00088

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	**: EEX5351/EEX6351 Digital Electronic Systems**
Academic Year	: 2020/21
Date	: 12th February 2022
Time	: 1400-1700hrs
Duration	: **3 hours**

General Instructions

1. Read all instructions carefully before answering the questions.

2. This question paper contains three (3) questions in SECTION A and three (3) questions in SECTION B on five (5) pages.

3. Answer ALL questions in SECTION A. [60 Marks], and answer any TWO questions from SECTION B. [40 Marks]

4. The answer to each question should commence from a new page.

5. Refer to the Annexure of the VHDL syntax given on page five (5) to write VHDL code, if necessary.

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. [60 Marks]

Electronic Dumbwaiter Lift (EDL)

The following description is about the Electonic Dumbwaiter Lift (EDL) used to transport foods and beverages between floors from/to the hotel's default location (kitchen). Your task is to analyse the following specifications and design the EDL controller using digital electronic components and logic gates. Figure Q1 depicts the typical view of the EDL controller.

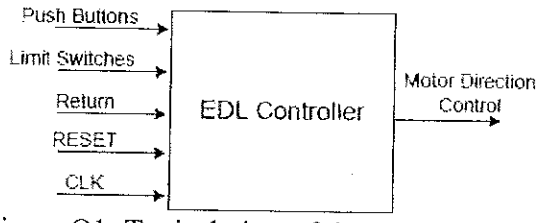


Figure Q1: Typical view of the EDL Controller

Specifications of the EDL Controller

1. The hotel owner initially plans to implement the system for three stories (ground floor, first floor, and second floor).
2. *Push Buttons* are used for selecting the appropriate floor to move the lift from the default location (ground floor kitchen).
3. *Limit Switches* will be used for identifying the floors when the lift reaches.
4. *Return* switches are used for sending the EDL to the default location.
5. *RESET* can be used to reset the EDL system.

[Q1]

(i) Draw a block diagram and identify inputs and outputs of the EDL controller. [5 Marks]

(ii) Draw a flowchart to represent the logical flow of the EDL controller. [10 Marks]

[Q2]

(i) Draw a state diagram for the EDL controller. [10 Marks]

(ii) Design a circuit diagram to perform the EDL controller operation. Clearly explain the working procedure of the EDL, indicating the internal functionality of the controller. You must show clearly how the controlling is made according to the given specifications and show the data paths. [15 Marks]

(iii) Write complete VHDL code for EDL and state the comments where necessory. [10 Marks]

[Q3]

(i) Briefly explain how you perform functional and structural testing with the fault model for EDL. [6 Marks]

(ii) List commonly used algorithms for testing digital circuits. [4 Marks]

SECTION B: Answer any TWO questions. [40 Marks]

[Q4] The following Figure Q4 depicts a waveform that has been reported during an unknown digital electronic system experiment.

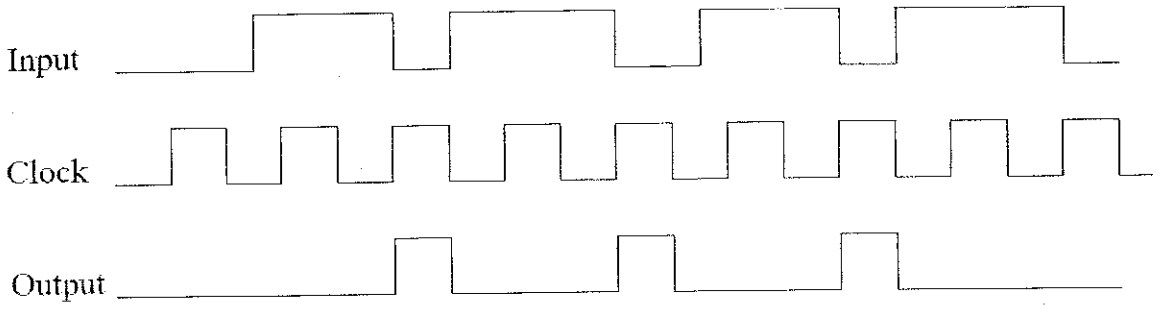


Figure Q4: Waveform of the unknown digital electronic system.

(i) Since you do not have access to the inside components of the circuits, what steps do you need to carry out to find a replacement circuit when considering a product upgrade? [4 Marks]

(ii) Draw a state diagram for the system and hence draw an ASM chart. [4 Marks]

(iii) Design a replacement circuit for the unknown digital electronic system using the digital electronic components such as logic gates, flip-flops, etc. [6 Marks]

(iv) Write the VHDL entity declaration for your circuit. [6 Marks]

[Q5] Figure Q5 depicts a digital circuit for path sensitised tests.

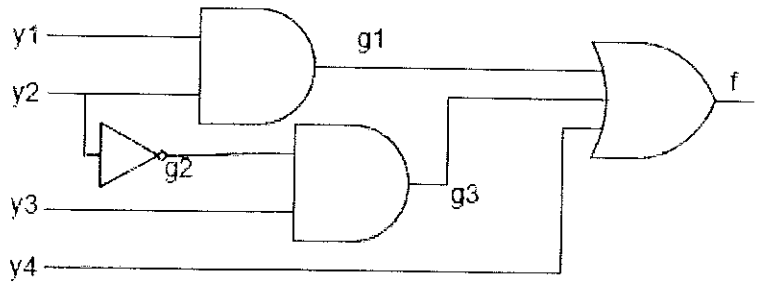


Figure Q5: A Digital Circuit

(i) Briefly Explain sensitisation and the need for that in IC design and test.

[4 Marks]

(ii) Find the input that needs to be applied on **y2**, **y3** and **y4** to sensitise the path **y1-g1-f**. (Draw a table and clearly show the path, sensitised apply, test, and fault detected)

[10 Marks]

(iii) Identify the complete test sets TS = {}

[6 Marks]

[Q6] Figure Q6 depicts a circuit that has inputs (X, CLK) and output ($\bar{Q}_A$) as below.

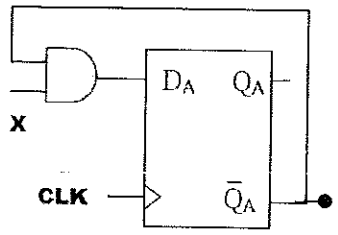


Figure Q6: A Circuit Diagram

(i) Draw a timing diagram for the circuit showing X, D_A, $\bar{Q}_A$ and CLK. Assume that X is initially 1 and after X becomes 0 for 10 ns, then X becomes 1 for 200 ns. Assume that the clock frequency (f_{CLK}) is 10MHz.

[08 Marks]

(ii) Modify the timing diagram drawn in above Q6 (a) for the following parameters

AND Gate propagation delay is 10ns and D flip-flop has HIGH to LOW propagation delay (t_{PHL}) = 80ns, LOW to HIGH propagation delay (t_{PLH}) = 60ns, set-up time (t_{SU}) = 20ns, hold time (t_H) = 0ns, maximum clock frequency (f_{CLKMAX}) = 20MHz.

[12 Marks]

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;
```