ELEX 3330 : Programmable Logic Devices
Term 201610

FINAL EXAMINATION
3:30 – 5:30 PM
May 24, 2016

Do not open this exam until you are told.

This exam is for:

Exam Version 1 A00123456

- *This exam has three (3) questions on six (6) pages. Answer all questions.*
- *The marks for each question are as indicated. There are a total of 24 marks (5 minutes/mark).*
- *Write your answers and all rough work in this exam paper* **and nowhere else**. *Make a note if you continue your answer on the back of a sheet.*
- *Books and notes are allowed. No electronic devices other than calculators are allowed.*
- *Each exam is equally difficult. Answer your own exam.*

Name: ______________________

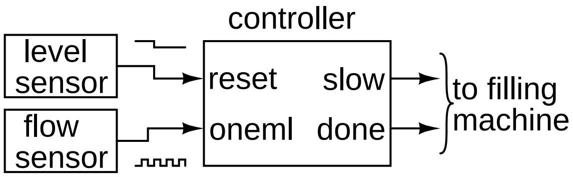
BCIT ID: ______________________

Signature: ______________________

Question	Mark
1	/10
2	/8
3	/6
Total	/24

Question 1 (10 marks)

This question asks you to design a controller for a bottle-filling machine.



The controller has two active-high inputs. `reset` is asserted whenever the level in the bottle is less than 2 ml. One pulse appears on `oneml` every time 1 ml is added to the bottle.

The controller has two active-high outputs: `slow` and `done`. `slow` should be asserted when approximately 402 ml or more has been poured into the bottle. Both outputs should be asserted when approximately 502 ml or more has been poured into the bottle. Both outputs should go low after `reset` is asserted (indicating a new bottle is in place).

(a) Write the VHDL entity declaration for the controller called `controller`.

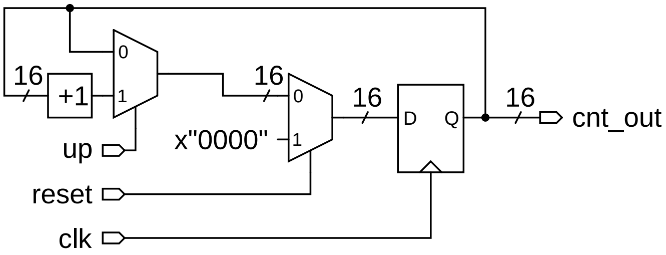
(b) Write a VHDL architecture that implements the controller.

Use `std_logic` and `unsigned` types only. You do not need to include `library` and `use` statements. You do not need to register the outputs. You can assume `reset` will be asserted before any counters overflow. Approximately means ±4.

Hints: use `oneml` *as a clock signal.* $2^9 = 512$.

Question 2 (8 marks)

Write a VHDL entity called `counter` and an architecture corresponding to the following schematic:



Use `std_logic` for single-bit signals (those where the bus width is not shown) and `unsigned` for multi-bit signals. The block labelled +1 adds one to its input. Signals labelled with names in the schematic are entity inputs or outputs. You may choose any name(s) for signals within the architecture.

Question 3 (6 marks)

Draw the schematic corresponding to the following VHDL code:

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

architecture rtl2 of exam is
   signal sig, sig_next : unsigned (7 downto 0) ;
   signal rst_n, sub, clk : std_logic ;
begin

  sig_next <=
    x"FF" when rst_n = '0' else
    sig-1 when sub = '1' else
    sig ;

  process(clk)
  begin
    if clk'event and clk = '1' then
       sig <= sig_next ;
    end if ;
  end process ;

end rtl2 ;
```

Label all signals with their names. Use conventional schematic symbols. Draw conditional assignments using multiple **two-input** multiplexers. Label each multiplexer input with the corresponding value of the select input. Show the bus widths for multi-bit signals.

ELEX 3330 : Programmable Logic Devices
Term 201610

FINAL EXAMINATION
3:30 – 5:30 PM
May 24, 2016

Do not open this exam until you are told.

This exam is for:

Exam Version 2 A00123456

- *This exam has three (3) questions on six (6) pages. Answer all questions.*
- *The marks for each question are as indicated. There are a total of 24 marks (5 minutes/mark).*
- *Write your answers and all rough work in this exam paper* **and nowhere else**. *Make a note if you continue your answer on the back of a sheet.*
- *Books and notes are allowed. No electronic devices other than calculators are allowed.*
- *Each exam is equally difficult. Answer your own exam.*

Name: ________________________

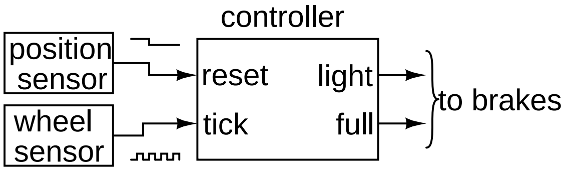
BCIT ID: ________________________

Signature: ________________________

Question	Mark
1	/10
2	/8
3	/6
Total	/24

Question 1 (10 marks)

This question asks you to design a controller for brakes on a drag-racing car.



The controller has two active-high inputs. `reset` is asserted whenever the car less than 2 m from the start. One pulse appears on `tick` every time the car moves forward 1 m.

The controller has two active-high outputs: `light` and `full`. `light` should be asserted after the car travels more than approximately 202 m. Both outputs should be asserted after the car travels more than approximately 252 m. Both outputs should go low after `reset` is asserted (indicating the car is back at the start).

(a) Write the VHDL entity declaration for the controller called `controller`.

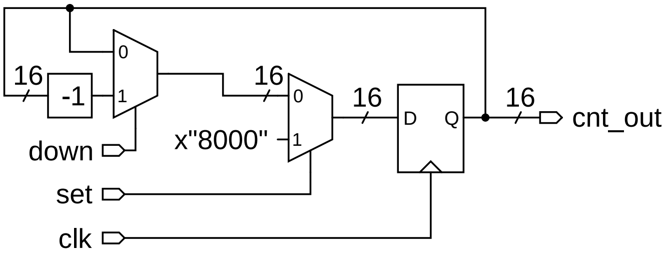
(b) Write a VHDL architecture that implements the controller.

Use `std_logic` and `unsigned` types only. You do not need to include `library` and `use` statements. You do not need to register the outputs. You can assume `reset` will be asserted before any counters overflow. Approximately means ±4.

Hints: use `tick` *as a clock signal.* $2^8 = 256$.

Question 2 (8 marks)

Write a VHDL entity called `counter` and an architecture corresponding to the following schematic:



Use `std_logic` for single-bit signals (those where the bus width is not shown) and `unsigned` for multi-bit signals. The block labelled `-1` subtracts one from its input. Signals labelled with names in the schematic are entity inputs or outputs. You may choose any name(s) for signals within the architecture.

Question 3 (6 marks)

Draw the schematic corresponding to the following VHDL code:

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

architecture rtl1 of exam is
   signal sig, sig_next : unsigned (7 downto 0) ;
   signal rst, add, clk : std_logic ;
begin

  sig_next <=
    x"00" when rst = '1' else
    sig+1 when add = '1' else
    sig ;

  process(clk)
  begin
    if clk'event and clk = '1' then
       sig <= sig_next ;
    end if ;
  end process ;

end rtl1 ;
```

Label all signals with their names. Use conventional schematic symbols. Draw conditional assignments using multiple **two-input** multiplexers. Label each multiplexer input with the corresponding value of the select input. Show the bus widths for multi-bit signals.