Verilog testbenches

Proposed problems

Oprițoiu Flavius
flavius.opritoiu@cs.upt.ro

October 16, 2024

Problem 1

Construct a Verilog module for a Full Adder Cell (FAC), and name it *fac*. The module has 3 1-bit inputs *x*, *y* and *ci*, and 2 1-bit outputs *z* and *co*. Write your code in a file called *fac.v*.

Problem 2

Construct a testbench that exhaustively verify the *fac* module implemented previously, and call it *fac_tb*. The content of the *fac_tb* module can be included in the same *fac.v* file, alongside the *fac* unit.

Build a script file, called *run_fac.txt*, for compiling, launching the simulation and running the simulation of the *fac_tb* module.

Problem 3

Using 2 *fac* instances, construct a 2-bit adder called *add2b*, having 2, 2-bit inputs *x* and *y* and the 1-bit input *ci*, together with the 2-bit output *o* and the 1-bit output *co*. Write your code in file *add2b.v*.

Construct a testbench module for verifying your implementation, called *add2b_tb* and the associated script file *run_add2b.txt*

Problem 4

Design a module *cmp2b* for comparison of 2-bit unsigned numbers, having 2, 2-bit inputs, x and y, for the 2 numbers to compare and 3 1-bit outputs: *eq*(equal), *lt*(less than) and *gt*(greater than). Write your code in file *cmp2b.v*.

Problem 5

Using module *cmp2b*, build a 4-bit comparator, called *cmp4b*, having 2, 4-bit unsigned inputs x and y, for the 2 operands and 3 1-bit outputs: *eq*, *lt* and *gt*. Write your code in file *cmp4b.v*.

Construct a testbench module for exhaustive verification of your implementation, called *cmp4b_tb* and the associated script file *run_cmp4b.txt*.

Problem 6

Build a module for addition of two integers represented in C1, called *c1_add4b* which has 2, 4-bit inputs x, y and a 1-bit input *ci* generating at its output signals z on 4 bits (no carry out because of the *end around carry*).

Construct a testbench module for exhaustive verification of your implementation, called *c1add4b_tb* and the associated script file *run_c1add4b.txt*.