Cpr E 381 Lab 10
Pipelined Datapath Implementation

Section 6.7, located on the CD that came with your book, builds a pipelined datapath in verilog using a behavioral description. The code in Figure 6.7.4 is a complete datapath in terms of control and data flow, but does not contain the code to implement a full instruction set. Take the code from this section as a starting point and complete a pipelined datapath.

Part 1 – Add instructions: The datapath does not contain the logic to implement all of the MIPS instructions. Modify the code from Figure 6.7.4 to support the following instructions:

R-Type Instructions	Function code
Add	100000 (already included)
sub	100010
and	100100
or	100101
slt	101010

I-Type Instructions	opcode
addi	001000
andi	001100
ori	001101
slti	001010

beq (already included)
bne

Pare 2 – Replace the instruction memory

The behavioral model uses an array of registers to implement the instruction memory. Instantiate a memory as in previous labs that can be initialized with a Memory Initialization File (MIF).

Part 3 – Run a small program on the pipelined datapath

1. Write a MIPS assembly program that will place the sum of the values between 1 and *i* in the memory location *i* for *i* = 1 to 10. You will need to convert the code to the binary encoding of the instructions chosen.

2. Simulate and verify the functionality of your program on the pipelined datapath.

    a. Load your program into the instruction memory.
    b. Add outputs from the main module that you can view in the waveform editor showing the data and address for memory write operations. Explain the time between writes in terms of the code you have written.