Proceedings of the CTD 2022
PROC-CTD2022-08
May 19, 2023

# Linearized Track Fitting on an FPGA

Joachim Zinsser [1], Sebastian Dittmeier, André Schöning

*Physikalisches Institut*
*Ruprecht Karls Universität Heidelberg, Germany*

ABSTRACT

For the ATLAS experiment at the High-Luminosity LHC, a hardware-based track-trigger was originally envisioned, which performs pattern recognition via associative memory ASICs and track fitting on an FPGA. A linearized track fitting algorithm is implemented (Track Fitter) that receives fit-constants corresponding to the track candidates from a database, performs a $\chi^2$-test of the track, and calculates the helix-parameters. A prototype of the Track Fitter has been set-up on an Intel Stratix 10 FPGA. The firmware was tested in simulation-studies and verified in hardware. The performance of the Track Fitter has been evaluated in extensive simulation studies.

PRESENTED AT

Connecting the Dots Workshop (CTD 2022)
May 31 - June 2, 2022

[1] Work supported by DFG Research Training Group 2058 and the Bundesministerium für Bildung und Forschung.

# 1 Introduction

For the High-Luminosity Upgrade, ATLAS' Inner Detector is replaced by the Inner Tracker (ITk) [1, 2]. With a planned peak instantaneous luminosity of $7.5 \cdot 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, a pileup of 200 inelastic proton-proton collisions is expected [3]. It was planned to include a Hardware Track Trigger (HTT) into the Event Filter System to reduce the load, for reconstructing tracks, on the processor farm [3], see Figure 1.

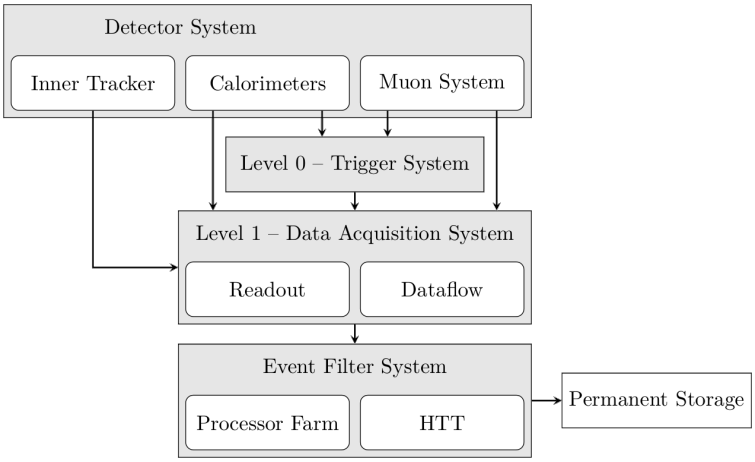


Figure 1: Summary of the ATLAS TDAQ dataflow.

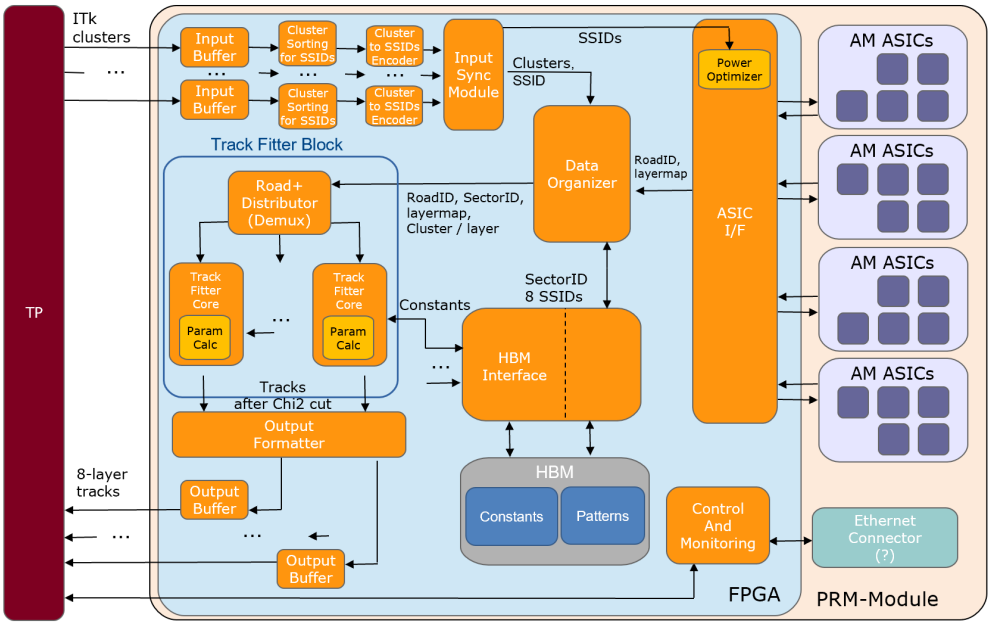


Figure 2: Firmware diagram of the PRM [4].

The HTT is split into several units, each performing tracking in regions of the ITk. The HTT consists of two types of units: Associative Memory Tracking Processors (AMTPs) and Second-Stage Tracking Processors (SSTPs) [5]. This work focuses on the Pattern Recognition Mezzanine (PRM), which is the central component of the AMTP. The PRM houses four groups of Associative Memory (AM) ASICs that organize detector hits into track candidates, as well as an Intel Stratix 10 FPGA for data handling and track fitting (see Figure 2).

## 2 Track Linking

The ITk consists of thirteen detector layers [1]. To conduct meaningful physics, hits that originate from the same particle have to be connected to a track. Examining all particle hit combinations from all layers leads to an unfeasible amount of calculations. Therefore, it is paramount that the number of track candidates is reduced as early as possible.

Several methods have been developed to solve that problem [6]. The HTT utilizes *pattern matching*, a highly parallel approach to track linking that is therefore well suited to an implementation in hardware [7].

By simulating a large amount of particle tracks using extensive Monte Carlo simulations [3] and storing the resulting patterns of hits before the experiment is set up, templates are generated to which the actual hit patterns are compared to. This technique not only finds track candidates very quickly (processing time grows linearly with number of hits) but also provides track parameters [7]. Since the number of possible tracks usually exceeds the available memory resources, patterns are stored for a reduced granularity detector model. Reducing the granularity increases the number of nearby tracks that cannot be differentiated. A trade-off between resolution and storage space has to be decided through simulation [7]. Although, for track linking only coarse hit information is used, the full hit information is recovered at a later stage for the track fitting.

The template matching is performed by the AM ASICs [8] in the following way: First, the resolution of the incoming data is reduced (see Figure 2, *Cluster to SSID Encoder*). Then all hit data of one event are loaded into the AM ASICs in series. If a data package arrives that matches an entry of the pattern bank, the entry gets flagged (see Figure 3, second row). When the AM ASIC receives a signal that the event has concluded, it checks which of its patterns have more than a given threshold of matched layers (see Figure 3, third row). Then the AM ASIC starts to return the identifiers of the matched patterns (see Figure 3, fourth row).

Since the AM ASICs have not been produced in time for the prototyping stage of the PRM development, an emulator was developed. Together with a custom interface, the emulator is implemented on the FPGA and follows the algorithm of the original ASIC design, although with reduced memory and an extra interface to access the AM database more directly.

## 3 Track Fitting

Charged particles follow a helix-trajectory in a homogeneous magnetic field. A helix fit takes a considerable amount of processing time and requires non-linear calculations which are not ideal to be performed on an FPGA.

Instead, the computationally expensive calculation of the fits is performed offline using the simulated tracks from the AM ASIC database. For small variation in the hit-data, the circular form of the track can be approximated as a parabola so that the fit parameters $p_i$ and the goodness $\chi^2$ vary in a linear way [9]. Therefore, the fit parameters and goodness can be approximated using the measured hit-data of a track candidate and the constants that describe the linear variation. The database of simulated tracks is divided into regions for which the constants of the linear equations are sufficiently similar. These constants are stored in a second database on the FPGA to be used in online track reconstruction.

Each parameter and each degree of freedom of the fit need their own constants for each of the eight detector layers that is used for track reconstruction. A full set of constants for a single track fit represent roughly 5 kbit of data that have to be made available to the FPGA at each clock cycle. This necessitates the usage of an FPGA with up to 16 GB, 512 GB/s High Bandwidth Memory (HBM) like the Intel Stratix 10 [10].

For the track fitting, the FPGA receives the full-resolution hit data $\vec{x}$ of a track candidate and retrieves the constants of the linear equations corresponding to the track candidates sector. The five helix parameters $p_i$ are calculated with the constants $\vec{C}_i$ and $q_i$ [3]:

$$p_i = \vec{C}_i \cdot \vec{x} + q_i \ . \tag{1}$$

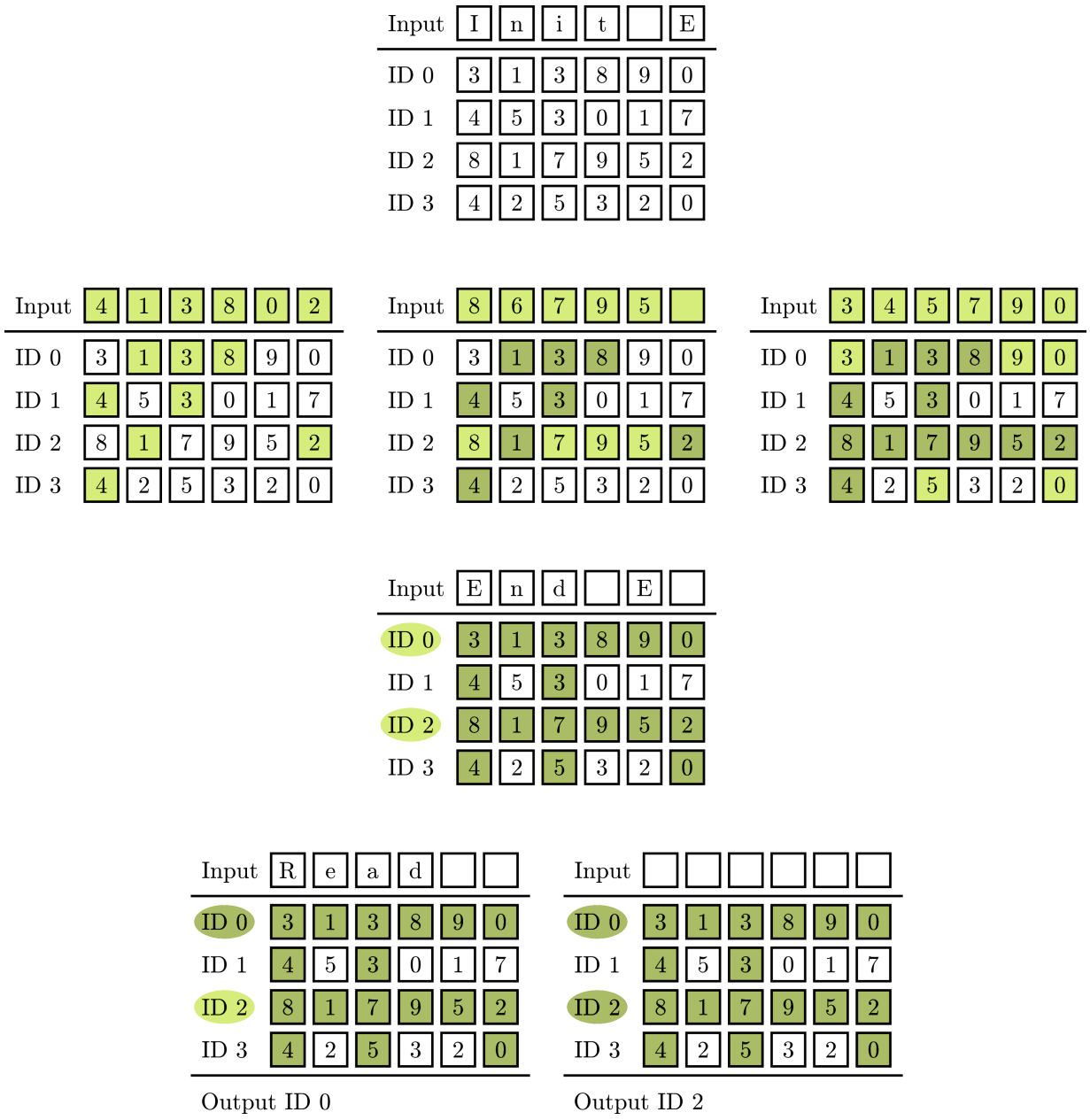


Figure 3: Simplified representation of the pattern recognition algorithm for a detector with six layers (columns) and a pattern bank with four entries (rows). The steps of the algorithm represent a single clock-cycle each and happen chronologically from top to bottom and left to right. The displayed event has up to three hits per detector layer from which two patterns are found.

For $J$ degrees of freedom, the goodness $\chi^2$ of the fit is calculated with the constants $\vec{S}_j$ and $h_j$ [3]:

$$\chi^2 = \sum_{j=0}^{J} \left( \vec{S}_j \cdot \vec{x} + h_j \right)^2 . \tag{2}$$

The resulting goodness is then compared with a preset threshold. Since only 20 % of the track candidates are expected to pass the goodness test [4], the goodness is calculated first to save resources on the FPGA.

# 4 Firmware Implementation

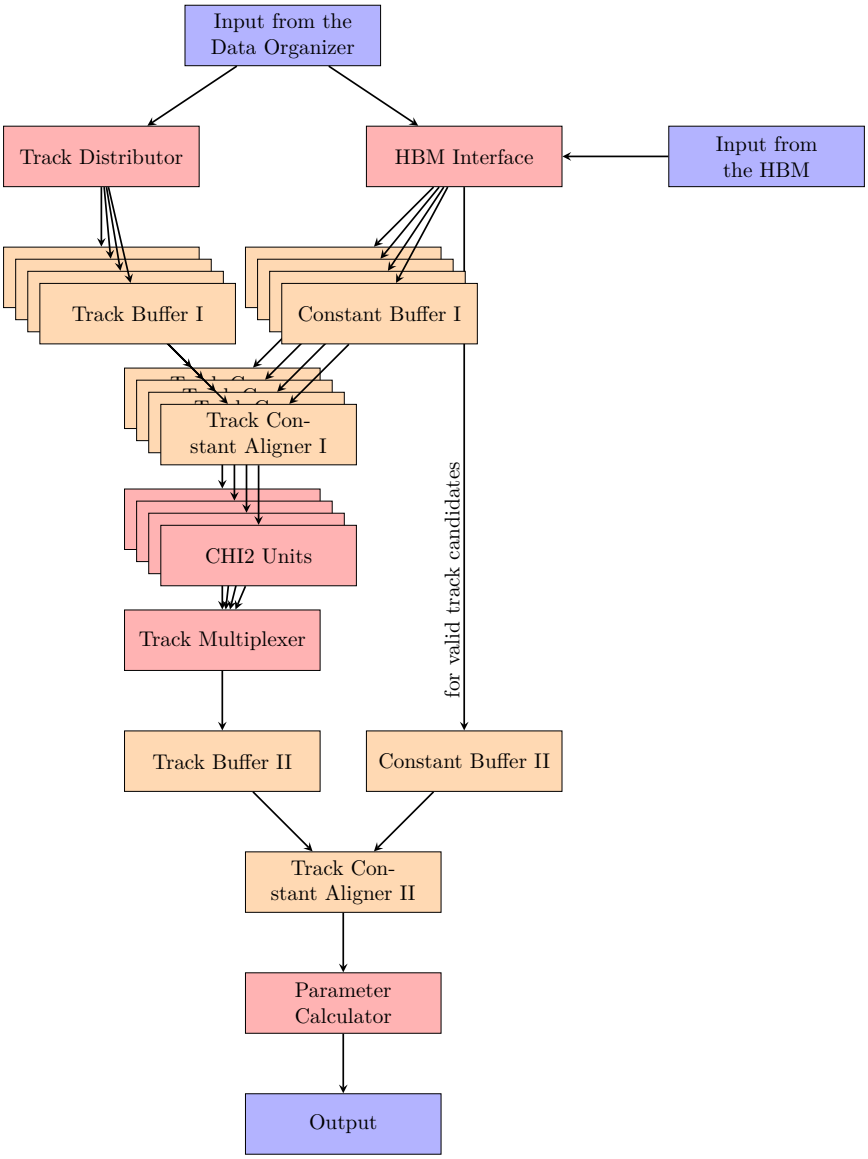


Figure 4: Data-Flow through the Track Fitter. Only connections used by *data*, i.e. tracks or constants are shown; connections that are solely used by requests, commands or status signals are omitted to increase readability.

The Track Fitter receives data packages that consist of a *road*, containing up to three hit clusters per layer*, the number of clusters per layer, the identifiers of the road, its event, and of the sector that contains the constants for calculating the goodness $\chi^2$ and the parameters of the track candidates from that road.

In Figure 4 the flow of the main data packages through the Track Fitter is visualized. The Track Distributor receives the entire data package from the Data Organizer, an entity outside of the Track Fitter (see Figure 2). It distributes the roads evenly between four channels and decodes all possible tracks from each road. The tracks and identifiers are then stored in the channel's Track Buffer I.

*The firmware is limited to three clusters so it can guarantee to match its required road-acceptance rate.

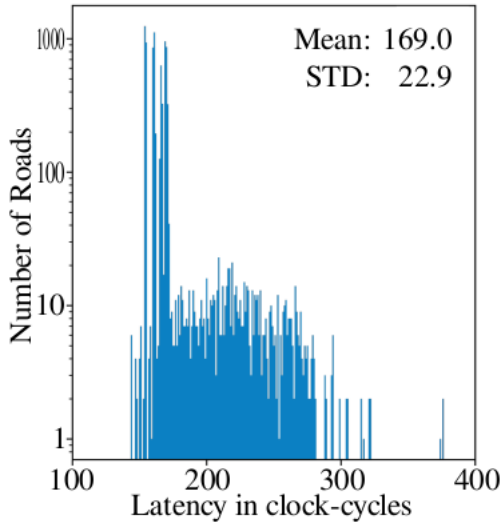


Figure 5: Histogram of the latency of the Track Fitter between receiving a track candidate and returning of the parameters (including the goodness of the fit). These data are taken from the PRM simulation.

The HBM Interface (HBM-IF) receives sector-identifiers and sends requests to the HBM for the constants from that sector. The HBM-IF receives the constants after an undetermined latency[†] and stores them internally. When the Track Buffer I receives data from the Track Distributor, it sends the sector-identifier to the HBM-IF to ask for the constants $\vec{S}_i$ and $h_i$ that are used in the goodness calculation, which are then stored in that channel's Constant Buffer I.

The Track Constant Aligner I checks the data stored in the upstream buffers to ensure that the tracks, which are sent to the CHI2-unit, are accompanied by the constants from the correct sector. There is no timeout but when the buffer is full, further constants are lost.

The CHI2 unit calculates the goodness of the track and applies a threshold cut. The track is sent to the Track Multiplexer if it passes the goodness test. The Track Multiplexer sends the tracks to the Track Buffer II and their sector-identifiers to the HBM-IF to request the constants necessary to calculate the track parameters.

The HBM-IF sends a second request to the HBM and later transmits the parameter constants to the Constant Buffer II. The Track Constant Aligner II performs the same operations as the Track Constant Aligner I and sends the tracks (and the identifiers and goodness) together with the correct parameter constants to the Parameter Calculator.

# 5 Results

Mentor Graphics' *Questa Sim* [12] has been used to simulate the firmware. Two simulation setups are examined: one for the entire PRM with the AM ASIC emulators and an HBM simulation model, the second one for the Track Fitter with an HBM emulator. The simulation runs with a frequency of 250 MHz, therefore the Track Fitter simulation is compliant with a road rate of 80 MHz. The Track Fitter simulation has a mean latency of 680 ns, dominated by the latency of the HBM (see Figure 5). That fulfills the road-rate and latency requirements of the Level 1 configuration of the ATLAS Phase-II Upgrade DAQ system (i.e. $\geq 65$ MHz [4] and $\leq 1$ µs respectively).

The PRM firmware successfully runs with a prototype Intel Stratix 10 FPGA. No tests have been performed with a frequency exceeding 100 MHz. To achieve the desired speed of 250 MHz on the prototype hardware, further optimization is necessary.

[†]99 % of constant-packages are returned to the HBM-IF within a latency of 850 ns [11].

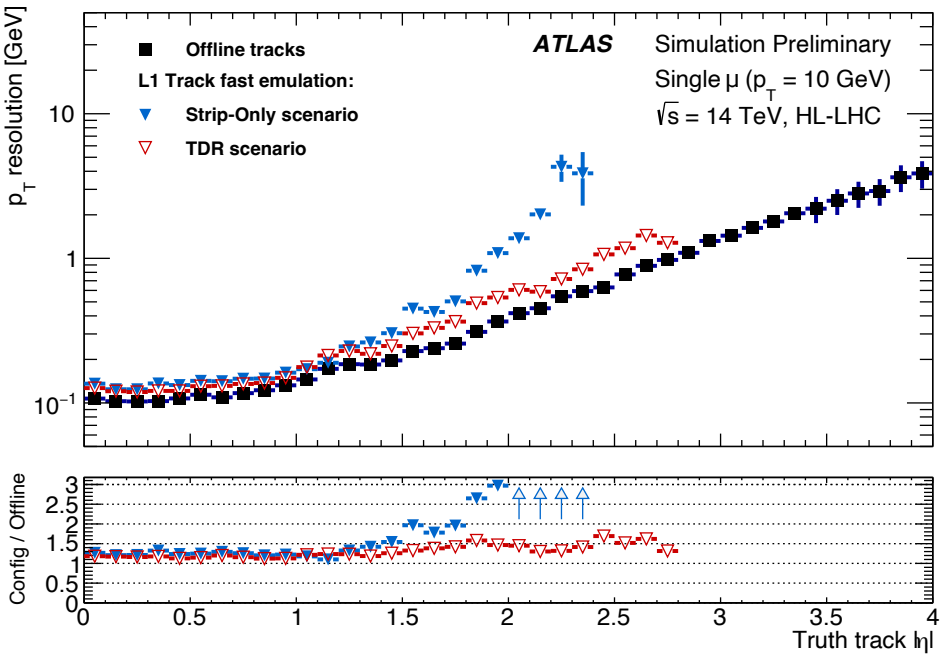


Figure 6: Top: Transverse momentum resolution of the software emulation of the HTT (red) and the offline simulation (black). Bottom: Relative transverse momentum resolution between those two simulations (red). These data refer to single muons with a transverse momentum of $p_{\mathrm{T}} = 10\,\mathrm{GeV}$ from pp-collisions with a center-of-mass energy of $\sqrt{s} = 14\,\mathrm{TeV}$ [13].

The Track Fitter calculates the fit parameters and the goodness of dummy data perfectly reliable within the precision of its number representation (single precision floats, downsized at the output to 16 bit signed fixed point numbers that use eight bits for the exponent). For a few data-samples, the fit parameters (and goodness) have been compared bit-by-bit to the expected values and good agreement was found. Due to the limited nature of available test data, an extensive measurement of fit performance has not been conducted but further studies are underway. Since the implementation in the Track Fitter is mathematically sound, the quality of the fits depends only on the quality of the linearization of the fit parameters and the size of the sectors (see also [9]). Software emulation studies have shown that the HTT is able to reconstruct 95 % of all tracks [13] with a momentum resolution that is not more than 75 % worse than the offline algorithm (see Figure 6). More information about these performance studies can be found in [13].

# 6 Conclusions

The linearized track fitting approach was proven to work in simulation and in prototype hardware. Even though the HTT project was canceled, the design presented here can be used for further developments and can possibly be implemented in a modified form as part of the ATLAS Event Filter System, see Figure 1. Investigations on the performance of the Track Fitter without the HBM and dedicated hardware are currently conducted.

## ACKNOWLEDGEMENTS

I am grateful to Dr. Alessandra Camplani and Kostas Axiotis for close and fruitful collaboration on the firmware development. I further appreciate the financial and educational support from the BMBF and the HighRR.

# References

[1] The ATLAS Collaboration, "Technical Design Report for the ATLAS Inner Tracker Pixel Detector", CERN LHCC-2017-021 ATLAS-TDR-030, June 2018.

[2] The ATLAS Collaboration, "Technical Design Report for the ATLAS Inner Tracker Strip Detector", CERN LHCC-2017-005 ATLAS-TDR-025, April 2017.

[3] The ATLAS Collaboration, "Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System", CERN-LHCC-2017-020 ATLAS-TDR-029, June 2018.

[4] The ATLAS Collaboration, "HTT ATLAS Electronics Specification for Pattern Recognition Mezzanine (PRM): WBS 1.3.1.3", December 2019.

[5] The ATLAS Collaboration, "HTT System Description and Overall Specification Document", ATL-COM-DAQ-2018-162, February 2019.

[6] Rainer Mankel, "Pattern Recognition and Event Reconstruction in Particle Physics Experiments", February 2004.

[7] Rudolf Frühwirth and Are Strandlie, "Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors", 2021.

[8] Alberto Stabile and the AM Design Team, "Phase-II Associative Memory ASIC Specifications (for the users)", March 2020.

[9] Hermann Kolanoski and Norbert Wermes, "Teilchendetektoren", 2016.

[10] Intel Corporation, "Intel Stratix 10 MX (DRAM system-in-Package) Device Overview", September 2020.

[11] Sebastian Dittmeier, "High Bandwidth Memory – Latency in Simulation Studies for 99.0 % and 99.9 % of Simulated Events", March 2020.

[12] Siemens Digital Industries Software, "The Questa Verification Solution", September 2021.

[13] The ATLAS Collaboration, "Performance Studies of Tracking-Based Triggering Using a Fast Emulation", ATL-DAQ-PUB-2023-001, February 2023.