RHODES UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

EXAMINATION: November 2019

COMPUTER SCIENCE HONOURS

MODULE: HLDA

Internal Examiners: Dr A. Herbert

MARKS: 120

DURATION: 2 hours

External Examiner: Prof. I. Sanders

GENERAL INSTRUCTIONS TO CANDIDATES

1. This paper consists of 9 pages with 3 sections and 16 questions.
2. All answers are to be written in the answer books provided. Please use a new book for each section.
3. You are required to answer all questions.
4. The Concise Oxford English dictionary may be used during this examination.

PLEASE DO NOT TURN OVER THIS PAGE UNTIL TOLD TO DO SO.

SECTION A: GENERAL THEORY [34 MARKS]

Question 1 **(4 Marks)**

Contrast a concurrent language such as VHDL with a sequential language like C#.

Question 2 **(4 Marks)**

How would one emulate sequential operation behavior in VHDL?

Question 3 **(6 Marks)**

What is a Lookup Table in an FPGA, and what is it used for? Explain your answer by use of an example.

Question 4 **(4 Marks)**

What physical factor determines the speed at which one can introduce and receive signals to a given logic design, and what would occur if one were to exceed this limit?

Question 5 **(4 Marks)**

Is a hardware description language like VHDL Turing complete? Explain your answer by use of an example.

Question 6 **(2 Marks)**

What does the constraints (.xdc) file do in a Vivado project?

Question 7 **(4 Marks)**

In the context of a Process Block, contrast the difference in behavior between a signal and a variable type.

Question 8 **(2 Marks)**

What is the “others” keyword used to signify in a case statement in VHDL?

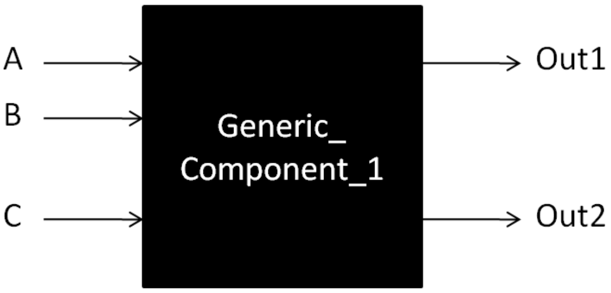
Question 9 **(4 Marks)**

Why might it be useful to write major functional blocks of VHDL code as separate components in a given project?

SECTION B: LANGUAGE USE [36 Marks]

Question 10 **(4 Marks)**

Complete the Entity Block below by completing the port instantiation code for the given RTL schematic. You can assume each input and output is a single bit.



```
ENTITY Generic_Component_1 IS
PORT(



    );
END                                               Generic_Component_1;
```

Question 11 **(4 + 2 = 6 Marks)**

Consider the two following snippets of VHDL code below:

Snippet 1:

```
A <= 10;
B <= 20;
C <= A + B;
```

Snippet 2:

```
C <= A + B;
A <= 10;
B <= 20;
```

a) Is Snippet 1 functionally equivalent to Snippet 2? Justify your answer.

b) Regarding Snippet 1, what concern should one have regarding the value assigned to C when either A and/or B's value is changed?

Question 12 **(4 + 6 = 10 Marks)**

Consider the following Process Block with existing std_logic inputs sel and clock in the Entity Block:

```
PROCESS (clock)
    IF (rising_edge(clock))
    THEN
        IF(sel = ‘1’)
        THEN
            -- do something here
        ELSE
            -- do something else here
        END IF;
    END IF;
END PROCESS;
```

a) Why is this code considered inefficient? Explain your answer with respect to how this given design’s logic would flow at runtime.

b) Optimise the given Process Block by rewriting the given VHDL code.

Question 13 **(2 + 6 = 8 Marks)**

Consider the following Architecture Block below with existing std_logic inputs a, b, and c, and std_logic output result:

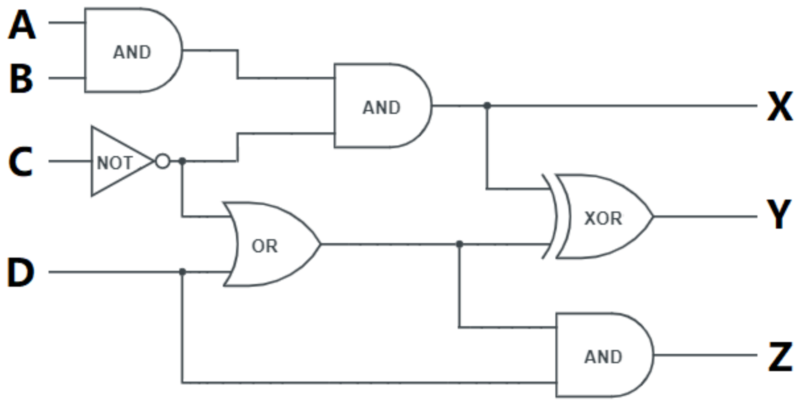
```
ARCHITECTURE behavior OF generic_component_2 IS
SIGNAL b_result : std_logic;
BEGIN
    b_result <= a AND b;
    b_result <= c XOR b_result;
    result <= b_result;
END behavior;
```

a) Why would an error arise during synthesis of this VHDL code?

b) Rewrite the given VHDL code by fixing the error identified in (a).

Question 14 **(8 Marks)**

Consider the following logic diagram:



Complete the given VHDL code below that would simulate the given logic diagram above.

```
library ieee;
use ieee.std_logic_1164.all;

ENTITY combi IS
PORT(
      A, B, C, D : IN  STD_LOGIC;
      X, Y, Z    : OUT STD_LOGIC
    );
END combi;

ARCHITECTURE behaviour OF combi IS

BEGIN



END behaviour;
```

SECTION C: PROBLEM SOLVING [50 Marks]

Question 15 **(25 Marks)**

An ALU is a component of modern processing devices that performs arithmetic operations. You are required to write the Architecture Block for an ALU that can perform the following four arithmetic operations: Addition, Subtraction, Multiplication, and Division. You can assume that for inputs A and B, that A always appears on the left-hand side of each operation, that is: A operation B.

In the given code below, the Entity Block receives signals A, B, and Op. A and B are treated as 8-bit binary inputs to the ALU, and Op is treated as a 2-bit binary input to select the required arithmetic operation.

The output signals from the Entity Block are Res and Err. Res is expected to show the 8-bit result of any requested arithmetic operation, and Err is used to signify that an error has occurred. The Err signal should be set to ‘1’ in the following cases:

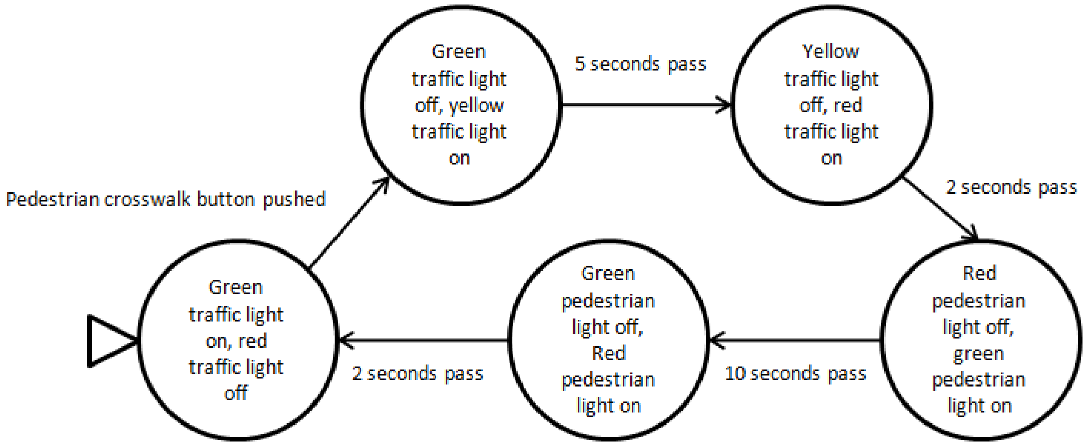
1. When an overflow occurs when performing an addition.
2. When an underflow occurs when performing a subtraction.
3. When B is “00000000” when performing a division (division by 0 error).

Err should be set to ‘0’ in all other cases.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_signed.all;
use ieee.numeric_std.all;

ENTITY alu IS
PORT(
      A   : IN  STD_LOGIC_VECTOR (7 DOWNTO 0);
      B   : IN  STD_LOGIC_VECTOR (7 DOWNTO 0);
      Op  : IN  STD_LOGIC_VECTOR (1 DOWNTO 0);
      Res : OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
      Err : OUT STD_LOGIC
    );
END alu;

ARCHITECTURE behaviour OF alu IS
BEGIN

END behaviour;
```

Question 16 **(25 Marks)**

Pelican is short for Pedestrian Light Controlled Crossing, and is commonly used to allow pedestrians to cross a road safely by stopping motor vehicles on a road by the use of traffic lights. Given below is a state machine that models a Pelican.


Green traffic light off, yellow traffic light on
5 seconds pass
Yellow traffic light off, red traffic light on
Pedestrian crosswalk button pushed
2 seconds pass
Green traffic light on, red traffic light off
2 seconds pass
Green pedestrian light off, Red pedestrian light on
10 seconds pass
Red pedestrian light off, green pedestrian light on

Your task is to complete the given VHDL code by implementing the given state machine. That is, you should design the logic flow for a pelican system.

In the given code below, the Entity Block receives signals Button and Clock. The signal Button is connected to the pedestrian crosswalk button in this system. The signal clock is connected directly to a 100 MHz clock, and can be used to accurately time all events in this system.

The signals output from the Entity Block are TRed, TYellow, TGreen, PRed and PGreen. The output signals TRed, TYellow and TGreen are connected to the red, yellow and green traffic lights facing the traffic, respectively. The output signals PRed and PGreen are connected to the red and green pedestrian lights facing the pedestrians, respectively.

You can assume that the system starts in the following state: Green traffic light and red pedestrian lights are on, all other lights are turned off, and no button press registered.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use                                        ieee.std_logic_unsigned.all;
use                                          ieee.std_logic_signed.all;
use                                               ieee.numeric_std.all;

-- continued on next page

ENTITY                            pelican                            IS
```

```
PORT(
      Button                  :        IN                 STD_LOGIC;
      Clock                       :       IN              STD_LOGIC;
      TRed                              :       OUT       STD_LOGIC;
      TYellow              :             OUT              STD_LOGIC;
      TGreen                      :          OUT          STD_LOGIC;
      PRed                              :       OUT       STD_LOGIC;
      PGreen                      :          OUT           STD_LOGIC
    );
END                                                         pelican;

ARCHITECTURE         behaviour         OF         pelican         IS
BEGIN

END                                                       behaviour;
```

END OF EXAMINATION