

Portable Muon Tracker

2012-2013 New Mexico Tech Muon Detector Design Team:
Wesley Chavez, Katherine Harriger, Nick Kallas, and William Maier

Executive Summary

In recent years, muon-imaging technology has seen an emergence in national security applications, providing a passive method for object interrogation. For this reason, a comprehensive database of muon flux information is needed in terms of altitude and location. This project aims to produce a portable muon tracker prototype using drift tube technology that will provide the necessary instrumentation to create such a database. Using an array of drift tubes and a fitting algorithm, the tracker will be able to detect incoming muons and re-create their trajectories as well as discriminate them from background radiation.

Introduction

Considering that the design of a detector of this nature relies heavily on the physical characteristics of muons, it is important to first understand their basic nature. Muons are elementary particles that have the same charge as electrons, but are 200 times more massive. Muons are created when cosmic radiation collides with molecules in the upper atmosphere. These collisions produce pion particles, which then decay into muons. Muons have the second longest known average lifetime for particles of similar nature, lasting roughly 2.2 microseconds. This characteristic allows muons to reach the surface of the earth. Because of their high mass, muons carry a high energy (average 3GeV) that grants them special penetrating properties. On average they are able to penetrate through 1-2 meters of steel and concrete and up to hundreds of meters of solid rock (Schultz).

Project Definition

The goal of this project is to design, develop, and assemble a unit capable of detecting cosmic ray muon flux as a function of elevation and zenith angle over time. Our customers at National Security Technologies (NSTec) are working on the development of various detectors that utilize cosmic muon flux in order to detect special nuclear material for use in national security applications. Our detector will supplement current research being done at National Security Technologies in muon imaging, by providing additional data to help characterize muon flux density at the earth's surface.

To accomplish the task of detecting muon trajectories, NSTec specified that we must employ a detector based on drift tube technology, similar to the detectors that they employ at their facility. With this constraint in mind, the project definition has become

the development and assembly of hardware and software in order to interface with a drift tube array. In addition, software will be developed to process particle data and re-assemble the trajectories of the muons passing through the device. To accomplish this, we have developed a proposed solution and are in the process of developing the system needed to accomplish this goal.

Requirements of the detection system:

- a. The ability to generate a tabulated list of time-stamped particle hits in the drift tube array.
- b. From the tabulated list, generate fitted trajectories.
- c. Determine if fitted trajectories are muons.
- d. Detect the number of muon hits in terms of elevation angle.

Proposed Design

Because our customers use drift tube technology to detect muons in their applications this was the first constraint we took into consideration. With the help of NSTec, we obtained an array of 96-drift tube detectors constructed by Decision Sciences International Corporation (DSC), which is the same company that manufactured the detector hardware at NSTec. In addition, our customers (as well as DSC) showed an interest in using National Instruments (NI) hardware and software to process the particle data and recreate the trajectories in order to ensure that all data acquisition hardware was off-the-shelf and the unit would be easily reproducible. As a result, National Instruments also agreed to sponsor our project and provide their hardware, for

the purpose of this project, free of cost. Taking all of this into consideration we made the following design selections:

Hardware:

Drift Tubes:

Different types of detectors can be used to detect cosmic particles such as muons, but we are using drift tubes for the purpose of this project. In recent years, our drift tube manufacturer, Decision Sciences International Corporation, has been using drift tube arrays, composed of multiple coaxial cylindrical drift chambers. Because muons typically interact with other particles through ionization (Schultz), drift tubes give us an effective way to accurately track them. These chambers are filled with a gas mixture to promote ionization, and a bias voltage of around 3 kV is applied to the center conductor. When ionizing particles pass through a chamber, paths of ion pairs are created, and electrons drift toward the anode (center) wire. As the electric field strength increases towards the anode, the free electrons create additional ions, and an electron “avalanche” produces an electric pulse on the anode wire. With data acquisition hardware and signal processing, we will be able to detect these pulses and determine the times and locations of each particle hit using these drift tubes.

The operating principle behind these drift tubes is illustrated in Figure 1. When a muon particle is incident on the drift tubes the muon particle will travel through the tubes. The time at which the muon particle is incident on the drift tubes is known as the initial time or T_0 . As the muon particle travels through the tubes, the gas molecules

inside the tubes are ionized. The electrons freed from this ionization will drift towards the positively charged wire at the center of each drift tube. The time it takes for the electrons to hit the center wire is known as the time of hit or T_{HIT} . When the electrons hit the center wire, the potential of the center wire will drop and a negative pulse will be produced. Finally, the radius at which the muon particle intersected the drift tube can be calculated from the difference in T_{HIT} and T_0 , and the drift velocity. The drift velocity is dependent on the composition of the gas in the tubes and is the same for each tube. This process will occur in each tube that the muon particle intersects.

We chose to layer four stacked drift tube

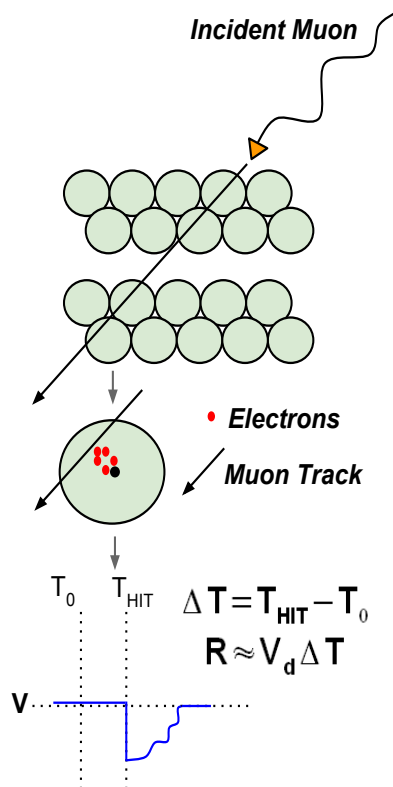


Figure 1: This is an illustration of the operating principle. An incident muon creates a track through the drift tubes. The radius of hit can be calculated from ΔT and V_d .

modules, each with 2 layers of 12 chambers, in a perpendicular arrangement for a total of 96 individual drift chambers. This gives us the minimum number of layers needed to recreate the full three dimensional trajectories of muons that passes through all

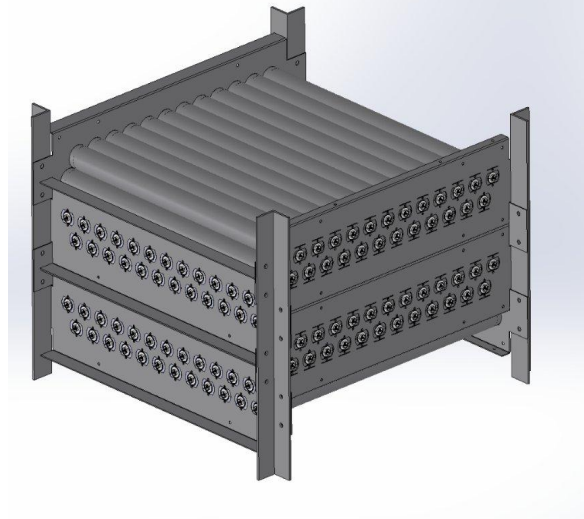


Figure 2: This is a rendering of the detectors we received for this project. It shows the four super modules layered in alternating orientation.

4 modules as well as provides a two-foot by two-foot detection area (Figure 2).

High Voltage Supply:

To bias the drift tubes, a variable high voltage supply was needed. After reviewing the feasibility of designing a custom bias supply, we decided that it would be difficult to design a unit that would be both safe to operate and provide the low noise level we needed to bias the tubes. Because of this we obtained a Bertan 205A-05R, a variable 0-5000 Volt DC supply from NSTec. The supply needed to be adjustable because the drift tubes had to be characterized in order to find optimal operating voltage. NSTec has also modified the supply to include a high

voltage BNC connector that allows us to safely connect it to our drift tube assembly without coupling additional noise onto the anode conductors in the tubes.

National Instruments Data Acquisition:

In addition to NSTec's support, National Instruments (NI) has offered to provide any hardware needed to complete this project. After evaluating the precise specification of our project, we determined that because of the high number of analog input channels and the sampling rate needed, the PXIe platform was the best choice. PXIe is a PC-based platform for measurement and automation systems. It offers seamless integration of NI instrumentation and provides us with the ability to select hardware modules that cater to our needs. In addition to the PXIe, we selected to use 6 FlexRIO FPGA cards, each fitted with a 16-channel NI-5751 digitizer module to achieve the 96 analog inputs, as well as offered us the capability of parallel processing incoming data and running real time data processing. This achieves the needed 50 MHz sample rate on all 96 channels as well as gives us the ability to preprocess incoming signals to reduce the amount of data we need to store on the PXIe control module.

Software:

LabVIEW:

The software was implemented using a graphical programming language developed by NI called LabVIEW. Despite our relative

inexperience with LabVIEW at the beginning of this project, it provided several key features that made it preferential to other programming languages given our hardware selections. These features include:

- a. Incorporation of built in programming modules and device drivers that are specifically designed to interface with NI hardware, such as the PXIe controller and FlexRIO FPGAs.
- b. Accommodation of an extensive library of functions that deal with data acquisition, advanced mathematics, signal processing, etc.

Simulation & Algorithm:

In order to fast-track algorithm design and eliminate the need for constant hardware testing we have made use of MATLAB simulations. We decided to implement these in MATLAB rather than our final operating language LabVIEW because we are more familiar with the language and MATLAB is better suited for tasks involving data processing. The simulation needed to be able to generate an ideal and a noisy data set. The ideal data served as a control used to determine the accuracy of the fitting algorithm we created. A least-squares linear regression technique is then used to transform the noisy data set into a trajectory. Several trajectories are generated in order to account for the ambiguity of the initial hit time (the time that the particle passes through the array). The generated trajectories are then run through a chi-square goodness-of-fit test that determines the optimal trajectory. We chose to employ

these simulations for a number of reasons, but first and foremost they have significantly improved the efficiency of our software development. Once we confirmed the function of our MATLAB algorithm we then ported it into LabVIEW using a built in module called MathSCRIPT.

Progress

So far, we have completed or made progress in many of the tasks needed for the success of the project. We have received all of the hardware necessary for the completion of this project as well as completed the assembly of the detection unit depicted in Figure 2. We have implemented software and hardware to interface our drift tubes with our NI PXIe processing unit. We are able to individually and simultaneously trigger on particle events on multiple detector channels as well as generate time-stamps corresponding to these triggers. Additionally we have completed and implemented the software architecture to transfer these time-stamps from the FPGAs to the host unit.

In MATLAB, we have produced simulations in both two and three dimensions to help us hone in on a final algorithm. At the moment, we have a regression algorithm that can recreate a muon track from noisy radius data. The chi-square algorithm is satisfactorily accurate and has been ported to LabVIEW.

Results

Since we have broken down the task of detecting and recreating muon trajectories

into a modular design (Appendix A) we have been able to complete each subsection of our project as follows:

Data Acquisition:

Since our design utilizes the 16-channel digitizers mentioned above it was first necessary to connect between the Very High Density Cable Interconnects (VHDIC) located on the digitizers and the 14-pin ribbon cable connections on the drift tubes. To do this we employ a stackable break out board and distribute the 16 ribbon connections over the 6-digitizer units (three of such connections are depicted in Appendix B).

On each analog input channel of the 6 FPGAs we have implemented a falling edge analog trigger that activates when the negative pulse of a particle event breaches a user input trigger level with a negative slope. To prevent false triggering caused by noise, we have also included hysteresis in our trigger so that once we have triggered the analog input level must return to a designated level above the trigger level before the channel can trigger again (Appendix C). We were able to confirm the function of our trigger by creating a program that showed where in time our trigger activated on an inverted pulse waveform as seen in Figure 2.

Once a trigger has been received, a time-stamp is generated based on a 40MHz clock that is distributed among the FPGAs to minimize jitter. The FPGA then takes the time-stamp and channel information and transfers it to a FIFO that passes the

information to the host PXI-e machine. Each FPGA employs two FIFOs that serve eight analog input channels each; this ensures that the data can be read more quickly and a loss of information will not be caused by an overflow of data in the FIFO.

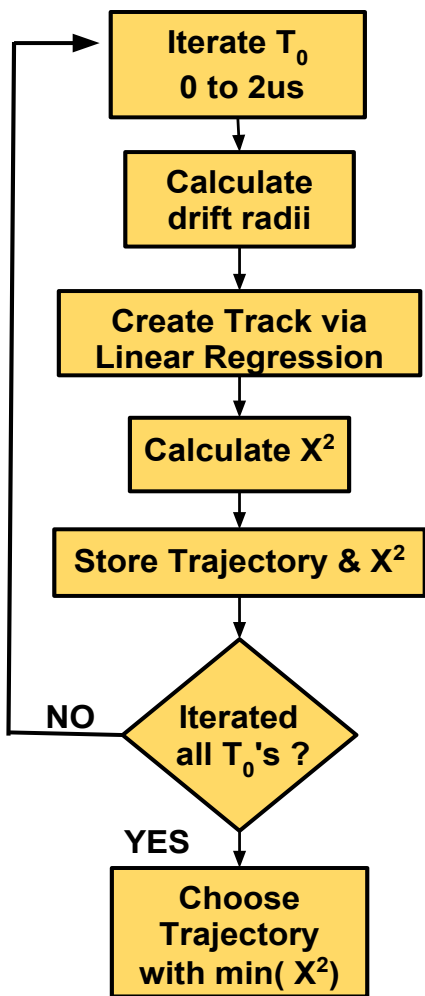


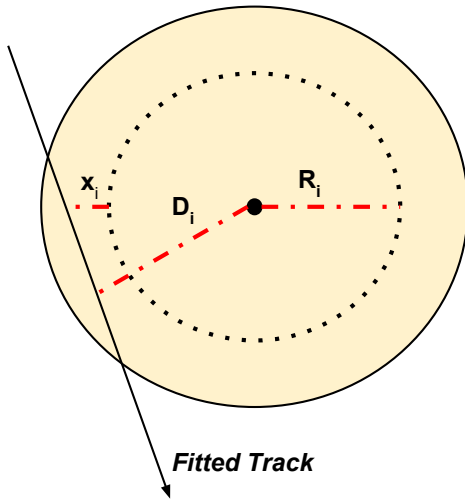
Figure 3: Flow Chart of the fitting algorithm

Fitting Algorithm:

The fitting algorithm generates the trajectory using linear regression by first creating an x- and y-coordinate that determines an approximate position in space at which the muon particle intersected the drift tube.

These (x,y) coordinates are calculated from the radius of hit and direction of travel. The direction of travel can be inferred from knowing the next and/or previous tube that was hit by the muon particle. Once the (x,y) coordinates for each tube that was hit are calculated, linear regression is applied to calculate a slope and intercept for the trajectory. Note that, x is width, y is height, and z is depth. It is assumed that the z-plane is zero, so that the conversion from 2D to 3D is simple. The next important calculation that the fitting algorithm uses is the chi-square goodness-of-fit test.

The chi-square goodness-of-fit test is a statistical test that calculates how well observed data fits expected data. In the case of the muon tracker, the observed data is the distance from the center of the tube to the fitted trajectory, and the expected data is the radius of the hit. We choose the T_0 that resulted in the minimum chi-square value, because this indicates the T_0 that resulted in the overall smallest difference between expected and observed data for the fitted trajectory, or in other words, the minimum chi-square value indicates the T_0 that results in the smallest error. Figure 4 illustrates the chi-square calculation. Note that σ is the standard deviation for the position of the center wire. The standard deviation for the center wire position can be different for each tube, but for our purposes it was accurate enough to assume a constant σ of 1 mm for each tube. Appendix D includes the commented code for the fitting algorithm.



$$x_i = D_i - R_i$$

$$X^2 = \sum_{i=1}^N \frac{x_i^2}{\sigma^2}$$

Figure 4: Chi-square goodness-of-fit test calculation. D_i is the distance from the center of the wire to the trajectory, the observed data. R_i is the radius of hit, the expected data. x_i is the difference between D_i and

Future Work

In order for this product to be considered complete and functional as specified by our customers we have outlined the remaining work required. The work we have yet to complete is as follows:

- Complete software on host machine to receive & correlate time-stamps for fitting algorithm.
- Apply algorithm to hardware and continue optimization.

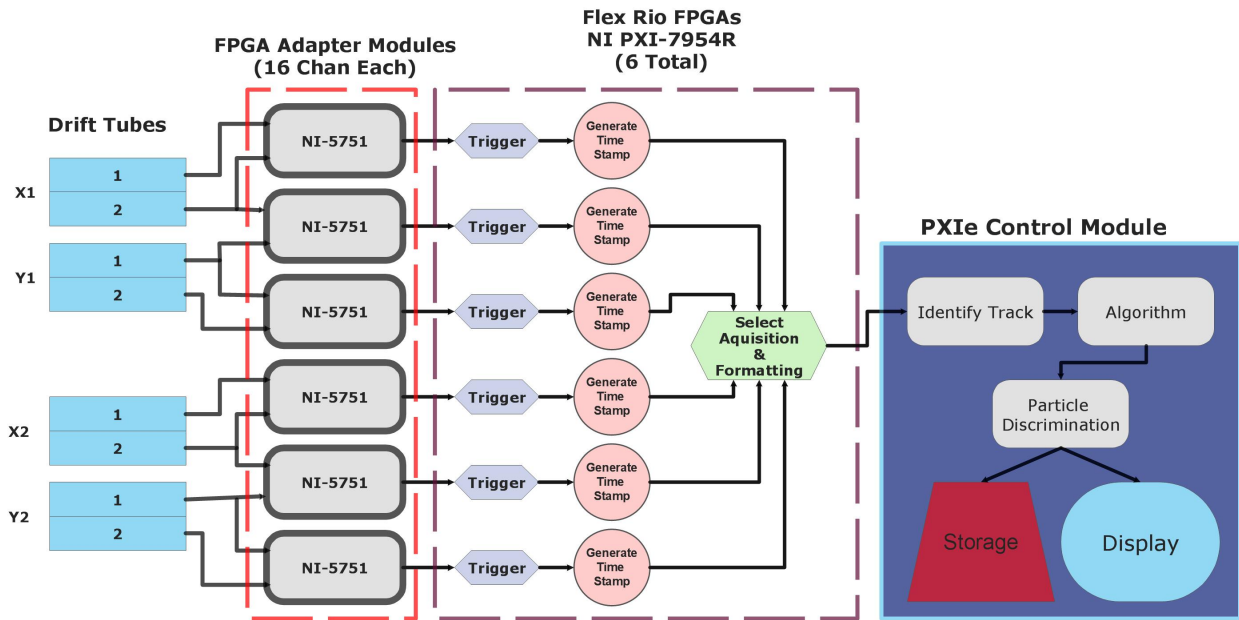
- Complete documentation for end user.

The final step of the project will be developing a portion of code to operate on the PXIe that will read time-stamp information and correlate it in such a way that it will only pass useable information to our fitting algorithm. Useable information refers to a set of time-stamps that correlate to a trajectory that has passed through all the layers of our drift tubes. In addition to having to pass through all eight layers of our detector we must confirm that the correlated time-stamps occur within a limited period of time corresponding to the maximum drift time of our tubes.

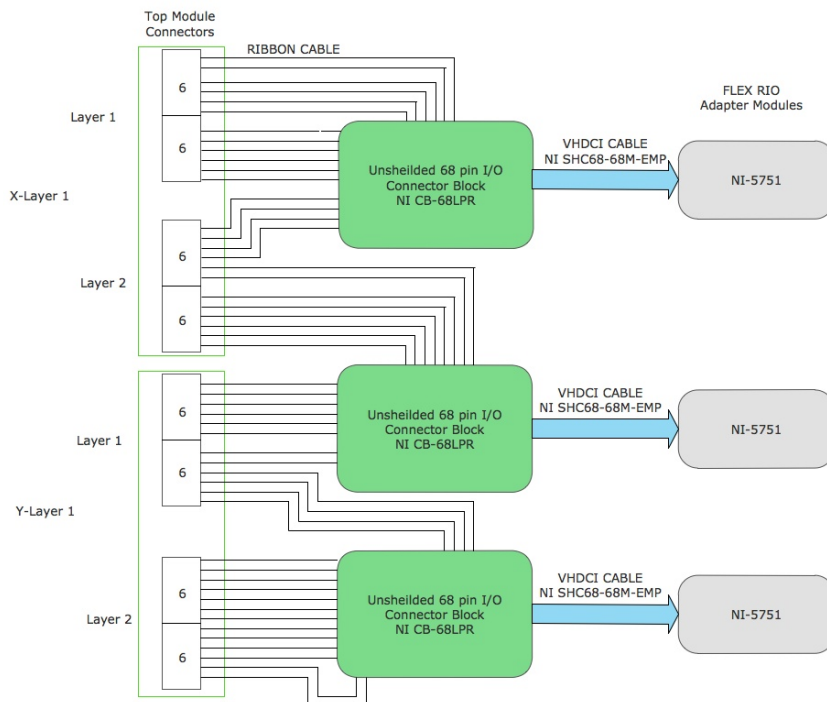
Next we must confirm that our algorithm functions in the LabVIEW environment by testing it with the aforementioned correlated data. Using MATLAB we were able to test it and confirm its functionality, but we have do not yet have the ability to test this algorithm in LabVIEW. We do not foresee this to be a significant problem, however it is possible that this may need additional debugging when it comes time to integrate it with the rest of the program.

Once the product is functioning within the specifications set by our customers, documentation will then need to be generated to provide them with the necessary instruction to use the detector. In addition we will include technical documentation both in our code and in the form of a document that details how our programs function.

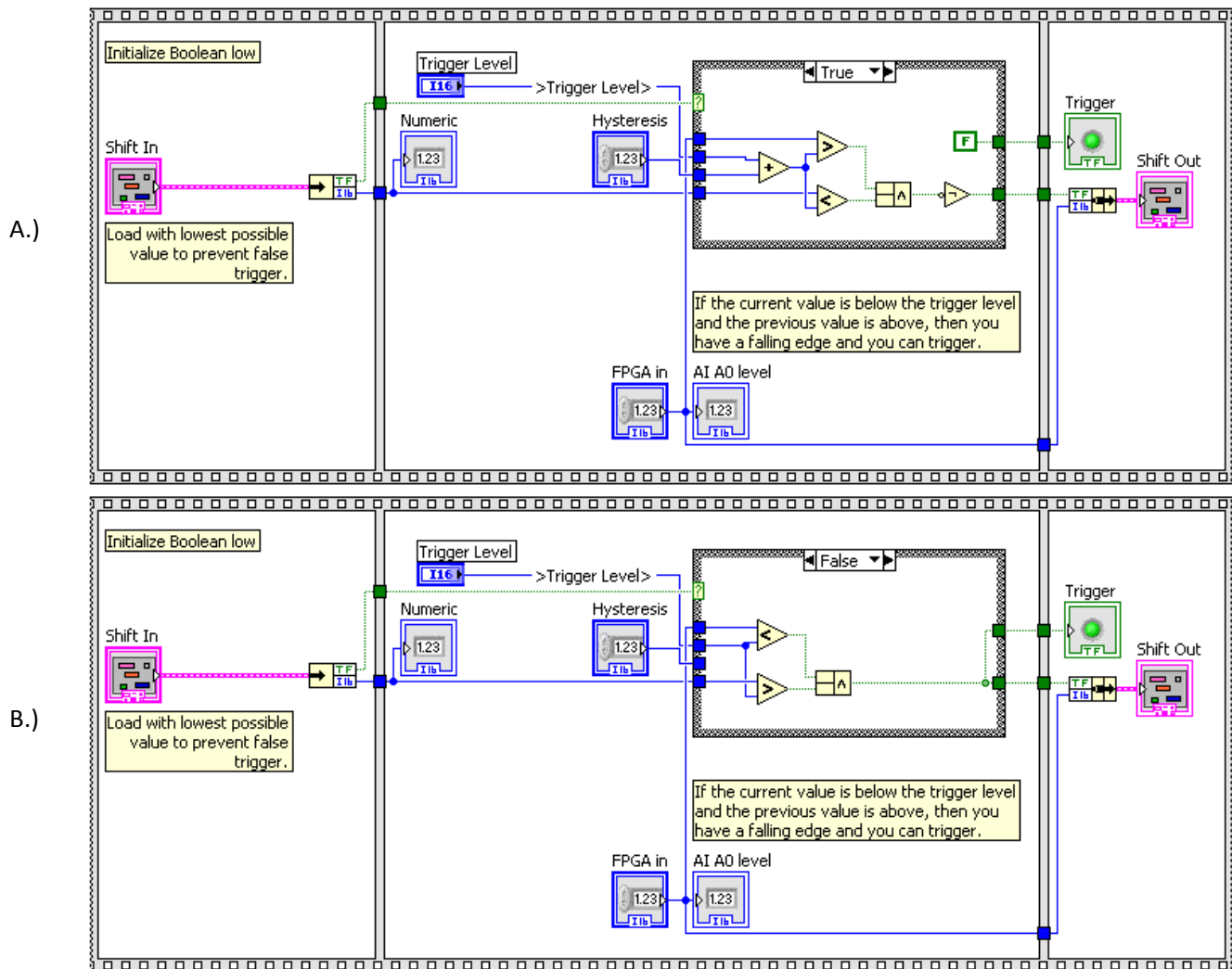
Appendix A



Appendix B



Appendix C



The trigger program begins in phase A, and operates by comparing the current input value (denoted "FPGA in") and the previous input value (denoted shift in) with a trigger level. If the current input is below the trigger level and the last input was above the trigger level, then we have confirmed a falling edge, and the trigger outputs a Boolean true. Once this occurs the trigger program changes to phase B upon the next iteration. In phase B the trigger program waits until the input breaches the hysteresis level with a rising edge before it returns to phase A and awaits another trigger.

Appendix D

```
%This function will computer the Trajectory of Muon Events, it's
goodness of
%fit Chi square value. This function is designed to run in a
LabVIEW
%Mathscript node. This function is to be run for each muon
event.
%
%INPUT
%time = 8x1 vector of time stamps for each hit
%tube = 8x1 vector of tube number (1-12) that was hit
%      - the 8 rows of these vectors correspond to the
respective layer
%OUTPUT
%Xfits = 8x1 X coordinates of muon hits
%Yfits = 8x1 Y coordinates of muon hits
%M = vector containing slope and intercept of fitted trajectory
%CHI = chi square goodness of fit test value

% Physical Parameters of Muon Tubes, center coordinates
center_coordX = []; %This is a 2D array that's rows are 1-8 for
each layer, and columns 1-12(for each tube in the layers), so
the x coordinate of the center of tube 6 in layer 1 is
center_coordX(1,6)
center_coordY = []; %2D structured the same as the one above, but
for the Y coordinates

for t0 = 0:(20*10^-9):(2*10^-6) %iterates through possible
values of t0, from 0 to 2us (20ns increments)
    t=time-t0; %find time of hits for each tube
    radii = Vd.*t; %radii values of each hit, calculated from t

    for n = 1:8
        centerX(n) = center_coordX(n,tube(n)); %finds center
coordinates for each hit by referencing 2d array of coordinates
of tube centers
        centerY(n) = center_coordY(n,tube(n));
        if n ~= 8
            if centerX(n+1)>centerX(n)
                SIGN = -1; %SIGN represents if
hit on left or right of center)
            else % if SIGN = -1, then
hit of left , SIGN = 1, hit on right
                SIGN = 1;
            end
        end
    end
end
```

```

end
if n == 8
    if centerX(n-1)<centerX(n)
        SIGN = -1;
    else
        SIGN = 1;
    end
end
end

%% Find X/Y Coordinates

for z = 1:7
    unit_vec = radii(z)*[1 0];
    center_vec =
[centerX(z+1)+SIGN(z+1)*radii(z+1),centerY(z+1)]-
[centerX(z)+SIGN(z)*radii(z),centerY(z)];
%% Angle of the slope of the trajectory is found by projecting
the unit_vect onto the directional center_vec
    Theta(z) =
acos(dot(unit_vec,center_vec)./(norm(unit_vec)*norm(center_vec))
);

end
% LAST TUBE LAYER

for z = 8;

    unit_vec = radii(z)*[1 0];
    center_vec = [centerX(z-1)+SIGN(z-1)*radii(z-
1),centerY(z-1)]-[centerX(z)+SIGN(z)*radii(z),centerY(z)];
end
    Theta(8) =
acos(dot(unit_vec,center_vec)./(norm(unit_vec)*norm(center_vec))
);
%% Calculation of the (X,Y) Spacial Coordinates of wear the
muon intersected the drift tube
    X1 = centerX(1) + SIGN(1)*radii(1)*cos(Theta(1));
    Y1 = centerY(1) + radii(1)*sin(Theta(1));

    X2 = centerX(2) + SIGN(2)*radii(2)*cos(Theta(2));
    Y2 = centerY(2) + radii(2)*sin(Theta(2));

    X3 = centerX(3) + SIGN(3)*radii(3)*cos(Theta(3));
    Y3 = centerY(3) + radii(3)*sin(Theta(3));

    X4 = centerX(4) + SIGN(4)*radii(4)*cos(Theta(4));
    Y4 = centerY(4) + radii(4)*sin(Theta(4));

```

```

X5 = centerX(5) + SIGN(5)*radii(5)*cos(Theta(5));
Y5 = centerY(5) + radii(5)*sin(Theta(5));

X6 = centerX(6) + SIGN(6)*radii(6)*cos(Theta(6));
Y6 = centerY(6) + radii(6)*sin(Theta(6));

X7 = centerX(7) + SIGN(7)*radii(7)*cos(Theta(7));
Y7 = centerY(7) + radii(7)*sin(Theta(7));

X8 = centerX(8) + SIGN(8)*radii(8)*cos(Theta(8));
Y8 = centerY(8) + radii(8)*sin(Theta(8));

%% Calculates Trajectory Parameters via linear regression using
the psuedo inverse
X = [X1;X2;X3;X4;X5;X6;X7;X8];
Y = [Y1;Y2;Y3;Y4;Y5;Y6;Y7;Y8];
G = [X ones(8,1)];
m = pinv(G)*Y;
Ynew = m(1)*X + m(2);

Tx(:,size(t0)) = X; % Store values for each t0
Ty(:,size(t0)) = Ynew;
Tr(:,size(t0)) = radii;
Tm(:,size(to)) = m; %slope and intercept for each
trajectory
end
%% Chi Square Goodness of Fit test
% Z plane = 0
center = [centerX centerY zeros(8,1)]; %make center 3D
s = 0.0393701; %%assuming 1mm standard deviation for each
wire (inches)
Z = zeros(size(Tx));
[m n] = size(Tx);
for j = 1:n % number of trajectories
    for i = 1:m %number of hits
        % Calculates distance from trajectory to wire
        D(i,j) = abs(dot((center(1,:)-[Tx(1,j) Ty(1,j)
Z(1,j)]),cross(center(i,:),[Tx(i,j) Ty(i,j)
Z(i,j)])))/(norm(cross(center(i,:),[Tx(i,j) Ty(i,j) Z(i,j)])));
        %Residuals for fit
        res(i) = Tr(i,j) - D(i,j);
    end
    chi_square(j) = sum((res.^2)./s^2);

end
% Finds index , k, such that chi_square is minimized
k = findnz(min(chi_square));

```

```
%returns corresponding Tx,Ty, and Tm
Xfits = Tx(:,k);
Yfits = Ty(:,k);
M = Tm(:,k);
CHI = min(chi_square);
```

References:

Schultz, J., Larry. 2003. *Cosmic Ray Muon Radiography*, Portland State University