Calculating the mean lifetime of muons using Teachspin apparatus and FPGA programming- Revisited

Dibya Bharati Pradhan[1]

[1] *School of Physical Sciences, National Institute of Science Education and Research, HBNI, Jatni-752050, India*

(Dated: November 27, 2022)

Abstract: Muons are charged leptons just like electrons but with a much greater mass. In this experiment, several cosmic ray muons are stopped each minute in a large volume of scintillator. The first part of this experiment used a Teachspin apparatus which consists of a scintillator detctor, PMT and an analyzer circuit to study the muon decay. By analyzing the spectrum of time delays between the entry of muon into the scintillator and the decay of muon, the mean lifetime was calculated. It used in-built FPGA inside the Teachspin apparatus and the muon physics software to calculate the muon lifetime values. In this part of the experiment, however, we study the signals without using Muon Physics software. We connect an 'Altera Cyclone IV FPGA' externally to detect the decay signals by using VHDL programming language in Quartus Prime software. This was quite challenging. It required identifying and distinguishing muon decay peaks from consecutive muon detection peaks and measuring the differences in the timescale of the peaks to obtain the decay. Exporting and saving the mean lifetime values calculated in the code algorithm in a separate file also becomes challenging when using an external chip. We successfully designed the code to detect decay pulses from sample signals either hard-coded or fed in from picoscope, but detecting decay peaks from very high frequency signals from the scintillation detector and creating text files to save muon lifetime values still remains incomplete due to the limitations of apparatus and softwares used for this part of the experiment.

INTRODUCTION

Muon is a truly elementary particle. It has a unitary negative electric charge of -1 and a spin of 1/2 and a much greater mass (105 MeV/c2). The antiparticle of muon is μ^+ , with a charge state +1. Muon decay is of great importance for the study of weak interactions which is one of the fundamental forces in nature. The muon mean lifetime value is one of the most precisely known constants. With advancement for methodology and techniques, it is now measured with a precision of a few parts per million (ppm). Low-energy muons are primarily created from decays of particles which are produced in the interactions of high-energy cosmic-rays with the atmosphere of earth. About 80 % of the cosmic rays at the sea level are muons, positively or negatively charged. They have a mean lifetime of 2.197 microseconds. Laboratory experiments show that muons have much smaller lifetime than the time it takes to reach the surface of the earth. However, the fact that muons are detected on Earth surface indicates the effects of special relativity i.e their decay is slowed down since they travel at very high speeds (time dilation).

In the first report of the experiment, we saw that a muon that enters, stops, and further decays in a scintillator produces two pulses separated by a few μs. One pulse observed when it enters the liquid, and a second one when it decays in the detector. The pulses are viewed with an oscilloscope or digitized and stored on the computer. The time difference between these correlated pulses is to be measured. After many such observations, a distribution of the number of counts vs. time difference is plot.The plot is an exponential decay curve, whose decay constant is the lifetime of the muon.

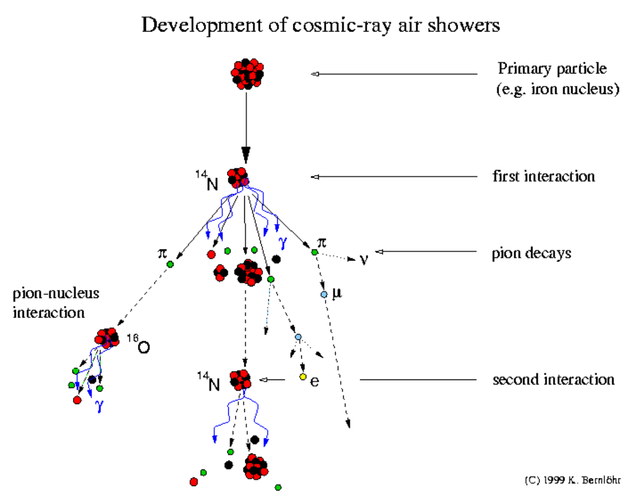

FIG. 1. *Production of muons in the higher atmosphere due to cosmic ray interaction.* [Source: Cosmic ray air showers]

Another method to get the decay counts and detect muons is by using an FPGA coupled with scintillator detector signals.

Here, our goal is to study the same phenomenon, i.e.,to calculate the mean lifetime of muons using an external FPGA, but by bypassing the Teachspin algorithm for recognizing muon decays and excluding the use of Muon physics software. This requires developing an algorithm for detecting muons and decay peaks directly from the scintillation detector discriminator or amplifier pulses and calculating the time difference between the detection and decay peaks of the signal as the mean lifetime. We achieved these goals by dividing this experiment to different parts as described in the methodology.

THEORY

Muon Formation

The major component of cosmic rays are protons and heavier nuclei (about 98%) and a very small amount of electrons(around 2%). When the cosmic ray collide with the nuclei of air molecules, secondary particles such as neutrons, pions (both charged and neutral),protons, kaons, photons, electrons and positrons are produced . These secondary particles undergo electromagnetic and nuclear interactions to produce yet additional particles. In this cascade process, muons are produced. Muons are elementary particles similar to electrons,but with much greater masses. Muons are classified as leptons and are produced due to the decay of pions in weak interaction. The physical quantities of muons are as follows:

Mass- $206.78\ m_e$
charge- $\pm e$
spin- $\frac{1}{2}$

Muon Decay

Muons decay into an electron and two neutrinos. For muons, the neutrinos are electron anti-neutrino and muon neutrino. Anti-muons, in mirror fashion, most often decay to the corresponding antiparticles: a positron, an electron neutrino, and a muon antineutrino. Neutrinos are produced due to lepton number conservation. The ratio of positive to negative muons is also not unity.This is due to the difference in the number of positive and negative pions in the atmosphere. Muon decay equations are as follows:

$$\mu^+ \to e^+ + \nu_e + \bar{\nu}_\mu$$
$$\mu^- \to e^- + \bar{\nu}_e + \nu_\mu$$

Muon Lifetime

The theoretical mean life time of muon is 2.197 μs.

For the decay, we can write;

$$N(t) = N_0\ exp(-\lambda t)$$

where N(t) is the number of surviving muons at some time t and N_0 is the number of muons at t = 0. Here, λ is called the “decay rate” that characterizes how rapidly a muon decays.

Say, D(t)dt is the time-dependent probability that a muon decays in the time interval between t and t + dt. If we had started with N_0 muons, then the fraction -dN/N_0 that would on average decay in the time interval between t and t + dt is just given by:(differentiating the above relation)

$$-dN = N_0\lambda\ exp(-\lambda t)\, dt$$
$$\frac{-dN}{N_0} = \lambda\ exp(-\lambda t)\, dt$$

Thus

$$D(t) = \lambda\ exp(-\lambda t)$$

This equation holds irrespective of the starting value of N_0 i.e. the distribution of decay times, for new muons entering our detector, is also exponential with the very same exponent used to describe the surviving population of muons.

The ”lifetime” τ of a muon is given by,

$$\tau = \frac{1}{\lambda}$$

Some muons completely pass through the detector. This generates a single detector output pulse. Other muons are stopped inside dectector and decay into electron or positron. The later event gives two consecutive pulse. The time interval between two consecutive pulse of this second type of event gives the lifetime of the decay.

The distance travelled from the upper atmosphere to reach the surface of the earth takes more time than their lifetime. So, Muons should not reach the ground even with the speed of light. But in reality, we observe muons on Earth’s surface. To explain this, we assume the velocity of muons to be around 0.9999c. Due to great speed of muon, the length that the muon travels in its frame would be contracted by γ (Lorentz factor). Thus, the muons would be able to reach the ground as per special relativity.
The observed muon lifetime obtained by us in the previous report of muon experiment is 2.122 μs.

Impedance matching

The connections from output of Discriminator and amplifier to oscilloscope are made through 50 ohm Coaxial Cable.Since the square pulses from discriminator are DC signals. The oscilloscope should be DC coupled with input impedance set to 50 ohm.Otherwise Signal loss and distortion due to reflection was observed. The reflection coefficient is given by:

$$R = \frac{Z_L - Z_0}{Z_L + Z_0}$$

If Z_L becomes very large compared to Z_0 then R $\approx$ 1. Then Signal is mostly reflected and and we get negligible transmitted signal.

Detector physics

The detector is a plastic scintillator. The active volume of the detector is in the shape of a right circular cylinder of dimensions 15 cm x 12.5 cm put at the bottom of the black anodized tube of aluminum alloy. Plastic scintillator is transparent organic material. It is made by mixing together one or more fluors and a solid plastic solvent having an aromatic ring structure. When a charged particle passes through the scintillator, it will lose some of its kinetic energy by ionization and atomic excitation of the solvent molecules. A part of this deposited energy is then transferred to the fluor molecules. The electrons of the fluor molecules are then excited to higher states. During radiative de-excitation, there is emission of light in the blue and near-UV portion of the electromagnetic spectrum. This has a typical decay time of a few nanoseconds.

To measure the muon's lifetime, we are only considering those muons that enter, slow, stop and then decay inside the plastic scintillator.

Such muons have a total energy of only about 160 MeV while they enter the tube. When a muon slows to a stop, the excited scintillator emits light. This light is detected by a photomultiplier tube (PMT), eventually producing a logic signal which triggers a timing clock.

After a muon stops, it decays into an electron, a neutrino and an anti-neutrino. Since an electron is much lighter than a muon, $m_\mu/m_e \approx 210$, the electron tends to be very energetic and produces scintillator light all along its pathlength. Some of the muon's total energy is also shared by the neutrino and anti-neutrino but they entirely escape detection. This second burst of scintillator light is also seen by the PMT which is used to trigger the timing clock. The muon lifetime is measured by the the distribution of time intervals between successive clock triggers for a set of muon decays.

Teachspin Apparatus

The electronics module contains all the electronics needed to run the experiment. Connections on the front panel helped examine the PMT signal and monitor that signal as it moves along the readout chain. First, the PMT pulses are amplified. Then they are compared against an adjustable threshold. Pulses that are above the threshold voltage are sent to timing circuitry implemented in a FPGA (field programmable gate array) chip. As soon as the first flash of the scintillator is received, the timing system starts. Now, if a second flash occurs within 20 microseconds of the first flash, the readout electronics measures the time interval between the two flashes. This data is then passed onto the lifetime display software.

If there is no second flash within 20 microseconds of the first one, the software simply records the pulse as a charged particle that has passed through the detector. Furthermore, the Communication circuitry transfers the data to a PC or laptop through either a serial or a USB port.

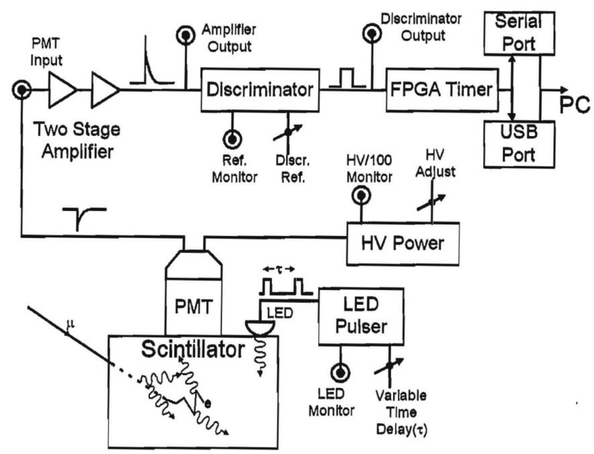

FIG. 2. *Block diagram of the readout electronics. The amplifier and discriminator outputs are present on the front panel of the electronics box. The HV supply is inside the detector tube.*

FPGA

Field Programmable Gate Arrays (FPGAs) are semiconductor ICs where a large majority of the electrical functionality inside the device can be modified for specific use. They are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. For the FPGA to function, the user provides a design in a hardware description language (HDL) or as a schematic design. In this experiment, we have used Altera Cyclone V FPGA board and VHDL as the HDL language. We have used intel Quartus softaware to complile and run the code. For the hardware, we used USB Blaster to successfully impliment the VHDL code on the FPGA board. We carried out basic combinational and sequential circuits. We implemented OR gate, AND gate and D flip-flop circuits on FPGA. The codes for all these are attached in the appendix. The diagram of pin planner of Quartus prime software can be found in figure **??**.

METHODOLOGY

In the first report of the Muon lifetime experiment , we measured the muon lifetime using Teachspin apparatus and Muon physics software. Then we learnt how to implement gates using FPGA and Quartus Prime software.

In this part, we aim to program the FPGA to detect decay pulses of muons from the scintillation detector and use the amplified output signal from the teachspin box to obtain the number of muon decays and store their decay timescales in a separate file bypassing the whole inbuilt decay mechanism of the teachspin.

This has been done in the following steps
First, the characteristic properties of the signal from the discriminator are noted. Next, an algorithm for detecting the decay of muons is discussed. We further make an inline muon decay simulation in the same VHDL file to test the algorithm. The whole setup is then modified and enhanced to work for an arbitrary external signal produced from picoscope simulating a muon decay. The estimated data saving mechanism is finally discussed.

The characteristic properties of the whole algorithm are the pulse height, pulse width, the rising and falling time and the time difference of the pulses. In principle, the pulses should be sharp peaks. The pulse height needs to be high (¿500 mV) with a narrow width (≈picoseconds) and high slew rate. However, the pulses detected by the scintillation detector have a peak of about 25 mV with a pulse width (≈ 2 nanoseconds) and very high slew rate.

The main algorithm behind the sections below are as follows:
A muon decays into an electron, an electron antineutrino and a muon neutrino. When a muon hits the detector, a pulse is generated which can be seen on the oscilloscope. After a few micro-seconds, it decays into an electron which again hits the detector and a second pulse is generated. If the time taken between these two pulses is below a certain reference (say 2.5 microseconds) then the pulses are counted as a single muon decay. If the time exceeds, then it is considered as just two different muons/electrons hitting the detector and no decay is counted.

Muon decay using a sample signal hard-coded using VHDL programming

In this part, we design a digital signal by coding through VHDL programming. Here, to detect coincidence counts/peak, we use the following algorithm,
If the time difference between two consecutive peaks is less than a given timescale, they are considered as detection and decay peak while a difference between pulses exceeding that time-limit does not contribute to decay.
We program FPGA such that, two LEDs activate as a switch is turned on. The 2nd LED glows only when it detects a decay pulse and the first LED glows for all the pulses in the signal. We assign the LEDs and switch referring to the PIN-planner provided in the Altera FPGA manual.
The algorithm is implemented using an in-built clock of FPGA. The clock has a frequency of 50 MHz. It is used as a counter to count the time for the signal.

The idea behind the code is as follows:
A variable is assigned to the 50MHz clock. The clock has 50,000,000 cycles each second. The variable is set to iterate after every 100,000,000 cycles = 2 seconds. So, 0.2 seconds would correspond to 10,000,000 cycles. The sample pulse we are trying to generate, will repeat after every 2 seconds.

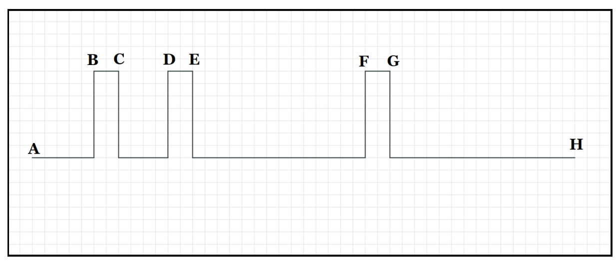

FIG. 3. *Generated signal for muon decay. The first pair of peaks account for a detected muon and a decayed muon respectivly, while the last peak accounts only for a detected muon.*

We start the pulse at A (at 0). As shown in figure 3, we generate three peaks of the signal with equal thickness BC (from 10,000,000 to 15,000,000), DE (from 25,000,000 to 30,000,000) and FG (from 60,000,000 to 65,000,000). All peaks have predefined fixed pulse width and time difference. The signal is generated such that, the first two peaks BC and DE lie close to each other within a time scale of 25,000,000 cycles. In other words, we are trying to make the 2nd peak as our decayed muon peak while the first peak is the detected muon and the third pulse is not a decay.
For an actual muon decay, the decay time should be set at around 112 counts = 2.24 seconds.

This signal is then passed to a output which works as follows. Two LEDs are assigned to blink for the obtained peaks in a regular fashion. LED 1 blinks at every peak detected while LED 2 blinks at decayed peaks. The bluntness the pulses gives an estimate of the time for which the LEDs glow. Additing to that, we also assign three separate LEDs to blink for the three different pulses. This was done to make sure we observe each signal separately.

A small green LED is blinks after every 2 seconds to indicate the end of each cycle of signal. After every 2 seconds, the signal starts its next cycle and we observe six LEDs blinking on the FPGA board. We are not yet counting the decayed peaks. Counting will be implemented in the later parts.

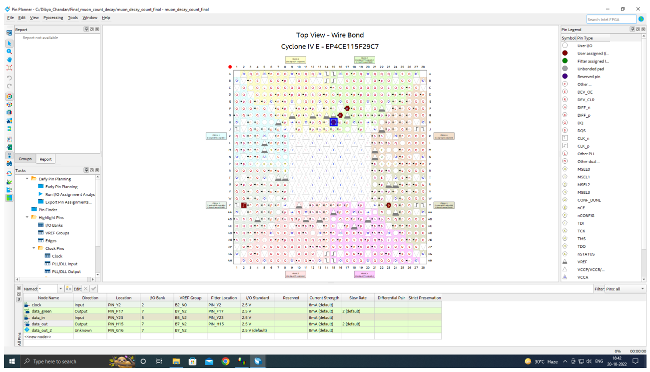

FIG. 4. *Pin planner of Quartus Prime software for muon count using arbitrary signal.*

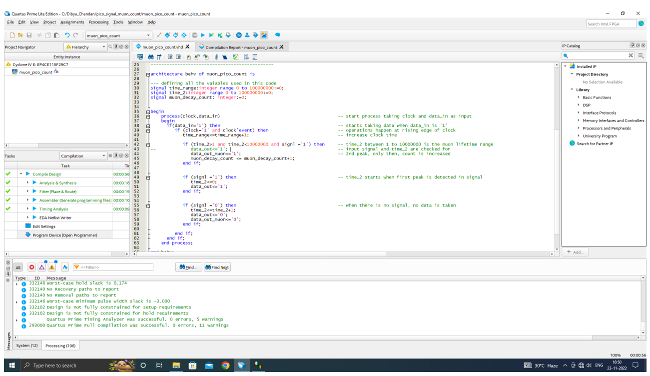

FIG. 5. *Code interface of Quartus Prime software for muon count using arbitrary periodic signal.*

Muon count using signal generated from Picoscope

The code is attached in the appendix.
In the last part, we hard-coded a periodic signal and used it to count the coincidence peaks. In practice however, signals are arbitrary and has to be feeded from the source to the device for counting the decays. Muon decays are random. They occur with multiple frequencies and energies.
In this part, we design the decay algorithm to feed a generated signal from a picoscope to FPGA via a 14-pin GPIO (General Purpose Input Output) pin set.

Picoscope:
A picoscope can work both as a function generator and an oscilloscope. To generate pulses, the AWG (Arbitrary Waveguide Generator) mode of the picoscope is used as it can generate the signals of shape and size as specified by the user. To act as an oscilloscope, the signal can be fed to channel A or channel B and view it on the computer screen using picoscope software. We design the signal as shown in figure 6 using AWG.
Just like the previous part, if two consecutive peaks have a time difference less than a given timescale, they are considered as decay peaks while exceeding that time does not contribute to decay.

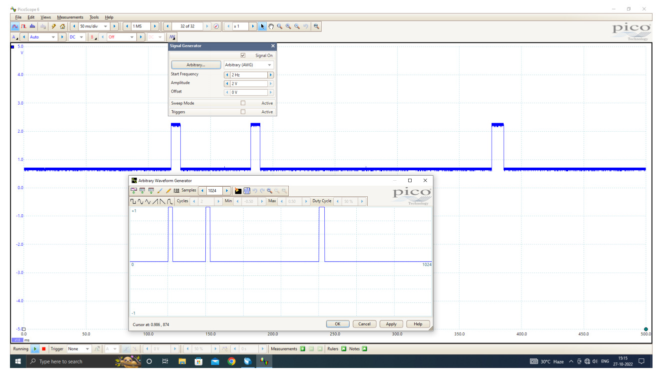

FIG. 6. *Sample signal generated using the AWG mode of picoscope. The voltage and amplitude are set to 2 V and 2 Hz respectively*

The idea behind the code is as follows:
The in-built 50 MHz clock is used as a counter to count the time for the signal. Two variables are assigned to the clock. One is the time counter for the increasing clock signal cycles and the other to calculate the time difference between two peaks from the picoscope signal . The threshold timescale for detecting two pulses to be considered as a decay is 10,000,000 cycles according to the 50 MHz clock. We set the frequency, shape and size of the pulse using AWG mode of the picoscope to be fed onto FPGA. We model the simplest case with a sample signal same as shown in figure 3. We set frequency of 1 Hz and an amplitude of 1 V.
The two variables count the number of total pulses and total number of decay pulses respectively. However, we could not find any mechanism to display those counts to the user on the pc screen using Quartus. We could only detect dacays by getting the LEDs blink. There is also no data saving implemented in this part.

Then the FPGA is programmed to blink LED-1 when any peak is detected and LED-2 when a decay is observed. We refer to the PIN-planner provided in the Altera FPGA manual.

Comparator coupled with detector signal

As mentioned previously, the teachspin apparatus has an inbuilt comparator (discriminator) to detect the signals above the set threshold and give a square wave of certain width and amplitude.
Instead of using the muon software to get the muon lifetimes, we tried to implement Altera FPGA to detect the decay signals directly from the teachspin output.
However, the circuit does not work properly. The typical pulse height required for the LEDs on the FPGA to glow is ¿500 mV, but the pulses obtained from the discriminator of teachspin are $\approx$ 25 mV. The pulse is not square pulse, rather it looks like that shown in figure 8
To account for this difference, we built an external fast comparator that gives a square pulse for every peak detected from the discriminator pulse.

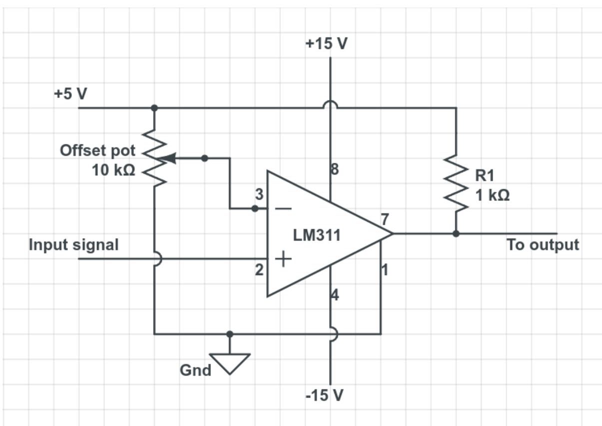

FIG. 7. *Circuit diagram of the high slew rate comparator circuit.*

We use 'LM311' comparator for this purpose. These comparators have much higher slew rate of about 18V/μs than common LM741 comparators having a slew rate of 18 V/μs.
Fast comparators are required because the slower ones will not give a sharp peak, rather they will rise and fall with a very wide delay or get distorted.
The circuit for the comparator is shown in 7. The comparator circuit involves a 10k potentiometer which is used to set the threshold voltage.

The detector is connected to the teachspin apparatus for the discriminator component. The output signal is then fed to the fast comparator circuit. The threshold of the potentiometer is set to some arbitrary value to check if the circuit is working properly. A function generator is used to first check the working of the circuit using an oscilloscope to plot. The frequency is then increased gradually as high as possible.

Our ultimate aim is to make a circuit that can detect two pulses within a certain time-scale and save the time difference between the two counts in a data file along with the number of counts. The decay counter code was already developed in the previous section. All we need is to feed in the pulses from the detector to the comparator, use the output square pulse from the comparator to detect muon decay and compute muon lifetime using FPGA and make a data saving mechanism to store the time difference between the decay pulse and detected pulse as the mean lifetime of muon.

Trial for saving data

In the 1st report, Teachspin apparatus combined with muon software not only counted the muons decays, but was also designed to store the data in a separate text file. But the FPGA we use has to be programmed for it to store the data from the output of the code we developed. Apparently, there is no way one can use the Altera FPGA alone to save the data to a separate text file. This could have been possible had we used Xilinx's FPGA. One possible way to do so using Altera and Quartus includes using the FIFO module of Quartus and saving the data externally using LabVIEW software. However, this part of the experiment is incomplete and hence, unsuccessful.
The procedure is as follows:
The output from the VHDL code data is saved into the small size RAM onboard the FPGA. FIFO, which stands for (First-In-First-Out), is used for buffering data before sending it off chip. It is also used for storing data for later processing. However, FIFO reads data once and sends the data to LabVIEW. LabVIEW is a graphical programming environment that engineers use to develop automated research, validation, and production test systems. It takes the data one by one from the FIFO and passes it to the file reader module.
In other words, LabVIEW creates a new text file every time our code is running and then saves the data line-by-line. The final file obtained contains all the data in a new text file. This file can give us the decay time-scales. We can analyze the data in this file for further analysis using some other software (python, in our case).

OBSERVATION AND RESULTS

Muon count using signal generated from Picoscope

Two LEDs are observed to blink at regular intervals. One blinking for each peak encountered in the pulse, and other for each decayed peak. Though the signal fed to the FPGA at this stage is periodic, but the code is capable of handling any random signal with any pulse frequency and an amplitude ¿0.5 V. We can change the frequency and amplitude of the input signal from the AWG to any arbitrary value and with the amplitude range 0-5 V, but for human eye to visibly distinguish the LED blinks, we

FIG. 8. *Image of discriminator output from Teachspin Box.*

set it to 1 Hz with a voltage more than 0.5 V. Picoscope being a small device has limited usage. It undergoes distortion when too high or too low frequencies are applied. The amplitude of the voltage also fluctuates. So, an external function generator can be used in place of Picoscope to provide higher frequencies. But for other frequency ranges, one needs to modify the threshold time under which two consecutive pulses are termed as one decay pulse as per our algorithm. We test the signal for higher frequencies up to 100 KHz using an external signal generator and oscilloscope. We did not observe any discrepancies in the output for designed algorithm and code. Hence, the code is well capable to work finely for random signals from detector.

Comparator coupled with detector signal

The high voltage supply required for the functioning of the PMT was supplied by connecting the detector to a high voltage supply. The **"Geco"** software was used to adjust the voltage value. For this experiment, the HV supply for PMT tube was kept at constant 700≈ V.
The comparator circuit is designed in such a way that it gives a peak applied voltage of 5V when the signal from the function generator is above the threshold voltage. We control the threshold voltage using the potentiometer. This allows only the voltages above a set voltage to be directed to the output pulse.
Ideally, it should be set about 15-20 mV for this experiment because the discriminator gives an output voltage of ≈ 25mV and the signal is very prone to noise below this voltage.
If the frequency of the function generator is changed, then it would also be required to change the time difference under which two consecutive peaks are considered to have decayed. Ideally, the frquency should be kept in the order of picoseconds. but the FPGA in-built clock limits the frequency to 50 MHz and the function generator input pulse also starts to distort beyond 100 KHz. Hence, it is not possible to reach the timescale of decay from the scintillation detector and the discriminator with the provided apparatus. This would required even more efficient set-up.
The input signal and comparator output can be seen in figure 9

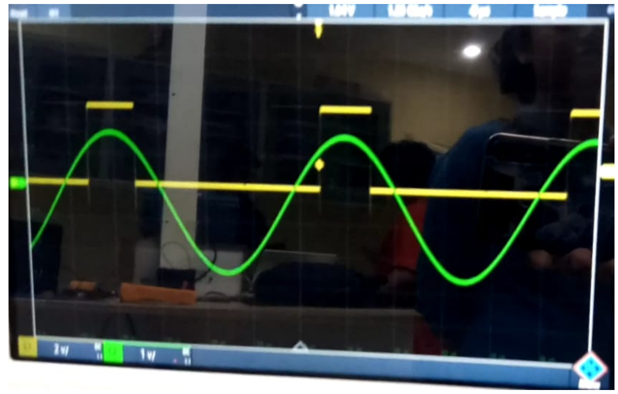

FIG. 9. *Input and output signals from the fast comparator. The green curve is the input sine curve with a frequency of 1 kHz while the yellow square pulse is the comparator output. The comparator gives a maximum voltage of signal only when the input signal is above the threshold voltage.*

Trial for saving data

We could not complete this part due to the limited time during the semester. We started to write the code file for FIFO but it was a very long code and required much more time to figure out the errors and finish the code. This would have helped communicate the FPGA with LabVIEW for saving data which could not be properly established.
However, it should be noted that, the procedure mentioned for this part is still a speculation. we can not confirm it until we successfully write a code to produce the data files.

CONCLUSION

Muon is one of the elementary particles and a very clean probe for the study of weak interaction. The muon lifetime experiment is considered to be one of the standard experiments today. In the first part of the report, we used Muon physics software that was pre-programmed to give us the mean lifetime values directly. Analysing the data, we showed that the results agree with the experimentally reported values of muon mean lifetime in the literature and the FPGA module showed expected results for the basic gates.

For this part of the project, we tried to detect muon decay from the detector signals without using the muon software. This was quite a challenge for us. We made our own code on Quartus Prime software in VHDL language to detect muon decay signals using FPGA. However, this code was only successful for sample signals hardcoded or generated from the picoscope. We could not successfully implement it on the actual detector signal because of the limitations on FPGA to detect very high freqency and low magnitude signals. We made a fast comparator to try resolve this issue. The goal was to connect fast and sharp muon pulses from the scintillator detector to FPGA and obtain the decay pulses. Though the circuit worked fine for signals from function generator yet the purpose failed since the FPGA built in clock has frequency of 50MHz while the function generator starts getting distorted beyond 100 KHz. This experiment would require FPGA clock to have frequencies in GHz or THz.

Apart from this, saving the timescale of decay pulses using Quartus or Labview imposed some difficulties due to the lack of information on the same. We expect that saving the data in text file somewhere in memory would require writing several code lines using VHDL in FIFO file and connecting the FIFO to LabVIEW. This particular part was left unexplored due to the lack of sources, information and time.

ACKNOWLEDGEMENTS

I would like to thank Dr. G. Santosh Babu and Dr. Varchashwi Kashyap for constantly guiding and supporting us throughout the experiment. I also show my appreciation to my lab-mate Chandan Kumar Sahu for his enthusiasm and fresh perspectives on the experiments conducted. I would also thank the lab technicians for helping to setup the apparatus wherever needed.

I. REFERENCES

[1] Muon Physics Teachspin manual

[2] Gorda, T., Russo, M., & Sloane, J. (2010). The Proper Lifetime of a Muon. *Lab Manual, Department of Physics, Rutgers University.*

[3] Particle Data Group

[4] Particle Data Group - Wikipedia

[5] Reiter, R. A., Romanowski, T. A., Sutton, R. B., & Chidley, B. G. (1960). Precise Measurements of the Mean Lives of $\mu+$ and $\mu-$ Mesons in Carbon. *Physical Review Letters*, 5(1), 22.

[6] DE2-115 FPGA User manual

[7] Cyclone IV GX FPGA Development Kit

Appendix A: Code for Muon decay using a sample signal hard-coded in VHDL language

```
------------------------------------------------------------------------------------------
--- Muon decay simulation VHDL Code
------------------------------------------------------------------------------------------

library ieee ;
use ieee.std_logic_1164.all;
use work.all;
------------------------------------------------------------------------------------------

entity muon_decay_count_final is
port( data_in: in std_logic;
    clock: in std_logic;
    data_out: out std_logic;
    data_out_2: out std_logic;
    data_green: out std_logic;
    data_pulse_1: out std_logic;
    data_pulse_2: out std_logic;
    data_pulse_3: out std_logic
);
end muon_decay_count_final;
------------------------------------------------------------------------------------------

architecture behv of muon_decay_count_final is

signal time_range:integer range 0 to 3*100000000:=0;
signal key:std_logic:='0';
signal count: integer:=0;
signal var: integer:=0;
signal muon_count: integer:=0;

begin
    process(clock,data_in)
    begin
        if(data_in='1') then
            if (clock='1' and clock'event) then
                time_range<=time_range+1;

                if (time_range =3*10000000) then
                    if (key ='1') then
                        data_out_2<='1';
                        muon_count<=muon_count+1;
                    end if;
                    data_out<='1';
                    data_pulse_1<='1';
                    var<=var+1;
                    key<='1';
                end if;
                if (time_range =3*15000000) then
                    data_out<='0';
                    data_pulse_1<='0';
                end if;
                if (time_range =3*22500000) then
```

```
                if (key ='1') then
                    data_out_2<='1';
                    muon_count<=muon_count+1;
                end if;
                data_out<='1';
                data_pulse_2<='1';
                var<=var+1;
                key<='1';
            end if;
            if (time_range =3*27500000) then
                data_out<='0';
                data_pulse_2<='0';
            end if;

            if (key='1') then
                count<=count+1;
            end if;
            if (count>3*25000000) then
                count<=0;
                key<='0';
                data_out_2<='0';
            end if;

            if (time_range =3*60000000) then
                if (key ='1') then
                    data_out_2<='1';
                    muon_count<=muon_count+1;
                end if;
                data_out<='1';
                data_pulse_3<='1';
                var<=var+1;
                key<='1';
            end if;
            if (time_range =3*65000000) then
                data_out<='0';
                data_pulse_3<='0';
            end if;

            if(time_range =3*99000000) then
                data_green<='1';
            end if;

            if(time_range =3*100000000)then
                time_range<=0;
                data_out<='0';
                count<=0;
                key<='0';
                data_out_2<='0';
                data_green<='0';
                data_pulse_1<='0';
                data_pulse_2<='0';
                data_pulse_3<='0';
            end if;
        end if;
    end if;
  end process;
end behv;
```

Appendix B: Comparator coupled with detector signals

```
------------------------------------------------------------------------------------------------------------------
--- VHDL code for Muon decay count using an arbitrary signal
------------------------------------------------------------------------------------------------------------------

--- Muon decay count using picoscope signal
--- works for arbitrary signal > 0.8 Hz

library ieee ;
use ieee.std_logic_1164.all;
use work.all;
use ieee.numeric_std.all;

------------------------------------------------------------------------------------------------------------------

--- define all the input pins taken from FPGA
--- these will go into the pin planner

entity muon_pico_count is
port(
    data_in: in std_logic;
    clock: in std_logic;
    data_out: out std_logic;
    data_out_muon: out std_logic;
    signl: in std_logic
);
end muon_pico_count;

------------------------------------------------------------------------------------------------------------------

architecture behv of muon_pico_count is

--- defining all the vaiables used in this code
signal time_range:integer range 0 to 100000000:=0;
signal time_2:integer range 0 to 100000000:=0;
signal muon_decay_count: integer:=0;

begin
    -- start process taking clock and data_in as input
    process(clock,data_in)
    begin
        if(data_in='1') then
            -- starts taking data when data_in is '1'
            -- operations happen at rising edge of clock
            if (clock='1' and clock'event) then
                time_range<=time_range+1;

                -- time_2 between 1 to 10000000 is the muon lifetime range
                -- input signal and time_2 are checked for 2nd peak, only then, count is increased
                if (time_2>1 and time_2<10000000 and signl ='1') then
                    data_out<='1';
                    data_out_muon<='1';
                    muon_decay_count <= muon_decay_count+1;
                end if;
```

```
                -- time_2 starts when first peak is detected in signal
                if (signl ='1') then
                    time_2<=0;
                    data_out<='1';
                end if;

                -- when there is no signal, no data is taken
                if (signl ='0') then
                    time_2<=time_2+1;
                    data_out<='0';
                    data_out_muon<='0';
                end if;

            end if;
        end if;
    end process;
end behv;
```