All,

Make sure for Lab 6 that you specify a 32-bit input for the **B Reg** and the **A Reg** (just like **Acc Reg** which has its own 32-bit input come from the 32-bit 2's compliment adder output); these 32-bit inputs should be loaded into the registers (in parallel) if the load signal is high.

For example: say we had an 8-bit shift register that had the ability to shift or load (like the 32-bit shift register we have in this lab) and we have the following values:

**8-bit input** - 01010101

**Current 8-bit register value** - 00000010

**Shift Right Signal** - 1

**Load Signal** - 0

Then the new value of the 8-bit register on the next CLK pulse should equal **00000001** (we shift to right by one). Now, if we instead had the following signals:

**8-bit input** - 01010101

**Current 8-bit register value** - 00000010

**Shift Right Signal** - 0

**Load Signal** - 1

Notice now we have the load signal high instead of the shift signal. Then the new value of the 8-bit register on the next CLK pulse should equal **01010101** (the value at the 8-bit input that we load into the register).

Also, the **B Reg** needs a 32-bit output as well; this output should represent the value of the 32-bit reg variable that you create inside this module. **Acc. Reg** also needs a 32-bit output as well as it feeds into one of the inputs of the 32-bit 2's Compliment Adder; this output should represent the value of the 32-bit reg variable that you create inside this module. We need these two 32-bit outputs from **Acc. Reg** and **B reg** because (in the end) these two outputs concatenated will be our 64-bit result of the Booth's signed multiply algorithm.

Hope that clears up some of the common questions that I had gotten from you all for this lab. If you still don't quite understand, feel free to email me what you would like to clarify further.