

**School of Science and Technology**

Embedded Systems ITEC40091

# Design of asynchronous counter using FPGA board.

By

Salim Sulaiman Maaji

N0726366

MRes Electronic Systems

**TUTOR(S) :** Pedro Machado

**Date:** 05/31/201

# Declaration

“I am aware of the University’s rules on plagiarism and collusion and I understand that, if I am found to have broken these rules, it will be treated as Academic Misconduct and dealt with accordingly. I understand that if I lend this piece of work to another student and they copy all or part of it, either with or without my knowledge or permission, I shall be guilty of collusion.

• In submitting this work I confirm that I am aware of, and am abiding by, the University’s expectations for proof-reading.

• I understand that I must submit this coursework by the time and date published. I also understand that if this coursework is submitted late it will, if submitted within 5 working days of the deadline date and time, be given a pass mark as a maximum mark. If received more than 5 working days after the deadline date and time, it will receive a mark of 0%. For referred or repeat coursework, I understand that if the coursework is not submitted by the published date and time, a mark of 0% will be automatically awarded.

• I understand that it is entirely my responsibility to ensure that I hand in my full and complete coursework and that any missing pages handed in after the deadline will be disregarded.

• I understand that the above rules apply even in the eventuality of computer or other information technology failures.”

# Abstract

Field Programmable Gate Array (FPGA) is an efficient reconfigurable integrated circuit platform and has become a core signal processing microchip device of digital systems over the last decade. With the rapid development of semiconductor technology, the performance and system integration of FPGA devices have been significantly progressed.

Contents

[Design of asynchronous counter using FPGA board. 1](#_Toc515492052)

[Declaration 2](#_Toc515492053)

[Abstract 3](#_Toc515492054)

[Introduction 4](#_Toc515492055)

[Problem statement 5](#_Toc515492056)

[Aims 5](#_Toc515492057)

[Objectives 5](#_Toc515492058)

[Components 5](#_Toc515492059)

[Literature 5](#_Toc515492060)

[Design methodology 6](#_Toc515492061)

[Integrated circuit design 6](#_Toc515492062)

[Behavioural design 10](#_Toc515492063)

[**Stage three** 11](#_Toc515492064)

[Results and discussion 11](#_Toc515492065)

[ModelSIm outputs 12](#_Toc515492066)

[Conclusion 14](#_Toc515492067)

[Future work 15](#_Toc515492068)

[Reference 15](#_Toc515492069)

# Introduction

The exponential growth in data usage and technological advancement in many fields of engineering, science and medicine has exceeded the pace of Moore’s law over the past 10 years. Companies are now searching for a better way possible to deliver to the market, energy efficient, cheaper and reliable components. A field-programmable gate array (FPGA) is a semiconductor device reprogrammed to perform a specific design application. Its application is evolving rapidly in the field of engineering, technology research and in peripherals market (Miller, 1994). FPGAs are utilized as hardware coprocessor for low-power and high performance heterogeneous computing system. (Gheolbănoiu, Petrică and Coţofană, 2015). One of the vital part of FPGA is up/down counter, and it has many applications especially in electronic voting system clocks, elevators and microwaves.

The main objective of this project is to design an up/down counter using Verilog programming. The output of the counter will be displayed on Seven Segment Display available on Altera DE1 SoC. The remainder of this report is organised as follows: Section II presents a literature review on FPGA and up/down counter design. Section III discusses the experimental methodology. Results and discussion are presented in Section IV. Concluding remarks are given in Section V.

## Problem statement

## Aims

The main aim of this project is to design an up/down counter using the Altera DE1-SoC.

## Objectives

* To carry literature on FGA design
* To be familiarized with Verilog cord
* To design frequency divider
* To design a seven segment display
* And to simulate the up/down counter circuit using Altera DE1-SoC board

## Components

* Quartus II design software
* Altera DE1-SoC development kit
* ModelSim

### Literature

FPGA is an emerging technology for system architects and board designers, it allows them to trade off costs, development time, performance and supportability without the NRE costs and fabrication time of a custom digital ASI. This programmable ICs has a wide range of potential applications including communications infrastructure, sensitive database access, industrial control, and high-performance signal processing. It contains millions of gates of logic, intensive memory and high-speed transceivers and it is widely being used for research and in an industry (Trimberger and Moore, 2014). One of the FPGA hardware design platform is DE1-SoC, the development kit provides ultimate design flexibility with a high-performance and low-power processor system.(Terasic Inc, 2016).



Figure 1DE1-SoC development board (top view) (Terasic Inc, 2016)

# Design methodology

## Integrated circuit design

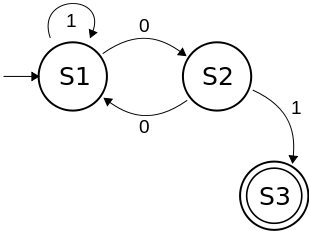
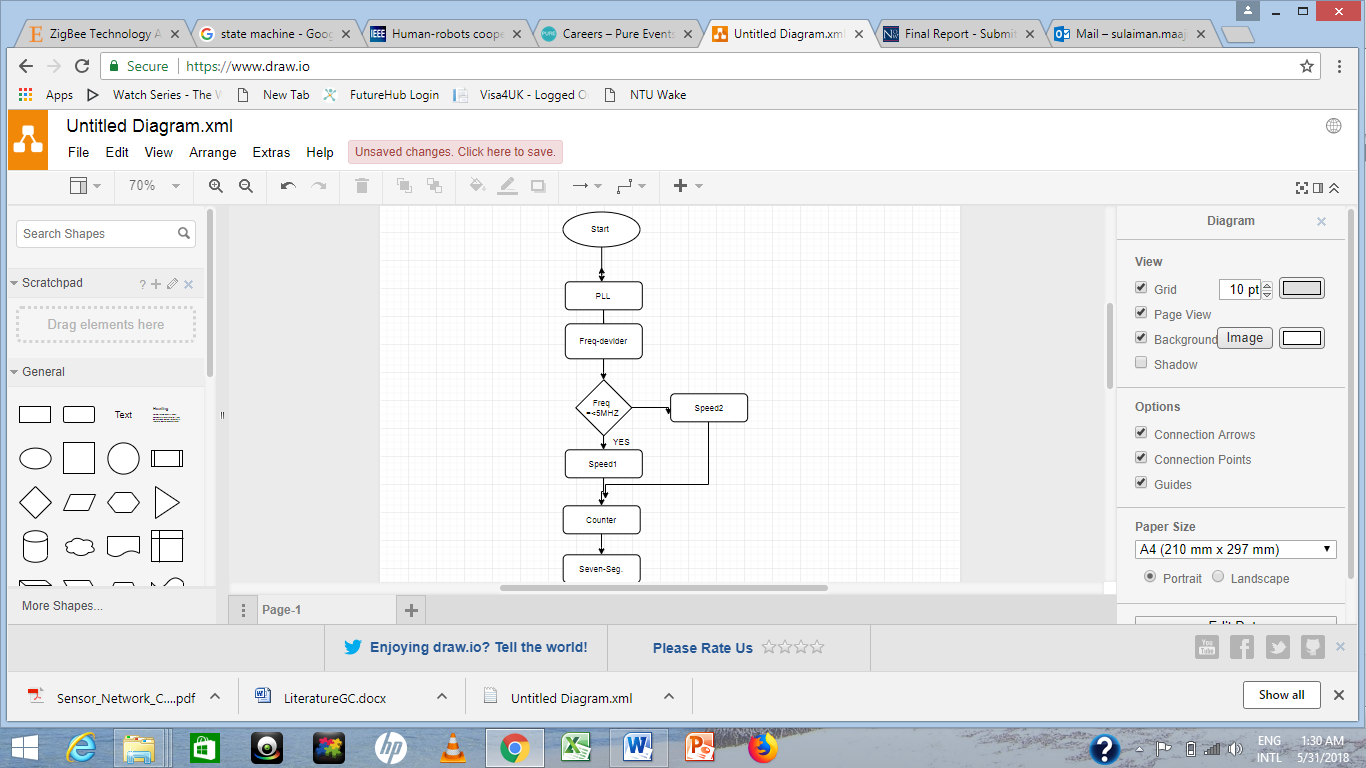


Figure 2 Flowchart Figure 3 Switch state machine

The design for the integrated circuit involves four main components:

**Phase-Locked Loop (PLL):**  is a feedback circuit that automatically adjusts the phase of a locally generated signal to match the phase of an input signal. This component lock an output signal’s phase relative to an input reference signal’s phase (Corporation, 2018).

The output signals may be any periodic waveform, but are typically sinusoids or digital clocks. The circuit is widely used for electronic design, control and measurement applications (Corporation, 2018). In this project the inputs for the PLL are 50 MHZ clock and VCC which is connected to the KEY3 on the main Altera board. The output of the PLL is periodic waveform, typically sinusoidal and is fed to the input of the clock divider as shown in fig.. below. One of the function of this connection is to make sure that, the entire system is asynchronous and to also make sure all the components are synchronised to the same clock

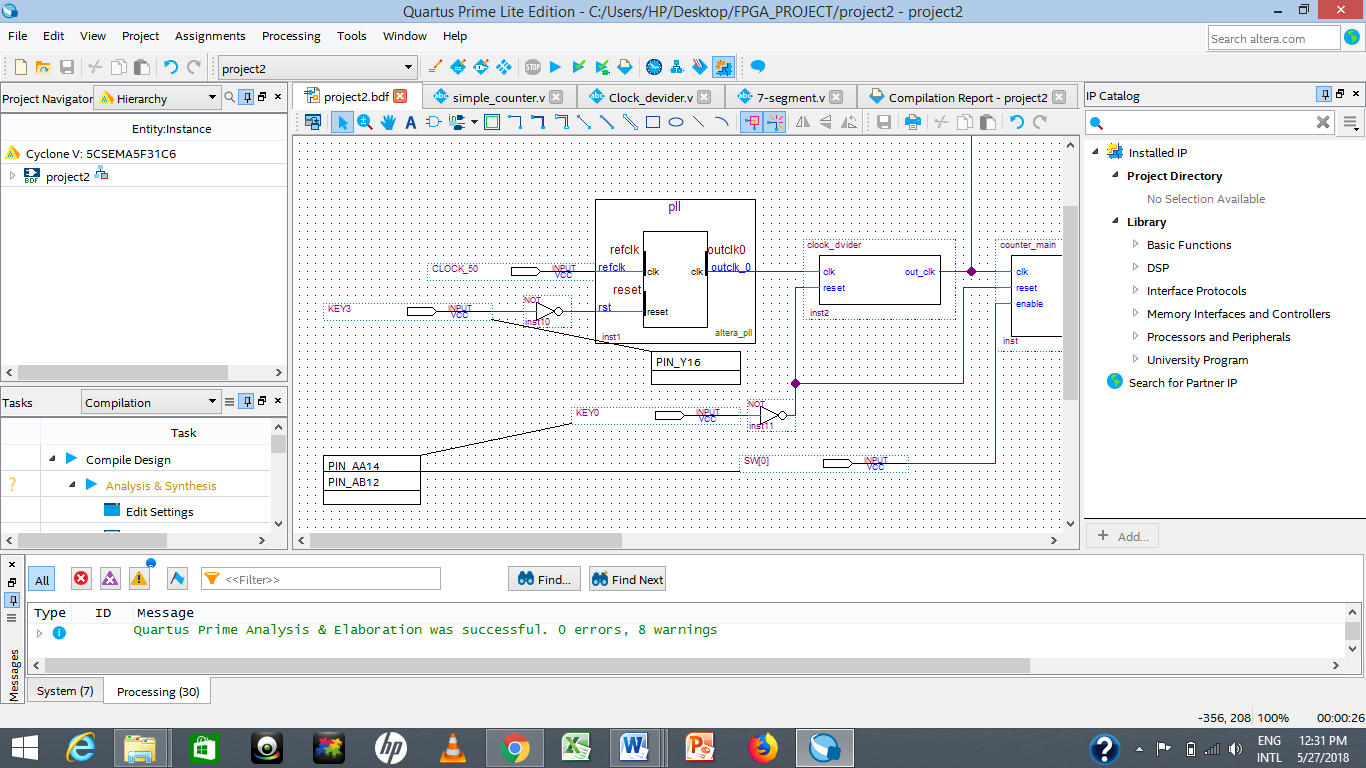


Figure 4Cross section of the top-level-design

**Clock divider:** also known as frequency divider provides an output clock signal that is a divided frequency of the input. This component plays an important role in digital circuit design. The inputs for the clock divider are reset which is connected to the KEY0 on the main Altera board and the clk which is an output of PLL ( 50 MHZ clock) fed to the clock divider as explained in the previous section. The circuit connection is shown below



Figure 5 Cross section of the circuit

The 50 MHz output of the PLL which is a square wave with a period of 20 ns is impossible for human eye to visualised, hence it requires a division as shown in the figure below.

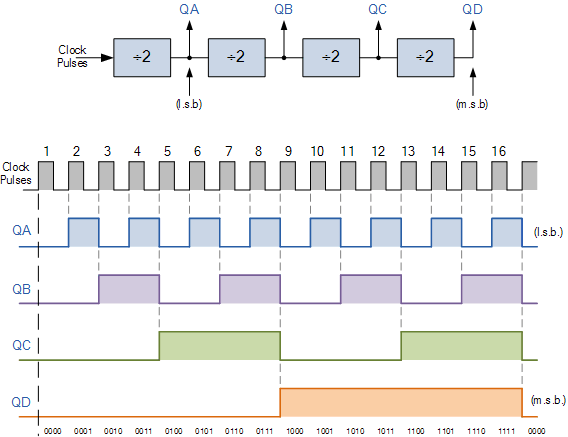


Figure 6 output for the clock divider (Corporation, 2018).

To make the hexadecimal display visible, a sequence of frequencies were generated from the fundamental frequency. A parameter named speed is created and assigned with a frequency value to be (50MHZ), the value of the speed is then divided accordingly to generate a desire frequency as shown in the table below.

Table 1 Complete cycle for the frequency divider.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **PARAMETER** | **VALUE** | **EVENT** |
|  | **Speed 1** | 0HZ – 25MHZ | Rising edge |
|  | 25MHZ – 50MHZ | Falling edge |
|  | **Speed 2** | 12.5MHZ – 25MHZ | Rising edge |
|  | 25MHZ - 37.5MHZ | Falling edge |
|  | 37.5MHZ -50MHZ | Rising edge |

These frequency values a then fed to the input of the binary counter.

**Binary counter:**

**Hex LED display:** hex LED displays are commonly used as alphanumeric displays by didgital engineers. This component can be used to show show any hex number between 0000 and 1111 by illuminating combinations the LEDs as sown below.

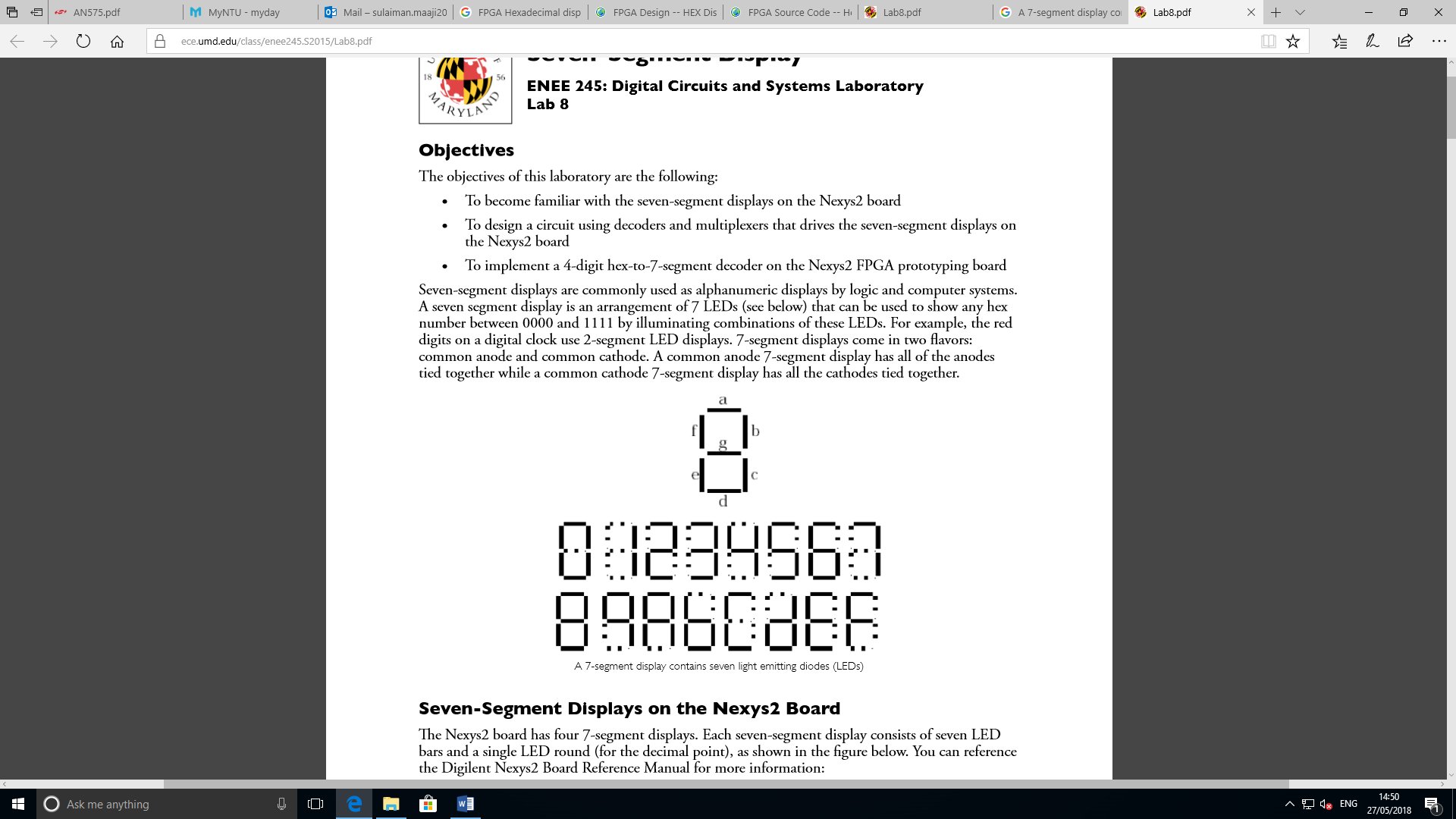


Figure 7Seven-Segment counting mode

(Lab, 2018)

Table 24-digit seven-segment

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Segments** | **a** | **b** | **c** | **d** | **e** | **f** | **g** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
| **1** | **1** | **0** | **0** | **1** | **1** | **1** | **1** |
| **2** | **0** | **0** | **1** | **0** | **0** | **1** | **0** |
| **3** | **0** | **0** | **0** | **0** | **1** | **1** | **0** |
| **4** | **1** | **0** | **0** | **1** | **1** | **0** | **0** |
| **5** | **0** | **1** | **0** | **0** | **1** | **0** | **0** |
| **6** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| **7** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |
| **8** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **9** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **A** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **B** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| **C** | **0** | **1** | **1** | **0** | **0** | **0** | **1** |
| **D** | **1** | **0** | **0** | **0** | **0** | **1** | **0** |
| **E** | **0** | **1** | **1** | **0** | **0** | **0** | **0** |
| **F** | **0** | **1** | **1** | **1** | **0** | **0** | **0** |

**Note: 1 = off and 0 = on**

The 4-digit seven-segment controller will take a clock and four characters (4-bit each) as an inputs from counter , and write the seven-segment control signals as an output

## Behavioural design

The behavioural design for the hexadecimal counter involves three stage:

**Stage one**

Creating a project in Quatus prime software and also top-level design, the software will automatically generate needed files for the project. Three files were created in preparation for Verilog programing. Simple\_counter.v where Verilog codes were generated for the project counter, cock\_divider.v a Verilog codes for the frequency divider and 7-segment.v for the seven segment display. The codes for these files are available at GITHUB.

**Stage two**

To create the top-level design, the Verilog files generated need to be converted by creating symbol file for current file, this process will convert the code files to blocks, symbol files for circuit connection. One of the most important part of this stage is addition of **PLL megafunction** to the project. PLL uses the on-board oscillator (DE1-SoC Board 50MHZ) to create constant clock frequency as the input to the counter. This process involves using IP variation file to open a MegaCore wizard in which a frequency of the clock will be set to 50 MHz. The operation mode is set to normal and the desire frequency to 5.0. All other components for the top level design are added, updated and connected. The integrated circuit designed is shown in figure below.

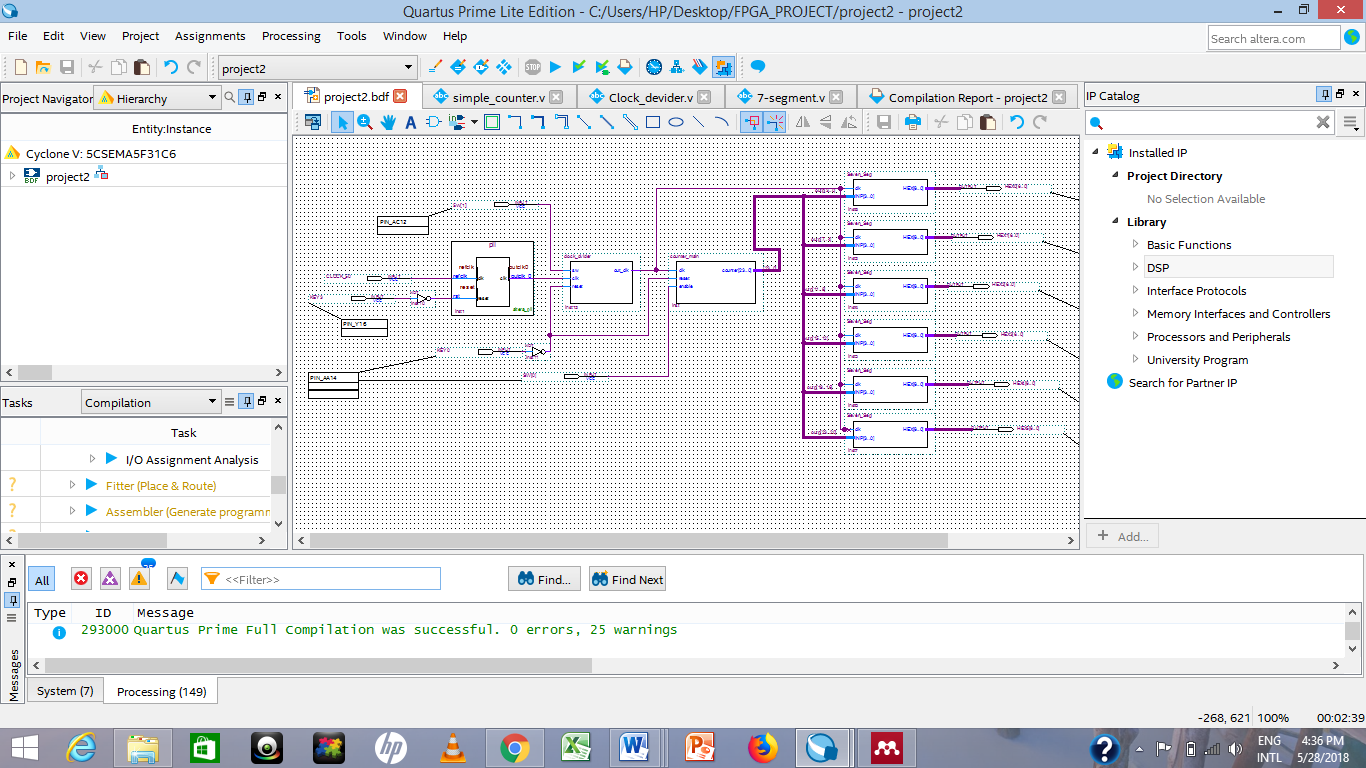


Figure 8Inegrated circuit design

## **Stage three**

To program the FPGA board, the entire project files need to be analysed and elaborated in preparation for the pin assignments too correlate with the actual pins on the board. The pin planner spread sheet for the specific pin assignment is provided in the user manual. Pins are assigned in this project accordingly, the cross-section for the pin planner is shown below

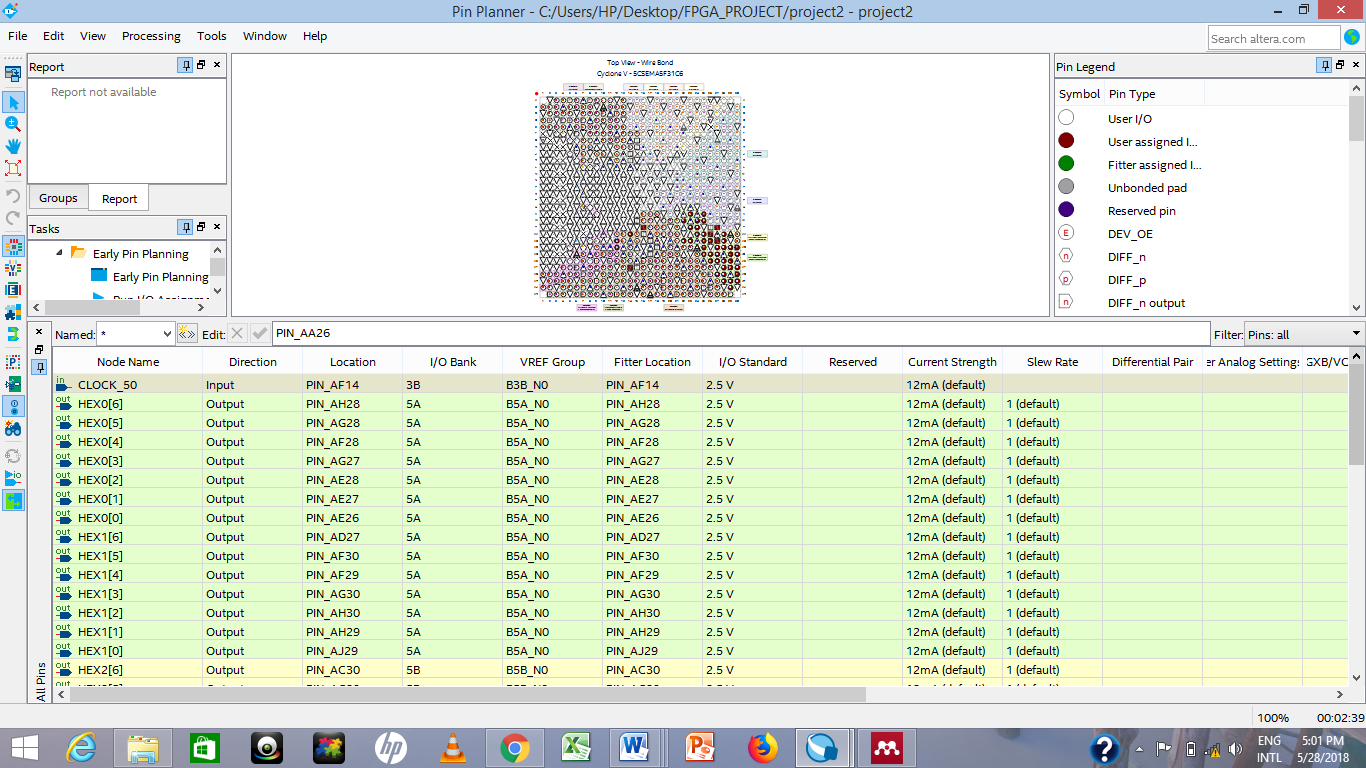


Figure 9 Pin assignment

On the figure above, column one is the pin name and column three represents the FPGA pin location.

The next step in the FPGA programing is creating a Default TimeQuest SDC file, this file is essential for the successful implementation of the design as the Quartus Prime TimeQuest Timing Analyser will be using during design compilation, for details referrer to the user manual (..2016). The next step is design is compilation in which the design is compiled into bitstream to be downloaded into FPGA.

# Results and discussion

From the result below it can be seen that the output for the seven-segment is counting as expected. The counting starts from 0 to 9, and continue counting from A to F, when the counting for one segment reached fff. The next segment starts from 1 to ffff as indicated below.

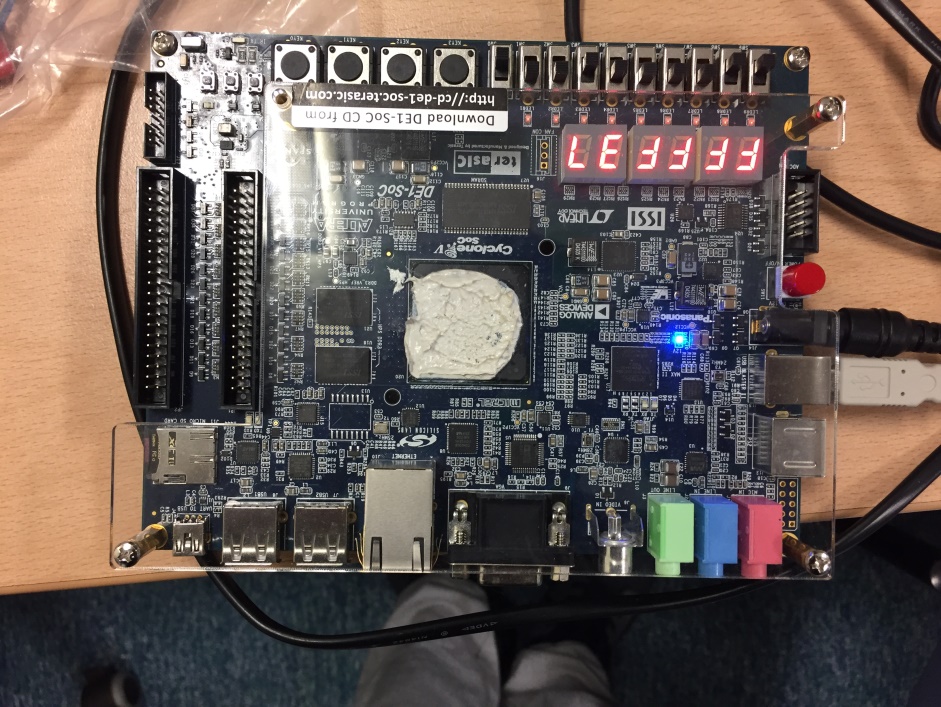


Figure 10 Seven =segment display on Altera board

One of the nobility for this design is that the system can be reset to zero by pressing key 0, stay on hold by pressing key 3 and the speed can be changed by moving sw1up or down.

## ModelSIm outputs

Simulation for the Modelsim is also performed to verify that the design operates correctly. The behaviour of the circuit is tested by assigning the input with a sequence of values as detail below in the figures below:

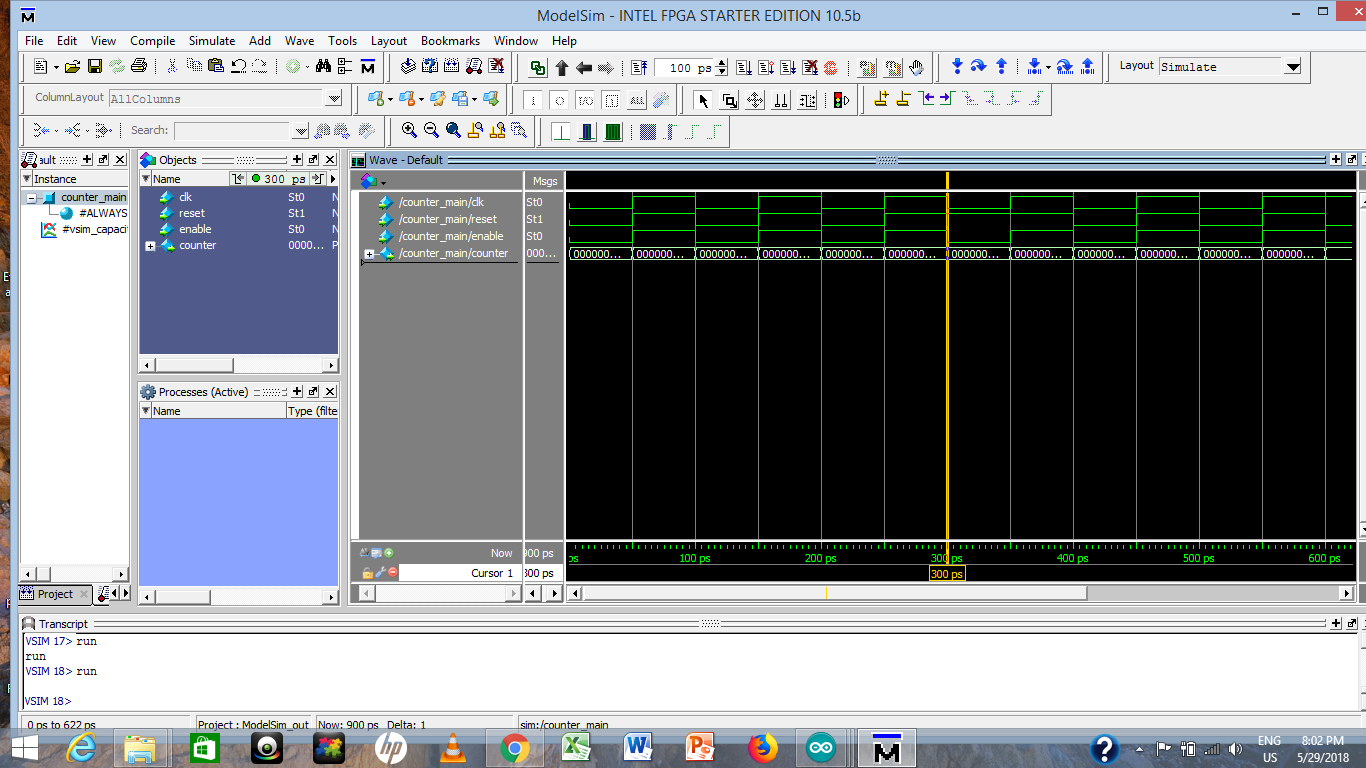


Figure 11 Output when clock is set to zero

As indicated above, from 0 to 300ps clock is set to 0, reset is =0. From 300ps to 600 reset is set to zero

**Clock divider**

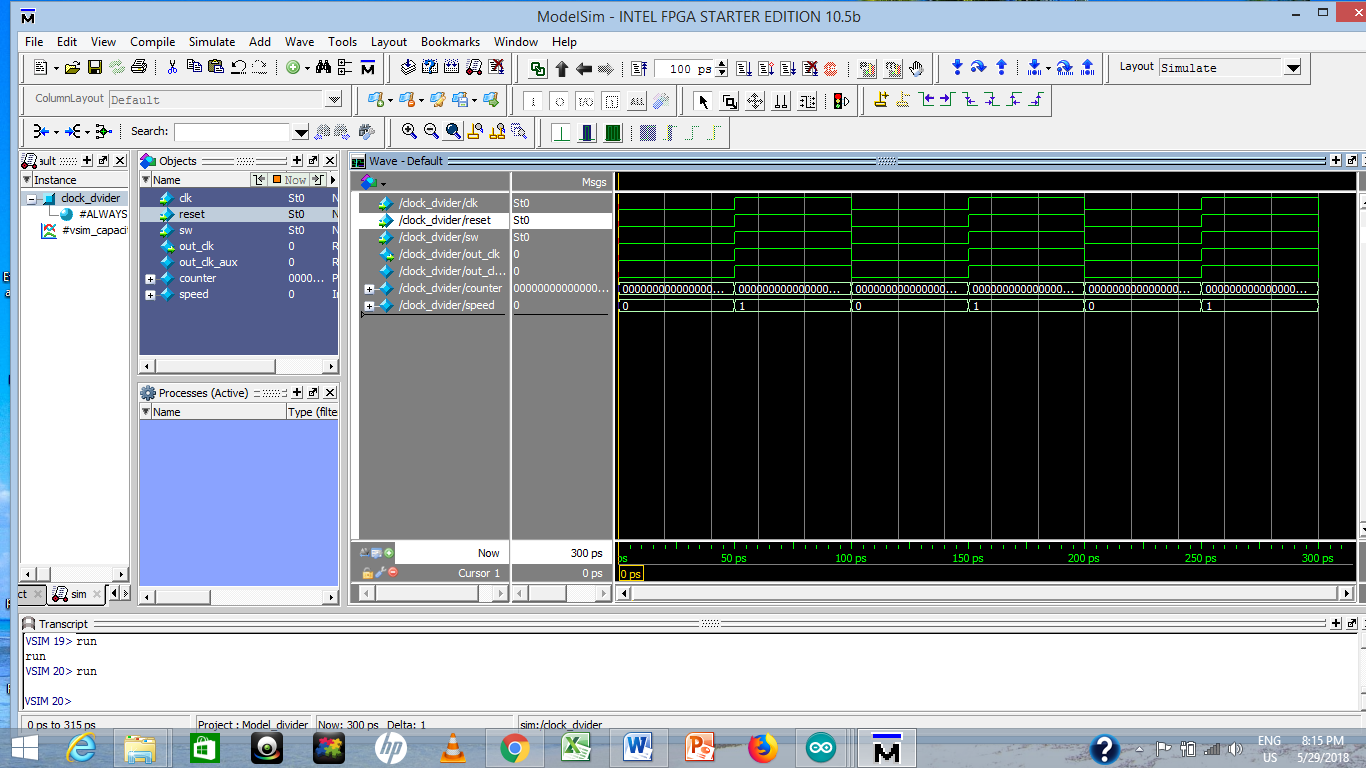


Figure 12output for the clock divider

As indicated above Clock = 0, sw = 0, reset = 0

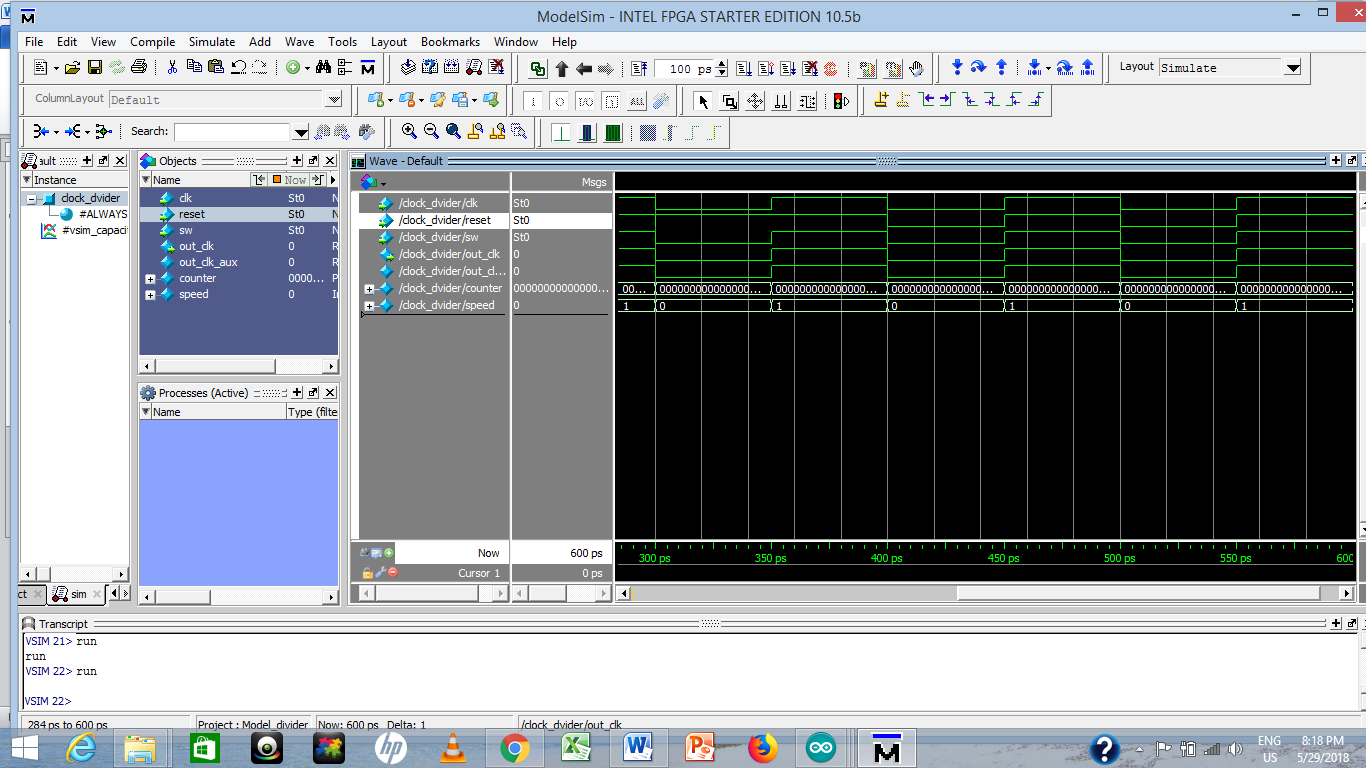


Figure 13Output when reset is 1

Clock = 0, sw = 0, reset = 1

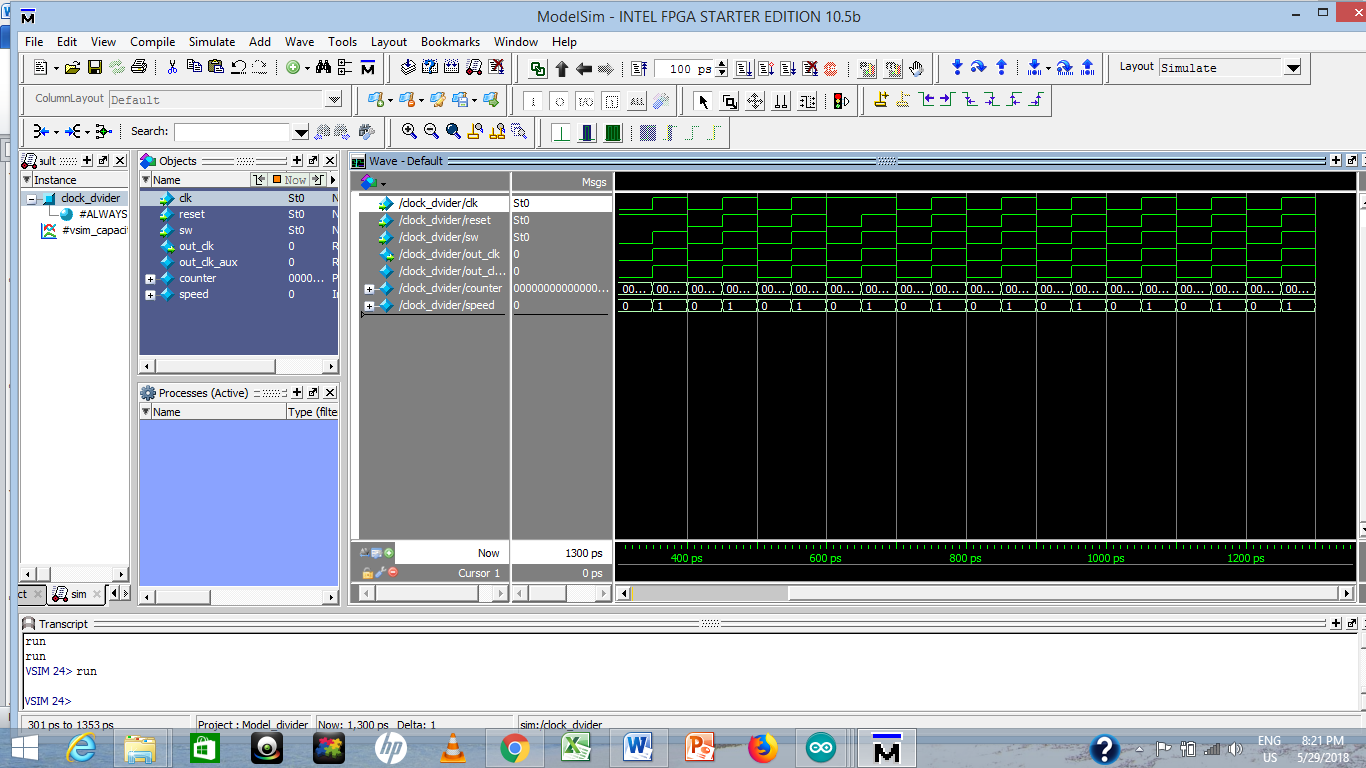


Figure 14 clock is forced to be 1

Clock = 1, sw = 0, reset = 1



Figure 15 Reset is set to 1

Clock = 1, sw = 1, reset = 1

# Conclusion

In conclusion an up/down counter is design using Verilog programming language. The output of the counter is displayed on Seven Segment Display available on Altera DE1 SoC. It is leant that the inputs for the PLL are 50 MHZ clock and VCC which is connected to the KEY3 on the main Altera board. The output of the PLL is periodic waveform, typically sinusoidal and is fed to the input of the clock divider. To make the hexadecimal display visible, a sequence of frequencies were generated from the fundamental frequency. A parameter named speed is created and assigned with a frequency value to be (50MHZ), the value of the speed is then divided accordingly to generate a desire frequency.

# Future work

The future work would involve design design an up/down counter using Verilog programming. The output of the counter will be displayed on Seven Segment Display available on Altera DE1 SoC.g a counter that counts from A to Z and be able to writes English words.

# Reference

Corporation, I. (2018) *Phase-Locked Loop Basics, PLL*. Available at: https://www.altera.com/support/support-resources/operation-and-testing/pll-and-clock-management/pll-basics.html.

Gheolbănoiu, A., Petrică, L. and Coţofană, S. (2015) ‘Hybrid adaptive clock management for FPGA processor acceleration’, in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1359–1364. doi: 10.7873/DATE.2015.0853.

Lab, D. C. & S. (2018) *Seven-Segment Display*. Available at: https://www.ece.umd.edu/class/enee245.S2015/Lab8.pdf.

Miller, W. (1994) ‘Real world applications for field programmable gate array devices-an overview’, in *WESCON/94. Idea/Microelectronics. Conference Record*, pp. 548–551. doi: 10.1109/WESCON.1994.403539.

Terasic Inc (2016) *DE1-SoC Development Kit User Manual*.

Trimberger, S. M. and Moore, J. J. (2014) ‘FPGA Security: Motivations, Features, and Applications’, *Proceedings of the IEEE*, 102(8), pp. 1248–1265. doi: 10.1109/JPROC.2014.2331672.