

Lab # 2 FAQ's:

Q: What should we turn in, in addition to the lab 2 specs?

A: You should turn in your graded lab 1, along with your complete lab 2.

Q: Lab #1, question 15 asks you to write a program to compute the remainder (or modulo) of two 8-bit unsigned integers A and B. The algorithm provided doesn't work for all cases. What should we do?

A: You should use a 9 bit register for your mod sequence of instructions at the least, or conversely, you could make all of your registers 9 bits wide, and thus all of your arithmetic operations 9 bit wide as well. Following is a revised algorithm for the mod problem:

```
byte modulo (byte A, byte B)
{
byte result=0;
byte count=0;

// catch the error case (you don't need to do this)
if (B==0) return A;

// align B by shifting up
while (A>=B){ /** Before, here is A>B **/
B = B << 1;
count++;};

// overshoot, so go back one, probably can avoid
// this in the lower level design
if (count >0){ count--; B = B >> 1;};

// successive subtractions until we get B back to
// where it was originally, then we're done
while (count>=0){ /** Before, here is count > 0 **/
if (A>=B) {A=A-B;}
count--;
B = B >> 1;
}
return A;
}
```

Q: Are we allowed to implement 16 bit registers even though instructions and memory accessed will only be 8 bits?

A: if its not explicitly prohibited in the assignment this is okay.

Q: What built in Xilinx macros can we use?

A: This is covered in the lab 2 spec. You can ONLY use registers, and instruction/data memories that are available through LogiBlox utility provided by Xilinx.

Q: If we are not allowed to use anything built in, then what can we use?

A: You can use any basic logic unit, such as AND, OR, XOR, NOT, and other BASIC logic gates. Using these gates, you can build anything, and everything more complex, through a careful use of macros YOU have created, to build up more complex units. For example, to build a ripple carry adder, you would: make a half adder macro -> using the half adder above, you would make a full adder macro -> then, you would combine these full adders in series, to make a ripple carry adder, that you can then use in your ALU to add, or subtract.

Q: Can we use any other tool except Xilinx, for example LogicWorks?

A: No, for these labs you have to use Xilinx tools.

Q: What is the point breakdown for the Lab 2 grading? And which TA graded which part?

A: TBD