ES 4 Individual Assessment 2

Tufts University

Due Tuesday, 11 May, 11:59 pm on Gradescope

Instructions:

1. This assignment is intended to help you and I evaluate your progress by pushing you to apply what you know in new ways. This may be a final assignment, but it's not a "final exam" that asks to you regurgitate everything we've talked about this semester.
2. You may consult your notes, the textbook, other books, or the internet while working on this assignment. However, you may not discuss it with anyone else (whether in the course or not) until after the deadline.
3. Please print this out or mark up the PDF directly, rather than using your own notebook. It makes it much easier to grade if everyone's answers are in the same place on the page. Submit your completed assessment on Gradescope, just like the homework.
4. If you have a question about the assessment, please post an *instructors-only* question on Campuswire.
5. Before starting, please put an 'X' somewhere on each line to indicate how confident you feel with each of the concepts so far:

   Flip-flops

   Clueless 0 1 2 3 4 5 6 7 8 9 10 Very confident

   Timing diagrams

   Clueless 0 1 2 3 4 5 6 7 8 9 10 Very confident

   Sequential logic with VHDL

   Clueless 0 1 2 3 4 5 6 7 8 9 10 Very confident
6. Please sign the statement below and include this page with your submission.

I certify that I have neither given nor received unpermitted aid on this assignment.

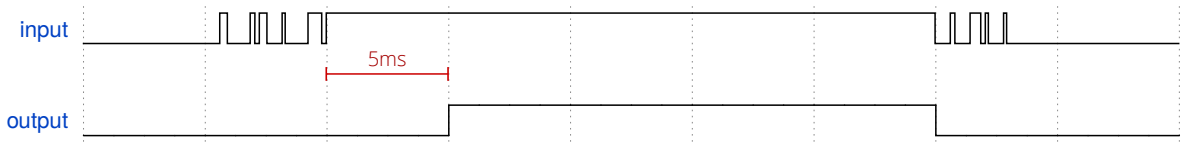
Signature: _____ Date: _____

Name: _____

Question 1: Debouncing

[10 pts] Since switches operate by mechanically flipping a metal contact, there is a possibility that the contact will rebound briefly when it hits the open or closed position. It setttles and makes continuous contact within milliseconds, but because the FPGA runs at tens of megahertz, it will observe all of these bounces and treat them as unique button presses.

Design a digital circuit (i.e., draw the logic diagram for a circuit) that will set the output to 1 if and only if the input has been 1 continuously for about 5 milliseconds.[1] Assume the FPGA source clock is 12 MHz.



Hint: If you search for this on the internet, you'll find lots of examples of using RC circuits or SR latches with SPDT switches. This problem is asking for an all-digital solution that works with SPST switches.

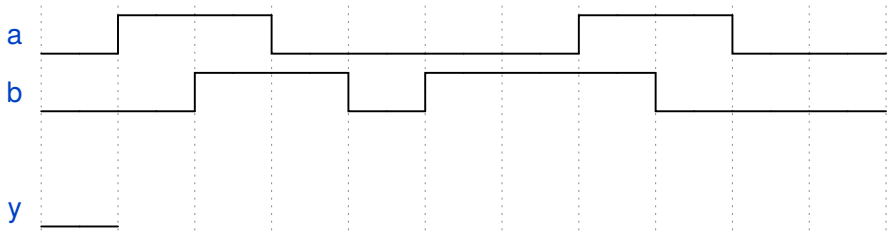
[1] The 5 ms is arbitrary; you can use anything from 4 to 6 ms if it makes your circuit simpler.

Question 2: Simulation and synthesis

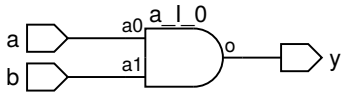
Suppose we have the following VHDL entity:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity makethis is
  port(
    a : in std_logic;
    b : in std_logic;
    y : out std_logic
  );
end;

architecture synth of makethis is
begin
  process(a) begin
    y <= a and b;
  end process;
end synth;
```

(a) [4 pts] Complete the timing diagram for this design to show what the output `y` will be as a function of `a` and `b`.



(b) [4 pts] Synthesizing this entity with Radiant or Yosys[2] gives the circuit below.



Explain to the best of your ability why synthesis produces this circuit.

[2] Yosys is the backend used to generate logic diagrams on VHDLweb.

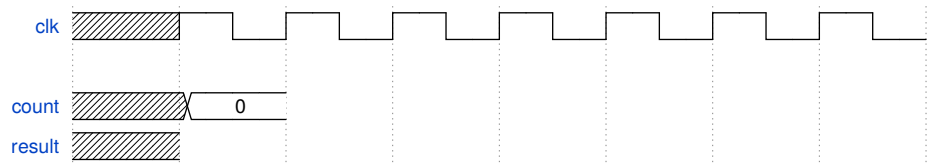
Question 3: Binary counters

[12 pts] Suppose we have the following VHDL entity which describes a simple 8-bit binary counter:

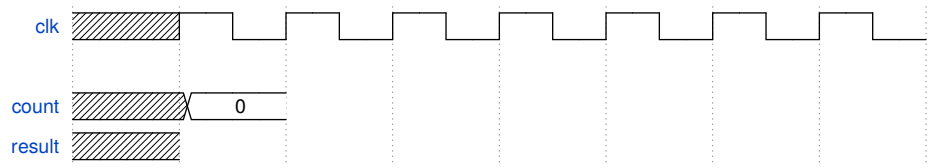
```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity counter is
  port(
    clk : in std_logic;
    result : out std_logic_vector(7 downto 0)
  );
end;
```

For each of the three architectures below, complete the timing diagram, and draw a logic diagram that would implement the circuit described by the architecture. *Hint: They're on separate pages so you have plenty of room to draw, but it may be helpful to compare them side by side.*

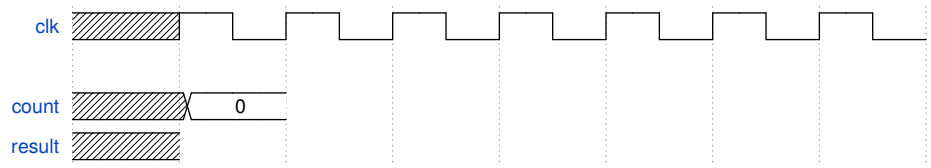
```
architecture A of counter is
signal count : unsigned(7 downto 0) := 8d"0";
begin
  process(clk) begin
    if rising_edge(clk) then
      count <= count + 1;
      result <= std_logic_vector(count);
    end if;
  end process;
end A;
```



```
architecture B of counter is
signal count : unsigned(7 downto 0) := 8d"0";
begin
  process(clk) begin
    if rising_edge(clk) then
      result <= std_logic_vector(count);
      count <= count + 1;
    end if;
  end process;
end B;
```



```
architecture C of counter is
signal count : unsigned(7 downto 0) := 8d"0";
begin
  process(clk) begin
    if rising_edge(clk) then
      count <= count + 1;
    end if;
  end process;
  result <= std_logic_vector(count);
end C;
```



Question 4: Parity checker

[8 pts] Complete the parity-checker problem on VHDLweb. As usual, you don't need to turn anything in besides completing the problem; your code is automatically saved each time you compile. And as always, make sure you're signed in, or you won't get credit.

Question 5: Reflection

(a) [1 pt] How long did you spend on this assessment?

(b) [1 pt] Now that you've completed the assessment, how well do you feel you understand each of the following topics?

Flip-flops

Clueless 0 1 2 3 4 5 6 7 8 9 10 Very confident

Timing diagrams

Clueless 0 1 2 3 4 5 6 7 8 9 10 Very confident

Sequential logic with VHDL

Clueless 0 1 2 3 4 5 6 7 8 9 10 Very confident

If your ratings changed, why?