Useful ModelSim Commands

There are many ModelSim commands that you can add to your .do file to control the simulation of your circuit. This handout discusses a few commands that can help you get started with ModelSim.

vlib *work*

Creates a design library called work into which your Verilog code will be compiled.

vlog *-novopt <file_name>.v* *(**vlog** -novopt mux.v)*

Compiles <file name>.v without optimizations. Turning off optimization allows you to log the internal signals according to how your code is structured. When optimization is used, the circuit may be restructured and internal nodes may disappear making debugging more difficult. If you have multiple .v files, you should compile them all. For hierarchical designs, you should compile the lower level design blocks before the higher level.

vsim *<module_name>* *(**vsim** mux)*

Starts the simulator using <module name> as the top level for simulation.

log *–r {/*}*

Logs all objects in the design. This way you can observe internal signals at any hierarchical level which is very helpful during debugging. This command only logs all objects, to actually view an internal signal you can add it to the waveform using the GUI.

add wave *{/*}*

Adds all objects of the top level simulation module to the waveform.

force *<object_name> <value>* *(**force** {SW[0]} 0)*

Sets input <object_name> (SW[0]) to <value> (0). ***Force*** command can be used with many different options and we will take a closer look at some examples, later.

run *<time_steps> [<time_units>]* *(**run** 10ns)*

Advances the simulation by the specified number of timesteps (10ns).

Periodic signals with ***force*** command:-

Sometimes you might want to stimulate your circuit using a periodic signal. An example of that is the clock signal. For sequential logic, your circuit will take a clock input, which is a periodic signal with a certain frequency. A useful option of the ***force*** command is *–r*, the syntax is as follow.

force *<object_name> <value_1> <time_1> , <value_2> <time_2> …. –r <time_3>*

This forces <object_name> to <value_1> at <time_1> time units after the current simulation time and forces it to <value_2> at <time_2> time units after the current simulation time. This cycle repeats after <time_3> time units after the current simulation time. An example of a clock with 100MHz would like this :

force *{clk} 0 0ns , 1 {5ns} -r 10ns*

run *300ns*

The first commands sets clk to after 0ns, then sets it to 1 after 5ns. This cycle repeats after 10ns. Fig. 1 shows the generated waveform from these commands.

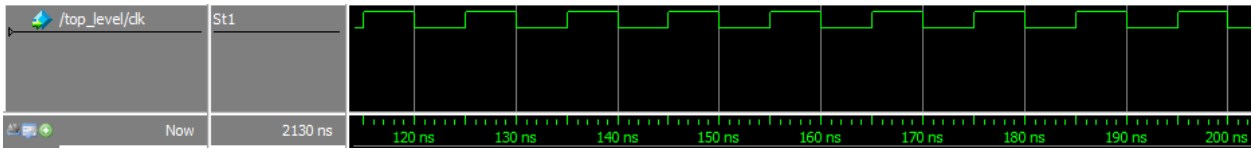

Fig.1 clk signal with 100MHz frequency.

Another use of the –r option would be to easily exhaustively simulate a small design. The code (mux.v) of the 2-to-1 mux we provided for lab 2 has 3 inputs (SW[0], SW[1] and SW[9]) and 1 output (LEDR[0]). We can use the –r option to simulate all possible input combinations with just 3 lines, as follows:-

force *{SW[9]} 0 0ns, 1 {40ns} -r 80ns*

force *{SW[1]} 0 0ns, 1 {20ns} -r 40ns*

force *{SW[0]} 0 0ns, 1 {10ns} -r 20ns*

run *100ns*

Fig. 2 shows the resultant waveform with all possible input combinations and the corresponding output.

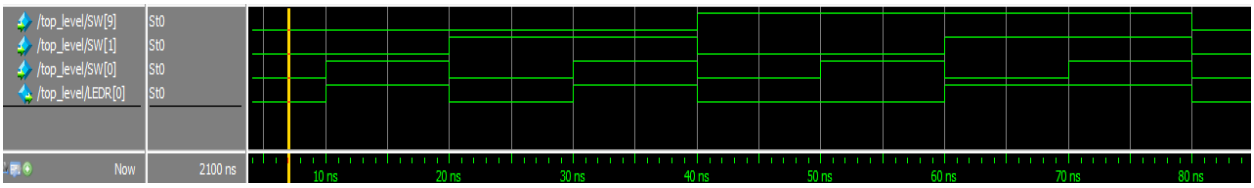

Fig. 2 Inputs & outputs of a 2-to-1 mux.