Billy Koech and Nicolas Weinger

CS 141: Computing Hardware

Lab 2: ALU

Testing and Simulation methodology

The ALU is tested by iterating over all the possible OP codes and for each OP code passing different values of input X and Y. To begin, the testbench tests the functionality of all the logic gates and the adder by passing the standard case inputs listed below:

1. Two positive numbers: X = 2, Y = 3.
2. A maximum 32-bit number and a zero: X = 4294967295, Y = 0.

The testbench then tests for the following special cases:

1. Two negative numbers: X = -2, Y = -3
2. A positive and a negative number: X = 3, Y= -3
3. A maximum 32-bit number and 1: X = 4294967295, Y = 1

The ALU passed the tests by fulfilling the following expectations:

* The sum of a max 32-bit number and 1 triggered the overflow. The sum of two negative number also triggered the overflow.
* The logic gates followed the expected outputs as per their respective truth tables.

SLT implementation

SRL implementation

The SRL is built by concatenating 32 zeros to the left of the input X to create a 64-bit bus. 32 MUXES are then connected to the 64 wires as per the sketch below. This allows for the previously occupied bits to be set to zero when shifting occurs.

The output of the SRL is then connected to one input of a 2:1 mux. This mux connects the SRL to the output Z only if the shamt Y is less than or equal to 11111 (32base10). Otherwise it connects the mux to ground (zeros) thus when the shamt is greater than 11111 then the output is 32’b0.

The SRL is tested by checking whether shifting the Xs specified above shift by their corresponding Ys.

SLL implementation

The SLL is similar to the SRL except that it concatenates 32 zeros to the right of the input X to create a 64-bit bus. 32 MUXES are then connected to the 64 wires as per the sketch below. This allows for the previously occupied bits to be set to zero when shifting occurs.

SRA implementation