LEFT BLANK INTENTIONALLY

ACKNOWLEDGEMENTS

Throughout the duration of my project I have had to call on the expertise and guidance of various individuals within my faculty and would therefore like to express my deep gratitude for their support.

Firstly, I am indebted to my supervisor Dr Alin Tisan for his support throughout a technically challenging project. His contributions on the project in general and in digital system design put me on the correct path, and his patience kept me driven to the end. I'd like to also express my gratitude to Professor Marcian Cirstea for helping me to choose such a fascinating and exciting project and for sharing with me some of his expertise on the subject matter. I'd like to thank Dr. Mike Hobbs for his initial guidance throughout my Research Methods Module which helped to layout the framework for this report. I'm also very grateful to Dr Jin Zhang for her subject matter expertise and the inspiration she gave me.

I am most grateful.

ABSTRACT

CONTENTS

LIST OF TABLES AND FIGURES

LEFT BLANK INTENTIONALLY

INTRODUCTION

Our society is currently undergoing a faster rate of technological advancement than ever before in human history. At the heart of technology are electronic systems that enable us control, connectivity and mobility, but what drives electronics? Electrical power is one of the most fundamental building blocks of any electronic circuit or system and converting that power from one form to another is a challenging task, that is constantly evolving.

As scientific and engineering endeavours become more ambitious the challenge of powering conversion systems increases. From space systems to defence technologies, electrical power requirements are becoming more exacting, heavily influenced by needs for lower operating costs, safety and efficiency, now being enforced or influenced by stringent industry standards. These standards are in place to ensure that power is efficient and safe for use by consumers ranging from domestic to industrial and defence applications.

Unclean or dirty electricity is one of the biggest causes of concern in electrical power standards. Harmonic, erratic surges, transients and electromagnetic interference can all cause havoc in power networks, resulting in wastage of energy and equipment damage or failure.

The IEC 61000-3-2:2018 standard for example, outlines limits for harmonic current emissions, and the IEEE 519-1992 standard stipulates requirements for harmonic control and outlines targets for Total Demand Distortion (TDD). Other parts of rectifiers also adhere to specific standards such as the IEEE 295-1969 standard for electronics power transformers. In fact, there are standards for just about every individual component and subsystem of a power conversion system such as semiconductor devices, transmission lines and rectifiers. There are also standards that advise on the overall performance, testing or measurements in a power conversion system such as the IEEE 120-1989 standard for tests and electrical measurements in power circuits.

In this project I paid special attention to rectification, the conversion of three-phase alternating current, AC to direct current, DC. I set about to model, simulate and analyse two main methods of doing so, namely uncontrolled and controlled rectification.

Three-phase AC power is the basis for many energy systems either through power generation, transmission or consumption. Devised by Nikola Tesla (1856 – 1943), three-phase AC power has resulted in more efficient and simplified generation and transmission of electrical energy. Whereas, DC power is at the heart of many electrical products, in particular digital systems.

OBJECTIVES

Using various academic and industry resources such as IEEExplore, Xilinx Academy and Mathworks Documentation my objective in this project was to create conceptual designs based on proven theoretical knowledge and then to create models that I could simulate and analyse. The main project stages were;

- Research
- Digital design for Thyristor Gate Control
- Calculations and modelling of controlled and uncontrolled bridge rectifiers
- Simulation and analysis of conceptual designs

Figures 1 and 2 below show the two main design concepts I undertook. Figure 1 shows the uncontrolled rectification of a three-phase AC voltage supply. Figure 2 shows my approach to digitally controlling rectification, using zero-crossing detection and thyristor gate control.

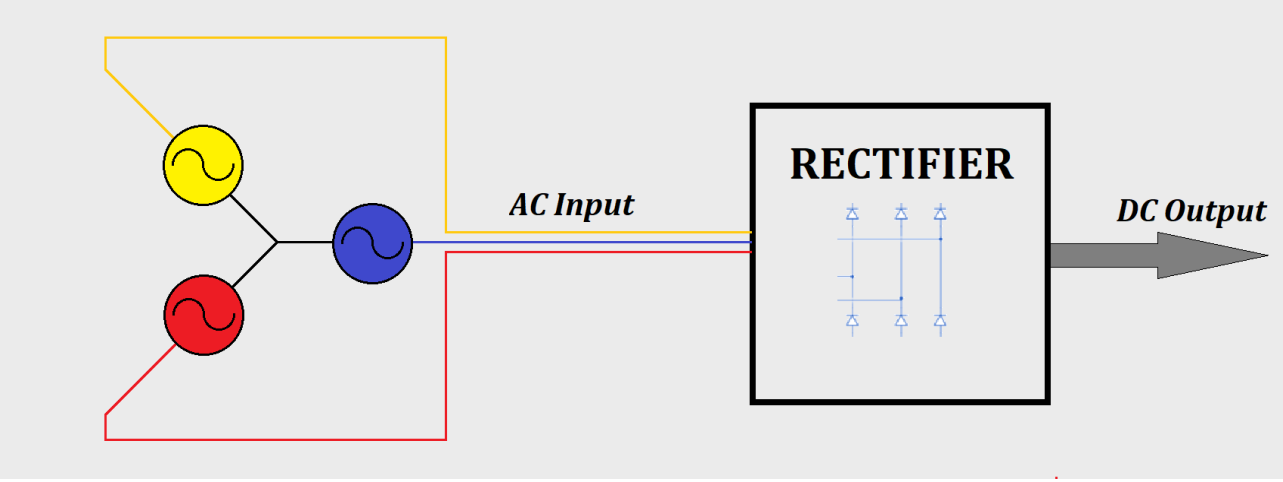


Figure 1. Uncontrolled rectification overview.

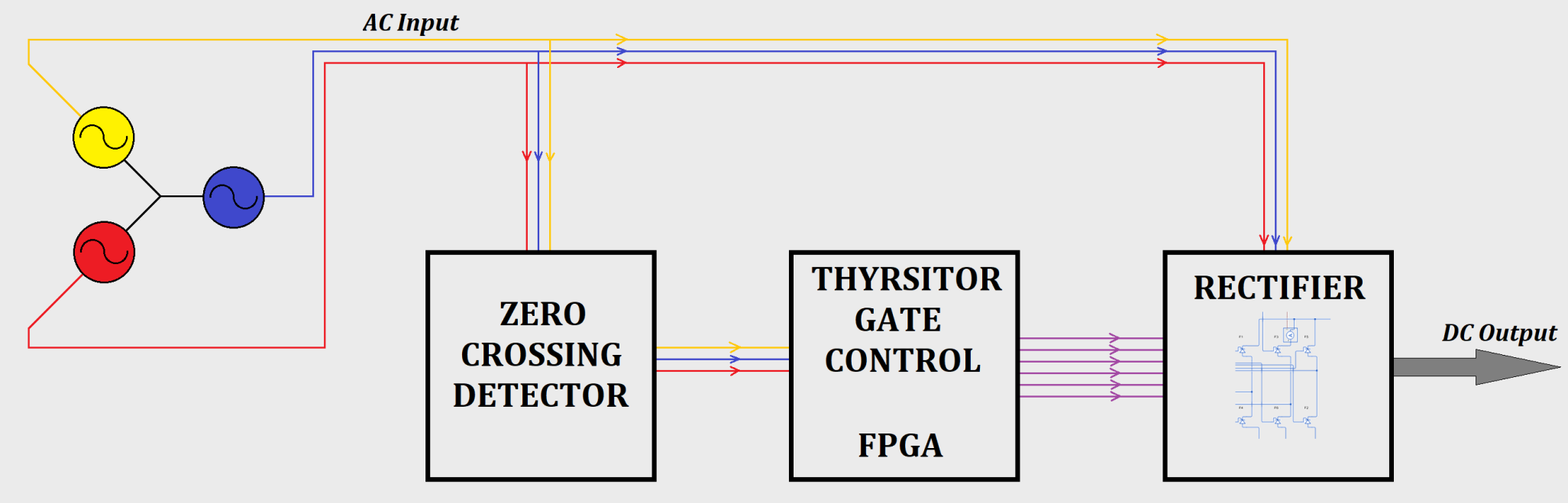


Figure 2. Controlled rectification overview.

PROJECT PLANNING

To facilitate efficient time management, I created a project schedule in the form of the Gantt chart shown. In the chart I highlighted key tasks and milestones, their interdependencies and allowed for some slack in the event of unexpected delays. The outcome of my time management initiatives will be reviewed in the critical analysis section at the end of this report.

CONCEPTUAL DESIGN, MODELLING, SIMULATION & ANALYSIS OF 3-PHASE CONTROLLED & UNCONTROLLED BRIDGE RECTIFIERS
Project Lead: Jacob Greasley

No.	Task Name	Start	Finish	Duration
1	Consultation			4
	Rectification Research			
2	Acquisition of literary sources	02-Jun	04-Jun	3
3	Note taking and paraphrasing	02-Jun	07-Jun	6
	Digital Design Research			
4	Develop Psuedo Code	08-Jun	09-Jun	2
5	Review Study Online Training Modules	08-Jun	13-Jun	6
▪	**PROJECT MILESTONE: Literary Review Completed**	14-Jun		
	VHDL Program Writing			
6	Write Functional VHDL Code	14-Jun	20-Jun	7
7	IO Planning	20-Jun	20-Jun	1
8	Timing Analysis	21-Jun	22-Jun	2
9	Rectifier Calculations	08-Jun	10-Jun	3
	Uncontrolled Rectifier Modelling			
10	Learning simulink modelling, tools, packages, parameters	23-Jun	29-Jun	7
11	Uncontrolled Rectifier Simulation & Analysis	30-Jun	05-Jul	5
12	Controlled Rectifier Modelling	06-Jul	26-Jul	20
▪	**PROJECT MILESTONE: Simulink Modelling Complete**	27-Jul		
	Controlled Rectifier Cosimulation & Analysis			
13	Modelsim setup & licensing	27-Jul	27-Jul	1
14	HDL Verifier add-on setup & licensing	27-Jul	27-Jul	1
15	HDL Verifier Configuration	28-Jul	10-Aug	14
16	Cosimulation and Analysis	11-Aug	24-Aug	14
	Report Writing			0
17	Literary Review	02-Jun	09-Jun	8
18	Methodology	30-Jun	24-Aug	56
19	Referencing	02-Jun	13-Jun	12
20	Others - Intro, Discussion, Critical Analysis, Conclusion	02-Jun	13-Jun	12
21	Final Editing & Structuring	25-Aug	07-Sep	14
▪	**PROJECT MILESTONE: Project Complete**	08-Sep		

Figure 3. Gantt Chart - Initial Project Schedule

BACKGROUND - LITERARY REVIEW

UNCONTROLLED RECTIFICATION

Single Phase Rectification Parameters

This sub-section reviews the **parameters** which I later specified for simulation and analysis of the rectifier circuits. To understand the 3-phase rectification process it is first necessary to look at the single-phase half-wave rectifier as this is the simplest form of rectification. The circuit is supplied by some AC source as shown in the figure below (Rashid, 2011).

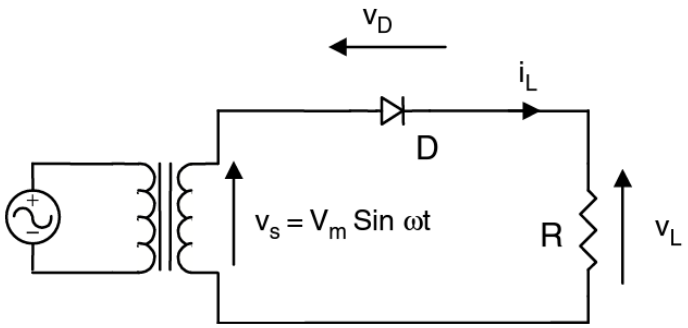


Figure 4. Simple single-phase diode rectifier (Rashid, 2011)

The circuit consists of only one diode, which blocks AC in one direction. The resulting output voltage across the load, R is illustrated in the graph.

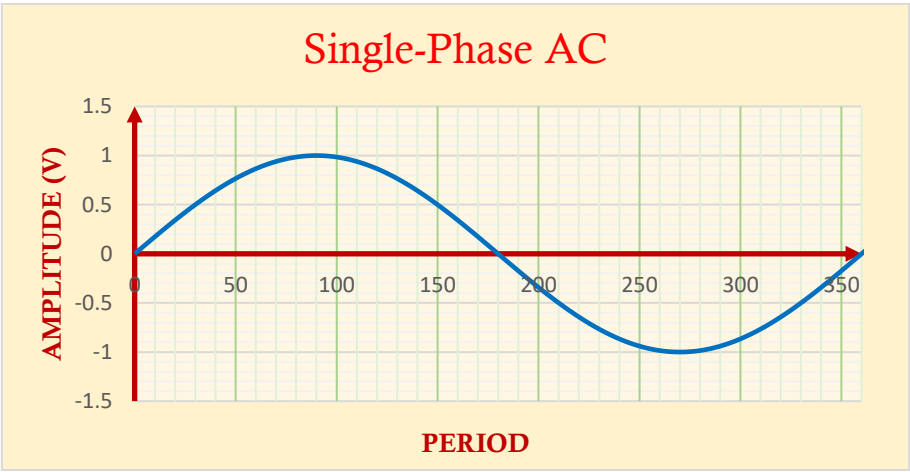


Figure 5. Single-phase AC waveform (Produced in MS Excel)

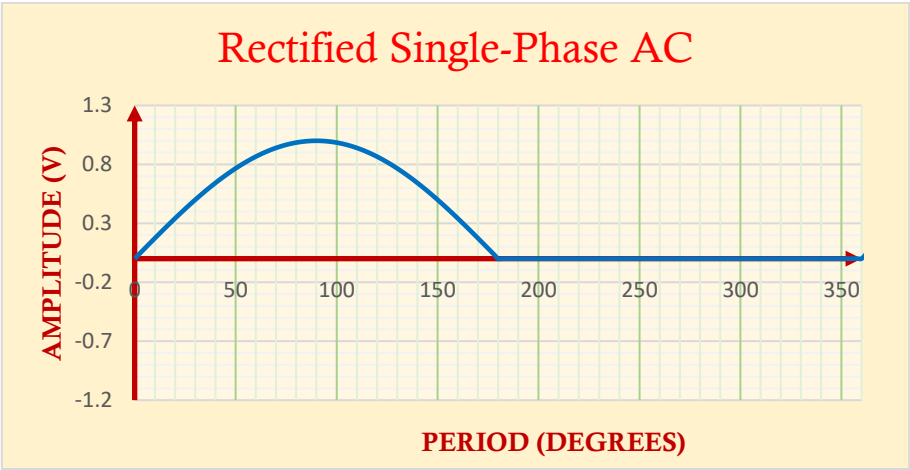


Figure 6. AC waveform after single diode recification

During the negative half-cycle the diode stops conducting.
The for a waveform, V_L the average value through the load, V_{DC} is given by the general equation;

$$V_{DC} = \frac{1}{T}\int_0^T V_L(t).dt$$

Where $V_L = 0 \ for \ \pi < \omega t < 2\pi$

Since $V_L = V_{max} sin(\omega t)$

This can be rewritten as,

$$V_{DC} = \frac{1}{2\pi}\int_0^{\pi} V_{max} sin(\omega t).d(\omega t)$$

$$= \frac{V_{max}}{2\pi}\int_0^{\pi} sin(\omega t).d(\omega t)$$

$$= \frac{V_{max}}{2\pi}\,[-\cos(t)]_0^{\pi}$$

$$V_{DC} = \frac{V_{max}}{2\pi}\,\{[-\cos(\pi)] - [-\cos(0)]\}$$
$$= \frac{V_{max}}{2\pi}\,\{[-(-1)] - [-(1)]\}$$
$$= \frac{V_{max}}{2\pi}\,\{1 + 1\}$$

Average Half-Wave Voltage → $\boldsymbol{V_{DC} = \frac{V_{max}}{\pi} = 0.318V_{max}}$

From the figure below of the full-wave rectifier, the DC voltage level will be double.

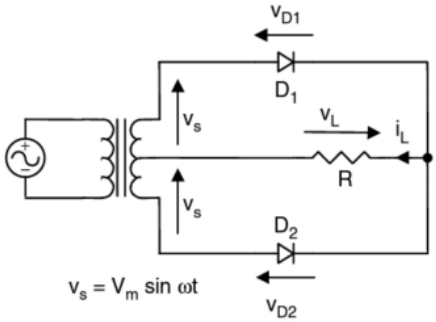


Figure 7. Full-wave diode rectification from centre-tapped transformer rectifier (Rashid, 2011)

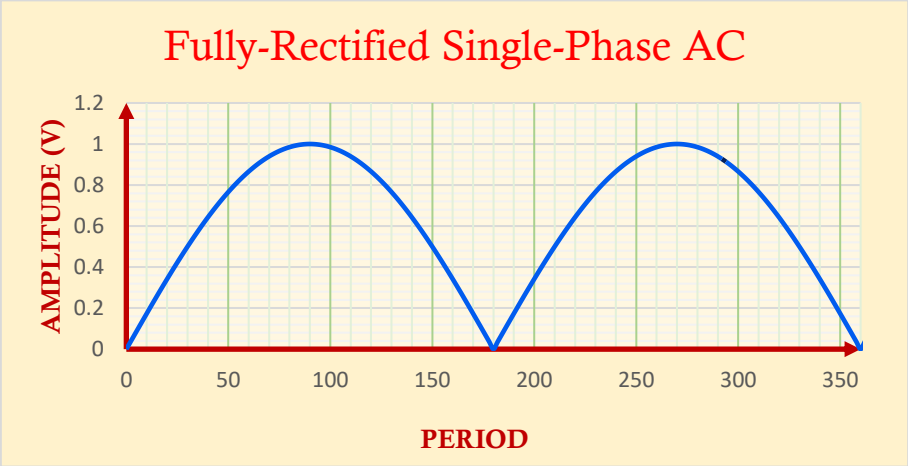


Figure 8. Full-wave rectified AC

Average Full-Wave Voltage → $\boldsymbol{V_{DC} = \frac{V_{max}}{\pi} = 0.636V_{max}}$

The R.M.S. value for the voltage across the load is given by,

$$\boldsymbol{V_{L_{rms}}} = \left[\frac{1}{T}\int_0^T V_L(t).dt\right]^{1/2}$$

As a result, the **RMS half-wave voltage** supplied to the load is given by,

$$\boldsymbol{V_{L_{rms}} = 0.5V_{max}}$$

and the **RMS full-wave voltage** supplied to the load is given by

$$\boldsymbol{V_{L_{rms}} = 0.707V_{max}}$$

Considering Ohm's Law, the relationship between the average current and the voltage across a purely resistive load is found as,

$$I_{dc} = \frac{V_{dc}}{R}$$

and the RMS value of that current,

$$I_{L_{rms}} = \frac{V_{L_{rms}}}{R}$$

Therefore, the **average half-wave rectifier current** the rectifier supplies to the load is

$$I_{dc} = \frac{\boldsymbol{0.318V_{max}}}{R}$$

and the **RMS half-wave rectifier current** is

$$I_{L_{rms}} = \frac{\boldsymbol{0.5V_{max}}}{R}$$

Whereas the **average full-wave rectifier current** produced is

$$I_{dc} = \frac{\boldsymbol{0.636V_{max}}}{R}$$

and the **RMS full-wave rectifier current**

$$I_{L_{rms}} = \frac{\boldsymbol{0.707V_{max}}}{R}$$

The ***rectification ratio,*** σ, is a comparison of the average power transmitted by the rectifier circuit to the RMS power value. It is a useful parameter for assessing the effectiveness of rectification. It's defined as

$$\sigma = \frac{P_{dc}}{P_{L_{rms}}} = \frac{V_{dc}T_{dc}}{V_{L_{rms}}I_{L_{rms}}}$$

For a half-wave uncontrolled rectifier

$$\sigma = \frac{(0.318V_{max})^2}{(0.5V_{max})^2} = 40.5\%$$

and for a full-wave rectifier

$$\sigma = \frac{(0.636V_{max})^2}{(0.707V_{max})^2} = 81\%$$

This clearly shows full-wave rectification to be twice as effective as half-wave rectification.

The ***form factor,*** *FF* shows the relationship between the RMS value (which is heating component) of the voltage or current output of a rectifier, and the average value.

$$FF = \frac{\boldsymbol{V_{RMS}}}{\boldsymbol{V_{DC}}} = \frac{\boldsymbol{I_{RMS}}}{\boldsymbol{I_{DC}}}$$

The **half-wave rectifier FF** is therefore given by,

$$FF = \frac{\boldsymbol{0.5v}}{\boldsymbol{0.318v}} = 1.572$$

The **full-wave rectifier FF** is therefore given by,

$$FF = \frac{\boldsymbol{0.707v}}{\boldsymbol{0.636v}} = 1.112$$

The ***ripple factor,*** *RF* is another useful parameter for assessing the bridge rectifier's performance. It is the ratio of the AC component of the output voltage to the DC and is an indicator of the quality of the output DC (Trzynadlowski, 2016).

$$RF = \frac{V_{AC}}{V_{DC}}$$

$$RF = \frac{\sqrt{V_{RMS}^2 - V_{DC}^2}}{V_{DC}} = \sqrt{\frac{V_{RMS}^2}{V_{DC}^2} - 1} = \sqrt{FF^2 - 1}$$

The smaller the ripple factor, the better the DC voltage. The ripple factor for a three-phase diode bridge rectifier is 0.042. Another way of representing the ripple factor is given by

$$RF = \frac{V_{AC}}{V_{DC}} = \frac{4V_m}{4\omega^2 LC(3\pi\sqrt{2})} = \frac{0.47}{4\omega^2 LC}$$

The ***transformer utilisation factor***, *TUF* shows the ratio between the dc power output of a rectifier and the *apparent power* provided by the source. Here, the source is the secondary windings of the transformer supplying AC to the rectifier circuit and its apparent power is derived with the RMS values of its voltage and current, V_S and I_S.

$$TUF = \frac{P_{dc}}{V_s I_s}$$

where

$$V_S = \frac{V_m}{\sqrt{2}} = 0.707V_{max}$$

For half-wave rectification,

$$I_S = \frac{0.5V_{max}}{R}$$

and full-wave rectification,

$$I_S = \frac{0.707V_{max}}{R}$$

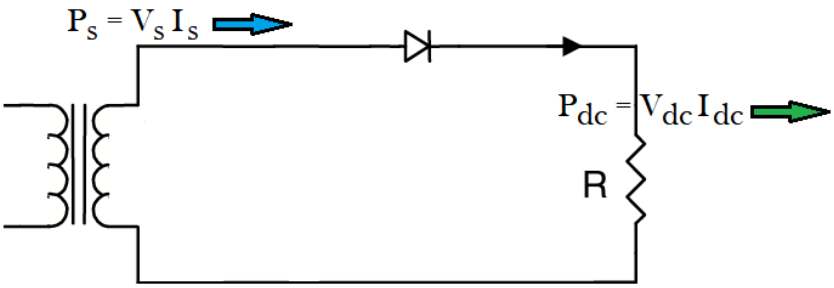


Figure 9.

The TUF calculation for a half-wave rectifier is as follows,

$$TUF = \frac{V_{dc}I_{dc}}{V_sI_s} = \frac{\mathbf{0.318V_{max}} \times \frac{\mathbf{0.318V_{max}}}{R}}{0.707 \times \frac{0.5V_{max}}{R}} = \frac{(0.318)^2}{0.707 \times 0.5} = 0.286$$

The TUF factor for this half-wave rectifier shows that for an apparent power input of 1VA, we have 0.286 VA dc power output. Otherwise stated, we require 3.496 VA power from the transformer to produce 1 VA of dc power from the rectifier. This is a poor efficiency.

Given that we know the following for a full-wave rectifier;

$$V_{dc} = 0.636V_{max}$$

$$I_{dc} = \frac{0.636V_{max}}{R}$$

$$V_S = \frac{V_m}{\sqrt{2}} = 0.707V_{max}$$

$$I_S = \frac{0.707V_{max}}{R}$$

We can derive the TUF value for full-wave rectification in a bridge rectifier,

$$TUF = \frac{0.636V_{max} \times \frac{0.636V_{max}}{R}}{0.707V_{max} \times \frac{0.707V_{max}}{R}} = \frac{(0.636)^2}{(0.707)^2} = 0.809 \approx 0.81$$

This TUF of the bridge rectifier shows a transmission of apparent power almost three times than that of the half-wave rectifier.

A real-world rectifier circuit aims to produce a DC output therefore it makes more sense to assess parameters in relation to V_{dc} rather than $V_{L_{rms}}$.

3-Phase Rectification

	Half-wave rectifier	Full-wave rectifier with center-tapped transformer	Full-wave bridge rectifier
Peak repetitive reverse voltage V_{RRM}	$3.14V_{dc}$	$3.14V_{dc}$	$1.57V_{dc}$
RMS input voltage per transformer leg V_s	$2.22V_{dc}$	$1.11V_{dc}$	$1.11V_{dc}$
Diode average current $I_{F(AV)}$	$1.00I_{dc}$	$0.50I_{dc}$	$0.50I_{dc}$
Peak repetitive forward current I_{FRM}	$3.14I_{F(AV)}$	$1.57I_{F(AV)}$	$1.57I_{F(AV)}$
Diode rms current $I_{F(RMS)}$	$1.57I_{dc}$	$0.785I_{dc}$	$0.785I_{dc}$
Form factor of diode current $I_{F(RMS)}/I_{F(AV)}$	1.57	1.57	1.57
Rectification ratio	0.405	0.81	0.81
Form factor	1.57	1.11	1.11
Ripple factor	1.21	0.482	0.482
Transformer rating primary VA	$2.69P_{dc}$	$1.23P_{dc}$	$1.23P_{dc}$
Transformer rating secondary VA	$3.49P_{dc}$	$1.75P_{dc}$	$1.23P_{dc}$
Output ripple frequency f_r	$1f_i$	$2f_i$	$2f_i$

Figure 10. Single-phase rectifier design parameters (Rashid, 2011)

The transformer parameters in the table show that single-phase rectification would require transformers with high apparent power (VA) ratings for high power applications. Because of this we turn to 3-phase AC systems for high power applications (Rashid, 2011).

There are two main types of 3-phase rectifiers. They are the ***star rectifier*** and the ***bridge rectifier***. The focus of this project report will be the bridge rectifier which has the highest transformer utilisation factor of any three-phase rectifier configuration.

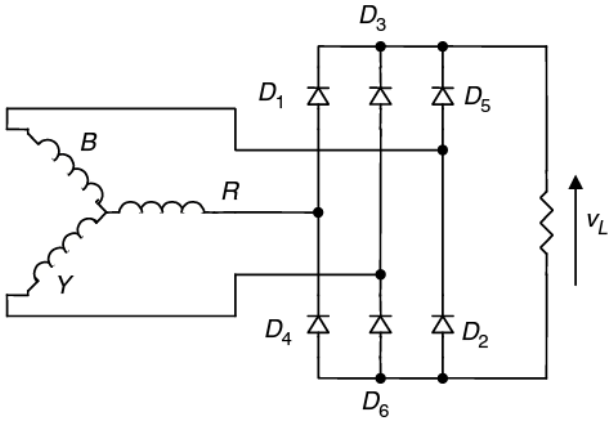


Figure 11. Three-Phase Bridge Rectifier (Rashid, 2011)

Diodes D1 – D6 are numbered in the order of their conduction sequence, and the time between the start of any two consecutive diodes is 60^o. They are supplied with a three-phase supply voltage as show in the figure below, where the phases are separated by 120^o.

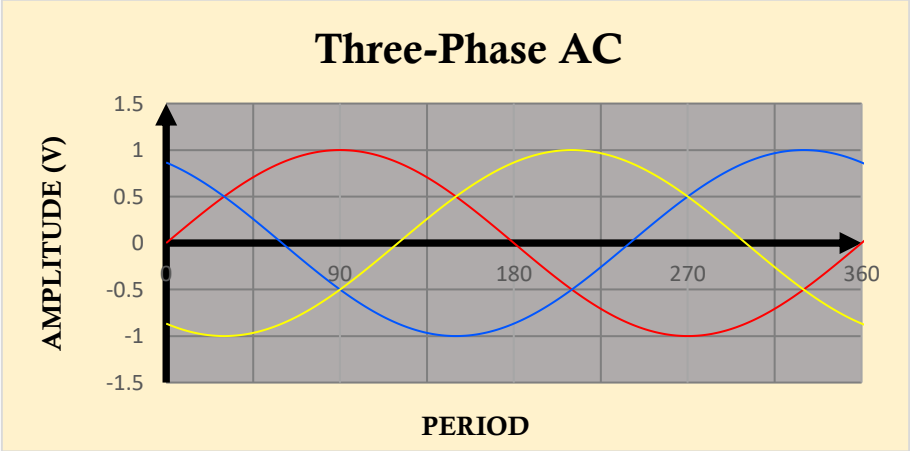


Figure 12.

The figure below shows the voltage and current waveforms of the bridge rectifier's dc output.

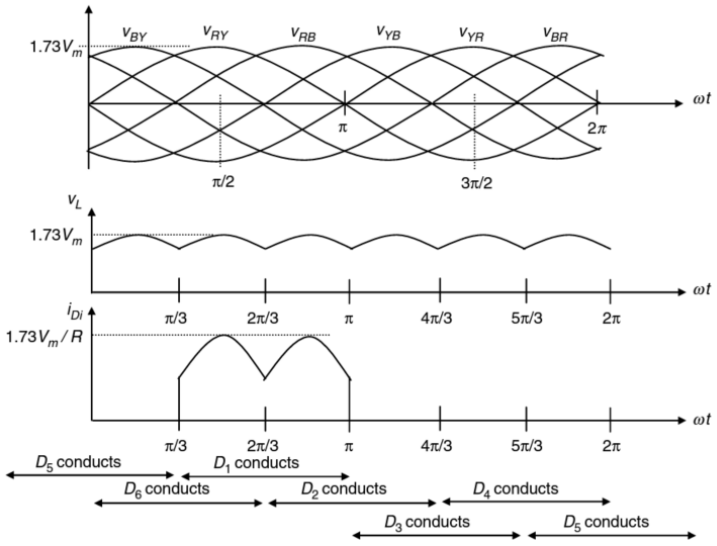


Figure 13. Voltage and current waveforms of 3-phase bridge rectifier output (Rashid, 2011)

For a star configured AC supply, the line-to-line voltage is found to be 1.73 times more than the phase voltage.

Recall that for a single phase,

$$V_{DC} = \frac{1}{T}\int_0^T V_L(t).dt$$

However, for a six-pulse DC output, the period under analysis is $\frac{2\pi}{6}$, so the average DC output of a 3-phase bridge rectifier with a peak voltage, V_m is given by;

$$V_{DC} = \frac{6}{2\pi}\int_{\pi/3}^{2\pi/3} \sqrt{3}V_m sin\emptyset \, d\emptyset$$

$$V_{DC} = V_m\frac{3\sqrt{3}}{\pi} = 1.654V_m$$

The RMS value is found to be

$$V_{L_{rms}} = V_m\frac{3\sqrt{3}}{\pi} = 1.655V_m$$

CONTROLLED RECTIFICATION

Thyristor Triggering Theory

Thyristors, commonly referred to as silicon-controlled rectifiers or SCRs, are three-terminal semiconductor devices with alternating p-type and n-type layers. Because of this construction thyristors have a better capacity to handle electrical power efficiently. Thyristors are controlled by a terminal called a gate. The other two terminals called the anode and cathode (Sueker, 2005).

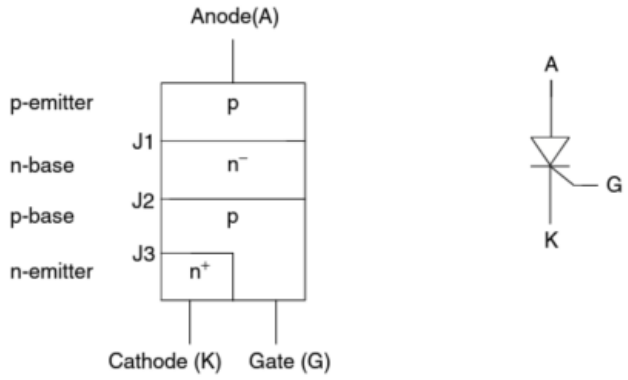


Figure 14. Thyristor construction (left) and schematic symbol (right) (Power Electronics Handbook, 2011)

At the gate terminal small DC triggering currents allow us to control the thyristor which can deliver large currents and withstand large voltages across the anode and cathode. This makes the thyristor ideal for a controlled bridge rectifier.

3-Phase Rectification

In phase-controlled rectifiers, power diodes are replaced with SCRs. Because we can limit the amount of time the SCR is switched on, the DC output of the rectifier circuit can be controlled. To do this we delay the firing pulse (switch-on signal) sent to the SCR. The longer the delay, the less DC voltage is supplied to the load. This delay is referred to as the *firing angle,* $\propto$ and is defined in relation to the period of the AC supply voltage.
Therefore, a firing angle of 0^o means that the SCR is switched on with no delay and therefore conducts 100% of the time. For a rectifier with *p* number of pulses, the maximum effective firing angle is $\frac{2\pi}{6} radians$ (Kazmierkowski et al, 2002).

For the continuous conduction condition, the relationship between the output voltage, V_{DC-CON} of a controlled rectifier and the output voltage of an uncontrolled rectifier, V_{DC-UNC} is given by

$$V_{DC-CON} = V_{DC-UNC} \cos(\alpha)$$

From this equation we see that as the firing angle rises above 90^o the DC output becomes negative. This means that the rectifier will have a lower potential than the load. Since the rectifier output current will always be positive, a negative voltage would mean that power is being drawn from the load.

SCR's are not practical for PWM rectifier control, however they are well-suited for phase-controlled rectification. The figure below shows the output waveforms for a 6-pulse bridge rectifier in the continuous conduction mode (Trzynadlowski, 2016).

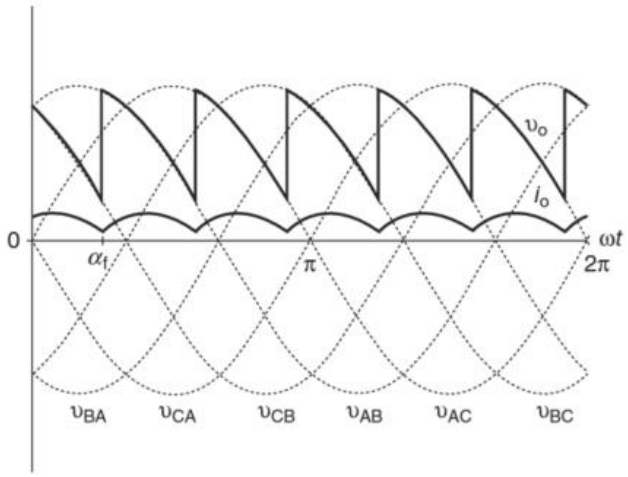


Figure 15. The voltage and current waveforms of six-pulse phase-controlled rectifier output (Trzyadlowshki, 2016)

The output voltage, V_{DC} (V_O in the diagram) in relation to the firing angle, α is given by

$$V_{DC} = \frac{1}{\frac{\pi}{3}} \int_{\alpha}^{\alpha+\frac{\pi}{3}} V_{max} sin\left(\omega t + \frac{\pi}{3}\right) d\omega t$$

$$= \frac{3}{\pi} V_{max} \cos(\alpha)$$

Continuous & Discontinuous Conduction Modes

Additionally, the load EMF and angle, and the firing angle affect the conduction mode. The graph below shows the continuous and discontinuous conduction areas a 6-pulse rectifier. Firing angles of 0^o, 30^o, 60^o, 90^o, 120^o and 150^o have their respective lines. The area under the line represents the continuous conduction mode for that firing angle (Kazmierkowski et al, 2002).

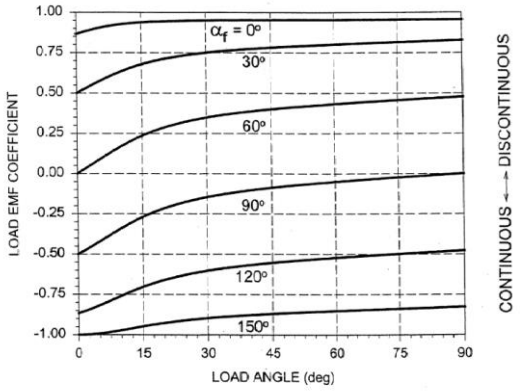


Figure 16. Conduction modes of 6-pulse controlled rectifier (Kazmierkowski et al, 2002)

The cross-over angle, α_C is the angle at which the input voltage exceeds (crosses over) the level of the load EMF (Trzyadlowshki, 2016).

$$\alpha_C = sin^{-1}(\varepsilon) - \frac{\pi}{3}$$

Where ε is the load EMF coefficient.

Because the load EMF can affect the biasing of the SCR, **not all values of the firing angle will cause conduction**.

Simplified, the SCR will not fire when $\alpha \leq \alpha_C$ or $\alpha \geq {}^{\pi}/_{3} - \alpha_C$

In contrast, the SCR will fire when $\alpha > \alpha_C$ or $\alpha < {}^{\pi}/_{3} - \alpha_C$

Rewritten as $\alpha_C < \alpha < {}^{\pi}/_{3} - \alpha_C$

Since we know the relationship between the cross-over angle and the load EMF coefficient ε, we can specify the SCR firing conditions as

$$\boldsymbol{sin^{-1}(\varepsilon) - \frac{\pi}{3} < \alpha < {}^{2\pi}/_{3} - sin^{-1}(\varepsilon)}$$

At α, the continuous output current is given as

$$i_{DC} = i_{DC}(\alpha + {}^{\pi}/_{3})$$

Given the load angle, φ and load impedance, Z we can determine the output current waveform in the continuous conduction mode.

$$i_{DC} = \frac{V_{max}}{Z}\left[\sin(\omega t + \frac{\pi}{3} - \varphi) - \frac{\varepsilon}{\cos(\varphi)} + \frac{\sin(\varphi - \alpha)}{1 - e^{-\frac{\pi}{3\tan(\varphi)}}} e^{\frac{\omega t - \alpha}{\tan(\varphi)}}\right]$$

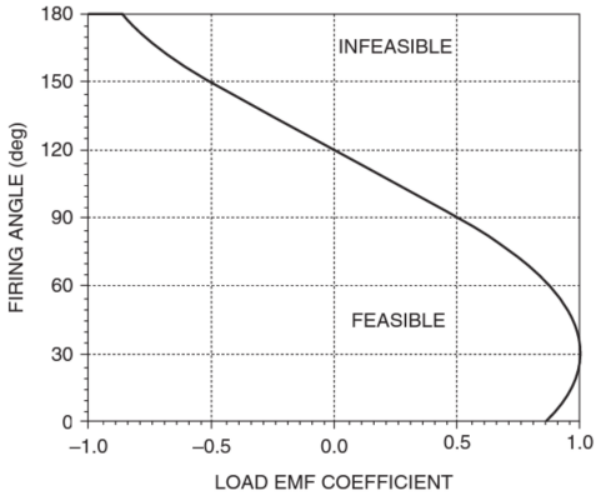


Figure 17. Feasible firing angles (Trzyadlowshki, 2016)

Given the condition for continuous conduction, $i_{DC}(\alpha) > 0$
Then

$$\varepsilon < \left[\sin(\alpha + \frac{\pi}{3} - \varphi) + \frac{\sin(\varphi - \alpha)}{1 - e^{-\frac{\pi}{3\tan(\varphi)}}}\right]\cos(\varphi)$$

$$V_{DC} = \frac{3}{\pi}V_{max}\left[2\sin\left(\alpha + \frac{\beta}{2} + \frac{\pi}{3}\right)\sin\left(\frac{\beta}{2}\right) + \varepsilon(\frac{\pi}{3} - \beta)\right]$$

Where the conduction angle, β is given by

$$\beta = \alpha_C - \alpha$$

In the continuous conduction mode $\beta = {}^{\pi}/_{3}$

3-PHASE STEPDOWN TRANSFORMER

Transformers are quite common in power supply systems where a large reduction is required from the mains supply. They make use of Faraday's law;

> **"The change of magnetic flux in coil in will induce an electromotive force in the coil that is proportional to that change."**

The ratio of the input to output voltages is equal to the ratio of the number of turns in the primary winding to that of the secondary winding. Using this principle, the transformer can produce a reduced or increased voltage output based on its turns ratio (Schmidt-Walter and Kories, 2007).

In this project, the mains supply of 230v ac is too large to be handled by a typical zero-crossing detector and PCB rectifier circuitry. More importantly the intended load dictates how large the step-down in voltage needs to be. The figure below shows a basic model of an ideal transformer. For much of the design and analysis carried out in the subsequent sections of this report, the ideal model is used.

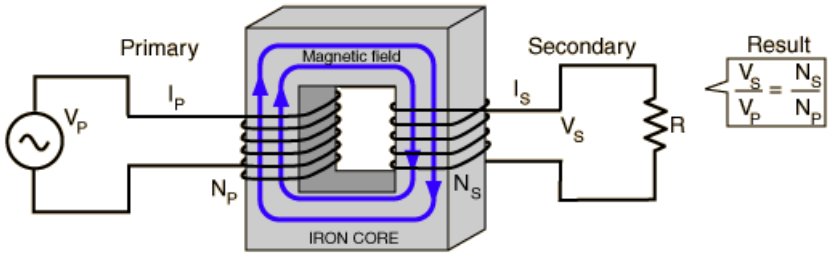


Figure 18. Basic transformer construction (adapted from www.Hyperphysics.com)

However, it is important to note that real transformers have other significant parameters. Real transformers have 'exciting currents' and 'leakage losses' as shown in the figure below (Sueker, 2005).

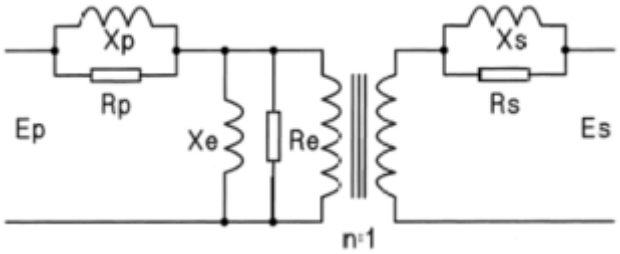


Figure 19. Model of real transformer (Power Electronics Design - Keith H. Sueker)

Exciting current is all the current needed to produce the desired flux density in the core of the transformer, plus eddy current and hysteresis losses. Exciting current only exists on the primary side of the core and does not typically affect the secondary side. As a result, it is usually not considered in transformer calculations.

On the other hand, leakage reactances (inductive) can be transferred from the primary to secondary windings. This can have a considerable effect on the efficiency of the rectifier system such as the introduction of harmonics, and electromagnetic interference which can in turn cause damage to the load or the power supply grid.

FILTERS

Undesired Effects

There are many 'side-effects' and external influences that affect the functionality of electronic circuits

Harmonics

The primary or *fundamental frequency* is the lowest frequency components of a current or voltage waveform. The fundamental frequency may often be composed of multiple frequency components, or *harmonics* which are caused by electromagnetic disturbances or distortions. Sometimes these harmonics exist by design whereas other times they are simply side effects of undesired or unpredictable influences (Trzynadlowski, 2016).

Electromagnetic Interference (EMI)

EMI refers to the effects caused by electromagnetism that can be produced within the circuit or from external sources. Electromagnetic interference and Electromagnetic compatibility is discussed in more detail in later sections of this report.

Transients

Transients are quick bursts of energy that result from the change in a steady state in a system. In power electronics transients arise from the switching actions of semiconductor devices such as SCRs, FETs, BJTs and so on.

The aim of filtering circuits to remove or minimise undesired properties from a signal or power supply, such as harmonics, EMI and transients (Bai & Mi, 2011).

DC Filters

Reservoir Capacitor

A simple capacitor can reduce the size of the ripples in the pulsed DC output of a rectifier.
φ is the charging period for the reservoir capacitor. It is dependent on the size of the capacitor and the internal resistance of the transformer. The output current, I_{DC} is equal to the average value of the diode current I_{diode} (Schmidt-Walter and Kories, 2003).

$$I_{diodeRMS} \approx 1.5 \dots 2 \cdot I_O$$

$$I_{diode-PK} \approx 4 \dots 6 \cdot I_O$$

The RMS value of the secondary transformer current is equal to the RMS value of the diode current. This must be considered when selecting the apparent power, S_N of the transformer.

$$I_S = I_{diodeRMS}$$

S_N needs to be approximately double the value of the output power, P_{OUT}. The peak-to-peak ripple voltage, V_{R-Pk} is usually set to 20% of the output voltage, V_{OUT}. This can be achieved by selecting suitable values for the filter capacitor, whose discharge time will be approximately half the period of the supply voltage, T. Given the relationship,

$$i = C\frac{dv}{dt}$$

The capacitance required is determined as

$$C_{fil} \approx \frac{I_{OUT} \cdot {}^{T}/_{2}}{V_{R-Pk}} = \frac{I_{OUT} \cdot {}^{T}/_{2}}{V_{OUT} \cdot 0.2}$$

For a mains supply of 50Hz, the capacitor will then be given by the formula

$$C_{fil} = \frac{I_{OUT}}{V_{OUT}} \cdot \frac{\frac{1}{50} \cdot \frac{1}{2}}{0.2} = \frac{I_{OUT}}{V_{OUT}} \cdot \frac{{}^{1}/_{100}}{{}^{1}/_{5}} = \frac{I_{OUT}}{V_{OUT}} \cdot \frac{5}{100}$$

$$C_{fil} = 0.05 \times \frac{I_{OUT}}{V_{OUT}}$$

If we assume the mains voltage is 10% less than its rated value and we neglect the diode voltage drops (0.6v), the minimum output voltage is given by

$$V_{O-min} \approx 0.9 \cdot V_N \cdot \sqrt{2} \cdot 0.8$$

Where the transformer rated voltage, $V_S \geqq V_{O-min}$

Voltage Regulators

Most power supply units incorporate a voltage regulator after the filter capacitor. If a regulator itself has a voltage drop of Pv, then it is important that the minimum output voltage of the filter circuit is Pv higher than the final regulated voltage, V_{final}

$$V_{O-min} = P + V_{final}$$

A Zener diode can be used in the simplest form of a voltage regulator. Typically it will be placed across the output of the filter circuit to provide a stable voltage output.

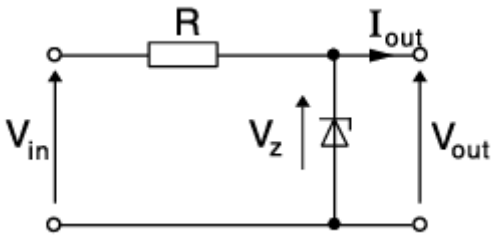


Figure 20. Use of a Zener diode to provide analog voltage stabilisation (Schmidt-Walter and Kories, 2003)

(RC, RL, LC Filters)

The filter circuit of the power supply aims to remove the AC components from the DC output of the rectifier. A basic passive filtering system can be made of a few combinations of resistors, capacitors and inductors.

The 3 most common voltage filters are the *inductor filter (RL)*, the *capacitor filter* (RC) and the *'choke' filter (LC)*.

The **inductor filter** also called an RL filter has an inductor in series with the load and a resistor parallel to it. The inductor blocks the AC component of the output of the rectifier passes but allows the DC through. The resistor in parallel with the load forms a voltage divider circuit so that the voltage across the load, V_{load} is given by;

$$V_{load} = V_{DC} \frac{R_{load}}{(R_i + R_{load})}$$

From this equation we can see that the output DC voltage is dependent on the size of the load.

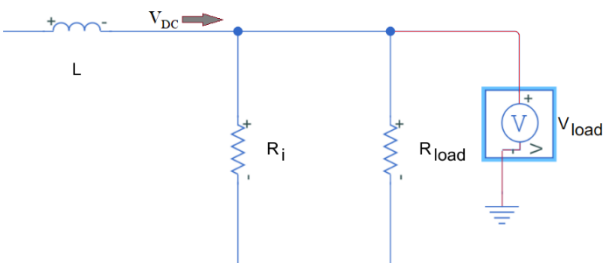


Figure 21. LR filter

The **capacitor filter** employs a resistor in series with the load and a capacitor in parallel to it. When the rectifier is conducting the capacitor charges up, but when the rectifier stops conducting, or when the voltage output falls to less that the voltage value of the capacitor, the capacitor discharges through the load. Because the capacitor delivers charge when the rectifier's output falls (causing ripples) the size of the ripples are greatly reduced. If the ripple current has frequency, *f* and the capacitor has a value, *C* then the parallel/shunt resistance given by the capacitor is given by;

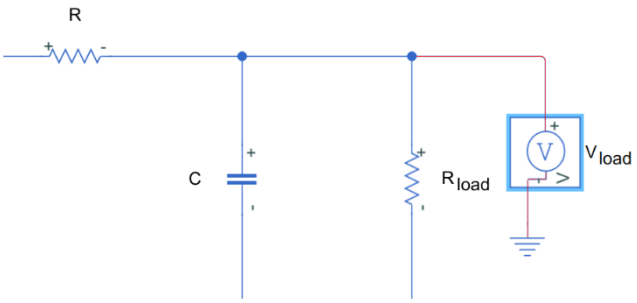


Figure 22. RC filter

$$R_{shunt} = {}^{1}/_{2fc}$$

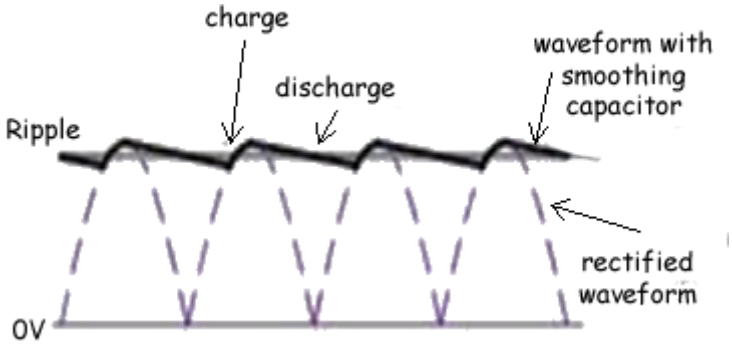


Figure 23. Effect of capacitor filter on pulsed DC (Cyberphysics, 2018)

The problem with the capacitor filter is that the larger the capacitor, the larger its current draw will be, which will destroy the diodes (or thyristors) if their current rating is exceeded. The formula that explains this relationship is;

$$Q = I \times t$$

This formula shows how the charge of the capacitor is reliant on the current it can draw.
On the other hand, in the inductor filter the number of ripples in the output is affected by changes in the load impedance. The series inductor also reduces the output voltage and current.

However, if we combined the series inductor with the shunt capacitor we can prevent damage to the thyristors and remove enough of the ripple without decreasing the total output current. This type of filter is called an **L-C filter** or an L-Section filter.

Transient Response of LC Filters

In analysis of the LC filter designers should not only asses the steady-state response, but also the transient response and how it affects the desired performance.

The time constant, τ is a parameter that characterises the behaviour of a first-order, linear time-invariant system. RC and RL filter circuits are examples of such first-order systems. However, the LC filter is a second-order system which is usually characterised by its natural frequency, ω_n and the damping ratio, ζ. Various amplitude responses to the LC filter are shown in the figure below. The ideal amplitude response is achieved when ζ is 1 (Shoaib, 2013).

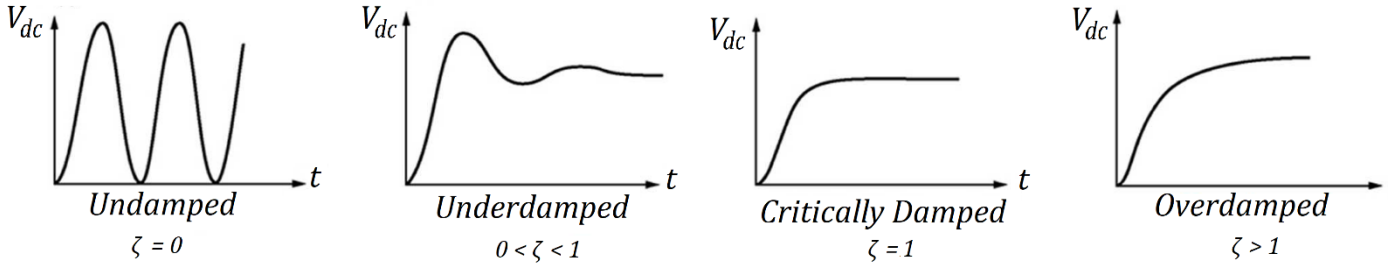


Figure 24. Effect of damping ratio on amplitude response of second-order filters (Queen Mary University, 2016)

For a single-ended LC filter as shown in the figure below,

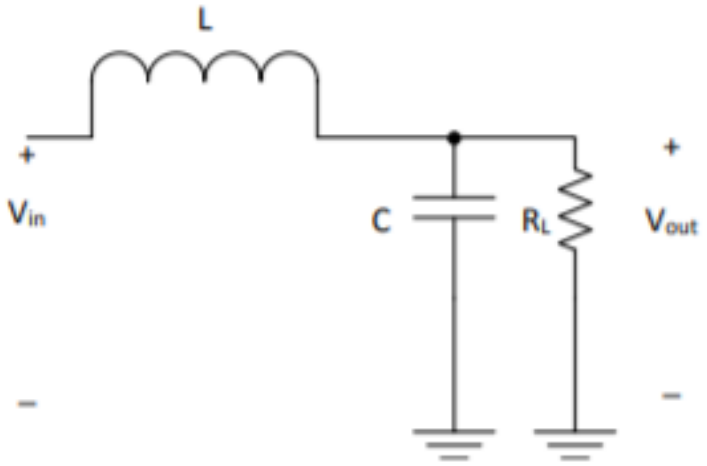


Figure 25. Single-ended LC filter (Texas Instruments, 2016)

The amplitude response of the LC filter is given by the formula;

$$\zeta = \frac{1}{2R_L\sqrt{\frac{C}{L}}}$$

HARMONIC CONTENT

The research of scientist Joseph Fourier established that any complex signal can be expressed as the sum of multiple sine waves, where the lowest frequency is known as the fundamental and smaller frequencies are multiples of this known as harmonics (Kim, 2017).

$$i = i_f + i_H$$

Where,

$$i_H = i_h + i_{DC}$$

So,

$$i = i_f + (i_h + i_{DC})$$

The figure below shows resulting current drawn from a linear load when a voltage is applied. This shows no distortion in the resulting current waveform.

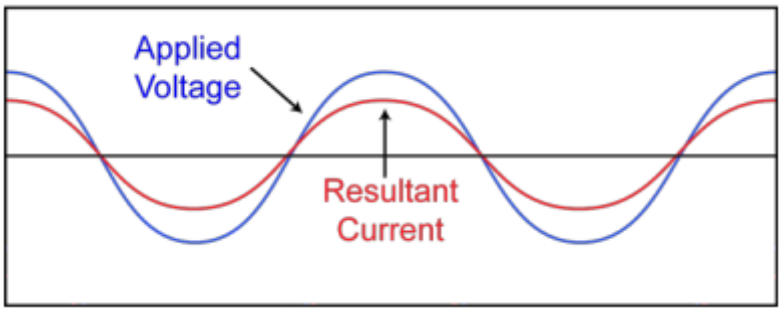


Figure 26. Distortions between current and voltage waveforms (Starline, 2018)

However, non-linear loads cause the waveforms of the currents they draw to be distorted. This results in harmonics being created. Distorted current into a non-linear load can also cause distorted voltage drops all throughout the power system. It also makes the system less efficient resulting in power losses and other problems to the power supply grid (Yaskawa, 2004).

We can limit voltage harmonics by minimising current harmonics. Typically, in an AC system, even harmonics are absent. Also, in a balanced three-phase system, odd harmonics that are multiples of 3 are typically absent. However, perfectly balanced systems rarely occur. As a result, the most prominent harmonics tend to be the **5th, 7th, 11th, and 13th.**

The output of the a three-phase bridge rectifier consists of six half-sinewaves. The Fourier expansion allows us to look at the components of this output. Fourier's expansion shows that a function, f(t) is made up of the sum of the fundamental component, the even harmonics and the odd harmonics. This is expressed in the equation;

$$f(t) = a_O + \sum_{k=1}^{\infty} [a_k \cos(k\omega_O t) + b_k \sin(k\omega_O t)]$$

Where the fundamental, a_O is the DC component, given by

$$a_O = \frac{1}{T} \int_{t_O}^{t_O+T} f(t)dt$$

and the odd and even harmonics are given by

$$a_k = \frac{2}{T} \int_{t_O}^{t_O+T} f(t)\cos(k\omega_O t)dt$$

$$b_k = \frac{2}{T}\int_{t_O}^{t_O+T} f(t)dt\sin(k\omega_O t)dt$$

For a 6-pulse rectifier the duration of each pulse is $T/6$ or $\pi/3$. This then gives the fundamental as

$$a_O = \frac{3}{\pi}\int_0^{\pi/3} f(t)dt = \frac{3}{\pi}\int_0^{\pi/3} I_S \cdot \sin(\omega t)dt$$

$$a_O = \frac{3}{\pi} I_S[-\cos(\omega t)]_0^{\pi/3}$$
$$a_O = \frac{3}{\pi} I_S[-0.5 - (-1)]$$
$$a_O = \frac{3}{\pi} I_S \cdot \frac{1}{2}$$
$$a_O = \frac{3}{2\pi} I_S = I_{DC}$$

and the harmonics as,

$$a_k = \frac{6}{\pi}\int_0^{\pi/3} I_S \cdot \sin(\omega t)\cos(k\omega_O t)dt$$

$$b_k = \frac{6}{\pi}\int_0^{\pi/3} I_S \cdot \sin(\omega t)dt\sin(k\omega_O t)dt$$

This is significant because we can predict the ripple components that need to be filtered out. For the LC filter, the only significant ripple component is the 2nd harmonic, given by.

In a harmonic analysis of the rectifier, one of the most important figures of merit is the ***Total Harmonic Distortion***. It is the ratio of the sum of all the harmonic frequencies to the fundamental frequency. Simply put, it is the measure of the total distortion in a signal.

Diode Rectifier Harmonics

The Total Harmonic Distortion can be derived as follows;

$$THD = \frac{RMS\ of\ Total\ Harmonics}{RMS\ of\ Fundamental\ Wave}$$

$$THD_i = \frac{I_H}{I_1} = \frac{\sqrt{I^2 - I_1^2}}{I_1} = \frac{\sqrt{\Sigma_{n\neq 1} I_n^2}}{I_1}$$

$$THD_V = \frac{V_H}{V_1} = \frac{\sqrt{V^2 - V_1^2}}{V_1} = \frac{\sqrt{\Sigma_{n\neq 1} V_n^2}}{I_1}$$

(Pejovic, 2007)

POWER FACTOR

The power factor is the ratio between the real and the apparent power transferred in rectifiers. For a diode 3-phase bridge rectifier the conversion efficiency would be 100% if we assume there is an ideal DC current output. The power factor would therefore be given as;

$$PF = \frac{P_i}{S_i} = \frac{P_o}{S_i} = \frac{V_{o,dc}I_{o,dc}}{\sqrt{3}V_{LL}I_L} \approx 0.955$$

where S_i is the apparent input power.

The degree of the firing angle has an equal phase shifting effect on the **input current** thereby reducing its power input from what it'd be without a phase shift (or in a similar diode rectifier). The power factor, affected by the firing angle, is therefore given by the equation;

$$PF = \frac{V_{DC}}{V_{max}} = \frac{3}{\pi}\cos(\alpha) \approx 0.95cos(\alpha)$$

CONCEPTUAL RECTIFIER DESIGNS - MODELLING, SIMULATION & ANALYSIS

DESIGN REQUIREMENT AND SPECIFICATION

My objective convert a standard UK 240 V_{RMS}, 3-phase AC mains supply to a pure DC voltage of 60v - 70v that could be further reduced or distributed to external subsystems. Within the scope of this project I was not working to any industry standards, however with those in mind I wanted to assess the quality of the DC supplied by the controlled and uncontrolled rectifier designs.

UNCONTROLLED DIODE BRIDGE RECTIFIER

The theory is that an uncontrolled rectifier should perform identically to a controlled rectifier with a firing angle of 0 degrees. As a result, I chose to model, simulate and analyse an uncontrolled rectifier circuit before attempting to doing so with the thyristor-controlled rectifier circuit. It was my belief that doing so would make the more complex thyristor-controlled rectifier project more manageable.

The uncontrolled rectifier utilises a diode bridge in place of a thyristor network. The figure below shows my Simulink model of the diode bridge rectifier.

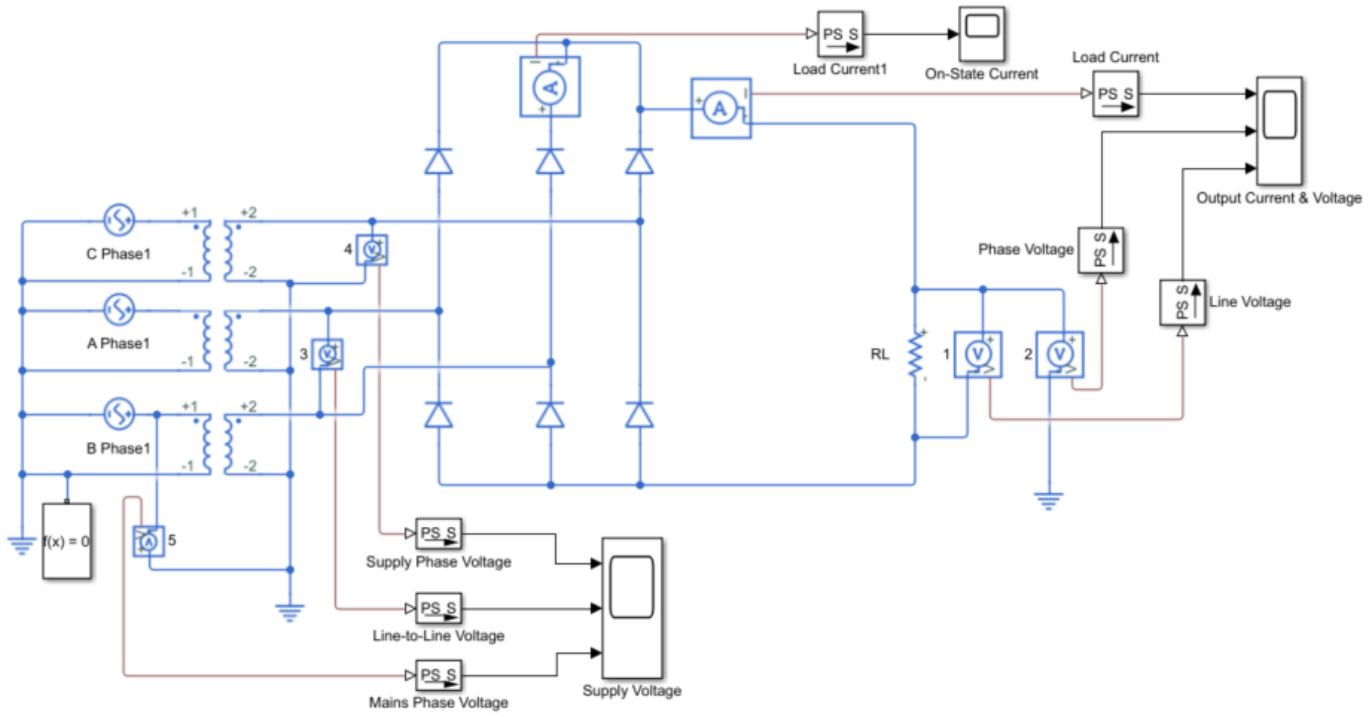


Figure 27. Diode bridge rectifier model

The model starts with the 3-phase ac supply. The Simulink block parameter value is set at 240V_{RMS} phase voltage. This is then fed into the primary windings of the transformer so will be called $V_{Primary}$. To achieve a 240V_{RMS} supply, the Peak Amplitude parameter in the settings of the AC Voltage Source block model was set to 340v.

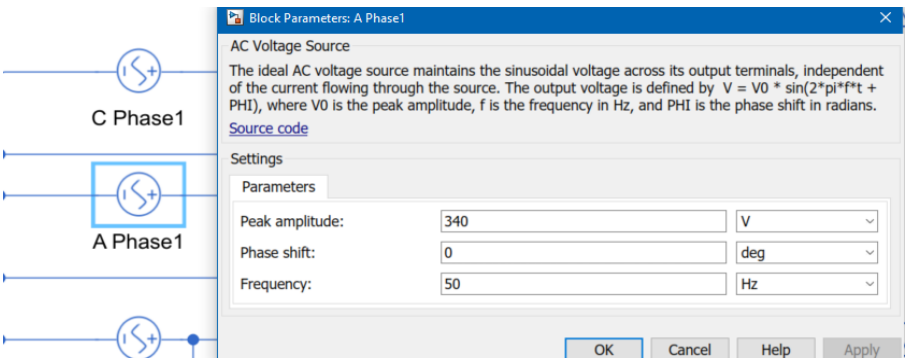


Figure 28. AC Voltage Source settings

This is based on the relationship;

$$V_{Primary-Pk} = V_{Primary-RMS} \times \sqrt{2}$$

$$\therefore V_{Primary-Pk} = 240 \times \sqrt{2} = 339.4113$$

$$V_{Primary-Pk} \approx 340V$$

The $240V_{RMS}$ of each phase was reduced using stepdown transformers with a winding ratio of **8:1.** The Ideal Transformer block model was used for this, producing a transformer output voltage (V_S) of $30V_{RMS}$. The peak voltage of the transformer supply is therefore calculated as;

$$V_{pk} = V_{S-RMS} \times \sqrt{2}$$

$$= 30 \times \sqrt{2}$$

$$\boldsymbol{V_{pk} \approx 42.43}$$

The six diodes in the bridge were set to have a forward voltage drop of 0.6v which is a typical value for locally supplied diodes. With these parameters set, the output DC voltage expected was calculated as follows;

$$V_{DC} = \frac{3\sqrt{3}}{\pi} \cdot V_{pk}$$

$$V_{DC} = \frac{3\sqrt{3}}{\pi} \cdot 42.43$$

$$\boldsymbol{V_{DC} = 70.18v}$$

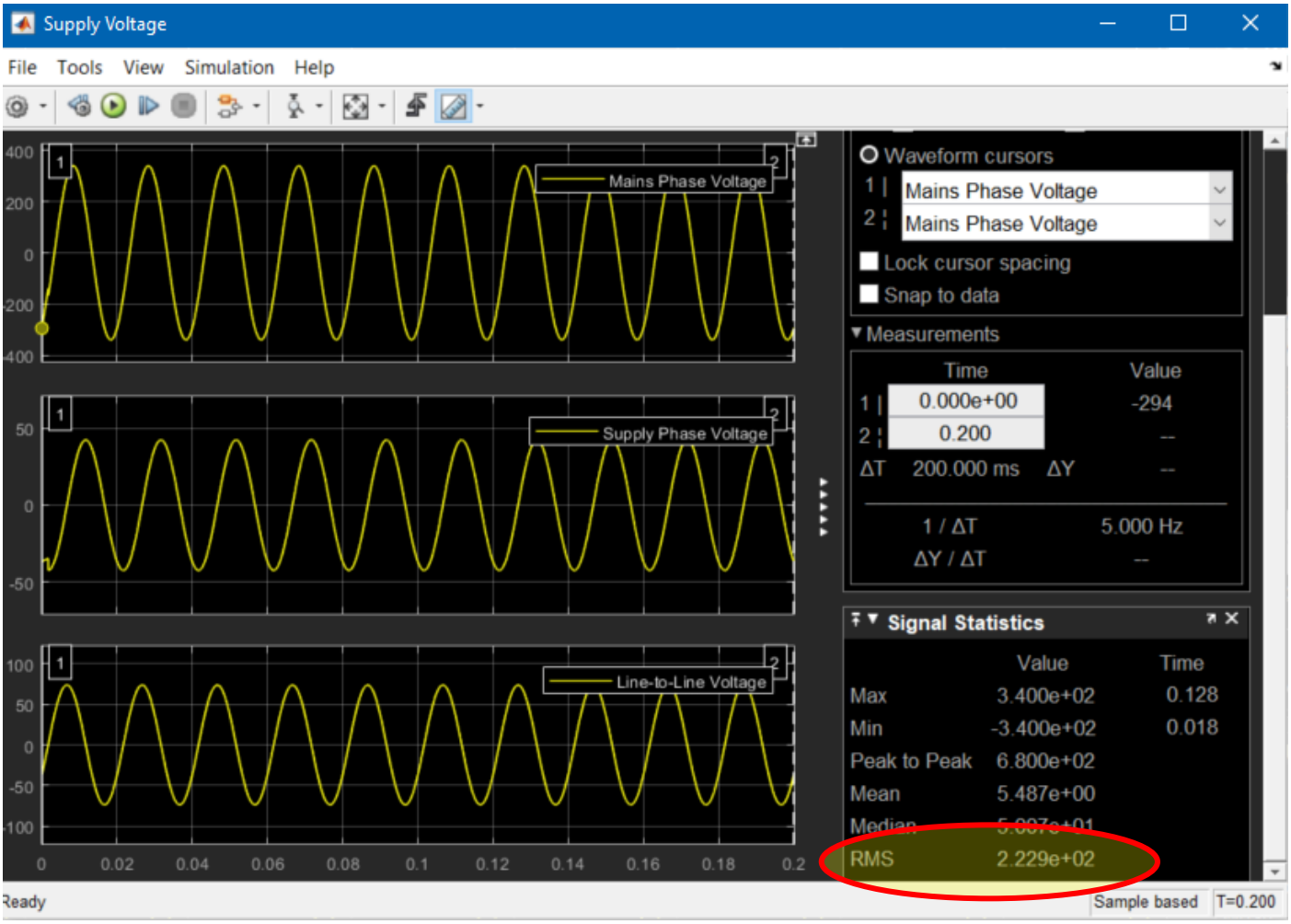


Figure 29. Transformer input and output voltage measurements

After simulation, the Supply Voltage scope showed a mains supply voltage of 222.90v. This is a considerable difference from the expected RMS reading of 240v. The cause of this I believed to have affected the final output. At the time of writing this report, I assumed that this is down to power factor losses on the input that the simulator may have been measuring.

However, if the output DC voltage V_{DC} is taken from the maximum output/peak output voltage (after smoothing filters) then the measured value of **71.98v** shown in the figure below, come quite close to the calculated value, $70.18v$

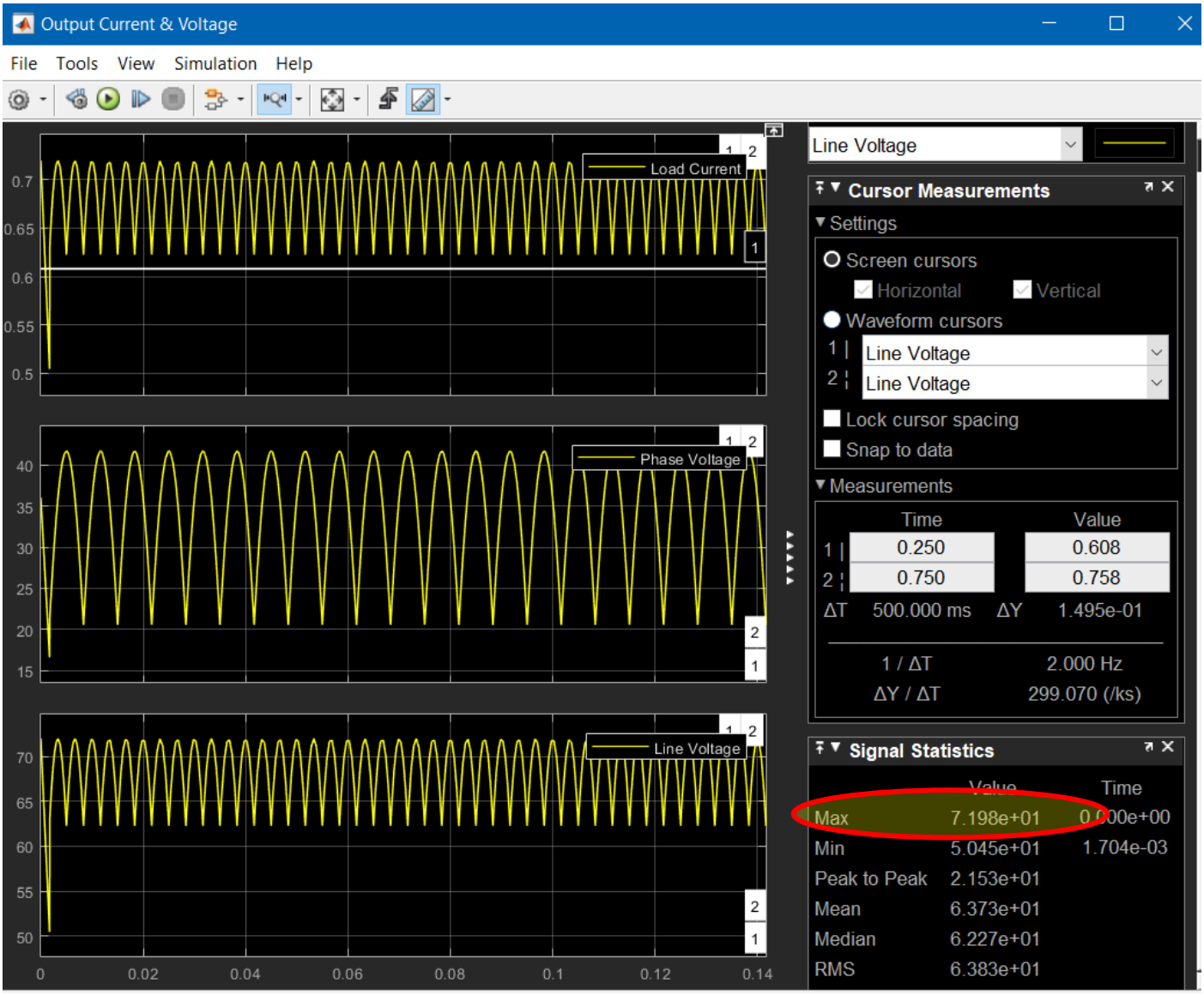


Figure 30. Rectifier DC output voltage and current.

CONTROLLED DIODE BRIDGE RECTIFIER

Figure 31 below shows the model that will allow us to simulate the fully-controlled bridge rectifier with the given phase angle, thereby making a comparison between the calculated and simulated result. Here the diodes of the uncontrolled bridge rectifier are replaced with thyristors whose gates are connected to the output ports of the tgc_firing program in the HDL block. The model also shows an LC filter network at the output of the rectifier.

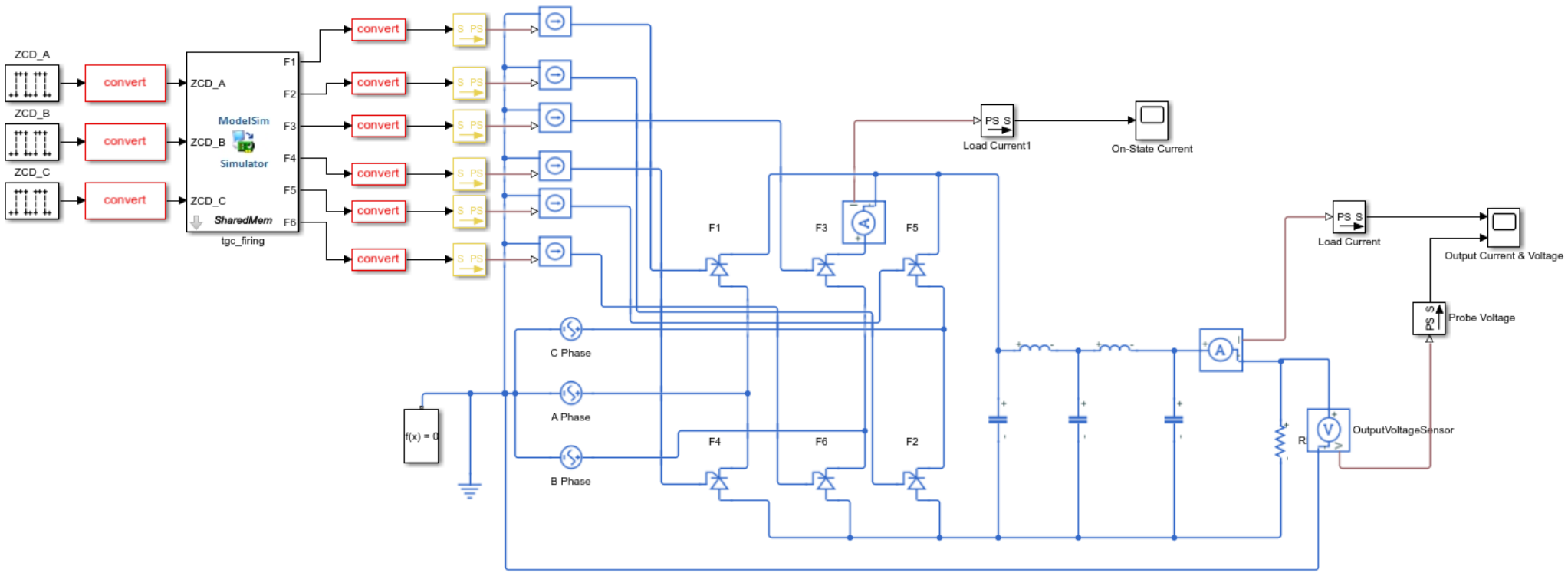


Figure 31. Controlled rectifier system model

Explanation of ZCD Model

An effective zero-crossing detector is critical for accurate timing of the thyristor firing sequence. Due to project constraints a functional zero-crossing detector was not implements into the control rectifier circuit. Instead, I simulated the output of a zero crossing detector by creating pulse generators, ZCD_A, ZCD_B and ZCD_C.

The firing angle is measured from the point at which the decreasing output of one rectified sine half-wave is exceeded by the subsequent sine half-wave of another phase. These are indicated by the green circles in the figure below. I realise that detection of this crossover point can be a simple means of synchronising the thyristor firing pulses. However, my approach was different. Instead I chose to simulate detection of zero crossings on the source voltages which would occur at the points shown in the red circles in the figure below.

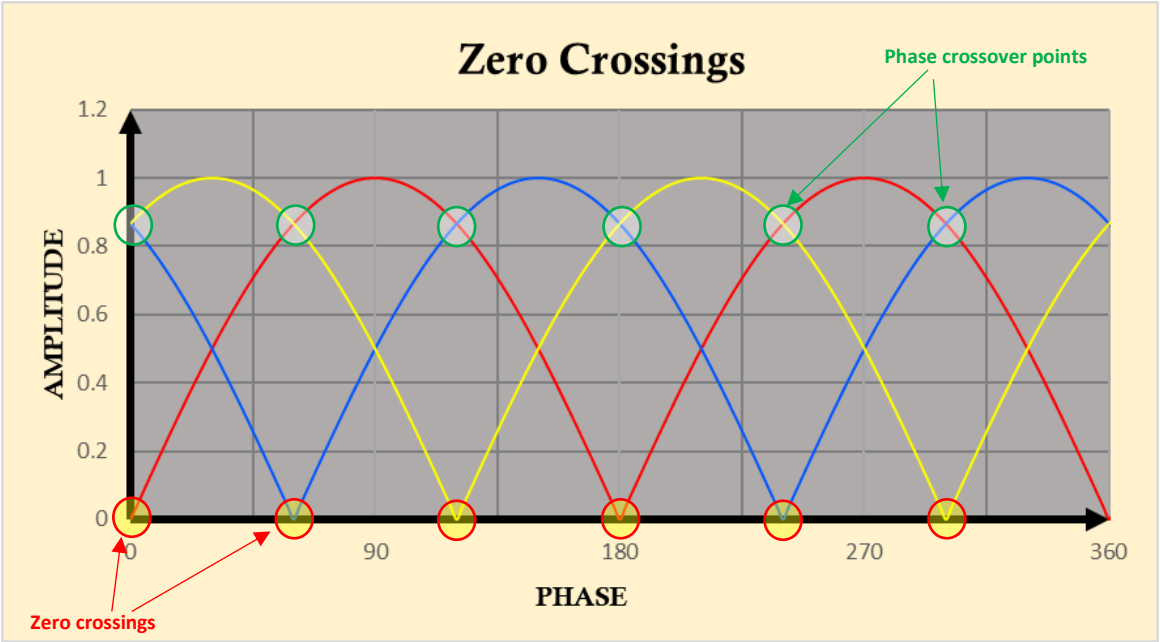


Figure 32. Phase angle reference points and zero crossings.

This is explained in further detail in the later section on the operation of the digital thyristor firing program.

Thyristor-Controlled Rectifier Calculations

Ripple Factor – The ripple factor of the diode bridge rectifier is 0.042.

$\boldsymbol{V_{DC}}$ - The calculated value of the DC voltage output in the uncontrolled bridge rectifier was found to be $70.18V$. As found in the literary review, I can use the uncontrolled V_{DC} value of the diode bridge rectifier to predict the V_{DC} value for a controlled rectifier with a given firing angle, α;

$$V_{DC-CON} = V_{DC-UNC}\cos(\alpha)\ .$$

For instance, with $V_{DC-UNC} = 70.18v$, and a firing angle of $20^{\circ}\left(\frac{\pi}{9} radians\right)$ used in the Thyristor Gate Control pulse firing program (described in the following section), the expected controlled DC voltage will be given by; At $\alpha = 20^{\circ}$,

$$V_{DC-CON} = 70.18\cos\left(\frac{\pi}{9}\right)$$
$$V_{DC-CON} = 65.948$$

Power Factor

From the literary review it was discovered that the power factor for the diode bridge rectifier producing an ideal DC current output is approximately 95.5% whereas the power factor for a controlled rectifier is given by;

$$PF \approx 0.95cos(\alpha)$$

In the digital program written for cosimulation in later sections, the firing angle is set at $\frac{\pi}{9}$ radians or 20°. Therefore, the expected power factor for the controlled rectifier would be given as;

$$PF \approx 0.95cos\left(\frac{\pi}{9}\right)$$
$$PF \approx 0.8927$$

Filter Calculations

From simulation, the measured output current, $I_{DC} = 0.854A$

The value for a filter capacitor with a discharge time of approximately $\frac{T}{2}$ is calculated as;

$$C_{fil} = 0.05 \times \frac{0.854}{70.18}$$

$$C_{fil} = 0.6mF$$

The scope measurement below shows the output waveform without the filter/smoothing capacitor.

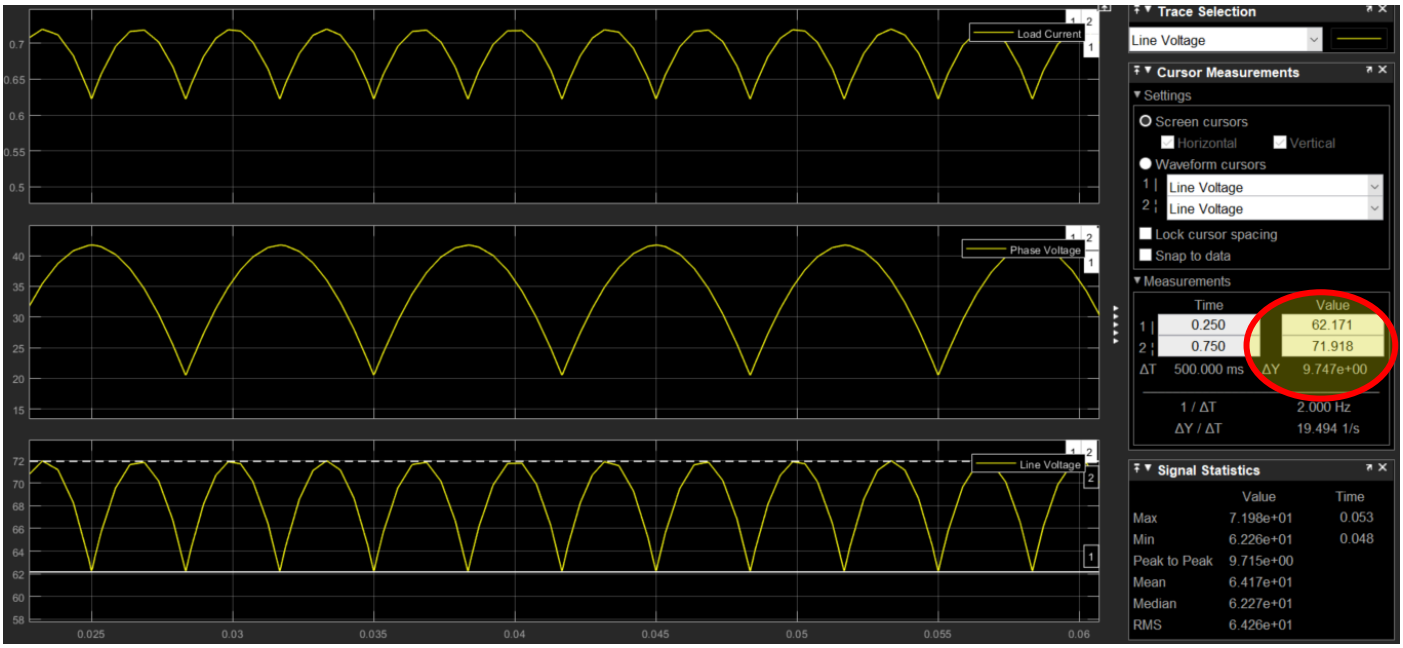


Figure 33. DC output ripple voltage measurement.

From the line voltage waveform, the ripple voltage is measured as approximately 9.7 volts. In the figure below the filter capacitor of 0.6mF (as calculated above), is put across the output.

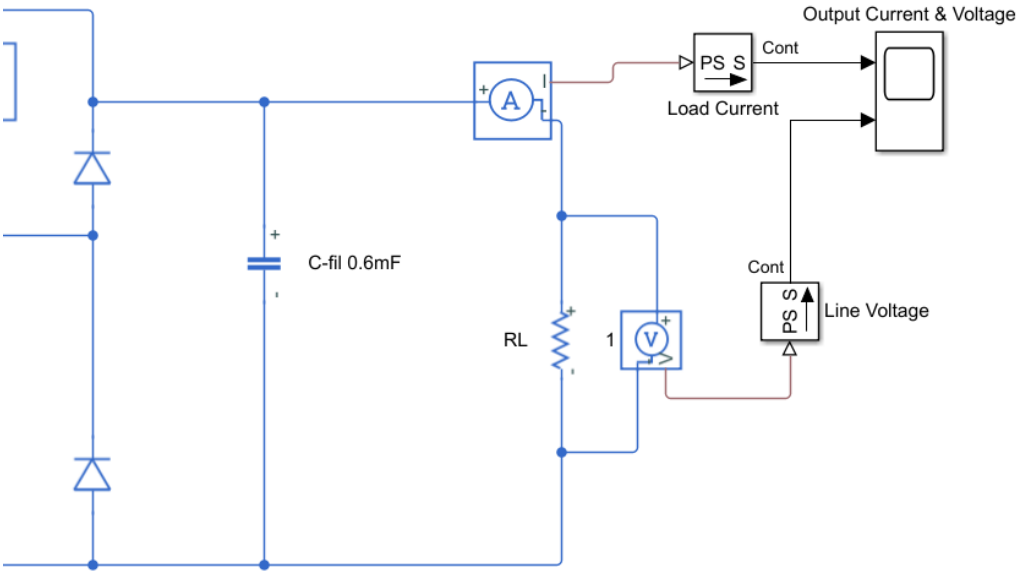


Figure 34. Smoothing capacitor applied to output of bridge rectifier

The resulting ripple shown in the figure below has been reduced to approximately one third the initial size.

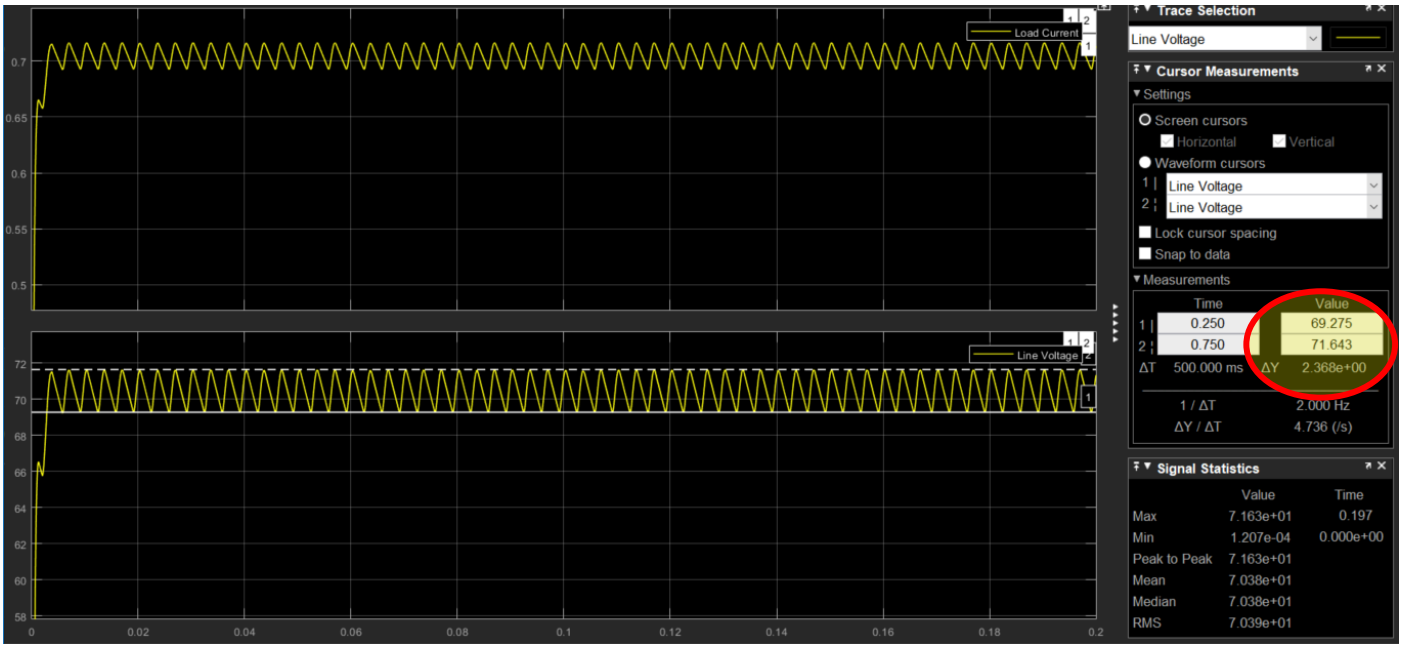


Figure 35.

If an inductor filter was used the ***critical inductance,*** L_C would be;

$$L_C = \frac{R}{3\pi m(m^2-1)f}$$

$$L_C = \frac{100}{3\pi 6(6^2-1)50}$$

$$L_C = 1.0105mH$$

In the figure below, we can see the placement of the inductor across the output of the rectifier.

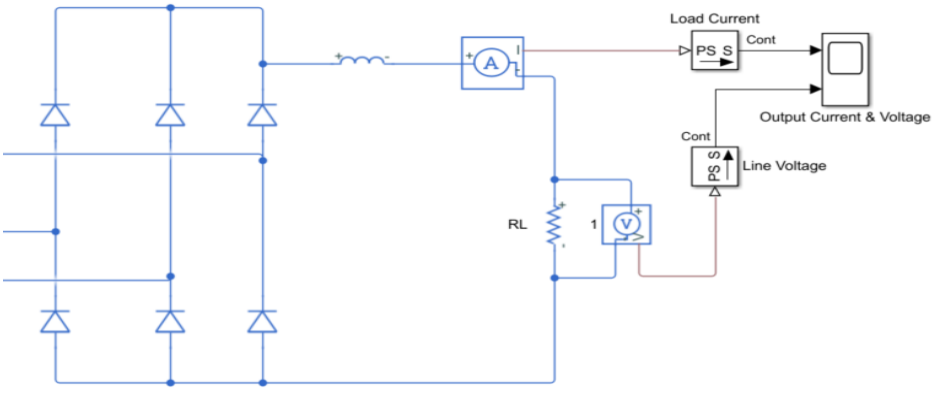


Figure 36. Inductor applied to output of bridge rectifier.

Below we see the resulting waveforms when using a 100mH and a 1H inductor.

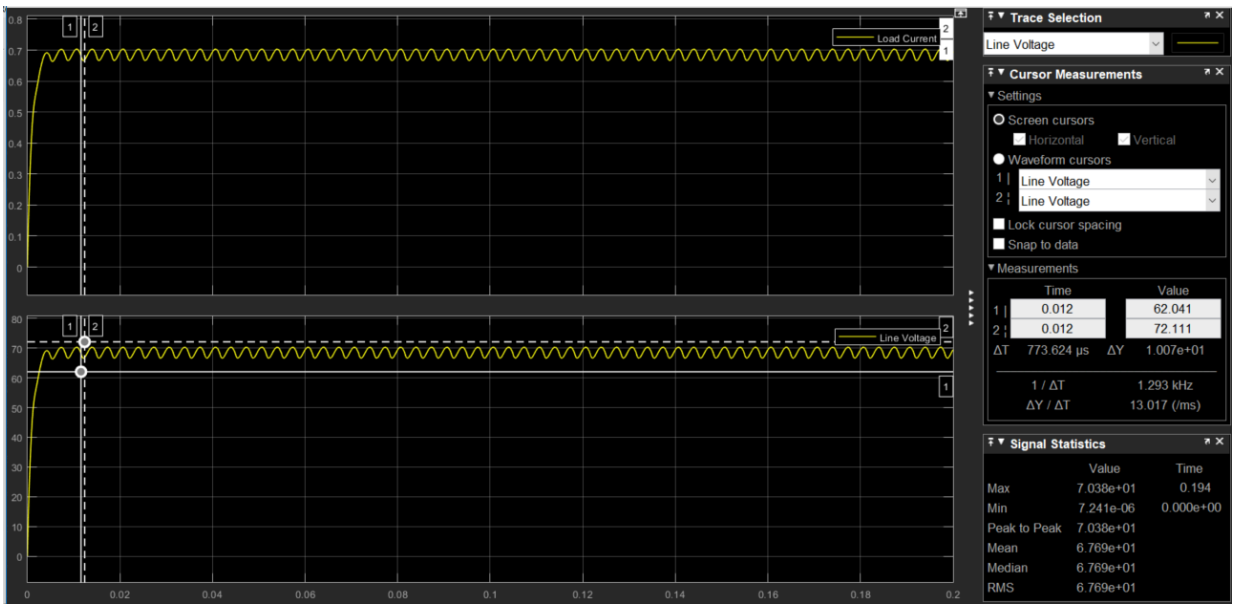


Figure 37. 100mH inductor filter used.

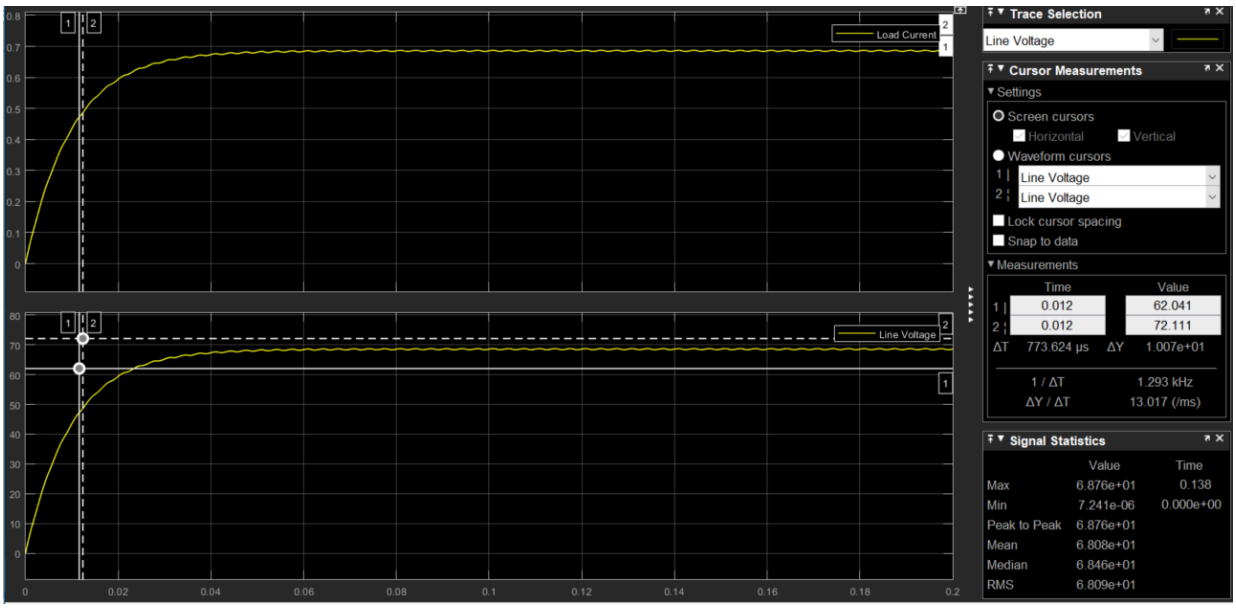


Figure 38. 1H inductor filter used

While the result using a 1H inductor seems ideal, in practice an inductor of that size is likely to introduce some problems with heating and electromagnetic interference. Along with these factors, its size will add to the cost and complexity of the PCB design and manufacture.

For this reason, I found it worthwhile to explore the potential usefulness of the LC (L-Section) filter. As discussed in the literary review, the L-Section filter allows us to increase the size of the capacitor and load without destroying diodes and affecting the output ripple significantly. I also found it allowed me to use generally smaller inductor and capacitor sizes than if I were only using a RL or RC filter.

An LC filter used in figure 39 below. An arbitrary initial value of **1mH** is selected from the middle range of a common local supplier. In Simulink the inductor's series/DC resistance is set to **0.844Ω**, as specified in the 19R105C model shown in the technical datasheet below. Arbitrarily, the **0.6mF** capacitor from previous examples was used.

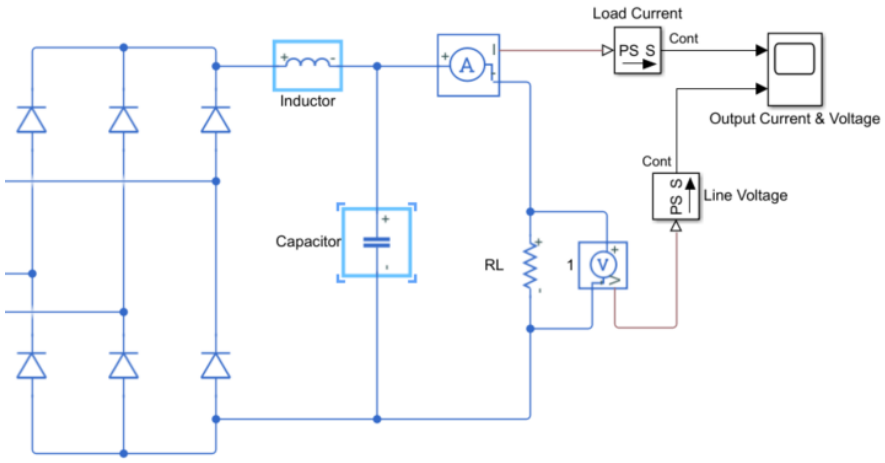


Figure 39. LC Filter

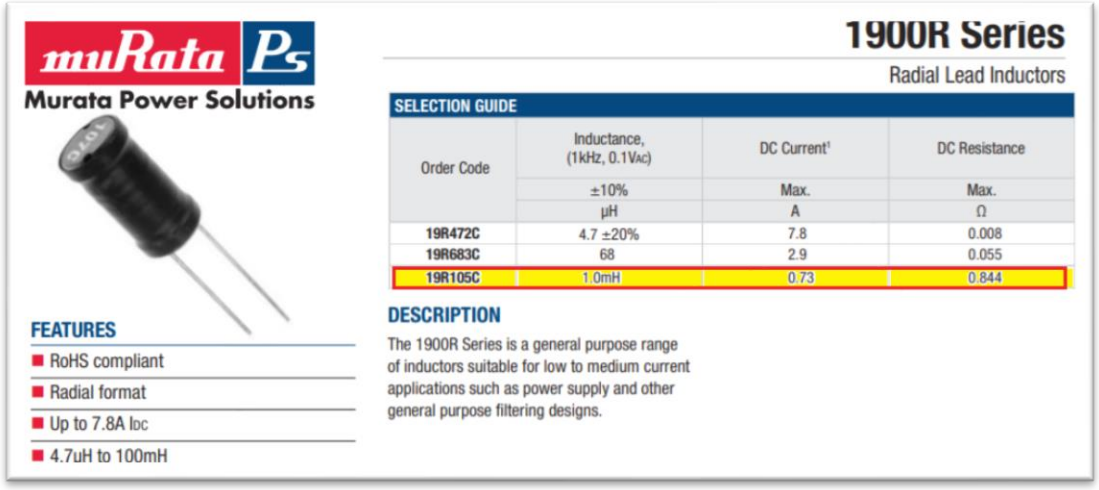


Figure 40. Murata brand general purpose inductors available at UK Farnell (adapted from www.farnell.com, 2018)

Below we see the result of the LC filter applied to the output of the rectifier. There is an 'overshoot' of the voltage at vertical cursor 1 before it settles at a sinusoidal pattern around cursor 2.

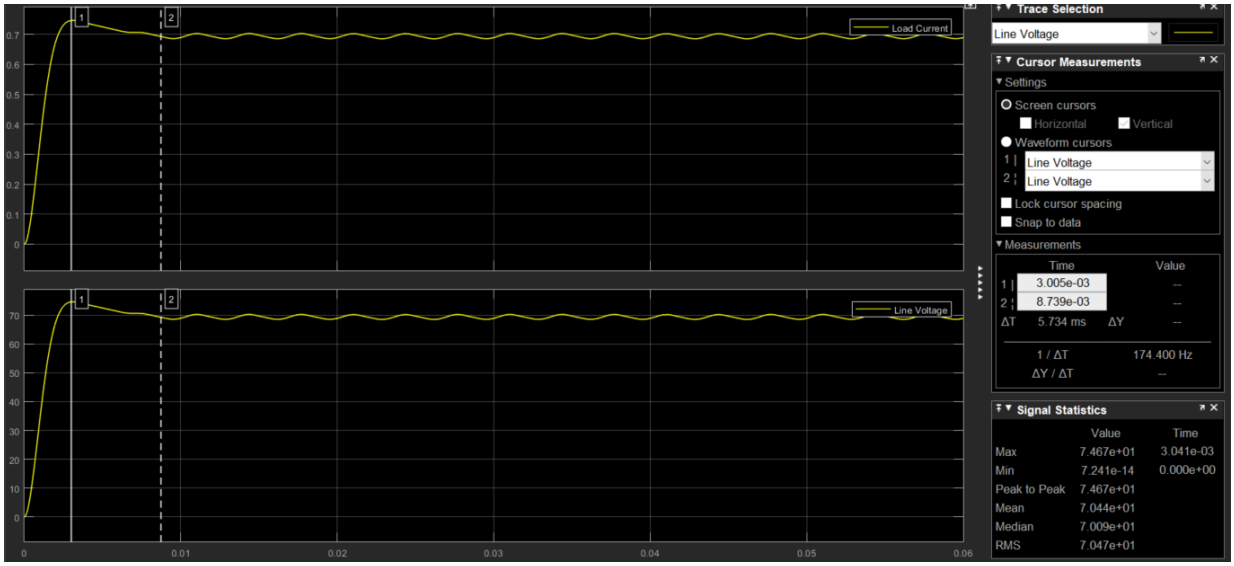


Figure 41. LC filter of 1mH & 0.844 Ω inductor and 0.6mF capacitor.

To eliminate this overshoot and to reduce the settling time, I hypothesised that the transient response of the LC filter had to be **critically damped**. However, the design and analysis of a suitable L-section DC filter prove to be more complex than I initially anticipated and therefore couldn't be completed within the constraints of this project. The

figure below shows the final filter design I used. The reservoir capacitor, C_{fil} is 1mF, inductor L_1 is 4.7mH and capacitor C_1 is 100uF, with the plot in figure 43 showing the resulting smoothed DC.

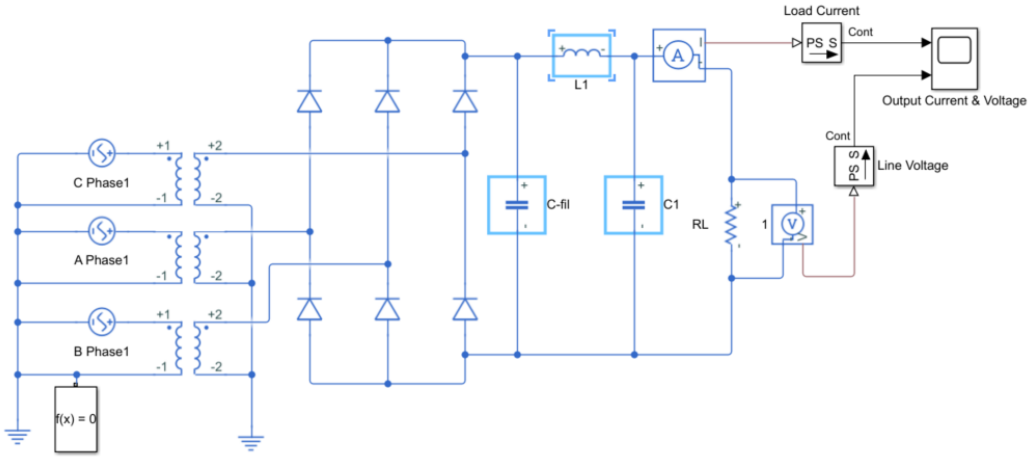


Figure 42. Final filter design.

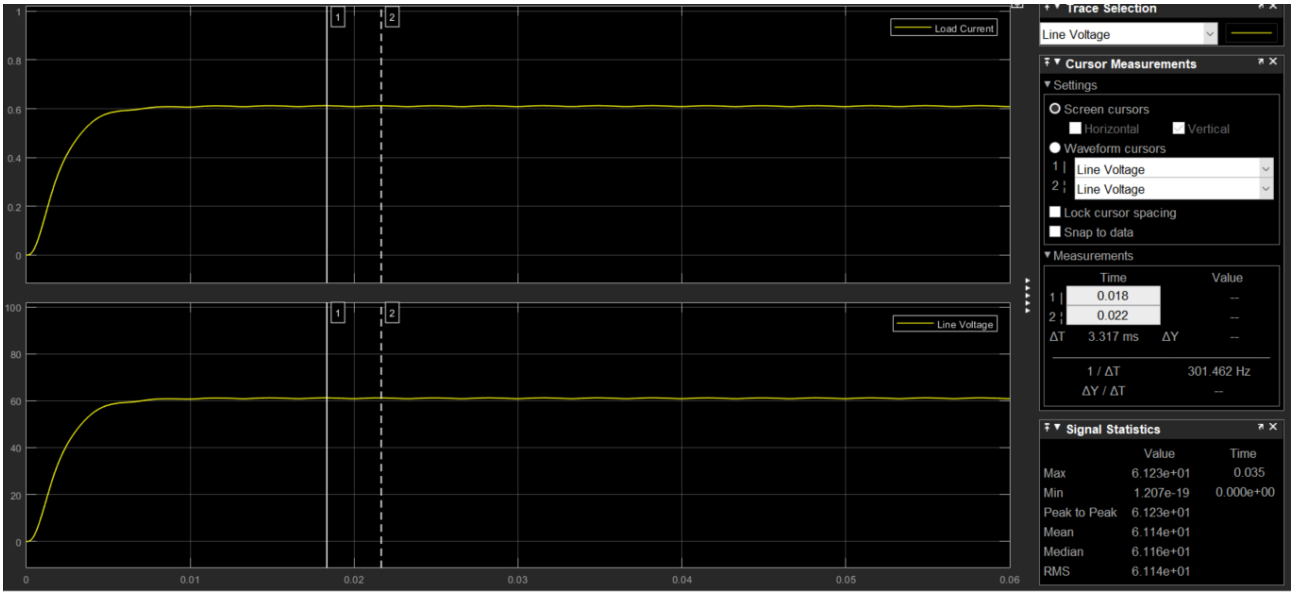


Figure 43. Final filter voltage output waveform.

HARMONIC ANALYSIS

As the designer, one of my main goals is to the **maximum power factor** I can in the system. Harmonic analysis involves resolving power waveforms into their sinusoidal components thereby allowing me to assess distortions that affect the real power output (heating effects). To carry out this analysis I used the Powergui FFT Analysis Tool in Simulink.

When assessing harmonics in the power supply, it is important to note that the rectifier itself acts a load which can cause distortions to the supply current. The first step in my approach was to observe harmonics, if any, a single AC

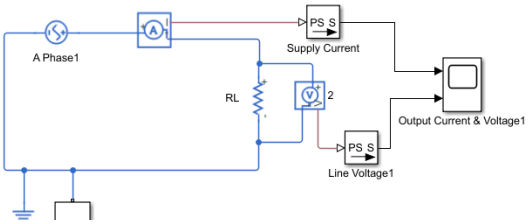


Figure 44. Single phase through a linear load.

phase going through a linear load. Figure 44 shows the model I used to do so. After logging data samples from the simulation to the workspace I was able to observe the supply current in the frequency domain.

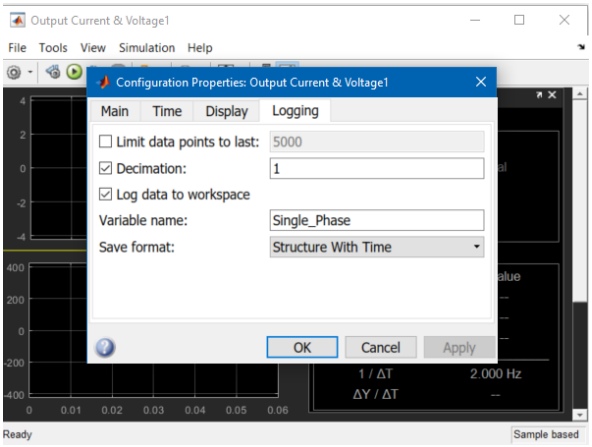


Figure 45. Simulink data logging configurations

The FFT analysis in figure 46 below shows that only the unaltered supply frequency exists through the load. No harmonics are generated from the model which is supposed to represent a linear load.

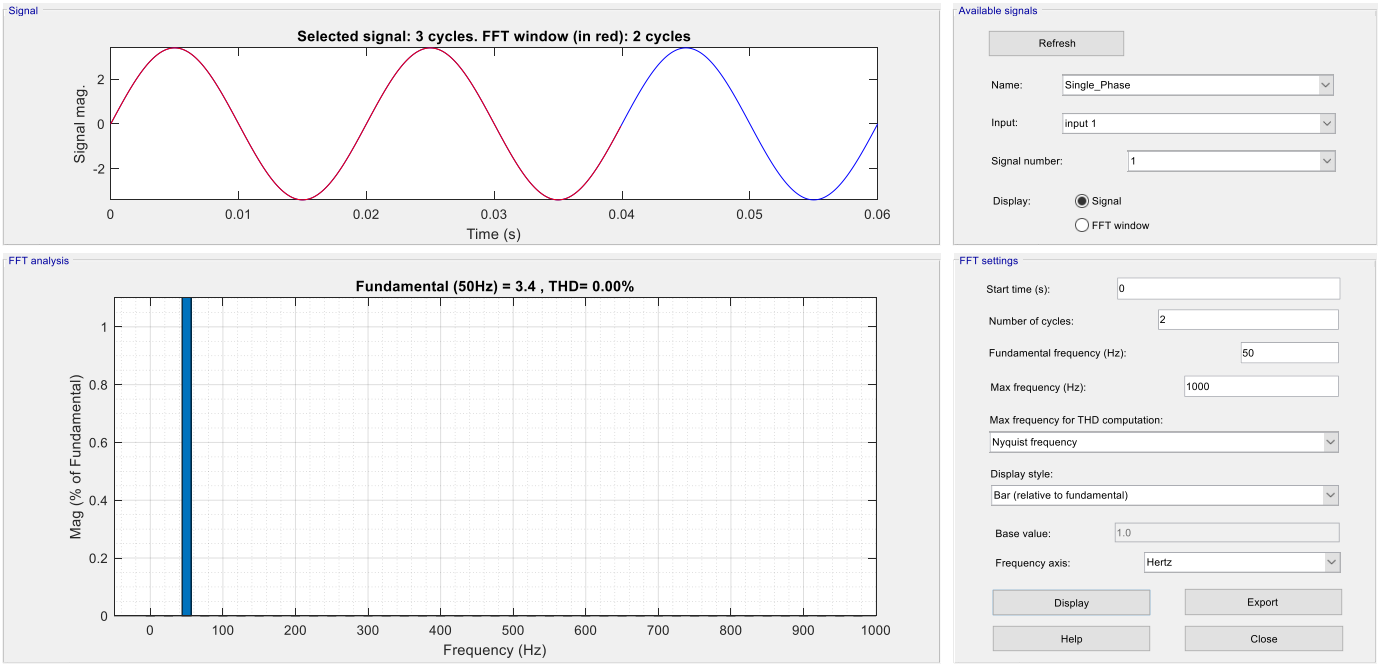


Figure 46. Supply current harmonics for linear load

However, when the single-phase supply is added to a simple bridge rectifier as shown in figure 47 many harmonics are generated as shown in figure 49.

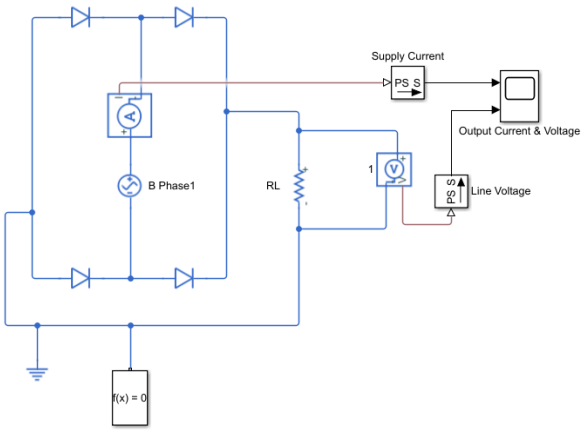


Figure 47.

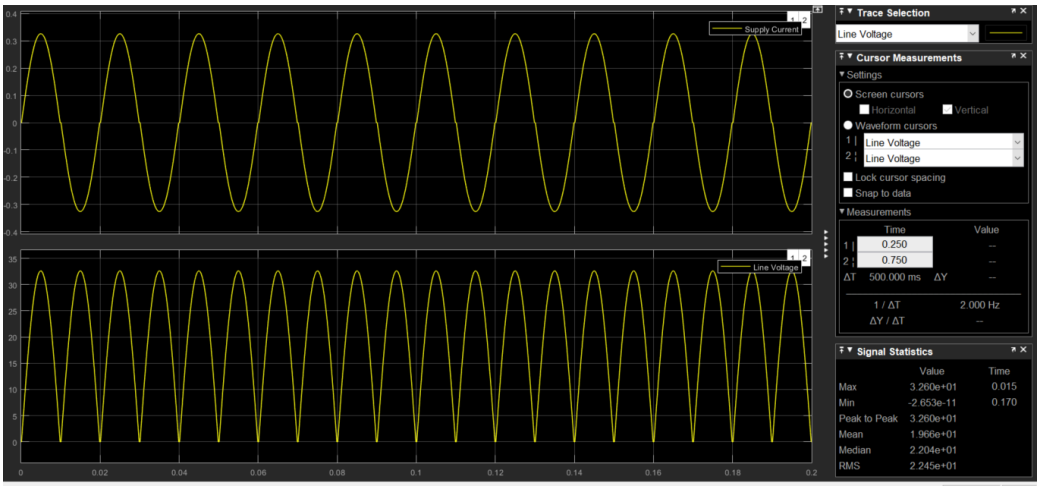


Figure 48.

The figure shows a significant magnitude in ever 8th harmonic.

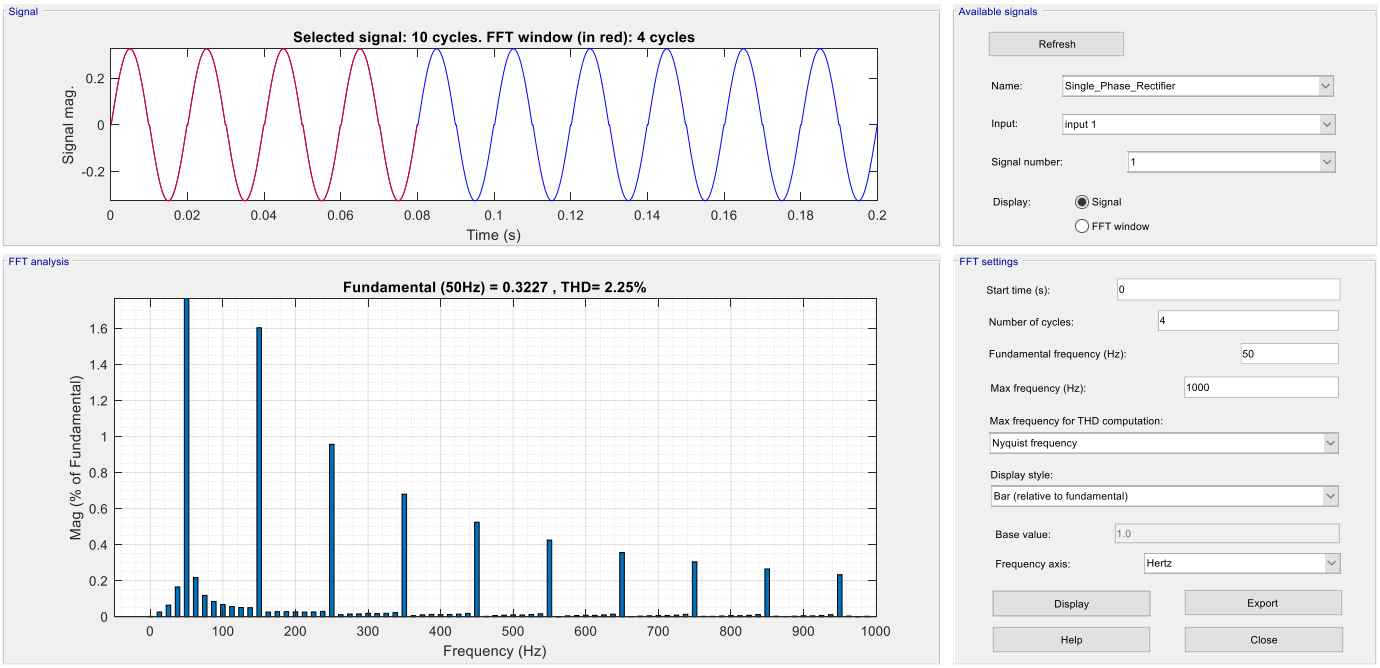


Figure 49. Supply current harmonics

As mentioned previously, it is important to remember that the model utilizes ideal transformers, whereas real transformers contain an inductive reactance. If real transistor models were we can expect further distortions to be present.

In figure 50 below I finally performed an analysis of the diode bridge rectifier.

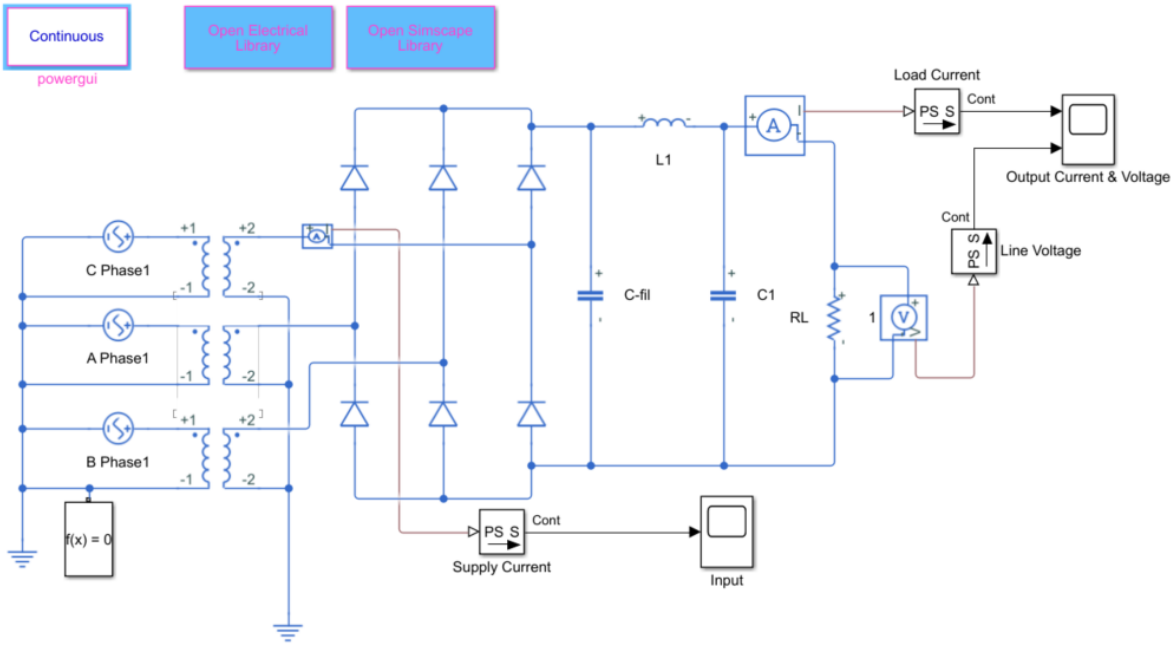


Figure 50.

We see in figure 51 the huge amount of distortions on the fundamental 50Hz current of a single-phase supply. The FFT analysis shows the effect of 6 diode rectifier in creating harmonic distortions. The THD measured is an incredible 401.84%.

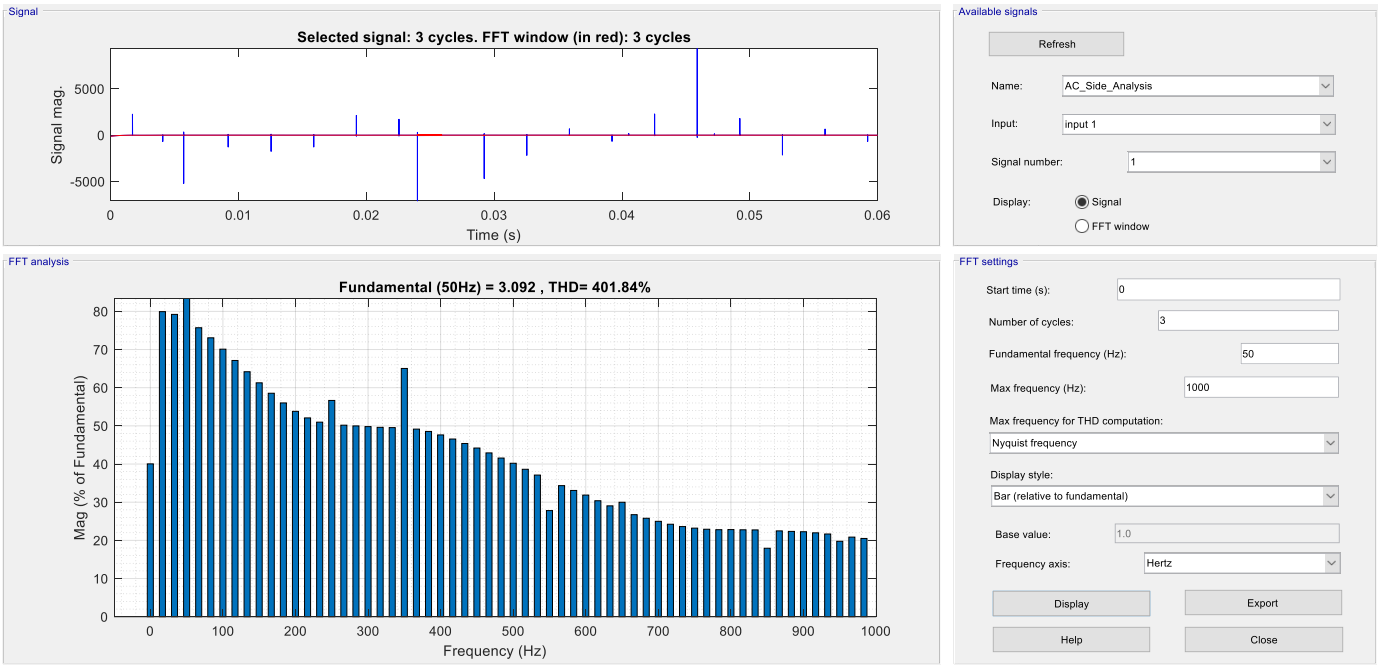


Figure 51. Supply current harmonics affected by rectifier load.

In figure 52 an FFT analysis on the output shows the frequency components in the theoretical DC output after the filter circuit. I say the DC is theoretical because I know it is not perfectly 'smooth'. Notice how the start time in the FFT settings is 0.01 seconds. This allows use to ignore the rise time in the output due to the transient response.

With the bar chart set to display harmonics relative to the DC component, the outputs were fairly predictable. At 300Hz there is a significant frequency component. This is due to the frequency of the six pulse output ripples which is six times higher in frequency than the mains supply at 50Hz.

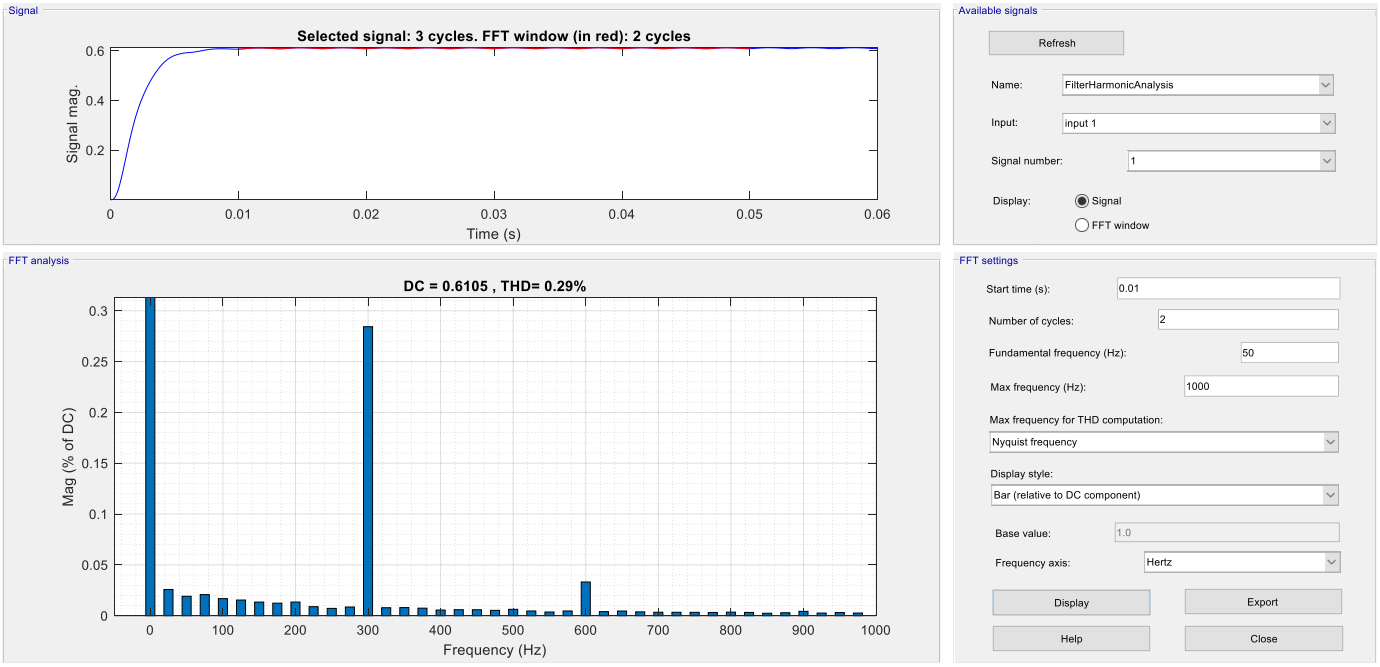


Figure 52. Rectifier output harmonics after transient effects

DIGITAL DESIGN METHODOLOGY

DIGITAL CONTROLLER DESIGN APPROACH

One early decision digital system designers need to make is whether to use an FPGA or Integrated Circuit (IC). For myself the decision was simple. I used an FPGA because it is much more suitable for prototyping. It allows designers like myself to change the program design several times for hardware testing without incurring significant costs. It is cheaper and has no manufacturing delay time.

FPGA Design Flow

My FPGA-based design approach involved defining the product specification, its logic coding and the layout of the physical design. I primarily used a top-down approach. With the top-down approach I first determined the functional behaviour I wanted in my system. Through this approach I expanded on intricate details as the design progressed.

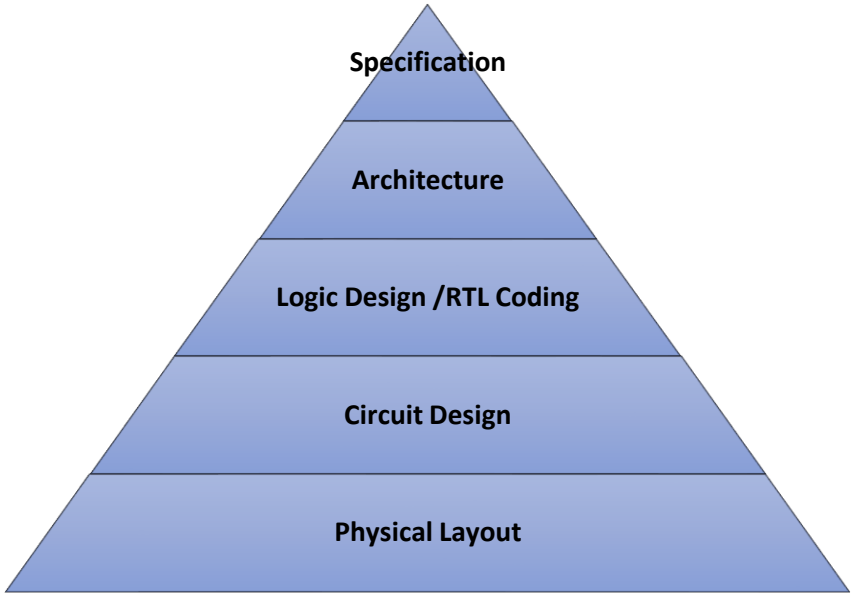


Figure 53. Top-down design flow

SYSTEM SPECIFICATION

The system operation involves identical processes for 3 phases which are delayed by 120° due to external signals (ZCDs). The pseudo code below describes the operation for one phase. The pairs of SCRs across each phase is derived from the commutation of the rectifier, simplified from the graph in the figure below.

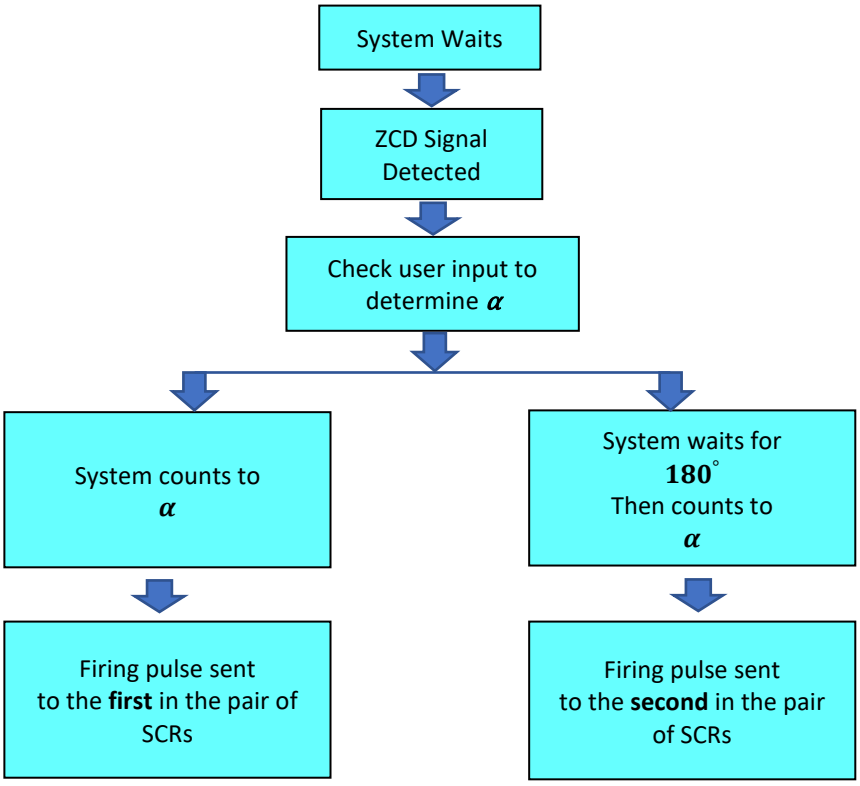


Figure 54. Logic behavious for an individual phase

The pairs associated with each phase are as follows, each separated by 180°;

Phase A (red) – F1 followed by F4
Phase B (yellow) – F3 followed by F6
Phase C (blue) – F2 followed by F5

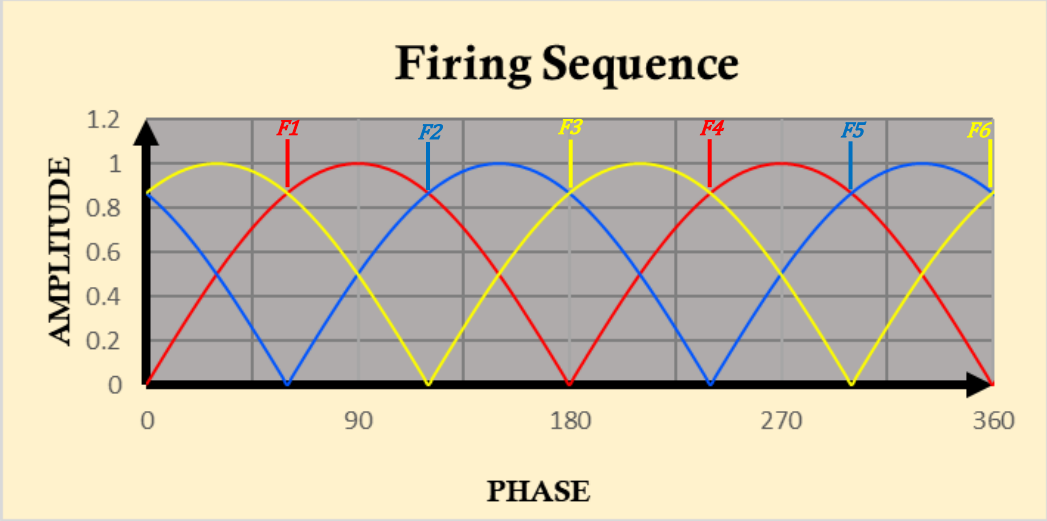


Figure 55. Phases and their associated SCR firing pulses.

LOGIC DESIGN AND ARCHITECTURE

The development of my logic design was a complex process involving many dozens of cycles of trial and error, debugging of syntax errors and over one hundred behavioural simulations. The figure below is a fair representation of the process.

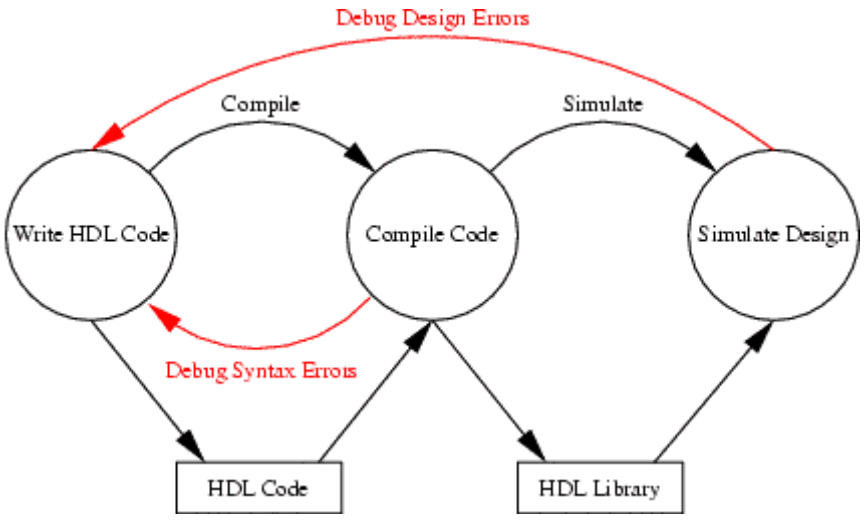


Figure 57. Logical design cycle (Tampere University, 2015)

PROGRAM EXPLANATION

The program has eleven main ports however in the source code other output ports can be found commented out that were used for debugging. The main ports are the reset and clock ports, RST and CLK, the three zero crossing detector inputs ZCD_A, ZCD_B and ZCD_C and six thyristor firing ports F1 to F6.

In addition to the ports there are some special signals used in the program. *Alpha* is the firing angle which, for a thyristor varies from 0^{o} to 180^{o} or 0 to π radians. However, from the figure below, we can see that the firing angle in a 3-phase bridge rectifier *alpha* varies from 0 radians to a maximum value of $2\pi/p$ radians.

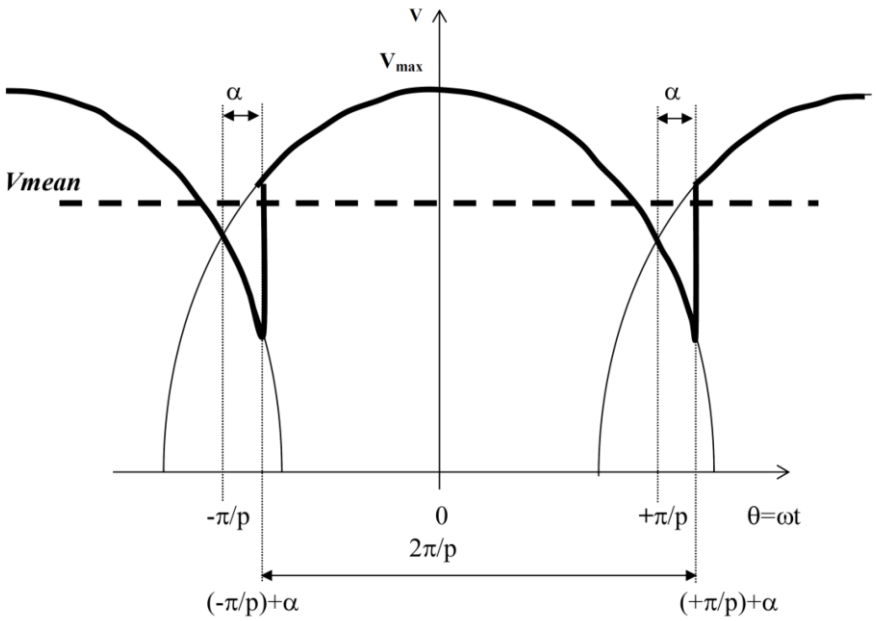


Figure 58. Firing angle for p-pulse rectifier (Anglia Ruskin University, 2018)

We know that the rectifier will produce a 6-pulse pulsating DC output. Therefore, the maximum value of alpha, α_{max} will be $^{2\pi}/_{6}$ or $^{\pi}/_{3}$. With a phase frequency of 50Hz, the value of α_{max} in seconds is given by;

$$\alpha_{max} = \frac{1}{6} \times T = \frac{1}{6} . \frac{1}{50Hz}$$

$$= 3.33^{\cdot} \times 10^{-3} \text{ seconds}$$

For experimental purposes I initially chose to work with a firing angle, α, one third of α_{max} or 20°. Therefore,

$$\alpha = \frac{1}{3} . \alpha_{max}$$

$$\alpha = \frac{1}{3} . 3.33^{\cdot} \times 10^{-3}$$

$$\boldsymbol{\alpha = 1.11^{\cdot} ms}$$

However, in this program we measure *alpha* in clock pulses. If the period of the main system clock is 10ns, then the number of clock pulses required for *alpha* is given by;

$$Alpha\ in\ clock\ pulses = \frac{Firing\ Angle\ [s]}{Clock\ Period\ [s]} = \frac{1.11^{\cdot} \times 10^{-3}}{10 \times 10^{-9}} = \boldsymbol{111,111\ pulses}$$

Alternatively, when the clock period is 10ns, the following simple formula can be used for the quick conversion of the firing angle in degrees (°) to clock pulses;

$$Alpha\ in\ clock\ pulses = \frac{\frac{Apha\ [°]}{360} \times \frac{1}{F}}{10 \times 10^{-9}}$$

This value is now applied to the signal *alpha* in the program architecture.

```
signal alpha : integer := 111111 ;
```

The signal HCSS for a given phase is the *Half-Cycle Selector Switch*. The start of a new half-cycle in the supply current is indicated by a change in the state of the HCSS, between '0' and '1'.

```
signal HCSS_A : std_logic :='1';
```

The Fire Count signals tells how many times the ZCD pulse has been detected. This is useful for debugging and also later on in the firing processes to stop certain outputs (F1 to F6) from being fired before a ZCD is even detected.

```
signal A_Fire_Count : integer range 0 to 1999999 := 0;
```

The Phase and Phase_Delayed signals are counters which indicate the current phase of the first half-cycle and the second half-cycle respectively. This count then allows for measuring/timing the size of the firing angle, *alpha*.

```
signal A_Phase : integer range 0 to 1999999 := 0; -- This is the phase of the FIRST half-cycle.
signal A_Phase_Delayed : integer range 0 to 1999999 := 0; -- This is the phase of the SECOND half-cycle.
```

The HCSS Trigger processes toggle the state of the HCSS signal which is used to determine which half-cycle is active and therefore fire the correct thyristor. The start of each half-cycle triggers a ZCD pulse represented with the Boolean state '1'. If the ZCD is connected to the very first sinusoidal output of the power supply, its output pulse '1' is at 0 degrees, the beginning of the first half-cycle which is associated with the positive current firing the first thyristor being conducted for that phase. The second pulse is at 180 degrees, the beginning of the second half-cycle, which is associated with the firing of the second thyristor being conducted for that phase.

```
HCSS_A_Trigger :
process (ZCD_A)
begin
   if (ZCD_A = '1') then
      HCSS_A <= not HCSS_A;   -- Toggle state
   end if;
end process;
```

Once the HCSS change has been detected, its new state initiates the related phase angle counter in the Phase Counter Processes. A HCSS value of '0' indicates that the supply voltage is in its **first half-cycle**.

```
A_Phase_Counter :
process (clk, HCSS_A)
begin
   if rising_edge(clk) then
      if (HCSS_A = '0') then  -- first half-cycle has begun
         A_Phase <= A_Phase + 1; -- Increment counter
      else
         A_Phase <= 0;         -- A_
         end if;
   end if;
end process;
```

Likewise, in the Delayed Phase Counter processes, the detection of a HCSS state change initiates the phase angle counter. A value of '1' indicates that the supply voltage is in its **second half-cycle**.

```
Delayed_A_Phase_Counter :
process (clk, HCSS_A)
begin
   if rising_edge(clk) then
      if (HCSS_A = '1') then
            A_Phase_Delayed <= A_Phase_Delayed + 1;
      else
         A_Phase_Delayed <= 0;
         end if;
   end if;
end process;
```

The Detect ZCD processes count the number of ZCD crossings detected in a given phase.

```
DETECT_ZCD_A: process(ZCD_A)
  begin
    if rising_edge(ZCD_A) then
      A_Fire_Count <= A_Fire_Count + 1;   -- Counts the number of times ZCD is detected.
      end if;
  end process;
```

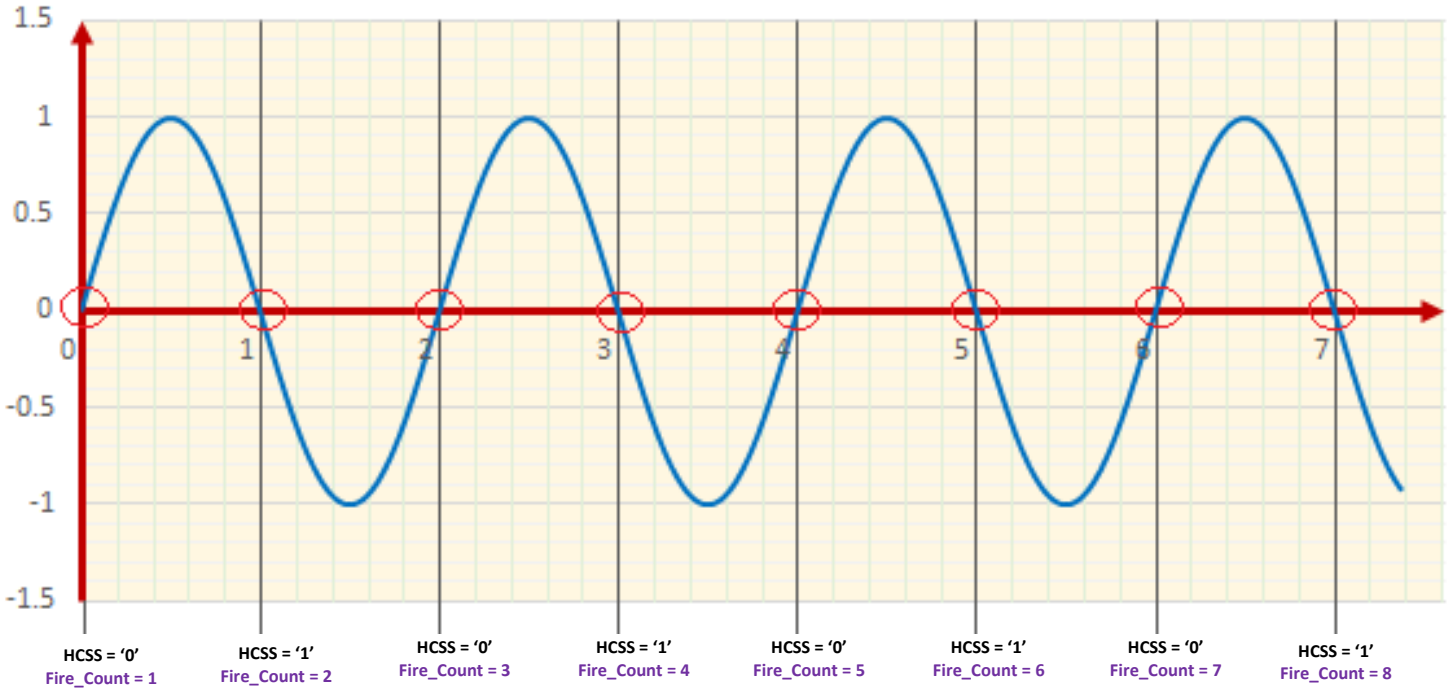


Figure 59. Behaviour of HCSS and Fire_Count signals for a given phase.

The TGC Duty processes are the main pulse firing algorithms. They measure the phase angle and then fire (output a '1') the thyristor via the relevant port (F1 to F6) when the phase angle has reached alpha. However, a problem arises. The phase angles are measured from the zero crossing with the horizontal axis, but alpha is measured from the intersection of any between the waveforms of the current phase and the previous one. I need to be able to measure alpha from the zero crossing.

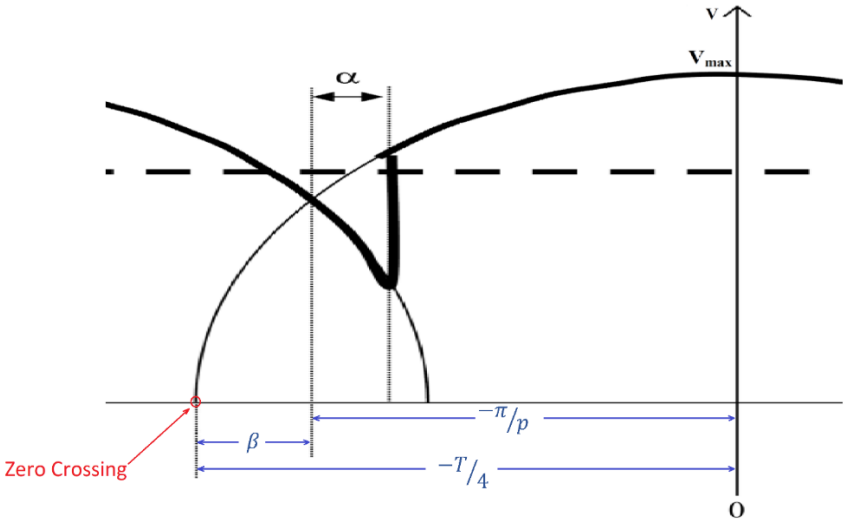


Figure 60. Relationship between the firing angle and zero crossing.

I therefore introduced the parameter β, which is gives the delay after the zero crossing before α can be measured. To determine the value of β I used the vertical axis through V_{max} as a reference point. By geometry the time between the zero crossing and V_{max} is half the duration of the half-cycle or quarter the duration of the full period, T.

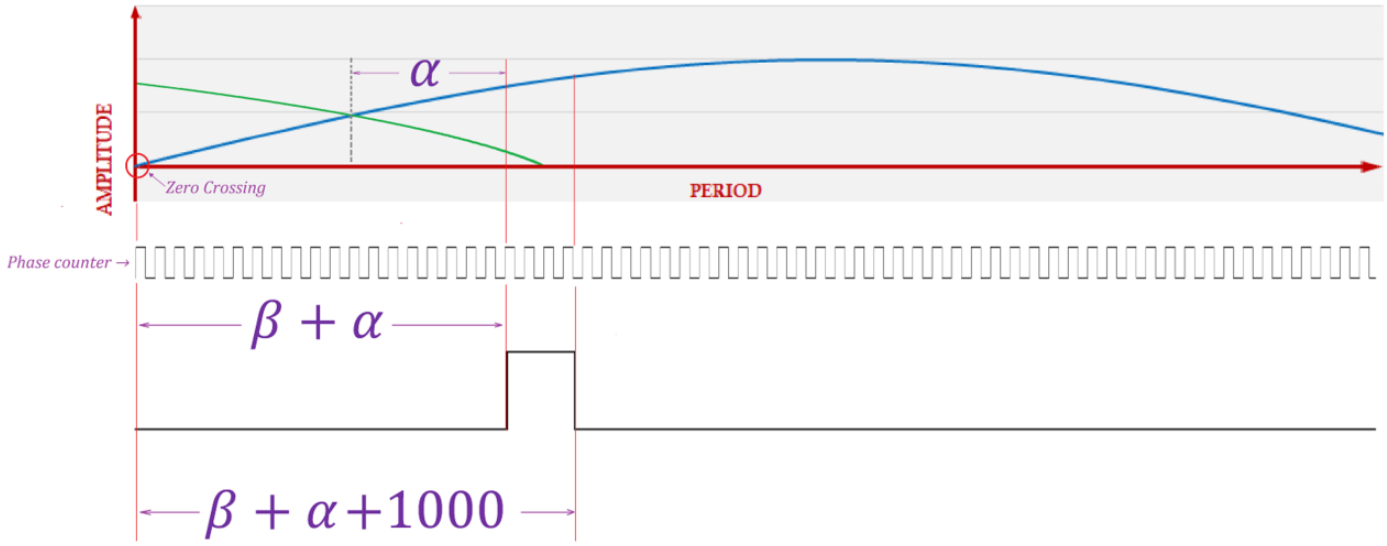


Figure 61. Pulse firing timing.

We know that the start of alpha is π/p from the peak voltage, V_{max}. Therefore, β is given by;

$$\beta = \frac{T}{4} - \frac{\pi}{p}$$

$$= \frac{0.02}{4} - \frac{0.01}{6} = 0.005 - 0.001667^{\cdot}$$

$$\beta = 3.33^{\cdot} ms$$

$$\beta \; in \; clock \; pulses = \frac{3.33^{\cdot} \times 10^{-3}}{10 \times 10^{-9}}$$

$$\beta \approx 333{,}333 \; clock \; pulses$$

Now with the timing between alpha and the zero-crossing known, the firing angle and pulse with can be achieved.

The TGC Duty processes are operated in pairs of processes for firing the two thyristors associated with a given phase. The second process of each pair sends its firing pulse 180° after the first process. In the examples shown below, the pairs **TGC_F1_DUTY** and **TGC_F4_DUTY**, fire the thyristors F1 and F4 which are both associated with the A-phase of the 3-phase current.

```
TGC_F1_DUTY : process(RST, CLK, A_Phase)   -- Fires pulse with given duty cycle.
  begin
      if (CLK'EVENT AND CLK='1') then
        if (A_Phase < (333333 + alpha))  then -- Before start of firing pulse: Don't fire until the given firing angle.
          F1_pulse <= '0';
        else if (A_Phase > (433333 + alpha)) then -- After end of the firing pulse: The difference between 433,333 and 333,333 give the width of
the pulse .
          F1_pulse <= '0';
        else if (A_Fire_Count > 0) then -- Once the first ZCD signal is detected, the firing pulse is sent (after 333333 + alpha,
                        -- but until 433333 + alpha
          F1_pulse <= '1';
        end if;
      end if;
      end if;
  end if;
end process;
```

```
TGC_F4_DUTY : process(RST, CLK, A_Phase_Delayed, A_Fire_Count)   -- Fires pulse with given duty cycle.
   begin
      if (CLK'EVENT AND CLK='1') then
         if (A_Phase_Delayed < (333333 + alpha))  then
            F4_pulse <= '0';
         else if (A_Phase_Delayed > (433333 + alpha)) then
            F4_pulse <= '0';
         else if (A_Fire_Count > 0) then
            F4_pulse <= '1';
         end if;
      end if;
      end if;
end if;
end process;
```

In reality, the pulse widths must be shaped according to the requirements of the thyristor, specified by its manufacturer.

BEHAVIOURAL SIMULATION

The simulation set sim_1 has the testbench file TGC_Firing_tb. The simulation settings used are unique to this simulation set and may be changed for other simulation sets. For behavioural simulation the default run time was 65ms, however this was reduced to 45ms for post synthesis timing simulations. A 45ms post synthesis timing simulation gives a full cycle of six firing pulses, F1 to F6. Notice this begins with F1 only after 20ns as the program code does not fire the F2 pulse in the first cycle (0ns – 20ns). At approximately 45ns a full cycle of 6 pulses has already been fired and is starting to be repeated with the firing of F1.

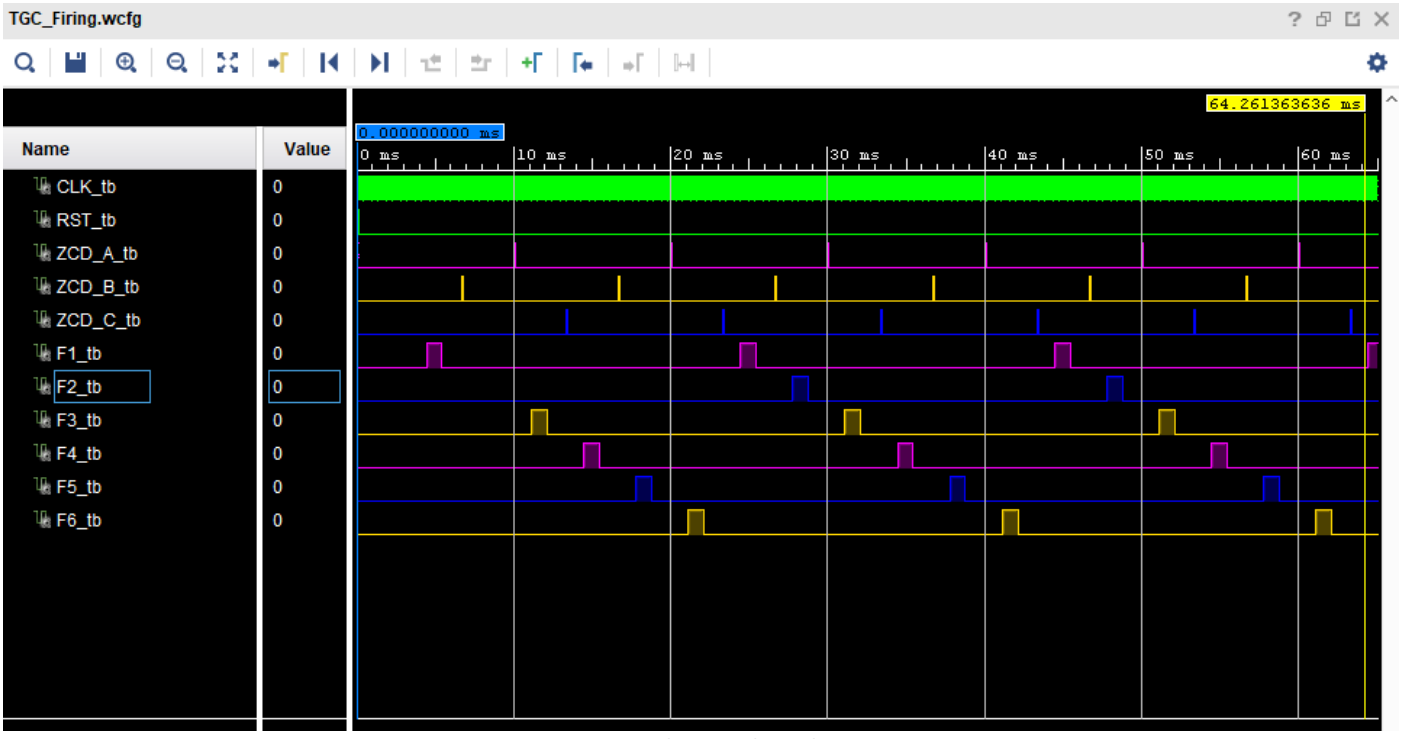


Figure 62. Behavioural Simulation

After ZCD_A changes state (‘0’ to ‘1’), it takes 10ns before the half-cycle counter can be updated on the next clock pulse, then it takes another 10ns before the F1 pulse is switched off on the next clock pulse. It therefore takes a total of 20ns for the F1 firing pulse to switch off, AFTER the ZCD pulse switches.

Figure 63. Post implementation timing simulation

CIRCUIT DESIGN AND PHYSICAL LAYOUT

I/O PIN PLANNING

Before timing analysis can be carried out the connectivity between the inputs and outputs of the FPGA and the PCB must be defined. In the industry, this process uses the expertise of system designers, PCB designers and FPGA engineers. Some of the things I wanted to achieve in the pin planning process include;

- Shortening signal trace lengths.
- Avoid crossing of signals.
- Maintaining high speed signal integrity.
- Using configurations that are compatible with other potential future devices.
- Identifying power and ground sources on the PCB

Vivado makes it possible to perform pre-RTL I/O planning. However, for my project I performed I/O from the elaborated RTL design. Using this method I can choose to use the IP cores from the Vivado IP catalogue and block diagrams from the IP integrator. The Vivado tools from the elaborated design also provide Design Rule Checks (DRCs) that make it easier to automate the checking of port assignments, I/O standards and other design specifications.

Initially I used the XDC constraints file to setup my I/O connections. However, to further investigate the features of the FPGA, I made use of the I/O planner, in the synthesized design. The I/O planner gives a visual representation which helps in defining I/O pin placement constraints. It graphically displays pins, die pads and I/O banks.

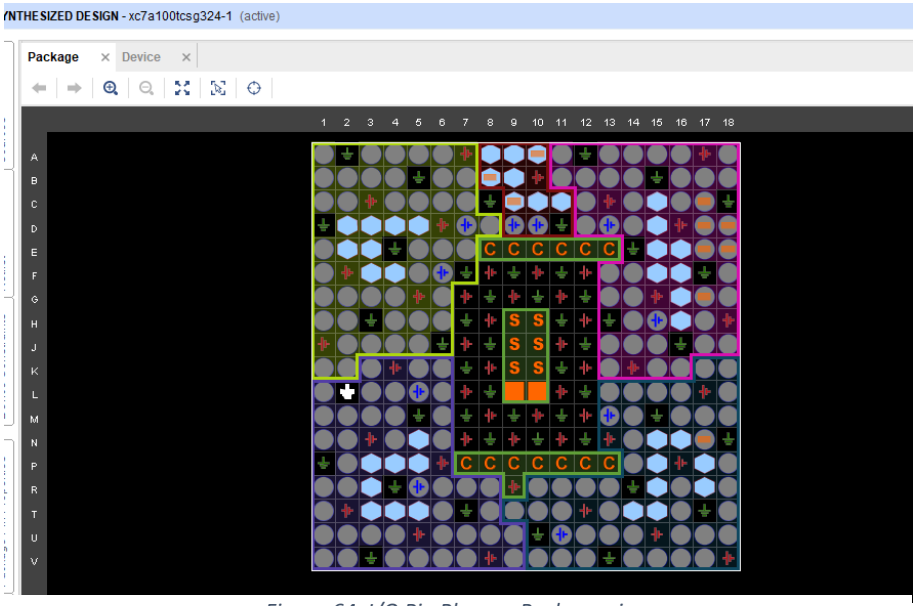


Figure 64. I/O Pin Planner Package view

Design Rules Checks from the I/O pin planner allowed me to perform error checking on the pin layout based on I/O assignment rules. The planner also gave me the facility to group related pins as shown below. However, I had no reason to perform any manual changes to the automated groupings.

Viewing the Device and Package views, I was able better understand the I/O bank relationship with the design's logic. When I select pins or banks in one view, the corresponding pins or banks were highlighted in the other view. I could also assign I/O ports from either view. This way I could see if the pins assigned in each bank meet the I/O banking rules and if they are grouped appropriately.

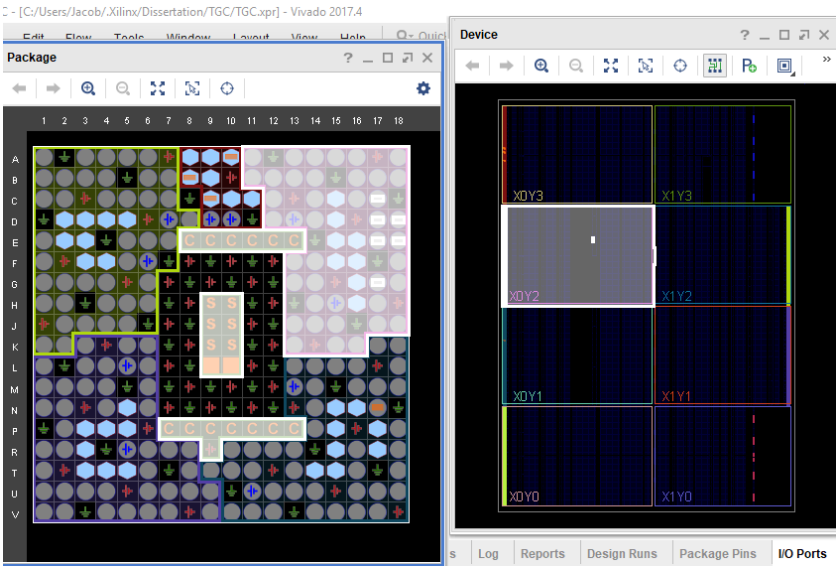


Figure 65. I/O Package view and corresponding device view.

When I select a pin in the Package view, it is highlighted in the Package Pins table which then allows me to see its properties such as the bank it's been assigned, the ports of logic program assigned to it, whether it's the positive or negative of a differential pair of pins, voltage standards and so on.

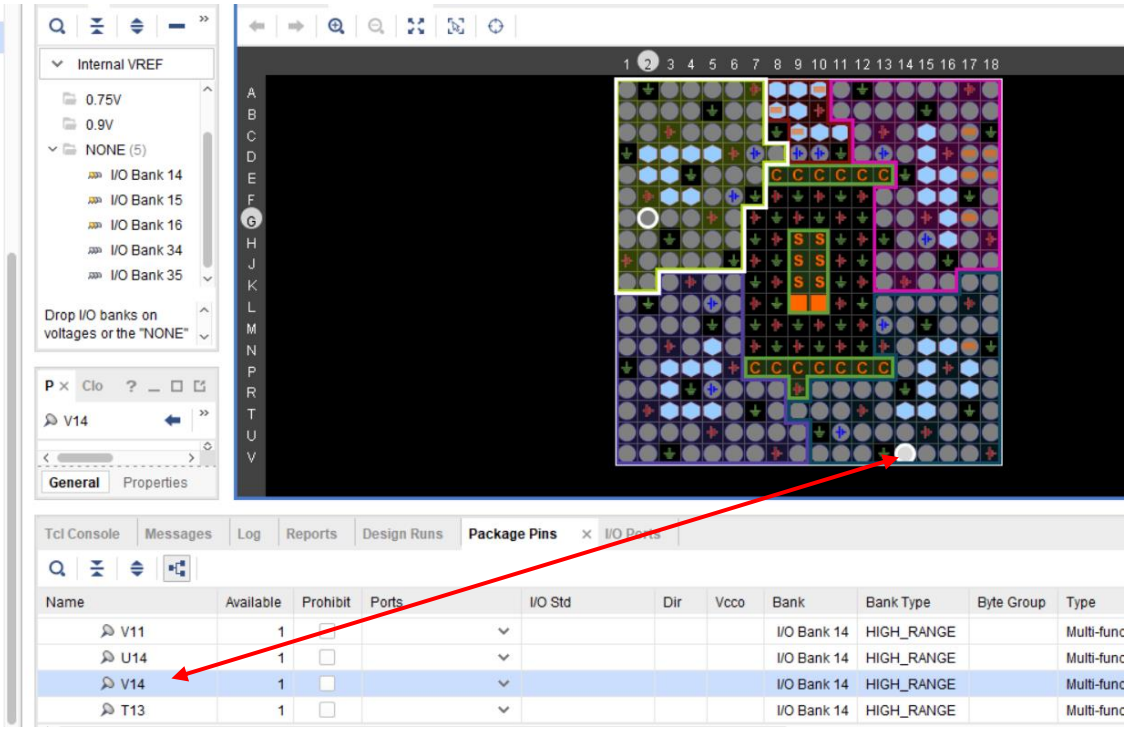


Figure 66.

A closer look at the Package view allowed me to see the differential pairs and clock capable pins. This was quite important as it helped in resolving an implementation problem I later encountered.

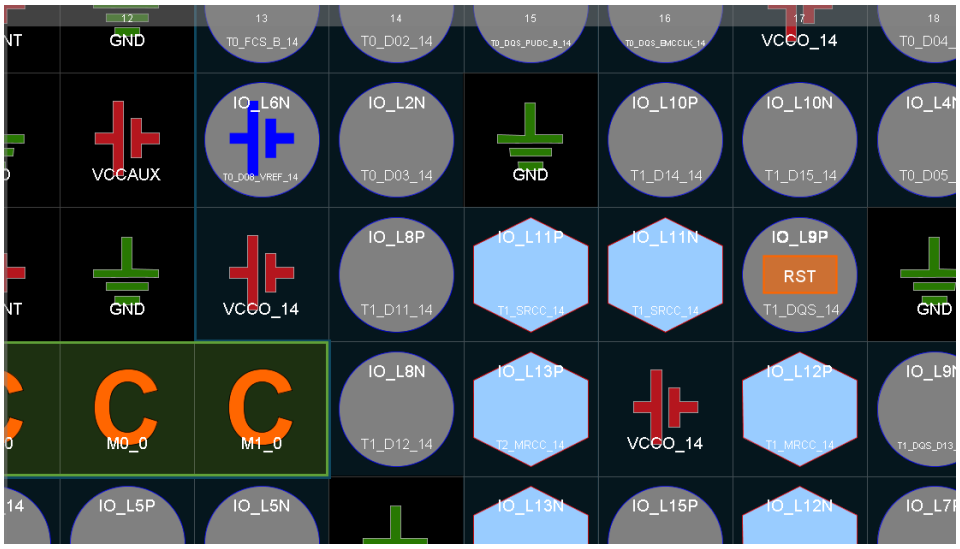


Figure 67.

Once I placed my completed pin assignments I ran a DRC check. The figure below shows the first DRC report generated.

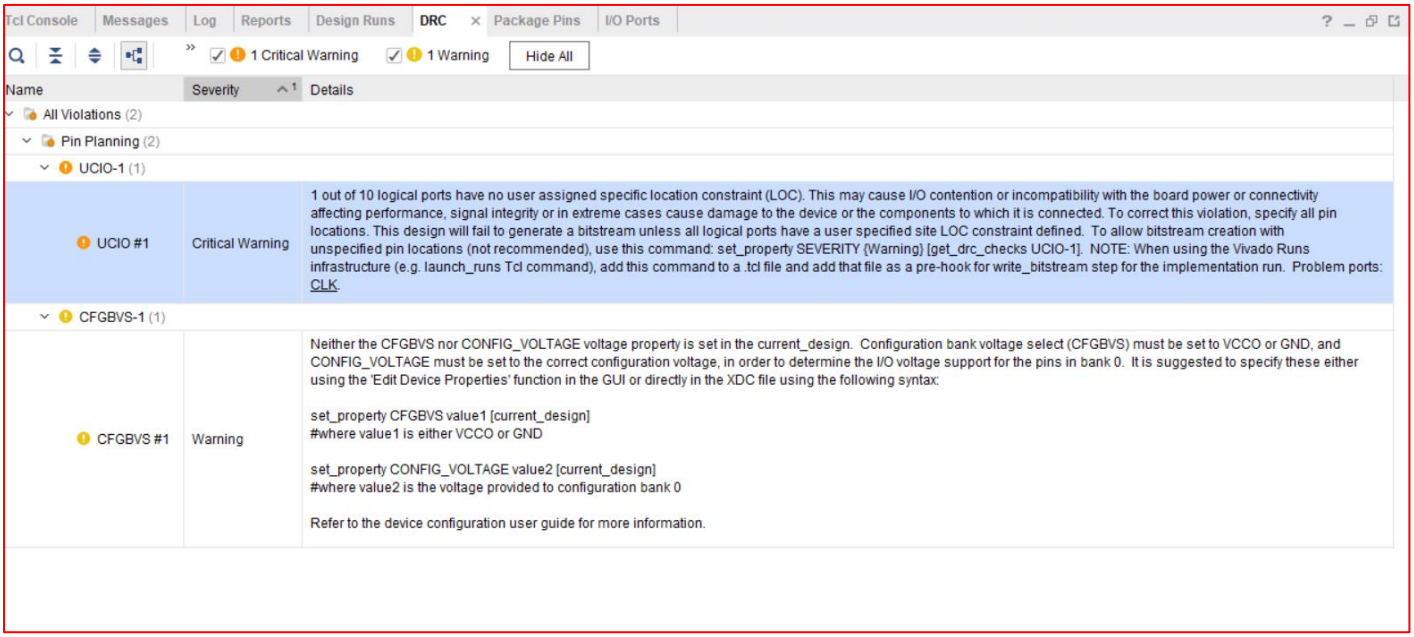


Figure 68. DRC warnings

The first warning, is identified as a critical one. It is saying that there is an unassigned pin. Simply put, there is a port that is not physically connected to anything. According to the design rules, this is not acceptable and will not generate a bitstream for programming the FPGA. The warning specifically identifies the CLK port.

If the port (CLK in this case) was indeed connected to the correct package pins, this could be deemed an incorrect DRC warning, and which could be overwritten/downgraded using the Tcl command;

set_property SEVERITY {Warning} [get_drc_checks UCIO-1]

However, the report was indeed correct as a quick look at the I/O Ports window in the synthesized design showed that the CLK signal was not assigned a package pin. Nexys 4 documentation indicates that a 100MHz crystal oscillator is connected to pin 3, so this was assigned to the CLK port. The second warning states that the configuration bank voltage (CFGBVS) select wasn't set for the design. This was to be expected as the value is empty by default. The CONFIG_VOLTAGE property was also not set. To remedy these two issues, I added the following tcl commands to the XDC file.

set_property CFGBVS VCCO [current_design]

set_property CONFIG_VOLTAGE 3.3 [current_design]

Taking these steps mentioned removed the DRC violations.

TIMING ANALYSIS

I performed a timing analysis to identify the signal path that causing the maximum delay.

Clock Networks Report

Unless stated otherwise, Vivado IDE assumes all clocks are related (by default). The ***set_clock_groups*** command could be used to stop the timing analysis from occurring between groups of clocks. However, this was not necessary in this project as there were no defined groups of clocks.

Using the clock network report, I was able to visualise the topology of the clocks used in my design as shown in figure 69 below. The tree view shows the clocks from I/O ports to the load, the network fanout of the clock nets, and a hierarchical view of the primary and generated clocks in the design. The report also shows the number of loads that are driven by each clock.

Figure 69. Report Clock Networks from synthesized design.

IBUF shown above is a differential input buffer. From IBUF we can see that the clock is propagating to the various firing pulse registers (F1 to F6), the phase registers (A_Phase, B_Phase and C_Phase) and the phase-delayed registers (A_Phase_Delayed, B_Phase_Delayed and C_Phase_Delayed).

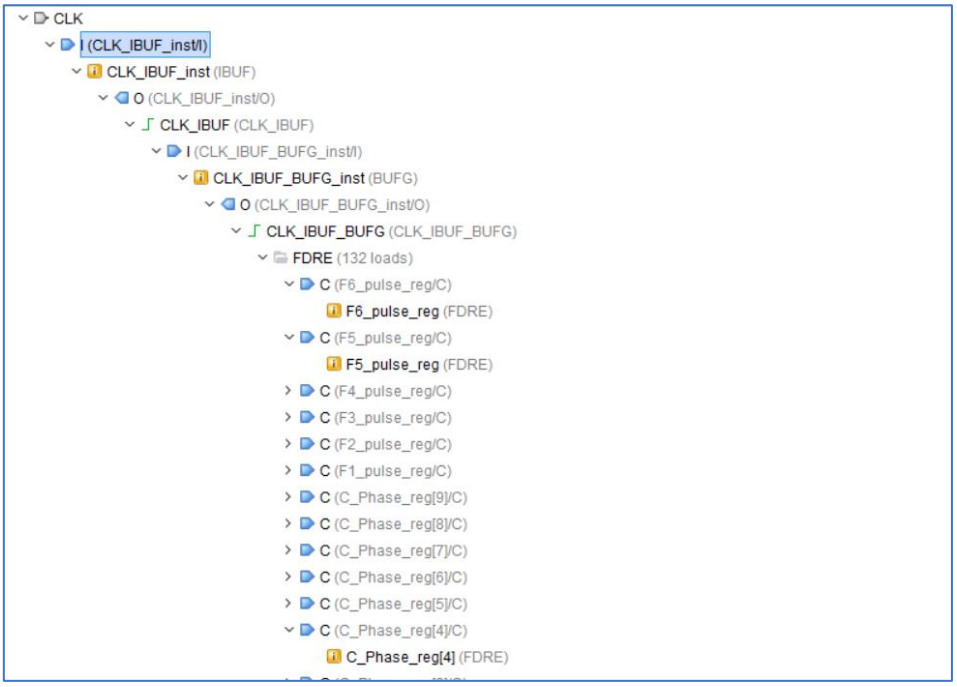


Figure 70.

From the clock network report (figure 69) we can see that the primary clock CKL and the ZCD_A, ZCD_B & ZCD_C signals (not really clocks) are treated as being asynchronous clocks because they are generated by separate (external) sources.

Creating Clocks

The only **true** primary clock in this system is CLK, the master clock coming from the Nexys 4 board. Whereas, ZCD_A, ZCD_B & ZCD_C are not clocks at all, but input signals that I expected to have a set period/frequency. However, in order to perform a post implementation timing analysis, these inputs along with the CLK input were assigned timing constraints in the XDC file as shown in Figure 70 below.

```
264
265 create_clock -period 10.000 -name CLK -waveform {0.000 5.000} [get_ports CLK]
266 create_clock -period 10000000.000 -name ZCD_A -waveform {0.000 1000000.000} [get_ports ZCD_A]
267 create_clock -period 10000000.000 -name ZCD_B -waveform {6666666.000 7666666.000} [get_ports ZCD_B]
268 create_clock -period 10000000.000 -name ZCD_C -waveform {13333333.000 14333333.000} [get_ports ZCD_C]
```

Figure 71. Creating clocks in XDC file.

The above commands simulate a master clock with time period 10ns and three zero-crossing detector input signals 1ms long or 1 million nanoseconds long that are separated by 6.66·ms, the equivalent to 120 degrees phase shift in a 50Hz signal.

Though this now simulates pulses of a set frequency coming from the external zero crossing detector, is important to remember that **in reality there may be some phase delays due to harmonics in the power supply**. Compensating for this would require more complex algorithms that would measure the phase delay then attempt to compensate for this in the firing logic.

I could have also created the clocks with phase offsets using the *create_clock* command or from the *Create Clock* option in the Timing Constraints window. This provides an alternative way of creating the ZCD signals for post implantation simulation.

The figure below shows the Create Clock window that would allow us to do this. *Source object* allows us to connect this new clock to an object already present in the program, such as ZCD_A.

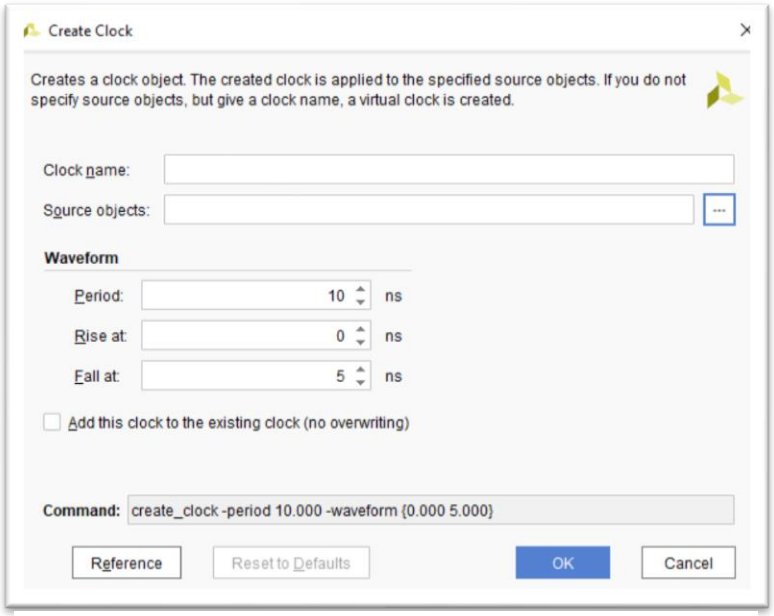


Figure 72. Create clock wizard.

Alternatively, the clocks can be written from the timing constraints window

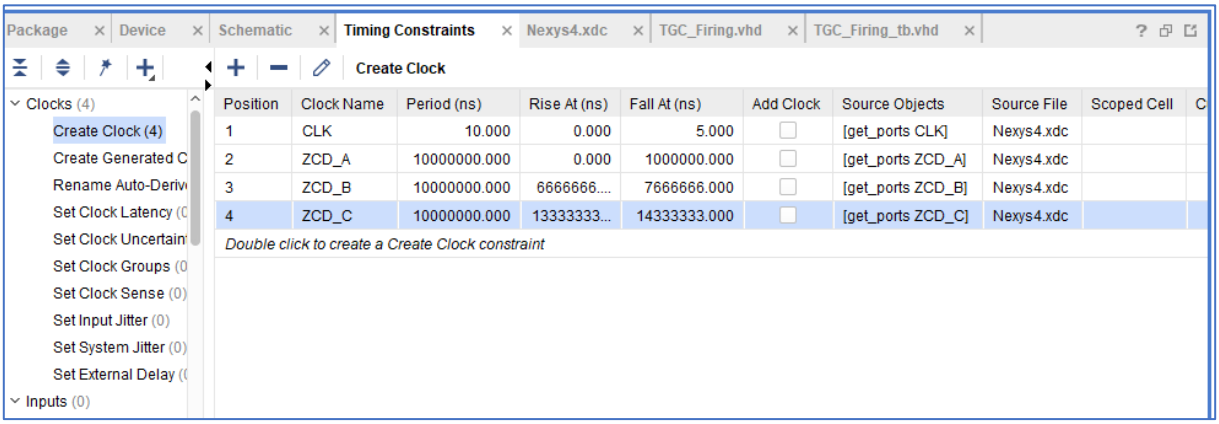


Figure 73.

Clock Jitter and Clock Uncertainty

The ***report_property*** command gives clock properties. Here we see that the jitter for the main clock CLK is 0.05 which is the variation of the edges of the clock from the ideal values.

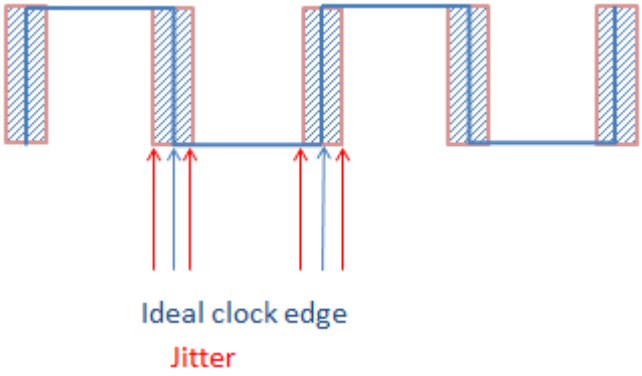


Figure 74. Clock jitter (VLSI Pro, 2016)

Likewise, there will be significant delays between the ideal user-defined ZCD clocks and the real-world ZCD input signals. In practice I believe jitter will have to be measured, however as this type of laboratory work is not a part of my project, I left the value blank.

```
INFO: [Timing 38-35] Done setting XDC timing constraints.
report_property [get_clocks CLK]
Property           Type     Read-only  Value
CLASS              string   true       clock
FILE_NAME          string   true       C:/Users/Jacob/.Xilinx/Dissertation/TGC/Nexys4.xdc
INPUT_JITTER       double   true       0.000
IS_GENERATED       bool     true       0
IS_PROPAGATED      bool     true       1
IS_USER_GENERATED  bool     true       0
IS_VIRTUAL         bool     true       0
LINE_NUMBER        int      true       263
NAME               string   true       CLK
PERI
SOUR
SYSTEM_JITTER      double   true       0.050
WAVEFORM           double*  true       0.000 5.000
```

Figure 75. Clock jitter as shown in clock properties

Figure 76. Input Timing Overview (Xilinx.com, 2016)

Clock Latency

For timing analysis, latency is automatically generated by Vivado. Alternatively, I could have adjusted latencies using the ***set_clock_latency*** command. For an initial timing analysis, I did not adjust the latency myself, however I want to acknowledge that in a more accurate analysis I'd need to consider latency of the Nexys 4 clock of the zero crossing detectors.

I/O Constraints and Virtual Clocks

To have an accurate model of the external interactions, input and output timing constraints must be defined. Vivado IDE can only identify the timing values within the boundary of the FPGA.

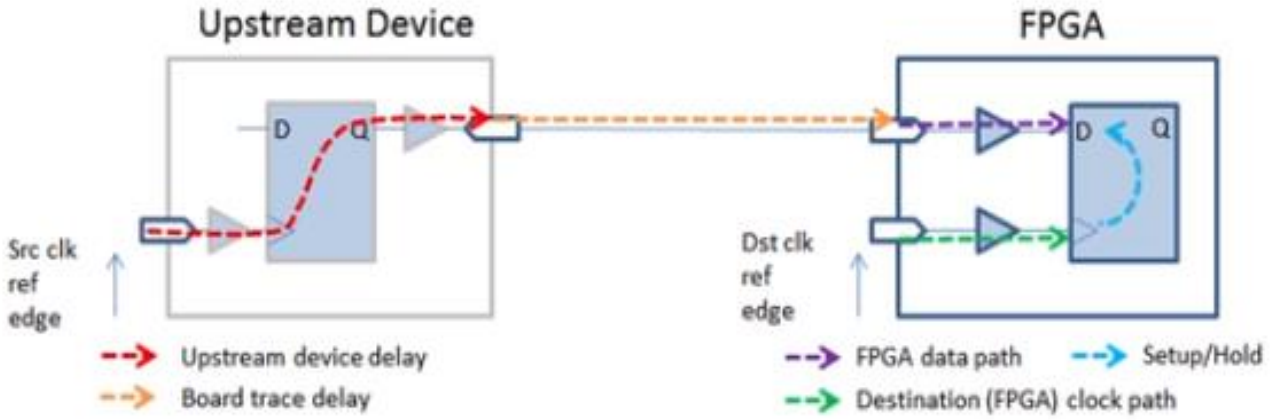


Figure 77. Upstream device timing delays (Xilinx.com, 2016)

The input timing path is generally the path between the launch register of the upstream/external device and the capture register of the FPGA. The input delay can be found from the sum of the clock-to-output delay of the external device and the trace delay in that device. In this case, the external upstream device is the Zero Crossing Detector circuit.

Setup time is the time required for the data to be stabilised before the active clock edge. Whereas the hold time is the amount of time the data should be stabilised after the active clock edge.

For setup analysis, the input delay should be set to the maximum. Similarly, for hold analysis, the input delay can be set to the minimum value.

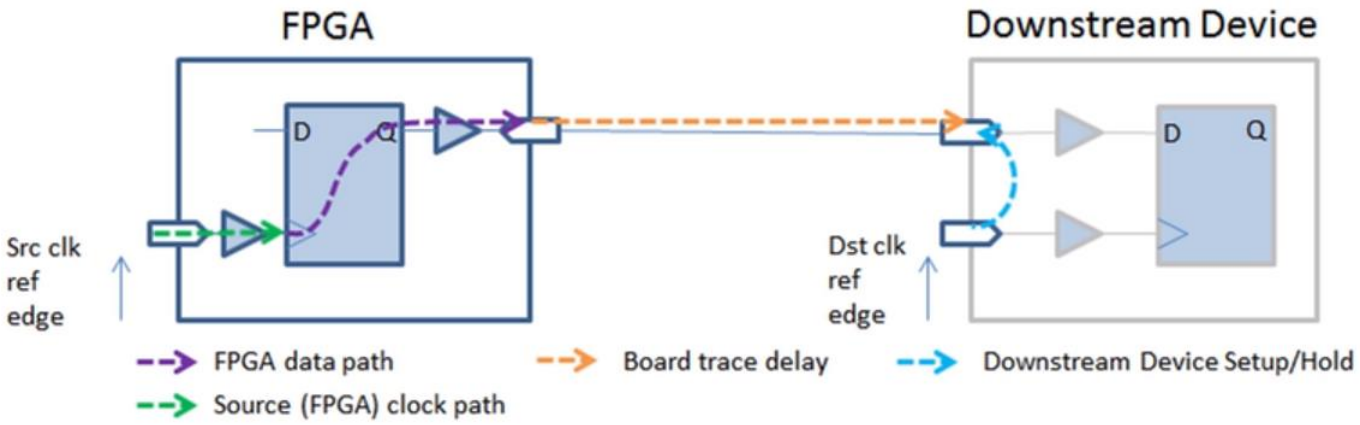


Figure 78. Downstream device timing delays (Xilinx.com, 2016)

For the output delay, the static timing path starts at the register in the FPGA and ends at the register in the downstream device. In this project the static timing path of the output will end at the gate of the thyristor. To fully define the static path a complete knowledge is required of the starting element, ending element, board trace delay, and other delays attributed to the external device (thyristor). I can speculate that the board trace delay can be approximated in PCB layout software packages such as Altium design. Whereas, the delay due to switching in the thyristor will have to be acquired from manufacturer datasheets.

Timing Report

In a bid to better understand the effects of the output delays, I first set the minimum and maximum output delays for firing pulse F1 to F6 as 0 using the Timing Constraints Wizard.

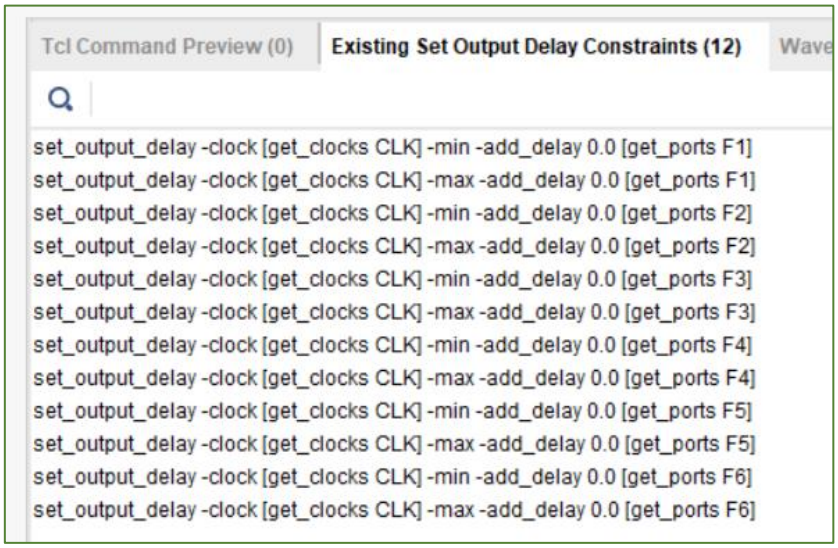


Figure 79. Output delay constraints

I then ran the timing summary report which showed where the many timing failures occurred. All the failures were listed under the inter-clock paths and the intra-clock paths associated with the ZCD inputs. The figure below is an example of one the delay paths that failed timing analysis. It shows a negative slack (-1.048ns) on path 21 of ZCD_A.

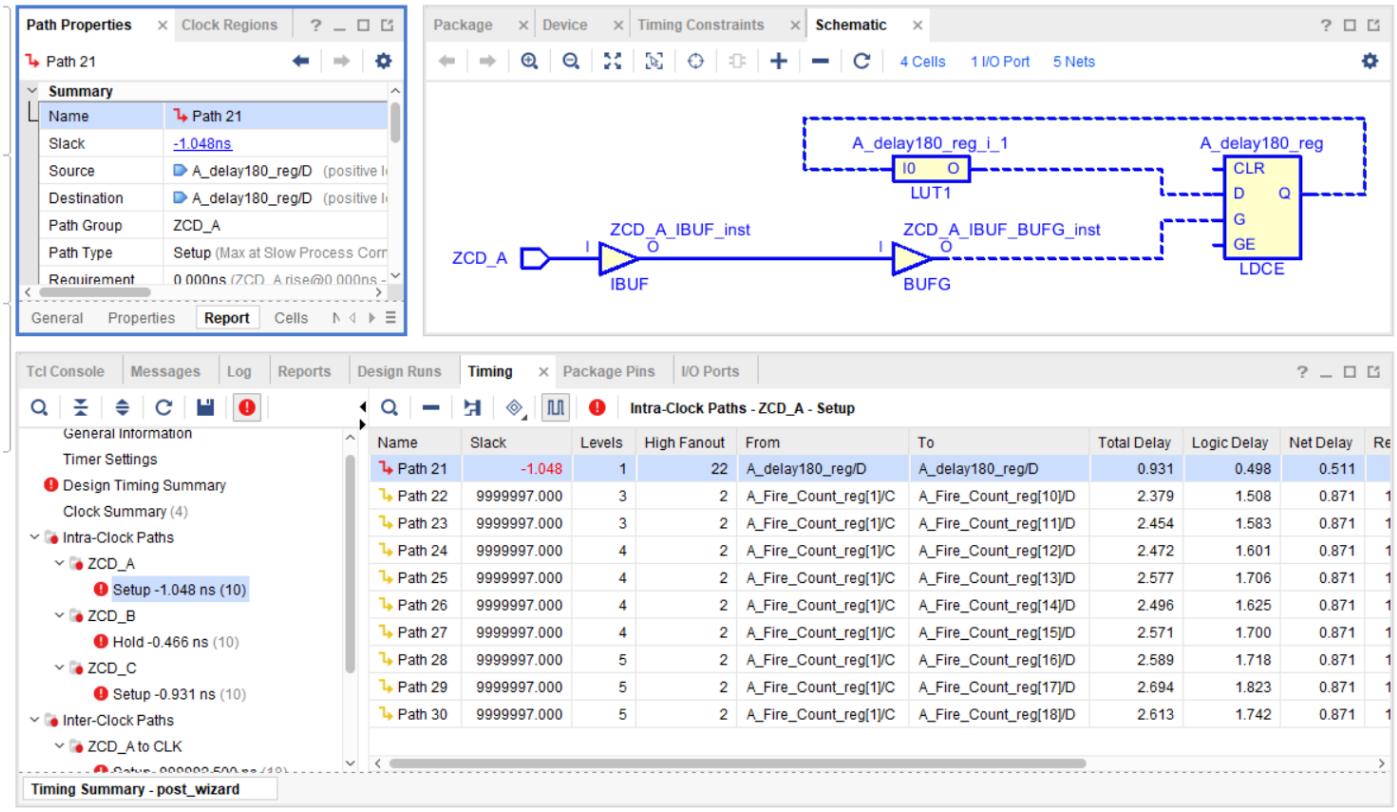


Figure 80. ZCD_A delay pathway

As a result, there were 22 endpoints that were failing timing.

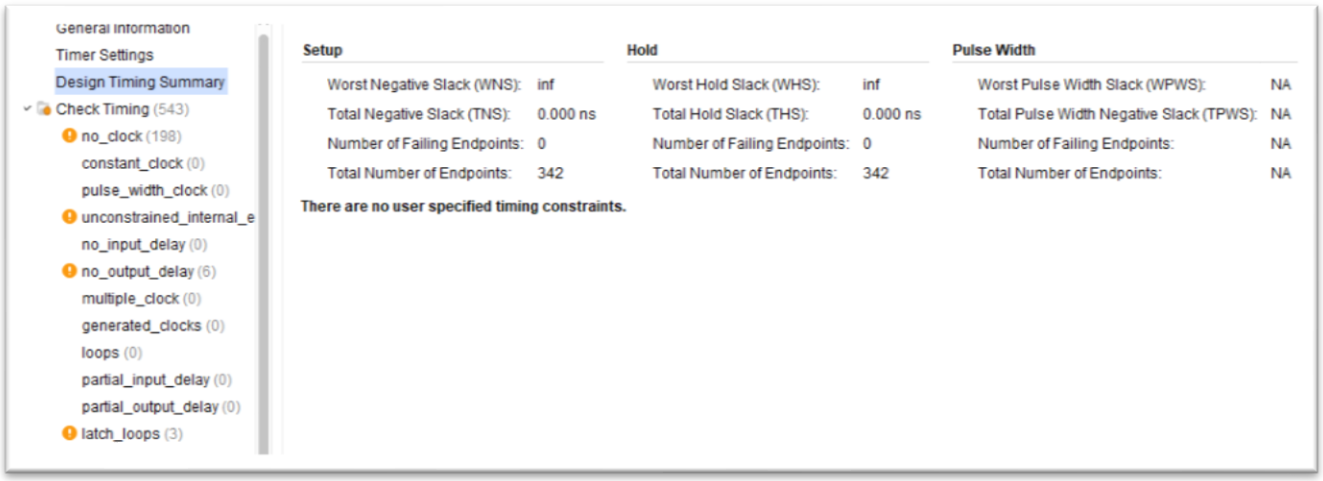


Figure 81. Implemented Design Timing Summary

In the timing summary above we can see that the worst negative slack and holding slack are infinite (inf). This is because the design has no clocks attached to it. For simulation purposes, the testbench TGC_firing_tb is providing its own CLK_tb stimulus, however for post implementation timing analysis I needed to create a clock signal. I have written the following line into the XDC file in order to generate a clock signal called sys_clk_pin for timing analysis.

create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK}];

After this was done, the post implementation timing analysis provided the following results.

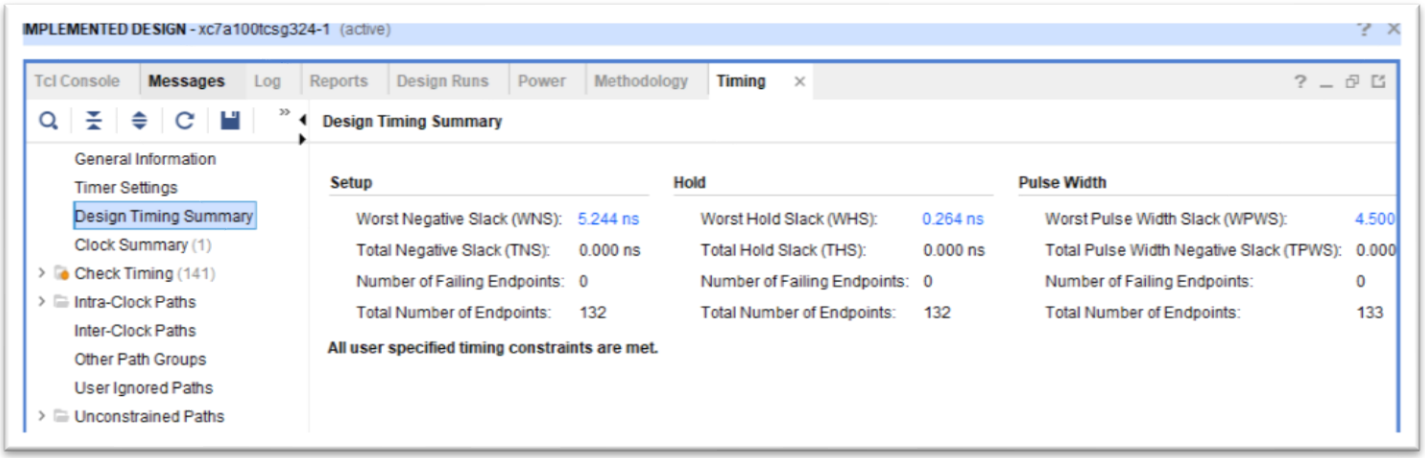


Figure 82. Post implementation timing summary

The figure below shows the negative slack in the design, for which I have highlighted the path in the schematic below.

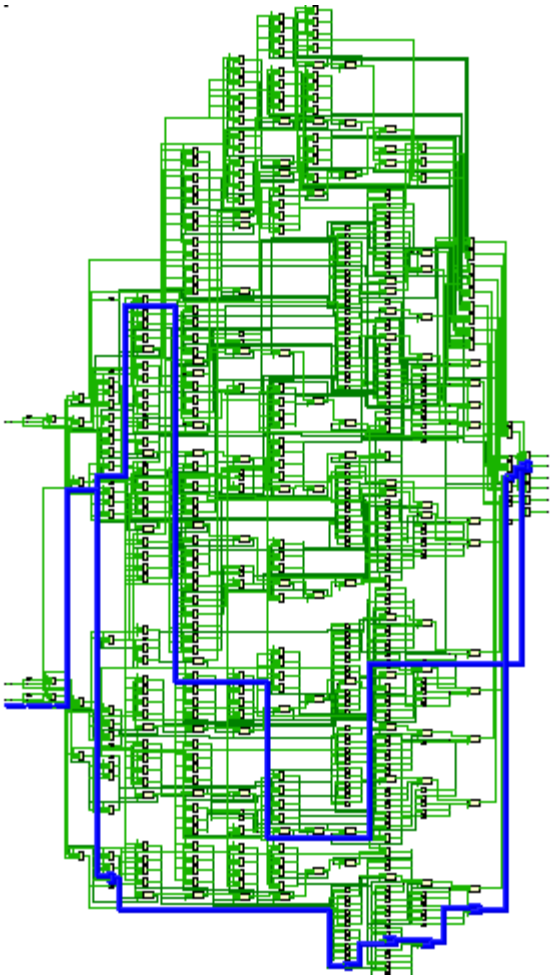


Figure 83. Maximum critical delay.

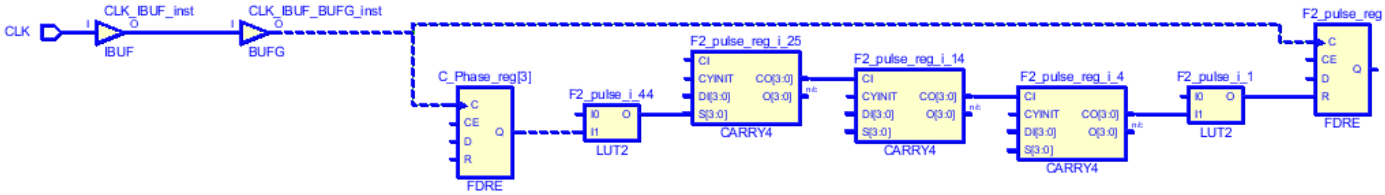


Figure 84. Detailed path of maximum delay.

At this stage there still remained three sets of warnings labelled as ‘Bad Practice’.
The first is that the registers A_Fire_Count, B_Fire_Count and C_Fire_Count were not being clocked.

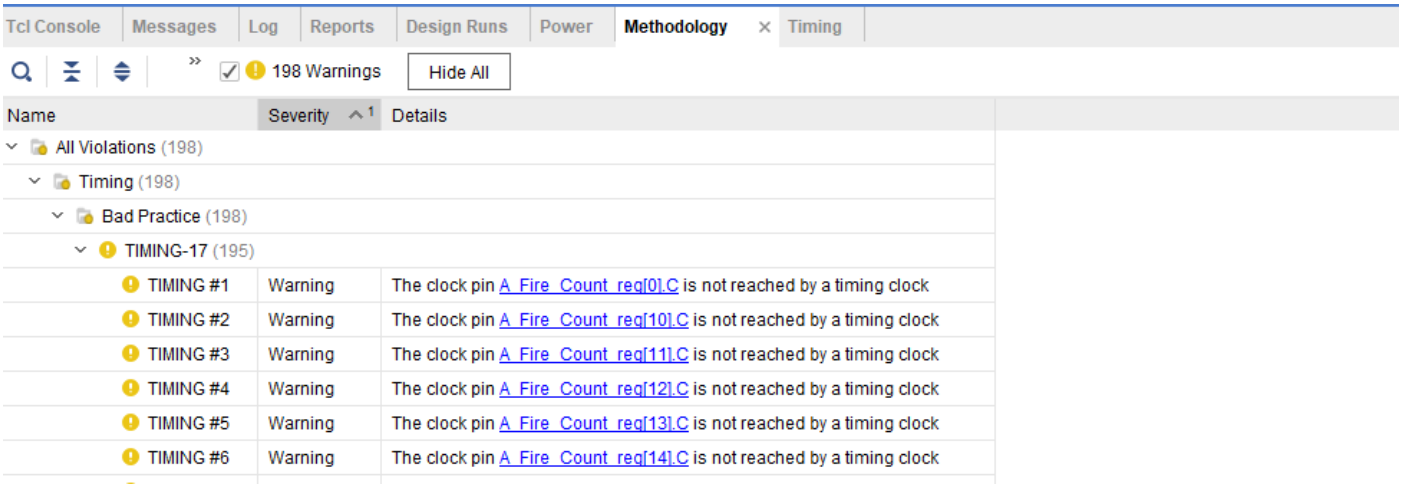


Figure 85. Warnings in post implementation methodology report

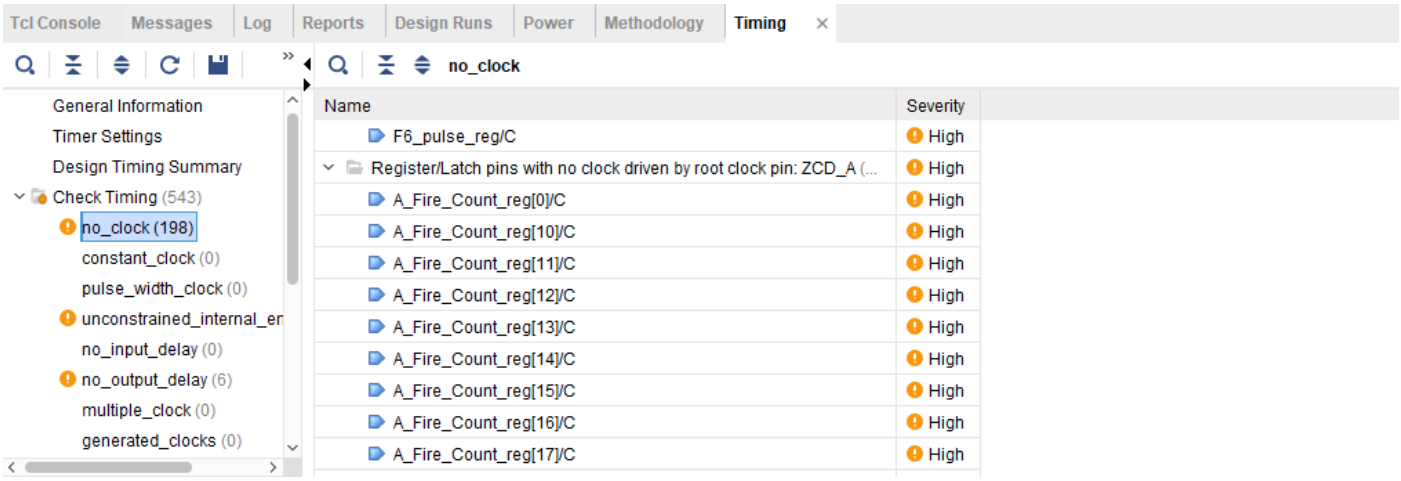


Figure 86. Warnings in post implementation timing report

This is represented in the schematic in the figure below using the broken green line. The example flip flop shown A_Fire_Count_reg [0] is a D-type flip flop with clock able and synchronous reset. With the synchronous reset (R) tied to ground (low) data (D) is loaded into the flip flop on the low-to-high clock transition (CE).

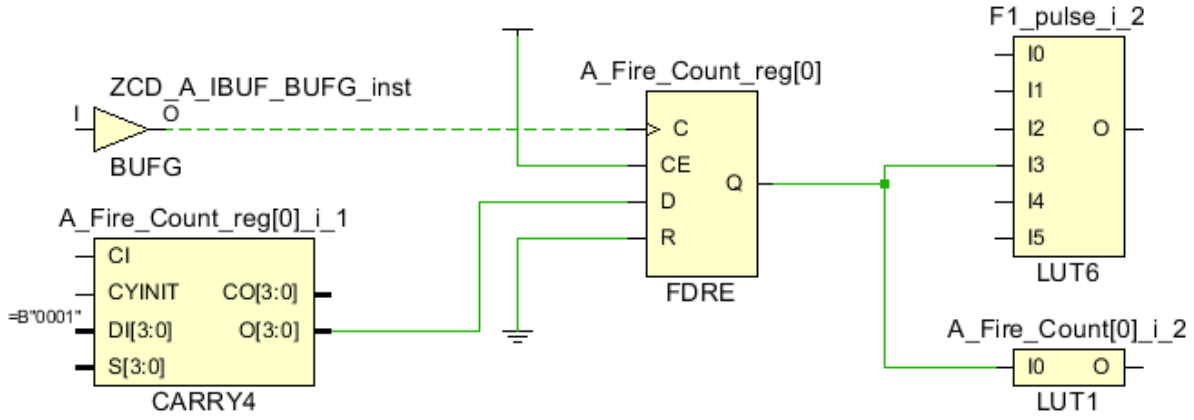


Figure 87. Unknown timing pathway

The source code that explains this behaviour is written as follows;

```
DETECT_ZCD_A: process(ZCD_A)
begin
if rising_edge(ZCD_A) then
A_Fire_Count <= A_Fire_Count + 1;    -- Counts the number of times ZCD is detected.
end if;
end process;
```

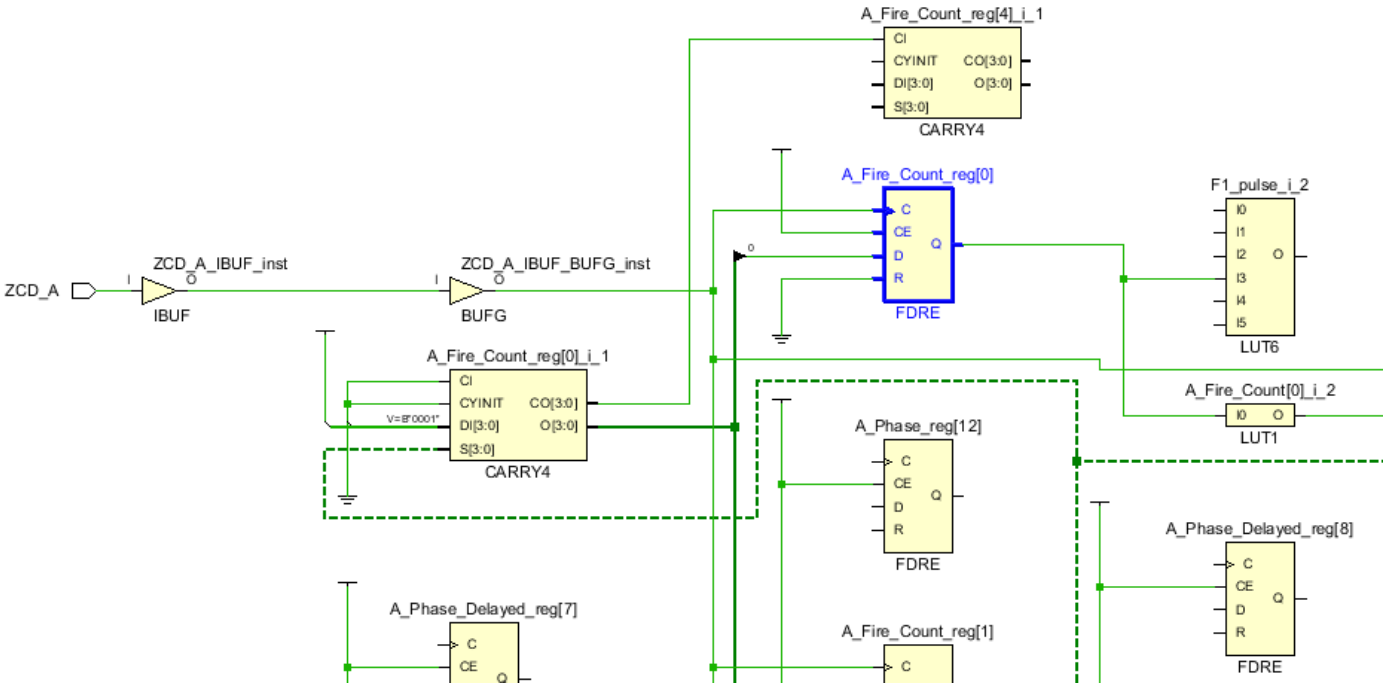


Figure 88. Unknown timing pathways

Along with the HDL code, the above wider schematic view shows that the operation is entirely dependent on the ZCD_A input. Since ZCD_A is an external signal which at this point has not been defined, it means that no timing information can be generated for the timing report.

The next violation reported in the post implementation methodology report is in relation to firing pulses F1 to F6. Here, the output delays are not defined because timing constraints have not been created. These constraints are derived from the interaction of the FPGA with the downstream device externa to the FPGA, being the thyristors/SCRs in this case.

All Violations (72)		
Timing (72)		
Bad Practice (72)		
TIMING-17 (63)		
TIMING-18 (6)		
TIMING #1	Warning	An output delay is missing on F1 relative to clock(s) sys_clk_pin
TIMING #2	Warning	An output delay is missing on F2 relative to clock(s) sys_clk_pin
TIMING #3	Warning	An output delay is missing on F3 relative to clock(s) sys_clk_pin
TIMING #4	Warning	An output delay is missing on F4 relative to clock(s) sys_clk_pin
TIMING #5	Warning	An output delay is missing on F5 relative to clock(s) sys_clk_pin
TIMING #6	Warning	An output delay is missing on F6 relative to clock(s) sys_clk_pin

Figure 89. Timing warnings due to missing timing constraints on firing pulses to thyristors.

The final set of warnings relate to the half-cycle selector switches, HCSS_A, HCSS_B and HCSS_C

All Violations (72)		
Timing (72)		
Bad Practice (72)		
TIMING-17 (63)		
TIMING-18 (6)		
TIMING-20 (3)		
TIMING #1	Warning	The latch HCSS_A_reg cannot be properly analyzed as its control pin HCSS_A_reg/G is not reached by a timing clock
TIMING #2	Warning	The latch HCSS_B_reg cannot be properly analyzed as its control pin HCSS_B_reg/G is not reached by a timing clock
TIMING #3	Warning	The latch HCSS_C_reg cannot be properly analyzed as its control pin HCSS_C_reg/G is not reached by a timing clock

Figure 90. Timing warnings due to missing timing constraints on Half-Cycle Selector Switches

In the figure below, HCSS_A_reg represents a transparent data latch with an asynchronous clear and gate enable.

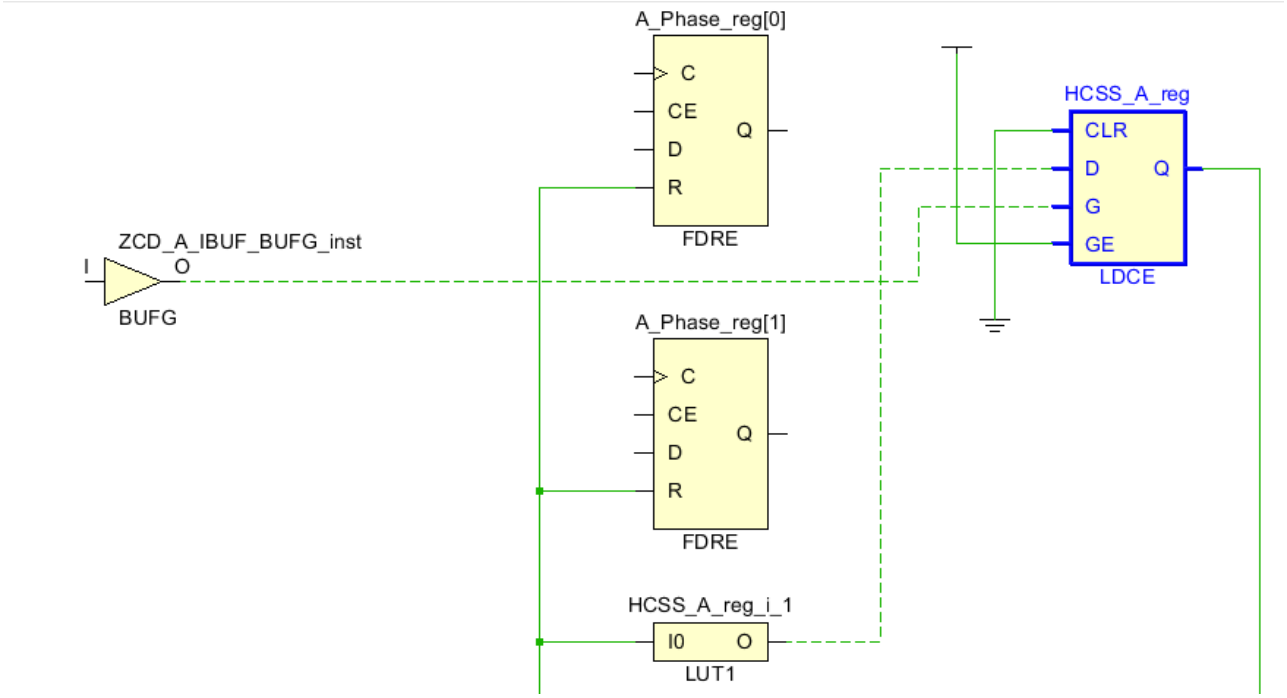


Figure 91. Schematic showing missing timing information on HCSS_A_reg

The clear input (CLR) is tied to ground to keep it from resetting the data on its output (Q). With the gate enable signal (GE) supplied, and the ZCD_A being high, the output of the latch (Q) equals the data from the look up table (D) HCSS_A_reg_i_1 .

The lookup table, HCSS_A_reg_i_1 actually receives feedback from the output of HCSS_A_reg

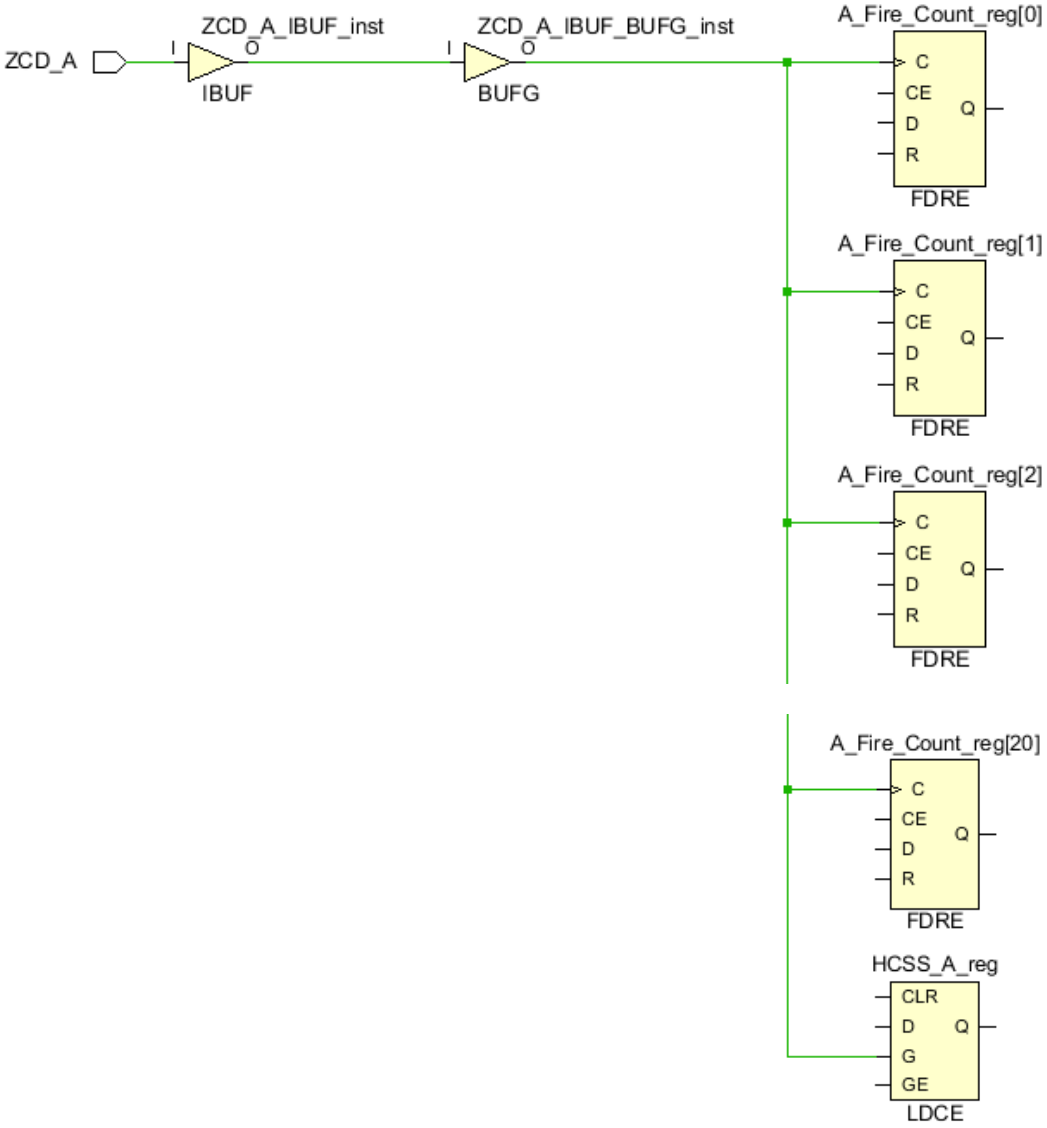


Figure 92.

MIXED SIGNAL ANALYSIS

HDL VERIFICATION

'Cosimulation' is a term used to describe the simulation of both the analog circuit and the digital program simultaneously, from within the same design environment. It is considered to be two simulations rather than one because in addition to Simulink's own simulator an external digital circuit simulator is required. In this case the external simulator used is ModelSim, a software package that fortunately provided a free student licence.

The purpose of cosimulation is to verify and assess the functionality of the HDL program within Simulink by providing it with stimulus and analysing the output. A block model allowed me to import my VHDL code into Simulink, allowing me to measure its performance.

TGC Cosimulation Model

Using the Cosimulation Wizard in HDL Verifier the VHDL file is imported, and the software compiles the file for operation in Simulink. The wizard asks for the ports it detected in the HDL code to be specified as either the input, output, clock or reset type. Ports which I only created for debugging were marked as unused.

The figure below shows the initial Simulink model for TGC cosimulation.

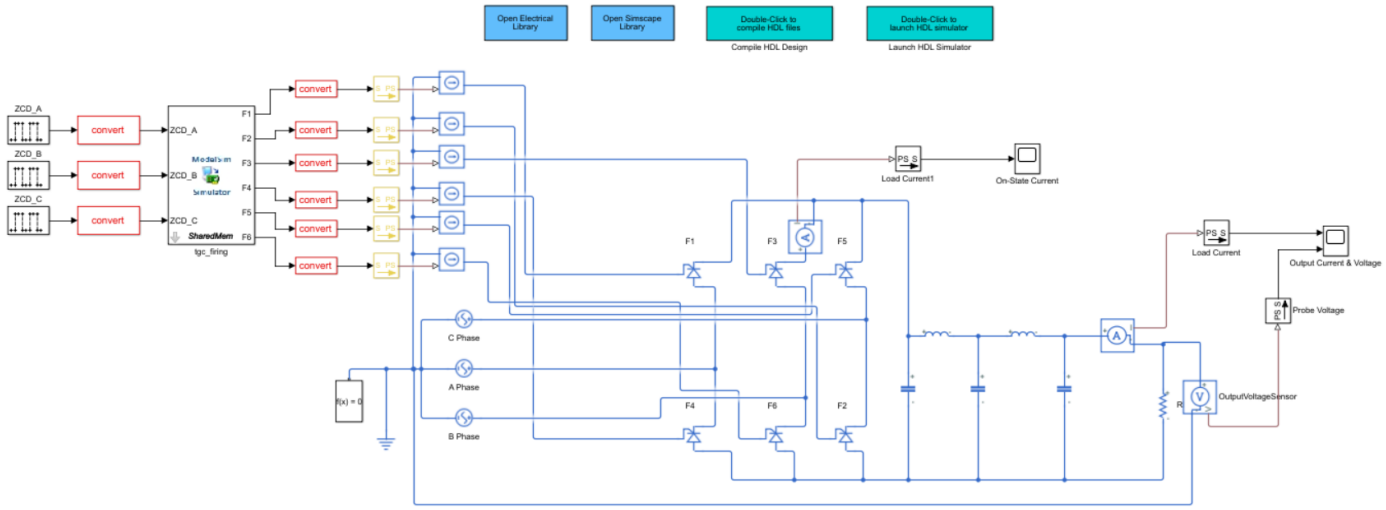


Figure 93.

Attempts at simulation continuously failed due to many varyinh warnings. To isolate problems, I decided to attempt cosimulation with a smaller, more basic VHDL program, before moving on to the main TGC program.

Simple Test Program

To do this, I wrote a simple AND logic program which I first tested in the Vivado IDE. With this model as a test bench I could then i) determine appropriate settings and values of parameters to have a successful HDL simulation in ModelSim and ii) determine appropriate settings and values of parameters to convert the HDL program's digital output of program into physical signals to be employed in Simulink.

As shown in the figure below, the objective of this program is simple. It reads two inputs and when their values are both logic '1', the output table is also logic '1'.

```
entity and_gate is
    Port (
            RST : in std_logic;
            CLK : in std_logic;
            in1 : in std_logic; -- Test input
            in2 : in std_logic; -- Test input
            out1 : out std_logic );
end and_gate;

architecture Behavioral of and_gate is
begin
-- ============ AND Gate Behaviour ===============
process (CLK)
begin
    if (rising_edge(clk)) then
            if ((in1 = '1') and  (in2 = '1'))
                    then out1 <= '1';
            else
                out1 <= '0';
                end if;
    end if;
end process;
end Behavioral;
```

```
in1_Pulses:     process
begin
                in1_tb <= '1';
                wait for 5ms;
                in1_tb <= '0';
                wait for 5ms;
end process;

in2_Pulses:     process
begin
                in2_tb <= '1';
                wait for 2ms;
                in2_tb <= '0';
                wait for 5ms;
end process;

end Behavioral;
```

Figure 94. AND Gate Test Program (left) and behaviour of input signals in testbench (right).

The table below shows the expected response through the output *out1*.

In1	In2	Out1
0	1	0
1	0	0
1	1	1

Figure 95. AND Gate Test Program Truth Table

I then created a simple model in Simulink with the HDL verifier block, as shown below, to test the AND Gate program. The main components are the input signal sources, the ModelSim simulator block, data-type converters, a controlled current source and a load.

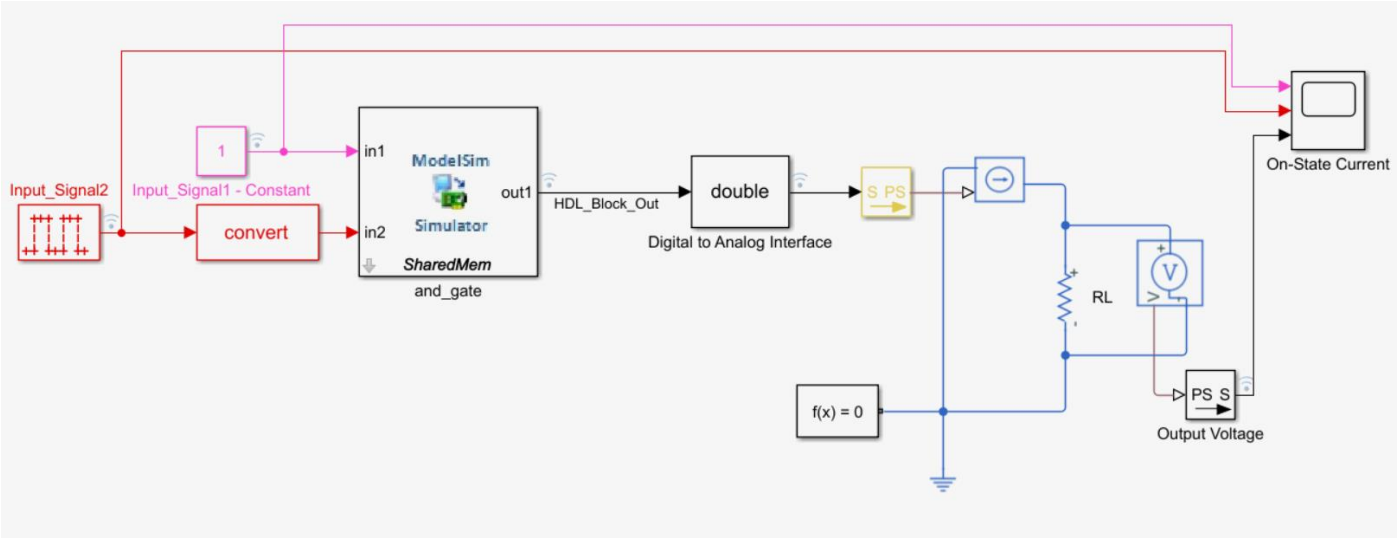


Figure 96. AND Gate Test Program cosimulation model.

The figure below shows behavioural simulation in the Vivado IDE. We can see two different signals being applied to inputs *in1* and *in2*. The output *out1* is operating as expected.

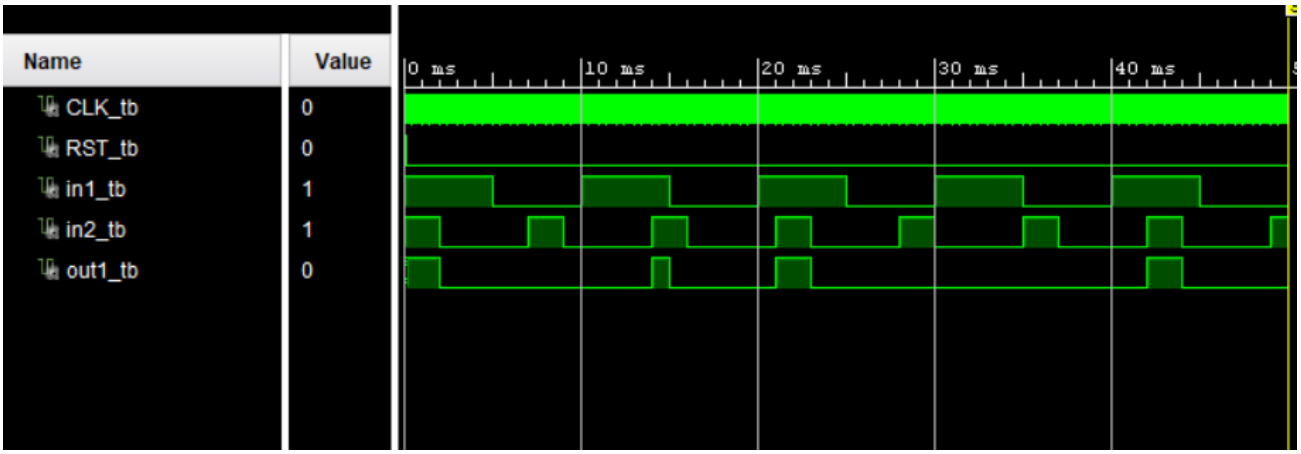


Figure 97. Behavioural simulation of the. AND Gate Test Program

From this point I had established that this vary basic VHDL code was functional, and I could therefore proceed to cosimulate the VHDL code with the Simulink test model.

I started with the creation of the simulator block using the HDL Verifier cosimulation wizard. From the Matlab command window was able to launch the cosimulation wizard with the command *cosimWizard*. First, I specified the HDL simulator being used. There are two digital design software programs available for HDL cosimulation in Simulink. They are the Incisive simulator from Cadence and the ModelSim simulator from the Mentor Graphics Corporation. I chose to use the ModelSim version simply because there was a free student edition available. After selecting *ModelSim* in the CosimWizard (which required that I download and install with a license) I had to add the AND Gate Test program hdl file, *and_gate_test.vhd*.

Next, the 'Shared Memory' connection method was chosen. This allows the ModelSim and Simulink to pass information to each other through shared PC memory. Once this is done, the wizard initiated compilation of my AND Gate Test program for launch in ModelSim.

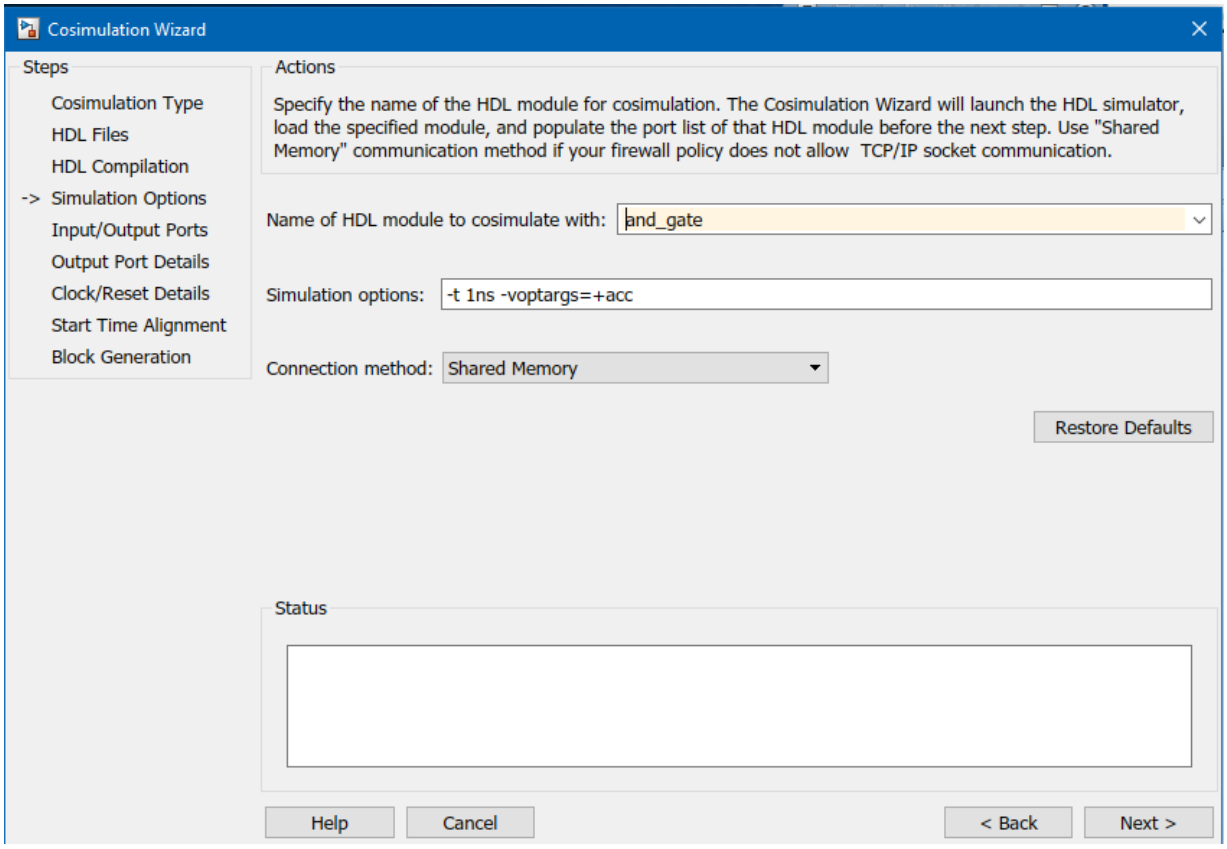


Figure 98. Cosim wizard – simulation options

After this step the simulator requests that I specify the input and output port types. The wizard helps by defining Clock and Reset signals through the ModelSim Tcl command prompt, whereas all other inputs and outputs are specified by myself the user.

The list of port names was automatically populated from the wizard's interpretation of the AND Gate Test programs HDL source code.

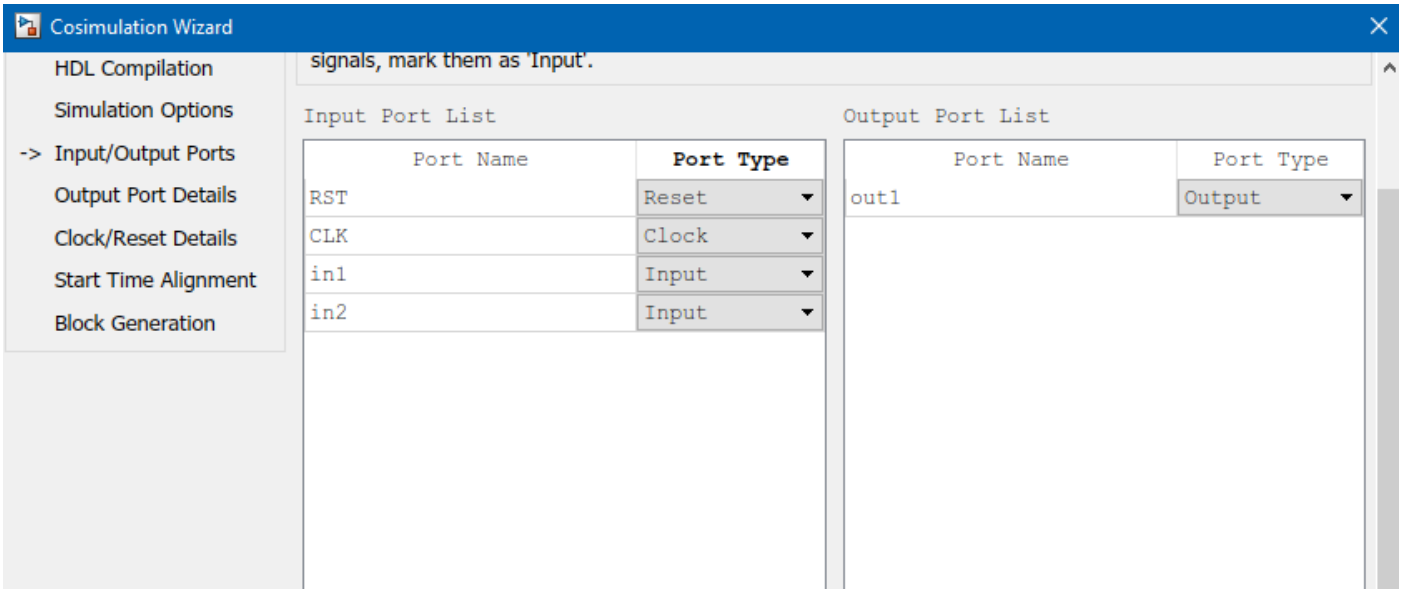


Figure 99. Cosim Wizard - setting ports

The next few steps are concerned with *timescaling* and are the critical parameters in achieving the desired cosimulation functionality, for both the AND Gate Test program and the Thyristor Gate Control design later on. An initial lack of understanding had resulted in dozens of failed outcomes before eventual success.

At the prompt to setup output port details in the cosim wizard, there are four parameters that must be set. They are the Sample Time, Data Type, Sign and Fraction Length. The Fraction Length and Sign fields were automatically populated based on the program code. Because Simulink doesn't actually know what is happening inside the HDL design being run in ModelSim, it can't automatically determine Sample Time and the Data Type on the HDL output ports. Therefore, these fields were available for editing by the designer in the wizard.

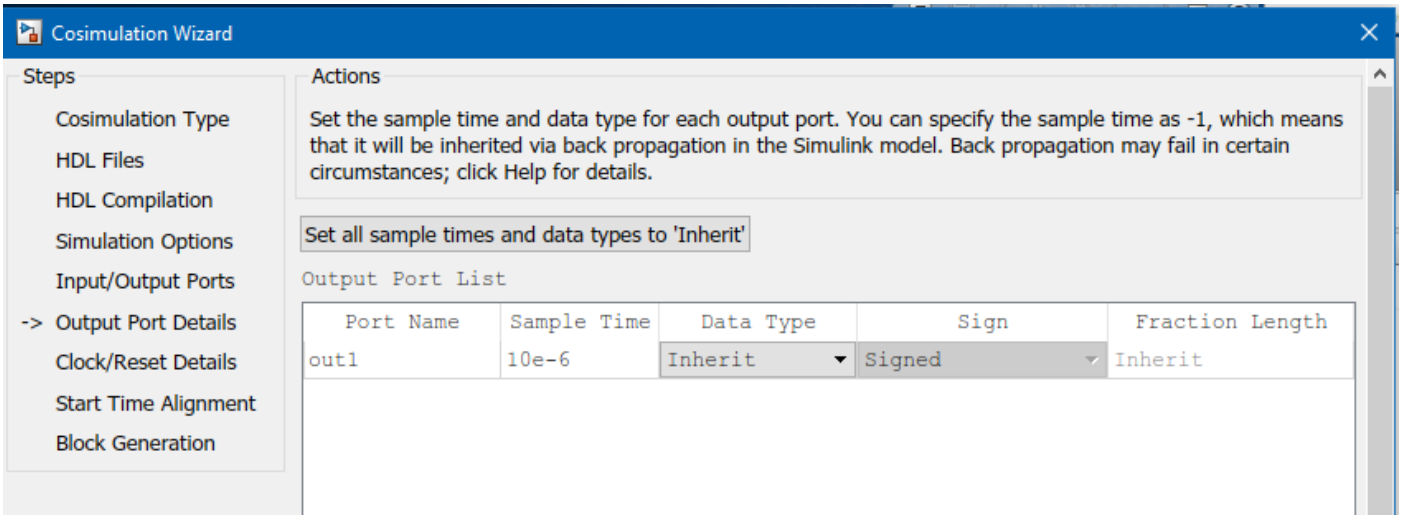


Figure 100. Cosim wizard – setting sample time.

Now, I know that the AND Gate Test program has a simple output in terms of '1s' and '0s', which is Boolean. However, I have selected the 'Inherit' option in this field. This means that through a feature called back propagation, Simulink will look forward in the signal path to see what the data type of the destination port/device is.

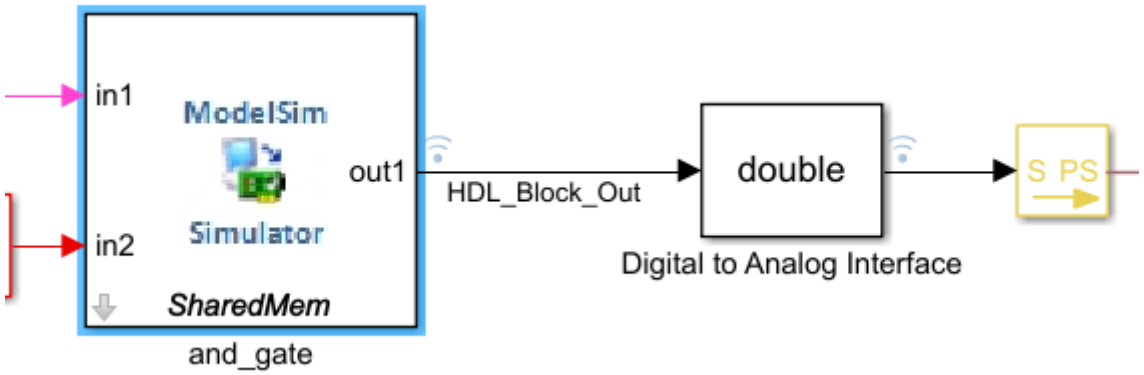


Figure 101. Modelsim HDL block output

From my knowledge of the final system design I knew that the destination block is of the double data type. However, *out1* was still set to inherit its data type through back propagation. This allows me the flexibility to change the destination device as necessary in future designs.

The next parameter to be edited was the sample time on the output port. This Simulink sample time (not to be confused with the engineering term) indicates when a block produces outputs and updates its state during simulation. The default value assigned by the wizard is '1', but this is far from the actual sample rate at the output of this design. I needed to open the Sample Time Legend in the design to automatically detect the true sample of the HDL simulator, but this wasn't posible until the comsimward had completed creating the cosimulation block and run its first instance of the HDL simulator from Simulink. As a result, I had to complete the initial cosimwizard setup using a sample time of '1' with the intention of adjusting it later on when the true sample time was determined.

The next step required that I set the time unit as nano seconds. Then, the final step in the wizard maps the timescales between the two simulation engines (Simulink and ModelSim). I opted to allow the cosimulation block to automatically determine this after the first time the cosim model is run.

The wizard then generates the cosimulation block with some 'model callbacks' shown in figure 98 below. One callback allows a quick compilation of the code for the HDL simulator.

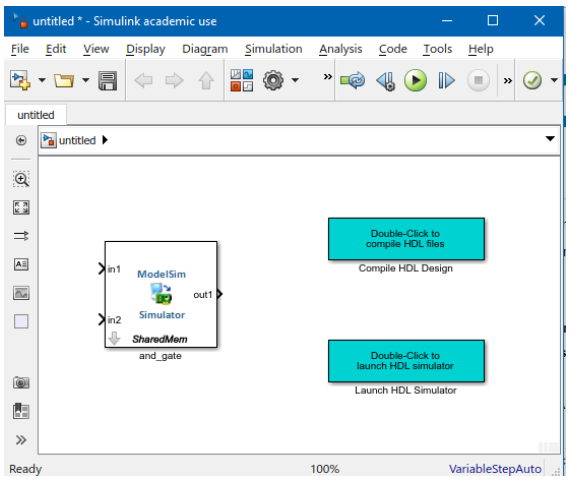


Figure 102. Cosimulation (HDL) block and two callbacks generated by the cosim wizard.

The other callback launches Modelsim, loaded with the AND Gate Test program and necessary Tcl commands. Simply copying callbacks from one design to another lead me to many frustrating failures, as I didn't at first notice that each callback was linked to a specific HDL program.

With the cosimulation block acquired, I was then able to build a testbench around it and determine the sample time of the model. The figure below shows the Sample Time Legend being generated once the ModelSim HDL simulator is opened. From the legend (figure 99) it states that the sample time is 10 microseconds. Going back into the properties of the cosimulation block I then changed the sample time of the output to 10 microseconds.

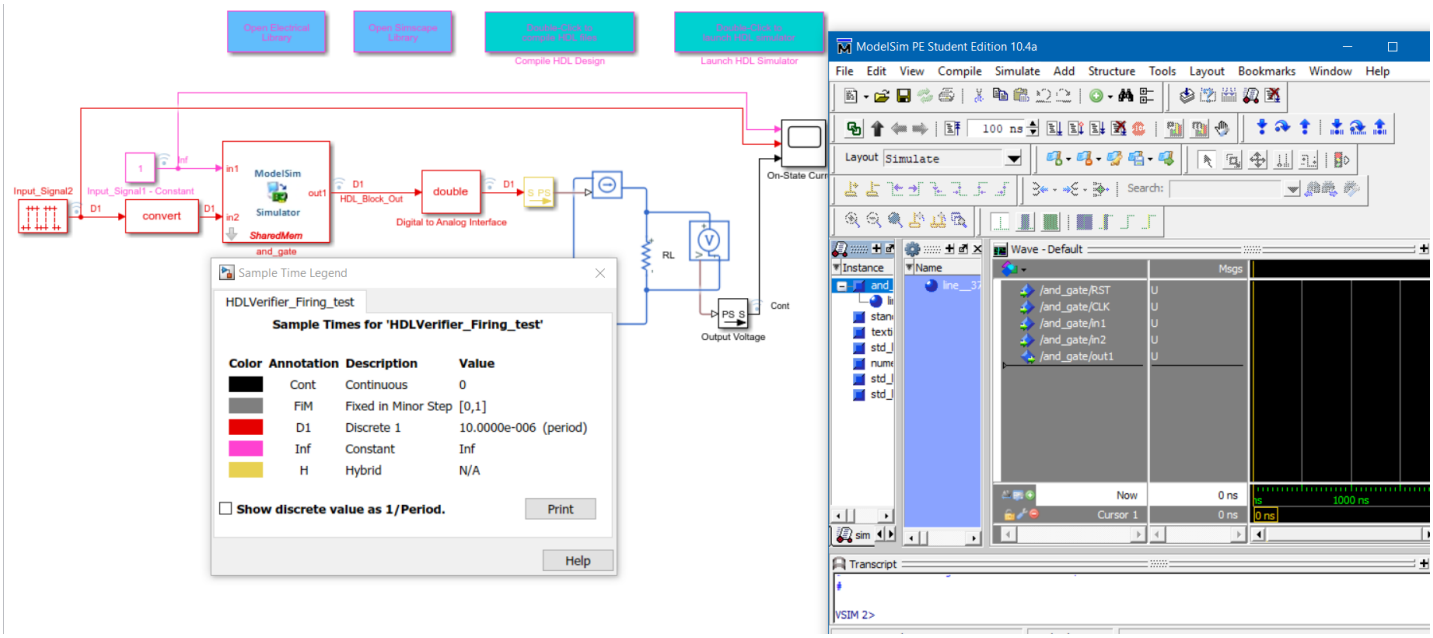


Figure 103. Sample Time Legend

In addition to the cosimulation block, other components such as input signal sources, data-type converters, and controlled current sources were created or configured. In the Simulink model (testbench), the input signal on *in1* of the cosimulation block is a constant. I set its value to '1' and the output data type to Boolean.

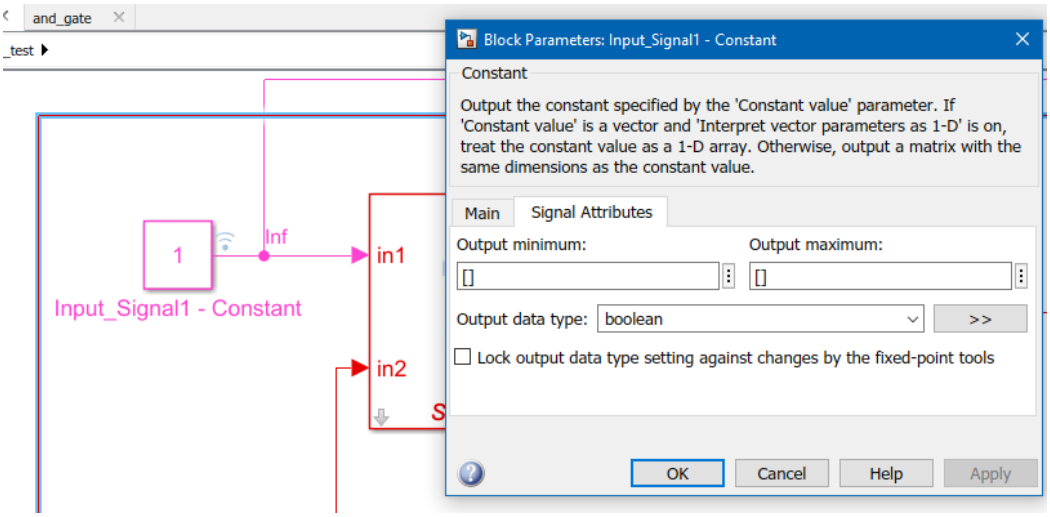


Figure 104. Constant input on in1

The input signal on *in2* was created from a square wave pulse generator.

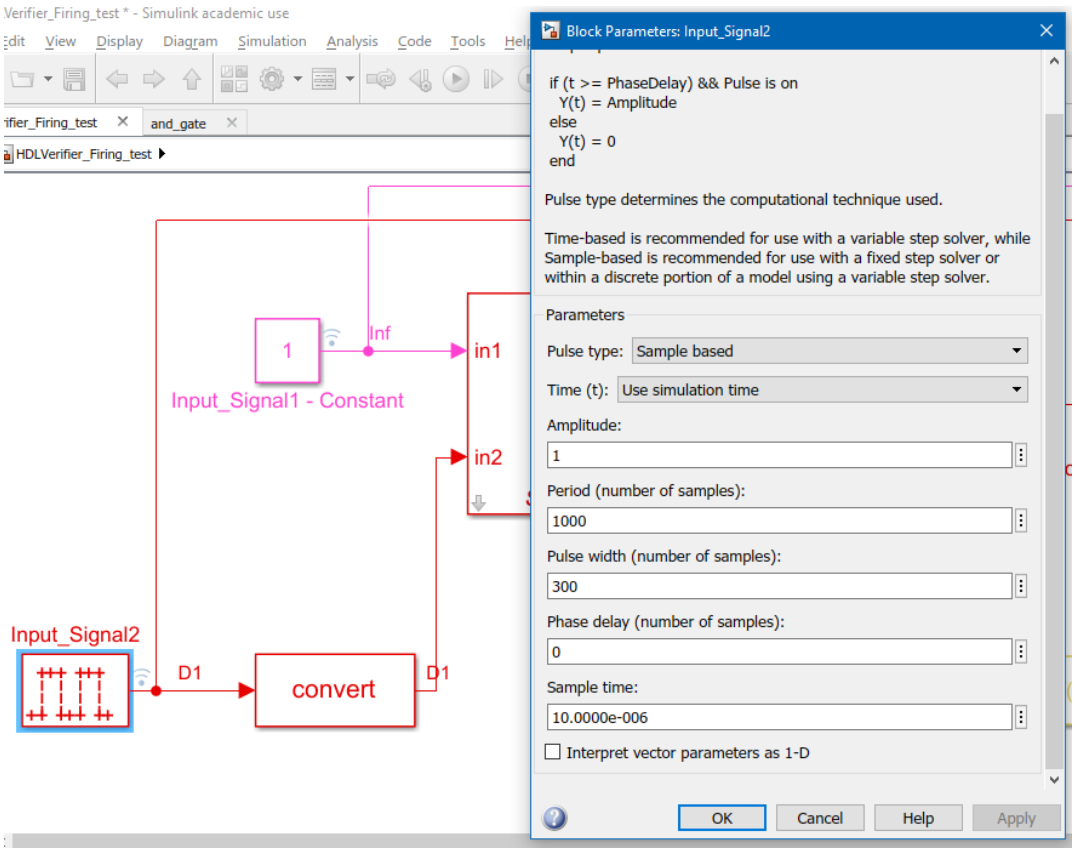


Figure 105. Square wave generator block parameters

The pulse type was set as 'sample based' rather than 'time based' due to the discrete signal required (Boolean) by the cosimulation block's input port. It was important for me to set the sample time on the input ports to the same as that of the output port, with the exception of input port *in1*, which is set as an infinite constant.

With a **sample time of 10e-6s, a period 1000 samples long would last for 1ms**. A data type conversion block was then used to convert the pulse generator's output to the Boolean data type to be read by the HDL program via port *in2*.

Beyond the output of the cosimulation block and the data type converter, there is also the Simulink-to-Physical Signal Converter. This now converts my unitless output, *out1* into a physical current for driving the controlled current source in the model.

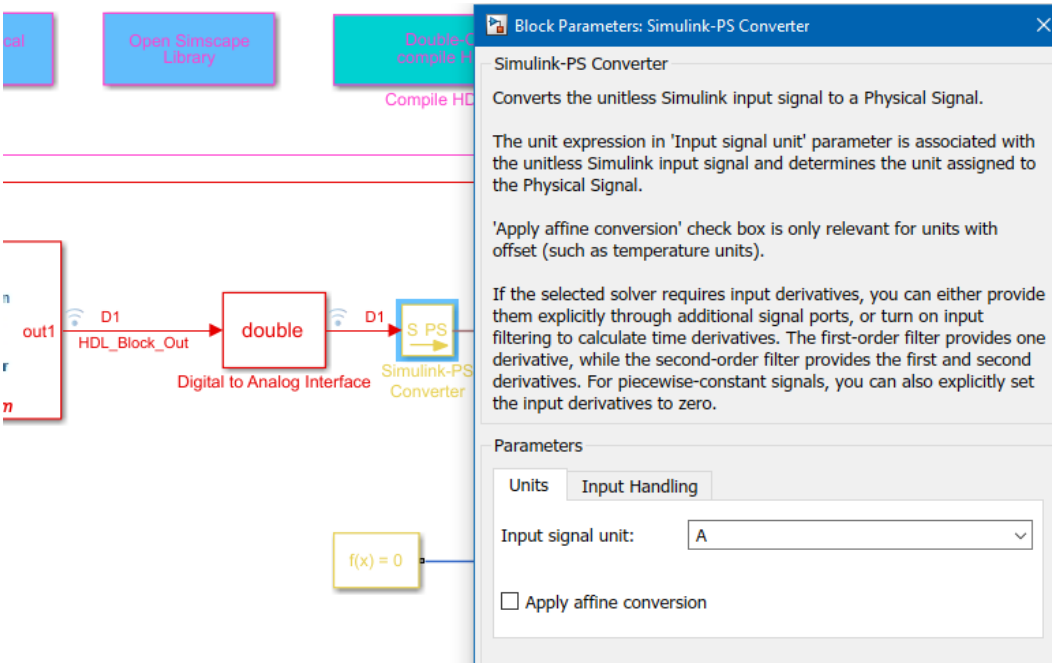


Figure 106. Simulink to current conversion in the S-PS block.

With the correct parameters properly set, the AND Gate Test program was successfully cosimulated as shown in the figure below.

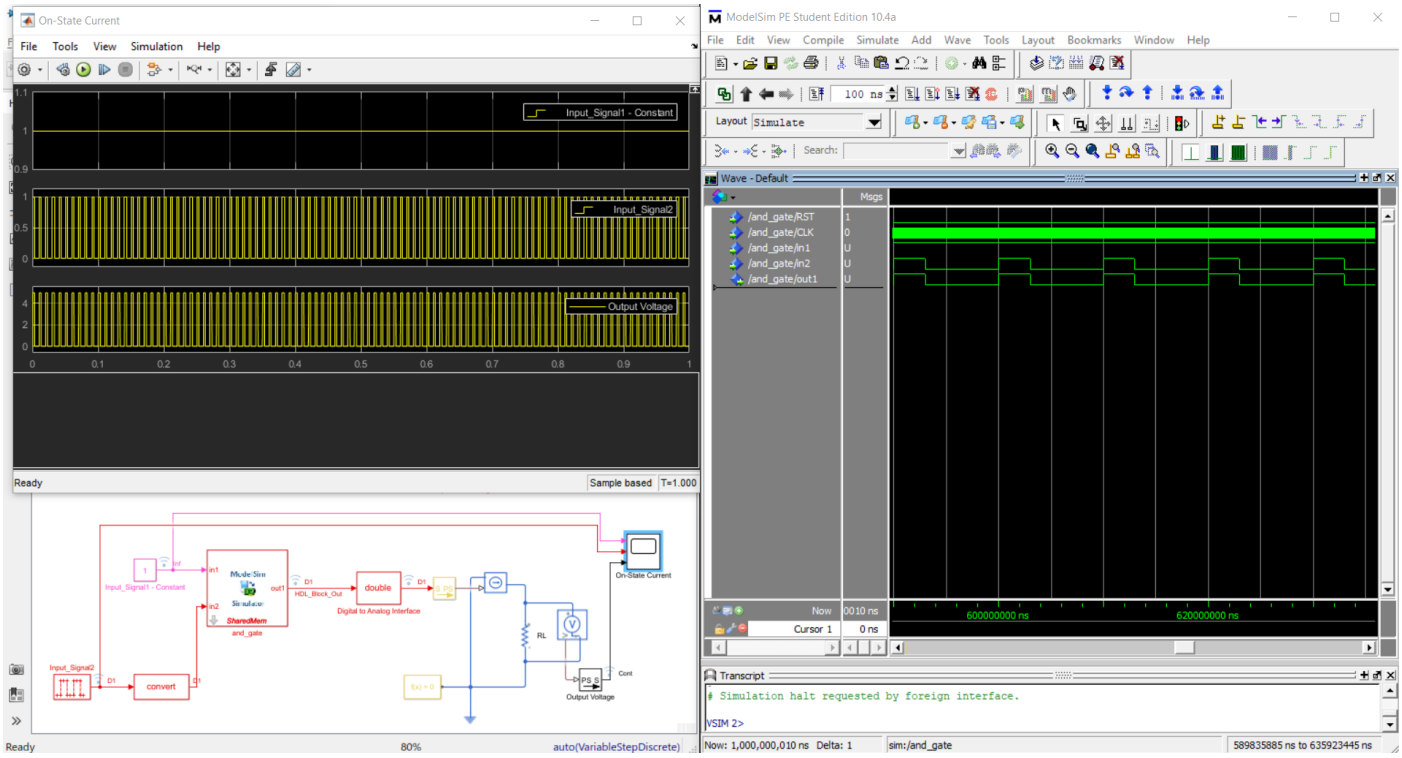


Figure 107. Succesful cosimulation of AND Gate Test program and testbench in Simulink

This finally enabled me to understand how to set up the cosimulation of my TGC program with the SCR bridge rectifier model. Before attempting to do this however, I decided to only test the firing pulses of the TGC program in Simulink testbench. The figure below shows the model for doing so.

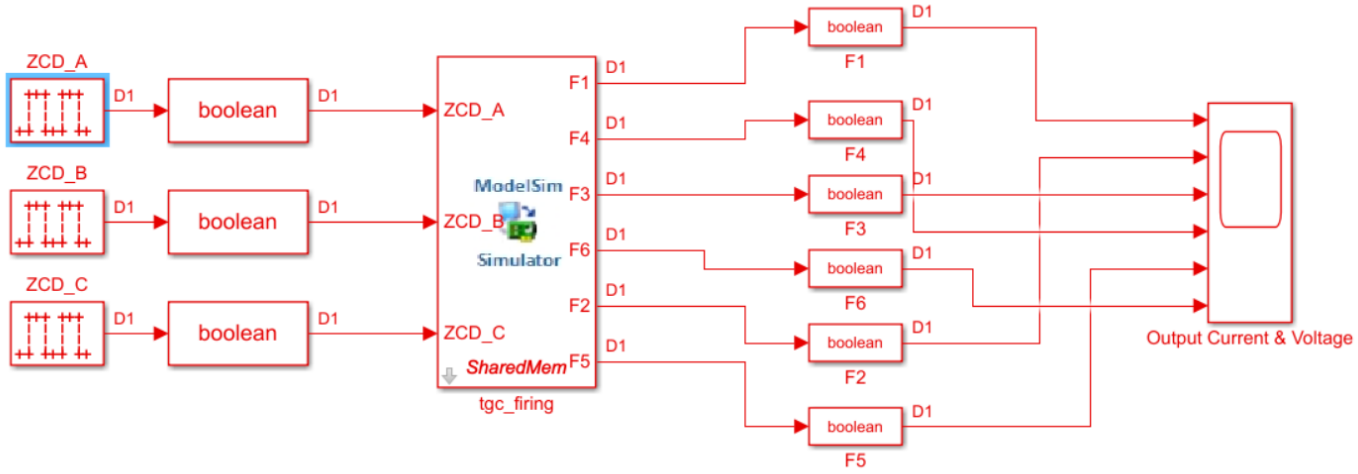


Figure 108. Model for testing firing pulses only

Following the same procedure performed in the cosimulation of the AND Gate Test program, I configured the input signals, HDL block, data type converters and then run the cosimulation to observe the function of the firing pulses. The figure below shows the output waveforms in the ModelSim HDL simulator, and also the output measured by the Simulink scope. This simulation didn't go as smoothly as expected. Two problems became apparent. Firstly, the simulation was incredibly slow and took over 2hours. Secondly, the waveforms of the firing pulses produced in cosimulation seemed to be phase shifted.

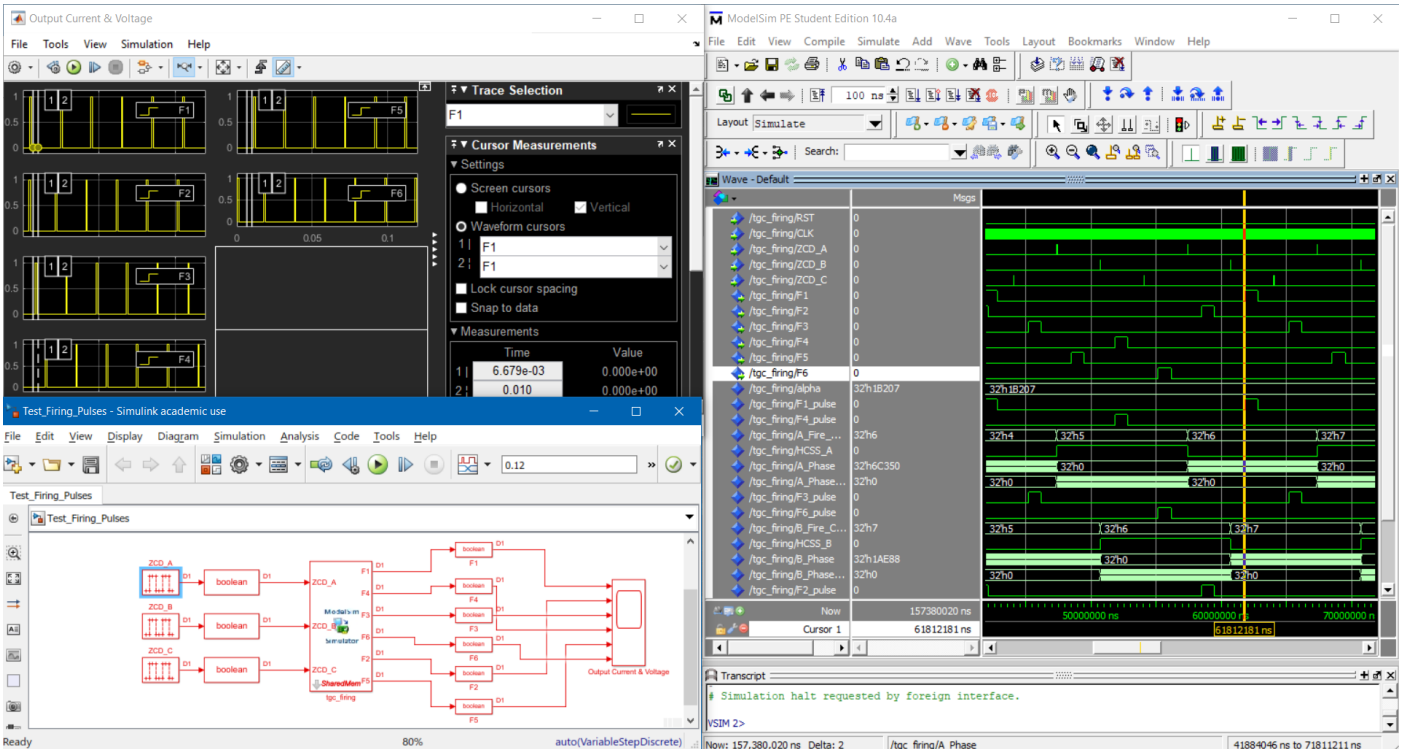


Figure 109. Successful cosimulation of AND Gate test program between Simulink and Modelsim

To better understand how the cosimulation result was different from the post implementation simulation in Vivado, we can look at a comparison of their waveforms.

I therefore went back into the HDL code in the Vivado IDE to attempt to find the source of the problem. Unfortunately however, due to time constraints I was unable to debug the problematic cosimulation by the project deadline.

DISCUSSION

The objective of this project was to conduct an investigation into AC power rectification, in particular for the controlled and uncontrolled bridge rectifier types. At the heart of the control rectification was the thyristor gate control (TGC) algorithm, but equally important is the **zero crossing detector** (ZCD) that feeds the TGC program with stimulus to act upon. In this report I never really elaborated into the inner workings of the ZCD. However, its operation and its limitations are quite important for the designer to be aware of. Simplified, there are several ways of implementing the functionality of a zero crossing detector. Firstly, there exist designs that make use of operational amplifiers (OpAmps) as comparators. There are also more simplistic analog circuit designs. One of the important features for consideration are the output desired from the ZCD. It must be suitable for the digital-in ports on the FPGA device, equivalent to an active high or active low signal as is required.

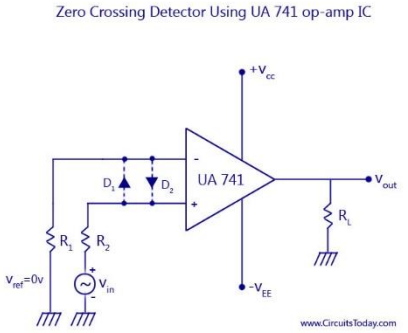


Figure 110. Simple OpAmp based ZCD (Circuits Today, 2014)

Towards the end of the literary review I paid some attention to unwanted effects in rectifier circuits and the use of **filter technology** to eliminate or minimise them. There was brief mention of the concept of **electromagnetic interference** however, in real-world designs often incredible efforts are placed into design for electromagnetic compatibility. Analysis of the effects of EMI is usually performed with highly powerful simulation programs such as Keysight's ADS, National Instrument' AWR or CST Microwave studio, all very popular industry tools. Filters can be used in rectifiers to counteract the effects of external EMI. However, in rectifiers an equal, if not larger concern would be about what **other systems can be affected by electromagnetic phenomena** generated from within the rectifier. Rectifiers, being part or a power supply, are often placed in larger more complex host systems which will have varying levels of susceptibility to EMI. It is a complex task for several disciplines to create a tailor-made solution for the EMC requirements of each system.

The uncontrolled rectifier's output was as expected, pulsating DC. When attempting smoothing using parameters of an actual local manufactured inductor, I was given an important reminder about the trade-offs between increased inductance and output voltage (due to increased inductor DC resistance).

Harmonic analysis of the rectifier output was very interesting. A special point I realised was that the rectifier circuit itself (diode network) acted as a non-linear load, causing harmonic distortion. What was interesting is that the significant harmonics seemed to be related to the number of diodes in the rectifier. Upon further investigation I learnt that certain types of loads produce predictable **'harmonic spectrum signatures'**. For rectifiers there's even a formula, h = (p*n) +/- 1 where h is the harmonic number, p is the number of pulses, and n is any integer.

My literary review touched on the **power factor (PF)** however I didn't go much more into it. From many literary sources I came across a link between the power factor and the total harmonic distortion. This makes sense as the power factor is affected by the real power in the circuit which is reduced when current and voltage waveforms are out of phase. This then brought my attention to **power factor correction circuits**, of which there are various methods being used today.

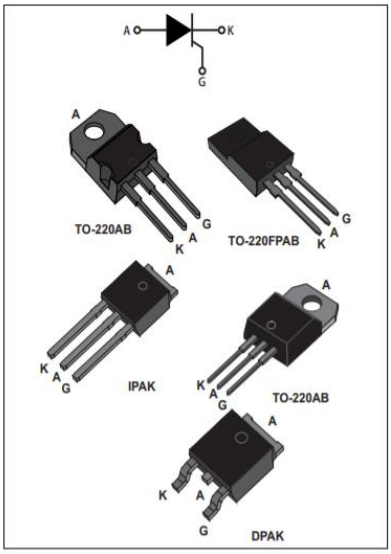
The topic of **transients** was touched on very briefly. The effects of transients in many electronic systems today can be considerable. Some transients can be caused by EMI but one would find that transients are an inherent problem in switching devices. According to the authors of Transients of Modern Power Electronics, nearly 70% of failures in power electronics occur in transient processes rather than in the steady state. It reinforces the belief that in the realm of transients and **transient analysis**, a lot is still unknown.

The topic of **thermal management** was also not addressed in this project but is a critical issue for power electronics engineers today. Insufficient thermal management can affect the reliability of systems and shorten their life expectancy. The best way to address thermal effects is in the design stages. The selection of suitable materials, fans, proper PCB layout and efficient use of heat sinks all make up an effective **thermal management strategy**.

In the digital design side of things there were several parameters there weren't accounted for by in the **timing analysis**. One issue highlighted was the delays between the **upstream and downstream devices** interacting with the FPGA and the **pathways between them**. This would normally be resolved in a collaboration between PCB design engineers and the FPGA designers. For example, PCB designers could help to better quantify trace delays and delays within the thyristors themselves. In addition to this there are digital design techniques that are often required to counteract the **effects of noise**.

In my system conceptual design, it can be observed that the **firing angle was not truly 'controllable'** by the user. Instead it is better described as a **preset control** of the rectifier, whereby the customer would have to request that the digital designer reprogrammed the FPGA to acquire a different firing angle based on the customer's requirements. To have a truly controllable firing angle, alpha, I would have to build in some form of input to the FPGA that would change it. It could be a potentiometer that would require an analog-to-digital converter (ADC), a digital keypad or even a multi-bit switch.

The **firing of the thyristors** themselves is worth a mention. In this design I showed firing pulses 1000 clock cycles (1 μs) long. However, in practice each thyristor will have its own requirement for the **shape, amplitude and the length of the firing pulse (gate triggering current)**. To create the correct shape may require additional analog circuitry. Additionally, one pulse may not always be successful in triggering the thyristor. One way to resolve this is to send a train of multiple firing pulses at a time.



Features

- On-state rms current, $I_{T(RMS)}$ 8 A
- Repetitive peak off-state voltage, V_{DRM}/V_{RRM} 600 and 800 V
- Triggering gate current, I_{GT} 0.2 to 15 mA

Figure 111. Thyristor gate triggering specifications.

CRITICAL EVALUATION

Some of the first problems I encountered were due to the fact that I overestimated what I could achieve within the timeframe report size of this report. Roughly four months and fifteen thousand words seemed like a vast amount of room to accomplish incredible things. As a result and as commented on by some of my academic advisors, the initial goals I set were too ambitious.

The Gantt chart in figure 111. below shows the tasks I had set myself. The orange bars the preceded blue bars are an indication the delay in starting a task compared to the target start date outlined in the introduction of this report. Since several tasks were dependent on the completion of others, one could observe the knock-on effect. The pink bars on the chart showed how much longer the actual project took in comparison with the initial estimates.

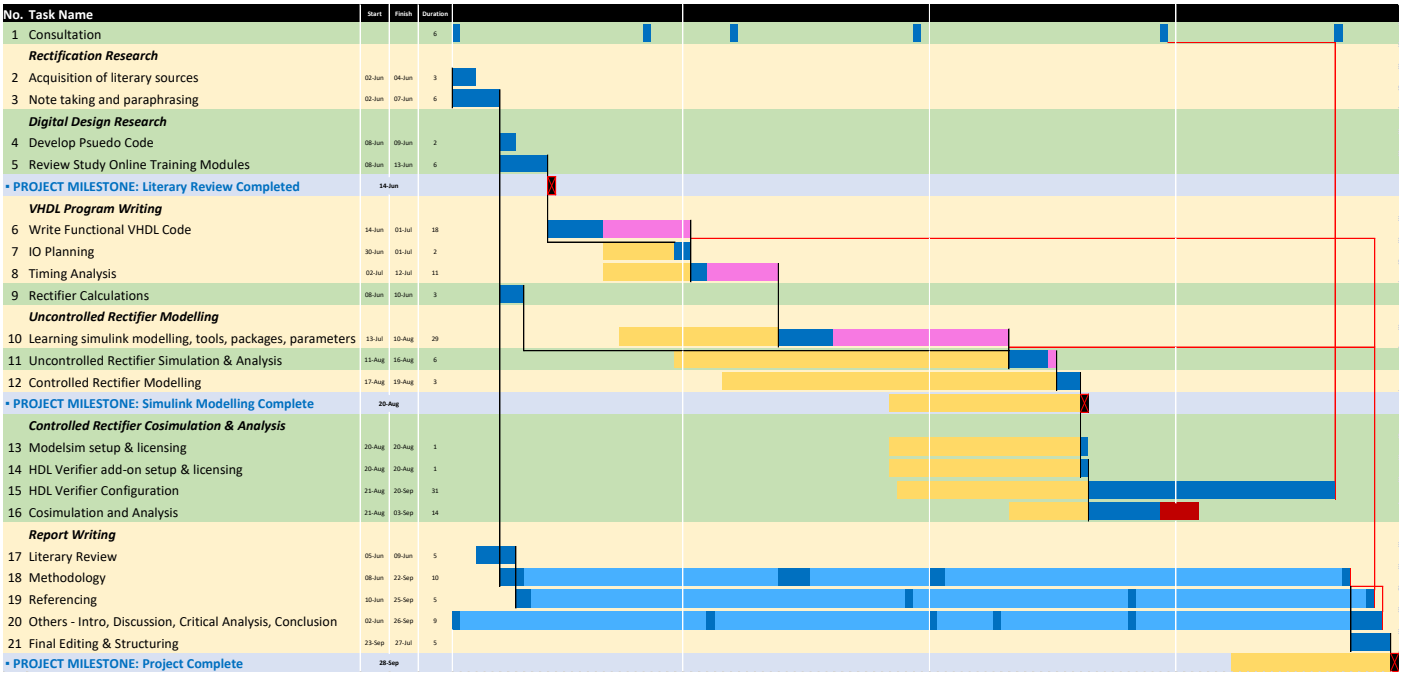


Figure 112. Gantt chart of actual project completion times.

The first main problems in keeping up with my planned project schedule arose from the difficulty in programming of the thyristor firing control algorithm, task number 6. From the chart it could be seen that this task took more than double the time I thought it would. The next major delay was due to the timing analysis, task number 8. In retrospect I do believe that perhaps I spend too much time on this area. I went too deeply into my analysis of all the features and parameters available in the Vivado IDE, many of which perhaps weren't sufficiently relevant to my project.

The last main delay came about in task number 10, the modelling of the uncontrolled rectifier. This was perhaps the most difficult task of all and resulted in the biggest delay in the timely completion of my project. For someone who has the relevant experience, this task could have been accomplished relatively quickly. However, my problem I was not sufficiently knowledgeable to begin with, and I had to undergo a very steep learning curve in order to get the Simulink to do what I wanted. A better approach would have been for me to get in touch with an expert in on Simulink in the earlier stages of my project.

The cosimulation and analysis in task number 15 was not even completed. This is the reason for the red bar at the dark red bar on the chart. Disappointingly, as the project deadline drew closer, I had to abandon it to focus on completing this report. That being said, the configuration of the HDL verifier add-on package was eventually successful as demonstrated with the AND Gate test program. This was tremendously satisfying.

I must confess that I also wasted a lot of time on other areas not reflected on the Gantt chart. One such area was the application of an LC filter. I digressed with an analysis of the filter's transient response, but in the end found it too complex to accomplish within the scope of my report.

Another area I spent time on but ultimately abandoned was the PCB design. I spent considerable time learning to perform PCB layouts in the Altium designer package. While this too felt rewarding, it didn't really add substance to the technicality of my project. It would have been more worthwhile if I were to perform this design to demonstrate an application of EMC principles. However, designing for EMC is a highly complex topic and skillset, which again was beyond the scope of my project. Figure 112 below shows a 3D view of a zero crossing detector circuit I modelled. This design never came to fruition as I abandoned it to focus on the more important matter of the rectifier modelling.

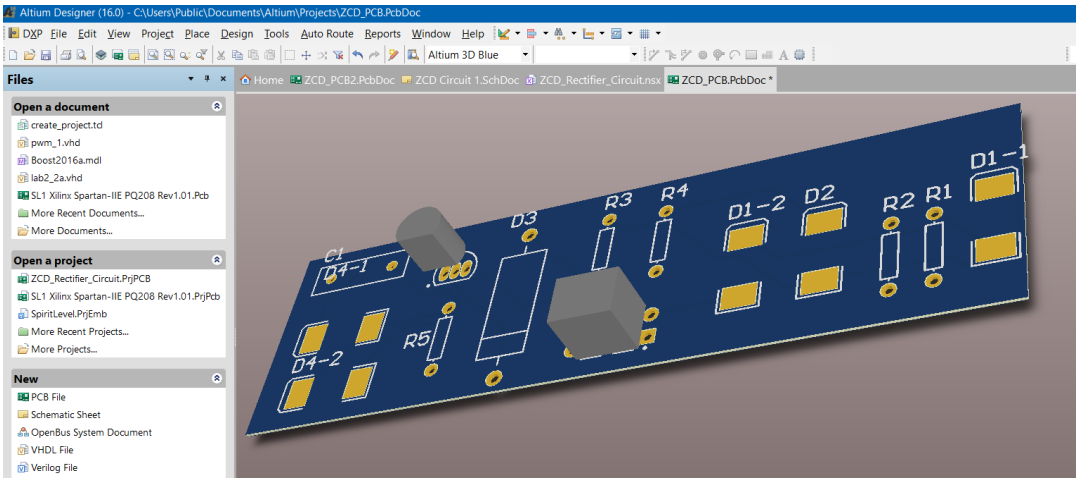


Figure 113. Aborted Zero-Crossing Detector PCB Design

Last but not least, I would like to comment on the Gantt chart shown above. It was an excellent idea, but due to the fact that I manually created it in MS Excel, it was quite tedious and not as effective as it could have been. Because the chart had to be manually updated, it was difficult for me to quickly notice and react to delays in my work. A better idea would have been for me to use an automated Gantt chart product. I did look for one online, but I could not find a cheap or free option that could perform exactly the way I wanted. That being said, the Gantt chart I created was ultimately very useful for making this critical self-analysis.

However, in thyristor-controlled bridge rectifier the ***ripple factor*** increases with the firing angle.
https://pdfs.semanticscholar.org/3a03/2a4d523d0737912be4c73863d71181d99699.pdf

Inrush Current – Power Electronics Handbook, Pg 166

CONCLUSION

Focus of the project and even the title changed a little.
Considerations in rectifier circuits being applicable to many other analogue designs in general.
Personal Objectives beyond this project (continuous professional development)

- High speed/frequency power applications
- RF power applications
- Digital signal processing
- Control systems

Design objective to be supported by well-established theory and research.
Emphasize learning curve and learning difficulties.
Personal objective: job market skill sets.

Power Electronics

- System modelling
- Analog calculations
- PCB Layout
- Electromagnetics and EMC
- Standards

Digital System Design

- HDL coding
- FPGA development
- Timing Analysis

Learning curves
Significance of timing analysis – max/critical delay path
Potential analogue circuit improvements – emi filter, resonant DC-DC converter, voltage regulator module.
Xilinx Learning Resources and Simulink learning resources (limited)

Design Shortcomings

Zero Crossing Detector not created.
PCB Delay paths not known
Digital delays
Real Transformer modelling not performed
SCR/Thyristor Firing pulse requirements not specified
Harmonic and transient analysis was limited

Potential improvements:

- User adjustable firing angle using input medium (keypad or rotary switch)
  - Analog-to-Digital conversion
- Feedback control (control systems) to automatically adjust firing angle or phase delays
  - Writing a transfer function for the system https://www.youtube.com/watch?v=14cMhrp5wlk
  - Analog-to-Digital conversion
    - ADCs: Discrete controllers measuring continuous signals… questions of resolution? Introduces quantisation error – trade off between number of bits required for sotring values, range of values that can be measured and the quantisation error that is acceptable.

Design validation through prototyping

Challenges

- Perhaps the biggest challenge was creating phase shifting from the incoming in each pair of TGC firing pulses.
- Simulator modelling

- Cosimulation modelling

No.	Task Name	Start	Finish	Duration	Rated Difficulty (1 - 10)
1	Consultation			6	
	Rectification Research				
2	Acquisition of literary sources	02-Jun	04-Jun	0	5
3	Note taking and paraphrasing	02-Jun	07-Jun	6	
	Digital Design Research				
4	Develop Psuedo Code	08-Jun	09-Jun	0	6
5	Review Study Online Training Modules	08-Jun	13-Jun	6	
	VHDL Program Writing				
6	Write Functional VHDL Code	14-Jun	01-Jul	18	8.2
7	IO Planning	30-Jun	01-Jul	2	
8	Timing Analysis	02-Jul	12-Jul	11	
9	Rectifier Calculations	08-Jun	10-Jun	3	
	Uncontrolled Rectifier Modelling				8.5
10	Learning simulink modelling, tools, packages, paran	13-Jul	10-Aug	29	
11	Uncontrolled Rectifier Simulation & Analysis	11-Aug	16-Aug	6	
12	Controlled Rectifier Modelling	17-Aug	19-Aug	3	
	Controlled Rectifier Cosimulation & Analysis				
13	Modelsim setup & licensing	20-Aug	20-Aug	1	
14	HDL Verifier add-on setup & licensing	20-Aug	20-Aug	1	9.7
15	HDL Verifier Configuration	21-Aug	20-Sep	31	
16	Cosimulation and Analysis	21-Aug	03-Sep	14	
	Report Writing				
17	Literary Review	05-Jun	09-Jun	5	
18	Methodology	08-Jun	22-Sep	10	6.5
19	Referencing	10-Jun	25-Sep	5	
20	Others - Intro, Discussion, Critical Analysis, Conclus	02-Jun	26-Sep	9	
21	Final Editing & Structuring	23-Sep	27-Jul	5	

Figure 114. Challenges

My work has only just begun.

Appendices

[WATCH → https://www.youtube.com/watch?v=K2LRXsKCW7w]

[topdowndesign.pdf]

(pdf here)

http://www.tkt.cs.tut.fi/tools/public/tutorials/mentor/modelsim/getting_started/gsms.html

[Digital Circuit Analysis &Design with Simu, Pg Appendix D-1~ VHDL Design Approach]
[VIVADO Digital coding guidelines https://www.xilinx.com/member/customer-training/introduction-digital-coding-guidelines.html]

PDF
Videos1
Video2 – Multiple simulation sets

https://www.youtube.com/watch?v=oscdJUjcR9I
https://www.youtube.com/watch?v=B73G4BuTpLo
https://www.youtube.com/watch?v=iwAnGWS1-h4
https://www.youtube.com/watch?v=pEj6LR-C84Y
https://www.youtube.com/watch?v=2V71i-ubG0U

https://uk.mathworks.com/help/Simulink/ug/types-of-sample-time.html
[https://www.xilinx.com/member/customer-training/io-constraints-virtual-clocks.html]

(LUT1 - PDF_Page315)

[Introduction to Modern Power Electronics, Pg 137&138 ~ Input Current & PF]
[Power Electronics Handbook, Pg 524 ~ Power Factor Definition]
[Introduction to Modern Power Electronics, Pg 14 ~ Power Efficiency & Conversion Efficiency]
[Introduction to Modern Power Electronics, Pg 15 ~ Power Factor]
[Introduction to Modern Power Electronics, Pg 136 ~ PF Waveforms]

STANDARDS
https://webstore.iec.ch/publication/28164
https://standards.ieee.org/standard/519-1992.html

[Power Electronics Handbook, Pg. 150-153 ~ 1-Ph Full & Half-Wave Rectifiers]

Nikunj Shah, Siemens
https://www.industry.usa.siemens.com/drives/us/en/electric-drives/ac-drives/Documents/DRV-WP-drive_harmonics_in_power_systems.pdf

Yaskawa America – must find printed academic source for relevant video

Reference missing for power factor!