Name: _____

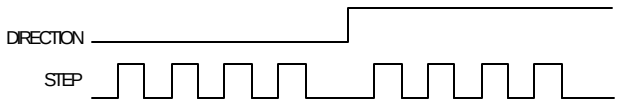
CoE 1502 Final Exam December 2001

You are given 2 hours for the exam. This is your own effort, therefore no communication with other students is allowed. Direct any questions to the professor or the TAs. All exam question sheets must be turned in along with the printouts.

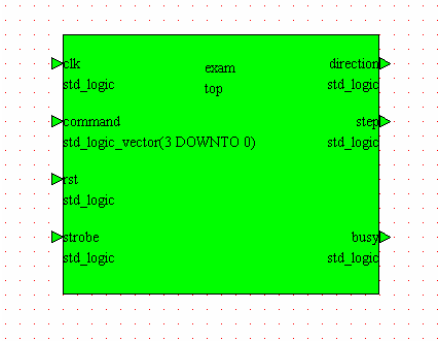
Complete the following exam by implementing the design in Renoir and simulating in ModelSim. You should turn in hard copies of all design source files and a printout of the wave file output from the do-file included in "examlib".

Stepping Motor Controller

Computer controlled mechanical positioning systems often use devices called stepping motors. For this exam you will implement a controller for a linear stepping motor device that operates as follows. The motor has two inputs: STEP and DIRECTION. The motor arm is moved in or out in a fixed unit, "a step", for each pulse received on the STEP input. There are 16 possible positions for the motor. The direction of movement is determined by the value of the direction input (0=out, 1= in). For example, operating the signals as shown below would cause the motor to move 4 steps outward, followed by 4 steps ininward.



For this exam, you are to design a controller for this device with an interface as defined by the symbol below:



The signals on this symbol are defined as

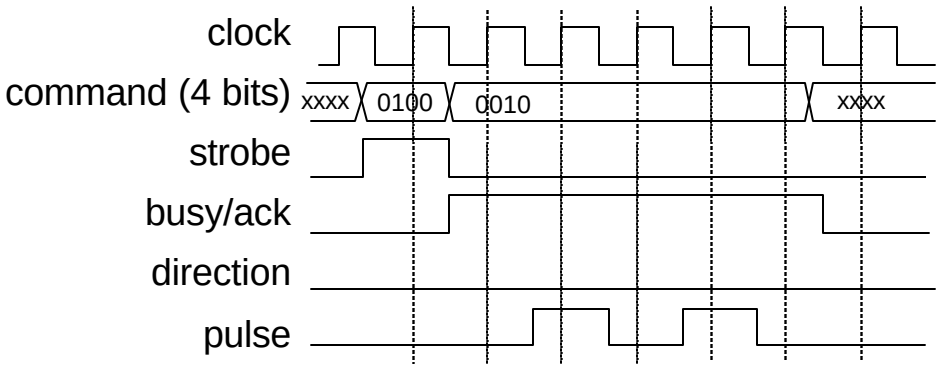
Signal/Bus	I/O	Description
command (4 bits) strobe (1 bit)	Input Input	Controller command words (4-bit nibbles) are transferred here When high, indicates that the first nibble of a command is on the command input
busy	Output	When high, indicates that the controller has recognized a command and is working on completing it. Also indicates that the source device should place the second nibble of a command on the input.
clk	Input	System clock
rst	Input	System reset
step	Output	Step pulse output to the stepper motor
direction	Output	Direction control to the stepper moter

There are three commands defined for the controller as defined in the table below

Command	Nibble 1	Nibble 2	Effect
STEP IN	1000	<step amount> (4-bits)	Causes the controller to step the motor in “step amount” number of steps
STEP OUT	0100	<step amount> (4-bits)	Causes the controller to step the motor out “step amount” number of steps
HOME	0010	None	Causes the controller to return the motor to the zero position.

The STEP IN and STEP out command are two nibbles (4 bits) long and require two transfers on the command inputs. The HOME command requires only a single nibble. In the case of two nibble commands, the source device will place the second nibble on the bus after the controller makes the BUSY/ACK signal high.

The following timing is an example STEP OUT 2 command.



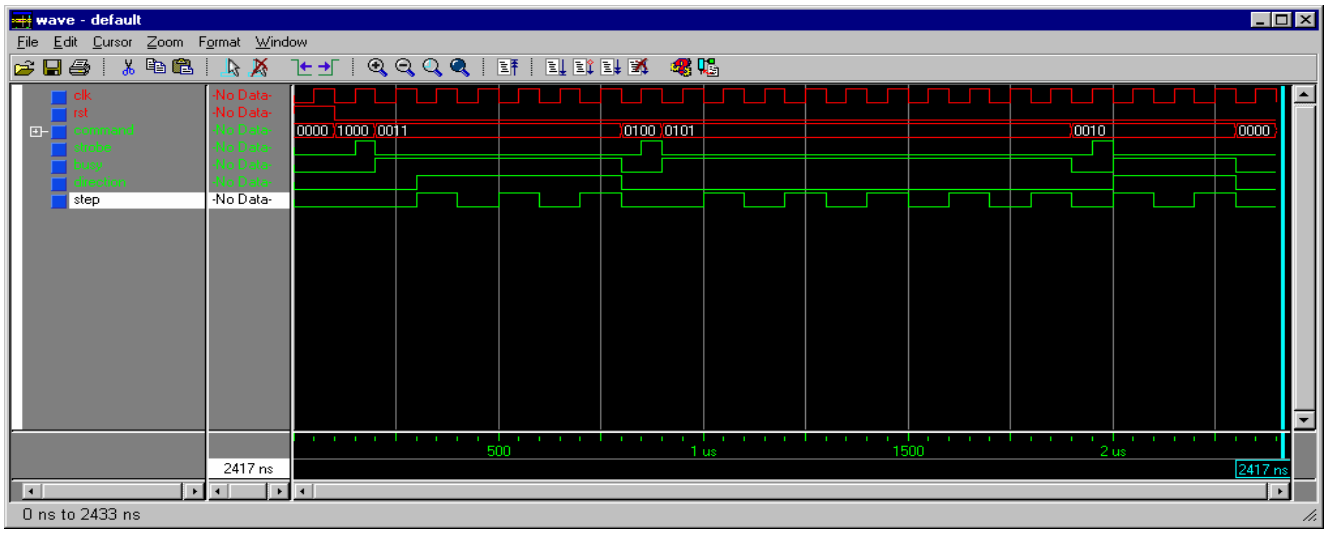


Figure 4: ModelSim results for correct solution

Implementation Notes

Your top level block diagram MUST BE NAMED 'top'. Create a new library called 'exam', mapped in your HOME DIRECTORY on the H drive. You will do your design in this library.

In order to implement this system, you will need access to the up/down counter component and a testbench that we have designed for you. It is in the examlib library (at I:\1502\examlib) and you'll have to set this up.

Submission

When you're done with your project, open the top_tb component in the examlib library and simulate it. When Modelsim opens, simply run the "exam.do" file which is located in I:\1502\examlib directory. This is the testbench for your design.

Components We Provide

The counter has the following symbol:

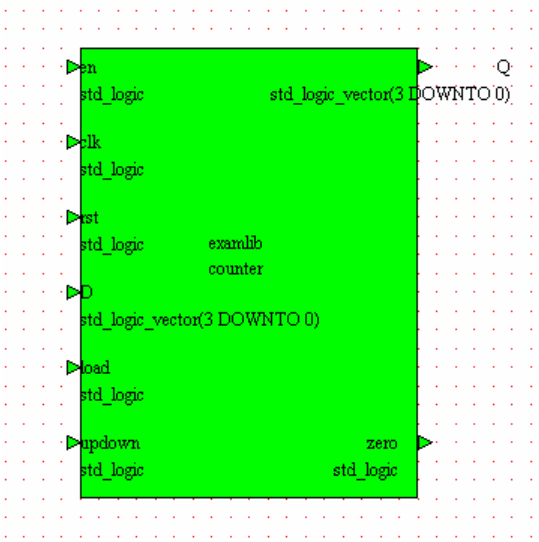


Figure 5: Up/down counter symbol

Signal/Bus	I/O	Description
en, clk, rst	input	Standard register en, clk, rst
load	input	Assert to load counter from D (counter must be enabled)
D	input	Value to be loaded
updown	input	Deassert to count down, assert to count up (load must be low and counter must be enabled)
Q	output	Current value of counter
zero	output	Asserted when counter value is 0

Note: This counter is rising edge activated. If you're unsure about the operation of the counter, you can view its VHDL code.

Hints

- The total design must have exactly one state machine controller (which you will design) and two of the up/down counters. These three components will be wired up in your block diagram.