Brief Component Descriptions

1. MUX5: Allows for two 5-bit STD_LOGIC_VECTOR inputs. Depending on the select bit's value, the output 5-bit STD_LOGIC_VECTOR will be one of the inputs. AKA Takes in two possible 5-bit inputs, allows selection of one or the other inputs to output.

2. MUX64: Takes in two possible 64-bit inputs. Allows for the selection of one or the other input as an output.

3. SignExtend: Takes in a 31-bit input. The left most bit of that input is then copied across a 31-bit value, which is concatenated to the left of the original 31-bit input.

4. ShiftLeft2: Shifts a 64-bit input two bits to the left. The two right most bits are set to 0.

Brief Explainations of Modeling Types

The components MUX5, MUX64, SignExtend, and ShiftLeft2 are all modeled as `dataflow` as they all contain combinational logic. The testbenches are all modeled as `structural` as they require a component and tests it.

Waveforms from GTKWave

Waveforms are all on seperate pages! Sorry): ! Had huge technical issues and didn't have time to properly crop the empty vertical space on the screenshots.

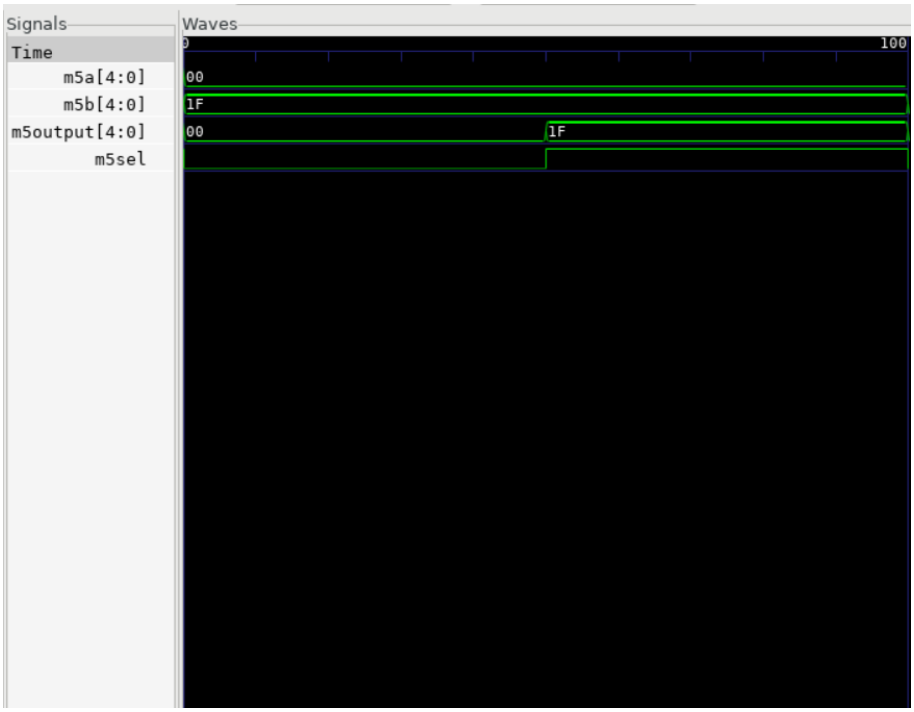

Figure 1: MUX5 Testbench Waveform - Inputs: m5a is all zeros (00) and m5b is all ones (1F). Note that when m5sel is low, m5output=m5a (00). When m5sel is high, m5output=m5b (1F).

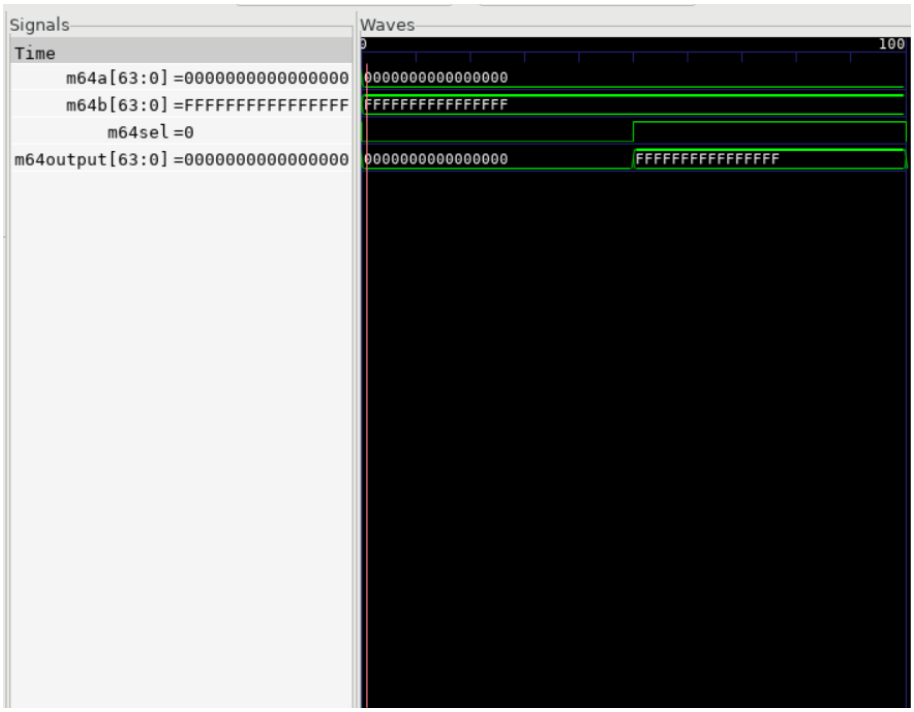

Figure 2: MUX64 Testbench Waveform - Inputs: m64a is all zeros and m5b is all ones. As soon as m64sel changes, we see that m64ouput changes which input it is equivalent to.

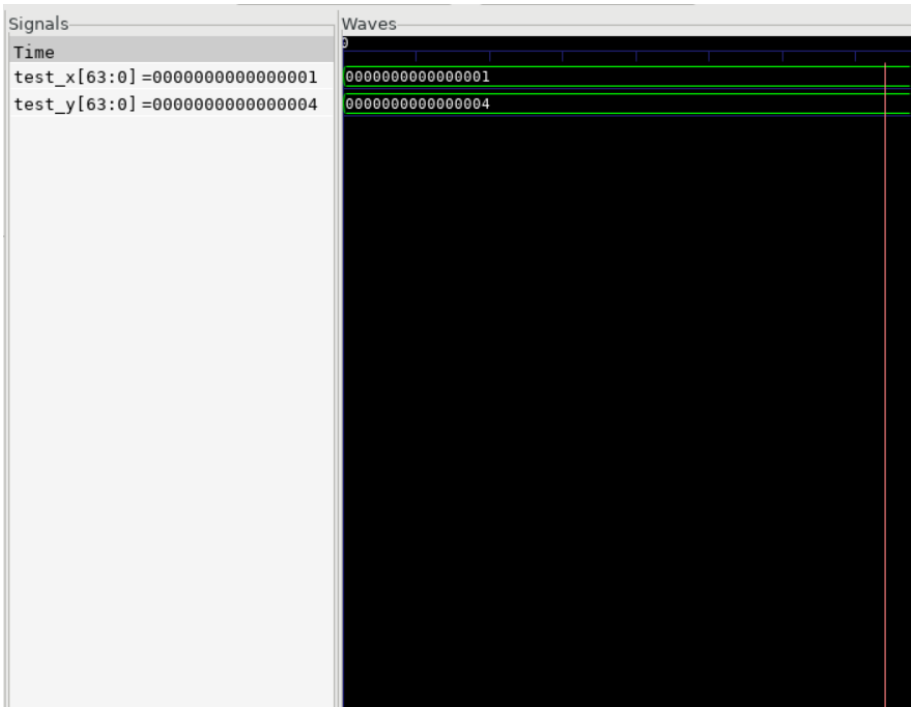

Figure 3: ShiftLeft2 Testbench Waveform - Input: test_x is a 64-bit vector with a 1 at the right-most bit. We can see that the output, test_y is hexadecimal 4, which indicates the 1 has been shifted to the left by two bits, and there are zeros in the two right-most bits.

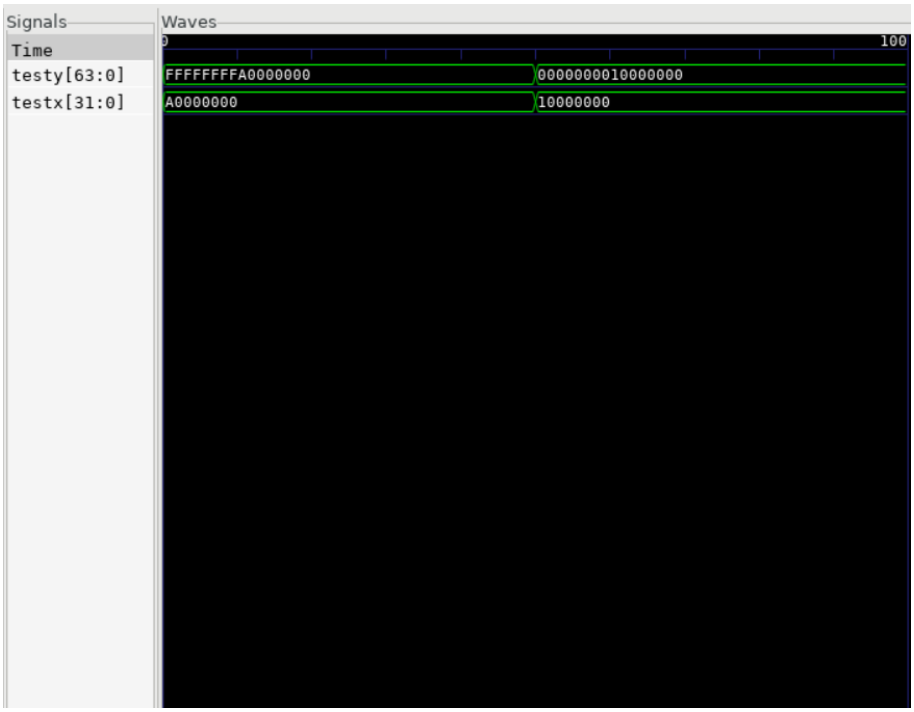

Figure 4: SignExtend Testbench Waveform - Input: testx has two different test cases, which we can see by the mux-like waveform. The first test case has testx as `10000000`. The resulting output is valid as the extended sign is all 0s.

The second test case has testx as `10000000` which can be seen on the right of the crossing waveforms. The output, testy, is then `FFFFFFFFA0000000` which is valid as it indicates the extended bit was a 1.