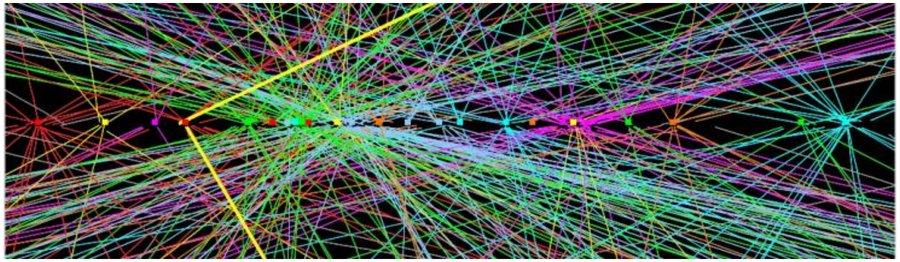
Characterization of Depleted Monolithic Active Pixel Sensor (DMAPS) Prototypes

TWEPP 2014
Topical Workshop on Electronics for Particle Physics

Leonard Germic, Miroslav Havranek, Tomasz Hemperek, Fabian Hügging, Tetsuichi Kishishita, Hans Krüger, Carlos Marinas, Theresa Obermann and Norbert Wermes
University of Bonn

Motivation: Pixel trackers for HEP experiments

universitätbonn



Pile up of interactions at High Luminosity LHC

→ Increased radiation dose

→ More pile up

Goal: Keep pixel tracker performance by using improved concepts!

	STAR	ALICE-LHC	ILC	ATLAS-LHC	ATLAS-HL-LHC
Timing [ns]	200 000	20 000	350	25	25
Particle Rate [kHz/mm²]	100	10	250	1000	10000
Fluence [n_{eq}/cm^2]	> 10^{12}	> 10^{13}	10^{12}	$2x10^{15}$	$2x10^{16}$
Ion. Dose [Mrad]	> 0.3	0.7	0.4	80	>500

MAPS

Hybrid

See talk by M. Winter in Plenary 6, Thursday 14:00

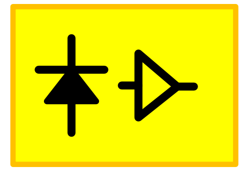
Which technologies can be used here?
Hybrid pixels, MAPS, something new .. ?

Concepts for future trackers

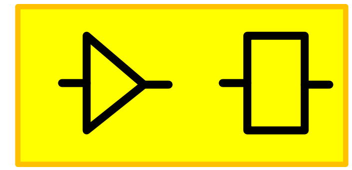
Hybrid Pixels with CMOS active sensor:

- **CMOS** (HV or HR) as a sensor (8”)
- Standard (or digital) only FE chip
- Example: CCPD (HVCMOS) on FE-I4 [1]

See talk by D. Münstermann in Plenary 8, Friday 10:10



Diode + preamp

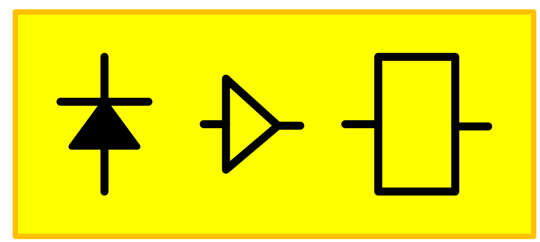


FE chip

Monolithic Active Pixel Sensor

- **MAPS** [2,3] usually on non-depleted epi substrate → **diffusion signal**, not suited for HL-LHC
- **Charge collection by drift → Fully depleted** MAPS (**DMAPS [4]**) on high resistive substrate
- Slow r/o (known from MAPS) usually does not meet LHC timing requirements

See poster by T. Kishishita



Diode + Amp + Digital

→ This talk is about test results from the sensor characterization of a DMAPS prototype

EPCB01 – Depleted Monolithic Pixel Sensor

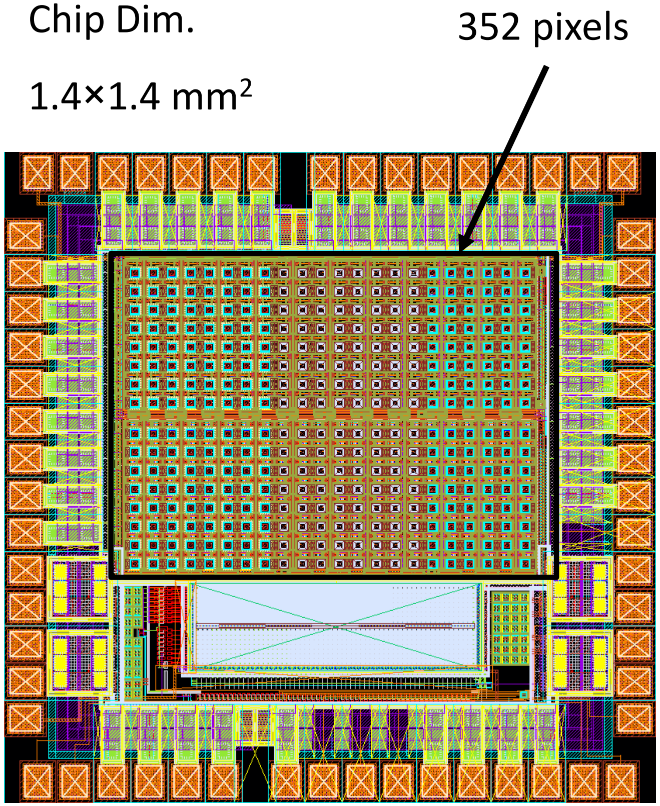
universitätbonn

Features

- Technology: ESPROS, 150 nm
- Deep p-type well to isolate full CMOS electronics
- 6 metal layers
- High resistive n-type bulk (> 2 kΩ cm)
- High voltage at sensor domain possible (~ 20 V)
- Backside p-implantation
- Thinned down to 50 µm

Chip Dim.

1.4×1.4 mm^2



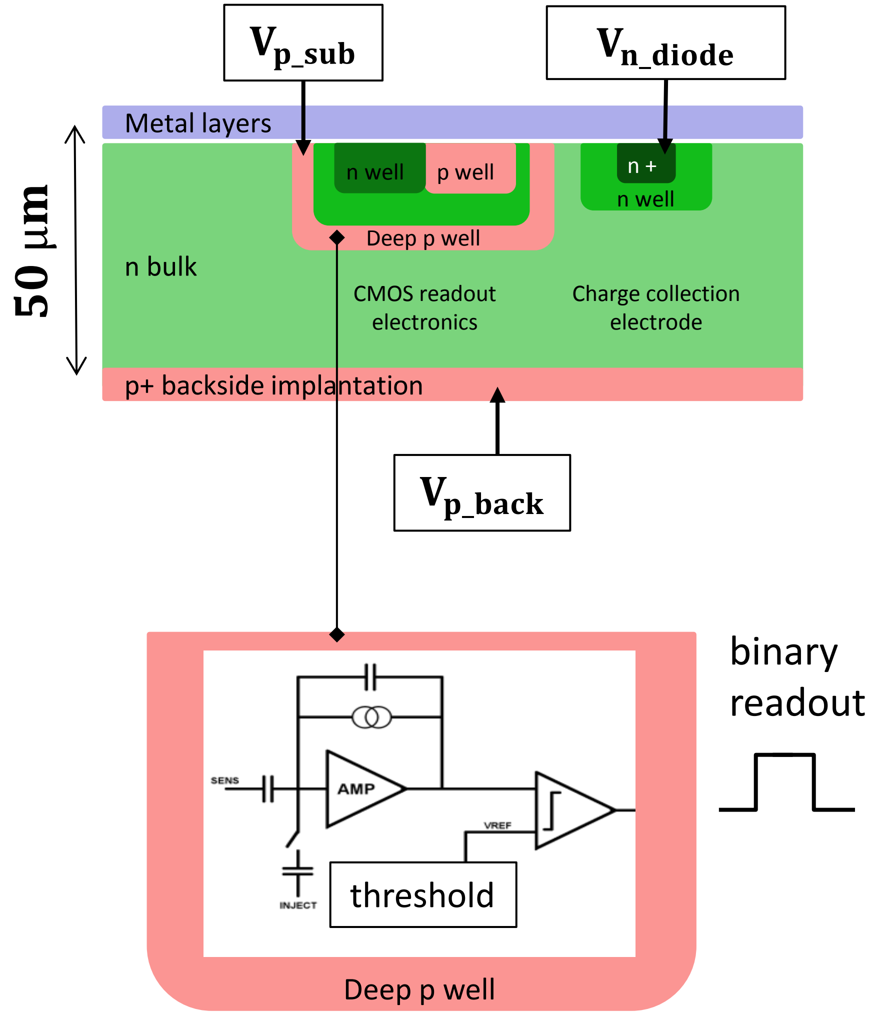
Pixel array features

- Pixel size: 40×40 µm^2
- Charge collecting n-implantation area ~ 20×20 µm^2

EPCB01 – Sensor and FE

Sensor

- N-implant is charge collecting electrode
- P-backside implantation
- Full CMOS electronics inside deep p-well



Front end

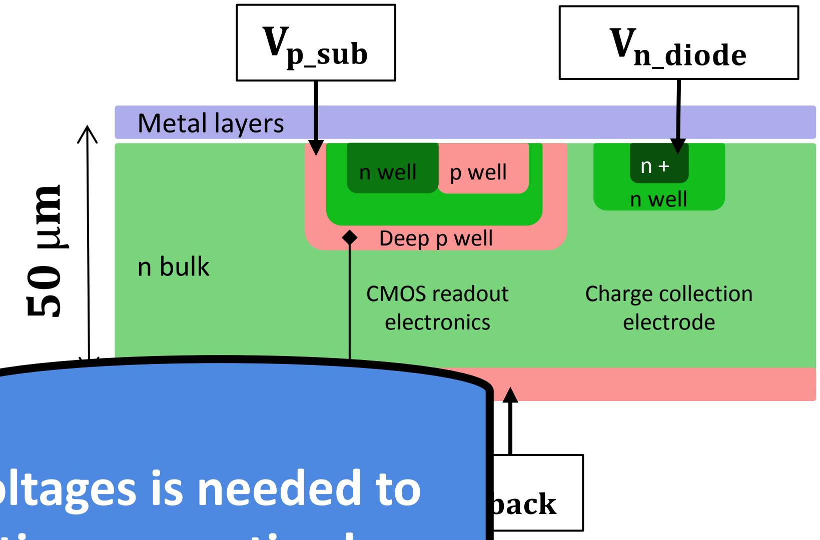
- CSA and comparator

EPCB01 – Sensor and FE

universitätbonn

Sensor

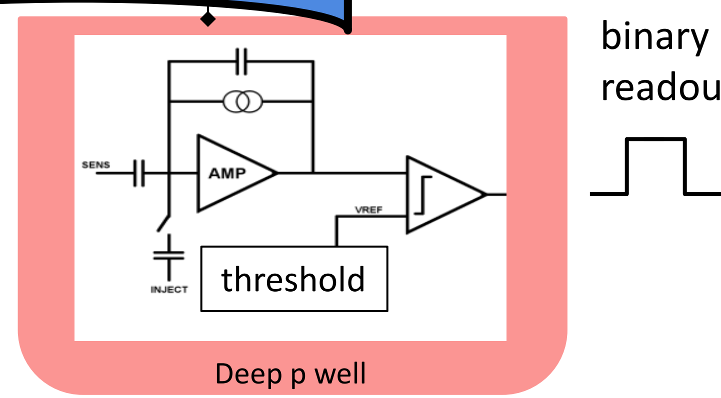
- N-implant is charge collecting electrode
- P-backside implantation
- Full CMOS electronics inside deep p-well



→ **Optimization of bias voltages is needed to characterize charge collection properties !**

Front end

- CSA and comparator



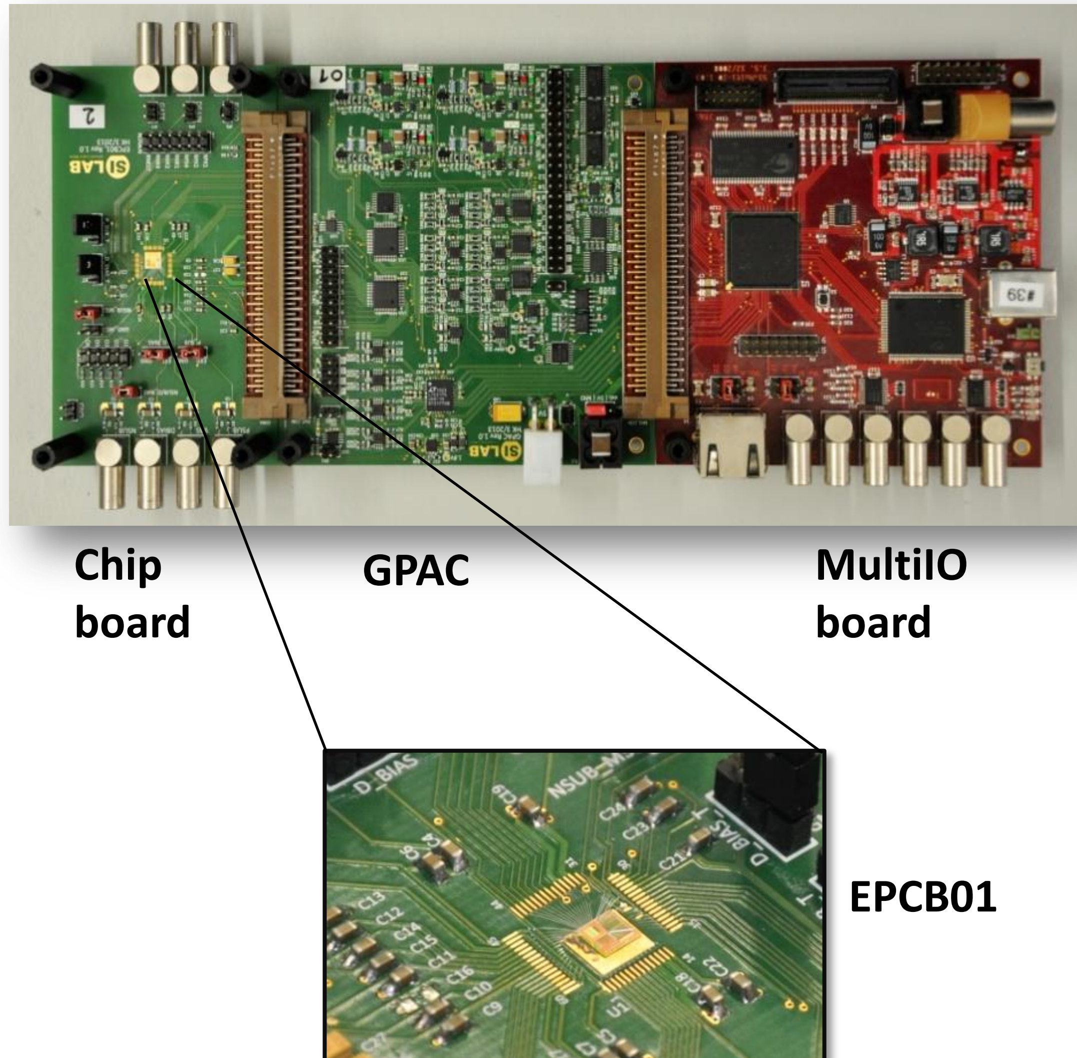
Test setup

Software

- Python project EPCB01 (based on modular test framework BASIL)

Hardware

- FPGA board: S3 MultiIO-board
- General purpose adapter card (analog and digital card with voltage sources, current sources, ADC, Injection circuitry)
- Chip carrier board



Sensor characterization measurements

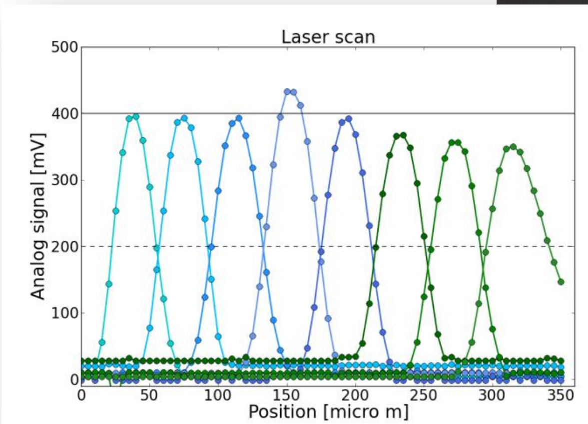
Laser

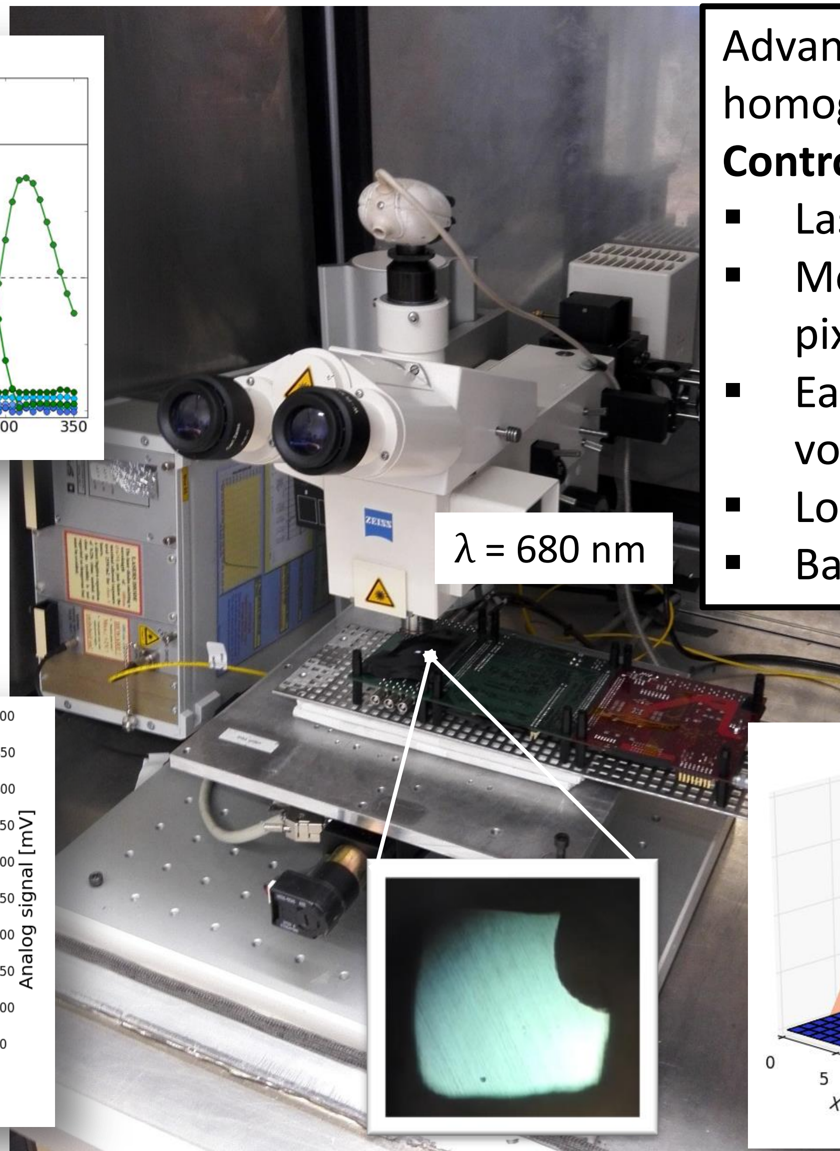
- Response homogeneity
- Bias parameter optimization

Sr 90 and electron beam

- Measure charge collected for MIP
- Cross check of laser results
- Relative CC dependence on bias parameters

Laser measurements

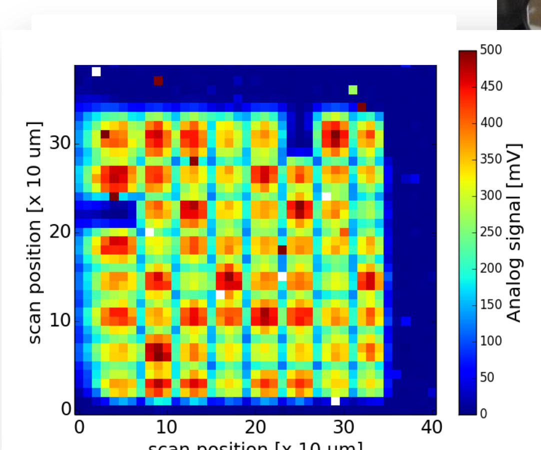


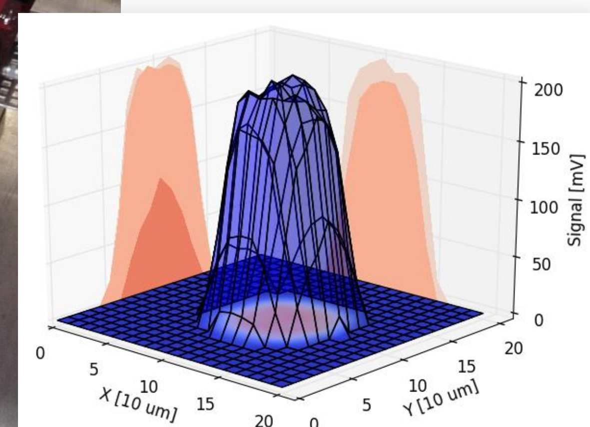


Advantages of laser system for homogeneity studies:

Controlled environment

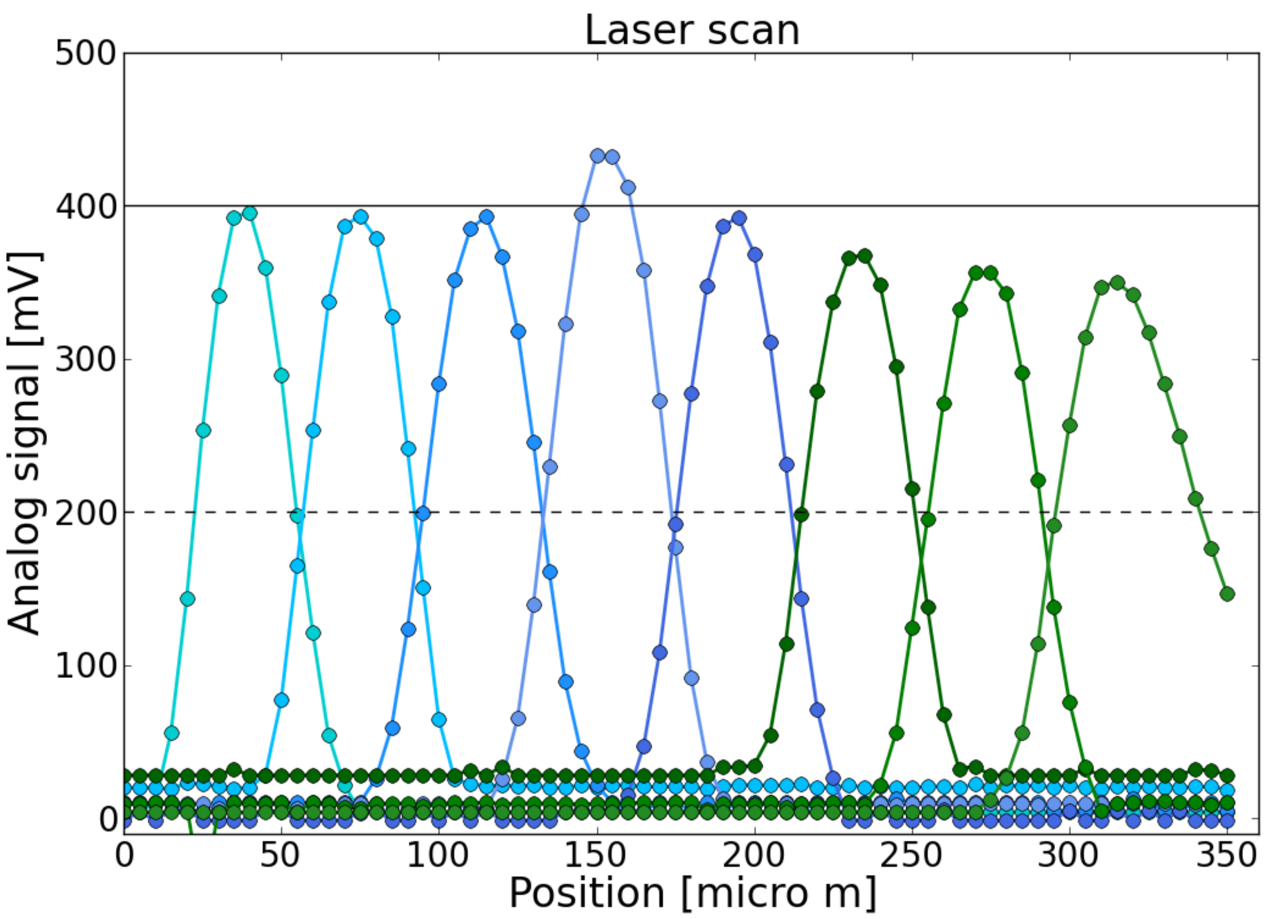
- Laser spot size ~ 3 µm
- Movable in 2 µm steps → In pixel studies
- Easy repitition → Fast voltage scans
- Long term stability ± 100 e
- Backside illumination





Laser scan 1D

universitätbonn



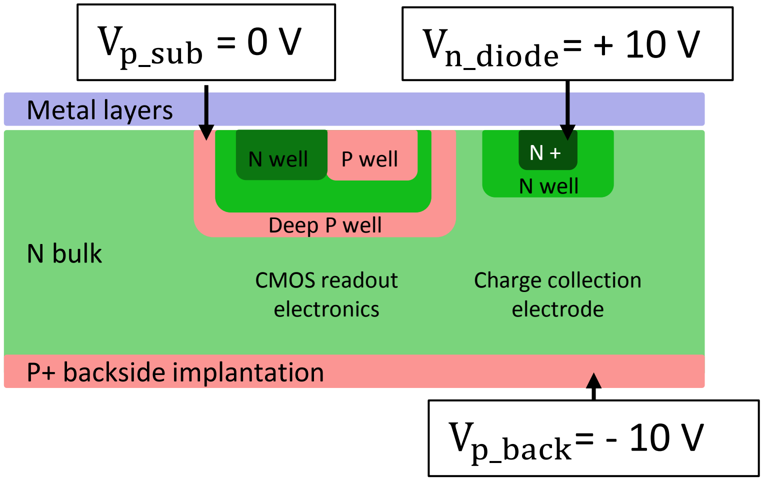
Laser moving direction

Laser scan over nine pixels

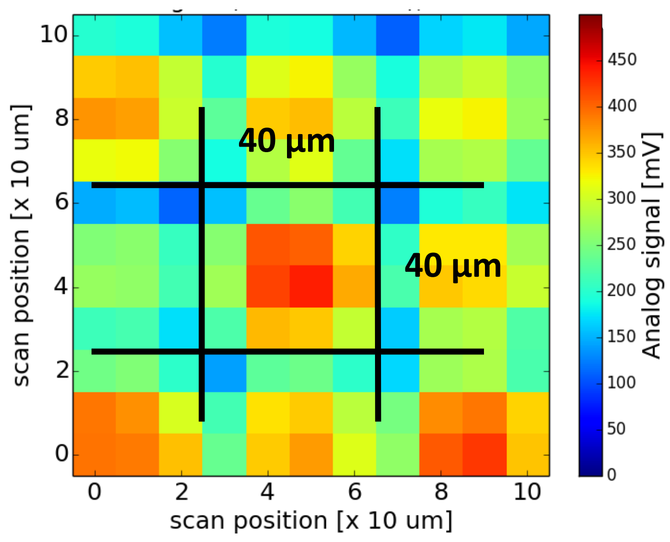
Scan settings

- Step size: 10 µm, 10 x 10 steps

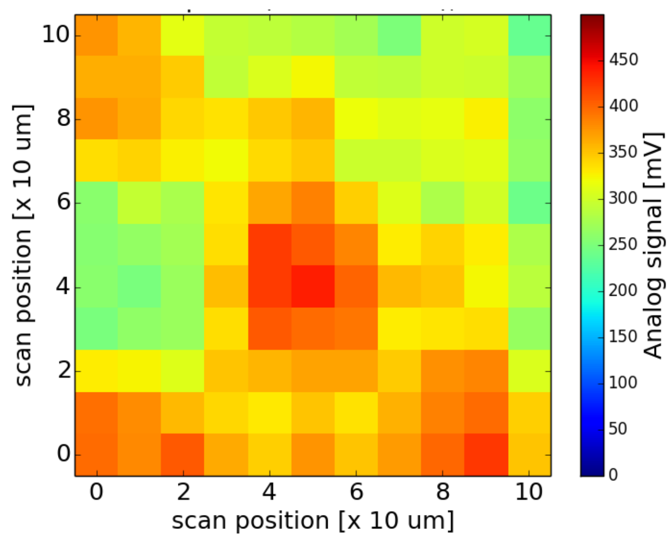
Remark: No fixed pattern noise correction (worst case)



Seed signal



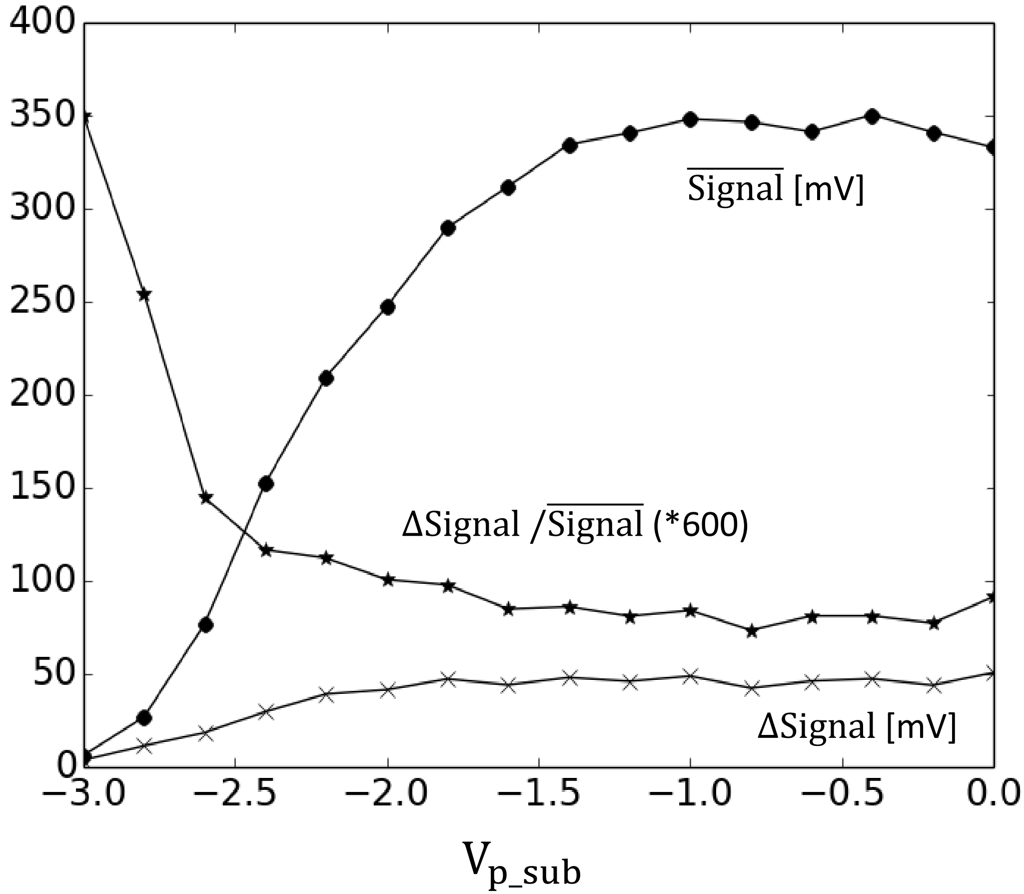
Cluster signal



→ Figure of merit: $\Delta\mathrm{Signal}/\,\overline{\mathrm{Signal}}$

Parameter scan of V_{p_sub} over nine pixels

ΔSignal and $\overline{\text{Signal}}$ versus V_{p_sub}

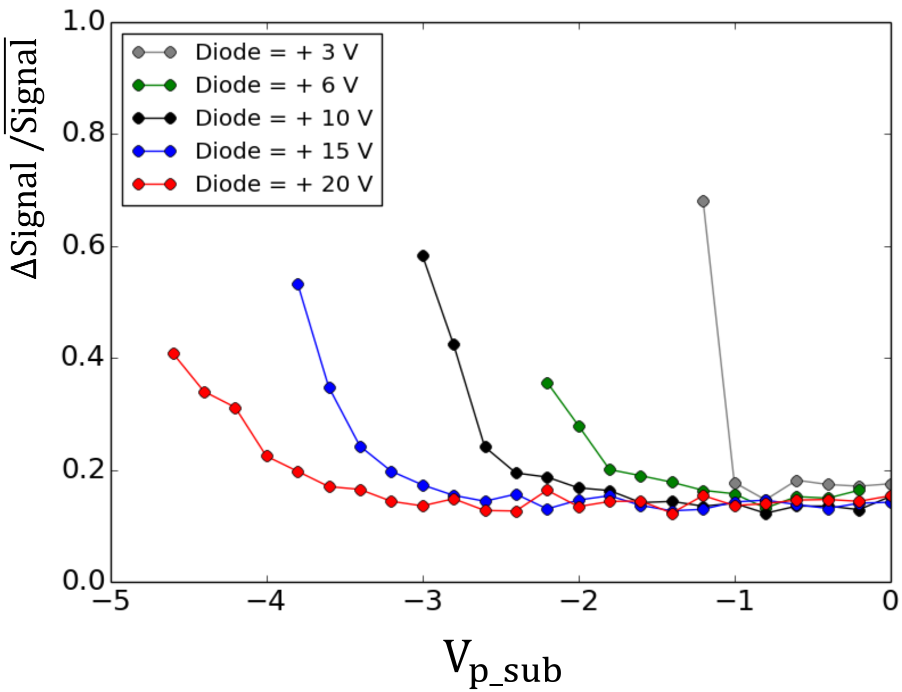


→ Best homogeneity
ΔSignal /$\overline{\text{Signal}}$ = 15 %
for V_{p_sub} from 0 to - 1.5 V

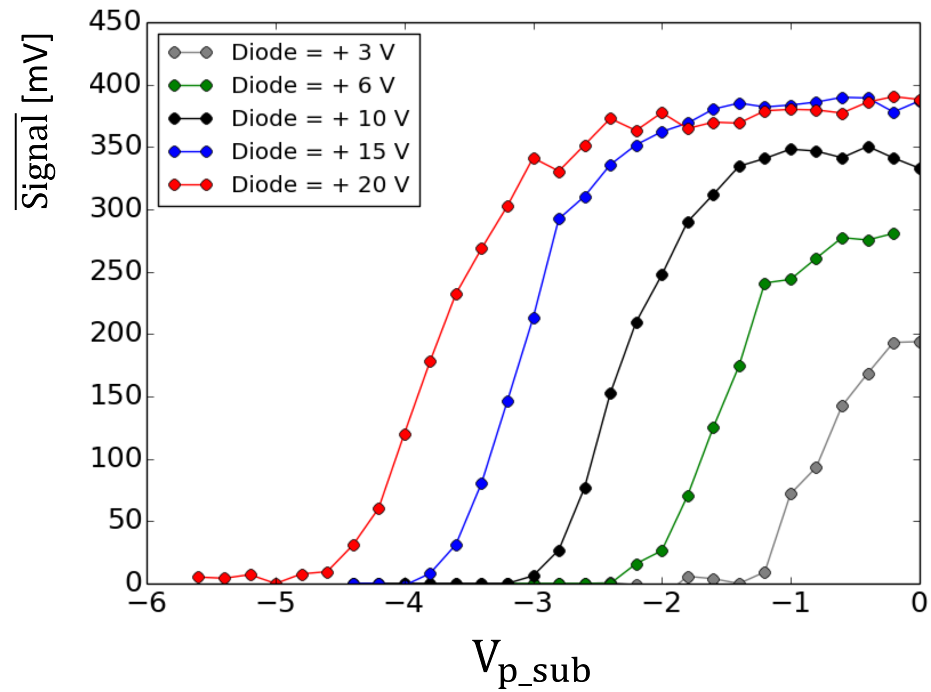
→ No charge collection
below V_{p_sub} = - 3 V

Parameter scan of V_{p_sub} and V_{n_diode} over nine pixels

Inhomogeneity



Amplitude



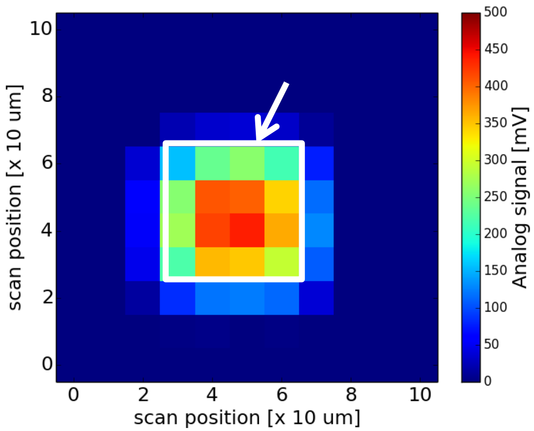
→ Depleted sensor at V_{n_diode} > + 15 V

→ With increasing V_{n_diode} , the range for V_{p_sub} with signal increases

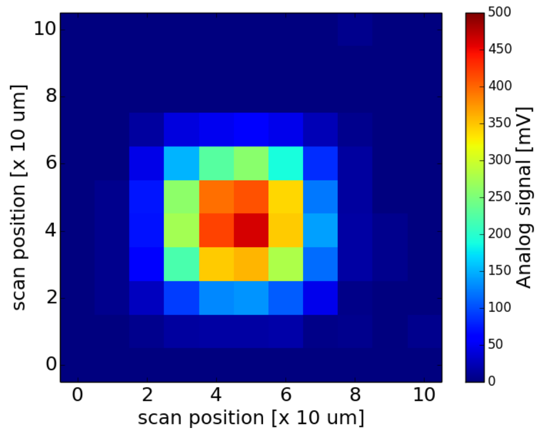
Signal of central pixel for different V_{p_sub}

universitätbonn

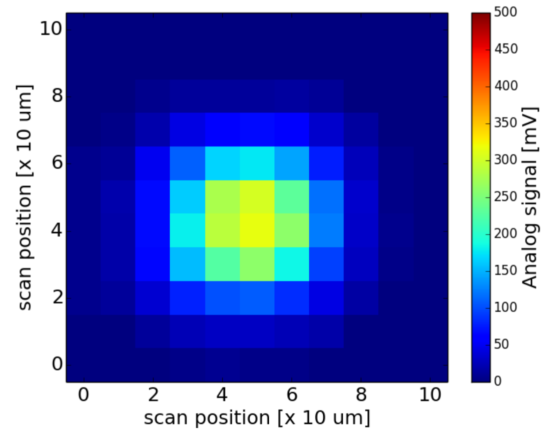
V_{p_sub} = 0 V



V_{p_sub} = - 1 V



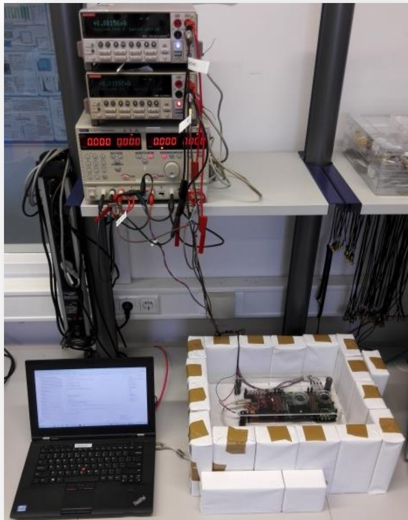
V_{p_sub} = - 2 V



→ Lowering V_{p_sub} the charge spread is larger and the seed signal is decreased

Sr 90 and electron beam measurements

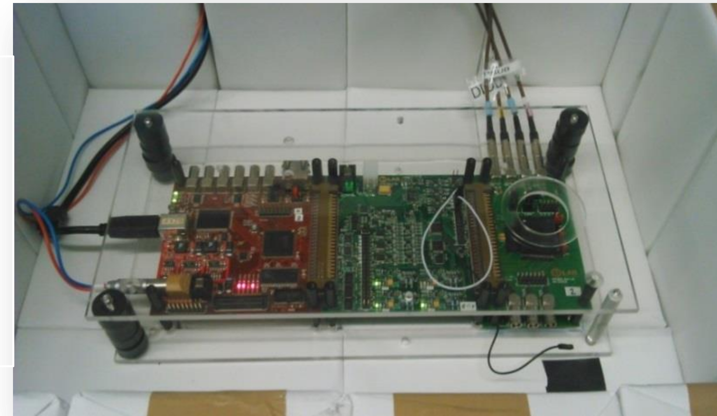
universitätbonn



← **Source setup** →

Sr90

Expect $Q_{Sr90}(MPV) = 3.8$ ke
in 50 µm silicon

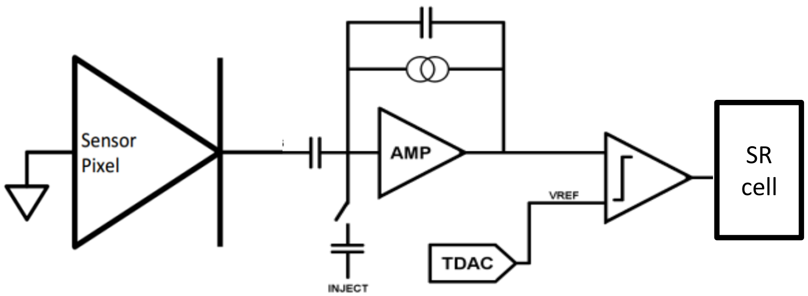


← **Electron beam setup**

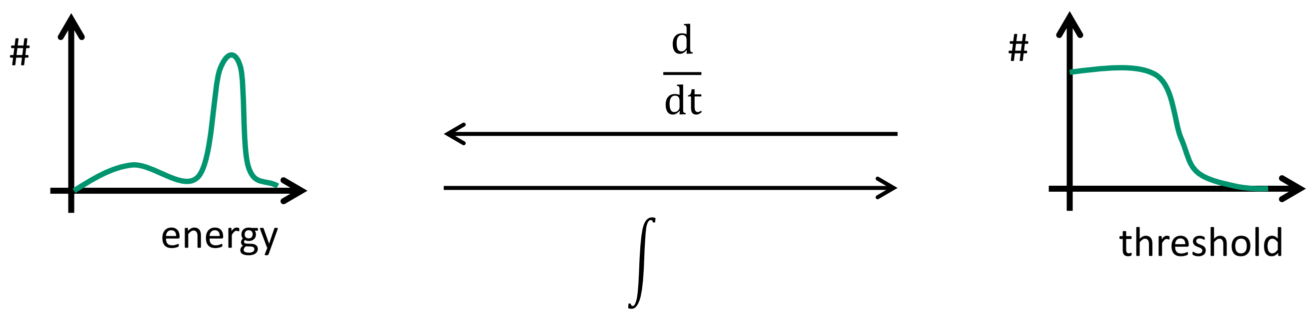
3.2 GeV electron beam

Expect $Q_{MIP}(MPV) = 3.3$ ke
in 50 µm silicon

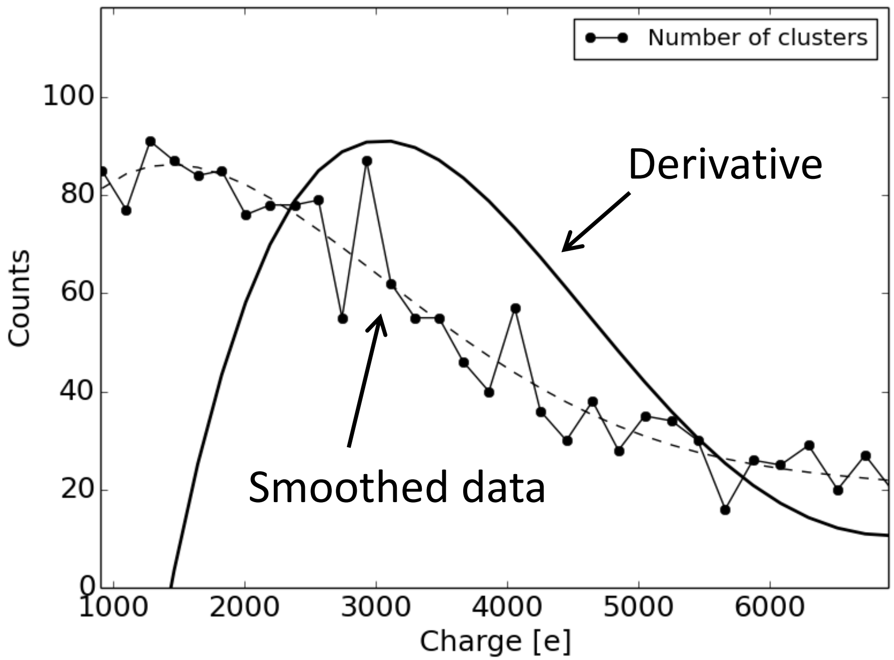
Energy spectrum from a binary output

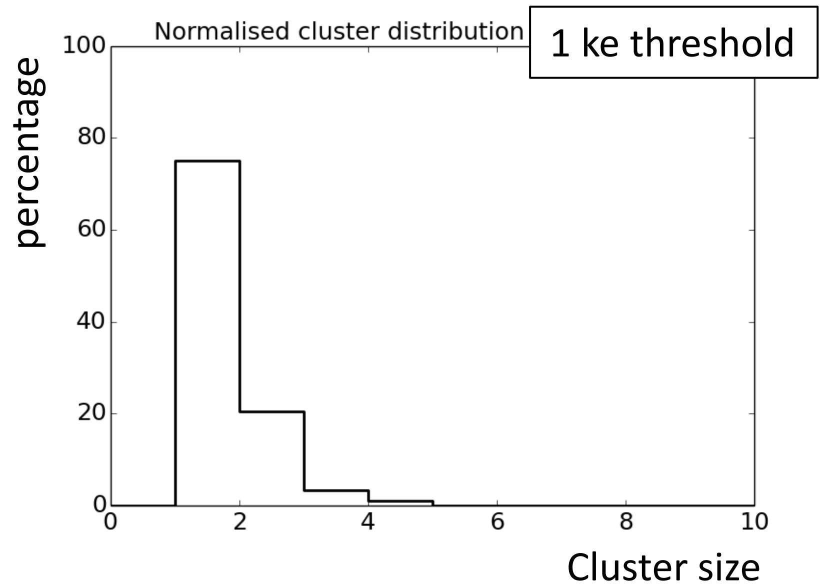


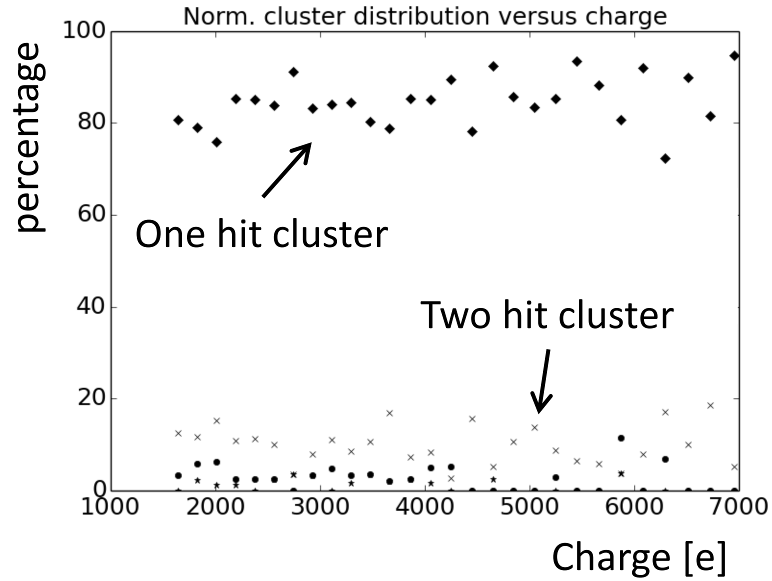
- Count cluster versus thresold voltage
- No trigger
- Analysis with 36 pixels
- Calibrate x-axis with internal injection, global calibration with cut at 7 ke



Results of Sr90 measurement



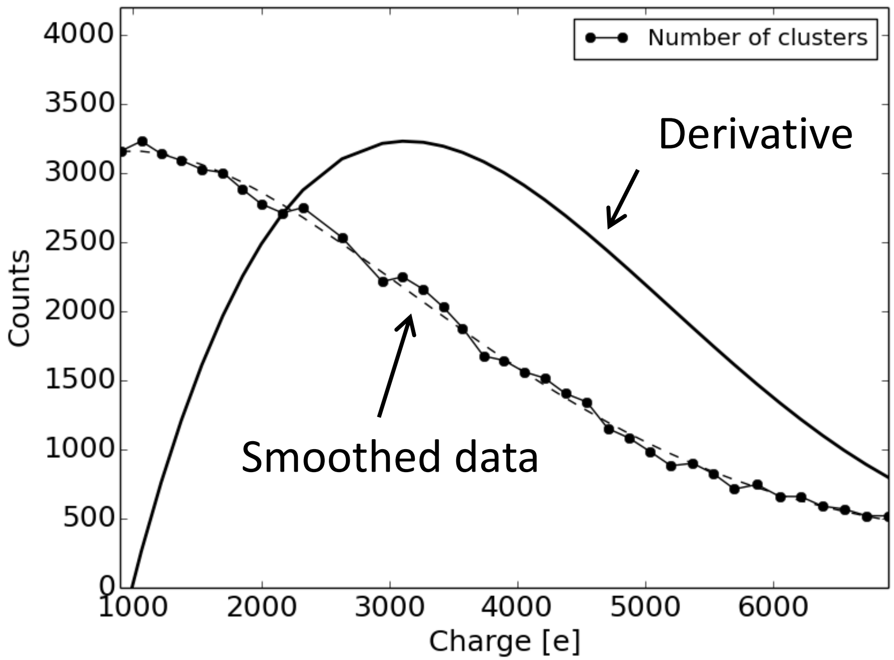


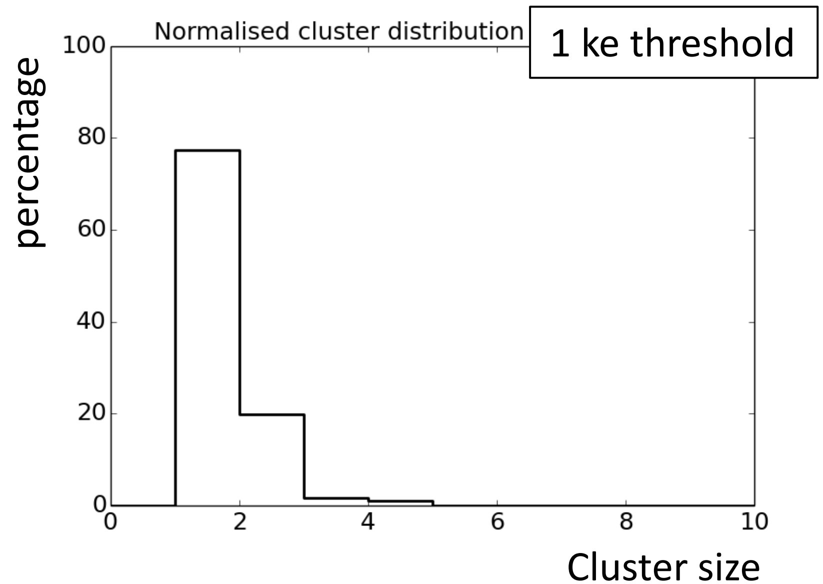


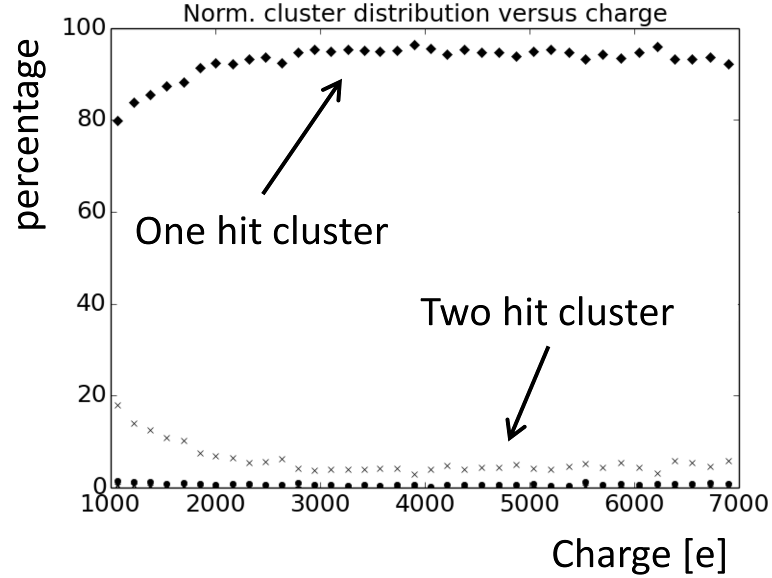
Reconstructed charge:

- 3.1 ke
- 15% less than expected (3.8 ke)

Results of 3.2 GeV electron beam (ELSA) measurement





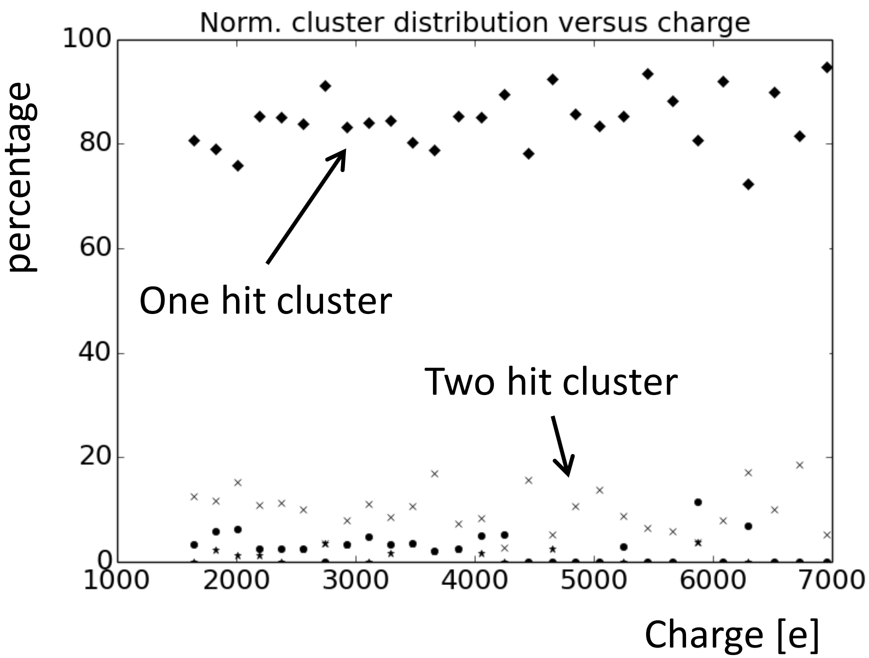


Reconstructed charge:

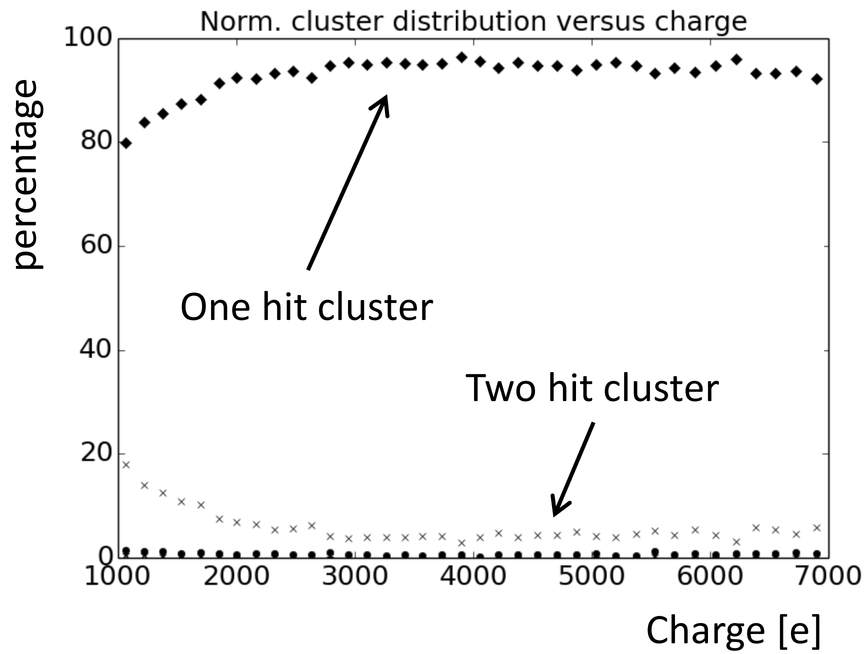
- 3.1 ke
- 7% less than expected (3.3 ke)
- At highest threshold rate is not zero (delta electrons)

Comparison of normalised cluster distributions Sr90/beam

Sr 90 collimated

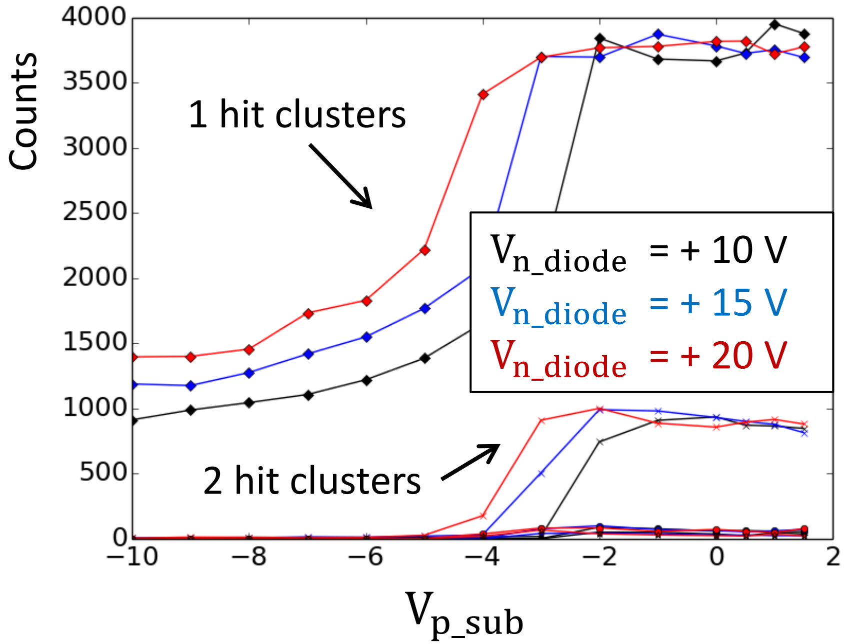


3.2 GeV electron beam

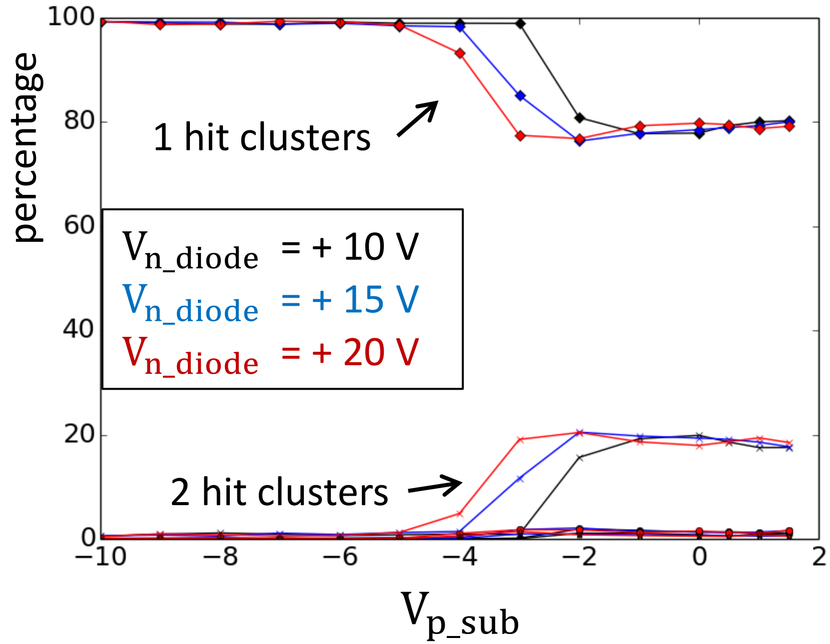


Parameter scan of V_{p_sub} and V_{n_diode} with electron beam

Absolute cluster distribution



Normalised cluster distribution

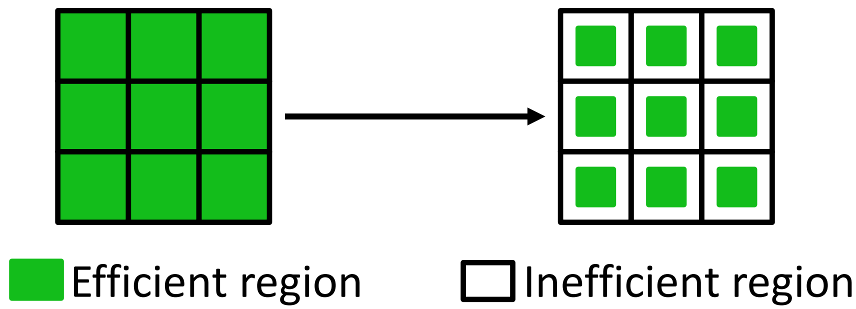


- At low V_{p_sub} the one hit cluster count rate drops to ~ ¼
- At low V_{p_sub} the cluster distribution changes from 80 % to ~ 100% one hit clusters
- Count rate decrease starts later with higher V_{n_diode}

Explanation of bias parameter scans

Observations:

- Count rate decreases to ¼ with decreasing V_{p_sub}
- Only one hit clusters at lower V_{p_sub}
- Count rate decrease starts ‚later' with increasing V_{n_diode}



→ Inefficient regions start to built up with decreasing V_{p_sub}

Fraction of count rate loss corresponds roughly to the fraction of n-implantation area to total area (~¼)

→ The inefficient regions are likely to be below the psubstrate.

Charge is partially lost.

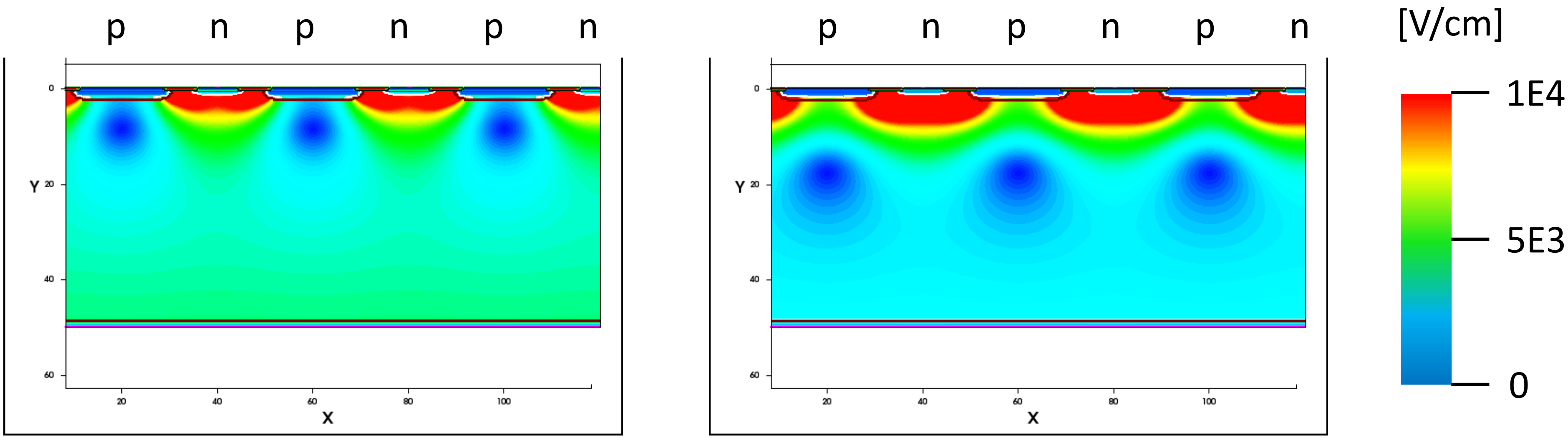
Possible reasons for charge loss:

- Trapping → Unlikely in unirradiated sensor
- Slower charge collection → Finite integration time of CSA of ~ 10 ns
- Competing electrode → Unlikely as psub implant is not attractive to electrons

→ **Next step:** TCAD Simulation (in progress)

TCAD simulation

V_{p_sub} = 0 V $\qquad$ V_{p_sub} = - 10 V



- Minimum of electric field shifts with lower V_{p_sub} towards backside
- Electric field is close to zero at the backside for V_{p_sub} = - 10 V

→ Effect on charge collection time is work in progress

Summary

Sensor characterization of the first DMAPS prototype in ESPROS technology has been performed

- Homoegeneity studies and bias parameter optimization with laser system
- Sr 90 and electron beam used for measurement of charge collection
- Biasing of the sensor is partially understood

Radiation hardness studies are work in progress

- Xray irradiation studies show transistors are radiation tolerant up to 50 Mrad TID
- Neutron irradiated samples arrived recently (up to 5E+14 neq)

New version with improved electronics submitted in Q2 2014, expected to arrive in Q4 2014

Thank you

References

[1] I. Peric and C Takacs, *Large monolithic particle pixel detectors in high-voltage CMOS technology. Nucl. Instrum. Methods A 624 (2010), 504-508*

[2] S. Senyukov et al., *Charged particle detection performances of CMOS pixel sensor produced in a 0.18 µm process with a high resistivity epitaxial layer. Nucl. Instrum. Methods A 730 (2013), 115-118.*

[3] R. Turchetta et al., *A monolithic active pixel sensor for charged particle tracking and imaging using standard VLSI CMOS technology. Nucl. Instrum. Methods A 458 (2001), 677-689.*

[4] M. Havranek et al., *DMAPS: a fully depleted monolithic active pixel sensor – analog performance,* 2014 *JINST* **1** P0514.

Related papers:

[5] ATLAS Collaboration. *ATLAS pixel detector electronics and sensors. JINST 3 (2008)*

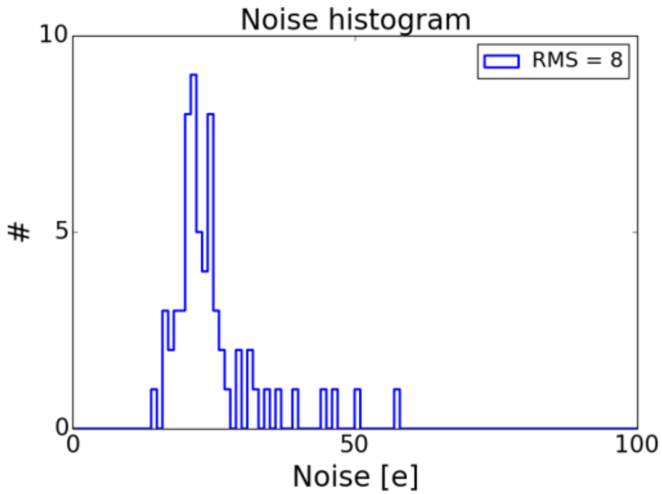
[6] Y. Mikami, *Monolithic Active Pixel Sensors (MAPS) in a Quadruple Well Technology for Nearly 100% Fill Factor and Full CMOS Pixels.* Sensors **8** (2008), 5336-5351.

[7] H. Bichsel, *Straggling in thin silicon detectors,* Reviews of Modern Physics, Vol. 60, No. 3, July 1988.

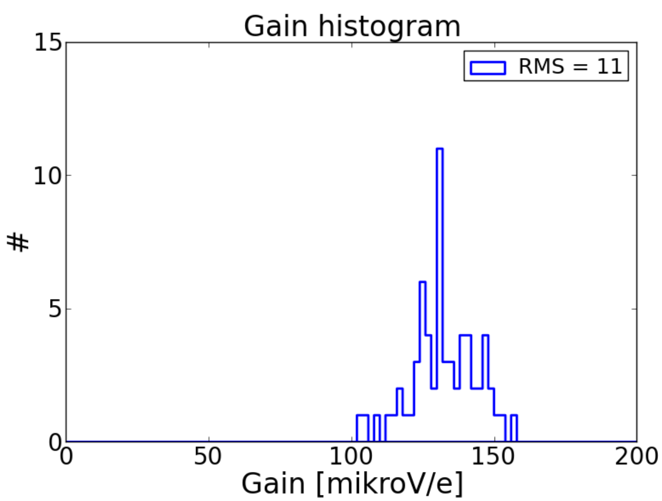
[8] Pohl et al., *A method for precise single pixel charge reconstruction for counting pixel detectors*, To be submitted

Backup

Front end performance



→ Noise ~ 25 e



→ Gain ~ 125 µV/e

EPCB02

- **EPCB02 is evolution of EPCB01**

  - cascode amplifier with high open loop gain

  - added 2 fF feedback capacitor -> uniformity

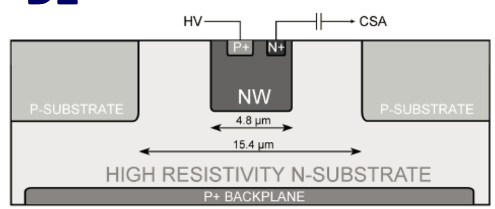
  - both charge collection electrodes are custom

  - input transistors are larger

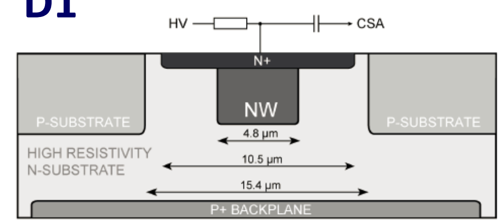
  - each pixel has analog output

  (multiplexed → unity gain buffer → IO pad)

D2



D1



- **EPCB02 will allow to study independently:**

  V1 vs V2 – effects of biasing (RB/DB)

  V1 vs V4 – cont. vs switched FE

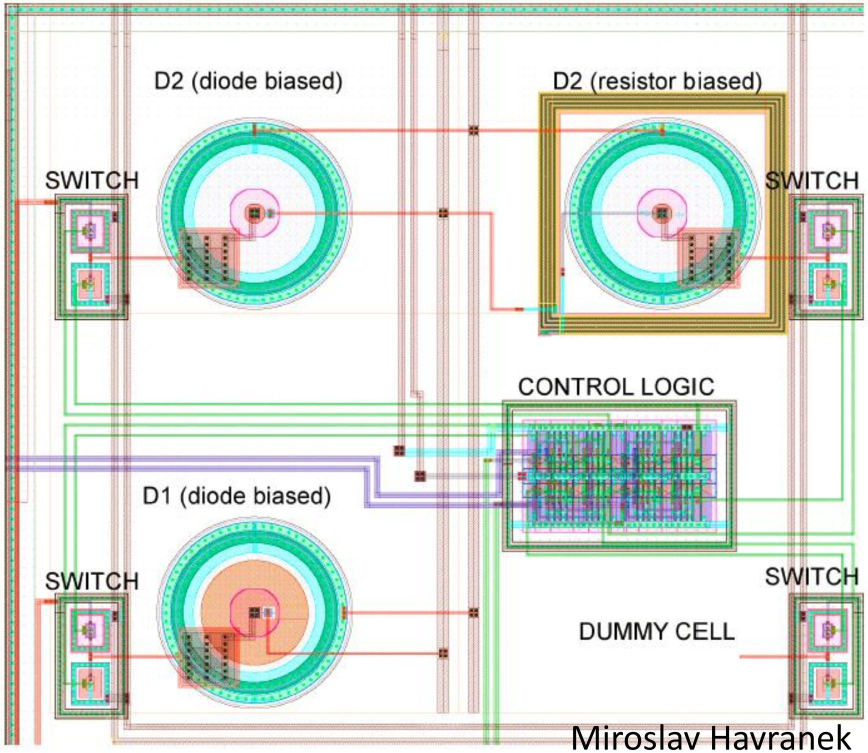
  V4 vs V5 – different sensor geom.

  V4 vs V6 – different transistor size

Variant	Sensor	Biasing	FE-type	M1 dim.
V1	D2	DB	CONT	1μ/300n
V2	D2	RB	CONT	1μ/300n
V3	D1	DC	CONT	1μ/300n
V4	D2	DB	SW	1μ/300n
V5	D1	DB	SW	1μ/300n
V6	D2	DB	SW	2μ/150n

Circuit for capacitance measurement in EPCB02

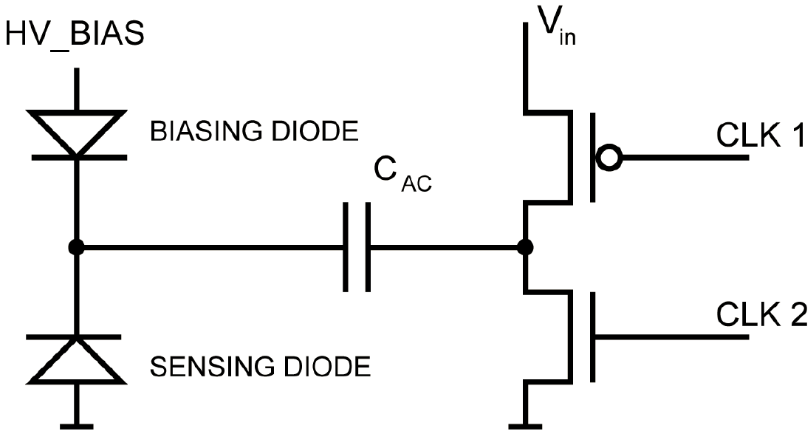
- Various layouts and biasing circuits can influence sensor capacitance
- C_d is important parameter for rise-time and noise optimization
- Charge-pump based circuit for capacitance measurement of DMAPS sensor



- **Two transistor charge pump:**

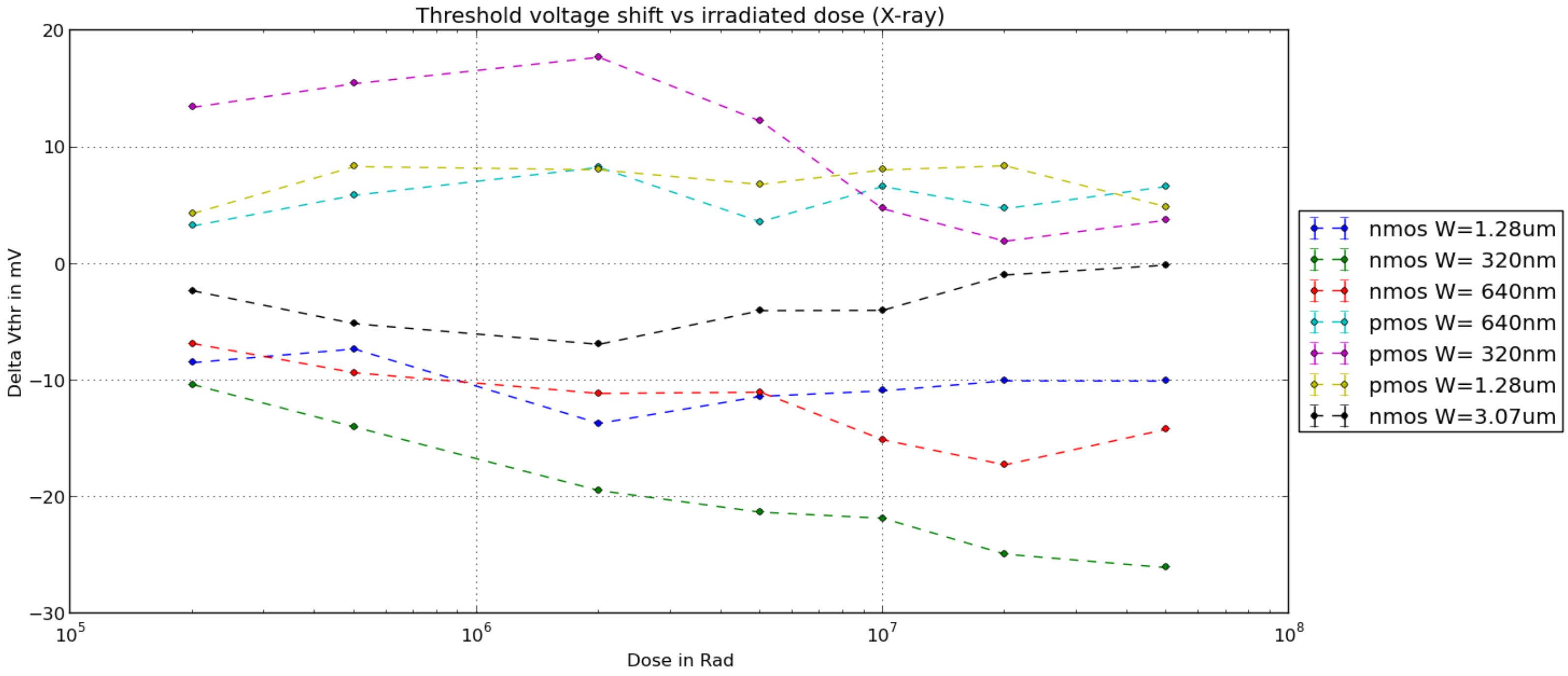
  - CV measurement => depl. voltage.

  - compare different geometries



X-Ray Irradiation Test Result

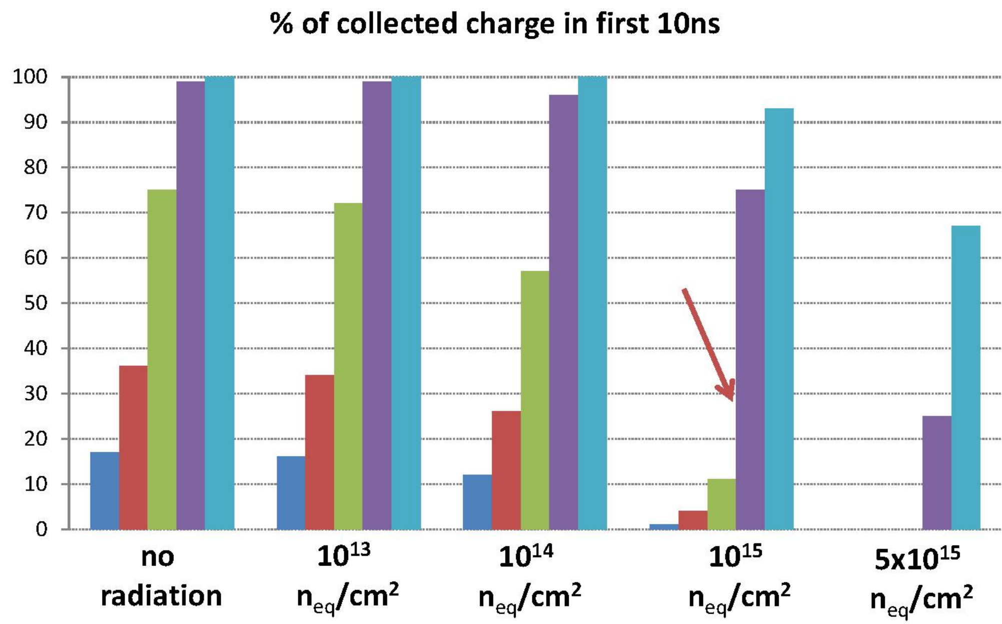
- Irradiation with X-Ray (60 keV) and Vanadium filter (peak ~ 5keV) up to 50 Mrad



- **Less than 25 mV shift after 50 Mrad of Total Ionizing Dose (TID)**

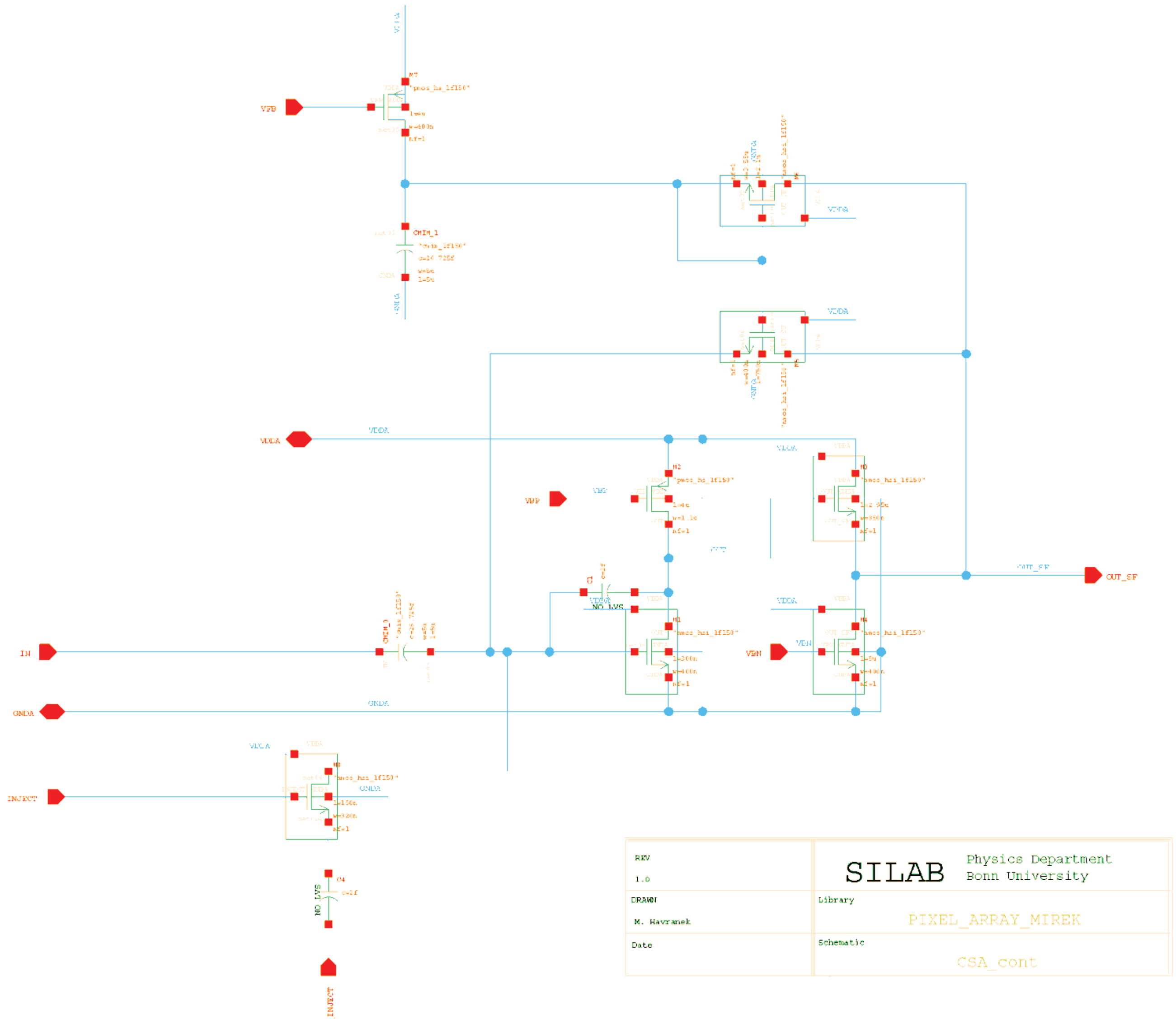
- Enclosed layout has lower threshold shift and recovers at 50 MRad

TCAD simulation of CC Time in dependence of fill factor and bias voltage and resistivity



	substrate resistivity [Ohm cm]	Bias [V]	Fill Factor [%]
(blue)	10	1	15
(red)	10	20	15
(green)	2k	1	15
(purple)	2k	20	15
(cyan)	2k	20	75

Transistor level schematic of CSA



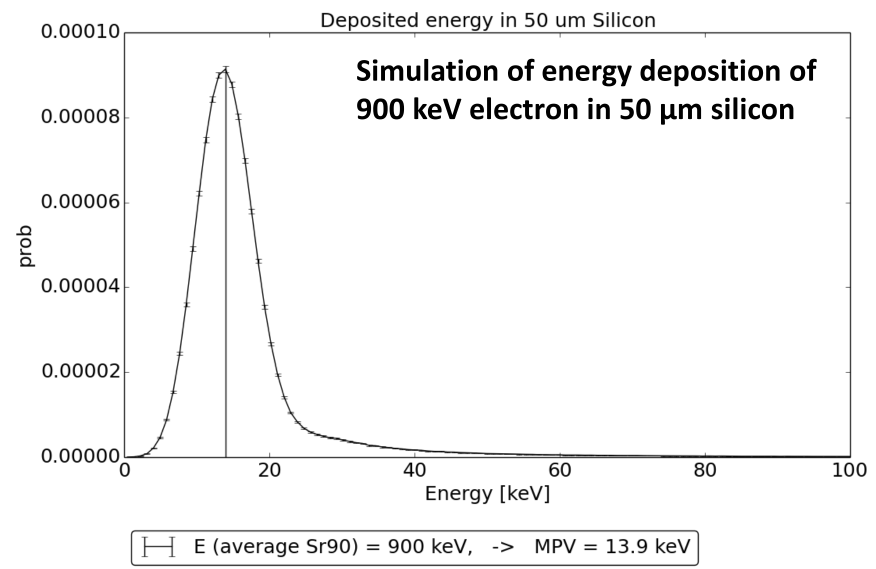
Expectation of energy deposition in 50 µm Si

Literature:

- The most probable value of energy deposition (MPV) for a **MIP** in 50 µm silicon is: MPV = 12 keV [1]
- → $Q_{MIP}(MPV) = 3.34\ ke$

Simulation of MPV for Sr90 source with pyPENELOPE for average energy of emitted energy spectrum Sr90 (900 keV) [2]:
MPV = 13.9 keV

- $Q_{Sr90}(MPV) = 3.86\ ke$



- $Q_{generated,\ MIP}(MPV) < Q_{generated,Sr90}(MPV)$

[1] Bichsel, Straggling in thin silicon detectors, Reviews of Modern Physics 60 (1988) 663-699 (page: 685 and 691)
[2] Sempau, Monte Carlo simulation of electron beams from an accelerator head using PENELOPE, Phys. Med. Biol. 46 (2001) 1163–1186

DMAPS in ATLAS

Inner layer

1. **Low power**
2. **Low material**
3. Occupancy
4. Resolution

↓

hybrid pixels (65nm? + sensor?)

Outer layer

1. **Low cost**
2. **Low power**
3. Low material
4. Resolution

↓

low cost hybrid pixels or monolithic?

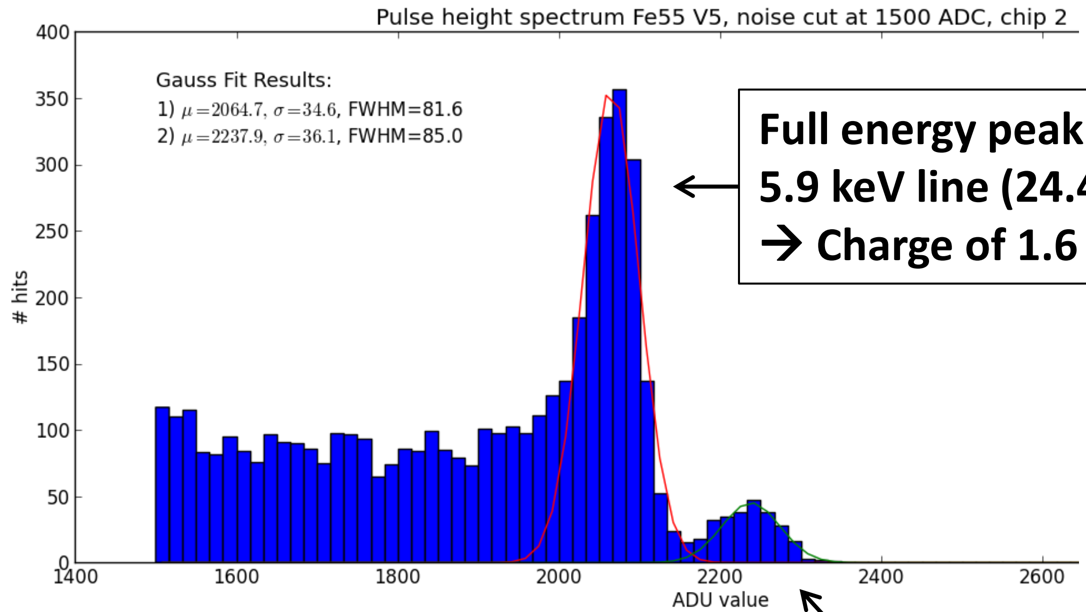
Single Pixel Spectrum with Fe55 –V5

Method:

- Setup:

Illumination with Fe55 from the backside

- Data taking for 10 hours
- Histogram of signal
- Cut at 1500 to reduce noise



Full energy peak: 5.9 keV line (24.4%) → Charge of 1.6 ke

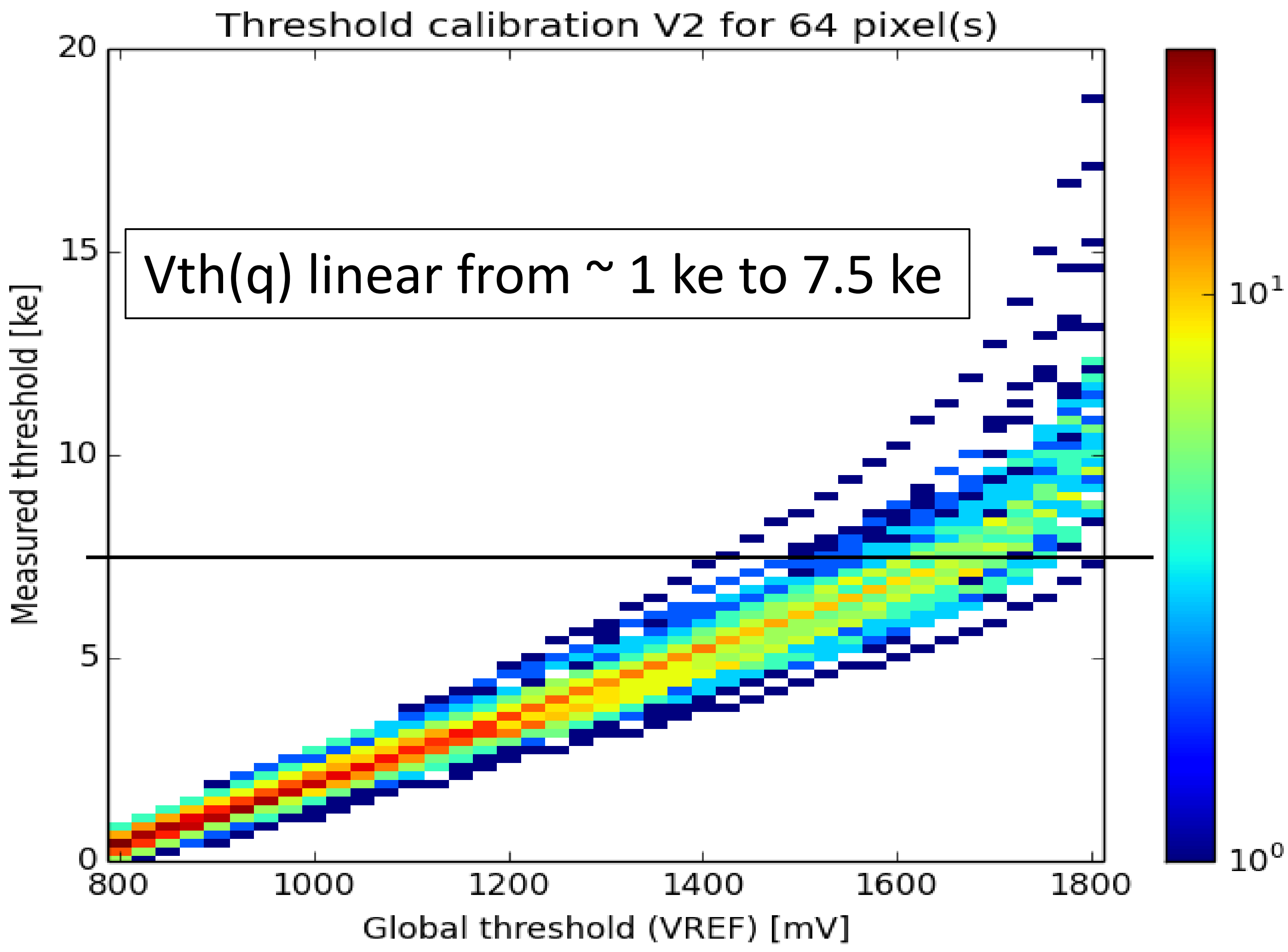
Full energy peak: 6.5 keV line (2,85%) → Charge of 1.8 ke

→ Both γ lines from Fe55 clearly visible in spectrum

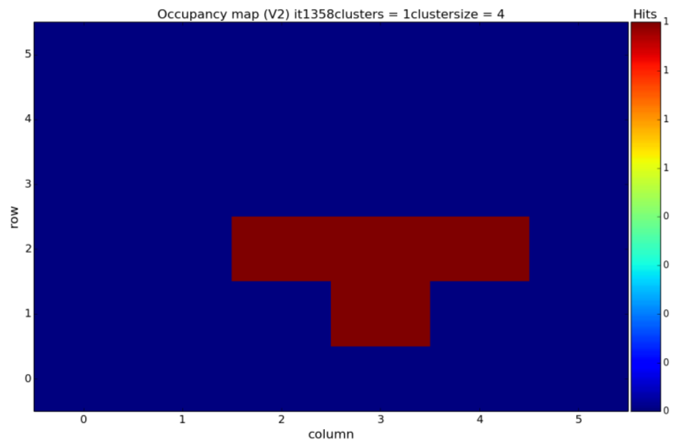
→ Sigma ~ 36 ADU (FWHM ~ 85 ADU)

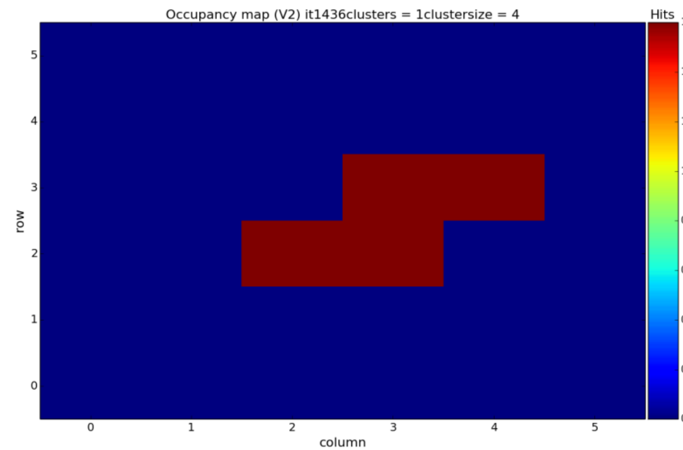
→ ~ 1.1 ADU / electron

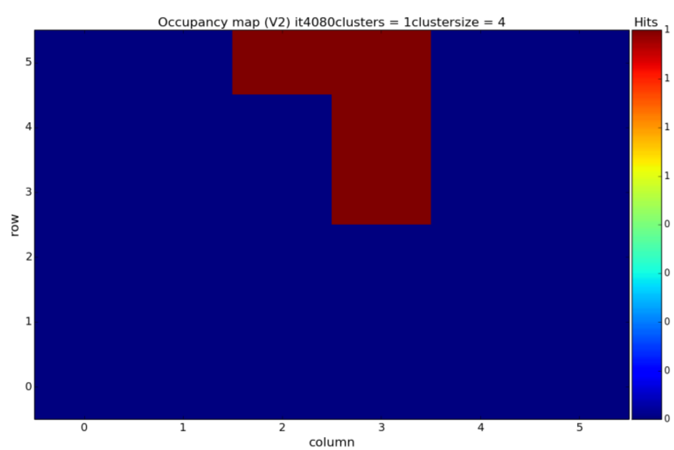
Threshold calibration electron beam and source data

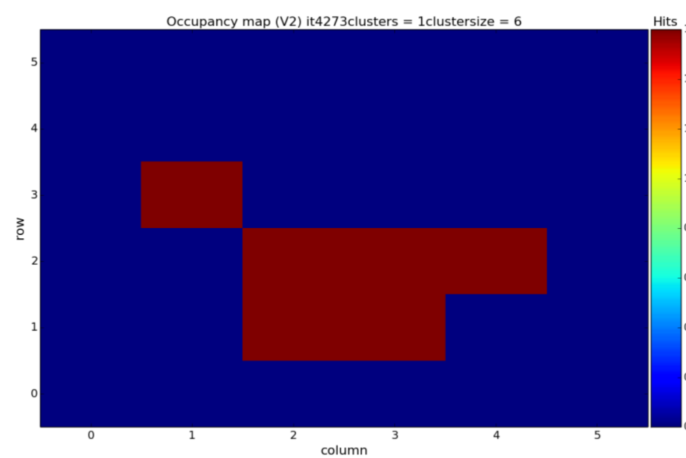


Exotic cluster examples



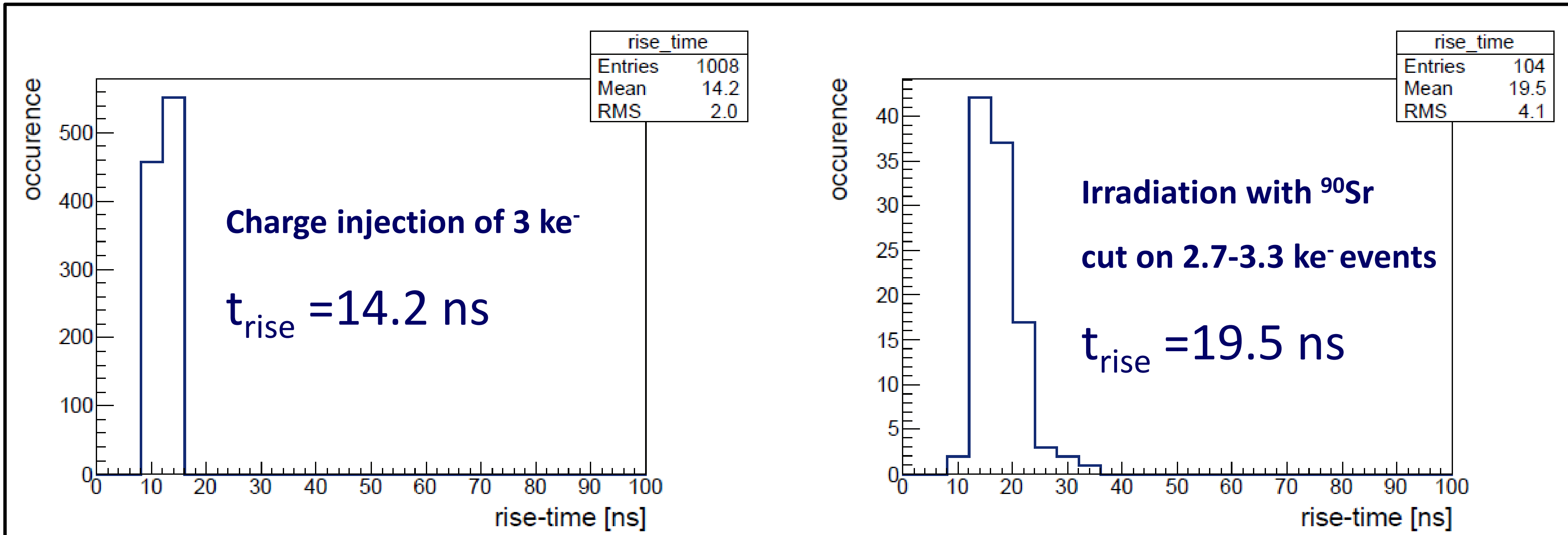






Signal rise-time

- Rise-time => 25% - 75% signal amplitude
- Upper limit on charge collection time



- Fast charge collection
- 50 % of the signal charge is collected for less than 19.5 ns

Effect of charge sharing and multiple scattering on the reconstructed charge

Multiple scattering → Due to higher energy → higher fraction of one hit clusters with beam as compared to Sr90

Charge sharing → Due to smallest angular spread → higher fraction of one hit clusters with beam

Effects on reconstructed charge:

Hit detection only if $\mathbf{V_q > V_{thr}}$

→ The bigger clusters have a higher fraction of not detected charge which is below threshold (more charge is not detected)

→ The reconstructed charge is expected to be less than the prediction for both sources

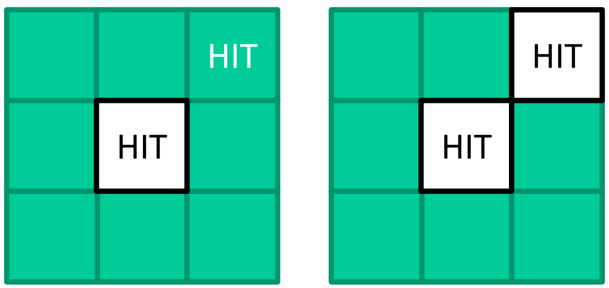
→ The effect is expected to be more pronounced for the Sr 90 measurement

Cluster algorithm

universitätbonn

1. Start with a cluster of size one.

2. Check if neighbours to cluster of one pixel distance are hit
   - Yes: add them to cluster → 2.
   - No: nothing → 3.



3. Store cluster and remove all clustered hits from hitmap

4. Start with 1.) until no hits are left

```
def clustering(data=None, pdf_name=None):
    """ Cluster data and return clusterized data with one pixel per cluster """
    """ data is of shape [frame_number][x_coord][y_coord] """
    """ pdf_name is the full path to the pdf output file """

    '''  Make copy of data for plotting '''
    data_raw = np.copy(data)

    print 'Shape of data to be clustered = ', data.shape
    '''  clusters is a one dimensional array and holds the number of clusters versus clustersize '''
    clusters =np.zeros(shape=(64))
    '''  cluster_pixel_data_for_thr_calib is output array containing clustered frames '''
    cluster_pixel_data_for_thr_calib = np.zeros(shape=(data.shape[0], data.shape[1], data.shape[2]))

    with PdfPages(pdf_name + '.pdf') as pdf_name:

        '''  Loop over all frames '''
        for it in range(0,data.shape[0]):
            print 'Frame it ............................................................ ', it, '/', data.shape[0]
            print 'Frame[it] = \n', data[it]
            coords = np.where(data[it][:][:] == 1)
            x = coords[0]
            y = coords[1]
            cluster_count = 0
            print 'x = ', x
            print 'y = ', y

            '''  Start to search for all clusters within one frame '''
            ''' Go over all hit x coordintes, make cluster and remove these from the hit x coordintes and then start over until nothing left '''
            while len(x)>0:

                cluster_count += 1
                '''  Define the first entry of the array holding all x coordinates of hit pixels to be the start for clustersearch '''
                x_y_cluster = [(x[0],y[0])]
                print 'Start values of cluster ', cluster_count ,' = ', x_y_cluster

                keep_searching = True

                iti = 0
                while keep_searching is True:
                    print 'Number of hits to be clustered = ', len(x)
                    iti += 1
                    start_len = len(x_y_cluster)
                    print 'Iterator of while loop = \t', iti ,'\t Clustersize of current cluster candidate = \t', start_len
                    for c_it in range(0,len(x_y_cluster)):
#                        print 'c_it = ', c_it
                        for x_it in range(0,len(x)):
                            x_diff_cluster_member = x_y_cluster[c_it][0]-x[x_it]
                            y_diff_cluster_member = x_y_cluster[c_it][1]-y[x_it]
                            if (x_diff_cluster_member == 0 or x_diff_cluster_member == 1 or x_diff_cluster_member == -1) and (y_diff_cluster_member == 0 or
y_diff_cluster_member == 1 or y_diff_cluster_member == -1):
                                # Here all neighbours are found, also those which already belong to cluster
                                x_y_cluster.append((x[x_it],y[x_it]))

                    x_y_cluster = sorted(set(x_y_cluster)) # remove duplicates from cluster data
                    print 'Iterator of while loop = \t', iti ,'\t Clustersize after search = \t', len(x_y_cluster)
                    if start_len == len(x_y_cluster):
                        keep_searching=False
                    if len(x_y_cluster) == len(x):# means all hits of this frame belong to only one cluster
                        keep_searching=False
                    print 'Keep searching bool = ', keep_searching

                print 'Finished a cluster with coordinates = ', x_y_cluster

                '''  Cluster finished, store it '''
                clustersize = len(x_y_cluster)
                clusters[clustersize] = clusters[clustersize] + 1
                cluster_pixel_data_for_thr_calib[it][x[0]][y[0]]+=1

                print 'Clustered frame[it] = \n', cluster_pixel_data_for_thr_calib[it]

                '''  Remove clustered hits from data '''
                for hit in range(0,len(x_y_cluster)):
                    data[it][x_y_cluster[hit][0]][x_y_cluster[hit][1]]=0
#                 print 'Mod hitmap after reduction of hits from ', cluster_count , ' clusters = ', data[it]

                '''  Make new x and y hit coordinates for next cluster '''
                print '... continue with next cluster'
                coords = np.where(data[it][:][:] == 1)
                x = coords[0]
                y = coords[1]
                print 'next x', x
                print 'next y', y

                if clustersize > 3:
                    plotting.plot_2D_hitmap(hist2d=data_raw[it], title='Occupancy map (V2) it' + str(it) + 'clusters = ' + str(cluster_count)+ 'clustersize =
' + str(clustersize), zlabel='Hits', z_max=None, filename=pdf_name, savename='test')

            print 'Found clusters: ', cluster_count
            if cluster_count > 1:
                plotting.plot_2D_hitmap(hist2d=data_raw[it], title='Occupancy map (V2) it' + str(it) + 'clusters = ' + str(cluster_count), zlabel='Hits',
z_max=None, filename=pdf_name, savename='test')

        plot_cluster_distribution(data=clusters, title='Cluster distribution 6x6 V2', filename=pdf_name, savename='test')
    print 'Clustering done.'
```

Charge sharing (geometrical effects)

Charge cloud spreads laterally due to diffusion and therefore charge can be shared between neighbouring pixels → clusters

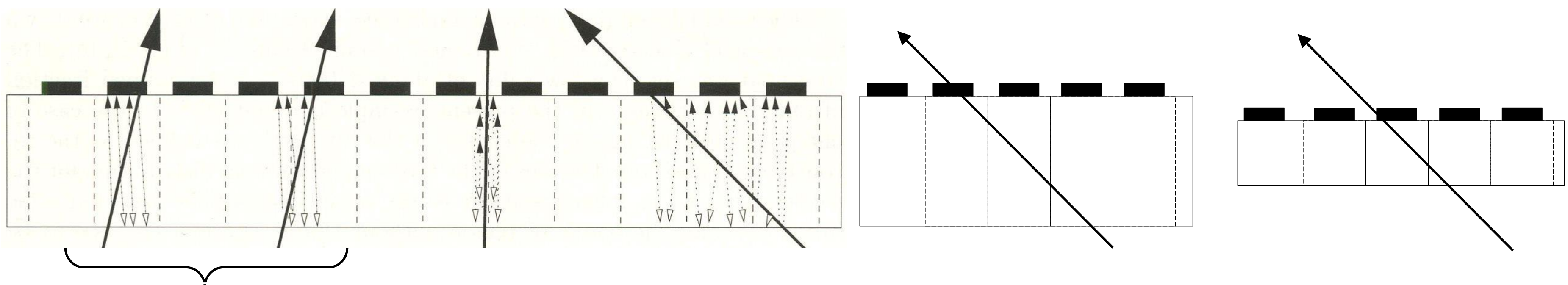
Charge sharing is geometrically affected by:

- Position of track with respect to sensor surface
- Geometry of pixel cell
- Angle of track with respect to sensor surface

Electrons:

Holes:

Incident particle:



Same angle but different positions
→ Left: Clustersize: 1
→ Right: Clustersize: 2

Track exactly on pixel edge
Clustersize: 2

Clustersize: 3

Clustersize: 3

Clustersize: 2

Effects of charge sharing on cluster size distribution

- Position of track with respect to sensor surface → Not known, same for Sr90 and electron beam
- Geometry of pixel cell → almost cubic, 40 µm x 40 µm x 50 µm, fixed
- **Angle** of track with respect to sensor surface → Different for Sr 90 and electron beam

→ The bigger the angular spread, the higher the fraction of bigger clusters.

→ The smaller the angular spread, the higher the fraction of one hit clusters.

Angular spread of Sr 90 > Angular spread Sr 90 collimated > focussed electron beam

Expectation:

→ Highest fraction of one hit clusters for electron beam measurement

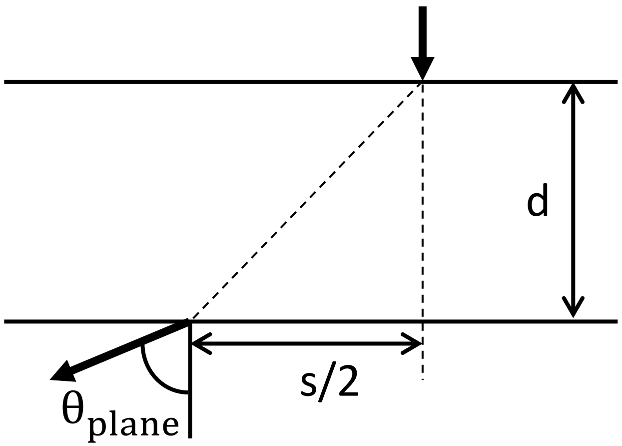
→ Smallest fraction of one hit clusters with uncollimated Sr 90 source

Effect of multiple scattering on the cluster size

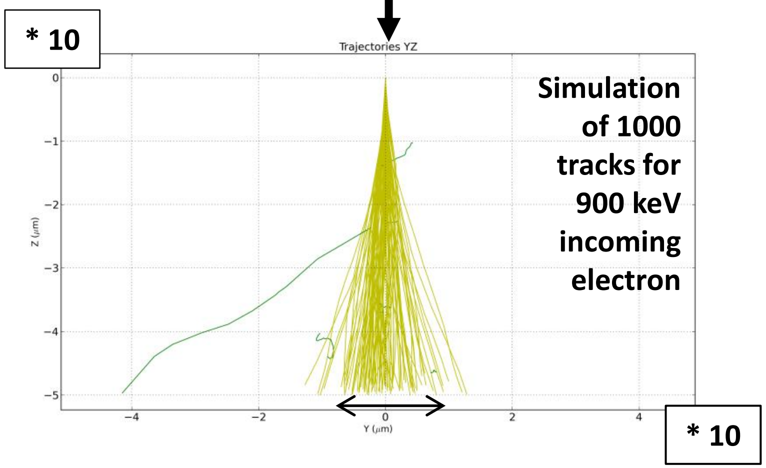
Multiple scattering

Deflection of particle in matter by coulomb scattering, can be calculated:

- $\theta_{plane} = \frac{13.6}{\beta p[MeV/c]} z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \ln \frac{x}{X_0}\right)$
- θ_{plane} = sigma of Gauss fit to central 98% of the scattering angles [3]
- $s/2 = d \cdot \tan\theta_{plane} \approx d \cdot \theta_{plane}$; d = 50 µm
- s(1 sigma) **= 20 µm for Sr 90 (2.28 MeV)**
- **s = 0.003 µm for 3.2 GeV electron beam**



[3] Bichsel, Groom, Klein, PDG Review 2010, p 290



Simulation of electron trajectories for 2.28 MeV incident particle agrees with calculation

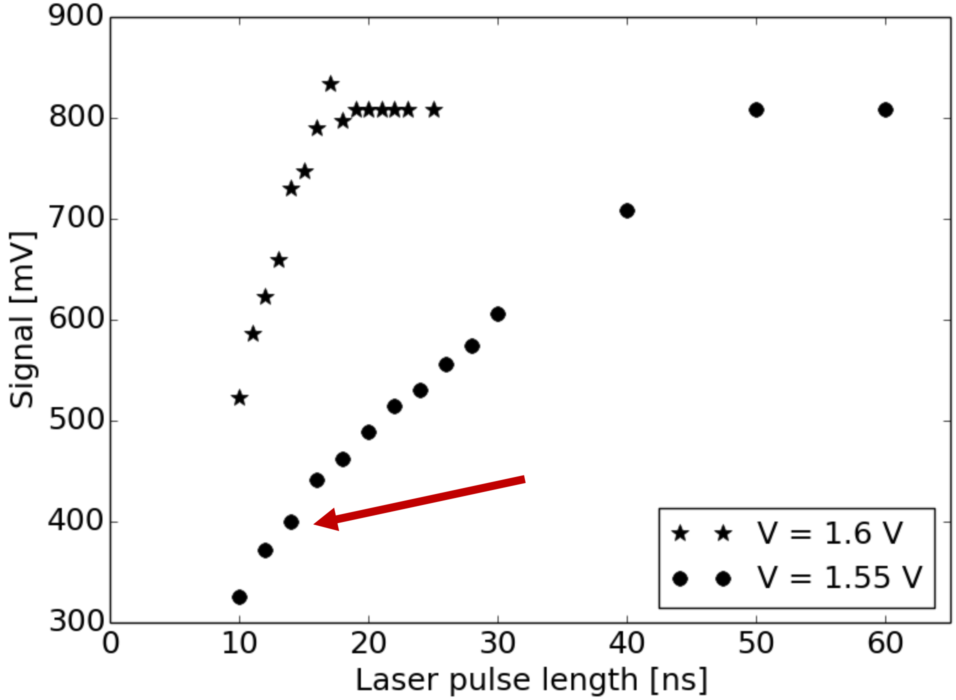
→ Expectation: Higher fraction of one hit clusters with electron beam measurement as compared to Sr 90 measurement

- β, p, z properties incoming particle
- $\frac{x}{X_0}$ traveled distance in units of radiation length

LASER measurements

Laser system linearity

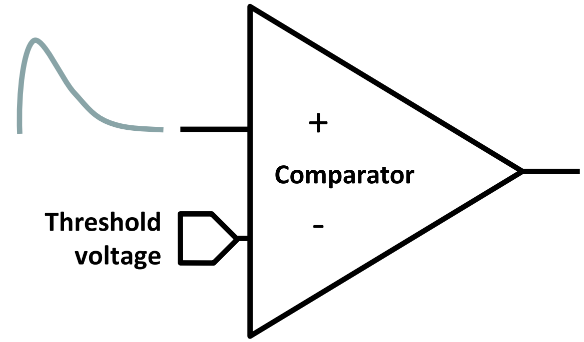
- Injection of laser can be configured by voltage and laser pulse width
- Linearity measured with complete system (Laser + EPCB01)



→ Typical setting: 14 ns, 1.55 V (linear region)

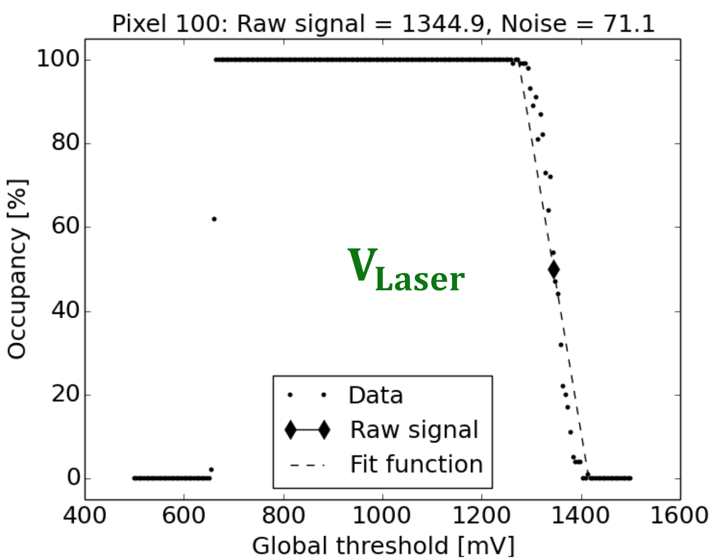
Method DAQ Laser scans

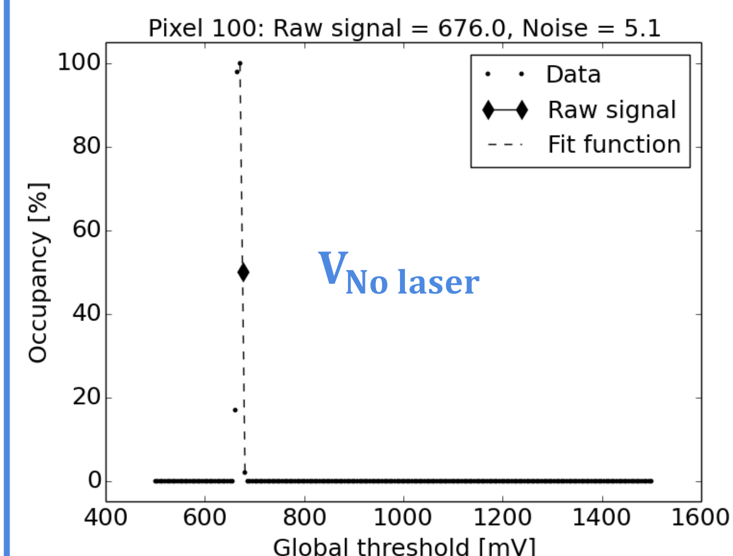
- Move laser over matrix and measure the signal



Signal:

- Threshold scan with laser as injection
- Scan range [500,1500] mV to cover signal and baseline
- V_{Laser} and σ_{Laser} determined from line fit





Noise cut:

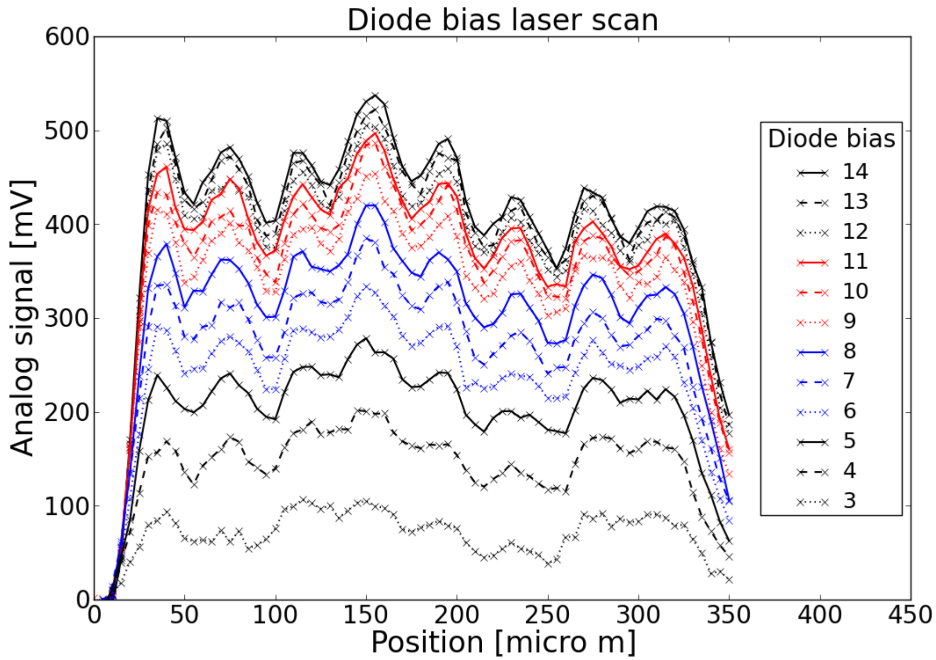
$V_{Signal} > V_{Cut}$

$V_{Cut} = 3 \cdot \sigma_{No\ laser}$

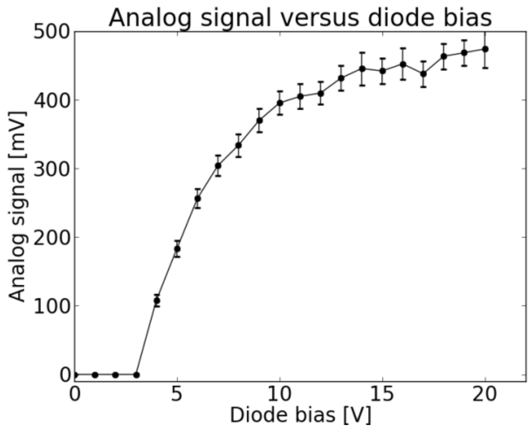
$$V_{Signal}[mV] = (V_{Laser}\,[mV] - V_{No\ laser}\,[mV]) \longrightarrow V_{Signal}[mV] = 669\text{ mV}$$

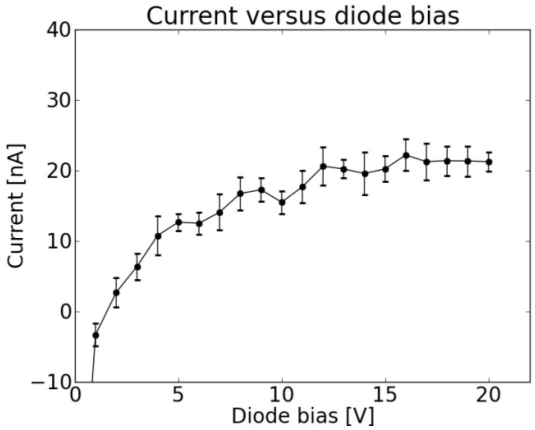
Diode bias scan with laser – V2

- Scan the same column with the laser for different diode bias settings



For one pixel and fixed laser position:



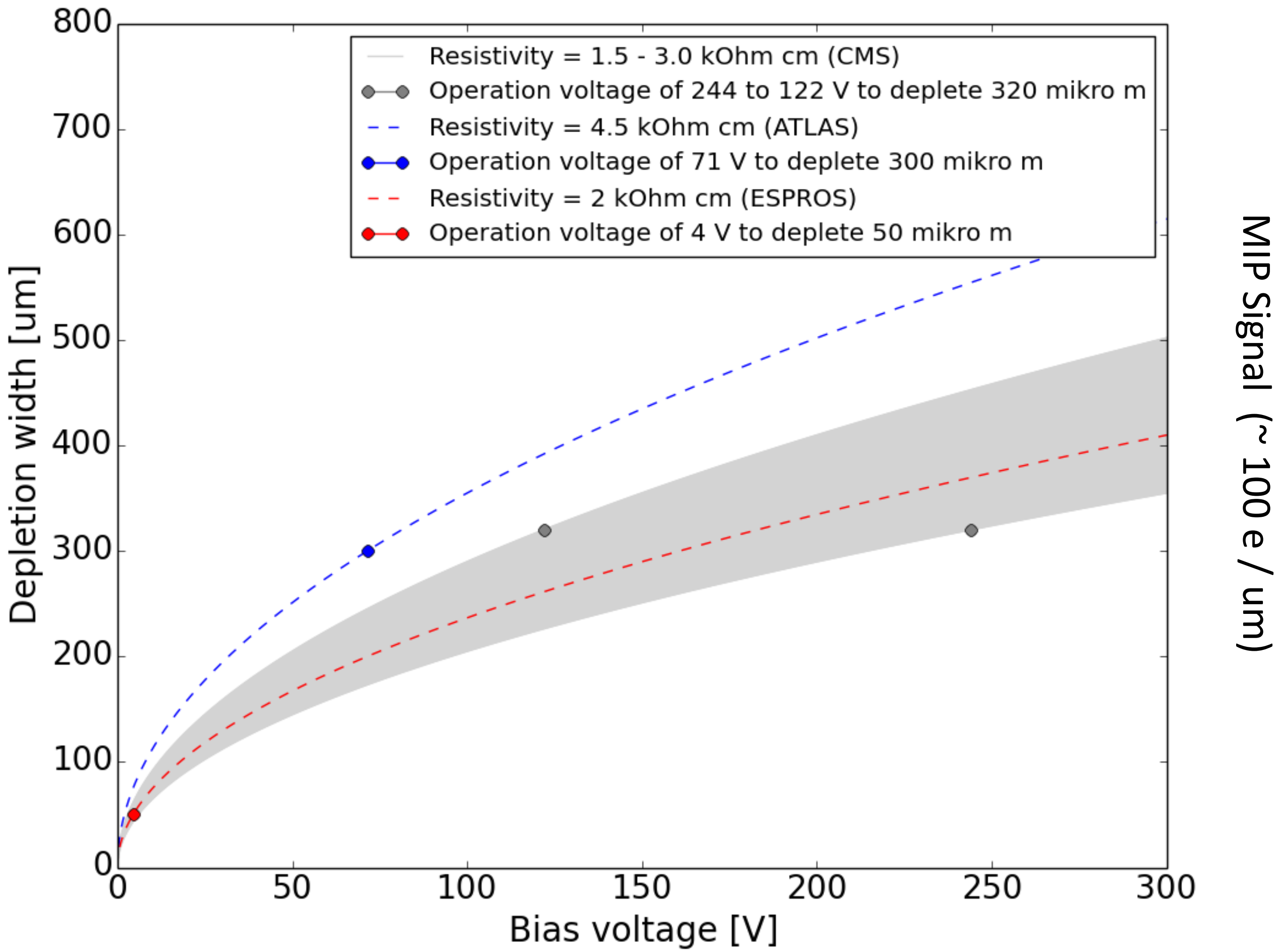


→ Increasing diode bias voltage results in higher signals, as expected

→ Saturation of signal above 10 V (→ depleted sensor)

→ Leakage current ~ 20 nA

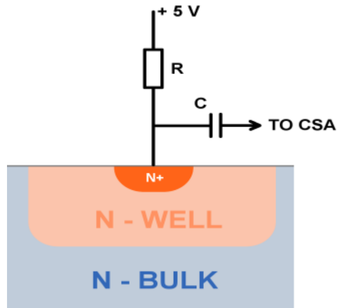
Depletion widths for different resistivities



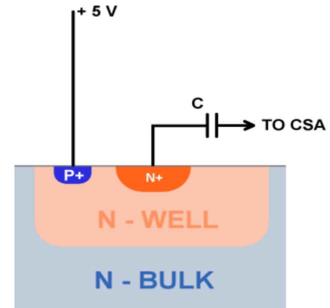
EPCB01 - Details

- Bias options:

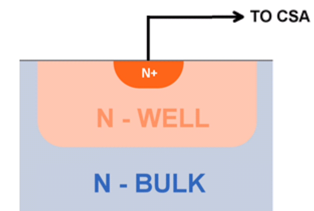
AC coupling
Resistor bias



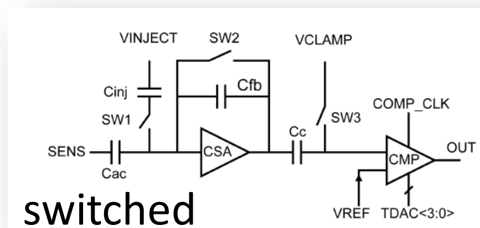
AC coupling
Diode bias

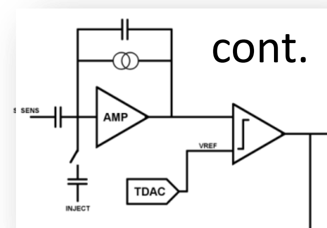


Direct coupling
Direct bias



- Pixel electronics options:



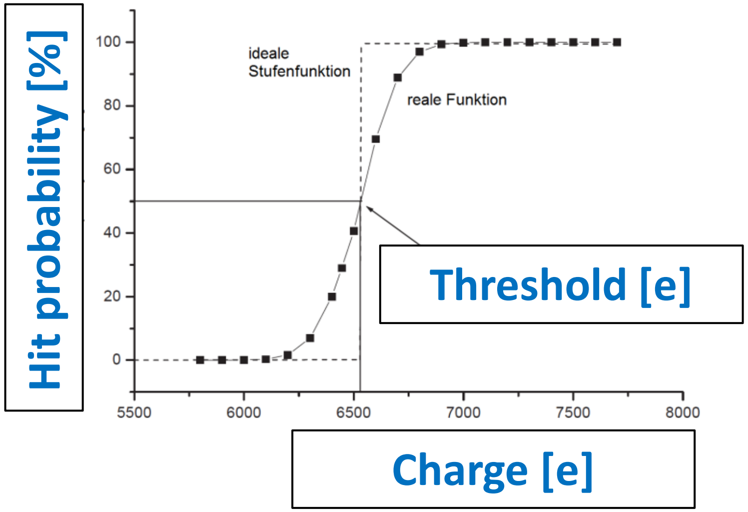


- Matrix versions:

Matrix version	Biasing and coupling	Analog FE
V1	Resistor + AC	Continuous
V2	Diode + AC	Continuous
V3	Direct + DC	Continuous
V4	Direct + DC	Switched
V5	Diode + AC	Switched
V6	Resistor + AC	Switched

Lab test results:
→ FE characterization of all matrices
→ Sensor characterization of diode bias only

Scurve



Threshold

- Injection of different charges
- Plot hits versus charge injection
- Ideal case: step function
- Noise → Scurve (step + gauss)
- Threshold: x – value of fit function at y = 50 %
- $P_{\text{Treffer}}(Q) = \Theta(Q - Q_{\text{Schwelle}}) \cdot \exp\left(-\frac{Q^2}{2\sigma^2_{\text{Rauschen}}}\right)$

Amplitude

- Injection of the same charge
- Variation of the threshold
- Plot hits versus threshold
- Ideal case: step function
- Noise → Scurve (step + gauss)
- Amplitude: x – value of fit function at y = 50 %
- $P_{\text{Treffer}}(Q) = \Theta(Q - Q_{\text{Schwelle}}) \cdot \exp\left(-\frac{Q^2}{2\sigma^2_{\text{Rauschen}}}\right)$