THE OPEN UNIVERSITY OF SRI LANKA
BACHELOR OF TECHNOLOGY - LEVEL 6
ECX6236 – PROCESSOR DESIGN
FINAL EXAMINATION 2012

DURATION: THREE HOURS

Date : **15th August 2013** **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. Electronic Sorting Machines (ESMs) are widely used in the industry. They are used for various purposes. For example, in the tea industry high quality leaves are selected by ESMs. In farms fruits and vegetables are graded using these machines. In order to achieve these tasks sorting in ESMs has been based on the physical properties of the sorting object i.e. colour, weight, and size and so on.

   In order to manufacture different kinds of ESMs it was decided to design a special purpose processor. This processor which is an Application Specific Integrated Circuit (ASIC) will help to simplify manufacturing and maintaining ESMs.

   Your task is to design a special purpose processor for ESMs (PESM). As the processor is used only for this purpose all inputs and outputs should be relevant to any kind of ESMs. The processor should receive all commands 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). According to the command received, the processor must perform the relevant task and output signals required for ESMs, if necessary. Since there can be several input ports in the processor to read different property values, you may need to have special instructions to configure the ports accordingly.

   The processor that you are going to design should be able to perform the sorting in any kind of ESMs. It should be able to receive appropriate values of the selected property of the object. Also values of different properties may be read simultaneously. Subsequently, the PESM should do necessary calculations if needed and it should output signals for controlling other units/devices in ESMs. In addition to that, there should be a mode to input and update necessary information and also for updating the program.

   As this is a special purpose processor, your design may differ from general purpose processors. You may use internal memory/registers to keep all the information and may include special functional units/ components along with a description. Clearly state any other assumptions you made (if any).

   a) Draw a diagram to show how to deploy the processor, PESM you designed for ESMs (once it is fabricated) when manufacturing ESMs.
   b) Write a short description on the working procedure of the complete system indicating the internal functionality of your processor.
   c) Accordingly, identify the necessary instructions needed for this processor.
   d) Using your instructions write a program to sort any kind of objects according to the selected properties for a given sorting criteria.

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) Integrate the VHDL codes for different entities in *Question (1.g)* of *Section A* to obtain a complete VHDL code for the PESM.
b) Briefly describe how you are estimating the performance of a processor. Estimate the performance of the PESM you designed in *Question 1.*

3.

a) How do you estimate the cost of a packaged integrated circuit?
b) Construct the state diagram of the Control Unit of the processor you designed in *Question 1.*

4.

a) Name three applications where you can use systolic architecture and describe briefly one of them.
b) Name five synthesising tools used in FPGA/ CPLD based design.
c) What are the advantages of using HDL for VLSI design?

5.

a) Write a Behavioural VHDL code for the *8-bit Full Adder* as given in the Fig 1.

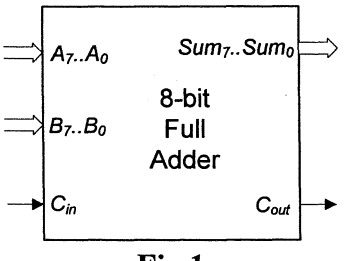

Fig 1

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.

c) Write a Structural VHDL code for the *unit* you designed according to your schematic diagram drawn for the question 1.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;
```