Laboratory 1

Objective

The objective of this laboratory is to provide you with experience in specifying behavioral and structural VHDL descriptions of familiar components.

Problem 1

Develop a behavioral VHDL model for a 4-to-1 word multiplexer (MUX). Your model should work with arbitrary length words; i.e., you should not place hard constraints on the lengths of inputs and outputs. Develop a test bench for your MUX that demonstrates each function.

Problem 2

Develop a behavioral VHDL model for a 1-bit, 2-to-4 demultiplexer. Your device should include an ENABLE signal as well as normal inputs and outputs. Develop a test bench for your VHDL demultiplexer model that demonstrates basic functionality. Simulate your design to demonstrate correctness.

Problem 3

Develop a behavioral VHDL model for a 4-bit shift register. Your shift register should implement functions for LOAD, HOLD, RIGHT SHIFT and LEFT SHIFT. In addition to regular inputs, your shift register should provide a SHIFT LEFT INPUT and a SHIFT RIGHT INPUT that input the value shifted into the right-most and left-most bits, respectively. Your register should also include an ENABLE input and a CLOCK input. Design your register to be a positive level-triggered device. Develop a test bench for your VHDL shift register that demonstrates each function.

Problem 4

Use your 4-bit shift register from Problem 3 to implement a structural VHDL model for an 8-bit shift register. This device should perform the same functions as the 4-bit shift register, but over 8 bits. Develop a test bench for your shift register that demonstrates each function.

Problem 5

Modify your design from Problem 3 to implement a rising edge triggered device. If you designed your system carefully, this should involve changing exactly one line of VHDL. Modify your test bench from Problem 3 to demonstrate the distinction between the two shift registers.

Problem 6

Use your shift register model from Problem 5 and your MUX model from Problem 1 to develop a 4-bit Grey-code counter. Your counter should implement LOAD, HOLD, COUNT UP and COUNT DOWN functions. Your counter should also include an ENABLE input and a CLOCK input. Design your counter to be a rising edge-triggered device.

Optional Homework

The objective of this homework assignment is to perform an analysis of three adder circuits. This analysis will help you understand the complexity versus speed trade-off of adders. The three adder types are:

- Ripple carry adder
- Full carry lookahead adder
- Hierarchical carry lookahead

You should adhere to the following constraints:

- Four inputs maximum per gate
- All gate delays are 2 nanoseconds
- Complex gates (such as XORs, MUXs) must be expressed in terms of AND and OR gates for computing delays and gate counts.

Do the following for each adder type and compare the results.

1. Develop expressions in terms of *n*, the number of bits in the words to be added, for the total number of gates required for implementing the adder circuit and for the time required to perform the addition. You should have the following 6 equations:

$t_{\text{gate\_delay}}\text{ripple}$	$= f(n)$	$\#\text{gates}_{\text{ripple}}$	$= f(n)$
$t_{\text{gate\_delay}}\text{full\_carry}$	$= f(n)$	$\#\text{gates}_{\text{full\_carry}}$	$= f(n)$
$t_{\text{gate\_delay}}\text{hierarchical\_carry}$	$= f(n)$	$\#\text{gates}_{\text{hierarchical\_carry}}$	$= f(n)$

2. Generate the three following plots for $n = 2, 4, 8, 26$ and 32
   - Plot 1. Total #gates
   - Plot 2. Total time
   - Plot 3. Gate × time complexity
3. Which design would you implement for a 16-bit adder based on the *gate* × *time* complexity calculations?