CPE462 - VHDL: Simulation and Synthesis - Fall'11

Prof. Nuno Alves (nalves@wne.edu), College of Engineering

Homework Assignment #12

Due Date: Friday, October 28th 2011

Homework policy: If I see something unusual about your work, you can be sure I will ask you about it. If you are unable to explain it to me why you chose some implementation method I will have to consider your entire solution wrong. Now, what is my definition of "unusual work"? Well, things like suspicious parts of code that may have been copied from a website or chunks of your implementation that are very close to one of your colleagues. I really encourage communication amongst yourselves, and I want you guys to have a good time learning while doing these exercises... but please don't blindly copy code. If you really must, at least acknowledge your sources.

NOTE: You are NOT allowed to initialize variables inside processes!

1. Design a circuit capable of counting the number of clock events; the number of rising edges+ falling edges, as shown on Figure 1. Print out the wave-form using a clock stimulator with a 10ns period

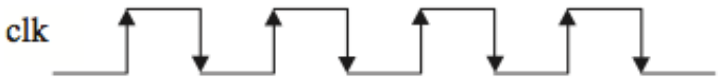


Figure 1 - Example of rising and falling edges

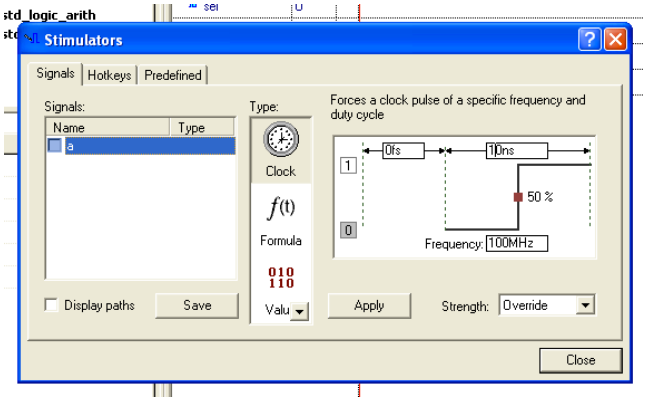
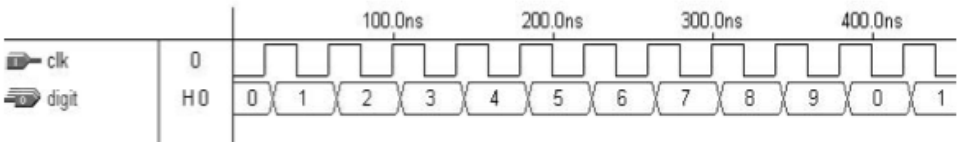


Figure 2 - How to set-up a clock stimulator in Active HDL

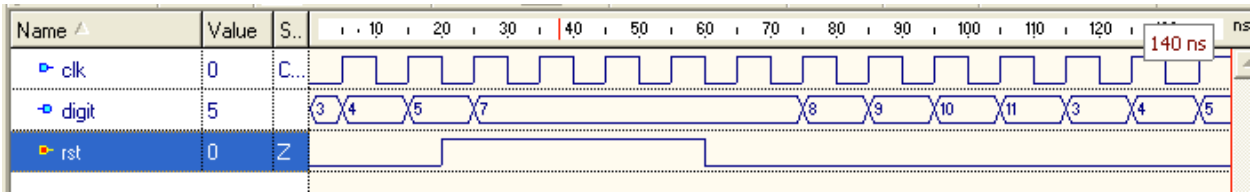
2. In class we discussed a one-digit counter. Below is the code we looked over in class that will implement a 0 to 9 to 0 counter.

```
1  ------------------------------------------
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.all;
4  ------------------------------------------
5  ENTITY counter IS
6     PORT (clk : IN STD_LOGIC;
7           digit : OUT INTEGER RANGE 0 TO 9);
8  END counter;
9  ------------------------------------------
10 ARCHITECTURE counter OF counter IS
11 BEGIN
12    count: PROCESS(clk)
13       VARIABLE temp : INTEGER RANGE 0 TO 10;
14    BEGIN
15       IF (clk'EVENT AND clk='1') THEN
16          temp := temp + 1;
17          IF (temp=10) THEN temp := 0;
18          END IF;
19       END IF;
20       digit <= temp;
21    END PROCESS count;
22 END counter;
23 ------------------------------------------
```



Before you start any modification, include another signal in your entity called **rst** of type std_logic. I would like you to modify this counter such that: The signal **digit** will count from 3 to 11 to 3 and the variable **temp** has the initial value of 7 (so when you start the circuit or when **reset** is triggered, you will start counting at 7).

Your output should look something like:



Extra-credit:

The following image shows the top-level diagram of a 7-level priority encoder. The circuit must encode the address of the input bit of highest order that is active. "000" should indicate that there is no request at the input (no bit active). Write a solution of this circuit only using sequential code!