

ECE 368—Spring 2009, Instructor: Prof. Shantanu Dutt

Lab 7 : Due Thurs. April 30

Purpose

The purposes of this lab are:

1. Use of the Synopsys Design Compiler to synthesize VHDL circuit descriptions to actual circuits.
2. Making you aware of the important fact that the design developed by the design engineer is the most important cog in the final circuit-realization process even in a CAD based design environment—CAD tools help ease the design description, simulation and final synthesis processes, but do not generate the initial design, whose quality (e.g., speed, hardware cost) is dependent on the designer's abilities (skill, knowledge and ingenuity).
3. Reading input data from a file.

Design Problem

Assume that the delay of a k -input gate of any type is $2k$ ns. Also assume the following two different 2:1 Multiplexor (Mux) delays: (i) 10 ns using a gate-based implementation, and (ii) 2 ns using a transmission gate (T-gate) implementation.

The total gate-input cost (TIC) of a circuit is defined as simply the total number of inputs across all gates in the circuit. For a 2:1 Mux use the following TIC values for its two different implementations: (i) TIC = 7 for a gate-based implementation, and (ii) TIC = 2 for a T-gate implementation.

- (1) (a) Code in structural VHDL description a 16-bit comparator the tree-structured design discussed in class and given in the Lecture Notes titled "Introductory Tour—Design Approaches and VHDL Description" (that discusses the "Divide-and-Conquer" design strategy). The comparator has two 16-bit inputs A and B that represent unsigned numbers, and a 1-bit output f defined as $f = 1$ if $A > B$ and $f = 0$ otherwise (if $A \leq B$). The structural description should use `for generate` and `if generate` statements and can be described iteratively (preferred) or recursively. **300**

Notes: (1) It seems from our initial explorations that the Synopsys Design Compiler (DC) does not support two-dimensional arrays; thus if describing your design iteratively, you need to define 5 arrays of `std_logic` signals, one for each level, of the appropriate sizes (16, 8, 4, 2, 1 bit vectors) and depending on the index j of the outer `for generate` loop use the proper array for the connecting input

and output signals using `if generate` statements.

(2) DC also does not seem to support the `time` type, so do not use generic time parameters in your components; instead hard-code the delays with appropriate `after` statements based on the above delay specs of the basic components; use any of the above given delays for the Mux component (it does not affect the final circuit synthesized, whose delay you need to determine manually, anyway).

(b) Generate a circuit of the above VHDL description using DC, and determine its critical path delay and its TIC for the two 2:1 Mux implementations mentioned above (so you will get two different critical path delays and TIC's). Show your work clearly. **200**

(2) (a) Give a *behavioral* VHDL description of the same 16-bit comparator using the MSB-to-LSB scanning approach discussed in class (and also given in the above-mentioned lecture notes). **300**

(b) Generate a circuit of the above VHDL description using DC, and determine its critical path delay and its TIC. Show your work clearly. **200**

(3) Compare the delays of the circuit synthesized from the VHDL structural description of the tree-structured design (two delays based on two different Mux implementations) with the delay of the circuit synthesized from the behavioral description, and give the rationale for the findings.

Similarly, compare the two TICs of the circuit synthesized from the VHDL structural description of the tree-structured design with the TIC of the circuit synthesized from the behavioral description, and give the rationale for the findings. **300**

Extra Credit Problem: Simulation Test Bench

Instantiate the two 16-bit comparators you have described (structural and behavioral) in a test bench that does the following **300**

1. A process that reads 20 test inputs (each test input is a pair of 16-bit vectors) from a file and inputs to them, one at a time, to each comparator.
2. Determine from the graphical interface, the output values of the two comparators and verify and show that they are equal for each input pair and correct.
3. Also determine from the graphical interface, the *worst-case* simulation delay, over these 20 inputs, of the outputs of the two comparators, and compare them. Also compare the *average* simulation delay of the two comparators over these 20 inputs.