A Project Report

On

Implementation of Multi-Channel Time-to-Digital Converter using FPGA

Submitted in partial fulfilment of the requirement of

University of Mumbai

For the Degree of

Bachelor of Engineering

in

COMPUTER ENGINEERING

Submitted by

Harshul Raina
Priyanka Sankannawar
Vishadh Sawant

Supervised by

Dr. Jyoti More

&

Mr. Irfan Mirza

Department of Computer Engineering

Fr. Conceicao Rodrigues Institute of Technology
Sector 9A, Vashi, Navi Mumbai - 400703

UNIVERSITY OF MUMBAI

2021-2022

APPROVAL SHEET

This is to certify that the project entitled

"Implementation of Multi-Channel Time-to-Digital Converter using FPGA"

Submitted by

Harshul Raina	**101851**
Priyanka Sankannawar	**101854**
Vishadh Sawant	**101855**

Supervisors : 1. Dr. Jyoti More

2.Mr. Irfan Mirza

Project Coordinator : Ms. Rakhi Kalantri

Examiners : 1. _____

2. _____

Head of Department : Dr. Lata Ragha

Date :

Place :

Declaration

We declare that this written submission for B.E. Declaration entitled "**Implementation of Multi-Channel Time-to-Digital Converter using FPGA**" represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declared that we have adhere to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have thus not been properly cited or from whom paper permission have not been taken when needed.

Project Group Members:

1. Harshul Raina, 101851

---

2. Priyanka Sankannawar, 101854

---

3. Vishadh Sawant, 101855

Abstract

This project deals with the design and implementation of a Multi-channel Time-to-Digital Converter (TDC) to measure time intervals with a picosecond level resolution. The hybrid counter designed for measuring the time interval uses a combination of a coarse counter and a fine counter technique that gives measurements with an overall resolution of 833.33 picoseconds. The coarse counter provides a longer measuring range whereas the fine counter provides the precision required in the concerned conditions. Thus, this design is efficient and reliable.
A Field Programmable Gate Array is used to implement the above design (FPGA). Using a Hardware Description Language, FPGAs can be designed to meet the needs of the developer (HDL). This is beneficial to the project since it provides: (1) high parallel processing capabilities (2) high reconfiguration flexibility (3) less development time.
The embedded system developed for this research is planned to be integrated with the Tata Institute of Fundamental Research's GRAPES-3 experiment. As a result, it's built to handle enormous amounts of data efficiently during communication and processing. Finally, the suggested FPGA-based TDC system is combined with a UART module to allow for smooth data transmission to and from the end user.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background

The earth's atmosphere is blasted on a daily basis by cosmic rays, which are high-energy particles that travel from the sun or further out in the universe. When a primary energetic particle enters the Earth's atmosphere, it collides with nuclei and molecules, producing secondary particles that share the energy of the original primary particle. The secondary particles continue to propagate, decay, and form new particles. This produces large air showers, which are a spray of particles. Leptons (muons, electrons, neutrinos) and gamma rays make up the majority of these showers. [1] As a result, when the ray reaches ground level, the air spray created by a single ray covers a broad region. The detection and observation of these air showers is carried out using huge muon detectors installed at ground level.

The study of cosmic rays in different regions of the Galaxy can be aided by the detection of diffuse Galactic gamma ray flux. The abundance and spectrum shape of relativistic particles in the Milky Way are encoded in the energy and angular distributions of photons. [2]

The GRAPES-3 experiment in Ooty, India, began as a cooperation between Mumbai's Tata Institute of Fundamental Research and Osaka City University in Japan. It is presently a collaboration between numerous Indian and Japanese institutes. An array of air shower detectors and a large area muon detector are used in the experiment to detect and analyse cosmic rays. Currently, approximately 400 scintillators covering a total area of 25,000 m^2 are being used to detect charged particles in large air showers caused by the interaction of high-energy cosmic rays with the atmosphere.[3].The GRAPES-3 experiment will investigate the following topics:

- In the galaxy and beyond, the origin, acceleration, and spread of >1014 eV cosmic rays.
- Existence of "Knee" in the energy spectrum of cosmic rays.
- Production and/or acceleration of highest energy (≈1020 eV) cosmic rays in the universe.
- Astronomy of multi-TeV -rays from neutron stars and other compact objects.
- Sun the closest astrophysical object, accelerator of energetic particles and its effects on the Earth.

The angle of incidence of cosmic rays is critical for determining the

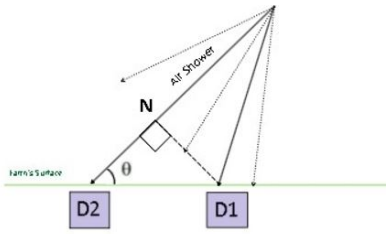

Figure 1.1: Detection of Air Showers

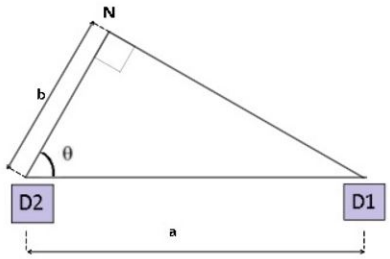

Figure 1.2: Calculating the Angle of Incidence of Cosmic Ray Air Shower

distance to the rays' source. The rays from a distant source reach the Earth's surface with a wavefront that has a very high radius curvature, as shown in Figure 1.1. The angle of incidence of the beams may be computed using trigonometry because this wavefront is a straight line. The graphic in Figure 1.2 explains how the angle of incidence θ can be determined:

$$\theta = \cos^{-1}\frac{b}{a} = \cos^{-1}\frac{v.\Delta t}{a} \tag{1.1}$$

$$v = \text{velocity of the ray (known)}$$
$$\Delta t = \text{time measured by TDC}$$

As a result, it's critical to carefully and precisely estimate the time intervals between the detection of rays at adjacent light sensors.

The goal of this project is to develop and build a multichannel time-to-digital converter (TDC) that can measure and transmit these time intervals. The coarse counter and fine counter are used in the hybrid counter for measuring the time interval. The coarse counter has a wider measuring range, whereas the fine counter has the precision required in the concerning circumstances. As a result, this combination architecture is both efficient and appropriate for achieving a TDC with picosecond resolution.

1.2 Motivation

In a variety of high-energy physical experiments, such as examining subatomic fine structure in fixed target studies and collision experiments, precise time interval measurements are frequently required. It is employed in this case as part of a wider experiment to observe and investigate diffuse galactic ray flux. The TDC is a prerequisite for understanding more about cosmic rays in various parts of the cosmos. The photons' angular distributions are utilised to learn about the density and spectral shape of relativistic particles throughout the Milky Way. As a result, this initiative

plays a minor but critical role in gaining a better understanding of the universe from the tiny spot named Earth. This influenced our decision to work on this project.

1.3 Aim and Objective

This project will use a Field Programmable Gate Array (FPGA) to develop and implement a multi-channel time-to-digital converter (TDC) (FPGA). With an accuracy of picoseconds, the developed hybrid counter will be able to monitor time intervals between two signals (start and stop). The UART protocol is used to efficiently communicate data between the FPGA and the end user.

1.4 Report Outline

There are six chapters in this report. The project's origins, motivation, and aims are discussed in detail beginning with the introduction in Chapter 1. The research on the topic is discussed in Chapter 2. It describes the project's technique as well as other precision time measurement and time-to-digital converters techniques.This chapter also contains a literature survey, which comprises a review and comparison of 6 previous research. The proposed system of the project is explained in Chapter 3 with details about the problem statement, scope, and components of the proposed system. Chapter 4 gives further details on the design of the system and its requirements. The results of the modules of the project implemented so far are presented in Chapter 5. Finally, Chapter 6 concludes the report and presents the future scope.

Chapter 2

Study Of the System

2.1 Field Programmable Gate Arrays (FPGA)

The semiconductor industry has been evolving at a rapid pace. Because of their cheaper manufacturing costs and quicker development time, FPGA devices have become increasingly popular for rapid system prototyping, logic emulation, and re-configurable computing.[4].

A field-programmable gate array (FPGA) is a type of integrated circuit that allows a client or designer to customise it after it has been manufactured. A hardware description language (HDL), similar to that used for application-specific integrated circuits, is used to specify the FPGA setup (ASIC). FPGAs have an array of programmable logic blocks and a hierarchy of "reconfigurable interconnects" that allow blocks to be linked together in various configurations, such as multiple logic gates. Figure 2.1 depicts the logic blocks that can be used to execute sophisticated combinational operations or simple logic gates such as AND and XOR.

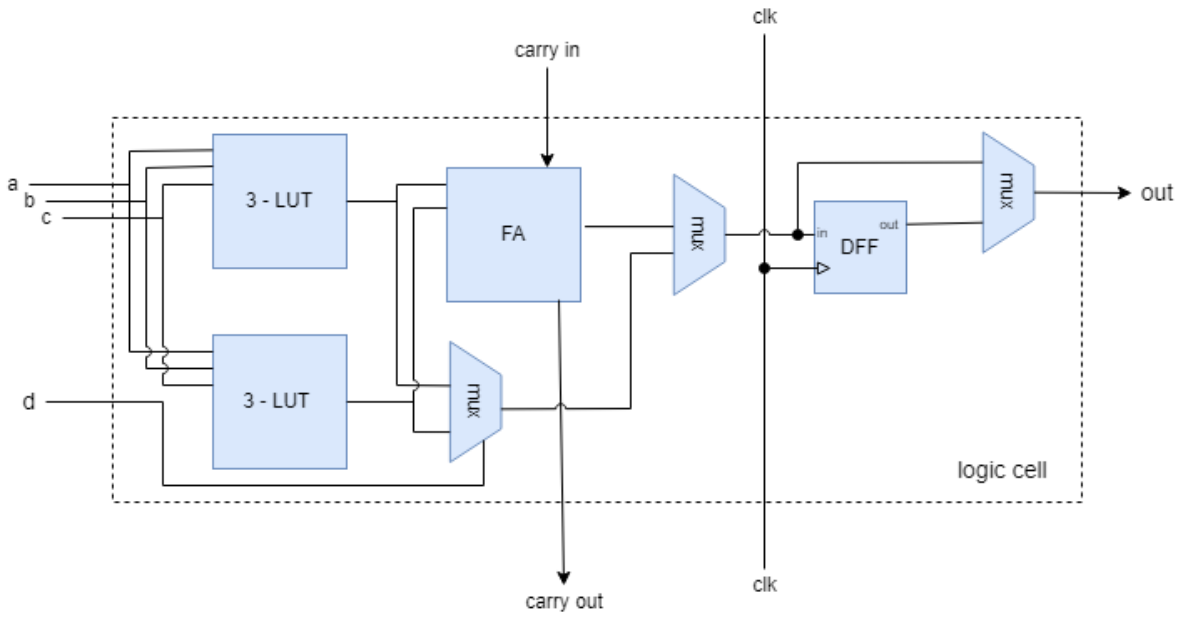

Figure 2.1: Simplified Illustration of a Logic Cell (LUT-Lookup table, FA-Full adder, DFF-D-type flip-flop)

2.1.1 Nexys A7 FPGA

The Nexys A7 board is a fully functional, ready-to-use digital circuit development platform based on Xilinx's latest Artix-7 Field Programmable Gate Array (FPGA). The Nexys A7 was the best choice for this project because of its large, high-capacity FPGA, substantial external memories, and assortment of USB, Ethernet, and other ports [5]. The suggested TDC, which takes advantage of the FPGA platform, has advantages in terms of ease of implementation, low cost, and quick development time.

The Nexys A7 FPGA has the following useful features [5] :

- 15,850 Programmable logic slices, each with four 6-input LUTs and 8 flip-flops (*8,150 slices)
- 4,860 Kbits of fast block RAM (*2,700 Kbits)
- Internal clock speeds exceeding 450 MHz
- Six clock management tiles, each with phase-locked loop (PLL)
- Memory of 128MiB DDR2, Serial Flash with microSD card slot.
- Powered from USB or any 4.5V-5.5V external power source.
- Two RGB LEDs and two 4-digit 7-segment displays.

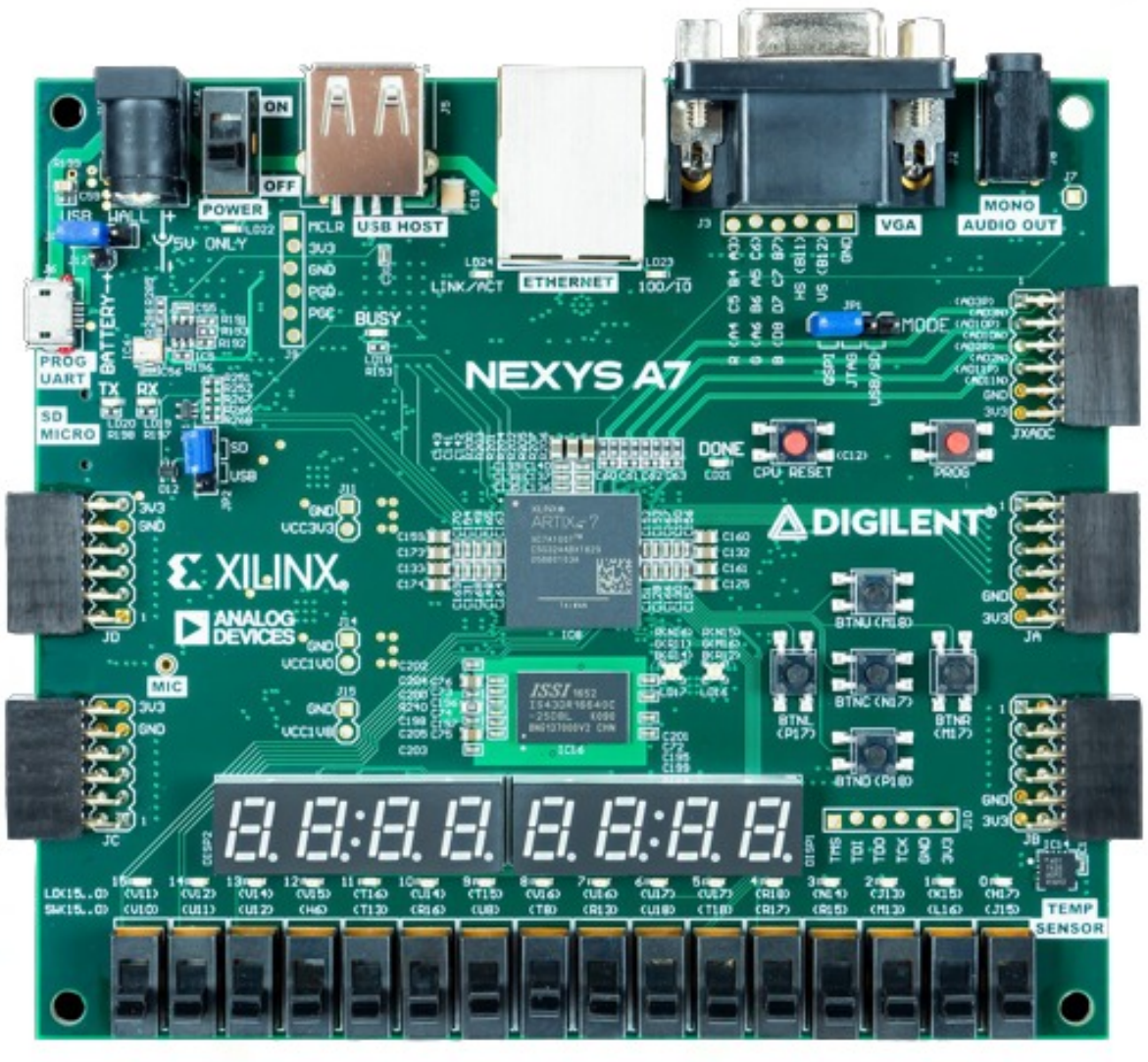

Figure 2.2: The Nexys A7 FPGA by Xilinx)

2.1.2 Mixed-Mode Clock Manager (MMCM) Module

Xilinx offers the Clocking Wizard IP core to help users generate the different clocks required for a specific design [6]. This wizard will properly instantiate the needed MMCMs and PLLs based on the desired frequencies and phase relationships specified by the user.

MMCM generates multiple clocks with defined phase and frequency relationships to a given input clock. It accepts up to two input clocks and up to seven output clocks per clock network. The output counters provide the synthesized clocks using a combination of DIVIDE, DUTY CYCLE, and PHASE. It automatically chooses the correct clocking primitive for a selected device and configures the clocking primitive based on user-selected clocking features[7]. The Figure 2.4 shows an example of how the output clocks may be configured.

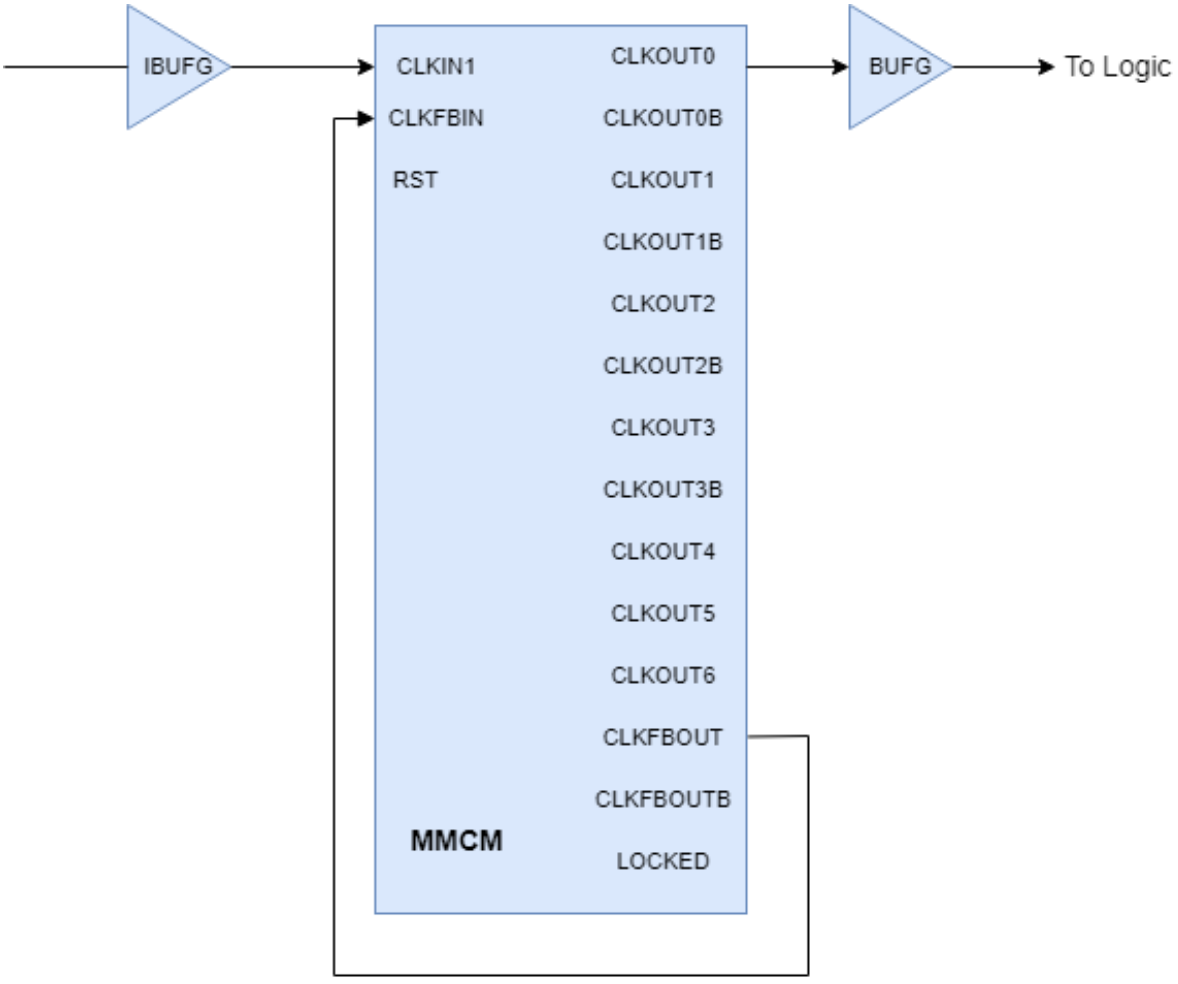

Figure 2.3: MMCM module with Internal Feedback

2.2 Vivado Design Suite

Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of HDL designs with additional features for system on a chip development and high-level synthesis. It supplies developers with the tools and methodology needed to leverage C-based design and optimized reuse, IP sub-system reuse, integration automation and accelerated design clo- sure. It also enables designers to work at a high level of abstraction while facilitating design reuse.

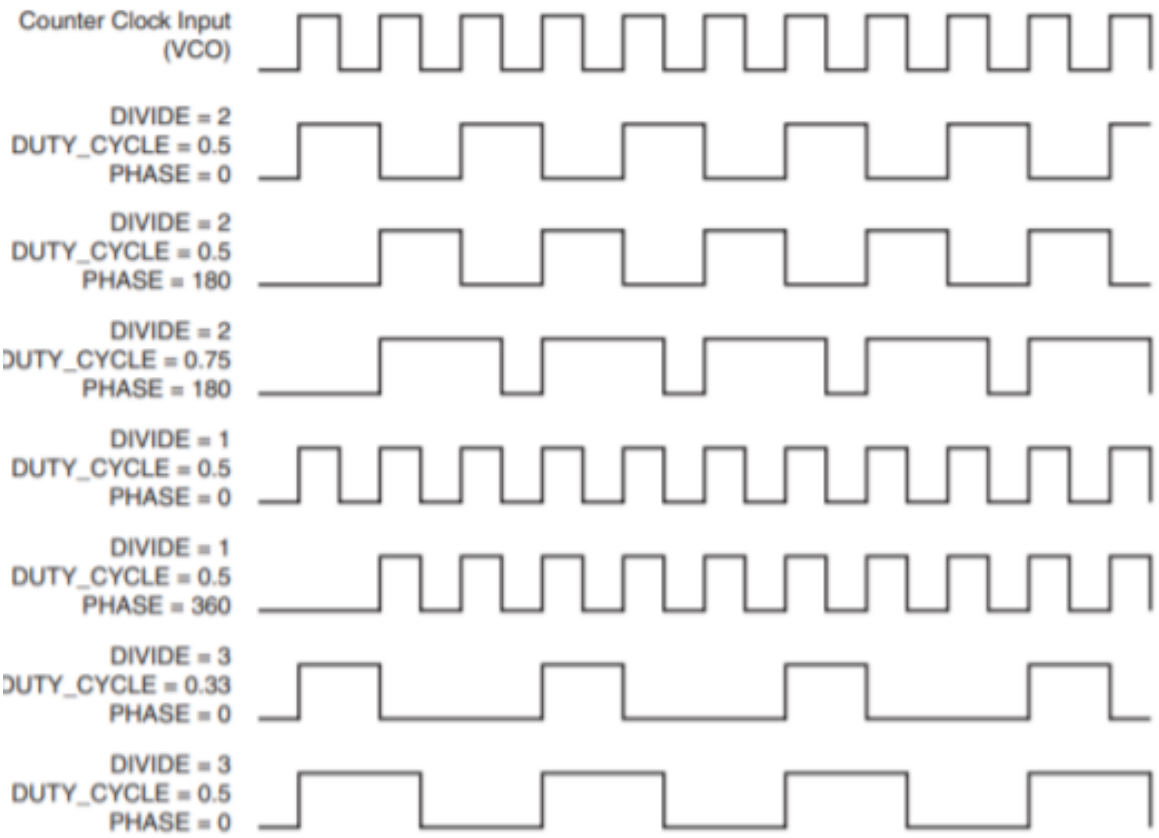

Figure 2.4: Output Counter Clock Synthesis Examples

The Vivado Design Suite provides an array of design entry, timing analysis, hardware debug, and simulation capabilities all encompassed in a single IDE. It enables behavioral, post-synthesis and post-implementation (functional or timing) simulations for the fully integrated Vivado Simulator and 3rd party HDL simulators.

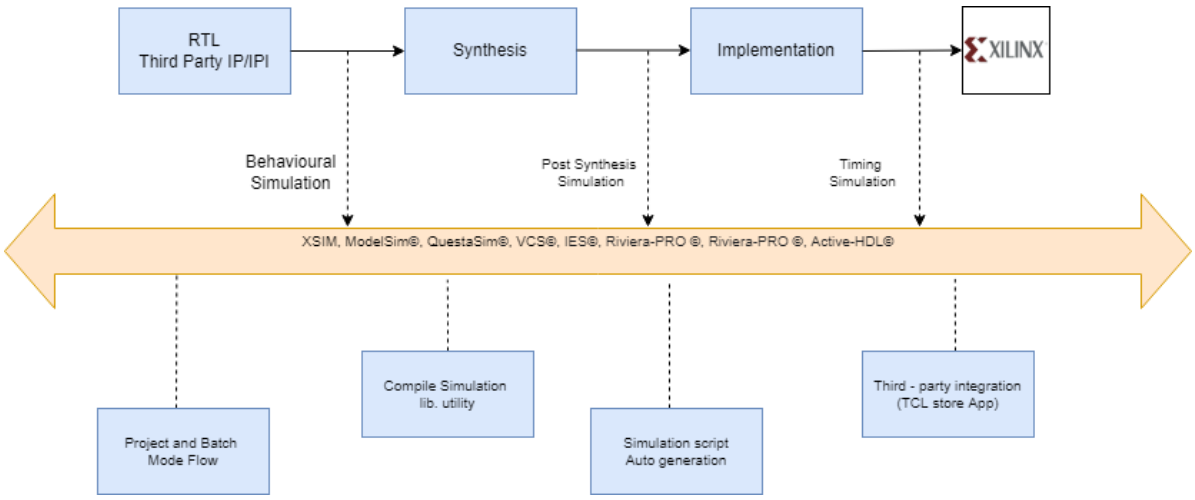

Figure 2.5: Simulation Flow in Vivado

2.2.1 Virtual Input/Output

The Virtual Input/Output (VIO) core is a customizable core that can both monitor and drive internal FPGA signals in real time. The number and width of the input and output ports are customizable in size to interface with the FPGA design. Because the VIO core is synchronous to the design being monitored and/or driven, all design clock constraints that are applied to your design are also applied to the components inside the VIO core. Run time interaction with this core requires the use of the Vivado® logic analyzer feature. It provides virtual LEDs and other status indicators through synchronous input ports.

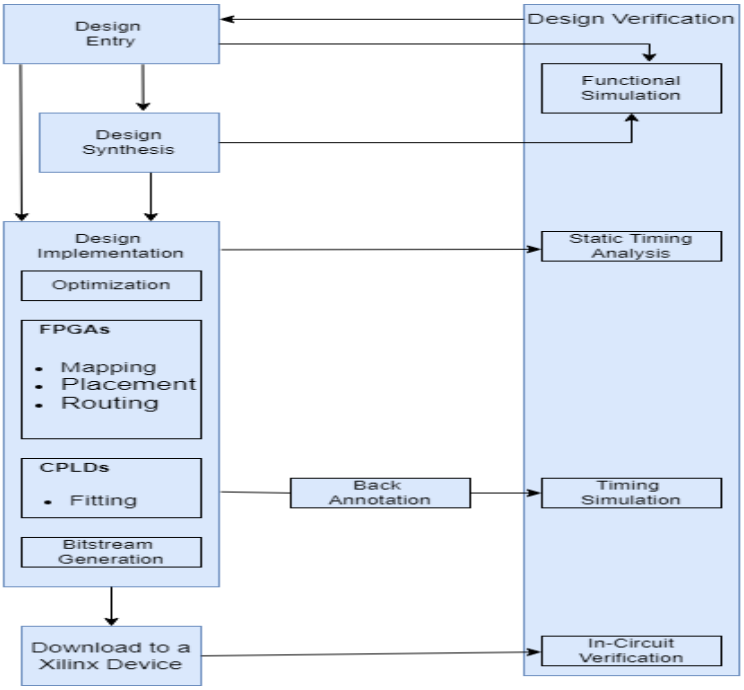

Figure 2.6: Xilinx Software Flowchart for FPGA Designs

2.2.2 Integrated Logic Analyzer

The customizable Integrated Logic Analyzer (ILA) IP core is a logic analyzer core that can be used to monitor the internal signals of a design. The ILA core includes many advanced features of modern logic analyzers, including Boolean trigger equations, and edge transition triggers. Because the ILA core is synchronous to the design being monitored, all design clock constraints that are applied to your design are also applied to the compo- nents inside the ILA core. The user can select the trigger width, data width, and data depth.

2.3 Various Available Techniques

2.3.1 TDC Architecture Approaches

2.3.1.1 Analog TDC methods

- **Time stretching method :**
  In the time stretching method, the capacitor is discharged and recharged by two current sources, where the recharging time is K times longer than the discharging time. Hence, the time resolution can be improved by the stretching factor In the time stretching method, the capacitor is discharged and recharged by two current sources, where the recharging time is K times longer than the discharging time.Hence, the time resolution can be improved by the stretching factor.
- **Time-To-Amplitude conversion method :**
  TDC based on a time-to-amplitude conversion method is realized by combining time-to-amplitude conversion and analog-to-digital converters (ADC). Through careful design and layout, analog TDC can obtain good resolution (about 8 ps) at the expense of high power consumption.

However, analog TDCs often suffer from large temperature drift and poor stability. Moreover, the area-consuming devices in analog TDC also hinder its implementation in integrated circuits [4].

2.3.1.2 Digital TDC methods

With the development of integrated circuit (IC) technologies, recent works have concentrated on all-digital TDCs which employ standard CMOS tech- nology to realize on-chip TDC. Due to their low area consumption and high conversion rate, TDCs based on application-specific integrated

cir- cuits (ASIC-based TDC) and field programmable gate arrays (FPGA-based TDC) have been widely used in recent years [4].

- **ASIC-based TDC :**
  Compared with FPGA-based TDC, ASIC-based TDC has the merits of fully customized circuits and precise control of the internal propagation delay. The resolution obtained by ASIC-based TDC can be 1–2ps. However, ASIC-based TDCs usually suffer from high development cost and long time-to-market [4].
- **FPGA-based TDC :**
  FPGA devices have become very popular for rapid system prototyping, logic emulation, and reconfigurable computing because of their much lower manufacturing cost and shorter development time [4].

2.3.2 For Coarse Measurement

2.3.2.1 Coarse Counter

It is the simplest counter which is edge-triggered and counts the number of clock signals that occurred between the start and stop signal. It considers the clock edge trigger that occurred after the start signal and the clock edge trigger that occurred after the stop signal, hence not an accurate counter in cases where the start and stop signal do not occur at the edge triggers of reference clock.

2.3.2.2 Statistical Counter

Since start, stop and clock signals are asynchronous, there is a uniform probability distribution of the start and stop signal-times between two subsequent clock pulses. This detuning of the start and stop signal from the clock pulses is called quantization error. For a series of measurements on the same constant and asynchronous time interval one measures two different numbers of counted clock pulses. Measuring a time interval using a coarse counter with the averaging method is relatively time consuming because of the many repetitions that are needed to determine the probabilities.

2.3.3 For Fine Measurement

2.3.3.1 Single Delay Line

In this architecture, buffers are used as a delay cells. The TDC based single delay line is suitable for technology scaling because of its fully

digital organization. Six buffers as delay cells are embedded into a charge-pump Delay Locked loop (DLL). Six delayed clocks with different delay can be generated by this method. A hit signal is mainly works as a sampling clock. The states of the six clocks will be samples into hit register when edge of the hit signal is positive [8].

Resolution of TDC based on single delay line is mainly dependent on clock period as well as number of buffers embedded into delay line. In FPGA clock period is limited by technology used. So by increasing number of delay elements can be useful. But Minimum number of delay cells and its delay time is also technology dependent. Moreover, there is also a problem of mismatching a delay cells in this method which does not allow integrating large number of delay cells. So, achieving higher resolution can be difficult. To improve the time resolution of this architecture, one can try to further divide the delay of the delay cells by using an array of delay lines or other techniques [8].

2.3.3.2 Array of Delay Line

For eliminating the issue of single delay line array of delay line can be used in which several Delay Locked Loops (DLLs) are employed in the array. Resolution is depends on time difference between delay cells in delay line. The reference clock is propagated by the array of DLLs. In this method, power dissipation and area will be high because use of several DLLs in array. For overcome this issue tapped delay line and vernier delay lines are widely used approaches [8].

2.3.3.3 Tapped Delay Line

Tapped delay line is consisting of one latch chain and one buffer chain. Start signal is given to buffer chain and stop signal is given to the latch chain. Delay of latch chain and buffer chain is different. In this architecture resolution is depends on difference of the delays between latch and buffer. This architecture is more area efficient than single as well as vernier delay line [8].

Achieving higher resolutions will require more effort in terms of optimization of delay element, placement and routing of delay cells in this architecture because the elements used are of different types [8].

2.3.3.4 Vernier Delay Line

The principle of the measurements originates from the Vernier ruler. Two delay lines are required. As shown in above figure, vernier delay line is consist of one latch chain and two buffer chain. Latch chain is mainly

used for holding the result. Start signal and stop signal both are given to the different buffer chains. By using the vernier method, the small time difference can be measured. To realize the TDC using vernier delay line, two DLLs should be employed. Thus, the synchronization of the multi-phase clock is very important in this circuit [8].

This architecture is more area efficient than single delay line and less area efficient than tapped delay line. Compared to the tapped delay line TDC this method requires more area in FPGA, but achieving finer resolutions is easier as the delay elements used are of the same type [8].

2.3.3.5 Gated Ring Oscillator

Gated ring oscillator based delay line contains gated ring oscillator, many counters and arbitrary adder. This TDC operates only when the "Enable" signal is high level and stops when this signal is at low level. The outputs of the gated ring oscillator can be used as the clocks which drive the counter to counting numbers. Counter will get reset when the enable signal is at low level. Binary adder will obtain the total number of all counters. And the measured time interval will be proportional to sum of counted numbers by binary counter [8].

2.3.3.6 Pulse Shrinking Delay Line

In pulse shrinking delay line, non-homogeneity is used to to create delay line. In this architecture, a Reset signal is used to ensure the Tout is at Low level at the beginning. The input time interval is shrieked with each cycle in the delay line with a fixed width. The output of the delay line is then feedback to the input to AND gate for circular operation. A high-resolution counter is driven by Tout and generates digital outputs which are proportional to the measured time interval [8].

In this kind of delay line, inverters are used as delay cells and two types of inverters are necessary. First type can be the standard inverter with the gain of one unit. The second type is the inverter with the gain of β unit. Because of the difference of the input capacitance and equivalent ON resistance, for fixed time interval pulse will be shrinking. This shrinking delay interval depends on the dimension of the transistors, threshold voltage, power supply, temperature and other parameters. Pulse shrinking delay line can be implemented in FPGA as well as standard CMOS technology whose cost is much higher than FPGA [8].

Vernier Ring Oscillator Delay Line

The important element of this design is two precise re-triggerable ring oscillators of very small difference in time periods (ΔT). These oscillators

are used to determine the time difference between two pulses START and STOP. The Start and Stop pulses will enable the slow and fast oscillators respectively. The slow oscillator (time period T1) is triggered by START and the fast oscillator (time period T2) is triggered by STOP. In this technique, T2 less than T1 and STOP arrives after START, at some point the rising edge of the fast oscillator will coincide with rising edge of the slow oscillator. This coincident will be detected by a phase detector. In this technique, slow oscillator and fast oscillator will serve as a clock coarse counter and fine counter respectively and the number of clock pulses (n1 and n2) will be counted. These counters will stop counting when the phase detector detects the phase coincidence of the oscillators [8].

Benefit of using the oscillators is that it reduces the matching requirements on the delay buffers used in vernier delay line. This feature is very useful to reduce the temporal uncertainties of Vernier delay line based TDCs which is mainly caused by delay variation of buffers. Disadvantage of this architecture is that it takes so many cycles to complete a single time interval (longer dead time). Conversion time is also high in this type of delay line than other delay lines which can detect time interval every cycle [8].

2.3.3.7 Oversampling method

In this method, single delay line with four buffers which can give exactly 90 degree phase shift has been used. Delay locked loop has been used to achieve exact 90 degree phase shift for this technique. Delay Locked Loop provides constant phase shift to output clocks. DLL has a Phase Detector which measures the phase error and to convert this information into voltage charge pump is required. Clock signal is given to the delay element. In this technique, four phase shifted clocks (0, 90,180, 270 deg) are generated. This four shifted clocks works as a clocks which triggers four different counters. Resulting count has four times higher resolution than reference clock frequency [8].

2.3.3.8 Tapped Delay Line using Multiplexer

The main advantage of vernier delay line over tapped delay line is that vernier delay line has same delay elements for giving delay in both delay lines rather, tapped delay line approach has different delay elements for both delay line which requires careful delay matching between latch and buffer chains. But tapped delay line is more area efficient as it uses fewer elements than vernier delay line. By using only a same element as

latch and buffer, it could possible to narrow down the delay difference. Therefore use of multiplexers instead of both latch and buffer is much useful. Carry chain multiplexers with dedicated routes can be used in FPGA for creating this kind of design [8].

The main advantage is that a multiplexer can be easily configured to behave as a buffer as well as a latch, just by changing the way the output is connected. This allows the two parts of the chain to be very precisely matched. It can be also called as hybrid approach. The latch and buffer chains are based on the Tapped Delay Line approach, but the individual delay elements are realized using the same underlying hardware, thus giving delay matching similar to the VDL method [8].

2.4 Related Works

2.4.1 Low resource FPGA-based Time to Digital Converter

Authors: Alessandro Balla, Matteo Beretta, Paolo Ciambrone, Maurizio Gatta, Francesco Gonnella, Lorenzo Iafolla, Matteo Mascolo, Roberto Messi, Dario Moricciani, Domenico Riondino

They devised and implemented a 32-channel TDC with a precision of 255 ps and low non-linearity effects on a Xilinx Virtex-5 FPGA in this article. Due to the need for a specialised data collecting system and interface, the system was constructed on the FPGA. The Nutt interpolation method and 4xOver-sampling technology were employed in the TDC's architecture. Integral Non-Linearity (INL) is the divergence of the input-output characteristic from the ideal straight line, whereas Differential Non-Linearity (DNL) is the deviation of a single quantization step from the ideal value of 1 Last Significant Bit (LSB). This work was examined in order to learn more about differential and integral non-linearity, as well as their testing and outcomes.[9]

2.4.2 Pico-TDC: a novel FPGA-based TDC with 2.2ps RMS timing resolution

Authors: T. Sui[1], Z. Zhao[2], S. Xie[1], Q. Huang[2], J. Xu[1], Q. Peng[3] [1] Huazhong University of Science and Technology, Wuhan, China[2] Shanghai Jiaotong University, Shanghai, China [4] Lawrence Berkeley National Laboratory, Berkeley, USA

Their goal was to create a low-cost, high-performance TDC that could

match the demands of the next generation sub-10ps TOF- PET camera. They provided a new way for constructing TDCs on FPGA termed Pico-TDC method in the study. The Pico-TDC approach is unique in that it uses single FPGA registers as low-precision TDCs and then combines many of those low-precision TDCs to create a high-precision TDC. This research was looked at to see what other measures may be taken to improve the TDC's performance.[10]

2.4.3 264 Channel TDC Platform Applying 65 Channel High Precision (7.2 psRMS) FPGA Based TDCs

Authors: Cahit Ugur[1], Grzegorz Korcyl[2], Jan Michel[3], Manuel Penschuk[3] and Michael Traxler[1]

[1] GSI Helmholtz Centre for Heavy Ion Research, Darmstadt, Germany

[11] Jagiellonian University, Krakow, Poland

[3] Goethe-University, Frankfurt, Germany

he architecture of a 65-channel TDC built on a single FPGA is thoroughly addressed in this work, as well as test results illustrating some of the quality measurements. For time measurements, the TDC uses the interpolation approach, and for precision, it employs the Wave Union Launcher method. On all channels, the TDC has a maximum precision of 7.2 ps RMS and a $<$ 14 ps RMS. To get around the minimum pulse width restriction, a semi-asynchronous pulse stretcher is used, which has been confirmed to allow for a pulse width measurement of $<$500 ps. This work describes the Semi-asynchronous stretcher, which was effective for limiting the minimum pulse width. [11].

2.4.4 A Multichannel High-Resolution (less than 5 ps RMS between two channels) Time-to-Digital Converter (TDC) Im- plemented in a Field Programmable Gate Array (FPGA)

Authors: Eugen Bayer[1], Peter Zipf [1] and Michael Traxler[2]

[1] University of Kassel, Germany

[2] GSI Helmholtz Centre for Heavy Ion Research, Darmstadt, Germany

This study provides a new concept for time interpolation that uses dedicated carry-chains and can do two time measurements in a single carry-chain per hit. Multiple ($>$2) measurements can be done in a single chain

per hit, resulting in a temporal resolution of $approx2$ ps RMS between two channels in this architecture. This work was examined in order to examine some advanced interpolation approaches for the tapped delay line method in greater depth. These strategies are used to improve a TDC channel's single channel resolution using the TDL method. An incoming start signal begins numerous interpolations in all procedures, resulting in N measurements that are averaged and recorded. [12].

2.4.5 Time-to-digital-converter based on multiple-tapped-delay-line

Authors: Dariusz Chaberski[1]
[1] Nicolaus Copernicus University, Poland

The idea, operation, analysis, design, and test results of a time-to-digital converter (TDC) based on multiple-tapped-delay-lines are described in this article (MTDL). Multiple TDLs with low equivalent resolution (323 ps on average) were used to achieve a measurement equivalent resolution of approximately 5:8 ps. For each hit, the aim of measurement is to combine all tapped-delay-line results into a single high-precision timestamp. This study included a detailed mathematical analysis that was required to complete this process. [13].

2.4.6 A Coarse-Fine Time-to-Digital Converter

Authors: Ya-Qian Chen[1], Li-Ya Meng[1] and Xiao-Gang Lin[1]
[1] Key Laboratory of Optoelectronic Technology and System, P.R. China

A high-precision TDC based on a three-level conversion technique is proposed in this study. A TDC with excellent resolution and a large dynamic range is required as integrated circuit technology advances. It's not enough to meet just one goal. The Coarse-Fine TDC suggested in this study contained three steps, similar to the nutt interpolation method, in terms of precision and dynamic range. This study was researched in order to gain a better understanding of these three stages.[14].

2.4.7 New Design-methodology of High-performance TDC on a Low Cost FPGA Targets

Authors: Foudil Dadouche, Timothé Turko, Wilfried Uhring, Imane Malass, Norbert Dumas, Jean-Pierre Le Normand

This project seeks to propose a Time-to-Digital Converter (TDC) design technique for low-cost Field-Programmable Gate Array (FPGA) targets. To begin, the paper shows how to take use of the presence of carry chains in the FPGA's elementary logic parts to improve TDC resolution. After that, it explains how to use the Chip Planner tool to place the system's partitions in user-defined physical zones. This allows TDC divisions to be placed in such a way that routing pathways are restricted. As a result, the user has effective control over the propagation delay over the connection network.The paper concludes with a case study demonstrating the design and construction of a high resolution TDC dedicated to a time correlated single photon counting system using the technique described. The INL, DNL, and mean Jitter values (22 ps rms, 13 ps rms, and 26 ps rms, respectively) found utilising a low-cost FPGA target Cyclone family are all highly promising and suited for a wide range of rapid applications.[15]

2.4.8 Low-Cost FPGA TDC With High Resolution and Density

Authors: Jiajun Zheng, Ping Cao, Di Jiang, Qi An

For years, time-to-digital converters (TDCs) in field programmable gate arrays (FPGAs) have been developed and have achieved exceptional performance. If TDC is implemented in an FPGA, a tradeoff between performance and cost must be addressed in an application with a large number of channels. Massive channels with a high event rate present significant obstacles for real-time transmission of huge data rates. As a result, developing time digitizers that can handle a high data rate while maintaining a satisfactory resolution and cost is critical. A TDC prototype with sandwich structure developed in a low-cost FPGA is shown in this research. It has 320 time measurement channels (ten TDC daughter cards) and can be used as a digitizer in compressed baryonic matter time-of-flight experiments to evaluate super module detectors. [16]

2.4.9 A fully fledged TDC implemented in field-programmable-gate-arrays

Authors: Jinhong Wang, Shubin Liu, Qi Shen, Hao Li, Qi An

The goal of this work is to construct a fully functional FPGA-based

TDC in XILINX XC4VFX60 FPGAs with self-testing, temperature variation correction, and trigger-matching capabilities. Self-testing is done using statistical methods, and the resolution of the delay chain is determined at its temperature and supplied voltage. The resolution varies with the temperature of the environment, thus a self-test from 30 to 60 degrees Celsius was used to compensate. The RMS of time measurement after compensation and INL calibration is less than 30 ps per channel of the total six, with a resolution of roughly 50 ps. Trigger-matching is done with content addressable memory and two programmable parameters: trigger-latency and matching window.[17]

2.4.10 Several Key Issues on Implementing Delay Line Based TDCs Using FPGAs

Authors: Jinyuan Wu (Fenni National Accelerator Laboratory, Batavia, IL, USA)

This work covers the development of the Wave Union TDC, a revolutionary FPGA TDC scheme that uses multiple measurements to improve time measurement precision, as well as various other subjects in FPGA delay line based TDCs. First, FPGA-specific concerns such as delay line selection in different FPGA generations and encoding logic are studied. Following that, common concerns for both FPGA and ASIC TDCs are explored, including coarse time counter implementation techniques, bin-by-bin calibration, and noise issues caused by single ended signals. The document describes several resource/power-saving design methods for various processing phases. [18]

Chapter 3

Proposed System

3.1 Problem Statement

To develop a multi-channel time-to-digital converter (TDC) with the help of a Field Programmable Gate Array (FPGA) device. The designed TDC will enable us to measure time delays between any two signals (start and stop) with an accuracy of sub-nanoseconds. The TDC should be integrated with a UART module for transmission of data.

3.2 Scope

The scope of our project is limited to the generation of a 'Multi-channel Time to Digital Converter (TDC) integrated with UART communication'. This means that our project scope is limited to the creation of Coarse Counter and Fine Counter along with their integration with UART communication. The above mentioned Coarse Counter will be used to measure longer time intervals (steps of 10 nanoseconds) and the Fine Counter will be used to measure smaller time interval (steps of 238 picoseconds for 14 bit fine counter and 416 picoseconds for 12 bit fine counter). This system can be deployed where precision timing measurement and large data transmission is required.

3.3 Proposed System

Our proposed system is a hybrid counter which uses the Nutt interpolation method for time measurement. The circuit for this will be developed on an FPGA board (Nexys A7 FPGA). Finally the communication in this system will be done using the UART protocol.

A TDC converts a time interval between different pulses into digital numbers. A TDC has two inputs, a start pulse which arrives first and a stop pulse which comes later. The difference between positive edges of start pulse to positive edge of stop pulse is called as input time interval. The resolution of this device is limited by the speed of the reference clock and it cannot be higher than a single clock period. For getting higher resolution, high clock frequency is needed which leads to major power dissipation. Constraints like on chip clock frequency of the particular device is the key factor that limits the application of this kind of architecture.

3.3.1 Nutt Interpolation Method

The nutt interpolation method is used to create the hybrid counter. The key benefit is that it can use the coarse counter to measure vast distances

and the fine counter to achieve resolution finer than the clock cycle period.

As illustrated in Figure 3.1 [9], the time interval T to be monitored is divided into three sections. The coarse counter measures one long interval, Δt12, in real time; the remaining two short intervals, Δt1 and Δt2 (at the beginning and end of the interval T), are measured by a high resolution TDC, i.e., the fine counter. The time between the positive edge of the STOP/START pulse and the next positive edge of the clock is measured by the fine counter.

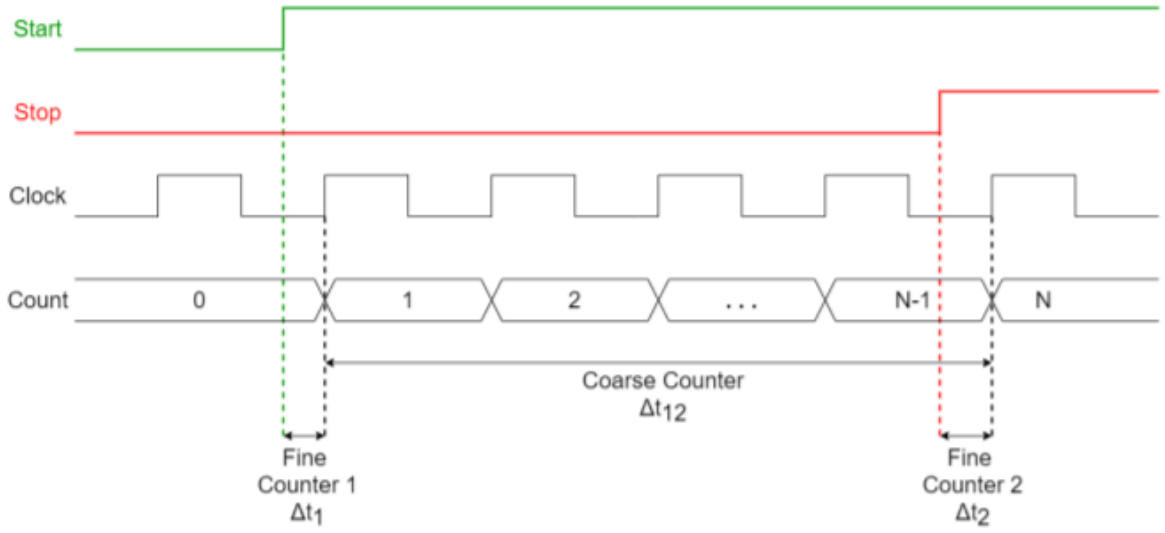

Figure 3.1: Nutt Interpolation Method

3.3.2 Coarse Counter

The coarse measurement in this project was done with a 100 MHz clock. As a result, the coarse counter measures time in 10ns increments. Because the fine counter cannot measure lengthy timing intervals, this is a key component. The coarse counter is a simple counter that counts the FPGA clock to measure time intervals.

3.3.3 Fine Counter

The fine counter is used to measure time intervals in the range of picoseconds. This timing measurement is difficult to achieve as the FPGA internal clock frequency is 100 MHz and thus direct counting at picosecond interval is not possible.

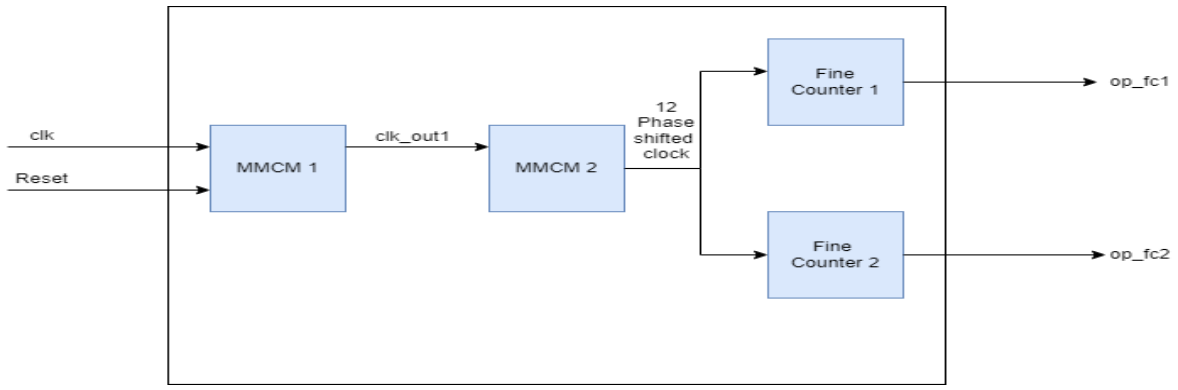

Figure 3.2: Simplified Block Diagram of 12-bit Fine Counter Module

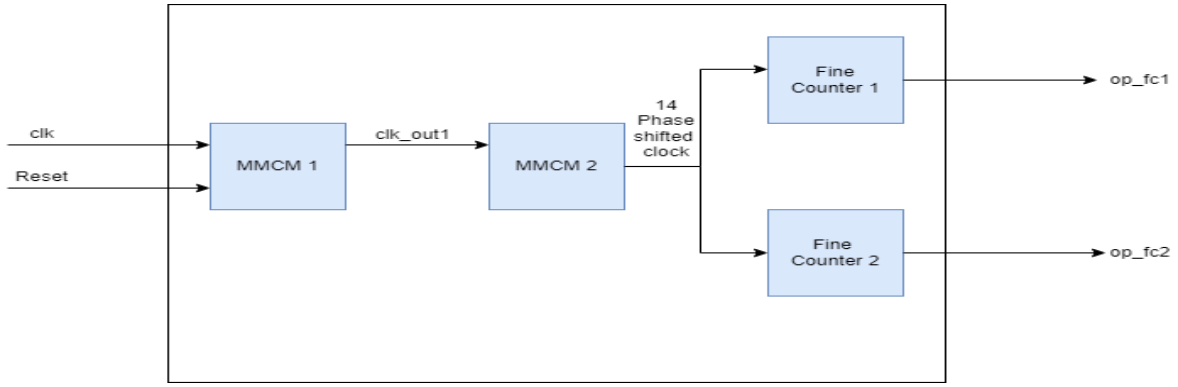

Figure 3.3: Simplified Block Diagram of 14-bit Fine Counter Module

In this system, the fine counter is implemented with the help of the Mixed-Mode Clock Manager (MMCM) module of the FPGA. Here we would be cascading two MMCM's to get 12/14 equally phase shifted clocks. While cascading the zero phase shifted clock from first MMCM is given as input to the second MMCM.

The clock of 100 MHz is given as input to MMCM which gives 12 equally phase shifted 100MHz clocks as output in 12 bit fine counter. These phase shifted clocks latch the start and stop signal to 12 bit thermometer code. The 12 bit obtained from thermometer code is used to calculate the fine measurement. Table 3.1 shows the clocks generated by the MMCM module for use in the 12 bit fine counter. Since, the time period of 100 Mhz clock is 10000 ps, the resolution of the fine counter is 10000 ps $\div$ 12 = 833.33 ps.

In the 14 bit counter, the clock of 100 MHz is given as input to MMCM which gives 14 equally phase shifted 300 MHz clocks as output in 14 bit fine counter. These clock also do latch the start an stop signal to 14 bit thermometer code and then this code is used to calculate the 14 bit fine measurement.Table 3.2 shows the clocks generated by the MMCM module for use in the 12 bit fine counter.Since the time period of the 300

MHz clock is 3333.33 ps, the resolution of the fine counter is 3333.33 ps ÷ 14 = 238.095 ps.

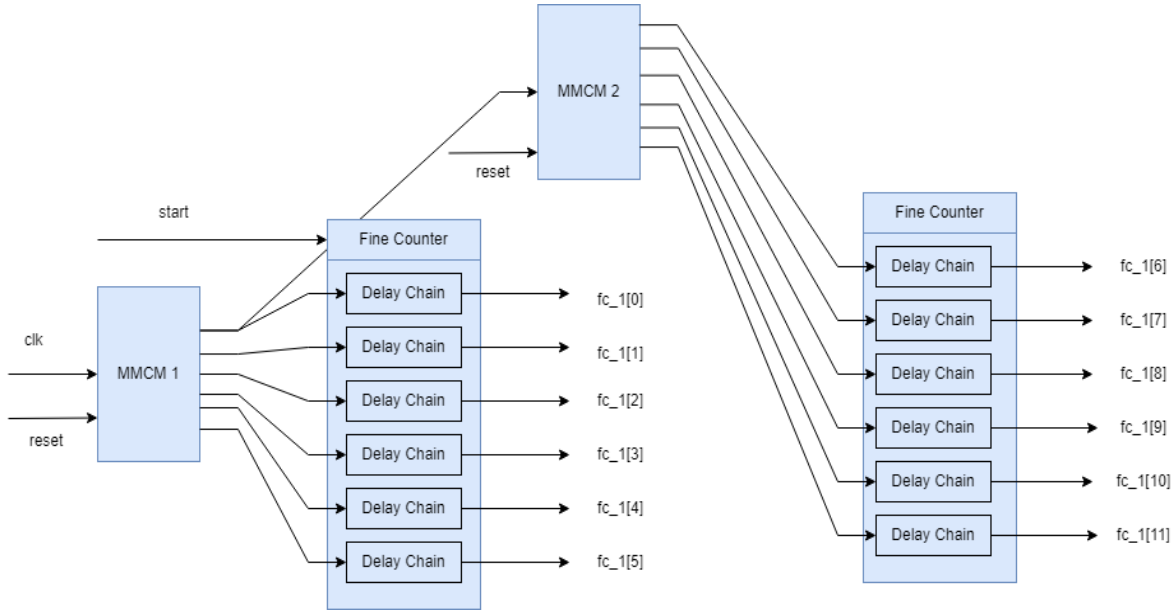

Figure 3.4: Flowchart for 12-bit Fine Measurement

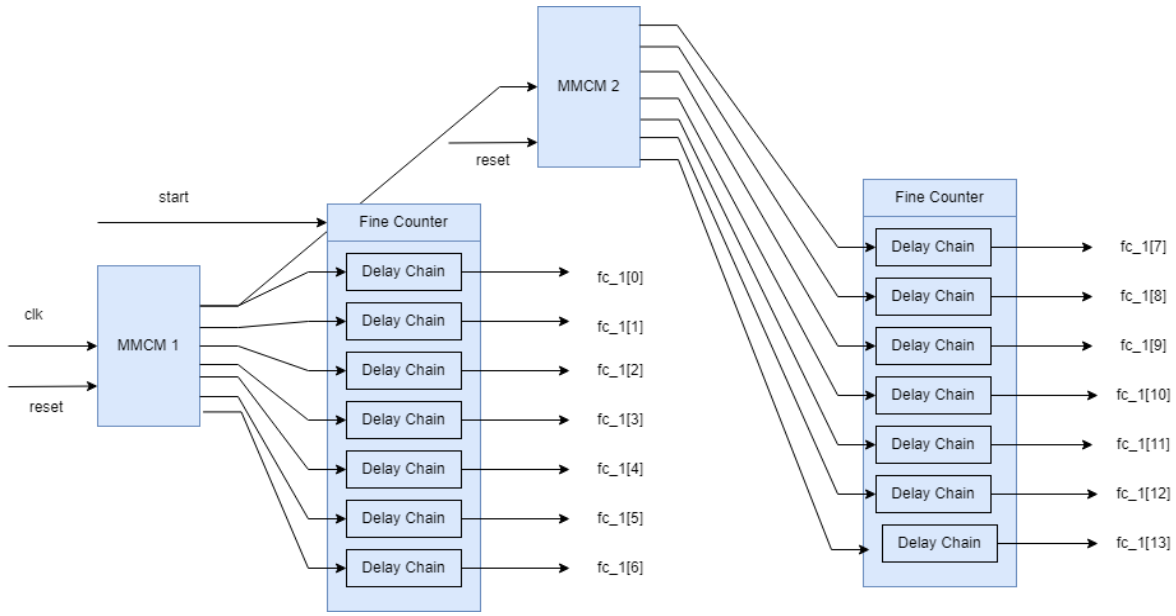

Figure 3.5: Flowchart for 14-bit Fine Measurement

	Output Clock	Frequency (MHz)	Phase (Degree)	Duty Cycle (%)
MMCM 1	clk_out1	100.000	0.000	50
	clk_out2	100.000	30.000	50
	clk_out3	100.000	60.000	50
	clk_out4	100.000	90.000	50
	clk_out5	100.000	120.000	50
	clk_out6	100.000	150.000	50
MMCM 2	clk_out7	100.000	180.000	50
	clk_out8	100.000	210.000	50
	clk_out9	100.000	240.000	50
	clk_out10	100.000	270.000	50
	clk_out11	100.000	300.000	50
	clk_out12	100.000	330.000	50

Table 3.1: Output Clocks of MMCM used in 12-bit Fine Counter

	Output Clock	Frequency (MHz)	Phase (Degree)	Duty Cycle (%)
MMCM 1	clk_out1	300.000	0.000	50
	clk_out2	300.000	25.000	50
	clk_out3	300.000	50.000	50
	clk_out4	300.000	75.000	50
	clk_out5	300.000	100.000	50
	clk_out6	300.000	125.000	50
	clk_out7	300.000	150.000	50
MMCM 2	clk_out8	300.000	175.000	50
	clk_out9	300.000	200.000	50
	clk_out10	300.000	225.000	50
	clk_out11	300.000	250.000	50
	clk_out12	300.000	275.000	50
	clk_out13	300.000	300.000	50
	clk_out14	300.000	325.000	50

Table 3.2: Output Clocks of MMCM used in 14-bit Fine Counter

Chapter 4

Design Of the System

4.1 UML diagrams

4.1.1 Use Case Diagram

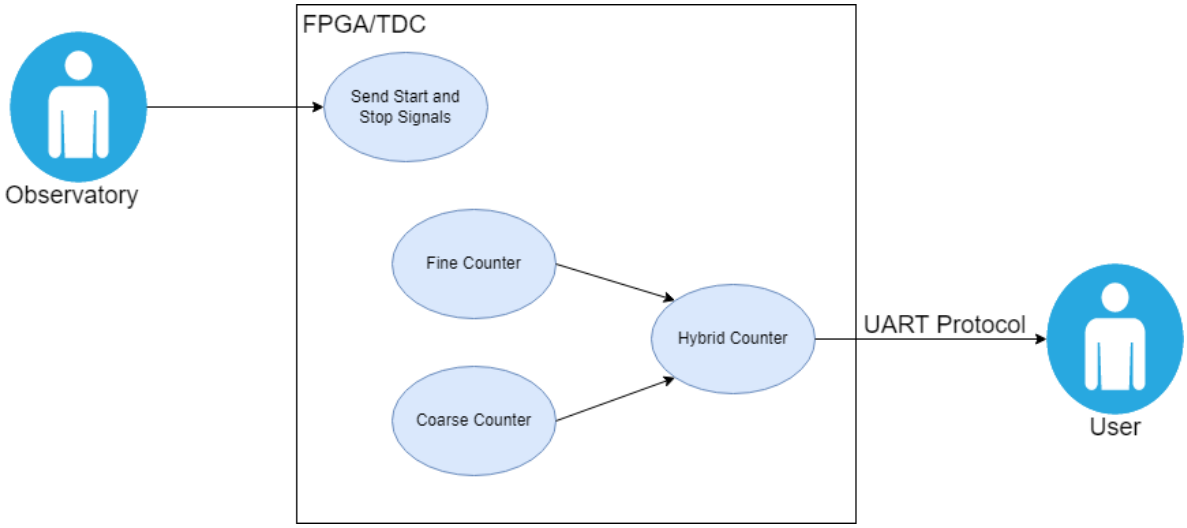

Figure 4.1: Use Case Diagram

The system's interaction with the outside world is depicted in the UML use case diagram below. The astronomical observatory and the user are the two characters in this diagram. Another receiving end of the same observatory may be the end user here. Using optical sensors, the observatory detects light rays and sends the start and stop pulses to the TDC system. The TDC system is built on an FPGA and measures and outputs the final time interval using a hybrid counter, which is a combination of coarse and fine counters. The end user reads this result.

4.1.2 Activity Diagram

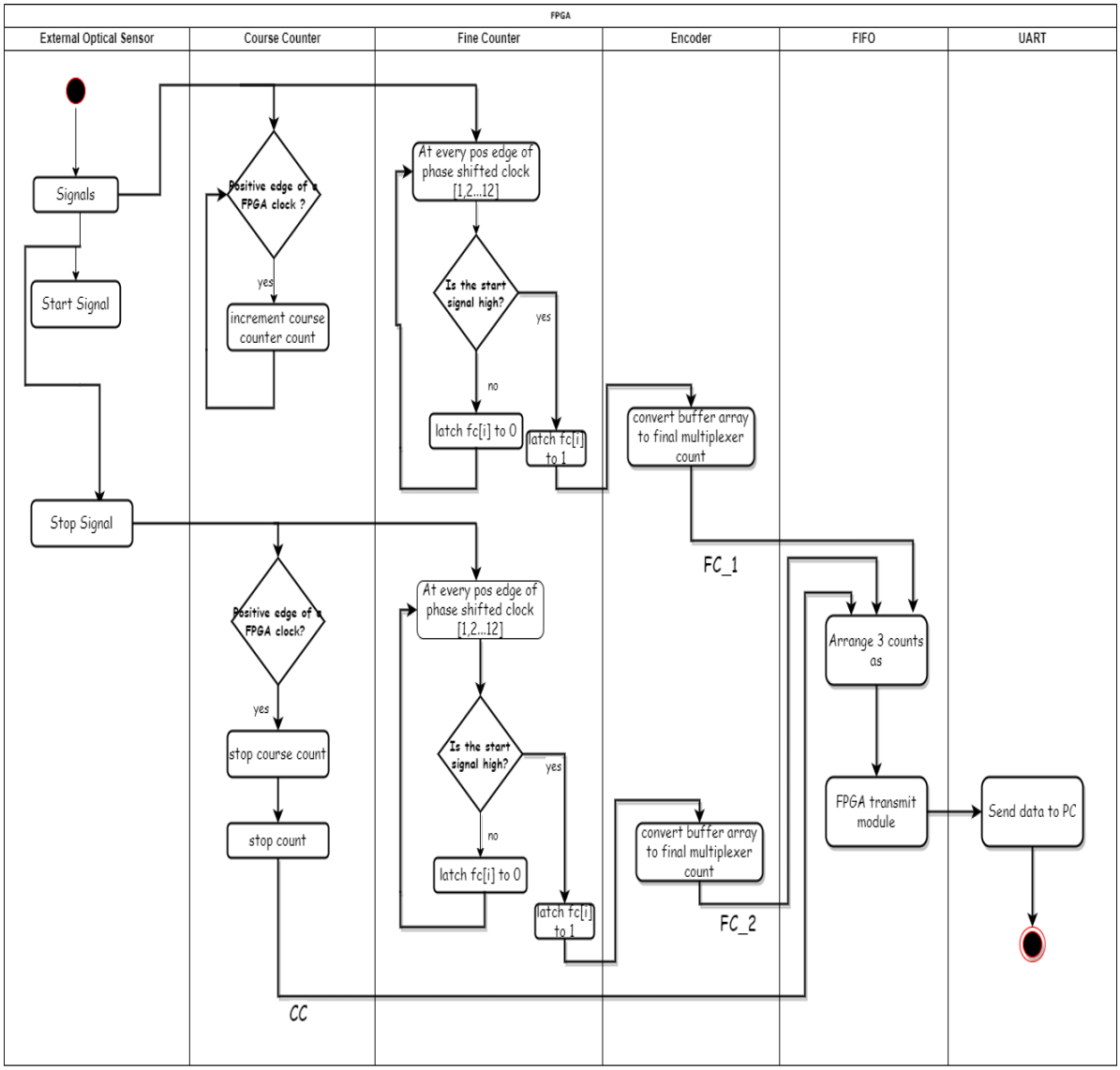

Figure 4.2: Activity Diagram

The Activity Diagram with the workflow within the system. It shows the choices, decisions, behaviour and control flow from starting point to ending point of the execution

4.1.3 Sequence Diagram

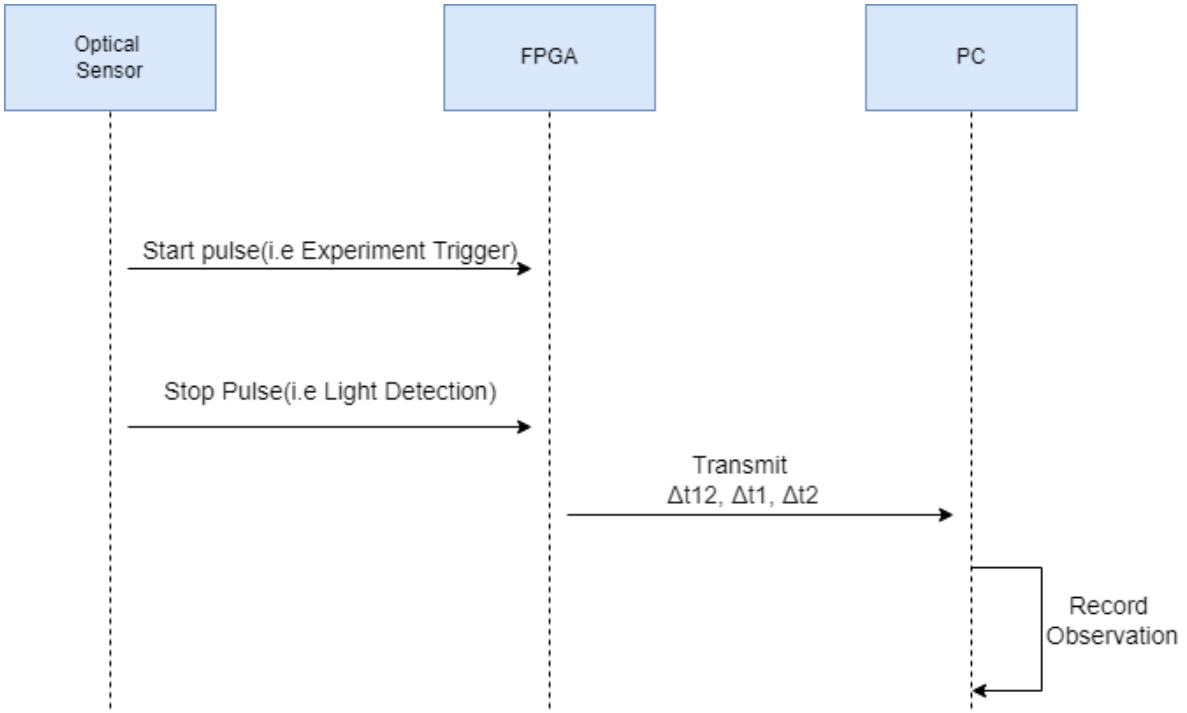

Figure 4.3: Sequence Diagram

The Sequence Diagram with object interactions arranged in time sequence. It is used represent the the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

The start of the experiment triggers the start pulse and the detection of light by the optical sensors creates the stop pulse. These two serve as input to the FPGA where the TDC counts the time interval between the two. The FPGA divides the time interval into 3 parts and generates 3 counts (Δtc, Δt1 and Δt2). The values are repeatedly generated and stored accordingly.

4.2 Software Development Life Cycle

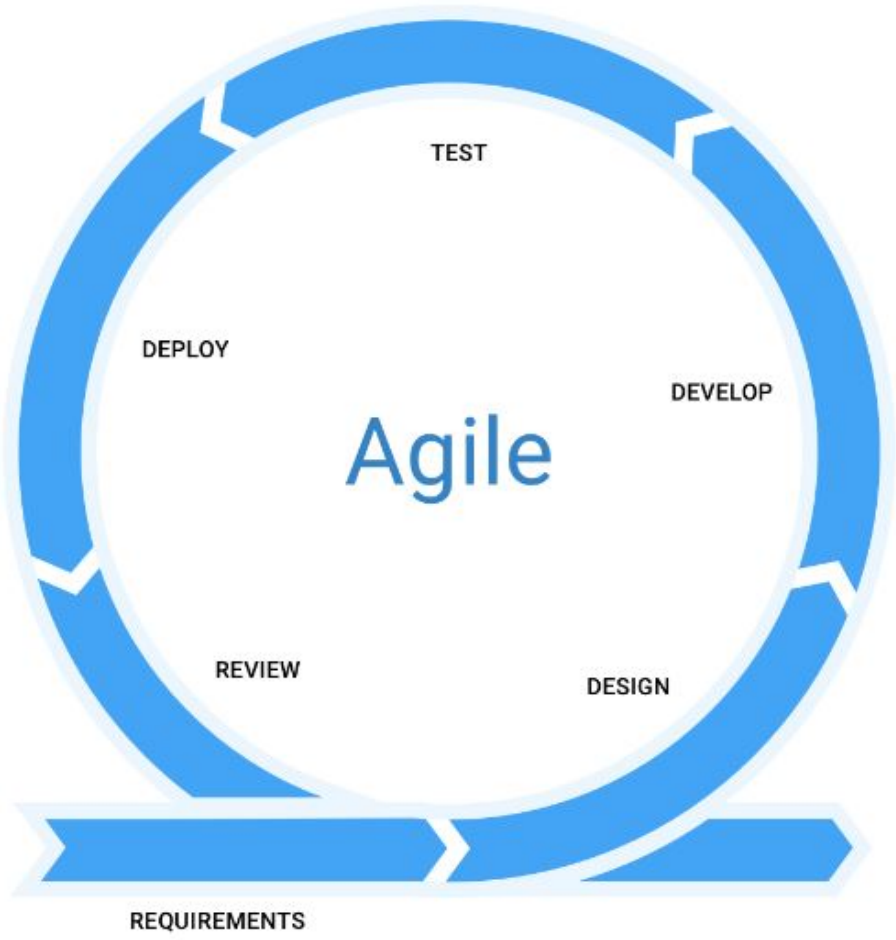

Figure 4.4: The Agile Software Development Life Cycle Model that was followed

The software life cycle model used here is the Agile development model. Agile methodology is based on collaborative decision making between requirements and solutions teams, and a cyclical, iterative progression of producing working software.

Work was done in regularly iterated cycles, or sprints, that usually lasted one to two weeks. Each sprint had a stack of new and old requirements known as the Backlog. Regular Scrum meetings (Daily meetings) took place between the development team members during a sprint. The Scrum meetings were performed to share the status of the work being completed on the backlog of the sprint and to identify potential issues to be added to the backlog of the next sprint.

4.3 Cost Analysis

The major cost of the project is of the Nexys A7-100T FPGA Kit. The entire project can be deployed on the FPGA kit along with a WIZ830MJ module, a passive communication PCB and an RJ45 Ethernet cable. As of 18 Nov, 2020, the Nexys A7-100T FPGA Kit costs $265. An additional cost of $52 would be incurred for the WIZ830MJ module, PCB manufacturing process and RJ45 Ethernet Cable. Thus, the total cost amounts to $317.

The expected development time is around 8 to 10 months

4.4 Hardware and software requirement

The following are the mandatory minimum hardware and software requirements to run the project.

- Hardware Requirements:
  - RG58 USB Coaxial cable
  - Nexys A7-100T FPGA Kit: FPGA to write the bit file on
- Software Requirements
  - Vivado Design Suite: Required only to edit the project.
  - Python3: Required only to run the UART receiver code created.
  - Adept2: Optional. To be used in windows only when Vivado is not available.

4.5 System architecture

4.5.1 Block Diagram

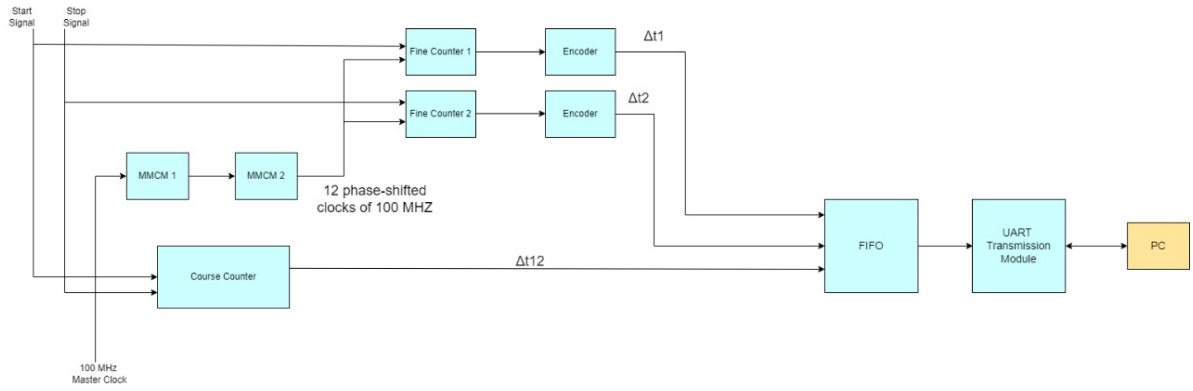

Figure 4.5: Block Diagram 12-bit

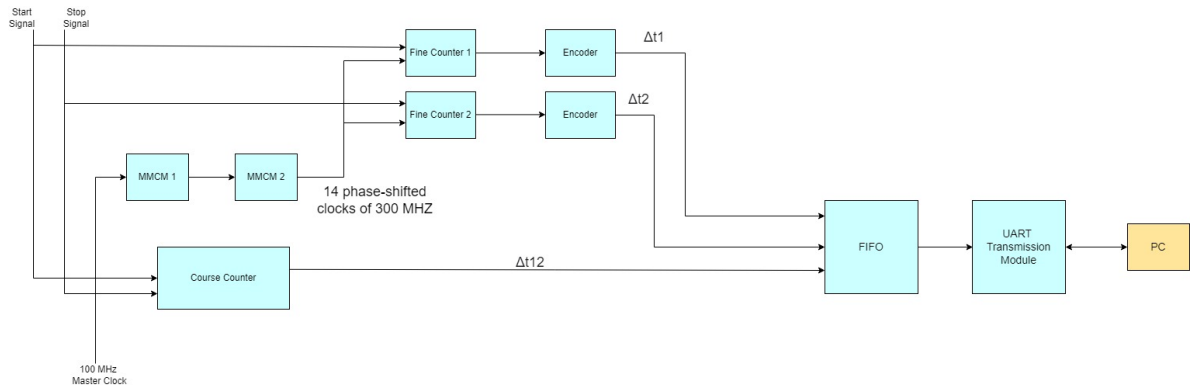

Figure 4.6: Block Diagram 14-bit

The start and stop pulses are transmitted to FPGA from the observatory. The fine counter module 1 is given start and clock pulse as input whereas fine counter module 2 is given stop and clock pulse as input. The fine counter outputs the time intervals Δt1 and Δt2. These are the count ofpicoseconds from the edge of the start and stop pulse respectively to the next positive edge of clock. The start, stop and clock pulse are given as input to the coarse counter module. This outputs the Δtc time interval which is the time between start and stop pulse in the order of 10ns. The coarse counter output and encoder outputs are passed to FIFO module which manages and arranges the data after its arrival. This final output is transmitted to the User/PC by using the UART module. The final count is calculated as follows : $\Delta T = \Delta tc + \Delta t1 \ \ \Delta t2$

Chapter 5

Result and Discussion

The coarse counter technique designed for this project was implemented as a gated counter. There are two inputs, the source signal and gate signal. The function of the gated counter is to count the number of source pulses occurring when the gate is open (HIGH). This is done by incrementing the count register if a positive edge of the source signal is encountered while the gate is open. The count is updated at each positive edge of the clock cycle. The circuit was described using Verilog Hardware Description Language in Vivado Design Suit software.

The fine counter of the system utilizes MMCM of FPGA. The input given to MMCM is the 100MHz primary clock. MMCM outputs twelve equally phase shifted clock of 100MHz. Fine counter is designed using these twelve output clock from MMCM. At every positive edge of clock the signal is latched. The 12 bit thermometer code is evaluated to get Fine Counter result.

5.1 Screenshots of the System

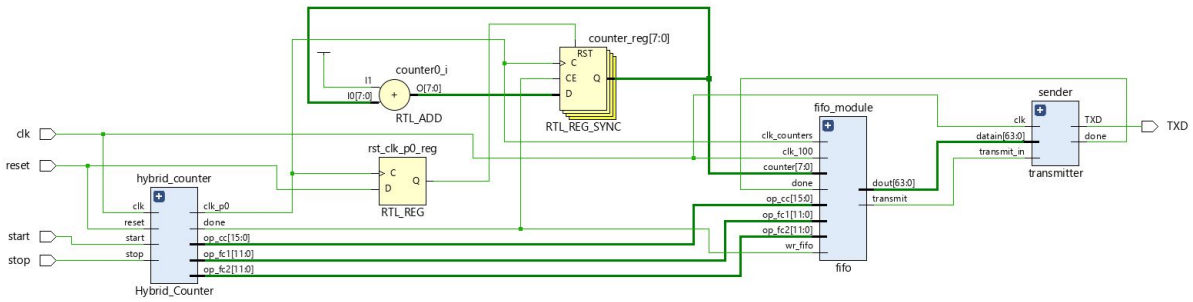

Figure 5.1: RTL Schematic of Top Module of TDC

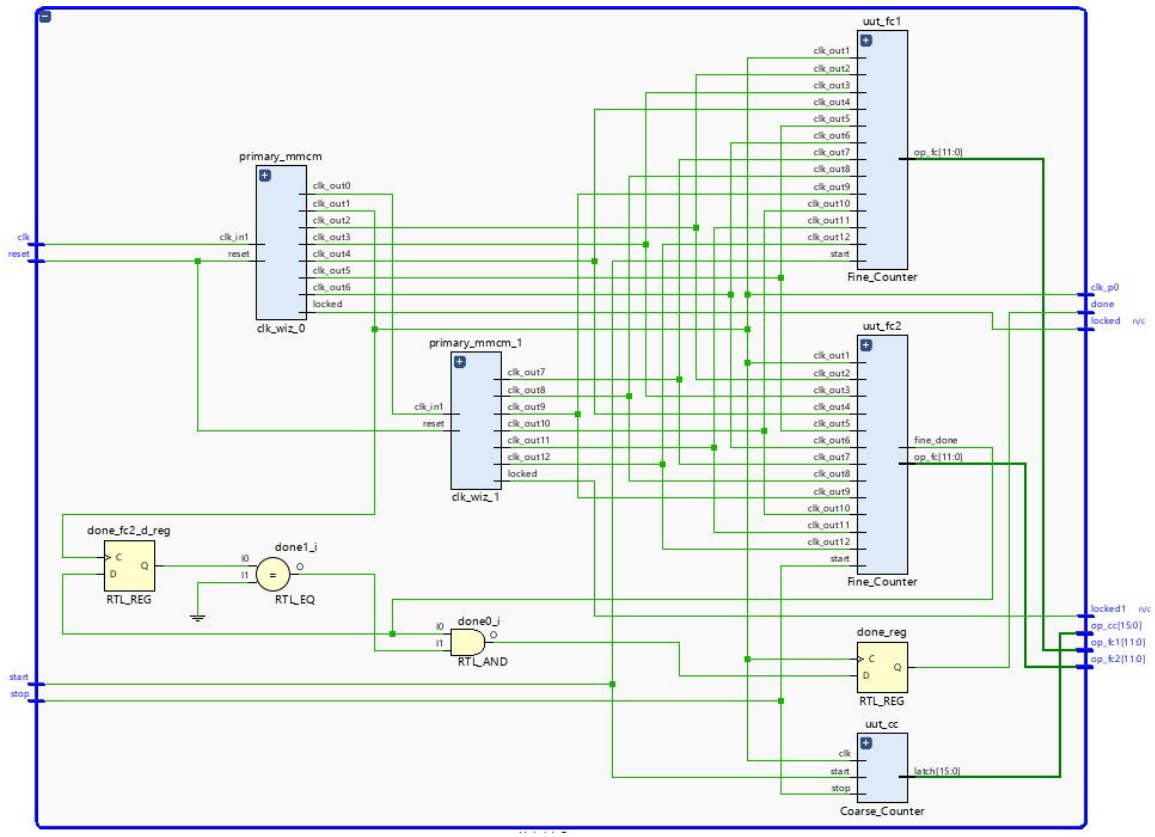

Figure 5.2: RTL Schematic of Top Module of Hybrid Counter

5.2 Sample Code

```
1  `timescale 1ns / 1ps
2
3  module top(
4      input wire start,
5      input wire stop,
6      input wire clk,
7      input wire reset,
8      output wire TXD
9      );
10
11     wire done_uart,transmit,done,locked;
12     reg wr_fifo,flag1,flag2,flag3;
13     reg [7:0] counter;
14     wire [11:0] op_fc1,op_fc2;
15     wire [15:0] op_cc;
16     wire [63:0] dout;
17     wire clk_p0;
18     reg rst_clk_p0;
19
20     initial
21     begin
22         wr_fifo=0;
23         flag1=0;
24         flag2=0;
25         flag3=0;
26         counter=0;
27     end
```

```
28
29      Hybrid_Counter hybrid_counter(
30          .start(start),
31          .stop(stop),
32          .clk(clk),
33          .reset(reset),
34          .op_fc1(op_fc1),
35          .op_fc2(op_fc2),
36          .op_cc(op_cc),
37          .locked(locked),
38          .locked1(locked1),
39          .done(done),
40          .clk_p0(clk_p0)
41          );
42
43      always @(posedge clk_p0)
44      begin
45          rst_clk_p0 <= reset;
46          if (rst_clk_p0)
47            counter <= 8'h0;
48          else
49            if (done)
50              counter <= counter + 1;
51            else
52              counter <= counter;
53      end
54
55      fifo fifo_module(
56          .clk_counters(clk_p0),
57          .clk_100(clk),
58          .op_fc1(op_fc1),
59          .op_fc2(op_fc2),
60          .op_cc(op_cc),
61          .counter(counter),
62          .done(done_uart),
63          .wr_fifo(done),
64          .transmit(transmit),
65          .dout(dout),
66          .rst(reset)
67          );
68
69      transmitter sender(
70          .clk(clk),
71          .transmit_in(transmit),
72          .datain(dout),
73          .TXD(TXD),
74          .done(done_uart)
75          );
76
77  endmodule
```

Listing 5.1: Top Module of the Project for 12 bit phase shift

```
1   `timescale 1ns / 1ps
2   module Fine_Counter (
3    input clk_out1,
4    input clk_out2,
5    input clk_out3,
6    input clk_out4,
7    input clk_out5,
```

```
 input clk_out6,
 input clk_out7,
 input clk_out8,
 input clk_out9,
 input clk_out10,
 input clk_out11,
 input clk_out12,
 input start,
 output reg [11:0] op_fc,
 output reg fine_done
 );

wire op1 ,op2 ,op3 ,op4 ,op5 ,op6 ,op7 ,op8 ,op9 ,op10 ,op11 ,op12 ;
reg state_flag;
reg [11:0] fc;
reg flag;
reg done ;
initial begin
flag =0;
done=0;
fc =12'b0;
op_fc = 12'b0;
end

always @ (*) begin
 fc [0] <= op1;
 fc [1] <= op2;
 fc [2] <= op3;
 fc [3] <= op4;
 fc [4] <= op5;
 fc [5] <= op6;
 fc [6] <= op7;
 fc [7] <= op8;
 fc [8] <= op9;
 fc [9] <= op10;
 fc [10] <= op11;
 fc [11] <= op12;
end

delay_chain value1(
    .clk_out1(clk_out1),
    .clk_outx(clk_out1),
    .start(start),
    .delayed_output(op1)
);

delay_chain value2(
    .clk_out1(clk_out1),
    .clk_outx(clk_out2),
    .start(start),
    .delayed_output(op2)
);

delay_chain value3(
    .clk_out1(clk_out1),
    .clk_outx(clk_out3),
    .start(start),
    .delayed_output(op3)
);

```

```
68  delay_chain value4(
69      .clk_out1(clk_out1),
70      .clk_outx(clk_out4),
71      .start(start),
72      .delayed_output(op4)
73  );
74
75  delay_chain value5(
76      .clk_out1(clk_out1),
77      .clk_outx(clk_out5),
78      .start(start),
79      .delayed_output(op5)
80  );
81
82  delay_chain value6(
83      .clk_out1(clk_out1),
84      .clk_outx(clk_out6),
85      .start(start),
86      .delayed_output(op6)
87  );
88
89  delay_chain_n value7(
90      .clk_out1(clk_out1),
91      .clk_outx(clk_out7),
92      .start(start),
93      .delayed_output(op7)
94  );
95
96  delay_chain_n value8(
97      .clk_out1(clk_out1),
98      .clk_outx(clk_out8),
99      .start(start),
100     .delayed_output(op8)
101 );
102
103 delay_chain_n value9(
104     .clk_out1(clk_out1),
105     .clk_outx(clk_out9),
106     .start(start),
107     .delayed_output(op9)
108 );
109
110 delay_chain_n value10(
111     .clk_out1(clk_out1),
112     .clk_outx(clk_out10),
113     .start(start),
114     .delayed_output(op10)
115 );
116
117 delay_chain_n value11(
118     .clk_out1(clk_out1),
119     .clk_outx(clk_out11),
120     .start(start),
121     .delayed_output(op11)
122 );
123
124 delay_chain_n value12(
125     .clk_out1(clk_out1),
126     .clk_outx(clk_out12),
127     .start(start),
```

```
128     .delayed_output(op12)
129 );
130 
131 always @ (posedge clk_out1)
132 begin
133     if (start == 0) begin
134         flag = 0;
135     end
136     else flag = 1;
137     if(flag == 1) begin
138         if(fc != 12'b0) begin
139             state_flag <= 1'b1;
140         end
141         if(state_flag == 1'b0) begin
142             op_fc <= fc;
143         end
144         if(state_flag == 1'b1) begin
145             done = 1;
146         end
147     end
148     else begin
149         state_flag = 1'b0;
150         done  = 0;
151     end
152     fine_done = done;
153 end
154 
155 endmodule
```

Listing 5.2: Top Module of the Project for 12 bit phase shift

```
1  import csv
2  import serial
3  class receiver():
4      while(1):
5          delay_inp=input("Enter tap value:   ")
6          file=open('C:\\Users\\Downloads\\TIFR\\Results_FG\\FunGen_%s.
               csv' % delay_inp, 'w', newline='')
7          try:
8              data = serial.Serial(port = "COM6", baudrate=9600,bytesize
                   =8, timeout=3, stopbits=serial.STOPBITS_ONE)
9          except:
10             print ("Failed to connect")
11             exit()
12         writer = csv.writer(file)
13         writer.writerow(["Bits Received", "FC Binary Reading", "No. of
               1's", "checksum","counter","fc1","fc2","cc","fc1 value","
               fc2 value","cc value"])
14         fans_prev=0
15         counter_prev=0
16         counter=0
17         no_of_reading=0
18         while (no_of_reading!=1050):
19             line = data.readline(8)   # argument is maximum no. of
                   bytes to read
20             checksum= 0
21             counter = 0
22             fc1     = 0
23             fc2     = 0
24             cc      = 0
```

```
            fc1_val = float(0)
            fc2_val = float(0)
            cc_val  = 0
            list_16 = list(line)
            if(len(list_16)==8):
                list_16 = list(filter(lambda x: x != 0, list_16))
                print(list_16)
                reading_hex_str = line.hex()
                reading_bin_str = "{0:08b}".format(int(reading_hex_str
                    , 16))
                print(reading_bin_str)
                ones = reading_bin_str.count("1")
                checksum = reading_bin_str[0:8]
                counter = reading_bin_str[8:16]
                fc1 = reading_bin_str[25:32]
                fc2 = reading_bin_str[41:48]
                cc  = reading_bin_str[48:]
                fc1_val = float(fc1.count("1")) * float(476.19)
                fc2_val = float(fc2.count("1")) * float(476.19)
                cc_val = list_16[2]*256+list_16[1]
                writer.writerow([list_16,reading_bin_str,ones,checksum
                    ,counter,fc1,fc2,cc,fc1_val,fc2_val,cc_val])
                writer.writerow([list_16,fc1,fc2,cc,counter,fc1-fc2])
                no_of_reading = no_of_reading + 1
        file.close()
        data.close()
    exit()
```

Listing 5.3: Python Script for data analysis

5.3 Testing

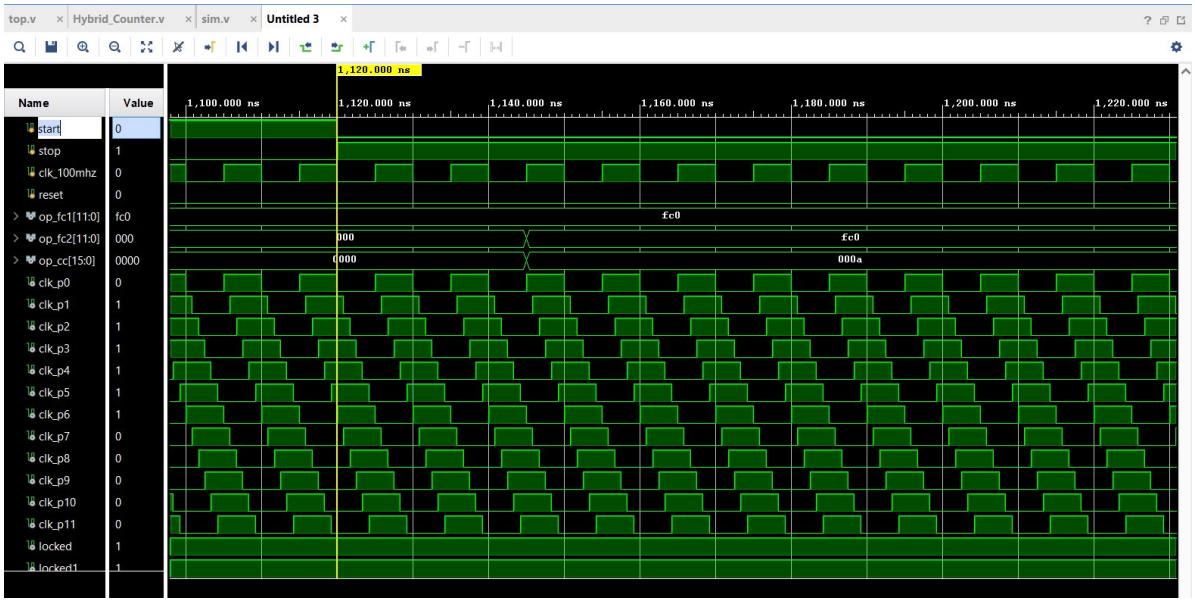

Figure 5.3: Timing Diagram of 12 phase shifted TDC

Figure 5.3 shows the timing diagram obtained when the above fine counter was simulated with sample signals. As shown, the fine counter latches at

every positive edge of clock if both clock and signal pulse are HIGH.

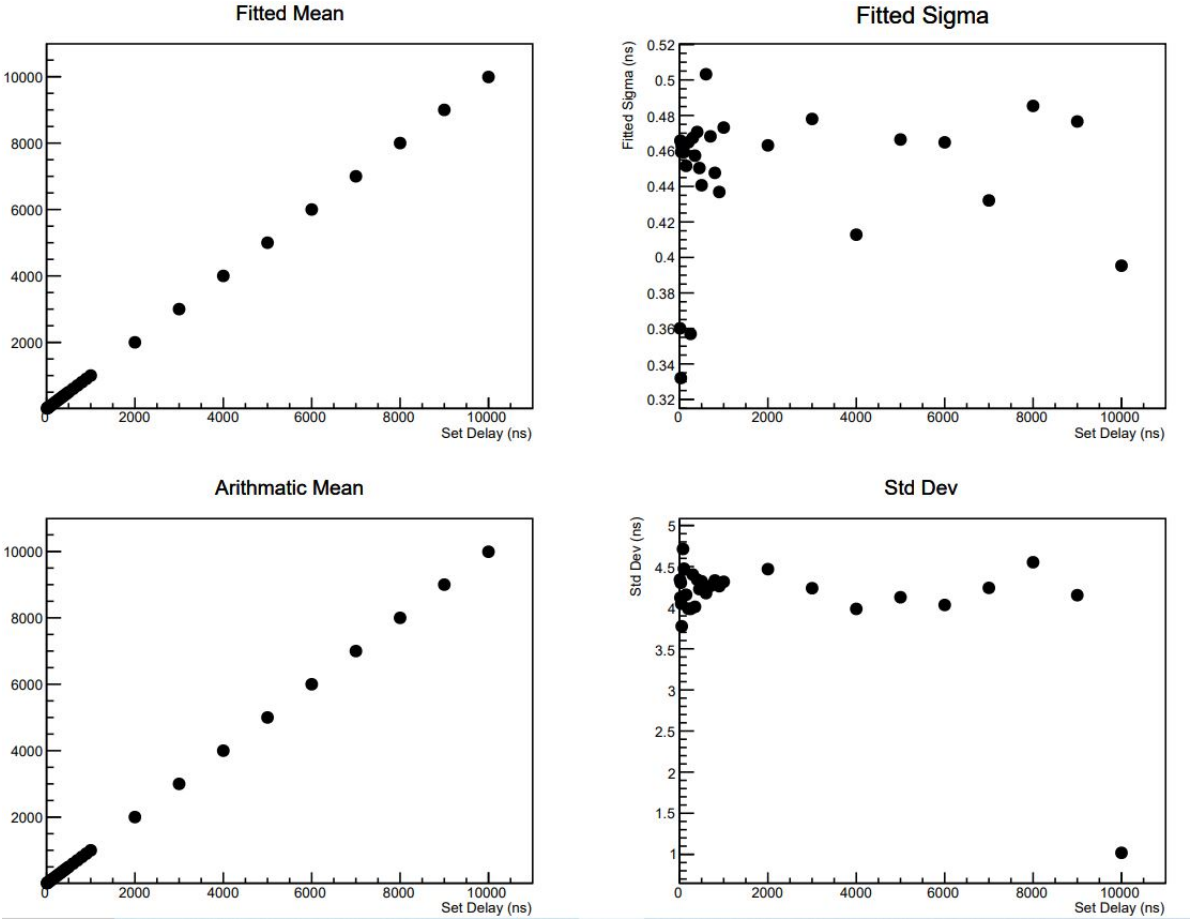

Figure 5.4: Testing of data received by 2Hz Frequency

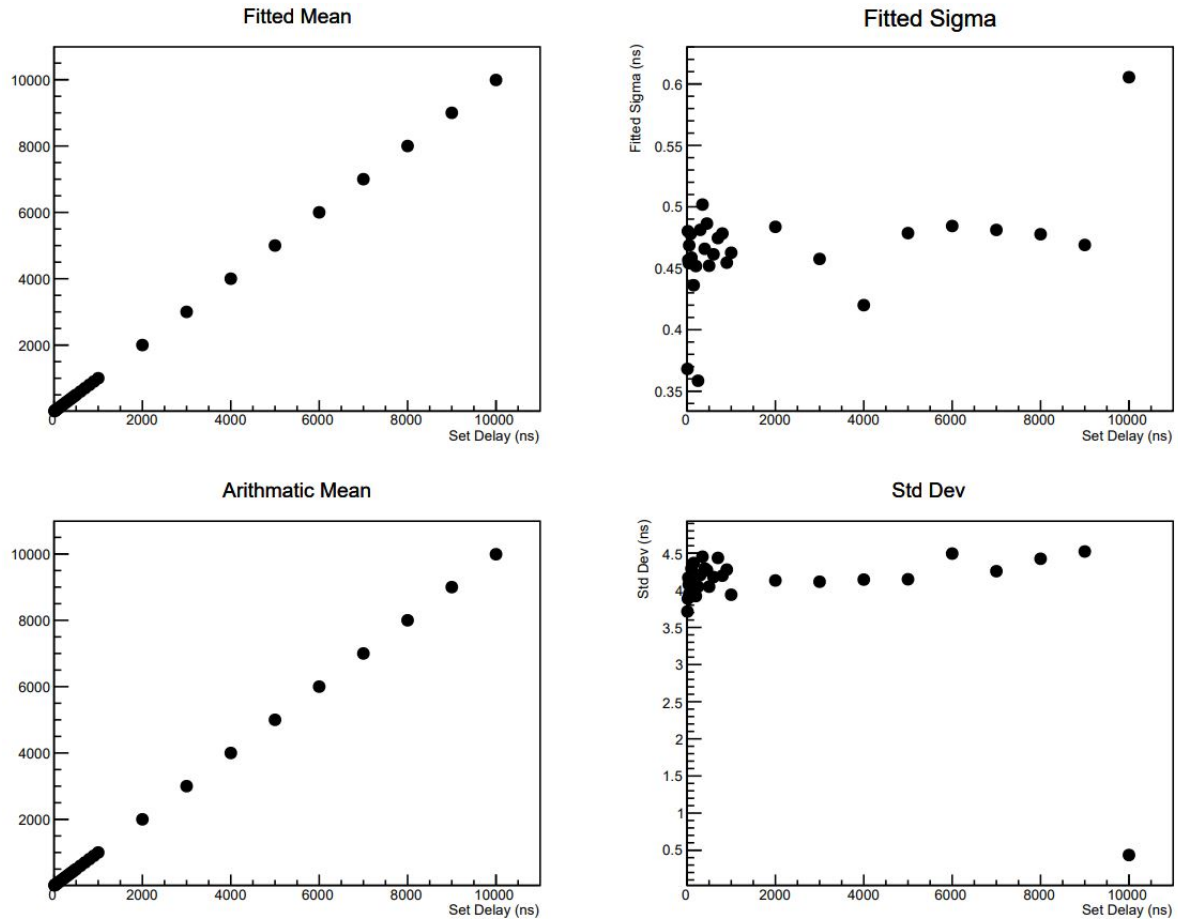

Figure 5.5: Testing of data received by 5Hz Frequency

Figure 5.4 and 5.5 gives the standard deviation of the data acquired by 2Hz and 5Hz frquency. As in the graphs we can see the values the standard deviation is minimum.

The acquired data by programming the given modules into the Nexys7 FPGA board. The data acquisition was done for two frequencies, 2Hz and 5Hz. Start and Stop pulses were generated for the said frequencies by a function generator with various delays between the start and stop. The delay range was 10ns, 20ns, 30ns, ...100ns, ...500ns, ...2000ns up to 10000ns. This was done for both frequencies independently. Data acquisition was done by the VIO IP core inside FPGA.

Figure 5.6 and 5.7 gives how the data is acquired at each delays, in the figures below the delay is shown for the 10ns, for both 2Hz and 5Hz frequency.

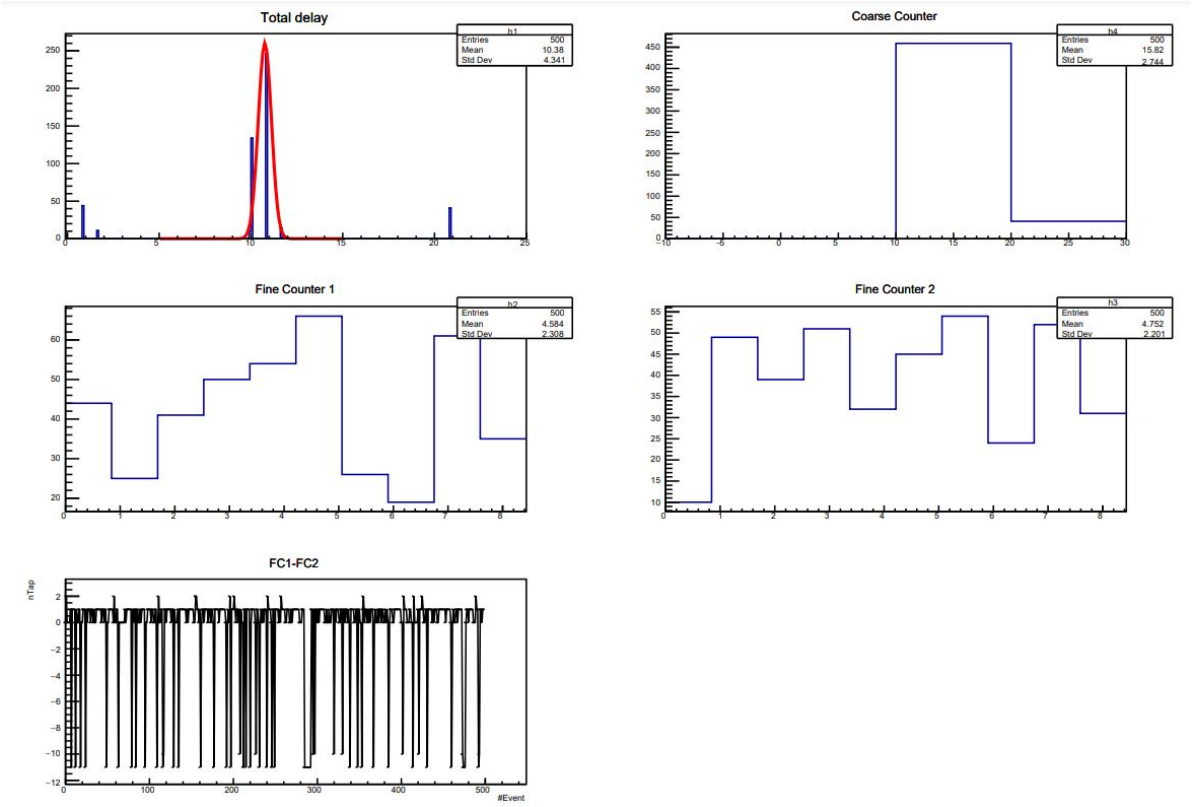

Figure 5.6: Data acquisation for 10ns of 2Hz frequency

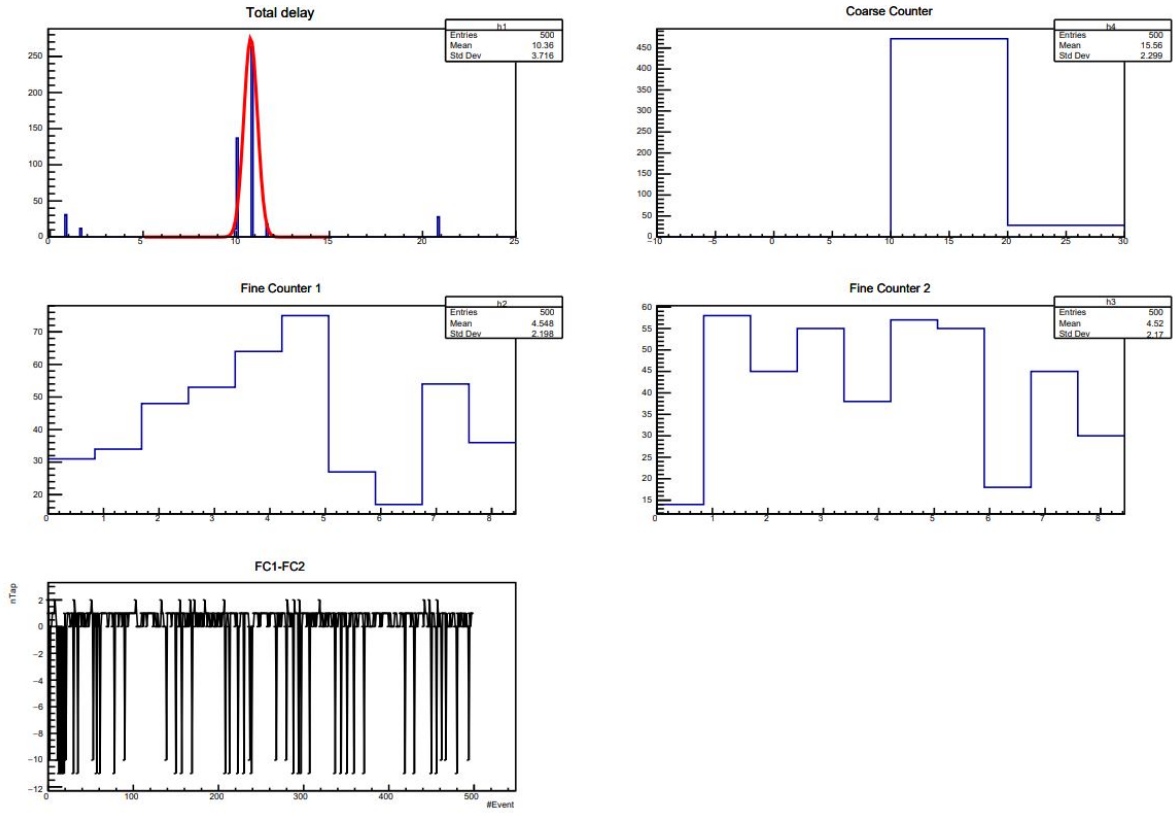

Figure 5.7: Data acquisation for 10ns of 5Hz frequency

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

In this report, the study of different techniques and devices used for implementing coarse counters, fine counters and time-to-digital converters has been presented. The advantages of FPGA-based TDCs are highlighted and explained.

6.2 Future Scope

In the future, this project can be integrated with TIFR's observatory at Ooty for the GRAPES-3 experiment. A similar but advanced design can also be used in the Compact Muon Solenoid (CMS) experiment built on the Large Hadron Collider (LHC) at CERN in Switzerland and France. The future scope includes implementation of this TDC in Light Detection and Ranging (LIDARs) that are used for advanced graphics or military applications.

References

[1] Y.-H. Kuo, *Determination of the angular distribution of cosmic rays at sea level*. PhD thesis, Massachusetts Institute of Technology, 2010.

[2] P. Lipari and S. Vernetto, "Diffuse galactic gamma-ray flux at very high energy," *Physical Review D*, vol. 98, no. 4, p. 043003, 2018.

[3] B. Hariharan, S. Ahmad, M. Chakraborty, A. Chandra, S. Dugad, S. Gupta, Y. Hayashi, H. Kojima, S. Inbanathan, P. Jagadeesan, *et al.*, "Energy sensitivity of the grapes-3 eas array for primary cosmic ray protons," *Experimental Astronomy*, vol. 50, no. 2, pp. 185–198, 2020.

[4] M. Zhang, H. Wang, and Y. Liu, "A 7.4 ps fpga-based tdc with a 1024-unit measurement matrix," *Sensors*, vol. 17, no. 4, p. 865, 2017.

[5] "Nexys a7 reference manual - digilent reference.." `https://digilent.com/reference/_media/reference/programmable-logic/nexys-a7/nexys-a7_rm.pdf`, July 2019.

[6] "7 series fpgas clocking resources - user guide.." `https://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf`, July 2018.

[7] "Xilinx inc. mixed-mode clock manager (mmcm) module.." `https://www.xilinx.com/support/documentation/ip_`, June 2009.

[8] Z. Soni, A. Patel, D. K. Panda, and A. B. Sarbadhikari, "Comparative study of delay line based time to digital converter using fpga," *International research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 9, pp. 1169–1175, 2017.

[9] A. Balla, M. M. Beretta, P. Ciambrone, M. Gatta, F. Gonnella, L. Iafolla, M. Mascolo, R. Messi, D. Moricciani, and D. Riondino, "The characterization and application of a low resource fpga-based time to digital converter," *Nuclear Instruments and Methods in*

Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 739, pp. 75–82, 2014.

[10] T. Sui, Z. Zhao, S. Xie, Q. Huang, J. Xu, and Q. Peng, "Pico-tdc: A novel fpga-based tdc with 2.2 ps rms timing resolution," in *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pp. 1–3, IEEE, 2017.

[11] C. Uğur, G. Korcyl, J. Michel, M. Penschuk, and M. Traxler, "264 channel tdc platform applying 65 channel high precision (7.2 psrms) fpga based tdcs," in *2013 IEEE Nordic-Mediterranean Workshop on Time-to-Digital Converters (NoMe TDC)*, pp. 1–5, IEEE, 2013.

[12] E. Bayer, P. Zipf, and M. Traxler, "A multichannel high-resolution (¡ 5 ps rms between two channels) time-to-digital converter (tdc) implemented in a field programmable gate array (fpga)," in *2011 IEEE Nuclear Science Symposium Conference Record*, pp. 876–879, IEEE, 2011.

[13] D. Chaberski, "Time-to-digital-converter based on multiple-tapped-delay-line," *Measurement*, vol. 89, pp. 87–96, 2016.

[14] Y.-Q. Chen, L.-Y. Meng, and X.-G. Lin, "A coarse-fine time-to-digital converter," in *ITM Web of Conferences*, vol. 11, p. 08006, EDP Sciences, 2017.

[15] W. U. I. M. N. D. J.-P. L. N. Foudil Dadouche, Timothé Turko, "New design-methodology of high-performance tdc on a low cost fpga targets," in *Sensors Transducers*, vol. 193, pp. 123–134, Sensors Transducers, 2015.

[16] D. J. Q. A. Jiajun Zheng, Ping Cao, "Low-cost fpga tdc with high resolution and density," in *IEEE Transactions on Nuclear Science*, vol. 64, IEEE, June 2017.

[17] Q. S. H. L. Q. A. Jinhong Wang, Shubin Liu, "A fully fledged tdc implemented in field-programmable-gate-arrays," in *IEEE-NPSS Real Time Conference*, vol. 16, IEEE, 2009.

[18] I. U. inyuan Wu (Fenni National Accelerator Laboratory, Batavia, "Several key issues on implementing delay line based tdcs using fpgas," in *IEEE Transactions on Nuclear Science*, vol. 57, IEEE, June 2010.

Acknowledgement

Success of a project like this involving high technical expertise, patience and massive support of guides, is possible when team members work together. We take this opportunity to express our gratitude to those who have been instrumental in the successful completion of this project. We would like to show our appreciation to **Dr. Jyoti More** for their tremendous support and help, without them this project would have reached nowhere. We would also like to thank our project coordinator **Mrs. Rakhi Kalantri** for providing us with regular inputs about documentation and project timeline. A big thanks to our HOD **Dr. Lata Ragha** for all the encouragement given to our team. We would also like to thank our principal, **Dr. S. M. Khot**, and our college, **Fr. C. Rodrigues Institute of Technology, Vashi**, for giving us the opportunity and the environment to learn and grow.

Project Group Members:

1. Harshul Raina, 101851

_____

2. Priyanka Sankannawar , 101854

_____

3. Vishadh Sawant, 101855

_____

Appendix A : Timeline Chart

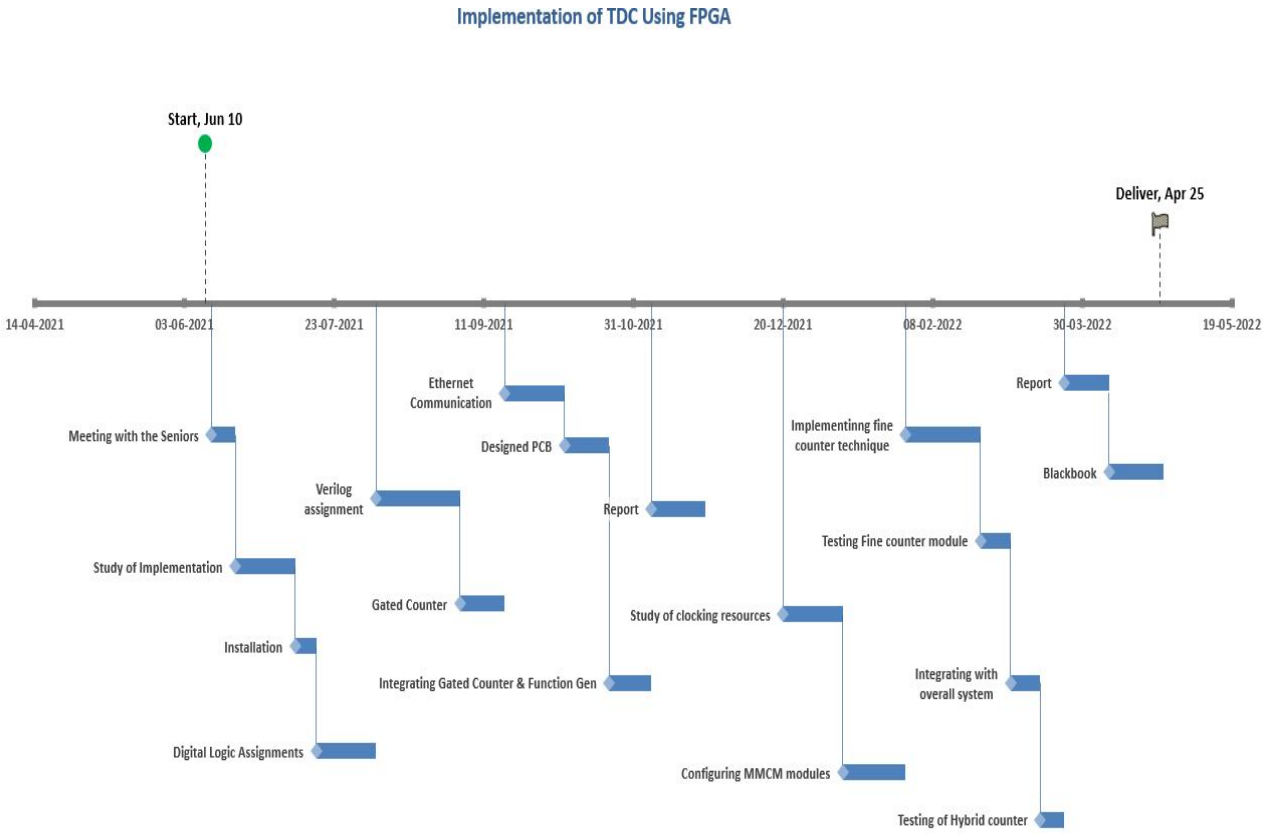

Figure 6.1: Timeline Chart