Lecture outline

- Subroutines
- Adding I/O to the Simple Computer
- SHIFT instruction
- Logic analyzer and disassembly
- **Debugging**

Lab 8 common problem

- In lab 8 prelab and early lab, you:
  - Add the shifter hardware and SHIFT instruction
  - Add IN and OUT instructions, including
    - adding the DECODE and EXECUTE states
    - adding the tri-state driver hardware
    - controlling IO_CYCLE and IO_WRITE
  - Add new signals to SCOMP's PORT statement
  - Write an assembly program from scratch
- All with nearly no intermediate testing!

Lab 8 step 7

- When you test your assembly program (lab step 7) it will almost certainly not work
- How do you find the problem?
- Isolate individual components!

Debugging process basics

- Start with the basics:
  - Do you have the correct Quartus project open?
  - Are you editing the correct asm file?
  - Is SCOMP's memory configured to use the correct mif file?
  - Is the PLL configured correctly (3/5 ratio)?

Debugging process example

- Start by making it easier to see what's going on by getting OUT working
- Isolate OUT:
  - Remove any INs, SHIFTs, and as much logic as possible from the assembly

```
Test:
    LOAD   Number
    OUT    LEDS
    JUMP   Test
Number: DW &B1010101
```

OUT not working

- If the above doesn't work, then there is most likely a problem with OUT
  - (but not absolutely; it could be other things)
- What does OUT need to work?
  - Correct DECODE => EX_OUT => EX_OUT2
  - IO_CYCLE, IO_WRITE, and IO_WRITE_INT
  - LPM_BUSTRI device (tri-state driver)

OUT working

- Once OUT is working, you can use it to test IN:

```
Test:
    IN     SWITCHES
    OUT    LEDS
    JUMP   Test
```

- Then test SHIFT
- Then move on to your full assembly code for step 6/7.