THE OPEN UNIVERSITY OF SRI LANKA
BACHELOR OF TECHNOLOGY - LEVEL 6
ECX6236 – PROCESSOR DESIGN
FINAL EXAMINATION 2014

DURATION: THREE HOURS

Date : 10th September 2015 **Time : 09:30 – 12:30 Hrs**

Answer **three** questions including the question in Section A and selecting two from Section B. Refer the Annexure for syntax of VHDL instructions

Section A

The following question is compulsory. It carries 70 marks.

1. In order to improve the transportation system in Sri Lanka, a number of expressways has been proposed and three of them are functioning at the moment. As in other countries toll (user fee) is collected for the use of expressways. Toll depends on the vehicle type, distance traveled (entry and exit point), and other factors. Generally toll collection has been done by manually. Nevertheless, Electronic Toll Collection (ETC) systems are also functioning in some expressways parallel to the manual system. ETC helps to identify vehicle and collect the toll electronically without stopping the vehicle when using a dedicated lane for ETC. Consequently it will reduce the delay at the toll booths.

   Your task is to design a special purpose processor for ETC (PETC), which can be used for developing a special unit to be used in expressways to collect toll either manually or electronically. The main functions of the ETC unit are vehicle identification, vehicle classification, transaction processing and communication with the server. Your scope is to design the processor, which controls the subunits of the ETC unit. You may assume that the communication, vehicle identification and classification will be handled by different subunits. Processor will control and access these subunits to perform necessary actions according to its executing instruction. Therefore the instruction set architecture (ISA) of the processor should facilitate to program the processor to get all functionalities of ETC unit according to inputs and give appropriate outputs accordingly.

   The ISA should be in a standard instruction format similar to an instruction set in general purpose processors (in the way of Opcode and Operands in an instruction format). Moreover it should be capable enough to develop programs for various ETC units, wherever they are installed e.g. at entry point or exit point, at manual or electronic toll collection points.

   As this is a special purpose processor, your design may differ from general purpose processors. You may include special functional units/ components along with a description. Clearly state any other assumptions you made (if any).

   a) Draw a block diagram to show how to deploy the processor (PETC) you designed (once it is fabricated) when manufacturing ETC unit.
   b) Write a short description on the working procedure of the ETC unit indicating the internal functionality of your processor and subunits of ETC unit. You have to show clearly how the toll is calculated according to factors such as vehicle type, entry and exit point and so on.
   c) Accordingly, identify the necessary instructions and design an ISA for this processor.

d) Using your ISA write a program which can calculate the toll manually or electronically and issue a receipt or update customer account respectively according to the lane used.
e) Draw a block diagram for the processor indicating all input and output signals. Clearly state all functions of each block inside the processor and show the data path.
f) Identify entities for which you need to write VHDL codes to synthesise the processor.
g) Write the behavioural/ structural VHDL codes for each entity except for the Control Unit of the processor. You may define the Control Unit as a component.

Section B

*Answer **two** questions from this section. Each question carries 15 marks.*

2.
a) Integrate the VHDL codes for different entities in *Question (1.g)* of *Section A* to obtain a complete VHDL code for the PETC.
b) Name the modelling methods available in VHDL. Which of the modelling methods is suitable for PETC processor? Justify your answer.

3.
a) Construct the state diagram of the Control Unit of the processor you designed in *Question 1*.
b) Briefly explain the steps that you have to follow to implement PETC on an FPGA.

4.
a) Briefly describe how you can estimate the performance of a processor. Estimate the performance of the PETC you designed in *Question 1.*
b) Which factors do you need to consider to estimate the cost of the PETC?

5.
a) Draw a block diagram for *D Flip-flop* with *1-bit* input signals *D, Clock, SET,* and *CLR* and write a Behavioural VHDL code for the same.
b) Draw a schematic diagram for a *4-bit serial-in serial-out shift register* (Fig 1) using the *D Flip-flop* in *Question 5.a.* The shift register shifts its stored data by one bit position at each tick of the clock. The serial input, S_{in}, receives a bit to be shifted into the register at each clock tick and the bit will be sent out at the serial output, S_{out}, after 4 clock ticks. When *CLR* is 1, shift register sets all its bits to 0 at a clock tick, otherwise it will function as described.

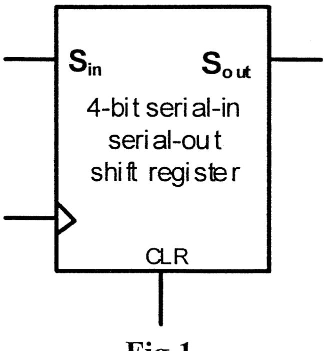

Fig 1

c) Write a Structural VHDL code for the *4-bit serial-in serial-out shift register* according to your schematic diagram drawn for the *Question 5.b.*

Annexure

Syntax of selected instructions of the VHDL

⌧
```
ARCHITECTURE architecture_name OF entity_name IS
  [declaration part]
BEGIN
  Concurrent statements part
END architecture_name
```

⌧
```
CASE expression IS
  WHEN value=> statements;
  WHEN value=> statements;
  WHEN OTHERS statements;
END CASE;
```

⌧
```
COMPONENT component_name
  PORT (port1_name : port1_type;
        port2_name : port2_type;
        ...);
END COMPONENT [component_name];
```

⌧
```
ENTITY entity_name IS
  PORT (port1 : port1_type;
        port2 : port2_type;
        ...);
END entity_name;
```

⌧
```
IF condition THEN
  Sequence of statements
  {ELSIF condition THEN
     Sequence of statements}
[ELSE
  Sequence of statements]
END IF;
```

⌧
```
LIBRARY library_name;
```

⌧
```
Instance_label: component_name PORT MAP (first_port, second_port,
                                          third_port, ...);
Instance_label: component_name PORT MAP (formal1=> actual1,
                                         formal1=> actual1,
                                         formal1=> actual1, ...);
```

⌧
```
[process_label:] PROCESS (signal1, signal2, ...)
                   [declaration part]
                 BEGIN
                   Sequential statements part
                 END PROCESS;
```

⌧
```
SIGNAL signal_name : signal_type;
```

⌧
```
TYPE type_name;
```

⌧
```
USE library_name.type_expression.inclussion;
```

⌧
```
WAIT FOR time_expression;
WAIT ON signal1, signal2, ...;
WAIT UNTIL condition;
```

⌧
```
WHILE condition LOOP
  Sequential statements
END LOOP;
```