CSCE 612 – Digital Systems Design Using VHDL
Homework #7

4- bit Binary Up/Down Counter

Functional Description: The block diagram along with the pin out description is given below. The counter has a RESETn (active low) input, which is used to reset the counter. It uses a single-phase clock CLKn (active low). The LDn (active low) signal is used to load the 4-bit data input DATA into the counter. The data input is the base value for counting. When the LDn is high, the counter starts counting. The CTRL input defines the direction (up/down) of counting. The counter has two outputs COUT, which is the count value output while counting, and CO, the overflow status flag. The overflow is set when the counter value reaches its limit when counting.

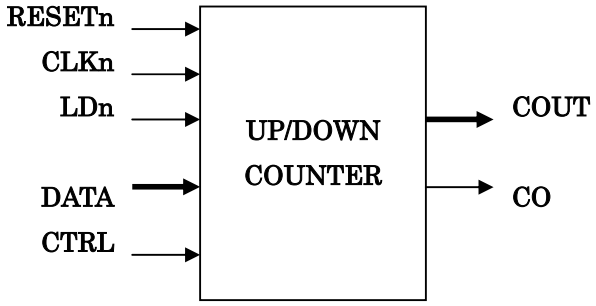

Figure 1. Block Diagram of Up/Down Counter

Port	Port type	Port Width	Description
RESETn	Input	1	Reset input, active low
CLKn	Input	1	Single phase system clock, active low
LDn	Input	1	Load data on DATA port, active low
DATA	Input	4	DATA input
CTRL	Input	1	Defines Direction of Counting
COUT	Output	4	Counter output
CO	Output	1	Indicates overflow

Table 1. Pinout Description for Up/Down Counter

Assignment: Create two different versions of the design model in VHDL, as follows: (1) using a predominantly behavioral modeling style; and, (2) using a predominantly structural style. Create one Entity declaration, which is to be stored in its own file, and create two different versions of the Architecture, each in its own file. Create one VHDL test bench structure that can be used to test both versions of the model, which is placed in its own file. Finally, create a Top Level for the model/test bench pairings that can be used to bind the elements of the system together.

Testbench: For testing purposes, use the Test Planning Worksheet to devise a test suite consisting of 6 test cases that will exercise both versions of your architecture realization. You should organize the test harness into a set of structured testing components that can be configured to execute in any order.

Simulation: For model verification, you should execute the test suite against both implementations of the Counter interface, collecting waveform output for your test cases, showing (1) that the models indeed show the same response to the test inputs, or (2) if these models behave differently, include an explanation as to why they are different.

Later, we will return to this example to contrast the modeling styles in terms of their impact on circuit synthesis using the Synopsys FPGA Compiler-II® tools.