Brian Reber
CprE 381 Lab3

Prelab:

Ch 6, Q 5

```
entity eight_or is
      port( data   : in std_logic_vector(7 downto 0);
            output : out std_logic);
end eight_or;

architecture behavior of eight_or is
begin
      mOr : process( data )
      begin
            if (data = "00000000") then output <= 0;
            else output <= 1;
            end if;
      end process mOr;
end behavior;
```

Ch 9, Q 2

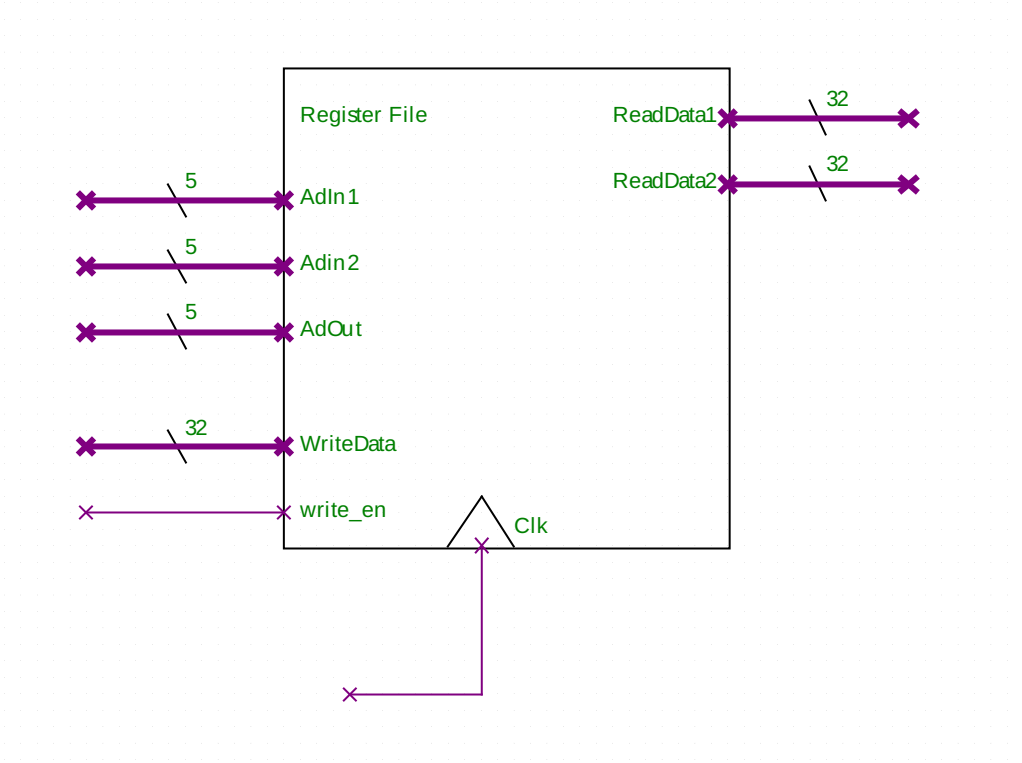
```
entity d_ff_sr is
      port( D,R  : in std_logic;
            CLK  : in std_logic
            Q,NQ : out std_logic);
end d_ff_sr;

architecture behavior of d_ff_sr is
begin
      mDff : process( D,R,CLK )
      begin
            if (S = '0') then
                  Q  <= '1';
                  NQ <= '0';
            elsif (R = '0') then
                  Q  <= '0';
                  NQ <= '1';
            elsif (rising_edge(CLK)) then
                  Q  <= D;
                  NQ <= not D;
            end if;
      end process mDff;
end behavior;
```

Group 9

Scott Connell – 6
Arjay Vander Velden – 6
Brian Reber – 6

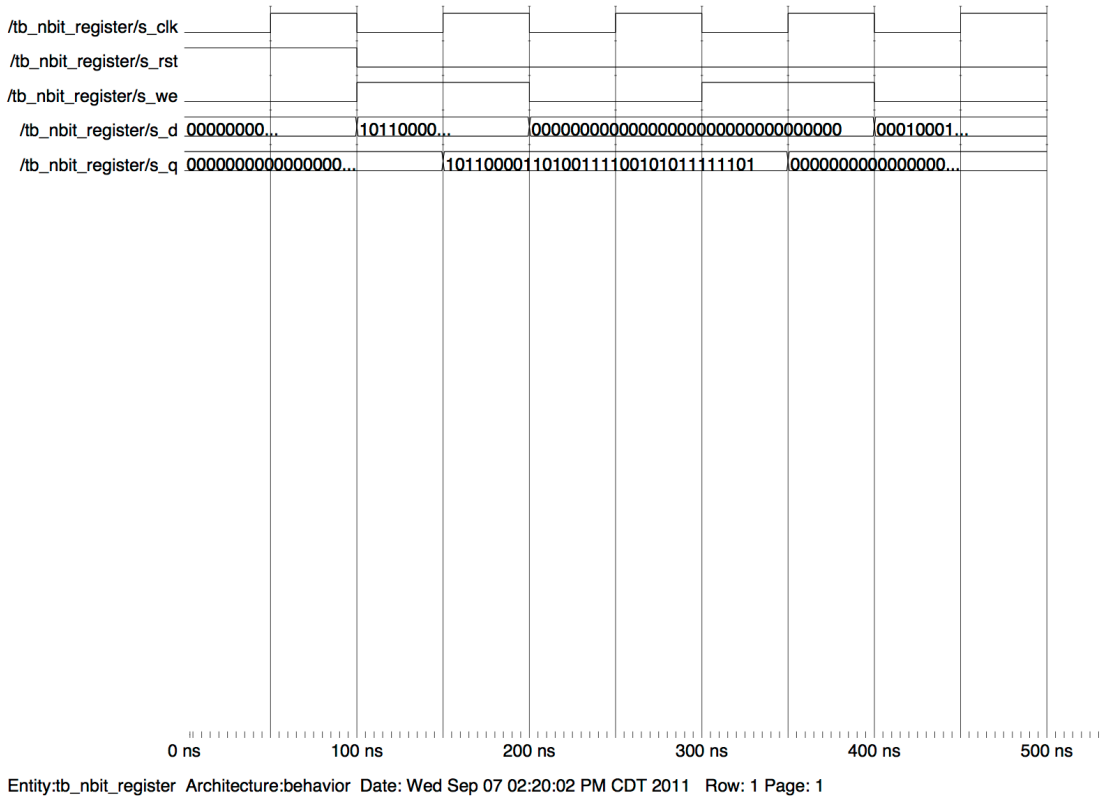
Part 1

a) We will need a port for the clock, as writing to the register file is a synchronous operation. This just needs to be a single bit. We will need three 5-bit ports for determining which of the 32 registers (2^5 = 32) we want to access (two for reading, and one for a destination/writing). We will need 3 data ports of width 32 bits, for our 32-bit data from the 32-bit registers.



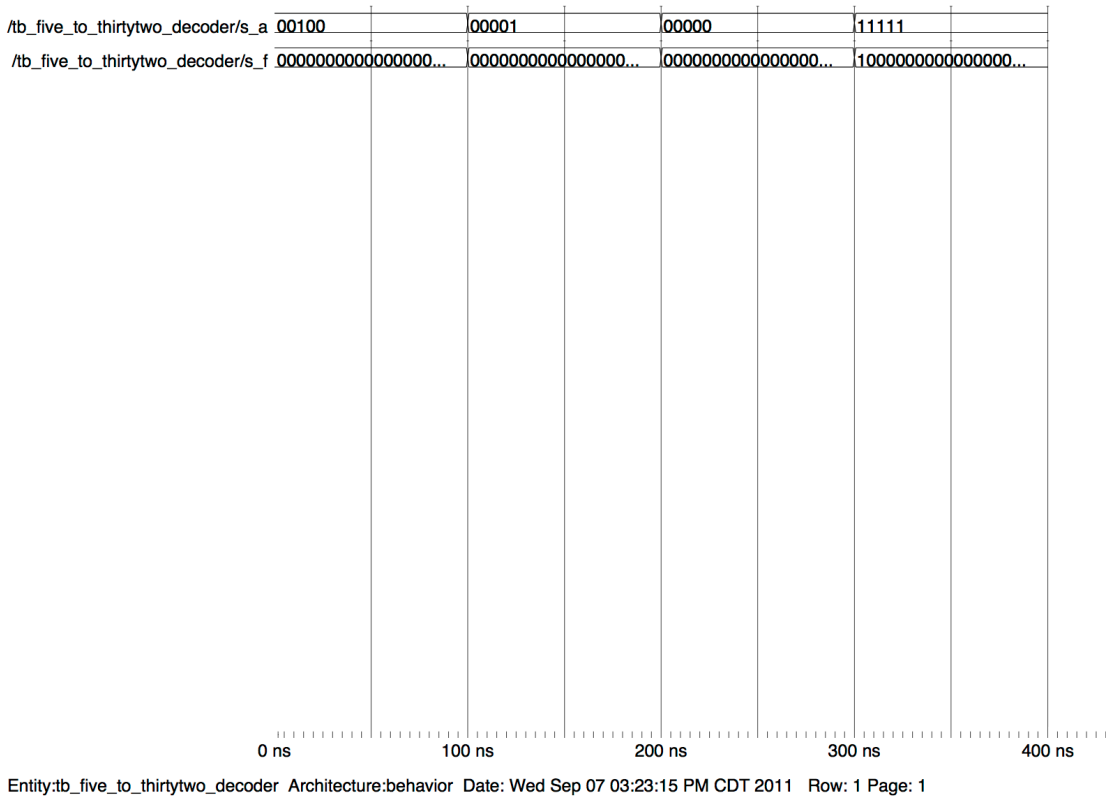
b) No questions

c) See the testbench for this problem. The waveform generated is here:

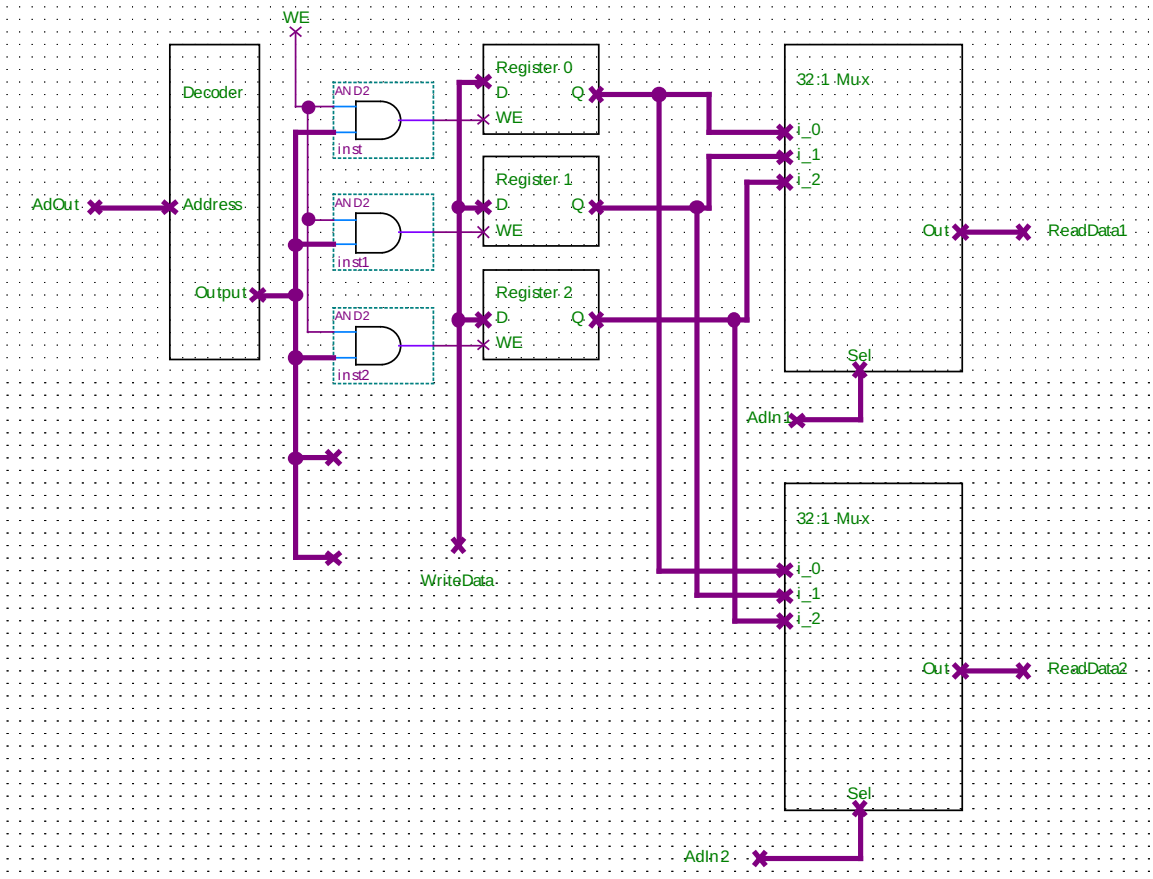


d) For accessing the register file, we will want a 5:32 decoder. This will allow us to uniquely access each of the 32 registers. Since there are 32 registers, we will need the decoder to decode to 32 bits, and the easiest way to do that is a 5:32 decoder (2^5 = 32).

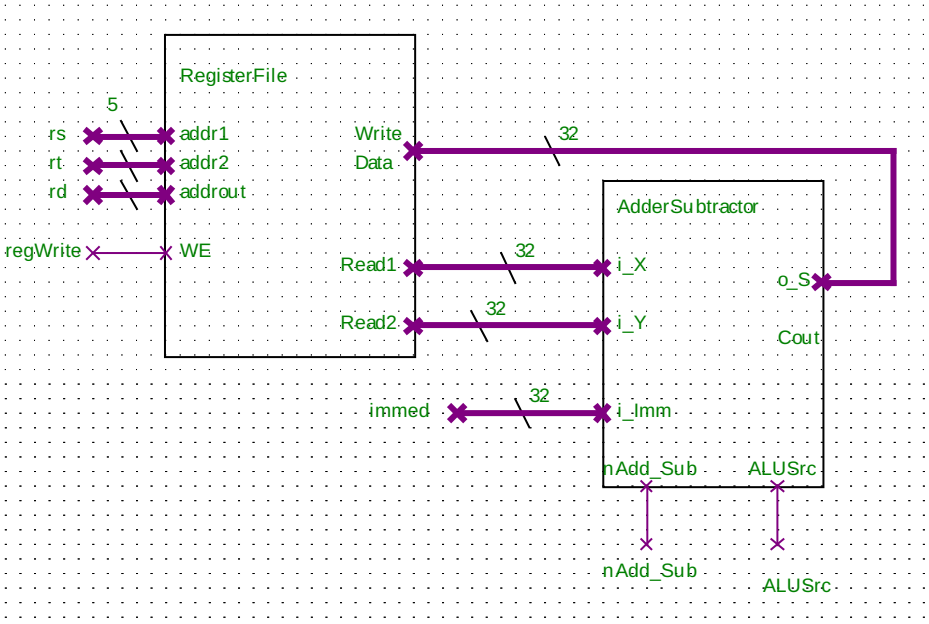
e) This is the waveform for the testbench included in the src folder.

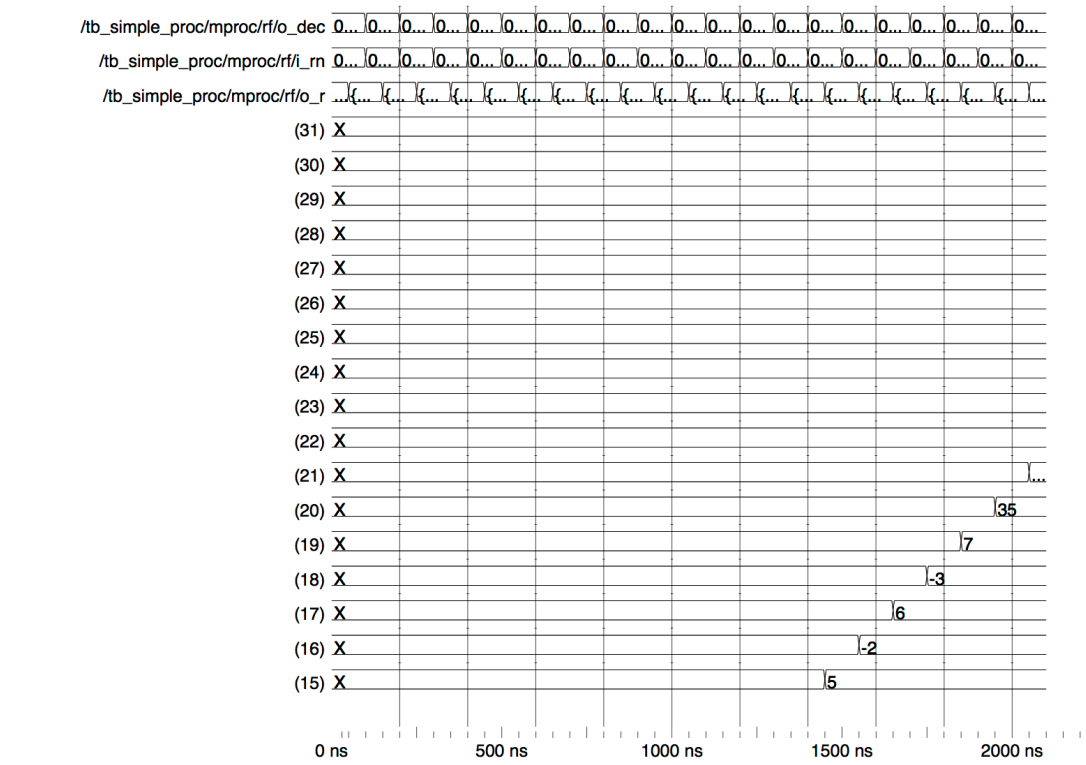


f) I chose to implement the 32 bit 32:1 mux using dataflow VHDL. This method means that I will assign the value of the output to one of the inputs depending on the value of the input selection signal. This seemed to be the simplest design as there wasn't a lot of repetition needed to connect ports.
g) N/A
h) Below is the circuit diagram for this.



Schematic for simplified MIPS processor:





Entity:tb_simple_proc Architecture:behavior Date: Mon Sep 12 11:26:39 AM CDT 2011 Row: 1 Page: 1