THE OPEN UNIVERSITY OF SRI LANKA
BACHELOR OF TECHNOLOGY - LEVEL 6
ECX6236 – PROCESSOR DESIGN
FINAL EXAMINATION 2015

DURATION: THREE HOURS

Date : **09th December 2016** **Time :** **0930 - 1230 Hrs**

Answer **three** questions including the question in Section A and selecting two from Section B.

Section A

The following question is compulsory. It carries 70 marks.

1. Databases are extensively used in applications today. Performance of these applications are mostly depending on the access time of the databases (disk access time) and the processing time of database queries. However these periods of time can be reduced if a major part of the database is cached in a special memory. Moreover processing of database queries in a separate unit will increase the performance further.

   Therefore it is decided to manufacture a pluggable database access card (DAC) to improve the performance of database applications. The DAC will be designed based on a special processor for database processing (PDP). Furthermore PDP should have a special cache memory for saving database tables for processing.

   Your task is to design the special processor – PDP, which can be fabricated later. Consequently the PDP can be used for building DAC as a plugin card for a computer.

   As the PDP is used only for the mentioned purpose, your design may differ from general purpose processors. The Instruction Set Architecture (ISA) of the PDP should be capable enough to develop necessary codes for an application with database. Before processing you need to bring a portion of the database into the special cache memory and update the original database located in the hard disk before discarding it from the memory. Your ISA should support basic operations related to a database such as reading and saving entries from/to a table, search an entry in a given field and retrieve information accordingly, and so on. You may also need to include other instructions in order to support other relevant operations.

   Your design of the processor must be simple. You may include special functional units/ components along with a description. You may limit number of tables and number of entries in a table when answering the questions.

   a) Draw a diagram to show how to deploy the PDP (once it is fabricated) for developing DAC for database applications. (*You do not need to design the DAC but need to identify the inputs and outputs of the PDP relevant to components of such systems.*)
   b) Write a short description on the working procedure of a DAC indicating the internal functionality of your processor (PDP).
   c) Accordingly, identify the necessary instructions needed for this processor and design the ISA.
   d) Using your instruction set write a simple program for accessing information from a database with two tables.
   e) Draw a block diagram for the processor. Clearly state all functions of each block inside the processor and show the data path. Indicate all input and output signals of the processor.

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. (Refer the Annexure for syntax of VHDL instructions).

Section B

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

2.

a) Briefly describe the tasks of a Processor Designer.
b) Construct the state diagram of the Control Unit of the PDP in *Question 1*.

3.

a) What are the steps that you follow to design a hardwired control unit of a processor?
b) Integrate the VHDL codes for different entities in *Question (1.g)* of *Section A* to obtain a complete VHDL code for the PDP.

4.

a) Briefly explain the factors that should be considered for estimating the price of the PDP designed in *Question 1*?
b) Estimate the performance (in CPI) of the PDP you designed in *Question 1*.

5.

a) When you implement the PDP on an FPGA you will not be able to check all internal signals of the processor other than input/output signals. Propose a technique that you can use in order to check all signals inside the FPGA. You are free to add new components into your design.
b) Construct a unit (draw a schematic diagram) to implement the following function using the *8-bit Full Adder*.

$$y = \sum_{i=1}^{n} i$$

Calculation must be stopped when it reaches the maximum possible value that can be handled by the Adder.

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;
```