Dead Layer Calculation and Error Analysis

April 4, 2015

What follows is the Procedure, calculation, and error analysis for calculating the inactive layer of silicon on the surface of the silicon detectors.

1 Experimental procedure

Two stationary alpha sources (Pu240 and Cf249) are used as calibration and another source (Pu238) is shifted in location to determine the energy difference between the two positions. Each data run was for 6 hours exactly. John Greene provided the sources.

This was done for detectors 1 and 3. 2 and 4 draw a lot of current when not at cyro temperatures, so we were afraid of damaging them when doing the measurements at room temperature. Each experiment was done under vacuum though. The pressure stayed within the range of several microtorr.

If needed, the original data files for each run can be found on the computer in the triangle room downstairs in the B8 directory with a folder labeled something along the lines of dead layers.

Currently, the final sorted files that I used (found on this computer) are: deadlayerdet1cent2.root, deadlayerdet1rt2.root, deadlayerdet3cent.root, and deadlayerdet3rt.root.

all the placements for the geometry of the mounts was done using pictures of the mounts to determine where the center of the sources was (only made a difference for pu238) otherwise, the cf did not give any problems for being so big and we don't actually know where the pu 240 is. The placement calculations are found in mathematica under the worksheet placement calculations. Use the calculation that matches the numbers in pixelanalysis.c

Each of the root files had to have histograms made from the data so that they could be easily fit. This was done using selector files in the bypixel directory. It does the histograms by pixel. These files start with Singles and don't look like your normal file names. Type "make" to compile anything you change. The resulting histogram files that were fitted are: dldet1cent2.root, dldet1rtrebinned.root, dldet3cent.root, dldet3rtrebinned.root. The right position had to be rebinned so that instead of having 1 bin per channel, there's 0.5 bins per channel. This was to help with the noise when fitting, since counts are lower for the right position.

IMPORTANT! fstrips run side to side when viewed from the stool and the bstrips run up and down. As viewed from the stool, the first pixel is in the lower right hand corner.

2 fitting

This brings us to the fitting part. This one was a massive pain. Various fitting codes can be found with the beginning deadlayerspix... Each one is used for something different, so use whichever one you want. The ones that have the same name as what you want have the parameters that correspond to a somewhat good fit for that thing. Here's an example of what the fits look like and the fitting function.

BUT TAKE CAUTION!!! it may be worth your time to attempt the fits again possibly with a better fitting function. Maybe set tau1 and tau2 equal to zero. The first time you did this, some tau's were orders of magnitude smaller than they should have been. This has a huge impact on the line center. Some of the

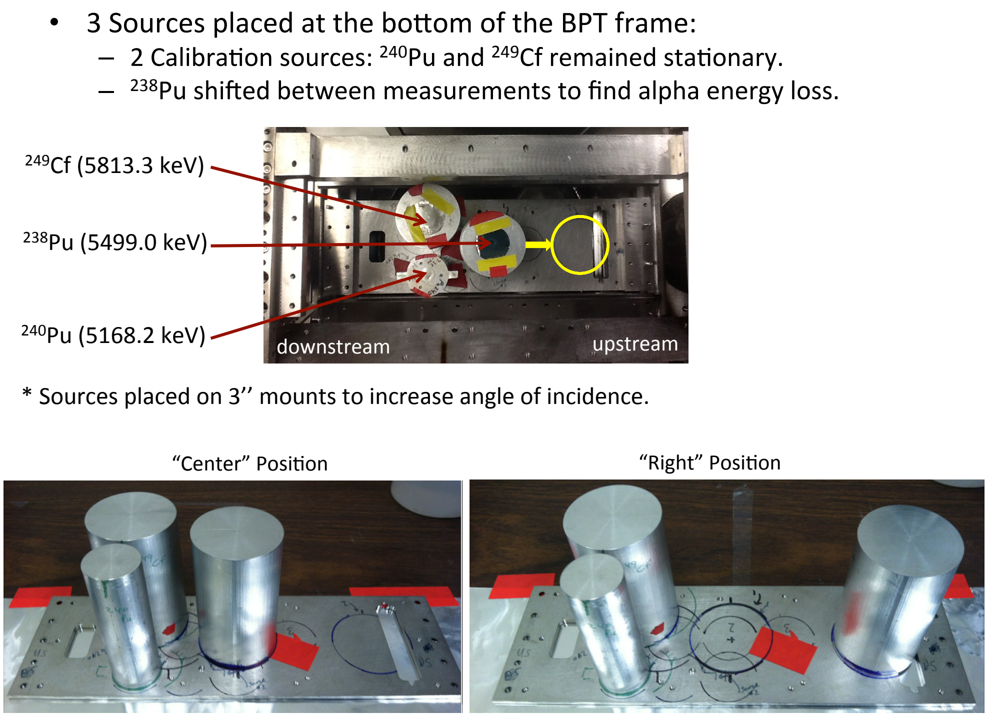

Figure 1: schematic of how the setup was put together

Fitting spectra:

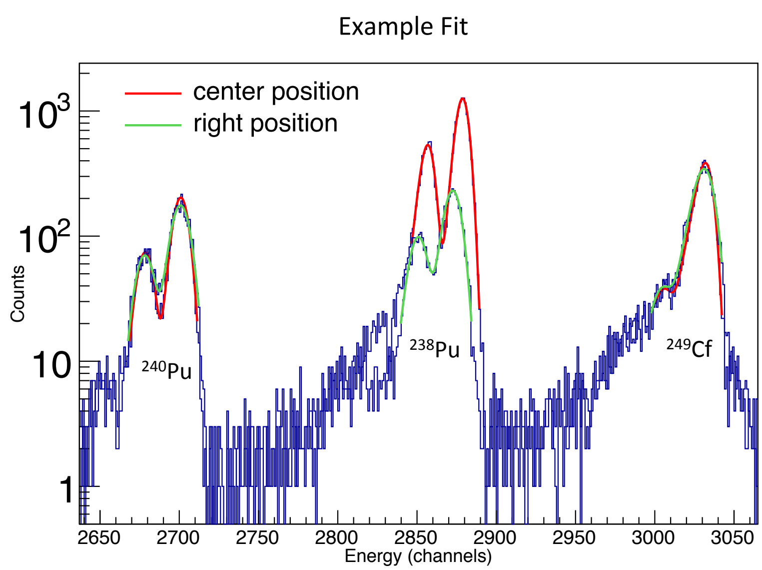

Figure 2: what a fit looks like

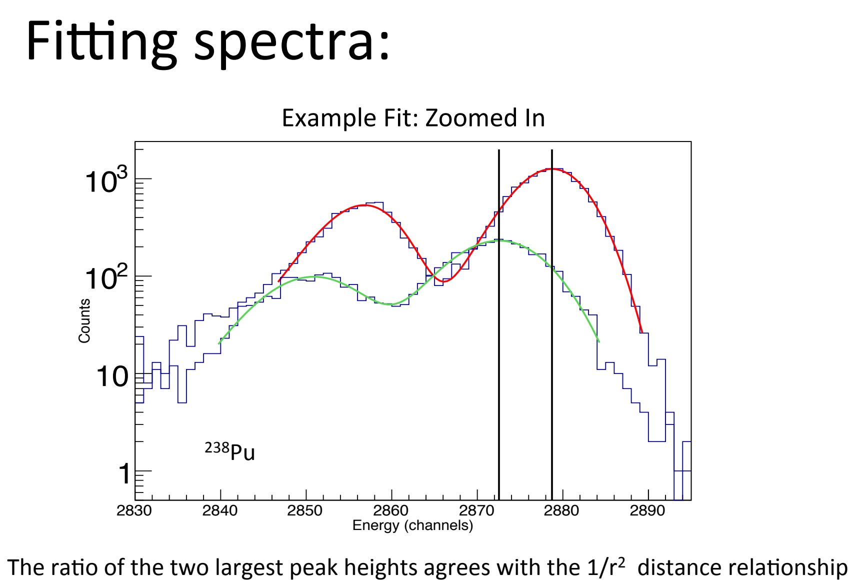

Figure 3: what a fit looks like zoomed in (this is a good figure that will likely end up in your thesis. The code for it is found in histoverlay.c

Fitting spectra:

$$\text{Fitting Function}(u_i) = \sum_{i=1}^{2} \frac{A_i}{2} \left(\frac{1-\eta_i}{\tau_i}\right) \exp\left(\frac{u_i-\mu_i}{\tau_i} + \frac{\sigma^2}{2\tau_i^2}\right) \operatorname{erfc}\left(\frac{1}{\sqrt{2}}\left(\frac{u_i-\mu_i}{\sigma} + \frac{\sigma^2}{2\tau_i^2}\right)\right) + \left(\frac{\eta_i}{\tau_i}\right) \exp\left(\frac{u_i-\mu_i}{\tau_i} + \frac{\sigma^2}{2\tau_i^2}\right) \operatorname{erfc}\left(\frac{1}{\sqrt{2}}\left(\frac{u_i-\mu_i}{\sigma} + \frac{\sigma^2}{2\tau_i^2}\right)\right)$$

*Modified from Bortels 1987.

- A_i: amplitude
- η_i: “percentage” parameter defining the contributions of 2 peaks used to define an alpha spectrum
- τ_i: defines low energy tail length
- σ: width of the convoluted Gaussian peak
- μ_i: Line center of the peak.

Figure 4: the fitting function and some info

- The calibrated energy difference between the 2 ^{238}Pu peaks was plotted vs the change in inversed angles to create a linear regression in which the slope (L) is the dead layer.

$$E^c_{pu238} - E^r_{pu238} - E_{src}\left(\frac{1}{\cos\theta_r} - \frac{1}{\cos\theta_c}\right) = L\left(\frac{dE_{pu238}}{dx}\right)\left(\frac{1}{\cos\theta_r} - \frac{1}{\cos\theta_c}\right) + B$$

*E_{src} represents the dead layer of the ^{238}Pu source

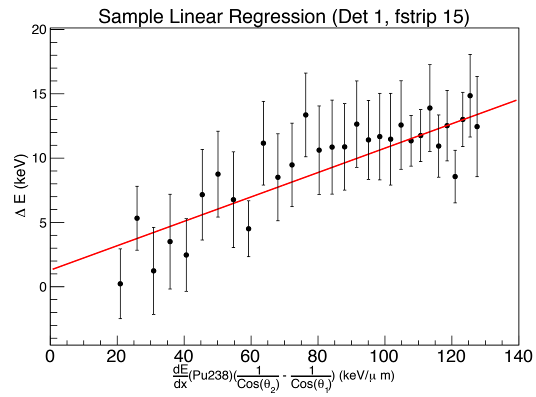

Figure 5: this is how we do

tau's had to be forced to be consistent, which caused to to inflate the error bar of those line centers if they had been forced. A large amount of error comes out of that, so if you can eliminate it, it may be worth it. (if only so that you don't have to do this experiment again.

If you want to play with the parameters and better understand them, the Alphapeakfit mathematica worksheet will be a fun game. Here's a brief explanation of each term though:

Ai: amplitude

η_i: percentage parameter defining the contributions of 2 peaks used to define an alpha spectrum

τ_i: defines low energy tail length (high= big tail, low= small tail)

σ: width of the convoluted Gaussian peak

μ_i: Line center of the peak.

The final fitted text files have names that allude to cats. As it happened, Dr. Livingston was sitting on your lap while you were doing this fitting for the upteenth time, so the final ones are named in his honor.

3 analysis

And now begins the fun part. All the analysis is found in "pixelanalysis" codes. The "new" one deals with all of the pixels together, and the original can only deal with one strip at a time. They both spit out different plots based on what you need. All of terms and error analysis is summarized here. The idea was to create a linear regression over a front strip (with the intercept floating and the slope would be the dead layer)

- **Energies of the alpha particles as emitted from each of the three sources**: In the code, they are listed as Epu240, Epu238, and Ecf249.The values were taken from Matt's analysis of the dead layers, though they agree with the NNDC. Their errors were deemed negligible and not included.* may have been ammended

- **Stopping powers corresponding to each of the alpha's energy in silicon**. They are depicted by a 3 value vector called dEdx, where the values are placed in the order of increasing alpha energy. These are found using a SRIM calculation (found in Helium in Silicon.xlsx)

- **Errors corresponding to the stopping powers provided by SRIM**. They are represented by the vector dEdxe in the analysis code. The calculations for these are found on the same spreadsheet.

- The vertical difference between the sources and the detector. Calculated by subtracting the heights of the mounts and wafers from the total difference between the lower plate and the detector (not measured: given by the original dead layer analysis). This value is eventually converted to micrometers

- **The dead layer value for detector 1** calculated in the last round of measurements by Adrian: meant to be used to offset systematic error when calculating the slope and intercept used to calculate the energy of the Pu238 alphas. In the code, it is labeled as prevdl with the error as prevdlerr.

- **locations of each of the sources on the mounting board** in reference to the first usable pixel (the outside strips are discarded). Units are in micrometers. The values are represented in the code as pu240x, pu240y, pu238cx, pu238cy, pu238rx, pu238ry, cf249x, and cf249y. "x" refers to the placement of the source compared to the first usable pixel in the direction of the beamline. "y" refers to the same, except perpendicular to the beamline. "c" (center) and "r" (right) refer to first and second placements of the pu238 source respectively.

- **The errors of the placements for each sources**. Units are also in micrometers. They are represented in the same manner as the original values in the code, except with an "err" at the end of each name. The errors themselves are dominated by the location of the source on the wafers. The pu240 source could not be located, so half the radius of the available space for the source to be was used as the error. For pu238, the source's location was well defined, so a 1mm error was used. For cf249, the apparent source dot was 8.33 mm in diameter, so an error of 4 mm was assumed.

- **fitted line centers of each source** units are in channels. They are represented in the code as pu240ec, pu240er, pu238ec, pu238er, cf249ec, and cf249er. In this case, "c" and "r" also refer to the separate fitting done for each placement of the pu238 source.

- **The line center errors** are represented by adding an "r" between the "e" and "r" or "c." The units are also in channels. The additional numbers added in quadrature represents the average 2σ spread of variation between hours as measured in the previous calibration. (standard deviation of each source used in an average over strips)

3.1 Error analysis

The end result should be a graph using one front strip (pixels varying over the back strip) of the calculated energy difference of the pu238 alphas between the two positions vs the stopping power of the pu238 alphas multiplied by the difference of the inverted cosines of the incident angles of each position (for pu238).

$$E_c - E_r = L\frac{dE}{dx}(Pu238)(\frac{1}{\cos\theta_r} - \frac{1}{\cos\theta_c})$$

Using the previously described inputs, this is how both the x and y errors are calculated.

3.2 X errors

Calculating the errors for $\frac{dE}{dx}(Pu238)(\frac{1}{\cos\theta_r} - \frac{1}{\cos\theta_c})$ was fairly straightforward. Each cosine was calculated as follows in this general form for each source/position.

$$\cos(\theta) = \frac{Height}{\sqrt{Height^2 + PosX^2 + PosY^2}}$$

Their error was propagated ($Height = H$, $PosX = X$, and $PosY = Y$).

$$\sqrt{(\frac{\sqrt{H^2+Y^2+X^2} - \frac{H^2}{\sqrt{H^2+Y^2+X^2}}}{H^2+Y^2+X^2})^2 Herr^2 + (\frac{HY}{(H^2+Y^2+X^2)^{\frac{3}{2}}})^2 Yerr^2 + (\frac{HX}{(H^2+Y^2+X^2)^{\frac{3}{2}}})^2 Xerr^2}$$

If the $\cos\theta$ of the center Pu238 position is $\cos\theta_c \pm Cerr_c$ and the right Pu238 position is $\cos\theta_r \pm Cerr_r$ The total X error for each pixel is:

$$\sqrt{(\frac{1}{\cos\theta_r} - \frac{1}{\cos\theta_c})^2 (\frac{dE}{dx}_{err}(Pu238))^2 + (\frac{dE}{dx}(Pu238)\frac{1}{\cos{\theta_c}^2})^2 Cerr_c^2 + (\frac{dE}{dx}(Pu238)\frac{1}{\cos{\theta_r}^2})^2 Cerr_r^2}$$

3.3 Y errors

The actual alpha energy from the Pu238 source was calculated by finding a linear slope and intercept using the 2 other sources and correcting for the approximate dead layer. (subscripts pu and cf refer to the Pu240 and Cf249 sources respectively, Ch refers to the fitted line center, and SLOPE refers to the first term in the rhs side of the equation)

$$Epu238 = (\frac{(E_{cf} - \frac{prevdl \frac{dE}{dx}_{cf}}{\cos\theta_{cf}}) - (E_{pu} - \frac{prevdl \frac{dE}{dx}_{pu}}{\cos\theta_{pu}})}{Ch_{cf} - Ch_{pu}})(Ch_{Pu238}) + (E_{pu} - \frac{prevdl \frac{dE}{dx}_{pu}}{\cos\theta_{pu}} - SLOPE \times Ch_{pu})$$

But! We're actually dealing with the difference of the 2 positions ("r" refers to the right position and "c" refers to the center.) Also, having the dead layer and/or not having it causes an approximately 0.3 shift in the total dead layer, therefore it's not important, so we took it out. However! The actual energies were still calculated using the original intercept method.

$$Epu238_c - Epu238_r = (E_{cf} - E_{pu})(\frac{Ch_{pu238,c} - Ch_{pu,c}}{Ch_{cf,c} - Ch_{pu,c}} - \frac{Ch_{pu238,r} - Ch_{pu,r}}{Ch_{cf,r} - Ch_{pu,r}}) = m_c(Ch_{pu238,c} - Ch_{pu,c}) - m_r(Ch_{pu238,r} - Ch_{pu,}$$

In the case of the pu being subtracted from the pu238, it was exchanged with the cf based on which had a lower error.

$$\sigma_m = ((\frac{(E_{cf} - E_{pu})}{(Ch_{cf} - Ch_{pu})^2})^2 \times (Ch_{err,cf}^2 + Ch_{err,pu}^2) + \frac{E_{cf,err}^2 + E_{pu,err}^2}{(Ch_{cf} - Ch_{pu})})^{\frac{1}{2}}$$

and thus the energy difference error is:

$$\Delta E_{err} = \sqrt{(Ch_{pu238,c} - Ch_{pu,c})^2 m_{c,err}^2 + (Ch_{pu238,r} - Ch_{pu,r})^2 m_{r,err}^2 + (m_{c,err})^2((Ch_{pu238,c,err}^2 + Ch_{pu,c,err}^2) + (m_{r,err})^2(((C}$$

It will be added in quadrature at the end with the source dead layer error.

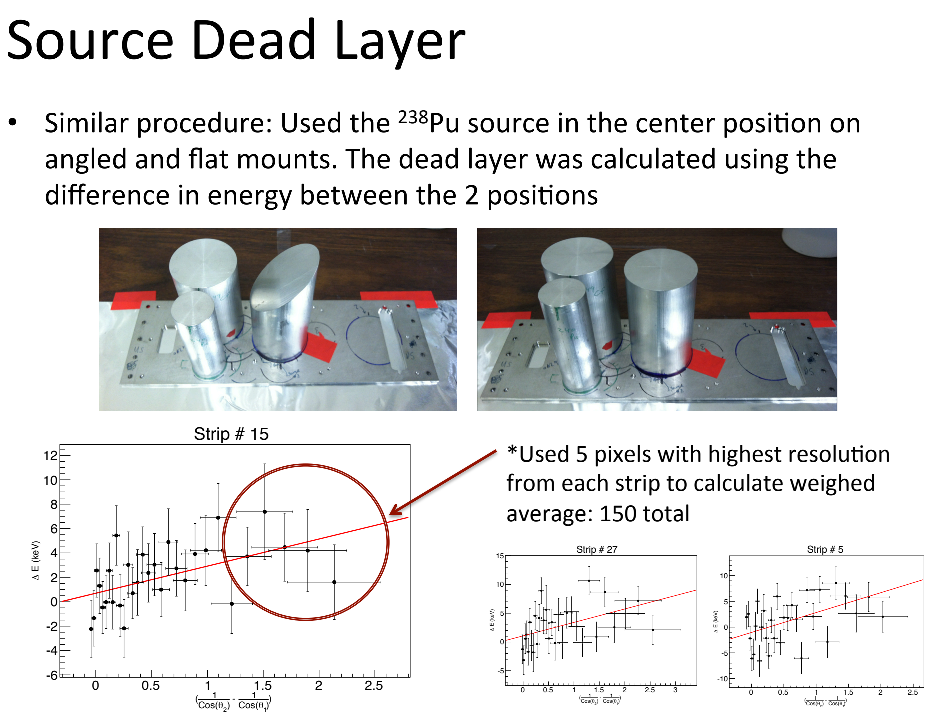

Figure 6: schematic of how the setup was put together

- Weighted average of 150 points: 2.7 ± 0.14 keV

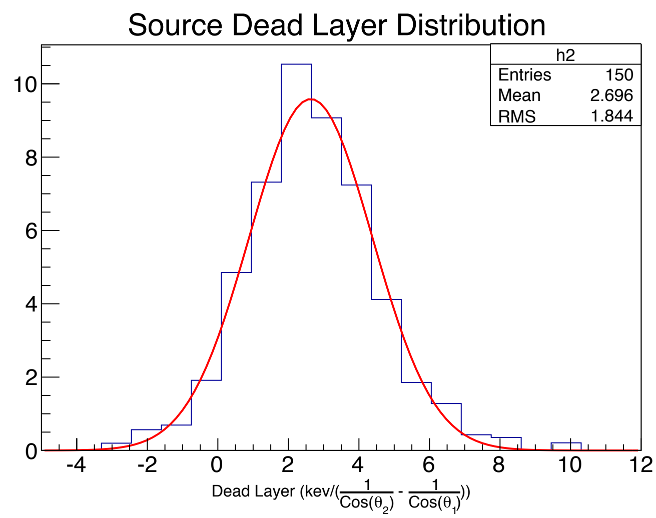

Figure 7: numbers

4 source dead layer

which brings us to the source dead layer. This was done in the same manner as the other dead layer, but we used a different angled mount.

since a linear regression was not possible, we picked the 150 points (last 5 bstrips) that had the highest resolution and averaged those. The 150 points were chosen by going bstrip by bstrip and seeing if they worked together. The 6th one did not. This produced a "dead layer: of 2.7 kev with an appropriate error. This was factored into the final number using the inverse cosines as would be expected. The files you need are: srcdeadlayerangle.root, srcdeadlayerflat.root, dlsrcflat.root, dlsrcangle.root, anglefinalsorted.txt, and flatfinalsorted.txt

5 last bits

we may have to do this experiment again, so be thinking of ways to make it better (the whole procedure) i.e, better sources, higher data taking rates, etc.

It was explored using other methods to analyze the data, (setting b=0) for the linear regressions by front strip (the calibrations didn't perfectly cancel out and we thought we were cheating, also made some weird things happen with the distribution) or by backstrip, which was also weird. For more information and lots of pictures, see the dlinfo slide show.

The final number that went into Matt's paper was the average of the 2 detectors with the floating b frontsrip analysis and then doubling their error bars. 115.4 ± 3.8 boom