

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

Lab 4 : Part I due Thurs. Feb. 26, Part II due Thurs. March 5

Purpose

The purpose of this lab is to: 1) Develop a fast design for an even-parity detector using full adders (FAs) and determine its delay analytically and empirically; 2) Describe tree-structured circuits using nested **generate** statements; 3) Use of multiple processes (used in the TB).

Design Problem

- (1) Design a fast n -bit even-parity detector—output = 1 if # of 1's in the n input bits is even, otherwise output = 0—where $n = 3^q$, using only the FAs that you have designed in a previous lab and **any one** of the following type of 2- or 1-input gates: AND/OR/NAND/NOR/NOT. Note that the parity detector discussed in class was an odd-parity detector. Clearly show all your work in obtaining the final design, and give the schematic of your design for a 27-bit parity detector. **100**
Hint: Determine if the basic operation/function on 3 bits is associative or not. In the latter case, use the D&C approach; the stitch-up function will be the basic operation on 2 bits. In the former case, also use D&C. However, you will need to determine what the stitch-up function is. Note that D&C implies that you break up a problem into ≥ 2 subproblems (not necessarily, into only 2 subproblems), and the # of subproblems you break up a problem into is partly dictated by the basic logic component(s) you have to design the circuit.
- (2) Analytically determine the delay of your n -bit even-parity detector assuming that a k -input XOR/XNOR gate is $3k$ ns. This is the *analytical* delay. Show your analysis and derivation clearly. **50**
- (3) Analytically determine the number of FAs and the other gate, if any, that you use in your design for an n -bit even-parity detector. Show your analysis and derivation clearly. **50**
- (4) Give a VHDL structural description of your n -bit even-parity detector using nested **generate** statements. Use a generic statement with size parameter q (where $n = 3^q$) and appropriate delay parameter(s), that correspond to the above-specified delay “formula” for XOR/XNOR gates, to pass down to the FA component which passes it (them) down to the XOR-gate component it uses. **200**

Test Bench

Part I: Demonstrate the working of the 27-bit even-parity detector using a test bench in which 41 27-bit inputs with numerical values $5i$, $i = 0, 1, \dots, 41$, are fed to the circuit. You will need to create this test bench yourself. **100**

Part II:

- (1) **Test Bench:** Instantiate a 3-bit, a 9-bit, a 27-bit, a 81-bit and a 243-bit even-parity detector in a test bench that provides 41 random test inputs to the above even-parity detectors (inputs will repeat in the case of the 3-bit circuit), respectively. Use five different processes to generate the different inputs to the five even-parity detectors. Furthermore, synchronize the input assertion across the five processes by having a **wait for x ns** statement in each process after the input assertion where $x = 1.2$ (analytical delay of the 243-bit even-parity detector). You will need to create this test bench yourself.

Hint: For generating random `std_logic` vectors with $t > 31$ bits (31 bits being the size of a positive integer and thus the size returned by the procedure `uniform_d()` that you are using for generating random integers, and from there the corresponding random `std_logic` vectors), break it up into $\lceil t/31 \rceil$ parts, get the random `std_logic` vector for each part, and then concatenate them using the ‘&’ operator.

Across these test inputs do the following in the test bench:

Determine from the graphical interface, the worst-case *simulation* delay, for each of the five even-parity detectors. Plot this delay versus size (n) on a linear and log scale graph (you can use Excel (on PCs only) or gnuplot (latter is available on Unix/Linux machines as well as PCs) to do the plots. What kind of growth function do you see? Discuss.

150

- (2) Compare the analytical and simulation delays for the 27-bit even-parity detectors you have obtained for your CLA and comment on any discrepancy giving possible reasons for it. **50**