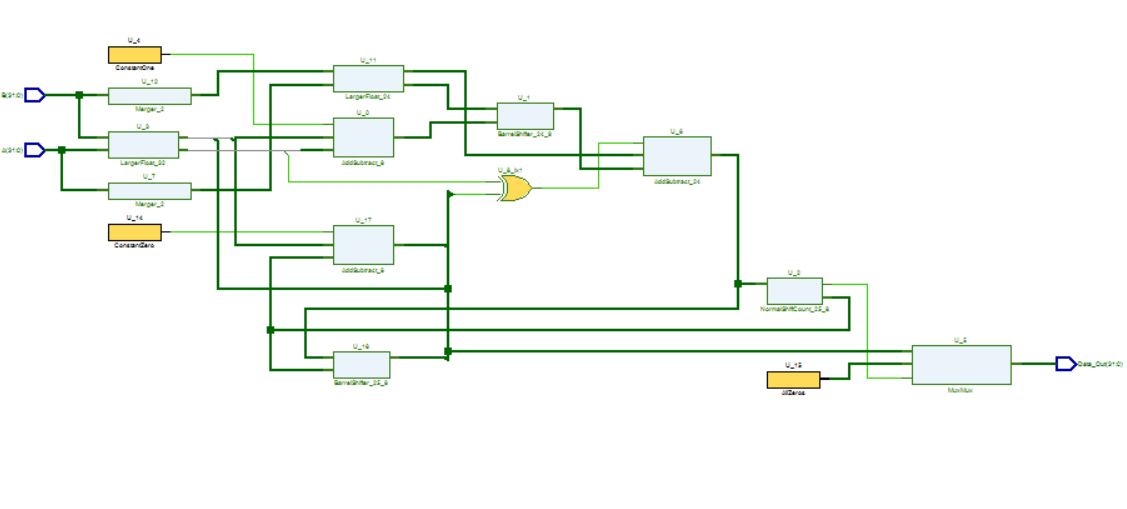
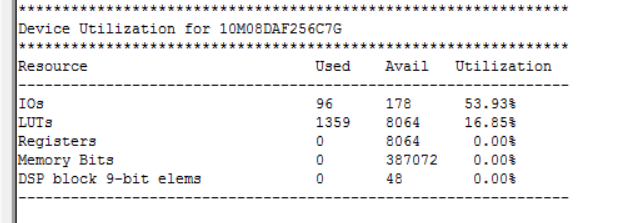
Jeffery You (jy9vm)

**Combinational Floating Point Adder**

*Schematic:*



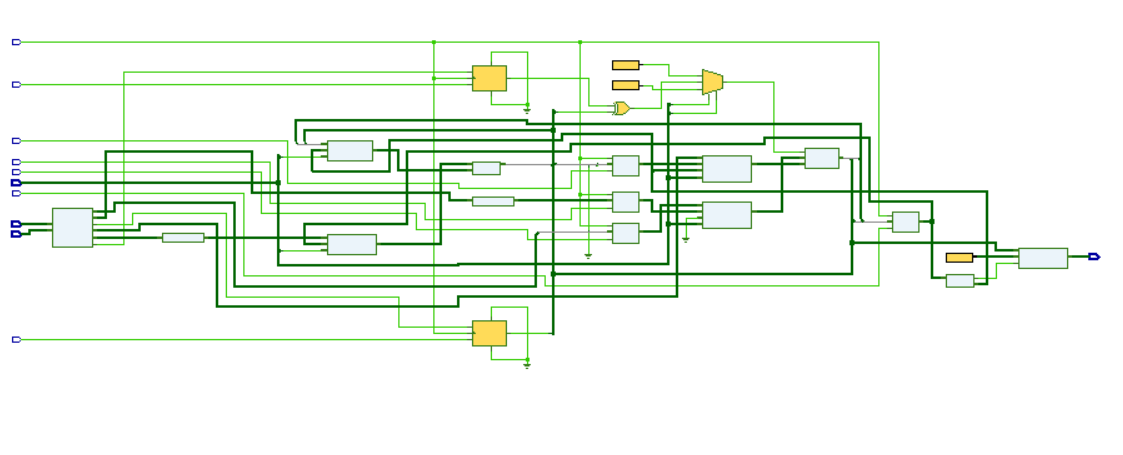
*Resource Requirements:*



There are no registers utilization for the FPGA requirement report for the combinational datapath.

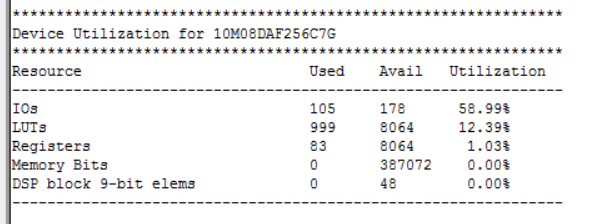
**Sequential Floating Point Data Path**

*Schematic:*



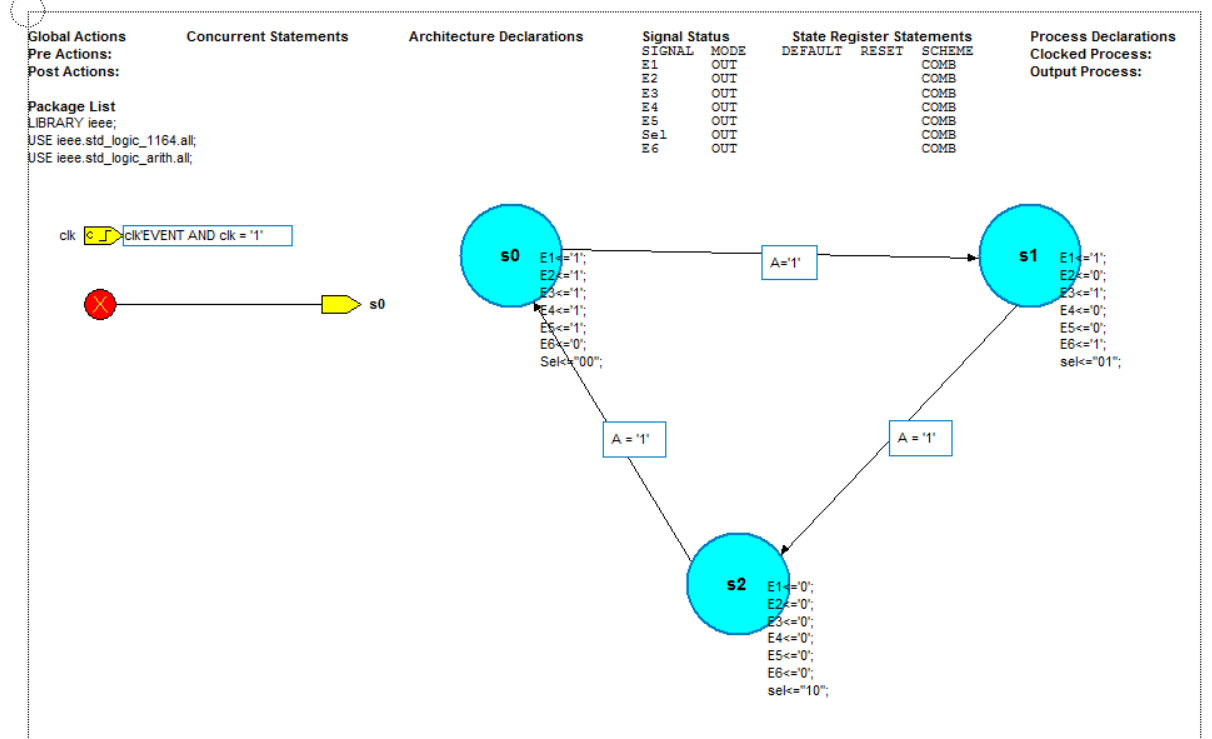
There are a lot of more wires associated with this design. It looks like the RTL made some of the MUXs boxes for some reason.

*Resource Requirements:*



There are now registers utilization now for the FPGA device.

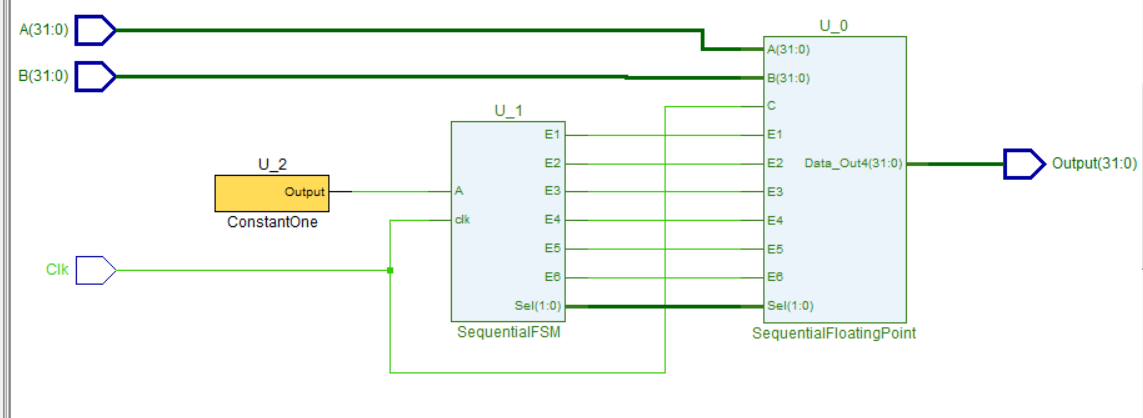
**Sequential Control Unit**



The FSM machine controls the enables on the registers to determine when the store and when to not store; these are my E’s. The Sel variable is the selector for the MUXs. The Sel variable changes depending on the clock cycle. It goes to “00” to “01” to “10” to indicate the cycle.

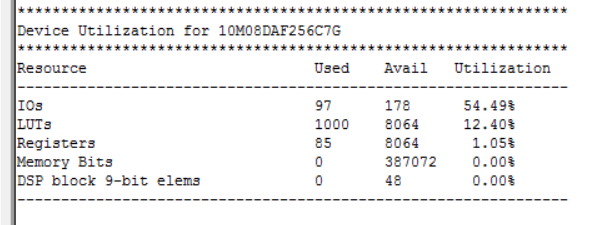
**Sequential Floating Point Adder (Data Path + Control Unit)**

*Schematic:*



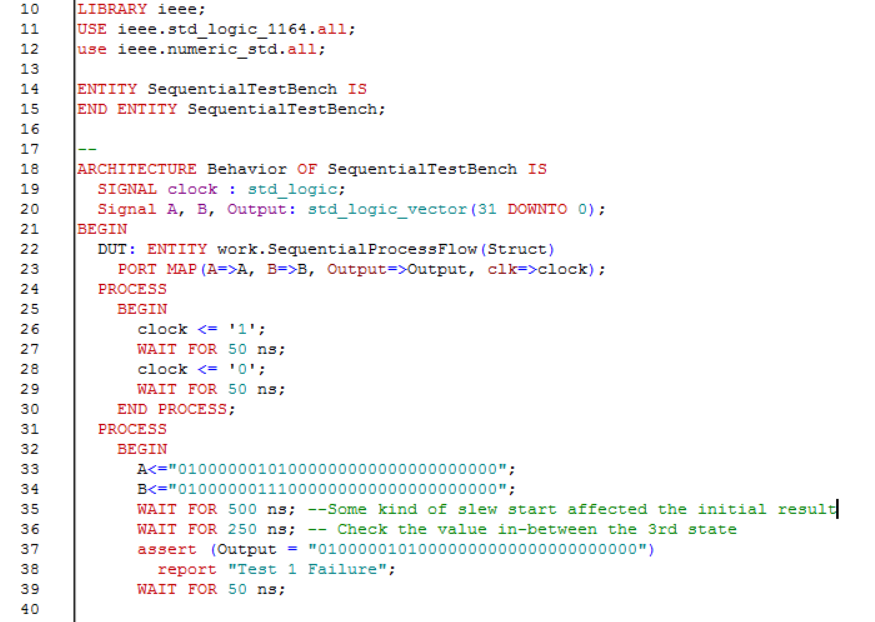
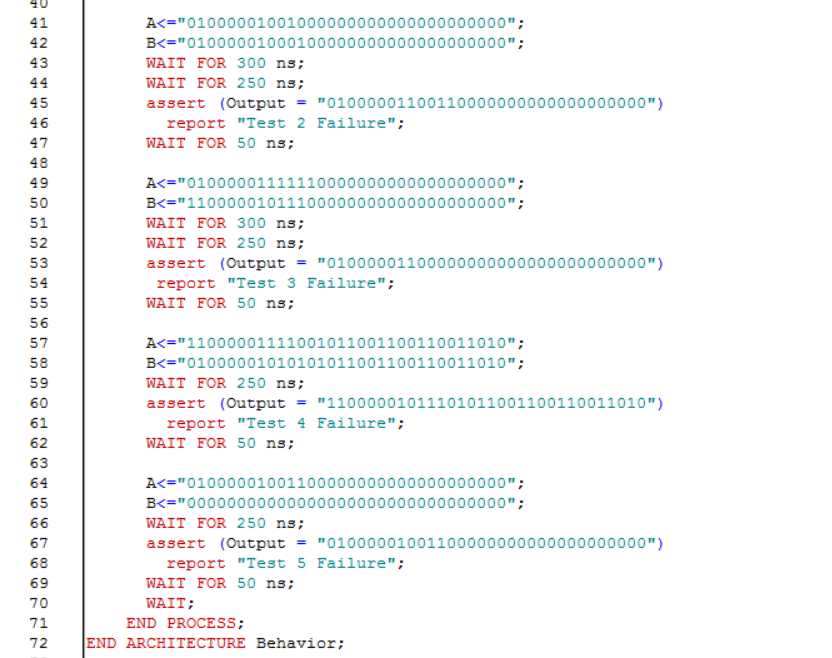
This schematic connects the FSM with the Sequential Datapath.

*Resource Requirements:*



Registers are now being used for the FPGA utilization.

**Sequential TestBench**



Testbench worked out well. I made the program report a test failure if there was a problem. I tested 5 different cases. Two different cases for adding two positive floating numbers, two different cases for adding two negative floating point numbers, one adding a negative and positive floating point numbers, and finally one adding a positive floating point number with a 0. I extend WAIT to 500 initially due to some clock skew delay for some reason that causes the first input entry not to register immediately. Within the cases, I extended to 6 clock cycles just in case the new immediate input entries does not register for the 1st 3 clock cycles. But the correct value is definitely given at the end of the 2nd 3 clock cycles. Also, I asserted within the middle of 3rd cycle and not towards the falling edge of the 3rd clock cycle to avoid any output changes when the input changes.