COMP 408: Verifying Simple Hardware Circuits in SMV

These exercises are intended to give you some practice using SMV. You should turn in your solutions in accordance with the instructions for documenting assignments in SMV on the course assignments page.

Exercise 1 (**Arithmetic Shifter**) An *arithmetic shifter* stores a bit-vector of some specified length. It has two control signals: shift-left and shift-right. When shift-left is asserted, the bits in the vector are shifted one bit to the left, with a 0 shifted into the rightmost bit. When shift-right is asserted, the bits in the vector are shifted one bit to the right, with a 0 shifted into the leftmost bit. For example:

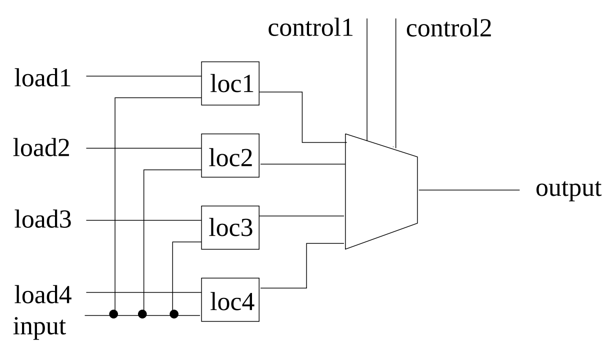
Contents of vector	Operation	New contents of vector
1 1 0 1	shift-right	0 1 1 0
1 1 0 1	shift-left	1 0 1 0

1. Write an SMV specification of a 4-bit arithmetic shifter. You should use a style similar to that used for the counter in the SMV manual (*i.e.*, make a cell module, where each cell corresponds to a single bit of the shifter).
2. Determine the properties you need to test in order to verify the shifter. Argue why the set of properties you propose is sufficient.
3. Use SMV to verify the properties given in response to part 2.

Exercise 2 (**Simple Memory**) Suppose we have a simple memory with a single input bus and four storage locations. Each storage location has its own load signal; when the load signal is asserted, the value currently on the input bus is stored in that memory location. There is a single output bus for the memory. The value on the bus is controlled (via a multiplexer, if you're familiar with that term) by a pair of output control signals. The values on these signals indicates which location's contents should be placed on the bus, according to the following table:

Control1	Control2	Location
0	0	1
1	0	2
0	1	3
1	1	4

The layout of this memory appears as follows:



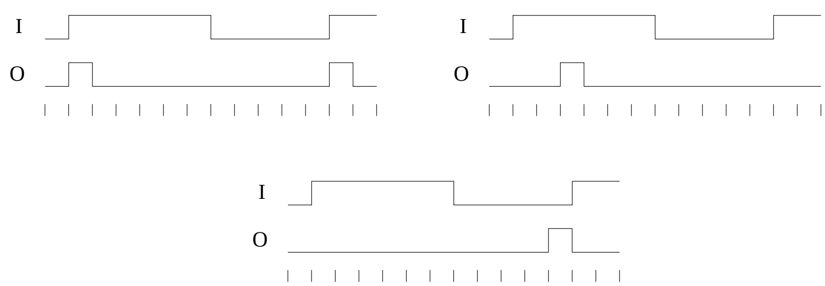
The desired properties of the memory are:

- Values are properly stored in the memory based on the load signals.
- Values are routed from the locations to the output bus as specified by the above table.

- Once a value has been loaded into a location, the location retains the value until the next load to that location occurs.

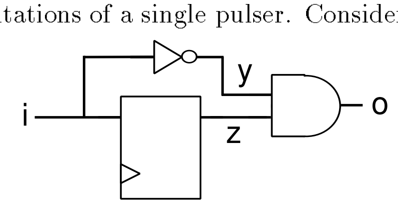
Copy the file *simplemem.smv* from `comp408/Assignments/1` on owlnet and use SMV to verify the above properties against that design.

Exercise 3 (**The Single Pulser**) A *single pulser* is a simple hardware device. It has a single input I and a single output O. The pulser produces an output pulse lasting one unit of time on O for each pulse of arbitrary but finite duration that appears on I. For purposes of this problem, we will restrict the behavior so that the output pulse corresponding to an input pulse must occur before the next input pulse starts. For example, the following three examples meet these requirements



but the following two do not: the first because the output pulse is too long and the second because there is no output pulse occuring before the second rising transition on I.

There are several possible implementations of a single pulser. Consider the following one:



1. Draw the state machine corresponding to this implementation.
2. Write the SMV specification from the state machine.
3. Determine the properties you need to test in order to verify the single pulser. Argue why the set of properties you propose is sufficient.
4. Use SMV to verify the properties given in response to part 3.