People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Power and Control

Final Year Project Report Presented in Partial Fulfilment of the Requirements for the Degree of

MASTER

In **Power Engineering**

Option: **Power Engineering**

Title:

Design and Implementation of an Overcurrent Digital Relay

Presented by:

- **MEDJBER Brahim**
- **BENMOUFFOK Youcef**

Supervisor:

Dr. METIDJI Brahim

Registration Number:……/2018

Dedication

Every challenging work needs self-efforts as well as guidance of Elders those who were very close to our heart.

My humble effort I dedicate to my sweet and loving

Mother and my **family members**,

Whose affection, love, encouragement and prays of day and night made me able to get such success and honor.

Along with all **my friends**, **hardworking** and respected **Teachers.**

Medjber Brahim

Dedications

It with great pleasure that I dedicate this humble work to my mother and father, brothers and sister, to my big family and all my friends whom I consider part of it.

I also want to dedicate this to anyone who taught me something new no matter how small the idea was, as it has contributed in making into me who I am today.

Youcef BENMOUFFOK

ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful
Alhamdulillah, all praises to Allah for the strengths and the blessings he gave us to complete this project.

We would like to express our deepest and sincere gratitude to our project Supervisor Dr. B.METIDJI. We have been very fortunate *and honored to work and study under your supervision, we would like also to thank the other teachers for their precious help during our work. Thank you very much.*

Last but not least, we are infinitely grateful to our family members, particularly our parents for their patience, unwavering support, continuous encouragement, and belief in us throughout our whole lives. We would have never made it this far without them beside us every step of the way.

Abstract

A model based design approach has been applied in this thesis for the development of an overcurrent protection relay, having both the instantaneous characteristics and the inverse time current characteristics as per IEEE standard C-37.112-1996. (ANSI 50, 51). This approach highly accelerates and simplifies the design.

MATLAB Simulink has been used for the development of the model and Simulink HDL coder has been used to generate the VHDL code. The code generation is supported by keeping certain constraints like fixed point data type in the design during system development. This generated code is used for verification and implementation of the designed model. The digital simulation of the generated codes and test bench has been performed on ModelSim Digital Simulator.

In this work, a hardware platform for power line communications was implemented to be used with the relay and give it a communicative functionality. This allows the relay to coordinate its decisions with its adjacent peers enhancing its selectivity.

Table of Contents

List of Figures

List of tables

List of Abbreviations

ADC — Analog to Digital Converter.
ASIC — Application Specific Integrated Circuit.
CAD — Computer Aided Design.
CPLD — Complex Programmable Logic Device.
CT — Current Transformer.
DFT — Discrete Fourier Transform.
EDA — Electronic Design Automation.
FPGA — Field Programmable Gate Array.
GPS— Global Positioning System.
GUI — Graphical User Interface.
HDL — Hardware Description Language.
IC — Integrated Circuit.
IEEE — Institute of Electrical and Electronics Engineers.
IGBT— Insulated Gate Bipolar Transistor.
I/O — Input/Output.
PLC — Power Line Communication.
PMU— Phasor Measurement Unit.
RMS — Root Mean Square.
TDS — Time Dial Setting.
VHDL — Very high speed integrated circuit Hardware Description Language.

VT — Voltage Transformer.

Introduction

For the few past decades, the human race has become strongly dependent upon technology, for it having infiltrated our lives in such a deep and complex manner that its continuity became a must. To ensure this continuity, huge investments have been made in the power domain from generation to distribution to the consumer's level.

A good part of those investments and researches were, and still are being made in the protection field, for it being the most critical one in the whole power system, since any failure in any subsystem of the latter, can be mitigated in terms of damage, if the protection system is working properly, whereas a failure in the protection system itself can lead to serious losses.

Protective relays form the spine of the power protection system, having first been introduced as simple electromechanical devices, they have now evolved to the more complex, smart and practical numerical type, complex for using intricate algorithms, smart for their use of microcontrollers, and practical for offering various types of protections, being very configurable and adaptive, and the communicative option most of them have opened the door to various new applications.

The numerical relay discussed in this project is of the overcurrent type, one of the oldest and most widely used electrical protective devices, and that is due to the fact that the vast majority of faults result in increase in current. This led the engineers who pioneered the concept of protective relays to think of setting a current threshold for the system that is to be protected and cut its power off whenever it is exceeded.

Throughout its four chapters, this work explains how to make a digital overcurrent relay, and equip it with the option of power line communications.

Chapter one will present theoretical background and give a general overview about overcurrent protection. Then comes the second chapter in which requirements of protection systems were highlighted presenting some reviews on the history of protective relays with a focus on digital relays and their general structure. Simulation of the designed model was done using Simulink and Modelsim, and the obtained results were discussed in chapter three. The last chapter will give a detailed walkthrough of the implementation process from signal conditioning to communication.

CHAPTER I

Overcurrent Relaying Philosophy

Overcurrent Relaying Philosophy

1.1 Introduction

This chapter is going to talk about power systems and their protection focusing on Overcurrent. The protective elements in overcurrent protection, namely Overcurrent relays, are discussed presenting their basic types.

1.2 Power System Protection

System protection is the art and science of detecting problems with power system components and isolating these components. The problems that power systems face are: Short circuits, abnormal conditions (Sag, Swell, Surge and Interruption), equipment failures.

1.3 Overcurrent Protection Schemes

Transmission and distribution systems are exposed to overcurrent flow into their elements. In an electric power system, overcurrent or excess current is a situation where a larger than intended electric current exists through a conductor, leading to excessive generation of heat, and the risk of fire or damage to equipment. Possible causes for overcurrent include short circuits, excessive load, transformer inrush current, motor starting, incorrect design, or a ground fault. Therefore, for normal system conditions, some tools such as demand - side management, load shedding, and soft motor starting can be applied to avoid overloads. In addition, distribution systems are equipped with protective relays that initiate action to enable switching equipment to respond only to abnormal system conditions. The relay is connected to the circuit to be protected via CTs and VTs according to the required protection function. [1]

1.3.1 Protection relay

Protection relay is a smart device that receives data compares them with reference values, and delivers results. Incoming data can be current, voltage, resistance or temperature. Results can include visual information in the form of indicator lights and/or alphanumeric displays, communications, control warnings, alarms, and power on and off. The diagram below answers the question of what is the protection relay. [1]

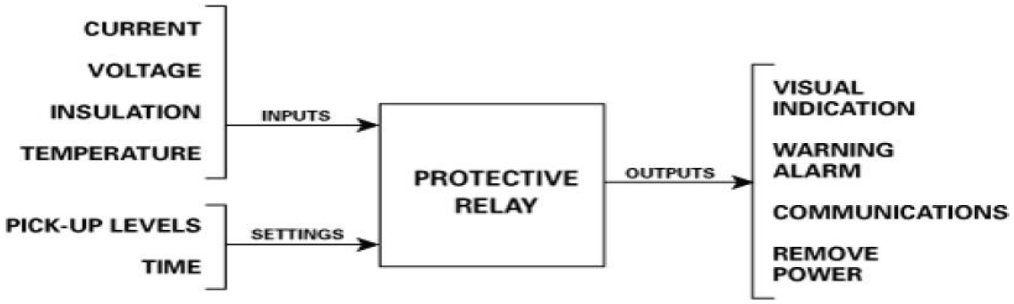

Figure 1.1 Incoming and Outgoing Data of a Relay.

1.3.2 Overcurrent relay

The basic element in overcurrent protection is an overcurrent relay. The ANSI device number is 50 for an instantaneous overcurrent (IOC) or a Definite Time Overcurrent (DTOC) and 51 for the Inverse Definite Minimum Time. There are three types of operating characteristics of overcurrent relays [1]:

- Definite (Instantaneous)-Current Protection,
- Definite-Time Protection and
- Inverse-Time Protection.

1.3.2.1 Definite (Instantaneous)-Current Protection

This relay is referred as definite (instantaneous) overcurrent relay. The relay operates as soon as the current gets higher than a preset value. There is always an inherent time delay of the order of a few milliseconds. [1]

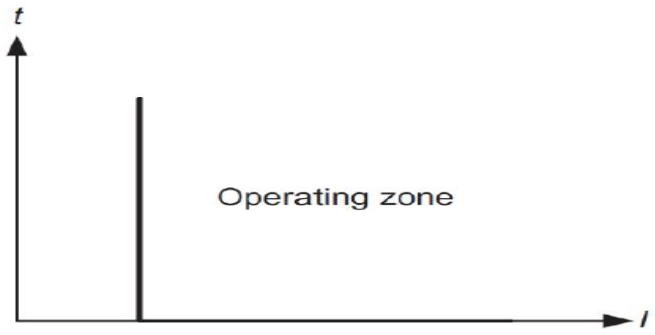

Figure1.2 Definite-Current Characteristic. [1]

1.3.2.2 Definite-time protection

In this type, two conditions must be satisfied for operation (*tripping*), current must exceed the setting value and the fault must be continuous for at least a time equal to the time setting of the relay. This relay is created by applying intentional time delay after crossing pick up value of the current. A definite time overcurrent relay can be adjusted to issue a trip output at definite amount of time after it picks up. Thus, is has a time setting and pick up adjustment. [1]

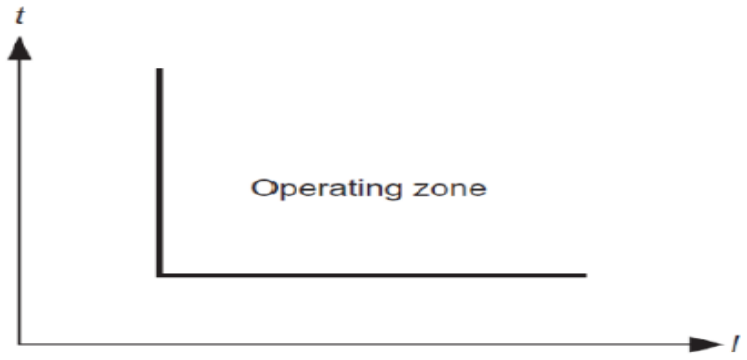

Figure1.3 Definite-Time Characteristic. [1]

1.3.2.3 Inverse-Time Protection

In this type of relays, operating time is inversely changed with the current. So, high current will operate overcurrent relay faster than lower ones. They are available with standard inverse, very inverse and extremely inverse characteristics. Inverse Time relays are also referred to as Inverse Definite Minimum Time (IDMT) relay. The operating time of both overcurrent definite-time relays and overcurrent inverse-time relays must be adjusted in such a way that the relay closer to the fault trips before any other protection. This is known as time grading. The difference in operating time of these two relays for the same fault is defined as discrimination margin.

The adjustment of definite-time and inverse-time relays can be carried out by determining two settings: time dial setting and pickup setting. The time dial setting adjusts the time delay before the relay operates whenever the fault current reaches a value equal to, or greater than, the relay current setting. The time dial setting is also referred to as the time multiplier setting. The trip time is determined following a set of equations defined by IEEE or IEC standards. These equations allow us to draw different curves

depicting the inverse property of these relays. More on IDMT relays and its characteristics will be discussed in chapter 3. [1]

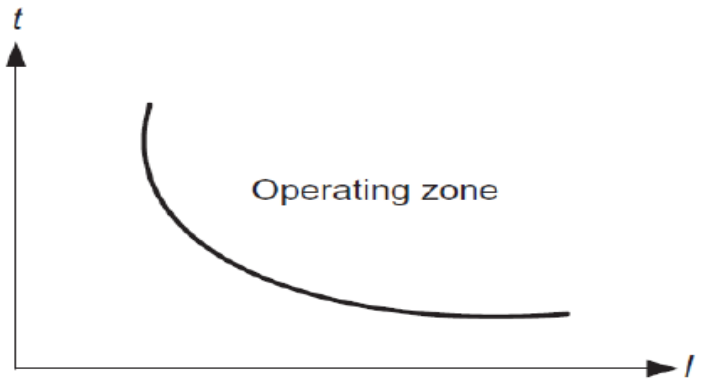

Figure1.4 Inverse-Time Characteristic. [1]

1.4 Conclusion

In this chapter, power protection has been introduced with a focus on overcurrent protective functions.

CHAPTER II

Digital Relays

Digital and Numerical Relays

2.1 Introduction

Protective relay technologies have been evolving over the last decades concomitantly with the developments in analog and digital computing technologies, and the requirements of power systems. In this chapter we are going to highlight the requirements of protection systems, review the history of protective relays with a focus on digital relays where we are going to discuss the general structure of them and their main parts.

2.2 Requirements of Protection Systems

2.2.1 Reliability

The most important requisite of protective relay is reliability. They remain inoperative for a long time before a fault occurs; but if a fault occurs, the relays must respond instantly and correctly. [2]

2.2.2 Selectivity

The relay must be operated in only those conditions for which relays are commissioned in the electrical power system. There may be some typical condition during fault for which some relays should not be operated or operated after some definite time delay hence protection relay must be sufficiently capable to select appropriate condition for which it would be operated. [2]

2.2.3 Sensitivity

The relaying equipment must be sufficiently sensitive so that it can be operated reliably when level of fault condition just crosses the predefined limit. [3]

2.2.4 Speed

The protective relays must operate at the required speed. There must be a correct coordination provided in various power system protection relays in such a way that for fault at one portion of the system should not disturb other healthy portion. Fault current may flow through a part of healthy portion since they are electrically connected but relays

associated with that healthy portion should not be operated faster than the relays of faulty portion otherwise undesired interruption of healthy system may occur. Again if relay associated with faulty portion is not operated in proper time due to any defect in it or other reason, then only the next relay associated with the healthy portion of the system must be operated to isolate the fault. Hence it should neither be too slow which may result in damage to the equipment nor should it be too fast which may result in undesired operation. [4]

2.3 History of Protective Relays

In the industry, protective relays have experienced mainly three generations of evolution [5, 6]: electromechanical (EM) relays, solid-state (SS) relays, and numerical relays.

2.3.1 Electromechanical Relays (EM)

EM relays were based on moving parts to perceive abnormal changes of current or voltages to generate the mechanical torque. In EM relays, the actuating forces are created by a combination of the input signals, stored energy in springs and dashpots. Therefore, this type of relay contains an electromagnet and a moving part. When the actuating quantity exceeds a certain predetermined value, an operating torque is developed and applied on the moving part. The torque will then cause the moving part to finally close a contact in order to energize the trip coil of the circuit breaker.

2.3.2 Solid State Relays

Solid State relays based on analog electronic devices such as transistors, diodes, and other electronic components, were the static replacements of EM relays which contains no moving elements. All of the functions and characteristics available with electromechanical relays can be performed by solid-state devices. SS relays use low power components with rather limited capability to tolerate extreme environmental factors or overvoltages and overcurrents. In general, SS relays are more accurate than EM relays and have the advantages of reduced size of solid-state devices. Furthermore, absence of mechanical moving component in SS relays which may normally bring noise and cause

contact problem, and less maintenance is also an improvement compared to the electromagnetic relays.

2.3.3 Digital Relays

Currently, most commercial relays are digital relays based on microprocessor technology, and sequential software programmability. This type of relay is introduced with the development of VLSI (Very Large Scale Integration) technology and fast microprocessors. The core of this type of relay is digital signal process (DSP) microprocessor. The usual relay inputs are power system voltages and currents. Analog signals will first be filtered and converted to the digital form by analog-to-digital converter (ADC). Then the relaying algorithm processes the sampled data to generate a digital output. The most obvious advantage of the microprocessor based relay is the flexibility due to their programmable approach. Various protection functions can be provided by numerical relays at low cast and compete with conventional relays.

2.4 General Structure of Digital relays

The generalized digital relay concept, which is directly derived from open system relaying, consists of a minimum set of hardware modules and functions of modern digital and numerical relays. **Figure 2.1** shows a diagram of the main parts of the digital relay proposed in this project.

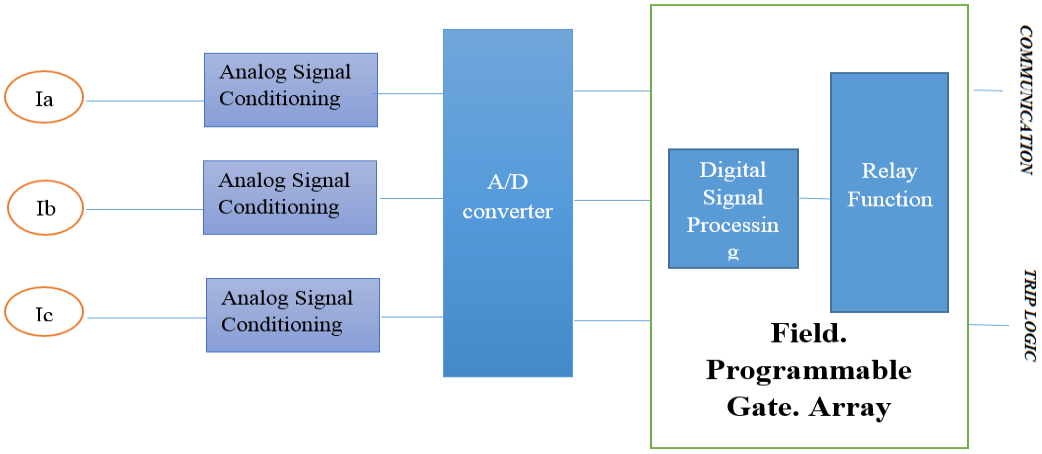

Figure 2.1 General Structure of the Proposed Numerical Relay.

The following hardware modules and functions constitute the generalized numerical relay:

- **Isolation and analog signal conditioning:** Current waveforms from current transformers are acquired and scaled down to convenient voltage levels for use in the digital and numerical relays. [7]

  - It is common for digital relays to use anti-aliasing filters before sampling an analog signal. That is because higher frequencies that are introduced with noise will raise the sampling rate needed (Fs) which, according to the Nyquist theorem, need to be at least twice as big as the highest frequency of the sampled signal. Since we are going to use a high sampling frequency (much bigger than the highest frequency that might be introduced), ant-aliasing filters are not going to be used.

- **Analog-to-digital conversion:** Digital processors can process numerical or logical data only, therefore the waveforms of inputs have to be sampled at discrete times. For this, each analog signal is passed through an Analog-to-Digital Converter (ADC) chip that is programmed to sample the inputs at the desired sampling rate.
- **Digital Signal Processing:** The current samples need to be digitally processed in order to extract the information needed for the relay function to make a decision. The processing is basically done using digital filters to extract the fundamental frequency component out of the input samples. The obtained samples then are used to calculate the RMS values of the currents, the values that are going to constitute the inputs to the relay function block.
- **Relay Function:** It contains an algorithm that makes the trip decision based on predefined values compared with input values.
- **Communication:** The issued trip signal must be transmitted to the other relay(s) in the protected area using pilot wire communication schemes.

2.5 Overcurrent Relay Algorithm

Several techniques have been used to model and design overcurrent relays over the years, each of them based on some mathematical equations describing the behavior of the relay and its time-current characteristics. In this thesis a technique is presented based on IEEE Standard, Inverse-Time Characteristic Equations for Overcurrent Relays.

2.5.1 Analog Part

For the relay to take any sort of action, it needs to gather data, readings of the real time values of voltage and current in the three phases. In this theses, only the values of the current in the three protected phases need to be read since it is an overcurrent relay.

Current sensors (or current transformers) are installed in the zone or device that needs to be protected, and are therefore external to this relay and not part of, but since their output values are standardized (either 1A or 5A), the signal conditioning circuitry can be designed regardless of the sensor.

The brain of this relay is an FPGA, it can't take currents of 1A and process them, neither can it process any current or any analog entity, thus, these currents need to be transformed into voltages of appropriate levels, and then these voltages need to be digitalized by an analog to digital converter, of which the sample rate and the resolution are selected depending on the performance expected from the relay.

The ADC used accepts values between 10V and -10V, so the output current of the CTs needs to be transformed to voltages oscillating in this interval, to reach this final stage many factors need to be taken into consideration:

- a safety margin should be left in the ADC's input, in order to be able to detect current surges and short circuit values, thus, if 1A RMS corresponds to the nominal current in our protected line, its peak value of 1.41A should be converted to voltage 5V peak instead of 10V, thus leaving 50% of the ADC's capability on standby as a safety margin.
- When converting currents to voltages, resistors were used, but the power consumption of these resistors should be taken into consideration since it translates to how much heat is dissipated by them, an effect that should be minimalized as much as possible. To do so, small resistors have been chosen (in the order of 0.5 ohm) and then operational amplifiers were but keeping the first constraint satisfied.
- When dealing with high frequencies (if high order harmonics need to be analyzed), the slew rate of the used amplifiers should also be taken into consideration, otherwise distortions would be introduced to the signal by the amplifiers themselves.

2.5.1.1 The Output of the Relay

If a fault occurs and gets detected by the relay, it needs to be isolated to avoid further damage, this is done using circuit breakers.

To transfer the action from the relay (FPGA in this case) to the circuit breaker, the trip coil of the latter must be energized, and that is done by closing its circuit (located in the circuit breaker) with a battery through a switch relay.

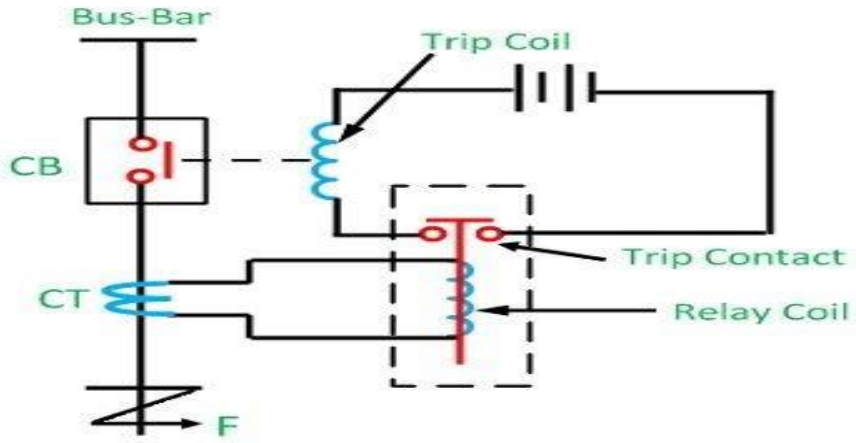

Figure 2.2 Connection of the Relay to the Contact of the Circuit Breaker. [8]

2.5.1.2 Communication Scheme:

A) Overview

To integrate smart grids, all modern digital relays must be communicative, and this for two main reasons:

Be able to communicate with other relays: to implement a pilot relaying scheme, which is a communication network implemented on the high voltage transmission line (T-line) to transmit "trip or don't trip" signal to and fro between two or more substations. The intent here is to trip the selected few circuit breakers as fast as possible when a fault strikes the T-line, therefore, protecting it.

Be able to communicate with dispatch or a monitoring center: to send real time values of voltage, current, power and other useful parameters (the relay can serve as a measurement device).

In this implementation, power line communications (PLC) was used, it is a communication technology that enables the user to send and receive data over the existing power lines, which makes of it an inexpensive and flexible technique as it can readily be used on any connected network.

B) Principle of Working of Power Line Communication

The information that is to be sent is generated in a digital form (square wave), encoded by any protocol suitable for the application, this signal is then modulated by a high frequency carrier and is injected to the power line by means that will be later on explained.

On the receiving end, a demodulator is placed and tuned to the carrier's frequency, the signal is then recovered and can be used to fulfill whatever purpose it was sent for.

2.5.2 Digital Part

2.5.2.1 Fundamental Frequency Extraction

When a fault occurs, harmonics are introduced with the fault current. In the case of overcurrent, the 5^{th} and the 7^{th} harmonics are usually the most significant types that affect the performance of the protective relays. When harmonic current exceeds the peak or rms thresholds, it may cause the relays to trip under conditions which would normally incur smooth running of the system without interruption. Thus the need for extracting the fundamental frequency by filtering out all the harmonics present. This is done using digital filtering algorithms tuned on the fundamental frequency (50 Hz).

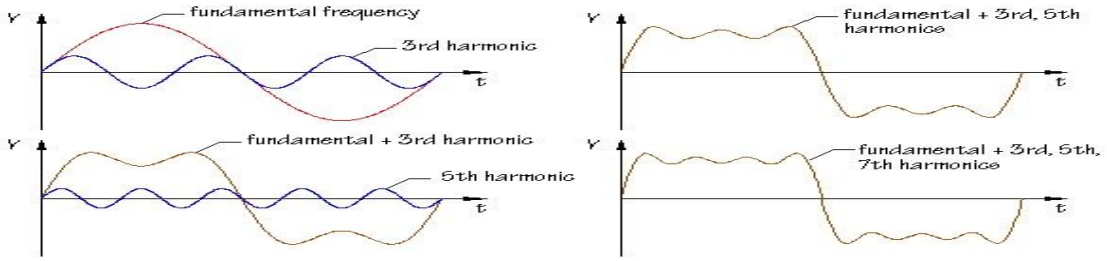

Figure 2.3 Waveforms of the Fundamental Frequency Mixed With Harmonics. [9]

Any sinusoid can be represented by phasor which is a rotating vector with a fixed amplitude, frequency and phase angle. The amplitude of the phasor is equal to the RMS

value of the sinusoid. The phase angle of the phasor is the distance of a point in the sinusoid from the reference.

A sinusoid can be given by:

$$X(t) = Xm.\sin(\omega.t + \phi) \tag{2.1}$$

ω being the frequency of the signal in radian per second, ϕ is the phase angle in radian and Xm is the peak amplitude. Fourier series representation of a function signal X (t) with N harmonics is:

$$X(t) = \mathrm{A}_0 + \sum_{k=1}^{N} A_k \ .\cos(2.\pi.k.f_0.t + \theta_k) \tag{2.2}$$

To eliminate the harmonic signals, a peak filter can be used at the frequency of 50Hz. Peak filters amplify frequencies in a narrow bandwidth around the cutoff frequency fc. FIR digital filters have better response specifications compared to IIR filters but the later are much faster, and in this domain of application, speed of operation is a critical element, besides, the nearest significant harmonic that can affect the measurement accuracy is the 5th (250 Hz) which is more than enough for an IIR filter to significantly attenuate all other frequency components.

(2.3) shows the transfer function of a peak filter:

$$H(z) = \frac{b_0(1+z^{-2})}{1+a_0z^{-1}+a_1z^{-2}} \tag{2.3}$$

Where a_0, a_1, b_0, and b_1 are filter coefficients.

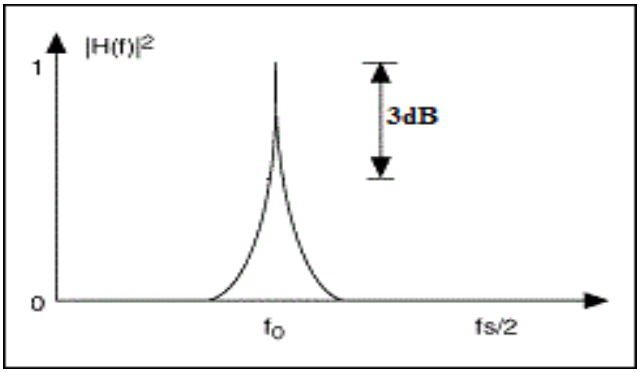

Figure 2.4 Frequency Response of a Peak Filter.

$Q = f_0/\Delta f$

Q denotes the sharpness of the peak. Increasing the value of *Q* results in a sharper peak filter.

2.5.2.2 Estimating the RMS Values

In the discrete form, the RMS value of the current form sampled signal Xm, can be obtained as:

$$I_{RMSm} = \sqrt{\frac{1}{N}\sum_{r=0}^{N-1} X_{m-r}^2} \quad (2.4)$$

Where,

$X_m = X(m.\ \Delta T)$.

ΔT = sampling interval.

N is the number of samples in a fundamental frequency cycle and it depends on the sampling rate.

$N = Fs/f_0$

'm' denotesthe m^{th} sample after the inception of a fault.

2.5.2.3 Modeling Time-Current Characteristics

Characteristics of inverse-time overcurrent relays are traditionally represented by a family of curves which depict contact closing times versus the relay currents. A time dial setting allows a desired contact closing time to be achieved for a specified operating current. Since time dial settings are continuous, it is possible to have an infinite number of characteristic curves [10]. As stated earlier in the previous chapter, IDMT relays' time-current characteristics can be modeled into equations that define sets of curves that differ from the type of the IDMT and the time dial used. The equations and the curves themselves differ from one another depending on the standard used. Starting from that, we can distinguish mainly two standards:

- IEC standard (Standard Inverse Time (SI) - Very Inverse Time (VI) - Extremely Inverse Time (EI) - Long Time Inverse).

- IEEE standard (Moderately Inverse Time (MI) - Very Inverse Time (VI) - Extremely Inverse Time (EI)).

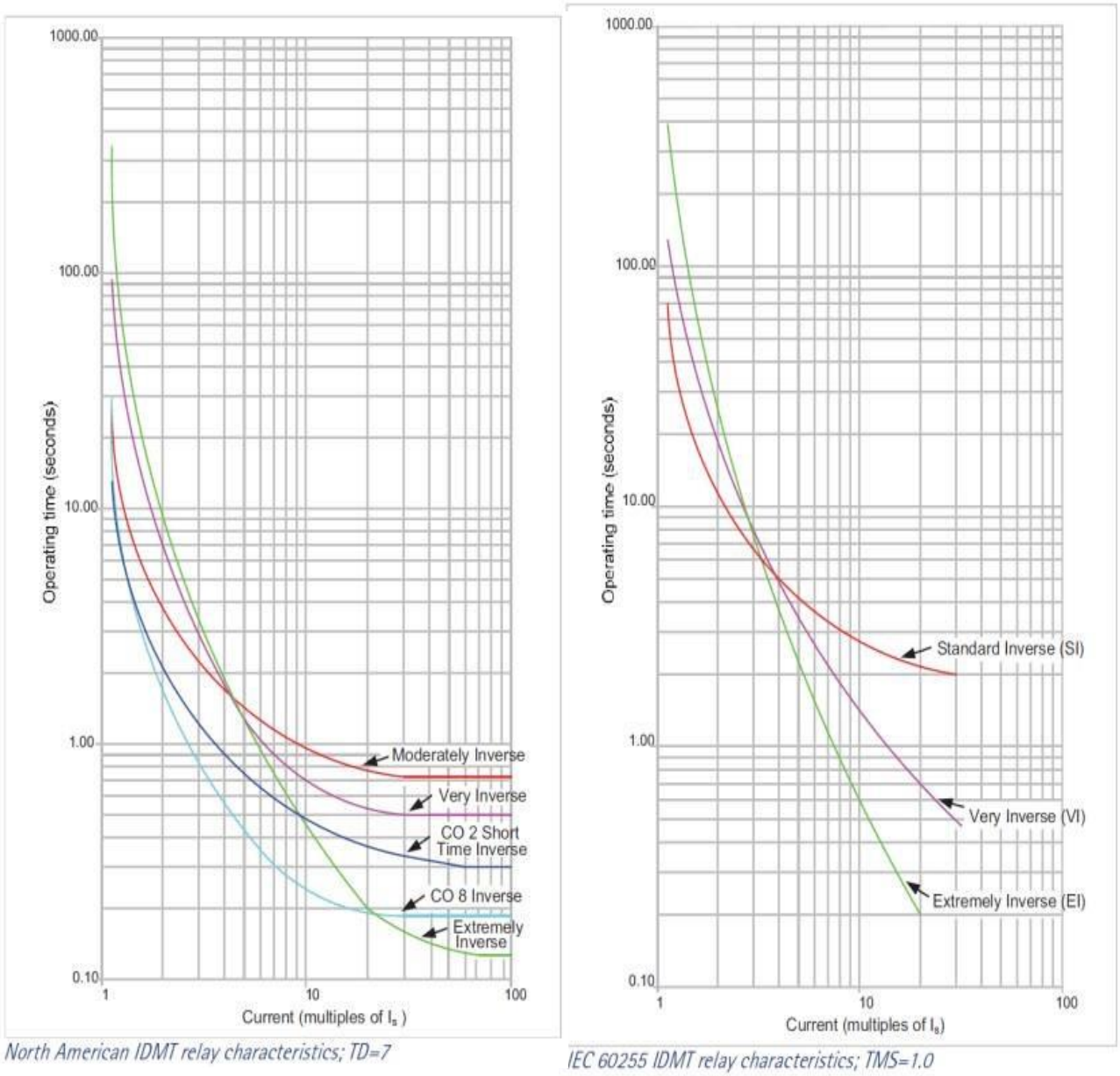

Figure 2.5 IDMT Curves According to IEEE and IEC Standards. [11]

This thesis will focus mainly on IEEE standard curves.

Time-current characteristics can be modeled using a set of equations. But before going through them, some definitions according to *IEEE Std C37.112-1996* [12] are to be stated:

- **Inverse-Time Overcurrent Relay:**

A current operated relay that produces an inverse time-current characteristic by integrating a function of current *F (I)* with respect to time. The function *F (I)* is positive above and negative below a predetermined input current called the pickup current. Pickup current is therefore the current at which integration starts positively and the relay produces an output when the integral reaches a predetermined positive set value. For the induction relay, it is the disk velocity that is the function of current *F (I)* that is integrated to produce the inverse time characteristic. The velocity is positive for current above and negative for current below a predetermined pickup current. The predetermined set value of the integral represents the disk travel, required to actuate the trip output.

- **Reset:**

The state of an inverse-time overcurrent relay when the integral of the function of current *F (I)* that produces a time-current characteristic is zero.

- **Reset characteristics:**

The time vs. current curve that depends the time required for the integral of the function of current *F(I)* to reach zero for values below current pickup when the integral is initially at the trip value.

- **Time dial:**

The time dial is the control that determines the value of the integral at which the trip output is actuated, and hence controls the time scale of the time-current characteristic produced by the relay. In the induction-type relay, the time dial sets the distance the disk must travel, which is the integral of the velocity with respect to time.

The set of equations describing the time-current characteristics are:

- Relay trip time in seconds **(M>1):**

$$\boldsymbol{t = TDS\left(\frac{A}{M^{p}-1} + B\right)} \tag{2.5}$$

- Relay reset time in seconds **(0<M<1)**

$$\boldsymbol{t = TDS\left(\frac{t_r}{M^{2}-1}\right)} \tag{2.6}$$

Where:

t = trip time (M> 1) or reset time (0<M< 1),

M = I_{input}/I_p (where I_p is pick-up current value),

TDS = time dial setting,

t_r = reset time in seconds for M = 0,

A, B, p = constants to provide selected curve characteristics.

Table 2.1 Constants Providing IEEE Standard IDMT Curves.

Characteristics	A	B	p	t_r
Moderately Inverse	0.0515	0.1140	0.02	4.85
Very Inverse	19.61	0.491	2.0	21.6
Extremely Inverse	28.2	0.1217	2.0	29.1

2.5.2.4 Emulating Relay Characteristics

After estimating the RMS value of the fundamental frequency current, the algorithm checks if the current has exceeded a threshold value. If the current is more than the threshold, the relay algorithm ensures that a command to trip the circuit breaker is issued after an appropriate time delay.

To determine the instant at which the trip command is issued, the following function is used:

$$\int y(t)dt > N_T \tag{2.7}$$

Where:

$y(t) = 0$, When the current is less than the pick-up value,

$y(t) = N_T / t_r(t)$, When the current is equal to or greater than the pick-up value.

In this equation, N_T is a target number and $t_r(t)$ is the operating time of the relay for the current observed at time *t*. The integration is started immediately after the inception of a fault i.e. when the relay current exceeds the pick-up value. The procedure is continued

until the integral $\int y(t)dt$, exceeds the target number N_T, at which time a trip command is issued. However, if the relay current, after having exceeded the pick-up value, falls below the nominal value, the process to reset the relay is started. This is done by modifying the value of the integral $\int y(t)dt$, in a manner that achieves the desired reset characteristic [11]. Because digital relays use values of currents sampled at ΔT second intervals, numerical integration can be performed using (2.8)

$$\sum_{i=0}^{k} Z_i > N_T \tag{2.8}$$

Where:

$$Z_i = N_T \cdot \Delta T / t_i \qquad \text{For (M>1)} \tag{2.9}$$

$$Z_i = -N_T \cdot \Delta T / t_i \qquad \text{For (0>M>1)} \tag{2.10}$$

t_i is the relay operating time corresponding to the RMS value of the i^{th} sample of current. The trip is generated when sum $\sum Z_i$ exceeds the target number N_T. K is the number of samples needed for the sum $\sum Z_i$ to become equal to N_T, so $K.\Delta T$ is the time to trip. t_i is the relay operating time corresponding to the RMS current estimated on receiving the m^{th} sample. If the current is less than the pick-up value, the relay is reset by reducing the value of the sum, in a manner that achieves the desired reset characteristics. The above design method gives best protection scheme as it checks the current variation at each sampling interval and hence if the current value falls below the pickup current value the reset procedure begins and it prevents the undesired tripping.

The target number N_T can be calculated using (2.11) [10]:

$$N_T > \text{int}\left[\left(\frac{t_r}{\Delta T}\right)^2 \right] \tag{2.11}$$

'int' denotes is the integer part.

2.6 Conclusion

This chapter reviewed the theoretical background that the proposed digital relay was based upon showing the basic equations based on *IEEE Std C37.112-1996* [12].

CHAPTER III

Simulation Results

Simulation Results

3.1 Introduction

Simulation of the proposed algorithm is done using Matlab/Simulink environment and Mentor Graphics Modelsim. The simulation will concern the digital part of the relay, the part that is going to constitute the program uploaded to the FPGA. The design of relay will be done using HDL optimized blocks in Simulink library. Using these blocks, Simulink's HDL coder will be able to generate the VHDL code for the whole model and that is after optimizing the whole design for code generation. This method of creating systems and models is known as Model Based Design.

3.2 Model Based Designs

Model-Based Design (MBD) is a technique to visually approach complex systems such as creation and usage of a mathematical model of a system under analysis or designing complex control. In Model-Based Design the system model is the center of the development process, from the development of requirements, to the design, implementation and validation of the system. The model becomes an executable specification that you continually refine throughout the development process. After model development, simulation shows whether the model works correctly.

Model based designing came as an alternative and solution to traditional workflows and the issues that are related to them. As far as traditional development workflows go, there exists several gaps that contribute to project delays and related costs: The first gap is the gap between the technology research and the requirements activities and the rest of the development process. These early activities are often performed in isolation from product design and the results are often thrown over the wall in the form of static specification documents. As a result, specification errors and algorithms that can be implemented remain hidden until they are more expensive to find and fix. The second gap exists between the tools and workflows used by the digital hardware and software component teams: Tools used to design and implement these components don't permit simulation of the whole system. This makes system level optimization and detection of integration problems difficult if not impossible. As a result of these gaps, system testing happens late in the development process which leads to high verification cost, expensive design rework and project delays.

3.3 Implementing the Proposed Design

The three phase relay in question has three inputs corresponding to the current samples for each phase. For simulation and design purpose, we are going to create input currents containing the fundamental, 5th and 7th harmonic of the line currents sampled at 2 kHz. The main block, namely IDMT_OC_RELAY has 4 outputs, three of them are trip signals corresponding to the 03 phases and a general trip signal that trips each time a fault happens in either of the phases.

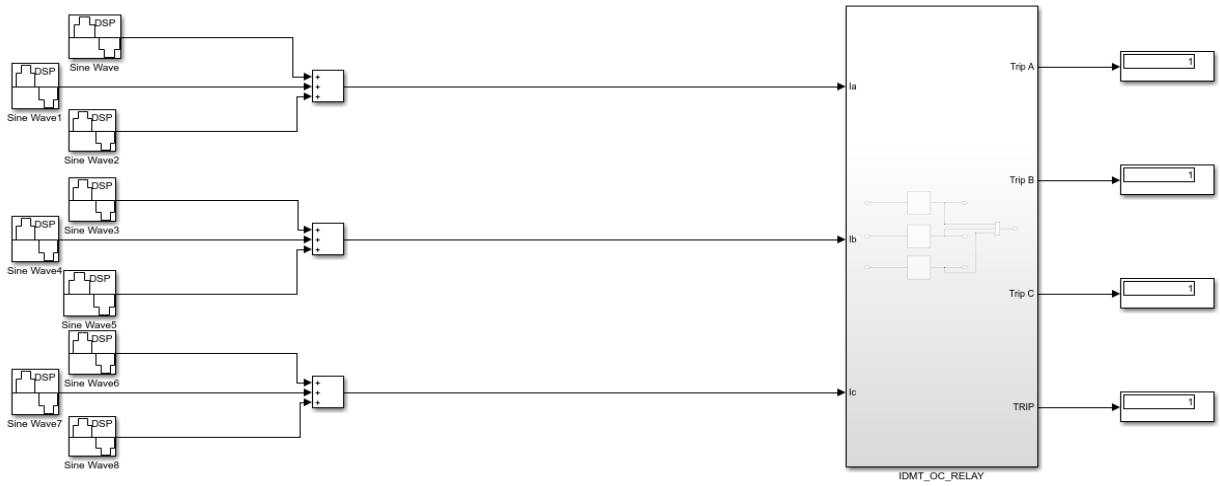

Figure 3.1 General Layout of the Overcurrent relay Implemented.

The main block contains three similar individual blocks, each for a phase.

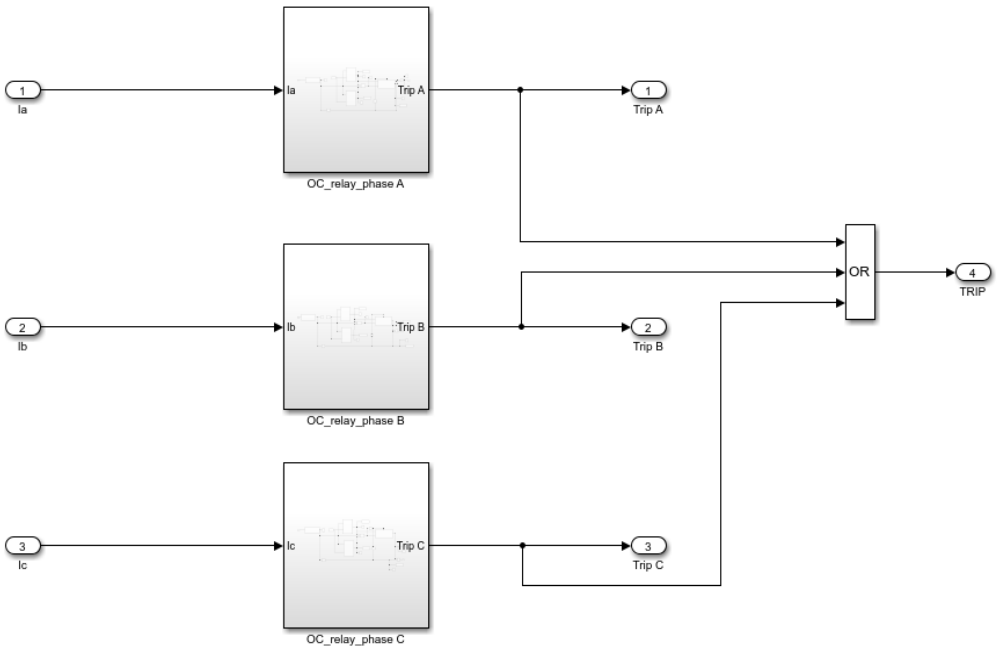

Figure 3.2 The Three Individual Relay Elements.

As the structure is the same for the three blocks, we'll take phase A as an example.

Each of the phase relays is constructed as depicted in **Figure 3.3**

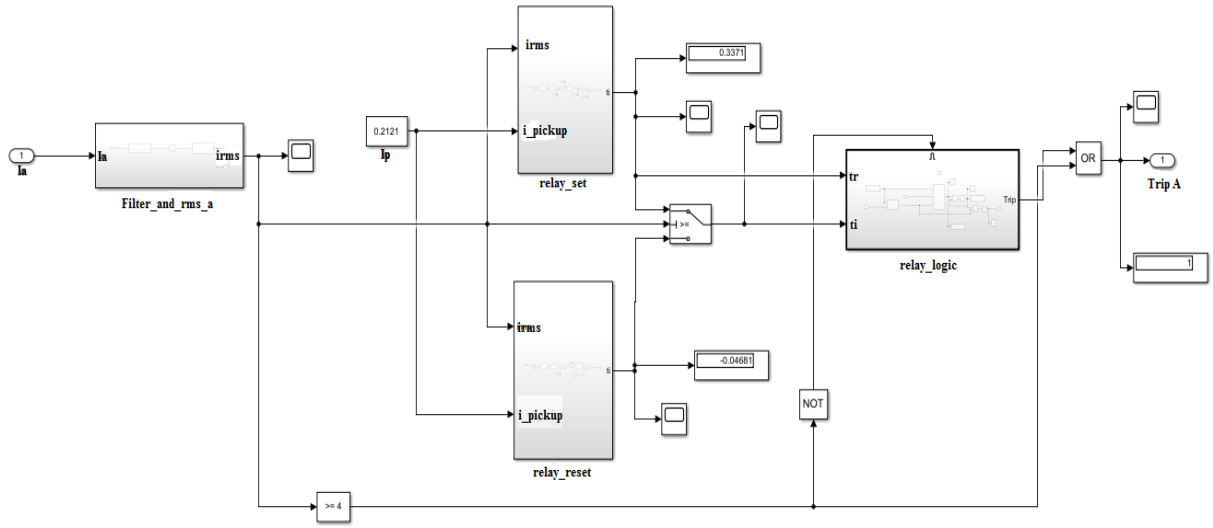

Figure 3.3 Construction of Phase Relay A.

The input current passes first through the **filter_and_rms_a** block which filters out the fundamental frequency component and then calculates its rms value. The fundamental frequency extraction is done using an IIR based second order peak filter around 50 Hz frequency with a bandwidth of 10 Hz. The filter is designed using the ***filterDesigner*** tool in MATLAB/SIMULINK. The tool can be invoked by the command *fdatool* or *filterDesigner* in Matlab command window. After the extraction is done, the rms calculation begins. The algorithm emulates the (2.4). At each sampling interval, the rms is calculated for the 40 most recent samples that constitutes the Data Window. Tapped delay block is used to output the samples as a vector to the embedded Matlab function that calculates the rms based on the previous equation as shown in **Figure 3.4**

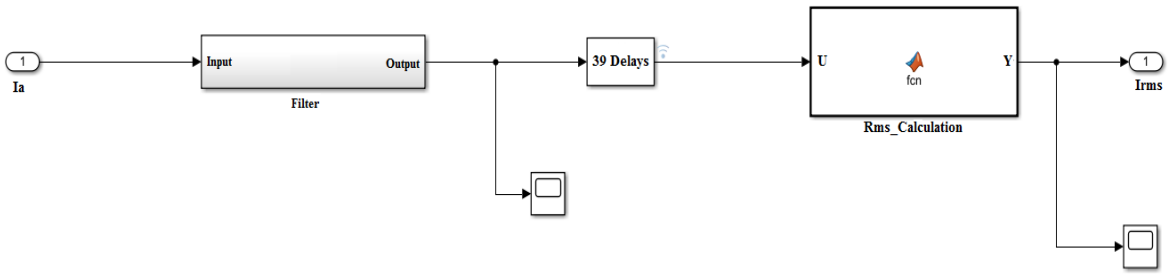

Figure 3.4 Filter_and_Rms_a Block.

The input wave form injected to the relay phase A (with amplitude 6 sampled at 2 kHz) is shown in **Figure 3.5.**

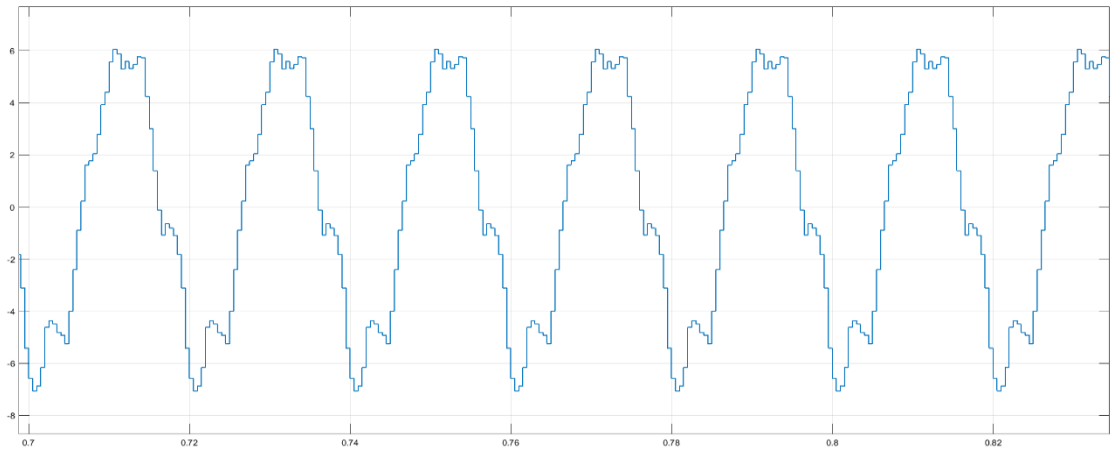

Figure 3.5 Waveform of Injected Current Ib.

After passing through the fundamental frequency extraction filter, the resultant waveform is shown in Figure 3.6

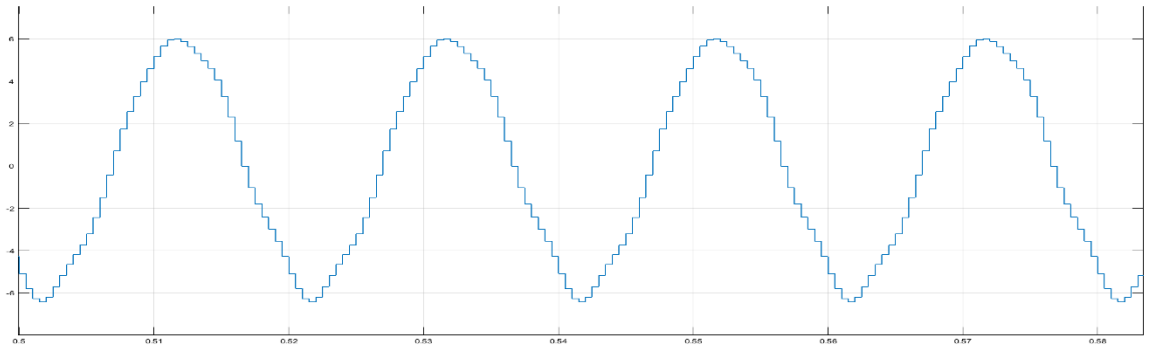

Figure 3.6 Waveform of the Filtered Current Ib.

Passing through the rms calculation phase, the output is in the form:

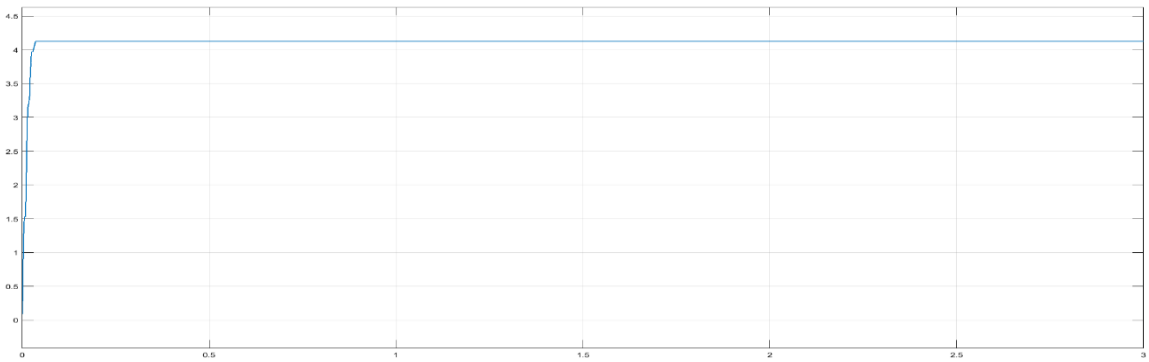

Figure 3.7 Output of the Filter_and_rms_a Block.

The calculated RMS is then used to perform relaying characteristics. The implemented relay contains both the Inverse Definite Minimum Time and the instantaneous functions. The RMS value is separately compared to two pickup currents, one for the instantaneous function and this one is usually a high value, and another one for the IDMT function. If the RMS exceeds the instantaneous pickup, the IDMT relay characteristics block is disabled and an instantaneous trip signal is issued. If not, the IDMT characteristic block is enabled to take a decision based on its inputs.

The comparison of the RMS with the pickup value for IDMT function is done using a switch that chooses the **relay_set** characteristics block if the RMS is greater than the pickup (M>0), otherwise it will switch to the **relay_reset** to engage resetting procedure. The set and reset blocks are realized using embedded Matlab functions that implement (2.5) and (2.6) respectively

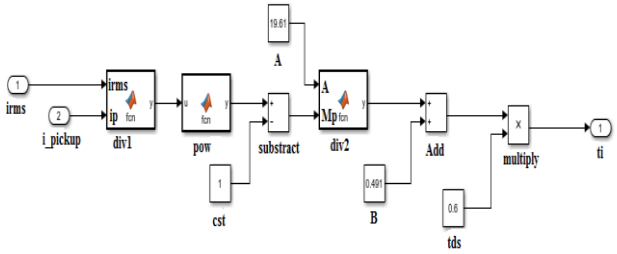

Figure 3.8 Relay_set Block.

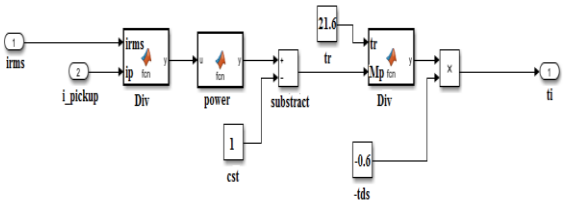

Figure 3.9 Relay_reset Block.

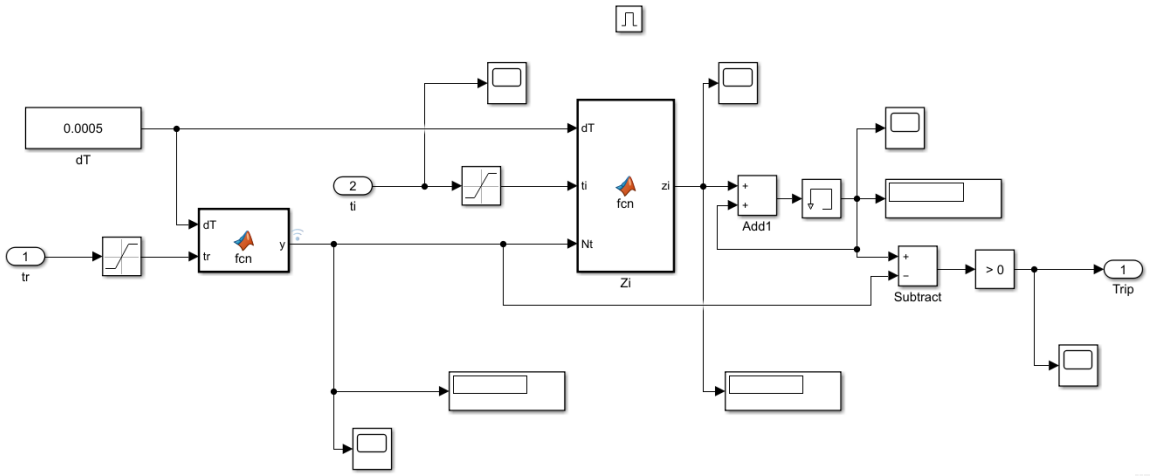

Figure 3.10 relay_trip_time_logic Block.

The set or reset time is fed to the **relay_trip_time_logic** block which emulates the delay between fault detection and trip action as shown in the above figure.

The block calculates Z_i based on (2.9) and (2.10) using an embedded Matlab function and performs the summation based on (2.8).

The target number N_T is also calculated based of (2.11) also using an embedded Matlab function. The output of which is shown in **Figure 3.11**

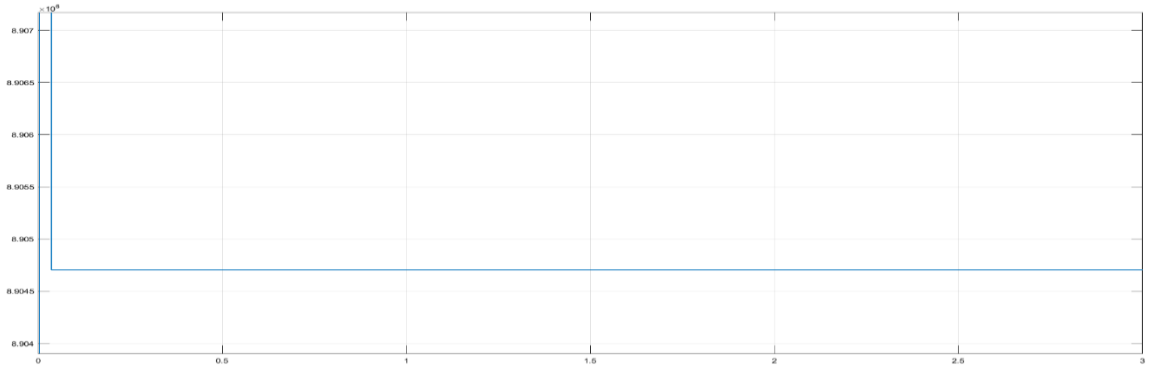

Figure 3.11 Target Number Calculator Output.

Similarly, the output of the sum $\sum_{i=0}^{k} Z_i$ is shown in **Figure 3.12**

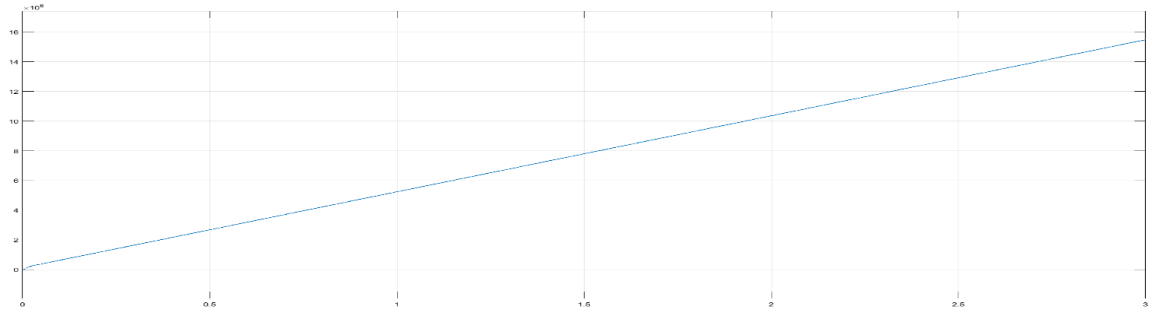

Figure 3.12 Output Sum of Z_i.

As soon as the sum exceeds the target number calculated, the relay issues a trip signal of Logic 1 as shown in **Figure 3.13**

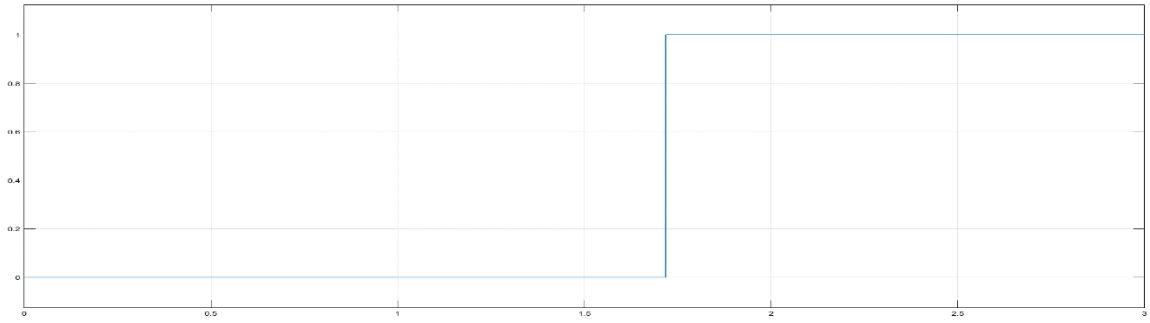

Figure 3.13 Trip Signal.

3.4 Simulation Results

The designed systems are simulated in two different environments: MATLAB and ModelSim for different values of TDS and pick-up current multiples value. The results are compared with the standard trip time as given by the **IEEE Std. 037.112-1996** for over-current relay. The Instantaneous characteristics are tested as well, giving each phase relay a different pickup value.

3.4.1 Instantaneous Characteristics

The amplitude of the fundamental frequency current injected to the relays will be 6. Different pickup currents are defined for each of the three phases relaying blocks:

- OC_RELAY_A: $I_{pickup} = 2$
- OC_RELAY_B: $I_{pickup} = 3$
- OC_RELAY_C: $I_{pickup} = 4$

The obtained results are shown in the table below:

Table 3.1 Instantaneous Overcurrent Injected and Respective Trip Times.

Relay phase	I_{peak} *injected*	I_{pickup}	*Trip time (s)*
A	**6**	**2**	**0.016**
B	**6**	**3**	**0.031**
C	**6**	**4**	**0.060**

The graphs for the trip time of the three cases are shown in the figures below:

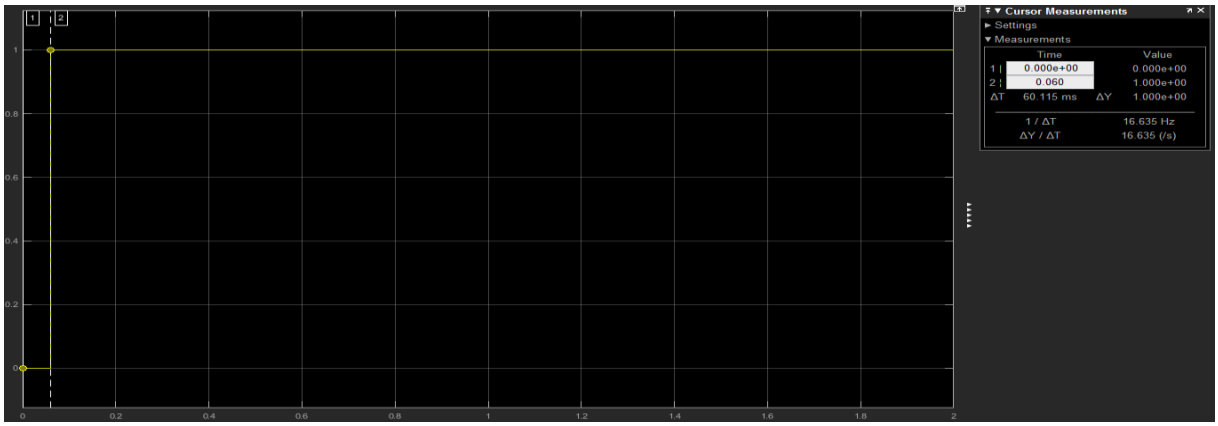

Figure 3.14 Trip Time for Instantaneous OC of Phase A.

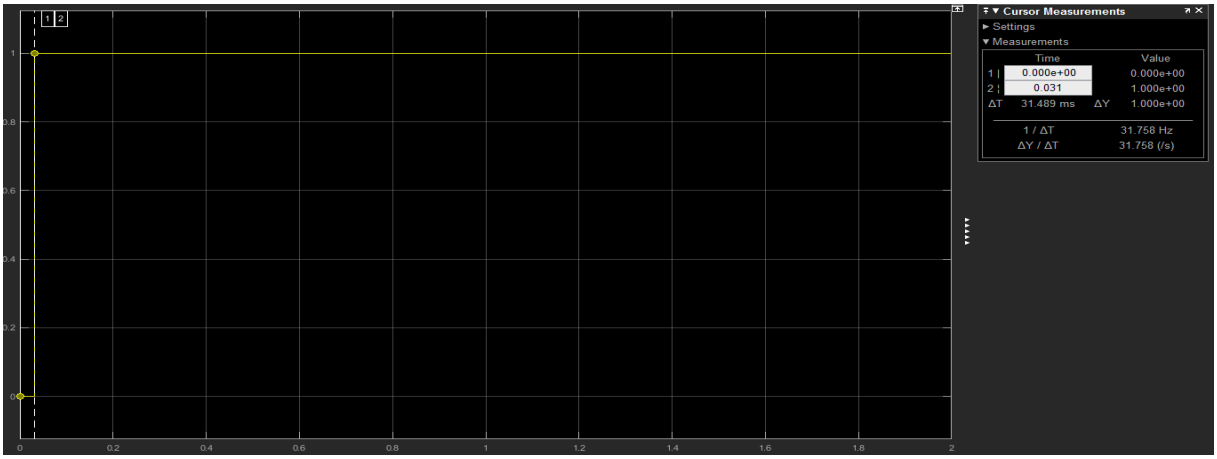

Figure 3.15 Trip Time for Instantaneous OC of Phase B.

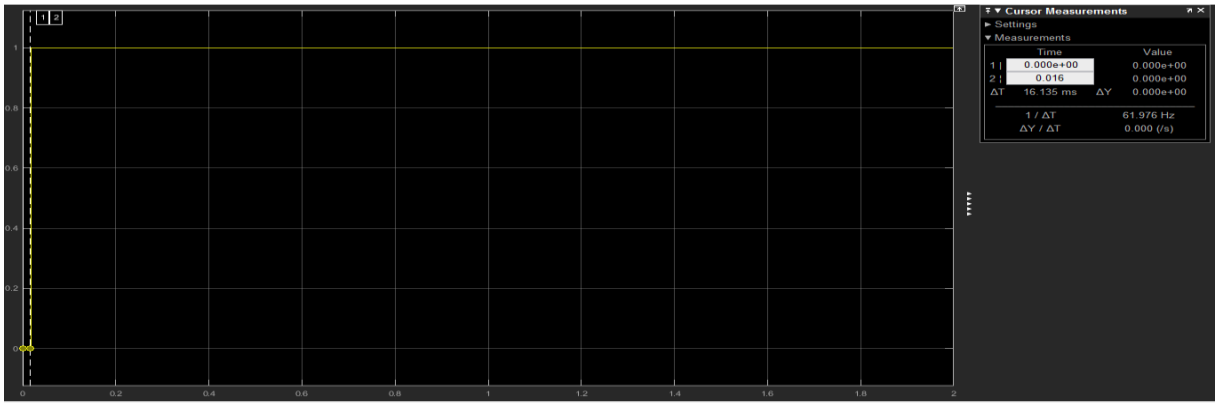

Figure 3.16 Trip Time for Instantaneous OC of Phase C.

3.4.2 IDMT characteristics

In this part, the three relay blocks will be taken advantage of, in order to investigate the trip times for 3 characteristic IDMT curves according to IEEE Std. 037.112-1996

- OC_RELAY_A will follow the ***Moderately Inverse*** curve (A = 0.0515; B= 0.1140; p= 0.02; tr= 4.85).
- OC_RELAY_B will follow the ***Very Inverse*** curve (A = 19.61; B= 0.491; p= 2.0; tr= 21.6).
- OC_RELAY_C will follow the ***Extremely Inverse*** curve (A = 28.2; B= 0.1217; p= 2.0; tr= 29.1).

Different values of TDS and pickup currents are used and the results are depicted in the tables below:

- **Moderately Inverse:**

Table 3.2 Emulated and Standard Trip Times for the Extremely Inverse Relay.

TDS	M = 2		M = 4		M = 6		M = 8	
	IEEE std	Emulated	IEEE std	Emulated	IEEE std	Emulated	IEEE std	Emulated
0.6	**2,281**	**2.288**	**1,167**	**1.157**	**0,915**	**0.893**	**0,796**	**0.759**
0.8	**3,042**	**3.049**	**1,556**	**1.550**	**1,22**	**1.204**	**1,061**	**1.028**
1.0	**3,803**	**3.815**	**1,945**	**1.939**	**1,525**	**1.512**	**1,326**	**1.303**
1.2	**4,563**	**4.581**	**2,335**	**2.331**	**1,83**	**1.822**	**1,592**	**1.573**
1.4	**5,324**	**5.348**	**2,724**	**2.721**	**2,135**	**2.127**	**1,857**	**1.842**
1.6	**6,085**	**6.110**	**3,113**	**3.112**	**2,44**	**2.440**	**2,122**	**2.110**

- **Very Inverse:**

Table 3.3 Emulated and Standard Trip Times for the Very Inverse Relay.

TDS	M = 2		M = 4		M = 6		M = 8	
	IEEE std	Emulated	IEEE std	Emulated	IEEE std	Emulated	IEEE std	Emulated
0.6	**4,216**	**4.228**	**1,079**	**1.034**	**0,63**	**0.571**	**0,481**	**0.401**
0.8	**5,622**	**5.648**	**1,438**	**1.406**	**0,841**	**0.789**	**0,641**	**0.583**
1.0	**7,027**	**7.066**	**1,798**	**1.776**	**1,051**	**1.007**	**0,802**	**0.757**
1.2	**8,433**	**8.470**	**2,158**	**2.142**	**1,261**	**1.223**	**0,962**	**0.920**
1.4	**9,838**	**9.873**	**2,517**	**2.505**	**1,471**	**1.437**	**1,123**	**1.087**
1.6	**11,244**	**11.269**	**2,877**	**2.871**	**1,682**	**1.651**	**1,283**	**1.249**

- **Extremely Inverse**

Table 3.4 Emulated and Standard Trip Times for the Moderately Inverse Relay.

TDS	M = 2		M = 4		M = 6		M = 8	
	IEEE std	Emulated	IEEE std	Emulated	IEEE std	Emulated	IEEE std	Emulated
0.6	**5,713**	**5.738**	**1,201**	**1.157**	**0,556**	**0.454**	**0,341**	**0.206**
0.8	**7,617**	**7.657**	**1,601**	**1.568**	**0,741**	**0.776**	**0,455**	**0.332**
1.0	**9,521**	**9.556**	**2,001**	**1.975**	**0,927**	**0.870**	**0,569**	**0.464**
1.2	**11,426**	**11.463**	**2,402**	**2.385**	**1,112**	**1.067**	**0,683**	**0.592**
1.4	**13,33**	**13.372**	**2,802**	**2.794**	**1,298**	**1.257**	**0,797**	**0.724**
1.6	**15,234**	**15.277**	**3,202**	**3.200**	**1,483**	**1.448**	**0,91**	**0.845**

After the realization of the model, HDL code generation can be started. But before that, data types should be converted to hardware friendly data types that is fixed-point. Within digital hardware, numbers are represented as either fixed-point or floating-point data types. Fixed-point conversion is realized using *GUI window* > *Analysis* > *Data Type Designer* > *fixed point tool*. The tool determines the minimum and maximum ranges of the model and proposes data type changes to fixed point for all the signal present in the design.

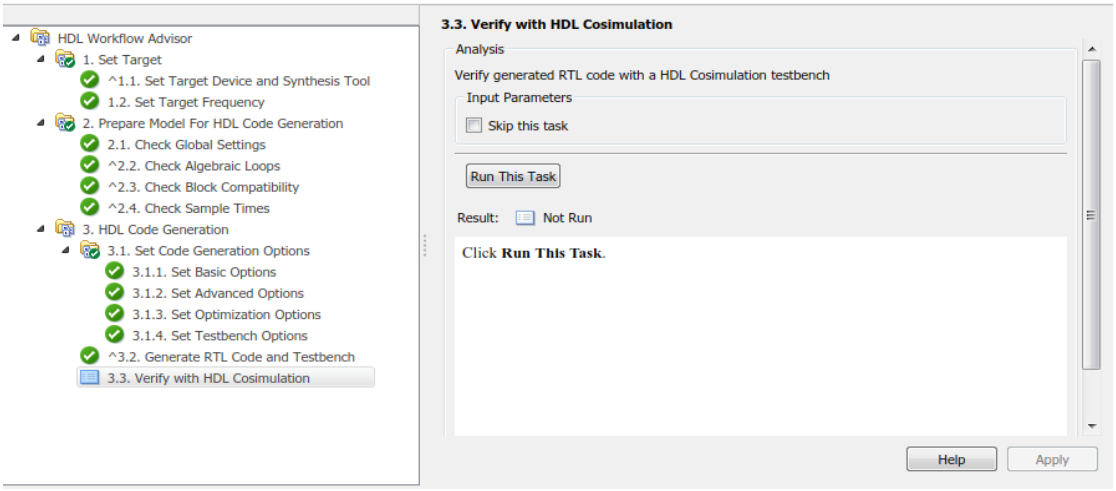

Figure 3.17 HDL Workflow Advisor.

After the conversion, checks are done to see if the model still works as before. The next step is to check if the model is HDL compatible with HDL compatibility checker that can be accessed with right click on the main block. Once that is done, code generation can begin following HDL workflow advisor from *GUI window > code > hdl workflow advisor*, through which the target device can be selected (**ALTERA CYCLONE IV E** in this case), as well as the testbench creation options and the digital simulation software (**Mentor Graphics Modelsim 10.5 se edition** in this case) .The advisor will also propose visual and naming optimization to some blocks for an optimal code generation. The HDL code is then generated along with a testbench for co-simulation with a third party software tool and that is *Mentor Graphics Modelsim 10.5 SE*.

HDL Coder generates *.vhd* files for each block present in the model as well as every embedded Matlab function used alongside a generation report where details about the generation process and the generated files are stated. The generated files are shown in **Figure 3.18**

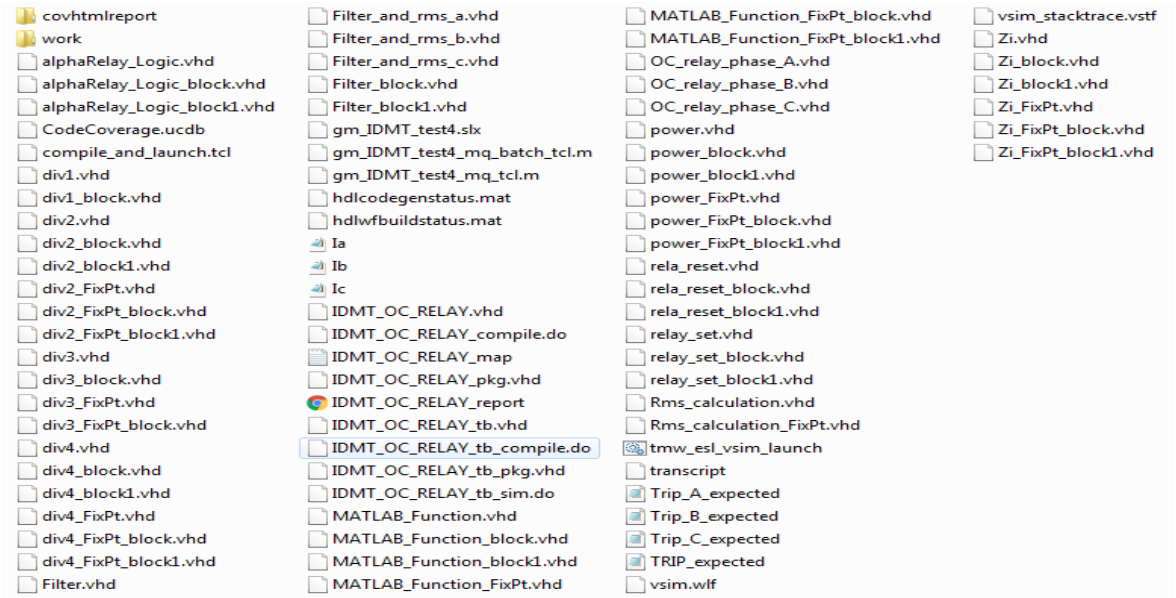

Figure 3.18 Generated hdl Files for the Designed Model.

Along with the *hdl* files, a testbench is generated with which the generated vhdl code can be tested and simulated. HDL Verifier tool namely Modelsim, contains the Cosimulation Wizard feature, which uses existing HDL code to create a customized MATLAB function (test bench or component), MATLAB System object, or Simulink HDL Cosimulation block.

The Cosimulation testbench generated for the generated code is shown in **Figure 3.19**

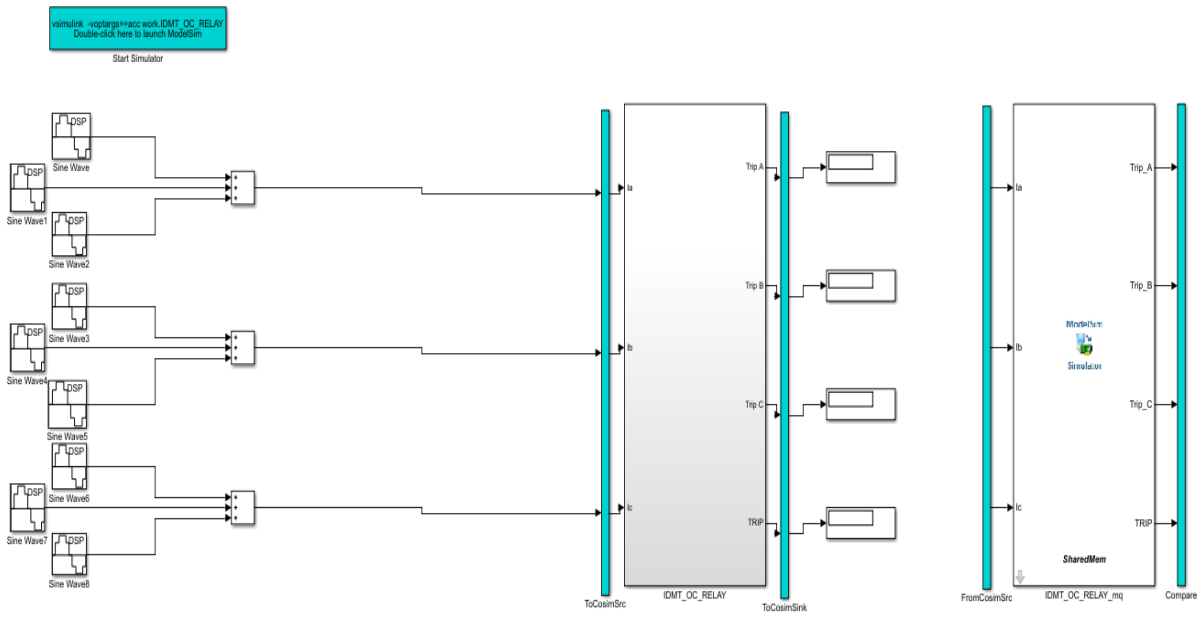

Figure 3.19 Generated Testbench and Co-Simulation Model.

By clicking on Start Simulator box on the top left of the model, the digital simulator Modelsim starts and automatically compiles the *hdl* files of the model in question and defines inputs and outputs. After the compilation is finished successfully, Modelsim is ready for co-simulation. It is to be noted that ModelSim simulation interval is from 0 ns to 60ns so the events that happen during the 3s run simulation in Simulink will be relative to that interval. For example, if a trip is to happen in 0.5s in Simulink, in the ModelSim output waveform the trip is seen in 10000ns.

- OC_RELAY_A will follow the ***Moderately Inverse*** curve with TDS = 0.6 and M = 2.
- OC_RELAY_B will follow the ***Very Inverse*** curve with TDS = 0.8 and M = 4.
- OC_RELAY_C will follow the ***Extremely Inverse*** curve with TDS = 1 and M = 6.

The output of the digital simulator is shown in **Figure 3.20**

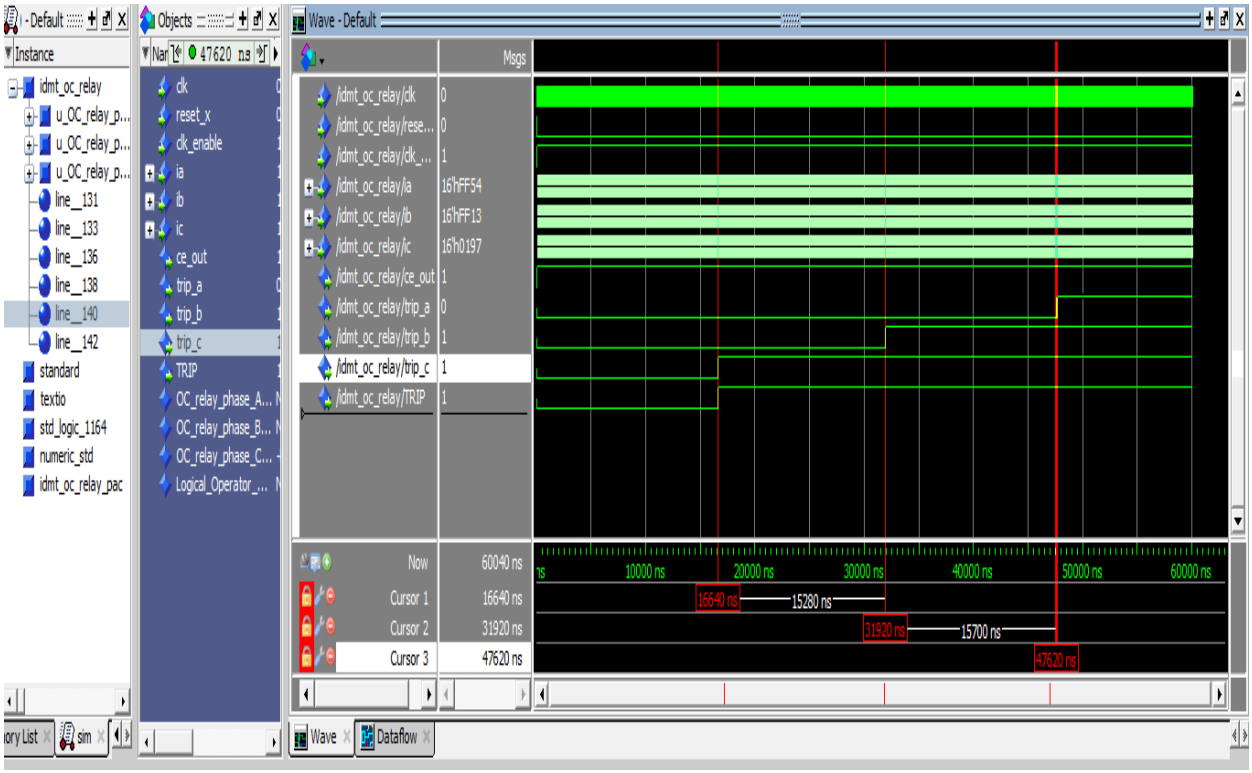

Figure 3.20 Output of the Digital Simulator.

Comparative graphs for the output trip times of Simulink and Modelsim are also generated and are show in the figures below where the first graph is the output of Modelsim, the second is Simulink output and the third is the error between the two.

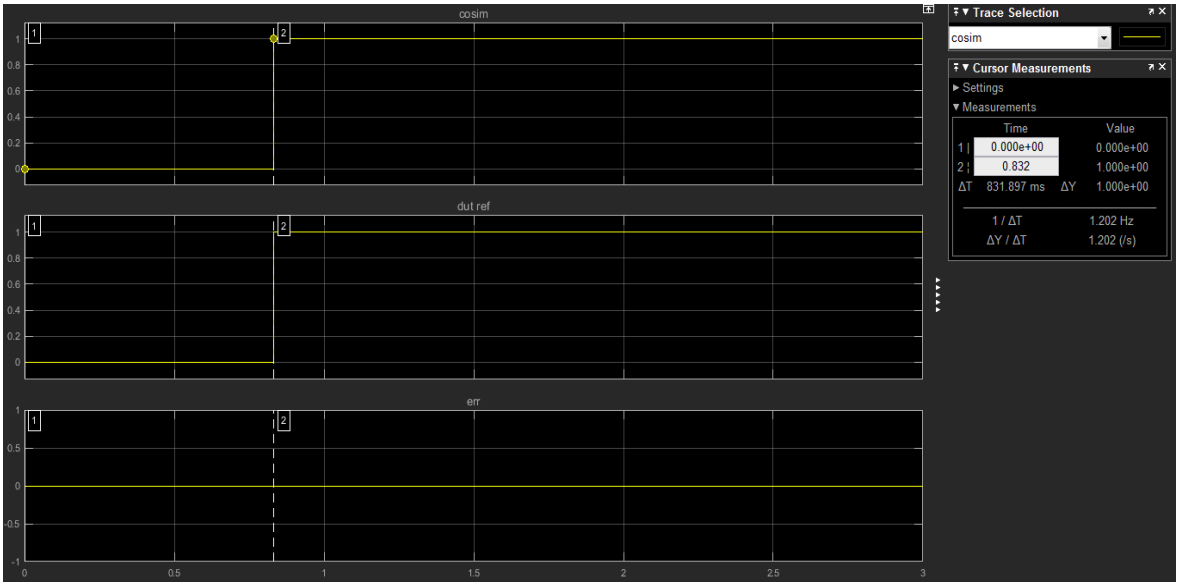

Figure 3.21 Comparative Trip Time for OC_Relay_A.

Figure 3.22 Comparative Trip Time for OC_Relay_B.

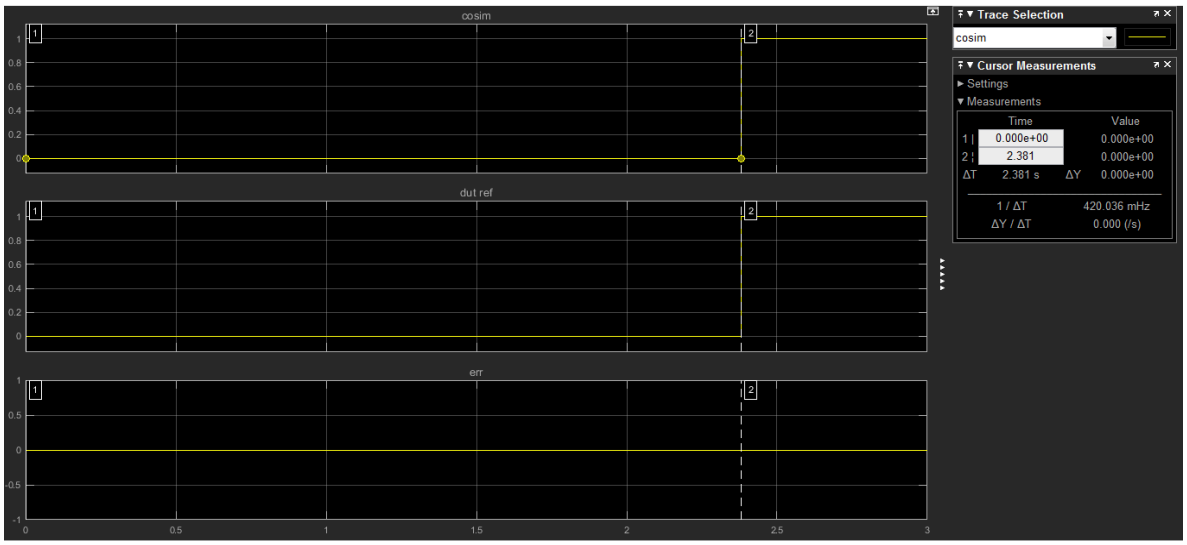

Figure 3.23 Comparative Trip Time for OC_Relay_C.

3.5 Results discussion

Microprocessor-based Overcurrent relays conform to the **IEEE Std C37.112-1996** [12] when they are implemented using (Eq 2.5), (Eq 2.6) and (Eq 2.7) meaning that the resultant trip time values corresponding to values in the range of 1.5 to 20 multiples of the pickup current are within the conformance bands shown in **Figure 3.25, Figure 3.26** and **Figure 3.27.** The upper and lower limits of the conformance bands are 1.15 and 0.85 times the standard trip times.

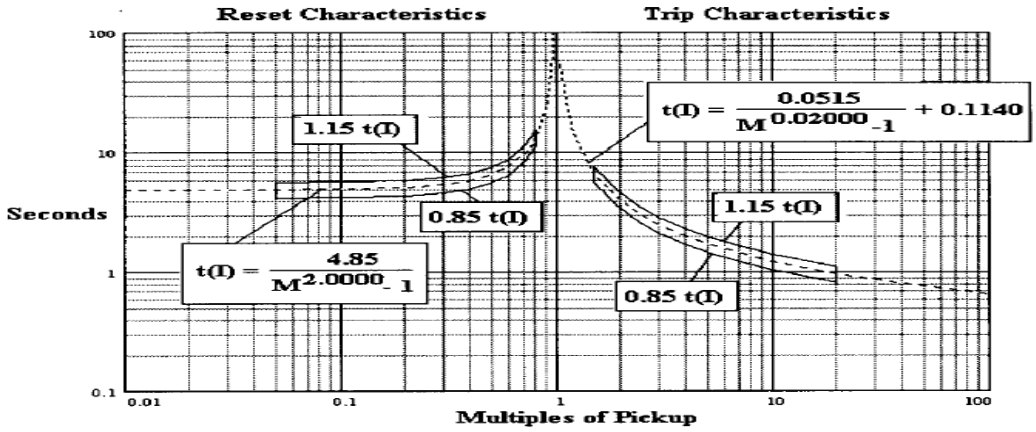

Figure 3.24 Standard Moderately Inverse Time-Current Characteristic with Standard Conformance Band Near the Middle of the Time Dial Range. [12]

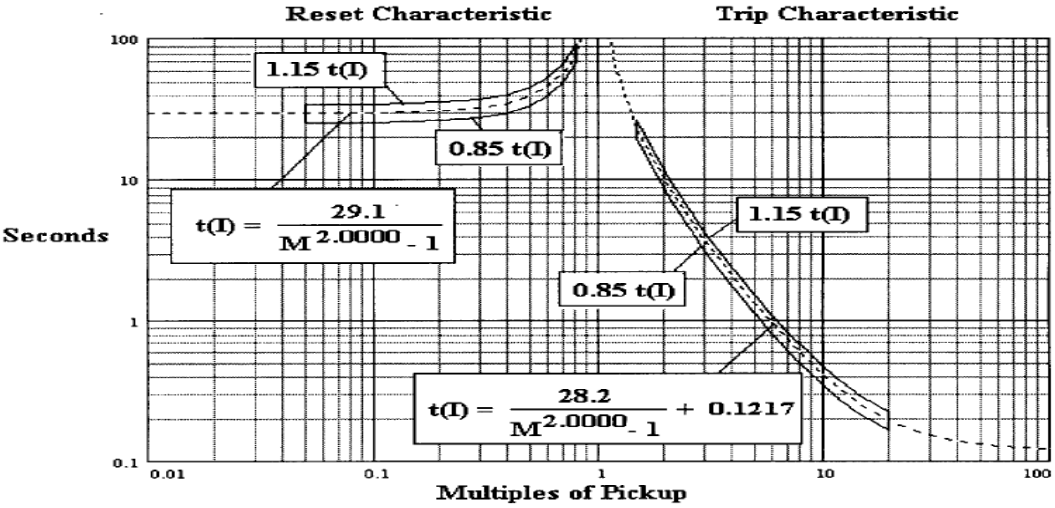

Figure 3.25 Standard Very Inverse Time-Current Characteristic with Standard Conformance Band Near the Middle of the Time Dial Range. [12]

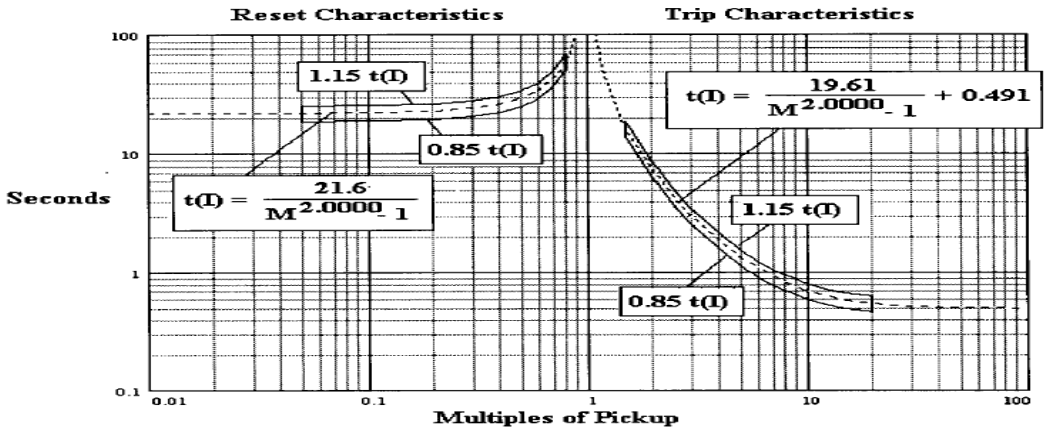

Figure 3.26 Standard Extremely Inverse Time-Current Characteristic with Standard Conformance Band Near the Middle of the Time Dial Range. [12]

- The results in **Table 3.2**, **Table 3.3** and **Table 3.4 show trip** times that are within the ranges defined by IEEE and thus the relay designed is conform according to the **IEEE Std C37.112-1996.**
- Simulink environment provided adequate tools in order to visualize the model being designed at each step of the work allowing the comparison between the expected values and the obtained one and this is one of the most important features of Model based Designs.
- Model Based Designing using Simulink for the purpose of code generation intended for hardware interface limits the resources and the blocks that can be used for modeling thus the need for developing new algorithms for some functions that do not possess HDL Optimized blocks. For example, in this thesis, rms calculation for the input samples was implemented using the basic equations for Root Mean Square because Simulink possesses a discrete rms block that is not HDL optimized.
- For an optimal code generation, the model needed to be divided into smaller blocks and Matlab embedded functions to simpler smaller functions to avoid errors in the generation of the code such as word-length related problems.
- The conversion to HDL passes through data type conversion from floating point to fixed point which is the data type that is accepted by most hardware platforms FPGA included and that can give outputs that are different from the original desired model. For an optimal data type conversation, fixed point advisor in Simulink runs necessary checks on the model and the configuration parameters to ensure the conversion is done correctly. Basic signals should be enabled for data logging where the advisor stocks their values from the original simulation and that is to propose data ranges for each signal and component in the design. Ranges based on which data types are proposed for the model elements.

3.6 Conclusion

In this chapter, the proposed algorithm was simulated using Simulink and Modelsim and the obtained results were in accordance with the standard results of IEEE.

CHAPTER IV

Implementation and Testing

Implementation and testing

4.1 Introduction

This chapter is a detailed walkthrough of the implementation process, explaining the methods used depicted with schematics.

4.2 Making a power source

In order to feed the FPGA used and the ADC with the needed power, a power source has been made using a center taped 220v to 24v transformer, two full wave rectifiers with their filters and three voltage regulators resulting in three outputs, +12v, -12V for the ADC and the op-amps contained in the signal conditioning circuit and +5V for the FPGA which has been enhanced by a high current transistor to increase the current delivery of the 5v voltage regulator and meet the FPGA required power.

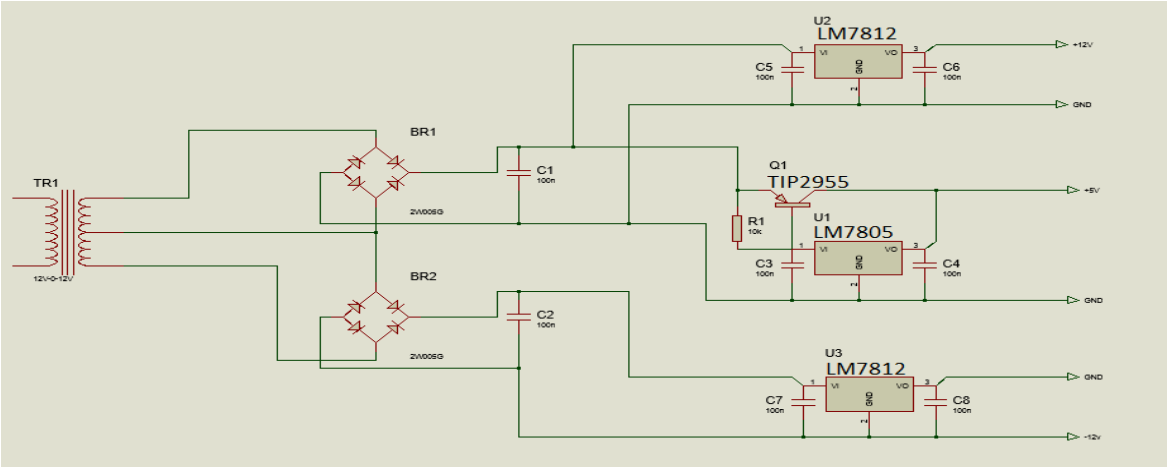

Figure 4.1 The Power Supply's Circuit Diagram.

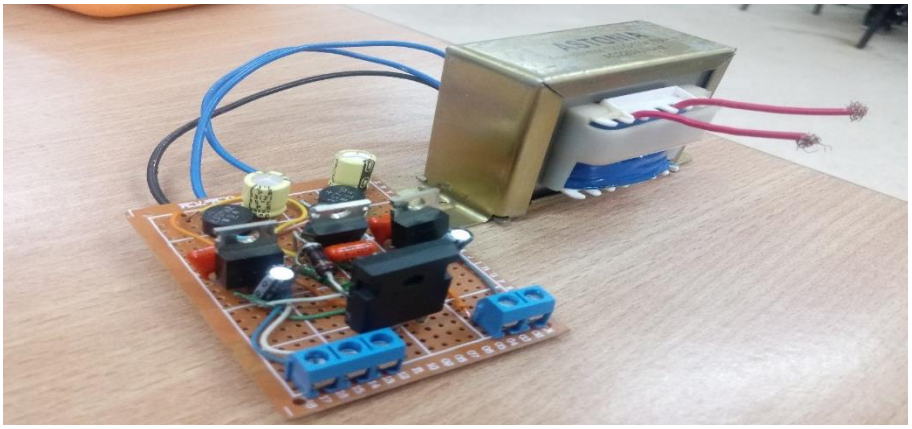

Figure 4.2 The Implemented Power Supply.

4.3 Making the signal conditioning circuit

To read the values of the CTs by the ADC, the output current of the latter was made to pass through 0.1Ω resistors resulting in 3 voltages with 0.141V peak when the secondary current of the CT is 1A, this voltage is then amplified using three operational amplifiers (one for each CT) with a gain of 10, resulting in a voltage of about 1.41V peak in the normal case, this voltage can safely be inputted to the ADC that can take values between +10V and -10V leaving enough margin for faults and short circuit currents.

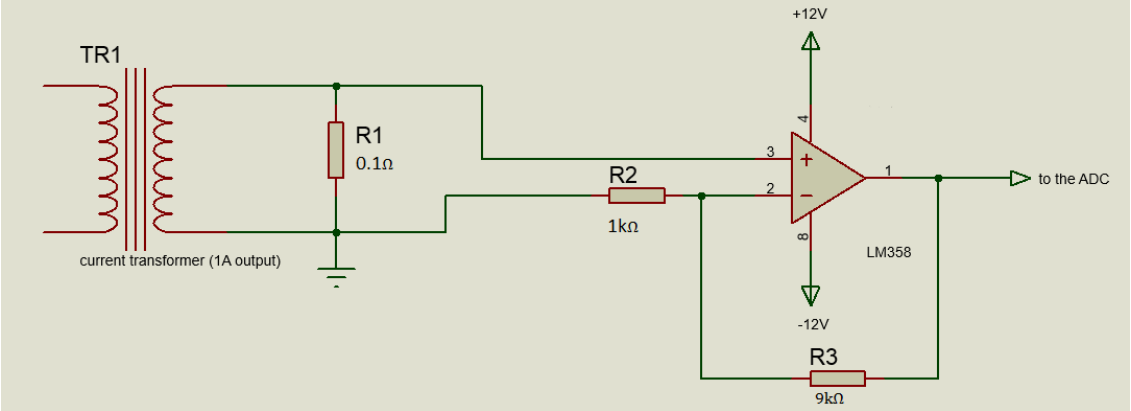

Figure 4.3 The Signal Conditioning Circuit Schematic (for one phase).

Figure 4.4 The Implemented Signal Conditioning Circuit.

4.4 The communication part

In this part, the way power line communication was implemented is methodologically explained, from emitting the information to receiving it.

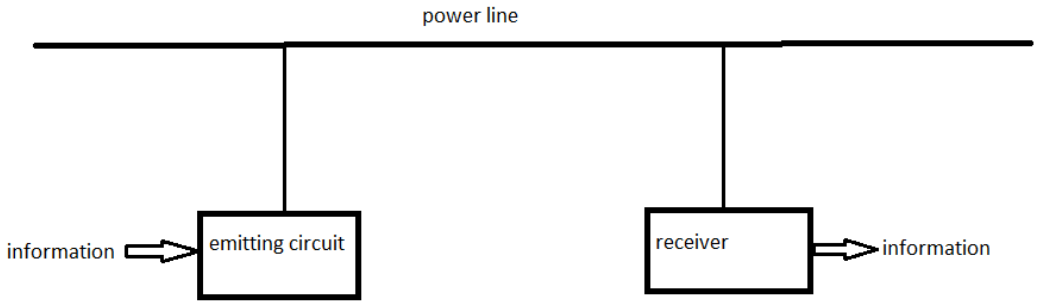

Figure 4.5 Functional Diagram of a Unidirectional Power Line Communication Installation.

4.4.1 The modulation

To modulate the signal, a carrier signal had to be generated, this was done using the ne555 timer in its astable mode using the following schematic:

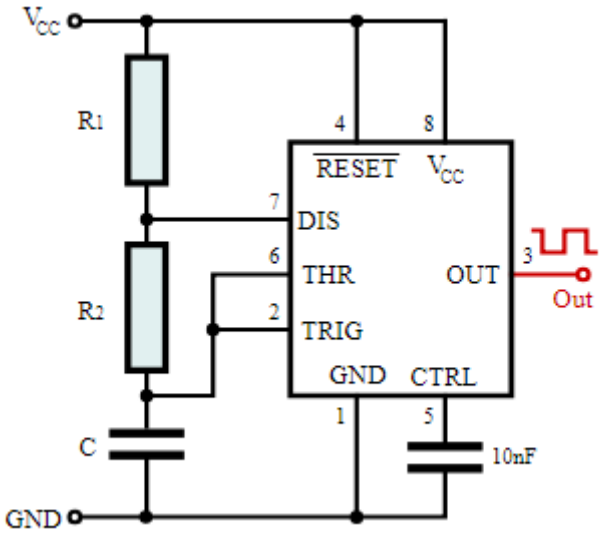

Figure 4.6 The Carrier Generating Circuit. [14]

The generated frequency is then calculated using the following equation:

$$f = 1/(T_{High} + T_{Low})\ \mathrm{T} \tag{4.1}$$

And the duty cycle (T high and T low individually), even though irrelevant to our application, can be calculated with this equation:

$$T_{High} = ln2 * C * (R1 + R2) \tag{4.2}$$

$$T_{Low} = ln2 * C * R2 \tag{4.3}$$

With the proper configuration, a carrier of 214 kHz having a duty cycle around 60% was generated in practice, the choice of this frequency will later on be explained.

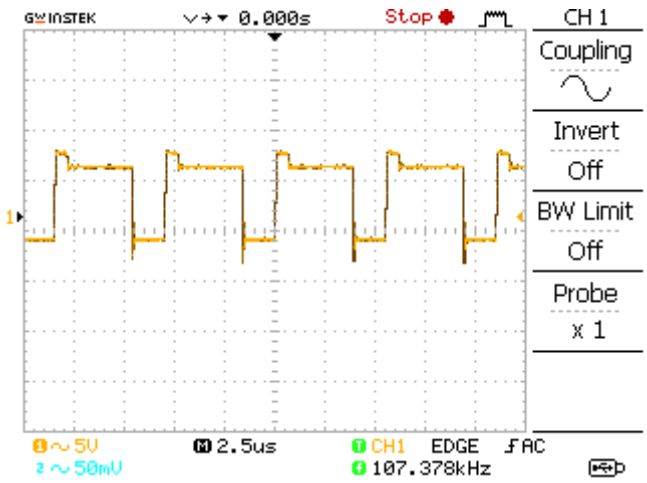

Figure 4.7 The Obtained Carrier Signal.

The image above shows the carrier signal.

The next step is now to multiply this carrier signal by the information that is to be sent, this is done with a small modification in the previously shown circuit, in the astable mode, pin 4 of the 555 timer plays the role of the enable pin, so, if V_{cc} is inserted the timer outputs the 214 KHz carrier, and if instead pin 4 is put to ground (or 0V is inserted) the timer outputs 0V. The trick is to connect pin 4 to the source of information to get a modulated waveform.

In this implementation another 555 timer was used to generate a 440 Hz square wave to simulate an information, the resulting modulated signal was the one shown in the next figure:

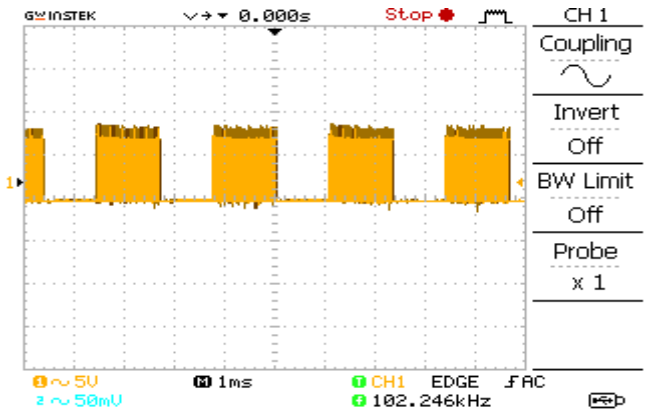

Figure 4.8 The Modulated Signal.

Zooming in the yellow section (that is in fact a high frequency square wave) the following can be seen:

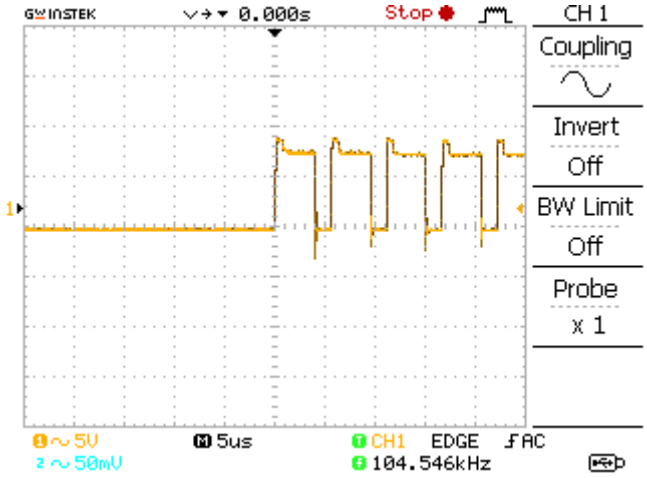

Figure 4.9 Closer Look on the Modulated Signal.

4.4.2 Amplifying the signal

The injected signal needs to have high current in order to arrive to the receiving end, that's why a high current high frequency IGBT (G4PH40KD) was used, with its gate triggered by the output of the modulator.

4.4.3 Injecting the signal to the grid

Now that the signal is modulated, it needs to be sent it through the power line in the safest way possible and the most efficient one losses wise.

This implementation was tested in the lab, in the domestic 220V plugs, so the protection was designed accordingly. First, current returns from the power line itself had

to be avoided, from here raised the need to modulate the signal, so that the spectrum of the information can be moved away from the spectrum of the power signal that is sitting on 50 Hz or 60 Hz with harmonics of either negligible frequency when compared to 214 kHz (where the modulated signal is located) or negligible amplitude when talking about high frequency harmonics.

The goal behind this separation is to use a high pass filter of whatever shape in its transient part, since the two signals are too far apart, allowing by such a filter for only the information carrying signal to pass through. A same filter is used at the receiving end to fulfill the same purpose, only this time the information comes out of the power line instead of going into it.

4.4.3.1 Designing the filter

The filter consists of a simple capacitor connected in series between the phase of the power line and the output signal of the modulator, a 0.1 micro farad capacitor was chosen so as to get the following impedances:

- At 50Hz:

$$Z = \frac{1}{2*\pi*50*0.1*10^{-6}} = 31.8\ \text{M}\Omega \quad (4.4)$$

- At 200 kHz:

$$Z = \frac{1}{2*\pi*200*10^{3}*0.1*10^{-6}} = 7957\ \Omega \quad (4.5)$$

Now to take safety one step further, isolation transformers were used, one at both the sending and the receiving ends to separate the power circuit from the electronic one and avoid any electrical shocks, the kmb5191-101 transformer was used since it can pass up to 500 kHz frequencies and for its compact size.

The final emitting circuit is shown below:

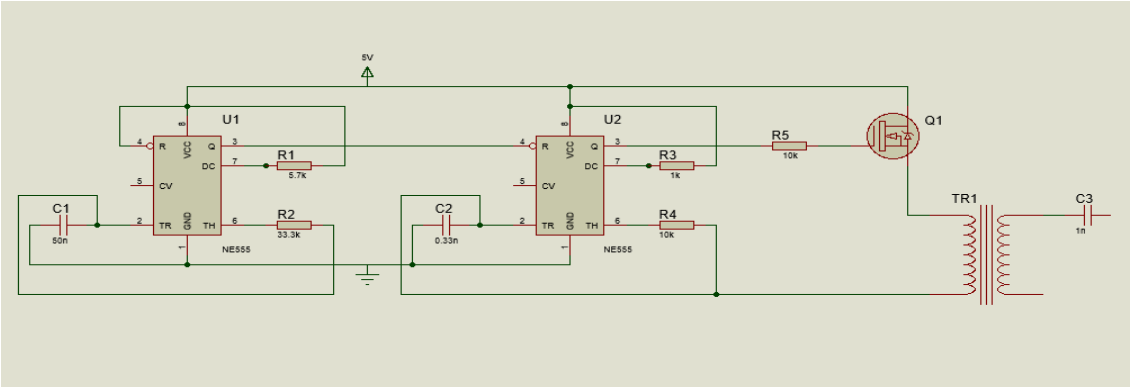

Figure 4.10 The Emitting Circuit Schematic.

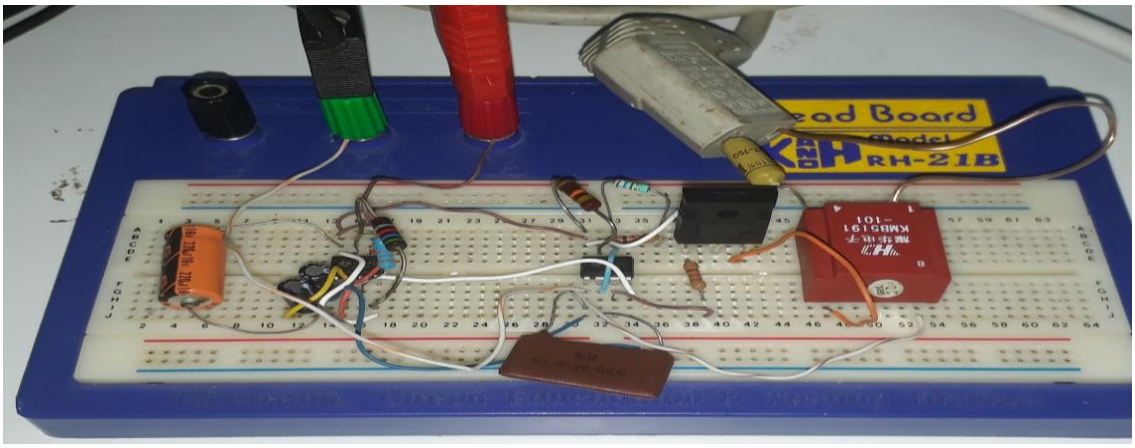

Figure 4.11 The Implemented Emitting Circuit.

4.4.4 Demodulating the signal

For this the lm567 tone decoder was used, which is a general purpose tone decoder designed to provide a saturated transistor switch to ground when an input signal is present within the passband.

It is implemented following the circuit shown below:

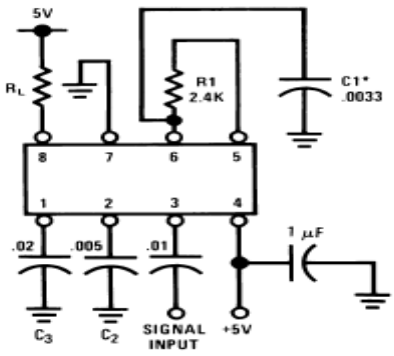

Figure 4.12 The Tone Decoder. [15]

It is tuned following the next equations:

$$f_0 = \frac{1}{1.1*R1*C1} \tag{4.6}$$

To calculate the frequency it has to detect (214 kHz)

And to calculate the bandwidth (4.7) is used

$$BW = 1070\sqrt{\frac{Vi}{(f_0*C2)}} \qquad \text{in \% of } f_0 \tag{4.7}$$

where:

Vi = Input voltage (volts rms), Vi ≤ 200mV

C2 = Capacitance at Pin 2(µF)

The received signal (given as input to the demodulator) was the following:

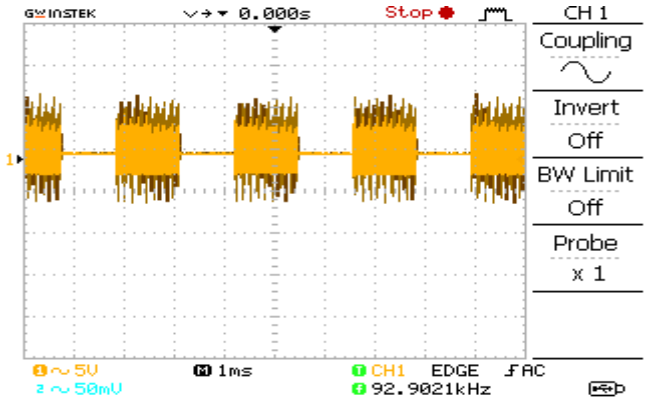

Figure 4.13 The Received Modulated Signal.

And this is how it looks like when zoomed on the time scale:

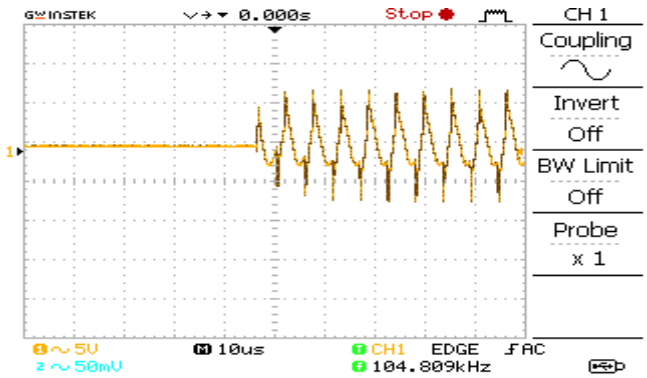

Figure 4.14 A Closer Look on the Modulated Signal.

It can be seen from the previous screenshot (that was taken from the oscilloscope) that the carrier is no longer a square wave, but it doesn't matter since it kept the same frequency, so it will easily be detected by our tone decoder.

And this can be seen in the output of the demodulator that is shown in the following figure

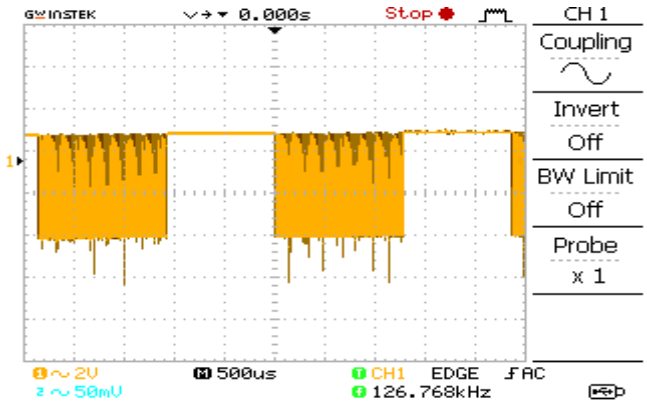

Figure 4.15 The Demodulated Signal.

Now that the signal is demodulated, it can be seen that the input signal is not fully reconstructed and a high frequency square wave is generated where we should have 0V, this was remedied by the use of a simple unidirectional low pass filter, and this led to the screenshot below, the one on the left, and the one on the right being the original signal that was sent

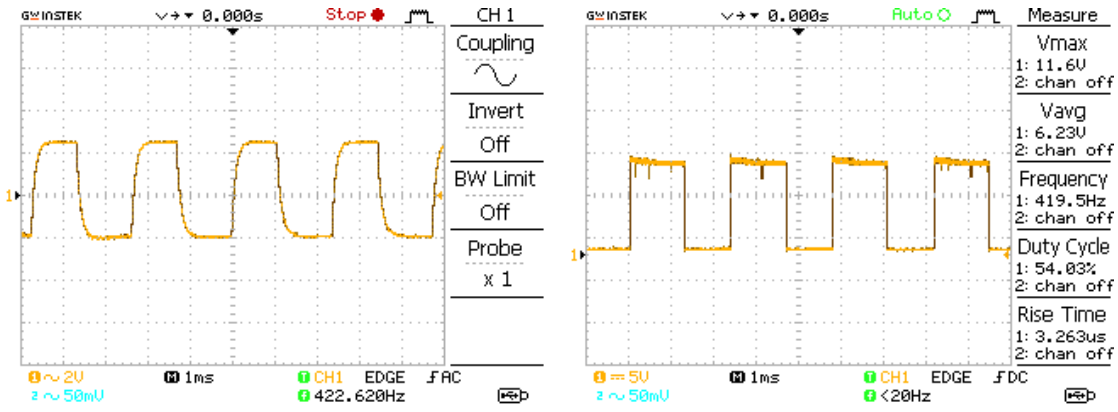

Figure 4.16 The Demodulated Signal after the Filter (left) vs the Original Signal (right).

This signal is very close to the original one that was sent, but to make a perfect match out of it and eliminate the effect of charge and discharge of the filter's capacitor a comparator having this signal as an input, and a reference that can be tuned by a

potentiometer was used until the exact same signal that was sent is recovered. The final result was the following:

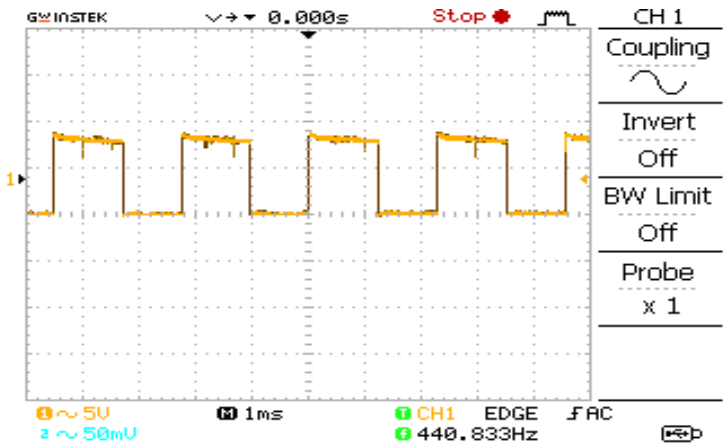

Figure 4.17 The Final Received Signal after the Comparator.

In the screenshots, frequency readings are not accurate and show that the signals do not match, while in fact they do match and this can be verified using the scale.

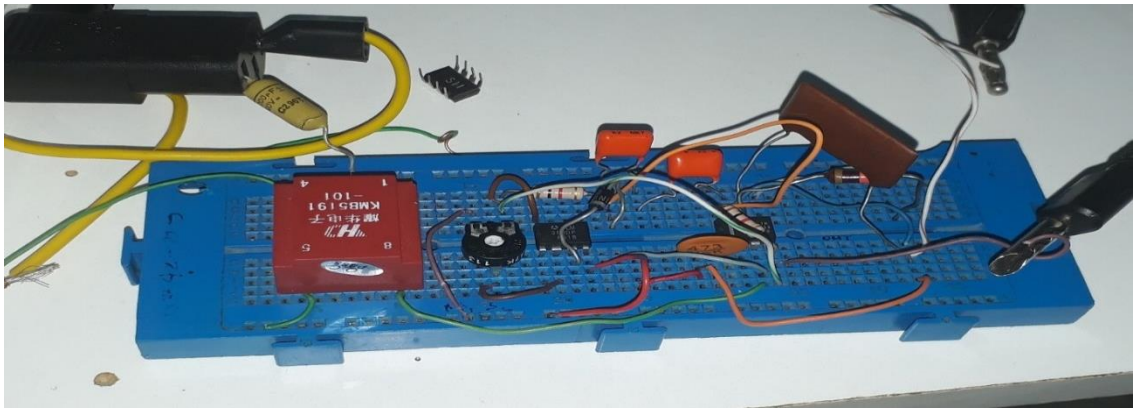

Figure 4.18 The Receiving Circuit Implementation.

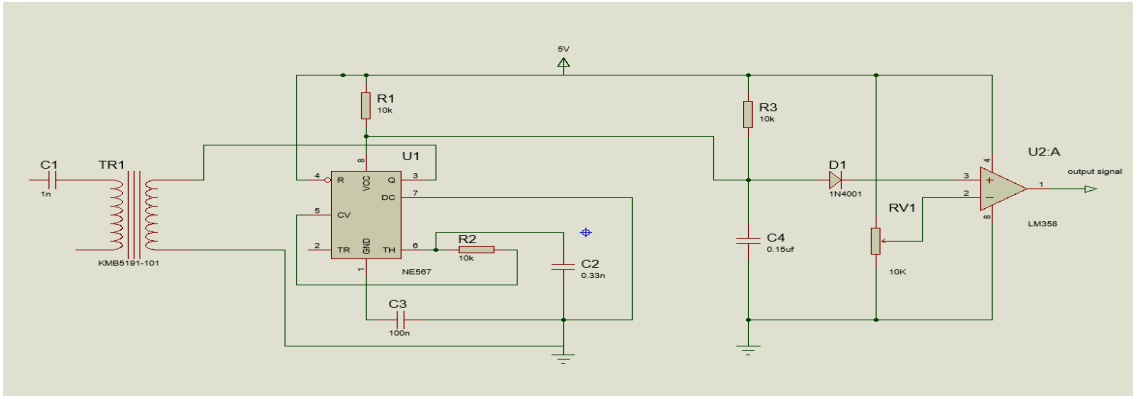

Figure 4.19 The Receiver Circuit Schematic

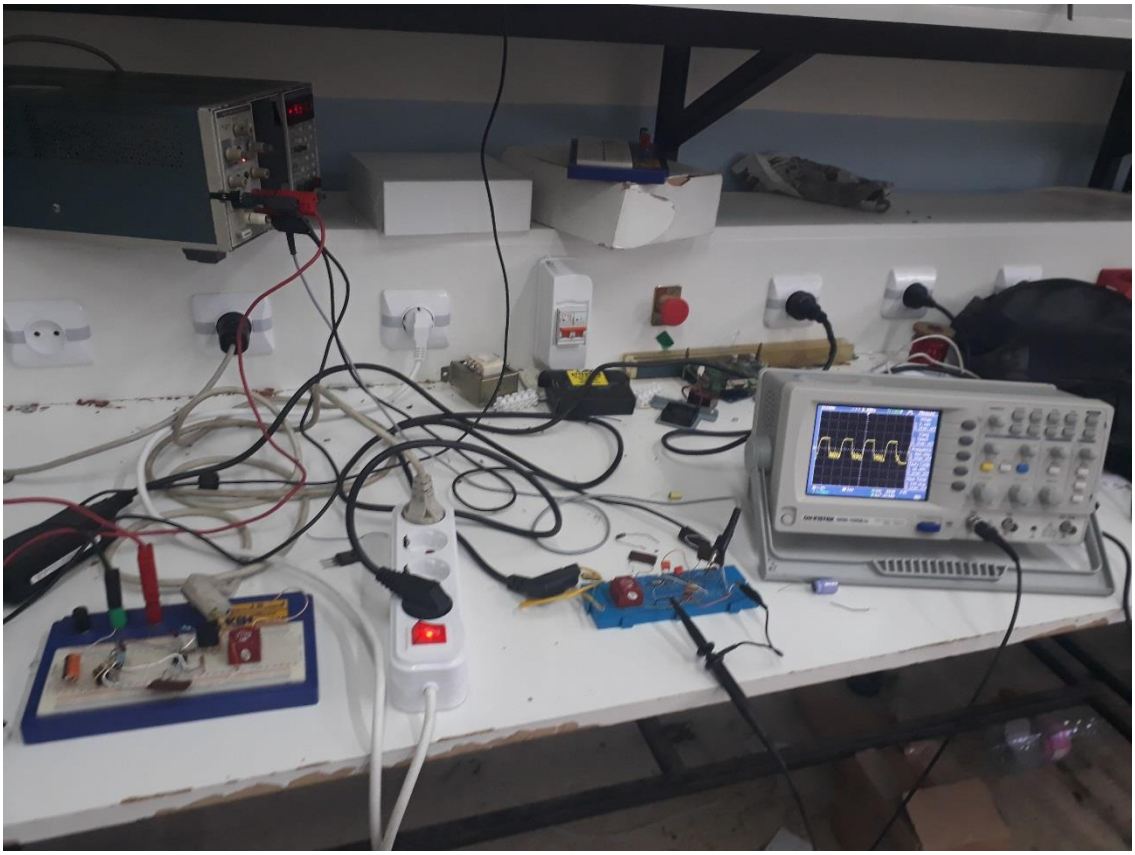

Figure 4.20 The Power Line Communication at Work.

4.5 Conclusion

This chapter has explained the hardware part implementation of the designed relay .All the implemented circuits throughout this chapter performed as expected and the results obtained were satisfying.

Conclusion

A model based design of digital overcurrent relay has been done in the present work. A modified technique for modelling inverse-time overcurrent relays has been used. The design is developed in Simulink environment. VHDL code of the model is generated using the Simulink HDL coder after making the model compatible for code generation. The functionality of the generated code is checked on ModelSim simulator. Test results indicate that the overcurrent relay accurately emulates the desired time-current characteristics both in Simulink and Modelsim digital verifier.

A hardware platform for power line communications has been separately implemented and tested, for it to be integrated to the relay and make it communicative, an essential function in any modern digital relay. When testing our PLC we were able to recover the exact same test signal that was sent.

Further Work

This work can be extended to other protection functions, namely earth fault, distance and differential protection algorithms. Directional element can also be added to make the designed relay more selective. Novel algorithms for current only directional detection have been the subject of many researches [16] [17] and can be used to improve the design proposed. More HDL optimized libraries can be added to CAD tools in order to facilitate Model Based designing.

This work can also further be extended to enter the PMU world (phasor measurement units) by equipping it with a GPS antennas, allowing us to take synchronized phasor measurements.

More improvements can also be done on the communication scheme, working with higher frequencies and appropriate protocols can increase the data transfer rate, and revisiting the hardware we can work on increasing the range of transmission.

References:

[1] G. Burati, "Overcurrent protection", in *Razdelilna in industrijska omrežja*, 2017.

[2] P. M. Anderson, “power system protection”, McGraw Hill, p.p. 413-414, 1998.

[3] Computer representation of overcurrent relay characteristics, IEEE Power Engineering Review, Vol.9, July 1989, pp. 50–51.

[4] F. Wang, and M. H. J. Bollen, Quantification of transient current signals in the viewpoint of overcurrent relay, IEEE Power Engineering Society General Meeting, Vol.4, 2003. [5] Stanley H . H or owitz and A r un G . Phadke,’ *Power System Relaying, Third Edition.2008*

[5] S. H. Horowitz, A. G. Phadke, Power system relaying, 3rd ed. , Research

Studies Press Ltd. and John Wiley & Sons, Ltd. , May 2008.

[6] ALSTOM, Network Protection & Automation Guide, 2012.

[7] Sandro Gianny Aquiles Perez, “Modeling Relays for Power System Protection Studies”, Ph.D

Research, University of Saskatchewan, Saskatchewan, Canada, July 2006.

[8] https://circuitglobe.com/wp-content/uploads/2016/06/basic-connection-diagram-of-connecting-relay-compressor.jpg

[9] http://homepages.westminster.org.uk

[10] T. Sidhu, M. Sachdev and H. Wood, "Design of a microprocessor-based overcurrent relay", [Proceedings] WESCANEX '91.

[11] http://myelectrical.com/Portals/0/DNNArticle/Files/718/WLWimages/Windows-Live-Writer-Electromechanical-Relays_BFC6-IEC60255Curves_4.jpg.

[12] IEEE Standard Inverse-Time Characteristic Equations for Overcurrent Relays.

[13] https://www.matlabhelp.com/model-based-design-assignment-help-33938.

[14] https://en.wikipedia.org/wiki/555_timer_IC.

[15] LM567/LM567C Tone Decoder datasheet.

[16] D. D. Tung and N. M. Khoa, “A Novel Algorithm of Directional Overcurrent Protection Relay Based on Adaptive Linear Neural Network,” *International Journal of Electronics and Electrical Engineering*, pp. 494–499, 2016.

[17] M. Nojavan, H. Seyedi, and M. Mehdinejad, “A novel scheme for current-only directional overcurrent relay,” *International Journal of Electrical Power & Energy Systems*, vol. 82, pp. 252–263, 2016.