The Muon Lifetime

Tycho Sleator, Burton Budick, and Andy Haas

Goals

The goal of this experiment is to measure the lifetime of the muon. Positively (and negatively) charged muons incident as cosmic rays are detected and stopped in a slab of aluminum. The positrons and electrons that result from the decay of these muons are also detected. The lifetime of the muon is determined from the distribution of delay times between muon and positron/electron detection.

Background

Positively charged muons ($\mu^+$) are created in the upper atmosphere of Earth from the decay of $\pi$ mesons. Although the lifetime of the $\mu^+$ particle is only around 2 $\mu$sec, due to relativistic time dilation many are present near sea level. (The muon flux at sea level is roughly $10^{-2}$/cm$^2$/ster/sec.) The $\mu^+$ decays according to

$$\mu^+ \rightarrow e^+ + \nu_e + \bar{\nu}_\mu \quad (1)$$

with an exponential decay rate given by

$$N(t) = N_0 e^{-t/\tau}, \quad (2)$$

where $N_0$ is the initial number of particles, $N(t)$ is the number of particles at time $t$, and $\tau$ is the $\mu^+$ lifetime.

In this experiment, you will determine the value of $\tau$ by measuring a distribution of decay events $N(t)$, using coincidence techniques to measure the time delay between the detection of $\mu^+$ and $e^+$ particles. The $\mu^+$ particles will be stopped in a slab of aluminum, where they will decay according to Equation (1). Scintillation detectors will be used to detect both the muons and the positrons; coincidence detection will select mainly the $\mu^+$ decay events. An FPGA circuit board will receive the signals from the scintillation detectors and perform the logic necessary to measure the time from the muon reaching the detectors until it decays. From the distribution of these data, which corresponds to the distribution of time intervals between the $\mu^+$ and $e^+$ detections, you will deduce the muon lifetime.

![Figure 1: Schematic diagram of the photomultiplier tubes.](image-url)
Apparatus

A schematic diagram of the experimental apparatus is shown in Figure 1. A charged particle passing through a scintillation detector is detected by the small flash of light produced in the plastic sheet. This light flash is converted to a pulse of electrons by the photomultiplier tube. The output from each of the photomultipliers is fed to the input of an amplifier/discriminator, which is used to filter out the noise inherent in the detector as well as to amplify the incoming charge pulse. The discriminator channels will fire an output pulse of negative voltage of a fixed size and duration, whenever the input signal (also a negative voltage pulse from the photomultiplier) passes a given threshold. The outputs from the amplifiers/discriminators are then fed into the FPGA board (Figure 2).

To measure only the muons and positrons associated with the muon decay, coincidence detection is used as follows. We wish to keep track of the timing between two events: (a) the detection of an incoming muon that will subsequently decay in the aluminum slab, and (b) the detection of the positron that results from the decay of the muon we detected in (a). We’ll use these two events to ‘start’ and ‘stop’ a clock; it is the distribution of delay times between the ‘start’ and ‘stop’ events that we want to use to determine the muon lifetime. The muons in (a) are the particles that are detected by both the A and B detectors (see Figure 1), but not the C detector, since we’re interested in the muons that stopped before reaching C. The positrons in (b) are the particles that are detected by the C detector. In principle you could also use those detected by the B detector, but that detector just fired during the start signal, so can suffer from photo-multiplier afterpulsing (false pulses that occur sometimes after large pulses). In other words:

\[
\begin{align*}
\text{‘start’ signal:} & \quad (A \text{ AND } B) \text{ AND } \overline{C} \\
\text{‘stop’ signal:} & \quad C
\end{align*}
\]  

For the FPGA board to properly detect each input signal, it must compare the input voltage to a reference voltage. When the input’s voltage is above the reference voltage, the FPGA registers the input as “ON”
The Muon Lifetime

(1), and otherwise “OFF” (0). This is called a “differential input”. (This is needed here because the FPGA can not deal directly with the negative voltage output from the discriminators.) The reference voltage for each input is controlled by turning the little screw on the corresponding variable resistor on the FPGA board, from 0 to -5V. See Figure 2 for details.

The FPGA has two debug outputs, which can be programmed to output various states within the FPGA, so you can see what the FPGA is doing. For instance, you can program debug 141 to be equal to input B, to check that the FPGA is correctly registering input B.

The FPGA also sends data out over a serial (UART) connection, which connects to the computer via a USB adapter. The “serial port utility” program on the computer will gather data from the FPGA in hexadecimal format.

![Figure 3: The output pulse from a discriminator channel for input A (solid yellow), the reference voltage (dashed yellow), and the debug output for input A from the FPGA.](image)

**Procedure**

1. **Adjust the detectors and discriminators**

In order to optimize the signal-to-noise ratio (SNR) of the photomultipliers, you’ll need to adjust the high voltage applied to each of these detectors. As shown schematically in Figure 4, the noise level from a photomultiplier will increase monotonically with voltage, while the signal output will reach a maximum value and then plateau. Thus, the highest SNR is achieved when the voltage is set close to the start of the ‘plateau region’.

Figure 6 shows a schematic diagram of the electronic circuit for the photomultiplier tubes (Philips Model 56AVP). The cathode is held at a large negative voltage relative to the anode (which is held close to zero volts). When a photon from the scintillator strikes the cathode, an electron is released from the cathode (photoelectric effect). An electric field generated by the power supply and resistor chain accelerates this electron towards the first dynode. When this electron strikes the dynode, more electrons are released from
the dynode, which then accelerate toward the next dynode. At each dynode, the number of electrons is multiplied by some factor. By the time the electrons reach the anode, there are a huge number of electrons—up to around 10^6 for each photon striking the cathode.

Use an electron source (supplied) to plateau the detectors: with the threshold level on the discriminator set to 25 mV, the minimum, measure the counting rate as a function of voltage applied. The discriminator thresholds are set by turning a little screw for each channel. The minimum threshold is when the screw is turned fully counter-clockwise. You’ll want to look at the input signal from each photomultiplier on the oscilloscope, as well that channel’s discriminator output, to check that the discriminator is properly firing for each pulse from the photomultiplier above threshold. Use BNC t-junctions.

Start at a high voltage value of -1000 V, and increase the voltage in 50 V increments up to a value of -1500 V. From these measurements you can determine where the start of the plateau region occurs, and thus where to set the HV. Leave the discriminator thresholds at the minimum settings for the rest of the experiment.

Note that in this experiment, you will be detecting events by observing coincidences between detector outputs. Even if there is a large rate of counts due to noise from each detector, if you are looking at coincidences between two detectors (due to, say a muon going through both detectors), the accidental coincidence rate between the output of the two detectors (due to noise) will be much less than the singles rate due to noise from each detector. You can take advantage of this fact to find the optimum signal to noise ratio for coincidence events.

2. Setup the FPGA

This should mainly be setup already, but here are instructions in case not:

1. Attach USB and mini USB ports to the computer
2. Attach BNC cables from discriminator outputs to the proper FPGA inputs (see Figure 2)
3. Hook up BNC cables from FPGA outputs to view the signals on scope
4. Compile and load firmware onto FPGA board (see below)
To reprogram the FPGA:

1. Start Quartus Prime 18.1 Web Edition

2. File >> Open Project >> Desktop/muon_lifetime-trig3_de0nano-firmware/serial1.qpf to view/edit the code and schematics

3. To change parameters of the module, like Ntick, open up serial1.bdf in Files in Project Navigator, and change the parameter of the SyncDelay module

4. To compile everything, press “play” button in toolbar, or Processing Start Compilation

5. To load that new firmware onto the FPGA board, Tools Programmer, in Hardware Setup of that new window choose USB Blaster, mode should be JTAG, then press Start

3. Check the FPGA inputs

Check that the FPGA can properly see each of the 3 input signals from the discriminators. Use the oscilloscope to look at the output from a discriminator, and a debug output from the FPGA set to correspond to that input. You can change what is output on 141 and 142 by sending commands e.g. “00” or “21” in hex over serial, using the serial port utility program. Make sure to have 8 bits, parity None, stop bits 1, and no flow control. The first digit of the byte is the higher 4 bits, which is the O141 output; the second digit is the lower 4 bits, which is the O142 output. When the negative reference voltage level is correctly set between the top and bottom of the discriminated signal voltage (about -780mV), you should see the FPGA debug output fire for every input discriminator pulse, as in Figure 3.

4. Calibrate the FPGA serial output

Program the FPGA such that the serial output corresponds to the correct coincidence logic for the start and stop signals. Make sure that the C input pulse from the discriminator is longer in time than the A and B pulses (why?). Adjust the discriminator pulse lengths to be 100ns for A and B, and 200 ns for C, using the other hidden screw for each discriminator channel. Check them on the oscilloscope.

Start the Serial Port Utility program and connect to prolific COM port (COM21?) to record data from FPGA board (with baud rate to 115200, and display data in hex format). Each byte (2 hex characters) represents the number of “ticks” between a start and stop signal, where each tick is about 80 ns. So “1d” would be 1*16+13=29 ticks, so 29 * 80ns = 2320 ns.

But don’t trust it! Use the double-pulse function generator to generate pairs of pulses that mimic the start and stop signals, and feed those signals into the FPGA instead of from the discriminator outputs. Using a set of known durations between start and stop signals from the pulse generator, calibrate the serial output data collected from the FPGA. How can you determine from this the systematic uncertainty on your calibration?

5. Take data and analyze it

Finally, connect the discriminator outputs back to the FPGA inputs, check that they are all working still, and then take a long (week!) run to collect muon decays. The rate is only 1 / minute. Write a python program to convert the hex data to (calibrated) durations, and then plot and fit the resulting distribution.

In your first attempt you will fit the data to an exponential in order to measure $\tau$. You will note that the fit is not especially good. The primary reason for this is that the cosmic ray muon flux consists of both positive and negative muons. Whereas the positive muons can only decay, the negative muons can disappear via decay and via capture on a proton. The latter process depends on the proton number to the fourth power. For aluminum the capture rate is about 1.5 times the decay rate. Therefore, you should try to fit the
data with a sum of two exponentials. There is also a background from random coincidences. What shape
would that have in the time distribution? Include the background in your fit as well.

See the book by Rossi, Cosmic Ray Particles for data on positive and negative muon fluxes at sea
level. Compare your measured total flux rate per square cm with Rossi’s. To what do you attribute any
discrepancies?

In your report, be sure to include an estimate of the experimental uncertainty in the determination of
the value for the lifetime \( \tau \), as well as a discussion of the source of errors, and how these errors might be
minimized or eliminated.

![Figure 5: Electric circuit for photomultiplier tubes, Philips Model 56AVP.](image)

**Pre-Lab Questions**

This lab involves counting statistics, which involves the Poisson distribution. Read about the Poisson
distribution and answer the following questions:

1. Consider an experiment that counts the number events in a given amount of time (where an “event” can
be, for example, the detection of a particle). Suppose that after many repetitions of the experiment,
you find that the average number of events in the given amount of time is \( \nu \). What is the standard
deviation of the results in your experiment?

2. Suppose the count rate (events/sec) from two detectors in the muon experiment is \( R_1 \) and \( R_2 \). What
is the rate of accidental coincidences if the width of the coincidence window (measured in sec) is \( W \)?

3. How can the result of Question 2 allow you to improve your count rate without significantly increasing
the noise in your experiment?

4. Assume events occur at a particular rate \( R \) (events/s). If you start looking for events at some time
(say \( t = 0 \)), what is the probability that the first event you detect occurs between times \( T \) and \( T + \Delta T \).
Assume that \( \Delta t \) is a small interval of time so that \( R\Delta T \ll 1 \). Hint: First calculate the probability
that zero events occur in time interval \( T \).
In your first attempt you will fit the data to an exponential in order to measure $\tau$. You will note that the fit is not especially good. The primary reason for this is that the cosmic ray muon flux consists of both positive and negative muons. Whereas the positive muons can only decay, the negative muons can disappear via decay and via capture on a proton. The latter process depends on the proton number to the fourth power. For aluminum the capture rate is about 1.5 times the decay rate.

Therefore, you should try to fit the data with two exponentials. See the book by Rossi, *Cosmic Ray Particles* for data on positive and negative muon fluxes at sea level.

Compare your measured total flux rate per square cm with Rossi’s. To what do you attribute any discrepancies?

References


Advanced info below - students shouldn’t need to know this, but might want to.

To change pins used for various FPGA functions: Assignments Pin Planner, change pins in the Location column, then recompile the project.

To add/change inputs or make larger code changes, open up serial1.bdf in Files in Project Navigator. Once you’ve added new inputs in the delayer.v file (for example), do File Create/update Create Symbol Files for current file, then right click on the SyncDelay module in the serial1.bdf sheet, and do Update Symbol or Block. The new input should now be visible on the module. Then add a new input pin (or output pin) using the Pin Tool in the toolbar, and wire it to the new input/output on the module using the Node Tool on the toolbar. Assign a pin to it in the Pin Planner, as above.