***Introduction***

This lab had four key tasks to it. The first task was to design and functionally verify a datapath which loads stimuli for arithmetic operations. The test could either be eyeballing or self-checking. The next task was to design and functionally test a control unit which acts as a finite state machine and was tested with a self-checking test bench. Once the control unit was properly tested, it was incorporated with the datapath to create an entire system which was then verified with a self-checking test bench. Lastly, it was all verified on the FPGA using a variety of switches and buttons to test the calculator functionality.

***Design Methodology***

*Datapath:* The datapath was designed with a mux, register file, alu, and an additional mux. The first mux received three inputs, In1, In2, and s1. The two bit input, s1, would select which input to receive choosing between In1, In2, 0, and the output of the ALU. The register file had several inputs, write enable, write address, two read enables, two read addresses, clock, and din from the mux. It then output two numbers into the ALU. The ALU would receive these numbers and depending on the value of c, it would perform either an addition, subtraction, XOR, or AND operation. Lastly, those values would go to the second mux where s2 would select between the values input from the ALU or 0. The resulting values would then be the output.

***Simulation Test Plan***

*Datapath:* The datapath was tested using a self-checking test bench. The test-bench would iterate through several values to test for all cases. It iterates through each integer value for In1 and In2 and then iterates through each operation. A register for the expected output was created in this lab and was used for verification when comparing the output to the expected result. Additionally, the value of the data in write address three of the register file was compared to the expected value. That is because the value stored in write address three of the register file contained the result of the operation between In1 and In2 or whatever was chosen by s1.

***FPGA Test Plan***

***Conclusion***

***Appendix***

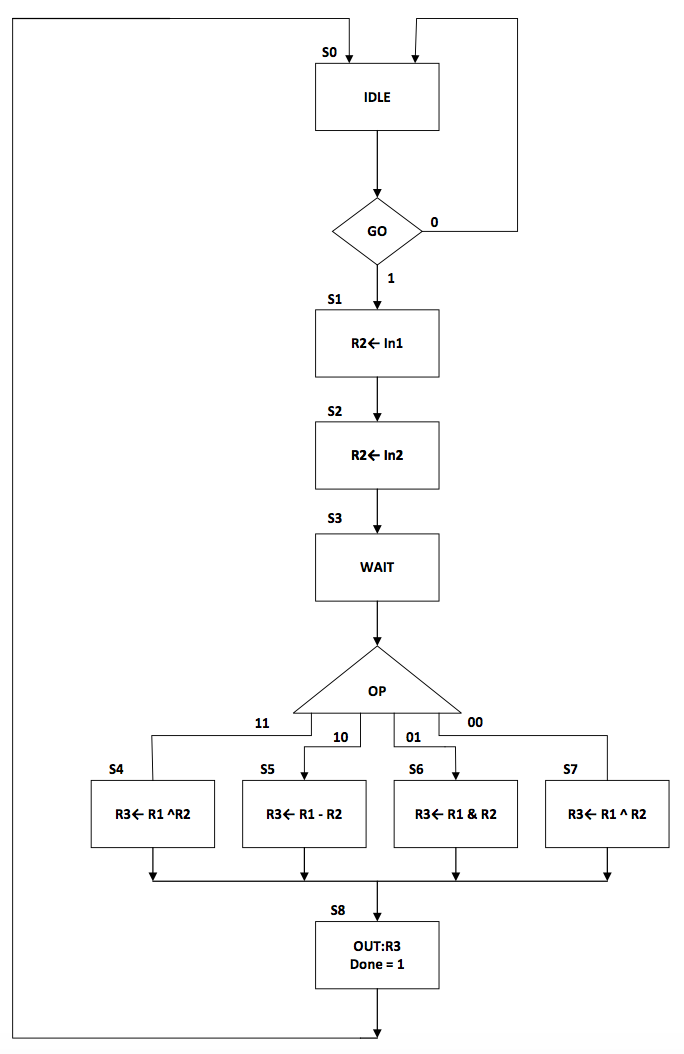
******

Figure 1: Control Unit ASM flow chart

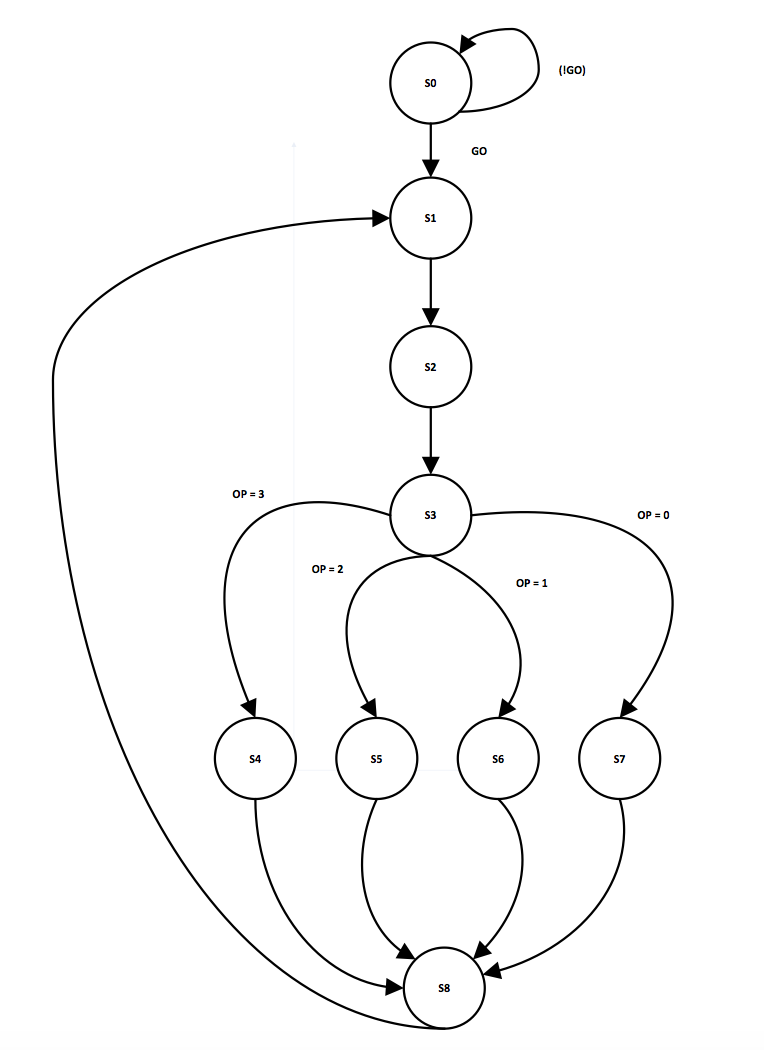


Figure 2: Control Unit FSM state transition diagram

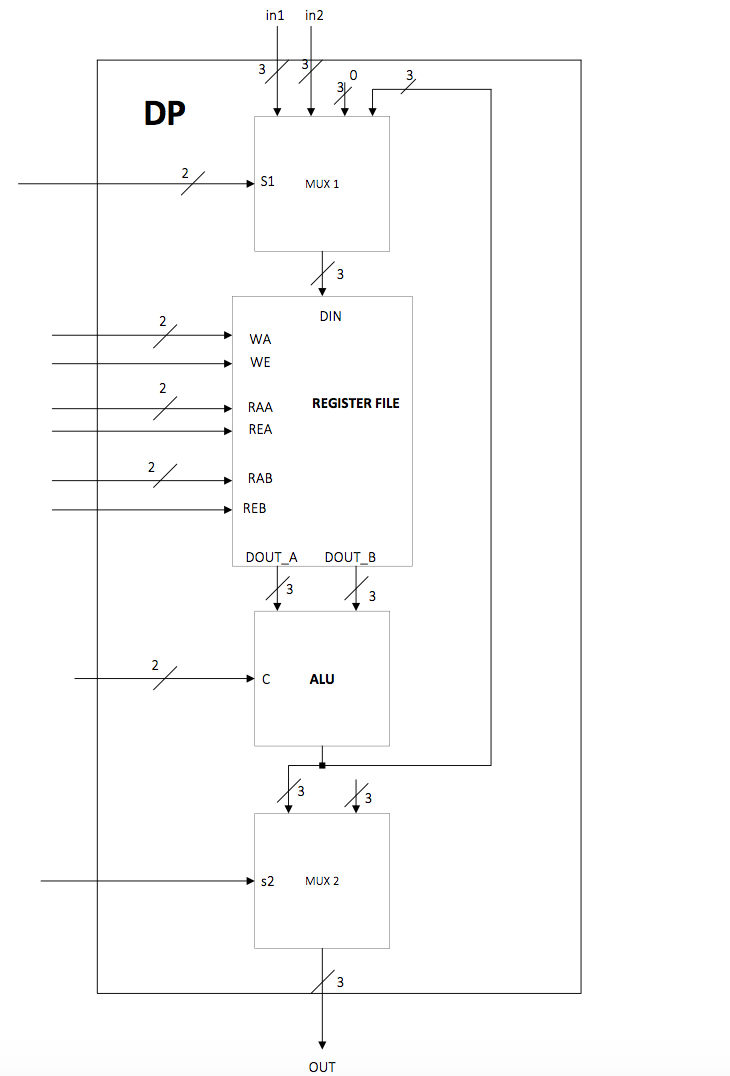


Figure 3: Datapath block diagram

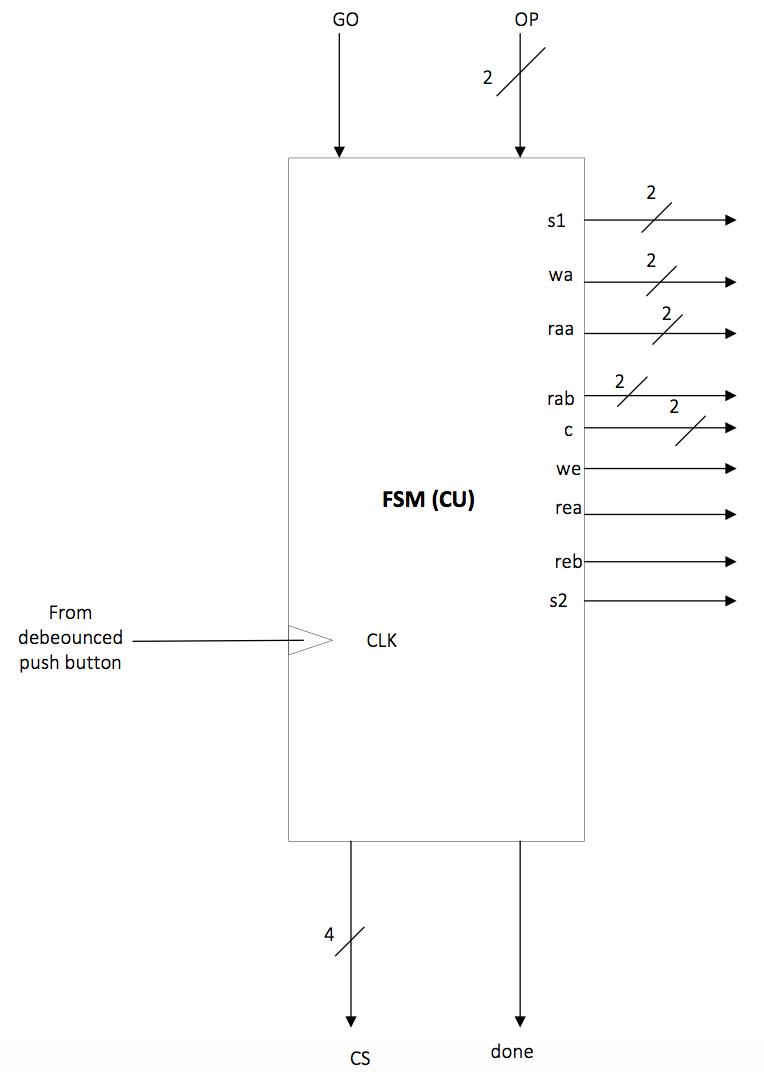


Figure 4: Control Unit block diagram

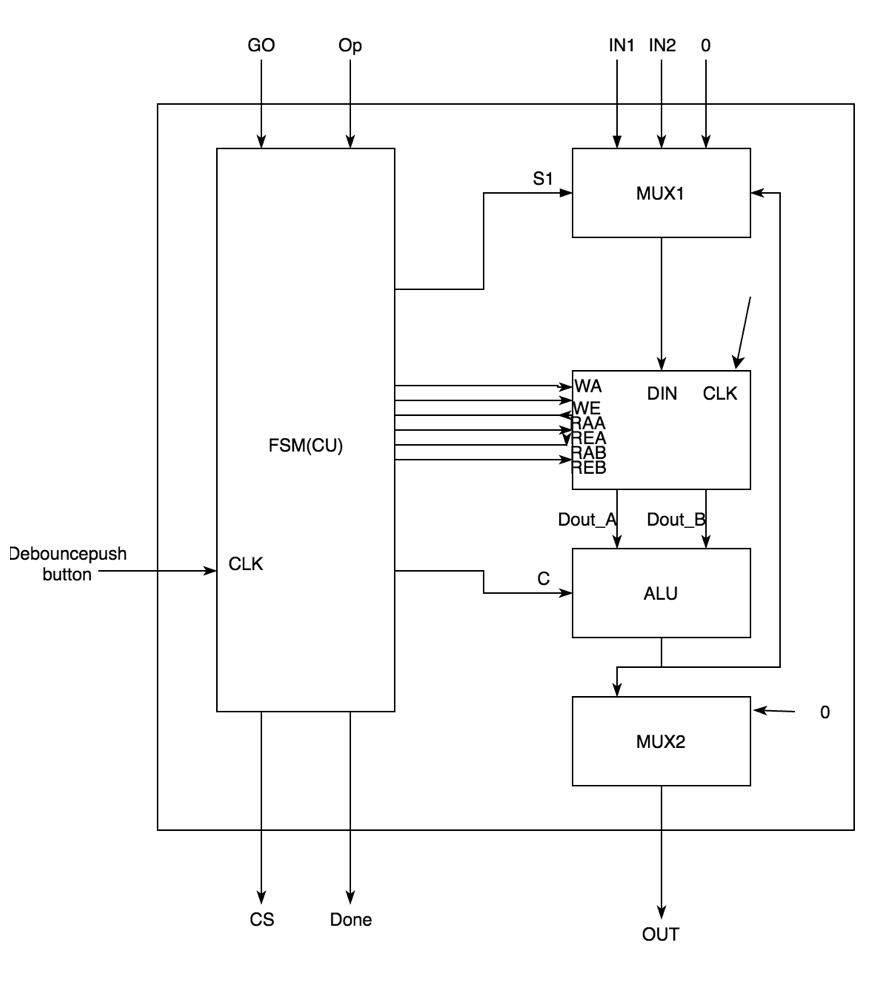


Figure 5: Top Level Design of FPGA block diagram

| Input | Outputs | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CS | S1 | WA | WE | RAA | REA | RAB | REB | C | S2 | Done |
| S0 | 01 | 00 | 0 | 0 | 0 | 00 | 0 | 00 | 0 | 0 |
| S1 | 11 | 01 | 11 | 1 | 0 | 00 | 0 | 00 | 0 | 0 |
| S2 | 10 | 10 | 0 | 1 | 0 | 00 | 0 | 00 | 0 | 0 |
| S3 | 01 | 00 | 1 | 0 | 0 | 00 | 0 | 00 | 0 | 0 |
| S4 | 00 | 11 | 1 | 1 | 1 | 10 | 1 | 00 | 0 | 0 |
| S5 | 00 | 11 | 1 | 1 | 1 | 10 | 1 | 01 | 0 | 0 |
| S6 | 00 | 11 | 1 | 1 | 1 | 10 | 1 | 10 | 0 | 0 |
| S7 | 00 | 11 | 1 | 1 | 1 | 10 | 1 | 11 | 0 | 0 |
| S8 | 01 | 00 | 0 | 0 | 1 | 11 | 1 | 10 | 1 | 1 |

Figure 6: FSM logic table for control unit

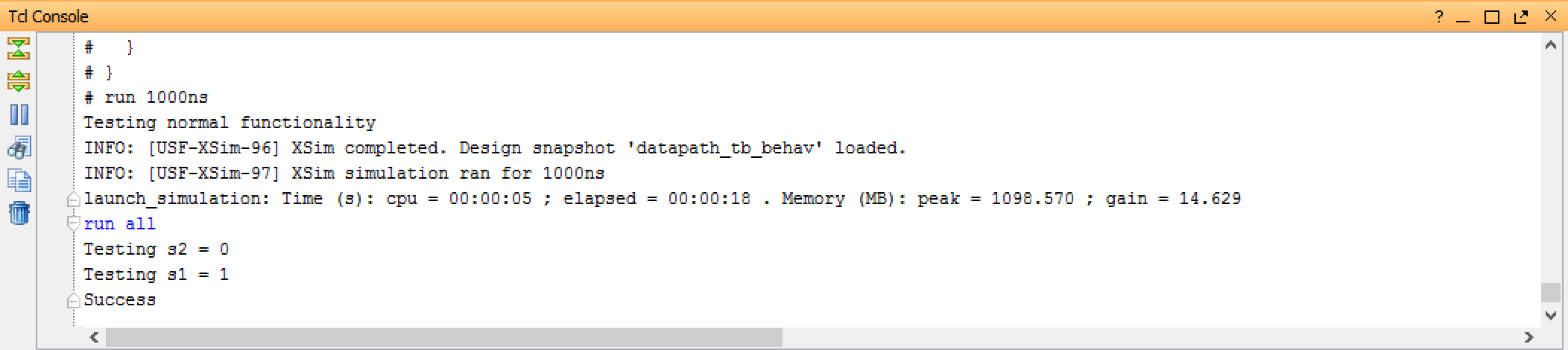


Figure 7: Console output of datapath simulation

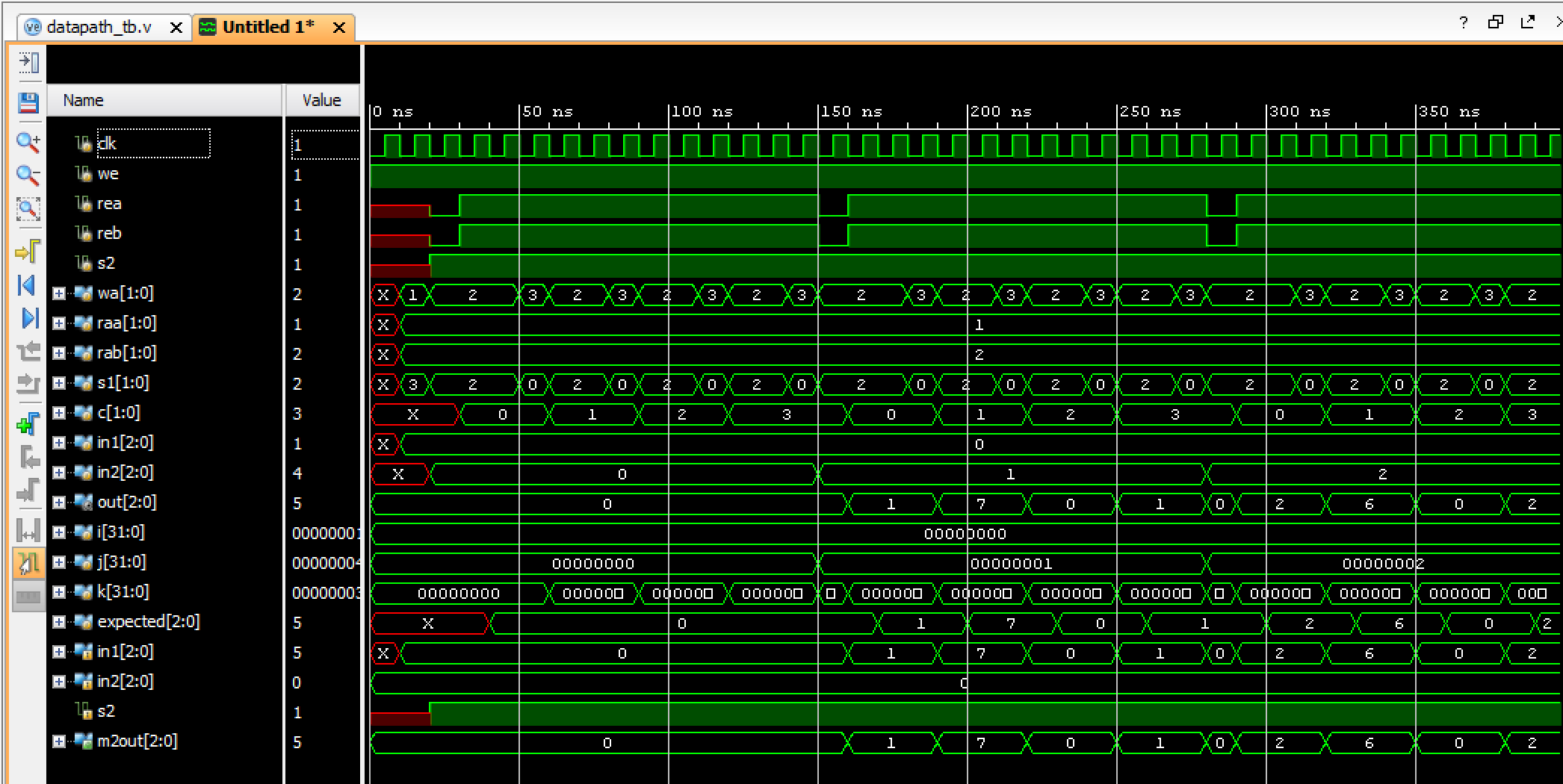


Figure 8: Waveform generated from datapath simulation with MUX2 to compare output to expected and out signals

***Source Code***