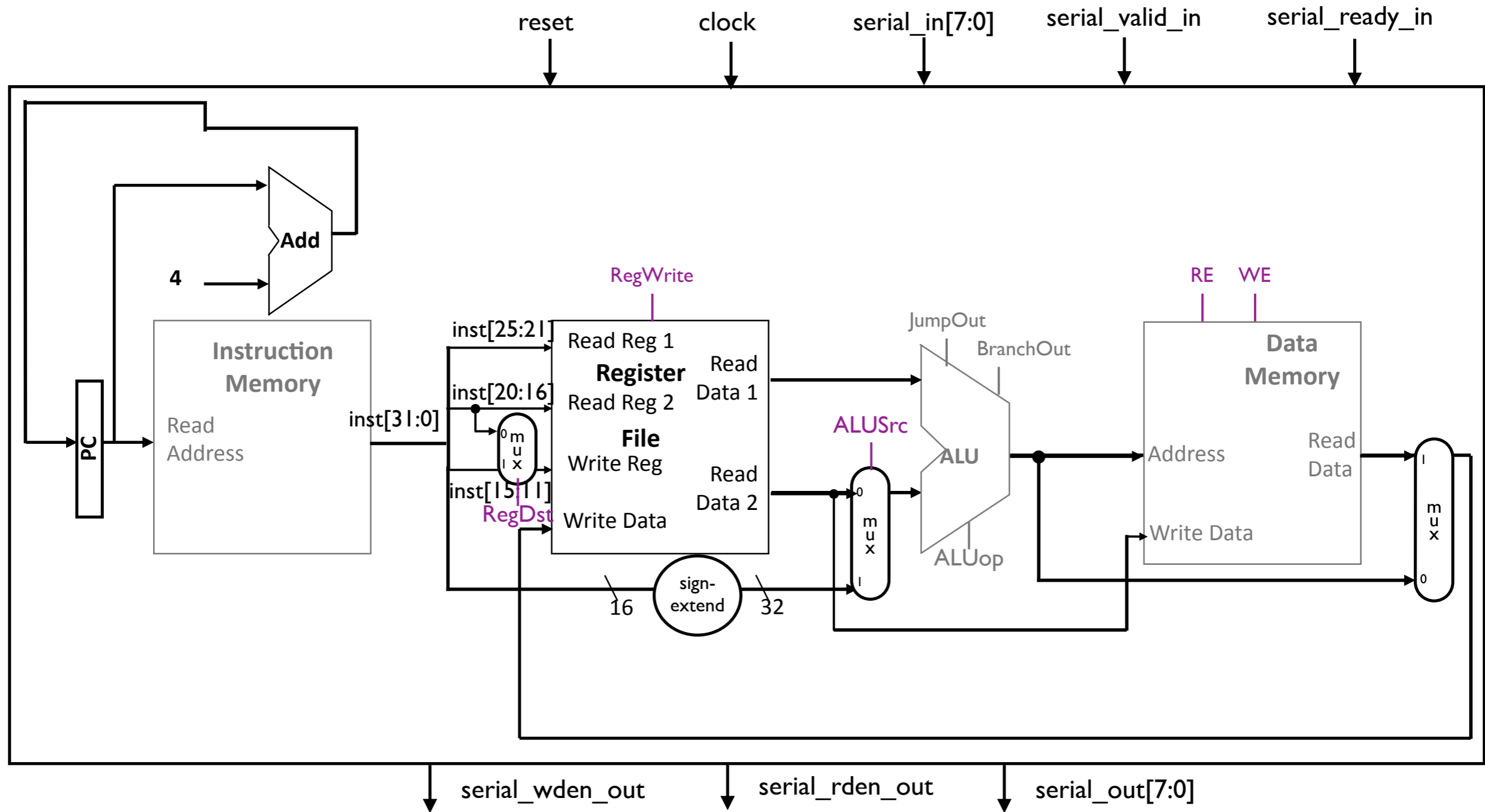
In Lab 2...

- You will be implementing the datapath elements for a single cycle processor
 - We provide ALU, data memory, instruction
 - You need to implement the rest
- Design the schematic of your processor first
 - What are the elements required for a single cycle processor
 - How to test each component
 - How to connect them together

Lab 2: single-cycle datapath

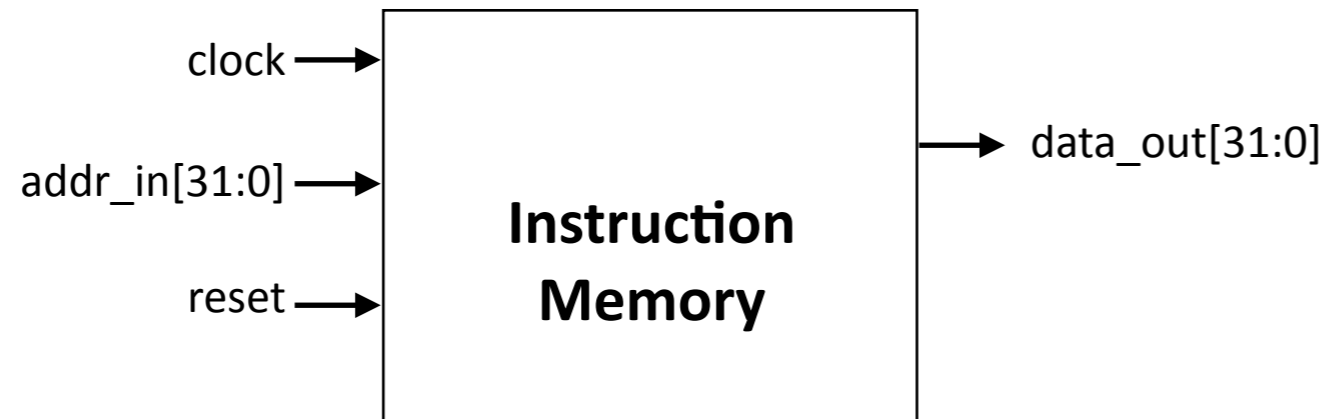


You have to implement the processor



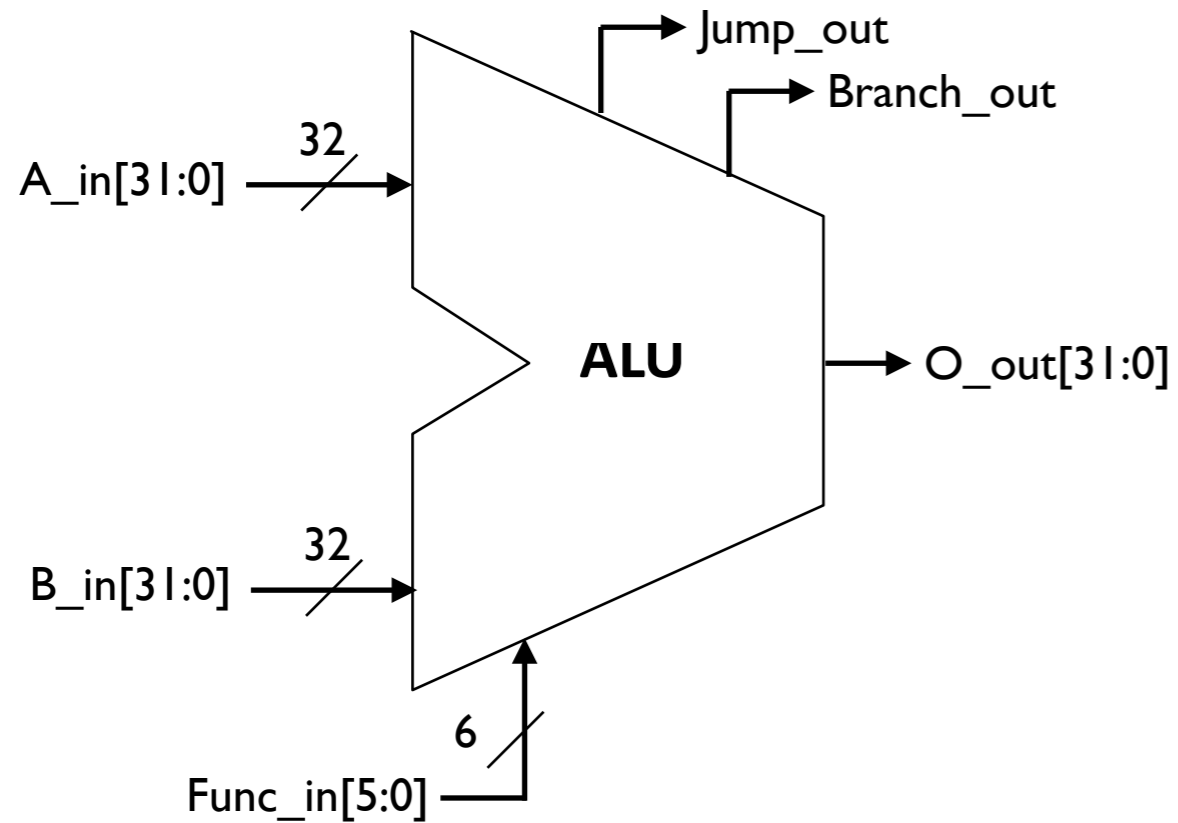
You have to implement the black parts!

Interface of instruction memory



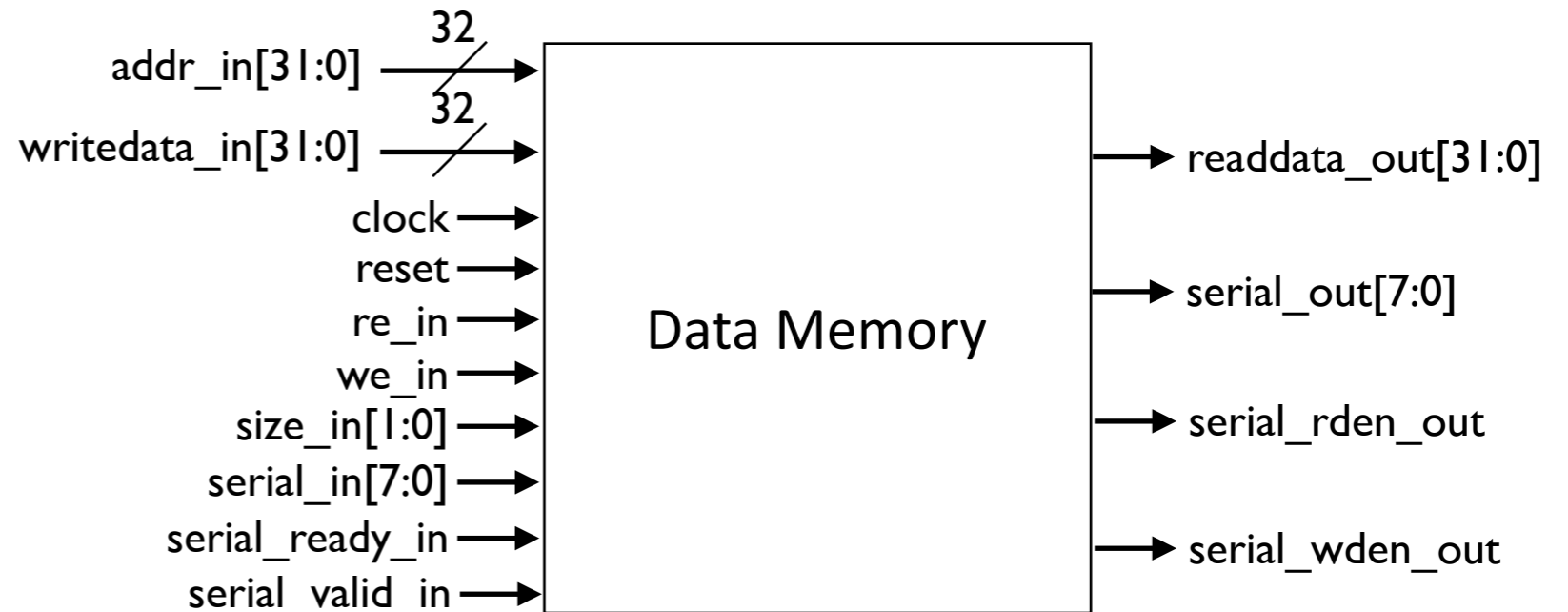
```
module inst_rom(  
    input clock,  
    input reset,  
  
    input [31:0] addr_in, //Connect to PC_next  
    output [31:0] data_out //Fetched instruction  
);  
parameter INIT_PROGRAM="c:/myfiles/blank.memh";  
endmodule
```

Interface of ALU



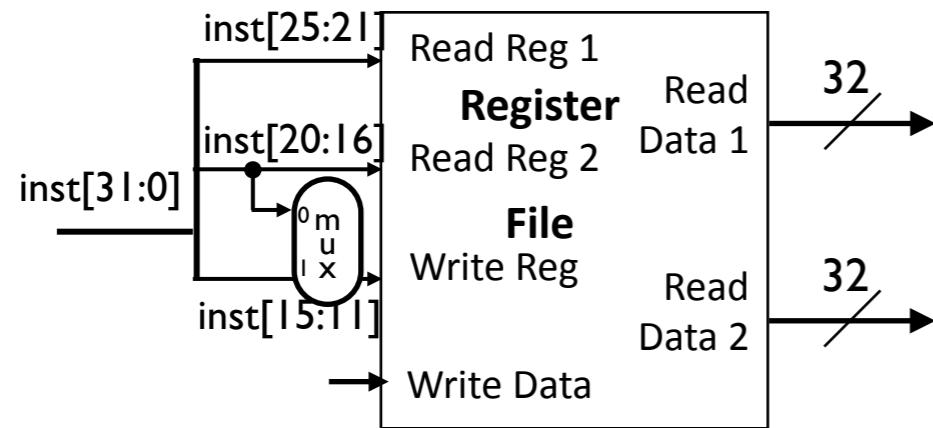
```
module alu(  
    input [5:0] Func_in,  
    input [31:0] A_in,  
    input [31:0] B_in,  
    output [31:0] O_out,  
    output Branch_out,  
    output Jump_out  
);
```

Interface of data memory



```
module data_memory(  
    input clock,  
    input reset,  
    input [31:0] addr_in, //Read/Write address  
    input [31:0] writedata_in, //Data to write to memory  
    input re_in, //Read Enable - set high when reading from memory  
    input we_in, //Write Enable - set high when writing to memory  
    output [31:0] readdata_out, //Data output for reads from memory  
    input [1:0] size_in, //Not used yet - hardwire to 2'b11  
    input [7:0] serial_in,  
    input serial_ready_in,  
    input serial_valid_in,  
    output [7:0] serial_out,  
    output serial_rden_out,  
    output serial_wren_out  
);  
parameter INIT_PROGRAM0="c:/myfiles/blank.memh";  
parameter INIT_PROGRAM1="c:/myfiles/blank.memh";  
parameter INIT_PROGRAM2="c:/myfiles/blank.memh";  
parameter INIT_PROGRAM3="c:/myfiles/blank.memh";
```

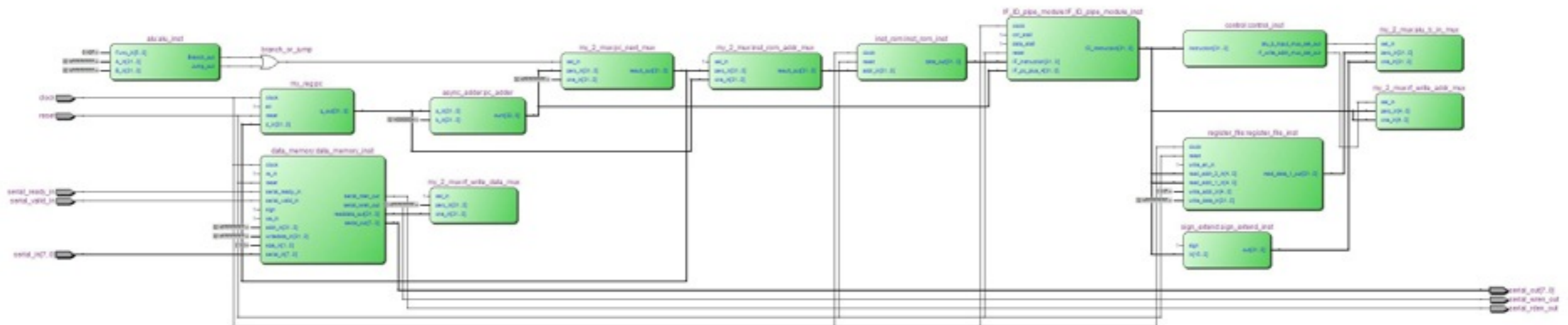

Your register file



- Should contain at least the above inputs/outputs
 - Does every instruction write back data?
- Is it clocked?
- Support reset
- Deal with \$zero

When you interview with us

- Show us your processor.v
- Show us your register_file.v
 - Demonstrate it works: you need to design a testbench for it.
- Show us your schematic
 - You may output it through the RTL viewer under analysis tool in Quartus II



Q & A