LAB 2

Timing and Delay

In this lab you will learn how to analyze the timing of logic circuits using unit-delay and timing simulation.

1.0 Overview

Digital logic circuits are constructed from electronic switches (transistors) connected by wires. The digital abstraction allows us to ignore the intrinsically analog nature of these circuits; it consists of two related simplifications of physical reality: a *functional* abstraction that allows us to transform continuous voltage waveforms into discrete two-valued logic waveforms, and a *temporal* abstraction that captures the causal relations among logic signals using appropriately-defined propagation delays. In this experiment, you will explore some of the timing aspects of logic circuits.

2.0 Preparation

- Read Sec. 3.6 and 5.2 of Wakerly on timing and delay.
- Read Sec. 5.10.1 and 5.10.2 of Wakerly to understand the timing behavior of ripple adders (we'll discuss these circuits in more detail later.)
- Consult the Xilinx Foundations on-line documentation and the Xilinx tutorial (follow link on class web page) on the use of Timing Simulation and on how to create "macros" and hierarchical schematics.

3.0 Design Specification

You will not be designing any logic circuits in this experiment. Rather, you will be studying the temporal characteristics of the three circuits shown in Figure 3. The circuit

FIGURE 3. Circuits to demonstrate propagation delay

(a)

(b)

(c)

in part (a) has no known useful application other than to illustrate that functionally identical signals may not necessarily have the same temporal behavior. The circuit in part (b) is called a ring oscillator; note that it has a feedback loop and hence is not combinational. The circuit in part (c) is a 4-bit ripple-carry adder; each macro is a full adder circuit similar to the one on the first homework assignment.

For each of these three circuits you are asked to demonstrate its timing behavior a) under a unit-delay model, b) using estimated delays after the circuit has been implemented, and c) with actual measurements on the logic board using an oscilloscope.

4.0 Notes

- This is basically a hands-on lab with a lot of busy work to collect data. You should create schematics for each of the three circuits, and simulate them in both the unit delay and timing modes. Note that you must check the **Produce Timing Simulation Data** box in the **Options** window of the implementation tool. This presumably causes estimated delays (due to the physical placement of the gates and routing of the wires inside the FPGA chip) to be back annotated on the circuit's netlist for use by the simulator. Make sure you add the **KEEP** attribute to all nets; otherwise, they may get optimized away (removed) by the software to produce a fast implementation.
- Learn how to use the measurement tool in the simulator to compute propagation delays between signal events.
- Learn how to find out what the delays of your gates are: they can be found in the simulator's **Device->Edit Timing Specification** menu. You will make things much easier if you label all of your gates and nets in the schematic; otherwise, you may be confronted with unfamiliar program-generated gate and net names.
- The gate delays generated by the implementation tool and used by the timing simulator have the form *InputDelay + OutputDelay* where InputDelay is a specific-input-to-gate-output propagation delay and OutputDelay is a common-to-all-inputs intrinsic delay. Both types of delay can have a low-to-high and a high-to-low value: for InputDelay, they are denoted as TpLH and TpHL; for OutputDelay, they are denoted as OUTDLYLH and OUTDLYHL.

5.0 Deliverables

5.1 Pre-Lab

1. Hardcopy of schematic and UCF files for all three circuits.
2. Hardcopy of unit-delay simulation results for all three circuits. For circuit (a), produce a simulation trace showing a sufficiently wide positive pulse applied to D0 and the resulting waveforms on outputs S1 to S4. For the ring oscillator, apply a high input to initialize the circuit, followed by a transition to 0 to initiate the oscillation. Show the waveforms on all nets, and "measure" the period. For the adder circuit, apply the input pattern on A, B, and CIN that will cause signal propagation along the entire carry chain (from CIN to COUT); show waveforms on all carry nets, in addition to the input and sum signals.
3. Hardcopy of the timing simulation results for the same scenarios in 2.

5.2 In-Lab

There are ten oscilloscopes. Your Lab GSI will demonstrate their use and help you measure the propagation delays for your circuits. We may decide, depending on how busy the labs are, to measure the delays for just one of the circuits. You should sketch the waveforms you see on the oscilloscope and record the delay measurements. Your Lab GSI will sign your experiment's cover sheet after this has been completed.

5.3 Post-Lab

Prepare your lab report as described in the *EECS270 Laboratory Overview* handout. Make sure you complete and include all parts of the report including the *Cover Sheet*, the *Design Narrative* section, and the *Design Documentation* section. Include as part of your design documentation all corrected pre-lab requirements. In your design narrative, point out any anomalies you encountered, and try to explain them as best you can. In the *Post-Lab Questions* section answer the following questions:

1. Assuming that the delay of each full adder macro is one unit, write an equation for the maximum propagation delay of an *n*-bit ripple-carry adder.
2. Re-derive the maximum propagation delay for your ripple-carry adder assuming that the delay of each gate is one time unit.
3. What makes the operation of the ripple-carry adder so slow? What would be a good method of enhancing its performance? Just give an English explanation; you don't need to design a new circuit.
4. Assuming unit delays, write an equation for the period of oscillation of an *n*-stage ring oscillator. What is the period when $n = 20$?