(27)

The Open University of Sri Lanka
Faculty of Engineering Technology
Department of Electrical and Computer Engineering

Study Programme	: Bachelor of Technology Honours in Engineering
Name of the Examination	: Final Examination
Course Code and Title	**: EEX6536/ECX6236 Processor Design**
Academic Year	: 2017/18
Date	: 10th February 2019
Time	: 0930-1230hrs
Duration	: **3 hours**

General Instructions

1. Read all instructions carefully before answering the questions.
2. This question paper contains four (4) questions in SECTION A and four (4) questions in SECTION B on five (5) pages.
3. Answer ALL questions in SECTION A. [70 Marks], and answer any TWO questions from SECTION B. [30 Marks]
4. The answer to each question should commence from a new page.
5. Refer the Annexure of the VHDL syntax given in page five (5) to write VHDL code.
6. This is a Closed Book Test (CBT).
7. Answers should be in clear handwriting.
8. Do not use Red colour pen, and clearly state your assumptions if any

SECTION A:

Answer ALL questions. [70 Marks]

Processor for Students' Learning Analytics (LAP)

Read the following scenario (as shown in italic). You are required to analyse the requirements of the ***LAP*** and design it according to the given specifications.

Today, Data analytics have significant role in BIG Data in Data Science for real-world applications. One identified particular area is the Education sector in the world. In this sector, learning analytics have vital work to monitor students' learning behaviours.

Learning analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs (wiki).

To fulfil this requirement, it is required to manufacture a special hardware module to accelerate the learning analytics methods for continuous monitoring of students' learning behaviours.

As an initial step, your task is to design a simple special-purpose experimental processor (called LAP) to calculate Overall Continuous Assessment Marks (OCAM) of a course. The criteria of the OCAM is given in the equation below.

OCAM = 2B-AVG(TMA1, TMA2, CAT1)*0.3 + AVG(LAB1, LAB2)*0.3 + DP*0.4

Where 2B-AVG - Average mark of the best two activities from TMA1, TMA2, and CAT1;
AVG - Average mark of LAB1 and LAB2;
DP - marks of the design project;
* - multiplication symbol.

Assume that there are 50 learners registered to the same course and all marks are available in the external database.

The ISA (Instruction Set Architecture) of LAP should be capable enough to develop a necessary program (codes) for the above application. Your ISA should provide basic operations related to computing average marks, select best activity, and other necessary operations. You may need a special memory (cache memory) to retrieve a portion of the database with activity marks (assume that the database is given to you) and other digital logic devices.

The designed processor must be simple. You may include other additional hardware units/components that align with the scenario. State all other assumptions clearly (if any), when you are answering the following questions.

[Q1]

(a) Draw a diagram and show how to deploy the LAP (once it is fabricated) for developing the special hardware module for the given application. (Show the inputs and the outputs of the LAP with the database). [10 Marks]

(b) Briefly describe the operation of the diagram drawn in Q1 (a) indicating internal functionality of LAP. [05 Marks]

[Q2]

(a) Tabulate the required instructions of LAP and design the ISA to fulfil the given specifications. [15 Marks]

(b) Using the ISA designed in Q2 (a), write a simple program to compute OCAM of the learners from the given database. [05 Marks]

[Q3] Draw a Block diagram for the LAP. Clearly state all functions of each block inside the processor and show the data path. Indicate all input/output signals of the LAP. [15 Marks]

[Q4]

(a) Identify the entities for which you need to write VHDL codes to synthesize the LAP. [05 Marks]

(b) Write behavioural VHDL codes for each entity except for the control unit of the LAP. (You may define the control unit as a component.) [15 Marks]

SECTION B:

Answer any TWO questions. [30 Marks]

[Q5]

(a) Construct a state diagram of the control unit of the LAP in Section A [10 Marks]

(b) Draw an ASM chart of the control unit based on the state diagram drawn in Q5 (a). [05 Marks]

[Q6]

(a) Briefly explain the factors that should be considered for estimating the price of the LAP designed in Section A. [08 Marks]

(b) Estimate performance (in CPI) of the LAP designed in Section A. [07 Marks]

[Q7]

(a) Briefly describe the task of a processor designer. [07 Marks]

(b) Briefly describe the steps needed to follow to design a hardwired control unit of a processor. [08 Marks]

[Q8]

(a) Draw a diagram and explain the operation of generic hardwired control unit by giving an example instruction. [07 Marks]

(b) Explain Fetch, Decode, and Execute cycle of the same instruction used in Q8 (a) with control signals and operations performed using RTL. [08 Marks]

Annexure

The 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;
```