Karlsruhe Institute of Technology

Beam position monitoring with a 5x5 HitPix3 matrix for ion beam therapy

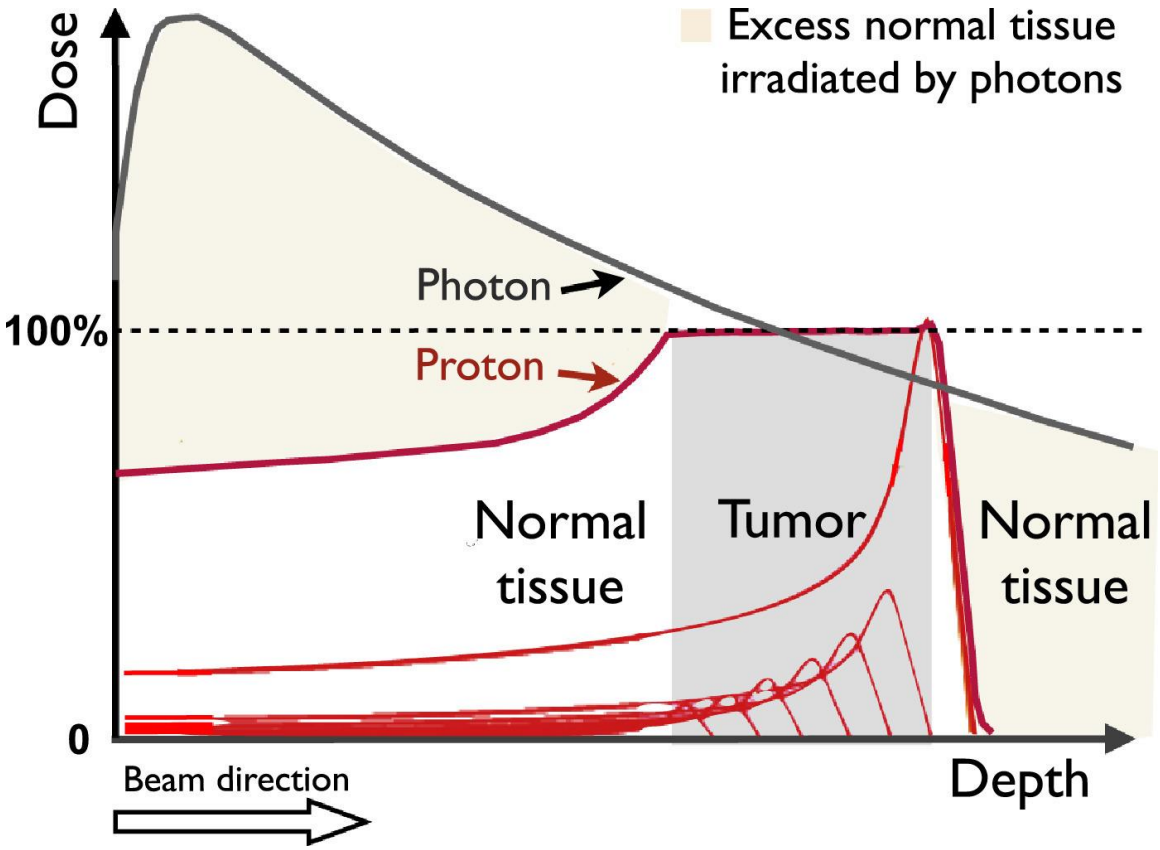
Master thesis by Baudry Bartels

28.07.2025

Supervisor: Professor Markus Klute
Advisors: Bogdan Topko, Alexander Dierlamm

Ion beam therapy as cancer treatment

- Goal: kill tumor cells with precise particle energy deposition
- Why Ion beams?
  - Precise energy deposition in depth due to Bragg peak
    - ➢ less damage to healthy tissue
- Scanning of position and energy of beam to cover full tumor volume (Raster scan)
- Beam monitoring essential to ensure precise execution



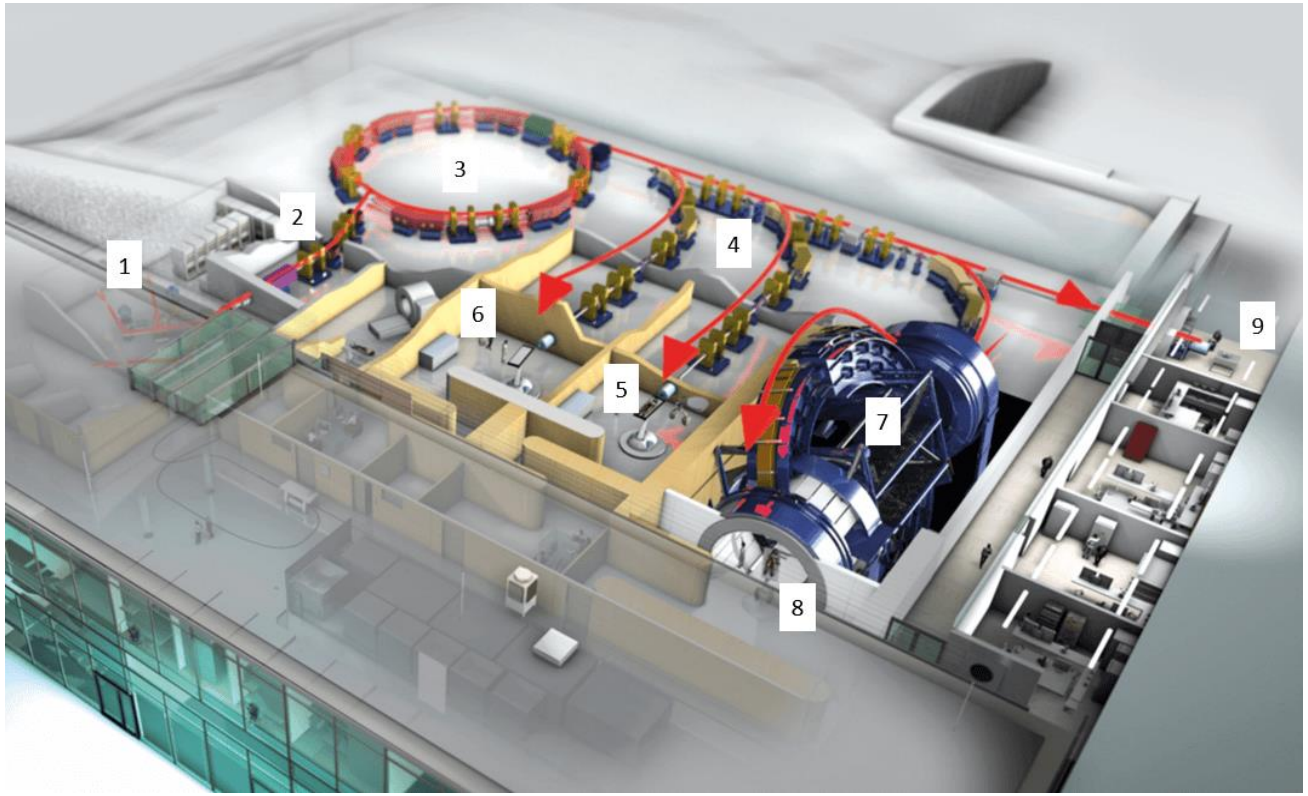
Energy deposition as a function of penetration depth for photons and protons
Reference: https://doi.org/10.1016/j.urolonc.2018.11.012

System at Heidelberg ion beam therapy center (HIT)

- Treatment plans with proton and carbon-ion beams
- Adjustable beam parameters:

parameter	steps	Proton	Carbon
Energy [MeV/u]	255	48 - 221	88 - 430
Width (FWHM) [mm]	4	8 - 20	4 - 12
Intensity [1/s]	10	$8 \cdot 10^7$ - $2 \cdot 10^9$	$2 \cdot 10^6$ - $8 \cdot 10^7$

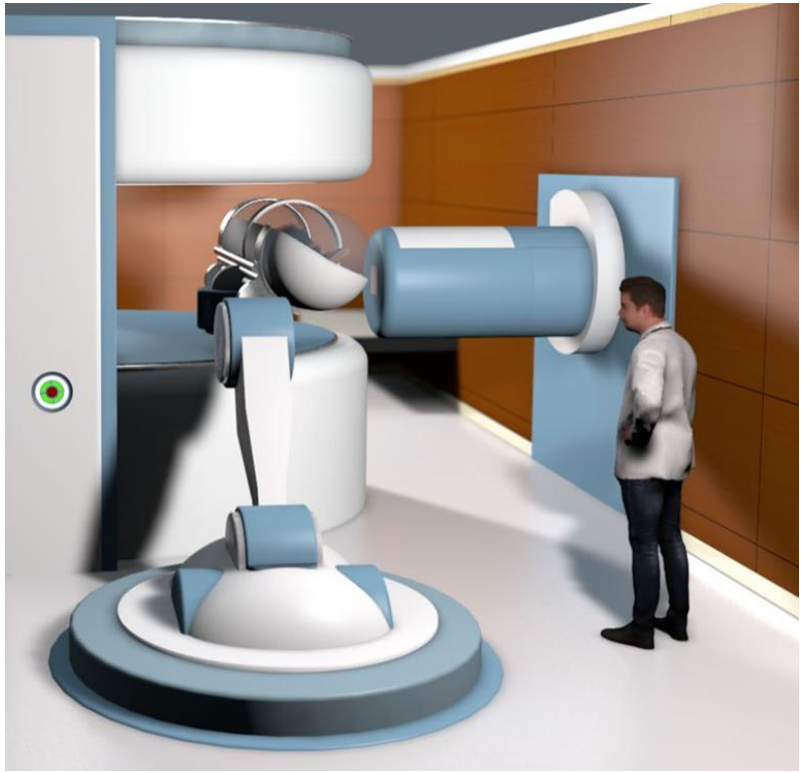
- Current beam monitoring:
  - 2 multi-wire chambers and 3 ionization chambers at beam exit
- Current tumor position monitoring:
  - CT scan prior to beam treatment



Beam acceleration and guidance system at HIT
Reference: https://www.klinikum.uni-heidelberg.de/interdisziplinaere-zentren/heidelberger-ionenstrahl-therapiezentrum-hit/leistungsspektrum

Future treatment ideas

- Current setup works well for fixed tumor positions
- Ambition: treat tumors in moving body parts (e.g. bowels)
- Idea: beam treatment during active magnetic resonance imaging (MRI)
- Problem: noisy environment, changing magnetic field
  - ➢ Current beam monitoring system not reliable inside the MRI



The ARTEMIS Project: MRI guided Ion-beam therapy
Credits: Heidelberg University Hospital

Beam monitoring requirements

Requirements:

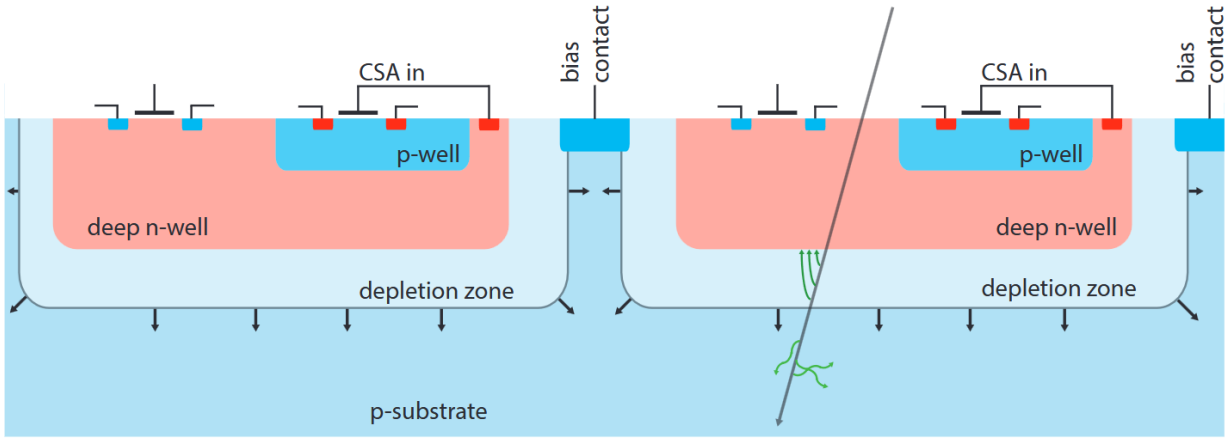
- High position (200 µm) and beam width (FWHM) (<5%) resolution
- Good intensity measurement (0.5% accuracy)
- Low latency for position and width measurement (100 µs)
- New: work in MRI environment

Implementation:

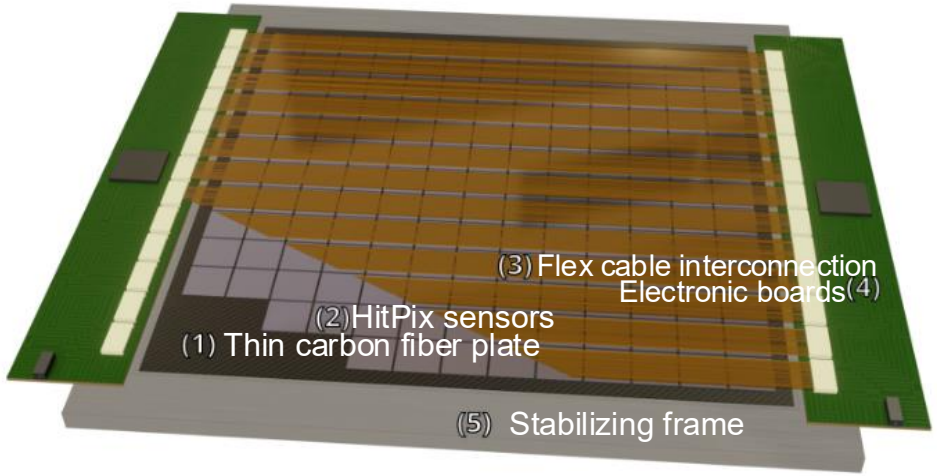
- Radiation hard for 5 years ($\Phi_{eq} \approx 10^{15}$ cm^{-2})
  - And/or simple to replace (cost/time efficient)
- Less than 0.35 mm water-equivalent thickness
- Cover target area of 25x25 cm^2

The HitPix sensor

- Idea: thin, pixelated sensor
  - ➢ Use monolithic active pixel sensor (MAPS) design
  - ➢ In pixel counter to handle high intensity
- Silicon based HV-CMOS detector
- Radiation hard (operable after $\Phi_{eq} \approx 10^{15}\,cm^{-2}$)
- good spatial resolution (~60 µm)
- Can be operated inside magnetic field/noisy environments
- Single sensor size: 2x2 cm^2
  - ➢ Create interconnected matrix to cover target area (25x25 cm^2)



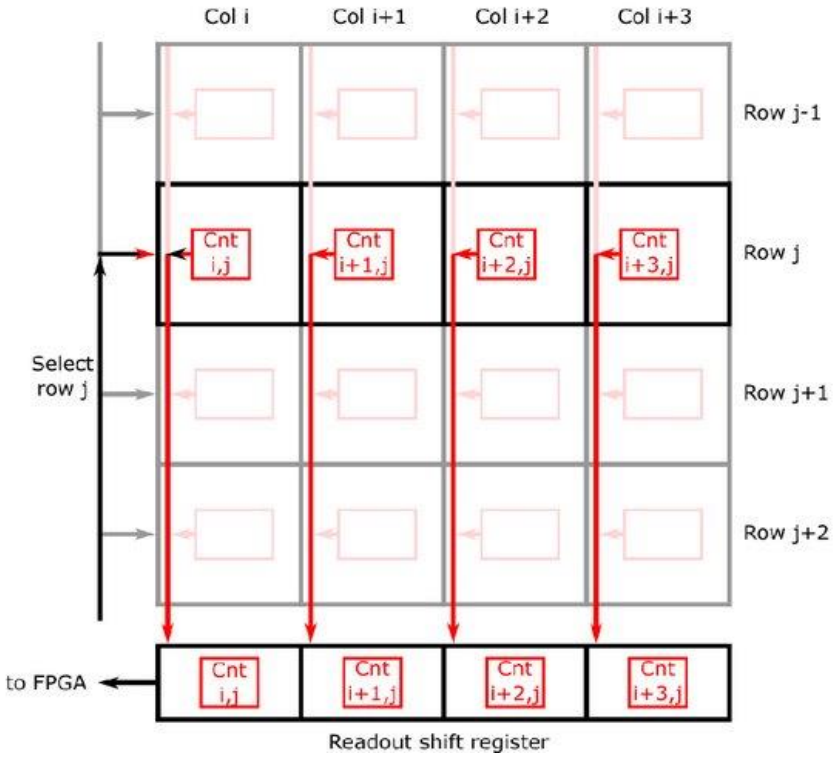
Simplified cross section of an HV-CMOS sensor
Reference: http://dx.doi.org/10.1088/1748-0221/10/04/C04007



Schematic 3D rendering of the full size matrix with 13 × 13 sensors
Reference: https://doi.org/10.3390/instruments7010009

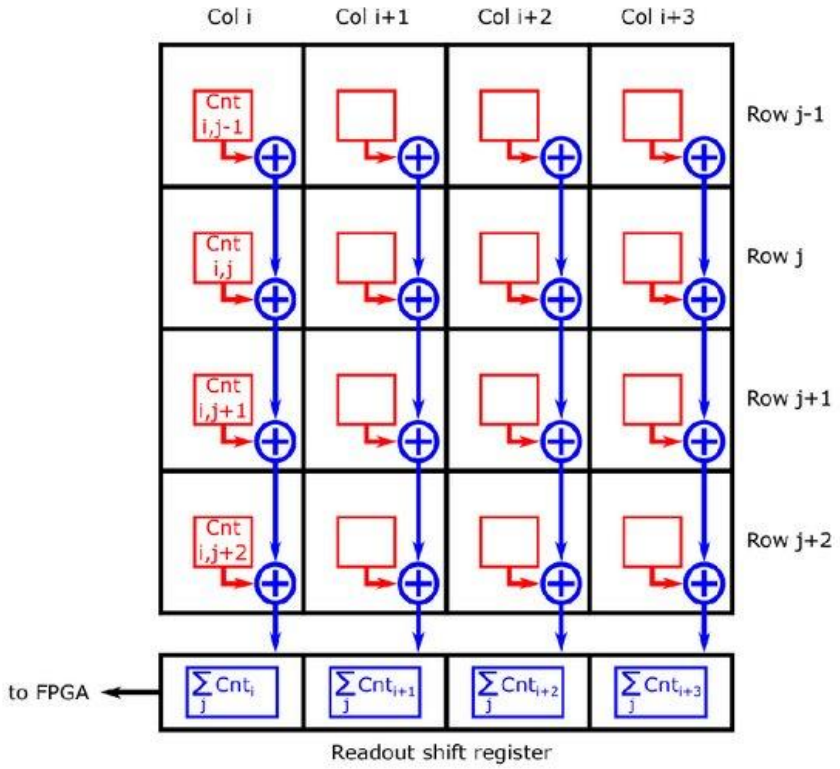
Sensor readout modes

Counter readout: Sensor readout pixel by pixel



→ Detailed information for each Pixel,
→ slow readout

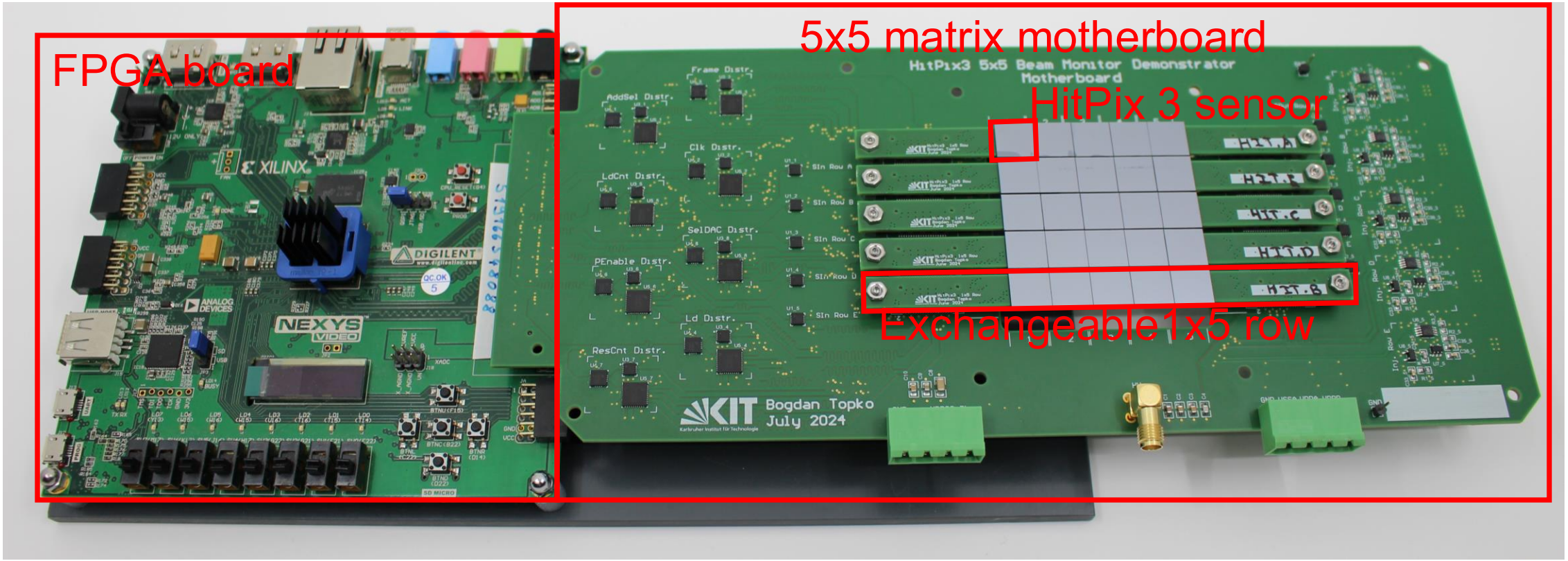
Adder readout: Rows and columns summed up to form x and y projections



→ Lose Details, keep 2D projection of hit profile
→ Faster readout (less data to pass)

The 5x5 HitPix3 demonstrator

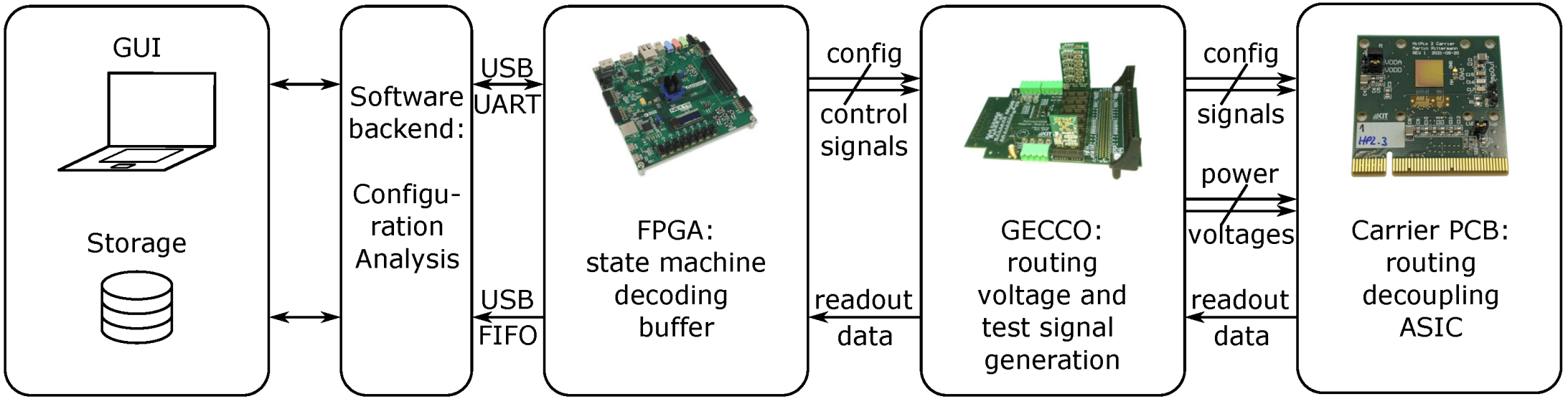
- HitPix3: 48x48 pixels, 200x200 µm pixel size



Picture of the 5x5 HitPix3 matrix mounted on the motherboard connected to the FPGA
Credits: Bogdan Topko

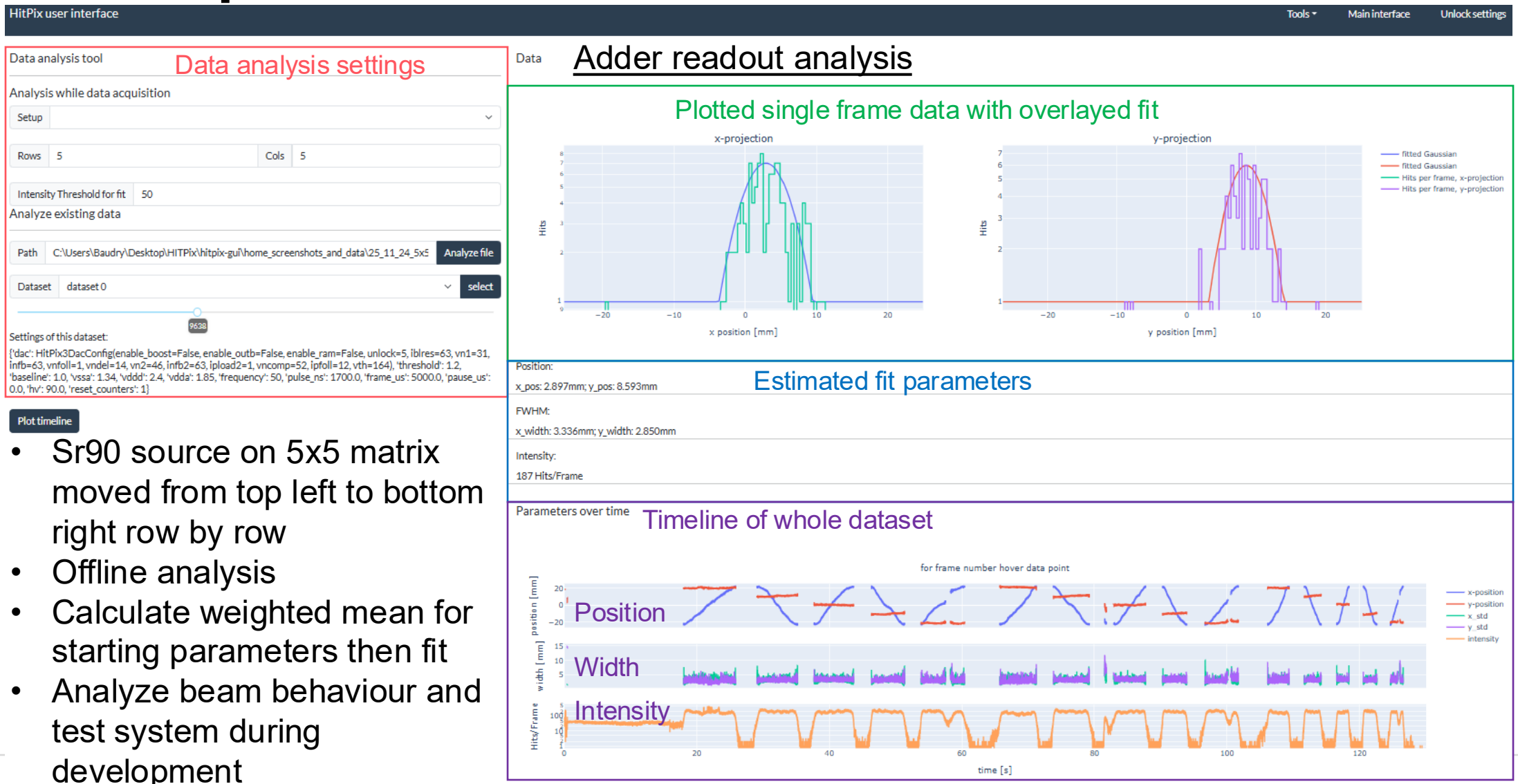
Readout system

- Pixel readout in sensor
- Data flow controlled by FPGA
- Measurement data is sent to PC via USB
- No online reconstruction at this point → Main task of my thesis



Simple block diagram of the detector readout
Reference: https://doi.org/10.3390/instruments7010009

Beam parameter reconstruction: PC level



- Sr90 source on 5x5 matrix moved from top left to bottom right row by row
- Offline analysis
- Calculate weighted mean for starting parameters then fit
- Analyze beam behaviour and test system during development

Why fit and not just use mean?

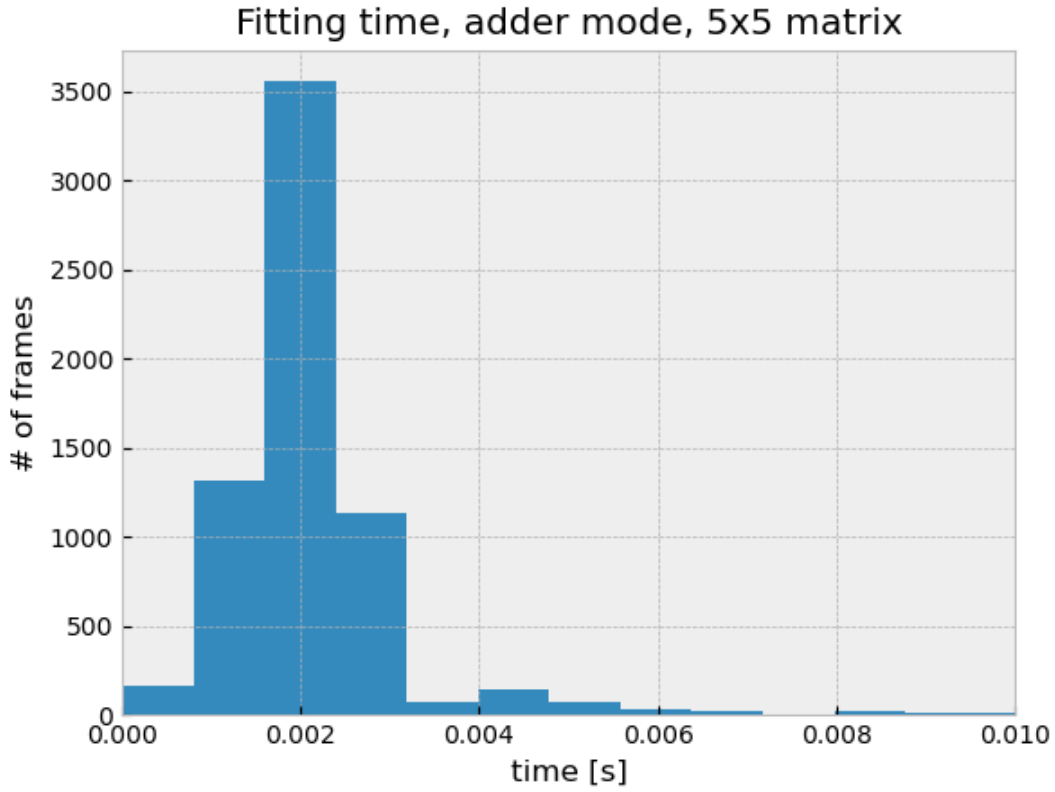
- Fit: higher accuracy especially at detector edges

But: mean calculation way faster

Screenshots of GUI page for beam reconstruction

Beam parameter reconstruction: PC level online

- For adder readout fitting time ~2-4 ms
- Fast enough for large frame sizes (5000 µs)
- Not fast enough to reconstruct every frame for sufficiently small frame sizes (ideally 50 µs to achieve 100 µs latency)
- Large frame sizes adequate for debugging and detailed beam diagnostics but not for the desired beam monitoring
- Procedure and system not optimized yet
  - Computing time could be reduced



Time needed for GUI fitting procedure for beam reconstruction

Intermediate summary PC level reconstruction

- Pre-existing GUI to initialize python scripts used for readout and configuration expanded with data analysis function
- Delivers user friendly depiction of detector data and simplifies handling the python readout scripts
- Data analysis latency too high for beam monitoring operation

  - ➢ Can be used for beam diagnostics

- Idea to reduce latency: Reduce data load between FPGA and control system

FPGA basics

- Reconfigurable integrated circuits
  - ➢ Can receive updates in contrast to dedicated microprocessors
- Consists of a finite number of logic slices, memory units, clock tiles and port connections
- Logic slices consist of look-up tables (LUTs) and flip-flops to assign and hold values
- Operates on bit-level, synchronized to a clock signal
- Parallel operations possible
- Ressources assigned to functions and connections via Hardware Description Language (HDL)

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

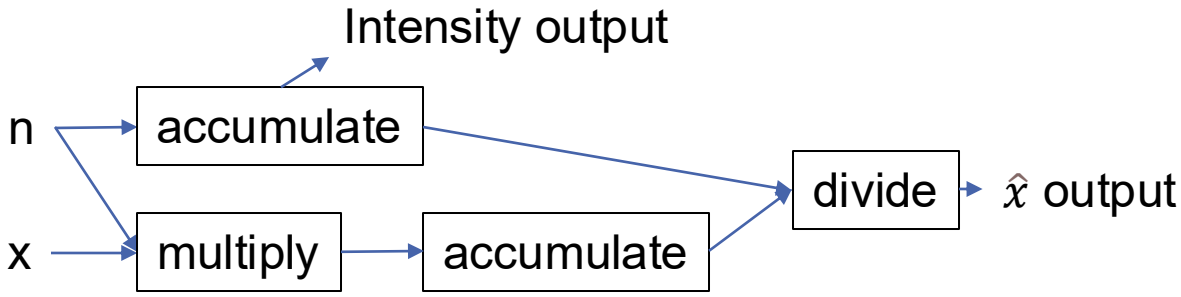
LUT of a boolean „and“ operation

Beam parameter reconstruction: FPGA level

- Write design block that does weighted mean calculation on FPGA
- Reduce data sent from FPGA to PC
- Loses details but keeps relevant parameters
- Instantly have relevant parameters

$$\hat{x} = \frac{\Sigma_i x_i n_i}{\Sigma_j n_j}$$

- Working steps:
  - Break down formulas into single calculation steps:



  - Generate control signals for single calculation blocks

Beam parameter reconstruction: FPGA level

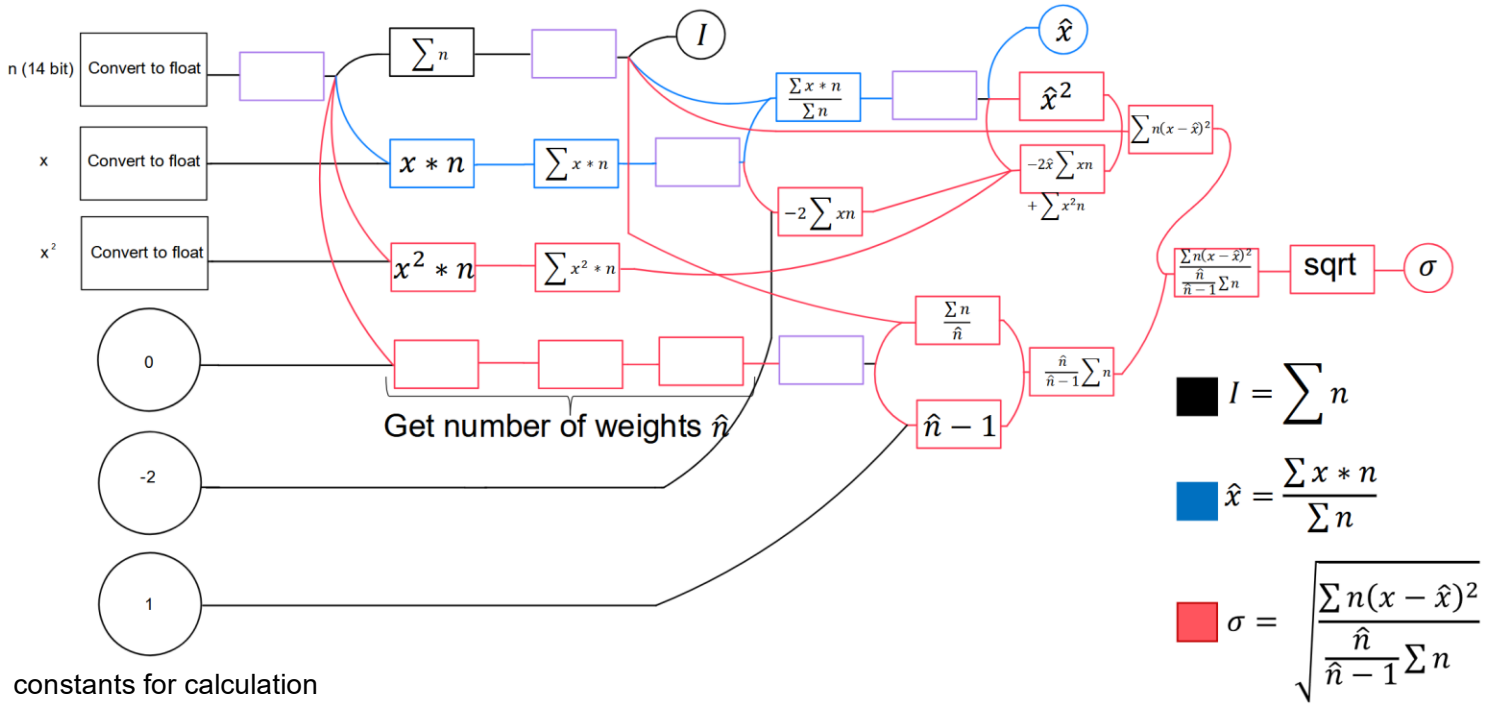
Inputs:

Number of hits n per pixel as 14 bit integer

Positional values x stored in writeable register (measured and uploaded to FPGA before data taking)

x^2 stored and not calculated to save time

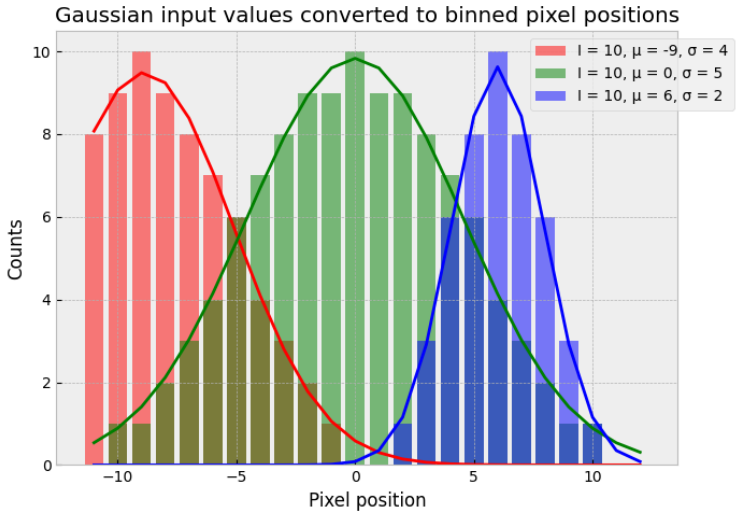
2 Control signals



Results: Intensity I , mean position $\hat{x}$, standard deviation σ as floating point values for each readout frame

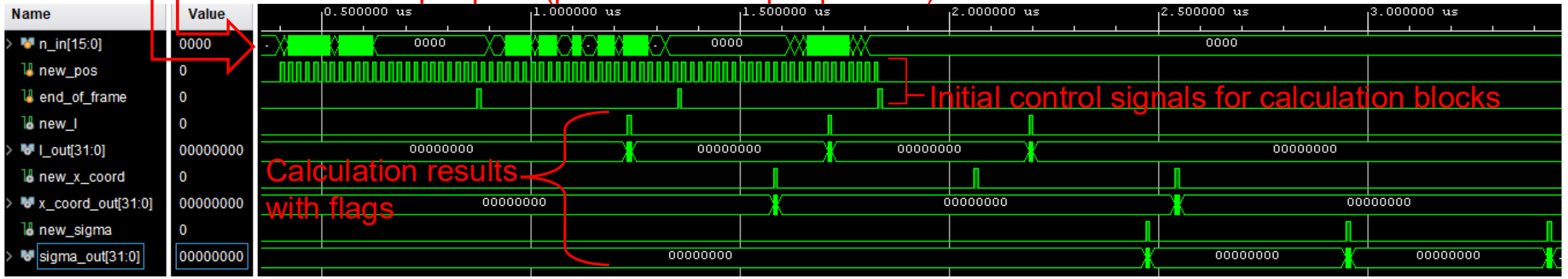
Diagram of the whole calculation broken down into its steps for FPGA implementation
Single calculations work at 32 bit floating point precision (later to be 16 bit floating point to further reduce data load)

Beam parameter reconstruction: FPGA level: simulation



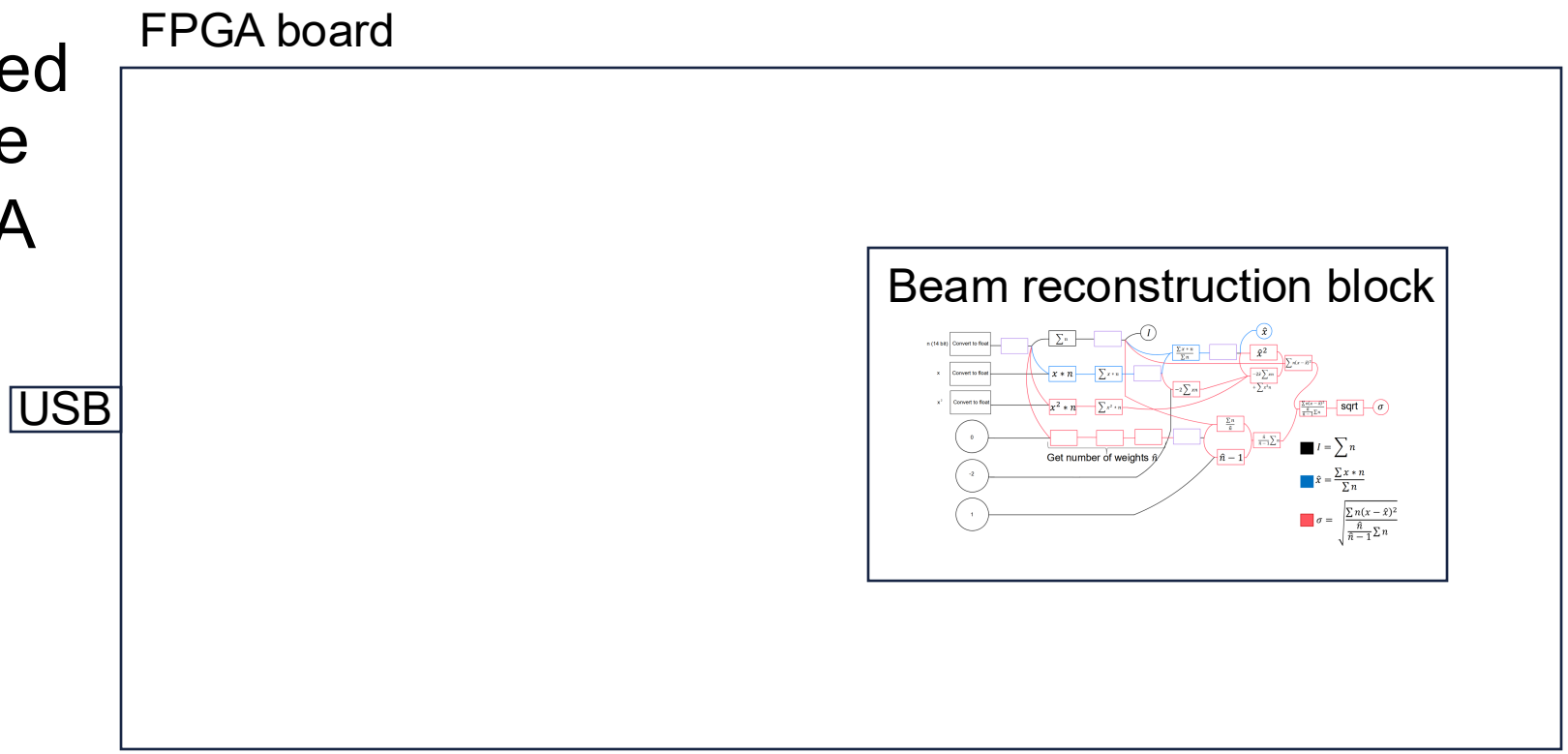
- Calculated values on FPGA match expectations
- Latency for first results after end of measurement frame (including read-in to calculation block):
  - simulation setup (24 pixel positions * 1 sensor, 1D):
    I: 0.84 µs; $\hat{x}$: 1.19 µs; σ: 2.08 µs
  - Scaled up to one row with 13 large sensors (96 pixels * 13 sensors, 1D):
    I: 25.32 µs; $\hat{x}$: 25.67 µs; σ: 26.56 µs
  - 100 MHz clock
- No data loss on consecutive frame input

Generated hits n per pixel (parallel read-in per position)



Beam parameter reconstruction: FPGA level: implementation

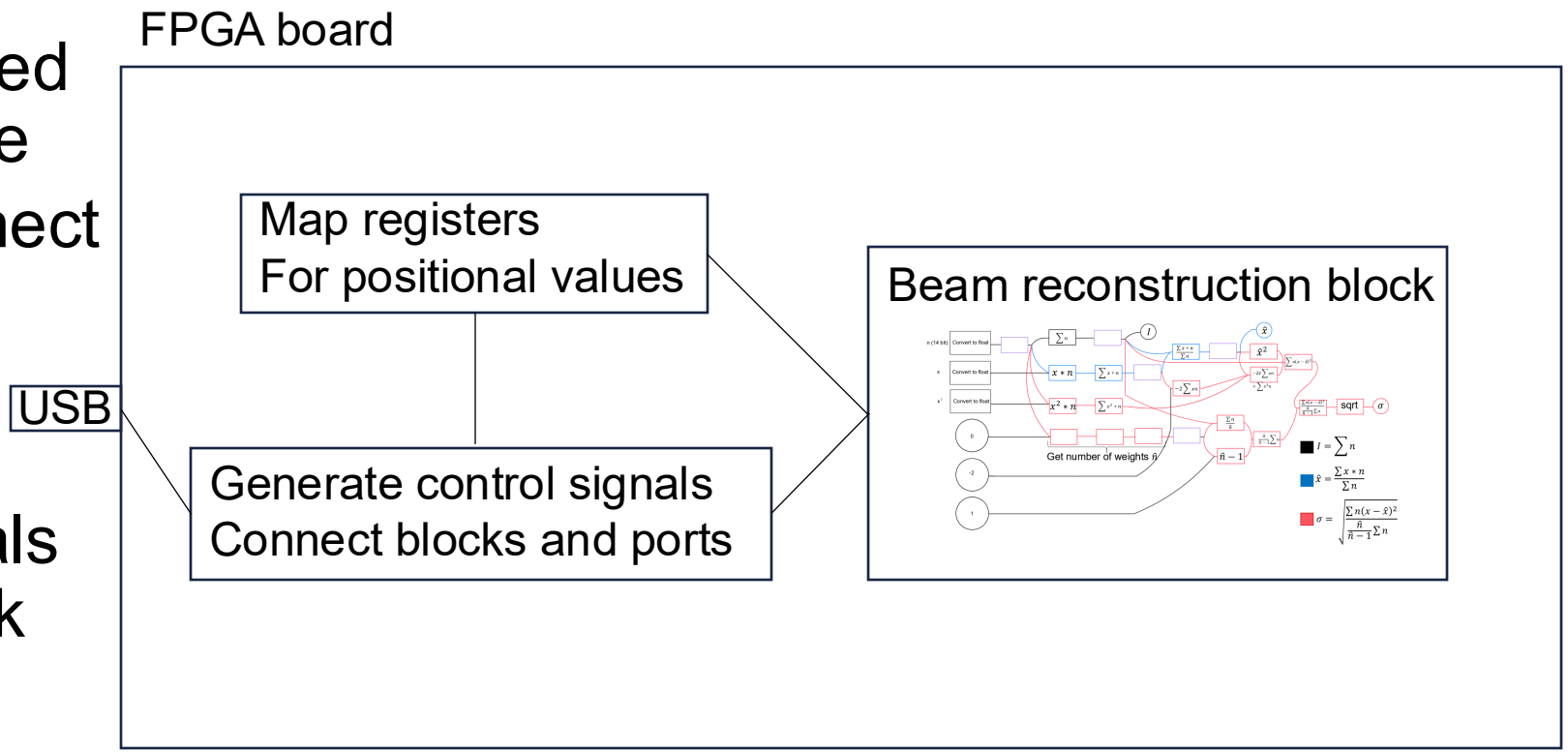
- Reconstruction designed to be insertable module
- No connection to FPGA board ports
- Need register to hold positional x values



Beam parameter reconstruction: FPGA level: implementation

- Reconstruction designed to be insertable module
- Create module to connect ports
- Create module to hold positional values
- Generate control signals for reconstruction block

- Can be combined with existing readout firmware



Beam parameter reconstruction: FPGA level: Resource utilization

- Reconstruction block design needs to fit on the FPGA besides the existing readout design
- Additionally require space for the map registers
- Beam estimator block design only: 7.03 % LUT utilization on a Nexys Video FPGA
- With 48 position register (1 HitPix3 chip): 8.51 %
- With 240 position register (1x5 HitPix3 row): 23.46 %
- Scaling up to 13x13 matrix not possible within simulation
- Note: includes both x and x^2 registers
  - ➢ can omit x^2 register and calculate for each position (trade utilization for computing time)

Beam parameter reconstruction: FPGA level: Scaling ideas

- Need to scale to 2D:
  - Use calculation module twice in parallel (faster, more resources)
  - Use calculation module two times in a row (slower, less resources)
- Need to scale to full matrix:
  - Remove x^2 map, calculate instead (slightly slower, much less resources)
  - In case x map needs to many resources:
    - Use counter for pixel positions and convert to mm seperately (less precision, much less resources)
- To be evaluated and tested

Outlook and next steps

- Go to 96x96 pixels for a single sensor
- Scale up the matrix to cover target area (25x25 cm^2)
- Improve on GUI to present fit results every n^{th} frame
- Implement beam reconstruction on FPGA to existing readout and check scaleability
- Plan to move to more powerful FPGA

Conclusion on beam monitoring

- Currently built a 5x5 HV-CMOS detector demonstrator matrix for beam monitoring
- Online beam reconstruction work in progress
- PC level reconstruction using a fitting procedure:
  - Good tool for offline beam diagnostics
  - For online analysis only large frame sizes or not every frame

- FPGA level reconstruction:
  - Reduces the data load sent to beam control system
  - Works in 1D, in principle in 2D as well
  - Needs more work:
    - Check scaleability
    - Implement in readout process