

Alex Kahng, George Zhang
CS141 – Lab 3 Part 1
Design and Simulation Description
Due: Oct. 7th 2016

Design

We had to design two aspects: a register and a tape. For the register, we allowed it to take in a parameter called SIZE, which defaults to 2. The input wire and output reg has SIZE number of bits. The default is 2 because we want to represent a 1, 0, and a blank. We coded 1 as 01, 0 as 00, and a blank as something with a leading 1 (so either 10 or 11). We allowed for a reset, which defaults to setting the value to a blank (default to -1), and we allowed for an enable, which sets the output to whatever the input to the register is. All of these changes occur on the positive edge of the clock that we pass in.

For the tape, we had a two-bit input, which will represent whether it should be a 1, 0, or blank (with the same encoding as in the register). The output is also two bits, as it also needs to show as a 1, 0, or blank. We create an in_exp reg and an out_exp wire, which stores the expanded version of the 8-bit number, so both are 16 bits. We also store 8-bit wires for enable bits and reset bits. We hook all of these bits up to each of the registers, which allows us to store each bit of our 8-bit number. On the positive clock edge, we check whether the mode inputted is a read (0) or a write (1). If it's a write, we set enable to 1 and reset to 0 and set the appropriate bits of in_exp to be equal to in. However, if it's a write and the reset bit is on, we set enable to 0 and reset to 1 in order to reset the register. On a read, we set all bits of enable to 0 and set out equal to the appropriate bits of out_exp.

Simulation

In order to simulate our register, we first created an alternating clock. We then set enable to 1, and tested all possible values of in to make sure that out is appropriately set. We then set enable to 0 and made sure that changing in does not change out. Finally, we turn the reset bit on and make sure out is set to blank (-1).

In order to simulate our tape, we move along each bit of the tape. For each bit, we first set the bit to be 1 if the index is even and 0 if odd. Since a write moves the head, we then move the head back and follow that up with a read. In this way, after three periods of the clock, we should read out what we wrote in. We do this for every bit to ensure that we are writing appropriately. We follow that up with 150 random tests of mode, move, reset, and in, and we check correctness by looking at the waveforms.