# Computer Organization

## Lab: Single-Cycle Datapath

Purpose

Learn how to implement instructions for a CPU.

Method

Implement the datapath for a subset of the MIPS instruction set architecture described in the textbook.

Files to Use

[datapath.circ](file:///C:\Users\springtj\Desktop\datapath.circ), [control.circ](file:///C:\Users\springtj\Desktop\control.circ), [cpu32.circ](file:///C:\Users\springtj\Desktop\cpu32.circ), [misc32.circ](file:///C:\Users\springtj\Desktop\misc32.circ), and [loop.mem](file:///C:\Users\springtj\Desktop\loop.mem).

What to Hand In

Completed datapath.circ, loop.mem, and table of control settings.

The file datapath.circ contains an incomplete implementation of the MIPS instruction set architecture (ISA). This datapath can execute the following instructions: add, sub, and, or, nor, slt, addi, lw, sw, and beq.

1. Complete the circuit for the datapath by adding appropriate multiplexors and logic gates and wiring them to the components that are already in place. Note that some of the wiring, such as the wiring for the instruction memory and data memory, is already done.
2. Connect the control lines from the Control and ALUcontrol subcircuits to the datapath as appropriate.
3. Translate the following assembly language program into hexadecimal machine language. (You may want to use the SPIM simulator to assist you with this task, but be careful.) Put the resulting machine language program (in hexadecimal) in the file [loop.mem](file:///C:\Users\springtj\Desktop\loop.mem).

loop: addi $t0, $zero, 17

sw $t0, 0($zero)

beq $zero, $zero, loop

1. Complete the table ([doc](file:///C:\Users\springtj\Desktop\control.doc)) showing how the control signals should be set for each instruction of the program above. Remember that ALUop is 2 bits and Operation is the 4-bit control for the ALU.
2. Test your datapath by executing a short program. You will have to manually set the control signals for the Control and ALUcontrol circuits. (When your assignment is graded, short programs will be run using working versions of the Control and ALUcontrol circuits.)
   1. Connect a 9-bit input to the top of the Control circuit.
   2. Connect a 4-bit input to the top of the ALUcontrol circuit.
   3. Reset the program counter to zero.
   4. Load your edited file loop.mem into the instruction memory.
   5. Set the control signals for the Control and ALUcontrol circuits to correspond to the first instruction of the program. In order to have the correct data stored in the registers or memory you must cause a rising edge on the appropriate control line after the other control lines have been set.
   6. Put the clock through one cycle to update the program counter.
   7. Go back to step 3 and repeat this procedure for the remaining instructions of the program.