Counter short tutorial on how to compile with Icarus verilog, how to simulate and export waves viewable by gtkwave.

1) Command for compiling the test bench and DUT (The DUT is included by the testbench by using `include directive):

iverilog -o counter_compiled.a counter_tb.v

Icarus verilog help output:
Usage: iverilog [-ESvV] [-B base] [-c cmdfile] [-g1|-g2|-g3.0]
[-D macro[=defn]] [-I includedir] [-M depfile] [-m module]
[-N file] [-o filename] [-p flag=value]
[-s topmodule] [-t target] [-T min|typ|max]
[-W class] [-y dir] [-Y suf] source_file(s)

2) Command for running simulation and exporting waves (for all the items available in design, see $dumpvars (1, first_counter_tb)):

vvp counter_compiled.out

Usage: vvp [options] input-file [+plusargs...]
Options:
-h Print this help message.
-l file Logfile, '-' for <stderr>
-M path VPI module directory
-M - Clear VPI module path
-m module Load vpi module.
-v Verbose progress messages.

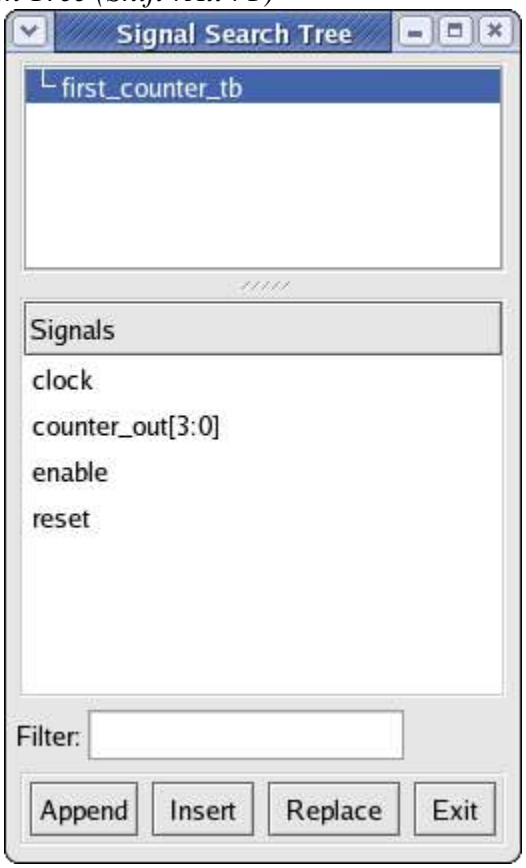
A dump file (counter.dump) will be generated

3) Loading waves into gtkwave:

gtkwave counter.dump

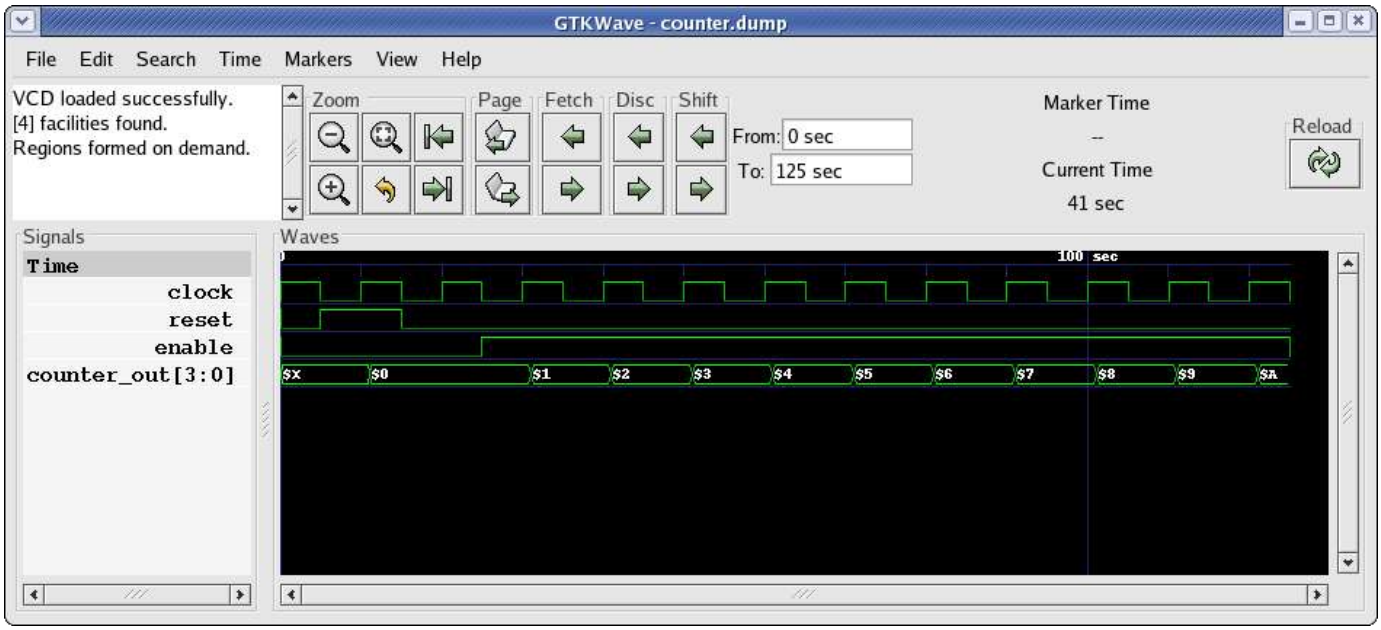
4) Adding signals:

4a) Go to Search/Signal Search Tree (Shift+Alt+T)



4b) Click on the signals you want to display and push Append/Insert (try both)

4c) After finishing, push exit to close the hierarchy window. The result should look like below:



You're now done. This is only a small example, just to let you know about the main flow. You should further play with the tools available in gtkwave in order to use it efficiently.

ENJOY!