ALPIDE general stuff & lab measurements

David
Feli
Feli Question
Maurice (?)
Bogdan

About ALPIDE

Please fill with general ALPIDE stuff from the operations manual and other docs :)
Stuff like: how many pixels, how are they numbered, what is a priority encoder, what is inside each pixel, how many charges are created for a passing MIP, what happens to the charge from when it gets created until we have a hit/no hit in a pixel, talk about thresholds (check block diagram and please explain it).
One Chip contains 524,288 pixels arranged in 512 rows and 1024 columns. If we look at the chip from the circuit side, the rows are numbered from 0 to 511 arranged from the top to the bottom of the chip, whereby the bottom is defined as the location of the periphery, so the part of the chip that takes care of biassing, readout and control of the pixels. The columns are numbered from 0 to 1023 arranged from the left to the right. The address of a pixel is determined by the row and column number.
The readout of the address is executed in the first instance by the priority encoder. Every priority encoder is connected to the pixels of two columns and hence 512 of them are integrated on a chip. If some pixels got a hit, the priority encoder selects one of them, calculates their address and transmits it to the periphery. Afterwards it deletes the memory of the pixel and the next pixel will be selected. The Multi Event Buffer (MEB) in the pixel can store information of up to 3 events. The MEB is where the priority encoder reads the information from. As mentioned below, the chip is divided in 32 regions. This can be looked at in Fig. 1. If one looks at the chip with the periphery on the bottom the pixel is separated into 32 regions counting from 0 on the left to 31 on the right site. Each region contains 16 double columns arranged from 0 to 15 from left to right. The double columns are connected by the priority encoder (in Fig. 1: straight white lines between pixels). Every pixel of a double column has an index. The indexing starts at the top left corner with 0 and pursues like a sinuous line: Index 1 will be the top right corner pixel, index 2 will be below Index 1, index 3 will be to the left of index 2, index 4 below index 3. The procedure continues in this way down to the bottom left corner with the index 1023. The index of the pixel is called the pixel address (Visualization in Fig. 2). Thus the position of every pixel can be defined by the number of the double column according to the whole chip (0 to 511 form left to right) and the pixel address to identify each pixel while data taking. The priority encoders (PEs) can be read out simultaneously for different regions, but only one at a time per region. This means that 32 priority encoders can be read out simultaneously, since there are 16 PEs in every region, 15 run throughs have to be finished, before the same PE can be read out again.

During this time the events are stored in the mentioned MEB. (What happens if the MEB is full? Are new events simply not stored, or are old ones overwritten, how likely is this to happen?)
Very unlikely, depending on readout time x16

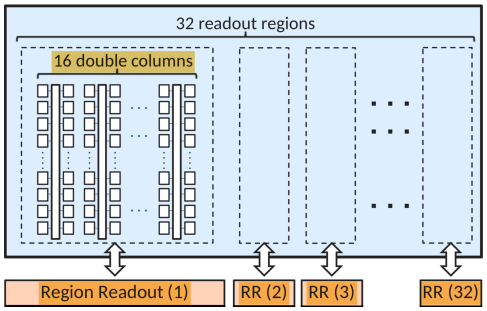


Fig. 1: Chip structure

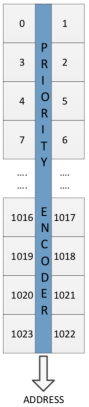


Fig. 2: Pixel address

Inside each pixel are components, which translate the produced charge of an incident particle in a readable signal for the priority encoder. A MIP usually creates a few thousand electron hole pairs in the silicon substrate of the pixel. Firstly we have the sensing diode, which collects the charge from the substrate and therefore produces a voltage pulse.
The front-end-stage with some amplifiers in it and the discriminator follows, which are shaping and triggering the signal. Firstly the signal becomes amplified. As a result the amplifiers will output a signal with a peaking time of around 2 µs. This is important to trigger the signal in the next stage. There the discriminator (=comparator) tests the signal on a specific editable threshold. If the front end output signal is too low, the discriminator won't give a signal as output. Only if the signal amplitude is higher than the Threshold setting, the discriminator will trigger and output a normalized square pulse. This pulse has a typical duration of up to 10 µs and is independent from the peak and trigger duration of the incoming signal.
Nextly the discriminated signal has to be stored until the priority encoder will read out the pixel memory. There are three latches to store the hit information in the so-called multi event buffer. The signal will only get through the gate (a logical AND) and hence stored in the MEB, if at the same time a STROBE signal is applied.

What are the internal DACs? What does each of them do? (in your own words try to briefly explain their effect on charge collection/reading, etc)
The DACs (Digital to Analog Converters) are used to configure the transistors of the front-end. By giving them a binary number (0 to 255), they input either current or voltage to the circuit over their transistor. In the front-end 6 voltage and 5 current (but on the referenced figure: 7 voltage and 4 current) DACs are placed. Over the DAC blocks one has the option to

directly connect the DAC-output to the pixel matrix, to monitor or to change the DAC-output. To monitor respectively change the output, the monitor pads DACMONV (for voltage) and DACMONI (vor current) are used.

Now we look on the impact of the DACs:
VRESET_D ensures that the charge of the collecting diode resets in certain periods, so the input signal “pix_in” can decrease to zero after a particle detection again. VRESET_D determines the reset voltage
The difference between VPLSE_LOW and VPLSE_HIGH sets the amount of charge, which will injected, e.g. for a Threshold test.
IBIAS is controlling the current of the source follower, directly connected to pix_in. A correct configured source follower then can let the incoming signal voltage propagate further.

VCASN, ITHR and IDB are influencing the discriminating threshold:
VCASN influences the baseline voltage proportionally. So for higher VCASN the baseline voltage becomes higher and so for the Threshold decreases
ITHR determines the signal pulse shape. For higher ITHR the pulse width and height are decresing and therefore the threshold increases.
IDB controls the current through its transistor (M7). The current have to exceed the corresponding value of IDB to activate the output node. For higher IDB the threshold increases.
VCASN2 counteracts the Miller effect (increasing capacitance).
VCASP?
VCLIP controls the clipping transistor. It determines the pulse length in the clipping mechanism.
IRESET?

To get an intuition for the DAC values and the corresponding ADC values (meaning the voltage/current corresponding to the DAC) a table 1. Will present all DACs, with their minimum and maximum ADCs and the default setting:

Internal DACs	Default (DAC)	Default (ADC)	Min. value (DAC=0)	Max. val.(DAC=255)
IBIAS	64	20 nA	0 nA	80 nA
ITHR	51	0.5 nA	0 nA	80 nA
IDB	64	10 nA	0 nA	80 nA
IRESET	50	5 pA	0.7 pA (nA in manual)	26 pA
VCASP	86	0.6 V	0 V	1.8 V
VCASN	57	0.4 V	0 V	1.8 V
VCASN2	62	0.44 V	0 V	1.8 V

VCLIP	0	0 V	0 V	1.8 V
VRESET_D	147	1.4 V	0.37 V	1.8 V
VPLSE_LOW	0	0.37 V	0.37 V	1.8 V
VPLSE_HIGH	255	1.8 V	0.37 V	1.8 V

One have to note, that the DAC values reasonably range from 0 to 255, which are corresponding to the minimum and maximum ADC value

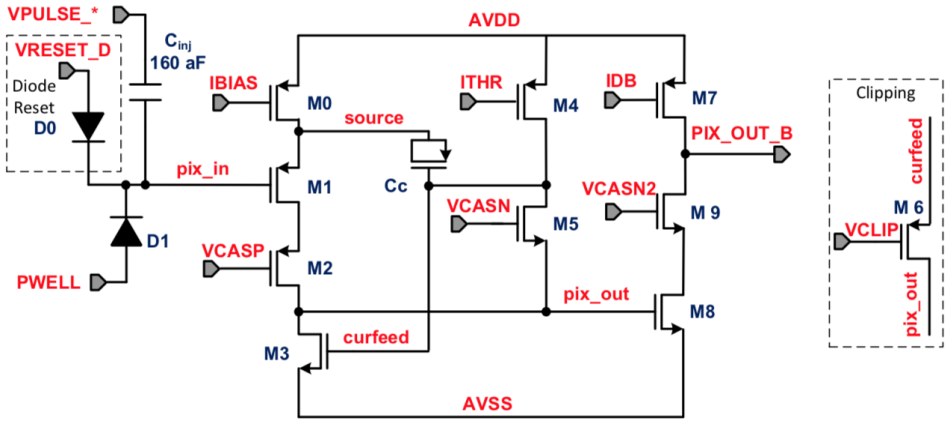


Fig. 3: Front-end scheme (taken from ALPIDE Operations Manual p.87 (Fig.4.1))

What are the regions of validity? Min, max (DAC and corresponding ADC).
What is the difference between analog and digital? What impact does it have for our chip?
An analog signal can carry information about multiple physical quantities. For example the Amplitude can tell us about the strength of the signal. This additional information is lost on a digital signal, where the signal itself just consists of either a 1 or a 0 (a hit or no hit). While it might be favourable in some situations to read out all the information of the analog signal, for the ALPIDE chip, it is important to know exactly *when* a particle hit *which* part of the chip, so a conversion to a digital signal, where one does not have to deal with noise or distortion after the signal exits the pixel, is advantageous.
What is a strobe signal? A strobe signal controls the storage of a discriminated hit in the MEB.
What is a busy or trigger signal? The busy signal controls datataking. If a chip goes in saturation of data processing capabilities, i.e. in triggered mode no buffer is available, busy signal will be enabled to stop latching data into the MEB, until one buffer is free again.
What do they control?
What readout modes do we have for ALPIDE? There are two main readout modes:
The triggered and the continuous one. In triggered mode the strobe signal will start only with an external trigger (i.e. scintillators) and lasts a few hundreds of ns. Furthermore the already stored data in the MEB is prioritised over new data, meaning that this will be discarded, if there is now free space.

In the continuous mode a periodically repeating signal replaces the external trigger to activate the strobe signal. In this mode the strobe signal is in the order of µs and hence longer than in triggered mode. Also, to be able to measure most of the time, since there is no trigger which tells us if there is something we want to detect, the duration between two strobe signals is very short (~100ns) compared to the signal itself. Unlike the triggered mode here the newest hit data is prioritized, meaning that stored data will be deleted, to free space for new data, even if it is not yet read out or not completely. In any case it will be reported, if some data was deleted.

What are analog/digital pulse injections? What do we achieve with them? Puls injection in means this context that a real pixel hit becomes emitated. The analog pulse is hereby injected through the charge injection capacitor. This directly imitates the pix_in signal. The digital pulse directly injects a pixel hit pulse and thus only the readout is still needed. "A digital-only pulsing mode is available, forcing the writing of a logic one in the pixel memory cells." (Alpide Manual).

One can use this injection methods to test, if the chip works properly and for example to determine its efficiency with specific settings. There are several tests, like a Threshold- or digital scan, which use this functionality.

Why do we prefer high resistivity substrates? What happens there?

A high resistivity substrate has the advantage that for the same biassing the depletion region is larger, compared to substrates with lower resistivity. This results in a better signal to noise ratio, meaning that the signal quality increases and therefore i.e. the efficiency increases. The width of the depletion region can be broadened by applying higher Vbb, this can range from 0V-6V, for our purposes -3V are used in order not to strain the chip too much. A greater depletion zone leads to more charge freed if a particle traverses.

What is the DAQ clock frequency? Why was it chosen as such? What does the clock provide?

Since the chip is designed for operation at the LHC at CERN, the nominal clock frequency of the chip is about 40.08 MHz. This is in agreement with. The reason for this frequence is the correspondence to the frequency, at which bunches are crossing at the LHC. The clock provides a timestamp for each event, so that they can be treated individually, and reconstruction/analyzation of the particle track is possible at a later stage.

Why do we like testbeams? Talk a bit about efficiency, position & timing resolution (if available). Clustering? (e vs heavy ions)

Test beams are important to characterize the chip precisely. They have the advantage of using highly relativistic particles with lots of information available. For example you would expect to know exactly what particles have been used, under what angle they enter the chip etc. They have the disadvantage of not always being available, owing to the fact that there are only few test beam facilities in europe.

We like testbeams, because those accelerate heavy ions on our target detector. The advantage of heavy particles compared to i.e. electrons is that they depose significantly more energy in the substrate. This results in e-h-pairs created in the process, which effects the detection efficiency positively, because it is more likely that the signal emerging out of more electrons overcomes the threshold.

Another point one has to consider is the effect of charge sharing. This effect describes that the charge created by an incident particle in the substrate of a pixel can move to other pixels

by diffusion. This seems to be stronger for increasing amounts of charge created in a pixel. This effect causes more than one pixel detecting a hit. A group of pixels next to each other, which detect a hit at the same timestamp is called cluster. Since the charge sharing effect is the main cause of clustering, higher energy deposit leads to larger clusters. Therefore heavy ions have bigger clusters as i.e. electrons.
Due to charge sharing and other effects the resolution of the detector can be higher than the actual pixel size, in our case a resolution 5µm is hoped for. As mentioned, charge sharing leads to a formation of clusters, which can either split, or without a gap (according to FP Sillicon Detectors Corryvreckan also takes split clusters into account). Clusters with a gap are not expected to occur though, except for when a pixel inside a cluster is masked, since that would require the pixels to in one chip to have a highly differing threshold.
The hit information has the shape of a box distribution with a uniform height. In case of one single pixel firing, the error, defining the resolution, can calculated by $\sigma_z = z/\sqrt{12}$ whereby z is the length of the pixel in z direction. For events with more than one pixel firing, the resolution becomes even better, since the position can be reconstructed better with more information over the distribution is given. Therefore a testbeam with heavy ions has the advantage of a better positional resolution.

Skim the documentation. Skip stuff about IB and OB, as they refer to the ITS2 design, not for single chips. Skip registry stuff.

Maybe also a bit of theory, to have it written down:

- MAPS vs hybrid sensors. A few words. Advantages/disadvantages
  - MAPS has the advantage of a low material budget and and the small sensitive size, one can reach with it, which leads to a higher resolution
  - The advantages of the MAPS are the disadvantages of the hybrid sensor, but the saving of material and size costs the MAPS readout speed and versatility, due to less charge collection compared to the hybrid sensor
- How is resistivity related to electron/hole mobilities
  - Resistivity ρ is the inverse of the conductivity $\sigma \sim (\mu_e + \mu_h)$, μ_i being the mobilities of the electrons/holes (hole equation: $\sigma = e\, n_i\, (\mu_e + \mu_h)$, n_i:intrinsic charge carrier concentration)
- Approximation formula for depletion region vs resistivity and voltage. How do we get to that formula
  - $d \simeq \sqrt{2\, \varepsilon\, \rho_n\, \mu_e\, (V_B)}$is the formula for the depletion depth d, with the dielectric constantε, the resistivity in the n-region and bias voltage V_B.
  - To get to this equation, one uses the poisson equation to describe the charge density distribution. By integrating this equation and combining the solutions for the p and n region, one can resolve the equation according to d. One get $d = \sqrt{2\, \varepsilon\, (V_B + V)\frac{(N_A + N_D)}{e\, N_A N_D}}$ with the donor and acceptor concentrations N_A, N_D and the diffusion voltage V. Assuming low doped N_D,meaning $N_A \gg N_D$and therefore the p-region-resistivity can be written as

$\rho_p \simeq (eN_D\mu_e)^{-1}$, neglectable V compared to V_B , we get to the above mentioned equation. One have also to assume thermal equilibrium.

- Charge carriers in Si: mobility, recombination, trapping, drift..
  - $\mu_e(300K) = 1350\frac{cm^2}{Vs}$
  - $\mu_h(300K) = 480\frac{cm^2}{Vs}$
  - Mobility defines the drift velocity with a certain electric field: $v = \mu E$
  - Recombination is the process of an electron and a hole annihilating and releasing a photon.
  - Due to impurities in a medium, recombination centers- an extra energy state in the gap of conduction and valence band- accur, where electrons and holes can get stuck for a period of time. If the corresponding partner for a annihilation travels by in this period, they can recombine. Therefore a recombination center is also called as trapping center
- Radiation damage
  - Radiation damage is a problem for semiconducting detectors, since they are sensitive to this effect and it limitates their lifetime.
  - This effect describes that incident particles, hitting the detector can knock out atoms out of the lattice structure. This results in lattice defects which form trapping centers. Account of this trapping centers, the energy resolution of the detector is directly affected. The resistivity of the substrate can change too. One can compensate this with an increased bias voltage, as well as the decreased resolution, if the damage is not too heavy.
  - Also leakage current can occur. If it exceeds a certain level, the detector can not be properly used anymore, as for every damage effect.

Lab measurements

What each (THR/NOISEOCC) tests do and why are they important. Why is fitting important?

THRESHOLD: Thresholds are important, because they are what ultimately determines whether a hit is detected or not. A threshold is set in Volts, if enough charge is freed inside the pixels depletion zone and is then collected by the collection diode a voltage pulse can be detected. If this is higher than a set threshold a hit is detected by the pixel. This principle can be seen in Fig. 4, for our digital signal the ToT (Time over Threshold) is not taken into consideration.

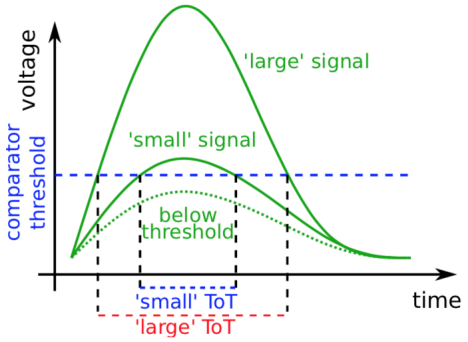


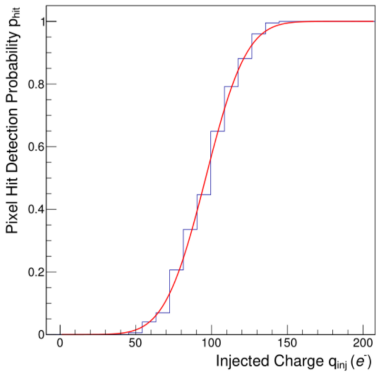
Fig. 4: Threshold Visualization

What we did is to influence the set threshold by varying several parameters (also mentioned above). These were: ITHR, VCASN and Vbb. VCLIP has to be set according to Vbb (Vbb=0: VCLIP=0 ; Vbb=-3: VCLIP=60).

We then looked at the results to determine how they influence the threshold, in order to then be able to set the threshold to a reasonable value for further measurements.

For the measurement a charge was injected on a pixel which ranged from 0 electrons to 200 electrons, 50 times for each charge value. The detector then measured, whether a pixel detected a hit for the injected charge or not (only appr.1% of pixels was hit, this can be set by varying NMASKSTAGES and PIXPERREGION).

For plotting injected charge on the x-axis, and hit detection or not, you would expect the graph to look like a step function, with the step at the set threshold. Due to noise and statistical fluctuations that is not exactly the case, but a smearing occurs, which turns the step function more into an s-curve. This can be seen in Fig. 5. The threshold is defined as the charge where the pixel hit detection probability reaches 50%. μ: Threshold; σ: temporal noise of a pixel.



$$p_{\text{Hit}}(q_{\text{inj}}) = \frac{1}{2}\left(1 + \text{Erf}\left[\frac{q_{\text{inj}} - \mu}{\sqrt{2}\sigma}\right]\right)$$

← Corresponds to a threshold value of roughly 100 e^-

Fig. 5: Smearing of threshold (from Maurice)

Fitting is important, because it is a model that gives key information about the bahviour of the measured quantity, for example for the threshold case a sigmoid seems to be a good fit, but a fit also needs to be motivated by the physics behind it. Thus, although a simple sigmoid fit appears to be a good fit, it lacks a physical motivation behind it, which is why a gaussian error function is preferable.
Once you have fit you are satisfied with you can from there on calculate an error for your value (gaussian error calculations, not statistical errors, those are possible in anyway, assuming you have enough measurements).
From the fit you can also calculate the χ^2 value. This is a value that gives an approximation of how well the measured data and fit go along, the closer χ^2 is to one, the better the fit.

NOISEOCCUPANCY: In order to detect a hit, an external trigger must tell the pixel that a particle has passed it, this could for example be scintillators.
In the case of a noise occupancy test the external trigger gives the signal to record, without a particle traversing. Thus the Chip measures whether it now detects a hit. If a pixel does indeed detect a hit it means that the set threshold has been surpassed. Since there has been no particle freeing charge it is either due to a faulty pixel, which then should be masked, or due to electronic noise. The noise is expected to be gaussian, which is why the Threshold is smeared in Fig. 5 and why a gaussian has been chosen for the fit.
Noise is closely related to the Fake Hit Rate (FHR). It is defined as:
N_hit: The numer of times a hit has been detected
N_pix: Number of pixels on chip (1024x512)
N_trg: The number of times the external trigger has sent the signal to record if a hit occurs

$$\text{FHR}_{\text{meas}} = \frac{N_{\text{hit}}}{N_{\text{pix}} \cdot N_{\text{trg}}}$$

What do we expect from them?
We expect to get a better understanding of how the chips work. For example we discovered that different chips seem to have quite differing resulting thresholds for the same settings (Some info about that can be found in the muon document).
Also we learned that the noise looks gaussian for our case as expected.

Notably we noticed that the distribution of thresholds looks like a Landau distribution and is thus not symmetrical around the mean value (Can we talk about this again?). Fig. 6 shows the distribution.

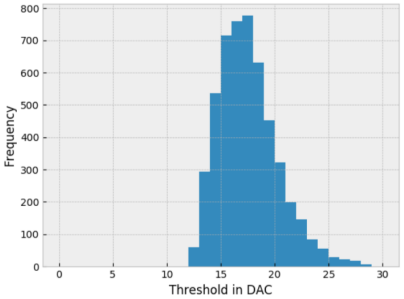


Fig. 6: Threshold distribution (Thanks Maurice :))

For each test, what do we do? What do we input, how is it treated inside the chip/pixel, what comes out? What do we do with that information?

For both threshold scan and noise occupancy test no actual source was used. Instead We used NIM logics to manually produce artificial signals. The chip proceeds as explained previously. The information from that leads to the results mentioned before.

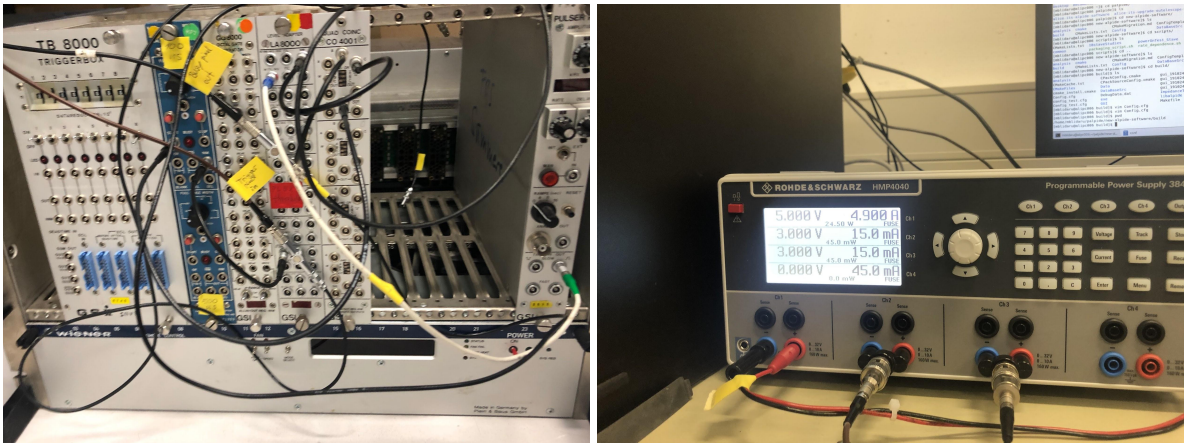


Image 1: The NIM logics inside the trigger box and setup used for both of the tests.

On the right side of Image 1 one can see the generator that has been used to power the chip. The left side shows the mentioned NIM Logics.