

FPix Module Testing Reference Guide

J. Antonelli

September 24, 2015

The document provides an overview of the module testing methodology and procedures used for the CMS Forward Pixel Phase I Upgrade project. It briefly introduces the individual components of a pixel module and describes the module assembly procedures. It then describes in greater detail all the individual tests performed during module testing and calibration, discussed in the other that the tests are normally performed. Finally, it describes the standard procedure for testing modules at Fermilab and saving the test results for visualization and grading purposes.

Note: This document can be edited by checking out the package here:
<https://github.com/jstupak/FPIXUtils>

Contents

1	FPix Module Components	4
2	FPix Module Construction	4
3	The Pixel Unit Cell	4
4	Modules Tests	7
4.1	PreTest	7
4.1.1	Purpose	7
4.1.2	Methodology	7
4.1.3	Output	8
4.2	PixelAlive Test	10
4.2.1	Purpose	10
4.2.2	Methodology	10
4.2.3	Output	10
4.3	Trimming Test	12
4.3.1	Purpose	12
4.3.2	Methodology	12
4.3.3	Output	14
4.4	PHOptimization Test	20
4.4.1	Purpose	20
4.4.2	Methodology	21
4.4.3	Output	22
4.5	GainPedestal Test	26
4.5.1	Purpose	26
4.5.2	Methodology	26
4.5.3	Output	26
4.6	SCurve Test	29
4.6.1	Purpose	29
4.6.2	Methodology	29
4.6.3	Output	30
4.7	BumpBonding Test	32
4.7.1	Purpose	32
4.7.2	Methodology	32
4.7.3	Output	32
4.8	IV Test	35
4.8.1	Purpose	35
4.8.2	Methodology	35
4.8.3	Output	35
4.9	FullTest	36

5	Module Testing at FNAL	37
5.1	Experimental setup	37
5.2	Safety concerns	38
5.3	Testing procedure	38
6	Test Result Databases	39
7	Module Grading Scheme	39

1 FPix Module Components

The pixel module for the CMS Phase I upgrade consists of several discrete components, manufactured by independent contractors. The silicon sensor is the active unit of the module, converting energy deposited by incident particles into an electrical signal. Each sensor contains 66,560 individual pixels. The standard pixel size is $100\ \mu\text{m} \times 150\ \mu\text{m}$, although some edge pixels are twice as large in the dimension perpendicular to the sensor edge.

The psi46 readout chip (ROC) is an 8 mm square integrated circuit that takes the signal from the sensor, processes it, and packages the data to be sent onward in the readout chain. Each ROC covers an array of 80×52 pixels (4160 pixels).

The high density interconnect (HDI) is a printed flexible circuit that accepts the data from multiple ROCs and combines it to be transmitted to the central CMS data acquisition (DAQ) system. The serialization of the data is performed by the token bit manager (TBM) which is a logic chip glued onto the HDI.

2 FPix Module Construction

Module assembly takes place in several independent steps at multiple locations. First, the sensor is attached to 16 ROCs by the company RTI using a bump-bonding technique. Balls of lead/tin solder, roughly $30\text{--}40\ \mu\text{m}$ in diameter, are deposited onto the ROCs. Then each ROC is aligned to the sensor and the two are pressed together to complete the electrical connection.

HDIs, produced by Compunetics, are delivered to Fermilab where all the surface components (including the TBM) are mounted.

The combination of 16 ROCs bump-bonded to a sensor is referred to as a “bare module”. RTI sends the bare modules to the two FPix assembly sites at Purdue University and The University of Nebraska. At the assembly sites, the completed HDIs are glued to the top of the sensor side of the bare modules. Then the readout pads of the ROCs are connected to the input pads of the HDI via thin wires in a process called wirebonding. The complete modules are tested for basic functionality, and any correctable errors (e.g. broken wirebonds) are addressed.

The modules are then shipped to one of three FPix testing sites: Fermilab (FNAL), University of Kansas (KU), and University of Illinois-Chicago (UIC). KU and UIC are equipped with X-ray testing setups, while FNAL is the central testing site where the final calibration and grading is performed. This document will detail the testing procedures at FNAL, as X-ray testing is covered in a separate document.

3 The Pixel Unit Cell

The bulk of module testing concerns the pixel unit cell (PUC) of the ROC. For this reason, it’s useful to discuss the design and operation of the PUC. Figure 1 shows the elements of the PUC, the double column periphery, and the controller and interface block.

It is labeled with some of the DAC registers that are most relevant for module testing. Terms in black boxes represent programmable DAC registers on the ROC. While all DAC memory registers are ROC-wide and exist at the periphery (except the 4 [Trim bits](#)), the diagram shows DACs next to the circuit element they directly affect. For the remainder of this document, these DAC registers will be denoted with [blue text](#). The process of calibrating a module is largely comprised of tests that find and set the optimal values for these parameters.

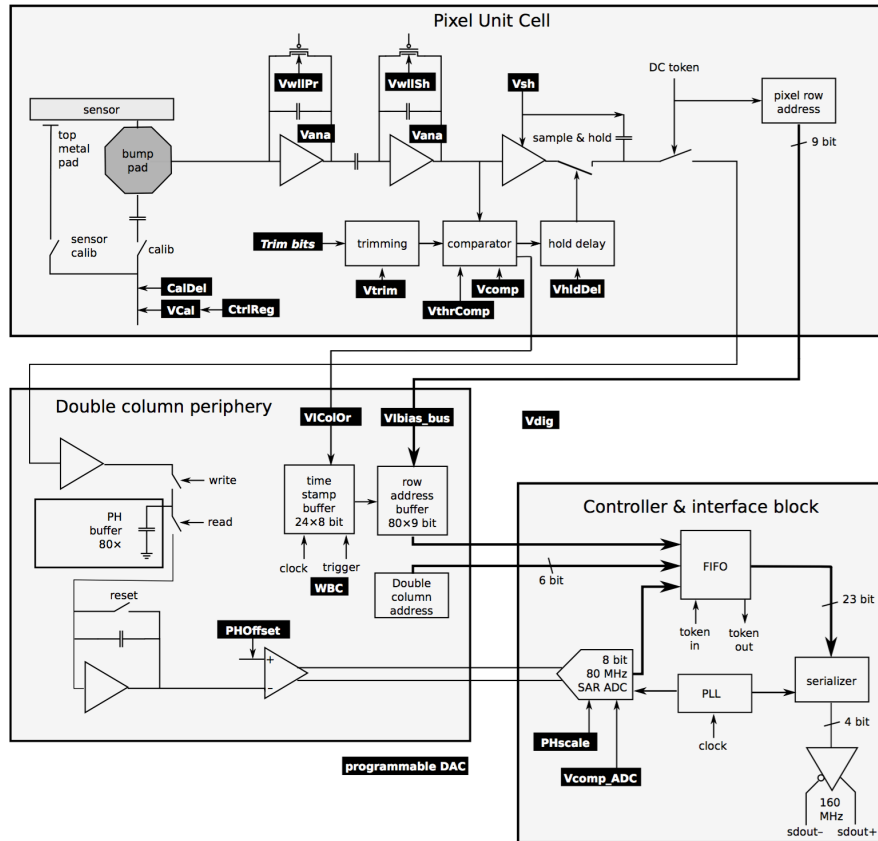


Figure 1: A schematic of the circuit components and the relevant programmable registers within the ROC.

The signal processing within the PUC begins with an amplifier and shaper that convert the incident current into a reformatted voltage signal. The signal is then fed into a voltage comparator, the threshold of which is controlled by $V_{thrComp}$ with additional inputs possible from V_{trim} and the [Trim bits](#). If the input signal exceeds the comparator's threshold, the signal is stored for readout and the pixel notifies the logic at the periphery of the double column that a hit has been registered and is ready to be transferred to the data buffer at the periphery. This initiates the double column drain mechanism, transferring pulse height (PH) values from any pixel with a hit registered

since the previous drain onto the 80-wide PH buffer on the periphery. In addition to the ability to process charge produced by particles passing through the silicon sensor, the ROC can produce an internal calibration signal whose amplitude is controlled via the V_{cal} DAC. This calibration signal serves as the input to the PUC in almost all of the tests described in this document. Table 1 shows a subset of DAC registers in the PUC and explains their usage.

Table 1: List of DAC registers relevant to module testing. DAC names are denoted with blue text.

DAC Name	Usage	Tests that optimize this DAC
V_{cal}	Calibration signal voltage. There are two V_{cal} ranges (low and high) with the high range being roughly 7 times larger. The choice of low/high range is set via the CtrlReg DAC.	None - set to an appropriate level depending on the test
V_{ana}	Voltage supplied to the analog part of the PUC (amplifier and shaper)	PreTest (SetVana)
CalDel	Delay in sending out calibration pulse with respect to the clock signal	PreTest (SetVthrCompCalDel)
V_{thrComp}	Voltage supplied to the comparator unit, inversely related to the signal voltage required to fire the comparator (i.e. the comparator threshold)	PreTest , Trimming
V_{trim}	Additional voltage supplied to the comparator unit to decrease the threshold (in conjunction with the Trim bits)	Trimming
Trim bits	4 bits per PUC that decrease the threshold of the comparator unit	Trimming
PHOffset	Constant offset added to the pulse height	PHOptimization
PHScale	Sets the slope of the PH vs. input signal strength curve	PHOptimization

4 Modules Tests

Modules are tested using the pXar software framework. More information on pXar can be found at <https://twiki.cern.ch/twiki/bin/viewauth/CMS/Pxar>.

With a few exceptions, plots created by pXar represent data from a single ROC. For this reason, most plot names contain a “C” (short for “chip”) followed by the ROC number (0-15). Most ROC plots have one entry per pixel, and are seen as 1D distributions or as 2D distributions (52×80) organized by the row and column indices of the pixels in the ROC. Since, in principle, tests can be performed multiple times, all plot names are appended with a version number. Together, the template for pXar plot naming is “PLOT_NAME_C#_V#”. When pXar creates a 1D distribution from a 2D ROC map, the 1D plot’s name is equal to the original 2D plot name prepended with “dist_”.

In the sample plots included in this document, ROC 0 is shown as an example. Analogous plots, produced for each of the 16 ROCs, are included in any pXar output file.

4.1 PreTest

4.1.1 Purpose

The **PreTest** is meant to put the module into an operational state in preparation for more specific tests. The basic operation of the ROC is checked and some of the default DAC settings are tested. The **PreTest** is made up of several subtests, each of which is discussed in turn.

4.1.2 Methodology

In the **ProgramRoc** subtest, the “programmability” of the ROC is tested. The value of the total current drawn by the analog amplifiers in all the PUCs (I_{ana}) is measured. This is done first with the nominal value of V_{ana} , and then again with V_{ana} set to zero. The difference in I_{ana} from these two measurements should be non-zero, signifying that V_{ana} is actually being applied to the ROC. This implies the ROC is capable of receiving commands to alter DAC values, i.e. it is programmable. The output of this subtest is shown in Figure 2.

The **SetVana** subtest sets the V_{ana} DAC so that the total current draw (I_{ana}) meets a target value (default is 24 mA). V_{ana} is iteratively adjusted from its default value until it comes within 0.25 mA of the target or until 10 iterations have been performed. The output is shown in Figures 3 and 4.

The **SetVthrCompCalDel** subtest optimizes the values of the $V_{thrComp}$ and $CalDel$ DACs. A sample pixel is chosen from within each ROC for this test and is assumed to be representative of normal pixel performance. A scan is performed over $V_{thrComp}$ and $CalDel$ to determine which combinations of them put the pixel into an operational state. At each

point in the 256×256 value parameter space, a number of calibration pulses (default is 5) are sent to the PUC of the chosen pixel. The number of hits successfully recorded by the pixel is then calculated, and the results are displayed in the 2D plane of V_{thrComp} vs. CalDel in what is referred to as a “tornado plot” (they looked more like tornados with the older, analog ROC). Figure 6 shows a tornado plot. The red region corresponds to 100% efficiency – the operational region for the PUC with respect to V_{thrComp} and CalDel . Recall that V_{thrComp} is inversely related to the minimum pulse height required to fire the comparator. The region of $V_{\text{thrComp}} > 150$ has zero efficiency because the threshold of the comparator is so low that the pixel is constantly firing on electronic noise, and when the calibration pulse is injected, the pixel is already in an unresponsive state waiting to be read out. By default V_{thrComp} is set 50 units higher than the lower edge of the operating plateau, and CalDel is set halfway between the left and right edges of the plateau at the chosen V_{thrComp} value. The black dot centered within the white square denotes the chosen working point for these two parameters. Figure 7 shows the values of CalDel chosen by the test.

4.1.3 Output

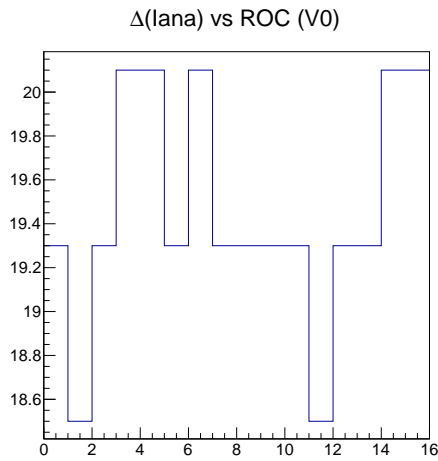


Figure 2: Created by the `ProgramRoc` subtest. Plotted is the difference in I_{ana} for cases of V_{ana} on/off, as a function of ROC number. Values for programmable ROCs are non-zero.

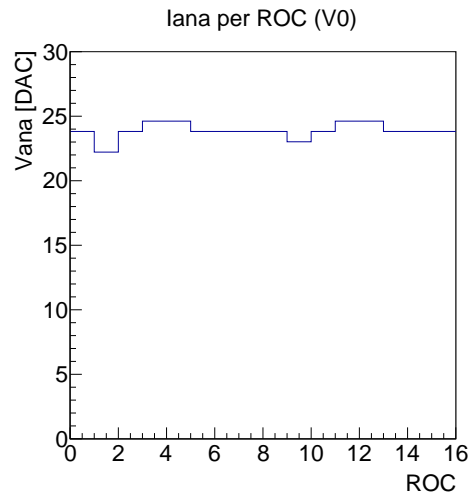


Figure 3: Created by the `SetVana` subtest. Plotted are the final values of I_{ana} (typo in y-axis label), as a function of ROC number. The default target value for I_{ana} is 24 mA.

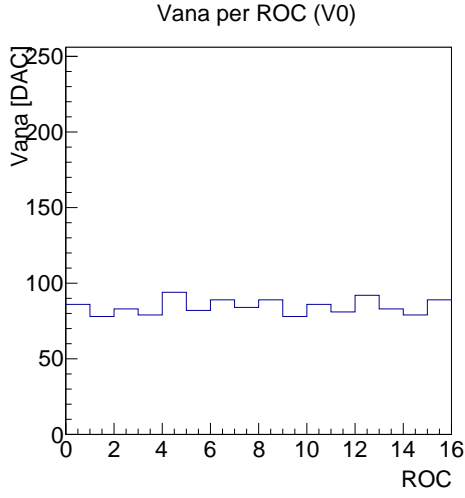


Figure 4: Created by the `SetVana` subtest. Plotted are the optimized values of V_{ana} , as a function of ROC number. For the default I_{ana} target of 24 mA, V_{ana} is roughly 80 DAC units.

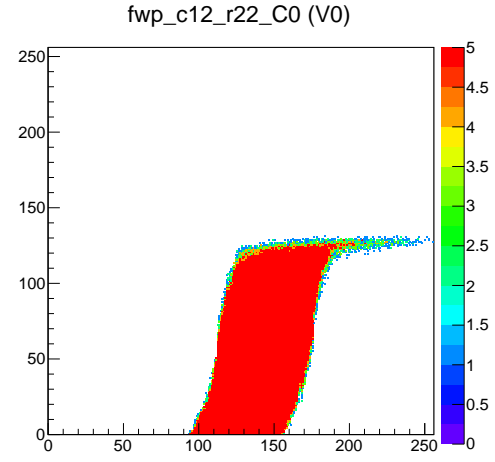


Figure 5: Created by the `FindWorkingPixel` subtest. Similar to Figure 6.

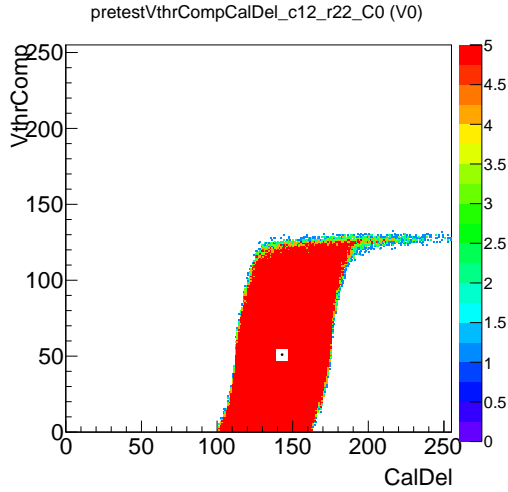


Figure 6: Tornado plot (efficiency in the $V_{thrComp}$ vs. $CalDel$ plane).

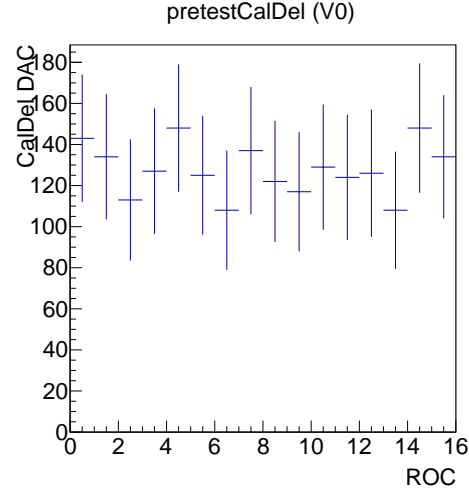


Figure 7: The chosen values of $CalDel$ as a function of ROC number.

4.2 PixelAlive Test

4.2.1 Purpose

Once the pretest is completed, we can be relatively sure that all non-faulty pixels should respond to the calibration pulse. The **PixelAlive** test uses this fact to identify faulty pixels. Because a pixel can be faulty in multiple ways, three subtests are run that each flag a different kind of faulty pixel: dead pixels, unmaskable pixels, and pixels with addressing errors.

4.2.2 Methodology

In the **Alive** test, a number of calibration pulses (default 10) are sent, in turn, to each pixel in the device and the number of signals recorded is measured. During this period, all pixels other than the one being tested are masked, so as not to produce spurious noise signals. Pixels with no recorded hits are flagged as dead and pixels with less than 100% efficiency are flagged as problematic. Output is shown in Figure 8.

In the **Mask** test, all pixels are disabled and the same procedure is performed. Pixels with efficiency above zero have a problem with the pixel masking procedure, and are flagged as bad. Output is shown in Figure 9.

In the **AddressDecoding** test, the same efficiency measurement is performed, pixel by pixel, but the order of the resulting data is checked. If the address of a given pixel is out of order, the recorded hit is given a negative pulse height value. Pixels with negative hits are flagged as faulty. Output is shown in Figure 10.

To distinguish between dead pixels and otherwise faulty pixels in output of the **Mask** and **AddressDecoding** tests, a special 3-state scheme is used: Entries with a weight of 1 denote good pixels, entries with a weight of 0 denote dead but not faulty pixels, and entries with a weight of -1 denote faulty pixels, i.e. masking (decoding) errors for the **Mask** (**AddressDecoding**) test.

4.2.3 Output

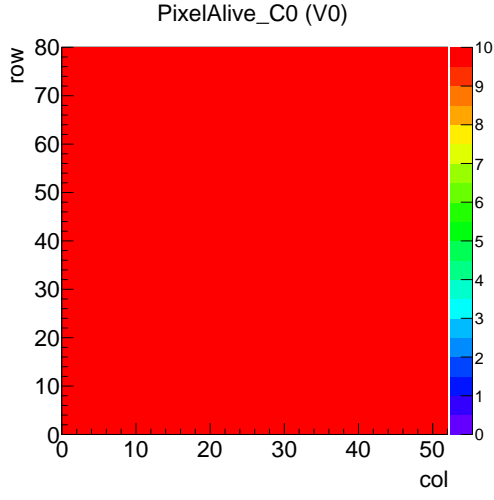


Figure 8: Created by the `Alive` test. Efficiency corresponds to bin content divided by number of triggers (here 10).

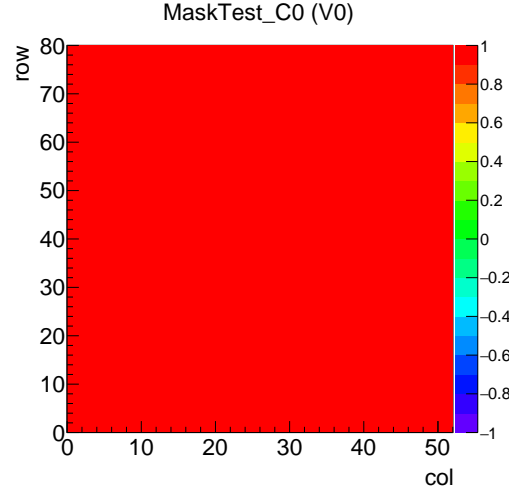


Figure 9: Created by the `Mask` test. 1 (good), 0 (dead), -1 (mask problem).

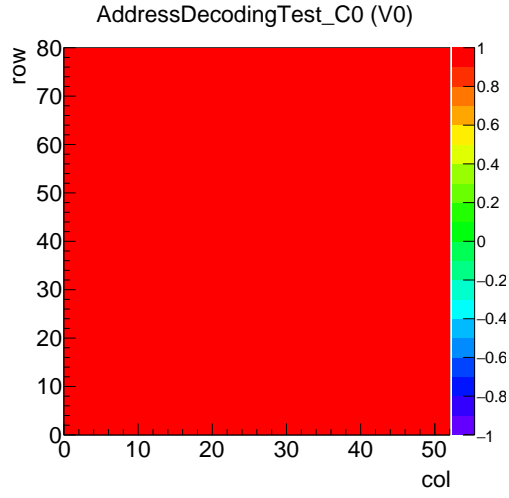


Figure 10: Created by the `AddressDecoding` test. 1 (good), 0 (dead), -1 (mask problem).

4.3 Trimming Test

4.3.1 Purpose

The **Trimming** test accounts for variations in the sensitivities of the 4,160 PUCs within a ROC. By using the combined effect of the V_{thrComp} , V_{trim} , and the **Trim bits**, it attempts to calibrate every PUC to the same threshold in terms of the calibration pulse strength. These DACs can be seen in Figure 1 as inputs into the “trimming” and “comparator” elements of the PUC. V_{trim} sets a course scale for the trimming and the **Trim bits** further fine tune the trimming. While V_{thrComp} and V_{trim} are ROC-wide values, there exist four **Trim bits** for every pixel in a ROC. “Trimming” a ROC using V_{trim} and the **Trim bits** effectively means reducing the signal amplitude required to fire the comparator. Once a module is trimmed, all pixels should turn on at a user-defined threshold (default is $V_{\text{cal}} = 35$ (low range)). Thus, input signals with less strength than this threshold will not fire the comparator and will not be registered as a hit. This ensures that all pixels within a module have a uniform level of zero-suppression. Additionally, the **TrimBits** subtest verifies that all trim bits are working by sequentially enabling each bit and observing its effect on the pixel threshold distribution.

s

4.3.2 Methodology

It’s important to understanding the relevant DACs for this test: V_{thrComp} , V_{trim} , and the **Trim bits**. V_{thrComp} sets a baseline threshold for the comparator unit. This threshold can be reduced (i.e. “trimmed”) through the combined effect of V_{trim} and the **Trim bits**. V_{trim} sets the ROC-wide scale for the effect of the **Trim bits**. Hence, it can be thought of as a multiplicative factor being applied to the **Trim bits**. It’s important to note that a trim bit is “enabled” when set to zero, such that all bits are active when **Trim bits** = 0 [0000] and all bits are inactive when **Trim bits** = 15 [1111]. Thus, the combined effect of V_{trim} and the **Trim bits** is proportional to $V_{\text{trim}} \times (15 - \text{Trim bits})$. The trimming is effectively turned off when either $V_{\text{trim}} = 0$ or when **Trim bits** = 15.

The **Trimming** test is made up of two subtests: the **Trim** subtest and the **TrimBits** subtest. The **Trim** test sets V_{thrComp} , V_{trim} , and the **Trim bits**, while the **TrimBits** test just validates that each of the **Trim bits** is functioning.

The **Trim** test is responsible for optimizing three DACs: V_{thrComp} , V_{trim} , and the **Trim bits**. As with other tests, these DACs are set independently for each ROC. The **Trim** test first optimizes V_{thrComp} . To remove the effect of V_{trim} and the **Trim bits**, V_{trim} is initialized to zero. With V_{cal} set to the target threshold (default is 35), it produces s-curves for all pixels with respect to V_{thrComp} . S-curve tests are described in more detail in Section 4.6. From the distribution of turn-on values (Figure 11), the entry with the lowest V_{thrComp} turn-on is chosen and V_{thrComp} is set to this value. In this process, pixels with turn-on values more than 2σ below the mean of the distribution are not considered. At this V_{thrComp} value, all pixels should turn on at V_{cal} values higher than

the target value. This is because the effect of V_{trim} and the Trim bits is to reduce the threshold of the comparator. Therefore, with V_{cal} set to the target value, none of pixels should be firing with V_{trim} set to zero. Furthermore, this minimum V_{thrComp} value is required to be at least 10 DAC units below the V_{thrComp} turn-off value for the pixel with the smallest V_{thrComp} turn-off. This ensures none of the pixels could potentially fire on noise at the this V_{thrComp} value.

With an optimal setting for V_{thrComp} , the test targets the optimization of V_{trim} . The effect of the Trim bits on the lowering of the comparator threshold is multiplied by the value of V_{trim} . V_{trim} should be set to the lowest value that still allows all pixels in the ROC to be successfully trimmed to the target value. In this way, the entire 4-bit range of the Trim bits is used, maximizing the precision of the trimming. Setting V_{trim} any higher than this would just compress the range of the Trim bits used. To find the minimal effective value of V_{trim} , the test first identifies the pixel with the highest V_{cal} turn-on, using the standard s-curve procedure (Figure 12). This is the pixel that requires the most trimming to have its V_{cal} threshold reduced to the target value. The test then enables all four Trim bits (Trim bits = 0), maximizing their effect. Using this pixel, the test performs an efficiency scan over the V_{trim} and V_{cal} DACs (Figure 13). Starting at high V_{trim} , the value of V_{trim} is iteratively lowered until the V_{cal} turn-on surpasses the target V_{cal} . This value of V_{trim} corresponds to the minimum value that can effectively trim this pixel, and it is chosen as the final V_{trim} value for the ROC. This procedure effectively sets the level of V_{trim} at which the chosen pixel needs all four bits enabled to be trimmed. Since this pixel is known to need the most trimming, all other pixels can by definition be trimmed with an equal or lesser number of trim bits enabled.

With the appropriate dynamic range for the trimming set using V_{thrComp} and V_{trim} , the test can proceed to fine tune the trimming for each individual pixel using the four Trim bits. This is achieved using a binary search tree algorithm. Starting with the Trim bits set to 7 [0111] (in the center of their 0-15 range), V_{cal} turn-on thresholds are determined via s-curves (Figure 14). For any pixel that fires below (above) the target V_{cal} value, the Trim bits value is increased (decreased) by 4, so that the amount of trimming is decreased (increased). The s-curve test is repeated (Figure 15), and the Trim bits values are increased or decreased by 2, as appropriate. The procedure is repeated two more times (Figures 16, 17), with an increment of one, so that all possible values of the Trim bits are within reach of the test. The turn-on thresholds after the final correction are shown in Figure 18. The optimized values of the Trim bits are shown in Figures 19 and 20. To validate that the trimming was reasonably effective, a final s-curve test is performed (Figures 21, 22).

The TrimBits test verifies that all Trim bits are programmable. Since (in conjunction with V_{trim}) the Trim bits alter the threshold of the comparator, enabling a given bit should have a measurable effect on the V_{cal} turn-on for the pixel in question. V_{trim} is set to the maximum value (255), and S-curves (with respect to V_{cal}) are obtained with all Trim bits disabled (Trim bits = 15 [1111]). The turn-on values extracted from

these curves serve as untrimmed baseline values (Figure 23). Then each bit is enabled in turn and new s-curves are recorded for values of **Trim bits** =14 [1110] (Figure 24), **Trim bits** =13 [1101] (Figure 26), **Trim bits** =11 [1011] (Figure 28), and **Trim bits** =7 [0111] (Figure 30). At each iteration, V_{trim} is reduced by a factor of two to cancel out the effect of which bit is being enabled. Then each set of turn-on values is subtracted from the baseline set, to isolate the effect of enabling a single trim bit. Distributions of these differences are shown for **Trim bits** =14 [1110] (Figure 25), **Trim bits** =13 [1101] (Figure 27), **Trim bits** =11 [1011] (Figure 29), and **Trim bits** =7 [0111] (Figure 31). If the difference is larger than zero, the trim bit in question is capable of being programmed and is having a noticeable effect. Pixels with a defective trim bit are flagged.

4.3.3 Output

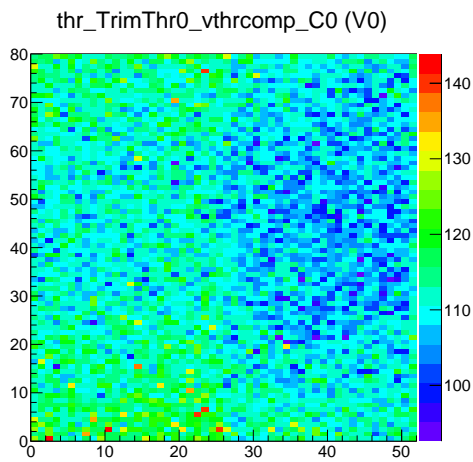


Figure 11: ROC map of V_{thrComp} turn-on thresholds with V_{cal} set to target value. Used to find pixel with minimum turn-on (least sensitive pixel).

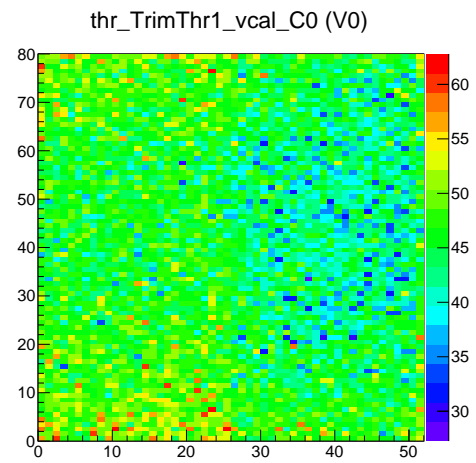


Figure 12: ROC map of V_{cal} turn-on thresholds with minimized V_{thrComp} value. Used to find most sensitive pixel, i.e. the pixel requiring maximum trimming.

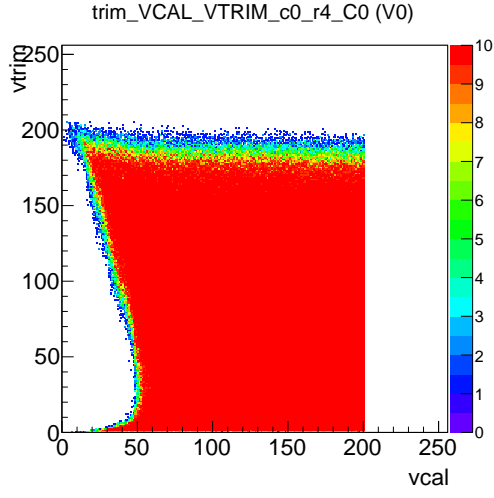


Figure 13: Efficiency in the V_{trim} vs. V_{cal} plane. Used to find the value of V_{trim} that corresponds to a turn-on at the target V_{cal} .

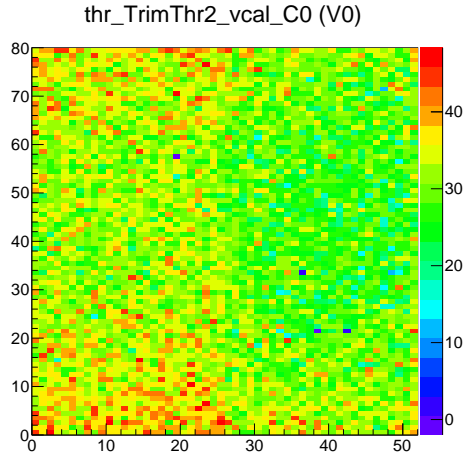


Figure 14: ROC map of V_{cal} turn-on thresholds with $\text{Trim bits} = 7$. Used as baseline for setting the Trim bits .

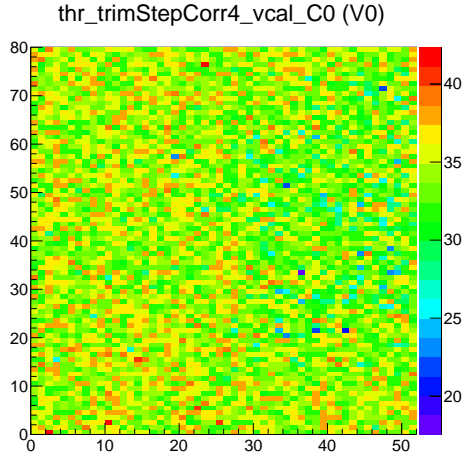


Figure 15: ROC map of V_{cal} turn-on thresholds after a Trim bits ± 4 correction with respect to Figure 14.

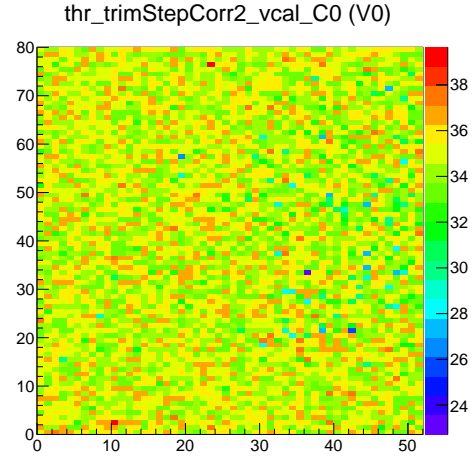


Figure 16: ROC map of V_{cal} turn-on thresholds after a Trim bits ± 2 correction with respect to Figure 15.

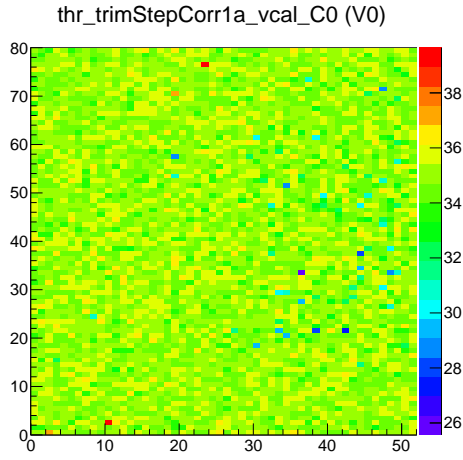


Figure 17: ROC map of V_{cal} turn-on thresholds after a Trim bits ± 1 correction with respect to Figure 16.

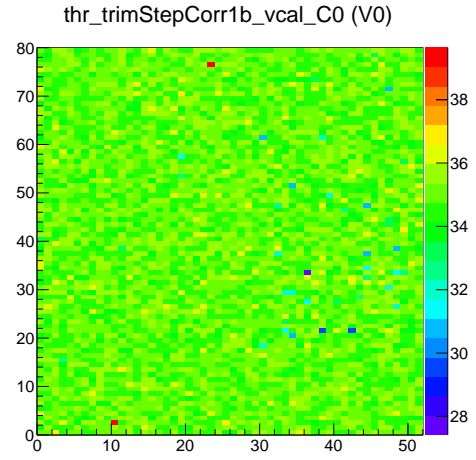


Figure 18: ROC map of V_{cal} turn-on thresholds after a Trim bits ± 1 correction with respect to Figure 17.

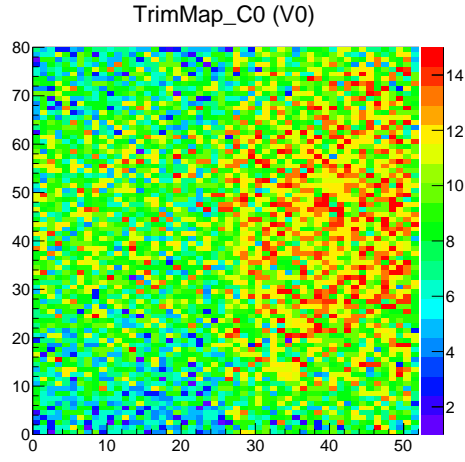


Figure 19: ROC map of the optimized trim bit values (0-15).

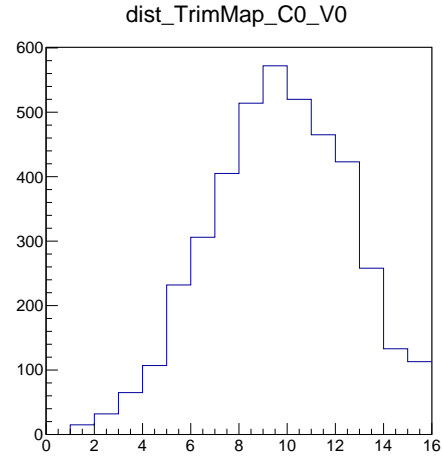


Figure 20: 1D distribution of the optimized trim bit values (0-15).

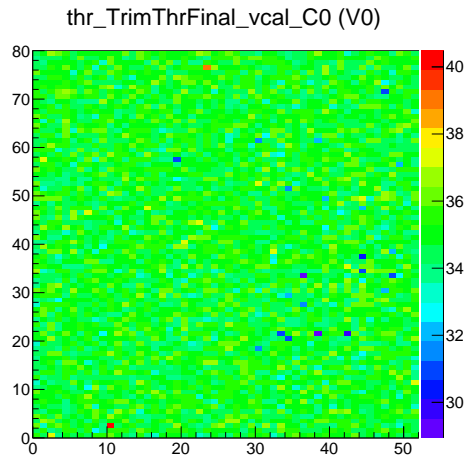


Figure 21: ROC map of the V_{cal} turn-on thresholds with optimized trim parameters.

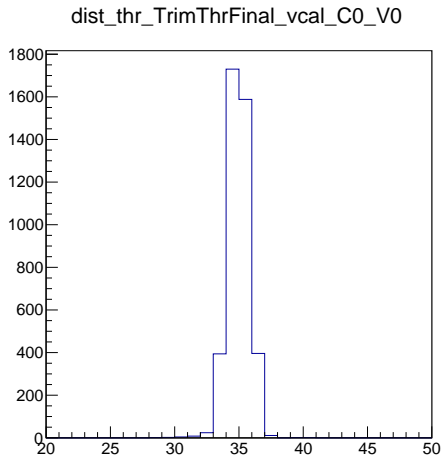


Figure 22: 1D distribution of the V_{cal} turn-on thresholds with optimized trim parameters. [original range 0-255]

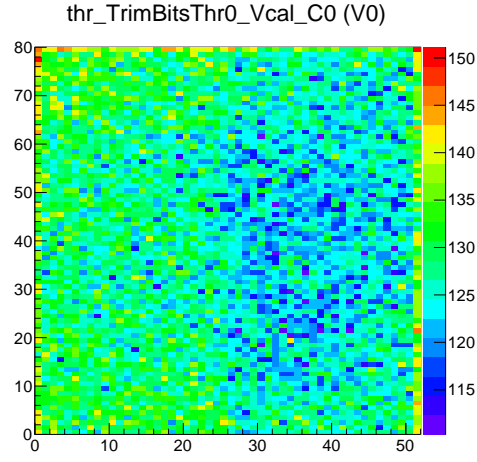


Figure 23: ROC map of untrimmed V_{cal} thresholds. Used as a baseline for the `TrimBits` test.

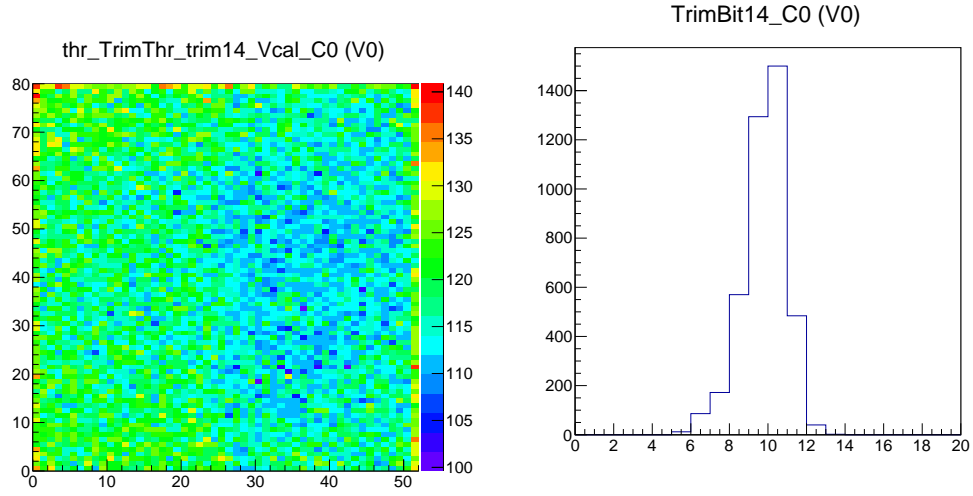


Figure 24: ROC map of V_{cal} thresholds with `Trim bits` = 14 [1110].

Figure 25: 1D distribution of difference of Figure 23 and Figure 24. Pixels with a faulty trim bit would populate the bin at zero.

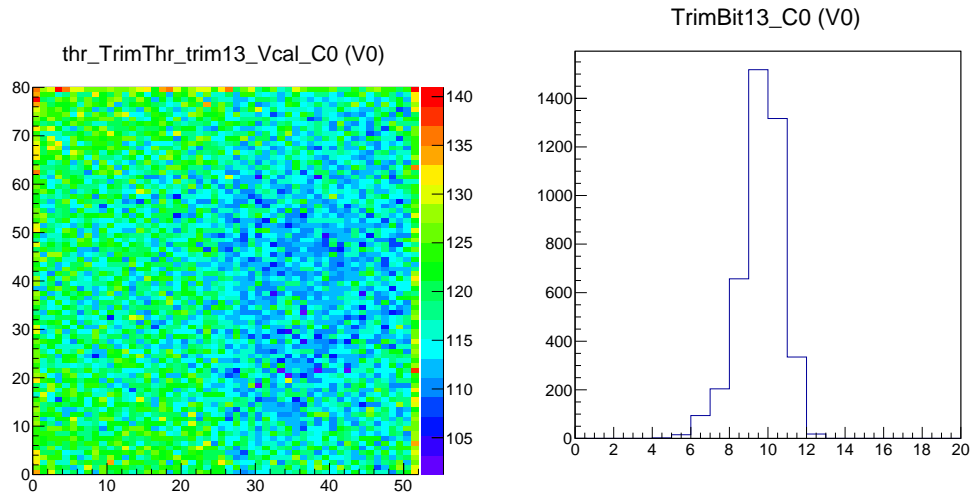


Figure 26: ROC map of V_{cal} thresholds with $\text{Trim bits} = 13$ [1101].

Figure 27: 1D distribution of difference of Figure 23 and Figure 26. Pixels with a faulty trim bit would populate the bin at zero.

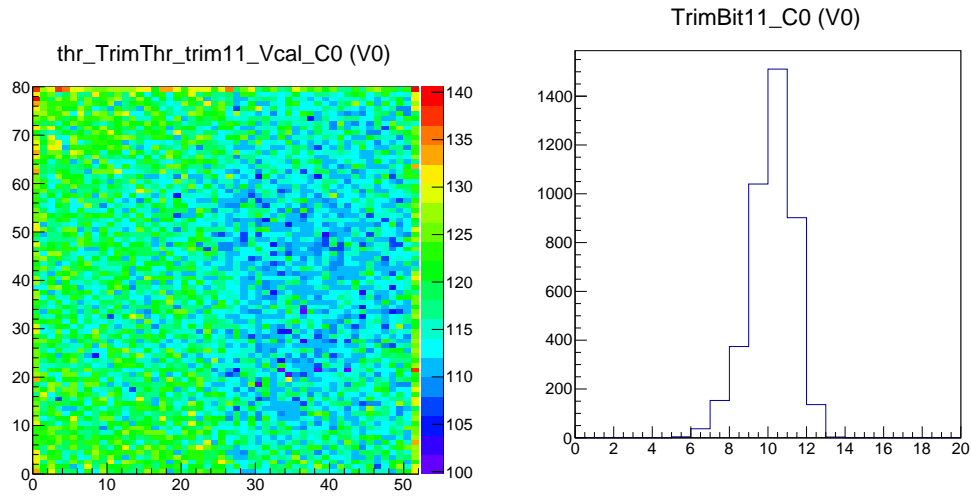


Figure 28: ROC map of V_{cal} thresholds with $\text{Trim bits} = 11$ [1011].

Figure 29: 1D distribution of difference of Figure 23 and Figure 28. Pixels with a faulty trim bit would populate the bin at zero.

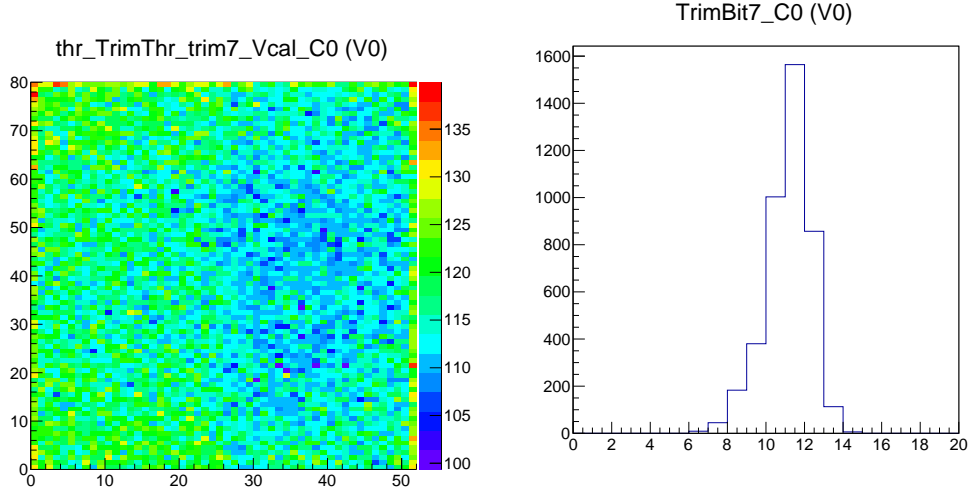


Figure 30: ROC map of V_{cal} thresholds with $\text{Trim bits} = 7$ [0111].

Figure 31: 1D distribution of difference of Figure 23 and Figure 30. Pixels with a faulty trim bit would populate the bin at zero.

4.4 PHOptimization Test

4.4.1 Purpose

The PHOptimization test is responsible for setting an appropriate dynamic range for the 8-bit ADC that digitizes the recorded pulse height. The ADC is located in the Controller and Interface Block of the ROC. It can be seen in the bottom right box of Figure 1. The two DACs used to configure the ADC are PHOffset and PHScale. PHOffset adds a constant offset to the pulse height measurement, while PHScale effectively sets the gain of the ADC. The PHOptimization test is designed to optimize these DACs based on a highly sensitive and highly insensitive pixel in the given ROC, or, in other words, pixels with a very high and very low inherent gain. To use the ADC most effectively, the range of the ADC PH response as a function of V_{cal} needs to be optimized. On the low end, the ADC should provide a PH well above noise for the low-gain pixel near the lowest V_{cal} that registers a hit on it. On the high end, the ADC should saturate for the high-gain pixel at some user-defined V_{cal} well below the maximum V_{cal} the ROC can provide. This allows the signal strength required for saturation to be measured and recorded for all pixels. These two conditions are translated into two constraints used to simultaneously find a solution for the values of PHOffset and PHScale. The constraints applied are 1: the low-gain pixel should have a reasonably large PH (default is 20) at its V_{cal} turn-on threshold, and 2: the high-gain pixel should saturate (PH reaches 255) at a user-defined signal strength (default is $V_{cal} = 100$ (high range)).

4.4.2 Methodology

The `PHOptimization` test proceeds in two steps: First, it identifies a low-gain and high-gain pixel for use in optimizing `PHScale` and `PHOffset`. In this process, care is taken not to choose anomalous pixels. Secondly, it maps the pulse height responses of these two pixels as a function of `PHScale` and `PHOffset`. The maps for these two pixels are then combined to find the optimal values for `PHScale` and `PHOffset`. After the optimal values for `PHScale` and `PHOffset` are set, several validation plots are made and recorded in the output file.

First, the test identifies a low-gain and high-gain pixels using the same methodology for each. Pixels within 3 columns or 5 rows of the edge of the ROC are not considered, unless a suitable pixel cannot be found using those fiducial cuts, in which case those requirements are relaxed. In each case, the PH for each pixel is obtained at a chosen V_{cal} (Figures 32, 34), and a 1D distribution of all PHs is created for each ROC (Figures 33, 35). From this distribution, a pixel is chosen that is separated from the edge of the distribution by a chosen percentile of safety margin. This helps avoid choosing an anomalous pixel. For finding the high-gain pixel, this process is performed with the maximum calibration pulse strength, $V_{cal} = 255$ (high range). The safety margin is configurable, and by default is 98%, i.e. a pixel will be chosen such that 98% of pixels in the ROC have lower PHs. For finding the low-gain pixel, the process is performed with a V_{cal} slightly higher than the V_{cal} trimming target, hard-coded to $V_{cal} = 60$ (low range). For the chosen low-gain pixel, the V_{cal} value is found at which the pixel starts firing, defined as the point in the PH vs. V_{cal} curve at which $PH=10$. This minimum V_{cal} value is recorded for use in the following step of the test.

Once the two pixels of interest have been identified, the test proceeds to the optimization of `PHScale` and `PHOffset`. A 2D scan is performed over `PHScale` and `PHOffset`, returning the PH at each point in the 2D parameter space (Figures 36, 37). For the low-gain pixel, the scan is performed at the minimum V_{cal} found in the previous step. For the high-gain pixel, the scan is performed at the user-defined ADC saturation target (default is $V_{cal} = 100$ (high range)). The optimal regime for the low-gain pixel is when the pulse height value is near to some small safety margin (default 20) above zero. The optimal regime for the high-gain pixel is when the ADC saturates at this V_{cal} , meaning that PH is near 255. Each of these regimes defines a band in the `PHOffset` vs. `PHScale` plane. The final values of `PHScale` and `PHOffset` are chosen from the intersection of these two bands (Figure 38).

With the optimal values of `PHScale` and `PHOffset` set, several validation scans are performed. First, PH vs. V_{cal} curves are produced for the chosen low-gain and high-gain pixels (Figures 39, 40). Additionally, PH maps are produced at three V_{cal} values, one low (defined as 10 DAC units above the minimum effective V_{cal} for the low-gain pixel) (Figures 41, 42), one medium ($V_{cal} = 50$ (low range)) (Figures 43, 44), and one high (chosen ADC saturation target) (Figures 45, 46).

4.4.3 Output

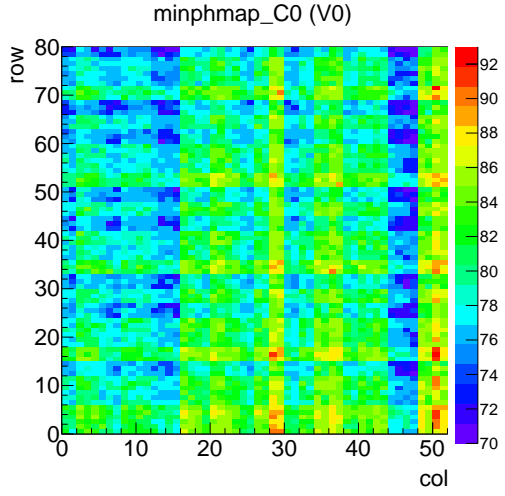


Figure 32: ROC map of pulse heights with $V_{cal} = 60$ (low range).

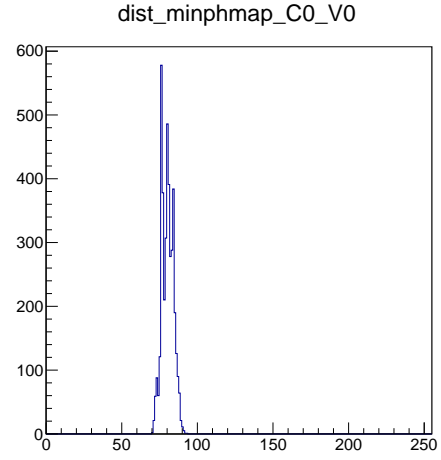


Figure 33: 1D distribution of Figure 32 used to identify low-gain pixel.

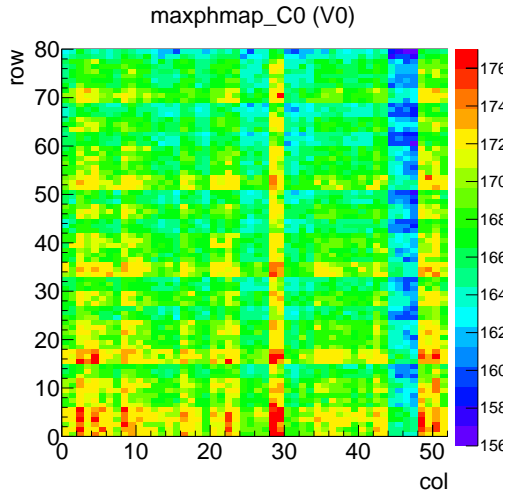


Figure 34: ROC map of pulse heights with $V_{cal} = 255$ (high range).

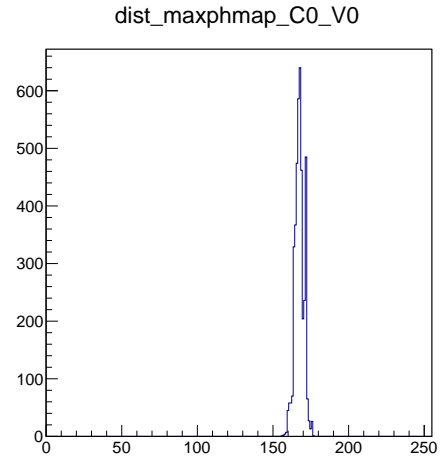


Figure 35: 1D distribution of Figure 34 used to identify high-gain pixel.

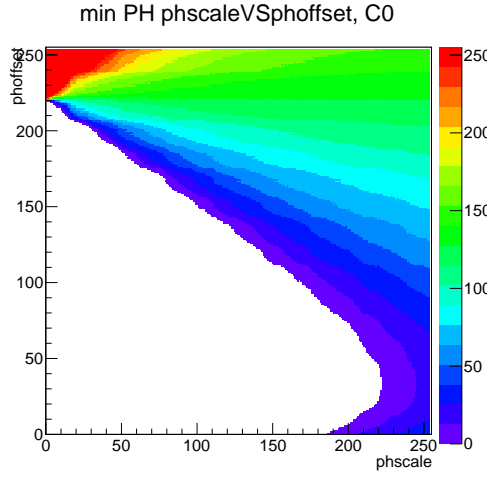


Figure 36: Measured PH in the PHOffset vs. PHScale plane for previously identified low-gain pixel with minimum V_{cal} required to fire the pixel.

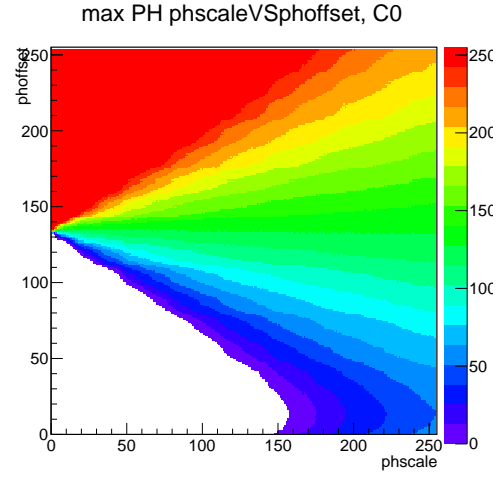


Figure 37: Measured PH in the PHOffset vs. PHScale plane for previously identified high-gain pixel with V_{cal} set to the chosen ADC saturation point.

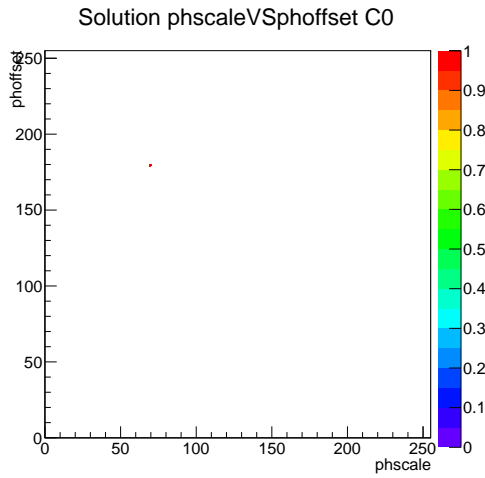


Figure 38: Overlap of optimal regimes in Figures 36 and 37. A red entry signifies an optimal pair of PHOffset and PHScale values.

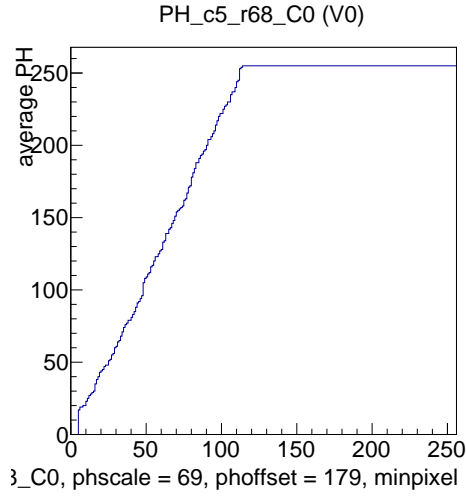


Figure 39: PH vs. V_{cal} curve for previously identified low-gain pixel.

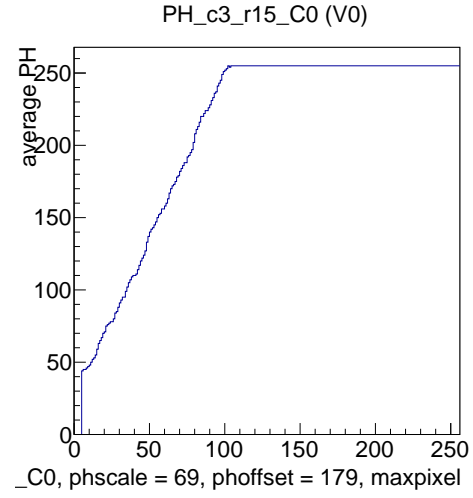


Figure 40: PH vs. V_{cal} curve for previously identified high-gain pixel.

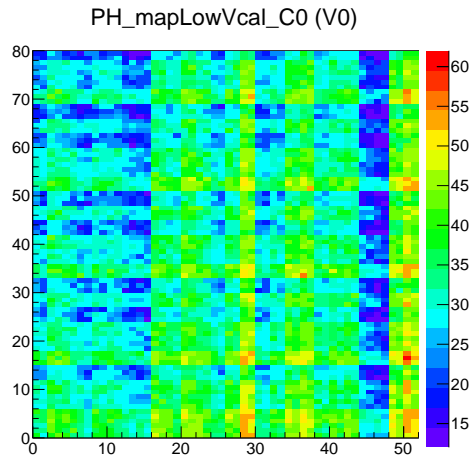


Figure 41: ROC map of pulse heights with V_{cal} 10 units above minimum V_{cal} for low-gain pixel.

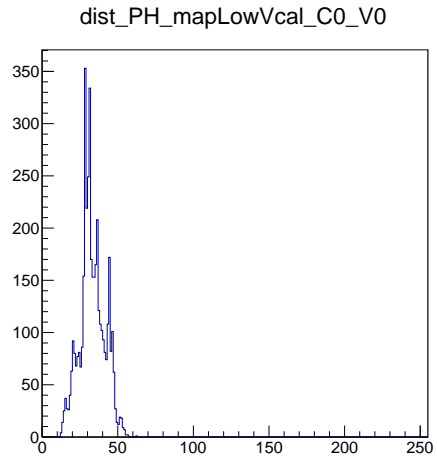


Figure 42: 1D distribution of Figure 41.

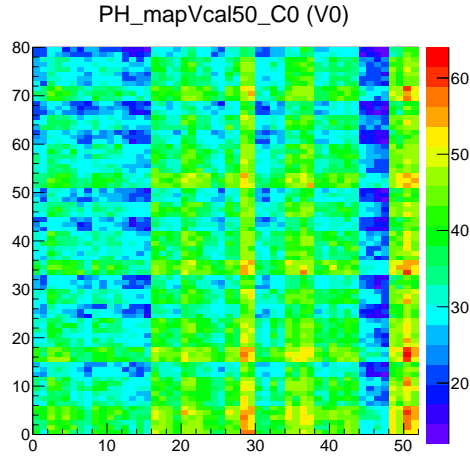


Figure 43: ROC map of pulse heights with $V_{\text{cal}} = 50$ (low range).

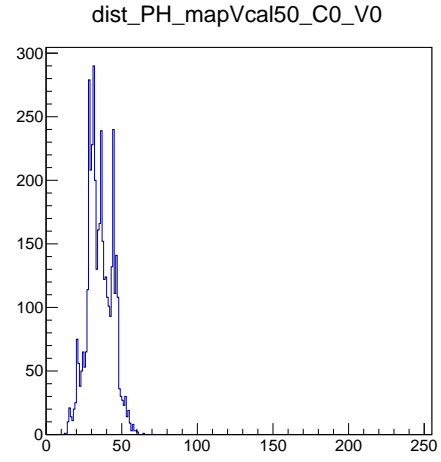


Figure 44: 1D distribution of Figure 43.

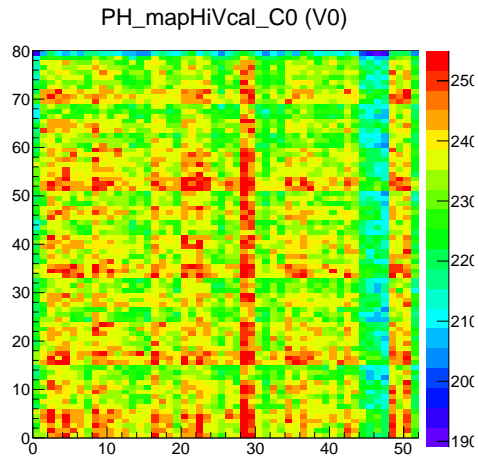


Figure 45: ROC map of pulse heights with V_{cal} set to target ADC saturation point.

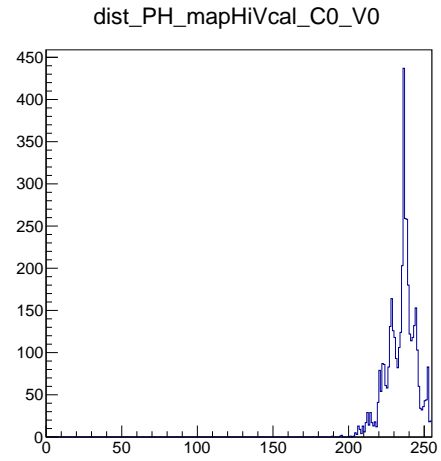


Figure 46: 1D distribution of Figure 45.

4.5 GainPedestal Test

4.5.1 Purpose

The `GainPedestal` test does not alter any DAC parameters, but merely evaluates and records the shape of the pulse height vs. V_{cal} distribution for each pixel. For each pixel, this curve is fitted and the fit parameters are stored for later use. Since variations in gain are expected between pixels in a ROC, the gain must be measured independently for each pixel so that the pulse height to be calibrated back to the input signal size, in units of the V_{cal} DAC (low range).

4.5.2 Methodology

For each pixel, a PH vs. V_{cal} curve is produced. To save time, the pulse height is sampled for a predetermined set of V_{cal} values instead of doing a complete V_{cal} scan. In the low V_{cal} range, the PH is measured in steps of 10 DAC units (configurable), from 10 to 255. Then, in the high V_{cal} range, the PH is measured for values of 30, 50, 70, 90, and 200. The results of these two scans are then combined into a PH vs. V_{cal} curve over the full V_{cal} range. This curve converts the high range V_{cal} values to their larger low range equivalents by multiplying by a factor of 7. This curve is then fit to an error function and the four fitted parameters and their errors are recorded. Parameter 0 corresponds to the V_{cal} value at the center of the error function, when the PH is halfway between zero and saturation (255). Parameter 1 is proportional to the width of the turn-on and is therefore inversely related to the gain of the pixel. Parameter 2 shifts the error function upwards, with a value of unity moving the floor of the function to zero. Parameter 3 corresponds to half the height of the function, and should be near 127.5 (255/2). The ROC-wide distributions for these four parameters and their relative errors are shown in Figures 47-54. Additionally, these parameters are stored for later use in text files with names of the form “phCalibrationFitErr35.C#.dat” (assuming the module was trimmed to $V_{cal} = 35$). Finally, the linearity of each pixel’s response is estimated. Using the fitted error function, the true integral of the function in the V_{cal} range from 0-200 (low range) is compared to a linear approximation. The quotient of these two values should be near unity for a linearly-responding pixel. The ROC-wide distribution of this quantity, dubbed the “non-linearity”, is shown in Figure 55.

4.5.3 Output

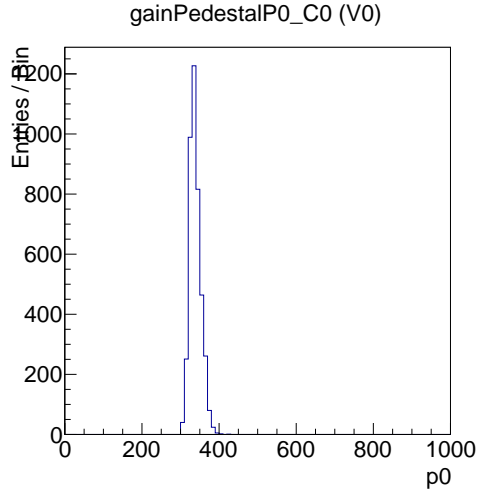


Figure 47: Distribution of fitted values of parameter 0, the V_{cal} center of the fitted error curve.

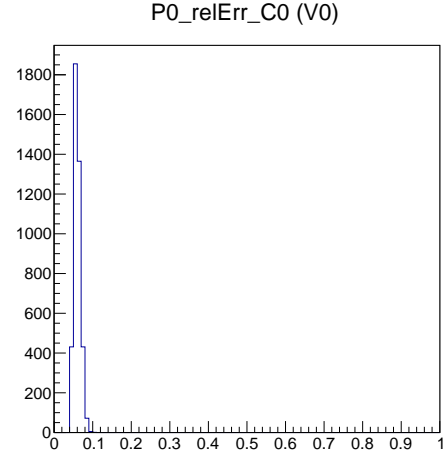


Figure 48: Relative errors on the fitted values of parameter 0.

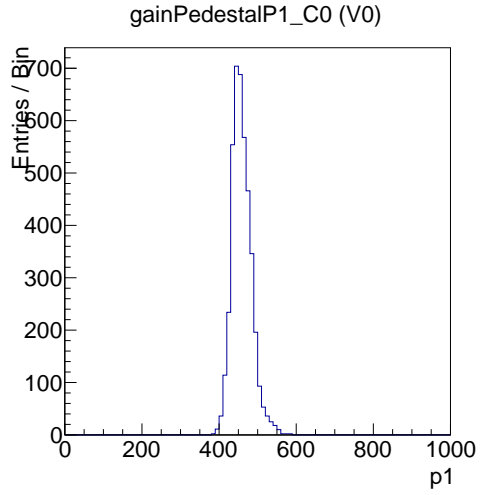


Figure 49: Distribution of fitted values of parameter 0, related to the error function width.

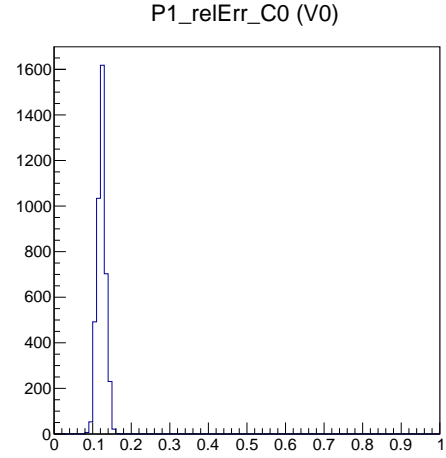


Figure 50: Relative errors on the fitted values of parameter 1.

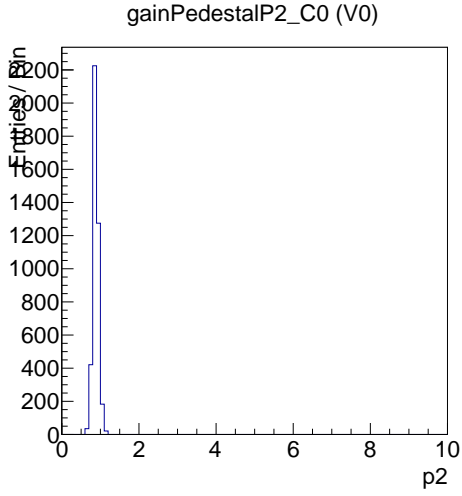


Figure 51: Distribution of fitted values of parameter 2, the vertical shift in the error function.

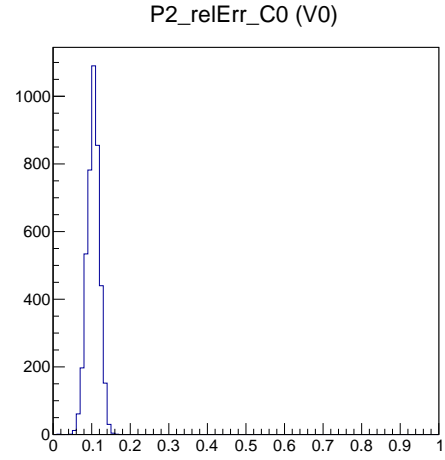


Figure 52: Relative errors on the fitted values of parameter 2.

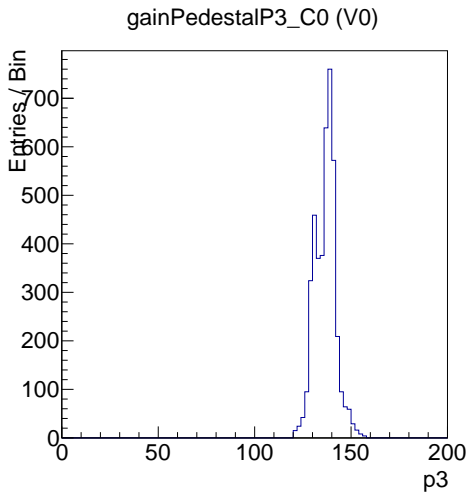


Figure 53: Distribution of fitted values of parameter 3, equal to half the height of the error function plateau.

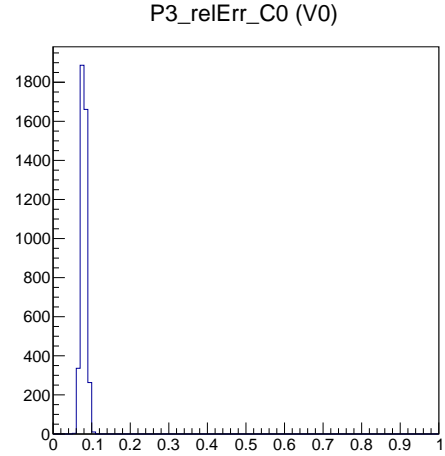


Figure 54: Relative errors on the fitted values of parameter 3.

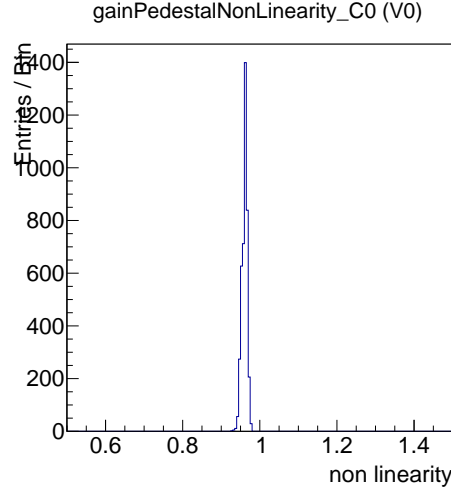


Figure 55: Non-linearity is defined as the ratio of the integral of the fitted error function to the integral of a linear approximation, in the V_{cal} range from 0-200 (low range).

4.6 SCurve Test

4.6.1 Purpose

The **SCurve** test as configured for FPix module testing has two primary purposes. First, it evaluates the efficacy of the **Trimming** test. Secondly, it measures the noise present in each pixel and can potentially flag abnormally noisy pixels. Both of these goals are achieved by fitting the curve of efficiency vs. V_{cal} for each pixel. The test gets its name from the shape of the error function used to fit the measured curve.

4.6.2 Methodology

An **SCurve** test is a generic type of test that measures a pixel's efficiency as a function of a chosen DAC. Assuming the pixel will not respond at low values of the DAC in question and will be fully efficient at some higher value of the DAC, the efficiency curve should have a region of zero efficiency, a transition region with increasing efficiency, and a plateau region with 100% efficiency. The shape of this curve mildly resembles the letter "S". If no noise is present in the system the turn-on curve will be a simple step function, with the horizontal floor and plateau connected by a vertical line. However, noise will reduce the slope of the line connecting the floor and plateau. Using this fact, an estimate of the noise in the system can be extracted from the width of the turn-on curve.

For each pixel, the s-curve is fit with an error function, which is the cumulative distribution function for a Gaussian curve. From this curve, three parameters are extracted. The turn-on threshold, defined as the DAC value at which 50% efficiency is reached, is

denoted by the shorthand “thr”. The width of the turn-on curve, defined as the width parameter of the fitted error function, is denoted by the shorthand “sig”. Finally, the turn-off threshold, defined as the point above the turn-on threshold at which efficiency drops back below 50%, is denoted by the shorthand “thn”.

For the **SCurve** test as configured for FPIx module testing, an **SCurve** test is performed over V_{cal} , in a V_{cal} range window around the target threshold for the **Trimming** test. This is because, post-trimming, the expected region of the turn-on of the s-curve is known. The number of triggers is configurable, but in order to extract accurate width values, a large number of triggers (default is 200) should be used.

4.6.3 Output

Figures 56-61 show the output of the **SCurve** test as used in the **FullTest** for FPIx module qualification. ROC maps and 1D distributions of the three parameters measured by the test are saved in the output file: the turn-on “thr” (Figures 56, 57), the width “sig” (Figures 58, 59), and the turn-off “thn” (Figures 60, 61). For s-curves with respect to V_{cal} , no turn-on is expected (it’s mainly useful for $V_{thrComp}$ s-curves). When no turn-off is observed, the test returns the maximum value of the DAC in question that is tested.

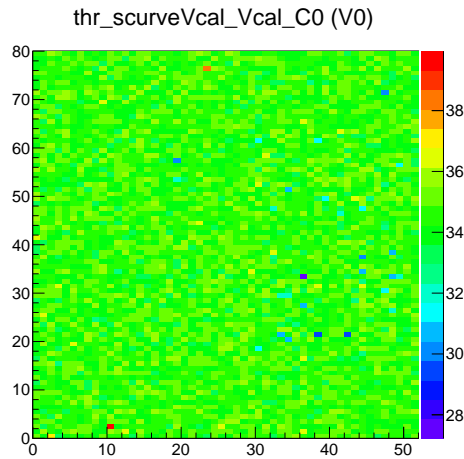


Figure 56: ROC map of the V_{cal} s-curve turn-on thresholds. Values should be near the trim target (default 35).

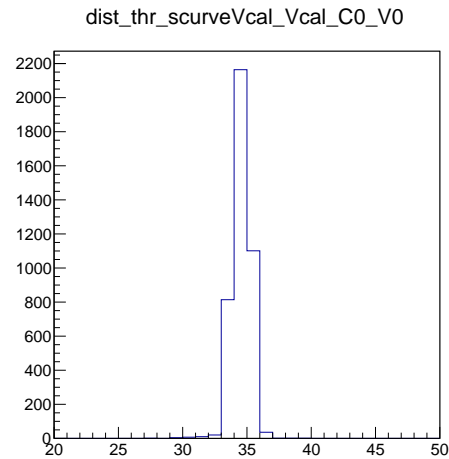


Figure 57: 1D distribution of Figure 56. [original range 0-255]

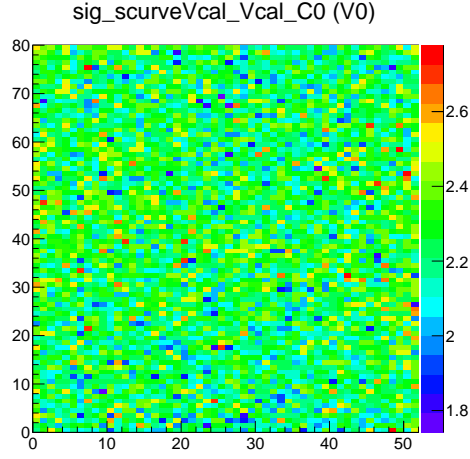


Figure 58: ROC map of the V_{cal} s-curve turn-on widths. This width is proportional to the noise in the system.

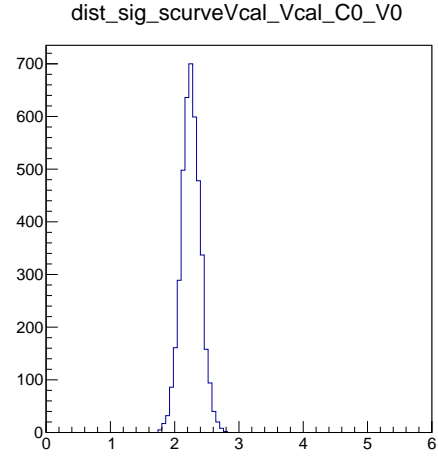


Figure 59: 1D distribution of Figure 58.

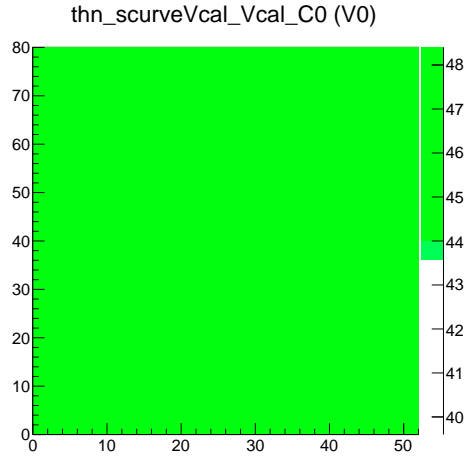


Figure 60: ROC map of the V_{cal} s-curve turn-off thresholds. For s-curves with respect to V_{cal} , no turn off is expected. Therefore this plot should be uniformly distributed at the maximum value of V_{cal} considered by the test.

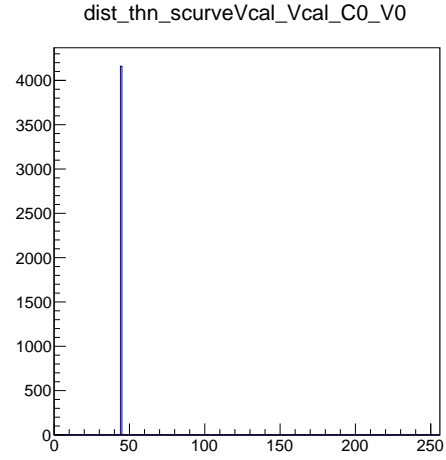


Figure 61: 1D distribution of Figure 60.

4.7 BumpBonding Test

4.7.1 Purpose

The **BumpBonding** test is meant to flag any problems with the bump bonds that connect the sensor to the ROCs and form the path for the charge generated in the silicon sensor to be recorded by the ROCs. This test takes advantage of a parallel route designed for the calibration signal in the ROC. This route can be seen at the top left of Figure 1, connecting the source of the V_{cal} calibration pulse to the “top metal pad” by closing the switch labeled “sensor calib” instead of the standard “calib” switch. The “top metal pad” is located near the edge of the ROC on the side facing the sensor. The calibration pulse can pass via capacitive coupling from this pad, across the ROC-sensor air gap, and into the silicon sensor. The calibration pulse then makes its way back into the ROC via the bump bond and can be read out as normal. Using this signal path that contains the bump bond, the presence of the bump and the quality of the electrical connection it establishes between the sensor and the ROC can be tested.

Currently in pXar there are four variants of bump bonding tests, optimized to different module types. For FPix modules, the standard test is the BB3 test. This document focuses on this specific **BumpBonding** test version.

4.7.2 Methodology

The **BumpBonding** test is essentially an **SCurve** test with respect to V_{thrComp} , with the calibration signal routed through the sensor instead of directly to the readout electronics. V_{cal} is configurable and by default is set to 250 (high range). In any case, V_{cal} should be set quite high since a large signal loss is expected going through the air gap from ROC to sensor. The number of triggers used is also configurable (default is 5). From the V_{thrComp} s-curve for each pixel, the turn on value is extracted, defined as the value of V_{thrComp} for which the efficiency surpasses 50%. For pixels with good bump bonds, these thresholds should be distributed normally, i.e. as a Gaussian curve. The distribution of the V_{thrComp} turn-on is fitted to a Gaussian, the mean and width of which are recorded. In this fit, peaks near 0 or 255 are excluded, since these most likely come from dead pixels. Since variations in the mean of this turn-on are observed in odd and even columns, this fitting is done separately for the odd/even columns in the ROC. Pixels that turn on at much higher than average V_{thrComp} values have higher than average signal loss. Leveraging this fact, pixels with turn-on values more than $5 \times \sigma$ above the mean are flagged as bad.

4.7.3 Output

Output from the **BumpBonding** test is shown in Figures 62-66. Figure 62 shows a ROC map of the raw turn-on values extracted from the V_{thrComp} s-curve for each pixel. Figures 63 and 64 show the 1D distributions of the even and odd columns in Figure 62, along with their best-fit curves and arrows located $5 \times \sigma$ above the mean of the curve. The means of these distributions are noticeably different, which is why these two distributions

are fit separately. These distributions are then recombined, but since they have different means and widths, it doesn't make sense to combine them directly. Instead, each entry is scaled to represent the number of standard deviations it is away from the mean. We refer to this scaled value as the fit “residual”. Figures 65 and 66 show a 2D ROC map and a 1D distribution of these fit residuals, respectively. Any pixels above the black arrow in Figure 66 are flagged as anomalous bumps.

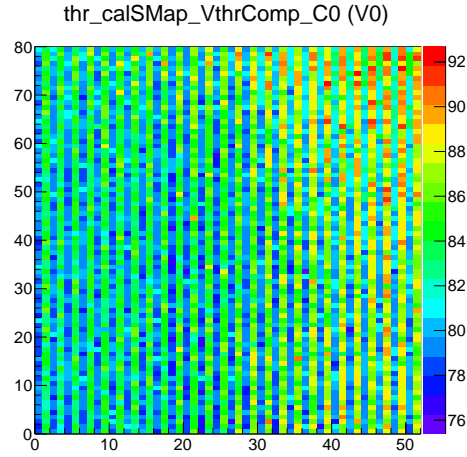


Figure 62: ROC map of the raw V_{thrComp} turn-on values when passing the calibration pulse through the sensor.

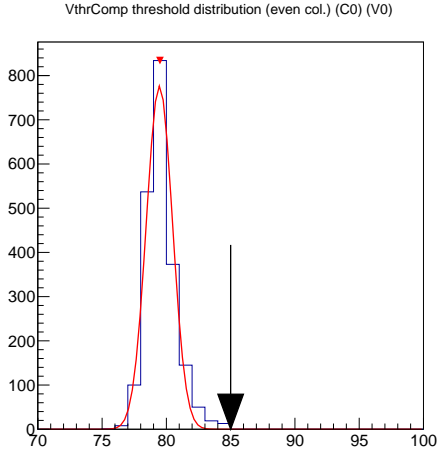


Figure 63: 1D distribution of the even columns in Figure 62. The red curve is a Gaussian fit to the data. The black arrow corresponds to 5σ above the fitted peak. [original range 0-255]

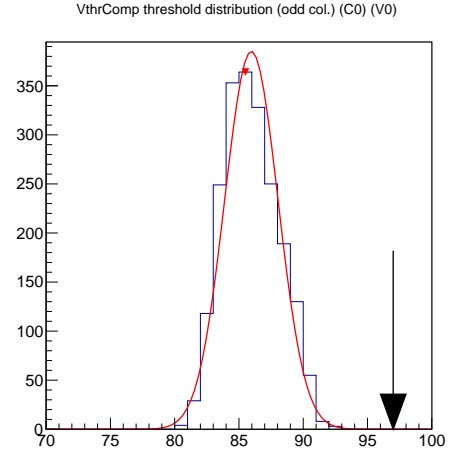


Figure 64: 1D distribution of the odd columns in Figure 62. The red curve is a Gaussian fit to the data. The black arrow corresponds to 5σ above the fitted peak. [original range 0-255]

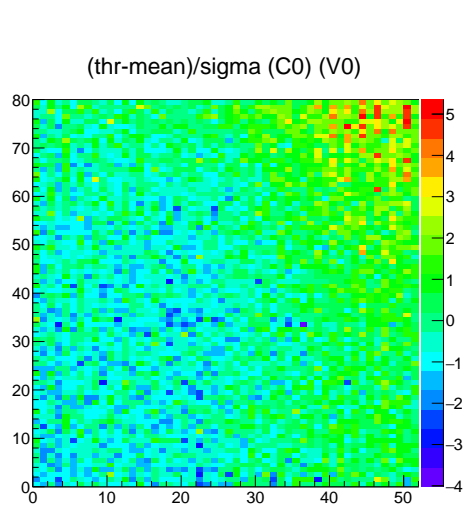


Figure 65: ROC map of the residuals of the V_{thrComp} turn-on values, calculated separately for even and odd columns.

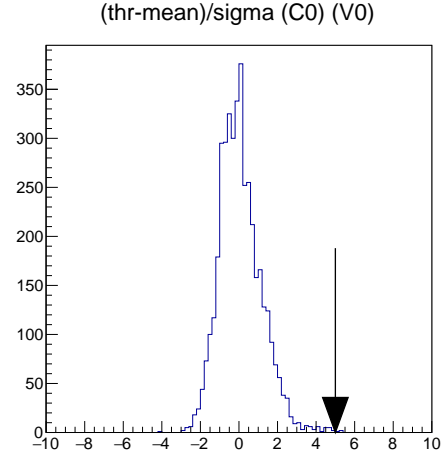


Figure 66: 1D distribution of ROC map in Figure 65. Pixels with turn-on values more than 5σ above the ROC mean (marked by the black arrow) are flagged as faulty. The first (last) bin of the plot contains the underflow (overflow) entries.

4.8 IV Test

4.8.1 Purpose

The IV test is unique in that it doesn't directly utilize the read-out functionality of the ROC. In fact, this test can be performed on a sensor before it's even bump-bonded onto a set of ROCs, or even before the sensor has been cut out of the original silicon wafer. In principle, the silicon sensor acts like a diode, allowing current to flow in the "forward" direction but not in the "reverse" direction. During operation, a potential difference is created across the sensor, in the plane transverse to the sensor face, by applying a "reverse bias" voltage. This potential difference is what draws the electrons created by a charged particle passing through the sensor towards the bump-bond to be collected. For a given sensor, this reverse bias has an operating range. If the bias is too small, the sensor will not be depleted of free electrons and will not operate properly. If the bias is too large, the sensor will break down and start behaving like a resistor instead of like a diode. The IV test is meant to determine the operating range of the sensor, bounded by the depletion voltage (lower limit) and the breakdown voltage (upper limit).

4.8.2 Methodology

This test is exceedingly simple, it merely performs a scan over a range of bias voltages and records the current draw of the system at each point. During the test, control is taken over the HV supply in order to effect a voltage sweep. The user can configure the range of voltages to scan (default is 0V-600V), the step size between measurements (default 10V), and the amount of delay time to allow the system to react to the voltage change and stabilize before taking a measurement (default is 1s). To protect the module from damage due to excessively high voltages, a compliance current is set (default is 100 μ A). If the current reaches this compliance level, the scan is aborted and the voltage turned off. The single output from this test is a plot of the current draw as a function of the bias voltage as shown in Figure 67. The module in question has an operating plateau from roughly -100V to -200V.

4.8.3 Output

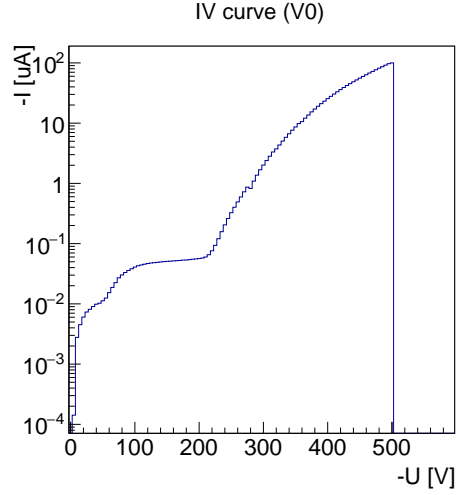


Figure 67: Current draw in the module as a function of (reverse) bias voltage. The module shown is fully depleted by -100V and shows a breakdown voltage near -200V.

4.9 FullTest

The suite of tests that is run to qualify forward pixel modules is called the **FullTest**. It is a sequential test suite composed of the following tests that are described in more detail in the previous sections.

- **PreTest**: configures the basic DACs to ensure that the ROC is in a functioning state.
- **PixelAlive**: checks that all pixels respond to an input signal and have the correct addresses associated with them.
- **Trimming**: sets all pixels to turn on with the same input signal strength
- **PHOptimization**: optimizes the dynamic range of the output pulse heights as calculated by the ADC serializer
- **GainPedestal**: measures and stores the gain of each pixel and checks the linearity of the ADC response
- **SCurve**: validates the trimming and measures pixel noise
- **BumpBonding**: flags potential issues in bump bond quality

In addition to the **FullTest**, each module undergoes an IV test that measures leakage current as a function of the reverse bias voltage. For technical reasons described in Section 5, this test is performed separately from the **FullTest**.

5 Module Testing at FNAL

5.1 Experimental setup

All module testing at FNAL takes place at the Silicon Detector laboratory (SiDet). There are two identical FPix module testing stands at SiDet situated next to each other. Figure 68 shows one such test stand. Up to four modules can be tested in parallel. The modules are connected via copper cables to adapter cards with SMK connectors. The adapter cards convert the data into SCSI format. A SCSI cable from each adapter card transfers the data to a digital test board (DTB), which is in turn connected via USB to a PC. There are two power supplies in the setup. The first provides the bias voltage to the sensors under test by way of a four-way splitter (top box in Figure 68). The other provides power to the cooling system of the coldbox (bottom box in Figure 68). The water chiller, vacuum, and coldbox work together to control the temperature of the modules under test. Each test stand also has a dry air (N_2) line that connects to the coldbox to help control the humidity inside it.

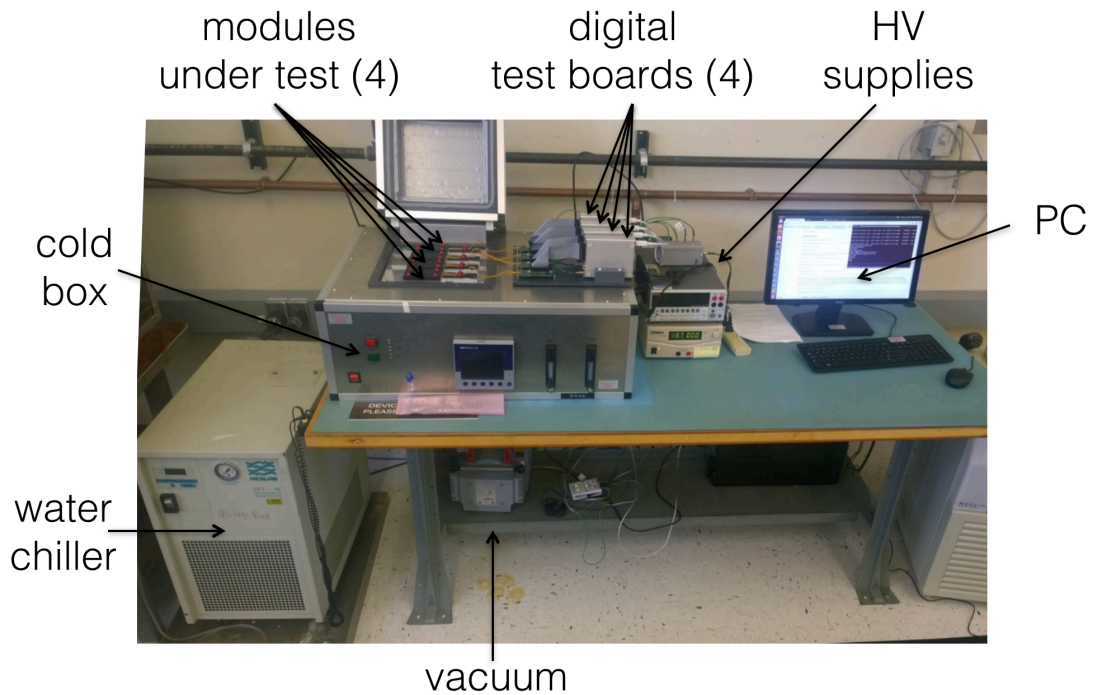


Figure 68: An annotated photo of one of two identical module testing setups at SiDet.

5.2 Safety concerns

There are several safety precautions that should be taken to minimize potential damage to the modules. Generally speaking, there are three sources of potential damage: physical damage, electrostatic discharge (ESD), and damage due to condensation. To avoid these hazards, the following guidelines should be followed whenever modules are involved.

- The module cover protects the HDI and sensitive wirebonds from mishandling. Do not remove it under normal testing circumstances.
- Likewise, the lid of the coldbox protects the modules from the afore-mentioned damages. Never open the cover while module testing is taking place. In fact, the handle of the coldbox lid will shock you if you touch it during testing, and it will hurt a lot. Once module testing is complete, wait until the coldbox has reached room temperature before removing the modules.
- Whenever handling a module, make sure you are grounded via the provided wrist straps and bracelets. There are also ESD-safe mats on the surface of the test stand table and on the floor underneath the test stand.
- When not being tested, modules should be placed in the pink ESD-safe sleeves and stored in one of the designated cabinets. These cabinets are fed with dry air to maintain low humidity.

5.3 Testing procedure

The standard test sequence for modules is the **FullTest**, as outlined in Section 4.9, following by an **IV** curve. Each module will be tested once near room temperature (17 C) and once closer to the operating temperature of the FPix in CMS (-20 C). The first 100 production-level modules all will undergo X-ray testing in between the **FullTest** at 17 C and the **FullTest** at -20 C. After the first 100, only a subset of modules (roughly 10%) will undergo X-ray testing.

The **elComandante** software package is used to facilitate parallel module testing. More information can be found here:

<https://twiki.cern.ch/twiki/bin/viewauth/CMS/ElComandante>

This package allows simultaneous control of all the dynamic elements of the test setup (four DTBs, HV supply, and coldbox). Once modules are installed in the coldbox, the water chiller is turned on (default water temperature is 10 C) and the coldbox is cooled to the desired temperature. Once the temperature has stabilized, the standard procedure is to run the **FullTest** in parallel on four modules, followed by four individual **IV** curves. The **IV** test doesn't directly use the DTB, it simply reads the current directly from the power supply. Since the same HV supply is used for all modules under test, parallel **IV** scans are not possible. When all tests have finished, the coldbox automatically heats (if necessary) until room temperature is reached. At this point the modules can be safely removed from the coldbox and placed in storage.

Specific instructions for the shifters who will be testing modules can be found here:

<https://twiki.cern.ch/twiki/bin/viewauth/CMS/FPIXUpgradeShifterInstructions>.

6 Test Result Databases

Once a module has been tested, the resulting information is uploaded onto two remote servers: One for visualization and one for documentation. The shifters will execute provided scripts that format the test results appropriately and upload them to both of these servers. The visualization is achieved via the **MoReWeb** software package which processes the test results and creates a set of html pages to help navigate through the results of a given module. This server can be accessed here:

<http://www.physics.purdue.edu/cmsfpix/MoReWeb/Results/Overview.html>

The documenting of test results and grading of modules is done by the Purdue FPix database, found here:

http://www.physics.purdue.edu/cmsfpix/Submission_p/index.php

This database tracks module components from their origins through their inclusion in FPix modules. It also stores all the configuration files and output root files generated during module testing. It has the functionality to search through the entire set of constructed modules, filtering on grade, location, etc.

7 Module Grading Scheme

Each FPix phase 1 upgrade module is given a grade based on various aspects of the testing results. The current grading scheme is preliminary and expected to evolve through the production process to account for specific component failure modes seen during production.

The current scheme is based on two categories (IV performance and single pixel defects) although it can be expanded to include X-ray results and many of the other **FullTest** measurements.

The IV performance criteria are two-fold. First, the module is graded on the amount of leakage current at the expected operating bias (-150V). Modules must have $I_{-150V} < 2\mu A$ to receive an A or $I_{-150V} < 10\mu A$ to receive a B. Modules with $I_{-150V} > 10\mu A$ receive a C. Secondly, the breakdown voltage for the module should be higher (more negative) than -150V. This is determined by looking at the ratio I_{-150V}/I_{-100V} . If the module has not reached the breakdown voltage by -150V, this ratio should be small, otherwise it should be large. Modules with $I_{-150V}/I_{-100V} < 2$ receive an A for this criteria, all other modules receive a C. The overall IV grade for the module is the worse of these two assigned grades (leakage current and breakdown voltage).

Single pixel defects are identified using the results of the **PixelAlive** and **BumpBonding**

tests. A sum per ROC is calculated of all pixels flagged as faulty by either of these tests. If this sum is less than 1% of pixels in the ROC, the ROC receives an A grade. Otherwise, if the sum is less than 4% of pixels in the ROC, the ROC receives an B grade. If not, the ROC receives a C grade. The module grade for single pixel defects is assigned as the grade of the worst ROC.

The final module grade is assigned as the worse of the IV grade and the pixel defect grade. In this way, each module is given either an A, B, or C grade that is recorded in the documentation server.