FPGA Based Readout for Silicon Strip Sensor Frontends

Karsten Koop

HISKP Uni Bonn

XXX. $\overline{\text{P}}$ANDA Meeting
2009-09-08

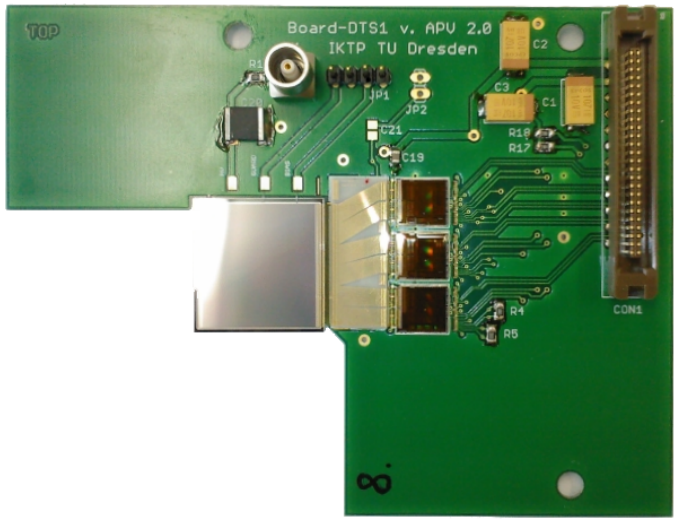
universitätbonn

Outline

1. Introduction
   - Current Test Station Setup
   - APV Output
   - FPGA based processing
2. FPGA based readout
3. Comparison
4. Tracking Station

universitätbonn

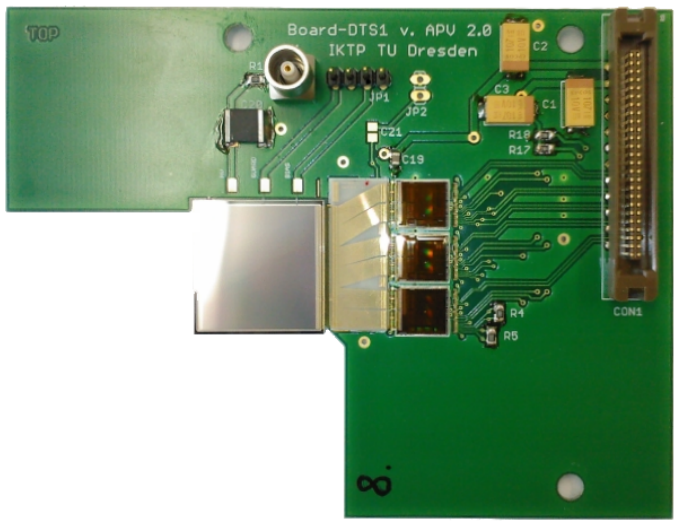
Current Test Station Setup



- Test Stations in Bonn and Dresden, using sensor prototypes
- Sensors on L-shaped test boards
- frontend chip is APV25-S1
- both single and double sided readout

universitätbonn HISKP

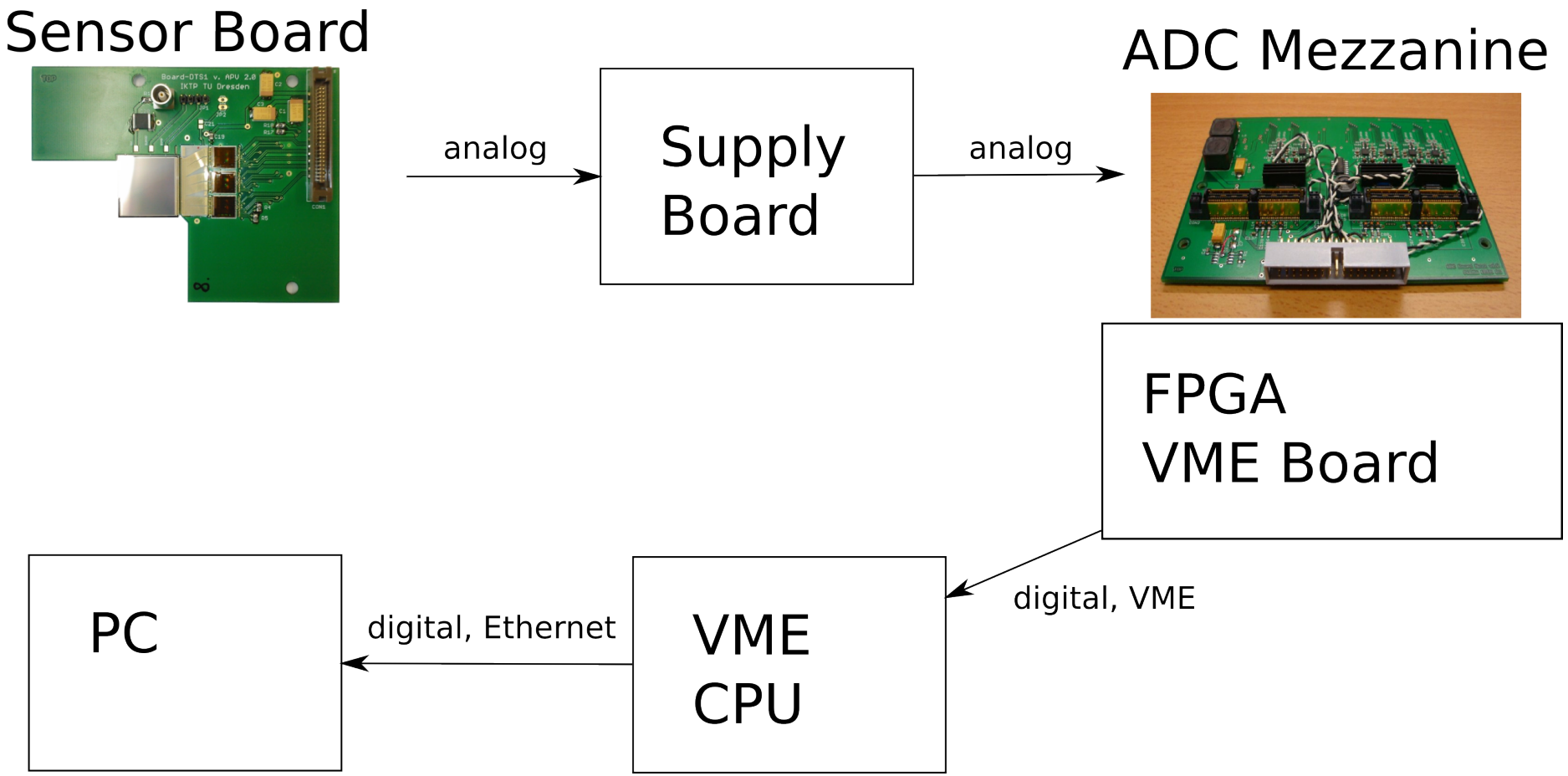
Current Test Station Setup



- Test Stations in Bonn and Dresden, using sensor prototypes
- Sensors on L-shaped test boards
- frontend chip is APV25-S1
- both single and double sided readout

universitätbonn HISKP

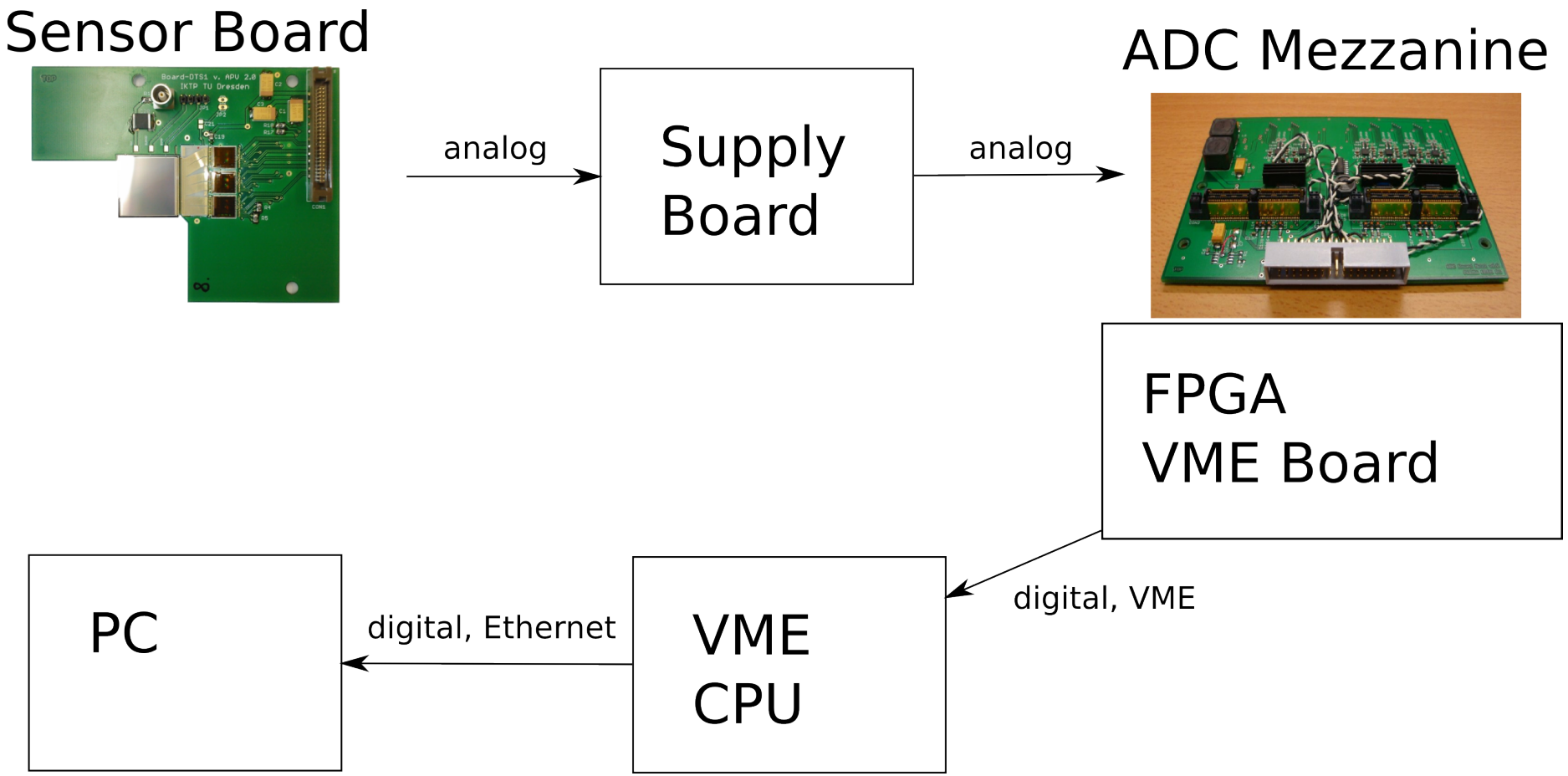
Current Test Station Setup



- ADC+FPGA first used as standard ADC, writing data to FIFO
- now using FPGA to analyse output

universitätbonn HISKP

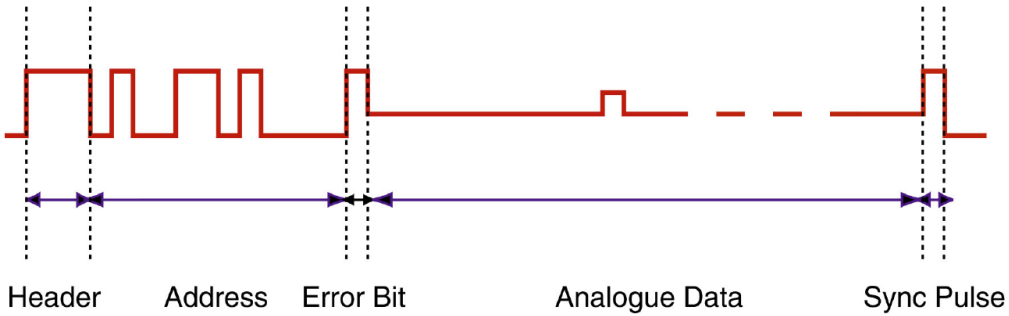
Current Test Station Setup



- ADC+FPGA first used as standard ADC, writing data to FIFO
- now using FPGA to analyse output

universitätbonn HISKP

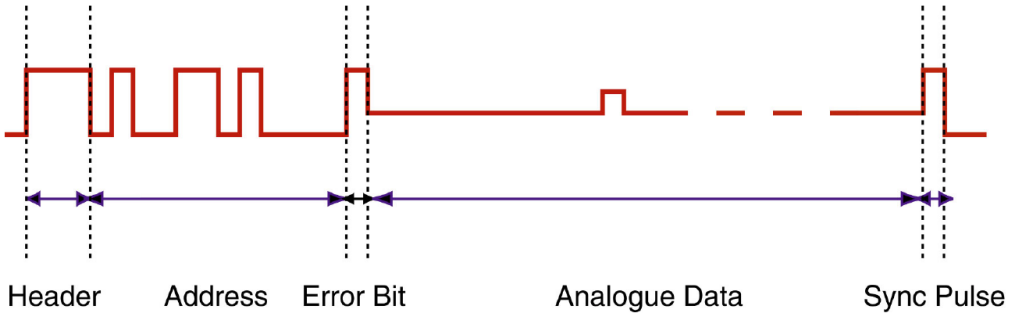
APV Output



- APV output is analog with digital header
- consists of frames of 128 channels each
- Multiple frames for one trigger

universitätbonn HISKP

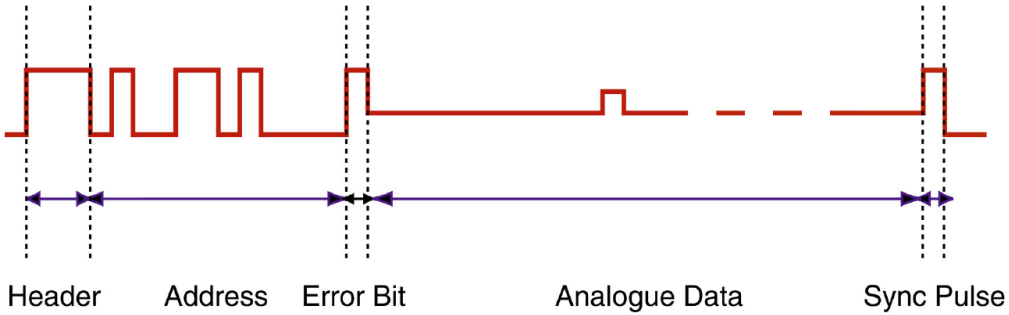
FPGA based processing



steps

1. find digital levels (from heartbeats)
2. find header, read frame
3. find baseline
4. subtract and track pedestals for individual channels
5. find channels which are above threshold $\longrightarrow$ hits
6. maybe: find hits belonging to one cluster
7. optional: track noise (std. dev. of pedestals)

FPGA based processing



steps

1. find digital levels (from heartbeats)
2. find header, read frame
3. find baseline
4. subtract and track pedestals for individual channels
5. find channels which are above threshold $\longrightarrow$ hits
6. maybe: find hits belonging to one cluster
7. optional: track noise (std. dev. of pedestals)

Implementation

PC

- Baseline: Median of channel values
- Pedestal: Mean of last 1000 values per channel
- Noise: std. dev. of last 1000 pedestals

FPGA

- Baseline: Mean of channels without hit (2 stages)
- Pedestal: $\text{oldMean} \cdot \frac{15}{16} + \text{newValue} \cdot \frac{1}{16}$
- Noise (Variance): $\text{oldNoise} \cdot \frac{127}{128} + \text{newValue}^2 \cdot \frac{1}{128}$

universitätbonn HISKP

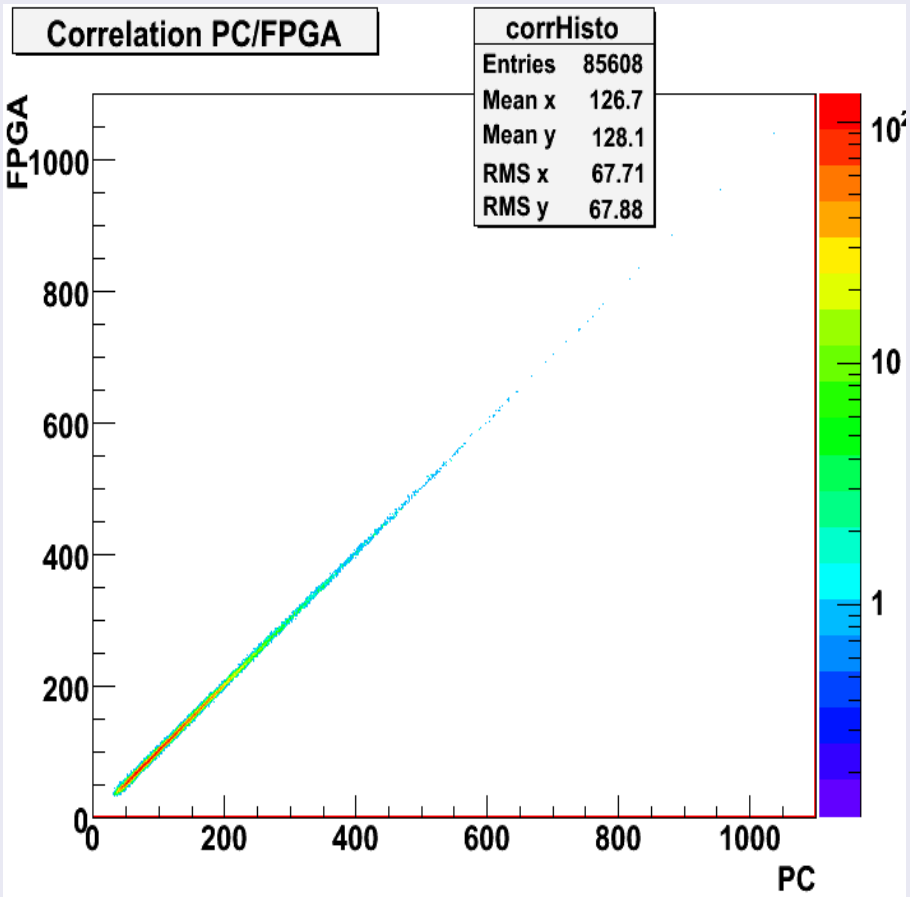
FPGA based readout

- data from 6 APV can be processed by one board
- FPGA allows parallel processing of data
- same data is stored in FIFO and processed by hitfinder/clusterfinder
- comparison of algorithms possible

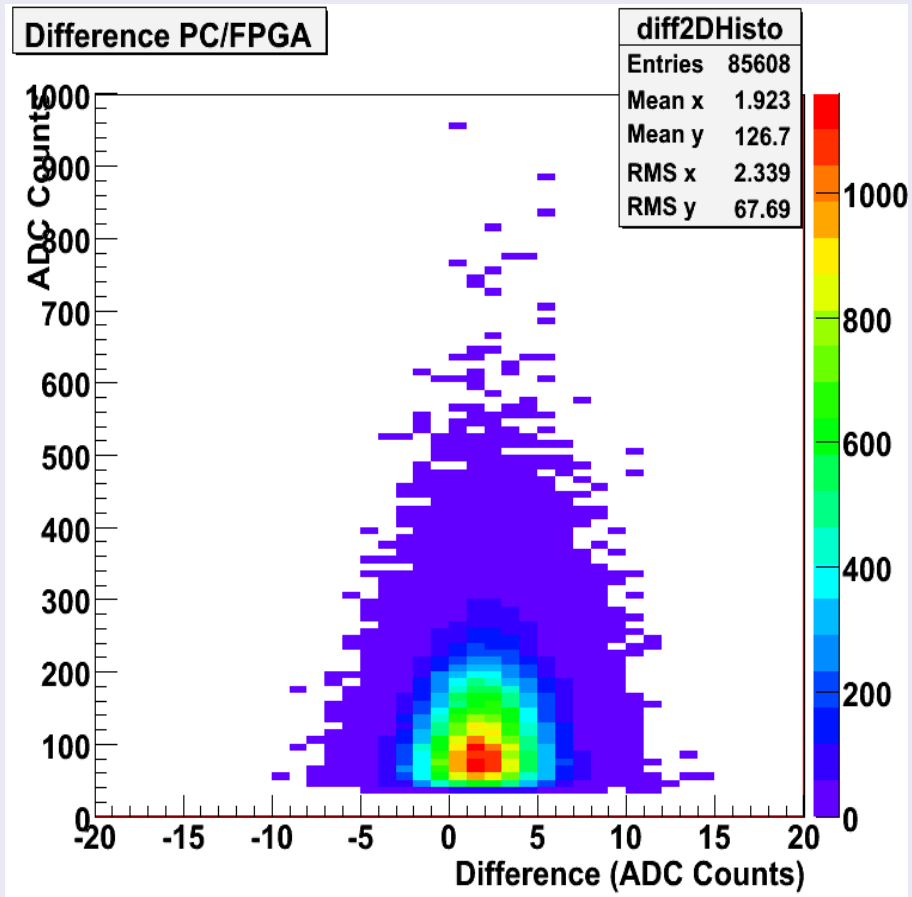
universitätbonn HISKP

Comparison between PC/FPGA

Correlation



Difference



universitätbonn HISKP

Comparison between PC/FPGA

- Values for pedestals and noise are almost identical
- Hits show small deviations, less than 5%
- Output format for hitfinder identical, can be directly compared
- Data reduction: approx. factor 100
- Reduced latency: readout rate increased by factor 1000

universitätbonn HISKP

Clusterfinder

Next step: Clusterfinder (1-dim):

- Takes output of all 6 hitfinders
- Builds clusters from hits in adjacent strips
- Calculates center of gravity, cluster sum and width
- Similar to software clusterfinder

universitätbonn HISKP

Summary

- FPGA based readout gives almost identical results
- Can be used to reduce data
- Works in realtime
- Clusterfinder for one dimension working

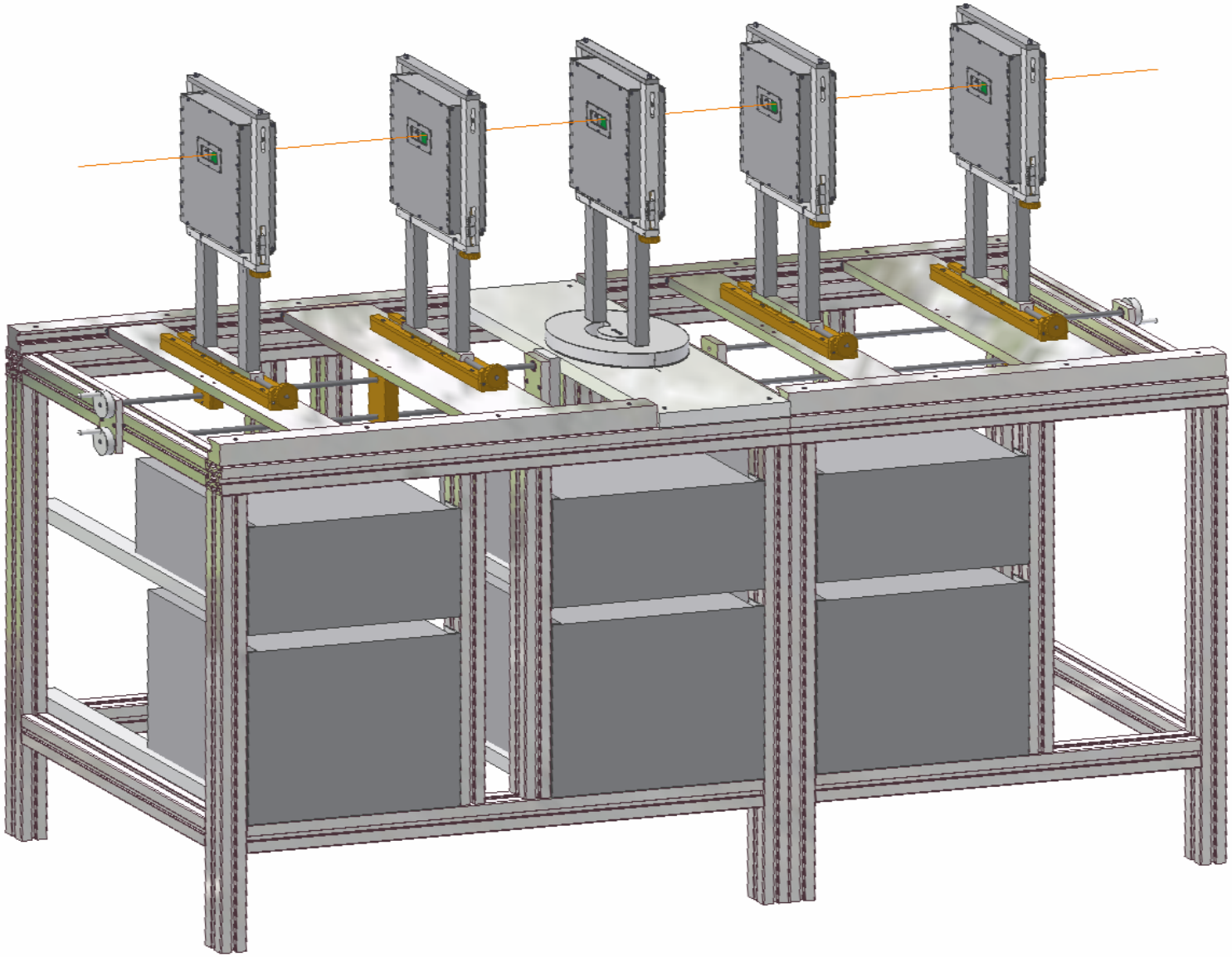
universitätbonn HISKP

Outlook

- 2-dim Clusterfinder: find corresponding clusters
- Test in tracking station
- Connect different sensors to FPGA, find tracklets

universitätbonn HISKP

Tracking Station



Tracking Station

universitätbonn HISKP