Contents

[Chapter 1 – Introduction [5%] 1](#_bookmark0)

[Chapter 2 – Design and Development [30%] 2](#_bookmark1)

[Section 2-1: Clock Divider Module [5%] 2](#_bookmark2)

[Section 2-2: Dice Logic Module [7%] 3](#_bookmark3)

[Section 2-3: Binary to Seven Segment Encoder [5%] 4](#_bookmark4)

[Section 2-4: Control modules (Function Selection and Display Control) [6%] 5](#_bookmark5)

[Section 2-5: My Dice Module (Top-level design) [7%] 6](#_bookmark6)

[Chapter 3 – Validation [35%] 7](#_bookmark7)

[Section 3-1: Submodules Validation [20%] 7](#_bookmark8)

[Section 3-2: MyDice (Top-level module) Validation [15%] 8](#_bookmark9)

[Chapter 4 – Conclusions and Critical Reflections [30%] 9](#_bookmark10)

[Appendix 10](#_bookmark11)

# Chapter 1 – Introduction [5%]

#### Around 200 words; Calibri; Font size 11; Line spacing 1.08.

In your own words, give an introduction covering the following:

* Briefly introduce FPGA in a couple of sentences;
* Give a brief introduction about the project, i.e the main aim and objectives.

# Chapter 2 – Design and Development [30%]

## Section 2-1: Clock Divider Module [5%]

#### Around 200 words; Calibri; Font size 11; Line spacing 1.08.

In your own words, explain the following:

* Your idea on how to achieve a 100 Hz signal from 50 MHz signals;
* How you implement your clock divider module in Verilog;

#### A Attach the RTL viewer diagram in this section to assist your discussion;

#### Your justification of using structural logic and behavioural logic.

If you have submodules for the clock divider, please also explain them here;

**Tips:** You can use diagrams/flow charts/tables to illustrate your design.

## Section 2-2: Dice Logic Module [7%]

#### Around 400 words; Calibri; Font size 11; Line spacing 1.08.

In your own words, explain the following:

* Briefly explain the function of the Dice Logic module;
* How you implement your Dice Logic module in Verilog;

#### Attach the RTL viewer diagram in this section to assist your discussion;

#### Your justification for using structural logic and behavioural logic.

If you have submodules for the dice logic module, please also explain them here.

**Tips:** You can use diagrams/flow charts/tables to illustrate your design. Enrich the logic in your discussion using different paragraphs or bullet points.

## Section 2-3: Binary to Seven Segment Encoder [5%]

#### Around 200 words; Calibri; Font size 11; Line spacing 1.08.

In your own words, explain the following:

* Briefly explain the function of the Binary to Seven Segment Encoder module;
* How you implement your Encoder module in Verilog;

#### Attach the RTL viewer diagram in this section to assist your discussion;

#### Your justification for using structural logic and behavioural logic.

If you have submodules for the encoder, please also explain them here.

**Tips:** You can use diagrams/flow charts/tables to illustrate your design.

**Note:** Please don’t worry if you don’t know how the binary to the BCD encoder I gave your works. However, a bonus will be considered if you give additional explanations.

## Section 2-4: Control modules (Function Selection and Display Control) [6%]

#### Ar Around 300 words; Calibri; Font size 11; Line spacing 1.08.

In your own words, explain the following:

* Briefly explain the function of these control modules;
* How you implement these modules in Verilog;

#### Attach the RTL viewer diagram in this section to assist your discussion;

#### Your justification for using structural logic and behavioural logic.

**Tips:** You can use diagrams/flow charts/tables to illustrate your design.

## Section 2-5: MyDice Module (Top-level design) [7%]

#### Around 400 words; Calibri; Font size 11; Line spacing 1.08.

In your own words, explain the following:

* Briefly explain the main design structure of your implementation of MyDice;

#### How the signal passed around during different input operations;

#### Attach the RTL viewer diagram in this section to assist your discussion;

#### Your justification for using structural logic and behavioural logic.

**Tips:** You can use diagrams/flow charts/tables to illustrate your design. Enrich the logic in your discussion using different paragraphs or bullet points.

# Chapter 3 – Validation [35%]

## Section 3-1: Submodules Validation [20%]

#### As much as you need; Calibri; Font size 11; Line spacing 1.08.

In your own words, explain the following:

* Briefly explain the test process for **ALL** of the sub-modules;
* Discuss the key testing steps;

#### Provide evidence of your validation results;

#### Include images from the ModelSim simulation to assist your discussion.

### If you have additional modules created, please also include them here!

**Tips:** Construct your discussion based on the ModelSim simulation result image and mark/label the image to echo your discussion.

## Section 3-2: MyDice (Top-level module) Validation [15%]

#### As much as you need; Calibri; Font size 11; Line spacing 1.08.

In your own words, explain the following:

#### Code your testbench based on the test process/settings outlined below;

#### Provide evidence of your validation results;

#### Include images from the ModelSim simulation to assist your discussion.

**Tips:** you can tweak your original code slightly to make the validation more effective. Test procedure:

* Set all the pins to default states:
  + ***Set:*** Low;
  + ***Sel1***: Low;
  + ***Sel2:*** Low;
  + ***Roll\_n:*** High;
* Set MyDice to the **“setting mode”** (i.e. ***Set*** High);
  + Test all combinations of the ***Sel1*** and ***Sel2*;**
  + Wait and check the MyDice output behaviour in each combination;
  + Trigger ***roll\_n*** (i.e. Low);
  + Wait and check the MyDice output behaviour;
  + Reset ***roll\_n*** back to the default state (i.e. High);
  + Set ***Sel1*** and ***Sel2*** to select ***d6*** mode (i.e. Low Low combination);
* Set MyDice to the **“playing mode”** (i.e. ***Set*** Low);
  + Check the “00” display before triggering ***roll\_n***;
  + Trigger ***roll\_n*** (i.e. Low);
  + Wait enough time to validate the MyDice’s counting behaviour;
  + Reset ***roll\_n*** back to the default state (i.e. High);
  + Wait and Check the MyDice output behaviour;
  + Ttrigger ***roll\_n*** (i.e. Low) once again;
  + Wait and check the MyDice output behaviour;
  + Reset ***roll\_n*** back to the default state (i.e. High);
* Finish the validation.

(Optional*-Try these to challenge yourself*) Additional test settings:

* Add two additional test outputs to translate **SevenSegMSB**/**SevenSegLSB** in binary;
* Display all the binary values in **unsigned binary format**;

# Chapter 4 – Conclusions and Critical Reflections [30%]

#### As much as you need; Calibri; Font size 11; Line spacing 1.08.

In your own words, try to answer the following questions

* What part of the project are you most proud of in your design? For example, achieving additional functions or parts you think your design works most successfully.
* Which part do you think is most challenging and how do you deal with this?
* If you have more time, what function do you want to develop? And how long do you think you need to finish them?

**Tips:** Try to give **specific examples** in this session rather than a vague discussion. Also, break your answer into different paragraphs.

# Appendix

Use this session to add additional plots (with some captions) as a show-off for anything you are proud of in this project.

#### Please also attach ALL your Quartus codes here (in pure text format, not image!).