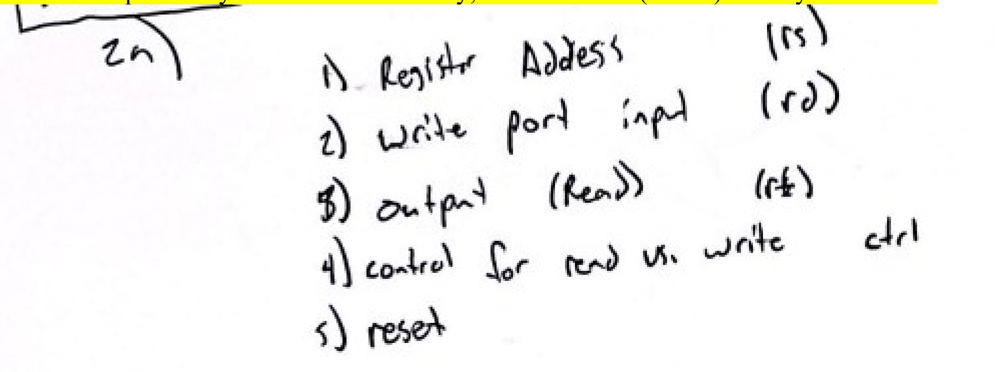
CprE 381, Computer Organization and Assembly-Level Programming

Lab 2 Report

Student Name Austin Beinder____

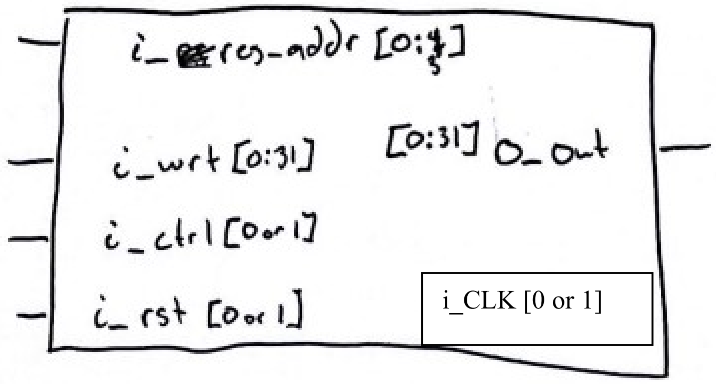
Submit a typeset pdf version of this on Canvas by the due date. Refer to the highlighted language in the lab document for the context of the following questions.

[Part 2 (a)] Draw the interface description (i.e., the "symbol" or high-level blackbox) for the MIPS register file. Which ports do you think are necessary, and how wide (in bits) do they need to be?



2n)

1) Registr Addess (rs)
2) write port input (rd)
3) output (Read) (rt)
4) control for read vs. write ctrl
5) reset

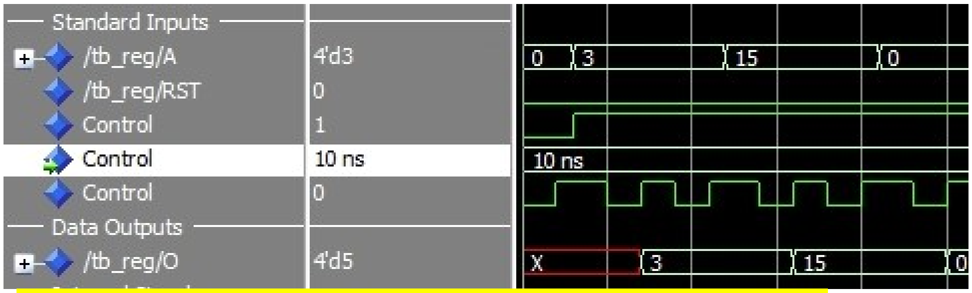


I think the register address needs to be 5 bits wide, that the data input should be 32 bits, the output should be 32 bits, the read vs write and rst should both be 1 bit. The clock is also neccesary and should be 1 bit. I think the design is asynchronous because it happens sequentially.

[Part 2 (b)] Create an N-bit register using this flip-flop as your basis.

See RegFil/reg.vhd

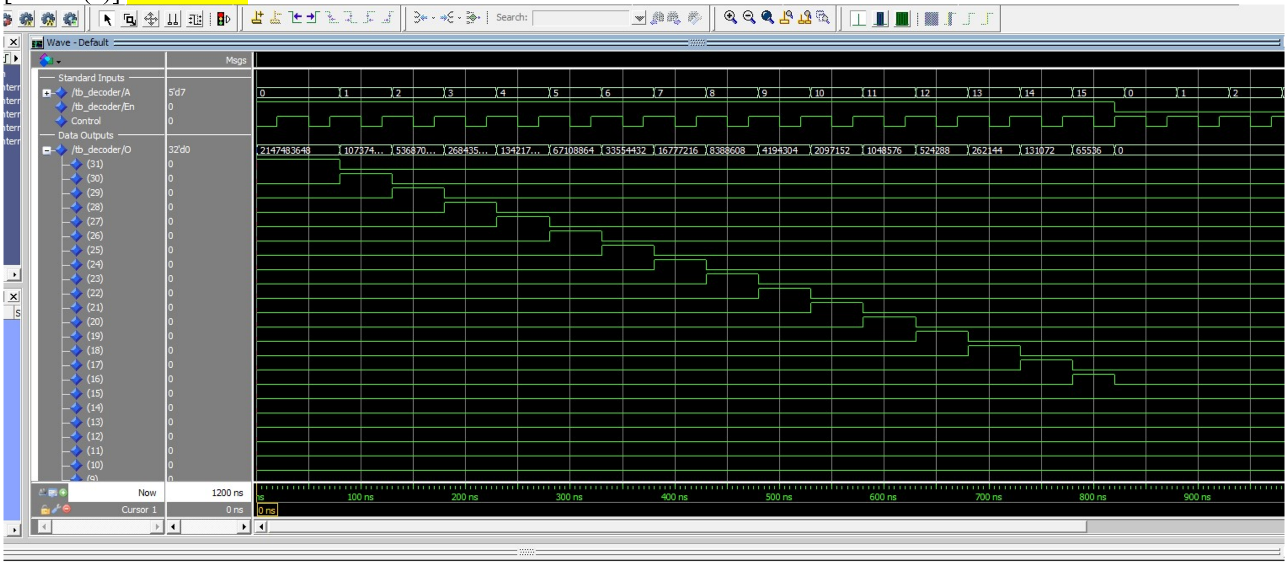
[Part 2 (c)] Waveform.



[Part 2 (d)] What type of decoder would be required by the MIPS register file and why?

In order to control the mips register file you would need a 5 to 32 bit decoder. This is because the decoder would be used to control the write enable pins on all of the registers, and there would need to be 5 bits in order to specify 32 possible locations.

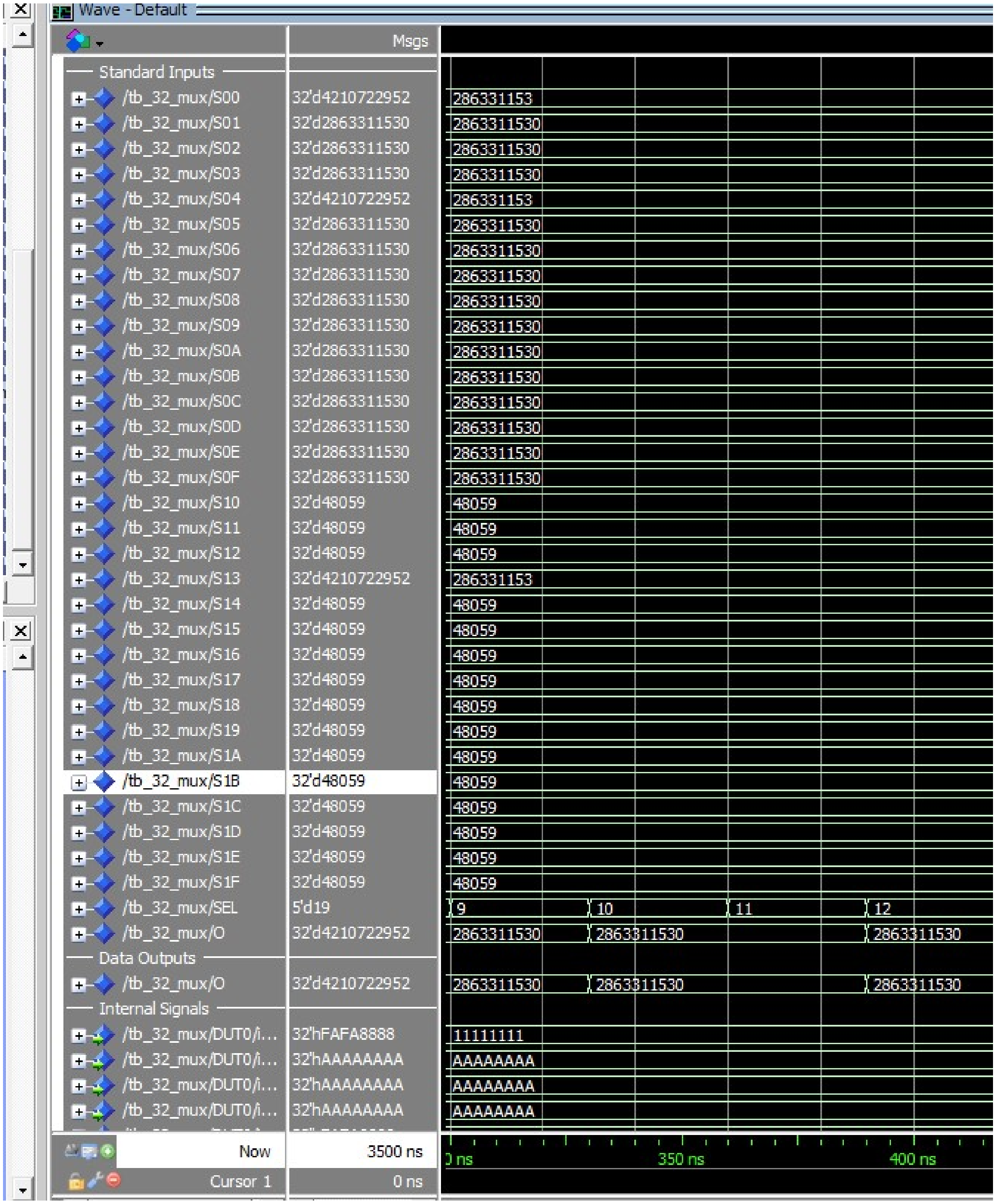
[Part 2 (e)] Waveform.



[Part 2 (f)] In your write-up, describe and defend the design you intend on implementing for the next part.

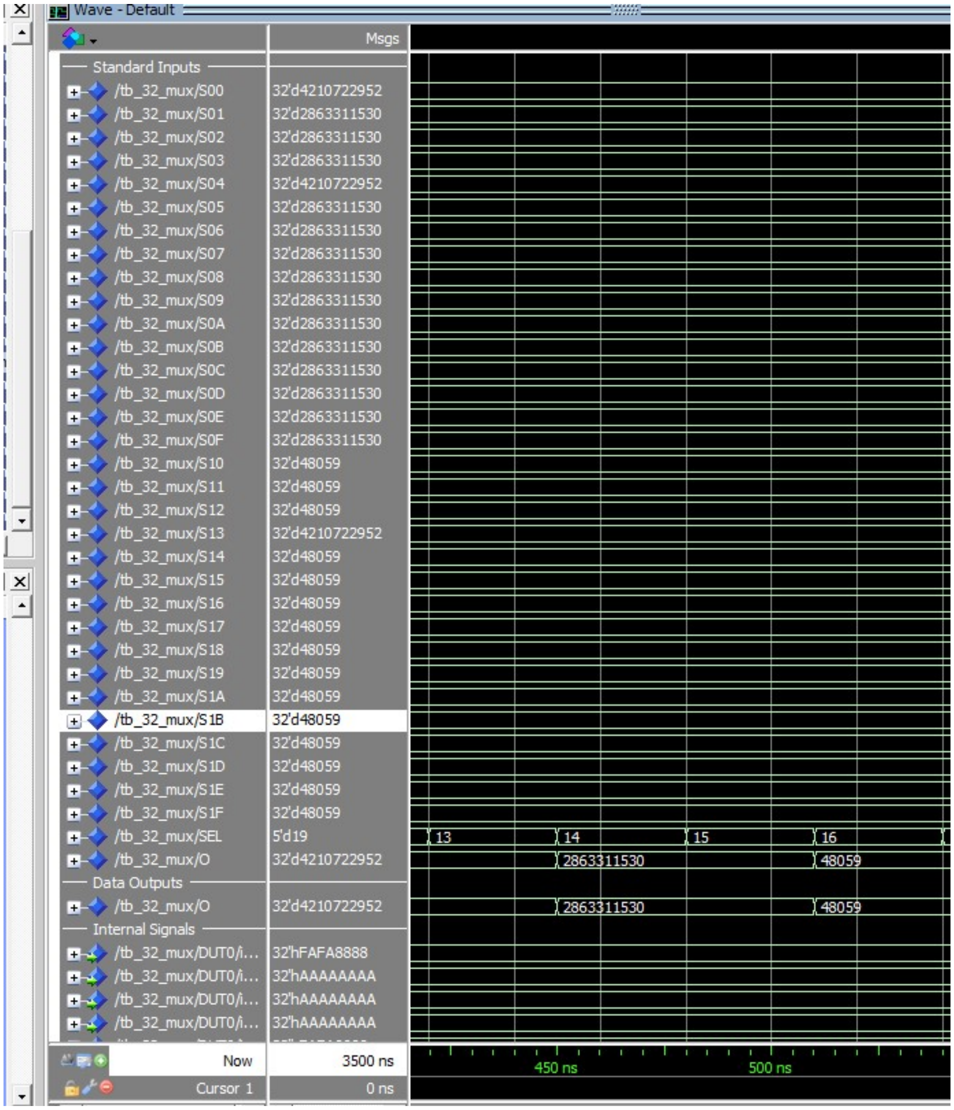
I plan to make a 32bit 32:1 multiplexer via a structural approach. I think this is the best idea as it provides the most intuition for how it will physically work, and I also don’t know how to do anything else.

[Part 2 (g)] Waveform.

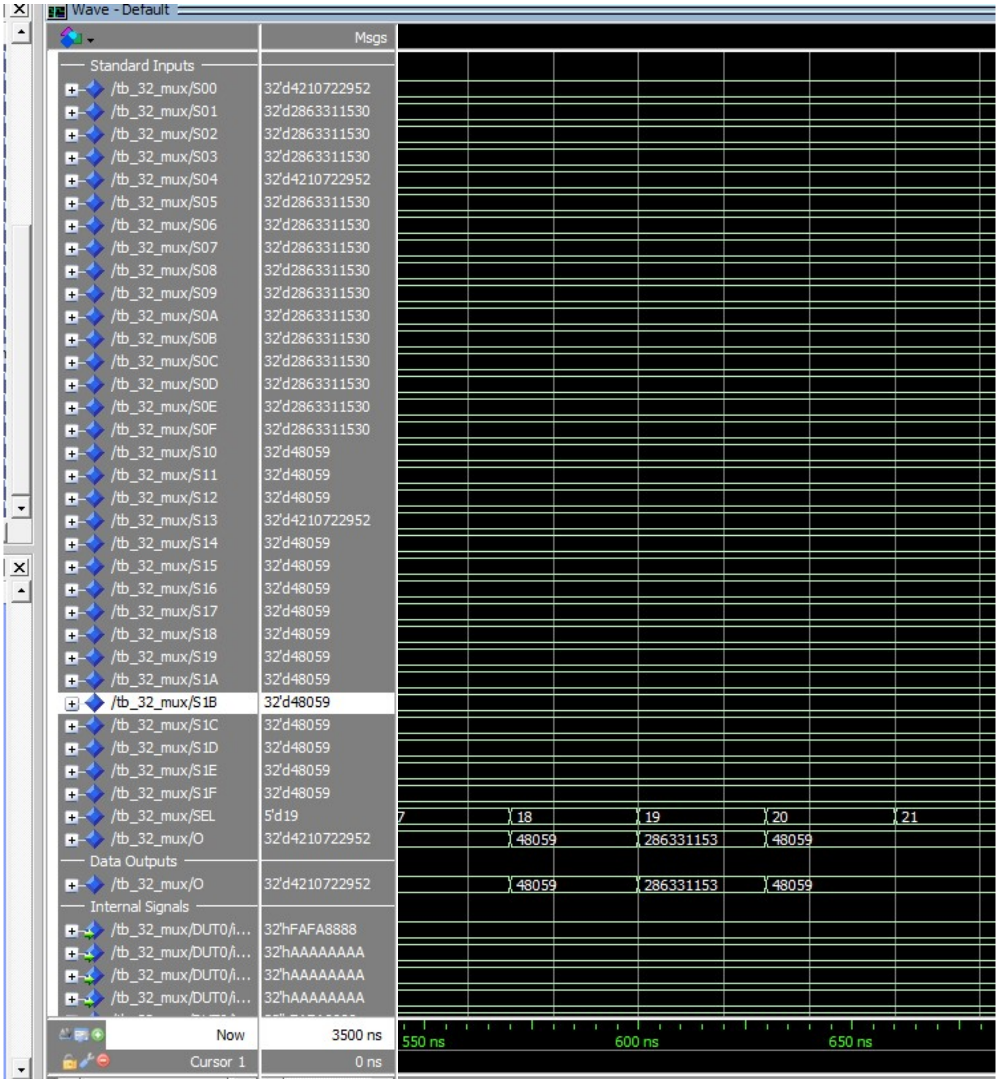


```
— Standard Inputs —
/tb_32_mux/S00        32'd4210722952   286331153
/tb_32_mux/S01        32'd2863311530   2863311530
/tb_32_mux/S02        32'd2863311530   2863311530
/tb_32_mux/S03        32'd2863311530   2863311530
/tb_32_mux/S04        32'd4210722952   286331153
/tb_32_mux/S05        32'd2863311530   2863311530
/tb_32_mux/S06        32'd2863311530   2863311530
/tb_32_mux/S07        32'd2863311530   2863311530
/tb_32_mux/S08        32'd2863311530   2863311530
/tb_32_mux/S09        32'd2863311530   2863311530
/tb_32_mux/S0A        32'd2863311530   2863311530
/tb_32_mux/S0B        32'd2863311530   2863311530
/tb_32_mux/S0C        32'd2863311530   2863311530
/tb_32_mux/S0D        32'd2863311530   2863311530
/tb_32_mux/S0E        32'd2863311530   2863311530
/tb_32_mux/S0F        32'd2863311530   2863311530
/tb_32_mux/S10        32'd48059        48059
/tb_32_mux/S11        32'd48059        48059
/tb_32_mux/S12        32'd48059        48059
/tb_32_mux/S13        32'd4210722952   286331153
/tb_32_mux/S14        32'd48059        48059
/tb_32_mux/S15        32'd48059        48059
/tb_32_mux/S16        32'd48059        48059
/tb_32_mux/S17        32'd48059        48059
/tb_32_mux/S18        32'd48059        48059
/tb_32_mux/S19        32'd48059        48059
/tb_32_mux/S1A        32'd48059        48059
/tb_32_mux/S1B        32'd48059        48059
/tb_32_mux/S1C        32'd48059        48059
/tb_32_mux/S1D        32'd48059        48059
/tb_32_mux/S1E        32'd48059        48059
/tb_32_mux/S1F        32'd48059        48059
/tb_32_mux/SEL        5'd19            9            10            11            12
/tb_32_mux/O          32'd4210722952   2863311530   2863311530                  2863311530
— Data Outputs —
/tb_32_mux/O          32'd4210722952   2863311530   2863311530                  2863311530
— Internal Signals —
/tb_32_mux/DUT0/i...  32'hFAFA8888     11111111
/tb_32_mux/DUT0/i...  32'hAAAAAAAA     AAAAAAAA
/tb_32_mux/DUT0/i...  32'hAAAAAAAA     AAAAAAAA
/tb_32_mux/DUT0/i...  32'hAAAAAAAA     AAAAAAAA

Now        3500 ns    0 ns          350 ns          400 ns
Cursor 1   0 ns
```

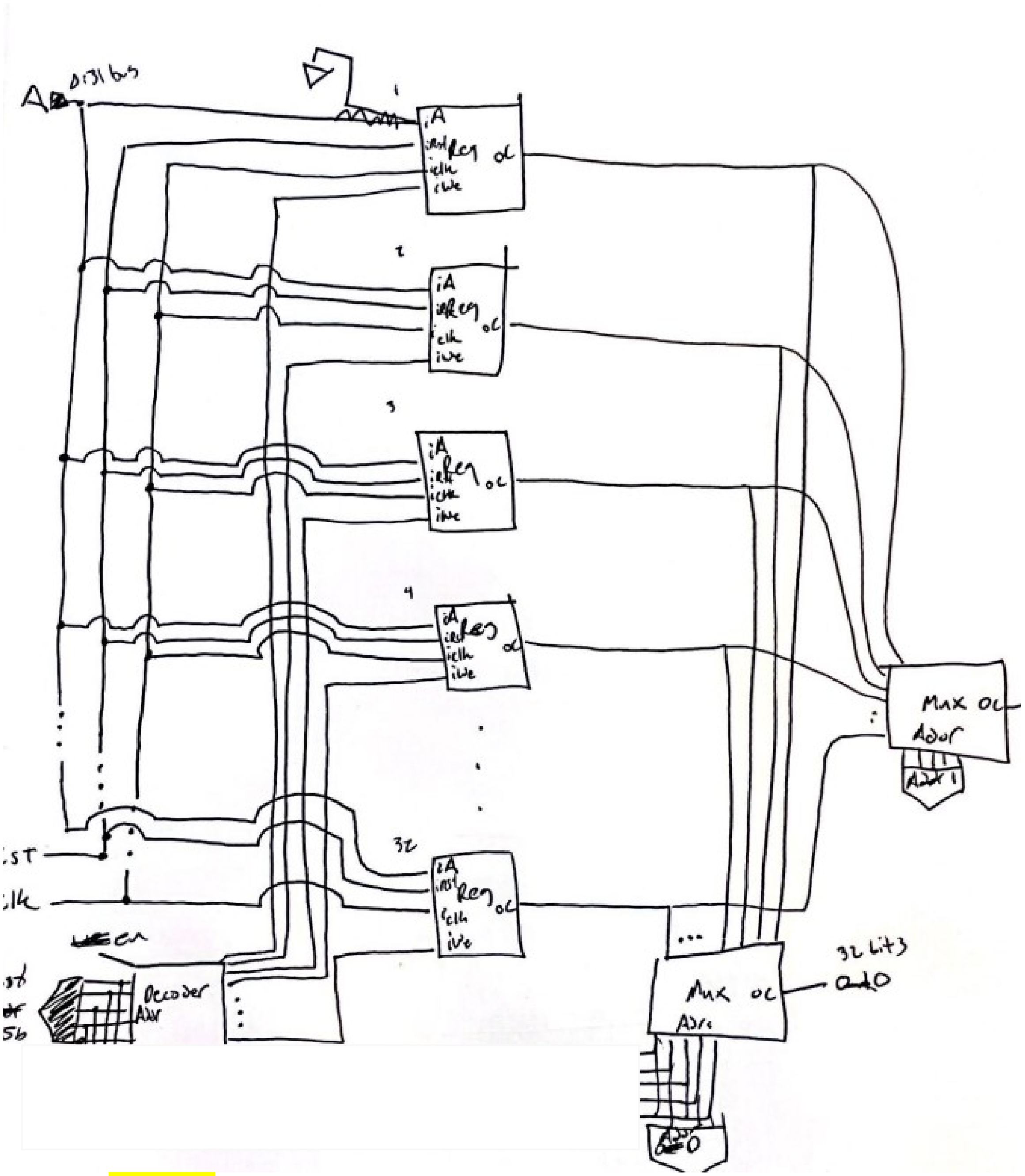


```
Wave - Default
                          Msgs
Standard Inputs
/tb_32_mux/S00            32'd4210722952
/tb_32_mux/S01            32'd2863311530
/tb_32_mux/S02            32'd2863311530
/tb_32_mux/S03            32'd2863311530
/tb_32_mux/S04            32'd4210722952
/tb_32_mux/S05            32'd2863311530
/tb_32_mux/S06            32'd2863311530
/tb_32_mux/S07            32'd2863311530
/tb_32_mux/S08            32'd2863311530
/tb_32_mux/S09            32'd2863311530
/tb_32_mux/S0A            32'd2863311530
/tb_32_mux/S0B            32'd2863311530
/tb_32_mux/S0C            32'd2863311530
/tb_32_mux/S0D            32'd2863311530
/tb_32_mux/S0E            32'd2863311530
/tb_32_mux/S0F            32'd2863311530
/tb_32_mux/S10            32'd48059
/tb_32_mux/S11            32'd48059
/tb_32_mux/S12            32'd48059
/tb_32_mux/S13            32'd4210722952
/tb_32_mux/S14            32'd48059
/tb_32_mux/S15            32'd48059
/tb_32_mux/S16            32'd48059
/tb_32_mux/S17            32'd48059
/tb_32_mux/S18            32'd48059
/tb_32_mux/S19            32'd48059
/tb_32_mux/S1A            32'd48059
/tb_32_mux/S1B            32'd48059
/tb_32_mux/S1C            32'd48059
/tb_32_mux/S1D            32'd48059
/tb_32_mux/S1E            32'd48059
/tb_32_mux/S1F            32'd48059
/tb_32_mux/SEL            5'd19
/tb_32_mux/O              32'd4210722952
Data Outputs
/tb_32_mux/O              32'd4210722952
Internal Signals
/tb_32_mux/DUT0/i...      32'hFAFA8888
/tb_32_mux/DUT0/i...      32'hAAAAAAAA
/tb_32_mux/DUT0/i...      32'hAAAAAAAA
/tb_32_mux/DUT0/i...      32'hAAAAAAAA
Now                       3500 ns
Cursor 1                  0 ns
```

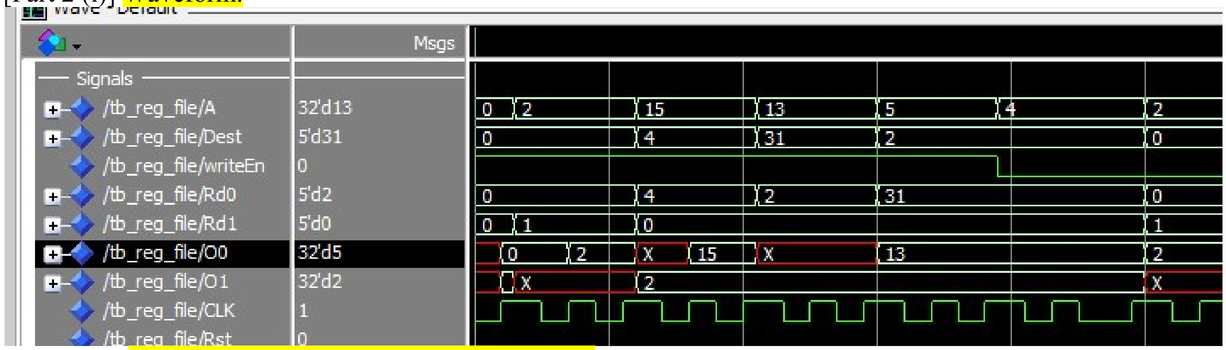


```
Standard Inputs
/tb_32_mux/S00        32'd4210722952
/tb_32_mux/S01        32'd2863311530
/tb_32_mux/S02        32'd2863311530
/tb_32_mux/S03        32'd2863311530
/tb_32_mux/S04        32'd4210722952
/tb_32_mux/S05        32'd2863311530
/tb_32_mux/S06        32'd2863311530
/tb_32_mux/S07        32'd2863311530
/tb_32_mux/S08        32'd2863311530
/tb_32_mux/S09        32'd2863311530
/tb_32_mux/S0A        32'd2863311530
/tb_32_mux/S0B        32'd2863311530
/tb_32_mux/S0C        32'd2863311530
/tb_32_mux/S0D        32'd2863311530
/tb_32_mux/S0E        32'd2863311530
/tb_32_mux/S0F        32'd2863311530
/tb_32_mux/S10        32'd48059
/tb_32_mux/S11        32'd48059
/tb_32_mux/S12        32'd48059
/tb_32_mux/S13        32'd4210722952
/tb_32_mux/S14        32'd48059
/tb_32_mux/S15        32'd48059
/tb_32_mux/S16        32'd48059
/tb_32_mux/S17        32'd48059
/tb_32_mux/S18        32'd48059
/tb_32_mux/S19        32'd48059
/tb_32_mux/S1A        32'd48059
/tb_32_mux/S1B        32'd48059
/tb_32_mux/S1C        32'd48059
/tb_32_mux/S1D        32'd48059
/tb_32_mux/S1E        32'd48059
/tb_32_mux/S1F        32'd48059
/tb_32_mux/SEL        5'd19
/tb_32_mux/O          32'd4210722952
Data Outputs
/tb_32_mux/O          32'd4210722952
Internal Signals
/tb_32_mux/DUT0/i...  32'hFAFA8888
/tb_32_mux/DUT0/i...  32'hAAAAAAAA
/tb_32_mux/DUT0/i...  32'hAAAAAAAA
/tb_32_mux/DUT0/i...  32'hAAAAAAAA
Now                   3500 ns
Cursor 1              0 ns
```

[Part 2 (h)] Draw a (simplified) schematic (i.e., components within the high-level blackbox) for the MIPS register file, using the same top-level interface ports as in your solution describe above and using only the register, decoder, and mux VHDL components you have created.

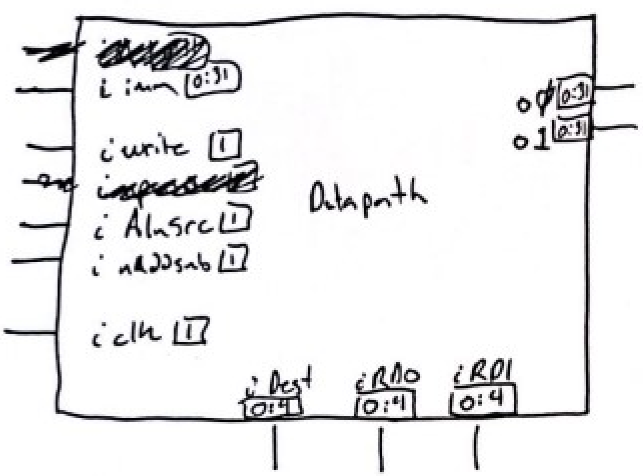


[Part 2 (i)] Waveform.

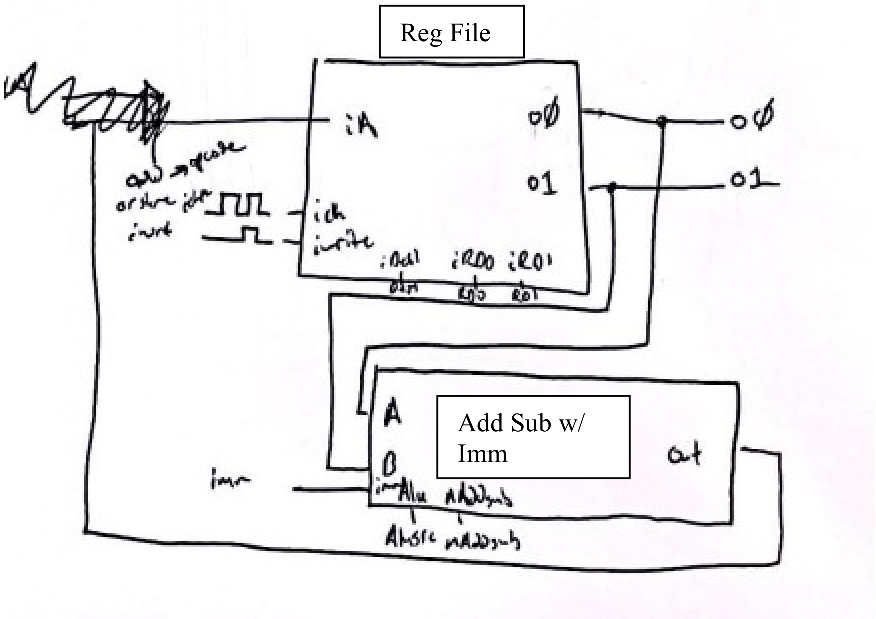


Signals	Msgs
/tb_reg_file/A	32'd13
/tb_reg_file/Dest	5'd31
/tb_reg_file/writeEn	0
/tb_reg_file/Rd0	5'd2
/tb_reg_file/Rd1	5'd0
/tb_reg_file/O0	32'd5
/tb_reg_file/O1	32'd2
/tb_reg_file/CLK	1
/tb_reg_file/Rst	0

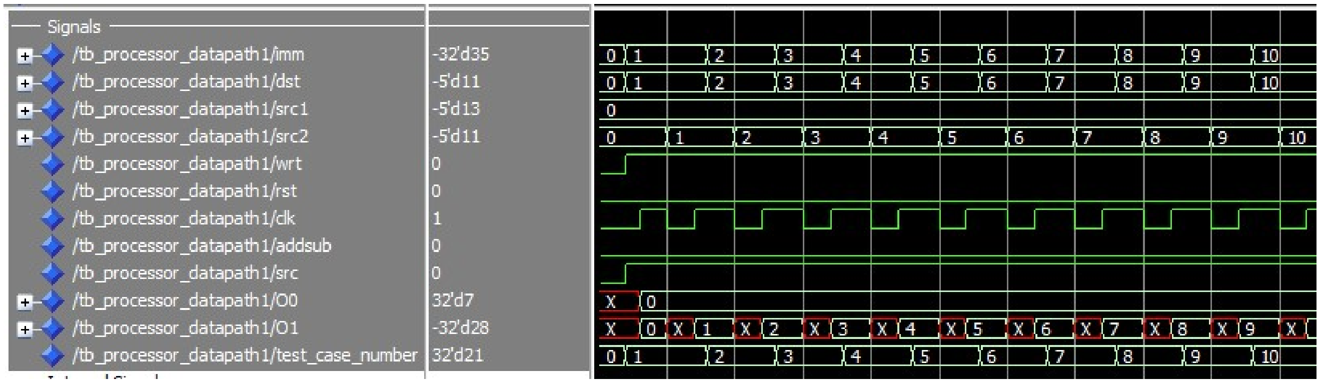
[Part 3 (b)] Draw a symbol for this MIPS-like datapath.



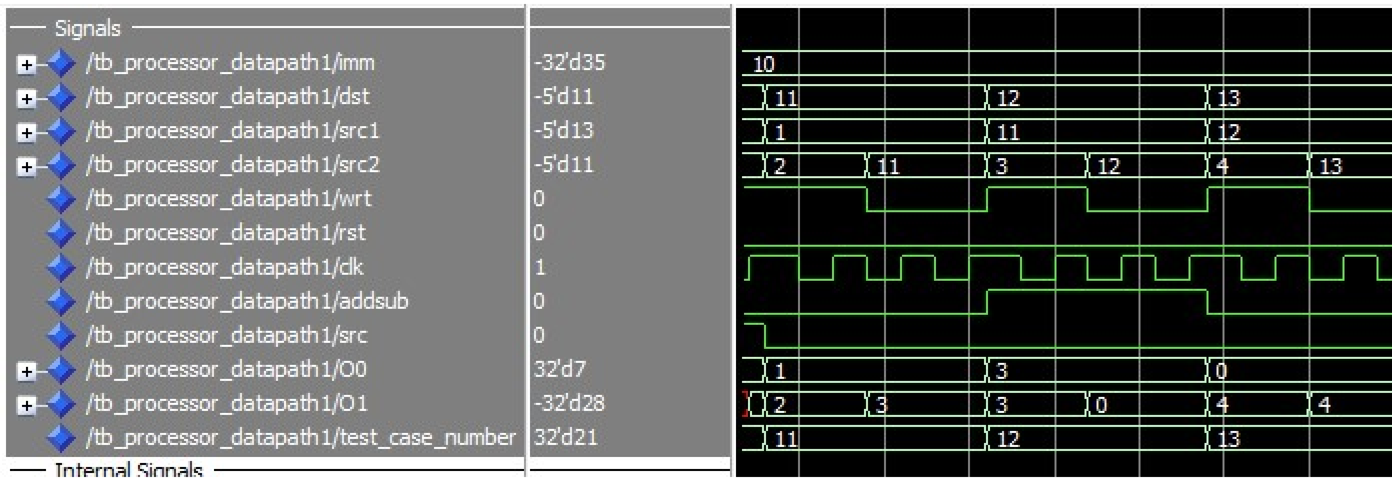
[Part 3 (c)] Draw a schematic of the simplified MIPS processor datapath consisting only of the component described in part (a) and the register file from problem (1).



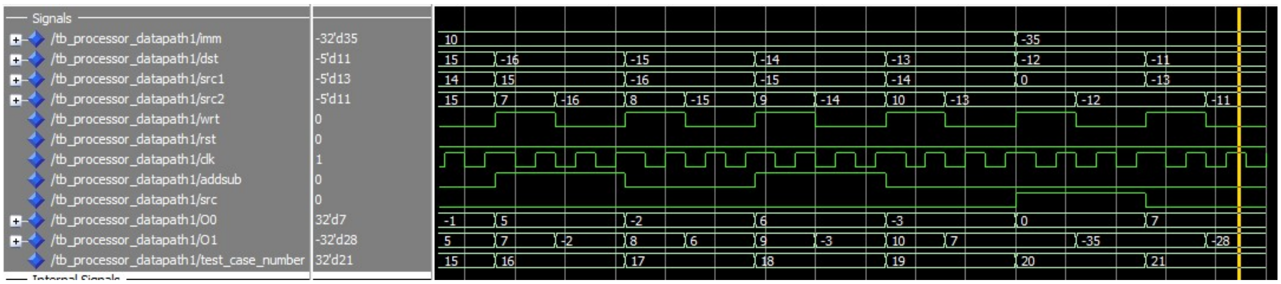
[Part 3 (d)] Include in your report waveform screenshots that demonstrate your properly functioning design. Annotate what the final register file state should be.



```
Signals
/tb_processor_datapath1/imm               -32'd35
/tb_processor_datapath1/dst               -5'd11
/tb_processor_datapath1/src1              -5'd13
/tb_processor_datapath1/src2              -5'd11
/tb_processor_datapath1/wrt               0
/tb_processor_datapath1/rst               0
/tb_processor_datapath1/clk               1
/tb_processor_datapath1/addsub            0
/tb_processor_datapath1/src               0
/tb_processor_datapath1/O0                32'd7
/tb_processor_datapath1/O1                -32'd28
/tb_processor_datapath1/test_case_number  32'd21
```



```
Signals
/tb_processor_datapath1/imm               -32'd35
/tb_processor_datapath1/dst               -5'd11
/tb_processor_datapath1/src1              -5'd13
/tb_processor_datapath1/src2              -5'd11
/tb_processor_datapath1/wrt               0
/tb_processor_datapath1/rst               0
/tb_processor_datapath1/clk               1
/tb_processor_datapath1/addsub            0
/tb_processor_datapath1/src               0
/tb_processor_datapath1/O0                32'd7
/tb_processor_datapath1/O1                -32'd28
/tb_processor_datapath1/test_case_number  32'd21
Internal Signals
```



```
Signals
/tb_processor_datapath1/imm               -32'd35
/tb_processor_datapath1/dst               -5'd11
/tb_processor_datapath1/src1              -5'd13
/tb_processor_datapath1/src2              -5'd11
/tb_processor_datapath1/wrt               0
/tb_processor_datapath1/rst               0
/tb_processor_datapath1/clk               1
/tb_processor_datapath1/addsub            0
/tb_processor_datapath1/src               0
/tb_processor_datapath1/O0                32'd7
/tb_processor_datapath1/O1                -32'd28
/tb_processor_datapath1/test_case_number  32'd21
Internal Signals
```

The final value should be -35 + 7 which is -28, which is observed in the bottom right of the output graph.

[Part 4 (a)] Read through the mem.vhd file, and based on your understanding of the VHDL implementation, provide a 2-3 sentence description of each of the individual ports (both generic and regular).

Data width: How many bits does a register store?

Addr_width: How many bits can we use to address registers?

clk: The clock signal is used to clock the registers and
instruct them to move to the next position. Clock is a standard input
in most processes that deal with memory.

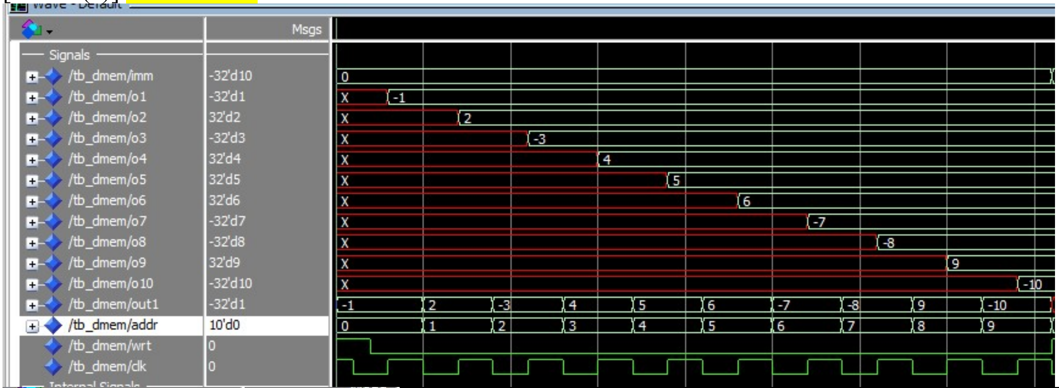
addr: The addr port dictates what address we wish to talk to. Similar to
the reg file it will dictate where we store or retrieve data to.

data: If we are trying to write data to the memory, this is the data we are
passing in.

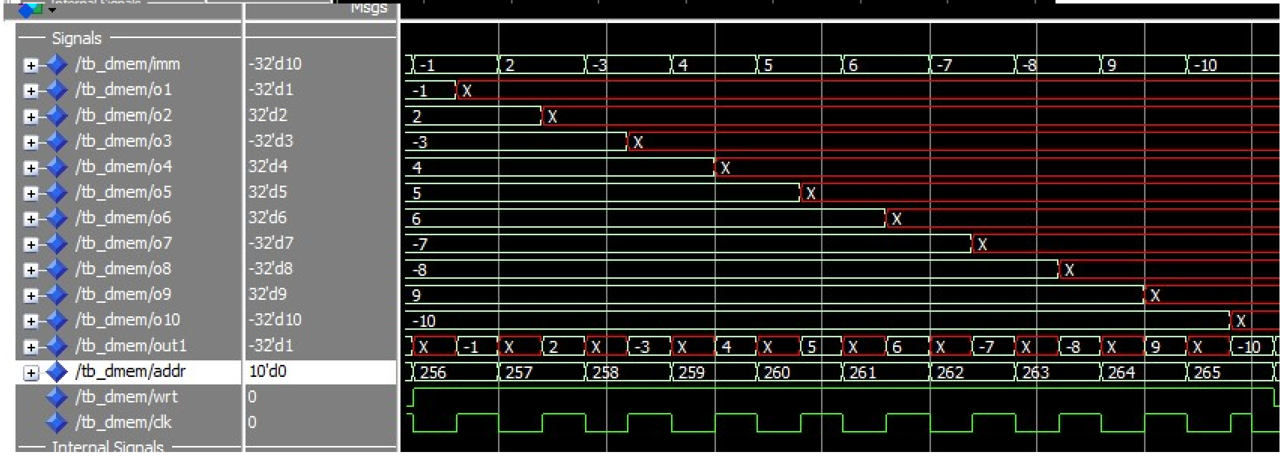
we: This is a write enable pin. If we wish to write to a register, it must be enabled.
Versus if we want to read from a register but not write it will not be enabled.

q: This is the data output of the memory. 1 clock cycle after setting an addr, this
port should read the contents of that address.

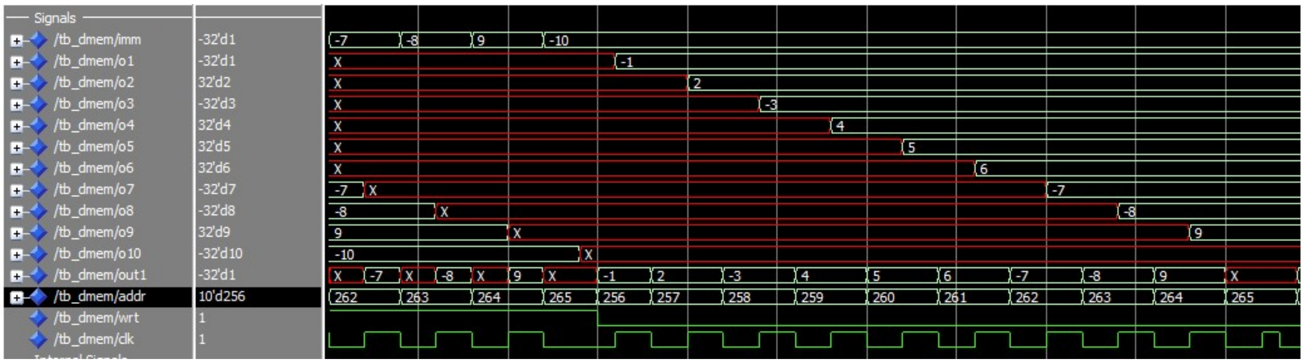
[Part 4 (c)] Waveforms.



Signals	Msgs
/tb_dmem/imm	-32'd10
/tb_dmem/o1	-32'd1
/tb_dmem/o2	32'd2
/tb_dmem/o3	-32'd3
/tb_dmem/o4	32'd4
/tb_dmem/o5	32'd5
/tb_dmem/o6	32'd6
/tb_dmem/o7	-32'd7
/tb_dmem/o8	-32'd8
/tb_dmem/o9	32'd9
/tb_dmem/o10	-32'd10
/tb_dmem/out1	-32'd1
/tb_dmem/addr	10'd0
/tb_dmem/wrt	0
/tb_dmem/clk	0



Signals	Msgs
/tb_dmem/imm	-32'd10
/tb_dmem/o1	-32'd1
/tb_dmem/o2	32'd2
/tb_dmem/o3	-32'd3
/tb_dmem/o4	32'd4
/tb_dmem/o5	32'd5
/tb_dmem/o6	32'd6
/tb_dmem/o7	-32'd7
/tb_dmem/o8	-32'd8
/tb_dmem/o9	32'd9
/tb_dmem/o10	-32'd10
/tb_dmem/out1	-32'd1
/tb_dmem/addr	10'd0
/tb_dmem/wrt	0
/tb_dmem/clk	0



Signals	
/tb_dmem/imm	-32'd1
/tb_dmem/o1	-32'd1
/tb_dmem/o2	32'd2
/tb_dmem/o3	-32'd3
/tb_dmem/o4	32'd4
/tb_dmem/o5	32'd5
/tb_dmem/o6	32'd6
/tb_dmem/o7	-32'd7
/tb_dmem/o8	-32'd8
/tb_dmem/o9	32'd9
/tb_dmem/o10	-32'd10
/tb_dmem/out1	-32'd1
/tb_dmem/addr	10'd256
/tb_dmem/wrt	1
/tb_dmem/clk	1

[Part 5 (a)] What are the MIPS instructions that require some value to be sign extended? What are the MIPS instructions that require some value to be zero extended?

What instructions require some value to be sign extended?
addi
slti
lb-8bit
lh-16bit

What instructions need to be zero extended?
addiu
sltiu
lbu
lhu

[Part 5 (b)] what are the different 16-bit to 32-bit "extender" components that would be required by a MIPS processor implementation?

What are the different extender components that would be

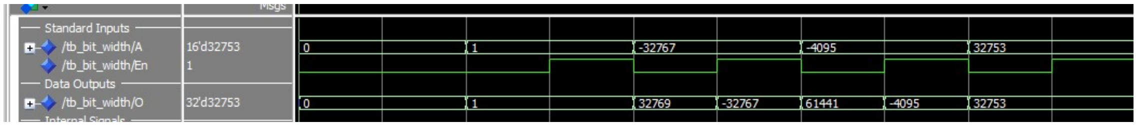
required by a MIPS processor?

You would need an extender that can either sign or zero extend,

and you would need a version that could do so for either 8

or 16 bit inputs.

[Part 5 (d)] Waveform.



Standard Inputs

/tb_bit_width/A	16'd32753
/tb_bit_width/En	1

Data Outputs

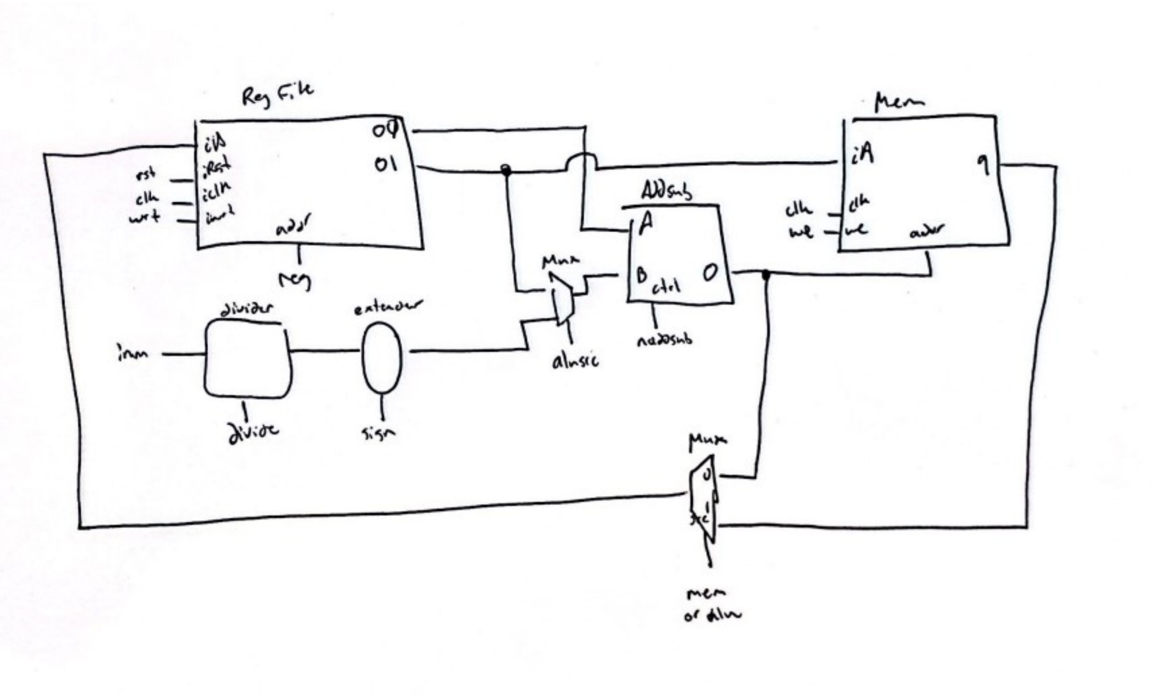
/tb_bit_width/O	32'd32753

[Part 6 (a)] what control signals will need to be added to the simple processor from part 2? How do these control signals correspond to the ports on the mem.vhd component analyzed in part 3?

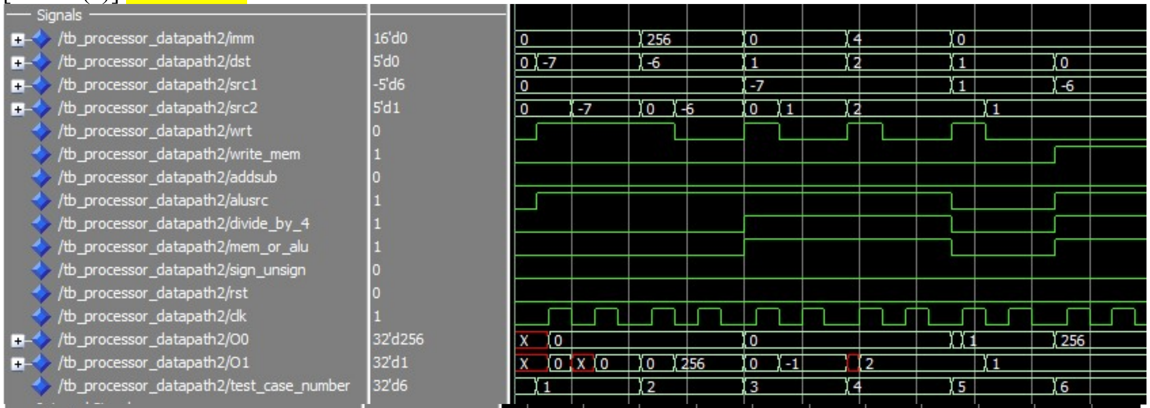
In addition to the control signals last time (write on registers, alusrc, naddsub, rst), we will also need write on memory, use memory or alu, signed or unsigned, and divide by 4 control bits.

The write on memory bit is used to control the write enable pin on the memory.

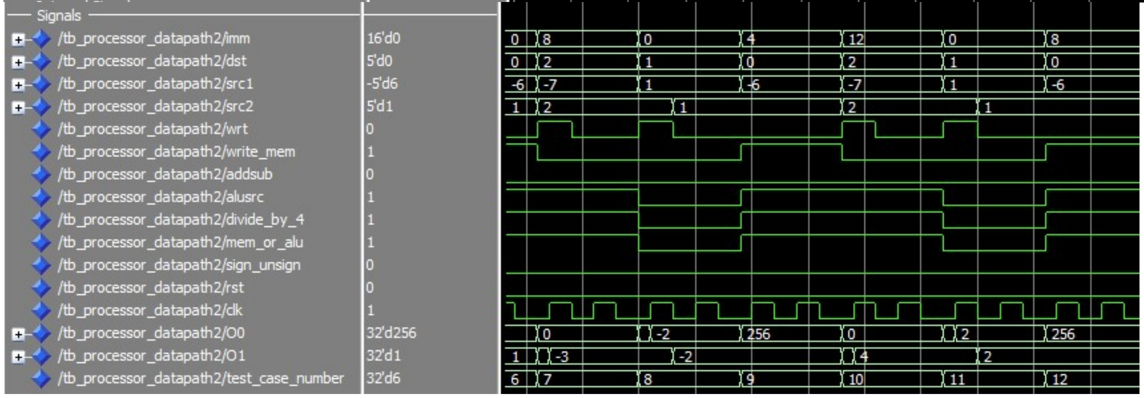
[Part 6 (b)] Draw a schematic of a simplified MIPS processor consisting only of the base components used in part 2, the extender component described in part 4, and the data memory from part 3.



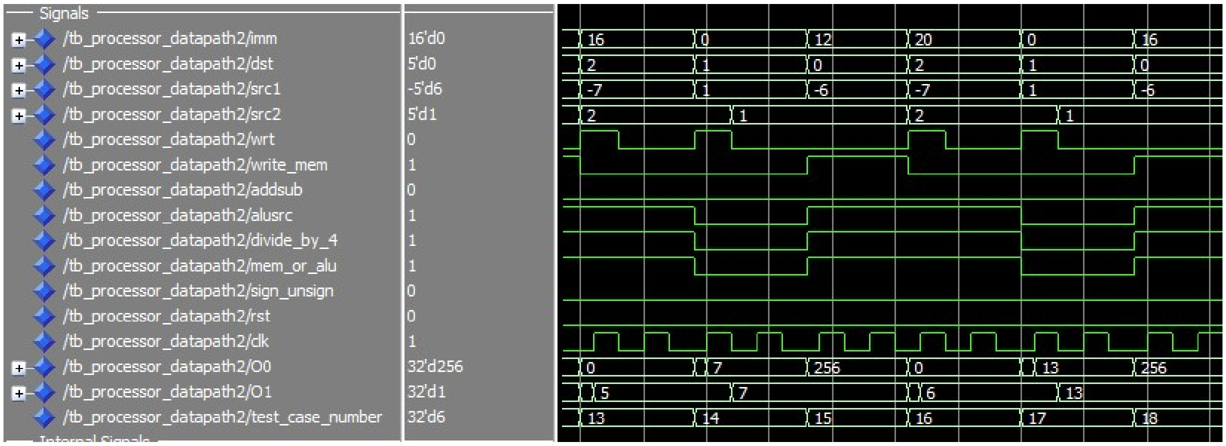
[Part 6 (c)] Waveform.



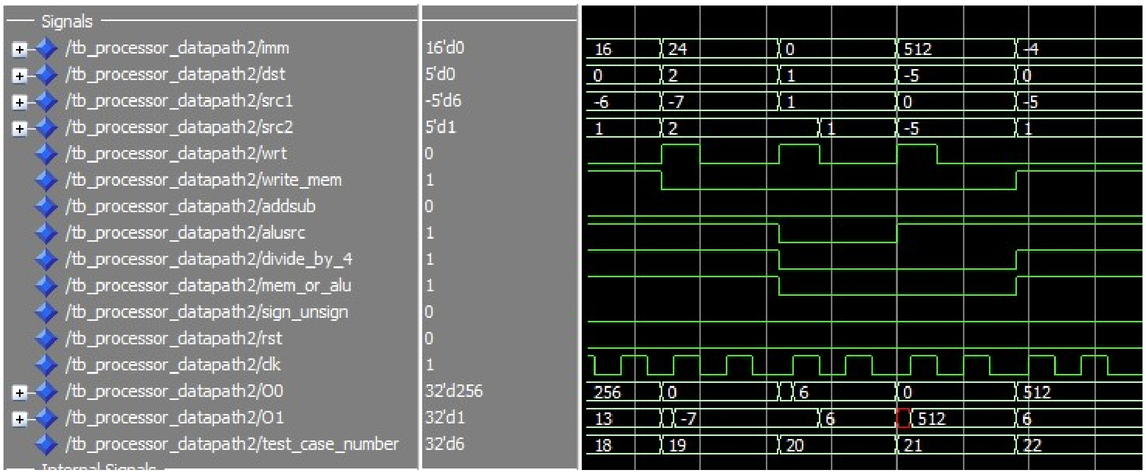
Signals	
/tb_processor_datapath2/imm	16'd0
/tb_processor_datapath2/dst	5'd0
/tb_processor_datapath2/src1	-5'd6
/tb_processor_datapath2/src2	5'd1
/tb_processor_datapath2/wrt	0
/tb_processor_datapath2/write_mem	1
/tb_processor_datapath2/addsub	0
/tb_processor_datapath2/alusrc	1
/tb_processor_datapath2/divide_by_4	1
/tb_processor_datapath2/mem_or_alu	1
/tb_processor_datapath2/sign_unsign	0
/tb_processor_datapath2/rst	0
/tb_processor_datapath2/clk	1
/tb_processor_datapath2/O0	32'd256
/tb_processor_datapath2/O1	32'd1
/tb_processor_datapath2/test_case_number	32'd6



Signals	
/tb_processor_datapath2/imm	16'd0
/tb_processor_datapath2/dst	5'd0
/tb_processor_datapath2/src1	-5'd6
/tb_processor_datapath2/src2	5'd1
/tb_processor_datapath2/wrt	0
/tb_processor_datapath2/write_mem	1
/tb_processor_datapath2/addsub	0
/tb_processor_datapath2/alusrc	1
/tb_processor_datapath2/divide_by_4	1
/tb_processor_datapath2/mem_or_alu	1
/tb_processor_datapath2/sign_unsign	0
/tb_processor_datapath2/rst	0
/tb_processor_datapath2/clk	1
/tb_processor_datapath2/O0	32'd256
/tb_processor_datapath2/O1	32'd1
/tb_processor_datapath2/test_case_number	32'd6



```
Signals
/tb_processor_datapath2/imm                16'd0
/tb_processor_datapath2/dst                5'd0
/tb_processor_datapath2/src1               -5'd6
/tb_processor_datapath2/src2               5'd1
/tb_processor_datapath2/wrt                0
/tb_processor_datapath2/write_mem          1
/tb_processor_datapath2/addsub             0
/tb_processor_datapath2/alusrc             1
/tb_processor_datapath2/divide_by_4        1
/tb_processor_datapath2/mem_or_alu         1
/tb_processor_datapath2/sign_unsign        0
/tb_processor_datapath2/rst                0
/tb_processor_datapath2/clk                1
/tb_processor_datapath2/O0                 32'd256
/tb_processor_datapath2/O1                 32'd1
/tb_processor_datapath2/test_case_number   32'd6
Internal Signals
```



```
Signals
/tb_processor_datapath2/imm                16'd0
/tb_processor_datapath2/dst                5'd0
/tb_processor_datapath2/src1               -5'd6
/tb_processor_datapath2/src2               5'd1
/tb_processor_datapath2/wrt                0
/tb_processor_datapath2/write_mem          1
/tb_processor_datapath2/addsub             0
/tb_processor_datapath2/alusrc             1
/tb_processor_datapath2/divide_by_4        1
/tb_processor_datapath2/mem_or_alu         1
/tb_processor_datapath2/sign_unsign        0
/tb_processor_datapath2/rst                0
/tb_processor_datapath2/clk                1
/tb_processor_datapath2/O0                 32'd256
/tb_processor_datapath2/O1                 32'd1
/tb_processor_datapath2/test_case_number   32'd6
Internal Signals
```