ATLAS IBL Pixel Module Electrical Tests

ATLAS Project Document No:	Institute Document No.	Created: 10/05/2012	Page: **1 of 41**
ATL-IP-QA-0031	***1221585***	*Modified: **06/01/2013***	*Rev. No.:* **2.2**

ATLAS IBL Pixel Module Electrical Tests Description

Abstract

This document provides a detailed descriptions of the electrical test carried on the ATLAS IBL pixel modules in different stages of the assembly of the ATLAS IBL.

Prepared by:	*Checked by:*	*Approved by:*
F. Hügging (ed.)	**C. Gemme**	**Maurice Garcia-Sciveres**
M. Backhaus	**H. Pernegger**	**Tayfun Ince**
	D. Ferrere	**Daniel Dobos**
	M. Garcia-Sciveres	**Alessandro La Rosa**
	G. Darbo	

Distribution List

History of Changes

Rev. No.	*Date*	*Pages*	*Description of changes*
1.0	30/05/2012		First issue.
1.1	03/06/2012		Small corrections and comments from Maurice.
1.2	04/06/2012		Comments from Giovanni and Heinz
2.0	06/01/2013	38	Second version with emphasis on data analysis & module selection
2.1	16/04/2013	38	Table for module penalties added
2.2	28/05/2013	41	Version for approval of the module selection

Table of Contents

1 INTRODUCTION

The purpose of this document is to provide a detailed description of the electrical tests to be performed on ATLAS IBL modules. These tests are part of a full quality control procedure for the production of the ATLAS IBL Detector.

These tests will be performed in different stages of the assembly procedure and item subject to them are:

Modules on Flex Frame (either 3D SC or planar DC)

Modules cut from Flex Frame (either 3D SC or planar DC)

For brevity all module assemblies will be collectively called *dressed modules*.

IBL modules will be either single chip modules (SC) consisting of one FE-I4 chip bump bonded to 3D sensor or double chip modules (DC) consisting of 2 FE-I4 chips bump bonded to one planar sensor tile. Detailed specification of the bump bonding process can be found in ATL-IP-CS-0029, the assembly process is described in ATL-SYS-AN-0001.

ATLAS IBL modules will need to operate for several years at the LHC, in a very harsh radiation environment, with essentially no opportunity to repair modules developing a failure. Sufficient information must be gathered for each module, in order for the selection procedure to select the modules which are more likely to operate in ATLAS with acceptable efficiency and performances.

A list of items for which test information is desirable is given in Table 1. Electrical tests have more sensitivity to some items than others. Mechanical and other non-electrical tests with different sensitivity are mentioned in this document for completeness.

Most of these properties may be affected by the environmental conditions, assembly operations and degradation with time, therefore must be assessed immediately after assembly, after thermal cycling and burn-in of single modules, after mounting on staves and after thermal cycling and burn-in of staves.

Table 2 presents the detailed breakdown of the tests and at which stage of the production they are performed. This table refers to the list in Table 1 to indicate which kind of information is gathered by each test. For completeness, mechanical as well as electrical tests are included in the table, but only the electrical tests, which are underlined in the table, are discussed in the rest of the document.

It is not excluded that reduced tests will be performed at some intermediate stage of the assembly, to check the success of specific operation like wire-bonding or potting. The execution and treatment of these tests is left to the laboratory responsible for module assembly, the only requirement is that they do not interfere with the standard sets described in this document.

Since the headers of the column are somewhat conventional, here follow the specification of when a set of measurement is performed:

Initial electrical tests (ASSY, BURN) cover the measurements needed to have complete room temperature comparison before and after module burn-in and thermal cycling, in order to detect infant mortality or bump failures pre and post burn-in tests are to be done in the same lab were burn-in is performed.

Full electrical characterization (FLEX) covers the full module characterization at the operating temperature of -15 °C. It will be the base for module sorting.

Receiving module tests (RECV) are fast functionality tests that should be performed any time a mechanical operation on a module is done (assembly, repair, shipping…) to check the module has not been damaged. This includes functionality tests immediately after the assembly, unless the assembly site is also a burn-in site and goes directly to the *Initial electrical tests*, and functionality tests before cutting the module from the flex frame.

Initial loading tests (LOAD) are fast functionality tests performed on modules immediately after the mounting on staves. At the end of the module loading to the stave all wings of the stave flex will be bended, glued and finally the wire bond connection to the module is made. So testing of the modules is only possible after the entire stave loading process is finished. At this stage either each module can be tested individually with the same test setup as for individual modules or another setup can be used which allows a parallel testing of all modules.

Electrical characterization of staves (STAVE) is performed at operating temperature at the end of the stave burn-in. Its aim is to check any degradation of module performances (early breakdown, increased noise, bump failures…) caused by thermal stresses after the module has been glued on the thermal support.

Table 1: information needed by the quality control procedure

1	**Mechanical issues**
1a	damage or defects on sensor or read out electronics
1b	wire bonding defects
1c	not compliance with design dimensions
1d	proper thermal contact with local support
2	**Sensor Properties**
2a	breakdown voltage behaviour
2b	leakage current distribution
3	**Bump bond quality**
3a	number of disconnected bonds
3b	number of shorted or almost-shorted pairs
4	**Performances**
4a	operational readout speed (40 or 160 Mbps per chip)
4b	number of electronically dead or unusable channels
4c	threshold tunability (at 1500 - 3000 e) in working conditions
4d	noise in working condition
4e	ToT tunability (ToT for one m.i.p. at 5) in working conditions
4f	number of noisy channels in working conditions
4g	Efficiency
4h	operation in corner conditions

Table 2: Module test list

Measurement	Information	Initial electrical tests			Full electrical characterization	Module receiving tests	Initial loading tests		Electrical test of stave and sector
(status code)		ASSY		BURN	FLEX	RECV	LOAD		STAVE
Optical inspection	**1a**					r			
Envelope check	**1c**	r							
X ray	**3ab**								
I-V scan	**2a**	r		r	c	r			c
Power regulators	**1ab**	r				r			
High current operation	**1ab**	r		r	c	r			
Digital test	**4ab**	r		r	c	r	r		c
Threshold tuning	**4bc**	r			c				c
ToT tuning	**4e**	r			c				c
Threshold scan HVon	**4bcd**	r	BURN-IN	r	c	r	r	BURN-IN	c
Power consumption	**1ab**			r	c				c
Threshold scan HVoff	**3a**	r		r	c	r	r		
Minimal threshold operation	**4c**			r	c				
Crosstalk	**3ab,4bg**	r		r	c				
Monleak	**1a,2ab**				c				
Timewalk	**4g**				c				
In-time Threshold scan	**4g**				c				
Red. operating range	**4h**	r							
Operating range	**4h**				c				
Noise suppression	**4fg**	r			c				c
Source scan	**3ab,4fg**	r			c				c
Thermal imaging	**1d**						r		r
Wire bond pull-test	**1b**	r							

c = at -20- -10 C on NTC
r = at 20-25 C on NTC

2 Operating conditions

Nominal working conditions for operation in ATLAS would be:

Vin 2.0 V,

VDET -30 V - -200V following the evolution of radiation damage for 3D sensor modules

VDET -80 V- -1000 V following the evolution of radiation damage for planar sensor modules,

module temperature, as measured on the module NTC, –20°C.

Therefore a detailed module characterization will be performed in such conditions.

This requires the module or the whole local support to be placed in an environmental chamber or a simple isolated box and cooled in dry atmosphere.

The mentioned bias voltages are the one expected at the dressed module level. IBL modules use 2 on chip power regulators generating the analog and digital supply voltage. The nominal input voltage V_{in} for proper operation of the power regulators is 2 V. No sensing of V_{in} is foreseen on the module level but on the adapter board, which connects the test system to the module a sense point on the flex side of the connector is available. The voltage drop from this sense point to module is expected to be 0.1 V. Thus nominal operating voltage should be increased by that amount.

Inside IBL the modules have to transmit the data electrically via LVDS over a distance of about 6 – 8 m. Therefore at least one the electrical shall be done with long electrical cables to verify the correct operation of the LVDS output transmitters and input receivers in realistic conditions.

Many functionality tests can be performed at room temperature, defined as a temperature between 20 and 25 C as measured on the module NTC. This may require no cooling if a single module or, if modules on local supports, when only one module is tested at the time. However, for production testing of single modules, a cold box is recommended even for room temperature tests, so that the module temperature can be uniformly controlled.

During all electrical tests the modules is supported by an aluminum plate, which is in direct connection to the FE-Chip backside and acts therefore as ESD precaution. The plate shall be grounded during the tests to minimize the risk of damage due to electrostatic discharge. The operator shall wear an ESD bracelet while handling the modules.

It is also of interest to know the operational margin a module has, to avoid using modules with a too narrow working range around the nominal conditions.

The operational margin will be determined repeating some of the tests shown in Table 2 in different conditions, according to the procedure defined in the devoted subsection.

3 USBpix setup

Module electrical characterization will be performed using a test system custom built by the ATLAS Pixel Collaboration. This setup and its performance is described in detail in “Development of a versatile and modular test system for ATLAS hybrid pixel detectors”. It will be referred as USBpix setup. For completeness a layout of the system is given in order to easily refer to specific equipment in the test description.

The USBpix test system is a modular test system in terms of hardware and software. The hardware consists of a Multi-IO board developed at Bonn University. The Multi-IO board provides a USB interface with a micro controller and is connected to an adapter card dedicated to the specific flavor of the readout chip. The Multi-IO board holds a FPGA and a 2MB onboard memory as well as the RJ45 connector for the EUDET telescope trigger interface. The Multi-IO board connected to the FE-flavor specific adapter card and has a physical size of 10cm x 10cm. Additional to the boards and a PC, only one low voltage (2.5 V, 1 A) and one high voltage power supply (1000 V, 10 mA) is needed.

All digital functionality is implemented in the FPGA, and the USB micro controller controls the standard scan routines using a custom firmware. Together with histogramming and data storage in the onboard memory by the FPGA, this enables processing a large amount of standard test routines without communication to the PC, thus saving processing time.

All connections for the front-end chip are routed to the adapter card, which provides the signals via LVDS transmitters or CMOS level shifters on a 50 pin connector. Using this connector the adapter card can be connected via a flat ribbon or standard Ethernet cable to the single chip support card, wafer probing or flex adapter card. All

power and data lines are routed via the flat ribbon cable. On the FE-I4 adapter card the minimal needed LVDS signals for chip operation are additionally routed via a RJ45 connector and power lines are routed through a dedicated power connector. This enables the usage of longer cables as used in test beam environments and irradiation facilities.

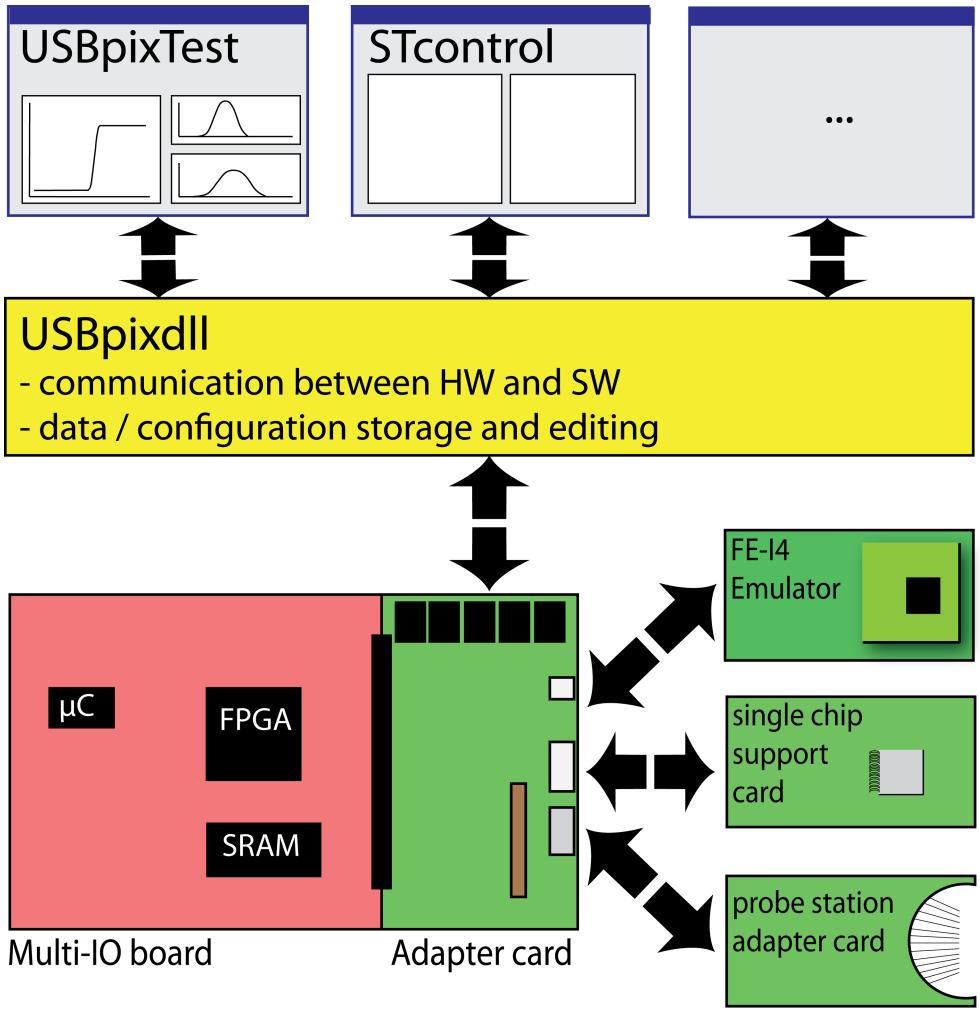

Figure 1: Schematic of USBpix test system

The software structure consists of two layers. The high level functionality like data analysis and multi dimensional scan control is implemented in the GUI layer. The two GUIs currently available communicate to the interface library USBpixdll, which takes over the USBpix hardware related communication to the Multi-IO board. The chip configuration and first data formatting is also handled in USBpixdll. This modular structure decouples high level software development from the FPGA firmware development, which enables working on the two major tasks simultaneously while updating the system to the needs of future readout chip generations.

One USBpix setup supports data acquisition of a single FE or SC module only due to resources limitation of the FPGA. A double USBpix setup consisting of standard USBpix setups with synchronized clocks has been developed to allow simultaneous data acquisition of DC modules. One of the boards acts as master board, which means this is the only board sending commands to both front-end chips of the module. The master is additionally used to get and store the data from front-end chip number one. The slave board is only used for data storage of front-end number zero. This structure is completely hidden within the hardware interface library USBpixdll and allows performing all test routines on both front-ends of a DC module in parallel.

The USBpix setup has been used for preproduction module characterization in lab and testbeam environment by several laboratories. For parallelization a dedicated "BURN-IN" adapter card, which replaces the standard USBpix FE-I4 adapter card, has been developed by university of Bonn, allowing to connect and operate up to four front-end chips in parallel and reading data from one of the four front-end chips. Operating the module in this sense means providing the clock and commands to the front-end chips. A double USBpix setup equipped with two of these adapter cards allows parallel operation of up to four double chip modules, while reading data of the modules successively. The primitive list used for test automation is able to switch between the modules of the batch, so that no further interaction is needed after having set up the system with all four modules.

The high level software supports handling of up to four module configurations connected to the setup and switching the data acquisition between these. This allows running the "dressed module tests" in batches of four modules. As the data for the modules are acquired successively the measurements described below need to be run for each module once. All modules will obtain the commands and therefore gain operation time. The primitive list used for test automation is able to switch between the modules of the batch, so the only interaction needed is connecting the modules.

The USBpix setup supports control of four 3.3 V CMOS IO pins, which will be used as control input of four relays placed on a high voltage interface card. These relays are used to open or close the high voltage connection from the power supply to the modules. This allows all combinations of high voltage selection; especially it allows running the IV-measurement successively and afterwards selecting the modules to obtain the high voltage using one single high voltage power supply.

Default scan procedures are defined in the high level USBpix software STcontrol and all parameters can be adjusted to the user's needs. Additionally STcontrol provides the functionality to define primitive lists with different types of items, which are processed successively. The item types are scans, tools and chip tests with a user selectable name, which will be called item name. All results are stored in a single ROOT file containing a tree structure. The tree has one branch for each item type. Different results with the same item type are discriminated by a user adjustable label, which will be called Scan label.

For the STAVE test another test system can be used which is able to operate up to 8 modules in parallel. This setup, called RCE system, developed by the SLAC group can perform the described tests in a similar way so that the results are compatible with the USBpix system results.

4 Tests descriptions

There is a general testing procedure which will be followed for all USBpix measurements:

Each laboratory will prepare a set of configuration files containing the information needed to run the system at each site (this file will contain the model of power supply, local directory for data storage and so on).

Measurements performed by USBpix are based upon parametric scans. These scans are standardized and their parameters are stored in configuration files. For each kind of measurement it is indicated which scan is to be selected.

Data for each modules are stored in a directory tree with the top level identified by the module S/N. For data integrity it is essential that *entering the S/N is the first operation, which is done after connecting a module to the system*.

If a module has already been tested (even in a different laboratory) a configuration files containing module information should be available and must be loaded into the software. If not, a new configuration must be created. It should contain the measured values of the capacitances used for charge injection and the measured slope of the VCAL DAC used for internal injection (used in most of the analog performances tests below). The measured values come from single chip probing and are available in electronic format from the probing sites.

Within the above mentioned directory tree all measurement data will be stored in a ROOT Tree. They will have standard names, defined combining the module serial number (5 digits for full modules) a status prefix (indicated in Table 2), followed by a 2 digit sequential number acting as measurement set identifier (in case a measurement set is repeated like in cross calibration or for receiving tests), and a test name indicated below for each test. The test name may be followed by an additional two digit version number automatically set by USBpix avoid accidental overwriting of measurements, in case, within a set of scans, some are repeated.

For data analysis it is assumed that all the files with the same `STATUS##` identifier are part of the same standard set, and the highest version for each `TESTNAME##` is the final measurement to be looked at, the others being preliminary, failed or otherwise incorrect scans.

In all the measurements requiring the HV to be applied to the detector, the module must be in a dark environment. Since the flex hybrid itself provides some screening from environmental light, even a simple cloth or cardboard additional screen may be acceptable.

For each test a description of its purpose and how it is implemented by the software is given. In most cases the operator should simply select from the USBpix menu the scan to be performed and that will be executed. If any other operation is needed it will be described.

In general, when a set of test is to be performed in determined conditions, the order in which the tests should be performed is the same as the one in which they are presented in this document. Main exception is the I-V scan, which is somehow uncorrelated to the other scans and for which the only requirement is thermal stability during the measurement and can be performed in the most comfortable stage during the test sequence. Since the biggest component of leakage current of a module before irradiation is actually a surface current, temperature for this test is also not critical.

After an initial tuning of the procedure the measurement sequence will be automatized as far as possible and the operator is required only to basically check the tests have been performed successfully and to store the data files in order to allow the subsequent analysis.

4.1 Dressed module tests

The scan definitions are stored in a primitive list as described above. This primitive list contains an item for every scan, tool or chip test. In the detailed test description below, the name of the according primitive list item as well as the label of the ROOT Tree containing the results will be provided.

4.1.1 I-V Scan

Purpose: check for sensor damages and/or HV shorts during different stages of the assembly.

Description:
The test consists in the measurement of the I-V curve. The measurement will be performed with unpowered front-end chips to avoid significant heat dissipation due to the power consumption of the chips. For the planar pixel sensor modules the measurement will be performed from 0 V to 300 V, with 3 V steps. The I-V measurement for the 3D pixel modules will be performed from 0V to 100 V at 1 V steps. The source meter is set with a current limitation of 20 µA and the measurement can stop when this value is reached.
If compared with the sensor tests performed before dicing, the modules usually have a higher leakage current. For planar sensors a leakage current in the 2 µA range at 80 V depletion voltage and an avalanche breakdown above 120 V is an acceptable result. For 3D sensors a leakage current of 2µA at 25 V depletion and an avalanche breakdown higher than 35V is an acceptable result. More information about the quality assurance of the sensors can be found in ATL-IP-QA-0030 and ATU-SYS-QC-0004.
No significant worsening of the leakage current is expected with respect to the bare module measurement. Problems may be expected if leakage current is one order of magnitude higher. Earlier breakdown or not monotonous pattern of the leakage current should trigger a rejection of the module.

Procedure:
The source meter positive output is connected to the central conductor of a coaxial cable coming from the High Voltage Switcher Card and the Flex Adapter Card and the negative output to the socket.

The following USBpix primitive list item scan must be performed with LV off:

- Scan item name: *I-V measurement*
  - Scan label: *I-V measurement*

The breakdown voltage, to be used later is conventionally defined as the voltage at which the dI/dV exceeds 5 µA/V.

4.1.2 Power Regulators

Purpose: After an assembly step or shipment, this test is meant to detect faulty chip with anomalous power consumption because of damages during the manipulation. It is usually performed in operating conditions (HV on, DAC's properly loaded, …), at the end of a standard digital test, in order to avoid excess digital current because of high noise in not-standard conditions.

Description:

On flex modules, only the current flowing into the regulator inputs is available. The test consists in measuring the current the module draws at power-up which is about 300 mA per front-end.

After that for each front-end the increase of current after configuration is registered:

1. All front-ends are configured
2. All front-ends are running a *Digital Test*, the current consumption is measured after startup of the *Digital Test*.
3. Read the analog regulator output voltage and current using the on chip generic ADC:
    - Tool item name: *Read GADC VDDA*
        - Chip test label: not user adjustable
    - Tool item name: *Read GADC IDDA*
        - Chip test label: not user adjustable

Procedure:

The above sequence of measurements is performed within the procedure of the *Digital Test* described below.

4.1.3 Digital Test

Purpose: to detect failures in the global and pixel registers that may affect the proper configuration of the module. Test the readout chain and detect defective channels.

Description:

This test consists of four parts:

1. write and read back of the front-end configuration registers, these are essential to the chip operation;
2. write and read back of the pixel register, i.e. the shift register used to configure every single pixel (providing local threshold tuning, masking…)
3. for each pixel, 200 pulses are injected at low frequency to the output of the discriminator, simulating the discriminator signal when a preamplifier pulse trigger the discriminator. This part of the test checks the readout chain from the pixel cell down to the data LVDS transmitter of the chip, but the first (out of five) ToT buffer for each pixel as well as the first (out of five) LVL1 counter per four pixel digital region. As the digital injection line is ORed to the discriminator output, this test is also sensitive to analog stuck high pixels.
4. for each pixel, 5 short pulses are injected at high frequency to the output of the discriminator, simulating the discriminator signal when a preamplifier pulse trigger the discriminator. This test checks the functionality of the five ToT buffers for each pixel as well as the proper functioning of the five LVL1 counters of each four pixel digital region.

Failure at point 1 means the chip cannot be used. Failures at point 2 are usually circumvented by isolating the column pair in which they happen, resulting in 720 unusable pixels. At point 3 and 4, failures in the readout chain in the chip periphery as well as single four pixel digital region and single pixels failures will be detected and the number defective channels is counted.

Procedure:

Switch on the low voltages.
HV can be either on or off for this test.
Execute a primitive list item of type *chip test* with the sub configuration *GLOBALREG*. Several bit patterns are predefined to check all bits of the global registers. All of them should be tested.
Execute a primitive list item of type *chip test* with the sub configuration *PIXELREG* and setting "Latch to be tested" set to *ALL*. Several bit patterns are predefined to check all bits of the pixel registers. All of them should be tested.
Execute the following items of the USBpix primitive list items:

- Chip test item: *GR test A* to *GR test I*
  - Chip test label: not user adjustable
- Chip test item: *PR test A* to *PR test F*
  - Chip test label: not user adjustable
- Scan item name: *Digital Test*
  - Scan label: *Digital Test*
- Scan item name: *Buffer Test*
  - Scan label: *Buffer Test*

4.1.4 Threshold Tuning

Purpose: Set a uniform threshold along the module close to the target threshold of 3000 electrons.

Description:
Before the pixel threshold tuning is started, the threshold needs to be globally adjusted for each front-end to the target threshold. The global threshold setting in FE-I4 based modules is adjusted by a combination of the two DACs VthinAltCoarse and VthinAltFine, which will be simplified called GDAC. Two adjustment algorithms is implemented in USBpix to adjust the GDAC automatically. A fast algorithm injecting the target charge and using a binary search adjusting GADC to a average occupancy of 50% is implemented. Additionally a slow algorithm performing full threshold scans for several GDAC settings and extrapolating to the GDAC value close to the target threshold. The slow GDAC tuning algorithm is more robust to single pixels with high noise hit occupancies.
The pixel tuning can be performed in two similar ways.
For FE-I4B this test can be performed in the so called "fast-tune" mode, where a fixed charge, corresponding to the desired threshold is injected several times into each pixel and the occupancy is measured for each pixel. The tuning algorithm then searches binary for the TDAC value closest to an occupancy of 50 %. This tuning procedure is extremely fast, but less robust in case of high noise hit occupancies.
The second method consists in the measurement of the threshold of all pixels, as described below to for the *Threshold Scan*. Again a binary search algorithm searches for each pixel for the TDAC setting closest to the target threshold.
Both algorithms can be adjusted to start with the current TDAC map of the module, so a "retune" of a pre-tuned module is possible.
Both algorithms show similar results, if the noise occupancy of the module is not too high.

Procedure:
The above data collection and is performed by a single USBpix primitive list items, with all power supplies on:

- Scan item name: *GDAC fastTune*
  - Scan label: G*DAC fast Tune*
- Scan item name: *TDAC fastTune 1/2*
  - Scan label: *TDAC fast Tune 1/2*

The results of TDAC determination are directly put into the module configuration file. The threshold tuning is affected by the ToT Tuning described below. So the TDACs will be repeated to re-tune the thresholds after the performing the ToT Tuning.

4.1.5 ToT Tuning

Purpose: Tune the ToT response to a m.i.p. of each pixel in order to have a uniform response to the collected charge in a time acceptable for operation in ATLAS. Calibrate the relationship between the measured ToT and the collected charge and afterwards perform a short verification measurement.

Description:
The pixel detector has an indirect pulse height information using the Time over Threshold (ToT) technique: the pulse shape is approximately triangular and the time by which the preamplifier output stays over the threshold is approximately proportional to the pulse height.
The slope of the return to baseline of the triangular pulse is determined by the feedback current of the amplifier, which can be tuned at the chip level changing the PrmpVbpf-DAC register and at the pixel level using the 4-bit FDAC pixel register.
The ToT tuning consists of three parts.
At first, the ToT response of all pixels to the charge deposited by a minimum ionizing particle (m.i.p.) is made uniform by proper setting of the IF and FDACs.
This is done by injecting a fixed charge of 16,000 e, corresponding to the approx. most probable energy loss in the 200 µm thick silicon sensor in planar silicon sensor case or 230 µm sensor in 3D silicon sensor case, and choosing the above mentioned DACs in order to have an average ToT response of 10 clock cycles. The number of clock cycles has been chosen taking into account that the preamplifier must return to the baseline within a short time to avoid pile-up due to the high hit rate in IBL. The subsequent step is to inject different charges in, compute for each ToT the average PulserDAC value resulting in this specific ToT, and build a look up table containing the charge injected by the average PulserDAC resulting in each ToT.
These calibrations will be used to translate ToT to charges when collecting data with real particles.
After tuning and calibrating the ToT, the result should be verified by injecting the target charge of the tuning 200 times and recording the ToT response for each pixel.
Since changing the feedback current also slightly affects the threshold, after IF and FDAC tuning the threshold tuning needs to be re-done.

Procedure:
Following items in the USBpix primitive list will perform the ToT Tuning with the specifications described above:

- Scan item name: *IF Tune*
  - Scan label: *IF Tune*
- Scan item name: *FDAC Tune*
  - Scan label: *FDAC Tune*

The results of IF Tune and FDAC Tune are automatically loaded to the module configuration.
Now perform again a *Threshold Tuning* starting from the pre-tuned TDAC map.
Finally perform the ToT calibration:

- Scan item name: *ToT Calibration*
  - Scan label: *ToT Calibration*
- Scan item name: *ToT Verification*
  - Scan label: *ToT Verification*

4.1.6 Threshold Scan

Purpose: This test performs a measurement of the threshold and noise of each pixel and is the central part of most of the calibration task. It can be performed in different conditions for calibration, detection of faulty cells and check of bumping defect.

Description:
A voltage pulse V is injected on the calibration capacitance C_{inj} of each pixel. That will generate a signal at the input of the preamplifier equivalent to the one generated by a charge $V \times C_{inj}$.

A set of 200 pulses is generated for different value of the injected charge (from 0 to ~10000 e, in ~50 e steps).
The number of collected hits for each injected charge is recorded and at the end of the scan an S curve is fitted. The 50% efficiency on the S-curve defines the threshold value. The steepness of the transition from no detected hits to full efficiency is inversely proportional to the noise, which can be so calculated.
The injected pulse comes from an internal chopper connected to the output of the PulserDAC (internal injection).
The internal injection circuitry as well as the injection capacitance will have chip to chip variations due to the fabrication process. Both (injection capacitance as well as the PulserDAC calibration) have been measured during the on wafer IC tests. The results of these measurements will be loaded from a data base for each module when the first module configuration is generated.
This test will also be repeated at different V_{in} values for the determination of the analog operational margin.
This test allows a determination of the disconnected bump if the results of the noise figure for each pixel between a scan with applied HV and a scan without applied HV are compared. If there is no difference in noise between HV off and HV on the bump of this pixel is not properly connected.

Procedure:
Low voltage power supply switched on.
Check HV supply status corresponds to the scan which must be performed. Usually the operating high voltage is used. HV off scans may be done with the HV off or with the HV output on, but at the -1 V setting to check the increase of the noise due to under-depleted sensor.
A threshold scan with HV on/off is performed by the USBpix primitive list items:

- Scan item name: *Threshold Scan HV on/off*
  - Scan label: *Threshold Scan HV on/off*

4.1.7 Crosstalk

Purpose: Measure the cross-talk fraction and detect bump defects resulting in increased capacitive coupling between pixels (too large bumps, small separation between sensor and FE electronics, pouring of glue in the sensor-FE interstitial region)

Description:
Because of capacitive or resistive coupling of the injected pixel with its neighbors, part of its charge can leak and be collected on the nearby channel.
This test is similar to an analog scan, but in this case the charge is injected in two over next pixels on the long pixel size, while only the pixel in between is enabled for readout.
Usually it is not possible to inject large enough charges to see cross-talk hits using the on chip injection circuitry on depleted FE-I4 based modules, due to the limitation of the a maximum inject able charge of 55 000 electrons. So measurable cross-talk indicates some excess coupling which can be caused by shorted or almost shorted bumps. In the past also penetration of glue in the interstitial region between the front-end chip and the sensor has caused an increase in the capacitive coupling.
USBpix plots for each channel the number of hits collected when injecting to its neighbors, producing crosstalk occupancy map. The sensitivity of this crosstalk test is therefore in the order of three percent assuming a threshold of 1500 electrons.
In the n-in-n pixel sensor design cross-talk hits are expected to be seen in under depleted operation due to the missing inter pixel isolation in this conditions. Therefore a not existing cross-talk in under depleted operation indicates unconnected bump bonds, so the cross-talk test is repeated in these conditions.

Procedure:
LV power is on, scan will be performed in HV on/off mode.
Perform standard USBpix primitive list item:

- Scan item name: *X-Talk Test HV on/off*
  - Scan label: *X-Talk Test HV on/off*

4.1.8 Monleak

Purpose: Measure of each pixel's leakage current to look for excess current due to localized earlier sensor breakdown.

Description:
In the *Monleak* scan the leakage current which is collected by each pixel is measured using the on chip generic ADC.
This kind of scan will be extremely useful during operation in ATLAS, as it allows to directly measure disuniformity of irradiation and to separate sensor leakage passing through the pixel from any possible surface path.
Before irradiation the leakage current is quite small, and the *Monleak* scan is almost insensible to the real leakage current. So any pixel showing a significant value is either defective or near a local early breakdown region of the sensor.

Procedure:
The module must be powered up. The depletion voltage on the power supply must be set to 300 V or 30 V resp. with current limitation at 20 μA. In case of early breakdown, the current limitation will implicitly take care to apply a lower voltage, without operator intervention.
A USBpix primitive list item is to be performed:

- Scan item name: *Monleak Scan*
  - Scan label: *Monleak Scan*

4.1.9 Timewalk

Purpose: Measure the relationship between the injected charge and the preamplifier response time.

Description:
The time between the shooting of the discriminator and the pulse injection depends on the pulse height, with high pulses firing almost immediately and pulses near threshold showing a long delay.
If the delay is more than about 20 ns, the pulse would not be assigned the proper beam crossing at the LHC. That results in an effective threshold which is higher than the one defined by the threshold scan.
This scan uses a delay circuitry in the front-end that sets a delay between the calibration pulse command and the signal which drives the charge .
By injecting a known charge and checking for which delay it starts to be associated to a wrong beam crossing it is possible to measure the firing time (as 25 ns minus the delay value) of the pulse. Repeating the process for several charges allows to reconstruct the full charge/delay relationship.
Rem.: injection timing must be set correctly, which needs the T0 scan to be executed beforehand.

Procedure:
Perform standard items:

- Scan item *Timewalk Scan*
  - Scan label: *Timewalk Scan*

4.1.10 In-time Threshold Scan

Purpose: measure the effective threshold that will be observed in ATLAS, i.e. the minimal value of charge that, because of timewalk, will have a delay below 20 ns.

Description:
This test is another way to view at the timewalk issue, by trying to get directly the in-time threshold.
A threshold scan is performed, but accepting only one beam crossing (during module characterization, usually 16 beam crossing are collected to be independent of exact timing).

In this case the S-curve of the threshold profile is cut down when the pulse height is still above threshold, but the hits is generated too late and is associated to the next beam crossing.
This new threshold value is the effective in-time threshold that the detector will observe in ATLAS and should be about 1500 e higher than the real threshold.

Procedure:
Perform a USBpix T0 scan with 55 ke analog injection

- Scan item name: *T0 Scan*
  - Scan label *T0 Scan*

The output of this scan is a injection delay which corresponds to the firing time of the 55 ke pulse.
Set the delay circuitry to this value + 5 ns.
Perform the modified *Threshold Scan*:

- Scan item name: *In Time Threshold Scan*
  - Scan label: *In Time Threshold Scan*

4.1.11 Operational margin

This test consists of repeating some of the previously mentioned scans in different conditions and requesting service records from the front-ends, which contain information of errors occurring in the front-ends.

V_{in} operational margin
It will be a scan of V_{in} from 1.8 V to 2.4 V in 0.05 V steps. For each step a *Digital Test* is performed according to section 4.1.3. Additionally service records are requested from the front-ends and the generic on chip ADC is used to measure the analog regulator output and current.

- Chip Test item: *GR Test Vin 1.80, 1.85, …*
- Chip Test item: *PR Test Vin 1.80, 1.85, …*
- Tool item: *Read GADC VDDA 1.80, 1.85, …*
- Tool item: *Read GADC IDDA 1.80, 1.85, …*

The first scan will be used by the module analysis to compute the official operational margin of the digital voltage.

Temperature operational margin
A series of measurements will be performed during burn-in time to check the power regulators and chip performance at lowest (-40°C) and highest (+30°C) possible temperatures during ATLAS operation. These are in particular regulator power up test, regulator stress tests (setting the front-end to high current state) and possibility to save power consumption after startup by lowering Vin from 2.0 V to 1.8 V after the regulators powered up. Additional the general module performance tests like electronic noise and noise hit occupancy measurements will be performed..

4.1.12 Noise Suppression

Purpose: Select pixels not usable for source measurements.

Description:
Taking data with a radioactive source requires triggering on the *hitbus* of the front-end. This is a fast OR of the discriminator outputs.
These source measurements can be saturated or paralyzed by noisy pixels or by pixels which keep the hitbus permanently stuck.
In the I4 version of the front-end the discriminator output and the digital injection line are connected using a logical OR, with its output directly entering the four pixel digital region. A pixel with stuck high analog

part or very high noise hit rate is therefore directly noticeable in the *Digital Scan* result. Such pixels may probably work well in ATLAS when an external trigger is used but they are just an annoyance in the source measurement below.
Purpose of this test is to identify pixels with a too high noise rate or keeping the hitbus stuck in order to mask them during source scans.

Procedure:
Mask all pixel which have recorded zero occupancy in the *Digital Scan* previously performed.

Mask all pixel which have shown a noise hit probability above 10^{-5} per bunch crossing (25 ns) in the *Noise Occupancy Scan* previously performed.

4.1.13 Noise Occupancy Scan

Purpose: measure the noise hit probability per bunch crossing (NOcc) for each pixel. Pixels with high noise hit rate will decrease the tracking performance and should therefore be masked in operation.

Description:
The NOcc is measured sending random triggers to the module and recording the occupancy of every pixel. The result is an upper limit of the noise hit probability for every pixel using the definition

NOcc = occupancy / sensitive time [bunch crossings]

With occupancy set to one, if a pixel did not recognize any hit. The sensitive time in bunch crossing is given by the number of triggers send to the module times the trigger multiplication mode set in the front-end. The USBpix system measures the sensitive time in bunch crossings directly by counting the number of received data headers.
The noise occupancy will be used to determine the minimal operation threshold. The threshold of the module is lowered stepwise and at each step a NOcc scan is performed. The number of masked, i.e. noisy pixel and the total noise hit rate will rise with lower thresholds. The minimal operational threshold is defined as the threshold, at which a predefined noise hit rate or number of masked pixel is reached.

Procedure:
The following USBpix primitive list item performs a noise occupancy scan:

- Scan item name: *NOcc Scan*
  - Scan label *NOcc Scan*

The output of this scan is an occupancy map, which needs to be further analyzed as described above.

4.1.14 Source Scan

Purpose: Identify pixels not answering to ionization because disconnected, merged, defective or badly tuned. Provide data for eventual recalibration of the charge calibration later, if needed.

Description:
The whole module should be exposed to the source until the number of events (hits) exceeds the target event number of 5 million hits in double chip and 2.5 million hits in single chip case. The number of events should be high enough to eventually allow using this data later for recalibration of the charge calibration.
The module exposed the radiation of a ^{241}Am X-ray source from the flex side, a decreased hit rate will be recognized below the passive components.
The FE internal self trigger mechanism must be used.

Procedure:

As the source needs to be placed on the module by the user, the *Source Scan* is not defined as a primitive list item. The measurement will be started from the Scan Panel using the predefined Scan FE_ST_SOURCE_SCAN. The correct result .ROOT file needs to be selected and the Scan label should be set to:

- Scan result file: *Same as used in FLEX test primitive list*
  - Scan label: *Source Scan*

4.1.15 Chip Serial Number Check

Purpose: Check the front-end serial number that was burned into the on chip EPROM during wafer probing and compare against data base.

Description:
The FE-I4B holds an EPROM with the chip serial number burned in at wafer probing test stage. This unique chip identifier can be read from the EPROM to the front-ends global configuration registers and then read by usual reading of the global registers.

Procedure:
A USBpix primitive list tool items and a chip test item are needed to perform the two step action:

- Tool item name: *Read Chip SN from EPROM*
- Chip test name: *Read Chip SN from GR*
  - Chip test label: not user adjustable

4.1.16 Voltage regulator and ADC calibration

Purpose: Calibrate the voltage regulator and ADC circuit of the FE-I4

Description:
The output voltage is of the regulators in FE-I4 are adjustable and the output characteristic must be calibrated. The on chip ADC must as well be calibrated. The calibration data needs to be stored in a database.

Procedure:
An USBpix primitive list tool items and a chip test item are needed to perform the two step action to calibrate the LDO output voltage:

- Scan item name*:* *LV Measurement FE0 / FE1 VDDA / VDDD*
  - Scan label: *LV Measurement F01 / FE1 VDDA / VDDD*
- Tool item name: *Set best FE0 / FE1/ VDDA / VDDD*

The output voltage of the regulator for VDDA is scanned again and the output voltage is measured using the internal generic ADC:

- Scan item name*:* *GADC Measurement FE0 / FE1 VDDA*
- Scan label: *GADC Measurement FE0 / FE1 VDDA*

This data will be used to calibrate the internal generic ADC offline

4.1.17 Service Record Tests

Purpose: Test of error messages and LV1 and BCID counters circuits of the FE-I4

Description:
Together with standard hit data the FE-I4 sends on request so called service records, which are the output of chip error counters. Additionally the most significant bits of the LV1 and BCID counter are send out as special data field in the output data stream. To ensure the functionality of the counters the error messages and counters are read for several triggers during a scan, e.g a source scan, see section 4.1.14, and checked whether the counters are incremented correctly.

Procedure:
A USBpix primitive list tool items and a chip test item are needed to perform the two step action:

- Tool item name: *Read error counters*
- Tool item name: *Read BCID/LV1 MSB*
- Chip test name: *Read error counters*
- Chip test name: *Read BCID/LV1*
- Scan result file*:* *REG_Error_Counter, REG_BCID, REG_LV1*
    - Scan label: REG_error, REG_counter

4.1.18 Timewalk Correction Test

Purpose: Test of the timewalk correction function of the FE-14

Description:
To ensure that small hits, i.e. hits with a ToT smaller than an adjustable threshold are assigned to the correct bunch crossing a logic in the FE-I4 is implemented. These logic copies small ToT hits in the direct neighborhood of a big ToT hit to the last and/or last but one bunch crossing. To check the correct function of this logic a big hit and small hit just above the discriminator threshold are injected simultaneously into 2 neighboring pixel per column and it is checked whether both hits are assigned to the same bunch crossing ID.

Procedure:
The following USBpix primitive list item performs an Timewalk Correction scan:

- Scan item name: *Timewalk Correction Scan*
    - Scan label *Timewalk Correction Scan*

5 Testing time and automation

The full electrical characterization of a module is likely to be one of the most time expensive efforts in the production of the ATLAS IBL detector. First experience with FE-I4A modules show that the time needed for the characterization is about 1 working day.

Automation of the procedures will allow a significant gain in time. After a significant statistics on final electronics and modules is gained, it will be possible to reduce the test sequence only to tests, which are shown to be effective in spotting out problematic modules. The goal is to reduce the full electrical characterization of a module to one 8h shift, during which the module can be left unattended for a significant fraction of the time.

Reduced test sets, like *initial loading* or *module receiving tests* should not take more than 15 minutes per module.

In table 3 are indicated the measured testing times.

Table 3: Observed test time

Status	**Time** (excluding source scan) [h]	**Source scan** [h]
ASSY	1.5	0.5
BURN	0.7	0.5
FLEX	3.0	0.5
RECV	0.5	-

Ideally the whole test procedure should be almost completely automated and performed in one run. Anyhow since the full characterization of a module may spread among several operator shifts, it may be useful for each module to have a traveller which consists of a check list whether a certain test has been done or not.

Automation is mainly achieved by the usage of USBpix primitive lists (macros). These primitives are under active development and one example can be found in the Appendix.

A further automatization is achieved by the usage of the burn-in setup of the USBpix, see figure . This setup allows a connection of up to 4 modules to one single readout board. So the USBpix setup can loop over all four modules successively without any human intervention. So with one setup for each test stage per testing site even for the FLEX test up to four modules can be tested within one day. This sets the maximum throughput per testing site to about 20 modules per week.

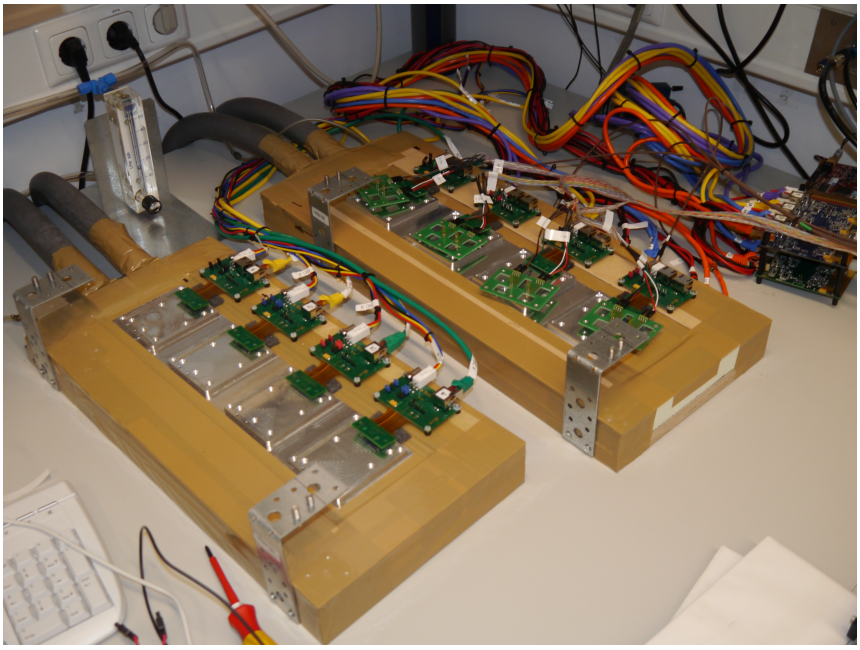

Figure 2: Picture of the multi module USBPix setup. Up to four modules could be connected to one test system and can be tested consecutively.

6 Data analysis, module selection and data recording

When a test set is completed, all data test must be imported in the *Module Analysis Framework* and an analysis performed to extract the significant information that must be uploaded into the database. The analysis applies a set of pre-defined cuts to the individual tests and eventually assigns a label to the module (green = Pass or usable, yellow = Pass but only 2nd choice for use, red = Fail or not usable, blue = error during data processing). If a blue label is assigned a manual reprocessing of the data or re-testing of the module is necessary. Figure 3 shows the summary of number of modules loaded and analysed by the analysis framework. For each module the result of the individual test is shown in a box and highlighted in green, red, yellow or blue depending on cut result. Only modules passed all individual cuts are selected.

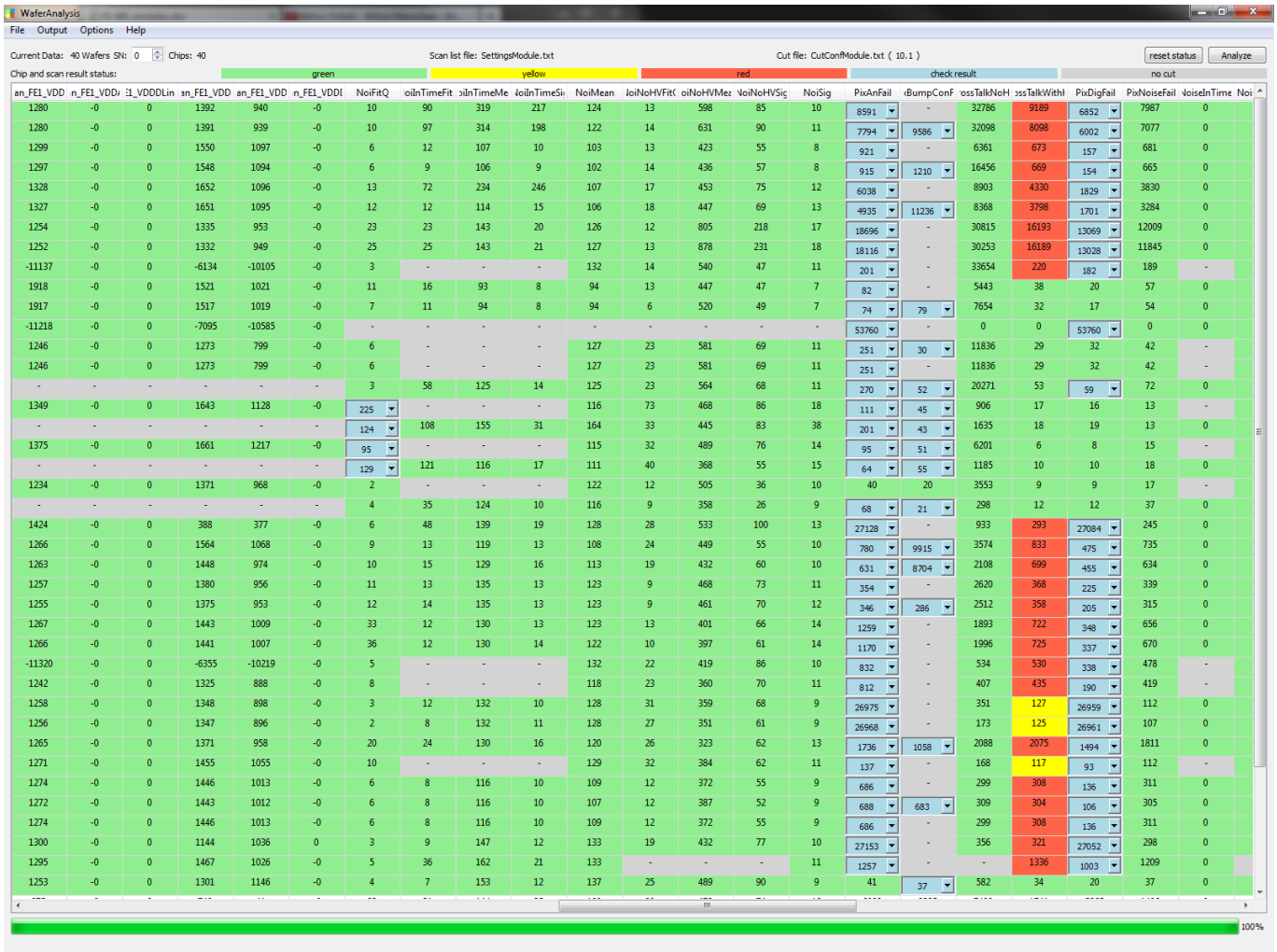

Figure 3: Screenshot from the Module Analysis Framework. It shows one module per row and the measured quantities as columns. Each field is green, yellow or red depending on the cuts. Blue fields demand the operator to decide or re-test the module.

Modules that fail the ASSY or BURN test stage can be reworked if possible. After such rework the testing sequence starts again with the ASSY test. Only modules, which passed both test stages will go to final test stage FLEX. For the module selection process mainly this FLEX test stage is relevant. Results from previous test stages or other tests like visual inspection, wire bond pull tests etc. can be used in the final module selection. For instance a rework of the module can lead to devaluation of the module from green to yellow. The framework creates an XML file consisting of a number of performance values. These parameters will be stored directly in the production database together with a set of meaningful performance plots for a quicker evaluation afterwards.

The data themselves will be handled as ROOT Trees and stored in ROOT DB files. When uploading test data to the production database (PDB) the corresponding ROOT DB file will be uploaded. Either for each status test (ex.: *FLEX/01, BURN/01, RECV/03…*) an individual file is uploaded or all status test files of one module are merged to one single ROOT DB file. The uploading will be done directly from the Module Analysis Framework once the operator has checked the analysis and assigned the final module label of the test data. Other information like the module summary file is stored as well.

7 Module selection criteria

Main criterion for the selection of modules is the number of working channels per module according to the specifications. For most of the tests described in this document one important result is the number of pixel failing this test, e.g non responding pixel in source scan, not tunable pixel, digitally dead pixel, disconnected pixel etc. In table 3 the cut criteria for these pixel cuts are summarized. Finally all bad pixel can be summed up to a final number of defective pixel. A module is accepted for IBL if it shows less than 1% defective pixel, i.e 270 defective pixel for a 3D SC module or 540 for a planar DC module.

Table 3: Summary of the pixel level cuts for module selection

|Threshold Pixel - Threshold Distribution Mean|	> 5 Sigma
|Noise Pixel - Noise Distribution Mean|	> 5 Sigma
|No HV Threshold Pixel - No HV Threshold Distribution Mean|	> 5 Sigma
|No HV Noise Pixel - No HV Noise Distribution Mean|	> 5 Sigma
Bump Connectivity (No HV Noise - Noise)	< 20 e
Analog Test	Occ != 200
Digital Test	Occ != 200
Crosstalk	Occ != 0
Shorted Pixel	Analog Hits < 5 AND Crosstalk Hits != 0
Noise Occupancy	Occ > $1x10^{-7}$
Source Scan Occupancy	Occ < 0.1 x mean Occ OR Occ > 4.5 x mean Occ
Noisy Pixels after 1500e^- NOcc scan	ENABLE != 1

Apart from this number of defective pixel many other criteria are applied which can discard a module. These criteria are power consumption at startup and after configuration, range of VDDA setting, perfect global GR and PR operation, LVDS data transmission with long cables, sensor leakage current at operational voltage, operational margin in terms of V_{in} and temperature. With the experience of the first 50 modules we will define the basic set of cut values. There are two different types of cuts: cuts marking a pixel as defective and global chip cuts. The later cuts can also be feature a third category between pass (green) and fail (red), namely blue. These are marking chips

for which some tests give unexpected results and should be either re-done or need some more checks. Table 4 is listing these module level cuts.

Table 4: Module level cuts for IBL module selection.

	Perfect	Green	Red	Blue
Threshold Distribution Mean after tuning to 3000e$^-$	> 2950 AND < 3050	<2950 OR > 3050	<2500 OR > 3500	< 2000 OR > 9000
Threshold Distribution Sigma	< 70	>70	>100	> 200
Noise Distribution Mean	> 90 AND < 160	< 90 OR > 160	< 85 OR > 180	< 80 OR > 200
No HV Threshold Distribution Mean	> 2500 AND < 3600	<2500 OR > 3600	<2000 OR > 5000	< 2000 OR > 9000
Source Scan empty BCID bin				!= 0
Source Scan empty LV1ID bin				!= 0
Hit Discr. Failing Columns				!= 0
Globar Register Test				fail
Pixels that fail a register test				> 54
Pixels Failing any Test	< 100	> 100	> 270	>500
Sensor breakdown voltage DC [V]	> 150	< 150	< 130	< 50
Sensor breakdown voltage SC [V]	> 50	< 50	< 30	< 20
Sensor dI/dV at operation point [A/V]	< 0.2	> 0.2	> 0.5	> 0.9
LV VDDA Maximum [mV]	> 1400	< 1400	< 1350	> 1550
LV VDDA Minimum [mV]			> 1400	< 1200
LV VDDD Maximum [mV]	> 1350	< 1350	< 1250	> 1600
LV VDDD Minimum [mV]			> 1150	< 850

Later on if time allows we can as well use the number of defective pixel per module for a ranking procedure of the modules. Tests measuring a global module property like high leakage current or smaller operational margin can be translated in a penalty in terms of defective pixel. In addition other criteria like a rework (chip rework, wire bond rework, module flex rework) or the usage of bad quality parts can also be translated into a number of additional bad pixel, see table 5 for a summary of the module penalties. This method allows grouping of the best modules in order to put them in areas of the IBL where the best modules are required. In this sense we define two categories of modules: Module with a number of bad pixel less than 270 per FE chip, i.e. 270 for 3D SC modules and 540 for planar DC modules resp., are of good quality or green. And modules showing more than 270 bad pixel are discarded from further integration or red. In addition modules showing less 100 bad pixel per FE-chip are of best quality.

Table 5: Penalties in numbers of bad pixel for IBL modules

Mechanical issues	
missing glue in wing area	100
re-bond of wire bonds (FE, bridge, HV)	10 times # of re-bonds
excess of glue on sensor edges	50
chip re-work	100
damage on module edges	50
Sensor Properties	
breakdown voltage change during assembly (yellow cut)	100
breakdown voltage change (red cut)	1000
non visible alignment marks	50
non visible alignment marks (> 2)	1000
Electrical Performance	
mean noise out of range (yellow cut)	50
width of tuned threshold distribution (yellow cut)	50
regulator output voltages out of range (yellow cut)	50
mean noise, threshold disp., regulator output (red cut)	1000
HV capacitor potting removed/thinned	30
Re-work of the reset capacitors	100
minimal operational threshold > 1,500e^-	50
Flux remanents visible in HV off scan	100

References

ATU-SYS-EP-0007 **Module Flex design FE-I4B.** *https://edms.cern.ch/document/1221314/2*

ATL-IP-CS-0029 **Technical Specification and Acceptance Criteria for the Bump Bonding of the IBL pixel modules.** *https://edms.cern.ch/document/1166857/1.2*

ATL-SYS-AN-0001 **ATLAS IBL Pixel Module Assembly.** *https://edms.cern.ch/document/1221780/1*

USBpix setup ***Development of a versatile and modular test system for ATLAS hybrid pixel detectors, NIM A 650 (2011) 37.*** *doi:10.1016/j.nima.2010.12.087*

ATL-IP-QA-0030 **R. Klingenberg, D. Muenstermann and T. Wittig, Sensor Specifications and Acceptance Criteria for Planar Pixel Sensors of the IBL at ATLAS.** *https://edms.cern.ch/document/1212891/*

ATU-SYS-QC-0004 **C. Da Via, M. Boscardin, G. Pellegrini, G-F. Dalla Betta, Technical Specifications and Acceptance Criteria for the 3D Sensors of the ATLAS IBL.** *https://edms.cern.ch/document/1162203/1*

8 Appendix

8.1 Example of an ASSY and a FLEX primlist

Label	Type	Index
select module	tool	00000
set threshold high	tool	00001
set PrmpVbp to 0	tool	00002
set PrmpVbpLeft to 0	tool	00003
set PrmpVbpRight to 0	tool	00004
set FDACVbn	tool	00005
switch on Vbias	tool	00006
read Ibias	tool	00007
IV_SCAN	custom PixScan	00008
power up LV	tool	00009
send soft reset	tool	00010
send BCR	tool	00011
send ECR	tool	00012
read Iin startup	tool	00013
configure module	tool	00014
LV_MEAS_FE0_VDDA	custom PixScan	00015
LV_MEAS_FE0_VDDD	custom PixScan	00016
LV_MEAS_FE1_VDDA	custom PixScan	00017
LV_MEAS_FE1_VDDD	custom PixScan	00018
set best VDDD FE0	tool	00019
set best VDDA FE0	tool	00020
set best VDDD FE1	tool	00021
set best VDDA FE1	tool	00022
RX_DELAY_SCAN 1	custom PixScan	00023
GADC VDDA Scan FE0	custom PixScan	00024
GADC PulserDAC scan FE0	custom PixScan	00025
GADC VDDA Scan FE1	custom PixScan	00026
GADC PulserDAC scan FE1	custom PixScan	00027
set PrmpVbp to 43	tool	00028
set PrmpVbpLeft to 43	tool	00029
set PrmpVbpRight to 43	tool	00030
send soft reset	tool	00031
send BCR	tool	00032
send ECR	tool	00033
reconfigure module	tool	00034
read Iin configured	tool	00035
RX_DELAY_SCAN 2	custom PixScan	00036
GR_TEST	chip test	00037
PR Test	chip test	00038
read EPROM to GR	tool	00039
read SN from GR	chip test	00040
read GADC AGND	tool	00041
read GADC AGND from GR	chip test	00042
read GADC IDDA	tool	00043
read GADC IDDA from GR	chip test	00044
read GADC Vref	tool	00045
read GADC Vref from GR	chip test	00046
read GADC Ileak	tool	00047
read GADC Ileak from GR	chip test	00048
read GADC Temperature	tool	00049
read GADC Temperature from GR	chip test	00050
DIGITAL_TEST High Threshold	custom PixScan	00051
ramp up HV	tool	00052
GDAC_FAST_TUNE 1	custom PixScan	00053
IF_TUNE 1	custom PixScan	00054
GDAC_FAST_TUNE 2	custom PixScan	00055
TDAC_FAST_TUNE 1	custom PixScan	00056
FDAC_TUNE	custom PixScan	00057
TOT_VERIF 1	custom PixScan	00058
TDAC_FAST_TUNE 2	custom PixScan	00059
DIGITAL_TEST 2	custom PixScan	00060
THRESHOLD_SCAN with HV	custom PixScan	00061
ANALOG_TEST	custom PixScan	00062
TOT_VERIF 2	custom PixScan	00063
CROSSTALK_CHECK with HV	custom PixScan	00064
save cfg.	tool	00065
set HitDiscCnfg 0	tool	00066
NOISE_OCC	custom PixScan	00067
create noisy pixel mask	tool	00068
create stuck pixel mask	tool	00069
save cfg. source	tool	00070
set HitDiscCnfg 2	tool	00071
ramp down Vbias	tool	00072
THRESHOLD_SCAN no HV	custom PixScan	00073
CROSSTALK_CHECK no HV	custom PixScan	00074
power off	tool	00075

Label	Type	Index
select module	tool	00000
set threshold high	tool	00001
set PrmpVbp to 0	tool	00002
set PrmpVbpLeft to 0	tool	00003
set PrmpVbpRight to 0	tool	00004
set FDACVbn	tool	00005
switch on Vbias	tool	00006
IV_SCAN	custom PixScan	00007
power up LV	tool	00008
send soft reset	tool	00009
send BCR	tool	00010
send ECR	tool	00011
read Iin startup	tool	00012
configure module	tool	00013
LV_MEAS_FE0_VDDA	custom PixScan	00014
LV_MEAS_FE0_VDDD	custom PixScan	00015
LV_MEAS_FE1_VDDA	custom PixScan	00016
LV_MEAS_FE1_VDDD	custom PixScan	00017
set best VDDD FE0	tool	00018
set best VDDA FE0	tool	00019
set best VDDD FE1	tool	00020
set best VDDA FE1	tool	00021
RX_DELAY_SCAN 1	custom PixScan	00022
GADC VDDA Scan FE0	custom PixScan	00023
GADC PulserDAC scan FE0	custom PixScan	00024
GADC VDDA Scan FE1	custom PixScan	00025
GADC PulserDAC scan FE1	custom PixScan	00026
set PrmpVbp to 43	tool	00027
set PrmpVbpLeft to 43	tool	00028
set PrmpVbpRight to 43	tool	00029
send soft reset	tool	00030
send BCR	tool	00031
send ECR	tool	00032
reconfigure module	tool	00033
read Iin configured	tool	00034
RX_DELAY_SCAN 2	custom PixScan	00035
GR_TEST	chip test	00036
PR Test	chip test	00037
read EPROM to GR	tool	00038
read SN from GR	chip test	00039
read GADC AGND	tool	00040
read GADC AGND from GR	chip test	00041
read GADC IDDA	tool	00042
read GADC IDDA from GR	chip test	00043
read GADC Vref	tool	00044
read GADC Vref from GR	chip test	00045
read GADC Ileak	tool	00046
read GADC Ileak from GR	chip test	00047
read GADC Temperature	tool	00048
read GADC Temperature from GR	chip test	00049
DIGITAL_TEST High Threshold	custom PixScan	00050
ramp up HV	tool	00051
GDAC_FAST_TUNE 1	custom PixScan	00052
IF_TUNE 1	custom PixScan	00053
GDAC_FAST_TUNE 2	custom PixScan	00054
TDAC_FAST_TUNE 1	custom PixScan	00055
FDAC_TUNE	custom PixScan	00056
TOT_VERIF 1	custom PixScan	00057
TDAC_FAST_TUNE 2	custom PixScan	00058
DIGITAL_TEST 2	custom PixScan	00059
THRESHOLD_SCAN with HV	custom PixScan	00060
ANALOG_TEST	custom PixScan	00061
TOT_VERIF 2	custom PixScan	00062
CROSSTALK_CHECK with HV	custom PixScan	00063
T0_SCAN	custom PixScan	00064
INTIME_THRESH_SCAN	custom PixScan	00065
TOT_CALIB_LUT	custom PixScan	00066
save cfg.	tool	00067
set HitDiscCnfg 0	tool	00068
NOISE_OCC	custom PixScan	00069
create noisy pixel mask	tool	00070
generate stuck pixel mask	tool	00071
save cfg. source	tool	00072
set HitDiscCnfg 2	tool	00073
ramp down Vbias	tool	00074
THRESHOLD_SCAN no HV	custom PixScan	00075
CROSSTALK_CHECK no HV	custom PixScan	00076
power off	tool	00077

8.2 Example of the initial cut settings of Module Analysis

```
#This file contains the settings for Wafer Analysis, sorted in different sections.
#For example it contains a list of scans, DCS names, chip calibration, Global register names
and
#PixControler settings that will be analyzed.
#For help see ReadMeForSettings.txt


[Scans]
```

```
#RXDelay=rx delay scan
#GR0=GR test all 0
Global Register Test=GR_TEST
DigitalTest=DIGITAL_TEST 2
AnalogTest=ANALOG_TEST
ThresholdScan=THRESHOLD_SCAN with HV
ThresholdScanNoHV=THRESHOLD_SCAN no HV
ThresholdScanInTime=INTIME_THRESH_SCAN
CrossTalkScanWithHV=CROSSTALK_CHECK with HV
CrossTalkScanNoHV=CROSSTALK_CHECK no HV
GADC_VDDAScan_FE0=GADC VDDA Scan FE0
GADC_VDDAScan_FE1=GADC VDDA Scan FE1
GADC_PlsrDACscan_FE0=GADC PulserDAC scan FE0
GADC_PlsrDACscan_FE1=GADC PulserDAC scan FE1
IVscan=IV_SCAN
SourceScan=FE_ST_SOURCE_SCAN Am
LV_Scan_FE0_VDDA=LV_MEAS_FE0_VDDA
LV_Scan_FE0_VDDD=LV_MEAS_FE0_VDDD
LV_Scan_FE1_VDDA=LV_MEAS_FE1_VDDA
LV_Scan_FE1_VDDD=LV_MEAS_FE1_VDDD

[DCSGraphAnaValues]
#IDDA1AP=IDDA1_UNCFG
#IDDA2AP=IDDA2_UNCFG
#IDDD1AP=IDDD1_UNCFG
#IDDD2AP=IDDD2_UNCFG
#IDDA1M=IDDA1_MIDI
#IDDA2M=IDDA2_MIDI
#IDDD1M=IDDD1_MIDI
#IDDD2M=IDDD2_MIDI
#IDDA1CONF=IDDA1_AFTER_CFG
#IDDA2CONF=IDDA2_AFTER_CFG
#IDDD1CONF=IDDD1_AFTER_CFG
#IDDD2CONF=IDDD2_AFTER_CFG
#BgAn=BgRef_An
#BgDig=BgRef_Dig
#BgAn0=BgRef_An0
#BgDig0=BgRef_Dig0
#BgAn1=BgRef_An1
#BgDig1=BgRef_Dig1
#BgAn2=BgRef_An2
#BgDig2=BgRef_Dig2
#IrefST=Iref Scan_ana

[GlobalRegValues]
#VthinAF=GlobalRegister_Vthin_AltFine
#SetSN=GlobalRegister_Chip_SN

[PixCrtlSetting]
#Iref_set=general_IrefPads

[ChipCalibration]
#PlsrDACoffs=Misc_VcalGradient0
#PlsrDACslopeST=Misc_VcalGradient1
#CapCalibST=Misc_CInjHi
```

```
[AddToTotalCount]
#AnalogTest=1
#AnalogTestChigh=1
#AnalogTestClow=1
DigitalTest=1
ThresholdScan=1
#ScurveFailed=1
#HitOrScan=1
#BufferTest=1
#CrossTalkScan=1
#CheckPixelDisable=1
#LatencyTest=1
#PR=1
#ESLtest=1
#ESLtestAfterECR=1
CrosstalkScanWithHV=1

[DACscanAnalysisSettings]
GADC_VDDAScan_FE0:VrefAnTune [DAC]:GADC [ADC Counts]:1:1e3:1e0
GADC_VDDAScan_FE1:VrefAnTune [DAC]:GADC [ADC Counts]:1:1e3:1e0
GADC_PlsrDACscan_FE0:PlsrDAC [DAC]:GADC [ADC Counts]:1:1e1:1e0
GADC_PlsrDACscan_FE1:PlsrDAC [DAC]:GADC [ADC Counts]:1:1e1:1e0
LV_Scan_FE0_VDDA:VrefAnTune [DAC]:Supply Voltage [V]:1e3:1e1:1e0
LV_Scan_FE0_VDDD:VrefDigTune [DAC]:Supply Voltage [V]:1e3:1e1:1e0
LV_Scan_FE1_VDDA:VrefAnTune [DAC]:Supply Voltage [V]:1e3:1e1:1e0
LV_Scan_FE1_VDDD:VrefDigTune [DAC]:Supply Voltage [V]:1e3:1e1:1e0

[PostProcessing]
#AddAptasicColumn=1
#CheckForAbortRun=1
TotalCount=1
#CorrectVrefs=1
CalculateResult=1
CheckBumpConnection=1
CheckPixelShorts=1

#CompareResults=1

[CheckForAbort]
#IDDA1AP
#IDDA2AP
#IDDD1AP
#IDDD2AP
#IDDA1CONF
#IDDA2CONF
#IDDD1CONF
#IDDD2CONF

[CalculateResult]
#ThrMeanCorr=CapCalib/CapCalibST*PlsrDACslope/PlsrDACslopeST*ThrMean
#ThrSigCorr=CapCalib/CapCalibST*PlsrDACslope/PlsrDACslopeST*ThrSig
#NoiMeanCorr=CapCalib/CapCalibST*PlsrDACslope/PlsrDACslopeST*NoiMean
#NoiSigCorr=CapCalib/CapCalibST*PlsrDACslope/PlsrDACslopeST*NoiSig
```

```
[CorrectBandGaps]
#BgAn0=IDDA1AP:IDDA2AP:IDDD1AP:IDDD2AP
#BgDig0=IDDA1AP:IDDA2AP:IDDD1AP:IDDD2AP
#BgAn1=IDDA1CONF:IDDA2CONF:IDDD1CONF:IDDD2CONF
#BgDig1=IDDA1CONF:IDDA2CONF:IDDD1CONF:IDDD2CONF
#BgAn2=IDDA1M:IDDA2M:IDDD1M:IDDD2M
#BgDig2=IDDA1M:IDDA2M:IDDD1M:IDDD2M

[CorrectTunVrefs]
#VrefDigMin:BgDig1:BgAn1
#VrefDigMax:BgDig1:BgAn1
#VrefAnMin:BgDig1:BgAn1
#VrefAnMax:BgDig1:BgAn1

[CapacitanceCorrection]
#Offset=-1360.
#Dividor=1.081
#Voltage=1000

[CompareResults]
#PlsrDACslopeST=PlsrDACslope:3:Blue
#CapCalibST=CapCalib:3:Blue
#IrefST=IrefBestDAC:0:Blue

[ZoomDistributionPlots]
#IDDD2AP
#TotalPixelsFail
#PixAnClowFail

[ExportToXML]
status=Status
#CapCalib=InjectionCapacitance
#TotalPixelsFail=FailingPixels
#PixAnFail=AnalogFailingPixels
#PixDigFail=DigitalFailingPixels
#TotalColumnsFail=FailingColumns
#ThrMeanCorr=ThresholdMean
#NoiMeanCorr=NoiseMean
#PlsrDACslope=PlsrDACslope
#IDDA2CONF=DigitalCurrent
#IDDD2CONF=AnalogCurrent
#RunAborted=RunAborted
#ChipSN2=SerialNumber

[CheckPixelShorts]
SearchDistance=-1/-1:-1/0:-1/1:0/-1:0/0:0/1:1/-1:1/0:1/1

[IVcurveAnalysis]
OperationPointSingleChip=-15
OperationPointDoubleChip=-80
SlopeAtBreakdown=0.0000001
CurrentLimit=0.00002
OPfitRange=1

[ExportOverviewData]
```

```
ChipNr:Module
IrefBestDAC:Iref
CapCalib:Cin[aF]
PlsrDACslope:VcalSlope[uV/DAC]
status:Status

[PlotSettings]
OutputPath=../results
STcontrolColorPalette=0
MapPlotRangeInSigma=8
HistogramPlotRangeInSigma=16
ScatterPlotRangeInSigma=24
1DHistFillColor=5
ScatterDotColor=1

[ConfigDataFilePath]
../config

#Cut settings configuration file. For help see ReadMe.txt

[File]
version=10.10

[FailPixel]
DigitalHits!=200
AnalogHits!=200
AnalogClowHits!=200
AnalogChighHits>200
AnalogHitsShortedPixel<5
HitOrDisHits!=0
HitOrEnHits!=10
LatencyEnHits!=100
LatencyDisHits!=0
BufferActualToTHits!=5
#CrossTalkNoHVHits!=0
CrossTalkShortedPixelHits!=0
DisabledHits!=0
#ScurveChi2<1
#ScurveChi2>100
ThreshSigmaOffset<5
ThreshSigmaOffset>5
NoiseSigmaOffset>5
ESLhits>1
#NoiseSigmaOffset<3
BumpConNoiDiff<10
BumpConScurveChi2<1
BumpConScurveChi2>100
BumpConSourceOcc<1
CrossTalkWithHVHits!=0
CrossTalkNoHVHits!=0
ThreshSigmaOffsetNoHV<5
ThreshSigmaOffsetNoHV>5
NoiseSigmaOffsetNoHV>5
SourceScanRelHits<0.10
```

```
SourceScanRelHits>4.5

[FailColumn]
TotalFailPixel>20
AnalogFailPixel>20
AnalogChighFailPixel>20
AnalogClowFailPixel>20
DigitalFailPixel>20
HitOrDisFailPixel>5
HitOrEnFailPixel>1
LatencyEnFailPixel>5
LatencyDisFailPixel>20
CrossTalkFailPixel>20
DisabledFailPixel>20
BufferActualToTFailPixel>20
ThresholdFailPixel>50
ScurveFitFailPixel>20
LatchENABLEFailPixel>20
LatchCAP0FailPixel>20
LatchCAP1FailPixel>20
LatchILEAKFailPixel>20
LatchTDAC0FailPixel>20
LatchTDAC1FailPixel>20
LatchTDAC2FailPixel>20
LatchTDAC3FailPixel>20
LatchTDAC4FailPixel>20
LatchFDAC0FailPixel>20
LatchFDAC1FailPixel>20
LatchFDAC2FailPixel>20
LatchFDAC3FailPixel>20
BumpConFailPixel>20
CrossTalkWithHVFailPixel>5
#CrossTalkNoHVFailPixel>5
ThresholdFailPixelNoHV>50
ScurveFitFailPixelNoHV>20

[RedChip]
IDDA1AP<-50
IDDA1AP>50
IDDD1AP<-50
IDDD1AP>150
IDDA2AP<-10
IDDA2AP>300
IDDD2AP<-10
IDDD2AP>150
IDDA1M<-50
IDDA1M>150
IDDD1M<-10
IDDD1M>150
IDDA2M<-10
IDDA2M>300
IDDD2M<-10
IDDD2M>500
IDDA1CONF<-50
IDDA1CONF>150
```

```
IDDD1CONF<-10
IDDD1CONF>150
IDDA2CONF<-10
IDDA2CONF>300
IDDD2CONF<-10
IDDD2CONF>500
DigitalFailPixel>10000
AnalogFailPixel>10000
AnalogChighFailPixel>10000
AnalogLowFailPixel>10000
ThresholdFailPixel>20000
LatencyEnFailPixel>4000
RunAborted!=0

CrossTalkWithHVFailPixel>200
CrossTalkWithHVFailColumns>2
#CrossTalkNoHVFailPixel>200
#CrossTalkNoHVFailColumns>2
ThresholdFailPixelNoHV>20000

[YellowChip]
IDDA2AP>114
IDDD2AP<0
IDDD2AP>20
IDDA2CONF<103
IDDA2CONF>125
IDDD2CONF<240
IDDD2CONF>300
Iref<1940
Iref>2060
IrefBestDAC<2
IrefBestDAC>13
IrefMax<2200
IrefMin>1800
PlsrDACmin>148
PlsrDACmin<30
PlsrDACmax<1000
PlsrDACmax>1400
GlobalFailRegister!=0
AnalogFailColumn!=0
AnalogClowFailColumn!=0
AnalogChighFailColumn!=0
DigitalFailColumns!=0
DisabledFailColumns!=0
LatencyFailColumns!=0
HitOrEnFailColumn!=0
LatencyEnFailColumn!=0
LatencyDisFailColumn!=0
InjDelMonotony>6000
HighFreqCurr>357
#0.2% of the pixels are allowed to fail
TotalPixelsFail>54
TotalColumnsFail!=0
#PlsrDACkink>90
#PlsrDACkink<60
```

```
#HitOrEnFailPixel>0

CrosstalkWithHVFailPixel>100
CrosstalkWithHVFailColumn!=0
#CrosstalkNoHVFailPixel>100
#CrosstalkNoHVFailColumn!=0

[BlueChip]
BumpConFailPixel>20
SC_CMD!=1
SC_DOB!=1
SC_ECL!=1
IDDD2AP>9
IDDA2AP<93
BufferActualToTFailColumn!=0
IrefLineFitQuality>6500
IrefLineFitQuality<1000
IrefSlope>-67
IrefSlope<-91
RXvalley<800
PlsrDACmax>1500
PlsrDACmin<0
PlsrDACkink<60
CapCalib>6800
CapCalib<5000
CapCalibLineFitQuality>100
VrefAnLineFitQuality>2000
VrefAnSlope<-1200
VrefAnSlope>-700
VrefDigLineFitQuality>2000
VrefDigSlope<-1200
VrefDigSlope>-700
HighFreqCurr<260
HitOrDisFailColumn!=0
ESLFailPixel>400
CrossTalkFailColumns!=0
ThresholdMean<2000
ThresholdMean>9000
ThresholdMeanSigma>100
#NoiseMean>200
#NoiseMean<110
PlsrDACslope<800
PlsrDACslope>2000
PlsrDAClineFitQuality>5000
NoiSigCorr<5
NoiSigCorr>25
NoiMeanCorr>150
NoiMeanCorr<90
ThrSigCorr<400
ThrSigCorr>1200
ThrMeanCorr>7000
ThrMeanCorr<1000
ThresholdDistFitQuality>30
ScurveFitFailPixel>300
NoiseDistFitQuality>40
```

```
AnalogFailPixel>54
AnalogChighFailPixel>54
AnalogClowFailPixel>54
DigitalFailPixel>54
HitOrDisFailPixel>5
HitOrEnFailPixel>54
LatencyEnFailPixel>54
LatencyDisFailPixel>54
CrossTalkFailPixel>54
DisabledFailPixel>54
BufferActualToTFailPixel>54
LatchENABLEFailPixel>54
LatchCAP0FailPixel>54
LatchCAP1FailPixel>54
LatchILEAKFailPixel>54
LatchTDAC0FailPixel>54
LatchTDAC1FailPixel>54
LatchTDAC2FailPixel>54
LatchTDAC3FailPixel>54
LatchTDAC4FailPixel>54
LatchFDAC0FailPixel>54
LatchFDAC1FailPixel>54
LatchFDAC2FailPixel>54
LatchFDAC3FailPixel>54
LatchENABLEFailColumn!=0
LatchCAP0FailColumn!=0
LatchCAP1FailColumn!=0
LatchILEAKFailColumn!=0
LatchTDAC0FailColumn!=0
LatchTDAC1FailColumn!=0
LatchTDAC2FailColumn!=0
LatchTDAC3FailColumn!=0
LatchTDAC4FailColumn!=0
LatchFDAC0FailColumn!=0
LatchFDAC1FailColumn!=0
LatchFDAC2FailColumn!=0
LatchFDAC3FailColumn!=0
#SR1_0>0
SR1_1>0
SR1_2>0
SR1_3>0
SR1_4>0
SR1_5>0
SR1_6>0
SR1_7>0
SR1_8>0
#SR1_9>0
#SR1_10>0
SR1_11>0
SR1_12>0
SR1_13>0
#SR1_14>0
SR1_15>0
#SR1_16>0
SR1_17>0
```

```
SR1_18>0
SR1_19>0
SR1_20>0
#SR1_21>0
#SR1_22>0
#SR1_23>0
SR1_24>0
SR1_25>0
SR1_26>0
SR1_27>0
#SR1_28>0
SR1_29>0
SR1_30>0
SR1_31>0
#SR2_0>0
SR2_1>0
SR2_2>0
SR2_3>0
SR2_4>0
SR2_5>0
SR2_6>0
SR2_7>0
SR2_8>0
#SR2_9>0
#SR2_10>0
SR2_11>0
SR2_12>0
SR2_13>0
#SR2_14>0
SR2_15>0
#SR2_16>0
SR2_17>0
SR2_18>0
SR2_19>0
SR2_20>0
#SR2_21>0
#SR2_22>0
#SR2_23>0
SR2_24>0
SR2_25>0
SR2_26>0
SR2_27>0
#SR2_28>0
SR2_29>0
SR2_30>0
SR2_31>0
#SR3_0>0
SR3_1>0
SR3_2>0
SR3_3>0
SR3_4>0
SR3_5>0
SR3_6>0
SR3_7>0
SR3_8>0
```

```
#SR3_9>0
#SR3_10>0
SR3_11>0
SR3_12>0
SR3_13>0
#SR3_14>0
SR3_15>0
#SR3_16>0
SR3_17>0
SR3_18>0
SR3_19>0
SR3_20>0
#SR3_21>0
#SR3_22>0
#SR3_23>0
SR3_24>0
SR3_25>0
SR3_26>0
SR3_27>0
#SR3_28>0
SR3_29>0
SR3_30>0
SR3_31>0
#SR4_0>0
SR4_1>0
SR4_2>0
SR4_3>0
SR4_4>0
SR4_5>0
SR4_6>0
SR4_7>0
SR4_8>0
#SR4_9>0
#SR4_10>0
SR4_11>0
SR4_12>0
SR4_13>0
#SR4_14>0
SR4_15>0
#SR4_16>0
SR4_17>0
SR4_18>0
SR4_19>0
SR4_20>0
#SR4_21>0
#SR4_22>0
#SR4_23>0
SR4_24>0
SR4_25>0
SR4_26>0
SR4_27>0
#SR4_28>0
SR4_29>0
SR4_30>0
SR4_31>0
```

```
#SR5_0>0
SR5_1>0
SR5_2>0
SR5_3>0
SR5_4>0
SR5_5>0
SR5_6>0
SR5_7>0
SR5_8>0
#SR5_9>0
#SR5_10>0
SR5_11>0
SR5_12>0
SR5_13>0
#SR5_14>0
SR5_15>0
#SR5_16>0
SR5_17>0
SR5_18>0
SR5_19>0
SR5_20>0
#SR5_21>0
#SR5_22>0
#SR5_23>0
SR5_24>0
SR5_25>0
SR5_26>0
SR5_27>0
#SR5_28>0
SR5_29>0
SR5_30>0
SR5_31>0
IDDD2CONF<240
ThresholdFailPixel>3000
SC_CMD>1
SC_DOB>1
SC_ECL>1
InjDelMinLVL1<1700
InjDelMinLVL1>1900
InjDelMaxLVL1<4200
InjDelMaxLVL1>5400
InjDelMaxLVL1>5400
InjDelMonotony<1000
#Iref<1970
#Iref>2030
#IrefMin>1530
#IrefMax<2360
#CrossTalkFailPixel>5
#BufferActualToTFailPixel>50
#BufferAllToTFailPixel>0
#VrefAnMin>750
#VrefAnMax<750
#VrefDigMin>600
#VrefDigMax<600
#BgAn<630
```

```
#BgAn>670
#BgDig<580
#BgDig>620
#HitOrDisFailColumn>0

CrosstalkWithHVFailPixel>53760
CrosstalkWithHVFailColumn>160
#CrosstalkNoHVFailPixel>53760
#CrosstalkNoHVFailColumn>160
ThresholdMeanNoHV<2000
ThresholdMeanNoHV>9000
ThresholdMeanSigmaNoHV>100
#NoiseMeanNoHV>200
#NoiseMeanNoHV<110
NoiSigCorrNoHV<5
NoiSigCorrNoHV>25
NoiMeanCorrNoHV>150
NoiMeanCorrNoHV<90
ThrSigCorrNoHV<400
ThrSigCorrNoHV>1200
ThrMeanCorrNoHV>7000
ThrMeanCorrNoHV<1000
ThresholdDistFitQualityNoHV>30
ScurveFitFailPixelNoHV>300
NoiseDistFitQualityNoHV>40
```

8.3 Definition of the PDB module fields

Identifier:

1. Sensor Identifier (int)
2. Sensor vendor (string)
3. FE0 serial number (int)
4. FE1 serial number (int)
5. Assembly/Test Lab (string)

Status:

6. Global Status (bool)

Configurations:

7. ASSY at room temp. (.cfg.ROOT cfg. file)
8. FLEX at operation temp. (.cfg.ROOT cfg. file)
9. FLEX Source Scan Cfg. (.cfg.ROOT cfg. file)

Sensor:

10. Leakage current @ operational voltage in ASSY (float)
11. Leakage current @ less than operational voltage in ASSY (float)
12. dI/dV @ operaion point in ASSY (float)
13. Breakthrough voltag in ASSY (float)
14. Leakage current @ operational voltage in FLEX (float)

15. Leakage current @ less than operational voltage in FLEX (float)
16. dI/dV @ operaion point in FLEX (float)
17. Breakthrough voltag in FLEX (float)

Calibration: (Measured using Generic ADC in FE-I4B)
18. PulserDAC Slope (float)
19. PulserDAC Slope (float)

Measurement Results (filled from FLEX test):
20. VDDA unconfigured (float)
21. Analog current unconfigured (float)
22. VDDA configured (float)
23. Analog current configured (float)
24. Supply voltage unconfigured (float)
25. Supply Current unconfigured (float)
26. Supply voltage configured (float)
27. Supply Current configured (float)
28. Temperature using GADC flag (bool)
29. Minimum supply voltage (float)
30. Minimum digital reference voltage (float)
31. Minimum analog reference voltage (float)
32. Number disconected bumps (int)
33. Number of digital unresponsive pixels (int)
34. Number of analog unresponsive pixels (int)
35. Number unresponsive pixels in source scan (int)
36. Number unresponsive pixels total (int)
37. Number of noisy pixels (int)
38. Threshold distribution mean (float)
39. Threshold distribution width (float)
40. Noise distribution mean (float)
41. Noise distribution width (float)
42. Number of pixels with X-talk
43. ToT distribution mean @ target charge (float)
44. ToT distribution width @ target charge (float)
45. InTime threshold distribution mean (float)
46. T0 (float)

Mechanical Tests:
47. Mass (float)
48. Wire bond pull test (float)
49. Number of visible allignment marks (float)

Optical inspection:
50. Status (string)

51. Comments in opt. inspection (.txt file)

Raw Data:
51. Raw data in .ROOT format from ASSY (.zip archive)
52. Raw data in .ROOT format from BURN_IN (.zip archive)
53. Raw data in .ROOT format from FLEX (.zip archive)
54. Raw data in .pdf format from ASSY (.zip archive)
55. Raw data in .pdf format from BURN_IN (.zip archive)
56. Raw data in .pdf format from FLEX (.zip archive)
57. Summary of ASSY (.txt file)
58. Summary of FLEX (.txt file)

Comments:
59. Comments (.txt)
60. Number of reworked wire bonds on FE0 (int)
61. Number of reworked wire bonds om FE1 (int)

Spares:
62. Spare2 (string)
63. Spare3 (string)
64. Spare4 (int)
65. Spare5 (int)
66. Spare6 (int)
67. Spare7 (float)
68. Spare8 (float)
69. Spare9 (float)
70. Spare10 (float)