Available online at www.sciencedirect.com
ScienceDirect
Nuclear Instruments and Methods in Physics Research A 566 (2006) 80–84
NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A
www.elsevier.com/locate/nima

Using FPGA coprocessor for ATLAS level 2 trigger application

Andrei Khomich*, Christian Hinkelbein, Andreas Kugel, Reinhard Männer, Matthias Müller

Institute of Computer Science V, University of Mannheim, B6, 23-29, 68131, Mannheim, Germany

Available online 30 May 2006

Abstract

Tracking has a central role in the event selection for the High-Level Triggers of ATLAS. It is particularly important to have fast tracking algorithms in the trigger system. This paper investigates the feasibility of using FPGA coprocessor for speeding up of the TRT LUT algorithm—one of the tracking algorithms for second level trigger for ATLAS experiment (CERN). Two realisations of the same algorithm have been compared: one in C++ and a hybrid C++/VHDL implementation. Using a FPGA coprocessor gives an increase of speed by a factor of two compared to a CPU-only implementation.

PACS: 29.85.+c; 29.40.Gx; 89.20.Ff

Keywords: High-energy physics; LHC; ATLAS; Trigger; Tracking; FPGA; Coprocessor

1. Introduction

ATLAS (A Toroidal LHC ApparatuS) [1] is one of the four detectors at the Large Hadron Collider (LHC) facility at the European Organization for Nuclear Research (CERN) which will start operation in 2007. It is a general purpose detector which will detect the products of proton–proton collisions with a center of mass energy of 14 TeV. From the inside to the outside, it consists of a pixel detector, a silicon strip detector and a transition radiation detector (TRT). These three tracking detectors are surrounded by electro-magnetic and hadronic calorimeters. The outermost sub-detector is the muon spectrometer.

The *pp* interaction that leads to the interesting physics process (referred to as the physics event) will be accompanied by several (~20) minimum bias interactions occurring simultaneously. The task of a trigger system is to select rare events and to suppress background events as efficiently as possible. The ATLAS trigger strategy foresees a reduction of the event rate at several levels: Level-1 (LVL1) and High-Level trigger (HLT) which in turn is subdivided on Level-2 (LVL2) and Event Filter (EF) [2]. The average execution time for LVL2 algorithms should be about 10 ms per event.

One of the most demanding tasks is the reconstruction of particle trajectories from the detector hits. To minimize processing time and to reduce the amount of transferred data only the information from the so-called regions of interest corresponding to areas where the LVL1 trigger found some interesting activity is analysed at Level-2. However, it is necessary to look at full subdetector in some special cases like B meson decays. Hadrons containing b quarks are pairwise produced and decay independently. One of the hadrons might decay semi-leptonically which could be triggered by e.g. an inclusive muon trigger. The muon from the decay of a B-particle does not indicate the direction of flight of the other B-hadron. For further selection, an unguided track search is necessary; this can be achieved by a track search in the full TRT [3]. Such a full scan requires a lot of computational power. This paper describes a proof of concept of the acceleration one of the LVL2 tracking algorithms for the Transition Radiation Tracker-Look-Up Table (TRT-LUT algorithm) with the help of a FPGA coprocessor.

The FPGA coprocessor—MPRACE (Multi-Purpose Reconfigurable Accelerator/Computing Engine) [4] developed at the University of Mannheim was used for this task. MPRACE is an FPGA-Coprocessor based on Xilinx

*Corresponding author.
E-mail address: khomich@ti.uni-mannheim.de (A. Khomich).

0168-9002/$ - see front matter
doi:10.1016/j.nima.2006.05.025

VirtexII FPGA and made as a 64 Bit/66 MHz PCI card. It is a universal coprocessor which is used for different applications.

The following conventions are used in the paper: the coordinate system has its origin at the interaction point. The z-axis is parallel to the beam direction and the radius r and the azimuthal angle ϕ are used to denote positions in the transverse plane.

2. Algorithm description

The TRT [5] provides tracking information and contributes to the electron identification. The TRT consists of a central barrel part and two end-cap sections. The sensitive elements are straw tubes with a single sense wire running down the centre. The barrel comprises 52 544 straws parallel to the beam axis in 73 layers; the end-caps comprise 319 488 radial straws in 18 multiplane wheels.

The TRT straws provide a two-dimensional position measurement, r–ϕ in the barrel and z–ϕ in the end-caps. Typically, the TRT provides 36 measurements on each track in the detector acceptance. This results in ~20 000 hits per event.

The following geometry has been used for simulation and reconstruction in the barrel part. The straws in each layer are arranged in a concentric cylinder. Within a layer all straws have equal distance in ϕ. There is a small shift in ϕ between layers.

The TRT-LUT [6] consists of a track candidate search followed by track fit performed to determine the track parameters. Since all the particle trajectories to search for can be calculated in advance, a histogramming method based on the Hough Transform is well suited for the initial track search in TRT. The Hough transform is a standard method in image analysis that allows recognition of global patterns in an image space by recognition of local patterns in a transformed space [7]. The algorithm is based on the idea that every hit in the three-dimensional detector image can belong to a number of possible (predefined) tracks characterised by different parameters. All such tracks (or roads) are stored in LUT. Thus, every hit increases the "probability" for the existence of these tracks by one (histogramming). The histogram for a single track consists of a "bow-tie" shaped region of bins with entries with a peak at the centre of the region as shown in Fig. 1. The bin at the peak of the histogram will, in the ideal case, contain all the hits from the track. The roads corresponding to the other filled bins share straws with the peak bin, and so contain sub-sets of the hits from the track. The histogram for a more complex event consists of a superposition of the entries from the individual tracks. The bins containing the complete set of points from each track can be identified as local maxima in the histogram. After a clean-up step followed by a fit, the final tracks are selected. The LUT-based Hough Transform algorithm for the TRT was implemented in C++ and integrated into software framework for ATLAS Trigger investigation. The algorithm was implemented in VHDL for the MPRACE board as well. This implementation was integrated into the same framework for performance study and comparison with C++ implementation.

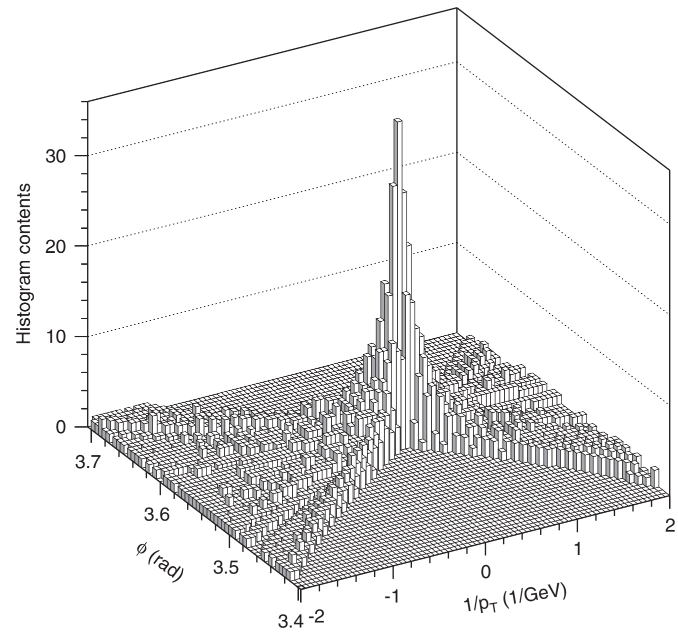

Fig. 1. Histogram due to a single muon in the barrel TRT.

The algorithm consists of the following steps:

Initial track finding: This uses a LUT-based Hough Transform to find potential track candidates. In the barrel, the Hough Transform is performed from (R, ϕ) space to $(\phi, 1/p_T)$ space. Each straw is assigned to ~120 roads (max. 130) and ~65 straws are assigned to each road.

Thresholding and local maximum finding: This selects potential track candidates and eliminates multiple reconstructed tracks.

Track splitting: Removes hits incorrectly assigned to a track, and splits tracks that have been erroneously merged. In this step the pattern of hits associated to a track candidate is analysed. If a potential track candidate contains N_{is} consecutive layers without a hit, the track is split into two separate candidates either side of the gap. Only one candidate with more than N_{thr} hits and with the first hit in the layer with the lowest radius is retained. For this step a "bin-ordered" LUT is constructed (each bin corresponds to road). The list of straws lying within the road is stored in the LUT.

Track fitting and final selection: Performs a least-square fit in the r–ϕ (barrel) or z–ϕ (end-caps) plane. The final track candidates are selected during this step.

Profiling of a C++ implementation of the TRT full scan algorithm shows that most of the computing time is spent in access of LUTs, incrementing of 8-bit numbers, and a