Measuring the lifetime of cosmic ray Muons through an FPGA based Time to Digital Converter

A thesis submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Electrical Engineering

by

Sultan Abdul Wadood
DE-33-EE

Department of Electrical Engineering,
College of Electrical and Mechanical Engineering,
National University of Science and Technology

Dedicated to

The loving memory of Uncle Hamid

Acknowledgements

There are a lot of people to whom I owe the completion of this project.
I received constant support from my parents, who a managed and planned the whole project and the constant travels to Lahore. Their interest and exhortation kept me on track throughout the year. All the work in Lahore would not have been possible without the help, support and services offered by my Late Uncle Hamid, Auntie Ghazala and my Grandmother. They provided me a home away from home. Uncle Hamid passed away during the course of the project. He provided me fresh respite from work every night through his jokes, life stories and experiences. May God grant him highest ranks in Jannah.
My peers always pulled me up whenever I needed help. I am especially grateful to Ahtisham for always providing a listening ear and valuable feedback to my ideas and problems regarding the project. Huzaifa Sajids and Junaid Alams help was invaluable regarding the algorithm development. Uzair Latifs patience with me regarding the lifetime experiment accelerated my project. Similarly Asif Ali, Farjad Hassan, Umar Nisar and many other friends always cooperated with me and I am grateful to them.
I could not thank my teachers enough. Dr. Hamid's constant guidance and experience regarding Digital System Design helped me on several occasions. Sir Sajid's lectures on Digital Design gave me a solid theoretical base. I am indebted to both of them.
My supervisor Dr.Mashhoods vision and advices were a great help in streamlining the project. He has been a great mentor throughout my undergraduate studies and I am grateful to him.
Perhaps the biggest contribution to this project was from Dr. Sabieh Anwar. From the concept to the implementation, he helped me form the big picture, smooth the little details and motivated me to go the last mile. The writing of this thesis has been an exercise in style and exposition on a new level for me. The environment and financial support provided at Physlab were top-notch. I hope experimental Physics thrives in Pakistan under his leadership. It has been an honor working with him.
As is the case with every journey, there are many unsung heroes who helped me on each step of the project. God bless them.
I thank God for giving me the strength to complete this project, and for providing such good-natured souls in my life.

Abstract

Measuring the lifetime of Cosmic Ray Muons using an FPGA based Time to Digital Converter

The lifetime of a cosmic ray muon is conventionally measured using a time to amplitude converter (TAC). But modern FPGAs can easily measure sub-nano second time intervals. We use a time to digital converter (TDC) based on the 4× oversampling technique to measure the lifetime. The TDC is implemented on a Virtex-5 FPGA. We first present an overview of the techniques for high resolution time measurements available in the FPGA and then present, in the second chapter, a detailed description of the 4× oversampling TDC. The last chapter explains the lifetime experiment in detail the lifetime experiment implemented and the use of TDC in it. The lifetime estimated is $2.167 \pm 0.15 \mu s$.

Contents

1 Introduction **1**
- 1.1 Digital Measurement of Time . . . 1
- 1.2 Muon Lifetime Measurement . . . 3
- 1.3 Organization of the Thesis . . . 3

2 Ultra-Fast time interval measurement schemes: A review **4**
- 2.1 Coarse Counting . . . 4
- 2.2 Fine Measurement Methods . . . 7
  - 2.2.1 Analogue Method: Charging a Capacitor . . . 7
  - 2.2.2 Digital Methods . . . 7
  - 2.2.3 Nutt Interpolation Method . . . 11
- 2.3 Errors in Time Measurement . . . 11
  - 2.3.1 Sources of Error . . . 11
  - 2.3.2 Non-Linearity and Jitter . . . 12

3 The 4×Oversampling Time to Digital Converter **15**
- 3.1 General Design . . . 15
- 3.2 Design of the Interpolator . . . 16
  - 3.2.1 Metastability, Setup and Hold Times . . . 16
  - 3.2.2 Encoding the Fine Count . . . 18
  - 3.2.3 Design of Counter . . . 20
  - 3.2.4 FIFO Buffer . . . 20
  - 3.2.5 Digital Clock Manager . . . 20
- 3.3 Performance Testing . . . 21

4 Measuring the Lifetime of Cosmic Ray Muons **28**
- 4.1 Introduction to cosmic ray muons . . . 28
- 4.2 The Lifetime Experiment . . . 29
  - 4.2.1 Scintillators . . . 30
  - 4.2.2 Photomultiplier Tubes (PMT's) . . . 30
  - 4.2.3 Discriminators . . . 32
  - 4.2.4 Logic Units . . . 32
  - 4.2.5 The Experimental Setup . . . 33
  - 4.2.6 Method and Results . . . 33
  - 4.2.7 Data Acquisition . . . 34
  - 4.2.8 Curve Fitting . . . 34
- 4.3 Implementing the Logic Unit in the FPGA . . . 38
  - 4.3.1 Algorithm for implementing the logic unit . . . 38
  - 4.3.2 Modification to the algorithm . . . 40
  - 4.3.3 Results of the Logic Unit implementation on the FPGA . . . 42
- 4.4 Sources of Noise and Errors . . . 43

4.5 Conclusion 43

Chapter 1

Introduction

Before Galileo invented the pendulum clock, the notion of time measurement was not well defined. Instruments such as sundials, sand glasses, and incense clocks provided a relatively crude sense of time. The motion of heavenly bodies and the variation of seasons provided one basis for most of the time measurement. Mammalian bodies also function on a Biological clock, which responds to the change in the environment's light intensity in a 24-Hour period [1]. The pendulum clock, however, offered a characteristic, reproducible time period with which to conduct measurements. For three centuries, it remained the standard time measuring instrument, both for the scientist and the laity. With the advent of twentieth century came the electric clock, based on the oscillations of a quartz crystal. Further precision was achieved by using atomic transition frequencies, resulting in the ultra-precise atomic clocks used in satellite navigation systems today[2].
The ability to measure such small intervals of time has revealed new, intricate phenomenon in the atomic realm. Among these processes is the atomic particles' lifetime, which can range from picoseconds to several seconds. This project focuses on digitally measuring the lifetime of cosmic ray muons which are charged particles created from the interaction of the cosmic rays with the Earth's atmosphere. The lifetime of a muon is on the order of microseconds. But this is a statistical parameter, and involves measuring intervals ranging from a nanosecons to tens of microseconds.
There are two phases of this project:

1. design of a digital sub-nano second time measuring instrument,and
2. measuring the lifetime of a cosmic ray muon using this instrument

1.1 Digital Measurement of Time

Explicit in the above primer of the history of clocks is the concept of periodicity. There is an underlying periodic process-a clock- whose periods are counted to measure time. The time interval thus measured is just an integral multiple of this period. As mentioned before, these clocks can be biological, electronic, atomic, or even chemical[3].
A clock is the heartbeat of a modern digital system, and drives all its synchronous logic. The speed of a digital system is also determined by the maximum time required for any process to complete between two synchronous elements. In a digital system, the clock is mostly electronic and derived from a crystal oscillator.
The time intervals measured by a digital counter/timer have a lower limit of the time period of the system clock. Thus the obvious method to increase the resolution of a digital timer is to increase the clock frequency, conceptually shown in Fig. 1.1. But the bottle neck in increasing the frequency is the rise and fall time of the transistor used in the system. As Fig. 2 shows,

an input pulse is delayed and skewed at the output.

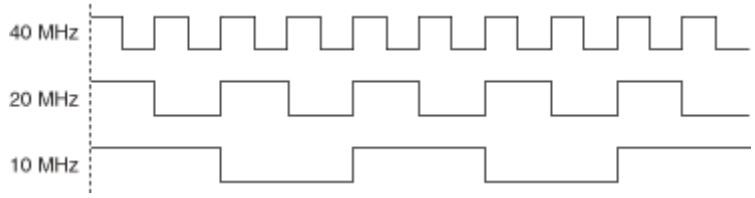


Figure 1.1: As clock frequencies increase, so does the timing resolution

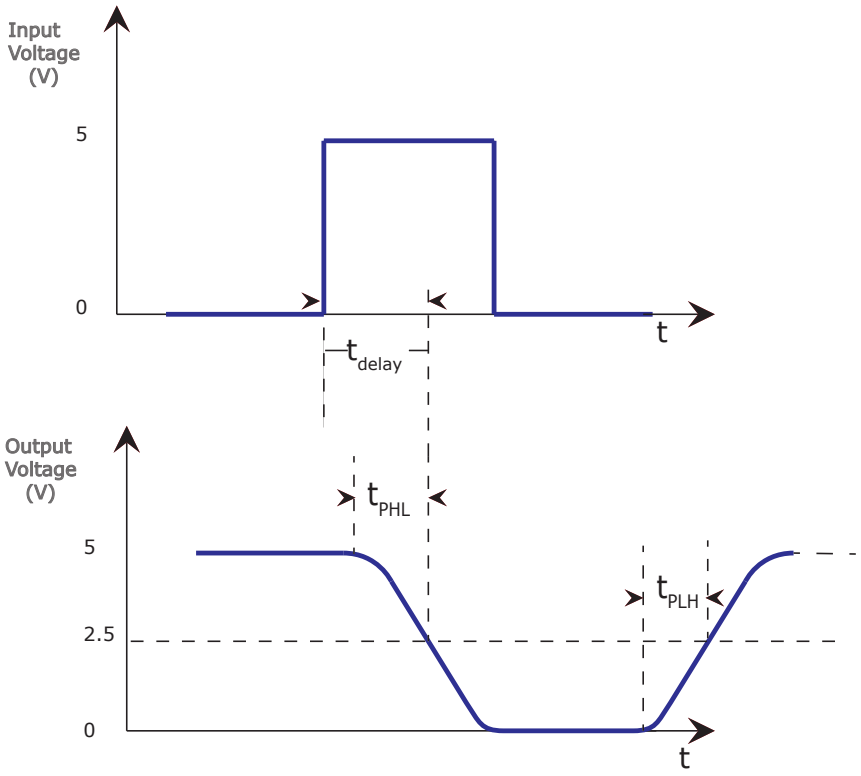


Figure 1.2: (a)An ideal Input pulse (b) Output pulse, delayed and edges skewed

The maximum frequency at which a digital system can operate is given by

$$f_{max} = \frac{1}{2(t_{PLH} + t_{PHL})} \tag{1.1}$$

where t_{PLH} denotes the transition time from low to high and t_{PHL} denotes the transition time from high to low. These transition times depend on the capacitances of the transistor, the traces/wires, and the input capacitances of all the logic gates driven by the transistor[4].
As we tread the trajectory of Moore's law, transistors shrink in size and so do the associated capacitances and delays. With every passing decade, digital systems are running on faster clocks.

But increasing speed comes at the cost of complex circuit designs. For example, signals faster than 300 MHz have a wavelength of less than one meter and RF PCB design techniques have to be observed to avoid interference and for maintaining signal integrity. These effects are however negligible in modern day circuits whose dimensions are on the scale of micro-meters.
The bottleneck, therefore, is the transistor technology used. Modern application specific integrated circuits (ASIC) reach speeds in giga hertz, while field programmable gate arrays (FPGA) are also knocking on the sub-nanosecond domain. We use Xilinx's LX50T Virtex-5 FPGA chip, and an oversampling technique, explained in the third chapter, to measure time intervals with an accuracy of 833ps.

1.2 Muon Lifetime Measurement

Conventional particle physics experiments are very expensive to setup. Time measuring modules are mostly analog time to amplitude converters(TAC), which provide timing resolution down to tens of picoseconds. The cost of such experiments kept them from being translated to the benchtop aimed at investigating the properties of cosmic ray muons, such as lifetime and speed distribution. Note that measuring muon lifetime is an important aspect of verifying Einstein's principle of special relativity. An key facet of this project, therefore, is to provide a low-cost digital, reconfigurable alternative to the advanced physics student to conduct lifetime and coincidence experiments. Naturally, this scheme can also be extended to scientific investigations of ultra-fast phenomenon such as single-photon counting in quantum optics and quantum communication [5, 6], fluorescence lifetime imaging [7], laser and plasma spectroscopy [8].
We use scintillators to capture and register Muons. High speed logic, coupled with the FPGA based time to digital converter (TDC) , is used to measure the lifetime of the muon. The experiment is described in detail in Chapter 4.

1.3 Organization of the Thesis

1. Chapter 2 gives an overview of the techniques for measuring picosecond time intervals digitally. An overview of the associated non-linearities and noise sources is also presented.

2. Chapter 3 describes the Oversampling TDC implemented on the Virtex-5 FPGA in a top-down fashion. Prototypical results are also presented, which helps us in characterizing the implementation against benchmarks.

3. Chapter 4 describes the muon lifetime experiment. The use of the TDC is emphasized in the measurement process. Finally, measured results are also presented and uncertainties are calculated.

Chapter 2

Ultra-Fast time interval measurement schemes: A review

As explained in the previous chapter, we are interested in measuring a sub-nanosecond time interval using the FPGA. There is no single method to go about this. This chapter examines in detail various digital techniques to measure time intervals down to picoseconds. In writing this chapter, I have benefitted greatly from J. Kalisz's comprehensive paper[9].

Generally speaking, high resolution time measurements can be of two types:

1. coarse count type, in which the period of the clock determines the timing resolution, and
2. fine count type, which makes use of interpolation methods.

The interpolation methods discussed in the chapter are:

1. Vernier delay method, which can be considered as the temporal analog of the Spatial Vernier measurement, and the
2. oversampling method, which is actually our implementation of the TDC.

In this chapter, we review the coarse and fine measurement schemes. These measurements, however, are also prone to electronic noise and errors. Therefore, an explanation of the noise sources and the non-linearity associated with digital counters concludes the chapter.

2.1 Coarse Counting

The simplest method to measure the time interval is to literally count the number of periods of a reference clock. The resolution of this method is $T_o = 1/f$.

As the incoming data is asynchronous, there is a uniform probability distribution of the start and stop edges being placed anywhere in the time period. The maximum quantization error may reach $\pm T_o$. When measuring a series of a constant and asynchronous intervals T, one obtains two possible outcomes, $T_1 < T$ and $T_2 = T_1 + T_0 > T$. The probability of each reading depends on the fractional part c of the fraction T/T_o.

$$p = p(T_1) = 1 - c, and \tag{2.1}$$

$$q = p(T_2) = c \tag{2.2}$$

The measured time interval is $pT_1 + qT_2$. The random uncertainty is the standard deviation of the binomial distribution and is given by $= T_o\sqrt{c(1-c)}$ as shown in Fig.2.1 and Fig.2.2

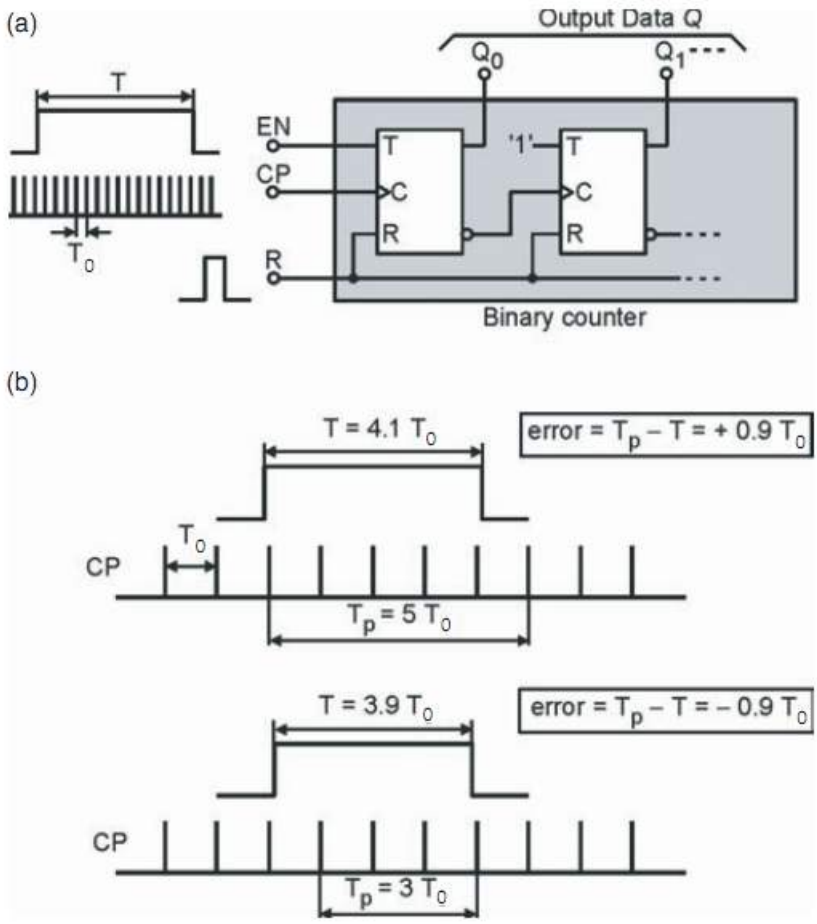


Figure 2.1: (a)A binary counter (b) possible maximum errors in the measurement.Figure is taken from [9]

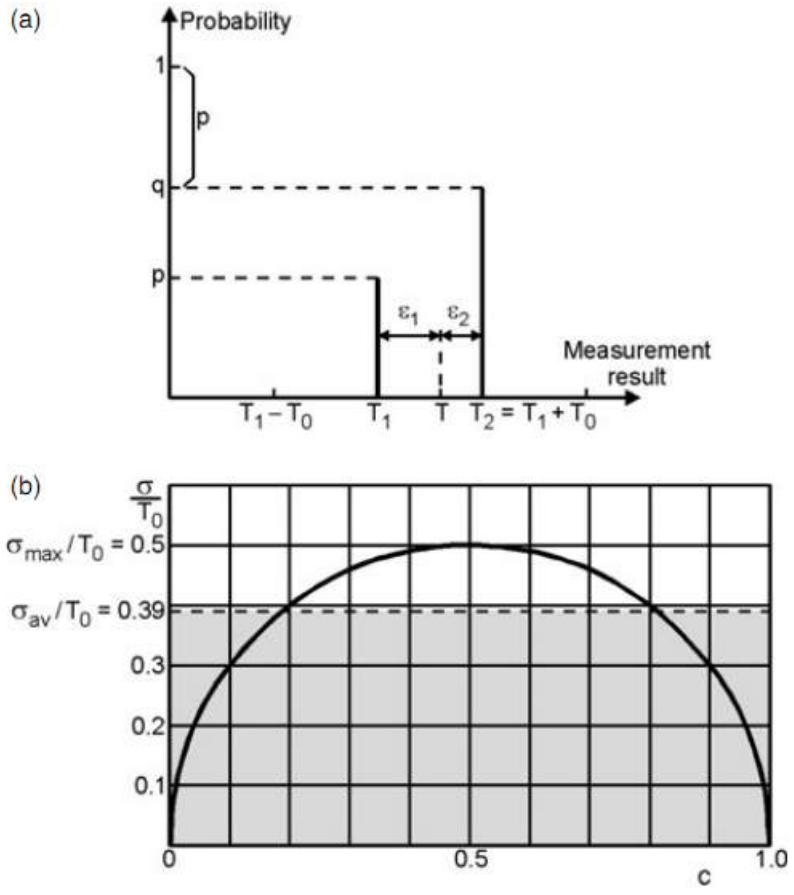


Figure 2.2: (a)The probability distribution of positive and negative errors (b) standard deviation of measurements shown as a function of c, the fractional part of the quotient T/T_o.Figure is taken from [9]

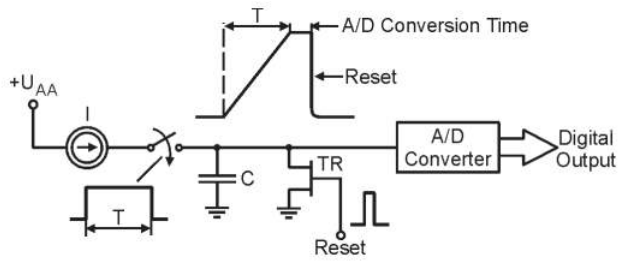


Figure 2.3: Conversion of time to voltage followed by A/D conversion. Figure taken from [9]

For a higher time resolution, one needs a clock with smaller time periods. To achieve the picosecond range, this means that frequencies above 1 GHz are required. Besides being expensive, implementing stable clocks of such frequencies on an embedded system such as a micro-controller, FPGA or even on an ASIC requires careful PCB designing employing RF techniques. One also needs the transistor technology that functions with the required rise and fall times, as discussed in the Chapter 1. All these constraints force us to look for alternative ways as opposed to using crystals oscillators of high frequencies. The next section discusses these methods.

2.2 Fine Measurement Methods

These methods provide the higher time resolution we are seeking. The analog method has been the conventional method and is still employed in many experiments. However, our focus will be the digital methods.

2.2.1 Analogue Method: Charging a Capacitor

The voltage on any capacitor charged by a constant current for time duration δt is directly proportional to δt. The A/D conversion time is called the dead time of the device. For accurate measurements, one needs a very precise current source, and a high resolution, fast A/D converter, thus the high cost of such instruments. Timer intervals as low as 20 ps have been measured with this method.

2.2.2 Digital Methods

Vernier Delay Line Technique

This technique is similar to length measurement on a Vernier scale.The configuration is shown in Fig. 2.4 There are two clocks with frequencies differing only slightly. The time resolution obtainable is $T = \frac{1}{f_1} - \frac{1}{f_2}$.
The Start and Stop pulses start the Signal Generators $SG1$ and $SG2$ which generate the clocks. These clocks drive counters. When the two clocks, of frequencies f_1 and f_2, coincide, as shown in Fig.2.4, the coincidence circuit resets the circuit and stops the counters[9].The time interval measured is given by

$$T = (n_1 - 1)T_1(n_1 - 1)T_2 \tag{2.3}$$

where n_1 and n_2 are the respective counts of counter 1 and counter 2 A much simpler method is based on the tapped delay line [10]. The Start signal propagates through the delay line, shown in Fig.2.5, with each element providing a delay of τ. The stop signal samples the start signal at the flip flops. The outputs of the flip flops provide a code, which are all consecutive ones

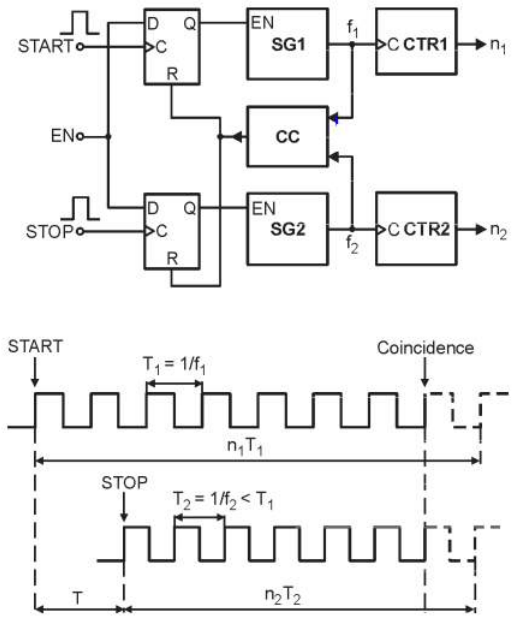


Figure 2.4: TDC Based on the Vernier Method: (a) Circuit Example (b) Example of conversion process when EN is asserted. Figure taken from [9]

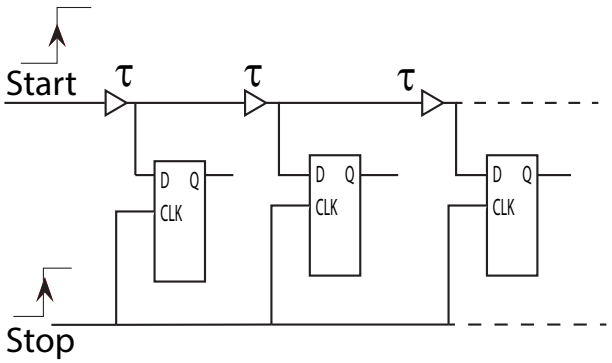


Figure 2.5: Multi Tapped Delay Line: Each output of the delay line is sampled by one flip-flop. This provides a time resolution of τ

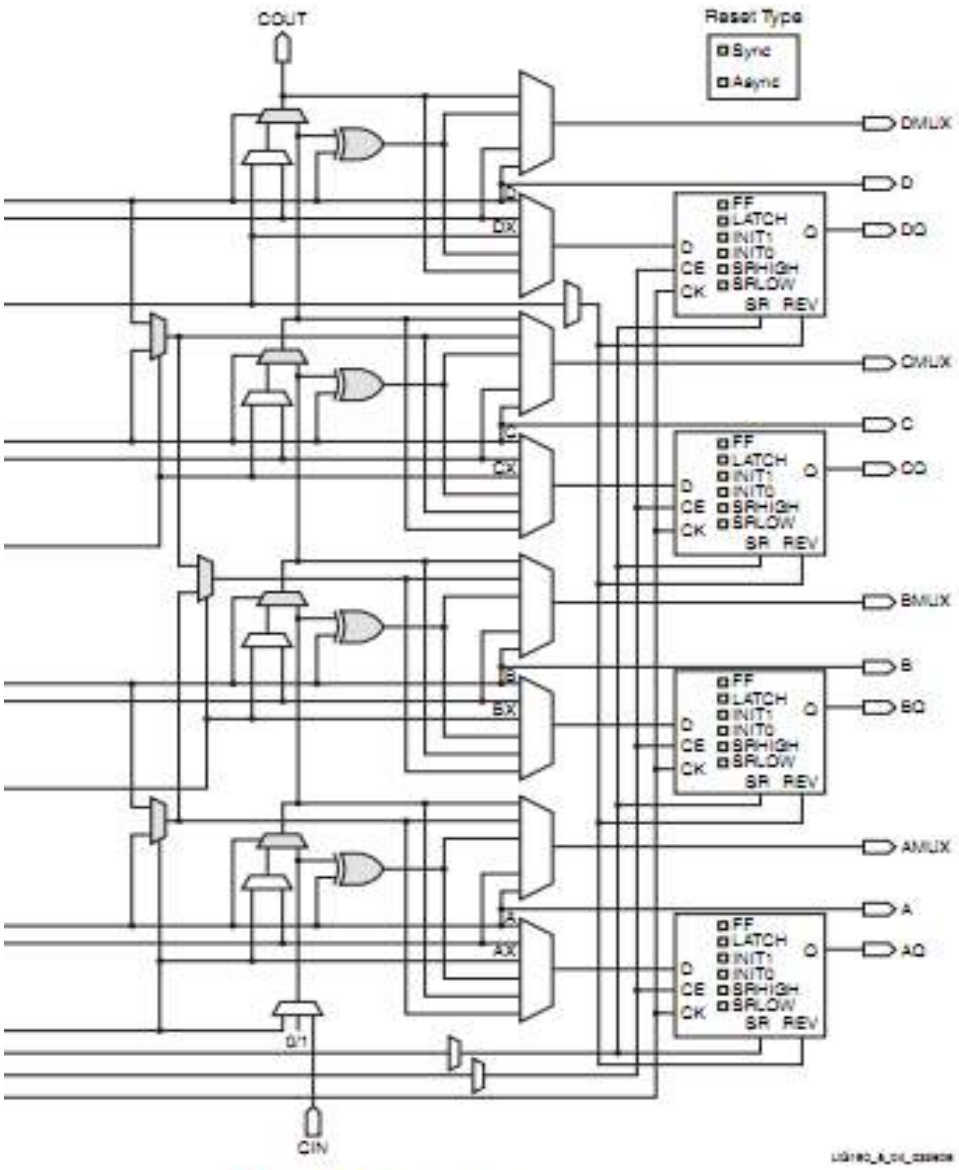


Figure 2.6: A Complex Logic Block of Virtex 5 FPGA. The Mux and XOR gate form the carry-Look ahead chain used as the delay line. Figure taken from 2.6

till the flip flop where the Start signal had reached before being sampled by the stop signal. By detecting the position of the 1 to 0 transition in this code, we can determine required time interval. The delay line is implemented in the Virtex-5 FPGA using the Carry Chain in the Configurable Logic Block, as shown in Fig.2.6. TDCs of 17ps resolutions have been implemented using this scheme [11].In another example, the τ is approximately 20ps, which has been voltage and temperature compensated [12].

There exists yet another method in which two delay lines with slightly different delays are used, shown in Fig.2.7. It is conceptually similar to the method of two oscillators with nearly equal frequencies. The first delay line comprises of latches each having delay τ_1 and are in a transparent mode. The second delay line comprises of non-inverting buffers each having delay $\tau_2 < \tau_1$. When the Start signal arrives, it propagates through the first delay line. The stop signal propagates through the second delay line. As the stop signal propagates faster, it catches up with the Start signal. The output of the latch at which the two signals coincide is set high. All previous latches are reset through a feedback signal and we are left with a single 1 in the whole series of latches. The position of 1 indicates the time difference between the Start and

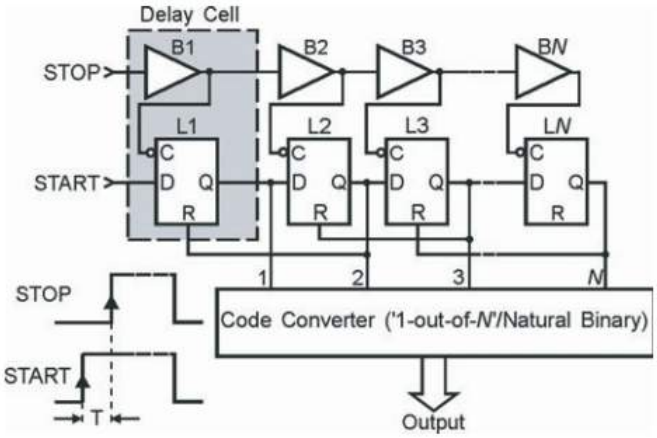


Figure 2.7: A Differential Tapped Delay Line. Figure taken from [9]

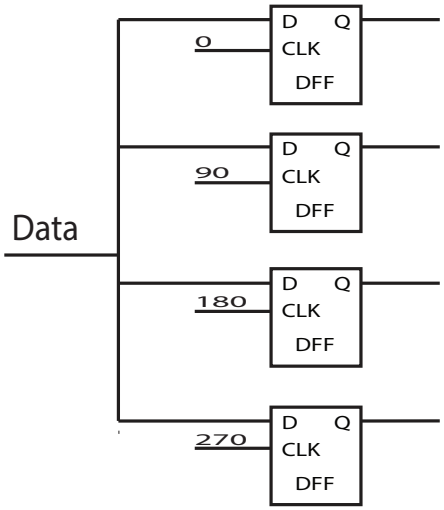


Figure 2.8: Sampling on four phase shifted clocks

Stop signal.

Oversampling

In this method, we increase the sampling frequency of the asynchronous data. Using phase-shifted versions of the clock to sample the incoming data, the resolution corresponding to the smallest phase shift induced can be achieved[10]. In Fig.2.8, the timing resolution has be increased by four times using four quadrature shifted phases of the clock. This is the scheme that we've implemented in our FPGA implementation.

We have discussed the Coarse and Fine count measurement methods. The former lacks high resolution but is simpler. The latter is complex to implement and has a short measurement range. The NUTT interpolation method, that is discussed in the following section, uses both of the methods to achieve a high resolution measurement over, theoretically, an infinite range of time.

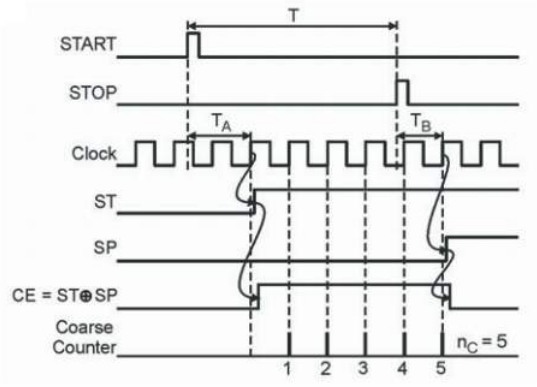


Figure 2.9: The Nutt Interpolation Method. Figure taken from [9]

2.2.3 Nutt Interpolation Method

This method combines the coarse counter measurement with a finer measurement implemented with the methods stated above[9]. As shown in Fig.2.9, the measured time interval is divided into three parts. Two of these are finer intervals which will be interpolated, while one is measured with the coarse counter. The measured time interval T is the sum of these three intervals, and is given as

$$T = n_c T_o + T_A - T_B \tag{2.4}$$

Here T_A is the time between the START signal and the next leading clock edge, T_B is the time between the STOP signal and the next leading clock edge, and T_o is the clock time period. n_c is the coarse count. Time intervals T_A and T_B are measured with the finer TDCs discussed previously.

No electric circuit is immune to noise, whether it is due to parasitic coupling, line interference, or the quantum noise inherent in any electrical quantity. Similarly, no two transistors are the same. There are always tolerances in the parameter values. In our measurement of time, therefore, we should trace the effect of noise on our measurement. This will enable us to characterize our system.

2.3 Errors in Time Measurement

Noise does not differentiate between the analog and digital domains. Furthermore, when measuring time intervals at such high resolution, the slightest variation in any parameter causes a drift in the reading. Thus understanding noise leads to a better appreciation of the working, limits and quantitative assessment of the measurement system.

2.3.1 Sources of Error

There are three main sources of error contributing to the standard uncertainty in any time measurement[9].

1. Quantization Error
2. Non-Linearity of the Interpolators

3. Jitter

Quantization Error

Whenever an analog value is digitized, there is loss of information. Furthermore, the START and STOP pulses are asynchronous, i.e., not related to the system clock; where they arrive respective to the clock edge determines the time interval measured. As described in section 2.1, there is a uniform probability distribution of the START and STOP signal falling at any time relative to the active clock edge.
If we consider all other noise sources suppressed, and consider only the effect of quantization error, the precision of the TDC can be evaluated as the function of the ratio $c = \frac{T}{T_o}$ by means of the expression

$$\sigma = T_o\sqrt{c(1-c)} \tag{2.5}$$

This is generic expression for the standard deviation of any binomial distribution with two complementary parameters, for example, $candq = 1-c$, as shown in Fig.2.2. The average value σ_{av} of the precision $\sigma(c)$ can be evaluated by integrating in the interval $0 < c < 1$:

$$\sigma_{av} = \frac{\pi T_o}{8} \tag{2.6}$$

When measured for an actual TDC, the data might show deviation from this graph, because of other noise sources i.e. jitter, skew etc. that we've assumed suppressed so far in our assumption.

2.3.2 Non-Linearity and Jitter

Any non-linearity in the measurement is due to the noise inherent in the circuit components and the noise induced by the environment. A jitter is variation in the signal period and is inherent in the crystal oscillator. This jitter is translated throughout the device.

Another big contribution to error is due to the net skew. The nets carrying the clock signal are of variable length, and thus have variable delay. Therefore, the clock edge does not arrive at each flip-flop simultaneously.

The delay of each flip-flip is also not identical and varies with the voltage and temperature.

Finally, the non-linearity of the interpolators is due to the skew of the clock signals, the skew of the input signals, the clock jitter, temperature and voltage dependence of the delay line, and variation in the flip-flop and transistor characteristics inside the device. Changes in temperature affect propagation speeds along the delay line and also along normal traces. The physical dimensions also cause non-linearity. In the tapped delay design, for example, the interconnect between elements of two different slices will have a larger delay than the connections inside a slice [11].
Good design practices, smart routing, which might often involve manual routing of the FPGA, help to reduce these non-linearities.
We characterize the non-linearity, or the bit error of a TDC by its differential non-linearity.

Differential Non-Linearity (DNL) of the TDC

The differential non-linearity (DNL) is the difference between an actual step width and that of 1 LSB. Consider Fig. 2.10 The x-axis is the analog time interval we intend to measure. The

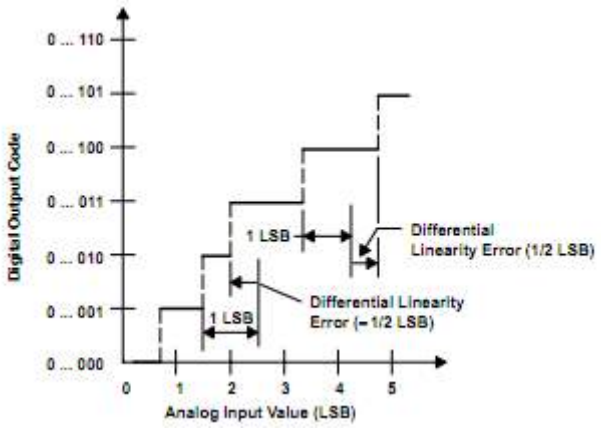


Figure 2.10: Differential non-linearity error. Note the positive and negative bin width errors. Figure taken from [13].

Y-axis shows the output binary codes. Ideally, the width of each code should be 1 LSB, but due to the inherent non-linearities discussed above this does not hold true in practice. The figure shows two errors: a negative one for a shorter bin width, and a positive one for a larger bin width[13].
To measure the DNL, we use a statistical code density test. The idea is to generate a large number N of randomly distributed pulses, and to make a histogram of all the results. Ideally the histogram should be a straight line on the Y-axis, with each bin registering, on average, the same number of counts. But each bin will have a DNL error, and thus its average will vary from the desired ideal value. If the TDC has Q bins, the mean number for each bin in the histogram should be $n' = N/Q$ [11].In practice, $n_c \neq n'$. The DNL for Cth bin will be : $DNL_c = (n_c/n) - 1$.
The cumulative sum of the DNL yields the Integral non linearity (INL)

$$INL_c = \sum_{i=1}^{c} DNL_i \tag{2.7}$$

Instead of generating truly random sequences, we can also input linearly increasing sawtooth like time intervals to a TDC and histogram the results [14]. This is analogous to measuring the DNL of an ADC. To generate such time intervals, such that each successive interval is a linear increment of the previous interval, we use two periodic pulses. The pulse are not harmonically related to each other. One is the START and the other is STOP signal. The START and STOP pulses slowly scan past each other. Output codes are generated as if the TDC samples a continuously varying sawtooth of frequency f_{sweep} at a sampling frequency f_{start}, where $f_{start} = f_o - f_{sweep}$ and f_o is the system clock. The input time intervals are evenly distributed in the range of $0 - T_o$, where T_o is the time period of the system clock. Any noise, skew and jitter further randomize the inputs. By plotting the histogram of the output codes, the DNL can be computed. We will measure the DNL of our system. This is described in Chapter 4.
All of the noise sources mentioned above are reflected in the DNL graph, although the individual contribution by each source may not be known. Overall, the DNL graph is a fair measure of the TDC's performance.
From all of the above methods, we use the Nutt Interpolation method with Oversampling technique to implement on our FPGA. The reasons for using the Oversampling technique are:

1. Easy implementation

2. Long range of measurement
3. Low device utilization
4. Low DNL [10]

The next chapter discusses in detail the oversampling TDC implemented on Virtex-5.

Chapter 3

The 4×Oversampling Time to Digital Converter

The TDC is essentially a high speed digital counter that uses an interpolation technique. As discussed in Chapter 2, the interpolation is actually of the over-sampling type. This chapter describes the TDC as a digital system. It discusses the TDCs architecture in a modular fashion. In particular, the oversampling flip-flops are discussed in detail, as they form the core of the design. To gauge its performance, the TDC was tested with high-speed clocks. The chapter concludes with these prototyping results.

3.1 General Design

Following is the overall architecture of the working system, also shown in Fig. 3.1:

1. Asynchronous Data: comprises of the START and STOP Signals whose time difference we want to measure
2. Flip-Flops: Comprises of the interpolater/overSampler and the digital counter
3. FIFO Buffer: Stores the time-stamps of the START and STOP Signals.
4. Serial Port and MATLAB: Data from the FIFO buffer is transferred serially to a MATLAB based data acquisition system. The baud rate is 128000 bits per second. We use FTDI's chip based serial to USB converter cable. A serial object in MATLAB is initiated which handles all the data transfers.

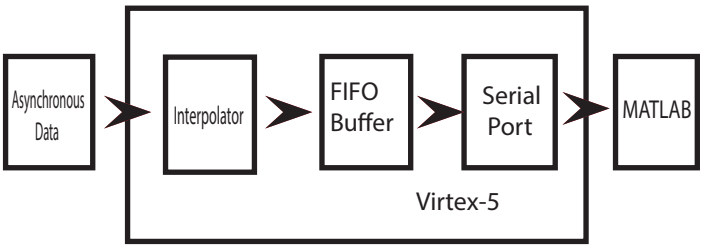


Figure 3.1: The Top-level Architecture

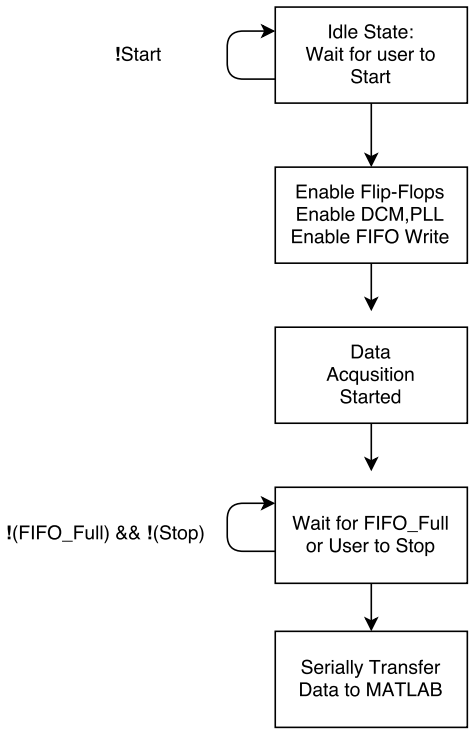


Figure 3.2: The algorithmic state machine of the design

Fig. 3.2 shows the algorithmic state machine of the whole design: When the user presses the Start button, the data acquisition is started and is done till the FIFO is full, or the user presses the Stop button. After that, the FIFO contents are transferred to MATLAB serially.

3.2 Design of the Interpolator

The basic design of the interpolator/oversampler that performs the fine measurement is shown in Fig.3.3: The asynchronous data is sampled on four clock phases [10, 15]. We have asynchronous data that is being sampled by four flip-flops, each of which runs on a 90 degree phase-shifted clock. By sampling on these four phases, the sampling frequency is increased by four. The clock used is 300 MHz. So our effective sampling frequency is 1200 MHz ,i.e., a time resolution of 833 ps. This is a 'fine' measurement, as discussed in the previous chapter.
Effectively, these four flip-flops are enough for our measurement. But as the incoming data is asynchronous, we can encounter metastability issues which are discussed in the next section.

3.2.1 Metastability, Setup and Hold Times

An edge-triggerred D-type Flip Flop transfers the input value to the output at a clock edge. The condition for proper transfer is that the data should not change for a certain time before and after the clock edge. The required time before the clock edge is called the setup time, and the time after the clock edge is called the hold time.
This time is effectively required for the transistor capacitance to charge/discharge to the high/low value. The flip-flop is a bistable circuit which has only two states i.e. 1 or 0. When a clock edge arrives inside the setup/hold time, the Flip-Flop enters an unstable state, i.e. the

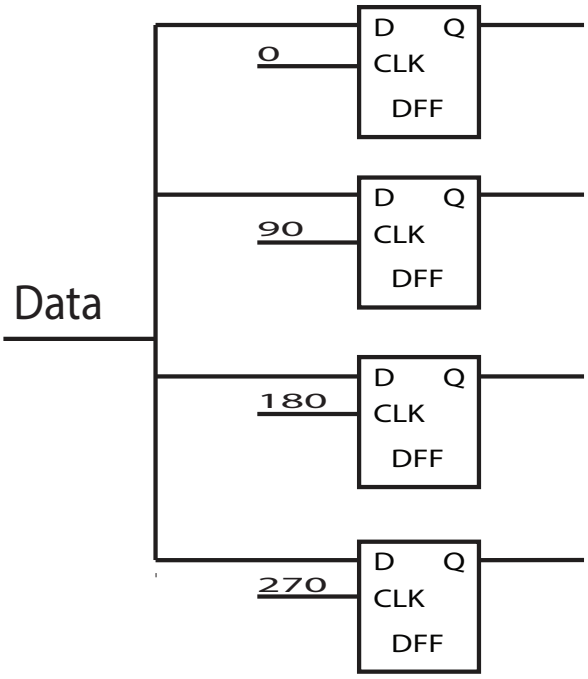


Figure 3.3: Basic 4xOversampler

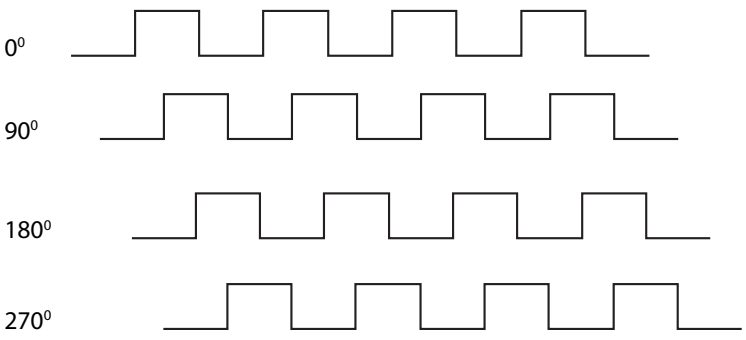


Figure 3.4: 4 Quadrature shifted clocks

voltage is neither high nor low, rather an intermediate one. There may also be oscillation in the circuit. This is a case of unstable equilibrium, as shown in the Fig.3.5: When in a metastable

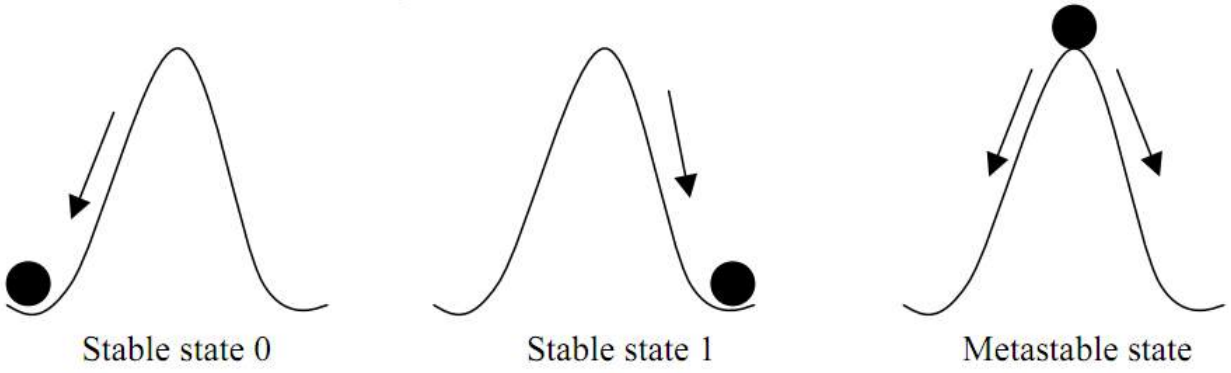


Figure 3.5: Metastability as a case of unstable equilibrium. Figure taken from [16]

state, any disturbance i.e. electronic noise in our case, will cause the circuit to reach in either of the two states. The mean time between failure (MTBF), that is the time between two meta stable events is given by

$$MTBF = \frac{1}{\text{failure rate}} = \frac{exp(\frac{t_r}{T})}{T_o f_{tn} f_{clock}} \tag{3.1}$$

where t_r=metastability resolution time, maximum time the output can remain stable without causing the system to fail, f_{tn} is the frequency of the asynchronous input, f_{clock} is the frequency of the sampling clock. T and T_o are the device characteristics [16]. So the chance that a metastable state remains for even one clock period is very small.
To avoid metastability, the standard procedure is to add another flip-flop at the output of the input flip-flop. In this way, the metastable output will not enter the system for at least another clock cycle, by which time the output has a near unity probability of being stabilized.

Besides adding flip-flops to remove metastability, we also need to synchronize the data from multiple clock domains. Consider the circuit in Fig.3.6. The successive stages of the flip-flops serve two-fold purposes:

1. avoiding metastability and
2. synchronising the four channels and bringing them into one clock domain.

Now that we have a stable, synchronized four bit fine count, we need to encode it before writing into the FIFO Buffer.

3.2.2 Encoding the Fine Count

The circuit in Fig.3.7 actually identifies the position of the posedge of the incoming asynchronous data. It tells us the quarter of the clock period in which the data arrived. To decide this, we need some extra logic.
Let the column(AAP,BBP,CCP,DDP) be defined by a four bit variable Encoder.
If(Encoder == 1110), the data arrived in the first quarter i.e. $0 - 90$ degrees.
If(Encoder == 1100), the data arrived in the second quarter i.e. $90 - 180$ degrees.
If(Encoder == 1000), the data arrived in the third quarter i.e.$180 - 270$ degrees.
If(Encoder == 1111), the data arrived in the fourth quarter i.e.$270 - 360$ degrees.

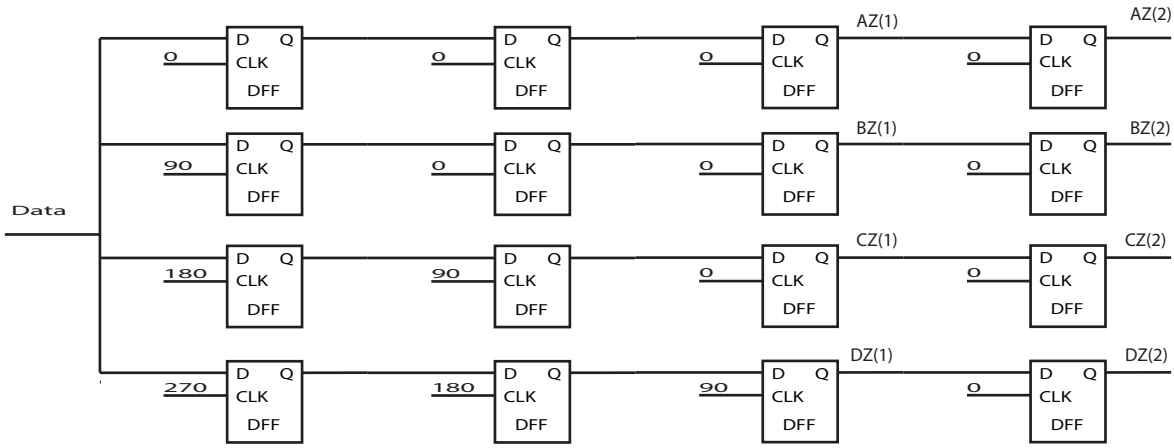


Figure 3.6: Avoiding metastability and synchronizing data

Since the arrival of data at the first flip-flop stage, it takes 3 clock cycles for the Encoder variable to encode the value. We subtract this value offline, so as to meet timing closure.

Now we have all the possible values of our fine count. For the coarse count, we need a counter running on the 300 MHz clock.

3.2.3 Design of Counter

The 12-bit counter keeps track of the coarse count. The counter starts, synchronously, on the posedge of the start signal and stops, synchronously, when the total count has been written into the FIFO buffer. Thus it stops on the negedge of the FIFO write enable signal. The counter's design is shown in Fig.3.8.

3.2.4 FIFO Buffer

The FIFO buffer we've used is 16 bit wide and 1024 elements deep. The write clock is 300MHz and the read clock is 100 MHz. This potential metastable (asynchronous) condition is resolved by using inbuilt gray counters and synchronization stages.

3.2.5 Digital Clock Manager

The four phase shifted clocks are generated through a built-in Digital Clock Manager, which generates these signals using a Delay Locked Loop. The TDC will not run until the DCM has locked i.e. the respective phase relation has not been established. The raw input 300 MHz clock is generated through a PLL.
All the modules of the system are shown in Fig. 3.9.

Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	140	28000	1%
Number of Slice LUTS	152	28000	1%
Number of Occupied Slices	53	7200	1%
Number with an unused Flip-Flop	44	184	23%
Number with an unused LUT	32	184	17%
Number of fully used LUT-FF pairs	108	184	58%
Number of Bonded IOBs	10	480	2%
Number of LOCed IOBs	10	10	100%
Number of BlockRAM/FIFO	4	60	6%
Total Memory Used(KB)	126	2160	5%
Number of BUFG/BUFGCTRLs	7	32	21%
Number of DCM_ADVs	1	12	8%
Number of PLL_ADVs	1	6	16%
Average Fan-out of Non-Clock Nets			2.57%

Table 3.2.5 shows that the SLICE LUTs and Slice register utilization is just 1% . The timing constraints that the design meets are shown in table 3.2.5.

Parameter	Value
Read Clock	100 MHz
Interpolator, Counter and FIFO Clock	400 MHz
Maximum skew of Interpolator Clocks, 0 and 90 Degree	300 ps
Maximum Skew of Counter Clock	1 ns
Maximum Skew of Input Data	400 ps
Maximum Clock Jitter	100 ps

The Timing score reported was zero. No manual placement and routing was performed.

3.3 Performance Testing

To measure the performance of our TDC, we input time intervals of known values and compare the measured time interval. As shown in Fig. 3.10, a time interval between the START and STOP pulses was generated by dividing a signal through a BNC-Tee, and passing one branch through a slightly longer wire. The time difference of arrival of the two pulses was first measured through a high speed oscilloscope(Picoscope 5203 1 GHz Sampling Frequency). Fig. 3.11 shows different delays measured by the TDC. We see that the standard deviation in the measurement decreases as larger delays are measured. The second test performed is a modification of the one shown in Fig. 3.10. Consider Fig. 3.12. The TDC now registers time stamps of pulses arriving at a single channel. The system is initialized arbitrarily. We use an internal FPGA to generate high frequency clocks.
For example, in Fig.3.13, we're calculating time stamps of an incoming periodic pulse. The pulse is generated through an on-chip PLL and is fed back to an input of the TDC. The upper plot shows the timestamps of a periodic pulse. The lower plot shows successive time difference between the input edges. We see 20ns being measured. Note that in this configuration, the TDC does not operate in a START/STOP mode, rather it registers timestamps of incoming pulses.
Note that in Fig.3.13 there are a lot of spikes in the 100 MHz plot, indicating faulty readings. This is because the clock was fed through a normal copper jumper wire, which does not maintain signal integrity at such high frequencies due to parasitic reactances, skin effects, and interference. Nevertheless, 42% of the readings correspond to 10 ns time intervals.
In all the above readings, the resolution of the measurement is 833 ps.
In the next chapter, we explain the use of the TDC to measure the time difference between the capture and decay of a cosmic ray muon. After measuring the time of a large number of such decay events, we can estimate the lifetime of the Muon.

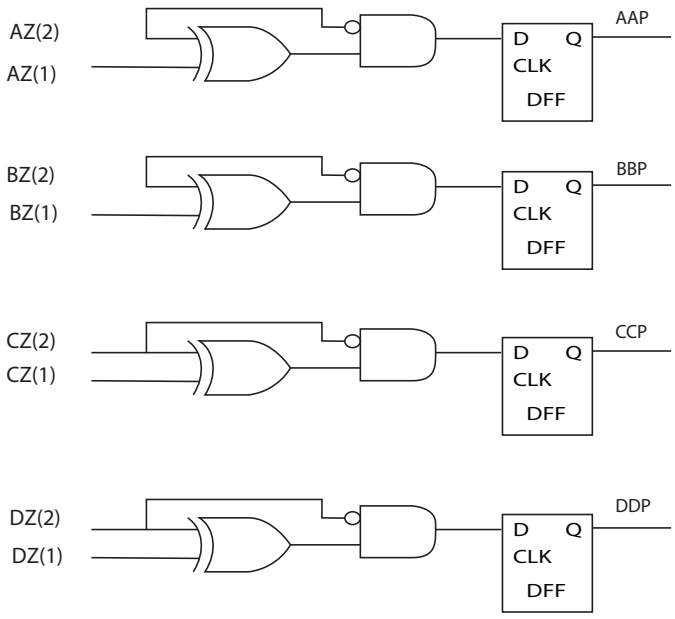


Figure 3.7: Posedge detection logic

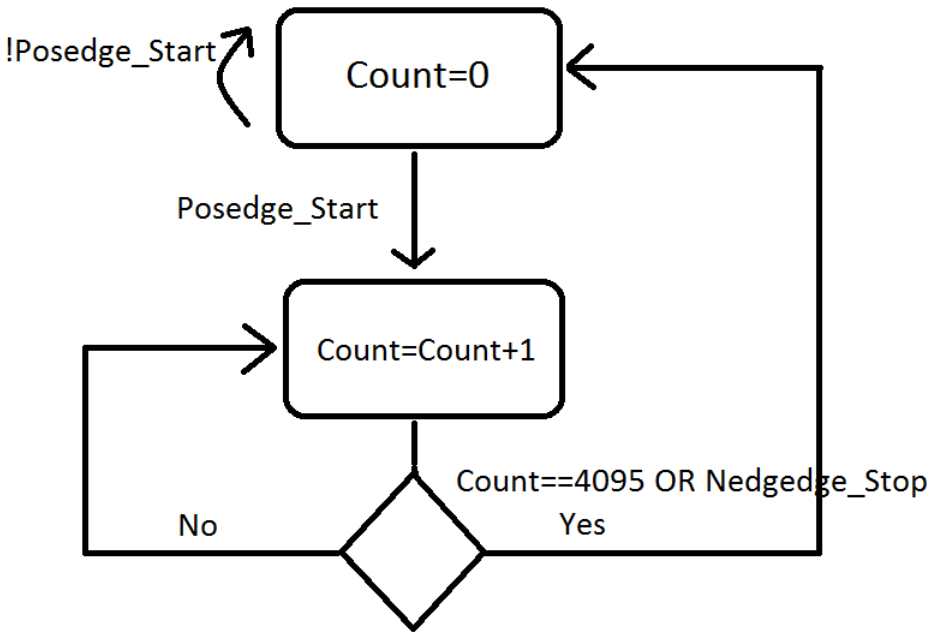


Figure 3.8: Finite state machine of the counter

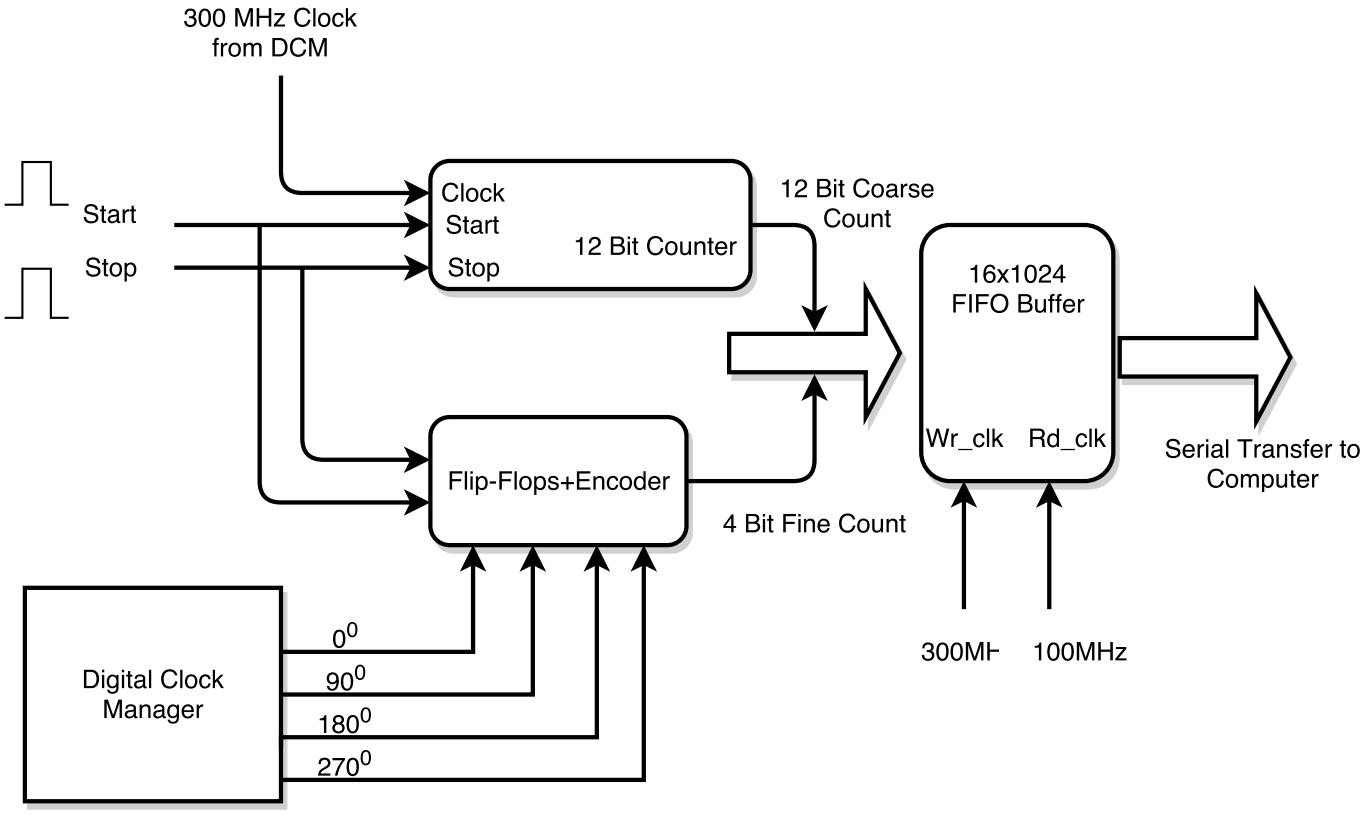


Figure 3.9: Module-level Diagram of the Interpolator

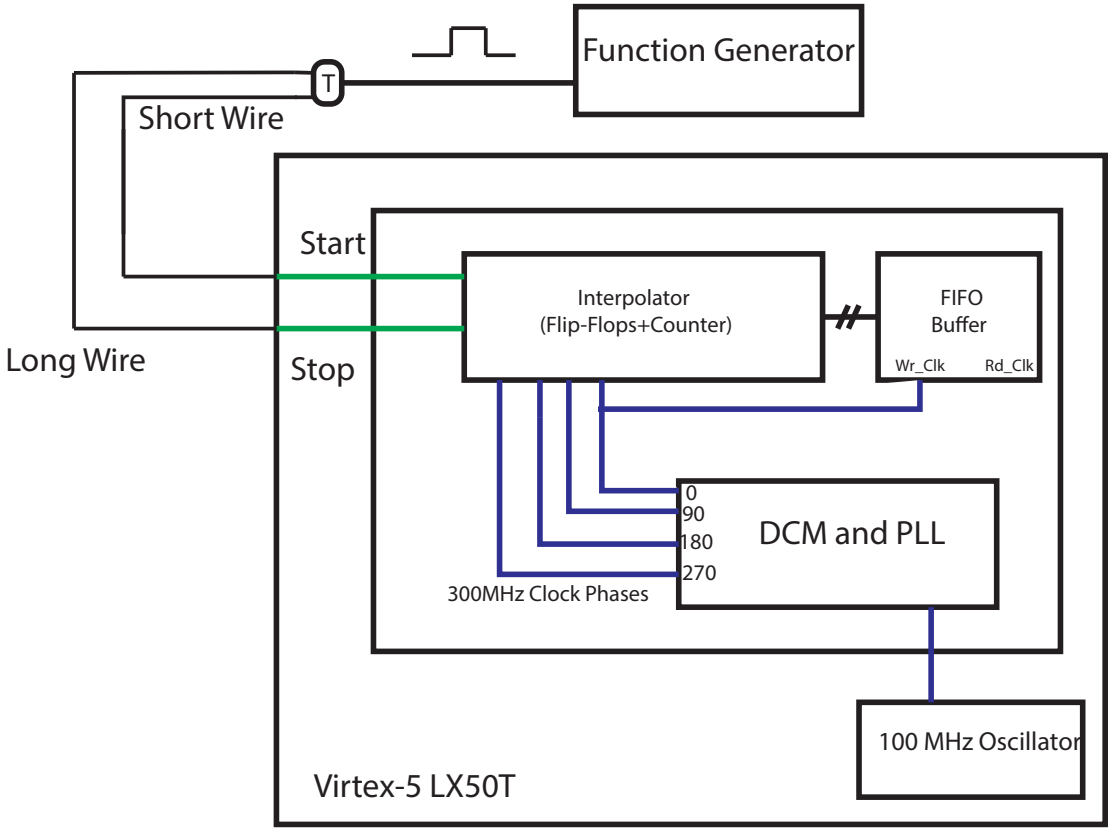


Figure 3.10: Measuring a known delay with the TDC

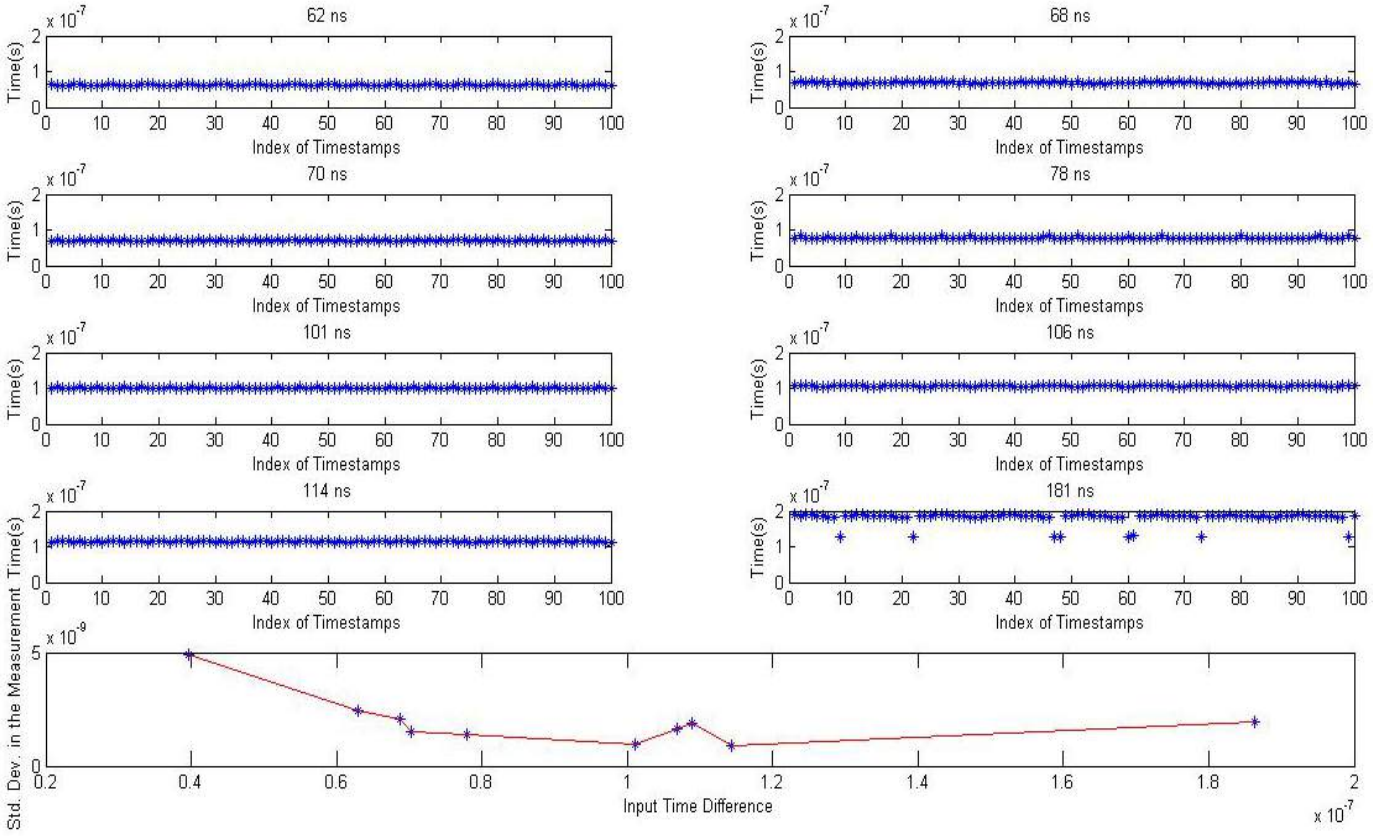


Figure 3.11: Measuring different delays with the TDC. As the time delay measured increases, the standard deviation in the measurements decreases, as shown in the bottom figure

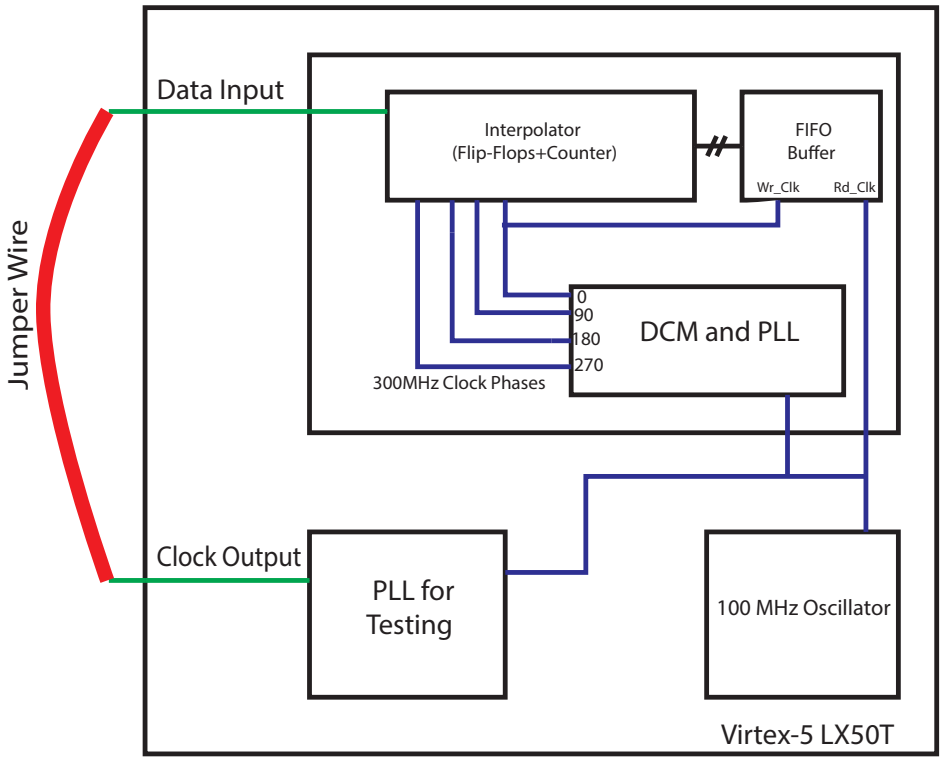


Figure 3.12: Recording timestamps of periodic waves of different frequencies synthesized inside the FPGA

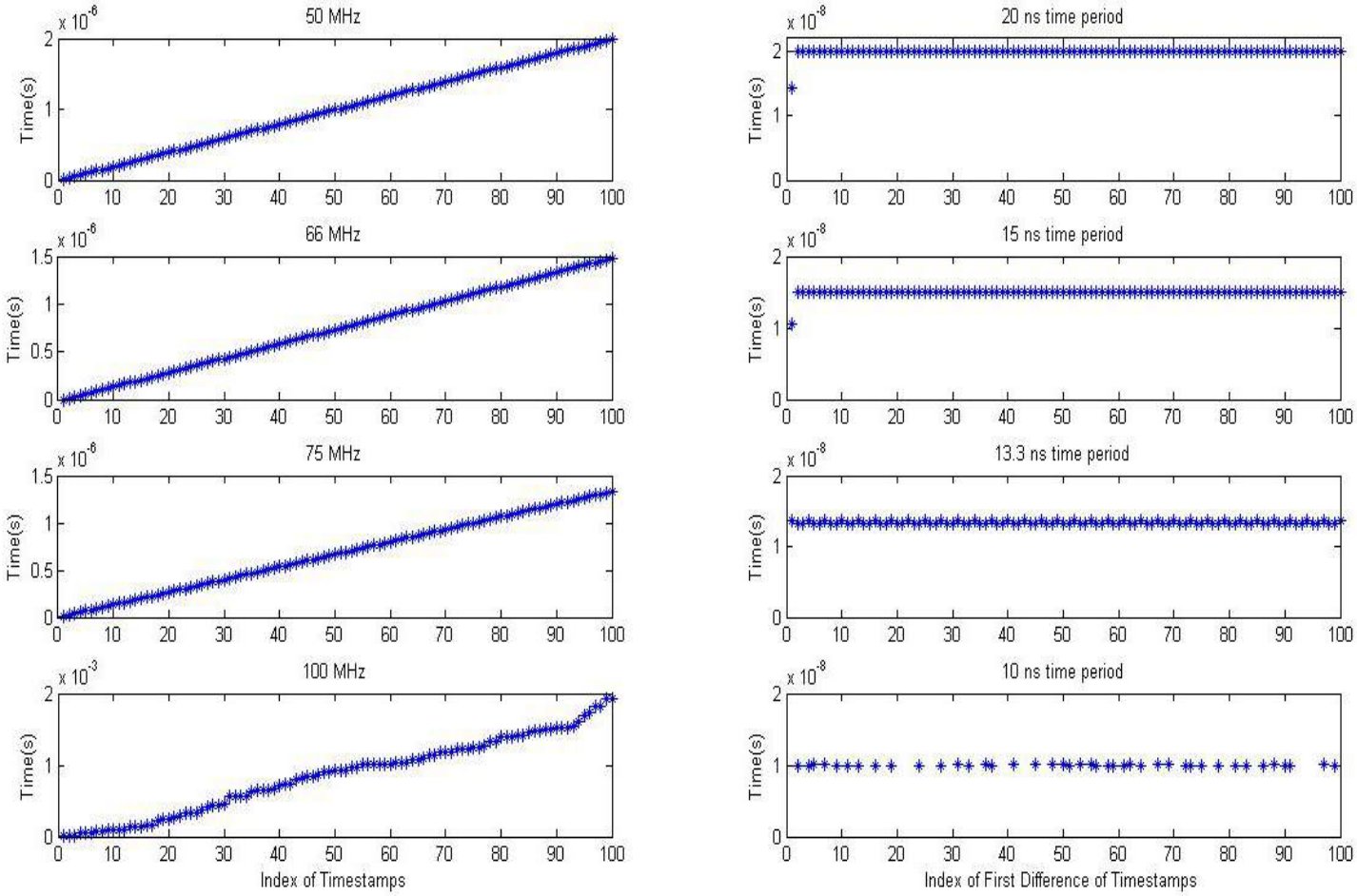


Figure 3.13: Recording timestamps of periodic waves of different frequencies synthesized inside the FPGA. The figures on the left show timestamps of different frequencies while the figures on the right show the difference between the consecutive timestamps, that is, the time period of the input pulse

.

Chapter 4

Measuring the Lifetime of Cosmic Ray Muons

In the previous chapter, an FPGA based TDC was characterized and prototyped. This TDC can now be used to perform high resolution time measurements. A specific example is measuring the lifetime of cosmic ray muons.
Conventionally, time to amplitude converters (TACs) are used in experiments such as lifetime measurements, coincidence detection and fluorescence spectroscopy. Muon lifetime experiments have been performed both with TAC's and TDC's [17, 18, 19, 20]. Relativistic effects in muon lifetime have also been measured with readily available transistors and micro-controllers [21]. We describe the use of a TDC in one such experiment. The chapter begins with describing the theoretical background and the setup of the muon lifetime measurement.
The use of the FPGA in this experiment is not just limited to time measurement. We harness the high speed programmable logic units in the Virtex-5 chip to perform coincidence detection, which was previously performed by independent logic unit modules.
The chapter is organised in the following sections:

1. Introduction to cosmic ray muons,
2. Description of the generic lifetime experiment.
3. Use of TDC in the lifetime experiment.
4. Use of TDC with logic units in the experiment.
5. Concluding section describing possible improvements and future experiments

In preparing this chapter, I have benefited greatly from Uzair Latifs report on muon lifetime measurement [22], performed as a design project at the Physlab, Lahore University of Management Sciences (LUMS).

4.1 Introduction to cosmic ray muons

Muons have a charge of $-e$ but are more massive as compared to the electron, possessing around 106 MeV.
When cosmic rays, which are composed mostly of helium and hydrogen nuclei, reach the earth, they interact with the atmosphere. There occur three types of interactions:

1. electromagnetic interaction,
2. inelastic Hadronic interaction, and

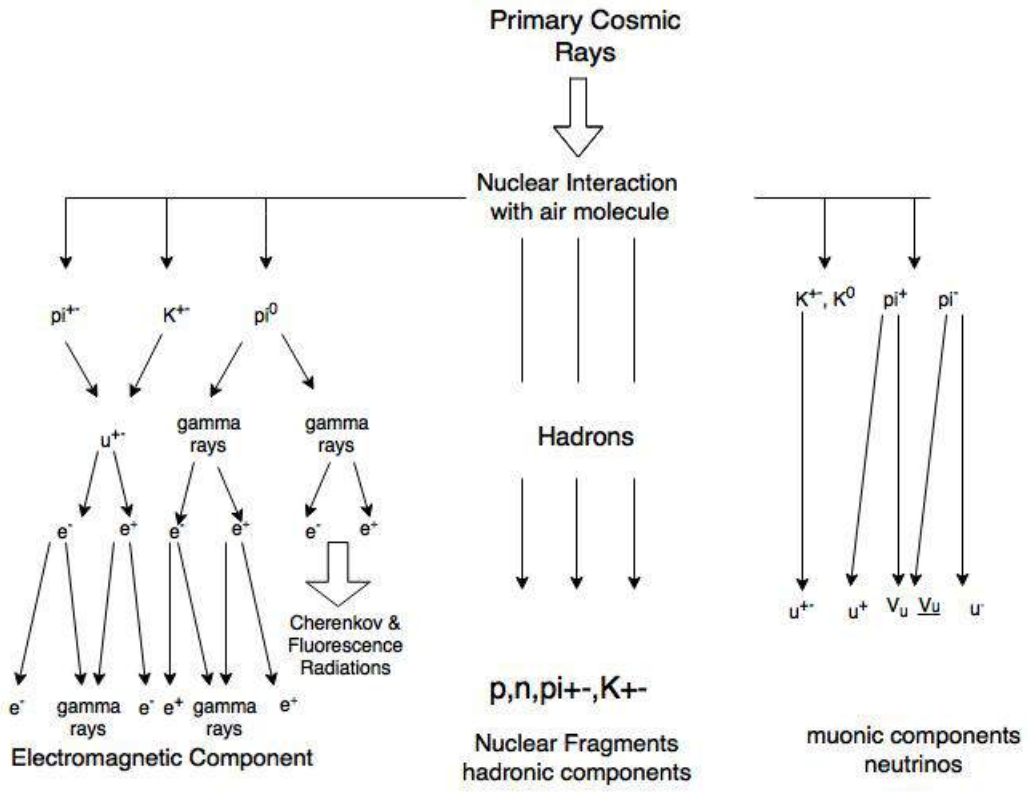


Figure 4.1: Creation of Muons and other particles in the atmosphere

3. nuclear decay interaction

Once the cosmic rays enter the atmosphere, there is a cascade of nuclear interactions. At the ground level, these are known as cosmic ray showers. They have two components to them: the 'soft' electromagnetic component and the 'hard' muonic component. Fig. 4.1 shows a general schematic representation of these cosmic ray showers. By measuring the mass, charge and energy of a particle at a certain altitude we can determine exactly the type of the particle its place in the cascade. Measurements are usually performed by detectors placed on hot air balloons, airplanes or by satellites high up in the atmosphere [22].
Muons are high energy particles created by the interaction of cosmic rays and the atmosphere. They are created approximately 30 km up in the atmosphere and have speeds approaching 0.98c. At this speed, they take around 10^4 seconds to reach the ground. Considering their lifetime is on the order of $10^{-6}s$, classical physics fails to explain the high flux of muons at the ground. This dilemma was resolved by special relativity, according to which the muon experiences length contraction and the observer in the Earth's frame experiences time dilation.

4.2 The Lifetime Experiment

If there are N_o stationary muons in a container at time t_o, then the number of muons at any later time t follow an exponential distribution:

$$N(t) = N_o \exp[\frac{-t}{\tau}] \tag{4.1}$$

where tau is the time after which 63% of the N_o muons will have decayed. We refer to τ as the lifetime of the muon. Before measuring the lifetime of a muon, we must consider that most

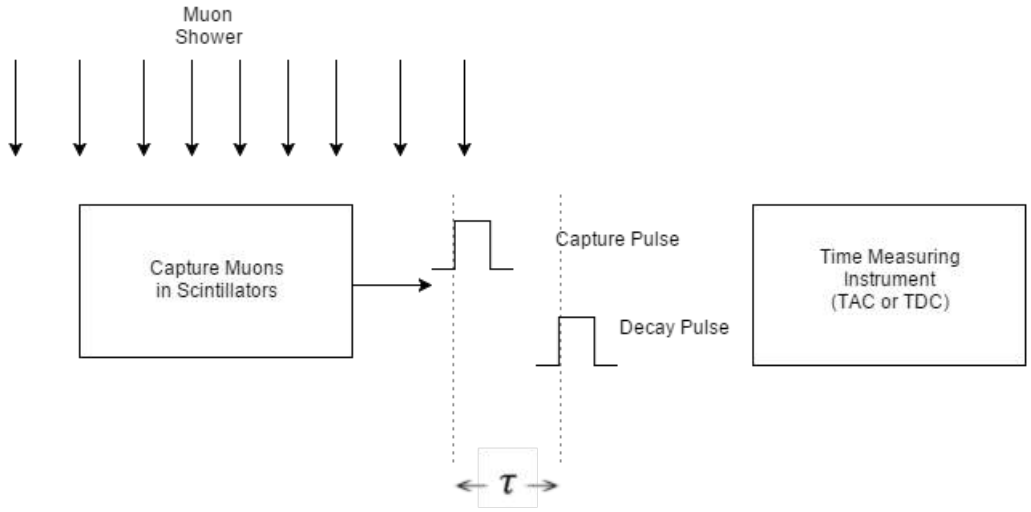


Figure 4.2: A Simple schematic of the lifetime experiment. A muon is captured inside the scintillator. After time τ, the muon decays. This time difference τ is the muon lifetime.

of the matter is transparent for these particles. Therefore, to capture them we need a dense material in which the probability of capturing the muons is high. Also, the material needs to give a signal when a muon enters it. Such materials are called scintillators.
Fig.4.2 shows the basic concept of the experiment. There is a constant muon shower upon the earth, and we capture a fraction of the shower through the scintillator. Suppose a single muon is captured. After time τ, it decays into an electron and two neutrinos. Both of the events, the capture and the decay, generate an electric pulse. The time measuring instrument measures the time difference between the two pulses, which corresponds to the lifetime of the captured muon. The same process is carried out with a high number of muons, and the lifetimes measured are histogrammed to get a probability distribution. This distribution is then fitted with eq.(4.1) to obtain the value of τ. Note that we measure the muon's lifetime in its rest frame.
A detailed schematic of the experiment is shown in Fig. 4.3. We aim to measure the time interval between a capture and successive decay of the captured muon. These time intervals are measured with the help of discriminators, logic units and the FPGA-based TDC. A brief explanation of the components used in Fig. 4.3 is presented first.

4.2.1 Scintillators

A scintillator is a material which fluoresces when a charged particle passes through it and excites its atoms. We use plastic, specifically Polyvinyl Toluene (PVT) scintillators which have a high optical output and a relatively quick signal decay time, of around 2 to 4 ns. Scintillators also exhibit after pulsing, which accounts for much of the noise in the signal [23]. The material PVT has a refractive index of 1.58. The three plastic scintillators used in this experiment were wrapped in highly reflective aluminized mylar sheets and then were made light tight by covering them with black tape.

4.2.2 Photomultiplier Tubes (PMT's)

These are extremely sensitive vacuum tubes which detect light in the UV, visible and near IR region. PMTs are able to amplify the current produced by light signal by about a 100 million times. This is allowed by the several dynode stages present in this tube and this enables us to detect even single photons in the scintillator.
Figure 4.4 explains the working of the PMT. A photon produced in the scintillator strikes the

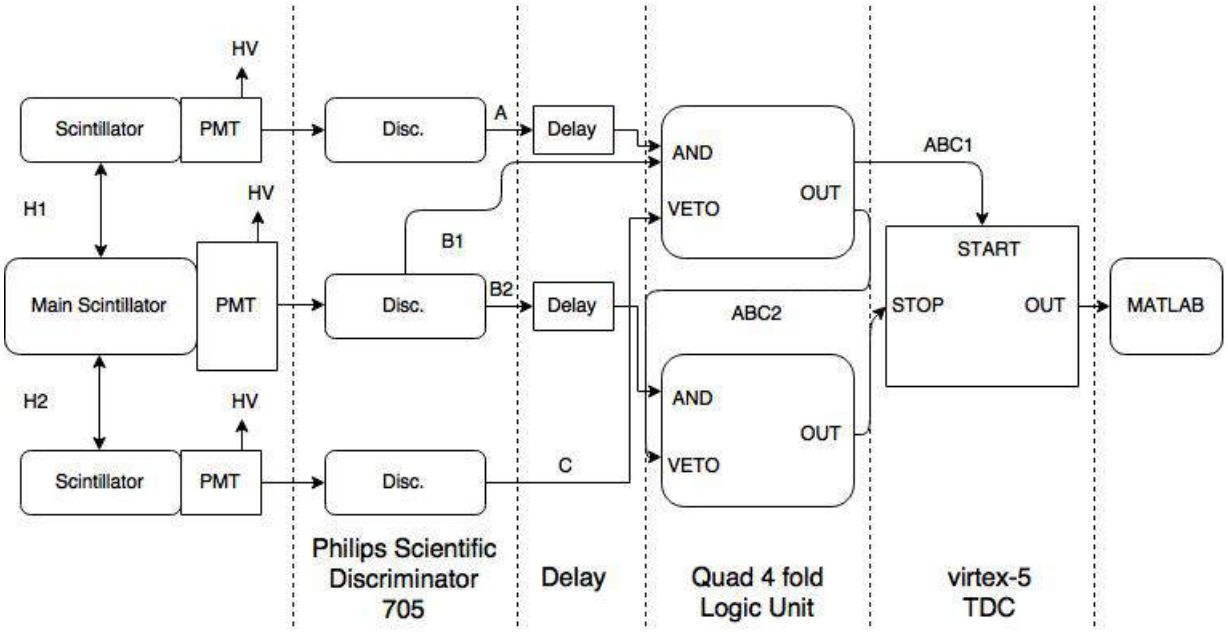


Figure 4.3: Conceptual organisation of the experiment for measuring the muon lifetime

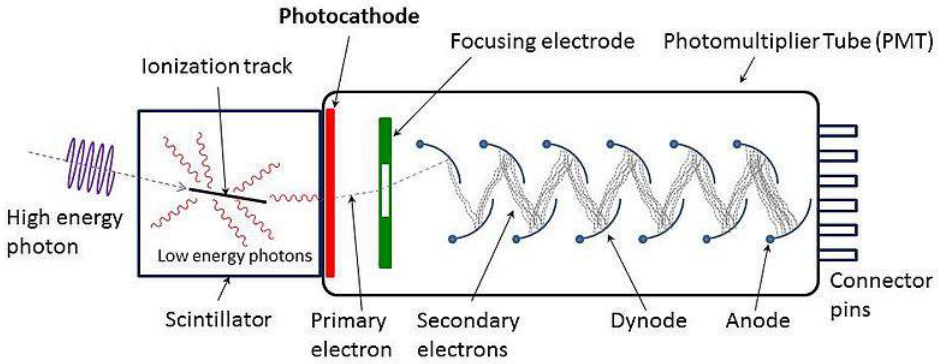


Figure 4.4: Schematic of a Photomultiplier Tube [Courtesy, Wikipedia "Photomultiplier Tubes"]

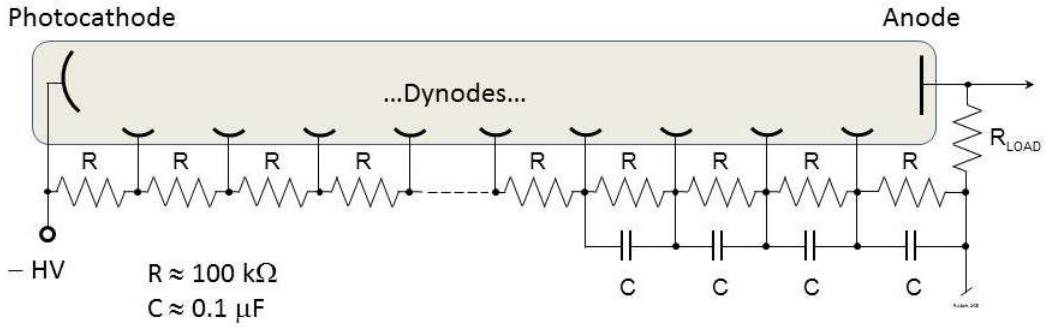


Figure 4.5: Circuit of a Photomultiplier Tube [Courtesy, Wikipedia "Photomultiplier Tubes"]

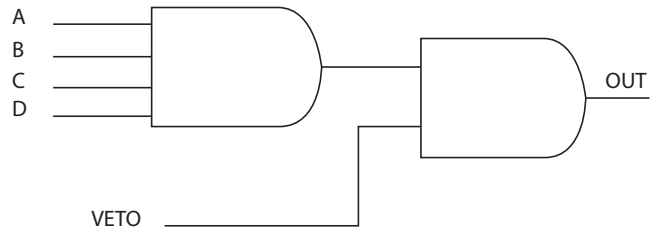


Figure 4.6: Schematic of the Logic Unit

photocathode which emits electrons due to the photoelectric effect. The electrons are collimated using electrodes and accelerated using intermediate electrodes called dynodes. Each successive dynode is more positive than its predecessor. At each dynode stage, the electronic signal is amplified. Ultimately, we receive a negative pulse at the anode. An electric circuit model of the PMT is shown in Fig.4.5. The anode is kept at ground while the photocathode is kept at a high negative voltage. The dynodes form a voltage divider.
We use three PMT's(1200P Rexon Components) and scintillators, which are collectively called the detectors. Detectors A and B are the smaller ones while detector B is the thicker one in the middle.

4.2.3 Discriminators

These are threshold detectors and convert distorted pulses received directly from the PMT's into square pulses. Whenever a PMT pulse exceed a certain threshold, the discriminator outputs a square wave whose width can be varied. The threshold can be adjusted to a level where low-level PMT noise can be discarded and does not pass from the discriminator. Nevertheless, any noise pulse that exceeds the threshold does feed through. The detector used is Phillips Scientific Octal Discriminator Unit Module (Model 705), shown in Fig. 4.7.

4.2.4 Logic Units

These are actually 4-input AND gates with a veto input which renders the AND gate output Null. Fig.4.6 explains the working of the Logic unit. Actual logic units and discriminators are shown in Fig.4.7. We use Phillip Scientific Quad Four Fold Logic Unit Module (Model 755).

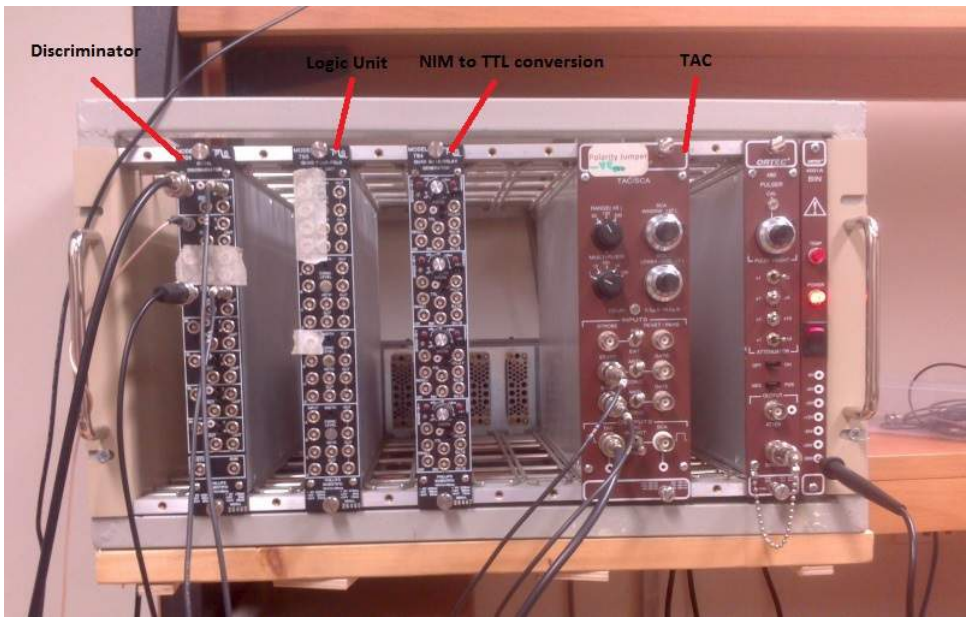


Figure 4.7: Discriminators and Logic Units

4.2.5 The Experimental Setup

The lifetime experiment is not as simple as depicted in Fig. 4.2. It is because of the constant muon flux, and the multitude of pulses generated in the scintillators. The combination of discriminators and logic units serve to eliminate the pulses produced due to noise, afterpulsing and other muons. To achieve this, we use 3 detectors A,B and C, shown in Fig. 4.8.
Detectors A and C are relatively thin and are generally unable to stop the muons. The detector B, however, is five times thicker, and has a high probability to capture the incoming muons. If a muon passed through all three detectors, all three would generate a pulse. These three pulses are together known as the ABC signal. If the muon stopped in B, C would not produce a pulse and we call this an $AB\overline{C}$ signal.
We measure the lifetime of a muon in its rest frame. This means that when the muon is stopped in detector B, an $AB\overline{C}$ signal is generated. After a certain time, the captured muon decays, and a B2 signal is generated by the release of either an electron or a positron depending on whether it was μ^- or μ^+ muon. Measuring the time difference between the $AB\overline{C}$ and the B2 pulses gives us an exponential time distribution the gradient of which is the mean lifetime of the muon.

4.2.6 Method and Results

To measure the lifetime, we need to measure time difference between arrival of two pulses, $AB\overline{C}$ and B2. We call the $AB\overline{C}$ signal as START signal, and the B2 signal as STOP signal.
To generate a START signal, we need to AND the A,B signals and set C as the Veto input so that the signal is discarded if C pulse arrives. But to AND the A and B signals, they need to be in phase i.e. the pulses should coincide in time. As the A pulse arrives earlier, because the muon entered the A detector before entering B, it needs to be delayed so as to be brought in coincidence with the B pulse.
Similarly, to generate the STOP,i.e. B2 signal, we need to input the B signal to a logic unit. But a B signal is also generated whenever a START signal, i.e. $AB\overline{C}$, is generated. To distinguish between the B pulse due to START signal and the actual B2 pulse, we input the START signal $AB\overline{C}$ into the veto of the B logic unit. We also need to delay the B pulse so that it coincides in time with the veto $AB\overline{C}$ for properly vetoing.

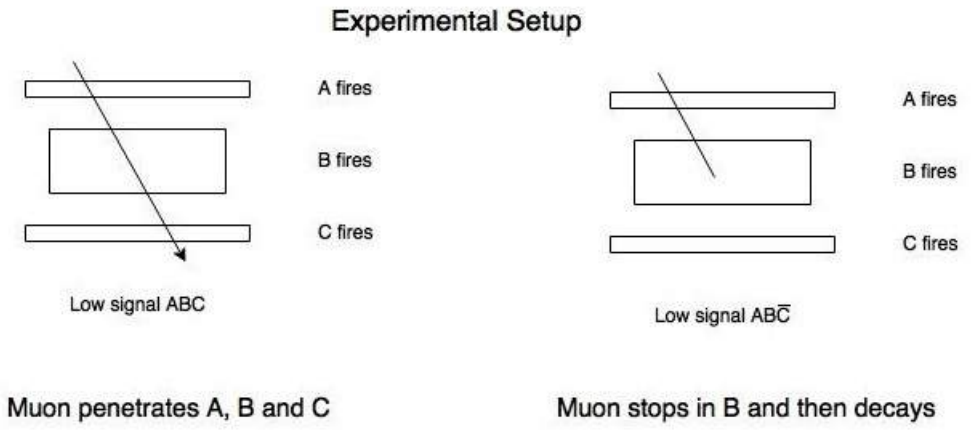


Figure 4.8: The three detectors are placed such that B is sandwiched between A and C for the experiment. When a muon passes through all three detectors, all three generate pulses.This is known as an ABC signal. When the muon is stopped in B, only A and B detector generate pulses. These together are known as the $AB\overline{C}$ signal.

4.2.7 Data Acquisition

We have a START and a STOP pulse from the logic units. We feed these pulses to the FPGA based TDC which then records the time stamps of the input signals. The flow of the FPGA's state machine is discussed in section 2.1. After the FIFO buffer is filled with the timestamps, we transfer them to MATLAB through serial communication.
A single FIFO buffer stores 1024 events. Typically, it takes approximately 5 hours to fill the buffer. The rate of muon events detected is then 0.056 count/sec (cps). We take 4 such runs and plot a histogram of the time stamps.
If we plot the time stamps as obtained from the FIFO buffer without histogramming, we obtain the data shown in Fig. 4.10. Each point in Fig.4.10 represents the time interval between a muon capture and subsequent decay event. The y-axis corresponds to the time in microseconds, while the x-axis corresponds to event's position in the FIFO buffer. We see that the bulk of the events are on the lower side of the y-axis, that is, most of the muons decayed very quickly. This is evident if we plot a histogram of these raw counts, as shown in Fig. 4.11. We see an exponential distribution. On the x-axis is the time, and the y-axis plots the frequency of the lifetime observed in a particular bin of time. In this graph, we have divided the whole of x-axis into 100 bins. Thus each bin corresponds to roughly $\frac{13\mu s}{100} = 130ns$.
Observe that the first bin shows zero counts. This is done intentionally as most of the B PMT noise resides in this bin and would easily distort the exponential distribution. As an example, consider the peaks on the left side in Fig.4.12, which is a histogram implemented in a conventionally implemented multichannel analyser (AMPTEK's MCA8000D).

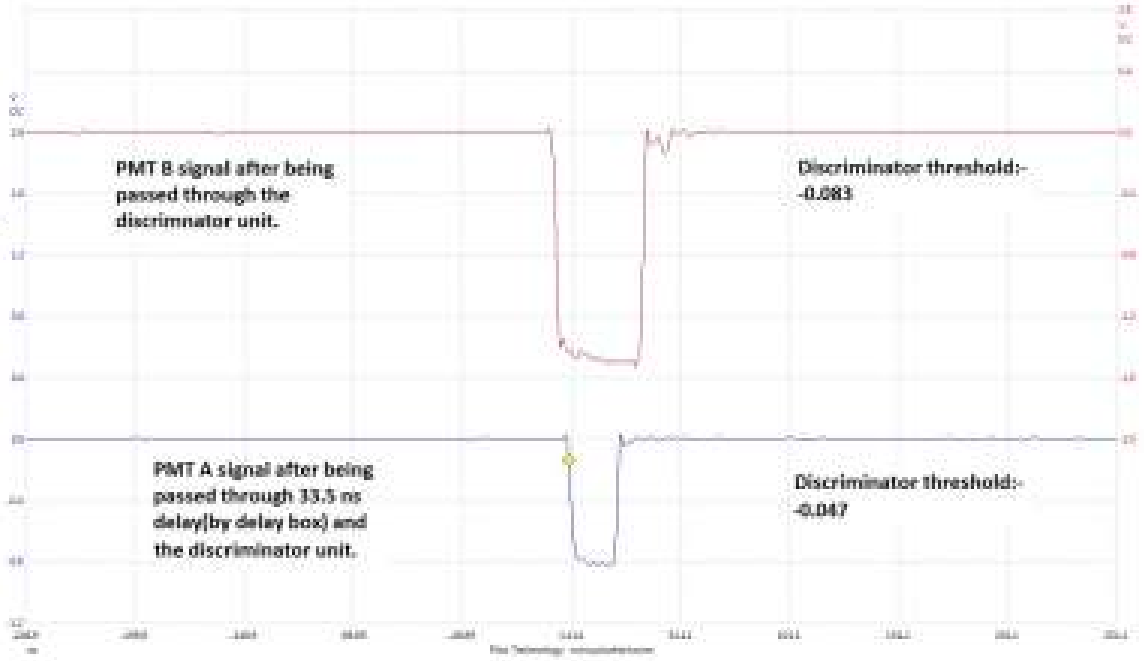
4.2.8 Curve Fitting

We fit the histogram in Fig.4.11 to the Eqs. below

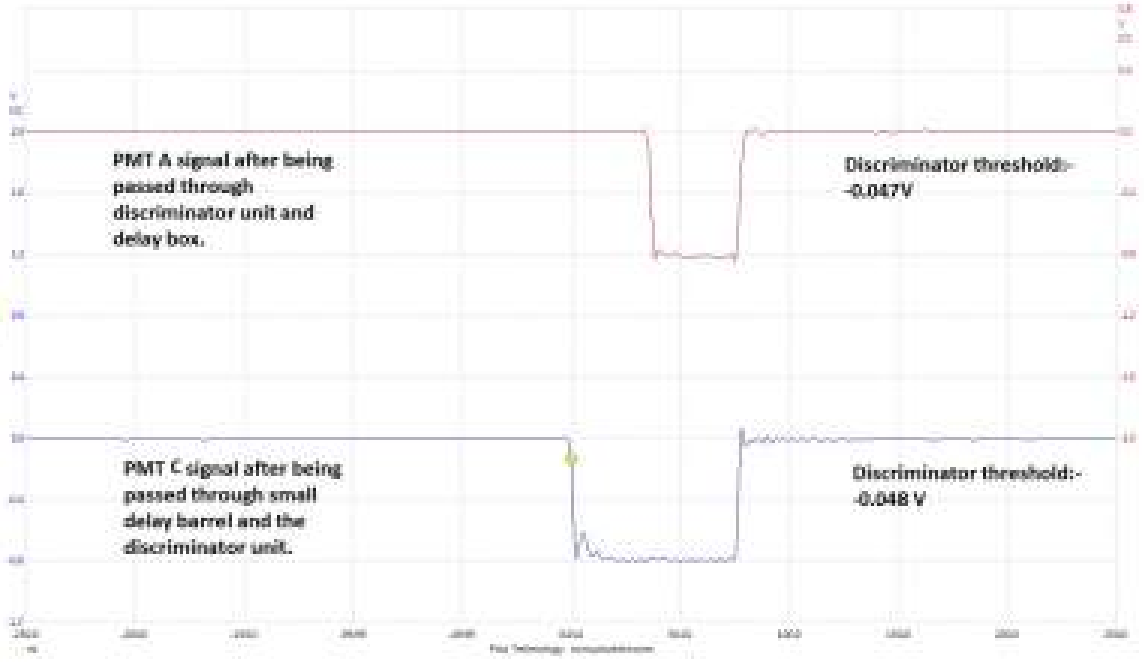
$$F(x) = A(0.44e^{\frac{-t}{1.7\times10^{-6}}} + 0.56e^{\frac{-t}{\tau}}) \tag{4.2}$$

As a result, we obtain the fit values for τ and A. In eq.4.2, 1.7×10^{-6}s is the estimated lifetime for the $\mu-$ lifetime. We use the MATLAB function *nlinfit* to perform curvefitting. The fitted, normalised probability distribution is shown in Fig. 4.13. The estimate for the

(a)


PMT B signal after being passed through the discrimnator unit.
Discriminator threshold:- -0.083
PMT A signal after being passed through 33.5 ns delay(by delay box) and the discriminator unit.
Discriminator threshold:- -0.047

(b)


PMT A signal after being passed through discriminator unit and delay box.
Discriminator threshold:- -0.047V
PMT C signal after being passed through small delay barrel and the discriminator unit.
Discriminator threshold:- -0.048 V

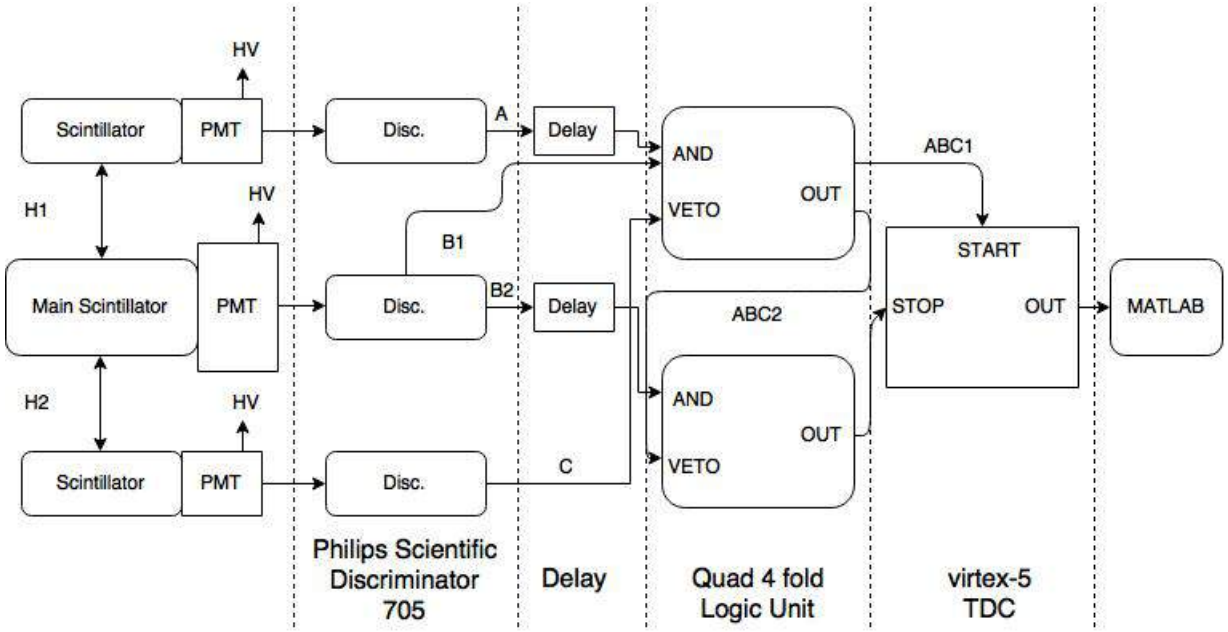


Figure 4.9: The Schematic of the Lifetime Experiment

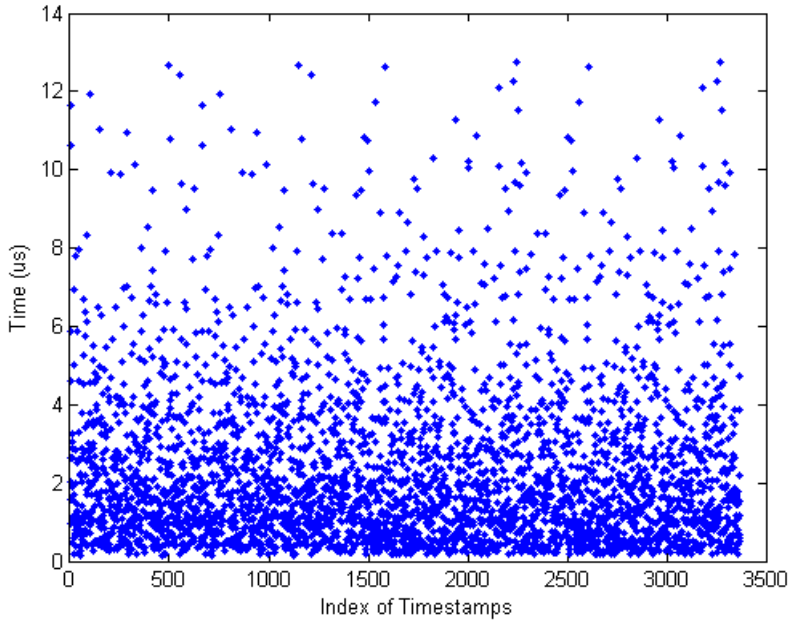


Figure 4.10: Timestamps of muon events as retrieved from the FIFO Buffer. The data shown is for three runs of the experiment, thus the 3000 plus timestamps.

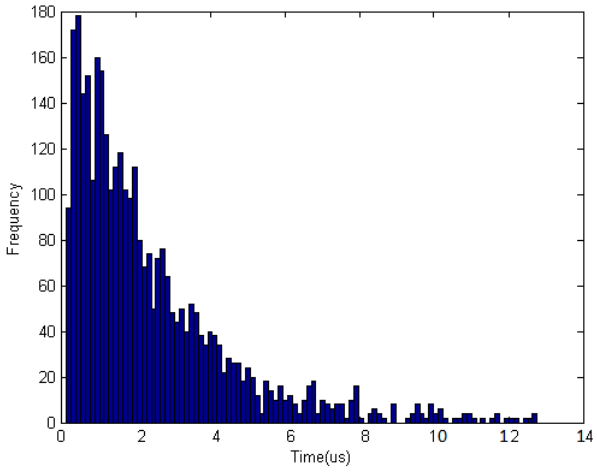


Figure 4.11: Histogram of the timestamps shown in Fig.4.10. Bin count is 100

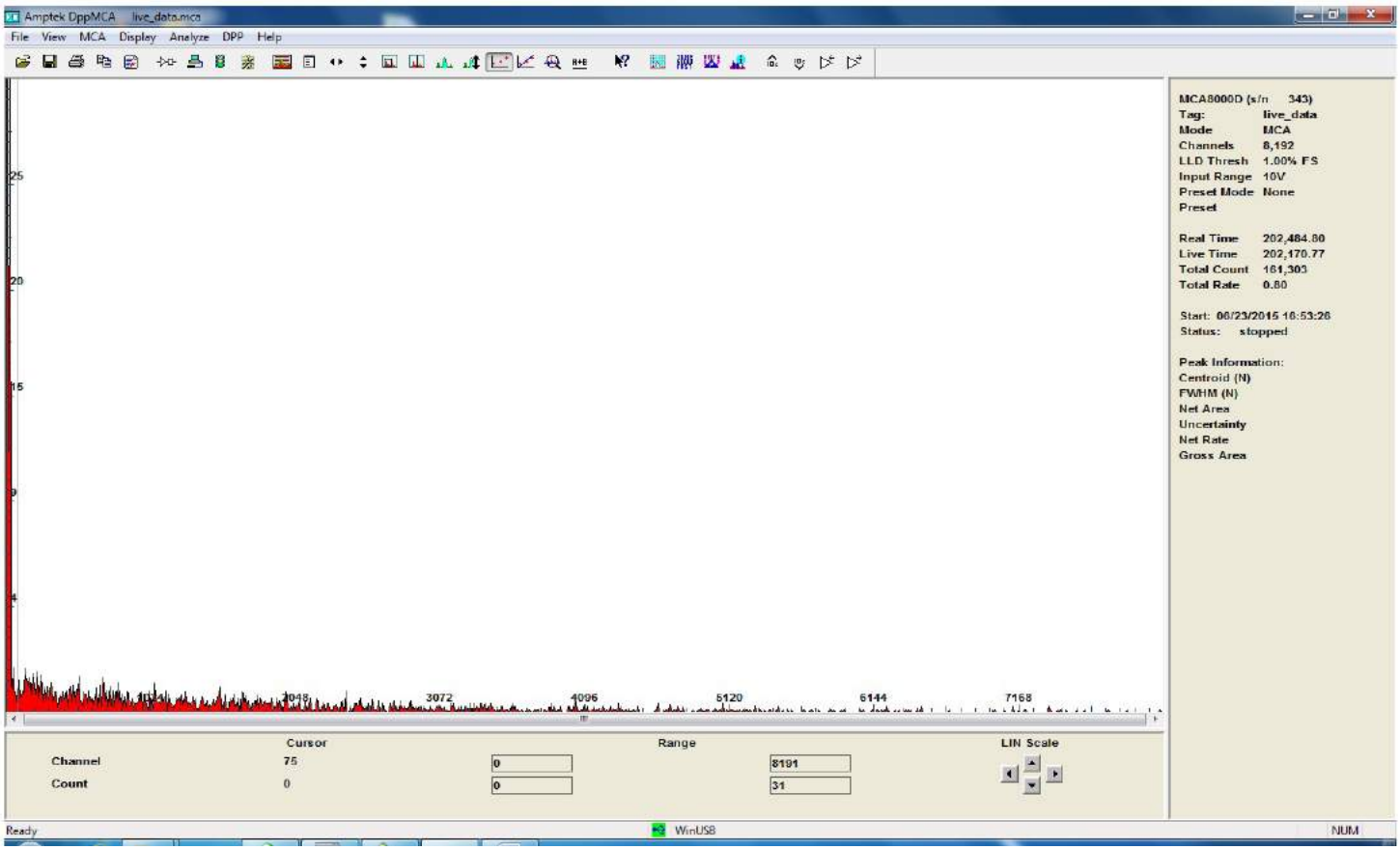


Figure 4.12: A histogram obtained through a Multichannel Analyzer(MCA) and a TAC. The peak on extreme left is the noise due to B PMT.

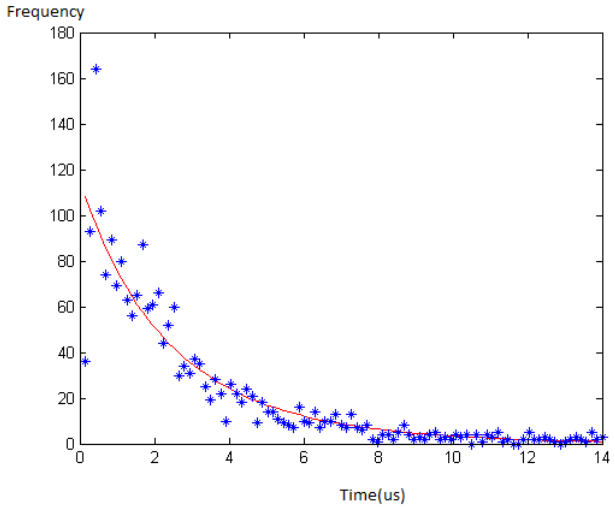


Figure 4.13: The red line shows the fitted data with the timestamps

lifetime parameter obtained is $2.3\mu s$ with 95% confidence bound interval of $(2.26 - 2.72)\mu$s. Alternatively, we have $2.3 \pm 0.12\mu s$, assuming normal distribution for the standard error.

4.3 Implementing the Logic Unit in the FPGA

In this section, we explain how the logic units are also implemented in the FPGA. The schematic of the experiment will be as shown in Fig. 4.14. The logic unit in the muon lifetime experiment is a sort of real-time filter that accepts only those of A, B and C pulses that, in zero noise conditions, would correspond to an actual muon capture and decay event. The Virtex-5 FPGA is also capable of performing high speed AND logic on two pulses. The AND gate, and all other combinational logic is performed through a six input LUT. The maximum speed of a V5LX50T chips LUT is 550 MHz (period of 1.81ns)[12]. The pulses generated by the PMTs and discriminators have a width of 50 ns. This means we can easily implement the coincidence logic on the FPGA. The following logic can be used to detect, from the conglomeration of pulses arriving from the PMTs, an actual muon decay event.

4.3.1 Algorithm for implementing the logic unit

Fig. 4.15 describes the algorithm used to detect an actual muon decay event from the conglomeration of pulses arriving from the PMT's. Here is a narrative.

1. The FPGA starts its measurement when it detects a posedge of an A pulse. Otherwise, it remains in this waiting state.

2. Once an A pulse is received, the FPGA waits for a B posedge. During this interval, if any C posedge is detected, the FPGA goes back to the initial state, since a C pulse indicates that a muon was not captured in PMT. During this state, any A pulses that are received are simply rejected, i.e., don't care conditions.

3. If a B posedge is received, the FPGA treats it as a START pulse and waits for the next B posedge i.e. the STOP Pulse. In this state, any A and C pulses are rejected.

Fig. 4.16 shows valid cases in which the system would actually measure the time between the B(START) and the B(STOP) pulse. Notice that the C pulse never comes in between the A and

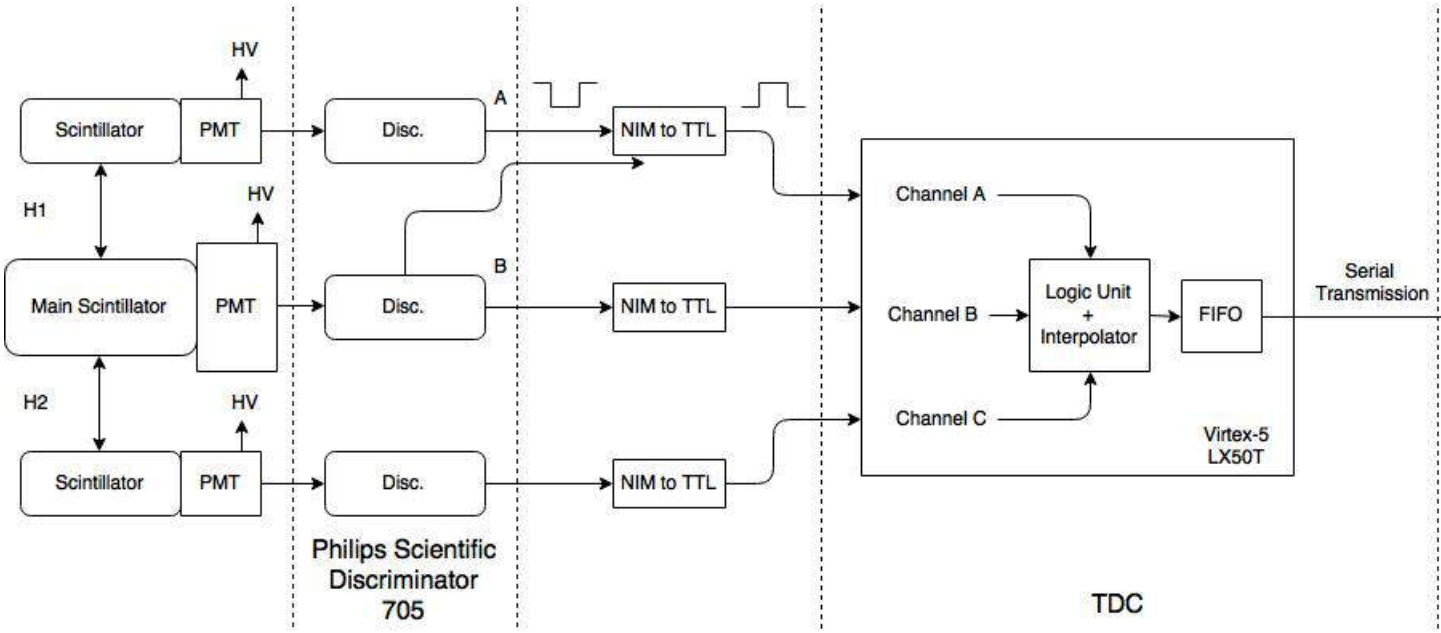


Figure 4.14: The lifetime experiment, with the logic units implemented on the FPGA

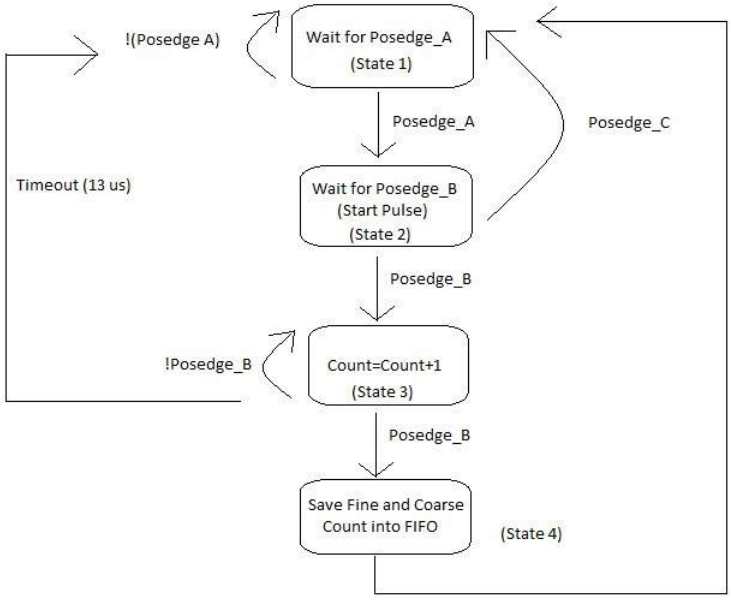


Figure 4.15: General Algorithm for a muon Even Detection

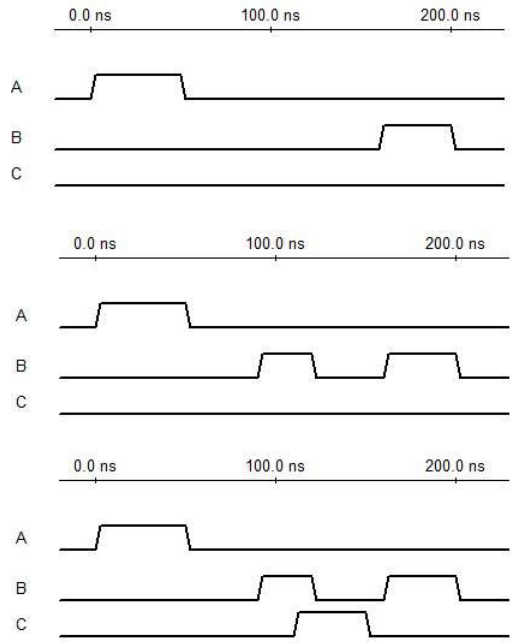


Figure 4.16: Conditions of valid START and STOP signals are satisfied in all three cases

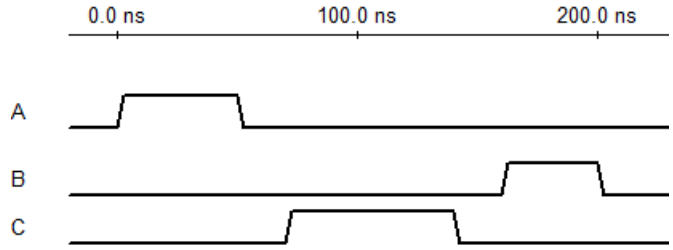


Figure 4.17: Conditions of valid START and STOP signals are not satisfied

B(START) pulse. If a C pulse does come in between an A and B pulse, it VETOes or discards the reading, as shown in Fig. 4.17. In absence of any PMT noise, this algorithm should give a fairly accurate exponential distribution. In practice, there will be noise contribution from other muon decays. But our PMTs give enough noise to make this algorithm fail, and deceive the FPGA into registering false muon decay events very fast, filling the 1024 deep buffer in under half an hour. We need to modify the algorithm to avoid this.

4.3.2 Modification to the algorithm

To make our algorithm avoid registering PMT noise and false muon events, we need to make the following changes to the algorithm.

1. In an actual muon decay event, there are precise time intervals (with a tolerance of around 5ns) between the A,B and C pulses which can be empirically determined by a fast oscilloscope. For example, the time interval between the posedges of A and C pulses,(when the oscilloscope is triggered by the A pulse) is seen to be separated by 2-8 ns. Note that we dont always observe an A and a C pulse on each trigger, but when we do see both, they are separated by a time interval ranging from 2-8 ns. Similarly, an A and B pulse are separated by a time interval of around 30-50 ns.

2. In the original algorithm, the system waits indefinitely for a B pulse once an A pulse is detected, until there's a timeout or a C pulse is detected. There can be two cases:

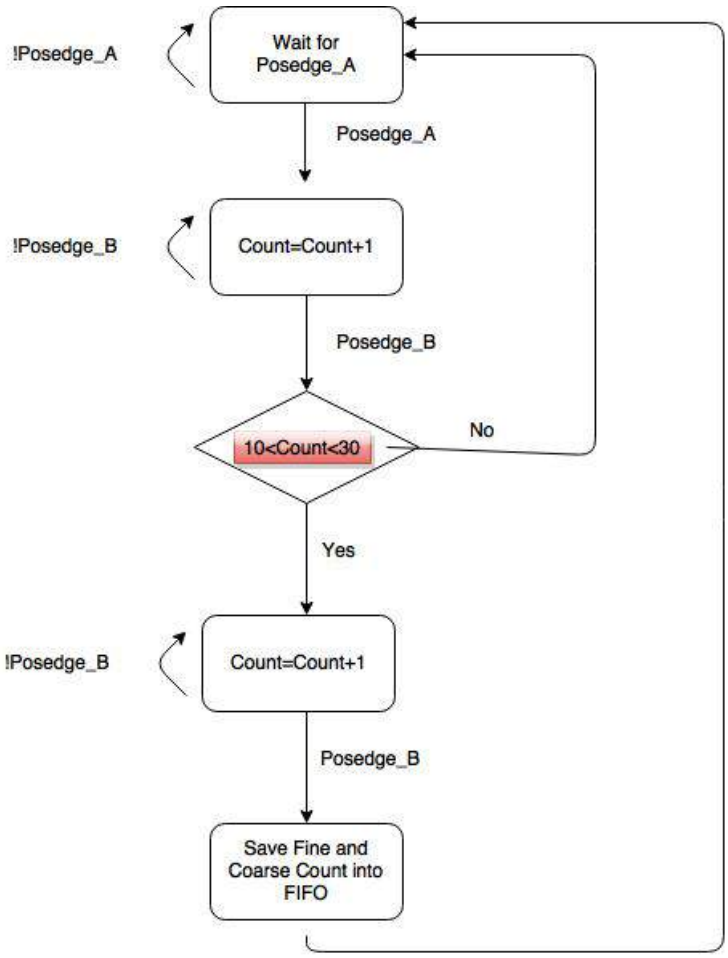


Figure 4.18: The revised ASM with the condition on the arrival of B(START) pulse

(a) PMT A pulse corresponds to a real muon capture in the B pulse. This would ensure that a B pulse arrives definitely 50-100 ns after the A pulse.

(b) PMT A pulse is caused by another muon, noise, or afterpulsing. This means that the PMT B pulse won't be corelated in time to the A pulse. This corelation is checked through the 50-100 ns range.

To summarise, if a PMT A pulse is followed by a PMT B pulse after 50-100 ns, there is little chance that it's a false event. The state machine moves to the next step.

3. So we just need to change State 2 in the original algorithm. When an A posedge arrives, the FPGA starts a counter running on 300 MHz (3.333 ns). If a B pulse arrives and the count is less than 30 i.e.($100ns$), the system moves to State 3, otherwise it remains in State 2. We can also add a lower threshold of 30 ns. If a B pulse arrives earlier than 30ns, the system remains in State 2.

4. The VETO function of C is performed by an AND gate between A and C pulses. To properly register VETO in all cases, the VETO output should be at least 3.333 ns long i.e. the pulses should coincide at least for 3.333ns. The width of the A pulse should be empirically adjusted and observed on the oscilloscope such that it overlaps with the coincident C pulse for at least this amount of time.

The new Algorithmic State Machine (ASM) is shown in Fig.4.18

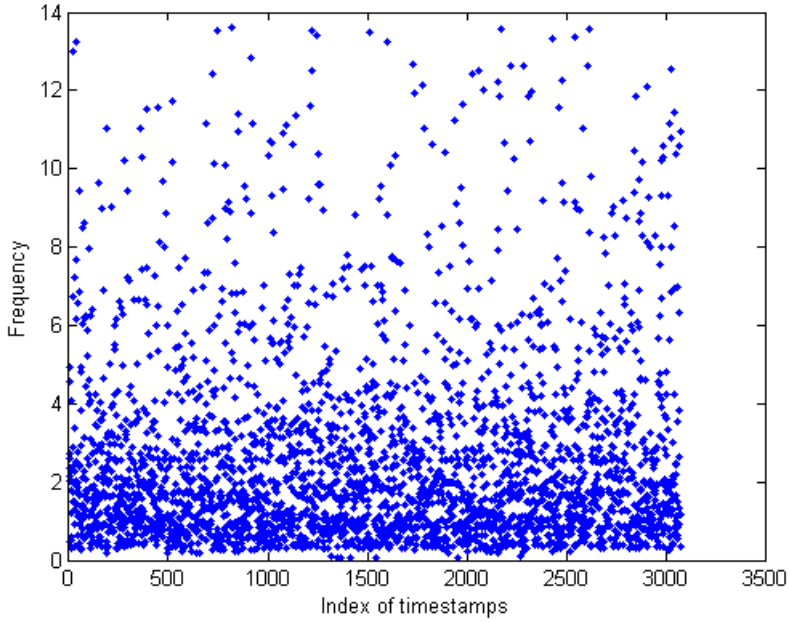


Figure 4.19: Raw timestamps from the FIFO buffer for the logic unit implemented on the FPGA

4.3.3 Results of the Logic Unit implementation on the FPGA

The time intervals from the FIFO buffer for 3 runs are given in Fig.4.19. The corresponding histogram plot is shown in Fig.4.20. The graphs shown are similar in nature to the ones previously shown with the logic units external to the FPGA. The estimate for the lifetime parameter is 2.167μs with a 95% confidence interval of $(1.869 - 2.465)\mu$ s). Alternatively, we have $2.167 \pm 0.15\mu s$, assuming normal distribution for the standard error. The goodness of fit parameters are shown table 4.3.3. The high sum of squared errors (SSE) is due to the data points on the left side of the distribution. A small deviation of, say 30 counts, can contribute 900 to the SSE. However, the R-square and adjusted R-square values are close to unity, indicating a good fit.

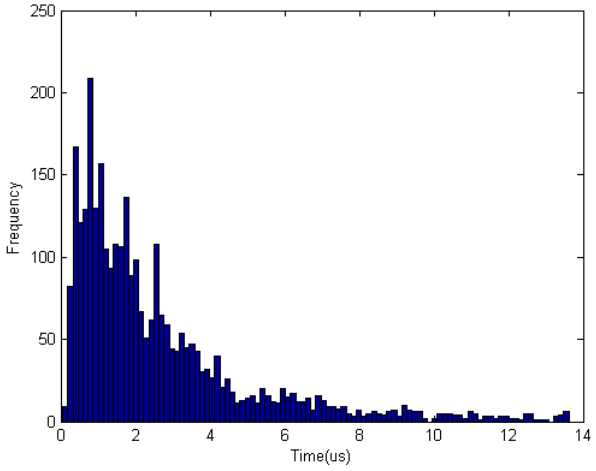


Figure 4.20: Histogram for logic unit and the TDC implemented on the FPGA

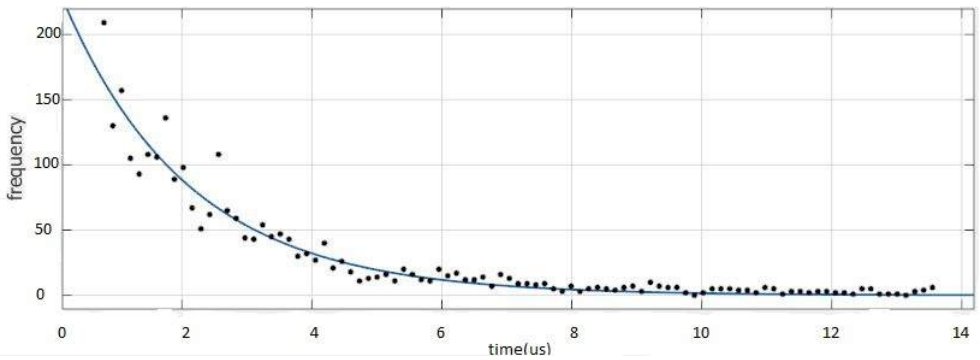


Figure 4.21: Fit Data vs Experimental timestamps

Parameter	**Value**
Sum of Squared Errors(SSE)	10090
R-square	0.9318
Adjusted R-square	0.9311
Root mean square error(RMSE)	10.42

4.4 Sources of Noise and Errors

The main source of Noise is the B PMT which exhibits lots of after pulsing and that is why we dont measure the time intervals from $0 - 40ns$ due to the PMT B Noise. These counts come at a very high rate and fill the FPGA buffer in around 20 minutes. The major cause is the high dark current of the PMTs at room temperature.
To quantify the effect of the PMT B noise, we tried to measure the probability of observing a B pulse later at a time t, once a B pulse was observed. PMT's A and C were turned off and the signals from PMT B were fed into the TDC, which recorded timestamps on the fly,as discussed in Chapter 2. 40,000 readings were taken and the successive difference between two B PMT pulses were calculated. These errors were then histogrammed. The histogram is a sharply spiked narrow gaussian curve with a mean of 13 ns and a standard deviation of 8 ns. Thus, there was negligible probability of finding a noisy B pulse 40 ns after a B2 pulse, that corresponds to a muon capture, was detected. Therefore, the statistical effect of PMT B noise is negligible.[23] Although we did not observe the FPGA temperature to rise after running for 10 hours, temperature variations are bound to occur, and that might also account for some statistical uncertainty, due to variation in wire delays and inherent TDC non-linearity. The setup runs on a UPS, so whenever the power shifts from line to battery or vice versa, the PMTs experience a current surge, which might also register false counts. All the above effects can be statistically avoided by taking a larger set of data.

4.5 Conclusion

The lifetime experiment conducted with the FPGA based TDC yields results that are comparable with those obtained using conventional TAC's. Increasing the resolution of the TDC would not significantly reduce the error in the results. To reduce the errors, one needs to characterise the DNL of the TDC. Also, there are certain peaks that appear consistently in the data that need to be better understood. This in turn depends on including the input-output responses of the discriminators, logic units and NIM to TTL converters, and analysing the signal path. Improvements in the algorithm can also help reduce noise and false detections.
The next step in the experiment would be to conduct a muon velocity experiment. Another pos-

sible dimension is to shift the TDC to low-end micro-controllers like Arduino, PIC etc. Besides lifetime measurements, the TDC can be used for other coincidence detection experiments and time-of-flight measurements like downconverted photons or fluorescence spectroscopy. These might require increasing the TDC resolution and possible use of other TDC methods discussed in Chapter 2.

Bibliography

[1] H.Underwood, Journal of Comparative Physiology **125**, 143 (1978).

[2] S. K. et al., Opt. Express **13**, 1249 (2005).

[3] M. Dolnik, I. Schreiber, and M. Marek, Physics Letters A **100**, 316 (1984).

[4] Sedra and Smith, *Microelectronic Circuits*, Oxford University Press, 2004.

[5] Y. C. O. Kwon and Y. Kim, Phys. Rev. A **53825** (2008).

[6] J. w. Pan et al., Rev. Mod. Phys. , 777 (2012).

[7] K. W. et al., Analytical Chemistry **74**, 5342 (2002).

[8] V. S. et al., The European Physical Journal D **5**, 97 (1999).

[9] J. Kalisz, Metrologia **41**, 17 (2004).

[10] L. Iafolla, Nuclear Instruments and Methods in Physics Research A **739**, 75 (2014).

[11] C. Favi and E. Charbon, FPGA **1**, 1 (2009).

[12] Xilinx, Virtex-5 f. p. g. a. data sheet:dc and switching characteristics, Data Sheet D. S. 202(v5.4), Xilinx, 2014.

[13] Understanding data converters, Application Report SLAA013, Texas Instruments, 1995.

[14] M. J. Loinaz and B. A. Wooley, IEEE Journal of Solid State Circuits **29** (1994).

[15] N. Sawyer, Data to clock phase alignment, Application Note XAPP225 (v1.3), Xilinx XAPP225 (v1.3), 2009.

[16] C. Wellheuser, Metastability performance of clocked fifos, Application Report SCZA004A, Texas Instruments, 1996.

[17] L. Liu and P. Solis, The speed and lifetime of cosmic ray muons, 2007.

[18] Measurement of muon properties in the advanced students laboratory", Lab manual, Hiedelberg University.

[19] Villasenor, AIP Conference Proceedings **1026** (2008).

[20] T. Gorda, M. Russo, and J. Sloane, The proper lifetime of a muon, Lab Manual,Department of Physics, Rutgers University, 2010.

[21] P. Singh and H. Hedgeland, Physics Education **50.3**, 317 (2015).

[22] U. Latif, Measuring muon lifetime and muon velocity, Master's thesis, Physics Department,Lahore University of Management Sciences, 2014.

[23] Photomultipler tubes r1307, Datasheet, Hamamatsu, 1998.