

**ELECTRONICS AND TIMING FOR THE  
AUGERPRIME UPGRADE AND CORRELATION OF  
STARBURST GALAXIES WITH ARRIVAL  
DIRECTIONS OF ULTRA HIGH ENERGY COSMIC  
RAYS**

**ROBERT HALLIDAY**

Submitted in partial fulfillment of the requirements for the degree of  
**Doctor of Philosophy**

Dissertation Adviser: Prof. Corbin E. Covault

Department of Physics

**CASE WESTERN RESERVE UNIVERSITY**

May 2019

**CASE WESTERN RESERVE UNIVERSITY  
SCHOOL OF GRADUATE STUDIES**

We hereby approve the dissertation of  
**Robert Halliday**  
candidate for the degree of **Doctor of Philosophy**\*

---

Committee Chair  
**Corbin E. Covault**

---

Committee Member  
**Benjamin Monreal**

---

Committee Member  
**John Ruhl**

---

Committee Member  
**David Kazdan**

---

Date of Defense  
**March 19, 2019**

\*We also certify that written approval has been obtained for any proprietary material contained therein.

# Contents

<b>Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xiii</b>
<b>Acknowledgements</b>	<b>xiv</b>
<b>Abstract</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 A brief history of Cosmic Rays and Particle Physics . . . . .	2
1.2 Motivation and Goals . . . . .	10
<b>2 Cosmic Ray Sky: The State of the Field</b>	<b>12</b>
2.1 Spectrum . . . . .	12
2.1.1 Fermi Shock Acceleration . . . . .	15
2.2 Sources At The Highest Energies . . . . .	18
2.2.1 Power Considerations . . . . .	19
2.2.2 Hillas Criterion . . . . .	20
2.2.3 Top Down vs. Bottom Up . . . . .	23

2.2.3.1	Top Down Models . . . . .	23
2.2.3.2	Bottom Up Models . . . . .	24
2.2.4	Composition . . . . .	27
2.2.5	Anisotropy . . . . .	28
2.2.6	Concluding Thoughts: Cosmic Ray Spectrum and Sources . . . . .	30
<b>3</b>	<b>Extensive Air Showers and Cosmic Ray Detectors</b>	<b>31</b>
3.1	Extensive Air Showers . . . . .	31
3.1.1	Particle Physics Basics . . . . .	33
3.1.1.1	Moving Particle Physics into Shower Physics . . . . .	36
3.1.2	Transport Equations . . . . .	39
3.1.2.1	Grammage, Interaction Length and Radiation Length . .	41
3.1.3	Atmospheric Losses . . . . .	43
3.1.4	Shower Parameters . . . . .	45
3.1.4.1	Heitler-Matthews Model . . . . .	46
3.1.4.2	Shower Universality . . . . .	49
3.1.4.3	Lateral Distribution Functions . . . . .	50
3.1.5	Monte Carlo Methods: CORSIKA . . . . .	51
3.2	Detection Methods . . . . .	53
3.2.1	Water Cherenkov . . . . .	54
3.2.2	Scintillation . . . . .	56
3.2.2.1	Comparison and Compatibility of Water Cherenkov and Scintillation Detectors . . . . .	58
3.2.3	Fluorescence . . . . .	61
3.2.4	Direct and Ice Cherenkov . . . . .	62
3.2.5	Radio . . . . .	65

3.3	Operating Experiments . . . . .	67
3.3.1	Pierre Auger Observatory . . . . .	67
3.3.1.1	Auger Surface Detector . . . . .	68
3.3.1.2	Auger Fluorescence Detector . . . . .	70
3.3.1.3	AugerPrime Upgrade . . . . .	72
3.3.2	TA . . . . .	72
3.3.2.1	Work Towards a Cross Calibration of Auger and TA . . .	74
3.3.3	Ice Cube . . . . .	75
3.3.4	CTA . . . . .	78
3.3.4.1	Schwarzschild-Coud� Telescope . . . . .	80
<b>4</b>	<b>Electronics and Firmware for the Auger Prime Upgrade</b>	<b>83</b>
4.1	Scientific Motivations for AugerPrime . . . . .	84
4.2	Decoupling the Shower Components: SD Composition Measurements . .	85
4.3	Board Overview . . . . .	89
4.4	Subsystems and Work Packages . . . . .	93
4.4.1	Scintillation Surface Detector . . . . .	96
4.4.2	Trigger Logic . . . . .	97
4.4.2.1	Trigger Hierarchy . . . . .	99
4.4.3	Slow Control . . . . .	100
4.4.4	Hardware Design and Integration . . . . .	101
4.4.5	Software Development . . . . .	102
4.4.6	Calibration . . . . .	102
4.5	GPS and Timing . . . . .	103
4.5.1	GPS Basics . . . . .	104
4.5.2	GPS Constellation . . . . .	107

4.5.3	GPS Receiver Communications Protocols . . . . .	108
4.5.3.1	Message Selection . . . . .	110
4.5.3.2	Sawtooth Correction: The Clock Granularity Message . .	111
4.5.3.3	WGS-84 . . . . .	113
4.6	GPS Integration . . . . .	113
4.6.1	The Purpose of GPS in AugerPrime . . . . .	114
4.6.2	Hardware Design . . . . .	115
4.6.2.1	Introduction: Xilinx Vivado <sup>TM</sup> Design Tools . . . . .	116
4.6.2.2	Implementing the UART . . . . .	119
4.6.2.3	Physical Connection . . . . .	121
4.6.3	GPS Communications Software . . . . .	122
4.6.3.1	Parser . . . . .	124
4.6.3.2	Classes and Packing . . . . .	125
4.7	Board Operating System and Bring-Up . . . . .	129
4.7.1	Boot Stages . . . . .	129
4.7.2	PetaLinux . . . . .	130
4.7.3	Bring-up Documentation . . . . .	130
4.7.4	Installing the Environment . . . . .	131
4.7.4.1	Necessary Downloads . . . . .	131
4.7.4.2	Necessary Packages . . . . .	132
4.7.4.3	Necessary Prerequisites . . . . .	132
4.7.4.4	Installation Procedure . . . . .	132
4.7.5	Configuring a Board for Boot Up . . . . .	133
4.7.5.1	Setting up Working Environment . . . . .	133
4.7.5.2	Configuring and Building PetaLinux . . . . .	134

4.7.5.3	Booting Your Build . . . . .	134
4.7.5.4	Adding Programs . . . . .	135
4.8	Conclusions: Reliability of the Integrated GPS and Operating System . . .	136
<b>5</b>	<b>Timing for AugerPrime</b>	<b>137</b>
5.1	Timing Basics . . . . .	138
5.2	GPS Receiver Selection Testing . . . . .	140
5.2.1	Concerns regarding the availability of the i-Lotus M12M . . . . .	141
5.2.2	Specifications: i-Lotus M12M vs. the SSR-6Tf . . . . .	142
5.2.3	Initial Bench Tests: Absolute GPS Timing . . . . .	142
5.2.3.1	Test Stand for Absolute GPS Timing . . . . .	143
5.2.3.2	Results for Absolute GPS Timing . . . . .	144
5.2.4	Relative GPS Time-Tagging . . . . .	145
5.2.4.1	Test Stand for Relative GPS Timing and Temperature Dependence . . . . .	146
5.2.4.2	Temperature and Relative Timing Test . . . . .	147
5.2.5	Conclusions . . . . .	149
5.3	AugerPrime Time-Resolution . . . . .	149
5.3.1	Time-Tagging Specifications and Ports . . . . .	151
5.3.2	Coincident Showers Method . . . . .	153
5.3.2.1	Shower Selection for Arrival Directions . . . . .	155
5.3.3	Coincident Showers Results . . . . .	155
5.3.4	Synchronization Cable Method . . . . .	157
5.3.5	Synchronization Cable Results . . . . .	162
5.3.6	Conclusions . . . . .	162

<b>6 Timing Equipment and Analysis</b>	<b>164</b>
6.1 Stability and Accuracy of Timing Systems . . . . .	164
6.2 CWRU Time Tagging Module: TIM . . . . .	166
6.2.1 TIM Hardware . . . . .	167
6.2.2 TIM Firmware . . . . .	169
6.2.3 Time-Tagging PL . . . . .	170
6.2.4 Communications PL . . . . .	172
6.2.5 TIM Software . . . . .	174
6.2.5.1 TIM Operating Software . . . . .	174
6.2.5.2 TIM Analysis Software . . . . .	177
6.3 Spatial Correlation of GPS Timing Errors . . . . .	178
6.3.1 Experimental Setup and Method . . . . .	179
6.3.2 Measurement . . . . .	181
6.3.3 Results . . . . .	183
6.3.3.1 Future Directions . . . . .	184
6.3.4 TIM@TA . . . . .	185
6.3.4.1 Engineering Challenges . . . . .	187
6.3.5 TIM@CTA . . . . .	188
6.3.5.1 Requirements . . . . .	189
6.3.5.2 Timing Transfer System . . . . .	190
6.3.5.3 TIMPrime: Time-tagging System . . . . .	191
6.3.5.4 Current Status . . . . .	193
<b>7 Correlation of UHECR arrival directions with Starburst Galaxies</b>	<b>194</b>
7.1 Charged Particle Astronomy . . . . .	195
7.1.1 SBGs and the Hillas Criterion . . . . .	195

7.2	Magnetic Field Modeling and JF12 . . . . .	195
7.3	Starburst Test Methods . . . . .	195
7.4	Starburst Test Results . . . . .	195
7.5	Conclusions . . . . .	195
<b>8</b>	<b>Conclusion</b>	<b>196</b>
<b>A</b>	<b>Glossary</b>	<b>197</b>

# List of Figures

1.1	Positron Picture by Carl Anderson . . . . .	4
1.2	Volcano Ranch Layout . . . . .	5
2.1	All Particle Spectrum (Swordy Plot) . . . . .	13
2.2	High Energy All Particle Spectrum . . . . .	15
2.3	Shock Acceleration . . . . .	16
2.4	Hillas Plot . . . . .	22
2.5	$X_{max}$ per Particle Type Contours . . . . .	27
2.6	First Harmonic in Anisotropy Analysis . . . . .	28
2.7	Anistropy Sky Map . . . . .	29
3.1	Feynman Diagrams and the Standard Model . . . . .	35
3.2	Density in Standard Atmosphere . . . . .	42
3.3	Vertical Depth and Altitude . . . . .	44
3.4	Heitler Model Diagram . . . . .	46
3.5	CORSIKA: Gamma Rays Versus Hadrons . . . . .	53
3.6	Cherenkov Light Diagram . . . . .	55
3.7	Auger and TA Scintillators . . . . .	58
3.8	AugerPrime Stations . . . . .	59
3.9	Auger and TA Surface Detector Apertures . . . . .	60

3.10	Fluorescence Reconstruction . . . . .	61
3.11	Auger and Fly’s Eye FDs . . . . .	63
3.12	Comparison of UHECR and VHE Gamma Ray Detectors . . . . .	64
3.13	Radio Emission Diagram . . . . .	66
3.14	Maps of Auger . . . . .	68
3.15	Housing of Auger PMT . . . . .	69
3.16	Auger Surface Station Diagram . . . . .	70
3.17	Auger Fluorescence Detector . . . . .	71
3.18	TA Map and Surface Detector . . . . .	73
3.19	Auger@TA Setup . . . . .	75
3.20	IceCube Diagram and DOM . . . . .	76
3.21	IceCube PeV Neutrino Events . . . . .	78
3.22	CTA Telescope Candidates . . . . .	79
3.23	SCT Camera and Optics . . . . .	81
4.1	$X_{max}$ Distributions . . . . .	86
4.2	Muon Content vs. Primary Over Depth . . . . .	88
4.3	UUB Picture . . . . .	89
4.4	Zynq™ Schematic . . . . .	91
4.5	UUB Block Schematic . . . . .	95
4.6	AugerPrime SSD Mount and Station Positions . . . . .	96
4.7	Auger Trigger Hierarchy . . . . .	100
4.8	AugerPrime Charge Histograms . . . . .	103
4.9	AugerPrime MIP vs. VEM . . . . .	104
4.10	GPS Diagrams . . . . .	106
4.11	Example of a Motorola Binary Message . . . . .	109

4.12	Simulated GPS Coverage over Auger and Cleveland . . . . .	110
4.13	Sawtooth Diagram . . . . .	112
4.14	Sawtooth Data . . . . .	112
4.15	UART Lite Top-Level Diagram . . . . .	117
4.16	UART Lite Block Diagram Representation . . . . .	119
4.17	UART Lite Customization Parameters . . . . .	120
4.18	Top Level Wrapper Example . . . . .	122
4.19	M12M Footprint Technical Draft . . . . .	123
4.20	M12M Physical Board Layout . . . . .	123
4.21	GPS Parser Loop . . . . .	125
4.22	GPS Receiver Process . . . . .	126
4.23	@@Ha Message Struct and Packing Function . . . . .	127
4.24	Handling Repeated Fields . . . . .	128
4.25	PetaLinux Config Screen . . . . .	130
5.1	Absolute Timing Test Diagram . . . . .	143
5.2	Absolute Timing Test Results . . . . .	144
5.3	Relative Timing Test Diagram . . . . .	146
5.4	SSR-6Tf Relative Timing . . . . .	147
5.5	SSR-6Tf Relative Timing . . . . .	148
5.6	SSR-6Tf Temperature Testing . . . . .	148
5.7	M12M Temperature Testing . . . . .	148
5.8	Timing Triplet Map . . . . .	151
5.9	Time-Tagging Module IP Element . . . . .	152
5.10	Distribution of Distances-to-Detector . . . . .	156
5.11	Coincident Shower Time Differences . . . . .	156

5.12	Distribution of Distances-to-Detector . . . . .	157
5.13	Synchronization Cable Timing Diagram . . . . .	159
5.14	ADC Trace Panel . . . . .	161
5.15	Sync-Cable Histogram of $t_{diff}$ . . . . .	162
6.1	CSAC Allen Deviation . . . . .	165
6.2	TIM Internals Diagram . . . . .	167
6.3	Zynq PS/PL Programmability Diagram . . . . .	168
6.4	TIM Time-Tagging PL . . . . .	171
6.5	TIM PL Block Diagram . . . . .	173
6.6	Spatial Correlations Diagram . . . . .	179
6.7	Time Streams Example . . . . .	182
6.8	Spatial Correlation Results . . . . .	183
6.9	Effects of a Thunderstorm on GPS Timing . . . . .	184
6.10	Auger@TA: SlimTIM . . . . .	187
6.11	CTA Timing Transfer System . . . . .	190
6.12	TIMPrime Time-Tagging PL . . . . .	192

# List of Tables

2.1	Power of Cosmic Accelerators . . . . .	20
2.2	Source Summary . . . . .	26
4.1	Auger Prime Institutions . . . . .	93
4.2	Auger Work Packages . . . . .	94
4.3	FPGA UART Utilization Comparison . . . . .	116
4.4	M12M/SSR-6Tf Header Pinout . . . . .	124
4.5	GPS Message Type . . . . .	128
5.1	Table of Register Assignments . . . . .	153

*To my family, past, present and future.*

# **Acknowledgements**

# Electronics and Timing for the AugerPrime Upgrade and Correlation of Starburst Galaxies with Arrival Directions of Ultra High Energy Cosmic Rays

ROBERT HALLIDAY

## **Abstract**

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

# Chapter 1

## Introduction

In the development of modern physics throughout the last century the science of cosmic rays, particles originating outside of Earth's atmosphere, has been a driving force. Since their first observation by Coulomb and subsequent exploration by Domenico Pacini and separately Victor Hess, the study of particles accelerated by extraterrestrial mechanisms has been a main source of physics knowledge, leading to the founding of particle physics, astrophysics, and more recently particle astronomy [**pacini**]. Cosmic rays are known to originate from a variety of sources, in particular low and medium energy cosmic rays are believed to originate from supernovae and their remnants, while the highest energy cosmic rays are still of unknown origin [**stanev**].

The study of Ultra High Energy Cosmic Rays (UHECRs) originated with experiments by Pierre Auger and his colleagues to detect extensive air showers, and was solidified by the work of John Linsley and collaborators in the experiments at Volcano Ranch [**linsley**]. The developments in the detection mechanisms and source identification that will be discussed in this document, follow along the line of experimentation propagated by Linsley, using large arrays of detectors spread over vast distances to detect extensive air showers. Throughout the latter half of the 20th century, various techniques were used to detect UHE-

CRs including films and electronic arrays of scintillators and water tanks. These efforts culminated in the construction of the Pierre Auger Observatory.

Initially built as an array of 1600 Water Cherenkov detectors and 24 Nitrogen Fluorescence telescopes, covering  $3600 \text{ km}^2$ , approximately the size of the American state of Rhode Island, the Pierre Auger Observatory has been the flagship detector for particles above an energy of  $10^{18} \text{ eV}$ . As time progressed, the observatory integrated many efforts to improve its ability to characterize and detect extensive air showers [**enhancements**]. After about a decade of operation, the possibility of upgrading the observatory's capabilities across all detectors was explored, and now we are on the threshold of executing an array-wide improvement in the electronics and detection abilities of the apparatus [**firstprime**]. This upgrade and an effort to correlate the arrival directions of UHECRs with starburst galaxies will be the main focus of this work.

## 1.1 A brief history of Cosmic Rays and Particle Physics

In the post World War One era of physics, many efforts were made to understand and utilize the advances of Quantum Mechanics and General and Special Relativity. In 1926, Erwin Schrödinger posited his eponymous equation, describing the dynamics of wavefunctions, the fundamental description of quantum particles. Shortly thereafter, in 1928, Paul A. M. Dirac proposed his equation which united the principles of Quantum Mechanics and Special Relativity [**dirac**]. It is with this advance, that the importance of cosmic rays takes center stage.

Integral to the Dirac equation, is the prediction of opposite-charge equal-mass particles for each known elementary particle. At the time, knowledge of elementary particles was extremely limited and this led only to the prediction of the positron. In 1933, Carl Anderson

published his paper confirming the existence of a “positive electron” which was quickly followed, just 3 years later, by the discovery of the muon [**positron**, **muon**]. In the years after this, cloud chambers with scintillator triggered cameras led to the discovery of many new particles, most of which were types of mesons.

Meanwhile, Pierre Auger and his collaborators were working to uncover particles of extremely high initial energies causing showers of lower energy particles spread over great distances [**firstshowers**]. He and his team did this by setting up Geiger-Muller and later scintillation counters at increasing separations from each other and watching the rate at which coincidences were detected as a function of their separation. They tried this at multiple sites while monitoring external conditions such as barometric pressure. With this experiment, the existence of particles estimated to have energies upwards of  $10^{16}$  eV was confirmed. While many of the advances in particle physics at this time were driven by lower energy cosmic rays, this represents the first venture towards detecting Ultra High Energy Cosmic Rays.

## Post World War II

After the second world war, the technological and science funding situations conspired to provide a previously-unseen level of funding for experimentation (particularly into the mid and late ’50s) [**scifund1**]. Additionally, the advent of the photomultiplier tube (PMT) and the application of the first computers gave cosmic ray and particle physicists new tools [**pmthistory**]. While versions of the photomultiplier tube existed in the 1930’s, the products available in the late 40’s and 50’s had been refined from their predecessors. The photomultiplier tube is a device consisting of a photocathode— a thin piece of metal off of which free electrons are generated from photons through the photoelectric effect— multiple metal leaves and an anode collect the freed charge. A large voltage is held between the anode

and the cathode such that when an electron comes off of the cathode, it is pulled towards the anode. It excites many more electrons off of the metal leaves and these electrons are similarly accelerated towards the anode by the high voltage. Through this process, a photomultiplier tube takes a photon and produces a measurable current on the order of nano or micro-amperes.

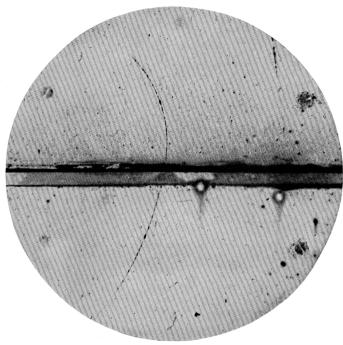


Figure 1.1: The photo published by Carl Anderson in [positron] in 1932, which led to the confirmation of the positron’s existence. The cosmic rays entering his cloud chamber were put under a magnetic field, hence the bending trajectory, and pictured in the center is a lead plate, meant to stop electrons.

During this time, we begin to see a split between particle physics and cosmic ray physics. While particle accelerators had existed since the 20’s (or earlier depending on how rigorously ‘particle acceleration’ is defined), they were now becoming much more useful as fundamental probes of physics in ways that cosmic rays simply could not match up to [**colliderhistory1**]. Fundamentally, the difference between studying particle physics using a collider versus using cosmic rays is in the predictability of the point and time of interaction. Cosmic rays, even today, provide access to much higher energy interactions, but they occur more-or-less randomly, while interactions in a collider occur in a predetermined location and time (some like to say “in a jar”). Due to this predictability, particle physicists need to only build one detection apparatus and guarantee the interesting portions of the collision will occur inside this apparatus. Meanwhile, cosmic ray physics at this time, especially at the highest energies, began to be engineered in a distributed way, opting for dispersed detectors in the vein of Pierre Auger’s experiments in the ’30s. Perhaps the most notable experiment<sup>1</sup> pursuing this direction was the Volcano Ranch Ex-

---

<sup>1</sup>There were many influential and noteworthy experiments that shaped the field, but here I will concentrate on an outline of experiments to show how the field of particle astrophysics advanced into the creation of the

periment, led by John Linsley [**linsley**]. Linsley greatly benefitted from the experimental agility of living in a time when a “large” collaboration was on the order of dozens of scientists (a trend started by Case Western’s own Albert Michelson). His work at Volcano Ranch began with the setup of an array of scintillation counters, commonly referred to as scintillators (although this is a slight misnomer, see section 3.2.2). These scintillators were placed at uniform distances apart from each other in a triangular grid. The grid spacing changed as the experiment progressed, but to give context, in 1962 it was 884m [**volranch**]. Due to the perhaps unimpressive speed of the electronics of his day, it was still debated whether the arrival times of the signals from each scintillation detector could be used to triangulate the arrival directions of the showers, thus increasing the grid spacing gave longer differences in arrival times and therefore made timing reconstruction a more viable approach. That said, for many of the great discoveries of Volcano Ranch, the coincidence window for detection was set such that only showers of about  $10^{\circ}$  zenith would be recorded.

It was in this configuration that Linsley’s group detected the first  $10^{20}$ eV comic ray. Along with a handful of other data points in this regime, Linsley was able to see the flattening of the spectrum [**volranch**], as will be discussed in the coming sections of this chapter. The effective radius of curvature of cosmic ray showers in various experiments was also noted at this time, and it was correctly assumed by Linsley and others in the field that this relates to the point of first interaction in the atmosphere, and in some ways the maximum of particle production in the shower.

As is often seen in science and technological devel-

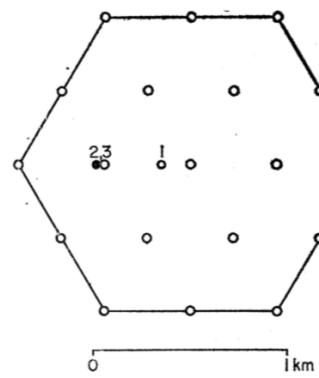


Figure 1.2: A diagram of Volcano Ranch in 1962 (**[volranch]**): the dark circle represents a new detector being added for in depth shower timing studies.

opment, great ideas are frequently had in multiple places independently around the same time. This is the case of the Air Fluorescence camera/technique. At Volcano Ranch by Linsley's group, as well as the Sydney Air Shower Array detectors called Fluorescence Telescopes (see subsection 3.2.3) were implemented. The design comes from the group of Kenneth Greissen, a prolific graduate student of prolific cosmic ray researcher Bruno Rossi, who was working on it through the mid 1960's but was somewhat unsuccessful (although his work did result in the construction of the Fly's Eye detectors) [**ultraray**]. The fluorescence telescope is one of the inventions that led to the deep understanding of cosmic ray showers as one could now visualize how the shower develops as it goes through the atmosphere. This allowed confirmation and further study of important quantities such as  $X_{max}$  and the elongation rate.

Before moving on (or back, as the case may be) to emulsion film detectors, two honorable and important mentions must be given. First, the Haverah park experiment of the University of Leeds in England, including Alan Watson, a great proponent and leader of UHECR research, became one of the top High Energy Cosmic Ray detectors around the same time as the deployment of Volcano Ranch (although it only truly came into full operation about 5 years after both experiments were commissioned). The array initially consisted of 4 water Cherenkov detectors in a semi-triangular grid (as triangular as it can be with only 4 detectors) and was triggered in much the same way as the Volcano Ranch Experiment [**haverah·lillicrap**]. While Volcano Ranch had a much larger area than Haverah Park at the time, more advanced electronics, and upgrades over the years allowed the experiment to produce data that was still being usefully analyzed well into the 1990's [**haverah·watson**]. In many ways, Haverah Park advanced the work of previous water Cherenkov experiments, which had problems with fungus and water purity [**haverah·who**]. A main breakthrough from the water Cherenkov technique was a more favorable cost vs. aperture curve over

scintillators (see subsection 3.2.1).

In a competing lineage to electronic detectors, emulsion film detectors were developed and deployed in the late 1930's by Marietta Blau, who deployed them high into the Alps [**ultraray blau**]. These films acted much like a film picture from a camera, except that they are exposed by either high energy ionizing radiation or X-rays (as in the analogous medical application). These films allowed cosmic ray researchers from the late thirties well into the eighties to see the types of tracks left by showers over long exposure times [**crapp**]. While these measurements are not inherently calorimetric, the precision given by being able to visually see the tracks of ionizing particles allows for particle counting and shower geometry methods to attempt to determine the primary energy. These types of experiments allowed researchers to directly visualize the nature and progression of cosmic ray showers, allowing them to confirm and give insight into the results of other experiments which were better at collecting larger numbers of showers at higher energies, but were all electronically operated and could not give intuitive insight.

### Dawn of the Modern Era

Looking forward from the early days of Volcano Ranch and Haverah Park, the late 1960's through the 80's opened new doors for cosmic rays physics, especially in terms of new "messenger" particles being looked at. In this vein, new detection techniques were provided by both the Whipple Telescope, the first Imaging Atmospheric Cherenkov Telescope (IACT) and the KamiokaNDE experiment, a proton decay experiment that accidentally became the first effective neutrino telescope. The Whipple telescope is based on a Davies-Cotton optics design from the mid 1950's which uses segmented mirrors to create a large aperture that is in turn used to collect the direct Cherenkov light from air showers caused by GeV to TeV gamma rays. This proved the viability of what are now frequently referred to

as gamma ray telescopes and almost all such designs are derived from the Whipple design. These include VERITAS (at the same site as the Whipple Telescope), MAGIC and HESS, as well as the upcoming CTA project.

On the neutrino front, the KamiokaNDE I and II experiments ran through the 80's, beginning in 1983, and while they intended to detect proton decay, their detectors were sensitive enough that they were able to detect atmospheric, solar and astrophysical neutrinos. The apparatus is a 3000 metric ton pool of water, which is ideal for a proton decay experiment as it contains a high concentration of hydrogen atoms and facilitates Cherenkov light production [**kamiokande**]. That said, the large fiducial volume of the detector, the sides of which are lined with 1000 PMTs specially designed by Hamamatsu, makes it such that any high energy particle interacting inside will be caught (as it must for their background rejection). Famously, this experiment caught neutrinos from Supernova 1987A, which were likely the first ever astrophysical neutrinos measured and identified.

In the transition from the late 1980's to the early 1990's, the field of particle astrophysics, especially UHECR astrophysics, began to resemble its current form. Through the interest of Nobel laureate James W. Cronin, and other particle physicists, as well as X-ray astronomers, the goal of figuring out the acceleration mechanisms of UHECRs came to the front of the field. Accordingly, a number of sensitive and precisely executed experiments were built; for the sake of brevity, I will highlight two: the Akeno Giant Air Shower Array (1990-2004) and CASA-MIA (1990-1997). The Akeno Giant Air Shower Array (AGASA) was a  $100 \text{ km}^2$  array in Japan at the Akeno Observatory consisting of surface and buried scintillation detectors and was built with the explicit intention of observing cosmic rays of  $10^{17} \text{ eV}$  or higher. Some infrastructure and detectors were already installed as early as 1984 from other ventures [**agasa**]. AGASA paved the way for the Auger Observatory, but was the direct predecessor, along with Fly's Eye, to the Telescope Array [**ultraray**].

CASA-MIA, or the Chicago Air Shower Array-Michigan Muon Array, set out with a slightly different objective. Instead of looking for the showers caused by hadronic primaries, the objective of CASA-MIA was to explore the arrival directions of Very High Energy (VHE) gamma rays. In the optical to low gamma ray energy regimes, only direct detection of photons is possible, usually by balloon- or space-borne detectors, but gamma rays above 10 GeV cause air showers much like UHECRs except scaled down and of much lower muon content and higher  $X_{max}$ . To detect these, about one thousand surface stations consisting of 4 scintillation counters (CASA) each, topped with an ancient lead (i.e. low radioactivity) sheet to encourage pair production in shower photons, and about one thousand more buried scintillation counters (MIA) were employed in a .23 km<sup>2</sup> array [[casamia](#)].

The basic idea is to catch the showers from  $\tilde{1}0^{14}$  eV -  $10^{16}$  eV gamma rays, and identify hadronic showers of similar energies through their muon multiplicity. Perhaps ironically, the CASA-MIA team provided much of the expertise and drive for the construction of Auger, but the detector itself is more of a functional predecessor to the High Altitude Water Cherenkov Experiment (HAWC), a gamma ray observatory which uses air shower techniques and direct particle detection. Through the CASA-MIA experiment, much was learned about how to set up a distributed array and how to design effective detection logic.

## **UHECR Physics Today**

This more or less brings us to the beginning of the current era of UHECR physics, which is dominated by the Auger Collaboration, of which I am a member, and the Telescope Array (TA). A more detailed description of both experiments will be given in subsection 3.3.1 and subsection 3.3.2, respectively, but to give context, the Pierre Auger Observatory started development in the mid 1990's and began construction in 2000 in Malargue,

Mendoza Province, Argentina, at the turn of the millennium. Operation began in 2004, and has continued through to today. The TA experiment began development around the same time as Auger began operation, and is located at Dugway Proving grounds, in a site expanded around the Fly’s Eye experiment. Both apparatuses measure UHECRs of  $10^{18}$  eV+ throughout the bulk of the array, and feature low energy infills for engineering tests and to expand their sensitivities to energies as low as  $10^{16}$  eV.

## 1.2 Motivation and Goals

Broadly, the work in this document aims to advance the progress towards an upgraded Pierre Auger Observatory, dubbed *AugerPrime*. Auger has successfully shown that air shower arrays can cover massive areas to obtain exposures large enough to precisely fill in the highest energy portions of the cosmic ray spectrum. Its upgrade is primarily motivated by a two part case. First, the composition of UHECRs is a vital measurement in moving the field of charged particle astronomy forward. By understanding the composition of the highest energy cosmic rays, we can work to find the rigidity of each particle as it propagates to the Earth. With this information,  $Z$  dependent anisotropy studies may elucidate, or help elucidate, the mystery of the origin of UHECRs.

The other component of the motivation in upgrading Auger, is a modernization effort. Inexpensive and powerful electronics have developed over the past 15 years since Auger was built, and in order to ensure the continued operation, and to take advantage of increased performance, the entire electronics package is being upgraded. The scintillation and radio detectors that are to be integrated into the upgraded surface detectors will need additional data handling and processing abilities. This electronics upgrade also supports the composition measurement goal, by adding the facilities necessary to measure added channels of

incoming data as well as a much more powerful onboard processing and programmable logic system. Adding a flexible programmable logic device will allow us to modify the functionality, especially the trigger logic, in situ.

In particular, the work in this document will concern the design of the programmable logic, the operating system software, the time-tagging logic, and the hardware and software integration of the new GPS model. Supporting these design tasks, I will also review verification data for the time-tagging system to determine the overall time resolution of the upgraded detector and show the results of testing GPS receiver models for the upgrade. Continuing along this path, results created using branches of the AugerPrime time-tagging hardware will also be included (e.g. quantification of spatial dependence of GPS timing errors, a miniaturized in situ timing system for AugerTA, etc.).

To round out the work included, I will also be reporting preliminary results from a follow-up to Auger's recent correlation of UHECR arrival directions with Starburst Galaxies. In this category, I will be using software written by the High Energy Astrophysics group at CWRU to execute and manage simulations of UHECR back propagation through the JF12 magnetic field model. We will simulate a meaningful portion of the available arrival direction data, and use a  $\chi^2$  test to infer the significance of the distribution of SBGs over an isotropic distribution.

# Chapter 2

## Cosmic Ray Sky: The State of the Field

In this chapter, the all particle spectrum will be reviewed via the famous Swordy plot, which will lead us through a brief discussion of its main features and their phenomenology. After discussing cosmic rays of all energies, I will explore how they reach their energies. This will start with the lower energies in earlier sections and move to the highest energy sources by the end. Finally, I will provide a discussion on how sources are constrained and the possible origins of UHECRs.

### 2.1 Spectrum

Cosmic radiation in the broadest sense concerns all particles which enter Earth's atmosphere from space. The spectrum of cosmic radiation (or cosmic rays) as measured spans about 12 orders of magnitude in energy and many more in flux. At the lowest end of the spectrum, the sources are relatively clear from power concerns; in particular they are believed to be injected by supernovae and supernovae remnants [**stanev**]. Detectors of lower energy (below the knee) cosmic rays, see temporal modulation of the flux based on the solar magnetic fields, an effect commonly called “solar wind”. Moving up in the energy,

## Cosmic Ray Spectra of Various Experiments

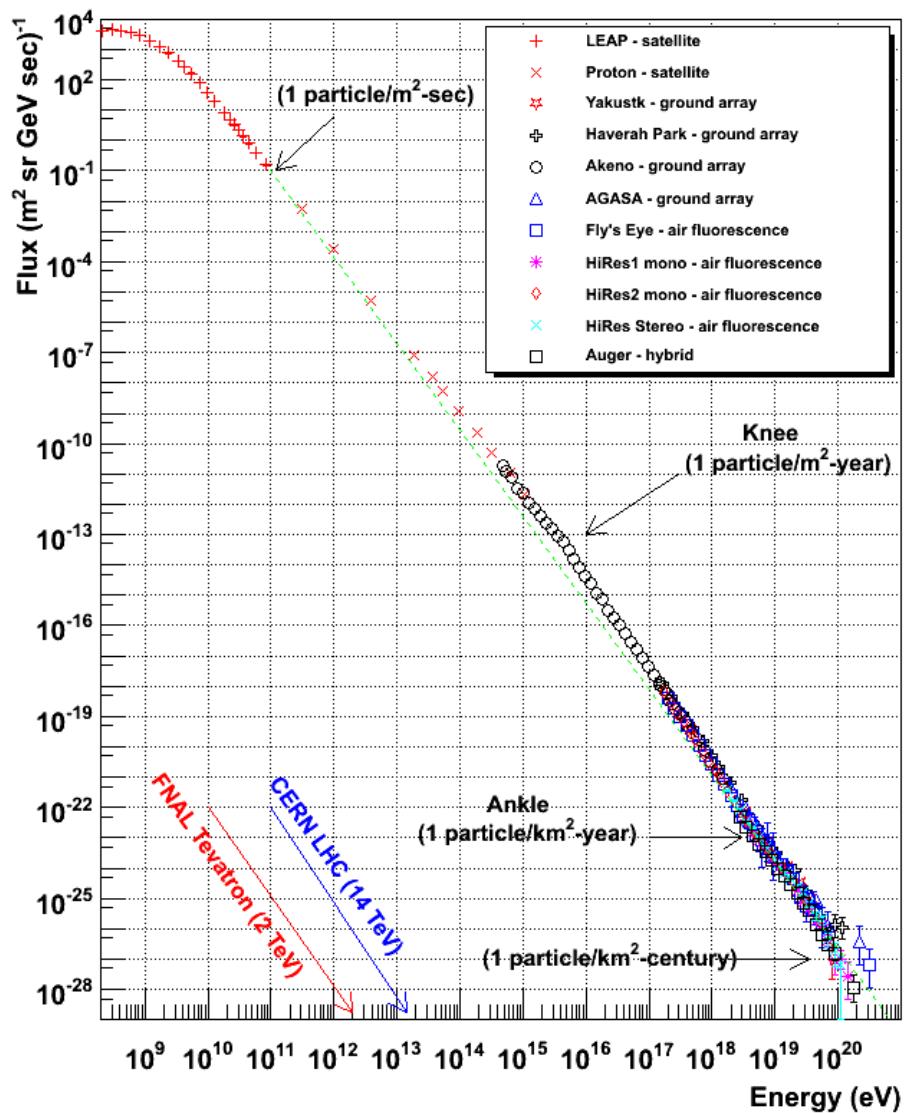


Figure 2.1: The all particle spectrum, often called the Swordy Plot, after Simon Swordy, updated and put together in `swordyplot` `swordyplot`.

we see two distinct breaks in the all-particle spectrum (pictured in Figure 2.1). At slightly under  $10^{16}$ eV, we see a break at what is called the “knee” into a steeper (harder) spectrum, and then a softening at the “ankle”.

Below the knee (lower than  $10^{16}$ eV), it is widely agreed that supernovae and supernovae remnants make up the sources of the largest portion of cosmic rays [**stanev, crapp**]. At these energies, charged particles are deflected far too much to be a reliable marker of sources, but gamma rays especially along with neutrinos and x-rays paint a fairly clear picture of supernovae remnants being the most probable source [**stanev, sean, foteini**]. Around the knee but before the ankle, the sources are slightly less clear since this is above the cutoff of major gamma ray and neutrino experiment’s sensitivities, but is below the sensitivities of UHECR detectors. In this regime, Fermi shock acceleration (often just Fermi acceleration, or the Fermi mechanism) provides a good explanation for the shape of the spectrum and the particular physical mechanism of acceleration. Further explanation of Fermi acceleration will follow in subsection 2.1.1. Additionally, some of these particles may be from the same sources as  $10^{18+}$ eV cosmic rays, but accelerated less efficiently, or with less charge.

At the ankle and above, we enter the territory of Ultra High Energy Cosmic Rays, which the main body of work in this thesis concerns. It is worth mentioning at this point, the final feature in the spectrum, which is the approximate cutoff at the GZK limit of  $5 \times 10^{19}$ eV. The Greissen-Zatsepin-Kuzmin (GZK) limit, is a particle-physics-based upper limit on the energy of long traveling protons. Observationally, this limit is obfuscated by the measured mixed composition of cosmic ray primaries, at which point one can think about it naively as a “per proton” energy limit [**futuregzk**]. Sources in this energy regime become more mysterious, and will be the topic of section 2.2.

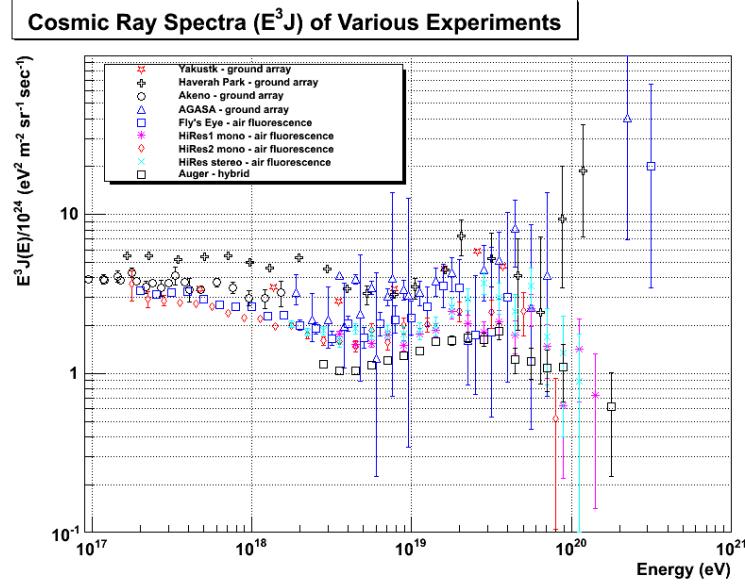


Figure 2.2: The high energy spectrum, multiplied by  $E^3$  to allow for greater discernibility.

### 2.1.1 Fermi Shock Acceleration

Fermi acceleration plays an important role in shaping the cosmic ray spectrum and, as noted in [crapp], provides an easy transition from below the knee to above it. The essence of shock acceleration, is that moving regions of relatively high magnetic field can turn around and effectively accelerate particles. These regions must be relatively sparse as collisions in them will thermalize particles, and the mechanism is specifically an explanation for non-thermal particles. These magnetic mirrors can take a number of different forms, although they are believed to usually be the bow shocks of large objects. For example, shock acceleration can be seen in the bow shock of even such a mundane astrophysical object as the Earth [shocks]. In Fermi's original 1949 paper, he begins by considering a plasma cloud.

Before discussing some useful and convincing features of Fermi acceleration in the first-order, let's first go through and show the energy gain from such a scenario, following the procedure in Chapter 11 of **crapp**. First, it should be made clear that while the simple

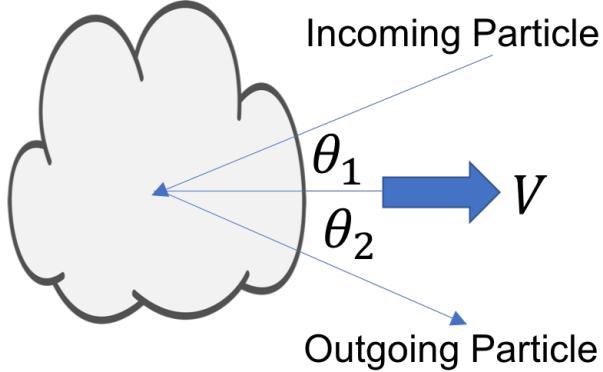


Figure 2.3: A simple diagram of the important quantities in Fermi acceleration.

calculation performed here depends on only the velocity of the plasma cloud (or bow shock) and the incoming and outgoing angles of the particle, it is assumed that inside the cloud, the particle can be deflected many times through smaller diffuse “collisions” with effectively random internal magnetic fields. Now, looking at the energy of the incoming particle in the frame of the plasma cloud, we have:

$$E'_1 = \gamma E_1 (1 - \beta \cos \theta_1).$$

where  $E_1$  is the incoming particles energy in the lab frame, and  $\gamma$  and  $\beta$  are the relativistic factors as commonly used in the literature of special relativity, and  $\theta_1$  is the angle between the plasma cloud’s velocity and the particle’s velocity as measured in the lab frame. In what may seem like an obvious statement, magnetic fields do no work, or equivalently the magnetic field “collisions” are completely elastic, and therefore the total energy of the particle on the way out in the cloud’s frame is the same as that on the way in, and so in the lab frame:

$$E_2 = \gamma E'_2 (1 + \beta \cos \theta'_2),$$

where all is as above less that  $\theta'_2$  is the particle’s outgoing angle as measured in the frame

of the cloud. Combining these two for a hyper-relativistic particle (i.e. assuming the dispersion relation for a photon), the change in energy can be written as:

$$\frac{\Delta E}{E} = \frac{(1 - \beta \cos \theta_1)(1 + \beta \cos \theta'_2)}{1 - \beta^2} - 1. \quad (2.1)$$

From here, we need only average appropriately over both angles. In the case of  $\cos \theta'_2$ , considering the particle can exit at any angle, and in this model none are preferred,  $\langle \cos \theta'_2 \rangle = 0$ . Phenomenologically, it is assumed that the probability of a deflection is proportional to the relative velocity between the cloud and the particle. This would come naturally by considering the ions within the cloud as little magnetic deflector dipoles. From this, the angular distribution of the deflected particle would be:

$$\frac{dn}{d \cos \theta_1} = \frac{c - V \cos \theta_1}{2c},$$

where we allow any incoming angle. This gives  $\langle \cos \theta_1 \rangle = -V/3c$ , for a final result of:

$$\xi = \frac{1 + \frac{\beta^2}{3}}{1 - \beta^2} - 1 \approx \frac{4}{3}\beta^2. \quad (2.2)$$

Here, we give  $\Delta E/E$  the convenient name  $\xi$ , for the fractional change in energy. This is perhaps the most basic but central result of the theory of second order Fermi acceleration. Given that  $\beta$  is strictly nonnegative, Equation 2.2 shows that simple interactions, which are elastic in the frame of the plasma cloud, can give energy to particles in the lab frame. Here, we have made the assumption that the cloud will be non-relativistic, although the treatment can be extended to relativistic clouds. In fact, when this treatment is extended to relativistic sources, they can gain as much as a factor of  $\gamma^2$  energy from their first interaction, which could be very large [**shocks**]. It is also common to do this exercise for sheets of magnetic

deflectors, resulting in first order Fermi acceleration.

There is nothing in particular which stops a particle from undergoing this process multiple times, even within the same region, in which case we have the relation:

$$E_n = E_i(1 + \xi)^n,$$

with  $E_n$  the energy after the  $n$ th cycle, and  $E_i$  the initial energy. From here, we can go a number of ways. By assuming a time per acceleration cycle, we can begin to consider how the age of the plasma cloud affects the maximum energy of a particle accelerated by it. Furthermore, it is hardly a stretch to invoke a probability of re-acceleration, and from this a power law spectrum comes out.

## 2.2 Sources At The Highest Energies

As mentioned previously, the sources of the highest energy cosmic rays ( $E > 10^{18}$ eV) are by no means obvious. It is still a very active topic of research, and the conclusions that are currently considered relatively concrete are more categorical than specific. To outline today's thinking on the topic, I will first discuss two of the easier conditions that are imposed on potential sources, namely power conditions (i.e. can a source actually produce enough UHECRs within a reasonable efficiency) and the Hillas Criterion, along with the Hillas plot. Briefly, the Hillas criterion compares the magnetic fields and radii of astrophysical objects in a magnetic containment scenario, to give a constraint on the energy of particles they could produce.

After discussing the two basic criterion, I will move on to a more specific discussion of the two primary acceleration scenarios, namely top down and bottom up, and then expand on the bottom up scenarios, which are currently more feasible (I will also elucidate that

statement). Finally, I will include a discussion of some of the hard facts about sources as we know them, namely the anisotropy of the highest energy cosmic rays, and their composition measurements.

### 2.2.1 Power Considerations

By considering the density of cosmic rays through integrating the spectrum, we can put a constraint on their energy density in the galaxy. Let's take, for ease of calculation, Gaisser's rough estimate of an energy density of  $\rho_E = 1 \text{ eV/cm}^3$  [**crapp**]. We then need to know the volume of the galactic disk, which can be estimated as :

$$V_D = \pi R^2 d \approx \pi \times (15 \text{ kpc})^2 \times 200 \text{ pc} \approx 4 \times 10^{66} \text{ cm}^3.$$

Knowing this, we find the power by taking the total energy contained in cosmic rays, and dividing it by the average confinement time (a result of the leaky box model), also known as the mean residence time,

$$P_{CR} = \frac{V_D \rho_E}{\tau_R} \approx 5 \times 10^{40} \frac{\text{ergs}}{\text{s}} = 5 \times 10^{33} \text{ W}.$$

Here, we have  $\tau_R$  as the mean residence time of a cosmic ray in the galaxy and I have broken the calculation out of the somewhat archaic ergs/s into watts. Continuing to follow Gaisser's treatment of the topic, a 10 solar mass ejection originating from a type II supernova with an average velocity of  $5 \times 10^8 \text{ cm/s}$  and a frequency of 30 years gives a power of  $P_{SN} = 3 \times 10^{42} \text{ ergs/s}$  or  $3 \times 10^{35} \text{ W}$ . With what would seem to be a fairly low efficiency, supernovae can easily power all of the observed cosmic rays below the knee. This, however, says nothing about the physical mechanisms of acceleration it simply gives a necessary but

Energy	Power (erg/s)	Power (W)
$> 100 \text{ TeV}$	$\sim 2 \times 10^{39}$	$\sim 2 \times 10^{32}$
$> 1 \text{ PeV}$	$\sim 2 \times 10^{38}$	$\sim 2 \times 10^{31}$
$> 10 \text{ PeV}$	$\sim 5 \times 10^{37}$	$\sim 5 \times 10^{30}$

Table 2.1: Here the power required by cosmic accelerators to account for portions of the spectrum with increasing energies are given. These calculations are out of [crapp].

not sufficient condition. Further analysis shows that there is no obvious mechanism for supernova blast waves to accelerate cosmic rays to  $> 10^{18} \text{ eV}$ , outside of their possible contribution to diffusive shock acceleration. Therefore, the calculation presented here pertains only to cosmic rays under approximately 100 TeV.

Looking at higher energies using the same technique, we see the requirements tabulated in Table 2.1. These are for rays of up to 10 PeV, and for higher energies, particularly those  $> 1 \text{ EeV}$ , a calculation beyond the scope of this work is required. That said, the point here is that power constraints can be an important requirement to rule out sources before looking for an actual physical explanation for their acceleration mechanism. This technique is commonly applied to low and mid energy cosmic rays, but can be used for UHECRs in more advanced computational models.

## 2.2.2 Hillas Criterion

Examining the relationship between a source's ability to confine particles and the highest energy particles it can produce gives us the venerable Hillas Criterion. First published in 1984, Hillas looked at the Larmor radius, i.e. the radius in which a particle of constant velocity is confined in a magnetic field, and considered the constraints on astrophysical accelerators that this gives [hillas]. This resulted in the Hillas plot or diagram, which has become a central tenant of the field [stanev]. Following the original treatment in his paper,

the Larmor radius for a hyper-relativistic particle is:

$$r_L = 1.08 \times \frac{1}{Z} \left( \frac{E}{10^{15} \text{ eV}} \right) \left( \frac{B}{1 \mu\text{G}} \right) \text{ pc},$$

where  $r_L$  is the Larmor radius,  $Z$  is the atomic number,  $E$  is the energy of the cosmic ray in question and  $B$  is the magnetic field of the potential accelerator. From here we can turn this equation around to give us the maximum energy from a particular accelerator, however we will need to look after a couple of details first. In order to truly account for the accelerating ability of an object, we have to account for the movement of the magnetic centers within the object. This involves correctly introducing a factor of  $\beta$ , and furthermore there is a factor of 2 that shows up in the criterion due to the radius being half the length of the circular accelerator. Putting these together, we arrive at:

$$\left( \frac{B}{1 \mu\text{G}} \right) \left( \frac{L}{1 \text{ pc}} \right) > \frac{2}{Z\beta} \left( \frac{E}{10^{15} \text{ eV}} \right),$$

in which  $L$  is the length of the accelerator, and  $\beta$  is the relativistic speed factor of its constituent shock producers. Turning this on its head and using the radius (in turn cancelling the 2), as is common practice, we arrive at what is now widely accepted as the Hillas condition or criterion (which does not explicitly appear in the original paper, [**hillas**]):

$$E_{\max} = \beta Z e \left( \frac{B}{1 \mu\text{G}} \right) \left( \frac{R}{1 \text{ kpc}} \right) \text{ EeV}. \quad (2.3)$$

In this particular version of it, out of [**stanev**], it is rewritten in EeV so as to be more useful in the context of UHECRs. From here, we can make a log-log plot in magnetic field versus radius and draw contours on it to represent the energy of particles produced along the contour. Such a plot is included in Figure 2.4.

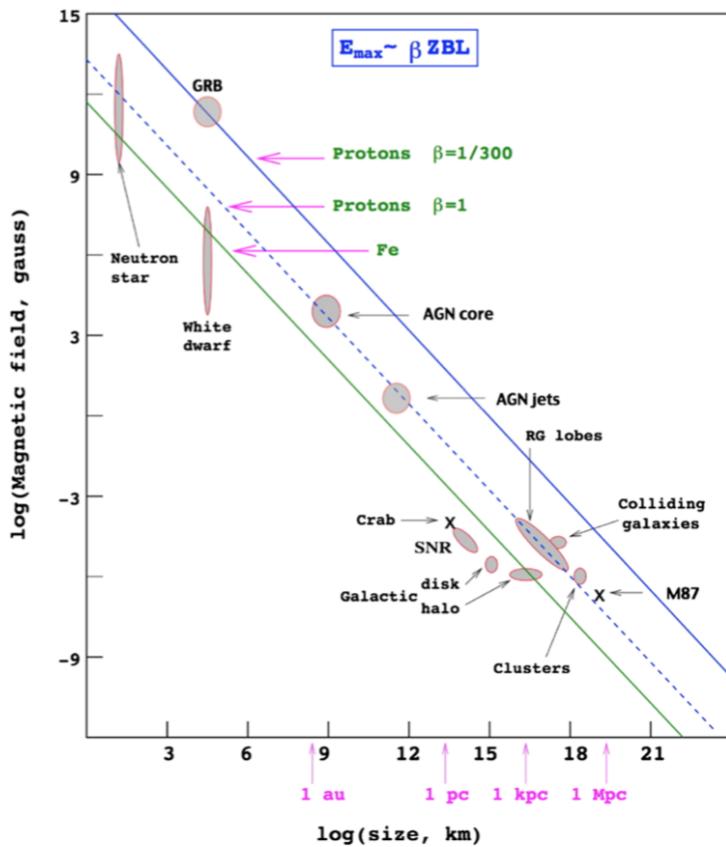


Figure 2.4: The famed Hillas Plot from [stanev], showing possible UHECR acceleration candidates organized by their magnetic fields and radii. The contours drawn here represent 100 EeV and the  $\beta$ s are those of the diffuse shock acceleration fronts of which the accelerator would be composed.

Again, it is important to emphasize that this is a necessary but not sufficient criterion for a cosmic accelerator to reach ultra high energies in its output spectrum. Examining Figure 2.4, it we can see that many astrophysical objects are ruled out at the highest energies, however the contours on this particular plot are drawn rather high at 100 EeV. That said, drawing contours at this energy appears to be the tradition started by Hillas in his original 1984 paper.

### 2.2.3 Top Down vs. Bottom Up

In the drive to understand the sources of UHECRs, there are effectively two over-arching categories of proposals. Those which involve the decay of supermassive relic particles, inflationary topological defects, or other Grand-Unified-Theory level particles, and those which invoke only the standard model of particle physics or less to explain acceleration via statistical or large astrophysical phenomena.

#### 2.2.3.1 Top Down Models

Many top-down models exist as the particles that they require are often natural byproducts of exotic theories in cosmology and particle physics. In a top down theory, a particle or topological defect decays and produces a spectrum of particles. In many models these decays should release in the range of  $10^3$ - $10^7$  EeV of energy [stanev].

Topological defects are proposed to have originated in fluctuations during phase transitions in the early universe. As the energy density of the early universe drops through inflation, models of topological defects predict it necessary to have regions where, in order to preserve causality, energy will be trapped. While these regions come in dimension 0-3, only magnetic monopoles of approximate dimension 0 and cosmic strings of dimension 1 should decay/break symmetry to produce UHECRs [stanev]. Some of the issues with

such models include that the dynamics of the strings and monopoles become important, and that there should be modulation/suppression of the GZK cutoff. In some models, the cosmic strings and monopoles must actually meet (in the case of monopoles, it must meet its anti-particle), and given their rarity, this greatly suppresses the flux of UHECRs compared to what is observed [**tds**]. It is also common in such models, to expect the topological defects to aggregate in the galactic halo. If this were true, then (in most models) the spectrum should not cutoff at extremely high energies, which is contrary to what is observed [**swordyplot**].

On the other hand, relic particles, a form of non-thermal cold dark matter, could have been formed in a freeze-out of the early universe, perhaps at the surface of last scattering, and would then wander the universe in a metastable state. At some later time, they would decay into some quantum-number-conserving mix of particles. The energies estimated here are generally around  $10^{12}$  GeV, and the particles would be affected by gravitational pull [**tds, stanev**]. Given this, it is expected that they would agglomerate at the centers of galaxies, in which case distinguishing them from AGN lobes/flares as accelerates could prove difficult.

Most importantly for both models, is that they would both be expected to produce large amounts of ultra high energy gamma rays and ultra high energy neutrinos. This is a serious problem given the current best results of the relevant experiments, namely Auger, TA and Hi-Res [**futuregzk, foteini**].

### 2.2.3.2 Bottom Up Models

Bottom Up models of UHECR acceleration invoke only known physical phenomena, and inside of that almost exclusively electromagnetic mechanisms. As is startlingly clearly laid out in the relatively recent review by **stanev**, there are truly only two physical mechanisms

in this category: shock/statistical/Fermi acceleration and so-called “one-shot” acceleration.

Shocks are outlined and discussed in subsection 2.1.1, but to review, any sort of abrupt region of fairly high magnetic field (compared to its surroundings) can act as a region of shock acceleration. Through stochastic processes, particles are accelerated by what are effectively considered point deflectors. While energy is not gained or lost in each interaction in the shock “cloud’s” frame, the particle experiencing deflections can ultimately gain energy through cycles of such deflections.

One shot acceleration mechanisms (also sometimes called inductive acceleration) largely consist of “compact spinning magnetized” objects, which in effect is the long way of saying rotating young neutron stars and in particular magnetars. In this regime, an object simply produces a terrifyingly large electric field, which is not uniform by terrestrial accelerator standards, but is uniform enough, to take an entering particle and boost it up to UHECR energies in one shot (hence the name). There are also some scenarios where regions of radio-loud Active Galactic Nuclei (AGNs) can produce large enough electric fields to accelerate particles to ultra high energies, but these are oft constrained by the density of particles surrounding them (high particle density surrounding an accelerator implies likely collisions and therefore thermalization).

Since a detailed description of each possible source scenario is perhaps a tedious endeavor for both the reader and the writer, a table is provided below to describe the important aspects of the major (and some minor) source scenarios. The data in this table is compiled from the very straightforward review by Letessier-Selvon and Stanev ([stanev], 2011), from the review by Berezinsky ([tds], 1999) and most importantly the thesis by Oikonomou ([foteini], 2014) which is probably the most straight forward review of bottom up source candidates I have seen.

Source Type	Shock / Inductive	Matter Density Issues	Description	Main Issues
AGN (radio-quiet)	Shock	Yes	AGNs in general are a rather appealing possible source being relatively abundant and very energetic.	Radio Quiet AGNs rely on shocks at the end of their jets, however the matter density is too great for them to be likely UHECR producers.
AGN (radio-loud)	Both [shearedjets]	None in the extended radio lobes	Radio loud AGNs, and particularly those of type FR-II can have both shock acceleration sites and inductive acceleration sites where there is little matter density to stop acceleration.	There are fewer of these than radio-quiet AGNs, and so it should be harder for them to account for the total flux of UHECRs [radioagn].
BL Lac Objects (subclass of Blazars)	Likely Shock, Possibly Inductive	Yes	Blazars are generally AGNs which are eating away at their accretion disks. Different configurations give different levels of variability in their emission. They are generally visible in the very high energy gamma-ray spectrum. A fairly recent paper by Murase et al. ([muraseblazar]) points out that these can satisfy the Hillas Criterion but does not firmly pin down a physical mechanism of acceleration.	The two main issues are that BL Lac objects are bright because of the matter they accrete, and so there are some issues with thermalization here. Additionally, they only barely meet the Hillas Criterion to get to the highest energies.
Gamma Ray Bursts (GRBs)	Shock	Not significant	Gamma Ray Bursts are believed to be the product of a relativistic fireball in a super-luminous supernova. Since 1995, these have been pointed at as a tantalizing source of UHECRs, however they are surrounded in minor controversy as they were first introduced when the insufficient AGASA data came out, which indicated a flat spectrum and no cutoff at the highest energies [waxmangrb, staney].	GRBs should produce large amounts of neutrinos in their interactions, since much of the relativistic fireball they are believed to consist of is leptonic. IceCube has, at this point, set constraints which make these somewhat less favorable [foteini].
Waves in the Interstellar Medium (ISM)/ Intergalactic Medium(IGM)	Shock	No	A variety of phenomena are capable of producing large shocks in the interstellar medium, or even in the relatively dense intergalactic medium around superclusters. Galactic mergers are considered a likely source of IGM shock waves, while supernovae and other star formation adjacent processes have been suggested to cause shocks in the ISM local to their Starburst regions [sbgshocks].	Definitive proof of shocks that satisfy the Hillas Criterion has not yet been shown for the ISM in Starburst regions, although many sources say there will be qualifying shocks in the similar but less dense magnetohydrodynamic fluid of the general ISM. Murase et al. in 2008 [muraseshocks] showed that the IGM can qualify for producing UHECRs around the ankle, although not at the highest energies.
Magnetars/ Pulsars	Both, but primarily inductive	No in most cases	Pulsars, a subclass of which are magnetars, are rapidly rotating neutron stars and are the primary candidates for inductive acceleration. From a theoretical standpoint, these are particularly promising as they can be shown to produce a spectrum similar to observations [pulsaruhecrs].	The viability of rotating neutron stars as UHECR sources is at the mercy of neutrino observatory data. In the near future, they could be ruled out if no high energy neutrinos are found [pulsarno].

Table 2.2: Here I've summarized some of the possible sources of UHECRs and discussed their advantages and disadvantages as models.

## 2.2.4 Composition

From a first principles perspective, the composition of cosmic ray primaries is one of the most important features of both sources and showers. However, the difficulties in determining the composition of primaries via the signal from surface detectors or even fluorescence telescopes are numerous. This will be discussed in greater detail in section 3.2. That said, by looking at the maximum of particle production, denoted  $X_{\max}$ , a statistical determination can be made, in particular over many data points, to find the average composition per energy. Doing so has been one of the triumphs of the Auger Observatory, and the data as they stand are a point of contention amongst contemporary experiments. A trend worth

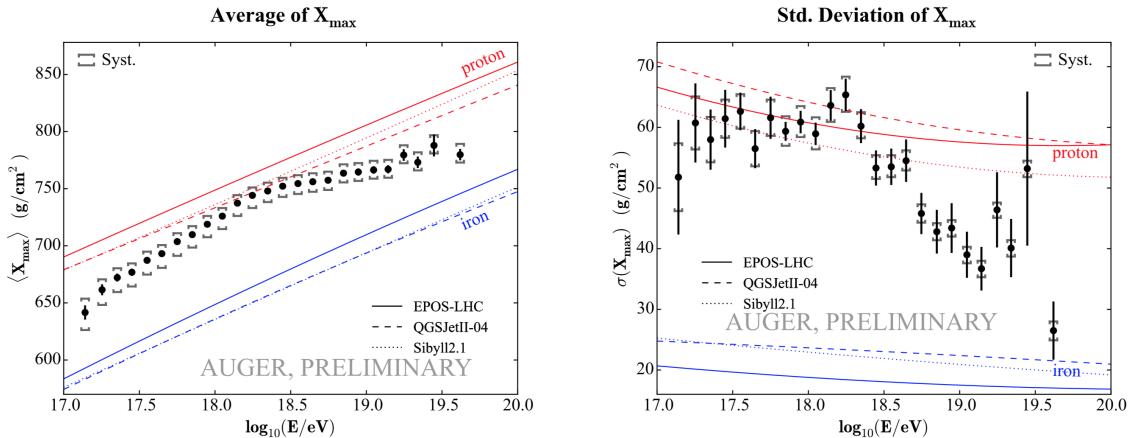


Figure 2.5: Left: the averages of  $X_{\max}$  as a function of energy with contour lines drawn for their values based on 3 of the premier hadronic interaction models. Right: the standard deviation of  $X_{\max}$  as a function of energy, with contours drawn similarly to those on the left. This figure is taken from an Auger Collaboration publication ([[futuregzk](#)])

noting in Figure 2.5 is the tendency for higher energy cosmic rays to be heavier in composition. If we look back at both the shock and inductive models of acceleration, both depend on  $Z$ , the atomic number directly, and so we expect this heavier composition at higher energies if these are correct.

One of the goals of the next generation of UHECR experiments is to better determine

the composition of primaries. While Auger and TA currently attempt to find the composition on a per event basis, the errors are large enough that composition dependent anisotropy studies are not possible. Being able to determine composition on an event by event basis would be a major advantage since it gives the ability to select high energy proton events, which should be deflected less than higher  $Z$  nuclei in galactic and extragalactic magnetic fields.

## 2.2.5 Anisotropy

ven since the first days of UHECR physics, one of the main questions has been whether the origin of UHECRs is galactic or extra galactic. In the late 90's, some conclusions were attempted with AGASA, Haverah Park, Fly's Eye and even Volcano Ranch data [tds]. Recently, the Auger Collaboration has published findings in Science ([anisotropy]), which show that at the  $5.2\sigma$  level, there is a dipole anisotropy in the arrival directions of UHECRs pointing away from the galactic center. This result represents the most definitive proof to date that the origin of UHECRs is extragalactic.

The analysis represented in Figure 2.7, essentially breaks down the sky along the direction of right ascension (i.e. direction of the Earth's rotation, effectively corresponding to galactic longitude) into a decomposition of cosines and sines of different frequencies. An example of how the data fits to these decomposing basis functions is given in Figure 2.6.

The important result here is that the maximum of the dipole is pointing away from the

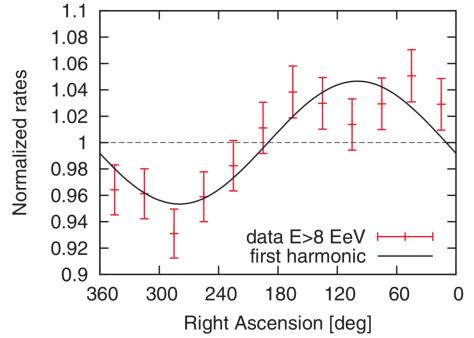


Figure 2.6: Pictured here is the first harmonic in the anisotropy decomposition done by the Auger Collaboration in [anisotropy]

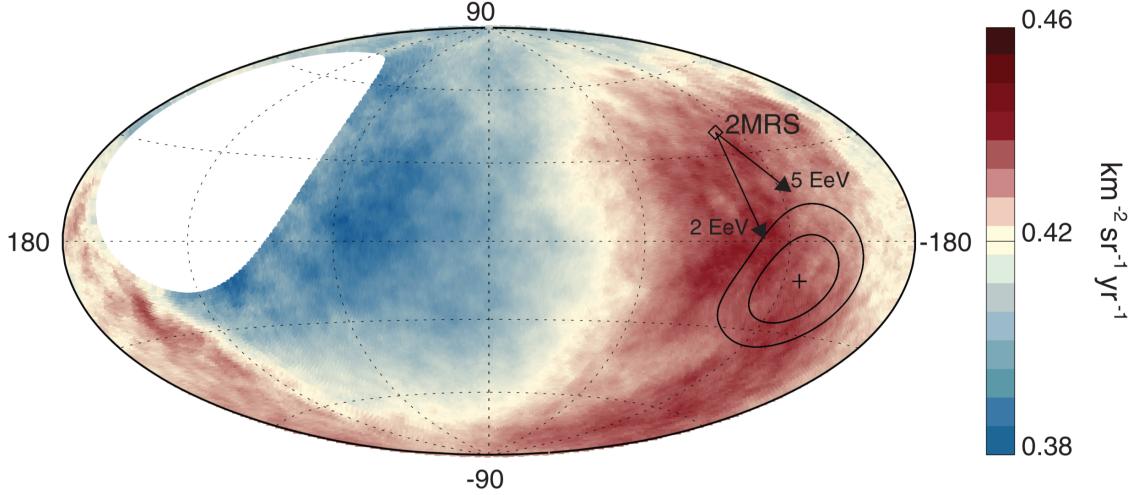


Figure 2.7: Pictured above is the sky map of UHECRs with energy  $> 8$  EeV smoothed by convolving with a  $45^\circ$  top hat function. This figure is taken from an Auger Collaboration publication ([\[anisotropy\]](#))

galactic center, which implies that the origin of most UHECRs is extragalactic. For all of the source models I've introduced, less the top-down ones, this is effectively assumed, however the extragalactic nature of UHECRs should not be taken for granted as it was a hotly debated topic for decades [[hillas](#), [tds](#)].

Additionally, as mentioned above, the “holy grail” of anisotropy studies would be a composition and energy dependent anisotropy study with high statistics, which is one of the probable final products of the AugerPrime upgrade.

## 2.2.6 Concluding Thoughts: Cosmic Ray Spectrum and Sources

Through decades of work, the cosmic ray spectrum has been carefully measured up to energies approaching  $10^{21}$  eV, and as the exposure of the current generation of detectors increases, I expect that more features of the spectrum will be elucidated in much the same way that the “second knee” has recently been. Furthermore, I expect that at least some of the mysteries regarding sources will be opened up by the coming upgrades and advances in technology, in particular the radio technique, the AugerPrime upgrade, CTA and IceCube Gen2. The nature of UHECR source candidates is such that they rarely only emit in the cosmic ray band, they also should be one or both of neutrino emitters and gamma ray emitters. With the advent of multi-messenger and higher aperture gamma ray and neutrino observatories, the field will get even closer to determining the correct source model.

Additionally, I would like to comment that many in the field look at finding a single source model as the “Occam’s razor” solution to the problem of “where do UHECRs come from?”. It seems to me and others that it may in fact be simpler if we consider that there are likely multiple types of sources with slightly different output spectra, but similar acceleration mechanisms. Since diffuse shock acceleration can happen at almost any size scale given a decently large magnetic field, there are likely multiple ways for a particle to reach the highest energies through bottom up models.

Finally, as far as top down models are concerned, I should note that it would be very interesting to look at the number of UHECR shower events as a function of long time periods. Perhaps 3 to 4 generations down the line, radio, optical or particle telescopes will have the aperture needed to observe showers on other planets, or perhaps other solar systems. If the origin of UHECRs truly is somehow attributable in full or in part to the decay of particles, then the simple laws of radioactive decay should apply, and one should be able to observe a decreasing exponential behavior in the number of events over time.

# **Chapter 3**

## **Extensive Air Showers and Cosmic Ray Detectors**

Through this chapter, I plan to give a description of some older and some more modern theories of how to quantify the important aspects of the air showers caused by Ultra High Energy Cosmic Rays. This will lead into a discussion of the various methods that have been employed to detect extensive air showers, and an overview of a few important experiments that employ them.

### **3.1 Extensive Air Showers**

When an atomic nuclei or single proton of  $10^{18} + \text{eV}$  (frequently called the primary) comes crashing into the upper atmosphere, particle physics at the very edge of humanity's current understanding ensues. After a violent disk of plasma created by slightly-beyond-the-verified-standard-model processes cools out into high energy Large Hadron Collider level interactions, the shower is carried down in phenomena quantified by a clever mix of particle physics and statistical mechanics. The first  $10^{-15}$ -or-so seconds of this process (after

the extrapolated standard model physics) are described by the basic principles of particles physics, and so the treatment in this chapter will start there. After this point, we typically become more concerned with how the particles “transport” from one species to another, and how much energy is deposited in the nearby air. The rate at which particles are moved from pure kinetic energy of the prompt secondaries into matter in the later secondaries is a main parameter of cosmic ray showers and its maximum,  $X_{max}$ , is perhaps the best known predictor of the primary particle’s composition.

After the shower undergoes its maximum of particle production, we usually become more concerned with the observable parameters on the ground, which can be seen by typical particle counting or calorimetric detectors. These observables are, in the case of the surface detector stations of Auger, derived from the signal recorded by one or more types of ground based detection. From this, a number of important observables can be extrapolated, however the two of central importance that I would like to mention are the muon-to-electron ratio ( $N_\mu/N_e$ ) and the lateral density function (LDF).  $N_\mu/N_e$  can be derived from a single analog to digital converter (ADC) trace, or more accurately derived from multiple ADC traces from two or more types of detectors, ideally with different responses to muons and electrons, such as a scintillator and a Water Cherenkov Detector (WCD).

In this chapter, I will give overviews of this process at each level. On the topic of particles physics, I will present a treatment which discusses the steps for calculating decay widths and cross sections of the electroweak force, but does not delve into the group theoretic fundamental structure underlying the modern theories of particles physics. It should be noted, the mathematics used to describe these extensive air showers (or just “showers”) was derived for lower energy primaries. After the initial higher energy processes, much of it is still entirely valid, but to tie the chapter into how this work is actually done, I will include a discussion of the premier air shower simulation package, CORSIKA, which is al-

most always used instead of hand calculations for modern experimental design, verification and data analysis.

### 3.1.1 Particle Physics Basics

To start from the beginning of the story, particle physics as we know it today, stems from the 1928 equation by Dirac [**dirac**]:

$$i\cancel{d}\psi = m\psi. \quad (3.1)$$

Outside of  $i$ , the square root of negative one, and  $m$ , the mass of the particle, each of these objects will need an explanation. The first,  $\cancel{d} = \gamma^\mu \partial_\mu = \gamma^\nu g_{\mu\nu} \partial^\mu$ , is the four dimensional partial derivative operator contracted with a set of the “gamma matrices” ( $\gamma^\mu$ ). There are a number of bases we can put these in, but the two most common are the Dirac basis and the Weyl basis. The Dirac basis is used for most calculations, while the Weyl basis is typically used when discussing the chirality of particles, in which case it can be used to decouple the Dirac equation into two easily solved second order differential equations.

Going back to Equation 3.1,  $\psi$  is an object called a bi-spinor, and represents the particle’s wavefunction. A spinor, is a vector with two complex entries which transforms under the SU(2) group. Frequently, these spinors are required to have magnitude 1, and so a bi-spinor is a four complex entry vector that mechanically looks something like:

$$\begin{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \end{pmatrix},$$

a mathematical structure which is required by the Dirac equation. In general, we impose this normalization condition on the wavefunction  $\psi$ :

$$\bar{\psi}\psi = \psi^\dagger\gamma^0\psi = 1.$$

Above, the  $\dagger$  represents the hermitian conjugate, and  $\gamma^0$  is the zeroth member of Dirac's gamma matrices. In connection with Schrödinger's quantum mechanics, the entries of these bi-spinors are frequently simple plane waves and represent both the spatial and temporal behavior of the particles.

From a utilitarian perspective, these bi-spinors and the results of the Dirac equation become useful when seen through the standard model Lagrangians, Feynman diagrams, and ultimately the path integral formulation [**crapp**]. While the path integral formulation is certainly the most powerful tool for calculating cross sections and decay widths, a discussion of how to use it is outside the experimentally-focused nature of this work. Let's proceed by discussing some of the basic processes that occur in the detection of astrophysical particles from a basic particle physics view.

The Lagrangian is, at the lowest level of complexity, a statement of a classical systems total energy. When we apply this concept to particles of the standard model however, we see that it becomes a statement of the basic possible interactions available through each force [**cottingham**]. For example, when we look at the Lagrangian density statement for the weak force as it pertains to interactions between electrons and neutrinos, we find a term for the charged current interactions of electron neutrinos and electrons, and its quantized operator structure like this: In Figure 3.1, the  $e$  and  $\nu$  terms are bi-spinors, as discussed earlier. The electroweak Lagrangian density,  $\mathcal{L}_{EW}$ , contains permutations of such interaction terms for each possible tree level (two vertex) interaction that can occur via the electroweak

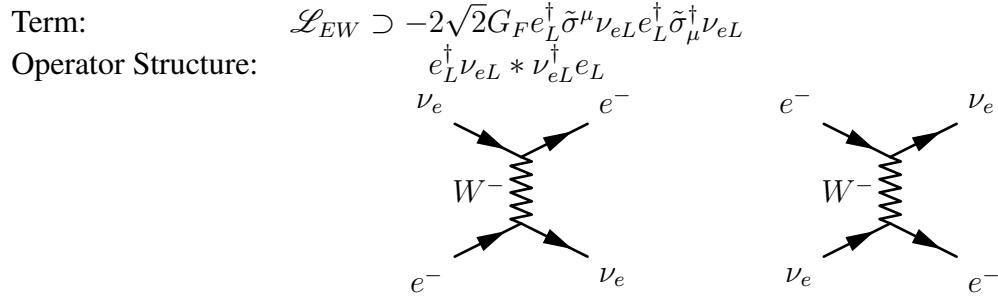


Figure 3.1: Here are two Feynman Diagrams and their corresponding terms in the electroweak Lagrangian density, along with the term's operator structure.

force. The interactions shown above involve neutrinos scattering off of electrons. These Feynman diagrams can be turned into a cross section calculation by tracing each vertex through to the end and multiplying each route together, and adding up the cross sections found by all possible topologically distinct permutations of the incoming and outgoing particles. Each vertex has a different term which we use to represent it, and this allows us to complete the calculation.

Once the expression for the diagram is assembled, we use the fact that the expression must produce a scalar quantity, and then invoke the Casimir trace theorem, which exposes the final result's sole dependence on the trace from the long expression of bi-spinors and gamma matrices. A number of these tricks are very well explained in **elempart**, which stands as a surprisingly good resource for many levels of students and researchers.

While walking through a calculation may be just outside the scope of this introduction, I will leave the steps summarized below:

1. Draw diagram
2. Write down scattering amplitude,  $\mathcal{M}$  by tracing through each outgoing particles path backwards, including the appropriate vertices and propagators as you go
3. Multiply  $\mathcal{M}$  by its Hermitian Conjugate to obtain  $|\mathcal{M}|^2$

4. Insert  $\gamma^0\gamma^0$  next to adjoint ( $\dagger$ ) operators to ‘undagger’ them
5. Commute  $\gamma$  matrices until you find a scalar term, i.e.  $u(p_i, s)\dots\bar{u}(p_j, s)$
6. Permute the bi-spinors in the scalars ( $u(p_i, s)$ , etc.) to be next to each other, so you can sum over the spins to make the spin-summed square magnitude  $\langle|\mathcal{M}|^2\rangle$
7. Take the traces and contract their results to produce dot and cross products of momenta (this can be done by FeynCalc [**feyncalc1**, **feyncalc2**])
8. Express the inner propagators in terms of outgoing momenta
9. Calculate kinematics, which directly translates to the arguments of the delta functions in  $d\text{LIPS}_n$
10. Insert  $\langle|\mathcal{M}|^2\rangle d\text{LIPS}_n$  into the scattering amplitude
11. Calculate the integral numerically (this can be done with Mathematica).

Additionally, I will note, as in the instructions above, I advocate using the FeynCalc ([**feyncalc1**, **feyncalc2**]) package in Mathematica to complete these calculations if they are not for pedagogical purposes. It is certainly a rite of passage to complete a decay width or scattering cross section calculation by hand at some point, but for research purposes, modern symbolic computation system are advanced enough to save a researcher a large amount of time. The experimental values of many cross sections can be easily looked up in the Particle Data Group webpage (<http://pdg.lbl.gov>, **pdg**).

### 3.1.1.1 Moving Particle Physics into Shower Physics

As we move from purely particle physics concepts into those that are applied to extensive air showers, we need to adopt a new formalism. Cosmics ray physicists studying showers,

generally wrap up the particle physics aspects into various forms of inclusive cross sections, where we take a number of integral averages over quantities which are relevant to the particle physics, but less relevant to the transport of particles through the atmosphere.

As treated by Gaisser in [crapp], we start by effectively pulling apart the cross sections we can calculate or measure from particle and collider physics (respectively) into the quantity  $\langle n_{ac}^{(b)} \rangle$ , which is the mean number of particles of type  $c$  produced in an interaction between particles of type  $a$  and  $b$ . This is a function of the energy of the interaction between these particles, and from a deeper perspective, a function of the transverse momentum in the interaction, as well as the ratio of the energy deposited into the product to the energy of the primary particle. Such averages exist for different numbers of particles interacting to create different multiplicities of products.

To begin working towards the statistical portion of extensive air showers, in the form of transport equations, we need to finish wrapping the particle physics into convenient quantities. The next step after taking cross sections and turning them into  $\langle n_{ac}^{(b)} \rangle$ , is to define the quantity  $F_{NN}(E, E')$  as:

$$F_{ac}(E_c, E_a) \equiv E_c \frac{dn_c(E_c, E_a)}{dE_c},$$

which is the unit-less inclusive cross section for a particle of species  $a$  to interact and produce  $dn_c$  particles of species  $c$  within an energy bin the size of  $dE_c$  around an average energy deposited into the outgoing particle  $E_c$  with incident energy  $E_a$ . This quantity has already been appropriately integrated over transverse momentum. These  $F_{ac}$  cross sections are useful in a number of calculations, and it is here where it becomes important to introduce the concept of correlated and uncorrelated fluxes.

In the history of cosmic ray physics, people have been concerned with particles of

energies where air showers are induced and observable, but also with energies where a detection method would never be able to tell whether two particles stemming from the same interaction were related in any way. The former, air showers, are considered a *correlated flux*, where an observer on the ground would be able to tell that the particles they are seeing come from the same primary interaction. However, many interesting phenomena and hints of fundamental physics are contained in the interactions from particles that are not high enough energy to cause correlated fluxes. In these uncorrelated fluxes, we consider the production that a portion of a species' spectrum induces. Each interaction that produces final state observable particles is not important, it is the total spectrum that we observe on the ground that makes the observable quantity in uncorrelated fluxes.

In general, we can extend the treatment covered herein from correlated fluxes to uncorrelated fluxes by invoking the spectrum weighted moment, which effectively tells you how a spectrum of particles of a particular type produce how many particles of another type. To this end, the transport equations for correlated and uncorrelated fluxes are closely related by whether or not we treat  $F_{ac}(E_c, E_a)$  via its own contribution integral in a transport equation, or if we further wrap it into a *spectrum weighted moment*. These moments make what Gaisser calls “approximation A”, which is effectively that all of the relevant quantities, but in particular the unit-less inclusive cross section, do not depend on energy. This is, of course, untrue over large energy scales, but it is easily believable over the energies at which uncorrelated fluxes are relevant.

The spectrum weighted moment is calculated as:

$$Z_{ac} = \int_0^1 (x_L)^{(\gamma-1)} F_{ac}(x_L) dx_L,$$

where  $x_L = E_c/E_a$ , is frequently used in the transport equations we are about to discuss,

when extending them to uncorrelated fluxes.

### 3.1.2 Transport Equations

Once the particle physics has been appropriately handled, the production of particles through the atmosphere is determined by transport equations, which effectively describe the production of particles from each species to the others as a function of depth. For example, a basic transport equation which describes the production of pions from nucleons follows:

$$\frac{d\Pi}{dX} = - \left( \frac{1}{\lambda_\pi} + \frac{1}{d_\pi} \right) \Pi + \int_0^1 \frac{\Pi(E/x_L) F_{\pi\pi}(E_\pi, E_\pi/x_L) dx_L}{\lambda_\pi(E/x_L)} \frac{x_L^2}{x_L^2} + \int_0^1 \frac{N(E/x_L) F_{N\pi}(E_\pi, E_\pi/x_L) dx_L}{\lambda_\pi(E/x_L)} \frac{x_L^2}{x_L^2}. \quad (3.2)$$

Here, we have a number of characters, but the relevant ones for an explanation of transport equations are  $\Pi$ , the number of pions, and  $N$  the number of nucleons, along with the  $F_{ac}(E_c, E_a)$  style inclusive cross sections [**crapp**]. While this is far too simple of a model to describe what happens in UHECR air showers, it stands to illustrate how these calculations can be handled analytically or numerically. The leftmost term on the right hand side of the equation represents the two modes in which the pion would exit the shower, through either an interaction that would effectively take it out of the shower, or that the pion would decay. The middle term represents the possibility that the pion will scatter elastically (in the particle physics sense) and remain a pion, while the right term represents nucleons colliding into anything (but probably a nitrogen molecule, i.e. other nucleons) and producing a pion.

To extend this system fully, we would write a term for each particle in the shower (after we exit the LHC-level-interactions portion) which can produce pions, and then do the same

for each other species of particles. In simple cases, we can compute these integrals and complete a computation of the abundance profiles of various species through the shower. In more complex cases, and perhaps in the most general case, we would apply a numerical solver to the set of differential equations to find such profiles.

An example applied to uncorrelated fluxes, is the electron neutrino production spectrum as measured by the IceCube experiment [**atmos**]. Through this process, and the extended process for Kaons, neutrinos are produced:

$$\begin{aligned} \pi^\pm &\rightarrow \mu^\pm + \bar{\nu}_\mu/\nu_\mu \\ &\downarrow \\ \mu^\pm &\rightarrow e^\pm + \bar{\nu}_e/\nu_e + \nu_\mu/\bar{\nu}_\mu \end{aligned}$$

for which we can write the muon production spectrum as:

$$\mathcal{P}_\nu = \frac{\epsilon_\pi}{X \cos \theta (1 - r_\pi)} \int_{E_\nu/(1-r_\pi)}^{\infty} \frac{\Pi(E, X)}{E} \frac{dE}{E} + \frac{\epsilon_K}{X \cos \theta (1 - r_K)} \int_{E_\nu/(1-r_K)}^{\infty} \frac{K(E, X)}{E} \frac{dE}{E}. \quad (3.3)$$

The objects in this equation are analogous to those in Equation 3.2, except here we have a term for the production of neutrinos from Kaons. After substituting in the integrals, we can use this to compute an energy spectrum using spectrum weighted moments ([**crapp**]):

$$\frac{dN_\nu}{dE_\nu} = \frac{N_o(E_\nu)}{1 - Z_{NN}} \left( \frac{\mathcal{A}_{\pi\nu}}{1 + \mathcal{B}_{\pi\nu} \cos \theta E_\nu / \epsilon_\pi} + 0.635 \frac{\mathcal{A}_{K\nu}}{1 + \mathcal{B}_{K\nu} \cos \theta E_\nu / \epsilon_K} \right),$$

where,

$$\mathcal{A}_{i\nu} \equiv Z_{Ni} \frac{(1 - r_i)^\gamma}{\gamma + 1} \text{ and } \mathcal{B}_{i\nu} \equiv \left( \frac{\gamma + 2}{\gamma + 1} \right) \left( \frac{1}{1 - r_i} \right) \left( \frac{\Lambda_i - \Lambda_N}{\Lambda_i \ln(\Lambda_i / \Lambda_N)} \right).$$

Where  $\gamma$  is the spectral index less one,  $r_i$  is the Molière radius of a given species and the  $\Lambda$  terms are interactions lengths. Frequently, a more easy to fit form is used in experimental applications, such as this one from an IceCube report on atmospheric neutrinos [**atmos**]:

$$\Phi(E_\nu) = CE_\nu^{-\alpha}(w_\pi + w_K),$$

That said, only the transport equations which do not use spectrum weighted moments, i.e. those for uncorrelated fluxes, are relevant to the propagation of UHECR extensive air showers in the atmosphere (or elsewhere). While these equations were important when researchers truly started to understand the nature of extensive air showers, we should revisit the fact that today careful and vast Monte Carlo simulations are the primary tool for understanding extensive air showers.

### 3.1.2.1 Grammage, Interaction Length and Radiation Length

After touching on the particle and statical physics processes involved in air showers, we need to discuss how the particles in the air shower interact with the atmosphere. Before discussing specific interactions, I would like to first introduce some important quantities dealing with height in the atmosphere and how particles transport through it.

How height is measured in UHECR physics may seem counter intuitive to the outside observer. In terms of how particles interact with the atmosphere, the density of the atmosphere is of primary importance, however, the density can change based on upper atmospheric weather. To account for this, the height in the atmosphere is measured by *vertical depth* (also called grammage, a quantity originally used in papermaking), which is the integrated density of the atmosphere along a line. This quantity can be thought of as how much material one would catch if one dropped a box straight down to a particular height.

Deriving further off of arguments presented in Chapter 3 of Gaisser [**crapp**], we find this relation:

$$X_v = \int_h^\infty \rho(h) dh$$

Now if we take a look at the US Standard Atmosphere of 1976 ([**standardatmos**]), which is still used today, the density profile (Figure 3.3), is not well described by any easy analytical function, and so we rely on approximations to describe it. One such approximation, from

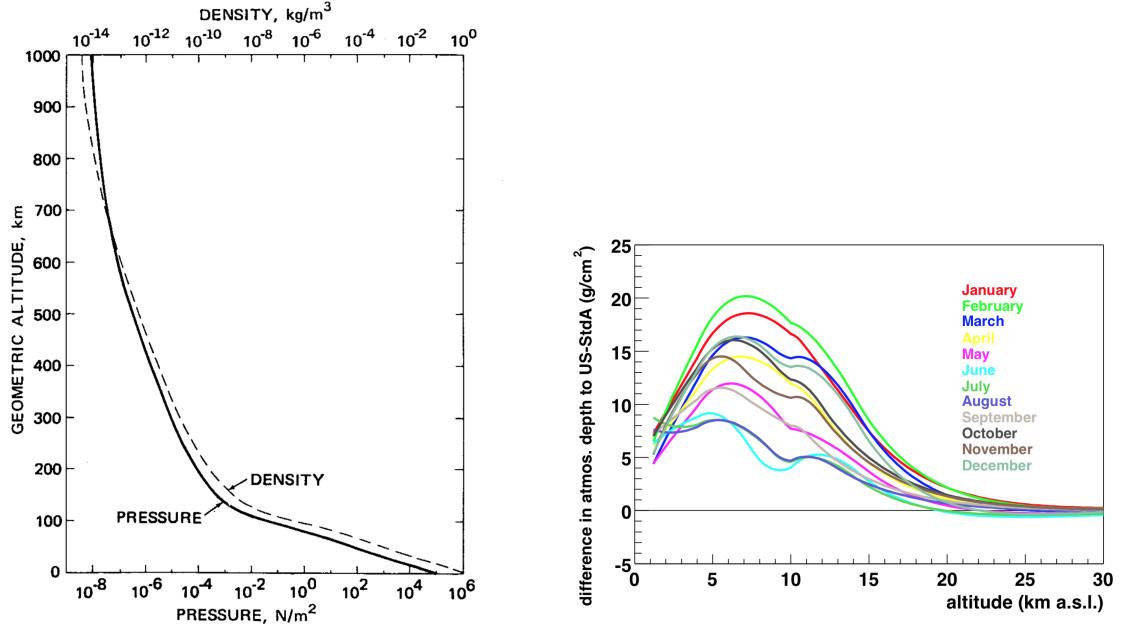


Figure 3.2: Left: Here we have the pressure and density of the standard atmosphere. This could be integrated as a function of height to find the grammage at a given layer of the atmosphere. This comes from the 1976 Standard Atmosphere [**standardatmos**]. Right: Differences in measured vertical depth at the Auger Observatory from the Standard Atmosphere model, from the Auger Collaboration in **augeratmos**.

**crapp2**, is plotted in Figure 3.2. This figure gives the basic properties of vertical depth, namely that it monotonically increases as a cosmic ray approaches the ground. As shown in Figure 2.5, the depth of maximum particle production, the best available predictor of primary composition, is measured in grammage. Equipped with this concept, we can begin talking about interaction and decay lengths.

The interaction length takes into account the probability of a species of particle interacting with anything in the atmosphere, and represents the length at which 1-1/e of a population would have interacted. We may also define interaction lengths for different processes, such as brehmstrahlung, Compton scattering of photons, or ionization. These are calculated as:

$$\lambda_j = \frac{\rho}{n_A \sigma_j^{\text{air}}} = \frac{A m_p}{\sigma_j^{\text{air}}}$$

where  $\rho$  is the density of the air,  $n_A$  is the number density of nuclei of mean mass  $A$ , and  $m_p$  is the mass of a proton. Note that much of the physically interesting part of the interaction length is contained in the cross section, which is the part we modify if we are concerned with a particular process.

The decay length of a particle that has available states to decay into, commonly muons or neutrons, is written as:

$$d_j = \rho \gamma c \tau_j,$$

with  $\gamma$  as the Lorentz factor of the particle,  $c$  the speed of light, and  $\tau_j$  the particle's decay time. Using this idea of defining lengths over which particles can undergo certain processes, we can continue on to defining radiation parameters relevant to cosmic ray showers.

### 3.1.3 Atmospheric Losses

In the energy regime that we are concerned with regarding cosmic ray secondaries ( $\approx 100$  MeV - TeV), there are two types of losses for charged particles that we must concern ourselves with, namely ionization and brehmstrahlung. Ionization does not scale nearly as strongly as brehmstrahlung. Brehmstrahlung is the basic interaction process of two charged particles interacting and ejecting a photon in the recoil. Cherenkov light is a much lower energy loss process. Both ionization and brehmstrahlung can be accounted for in a simple

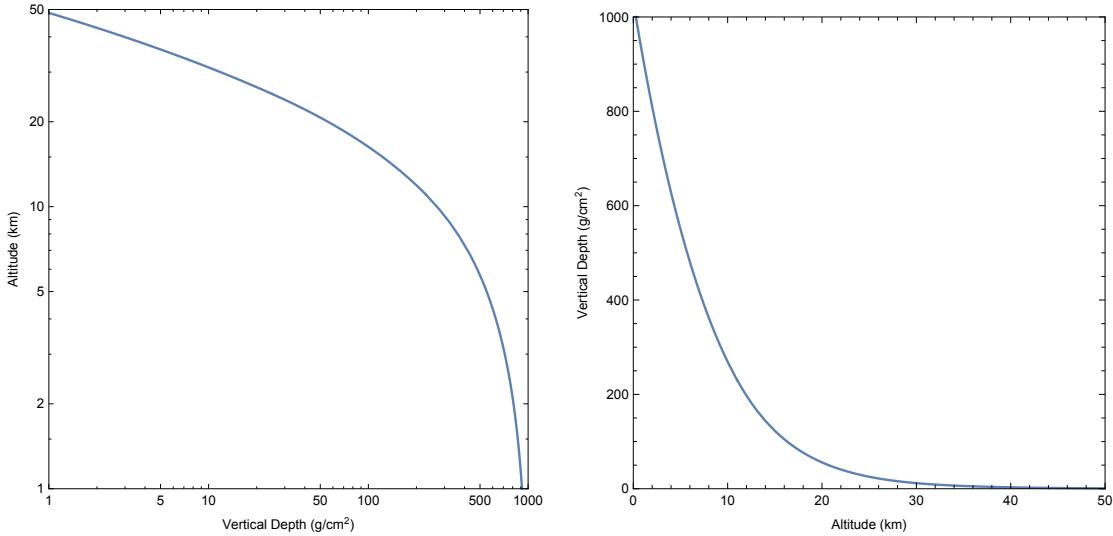


Figure 3.3: Based on the description and approximation in **crapp2**, these are some simple plots of vertical depth versus altitude on logarithmic axes, and the inverse on regular axes. These are presented to give some intuitive context to the discussion.

differential equation from **crapp2**:

$$\frac{dE}{dX} = -\alpha - \frac{E}{X_0}, \quad (3.4)$$

whose general solution is:

$$E(X) = Ce^{-X/X_0} - \alpha X_0.$$

Above,  $\alpha$  is the ionization loss in MeV/(g cm<sup>-2</sup>), and  $X$  is the depth traversed.  $X_0$  is the radiation length, which is material specific. The left term in Equation 3.4 represents ionization and approximates no energy dependence in the process while the right term represents brehmstrahlung. The additive term that arises in the solution is the *cutoff energy*,  $E_c = \alpha X_0$ .

The analogous and most prominent loss for high energy photons is pair production, through which a photon becomes a positron and an electron, which continue to feed the shower while moving the mean energy per particle down. The probability of pair produc-

tion in each photon is  $7/9$  per radiation length. At lower energies, such as medical x-rays, photoelectric losses and Compton scattering dominate [[pdg](#)].

### 3.1.4 Shower Parameters

On its way through the atmosphere, a shower goes through a number of phases, as mentioned in the introduction to this chapter. At the *point of first interaction* the particle undergoes interesting very high energy physics, which can have effects on the shower's dynamics, but on average these effects largely wash out. As the shower leaves it's first couple of interactions, the primary particle species of importance will have been determined, in part at random and in part due to the composition of the primary. The terminology used in the field of extensive air showers (EAS) to refer to the different constituents of the shower, known as components is:

- *Electromagnetic Component* (frequently just *EM component* or *EM cascade*): the component of the shower composed of electrons, positrons and photons. Generally, by the time the shower reaches surface detectors such as those in Auger most of the positrons have annihilated in the atmosphere, leading to a net charge in the shower, an essential feature for radio detection [[schroeder](#)].
- *Muonic Component*: from the initial hadronic interaction at the beginning of the shower, the hard quark-heavy collisions produce charged pions, whose primary decay channel is into muons and neutrinos. This makes the muonic portion of the shower an essential observable for determining the primary's composition. The muonic component effectively serves as a proxy for the number of quarks, and therefore nucleons which initiated the shower.
- *Hadronic Component*: while it is rarely a direct contributor to the final detectible

signal, the hadronic component of the shower, largely composed of pions and some  $K_L$  ( $K$ -long) mesons, is important in some aspects of the shower formation before shower maximum. Interestingly, the largely unobserved hadronic component has been shown to be perhaps the only tell tale sign of Quark-Gluon Plasma in EAS by **danielle**

### 3.1.4.1 Heitler-Matthews Model

In the commonly used toy model by **heitler**, and it's extensions such as [**matthews**], Particles come from the initial interaction and either pair produce or undergo brehmstrahlung at each “interaction length”, which does not necessarily correspond to an actual interaction length as introduced previously. As these generations continue, they reach the cutoff energy where their losses come primarily from lower energy processes that do not contribute to the shower. This is the same  $E_c$  as in subsubsection 3.1.2.1.

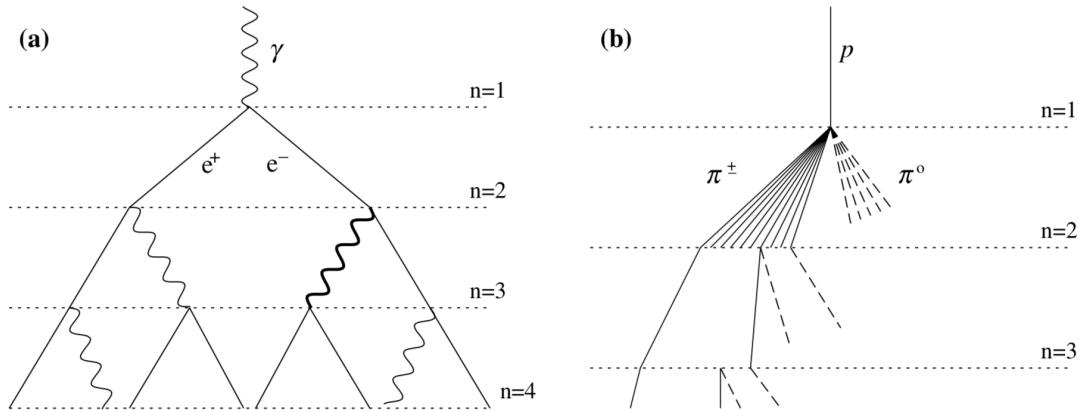


Figure 3.4: Left: a photon-initiated shower in the Heitler model. Right: the extension to hadronic processes by **matthews** (also the source of this diagram).

In the simple Heitler model, diagrammed in frame (a) of Figure 3.4, the maximum of

particle production,  $X_{max}$ , is given by ([**stanev2**]):

$$X_{max} = \lambda \ln \left( \frac{E_0}{E_c} \right).$$

The classical model does not take into account pionization from hadronic processes, however **matthews** extends the simple model to account for this, and is often used as a moderately more accurate account of shower physics, which is much more applicable to UHECR physics. To walk through a form from the Heitler-Matthews model [**matthews**], we start by assuming the average height of first interaction for a proton initiated shower:

$$X_0 = \lambda_I \ln 2,$$

where  $\lambda_I$  is the interaction length of the proton, which is a function of energy effectively proportional to  $-\ln E_0$ . Through arguments about how the hadronic portion of the shower feeds the EM cascade, we can write the production maximum for such a proton shower as:

$$X_{max} = X_0 + \lambda_r \ln \left( \frac{E_0}{3N_{ch}E_c} \right).$$

Here,  $N_{ch}$  is the charged pion multiplicity of hadronic production, i.e. how many charged pions are created per high energy hadronic interaction. The three in the denominator comes from the fact that 1/3 of the produced pions will be neutral, and therefore feed the EM component of the shower. This form can be conveniently rewritten as:

$$X_{max} = X_{max}^\gamma + X_0 - \lambda_r \ln(3N_{ch}). \quad (3.5)$$

This form isolates the pure EM shower maximum,  $X_{max}^\gamma$ , from the hadronic contributions,

which necessarily decrease the depth of interaction, i.e. hadronic showers will always reach maximum production higher in the atmosphere than an equal energy photonic shower. This form is rewritten with elasticity  $K_{el}$  accounted for in **stanev2** as:

$$X_{max} = X_0 \ln \left( \frac{2(1 - K_{el})E_0}{3N_{ch}E_c} \right).$$

The elasticity is the portion of the energy of the leading hadrons (the particles coming out of the initial very high energy interactions) that gets translated into pions. This effectively lowers  $X_{max}$  by stealing energy away from the EM component of the shower and hoarding it until after it is too late to contribute at the shower maximum.

Another important feature of EAS that the Heitler-Matthews model reveals is the *superposition model* or *superposition principle*. Effectively, it is postulated that while Quantum Chromodynamics is extremely difficult to model, we can ignore the inter-nucleon interactions of atomic nuclei primaries and view each nucleon as starting its own shower. In this way, when we consider the showers caused by a particle of atomic mass  $A$  and energy  $E_0$ , we assume, in almost every possible way, that this is the same as a shower consisting of  $A$  subshowers of energy  $E_0/A$ . The modified expression for  $X_{max}$ , again from **matthews**, is (with everything else as in Equation 3.5):

$$X_{max}^A = X_{max} - \lambda_r \ln A.$$

This tells us that, as it is in the case of photons to protons, the shower maximum of nuclei should be strictly lower (i.e. higher altitude) than protons, up to statistical fluctuations. A final quantity of historical and intuitive (although perhaps not mechanical) importance to the topic of EAS, is the *elongation rate*. Early in the study of EAS, Linsley introduced this quantity as the base-10 logarithmic derivative of  $X_{max}$  with respect to energy [**stanev**,

**[crapp2, matthews].** Ultimately, this quantity is used in today's literature to give an intuitive understanding of how much more penetrating a higher energy primary is compared to a lower energy primary, but in the beginning of EAS studies, it was considered to be one of the few quantifiable observable quantities.

### 3.1.4.2 Shower Universality

In recent years, the ideas behind *shower universality* have become increasingly common and important. Shower universality is, in some sense an approximation, and in another a statement, that showers ultimately end up having the same production profiles as a function of depth. Universality extends only to the EM component of showers, and says that the *age parameter*,  $s$  (a function of vertical depth in the shower), and the lateral distance from the core are the only relevant parameters for a shower up to scaling for energy. The definition of the age parameter, which will also show up in the discussion about LDFs (subsubsection 3.1.4.3), is [crapp2]:

$$s = \frac{3}{1 + 2\frac{X_{max}}{X}}. \quad (3.6)$$

Shower universality is motivated by the observation that shower profiles, in particular electron and photon spectra, are approximately the same at  $X_{max}$  for any shower [**universality**]. This is shaped by the processes of production and absorption in each spectra reaching equilibrium (as they must for it to be a maximum of particle production). There is some relatively widespread misunderstanding in the community about the extent to which shower universality is a fact, and to what extent it is a convenient approximation. Of course, it is a mathematical fact that in order to have a maximum of particle production, the competing processes of particle creation and destruction/annihilation (and ultimately removal

from the shower), must reach equality at some point [**crapp2**]. Since the spectra of EM cascades are shaped by absorption and multiplication, it is in this way shower universality is a fact. That, however, does not guarantee that fluctuations induced by longer-lived hadronic processes can't shape the profile somewhat, and that universality completely neglects the production of muons, a tell-tale sign of primary composition. In these ways, it is a useful approximation, which allows for simple but deep analyses in many situations [**universality**].

### 3.1.4.3 Lateral Distribution Functions

Some detection methods, especially those based on particle detection and counting like water Cherenkov detectors and scintillators only record a shower's footprint on the ground, while others, namely air fluorescence and air Cherenkov give lateral profiles of the shower's development [**schroeder**]. That said, particle detection methods tend to have very desirable operational qualities, especially near 100% uptime, lower failure rates, and a lack of moving parts.

To generally describe the lateral distribution functions (LDFs) used in UHECR physics, they tend to depend on  $s$ , the age parameter (Equation 3.6),  $x = r/r_i$ , the distance in Molière radii and some constants, usually related to fluctuations or the total number of particles. A rather complex pair of parameterizations for the electronic and muonic components of the shower are given in **crapp2**, based on work in the 60's by Kenneth Greisen amongst others. These are:

$$\begin{aligned} \rho_{e^\pm}(s, x) &= C_1(s)x^{s-2} (1+x)^{s-4.5} (1+C_2x^d), \\ \rho_\mu(r) &= \frac{\Gamma(2.5)}{2\pi\Gamma(1.25)^2} \left(\frac{1}{320}\right)^{1.25} N_\mu r^{-.75} \left(1 + \frac{r}{320}\right)^{-2.5}. \end{aligned} \tag{3.7}$$

The derivation for the muonic component shown above, assumes that no muons are lost in

the shower. Counter intuitively, however, in order to adjust this LDF for different heights of the shower, we adjust  $N_\mu$  using any number of empirical formulae. Additionally, it is worth noting that the muon LDF cannot, in this form, be adjusted for differently inclined showers. Frequently, before analysis is done using an LDF like this, the shower is effectively “rotated” so it can be considered vertical. The reconstruction used by the Auger Collaboration uses the “Auger LDF”, which is given by:

$$S(r) = \tilde{C} \left( \frac{r}{r_s} \right)^{-\beta} \left( 1 + \frac{r}{r_s} \right)^{-\beta},$$

where  $r_s$  is a characteristic distance used (500m or 1000m in most applications),  $\tilde{C}$  is the effective normalization constant and  $\beta = a + b \sec \theta$  is a shape factor, which can be used to characterize fluctuations and directional arrival of the shower.

Such LDFs exist in numerous forms and for numerous applications. One interesting aside, is the current development of Radio detection LDFs. Since radio detection is a relatively young field, LDFs motivated from a theoretical perspective have not been widely developed. In **schroeder**, an in depth discussion of this is given. Furthermore, **ldfcat** gives a rather excellent account of various LDFs, their parameters, and many other aspects of detection and detector engineering for UHECRs.

### 3.1.5 Monte Carlo Methods: CORSIKA

At time-of-writing, almost all of the aspects of shower and particle physics mentioned in this chapter are, at the professional level, handled by Monte Carlo simulations, and the most consummate package to do so is called CORSIKA [[corsika](#)]. This package is used for showers from the low TeV range to the high EeV range and effectively bypasses the need for statistical accounting of species transport by invoking detailed models of particle physics

at each step. As I understand it, CORSIKA uniquely takes into account certain aspects of the atmospheric transport and kinematics, but outside of these, relies on invoking the best model of interactions at each level. These include EPOS-LHC, Sybil and QGSJET at the absolute highest energies, down through GHEISHA and FLUKA, and then uses GEANT for its modeling of interactions with matter.

Studies for most major detectors in VHE and UHE astrophysics have been done with CORSIKA. Design for Auger used CORSIKA amongst other packages as there was competition amongst competing Monte Carlo codes at the time. Going into the mid 2000's, CORSIKA came to dominate the field and is now the gold standard for EAS simulations. Notably, it was used in validation for Auger (**corsikauger**) and design work for CTA (**ctasika**).

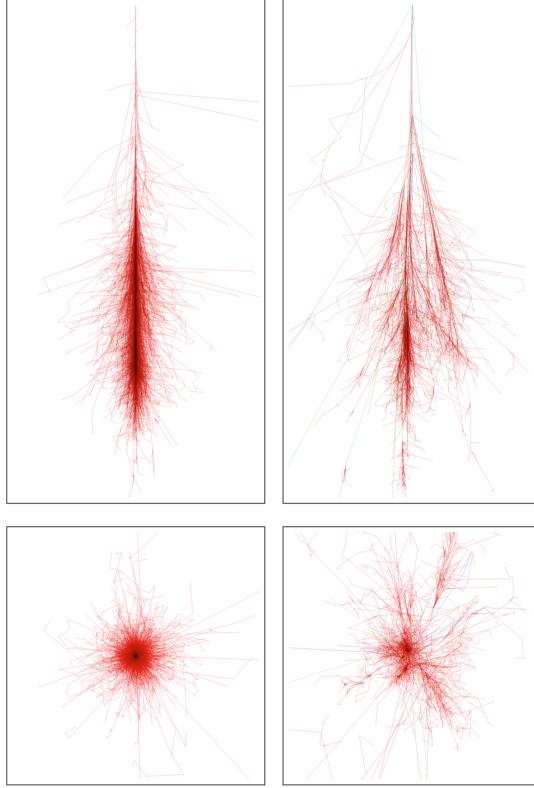


Figure 3.5: Left: A gamma ray shower of 100 GeV, longitudinal profile above, lateral profile below. Right: A proton shower of the same energy and same cross sections in each panel. Note the spread of the hadronic shower is much larger, and if you zoom in more closely, you can see the blue hadron and green muon tracks in addition to the red EM component. These images were published in [steinke](#), but are originally from Fabian Schmidt.

## 3.2 Detection Methods

Over the years since Pierre Auger first took to the Alps and detected coincidences, many different techniques have been used to detect and identify extensive air showers. Some of the first versions of these are covered in section 1.1, while the section below will exclusively cover techniques used successfully by modern experiments. There are a number of techniques currently under development, including radar imaging of the plasma disk left after the first interaction and molecular fluorescence, but for the sake of brevity and accuracy, discussions of these will be left out of this work.

### 3.2.1 Water Cherenkov

Detection of the Cherenkov light made by particles traversing water at speeds greater than the speed of light in the medium is a technique only a couple of decades older than particle physics itself. The first use of Cherenkov light for detecting extensive air showers was by the team led by N.A. Porter at the Harwell Laboratory in England as part of the Atomic Energy Research Establishment (one of the many advances that came out of the greatly increased research funding of the early '50s [[scifund1](#), [firstcher](#)]). He and his team were the first to figure out how to stop bacterial and fungal growth long enough to create a stable and viable detector [[firstcher](#)].

This technique allowed for measurement of a direct proxy of the showers total energy at a point making it a very favorable technique, not to mention that water is much cheaper than a comparable volume of scintillator, and lends itself to detectors with a wide aperture for many zenith angles. It was also N.A. Porter who figured out the fruitful application of highly reflective liners to keep a maximal amount of light in the detection volume, where it might scatter into the photomultiplier tube for detection.

The Haverah Park team used Water Cherenkov detectors in their first proper air shower measurement application, which then served as a stepping stone on the way to deploying the Auger Observatory. As documented in [firstcher](#), Alan Watson, one of the visionaries of Auger, explains that many of the reasons that Porter found Water Cherenkov Detectors (WCDs) useful translated into the decision to use them in Auger.

Physically, Cherenkov light arises from the interactions of fast particles with their medium. As mentioned above, the naive explanation for Cherenkov light, is that it occurs when a charged particle is traveling faster than the speed of light in the medium it is traversing. Many texts and educators like to make the analogy that it is like the shockwave which causes sonic booms in air, and although this analogy is useful, it is truly completely

different dynamics which govern these physical processes.

As a charged particle moves through water at any speed, its interactions with the molecules cause slight changes in the polarization of the medium, however when the phase velocity of the light is greater than that allowed by the medium, the medium emits light in effective concentric spherical shells. This light, however, destructively interferes within what would be referred to as the particles “Mach cone” and, from a naive point of view, constructively interferes in the forward facing Cherenkov cone. An excellent and very in depth discussion of this is given in chapter 23 of **zangwill** and a diagram of some of the relevant quantities is shown in Figure 3.6. The angle of the Cherenkov cone is:

$$\sin \theta_c = \frac{1}{n\beta} \approx \frac{1}{n} \quad (3.8)$$

I make the above approximation as the particles we are discussing, even at low energies

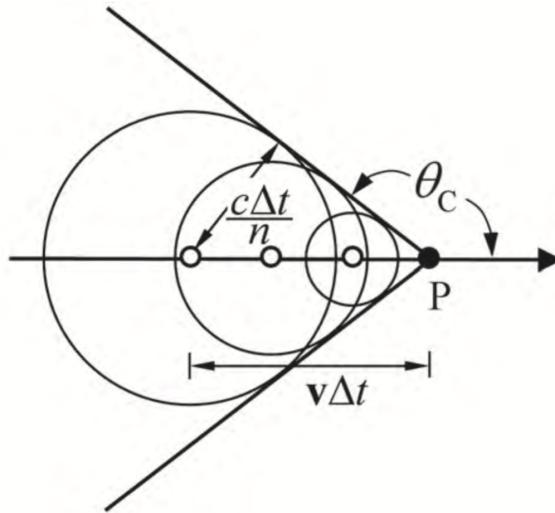


Figure 3.6: Pictured here is a diagram from **zangwill** showing some of the relevant quantities in computing the fields for Cherenkov light emission. Particularly prominent here is  $\theta_c$  and the Mach cone, which is the complementary angle to  $\theta_c$ . The  $v$  is the velocity of the particle and the circles represent the spherical emissions of the medium.

compared to the scale of UHECR showers, are approximately moving at the speed of light (of course, for some phenomena how close their speed is to  $c$  is important, here it is not). This angle can be very small, but luckily for the case of WCDs, this light is emitted inside the detector enclosure and reflected upwards by the liner (the air Cherenkov technique is not so lucky in this regime). This Cherenkov emission is constant as the particle traverses the detector and so longer traversals will leave larger signals to be seen by the photomultiplier tubes.

The Auger WCD is diagrammed in Figure 3.16 and with a radius of 1.6m on the top and a height of 1.2m on the sides, it holds about 12 metric tons of water. The most successful technique that has been used to purify the water in Auger tanks is simply to start with relatively pure water, let the microbes and fungi grow in it, and then wait until they have exhausted their food supply. In a light tight tank, there will be no algae or otherwise to fuel an ecosystem and therefore the “gunk” will eventually die, falling to the bottom harmlessly.

Finally, before moving on to scintillation technologies, we must dwell on an important technical point around calibration of a WCD. In Auger, we use a unit called VEM or Vertical Equivalent Muons as a standard measure of the signal produced in an air shower. By making a histogram of pulse heights over many events, one can determine the minimum energy deposited by a muon, which corresponds to it going through the shortest possible path in the detector, or vertically.

### 3.2.2 Scintillation

Scintillation counters or detectors, frequently referred to simply as scintillators, are perhaps the most venerable and longest lasting detection technique. In early experiments, going all the way back to **muon** and earlier, scintillators were used to trigger the cameras researchers used to photograph the events they were detecting. The first prominent use of scintillators

for very high energy air shower detection was by Linsley at Volcano Ranch, where they were used in much the same way they currently are by a number of experiments.

Scintillators are usually made of crystals or plastics, and are luminescent, or emit visible light, particularly when exposed to ionizing charged particles [**scints**]. The scintillators that are useful for particle detection (versus, say, medical or metrology applications) operate on the principles of fluorescence. When a charged particle excites the scintillator, the molecules in the material enter a very shortly held metastable state which is usually lower energy than the initial excitation, and then decay releasing a photon of a wavelength outside of the materials absorption bands. This allows light created by the scintillator to travel through it without having to worry about unnecessary attenuation from processes like the photoelectric effect, which would be the dominant loss in this regime. The minimum energy of the excitation determines the Minimum Ionizing Particle or MIP energy, which is the correct analog of VEM in water Cherenkov.

While scintillators have a number of nice properties, perhaps the best of these for astrophysics is that they are compact and relatively robust, while maintaining the ability to distinguish individual particles under the right conditions. On the other hand, most scintillators degrade over time, and should not be exposed directly to ambient light if avoidable. They also, depending on their form factor, are prone to internal cracking.

Organic scintillators, which AGASA, TA and AugerPrime all use, have short nanosecond or sub-nanosecond response times to though-going particles [**augerscint**, **tadet**, **scints**]. They are extruded from what is, in effect, appropriately doped styrofoam [**fermiscint**]. Due to their extrusion process, they can be manufactured in a number of form factors. Additionally, both inorganic and organic scintillators come in liquid forms, and can be as simple as chemically simple as liquid Xenon (e.g. direct detection of dark matter experiments).

In the AugerPrime implementation, many of the same principles as water Cherenkov

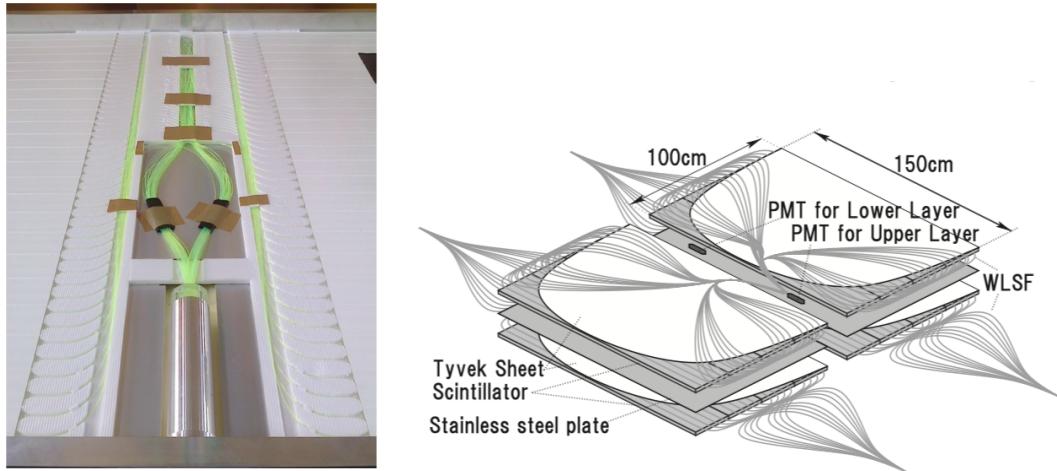


Figure 3.7: Left: A photo of the internal fiber structure and scintillator panels of the SSD for the AugerPrime Upgrade from **toprime**. Right: A diagram of the two layer structure of the TA surface detector's scintillator panels, from Abu-Zayyad et al., 2012 [**tadet**].

are used in the design of the scintillator housing. In particular, a highly reflective titanium oxide layer is used to coat the sides of the scintillator panels, which allows reflection of pulses increasing the fairly short ( $\approx 10$  cm) attenuation length of the material. Birefringent wavelength shifting optical fibers are used to transport the light around the edges of the panel to the photomultiplier tube. The detectors are to be called SSDs or Scintillator Surface Detectors, and will consist of  $4 \text{ m}^2$  of aperture atop each of the 1660 stations in the array [**augerscint**].

### 3.2.2.1 Comparison and Compatibility of Water Cherenkov and Scintillation Detectors

Scintillators and water Cherenkov detectors (WCDs) operate at the highest level in very similar ways. The detector volume, either the water or the plastic scintillator, is connected to a PMT (for the scintillator, it is usually mated using an optical glue or grease, and for the water it is partially submerged). While scintillators do not carry the issues of water purity and PMT suspension, they are more expensive and fragile. Additionally, they are

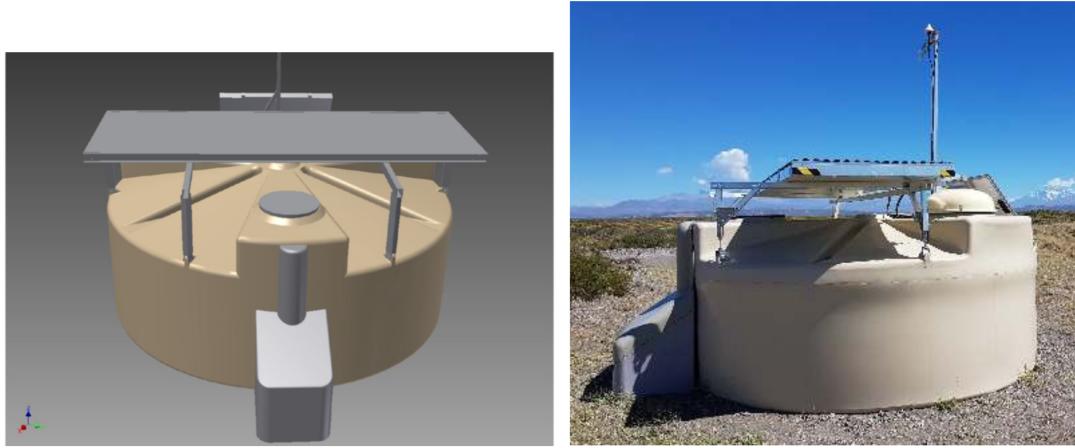


Figure 3.8: Left: a 3D mock-up of the AugerPrime station with scintillators prominently featured on top, from [\[toprime\]](#). Right: A picture of the first SSD in the field from [\[augerscint\]](#).

relatively two dimensional thin sheets and often require a lead layer on top to encourage pair production from the photonic portion of an incoming extensive air shower. WCDs, on the other hand, have a favorable aperture vs. zenith profile, allowing faithful detection and reconstruction of high zenith angle showers (I have plotted the WCD from Auger and the SD from TA’s apertures in Figure 3.9). In defense of scintillators, the deployment process for WCDs frequently involved trucks of purified water, flatbeds which can only carry two or three stations and sometimes even helicopters. Logistically, WCDs can be painful.

Perhaps Figure 3.9 may appear as a relatively unfortunate fact for scintillators, but looking forward, there is a good argument to be made that radio detection and scintillation detection are complimentary techniques. As the radio detection technique advances, upward facing scintillators with high aperture to vertical showers, and radio antennae with high aperture to inclined showers (more on this in subsection 3.2.5) provide similar measurements, i.e. high EM and low muon sensitivity. In this way, perhaps the best detector one could build would utilize all three of these techniques to accurately determine composition with high aperture at all angles and relatively low cost for the performance. Modern

electronics have made handling this data rate possible and the only seriously needed technology is cheap radio repeaters, which exist but have not been readily configured for use in air shower arrays.

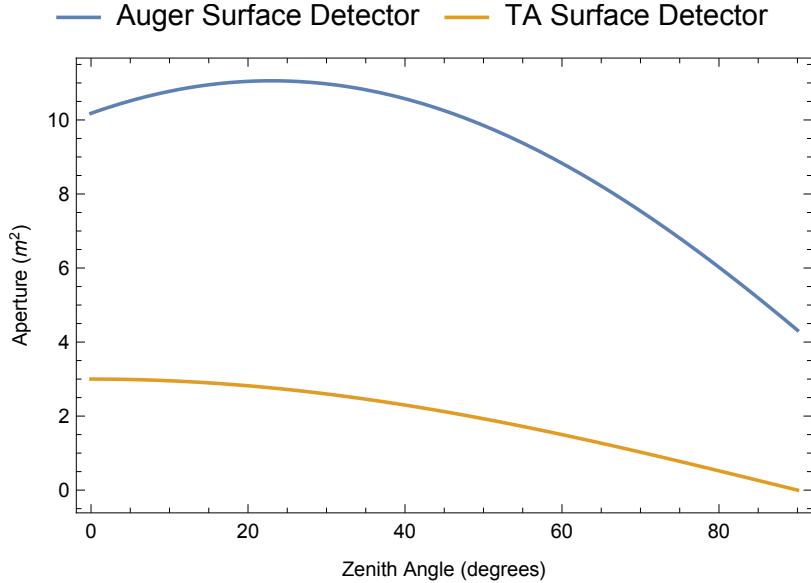


Figure 3.9: Here is a comparison of the aperture as a function of zenith angle for the Auger and TA surface stations. Water Cherenkov detectors tend to allow higher apertures for lower cost, but are logically inconvenient.

Historically and even in contemporary experiments, water Cherenkov and scintillation detectors have been seen as unnecessarily redundant. However, in considering ways to better measure the composition of UHECR primaries, the Auger Collaboration has decided to pursue the use of both detector types in tandem. This allows one to use the different muon and electron efficiencies of water and scintillator detectors to determine the number of muons and electrons that hit a station during a shower. As mentioned previously, it is this quantity,  $N_\mu/N_e$  that is perhaps the best estimator of composition that is available to surface detectors.

### 3.2.3 Fluorescence

Originally pioneered by the Fly's Eye experiment (in conjunction with Volcano Ranch), the very successful Nitrogen Fluorescence technique provides perhaps the most visually relatable detection of UHECR showers. Besides Fly's Eye and its successors, Auger and TA also operate fluorescence telescopes [**ultraray**, **flyseye**].

As charged particles from an air shower traverse the atmosphere, they excite the metastable states of the Nitrogen atom, which in turn emit light in the visible spectrum. The key detail of this process, and the fact which makes the fluorescence technique successful is that the emission of this light is isotropic. Because of this, an array of telescopes can be set up to view a fiducial volume of atmosphere. Since the amount of fluorescence is effectively

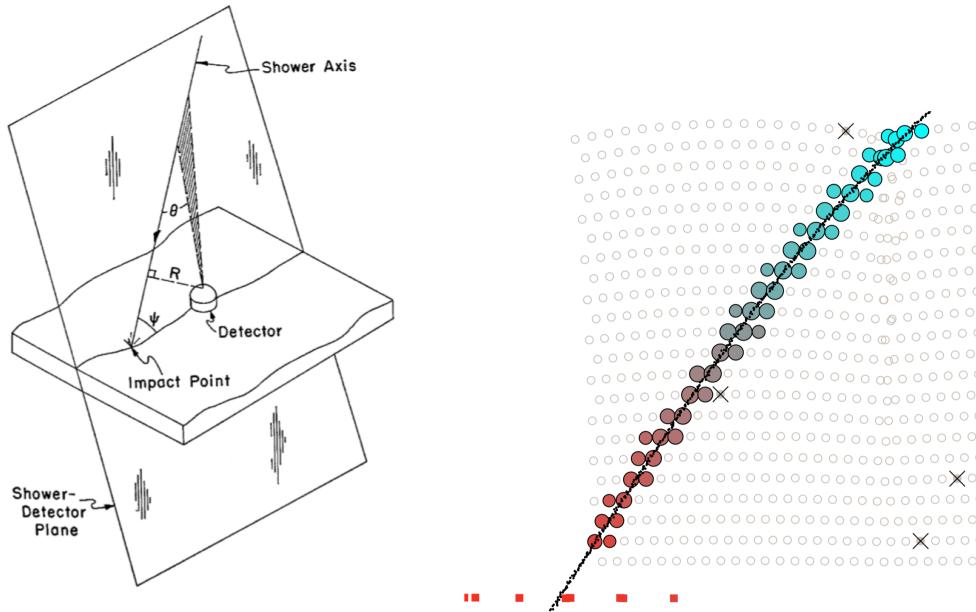


Figure 3.10: Left: A diagram of the principle at work in FD reconstruction. This is originally from the doctoral thesis of a student of Greisen named Brunner, reproduced in [**ultraray**]. Right: A reconstructed track from the Auger FD, taken from [**fluorauger**].

proportional to the high energy part of the EM component of the shower, the fluorescence technique allows for direct observation of  $X_{max}$ . In fact this was one of the driving forces

behind the construction of the Fluorescence Detector (FD) at Auger; accurate measurements of  $X_{max}$  by the FD gives the approximate composition of primaries (averaged over many showers), while the 100% uptime of the WCDs fills out the spectrum.

Technically, fluorescence telescopes work on a very basic reflector design. Usually a large mirror of some sort focuses the light onto the focal plane, which is packed photomultiplier tubes. These are almost universally in a hexagonal tiling [**flyseye**]. In modern applications, many times the primary mirror of the telescope is segmented [**fluorauger**]. This allows for greater field of view and more light collection, while avoiding many of the complications of segmented telescopes since the PMT pixels are so large the point spread function (PSF) of the telescope does not have to be nearly ideal. Accordingly, no adaptive optics or other corrections are needed.

A pitfall of fluorescence telescopes is that, much like optical telescopes, they can only be operated on nights with a clear moonless sky. This brings their typical duty cycle down to the range of 5-10% [**flyseye**, **fluorauger**, **ultraray**]. While this sounds bleak, the lack of exposure from time is made up for with the increased instrumented volume. Furthermore, the aperture of the FD array is a function of the energy of showers. In the Auger FDs, low energy showers are not even collected throughout the entire array due to insufficient light, however  $10^{19} + \text{eV}$  showers are collected even outside of the array in some cases. At  $10^{17.5}$  eV we have  $900\text{km}^2\text{sr}$  and at  $10^{19}$  eV, we have  $7400\text{km}^2\text{sr}$  of aperture [**fluorauger**].

### 3.2.4 Direct and Ice Cherenkov

As we discussed in the Water Cherenkov section, Cherenkov light is emitted any time a charged particle traverses a medium faster than the local speed of light. Since the particles in an air shower are all effectively moving at the speed of light, we need not wait for the shower to hit a relatively high index of refraction medium like water, we can detect the

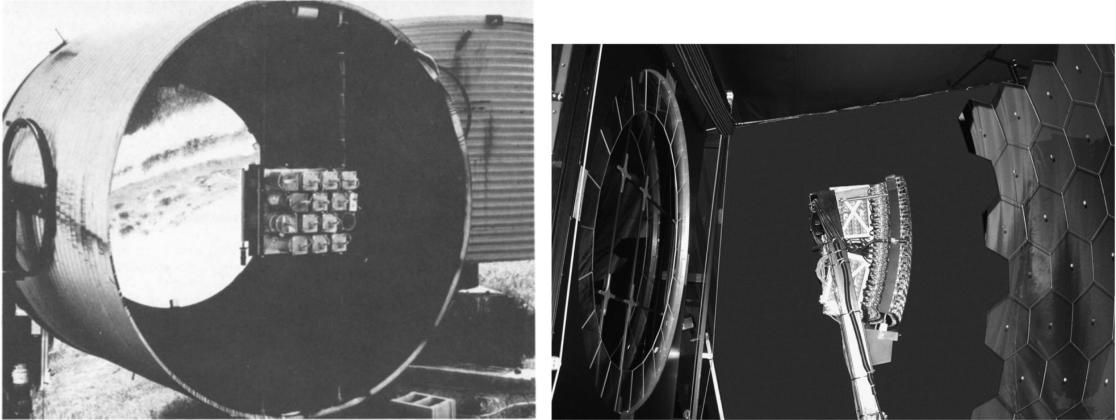


Figure 3.11: Left: A telescope from the original Fly’s Eye detector, published in **flyseye**. In the center, we have the camera module contained in the stainless steel drum. Right: One of the Auger Fluorescence detectors. In the middle of the picture between the aperture on the left and mirror on the right is the honeycomb lattice of PMTs.

Cherenkov light produced from the air. I have also put Ice Cherenkov in this category since, from a hardware point of view, they are very similar techniques.

The technique was first pioneered in the same era as N.A. Porters work at the Harwell laboratory. John Jelley and Bill Galbraith went off of work by P.M.S. Blackett, a Nobel prize winning cosmic ray physicist, and looked for pulses from the night sky with an oscilloscope. Over time, their experiments became more complex and the eventually proved conclusively that they were detecting Cherenkov light produced in air. Interestingly, Blackett was convinced that we would be able to lay back on a dark night and view the Cherenkov pulses with his bare eyes [**firstcher**]. The technique was later used with mild success in air shower physics.

Ultimately, the first very successful application of air Cherenkov was in Imaging Atmospheric Cherenkov Telescopes. In this configuration, gamma rays enter the atmosphere and interact at relatively high vertical depth (i.e. low altitude). They produce almost exclusively electromagnetic showers (as seen in Figure 3.5), wherein the electrons, which have a relatively low threshold for producing Cherenkov light in air, create a large and relatively

bright cone which hits the telescopes optics and is then focused into an array of photomultiplier tubes. Some of the experiments that make use of this are VERITAS, MAGIC and HESS. The original Whipple observatory operated into the new millennium as well. These experiments will be superseded by the Cherenkov Telescope Array, or CTA, in the next decade.

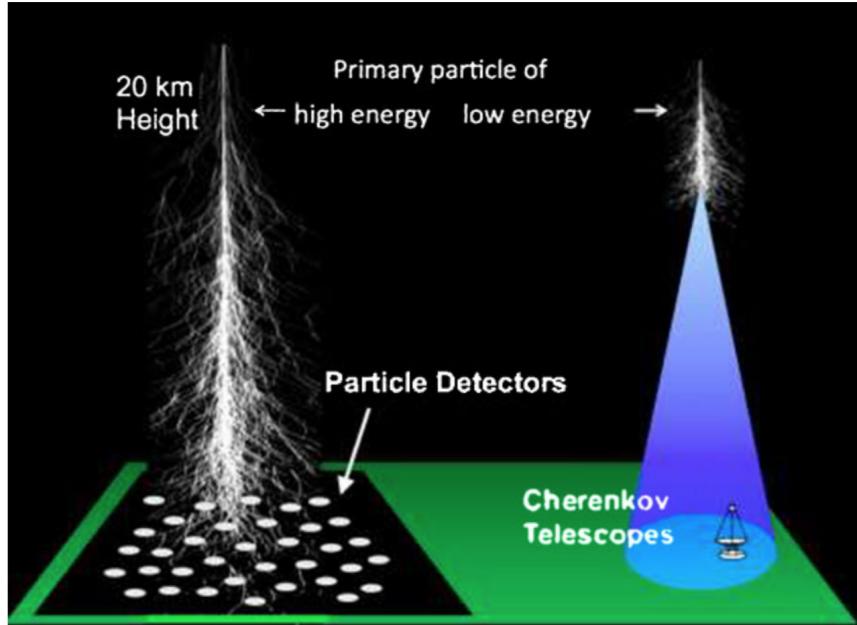


Figure 3.12: This diagram from **ultraray** shows the basics of the IACT method of gamma ray detection in comparison to the standard method of UHECR detection. To be fair to HAWC, Milagro and CASA-MIA, there are a number of surface detector arrays for high energy gamma rays.

In UHECR physics, a number of experiments have toyed with the idea of directly detecting the Cherenkov radiation from extensive air showers. The issue with this technique, and the reason it is not more widely adopted, is firstly that the Cherenkov cone from vertical showers is relatively small, while the Cherenkov light from oblique showers is usually too attenuated to be detected. Furthermore, this technique effectively involved exposing bare or lightly filtered photomultiplier tubes to the sky. This tends to be a fairly risky proposition since weatherproofing is non-trivial and requires moving parts to allow for remote opera-

tion. On top of all this, these sky facing PMTs can only be operated on clear moonless nights. Much of this discussion is informed by **schroeder**, which serves as an excellent review of all of UHECR physics, and not just radio detection.

In defense of the direct air Cherenkov technique, an array of this style would be able to faithfully distinguish the  $Z$  number of the primary to  $\approx 5\%$  as long as it has a short enough time resolution ( $< 2$  ns) [**swordyair**]. This type of experiment does exist, although the only one I am aware of is TAIGA (formerly Tunka-133). Briefly, it is an array of upward facing photomultiplier tubes that view the night sky when they can.

The form of this technique used by IceCube, which could be described as an ice Cherenkov experiment, takes strings of photomultiplier tubes with onboard electronics more than 1 km long and immerses them in the Antarctic ice. The tubes face down towards the center of the Earth (a useful quality for background rejection) since the neutrinos they are looking for are more likely to interact on their way through the Earth than through the atmosphere. In a naive sense, IceCube is an array of Tunka-133s in layers, facing downward. Since the index of refraction is higher in the ice, the particles excited and created by the neutrino interactions in the ice will produce wider Cherenkov cones and therefore it will make detection easier.

### 3.2.5 Radio

While the possibility of detecting air showers via radio antennae had existed at least by the 90's (it was considered as the technique for the Auger surface detectors [**firstcher**]), this method of detecting UHECR showers has really come to fruition in the last decade. At time of writing, a number of experiments including LOFAR, KASCADE-Grande, Auger and IceTop (IceCube's surface detector) are adding radio components to their detectors [**schroeder**]. Much of this work as it pertains to UHECRs was initially done in the Auger

Engineering Radio Array (AERA) [[radioaera](#)]. The physical basis of the technique arises

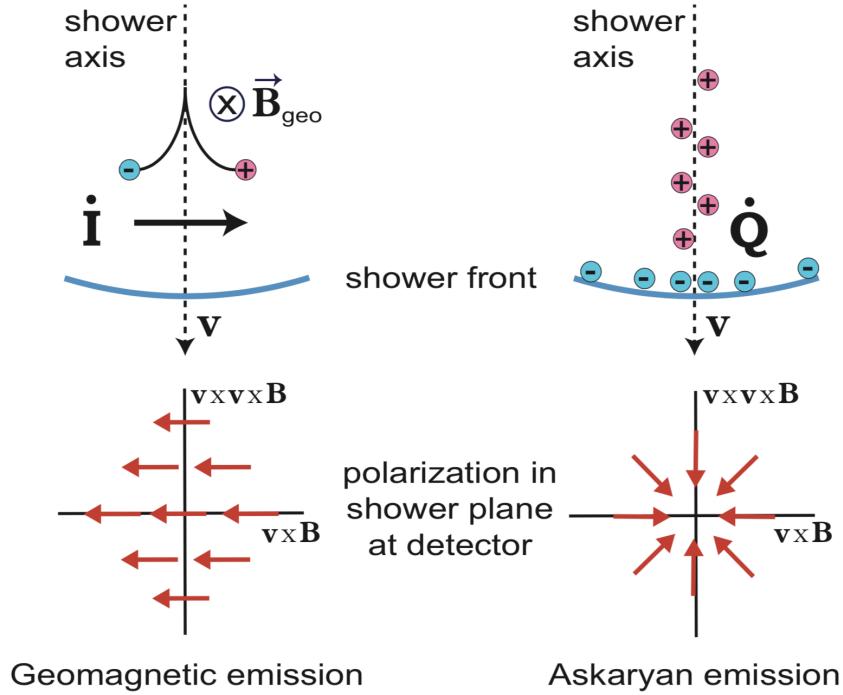


Figure 3.13: This diagram shows, in the top left, the kinematics of the geomagnetic effect and under it the polarization. On the right it shows the Askaryan effect and below it, the radial polarization it produces. Sourced from the excellent review by **schroeder**.

from two effects, the geomagnetic effect and the Askaryan effect. As the shower progresses, the large numbers of positively charged and negatively charged electrons are separated over large distances by Earth's magnetic field. This phenomena of charge separating is the same principle as the operation of a simple Hertz dipole antenna, however it has no limit to how separated it will become. This means that a wave packet of increasing wavelength per time is created. This effect is detectable in and of itself so long as the shower is perpendicular enough to the geomagnetic field and creates linearly polarized light.

Joining in concert with the geomagnetic effect, the Askaryan effect arrises from positron depletion in the shower. The essence of the Askaryan effect is positrons annihilating in the medium as the shower progresses. This leaves an excess of electrons in the shower and as

the total charge of the shower changes ( $\dot{Q}$ ), they create radially polarized waves in radio wavelengths. In matter, the Askaryan effect dominates dramatically over the geomagnetic effect, while in air the lower density makes it such that the Askaryan effect is still prominent but not as important as the geomagnetic effect [schroeder]. A diagram of both effects is shown in Figure 3.13.

Similarly to air Cherenkov, radio only emits within and near the edges of the Cherenkov cone of the shower. The technique is saved in the UHECR regime (in contrast to air Cherenkov), because of the geometrically expanding cone of highly inclined showers. In other words, for inclined showers the longer time of propagation allows the Cherenkov cone to spread, therefore making it detectable by multiple Auger stations. This, coupled with the radio-transparency of the atmosphere, has made radio detection a fairly successful technique.

### 3.3 Operating Experiments

In this section, I will review a sampling of modern experiments, each of which correspond to a technique covered in the previous section. These experiments were largely chosen as they are the most prominent in their field. Furthermore, these sections will be dedicated to the conceptual detection aspects, and the technical aspects of the experiments. Relevant results from the experiments are included throughout this document, where applicable.

#### 3.3.1 Pierre Auger Observatory

The Pierre Auger Observatory is a large UHECR observatory in Malargüe, Mendoza Province, Argentina. It consists of 1660 surface detectors and 24 Fluorescence detectors instrumenting an area of  $>3000 \text{ km}^2$  (maps presented in Figure 3.14). The fluorescence telescopes

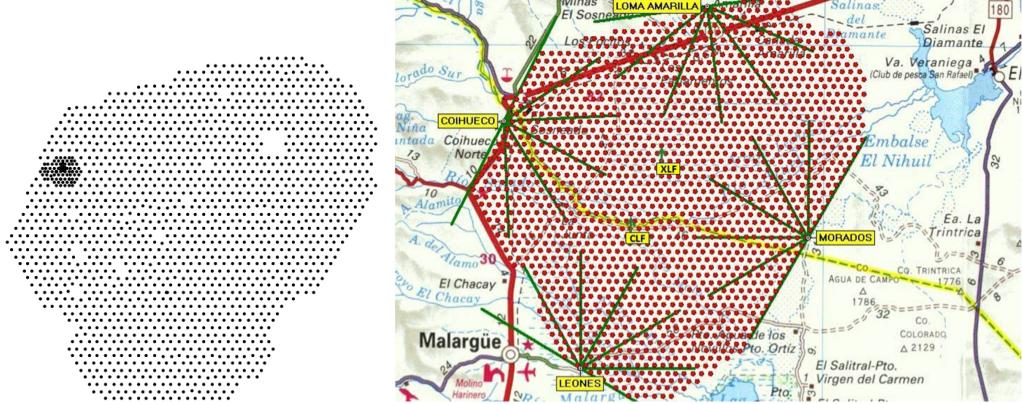


Figure 3.14: Left: A dot map of my own creation showing the positions of the surface detectors. Note that some areas are not instrumented due to terrain. This dot map makes the position of the infill and engineering arrays obvious by the bunching towards the top left. Right: The standard mapping diagram used by the collaboration, showing the FDs and SD positions as well as local geography, from [auger2015].

have a duty cycle of about 15%, while the surface detectors have >99% uptime, interrupted primarily by solar power outages from the rainy season. A rather large spacing of 1.5 km between stations in a hexagonal grid ensures maximum aperture in the desired energy range of  $10^{18.5} + \text{eV}$ . From the very large instrumented area and near 100% uptime of the array, the observatory has a total exposure of  $40,000 \text{ km}^2 \text{ sr yr}$  [auger2015]. This makes the observatory the largest and highest exposure in the world.

Design began in 1995 after the idea was conceived by Jim Cronin and Alan Watson. Prototyping ran until 2002 when construction officially began. This ran until 2008, when the full array was commissioned, however the array started taking data in 2004. While a number of designs for the surface detectors were considered, including water Cherenkov, scintillation, RPCs or radio detectors, ultimately the water Cherenkov technique was found to be the best in terms of cost per aperture per angle.

### 3.3.1.1 Auger Surface Detector

The Auger Surface Detector consists of a large tank of about 12 metric tons of water, 3.6 m in diameter and 1.2 m tall (1.2 meters is chosen as it is optimal for Cherenkov production versus attenuation), made of polyethylene with a Tyvek reflective liner. It has 3 photomultiplier tubes mounted on top, each housed in an air tight container (Figure 3.15), slightly submerged in the water thereby optically coupling them to their detection volume. The photomultiplier tubes are 9" with eight dynodes (model Photonis XP1805/D1). They have their preamplifiers and high voltage supplies built into their base. Digging in to the technical details, photomultiplier tubes are sampled at two different gains by a 40 Mhz, 10 bit flash Analog-to-Digital Converter (ADC, Analog Devices model AD9203). Their output is fed into either two Altera ACEX FPGAs, or one Altera Cyclone, depending on the date of construction. Later models have the single FPGA; there is also a block ram that is moved into the Cyclone in the newer version. Trigger logic is handled in the FPGA, and the appropriate outputs are made available from there to feed data into the onboard computer. The processing system is a IBM PowerPC 403 GCX at a clock rate of 80 MHz with 32MB of RAM, 2MB firmware storage.

For timing, the stations employ a custom ASIC with a 100Mhz counter which takes in the sawtooth granularity correction from the GPS units. The model chosen for the GPS timing receiver is the Motorola UT+ which features a 20Mhz clock capable of 25 ns RMS accuracy with the clock granularity message (sometimes called the sawtooth, after the shape it makes over time).

Communications is handled by a custom radio from the University of Leeds electrical

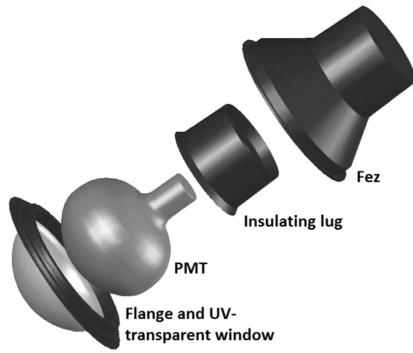


Figure 3.15: This is a diagram of the housing that the PMT is situated in to be partially immersed (i.e. optically coupled) into the water in the WCD. Figure from **auger2015**.

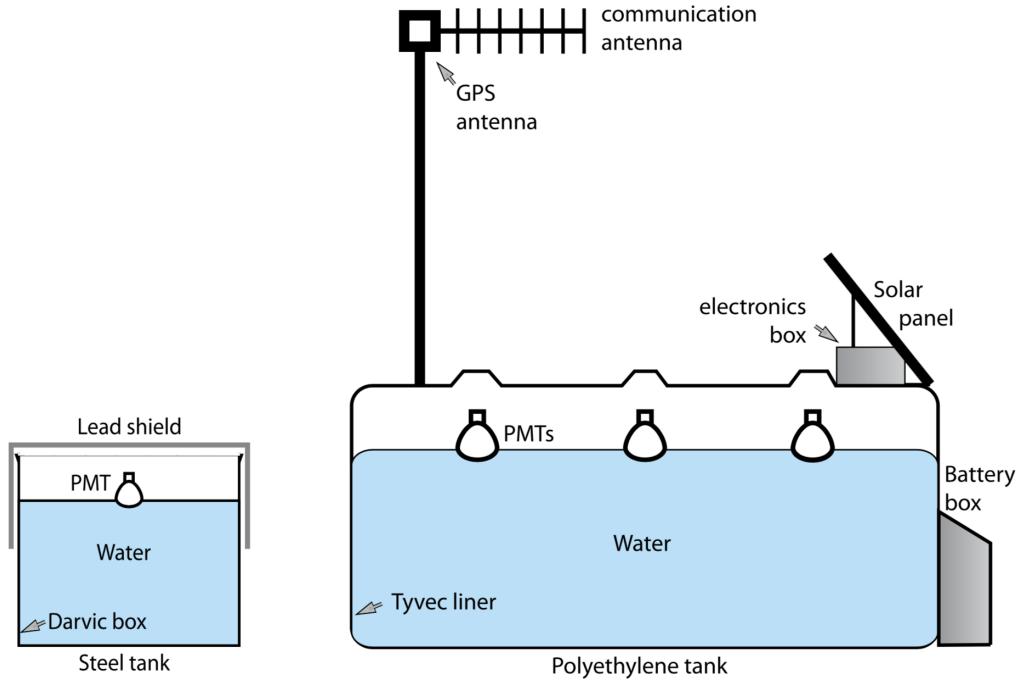


Figure 3.16: Above is a schematic diagram of the WCDs used by (left) N.A. Porter, the leader of the team that created the water Cherenkov technique, and (right) the WCD used by Auger. In some sense, the Auger WCD is the grandchild of the Porter version. From [ultraray](#).

engineering team which runs in the hundreds of bits per second range. Power is handled by a solar panel and car battery. Once the stations are set up, they are fairly robust and the main sources of maintenance are standard electronics errors, weather effects (in some situations, the SD is the tallest metal object in a large area, which is to say lightning strikes do destroy them sometimes), and the occasional issue with the local goats climbing onto the top of the tanks and knocking over solar panels and antennae.

### 3.3.1.2 Auger Fluorescence Detector

Some of the details of the FD has already been discussed in subsection 3.2.3, but I will reiterate and expand on these here. The collection of 4 sites with 6 telescopes each are usually collectively referred to as the Fluorescence Detector, although sometimes the term

FDs is used to refer to the individual telescopes. Each telescope has a  $30^\circ \times 30^\circ$  field of view, and each station with 6 telescopes then has a  $180^\circ$  field of view around it.

The telescopes are equipped with rectangular aperture mirrors, with a set of corrective lenses in front of them. Although the mirror's diameter is 3.4 m, the effective aperture is close to 1.5 m diameter due to the geometry of the housing [fd diam], shown in Figure 3.17. Each telescope is equipped with a garage-door style shutter, which is operated automatically, but monitored at night to ensure faithful operation, which is shown open for maintenance in Figure 3.17. There is also a backup cloth shutter that can be employed in case of emergency. Most weather issues are also handled automatically, and the corrective lenses and filters at the aperture work as a window to keep the cool air from the air conditioners inside. Precautions are taken to keep the facility clean as well [auger2015]. Scientifically, the FD works as a very important calibration component for the SD, as well

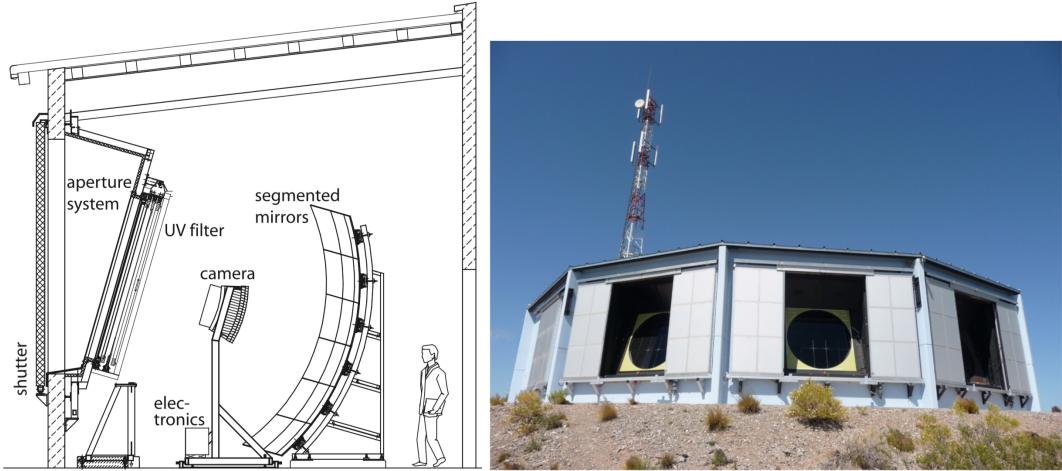


Figure 3.17: Left: A diagrammatic representation of the components of the fluorescence detector. Right: A photo of the fluorescence detectors with their shutters open for maintenance. Both from [auger2015].

as the science that it accomplishes alone. The events caught by the FD and SD simultaneously are called “golden hybrids” and are used to determine the correct SD energy estimators to use. This is due to the fact that, at time of writing, the errors and energy

calibration of fluorescence detectors are the best understood amongst all of the UHECR detection techniques [**schroeder**]. In this way, the energy is determined from the SD and from the FD and then used to recalibrate the energy estimators used for the SD.

One rather interesting aspect of the FD's construction, is that each station (i.e. each 6 telescope bundle) is controlled by a single computer called an *EyePC*. The readout electronics for each telescope are connected via firewire (a very interesing choice of connection) to what is called an MPC. These MPCs are basically single board computers which bring the firewire data in, buffer it, and then send it across a Base-100 ethernet connection to the EyePC [**auger2015**].

### 3.3.1.3 AugerPrime Upgrade

AugerPrime, an upgrade to Auger that is currently underway and the main focus of the original work in this document, will provide new electronics to each of the 1660 stations. This will include new ADCs, new trigger processing, a new central FPGA and processing system on the same chip and a number of other upgrades. These will be discussed in depth in chapter 4. In finality, the upgrade will add radio and scintillation detectors to each station in addition to the improved electronics.

## 3.3.2 TA

On the Dugway Proving Grounds in Utah, the historic site of CASA-MIA and Fly's Eye, the former AGASA group has constructed and operates a detector, the Telescope Array (TA), which is similar in nature to Auger. Schematically, it has FD and SD components which can operate in hybrid mode, like auger. There are a handful of differences and I will summarize a few key points below [**tale**].

- TA uses double-stacked scintillators, while Auger uses water Cherenkov detectors.

- Auger views the southern hemisphere's sky; TA views the northern hemisphere's sky.
- Outside of trigger processing, almost all in-situ analysis for the Auger SD is done on the processing system, while TA moves much of this to the FPGA. This gives faster event registration, but lends itself to operational opacity.
- TA employs a more rigorous data-blinding process, while Auger data are available to collaboration members on a daily basis.
- Auger does not section its surface array, meaning that all coincidences of any subset of stations will be recorded, while TA sections its array (see Figure 3.18) into three parts, making data collection easier but preventing a small number of detections on the edges of the sections.

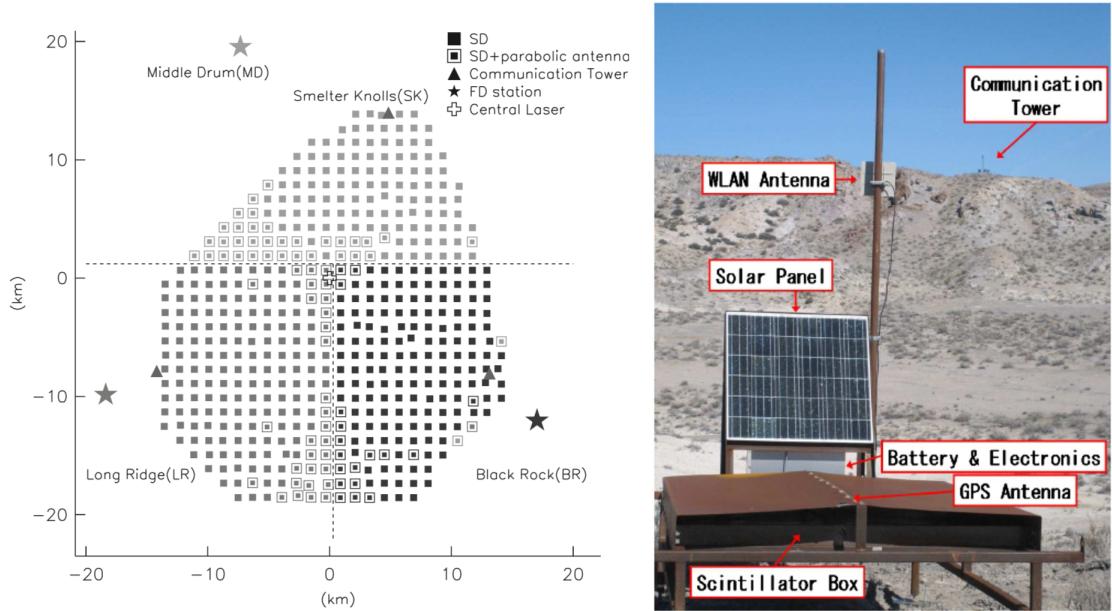


Figure 3.18: Left: A map of the telescope array. Right: The TA surface detector with components diagrammed. Both from [tasd].

The surface detector for TA uses 507 stations with a 1.2 km hexagonal spacing. As with Auger, these detectors have 100% uptime and are used to measure the spectrum with rela-

tively high statistics [**tasd**]. These detectors instrument a total area of 700 km<sup>2</sup>, of which the area covered by their FD is a subset. The electronics consist of a CPU board which is fed data by an FPGA. The CPU is a Renesas SH-4 running at 266Mhz, which is interestingly the same unit that powers the 1998 Sega Dreamcast. The programmable logic is provided by a Xilinx SPARTAN3 FPGA, and the FADC is a 50 MHz 12-bit Analog Devices AD9235.

### 3.3.2.1 Work Towards a Cross Calibration of Auger and TA

There are a number of unresolved questions in the field of UHECR physics, and a handful of these revolve around tensions between the data collected by Auger and the data collected by TA. In particular, Auger has shown that in its spectrum, as the energy increases, the composition gets heavier [**heavycomp**]. TA reports the opposite. Both collaborations use fluorescence measurements as the standard of measuring showers, although Auger is uniquely equipped to measure composition with the SD array. Going forward, the AugerPrime upgrade will allow a much more accurate determination of composition from Auger's SD array.

Additionally, TA was originally built to answer a question that haunted Auger, which is the issue of different energies as reconstructed by the FD and SD. Fluorescence techniques are considered to be powerful enough that the SD at Auger is calibrated based off measurements made with it, however when the SD reconstruction is done from first principles, it does not agree with the FD reconstruction [**tale**].

In an attempt to resolve these tensions, the High Energy Astrophysics group at CWRU along with the Auger group at the Colorado School of Mines have undertaken the task of setting up an Auger WCD in the TA facility for direct cross calibration [**sean**]. Ultimately, this involved preparing most of the equipment to run a detector of our own design in the

TA, since the TA stations do not have accessible local triggers.

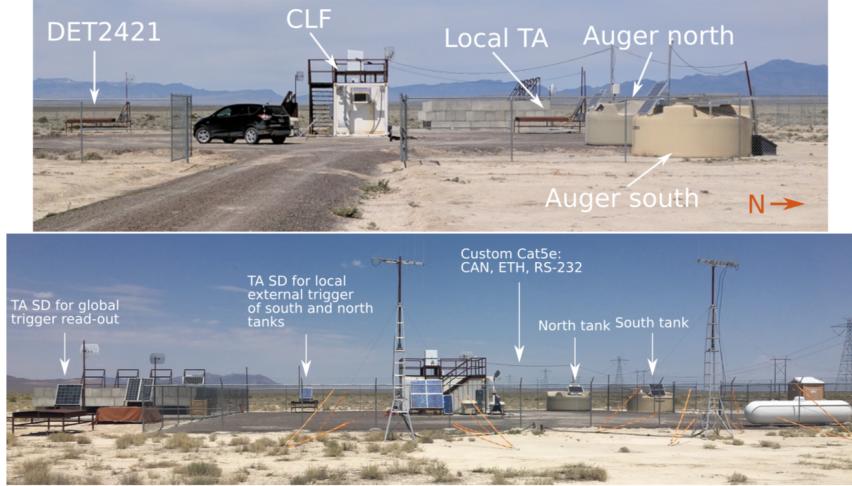


Figure 3.19: Here are two photos of the Auger@TA site with components labelled. From [sean](#).

In all, our detector, and Auger WCD of the same design deployed in Argentina, and Auger North WCD (a prototype for a cancelled project) and two TA detectors are running at the Extreme Laser Facility (panoramas shown in Figure 3.19). One of the TA stations reads out global TA triggers, while the other is effectively used for it's detection electronics and hooked up to all custom processing, including discriminators, coincidence gates and time tagging electronics. The construction of the time tagging electronics will be discussed in subsection 6.3.4.

### 3.3.3 Ice Cube

Deep under the antarctic ice, the IceCube collaboration has constructed  $1\text{km}^3$  detector detector, aiming at searching for neutrinos over a wide range of energies. Neutrino detectors must be designed around the fact that, while there are a lot of neutrinos, they have a very low probability to interact in a given volume. That probability to interact, just like charged particles and photons, is still related to the density of the medium they are traversing, but

in the case of neutrinos, one cannot guarantee an interaction even after passing through the Earth.

The construction of IceCube reflects the neutrinos' improbability of interaction by employing 3.5km long cables to sink 60 detectors, or Digital Optical Modules (DOMs) 1.5km below the surface of the ice antarctic. These DOMs face downward, looking for upward going neutrino interactions (although IceCube is still sensitive to downward going showers as well). Each string of detectors is spaced 125m from the closest other strings in a hexagonal configuration. This leads to a total instrumented area of  $1\text{km}^3$ , although they are sensitive to events that occur outside the detector as well, as long as enough of the shower makes it into their detector [**icebasic**].

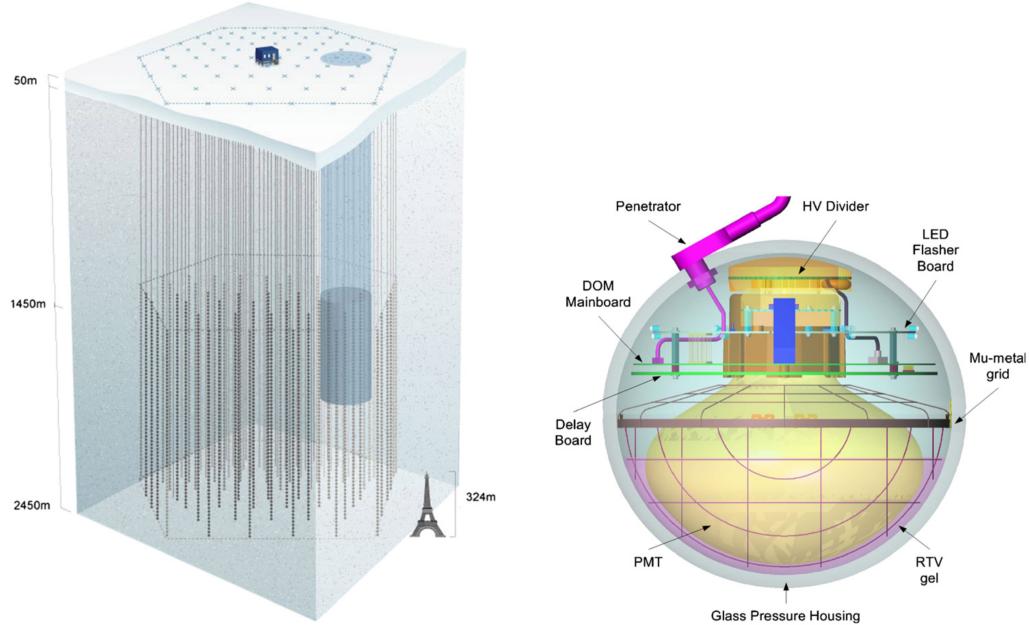


Figure 3.20: Left: A 3D rendering of the design of IceCube. This schematically shows how deep the strings go. Right: The Digital Optical Module rendered with parts diagrammed. While it does not look like an Auger Tank, it is effectively performing the same functions as one. Both taken from [**icedom**].

IceCube's predecessor, AMANDA (Antarctic Muon And Neutrino Detector Array), worked on many of the engineering issues with IceCube. The most prominent of these in

terms of detector electronics, was that the impedance of the long cables used to readout the detectors would mar the pulse width, extending it from a handful of nanoseconds to milliseconds. Reversing this pulse shaping could get you an accurate arrival time, but not individual photoelectron pulses. Using optical signals was possible, but the optical transmission technology at the time did not allow for a high enough dynamic range in analog pulses [**icebasic**]. Ultimately, both methods pointed to the need for a detector of this type to digitize its pulses in-situ.

Interestingly, the DOM technology for IceCube is parallel in development and conceptually close in operation to an Auger WCD. Once a DOM is frozen into the antarctic ice, it cannot practically be retrieved, and so it must have a very low failure rate (90% survivability over 15 years is the target, 94% is the current measured rate) [**icedom**]. The DOM must perform all of its calibrations in-situ less timing calibration.

The timing system for IceCube consists of a number of DOMHubs, which use custom timing boards to distribute GPS disciplined timing signals throughout the array. Each DOM's timing is calibrated in a call and response fashion through what they refer to as RAPCal, or Reciprocal Active Pulsing calibration. This precise timing calibration helps DOMs with local triggers, which are passed up and down the cables to neighboring detectors to determine if a 1/4 photoelectron trigger is noise or an event. The DOM's electronics include a type of flash ADC and two digitizers. The two digitizers, called Analog Transient Waveform Digitizers, or ATWDs are used, one at a time, to digitize traces from events that trigger the DOMs threshold and are read out by their central data center. This pairing allows a significant reduction of dead time [**icebasic**, **icedom**].

Neutrinos fluxes in general provide some of the best constraints on sources for UHECRs, since high energy hadronic interactions at the source of UHECRs should cause charged pions to emit relatively high energy neutrinos [**foteini**, **pulsarno**]. It is frequently suggested

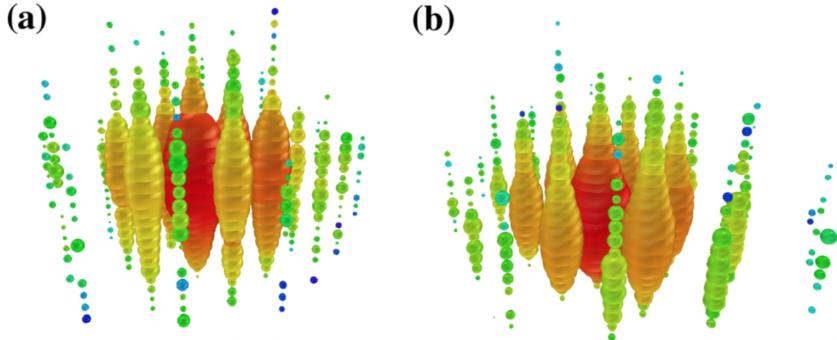


Figure 3.21: Pictured here are the two IceCube PeV neutrinos, where (a) is the earlier August 2011 event, and (b) is the January 2012 event. Plots from [\[pevneuts\]](#).

that without a measurement of high energy astrophysical neutrinos, it is difficult to prove any acceleration mechanism. As of yet, IceCube has detected two PeV neutrinos (Figure 3.21) but nothing of higher energy. In general, models of cosmic accelerators desire  $10^{17}$  eV+ neutrinos.

### 3.3.4 CTA

For the past 30 years, the Air Cherenkov has been the primary method for Very High Energy (VHE) gamma ray astronomy (with significant contributions from others such as HAWC and CASA-MIA). As many US and European groups began thinking about the next generation of Imaging Atmospheric Cherenkov Telescopes (IACTs), it was realized that gamma ray physicists and astronomers from around the globe would have to pool resources to build a bigger array of telescopes. Hence, the Cherenkov Telescope Array (CTA) was conceived.

Passing on the legacy of HESS, VERITAS and MAGIC, CTA will consist of two main sites, one in the Canary Islands, at La Palma on the Roque de Los Muchachos site, and another in Chile at the European Southern Observatory's Paranal site [\[ctaong\]](#). The sites have been chosen to maximize the performance per unit time, and minimize the cost of

land and logistics [**tarek**]. The array will consist of 3 or 4 telescope designs, with large and medium telescopes at the north site, and all three sizes of telescopes at the southern site. In air Cherenkov detectors, larger telescopes focus more light and thereby make lower energy events detectable, while multiple smaller telescopes allow for a wider field of view with less light collecting area, making them effective for high energy events.

Of the current designs for CTA telescopes, there is one model in the Large Sized Telescope (LST) category, two in the Medium Sized Telescope category (MST) and 3 in the Small Sized Telescope (SST). Ultimately, the collaboration will likely decide on 1 SST, while employing two MSTs of one type per site. The LST, one MST and one SST use the tried and proven Davies-Cotton optics design, while the rest use forms of two mirror optics. In general, two mirror optics are more delicate, but allow for a wider field of view, and therefore more showers from gamma rays detected. A “family photo” of the potential designs is shown in Figure 3.22.



Figure 3.22: In this figure, from **ctaong**, we can see all six of the candidate telescope designs. While a review of each design is outside the scope of this work, I would encourage those interested to see the paper this is from, or to simply search any of the above telescope names.

In CTA, the workhorse telescopes will be the Medium Sized Telescopes (MST). This is also the only design range in which deploying two different types of telescopes, with different optics and cameras, is being considered. To comment on this briefly, using two instruments can be more difficult than having a unified approach due to differing development, maintenance and analysis issues, but having them developed relatively independently can give important cross checks. Furthermore, from a political perspective, a country will understandably be more likely to fund the construction of an apparatus if scientists from the country played an important role in its development. From this perspective, accepting two MST designs can provide more infrastructure funding for both groups and sites.

To briefly take a broader view, in many of the Cosmic Accelerator candidates that I've discussed in section 2.2, large leptonic populations are expected to be accelerated in addition to UHECRs. These leptons generally interact in the source and should produce VHE to UHE gamma-rays. With powerful enough optics, and a long enough exposure CTA and other gamma ray observatories should be able to find events at the top of their energy spectra which perhaps point back to UHECR sources. More data collection and higher statistics should aid in any future multi-messenger analysis involving VHE gamma-rays, UHE neutrinos and UHECRs.

### **3.3.4.1 Schwarzschild-Coudè Telescope**

The northern MST is a design originally spawned by Vladimir Vassiliev of University of California, Los Angeles. This design was under consideration well before it became a candidate in CTA. Originally, this work was to culminate in the Advanced Gamma-ray Imaging System, but the team saw a more feasible funding opportunity by joining CTA [**agis**]. With a mostly US and partly Italian team, the prototype of this telescope has been set up at the Fred Lawrence Whipple Observatory (which is, perhaps confusingly, not the

site of the original Whipple Telescope). On Januray 31, 2019, this telescope saw first light.

The optics and camera design make the Schwarzchild-Coud  r Telescope an extremely advanced and unique design. Outside of my own interest as a collaborator on this project, I must say that this telescope is approximately an order of magnitude/generation more advanced in any given dimension than even the other CTA telescopes. More pixels will be active in the focal plane of this camera than every IACT currently in use (at time of writing) in the world, combined.

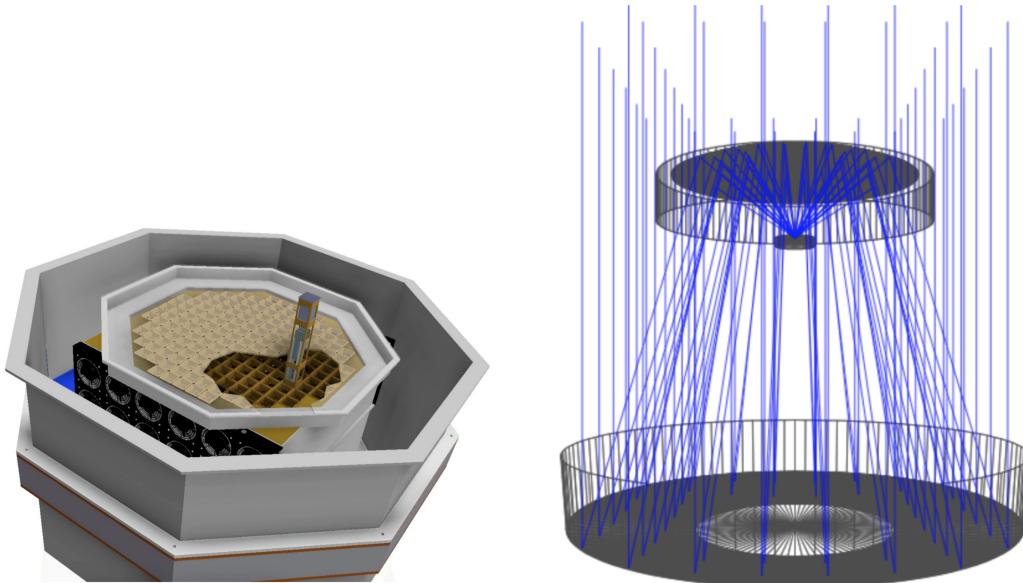


Figure 3.23: Left: The SCT camera design rendered, from **vladscope**. Note the detector modules being inserted into the lattice and the muffin fans mounted below for cooling. Right: A diagram of the camera's optics with a ray tracing simulation overlaid [psctoptics].

The two mirror Schwarzchild-Coud  r have been known about since the turn of the 20th century, but mirror finishing technology has only recently reached the needed precision-at-cost to make this telescope a viable option. The optics, shown in Figure 3.23, give this telescope an  $8^\circ$  field of view in each dimension, allowing a larger collecting area over the night sky than any previous or planned IACT [psctoptics]. The optics also focus the image onto a focal plane that is narrower than previous designs, ruling out the use of PMTs in the

camera.

The camera, due to optical constraints, must then use Silicon Photomultipliers (SiPMs), which are small silicon-based avalanche photodiodes, that effectively rely on the breakdown of their charge binding structure when put under a relatively high voltage (high for nature, low for PMTs). These allow us to fit 11.2 kPixels in the focal plane. At this resolution, it should be (and from the first couple of events, is) possible to accurately determine the morphology of the events, and therefore very efficiently reject cosmic ray events. Currently, the telescope is operating and needs a time-tagging system to allow for cross calibrations and meaningful event determination. This project is underway and will be discussed in subsection 6.3.5.

# **Chapter 4**

## **Electronics and Firmware for the Auger Prime Upgrade**

Operating a modern air shower detector like Auger requires a large amount of computing power, both centrally for data collection, packaging and reconstruction, as well as in situ in the case of a large array. We will start by discussing the scientific motivations for the AugerPrime upgrade, which in turn dictate the station electronics requirements. Then, a description of the science and mathematics behind decoupling the electron and muon signals in the SD is given. Subsequently, we will have an overview of the Upgraded Unified Board (UUB), and describe its main features. The organization of work packages for the upgrade will be discussed briefly, and any features not covered in the board overview will be discussed as individual work packages, less GPS and Timing which has its own section towards the end of the chapter. After the work packages, I will discuss the operating system of the UUB and how it was brought-up. The chapter continues into a discussion of the GPS software and hardware integration. Finally, we end on a walk-through of the boards operating system and bring up procedure, focusing on the author’s documentation on the topic.

## 4.1 Scientific Motivations for AugerPrime

Since the days of John Linsley, much of UHECR science has been motivated by measurements of spectra, and while attempts to determine the composition of cosmic ray primaries have been successful at lower energies (especially in satellite- and balloon-borne experiments, e.g. **tiger**) they have not been the primary focus of the UHECR field. However, at this point the spectrum is resolving (Figure 2.2) and it seems that a serious leap in our ability to understand sources is presented in shower-by-shower composition measurements. Auger’s Fluorescence detector already measures  $X_{max}$ , and therefore composition, fairly well, but by separating the muonic and hadronic component of showers, we have an opportunity to employ the entire surface detector array with its 100% uptime in composition based measurements. This is the basis of the AugerPrime program (Prime, although stylized lower case, is an acronym for Primary Ray Identification through Muons and Electrons).

In order to deconvolve the muonic and hadronic component of the shower, new hardware for the surface detector is required. Towards this end, each station in the array will be equipped with a Scintillation Surface Detector (SSD) and a radio antenna (preliminary details in [**dutchaera**]). Each of the instruments on an upgraded SD station has a different response to the EM and muonic components of the shower, and we can exploit this to separate the muon and electron components of the shower (see subsection 4.4.1 and [**primeuniv**, **univpart**]). Finally, since a shower’s muon content depends on the number of initial hadrons and therefore the atomic number of the primary, we can use the muon to electron ratio,  $N_\mu/N_e$ , to find the composition. In practice, this actually is accomplished by finding the depth of shower maximum via shower universality (subsubsection 3.1.4.2[**univpart**]). Ultimately, this is correlated with  $Z$  via simulation data such as

those shown and discussed in subsection 2.2.4, specifically Figure 2.5.

As a consequence of the ability to better determine the muon content of showers, we will be able to verify models of particles physics based on extrapolations from LHC data. In particular, the ultra high energy models that CORSIKA (subsection 3.1.5, [corsika]) employs predict more muons than are observed in the SD [mudef]. It is unclear if this is an issue with the muon production in the models or if it is a problem with the muon content measurement of the SD. An upgrade of the surface detectors is required to answer these questions, as well as make detailed shower-by-shower composition measurements. This upgrade takes the form of AugerPrime.

## 4.2 Decoupling the Shower Components: SD Composition Measurements

As mentioned above, the primary goal of the AugerPrime upgrade is to determine the composition of primaries with the Surface Detector. The effective workflow here is to find the muon content of the shower, which we can relate to  $X_{max}$ , which can then be used to give an estimate of the primary composition [primeuniv, toprime, primepdr, xmaxcomp]. Stepping into a low energy particle physics point of view, we can consider the question of how we unravel the components of a shower using the efficiencies of two different detector types. Intuitively, we should reach for a matrix inversion problem, where we somehow invoke the detection efficiencies of each detector type and use these to solve for each component of the signal, starting with the mixed signals. This is a classic linear algebra problem, and the question is how to effectively state the problem in a way that will be useful to us, which is taken care of by matinv.

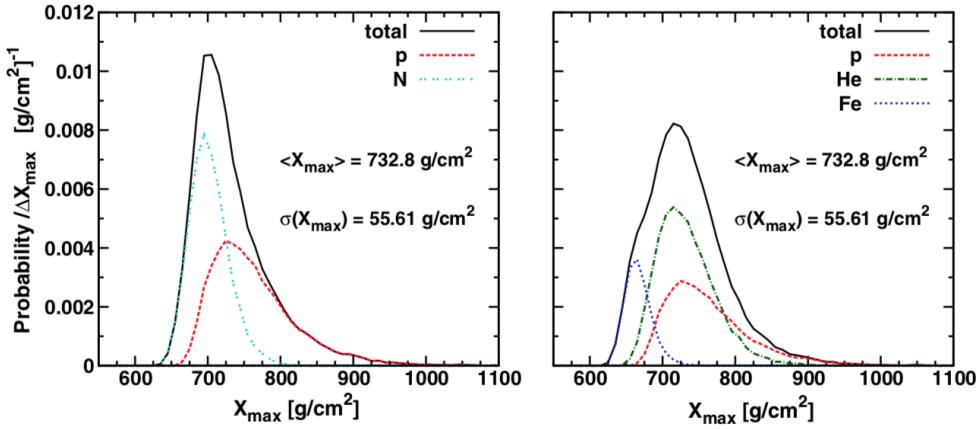


Figure 4.1: These plots out of `xmaxcomp` give some visually simple intuition for the width and mean of the  $X_{\max}$  distributions for different primaries. It should be clear from these that a concrete statement of primary composition is out of reach using this technique, but we can assign probabilities of an event's composition.

This publication ([**matinv**]) was originally meant as a proposal for the AugerPrime new detector technology, in which the tank would be split into two parts with an extra rubber liner. The liner would have significant stopping power for the EM component of the shower, thereby effectively creating two different detectors in the same tank. This was rejected because, somewhat unintuitively, it costs much more to overhaul and refill the water in 1600 stations than it does to simply mount a scintillator on top. The electronics costs between the two plans were comparable.

Ultimately, the work towards a separated tank structure produced the matrix inversion method previously mentioned, but for a different context. The framework for using this matrix inversion technique on scintillators and water Cherenkov detectors is documented in `primepdr` and the mathematics follows:

$$\begin{pmatrix} S_{SSD} \\ S_{WCD} \end{pmatrix} = \begin{pmatrix} \lambda A_{SSD} & A_{SSD} \\ \beta A_{WCD} & A_{WCD} \end{pmatrix} \begin{pmatrix} f_{EM} \\ f_\mu \end{pmatrix}, \quad (4.1)$$

where  $S_i$  is each detector's respective signal,  $A_i$  is each detector's respective area,  $\lambda$  is the MIP/VEM/ $\cos \theta$  ratio,  $\beta = \cos \theta + 2h/(\pi R) \sin \theta$  with  $h$  and  $R$  the height and radius of the WCD, respectively, and  $f_i$  the flux of the respective shower component. Solving this linear algebra problem to give the fluxes is then trivial. One thing to note about this method is that we still have to put it through the statistical filter of  $X_{max}$  to composition correlations. Another is that it does not depend on better timing. Ultimately, this method depends on shower universality in order to be valid, and specifically it relies on the separability of four components of the shower [**primepdr**, **univpart**]:

$$S_{tot} = S_{em}(r, DX, E) + N_\mu^{rel} (S_\mu^{ref}(r, DX, E) + S_{em}^\mu) + (N_\mu^{rel})^\alpha S_{em}^{low-energy}(r, DX, E) \quad (4.2)$$

From left to right the  $S_i$  here are: the electromagnetic component, the pure muonic component, the EM component created by the muonic component and the electromagnetic component from neutral pions. The explicit dependence on  $X_{max}$  here is a consequence of assuming shower universality, although the details behind this are outside the scope of this work. To drive the point home, however, it is with these techniques (universality and matrix inversion) that we can decouple this signal from muons and the signal from the EM component of a shower to give an estimation of  $X_{max}$ .

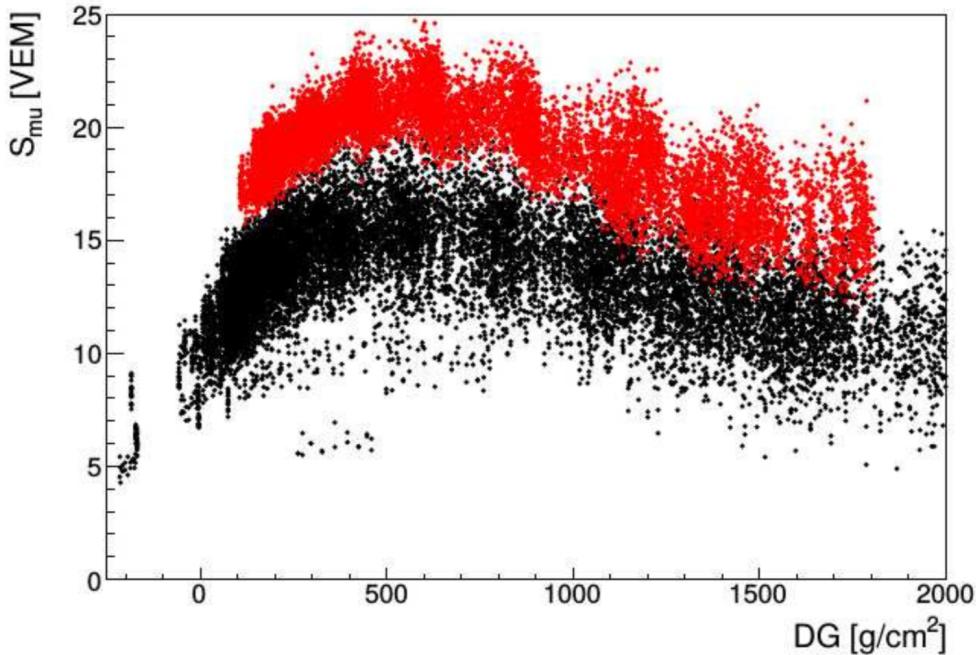


Figure 4.2: This plot, originally from [\[primeuniv\]](#), shows the number of muons as a function of depth and primary composition.

After discussing this fairly solid and relatively straightforward technique, it is necessary to mention that there *should* be a better way to relate the number of muons in the shower directly to the primary composition, without relying on the statistical variability of  $X_{max}$ . Hints of this are shown in Figure 4.2, where we can see that the number of muons is only lightly dependent of  $X_{max}$ . A number of techniques have been looked at and successfully simulated to find the number of muons in a shower, and the better timing does help with this ([\[mupart, primeuniv\]](#)) but a stable and accurate technique to take the observables and turn them into a measure of the primary composition has not been established. This would likely have been accomplished already if funding for AugerPrime had continued in the US.

## 4.3 Board Overview

The Upgraded Unified Board is effectively a drop-in replacement for the Unified Board, fitting in the same electronics enclosure and connecting to the same radio, GPS antenna, power system and photomultiplier tubes. The board is powered by a Zynq™ 7Z020 SoC/F-PGA (System-on-a-Chip/Fully Programmable Gate Array), and consists effectively of three layout sections: the Zynq™ and interfaces, the front end inputs and digitizing, and slow control. In this section we will discuss the context and general features of the UUB, paving the way for a detailed investigation of its subsystems including the GPS Integration and Board Bring-up detailed in section 4.6 and section 4.7, respectively.

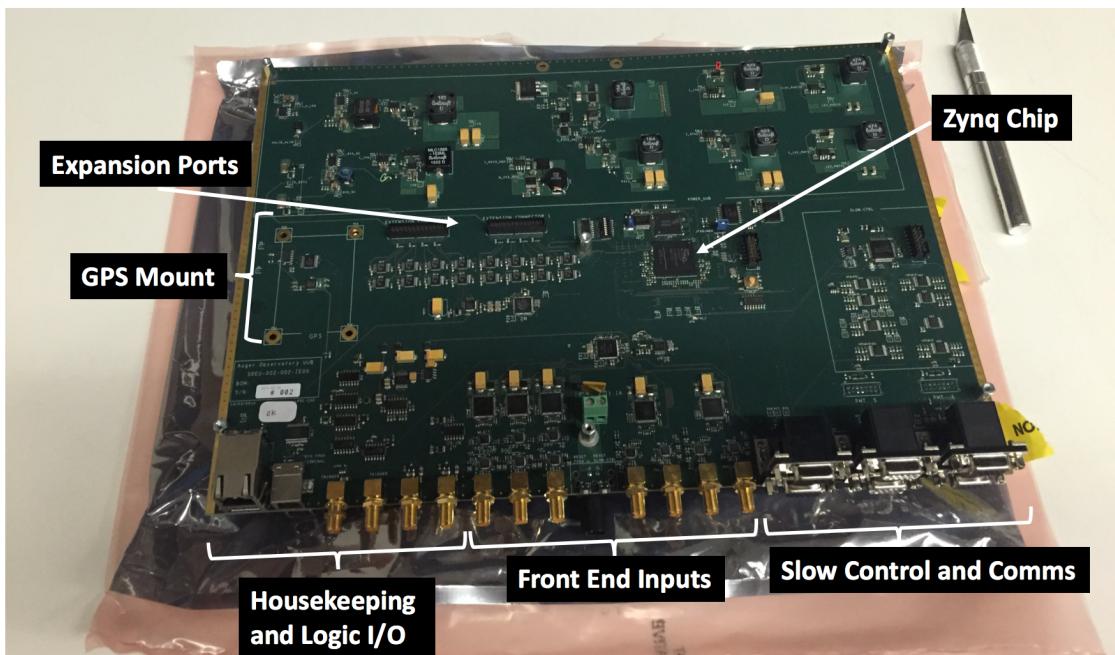


Figure 4.3: A Picture of the UUB with some of the basic parts and connections labeled.

The Zynq™ chip provides most of the functionality of the UUB, enclosing event encoding including trace recording, time-tagging, trigger logic, communications packet creation, housekeeping, monitoring and more. This chip is split into two essential parts the Processing System (PS) and the Programmable Logic (PL). In the diagram in Figure 4.4, the

PL is not explicitly shown because of its undetermined nature, but what is made clear is the architecture of the PS. One potentially confusing part about this system is that there are both “hard” and “soft” peripherals, such as GPIOs (General Purpose Input Output) and UARTs (Universal Asynchronous Receiver/Transmitter). These devices live both in hard silicon on the processing side, where parameters of them are effectively programmable but the device itself exists whether or not you use it, and in the programmable logic, where they are completely customizable.

The CPU in the processing system consists of two ARM Cortex<sup>TM</sup> A9 cores running at 333Mhz. This effectively allows an appropriate operating system to invoke parallel processing to decrease dead time and increase reliability of the detector. An example where this is especially useful is interpreting GPS messages through the receivers rather slow baud rate. At a 9600 bits per second rate of communication, the processing system must wait for 150 or more bytes to come in, corresponding to more than a microsecond of dead time. In this situation, the second core will handle all other functions, in particular critical data handling. That said, the PL firmware is designed to save all important information, i.e. anything that is science-critical, until it is ready to be read out. Which is to say, unlike in some applications, we are not in any dire hurry to read things out. Furthermore, interrupts have been implemented in the processing system for both shower and muon triggers, and Direct Memory Addressing (DMA) is used to transfer the data from the trigger buffers to the PS with minimum processing power and maximum speed.

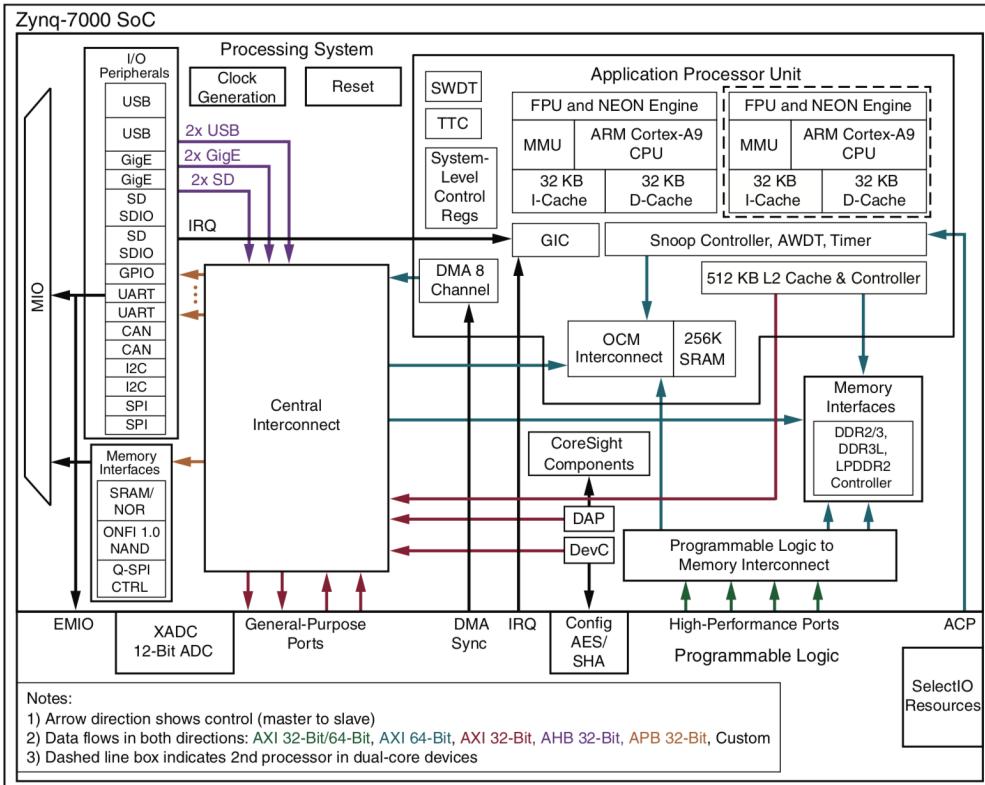


Figure 4.4: A schematic of the internals of the PS in the Zynq™ chip, from [zynq](#).

The external features of the physical board layout are shown in Figure 4.3, while the high level logical and signaling layout is shown in Figure 4.5. A quick examination of Figure 4.5, shows that the Zynq™ chip is truly the hub of all of the activity on the board, in fact even external clocks are distributed through it. The main external external clock is the Abraccon ABLJO-V-120.000MHZ-T2, a VCXO (Voltage Controlled Crystal Oscillator) whose control voltage is set by an SPI bussed DAC. This VCXO was chosen for its low jitter of .1ps, however a number of options have been discussed for future versions of the UUB. These include temperature compensated and oven compensated oscillators to deal with the relatively rapid temperature changes (see **brandt** for more about the desert temperatures). The speed of the oscillator was chosen to match the speed of the AD9628

120MS/s ADCs.

It is worth noting that this is a fundamentally different timing design scheme than the original Unified Board. In the UB, the time tagging board is external to the entire system (i.e. goes in an expansion header) and runs at 100Mhz, while the ADCs run at 40Mhz; in the UUB, time tagging and trace recording are synchronized to the same clock, both out of convenience and to reduce redundant components in the board for power concerns. In early versions of the UUB, the 120Mhz clock was to be filtered through a jitter reduction chip. Ultimately, it was decided by the timing team (the author included), initially led by Patrick Allison of OSU, that the small fractions-of-a-picosecond reduction in the frame jitter of the ADCs would not affect any science goals (note, this would affect the recorded shape and smoothness of the shower front, but not at a significant level). The chip was rather power hungry, and its removal left us with a slightly easier power budget.

Signal processing-wise, the analog portion of the front end is largely ported over from the Unified Board (UB, previous control board for Auger), and is effectively designed to ensure pulse integrity from the photomultiplier tubes. Additionally, it includes amplifier electronics for each channel and an anti-aliasing filter, to ensure faithful reconstruction of relatively high frequency features. Finally, the board also has a number of power delivery features. It serves as a hub for the various voltages that need to go to its detectors and other peripherals. As a note, it is worth mentioning that the power delivery has caused some amount of engineering headaches as it can induce voltages in other lines that are routed above, below or even just nearby.

## 4.4 Subsystems and Work Packages

In order to divide the large amount of work that the collaboration did towards upgrading the surface detector, the effort was split into work packages. A table is provided in, with a table of participating institutions and their tasks in Table 4.1. In the subsections under this header, we will discuss a few of the work packages that relate to the tasks completed for this dissertation. If you are looking for details on those systems not explicitly discussed here, check section 4.3, if it is not there, then we have deemed it outside the scope of this work, for example board assembly, or array simulation. When the work packages

	WP1	WP2	WP3	WP4	WP5	WP6	WP7	WP8	WP9	WP10
BUW, Wuppertal - Germany				Resp.	Part.	Part.				
CNEA, Bariloche - Argentina					Part.	Part.		Part.		
ITEDA, Buenos Aires - Argentina					Part.			Part.		
CWRU, Cleveland - USA				Resp.	Part.	Part.				
FNAL, Chicago - USA					Part.			Part.	Part.	Part.
FZU, Prague - Czech Republic					Part.					
INFN, Lecce - Italy	Resp.	Part.			Part.	Part.				
INFN, Torino - Italy	Part.				Part.	Part.	Resp.			
IPNO, Orsay - France	Part.				Part.			Part.		Resp.
KIT, Karlsruhe - Germany					Part.		Part.		Resp.	
LOD, Lodz - Poland		Part.			Part.					
LPSC, Grenoble - France				Part.	Resp.	Part.		Part.		Part.
MTU, Houghton - USA		Resp.	Part.		Part.	Part.	Part.		Part.	
OSU, Columbus - USA	Part.	Part.			Part.	Resp.		Part.		Part.
PAO, Malargue - Argentina					Part.			Resp.	Part.	
SU, Siegen University					Part.			Part.		

Table 4.1: A diagram of work packages and the institutions involved in them. This is from the publicly available development plan, **devplan**. It is worth noting that some institutions such as Fermilab (FNAL) and Ohio State (OSU) made important contributions early on, but were ultimately forced to pull out of the project due to funding restrictions.

were initially designed, the new detector type had not yet been chosen and a number of contenders were still moving forward with their research. Eventually, the decision was made to use the scintillation detector over the various other proposals, which included a small PMT to pick up saturating showers ([**dynrang**]), splitting the tank in two for muon electron separation, and installing Resistive Plate Chambers under each station, amongst

WP1	Analog PMT signal processing development
WP2	Trigger development
WP3	Time tagging development
WP4	Slow Control development
WP5	UUB Hardware Design & Integration
WP6	UUB software development
WP7	Calibration and Control tools development
WP8	Assembly, Deployment and Validation
WP9	Simulations and Science Validation
WP10	Project Management

Table 4.2: A table of the different work package designations. Some of these will be discussed in the coming sections.

others. When the decision was made, SSD calibration and construction was added to WP1.

Minutes and official documentation are publicly available here.

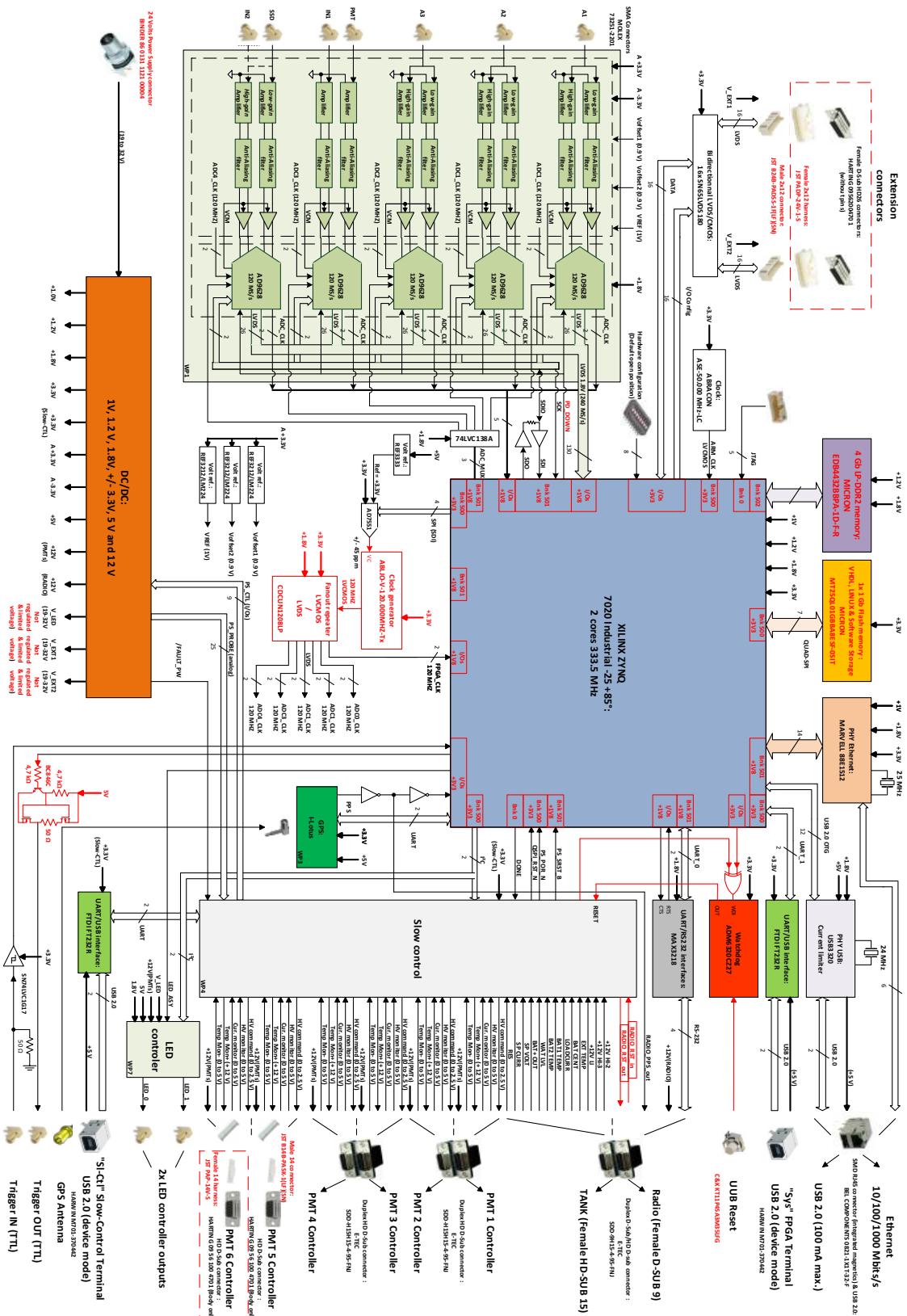


Figure 4.5: A block schematic of the UUB, displaying the major components and their interconnections, from **tiina**.

#### 4.4.1 Scintillation Surface Detector

While some details of the SSD have already been discussed in subsection 3.2.2, we will review them here and provide a few more technical diagrams and pictures. The Scintillation Surface Detector is an organic extruded plastic fluorescent scintillator of about  $3.8\text{m}^2$  manufactured by Fermilab. As mentioned above, the addition of this detector should give us a handle on the muon content of the shower, in conjunction with the water Cherenkov portion of the detector.

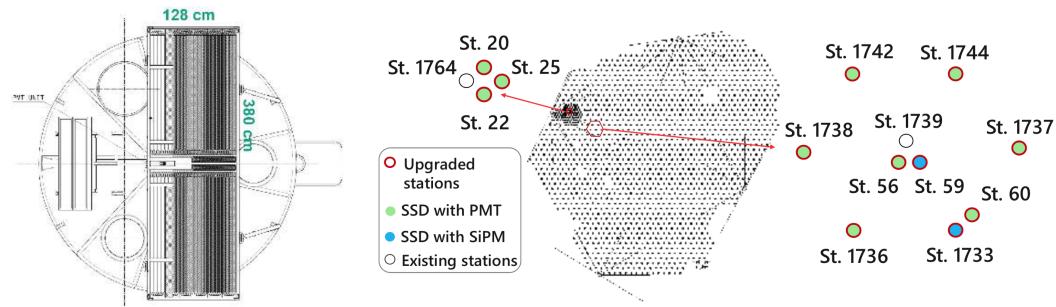


Figure 4.6: Left: A schematic of the SSD mount originally shown in **augerscint**. Right: The positions of the SSD carrying stations from **firstdat**.

In the final stage of AugerPrime, each SD station will be fitted with a scintillator which sits on top of the station in a specially designed mount, pictured in Figure 4.6. As pictured in the left panel of Figure 3.7, Wavelength Shifting Fibers (WSFs) pickup and transport the scintillation light to the photomultipliers. To determine the best choice for photomultiplier tubes, both silicon photomultipliers and photomultiplier tubes (Hamamatsu R9420) were evaluated, with the PMTs being the final choice ([**augerscint**, **firstdat**]).

Each scintillator bar, of which the larger panel is comprised (48 in total), is coated with titanium oxide on the top for handling and for reflection of usable signals back into the bulk of the scintillator. On top of the panels is a corrugated steel heat pipe structure which allows air to flow, and reflects the sunlight, thereby keeping the SSD at ambient temperature. The

fibers are bundled together at the center of the SSD and aggregated at the ‘cookie’ in front of the PMT. The first SSDs were commissioned to the Engineering Array on Sept. 16, 2016.

#### 4.4.2 Trigger Logic

In large part, the trigger logic for AugerPrime was designed by Dave Nitz of Michigan Technological University. It is implemented in the Xilinx Zynq™ chip, on its Artix-7 FPGA, and is fed by the analog front end through the digitizers (discussed and named in section 4.3). In the early days of Auger, a number of triggers were tested, but the absolute most successful was the Time-Over-Threshold (TOT) trigger. Single bin and MoPs (Multiplicity of Positive Steps) triggers are also available. Some of the triggers also have compatibility modes, in which they are effectively downsampled to 40Mhz and the frequency response of the UB ADCs is modeled and applied to them. Since much of the work done for timing in chapter 5 and chapter 6 involves triggering or using the signal chain from the triggers, we will review the nature and functions of the triggers available on the UUB.

Below, we will divide up the triggers into the Compatibility, Current and Calibration categories. The compatibility triggers allow old analysis routines to be used on the data. The current triggers are the state of the art triggers that have been developed for the UUB. These use more advanced techniques than the compatibility triggers, and take advantage of the digital signal processing capabilities of the Zynq™. Finally, the calibration trigger looks for single through going muons which can then be used in the charge histogram to set the VEM level (see subsection 4.4.6). These descriptions and more technical information about the UUB’s PL can be found in **triggers**.

##### Compatibility Triggers:

1. **Compatibility Single Bin Trigger:** in parity with the triggers on the UB, this trigger operates on the downsampled ADC traces, and simply asks “have we seen a bin greater than the set threshold?” It can be programmed with different thresholds for each PMT, and can be programmed for different coincidence multiplicity requirements via a control register. From a science perspective, this is to detect the strong signal from single through going muons and is especially effective at detecting highly inclined showers where the atmosphere has depleted the EM component of the shower.
2. **Compatibility Time-Over-Threshold Trigger:** originally intended to decrease the data rate of stations by ensuring they only trigger on real showers, the TOT trigger applies a threshold to each ADC bin in a running 120 bin window. When a particular multiplicity, set in the control register, of consecutive bins are above threshold, it trips the TOT trigger, which is then elevated to a T2 event (i.e. sent to the Central Data Acquisition System, CDAS).

### Current Triggers:

1. **Time-Over-Threshold, Deconvolved:** the TOTd trigger is the advanced form of the TOT trigger, employing an exponential deconvolution, which effectively amplifies the signal towards the end of a shower trace. By doing so, we can get a more accurate count of the number of bins which correspond to usable shower data, and therefore the multiplicity that we specify in the control register will more accurately reflect the length of the shower. Besides the deconvolution, it works in effectively the same way as the shower.
2. **Multiplicity-of-Positive-Steps:** the MoPS trigger looks for the signal in a trace to rise after each bin. Each time it rises, that number of steps is added together to form

the multiplicity, and if the trace multiplicity hits the multiplicity number set in the control register, it launches a trigger.

3. **Full Bandwidth Single Bin Trigger:** This is truly the same as the old compatibility single bin trigger, but at the full 120Mhz clock rate.

**Muon Trigger:** this trigger allows the stations to trigger on uncorrelated fluxes of through going muons for calibration. In particular, it has programmable delays for comparing the signal of the SSD and WCD for MIP vs. VEM curves.

#### 4.4.2.1 Trigger Hierarchy

To efficiently balance the data constraints of an array as large as Auger, a hierarchy of triggers must be established [**ubtriggers**]. At time of writing, the new triggers of the UUB have not been implemented in this hierarchy. That said, we can elucidate how the old triggers contribute. As mentioned above, the two previously implemented triggers are the TOT and the Single Bin triggers. The Single Bin (SB) trigger is meant for the short but high pulses from single through going muons, while the TOT trigger is meant for showers. The TOT trigger represents a more significant signal that will always be associated with some type of shower, whereas the SB trigger can come from showers or uncorrelated fluxes, and is therefore less trustworthy.

There are three levels of triggers in the SD array: T1, T2 and T3. T1 and T2 live in the station, where T1 can be either noise or a shower, but T2 is meant to represent events that are shower candidates. T1 rates are used for calibration, and do not necessarily require a coincidence amongst all three photomultiplier tubes, while T2s represent either passing a TOT trigger or passing a very high Single Bin trigger. T2s are sent to the Central Data Acquisition System (CDAS) over radio and CDAS makes a decision to elevate to T3 based on the adjacency of stations it is receiving T2s from and the time window in which

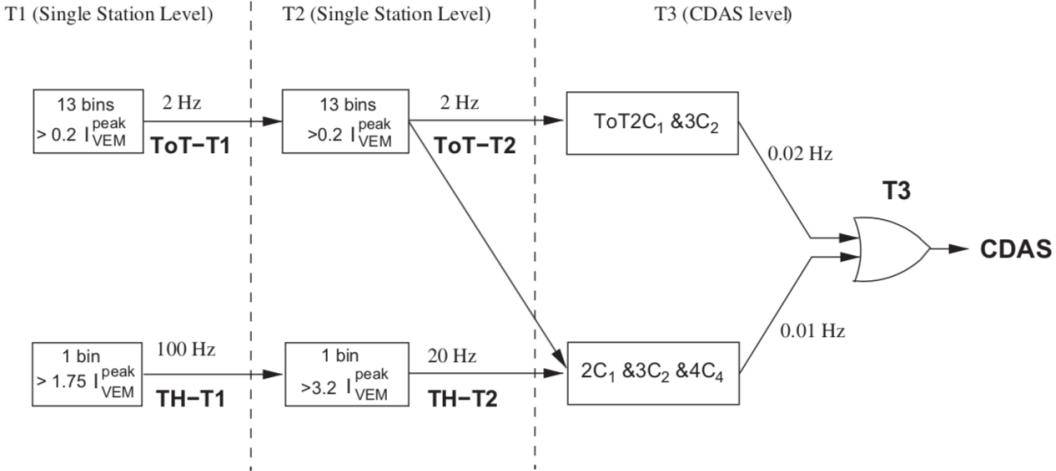


Figure 4.7: A diagram out of Abraham et al. [ubtriggers]. This shows what triggers and what signal strengths lead to what rates. It also shows when events are elevated. At the CDAS level, the letters, numbers and “ $\&$ ” signs represent the geometric configuration required to invoke a T3 trigger, where  $C_n$  is a detector from the  $n$ th circle out from the estimated shower core. See [ubtriggers] for more information.

it receives them. If appropriate conditions are met (represented in Figure 4.7), CDAS sends out a T3 “request” and relevant stations are read out, and their trace data, along with housekeeping and calibration data, are stored.

#### 4.4.3 Slow Control

When deploying autonomous detector stations over such a wide area as Auger, it is of the utmost importance to ensure that there are redundant solutions to solve as many problems as possible without physical maintenance. According, the UUB has three lines of defense: (1) Software controls, (2) Slow control and (3) Watchdog system. Many issues can be avoided by careful and clever software, but for those that can’t, having a redundant Linux system on board can save many otherwise untenable situations. This is one of the primary functions of the slow control module, largely developed at the University of Wuppertal by Karl Heinz-Becker and his team. The other main functions of slow control are to monitor various

voltages, the temperature and other easily measured conditions, as well as communicating with the radio, setting the PMT parameters and taking care of some of the housekeeping tasks. We discuss this here since GPS telemetry and PPSs are used by the slow control module to help monitor and save the UUB in problematic situations.

To ensure the integrity of the system, the slow control module itself makes sure that the Zynq™ chip is still running well. By communicating with it and monitoring its reference voltages, slow control can determine if operation is proceeding smoothly, or if there are issues. If there is something that needs to be addressed, the slow control can send a reset pulse to the Zynq™ chip. It also helps manage the correct boot order of the Zynq™; in particular, the external block RAM must be reset on startup to ensure proper functionality. The Watchdog chip (to the left of slow control in Figure 4.5), monitors the voltage parameters of both the Zynq™ and slow control, and is programmed to send a reset pulse to either or both as a last line of defense against failure.

#### 4.4.4 Hardware Design and Integration

Integrating all of the subsystems produced and creating a board layout that facilitates the necessary interconnections of each component was largely lead by the electrical engineers in Grenoble, FR at LPSC. From Table 4.1, we can see that all teams were responsible for helping integrate their component. In effect, this meant uploading a packaged “IP” (Intellectual Property, a Xilinx term for their block diagram inline packages) to a central github repository if you had designed custom hardware, or sending a specification of what IPs you want placed in the block diagram and what they need to connect to if you are using Xilinx IPs.

Among the constraints in board layout and integration, we have layering and induction concerns, thermal and mechanical stress concerns, crowding, power and finally FPGA us-

age concerns. The Xilinx tool set which is provided to work on the Zynq™ chip gives tool to manage power and FPGA concerns, while layering and induction can be simulated through SPICE or modeled in ORCAD. Mechanical and thermal stresses are simulated before board production, but in my experience working on the UUB, these things often only come out once a board has been produced. Hence, there is a pre-production run of boards ordered in projects like this, where the non-obvious issues are worked out.

#### **4.4.5 Software Development**

For the UUB, much of the software could be ported from the old UB as it was largely written in C. This involves using the PetaLinux command line utilities to instantiate programs to be cross-compiled, and then checking that the port worked. While the software for the UB is rather complex, only certain components needed to be ported for mission-critical tasks. Some parts were rewritten and others were simply transferred over. This work was done by Patrick Allison of OSU and later by Ricardo Sato who works directly for Auger as our science coordinator, along with contributions from the author and others towards the software pertaining to our specific work packages, i.e. GPS communications software.

#### **4.4.6 Calibration**

A number of calibrations must be done for AugerPrime, and as with the initial Auger detectors they must be done autonomously in situ. From a science perspective, the chief amongst these is the MIP/VEM calibrations. Since triggers are based on a station’s VEM measurement, or in the SSD, the MIP measurement, a poor calibration can directly affect a station’s science performance. The general method of finding calibrations for the SSD and WCD is to look at traces whose recording is initiated by the muon trigger. These are stored and analyzed at the station level, and used make a histogram of the charge deposited

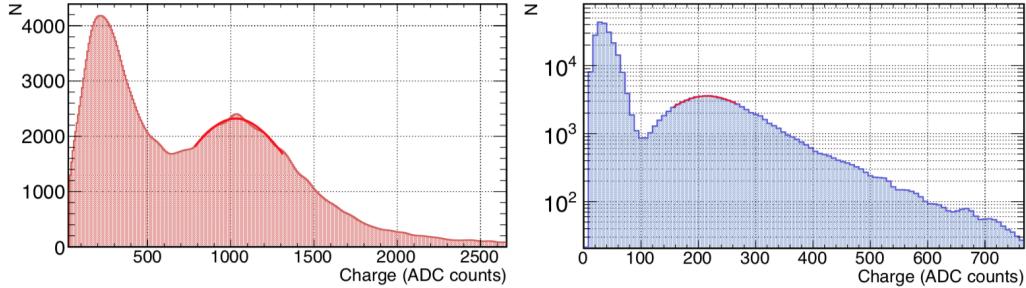


Figure 4.8: This charge histogram, from **zizhao**, shows counts from the muon trigger.

(calculated from integrating the traces). This reveals two distinct peaks. The left is the noise pedestal, which is used to find the appropriate ADC count offset. The right is the MIP/VEM peak (depending on whether you are looking at a WCD or an SSD). This is fit with a Gaussian and the center is used as the VEM value in all of the stations internal trigger calculations. Examples of the charge histograms for an upgraded station running in the EA are shown in Figure 4.8.

There are other important calibrations, such as the high voltage calibration. This is done by looking at the noise level in the ADC traces and handing histograms of that off to slow control. This is a largely straightforward process which, usually goes well; when it doesn't succeed stably, it usually indicated an issue with the PMT which will have to be fixed manually. Additionally, in order to determine important parameters from deconvolving the different components of the shower, MIP vs. VEM diagrams must be made (see section 4.2 for more details). We have shown one of these in Figure 4.9.

## 4.5 GPS and Timing

Amongst the work packages shown in Table 4.1 and Table 4.2, the one that I and my colleagues at Case Western are responsible for is the timing and GPS integration. In order to put together any cogent picture of an air shower event, each station must know what

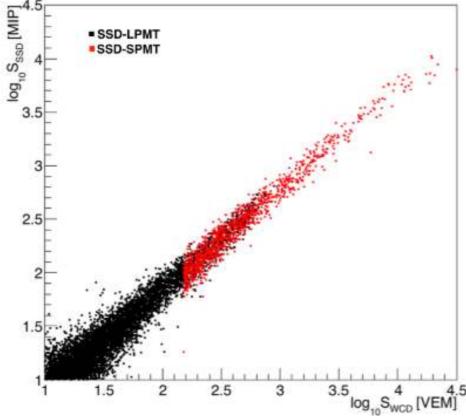


Figure 4.9: This diagram from **dynrang** shows a typical MIP vs. VEM calibration curve. While it looks linear, a closer inspection shows that the VEM signals are slightly higher than the MIP signals, and this makes sense because a muon will deposit more energy over its  $>1\text{m}$  traversal of the WCD than it will in its short traversal of the SSD.

time it is according to some shared standard. In smaller experiments, it is sufficient to simply collect all data at a central location using the same cable lengths and electronic delays in each detector. This was, for example, the configuration of Volcano Ranch and AMANDA. In medium sized experiments, where stations can still be connected via wires, but data transfer must be asynchronous, time is often distributed over networks or through distributed protocols. Examples of this include the Large Hadron Collider and IceCube (see subsection 3.3.3).

For Auger, connecting each station with a wired connection is not technically nor financially feasible, and so time must be distributed in some other way. The cheapest and most accurate solution is to use the constellation of atomic clocks that float over our heads 24/7, delivering accurate timing information for free: the Global Positioning System (GPS).

### 4.5.1 GPS Basics

To inform the coming sections, namely the GPS integration in section 4.6, the spatial correlations in section 6.3 and the receiver testing in section 5.2, we will give an informative

overview of the relevant facets of the GPS.

Behind GPS is the basic idea of broadcasting the position and current time of a set of satellites, then multiple of such signals are caught by the receiver, which uses its own clock to determine the time-of-flight of the signal. These times-of-flight are then converted into what are called pseudoranges,  $\rho_i = c(t_r - t_i)$ , where  $t_r$  is the time on the receiver's clock when the signal was received and  $t_i$  is the time on  $i$ th satellite's clock, in its reference frame at time of transmission. We can then compare each pseudorange with equations measuring the distance between the satellite and receiver to create a system of equations which will herein be referred to as the GPS equations,

$$|r - r_i| = \rho_i.$$

Here  $i$  is an index that runs over the available satellites, of which there must be at least four to fix position and time. More explicitly, these equations are:

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= c^2(t_r - t_1 + \Delta t)^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= c^2(t_r - t_2 + \Delta t)^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 &= c^2(t_r - t_3 + \Delta t)^2 \\ (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 &= c^2(t_r - t_4 + \Delta t)^2, \end{aligned} \tag{4.3}$$

where  $\Delta t$  is a correction term to be applied. We can see that these equations can be manipulated in various ways. If we are only concerned about position, we can forget solving for the time coordinate and use only three satellites (i.e. equations). If we want to know only our current time  $t_r$ , we can fix our last known position and only need to use one equation to find the time. It is common practice to re-parametrize the problem to use pre-corrected

pseudoranges and solve for the unaccounted-for error. In equation 4.3, we can think of this as moving the  $\Delta t$  correction term outside and solving for it, assuming we have the correct  $t_i$ .

Of course, in practice we do not want to throw any information out; furthermore, we are dealing with a non-linear system of equations. These do not always admit an easy solution and can have somewhat unpredictable behavior. This is why, in any receiver's electronics, the GPS equations are always solved using either a least squares method, or an iterative solution allowing the receiver to use as many satellites as it can see. These methods allow use of inconsistent sets of equations, meaning that we can use more equations than degrees of freedom, and obtain a more accurate solution. As Langley notes in [gpsworld], almost all receivers use iterative methods to solve these equations because they can do so quickly and to a higher degree of accuracy.

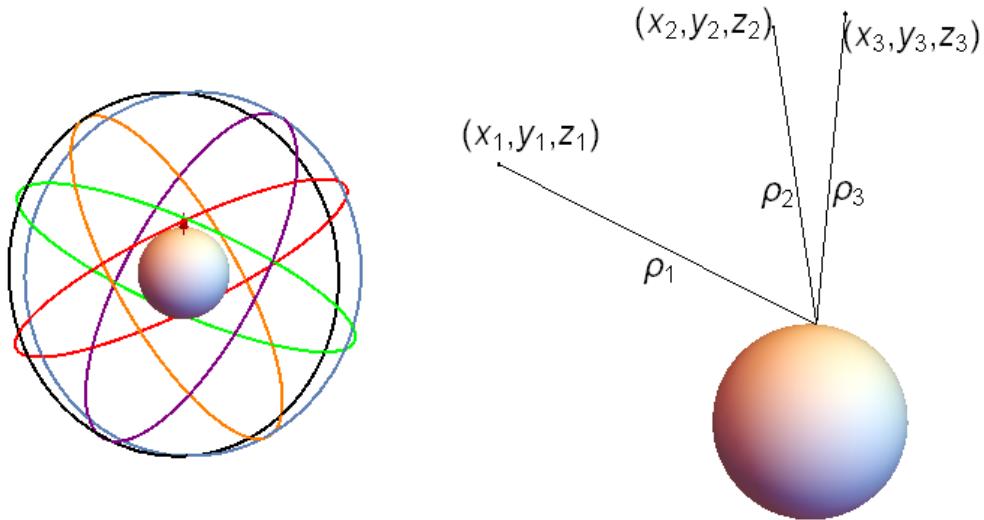


Figure 4.10: Left: A diagram of the available GPS constellation orbits of the original 24 satellites. The red arrow indicates the North Pole. Right: this diagram shows the coordinates and quantities used in GPS calculations.

## 4.5.2 GPS Constellation

The Global positioning system was fully brought into service in 1995 with a constellation of twenty four satellites. The original specification for the system was largely dictated by the military's requirements, and today they still run a high frequency encrypted broadcast over the GPS system. As originally designed, a slow rolling local error was added in to the civilian channel, to ensure that only the military was able to get the initially specified minimal  $10 - 15$  m spherical error probable (SEP: the radius from the mean of a collection of points within which 50% of the points lie). Civilian users received this slow rolling error, officially called Selective Availability resulting in inaccuracies as large as 100 m [thompson]. Although this is not a crippling error if you are trying to navigate a ship or airplane, it is certainly too far for a driver to decide what street they are on or a detector station to obtain accurate nanosecond timing. In 2000, Selective Availability was turned off by executive order, and the next generation of satellites will be built without this feature.

When the military published the unclassified portions of the GPS standard [milspec], they did not include the exact relativistic corrections necessary. The standard includes an approximate correction in the form of  $\vec{v} \cdot \vec{r}/c^2$  ( $v$  and  $r$  are the velocity and position of the satellite,  $c$  is the speed of light), but this turns out to not be nearly enough to reach their desired accuracy. This is probably a mechanism to offload the programming of corrections to the receiver manufacturer; in fact, each manufacturer keeps their positioning algorithms and corrections as trade secrets [ptoday].

The original geometry of the GPS constellation was 24 satellites at  $55^\circ$  inclination in six different orbits, each azimuthally separated by  $60^\circ$  (see figure 4.10). It has since been expanded to 32 satellites for greater worldwide coverage. With the original geometry, a user should always have at least six satellites in view, allowing for redundancy in the constellation and giving more data for the numerical solution of the positioning equations.

### 4.5.3 GPS Receiver Communications Protocols

Since we will soon discuss the integration of the new GPS receivers for AugerPrime, we will give a general overview of the communication protocol used by our GPS receivers.

Due to Motorola's early dominance in the GPS market, many receivers that are not for maritime applications use the "Motorola Binary" (MB) communications protocol. For the stations running in Auger currently (less the EA), the Motorola Oncore UT+, is the chosen GPS unit. Information on the GPS units we tested for AugerPrime is in subsection 5.2.2. The 'Maritime Protocol 2000' from the National Maritime Electronics Association (NMEA) takes an approach of simplicity in interpretation and integration. This protocol uses plain ASCII letters and packet spacing to make message relatively human readable. It also can operate at a faster baud rate to make up for the lower information density.

The Motorola Binary protocol specifies that all connected devices will operate at 9600 baud (or bits per second). The way it is written uses bit and byte packing techniques to absolutely maximize the amount of information contained in a message. In general, it is absolutely not human readable and even has to be interpreted to fit into standard memory structures. In general, the fields in MB are character, short, integer or sub-byte, i.e. 1, 2, 4 or <1 bytes. In the cases where they are less than one byte, each bit generally represents a boolean field.

An example of the type of messaging that would be sent in MB is shown in Figure 4.11. In this specification, we can see how the bytes of different fields are packed in. It should be noted that MB is a big-endian format, i.e. the first byte (left-most if you like to hearken back to the tape machine formalism) represents the highest value.

Each message is on the order of 100 or so bytes, and so simple math tells us that the 154 byte Time, Position and Status @@Ha message will take  $(154 \times 8)/9600 = .12s$  or more

than a tenth of a second, and therefore properly handling this long time-of-transmission in software in a non-blocking way is important and will be discussed in both the context of the CWRU HEA group's TIM instrument (section 6.2) and the Auger UUB (section 4.6).

```
@@Hamddyhmsffffaaaaoooohhhmmmmaaaaoooohhhh
mmmmVVvhddtimsidd (repeat 'imsidd' series for remaining 11 channels)
ssrrccooooTTushmvvvvC<CR><LF>
```

Date	
m = month	1..12
d = day	1..31
yy = year	1998..2079
Time	
h = hours	0..23
m = minutes	00..59
s = seconds	0..60
ffff = fractional second	0..999,999,999 nanoseconds
Position (Filtered or Unfiltered following Filter Select)	
aaaa = latitude in mas	-324,000,000..324,000,000 (-90°..+90°)
oooo = longitude in mas	-648,000,000..648,000,000 (-180°..+180°)
hhhh = GPS height in cm	-100,000..+1,800,000 (-1000..+18,000m)*
mmmm MSL height in cm	always 0,000,000 with M12M
Position (Always Unfiltered)	
aaaa = latitude in mas	-324,000,000..324,000,000 (-90°..+90°)
oooo = longitude in mas	-648,000,000..648,000,000 (-180°..+180°)
hhhh = GPS height in cm	-100,000..+1,800,000 (-1000..+18,000m)*
mmmm = MSL height in cm	always 0,000,000 with M12M

Figure 4.11: Pictured here is the @@Ha message from the i-Lotus M12M users manual [**m12muser**]. We include this as an example of how the Motorola Binary format is packed and interpreted.

MB as implemented on the receivers we tested for AugerPrime, which utilized identical command sets, offers a number of commands which are used to provide the essential timing functionality desired. In the early days of Auger, the period of Selective Availability was still at the forefront of the community's thinking and there were still valid questions about how well the GPS constellation would work in South America. If we look at Figure 4.10, we see that the inclination of the orbits, and the fact that they have Half Sidereal Day (HSD) periods, means that they will process around the Earth and there even spacing will

effectively provide even coverage everywhere. A simulated plot of coverage is shown in Figure 4.12.

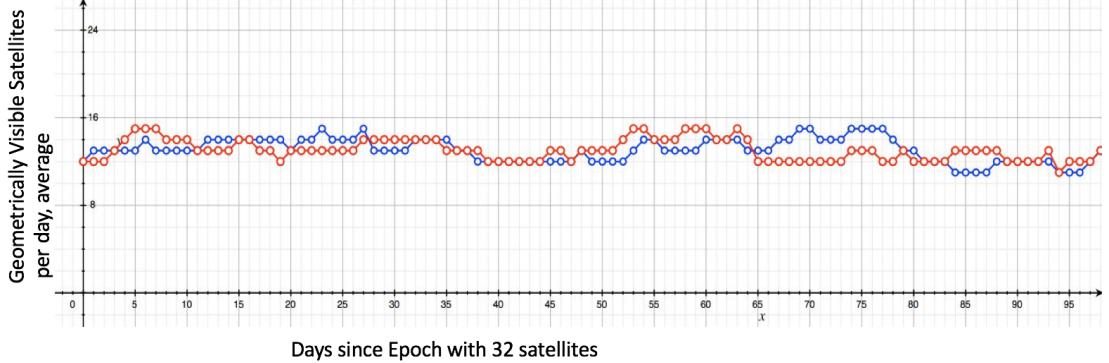


Figure 4.12: This plot, created by then-undergraduate Peter Fedrizzi under my supervision, shows the average number of satellites visible over Malargüe (i.e. Auger, for our purposes) and Cleveland in a given day.

#### 4.5.3.1 Message Selection

The aforementioned concerns led the designers of the UB software to choose three UT+ messages to be recorded in each SD station: @@Ea, @@En and @@Bb. Respectively these are the 8-channel: Time/Status, T-RAIM, and Satellite Status messages. @@Ea is in parity with @@Ha, where both give all of the information the receiver can provide about its positioning and timing solutions, as well as all operational information about the receiver, including temperature, satellite (Space Vehicle, SV) health and clock speed amongst others. @@Hn and @@En are the Time-Receiver Autonomous Integrity Monitoring messages. These contain estimates on the accuracy of the timing solution, as well as the very important Clock Granularity Message (see subsubsection 4.5.3.2), and some further information on the health of the constellation. Finally, the @@Bb message, implemented similarly in all of the relevant receivers, gives relevant signal processing and local orbital parameters (elevation and azimuth from the receiver).

In the UUB, we (Ricardo Sato and the author), looked at the actual science goals and

realized that we were using a lot of processing power parsing and saving messages that did not contribute to the physics we are trying to accomplish. Namely, the @@Ha message can be replaced by the approximately 75% shorter @@Hb message, which does not give irrelevant information about the receiver's operation, and instead sticks to only useful time and positioning data. Beyond that, the @@Bb message was cut out entirely since concerns about the operation of GPS have more or less subsided (see section 6.3 for some that remain). This brings us from 324 bytes per second to parse, down to 132 bytes total. If need be, all of the messages we've mentioned above and others are implemented in the UUB software and can be enabled remotely (see section 4.6).

#### 4.5.3.2 Sawtooth Correction: The Clock Granularity Message

If we take a look at the M12M's data sheet, it shows a <2 ns PPS (Pulse-per-Second) accuracy, however it sports a 40Mhz oscillator with 25ns coarse grained timing accuracy. The mechanism through which it achieves its higher PPS accuracy than the oscillator would normally allow for is by accurately calculating the phase of a clock cycle in which the PPS should truly land and communicating this via its serial connection in the @@Hn message. This is called the clock granularity message, which takes the form of a bipolar sawtooth pattern, showing the precession of the true PPS around the receiver's clock.

The mechanics of this precession are, at surface level, easy to understand (see Figure 4.13): the receiver's oscillator is not perfectly trained to the GPS constellation, and we need not require it to be. If we ask about the specific form of the sawtooth over a given interval, such as that shown in Figure 4.14, more complicated dynamics come into play. Some of these will be discussed in section 6.3, but to summarize, all of the effects that can come into play in the receiver, such as sky conditions and temperature modulation affect this pattern.

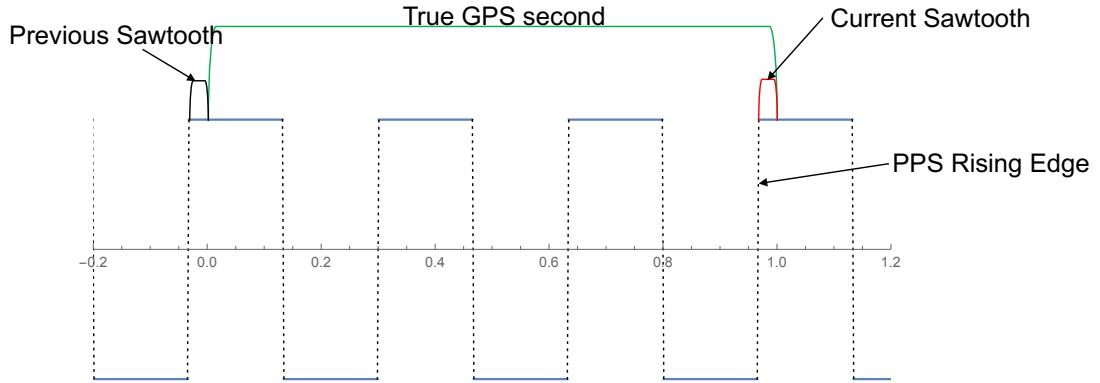


Figure 4.13: Here we have a schematic diagram showing how the sawtooth correction works. By finding the phase difference between the leading PPS edge and the actual beginning of a GPS second, the receiver can deliver a serial message which allows the user to obtain a very accurate beginning-of-second time. In this diagram, we have used 4Hz for the square wave to represent the GPS Receiver's clock.

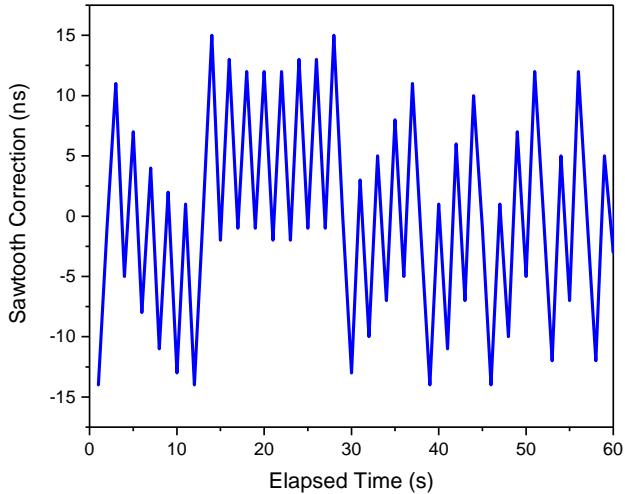


Figure 4.14: Provided here is a plot of 60 seconds of M12M sawtooth corrections. This plot shows the sawtooth behavior, as well as a phase alignment shift around 15 seconds, where, because of the precession of the true PPS around the output PPS, we see a change in the sawtooth pattern and then a change back. These occur on the time scale of anywhere from hours to minutes, depending on the sky conditions and how long the receiver has been running.

#### 4.5.3.3 WGS-84

To calculate distances and understand altitudes and positions as needed both in Auger and for a number of other HEA group activities (e.g. subsection 6.3.4, section 6.3), we will need an introduction to the useful details of GPS coordinates.

The coordinate system used by GPS is called the World Geodetic Survey of 1984, or WGS-84. While there are plenty of technical details to discuss, they are largely outside the purview of this work. WGS-84 represents an extremely detailed model of the earth including tidal and tectonic effects. For us, the only relevant portions is referred to as the WGS-84 Ellipsoid, and it tells us three very important parameters. From **wgs84**, the semi-major axis of the coordinate system the Earth is embedded in via this standard is  $a=6378137$  m, and the semi-minor axis is  $c=6356752.3$  m. The radius as a function of latitude is:

$$r = \left( \sqrt{\frac{\cos(\theta)^2}{a^2} + \frac{\sin(\theta)^2}{c^2}} \right)^{-1}, \quad (4.4)$$

where  $\theta$  is the latitude. This is a simply derived fact of 2D geometry. For the purposes of this work, this is really all we need to know about WGS-84.

## 4.6 GPS Integration

To truly integrate the chosen GPS unit, the Synergy SSR-6Tf, we need to address its hardware requirements, and its software requirements. Before beginning this discussion, we would like to note that the i-Lotus M12M receiver and the SSR-6Tf are entirely interchangeable outside of minor performance differences which will be discussed in section 5.2. Some design choices surrounding the GPS integration have already been discussed in this chapter, so this particular section represents the majority of the software and

firmware work, but not all of it (e.g. subsubsection 4.5.3.1). To begin, we will discuss the firmware design tools, and then we will move into a discussion of the actual software architecture. Along the way, we will discuss the design choices that were made by the timing team.

### 4.6.1 The Purpose of GPS in AugerPrime

In implementing GPS receivers for Auger and the AugerPrime upgrade, we give the mission-critical ability of synchronizing time across the vast distances covered by the SD. The time differences in stations for moderately inclined showers will be on the order of small numbers of microseconds, but for near vertical showers, accuracy on the order of tens or hundreds of nanosecond is required to obtain sub-degree reconstruction. This accuracy is already marred by the variable width of the shower front, and so we can keep our aggregate error down by having the best timing and time synchronization possible.

Beyond timing, the AugerPrime upgrade gives the collaboration the opportunity to implement GPS receivers which have positioning capabilities that are more accurate than those of human surveyors. Upon construction of the array, trepidations related to Selective Availability and the relatively poor accuracy of the UT+ led the collaboration to employ surveyors to find accurate positions for all 1660 SD stations. These were then uploaded to the UT+ models. Ultimately, it was found that the UT+ models, by taking week long survey campaigns, could find their position better than the human surveyors anyway, but with the new SSR-6Tf receivers, excellent positioning data should be immediately available and a survey campaign can be used to obtain positions to much greater accuracies than human surveyors.

## 4.6.2 Hardware Design

The first step in implementing the AugerPrime GPS receiver is to build a firmware that allows communication with it. The prime character here is the UART or Universal Asynchronous Receiver and Transmitter. A UART allows serial communication without clock synchronization between two devices. The M12M and SSR-6Tf receivers utilize a slightly modified version of the RS-232 communications protocol. Namely, they use a logical-high voltage of  $2V < V_{high} < 3V$ . Precariously, this lands between the commonly used 1.8V and 3.3V logic standards, however we have extensively tested it at 3.3V and it works as expected, even for years on end. In choosing a UART and implementing it, we have to consider a few criteria:

- Reliability- How long will the UART run for without resets or software-based housekeeping?
- Simplicity- How much FPGA resources will the UART take up?
- Flexibility- Can the UART settings be changed, or are they hard coded?

Ultimately, the choice of UART boils down to the latter two conditions, since both options that we looked at were extremely reliable as far as our testing could discern. Since, as was mentioned before, in the UUB design, Zynq™ slice space is at a high premium, whichever UART is chosen should be evaluated based on being able to accomplish the task at hand with an absolute minimum of resource utilization.

From a logical point of view, an FPGA is simply a device which connects sets of logical gates (AND, OR, XOR, etc.) to emulate actual hardware functions. A known principle at work here is that any piece of digital hardware or software can be broken down into a logical construct using logic operations. FPGA resources boil down into units of slices. A slice contains three high level types of resources which are used to create the requested

set of logical gates. These are LUTS, registers and multiplexers. The Look-Up Tables, or LUTs, give the FPGA the ability to handle functions which need to take a semi-continuous input and map it to a semi-continuous output. Registers and flip-flops (at the level of this task, these are effectively the same thing) allow the storage of information, and the synchronous passing of it along a data path. Finally, the multiplexers, which are effectively invisible to the firmware programmers, are used by the design software to make the connections between various gates. In some sense they act as hardware relays, but without any electromechanical movement.

For AugerPrime, we will be using the UART only to communicate with one device at one baud rate. Therefore, flexibility is the less important criteria, and simplicity dictates our UART choice. In Table 4.3, the resource utilization for the UART 16550 ([**16550**]), a long standing industry standard, and that for the UART Lite ([**lite**]), a Xilinx original IP, are shown. Further on, we have a top-level diagram for the UART Lite in Figure 4.15. Based on the lower resource utilization, we made the recommendation that we use the UART Lite. In shakedown testing it performed admirably and was, in finality, used in the UUB and all of the TIM-based projects the CWRU group took part in.

Device	Slices	Slice Registers	LUTs
UART 16550	118	308	347
UART Lite	49	79	119

Table 4.3: Here are the different resource utilizations for the relevant configuration of the two Xilinx UART IPs in question [**lite**, **16550**].

#### 4.6.2.1 Introduction: Xilinx Vivado™ Design Tools

Until relatively recently, FPGAs and their programming were riddled with hazards. For many FPGAs until the early 2000s, they could only be used once, i.e. once a firmware design was implemented in them, it could not be changed. They were, therefore, useful

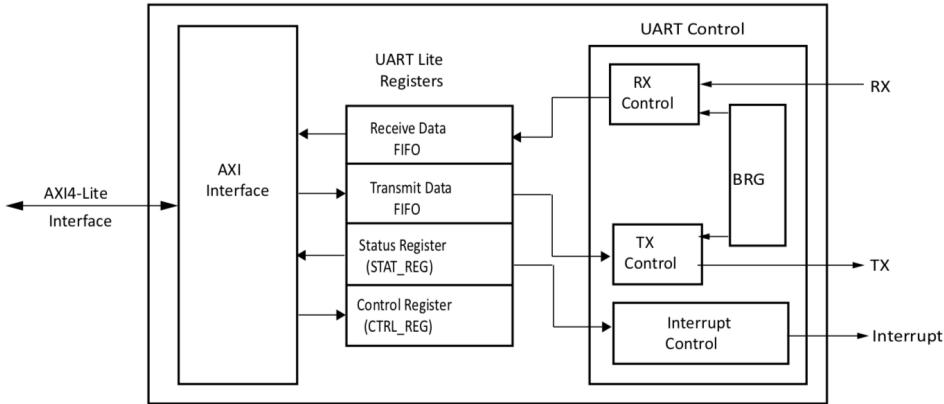


Figure 4.15: From **lite**, we have the top-level diagram for the UART Lite IP. This is identical to the top-level diagram for the UART 16550, but its resource utilization is lower since the BRG (Baud Rate Generator) is locked in during firmware synthesis and implementation.

only in testing designs that would most likely become their own Integrated Circuit (IC) or Application Specific Integrated Circuit (ASIC, IC usually refers to a single chip, while an ASIC is almost always a type of printed board with ICs on it.) In recent years, they have become practically infinitely reprogrammable. This is one advance that makes them absolutely ideal for AugerPrime, where we can actually rewrite the firmware and test it in the lab to implement new triggers especially, but any other minor firmware change as well. After testing, the new firmware can be sent directly and wirelessly to each of the 1600 stations. From a science point of view, this is golden; we now can modify the way our detector operates to look for more interesting patterns or to more efficiently detect mundane events.

Perhaps the more important advance than the flexibility of FPGA program, is the simplicity in the design tools. Most FPGA manufacturers now have fully graphical design tools, which greatly lowers the effective “barrier for entry” in FPGA programming. Traditionally, FPGAs are programmed in hardware verification languages (sometimes called hardware design languages) to produce what is called a *bitstream* (we have been veiling this

with the term “firmware design”, the final product of a firmware design is a bitstream). The design suite that was used for AugerPrime of course comes from Xilinx and really consists of two important products, Vivado<sup>TM</sup> and the Source Developer’s Kit (SDK). Vivado<sup>TM</sup> is used to create the bitstream, and allows both graphical block diagram editing as well as hardware design language editing. To stymie confusion, from here on out, hardware design language refers to any available hardware verification language, HDL, Verilog or VHDL.

The general work flow in Vivado<sup>TM</sup> is to start a project, which prompts you to select the type of hardware you are using. Vivado<sup>TM</sup> knows the descriptions for all Xilinx programmable devices and imports a specification describing the devices abilities and port structure. The port structure essentially lists all of the available pins on the device. In some cases such as when working with their Zynq<sup>TM</sup> evaluation board, the ZedBoard<sup>TM</sup>, the software already knows how the port structure maps onto physical ports, and can take advantage of this. In the case of the UUB, all ports must be manually mapped. This involves the use of a *design constraints file*, in the .xdc format. These files allow specification of a logic standard and an external port to be mapped to a physical pin on the Zynq<sup>TM</sup> chip.

All of the other functions of Vivado<sup>TM</sup> pertain specifically to what is happening on the FPGA. This is where the *block diagram* comes in. In Vivado<sup>TM</sup>, there are 3 interconnected units you have to manage. The block diagram is the portion where you can invoke IP, which stands for Intellectual Property, but effectively these are firmware versions of hardware devices, i.e. ICs. The block diagram allows you to spatially arrange and interconnect IPs and is where most of the work in Vivado<sup>TM</sup> is done. Most standard electronics have a corresponding IP and utilizing it involves invoking the IP, connecting its buses and ports to the other elements of the block diagram, customizing the IP in whatever way you need, and finally, if needed, mapping it to an AXI (Advanced eXtensible Interface) address. AXI is the primary communication bus in many ARM devices, and certainly is the most widely

used in the UUB.

Beyond the constraints file and the block diagram, the last design parameter we want to control at a high level is the top level wrapper. This is a piece of hardware design language (Vivado™ supports both Verilog and VHDL generation) which relates the external connections of the constraints file to the internal connections of the block diagram. It is in here that we can, for example, connect a signal line to an external port. We can also map such a signal line to split off into an LED for monitoring, or connect it to two ports to allow a signal to be used by multiple IP in the block diagram.

#### 4.6.2.2 Implementing the UART

As elucidated in subsubsection 4.6.2.1, we have to look after three tasks to implement the UART in the UUB. First, we instantiate the block diagram and make our customizations, then we add a connection to the top level wrapper, and finally we make a external connection with the constraints file. Adding the UART Lite is as simple as right clicking on

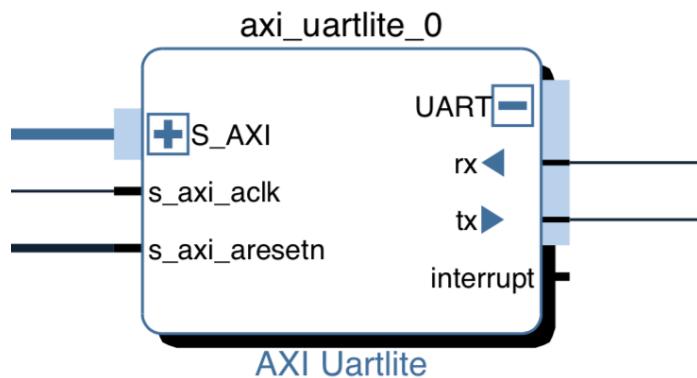


Figure 4.16: This is the UART Lite's block on the block diagram. It is shown here to give some idea of what the IPs in the block diagram look like and to show the ports that must be connected for it.

the block diagram in Vivado™ and searching for it. Upon adding it, we are greeted with Figure 4.17. Here we can select the baud rate and parity bits we want, which are respec-

tively 9600 and 0 for the SSR-6Tf. We can then accept these parameters and move to the AXI Mapping tab, where Vivado's™ built-in address mapping tool does a fine job of taking care of this part of the task automatically. Next, we have to make the connection

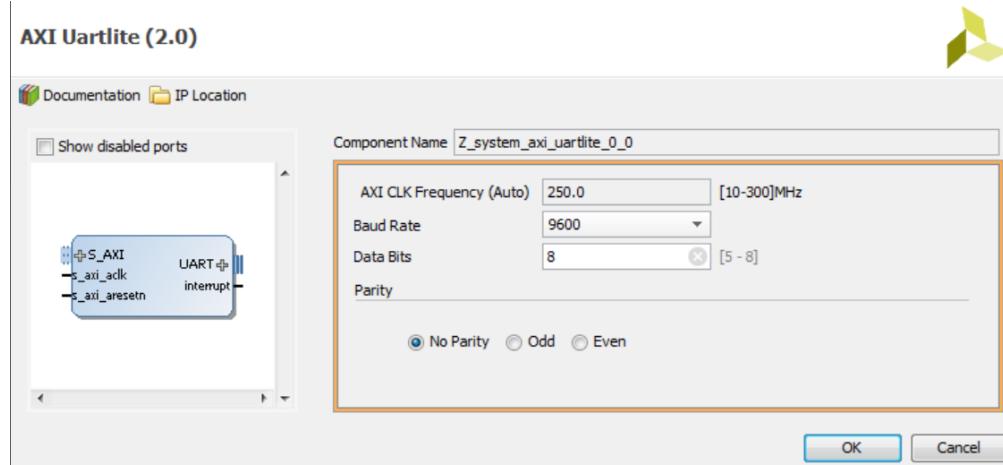


Figure 4.17: In this Vivado™ window, we are given our rather sparse options to customize the UART Lite. Those shown here are the correct configuration for the SSR-6Tf GPS receiver.

from the PL to the top level wrapper. Vivado™ can do this automatically, but it will destroy any custom firmware you have written into your wrapper. We would endorse saving your wrapper elsewhere, allowing Vivado™ to automatically generate the wrapper, and then re-inserting your custom PL code appropriately. This can sometimes be a moderately confusing and lengthy task and in my experience, it must be done in one sitting or you will certainly forget some detail of what you have done. This is also an excellent place to take good notes on what you have done. In figure Figure 4.18, we have shown a snippet of the top level wrapper for a custom Vivado™ project. The UUB's top level wrapper is not publicly available, although the hardware description file which contains binarized information from the top level wrapper is available through the [uub-firmware github](https://github.com/auger-prime-sde/uub-firmware), <https://github.com/auger-prime-sde/uub-firmware>, under work package 5.

In UUB’s wrapper, these signals are called *GPS\_RX* and *GPS\_TX* for GPS receive and transmit, while in Figure 4.18, the analogous signals are *sin* and *sout*. To clarify, *signal* is the correct hardware design language term for this object, which represents a physical signal carrying wire. Once we have the wrapper taken care of, we need only attend to our final task: correctly constraining the signals to physical ports or pins on the Zynq™ chip. This is as simple as adding two lines per signal to the constraints file (usually just called *constraints.xdc*). One line specifies the signaling standard, e.g. 1.8V or 3.3V, and the other specifies the “package pin” or physical site on the Zynq™ chip that the signal is to end up in. In the case of integrating the SSR-6Tf into the UUB, these lines are:

```
set_property PACKAGE_PIN V10 [get_ports GPS_TX]
set_property PACKAGE_PIN V9 [get_ports GPS_RX]
set_property IOSTANDARD LVCMOS33 [get_ports GPS_TX]
set_property IOSTANDARD LVCMOS33 [get_ports GPS_RX]
```

With this, we should have firmware for a working UART Lite. This is ultimately how GPS communications are implemented in the firmware for AugerPrime. Specifications of the ports and pins are publicly available in **uubconst**.

#### 4.6.2.3 Physical Connection

On the UUB and in the various testing enclosures, the GPS unit is mounted on a 60mm×40mm pad, with a 20 pin connector. The technical drafting diagram is shown in Figure 4.19 and the physical board layout in Figure 4.20. The specification of most importance to the work described above is in Table 4.4, although ultimately the routing of these pins is handled by the integration team at LPSC Grenoble, headed by Eric Lagorio, one of the lead electrical engineers on AugerPrime. I should also clarify that while these images are for the M12M, the analogous images for the SSR-6Tf are only cosmetically different. The pinout is the

```

181     wr_en);
182     inout [14:0]DDR_addr;
183     inout [2:0]DDR_ba;
184     inout DDR_cas_n;
185     inout DDR_ck_n;
186     inout DDR_ck_p;
187     inout DDR_cke;
188     inout DDR_cs_n;
189     inout [3:0]DDR_dm;
// 190     inout [31:0]DDR_dq;
191     inout [3:0]DDR_dqs_n;
192     inout [3:0]DDR_dqs_p;
193     inout DDR_odt;
194     inout DDR_ras_n;
195     inout DDR_reset_n;
196     inout DDR_we_n;
197     inout FIXED_IO_ddr_vrn;
198     inout FIXED_IO_ddr_vrp;
199     inout [53:0]FIXED_IO_mio;
200     inout FIXED_IO_ps_clk;
201     inout FIXED_IO_ps_porb;
202     inout FIXED_IO_ps_srstb;
203
204     input REF1PPS;
205     input TEST1PPS;
206     input GRES;
207     output LED;
208     output LED2;
209     output LED3;
210
211 //   input [31:0]din; //might need to comment these back in, really not sure what
212 //   input [1:0]gpio_i_tri_i;
213 //   input [31:0]gpio_2_tri_i;
214 //   output [0:0]gpio_tri_o;
215     input sin;
216     input sin_1;
217     output sout;
218     output sout_1;
219     input rd_en;
220     input wr_en;
221
```

```

Figure 4.18: This Vivado™ window shows the Verilog wrapper used in a custom project. As mentioned in the text, the UUBs wrapper is not publicly available, however it is similar with a bit more going on.

same and so are the features, however the SSR-6Tf users manual is still in draft form and not ready to be released to the public. We in the HEA group at CWRU have been granted a copy for use, but not distribution.

### 4.6.3 GPS Communications Software

After implementing the hardware to communicate with the GPS, we need to write the software. This was a joint effort between Ricardo Sato and myself. The software really consists of 2 main parts: the initial message handling is operated through what we call a

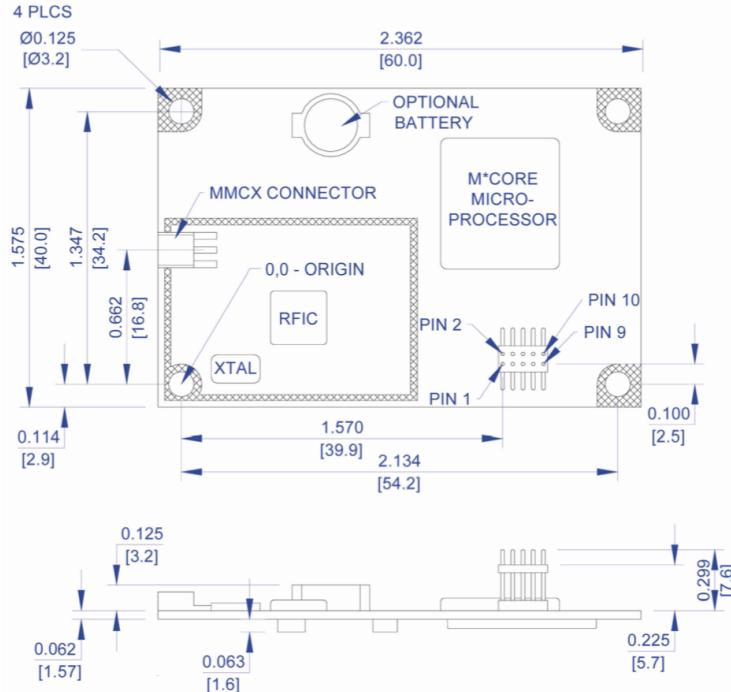


Figure 4.19: Here is a technical drafting of the M12M and SSR-6Tf's shared footprint from [m12muser](#).

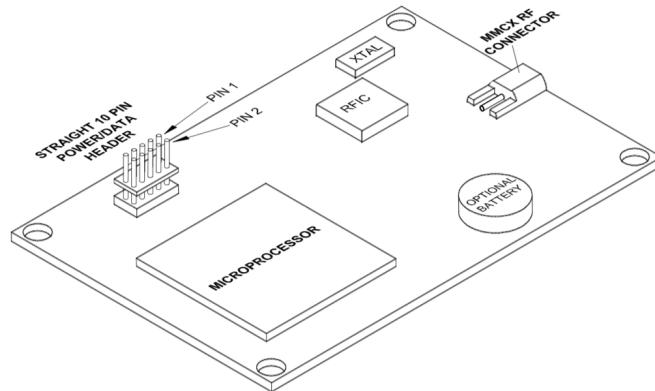


Figure 4.20: This rendering shows the physical layout of the M12M receiver, but more importantly for our purposes, it shows the pin ordering which corresponds to Table 4.4. From [m12muser](#).

*parser*, and then the data is put into classes using *packing functions*. We should note now, that a careful reading of subsection 4.5.3 is an absolute prerequisite for understanding what

| Pin # | Signal Name  | Description                 |
|-------|--------------|-----------------------------|
| 1     | TxD1         | Transmit Data (3V logic)    |
| 2     | RxD1         | Receive Commands (3V logic) |
| 3     | +3V PWR      | Regulated 3Vdc Input        |
| 4     | 1PPS         | 1 pulse-per-second output   |
| 5     | Ground       | Signal and Power common     |
| 6     | Battery      | Optional External Backup    |
| 7     | Reserved     | Not currently used          |
| 8     | RTCM In      | RTCM correction input       |
| 9     | Antenna Bias | 3V-5V antenna bias input    |
| 10    | Reserved     | Not currently used          |

Table 4.4: This table from **m12muser** shows the all-important pinout. These pins are then routed, some into the FPGA and some into other components on the UUB.

will be said in this subsection.

#### 4.6.3.1 Parser

Essentially, the parser performs two functions, it figures out when a message has been received and then it figures out what type of message it was. These functions are accomplished in the UUB in the main loop of a file called *gps\_main.c*. The *select* command, shown in-situ in Figure 4.21 is the real workhorse of the parser. This command takes five arguments: the first is an integer describing the place in the second argument, which is a file descriptor with read privileges. The second and third arguments are unused in the AugerPrime software as they wait for a file descriptor to become available for writing and to monitor a file descriptor for exceptions, respectively. The final argument gives a timeout, which in our case is set to 10s. When the *select* command is activated, it passes the point in the readable file descriptor to *gps\_receiver\_read\_msg*, which takes in the message until it encounters a valid message. This is specified by a carriage return and line feed, proceeded by a correct XOR checksum of the entire message. Due to the nature of XOR, it is sufficient to take the XOR of the message from the very beginning, before the first “@” and go all

```

while(1){
    aux=readfds;
    timeout.tv_sec=10;
    timeout.tv_usec=0;
    if(select(gps_rec_fd+1,&aux,NULL,NULL,&timeout)==1){
        usleep(80000);
        read_flag=1;
        do{
            msg_buff_len=gps_receiver_read_msg(gps_rec_fd,msg_buff,read_flag);

```

Figure 4.21: Shown here is the main loop of the AugerPrime GPS message parser. Note that the select statement effectively does all of the work of monitoring for changes in the file descriptors.

the way through the checksum. This should result in a value of 0x00, indicating that the message XOR matched the checksum sent by the receiver. Once this is true, *gps\_receiver\_process* is activated, with the data from the most recent message. This function, shown in Figure 4.22, figures out the message type from the available messages implemented. Finally, the type is passed back into the parser loop where a set of if/else statements activate the appropriate packing function and extract the needed data. The ultimate product of this loop is the *gps\_process\_info* function, which globalizes the information parsed and packed here for the rest of the UUB software to make use of.

#### 4.6.3.2 Classes and Packing

Once the *gps\_receiver\_process* function has decided on the correct message type, the message gets passed back to the parser process in *gps\_main* where it is packed into a struct for readout. Packing these data into structs allows and sensible access to any of the parameters contained within, and is widely considered the “correct” way to handle serialized data. From a programming point of view, creating these structs and packing functions is relatively straightforward.

To create a struct and packing function for a particular message, we need only look in the M12M User’s Guide [**m12muser**], and create the appropriate fields for each piece

```

int gps_receiver_process(unsigned char *msg,int len)
{
    if(GPS_RECEIVER_MIN_MSG_LEN<=len){
        if(memcmp((msg+2),"Hb",2)==0){
            return(1);
        }
        if(memcmp((msg+2),"Hn",2)==0){
            return(2);
        }
        if(memcmp((msg+2),"Ha",2)==0){
            return(3);
        }
        if(memcmp((msg+2),"Bb",2)==0){
            return(4);
        }
        if(memcmp((msg+2),"As",2)==0){
            return(5);
        }
        if(memcmp((msg+2),"Cj",2)==0){
            return(6);
        }
        return(0);
    }
    return(1);
}

```

Figure 4.22: Here is the process which decides what type of message has been received. It simply asks if the first two letters match any of the pre-assigned messages. If not, it passes back to the *gps\_main* process and resets the receiver's messaging parameters.

of data we desire to store. In keeping with good practices, we have made all of the data from any of the possible messages available, except for bit fields, which are generally not collected in housekeeping. As an example of making a struct, we can look back at Figure 4.11, and compare it with Figure 4.23. In Figure 4.11, we see that each parameter has a big-endian set of bytes that it corresponds to. The largest of these is an int, and the smallest is a char, although as mentioned before, some chars represent bit fields.

Before continuing on, we should also address the necessity of the packing function. Why is it that we cannot just, for example, *memcpy* the message onto an initialized struct? As it happens, a painful lesson we learned very early on, is that when the *C* programming language creates structs it organizes the information in them sequentially, as requested by the user, but perhaps not in a spatially intuitive manner. In order to efficiently access the stored information, it spaces the data in memory such that it will not have to access mixed elements within a 4 byte range. To give an example, if we request:

```

struct ha_msg
{
    char month;
    char day;
    short year;
    char hours;
    char minutes;
    char seconds;
    unsigned int frac;
    int lat;
    int lon;
    int height;
    int msl;
}

int gps_receiver_get_info_Ha(struct ha_msg *p, unsigned char *msg)
{
    p->month=msg[0];
    p->day=msg[1];
    p->year=msg[2]<<8 | msg[3];
    p->hours=msg[4];
    p->minutes=msg[5];
    p->seconds=msg[6];
    p->frac=(msg[7]<<24) | (msg[8]<<16) | (msg[9]<<8) | msg[10];
    p->lat=(msg[11]<<24) | (msg[12]<<16) | (msg[13]<<8) | msg[14];
    p->lon=(msg[15]<<24) | (msg[16]<<16) | (msg[17]<<8) | msg[18];
    p->height=(msg[19]<<24) | (msg[20]<<16) | (msg[21]<<8) | msg[22];
    p->msl=(msg[23]<<24) | (msg[24]<<16) | (msg[25]<<8) | msg[26];
}

```

Figure 4.23: Left: This is the code snippet where the struct for an @@Ha message is contained, note the parity in field with Figure 4.11. Right: here we have the corresponding packing function, which takes the contiguous data in the serialized Motorola Binary message, and maps it to the correct fields in the struct for easy data access.

```

struct identification

int bignumber1;

char middleinit;

int bignumber2;  };

```

and we take a look at the memory structure, upon accessing the 0th byte through the 3rd, we will get parts of *bignumber1*, and if we access the 4th byte we will get *middleinit*, but if we access the 5th byte, we will get 0. This is because C has aligned *bignumber2* such that it need not access mixed types within each 4 byte block. This pattern continues if we were to change *middleinit* to three 2 byte short fields such that now there will be 8 bytes in between *bignumber1* and *bignumber2*, but only 6 of them will be filled. Hence, we really need a map from the scenario where data is serialized continuously, to that where the data is packed in an easily accessible struct. This is the purpose of the packing function.

An important detail is the handling of repeated fields in a message; an example is given in Figure 4.24. The repeated data types are instantiated in their own struct, which becomes a substructure of the greater message. These are packed by a *for* loop, with the offset from the beginning of the message noted (an *int* was used for convenience; it is now a *const*).

|                               |                     |                              |
|-------------------------------|---------------------|------------------------------|
| Channel Data                  |                     |                              |
| i = SVID                      | 0...37              |                              |
| m = mode                      | 0...8               |                              |
| where:                        |                     |                              |
|                               | 0 = Code Search     | 5 = Message Sync Detect      |
|                               | 1 = Code Acquire    | 6 = Satellite Time Available |
|                               | 2 = AGC Set         | 7 = Ephemeris Acquire        |
|                               | 3 = Freq Acquire    | 8 = Available for Position   |
|                               | 4 = Bit Sync Detect |                              |
| s = signal strength           | 0...255             |                              |
| I = IODE                      | 0...255             |                              |
| dd = channel status (16 bits) |                     |                              |

```

struct ha_sat
{
    char satid;
    char trakmode;
    unsigned char sigstrength;
    unsigned char iode;
    short chanstat;
};

int i;
int offset=52;
for(i=0;i<12;i++){
    p->sats[i].satid=msg[offset+1+6*i];
    p->sats[i].trakmode=msg[offset+2+6*i];
    p->sats[i].sigstrength=msg[offset+3+6*i];
    p->sats[i].iode=msg[offset+4+6*i];
    p->sats[i].chanstat==(msg[offset+5+6*i]<<8) | msg[offset+6+6*i];
}

```

Figure 4.24: Top: the manual entry from **m12muser** for the repeated field of channel-wise satellite data. Bottom Left: the struct corresponding to one channel. Bottom Right: the packing function which iterates over the contiguous data, putting it into the channel structs.

Taking a look at Figure 4.23, we can see in the right panel, the packing function corresponding to the struct in the left panel and the message in the User's Manual shown in Figure 4.11. Each of the allowed messages has such a corresponding struct created in the same way: (1) examine the manual (**m12muser**), (2) create a struct corresponding to the data in the message, (3) write a packing function for the message, (4) add the message header to *gps\_receiver\_process* and finally (5) add the packing function and any relevant readout to the *while* loop in *gps\_main* (under the select block). The messages implemented currently are likely the only ones which will be relevant for future use, and are as follows:

| Message Header | Content Summary                                                |
|----------------|----------------------------------------------------------------|
| @@Ha           | Time/Position/Receiver Status, Detailed                        |
| @@Hn           | T-RAIM + Sawtooth Information                                  |
| @@Hb           | Short Time/Position                                            |
| @@Bb           | Satellite Statuses and Information                             |
| @@As           | Position Hold Parameters                                       |
| @@Cj           | Receiver Identification Message (all ASCII, no packing needed) |

Table 4.5: Above are the message types implemented in the finalized version of the UUB GPS code.

## 4.7 Board Operating System and Bring-Up

To accomplish the goal of finding the composition of primaries through the SD, we need to add new detectors to each station. Handling the data from these new detectors will require a new board, the UUB, and that board requires an operating system. In this section, we will explain how that operating system was set up for the UUB, a project which the author headed in the beginning phases. This section is structured to first give a brief overview of the boot stages and PetaLinux, and then to give the documentation we created for compiling and booting PetaLinux, and finally we will give a “crash course” on how programs are added to the PetaLinux compilation.

### 4.7.1 Boot Stages

Four stages are required for the UUB to start up. They are listed and defined below:

1. Bootstrapper: Initializes the Zynq<sup>TM</sup> and is hardware specific but can be made trivially from a board support package.
2. First stage bootloader: Provides resources for second stage bootloader. This is made for Zynq<sup>TM</sup> chips in general, not for a specific board.
3. Second stage bootloader: The most commonly used solution and the one we will employ here is a program called U-Boot, which is available preconfigured for Zynq. U-Boot is effectively a stripped down Linux kernel which gives a large enough command set to do some debugging and launch the main kernel.
4. Operating system: Here we need a true Linux operating system, like PetaLinux or Ubuntu. It will be configured using a device tree which is available for the UUB

through the Xilinx Design Tools. It will be cross compiled on an external Linux platform using the provided Xilinx toolchain (Code Sourcery).

### 4.7.2 PetaLinux

To fulfill our operating system needs, the UUB will make use of the Xilinx recommended PetaLinux operating system. PetaLinux is easily compiled for ARM systems and gives all of the needed functionality in a package smaller than 10MB, although this can be expanded for greater functionality. The PetaLinux SDK has a number of useful options for all manners of communication, including a lightweight SSH server, drivers for many commonly used peripherals, and an open source configuration, allowing even kernel level modifications. Ultimately, however, the decision to use PetaLinux over any other operating system was made because it is integrated into the Xilinx ecosystem, making development much easier.

### 4.7.3 Bring-up Documentation

```
INFO: Checking component...
INFO: Generating make files and build linux
INFO: Generating make files for the subcomponents of linux
INFO: Building linux
[INFO ] pre-build linux/rootfs/fwupgrade
[INFO ] pre-build linux/rootfs/peekpoke
[INFO ] pre-build linux/rootfs/uWeb
[INFO ] build system.dtb
[INFO ] build linux/kernel
[INFO ] update linux/u-boot source
[INFO ] generate linux/u-boot configuration files
[INFO ] build linux/u-boot
[INFO ] Setting up stage config
[INFO ] Setting up rootfs config
```

Figure 4.25: This is the screen you are greeted with upon running *petalinux-config* in your project directory. From here you can configure the operating system with whatever options you desire.

Better than simply explaining the work we completed for the board bring-up, we can give here the final result, which was the documentation displayed in the following sections. This documentation outlines a method to bring-up PetaLinux for an arbitrary hardware image, given that it satisfies the requirements in Section 4.7.4.3. Start to finish it takes about 6 hours, only about an hour of which is foreground tasks; the majority of the time is spent downloading, installing and compiling. The document has been written for someone who is fairly competent with the Vivado™ design suite, and vaguely familiar with Linux. If you need help with the design suite you should try the first few Zynq™ hardware and software tutorials, after which you can send the author your questions via email if needed. If you are not familiar with how to use a Bash terminal or the intricacies of Linux, there are a plethora of online resources and Stack Exchange is usually helpful in this way.

## 4.7.4 Installing the Environment

### 4.7.4.1 Necessary Downloads

- VMware Player (*75MB*)

Found on the VMware websites downloads, free to use for non-commercial purposes

- CentOS 6.6 i386 (*3.6GB*)

Download from “older versions” area, make sure you get i386 and NOT x86\_64

- Vivado™ and SDK 2014.4 for Linux (*4.9GB*)

Can be acquired off the Xilinx website under the Downloads section

- PetaLinux SDK (*1.2GB*)

Located in a different section of the Xilinx Downloads

- TeraTerm or equivalent serial terminal emulator

#### **4.7.4.2 Necessary Packages**

yum these packages after booting the VM and before installing anything else:

```
dos2unix iproute gawk gcc git gnutls-devel net-tools ncurses-devel tftp-server zlib-devel
flex bison libstdc++.i686 glibc.i686 libgcc.i686 libgomp.i686 ncurses-libs.i686 zlib.i686
redhat-lsb
```

#### **4.7.4.3 Necessary Prerequisites**

- .hdf Hardware Description File from Vivado<sup>TM</sup>

Found in the SDK directory in the Vivado<sup>TM</sup> project folder

##### **Design Requirements:**

Must have TTC enabled (Under MIO Configuration for the Zynq<sup>TM</sup> PS in Vivado<sup>TM</sup>)

Must have DDR enabled (Follow the first two Zynq Hardware tutorials that come with the board)

Should use PS UART (Also in first two Zynq HW tutorials)

- .bit Bitstream file from Vivado<sup>TM</sup>

Can be exported separately from Vivado<sup>TM</sup> 2014.4

#### **4.7.4.4 Installation Procedure**

1. Install VMware Player
2. Create a VM with the CentOS 6.6 image (must be 32-bit), you should a lot 40GB

At this point you will want to open a terminal in the VM and assume superuser privileges:

Either use the command “su” to log in as root

Or add yourself to the sudoers list and use “sudo” in the following steps as appropriate.

3. Install the packages listed in section 4.7.4.2.

run “yum install <package list>”

4. Unzip the .tar.gz file for Vivado™ and SDK.

5. Open the directory you unzipped the installer into and run ./xsetup with root privileges.

6. Follow the steps and be sure to install Vivado™, SDK and the cable drivers. You want Vivado™ Design Edition, and add SDK and Cable drivers. It will by default install to the correct location, /opt/Xilinx/.

7. Run the PetaLinux 2014.4 installer with root privileges (“./petalinux-v2014.4-final-installer.run /opt/pkg/” to install to /opt/pkg/. Be sure to “mkdir /opt/pkg/” before attempting install).

## 4.7.5 Configuring a Board for Boot Up

### 4.7.5.1 Setting up Working Environment

1. Change directories to the Vivado™ install directory (/opt/Xilinx/Vivado/2014.4/) and “source ./settings32.sh”
2. Change directories to the PetaLinux install directory and “source ./settings.sh”

Ignore the ‘no tftp’ error, it does not matter for our purposes.

#### 4.7.5.2 Configuring and Building PetaLinux

1. In a directory of your choice, run “`petalinux-create --type project --template zynq --name <name of project>`” where you’ve put in the name of your project, without spaces. It will create a new directory named after the project in the directory. The directory it creates is called the “project directory”
2. Place your Bitstream (.bit) and Hardware Description File (.hdf) in a subdirectory to your project’s main directory. We will call it “`./SUPPORT`” for the purposes of this documentation.
3. In that directory run “`petalinux-config --get-hw-description=./SUPPORT/`”
4. this should take a minute, at which point you can configure your kernel and hardware settings. Other configuration options are available and can be found in the PetaLinux command line reference([**commands**]). The default settings should be fine in most cases. Exit and save. It will take another minute to finish configuration.
5. Now we are ready to build, simply run “`petalinux-build`”
6. (OPTIONAL) You can package the project if you plan to boot it off an SD card or flash memory. You will need to reference the bitstream to do this.  
Run “`petalinux-package --boot --fsbl ./images/linux/zynq_fsbl.elf --fpga <bitstream file> --u-boot`”.

#### 4.7.5.3 Booting Your Build

In this documentation we will only cover booting off of JTAG. Documentation for booting off of the SD card can be found in the Reference Guide [**reference**].

It is also worth noting that VMware player handles whether your physical computer or your virtual computer has which component of the UUB plugged into it. That is, if you want to run TeraTerm in Windows and run the PetaLinux console on your VM you can do so by connecting the appropriate devices (Cypress USB UART is the standard I/O, the JTAG is the future devices Digilent USB) in the top right panel of the VM.

1. Plug in all the required cable for your board. Be sure to have an output to a terminal.

Start the Terminal emulator such as TeraTerm after powering on the board.

2. Run these commands in order, pausing between each:

```
petalinux-boot --jtag --fpga --bitstream <bitstream file (.bit)>  
  
petalinux-boot --jtag --fsbl ./images/linux/zynq_fsbl.elf --extra-xmd "init_user"  
  
petalinux-boot --jtag --u-boot  
  
petalinux-boot --jtag --kernel
```

3. You should now see the terminal emulator starting to show the output of the board booting Linux. Login with username: “root” password: “root”.

#### 4.7.5.4 Adding Programs

To add programs to the PetaLinux kernel, we need only run this command in the project directory:

```
petalinux-create -t apps -n <program name> --enable
```

where we replace <program name> with whatever we are naming the application. For porting programs from the UUB, this was done for each component program, and then they were debugged on an actual UUB (not an evaluation board). With this workflow, your

program will be compiled every time you compile PetaLinux, although you can use the petalinux-build command to build individual applications.

## **4.8 Conclusions: Reliability of the Integrated GPS and Operating System**

At time of writing, the PetaLinux operating system and the GPS software and firmware have been running in the Auger Engineering Array (EA) for more than two years. Beyond having no reported issues with either system, we have not received any request for further functionality. From the lack of issues, we can say that the integration and board bring-up were successful.

# Chapter 5

## Timing for AugerPrime

In this chapter, we will investigate the timing performance of two aspects of the Auger-Prime upgrade. To set the stage, we will have a discussion of some timing basics. After this, we move on to the topic of GPS receiver selection for AugerPrime. We motivate this endeavor, discussing the need for a new receiver selection due to availability, and then talk about the absolute, relative and temperature timing tests done for the M12M and SSR-6Tf receiver models. To conclude this section, we justify our recommendation of the SSR-6Tf, which has been accepted and acted on by the collaboration.

The measurement for the time resolution of the upgraded SD stations is then outlined. We give our methods for two different measurements of the time resolution, one of which requires us to calculate a physical error factor for arrive directions of triggering showers, and the other which involves using a cable to time synchronization pulses between two detectors. We find the time resolution to be  $\sim 8\text{ns}$  and that both methods are in agreement.

## 5.1 Timing Basics

In order to keep time in any context, we must first start with a clock. Even back to the days of Galileo or John Harrison, clocks have been based on the simple principle of an oscillator, keeping time by ‘ticking’ at a precise rate. In today’s context, most clocks are kept by electronic oscillators with a variety of methods employed to generate the signal. In the case of the UUB, this is an Abracon ABLJO-V-120.000MHZ-T2, which uses the 3rd overtone of a quartz crystal to keep time. By modulating the voltage across this VCXO, we can control its vibrational frequency.

The field of air shower physics usually requires its time-tagging hardware to accomplish two tasks. First, it must time the occurrence of events, which can be broadly described in terms of timing the rising edge of a logic pulse. Generally, the main task of trigger hardware and firmware is the generation of a logic pulse which corresponds to the time of the trigger, often offset by some known amount. The second task of a time-tagging module in an experiment such as Auger, is to accomplish frequency distribution, which is technical jargon for synchronizing clocks at a very fine level.

For frequency distribution, Auger uses the GPS constellation’s space-borne bank of atomic clocks. Using the techniques outlined in subsection 4.5.1, the GPS receiver can calculate the time of a GPS second to a handful nanosecond’s accuracy. These GPS seconds are timed by the receiver’s 1PPS, which comes in the form of a logic pulse. By counting the ticks of an oscillator from when a 1PPS comes in to when an event occurs, and then from the event to the next 1PPS, we can interpolate to find the point in the second when the event occurred. This process is encapsulated in Equation 5.1 below.

$$t_e = \frac{N_e}{N_{i+1,PPS} - N_{i,PPS}}. \quad (5.1)$$

Here, we have  $N_e$ , the number of counts up to an event,  $N_{i,PPS}$  the counter value at the preceding PPS, and  $N_{i+1,PPS}$ , the counter at the current (i.e. subsequent) PPS. This gives us  $t_e$ , the time (in seconds) of an event within a GPS second, and then we can simply use the telemetry messages from the receiver or the counter from the time tagging module to put together the GPS second. There is some difficulty in finding the number of leap seconds to date, but this can be requested from the receiver periodically. With these pieces of information, we can put together a full decimal time-tag of when an event occurs, allowing reconstruction of air showers and comparison with data from other experiments.

To get the highest possible accuracy in time tagging, we will need to apply the clock granularity message. This has two effects on Equation 5.1: first, the clock granularity messages (subsubsection 4.5.3.2) from the proceeding and current PPS events give the information that we need to adjust how many seconds the number of calibration counts,  $N_{cal} = N_{i+1,PPS} - N_{i,PPS}$ , represents. The preceding sawtooth message then tells us how the end of second represented by the PPS needs to be shifted relative to GPS time to be correctly aligned.

We address the first problem by applying a correction to  $N_{cal}$ :

$$N'_{cal} = (N_{i+1,PPS} - N_{i,PPS}) \left( 1 + 10^{-9}(\Delta t_{curr} - \Delta t_{last}) \right),$$

where the  $\Delta t_i$  correspond to the preceding and current saw tooth (*last* and *curr*, respectively) in nanoseconds, as it is delivered by the receiver. The second problem is fixed by simply adding the preceding sawtooth ( $\Delta t_{last}$ ) to the event time. This gives us a new  $t'_e$  of:

$$t'_e = \frac{N_e}{(N_{i+1,PPS} - N_{i,PPS}) \left( 1 + 10^{-9}(\Delta t_{curr} - \Delta t_{last}) \right)} * 10^9 + \Delta t_{last}. \quad (5.2)$$

In this formulation, we have added the factor of  $10^9$  to put  $t'_e$  into units of nanoseconds,

which is the usual convention for the type of timing accuracy measurements we will be doing in this work. This is in line with the framework set forth by the original group who worked on GPS verification for the initial Auger construction (documented in **firsttag**).

For final science purposes, we usually want to give event times in number of GPS seconds, including the lengthy portion of the event time which will come after the decimal. While none of the measurements and activity documented in this chapter made use of it, a good place for help determining the number of leap seconds, and incidentally the conversion between UTC and GPS time, is the LIGO webpage on the topic, <https://www.gw-openscience.org/gps/>. The discussion on timing standards and accuracy is continued in section 6.1.

## 5.2 GPS Receiver Selection Testing

In order to operate an observatory like Auger, precise timing synchronization is needed amongst the SD stations. This is accomplished via GPS timing boards. While the original receiver for the AugerPrime upgrade, the i-Lotus M12M [**m12mspec**], was initially chosen before the author began working with the Auger Collaboration, the manufacturer was unable to meet the volume of our request when the time came to deliver. Accordingly, we had to test a new GPS unit, the Synergy SSR-6Tf. During testing, the manufacturer came back and informed us that they had found the needed parts. At this time we had made significant progress towards vetting the new receiver.

For this section, we will first discuss the availability situation as mentioned above in detail. We then move on to a brief discussion of the two models of receivers. After this, the first test, for absolute timing, is outlined and then results are shown. We then discuss the method for the relative timing and temperate dependence tests, and show their results.

Finally, we make the recommendation to purchase the SSR-6Tf and justify it from the data and context.

### **5.2.1 Concerns regarding the availability of the i-Lotus M12M**

i-Lotus is a company in Singapore that received technology from Motorola who no longer manufactures GPS receivers (theirs were the units used for the UB). The i-Lotus M12M [**m12mspec**] has been long anticipated as the selected GPS receiver for the UUB. Currently, all but two UUBs within the Engineering Array are operating with i-Lotus M12M GPS receivers.

In late 2018, a purchase order was placed to obtain all M12Ms required for the Auger upgrade. The order was placed via Synergy Systems LLC, the licensed reseller for i-Lotus in the US. The hope and expectation was that i-Lotus would accept the order and deliver the required M12M units to CWRU where our group would begin to validate and calibrate the units for eventual deployment into the upgraded array.

However, after the purchase order was placed, we were informed that i-Lotus would have substantial delays of at least several months in responding to our order due to an unavailability of component parts. By late 2017, our options were (1) to wait at least several months with the expectation that the parts shortage might improve, or (2) to consider an alternative to the i-Lotus M12M called the Synergy SSR-6Tf [**ssr**].

Since this time, we have been working with representatives from Synergy Systems to consider our options and to explore and validate the performance of the SSR-6Tf as a possible alternative to the M12M. During the intervening months, while testing the SSR-6Tf, we received updates that indicate that i-Lotus may now have components to fill some or potentially all of our order for the M12Ms. At this time, the group at CWRU has completed bench tests and temperature dependence tests to compare the performance and reliability of the M12M and the SSR-6Tf. Additionally, two SSR-6Tfs are installed in the field (in

Trak Jr. and Clais Jr.) and have been operating for over 6 months without issue.

### **5.2.2 Specifications: i-Lotus M12M vs. the SSR-6Tf**

The SSR-6Tf is a GPS receiver made specifically for accurate timing applications. The unit is made by Synergy Systems (the same company that acts as the reseller for i-Lotus in the US). The SSR-6Tf is designed to optionally operate in “compatibility mode” that provides nearly identical functionality to that of the i-Lotus M12M with comparable or better timing performance. In compatibility mode, the SSR-6Tf is designed to function as a “drop in replacement” for the M12M.

Both units specify operating temperatures in the range -40C to 85C. The i-Lotus draws 123 mWatts power while the SSR-6Tf draws 155 mWatts. Both receivers have the same form factor, pinout and antenna connectors.

By definition, timing accuracy here relates to the accuracy of timing for 1 Pulse-Per-Second (1PPS) output pulses. For every measurement reported here, we apply granularity corrections (so-called ‘sawtooth’, as explained in subsubsection 4.5.3.2) which are reported for each 1PPS pulse via internal serial line from the GPS receiver.

### **5.2.3 Initial Bench Tests: Absolute GPS Timing**

Our first tests were conducted on the bench. GPS antenna signals were routed into the laboratory from a roof-top antenna. Both the SSR-6Tf and the M12M have been exercised for many 100’s of hour in our laboratory without fault.

We find that the SSR-6Tf as delivered by Synergy cold starts into Motorola compatibility mode. For both cold and warm starts, the SSR-6Tf is much faster than the M12M (usually seconds vs. minutes).

### 5.2.3.1 Test Stand for Absolute GPS Timing

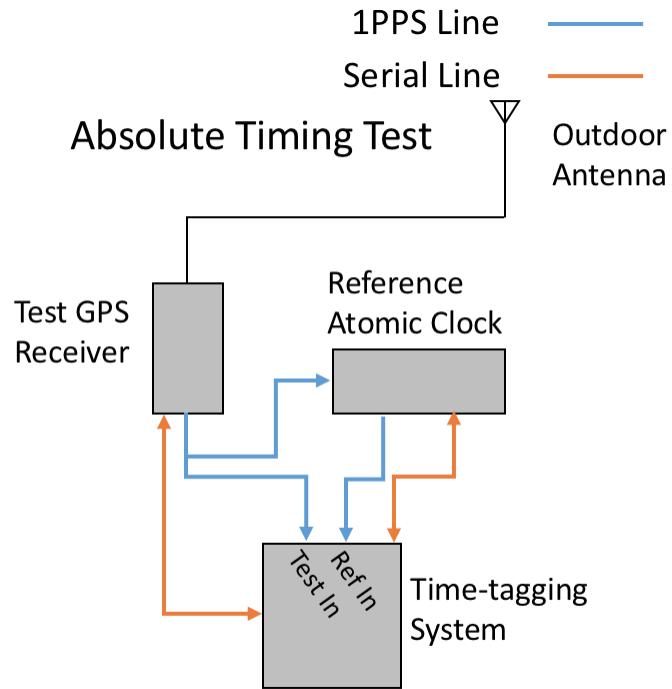


Figure 5.1: Bench test schematic of time-tagging system to measure absolute timing of the Synergy Systems SSR-6Tf vs. the i-Lotus M12M GPS receivers.

For our initial tests, we compare absolute time-tagging performance of the SSR-6Tf against that of the M12M over a range of time-scales using a GPS-disciplined Atomic Clock. Before each test, the atomic clock is trained for several hours to the GPS constellation's timing. The clock we are using is a FS725 Rubidium frequency standard from Stanford Research Systems, which is trained (disciplined) by the GPS constellation. Individual 1PPS produced by the disciplined FS725 provide an accurate time standard to better than a few 100s of picoseconds over timescales from seconds to days.

Figure 5.1 shows a schematic diagram for the absolute timing test stand. Our bench test timing test stand, TIM (see section 6.2 for more details), is based on a ZedBoard™ and runs our own 750 Mhz time-tagging system firmware through a GPIO interface. Operation of the board is controlled by a Standalone linux operating system script. This script controls

the time tagging system firmware, data logging, and serial communications with both the GPS and the atomic clock. On a pulse-by-pulse basis we measure the timing differences between the 1PPS from the atomic clock and the corrected 1PPS from each receiver. The final result will be a plot of the standard deviation of the pulse arrival times as a function of timing window scanned over to calculate the variance. The derivative of this is directly related to the Allen Deviation.

### 5.2.3.2 Results for Absolute GPS Timing

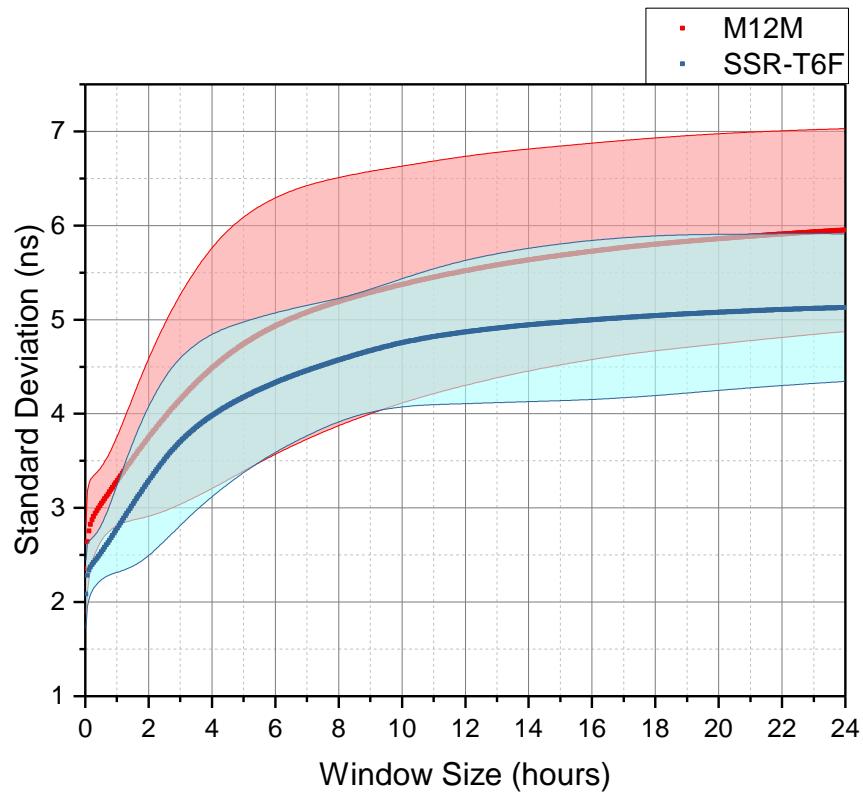


Figure 5.2: Absolute time-tagging: The standard deviation of each receiver's 1PPS compared to the 1PPS of the FS725 atomic clock. The i-Lotus M12M is shown in red. The Synergy Systems SSR-6Tf is shown in blue. The shaded areas correspond to a 1- $\sigma$  error region for each receiver.

Figure 5.2 shows our results for absolute time-tagging of sawtooth-corrected 1PPS GPS timing for the M12M vs. the SSR-6Tf. We show the RMS difference between the measured 1PPS for each receiver vs. the atomic clock standard as measured on timescales ranging from a few seconds to 24 hours.

In terms of short-term timing accuracy (timescales less than a few minutes), the SSR-6Tf reports a timing precision of 2.3 ns while the M12M reports a timing precision of 2.8 ns. Absolute timing errors gradually increase for both receivers over timescales of a few hours, presumably due to drifts in the electron content of the ionosphere. Over many hours, the long-term absolute timing resolution of the SSR-6Tf is generally better than about 5 nanoseconds while the M12M is closer to 6 nanoseconds. We find that on any time-scale from seconds to hours, the SSR-6Tf outperforms the M12M by approximately one nanosecond for absolute timing accuracy. We note that in terms of the performance specifications required for the AugerPrime upgrade (e.g., better than 8 nanoseconds absolute timing) both the SSR-6Tf and the i-Lotus M12M meet the required specification.

#### 5.2.4 Relative GPS Time-Tagging

Although absolute timing accuracy is an important performance parameter for GPS timing, in the field at Auger the *relative timing* between two receivers is the more important quantity, since only timing differences between receivers will impact the reconstruction. To verify relative timing accuracy, we developed a second test stand configured to accept telemetry data from two GPS units and to compute the difference in arrival times of their (sawtooth corrected) 1PPS time signals. We also use this configuration to explore possible temperature dependence of the arrival time of the 1PPS on the order of nanoseconds.

### 5.2.4.1 Test Stand for Relative GPS Timing and Temperature Dependence

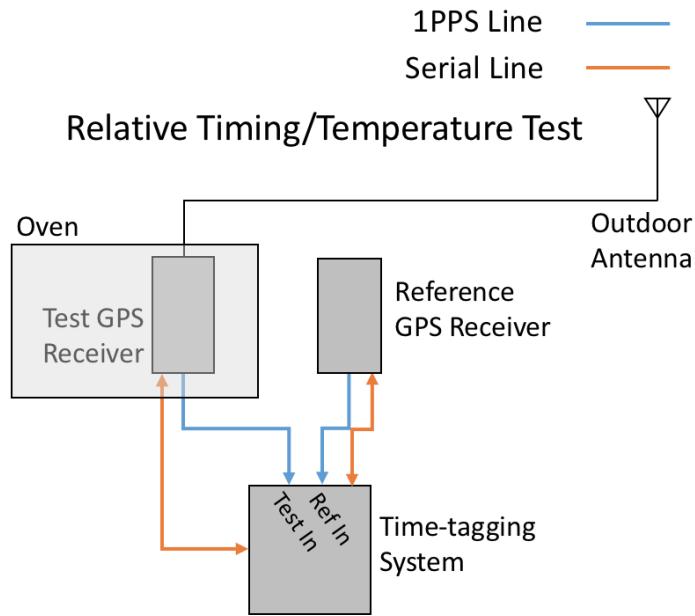


Figure 5.3: Bench test schematic of time-tagging system to measure relative timing of the Synergy Systems SSR-6Tf vs. the i-Lotus M12M GPS receivers, including temperature dependence.

Figure 5.3 shows the schematic setup of our test for relative timing and temperature dependence. The configuration is closely matched to that used for previous time-tagging calibration and temperature dependence measurements conducted and reported by the CWRU group [brandt]. For relative GPS time-tagging we select two GPS receivers of the same model and then measure the relative arrivals of their sawtooth-corrected 1PPSs. These measurements provide a series of time differences from which timing precision (standard deviation) can be computed over long-duration tests.

From previous results using a 250 MHz version of this system, the results of the temperature and relative timing tests are available in the linked document and will be cited below.

### 5.2.4.2 Temperature and Relative Timing Test

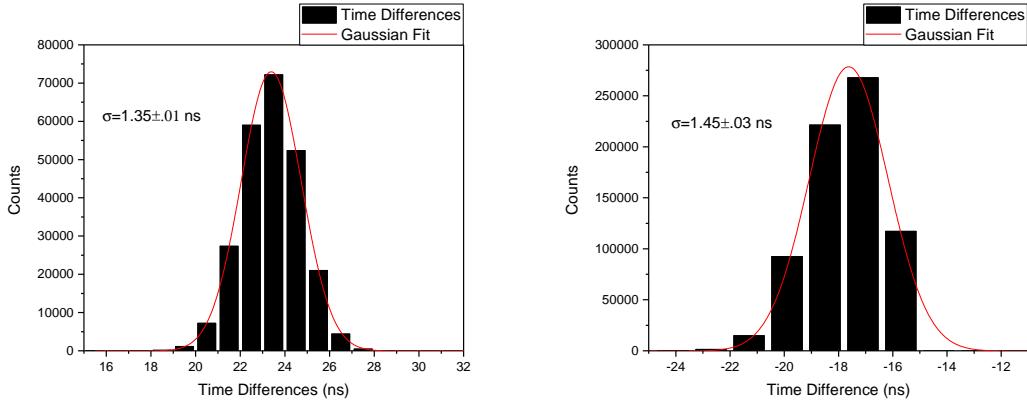


Figure 5.4: Left: the relative timing test for the SSR-6Tf. Right: the relative timing test for the M12M. These are the time differences of the sawtooth corrected

In comparison to the  $\sim 1.45\text{ns}$  standard deviation in the time differences of the M12Ms, the SSR has a standard deviation of 1.3 ns, which is the instrumental uncertainty of the 750 Mhz time-tagging system used for this testing [brandt]. It is worth noting that it's possible the SSR-6Tf has an even better timing accuracy, but it has reached the ability of our test bench to time-tag it. To reiterate, these time differences represent two identical receivers timed against each other with the TIM unit (section 6.2). The data have been filtered for counter rollover and sawtooth rollover outliers which account for less than 1% of the second by second time differences recorded. These measured timing precisions are in line with the manufacturer's presented figures [ssr, m12mspec].

For the temperature dependence testing, the profile of temperatures is derived from a study of the weather at the Auger site [brandt]. The M12M shows some temperature dependence during extreme temperature ramps, but the mean of its 1PPS arrival time does not depend on temperature, whereas SSR-6Tf does show direct dependence on the temperature. In spite of this, the performance of the SSR is still better than that of the M12M when put under thermal stress.

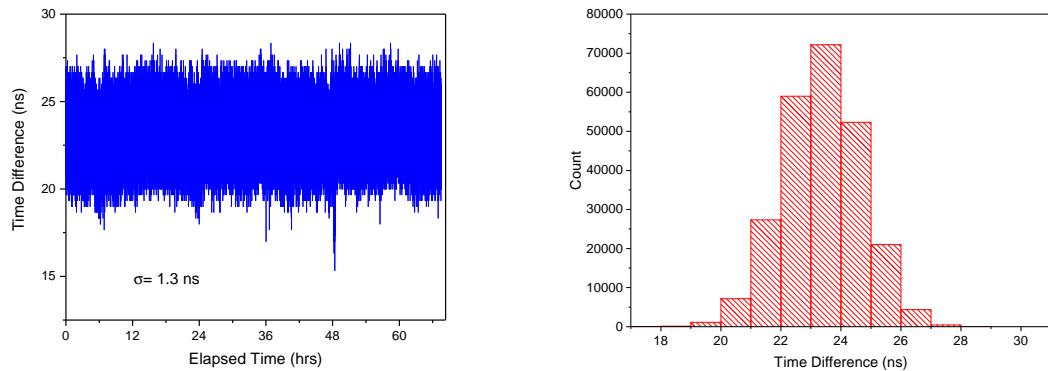


Figure 5.5: Left: A time series test of the SSR-6Tf's performance with sawtooth correction measured against another sawtooth corrected SSR-6Tf. The data has been corrected for counter and sawtooth rollover errors. Right: A histogram of the time series.

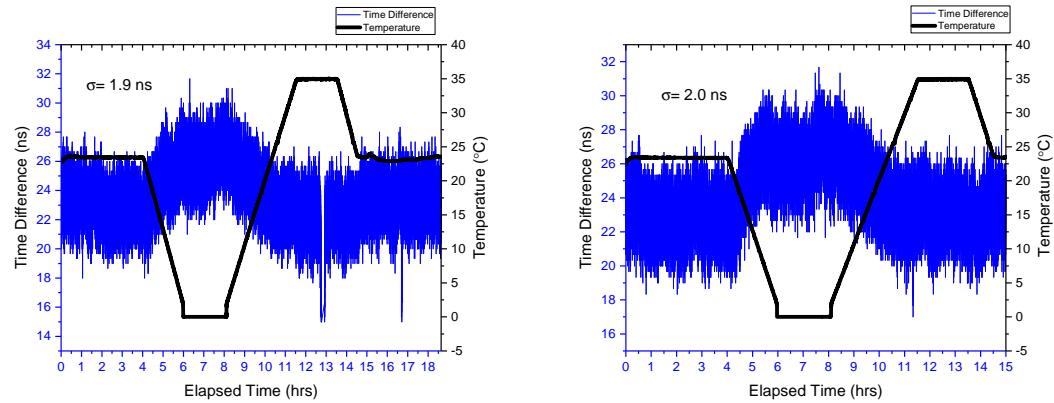


Figure 5.6: Two runs from the temperature testing of the SSR-6Tf.

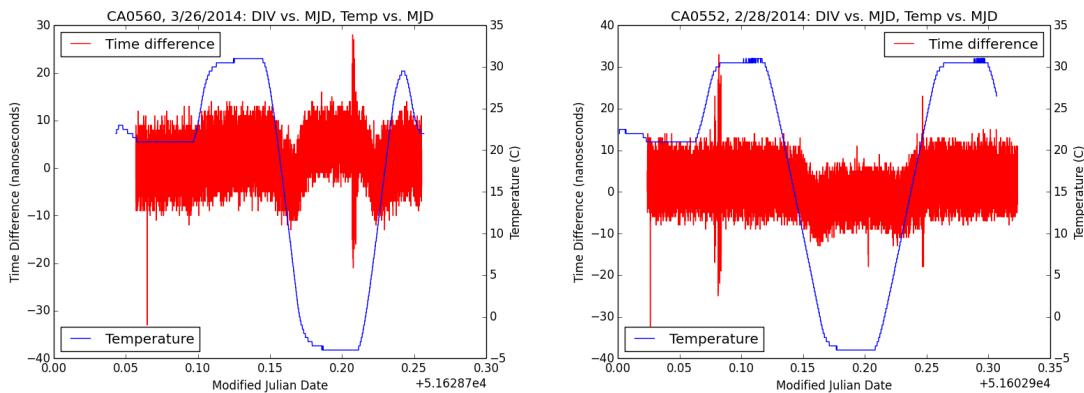


Figure 5.7: Plots from Dan Brandt's original analysis of the M12M's temperature dependence, available in **brandt**.

### 5.2.5 Conclusions

We have tested the M12M and SSR-6Tf receivers using two different methods, first looking at the absolute timing accuracy over different time scales and second looking at the timing of each receiver relative to another receiver of the same model under temperature variation. The absolute accuracy test shows that, while both models perform satisfactorily for use in AugerPrime, the SSR-6Tf has slightly better performance over all time scales. According to our relative timing tests, the SSR-6Tf outperforms the M12M; where the M12M has a standard deviation in time differences of  $\sim 1.45$  ns, the SSR has a relative timing accuracy of  $\sim 1.3$  ns or better. The temperature dependence testing showed that M12M receivers have a temperature dependence during large temperature changes but stabilize when the temperature does. On the other hand, the SSR-6Tf shows a direct temperature dependence where the mean of the time differences depends on the temperature. If the SSR was less accurate, this could cause issues, but even with this direct dependence the SSR is more accurate than the M12M.

All things considered, the SSR-6Tf is a newer receiver which outperforms the M12M in all relevant parameters. Pending testing in the field over the next few weeks, our recommendation for the Auger Prime upgrade is to purchase the SSR-6Tf receivers when Synergy is ready to deliver their final version.

## 5.3 AugerPrime Time-Resolution

An essential parameter of shower reconstruction for Auger is the time resolution of its detectors. This affects directional reconstruction amongst other observables, and if the time-resolution is low enough it may contribute to helping determine the muonic content of incident air showers (see **mupart** and section 4.2). With the integration of AugerPrime

completed, the time-tagging module has been added in a similar way to other IP (see section 4.6). At this point, it has been running continuously for over two years in the engineering array without issue, thereby confirming its reliability. This then leaves us to confirm the timing performance of the whole system, end-to-end.

In this section, we will first outline the basic facets of the time-tagging system. We will then move on to a discussion of each of two methods to measure the time resolution of the upgraded stations and the results of the measurements. First we will discuss a method using only air showers, and then a method using a synchronization cable. Finally, we will compare the results of these two tests and their uncertainties.

The timing tests will be performed in the stations Trak Jr., Clais Jr. and Peteroa Jr., which have been dubbed the ‘timing triplet’ and are some of the engineering array stations most frequently used for UUB verification and testing. A map of their positions is shown in Figure 5.8.

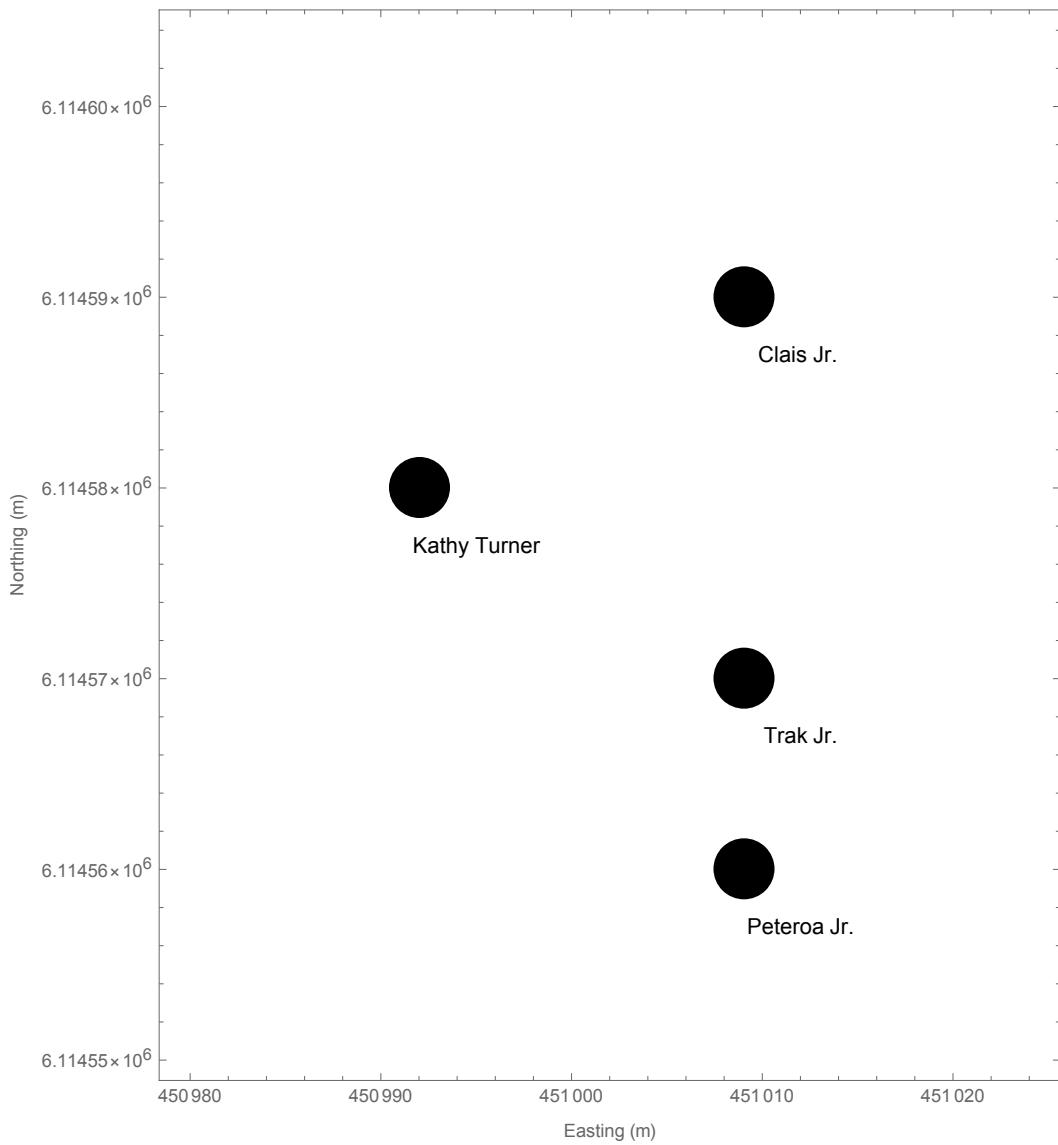


Figure 5.8: Here we have a map of the relative positions of the stations in the EA. The distance between Trak Jr. and Clais Jr. is  $\sim 20$ m and the distance between Trak Jr. and Peteroa Jr. is 11m (this is somewhat obscured since we have used the global Auger coordinate system on these axes).

### 5.3.1 Time-Tagging Specifications and Ports

In comparison to the time-tagging module for TIM, shown in Figure 6.4, the final time-tagging module for AugerPrime integrates all of the GPIO connections into internal wires and registers and makes some of the calculations that are done in post-processing for TIM.

This change is a result of the SDEU meeting held at Michigan Technological University in October of 2014, where a new specification for the system was written, eschewing the individual GPIO connections for a more efficient structure of internal connections and a single AXI connection to communicate with the PS. Through this connection, Direct Memory Addressing (DMA) is done to present the module's outputs directly into the memory of the processing system. This puts the burden of transferring data on the PL which will handle it constantly and without extra overhead. The PS then has direct access to this information when forming T3 packets and taking care of housekeeping tasks.

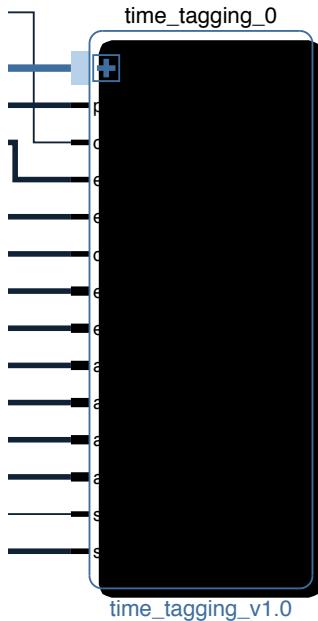


Figure 5.9: Here we have the block diagram for the time-tagging module in the UUB. This shows the connections to the AXI bus and the external connections to other PL or UUB locations.

The block diagram and AXI I/O of the time-tagging module are shown in Figure 5.9 and Table 5.1 respectively.

The block diagram inputs and outputs interface to other parts of the PL or UUB, while the AXI I/O provides the means to communicate the relevant information to the PS. There are effectively three types of timing going on in the module. The first, the ‘fast’ time tagging line is meant for showers, the ‘slow’ time-tagging line is meant for muons and there is a final line with a much simpler signal chain and register structure for the GPS calibration.

Each of the muon and shower lines have four registers fed by counter values from the 120Mhz clock and four registers (one tied to each of the 120Mhz clock registers) which contain the GPS second. These are multiplexed to decrease dead time, i.e. when one is full, an address register containing the next writable timing register's address is incremented, and then when a new event comes in we

| AXI REGISTER NAME | ADDRESS OFFSET | DESCRIPTION                                                          |
|-------------------|----------------|----------------------------------------------------------------------|
| Onanosec          | 0              | value of nanosecond (fast) counter at time of fast trigger           |
| Oseconds          | 1              | value of seconds counter at time of fast trigger                     |
| c120mout_sb       | 2              | value of nanosec. counter at last pps occurrence and fast trigger    |
| c120calout_sb     | 3              | value of calibration counter at last pps occurrence and fast trigger |
| slowtriggerns     | 4              | value of nanosecond counter at time of slow trigger                  |
| slowtriggersec    | 5              | value of seconds counter at time of slow trigger                     |
| c120mout_mb       | 6              | value of nanosec. counter at last pps occurrence and slow trigger    |
| c120calout_mb     | 7              | value of cal. counter at last pps occurrence and slow trigger        |
| timeseconds       | 8              | value of seconds counter at last pps occurrence when read by ps      |
| c120mout_ps       | 9              | value of nanosec counter at last pps occurrence when read by ps      |
| c120calout_ps     | 10             | value of cal counter at last pps occurrence when read by ps          |
| c120deadout       | 11             | value of dead counter at last pps occurrence when read by ps         |
| teststatus        | 12             | value of event status bits for test purposes only                    |
| ttagctrl          | 13             | control register contains time tag soft reset control bit            |
| ttagid            | 14             | reads back the binary ascii value of the letters "ttag"              |
| spare             | 15             | write and read any value                                             |

Table 5.1: These tables, reproduced from **bobttag**, show the register assignments for the AugerPrime Time-Tagging module.

repeat the process, writing the new events time-tag into the new register and again incrementing the address register appropriately. These, and all other AXI-connected registers are 32-bits wide.

### 5.3.2 Coincident Showers Method

When Auger was initially commissioned, the timing resolution of the stations was determined via a ‘coincident showers’ method. This method was not detailed in any publication, but its results are briefly discussed in **fdiam**. The basic method that they use is to calculate a Gaussian uncertainty quantifying the delay between stations from the arrival directions of low energy showers (the *physical dispersion* as it is sometimes referred to). They then take the coincident time-tags within a physically reasonable window, find their width by fitting a Gaussian to it, and then they subtract off the error from the arrival directions in quadrature.

We will use this same method to analyze coincident time-tags between Trak Jr. and Pteroia Jr. In general, we pick a month’s worth of data. We cut out time-tags outside of 150ns difference; this range allows us to ensure that there are not a large amount of unphysical outliers in the coincident showers. After this, we fit a Gaussian to the distribution and take its standard deviation as the uncorrected time resolution,  $\sigma_{tot}$ . We will evaluate the time

differences according to the formula shown in Equation 5.2.

To form the uncertainty from the time delay between stations, we can take the commonly found derivation of the distance from a point to a plane and use it to calculate the distance the shower plane must travel after it has hit one station to be seen by the other station. We can use spherical coordinates to describe the arrival direction of the shower and load this data directly from Auger's *Herald* data product. The distance the plane must travel as a function of local zenith angle  $\theta$  and azimuth  $\phi$  is:

$$d \sin \theta \sin \phi, \quad (5.3)$$

where  $d$  is the separation between the stations we are testing. To be clear,  $\theta$  is defined to be the zenith angle, i.e. 0 if the shower hits perpendicular to the atmosphere. With this established, we feed  $\theta$  and  $\phi$  values into Equation 5.3, make a histogram of the delays and then measure its width by fitting a Gaussian to it. This distribution will be referred to as the ‘distance-to-detector’ histogram. We then divide the fitted width by  $c$  and put the resulting time delay in units of nanoseconds. To find the final time-resolution, consider the following statement of the total timing error:

$$\sigma_{tot} = \sqrt{\sigma_{shower}^2 + \sigma_{det1}^2 + \sigma_{det2}^2} = \sqrt{\sigma_{shower}^2 + 2\sigma_{det}^2} \implies \sigma_{det} = \sqrt{\frac{\sigma_{tot}^2 - \sigma_{shower}^2}{2}}.$$

We assume only that the detectors involved in the timing analysis will have the same timing errors (they are identical). Care has been taken by the Observatory’s on-site scientists to ensure the firmware version is the same between the testing stations. At this point we have everything we need to calculate the time resolution using coincident showers.

### 5.3.2.1 Shower Selection for Arrival Directions

Ab initio, it may be advantageous for us to make some energy cuts on the showers we use to assess the error due to shower arrival directions. We make this statement here for clarity and transparency: any selection of events gives approximately the same calculated standard deviation of the distribution of time delays due to arrival directions. The directly calculated standard deviation for the selections of data corresponding to  $<1$  EeV,  $1$  EeV  $<E <20$  EeV,  $20$  EeV  $<E <40$  EeV and  $>40$  EeV all give standard deviations with .1 ns of 18.25 ns ( $18.28 \pm .08$  ns as a measurement). Cuts for low energy events give non-Gaussian distributions, with a high center peak and heavy tails on either end. Attempts to fit a Gaussian to these fail without excessive data massaging or additional degrees of freedom in the fit. Only the highest energy bin from above has a convergent fit.

Looking at this, we will proceed to use the fitted standard deviation for the highest energy bin, making the assumption that this is sufficient. There is some physical motivation to do this since high energy showers will be guaranteed to trigger both detectors if they are within a suitable distance, while even very close low energy showers may not cause a co-trigger. If we find that using this assumption delivers unphysical or disagreeing results with the synchronization cable method, we can revisit this assumption and look at methods of handling non-Gaussian noise.

### 5.3.3 Coincident Showers Results

Proceeding as outlined above, we find the distribution of distances-to-detector to be  $5.4 \pm 1.1$  m (error from fitting) with an adjusted  $r^2$  of .98, indicating an acceptable but not great fit. From this, we get a standard deviation of the time delays of  $18.1 \pm 3.5$  ns. The distribution and fit are shown in Figure 5.10.

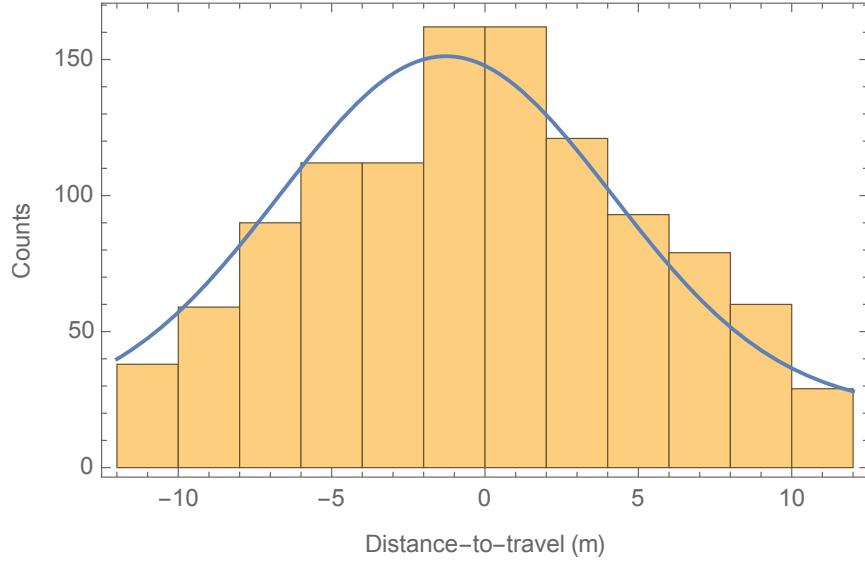


Figure 5.10: Shown here is a histogram and Gaussian fit of the distance-to-detector calculated based on Equation 5.3 and the energy cut discussed in subsubsection 5.3.2.1.

To find the distribution of time differences caused by coincident showers on the two stations, we apply the methods outlined above to data from January and February of 2018. These months are chosen arbitrarily. Their data are displayed in Figure 5.11.

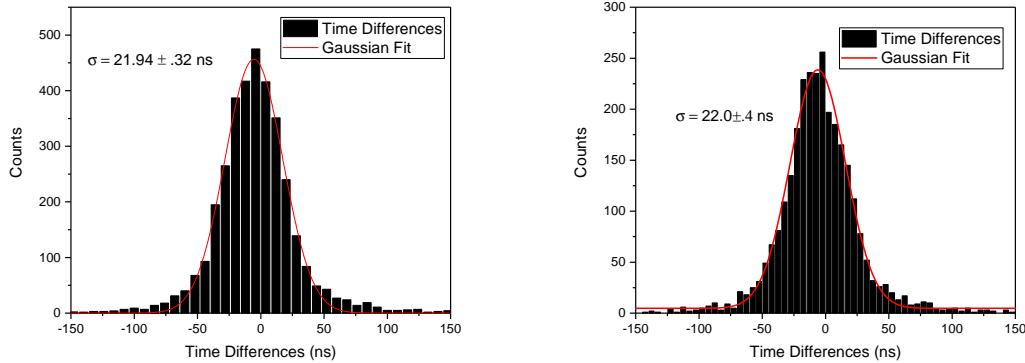


Figure 5.11: Here we have the distributions of time differences between Trak Jr. and Peteroa Jr. for January and February of 2018 each fitted with a Gaussian whose standard deviation is shown in the figure. Each distribution contains  $\sim 3600$  data points.

We can pick either month, their standard deviations are within uncertainties of each other. Averaging them and propagating uncertainties we get an uncorrected time resolution

of  $21.97 \pm .25$  ns. From here, we can calculate our final result with propagated uncertainties:

$$\sigma_{det} = \sqrt{\frac{21.97^2 - 18.08^2}{2}} = 8.8 \pm 3.6 \text{ ns.} \quad (5.4)$$

### 5.3.4 Synchronization Cable Method

Another way to measure the timing of the closely placed EA stations is to connect a cable to the trigger output of one station and bring it into one of the ADC channels of the other. This connection was made between Trak Jr. and Clais Jr. and stayed in place from early 2018 until August of that year, when the second version of the UUB was installed. The UUBv2 is currently being debugged and will soon be operational.

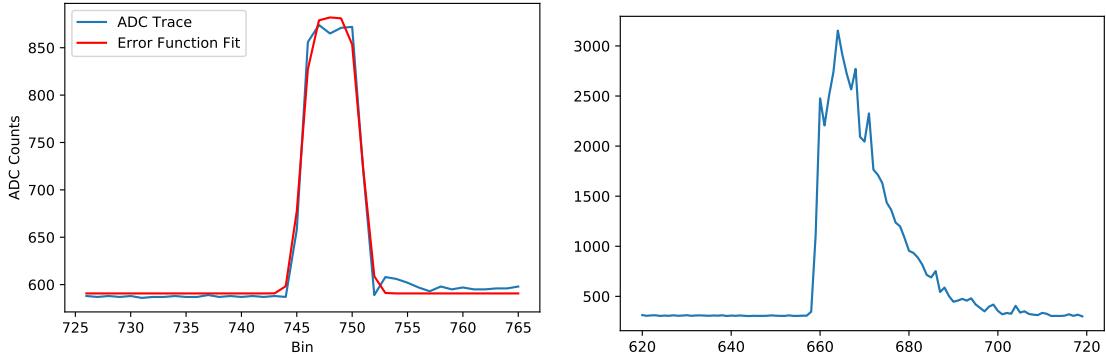


Figure 5.12: Here we have the distributions of time differences between Trak Jr. and Peteroa Jr. for January and February of 2018 each fitted with a Gaussian whose standard deviation is shown in the figure. Each distribution contains  $\sim 3600$  data points.

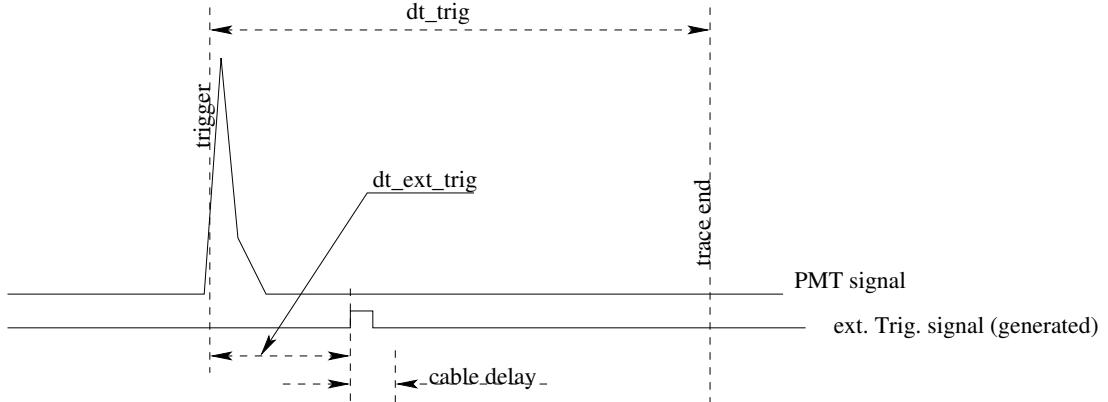
By timing the trigger in Clais Jr. and differencing it with the arrival time of the synchronization pulse (trigger out) in Trak Jr. we can test the end-to-end timing resolution of the detector without having to account for any physical dispersion from showers. It should also be kept in mind that we are looking for the width of a distribution, so fixed offsets in the time difference calculation can be disregarded. The trigger in Clais is fixed at 1/3 of the way through the trace in each trigger by the output of the ring buffer it is kept in. The

GPS time is latched by the time-tagging system at the end of the trace recording, and we will need this and sawtooth corrections for this calculation. The event time calculation will come from Equation 5.2. We will find the leading edge of the synchronization pulse by fitting an error function to it of the form:

$$f(x) = \frac{a}{2} (\operatorname{erf}(x - x_1) - \operatorname{erf}(x - x_2)) + y_0, \quad (5.5)$$

where  $x_1$  is the center of the rising edge,  $x_2$  is the center of the falling edge,  $a$  is the maximum of the pulse over the noise floor, and  $y_0$  is the noise floor. A sample fitting is shown in Figure 5.12. We enforce  $x_1 < x_2$  and use suitable automatically calculated initial values for our fitting. This fitting converges on almost all pulses it has been applied to. A sample collection of ADC recordings are shown in Figure 5.14.

## At Clais Station



## At Trak Station

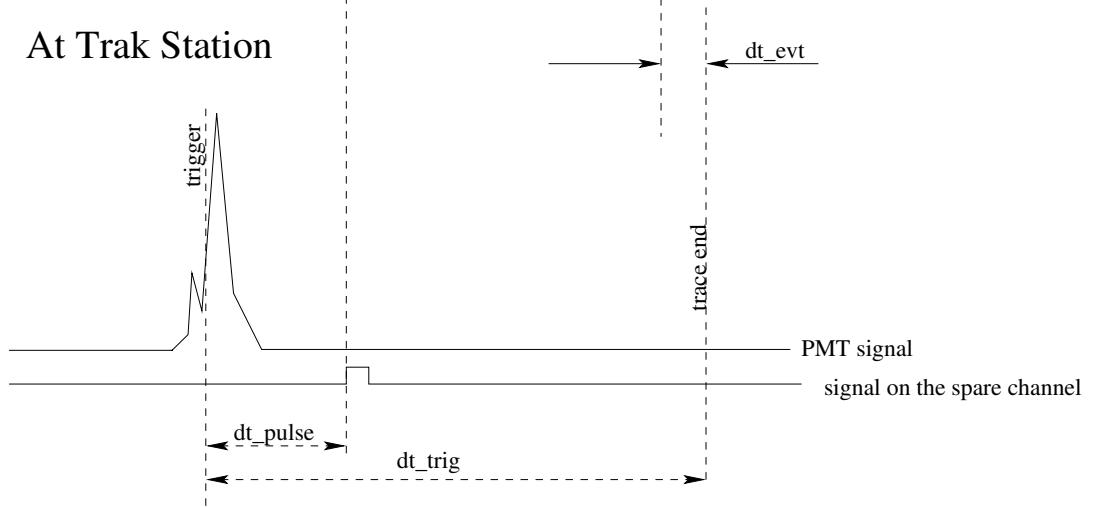


Figure 5.13: In this figure from Ricardo Sato, we have the time structure of the ADC traces in Trak Jr. and Clais Jr. This diagram serves to label the various time differences in the problem of finding the UUB's time resolution via synchronization cable.

If we look to Figure 5.13, we can see some of the time structure of the shower and Calibration pulses. Consider that the ADC from Clais Jr. ends at  $t_{Clais}$  and the ADC trace from Trak ends at  $t_{Trak}$  and these are both absolute GPS times. Then the time when Clais Jr. was triggered is  $t_{trig} = t_{Clais} - dt_{trig}$ . As mentioned above,  $dt_{trig}$  is a fixed quantity representing 1/3 of the way into the trace. As denoted in the diagram, the time that Trak Jr. will see this pulse is  $t_{sync} = t_{Trak} - dt_{trig} + dt_{pulse}$ . the times we will histogram are then:

$$t_{diff} = t_{sync} - t_{trig} = t_{Clais} - t_{Trak} + dt_{pulse}. \quad (5.6)$$

The total observed time differences from the stations result in a standard deviation of:

$$\sigma_{tot} = \sqrt{\sigma_{det1}^2 + \sigma_{det2}^2} = \sqrt{2\sigma_{det}^2} \implies \sigma_{det} = \sqrt{\frac{\sigma_{tot}}{2}},$$

where  $\sigma_{tot} = \sigma(t_{diff})$ . With this stated, we have our complete method of timing analysis using the synchronization cable.

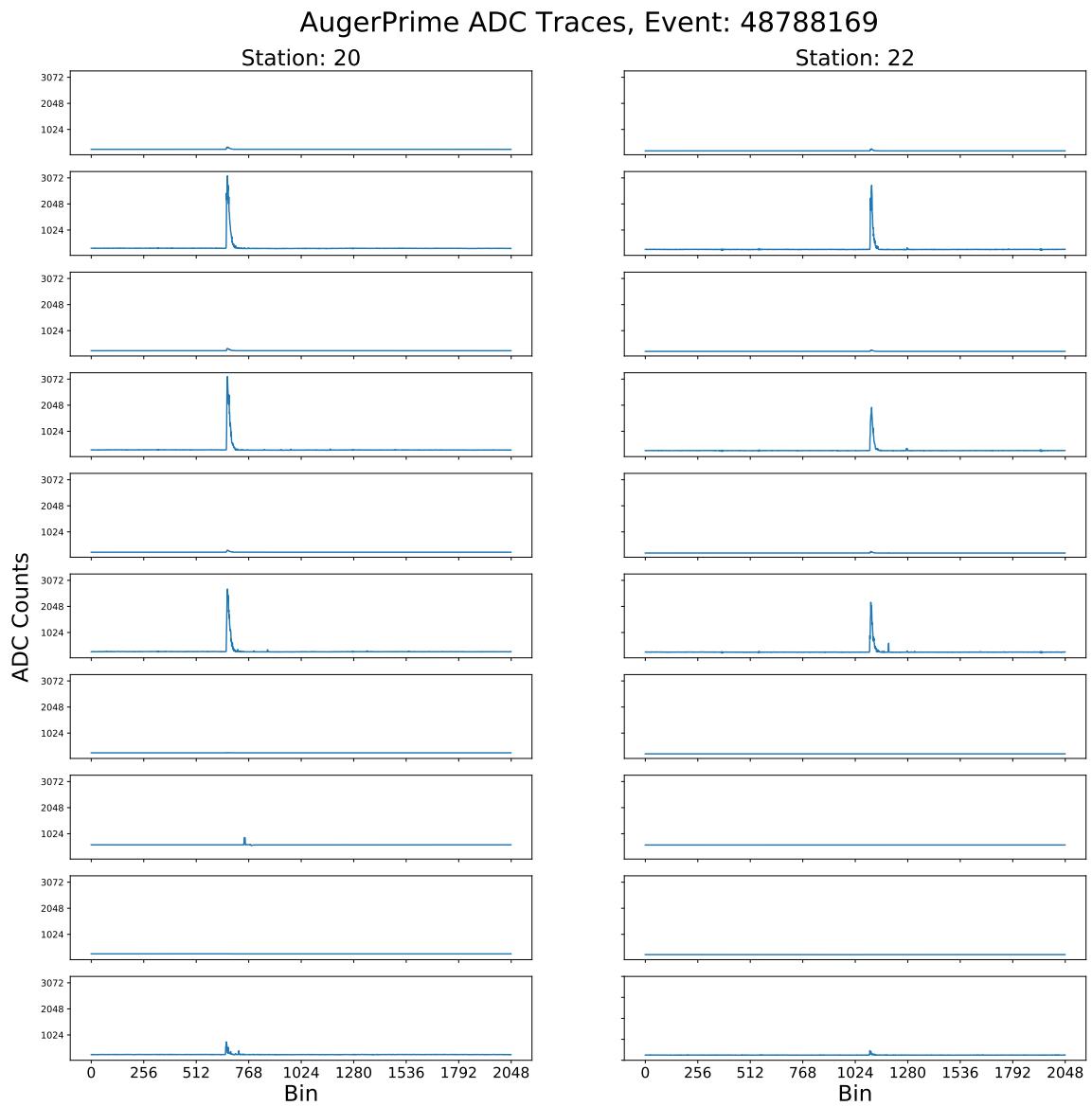


Figure 5.14: This panel of ADC traces corresponds to one air shower in the engineering array. We can see, for example in the high gain channel (2nd down on the left), the air shower wavefront and in the 8th pane on the left, we have the synchronization pulse.

### 5.3.5 Synchronization Cable Results

Taking data from the month of June 2018, we run our analysis as described in the previous section. The histogram and fit of time differences are shown in Figure 5.15, where we can see that the standard deviation of the fit Gaussian is  $\sigma_{tot} = 11.94 \pm .21\text{ns}$ , which gives us a final result:

$$\sigma_{det} = \frac{\sigma_{tot}}{\sqrt{2}} = 8.44 \pm .15 \text{ ns.} \quad (5.7)$$

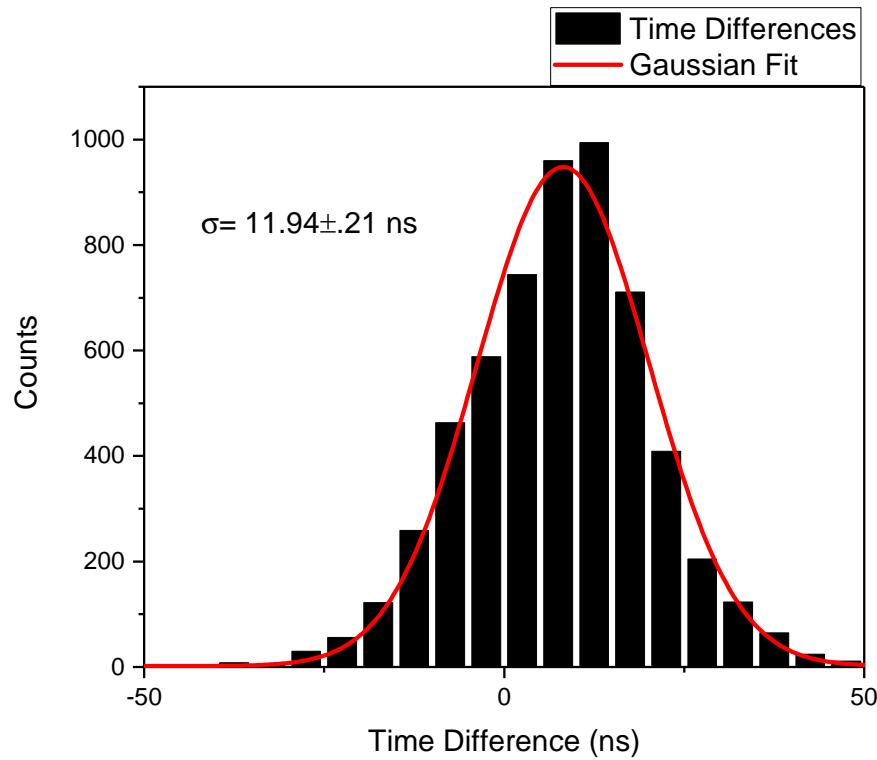


Figure 5.15: Here we have the final  $t_{diff}$  histogram with the standard deviation from the fit labeled on it.

### 5.3.6 Conclusions

In this section we discussed two different methods of getting the time resolution of the UUB. One of these, the ‘coincident showers’ method required us to calculate a factor rep-

resenting the physical dispersion in timing caused by the directionality of the air showers. After surveying different energy ranges and finding that the highest energy cuts provided a good Gaussian fit, while retaining the same calculated standard deviation of all other energy ranges investigated. We used the parameters from this fit to estimate a  $\sim$ 18ns dispersion factor which, when subtracted in quadrature from the measured  $\sim$ 22ns uncorrected time resolution, gave us a final measurement from the first method of  $\sigma_{det} = 8.8 \pm 3.6$  ns.

After this method, we discussed the ‘synchronization cable’ method, in which we time the arrival of a pulse from one station to the other. Since we need to calculate GPS time stamps at both stations, as well as track the arrival time using an ADC, this method provides a true end-to-end test of the timing resolution. Following this route, we arrive at a time resolution of  $\sigma_{det} = 8.44 \pm .15$  ns.

Given the large uncertainty induced from the calculation of the physical dispersion due to directionality of the arriving showers, we find that the ‘synchronization cable’ method is a better measurement of the time resolution. We therefore accept  $\sigma_{det} = 8.44 \pm .15$  ns as our final measurement, although the measurement using the ‘coincident showers’ method is in good agreement with this result.

# Chapter 6

## Timing Equipment and Analysis

Throughout this chapter, we will document a number of the HEA group timing efforts. Starting with a discussion of the hardware behind the Timing Instrumentation Module (TIM), we will walk through the various tasks that TIM was used for. We will go through the work we did on correlating timing errors of GPS as a function of spatial separation, the Auger@TA cross calibration and the new TIMPrime for the CTA pSCT (more context and motivation in subsection 3.3.4).

### 6.1 Stability and Accuracy of Timing Systems

Before proceeding into a discussion of the timing tasks in the HEA group, we first need an overview on the topic of the stability of frequency standards and the accuracy of time-tagging devices.

The accuracy of time-tagging standards is relatively straightforward. We usually refer to a time-tagging device's accuracy or resolution, which is essentially the width of the distribution of the device's measurement of an event time, versus the known event time. For example, we could use a pulse generator with a known frequency and see if we can

reconstruct the period corresponding to that frequency with our timing device. If we trust our pulse generator, we can use it to create a histogram of time differences between the known period and the measured one. The width of this distribution will give the time resolution. The minimum time resolution in a simple rising-edge-based time-tagging system is the coarse timing granularity given by the inverse of the time-tagging clock's frequency.

This can be a function of the period of the signal you try to calibrate off of, but for our purposes we will ignore this unless otherwise needed. For most of the measurements in this chapter, we will be using one second periods. This quantity can be called the “jitter” of a timing pulse if we reverse our measurement and look at how good the frequency standard is. Jitter, however, can be a somewhat obfuscated term and is usually used in electrical engineering to quantify how clean a fast periodic signal is (e.g. the distribution of derivatives of time differences between the rising edges of a 40Mhz square wave).

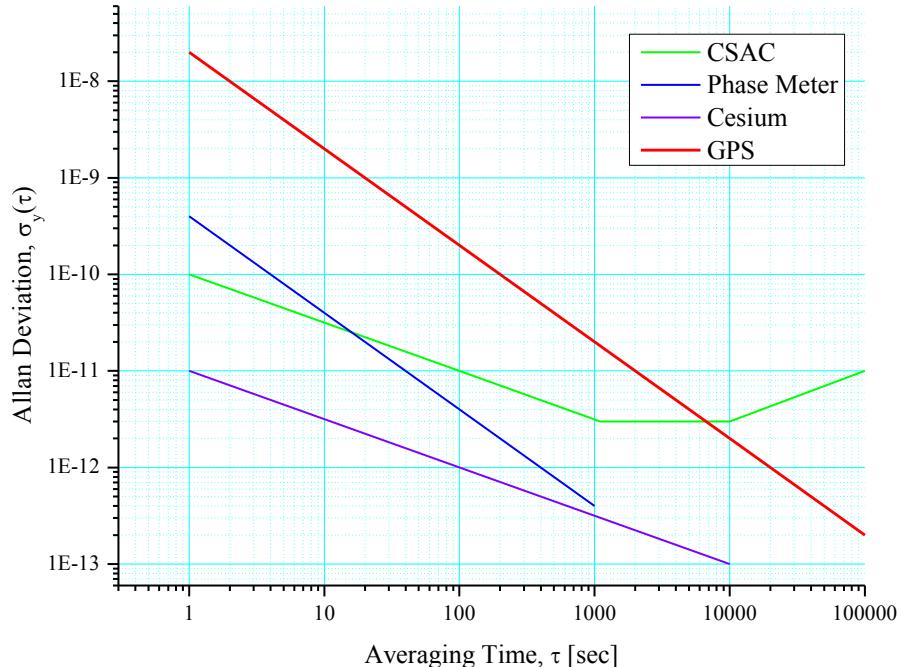


Figure 6.1: Allen Deviation of a number of sources from **esac**. This diagram shows the longevity of frequency standards, but does not reflect that even a cesium source will inflect up eventually if not disciplined to a bank of clocks, averaging out each other's fluctuations.

From a technical point of view, the correct quantity to look at for a frequency standard is the Allen Deviation. This is essentially the variance of the time derivative of  $N_{cal}$  over an increasing window. Theoretically, all timing standards should have Allen Deviations that tend toward 0, since averaging over the frequency differences over long periods continues to make the variance smaller. In reality, however, we have to deal with the fact that a single frequency standard will always drift due to physical limitations. In this document, we will use Allen Deviation in a comparative context to make engineering decisions; a plot from the manual of a Chip Scale Atomic Clock (CSAC) that we used shows some of the aforementioned characteristics (Figure 6.1).

In evaluating most time standards used in astrophysics, the most valuable figure is the standard deviation of the PPS. For the M12M receiver, it is given as 2 ns, which is the standard deviation of the distribution of PPS arrival times versus some better timing standard (see section 5.2. Here at Case, this is a rubidium frequency standard.

## 6.2 CWRU Time Tagging Module: TIM

The HEA group at CWRU has a venerable history of solving timing tasks in the field and in the lab using homemade equipment. The Timing Instrumentation Module follows previous solutions, being a small, self-contained time-tagging unit, designed for high accuracy over events with a low occurrence rate. Previous timing efforts in the group include finding the timing offset between the Auger FD and SD, and testing of GPS receivers for the initial Auger installation.

Given that we had to build and test firmware and software for the UUB, as documented in chapter 4, we needed a test platform. This test platform was a ZedBoard<sup>TM</sup> which ultimately became the prototype for TIM. We applied our system to a number of tasks, which

will be described in the coming sections. Before talking about applications, we will give an overview of the hardware, firmware and software of TIM.

In terms of purpose, TIM is dedicated to time-tagging tasks, both synchronous and asynchronous. Functionally, TIM does three things constantly: it runs and reads out the time-tagging system in its PL, it interprets the messages sent to it from the GPS receiver, and it ties these two outputs together into a data stream that can be parsed in post or in-situ depending on a particular application's requirements.

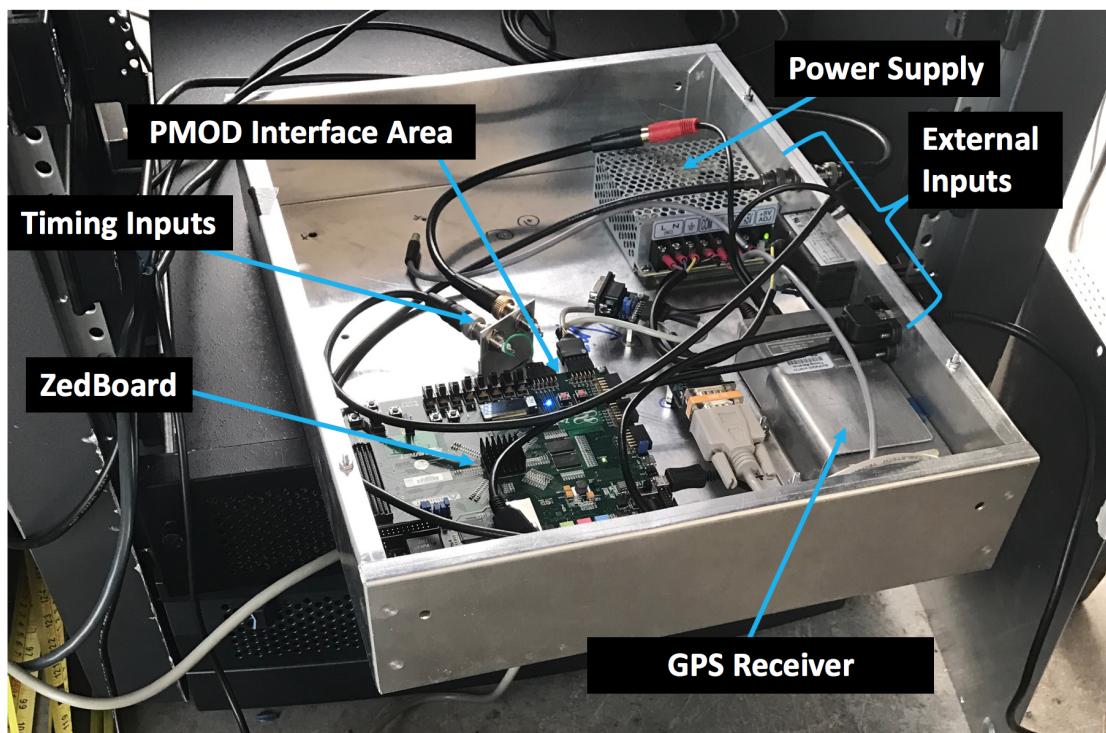


Figure 6.2: In this labeled photo, we have the major components of the TIM system as set up for its use in the spatial correlation of GPS timing drifts (section 6.3).

### 6.2.1 TIM Hardware

In parity with the UUB, the ZedBoard<sup>TM</sup> serving as the hub of TIM's abilities runs off of a Zynq<sup>TM</sup> 7Z020 chip. It runs a double core ARM Cortex A9 processor with an Artix-7 FPGA for its programmable logic device (given the acronyms PL or PLD). The ZedBoard<sup>TM</sup>

is designed by Avnet to showcase the Zynq™'s abilities, pinning out its various features into banks of usable inputs and outputs, some of which are programmable. A diagram of its high level architecture is given in Figure 6.3, while a longer description of the Zynq™ itself is in section 4.3.

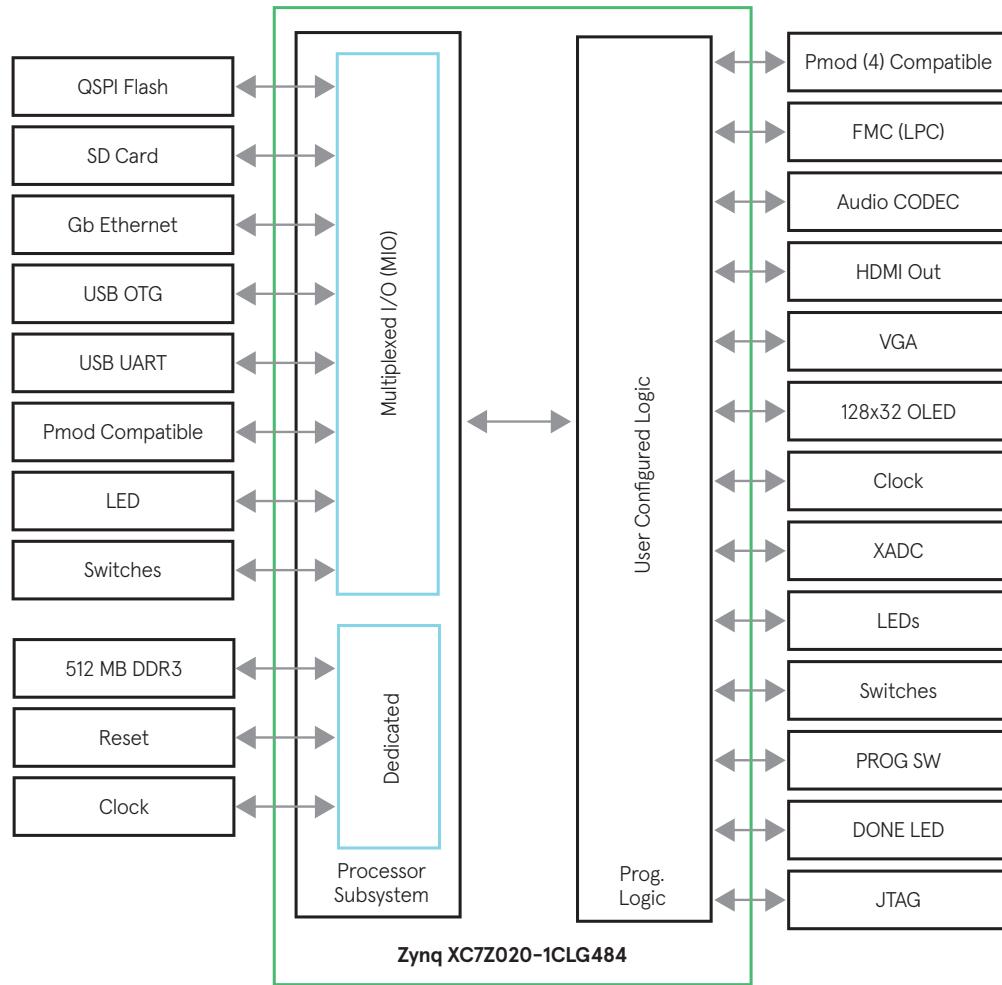


Figure 6.3: This diagram, from **zbspec**, shows the components of the ZedBoard™ and whether they are connected to the PS or PL side.

The ZedBoard™ features a footprint that is small enough to allow its use in portable lightweight systems, but not so small that it is crowded to work on. It is designed for learners so care is taken in its construction to be forgiving. All inputs are designed to handle any of the Zynq™'s available logic levels, and banks of PMOD connectors make it

compatible with standard Arduino™ accessories. These accessories are widely available and can be used to accomplish almost any standard function, although for our purposes, we simply used the RS-232 PMOD adapter and a homemade BNC bulkhead for timing pulses. Firmware and software can be loaded onto the ZedBoard™ via a built-in USB to JTAG converter for testing and via SD card for production versions.

Accompanying the ZedBoard™ we have a GPS receiver, a power supply, and frequently a peripheral under TIM’s command, or the facilities to make TIM a peripheral under another system’s command. In our test campaigns, the peripherals we used were both atomic clocks. They were the Microsemi SA.45s CSAC ([csac]) and the Stanford Research Systems FS-725 Rubidium Frequency Standard ([fs725spec]). A detailed description of the actual mechanisms of operation of these atomic clocks is well outside the purview of this work, but in both cases they take advantage of optical transitions, radio emission and mass spectroscopy to produce a precise frequency. Unfortunately, that does not mean they immediately know exactly when a second is; the atomic clocks still need to be trained to the GPS constellation to produce a precise PPS.

The GPS receiver we used for most of our testing and other work with TIM was an i-Lotus M12M, although TIM was used to test the Synergy SSR-6Tf that will be used in AugerPrime. Both of these will be described in subsection 5.2.2. The ensemble is rounded out with a 50W power supply with 5V and 12V rails and a fused wall-power bulkhead.

### 6.2.2 TIM Firmware

TIM’s firmware is written in much the same way as the UUBs, as described in section 4.6. Unless you have a solid understanding of how FPGA firmware is written in Vivado, section 4.6 and its subsections should be considered a prerequisite for this section. With that said, TIM’s firmware consists of two main facets: the time-tagging, which lives in the top

level wrapper, and the communications which live in the block diagram level. All inputs and outputs are tied in the constraints file to PMOD connectors on the ZedBoard<sup>TM</sup>.

The basic setup is always the same: the Zynq<sup>TM</sup> takes in signals from the GPS receiver and the time-tagging system. The variability of the multiple versions of TIM manifests in how TIM outputs its data and whether or not it needs to communicate with another peripheral.

### 6.2.3 Time-Tagging PL

Looking into the details of the timing PL, we can see below the actual Verilog code which controls the time-tagging system. This corresponds to the more human readable Figure 6.4. The Basic idea behind the time-tagging system is to have two 32-bit registers latched by pulses from ‘test’ and ‘ref’ inputs off of a 750Mhz counter. To put that in a different way, the time-tagging system has two register which can store values. When a pulse comes down either of the input lines, it causes the value of a counter to be read into the corresponding register and held until a signal is received from the processing side. The input lines are double synchronized, which ensures that multiple pulses will not be generated in the event of a misaligned input pulse.

```
Counter Counter1(CLK_750, regD, GRES);
//Use LED to display that counter is working.
assign LED = regD[28];
//Each of these LEDs corresponds to the latching flip-flop
assign LED2 = q5;
assign LED3 = q6;

//Double synchronize REF and TEST 1PPS inputs to 750MHz clock
DQ1 dqff1(CLK_750, RE{ }F1PPS, q1, GRES);
DQ1 dqff2(CLK_750, q1, q2, GRES);
DQ1 dqff3(CLK_750, TEST1PPS, q3, GRES);
DQ1 dqff4(CLK_750, q3, q4, GRES);

//Set and clear the status bits when STRB is high
```

```

// or when global reset is called.
//q5 status bit is set when REF1PPS is latched
//q6 status bit is set when TEST1PPS is latched
INVgate invert(NSTRB, STRB);
ANDgate RESTest(CLR, NSTRB, GRES);
DQ2 dqff5(q2, 1'b1, q5, CLR);
DQ2 dqff6(q4, 1'b1, q6, CLR);

//Continuously update registers unless status flags are high
Reg1 register1(CLK_750, q5, regD, RA);
Reg1 register2(CLK_750, q6, regD, RB);

```

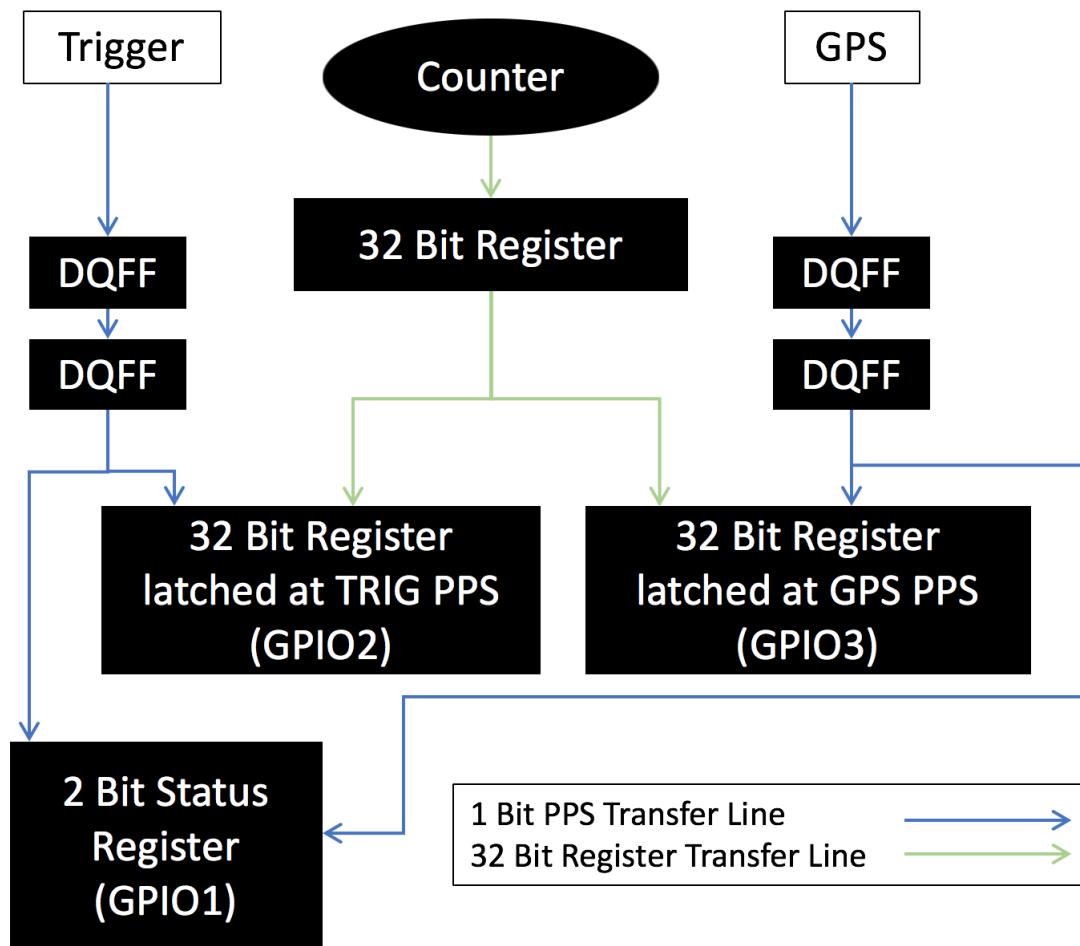


Figure 6.4: This diagram represents the time-tagging logic shown in the code section above (6.2.3).

This time-tagging system is derived from an early version of the AugerPrime time-tagging module, with the main modification being the increased speed and the added status

LEDs. In the iteration shown above, the system is optimized for precise tagging of low-rate triggers. Using only software modifications, it can be made to do synchronous time-tagging for frequency standard testing, or asynchronous time-tagging for real science event tagging.

### 6.2.4 Communications PL

In terms of communications with the GPS receiver and any peripherals, TIM uses the same setup as the UUB: a UART Lite ([lite]) for each serial line required. The UART Lite must be hard programmed with the desired baud rate and parity bit settings. The convention for these is 9600bps for GPS receivers and 57600bps for atomic clocks. In the case of the TIM unit deployed in the Telescope Array, it is operated at the maximum 115200bps rate where it is used as the main output of the system. In all other applications, the built-in USB UART of the ZedBoard<sup>TM</sup> is used at an emulated 115200 baud rate. This chip is actually an on-board USB to Serial converter.

As mentioned before, the process of implementing the UART Lite on a Zynq<sup>TM</sup> chip is well documented in section 4.6. The full block diagram for TIM is shown below, in Figure 6.5. The 0th UART Lite stays connected to the GPS receiver, while the 1st UART Lite is designated for peripheral communication or system output. Internal communications with the time tagging system are accomplished through the General Purpose Input/Output (GPIO) blocks shown in Figure 6.5, where the 0th GPIO causes the reset of the status and data registers and corresponding flip-flops, the 1st GPIO reads the status registers and the 2nd and 3rd GPIOs read out the data registers.

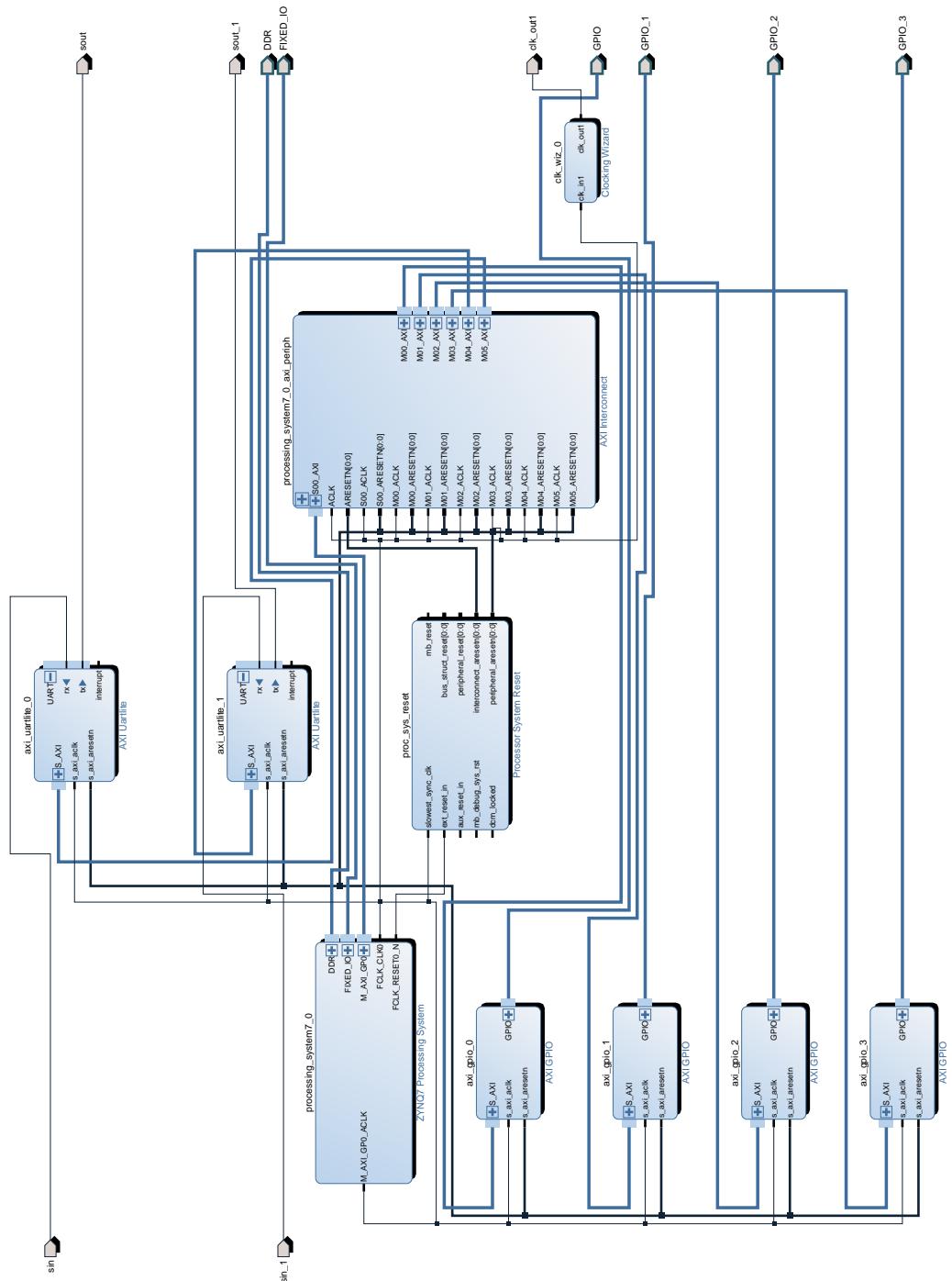


Figure 6.5: Here we have the full block diagram for the standard use case of TIM.

## 6.2.5 TIM Software

The software for TIM can be broken down into two parts. First we will discuss the software which operates the device itself, and then we will move into the analysis package.

### 6.2.5.1 TIM Operating Software

The software for operating TIM can be cleanly broken down into two categories corresponding to the readout of the time-tagging system and the readout of the GPS telemetry. These two tasks are implemented in C++ in a simple *while(True)*. Since the critical data are held in either the UART or the timing registers until readout, we need only ensure that the software will read them out in a relatively timely manner. The interactions through software ultimately limit the event rate of the original version of TIM (see subsection 6.3.5). The software is written for Xilinx's Standalone Operating System, a bare bones OS giving almost all control outside of very basic functions to the developer.

To implement control of the time-tagging firmware, we can simply check the status register and read it out when either both flags are high for synchronous time tagging, or when one flag is high for asynchronous time-tagging. This will be discussed in explicit detail in subsection 6.3.4. The block of code shown below accomplishes this task for synchronous time-tagging.

```
if (current_status == 0x03) {  
    refin = XGpio_DiscreteRead(&refin_device, REFIN_CHANNEL);  
    //read refin register  
    testin = XGpio_DiscreteRead(&testin_device, TESTIN_CHANNEL);  
    // read testin register  
    print("Pair: ");  
    printf("%010lu, ", refin); // print refin value to stdio  
    printf("%010lu\n", testin); // print testin value to stdio  
    // clear status register and start over  
    XGpio_DiscreteWrite(&clrstatus_device, CLRSTATUS_CHANNEL, SET);  
    XGpio_DiscreteWrite(&clrstatus_device, CLRSTATUS_CHANNEL, CLEAR);  
    XUartLite_Send(&acuart, buffsend_ac, sizeof(buffsend_ac)); }
```

From this, we can see that the software tasks of time-tagging boil down to readout and reset. To briefly explain the code itself, we have used all Xilinx driver functions to interact with the time-tagging system. These are the *XGpio* commands, which come in read and write varieties. The read commands simply output the value of the register they are reading out, while the write commands set the value of the final argument into the register they are addressing. This block first reads in, then prints the data registers, and subsequently resets them with the write commands. The final *XUartlite\_Send* is for polling a peripheral's status synchronously. If no peripheral is connected, this can be ignored or removed. The *0x03* check represents the hexadecimal value of the status register when both latch flags are high. If we were to, for example, add channels for time-tagging, we would need to modify this block accordingly, adding readouts for the new channels and changing the status check hex code appropriately.

The GPS communications portion of the operating software is somewhat more complicated. It starts by checking if there are any new bytes in the UART to load. This occurs at the beginning of each iteration of the *while(True)* loop. Then, if it finds new bytes, it copies from a temporary buffer to a message buffer. After this, it increments a counter corresponding to the write position in the message buffer, and checks to see if the ‘end-of-message’ characters (carriage return and line feed) are in the buffer after copying. If it finds these, it then scans the message buffer for the beginning of a message. Finally, it checks if the byte before the end-of-message is the correct XOR checksum of all of the characters from the beginning of the message up to the byte before last. If this check is passed, the message is output and cut from the message buffer. The remaining characters in the message buffer are copied to the beginning and the process starts again.

If for some reason the message does not correspond to a known format, (e.g. unrecognized message type, transmission errors etc.), the contents are dumped to output in decimal

format. This was chosen for debugging purposes, as it makes it easy to use an ASCII table to readout the message header. If the checksum fails, the parser continues as though it had not received an end-of-message. Upon finding a new healthy message in the buffer, it will delete all previous contents. The procedure is the same if it fills its entire message buffer without a healthy message.

In general, interfacing with atomic clocks is a simpler process than interfacing with the receiver. For the two atomic clocks we used, one needs only send a polling message and receive an ASCII string. This is done synchronously, and we do not need any complex parsing; the message can be directly printed to output.

In terms of output, all of TIM's messages are human readable, which ensures that the instrument is useful in the field without having to invoke any complex software to interpret the output. This complexity is pushed into the output parser software, to be discussed in the next section, subsubsection 6.2.5.2. An example of TIM's output is shown below.

```
Pair: 4220156538, 4220156364
you got a Ha message. HA!

healthy message
day: 23
month: 7
year: 2017
hour: 20
minutes: 40
seconds: 25
latitude: 149411101 longitude: -293785686 altitude: 20855
number of satellites visible: 9
tracked: 9
satid: 2 Track: 8 Signal: 51 IODE: 178 Chanstat: 8 160
satid: 6 Track: 8 Signal: 52 IODE: 51 Chanstat: 8 160
satid: 12 Track: 8 Signal: 50 IODE: 56 Chanstat: 8 160
satid: 19 Track: 8 Signal: 49 IODE: 75 Chanstat: 8 160
satid: 17 Track: 8 Signal: 52 IODE: 47 Chanstat: 8 160
satid: 5 Track: 8 Signal: 50 IODE: 43 Chanstat: 8 160
satid: 25 Track: 8 Signal: 44 IODE: 22 Chanstat: 8 160
satid: 9 Track: 8 Signal: 49 IODE: 65 Chanstat: 8 160
satid: 23 Track: 8 Signal: 49 IODE: 35 Chanstat: 8 160
```

```
you got a Hn message.
```

```
sawtooth: -2
CSAC Message: 169
```

Some of the details here are only relevant in context, but a quick examination shows that we get time and position information, as well as the counter values (after *Pair:*) of the synchronous time-tag. First the time/position/status message is parsed, and then the T-RAIM message is parsed. See Table 4.5 for more details on the messages, and subsection 4.5.3 for information on the Motorola Binary protocol.

#### 6.2.5.2 TIM Analysis Software

To accommodate TIM's different use cases, the analysis software is broken into three python scripts which deal with all synchronous time-tagging situations. Asynchronous time-tagging analysis software will be discussed where relevant in the coming sections. In a synchronous single TIM setup, as in section 5.2, the parser is the only necessary software. In dual TIM situations, such as section 6.3, we will need all three pieces. These scripts are:

- **Parser** (`advanced_parser3.py`): this piece of software goes line by line through TIM's human readable output and packs the data into a .csv file and calculates the time difference between the synchronous inputs with sawtooth correction. It tags each data point with an integer representing time from an arbitrary epoch, for use in `joiner.py`. If it fails to find all of the required data, it skips that second and looks for the next fresh second's worth of telemetry. The parser is used in single TIM applications such as GPS testing to produce the data set for analysis in Origin or another chosen program, and in dual TIM applications to prepare the data for joining.
- **Joiner** (`joiner.py`): the joiner simply takes two parsed .csv files and matches each line in both files by epoch. This is only needed in dual TIM applications. If it cannot find

a match, the line is skipped. The joiner assumes temporally sequential data points in both files.

- **Analyzer** (`small_analyzer2.py`): to finish a dual TIM analysis, we employ the analyzer, which takes the joined data files and finds time differences of the time differences on each station. It also can be configured to compare atomic clock telemetry. It uses an exact plumb line distance formula to calculate separations between two TIM stations.

## 6.3 Spatial Correlation of GPS Timing Errors

To address concerns in Auger about the possibility of time drifts on the order of tens or possibly hundreds of nanoseconds, we set out to measure and correlate the timing errors of GPS receivers separated by varying distances. In doing so, we are probing ionospheric fluctuations, which are known to be the largest contributor to spatial and timing errors in GPS [**milspec**]. This technique is becoming more common in recent years [**atmoex**, **raulgps**] as a method of characterizing the electron content of the ionosphere. In this section, we will walk through the experimental setup and some of the leg work that was required to finalize the measurement campaign’s strategy. This will include a brief discussion of how to train an atomic clock. Finally, we will discuss the results of the measurement campaign and what can be done to improve the accuracy and meaningfulness of the results.

As shown in Figure 6.1, the GPS constellation is the most accurate timing source over the long term. This is a consequence of its constant training to the US Naval Observatory’s bank of Cesium atomic clocks. That said, over short time scales, GPS is less accurate since the oscillator in the receiver has a relatively low accuracy compared to the constellation it is trained off of due to its low frequency and ionospheric (and other) systematic disruptions.

### 6.3.1 Experimental Setup and Method

If our objective is to measure the timing drifts of GPS receivers, we will first need a frequency source which is more accurate and stable than the receiver on relatively short time scales; this will take the form of an atomic clock. We will then couple this to a TIM unit and use its time-tagging abilities to find the timing error of the GPS receiver. We then need to find a suitable method for testing, in particular we will need a site (or sites) and a time interval over which to test the correlations.

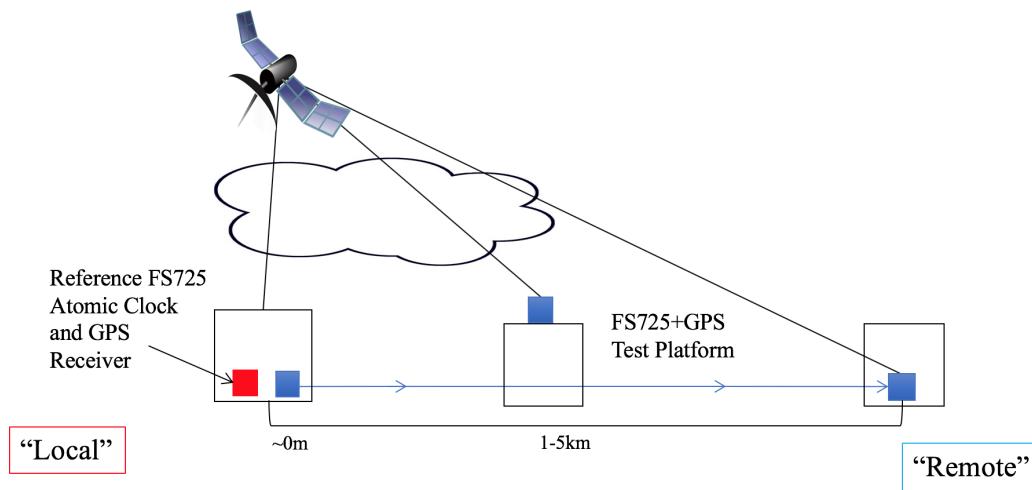


Figure 6.6: Here we have a schematic overview of the setup of our spatial correlation tests.

In order to understand our choice of atomic clocks, we must first review the different data collection techniques we tried. The testing here can be neatly divided into three phases: (1) car-borne short term (approximately one hour) campaign, (2) car-borne medium term campaign, and (3) desktop long term campaign. Ultimately, due to the long training times required by the atomic clocks, we only took usable data from the third campaign.

For the atomic clock, we tested two models. The first was the SA.45s CSAC from Microsemi, and the second was the FS725 Rubidium Frequency Standard from Stanford Research Systems. The CSAC comes in a very small form factor, approximately one square

inch, and is intended for applications where GPS may not be available. When we purchased the first CSAC from Microsemi, the company had not completely worked out all fabrication issues and was relatively straight forward, informing us of this via email. We proceeded to purchase a CSAC as the temperature range issues they were fixing would not affect our campaign. However, we found after our first attempted data taking campaign that the CSAC only stays accurate to the GPS PPS for about 15 minutes after it is no longer being directly trained. Attempts at removing this error with a linear deviation model were unsuccessful. After testing the CSAC, we decided the small form factor was not worth the compromise in accuracy and we obtained two FS725s. These larger desktop atomic clocks feature a much lower Allen deviation than the CSACs, although they are bulky and power hungry in comparison [[fs725spec](#)]. In addition to eschewing the CSAC, we also incorporated the knowledge that we cannot go for as long of a campaign if we do not somehow train the atomic clock.

As long as the time scales of the atomic clock's training process (24-72 hours) are significantly longer than the time scales of GPS fluctuations (tens of minutes or a small number of hours), we should be able to pick out fluctuations in the GPS receiver's timing even while training to the atomic clock. The long training cycle of the atomic clock is born out by preliminary testing data (i.e. if we look at the excursions of the PPS as measured by the clock, we see them settle to an average of zero over a small number of days). The atomic clock uses a simple process of seeking the correct frequency by training away the differences between its calculated PPS phase, and that of the GPS receiver. This is detailed in the user manual [[fs725man](#)].

Upon testing the FS725 in a car-borne test, we quickly discovered how temperamental the clock's training can be. Since the stability depends on the temperature and magnetic field, the clock's PPS would drift upon being moved. Compounding this is the fact that

we cannot maintain an accurate enough GPS fix for nanosecond timing while moving the vehicle, and the transitions from moving to setup were not smooth (i.e. moving from the car antenna to a standing antenna is disruptive). Furthermore, the setup required a fair amount of power with the FS725 and monitoring equipment, consisting of a Raspberry Pi and LCD monitor. In an attempt to keep the atomic clocks running and trained during transitions into and out of the vehicle, we ‘buffered’ the power lines with large uninterruptible power supplies.

Since the atomic clocks were extremely sensitive to being moved around, only the final campaign was successful in taking workable data. This consisted of moving one atomic clock to new locations in the University Circle area every week and a half. The difficulty here was finding willing participants, who would let us mount this apparatus in their apartment, and finding ways to give the antenna as close to full sky coverage as possible. These participants were Case graduate and undergraduate students, and friends of the author.

To summarize, after attacking the problem of how to measure these spatial correlations multiple ways, the trial which we finally employed was a campaign of a month and a half, in which the TIM setups were moved to five different separations, four of which yielded usable results.

### 6.3.2 Measurement

The objective of the experiment here is to measure how correlated the timing signals of two GPS units are a function of spatial correlation. To accomplish this, we will need the two ‘time streams’ of each TIM setup, i.e. their second-by-second time differences between the GPS PPS and the atomic clock’s PPS. Once obtained, we take the difference of each set of second-by-second time differences and make a histogram of it. We calculate the

standard deviation of each distribution. If the noise is completely uncorrelated, then we expect standard deviation of the difference between the time streams to be the square root of the sum of the squares of the standard deviations of each time stream's distribution, i.e. the standard deviations of each time stream added in quadrature. We will call this quantity the maximum error. If there is correlation in the noise, we will see the standard deviation of the difference between the time streams to be less than the maximum error. An example of time streams being subtracted is shown in section 1.1 To be complete, we can

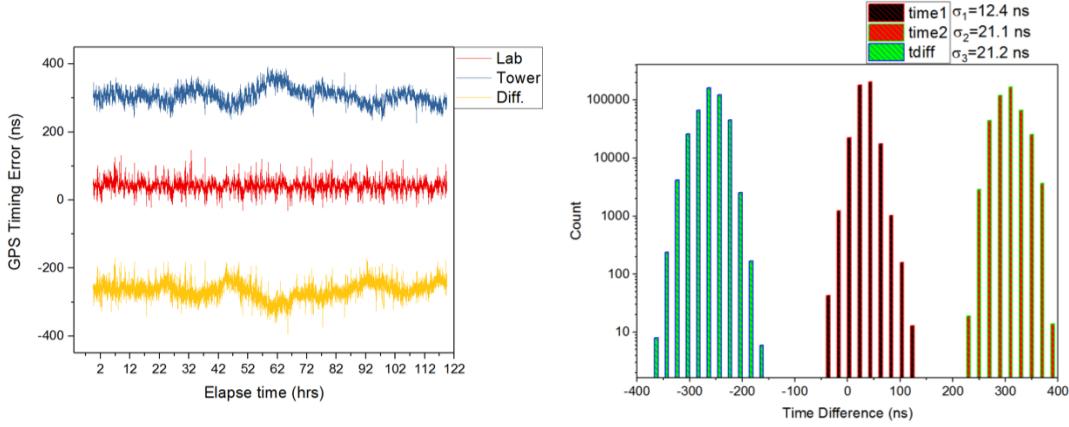


Figure 6.7: Left: an example from unused engineering data of the 3 time streams we need to create a data point. In this, the tower referred to is the Ham Radio club's tower on the roof of Case's Glennan Building.

also attempt to assess the amount of noise the atomic clock's training adds. The strategy for quantifying this was to train one atomic clock off of another atomic clock, trained off of the GPS constellation. The width of the Gaussian distribution of the time differences between the atomic clocks gives the training error, which was found to be 3.3ns.

In order to assess our errors, we will calculate multiple standard deviations for each site using a sliding 4 hour window. With this, we can make statistical error bars. The final products will include a plot of the standard deviation at each separation for the stations, the differenced time stream and the max error. If the max error is above the deviation of the subtracted time stream, we will consider the signals correlated.

### 6.3.3 Results

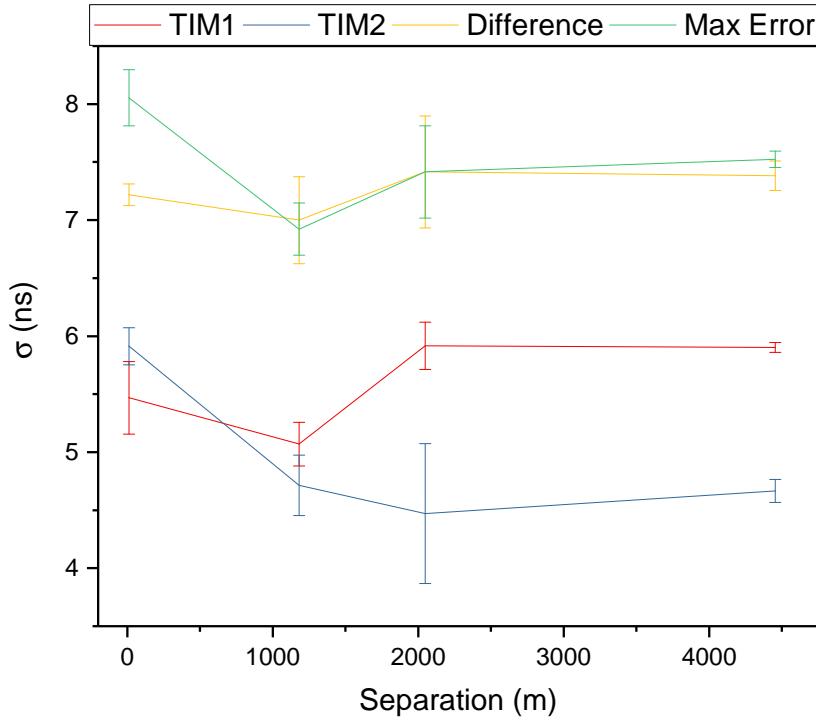


Figure 6.8: In this plot, we have the final product of experiment on the spatial correlations of GPS timing errors. Detailed explanations of this plot are given in the previous section. The max error is the standard deviation of each TIM setup added in quadrature, while the differences are the standard deviation of the second-by-second differences in the timing errors of the two TIM stations.

In Figure 6.8, we have the final results of the analysis of the timing errors. This plot shows that we can only claim spatial correlation of GPS errors out to a couple hundred meters. That said, it also does not show any serious timing drifts, which would have dramatically increased the standard deviation of the differences between the stations. There is still some possibility of a long term phase in their timing and we will discuss how to determine this in subsubsection 6.3.3.1. An interesting and unexpected observation we made was a severe thunderstorm coming in over Cleveland. This is shown in Figure 6.9, where the standard deviation jumps from 6.7 ns, to 15.0 ns.

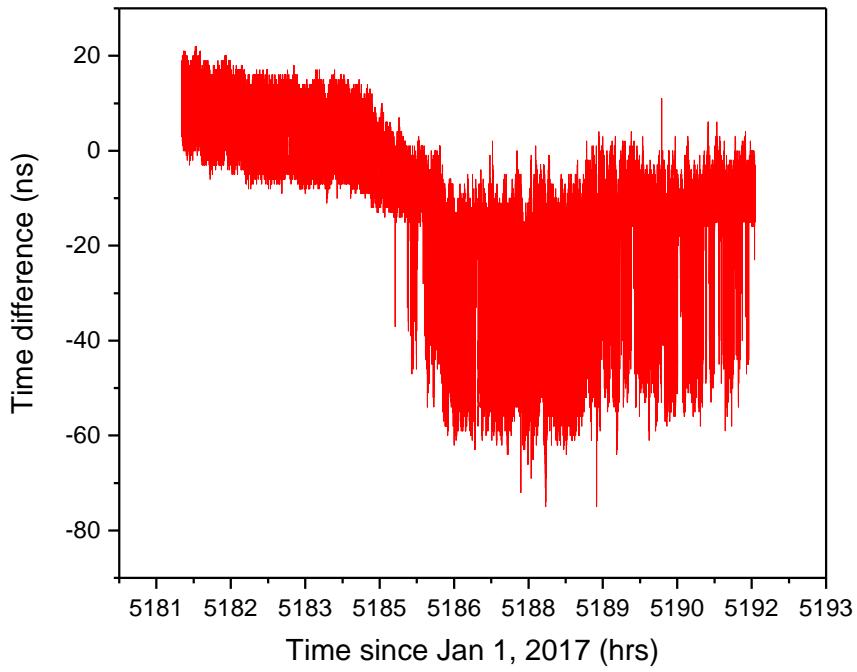


Figure 6.9: Here we have a storm which came in while we were taking data. At hour 5185.5 as marked on the graph, the storm hits and the timing solution of the GPS receiver becomes erratic, increasing its standard deviation by almost a factor of 3.

### 6.3.3.1 Future Directions

If we were to undertake this experiment again, one of the following would need to be a prime focus:

- Understanding and simulating the training algorithm of the FS725: this is certainly possible and we began working on this with an undergraduate but the project never came to fruition. The user's manual for the device explains how the training algorithm works, although some of the parameters of it are not clearly laid out. If we were to continue down the path, the author would recommend contacting Stanford Research Systems for their assistance in the simulation.

- Conducting the test using kilometer long cables: instead of using two atomic clocks as the time standard, one could get away with using one atomic clock at a central location and moving two stations away while they are both connected via BNC cable. The hard work here would be characterizing the signal losses of the BNC cable (likely requiring amplification of the signal) and finding a suitable location. The Case farm administration has invited us to complete this test on their facilities. It is worth noting that the price of CCTV BNC cables is relatively low and the HEA group's homemade discriminators could deal with the pulse marring that would occur over the long distances.

### 6.3.4 TIM@TA

In this section, we will discuss the role of TIM in the Auger@TA effort. This work resulted in one publication and another in progress [**augerta1**]. We will include a brief discussion of the general effort towards the cross calibration, and then a description of the timing hardware built for the effort, and some of the engineering details that had to be addressed. Scientific motivations for Auger@TA are laid out in subsubsection 3.3.2.1. The TIM unit sent to TA uses a smaller footprint board and so it will be referred to as SlimTIM.

Towards cross-calibrating Auger and TA, members of our group were allowed to access the TA Central Lasing Facility, which has the necessary network connections and amenities to run a set of detector stations. In particular, the CLF can provide power and an internet connection, and is accessible by road. Additionally, there are TA detector stations located at the CLF which we used for the in-situ cross calibration.

The ideal situation to allow for a direct Auger VEM vs. TA MIP calibration, is two stations co-located as closely as possible. This would be feasible but the TA electronics, to their credit, do more of the work in the PL than the Auger electronics do. This means that

they will have faster processing times and less worries about bottlenecks, but it also means you must have a skilled electrical engineer and HDL source code to make modifications to the station. Furthermore, TA blinds its data sets for a year after they are taken. This means that events cannot be read out and accessed by anyone, stifling a cross-calibration.

This problem was addressed by effectively constructing a new station utilizing TA's signal chain up to the digitizers (i.e. scintillator, PMT and preamplifier), and using custom and store bought electronics for the rest of the station. The chosen electronics include a single board computer, a PicoScope for digitization, a comparator and AND gate for triggering, and finally SlimTIM for time-tagging.

SlimTIM takes the firmware from TIM and ports it to a smaller, MicroZed<sup>TM</sup> package. The MicroZed<sup>TM</sup> performs essentially the same functions as a ZedBoard<sup>TM</sup>, except most outputs are not pinned out and it therefore requires an expander board to make use of the majority of the PL I/O ports. Outside of this, the MicroZed<sup>TM</sup> runs the exact same Zynq<sup>TM</sup> chip as the ZedBoard<sup>TM</sup> and is capable of the same functionality.

Ultimately, SlimTIM enabled the time-tagging which led to the MIP vs. VEM comparisons currently used by the Auger@TA working group. At this point, the device has been functioning in-situ without interruption for over 2 years.

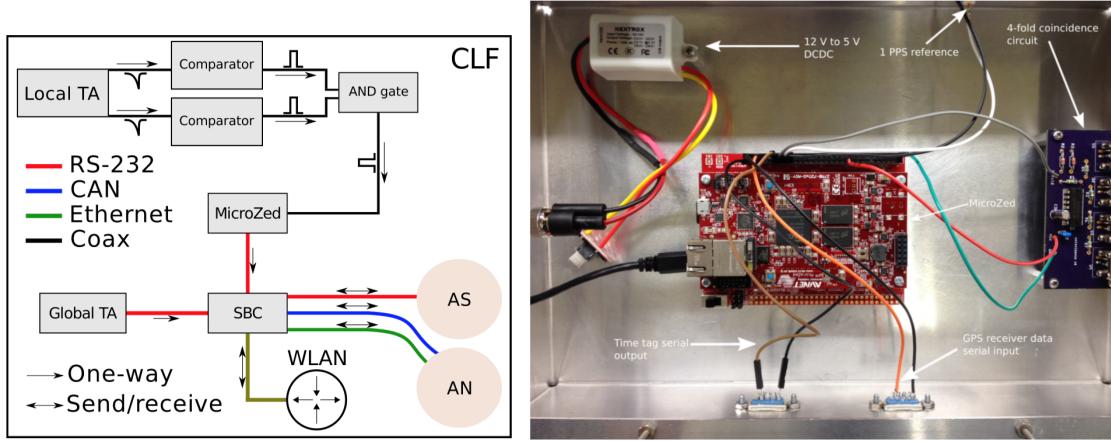


Figure 6.10: Left: A block diagram of the custom detector deployed for Auger@TA. Right: A labeled photo of SlimTIM in the project box. Both from [sean](#).

### 6.3.4.1 Engineering Challenges

A number of the capabilities of TIM had to be modified to meet the requirements of the Auger@TA project to create SlimTIM. Fundamentally, the biggest difference is that TIM had only been used for synchronous time-tagging. Other fine points include the firmware and software port over to the MicroZed™, which was non-trivial, and changing the output connection from USB to RS-232. We will discuss these here in the order of firmware changes, then software changes.

To implement these changes we started with the firmware. In Vivado™, we can re-configure the peripheral UART Lite to output at 115200bps, the maximum safely possible over a standard DB-9 serial RS-232 line. This will drop the amount of dead time required by both SlimTIM to send the time-tags, and the Single Board Computer (SBC), to receive them. For simplicity, we will set the standard I/O to output over this UART, thereby curbing the need to develop and debug further serial communications software. Some debugging was required after the rest of the system was developed where we ultimately found that the signals coming off of the MicroZed's™ serial in and serial out pins from the programmable

logic were outputting inverted signals. It is unclear if the inversion occurs due to some facet of PL programming, or if this is an error in the manufacture of the MicroZed™. What is clear, however, is that this can be fixed by putting a PL inverting gate in the serial output signal chain. Discovering this error required a number of hours of debugging and oscilloscope viewing time.

Setting the Standard I/O (frequently referred to as *stdio*) seems simple, but we must keep in mind that we do not have only one effective operating system. On a standard PC, *stdio* is passed from the BIOS to the operating system. In an embedded system context, we do not have BIOS to handle this and other basic functions, so *stdio* transfer must be made explicit. This can be done by opening the configuration panel for the FSBL in Xilinx SDK. Anytime we have multiple possible *stdio* options, we must remember to check this. In many cases, SDK seems to choose *stdio* for us, and it does not always make the correct choice.

Finally, we need to make the software change which will allow SlimTIM to perform asynchronous time-tagging. Referring to the code block at 6.2.5.1, we can see that the operating software for TIM checks to see if both bits of the status register are high before it is readout. To appropriately modify this, we can copy this block twice and put both copies above the original. Then, by changing the hex check to 0x01 and 0x02 on the respective blocks, cutting out the lines corresponding to the other register's readout, and leaving the reset in, we are prepared to catch all three possible cases: TESTIN goes high, REFIN goes high, and both go high (within a handful of PS clock cycles).

### 6.3.5 TIM@CTA

As described in subsection 3.3.4, the HEA group at CWRU has been charged with developing and implementing the timing electronics and infrastructure for the pSCT in CTA. The

timing work here will enable the telescope’s abilities for cross-calibration, multi-messenger astronomy and searches for transients. Below we will first describe the requirements of the system, and then move into our technical solution in two parts, and finally give a status report of the progress so far. Ultimately, the efforts here will be completed without the author, but we venture to describe as much of the work here as possible.

### 6.3.5.1 Requirements

While previous work described in this document has largely pertained to timing and electronics of water Cherenkov detectors, the pSCT is a large gamma ray telescope, and so it has many operational differences but retains many technological similarities. Amongst these differences is the much higher rate of events. Since the pSCT will be working in the high GeV to mid TeV energy range [**ctaong**], the technical requirement given for trigger handling is 1 khz. Before this project, the fastest verified rate of TIM’s time-tagging was 20 hz. We will therefore focus on the increase in event rate handling capabilities in Figure 6.12. The time-tagging accuracy specification for the pSCT is 1ns; however, this is for stereoscopic reconstruction amongst multiple pSCTs.

The other relevant technical task is surrounding this effort is the transfer of logic pulses from the telescope’s camera to the trailer where the time-tagging system will necessarily be located. All lines in and out of the telescope must be fiber optic for lightning considerations and so there is no way to accomplish the task of creating and offloading the time-tags without some sort of fiber optic transfer setup. With this in mind, Corbin Covault, Bob Sabin and myself have set out to put together both a timing pulse transfer system (subsubsection 6.3.5.2) and a high speed time-tagging system which we will call TIMPrime (Figure 6.12).

### 6.3.5.2 Timing Transfer System

The Timing Transfer System (TTS within this work) will consist of a laser emitter box, called the L-Cube, coupled to a long fiber optic line that leads to the pSCT trailer ('the trailer' for brevity) [**lcube**]. On the input end of the L-Cube, we will have a connection to the backplane of the pSCT camera, which outputs TTL pulses upon any pixel level triggers in the camera. This rate should be about 1 khz, while the L-Cube is capable of handling rates as fast as 5 khz. On the other end of the TTS is a ThorLabs fiber optic coupling and

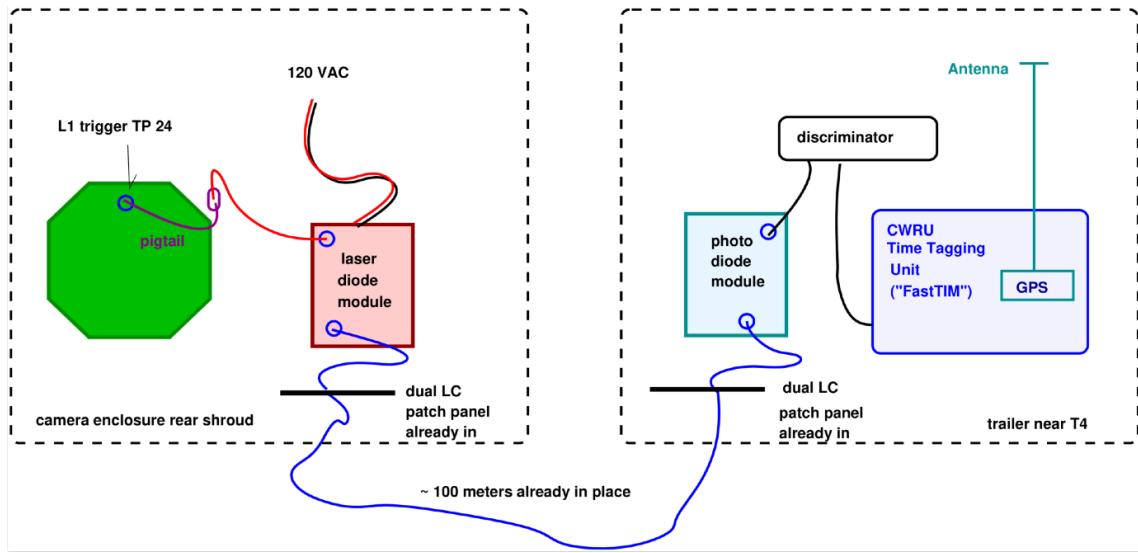


Figure 6.11: Here we have an end-to-end diagram of the TTS, in which we can see the TTL → Fiber Optic → TTL → Time-tagging structure. This diagram is courtesy of Corbin Covault, and has not yet been published.

photo-diode that we use to reconvert the optical pulse into a logic pulse to be transferred over a standard copper line. The pulse height coming out of the photo-diode is proportional to the pulse height coming in, and is not necessarily a standard logic format or shape, and so we use one of our homemade HEA lab discriminators built by Bob Sabin. This takes any pulse over threshold and outputs a 3.3V logic pulse instead. TIMPrime then times these pulses.

### 6.3.5.3 TIMPrime: Time-tagging System

Looking back to the time-tagging PL of TIM (subsection 6.2.3), we note that it is intended for very accurate time-tagging of low rate event pulses. As per the requirements of the pSCT, we will need a high event rate while maintaining a fair level of accuracy. To accomplish the event rate increase, we propose to replace the TEST count register in TIM's PL with a First-In-First-Out or FIFO buffer. The Xilinx IP for this is called the *AXI FIFO Generator [fifo]*, and it features a number of options.

In customizing the FIFO, we first select the width to be 32 bits, to match the counter. We then can choose any practical depth for the FIFO, and so we will, somewhat arbitrarily, pick 1024 ‘words.’ In the Xilinx parlance, ‘words’ translate to FIFO entries and not 2 byte data types as they might be elsewhere. This depth ensures that there will still be FIFO room even if the PS takes longer than expected to receive a GPS message or readout the REF (GPS PPS) register. In terms of accuracy, we must make a small sacrifice moving from 750Mhz to 400Mhz on the counter frequency due to the requirements of the FIFO [fifo]. This brings our coarse timing accuracy down from 1.3ns to 2.5ns, which does not meet the pSCT requirements but should be more than accurate enough for the activities mentioned at the beginning of the section.

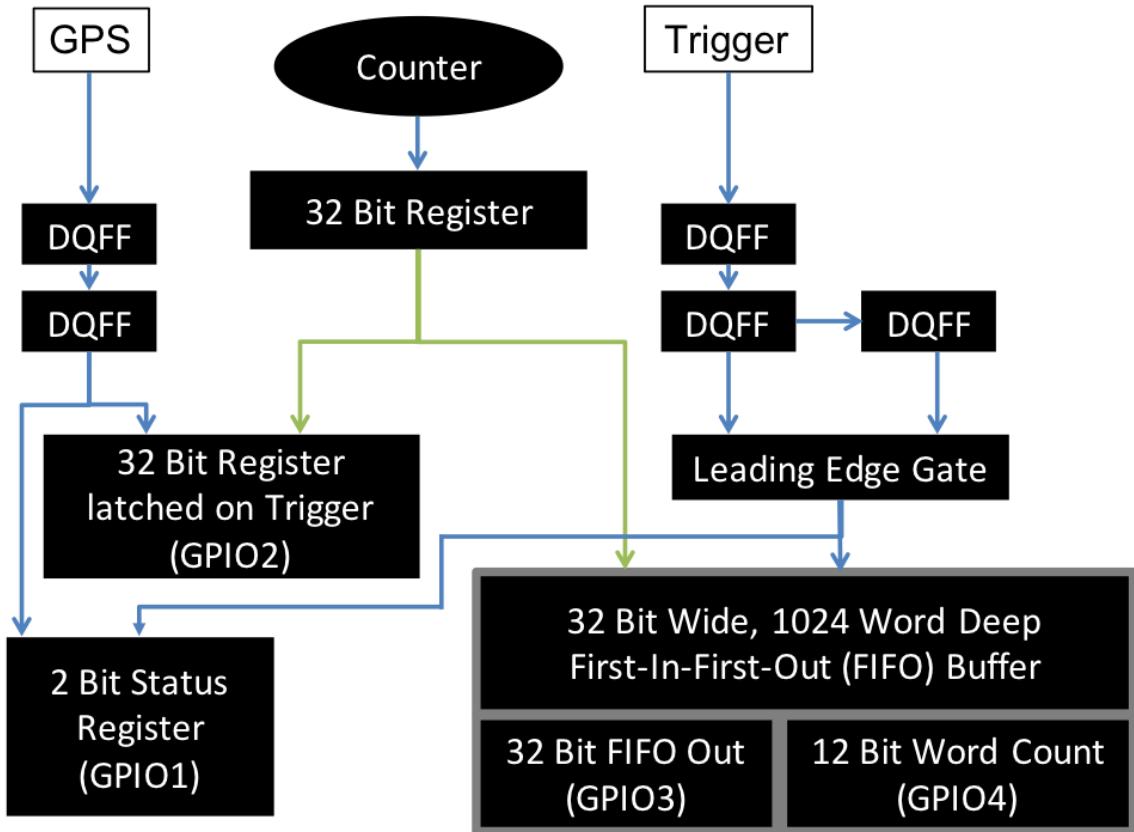


Figure 6.12: In this diagram, we can see the modifications made for TIMPrime to handle the higher data rate on the right signal line. Compare to Figure 6.4.

Next, we will have to ensure that the most current counter value is always available to the FIFO. This consists of going into the top level wrapper and either changing the control bits for the TEST/trigger register, or bypassing it altogether. At this time, the goal is to have it double synchronized, so an extra register has been added for a total of two buffer clock cycles from the counter to the FIFO. Additionally, since we will need to know how full the FIFO is to appropriately read it out, we add a 10-bit GPIO, which will be the 4th in the system, to read this out.

Perhaps the largest concern, and the work that is still under way is the read and write enable logic. The ports that we are concerned with here are respectively *rd\_en* and *wr\_en*. To simplify the read and write enable logic, we choose a First-Word-Fall-Through

(FWFT) configuration for our FIFO, which allows the FIFO to be read asynchronously without asserting a *rd\_en* pulse. Without this, the FIFO has to readout synchronously and data stays hidden until it is being read. The FWFT configuration alleviates the need for precise synchronization of the FIFO data buffer and the GPIO data input.

According to the IP's manual, **fifo**, counter values will be read into the FIFO upon asserting a *wr\_en* (it looks for a leading edge) and then they will move directly through to the readout registers. Upon a *rd\_en* leading edge, the current readout entry will be trashed and the next sequential entry will move to be readout. In this way, we simplify the software requirements both in terms of complexity and in terms of urgency.

The software for reading out the FIFO as it stands is a simple while loop where the FIFO is readout while more than 5 entries are waiting. This abates the issue in FWFT FIFOs that their word count can be off by a handful of entries. The block which reads the REF (GPS PPS) counter register will be changed to look for 0x01, and all other components of the software will remain the same. With this configuration, we believe we can build a timing system meeting the 1 khz requirement of the pSCT.

#### 6.3.5.4 Current Status

**SAVING THIS SECTION FOR LATER, should be short and sweet**

# Chapter 7

## Correlation of UHECR arrival directions with Starburst Galaxies

In this chapter, we will investigate the correlation of Starburst Galaxies with the arrival directions of UHECRs invoking the JF12 model with uncertainties to give a physical account of the spread of Galactic and Extragalactic Magnetic Fields (GMF and EGMF, respectively) [[jf12](#), [sean](#)]. We begin by explaining some of the goals of charged particle astronomy and evaluating the Hillas Criterion on Starburst Galaxies.

## **7.1 Charged Particle Astronomy**

### **7.1.1 SBGs and the Hillas Criterion**

## **7.2 Magnetic Field Modeling and JF12**

## **7.3 Starburst Test Methods**

## **7.4 Starburst Test Results**

## **7.5 Conclusions**

# **Chapter 8**

## **Conclusion**

# Appendix A

## Glossary

**Glossary AXI:** Advanced eXtensible Interface, commonly used bus for transferring data in embedded devices and smart phones.

**MOPS:** Multiplicity Of Positive Steps, a trigger type in the UUB.

**PL:** Programmable Logic, this is the firmware and logic features running on the FPGA portion of the Zynq™ chip.

**PPS, 1PPS:** Pulse per Second or One Pulse Per Second, usually refers to the GPS synchronized pulse put out by a time standard such as an atomic clock or GPS receiver.

**PS:** Processing System, this is the half of the Zynq™ chip that hosts the operating system and functions as a system-on-a-chip.

**SEP:** Spherical Error Probable, the radius from the mean of a collection of points within which 50% of the points lie, used in GPS parlance.

**TOT:** Time Over Threshold, a type of trigger used in UHECR detectors.

**UB:** Unified Board, the original integrated electronic board for the Auger observatory, predecessor to the UUB.

**UUB:** Upgraded Unified Board, the board being fabricated for the AugerPrime upgrade.

**UHECR:** Ultra High Energy Cosmic Ray, Cosmic rays with energies  $>10^{18}$  eV.

UART: Universal Asynchronous Transmitter/Receiver, the device which handles asynchronous communications, such as those for an RS-232 connection.

VCXO: Voltage Controlled Crystal Oscillator, and oscillator whose voltage is controlled by a changeable reference.