THE OPEN UNIVERSITY OF SRI LANKA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FINAL EXAMINATION 2008
BACHELOR OF TECHNOLOGY - LEVEL 6 064

ECX6236 – PROCESSOR DESIGN

DATE: APRIL 01, 2009 TIME: 0930-1230 HRS

Answer **three** questions from Section A and B, including question 1 in Section A.

Section A

This question is compulsory and carries 70 marks.

1.
Automation and the use of robots are nowadays rapidly increasing in the field of agriculture. Seedling is such a function performed by these robots. Seedling robots place seeds in regular spaces in fields and record the coordinates of position. Assume that you have already designed an application-tailored processor for a seedling robot.

It is possible to reduce the quantity of chemical fertiliser to a larger extent by using a robot that precisely sprays fertilizer to the exact location. This reduction causes to minimise the release of harmful inorganic fertiliser and the significant cost cut down on fertiliser.

You are to design a special functional unit to control the above process. Assume the system is supposed to spray fertiliser on the locations that you have already dropped seeds during seedling. Once developed, the unit will be implemented inside the existing processor which used to control the seedling process. Therefore your design should be interfaced with the internal entities of the processor and external entities that control the spraying process.

Your functional unit should start the process whenever it receives a trigger signal from the control unit of the existing processor and should generate a signal once the spraying is done at a particular location.

A readymade hardware module that sprays specified volume of fertiliser is available to you. With this module, you can specify eight volume levels (volume level 1 to volume level 8) to release fertiliser. Minimum volume of fertiliser is released by the volume level 1. The volume of fertiliser released by the next volume level is twice as the current volume level (E.g.: If x ml of fertiliser is released by volume level 1, volume level 2 will release $2x$ ml of fertiliser. Likewise volume levels 3, 4, 5, 6, 7 and 8 will respectively release $4x$, $8x$, $16x$, $32x$, $64x$ and $128x$ ml of fertiliser). Volume of the fertiliser container attached to the ready made hardware module is 100 times of the volume level 8.

In order to spray at a particular location, first, your functional unit should send the required volume level to spray unit. Then it should send a trigger signal to start spraying process. Once the spraying is done the spraying unit generates an acknowledgement. Therefore after triggering the spray unit, the functional unit has to wait until the acknowledgment of completion.

Your functional unit should track the remaining volume of the fertiliser in the container by recording the level of volume issued at each spray. Also it should generate a signal when the fertiliser is over.

Assume the locations of seeds are in a memory which is accessible to your special functional unit. This unit is expected to get the coordinates of the next location and send it to the mobility control unit. A memory word is 16-bits wide. Coordinates of a location (X and Y) is stored at two consecutive 16-bit memory words. Coordinates of two nearest locations are stored consecutively in the memory. X coordinate of the first location is always found at 0000H location. Assume the trigger signal send to the special functional unit by the existing processor indicates the arrival at the next location.

Once the new unit is ready it will be incorporated to the existing processor. Making changes to the existing processor is out of scope of your answer.

You may declare your own assumptions whenever it is necessary.

a) Identify inputs and outputs of the special functional unit. Indicate the width of each input and output.

b) Briefly explain the internal functionality of the special functional unit.

c) Identify the sub units/entities within the special functional unit. Clearly mention the function performed by each sub unit/entity.

d) Draw an internal block diagram of the special functional unit.

e) Draw timing diagrams or state diagrams as appropriate for each sub unit/entity identified in question 1(c).

f) Identify the number of clock cycles required for different operations that are to be performed by the special functional unit.

g) Write VHDL codes for a sub unit/entity which has at least five states in a state diagram/timing diagram drawn in question 1(e).

Section B

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

2.

a) Assume you have a finalised VHDL model of a hardware design and you are to implement it on a FPGA, briefly describe the key steps to be followed. You may present the answer under following topics: Synthesis, Place and Route and Verification

b) What is reconfigurable computing?

3.

a)

i) List three reasons behind the global trend in multi-core processor design.

ii) Maintaining the constant performance enhancement with the increasing number of cores has become a challenge in today's processor design. Identify a possible reason for that. Justify your answer.

b)

i) Distinguish three different modelling techniques available in VHDL.

ii) List two advantages of using a combination of modelling techniques in a VHDL design.

4.

a) List two significant reasons for implementing an Application Specific Instruction Processor (ASIP).

b) The requirement to connect entity P and Q are as follows. It should be possible to send a signal from P (via port B) to Q (via port A) and to send a signal from Q (via port A) to P (via port C) whenever necessary. Port B is an output port, port C is an input port and port A is an *inout* type port. Width of each port is 1- bit. Possible values at port A are 1, 0 or Z. possible values for port B are 0 or 1).

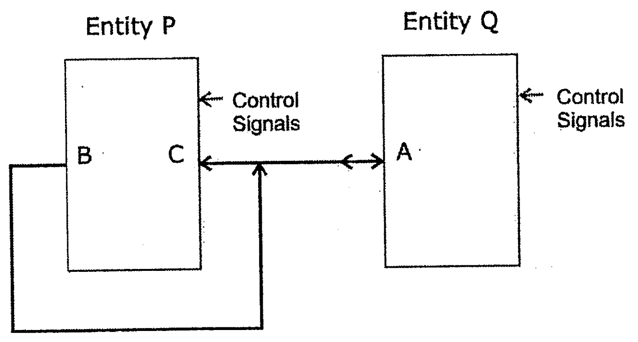

Figure 1

One proposal was to connect Entity P and Q as in figure 1. When simulated, it did not perform in the expected way.

i) What could be the problem with this model?

ii) Suggest how you can change this set up to overcome the problem and redraw the diagram accordingly.

5.
Draw a state diagram for the Entity A in figure 2.

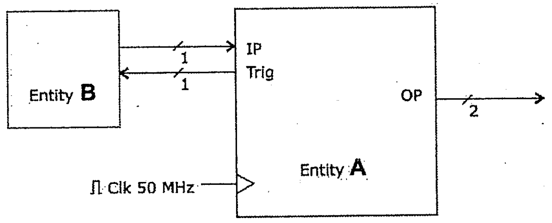

Figure 2

Entity A and Entity B are connected as shown in figure 2. Entity A is used to measure the width of a pulse generated by the entity B in terms of number of clock cycles.

- In order to initiate the measuring process, entity A is suppose to send a trigger pulse to the output port *Trig*.
- Then it counts the number of clock pulses during the period where input port IP is high.
- Response time of entity B is not known. If no high input received within 10 clock cycles, Entity A has to send the trigger pulse again.
- At the end of the counting process it sends a corresponding value as shown in the following table (table 1) to the output port *OP*. Respective output values, at output port *OP*, is shown below. The output will be effective from the next rising edge of the clock.
- The measuring process is to repeat by 50 ms intervals.

Table 1

Clock cycle count	OP (binary values)
1	00
2	01
3	10
4	11

Following is a sample timing diagram (figure 3) for the measurement of a pulse where the pulse count is 2. The respective binary value at output port OP is '01'.

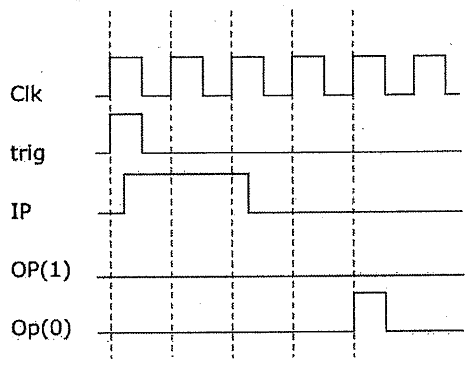

Figure 3

Assumptions:

- Maximum number of clock pulse count is 4.
- Entity B is positive edge triggered.
- Entity A samples port IP at each rising edge of the clock during the counting process.

Annexure

Syntax of Selected VHDL Statements

```
▪ 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;
```