



CERN-LHCC-2013-007  
ATLAS-TDR-021-2013  
June, 2013

# ATLAS

Fast Tracker  
Technical Design Report





# Technical Design Report

## Fast Tracker (FTK)

Issue: 1  
Reference: CERN-LHCC-2013-007  
ATLAS-TDR-021  
Created: June 2013  
Last modified: July 2013  
Prepared by: ATLAS Collaboration

*CDS: <http://cdsweb.cern.ch/record/1552953/>  
TWiki: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/TriggerFTKPublicResults>*

## The ATLAS Institutes

### **Argentina**

*Universidad de Buenos Aires, Buenos Aires*

*Universidad Nacional de La Plata, La Plata*

### **Armenia**

*Yerevan Physics Institute, Yerevan*

### **Australia**

*University of Adelaide, Adelaide*

*Research Centre for High Energy Physics, Melbourne University, Melbourne*

*University of Sydney, School of Physics, Sydney*

### **Austria**

*Institut für Astro- und Teilchenphysik, University of Innsbruck, Innsbruck*

*Fachhochschule Wiener Neustadt (FHWN), Wiener Neustadt*

### **Azerbaijan Republic**

*Institute of Physics, Azerbaijan Academy of Sciences, Baku*

### **Republic of Belarus**

*Institute of Physics, National Academy of Sciences of Belarus, Minsk*

*National Centre for Particle & High Energy Physics, Minsk*

### **Brazil**

*Universidade Federal de Juiz de Fora, Universidade Federal do Rio De Janeiro, COPPE/EE/IF, Rio de Janeiro, Universidade Federal de Sao Joao del Rei and Universidade de Sao Paulo*

### **Canada**

*University of Alberta, Edmonton*

*University of Carleton, Carleton*

*University of Montreal, Group of Particle Physics, Montreal, Quebec*

*Department of Physics, McGill University, Montreal*

*Simon Fraser University, Burnaby, BC*

*Department of Physics, University of Toronto, Toronto*

*TRIUMF, Vancouver and York University, Toronto*

*Department of Physics, University of British Columbia, Vancouver*

*University of Victoria, Victoria*

### **CERN**

*European Laboratory for Particle Physics (CERN), Geneva*

### **Chile**

*Joint team from Pontificia Universidad Católica de Chile, Santiago and Universidad Técnica Federico Santa María, Valparaíso*

**China**

*Chinese cluster formed by IHEP Beijing, Nanjing, Shandong and Heifei*

**Colombia**

*Universidad Antonio Narino (UAN), Bogotá*

**Czech Republic**

*Palacký University, Olomouc*

*Academy of Sciences of the Czech Republic, Institute of Physics and Institute of Computer Science, Prague*

*Charles University in Prague, Faculty of Mathematics and Physics, Prague*

*Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Faculty of Mechanical Engineering, Prague*

**Denmark**

*Niels Bohr Institute, University of Copenhagen, Copenhagen*

**France**

*Laboratoire d'Annecy-le-Vieux de Physique de Particules (LAPP), CNRS-IN2P3, Annecy-le-Vieux*

*Laboratoire de Physique Corpusculaire, Université Blaise Pascal, CNRS-IN2P3, Clermont-Ferrand*

*Laboratoire de Physique Subatomique et de Cosmologie de Grenoble (LPSC), CNRS-IN2P3, Université Joseph Fourier, Grenoble*

*Centre de Physique de Particules de Marseille (CPPM), CNRS-IN2P3, Marseille*

*Laboratoire de l'Accélérateur Linéaire (LAL), CNRS-IN2P3, Orsay*

*Laboratoire de Physique Nucléaire et de Hautes Energies (LPNHE), Universités de Paris VI et VII, CNRS-IN2P3, Paris*

*Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA), DSM/IRFU, Centre d'Etudes de Saclay, Gif-sur-Yvette*

**Georgia**

*Institute of Physics of the Georgian Academy of Sciences and Tbilisi State University, Tbilisi*

**Germany**

*Physikalisches Institut, University of Bonn, Bonn*

*Deutsches Elektronen-Synchrotron (DESY), Hamburg and Zeuthen*

*TU Dortmund, Experimentelle Physik IV, Dortmund*

*Technical University Dresden, Dresden*

*Fakultät für Mathematik und Physik, Albert-Ludwigs-Universität, Freiburg*

*Justus-Liebig-Universität, Giessen*

*Fakultät für Physik, II. Physikalisches Institut, Georg-August-Universität, Göttingen*

*Ruprecht-Karls-Universität Heidelberg, Kirchhoff-Institut für Physik and Zentrales Institut für Technische Informatik (ZITI), Heidelberg*

*Institut für Physik, Humboldt Universität, Berlin*

*Institut für Physik, Universität Mainz, Mainz*

*Sektion Physik, Ludwig-Maximilians-Universität München, München*

*Max-Planck-Institut für Physik, München*

*Fachbereich Physik, Universität Siegen, Siegen*

*Fachbereich Physik, Bergische Universität, Wuppertal*

*Julius-Maximilians-University, Würzburg*

### **Greece**

*National Technical University of Athens, Athens*

*University of Athens, Athens*

*University of Thessaloniki, High Energy Physics Department and Department of Mechanical Engineering, Thessaloniki*

### **Israel**

*Department of Physics, Technion, Haifa*

*School of Physics, Tel Aviv University, Tel Aviv*

*Department of Particle Physics, The Weizmann Institute of Science, Rehovot*

### **Italy**

*Dipartimento di Fisica dell' Università di Bologna e I.N.F.N., Bologna*

*Dipartimento di Fisica dell' Università della Calabria e I.N.F.N., Cosenza*

*Laboratori Nazionali di Frascati dell' I.N.F.N., Frascati*

*Dipartimento di Fisica dell' Università di Genova e I.N.F.N., Genova*

*Dipartimento di Fisica dell' Università di Lecce e I.N.F.N., Lecce*

*Dipartimento di Fisica dell' Università di Milano e I.N.F.N., Milano*

*Dipartimento di Scienze Fisiche, Università di Napoli 'Federico II' e I.N.F.N., Napoli*

*Dipartimento di Fisica Nucleare e Teorica dell' Università di Pavia e I.N.F.N., Pavia*

*Dipartimento di Fisica dell' Università di Pisa e I.N.F.N., Pisa*

*Dipartimento di Fisica dell' Università di Roma I 'La Sapienza' e I.N.F.N., Roma*

*Dipartimento di Fisica dell' Università di Roma II 'Tor Vergata' e I.N.F.N., Roma*

*Dipartimento di Fisica dell' Università di Roma III 'Roma Tre' e I.N.F.N., Roma*

*Dipartimento di Fisica dell' Università di Udine e I.N.F.N., Udine, I.C.T.P. e I.N.F.N., Trieste*

### **Japan**

*Hiroshima Institute of Technology, Hiroshima*

*KEK, High Energy Accelerator Research Organization, Tsukuba*

*Kobe University, Kobe*

*Department of Physics, Kyoto University, Kyoto*

*Kyoto University of Education, Kyoto*

*Kyushu University, Kyushu*

*Nagasaki Institute of Applied Science, Nagasaki*

*Nagoya University, Nagoya*

*Faculty of Science, Okayama University, Okayama*

*Osaka University, Osaka*

*Faculty of Science, Shinshu University, Matsumoto*

*International Center for Elementary Particle Physics and Department of Physics, The University of Tokyo*

*Tokyo Institute of Technology, Tokyo*

*Physics Department, Tokyo Metropolitan University, Tokyo*

*Institute of Physics, University of Tsukuba, Tsukuba*

*Waseda University, Tokyo*

#### **Morocco**

*Faculté des Sciences Ain Chock, Université Hassan II, Casablanca, Université Mohamed Premier et LPTM, Oujda, Université Cadi Ayyat et LPHEA, Marrakech, CNESTEN et Université Mohamed V, Rabat*

#### **Netherlands**

*FOM - Institute SAF NIKHEF and University of Amsterdam/NIKHEF, Amsterdam*

*Radboud University Nijmegen and NIKHEF, Nijmegen*

#### **Norway**

*University of Bergen, Bergen*

*University of Oslo, Oslo*

#### **Poland**

*Institute of Nuclear Physics (IFJ PAN), Polish Academy of Sciences, Cracow*

*Faculty of Physics and Applied Computer Science, AGH University of Science and Technology and Marian Smoluchowski Institute of Physics, Jagiellonian University, Cracow*

#### **Portugal**

*Laboratorio de Instrumentacao e Física Experimental de Partículas (LIP), in collaboration with: University of Lisboa, University of Coimbra, University Católica-Figueira da Foz, University Nova de Lisboa, Lisbon and University of Granada*

#### **Romania**

*National Institute of Physics and Nuclear Engineering, Institute of Atomic Physics, Bucharest, ITIM, Cluj Napoca, West University, Timisoara and University Politehnica Bucharest*

**Russia**

*Institute for Theoretical and Experimental Physics (ITEP), Moscow*

*P.N. Lebedev Institute of Physics, Moscow*

*Moscow Engineering & Physics Institute (MEPhI), Moscow*

*Moscow State University, Moscow*

*Budker Institute of Nuclear Physics (BINP), Novosibirsk*

*State Research Center of the Russian Federation - Institute for High Energy Physics (IHEP), Protvino*

*Petersburg Nuclear Physics Institute (PNPI), St. Petersburg*

**JINR**

*Joint Institute for Nuclear Research, Dubna*

**Serbia**

*Institute of Physics, University of Belgrade and Vinca Institute of Nuclear Sciences, Belgrade*

**Slovak Republic**

*Bratislava University, Bratislava, and Institute of Experimental Physics of the Slovak Academy of Sciences, Kosice*

**Slovenia**

*Jožef Stefan Institute and Department of Physics, University of Ljubljana, Ljubljana*

**South Africa**

*University of Cape Town, University of Johannesburg (UJ) and University of the Witwatersrand (WITS), Johannesburg*

**Spain**

*Institut de Física d'Altes Energies (IFAE), Universitat Autònoma de Barcelona, Bellaterra (Barcelona)*

*Physics Department, Universidad Autònoma de Madrid, Madrid*

*Instituto de Física Corpuscular (IFIC), Centro Mixto UVEG-CSIC, Valencia and Instituto de Microelectrónica de Barcelona, Bellaterra (Barcelona)*

**Sweden**

*Fysiska institutionen, Lunds universitet, Lund*

*Royal Institute of Technology (KTH), Stockholm*

*Stockholm University, Stockholm*

*University of Uppsala, Department of Physics and Astronomy, Uppsala*

**Switzerland**

*University of Bern, Albert Einstein Center for Fundamental Physics, Laboratory for High Energy Physics, Bern*

*Section de Physique, Université de Genève, Geneva*

**Taipei**

*Academia Sinica, Taipei*

**Turkey**

*Department of Physics, Ankara University, Gazi University and TOBB ETU, Ankara*

*Department of Physics, Bogazici University, Dogus University and Gaziantep University, Istanbul*

**United Kingdom**

*School of Physics and Astronomy, The University of Birmingham, Birmingham*

*University of Sussex, Brighton*

*Cavendish Laboratory, University of Cambridge, Cambridge*

*University of Warwick, Coventry*

*School of Physics & Astronomy, University of Edinburgh, Edinburgh*

*Department of Physics and Astronomy, University of Glasgow, Glasgow*

*Physics Department, Lancaster University, Lancaster*

*University of Liverpool, Liverpool*

*Department of Physics, Queen Mary and Westfield College, University of London, London*

*Department of Physics, Royal Holloway, University of London, Egham*

*Department of Physics and Astronomy, University College London, London*

*Department of Physics and Astronomy, University of Manchester, Manchester*

*Department of Physics, Oxford University, Oxford*

*Rutherford Appleton Laboratory, Science and Technology Facilities Council, Didcot*

*Department of Physics, University of Sheffield, Sheffield*

**United States of America**

*State University of New York at Albany, New York*

*Argonne National Laboratory, Argonne, Illinois*

*University of Arizona, Tucson, Arizona*

*Department of Physics, The University of Texas at Arlington, Arlington, Texas*

*Lawrence Berkeley National Laboratory and University of California, Berkeley, California*

*Physics Department of the University of Boston, Boston, Massachusetts*

*Brandeis University, Department of Physics, Waltham, Massachusetts*

*Brookhaven National Laboratory (BNL), Upton, New York*

*University of Chicago, Enrico Fermi Institute, Chicago, Illinois*

*Nevis Laboratory, Columbia University, Irvington, New York*

*University of Texas at Dallas, Dallas, Texas*

*Department of Physics, Duke University, Durham, North Carolina*

*Department of Physics, Hampton University, Virginia*

*Department of Physics, Harvard University, Cambridge, Massachusetts*

*Indiana University, Bloomington, Indiana*

*Iowa State University, Ames, Iowa*  
*University of Iowa, Iowa City, Iowa*  
*University of California, Irvine, California*  
*Louisiana Tech University, Louisiana*  
*University of Massachusetts, Amherst, Massachusetts*  
*Massachusetts Institute of Technology, Department of Physics, Cambridge, Massachusetts*  
*Michigan State University, Department of Physics and Astronomy, East Lansing, Michigan*  
*University of Michigan, Department of Physics, Ann Arbor, Michigan*  
*Department of Physics, New Mexico University, Albuquerque, New Mexico*  
*Department of Physics, New York University, New York*  
*Northern Illinois University, DeKalb, Illinois*  
*Ohio State University, Columbus, Ohio*  
*Department of Physics and Astronomy, University of Oklahoma*  
*Oklahoma State University, Oklahoma*  
*University of Oregon, Eugene, Oregon*  
*Department of Physics, University of Pennsylvania, Philadelphia, Pennsylvania*  
*University of Pittsburgh, Pittsburgh, Pennsylvania*  
*Institute for Particle Physics, University of California, Santa Cruz, California*  
*SLAC National Accelerator Laboratory, Stanford, California*  
*Physics Department, Southern Methodist University, Dallas, Texas*  
*State University of New York at Stony Brook, New York*  
*Tufts University, Medford, Massachusetts*  
*High Energy Physics, University of Illinois, Urbana, Illinois*  
*Department of Physics, Department of Mechanical Engineering, University of Washington, Seattle, Washington*  
*Department of Physics, University of Wisconsin, Madison, Wisconsin*  
*Yale University, New Haven, Connecticut*

# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
<b>2</b>	<b>System Overview</b>	<b>12</b>
2.1	Functional overview . . . . .	12
2.1.1	Introduction . . . . .	12
2.1.2	The FTK architecture . . . . .	13
2.2	Physical overview . . . . .	16
2.2.1	Physical description of the system . . . . .	16
<b>3</b>	<b>Performance</b>	<b>17</b>
3.1	Tracking Performance . . . . .	18
3.1.1	FTK Performance in Single Particle Samples . . . . .	18
3.1.2	FTK Performance in High-Luminosity Samples . . . . .	25
3.2	Event Object Performance . . . . .	26
3.2.1	Lepton Isolation . . . . .	26
3.2.2	b-jet tagging . . . . .	27
3.2.3	Primary Vertex Finding . . . . .	41
3.2.4	Hadronic taus . . . . .	46
3.3	Case Study: Higgs to $\tau\tau$ channel improvements . . . . .	53
3.4	Discussion of selected FTK use cases . . . . .	53
<b>4</b>	<b>Hardware</b>	<b>59</b>
4.1	The FTK System: a data driven pipeline . . . . .	59
4.1.1	Data flow between boards and the Hold mechanism . . . . .	60
4.1.2	The numbers of roads and fits: pattern bank optimization . . . . .	61
4.1.3	Internal monitoring and diagnostics: the Spy Buffer system . . . . .	64
4.2	Dual-output HOLA . . . . .	65
4.3	FTK Input Mezzanine (FTK_IM) . . . . .	66
4.3.1	Hardware description . . . . .	67
4.3.2	Clustering algorithms . . . . .	68
4.3.3	Hardware status . . . . .	68
4.4	Data Formatter . . . . .	69
4.4.1	Introduction . . . . .	69
4.4.2	The Data Formatter System . . . . .	69
4.4.3	Bandwidth Requirement Estimation . . . . .	73
4.5	Processor Unit AUX . . . . .	74
4.5.1	AUX functionality . . . . .	74
4.5.2	Data rates . . . . .	77
4.5.3	Input FPGAs . . . . .	77
4.5.4	Data Organizer . . . . .	78
4.5.5	Track Fitter . . . . .	79
4.5.6	First Stage Hit Warrior . . . . .	81
4.5.7	Wild-Card Processing . . . . .	82
4.6	Associative Memory System . . . . .	82
4.6.1	The AMBFTK . . . . .	82
4.6.2	The LAMBFTK mezzanine . . . . .	83
4.6.3	The AM chip . . . . .	85

4.6.4	Test of the integrated AM system . . . . .	87
4.6.5	The evolution of the full FTK AM system . . . . .	88
4.6.6	Voltage and power for the AM chip core . . . . .	93
4.7	Second Stage Board . . . . .	93
4.7.1	Design features . . . . .	93
4.7.2	Functional Layout, Interconnections, and Physical Design . . . . .	94
4.7.3	Input Data Flow . . . . .	95
4.7.4	Primary Functional Elements . . . . .	95
4.7.5	Output Data Flow . . . . .	102
4.8	FTK Level-2 Interface Crate (FLIC) . . . . .	102
4.8.1	FLIC Functionality . . . . .	102
4.8.2	FLIC Design . . . . .	103
4.8.3	FLIC Data Transmission . . . . .	105
4.9	Hardware infrastructure . . . . .	105
4.9.1	Power consumption . . . . .	106
4.9.2	Cooling . . . . .	106
4.9.3	Detector Control System (DCS) . . . . .	106
4.10	Timing . . . . .	107
4.10.1	FTK timing results at 69 pile-up . . . . .	108
<b>5</b>	<b>Simulation</b>	<b>108</b>
5.1	Simulation description . . . . .	110
5.2	Simulation setup and configuration files . . . . .	110
5.2.1	Internal configuration files . . . . .	110
5.2.2	Production of sectors and constants . . . . .	111
5.2.3	Production of patterns . . . . .	113
5.2.4	Configuration file distribution for TDR simulation . . . . .	113
5.3	Evolution and integration with the ATLAS production system . . . . .	114
5.4	Use of new computing technology . . . . .	117
<b>6</b>	<b>Online software</b>	<b>117</b>
6.1	Configuration and control . . . . .	117
6.1.1	Configuration . . . . .	117
6.1.2	Control . . . . .	118
6.2	Monitoring . . . . .	119
<b>7</b>	<b>Vertical Slice Performance</b>	<b>119</b>
7.1	Dual-output HOLA tests . . . . .	120
7.2	Vertical slice test . . . . .	120
7.3	Next steps . . . . .	121
<b>8</b>	<b>Phase-II Compatibility</b>	<b>122</b>
8.1	Baseline upgrade path . . . . .	123
8.1.1	Input to the Fast Tracker and the Data Formatter system . . . . .	123
8.1.2	The Processing Units . . . . .	124
8.1.3	Second stage board, and FLIC . . . . .	124
8.1.4	Brief summary . . . . .	125
8.2	AMchip R&D . . . . .	125
8.3	ATCA-based upgrade path with full replacement . . . . .	125

<b>9</b>	<b>Project Management</b>	<b>126</b>
9.1	Management Structure . . . . .	127
9.1.1	Institute Board . . . . .	127
9.1.2	Project Leader and Technical Coordinator . . . . .	127
9.1.3	Management Board . . . . .	128
<b>10</b>	<b>Cost and Funding</b>	<b>128</b>
<b>11</b>	<b>Sharing of Work and Responsibilities among Institutions</b>	<b>128</b>
<b>12</b>	<b>Schedule and Milestones</b>	<b>130</b>
<b>13</b>	<b>Appendix I: AM chip design</b>	<b>131</b>
<b>14</b>	<b>Appendix II: AM cooling tests</b>	<b>134</b>
<b>15</b>	<b>Glossary</b>	<b>136</b>
<b>16</b>	<b>The ATLAS Collaboration</b>	<b>138</b>
<b>17</b>	<b>Acknowledgements</b>	<b>155</b>

# 1 Introduction

Run I of the LHC [1] has already taught us much about physics at the TeV scale. First, a Higgs Boson has been found at a mass of approximately 125 GeV [2]. It has been observed in bosonic decay modes, however there is not yet sufficient evidence to prove that it also couples to fermions. The Higgs fermion measurements depend critically on the  $\tau$  and  $b$ -decay modes, the highest rate fermion decay channels for a Higgs of this mass. However, these signatures suffer from large QCD backgrounds and low trigger efficiency. The second lesson is that if new physics beyond the Standard Model exists at the electroweak scale, it is not minimal or near minimal SUSY at a low mass scale. The Run I analyses already probe mass scales of supersymmetric quarks and gluinos of over a TeV [3]. The limits on new heavy resonances also far exceed 1 TeV [4], suggesting that either the new physics is at even higher mass scales, or that it is hiding in experimentally hard-to-reach locations. Light  $\tilde{t}$ ,  $\tilde{\tau}$  and  $\tilde{b}$  searches feature final states with heavy quarks and leptons and have the weakest limits at well below a TeV. These supersymmetric heavy fermions are often difficult to trigger on and reconstruct experimentally because they lead to moderate momentum  $b$ -jets,  $\tau$ -leptons,  $t$ -quarks and missing energy. Many alternative models have privileged roles for third generation particles, but multi- $b$  and multi- $\tau$  final states are particularly challenging to select in the trigger because of the large QCD multi-jet rates.

Given these observations, the goals of the LHC physics program in Run II and beyond will be to precisely measure all possible branching ratios of the new Higgs particle and continue the search for new physics, maintaining efficiency for experimentally challenging signatures such as moderate momentum heavy particles. In order to achieve these goals, the trigger must be as flexible as possible, with a particular emphasis on good  $\tau$  and  $b$  selection.

The high instantaneous luminosity expected for the Phase 1 upgrade [5] to the LHC will pose challenges to achieving these goals for the trigger system. The existing ATLAS trigger system [6–8], consisting of a hardware-based Level 1 and a CPU-based High Level Trigger (HLT) with its Level 2 and Event Filter, was designed to work well at the LHC design luminosity. However after the planned luminosity upgrade, the detector environment will be complicated by the increase in detector activity arising from many simultaneous interactions. Additionally, the proposed upgrades to the hardware Level 1 Trigger [9] will allow an increased rate into the HLT. Because of its fine resolution and granularity, tracking information is critical for distinguishing which events triggered by the Level 1 should be kept for further processing. However, extensive tracking in such environments [10] is prohibitively expensive in terms of processing time per event or computing cores needed. Therefore, it is used sparingly for specific regions of interest (ROI) which have already been identified as potentially interesting by the Level-1 trigger and for full-event tracking at low rates ( $\approx$ few kHz). This approach has limitations in several cases. First, there is a limit to either the number or size of ROIs processed by the HLT, which forces additional non-tracking cuts to be applied, resulting in reduced efficiency or higher thresholds for the objects considered. Second, there are cases where global event information, such as the location of the hard interaction vertex or number of primary vertices in the event, are useful for object selections or corrections to the other detector quantities. An example of the latter can be seen in a simulated sample of  $WH$  events with leptonic  $W$  decay and  $H \rightarrow b\bar{b}$  at a mean pile-up of 69. With the standard jet finding algorithm, correction of the energy to the hadron scale, and a correction for the mean pile-up, there are on average 21 (11) jets with  $p_T > 25$  (50) GeV and  $|\eta| < 2.5$ <sup>1</sup>. By using tracking to select those jets pointing to the hard scattering vertex (requiring that more than 50% of the jet's track  $\Sigma p_T$  comes from tracks pointing to the primary interaction vertex in the event), the average number of jets is reduced to 3.8 (2.8).

We propose to build a system of electronics, the Fast TracKer or FTK, which will do global track

---

<sup>1</sup>ATLAS uses a right-handed coordinate system with its origin at the nominal interaction point (IP) in the centre of the detector and the  $z$ -axis coinciding with the axis of the beam pipe. The  $x$ -axis points from the IP to the centre of the LHC ring, and the  $y$ -axis points upward. Cylindrical coordinates  $(r, \phi)$  are used in the transverse plane,  $\phi$  being the azimuthal angle around the beam pipe. The pseudorapidity is defined in terms of the polar angle  $\theta$  as  $\eta = -\ln(\tan(\theta/2))$ .

reconstruction after each level-1 trigger to enable the level-2 trigger to have early access to tracking information. The ATLAS Inner Detector [6, 11] is shown in Fig. 1. FTK will use data from the pixel and semiconductor tracker (SCT) detectors as well as the new Insertable B-Layer (IBL) pixel detector [12].

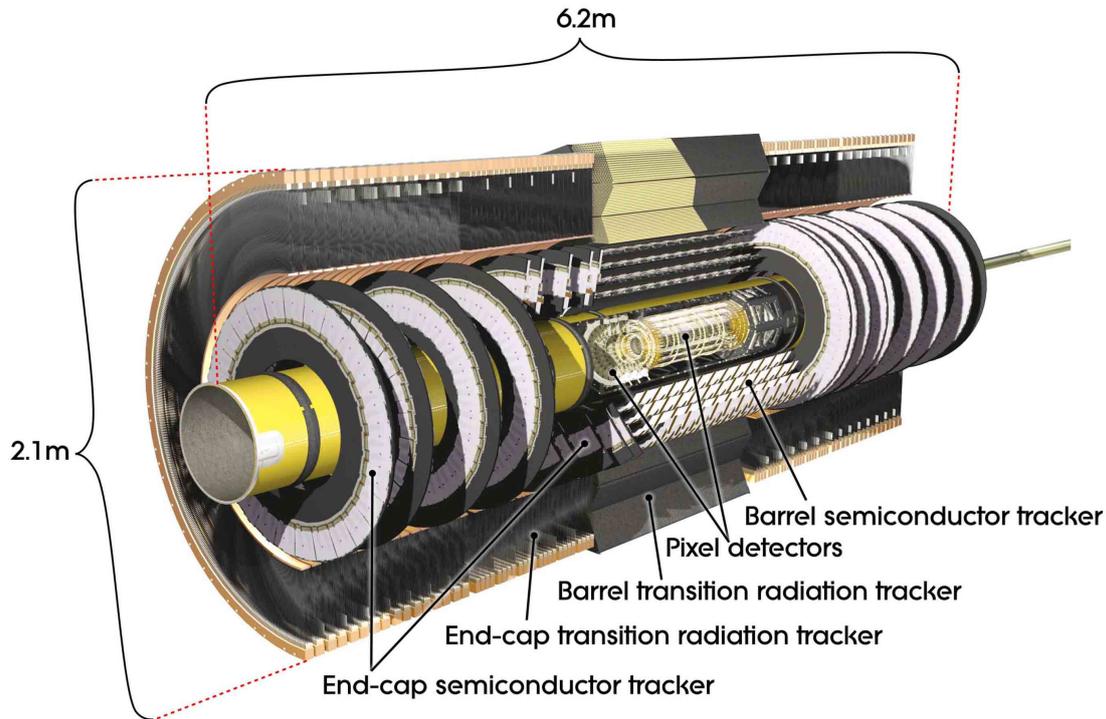


Figure 1: The ATLAS Inner Detector used through the 2012 data run. It consists of barrel and end-cap pixel and semiconductor trackers (SCT) as well as transition radiation trackers (TRT).

FTK will move track reconstruction into a hardware system with massively parallel processing that produces global track reconstruction with good resolution shortly after the start of level-2 processing. FTK is based on the very successful CDF Silicon Vertex Trigger (SVT) [13]. FTK tracks, freed from the CPU constraints of Level-2 tracking, will be an important tool box for the ATLAS HLT, allowing it to circumvent some of these limitations and have improved event selection.

A system overview is presented in section 2. FTK performance for individual tracks and physics objects is summarized in section 3. The hardware is described in section 4, FTK simulation is presented in section 5, and the software needed to operate the system is discussed in section 6. Section 7 describes the vertical slice that operated at Point-1 during the end of the 2012 run, while section 8 considers the future of FTK in Phase-II [14]. Project management is described in section 9, and the cost and funding of the system is considered in section 10. Institutional responsibilities are described in section 11, with the schedule and milestones presented in section 12. A glossary is provided at the end of the document.

## 2 System Overview

### 2.1 Functional overview

#### 2.1.1 Introduction

FTK is an electronics system that rapidly finds and reconstructs tracks in the inner-detector pixel and SCT layers for every event that passes the level-1 trigger. It uses 12 logical layers (4 pixel including

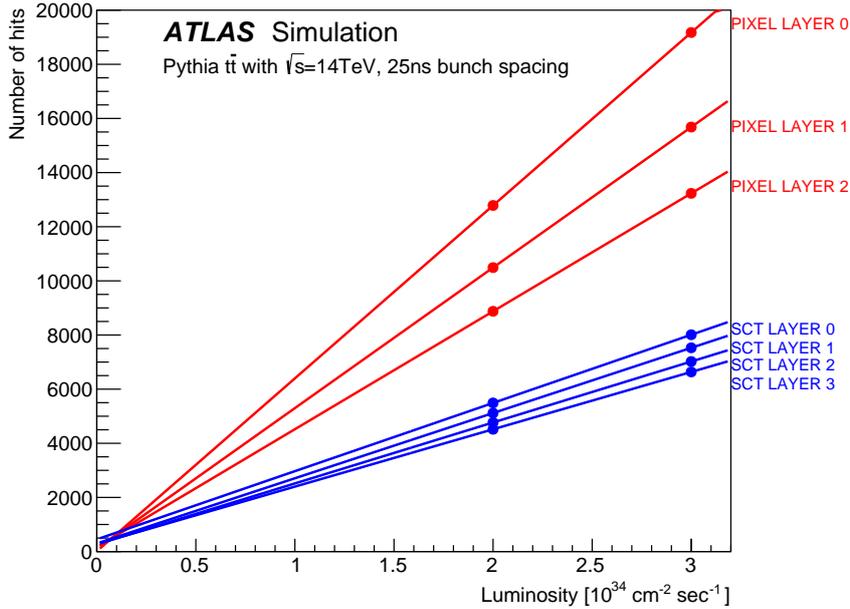


Figure 2: The number of hits in the barrel region for each silicon layer as a function of luminosity. For an SCT layer, the number of hits is the sum of those on the axial and stereo sides.

IBL; 8 SCT for the axial and stereo sides of the 4 physical layers) over the full rapidity range covered by the barrel and the disks<sup>2</sup>. Dual-output HOLAs, which replace the existing HOLA output mezzanine cards in the pixel and SCT Read Out Drivers (ROD), provide FTK with an identical copy of the silicon data being sent to the DAQ Read Out System (ROS) input buffers. FTK receives the hits at full rate as they are sent from the RODs following a level-1 trigger. After processing, FTK fills ROSs with the helix parameters and hits for all tracks with  $p_T$  above a minimum value, typically 1 GeV. The level-2 processors can request the track information in a Region of Interest or in the entire detector.

Handling the data flow at a luminosity of  $3 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  is a serious design challenge. Figure 2 shows the number of hits in the barrel region for each silicon layer as a function of luminosity. Note that the  $y$  intercept, which is barely visible above the origin, represents all of the hits from the hard scattering process!

### 2.1.2 The FTK architecture

To deal with the large input data rate as well as the large number of track candidates due to the hit combinatorics at high luminosity, the FTK is highly parallel with the system segmented into  $64 \eta - \phi$  towers, each with its own pattern recognition hardware and track fitters.

FTK operates in two stages. In the first stage, 8 of the 12 silicon layers are used to perform pattern recognition and do the initial track fitting. Pattern recognition is carried out by a dedicated system called the Associative Memory (AM) [15] which finds track candidates in coarse resolution roads. For each road containing silicon hits on all layers or all except one, the track parameters are calculated using the full resolution hits. Hit as used here refers to a particle's energy deposition cluster centroid. Each combination of one hit per logical layer within the road is fit to determine a goodness of fit.

<sup>2</sup>For the performance studies presented in section 3, 11 logical layers are used since full ATLAS simulation samples with the new IBL pixel layer are not yet available. The addition of the IBL should improve FTK performance. This will be addressed in the forthcoming ATLAS TDAQ TDR.

Track fitting is done rapidly by replacing a helical fit with a simple calculation that is linear in the local hit position in each silicon layer. The terms “track fitting” and “track fitter” are used throughout this document. However instead of an actual fit being done, the helix parameters and  $\chi^2$  components are estimated from the linear calculation (see sections 4.5.5 and 5.2.2). The calculation is a set of scalar products of the hit coordinates and pre-calculated constants that take into account the detector geometry and alignment. The helix parameter resolution obtained using this method is nearly as good as that obtained with a helical fit if the region covered by a set of constants is small. In FTK, that region, called a sector, consists of a single silicon module in each layer, typically a few centimeters wide.

Tracks passing a relatively loose  $\chi^2/\text{ndof} < 6$  cut are kept. If there are hits in all layers and the  $\chi^2$  fails the cut but is not extremely large, the track is refit a number of times with the hit on one of the layers dropped each time. This “majority recovery” allows for the loss of a single hit due to detector inefficiency with a random hit picked up instead. If two tracks in a road passing the  $\chi^2$  test have fewer than 6 non-matching hit coordinates, they are considered to be duplicate tracks and the best track is retained based on the  $\chi^2$  and the numbers of pixel and SCT layers that have a hit.

Tracks from the first stage pass to the second stage where they are extrapolated into the 4 logical silicon layers not used in stage 1. Nearby hits in those layers are found and the tracks are refit using the hits in all 12 layers. Track candidates are allowed to have 2 missing hits, but not a missing hit in both a pixel layer and an SCT layer. After a tighter  $\chi^2/\text{ndof} < 4$  cut is applied, duplicate tracks are removed using the same non-shared hit requirement as in the first stage. In this stage, tracks available for duplicate track removal are not restricted to a common road.

The Associative Memory rapidly carries out what is usually the most CPU intensive aspect of tracking by employing massive parallelism - processing hundreds of millions of roads nearly simultaneously as the silicon data pass through FTK. The road width must be optimized. If it is too narrow, the needed size of the Associative Memory and hence the cost of the system is too large. If roads are too wide, the load on the track fitters can become excessive due to the large number of fake roads and uncorrelated hits within a road. This increases both the number of roads the track fitter must process and the number of fits within the road due to the hit combinatorics. To make maximal use of the available pattern space, FTK employs a “variable resolution” feature. In each pattern, the width can be varied independently layer by layer.

Figure 3 shows a functional sketch of FTK, which is FPGA based with the exception of one specially designed custom chip for the associative memory.

The large amount of processing required to do global tracking at a 100 kHz level-1 trigger rate at luminosities up to  $3 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$  necessitates organizing FTK as a set of independent engines, each working on a different region of the silicon tracker. There are 64  $\eta - \phi$  towers, 16 in  $\phi$  by 4 in  $\eta$ . This segmentation generates some inefficiency at tower boundaries that is removed by allowing an overlap in these regions. The overlap in  $\phi$  covers the range of track curvature and multiple scattering, while that in  $\eta$  also takes into account the size of the beam’s luminous region in  $z$ .

The pixel and strip data are transmitted from the RODs on S-LINK fibers and received by the Data Formatters (DF). DF mezzanine cards, the FTK.IM, perform cluster finding, two-dimensional for the pixel layers. Clusters consist of pixels connected side-by-side or diagonally. Clusters are truncated at 4 pixels in the  $\phi$  direction and 5 pixels in the  $z$  or  $r$  direction for the barrel and endcap, respectively. The maximum size is programmable up to  $7 \times 9$  pixels. The DFs reorganize the data into the FTK  $\eta - \phi$  towers and transmit the cluster centroids to the core crates containing the pattern recognition and track fitting hardware<sup>3</sup>. The barrel layers and the end-cap disks are grouped into logical layers so that there are 12 layers over the full rapidity range (see Fig. 4). For the disks, this means assigning different disks to the same logical layer. The hits in each logical layer are received in a core crate and sent to the

---

<sup>3</sup>Once the raw silicon data are clustered in the FTK input, the cluster centroids are passed through the system and processed. As noted previously, the term “hit” in the FTK system means a cluster centroid.

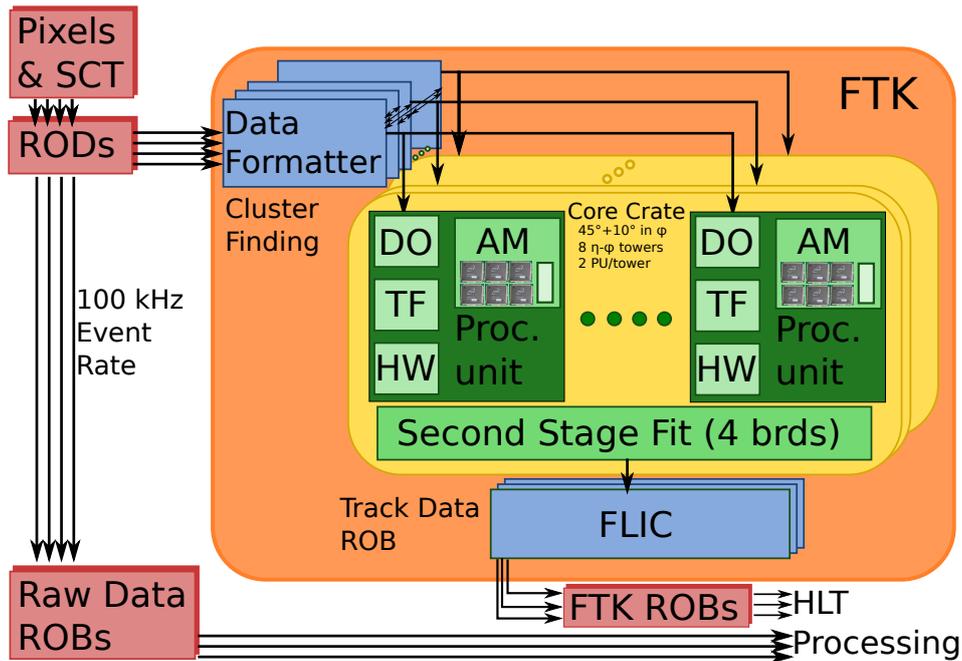


Figure 3: Functional sketch of FTK. AM is the Associative Memory, DO is the Data Organizer, FLIC is the FTK-to-Level-2 Interface Crate, HW is the Hit Warrior, ROB is the ATLAS Read Out input Buffer, ROD is a silicon detector Read Out Driver, and TF is the Track Fitter. Second Stage Fit is referred to as the Second Stage Board elsewhere in the document.

Data Organizers (DO). The hits are also converted to a coarser resolution (Super-Strips or SS) that is appropriate for pattern recognition, with the Super-Strips sent to both the Associative Memory and the Data Organizers.

The Data Organizers are smart databases where full resolution hits are stored in a format that allows rapid access based on the pattern recognition road ID and then retrieved when the Associative Memory finds roads with the requisite number of hits.

The Associative Memory boards contain a very large number of preloaded patterns corresponding to the possible combinations of a superstrip in each of 8 silicon layers for real tracks. Currently 3 pixel layers (the B-layer and the 2 outer pixel layers) and 5 SCT layers (the 4 axial layers and 1 stereo layer) are used for pattern recognition. The patterns are determined in advance from a full ATLAS simulation of single tracks. The AM is a massively parallel system in that all patterns see each silicon hit nearly simultaneously. As a result, pattern recognition in FTK is complete shortly after the last hit has been transmitted from the silicon RODs. When a pattern has 7 or 8 layers hit (such patterns that contain track candidates are referred to as roads), the AM sends the road ID number back to the Data Organizer which fetches the associated full resolution hits and sends them and the road number to the Track Fitter (TF). The TF has access to a set of constants for each detector sector which consists of a physical silicon module in each layer. Because a sector is quite narrow, the TF can provide high resolution helix parameters using a simple calculation that is linear in the position of the hit in each layer. Fitting a track is thus extremely fast since it consists of a series of integer multiply-and-accumulate steps. In a modern FPGA, approximately  $10^9$  track candidates can be fit per second. Following fitting, duplicate track removal (the Hit Warrior or HW function) is carried out among those 8-layer tracks in a road that pass the  $\chi^2$  cut.

The Second Stage Boards (SSB) receive from the Data Formatters the cluster centroids in the 4 layers

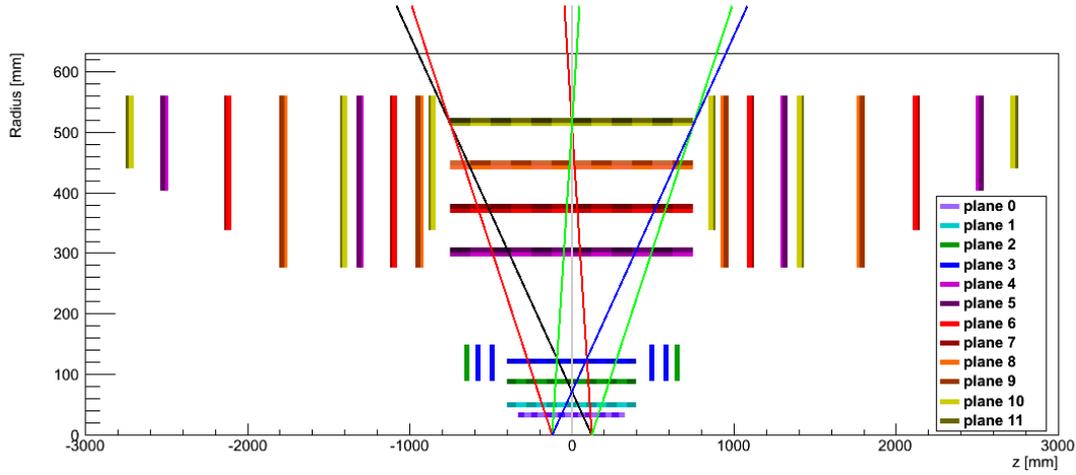


Figure 4: The assignment of barrel layers and end-cap disks to FTK logical layers. Layers 0-3 are pixels; the rest are SCT. The division into 4  $\eta$  regions with the appropriate overlap is indicated by the thin colored lines: two endcap regions, one to the left of the black line and the other to the right of the blue line; and two barrel regions, one between the red lines and the other between the green lines.

not used in stage 1. When a track passes the stage-1 criteria, the road number and hits are sent to the SSB. The track is extrapolated into the 4 additional layers, nearby hits are found, and a full 12-layer fit is carried out. A minimum of 3 hits in the 4 layers is required. Duplicate track removal is again applied to the tracks that pass the  $\chi^2$  test, but now tracks in all roads are used in the comparison. SSB output tracks consisting of the hits on the track, the  $\chi^2$ , the helix parameters, and a track quality word that includes the layers with a hit are sent to the FTK-to-Level2 Interface Crate (FLIC). The FLIC organizes the tracks and sends them to the High Level Trigger ROSs using the standard ATLAS protocols, and carries out monitoring functions. The FLIC crate is organized so that in the future global event functions can be carried out in other cards in the crate.

## 2.2 Physical overview

### 2.2.1 Physical description of the system

**Crates and their contents** FTK as configured for a luminosity of  $3 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$  consists of 13 crates. The Data Formatter boards, which receive the data from the pixel and SCT RODs and transmit cluster centroids to the core crates, occupy 4 crates. They are ATCA shelves with full-mesh backplanes to facilitate the massive reorganization of the silicon hits from the readout optical fibers to the  $\eta - \phi$  tower organization of FTK. There are 8 core crates, each covering  $45^\circ$  in azimuth. They are VME crates because the Associative Memory boards (AMB) need a full 9U height to hold all of the AM chips and because the auxiliary card (AUX) behind each AMB is 280 mm deep in order to hold the Data Organizers and first-stage track fitters. A final crate contains the FLIC cards that send the FTK data to the High Level Trigger ROSs. The FLIC crate is also an ATCA shelf, but with a dual-star backplane. This will allow additional boards to be built in the future to carry out global event operations.

Figure 5 shows the organization of a core crate. There are 16 Processor Units (PU), 2 per FTK  $\eta - \phi$  tower, each consisting of an Associative Memory board with an AUX card behind it. Four PUs send their 8-layer tracks to a common Second Stage Board. Twelve-layer tracks from each pair of SSBs are sent to the FLIC crate.

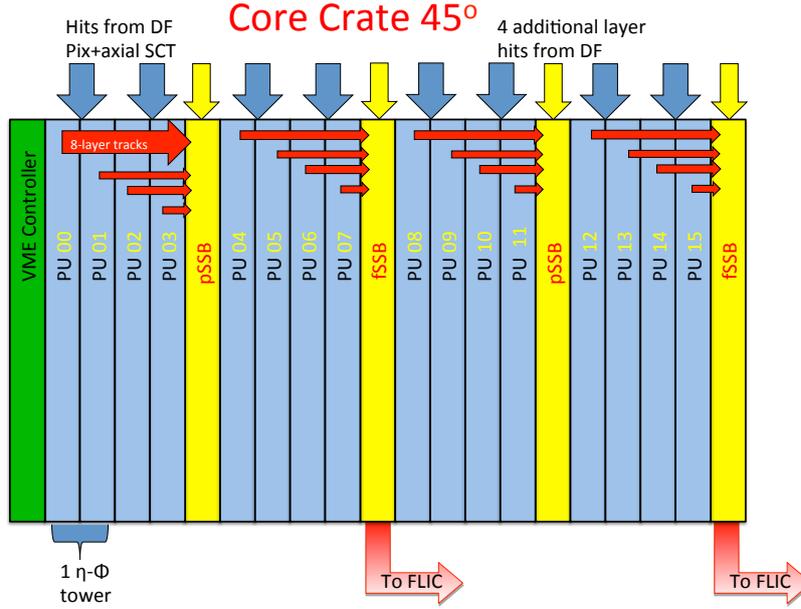


Figure 5: Layout of an FTK core crate and the interboard and intercrate data flow.

**Data into and out of FTK** The pixel and SCT data are sent to the Data Formatters from the RODs on standard ATLAS S-LINK optical fibers [16]. For the traditional ATLAS RODs, the HOLA mezzanine cards will be replaced by dual-output HOLAs which have two optical outputs that transmit identical data to the DAQ ROSs and to the FTK Data Formatters. This is accomplished by sending the output of the serializer to two drivers, each of which feeds an optical transmitter. The new IBL RODs [12] have the dual-output HOLA function built into the board.

The event output from FTK consists of tracks, each containing 5 helix parameters, the final fit  $\chi^2$ , the cluster coordinates, and a track quality word. The tracks will be organized in an  $\eta - \phi$  array to facilitate fetching tracks for a region of interest. The FLIC cards will send data to the ROSs on 8 optical fibers, one for each core crate (45° azimuthal slice).

**Racks** With 2 crates per rack, FTK will need 7 racks, 4 holding VME crates and 3 holding ATCA shelves. The 8 core crates however have substantial power and cooling requirements due to the power consumed by the large number of Associative Memory chips and the need for good air flow through the AUX card region of the crates. The power and cooling requirements for both the VME and ATCA crates are discussed in section 4.9.

### 3 Performance

This section describes the utility of the FTK as a toolbox for the HLT by detailing the performance on both single particles and multi-track objects. Section 3.1 presents the efficiencies and resolutions for single particles, as well as the efficiencies and fake rates in high pile-up samples. Multi-track objects are discussed in Section 3.2, beginning with muon isolation in 3.2.1, followed by  $b$ -jet performance in 3.2.2, then primary vertex finding performance in 3.2.3, and concluding with  $\tau$  leptons in 3.2.4. As an example of how FTK can contribute to the Phase I physics goals, the improvement in  $H \rightarrow \tau\tau$  trigger selections

is explored in 3.3. This section concludes with a discussion of possible uses for FTK in the HLT chains in 3.4.

For these studies, the pattern bank configuration of 16.8M patterns per tower<sup>4</sup> is used. Monte Carlo studies are performed using two different instantaneous luminosity conditions which are produced by superimposing additional interaction vertices on the hard scatter interaction. To approximate average instantaneous luminosities of  $1.7 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ , the number of additional interactions is taken from a Poisson distribution centered at 46. This sample is referred to as the 46 pile-up sample. Conditions of  $2.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  are approximated by adding a number of additional interactions drawn from a Poisson distribution centered at 69, and the sample is referred to as the 69 pile-up sample.

We perform our studies on the following datasets:

- inclusive  $t\bar{t}$ , for general purpose studies.
- vector boson fusion produced  $H \rightarrow \tau^+\tau^-$  decays with  $m_H = 120 \text{ GeV}$ , for studies of  $\tau$  performance and Higgs sensitivity improvements.
- gluon fusion produced Higgs with  $H \rightarrow b\bar{b}$  decays with  $m_H = 120 \text{ GeV}$ , for studies of moderate  $p_T$   $b$ -jets.
- gluon fusion produced Higgs with  $WH \rightarrow q\bar{q}$  decays with  $m_H = 120 \text{ GeV}$ , for backgrounds. This sample was chosen because it has a  $p_T$  spectrum similar to that of the  $H \rightarrow \tau^+\tau^-$  and  $H \rightarrow b\bar{b}$  events.

The ATLAS detector geometry used in these fully simulated events does not include the IBL, therefore an 11-layer FTK configuration is used. Comparisons are made to both truth and offline tracks. When comparing to offline, the recommended “high pile-up” cuts [10] are applied to the offline tracks. Tracks are required to have at least 9 silicon hits and not miss any hits (holes) in active pixel modules which they cross. If a track crosses an inactive module, the track is considered to have had a hit in that layer. These cuts control contributions from fake tracks. In some cases, additional requirements are applied to FTK tracks depending on the needs of the study; they are noted when appropriate. Due to the assignment of physical detector layers to logical layers used in FTK pattern recognition, we have designated  $\eta = 1.1$  as the transition point between the barrel and the end-cap.

### 3.1 Tracking Performance

In this section, the FTK performance is first studied in single particle MC samples with no pile-up. The resolution in several kinematic variables is presented and compared to the performance of the offline track reconstruction. The efficiencies for finding muons and pions are compared.

The effect of additional pile-up interactions is then studied in Higgs  $\rightarrow \tau^+\tau^-$  MC samples. Resolutions, efficiencies, and the rate of fake FTK tracks are studied in the presence of 46 and 69 additional pile-up interactions.

#### 3.1.1 FTK Performance in Single Particle Samples

Single muon and pion MC samples were simulated with a flat distribution in the particle helix parameters:  $\eta, \phi, |d_0| < 2.2 \text{ mm}, |z_0| < 120 \text{ mm}$  and curvature,  $1/(2p_T)$ . The resolution is compared to that of the offline track reconstruction. Although pile-up is absent, offline tracks are selected with the standard “high-pile-up” cuts:

---

<sup>4</sup>see barrel bank configuration option 2.2 and end-cap option 2.3 in Section 4.1.2

- Number of pixel plus SCT hits on the track plus the number of inactive modules crossed by the track must be at least nine.
- There must be no pixel holes on the track.

In addition to the hit requirements above, the following phase space cuts are applied to the muons to ensure that the tracks are within the FTK training phase space. These cuts are also applied to the offline tracks when taking the efficiency of the FTK tracks with respect to the offline tracks:

- The impact parameter,  $|d_0|$ , must be less than 2 mm.
- $|z_0|$  must be less than 110 mm.

Truth particles are matched to FTK tracks based on the fraction of hits on the track which were produced by the particle. Tracks with at least 50% of the hits coming from a single truth particle are considered matched.

Figure 6 shows the track-finding efficiency for FTK tracks with respect to truth particles for muons and pions versus various kinematic quantities. Muons, which do not undergo hadronic interactions in the detector and typically leave hits in all silicon layers, have the highest overall efficiency, at an average of 93%. The 7% inefficiency is primarily due to the geometric coverage of the pattern bank. There is a very small ( $< 1\%$ ) contribution from the fit quality cuts. The pions have 5% lower absolute efficiency in the barrel and up to 20% lower efficiency in the end-caps. This difference is due to the significantly higher interaction cross-section of pions, and it follows the overall material distribution in the detector.

The most relevant efficiency measure for FTK is the relative efficiency with respect to tracks found by the offline tracking algorithm. This is shown in Figure 7 for muons and pions for various kinematic quantities. The efficiency is defined by

$$\frac{N_{\text{Off}}(dR_{\text{Off-FTK}} < 0.05)}{N_{\text{Off}}}$$

where  $N_{\text{Off}}$  is the number of offline tracks and  $N_{\text{Off}}(dR_{\text{Off-FTK}} < 0.05)$  is the subset with an FTK track within a cone of  $dR = \sqrt{(\eta_{\text{FTK}} - \eta_{\text{Off}})^2 + (\phi_{\text{FTK}} - \phi_{\text{Off}})^2} < 0.05$ . The offline tracks are required to satisfy the cuts listed in the previous section. The FTK tracks, which already require at least nine hits for most tracks, have no additional cuts.

Here the relative inefficiency between muons and pions is reduced to less than 1% because the nuclear interaction probability affects offline and FTK tracking similarly. The inefficiency with respect to offline which remains is due to the geometric coverage of the banks.

Resolutions with respect to truth muon parameters were also studied as a function of the curvature of the muon in Figure 8. The resolution is defined as the width of a gaussian fit in a range of 3 times the RMS of the distribution. The same definition is used for the offline tracks to facilitate an accurate comparison. FTK track errors are parameterized in terms of these quantities for applications such as  $b$ -tagging and primary vertex finding. At large absolute values of curvature, where the resolution on the transverse quantities  $d_0$ ,  $p_T$  and  $\phi$  is dominated by multiple scattering, the FTK and offline have similar resolutions. At small values of curvature the FTK hit resolution and fit precision contribute to a decreased relative resolution. The longitudinal quantities such as  $\eta$  and  $z_0$  show a worse resolution with respect to offline even at low curvature. The FTK  $z_0$  resolution is three times that of the offline tracks at small curvature.

Figure 9 shows the resolution of  $d_0$  and curvature as a function of  $\eta$  for low and moderate  $p_T$  tracks separately. For the low  $p_T$  tracks the FTK and offline have very similar performance. However, for tracks with  $p_T > 5$  GeV the performance significantly degrades with respect to offline in the end caps. The resolution is driven by our fitting constants, which are derived from low momentum muons as described

in Section 5.2.2. When we use fitting constants derived from muons above 3 GeV, as shown in blue in the figure, we recover a significant portion of the resolution increase. We are modifying our training so that we no longer predominately use low momentum tracks whose resolution is dominated by multiple scattering, especially in the forward direction.

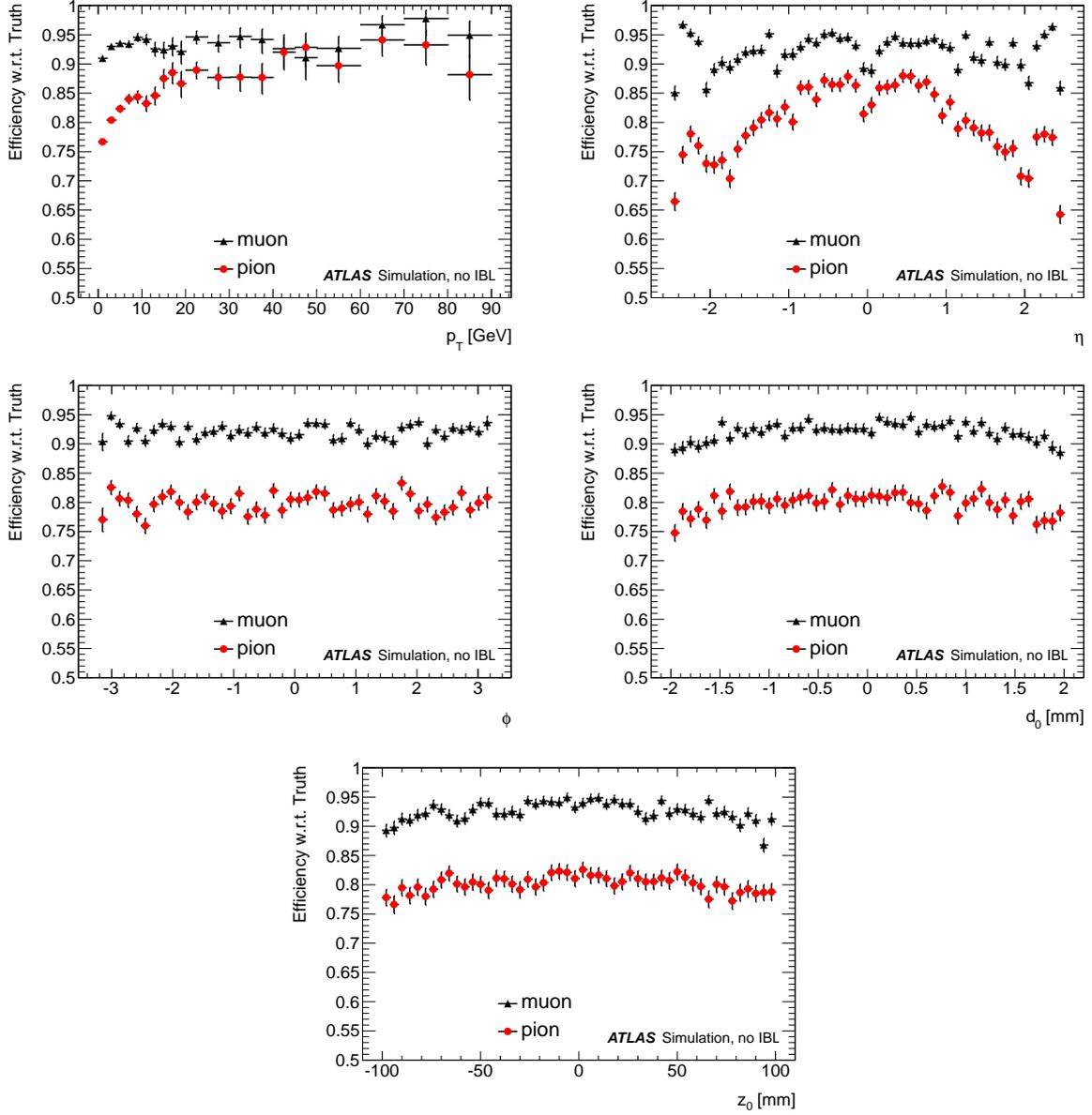


Figure 6: Absolute efficiency with respect to truth particles in muon and pion samples versus  $p_T$ ,  $\eta$ ,  $\phi$ ,  $d_0$ , and  $z_0$ . The truth particles are required to have a  $p_T > 1$  GeV.

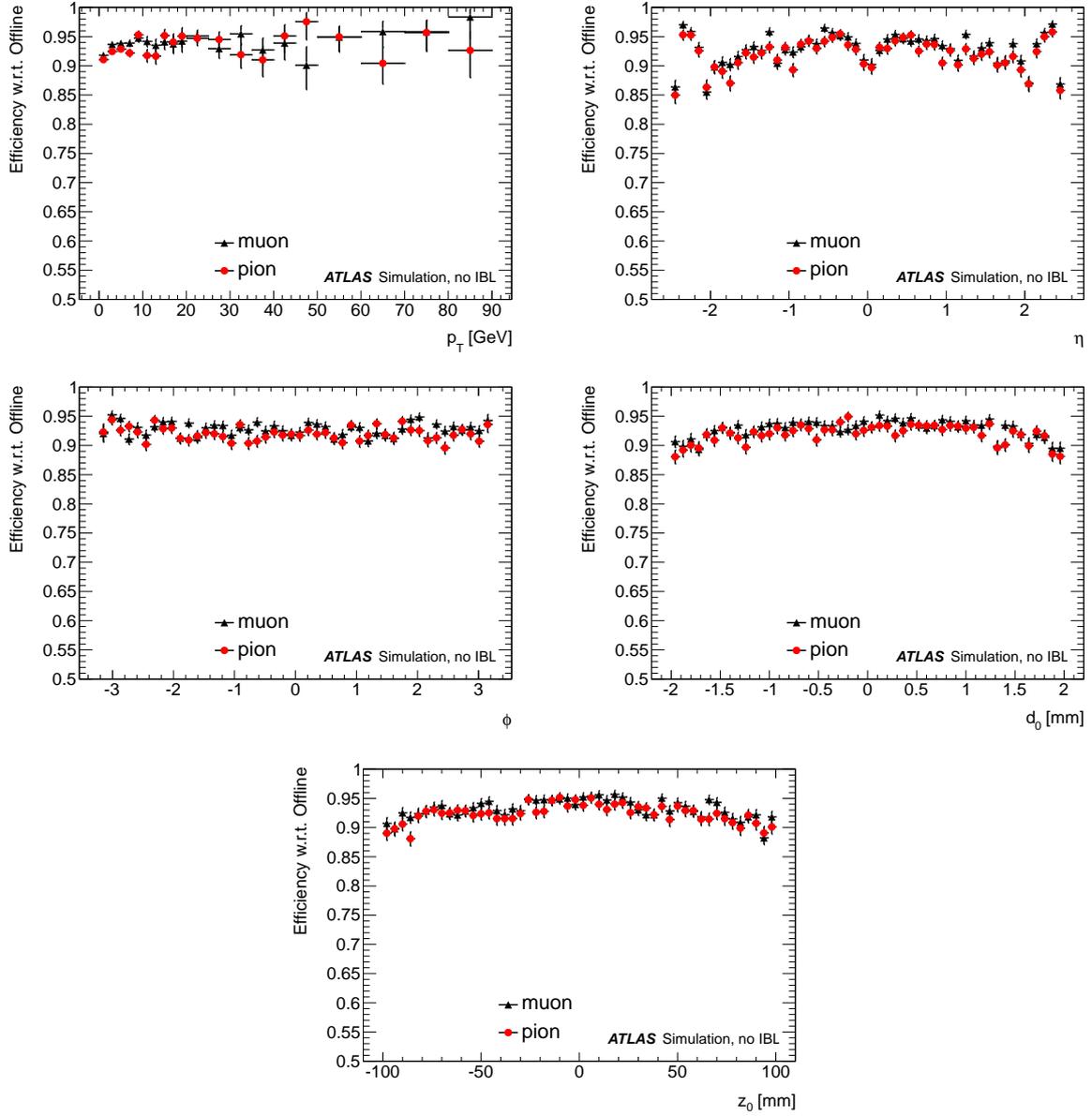


Figure 7: FTK efficiency with respect to offline in muon and pion samples versus  $p_T$ ,  $\eta$ ,  $\phi$ ,  $d_0$ , and  $z_0$ .

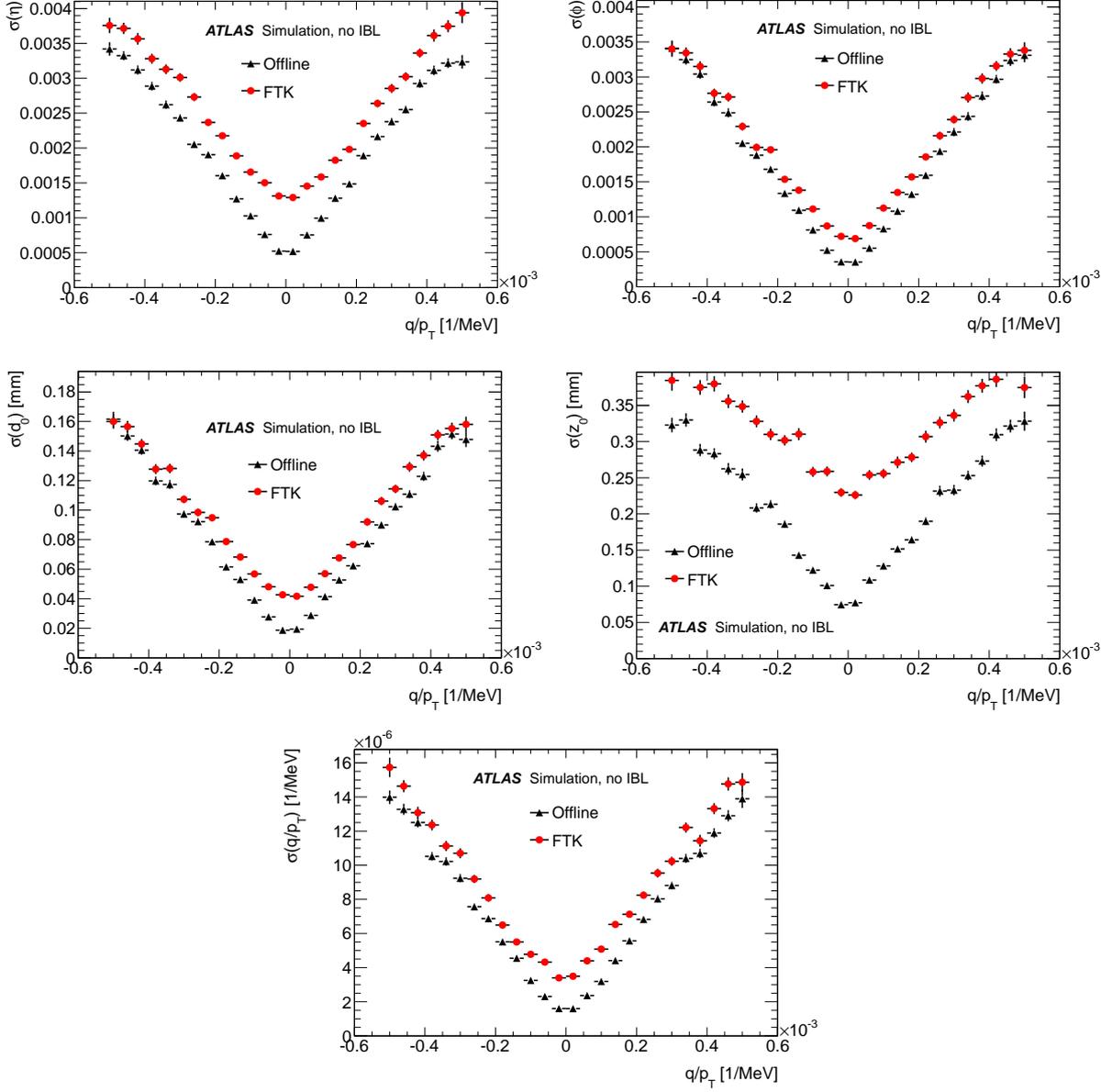


Figure 8: Muon resolution in  $\eta$ ,  $\phi$ ,  $d_0$ ,  $z_0$ , and  $1/p_T$  as a function of curvature, for FTK (red) and offline (black) tracks. See the text concerning the  $z_0$  resolution.

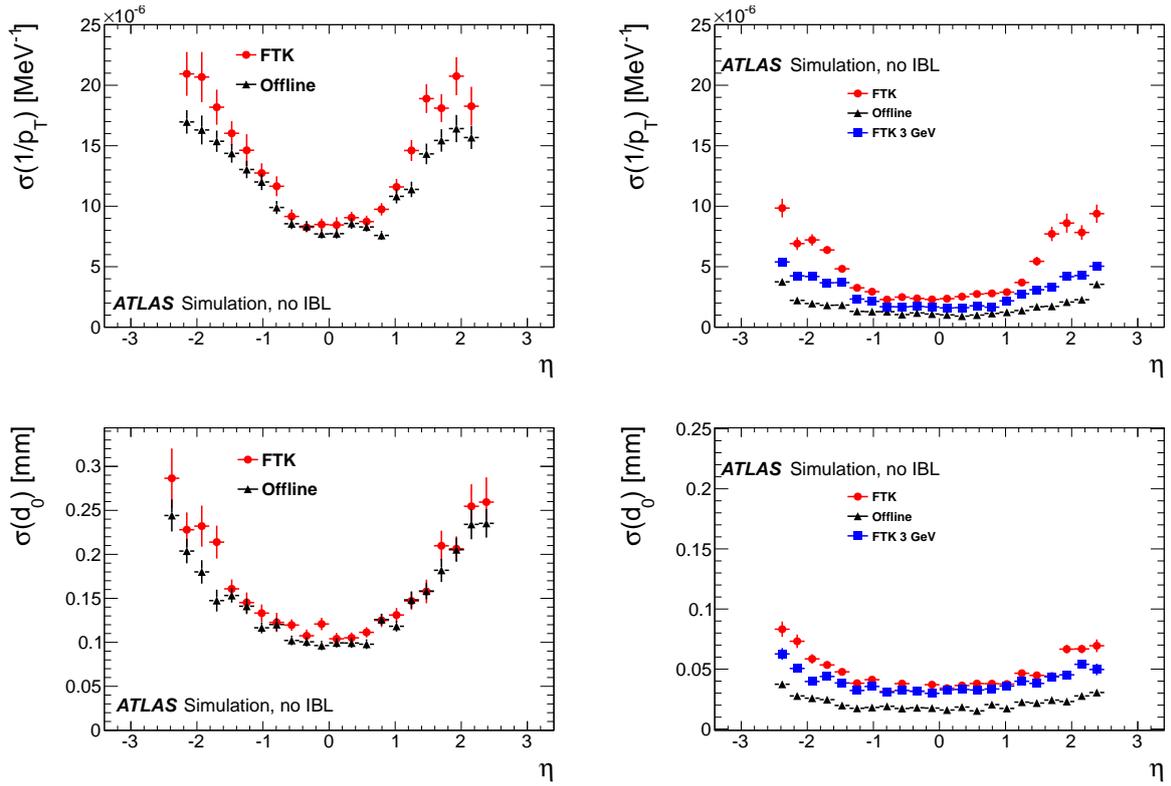


Figure 9: Muon resolution in curvature and  $d_0$  as a function of  $\eta$ , for FTK (red) and offline (black) tracks. The left figures are for muons with  $1.0 < p_T < 1.25$  GeV, the right are for  $p_T > 5.0$  GeV. The blue points are generated with a special set of FTK fitting constants which are tuned with muons with  $p_T > 3$  GeV.

### 3.1.2 FTK Performance in High-Luminosity Samples

In this section the FTK performance is evaluated in 46 and 69 pile-up samples. First we compare the FTK efficiency with respect to offline tracks in a  $t\bar{t}$  sample. The offline tracks are required to pass the high pile-up cuts specified at the beginning of this section. The FTK tracks are matched to offline tracks using a cone of  $dR = \sqrt{(\eta_{\text{Off}} - \eta_{\text{FTK}})^2 + (\phi_{\text{Off}} - \phi_{\text{FTK}})^2} = 0.05$ .

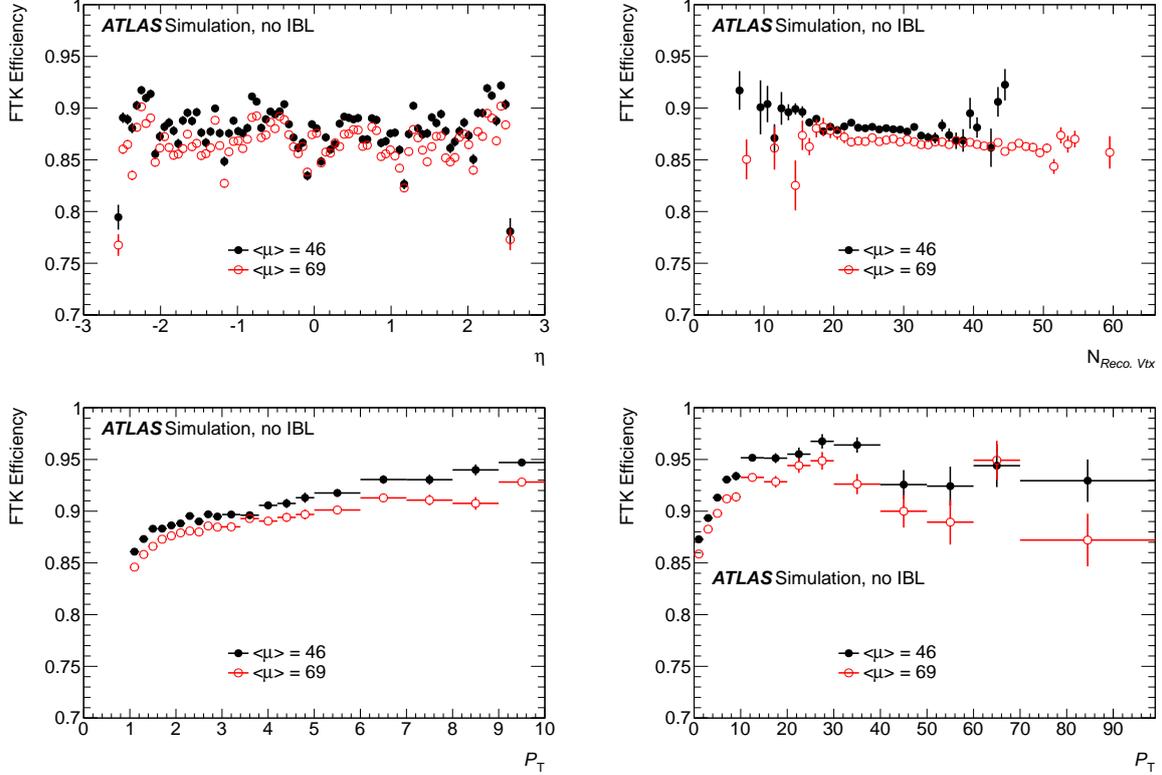


Figure 10: Efficiency of FTK tracks with respect to offline tracks as a function of  $p_T$ ,  $\eta$ , the number of offline reconstructed vertices  $N_{\text{RecoVtx}}$ , and  $p_T$  in a sample with 46 (black) and 69 (red open circles) pile-up events.  $p_T$  is shown for  $p_T < 10$  GeV in the bottom left and the full range on the bottom right.

Figure 10 shows the comparison of the performance in two pile-up environments for several variables. The efficiency is generally flat in  $\eta$  with bin-by-bin fluctuations dominated by the number of detector modules crossed by a track which varies slightly as a function of  $\eta$ . The efficiency shows a slight decrease of a few percent going from 10 to 60 reconstructed primary vertices per event. The two samples have different efficiencies for the same average number of reconstructed vertices because the efficiency is plotted vs. the number of reconstructed vertices and the 69 pile-up sample has a lower average vertex reconstruction efficiency. The efficiency is 83% for tracks with  $p_T$  of 1 GeV and rises to above 90% for  $p_T > 10$  GeV. There is a slight loss in efficiency for tracks with  $p_T > 40$  GeV. The 69 pile-up samples is slightly less efficient than the 46 pile-up sample, illustrating the same effect as seen in the slope in the upper right plot of Figure 10.

In order to obtain an accurate estimate of the fake rate with FTK tracks, a special sample of  $H \rightarrow \tau^+\tau^-$  MC were generated with the full truth particle record for all interactions in the event. Figure 11 shows a comparison of the fraction FTK tracks with less than 50% of their hits coming from a single truth particle in 46 and 69 pile-up samples as a function of  $\eta$  and  $p_T$ . The fake rate versus  $\eta$  is 3% in the barrel region

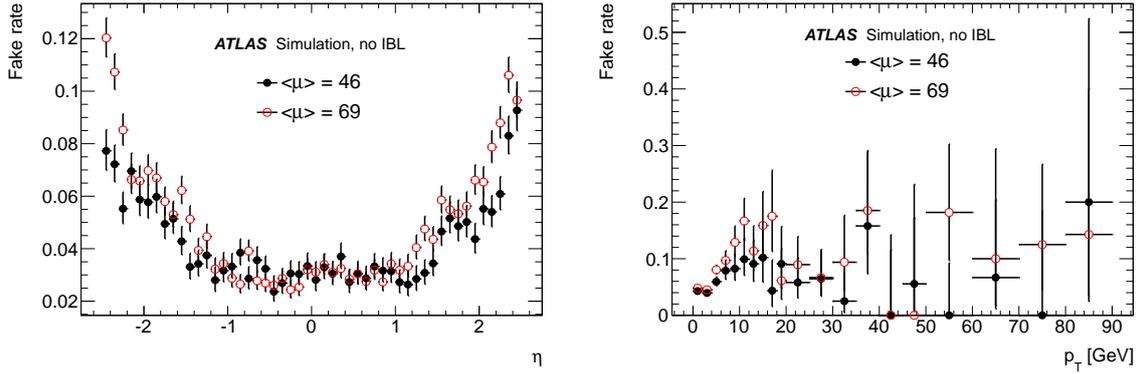


Figure 11: Fraction of FTK tracks not matched to truth as a function of  $\eta$  (left) and  $p_T$  (right). Both the 46 (black) and 69 (red) simultaneous interactions samples are shown.

and rises to 8% in the end-caps for the 46 pile-up sample and 10% for the 69 PU sample. The fake rate is 5% for  $p_T < 4$  GeV and rises for higher momenta. The fake rate in the end-caps could be improved by using a stricter  $\chi^2$ -cut, at the cost of lower efficiency for true tracks.

## 3.2 Event Object Performance

### 3.2.1 Lepton Isolation

High  $p_T$  lepton signatures will remain a cornerstone of the physics program in Run 2 and beyond. Here we consider the case of muons from W decays. In order to separate muons from electroweak boson decays and muons from heavy flavor decays or meson decay-in-flight, an isolation requirement is typically imposed. Track based isolation has the advantage that tracks coming from the primary interaction vertex can be selected, minimizing the effects of pile-up. Calorimeter isolation is in principle more powerful because it is not sensitive to fluctuations of the charge fraction of jets, however it has a strong dependence on pile-up because it integrates all of the energy in a cone around the muon, regardless of which interaction produced it. As a result, only track based isolation is currently used in the HLT muon triggers [17]. In this section we study the use of FTK tracks in track based isolation. Studies of corrections to calorimeter isolation that could lessen its dependence on pile-up will be studied for the forthcoming ATLAS TDAQ Phase I upgrade TDR.

To study track isolation we pick a signal sample of muons from W decays, and a background sample of muons from heavy flavor decays. Our signal muons come from a MC sample of  $WH \rightarrow \mu\nu q\bar{q}$  and our background muons are filtered from semi-leptonic  $B$ -hadron decays in  $WH \rightarrow \mu\nu b\bar{b}$  and  $t\bar{t}$ . In the background sample, truth matching is employed to make sure the muon came from the  $B$ -hadron decay and not the  $W$ .

Events are selected from MC requiring that the lowest unpre-scaled Level-2 single muon trigger, L2\_mu20, has fired in the event. A muon candidate is then matched to the region of interest in the detector where the trigger has fired. To avoid trigger turn-on effects, we require that the muon candidate have a transverse momentum larger than 20 GeV.

Muons are classified at truth level as coming from a  $B$  or a light hadron, or a  $W$  boson decay using a  $dR = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$  matching between the truth and reconstructed muons track parameters.

Tracks in the isolation cone are required to satisfy the high pile-up cuts described at the beginning of this section. To identify the offline muon, an offline track must additionally pass the following muon identification criteria [18]:

- The impact parameter must be less than 0.2 mm
- The number of B-Layer hits must be greater than 0
- There must be a successful extension to the TRT when the track is inside the TRT fiducial volume ( $0.1 < |\eta| < 1.9$ )

The FTK and offline track isolation are computed as follows. Tracks are selected which have a  $|z_0 \times \sin(\theta)|$  separation less than 1.5 mm from the highest  $p_T$  track matched to the muon. The  $p_T$  is summed in a cone of size  $R_{\text{cut}}$ , the quantity denoted  $\sum_{dR < R_{\text{cut}}} p_T$ , and the highest  $p_T$  track is removed from the sum. The relative track isolation is computed as

$$(\sum_{dR < R_{\text{cut}}} p_T - p_T^{\text{lead}}) / p_T^{\text{lead}} \quad (1)$$

where  $p_T^{\text{lead}}$  is the leading track  $p_T$ . If the muon is well isolated as in the decay of a  $W$  boson the relative isolation should be small. If the muon arises from a  $B$  hadron decay it is likely to have poorer track isolation because it is embedded in a jet.

The relative isolation for two different cone sizes (0.2 and 0.3) when the muon is truth matched to a  $W$  boson or  $B$  hadron in central ( $|\eta| < 1.1$ ) or forward regions ( $|\eta| > 1.1$ ) of the tracking system are shown in Figures 12 and 13. Good agreement is seen between FTK and offline in both the central and forward regions.

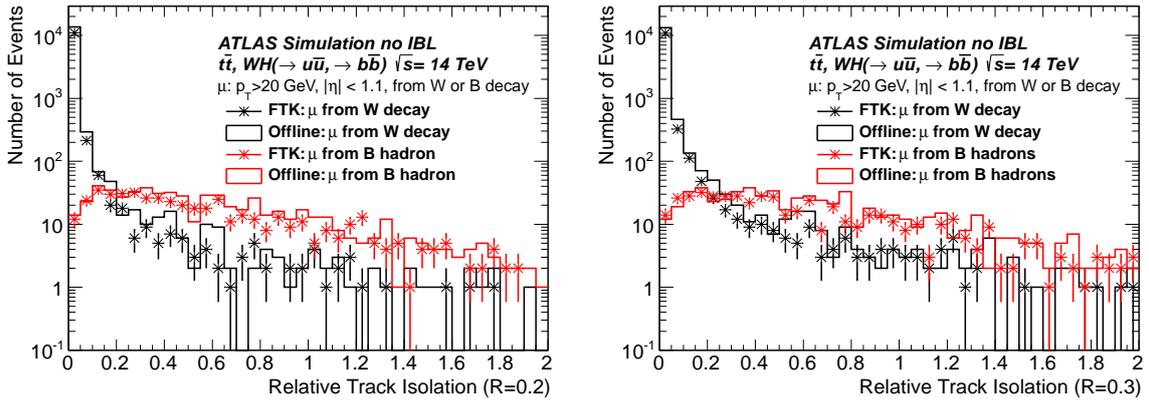


Figure 12: Relative track isolation plotted for muons from  $W$  decays (Black), from  $B$ -hadron decays (Red). Plots on the left(right) are computed using a cone of size  $dR < 0.2(0.3)$ . In all plots muons are required to be in the central region ( $|\eta| < 1.1$ ).

The efficiency for muons from  $W$ s versus the rejection of muons from  $B$ -decays is plotted in Figure 14 for both the  $R = 0.2$  and  $R = 0.3$  cones. In the barrel region FTK and offline have similar performance for efficiencies of greater than 97%. In the forward region FTK has less rejection than offline for fixed efficiency for all points except at very high efficiency and low rejection. For both the FTK and Offline the cone of  $R = 0.2$  performs better than  $R = 0.3$ . Figure 15 shows the stability of the efficiency of the isolation cut as a function of the number of primary vertices in the event.

### 3.2.2 b-jet tagging

New physics that couples to heavy fermions may be rich in final-state  $b$  quarks, but not necessarily other easily identifiable objects which can be used in the trigger. Given the large QCD production of light

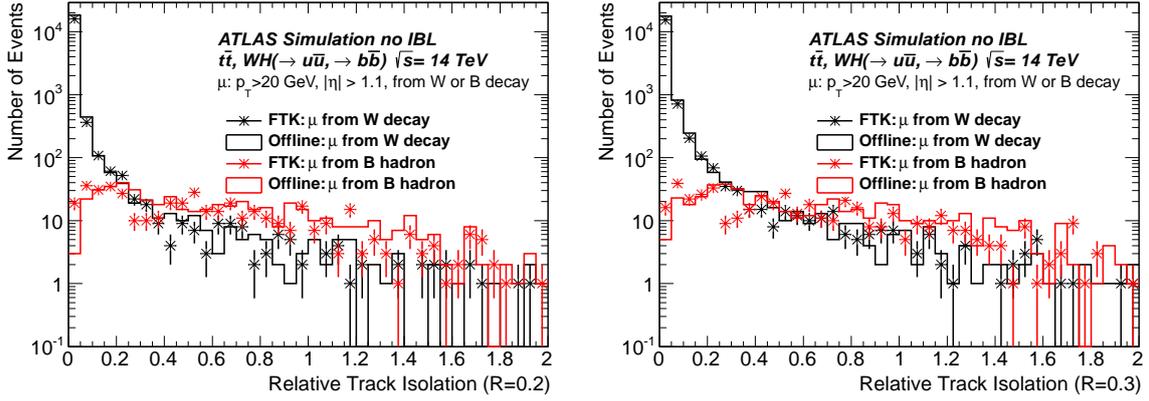


Figure 13: Relative track isolation plotted for muons from  $W$  decays (Black), from  $B$ -hadron decays (Red). Plots on the left(right) are computed using a cone of size  $dR < 0.2(0.3)$ . In all plots muons are required to be in the forward region ( $|\eta| > 1.1$ ).

quarks and gluons, it is important that the ATLAS trigger efficiently select jets from  $b$ -quarks while providing a large rejection factor against other jets. This is a challenging task for the current Level-2 trigger because purely jet-based triggers either have very high thresholds for single jets, or have many jets in multi-jet triggers. The single jet thresholds are typically too high for  $b$ -tagging to be useful, and the multi-jet triggers produce too many ROIs for the time and CPU constraints at Level-2. The availability of FTK tracks following a Level-1 trigger removes the constraints on the number ROIs to be considered for  $b$ -tagging and allows the use of sophisticated algorithms to select  $b$ -quark jets at Level-2. This opens up the possibility of doing  $b$ -tagging on multi-jet triggers at a higher rate than currently possible and could potentially give the ATLAS trigger sensitivity to models with many moderate momentum  $b$ -quarks which were previously inaccessible, e.g., see models discussed in [19].

In the remainder of this section, the basic  $b$ -tagging performance with FTK is presented. The quantitative enhancements to specific physics analyses will be included in the upcoming TDAQ technical design report.

Throughout this section the performance of FTK tracking and  $b$ -tagging is shown using the fully simulated samples of  $t\bar{t}$  events described at the beginning of this section. These events provide realistic samples of both signal  $b$ -jets, from the decaying top quark, and background jets, from hadronically decaying  $W$ -bosons, initial and final-state radiation, and from additional pile-up interactions.

To suppress the rate of fake tracks the cuts described in Section 3 are applied. In addition, the following requirements are imposed:

- $|d_0| < 2$  mm
- $|z_0 \sin(\theta)| < 1.5$  mm
- $B$ -layer hit

The FTK track finding efficiency for this selection prior to the  $B$ -layer requirement is presented in section 3.1 for single particles and for inclusive physics events. Note that these cuts necessitate the knowledge of the primary vertex location which is not possible in normal trigger processing without FTK. For these studies the offline primary vertex is used for both FTK and offline tracks.

For accurate  $b$ -tagging, good performance in dense jet environments is necessary. Figure 16 shows the  $\Delta R$  distribution between the tracks in a jet and the jet axis for both offline and FTK tracks. The plots

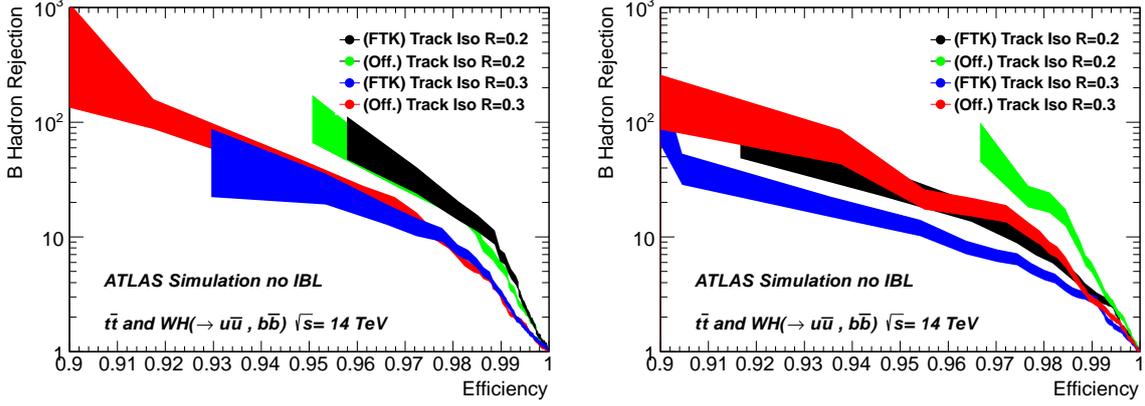


Figure 14: Efficiency versus rejection curves found in R=0.2 and 0.3 cones around a (a) central or (b) forward muon candidate that has originated from a  $W$  boson or  $B$  hadron. The central region's FTK efficiency versus rejection curves are in agreement with those observed in offline. The forward region FTK efficiency versus rejection is less efficient for similar  $B$  hadron rejection than observed in offline. The band represents the statistical uncertainty.

are normalized to unity to remove any global tracking efficiency differences. Good agreement is seen between the two distributions in both the forward and barrel regions showing that the performance of the two track finding methods is similar in the jet environments. The relative FTK track finding efficiency and fake rate with respect to offline is shown in Figures 17 and 18, both for pile-up of 46 and 69. The figures show that the FTK tracking performance does not degrade with respect to the offline performance in the denser jet environments. The efficiency with respect to the  $\Delta R$  between the track and the jet axis rises at small  $\Delta R$  because the efficiency is determined with  $dR < 0.05$  matching between FTK and offline tracks and the core of the jets has a higher track density. The fake rates are somewhat lower than shown in Section 3.1 due to the fact that the fake rate here is determined from FTK tracks which do not have an offline track in a cone of  $dR < 0.05$ , whereas the previous plots use the full truth information.

The transverse impact parameter distribution is the basic input to  $b$ -tagging. This distribution is sensitive to the long lifetime of particles within the  $b$ -jet and provides a means of separating  $b$ -jets from light-flavor jets. The transverse impact parameter of tracks associated to light-flavor and heavy-flavor jets is shown in Figure 19 for both the FTK and the offline tracks. The impact parameter distribution is signed such that track displacements in the direction of the jet are given positive values, while tracks with displacements opposite of the jet direction are given negative values. A longer positive tail is present for the tracks associated to heavy-flavor jets for both the offline and FTK tracks, as is expected for particles from  $B$ -hadron decays. The light flavor jets have a symmetric distribution with much smaller tails, allowing for good separation between the two classes of jets. This separation is seen in both the  $\mu = 46$  and the  $\mu = 69$  samples and the relative performance does not degrade in the higher pile-up sample. Because it has a larger impact parameter resolution, the FTK tracks have a broader distribution.

The transverse impact parameter significance, defined as the transverse impact parameter divided by its associated uncertainty, is shown in Figure 20. The FTK track  $d_0$  uncertainty described in Section 3.1.1 is parameterized in  $\eta$  and  $p_T$ . The offline tracks use the uncertainties from the covariance matrix of the track fit. Again, separation is seen for the tracks associated to light-flavor and heavy-flavor jets. The differences between FTK and offline seen in Figure 19 have been absorbed by their respective resolutions and good agreement is seen for the  $b$ -quark jets. The light-flavor jet distribution has more events in the high significance tail, however good separation between light and heavy flavor jets is still seen.

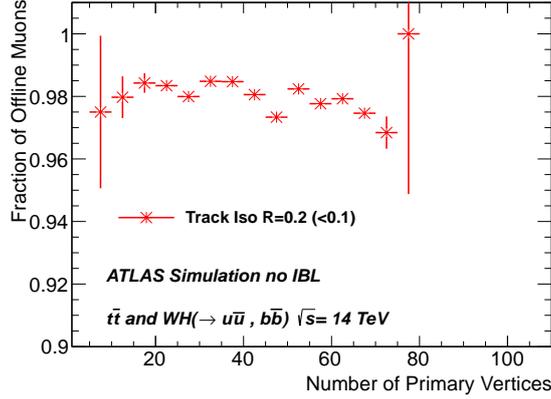


Figure 15: Efficiency of a relative isolation cut of 0.1 for tracking isolation as a function of the number of primary vertices in the event.

The number of tracks associated to a jet in the tail of the transverse impact parameter significance distribution is shown in Figure 21. Here, the tail is defined as tracks with transverse impact parameter significance greater than four. For both the offline and FTK tracking, the number of tracks with large impact parameter significance is significantly larger for  $b$ -quark jets than for light-flavor jets. The good resolution parametrization for FTK tracks results in a good agreement for the number of tracks with large impact parameter significance in  $b$ -jets, both for the barrel and the end-caps. For light jets FTK shows nearly the same performance as the offline in the barrel, while more tracks with large impact parameter significance are observed in the end-caps.

As a baseline for studying  $b$ -tagging performance at Level-2 using FTK, a simple impact parameter based  $b$ -tagger has been developed. This  $b$ -tagger uses the transverse impact parameter significance of tracks associated to jets in a likelihood to distinguish  $b$ -jets from light-flavor jets. No attempt is made to use secondary vertex information. The input distributions to the likelihood are taken from a set of events independent of those used to test the performance. In the remainder of this section, the performance of the FTK  $b$ -tagging is shown using this  $b$ -tagging algorithm. It is planned to explore the secondary vertex information in a more refined  $b$ -tagging algorithm.

It is important that the events selected by an FTK-based trigger would be useable by the offline tagging algorithms, therefore we evaluate the performance of the offline algorithms assuming several FTK  $b$ -tagger working points. Figures 22 and 23 show the efficiency versus light-jet rejection of offline  $b$ -tagging algorithms with respect to the FTK selection. Two offline  $b$ -tagging algorithms are evaluated at three different FTK working points at 46 and 69 pile-up interactions. The offline algorithms considered are the IP2D tagging algorithm [20], shown in Figure 22, and the more sophisticated JetFitterCombNN  $b$ -tagging algorithm [21], shown in Figure 23. IP2D is a simple likelihood-based tagger utilizing the transverse impact parameter significance and is robust to pile-up effects. JetFitterCombNN exploits the topology of  $b$ -hadron decays by finding the  $b$ - and  $c$ -vertices in the jet and using them to calculate the flight path of the  $b$ -hadron. As can be seen in the figures, with offline operating points slightly below the online FTK operating point, good online rejection can be achieved with little loss in performance. For the offline IP2D tagger offline working points of 70% can be used with little loss in efficiency in the barrel. For JetFitterCombNN, using an offline working point of 50% efficiency, a factor of 18 in light-flavor rejection at Level-2 can be achieved with little loss in offline  $b$ -tagging performance. At a typical working point of 60% some offline rejection is sacrificed for a factor 18 online rejection.

We also examine the performance of the FTK  $b$ -tagging algorithm in a sample of jets selected by

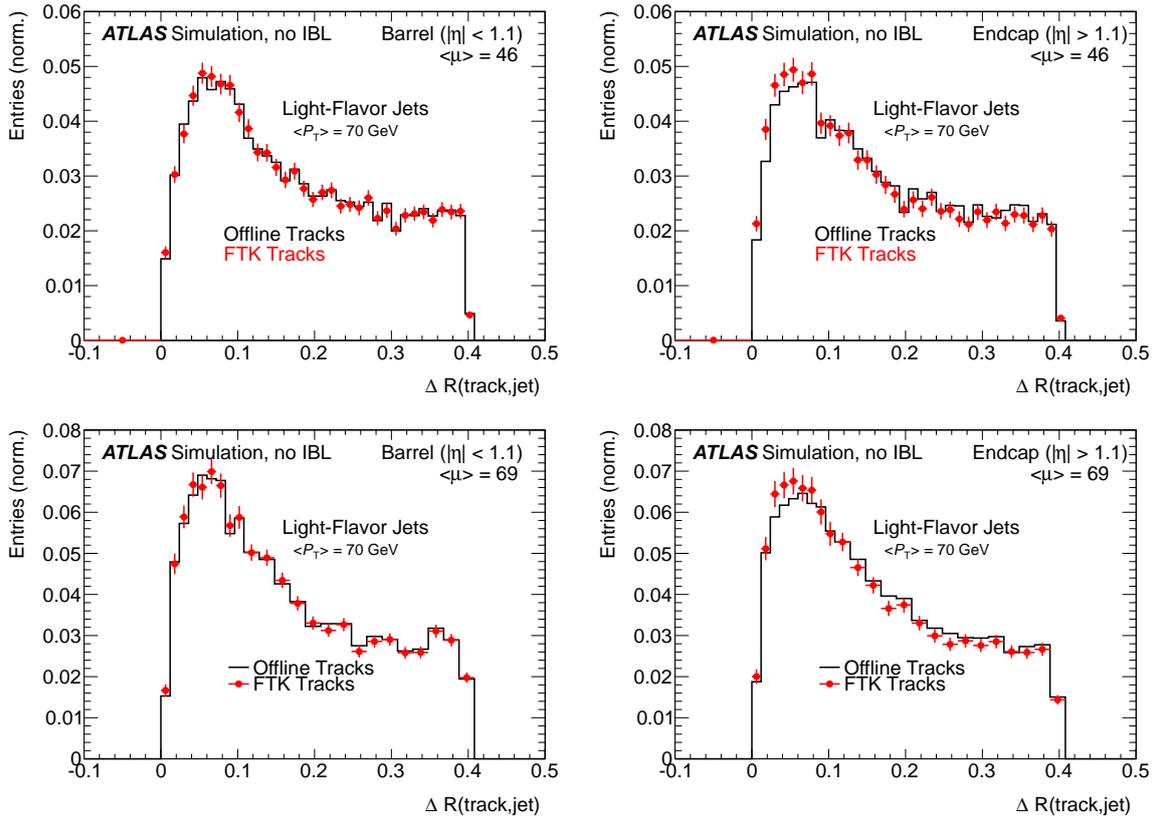


Figure 16:  $\Delta R$  between tracks and the axis of light-flavor jets. The solid lines show the distribution for the offline tracks, whereas the points show the distribution for the FTK tracks. The left-hand plots show the distribution in the barrel and the right-hand plots show the distribution in the end-caps. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

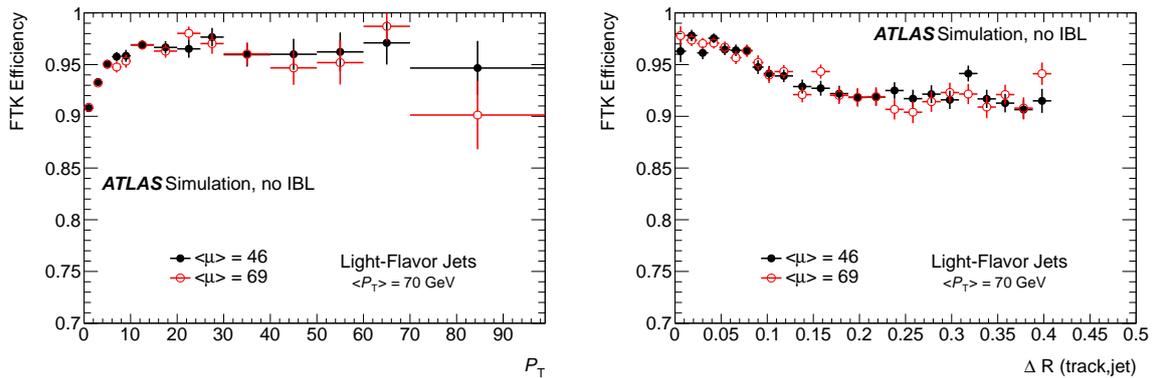


Figure 17: Efficiency of FTK tracks with respect to offline tracks in light-flavor jets as a function of  $p_T$  and distance from the jet axis in a sample with 46 (black) and 69 (red) pile-up events.

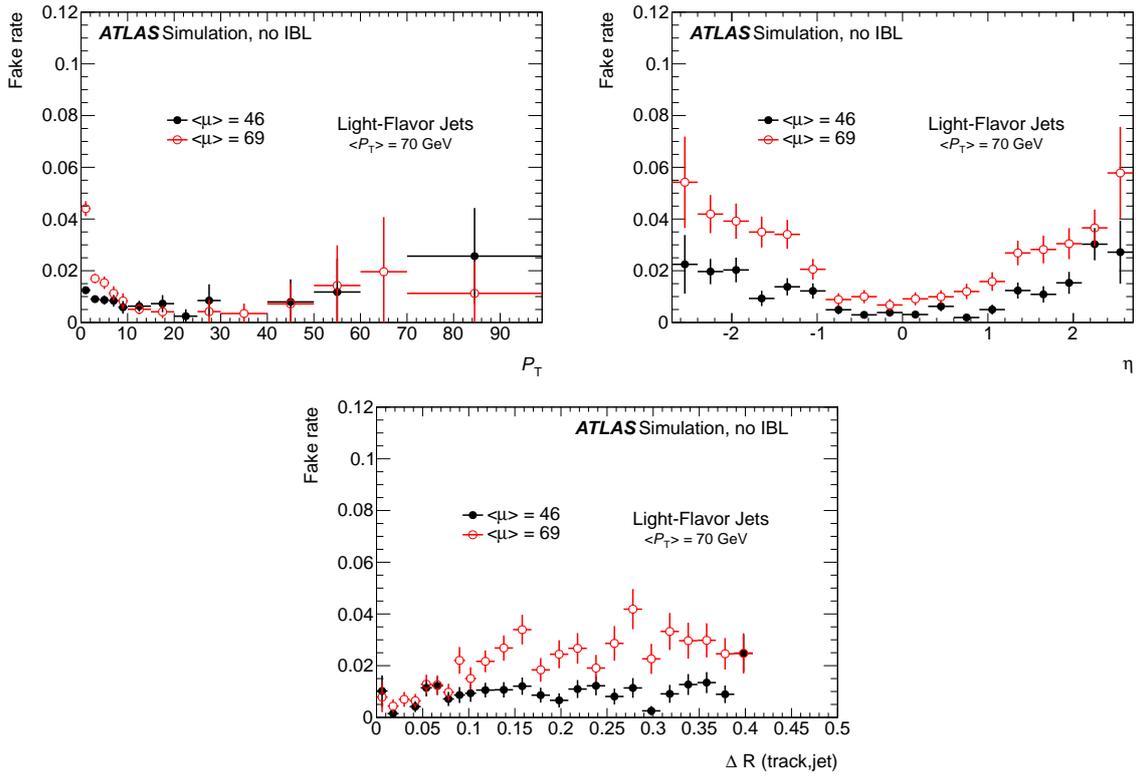


Figure 18: Fake rate of FTK tracks with respect to offline tracks in light-flavor jets as a function of  $p_T$ ,  $\eta$ , and distance from the jet axis in a sample with 46 (black) and 69 (red) pile-up events.

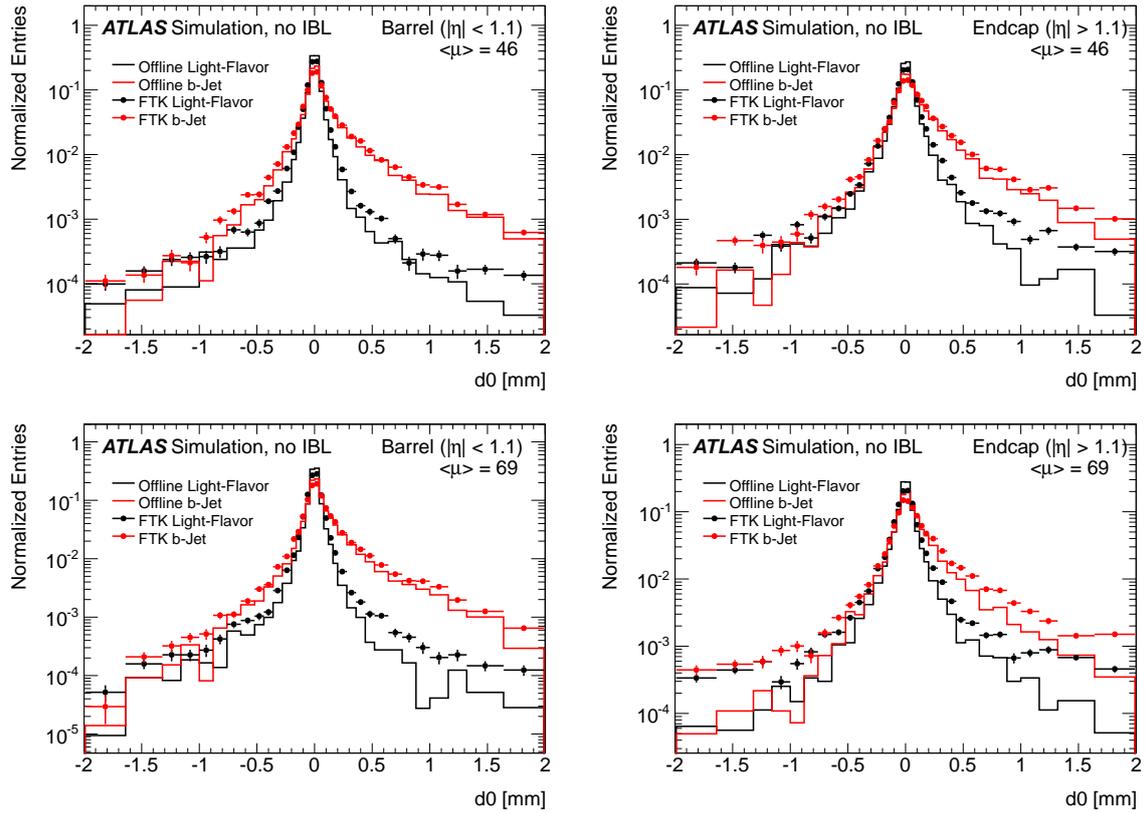


Figure 19: Transverse impact parameter of tracks associated to light-flavor (black) and heavy-flavor (red) jets. The solid lines show the distribution for the offline tracks, whereas the points show the FTK tracks. The left-hand plots show the distribution in the barrel and the right-hand plots show the distribution in the end-caps. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

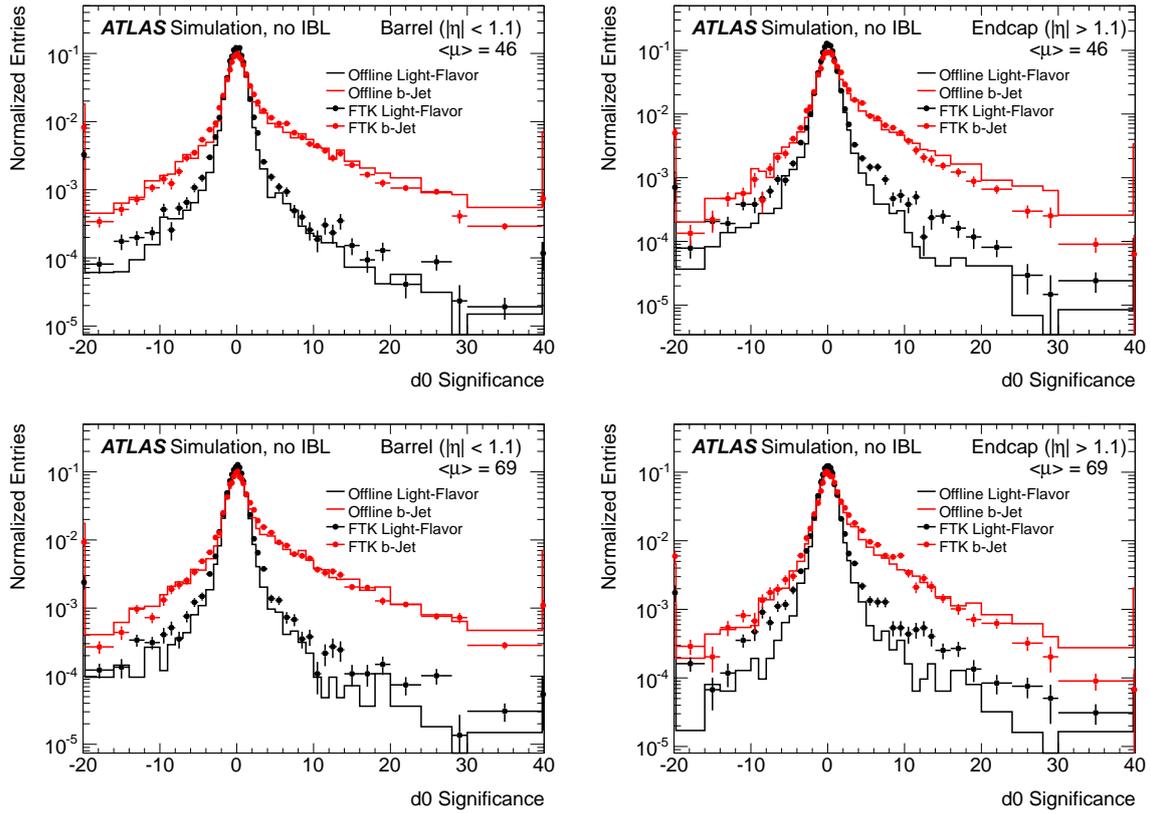


Figure 20: Transverse impact parameter significance of tracks associated to light-flavor (black) and heavy-flavor (red) jets. The solid lines show the distribution for the offline tracks, whereas the points show the FTK tracks. The left-hand plots show the distribution in the barrel and the right-hand plots show the distribution in the end-caps. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

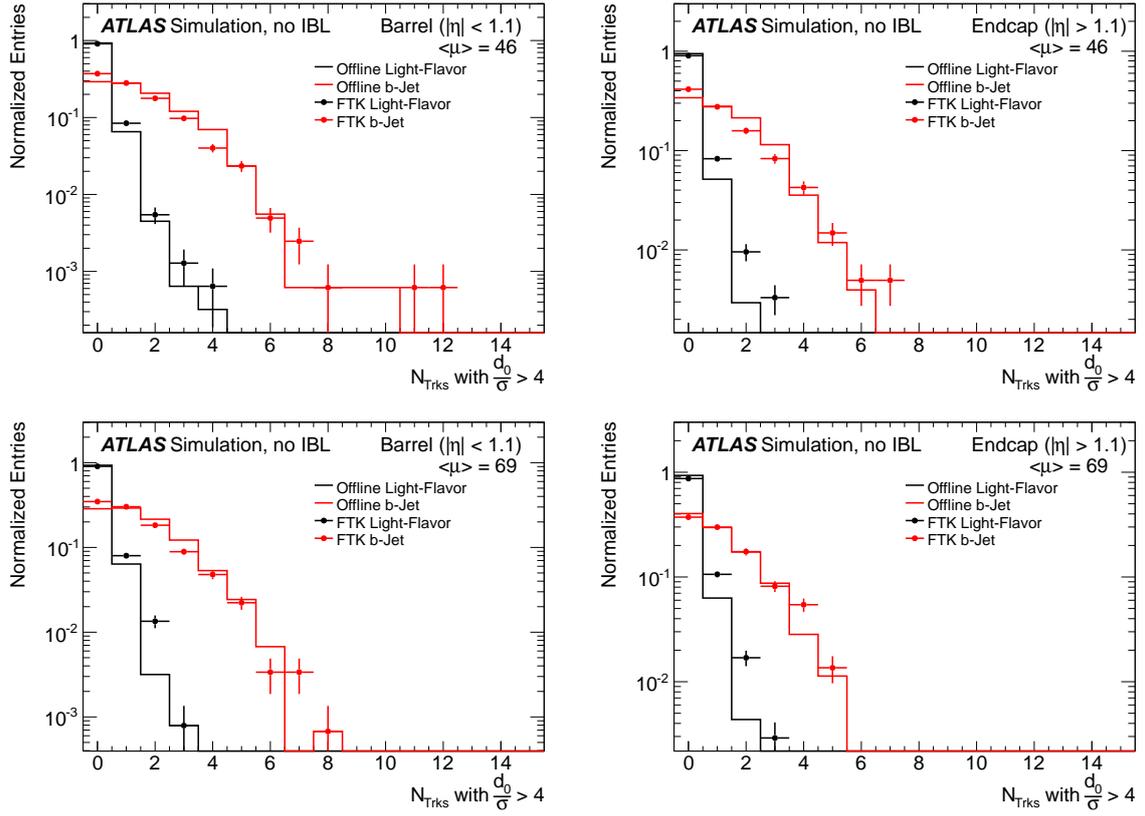


Figure 21: Number of tracks with transverse impact parameter significance larger than 4 in light-flavor (black) and heavy-flavor (red) jets. The solid lines show the distribution for the offline tracks, whereas the points show the FTK tracks. The left-hand plots show the distribution in the barrel and the right-hand plots show the distribution in the end-caps. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

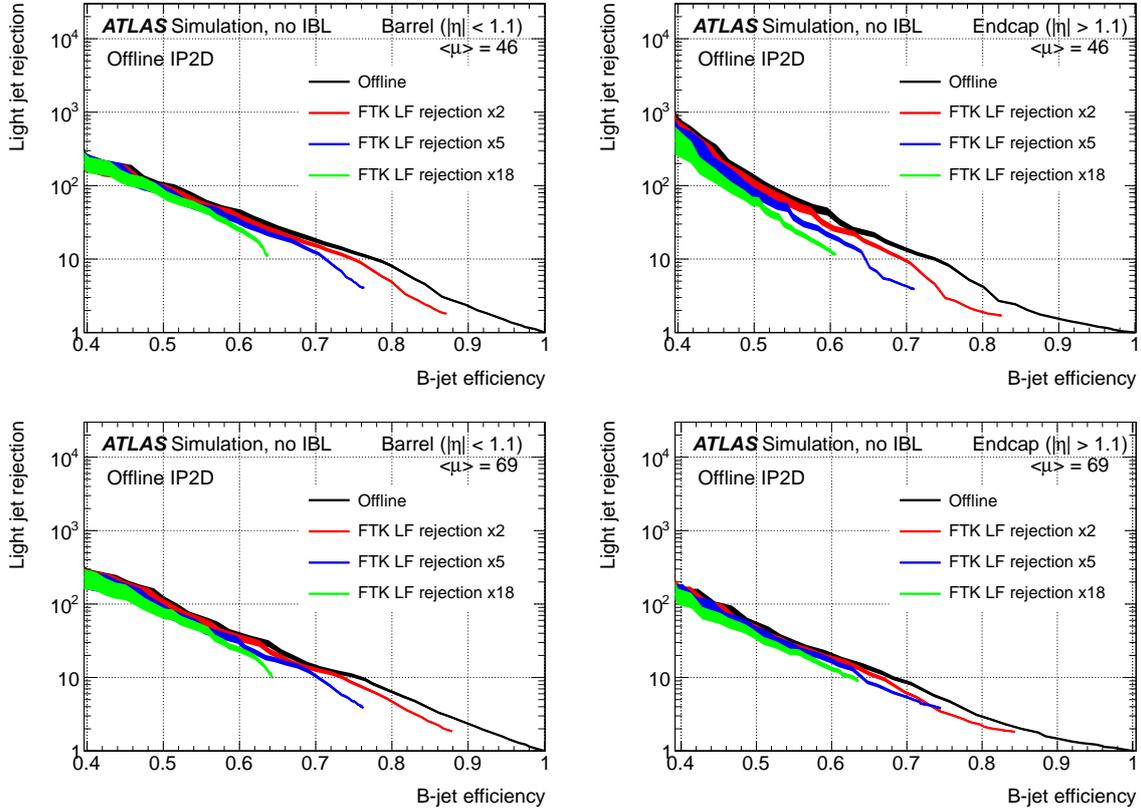


Figure 22: Performance of the offline IP2D  $b$ -tagging algorithm on samples selected using FTK. The curves show the total light-flavor rejection as a function of the total  $b$ -jet efficiency. The black curve gives the offline performance if no FTK selection is applied, the red curve shows the combined offline and FTK performance where an FTK selection giving a factor of 2 light-flavor rejection has been applied, the blue curve gives the performance where an FTK selection gives a factor of 5 light-flavor rejection has been applied, and the green curve gives the performance when an FTK selection with a light-flavor rejection of a factor of 18 has been applied. The plots on the left-hand side show the performance in the barrel, the right-hand plots show the performance in the end-cap. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample. The widths of the curves indicate the size of the statistical uncertainties.

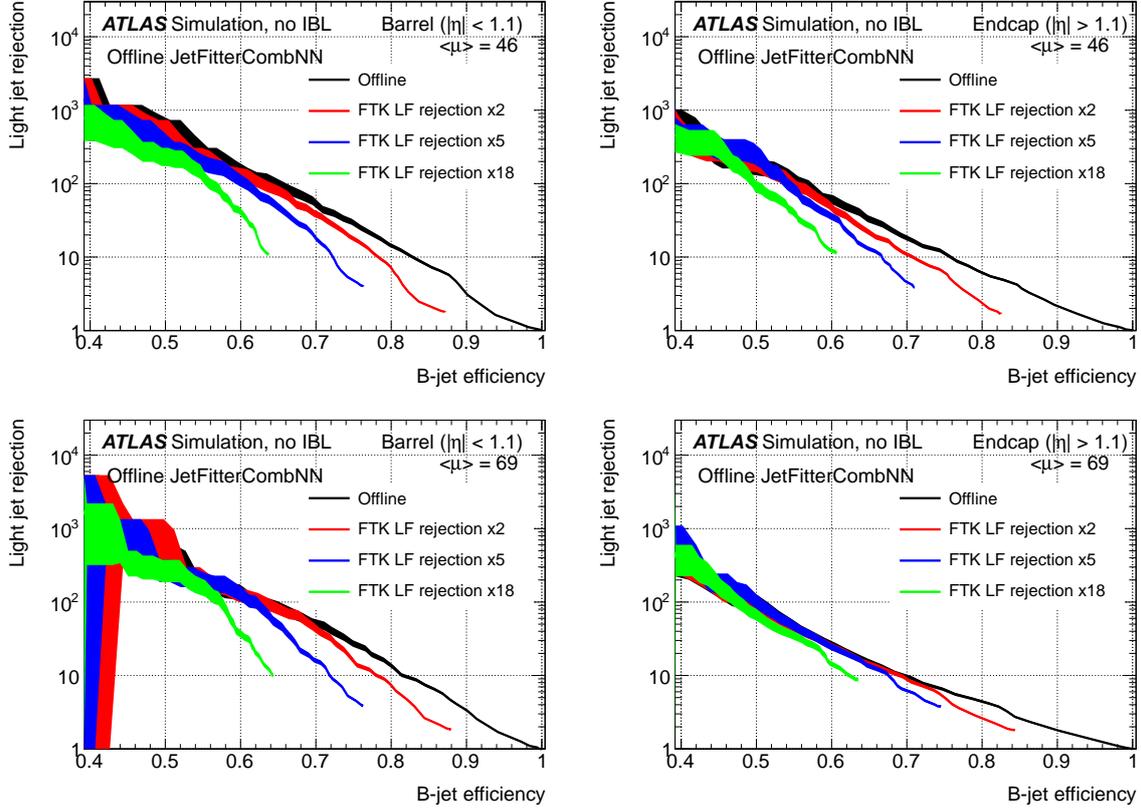


Figure 23: Performance of offline JetCombNN  $b$ -tagging algorithm on samples selected using FTK. The curves show the total light-flavor rejection as a function of the total  $b$ -jet efficiency. The black curve gives the offline performance if no FTK selection is applied, the red curve shows the combined offline and FTK performance where an FTK selection giving a factor of 2 light-flavor rejection has been applied, the blue curve gives the performance where an FTK selection gives a factor of 5 light-flavor rejection has been applied, and the green curve gives the performance when an FTK selection with a light-flavor rejection of a factor of 18 has been applied. The plots on the left-hand side show the performance in the barrel, the right-hand plots show the performance in the end-cap. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample. The widths of the curves indicate the size of the statistical uncertainties.

the offline  $b$ -tagging algorithms as is shown in Figures 24 and 25. For a given offline efficiency the corresponding FTK efficiency is plotted for three different FTK working points. These plots show for the various FTK working points what the trigger efficiency of the event selection would be for jets that are selected on offline analyses with varying efficiency. For example, at a factor of 5 of FTK rejection of light-flavored jets, 90% of offline events passing the 60% JetCombNN  $b$ -tagger in 46 pile-up events would be selected at the trigger level, leading to an overall efficiency of 54%. For multi- $b$  final states, this level of  $b$ -tagging would allow the rate of multi-jet events to be significantly reduced at Level-2, while remaining efficient for signal. This Level-2 rejection could allow for higher Level-1 multi-jet rates, e.g., from lowering of the jet  $p_T$  thresholds.

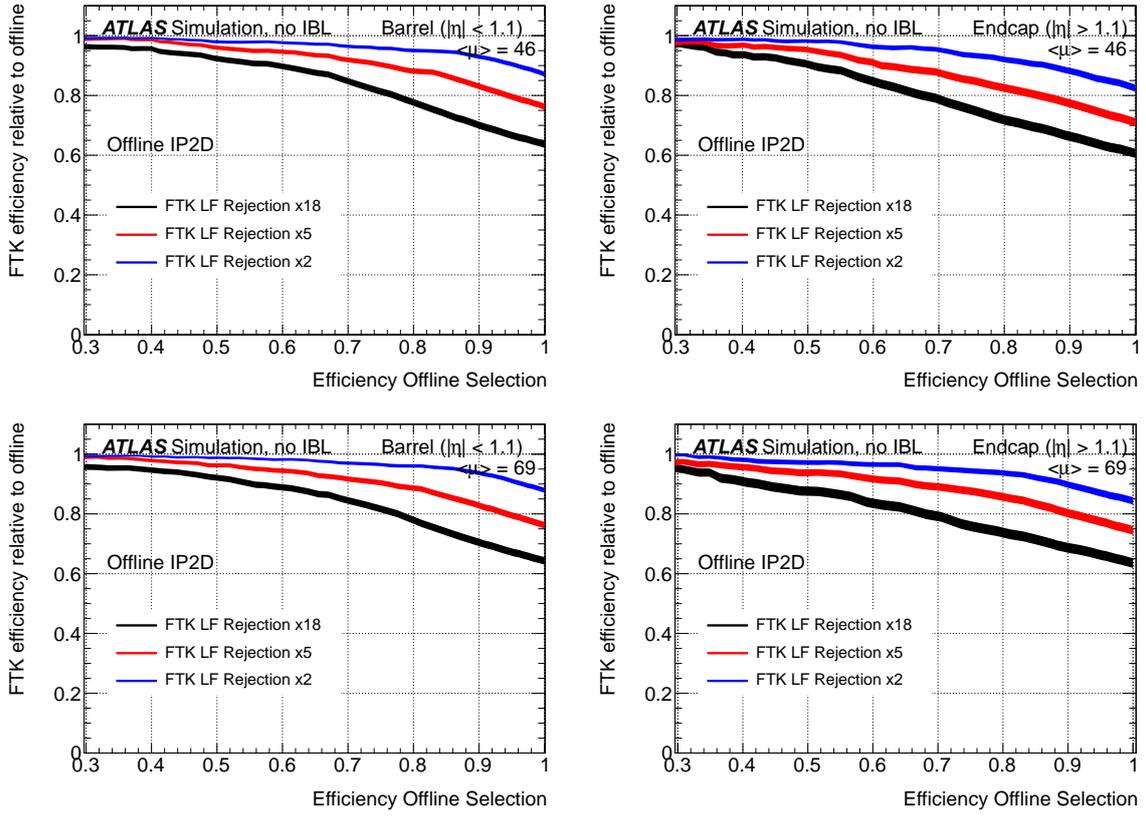


Figure 24: Efficiency of the FTK  $b$ -tagging algorithms as a function of the IP2D offline selection efficiency in the barrel (right) and in the end-cap (left) for 3 different FTK working points. The black curve shows the FTK operating point that gives a factor of 18 light-flavor rejection, the red curve shows the FTK operating point with a factor of 5 light-flavor rejection, and the blue curve the FTK operating point with a factor of 2 rejection. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample. The widths of the curves indicate the size of the statistical uncertainties.

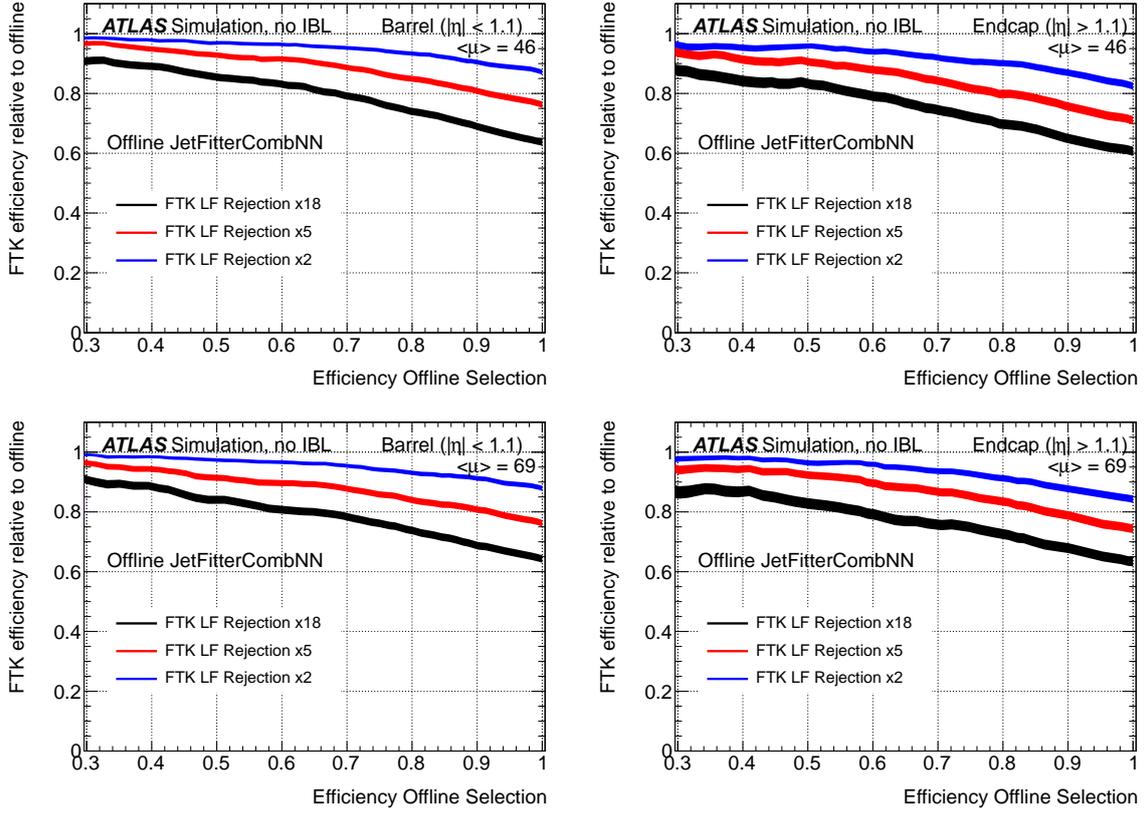


Figure 25: Efficiency of the FTK  $b$ -tagging algorithms as a function of the JetCombNN offline selection efficiency in the barrel (right) and in the end-cap (left) for 3 different FTK working points. The black curve shows the FTK operating point that gives a factor of 18 light-flavor rejection, the red curve shows the FTK operating point with a factor of 5 light-flavor rejection, and the blue curve the FTK operating point with a factor of 2 rejection. The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample. The widths of the curves indicate the size of the statistical uncertainties.

### 3.2.3 Primary Vertex Finding

Primary vertex identification is an important part of offline analyses. There are many pile-up dependent corrections that make use of the number of primary vertices in offline analyses, and an accurate determination of the hard scatter interaction position allows for track selection for a variety of applications. For example, it makes the determination of the Jet Vertex Fraction (JVF), the fraction of tracks associated with a jet coming from the primary vertex, possible. However, because primary vertex finding necessitates full detector tracking and is therefore very CPU intensive, it is not possible to determine the primary vertex in standard trigger processing. The HLT vertex finding is only run on a select subset of event used for determining the beamspot position. During 2012 data-taking primary vertex finding for the beam spot algorithm was the most resource intensive task on the HLT farm. The time spent on pattern recognition for full scan tracking was  $\approx 200$  ms per event, while the fitting of vertices was performed in  $\approx 150$   $\mu$ s.

Because FTK performs full scan tracking in tens of  $\mu$ s, it can improve the vertex finding in a number of ways:

- Provide tracking for the online vertex finding algorithm.
- Provide clusters of tracks in  $z$  to seed the HLT vertex finding algorithm.
- Provide coordinates of the vertex with the highest  $\Sigma p_T$ , which can be assumed to be the hard scatter interaction.
- Provide event level tracking information such as multiplicity and  $\Sigma p_T$  to select events for HLT track fitting and vertex finding.

Here we use FTK tracks in the online vertex finding algorithm and look at the resulting performance.

The algorithm used in this study is the ATLAS HLT beam spot finding and primary vertex reconstruction algorithm [22] adapted for FTK tracks. The primary vertex reconstruction starts by forming a cluster of tracks within a window in  $z$  seeded by the highest  $p_T$  track in the event record. The window width is determined by the average resolution in  $z$ . As each track is associated with a cluster an updated  $z_0$  position is calculated as the average of the track  $z_0$  positions weighted by the track  $\sigma_{z_0}$ . The tracks associated to the cluster are removed from the track list and when this step of clustering is finished the remaining highest  $p_T$  track is used to seed the next clustering pass. This procedure is repeated until there are no unclustered tracks remaining.

The track clusters are then fed into a fast primary vertex fitter that employs a decorrelating measurement transformation to reduce the computation time. The vertex fitter in turn performs iterations on the reconstructed vertex while dropping tracks whose  $\chi^2$  contributions exceed a predefined limit. The resulting vertices are subjected to additional quality cuts and the final positions are reported. The highest sum  $p_T$  vertex identified as the hard scatter vertex. For the FTK primary vertex reconstruction, the window used to cluster tracks is 0.35 mm. The uncertainties on the FTK tracks come from a parametrization of the diagonal elements of the covariance matrix as a function of  $p_T$ .

$t\bar{t}$  events with 46 and 69 additional interactions per crossing are used to study the vertex finding performance. Figure 26 shows the number of primary vertices found by both FTK and offline as a function of the number of interactions in the event. The performance shows a linear correlation with the number of interactions for both FTK and offline. Figure 27 shows the number of FTK tracks associated with vertices which pass the selection criteria, as well as the  $\Sigma p_T$  distribution for primary vertices in the two different pile-up scenarios. When considering all vertices in the event, which are dominantly minimum bias vertices, the 46 pile-up sample shows a slightly higher average number of tracks per vertex and a slightly narrower  $\Sigma p_T$  per vertex distribution. When isolating the hard scatter vertex, the 69 pile-up sample shows slightly more tracks per vertex and roughly the same  $\Sigma p_T$ , indicating some contamination of tracks from other vertices, but the tracks are at low enough momentum so as not to

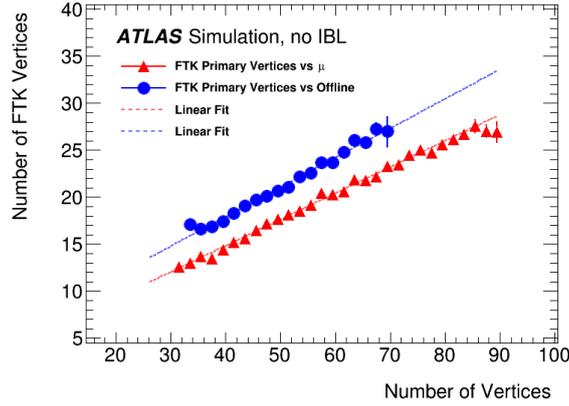


Figure 26: Number of primary vertices reconstructed using FTK tracks vs the number of in-time pile-up interactions (Red) and the number of vertices reconstructed by offline tracking (Blue) in  $t\bar{t}$  MC.

affect the vertex kinematics. The  $\sigma_z$  of the FTK primary vertices, as reported by the final vertex fit, is dependent on the number of tracks associated with the vertex and, by extension, the  $\Sigma p_T$  of the vertex. This behavior can be seen in Figure 28 where the  $\sigma_z$  vs number of tracks and  $\Sigma p_T$  are shown for all primary vertices in the event. No pile-up dependence is seen in these distributions.

In  $t\bar{t}$  MC the hard scatter vertex has, on average, 45.5 associated FTK tracks at 46 pile-up interactions and 46.2 at 69 pile-up, as shown in the bottom of Figure 27. In these samples the hard scatter vertex is well reconstructed with a  $\sigma_z$  as reported by the vertex fit of  $56 \mu\text{m}$  with an uncertainty of  $19 \mu\text{m}$  in the transverse position. The  $\sigma_z$  distribution is shown in Figure 29. The 46 and 69 pile-up samples show comparable performance. The top two plots of Figure 30 shows the dependence of the hard scatter vertex properties on the number of interactions in the event. There is a slight increase in the number of tracks per vertex with increasing interactions per event, and the  $\Sigma p_T$  is relatively insensitive. Also shown is the dependence of the  $\sigma_z$  on the number of interactions in the event and the FTK tracks associated with the hard scatter vertex. There is no dependence of  $\sigma_z$  on the number of interactions and  $\sigma_z$  decreases with increasing tracks per vertex in the full plotted range. Figure 30 also plots the resolution of the vertex  $z$  position with respect to the truth position for both pile-up samples. The distributions are centered at zero and have a width of 0.087 mm (0.096 mm) for 46 pile-up (69 pile-up). The efficiency to match the FTK hard scatter vertex with the MC truth vertex as a function of the number of in-time interactions is also plotted in Figure 30. The efficiency is close to 100% for low number of interactions and decreases to 90% for higher number of interactions. For the inclusive  $\langle \mu \rangle$  sample of 46 the efficiency is  $98.1 \pm 0.4 \%$  as compared to an offline vertex-finding efficiency at  $\langle \mu \rangle = 40$  of 99% [23].

The studies presented here are for a full vertex finding algorithm designed for precise beamspot reconstruction. We are investigating fast histogramming of the  $z$ -positions of tracks for one dimensional vertex finding. This type of algorithm would be most useful for vertex counting, but could also prove effective for identifying the position of the hard scatter vertex. These advanced studies will be presented in the TDAQ Phase I Upgrade TDR.

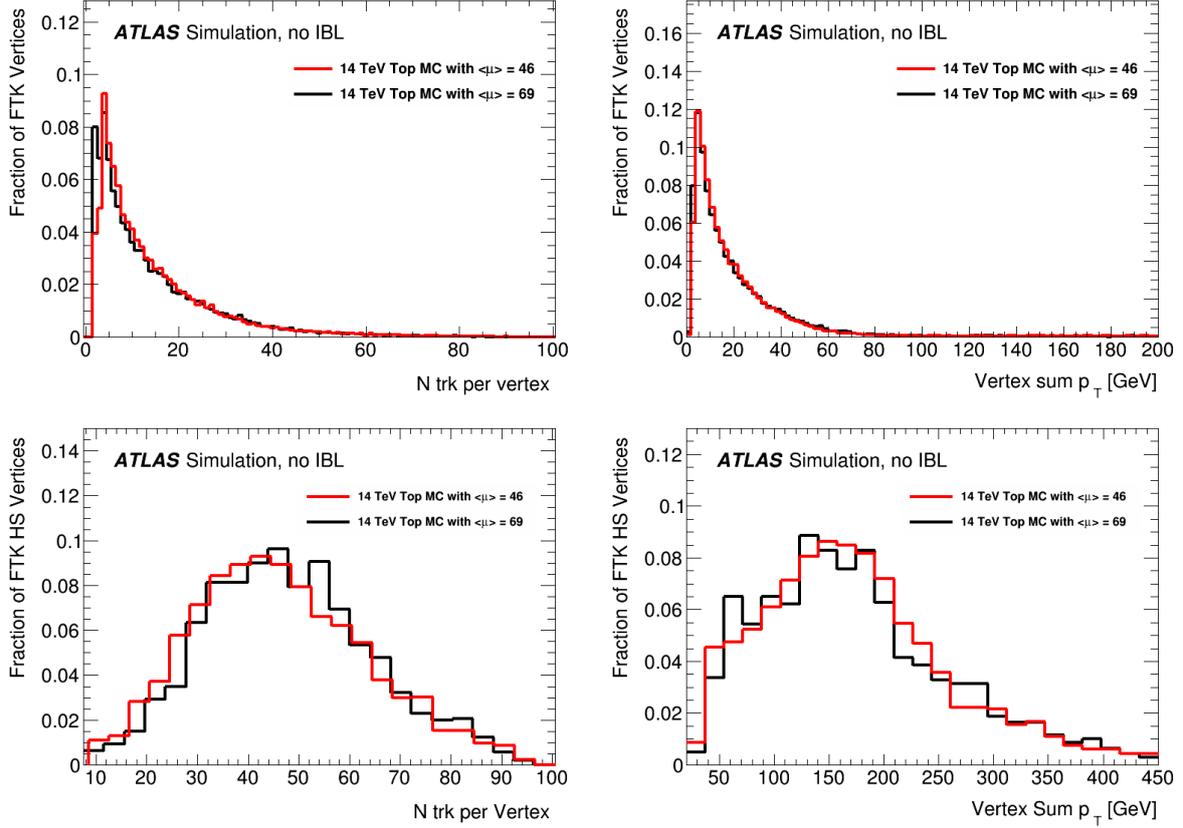


Figure 27: Top Left: Number of FTK tracks per primary vertex in  $\langle \mu \rangle = 46$  (Red) and  $\langle \mu \rangle = 69$  (Black) in 14 TeV  $t\bar{t}$  MC. Top Right:  $\Sigma p_T$  of FTK tracks associated to each primary vertex in  $\langle \mu \rangle = 46$  (Red) and  $\langle \mu \rangle = 69$  (Black) in 14 TeV  $t\bar{t}$  MC. Bottom Left: Number of FTK tracks per hard scatter vertex in  $\langle \mu \rangle = 46$  (Red) and  $\langle \mu \rangle = 69$  (Black) in 14 TeV  $t\bar{t}$  MC. Bottom Right: Sum  $p_T$  of FTK tracks associated to the hard scatter vertex in  $\langle \mu \rangle = 46$  (Red) and  $\langle \mu \rangle = 69$  (Black) in 14 TeV  $t\bar{t}$  MC.

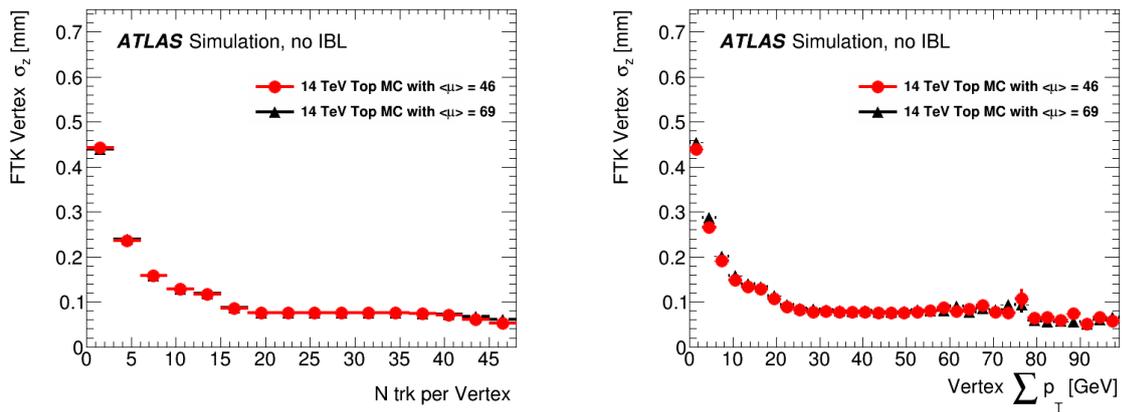


Figure 28: For all primary vertices found using FTK tracks in 14 TeV  $t\bar{t}$  MC. Left: Number of FTK tracks per primary vertex vs the uncertainty in  $z$ -direction of FTK primary vertices. Right:  $\Sigma p_T$  of tracks associated to each primary vertex vs the uncertainty in  $z$ -direction.

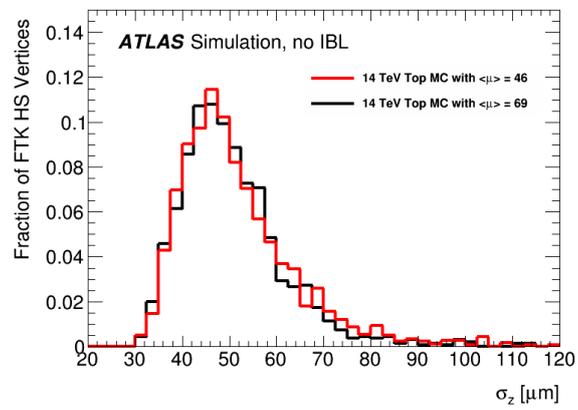


Figure 29: Uncertainty in  $z$ -direction as reported by the vertex fit of the hard scatter FTK primary vertex in  $\langle \mu \rangle = 46$  (Red) and  $\langle \mu \rangle = 69$  (Black) 14 TeV  $t\bar{t}$  MC.

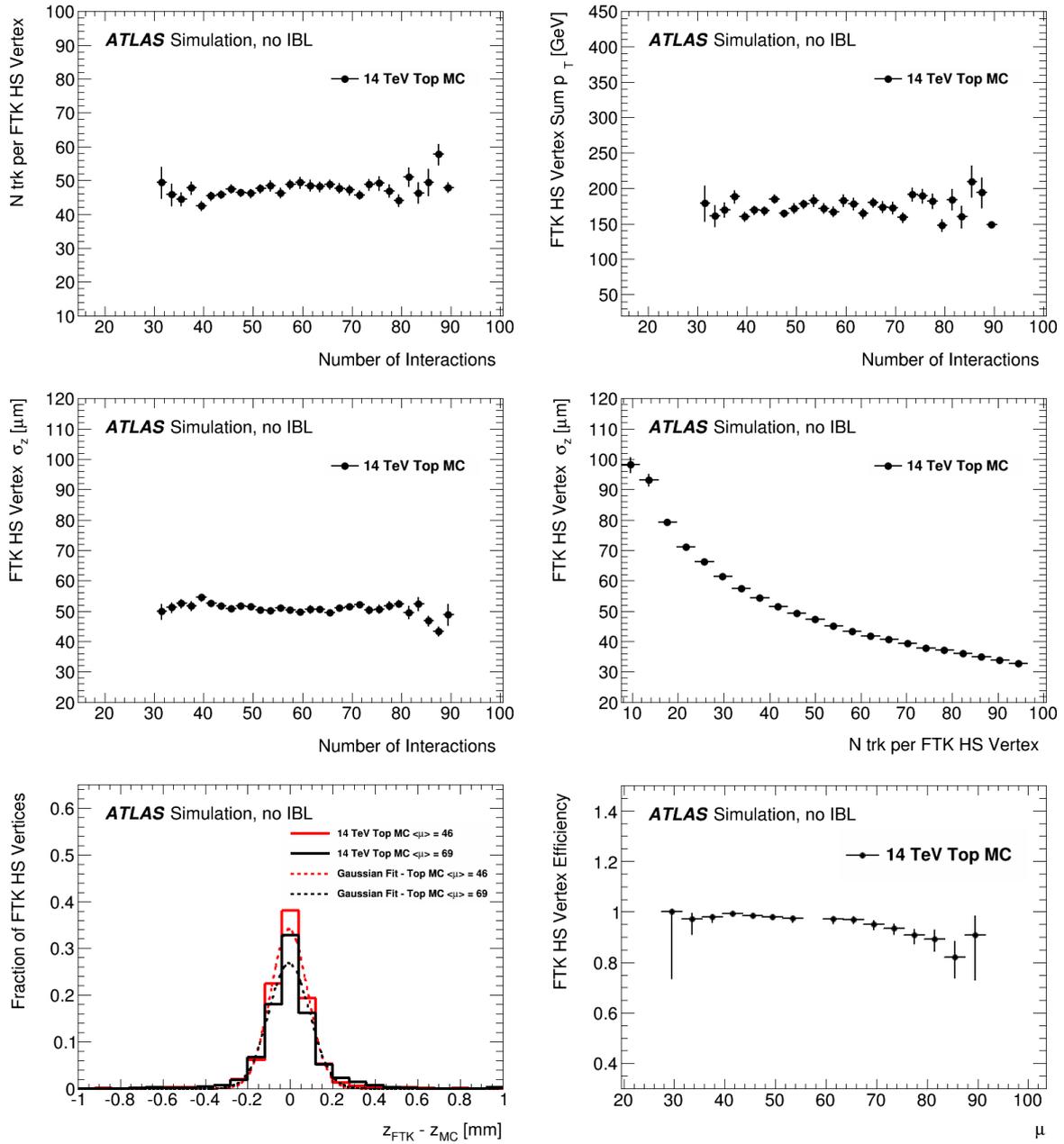


Figure 30: Plots for hard scatter vertices in 14 TeV  $t\bar{t}$  MC. Upper Left: Number of FTK tracks associated with the hard scatter vertex vs the number of in-time pile-up interactions. Upper Right: The sum  $p_T$  of the hard scatter vertex vs the number of in time pile-up interactions. Middle Left: The  $z$ -position uncertainty as reported by the vertex fit vs the number of in time pile-up interactions. Middle Right: The  $z$ -position uncertainty as reported by the vertex fit vs the number of FTK tracks associated with the hard scatter vertex. Lower Left: The resolution of the  $z$  coordinate of the FTK hard scatter vertex compared to MC truth. Lower Right: Efficiency of matching the FTK Hard Scatter vertex to the MC truth vertex in  $z$  as a function of the number of in time interactions.

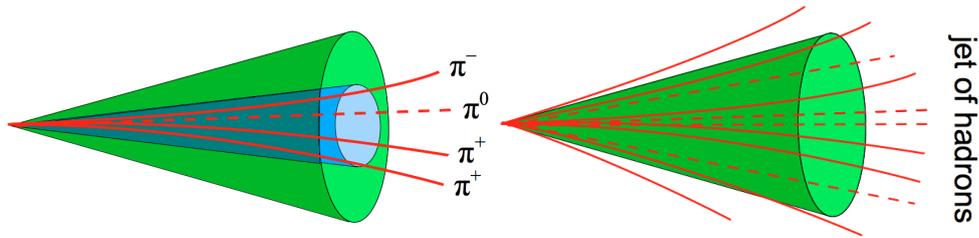


Figure 31: The left image shows a three-prong  $\tau$  decay, with the decay products clustered in the narrow blue cone. The right image shows a jet of hadrons where the particles are distributed throughout the large cone.

### 3.2.4 Hadronic taus

The  $\tau$ , the heaviest known lepton, plays a special role in Standard Model (SM) processes such as Higgs boson decays. It is also important in many searches for physics beyond the SM, such as searches for supersymmetry and for exotic particles. With a mass of 1.8 GeV, the third generation  $\tau$  lepton is the only lepton that can decay into either hadrons or another lepton with a neutrino. Hadronically decaying  $\tau$ s typically decay into one or three charged hadrons, which are called 1-prong and 3-prong decays, respectively. Additional neutral  $\pi^0$  particles may also be present in the decay. Analyses with hadronically-decaying  $\tau$  final states face unique challenges. This decay signature is quite similar to that of jets of hadrons, as is shown schematically in Figure 31 and the extremely large production cross section of multi-jet events make these jets a major background.

To separate  $\tau$ s from jets the decay topology is used. The  $\tau$  signal features 1 or 3 charged particles in a very narrow cone with little or no activity in a surrounding isolation cone, whereas jets typically have activity distributed throughout the isolation cone. When tracking is not available in early trigger levels, calorimeter-based selection requires a narrow isolated jet. This is the principle of the ATLAS Level-1  $\tau$  trigger. The current Level-2 selection first imposes calorimetric cuts in order to reduce the ROI rate to a level acceptable for the tracking algorithms, then tracking is performed on the remaining ROIs [24]. This two step approach results in a loss of efficiency. Here we propose rapid rejection of the QCD background by using FTK tracks at the beginning of the Level-2  $\tau$  selection process without a selection on the calorimetric variables, followed by an Event Filter (EF) selection.

To check the performance we use a sample of VBF  $H \rightarrow \tau\tau \rightarrow lh$ ,  $H \rightarrow \tau\tau \rightarrow hh$  for the signal and a sample of  $WH \rightarrow \ell\nu qq$  for background. The background sample was chosen to ensure the presence of jets in the event which had a similar  $p_T$  spectrum to the signal events, however we find that the majority of the jets passing the  $\tau$  Level-1 ROI are from pile-up. All samples are available with  $\mu = 46$  and  $\mu = 69$  at  $\sqrt{s} = 14$  TeV.

We first check the FTK tracking performance in the  $\tau$  cone and compare to the standard HLT  $\tau$  tracking, TauB, and to the offline tracking. We select tracks that are within  $\Delta R = 0.2$  of an offline  $\tau$  ( $\tau p_T > 20$  GeV,  $|\eta| < 2.49$  and ID selection based on a BDT with a working point, medium, of 50% efficiency) and that pass the following requirements:

- $p_T > 2$  GeV
- $|d_0| < 2$  mm
- $|z_0| < 100$  mm
- All tracks in the  $\tau$  cone have to be within 2mm in  $z_0$  of the leading track

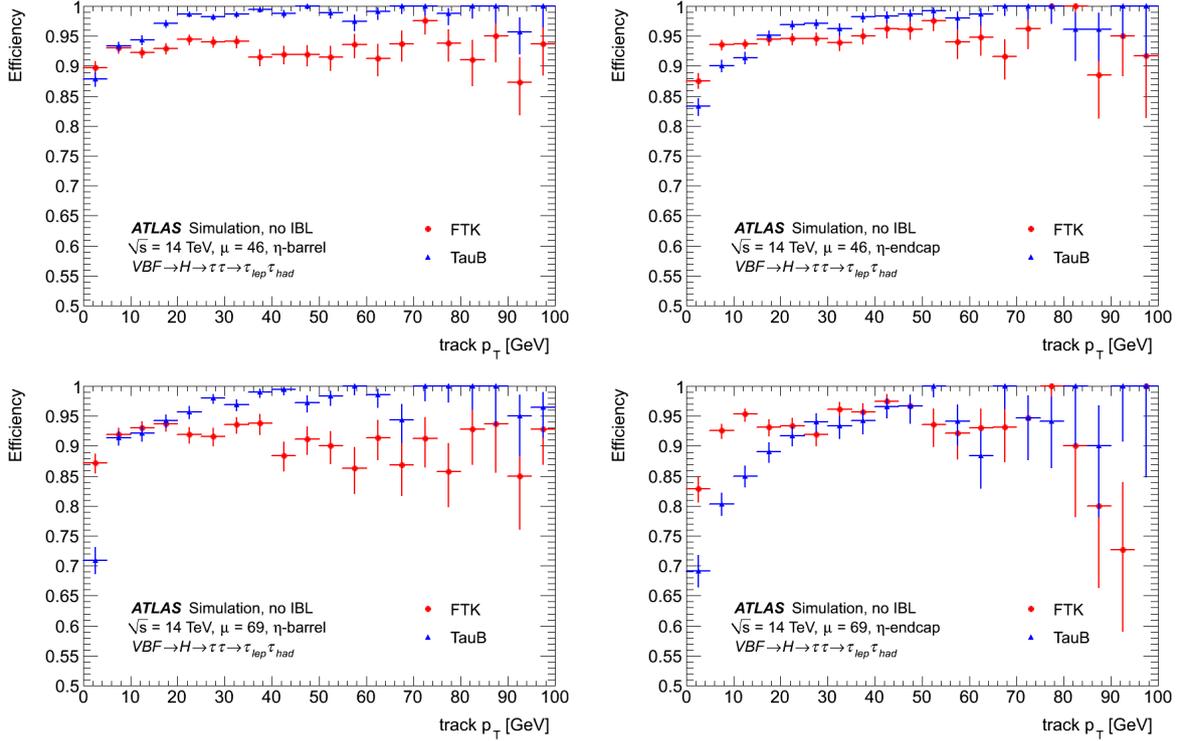


Figure 32: Efficiency of FTK and TauB tracks with respect to offline for events with  $\eta < 1.1$  (left) and with  $\eta > 1.1$  (right). The upper plots are for the  $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

We measure the efficiency of the FTK and TauB tracks with respect to offline tracks and show it as a function of  $p_T$  separately for the barrel and for the end-cap (see Figure 32). FTK tracks are more efficient than TauB tracks at low  $p_T$ . At high  $p_T$  they are less efficient due to the geometric coverage of the pattern banks.

The Level-2 standard calorimeter selection is applied to a Level-1  $\tau$  RoI. The standard Level-2 selection applies a cut on the total  $E_T$  and on the variable  $f_{core}$ , which is the ratio of the transverse energy in a cone of radius 0.1 to the transverse energy in a larger cone of radius 0.4 around the tau direction. As mentioned previously, these cuts are applied first in order to reduce the ROI rate to a level acceptable for the Level-2 tracking. These requirements on  $E_T > 15$  GeV and  $f_{core} > 0.75$  (shown in Figure 33) keep 79% of the signal (that pass Level 1) while rejecting 48% of the background.

The Level-2 tracking reconstruction is run on ROIs passing these cuts and the number of tracks and other tracking information is obtained. The highest  $p_T$  track in the ROI is found and two cones are made: a signal cone with a  $\Delta R < 0.1$ , and an isolation cone with a cone  $\Delta R < 0.3$ . The Level-2 algorithm then requires that at least one and less than 6 tracks are found within the signal cone while no requirement is applied on the number of tracks in the isolation cone. Lastly, other combined tracking and calorimetric variables are used to further reject the QCD background.

Figures 34 and 35 show the distribution of the tracks that satisfy the requirements described above inside the signal cone (within  $\Delta R < 0.1$  of an offline reconstructed  $\tau$  with a medium BDT requirement) for the TauB, FTK and offline tracks, for signal and background respectively. Two clear peaks are visible for the 1-prong and 3 prong decays in the barrel. In the end-caps the effect is less prominent for all track types due to lower tracking efficiency. Figure 36 and 37 shows the distribution of the tracks inside the

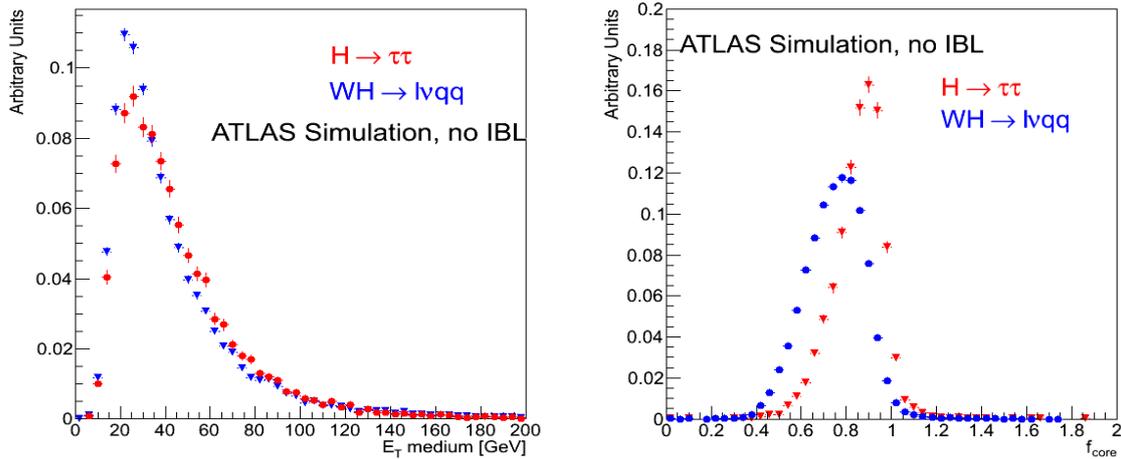


Figure 33: L2 calorimeter variables for the  $H \rightarrow \tau\tau$  and the background sample  $WH \rightarrow \ell\nu qq$ : Minimum  $E_T$  (left) and  $f_{core}$  (right). See the text for the definitions of these variables.

isolation cone (within  $\Delta R < 0.3$  of an offline reconstructed  $\tau$  with a medium BDT requirement) for signal and background.

The use of the tracking information at the start of Level-2 processing would recover the inefficiency lost from the coarse calorimeter requirements. It would also allow EF-style calorimeter and tracking algorithms to be run on the ROIs which pass the tracking cuts, thereby recovering low  $p_T$  efficiency.

To fully exploit the FTK potential we build an algorithm that selects FTK tracks at Level-2 and looks for the highest  $p_T$  track in a cone  $\Delta R = 0.2$  with respect to the Level-1 cluster direction. From these tracks it builds two cones as is done in the standard algorithm. We then require:

- number of tracks in the signal cone equal to 1, 2 or 3
- number of tracks in the isolation cone less than 2

In order to compare to the standard L2 algorithm we measure the  $\tau$  reconstruction efficiency for 1-prong and 3-prong  $\tau$  decays defined as:

$$\epsilon_{sig}^{n-prong} = \frac{\# \text{ of true visible hadronic } \tau \text{ with } n \text{ reconstructed tracks passing L2 and matched to L1}}{\# \text{ of true visible hadronic } \tau \text{ decays with } n \text{ prongs, matched L1}}$$

Figures 38- 39 show the  $\tau$  reconstruction efficiency for  $\mu = 46$  and  $\mu = 69$  pileup respectively. The denominator used to calculate the efficiency is the number of generated  $\tau$  leptons that are successfully matched to a Level-1  $\tau$  cluster (Level-1 Tau, 8 GeV threshold). We observe an improvement in efficiency of 50-70% with respect to the standard algorithm.

We also show the  $\tau$  reconstruction efficiency for 1-prong and 3-prong with respect to offline reconstructed  $\tau$ s (Figure 40) for the  $\mu = 46$  sample. Here the efficiency improvement is only  $\approx 10\%$ . This is due to the fact that the offline BDT and selection is biased toward events that are selected using calorimetric information. Therefore, in order to fully exploit the possible gain in efficiency allowed by FTK, the offline selection will need to be re-optimized. Figures 38- 39 indicate that there is significant room for improvement.

A significant increase in the efficiency that also had a significant increase in fake rate would not be useful for the trigger. Here the fake rate is defined as the ratio of the number of jets passing the  $\tau$

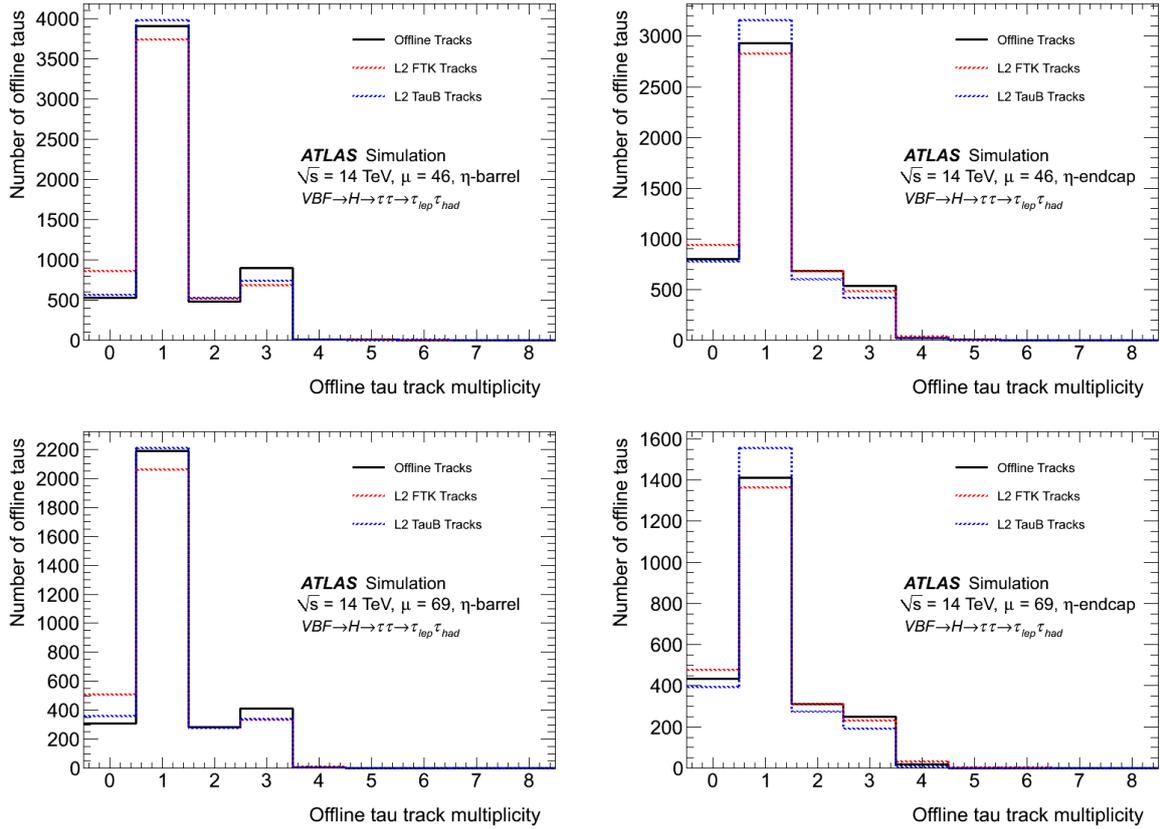


Figure 34: Number of tracks inside the signal cone ( $\Delta R = 0.1$  from an offline reconstructed  $\tau$  with a medium BDT requirement) for offline tracks (black), tauB track( blue) and FTK tracks (red) for  $\eta < 1.1$  (left) and  $\eta > 1.1$  (right). The upper plots are for the  $H \rightarrow \tau\tau$   $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

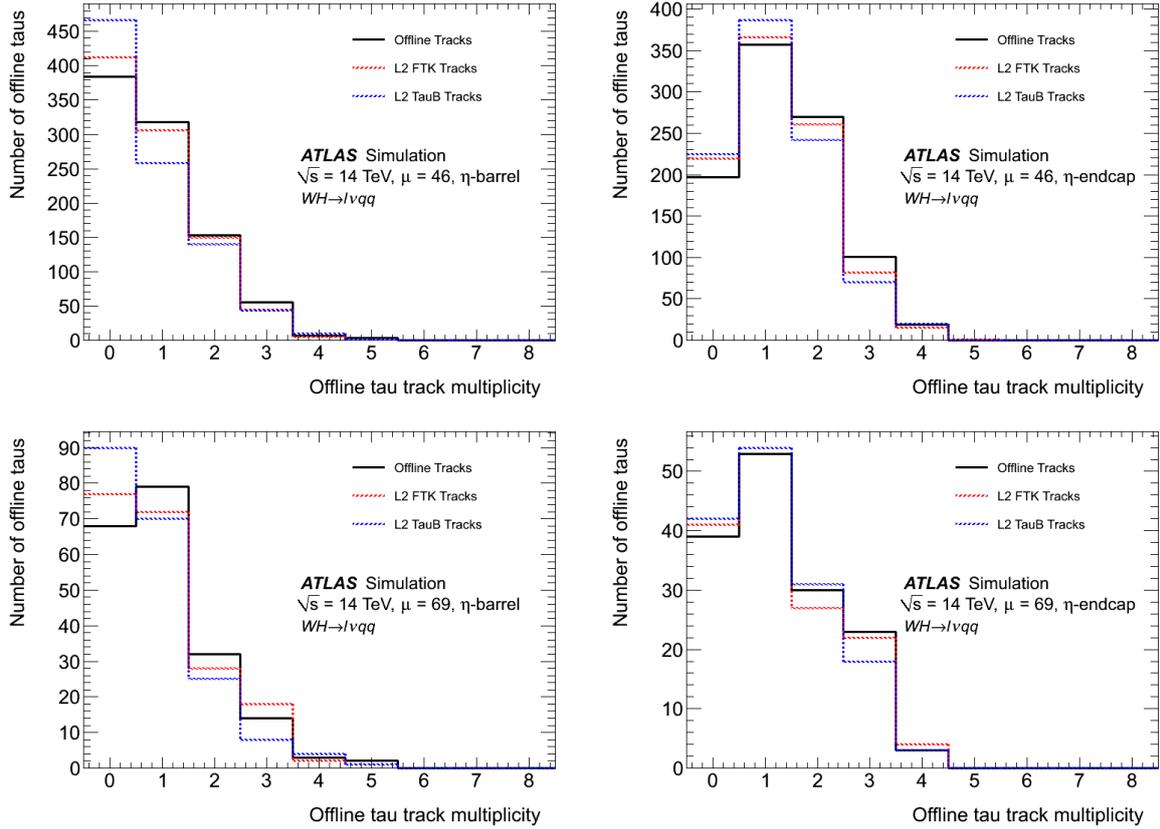


Figure 35: Number of tracks inside the signal cone ( $\Delta R = 0.1$  from an offline reconstructed  $\tau$  with a medium BDT requirement) for offline tracks (black), tauB track (blue) and FTK tracks (red) for  $\eta < 1.1$  (left) and  $\eta > 1.1$  (right). The upper plots are for the  $WH \rightarrow lvqq$   $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

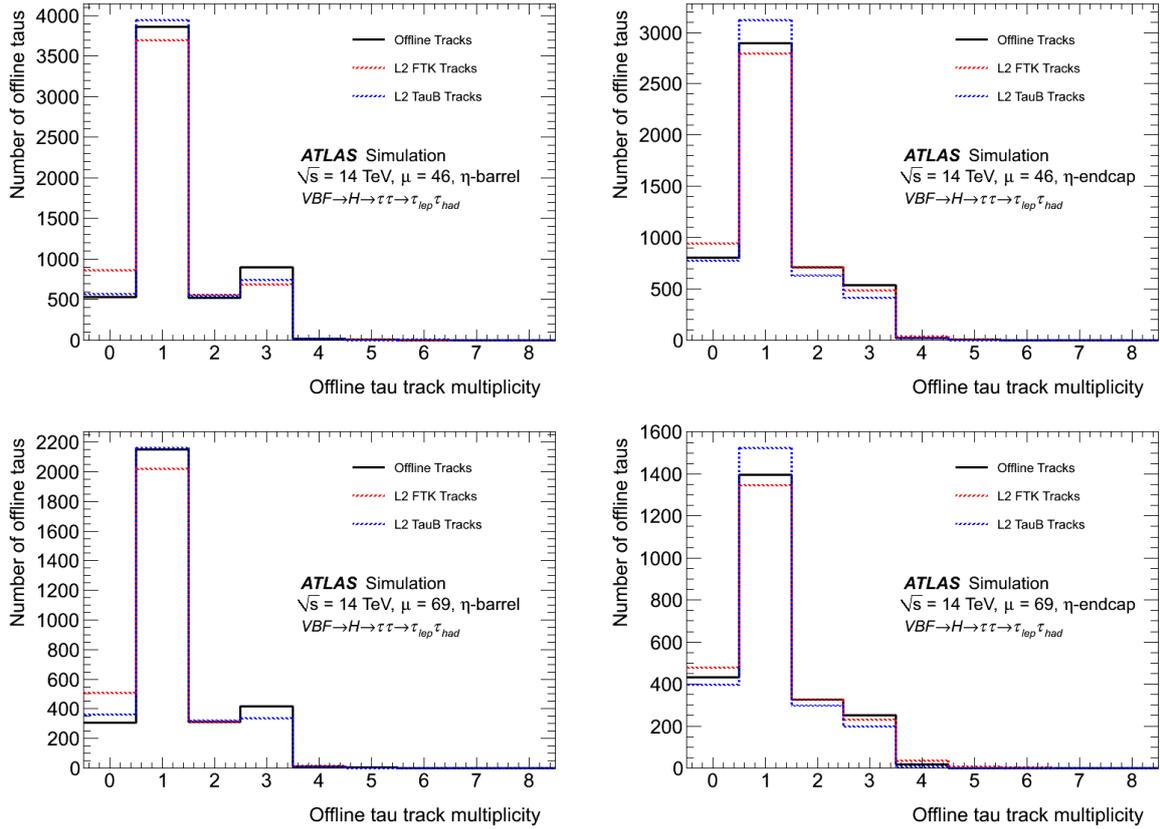


Figure 36: Number of tracks inside the isolation cone ( $\Delta R = 0.3$  from an offline reconstructed  $\tau$  with a medium BDT requirement ) for offline tracks (black), tauB track( blue) and FTK tracks (red) for  $\eta < 1.1$  (left) and  $\eta > 1.1$  (right). The upper plots are for the  $H \rightarrow \tau\tau$   $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

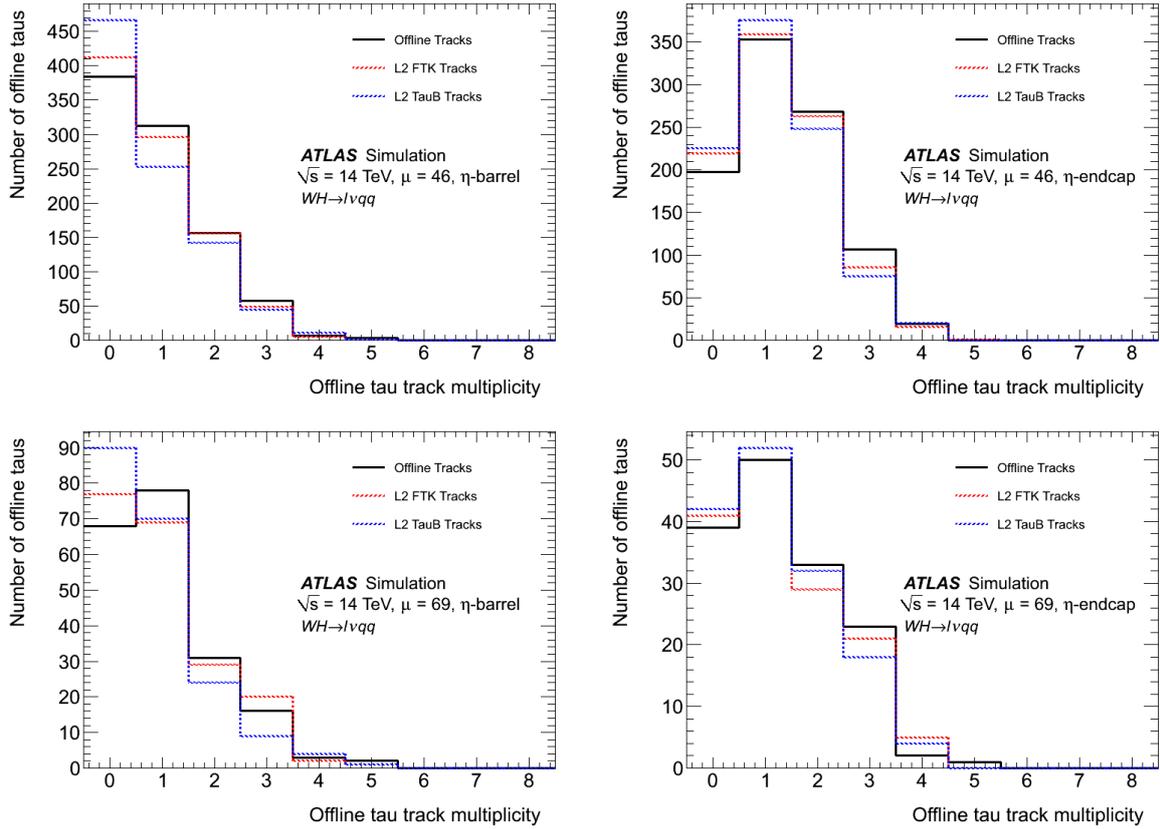


Figure 37: Number of tracks inside the isolation cone ( $\Delta R = 0.3$  from an offline reconstructed  $\tau$  with a medium BDT requirement ) for offline tracks (black), tauB track( blue) and FTK tracks (red) for  $\eta < 1.1$  (left) and  $\eta > 1.1$  (right). The upper plots are for the  $WH \rightarrow lvqq$   $\mu = 46$  sample and the lower plots are with the  $\mu = 69$  sample.

reconstruction at L2 to the total number of reconstructed jets that pass Level 1. It is shown in Figure 41 for the  $WH \rightarrow \ell\nu q\bar{q}$  sample at  $\mu = 46$  and in Figure 42 for  $\mu = 69$ . These plots show that FTK could reduce the background by a factor of 10 (with respect to Level 1) using only FTK track cuts at Level-2. This level of rejection is sufficient to allow the EF algorithms to run on the ROIs that pass the track cuts. Then the more refined variables the EF calculates would be used to further bring down the jet background.

Several improvements are expected in the  $\tau$  sector. In particular, this study improves the efficiency by simply applying a cut on the number of tracks in the signal and isolation cone. We could exploit further the FTK capabilities by implementing a  $\tau$ -vertex association to reduce the pileup dependence and increase further the background rejection. Additionally we could use other track variables to implement a high efficiency multivariate selection before EF. All these possibilities will be studied for the TDAQ Phase I Upgrade TDR.

### 3.3 Case Study: Higgs to $\tau\tau$ channel improvements

In the previous section we have shown the performance of FTK in  $\tau$  tagging. We obtain a high efficiency with respect to truth  $\tau$ s while imposing a strong rejection over QCD background at the beginning of Level 2 processing. In order to estimate the increase in the yield of  $H \rightarrow \tau\tau$  events, we use the  $\mu = 46$  pileup samples and we run the full trigger chain. At Level-2 we first use the FTK optimized algorithm. The predicted triggers<sup>5</sup> for 2015 are L1\_TAU15\_MU15 ( 5 khz) and 2\_TAU11\_JET60 ( 5 khz). These are not available yet in the menu so we start from similar L1 triggers with the closest rate:

- L1\_TAU11\_MU10 ( 10 khz) for  $H \rightarrow \tau\tau \rightarrow lh$
- L1\_2TAU8\_TAU15\_J15 ( 5.5 khz) for  $H \rightarrow \tau\tau \rightarrow hh$

The offline analyses require the presence of a reconstructed  $\tau$  that passes a BDT MEDIUM requirement (that is 50% efficient) and a BDT for the rejection of the electrons and muons [25]. The  $h - lep$  analysis requires one  $\tau$  with  $p_T > 30$  GeV and one electron or muon with  $p_T > 25$  GeV. The analysis with 2 hadronically decaying  $\tau$ s requires 2 offline  $\tau$ s with  $p_T > 25$  GeV with an additional lepton veto.

We estimate that the improvements at Level 2 would translate into a 22% increase in acceptance in the  $h - lep$  analysis and 28% for the  $h - h$  analysis. We stress that these numbers should benefit from a re-optimization of the offline selection algorithms. The optimization would be able to take into account the 50-70% increase of the events that pass an FTK-based Level-2 trigger in the  $\tau$   $p_T$  region of interest for the light Higgs. The increase in efficiency is even more significant with 69 simultaneous interactions where the improvement from a track based selection reaches a factor of two (see Fig.39 for the 3-prong selection). These new events have a different calorimetric/tracking composition compared to the ones selected by the current trigger algorithm, and the offline analysis could exploit these new features to enrich the final Higgs samples. The Level-2 and Event Filter algorithms should also be re-optimized for the FTK selection. This could potentially lead to lower Level-1  $\tau$   $p_T$  thresholds which would directly impact the sensitivity of the analyses.

### 3.4 Discussion of selected FTK use cases

The use of FTK in the HLT will be discussed extensively in the forthcoming Trigger and Data Acquisition Phase I Upgrade Technical Design Report. This section discusses several areas under study within the ATLAS trigger group, organized by Trigger subgroups.

---

<sup>5</sup>Triggers are represented in the form <level>.<multiplicity><type><threshold>, e.g. L1\_2TAU8 requires 2 level-1 Tau objects with an 8 GeV threshold. Further level-1 trigger items in the name separated by an underscore are “anded” in the trigger.

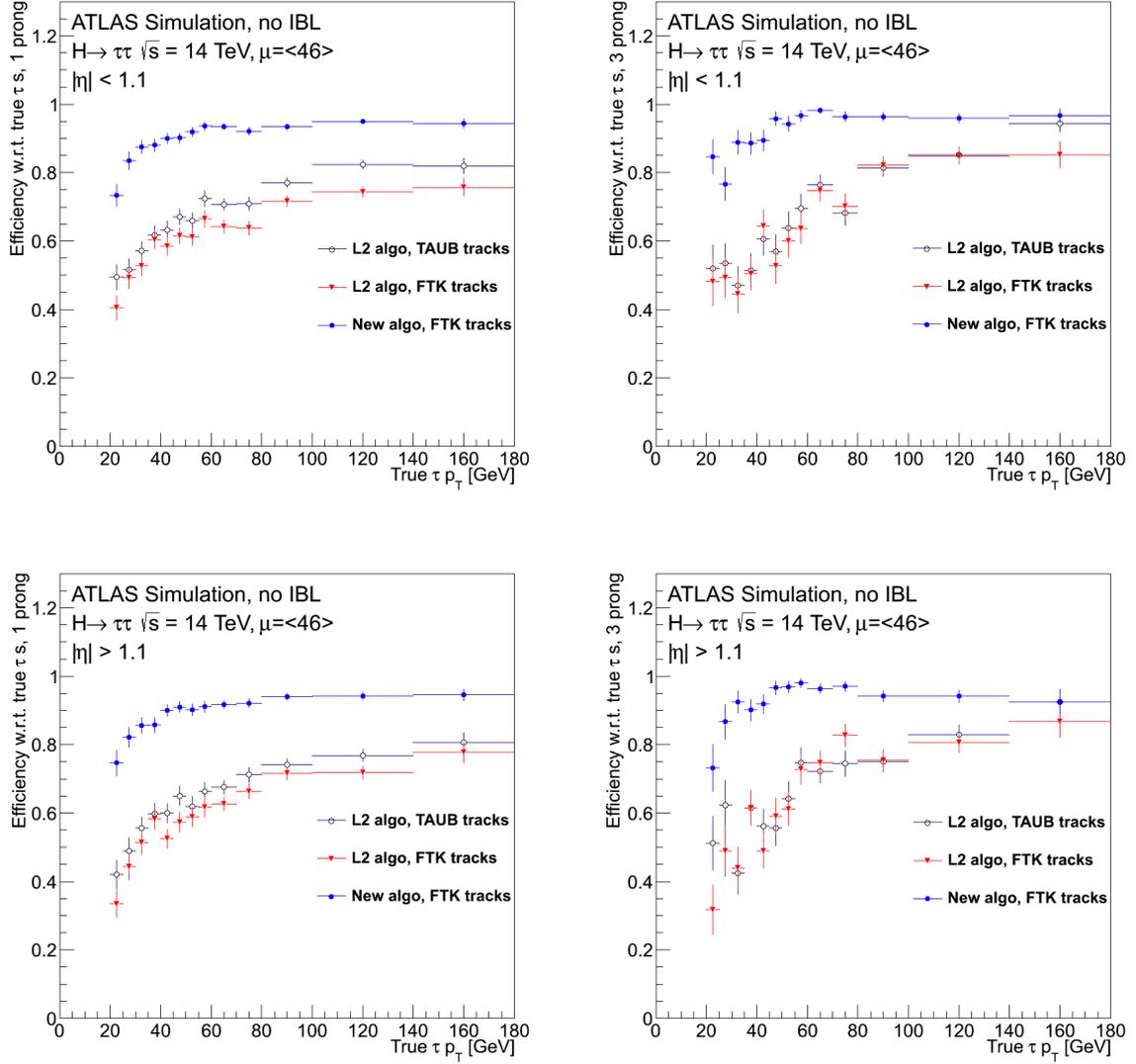


Figure 38: Efficiency of the Level 2 algorithms with respect to true  $\tau p_T$  for 1-prong (left) and 3-prong case (right). The upper plots are for the barrel region ( $\eta < 1.1$ ) and the lower ones are for the end-cap ( $\eta > 1.1$ ). Three working point are shown, black is the efficiency for the standard L2 algorithm, the red shows the efficiency for the Level 2 style algorithm using FTK tracks and the blue shows the efficiency for the FTK track based-algorithm.

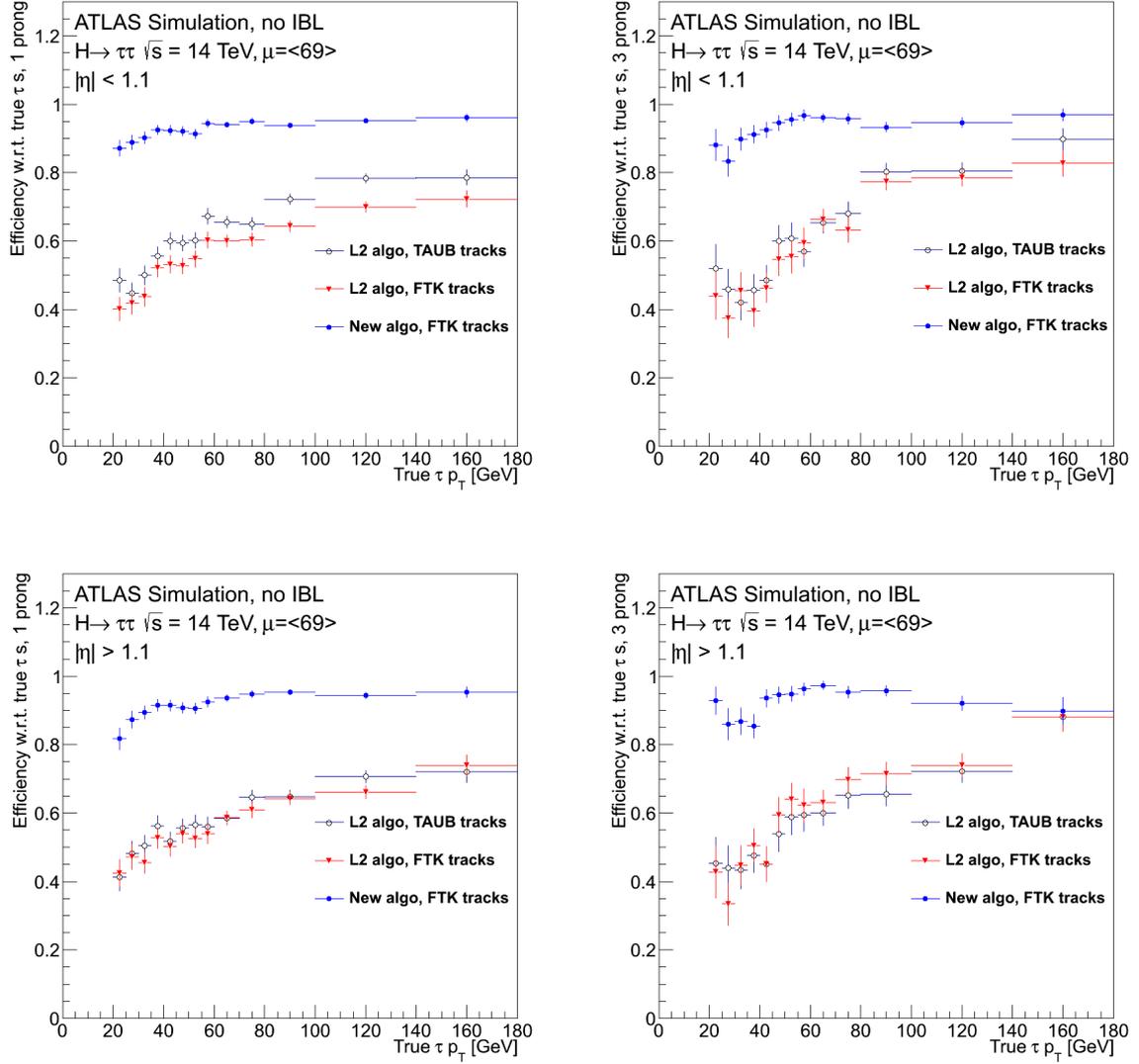


Figure 39: Efficiency of the Level 2 algorithms with respect to true  $\tau p_T$  for 1-prong (left) and 3-prong case (right) for the 69 pile-up sample. The upper plots are for the barrel region ( $|\eta| < 1.1$ ) and the lower ones are for the end-cap ( $|\eta| > 1.1$ ). Three working point are shown, black is the efficiency for the standard Level 2 algorithm, the red shows the efficiency for the Level 2 style algorithm using FTK tracks and the blue shows the efficiency for the FTK track-based algorithm.

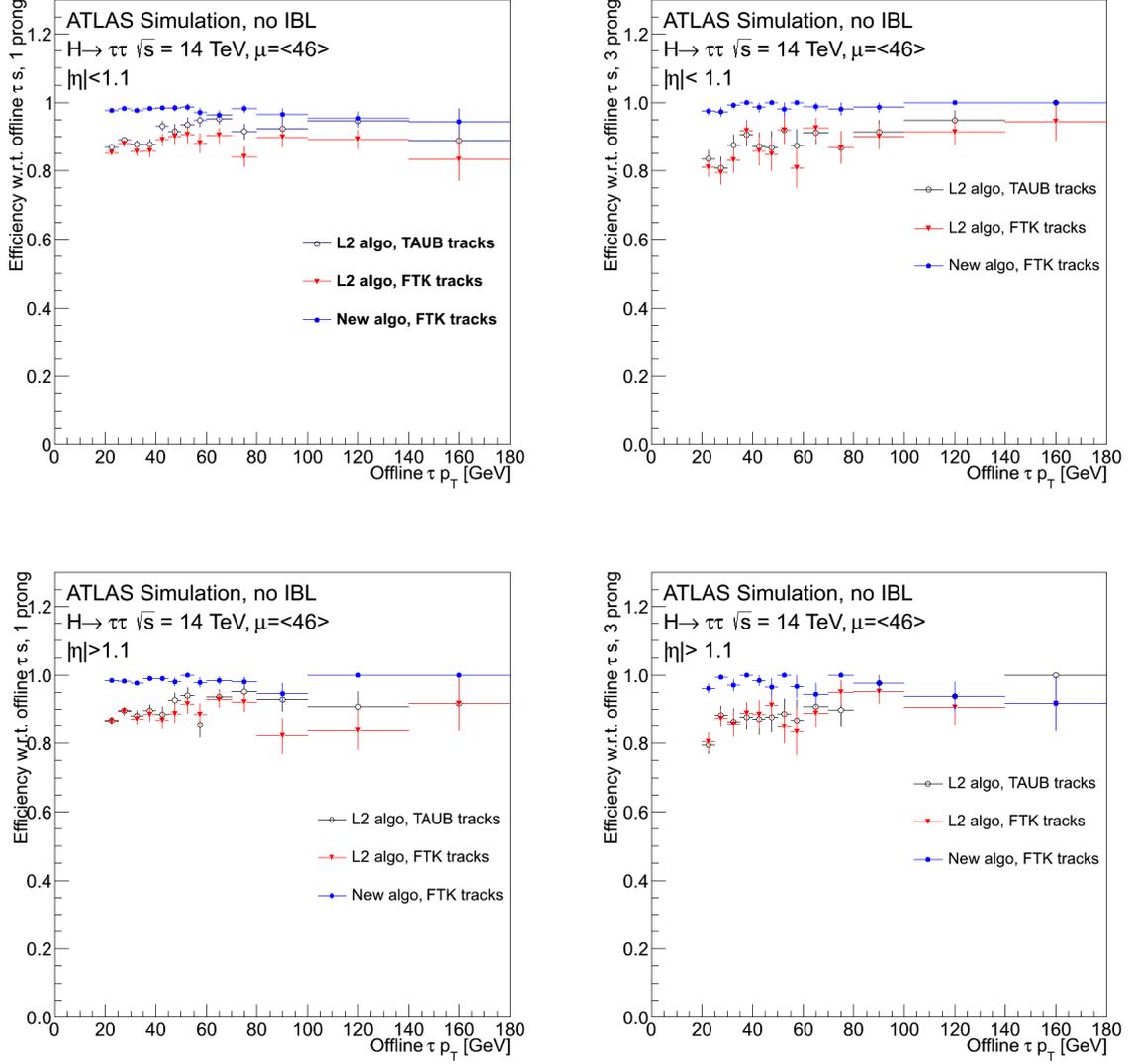


Figure 40: Efficiency of L2 w.r.t. offline  $\tau p_T$  for 1-prong (left) and 3-prong case (right) for the 46 pile-up sample. The upper plots are for the barrel region ( $\eta < 1.1$ ) and the lower ones are for the end-cap ( $\eta > 1.1$ ). Three working point are shown, black is the efficiency for the standard L2 algorithm, the red shows the efficiency for the L2 style algorithm using FTK tracks and the blue shows the efficiency for the FTK track-based algorithm. The offline taus are required to have exactly three tracks in the 3-prong case.

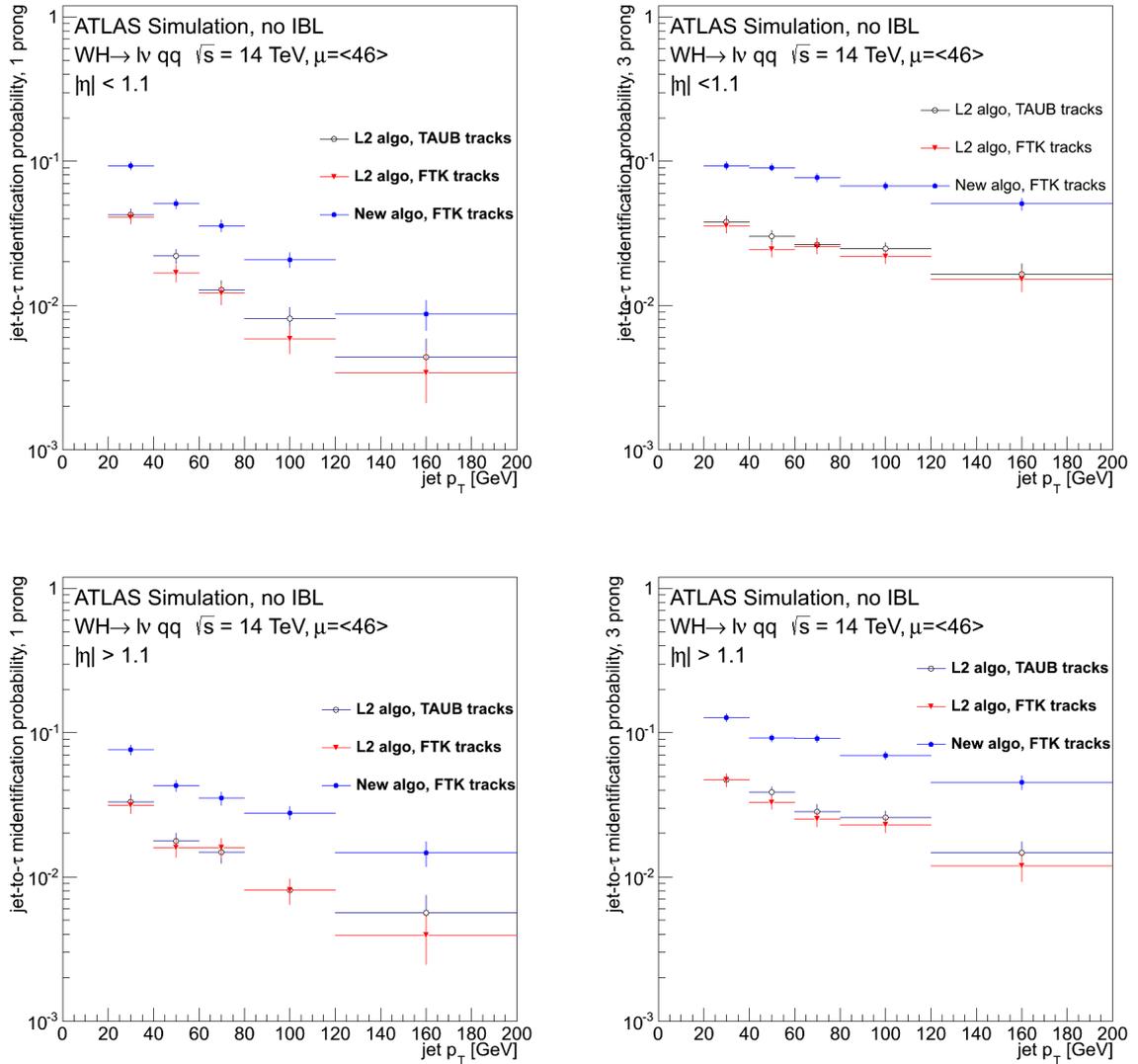


Figure 41: Jet to  $\tau$  misidentification probability at Level 2 for 1-prong (left) and 3-prong case (right) at a pile-up of 46. The upper plots are for the barrel region ( $\eta < 1.1$ ) and the lower ones are for the end-cap ( $\eta > 1.1$ ). Three working point are shown, black is the fake rate for the standard Level 2 algorithm, the red shows the fake rate for the Level Level 2 style algorithm using FTK tracks and the blue shows the fake rate for the FTK track-based algorithm.

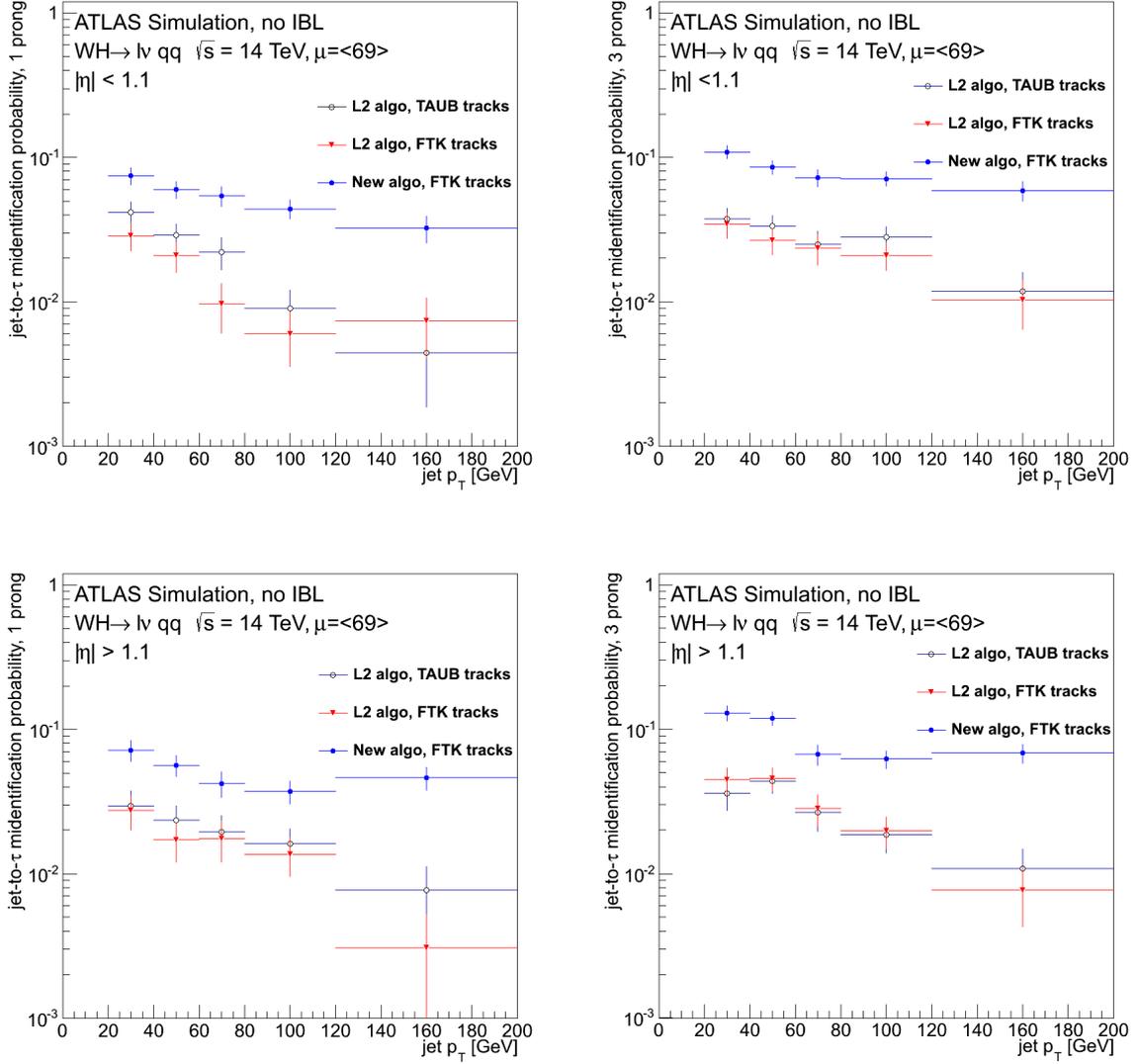


Figure 42: Jet to  $\tau$  misidentification probability at Level 2 for 1-prong (left) and 3-prong case (right) at a pile-up of 69. The upper plots are for the barrel region ( $\eta < 1.1$ ) and the lower ones are for the end-cap ( $\eta > 1.1$ ). Three working point are shown, black is the fake rate for the standard Level 2 algorithm, the red shows the fake rate for the Level Level 2 style algorithm using FTK tracks and the blue shows the fake rate for the FTK track-based algorithm.

**The Jet trigger group** is considering use of FTK tracks in three different ways. The first is to apply energy corrections based on the number of primary vertices found by FTK. These corrections would bring the trigger jets in closer alignment with the jets used in the offline. They will also use the primary vertex information to apply jet vertex fraction cuts. This is useful for multi-jet triggers as well as selecting jets for  $b$ -tagging. Use of FTK tracks for  $b$ -tagging will also be extensively studied in order to improve on the preliminary results shown here. Either a sophisticated algorithm based on FTK tracks could be used, or a simple FTK-based tagger could provide early rejection, freeing the HLT to use a near-offline quality tagger on the jets passing the selection. Lastly, there are studies of using FTK tracks to recover efficiency for low  $p_T$  or close-by jets.

**The Muon and Electron trigger groups** are both investigating using FTK vertices to control the pile-up dependence of calorimeter isolation. The Muon slice is also investigating using FTK tracks for their track-based isolation due to the increased flexibility of cone sizes over ROI based tracks. Additionally, the muon group is exploring the possibility of recovering Level 1 trigger inefficiencies for dimuon events by looking for an isolated high  $p_T$  FTK track and using it to seed a search for matching MDT segments. The electron group will explore the possibility of correcting electron shower shapes for pile-up contributions using FTK information.

**The Missing Transverse Energy trigger group** is investigating using track and/or primary vertex information to improve the Level 2 missing energy resolution.

**The Tau trigger group** will further optimize the use of FTK tracks in the tau chains following the discussion in Section 3.2.4. They may also be able to correct calorimeter shower shapes for pile-up using FTK tracks. Additionally they will explore the possibility of recovering inefficiencies due to the high Level 1 thresholds by searching for isolated one or three track objects in events triggered by leptons. This is of particular interest for associated Higgs production channels where the W or Z boson decays to an electron or muon and the Higgs decays to a pair of  $\tau$ s.

**The Beamspot group** is interested in using FTK tracks either a) as a direct input to vertex finding, b) by refitting them to improve the parameter resolution before input to the vertex or c) by counting the number of tracks or vertices to select the best events for processing by the standard HLT tracking and vertex finding.

In addition to the specific studies mentioned above, we will study the effect of refitting FTK tracks using the Level-2 tracking software to improve the track resolution. This could be used in specific cases, such as when doing  $\tau$ -finding or  $b$ -tagging.

## 4 Hardware

An overview of the FTK system architecture is presented in section 2.1 (see especially Fig. 3). In the following, we give more details on the hardware implementation of this architecture.

### 4.1 The FTK System: a data driven pipeline

FTK has a very large number of devices organized in pipelines connected by thousands of specialized links: 16400 dedicated custom chips (AM chips) that perform pattern matching and 2000 FPGAs for all other functions. Figure 43 shows the organization of this large computing system. The black arrows show the connections on fibers and the yellow rectangles show high frequency connections between mezzanine cards and motherboards or between two boards in the same crate slot, like the Associative Memory board and the auxiliary card behind it.

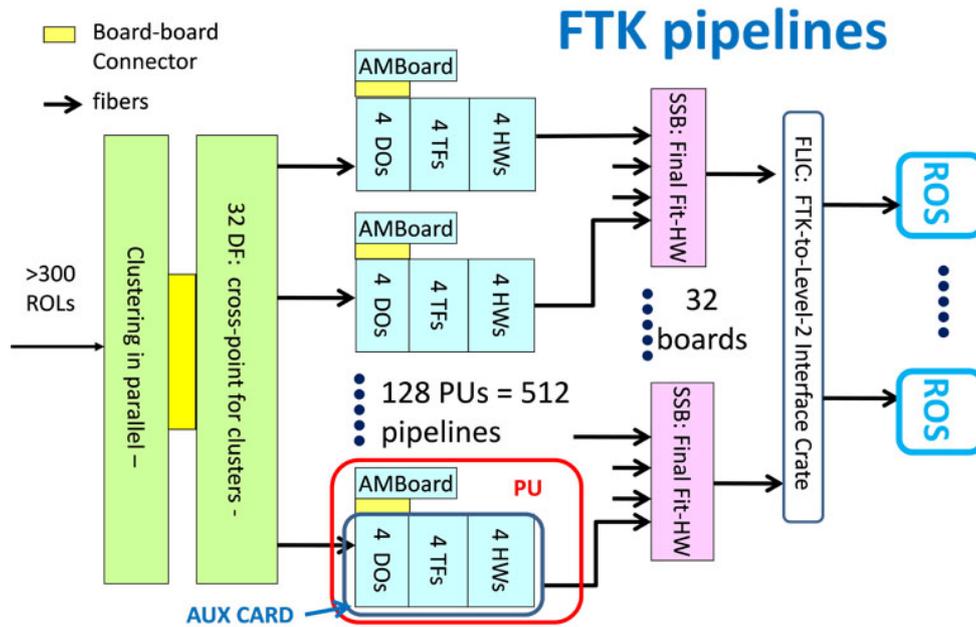


Figure 43: FTK as a set of 512 pipelines up to the final 12-layer fit in the SSBs.

#### 4.1.1 Data flow between boards and the Hold mechanism

A simple communication protocol is used for data transfers in FTK. The data flow through serial links connecting one source to one destination. The protocol is a simple pipeline transfer driven by control words, for example idle words and alignment words. An 8b/10b encoding is used in the serial data stream in order to provide effective error detection, *i.e.* a 32-bit word is transmitted as 40 bits. The idle word is transmitted when no valid data is available. On each link the FTK information is transmitted in words (FTK words) whose format depends on the kind of information being processed in that portion of the FTK pipeline. Alignment words are periodically transmitted between FTK words. Input FTK words in each processing step of the pipeline are pushed into a de-randomizing FIFO buffer. All the words that are not identified as control words are pushed into the FIFO (write-enable signal asserted to the FIFO). The FIFO is popped by whatever processor sits in the destination device. The source and destination devices are two separate logic functions in the pipeline which can be on separate boards or even be two functions in the same large FPGA, for example between the Data Organizer and the Track Fitter (see section 4.5).

To maximize speed, no handshake is implemented on a word-by-word basis. A hold signal (HOLD) is used instead as a loose handshake to prevent loss of data when the destination is busy. If the destination processor does not keep up with the incoming data, the FIFO produces an Almost Full signal that is sent back to the source as the HOLD signal. The source responds to the HOLD signal by suspending data flow. Using Almost Full instead of Full gives the source enough time to stop. Since the source is not required to wait for an acknowledge signal from the destination device before sending the next data word, data can flow at the maximum rate compatible with the link bandwidth even when transit times are long. The standard FTK clock frequency is 100 MHz for 16-bit words, which corresponds to 2 Gb/s for serial transmission. Some links run at transmission speeds up to 6 Gb/s.

The HOLD signal travels in the direction opposite to that of the data, from destination to source. It is transmitted as a single ended signal when two devices are on the same board, like the Data Organizer and Track Fitter on the AUX card, or when two boards are directly connected by a connector, for example the FTK\_IM input mezzanine and its Data Formatter motherboard, or between the AUX card and the

Associative Memory board in the Processing Unit. When two devices are far apart and the connection is made by fibers, the Hold mechanism is implemented with the XON/XOFF protocol (see section 4.2), which we tested on the S-LINKs between the Dual-Output HOLA and the FTK\_IM (September 2011, tests performed at point-1 before the Dual-Output HOLA production).

When the FTK information is organized into a packet of words, a specific bit in the FTK word is defined as an End Packet bit (EP). The EP bit marks the last word of the packet. The End Event (EE) word separates data belonging to different events on each transmission link. It is marked by a specific control word and signifies the end of the data stream for the current event. The EE word can be expanded to a packet if the End Event information requires more than one word. Each device will assert an EE word or EE packet in its output stream after it has received an EE word or EE packet in each input stream and it has no more data to output. The EE word has a special format used to tag the event and to report the parity and any error flags.

Any device or function in the pipeline can have many input streams, and events arriving from different sources have to be synchronized since the same event can arrive on different inputs at different times. The device inputs have FIFOs for this purpose whose depth covers fluctuations in the device processing time and arrival time of input data. In the following sections critical points will be identified for which particularly deep FIFOs are assigned.

When the device starts to process an event, words are popped from the input FIFOs for the various input streams. The data is processed and results are sent to the output stream. When the End Event word is received on an input stream, no additional data is read from that FIFO until the End Event word is received on all the other input streams. The device can issue a Hold signal if a FIFO becomes almost full, causing back pressure, but the goal is to have the FIFO deep enough to limit back pressure as much as possible. The End Event words from the input streams are checked to make sure they contain the same event tag. Upon detection of different event sequences, a severe error is issued and the system must be resynchronized. Once the event is completely read out from the input FIFOs and the device finishes its processing, the event is closed by sending an End Event word to the output with the same event tag as in the input streams.

The standard ATLAS event headers and trailers propagate with the FTK event and are sent by the FLIC to the High Level Trigger ROSs.

#### **4.1.2 The numbers of roads and fits: pattern bank optimization**

As noted above, data is transferred from a source device to a destination device in the pipeline (Figure 43) on serial links typically operating at 2 Gb/s, to transfer 16-bit words at a 100 MHz rate. For a level-1 trigger rate of 100 kHz, the average event transmission time on a link cannot exceed 10  $\mu$ s and thus the average number of words per event per link cannot exceed 1000. The degree of parallelization in FTK is designed to satisfy this requirement.

The FTKSim simulation program described in Section 5 processes complete events produced by the full ATLAS detector simulation and creates the same list of tracks that will be produced by the FTK hardware. Since FTKSim produces all the event information at each point in the pipeline, we can analyze the average size of the event data in each pipeline connection and its fluctuations. The design specifications for each board are based on this information as described in the following sections.

In the FTK system, optimizing the pattern recognition resources versus the track fitting resources was tuned during the design. Because of the finite width of a pattern, a road may contain silicon hits from different particles. This produces fake roads through which no single particle fully passes. Fake roads, whose track candidates will all be rejected when fit at full hit resolution, are produced at a rate that increases with increasing pattern width and increasing hit occupancy. The number of roads found can become much greater than the actual number of tracks at high detector occupancy. Since the number of tracks to be fit is determined by the number of roads to be processed and the combinatorics of 1 hit per

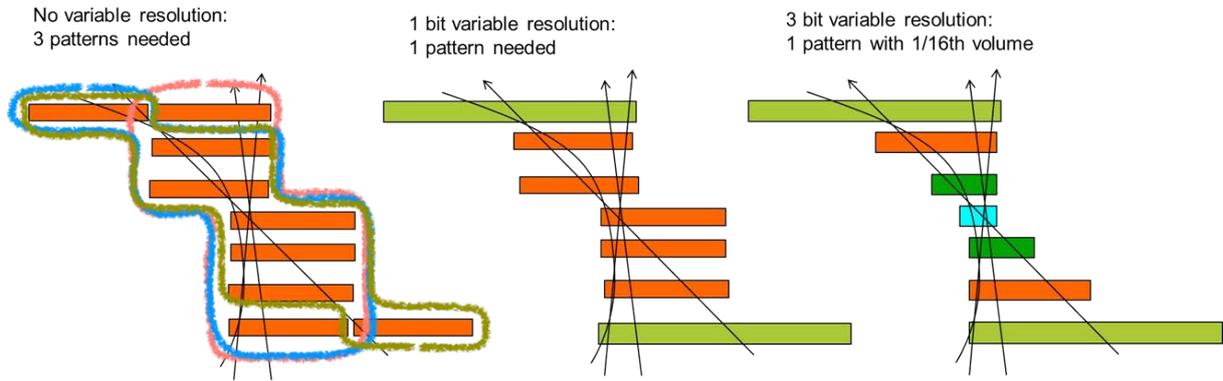


Figure 44: The sketches from left to right show the ability of variable resolution to reduce the number of Associative Memory patterns and the pattern volume available to random hits. The fuzzy lines in the left sketch show the 3 patterns needed to accept the tracks. The color of a Super-Strip indicates its area.

layer in a road, the fitting workload depends on the average ratio of the number of matching roads to the number of actual tracks in the event,  $\langle \text{roads}/\text{track} \rangle$ , as well as on the average number of hit combinations per road,  $\langle \text{combinations}/\text{road} \rangle$ .

For this reason the quality of the pattern bank is characterized by both its coverage and by the rate of fake roads it generates. The coverage is defined as the fraction of tracks that match at least one pattern in the bank. It describes only the geometric efficiency of the bank. Track efficiency on the other hand includes the effect of all the algorithms used in FTK, the clustering of single detector channels before the AM and track fitting whose efficiency is determined by a  $\chi^2$  cut. Given a certain pattern size, the coverage of the bank can be increased by adding patterns to the bank. Unfortunately, however, the number of fake roads is roughly proportional to the number of patterns in the bank. Moreover, as the luminosity increases, the fake rate increases rapidly because of the increased detector occupancy. To counter that, we must reduce the width of our patterns. If we decreased the pattern width using the standard AM technology, the number of patterns would increase sharply, making the overall system very large and extremely expensive.

We have proposed an elegant solution to this problem [26]: variable resolution patterns. Each pattern and each detector layer within a pattern can have an optimal width that is implemented using a “Don’t-Care” (DC) feature (inspired by ternary content-addressable memories or CAMs) to increase the width of a pattern for any detector layer when that is more appropriate. In other words we use patterns of variable shape. As a result we can reduce the number of fake roads while keeping the efficiency high and avoiding the excessive bank size that would occur from an overall reduction in pattern width (see Fig. 44).

Optimized pattern shapes are quite different for high coverage and low coverage patterns. Because many different particle trajectories pass through a high coverage pattern, reducing its width would produce many narrower patterns and not significantly reduce the fake rate. Consequently these are retained as wider patterns. On the other hand, low coverage patterns, which are needed for an overall tracking efficiency above 90%, are consistent with many fewer trajectories. They typically are made narrower in one or more detector layers, thus reducing the fake rate.

The choice of wider or narrower road width is made layer by layer using the simulation. For example, when we split the road in a layer into two parts, in some cases both halves are traversed by real tracks while in other cases just one of them contains real particle trajectories. For the former, the layer is used with the DC feature enabled to widen the road by a factor of 2 by ignoring the least significant bit of the pattern word during the comparison with silicon hits. For the latter, the layer is used at full resolution, reducing by a factor of two the area in that layer that the road offers to uncorrelated hits from other tracks.

Option	DC bits	Road Width	# of patterns	Efficiency (%)	<roads>	<fits>
1	$[(1, 0)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 36)_{\text{pix}}, 32_{\text{sct}}]$	34M	90	3.8k	23k
2.1	$[(1, 1)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 32_{\text{sct}}]$	21M	95	3.9k	33k
2.2	$[(1, 1)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 32_{\text{sct}}]$	18M	94	3.4k	28k
2.3	$[(1, 1)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 32_{\text{sct}}]$	16.8M	93	3.2k	26k
3	$[(1, 2)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 144)_{\text{pix}}, 32_{\text{sct}}]$	8M	95	4k	60k
4	$[(1, 1)_{\text{pix}}, 2_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 64_{\text{sct}}]$	8M	96	6k	98k

Table 1: Results of the Don't-Care study in the barrel. The numbers of patterns loaded and the roads and fits per event are per  $\eta - \phi$  tower. The efficiencies quoted are for events containing a single muon.

Option	DC bits	Road Width	# of patterns	Efficiency (%)	<roads>	<fits>
1	$[(1, 0)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 36)_{\text{pix}}, 32_{\text{sct}}]$	34M	87	5.8k	33k
2.1	$[(1, 1)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 32_{\text{sct}}]$	18M	91	7.1k	56k
2.2	$[(1, 1)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 32_{\text{sct}}]$	16.8M	91	6.9k	55k
2.3	$[(1, 1)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 32_{\text{sct}}]$	15M	91	6.2k	50k
3	$[(1, 2)_{\text{pix}}, 1_{\text{sct}}]$	$[(30, 144)_{\text{pix}}, 32_{\text{sct}}]$	8M	92	5k	90k
4	$[(1, 1)_{\text{pix}}, 2_{\text{sct}}]$	$[(30, 72)_{\text{pix}}, 64_{\text{sct}}]$	8M	93	9k	154k

Table 2: Results of the Don't-Care study in the endcap. The numbers of patterns loaded and the roads and fits per event are per  $\eta - \phi$  tower. The efficiencies quoted are for events containing a single muon.

Variable resolution was implemented in the simulation and used to optimize the Super-Strip sizes. A  $WH$  sample with a mean of 69 pileup events was chosen as typical of the high occupancy multijet events that come out of the level-1 trigger. The pattern banks described in section 5 were used. The narrowest road width has SS sizes that are 16 strips in the  $r - \phi$  SCT layers and 15 (36) pixels in the  $\phi(z)$  direction in the pixel layers. This is indicated with the road width notation  $[(15, 36)_{\text{pix}}, 16_{\text{sct}}]$ . We studied the impact on the number of matched roads and candidate tracks to be fit for different road widths, *i.e.* the number of Don't-Care bits used for each detector layer. For example, if the maximum SS width in some layer is increased by a factor two, we say that a single DC bit (1) has been activated in that layer. A bank with  $[(0, 0)_{\text{pix}}, 1_{\text{sct}}]$  DC bits uses the narrowest road width in the pixel layers for all patterns but allows one DC bit to be set on the SCT layers, increasing the width by a factor of 2, from 16 to 32 strips, when the simulation finds that it is best for that pattern. The AMchip allows up to 6 DC bits in a layer. For example, a bank with Don't-Care bits  $[(3, 2)_{\text{pix}}, 1_{\text{sct}}]$  allows a large width increase in the pixel layers: in this case the SS size can be as large as 120 pixels in the  $\phi$  direction and 144 pixels in the  $z$  direction (again we decide on the optimal number of DC bits on a pattern-by-pattern basis based on simulation). Although at present the DC configuration is the same for all pixel layers and the same for all SCT layers, this is not required. We may find with future study that it is better, for example, to have different DC settings for inner and outer SCT layers.

Tables 1 and 2 show the results from the pile-up = 69  $WH$  sample for selected configurations. The 8 layers used for pattern recognition are the three external pixel layers (IBL is not yet in the simulation), the four axial SCT layers, and the third stereo SCT layer from the inside. Table 1 is for the barrel and Table 2 is for the endcap. The numbers of patterns loaded, roads and fits are per  $\eta - \phi$  tower.

The options are ordered by decreasing number of patterns. The maximum allowed is 16.8 M, the number of memory locations on the two Associative Memory boards in a tower (see section 4.6). Thus option 1 cannot be used. For the option 2 configurations, we reduced the bank size and efficiency to satisfy this constraint. There are two other hardware constraints that must be satisfied (see section 4.5): the average number of roads (<roads>) and fits (<fits>) per tower per event should not exceed 16 k roads and 80 k fits respectively. For this reason we have chosen option 2.2 for the end-cap and 2.3 for the

	MC11	MC12	run 215091 (Dec. 2012)
# of bad modules in the 1 <sup>st</sup> stage	100	104	137
# of bad modules in the 2 <sup>nd</sup> stage	129	129	165

Table 3: The number of bad silicon modules in the layers used in 1<sup>st</sup>-stage and 2<sup>nd</sup> stage FTK tracking.

barrel as the default FTK configuration for our performance studies. However we note that for a given DC configuration, reducing the number of patterns (going from 2.1 to 2.2 to 2.3) reduces the number of roads and fits while minimally reducing the efficiency. Thus option 3 might also work, since with a little tuning the number of fits in the endcap could fall into the acceptable range while requiring an AM bank less than half the size of option 2. Although future studies of more DC options might result in a better choice, for now we work with option 2.2/2.3.

The numbers above were obtained with the standard requirement of hits on at least 7 of the 8 layers used in stage 1. However to minimize the loss in efficiency in the transition region between the barrel and endcaps, we reduce the requirement in that region only. This increases the numbers of roads and fits in that region, but the numbers still satisfy the hardware limits.

We have also studied the data flow for pile-up = 46, which is likely to occur early after LS1. We find that we can cover the barrel region with 16 Processor Units, each one covering two  $\eta - \phi$  towers.

**Impact of using the Wild Card option:** As described in section 4.5.7, the wild-card option is used to reduce the impact of dead silicon modules on tracking efficiency by considering all Super-Strips in such modules to contain a hit during the pattern recognition stage. This reduces the tracking constraints in those regions and thus increases the number of fake roads and the number of fits to be done. The study above was repeated using the selected DC configuration but now with the use of wild cards. Our MC11 simulation has the number of bad modules not very different than those in MC12 or in a run near the end of 2012 data taking (see Table 3).

Figures 45(a) and 45(b) show the relative increase in matched roads and fits compared to not using the wild card option. Averaged over all towers, the number of roads goes up by a factor of 1.3 and the number of fits increases by a factor of 1.2. However, since dead modules and thus the wild card use will not be uniform over the detector, we have plotted the increase as a function of the number of broken modules per tower. In the plots the worse case increase is 2 and 1.7 for roads and fits, respectively. Even for the higher dead module multiplicities, the rates remain within our hardware capability.

However it is clear that there are limits to the use of the wild card, limits that could come from excessive data flow. In that case efficiency and data flow would have to be balanced. If a detector layer has too many broken modules in a tower, that layer can be replaced by a different layer in the pattern bank for that tower.

#### 4.1.3 Internal monitoring and diagnostics: the Spy Buffer system

FTK processes a large quantity of data, little of which winds up in the event record. If an error occurs, properly diagnosing its source requires access to the data at every step in the FTK pipeline. To accomplish this, we implement the Spy Buffer system, which consists of Spy Buffers in the input and output of each board and between major functions on the board. A Spy Buffer is a circular memory and a register that contains its status. This memory is continuously written with the data being processed by the board.

The write operation is stopped when a Freeze signal is asserted to preserve the data already written. The Freeze signal has 3 possible sources. (1) When an error is detected on a board, Freeze is asserted to all Spy Buffers on that board. (2) When an error is detected on a board, Freeze is sent to the board(s) immediately upstream of it to freeze their output Spy Buffers. (3) There is a bit in the Event Trailer

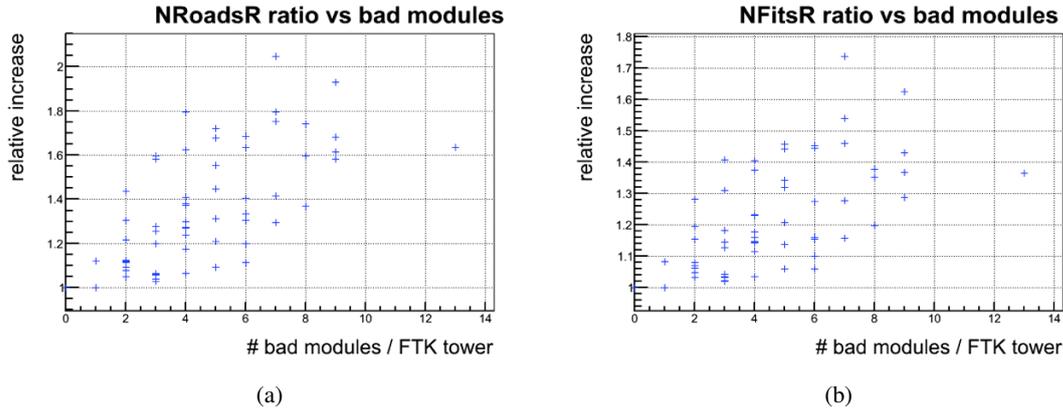


Figure 45: The relative increase in the number of (a) roads and (b) fits with the wild card enabled. The scatter is partially due to the location of the dead module in the tower.

record that tells all FTK boards to freeze their Spy Buffers after processing the current event. This last option enables events without error flags set to be read out and compared with simulation to ensure that there aren't subtle problems in the hardware. After Freeze is set, no data can be written into the memory and the content of the memory is read through VME access.

For each Spy Buffer there is a Status Register that contains a pointer to the first free memory location, an overflow bit that indicates if the memory has been written more than once, and the Freeze bit.

Spy Buffers are small since we want to use them to monitor or analyze a single event. Each Spy buffer will contain 4-8 average events. Since the maximum average number of words per event that can be transmitted on a FTK link is 1000, each single Spy Buffer will be 4-8 k locations deep.

## 4.2 Dual-output HOLA

The SCT and pixel RODs send the silicon hit data to the DAQ ROSs after each level-1 accept. HOLA output mezzanine cards on the RODs convert the parallel data to serial and then transmit it over optical fibers using the S-LINK protocol. In order to provide FTK with the data, we designed a HOLA with two identical output streams. The functional diagram of the dual-output HOLA is shown in Fig. 46.

The 32-bit silicon hit words enter the HOLA at up to 40 MHz. The data are read from the FIFO at 50 MHz and converted to 16-bit words at 100 MHz. The external serializer-deserializer (SERDES) in the original HOLA is now replaced with a pair of SERDES megafunctions within the more modern Altera FPGA. Since they receive data from a common FIFO, the data being sent out are identical. The SERDES outputs feed a pair of optical transceivers. Moving the SERDES onto the FPGA reduces power and cost while retaining 90% of the original HOLA firmware.

The handling of the return control data, shown as pink arrows in the figure, is unique to the dual-output HOLA. The channel that feeds the DAQ ROS implements the full S-LINK protocol including link-down, link-reset, and general return lines. The FTK channel however only implements the XON/XOFF part of the S-LINK protocol to stop and start data flow. This is important to enable FTK modules to be installed and commissioned during a run without interfering with ATLAS data taking. (The  $\eta$ - $\phi$  coverage and computing power of FTK will be increased during 2015-2017.) Since we want XON/XOFF flow control functionality during normal FTK operation, but not during installation and initial commissioning, an FTK\_XOFF\_enable register implemented on the HOLA is set to 0 on HOLA power up and not set to 1 until a specific control sequence is sent by the FTK\_IM mezzanine card on the Data Formatter. FTK\_XOFF\_enable can also be set back to 0 if desired and it is automatically set to 0 if the link to FTK goes down. This scheme has been exercised extensively in a test stand and in the ATLAS DAQ at Point

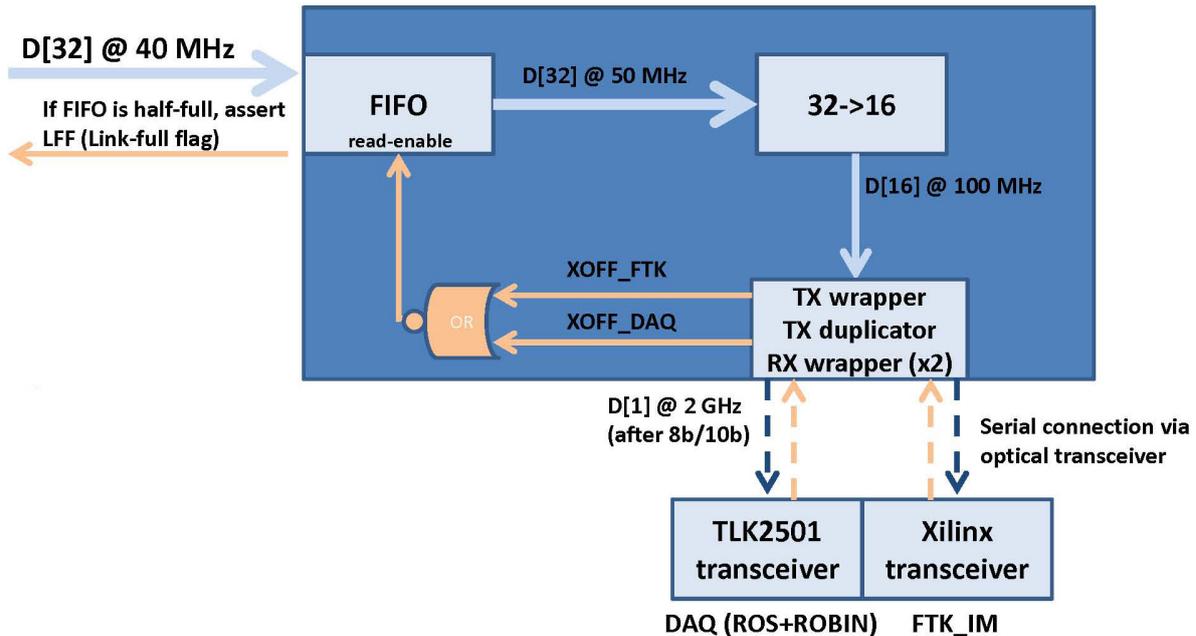


Figure 46: Functional sketch of the dual-output HOLA. The blue arrows show the flow of the silicon hit data, while the pink arrows show the return control data.

1 and found to work without error (see section 7).

The handshake required to start S-LINK data transfer occurs between the HOLA and the DAQ ROS. S-LINK comes up even if the fiber to FTK is unplugged. On the FTK side, we use a modified S-LINK receiver that does not require an initial handshake. Since the two fibers from the HOLA transmit identical data, once a fiber is connected to an FTK board, that card sees the silicon data almost immediately. The receiving FTK board is responsible for informing the DAQ system that a ROD-to-FTK link that is supposed to be operating is in fact down. The Data Formatter does this by setting an error bit in the End Event word. When the link is re-established, a global INIT is sent to all FTK boards.

A prototype dual-output HOLA was produced early in 2011 and it went through a series of thorough tests including a bit-error-rate test to more than  $1 \times 10^{15}$  bits without error. The XON/XOFF scheme was also fully tested. Following a CERN production readiness review, we produced and tested all of the HOLAs that were to be used with the full FTK system since the HOLAs had to be installed during LS1. As a result of the recent decision to increase the number of RODs in the silicon system for data taking starting in 2015, we will produce and test a small number of additional HOLAs prior to the summer installation date. A photograph of the dual-output HOLA is shown in Fig. 47.

### 4.3 FTK Input Mezzanine (FTK\_IM)

The functions of the FTK\_IM are to receive the pixel and SCT data from the Inner Detector RODs, to perform clustering, and then to forward data to the Data Formatter main board. Clustering will be performed on both pixel and SCT data with the dual purpose of reducing the amount of data to be processed by downstream FTK functions and determining the cluster center to improve spatial resolution. Full clustering of pixel data is performed, and the partial clustering done in the SCT and IBL detectors upstream of FTK is completed by the FTK\_IM.

Each FTK\_IM will receive up to 4 S-LINK fibers from Inner Detector RODs. On board there are two FPGAs each receiving data from two S-LINK channels. The four data streams received by the FTK\_IM

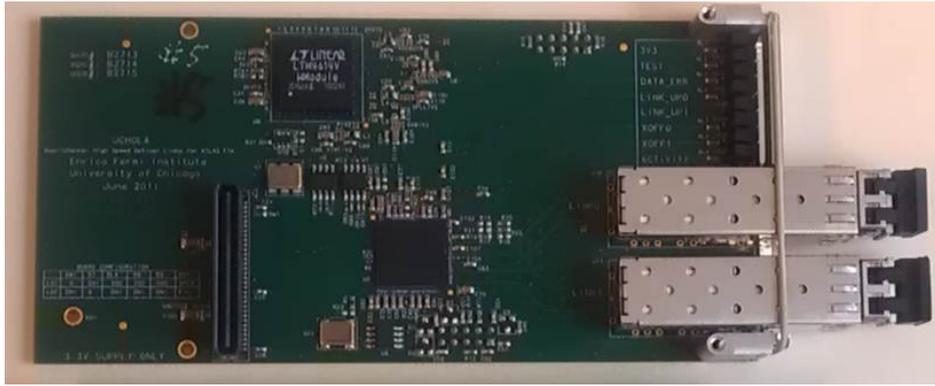


Figure 47: A dual-output HOLA.

will be processed independently and sent over independent channels to the Data Formatter. In this way event synchronization will only be performed on the Data Formatter board. In order to make better use of resources on the FTK\_IM FPGAs, each FPGA will receive one pixel S-LINK and one SCT S-LINK. This means that if the full 128 FTK\_IMs are installed the Data Formatter + FTK\_IM system will be able to received up to 256 pixel S-LINKs and 256 SCT S-LINKs.

#### 4.3.1 Hardware description



Figure 48: FTK\_IM prototype.

The FTK\_IM functions are implemented in a mezzanine card with a size of 149mm x 74mm that will connect to the Data Formatter main board with a High Pin Count (HPC) FMC connector. Figure 48 shows the FTK\_IM prototype. The four SFP+ connectors on the left side of the card receive the 4 S-Link channels and are directly connected to the two Spartan6 FPGAs (XC6SLX150T-3FGG484C) on the right side of the FTK\_IM. Each FPGA receives two links, processes data independently, and transmits the output to the FMC connector. Each FPGA is also equipped with a 18Mb external SRAM and a 32Mb flash memory. These memories are not used in the current firmware implementations. The SRAM can be used to implement a large look-up table if a more sophisticated clustering algorithm will need it. The FTK\_IM is powered from the FMC connector, but has also an external power connector for tests without the Data Formatter motherboard. The JTAG chain for FPGA configuration is accessible both from the FMC connector and an external connector. The chosen SFP optical transceiver supports a line speed of up to 2.0 Gbps. The Spartan6 FPGA is able to run the S-LINK up to 3.1 Gbps. This makes it possible to

increase the S-LINK speed by replacing the SFP and adjusting the firmware.

### 4.3.2 Clustering algorithms

The SCT clustering algorithm decodes strip hits already associated with a cluster by the ABC130 front-end chip and extends the cluster to consecutive strips that are split at the ABC130 boundaries.

The Pixel clustering algorithm is more complicated due to its 2D nature. In order to combine speed with quality, we split the algorithm into two steps. The first step groups together hits belonging to the same cluster.<sup>6</sup> The second step analyzes the hits in the cluster in order to calculate the cluster properties.

The baseline algorithm for the first step is described in [27]. It has a complexity that scales linearly with the number of hits in the module. This is important because it means that the rate at which hits are processed is practically independent of instantaneous luminosity. Hits are loaded into a matrix that replicates a small fraction of a pixel module (164x8 pixels) and clusters are formed using local logic. With this approach the processing time in clock cycles is  $2 \times N_{\text{hits}} + 2 \times N_{\text{clusters}}$  which corresponds to approximately 3 clock cycles per hit independent of the occupancy. With the FPGA we have chosen, the largest in the Spartan6 family, logic resource usage is approximately 60%, making it a challenging firmware project.

We are working on an improved version of the algorithm that has some dependence on the occupancy but uses a much smaller region of the pixel module (a grid between 7x6 and 13x10 pixels). The logic resource usage scales with the area of the grid and the speed scales with the square root of the area of the grid for this algorithm. Since more cores can be instantiated in the FPGA, we expect a gain of at least a factor of 10 in overall speed which will allow us to loop over hits multiple times in busy detector regions.

For each hit received, we consider neighboring hits with a maximum distance of 4 pixels along the  $\phi$  direction and 5 pixels along the orthogonal direction<sup>7</sup> in the 7x6 grid. The maximum distance can be extended to 7 pixels along the  $\phi$  direction and 9 pixels along the orthogonal direction in a 13x10 grid. After loading neighboring hits in the grid, hits forming a cluster with the first hit received are identified and sent to the second algorithm step. Hits within a small distance of the first hit received but not part of the same cluster will be processed again until the cluster they belong to is identified. This algorithm reduces the logic needed by clustering in a grid not larger 7x6 pixels (or up to 13x10).<sup>8</sup> This means that clusters exceeding 4 pixels along  $\phi$  or 5 pixels along the orthogonal direction will be truncated. This is a good compromise that reduces the usage of resources on the FPGA and is compatible with the chosen Spartan6 device. The implementation of this algorithm is in progress.

The second step of the algorithm determines the center of the cluster assuming equal weight for all hit pixels. An improved cluster centroid determination based on Time-over-Threshold (ToT) information is being evaluated. The FPGA resources will not be a limitation for simple algorithms.

### 4.3.3 Hardware status

A first version of the FTK.IM was produced with a connector compatible with the EDRO board [28, 29] to which it was connected in the FTK vertical slice test (see section 7). The first prototype FTK.IM with the FMC connector has been assembled and is currently under test. Firmware that implements the SCT clustering exists. For the pixels, we have simulated the algorithm that will be used in the firmware.

---

<sup>6</sup>We define a cluster as a set of hits that are contiguous either through a common side or a common corner.

<sup>7</sup>The  $z$  direction for barrel modules, and the  $r$  direction for modules in the disks

<sup>8</sup>The grid is 7 pixels wide along the  $\phi$  direction to allow clusters extending to either  $\phi$  side. Hits will be presorted in the orthogonal direction so the grid extends only to one side in the orthogonal direction.

## 4.4 Data Formatter

### 4.4.1 Introduction

As noted previously, FTK is organized as a set of parallel processor units within an array of 64  $\eta$ - $\phi$  towers. The  $\phi$  divisions are shown in Figure 49 and the  $\eta$  divisions are shown in Fig. 4. To avoid inefficiency at tower boundaries, the towers must overlap because of the finite size of the beam's luminous region in  $z$  and the finite curvature of charged particles in the magnetic field. The Data Formatter system receives the hits from the Pixel and SCT readout drivers (RODs) through input clustering mezzanine cards (FTK\_IM), remaps the ATLAS Inner Detector geometry to match the FTK  $\eta$ - $\phi$  tower structure, performs data sharing in overlap regions, and delivers the hits to the Processor Units.

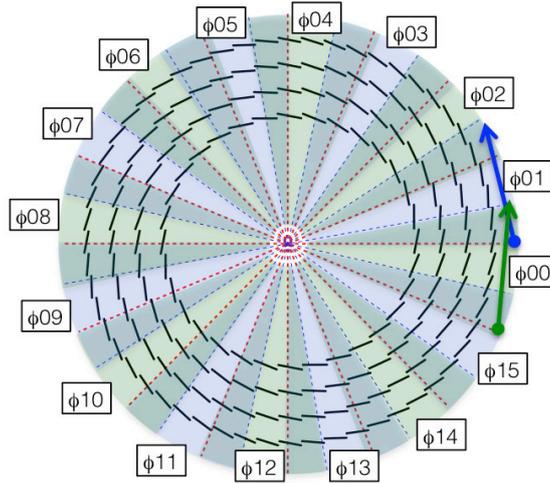


Figure 49: 16 FTK  $\phi$  sectors. The sectors overlap to avoid inefficiency due to finite curvature of low momentum tracks. As an example the coverage of the  $\phi00$  sector and  $\phi01$  sector is shown with green and blue arrows respectively in the figure.

The input S-LINK pixel and SCT fibers are assigned to FTK  $\eta$ - $\phi$  towers so as to minimize the data sharing. Figure 50 illustrates data sharing that is needed among the FTK  $\eta$ - $\phi$  towers in the  $64 \times 64$  matrix after optimizing the input link assignments. Off diagonal elements represent the data sharing between different towers. The red boxes show how the system is divided into 4 crates to minimize inter-crate data transfer. Figure 51 shows an example of the needed interconnections among the FTK  $\eta$ - $\phi$  towers for the assignment shown in Fig. 50 in which each crate contains the 16  $\eta$ - $\phi$  towers within four  $\phi$  sectors.

### 4.4.2 The Data Formatter System

The Data Formatter hardware design is dominated by the input and output requirements, but it also maintains the flexibility needed to accommodate future expansion and allow changes in both the number of input links and the Inner Detector module-to-fiber assignments. Based on the design requirements, a system based on a full-mesh Advanced Telecom Computing Architecture (ATCA) backplane interconnect is found to be a natural solution for the DF design. Each of the boards is connected to each other with multiple point-to-point links in the full-mesh ATCA backplane. The key design features of the Data Formatter system are:

- One Field Programmable Gate Array (FPGA) is assigned to each FTK  $\eta$ - $\phi$  tower;
- Each DF board contains two FPGAs;

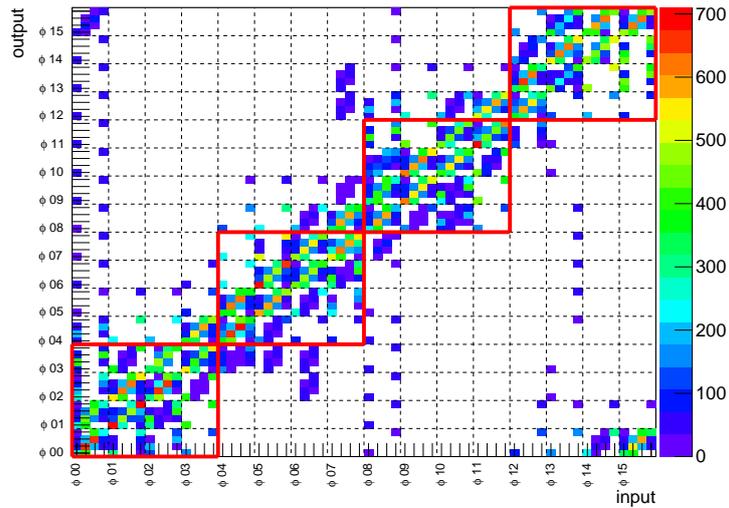


Figure 50: This  $64 \times 64$  matrix shows the data sharing among the FTK  $\eta$ - $\phi$  towers in the Data Formatter system. The red boxes indicate the assignment of towers to 4 DF crates to minimize inter-crate data sharing. The color scale indicates the number of clusters shared between towers per event for 8 TeV data with 50 ns bunch separation and  $\langle \mu \rangle = 30$ .

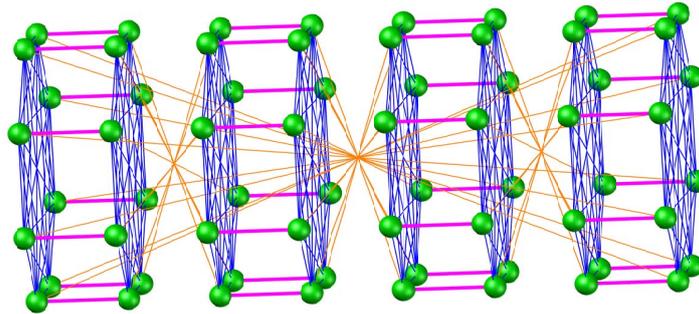


Figure 51: The interconnections for data sharing among the 64  $\eta$ - $\phi$  towers within 4 DF crates. A green ball represents one FTK tower. Blue lines represent data sharing between the upper FPGAs or lower FPGAs in a shelf. The red lines represent data sharing between an upper and a lower FPGA, and the yellow lines represent inter-shelf data sharing.

- 8 boards (16 FTK  $\eta$ - $\phi$  towers) are assigned to each ATCA shelf (see Figure 51);
- The system consists of 64 FPGAs, 32 Data Formatter boards, and 4 ATCA shelves.

**Design Concept** Figure 52 shows the design concept of the Data Formatter board. Fiber links from RODs are received on input FTK.IM mezzanine cards which perform hit clustering. Cluster centroids from the mezzanine card are fed into one FPGA on the Data Formatter board (yellow lines). The two FPGAs share data with each other (orange line), with other boards on the same shelf (pink line), and with other shelves (green line). Finally trigger tower data are sent downstream to the FTK core crates (blue line). The Rear Transition Module (RTM) is used to send the data downstream as well as to perform inter-crate data sharing.

The Data Formatter board is equipped with three types of internal data-sharing paths. The first is

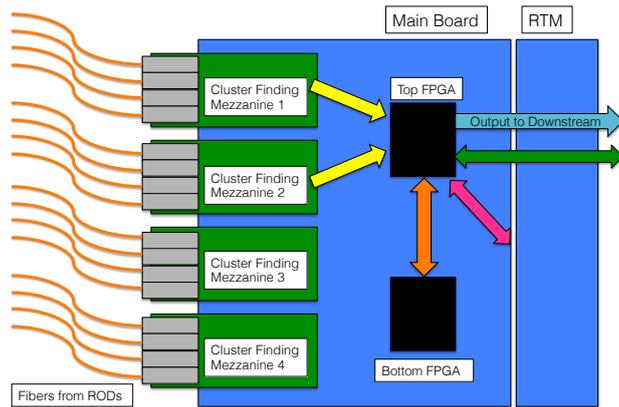


Figure 52: Data Formatter design concept. Each DF board supports up to four FTK.IM mezzanine cards and two large FPGAs. The RTM board contains fiber optic transceivers for sending data downstream to the FTK core crates.

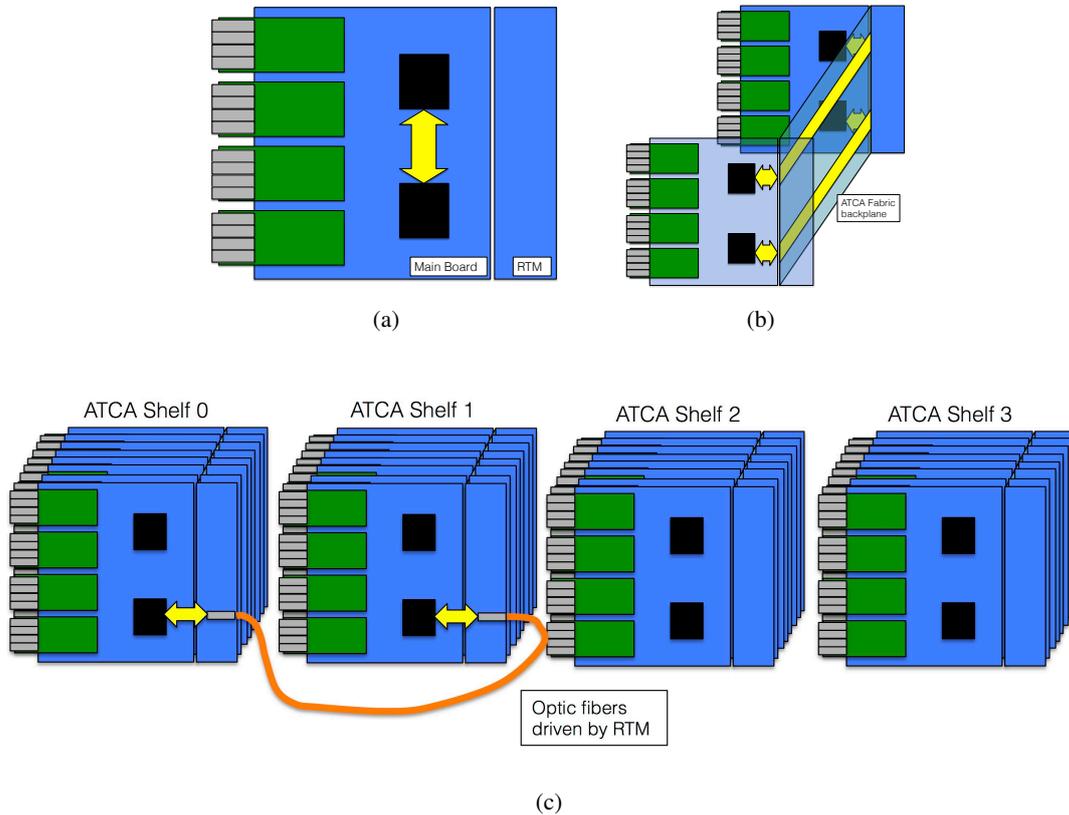


Figure 53: Three types of data sharing in the Data Formatter. (a) A local bus connects the two FPGAs on the board. (b) All FPGAs in the shelf are directly connected using the ATCA backplane fabric interface. (c) Inter-shelf data sharing uses RTM transceivers. These links are driven by serializers and deserializers implemented in the FPGAs.

a local bus that connects the FPGAs on the same Data Formatter board (Fig. 53(a)). The second is the fabric interface in the full-mesh ATCA backplane. All top FPGAs in the shelf are directly connected over high speed serial links in the backplane, as are all bottom FPGAs (Fig. 53(b)). The third path is for inter-shelf data sharing using fiber optic transceivers on the RTM (Fig. 53(c)). All FPGAs in the system are interconnected with the three types of data sharing links. The routing table is implemented in firmware and therefore is user defined and flexible.

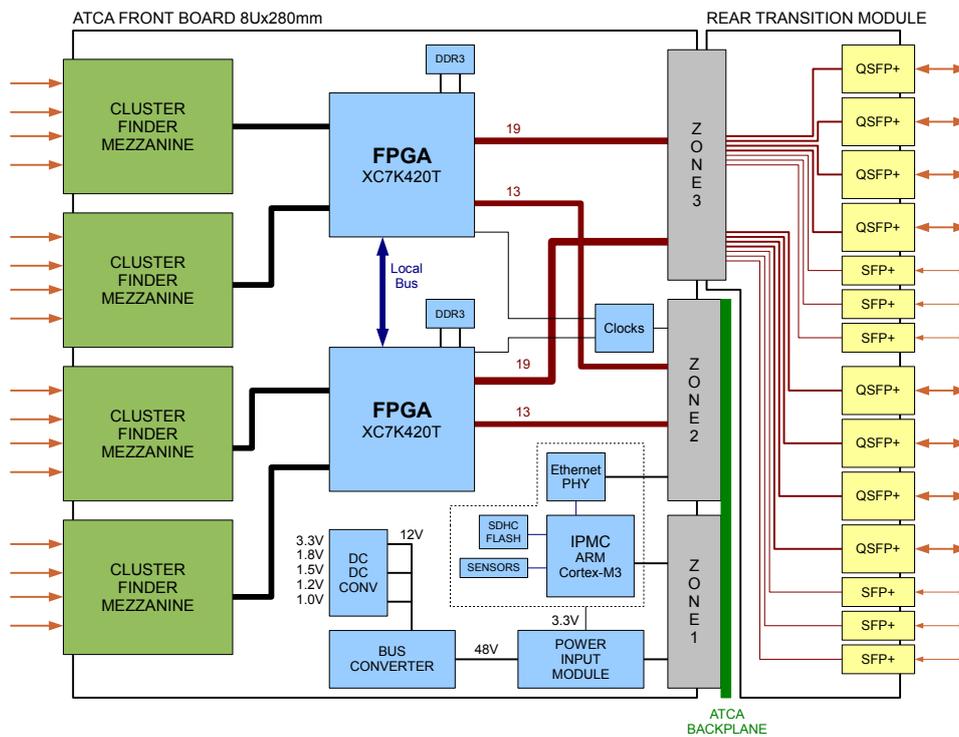


Figure 54: The Data Formatter board block diagram with the rear transition module.

**Board-Level Design Implementation** The block diagram of the Data Formatter board, nicknamed Pulsar IIa, is shown in Fig. 54. The FPGAs are the DF's processing engines and are interfaced to the input mezzanine cards (through the FPGA Mezzanine Card (FMC) connectors), the ATCA full mesh backplane (through the Zone-2 connector), and the RTM fiber transceivers (through the Zone-3 connector). The FPGA high speed SERDES transceivers (GTX) connect to the ATCA backplane and RTM fiber transceivers (dark red lines). The two FPGAs on a DF also share data with each other over a high speed low voltage differential pair (LVDS) local bus (blue line).

Additional circuitry on the Data Formatter board is used for power distribution, slow control, and clock distribution. Each DF board has a microcontroller interfaced to the Zone-1 connector. It is used for ATCA Shelf management (through the IPMB bus) and slow controls (through 100BASE-T Ethernet). The prototype board is shown in Fig. 55.

**A Scalable and Flexible Architecture** The Data Formatter system described above supports up to 128 input mezzanine cards. Each mezzanine card supports up to four SFP+ optical transceivers for a total of 512 input links. This is significantly more than the total number of Pixel and SCT readout links used

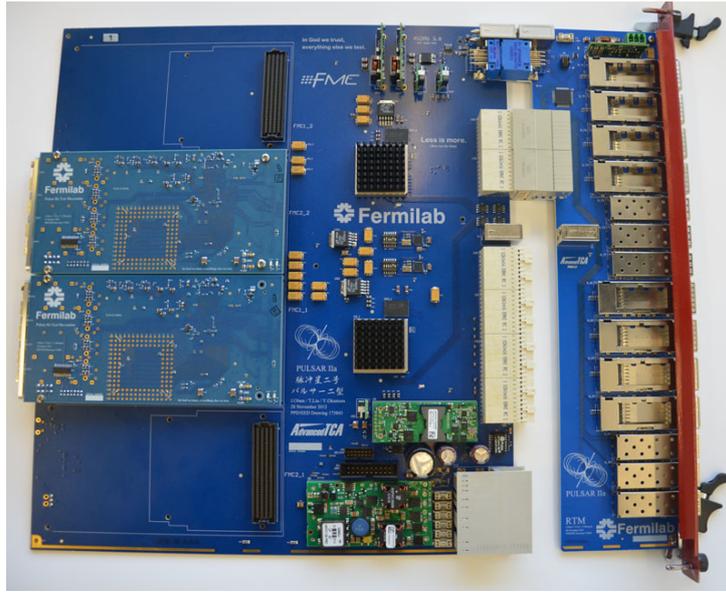


Figure 55: Prototype DF with its RTM and two test mezzanine cards.

in the 2012 run (222 links). The number of links after LS1 will be 214 (128) from the pixels (SCT), including IBL. Note that 8 slots are used per ATCA shelf while the Data Formatter is designed to be able to use all 14 available slots. The extra six are available for future expansion.

RTM boards support up to eight QSFP+ and six SFP+ optical transceivers; half of the transceivers are directly connected to the top FPGA and the other half are directly connected to the bottom FPGA. Optical or copper serial transceiver modules may be installed in any SFP+ or QSFP+ location on the RTM. The Data Formatter sends data to Processor Unit AUX cards for the first stage tracking and Second Stage Board for the second stage tracking. The SFP+ and QSFP+ transceivers may also be used for extra board-to-board data transfers (e.g. inter-shelf links).

#### 4.4.3 Bandwidth Requirement Estimation

Bandwidth requirements for the Data Formatter design were developed using ATLAS data. A software model of the data sharing was created so that data flow in the system could be simulated using actual data taken in 2012. These results were scaled and extrapolated to the target LHC operating conditions of  $\sqrt{s} = 14$  TeV,  $\langle \mu \rangle = 70.0$ , and 25 ns bunch spacing. The reduction in data volume due to hit clustering was also included in the estimation. Under these conditions the Data Formatter design was found to meet bandwidth requirements for all internal data links. Table 4 shows the estimated bandwidth for output links sending data downstream and internal links sharing data in the DF system. The capacity is significantly larger than needed. The link merging points within the DF also have capacity much greater than required for  $3 \times 10^{34}$  operation.

The full-mesh ATCA backplane makes possible many equivalent alternate paths among FPGAs, allowing further optimization of the routing algorithm and improvement in the bandwidth use in the system. The current data-driven estimate does not include the IBL, which will be added in the future after final decisions are made for the IBL module-to-fiber mapping.

	BW Required	Maximum Capacity
AUX	8.98	48
SSB	2.37	6
Fabric	2.62	6
Local Bus	7.52	24
Inter-Crate	5.58	12

Table 4: Summary of the expected bandwidth requirements (in Gb/s) and the preliminary system bandwidth capacity (in Gb/s) in the DF system. The maximum capacities shown are given for the less expensive of the two chips being considered.

## 4.5 Processor Unit AUX

Pattern recognition and first stage track fitting are done in the Processor Units (PU) in the core crates. Each PU consists of an Associative Memory VME card (AMB) with a large auxiliary board (AUX) behind it. The AMB is described in section 4.6.

### 4.5.1 AUX functionality

The AUX receives hits from the Data Formatters for the 8 silicon layers used in the first stage of track reconstruction. It stores the hits in the Data Organizer (DO), a smart database built on the fly that allows rapid retrieval of hits in a road, and sends the hits to the AMB with coarser resolution appropriate to pattern recognition (Super-Strip or SS). When the AMB finds a road with hits on at least 7 of the 8 layers, the road number is sent to the AUX which then retrieves all of the hits in the road. Rapid hit retrieval is possible because hits are stored in the Data Organizer by SS address, and a road consists of a single Super-Strip in each layer. The hits, the road number, and the sector number are transferred to the track fitter (TF).

The Track Fitter calculates the  $\chi^2$  for each combination of one hit per layer in the road (Nominal fit) using linear constants stored by sector number. The same constants are used when there are hits in only 7 of the 8 layers (Majority fit) by calculating the hit location in the missing layer that would minimize the overall  $\chi^2$ . Any combination with  $\chi^2 < \chi^2_{\text{cut}}$  is sent out of the TF. If a combination with hits on all 8 layers has a  $\chi^2$  above  $\chi^2_{\text{cut}}$  but below  $\chi^2_{\text{high}}$ , that track candidate is refit 8 times with one hit dropped each time in order to allow for detector inefficiency and picking up a random hit (Recovery fit). If a refit satisfies the goodness of fit requirement, it is accepted.

The tracks forwarded from the Track Fitter go to the Hit Warrior (HW) function for duplicate track removal. If two tracks in the same road share more than a programmable number of hits, only the higher quality track is kept. The quality is defined based on the  $\chi^2$  and how many pixel and SCT layers have a hit. Tracks exiting the HW are forwarded to the Second Stage Board where hits on the other 4 detector layers are added.

A functional diagram of the AUX is shown in Fig. 56. The hits from the Data Formatters enter through two QSFP+ connectors with one silicon layer per fiber. Within the Input FPGAs, the superstrip number for each hit is generated and sent through the VME P3 connector to the Associative Memory Board. The Super-Strip number and the full-resolution hit coordinates are sent to the 4 Processor FPGAs, each of which contains a Data Organizer and Track Fitter to handle a quarter of the Associative Memory Board (one LAMB mezzanine card). Pattern addresses for roads containing at least 7 hit layers are sent back from the AMB through the P3 connector. They are received by the processor chips which extract the hits on the road and carry out the fits. The tracks that pass a  $\chi^2$  cut are sent to one of the input FPGAs where the duplicate track removal is done, and the remaining tracks are sent to the Second Stage Board through an SFP+ fiber connector. The top-side assembly drawing of the AUX is shown in Fig. 57.

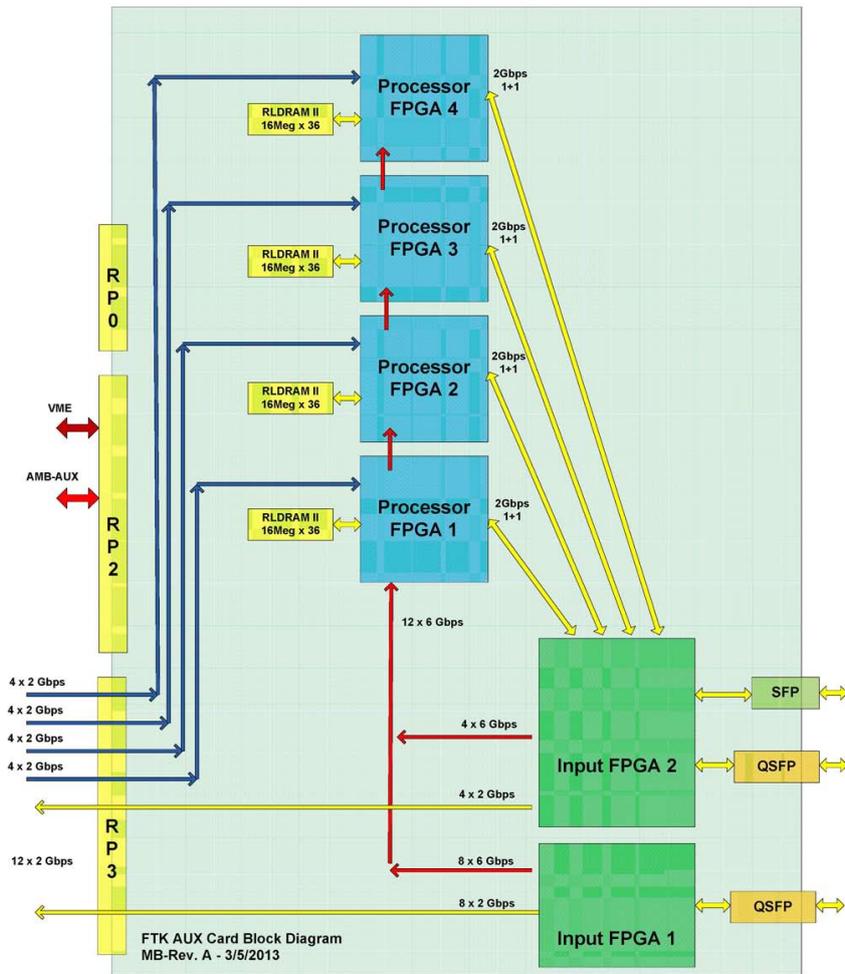


Figure 56: Functional diagram of the AUX card showing the data flow on the board.

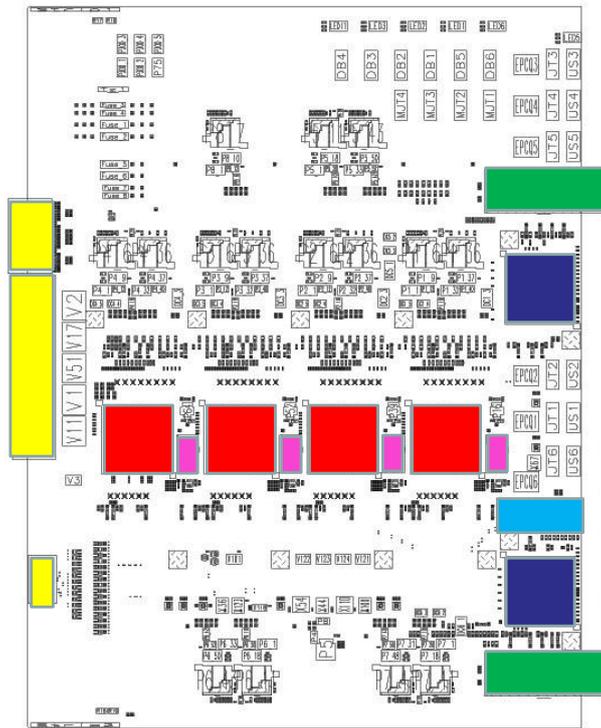


Figure 57: Top-side assembly drawing of the AUX card. The four Processor FPGAs are in red and the Input FPGAs are in dark blue. The VME connectors are in yellow, the QSFP+ connectors that receive the hits from the Data Formatters are in green, and the SFP+ connector that sends data to the Second Stage Board is in light blue. The external memory used by each Processor FPGA is in light purple.

Layer	barrel	endcap
IBL	-	-
Pixel 0	440	440
Pixel 1	380	360
Pixel 2	290	330
SCT 0	210	300
SCT 1	210	280
SCT 2	210	280
SCT 3	210	270
SCT 4	200	250

Table 5: Mean number of hits sent to an AUX card in a barrel or endcap tower for  $WH$  events with a mean of 69 pileup events.

#### 4.5.2 Data rates

A sample of simulated  $WH$  events with an average pile-up of 69 was used to estimate the number of hits that will be sent from the Data Formatters to an AUX card. The numbers in table 5 include the overlap regions across towers. A QSFP+ fiber has a maximum data rate of 6 Gb/s. With 32 bits per word and a 100 kHz level-1 trigger rate, 1500 hits per event could be transferred for each silicon layer. This provides significant margin over the numbers in Table 5. The 2 Gb/s serial links carrying the Super-Strip for each hit from the input FPGAs to the Associative Memory Board have similar margin; the 16-bit words are carried on 2 links per pixel layer and 1 link per SCT layer.

The 69 pile-up  $WH$  sample was also used to estimate the number of roads that will be sent from the Associative Memory Board back to the AUX for track fitting (see Tables 2 and 1). The number of roads processed by an AUX is half of the numbers given in the tables since there are two Processor Units per  $\eta - \phi$  tower. There are 32 2-Gb/s serial links per tower ( $16/\text{PU} \times 2 \text{ PUs/tower}$ ) each capable of carrying 500 32-bit words per event. This provides adequate data flow margin. Future studies will optimize the rapidity boundary between the barrel and endcap regions and the use of variable resolution to better equalize the numbers of roads in the two regions.

The number of good tracks coming out of an AUX card for a mean pileup of 69 is approximately 50 per event. Ten words will be sent per track: the 8 hits, the road number, and a word containing the sector number and the Hit\_Map which indicates which layers had hits. At a 100 kHz level-1 trigger rate, this corresponds to 50M 32-bit words/s, which can be accommodated with a single 6 Gb/s SFP+ fiber from the AUX.

#### 4.5.3 Input FPGAs

For each silicon hit received from a Data Formatter, the Input FPGA looks up the Super-Strip address using the SSMAP memories. The SS address is needed both for pattern recognition in the Associative Memory Board and for the location in the Data Organizer where the hit will be stored. There is a separate SSMAP for each layer to take advantage of the parallel input of the layer-by-layer data from the Data Formatters. Because each module in a given detector is identical, the local hit position in the module is converted into a local SS ID. The tower SS address is obtained by adding the local SS ID to the SS offset for that module.

Input FPGA 1 handles the pixel layers and one SCT layer, while Input FPGA 2 handles the other SCT layers. Since the hit position is the address into the SSMAP, larger memory is needed for a pixel layer than an SCT layer since a pixel hit position is specified by two coordinates. As a result, the total memory needed for the SSMAP is 6 Mb in Input FPGA 1 and less than 1 Mb in Input FPGA 2, both

of which are much less than the 20 Mb internal memory available in the Altera Aria V FPGAs that are being used.

The Input FPGAs send the Super-Strip IDs to the AMB on 12 serial links, one for each SCT layer and 2 per pixel layer<sup>9</sup>. An additional 12 serial links will carry the hit data from the Input FPGAs to the Data Organizers in the Processor FPGAs, again one link per SCT layer and 2 per pixel layer. Now however, each link has to carry both the SS ID (16 bits) and the hit word (32 bits for a pixel hit, 16 bits for a SCT hit). Here again there is significant margin since these links are designed for 6 Gb/s transmission. To simplify the work of the DO for the pixel layers as described below, the data from the Input FPGA to the DO are split between the two links based on the SS ID.

The data from the Input FPGAs go only to the nearest Processor FPGA. The data is then rapidly sent to the other Processor FPGAs in a daisy chain manner using Altera's Reverse Serial Loop-back connection in the FPGA transceivers.

Input FPGA 2 also implements the Hit Warrior function as described below.

#### 4.5.4 Data Organizer

The Data Organizer stores the detector hits in a smart database that allows rapid extraction of all hits within a road. The database is filled with the hits that are received from the Data Formatters at the beginning of each event. Hits are stored according to Super-Strip number so that when a road address is received from the Associative Memory Board, all hits in each detector layer associated with the Super-Strip in the road can be quickly transferred to the Track Fitter. The DO processes the data from the 8 silicon layers in parallel so that it can handle the expected average input rates of 1 hit per layer per 25 ns and 1 road per 10 ns, based on the numbers in Tables 2, 1, and 5. The DO firmware can process a hit every 5 ns and a road every 5-7 ns depending on the pattern bank configuration.

The DO has two states, WRITE and READ. The firmware is duplicated within each processor, with one DO in WRITE mode on event  $n$  and the other in READ mode on event  $(n-1)$ . In WRITE mode, hits received from the Data Formatters are written into the DO. In READ mode, hits are extracted for each road address sent by the Associative Memory Board. When WRITE mode is finished, the state is switched to READ mode for the same event. When READ mode is finished, the state is switched to WRITE mode for a new event. The READ and WRITE modes form a two step pipeline for each event.

**WRITE MODE:** Hits are received on the 12 serial links from the Input FPGAs, 1 link per SCT layer and 2 links per pixel layer. Each pixel hit consists of a 16-bit Super-Strip number and a 32-bit hit word. For SCT hits, the 16-bit SS number and the 16-bit hit information are packed into a single 32-bit word. The hits within a single SS are sent consecutively to the DO from the Data Formatter.

The DO uses three sets of on-chip memory for each detector layer: the Hit List Memory (HLM), the Hit List Pointer (HLP), and the Hit Count Memory (HCM). The HLM sequentially stores each hit received from the DF. The HLP stores the HLM address of the first hit stored for each SS. The HCM stores the number of hits in each SS.

A potentially serious problem is how to reset all of the HCM locations at the beginning of a new event to ensure that stale hits from a previous event won't be used. There are so many SSs that it is time prohibitive to physically reset all of these memory locations. However the problem is avoided because the Data Formatter sends all of the hits in a Super-Strip to the AUX consecutively. When the DO sees a new SS address, it knows this is the first hit in that SS and resets that one HCM location. This scheme does not reset the HCM for those SSs without a hit in the current event. That is not a problem since no road coming out of the AMB will request hits in those SSs.

---

<sup>9</sup>For pattern recognition using 3 pixel layers and 5 SCT layers, 11 of the serial links are used. All 12 are utilized for a 4 pixel, 4 SCT configuration.

The use of variable resolution patterns complicates this simple picture. The width of a superstrip in any layer can be different for different patterns. Some patterns will use the narrowest Super-Strip, others will use 2, 4, or 8 times that width. To solve this, the HCM is divided into 8 memory banks each 2k deep. The depth is the maximum number of the widest Super-Strips (Widest-SS), and the 8 banks for a given address hold the hit counts for the 8 narrowest SSs within the Widest-SS. There are 2k 1-bit registers that specify whether a Widest-SS HCM has been reset during the current event (collectively called the Data Count Reset or DCR). Those 2k bits are all set to 0 at the beginning of an event. When a hit arrives from a Data Formatter, the DCR bit for that Widest-SS is checked. If it is 0, signifying that the Widest-SS has not been zeroed, the count in the 8 HCM banks for that Widest-SS address is zeroed and the DCR bit is set to 1. If the DCR bit had previously been set to 1, then nothing is done. The hit is then processed, inserting it into the HLM, incrementing the HCM, and storing the HLM address in the HLP if this is the first hit for that narrow SS.

The pixel layers provide an additional challenge since data for each layer is arriving on two serial links. These links can be handled in parallel by having the layer's HLM split into 2 independent memories, one storing hits whose Widest-SS address is even and the other storing hits whose Widest-SS address is odd. The HLP and HCM memories are similarly split.

For each pixel (SCT) layer, the HLM contains up to 1k 32-bit (16-bit) hits, with the depth being  $> 6\sigma$  above the mean number of hits at  $3 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ . With *WH* simulation we have verified that the hit distribution for a tower follows a Gaussian distribution. If an event would overflow the HLM, an error bit is set. The HLP is a memory of 10-bit words 16k deep, the maximum number of superstrips in a tower. The pixel (SCT) HCM consists of 8 banks of 2k words each 5 (3) bits wide. These memories are needed for each of the two Data Organizers in a Processor FPGA. When implemented in the chip, the total memory used is 7 Mb out of the 24 Mb available in the Altera Aria V chosen for the Processor FPGAs.

**READ MODE:** Matched roads are sent from the Associative Memory Board to each Data Organizer on 4 serial links. The 32-bit word contains the road address and an 8-bit Hit\_Map that tells the DO which layers in the road contain hits. In order to retrieve the hits from the HLM, the Super-Strip for each layer must be obtained. The AM\_Map provides this information. For each of the 2M road addresses covered by a Processor FPGA, the AM\_Map contains a wide 144-bit word. There are 16 bits for each of the 8 layers, which includes the 14-bit SS address and 2 bits for the width of the SS for that pattern (the variable resolution feature). In addition there is a 16-bit sector number that is needed by the Track Fitter. The total memory for the 2 Data Organizers in an FPGA is 576 Mb, which is external to the FPGA. The memory chips are 16M $\times$ 36 bits organized in 8 banks. The maximum speed is 533 MHz, while we run it at 400 MHz which gives a 1.25 ns read time per word. With a burst length of 4, the 144-bit wide word is read every 5 ns.

The Super-Strips for the 8 layers are then processed in parallel. The narrow SSs specified by the SS widths for that road are looped over, and for each the hits are extracted from the HLM starting at the address found in the HLP, with the number of hits specified in the HCM. The hits along with the road number, the Hit\_Map, and the sector number are stored in FIFOs for access by the Track Fitter. Since each detector layer is operating in parallel, there is a FIFO per layer. Each word is 32 bits wide, and our baseline design has 2000-word deep FIFOs.

#### 4.5.5 Track Fitter

Track candidates are fit using a linear calculation in place of a helical fit [30]. For a region of the detector sufficiently small, the linear approximation gives helix parameter and  $\chi^2$  resolutions close to those of a full helical fit (see section 3.1). The region we use is a sector, consisting of a physical silicon module in

Fit type	Number of scalar products (array size)				Total multiplications DSPs used
	$\chi$ -components	Pixel guess	SCT guess	Squaring	
Nominal	6(12)	-	-	1(6)	78
Pixel	6(12)	2(6) + 8(2)	-	1(6)	106
SCT	6(12)	-	1(6) + 7(1)	1(6)	91
Recovery	-	6(6) + 24(2)	5(6) + 35(1)	8(6)	149

Table 6: Summary of the number of products required in the Track Fitter for fits with hits in all layers (Nominal), Majority fits with one missing Pixel layer (Pixel), Majority fits with one missing SCT layer (SCT), and recovery fits (Recovery). The scalar product counts are separated based on the size of the array, and the stage of the fit calculation. A “-” is used when a calculation is not required by the fit. Each multiplication uses 1 DSP on the Arria V chip.

each layer. In the first-stage fit, there are 8 layers with 11 measured coordinates when there are hits on all layers (3 pixel layers each provide two coordinates). Since there are 5 helix parameters, the fit has 6 degrees of freedom each of which gives a function of the coordinates that should equal zero. The  $\chi^2$  is the sum of the squares of those functions:

$$\chi^2 = \sum_{i=1}^6 \left( \sum_{j=1}^{11} S_{ij}x_j + h_i \right)^2 \quad (2)$$

where  $S_{ij}$  and  $h_i$  are precalculated constants for that sector (see section 5.2) and  $x_j$  are the hit coordinates.

To decide which tracks to pass to the second stage, the first stage only calculates the  $\chi^2$ , not the helix parameters. In the second stage, when all 12 silicon layers are used, the  $\chi^2$  and the 5 helix parameters are determined. The helix parameters ( $\tilde{p}_i$ ) are also estimated with a linear calculation:

$$\tilde{p}_i = \sum_{l=1}^N C_{il}x_l + q_i \quad (3)$$

where  $C_{il}$  and  $q_i$  are constants and  $x_l$  are the  $N$  hit coordinates.

To avoid a large penalty for detector inefficiency, we allow one layer to not have a hit (Majority Fit). In that case, we can still use the same set of constants to calculate  $\chi^2$  provided that we have an estimate of where the hit in the empty layer would have been. The missing hit coordinate can be estimated using a linear function of the other hit coordinates. (This equation comes from minimizing the  $\chi^2$  over the position of the hit in the empty layer.) If the missing layer is in the SCT, we estimate one coordinate. If the missing layer is in the pixel detector, we estimate two coordinates.

As noted above, there are different types of fits: (a) a Nominal 8-layer fit that can be followed by Recovery fits if the  $\chi^2$  cut fails, (b) a Majority fit in which the missing hit is in a pixel layer, and (c) a Majority fit in which the missing hit is in a SCT layer. The fractions of the different kinds of fits needed were determined from simulation, with results that were rather insensitive to the parent monte carlo sample. As a result we have 1 full/recovery fitter, 3 pixel-Majority fitters, and 5 SCT-Majority fitters, with 9 fits done in parallel in order to reach the design goal of 1 fit per ns in each FPGA, as shown in the functional diagram in Fig. 58.

The Altera Aria V FPGA has 1156 DSPs of which we use 1000. The number of DSPs in a fitter varies with the type of fit being done (see Table 6). The scalar products are done using  $N$  multipliers where  $N$  is the array length, followed by a parallel adder for summing the products. So a 12-dimensional scalar multiplication uses 12 DSP multipliers and one parallel adder with 12 inputs.

A road enters the Track Fitter in the Road Organizer which determines which kind of fit is to be done and then puts the road information into the appropriate road FIFO. Based on the road’s sector number,

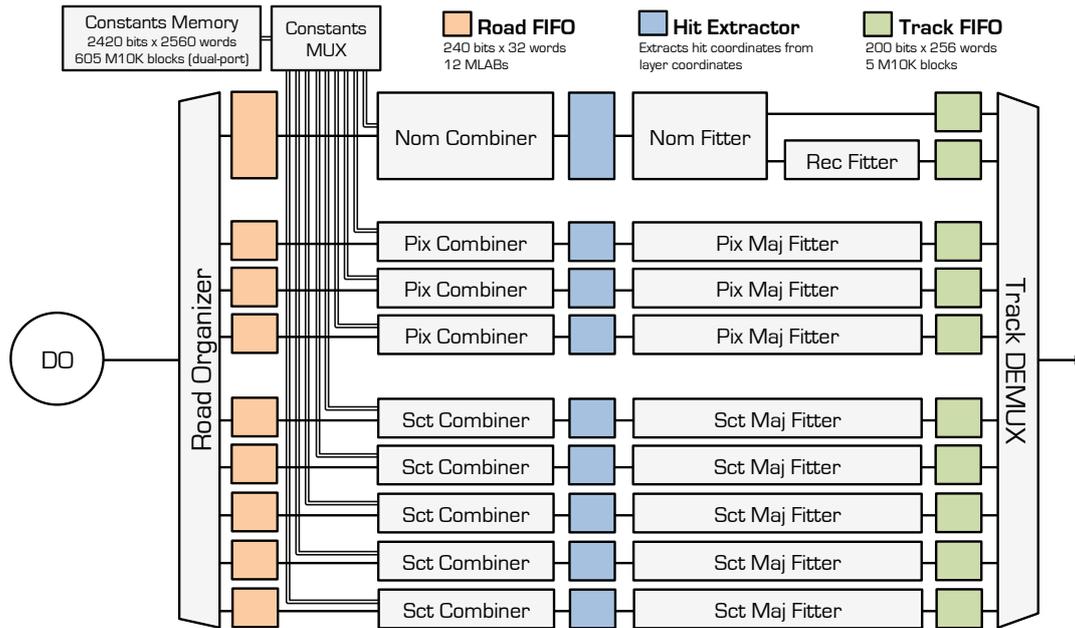


Figure 58: Functional diagram for the track fitter in an FPGA.

the Combiner fetches the constants needed for the fit. A set of constants occupying 2.4 kb is fetched in parallel in a single clock cycle. The Combiner then prepares the hit combinations to be fit, one hit per layer. The Hit Extractor converts the hits into the format needed in the DSP and pairs a hit with the constant by which it is to be multiplied. The linear fit calculation is then executed in a pipeline and tracks passing the  $\chi^2$  cut are pushed into a Track FIFO. The latency from the Extractor to the  $\chi^2$  cut is 64 clock cycles (5 ns/cycle). To avoid very long event latencies, a maximum of 500 fits per road are executed. Simulation at high luminosity shows that this restriction has negligible effect on tracking efficiency.

There are < 10k sectors in a tower. If the patterns were assigned to the 8 LAMB mezzanine cards in the tower such that each sector is in only one LAMB, then we would have 1.25k sectors per TF. To allow flexibility in balancing the load among the LAMBs, we provide constant space in the Track Fitter for 2.5k sectors. As a result, the memory needed for the constants is 6.2 Mb, and the total memory used by the TF is 7-8 Mb. When added to the 7 Mb used by the DO, a total of about 15 Mb is needed out of the 24 Mb in the Altera Aria V. This leaves substantial memory for FIFOs and Spy Buffers.

#### 4.5.6 First Stage Hit Warrior

The task of the Hit Warrior is to remove duplicate tracks. It is run in both the first and second stages. In the first stage, only tracks within the same road are compared. If a track shares more than a predetermined number of hits with a previously stored track in that road, the track with the higher quality factor is retained. The quality factor depends on the  $\chi^2$  and the number of pixel and SCT layers with hits. After all the tracks in a road have been processed, those tracks remaining are transferred to the Second Stage Board.

The Hit Warrior implementation is described in section 4.7.4.

#### 4.5.7 Wild-Card Processing

In order to retain good efficiency for tracks passing through a silicon module that is disabled for a significant period of time, the Associative Memory Board has the ability to apply the Wild-Card option. For purposes of pattern recognition, all superstrips in the dead module are treated as if they contain a hit. The Hit\_Map for matched roads passing through that module will have that layer set to 1, as for a real hit.

The operation of both the Data Formatter and Track Fitter is altered for roads with a Wild-Card. This is accomplished by modifying the DO's AM\_Map for that road and the TF's fit constants for that sector. For the former, the superstrip number for the layer with the Wild-Card will be set to a value that tells the DO not to fetch a hit from the HLM. The DO will then change the Hit\_Map bit for that layer to 0 so the TF knows there is no hit to be fetched from its input FIFO. For the latter, the fit constants will be those needed for a track missing a hit in the wild-card layer. The  $\chi^2$  cut in the TF will be appropriate to one fewer degree of freedom. Since the wildcard implementation requires modifying the pattern stored in the Associative Memory Board and the pattern and fit constants stored on the AUX, it is only foreseen to be used for semi-permanently disabled modules.

### 4.6 Associative Memory System

The associative memory system carries out pattern recognition at the high silicon detector readout rate by comparing hits at reduced resolution with a very large number of prestored patterns nearly simultaneously. The AM system consists of the Associative Memory chip (AMchip), an ASIC designed and optimized for this particular application, and two types of boards, a VME board (AMB) on which are mounted local associative memory boards (LAMB), a mezzanine that holds the AMchips. Both the AM chip and the boards have a long development history described in the FTK Technical Proposal. We report here only on the latest version, AMBFTK, LAMBFTK and AMchip04, developed specifically for the FTK project. We also describe their future evolution that includes an AMchip containing 128k patterns.

#### 4.6.1 The AMBFTK

The AMBFTK [31] is a 9U VME board on which 4 LAMBFTKs are mounted. Figure 59 shows the AMBFTK layout, highlighting one of the LAMBs (in yellow). A network of high speed serial links characterizes the bus distribution on the AMBFTK: 12 input serial links (in red) that carry the silicon hits from the P3 connector to the LAMBs, and 16 output serial links (each blue arrow represents 4 links) that carry the road numbers from the LAMBs to P3. These buses are connected to the AUX card through a high frequency ERNI P3 connector. A custom board profile was studied and simulated at the CAD station to guarantee a perfect board-to-board closure of the P3 connector without backplane support in that region. The data rate is up to 2 Gb/s on each serial link. Thus the AMBFTK has to handle a challenging data traffic rate: a huge number of silicon hits must be distributed at high rate with very large fan-out to all patterns ( 8 million patterns will be located on 128 AMchips on a single AMBFTK) and a similarly large number of roads must be collected and sent back to the AUX.

The motherboard has flexible control logic placed inside a group of FPGA chips visible in the figure. They are Xilinx Spartan6 FPGAs which have Low-Power Gigabit Transceivers (GTP). Ultra-fast data transmission requires specialized, dedicated on-chip circuitry and differential I/O capable of coping with the signal integrity issues present at these high data rates. Spartan-6 LXT devices have two 8-gigabit transceiver circuits. The incoming hits are received by the GTPs in the two input FPGAs (red boxes) and saved in large derandomizing FIFOs that are 4k words deep per link. Outgoing road IDs from the LAMBs (4 links/LAMB) are sent to the FPGAs (in the blue boxes) near the P3 connector. The FPGA inside the grey box is the Control chip. It is connected to all of the FPGAs in the AMBFTK to control

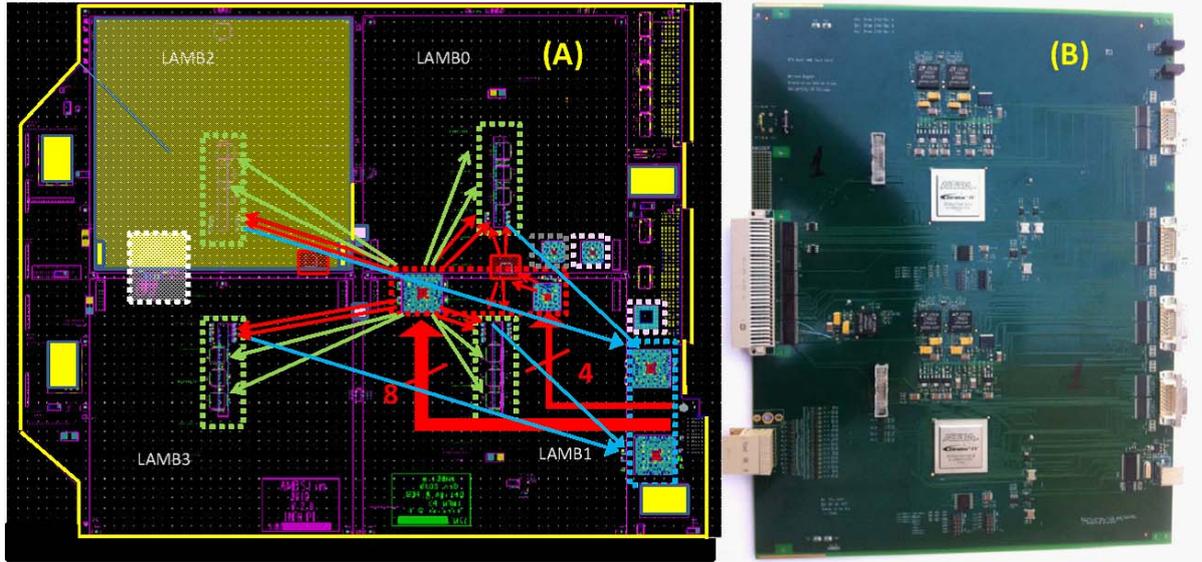


Figure 59: The AMBFTK board (A) showing the data paths to (blue) and from (red) the AUX card (B) that sits behind it. The large yellow square is one of the 4 LAMBs that almost totally cover the AMBFTK.

event processing and handle error conditions.

#### 4.6.2 The LAMBFTK mezzanine

The LAMBFTK and the AMBFTK communicate through an SMD connector placed in the center of the LAMB (inside the yellow central rectangle in Fig. 60, corresponding to the 4 green boxes in Fig. 59A). Each LAMB contains 32 AMchips, 16 on each side of the board, as in previous designs. However, the AM chips are new, AMchip04 [32]. They are 65 nm standard cell devices except for groups of 64 patterns each of which is a single custom cell designed to maximize the pattern density and minimize power consumption. They are in thin LQ208 packages (1.4 mm thick) and contain the stored patterns along with the readout logic. They have a core voltage of 1.2 V and an I/O voltage of 3.3 V. A large current (up to 25 A per LAMB) is provided through the two connectors in the yellow boxes at the bottom of Fig. 60 to support the 1.2 volts. The 3.3 V current is provided by the mezzanine central SMD connector. The 32 AM chips in each LAMB are connected as eight 4-chip pipelines (4 on the top side and 4 on the bottom side of the mezzanine). Each AM chip receives an input road bus from its upstream chip (in Fig. 60 the green arrows show the direction of the road flow) which it multiplexes with its internally matched roads onto a single output bus to its downstream chip. The roads flow down the 8 pipelines and are collected and merged into 4 streams by the two GLUE chips shown on the bottom of the LAMB inside green boxes. The GLUE chips are Spartan6 FPGAs containing high-speed GTP serial links that are used to transfer roads to the motherboard through the SMD connector.

The hits for each event, organized into 8 buses, 1 for each of the 8 detector layers, arrive at the LAMBFTKs from the AMBFTK through the SMD connector and are fed partially in parallel, partially serially. They are distributed to the 32 AMchips with a four-fold fan-out to distributor chips through the Input Distributor (INDI) chips shown in the red (Spartan6 FPGAs) and blue (Xilinx CPLDs) boxes. The mixed nature of this data transfer (half serial buses, half parallel) is due to the AMchip04 not having serial I/O. Serial transmission is preferred, but AMchip04 requires a translation from serial to parallel of each input bus before distribution of the data to the AM bank. This translation would have required too

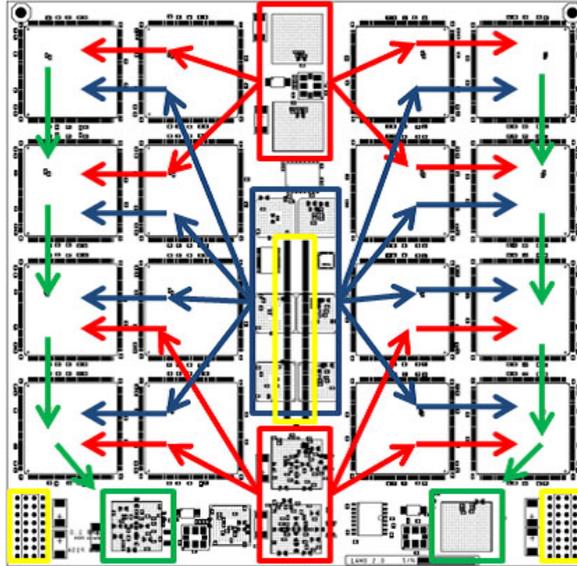


Figure 60: The LAMBFTK board showing the dataflow into and out of the board as described in section 4.6.2.

much space if applied to all 8 buses. (See section 4.6.5 for the upcoming change to all-serial data flow.)

The 4 buses that are distributed serially by the AMBFTK are received by the GTPs in the Spartan FPGAs, while the CPLDs receive the parallel buses. The CPLDs are located in the center of the board; the left column distributes the 4 buses to the left half of the board (see blue arrows), while the right column distributes the same buses to the right. There are also multiple Spartan chips, one set placed at the top and one at the bottom of the LAMBFTK, and they distribute their output to the bottom or top half of the board. The red arrows show how each bus is split four ways and distributed by the Spartan FPGAs while the blue arrows show the bus distribution by a pair of CPLDs.

**Event Processing:** When the Control chip starts to process an event, hits are popped in parallel from all the hit input FIFOs and are simultaneously sent to the four LAMBs. An End Event (EE) word, which includes the event tag, separates hits belonging to different events. The data in different streams have to be synchronized to guarantee that hits from the same event are being processed by the AM patterns. The input FPGAs contain deep FIFOs for this purpose, one for each serial link. If occasionally a FIFO becomes Almost Full, a HOLD signal is sent to the upstream board, which suspends data flow until more FIFO locations become available. The Almost Full threshold is set to give the upstream board enough time to react. The 12 Holds for the input serial links are sent back to the AUX through the P2 connector. Holds are also sent by the AUX to the AMBFTK to stop sending roads if an AUX FIFO receiving one of the road links becomes Almost Full.

AMchip04 is able to process two events using a two step pipeline: (1) hit loading, and (2) pattern matching and readout. While hits from one event (event  $N$ ) are being downloaded into the LAMBs, locally matched roads from the previous event (event  $(N - 1)$ ) are being collected from the LAMBs and sent to the AUX. When the EE word is received on a hit stream, no more words are popped from the relevant FIFO until the EE word containing the same event tag is received on all 12 hit streams. At that time, event  $N$  is fully loaded into the AM chips. Once the LAMBs have sent the last of the roads for event  $(N - 1)$  to the AUX, including the 16 EE words on all the 16 road serial links, event  $(N - 1)$  is considered fully processed. At that time, the Control chip sends INIT to push events forward in the pipeline: event  $(N - 1)$  is deleted by the AMBFTK since it is fully processed, the hit information for event  $N$  is copied

into the pattern matching and readout part of the AMchip, the hit portion of the pattern bank is reset for a new event, and the read-out of the roads from event  $N$  starts to complete the processing of event  $N$ . Finally, the hits from a new event,  $(N + 1)$ , are popped out of the input FIFOs and sent to the AM chips.

### 4.6.3 The AM chip

The architecture, technology, and prototype design of the new AM chip, AMchip06, maximize pattern density, minimize power consumption and improve functionality with respect to previous versions.

A full custom cell is the most important design change for the new ASIC AM chip. It includes all the hardware necessary for the elementary functions of a single pattern layer: SRAM bits, comparators, and a final latch to store the match. In the previous AM chip, AMchip03, these functions were implemented with a set of standard cells, unavoidably occupying more silicon area. We get a reduction of more than a factor two in the area required for a single pattern layer. Since the bank patterns occupies roughly 70% of the full AM chip (50% for the 14 mm<sup>2</sup> prototype), the new design provides a factor of 2 more patterns per chip.

The global readout and control logic is implemented with standard cells to minimize the development time. The density gain of the new full-custom cell combined with the gain due to scaling from 180 nm technology to 65 nm produces an estimated global increment in the number of patterns of a factor of 15. We can also expand the silicon area, since the package we plan to use can house a 16 mm × 16 mm chip, while the AMchip03 was just 10 mm × 10 mm. Considering that for ATLAS we need 8-layer patterns, while in CDF we used 6 layers, the final gain can be as large as a factor of 28. For the goal of  $128 \times 10^3$  patterns per chip, a die of 14 mm × 14 mm would be sufficient.

The full custom cell also offers the opportunity to implement important new strategies to reduce the power consumption of the chip. This is a crucial issue because the pattern density growth will eventually be limited by power consumption. The clock cycle is limited to  $\geq 10$  ns by the board complexity: each hit found in the detector has to be distributed to 128 AM chips per board, with a very high fan-out factor. However inside the 65 nm chip the 10 ns clock cycle is conservative. This clock period is used to ease the distribution of the data through the chip. In addition we can reduce the speed of the match operation in exchange for reduced power consumption. So we perform the pattern comparison using a “pre-match” technique that first compares the 4 least significant bits of each layer word, then after a successful pre-match the remaining bits are compared. The “pre-match” technique can save up to 80% in power consumption. The overall power consumption is slightly larger than in the old device, despite the large jump in pattern density and the higher operating frequency. Power consumption is another reason to limit the maximum operating frequency to 100 MHz.

The features of the new AMchip, and in particular the performance of the full-custom cell, have been evaluated in order to validate our expectations and to gain experience for further improvement of the associative memory technology. For these purposes we designed a MPW (Multi Project Wafer) prototype of the new AM chip with 65 nm technology. The main goal was to verify that the full-custom associative memory cell works properly and to verify the expected gains in pattern density and power usage. The AMchip04 MPW prototype uses a reduced silicon area of 14 mm<sup>2</sup> and is designed to store 8192 patterns, with an estimated core power consumption of 70 mW from the custom cells. The power consumption is the most difficult parameter to predict. The expected values for the custom cell were validated by measurements with the real device (see the next section). The remaining logic of the AMchip04 is implemented with standard cells and is very similar to the AMchip03 logic. The main change is that the new chip will perform pattern recognition with 8 layers instead of 6 or 12. The tests of AMchip04 show a power consumption of approximately 230 mW for the core logic that includes the full-custom pattern cells, the bitline switching and the standard cell logic contributions.

Table 7 compares the performance of AMchip03 with the expected performance of a full AMchip06 extrapolated from the current MPW AMchip04 design. AMchip05 is a small area, low cost MPW,

functionally equivalent to AMchip06, the only difference being the small pattern bank (only 4k patterns). It will be used in early testing of the AMBSLP and LAMBSLP boards (see section 11).

	AMchip03	AMchip04	AMchip06	Effect
Technology	180 nm	65 nm	65 nm	×8 pattern density
Clock freq.	50 MHz	100 MHz	100 MHz	faster, higher power consumption
Die size	10 × 10 mm <sup>2</sup>	14 mm <sup>2</sup>	12 × 12 mm <sup>2</sup>	×1.5 increased area
Core voltage	1.8 V	1.2 V	1.2 V	lower power consumption
Core power	1.3 W	0.23 W	3.6 W	
Selective precharge	No	Yes	Yes	~ 80% power saving
Full custom	No	Yes	Yes	×2 pattern density
Layers	6 (or 12)	8	8	$\frac{3}{4}$ pattern density
Patterns/chip	5k	8k	128k	
Bits/layer	up to 18	up to 15	up to 16	
Ternary bits/layer	N/A	3 to 6	3 to 6	better S/N (see text)

Table 7: AMchip parameters for the previous and new designs. The 3.6 W core power for AMchip06 is for 128k patterns. For a 64k-pattern chip, the core power would be 1.8 W.

**The AMchip working principle:** The AMchip is based on an array of CAM cells (Fig. 61). Columns are used to distribute the hit information over vertical buses called search bit lines (or just bitlines), whereas rows are used for the write lines (the signals that enable the write operation) and for the match lines (the signal that identifies a match). Each bitline bus consists of 36 lines: the 18 bits in a hit and the 18 corresponding inverted bits. Each row in the array corresponds to one pattern. A row is organized in sub-blocks of 18 CAM cells that we call a layer block. Each layer block stores the coarse resolution hit position in that layer for a pre-calculated track trajectory. A pattern is composed of 8 layer blocks, so it can identify a track crossing up to 8 detector layers. Each pattern contains the logic necessary to compare the stored position with actual hit position for each hit in one event. A pattern matches when all, or almost all, of the layers match hits in the input data for one event.

The event hit positions are received over 8 input buses of 15 bits each. This limits the maximum number of positions to 32,000 for each layer. This might seem a strict limit, however since different AM chips (or different groups of AM chips) can independently process data from different parts of the detector, 15 bits are enough to reach a granularity below 10 SCT strips or an area of 12×18 pixels, respectively, along the  $r\phi$  and  $z$  directions. The choice of 15 bits was based on the performance needs for FTK. The positions stored in each layer block are encoded in the following way. Of the 18 CAM cells each storing one bit, 12 are used to store the 12 most significant bits of the word, while the other 6 are arranged in 3 pairs and used as ternary cells storing either 0, 1 or X values<sup>10</sup>. The X value means don't care, so the hit present on the hit bus will match the stored word regardless of the values of the bits set to X. The use of the "don't care" (DC) feature, as in ternary CAMs, allows us to have variable size patterns [26]. Normally the least significant bit of each layer corresponds to a fixed area on the detector (e.g., 20 consecutive SCT strips). This area is the width of the matching window implemented by one pattern. When a bit is set to the DC value, the effective pattern size for that bit is doubled because it will match two numbers. With 3 DC bits, we can enlarge the coincidence window up to a factor of 8 for each pattern and for each layer independently. In this way patterns can be tailored to maximize the acceptance for valid tracks, while reducing the probability of matching spurious hit combinations. In other words each pattern has a better signal-to-noise ratio. Therefore, AM patterns are employed in a smarter and

<sup>10</sup>In AMchip04, the total number of bits used to encode a hit can be reduced from 15 to 12 while increasing the number of ternary cells from 3 to 6. Intermediate steps with 4 or 5 ternary cells are also possible.

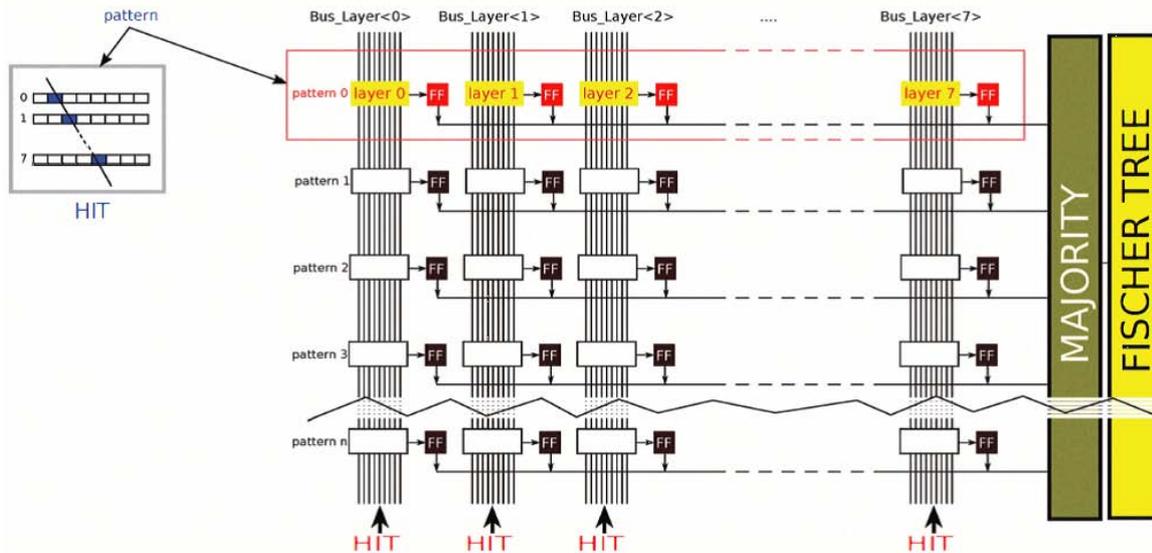


Figure 61: The AMchip array.

more efficient way. Patterns using only two ternary cells per layer are as effective as an increase by a factor 3 to 5 in the number of patterns without ternary cells [26]. This gain required an increase in chip area of just 3 CAM cells per layer which corresponds to 17% of the layer or 1 mm<sup>2</sup> total area. This is a very important improvement in associative memory technology.

Input data in the columns are compared in parallel with the data stored inside the layer blocks. If a layer block matches all 18 bits (accounting for DC bits in ternary cells), a Set-Reset Flip-Flop (SR-FF) is set to the high logic value. As we can see in Fig. 61, the majority block counts the number of SR-FFs set to 1. If this number is equal to 6, 7 or 8, the data are transferred to the AM readout block which is able to generate the address of the matched pattern by using a priority list. This block is based on a modified Fischer tree [33]. It is worth repeating that the majority block identifies not only matches of all layers, but also partial matching in which one or two layers for a given pattern are not hit<sup>11</sup>.

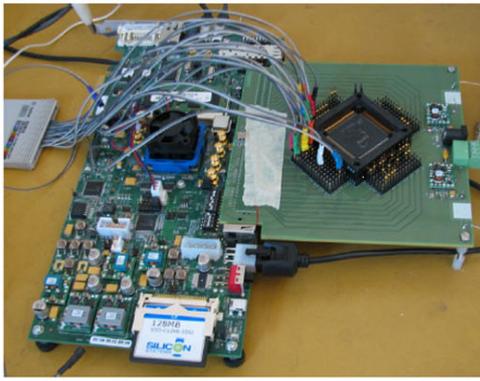
A detailed description of the chip design is given in Appendix I.

#### 4.6.4 Test of the integrated AM system

**AMchip04 Test Results:** The first prototype of AMchip04, which had an excellent yield above 80%, was tested and it is fully functional. Figure 62(a) shows the test setup. A Xilinx demo board ML605 was used as a pattern generator with the Virtex FPGA configured as a MicroBlaze RISC processor using the Linux operating system. The first tests were executed using an Agilent 16902B Logic Analysis System as the output analyzer. The test was performed by storing 8000 different patterns and by applying a sequence of 4000 different matching patterns at the input. This corresponds to the worst case condition when every input pattern matches. We observed excellent agreement between simulated and measured parameters. A small portion of the output is shown in Fig. 62(b). The prototype output data match the simulated data even at 100 MHz. More extensive tests were executed using the Xilinx demo board as the pattern generator and logic analyzer.

The power consumption was measured at frequencies from 15-100 MHz and is linear (see Fig. 63). We studied in detail the 50 MHz point and measured the various contributions to the ~ 100 mA total. Roughly 25 mA are drawn if no hits are distributed (mainly clock distribution), while the standard cell

<sup>11</sup>Our current plan is to only process roads in which all layers or all except one have a hit.



(a)

address	data				
	meas.	sim.			
C8600F98	1D00	<= 1D00	0 <= 0	0 <= 0	1 <= X
C8602530	1D40	<= 1D40	0 <= 0	0 <= 0	1 <= X
C8603AC8	1D80	<= 1D80	0 <= 0	0 <= 0	1 <= X
C8601060	1DC0	<= 1DC0	0 <= 0	0 <= 0	1 <= X
C86025F8	1E00	<= 1E00	0 <= 0	0 <= 0	1 <= X
C8603B90	1E40	<= 1E40	0 <= 0	0 <= 0	1 <= X
C8601128	1E80	<= 1E80	0 <= 0	0 <= 0	1 <= X
C86026C0	1EC0	<= 1EC0	0 <= 0	0 <= 0	1 <= X
C8603C58	1F00	<= 1F00	0 <= 0	0 <= 0	1 <= X
C86011F0	1F40	<= 1F40	0 <= 0	0 <= 0	1 <= X
C8602788	1F80	<= 1F80	0 <= 0	0 <= 0	1 <= X
C8603D20	1FC0	<= 1FC0	0 <= 0	0 <= 0	1 <= X
C86006BC	3FC0	<= 3FC0	0 <= 0	1 <= 1	1 <= X

(b)

Figure 62: Test of the AMchip04 prototype: (a) test-bed, (b) results

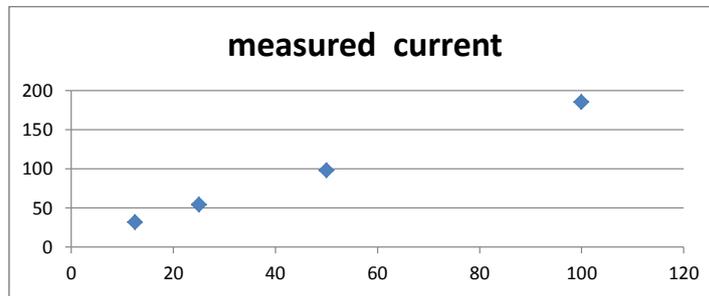


Figure 63: Total measured current (mA) in an intensive test as a function of frequency (MHz).

contribution (mainly due to the bitline swapping) is 40 mA (the estimated value was 32 mA) and finally the full-custom contribution is measured to be 35 mA (estimated value 29 mA). The measurement at 100 MHz extrapolates for a chip 8 times larger to 1.5 A or a power consumption of 1.8 W per chip. We are implementing strategies to further reduce it in the next version of the chip.

**AMBFTK and LAMBFTK Test Results:** The first AMBFTK prototype and its mezzanine were tested before soldering in AMchip04 prototypes (see Fig. 64). This was an important test since the two boards have a very large network of serial-links. For testing these connections, sequences of hits were downloaded through VME into the AMBFTK input FIFOs. When the FIFOs were full, their outputs were enabled and the data were sent at 2 Gb/s to the LAMB on the long serial links. The Spartan receivers in the LAMB knew the sequence and incremented an error counter if an input word was not as expected. We ran the test for ~ 70 hours with no error on any serial link on the board, including those crossing the SMD LAMB connector. After the board tests, a few AMchip04s were mounted on the LAMB. To test the complete system, we sent fake hits compatible with the downloaded pattern bank, read the output roads from the AMchip output FIFOs, and compared them with the expectation from simulation. Again the system worked correctly at 100 MHz, as expected.

#### 4.6.5 The evolution of the full FTK AM system

The power consumption of the chip requires us to provide a large number of VDD and GND pins. Already with the current small area chip (14 mm<sup>2</sup>), we are using all of the available pads on the LQ208

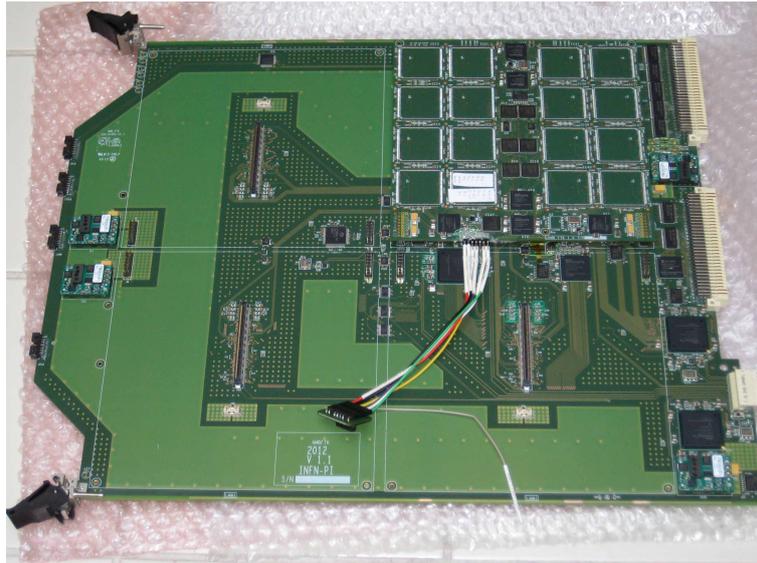


Figure 64: The AMBFTK with one LAMBFTK installed.

package. Increasing the chip area requires a change to a BGA package. Even so, the complexity of the LAMBFTK makes it extremely difficult to add a lot of additional VDD and GND pads and vias with the existing board design.

These problems led us to a different strategy for the processing unit, based totally on serial link communications, the Serial Link Processor (SLP). We are designing new boards, the AMBSLP and the LAMBSLP. The main differences between the system we have now (AMBFTK and LAMBFTK) and the new SLP are as follows.

- We bought an IP from Silicon Creations to provide serialized I/O buses inside the AMchip. The Silicon Creations Multirate SerDes (Serializer/Deserializer) macro includes all high-speed analog functions for a single channel (or multi-channel) serializer and deserializer and is optimized for low power operation at data rates from 584 Mb/s to 2.4 Gb/s. Alternative 10b and 40b input and output data paths simplify the design of link layers created from RTL using regular standard cells and regular synthesis, placement, and routing. AMchip05 will have 8 serialized Hit inputs and one output serialized Road address bus. In addition, there are also two input serialized buses that transmit addresses of fired patterns from neighboring chips. The two inputs are internally multiplexed with patterns found in the chip itself and sent to the single output.
- We investigated sophisticated packaging options and found the best solution, given our available funds, to be a wire bond BGA, with I/O optimized to work at 2 Gb/s.
- We decreased power consumption by reducing the core voltage to 1 V. This complicates the AMBSLP because the AMchip will need a core voltage of 1.2 V for the Silicon Creation IP and 1 V for the rest of the chip. In addition, the voltage for the I/O for both FPGAs and AMChips will be 2.5 V. A study is ongoing to further reduce the custom cell voltage down to 0.8 V.

As a consequence of the all-serial strategy, the LAMBSLP layout has been squeezed compared to the LAMBFTK to leave space for additional DC-DC converters and filters. Figure 65 shows the floor planning performed in CADENCE for one fourth of the new LAMBSLP. (The four sectors will be identical, so we will produce a first prototype that covers only one fourth of the mezzanine.) The AMchips are the

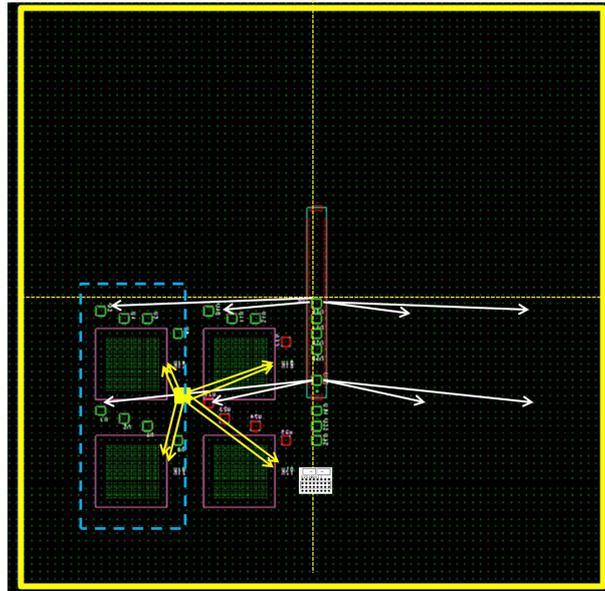


Figure 65: Floor plan for  $\frac{1}{4}$  of the LAMBSLP showing the data flow.

large squares ( $23 \times 23 \text{ mm}^2$  BGA). In the  $\frac{1}{4}$ -board design, 4 AMchips are placed on the top of the board. They are organized into two pairs, one on the left (shown inside a blue dashed box) and one on the right.

Each AMchip receives a low jitter LVDS input clock generated by a crystal placed in the middle of the  $\frac{1}{4}$ -board. The yellow box in Fig. 65 contains the crystal and a fan-out buffer in a common package to provide the clock to all 4 AMchips. The only signal that is asynchronous with the local clock is the INIT signal provided by the AMBSLP, but it is wide enough to be latched by the local clock.

All the other small chips in the figure are 2.5 GHz including the 1:4 LVDS fan-out buffers used to distribute the Hit input buses. From the SMD connectors the 8 Hit buses arrive on serial links. We are focusing on the design of the bottom portion since the top one is identical. Eight incoming serial links have to be distributed point-to-point to 4 AMchips in each quarter of the LAMB for a total of 32 links. This is a challenging network to design since each serial link must be a differential pair with uniform impedance, and the routing must be done on a single PC layer with no discontinuity except for the final connection to the pins. To accomplish this, we have implemented a tree of fan-outs that provide distribution through parallel vertical or horizontal lines with no crossings. As a result the routing is done with a minimum number of layers.

The pin-out on the AMchip is chosen so that the routing appears very regular with parallel vertical lines, both in the right and left quartets. Figure 66 shows how the chip differential inputs (8 hits inputs, 2 address inputs and one clock input) and one output are distributed on the border of the BGA.

Figure 67 shows the distribution of the 32 input links in the quarter-LAMB. The links are horizontal or vertical using only 3 layers (the white, pink and green planes). In addition there will be 8 other horizontal links that will connect the address buses to be merged into a single link to the SMD connector. The LAMB will have 4 output links, one per  $\frac{1}{4}$ -LAMB. The new LAMBSLP is better organized than LAMBFTK which had very complex routing (see Fig. 68). As seen in Fig. 65, the new LAMBSLP can also be substantially smaller than the LAMBFTK (the yellow external square shows the size of the old LAMB). This is important because it leaves free space on the AMBSLP to add new larger DC-DC converters (see Fig. 69).

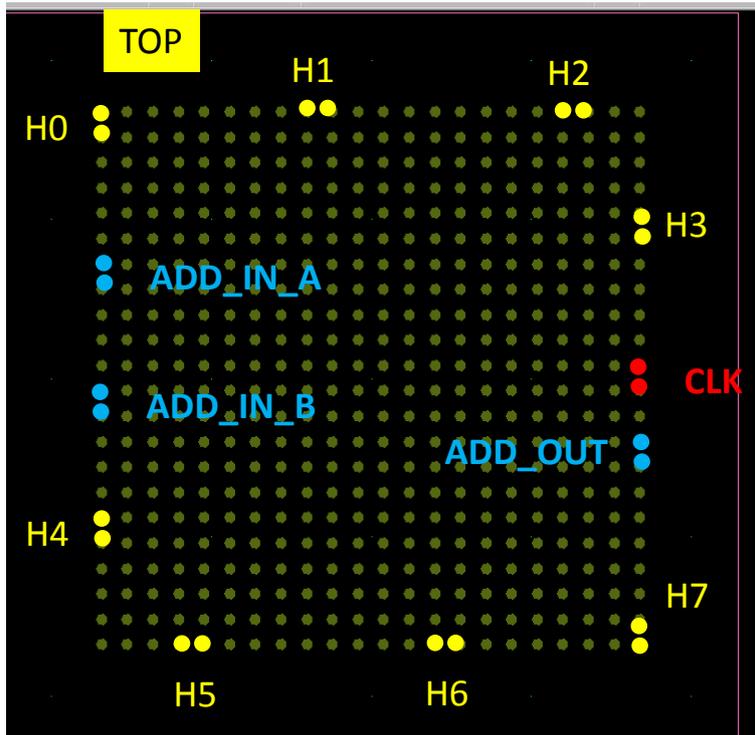


Figure 66: Serial link pinouts on the BGA.

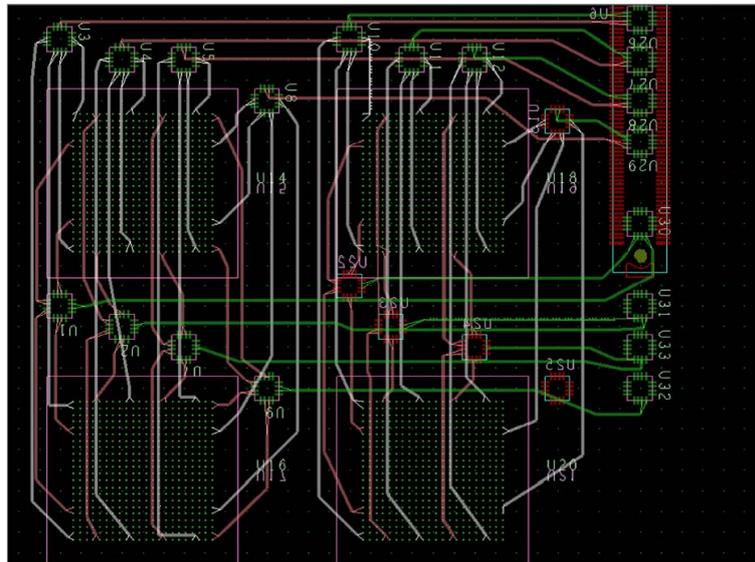


Figure 67: Routing of the 64 serial links needed to distribute the Hits to 4 AMchips

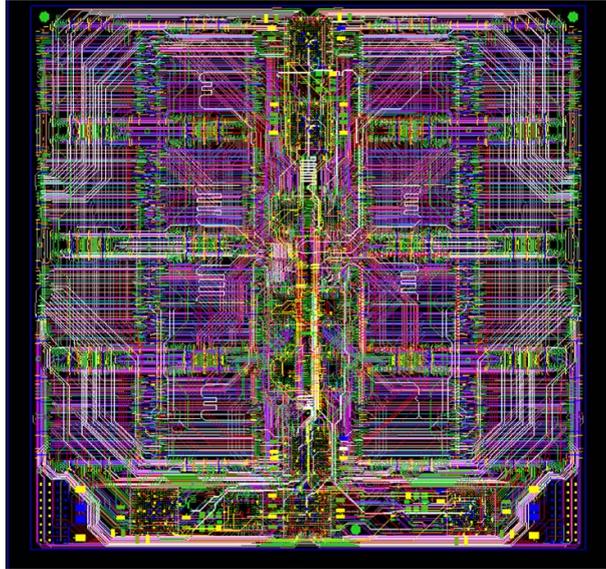


Figure 68: LAMBFTK routing.

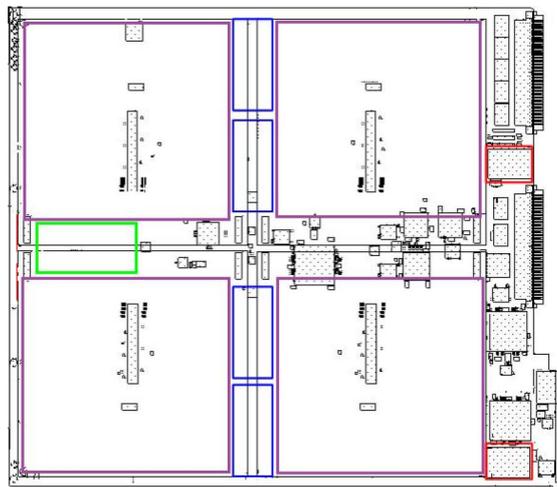


Figure 69: AMBSLP with the needed DC-DC converters: the 4 blue ones in the center provide 40-50 A at 1 V to each LAMB for the AMchip core, the green one provides up to 80 A at 2.5 V (I/O) for the whole board, and the red one provides 3 A to each LAMB at 1.2 V for the IP cores.

## 4.6.6 Voltage and power for the AM chip core

Large current must be provided for the core of the AM chips. An extrapolation of the AMchip04 current measured at 100 MHz implies that the current needed for the AMBSLP is 48 A per LAMB at 1.2 V, for a total consumption of 230 W per AMBSLP. A significant power reduction can be obtained by lowering the voltage, which also produces a current reduction (during AMchip04 tests we observed that operating at 1 V produced a reduction of roughly 20% in current). We studied two different approaches to reducing the voltage:

1. The existing custom cell can run at 1 V, since it is much faster than needed for our slow clock period of 10 ns. Using 1 V would imply (a) rerunning the simulations for full characterization at 1 V, (b) substituting the appropriate library for standard cells, and (c) using voltage shifters to interface the core of the AM bank with the core of the new SERDES IP which needs 1.2 V. For 1 V we would use one 24 V  $\rightarrow$  1 V DC-DC converter per LAMB, which can produce 50 A for a total power of 200 W. This solution could provide the needed current even if the AMchip were operated at 1.2 V, like AMchip04 (option 3).
2. We could gain much more by redesigning the custom cell to operate at lower voltage and making small changes at strategic points. This is the ongoing work that could produce a significant decrease in power consumption. We would use one 12 V  $\rightarrow$  0.8 V DC-DC converter per LAMB which could produce 40 A for a total power of 128 W. We would also need voltage shifters to interface the voltage needs of the different sections (standard cell logic and IP core).

In conclusion the AM chip core power will be less than 1.5 W for option 1, substantially less than 1.2 W for option 2, and 1.8 W for option 3. The final decision will be made in the near future (review at the end of June) taking into account the advantages of 2 on one side but also the schedule constraints on the other side. In the worst case, 3, the AMBSLP would require 230 W, and in the best, 1, probably half that amount.

Details of the cooling tests that were carried out are given in Appendix II.

## 4.7 Second Stage Board

In the second stage, tracks from the first stage are received from the AUX card and combined with hits from Data Formatters (DF) for the remaining layers. The second stage is needed to reduce fake tracks which occur at a large rate at high luminosity. It also improves helix parameter resolution (especially  $z_0$ ) since it performs fits using all 12 detector layers.

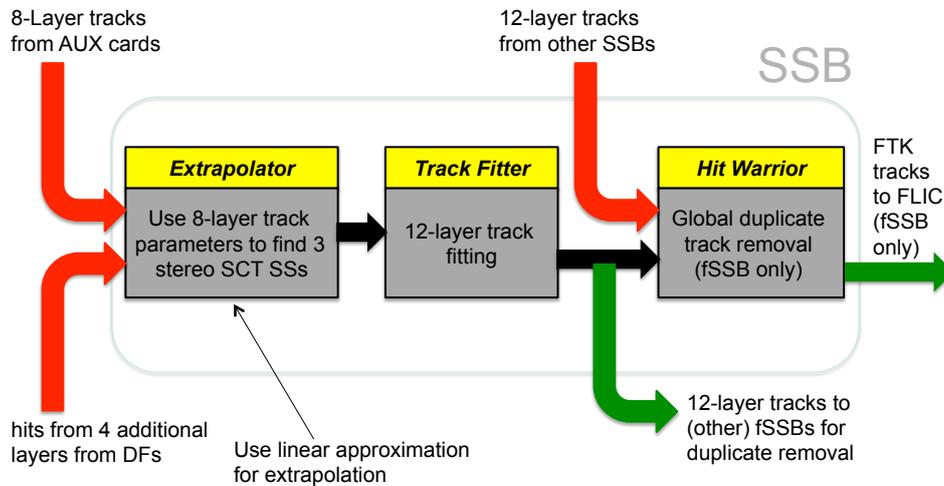
Each Second-Stage Board (SSB) receives through a Rear Transition Module (RTM) the output from 4 AUX cards and the hits on the additional layers from the Data Formatter system for the  $2\eta - \phi$  towers associated with those AUX cards. The SSBs perform duplicate track removal (currently based on the number of shared hits between tracks) and also share track data with other SSBs for  $\eta - \phi$  overlap removal. The SSBs merge FTK data within a core crate for output to the FLIC crate via two fiber-optic connections on the RTM.

### 4.7.1 Design features

The SSB functions are summarized in Fig. 70. These functions are implemented in firmware loaded into separate FPGAs on each SSB. They are:

- *Extrapolator*: Use 8-layer track information to compute likely positions of hits in the other 4 layers for use in 12-layer track fitting;
- *Track Fitter (TF)*: Determine  $\chi^2$  and the helix parameters from hits in roads using 12 silicon layers;

# SSB Functional Diagram



Note: All SSBs will be identical hardware

1

Figure 70: Diagram summarizing the three primary functions of the SSB.

- *Hit Warrior (HW)*: Remove duplicate tracks based on the number of shared hits.

Because of the need to do duplicate track removal across tower boundaries due to the tower overlap, it is most efficient to concentrate the HW function in those SSBs that send the final tracks to the FLIC cards. Consequently there are two types of SSBs in the system that alternate in SSB slots within a core crate and have identical Extrapolator and TF functions but different HW functions to perform duplicate track removal:

- *Preliminary (or “pre”) SSB (pSSB)*: Sends its tracks to the  $+\phi$  neighboring fSSB in the same core crate for output to the FLIC. Also sends its tracks to the neighboring fSSBs (excluding the fSSB just mentioned) solely for use in duplicate track removal. Since the pSSB doesn’t do duplicate track removal, but sends its tracks to adjacent fSSBs where that function is carried out, the “HW” on a pSSB is essentially just a fanout.
- *Final SSB (fSSB)*: Receives tracks from its own TF and the  $-\phi$  neighboring pSSB for output to a FLIC. These tracks are only output if they are not duplicate with any tracks from two  $\phi$  neighboring pSSBs (excluding the pSSB just mentioned) and two  $\phi$  neighboring fSSBs.

Its important to note that all SSBs will have identical hardware; only the firmware in the HW will vary.

## 4.7.2 Functional Layout, Interconnections, and Physical Design

Each SSB plugs into a Rear-Transition Module (RTM) located in the back of the VME crate using an ERNI-204781 connector. The purpose of the RTM is to provide a fiber optic transceiver interface to other boards in the FTK system. The RTM contains a series of SFP+ modules with signals routed to a connector in the P3 area and mated to an equivalent connector on the SSB. Special considerations

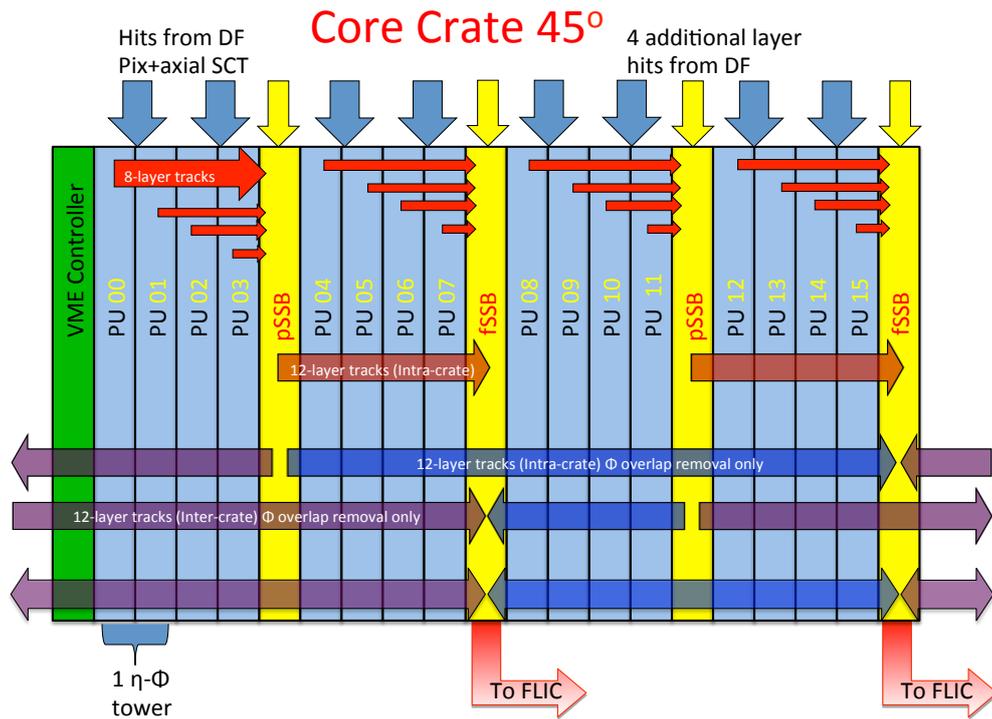


Figure 71: FTK core crate showing the data into and out of the SSBs.

are given to the P3 connector and proper power/grounding on the RTM and SSB to accommodate the multiple high-speed serial lines.

A diagram showing the boards in an FTK core crate is shown in Fig. 71 with emphasis on SSB inter-crate and intra-crate data flow. A diagram showing the functional layout for the pSSB and fSSB are shown in Fig. 72 and Fig. 73, respectively. The physical layout of the SSB and its rear transition module is shown in Fig. 74.

#### 4.7.3 Input Data Flow

At a 100 kHz level-1 trigger rate, the track information sent by the Processor Unit AUX card corresponds to 50 Mwords/sec, which can be accommodated with a single SFP+ fiber link per AUX card (see section 4.5.2).

The SSB receives from the Data Formatter hits from 4 silicon layers. Based on Table 5 and the worst case of an inner pixel layer plus 3 SCT stereo layers as input, the data volume to the SSB is estimated to be 3.6 Gb/s per link, which is below the maximum bandwidth of a single SFP+ link.

#### 4.7.4 Primary Functional Elements

**VME Interface** The VME interface will be used for board configuration and diagnostics, read/write access to memory used to store the 12-layer constants, and for readout of the SpyBuffer memory. It will also provide the ability to change the FPGA firmware, which is especially useful during the early

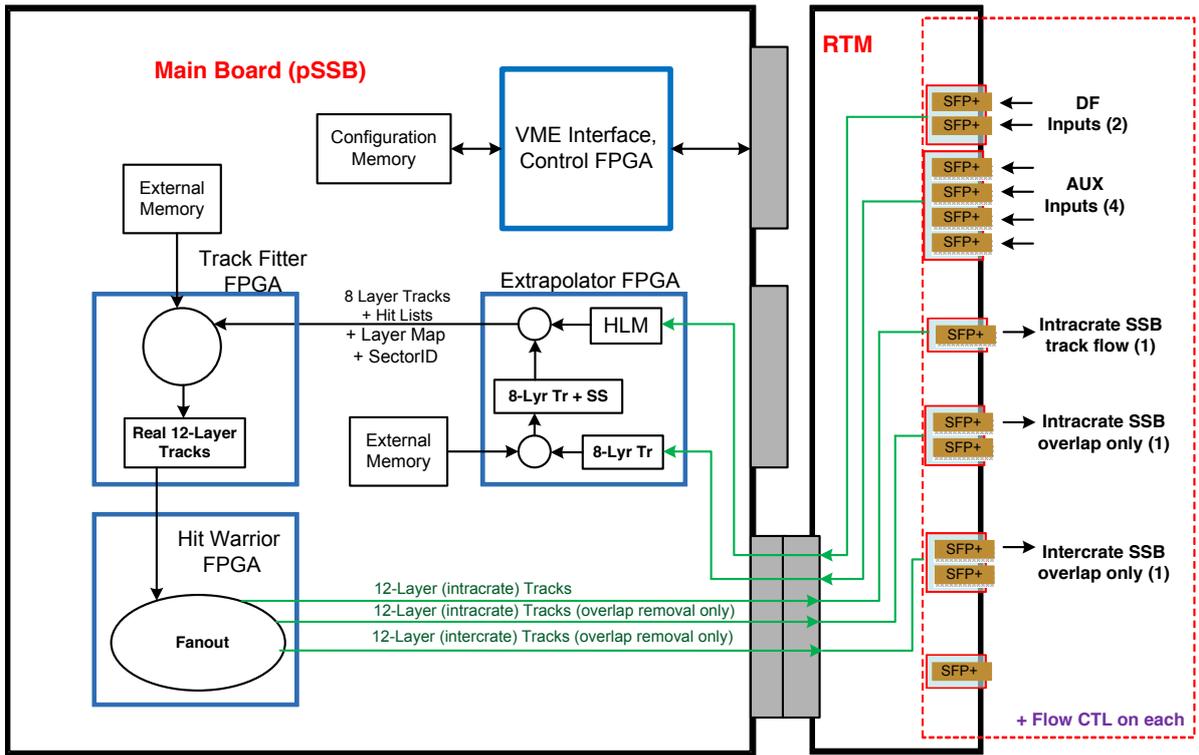


Figure 72: Diagram showing the functional layout of the pSSB.

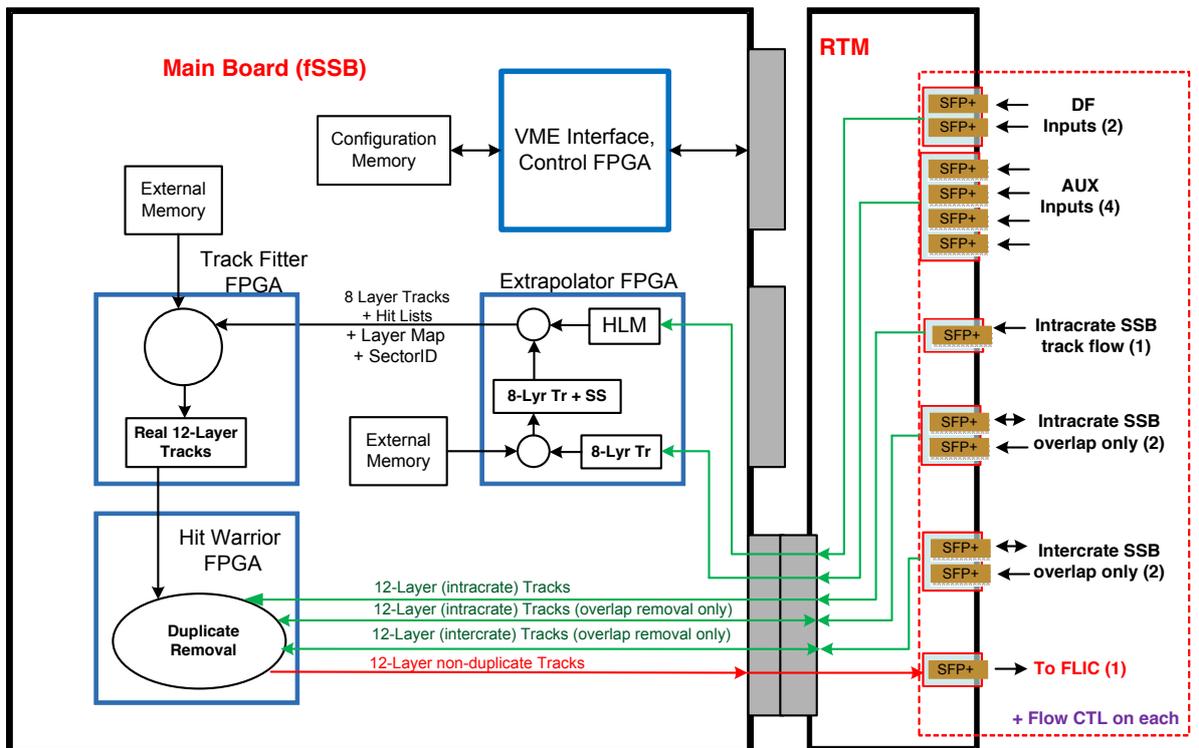


Figure 73: Diagram showing the functional layout of the fSSB.

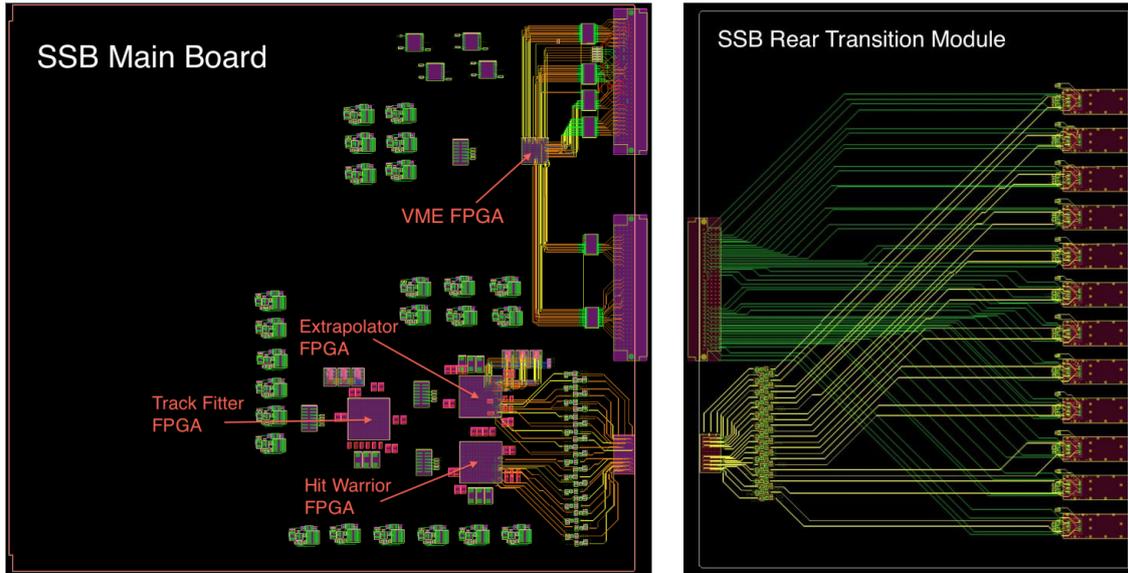


Figure 74: The physical layout of the SSB and its rear transition module.

development phase.

**Multi-Stream Synchronization** The SSB has multiple points where data streams from off-board sources must be merged together. They are not synced with one another, but they are sent in packets that contain a 32-bit Extended Level-1 ID (ELID) which provides unique packet identification. The SSB synchronization engine (SyncEngine) provides an algorithm to line up packet streams to a common ELID and performs a post-alignment check on subsequent ELIDs to maintain packet synchronization.

The SyncEngine is designed to be scalable and accommodate a variable number of input streams. It is implemented at the front end of the Extrapolator, Track Fitter, and Hit Warrior functions of the SSB. The Extrapolator receives 4 streams of data from the Data Formatter and 4 streams of data from the AUX cards. The Hit Warrior may receive data from an onboard Track Fitter and from Hit Warriors on 5 other SSBs. In both cases, packets from multiple sources must be processed from the same event.

**Extrapolator** There are a total of 12 layers in the FTK system. In the first stage, 8 layers are used<sup>12</sup>. In the second stage, information from the other 4 layers is used to further improve the track quality. The purpose of the Extrapolator is to compile hit lists in the layers not used in the first stage so the 2nd-stage Track Fitter can loop over them.

There are four basic Extrapolator steps:

1. Obtain the most likely hit position in each unused layer;
2. Look up the Super-Strip ID (SSID) that contains the hit coordinate;
3. Compile a list of hits in that SS and the 2 adjacent ones;
4. Do a majority logic test to check that there are at least 10 layers with real hits. If so, then pass the track to the TF.

A functional diagram for the Extrapolator is shown in Fig. 75. The Extrapolator FPGA requires

<sup>12</sup>The default is 3 pixel layers, 4 SCT axial layers and 1 SCT stereo layer.

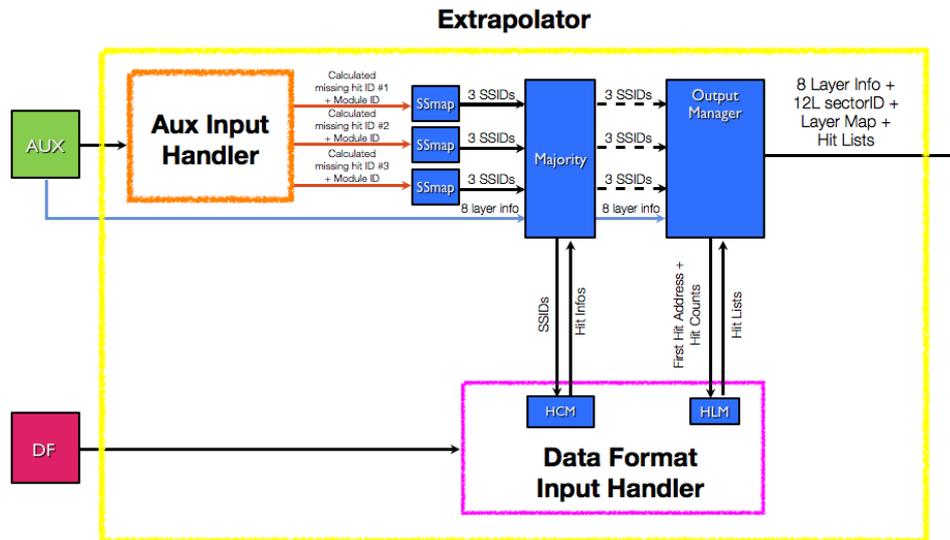


Figure 75: Functional description of the extrapolator. Tracks sent from the Aux input are used to calculate the most likely hit coordinates in the additional layers, while the Data Formatter input hit lists are inserted into a structured array. Once hits have been found, the majority logic is applied to determine whether this is a valid track candidate. If so, the output manager reads the hit lists, compiles the output stream, and passes it to the Track Fitter.

access to sets of sector constants to do the extrapolation. There are 36 24-bit constants in a set. With 26,000 sectors per board, this is approximately 3MB. Given an expected rate of 200 tracks/SSB/event and a 100 kHz level-1 trigger rate, on average 50ns are available per constant set or 1-2ns per constant. Xilinx provides Memory Interface modules to handle signal exchanges between discrete off-chip memory and user application code within an FPGA. For the Extrapolator, we use a 48-bit wide data bus to handle two constants per memory address and 8-burst access to DDR3 memory chips on-board the SSB. Modular VHDL code implemented in the Extrapolator handles read/write operations for the Memory Interface.

**Track Fitter** The Track Fitter (TF) firmware developed for the AUX (see section 4.5.5) is being modified for the SSB to carry out the 12-layer fits. The TF functional diagram is shown in Fig. 76. The inputs from the extrapolator are first sent to the Road Organizer (RO) and the constants MUX. The job of the RO is to look for the next available fitter block and forward the 8-layer track and candidate hits in the missing layers. Which type of fitter block is used depends on whether all 4 missing layers have candidate hits or not. If they do, the NOM fitter blocks is used,; if not, the SCT fitter blocks are used. At the same time, the 12-layer sector ID is sent to the constants MUX where the necessary preloaded constants are pulled from memory and forwarded to the same fitter block.

Inside the fitter blocks, the 8-layer track and candidate hits are first loaded into a FIFO. They are sent to the combiner which builds all possible track combinations from the hits and sends them to the fitter along with the constants. Each of these tracks is also fed into a shift register for temporary storage while the fit is being performed. The extractor extracts the existing 8-layer hit coordinates and arranges all 12 hits in an easy to use format for the fitter.

The fitter uses the constants and the 12 hit coordinates to calculate the helix parameters and  $\chi^2$ . If the  $\chi^2$  is below a cut value, the fit FIFO is informed that the track is to be saved.

If the  $\chi^2$  is above the cut and this is an SCT fitter block, the track is dropped. However, in the NOM fitter block, the majority recovery is done at this time. Four new tracks candidates are fit, by not using

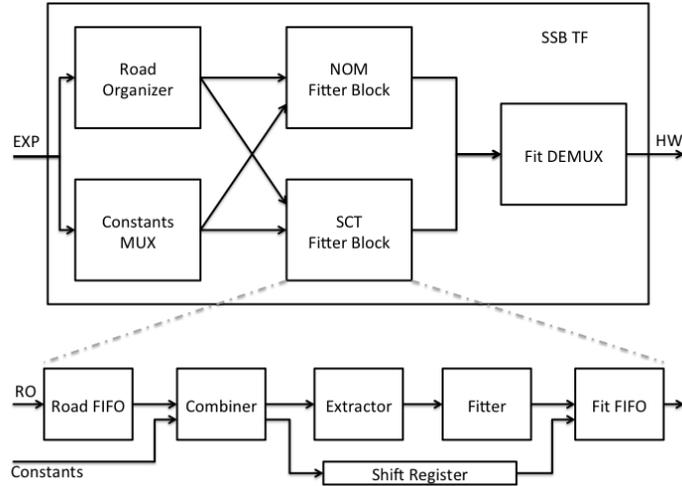


Figure 76: Functional diagram of the SSB track fitter. The SCT fitter block is enlarged below. The NOM fitter block (not shown) has a recovery fitter which differentiates it from the SCT fitter blocks.

Track 1	Track 2	Decision
12 layers	12 layers	Remove track with highest $\chi^2$
12 layers	11 layers	Remove 11-layer track
11 layers	11 layers	Remove track with highest $\chi^2$

Table 8: Example duplicate track removal criteria for the Hit Warrior

the hit coordinate from one of the 4 new layers. All 4 of these tracks are fit in parallel and the track with the best  $\chi^2$ , assuming it is below the cut, is read into the fit FIFO.

The successful 12-layer fits are sent to the fit DEMUX where the fits from the many fitter blocks are recombined into a single output stream which is sent to the Hit Warriors (HW) for duplicate removal.

**Hit Warrior** The main task of the Hit Warrior (HW) is the deletion of duplicate tracks in an event. There are two versions of the HW, one on the AUX that deals with 8-layer tracks and only removes duplicates within a single road, and another version on the SSB that removes duplicate 12-layer tracks within an entire event.

In order to maintain high efficiency in the Track Fitter (TF), two hits in a track are allowed to be missing. Therefore, the HW must count how many hits are in each track in addition to how many hits the two tracks share. Two tracks are considered duplicates if they share more than a certain number of hits,  $N_{\max}$ , a value that is easily configurable within the code and will be tuned to maintain high efficiency and low duplicate rate.

When two tracks share more than  $N_{\max}$  hits, they are considered duplicates and one is removed. The simplest selection criteria are given in Table 8. The HW code allows for more advanced criteria to be easily added in the future. For example, the pixel layers and SCT layers are counted separately which would allow for decision schemes that favor missing SCT layers over missing pixel layers.

The HWs, both pSSB and fSSB, use the same Xilinx Virtex6 FPGA. As noted above, the pSSB only uses the HW as a fanout. The pSSB HW has only one input coming from the onboard Track Fitter, and the data is fanned out to the three neighboring fSSBs.

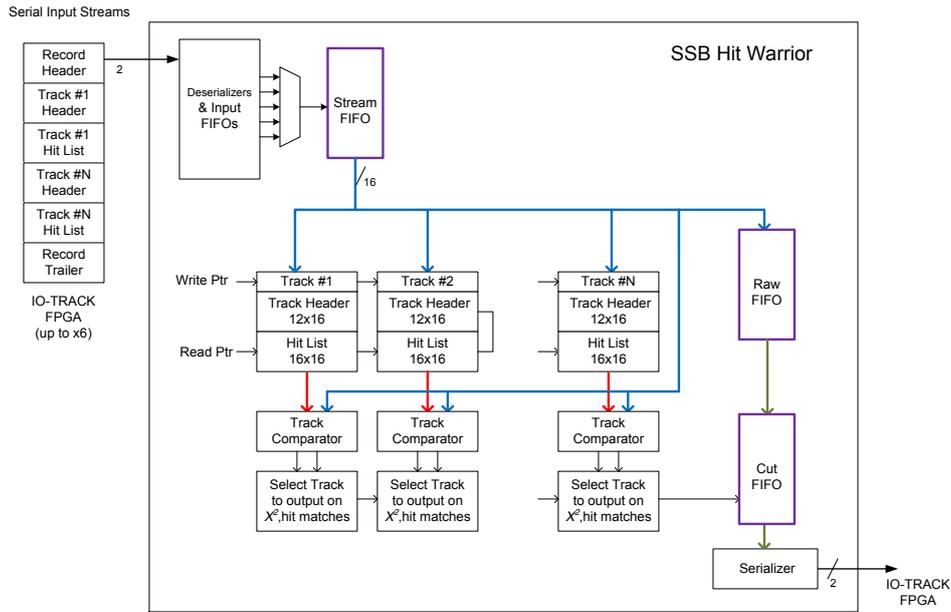


Figure 77: Functional diagram of the fSSB Hit Warrior. Not shown is the onboard Track Fitter data being passed through the HW and sent out to 2 other fSSBs.

The functional diagram of the fSSB HW is shown in Fig. 77. All memory elements used in the design are internal to the FPGA; no external memory is necessary. The fSSB HW receives 6 serialized input streams. One comes from the onboard Track Fitter, three are from other SSBs in the same crate, while two come from SSBs in a neighboring crate (see Fig. 71). This allows duplicate removal on a global scale.

The fSSB HW has three serialized output streams. One is the primary output that is sent to the FLIC. The other two are simply a pass-through of the onboard Track Fitter data to the neighboring fSSBs for overlap removal. Once the signals are received, they are deserialized and then sent to a synchronizer which synchronizes the six streams by `Level1_ID` and merges them into a single stream that is stored in the Stream FIFO.

The data are pulled out of the Stream FIFO one event at a time and transferred to the Raw FIFO. While this is happening, the track frames are also being loaded into individual RAM elements. The output of each RAM element is tied to its own track comparator. The track comparators are also connected to the Stream FIFO.

The first track in an event is loaded into RAM element 1 and the Raw FIFO, 1 clock cycle for each of the 12 layers plus 12 clock cycles for the track header. The second track is then loaded into RAM element 2, the Raw FIFO, and is compared to track 1 simultaneously. The third track is then loaded into RAM element 3, the Raw FIFO, and compared to both track 1 and track 2 simultaneously. The last track is loaded into RAM element N, the Raw FIFO and simultaneously compared to all previous tracks received in the event.

This means that by the time all tracks are transferred from the Stream FIFO to the Raw FIFO, all track comparisons have already been made. In order for the track comparisons to be made this quickly, it is necessary that the track data line up perfectly, so that all the RAM elements can share common pointers. This is why the hits must be in the same order every time and why placeholders are needed for missing hits.

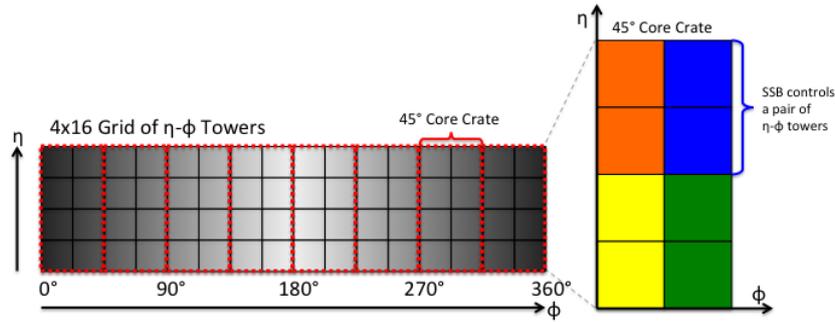


Figure 78:  $\eta - \phi$  mapping within a core crate. Each color represents the 2  $\eta - \phi$  towers that are connected to a single SSB.

Once all the comparisons between tracks have been made, the tracks that survived are transferred from the Raw FIFO to the Cut FIFO. Adding this extra stage to the HW allows for the removal of random gaps that would otherwise be created in the stream upon the removal of a track. Finally, the event is emptied from the Cut FIFO, serialized and sent out of the FPGA. The event is sent out as one continuous record, but there are idle gaps between events.

It is necessary for the entire event to be received by the Raw FIFO before it can start to transfer its data to the cut FIFO. This is because the last track in the event has the potential to be a duplicate to any other track in the event. Therefore, all comparisons must be made before the transfer can begin. However, as the event is being transferred to the Cut FIFO, the next event can be loading into RAM and the Raw FIFO simultaneously.

Similarly, the Cut FIFO cannot be drained until the entire event is received so that the entire event is sent out as a single record. Again, this can be done in parallel with the other processes. This allows the HW to process 4 events simultaneously: one event being read in and synchronized, a second event being loaded into the Raw FIFO while simultaneously having its tracks compared to one another, a third event being transferred into the Cut FIFO removing random holes created by track removal, and a fourth event that is being sent out of the FPGA.

Each core crate covers the full rapidity range and  $45^\circ$  in  $\phi$ . There are 8  $\eta - \phi$  towers in a crate each covering one quarter of the rapidity range and  $22.5^\circ$  in  $\phi$ . A crate has an overlap of  $10^\circ$  with the neighboring crate that must be checked for duplicates. There is also internal overlap between boards in the crate that can produce duplicates. As shown in Fig. 78, each SSB processes 2  $\eta - \phi$  towers (4 AUX cards) with an overlap in  $\eta$ , and the neighboring SSB also has overlap in  $\eta$  for the same  $\phi$  strip. For this reason, it is easier to group the SSBs in pairs so that half the crate is dedicated to the lower  $22.5^\circ$  of  $\phi$  and the other half of the crate is dedicated to the upper  $22.5^\circ$  in  $\phi$ .

The  $\eta$  overlap removal is easily done by linking the 2 SSBs (one pSSB and one fSSB) within a given  $\phi$  strip to each other. All  $\eta$  overlaps are contained within a single pair of SSBs. This leads to 2 pairs of SSBs within a crate, each processing tracks for one  $\phi$  strip.

The  $\phi$  overlap duplicate removal is more complicated since each pair of SSBs shares tracks with the other pair of SSBs within the crate and with a third pair of SSBs in a neighboring crate. This leads to the wiring scheme shown in Fig. 79. Each fSSB has 2 primary inputs (shown in green) which are the only tracks that the specific fSSB will ever output. This means each fSSB only outputs good tracks within its given  $22.5^\circ$  of  $\phi$ . The primary inputs are read into the HW first and loaded into RAM. As the tracks are loaded into RAM, they are simultaneously compared to one another and scanned for duplicates as outlined earlier. Then the tracks from the 4 secondary inputs (red going to increasing  $\phi$  and blue to decreasing  $\phi$ ) are read into the HW. Note that these tracks do not need to be loaded into RAM, as they

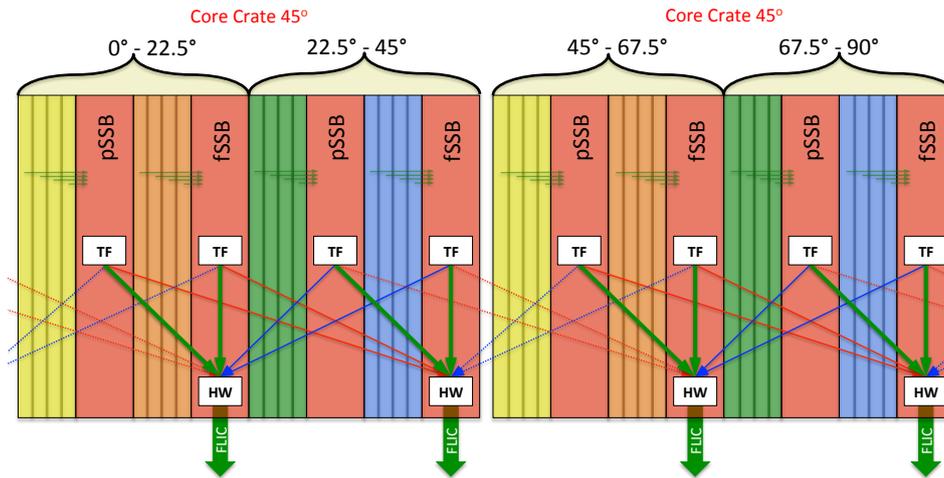


Figure 79: Wiring scheme for connecting SSBs together for global duplicate track removal. All data flows from Track Fitter to HW. Green arrows show the flow of data, while red and blue copies of data are sent to neighboring HWs for overlap removal. All communication between SSBs is handled inside the HWs. Data that simply passes through a pSSB HW is shown as a direct link between the originating Track Fitter and its target HW.

are only brought into this HW to find potential bad overlap tracks within the given  $\phi$  strip and remove them. Since either one of the 2 neighbors could have the better of the two tracks, this communication between neighboring  $\phi$  strips needs to be bidirectional. Each surviving track will then be sent out of the crate to the FLIC on one of two distinct connections corresponding to different  $\phi$  strips.

This method requires 2 fSSBs within each crate, each with 6 inputs to the HW. Each fSSB is connected to the other 3 SSBs within the crate and 2 external SSBs within the neighboring crate.

#### 4.7.5 Output Data Flow

The data sent from the SSB to the FLIC consists of event records, one record per event containing a header, track data blocks (track frames), and a trailer. Each track frame contains the data from a single track: a header and the track words (see section 4.8.3).

### 4.8 FTK Level-2 Interface Crate (FLIC)

After an event is processed by the core crates and the track information produced, the data is sent to the FTK Level-2 Interface Crate (FLIC) from the fSSBs as encapsulated records. The FLIC receives these records, processes them, and then sends the output to the High Level Trigger ROSs where the data is made available to the Level-2 trigger (LVL2).

#### 4.8.1 FLIC Functionality

Each core crate has two data links to the FLIC, one for each  $22.5^\circ$   $\phi$  region. The data received by the FLIC is buffered in a FIFO, pending processing. After processing, the data is then stored in an output FIFO, pending transfer to the ROSs. As for data processing in the FLIC, for the baseline design there is no full event assembling. The FLIC combines the two data streams from each core crate, formats it into ATLAS ROD fragments, and then sends this information directly to the ROSs as a single data stream. The High Level Trigger requests these fragments individually and assembles them as necessary.

However, we preserve the option to include additional functionality in the FLIC, such as primary vertex finding or beam spot determination. The feasibility of such options are being studied and the hardware implications investigated.

The event data across the entire FLIC is identified by the Level-1 ID (L1ID), along with other information which is part of the data record header sent by the fSSBs. Since each core crate may require a different time budget for processing a given event, the records from the individual core crates could arrive at the FLIC at different times but always less than a maximum time relative to the time of the event, and always time-ordered. A core crate sends a record even if there is no track information from its detector region for a given event so that the LVL2 knows that no event fragments have been lost.

The core crates operate in data-push mode. The FLIC can assert flow control back to the core crates if the buffer is nearly full due to either the FLIC or the ROSs. The FLIC also operates in data-push mode and the ROS can assert flow control back.

In addition the FLIC provides monitoring and diagnostic capability. The system has an interface to the TDAQ infrastructure to facilitate monitoring and control. For monitoring, spy buffers are implemented so that the overall performance of the system can be monitored. For diagnostics, it has the ability to process both fixed pattern data and user-defined data through the processing chain.

#### 4.8.2 FLIC Design

The FLIC is implemented in a single separate crate (shelf). Advanced Telecommunication Computing Architecture (ATCA) technology is chosen for this subsystem. A 6-slot shelf with full mesh backplane is used but the backplane could work in the dual-star mode as needed. Furthermore the 14-slot shelf could be adapted if there is a universal shelf recommendation from the ATLAS collaboration.

There are two Input Cards and two Output Cards. An Input Card + Output Card pair services four core crates. Each core crate uses two SFP links to send data to the front panel of the FLIC Input Card. The data from core crates are processed on the Input Card and then passed via the Zone 3 connector of the ATCA shelf to the Output Card, which is a Rear Transition Module (RTM). The Output Card holds eight SLINK connectors, two for each output channel. If a ROS can handle the 100 KHz request rate, one of the two SLINK connectors will be used to send data to the ROS. Otherwise the output data of each channel will be sent to two ROSs through the two SLINK connectors by a simple Round-Robin algorithm. Figure 80 illustrates the design concept.

ATCA technology allows the implementation of full event processing within the shelf as a future option. The architecture required for such a functionality is the dual-star backplane configuration, where every payload blade (Input Card) can communicate with two dedicated slots in the shelf. One slot is needed for control and monitoring, while the other slot could be used for future functions, such as advanced event processing or triggering. To facilitate an optional trigger processor in the future, the event data streams are copied to Zone 2, utilizing one of the two star connections in the backplane. A prototype Input Card with this implementation is shown in Fig. 81. Each of the mesh networks on the backplane provides four pairs of send and four pairs of receive connections to the dedicated processors in the shelf. The trigger connection would use one of these sets. The combined transmission rate of the four pairs is 10 Gb/s.

To facilitate the control functionality and the readout of monitor information, a processor blade uses the second of the two star connections on the Zone 2 backplane. This blade communicates with the TDAQ infrastructure over standard Ethernet. The blade is envisaged to be a commercial processor, and must be compliant with the ATCA Zone 2 connectivity for 10 Gb/s communication.

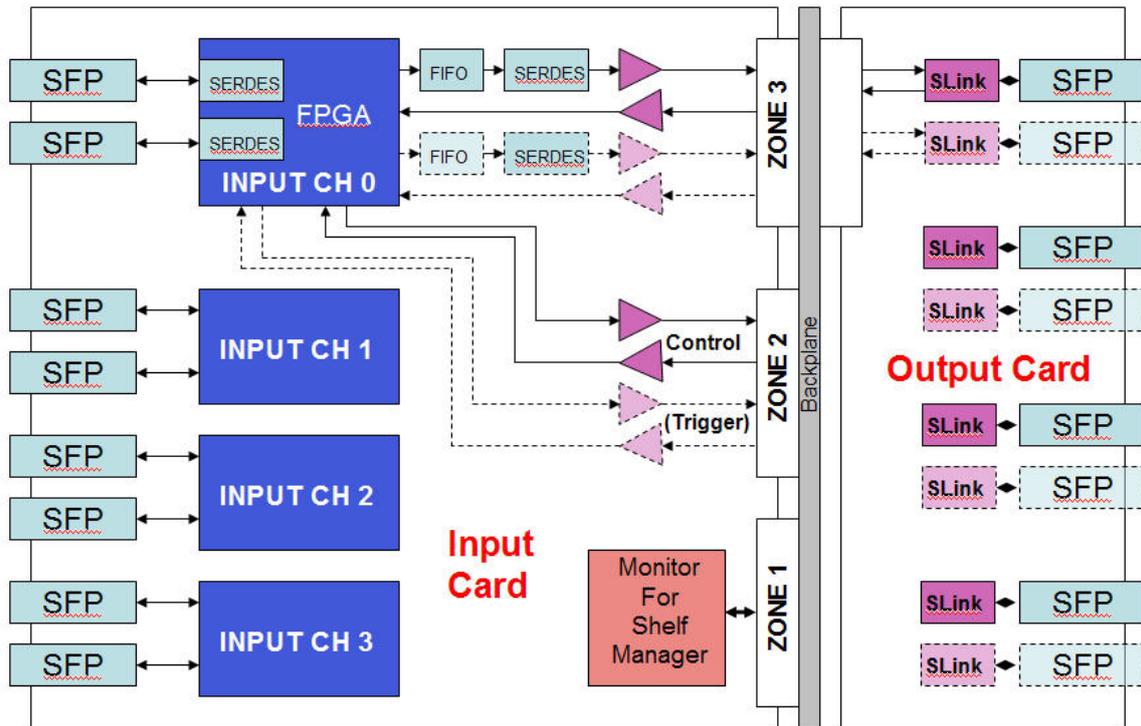


Figure 80: Design concept of the FLIC.

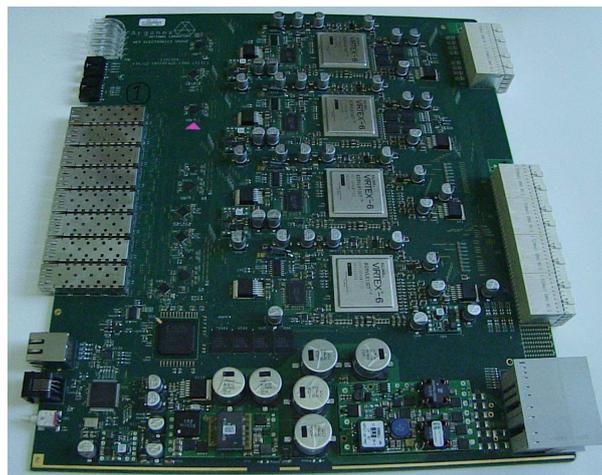


Figure 81: A prototype of the FLIC Input Card.

### 4.8.3 FLIC Data Transmission

At the LHC luminosity of  $3 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ , a typical ATLAS event averages  $\sim 300$  tracks with  $p_T > 1$  GeV, based on ATLAS simulation. The content of the input data to the FLIC includes the track parameters and the hit information of every track. The data size is expected to be  $\sim 100$  bytes per track, plus an overhead of  $\sim 40$  bytes per event for header and trailer. This leads to the average data rate  $\sim 3$  GB/s to the FLIC and  $\sim 3$  Gb/s from each core crate, assuming that the data is evenly distributed among the eight core crates. The FLIC does not increase the overall data size during the processing but adds only a few additional words of header and trailer, which is small compared to the size of the data received. Therefore the output data rate is similar to the input data rate.

The data records from the core crates to the FLIC are in 16-bit word format and each record corresponds to one event (one L1ID). An input record contains multiple track frames, with each containing information on a specific track in that event, including the helix parameters as well as hit information. There may be a variable number of frames in a record. A record begins with a record header that describes the event. The first word contains a fixed binary code to provide a way to identify the start of a record. A record is terminated by the record trailer which contains a series of words with fixed binary codes to identify the end of the record. The track frames contain the actual tracking information. A track frame begins with the track level information of twelve 16-bit words, which contain the track helix parameters. This is followed by a hit data block consisting of sixteen 16-bit words that enumerate the hit data ordered by detector layer.

The track frame from the core crates contains the tower number and the sector number fields. In order for the High Level Trigger to use FTK track information without reading another additional large geometry database, these two fields need to be translated into the detector module number by the FLIC. The module number table is loaded into the FLIC during configuration at system start-up, or whenever new geometry maps are generated. As the record is received from the core crates and being buffered into a FIFO, the tower number and sector number received in the track header of each track frame are used to provide the base address to the fast static RAM lookup table. As each word of the track frame is received from the core crate, a pipelined lookup of module numbers from the static RAM is merged with the incoming data to form a differently formatted data word that contains all the information from the core crate plus the associated module number.

The ROS expects the data to be in the standard ROD format, so the FTK ROD fragment is in 32-bit word format. The FLIC uses the information from fSSBs to create the ROD header but fixes the subdetector ID to that assigned to FTK, 0x7f. Following the standard ROD header, the first 32-bit word contains the number of tracks above low and high  $p_T$  thresholds. Then the track records follow, with each fixed length track record defining a single track. The last two long words following the track record(s) are the core crate status (received from the core crate) and the FLIC status (presently undefined). At the end is the standard ROD trailer.

## 4.9 Hardware infrastructure

FTK will occupy a total of 7 racks on the second floor of USA15: two for the Data Formatters, four for the core crates, one for the FLIC, plus one more as a spare if possible. The FTK ROSs will stay in the TDAQ rack area, where one rack is needed. A standard ATLAS rack has 10 kW electrical power and 10 kW cooling capacity though higher electrical power and cooling capacity are possible with modifications. This limits to two the number of 14-slot ATCA shelves or 21-slot 9U VME crates per rack.

The racks along with the power distribution and cooling will be installed by ATLAS OPM during LS1. The fibers connecting the PIX/SCT RODs (FTK ROSs) will be laid out from the ROD racks (to the ROS racks). The number of fibers will match the number of existing PIX/SCT RODs, plus the IBL RODs and the additional SCT RODs to be installed. The control network connection and DCS connection will

be routed into the racks via either a switch in the rack or from an adjacent rack. For those racks holding core crates, a file server will be installed for each rack to store the pattern bank and fitting constants associated with the crates in that rack.

Commercial ATCA shelves will be used for the DF and the FLIC, but the choices will be based on the ATLAS recommendation so that central service support is secured for the long term. A series of modifications of the conventional VME crate are needed for the core crate to accommodate the voltage requirements of the Associative Memory Board and the AUX card. The modification exercise is being performed within both a CDF crate and a standard Wiener crate. The intermediate and final choice will be based on the technical complexity and being ATLAS compliant.

#### **4.9.1 Power consumption**

For the ATCA system, the maximum power dissipation per blade is ~300 W, and per RTM ~30 W. With the extra power needed for the shelf manager and the fan trays, the maximum power consumption per shelf could take 5.1 kW or higher, but the DF system and the FLIC system will work well below that limit. Each shelf has its own power supply which will satisfy the ATLAS requirements on efficiency, fluctuation and load protection. The power supply sits at the bottom of the shelf and will not block access to the RTMs.

The power consumption of the core crates is not yet fully determined. In the baseline design, a fully loaded core crate will host 16 Associative Memory boards with up to 2048 AM chips, 16 AUX cards and 4 Second Stage Boards. The test results of the prototype AMChip04 chip extrapolated to 64K patterns would need ~1.8 W for the core. The serial I/O links for an AM board would require more than 140 W. Thus one core crate would need more than 7 kW of power with such a design. R&D is underway to reduce the chip power consumption as described in section 4.6.5. This will result in lower overall power consumption.

#### **4.9.2 Cooling**

Vertical airflow is to be used for cooling. For an ATCA blade 40-50 cubic feet per minute airflow is enough and this corresponds to an overall 910-1130 m<sup>3</sup>/hour for the 14-slot shelf. This is compatible with the 1180 m<sup>3</sup>/hour capability of the turbine of the ATLAS racks. The fan trays of the two shelves in the same rack will be set to deliver the same airflow or to run at full speed.

Cooling tests are being performed for the core crate with the default ATLAS rack cooling configuration. If the airflow is not adequate to cool the fully loaded AM board, the rack cooling infrastructure would be upgraded.

#### **4.9.3 Detector Control System (DCS)**

FTK will implement DCS components in the existing ATLAS DCS [34] to supervise and monitor the hardware and its infrastructure. The ATLAS DCS applications are implemented in the framework of a commercial Supervisory Controls And Data Acquisition (SCADA) product, ProzessVisualisierung and Steuerungssystem (PVSS).

For each ATCA shelf the shelf manager performs the control functions of the individual blades and Field Replaceable Units (FRU, such as the fan-tray and the power supply), such as power on/off, fan speed adjustment, etc., and it also monitors the states of the blades and FRU, such as the fan speed, the temperature, etc. The shelf manager uses the standard IPMI protocol, and some open source software frameworks (IPMItool, etc) are available. The ATLAS collaboration is investigating the methods (such as using openHPI) to communicate the IPMI platform to the PVSS. The FTK DCS components for the Data Formatters and the FLIC will adopt the ATLAS recommendation.

The industrial standard VME crates used for core crates can accept operation commands such as switch on/off and reset, and can provide monitoring of the general status such as power on/off, error, etc. of the cooling fans, and of temperature at different points. The FTK DCS components for the core crates will be organized under the existing TDAQ DCS, and a Local Control Station (LCS) PC with CANbus interface card will connect via CANbus to the core crates. The industry standard OPC protocol will be used as the software protocol for communications. The needed components in PVSS will be implemented, such as FTK panel in the Finite State Machine (FSM), configuration database, alarm system, etc.

#### 4.10 Timing

A simulation tool for calculating FTK latency was developed for tuning the system architecture and parameters and to ensure that FTK can handle a 100 kHz level-1 trigger rate at high luminosity. The system is divided into functional blocks: Data Formatter, Data Organizer write mode (receiving hits from the DF and sending Super-Strips to the Associative Memory board), Associative Memory Board, Data Organizer read mode (receiving roads from the Associative Memory board and sending roads and hits to the Track Fitter), Track Fitter, Hit Warrior, Second Stage Board. The focus so far has been on the most time consuming steps, from Data Organizer write mode through Track Fitter. The Data Formatter should add little to the overall latency since each cluster found is immediately sent to the Data Organizer. Thus the Data Formatter and Data Organizer execution times almost completely overlap. Similarly, the Hit Warrior has a relatively short latency for each track that enters before the track is either sent to the output or discarded.

For each functional block, the time of the first and last words into and out of the block are calculated. Since each core crate operates independently, the FTK event execution time ends when the last word exits the busiest crate for that event. The execution time for a block depends on the number of input words, the processing time per word, and the number of output words. We estimate the processing time per word for each block type from the architecture and our experience with prototypes (see Table 9). The numbers of input and output words for each block come from FTKSim events. The results reported here use  $Z \rightarrow \mu\mu$  events with 69 pile-up. The solid angle covered by a core crate is divided into  $\eta - \phi$  towers. Each tower has its own unique set of Associative Memory patterns. However the hits from the Data Formatters often have to be sent to more than one tower due to curvature in  $\phi$  for  $\geq 1$  GeV tracks, the length in  $z$  of the LHC luminous region, and multiple scattering. The amount of hit duplication was determined from FTKSim and used by the timing simulation. These studies assume very large input FIFO buffers. During the firmware design phase, we will do queuing studies to determine the needed depth of each input buffer.

Function	Data word processing rate (Number of parallelization / tower)	Latency for a data word
DF	-	1000 ns
DO write	200 MHz (1)	40 ns
AM	200 MHz (8)	200 ns
DO read	200 MHz (8)	500 ns
TF	1 GHz (8)	300 ns
SSB	1 GHz (0.5)	300 ns

Table 9: The properties of the functional blocks currently implemented in the FTK timing simulation. A dash indicates that the speed is high enough to have negligible effect on the overall timing.

### 4.10.1 FTK timing results at 69 pile-up

Figure 82 shows example timing charts for individual  $Z \rightarrow \mu\mu$  events with 69 pileup. In each case, the timing is shown for the core crate that finishes processing the event last. For the event on the left, the previous event had already completed processing before the new one arrived. For the event on the right, FTK had not completed processing the previous event and thus waits to start the new event.

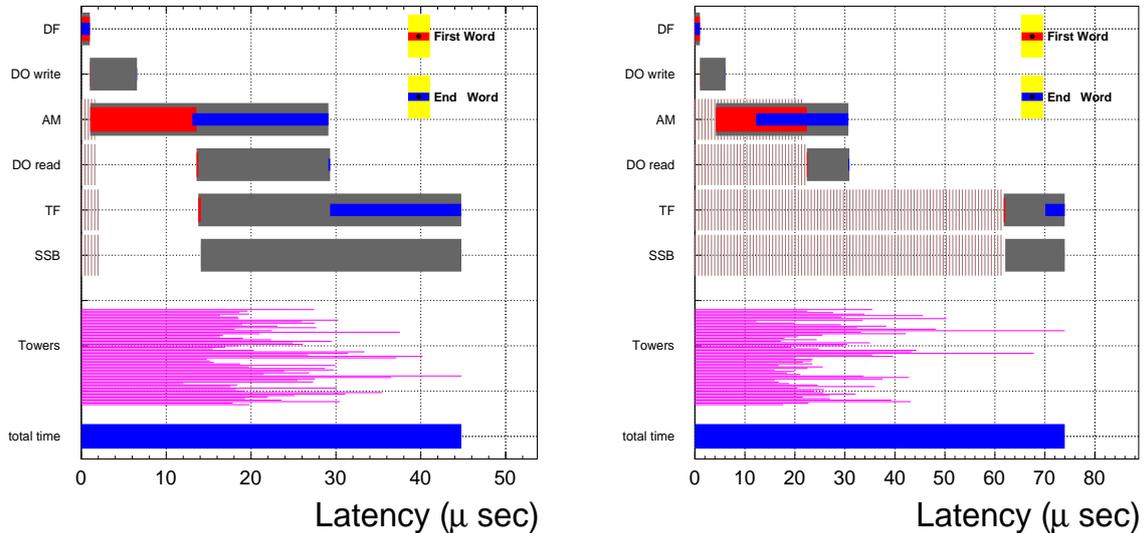


Figure 82: FTK latency for two events at 69 pile-up. In both cases, the timing of the functional blocks is given for the core crate (region) that takes the most time. The time for each of the 64 regions is shown below that, with the total latency shown in the bottom bar. For the event on the left, FTK had completed processing the previous event before the current event arrived. For the event on the right, FTK was still processing the previous event (shown as the light shading in the TF bar).

To get higher statistics on event execution times and see whether FTK can handle the 100 kHz level-1 trigger rate, we analyzed a larger sample of events. If this rate were too large for our system, we would see the event latency (from input hits available to event completion) steadily increase as FTK falls behind, working on a stack of previous events before getting to the current one. This does not happen, as seen in Fig. 83. Some events take longer than others to do global tracking, but after such an event the latency quickly returns to the typical range. The bottom line is FTK operates well for a 100 kHz level-1 trigger rate at 69 pile-up events.

To see the worst-case performance, we study a  $t\bar{t}$  sample with 69 pile-up which has more high- $p_T$  jet activity than for the average event selected by the level-1 trigger. This would produce more towers with high silicon hit occupancy and thus longer latency. Figure 84 shows that while the typical latency does increase, the system still operates well.

## 5 Simulation

To evaluate FTK performance and test its algorithms, we developed a software emulation of FTK, which is challenging since it has to emulate a massively parallel hardware system (see section 2). It is a functional emulation of the hardware, not a bit-by-bit simulation, which reproduces in detail the logic of each stage of FTK processing and as such has been invaluable in designing the system and maximizing its

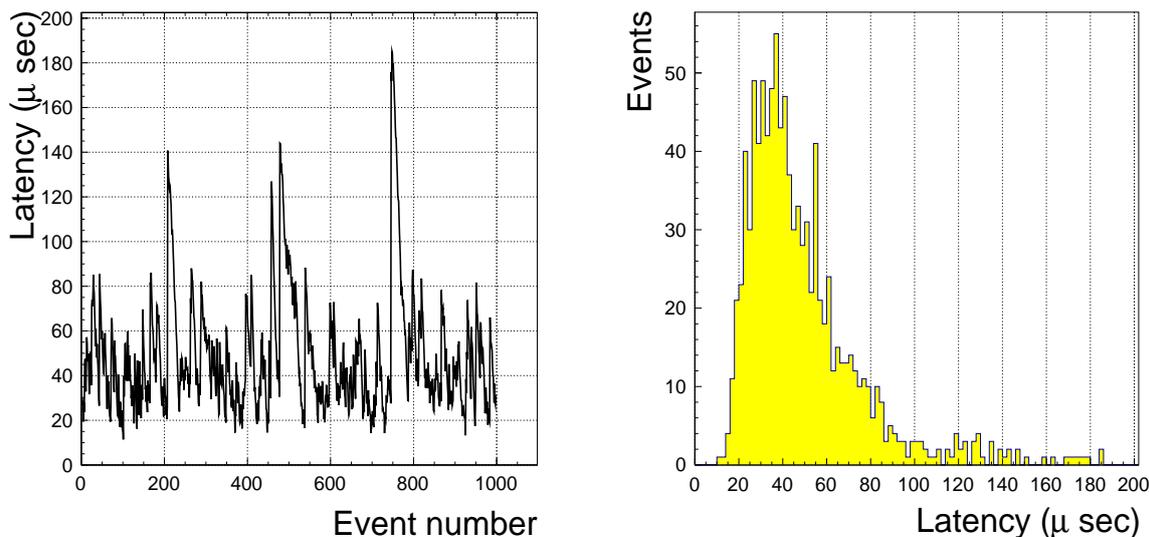


Figure 83: FTK latency for  $Z \rightarrow \mu\mu$  events with 69 pileup. For each event, the execution time starts when the event is available ( $10 \mu\text{s}$  after the previous event, corresponding to a 100 kHz level-1 trigger rate) and ends when the FTK has completed analyzing that event. The event-by-event latency is shown on the left, and a histogram of the latency is seen on the right.

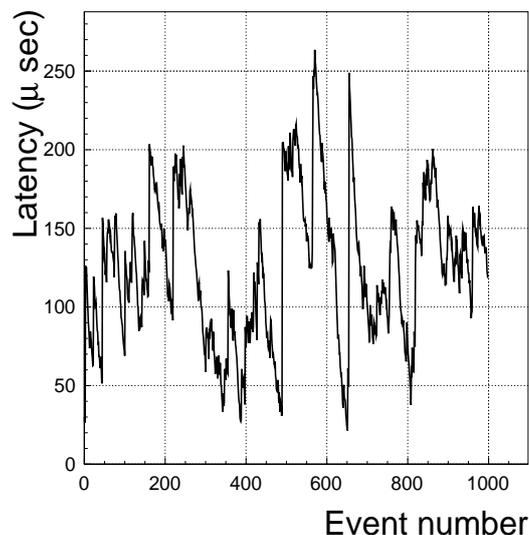


Figure 84: FTK event-by-event latency for  $t\bar{t}$  events with 69 pileup. For each event, the execution time starts when the event is available and ends when the FTK has completed analyzing that event.

capabilities. Some differences with respect the existing firmware prototypes exist for practical reasons: data types are more relaxed and transmission protocols are not emulated. These differences have no measurable impact on the results and can be removed once final versions of the FTK firmware exist.

The existing simulation code consists of a set of elements integrated into the ATLAS software framework, Athena, so that the simulation can be run within that environment. Additional tools for producing

the training samples needed by the simulation also exist. In the future, the simulation will be used for the generation of large Monte Carlo samples and will get completely integrated in the production framework. Integration of the FTK simulation into the overall ATLAS simulation process is discussed in Sec. 5.3.

## 5.1 Simulation description

FTK simulation can be performed within the Athena framework or as a standalone program to provide maximum flexibility. The simulation consists of steps that reproduce the logical functions of the hardware and then convert the results into trigger or other data formats to allow emulation of the trigger selection. In the description of the simulation steps below, the small differences relative to the hardware are noted.

The Data Formatter simulation is divided into two parts. The hardware clustering algorithm will be moved into this step once a number of possible algorithms have been tested. Hits are sent to the proper  $\eta - \phi$  towers, but with a flexible scheme based on the silicon detector module position rather than its read-out link. This doesn't reproduce the data sharing among Data Formatter boards, which is studied using a separate program, but it does emulate the data flow after that.

The functions of the FTK Processor Units (AM board and AUX card) are divided into two logic steps: road finding and track fitting.

The road finding routines simulate the Associative Memory and the Write mode of the AUX card including the calculation of the Super-Strip address for each hit. At present, pixel clustering compatible with the FTK\_IM algorithm is done here. In calculating the Super-Strip ID, there are small differences relative to the hardware due to the number of bits per word being less important in simulation. A technical difference occurs in the Super-Strip matching, where a linear scan of the pattern bank is avoided and the variable resolution feature is implemented in a two step procedure. The pattern bank emulation is very flexible, allowing the working point to be easily changed to find the one that is optimal.

Emulation of track fitting, corresponding to the AUX Read mode and Track Fitter, is done after a block of events is completely emulated in the road finder. Small differences between simulation and the hardware exist in the work flow and in the use of floating point arithmetic. Here again tests have shown that these result in insignificant changes in the output. Both the first stage (8-layer) and second stage (12-layer) calculations are done. The choice of layers used in the two stages and other parameters can be changed at runtime to allow optimization studies to be easily carried out. The hits in each of the 4 layers added in the second-stage fit are found by extrapolating the first-stage track into that layer using a linear calculation, as in the hardware, to get the expected hit location. Hits are searched for in a region 3 Super-Strips wide centered on that position.

An early test of the agreement between the software emulation and hardware was done with the FTK vertical slice as described in section 7 where excellent agreement is reported.

## 5.2 Simulation setup and configuration files

To reproduce the hardware setup, FTK simulation needs special configuration files that describe internal parameters and running conditions. There are two types: internal configuration files, describing the setup of the system and mapping the ATLAS Inner Detector to FTK, and files produced using a training procedure. The latter is done using a standalone package that performs these intensive tasks using the Grid resources.

### 5.2.1 Internal configuration files

The basic FTK configuration requires a small set of files that map the silicon layers to the FTK processor.

**Plane map:** It maps the physical ATLAS silicon layers to the FTK logical layers. Because we have a system that fits the tracks in two stages, we have a plane map for each stage. This file is a table in ASCII format, with an entry per ATLAS physical layer.

**Tower map:** It defines the boundaries of each FTK  $\eta$ - $\phi$  tower for the simulation, reflecting the hardware boundaries. The file contains a set of tables, one per tower, each having one entry per physical silicon layer that specifies the minimum and maximum module ID along  $\phi$  and  $\eta$ .

**Super-Strip map:** It specifies the size of the Super-Strips used in the construction of the pattern banks. Each Super-Strip is defined as the number of adjacent channels of the SCT sensors; for the pixel sensors the Super-Strip is a rectangle and the numbers of channels in each direction are given. The Super-Strip map is a table with one entry for each physical silicon layer so that a different Super-Strip size can be set in each layer. Usually the Super-Strip size is the same in all layers of the same subdetector.

Because of the variable resolution feature in the AM chip, we need two Super-Strip maps to completely define a pattern bank. They specify the minimum and maximum size of the Super-Strips, with the maximum Super-Strip size in a layer being a power of 2 larger than the minimum:  $\Delta SS_{max} = \Delta SS_{min} \times 2^{N_{DC}}$ , where  $N_{DC}$  is the maximum number of variable resolution Don't-Care bits for that layer.

## 5.2.2 Production of sectors and constants

The generation of a FTK configuration starts by identifying the sectors, each consisting of one silicon module in each logical layer. A sector is an important unit in FTK because it is the region in which the linear fit with one set of constants is performed.

The list of sectors is determined with a training procedure that uses a large number of single muon events, typically 300 million, to cover all possible combinations of modules compatible with tracks coming from a region around the nominal beam spot position. Negative and positive muons are generated with a flat distribution in the following perigee parameter phase space:  $d_0 \in [-2.2, 2.2]$  mm,  $z_0 \in [-120, 120]$  mm,  $1/p_T \in [-0.8, 0.8]$  GeV<sup>-1</sup>,  $|\eta| < 3$ , and  $\phi \in [-\pi, \pi]$ .

The training events are produced using the full ATLAS simulation in a release with the appropriate detector geometry. Default options are used except for the digitization, where we decrease the amount of noise in the detector. In order to use very clean events and avoid contribution from noise, the hits are filtered, suppressing all the hits coming from secondaries and noise. Events are excluded that do not have hits in all the logical layers defined as active in the plane-map. If a track has clusters in two overlapping modules in the same logical layer, the hits in the module with the greater  $\phi$  and  $\eta$  index are removed. This does not induce biases, but it reduces the number of sectors that are identical except for one module, a category that produces an increased number of duplicate tracks without improving tracking efficiency or performance. Events with hits in non-adjacent modules or with more than two hits in a logic layer are rejected to reduce the contribution from noise. This produces a sequence of one module per layer for each single muon.

After processing all the training events, the complete list of sectors for each of the  $\eta$  -  $\phi$  towers is made and sorted by the frequency of tracks in each sector in decreasing order. The frequency of a sector is an important metric and is often referred to as *coverage*. The coverage is related to the probability that a sector is hit by a track, and thus it is proportional to its volume in the phase space of the track parameters. All the sectors found by this procedure are unique, no duplicates are allowed in the same tower or in other towers.

For each sector, the linearized fit constants are produced. The linear fit procedure expresses the track

parameters ( $\tilde{p}_i$ ) as:

$$\tilde{p}_i = \sum_{l=1}^N C_{il} x_l + q_i \quad (4)$$

where  $C_{il}$  and  $q_i$  are constants valid in a sector and  $x_l$  are the  $N$  hit coordinates. The coordinates are expressed as the local cluster centroid position within each module: two coordinates for pixel hits and one for SCT hits. A sector has been shown to be a small enough detector region for the linear approximation to give good resolution. We thus need to evaluate the constants in (4) for each sector.

To evaluate the fit constants we want the parameters that minimize the distance between the linear parameter ( $\tilde{p}_i$ ) and the true parameter value ( $p_i$ ):

$$\min \left( \langle (\tilde{p}_i - p_i)^2 \rangle \right) \quad \forall i = 1, \dots, 5 \quad (5)$$

where  $i$  is a counter over the 5 helix parameters. The solution of (5) for the  $i$ th parameter can be extracted from:

$$\sum_{m=1}^N (\langle x_l x_m \rangle - \langle x_l \rangle \langle x_m \rangle) C_{mi} - \langle x_m p_i \rangle + \langle x_m \rangle \langle p_i \rangle = 0 \quad \forall l = 1, \dots, N \quad (6)$$

$$\sum_l C_{il} \langle x_l \rangle + q_i = \langle p_i \rangle \quad (7)$$

where the averages are over the training muons. In (6) it is possible to identify the covariance matrix between the coordinates:

$$\{V\}_{lm} = (\langle x_l x_m \rangle - \langle x_l \rangle \langle x_m \rangle) \quad (8)$$

The solution of the linear system in (6) is obtained using the inverse of the covariance matrix  $V^{-1}$ :

$$C_{il} = \sum_m V_{lm}^{-1} (\langle x_m p_i \rangle - \langle x_m \rangle \langle p_i \rangle) \quad \forall i, l \quad (9)$$

The constants  $q_i$  are then obtained by inserting the  $C_{il}$  coefficients from (9) into (7).

From the covariance matrix in (8) we can evaluate a quality parameter that in the limit of the validity of the linear approximation is distributed as a  $\chi^2$  and can be calculated as:

$$\chi^2 = \sum_{i,j=0}^N (x_i - \langle x_i \rangle) V_{ij}^{-1} (x_j - \langle x_j \rangle) \quad (10)$$

Based on a principal component analysis [30], decomposing (8) into its eigenvalues and eigenvectors produces 5 large eigenvalues that don't contribute to  $\chi^2$ , while the others are very small and make real contributions. We can reduce (10) to the form:

$$\hat{\chi}^2 = \sum_{i=1}^{N-5} \left( \sum_{j=1}^N A_{ij} x_j + k_i \right)^2 \quad (11)$$

where  $A_{ij} = u_{ij} / \sqrt{e_i}$ , with  $u_{ij}$  the component  $j$  of the eigenvector  $i$  and  $e_i$  its eigenvalue, while  $k_i = \sum_k A_{ik} \langle x_k \rangle$ .

The calculations are repeated for each sector. For sectors with low coverage, the covariance matrix cannot be inverted, and so the sector is rejected. Typically sectors in which at least 5 training tracks are found produce good constants.

The sequence of steps presented here, finding the sectors and evaluating the fit constants, is done twice, once for the first-stage fit (8 layers) and again for the second-stage fit (12 layers).

### 5.2.3 Production of patterns

The patterns can also be produced from a sample of training muons. However the number of training tracks needed to produce the patterns with sufficient coverage is extremely large and would take too much time using full ATLAS simulation. To estimate the size of the training sample we start with the 1 billion patterns that will exist in the hardware. As will be described shortly, we begin by generating a bank with narrower patterns, which means generating 10 or 20 times more patterns, plus a safety margin of about a factor 2 to minimize the impact of unusual trajectories in the training sample. Finally the average number of tracks in a pattern is  $\sim 10$ . The net result is that  $\sim 400$  billion tracks are needed to fully train a pattern bank. This huge number makes it impractical to use single muon events generated with the full ATLAS simulation.

Instead we found a way to significantly speed up the process by inverting the track-fitting linear approximation. We generate 5 random numbers uniformly distributed in the track parameters ( $\hat{p}_l$ ), as in the production of the sectors, plus  $N - 5$  random numbers Gaussianly distributed, which represent the constraints ( $\hat{\chi}_m$ ), and then solve the following linear equations:

$$\begin{cases} \sum_j C_{lj}x_j + q_l = \hat{p}_l & \forall l = 1, \dots, 5 \\ \sum_j A_{mj}x_j + k_m = \hat{\chi}_m & \forall m = 1, \dots, N - 5 \end{cases} \quad (12)$$

where  $C_{lj}$ ,  $A_{mj}$ ,  $q_l$ , and  $k_m$  are the same as in equations (4) and (11) for a given sector. The coordinates  $x_j$  can then be extracted and transformed into a sequence of Super-Strips. This provides efficient generation of all the tracks needed to produce a pattern bank. A normal bank production can be done by submitting about 10k independent jobs, which do not need to exchange information in a way similar to the simulation jobs described in section 5.1.

As was done in the case of sector creation, the patterns are ordered according to track frequency in descending order. This list of patterns is produced at high resolution, far exceeding the capacity of the hardware. Significant reduction of the bank size can be achieved by grouping the patterns and using the variable resolution feature of the AM chip using the Don't-Care bits, and finally by removing the patterns with the least coverage, keeping the number of patterns that fit in the hardware as discussed in section 4.1.2. The process of grouping high resolution patterns into lower resolution ones and then enabling Don't-Care bits when appropriate is shown in Fig. 85.

The first step is clustering the existing patterns, which were created using the high resolution Super-Strip size  $\Delta SS_{\text{high-res}}$ , into the widest patterns that would be used in the AMchips, with a Super-Strip size  $\Delta SS_{\text{AM}} = \Delta SS_{\text{high-res}} \times 2^{N_{\text{DC}}}$ , where  $N_{\text{DC}}$  is the number of Don't-Care bits to be used. When an AM pattern is generated, all the compatible high resolution patterns are kept and the connections between high resolution and AM patterns are saved. The high resolution pattern's relative position within the AM pattern is important for setting the DC-bits. Figure 85 shows how the high resolution patterns are grouped into the wider Super-Strip patterns and how the variable resolution finally shapes the patterns. Since the DC implementation will be done when the pattern bank is initially produced, the bank loaded into the simulation will be equivalent to the one loaded into the hardware.

### 5.2.4 Configuration file distribution for TDR simulation

The many static configuration files are very small and are distributed with the simulation code bundle. The files obtained by the training procedure, the fit constants and pattern bank files, are much larger and will exist in different versions that depend on the detector and beam conditions. These large files are stored as Grid data sets and distributed to the computing tiers using the standard data movement tools developed by ATLAS.

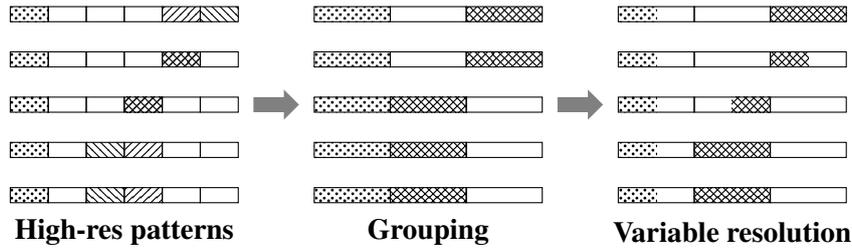


Figure 85: The figure shows the high-resolution bank on the left and the AM bank after the patterns are grouped at lower resolution, prior to using Don't-Care bits, at the center. The cross-hatched Super-Strips are those shared by two high resolution patterns. The right figure shows how the DC bits change the shape of the AM patterns: the low resolution pattern produced by a single high resolution one is restored to high resolution, while the pattern produced from the merger of two high resolution patterns has some layers with high resolution and some with low resolution to provide a narrower pattern where possible.

### 5.3 Evolution and integration with the ATLAS production system

Sec. 5.1 describes the simulation of the FTK system that we developed and used for the design of the hardware. The major technical challenges were the large execution time needed to simulate a massively parallel system of specialized hardware and the amount of memory needed to store the  $\sim 10^9$  patterns that will be in the Phase-I Associative Memory pattern bank. For the latter, each pattern in the simulation is represented as a sequence of one integer per detector layer plus one to store the sector ID which is required for the linear fit. For the 8 layers used in pattern recognition and 32-bit precision, almost 36 GB are needed to store the whole pattern bank. To provide an efficient matching scheme so that a linear scan of the bank doesn't have to be done for each silicon hit, we index the patterns by Super-Strip in each layer at the price of almost doubling the pattern bank size. This makes the memory per pattern 68 bytes, with a total memory for the bank of almost 70 GB. Fig. 86 shows the total memory required as a function of the number of FTK patterns.

In this section, we consider how the resource requirements could evolve in the near future and the resulting impact of the FTK simulation on the ATLAS production system [35]. The long-term goal is a manageable simulation for the full FTK system that will operate during Phase-I, with 1G patterns distributed over 128 Processing Units. We also present plans for the evolution of FTK after LS1 with 16 Processing Units, corresponding to a total of 128M patterns.

The main computing limitations affecting the structure of the FTK simulation are 2 GB of RAM and 3 GB of virtual memory per core. Other constraints such as working area size are less important. Simulation is done with the same  $64 \eta - \phi$  towers as in the hardware. Given the amount of memory needed (Fig. 86), all the towers cannot be processed in one job. The hits belonging to different towers are processed by parallel jobs with an organization similar to that of the hardware. Typically the memory for one job slot would allow 1 or 2 towers to be processed, meaning the we would process an event in 64 or 32 jobs. The current baseline design is sketched in Fig 87. It requires the digitization files to be staged in storage elements for reading by the many independent FTK simulation jobs that are run in parallel on different machines. The files and jobs needed are shown in Table 10 for the current limits on the available memory per job. With some memory optimization like mapping the Super-Strips using 18 bits, the number of jobs might be reduced to 16 or less. The actual number will depend on code optimization, the number of Processing Units installed in FTK, and the available RAM per job slot. For simulation of the partial FTK system that will be installed after LS1, the number of jobs will be greatly reduced since they scale linearly with the size of the AM pattern bank. At that time, it may be possible to run the simulation in a single job.

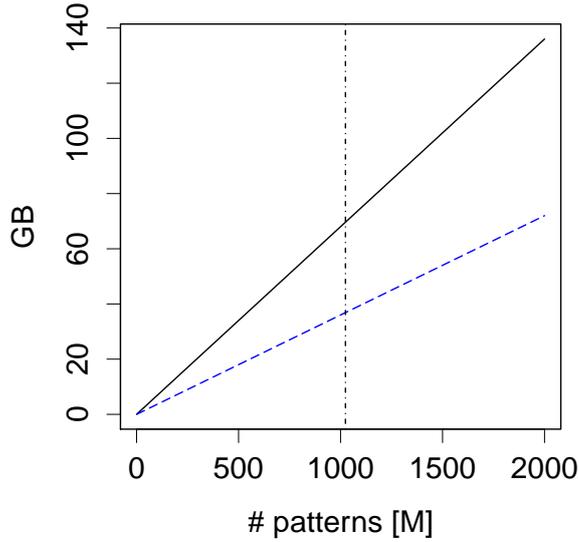


Figure 86: The memory required to emulate the AM boards. The dashed line shows the memory needed for the pattern bank as function of the number of patterns. The solid line is the total memory needed to store the pattern bank and the layer-by-layer Super-Strip indexing structure. The vertical line shows the number of patterns that can be stored in the hardware.

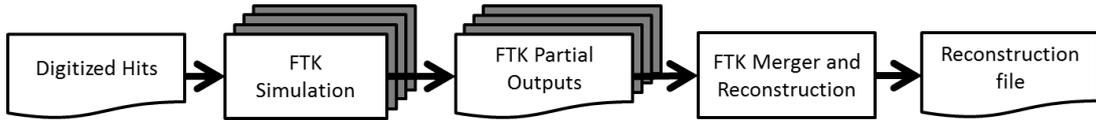


Figure 87: The current implementation work flow for running the FTK simulation in the production system. Input digitization files are processed by several jobs, producing partial output files that are merged prior to running the reconstruction.

Step	Filetype	Size	N Files	N Jobs	N Jobs
			full FTK	full FTK	FTK 2015
Simulation	Digitization (pre-FTK)	500 MB	1 – 10	64/N	4
	Pattern bank (no compression)	570 MB	N		
	Fit constants	12 MB	N		
Merging	FTK digitization fragments	5 MB	1 – 10	1	1
Reconstruction	Full reconstruction output	500 MB	1 – 10		1

Table 10: The numbers of files and jobs for full simulation with the most pessimistic assumption of 1 FTK tower per job. The number of jobs will depend on how many of the 64 towers can be grouped into a single job based on the memory available per job. The parameter N defines how many towers are simulated in a single job. The file sizes are given for 50-event files.

The current implementation of the FTK simulation work flow is shown in Fig. 87. Depending on the evolution of multiprocessing and multi-threading in ATLAS software, the work flow could evolve to that shown in Fig. 88. In order to optimize the distribution of files for the scheme described in Fig. 87, the FTK simulation jobs and the “FTK merger Reconstruction” job would have to be executed at the same grid site. The new scheduling with job merging in ProdSys2 [36] should make managing FTK production

easier in the future. The FTK simulation output size for Phase-I conditions is 100 kB per event which doesn't have a large impact on the total simulation event size.



Figure 88: This sketch shows a possible future scheme if ATLAS jobs are performed using multiple cores in a single machine. The digitization jobs could use  $N$  cores in a multi-thread or multi-process submission, the FTK simulation could then be performed in a single multi-thread job, and the reconstruction could be performed in the same job or in an additional step.

The execution time is shown in Table 11 for the full set of 64 partial simulation jobs in the existing production system. Times are measured using high-pileup events ( $\mu = 69$ ).

We have identified a number of ways to further reduce FTK simulation processing time. The ATLAS software development team expects the times shown in Table 11 to improve by a factor  $\sim 2.5$  by 2015 due to the use of newer compilers and of code optimization to make better use of the vectorization available in modern CPUs. The FTK simulation times in the table will have a similar or larger improvement when we rewrite our algorithms explicitly for vectorization since our simple memory structures and matrix multiplication are ideal for vectorizing. Moreover we estimate processing time reduction in the range of 30% to 60% by optimizing memory use and the algorithm work flow.

Part	Time/Evt (HS06)	Time/Group	Tot time
FTK Init		$\sim 1$ h	$\sim 85$ h
FTK Simulation	191 sec (1380)	26 h	2100 h
Reconstruction	160 sec (1200)		1750 h

Table 11: The table shows the execution time for the current version that is run on the ATLAS production system. The sample is 40k 69-pileup  $H \rightarrow \tau\tau$  events, with FTK simulation run on groups of 2500 events each. Initialization is performed for each group. Thus the contribution of initialization to the total time per event will drop as larger numbers of events per group are run.

Table 12 gives the extrapolated FTK processing time per event, including the improvements discussed above, to be compared with 4000 HS06 that is expected for ATLAS simulation and reconstruction in 2015 [37]. The FTK CPU time in 2015, when the pattern bank will be only 1/8 of that in the final system, corresponds to about a 1% of the total ATLAS processing time per event at that time. The extrapolated FTK CPU time for the full FTK system and Phase-I luminosity would be about 8% of the processing time per event if the overall ATLAS processing time did not increase from 2015.

Simulation + Reconstruction (HS06) for 2015	4060
FTK Simulation (HS06) for 2015	22 – 60
FTK Simulation (HS06) for 2018-2021	180 – 500

Table 12: ATLAS simulation plus reconstruction CPU time per event extrapolated to 2015 with expected improvements [37] compared with the FTK simulation time assuming improvements in the algorithms and better compilers. The FTK 2015 number corresponds to 16 FTK Processor Units; the Phase-I number assumes the full 128 Processor Units. The range for FTK simulation timing represents the range of improvement from code optimization.

In addition to full simulation, there are other methods to emulate the FTK. Simulation could be executed only for those events that require FTK in the trigger chain as determined by the level-1 trigger

bits. Events whose level-1 triggers require only regional use of FTK tracks could exploit regional FTK simulation. In some cases, a fast simulation could be used in which the parameterized effect of the AM-based pattern recognition is applied to the truth information and then FTK track fitting is performed. In this method, all the complications related to memory use are avoided. The quality of the tracks won't change, although fake tracks will not be present, a feature that is shared by other fast Monte Carlo techniques.

In the context of the ever increasing requirements for high statistics simulation samples, a fast simulation and digitization/resonstruction framework is being developed in ATLAS (ISF) [38], and there should be opportunities to improve the speed of the FTK simulation in this context, albeit at the cost of some accuracy in the predictions which will have to be studied case by case, depending on the physics.

## 5.4 Use of new computing technology

We plan to take advantage of the evolution of the production system toward greater use of multi-thread and multi-process implementation. If, as predicted, 24-core machines are standard when FTK simulation is used for large Monte Carlo production ( $\sim 2016 - 2017$ ), a single machine with 3 – 4 GB of memory per core could allocate all the memory we need in a single node. We would naturally and efficiently use multiple cores due to the parallel tower structure of FTK simulation. Even if there were not as many cores or as much memory as we could use, we would still benefit from running fewer sequential jobs with more towers per core. The possible benefits are showed in Table 13. The total CPU time in this case is not reduced, but the real time is and the management of the jobs is simplified.

N Cores	Job Memory	N Towers/Job	N Steps	Time/Evt
24	72 GB	64	1	$\sim 8$ sec
8	24 GB	16	4	$\sim 33$ sec

Table 13: The table shows how the average execution time per event changes by the use of multiple cores. The real execution time assumes full parallelization and a balance among the cores.

Another possibility, if Grid computing moves in this direction, is the use of parallel coprocessors, like the Intel MIC or GPU. Their use could be very beneficial because our algorithms run in parallel and use very basic algebraic operations and a small number of data types, characteristics that are well suited to these coprocessors.

## 6 Online software

### 6.1 Configuration and control

The FTK configuration and control functions will be implemented in the ATLAS online framework.

#### 6.1.1 Configuration

As with all ATLAS DAQ components, the FTK configuration will be handled via the OKS database. The OKS database service will provide the description of the FTK hardware and software. These descriptions include the running configurations of all FTK applications during data taking and also provide the configuration parameters for them.

OKS is an object oriented database and its main persistent storage is provided by XML files. Schema files provide the blueprint of the data base structure (the classes), while the data files contain the instances of the defined classes. The data file will define the top FTK segment, nested sub-segments for the individual components from the DFs to the FLICs and ROSs and down to their sub-components, software

applications for each component and their relations with parameters, and input values for the configuration parameters. It is desirable to define FTK components as resources (in the run control terminology), at least at the board level if not down to the channel/link level so that they can be easily enabled/disabled in the partition.

The pattern bank and fitting constants will be needed to configure the core crates. Since the size of the pattern bank and fitting constants is large (a few hundred GBs), they will be broken into separated files (eight, with each corresponding to the detector coverage of a core crate) and stored on different local file servers located in the racks holding the core crates. During the FTK configuration these files will be loaded into the core crates in parallel. Preliminary tests show that the time needed to load the pattern bank and fitting constants is large (15 minutes) but the loading process is only necessary on rare occasions, such as a power cycle of the crate or a new version of the pattern bank. New pattern banks due to a change in the ATLAS geometry or a large change in the beam position should happen only a few times per year. The new banks will be prepared in advance, taking a few days, and then downloaded between fills. The more common change will occur when the list of dead silicon modules changes, requiring a change in the use of the Wild Card option. The job needed to update the pattern bank in this case will only take a few hours, and again the bank will be downloaded between fills. Furthermore the loading process will be improved and optimized as the hardware progresses. Since all the boards in a crate configure in parallel, the concurrent access to the local file server may require crate level proxies but the proxy implementation already exists in the current TDAQ infrastructure.

### 6.1.2 Control

The existing ATLAS Run Control system [7] is built up from a hierarchical set of controllers. The FTK controller applications will be developed and organized in the same structure, i.e., a hierarchical tree corresponding to the tree of the FTK segment. These controllers will take corresponding actions to state transitions during data taking. They will (1) interact with the OKS database to handle the configuration information of the elements, (2) be responsible for reacting to commands passed from above and for passing commands down, as well as communicating with other applications in the partition, (3) carry out message reporting through the Message Transporting System (MTS), (4) interact with the Information Service (IS) [39] to publish operational statistics and status information, and with Online Histogramming Service (OH) to publish monitoring histograms, and (5) interact with the Diagnostics and Verification System (DVS) [40] to check the status of the system components and report and possibly diagnose problems detected.

The controller(s) for the DFs and the FLIC will run on the processor blade(s) located either in the shelf (crate) or in a separate rack with Ethernet connection to the shelf. The controllers for the core crate components will run on the single board computer (SBC) in the VME crate. Since there are several components in a given core crate including AM boards, AUX cards and SSBs, two options are being considered, either to develop individual controllers for these board types or to develop a combined controller with distinguishable methods. The final strategy will minimize the total FTK configuration time by maximizing the parallelism among the components, while not overloading the VME bus that feeds all the components in the crate.

In addition, error handling and reporting must be built into these control applications and they must incorporate on-the-fly error handling as well as mechanisms to allow for reconfiguring the system and removing faulty components with minimal data-taking downtime. To perform this, all controllers will be designed to react to the formatted messages that are also reported to the MTS. If a FTK component malfunctions and propagates BUSY back, based on the predefined rules the online expert system will mask the component or the associated FTK slice out while data taking continues (stopless removal). If a FTK component loses synchronization to the reference signal, the online expert system will allow this system to recover before resuming data taking. If a component that had been previously disabled in

the same run is fixed, the online expert system will re-include it into the ongoing data taking (stopless recovery). The exact granularity of a removable/recoverable unit/slice (a tower, a board, a channel, etc.) will be determined based on the impact on the physics performance and the operational cost. These procedures will require the online expert system to temporarily hold the trigger and synchronize the related part with the rest of ATLAS. During commissioning, flow control from FTK to the PIX/SCT RODs will be disabled, thus the stopless removal will be initiated based on the error statistics seen on the FTK ROSSs. During normal operation, it is necessary to hold the trigger while an action is being taken.

## 6.2 Monitoring

FTK will use the standard ATLAS monitoring framework to monitor the hardware status and the physics data integrity. The infrastructure and environment monitoring (crate power, cooling, etc) will be done through the ATLAS DCS system (see section 4.9.3). For online monitoring, FTK hardware-specific information and physics event data will be reported to the monitoring services, gathered and processed by the monitoring facilities across all instances, then presented by the monitoring presenters. Standard HLT monitoring will be used to monitor distributions of raw track information and HLT objects reconstructed from FTK tracks. Furthermore the Data Quality Monitoring Framework (DQMF) [41] performs data quality assessment by analyzing various monitoring data through user-defined algorithms. Offline, Tier0 monitoring will compare FTK tracks to offline tracks and also compare HLT objects reconstructed with FTK tracks to those reconstructed with offline tracks. The resulting plots will be used to determine the data quality.

The individual FTK boards have registers to store intermediate information such as the number of events received and dispatched, the link XOFF, processing errors, etc. All boards have FIFOs to handle the input and output data. The register values and FIFO states will be reported to the IS or filled into OH histograms via standard software applications.

All FTK boards will contain Spy Buffers to store data in various processing stages. They are circular buffers that hold event data that can be read out for diagnostic purposes (see section 4.1.3).

A processor blade will be deployed in the FLIC shelf. A monitoring application analogous to detector ROD monitoring could be implemented with the GNAM facility [42]. The fragments from the eight output channels could also be assembled into a full event record to allow advanced monitoring algorithms like ATHENA analysis to be performed.

TDAQ software provides methods for sampling event fragments in an individual ROS or fully built events on the HLT nodes (via the Data Collection Manager, DCM). ATHENA monitoring analysis can be performed either on FTK data alone or in correlation with the SCT/pixel ROD data. If FTK monitoring needs samples of events usually rejected by early HLT steps, a dedicated selection algorithm that tags the desired events will be developed and integrated inside the steering code.

The traffic load introduced by FTK monitoring will be configurable to ensure that the total bandwidth required for monitoring a single core Crate does not exceed the bandwidth of either the VME bus or the SBC Ethernet interface. At the same time the aggregated data rate from FTK will not exceed a pre-allocated fraction of the ATLAS control network bandwidth.

## 7 Vertical Slice Performance

The Vertical Slice is the system that we are using to integrate FTK into the TDAQ and test FTK prototypes. We wanted to start the integration as early as possible by using early FTK prototypes during the end of the 2012 run. As the FTK development proceeds, the FTK prototypes described in this TDR will become available. After tests at the home institutions, they will be included in the vertical slice for integrated tests with other FTK prototypes and with TDAQ.

## 7.1 Dual-output HOLA tests

The first test at CERN of an FTK prototype was performed in September 2011. The dual-output HOLA prototypes, which had been extensively tested in Chicago to a bit error rate  $< 10^{-15}$ , were available along with a first version of the FTK\_IM prototype. The dual-output HOLA was installed in the BOC card of pixel and SCT RODs in the SR1 building at Point-1. Using the internal ROD simulator, RODs were used to send fake data through the dual-output HOLA card which successfully transferred data to both a Robin card and a FTK\_IM card. There were no errors while data was flowing for more than a day, corresponding to about 1 billion bits. Data flow was also exercised with a light attenuator and found to work without error down to -14db. We also tested the XON/XOFF feature. We verified that:

- By default the dual-output HOLA ignores XOFF from the FTK channel;
- The FTK\_IM can send an enable\_XOFF command through the SLink return lines to enable the dual-output HOLA to respond to XOFF from the FTK channel;
- When the FTK\_IM has sent the enable\_XOFF command and it asserts XOFF the dual-output HOLA stops data flow;
- When XOFF from the FTK is ignored, the FTK fiber can be plugged/unplugged at will without interfering with data flow on the normal DAQ channel;
- The DAQ channel always complies with XOFF signals.

This last feature was also verified during late 2012 and early 2013 tests while data taking was in progress. This is an important feature for installing and commissioning additional FTK boards during post-LS1 running.

## 7.2 Vertical slice test

During the last part of Run-I we used existing prototype boards to begin integration tests of FTK in the ATLAS TDAQ environment during data taking. The goal was to execute the first part of the FTK algorithm, AM pattern recognition without Track Fitting, and to gain experience with TDAQ integration.

This activity started with the installation of a subset of dual-output HOLAs to allow the FTK vertical slice to be connected to the detector. The hardware available was:

- dual-output HOLAs,
- early FTK\_IM prototypes,
- EDRO boards designed for the LUCID project [28],
- early AM board prototypes from a silicon R&D + AM trigger test-beam experiment [29].

The existing boards allowed us to receive data from one detector tower corresponding to about 45 degrees in  $\phi$  and half of the barrel in the  $z$  direction. Figure 89 shows the detector ladders and modules for two vertical slice  $\eta-\phi$  towers. Two FTK\_IMs on an EDRO board can receive up to 8 SLinks from the detector and do data conversion in a way similar to the offline bytestream converters so that the EDRO receives detector hits identified by strip number as in the offline and a global module number. The EDRO board can perform the basic functions of the Data Formatter and Data Organizer boards: remapping input data to the appropriate logical layers for the AM, converting hits to superstrips, and sending superstrips to the AM board. The EDRO then waits for roads from the AM board and sends those roads, and optionally input hits, out to the ROS. The AM board instrumented with AM chips built for the CDF experiment

can store up to 640k patterns for pattern recognition. This is much less than the 8M patterns per board planned with the new FTK AM board and AM chip, but it is sufficient to verify the working principle and integration.

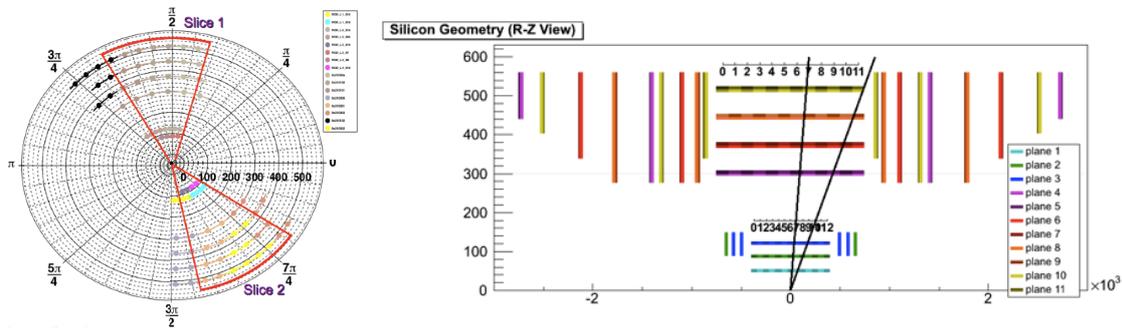


Figure 89: (left) Silicon modules corresponding to two vertical slice  $\eta - \phi$  towers ( $r\phi$  projection). (right) Silicon modules corresponding to each vertical slice  $\eta - \phi$  tower ( $rz$  projection)

During September-November 2012, we installed the vertical slice crate in USA-15 and prepared it to receive data from 4 SCT RODs. At that time, the FTK system was working in pass-through mode, but the vertical slice crate was included in run control and the two VME boards were configured with dedicated plugins through run control. We used spare beam-off time in addition to the technical stop and machine development periods to take data with the FTK vertical slice included in the ATLAS partition. We verified that data could flow through the FTK system up to the maximum level-1 trigger event rate of 70 kHz. This was mostly a test of integration and synchronization because there was no data processing.

At the beginning of 2013, we continued software and firmware development to decode input data, map detector hits to AM layers, and prepare a pattern bank with SS numbering compatible with the AMchip. Then during the last few weeks of 2013 data taking, FTK was again included in the ATLAS partition. Initially it was included only during beam-off operations. A low rate of BCID mismatches was observed, traced to a bug in the firmware, and resolved. FTK was then included in data taking during the 2.76 TeV pp fills, specifically in run 219171. At that time, the decoding of SCT data was not perfect; some features of the offline bytestream conversion needed to be added to the decoding firmware. We looked at the acquired data to compare the list of patterns found by the FTK hardware in each event with the list of patterns found by the FTK simulation running on the raw SCT data. After correcting the simulation for the known differences in hit decoding, we found that all expected patterns were detected by the hardware. Unexpected patterns were found at a rate of less than 1 per mil. The cause for these additional roads are hits from two consecutive events that are merged on rare occasion, which would be fixed by a minor firmware change. Overall we were able to get early FTK prototypes included in ATLAS data taking and to successfully compare the preliminary FTK output to FTK simulation. While this early commissioning was in progress there was no impact on ATLAS data taking.

### 7.3 Next steps

The FTK prototypes that are being being manufactured and tested at home institutions will be integrated together at CERN. For this purpose, we will be using a vertical slice setup in the TDAQ test lab during LS1. As shown in Fig. 92, global integration tests will be carried out in the vertical slice in mid-2014. Later that year, AM boards containing the AMchip06 chips will be tested. Toward the end of the long shutdown we will start again to use the vertical slice crate at Point-1 for integration tests with the Inner Detector and TDAQ. All pixel and SCT layers will be used and the boards will be the final FTK prototypes.

## 8 Phase-II Compatibility

The increased complexity of tracking in the Phase-II environment [14] will require an increase in FTK computing power. During Phase-II the mean pile-up with luminosity leveling will be 115-140 events per bunch crossing [5]. The maximum average event pileup to be used for design purposes is 200 pp interactions per bunch crossing to be compared to a maximum average of 80 for Phase I. In addition, the L1 accept rate, which is the input rate to the FTK, could increase to 200 kHz. This means a factor of 5 higher data flow, measured as event rate times event size. Furthermore the factor of 2.5 increase in pileup has an exponential effect on the complexity of the pattern recognition problem. Designing a Phase-II system now would be very costly. On the other hand, given the expected improvement in electronics between now and the start of Phase-II in 2023, more powerful systems will be possible in the future. Two upgrade paths are discussed below. The baseline path minimizes new hardware. The other is a full replacement of the FTK electronics, which makes maximum use of newer technologies to make a smaller and more modular system. Further studies will be required to understand if the latter is needed.

For Phase-II the L1Track trigger system will be installed. It is expected to do tracking in regions of interest covering an average of  $\sim 10\%$  of the detector phase space in an event and to have a higher  $p_T$  threshold, perhaps 5-10 GeV. During Phase-II the FTK will provide functionality for the HLT beyond the capabilities of L1Track:

- full tracking for primary vertex finding, Jet Vertex Fraction for assigning objects to the hard scattering vertex, and identification of objects outside of the level-1 regions of interest,
- low  $p_T$  tracking for all algorithms that need low  $p_T$  tracks, e.g.  $\tau$  leptons,  $b$ -jet tagging, lepton isolation, and possibly energy flow.

The driving motivation for the upgrade is the increase in L1 accept rate and event pileup rather than the change in detector geometry which would be largely compatible with the current hardware. The baseline geometry for the new Inner Tracker (ITK) is described in [14]. The ITK will be a full replacement of the Inner Detector. It will be an entirely silicon detector extending to a radius of 1m. Both the pixel and the strip detectors will have higher granularity than the current detector. The geometry is designed to have each track cross 4 pixel layers and 5 silicon strip modules for a total of 14 measurements: 4 from pixels, 5  $r\phi$  and 5 stereo from the strips. A good compromise between efficiency and fake track rejection is achieved by requiring at least 11 clusters on a track as described in [14]. The upgraded FTK could require 7 out of 8 layers in pattern recognition, using the 4 pixel layers and 4  $r\phi$  strip layers (or 3 pixel layers and all 5  $r\phi$  strip layers). The requirement of 7 out of 8 layers for the AM will provide good rejection with minimal efficiency loss as was seen for the Phase-I FTK. This means that the only change for FTK would be increasing the number of layers used in the second stage fit by 2. If additional rejection is needed at the AM level, it will be possible to include more layers in the AM pattern recognition. The number of patterns needed with more layers can be kept under control with a more extensive use of variable resolution, *i.e.* Don't-Care bits. As described below, the most likely scenarios foresee that a new AM chip will be needed for Phase II in order to profit from the expected advances in microelectronics. The new ITK layout will have more modules and will extend to a radius of 1m. Thus the number of FTK sectors used for track fitting will increase, a number that will have to be evaluated.

The baseline upgrade option described below will retain the 1 GeV  $p_T$  threshold for FTK tracks, the same as in Phase-I, which should be possible with newer electronics. If Phase-II performance studies show that the  $p_T$  threshold can be increased to 2 GeV, there would be a significant reduction in cost since the number of patterns needed for a given efficiency (*i.e.* the number of AM chips), the number of fits (*i.e.* track fitter hardware), and the size of the overlap region in the  $\phi$  direction would all decrease by a factor of 2. Additional savings could come from hit filtering ideas that have been suggested [43]. A hit filtering scheme based on cluster size is shown to be effective in the rejection of hits associated with tracks with

$p_T$  below 400 MeV for the innermost strip layer, although the rejection power for pixels would be small. Filtering based on  $p_T$  using hits in 2 closely spaced axial sensors would make a dramatic reduction in the hits that have to be processed, but the double-axial layers have been shown not to be optimal for physics performance and this geometry is unlikely to be employed. We note that while such hit filters would be implemented on the detector for L1Track, for FTK they could be implemented at the FTK input stage. We stress that FTK does not require these filtering schemes, rather they would further reduce hardware to be replaced. The discussion below assumes no hit filtering.

## 8.1 Baseline upgrade path

In order to estimate which elements of FTK will have to be replaced, we compare the hardware limits in all parts of FTK with Phase-II conditions through the FTK pipeline. With some compromise it seems possible to use most of the FTK boards in Phase-II.

### 8.1.1 Input to the Fast Tracker and the Data Formatter system

The FTK input is received from SLinks on the FTK\_IMs (see 4.3). Factors affecting the input data rate are:

- $\times 2$  due to the increase in L1 accept rate,
- $\times 2.5$  due to increased pileup,
- $\sim +12\%$  due to one additional detector layer,
- $\approx -30\%$  for pixel layers due to pair clustering in the FE-I4 and FE-I5 front-end chips,
- increase in the number of pixels/cluster since the inner two pixel layers are expected to have 25um x 150um wide pixels.

Overall the input data flow will increase by a factor of approximately 5. Simply increasing the number of FTK\_IMs and consequently DFs by a factor of 5, although feasible, is not realistic. Instead we would replace the FTK\_IMs with higher input bandwidth devices. This will most likely be needed in any case since faster fibers will certainly be used during Phase-II. The bandwidth available on the FTK\_IM to DF connector (FMC connector) exceeds 70 Gbps which is more than 5 times the current FTK\_IM input bandwidth of 8 Gbps. Clustering is performed in the FTK\_IMs which will output the cluster centroids for downstream processing. The increased number of pixels per cluster will only affect dataflow in the FTK\_IMs, not in the downstream logic. An interesting alternative solution that can be explored in collaboration with the ITK group is to place the DF boards in ATCA shelves shared with the ITK RODs. This approach would have the big advantage of reducing the overall fiber usage. In this scenario the FTK\_IMs will work as co-processors for the DF boards. After a study of cluster size filtering in the strip detectors, we will include this algorithm in FTK\_IM or DF logic to reduce downstream data flow.

We also must understand whether the current DF system of 32 DF boards will be sufficient for Phase-II. The bandwidth limits of the current system are given in section 4.4.3. The bandwidth to the AUX card would allow a factor of 5 increase, while other channels have safety margin factors between 2.5 and 3.8. Assuming the cluster size filter gives a factor of 2 reduction in data from strips, all data flow related to strips is already compatible with Phase-II. In the worst case, the DF system might need double the current bandwidth which could be provided by one of three options. The first is doubling the existing cards (or equivalent cards with newer components). The second is a replacement of the existing cards with boards at least twice as fast. The third option is probably the most interesting. Given the modularity of the DFs and the ATCA crate, a few newer DFs with higher bandwidth could be added to complement the existing

ones. A dedicated study will be needed to determine the number of new DFs and their configuration. If ATCA backplanes ready for a 40 Gbps fabric are chosen for the initial FTK system, they would have a factor of 2 safety margin for the fabric interface with respect to Phase-II needs.

### 8.1.2 The Processing Units

The Processing units represent the core processing power of FTK. They are made of the AUX card, the AM board and the LAMB that houses the AMchips. The processing power is given by the FPGAs in the AUX cards and by the AMchips. For the processing units, we consider two scenarios. The first is a minor upgrade where only the LAMBs are replaced. The second is a full replacement. The DF to AUX bandwidth is already compatible with Phase-II. Regarding the data flow of superstrips, the AUX to AMB links include a factor of  $\approx 3$  safety margin in bandwidth and the AMB to LAMB links include a factor of  $\approx 2$  safety margin in bandwidth. The needed factor of 5 increase in bandwidth can be achieved for AUX to AMB communication by increasing the number of  $\eta$ - $\phi$  towers from 64 to 128. For smaller  $\eta$ - $\phi$  towers the overlap region becomes more important. This means that to actually reduce the bandwidth we might need to increase the  $P_T$  threshold for tracks from 1 GeV to a value below 2 GeV or accept some inefficiency for tracks with  $p_T < 2$  GeV at the tower boundaries. Whether the threshold can be kept close to the currently used 1 GeV will need a dedicated study which considers the effect of cluster size data reduction. In this scenario, the increase in pattern recognition power of the AMchips must be sufficient to keep under control both the number of roads, in order to not exceed the output bandwidth of the LAMBs, and the number of fits, in order to not exceed the computing power of the track fitters on the AUX card. Whether these limits will be satisfied by the next generation AMchip will depend on the AMchip R&D results. One issue that must be investigated before FTK production is how to make the AMB-LAMB connector compatible with the powering needs of the next generation LAMBs.

As an alternative, the upgrade in processing power can be shared between the new AMchips and new Track Fitters. In this scenario, the new LAMBs will include next generation AMchips along with 2022-era FPGAs for track fitting beyond the Gfit/s/FPGA. The upgrade in track fitting power on the LAMB board will allow first-stage track fitting on the LAMBs themselves, thus making the high speed connections between the AMchips and the Track Fitter more local. We expect that future AMchip R&D will provide in a single package AMchip die(s) along with one FPGA die to further reduce the size of the system and improve the AMchip to Track Fitter I/O. This scenario is probably the best one because it allows both the AM and TF power to be increased with a minimal impact on the hardware to be replaced. The replacement of the TF would handle the larger number of FTK sectors with the ITK. The only issue is that over the AUX to AMB to LAMB data path we would need to transfer hits instead of SSs which would require a factor of 2 more bits for pixels<sup>13</sup>. Some bits can be saved dropping the information on the cluster size that is currently not used in any stage of the FTK. A dedicated study will be needed to evaluate the actual limit of this upgrade path. The outcome might well be that in order to satisfy the worst case Phase-II conditions, the existing 128 Processing Units with new LAMBs will need to be complemented by some additional PUs as proposed for the DF system. The new DF with PU functions could be squeezed into a single board as described in section 8.3.

### 8.1.3 Second stage board, and FLIC

Given the uncertainties in the earlier parts of the system, it makes little sense to discuss the compatibility of the SSBs and FLICs because it depends on the output of the first stage. Since the 32 SSBs and 2 FLICs contain a fraction of the FTK computing power, their replacement is not a big cost driver. Several options are possible depending on the FTK upgrade path that will be chosen.

---

<sup>13</sup>Currently SSs are 16 bits for all layers, while pixel hits are 32 bits and SCT hits are 16 bits.

### 8.1.4 Brief summary

Based on what we know today, we can say that:

- A new input path to FTK will be needed. This will be either new FTK.IMs or DFs moved into ITK ATCA crates to receive data from the RODs over the ATCA fabric;
- Additional DFs are likely to be needed;
- A new AMchip and new LAMBs will probably be needed with or without Track Fitters on board;
- If additional bandwidth to the LAMBs is required, the existing 128 PUs will be complemented by additional processing units exploiting 2022 technology;
- Upgrade of Second stage boards and FLIC will be evaluated later.

## 8.2 AMchip R&D

For the requirements on AM chip, the most important change in Phase-II is the increase of the maximum expected average pileup from 80 to 200 pp collisions per bunch crossing. The greater tracking complexity necessitates increased computing power in FTK and as a result the AM chip will have to be upgraded. In the baseline ITK design, no hit filtering is implemented. A filter on cluster size can be implemented at the input of FTK, but we won't consider that gain in the numbers below. For maximum average pileup of 200, more AM patterns will be needed in order to have a comparable Super-Strip occupancy to that in the current FTK. Furthermore the lengthening of the luminous region from  $\sigma_z = 5.5\text{cm}$  to  $\sigma_z = 7.5\text{cm}$  will proportionally increase the needed number of patterns. The required pattern bank size has been extrapolated from the Phase-I system. Based on our experience, increasing the AM resolution by a factor of 2 to compensate for a factor of 2 increase in average pileup requires roughly a factor of 7 more patterns. Using this scaling for the hit density and luminous region, we can expect that in Phase-II we will need to increase the total number of patterns in the AM system by a factor of 8-16 over the 1 billion FTK patterns in Phase-I. It is possible that a more extensive use of variable resolution, possibly in combination with using more layers in the pattern recognition, will reduce the need for more patterns, but this will require additional study of pattern generation. Since the associative memory works with a reduced resolution for pixels and strips, it will not be affected by the increase in pixel granularity.

Looking at the history of the AM device, we had the first AMchip designed in the '90s and operational in 2000 with 128 patterns/AMChip [15]. The second generation AMchip was designed in 2003-2004 and started operating in 2005 with 5120 patterns/AMchip [44], and today technology allows us to plan the AMchip06 to have 64k or 128k patterns in 2014 [45]. Even the most ambitious goal (without any hit filtering assumptions) of 1M or 2M patterns/AMchip in 2022 seems within reach. With this pattern density 8000-16000 AMchips<sup>14</sup> will be needed to reach 8-16 billion patterns. Certainly it will require significant R&D to achieve the needed pattern density and to keep power consumption under control. One possibility is to stack more than one die in a single AMchip package to save area on the PCB. This approach helps to increase the area density but still requires a reduction in power consumption per pattern. The L1Track may also use the AM technology. It is possible to merge the requirements for FTK and L1Track into a common design and thus share the development costs.

## 8.3 ATCA-based upgrade path with full replacement

The discussion above suggests that the AMchip will have to be replaced to meet the Phase-II challenges. Expected technological advances will allow more hardware to be placed in the same board area. In

---

<sup>14</sup>This is the same number of AMchips planned for the Phase-I FTK.

particular for the AMchip we foresee the possibility of stacking more than one device in a single package. The Data Organizer and Track Fitting functions could be moved onto the LAMB board closer to the AMchips, or perhaps even inside the AMchip package itself by stacking one FPGA on top of the AMchip dies. Figure 90 shows multiple dies wire-bonded in the same package, a technology that exists today.

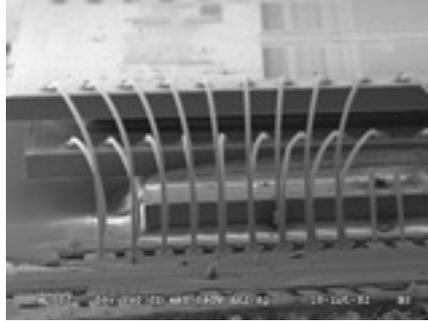


Figure 90: Multiple dies wire-bonded in the same package (see <http://www.aseglobal.com/>).

An FTK LAMB will hold 16 AMchips with 128k patterns each. Given the expected increase in the number of patterns per AMchip, we foresee by 2022 a LAMB with 16 newer AMchips each storing 1-2M patterns and sufficient track fitting power. Assuming we replace the VME crates with a system fully based on ATCA, the new LAMBs could be put directly on the ATCA board thus merging the DF function with the AMB and AUX card functions in a single main board. In this scenario, an ATCA rear transition module would be used to receive data from the ITK and to share data among the crates. The ATCA main board would perform pixel clustering and data distribution as the DF is currently doing. The full upgraded FTK system could have a size similar to that of the current FTK system with a total of 12 ATCA crates. Assuming as now there are 512 LAMBs distributed over 128 boards, they could be organized as 11 boards per shelf times 12 ATCA shelves, or 8 boards per shelf times 16 ATCA shelves. The remaining ATCA slots would house the SSB and FLIC functions. In this scenario a newly designed ATCA card could be made suitable for all the FTK functions. The fully ATCA scenario would be pursued only if dedicated studies of the FTK upgrade for Phase-II show that a majority of the components would have to be replaced.

## 9 Project Management

FTK will be built by a group of ATLAS institutions that commit to designing, constructing, and commissioning the system. Since trigger malfunctions, either obvious or subtle, can have serious consequences for physics analysis, the FTK institutions will also be responsible for long-term monitoring and maintenance of the system. Many of the FTK institutions have long experience in designing and maintaining trigger hardware for high luminosity hadron collider experiments as well as in extracting science from the data selected by that hardware.

The FTK construction project will have a distinct organization and management structure, but it will remain embedded within the overall TDAQ project, as it has been in the past. There are regular discussions between FTK and TDAQ leadership, and the FTK Project Leader reports to the TDAQ Steering Group and the Upgrade Steering Committee. Today a majority of FTK institutions are members of the TDAQ Institute Board (TDIB). All FTK institutions will be encouraged to become members of the TDAQ Institute Board.

## 9.1 Management Structure

The FTK project will have a Project Leader (PL), a Technical Coordinator (TC), a Management Board (MB) consisting of leaders of individual tasks, and an Institute Board (IB). The roles of each are described below. The organization is shown in Fig. 91.

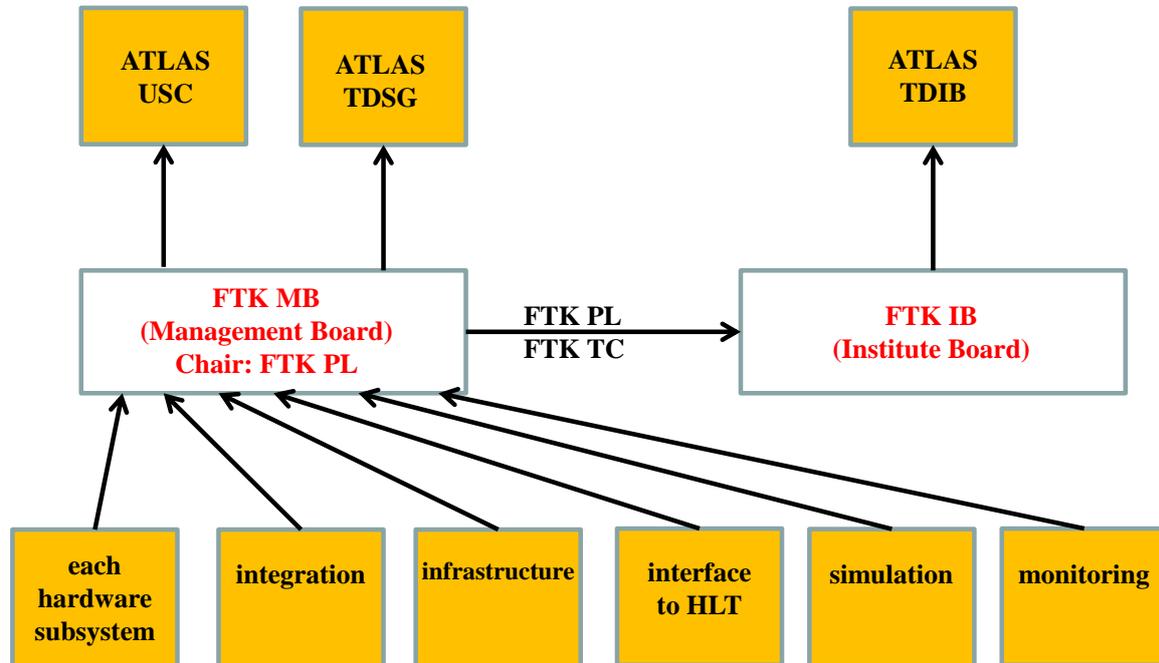


Figure 91: The organization chart for the FTK project.

### 9.1.1 Institute Board

As noted above, all FTK institutions will be encouraged to become members of the TDAQ Institute Board which has overall responsibility for the ATLAS TDAQ systems. The leader of the FTK effort at each FTK institution will form the FTK Institute Board (FTK-IB). This body will function as a subcommittee of the TDIB for the duration of the construction project. The FTK-IB will elect the Project Leader and confirm appointments made by the Project Leader to other positions in the management team. Although the Project Leader may choose to discuss organizational issues with the FTK-IB, it is expected that, as has been the case in the past, all discussions regarding FTK, its implementation, and its capabilities will continue to involve the entire FTK group, typically at its weekly meeting. Once FTK construction is complete, the FTK construction project and the FTK-IB will be dissolved.

### 9.1.2 Project Leader and Technical Coordinator

The FTK project is led by a project leader (PL), elected by the IB from candidates selected by an IB-appointed search committee. The PL is responsible for overseeing the design, construction, installation, commissioning, and maintenance of FTK, and for regularly communicating the status of the project to both ATLAS management and the TDAQ management team.

The PL will be elected after the approval of the FTK TDR by the LHCC. The PL term is 2 years, with periodic reelections as prescribed by the ATLAS collaboration. The PL will appoint members of

the management team including the Technical Coordinator. Such appointments must be ratified by the IB.

### 9.1.3 Management Board

The Management Board consists of the leaders of the major FTK activities. Management Board members are responsible for the activities being effectively executed, including meeting milestones and overseeing resources. The major activities at present are the design and construction of each of the hardware subsystems, testing and integration in the Vertical Slice at CERN, infrastructure issues (power, cooling, etc.), interface to the HLT, FTK simulation, and monitoring. The major activities will change with time.

## 10 Cost and Funding

A cost estimate for the construction of FTK was made at the time of the Phase-II letter of intent. That was an early estimate, done before the boards or AM chip were fully designed and engineered. Most of the system has now been designed by engineers, and we find that the total cost of the system has increased by approximately 10%. This is partially the increased cost of manufacturing the ASIC chip and partially the need for more powerful FPGAs in the system.

Our current understanding of the core cost for producing FTK is shown in table 14, which also shows the distribution of the funding over the collaborating institutions. The cost basis for the estimate is as follows. For boards in the prototyping stage, the cost is based on industry quotes for the PC boards, parts, and assembly. For boards with a new version in design, the cost is based on the cost of the previous version and the parts that are to be used. For the AM chip, the one ASIC in the system, the cost is based on a quote for production and an estimate for testing at the production house, since we do not yet have a quote for that. But the largest uncertainty in the budget for the AM chip is the future dollar/Euro exchange rate; 65% of the chip's funding is in Euros, while the cost of production will be billed in dollars.

The cost estimates were made in Euros and dollars. They were converted to Swiss Francs on May 15, 2013, at the exchange rates for that day: 1 Euro = 1.243 SF, 1 dollar = 0.957 SF. The US funds have been fully received. The other funds are based on discussions over the past several years with the funding agencies.

## 11 Sharing of Work and Responsibilities among Institutions

The FTK institutions have all been working on the development of the project and are committed to its successful construction and operation. Overall system monitoring and software development including simulation will be a shared responsibility of all the institutions. Here we specify the individual responsibilities agreed upon within the FTK collaboration. Each institution that is building hardware is committed to long-term monitoring and maintenance of that subsystem.

1. **Argonne (USA):** Argonne is responsible for the design, construction, testing, integration, and commissioning of the FLIC boards. This work will be done with Northern Illinois University.
2. **AUTH (Greece):** AUTH will participate in the firmware development for the AMBSLP, LAMB-SLP, and FTK\_IM boards as well as in the testing, integration, and commissioning of those boards.
3. **Chicago (USA):** Chicago is responsible for the design, construction, testing, integration, and commissioning of the HOLA cards and the AUX boards. In addition, Chicago assumes the maintenance responsibility for the Data Formatter boards that have been designed by Fermilab.

ITEM	COST	IT	US	CH	JP	DE	FR (LPNHE)	GR (AUTH)
HOLA: produced and paid in 2011								
FTK_IM	336	168			168			
DF	297		297					
AUX	847		847					
AM boards & LAMBs	684	595		65			12	12
AM chip	1265	616	456	25	44	124		
SSB	247		247					
FLIC	34		34					
Infrastructure	230		155		75			
<b>TOTAL</b>	<b>3940</b>	<b>1379</b>	<b>2036</b>	<b>90</b>	<b>287</b>	<b>124</b>	<b>12</b>	<b>12</b>

Table 14: The core costs of FTK in thousands of Swiss Francs. Following the cost of each item is the source of the funds - Italy, United States, Switzerland, Japan, Germany, France, and Greece. The cost of the AM chip includes the masks, wafers, packaging, testing, and production. The infrastructure includes crates, shelves, fans, power supplies, and fibers.

4. **Geneva (Switzerland):** Geneva is responsible along with Pisa for the design and construction of the LAMBSLP boards. Geneva, Pisa, and AUTH are responsible for LAMBSLP testing, integration, and commissioning.
5. **Heidelberg (Germany):** Heidelberg is responsible for testing the production AMchip06 chips and will participate in prototype development and characterization.
6. **Illinois (USA):** Illinois is responsible for the design, construction, testing, integration, and commissioning of the Second Stage Boards.
7. **Laboratori Nazionali di Frascati (LNF-INFN, Italy):** LNF is responsible along with Waseda for the design, construction, testing, integration, and commissioning of the FTK\_IM boards. AUTH will collaborate on the 2-D pixel clustering firmware. LNF is also responsible for the AMchip06 custom cell design and for coordinating the FTK simulation.
8. **LPNHE (France):** LPNHE is responsible along with Milan for the design and construction of the AMchip06.
9. **Melbourne (Australia):** Melbourne will participate in the testing, integration, and commissioning of the AMBSLP boards.
10. **Milan (INFN, Italy):** Milan is responsible along with LPNHE for the design and construction of the AMchip06. It is also responsible for the production of the INFN test stand for AM chip prototype characterization.
11. **Northern Illinois (USA):** Northern Illinois is responsible for working with Argonne on the design, construction, testing, integration, and commissioning of the FLIC boards.
12. **Pavia (INFN, Italy):** Pavia is responsible for FTK infrastructure and the test stand for LAMBSLP board production.
13. **Pisa (INFN, Italy):** Pisa is responsible for the design and construction of the AMBSLP boards. Pisa is responsible along with Melbourne and AUTH for the testing, integration, and commissioning of those boards. Pisa is responsible along with Geneva for the design and construction of the

LAMBSLP boards. Geneva, Pisa, and AUTH are responsible for LAMBSLP testing, integration, and commissioning.

14. **Waseda (Japan):** Waseda is responsible with LNF for the design, construction, testing, integration, and commissioning of the FTK\_IM boards. Waseda will also participate with INFN in the AM chip system commissioning and in producing and managing the AM pattern banks.

Table 15 is a summary of the hardware responsibilities.

Institution	Country	HOLA	FTK_IM	DF	AMB	LAMB	AMchip	AUX	SSB	FLIC
Argonne	United States of America									√
AUTH	Greece		√		√	√				
Chicago	United States of America	√		√				√		
Geneva	Switzerland					√				
Heidelberg	Germany						√			
Illinois	United States of America								√	
LNF-INFN	Italy		√				√			
LPNHE	France						√			
Melbourne	Australia				√					
Milan	Italy						√			
Northern Illinois	United States of America									√
Pavia	Italy					√				
Pisa	Italy				√	√				
Waseda	Japan		√				√			

Table 15: Construction responsibilities of FTK institutions.

## 12 Schedule and Milestones

The FTK team's major tasks are to complete component design, test each board independently, test each board with those boards to which it communicates, integrate the board types and carry out global integration, and prepare a system for data taking in 2015. The schedule for the next 2 years is shown in figure 92.

Many of the prototype boards are either currently being tested, are being manufactured, or are about to be manufactured. Their testing is scheduled for this summer, to be followed by testing of multiple cards in the autumn and winter and then global testing in spring, 2014. There are two exceptions. The HOLA cards that were expected to be needed were all produced and tested over a year ago because they had to be installed early in LS1 in order not to negatively impact the work that the silicon groups have to do during LS1. They will be installed at Point-1 in July of this year. There was a recent ATLAS decision to increase the number of silicon detector RODs for 2015 data taking. We have enough HOLAs for them all, but our stock of spare boards will drop to 0. As a result, we are building an additional 40 HOLAs, which will be produced by the early part of the summer.

The second exception is the Associative Memory system, whose schedule is dominated by the AM chip. The AM chip still has a few development steps; the planned submissions are shown in pink in the schedule. The first one is the AMchip05 MPW submission to produce a small chip (12 mm<sup>2</sup>) in the final

Months	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6
<b>Tasks</b>	2013						2014						2015											
Dual Output HOLA													Global Int.											
FTK Input Mezzanine				w DF/IBL/RODs									Global Int.											
Data Formatter		test		w AUX									Global Int.											
Miniasic		test																						
AMchip05		tapeout			test																			
AMchip06											tapeout				test									
AMBSLP-Mini-LAMBSLP			test																					
AMBSLP-LAMBSLP					w AM05	w AUX							Global Int.				w AM06	8-16 PUs						
AUX CARD		test		w DF	w AMBSLP								Global Int.					8-16 PUs						
Second Stage Board (SSB)		test		w FLIC/AUX									Global Int.											
FTK Level-2 Interface Crate (FLIC)		test		w SSB - ROS									Global Int.											

Figure 92: The FTK schedule for the next 2 years. Green represents independent testing of a board, while blue signifies testing a board with the boards with which it communicates. Yellow represents global integration of the system elements, and orange is production and commissioning.

BGA package. This chip will be used to test the AM and LAMB boards alone and as part of the partial and global integration tests. The second one will be for the final 128k-pattern AMchip06, which will be submitted after the AMchip05 has been successfully tested. The availability of the final chip dictates when the final Processor Units can be installed. We will have 16 Processor Units for 2015 data taking, which will cover the full barrel region of the detector for pile-up up to 46. The number would drop only if there is an unexpectedly low yield of the AMchip06 or the LAMB and AMB boards that hold them. In 2016, we would have 32 Processor Units, which would cover the entire detector again assuming pile-up as high as 46. The remaining Processor Units would be built and installed as needed by the LHC luminosity profile and allowed by the European funding profile.

The major milestones for the project are:

1. Submission of the AMchip06 in the spring of 2014
2. Installation of a system to cover the barrel region for pile-up 46 by July, 2015
3. Installation of a system to cover the entire detector for pile-up 46 by the end of 2015.
4. Installation of a system to cover the entire detector for full Phase-I pile-up by the end of 2018.

The regions covered by milestones 2 and 3 would drop if the yield of AMchip06 or the LAMB and AMB boards is much less than for our recent prototypes.

### 13 Appendix I: AM chip design

**The single layer:** In more detail, the layer block shown in Fig. 93 consists of a current source, 4 NAND CAM cells, 14 NOR CAM cells, and a SR-FF. The current source charges the match line with a constant current only if the data stored inside the cells match the input data on the bit lines. On the contrary, if at least one of the NAND cells does not match the input data it breaks the match-line, preventing it from being charged. If a NOR cell does not match, a path from the match line to ground is created, and consequently, the match line is discharged. In terms of node voltages, if all the cells match the data input, the match line is charged up to a high logic value (about 1 V); on the contrary, if at least one of cells does not match, the match line is not charged and the voltage is approximately 0 V. To save power we have used two different match-line driving schemes: (1) current race scheme, and (2) selective pre-charge scheme. This is the reason we have used two different types of cells. Both schemes are based on standard CAM designs [46].

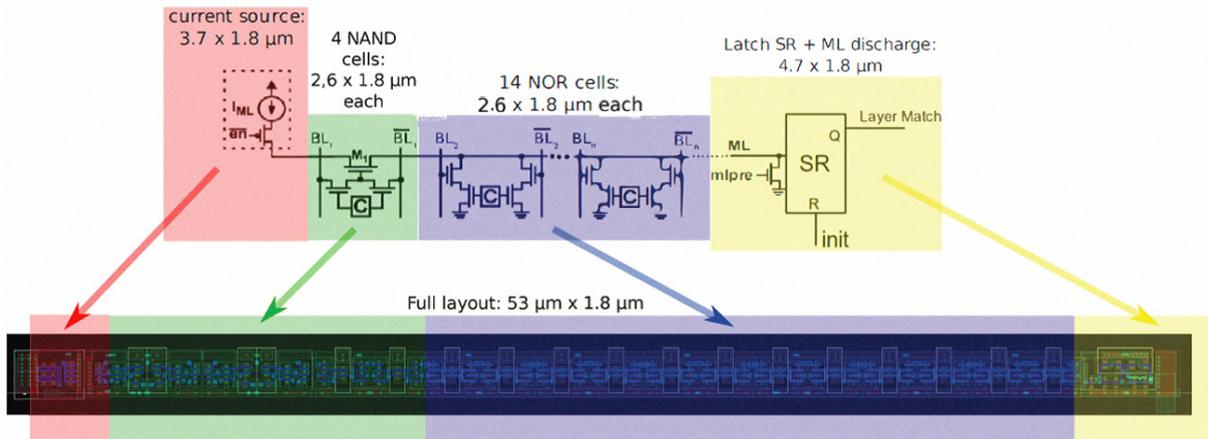


Figure 93: Schematic and layout of a single layer.

Since our goal is to increase cell density, we minimized the area of each cell. To this end, we designed a full-custom layer block. That enabled us to place staggered transistors with a gate width equal to the minimum value allowed by the technology rules. To guarantee compatibility and easy integration with the standard cell library, we designed all the cells using a VDD-to-VSS wire pitch equal to  $1.8 \mu\text{m}$  and VDD/VSS wire widths of  $0.33 \mu\text{m}$ . The layer block size and the size of each component are given in Table 16. The transistors in the current source were carefully designed to ensure that they are identical (same W/L ratio, same gate orientation, same number of contacts and distances, and same diffusion areas) to the transistors of the reference block to ensure good current matching. A single reference is used for 64 current sources to save area and power. In the SR-FF block, we also used a reset transistor controlled by the initialization signal.

Current source	$3.7 \mu\text{m} \times 1.8 \mu\text{m}$
NAND cell	$4 * 2.6 \mu\text{m} \times 1.8 \mu\text{m}$
NOR cell	$14 * 2.8 \mu\text{m} \times 1.8 \mu\text{m}$
SR-FF	$4.7 \mu\text{m} \times 1.8 \mu\text{m}$
Total	$55 \mu\text{m} \times 1.8 \mu\text{m}$

Table 16: AMchip layer block sizes

**Full Custom Array:** To design the 64-half-pattern block (Fig. 94) we created an array of  $64 \times 2$  layers called TOP. We then duplicated this block to produce the 64-half-pattern block ( $64 \times 4$  layers) called TOP2. TOP2 is a large full-custom block designed using Cadence Virtuoso and the TSMC design kit at 65 nm. The upper part of Fig. 94 shows the floor plan of the  $64 \times 2$  layers (TOP), while the lower part shows two TOP2 blocks with the majority and Fischer tree logic between them. The majority and the Fischer tree blocks were designed using standard cell logic.

**Standard Cell and Full Custom Integration:** The entire chip was designed using a hybrid approach. More repetitive regions were created with a full-custom design, while the more complex logic was produced with a standard-cell implementation. To place and route standard cells, we used Cadence Encounter. Figure 95 shows the floor-plan of the entire chip. The AMchip has an area of  $14 \text{ mm}^2$  ( $3510 \mu\text{m} \times 3985 \mu\text{m}$ ). The memory is organized as an array of  $22 \text{ columns} \times 12 \text{ rows}$  of full-custom macro blocks (TOP2). The majority logic and the Fischer tree are placed between two TOP2 blocks.

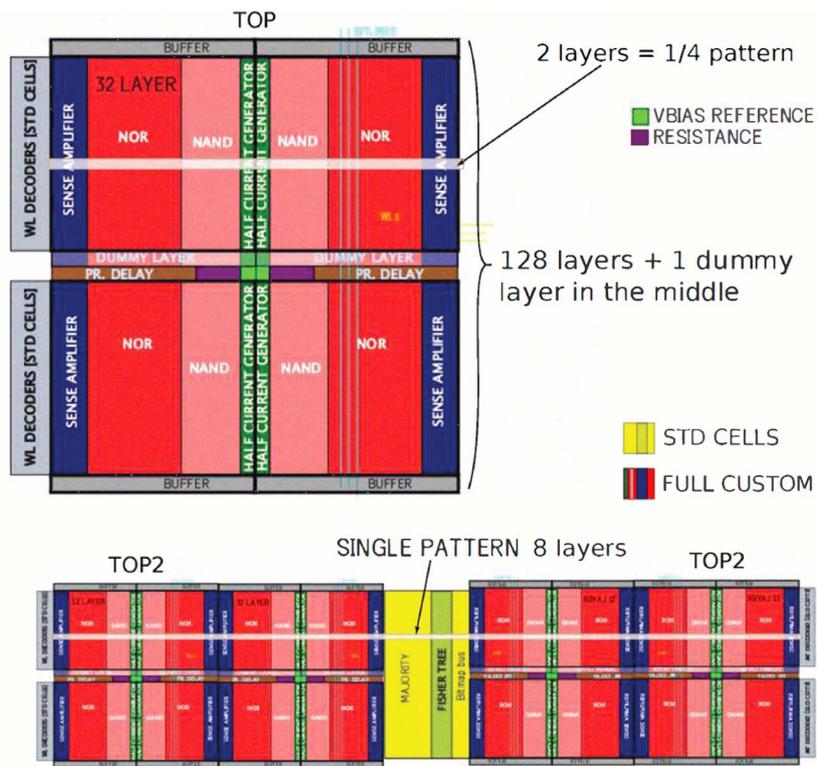


Figure 94: Full custom floor-plans. The block in the upper left is an array of  $64 \times 2$  single layers (TOP)

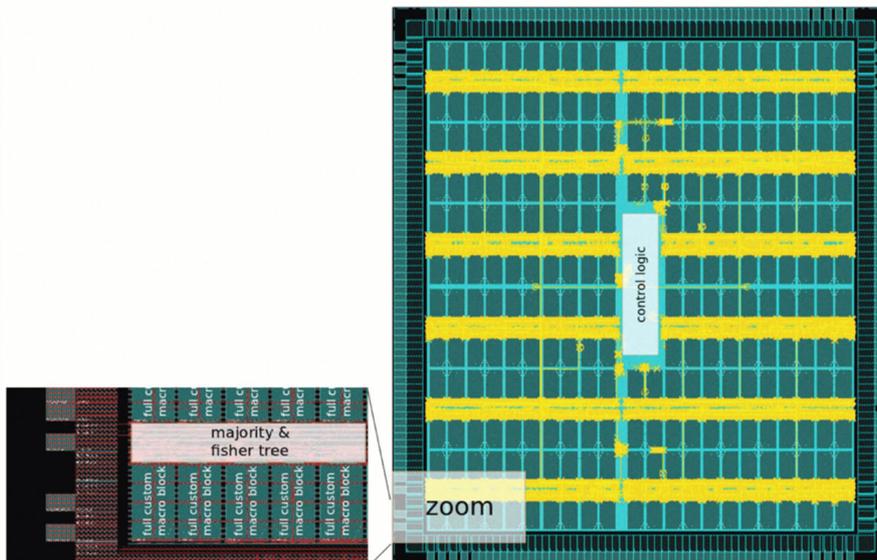


Figure 95: Full AMchip floor-plan.

The blocks not designed with full-custom logic were automatically placed by Encounter using a flat description of the logic. In addition, to decrease routing congestion, we designed fence areas that contain the majority, Fischer tree, and 4 TOP2 macro blocks. We placed the left-lower TOP2 without rotation, the right-lower TOP2 with horizontal mirroring, the left-top TOP2 with vertical mirroring, and the right-top TOP2 with a rotation of 180°. To prevent routing congestion, we also designed partial placement blockages (13.2% of cells are allowed) in the narrow routing channels at the boundary between two TOP2 blocks. In this way only bitline buffers are placed in these narrow channels. The 13.2% value was carefully calculated for this purpose. In the middle of the chip, we left free space for the control logic and the JTAG machine. In the surrounding area we placed a ring of 208 pads.

We estimated that the chip would have current consumption of about 70 mW for the core 1.2 V from the full-custom cells only. For this reason, metal-1 and metal-6 strips (the two main metals used for powering) are not sufficient to guarantee good power distribution. Hence we placed horizontal power strips in metal-5 (width = 3600 nm). These strips were placed in a staggered manner to prevent routing congestion. We also placed a large number of power and ground pads to guarantee adequate power supply for the core (Fig. 95).

We chose bidirectional pads with a driving current of 2 mA or 4 mA. Even with only 2 mA the estimated pad to output time (including the PCB line capacitance for up to 3000 mils) is less than 4 ns. Since the power consumption of the chip will be significant (~230 mW), we filled the empty spaces with decoupling capacitors to fully filter the power. The expected power consumption of the I/O pad ring is 100 mW at 3.3 V.

For placement and routing, we used Foundation Flow by Cadence Encounter which contains important optimization steps: pre-CTS (Clock Tree Synthesis) optimization, post-CTS optimization, and post-route optimization. These steps were performed to enhance the timing performance of the chip. The clock tree was generated and results confirm that the clock distribution is good. The maximum clock skew is equal to 400 ps. The timing constraints are described in a .sdc file which contains: (1) setup time for input clocked registers ranging from 0.1 ns to 2.5 ns, (2) output time after the clock for all outputs ranging from 0.1 ns to 2.5 ns, and (3) a minimum clock period equal to 10 ns. All these constraints were fulfilled in each optimization step. Table 17 gives a summary of the area occupied by each part of the chip. The largest areas are taken by the full custom TOP2 blocks, the pad ring, and the big vertical channels.

Region	Size	Relative size
Big vertical channels (maj., readout)	2.018 mm <sup>2</sup>	14.34%
Horizontal channels	0.178 mm <sup>2</sup>	1.26%
Small distributed channels	0.880 mm <sup>2</sup>	6.25%
TOP2	7.050 mm <sup>2</sup>	50.13%
Central region	0.415 mm <sup>2</sup>	2.95%
Pads (including boundary scan)	3.522 mm <sup>2</sup>	25.04%

Table 17: AMchip region block sizes

## 14 Appendix II: AM cooling tests

The final FTK core crate will contain up to 16 AMBFTK boards. If we assume that an AMchip06 core uses 1.2 W as described in section 4.6.6, a single AMBSLP will consume 150 W for the AM chip core, 125 W for serial link distribution, and 15 W for the SERDES IP on the AM chips, for a total of 4.6 kW only considering the AMBSLP boards in the crate.

We assembled a stand for cooling tests as shown in Fig. 96. The front of the crate contained 9U boards with enough resistors to produce a load of 4.5 kW. Temperature sensors were mounted on these boards. We placed in the middle of the crate an old AM board (black front panel in the figure) containing 4 LAMBFTKs filled with chips on the top side to check the cooling capability of cold air flow produced by powerful fans placed below the crate.

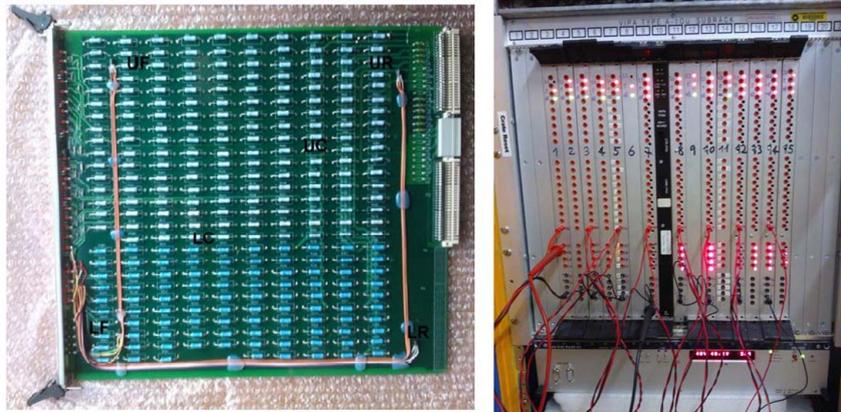


Figure 96: The test stand used for cooling tests.

Two fan packs were tested, one from Wiener and one from the CDF experiment. The results are shown in Fig. 97. The Wiener device gives generally worse results, however the tests are not yet complete. In the case of the CDF fan, we need to understand the peaks near 80°C at the front of slot 11. We also want to activate the chiller to circulate cold water in the heat exchanger above the crate. Finally we have to repeat the study with the AM board operating. Until now, the AM board has been turned on but without receiving input data, so the AM chips were consuming relatively little power due to the clock distribution. The board contains CDF AMchip03's, whose core consumption is 1.8 W, the maximum expected for AMchip06 if we mount LAMBs with chips on both sides. Moreover, the AMchip03 package thickness is large (> 3 mm), substantially decreasing the air flow. Thus the tests will be conservative. If they show the need to have fewer AM boards in a crate, we could mount 3 crates per rack with 8-12 boards in each instead of 16. We could also increase the heat exchange between the chips and the air using standard techniques if necessary.

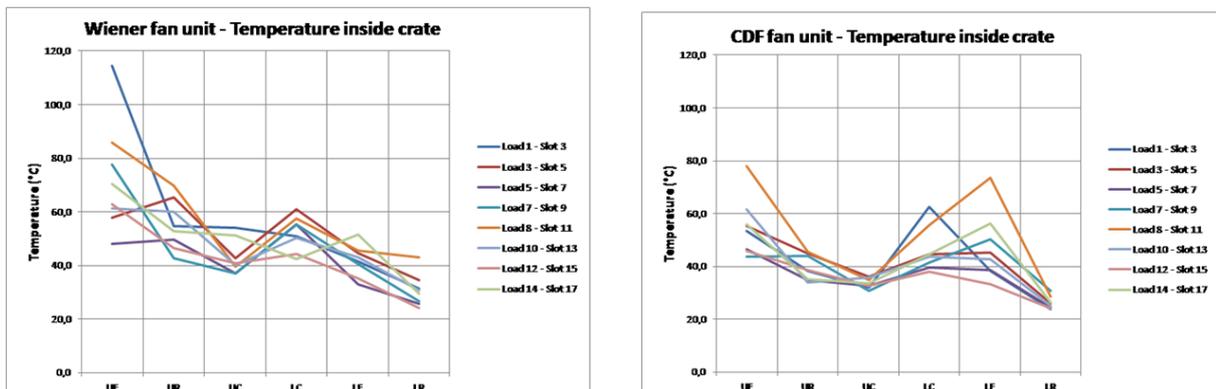


Figure 97: Results of cooling tests with the Wiener and CDF fan units. Results are shown for different slots in the crate and for different positions relative to a board - upper and lower; front, center, and rear.

## 15 Glossary

AM	Associative Memory - A content addressable memory that checks whether a prestored pattern is matched by any input hit during an event.
AM_MAP	A memory bank used in the Data Organizer in which a road address is input, and the superstrip address and width for each detector layer are returned.
AMB	AM Board - The VME board in which the pattern recognition is done.
AMchip	The ASIC content addressable chip that is the heart of the pattern recognition system.
ATCA	Advanced Telecom Computing Architecture - The board technology chosen for the FTK input and output stages. The ATCA crate is referred to as a shelf.
AUX	The VME auxiliary board that sits behind an Associative Memory Board and does the first stage track fitting.
BOC	The rear output interface card for a Read-Out Driver.
CAM	Content addressable memory.
DC	Don't Care - A feature employed in the AMchip that allows one or several bits to be ignored in the comparison between a pattern and an input hit, <i>i.e.</i> a match will be declared if the other hit bits agree with the pattern, whether or not the DC bits agree.
Constants	In FTK track fitting, the helix parameters and components of the $\chi^2$ are written as linear functions of the hit position in each detector layer. For each small region of the detector, referred to as a sector, there is a set of constants for the linear relations that are precalculated and stored in the hardware.
Coverage	The fraction of tracks with $p_T$ above 1 GeV that pass through one of the patterns in a pattern bank is the coverage of that bank.
DCR	Data Count Reset - In the Data Organizer, the DCR shows whether the hit count in a W-SS has been reset for the current event.
DF	Data Formatter - A board that holds up to 4 FTK_IM mezzanine cards, maps the pixel and SCT hits to the FTK $\eta - \phi$ tower map, and sends the hits to the appropriate AUX card and Second Stage Board.
DO	Data Organizer - A database that stores silicon hits by Super-Strip address so that they can be rapidly retrieved when an Associative Memory Board sends the address of a road.
EE	End Event - A word or packet indicating the end of a data stream for the current event.
EP	End Packet - A word indicating the end of the current data packet, containing for example a track.
FLIC	FTK Level-2 Interface Crate - Contains the boards that send the FTK tracks to the DAQ ROSSs.
fSSB	Final SSB - Carries out the final 12-layer track fit, does duplicate track removal, and sends the tracks to a FLIC board.
FTK_IM	FTK Input Mezzanine - Mounted on a Data Formatter, it receives pixel and SCT hits from the silicon RODs and does cluster finding.
HCM	Hit Count Memory - In the Data Organizer, the HCM contains the number of hits in an Super-Strip for the current event.
Hit_Map	A word sent with a road address indicating which layers had a hit in the road.
HLM	Hit List Memory - The Data Organizer memory containing the hits in the current event ordered by Super-Strip address.

HLP	Hit List Pointer - In the Data Organizer, the HLP gives the starting HLM address for hits in a Super-Strip.
HLT	High Level Trigger - The HLT consists of the level-2 trigger and Event Filter.
HOLA, dual output	The dual-output HOLA replaces the silicon ROD HOLA and provides two identical copies of the data, one to the DAQ and the other to FTK.
HOLD	A signal from a destination device to a source device telling the source to pause data transmission because the destination FIFO is almost full.
HW	Hit Warrior - Removes duplicate tracks based on shared hits.
IS	Information Service - Publishes operational statistics and status information.
JVF	Jet Vertex Fraction - For a jet, the $\Sigma p_T$ of the tracks pointing to the hard scattering vertex divided by the $\Sigma p_T$ of all the tracks in the jet.
LAMB	Local Associative Memory Board - A large mezzanine board on the Associative Memory Board that holds the AM chips.
Majority Fit	A fit of a track that does not have a hit in every layer.
MTS	Message Transporting System - Transports online messages.
Nominal Fit	A fit of a track with a hit in every layer.
Pattern	One of the track trajectories precalculated and loaded into the AM. It consists of one Super-Strip for each of the 8 layers used in pattern recognition.
Pattern bank	The set of patterns loaded into the FTK hardware.
pSSB	Preliminary SSB - Carries out the 12-layer track fit, sending the tracks to its neighboring fSSB for transmission to a FLIC card after duplicate track removal.
PU	Processor Unit - An Associative Memory Board and the AUX card behind it.
ROB	Read Out Buffer - The input buffer in a DAQ ROS.
ROD	Read Out Driver - A module that transmits data to a ROS upon a level-1 trigger.
ROS	Read Out System - It received data from a ROD for use by the HLT.
QSFP	Four-fiber optical connector.
Recovery Fit	If a Full Fit fails, the track is refit dropping one hit at a time.
Road	A pattern in which the requisite number of hits has been found.
RTM	Rear Transition Module - In an ATCA shelf, the small card behind a main board.
Sector	A small region of the detector within which the linear track fitting approximation gives good resolution. A sector consist of one physical silicon module in each detector layer.
SFP	A single-fiber optical connector.
S-LINK	A serial data transmission protocol.
SLP	Serial Link Processor - The new version of the AMB and LAMB in which all data are carried by serial links.
Spy Buffer	A circular buffer that can be frozen and read out, without stopping data flow, to inspect the data along each major pipeline stage in FTK.
SS	SuperStrip - The silicon detector width used in pattern recognition. For pixels, the width is specified in 2 dimensions.
SSB	Second Stage Board - A pSSB or fSSB.
SSMap	A memory bank in which a hit coordinate is input, and the superstrip address is returned.
TF	Track Fitter - A linearized track reconstruction.
Wild Card	To avoid inefficiency when a silicon module is disabled, the pattern recognition can treat each Super-Strip in the module as having a valid hit.
W-SS	The widest Super-Strip possible in a given layer for a variable resolution AM.
XON/XOFF	The destination-to-source flow control signal used in S-Link serial communication.

## 16 The ATLAS Collaboration

### **Argentina**

#### *Buenos Aires*

M.L. Gonzalez Silva, G. Otero y Garzon, R. Piegaia, G. Romeo

#### *La Plata*

M.J. Alconada Verzini, F. Alonso, X.S. Anduaga, M.T. Dova, F. Monticelli, M.F. Tripiana

### **Armenia**

#### *Yerevan*

H. Hakobyan

### **Australia**

#### *Adelaide*

P. Jackson, N. Soni

#### *Melbourne*

E.L. Barberio, S. Diglio, K. Hamano, D. Jennens, T. Kubota, A. Limosani, G. Nunes Hanninger, P. Rados, Q.T. Shao, K.G. Tan, G.N. Taylor, W.M. Thong, M. Volpi, M.J. White

#### *Sydney*

A. Bangert, C.W. Black, C. Cuthbert, G.-Y. Jeng, N.D. Patel, A.F. Saavedra, M. Scarcella, K.E. Varvell, I.J. Watson, B. Yabsley

### **Austria**

#### *Innsbruck*

S. Franz, P. Jussel, E. Kneringer, W. Lukas, K. Nagai, E. Ritsch, A. Usanova

### **Azerbaijan**

#### *Baku*

O. Abdinov, F. Ahmadov, N. Huseynov, F. Khalil-zada

### **Brazil**

#### *Rio de Janeiro UF*

Y. Amaral Coutinho, L.P. Caloba, C. Maidantchik, F. Marroquim, A.A. Nepomuceno, J.M. Seixas

#### *UFJF*

A.S. Cerqueira, L. Manhaes de Andrade Filho

#### *UFSJ*

M.A.B. do Vale

#### *USP*

M. Donadelli, M.A.L. Leite

### **Canada**

#### *Alberta*

A.I. Butt, K. Chan, D.M. Gingrich, R.W. Moore, J.L. Pinfeld, A. Saddique, A. Sbrizzi, HS. Subramania, F. Vives Vaque

*Carleton*

A. Bellerive, G. Cree, D. Di Valentino, T. Koffas, J. Lacey, J.F. Marchand, T.G. McCarthy, F.G. Oakham, F. Tarrade, R. Ueno, M.G. Vinciter, K. Whalen

*McGill*

C. Belanger-Champagne, B. Chapleau, S. Cheatham, F. Corriveau, R. Mantifel, S.H. Robertson, M.C. Stockton, M. Stoebe, B. Vachon, K. Wang, A. Warburton

*Montreal*

J-F. Arguin, N. Asbah, G. Azuelos, J. Bouchami, F. Dallaire, M. Davies, L. Gauthier, M. Giunta, C. Leroy, R. Rezvani, P. Soueid

*Simon Fraser Burnaby*

E. Dawe, J. Godfrey, J. Kvita, D.C. O'Neil, M. Petteni, B. Stelzer, A.J. Tanasijczuk, H. Torres, M. Trottier-McDonald, J. Van Nieuwkoop, M.C. Vetterli

*TRIUMF*

G. Azuelos, A. Canepa, S.V. Chekulaev, D. Fortin, D.M. Gingrich, A. Koutsman, F.G. Oakham, E. Perez Cordina, P. Savard, D. Schouten, R. Seuster, O. Stelzer-Chilton, R. Tafirout, I.M. Trigger, M.C. Vetterli

*Toronto*

O.S. AbouZeid, B. Brelier, T. Farooque, B. Fatholahzadeh, N. Ilic, J. Keung, P. Krieger, G. Mc Goldrick, R.S. Orr, R. Polifka, M.S. Rudolph, P. Savard, S. Schramm, P. Sinervo, T. Spreitzer, J. Taenzer, R.J. Teuscher, P.D. Thompson, W. Trischuk, N. Venturi

*Vancouver UBC*

W. Fedorko, C. Gay, Z. Gecse, S.B. King, A. Lister, C.W. Loh, W.J. Mills, S. Swedish, S. Viel

*Victoria*

J. Albert, V. Bansal, F. Berghaus, F.U. Bernlochner, L. Courneyea, C. David, M. Fincke-Keeler, R. Keeler, R. Kowalewski, M. Lefebvre, J-R. Lessard, C.P. Marino, A.C. Martyniuk, R.A. McPherson, E.A. Ouellette, J. Pearce, R. Sobie

*York*

J.A. Benitez Garcia, A.C. Florez Bustos, G. Palacino, W. Taylor

**Chile**

*Santiago*

E. Carquin, G. Cottin, M.A. Diaz

*Valparaiso*

W.K. Brooks, S. Kuleshov, R. Pezoa, F. Prokoshin, R. White

**China**

*Beijing*

Y. Bai, Y. Fang, S. Jin, F. Lu, Q. Ouyang, L.Y. Shan, J. Wang, D. Xu, L. Yao, X. Zhuang

*Hefei*

J. Gao, L. Han, Y. Jiang, B. Li, J.B. Liu, K. Liu, M. Liu, Y. Liu, H. Peng, C. Xu, L. Xu, Z. Zhao, Y. Zhu

*Nanjing*

S. Chen

*Shandong*

C. Feng, P. Ge, L.L. Ma, X. Zhang, C.G. Zhu

*Shanghai*

H. Yang

**Colombia**

*UAN Bogota*

M. Losada, L. Mendoza Navas, G. Navarro, C. Sandoval

**Czech Republic**

*Olomouc*

P. Hamal, M. Hrabovsky, L. Nozka

*Prague AS*

J. Chudoba, J. Hejbal, T. Jakoubek, O. Kepka, A. Kupco, V. Kus, M. Lokajicek, R. Lysak, M. Marcisovsky, M. Mikestikova, M. Myska, S. Nemecek, D. Roda Dos Santos, P. Ruzicka, P. Sicho, P. Staroba, M. Svatos, M. Tasevsky, T. Tic, V. Vrba

*Prague CTU*

K. Augsten, P. Gallus, J. Gunther, J. Jakubek, Z. Kohout, V. Kral, S. Pospisil, V. Simak, T. Slavicek, K. Smolek, M. Solar, J. Solc, B. Sopko, V. Sopko, M. Suk, D. Turecek, V. Vacek, M. Vlasak, P. Vokac, Z. Vykydal, M. Zeman

*Prague CU*

P. Balek, P. Berta, I. Chalupkova, T. Davidek, J. Dolejsi, Z. Dolezal, E. Fullana Torregrosa, P. Kodys, R. Leitner, J. Novakova, V. Pleskot, M. Rybar, M. Spousta, T. Sykora, P. Tas, S. Todorova-Nova, S. Valkar, V. Vorobel

**Denmark**

*Copenhagen NBI*

A. Alonso, N. Boelaert, M. Dam, G. Galster, K. Gregersen, J.B. Hansen, J.D. Hansen, J.R. Hansen, P.H. Hansen, S. Heisterkamp, S. Jakobsen, M.D. Joergensen, P. Kadlecik, A.E. Loevschall-Jensen, S. Mehlhase, J. Monk, T.C. Petersen, A. Pingel, M. Simonyan, L.A. Thomsen, C. Wiglesworth, S. Xella

**France**

*Annecy LAPP*

N. Berger, M. Delmastro, L. Di Ciaccio, T.K.O. Doan, S. Elles, C. Goy, T. Hryn'ova, S. Jézéquel, H. Keoshkerian, R. Lafaye, J. Levêque, V.P. Lombardo, N. Massol, E. Petit, H. Przysieznik, E. Sauvan, M. Schwoerer, O. Simard, T. Todorov, I. Wingerter-Seez

*Clermont - Ferrand*

D. Boumediene, E. Busato, D. Calvet, S. Calvet, J. Donini, E. Dubreuil, N. Ghodbane, Ph. Gris, C. Guicheney, H. Liao, D. Pallin, D. Paredes Hernandez, F. Podlyski, C. Santoni, T. Thevenaux-Pelzer, L. Valery, F. Vazeille

*FR-CCIN2P3 Tier-1*

G. Rahal

*Grenoble LPSC*

S. Albrand, J. Brown, Q. Buat, B. Clement, J. Collot, S. Crépe-Renaudin, B. Dechenaux, T. Delemontex, P.A. Delsart, C. Gabaldon, M.H. Genest, J-Y. Hostachy, E. Laisne, B.T. Le, F. Ledroit-Guillon, A. Lleres, A. Lucotte, F. Malek, C. Monini, J. Stark, X. Sun, B. Trocmé

### *LPNHE-Paris*

T. Beau, M. Bomben, S. Bordoni, G. Calderini, F. Crescioli, O. Davignon, S. De Cecco, A. Demilly, F. Derue, M.W. Krasny, D. Lacour, B. Laforge, S. Laplace, O. Le Dortz, G. Lefebvre, K. Liu, B. Malaescu, G. Marchiori, I. Nikolic-Audit, J. Ocariz, C. Rangel-Smith, M. Ridel, L. Roos, S. Trincaz-Duvoid, F. Vannucci

### *Marseille CPPM*

L. Alio, M. Barbero, C.P. Bee, C. Bertella, N. Bousson, J.C. Clemens, Y. Coadou, F. Djama, L. Feligioni, J. Gao, D. Hoffmann, F. Hubaut, E.B.F.G. Knoops, E. Le Guirriec, B. Li, J. Maurer, E. Monnier, Y. Nagai, J. Odier, P. Pralavorio, A. Rozanov, T. Serre, M. Talby, N. Tannoury, E. Tiouchichine, S. Tisserant, J. Toth, F. Touchard, M. Ughetto, L. Vacavant

### *Orsay LAL*

S. Abdel Khalek, A. Bassalat, S. Binet, C. Bourdarios, J.B. De Vivie De Regie, L. Duflot, M. Escalier, L. Fayard, D. Fournier, J.-F. Grivaz, T. Guillemain, S. Henrot-Versille, J. Hrivnac, L. Iconomidou-Fayard, J. Idarraga, M. Kado, N. Lorenzo Martinez, A. Lounis, N. Makovec, L. Poggioli, P. Puzo, A. Renaud, D. Rousseau, G. Rybkin, J.B. Sauvan, J. Schaarschmidt, A.C. Schaffer, E. Scifo, L. Serin, S. Simion, R. Tanaka, H.L. Tran, D. Zerwas, Z. Zhang

### *Saclay CEA*

H. Abreu, H. Bachacou, F. Balli, F. Bauer, N. Besson, J.-B. Blanchard, N.M. Bolnet, M. Boonekamp, L. Chevalier, F. Deliot, J. Ernwein, A.I. Etienne, A. Formica, P.F. Giraud, H.M.X. Grabas, C. Guyot, S. Hassani, W. Kozanecki, E. Lançon, J.F. Laporte, C. Maiani, P. Mal, J.A. Manjarres Ramos, B. Mansoulie, H. Martinez, N. Meric, J.-P. Meyer, L. Mijović, E. Mountricha, V. Nguyen Thi Hong, R. Nicolaidou, A. Ouraou, E. Protopapadaki, B. Resende, C.R. Royon, L. Schoeffel, Ph. Schune, Ph. Schwemling, J. Schwindling, D. Tsiou, N. Vranjes, M. Xiao, C. Xu

## **Georgia**

### *Tbilisi IP*

E.G. Tskhadadze

### *Tbilisi SU*

T. Djobava, J. Khubua, G. Mchedlidze, M. Mosidze

## **Germany**

### *Berlin HU*

M. Aliev, E. Bergeas Kuutmann, F.M. Giorgi, S. Grancagnolo, G.H. Herbert, R. Herrberg-Schubert, I. Hristova, O. Kind, H. Kolanoski, R. Kwee, H. Lacker, M. Leyton, T. Lohse, A. Nikiforov, P. Rieck, H. Schulz, D. Wendland, M. zur Nedden

### *Bonn*

T. Abajyan, O. Arslan, M. Backhaus, P. Bechtle, I. Brock, M. Cristinziani, W. Davey, K. Desch, J. Dingfelder, W. Ehrenfeld, G. Gaycken, Ch. Geich-Gimbel, J. Glatzer, L. Gonella, P. Haefner, S. Hageboeck, M. Havranek, D. Hellmich, S. Hillert, F. Huegging, J. Janssen, G. Khoriani, P. Koevesarki, V.V. Kostyukhin, J.K. Kraus, J. Kroseberg, H. Krüger, C. Lapoire, M. Lehmann, A.M. Leyko, J. Liebal, C. Limbach, T. Loddenkoetter, S. Mergelmeyer, K. Mueller, G. Nanava, T. Nattermann, A.-E. Nuncio-Quiroz, D. Pohl, S. Psoroulas, B. Sarrazin, S. Schaepe, M.J. Schultens, T. Schwandt, F. Scutti, J.A. Stillings, J. Therhaag, J.-W. Tsung, K. Uchida, M. Uhlenbrock, P. Urquijo, A. Vogel, E. von Toerne, P. Wagner, T. Wang, N. Wermes, P. Wienemann, L.A.M. Wiik-Fuchs, K.H. Yau Wong, R. Zimmermann, S. Zimmermann

### *DESY*

S. Argyropoulos, I. Bloch, S. Borroni, J.A. Dassoulas, J. Dietrich, V. Ferrara, M. Filipuzzi, C. Friedrich,

A. Glazov, L.S. Gomez Fajardo, J. Goncalves Pinto Firmino Da Costa, K-J. Grahn, I.M. Gregor, A. Grohsjean, K.H. Hiller, A. Huettmann, M. Jimenez Belenguier, J. Katzy, T. Kono, T. Kuhl, C. Lange, M. Lisovyi, E. Lobodzinska, D. Ludwig, M. Medinnis, S. Mättig, K. Mönig, T. Naumann, T. Perez Cavalcanti, R.F.Y. Peters, S.M. Piec, V. Radescu, I. Rubinskiy, G. Sedov, S. Shushkevich, D. South, M. Stanescu-Bellu, M.M. Stanitzki, P. Starovoitov, N.A. Styles, K. Tackmann, P. Vankov, M. Viti, C. Wasicki, M.A. Wildt, E. Yatsenko, E. Yildirim, H. Zhu

#### *Dortmund*

M. Bunse, I. Burmeister, H. Esch, C. Gössling, J. Jentzsch, C.A. Jung, R. Klingenberg, I. Reisinger, T. Wittig

#### *Dresden*

P. Anger, P. Czodrowski, F. Friedrich, J.P. Grohs, C. Gumpert, M. Kobel, K. Leonhardt, W.F. Mader, M. Morgenstern, X. Prudent, C. Rudolph, U. Schnoor, F. Seifert, P. Steinbach, A. Straessner, A. Vest, S. Wahrmond

#### *Freiburg*

G. Aad, S. Amoroso, T. Barber, R. Bernhard, M. Boehler, R. Bruneliere, F. Buehrer, V. Consorti, A. Di Simone, M. Fehling-Kaschek, M. Flechl, C. Giuliani, G. Hertel, K. Jakobs, P. Jenni, A.K. Kopp, S. Kuehn, K. Köneke, S. Lai, U. Landgraf, K. Lohwasser, R. Madar, K. Mahboubi, U. Parzefall, M. Rammensee, T.C. Rave, Z. Rurikova, N. Ruthmann, C. Schillo, E. Schmidt, M. Schumacher, F. Siegert, K. Storig, J.E. Sundermann, K.K. Temming, S. Thoma, V. Tsiskaridze, F.C. Ungaro, M. Venturi, I. Vivarelli, H. von Radziewski, T. Vu Anh, C. Weiser, M. Werner, S. Winkelmann, S. Zimmermann

#### *Giessen*

M. Düren, K. Kreutzfeldt, H. Stenzel

#### *Goettingen*

K. Bierwagen, U. Blumenschein, O. Brandt, D. Evangelakou, M. George, L. Graber, J. Grosse-Knetter, M. Hamer, C. Hensel, G. Kawamura, M. Keil, A. Knue, F. Kohn, N. Krieger, K. Kroeninger, B. Lemmer, E. Magradze, J. Meyer, J. Morel, O. Nackenhorst, S. Pashapour, R.F.Y. Peters, A. Quadt, A. Roe, A.L.S. Schorlemmer, L. Serkin, E. Shabalina, T. Vazquez Schroeder, J. Weingarten

#### *Heidelberg KIP*

G. Anders, V. Andrei, Y. Davygora, T.A. Dietzsch, M. Dunford, P. Hanke, J.I. Hofmann, E.-E. Kluge, H. Laier, V.S. Lang, K. Meier, F. Mueller, S. Poddar, V. Scharf, H.-C. Schultz-Coulon, R. Stamen, M. Wessels

#### *Heidelberg PI*

C.F. Anders, M. Giuliani, G. Kasieczka, T. Klimkovich, R. Narayan, S. Schaetzel, S. Schmitt, A. Schoening, H.K. Soltveit

#### *Heidelberg ZITI*

T. Colombo, A. Kugel, I. Sacco, N. Schroer

#### *Mainz*

O. Arnaez, W. Blum, V. Büscher, R. Caputo, F. Ellinghaus, O.C. Endner, E. Ertel, F. Fiedler, C. Goring, C. Handel, T. Heck, M. Hohlfeld, P.J. Hsu, T.A. Hülsing, W. Ji, M. Karnevskiy, K. Kleinknecht, S. Koenig, L. Köpke, M. Lungwitz, L. Masetti, J. Mattmann, C. Meyer, D. Moreno, S. Moritz, T. Mueller, A. Neusiedl, R. Poettgen, H.G. Sander, C. Schmitt, M. Schott, C. Schroeder, N. Schuh, U. Schäfer, E. Simioni, S. Tapprogge, S.J. Wollstadt, C. Zimmermann

#### *Munich LMU*

S. Adomeit, S. Beale, S. Becker, O. Biebel, J. Bortfeldt, P. Calfayan, B.K.B. Chow, J. de Graat, G. Duckeck, J. Ebke, J. Elmsheuser, A. Engl, C. Galea, C. Heller, R. Hertenberg, F. Legger, J. Lorenz, A. Mann,

C. Meineck, T. Nunnemann, L.B. Oakes, F. Rauscher, P. Reznicek, A. Ruschke, M.P. Sanders, D. Schaile, J. Schieck, C. Schmitt, D. Vladoiu, R. Walker, J.Z. Will, J. Wittkowski, A. Zibell

#### *Munich MPI*

T. Barillari, S. Bethke, B. Bittner, J. Bronner, G. Compostella, G. Cortiana, M.J. Flowerdew, P. Giovannini, M. Goblirsch-kolb, T. Ince, A.E. Kiryunin, S. Kluth, O. Kortner, S. Kortner, H. Kroha, A. Macchiodo, A. Manfredini, S. Menke, H.G. Moser, M. Nagel, R. Nisius, H. Oberlack, C. Pahl, R. Richter, D. Salihagic, R. Sandstroem, P. Schacht, Ph. Schwegler, F. Sforza, S. Stern, S. Stonjek, S. Terzo, M. Vanadia, H. von der Schmitt, P. Weigell, A. Wildauer, D. Zanzi

#### *Siegen*

N.B. Atlay, P. Buchholz, H. Czirr, I. Fleck, B. Gaur, K. Grybel, I. Ibragimov, K. Ikematsu, M. Rammes, O. Rosenthal, V. Sipica, W. Walkowiak, M. Ziolkowski

#### *Wuerzburg*

P. Fleischmann, A. Redelbach, M. Schreyer, G. Siragusa, R. Ströhmer, J.Y.C. Tam, T. Trefzger, S.W. Weber

#### *Wuppertal*

M. Barisonzi, K. Becker, T.A. Beermann, J. Boek, T.T. Boek, T. Cornelissen, D. Duda, G. Ernis, J. Fischer, S. Fleischmann, T. Flick, K. Hamacher, T. Harenberg, D. Hirschbuehl, S. Kalinin, S. Kersten, A. Khoroshilov, S. Kohlmann, M. Mechtel, P. Mättig, M. Neumann, S. Patariaia, M. Sandhoff, G. Sartisohn, P. Sturm, W. Wagner, C. Zeitnitz

### **Greece**

#### *Athens*

S. Angelidakis, A. Antonaki, S. Chouridou, D. Fassouliotis, N. Giokaris, P. Ioannou, K. Iordanidou, C. Kourkoumelis, A. Manousakis-Katsikakis

#### *Athens NTU*

T. Alexopoulos, M. Byszewski, M. Dris, E.N. Gazis, G. Iakovidis, K. Karakostas, N Karastathis, S. Leontsinis, S. Maltezos, E. Mountricha, K. Ntekas, E. Panagiotopoulou, Th.D. Papadopoulou, G. Tsiopolitis, S. Vlachos

#### *Thessaloniki*

K. Bachas, C. Gentsos, I. Gkialas, D. Iliadis, K. Kordas, V. Kouskoura, S. Nikolaidis, I. Nomidis, K. Pappageorgiou, C. Petridou, D. Sampsonidis, O. Sidiropoulou, K.L. Sotiropoulou

### **Israel**

#### *Technion Haifa*

A. Di Mattia, R. Kopeliansky, E. Musto, Y. Rozen, S. Tarem

#### *Tel-Aviv*

H. Abramowicz, G. Alexander, N. Amram, G. Bella, O. Benary, Y. Benhammou, E. Etzion, A. Gershon, O. Gueta, N. Guttman, Y. Munwes, Y. Oren, I. Sadeh, Y. Silver, A. Soffer, N. Taiblum

#### *Weizmann Rehovot*

R. Alon, L. Barak, S. Bressler, Z.H. Citron, E. Duchovni, O. Gabizon, E. Gross, J. Groth-Jensen, D. Lelouch, L.J. Levinson, G. Mikenberg, A. Milov, D. Milstein, I. Roth, O. Silbert, V. Smakhtin, O. Vitells

### **Italy**

#### *Bologna*

L. Bellagamba, M. Bindi, D. Boscherini, A. Bruni, G. Bruni, M. Bruschi, D. Caforio, M. Corradi,

S. De Castro, R. Di Sipio, L. Fabbri, M. Franchini, A. Gabrielli, B. Giacobbe, F. Giannuzzi, F.M. Giorgi, P. Grafström, M.K. Jha, F. Lasagni Manghi, I. Massa, A. Mengarelli, S. Monzani, M. Negrini, M. Piccinini, A. Polini, L. Rinaldi, M. Romano, C. Sbarra, N. Semprini-Cesari, R. Spighi, S.A. Tupputi, S. Valentinetti, M. Villa, A. Zoccoli

#### *Cosenza*

M. Capua, G. Crosetti, L. La Rotonda, V. Lavorini, A. Mastroberardino, A. Policicchio, D. Salvatore, M. Schioppa, E. Tassi

#### *Frascati*

A. Annovi, M. Antonelli, M. Beretta, H. Bilokon, A. Castegnaro, V. Chiarella, M. Curatolo, R. Di Nardo, B. Esposito, M. Gatta, C. Gatti, P. Laurelli, G. Maccarrone, A. Sansoni, M. Testa, E. Vilucchi, G. Volpi

#### *Genova*

D. Barberis, G. Darbo, A. Favareto, A. Ferretto Parodi, G. Gagliardi, C. Gemme, E. Guido, P. Morettini, B. Osculati, F. Parodi, S. Passaggio, L.P. Rossi, C. Schiavi

#### *Lecce*

G. Chiadini, E. Gorini, N. Orlando, M. Primavera, S. Spagnolo, A. Ventura

#### *Milano*

G. Alimonti, A. Andreani, A. Andreazza, M.I. Besana, L. Carminati, D. Cavalli, S.M. Consonni, G. Costa, M. Fanti, D. Giugni, I. Koletsou, T. Lari, V. Liberali, L. Mandelli, F. Meloni, C. Meroni, L. Perini, C. Pizio, F. Ragusa, S. Resconi, S. Shojaii, R. Simoniello, A. Stabile, G.F. Tartarelli, C. Troncon, R. Turra

#### *Napoli*

A. Aloisio, M.G. Alviggi, V. Canale, G. Carlino, G. Chiefari, F. Conventi, R. de Asmundis, M. Della Pietra, D. della Volpe, C. Di Donato, A. Doria, R. Giordano, P. Iengo, V. Izzo, L. Merola, S. Patricelli, E. Rossi, A. Sanchez, G. Sekhniaidze, G. Zurzolo

#### *Pavia*

R. Ferrari, M. Fraternali, G. Gaudio, M. Livan, A. Negri, G. Polesello, D.M. Rebuzzi, A. Rimoldi, V. Vercesi

#### *Pisa*

R. Beccherle, F. Bertolucci, V. Bevacqua, V. Cavasinni, R. Cipriani, S. Citraro, M. Dell'Orso, T. Del Prete, S. Donati, P. Giannetti, P. Luciano, C. Luongo, D. Magalotti, M. Piendibene, C. Roda, Z. Zinonos

#### *Roma I*

F. Anulli, G. Artoni, P. Bagiacchi, P. Bagnaia, C. Bini, A. D'Orazio, D. De Pedis, A. De Salvo, C. Dionisi, S. Falciano, A. Gabrielli, S. Gentile, S. Giagu, V. Ippolito, M. Kuna, F. Lacava, F. Lo Sterzo, C. Luci, L. Luminari, A. Messina, A. Nisati, E. Pasqualucci, E. Petrolo, L. Pontecorvo, M. Rescigno, S. Rosati, F. Safai Tehrani, A. Sidoti, E. Solfaroli Camillocci, R. Vari, S. Veneziano

#### *Roma II*

G. Aielli, R. Cardarelli, G. Cattani, A. Di Ciaccio, B. Liberti, F. Marchese, L. Mazzaferro, A. Salamon, R. Santonico

#### *Roma Tre*

A. Baroncelli, M. Biglietti, V. Bortolotto, F. Ceradini, B. Di Micco, A. Farilla, E. Graziani, M. Iodice, D. Orestano, F. Petrucci, C. Stanesco, M. Trovatelli

#### *Trieste ICTP*

B.S. Acharya

*Udine*

M. Alhroob, S.F. Brazzale, M. Cobal, U. De Sanctis, M.P. Giordani, M. Pinamonti, K. Shaw, R. Soualah

*Udine cluster*

B.S. Acharya, M. Alhroob, S.F. Brazzale, M. Cobal, U. De Sanctis, M. Pinamonti, K. Shaw, R. Soualah

**Japan**

*Hiroshima IT*

Y. Nagasaka

*KEK*

Y. Arai, Y. Ikegami, M. Ikeno, H. Iwasaki, J. Kanzaki, T. Kohriki, T. Kondo, Y. Makida, S. Mitsui, K. Nagano, K. Nakamura, M. Nozaki, S. Odaka, O. Sasaki, Y. Suzuki, Y. Takubo, S. Tanaka, S. Terada, K. Tokushuku, S. Tsuno, Y. Unno, M. Yamada, A. Yamamoto, Y. Yasu

*Kobe*

Y. Inamaru, M. King, T. Kishimoto, T. Kitamura, H. Kurashige, R. Kurumida, T. Matsushita, A. Ochi, S. Shimizu, H. Takeda, K. Tani, I. Watanabe, Y. Yamazaki, L. Yuan

*Kyoto*

M. Ishino, T. Sumida, T. Tashiro

*Kyoto UE*

R. Takashima

*Kyushu*

K. Kawagoe, S. Oda, H. Otono, J. Tojo

*Nagasaki*

M. Shimojima

*Nagoya*

M. Aoki, S. Hasegawa, L. Morvaj, T. Ohshima, Y. Takahashi, M. Tomoto, J. Wakabayashi, K. Yamauchi

*Okayama*

I. Nakano

*Osaka*

K. Hanagaki, M. Hirose, J.S.H. Lee, M. Nomachi, W. Okamura, Y. Sugaya

*Shinshu*

Y. Hasegawa, T. Takeshita

*Tokyo ICEPP*

G. Akimoto, S. Asai, Y. Azuma, T. Dohmae, Y. Enari, K. Hanawa, N. Kanaya, Y. Kataoka, T. Kawamoto, S. Kazama, K. Kessoku, T. Kobayashi, Y. Komori, T. Mashimo, T. Masubuchi, H. Matsunaga, T. Nakamura, Y. Ninomiya, T. Okuyama, H. Sakamoto, Y. Sasaki, J. Tanaka, K. Terashi, I. Ueda, H. Yamaguchi, Y. Yamaguchi, S. Yamamoto, T. Yamamura, T. Yamanaka, K. Yoshihara

*Tokyo MU*

U. Bratzler, C. Fukunaga

*Tokyo Tech*

M. Ishitsuka, O. Jinnouchi, T. Kanno, M. Kuze, R. Nagai, T. Nobe

*Tsukuba*

K. Hara, T. Hayashi, S.H. Kim, K. Kiuchi, M. Kurata, F. Ukegawa

*Waseda*

T. Iizawa, N. Kimura, T. Korikawa, T. Mitani, Y. Sakurai, K. Yorita

**Morocco**

*CNESTEN Rabat*

*Casablanca*

D. Benchekroun, A. Chafaq, M. Gouighri, A. Hoummada, S. Lablak

*Marrakesh*

M. El Kacimi, D. Goujdami

*Oujda*

S. Boutouil, J.E. Derkaoui, M. Ouchrif, Y. Tayalati

*Rabat*

R. Cherkaoui El Moursli

**Netherlands**

*Nijmegen*

G.J. Besjes, S. Caron, V. Dao, N. De Groot, F. Filthaut, P.F. Klok, A.C. König, A. Salvucci

*Nikhef*

R. Aben, L.J. Beemster, S. Bentvelsen, E. Berglund, G.J. Bobbink, K. Bos, H. Boterenbrood, A. Castelli, A.P. Colijn, P. de Jong, C. Deluca, L. De Nooij, P.O. Deviveiros, S. Dhaliwal, P. Ferrari, S. Gadatsch, D.A.A. Geerts, F. Hartjes, N.P. Hessey, N. Hod, O. Igonkina, P. Kluit, E. Koffeman, H. Lee, T. Lenz, F. Linde, J. Mahlstedt, J. Mechnich, I. Mussche, J.P. Ottersbach, K.P. Oussoren, P. Pani, N. Ruckstuhl, N. Valencic, P.C. Van Der Deijl, R. van der Geer, H. van der Graaf, R. Van Der Leeuw, I. van Vulpen, W. Verkerke, J.C. Vermeulen, M. Vranjes Milosavljevic, M. Vreeswijk, H. Weits

**Norway**

*Bergen*

T. Buanes, T. Burgess, G. Eigen, A. Kastanas, W. Liebig, A. Lipniacka, P.L. Rosendahl, H. Sandaker, T.B. Sjursen, B. Stugu, M. Ugland

*Oslo*

L. Bugge, D. Cameron, B.K. Gjelsten, E. Gramstad, F. Ould-Saada, K. Pajchel, M. Pedersen, A.L. Read, O. Røhne, L. Smestad, S. Stapnes, A. Strandlie

**Poland**

*Cracow AGH-UST*

L. Adamczyk, T. Bold, W. Dabrowski, M. Dwuznik, I. Grabowska-Bold, D. Kisielewska, S. Koperny, T.Z. Kowalski, B. Mindur, M. Przybycien

*Cracow IFJ PAN*

E. Banas, P.A. Bruckman de Renstrom, D. Derendarz, E. Gornicki, W. Iwanski, A. Kaczmarska, K. Korcyl, Pa. Malecki, A. Olszewski, J. Olszowska, E. Stanecka, R. Staszewski, M. Trzebinski, A. Trzupek, M.W. Wolter, B.K. Wosiek, K.W. Wozniak, B. Zabinski, A. Zemla

*Jagiellonian*

## **Portugal**

### *Portugal LIP*

J.A. Aguilár-Saavedra, S.P. Amor Dos Santos, A. Amorim, N. Anjos, J. Carvalho, N.F. Castro, P. Conde Muiño, M.J. Da Cunha Sargedas De Sousa, M.C.N. Fiolhais, B. Galhardo, A. Gomes, P.M. Jorge, L. Lopes, J. Machado Miguens, A. Maio, J. Maneira, M. Oliveira, A. Onofre, A. Palma, B. Pinto, H. Santos, J.G. Saraiva, J. Silva, A. Tavares Delgado, F. Veloso, H. Wolters

## **Republic of Belarus**

### *Minsk AC*

S. Harkusha, Y. Kulchitsky, Y.A. Kurochkin, P.V. Tsiarshka

### *Minsk NC*

S. Yanush

## **Romania**

### *Bucharest IFIN-HH*

C. Alexa, E. Badescu, S.I. Buda, I. Caprini, M. Caprini, A. Chitan, M. Ciubancan, S. Constantinescu, C.-M. Cuciuc, P. Dita, S. Dita, A. Olariu, D. Pantea, M. Rotaru, G. Stoicea, A. Tudorache, V. Tudorache

### *ITIM*

G.A. Popeneciu

### *Politehnica Bucharest*

G.L. Darlea

### *Timisoara WU*

## **Russia**

### *JINR Dubna*

F. Ahmadov, I.N. Aleksandrov, V.A. Bednyakov, I.R. Boyko, I.A. Budagov, G.A. Chelkov, A. Chep-lakov, M.V. Chizhov, D.V. Dedovich, M. Demichev, G.L. Glonti, M.I. Gostkin, N. Huseynov, S.N. Kar-pov, M.Y. Kazarinov, D. Kharchenko, E. Khramov, V.M. Kotov, U. Kruchonak, Z.V. Krumshteyn, V. Kukhtin, E. Ladygin, I.A. Minashvili, M. Mineev, V.D. Peshekhonov, E. Plotnikova, I.N. Potrap, V. Pozdnyakov, N.A. Rusakovich, R. Sadykov, A. Saprionov, M. Shiyakova, V.B. Vinogradov, A. Zhem-chugov, N.I. Zimin

### *Moscow FIAN*

A.V. Akimov, I.L. Gavrilenko, A.A. Komar, R. Mashinistov, P.Yu. Nechaeva, A. Shmeleva, A.A. Sne-sarev, V.V. Sulin, V.O. Tikhomirov

### *Moscow ITEP*

A. Artamonov, P.A. Gorbounov, V. Khovanskiy, P.B. Shatalov, I.I. Tsukerman

### *Moscow MEPhI*

A. Antonov, K. Belotskiy, O. Bulekov, V.A. Kantserov, A. Khodinov, A. Romaniouk, E. Shulga, S.Yu. Smirnov, Y. Smirnov, E.Yu. Soldatov, S. Timoshenko

### *Moscow SU*

A.S. Boldyrev, L.K. Gladilin, Y.V. Grishkevich, V.A. Kramarenko, V.I. Rud, S.Yu. Sivoklov, L.N. Smirnova, S. Turchikhin

### *Novosibirsk BINP*

A.V. Anisenkov, O.L. Beloborodova, V.S. Bobrovnikov, A.G. Bogdanchikov, V.F. Kazanin, A.A. Korol,

V.M. Malyshev, A.L. Maslennikov, D.A. Maximov, S.V. Peleganchuk, K.Yu. Skovpen, A.M. Soukharev, A.A. Talyshev, Yu.A. Tikhonov

*Petersburg NPI*

O.L. Fedin, V. Gratchev, O.G. Grebenyuk, V.P. Maleev, Y.F. Ryabov, V.A. Schegelsky, E. Sedykh, D.M. Seliverstov, V. Solovyev

*Protvino IHEP*

A. Borisov, S.P. Denisov, R.M. Fakhruddinov, A.B. Fenyuk, D. Golubkov, A.V. Ivashin, A.N. Karyukhin, V.A. Korotkov, A.S. Kozhin, A.A. Minaenko, A.G. Myagkov, V. Nikolaenko, A.A. Solodkov, O.V. Solovyanov, E.A. Starchenko, A.M. Zaitsev, O. Zenin

**Serbia**

*Belgrade IP*

J. Krstic, D.S. Popovic, Dj. Sijacki, Lj. Simic

*Belgrade Vinca*

T. Agatonovic-Jovin, I. Bozovic-Jelisavcic, P. Cirkovic, J. Mamuzic

**Slovak Republic**

*Bratislava*

R. Astalos, L. Batkova, T. Blazek, P. Federic, I. Sykora, S. Tokár, T. Ženiš

*Kosice*

J. Antos, D. Bruncko, J. Ferencei, E. Kladiva, P. Strizenec

**Slovenia**

*Ljubljana*

V. Cindro, M. Deliyergiyev, A. Filipčič, A. Gorišek, B.P. Kerševan, G. Kramberger, I. Mandić, B. Maček, M. Mikuž, A. Tykhonov

**South Africa**

*Cape Town*

A. Hamilton

*Johannesburg*

M. Arousseau, E. Castaneda-Miranda, S. Yacoob

*Witwatersrand*

G.D. Carrillo-Montoya, Y. Huang, K.J.C. Leney, B.R. Mellado Garcia, W.B. Quayle, X. Ruan, O.E. Vickey Boeriu, T. Vickey

**Spain**

*Barcelona*

J. Abdallah, M. Bosman, R. Caminal Armadans, M.P. Casado, M. Cavalli-Sforza, M.C. Conidi, A. Cortes-Gonzalez, X. Espinal Curull, P. Francavilla, V. Giangiobbe, G. Gonzalez Parra, S. Grinstein, A. Juste Rozas, I. Korolkov, E. Le Menedeu, M. Martinez, L.M. Mir, J. Montejo Berlingen, J. Nadal, A. Pacheco Pages, C. Padilla Aranda, X. Portell Bueso, I. Riu, V. Rossetti, F. Rubbo, A. Succurro, S. Tsiskaridze

*Madrid UA*

V. Arnal, F. Barreiro, J. Cantero, H. De la Torre, J. Del Peso, C. Glasman, J. Llorente Merino, J. Terron

*Spain Granada*

J.A. Aguilar-Saavedra

## *Valencia*

S. Cabrera Urbán, V. Castillo Gimenez, M.J. Costa, F. Fassi, A. Ferrer, L. Fiorini, J. Fuster, C. García, J.E. García Navarro, S. González de la Hoz, Y. Hernández Jiménez, E. Higón-Rodríguez, A. Irlés Quiles, M. Kaci, C. Lacasta, V.R. Lacuesta, L. March, S. Marti-Garcia, V.A. Mitsou, M. Miñano Moya, R. Moles-Valls, M. Moreno Llácer, E. Oliver Garcia, S. Pedraza Lopez, M.T. Pérez García-Estañ, E. Romero Adam, E. Ros, J. Salt, V. Sanchez Martinez, U. Soldevila, J. Sánchez, E. Torró Pastor, A. Valero, E. Valladolid Gallego, J.A. Valls Ferrer, M. Villaplana Perez, M. Vos

## **Sweden**

### *Lund*

S.S. Bocchetta, L. Bryngemark, A. Floderus, A.D. Hawkins, V. Hedberg, G. Jarlskog, E. Lytken, B. Meirose, J.U. Mjörnmark, O. Smirnova, O. Viazlo, M. Wielers, T.P.A. Åkesson

### *Stockholm*

Y. Abulaiti, K. Bendtz, O. Bessidskaia, C. Bohm, C. Clement, K. Gellerstedt, S. Hellman, K.E. Johansson, K. Jon-And, H. Khandanyan, H. Kim, P. Klimek, J. Lundberg, O. Lundberg, D.A. Milstead, T. Moa, S. Molander, A. Papadelis, A. Petridis, P. Plucinski, S.B. Silverstein, J. Sjölin, S. Strandberg, M. Tylmad, B. Åsman

### *Stockholm KTH*

J. Jovicevic, E.S. Kuwertz, B. Lund-Jensen, A.K. Morley, J. Strandberg

### *Uppsala*

R. Brenner, C.P. Buszello, E. Coniavitis, T. Ekelof, M. Ellert, A. Ferrari, C. Isaksson, A. Madsen, D. Pelikan

## **Switzerland**

### *Bern*

M. Agustoni, L.S. Ancu, H.P. Beck, C. Borer, A. Cervelli, A. Ereditato, V. Gallo, S. Haug, T. Kruker, L.F. Marti, B. Schneider, F.G. Sciacca, M.S. Weber

### *CERN*

M. Aleksa, C. Anastopoulos, N. Andari, G. Avolio, M.A. Baak, M. Backes, D. Banfi, M. Battistin, M. Bellomo, O. Beltramello, D. Berge, M. Bianco, J. Boyd, H. Burckhart, S. Campana, M.D.M. Capeans Garrido, T. Carli, A. Catinaccio, J.R. Catmore, A. Cattai, C.A. Chavez Barajas, J.T. Childers, D. Chromek-Burckhart, A. Dell'Acqua, A. Di Girolamo, B. Di Girolamo, F. Dittus, D. Dobos, J. Dopke, A. Dudarev, M. Dührssen, N. Ellis, M. Elsing, G. Facini, P. Farthouat, P. Fassnacht, S. Franchino, D. Francis, D. Froidevaux, V. Garonne, F. Gianotti, D. Gillberg, J. Godlewski, L. Goossens, B. Gorini, H.M. Gray, M. Hauschild, R.J. Hawkings, M. Heller, C. Hensens, A.M. Henriques Correia, L. Hervas, A. Hoecker, Z. Hubacek, M. Huhtinen, M.R. Jaekel, H. Jansen, P. Jenni, R.M. Jungst, M. Kaneda, T. Klioutchnikova, K. Lantsch, M. Lassnig, G. Lehmann Miotto, B. Lenzi, D. Macina, S. Malyukov, L. Mapelli, B. Martin, A. Messina, J. Meyer, S. Michal, A. Molfetas, G. Mornacchi, A.M. Nairz, Y. Nakahama, G. Negri, M. Nessi, M. Nordberg, C.C. Ohm, S. Palestini, T. Pauly, H. Pernegger, B.A. Petersen, K. Peters, K. Pommès, A. Poppleton, G. Poulard, S. Prasad, M. Raymond, C. Rembser, L. Rodrigues, S. Roe, D. Salek, A. Salzburger, D.O. Savu, T. Scanlon, S. Schlenker, K. Schmieden, C. Serfon, A. Sfyrla, C.A. Solans, G. Spigo, G.A. Stewart, F.A. Teischinger, H. Ten Kate, L. Tremblet, A. Tricoli, C. Tsarouchas, G. Unal, W. Vandelli, D. van der Ster, N. van Eldik, R. Vigne, R. Voss, R. Vuillermet, P.S. Wells, T. Wengler, S. Wenig, P. Werner, H.G. Wilkens, F. Winklmeier, J. Wotschack, L. Zwalinski

### *Geneva*

G. Alexandre, G. Barone, P.J. Bell, W.H. Bell, E. Benhar Nocchioli, J. Bilbao De Mendizabal, F. Bucci, R. Camacho Toro, A. Clark, C. Doglioni, D. Ferrere, S. Gadomski, S. Gonzalez-Sevilla, M.P. Goulette,

J.L. Gramling, F. Guescini, G. Iacobucci, A. Katre, A. La Rosa, B. Martin dit Latour, P. Mermod, C. Mora Herrera, D. Muenstermann, S. Nektarijevic, M. Nessi, K. Nikolics, A. Picazio, M. Pohl, G. Pásztor, K. Rosbach, S. Vallecorsa, X. Wu

## **Taiwan**

### *Taipei AS*

S. Hou, D.O. Jamin, C.A. Lee, S.C. Lee, B. Li, S.C. Lin, B. Liu, D. Liu, R. Mazini, D.A. Soh, P.K. Teng, J. Wang, S.M. Wang, Z. Weng, L. Zhang

## **Turkey**

### *Ankara*

O. Cakir, A.K. Ciftci, R. Ciftci, H. Duran Yildiz, S. Kuday

### *Bogazici Istanbul*

M. Arik, S. Istin, V.E. Ozcan

### *Dogus*

S.A. Cetin

### *Gazi*

M. Yilmaz

### *Gaziantep*

A. Beddall, A.J. Beddall, A. Bingul

### *TOBB*

S. Sultansoy

### *Turkey Atomic EA*

## **United Kingdom**

### *Birmingham*

B.M.M. Allbrooke, L. Aperio Bella, H.S. Bansil, J. Bracinik, D.G. Charlton, A.S. Chisholm, A.C. Daniells, C.M. Hawkes, S.J. Head, S.J. Hillier, T. McLaughlan, R.D. Mudd, J.A. Murillo Quijada, P.R. Newman, K. Nikolopoulos, J.D. Palmer, M. Slater, J.P. Thomas, P.D. Thompson, P.M. Watkins, A.T. Watson, M.F. Watson, J.A. Wilson

### *Cambridge*

S. Ask, N. Barlow, J.R. Batley, F.M. Brochu, W. Buttinger, J.R. Carter, J.D. Chapman, S.T. French, J.A. Frost, T.P.S. Gillam, J.C. Hill, S. Kaneti, T.J. Khoo, C.G. Lester, V. Moeller, T. Mueller, M.A. Parker, D. Robinson, T. Sandoval, M. Thomson, C.P. Ward, S. Williams

### *Edinburgh*

W. Bhimji, T.M. Bristow, A.G. Buckley, P.J. Clark, C. Debenedetti, N.C. Edwards, F.M. Garay Walls, R.D. Harrington, A. Korn, V.J. Martin, B.J. O'Brien, S.A. Olivares Pino, M. Proissl, A. Schaelicke, K.E. Selbach, B.H. Smart, A. Washbrook, B.M. Wynne

### *Glasgow*

S.E. Allwood-Spiers, R.L. Bates, D. Britton, P. Bussey, C.M. Buttar, A. Buzatu, C. Collins-Tooth, S. D'Auria, T. Doherty, A.T. Doyle, S. Ferrag, J. Ferrando, D.E. Ferreira de Lima, A. Gemmell, U. Gul, N.G. Gutierrez Ortiz, D. Kar, A. Moraes, V. O'Shea, C. Oropeza Barrera, D. Quilty, T. Ravenscroft, A. Robson, R.D. St. Denis, G. Steele, A.S. Thompson, K. Wraight, M. Wright

### *Lancaster*

L.J. Allison, A.E. Barton, G. Borissov, E.V. Bouhova-Thacker, A. Chilingarov, W.J. Dearnaley, H. Fox, K. Grimm, R.C.W. Henderson, G. Hughes, R.W.L. Jones, V. Kartvelishvili, R.E. Long, P.A. Love, H.J. Maddocks, M. Smizanska, J. Walder

### *Liverpool*

P.P. Allport, A.C. Bundock, S. Burdin, M. D'Onofrio, P. Dervan, C.B. Gwilliam, H.S. Hayward, J.N. Jackson, M. Jackson, T.J. Jones, B.T. King, M. Klein, U. Klein, J. Kretzschmar, P. Laycock, A. Lehan, S. Mahmoud, S.J. Maxfield, A. Mehta, S. Migas, J. Price, Y.J. Schnellbach, G. Sellers, J.H. Vossebeld, P. Waller, B. Wrona

### *London QMUL*

M. Bona, L. Cerrito, K. Ellis, G. Fletcher, J.R. Goddard, R. Hickling, M.P.J. Landon, S.L. Lloyd, J.D. Morris, E. Piccaro, E. Rizvi, G. Salamanna, G. Snidero, M. Teixeira Dias Castanheira

### *London RHBNC*

M.A. Alam, T. Berry, V. Boisvert, T. Brooks, R. Cantrill, N.J. Cooper-Smith, G. Cowan, L. Duguid, C.A. Edwards, S. George, S.M. Gibson, R. Gonçalo, J.G. Panduro Vazquez, Fr. Pastore, M. Rose, F. Spanò, P. Teixeira-Dias

### *London UC*

S. Baker, P. Bernat, S.P. Bieniek, J.M. Butterworth, M. Campanelli, D. Casadei, R.T. Chislett, I.A. Christidi, B.D. Cooper, A.R. Davison, E. Dobson, C. Gutschow, G.G. Hesketh, E. Jansen, N. Konstantinidis, L. Lambourne, M. Nash, E. Nurse, M.I. Ochoa, A.D. Pilkington, R. Prabhu, P. Sherwood, B. Simmons, C. Taylor, D.R. Wardrope, B.M. Waugh, P.A. Wijeratne

### *Manchester*

J. Almond, M. Borri, G. Brown, V. Chavda, B.E. Cox, C. Da Via, A. Forti, J. Howarth, K.D. Joshi, J.A. Klinger, F.K. Loebinger, J. Masik, T.J. Neep, A. Oh, M. Owen, J.R. Pater, J.E.M. Robinson, L. Tomlinson, S. Watts, S. Webb, M.J. Woudstra, T.R. Wyatt, U.K. Yang

### *Oxford*

R. Apolle, A.J. Barr, C.R. Boddy, J. Buchanan, R.M. Buckingham, A.M. Cooper-Sarkar, A. Dafinca, E. Davies, E.J. Gallas, S. Gupta, C. Gwenlan, D. Hall, C.P. Hays, J. Henderson, J. Howard, T.B. Huffman, C. Issever, R.S.B. King, L.A. Kogan, A. Larner, A. Lewis, Z. Liang, S.S.A. Livermore, C. Mattravers, R.B. Nickerson, K. Pachal, A. Pinder, A. Robichaud-Veronneau, N.C. Ryder, C. Sawyer, D. Short, J.C-L. Tseng, T. Vickey, G.H.A. Viehhauser, A.R. Weidberg, C.J.S. Young, J. Zhong

### *RAL*

T. Adye, R. Apolle, J.T. Baines, B.M. Barnett, S. Burke, E. Davies, A. Dewhurst, D. Emelianov, B.J. Gallop, C.N.P. Gee, S.J. Haywood, J. Kirk, C. Mattravers, S.J. McMahon, R.P. Middleton, W.J. Murray, M. Nash, P.W. Phillips, D.P.C. Sankey, W.G. Scott, F.J. Wickens, M. Wielers

### *Sheffield*

D. Costanzo, T. Cuhadar Donszelmann, I. Dawson, G.T. Fletcher, M.C. Hodgkinson, P. Hodgson, P. Johansson, E.V. Korolkova, B. Lopez Paredes, J.A. Mcfayden, P.S. Miyagawa, S. Owen, E. Paganis, K. Suruliz, D.R. Tovey, A. Tua

### *Sussex*

V. Bartsch, A. De Santo, Z.J. Grout, S. Martin-Haugh, C.J. Potter, A. Rose, F. Salvatore, I. Santoyo Castillo, M.R. Sutton

### *Warwick*

S.M. Farrington, P.F. Harrison, M. Janus, C. Jeske, G. Jones, T.A. Martin, E. Pianori

## **United States of America**

### *Albany*

W. Edson, J. Ernst, S. Guindon, V. Jain

### *Argonne*

J.T. Anderson, L. Asquith, B. Auerbach, R.E. Blair, S. Chekanov, G. Drake, E.J. Feng, A.T. Goshaw, T. LeCompte, J. Love, D. Malon, D.H. Nguyen, L. Nodulman, A. Paramonov, L.E. Price, J. Proudfoot, R.W. Stanek, P. van Gemmeren, A. Vaniachine, R. Yoshida, J. Zhang

### *Arizona*

E. Cheu, K.A. Johns, V. Kaushik, C.L. Lampen, W. Lampl, R. Leone, P. Loch, F. O'grady, J.P. Rutherford, F. Rühr, M.A. Shupe, E.W. Varnes, J. Veatch

### *Arlington UT*

A. Brandt, D. Côté, S. Darmora, K. De, A. Farbin, J. Griffiths, H.K. Hadavand, L. Heelan, C.M. Hernandez, M. Maeno, P. Nilsson, N. Ozturk, R. Pravahan, E. Sarkisyan-Grinbaum, M. Sosebee, A.R. Stradling, G. Usai, A. Vartapetian, A. White, J. Yu

### *Berkeley LBNL*

A.M. Bach, R.M. Barnett, J. Beringer, J. Biesiada, G. Brandt, P. Calafiura, L.M. Caminada, A. Cerri, F. Cerutti, A. Ciocio, R.N. Clarke, M. Cooke, K. Copic, S. Dube, K. Einsweiler, M. Garcia-Sciveres, M. Gilchriese, C. Haber, M. Hance, B. Heinemann, I. Hinchliffe, M. Hurwitz, W. Lavrijsen, C. Leggett, P. Loscutoff, Z. Marshall, A. Ovcharova, S. Pagan Griso, K. Potamianos, A. Pranko, D.R. Quarrie, M. Shapiro, L.A. Skinnari, A. Sood, M.J. Tibbetts, V. Tsulaia, D. Varouchas, J. Virzi, D.R. Yu

### *Boston*

S.P. Ahlen, C. Bernard, K.M. Black, J.M. Butler, L. Dell'Asta, L. Helary, M. Kruskal, J.T. Shank, Z. Yan, S. Youssef

### *Brandeis*

S. Aefsky, C. Amelung, J.R. Bensinger, L. Bianchini, C. Blocker, L. Coffey, R.K. Daya-Ishmukhametova, E.A. Fitzgerald, S. Gozpinar, D. Pomeroy, G. Sciolla, S. Zambito

### *Brookhaven BNL*

D.L. Adams, K. Assamagan, M. Begel, H. Chen, V. Chernyatin, R. Debbe, M. Ernst, T. Gadfort, H.A. Gordon, X. Hu, A. Klimentov, A. Kravchenko, F. Lanni, D. Lynn, T. Maeno, H. Ma, J. Metcalfe, P. Nevski, H. Okawa, D. Oliveira Damazio, F. Paige, S. Panitkin, M.-A. Pleier, V. Polychronakos, S. Protopopescu, M. Purohit, S. Rajagopalan, G. Redlinger, J. Schovancova, S. Snyder, P. Steinberg, H. Takai, M.C. Tamsett, N. Triplett, A. Undrus, T. Wenaus, S. Ye, A. Zaytsev

### *Chicago*

J. Alison, K.J. Anderson, M. Bogdan, A. Boveia, F. Canelli, Y. Cheng, G. Choudalakis, M. Fiascaris, R.W. Gardner, I. Jen-La Plante, A. Kapliy, H.L. Li, S. Meehan, C. Melachrinou, F.S. Merritt, C. Meyer, D.W. Miller, Y. Okumura, P.U.E. Onyisi, M.J. Oreglia, B. Penning, J.E. Pilcher, M.J. Shochet, F. Tang, L. Tompkins, J.M. Tuggle, I. Vukotic, J.S. Webster

### *Columbia*

A. Altheimer, T. Andeen, A. Angerami, T. Bain, G. Brooijmans, Y. Chen, B. Cole, J. Guo, D. Hu, E.W. Hughes, N. Nikiforou, J.A. Parsons, A. Penson, D.V. Perepelitsa, M.I. Scherzer, M. Spousta, E.N. Thompson, F. Tian, P.M. Tuts, D. Urbaniec, E. Williams, E. Wulf, L. Zhou

### *Dallas SMU*

T. Cao, K. Dindar Yagci, A. Firan, J. Hoffman, S. Kama, R. Kehoe, A.S. Randle-Conde, R.R. Rios, S.J. Sekula, R. Stroynowski, H. Wang, J. Ye

*Dallas UT*

M. Ahsan, J.M. Izen, X. Lou, H. Namasivayam, K. Reeves, W.C. Wong

*Duke*

A.T.H. Arce, D.P. Benjamin, A. Bocci, B. Cerio, K.D. Finelli, E. Kajomovitz, A. Kotwal, M.K. Kruse, S. Li, M. Liu, S.H. Oh, C.S. Pollard, C. Wang

*Hampton*

V.I. Vassilakopoulos

*Harvard*

J. Barreiro Guimarães da Costa, A. Belloni, B. Butler, P. Catastini, G. Conti, M. Franklin, J. Huth, L. Jeanty, D. Lopez Mateos, K.M. Mercurio, C. Mills, M. Morii, H.P. Skottowe, W.R. Spearman, A.L. Yen, G. Zevi della Porta

*Indiana*

S. Brunet, H. Evans, P. Gagnon, F. Luehring, H. Ogren, J. Penwell, J. Poveda, D. Price, D. Whittington, D. Zieminska

*Iowa*

D. Cinca, R.P. Gandrajula, G. Halladjian, U. Mallik, R. Mandrysch, N. Morange, Y. Pylypchenko, R. Zaidan

*Iowa State*

C. Chen, J. Cochran, F. De Lorenzi, F. Dudziak, N. Krumnack, S. Prell, A. Ruiz-Martinez, S. Shrestha, K. Yamamoto

*Louisiana Tech*

C. Bernius, R. Dhullipudi, Z.D. Greenwood, L. Sawyer, A. Sircar, R. Subramaniam, M.C. Tamsett

*MIT*

F.E. Taylor

*Massachusetts*

B. Brau, G. Colon, C. Dallapiccola, A. Meade, E.J.W. Moyses, P. Pais, E. Pueschel, T. Varol, D. Ventura, S. Willocq

*Michigan*

A.J. Armbruster, M.A. Chelstowska, T. Dai, E.B. Diehl, J. Dubbert, H. Feng, C. Ferretti, S. Goldfarb, D. Harper, D. Levin, L. Liu, X. Li, S.P. Mc Kee, H.A. Neal, N. Panikashvili, J. Qian, D. Scheirich, J. Searcy, R.P. Thun, S. Walch, A. Wilson, G. Wooden, Y. Wu, L. Xu, D. Zhang, B. Zhou, J. Zhu

*Michigan SU*

B. Alvarez Gonzalez, G. Arabidze, R. Brock, S. Caughron, P. Ge, R. Hauser, D. Hayden, J. Huston, J. Koll, J.T. Linnemann, B. Martin, B.G. Pope, R. Schwienhorst, H.J. Stelzer, D. Ta, K. Tollefson, P. True, H. Zhang

*NYU New York*

B. Budick, K. Cranmer, A. Haas, L. Hooft van Huysduynen, B. Kaplan, K. Karthik, R. Konoplich, A. Krasznahorkay, S. Kreiss, G.H. Lewis, A.I. Mincer, P. Nemethy, R.M. Neves, K. Prokofiev

*New Mexico*

I. Gorelov, M.R. Hoferkamp, S.C. Seidel, K. Toms, R. Wang

*Northern Illinois*

R. Calkins, D. Chakraborty, S. Cole, J.G. Rocha de Lima, C. Suhr, A. Yurkewicz, V. Zutshi

*Ohio SU*

M.J. Fisher, K.K. Gan, R. Ishmukhametov, H. Kagan, R.D. Kass, H. Merritt, J. Moss, A. Nagarkar, D.T. Pignotti, Y. Yang

*Oklahoma*

B. Abbott, P. Gutierrez, D.K. Jana, A. Marzin, R. Meera-Lebbai, S. Norberg, M. Saleem, H. Severini, P. Skubic, J. Snow, M. Strauss

*Oklahoma SU*

B. Abi, A. Khanov, F. Rizatdinova, D. Sidorov, J. Yu

*Oregon*

J.E. Brau, E. Brost, S. Majewski, C.T. Potter, E. Ptacek, P. Radloff, A. Reinsch, M. Shamim, N.B. Sinev, D.M. Strom, E. Torrence

*Pennsylvania*

K. Brendlinger, J. Degenhardt, S. Fratina, S. Heim, E. Hines, T.M. Hong, B. Jackson, J. Kroll, J. Kunkle, C.M. Lester, E. Lipeles, D. Olivito, R. Ospanov, R. Reece, J. Saxon, D. Schaefer, J. Stahlman, E. Thomson, A.N. Tuna, H.H. Williams

*Pittsburgh*

R.M. Bianchi, J. Boudreau, C. Escobar, T. Kittelmann, J. Mueller, D. Prieur, K. Sapp, V. Savinov, R. Yoosoofmiya

*SLAC*

I. Aracena, J. Backus Mayes, T. Barklow, R. Bartoldus, H.S. Bawa, J.E. Black, J.G. Cogan, T. Eifert, B.G. Fulsom, Y.S. Gao, N. Garelli, P. Grenier, P. Hansson, M. Kagan, M. Kocian, T. Koi, A.J. Lowe, C. Malone, R. Mount, T.K. Nelson, G. Piacquadio, A. Salnikov, A. Schwartzman, D. Silverstein, E. Strauss, D. Su, M. Swiatlowski, M. Wittgen, C. Young

*Santa Cruz UC*

D.S. Damiani, A.A. Grillo, A.M. Litke, W.S. Lockman, P.M. Manning, J. Mitrevski, J. Nielsen, H.F.-W. Sadrozinski, B.A. Schumm, A. Seiden

*Seattle Washington*

M. Beckingham, D. Blackburn, A. Coccaro, A.G. Goussiou, O.M. Harris, S.-C. Hsu, J.S. Keller, H.J. Lubatti, M. Marx, N. Rompotis, R. Rosten, J. Rothberg, M. Verducci, G. Watts

*Stony Brook*

A. Ahmad, S. Arfaoui, K. Chen, B. DeWilde, R. Engelmann, J.J. Goodson, V. Grassi, J.A. Gray, J. Hobbs, J. Jia, H. Li, B.E. Lindquist, P. Mastrandrea, R.L. McCarthy, S. Mohapatra, D. Puldon, M. Rijssenbeek, R.D. Schamberger, J. Stupak, D. Tsybychev

*Tufts*

P.H. Beauchemin, S. Hamilton, E. Meoni, A. Napier, K. Sliwa, J. Wetter

*UC Irvine*

S. Farrell, D. Gerbaudo, A.J. Lankford, L. Magnoni, A.S. Mete, A. Nelson, K. Rao, M. Relich, D.A. Scanicchio, M. Schernau, A. Taffard, B. Toggerson, G. Unel, M. Werth, D. Whiteson, N. Zhou

*UI Urbana*

M. Atkinson, A. Basye, N. Benekos, V. Cavaliere, P. Chang, J. Coggeshall, D. Errede, S. Errede, M. Kasten, K. Lie, T.M. Liss, A. McCarn, M.S. Neubauer, I. Vichou

*Wisconsin*

Sw. Banerjee, X. Chen, L.R. Flores Castillo, O. Gutzwiller, A.S. Hard, H. Ji, X. Ju, L. Kashif, A. Kruse, Y. Ming, Y.B. Pan, M.I. Pedraza Morales, H. Wang, W. Wiedenmann, S.L. Wu, H. Yang, G. Zobernig

*Yale*

J. Adelman, O.K. Baker, S. Bedikian, C. Cuenca Almenar, J. Cummings, Z. Czyczula, S. Demers, J. Erdmann, F. Garberson, T. Golling, D. Guest, A. Henrichs, T. Lagouri, L. Lee, A.G. Leister, A. Loginov, P. Tipton, R. Wall, B. Walsh, X. Wang

## **17 Acknowledgements**

We acknowledge the important contributions to the FTK design made by Ted Liu and Jamieson Olsen from Fermilab and Raffaele Tripiccionone from Ferrara.

Our deep appreciation goes to the FTK institutes, their technical staffs, and their funding agencies, including the Seventh Framework Programme (FP7), for invaluable help in designing the FTK. We thank our ATLAS referees for their critical review and constructive suggestions: John Baines (Chair), Francis Anghinolfi, Tony Doyle, Markus Elsing, Hal Evans, and Klaus Moenig. We also thank ATLAS Management, the Upgrade leadership, and the TDAQ management team for their guidance during the TDR process.

We are grateful to the members and institutes of the ATLAS collaboration whose extraordinary detector the FTK will augment. We also acknowledge CERN and its departments for the LHC, whose steadily rising luminosity translates to increasing challenges for the trigger.

## References

- [1] L. Evans and P. Bryant, “LHC Machine”, JINST 3, S08001 (2008).
- [2] ATLAS Collaboration, “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”, Phys.Lett. B716, 1 (2012); CMS Collaboration, “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC”, Phys.Lett. B716, 30 (2012).
- [3] ATLAS Collaboration, “ATLAS susy searches - 95% cl lower limits (status: Plhc 2013)”, [https://twiki.cern.ch/twiki/pub/AtlasPublic/CombinedSummaryPlots/AtlasSearches\\_susy\\_lhpc2013.pdf](https://twiki.cern.ch/twiki/pub/AtlasPublic/CombinedSummaryPlots/AtlasSearches_susy_lhpc2013.pdf).
- [4] Atlas Collaboration, “ATLAS exotics searches\* - 95% cl lower limits (status: May 2013)”, [https://twiki.cern.ch/twiki/pub/AtlasPublic/CombinedSummaryPlots/AtlasSearches\\_exotics\\_may13.pdf](https://twiki.cern.ch/twiki/pub/AtlasPublic/CombinedSummaryPlots/AtlasSearches_exotics_may13.pdf).
- [5] L. Rossi and O. Brüning, “High Luminosity Large Hadron Collider A description for the European Strategy Preparatory Group”, CERN-ATS-2012-236 (2012).
- [6] ATLAS Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider”, JINST 3, S08003 (2008).
- [7] ATLAS Collaboration, “ATLAS high-level-trigger, data acquisition and controls”, CERN-LHCC-2003-022 (2003).
- [8] ATLAS Collaboration, “Performance of the ATLAS Trigger System in 2010”, Eur.Phys.J. C72, 1849 (2012).
- [9] ATLAS Collaboration, “Letter of Intent for the Phase-I Upgrade of the ATLAS Experiment”, CERN-LHCC-2011-012 (2011).
- [10] ATLAS Collaboration, “Performance of the ATLAS inner detector track and vertex reconstruction in the high pile-up lh environment”, ATLAS-CONF-2012-042 (2012).
- [11] ATLAS Collaboration, “The ATLAS Inner Detector commissioning and calibration”, Eur.Phys.J. C70, 787 (2010).
- [12] ATLAS Collaboration, “ATLAS Insertable B-Layer Technical Design Report”, CERN-LHCC-2010-013 (2010).
- [13] S. Belforte et al., “The CDF trigger silicon vertex tracker (SVT)”, IEEE Trans.Nucl.Sci. 42, 860 (1995); J. Adelman et al., “The Silicon Vertex Trigger upgrade at CDF”, NIM A572, 361 (2007).
- [14] ATLAS Collaboration, “Letter of Intent for the Phase-II Upgrade of the ATLAS Experiment”, CERN-LHCC-2012-022; LHCC-I-023 (2012).
- [15] F. Morsani *et al.*, “The AMchip: A VLSI associative memory for track finding”, Nucl. Instrum. Meth. A315, 446 (1992).
- [16] E. van der Bij et al., “S-LINK, a data link interface specification fo the LHC era”, IEEE Trans.Nucl.Sci. 44, 398 (1997).
- [17] ATLAS Collaboration, “Performance of the ATLAS muon trigger in 2011”, ATLAS-CONF-2012-099 (2012).

- [18] ATLAS Collaboration, “Determination of the muon reconstruction efficiency in ATLAS at the Z resonance in proton-proton collisions at  $\sqrt{s}=7$  TeV”, ATLAS-CONF-2011-008 (2011).
- [19] B. Batell et al., ”Higgs Couplings and Precision Electroweak Data”, JHEP 1301, 139 (2013).
- [20] ATLAS Collaboration, “Performance of Impact Parameter-Based b-tagging Algorithms with the ATLAS Detector using Proton-Proton Collisions at  $\sqrt{s} = 7$  TeV”, ATLAS-CONF-2010-091 (2010).
- [21] ATLAS Collaboration,, “Commissioning of the ATLAS high-performance b-tagging algorithms in the 7 TeV collision data”, ATLAS-CONF-2011-102 (2011).
- [22] ATLAS Collaboration, “Characterization of Interaction-Point Beam Parameters Using the pp Event-Vertex Distribution Reconstructed in the ATLAS Detector at the LHC”, ATLAS-CONF-2010-027 (2010).
- [23] Atlas Collaboration, “Vertex Selection Efficiency”, [http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/IDTRACKING/PublicPlots/ATL-COM-PHYS-2012-474/hs\\_select\\_prelim.eps](http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/IDTRACKING/PublicPlots/ATL-COM-PHYS-2012-474/hs_select_prelim.eps).
- [24] ATLAS Collaboration, “Performance of the ATLAS tau trigger in 2011”, ATLAS-CONF-2013-006 (2013).
- [25] ATLAS Collaboration, “Search for the standard model higgs boson in h to tau tau decays in proton-proton collisions with the ATLAS detector”, ATLAS-CONF-2012-160 (2012).
- [26] A. Annovi et al., “A new variable-resolution associative memory for high energy physics”, (Int. Conf. on Advancements in Nuclear Instrumentation Measurement Methods and their Applications, ANIMA), Proc. IEEE 116, 117 (2011).
- [27] A. Annovi and M. Beretta, “A fast general-purpose clustering algorithm based on {FPGAs} for high-throughput data processing”, NIMA 617, 254 (2010).
- [28] A. Annovi et al., “The EDRO board connected to the Associative Memory: a “Baby” FastTracker processor for the ATLAS experiment”, ATL-DAQ-PROC-2011-018 (2011).
- [29] L. Fabbri et al., “The high rate data acquisition system for the SLIM5 beam test”, Nucl.Instrum.Meth. A617, 321 (2010).
- [30] H. Wind, “Principal component analysis, Pattern recognition for track finding, Interpolation and functional representation”, CERN-EP-INT-81-12-REV (1982).
- [31] A. Andreani et al., “The AMchip04 and the processing unit prototype for the FastTracker”, IOP J. Instr. 7, C08007 (2012).
- [32] A. Annovi et al., “Associative memory design for the fast track processor (FTK) at ATLAS”, Proc. IEEE Nuclear Science Symp. and Medical Imaging Conf., 141 (2011).
- [33] P. Fischer, “First implementation of the MEPHISTO binary readout architecture for strip detectors”, Nucl. Instr. and Meth. in Phys. Res. Sect. A 461, 499 (2001).
- [34] A. Barriuso Poy et al., “The detector control system of the ATLAS experiment”, JINST 3, P05006 (2008).
- [35] ATLAS Collaboration, “ATLAS computing: Technical design report”, CERN-LHCC-2005-022 (2005).

- [36] K. De et al., “ATLAS Production System Upgrade”, ATL-COM-SOFT-2013-007 (2013).
- [37] ATLAS Collaboration, “ATLAS Computing Resource Requirements for Data-taking in 2013-2015”, in preparation.
- [38] W. Lukas, “Fast Simulation for ATLAS: Atlfast-II and ISF”, ATL-SOFT-PROC-2012-065 (2012).
- [39] S. Kolos et al., “High-performance scalable Information Service for the ATLAS experiment”, ATL-DAQ-PROC-2012-021 (2012).
- [40] M. Barczyk et al., “Verification and diagnostics framework in ATLAS trigger / DAQ”, arXiv:hep-ex/0305106.
- [41] S. Kolos et al., “A software framework for Data Quality Monitoring in ATLAS”, J.Phys.Conf.Ser. 119, 022033 (2008).
- [42] P. Adragna et al., “GNAM: A low-level monitoring program for the ATLAS experiment”, IEEE Trans.Nucl.Sci. 53, 1317 (2006).
- [43] A Schöning, “A self seeded first level track trigger for ATLAS”, Journal of Instrumentation 7, C10010 (2012).
- [44] A. Annovi *et al.*, “A VLSI processor for fast track finding based on content addressable memories”, IEEE Trans.Nucl.Sci. 53, 2428 (2006).
- [45] A. Annovi *et al.*, “Associative Memory Design for the FastTrack Processor (FTK) at ATLAS”, ATL-DAQ-PROC-2011-045 (2011).
- [46] K. Pagiamtzis and A. Sheikholeslami, “Content-addressable memory (CAM) circuits and architectures: 121 A tutorial and survey”, IEEE J. Solid-State Circ. 41, 712 (2006).



