FPGA-based Cherenkov Ring Recognition in Nuclear and Particle Physics Experiments

Ming Liu‡†, Zhonghai Lu†, Wolfgang Kuehn‡, Axel Jantsch†

‡ II. Physics Institute, Justus-Liebig-University Giessen (JLU), Germany
† Dept. of Electronic Systems, Royal Institute of Technology (KTH), Sweden

mingliu@kth.se

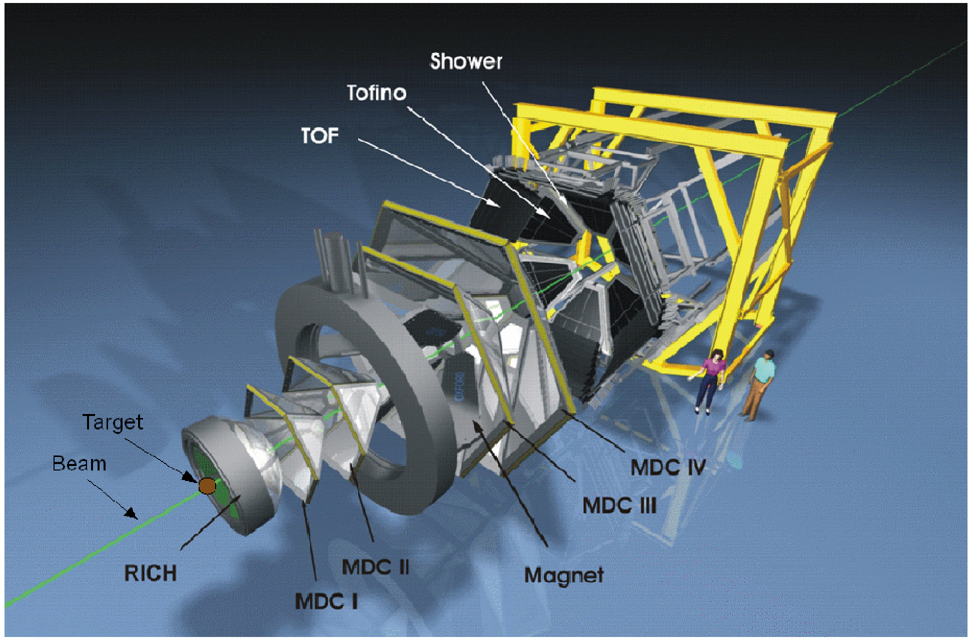
Outline

1. Introduction and Motivation
2. Algorithm Description
3. Implementation on the FPGA
4. Experimental Results
5. Summary

Introduction and Motivation

- **Nuclear & particle physics experiments**
  - Study the elementary constituents of matter and interactions between them
  - Particles created by energetic collisions of others
  - Complex detector system

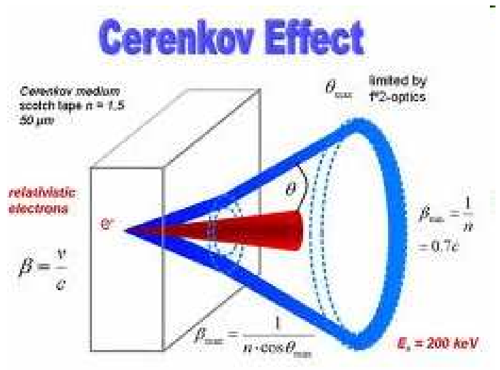
- **Trigger algorithms**
  - Filter detector-generated reaction events by identifying interesting reactions and discarding noise data
  - Large raw data rate & complex pattern recognition algorithms
  - Online (FPGA, DSP, …) or offline (PC cluster)
  - e.g. Cherenkov ring recog., MDC tracking, …

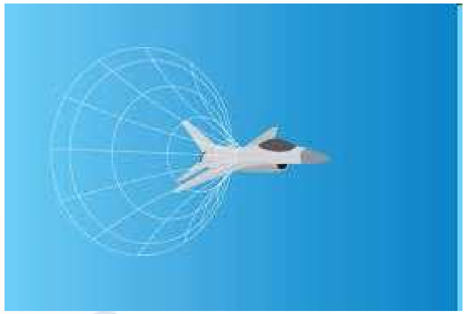


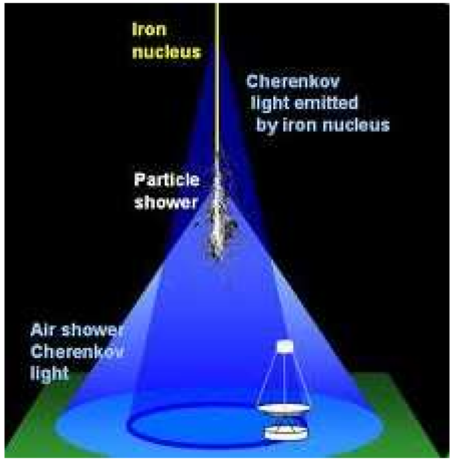
HADES detector system

Introduction and Motivation

- Cherenkov effect
  - Light emission occurs in a transparent substance, when a charged particle travels through the material with a speed faster than the light speed in that material
  - Nobel Prize of physics in 1958
  - Theory foundation of the Ring-Image CHerenkov (RICH) detector
  - Used to identify charged particles by recognizing the reflected ring shape on the detector
- Motivation: to implement efficient computation of ring recognition



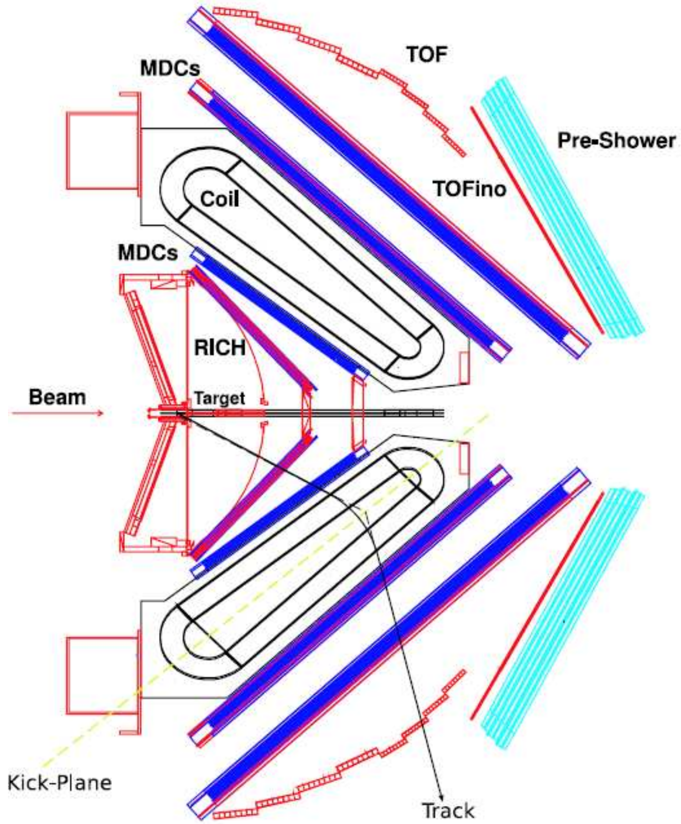




Algorithm Description

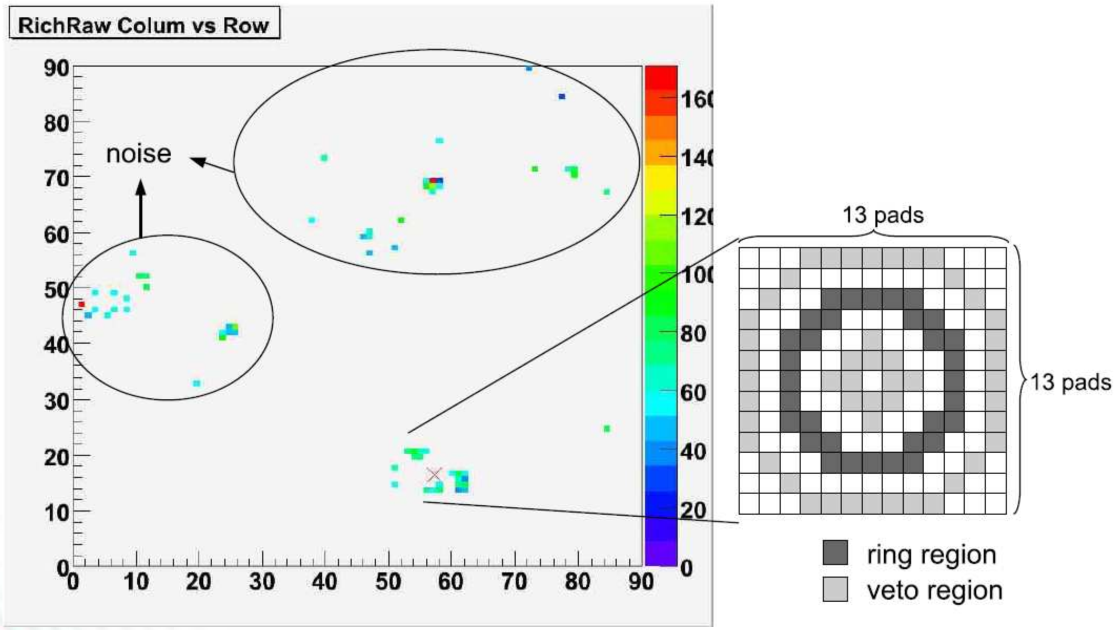
- RICH & MDC detectors in HADES
  - MDC detectors for particle tracking (straight segments of flight in MDCs)
  - RICH detector to identify dilepton pairs (e^+e^-)
  - MDC-RICH correlation: dilepton pairs identified only if both tracks in MDCs and rings on RICH recognized

ps: Mini-Drift Chamber (MDC) detectors



Lateral cut-away view of the HADES detector system

Algorithm Description

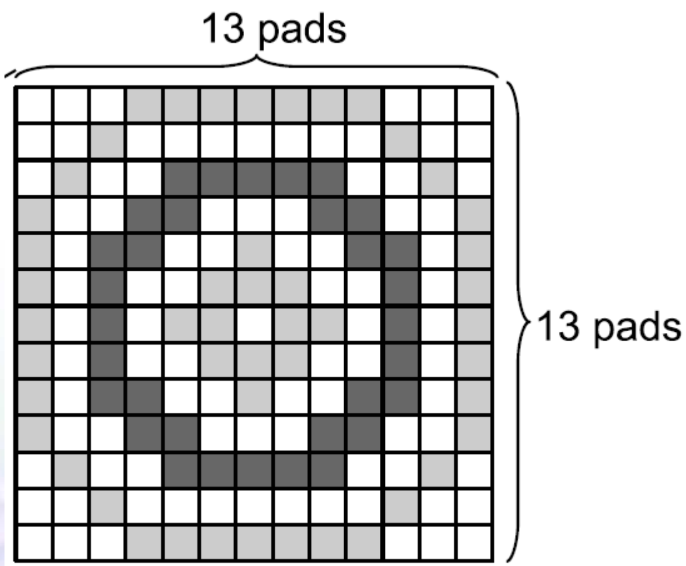


Detector-matching algorithm[1]:

- RICH plane resolution: 96 x 96
- Hit pixels from the Cherenkov light or noise
- Ring diameter fixed as 8 pixels (velocity of e^+e^-)
- Ring pattern searched in the mask region of 13 x 13
  - Inner & outer veto regions
  - Ring region

1 G. Agakichiev, et al., The high-acceptance dielectron spectrometer HADES, The European Physical Journal A 41, pp. 243 – 277, 2009.

Algorithm Description



- Hit position judgment:
  - Inner veto region
  - Outer veto region
  - Ring region

```
-- (x1, y1) : position coordinate of the hit pixel
-- (x0, y0) : position coordinate of the ring center candidate
-- x_distance = abs(x1 – x0); : distance on x axis from the hit pixel to the center candidate
-- y_distance = abs(y1 – y0); : distance on y axis from the hit pixel to the center candidate
-- hop =  x_distance + y_distance; : hop distance from the hit pixel to the center candidate

position_result <= FARAWAY_NOISE_REGION; -- by default the hit pixel is deemed as noise

if (hop <=2) then
      position_result <= INNER_VETO_REGION; -- the hit pixel falls into inner_veto_region
end if;

if (((hop = 5 or hop = 6) and x_distance <= 4 and y_distance <= 4)
  or (y_distance = 0 and x_distance = 4)
  or (y_distance = 4 and x_distance = 0)) then
      position_result <= RING_REGION;            -- the hit pixel falls into ring_region
end if;

if ((hop = 9 and x_distance <= 6 and y_distance <= 6)
  or (y_distance = 6 and x_distance <= 2)
  or (x_distance = 6 and y_distance <= 2)) then
      position_result <= OUTER_VETO_REGION; -- the hit pixel falls into outer_veto_region
end if;
```

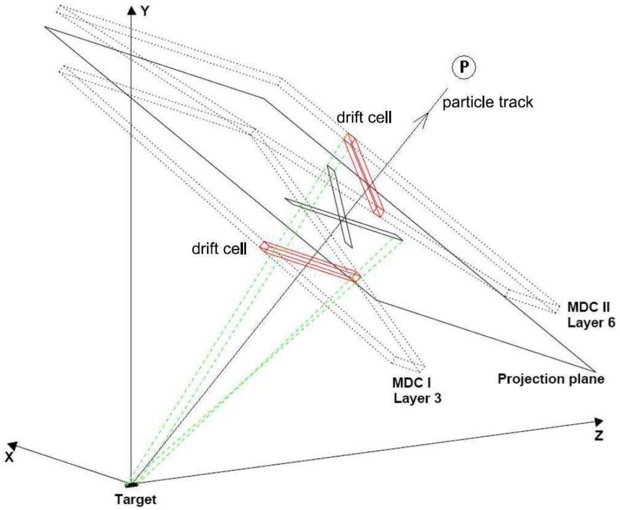
Algorithm Description

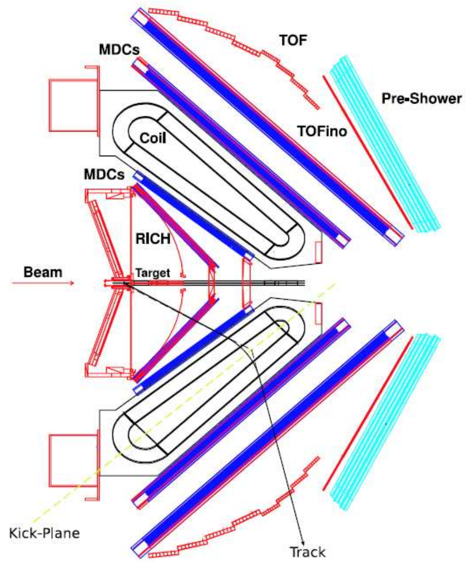
- To accumulate the hit number of the ring_region, inner_veto_region and outer_veto_region
- Respective programmable thresholds
- Ring pattern identified, if

$$\text{ring\_region} \geqslant \text{threshold}_{\text{ring}}$$
$$\text{inner\_veto\_region} \leqslant \text{threshold}_{\text{inner}}$$
$$\text{outer\_veto\_region} \leqslant \text{threshold}_{\text{outer}}$$

Algorithm Description

- Challenge: to specify the ring center candidates
  - Blind search on all 96 x 96 pixels (computation inefficient)
  - Specific search on specified ring center candidates
- MDC tracklets used to point out ring center candidates
  - To prolong the MDC tracks, pointing on the RICH detector
  - MDC-RICH correlation
  - Coordinate conversion from MDC to RICH using a LUT
  - Resolution difference (MDC 128 x 256, RICH 96 x 96)
  - Search window (e.g. 5 x 5) to avoid pattern loss



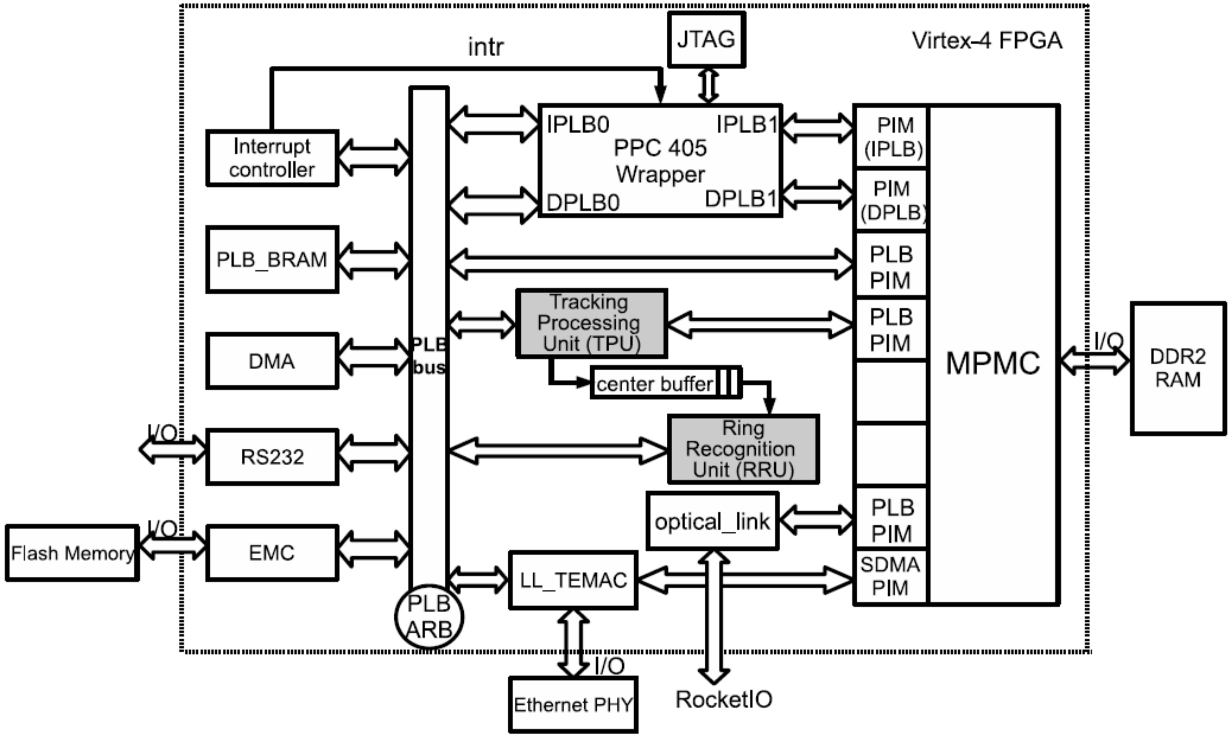


Solution Comparison

Previous solution in HADES[1,2]	New solution
MDC tracking on PCs (offline), RICH ring recog. on FPGAs	MDC tracking & RICH ring recog. on FPGAs
Blind search of ring patterns	Specific search with MDC-RICH correlation
Needs to reconstruct the hit plane in a memory device	No need to reconstruct the hit plane
Treat all pixels as potential ring centers for computation.	Traverse only all the hit pixels for the specified ring center candidates.
HW implementation on obsolete FPGAs	HW implementation on modern SoC FPGAs
Inadequate online data reduction rate for mass storage	Efficient online data reduction, storage and mitigation of offline computation

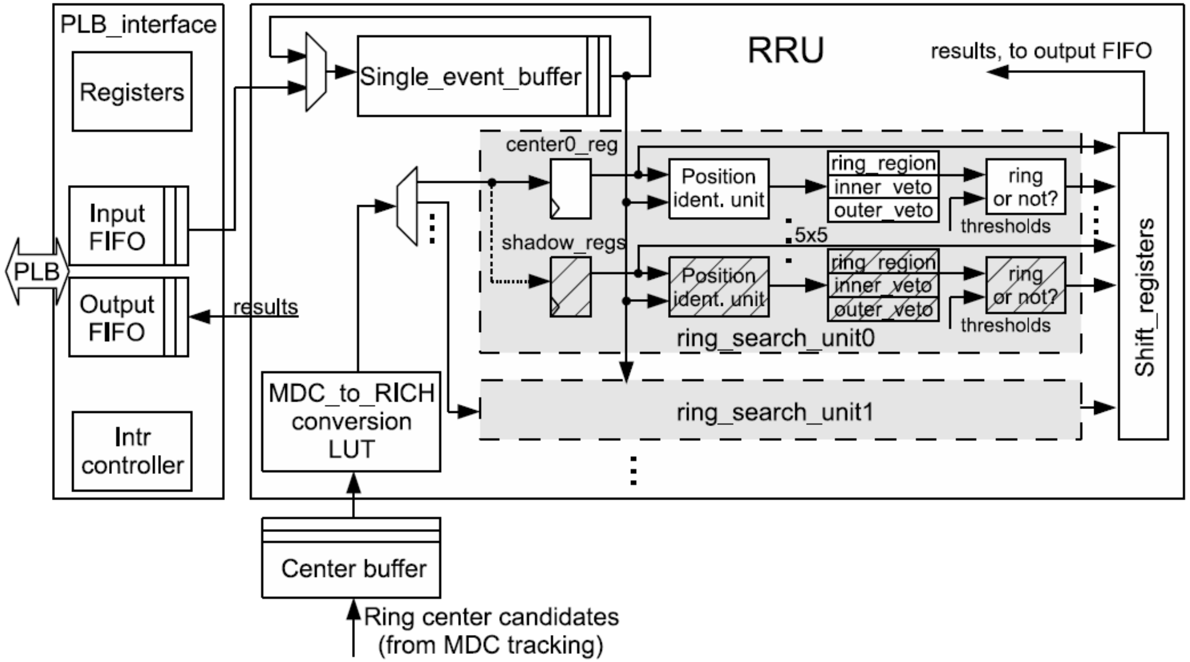
1 J. Lehnert, et al., Ring Recognition in the HADES Second-level Trigger, *Nuclear Instruments and Methods in Physics Research A 433*, pp. 268 - 273, 1999.
2 J. Lehnert, et al., Performance of the HADES Ring Recognition Hardware, *Nuclear Instruments and Methods in Physics Research A 502*, pp. 261 - 265, 2003.

HW Implementation



- FPGA Implementation
  - Bus and MPMC-based SoC design
  - DDR2 memory for data buffering
  - Input and output links: optical & Ethernet
  - Algorithm engines: Tracking Processing Unit (TPU) for MDC tracking, and Ring Recognition Unit (RRU) for RICH ring recognition

HW Implementation



- Modular design
  - PLB interface to supply the raw sub-events and collect the results
  - single_event_buffer for traversing all the hits of one sub-event
  - Center candidates loaded from the MDC tracking results
  - ring_search_unit to find ring patterns
    - Multi-core for parallel processing
    - Shadow cores for 5 x 5 search window

Experimental Results

Resources	RRU (1 ring_search_unit)	RRU (2 ring_search_unit)	system (RRU of 1 ring_search_unit)
4-input LUTs	4723 out of 50560 (9.3%)	8186 (16.2%)	21933 (43.4%)
Slice Flip-Flops	3663 out of 50560 (7.2%)	5190 (10.3%)	17185 (34.0%)
Block RAMs	31 out of 232 (13.4%)	31 (13.4%)	104 (44.8%)

- 1 ring_search_unit = 25 parallel cores (5 x 5 search window)
- Resource utilization on Virtex-4 FX60, acceptable
- Timing: 160 MHz for RRU core, 100 MHz for PLB interface

- Performance measurements
- ~ 100s – 1000s ns/Sub-event
- Processing speed estimated in the order of magnitude of MSub-events/s
- Original design[1]: 49 KSub-events/s @ 2 VME boards & 12 FPGA chips
- Speedup of 10s – 100s times
  - Algorithm improvement using MDC-RICH correlation
  - Parallel & pipelined high-speed design structure

1 J. Lehnert, et al., Performance of the HADES Ring Recognition Hardware, *Nuclear Instruments and Methods in Physics Research A 502*, pp. 261 - 265, 2003.

Summary

- Cherenkov Ring recognition algorithm introduced
- Algorithm improved by introducing the MDC-RICH correlation to point out ring center candidates (computation efficient)
- Traverse all the hit pixels for position judgment, instead of the blind search in the previous design
- Implemented on the FPGA for online processing
- Large performance speedup observed
- Future work: dynamically reconfiguring TPU and RRU, to automatically synchronize their processing

Thanks for your attention!

Acknowledgment to my colleagues, especially Dr. Vladimir Pechenov and Dr. Bjoern Spruck for their algorithm related work.