Stream Communication and Computation in the Eight-meter-wavelength Transient Array (ETA) Radio Telescope

Brian S. Martin

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Masters of Science
in
Electrical Engineering

Dr. Cameron D. Patterson, Chair
Dr. Steven W. Ellingson
Dr. John H. Simonetti

September 12, 2008
Bradley Department of Electrical and Computer Engineering
Blacksburg, Virginia

Keywords: FPGA, configuration, Radio, Beamforming, Astronomy
Copyright 2007, Brian S. Martin

Stream Communication and Computation in the Eight-meter-wavelength Transient Array (ETA) Radio Telescope

Brian S. Martin

(ABSTRACT)

The Eight-meter-wavelength Transient Array (ETA) system is a unique implementation of an array-based radio telescope. The instrument is designed to further astronomy by detecting and characterizing dispersed pulses received between 29–47 MHz. To aid data processing of radio signals received through 24 antennas, the ETA system performs real-time stream processing as data is passed from antennas to hard disk storage. The processing includes digital sampling, downconversion, filtering, Fast Fourier Transforms, and beamforming operations and is performed by 28 commercial-off-the-shelf (COTS) FPGA boards. Sixteen of the FPGA boards constitute the reconfigurable computing cluster (RCC) which performs the FFT and beamforming operations and is the focus of this thesis. The FPGA-based architecture allows the RCC to provide the high computational and communication throughput required by the ETA system. In addition, the FPGA design allows for a custom processing data path, parallel processing, global synchronization, and rapid development at a low cost.

This work was supported by the National Science Foundation under Grant No. AST-0504677, and by the Pisgah Astronomical Research Institute.

Acknowledgments

Thanks to Dr. Cameron Patterson, my academic advisor and committee chair, for his support throughout my time at Virginia Tech. It is through his help and guidance that I was welcomed into the Virginia Tech CCM Lab and joined the ETA project.

Thanks to Dr. Steven Ellingson and Dr. John Simonetti, my committee members, for their guidance throughout my work on the ETA project.

Thanks to the members of the CCM Lab who helped and befriended me during my time at Virginia Tech, including Eric Lorden, Joshua Edmison, Alexander Marschner, John Bowen, Timothy Dunham, Jorge Suris Pietri, Stephen Craven, and Matt Shelburne.

Thanks to my Parents Jerry David and Leslie Ann Martin who raised me and have always encouraged me to excel in life.

A special thanks to my wife and companion Christine Martin for her continued love, support, and encouragement.

Thanks to the National Science Foundation for supporting this work under Grant No. AST-0504677.

Thanks to the Pisgah Astronomical Research Institute and personnel for providing a location for ETA and constructing the antenna masts and underground coaxial cable system.

Thanks Vince Eck, Rick LeBlanc, Punit Kalra, and the Xilinx Systems Engineering Group for providing early access to the ML310 FPGA development board.

Contents

List of Figures viii

List of Tables xi

1 Introduction 1

1.1 Motivation 1

1.2 Contributions 2

1.3 Thesis Organization 3

2 Background 4

2.1 Radio Astronomy 4

2.1.1 Radio Frequency Emissions 6

2.1.2 Dispersion and Dedispersion 6

2.1.3 Dipole Arrays 8

2.2 Algorithms 9

2.2.1 Fast Fourier Transform 9

2.2.2 Digital Beamforming 10

2.3 FPGAs 11
2.3.1 Xilinx Virtex-II Pro 2VP30 FPGA Resources 11
2.3.2 Development Tools 13
2.4 Application-Specific Processing 14
2.5 Custom Communication 15
2.5.1 LVDS Signaling 16
2.5.2 Parallel Communications 17
2.5.3 High-Speed Serial Communications 17
2.6 Clocking and Synchronization 20

3 System Overview **22**

3.1 Overview 23
3.2 RF Antenna and Amplifier Systems 25
3.3 Digital Receiver Boards 27
3.4 Reconfigurable Computing Cluster (RCC) 30
3.5 Acquisition PCs 31
3.6 Control PC 32

4 RCC Design Considerations **35**

4.1 System Modes 35
4.1.1 Raw Mode 36
4.1.2 Beamforming Mode 37

4.1.3 FFT Beamforming Mode 39
4.1.4 Test Modes 40
4.2 Networking 42
4.3 Synchronization 44
4.4 Remote Access to System Nodes 45
4.5 FPGA Resources 46

5 Reconfigurable Computing Cluster **47**

5.1 Overview 47
5.2 Raw Mode 48
5.2.1 Outer Nodes 50
5.2.2 Inner Nodes 51
5.2.3 Test and Validation 52
5.3 Beamforming Mode 53
5.3.1 Outer Nodes 54
5.3.2 Inner Nodes 58
5.3.3 Test and Validation 59
5.4 FFT Beamforming Mode 61
5.4.1 Outer Nodes 61
5.4.2 Inner Nodes 63
5.4.3 Test and Validation 64

6 Results and Analysis **66**

6.1 Partitioning 67

6.2 Synchronization 69

6.3 Computational Performance 71

6.4 Communication Performance 72

6.5 Resource Efficiency 74

6.6 Scalability 77

6.6.1 Scalability of Antennas 77

6.6.2 Scalability of Beams Formed 78

6.6.3 Scalability of Observation Time 79

6.6.4 Scalability Summary 80

7 Conclusions **82**

7.1 Summary and Contributions 82

7.2 Conclusions 85

7.3 Future Work 85

Bibliography **87**

List of Figures

2.1 Emitted electromagnetic pulse consisting of a broad range of frequencies. . . 7

2.2 Frequency spectra series over time. 8

2.3 Above: Shifted frequency spectra over time. Below: Reconstructed transient pulse. 8

2.4 Xilinx Virtex-II Pro 2VP30 FPGA architecture overview. 12

2.5 Single-ended vs differential signaling. 16

2.6 Digital transmission without pre-emphasis. 19

2.7 Digital transmission with pre-emphasis. 19

3.1 ETA system design overview. 23

3.2 ETA antenna stand displaying two orthogonal antennas. 24

3.3 Key dimensions of the ETA dipole-like antennas. 26

3.4 ETA antenna stand arrangement. 26

3.5 Uncovered preamplifier and balun circuit board. Coaxial cable (left) and dipole (right) connections are also shown. 26

3.6 Analog receiver for the ETA system. 26

3.7 The received signal spectrum at the ETA antenna terminals (solid line) and the expected galactic noise spectrum (dashed line). 27

3.8 The Altera Stratix DSP development board (S25) used as the digital receiver nodes in the ETA radio telescope. 28

3.9 Overview of the hardware design implemented on the Altera Stratix FPGAs. 29

3.10 Transfer function for the digital signal from the ADCs to the Stratix FPGAs. 33

3.11 Transfer function for the digital signal from the $Fs/4$ Shift Left block to the first Multirate FIR filter. 33

3.12 Transfer function for the digital signal from the Complex Tuner to the second Multirate FIR filter. 33

3.13 Transfer function for the digital signal from the second Multirate FIR filter to the output formatter and the RCC. 33

3.14 ML310 board and adapters. The two adapter boards provide connections to eight InfiniBand cables (bottom left adapter board),"Blue Ribbon" cable (bottom right adapter board, upper connection), and the EDT cable (bottom right adapter board, lower connection). 34

5.1 ML310 to S25 and EDT interface adapter board. top: ML310 connection, middle: MICTOR to S25 connection, bottom: EDT connection. 48

5.2 ML310 to InfiniBand interface adapter board. top: ML310 connection, bottom: eight Infiniband connections. 48

5.3 RCC raw mode system operation overview. 50

5.4 Raw mode hardware implementation overview of an RCC outer node. 51

5.5 Raw mode hardware implementation overview of an RCC inner node. 51

5.6 RCC beamforming mode system operation overview. 54

5.7 RCC beamforming mode outer node FPGA implementation overview. 55

5.8 Beamforming module implementation for the beamforming mode. Forms four beams from two antenna inputs. 56

5.9 Replicate by four input and output timing. 56

5.10 RCC beamforming mode inner node FPGA implementation overview. 58

5.11 Adder tree module implementation. 59

5.12 RCC FFT beamforming mode outer node FPGA implementation overview. . 62

5.13 Beamforming module implementation for the FFT beamforming mode. Forms four beams from two antenna inputs. 64

5.14 Beamforming module implementation for the FFT beamforming mode. Forms four beams from two antenna inputs. 65

6.1 Data processing flow for a single polarization. 67

List of Tables

6.1 Resource Utilization for the FFT Beamforming implementation. 74

6.2 Logic registers required by each implementation. 75

Chapter 1

Introduction

1.1 Motivation

Over the past century radio astronomy has been used to study the universe and has resulted in discoveries which shape scientific understanding and theory. The Eight-meter-wavelength Transient Array (ETA) radio telescope is designed to provide new insights into the study of electromagnetic waves emitted from objects in space. The ETA instrument is designed to detect electromagnetic emissions between 29–47 MHz, a frequency range which until recently has received little attention from radio astronomers [1] [2]. The ETA radio telescope is capable of detecting transient pulses which may be emitted from the self-annihilation of primordial black holes (PBHs) [3], prompt emissions associated with gamma ray bursts (GRBs) [4], explosions, and other unexplored astronomical events. The study of these transient pulses may further research and theory regarding astrophysics, black holes, and extra spatial dimensions [5] [6].

ETA is able to observe nearly the entire sky and remain sensitive enough to be limited by the radio frequency (RF) noise background of the universe. These features are important in detecting transient pulses from unknown sources across the sky. In order to detect a transient

pulse, a large amount of processing must be applied to the received RF signal. The ETA system is capable of performing a portion of this processing in real-time as data is streamed from dipole-like antennas to PC hard disk storage. ETA is also designed from consumer off-the-shelf (COTS) components, allowing for low-cost and rapid construction which is scalable to future technologies. The design and development of the ETA radio telescope is one of the first systems completed with the intent to observe transient pulses and will further research and theory in physics and radio astronomy.

1.2 Contributions

Presented in this thesis is an overview of the ETA system design and a detailed description of the ETA real-time computation network. The main focus will be on the computation network which is comprised of sixteen FPGA (Field-Programmable Gate Array) development boards connected with point-to-point gigabit serial links. The collection of development boards and interconnecting network is referred to as the reconfigurable computing cluster (RCC) and is capable of performing large amounts of computation while data is streamed through the ETA system. In developing the RCC several important design factors were considered including data processing mode and computation, synchronization and partitioning, communication resources and networking. The RCC implementation demonstrates how processing can be distributed across a collection of off-the-shelf FPGAs to perform real-time and application specific processing.

FPGAs are well suited to data streaming and computationally intensive tasks because internal hardware components and communication devices can be configured to process data as it passes through the system. Even though general purpose processors and digital signal processors (DSPs) operate at much higher clock frequencies, they do not allow for custom hardware processing or communication elements in the data path. Application specific integrated circuits (ASICs) allow for custom processing but are costly and require excessive

development time. The RCC processes data from 24 antennas and performs complex operations such as beamforming and fast Fourier transforms (FFTs). The RCC is capable of performing 66 GMACs/s (giga-multiply-accumulates per second) during burst operation, but normally operates at 7.9 GMACs/s average. ETA demonstrates that in a short time an FPGA cluster can be developed to perform meaningful streaming computation at a low cost.

The FPGA cluster is well equipped to handle the computational load and allows for global and local synchronization throughout processing of the antenna inputs. ETA uses many techniques to source, process, and store data synchronously, creating a systolic system which may be difficult to replicate in other technologies. Ample routing and communication resources available in FPGAs also make them desirable for the RCC design. In addition to computational ability, the RCC is capable of streaming large amounts of data between RCC nodes and in or out of the RCC itself. FPGA Multi-Gigabit Transceivers (MGTs) along with simple adapter boards allow the RCC to have sufficient connectivity and bandwidth. The RCC network streams data at 2.8 GBs/s which is roughly half the full throughput capacity. The RCC implementation demonstrates how synchronization, connectivity and high throughput can be maintained across a multi-node stream-processing system.

1.3 Thesis Organization

This thesis is organized into seven chapters. Chapter 1 presents the motivation and contributions of this work. Chapter 2 discusses background and related information used in the thesis. Chapter 3 provides a system level description of the ETA radio telescope. The remainder of the thesis focuses on a more detailed description and analysis of the RCC design. Chapter 4 reviews major design considerations as they apply to the RCC. Chapter 5 gives an in-depth description of the RCC implementation, testing, and functionality. An analysis of the RCC system design is provided in Chapter 6. Lastly, Chapter 7 summarizes the accomplishments and discusses future work.

Chapter 2

Background

This chapter discusses relevant background material pertaining to ETA's purpose and system design. A portion of the background material is specific to the development of the reconfigurable computing cluster, which performs real-time processing and is the focus of this thesis. Section 2.1 provides a brief introduction to radio astronomy, radio frequency emissions, dispersion, dedispersion, and dipole arrays. Section 2.2 introduces the algorithms which are used to process received radio signals including the Fast Fourier transform (FFT) and digital beamforming. Section 2.3 describes basic FPGA device resources and development tools with particular emphasis on the devices used in the RCC implementation. Sections 2.4, 2.5, and 2.6 discuss the how the use of FPGAs facilitate application specific processing, custom communication, and synchronization within the ETA system.

2.1 Radio Astronomy

Astronomy has existed for generations as a means of searching for answers to probing scientific and theoretical questions. Commonly people picture astronomers looking through optical telescopes to pear into the vast expanse of space. However astronomy has evolved

much beyond the study of what is visibly seen. Astronomical objects and events emit electromagnetic waves or pulses. These waves can be caused by many events ranging from normal molecular interactions to large explosions. A small portion of electromagnetic waves fall into the visible light spectrum and have been seen or studied with optical devices. Unseen electromagnetic waves also travel through the sparse gas of space, known as the interstellar medium, to arrive at the earth. Emitted electromagnetic waves span a large range of frequencies and are studied to further science and astronomy. The study of electromagnetic waves emitted from celestial objects is known as radio astronomy [7].

An expanded view of the universe and discoveries about physics under extreme conditions have resulted from the study of emitted electromagnetic waves. Radio astronomy has resulted in the discovery of pulsars, quasars, and radio galaxies. New theories of the universe and its formation have also been established. Until recently, a majority of radio astronomy has been performed with large reflector or dish antennas [7]. At frequencies below 100 MHz, the required size of traditional dish antennas becomes unmanageable. In addition, traditional radio telescopes have been more focused on looking at a particular object or region in space rather than doing broad surveys over the entire sky. A new class of radio telescopes, consisting of many low-cost dipole antennas, are being developed to address these issues [2].

ETA uses multiple, inexpensive dipole-antennas to create a system sensitive enough for radio frequency observations between 29–47 MHz. Other low-frequency dipole arrays which function on a much larger scale are also under construction. The Low Frequency Array (LOFAR), located in the Netherlands, is in the first stage of construction. The first LOFAR site performed initial observations using 96 antennas in October of 2007 [8]. The Long Wavelength Array (LWA), planned to be constructed in New Mexico, and the Murchison Widefield Array (MWA), which is to be constructed in Australia, are working with demonstrator arrays and prototyping systems [9] [10]. While the scale of the ETA radio telescope is much smaller in size and cost, ETA is capable of providing relevant contributions to the field of radio astronomy by sensing and characterizing received transient electromagnetic waves.

2.1.1 Radio Frequency Emissions

Electromagnetic waves are emitted from all objects both in space and on earth. Electromagnetic energy emitted in the infrared frequency range can be sensed as heat, while energy emitted in the visible light spectrum can be viewed in a range of colors. Through the use of antennas and electronic equipment the ETA radio telescope is able to sense energy from similar electromagnetic waves at frequencies between 29–47 MHz. Any emission in this frequency range not emitted from astronomical events is viewed as radio frequency interference (RFI) or noise. Radio frequency noise limits observation sensitivity and can be caused by a variety of sources both terrestrial and galactic. The background noise of the universe at 29–47 MHz is significant and can limit the sensitivity of a receiving radio system. A system which is limited by the RF noise of the universe, rather than instrumentation noise, is referred to as being Galactic noise-limited [2]. Large-scale astronomical events or explosions are of particular interest as they release large amounts of electromagnetic energy instantaneously and across a broad range of frequencies. Current scientific theory supports that detectable transient events of this nature should occur on a regular basis [3] [4] [5].

2.1.2 Dispersion and Dedispersion

The energy spectrum produced from an instantaneous electromagnetic pulse is composed of a broad range of frequencies and is shown in Figure 2.1. A pulse generated in distant space must propagate through the interstellar medium to arrive at the Earth. The ionized components of the interstellar medium cause the wave velocity to be frequency dependent, with higher frequency signals traveling faster than lower frequency signals. As result, the pulse is broadened, and over sufficient distance, can be undetectable unless it is reconstructed. The dispersed signal has a unique frequency signature that can be used to reconstruct the original pulse. The spreading of an electromagnetic pulse caused by a frequency dependent propagation speed is known as dispersion. The amount of dispersion which takes place is used to determine a dispersion measure (DM). The dispersion measure along with galaxy models

can be used to estimate distance or draw conclusions about the event and the interstellar medium through which a pulse traveled [11].

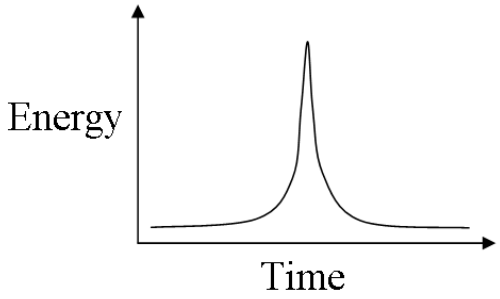

Figure 2.1: Emitted electromagnetic pulse consisting of a broad range of frequencies.

Due to the large distance traveled, the dispersed pulse may be difficult to notice or detect unless the signal is reconstructed. The process of reconstructing a dispersed electromagnetic pulse is called dedispersion. To dedisperse a received radio signal, many samples must be taken over the duration of the dispersed signal. The samples are then divided into sequential blocks of data on which a Fourier transform is performed. The Fourier transform is used to determine the amount of energy contained in granular frequency bands or bins. Stacking many of these frequency snapshots over time is the first step in dedispersing a transient pulse. Figure 2.2 provides an exaggerated view of what a sequence of frequency spectra may look like over time. The frequency spectra are then shifted relative to one another and summed to produce the original pulse. An example of shifted spectra and reconstructed pulse are shown in Figure 2.3. In practice, dispersed pulses are much less evident and are hidden amid noise and interference. Signal processing and additional filtering is required produced a dedispersed signal. To add to the difficulty of identifying a pulse, the amount by which the spectra should be shifted is unknown. Unless the signal is shifted by nearly the correct amount, the reconstructed signal will appear as noise.

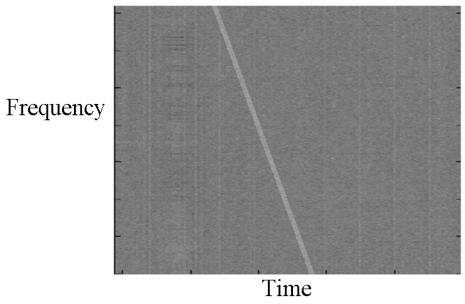

Figure 2.2: Frequency spectra series over time.

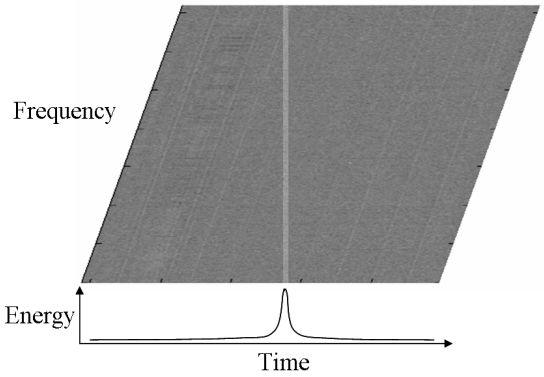

Figure 2.3: Above: Shifted frequency spectra over time. Below: Reconstructed transient pulse.

2.1.3 Dipole Arrays

Traditional dish antennas must increase in size as they become sensitive to lower frequency radio signals. It is expensive and impractical to construct such antennas to operate at frequencies below 100 MHz. Dipole arrays however can be used to construct inexpensive antennas which operate at low frequencies and have a large field of view [2]. A dipole antenna is created by positioning two center-fed conductors in a line. Positioning multiple antennas in a grid-like pattern and combining their received signals can produce collective signal sensitivity. Processing can be done on the multiple antenna feeds to produce accurate measurements and focus observations to a specific region of the sky. Combining the received signal from multiple dipole antennas typically requires a significant amount of processing.

2.2 Algorithms

2.2.1 Fast Fourier Transform

The Fast Fourier Transform is one of the key mathematical operations used in the ETA radio telescope processing chain. The Fourier Transform is used to transform a time varying function into a frequency varying function and is defined by the equation below where f is frequency, t is time, $x(t)$ is a time varying function, and $X(f)$ represents the frequency varying Fourier transform of the input function $x(t)$ [12].

$$X(f) = \int_{-\infty}^{\infty} x(t)\ e^{-i2\pi ft}\, dt \qquad \textit{for every real number } f \tag{2.1}$$

An altered form of this equation is used to produce a discrete Fourier transform (DFT). The DFT is computed on a series of N time-samples, rather than on a continuous input function, and produces a series of N frequency-samples. The discrete Fourier transform is defined by the equation below where N is the number of sequential samples on which the DFT is performed, $x(n)$ is a series of N complex time-samples and $X(k)$ is the DFT of the input sample stream $x(n)$. $X(k)$ is comprised of a series of N complex frequency-samples. A DFT which is computed on N time-samples is referred to as an "N-point DFT" [12].

$$X(k) = \sum_{n=0}^{N-1} x(n)\ e^{-\frac{2\pi i}{N}kn} \qquad k = 0, 1, \ldots, N-1 \tag{2.2}$$

The fast Fourier transform (FFT) is an algorithm which can be used to calculate the DFT. The Cooley-Tukey algorithm is the most common FFT algorithm used for calculating the discrete Fourier transform and is used in ETA. The Cooley-Tukey algorithm efficiently utilizes processing resources and storage elements to produce the DFT from a series of input samples [13]. For the purpose of this thesis, an N-point FFT transforms a series N of time-samples into a series of N frequency components which are contained within the original sample sequence. The FFT used in processing for the radio telescope is a 1024-point FFT. Thus

1024 samples, taken at successive time intervals, provide the input to the FFT operation and the resultant output is 1024 frequency-samples. Each sample spans a small frequency band or bin of about 3.7 kHz. The transformed frequency-samples are combined with others and used for the dedispersion computation.

2.2.2 Digital Beamforming

The process by which multiple antenna signals are combined to select a region of the sky is called beamforming. Spatial beamforming is done by multiplying complex time-sampled data with a complex weighting coefficient and then summing the products across M available elements. A simplified form of the beamforming equation is given below where M is the number of elements, $x_m(n)$ is a sample stream from the m^{th} antenna, w_m is the weighting coefficient for the m^{th} antenna, and $y(n)$ is the beamformed or combined sample stream. The weighting coefficient is used to scale the magnitude of each input stream and to select or steer the beam direction [14].

$$y(n) = \sum_{m=1}^{M} x_m(n) \cdot w_m \tag{2.3}$$

Frequency domain beamforming combines equivalent samples of adjacent antenna data-streams after an FFT has been performed on the initial time-sampled data. Frequency domain beamforming also requires that each of the frequency bins produced from the FFT be multiplied by a separate weighting coefficient. The frequency domain beamforming equation is given below where $X_m(k)$ is the N-point FFT of the complex input stream $x_m(n)$, $W_m(k)$ is the complex weighting coefficient vector corresponding to each antenna and k^{th} frequency bin, and $Y(k)$ is the beamformed or combined frequency series. The weighting coefficients are used for scaling the magnitude of individual antenna bins and for selectively filtering frequency bins [14].

$$Y(k) = \sum_{m=0}^{M-1} X_m(k) \cdot W_m(k) \qquad for\ k = 0, 1, \ldots, N-1 \tag{2.4}$$

2.3 FPGAs

Field Programmable Gate Arrays are semiconductor devices which contain reconfigurable logic and interconnect. Unlike other semiconductor devices, FPGAs do not have a defined application at the time of manufacturing. The device's configurable resources can be used to implement custom logic or communication functions to fit a specific application. While the ETA system uses FPGAs from both market leaders, Xilinx [15] and Altera [16], the discussion of FPGAs will be focused toward the Xilinx Virtex-II Pro 2VP30 FPGA and associated development board as this is the device used in the reconfigurable computing cluster. FPGAs are well suited to the ETA radio telescope design because they allow for rapid development, flexible architectures, real-time stream processing, application specific computation, and custom communication.

2.3.1 Xilinx Virtex-II Pro 2VP30 FPGA Resources

A majority of an FPGA is comprised of configurable logic elements which consist of a look-up-table (LUT), a flip-flop, and associated logic. From this basic building block and the configurable region and interconnect can be used to implement complex stream or state-based processing. In addition, most FPGAs also provide dedicated resources for commonly used functions such as multipliers, DSPs, microprocessors, storage elements, clock management devices, high speed transceivers, and other I/O resources. The 2VP30 FPGA is one of many devices produced by Xilinx and is discussed in this section [17].

The 2VP30 FPGA is equipped with 30,816 logic elements, 136 18×18 bit multiplier blocks, 2 Power PC microprocessors, 136 dual-port 18-kbit memory elements (BRAMs), 8 high speed

serial transceivers (MGTs), 644 user I/O pads, and 8 digital clock managers (DCMs). An architecture overview of the 2VP30 FPGA is shown in Figure 2.4.

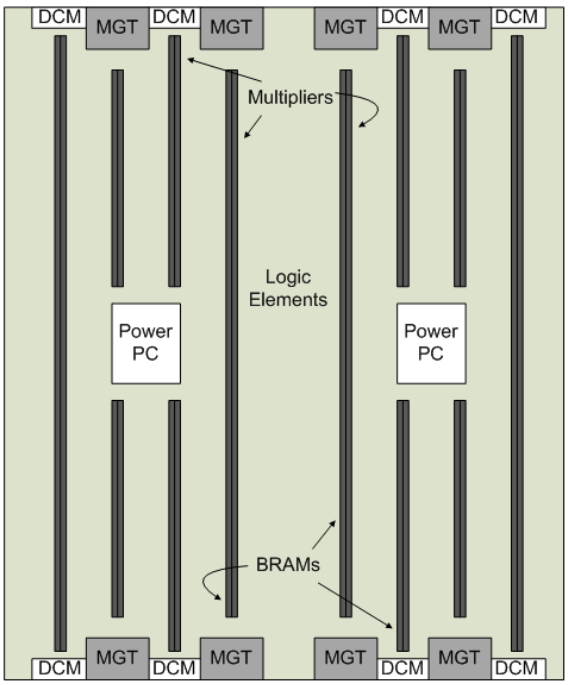

Figure 2.4: Xilinx Virtex-II Pro 2VP30 FPGA architecture overview.

While not all of these elements are used in ETA's FPGA implementations, the FPGAs provide sufficient processing and communication resources for the RCC computation. Multiplier blocks are used to perform the complex multiplication required by the FFT and beamforming computations. The BRAMs are used to store partial computations, compensate for communication protocol delays, and provide adequate buffering when crossing clocking regions. The MGTs are used to perform the high-speed serial communication between other RCC nodes. The remainder of FPGA design are implemented in the configurable logic elements and routing. The two PowerPC processors are not used in the RCC implementation.

2.3.2 Development Tools

Xilinx also offers a suite of development and debugging tools which were used in the RCC design and implementation. The Xilinx tools aid designers in developing and debugging FPGA-based systems. The tools used in the development of the RCC include ISE, CoreGen, and ChipScope Pro.

ISE is a development environment provided for Xilinx FPGAs. It allows projects to be managed, updated, compiled, and mapped to FPGA resources more readily. ISE is used in the design of the RCC to implement the RCC hardware [18].

CoreGen is another Xilinx utility which is normally included with the ISE installation. The utility allows a user to create common FPGA elements that can be imported into an FPGA design. CoreGen facilitates the RCC design process by creating processing, communication, and storage blocks which are easily instanced in FPGA designs and are referred to as IP blocks. Specifically CoreGen was used to create the FFT, complex multiplication, summation, Aurora communication, and single-port or dual-port buffers. The use of generated IP blocks greatly decreased the design time required to create the RCC hardware implementations [19].

ChipScope is the final Xilinx utility discussed in this section. The ChipScope development tool is used for viewing the internal signals and timing of an FPGA design. After a design is created, a ChipScope module can be added to the design and connected to user-specified signals. After the design is downloaded to the FPGA, the ChipScope software interface allows a designer to view the waveforms of the selected signals in a logic analyzer format. Trigger conditions can be set and allow users to view the occurrence of relevant events. ChipScope is used in the RCC design for debugging and timing verification [20].

2.4 Application-Specific Processing

FPGAs can be configured to perform custom processing focused toward a specific application. While ASICs can also be tailored to a specific application, they are more costly in terms of production and development time. Furthermore ASICs offer a static design which can not be modified, updated, or changed. FPGAs provide for rapidly developed, flexible, and custom processing implementations at an off-the-shelf cost. When compared to general purpose processors or digital signal processors, FPGAs operate at a much lower clock frequency. Even at lower clock rates, FPGAs can provide improved performance through increased parallelism, custom processing blocks, and application specific data paths [21] [22] [23].

Processing in the RCC is well suited to FPGAs because the operations performed are highly parallel. Processing multiple antenna data streams requires many multiply and accumulate operations. These operations are simple and form the building blocks of digital signal processing. Even though general purpose processors and DSPs use dedicated logic blocks to perform multiplication and accumulation, neither device can customize the number of elements or data flow to a specific application. FPGA solutions can implement as many multipliers or accumulators as needed and mold the processing data path to fit the processing algorithm. The increased parallelism achieved with FPGAs gives them a distinct processing advantage over general purpose and digital signal processors.

FPGAs also improve the RCC performance by enabling custom processing blocks to be added to the system. Repetitive, frequently used, or complicated operations can be optimized into a single module and incorporated into a design. The optimized hardware can be instanced multiple times in a design to speed or parallelize processing. Traditional processors and cluster-based processing rely on the optimization of software algorithms to improve performance. Alternatively, FPGAs optimize the actual hardware to accelerate frequently used or time-intensive operations. The computation required to perform complex multiplication and 1024-point FFTs are two examples. Designated blocks for multiplication and data storage are also used to further increase performance and utility within an FPGA design. Tailoring

logic to the radio telescope application enables stream processing in the RCC design.

FPGAs are also effective at implementing custom data paths which can increase throughput and are essential to the RCC implementation. As mentioned, traditional processors do not have a method of processing a stream of data through a series of computational blocks. Instead intermediate results must be stored and processed sequentially. One of the benefits of FPGA-based designs is that the entire data path and system architecture can be designed around the processing performed. In the radio telescope application, multiple antenna data streams feed into processing nodes and are combined with samples from adjacent data streams. Thus the RCC architecture is constructed to reflect this processing path. Internally each FPGA passes data directly from one processing block to the next rather than re-processing it in a single block. The ability to create an architecture and processing path which reflect data flow through the system make FPGAs a desired platform for the RCC design.

2.5 Custom Communication

To develop custom processing data paths and architectures, FPGAs provide ample communication resources. Large numbers and a variety of I/O options allow FPGAs to be used effectively in many multi-device communication topologies. In addition to processing capability, ETA also requires routing and communication resources to enable stream-processing. Low-voltage differential signaling (LVDS), parallel bus, and high-speed serial transceivers are used in the ETA design to enable high-bandwidth streaming communication and to improve data flow through the system. Xilinx Virtex-II Pro 2VP30 FPGAs enable the communication through the use of many LVDS parallel and high-speed serial connections. This section explains each of these communication methodologies in more detail.

2.5.1 LVDS Signaling

Single-ended data transmission occurs by transmitting the desired signal through one conductor and using another conductor for ground or common terminal. Single-ended connections are susceptible to noise and interference from external and adjacent transmissions. In addition, signal degradation over distances can be severe and limit the effectiveness of transmission links. Differential signaling provides more stable and noise-resistant communications by transmitting the desired signal through one conductor and sending the inverted form of the transmitted signal on the adjacent conductor. The two conductors, through which the signals pass, are tightly coupled through PCB, connectors, and twisted pair cables to produce a more robust transmission scheme [24]. Figure 2.5 shows the transmission produced using single-ended and differential signaling.

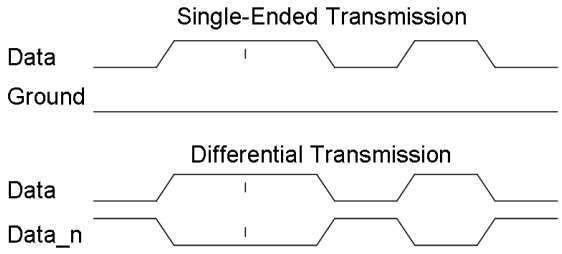

Figure 2.5: Single-ended vs differential signaling.

Differential signaling is well known to produce stable communications, resistant to noise and signal degradation because of close coupling and similar interference effects between lines. The added stability of differential signaling allow transmissions to function with a lower voltage difference or swing. The increased stability and decreased voltage swing enable signals to transition quicker and operate at higher transmission frequencies. The 2VP30 FPGA provides many LVDS I/O connections to allow data to be transmitted between system components. Low-voltage differential signaling is used in the ETA design to produce stable high-frequency data communication links [24].

2.5.2 Parallel Communications

To increase communication bandwidth, additional data transmission lines can be added to an interface. Adding data lines to a communication link and maintaining the same control signals creates parallel or bus communication. For chip-level, board-level, and some system-level designs, parallel communication enables increased data throughput by sending data in parallel rather than sequentially. PCI and memory buses are well known examples of parallel communication. While parallel communication offer greater throughput compared to a single connection, it is not without its drawbacks. Wide parallel bus connections can be bulky and physically difficult to manage as transmission distances increase. Links also operate at lower clocking rates to compensate for differing delay effects between data lines. ETA uses parallel communication coupled with LVDS signaling to produce robust communication to and from the RCC.

2.5.3 High-Speed Serial Communications

High-speed serial transmissions often utilize LVDS signaling to enable high-bandwidth links by increasing the signal rate. High-speed serial links can operate at higher frequencies because synchronization between data or clock lines is not required. The signal clock is recovered from the serial transmission requiring frequent signal transitions. Furthermore an equal amount of logic-high and logic-low values must be transmitted to maintain a DC-balance and reduce inter-symbol interference. Serial communication has developed to operate at clock frequencies much higher than parallel communication. As serial transmission rates increase, serial communication is becoming preferred over parallel communication. Serial links require fewer conductors and are therefor cheaper and more easily managed. In the consumer market the shift from parallel to serial communication can be seen with the development of the PCI Express and SATA interfaces, which are comprised of a number of serial communication links rather than a parallel bus. ETA uses high-speed serial connection between the processing elements of the RCC to achieve increased throughput using fewer transmission lanes.

To enable high-speed serial communication the 2VP30 FPGA is equipped with eight RocketIO Multi-Gigabit Transceivers. The transceivers use an LVDS-like serial transmission to produce high-throughput data links. The Virtex-II Pro MGTs are capable of data transmission up to 3.125 Gbits/s which is sufficient for RCC communication. The RocketIO MGTs also offer options for fine tuning the high-speed data links including adjustments to the differential voltage swing, pre-emphasis, and 8b/10b encoding. These tuning features along with the Aurora protocol, used for the serial links and provided as Xilinx IP, are described below.

Differential swing is the voltage difference between a logic-high and a logic-low signal. The differential voltage of the Xilinx Virtex-II MGTs can be set to values from 400 mV to 800 mV depending on the application and transmission line length. Typically longer transmission lines required a higher voltage swing to compensate for signal degradation along the transmission line. A higher differential voltage also requires a greater transition time and can limit the operating frequency. The optimal differential voltage is application dependent but should provide adequate signal quality at the desired frequency. The differential voltage is adjusted on the RCC high-speed serial links to improve signal quality between FPGA devices [17].

As digital signals propagate through a transmission line, high-frequency components associated with signal transitions attenuate more than lower-frequency components. At transmission frequencies over 1 GHz, the added distortion may prevent correct signal reception. Pre-emphasis creates stronger transitions in a transmitted signal to compensate for the high-frequency attenuation which occurs in a transmission line. Properly applying pre-emphasis to signal transmissions improves the received signal quality over the link. Figure 2.6 shows a transmitted signal without pre-emphasis, and Figure 2.7 shows the signal with pre-emphasis added to increase the voltage swing at transition boundaries. The RocketIO MGTs provide pre-emphasis adjustments between 0% and 33%. Adjustments to pre-emphasis are used to improve signal quality of the RCC high-speed serial links [17].

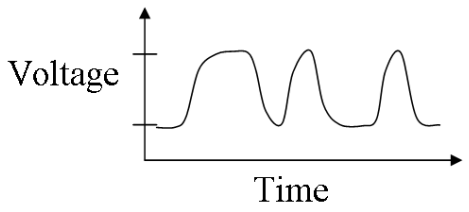

Figure 2.6: Digital transmission without pre-emphasis.

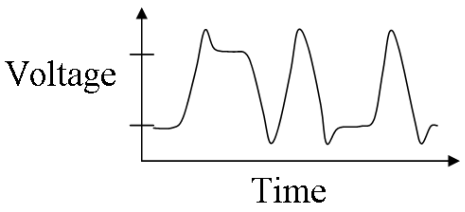

Figure 2.7: Digital transmission with pre-emphasis.

High-speed serial links require evenly distributed logic-high and logic-low values within the transmitted data. The even distribution creates a signal with little DC bias and frequent signal transitions. Maintaining a DC-balance across a transmission line reduces inter-symbol interference, and frequent transitions ensure that the signal clock can be recovered at the receiver. Since the ETA system data stream is not an evenly distributed signal, 8b/10b encoding is applied to the data stream converting every 8-bit sequence into a more evenly distributed 10-bit sequence. The 8b/10b encoding scheme also enables bit-error detection and allows for additional control characters. The RocketIO MGTs can apply 8b/10b encoding to guarantee the transmission of evenly distributed signals. The RCC high-speed serial links utilize 8b/10b encoding to maintain DC-balance and frequent transitions in transmitted signals [17].

Aurora Protocol

Aurora is a lightweight serial protocol developed by Xilinx and available as a CoreGen-produced IP block. Aurora is used as the RCC link layer protocol because it creates high-speed serial links between FPGAs with little protocol or configurable overheads. The Aurora protocol automatically initializes the transmission channel and sends idle characters to maintain the link and clock synchronization at the receiver. The protocol is implemented in streaming operation such that data may pass through the links as it becomes available. Periodically the transmission line becomes unavailable for a few cycles in order to transmit a

clock compensation sequence. The sequence is used to correct for small frequency differences between the transmitting and receiving end of a link. 8b/10b encoding is also applied to the transmitted signal to maintain frequent signal transitions and no DC bias. The Aurora protocol allows for a lightweight interface for communicating between FPGAs in the RCC design [25].

2.6 Clocking and Synchronization

Customizing the processing path to fit system data flow allows for both data streaming and synchronization with little overhead. Processing can be performed in parallel and expanded to perform the required processing as data passes through the system. Processing paths can also be constructed with equal or similar processing delays to facilitates synchronization throughout processing. Synchronization is maintained in ETA to produce a system which is system, source, and self synchronous. These aspects of synchronization are discussed below. System synchronous refers to global system synchronization while source and self synchronous provide local synchronization during communication.

System synchronization refers to how data is maintained in a global system. Global synchronization is achieved in the ETA system by sampling the analog antenna feeds at the same frequency and embedding synchronization data within transmitted signals. Slight differences in the sampling frequency can prevent system synchronization from being maintained. Channels which produce samples at slightly different rates cause drift between channels and inevitably result in the overflow of internal buffers. To maintain an identical sample rate across all channels, global clock and reset signals are distributed between all of the A/D converters on the S25 receiver boards. The distributed clock enables synchronous data sampling across all of ETA's antenna feeds. The distributed reset allows sample and frame counters to be produced identically across the multiple processing paths. These counters are used to synchronize samples before they can be combined with those of adjacent channels. The ETA

design is system synchronous because data is sourced, processed, and stored synchronously between global system components.

Source synchronous communication occurs when data samples are synchronized according to a clock and control signals transmitted with the data. Traditionally parallel bus communication transmits multiple data lines, control signals, and a clock making it source synchronous. The receiving end of these communications use the clock and control signals to determine sample boundaries and register incoming data. A dual-clock interface is commonly used to transition from the receiving clock to a local clock used in the design. Dual-ported BRAMs available in FPGAs facilitate the transfer of data between clocking domains within an FPGA implementation. Parallel, source synchronous communication is used in the ETA system design for communication from the S25 receiver boards to the RCC, and from the RCC to the PC data acquisition cards.

Communication which embeds the sample clock and control signals within transmitted data is referred to as self synchronous communication. Since high-speed serial protocols do not transmit a designated clock, the sample clock must be extracted from incoming data or recreated at the receiving end of the connection. The high-speed Aurora serial connections between RCC nodes produce self synchronous communications in the ETA system design.

Chapter 3

System Overview

The ETA radio telescope is designed to detect transient pulses received from celestial emissions. In order to observe unpredicted transient events, the telescope must have a large viewing area and be limited by the galactic background noise. It has been shown that sufficient sensitivity and viewing area can be achieved at low frequencies through the use of inexpensive dipole-like antennas [2]. The ETA system also aids data analysis by performing real-time processing of the received RF signal. System processing is performed by commercially available Altera and Xilinx FPGA development boards. The use of commercially available parts and FPGAs facilitate ETA system design and reduce development time. The system must also be capable of recording large amounts of data for an extended period of time. Data capture and system control must have the capability to be managed off-site as the ETA system is near Rosman North Carolina a five hour drive from the Virginia Tech campus. This chapter gives a brief overview of the entire system and then describes the individual elements in greater detail. The overview contained in this chapter does not discuss the RCC in detail but is meant to proved a basis for further discussion of the RCC in Chapters 4 and 5.

3.1 Overview

As described, the ETA radio telescope is designed for low-frequency operation between 29–47 MHz, full sky observation, high throughput, real-time digital signal processing, and continuous remote operation. Figure 3.1 provides a graphical representation of the ETA system architecture as described in this section. The system is comprised of 24 dipole-like antennas and feed systems, 24 analog receivers, twelve digital receiver Altera FPGA boards, sixteen Xilinx FPGA boards, four acquisition PCs, one control PC (not shown in Figure 3.1), and interconnect.

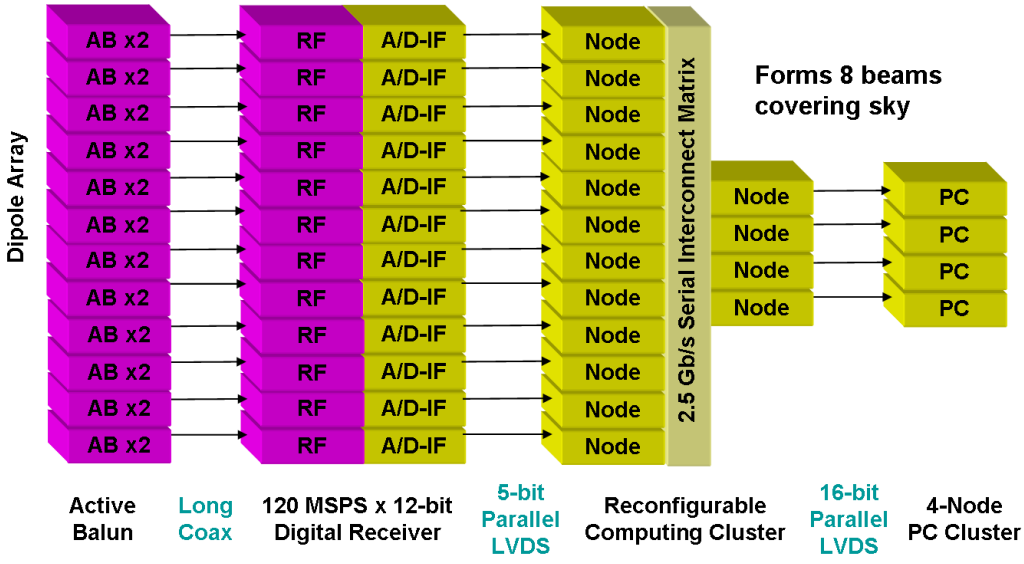

Figure 3.1: ETA system design overview.

The array of 12 dual-polarized dipole-like stands receive radio frequency signals. Figure 3.2 shows one of the ETA antenna stands. Each stand consists of two orthogonally placed active antennas with corresponding feed systems. The two orthogonal polarizations provide a total of 24 radio frequency inputs to the system. The feed system for each of the 24 antennas is located inside the protective PVC tubing and is comprised of a preamplifier, balun, and coaxial cable. The coaxial cable carries the signals underground to a central building and

connects to the analog receivers. The 24 analog receivers amplify and filter the dipole inputs before they are converted to digital signals in the digital receiver nodes.

Figure 3.2: ETA antenna stand displaying two orthogonal antennas.

The twelve receiver nodes are implemented on Altera Stratix DSP development boards and are referred to as S25s [26]. Each S25 digitally samples two antenna feeds at 120 Msps. The digital signal is then processed in the Stratix EP1S25 FPGAs to perform the digital filtering, down conversion, and multiplexing of two antenna data streams. The multiplexed data stream is then passed through a source synchronous 5-bit parallel connection to the reconfigurable computing cluster (RCC).

The RCC is a two-level network of sixteen Xilinx ML310 FPGA development boards, referred to as ML310s [15]. Twelve RCC outer nodes receive digital data streams from the S25 boards, perform the required digital signal processing, and pass a resulting data streams to two of four RCC inner nodes. The network between RCC outer and inner nodes is formed using the Aurora protocol over InfiniBand cables. Aurora provides a light-weight, high speed, self synchronous, serial protocol well suited to data streaming in the ETA architecture. The RCC inner nodes complete the required signal processing, provide additional filtering, and output data for acquisition to one of four PC acquisition nodes.

Each acquisition node is capable of recording data continuously for up to one hour. This results in approximately 200 GBs of data per PC or 800 GBs aggregate. Data is then

archived to tape for easy transport, analysis, and long-term storage off site. Not shown in Figure 3.1 is the control PC. The control PC allows the operation of the system to be observed, controlled, or modified remotely. The control PC connects to the four acquisition nodes through a Gigabit Ethernet switch and to the sixteen RCC nodes through two 8-port serial RS232 PCI cards. Through the RS232 interface, the RCC nodes can be controlled or monitored with a hardware UART controller. Users can log into the control PC from the Internet to manage system status and acquisitions. The remainder of this chapter describes the ETA system elements in more detail.

3.2 RF Antenna and Amplifier Systems

Low-frequency, transient pulse searches require antennas and feed systems that have a wide field of view and are limited in sensitivity by the radio frequency noise produced in the galaxy. This can be attained with a simple low cost inverted V dipole-like antenna and feed system [2]. The dipole antenna arms are constructed from 3/4-in (1.9 cm) x 3/4-in aluminum L-shaped angle stock that is 1/8-in thick. Angle stock was chosen because it creates rigid, low-cost antennas with the desired bandwidth and are easily constructed. The total length of a dipole, including both arms and feed gap is 3.8 m. This gives the dipoles a resonant frequency centered near the middle of the ETA frequency range at 38 MHz. The dipole arms are also bent down at a 45 degree angle which broadens the antenna pattern. Each of the twelve antenna stands has one antenna placed in a north-south orientation and another placed orthogonally in an east-west orientation. This results in twelve dual-polarized antenna stands. The antenna terminals are located at the top of a two meter high mast which corresponds to approximately one-quarter wavelength above the ground at resonance. The mast is constructed from PVC electrical conduit with a four inch diameter. The antenna dimensions are shown in figure 3.3 and the twelve stand arrangement is shown in figure 3.4. [27]

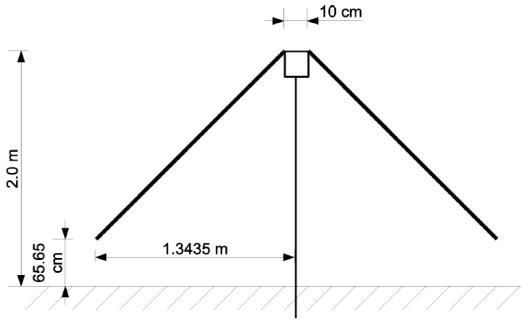

Figure 3.3: Key dimensions of the ETA dipole-like antennas.

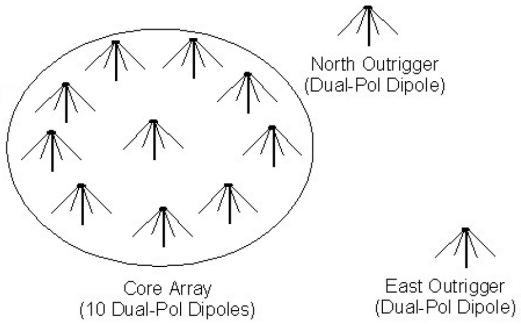

Figure 3.4: ETA antenna stand arrangement.

Each dipole is connected through a preamplifier and balun circuit to a coaxial cable inside the mast. Figure 3.5 shows the preamplifier and balun circuit [27]. The coaxial cable carries each signal to a central building and is then processed through an analog receiver. The analog receiver amplifies and filters the incoming signals and is shown in figure 3.6.

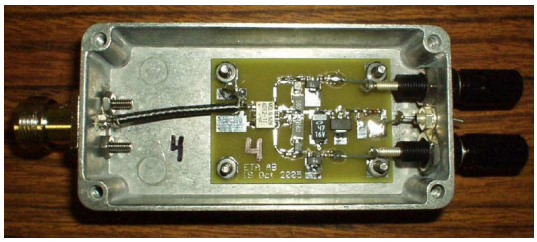

Figure 3.5: Uncovered preamplifier and balun circuit board. Coaxial cable (left) and dipole (right) connections are also shown.

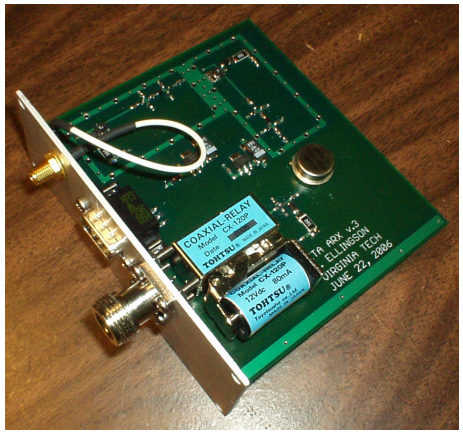

Figure 3.6: Analog receiver for the ETA system.

The antenna and feed system design has been tested and showed that it is galactic noise limited and that further design optimizations would provide little if any benefit [27]. Figure 3.7 shows the received signal at the antenna terminals. The figure shows areas of man-made RFI outside the 29-47 MHz band and that the in-band noise level follows the expected

galactic noise background curve.

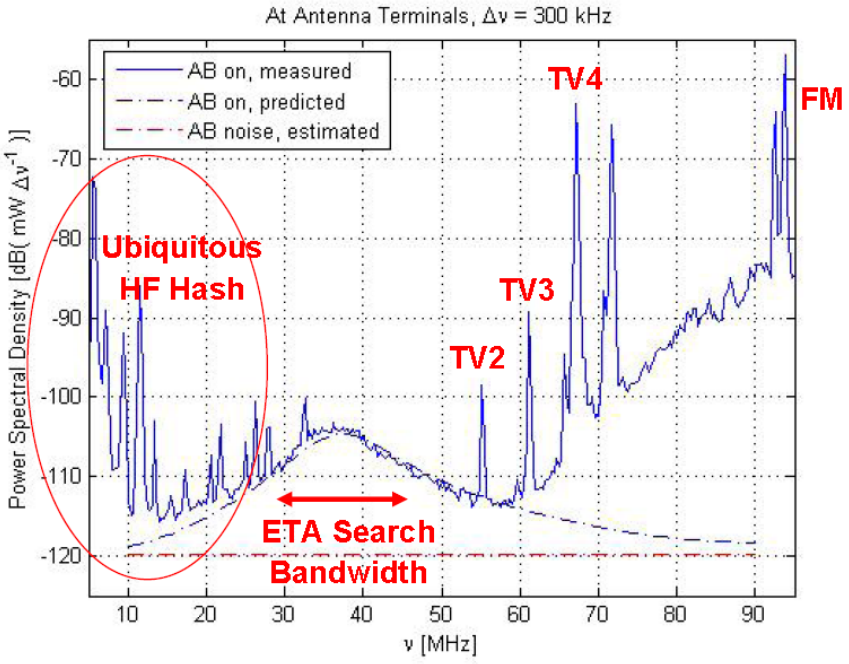

Figure 3.7: The received signal spectrum at the ETA antenna terminals (solid line) and the expected galactic noise spectrum (dashed line).

The simple antenna design functions well for the ETA system because the antennas and amplifiers can be produced easily and cheaply. The antennas achieve the maximum possible performance in that their sensitivity is limited by galactic noise over the frequency band of 29–47 MHz. From the output of the analog receivers, each signal passes to the digital receiver nodes where the signal is digitally sampled and processed further.

3.3 Digital Receiver Boards

The digital receiver nodes digitally sample, filter, and downconvert the analog antenna inputs. Since designing custom hardware to perform this function would be costly and time consuming, receiver functionality is implemented on Altera Stratix DSP development boards (S25s) [26]. The S25 is used because it contains the necessary hardware elements for the

design and is familiar to the design team. Each board contains two analog to digital converters (ADCs) capable of sampling at 125 Msps and can therefore process two antenna inputs. All 24 analog input streams can be digitally sampled using twelve S25 boards. The on-board Stratix EP1S25 FPGA [28] performs the necessary signal processing to filter and downconvert the signal. The board also contains ample I/O connectors for data transfers. The S25 board is shown in Figure 3.8.

Figure 3.8: The Altera Stratix DSP development board (S25) used as the digital receiver nodes in the ETA radio telescope.

The on-board ADCs in this design sample the analog stream at a sampling frequency (F_s) of 120 Msps. The digitally sampled signal is then sent to the Stratix FPGA for digital processing. Figure 3.9 summarizes the processing which takes place in the Stratix FPGA.

Figure 3.10 shows the transfer function of the signal sent to the Stratix FPGA from the ADCs. In the Stratix chip, the $F_s/4$ Shift Left block downconverts the digital signal to 1/4 of the sample frequency, or 30 MHz. This is done because downconversion by $F_s/4$ is a special case, easily implemented in hardware by changing the signs of the sampled data stream. This downconversion shifts the passband center frequency from 38 MHz to 8 MHz and maintains the sample rate of 120 Msps. The output transfer function from this block is

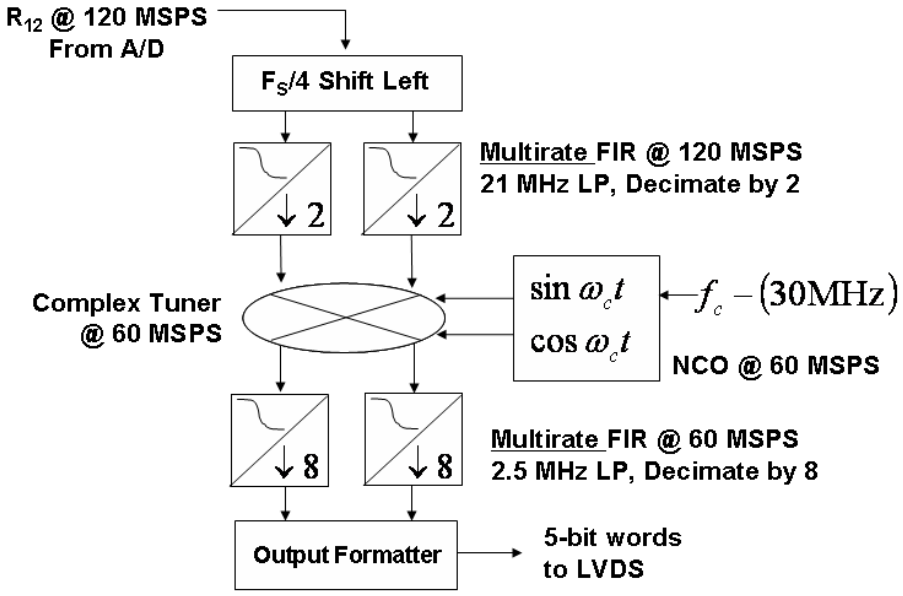

Figure 3.9: Overview of the hardware design implemented on the Altera Stratix FPGAs.

shown in Figure 3.11 and is then passed through a 21 MHz low-pass FIR filter which also reduces the sample rate to 60 Msps. The filter reduces aliasing and suppresses out of band signals which may be aliased into the baseband during subsequent processing steps. The signal maintains a center frequency at 8 MHz but has a sample rate of 60 Msps. A 60 Msps numerically controlled oscillator (NCO) and complex tuner are then used to shift the center frequency from 8 MHz to 0 MHz. The transfer function at the output of the complex tuner is shown in Figure 3.12 and passes through a second multirate FIR filter. The second FIR is a 2.5 MHz low-pass filter with an input sample rate of 60 Msps which is reduced to 7.5 Msps at the output. The transfer function at the output of the second FIR is shown in Figure 3.13 and provides a 5 MHz passband to the rest of the system.

Using this method any 5 MHz band between 29–47 MHz can be selected by making minor adjustments to the NCO. Since each S25 board process two data streams, a majority of this circuitry is replicated for a second input. At the end of this process the two signals are each represented using a complex 7-bit I, 7-bit Q value, giving 28 data bits which are

transmitted to the RCC outer nodes. The Output Formatter prepares data for transmission by multiplexing the two streams, a sample counter, and start of frame signal to low-voltage differential signal outputs. Five data bits along with a clock are transmitted in parallel at 60 MHz, or 37.5 MBs/s, over Precision Interconnect Blue Ribbon brand coaxial cable with MICTOR connectors [29] to RCC outer nodes. The mini-coaxial ribbon cables were chosen because they have excellent noise immunity, produce minimal RFI, and provide sufficient throughput.

Each S25 board performs its signal processing in parallel with the other S25s. It is important for the boards to synchronously source data at identical sample rates. If data is not sourced at the same rate, downstream synchronization buffers in the RCC will overflow and result in synchronization and data loss. To maintain a synchronous system, one S25 generates the clock and reset signals for all of the S25 boards. The result is increased clock skew between boards but enables each S25 board to output data at the same sample rate. Multiple data streams are then synchronized using the embedded counter and start of frame bits after they are received in the RCC nodes.

3.4 Reconfigurable Computing Cluster (RCC)

The reconfigurable computing cluster consists of sixteen Xilinx ML310 development boards (ML310s) [15] that are networked using InfiniBand cables [30]. Similar to the S25s, the Xilinx ML310 is used because it is familiar to the design team and provides sufficient hardware components and I/O connections. The nodes are divided between twelve outer nodes, which connect directly to the digital receiver nodes, and four inner nodes, which connect directly to the acquisition PCs. The purpose of the RCC is to implement signal processing, provide a reconfigurable network that can be modified as required, and reformat the data stream for acquisition. Figure 3.14 shows an ML310 board and the two custom adapter boards required to connect the S25s, RCC network, and acquisition PCs.

The ML310 boards use Xilinx Virtex-II Pro FPGAs which contain eight multi-gigabit transceivers [31]. The MGTs allow for up to eight InfiniBand connections to each board. The Aurora lightweight, point-to-point, serial protocol [25] is implemented over the InfiniBand connections. Using these links, the twelve RCC outer nodes stream processed data to the RCC inner nodes. The inner nodes combine data from multiple outer nodes and send the resultant data stream, through a 16-bit parallel connection, to the acquisition PCs. Each RCC node also communicates with the control PC to send and receive status or control information. The connections are made through the serial port using a UART and physical RS232 connection and allow the RCC nodes to be observed and updated through the control PC. A more detailed discussion of the RCC is provided in Chapters 4 and 5.

3.5 Acquisition PCs

The four acquisition PCs serve as temporary data storage which is offloaded to LTO-3 tape [32] once an acquisition is complete. To accomplish this, the PCs must be able to receive and store data at a high and continuous rate. Dell SC430 servers running Red Hat Enterprise Linux are used for the acquisition PCs. Each PC is equipped with an Electronic Design Team (EDT) PCI CDa LVDS/S600E [33] data acquisition card and two additional hard drives for rapid data storage. The EDT cards provide 16-bit parallel LVDS data transfers from the inner RCC nodes. To achieve higher continuous acquisition speeds, the hard drives are software configured as a RAID Level 0 partition. After testing it was found that the system could continuously record data at 60 MBs/s for one hour without dropping samples. One hour corresponds to the ETA viewing duration during which 200 GB of data will be recorded per acquisition PC. Alternatively, data can also be written at 30 MB/s for 2 hours in order to extend viewing time. After recording data to the PCs, the information is written to LTO-3 tapes for easy data removal, transport, and longterm storage. Each acquisition PC is also connected to the control PC through a Gigabit Ethernet switch. The connection allows the control PC to view system states, start or stop data acquisitions, and transfer

data to tape.[34]

3.6 Control PC

The Fedora Linux-based control PC is the central means of accessing, controlling, and observing the system. The computer can be accessed through the Internet and allows remote access to both the RCC nodes and acquisition PCs. To access the acquisition PCs, an additional Ethernet adapter card connects through an Ethernet switch to the acquisition PCs. The connections to the RCC nodes are made using two Lava Link Octopus-550 Eight Port Serial PCI cards [35] which were also added to the control PC. Each card allows for eight serial port connections giving a total of sixteen serial connections. Each serial port connects to one of the sixteen RCC nodes allowing each of them to be accessed from the control PC. Scripts run on this PC can update RCC nodes with new coefficients, check status information, and start or stop data acquisitions.

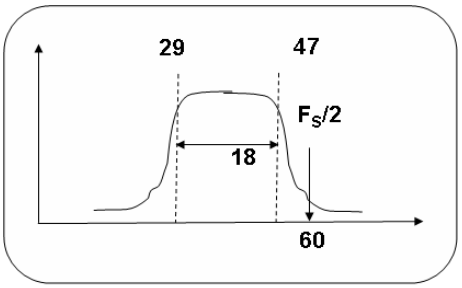

Figure 3.10: Transfer function for the digital signal from the ADCs to the Stratix FPGAs.

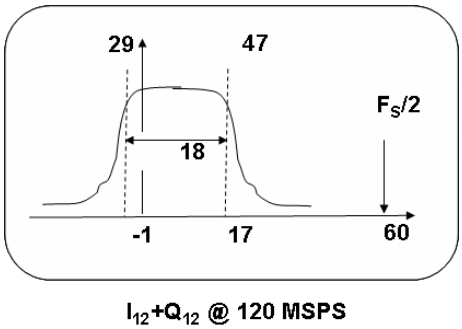

Figure 3.11: Transfer function for the digital signal from the $Fs/4$ Shift Left block to the first Multirate FIR filter.

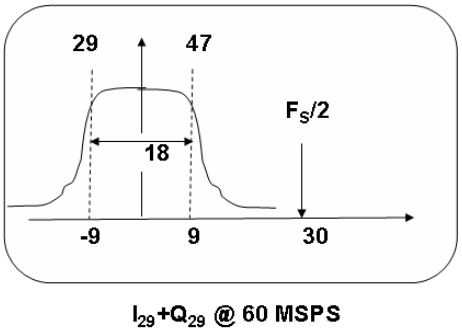

Figure 3.12: Transfer function for the digital signal from the Complex Tuner to the second Multirate FIR filter.

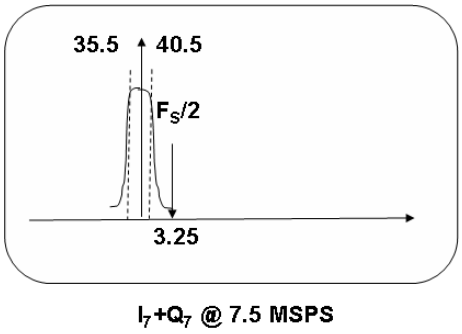

Figure 3.13: Transfer function for the digital signal from the second Multirate FIR filter to the output formatter and the RCC.

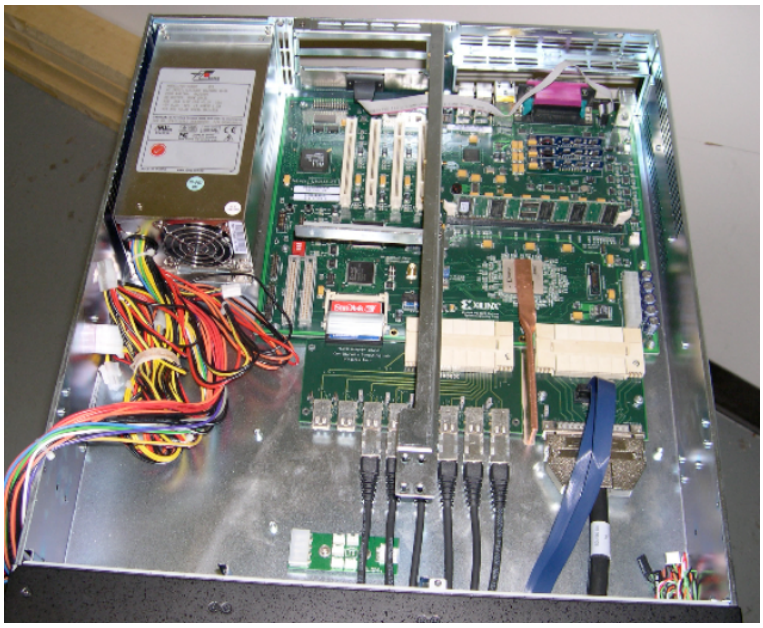

Figure 3.14: ML310 board and adapters. The two adapter boards provide connections to eight InfiniBand cables (bottom left adapter board),"Blue Ribbon" cable (bottom right adapter board, upper connection), and the EDT cable (bottom right adapter board, lower connection).

Chapter 4

RCC Design Considerations

The RCC provides real-time, synchronous, stream processing for the ETA radio telescope. The RCC nodes must be able to process the incoming antenna data streams continuously as data is passes through the system. RCC computation is performed by sixteen Xilinx FPGAs which are networked using high-speed serial interconnect using the Aurora protocol. FPGAs are used in the RCC architecture because they can perform custom, complex computation, are equipped with high-speed communication resources, and provide a rapid development environment. The function of the ETA radio telescope imposes a number of design consideration on the RCC. This chapter discusses how the system design is affected by multiple modes of system operation, networking, synchronization, remote access to system nodes, data rates and dataset sizes, bit error rates, and FPGA resource utilization.

4.1 System Modes

For incremental development and ETA system flexibility, the RCC must be capable of multiple operating modes. The first mode, Raw Mode, passes raw data from the S25 digital receivers to the acquisition PCs without data processing. The next two modes, Beamform-

ing Mode and FFT Beamforming Mode, process data to compress it and reduce off-line processing. Beamforming Mode performs time-series or spatial beamforming on the received data streams, and FFT Beamforming Mode performs frequency domain or spectral beamforming on the received data streams. The remaining modes are for testing purposes and facilitate validation and troubleshooting. The following sections discuss the RCC operating modes in more detail.

4.1.1 Raw Mode

Raw data acquisition mode allows the system to record the raw digitally sampled data stream produced at the output of the S25 boards. In this mode of operation the RCC must decode and multiplex multiple antenna streams. The number of multiplexed antenna streams is limited by the to disk throughput of the SC430 acquisition PCs. The SC430s are only capable of recording data at about 60 MBs/s for one hour. Each antenna stream is composed of 2 bytes (14 data bits and 2 counter bits) sent at 7.5 Msps. The counter bits can be combined with those of an adjacent antenna stream to produce a 4-bit sample ID counter. This gives a throughput data rate of 15 MBs/s per antenna data stream. One acquisition PC can therefore record data from four antenna data streams over the span of one hour. In this mode ETA as a whole is capable of recording data from sixteen antennas over an hour at a recording rate of 240 MBs/s aggregate. This yields a data set of nearly 800 GBs or 200 GBs per PC.

Raw data acquisition mode is not computationally intensive but does require system resources for routing, decoding, and multiplexing the incoming data streams in real time. As discussed above, each acquisition PC will record the data streams of four antennas. A combined data stream can be produced by using two RCC outer nodes which will decode and forward data to a single RCC inner node. The inner node multiplexes the data streams and outputs the result to the acquisition PCs. Twelve RCC nodes are required leaving four outer nodes unused. Since data is not being processed but simply passed through the RCC

network, the link bandwidth required between outer and inner nodes is 30 MBs/s per link or 240 MBs/s aggregate. Data sets recorded in Raw mode can be archived to tape and analyzed off-site to search for relevant events which may have been recorded. A more detailed description of this implementation is given in Chapter 5.

Raw mode can be used to validate system operation by observing known galactic RF pulse sources such as the Crab Pulsar. Data recorded in this mode is also valuable because it has not been processed and is simply the digital feed from the digital receiver nodes. Hence the same data set can be processed using a variety of techniques which may search for specific or unknown events. In this way the data sets are very flexible and can be used for many applications. The drawback of storing unprocessed data is that off-line processing time increases dramatically. A powerful PC may require multiple days to process a 200 GB data set. A significant amount of processing can take place in the RCC FPGAs during real-time operation to reduce off-line processing time. Furthermore the initial processing that can be performed in the RCC reduces the amount of data that must be stored and does not seriously limit the uses of a recorded data set.

4.1.2 Beamforming Mode

The Beamforming Mode implements spatial beamforming, as described in section 2.2.2, to transform and combine the incoming data streams. A number of patrol beams covering the sky are produced and recorded for further analysis and storage. Beamforming increases system sensitivity in a particular region of the sky and can suppress regions which would be less likely to produce desirable results, such as near and below the horizon. This section discusses the impact real-time beamforming has on the RCC system design.

To perform spatial beamforming in real time, the RCC must provide sufficient computation, routing, and synchronization capability. The digitized antenna streams must be synchronized, multiplied by corresponding beamforming coefficients, and summed with corresponding samples of adjacent antenna streams. Furthermore, this mode must be capable of utilizing

all 24 antenna inputs to form the individual beams. The beamforming computation is simplified by the fact that the two polarizations remain separate throughout the beamforming computation.

Twelve antenna inputs form a beam of the north-south polarization, and the remaining twelve antenna inputs form a beam of the east-west polarization. A pair of beams are referred to separately as two single-polarization beams or jointly as a single dual-polarized beam. Since the north-south and east-west polarizations are processed separately, the RCC can be partitioned and replicated for each polarization. The north-south polarization uses eight of the RCC nodes and the east-west polarization uses the remaining eight nodes to perform the beamforming computation. The required computation causes a large amount of bit growth, and the twelve single-polarization data streams must be combined into one stream. This makes networking and network bandwidth between the RCC nodes an important design factor to consider. Due to bit growth through the complex multiply accumulate operation, the number of bits in the data path increases from 2 bytes to 4 bytes per sample. A link bandwidth of 120 MBs/s allows four 32-bit data streams to pass through a single link sampled at 7.5 Msps. The ability to combine twelve streams also requires a multi-connection routing network. Network and link bandwidth will be discussed further in Section 4.2.

The number of beams which can be formed is again limited by the speed and duration of writing to disk in the acquisition nodes. Each beam is similar to the raw data format encoding and throughput. The beamforming computation combines multiple streams but maintains the same sample rate of 7.5 Msps. A single sample of a single beam consists of 2 bytes (14 data bits and 2 counter bits) and gives a required throughput of 15 MBs/s for each beam that is formed. Beamforming mode can then record four single-polarization beams per acquisition PC or sixteen single-polarization beams aggregate before the 60 MBs/s per PC limitation is reached. Eight beams is desirable in the ETA system because it allows a majority of the visible sky to be observed. Each of the eight beams require a single complex multiplier coefficient for each of twelve antenna input stream. Each RCC outer node connects to two antenna streams, and thus must have a coefficient table containing sixteen complex

beamforming coefficients. In addition the coefficients must be able to be updated or adjusted remotely.

While the disk recording speed limits the number of beams that can be formed, other design factors need to be considered for the hardware implementation of the beamforming algorithm. Including how many bits represent the beamforming coefficients and which bits to keep after a hardware multiply and accumulate operations. Both of these factors can greatly effect system sensitivity and precision. A software model is used to investigate how bit representations and the maintaining of bit widths effect actual system sensitivity readings. Analysis from this model showed that little quantization is produced by representing the two beamforming coefficients as 32-bit signed complex values. Based on the coefficient values and expected input ranges, the desired output bits shift position. The RCC design is created so that the output bit selection is variable and can be adjusted by the user. The software model also revealed that if bit selection is done by either removing the least significant bits or applying standard rounding rules, an undesirable DC bias is introduced into the system. To avoid adding a DC bias a convergent rounding scheme is implemented. The actual beamforming mode implementation will be discussed further in Chapter 5.

4.1.3 FFT Beamforming Mode

FFT Beamforming Mode performs spectral beamforming and requires computation similar to spatial beamforming. Spectral beamforming is expanded to form beams on frequency domain samples rather than time series samples. FFT Beamforming Mode continues to form eight dual-polarization beams, require a link bandwidth of 120 MBs/s, and disk data rates of 60 MBs/s per PC. Additional capability is added in FFT Beamforming Mode to compute a 1024-point Fast Fourier Transform and expand the number of multiplying coefficients. This section discusses the impact spectral beamforming has on the RCC system design.

In FFT Beamforming Mode a 1024-point FFT is added to the processing path before the beamforming computation. The FFT transforms 1024 time-series samples to 1024 spectral or

frequency channels. Each channel or bin contains the amplitude and phase information for a corresponding frequency band of the incoming time-sampled series. Thus a sine wave would produce an output in a single frequency bin corresponding to the frequency of the sine wave. Spectral beamforming also requires that each bin has a different beamforming coefficient. Spectral beamforming changes the design by expanding the data width, requiring additional bit reduction or bit selection modules, and increasing the number of required multiplication coefficients.

The FFT core increases the required FPGA resources and data width through the RCC design. The input to the FFT is represented using 14 bits (7-bit I, 7-bit Q) and the output is represented by 38 bits (19-bit I, 19-bit Q). The multiplication and summation blocks cause additional bit growth after the FFT. Before transmitting data over the network, another bit selection module is added to the system. The desired bits are selected and transmitted over the RCC network to be processed further by RCC inner nodes. Convergent rounding is applied to again avoid introducing a DC bias into the system. The additional bit selection modules maintain a manageable data width through the design and reduce the amount of data transmitted through the RCC network. This reduction limits the required RCC link bandwidth to 120 MBs/s which is the same bandwidth required in the Beamforming Mode.

The spectral beamforming algorithm is more complex because beamforming is performed on individual spectral bins rather than the antenna streams. Each outer node forms sixteen beams (eight beams per two input antenna streams). All of the 1024 spectral bins in each beam must have a separate beamforming coefficient. Therefore the outer node coefficient tables must be expanded from 16 to 16284 complex coefficients. The added coefficients requires a large amount of FPGA storage that is still capable of being updated remotely.

4.1.4 Test Modes

The design of any complex system should also assist debugging or troubleshooting. Various test modes are designed into the RCC which make the design easier to test, validate, and

debug. A majority of these test modes inject synthesized data into the system at key locations, confirming that processing blocks are performing the desired computation and identifying processing blocks which may be causing a problem. This section gives a few examples of test modes present in the RCC design and how they might be used.

Several test modes are used to verify the operation of the beamforming algorithm, which performs complex multiplication and summation over twelve input data streams. One of the initial test modes allow the user to set the inputs of the beamforming module to a constant value through the UART control interface. These constants are then be multiplied and summed to produce an expected output at the end of the computation. By setting the beamforming module inputs to several sets of constants, the computation can be verified. However this test mode does not check that each sample is being combined with the correct samples from adjacent data streams. Another mode further tests the beamforming computation and sends input values, generally counters, to the beamforming module. This test produces a known output stream that can be verified. If the output stream is incorrect, samples are not being synchronized properly before they are combined with other data streams. Through the process of setting new coefficients and input values, errors in the system design become evident and are corrected. The test modes remain in the final system implementation and can be used to narrow down the source of problems which may occur under actual operation.

An additional test mode enables the RCC network to be examined for both basic function and bit errors. Counters are sourced across all of the Aurora data links in the RCC. The receiving end of each link analyzes the incoming counters and reports any errors which occur. The link error information can then be read back through the ML310 UART interface to determine which links produce errors and how frequently. Through the use of this mode, all of the links in the system can be tested rapidly and problematic links can be repaired through swapping cables or adapter boards. After stable communication is established, this mode can be used to measure the bit error rate performance of the network. By periodically reading back the error information over an extended period of time, the bit error rate can be estimated with reasonable accuracy. Testing has shown that the RCC network can stream

data over the period of 5 hours with fewer than 10 bit errors over the entire network. Thus an 800 GB data set recorded over an hour will have on the order of 2-5 bit errors which is well within the tolerance range of the noise-dominated received radio signal and is achieved without data retransmission.

4.2 Networking

The RCC network must be capable of streaming data from multiple outer nodes to a single inner node at high rates. Due to the limited buffering and continuous data streams which must pass through the RCC network, it is difficult to implement packet-based protocols which allow for data retransmission or flow control. Data retransmission is undesirable because a single retransmission would stall the entire data processing chain. In traditional systems this is accounted for with the use of flow control signaling which inform the transmitting side of the link that is should stop sending data. This would back up the system and ultimately result in data loss unless large amounts of buffering were added to the system. The network is also highly noise immune and resistant to bit errors.

Each Xilinx XC2VP30 FPGA has eight MGTs allowing for raw data transfer speeds of around 300 MBs/s (2.5 Gbits/s), ideal for data transmission [31]. Normally this data rate is not achieved due to protocol overhead. The Aurora protocol is used for the RCC design because it is simple to implement both in terms of protocol signaling and resource requirements. As discussed previously, the maximum data rate required for each link is 120 MBs/s or one third of the MGT capability. This being the case additional steps are taken to reduce the bit error rate with the extra bandwidth. The ETA system RCC data links use 8b/10b encoding to maintain a zero DC balance and reduce the bit error rate over the transmission line. This technique requires a larger number of bits to transmit data but can detect and correct minor errors more readily. The increased signal quality achieved by the use of 8b/10b encoding is preferred over increased bandwidth because the ETA system utilizes only 120 MBs/s of the

available link bandwidth.

Other methods including lowering the transmission clock frequency, increasing the differential voltage swing, and adjusting the pre-emphasis are applied to reduce the bit error rate and increase noise immunity. Rather than running the RCC network at a maximum clock frequency of 156.26 MHz, the 125 MHz system clock is applied to the RCC communication structure. After accounting for Aurora protocol and encoding overheads, the maximum data transfer rate is then reduced to approximately 250 MBs/s. The system will use approximately half the available link bandwidth and allow for ample bandwidth overhead to avoid link contention. As discussed previously in Chapter 2, lowering the communication clock frequency, increasing the differential voltage, and adjusting the amount of pre-emphasis produce more stable, error-free, and noise immune communications.

The final considerations which improves network signal quality deals with the physical transmission links between the RCC nodes. In transmission from FPGA to FPGA, care has been taken to ensure good signal quality in the physical links. Low-voltage differential signaling is used and routed side by side on PCB and or sent on a twisted paired transmission cable. Both of these design items are taken into account for designing the RCC network. The ML310 adapter boards were designed according to high speed signaling design rules, and connect to the ML310 boards through impedance-controlled connectors. The physical cabling between nodes is done with high speed InfiniBand cables and connectors. Anticipating that not all of the ports on the adapter boards would function as desired, redundant output connections are used. RCC outer nodes send data output to six of the adapter board InfiniBand ports when only two of them are used at a time. The redundant output ports allow error-prone ports to be avoided and stable connections selected. Chapter 5 describes the routing connections made between nodes in more detail.

4.3 Synchronization

During raw mode operation, synchronization is not required because the inputs simply pass through the system without being processed. However both of the beamforming modes require sample-by-sample synchronization throughout system processing to ensure correct FFT and beamforming computations. The S25 digital receiver boards output a multiplexed combination of two antenna streams, 4-bit sample ID counter, and start of frame signal. To enable synchronization across all of the receiver boards, a single master clock and reset is distributed to the twelve S25 boards which allows the ID counter and start of frame bits to be synchronous across the S25s. Internal to each S25, the master clock is used to derive the sample clock transmitted along side a 5-bit parallel bus to the RCC outer nodes. Since the outer nodes receive a multiplexed data stream, the two antenna samples, ID bits, and start of frame bits are inherently synchronized after being decoded. Synchronization must then be maintained as samples are processed through the RCC outer nodes and transmitted to the inner nodes.

The inner nodes must explicitly synchronize data received from six outer nodes. The ID counter and start of frame signals, which are synchronously sourced in the S25s, are used to enable this synchronization. The 4-bit sample ID counter wraps around repeating its initial value after 16 data cycles. Synchronization must therefore occur to within ± 8 data cycles and should probably occur within ± 4 data cycles to ensure that data from a current sample is not synchronized with data from a sample which is offset by 16. In practice, if processing paths and connecting cables have been designed to have identical delays, samples do not differ by more than 2 data cycles. The start of frame bits are specifically used in the FFT beamforming mode to signal the start of a 1024 sample sequence which is fed into the 1024-point FFT processing block. To ensure that data sets do not overlap, the start of frame signal is transmitted every two data frames or 2048 data cycles. Synchronization is maintained throughout processing by designing the data paths with identical processing delays or buffering. Parallel processing paths are well suited for the FPGA development

environment, and enable data computation to occur synchronously and at high data rates. The RCC registers any loss of synchronization as this will result in data corruption.

4.4 Remote Access to System Nodes

The ETA radio telescope is located near Rosman NC, approximately 260 miles by road from Virginia Tech campus. The distant location requires remote access through the Internet to the ETA system. The control PC connects to the Internet, four acquisition PCs, and sixteen RCC nodes and acts as the gateway to access and manage the ETA system. The following section discusses the impact remotely observing and updating has on the RCC system design.

The control PC is equipped with two 8-port RS232 expansion cards that interface with the RS232 ports on the RCC nodes. Each RCC node utilizes a hardware UART interface to send or receive status information to or from the control PC. These connections allow easy access to the ML310s through a serial interface such as Minicom [36] or through software programs or scripting. The remote interface to the RCC nodes allows for both setting new mode or coefficient information and reading back current mode or system status information. A scan chain, consisting of a series of shift registers, is used to pass mode information into or out of the RCC nodes.

Mode information that can be passed into the system includes: the desired operating mode, complex beamforming coefficients, select bit settings, test mode constants or values, and reset signaling. Mode and status information that can be read out of the system includes: all of the mode information that can be passed into the system, link status and error codes, synchronization status and error codes, test modes and error codes, and system clock counters. The UART interface and scan chain allow a simple means of viewing or modifying system operation. The error codes read back from the system help to pinpoint problems which may occur or certify that no errors have occurred during an acquisition. By viewing the system status and error codes before, during, and after an acquisition a user can check

system integrity for each acquisition.

4.5 FPGA Resources

Field Programmable Gate Arrays provide a desirable computing platform for radio telescope implementations. Computations can be implemented in parallel and with custom hardware to perform real-time signal processing as data is being streamed through the system. Traditional and even high-end PCs are not preferred for computationally intensive streaming applications because they do not have specialized processing resources and may take an indeterminate number of clock cycles to process data. Another problem with PCs or even PC clusters is that the routing and protocol stack overhead required. FPGA-based platforms can create high speed custom networks for passing data directly to other processing nodes. This network bandwidth, customization, and low overhead is difficult to find in traditional computer resources.

FPGAs also permit rapid development when compared to that of an ASIC or other custom hardware components. By using commercially available FPGA-based development boards, the RCC was designed rapidly and with a large amount of flexibility in both computation and communication resources. The development of the RCC node implementations were further accelerated by the use of design tools available from Xilinx. A few of these tools include the ISE Project Manager for creating and compiling designs, CoreGen for customizing common computational blocks which exist as Xilinx IP, and ChipScope which adds logic analyzer observability to a design and aids debug or validation testing.

Chapter 5

Reconfigurable Computing Cluster

5.1 Overview

The reconfigurable computing cluster performs real-time, stream processing and provides system routing in the ETA radio telescope. The RCC requirements are well suited to FPGAs because they provide high real-time computational throughput and multiple high-speed communication interfaces. The RCC is comprised of sixteen FPGA nodes which are networked using the Aurora protocol over InfiniBand cabling. Each node uses a Xilinx ML310 development board and is equipped with a Virtex-II Pro 2VP30 FPGA. The ML310 development board is used because it provides the necessary computational and communication hardware resources, is familiar to the design team, and is available for a reduced cost through the Xilinx University Program. The sixteen RCC nodes are divided between twelve outer nodes, which receive data from the twelve S25 digital receiver boards, and four inner nodes, which transmit data to the four acquisition PCs. The adapter board shown in Figure 5.1 is used to create the physical connection to both the S25s and acquisition PCs. The outer nodes perform a large portion of the required processing and pass data to the RCC inner nodes through the Aurora over InfiniBand network. The adapter board shown in Figure 5.2 is used to create the network connections between RCC nodes. The inner nodes synchronize the

data streams from the outer nodes and combine the streams to form the acquisition output data stream. The RCC was developed in three stages related to the operating modes discussed in the Chapter 3. Each of the three implementation stages - raw mode, beamforming mode, and FFT beamforming mode - are discussed in the following section. Each implementation has all the functionality of the previous implementations and thus the beamforming and FFT beamforming sections will focus on the additions that are made to accommodate the new mode of operation.

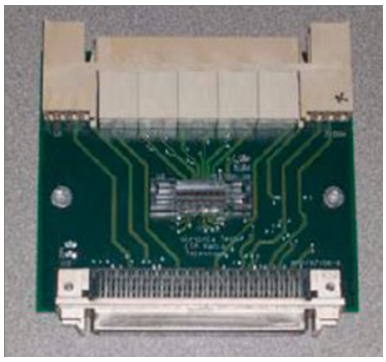

Figure 5.1: ML310 to S25 and EDT interface adapter board. top: ML310 connection, middle: MICTOR to S25 connection, bottom: EDT connection.

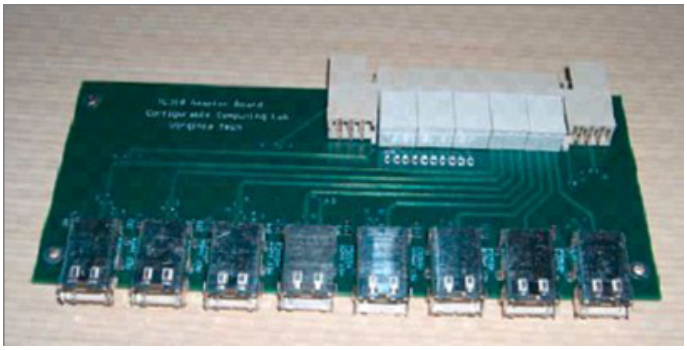

Figure 5.2: ML310 to InfiniBand interface adapter board. top: ML310 connection, bottom: eight Infiniband connections.

5.2 Raw Mode

The RCC raw mode allows eight of the S25-sourced data stream to pass directly through the RCC to the acquisition PCs. Data reduction is not performed but the input stream is decoded and reformatted for acquisition. The design is further simplified because it does not implement the UART control interface which is accessed through the control PC. Raw mode provides a necessary link to complete the ETA system path from antennas to storage and is

used to both certify system operation and store unprocessed data from the S25 digital receiver boards. This mode is less complex than either beamforming or FFT beamforming modes and is designed early so that full system integration testing can be performed. The raw mode implementation allows the full system to be tested and debugged before adding beamforming. The uncompressed recorded data can be interpreted in a variety of ways to assure system functionality or search for astronomical events. Recordings are currently processed to search for dispersed pulses which may have been observed. This section discusses the raw mode implementation, outer and inner node FPGA designs, and the testing and validation used.

As shown in Figure 5.3, raw mode operation utilizes eight of the RCC outer nodes and all of the four RCC inner nodes. Each outer node receives an input stream from an S25 digital receiver; this data stream is a multiplexed output of two antenna data streams, 4-bit sample ID counter, and start of frame bits. The outer nodes decode this input stream and pass it through the RCC InfiniBand network to the inner nodes. Synchronization across multiple data streams is not used in this mode. The sample ID counter is passed through the system with the antenna data streams, and the start of frame bits are stripped from the signal. The RCC inner nodes receive data from two outer nodes, multiplex the data streams, and encode the output acquisition data stream which is then sent to the acquisition PCs. The sample ID counter is stored to disk along with the antenna data samples and is used in off-line processing to correlate between data streams. The RCC network in raw mode operation connects two outer nodes to a single inner node. Figure 5.3 shows the RCC connections and it is noted that there are no links between adjacent data paths. The system can store data from any of 4, 8, 12, or 16 antennas simply by enabling or disabling data recording on the acquisition PCs. The ability to use only a portion of the antennas is beneficial during development or when not all of the antennas, receivers, and digital receivers are operational.

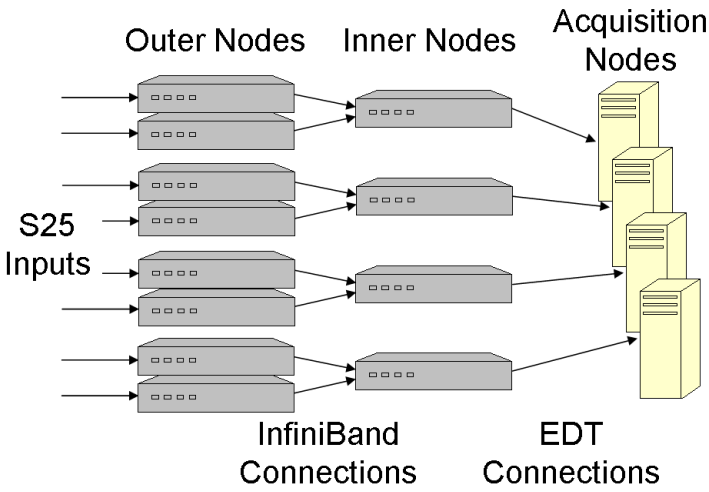

Figure 5.3: RCC raw mode system operation overview.

5.2.1 Outer Nodes

The raw mode outer FPGA implementation is shown in Figure 5.4 and is meant to be a simplified implementation to establish the data path, perform full system integration, and record digitally sampled but unprocessed antenna feeds. The S25 input stream is received at 37.5 MBs/s and contains two digitally sampled antenna streams with a corresponding sample ID counter. The two streams and ID counters are decoded to recreate the 7.5 Msps data stream produced in the S25 boards. Each antenna stream is represented using a 14-bit complex value (7-bit I, 7-bit Q) and the sample ID is represented using a 4-bit value. The resulting 32-bit data path matches the width used by the Aurora protocol interface. Data passes directly from the input decode to the Aurora transmission IP block. Using the Aurora protocol, data is transmitted at 30 MBs/s to the RCC inner nodes. While only one of the Infiniband connections is connected to an inner node, the output data stream is sent to all available Aurora connections. The redundant connections are later used for testing or to re-route faulty links.

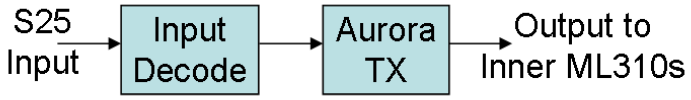

Figure 5.4: Raw mode hardware implementation overview of an RCC outer node.

5.2.2 Inner Nodes

Similar to the RCC outer node implementation, the inner node FPGA design is simplified for rapid development and system integration. Figure 5.5 shows an overview of the raw mode hardware implementation of the RCC inner nodes. The Aurora interface receives the transmitted 32-bit words at a rate of 30 MBs/s. Since an inner node receives data from two outer nodes the agregate input rate is 60 MBs/s. A buffered multiplexer is used to combine the two data streams and to account for any cycle by cycle differences which may occur between the outer node data streams. The two 32-bit data streams feed the inputs of the multiplexer where they are buffered and read alternating between the two streams. The output stream maintains the 32-bit data width but has a combined flow rate of 15 Msps. Data is then streamed to the output encoder which reformats the data for output to an acquisition PC. The PC's EDT data acquisition card has a 16-bit parallel LVDS interface that interleaves sample ID bits with antenna data bits. Data is output continuously through the EDT connection at a rate of 60 MBs/s and stored to disk by the acquisition PCs.

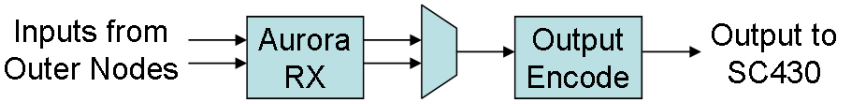

Figure 5.5: Raw mode hardware implementation overview of an RCC inner node.

5.2.3 Test and Validation

The raw mode of operation is tested in two ways: using ChipScope to verify internal timing and signaling is correct, and sourcing various input streams to analyze the results which are stored to disk. To facilitate testing and development, a workbench portion of the ETA system is constructed. The test setup for the raw mode implementation consists of one S25 board, two ML310s, and one acquisition PC. In the test setup the S25 sources data to one of the ML310s which contains the outer node configuration. The data stream is processed and transmitted through the Aurora connections as described in Section 5.2.1. Since the output data stream is sent to all available Aurora connections, the two outer nodes are simulated by connecting two of the output ports to the second ML310 containing the inner node implementation. Since the inner node does not analyze or compare the two input data streams, it is irrelevant that they are actually sourced by the same board. The inner node functionality was tested by evaluating the output of the second ML310 board at the acquisition PC. The PC uses a C program designed to interface with the EDT PCI adapter card [33] to record data in the desired block size and verify correct data acquisition.

To certify initial operation, counter data is sourced in the S25, streamed through the outer and inner ML310s, and recorded on disk. While operating in this test setup, ChipScope Analyzer cores are added to both the inner and outer node implementations. These cores allowed critical locations in the design and data path to be observed for timing correctness. Some of these locations include the output of the data decode, Aurora interface signaling, buffered multiplexer functionality and timing, and EDT output formating and timing. After correct timing and signaling are observed, correct data reception is verified manually by viewing the contents of recorded files. Final testing is performed and demonstrates that the system can record 200 GBs of data continuously at 60 MBs/s. The C program is updated to record 200 1-GB files and then read them back to certify that data is recorded correctly. Any errors found during file verification are tabulated and reported to the user console. This test requires approximately 1 hour to record data and 1.5 hours to read back and verify

data correctness. Through the use of scripts, the C program is run and output files are deleted multiple times to guarantee system operation. Testing beyond this requirement is also performed and shows that recording data at rates above 65 MBs/s or in quantities larger than 250 GBs results in lost or dropped data samples. The data rate limitation is due to the limited to disk throughput available in the acquisition PCs. The 250 GBs limitation results from a decrease in the throughput rates as the internal hard disks become full and data is recorded on inner tracks. This testing verifies that the raw mode implementation is stable and able to be installed at PARI.

5.3 Beamforming Mode

The RCC beamforming mode allows the twelve S25-sourced, time-series data streams to be processed and combined through the RCC. Each antenna stream is multiplied by a complex coefficient for each beam formed and then summed with samples having the same sample ID, beam number, and polarization. The processing requires the synchronous summation of twelve antenna input streams which are combined into a beam of a single polarization. Since beams of differing polarizations are formed independently of each other, the design is simplified by dividing the system between polarizations. Six outer nodes and two inner nodes are used to perform the beamforming computation on each of the North-South and East-West polarizations respectively, making use of all sixteen RCC nodes. Eight beams of each polarization are formed producing a total of sixteen single polarization beams. This section provides an overview of the beamforming mode implementation, discusses the outer and inner node FPGA designs, and describes the testing and validation which was performed on the beamforming mode implementation.

The system overview of the beamforming mode implementation is shown in Figure 5.6. A majority of the beamforming computation is performed in the RCC outer nodes. The inner nodes synchronize data streams from six outer nodes and combine them through a

summation. A UART scan chain interface, which connects through RS232 to the control PC, is also added to the design. Through the UART interface a user can select between the raw and beamforming modes, change beamforming coefficients, read back system information and error codes, and enable various test modes. The UART interface is also helpful during development to read back system information or test partially implemented modules.

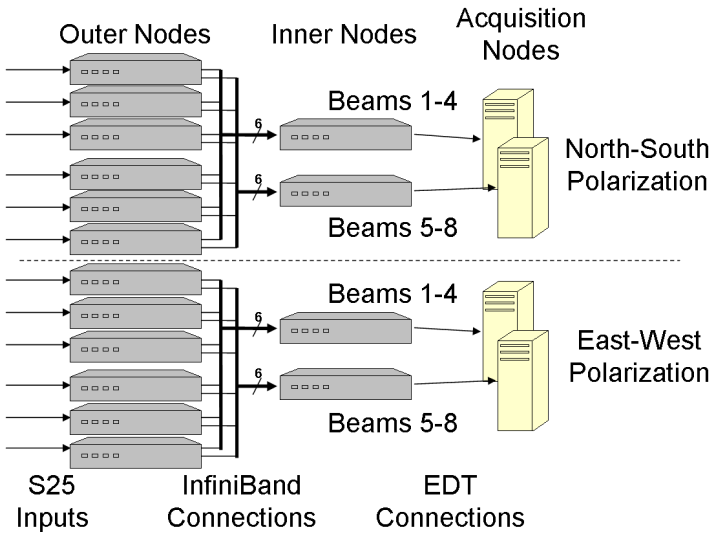

Figure 5.6: RCC beamforming mode system operation overview.

5.3.1 Outer Nodes

The hardware implementation of the RCC outer nodes becomes significantly more complex since a majority of the beamforming computation is added to these nodes. The beamforming computation may also be bypassed to retain the same functionality as the previous raw mode implementation. Each outer node forms eight single polarization beams, four of which are output to an inner node and the remaining four are output to an adjacent inner node. To match that architecture, the beamforming computation is partitioned between two identical modules, which form an output stream of four beams and forwards data to the output select and Aurora interface. Synchronization between the two antenna streams is maintained

throughout the beamforming computation by passing samples through similar logic and delays until they are combined. The design overview for the beamforming outer node FPGA design is shown in Figure 5.7. Also shown in the Figure is the UART control interface and control signals to internal modules. As inferred by the Figure, the control interface is used to monitor system function, select between test and S25 inputs, modify beamforming parameters, and select the desired output.

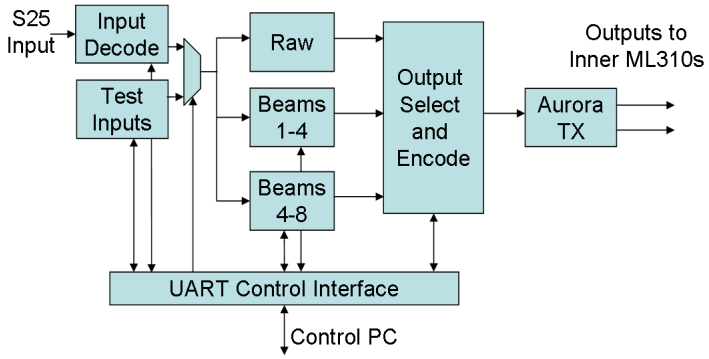

Figure 5.7: RCC beamforming mode outer node FPGA implementation overview.

Beamforming Module

The beamforming modules, labeled as Beams in Figure 5.7, perform the complex multiplication and summation of two antenna inputs for each of four beams. An overview of the beamforming processing path is shown in Figure 5.8. As shown the input is replicated four times for each of the four beams constructed. The input and output waveforms of the Replicate by 4 block are shown in Figure 5.9. The input rate of 7.5 Msps is thus quadrupled to 30 Msps at the output of the Replicate by 4 functional block. The internal clocking of this block and subsequent logic must therefore exceed 30 MHz. The ML310 input clock operates at 125 MHz and is divided to produce the 62.5 MHz clock used for the design. The 62.5 MHz clock frequency provides stable communication over data links and a sufficient margin to handle the small backlogs occurring as result of communication overheads. Input buffering

is also provided to ensure that successive samples are not lost as the input is replicated. In parallel the 4-bit ID counter is augmented to six bits and used to differentiate between the replicated samples. The two added bits designate each replicated sample to a different beam and select the correct multiplying coefficient from the Coeff Table. Each coefficient table contains a 32-bit complex coefficient (16-bit I, 16-bit Q) for each of the four beams formed and cycles through coefficients as samples are streamed into the complex multipliers. After the replicated samples are multiplied by a corresponding beamforming coefficient, they are summed and rounded to reduce the number of bits transmitted to the RCC inner nodes. The partially beamformed samples are then recombined with the original 4-bit sample ID counter and passed to the output of the module. Even though the replication and multiplication of the two antenna streams occur through separate physical logic, the delays that occur on each path are identical and maintain synchronization.

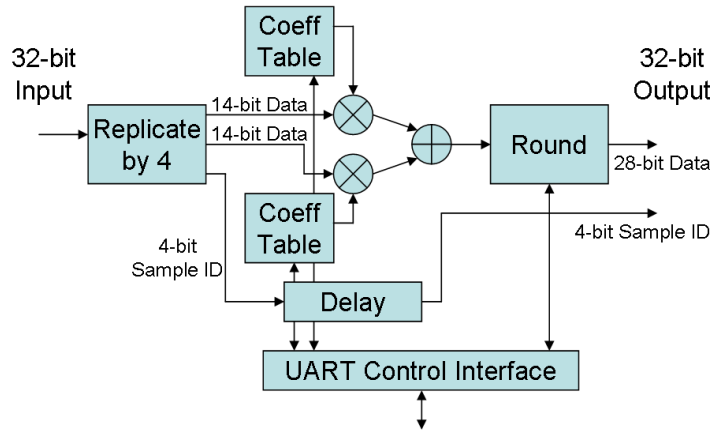

Figure 5.8: Beamforming module implementation for the beamforming mode. Forms four beams from two antenna inputs.

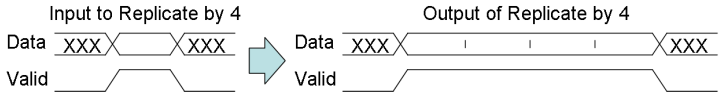

Figure 5.9: Replicate by four input and output timing.

Rounding is required before the samples can be transmitted over the Aurora connections due to data width expansion through the multiply and accumulate operations. The two 14-bit (7-bit I, 7-bit Q) inputs are multiplied and summed to produce a single 50-bit (25-bit I, 25-bit Q) sample. The 32-bit interface of the Aurora protocol limits the width of transmitted data and requires bit reduction through rounding. A portion of these bits are removed without significantly affecting precision especially since only fourteen bits of each sample will be stored to disk. The rounding module selects 28 bits (14-bits I, 14 bits Q) from the 50-bit signal and passes them to the output. The remaining four bits contain the ID counter originally sourced with each sample. Users select the desired 28 bits from the available 50-bit signal through the UART control interface and may vary the selected bits based on the chosen multiplication coefficients. To avoid adding a DC bias to the signal path, rounding-half-up is implemented in this and other rounding modules. Rounding-half-up is chosen for this implementation because it is easily implemented in hardware by adding .5 and truncating the binary number representation. More complex rounding schemes, such as convergent rounding, were not needed for the RCC implementation.

UART Control Interface

The UART control interface allows the system to be monitored or updated with new coefficients and mode information. As shown in Figures 5.7 and 5.8, through this interface input data can be modified or selected, beamforming coefficients are updated, rounding position is set, and output data streams are chosen. Additional connections to internal buffers monitor data flow through the system. Buffer overflows are tallied and reported through the UART control interface with the other system information mentioned. The control interface is useful for both design and system integration testing.

5.3.2 Inner Nodes

The inner node beamforming FPGA implementation is less complex than that of the outer nodes. RCC inner nodes synchronize input data from six outer nodes, combine similar time and beam samples through a summation, and format the data for acquisition. As with the outer node beamforming implementation, raw mode pass through can be enabled through an output multiplexer. The inner node FPGA implementation is shown in Figure 5.10. Synchronization is accomplished by buffering samples of all input streams and comparing sample IDs as values are read from the buffers. The system gains and maintains synchronization by aligning the ID values of all input streams. Synchronization across the input channels or loss of synchronization can be monitored through the UART control interface. After the data streams are synchronized, rounding takes place in the Adder Tree module and is shown in Figure 5.11.

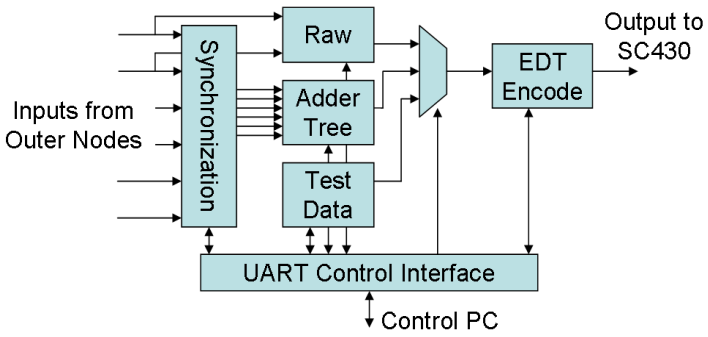

Figure 5.10: RCC beamforming mode inner node FPGA implementation overview.

The Adder Tree sums the six 28-bit (14-bit I, 14-bit Q) synchronized values and rounds the result at a user specified location to produce a 14-bit (7-bit I, 7-bit Q) output data stream. Rounding is implemented as described in Section 5.3.1. Both the 28-bit input and the 14-bit output signals contain samples from each of the four beams as produced from the outer nodes. The EDT PCI card on the acquisition PCs has a 16-bit parallel LVDS interface. Each 14-bit output value is combined with two bits of the sample ID. Given that there are four (one corresponding to each beam formed) values with the same sample ID, the sample ID is

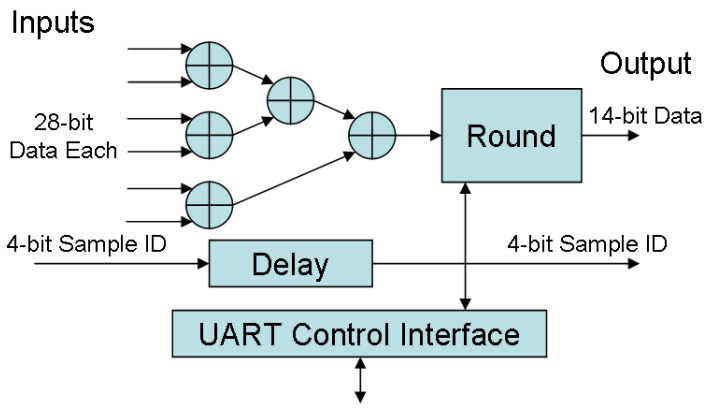

Figure 5.11: Adder tree module implementation.

transmitted to the acquisition PC twice. The sample ID is used further in off-line processing to align data across multiple PCs and as an initial data integrity check. When the system is operating, the sample ID is easily extracted to properly align data.

The UART control interface is also added to the inner nodes and can be used to monitor received sample ID, buffer overflows, and synchronization errors. Errors received during an acquisition can be read from the system to ensure that the stored data is not corrupt. The UART controller is also used to monitor the Aurora link, modify test inputs, select the final rounding location, and select the data stream transmitted to the acquisition PCs.

5.3.3 Test and Validation

Workbench testing for the beamforming mode is done through the use of a system model similar to the one described in Section 5.2.3. Two ML310s are used, one simulating six outer nodes and the other simulating an inner node. The single outer node board sources six output signals which are tied to the six inputs of the inner node. In the full system the six inputs are sourced from different outer nodes. ChipScope Analyzer is used to verify timing through the system as with the previous mode test setup. The UART control interface is also

used during testing to effectively detect and localize system errors. Following initial testing through these two interfaces, test data is driven through the system and output results are inspected. To aid the process of developing test inputs and verifying the output produced, a software model programmed in C is used. The C model also aids in selecting coefficient bit widths and rounding implementations. The output results from this model show that 32-bit multiplying coefficients (16-bit I, 16-bit Q) and round-half-up rounding produce sufficient precision and avoid adding a DC bias to the system. The C model creates an easy way to develop and verify test input values. As with most system designs, exhaustive testing of all possible input combinations would be infeasible. Thus several test vector inputs are applied to the system and ensure proper functionality. System integration is facilitated through the test inputs and monitoring connections which are accessed through the UART control interface and control PC.

The ability to identify poor physical connections rapidly and replace them using alternate cables or adapter boards dramatically reduces installation time. System monitoring of individual Aurora connections allow poor connections to be identified quickly and redundant ports allow faulty ports to be left unused. Link integrity is also improved by increasing the differential voltage to its maximum 800 mV and adjusting pre-emphasis to match cable lengths. A pre-emphasis of 10% and 20% produced the best signal quality for two and three meter InfiniBand cables respectively.

Test inputs and test modes also enable rapid system integration and testing. Using the test modes, counter data can be injected across all of the Aurora connections and verified at the inner nodes. Any errors in the received data are recorded and can be read from the control PC. Through this testing procedure, the bit error rate over each Aurora link in the system is evaluated. In general the links pass data with no detected errors and the worst links produced a BER on the order of 10^{-12}. This analysis shows that under normal operation, a 200 GB data set will contain fewer than 10 bit errors caused by the data transmission over the Aurora connections. All other connection in the system operate at a lower transmission rates and have a negligible BER. Following initial link validation, additional integration

testing is performed using the test vectors created from the C simulation and lab testing. Data sets acquired with this implementation have been recorded and are currently being evaluated for dispersed pulses.

5.4 FFT Beamforming Mode

The FFT Beamforming mode utilizes all of the RCC nodes in the same beamforming configuration shown in Figure 5.6. FFT beamforming partitions the system between the North-South and East-West polarizations and forms eight beams of each polarization. The main additions to this mode is that a 1024-point FFT is performed on the S25 input before the beamforming operation. The FFT transforms a block of 1024 time samples into 1024 frequency samples. Beamforming is then performed on each of the 1024 frequency spectra channels separately. To synchronize the FFT computation across the 12 outer nodes, a start of frame (SOF) signal is added to S25 input signal. This signal strobes every 2048 cycles and signals the boundary of FFT operations to the outer nodes. By synchronizing to this signal, the outer nodes will perform an FFT on the same 1024 block of input time series samples. Beamforming is then performed on each of the frequency samples across the antenna inputs. These additions require the outer node FPGA implementation to have start of frame synchronization, a wider data path, larger buffers, and expanded coefficient tables. The inner node FPGA implementation changes little but utilizes a new synchronization method and expands the size of storage buffers to accommodate the larger data bursts. This section describes the outer and inner node implementations and validation testing performed for the FFT beamforming mode.

5.4.1 Outer Nodes

The FFT beamforming mode outer node FPGA implementation requires significant modifications to the design, but the high level description remains very similar the beamforming

mode description. The previous raw and beamforming modes are implemented as subsets of the FFT beamforming mode. They are selected by bypassing the FFT and performing the required computation. The computation required for the beamforming and FFT beamforming modes is performed in the same beamforming module. The difference between the modes is the input selection and the user defined beamforming coefficients. The overview for the outer node FPGA implementation of the FFT beamforming mode is shown in Figure 5.12.

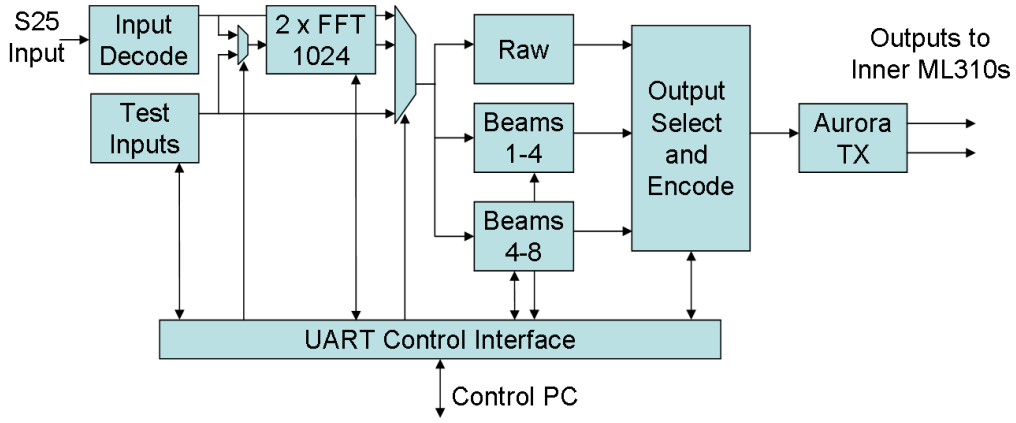

Figure 5.12: RCC FFT beamforming mode outer node FPGA implementation overview.

Following the S25 input decode a 1024-point FFT module is added for each of the two 14-bit antenna input streams (7-bit I, 7-bit Q). Due to the blocking nature of the FFT, 1024 consecutive samples are buffered and then streamed into the FFT modules. The least significant bits of the 38-bit FFT output (19-bit I, 19-bit Q) are rounded to produce a 36-bit (18-bit I, 18-bit Q) value which can be input to the Virtex-II 18x18 multipliers used for the complex multiplication. This does not affect the final output as the significant bits at this point are between bits 13 and 6. To accommodate the increased input data width (36-bit versus 14-bit), the two beamforming modules have expanded data paths. While the system is operating in beamforming mode, the FFT is bypassed and the 14-bit input is sign extended to 36 bits. In addition the beamforming module is modified so that each of the four beams

are formed in parallel rather than sequentially. Since beamforming is performed on each of the 1024 frequency channels, the coefficient tables are expanded to contain 1024 32-bit coefficients for each beam and input signal. A single outer node performs this computation on two inputs and forms eight beams thus requiring sixteen coefficient tables containing a total of 16384 32-bit coefficients. To operate in the previous beamforming mode, the coefficients contained in each table are set to the same value. The FFT output also produces an 11-bit ID value corresponding to the 1024 frequency channel outputs. This ID value is used to address the coefficient tables and select the proper multiplication coefficient. Due to the delay induced by addressing the coefficient tables, delays are added to the multiplier input paths. The output of the multiply accumulate is then rounded to 30 bits (15-bit I, 15-bit Q), buffered, and multiplexed with three other beams to produce a single stream with alternating samples from each of four beams. The 30-bit beamformed output is combined with the SOF signal to be transmitted to the RCC inner nodes and is used for synchronization. The position of the SOF signal is not tied to a specific FFT sample, but is used to synchronize data across multiple nodes or PCs. In practice the start of frame strobe is offset by eighteen samples from the first FFT output. Slight modifications are also made to the UART control interface to provide additional monitoring and interfacing to the FFT beamforming mode. The beamforming module that is used in this mode is shown in Figure 5.13.

5.4.2 Inner Nodes

The inner node FFT beamforming implementation overview is the same as that specified in the beamforming mode and shown in Figure 5.10. The main modifications to the FFT inner node FPGA design is the synchronization method and increased data path width. Rather than synchronizing to the sample ID counter, synchronization is attained using the SOF signal which was originally sourced in the S25 boards and passed through the outer nodes. In addition, the adder tree data path is expanded to allow for a 30-bit rather than 28-bit input. A few modifications were also added to the UART control interface to better detect

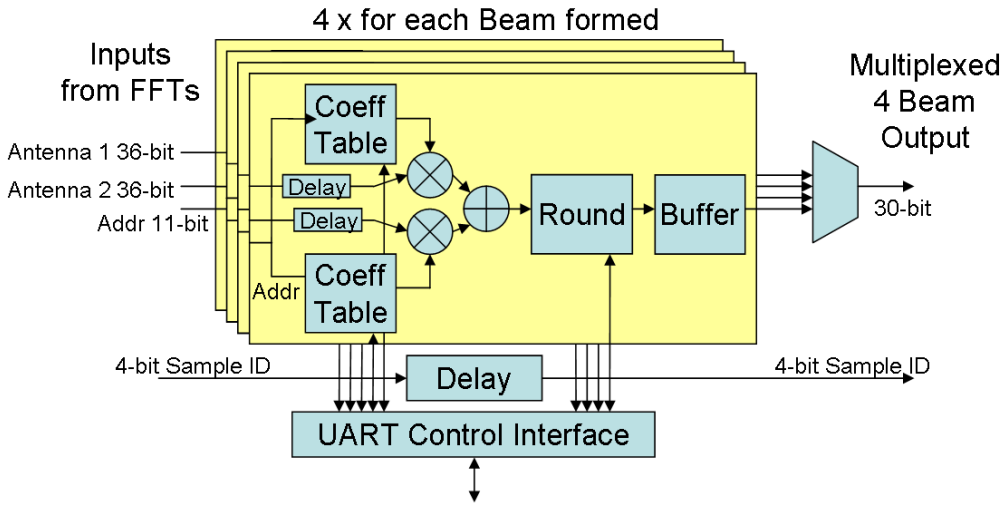

Figure 5.13: Beamforming module implementation for the FFT beamforming mode. Forms four beams from two antenna inputs.

and report errors.

5.4.3 Test and Validation

Testing for the FFT beamforming mode is similar to the testing of previous modes. Lab testing is done using two ML310 boards, one for each of the inner and outer node implementations, and one acquisition PC. The outer node ML310 receives, processes, and sends data out of the six Aurora connections to the ML310 inner node. The inner node synchronizes the data streams received through the Aurora connections, combines them, and passes the result to the acquisition PC. ChipScope is used to view and verify internal signaling and timing. The hardware implementation is first tested in the beamforming mode of operation since test vectors are already created to test a large portion of the system's functionality including the beamforming computation, data flow through the system, and synchronization. Following this, further testing is done to validate added functionality. Test constants are input to beamforming modules and multiplying coefficients are set to different values to

certify that coefficient tables are properly cycling and multiplying inputs. Final testing is done by implementing the FFT beamforming mode and monitoring the output and status signals through the system. Due to the lack of an accurate FFT system model, test vectors are not developed to verify system functionality with the FFT operation. While progress has been made toward creating an FFT system model this is left for future work.

Full system integration testing is done by following the test procedures listed above on the complete system. One of the main focuses of full system testing is certifying that synchronization is maintained throughout the entire system. This is facilitated through the control PC which connects to each of the RCC nodes. Following the initial system validation, physical testing is performed by connecting the full system, generating an in-band tone, and viewing the output. Figure 5.14 shows the results from this test. The graph shows the magnitude of the complex output signal. The broad region in the middle of the figure is the DC value present in the signal, and the sharp spike to the left is the detected signal tone. The proper detection of the generated shows correct operation of the FFT beamforming implementation.

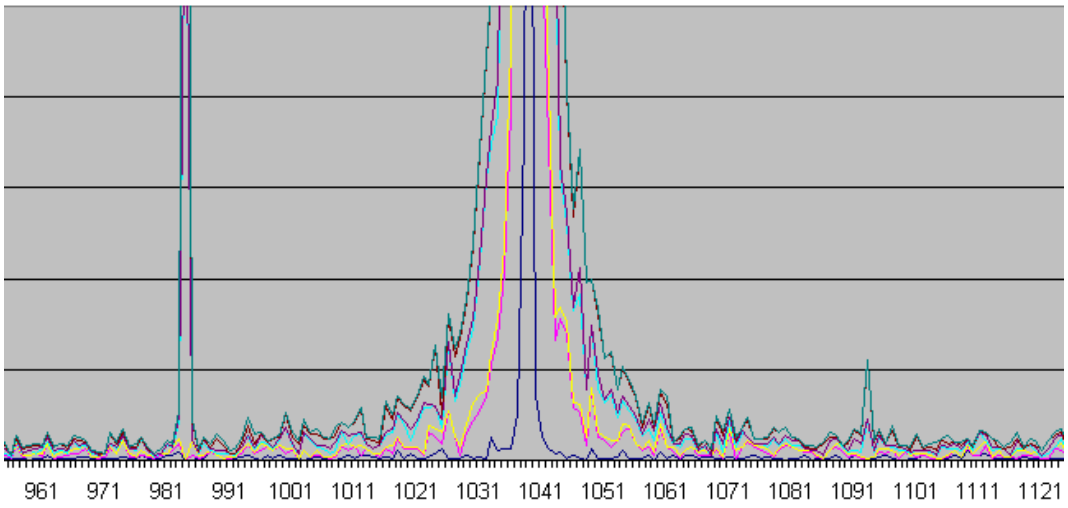

Figure 5.14: Beamforming module implementation for the FFT beamforming mode. Forms four beams from two antenna inputs.

Chapter 6

Results and Analysis

The ETA radio telescope processes data synchronously in real-time as digitally sampled radio signals pass through the RCC and are stored on hard disks in the Acquisition PCs. The RCC implementation decodes 24 antenna data streams, processes, combines, and reformats the streams for acquisition. Sixteen Xilinx FPGA development boards make up the RCC which breaks down into a two-level hierarchy of twelve outer and four inner nodes. Outer nodes synchronously process digitally sampled antenna streams and transmit the result to RCC inner nodes. Inner nodes receive, synchronize, and sum multiple streams. The resulting streams are reformatted and transmitted to the acquisition PCs for recording. The RCC implementation demonstrates how multiple FPGAs can be partitioned to perform real-time, streaming computation- and communication-intensive application. This chapter analyzes the RCC implementation and discusses system partitioning, synchronization, computation performance, communicational performance, resource utilization, resource efficiency, scalability, and design improvements.

6.1 Partitioning

Partitioning is an important consideration in any system level design and specifically to streaming applications. Data streaming requires that in-line processing occur continuously and with deterministic latencies. The added processing latency is not important in the ETA radio telescope, but latency must be compensated for by parallel processing or data buffering. The RCC data path is partitioned to process data continuously as it is streamed through the system. This section contrasts RCC partitioning with to other processing platforms.

The RCC implementation uses FPGAs to match data processing with data flow through the system. Figure 6.1 shows the processing required to perform an FFT and form eight single-polarization beams on twelve input streams. To complete the processing for the second polarization, the processing path is replicated. Each input is processed independently until the summation and can be thought of as leaf processing. Leaf processing is represented in Figure 6.1 by the box outlining the FFT and multiplication elements. The FPGA development boards used in the RCC allow the processing flow to map directly to physical hardware. Outer nodes perform the processing for two input streams or two leaves from Figure 6.1, limited by the available I/O connections and FPGA resources.

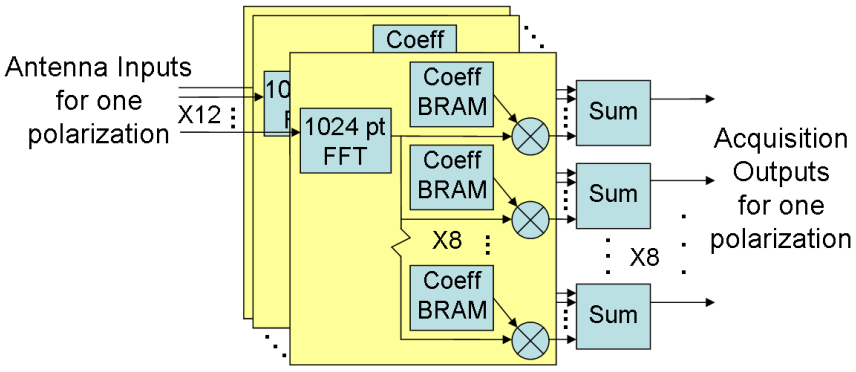

Figure 6.1: Data processing flow for a single polarization.

A hierarchical summation is used to combine signals following leaf processing. In accordance with data flow through the RCC, the outer nodes also sum the two processed leaves.

Subsequent summation and signal combining takes place in the RCC inner nodes where input values are received, synchronized, and summed. Ideally the summation shown in 6.1 would take place in a single inner node, but due to physical connection availability and disk throughput limits, two inner nodes are required to sum the outer node signals. The actual summations are partitioned between the two inner nodes, each node performing the summation of beams. FPGAs further allow the direct mapping of the data flow to physical hardware inside each node as described in Chapter 5. Other processing platforms such as general purpose processors or DSPs are not as well equipped to handle data flow processing.

Processors by nature perform operations sequentially and have evolved peripherals to funnel operations more quickly through the sequential device. Supercomputer and multi-core processors have become more parallel, but still rely on the sequential operation of each element. Processors in a variety to types provide a static architecture to which data flow must be adapted. When applied to stream processing, a processor-based architecture must have sufficient processing elements, bandwidth, and buffering to compensate for indeterminate process latencies or delays which may be introduced by the complexities of the processor, communication, and memory interactions. The FPGA architecture used for the RCC design is more capable of adapting system data flow to physical hardware and thus better than processor-based architectures for implementing real-time, stream-based processing.

While processor architectures are static, ASICs can be designed for custom applications. An ASIC can be designed to meet the processing requirements of a system more efficiently in terms of latency and performance than either processors or FPGAs. Even though ASICs provide superior performance, they are rarely used in low-production or low-cost applications due to the required development cost and time. The RCC design uses commercially available FPGA boards that are readily available at a low cost. The use of FPGAs in the RCC design also allows the system to be rapidly developed and customized to the application.

6.2 Synchronization

In addition to aiding system partitioning and data flow, the FPGA architecture chosen for the RCC facilitates synchronization throughout real-time processing. The digital receiver boards use globally distributed clock and reset lines to sample and transmit data to the RCC. Each of the S25 sourced data stream therefore produces and transmits data to the RCC at an identical rate. The distributed clock used to sample data is vital to synchronizing data flow through the system because even a small difference in clock or processing rate will result in buffer overflows and data corruption over time. In order for the RCC to synchronize the data streams, each S25 uses the global reset to synchronously produce a start of frame (SOF) signal and sample counter. The RCC uses the SOF signal and sample counter to synchronize multiple data streams.

Synchronization within the RCC takes place both by virtue of the FPGA processing architecture and by implemented synchronization stages. The encoded stream received by the RCC outer nodes is by nature a combination of two synchronized antenna streams. To maintain synchronization through the RCC outer nodes, the streams are processed in parallel and maintain identical latencies. Samples produced at the output of an outer node are thus synchronous between beams and beam samples. Multiple signals are then combined in the RCC inner nodes and synchronization must be done explicitly.

While all of the data streams are generated synchronously in the S25 digital receiver boards, an unknown and unequal amount of transmission delay is added to each path. The transmission delay results from a combination of physical transmission line differences, cycle-by-cycle protocol differences, and jitter. The RCC inner nodes must therefore use the encoded SOF and sample counter to synchronize the six input data streams. The synchronization is done by buffering the input data streams and aligning the SOF and counter signals to identical values. Initially, the synchronization module continuously attempts to synchronize all channels. While synchronization should not be lost following the initial link setup, a loss of synchronization would result in output data corruption until all of the links are re-established. Any

loss in synchronization is reported through the UART system interface and can be monitored throughout a data acquisition. The RCC inner nodes are capable of synchronizing data streams that are offset by less than 40 data cycles (1.28 μs). In practice the variance between data streams is less than 5 data cycles and can be read from the UART control interface. A variance greater than 5 data cycles likely indicates incorrect system processing in upstream nodes.

Following synchronization in the RCC inner nodes, the streams are processed, combined, and stored to disk in the acquisition PCs. The data from multiple beams is stored separately between the four acquisition PCs and requires synchronization to perform further processing outside the ETA system. To enable synchronization in off-line processing, the SOF and sample counter are delayed to match inner node processing latency and then stored to disk with their corresponding data samples. The stored SOF and sample counter are used in off-line processing to correlate between the stored data from separate acquisition PCs. Synchronization is done by aligning the SOF and counter values, but differs in that off-line synchronization and processing is done through software rather than implemented in FPGA hard logic.

The RCC implementation maintains processing synchronization through the attributes provided by an FPGA-based architecture. The data path is customized with predictable and identical latencies and the inner nodes are able to compensate for small transmission delay differences which occur. The FPGA architecture enables this synchronization by allowing parallel processing or flow paths and without inversely affecting processing time or data streaming. Synchronization in processor-based systems would require extra processing time and may therefore adversely affect performance if there is not sufficient processing headroom available.

6.3 Computational Performance

The RCC performs the FFT and beamforming computations which require substantial multiply and accumulate operations (MACs) and on-chip interconnect. FPGAs are beneficial in performing the required processing because they allow for greater parallelization, buffering, and custom on-chip interconnect. FPGAs do however operate at a much lower processing frequency than processors including DSPs. Furthermore since the FPGA architecture can be customized to fit the application, fewer hardware overheads are required. This section discusses the computational performance of the RCC.

In addition to encoding, synchronization, and communications logic; the RCC uses 88 multiply-accumulate blocks per outer node, or 1056 aggregate, to perform the real-time FFT and beamforming operations. To allow the FFT to operate on 1024 data samples, the samples are first buffered and then streamed into the FFT and beamforming logic. The result is computation which operates in a burst and wait schedule. Under burst operation, each MAC will perform operations at the processing clock frequency of 62.5 MHz. In aggregate the RCC is capable of performing 66 GMACs/s under burst operation, but normally operates at an average rate of 7.9 GMACs/s. The RCC architecture and internal interconnect allow each of the MAC units to be utilized continuously during streaming operation.

Although high-end processors or DSPs are capable of achieving similar MACs/s performance, their architectures require extra operations and processing cycles to handle driver-level interactions, memory management, and I/O interfacing. The processor interactions and calculations would therefore be much less efficient and require a significant amount of processing overhead to allow for real-time processing. Conversely, system interfacing and communication in the RCC do not adversely affect processing performance. Internal to the FPGAs, large amounts of routing resources make it possible for processing units to be connected to many other components with wide data paths. The interconnection matrices available within FPGAs further aids processing efficiency and streaming in RCC system processing. Communication between FPGAs utilize separate, customized logic modules and do not af-

fect processing performance in the RCC. FPGAs achieve high performance by parallelizing computation modules rather than time-multiplexing computation at high rates.

6.4 Communication Performance

The communication capability of the RCC also aids ETA's ability to perform stream processing in real time. In order to achieve high processing performance, data must be streamed to and from processing blocks at high throughput rates. The FPGAs used in the RCC design provide ample I/O connections and allow for parallel, high-speed serial, and LVDS transmissions. Furthermore, due to the complex nature of the FFT and beamforming operations, the RCC acts as a router to allow processed data to flow and be combined through the system. This section discusses the communication performance of the RCC.

The connection between the digital receiver boards and the RCC outer nodes uses a source-synchronous, 5-bit parallel, LVDS link. Both LVDS signaling and the use of shielded Blue Ribbon cable reduce the effects of interference over the link. The connection allows for a stable, high-throughput link which transmits data to the RCC at 37.5 MBs/s per outer node or 450 MBs/s aggregate. The integrity of this link is measured by producing counter data from the digital receiver boards and using an RCC test mode to verify correct data reception in the RCC. Data errors are then recorded and read back through the UART control interface. Throughout testing no bit errors have been detected over this link making the BER immeasurably small and negligible.

The second high-throughput network connects the RCC outer and inner nodes. As a result of the FFT, beamforming, and rounding operations in the RCC outer nodes, the data width is expanded from 14 bits (7-bit I, 7-bit Q) to 30 bits (15-bit I, 15-bit Q). In addition, the number of samples increases by a factor of four to form the eight single polarization beams produced in each outer node. To accommodate the data expansion of outer node processing, two 120 MB/s links per outer node (2.88 GBs/s aggregate) are required for connecting

outer and inner RCC nodes. The FPGA MGTs allow for high-speed, self-synchronous serial connections which are made with InfiniBand interconnect. The MGTs allow a 32-bit parallel input to be serialized and transmitted at 20 times the 125 MHz clock rate. The MGTs also allow adjustments to improve signal quality as discussed in Chapters 2 and 5. The light-weight Aurora protocol is used over the InfiniBand links because it requires a relatively small amount of processing resources and communication overhead when compared to other protocols. The main considerations of the Aurora protocol is that it uses 8b/10b encoding to avoid a DC bias in the transmission line and requires a periodic clock correction sequence to be transmitted. Both of these elements are accounted for with normal process buffering and sufficient bandwidth available. Due to the streaming nature of the ETA system and hardware resource limitations, protocols which use data retransmission or additional bit-error correction could not be used. After establishing and tuning system links, testing is also performed by driving counter data over the links and verifying reception at the RCC inner nodes. Testing results show that links with increased BERs often result from a poor connection between InfiniBand cables and adapter board receptacles. Error prone links are remedied by re-seating connectors or using an alternate port or adapter board. Additional testing shows that a BER less than 10^{-12} is achieved by the RCC network, which is within the InfiniBand specification and will result in no more than a few bit errors in an entire 800 GB recorded data set.

The final link discussed in this section connects the RCC inner nodes to the acquisition PCs. Since data is combined through the RCC inner node processing, the PC throughput requirement is much lower than the RCC network requirement. Each inner node combines six input streams and rounds the value to extract the most relevant bits which are then stored to disk. The link from the inner nodes to acquisition PCs is created using a source-synchronous, 16-bit parallel LVDS connection that transmits 60 MBs/s. Similar to the S25 to RCC outer node connections, the links operate at a slower clock frequency and do not have a measurable BER. The RCC design takes advantage of the communication resources available within the Virtex-II Pro FPGA architecture to establish high throughput data

links between processing nodes. Both parallel and high-speed serial connections are used in the RCC design and establish reliable communications that enable real-time, stream-based processing.

6.5 Resource Efficiency

The Virtex-II Pro FPGA provides ample processing and communication hardware for the outer and inner node implementations. A majority of the real-time processing takes place in the outer nodes and thus requires more processing resources than the RCC inner nodes. Table 6.1 summarizes the resources used for the FFT Beamforming outer and inner node implementations as well as the total resources available in the Virtex-II Pro FPGA. As shown below the outer nodes use 70% and 82% of the available registers and LUTs which makes additional expansion of the outer node design difficult to route. The usage of other FPGA resources including multipliers, block RAMs, I/Os, and MGTs is also summarized in Table 6.1. While a large portion of the available resources are used for the outer node implementation, performance and throughput are limited by the to disk throughput as opposed to available FPGA resources. This section evaluates the efficient use of logic based on the resources used for processing rather than communication.

Table 6.1: Resource Utilization for the FFT Beamforming implementation.

FPGA Nodes	Registers	4-input LUTs	Multipliers	Block RAMs	I/Os	MGTs
RCC Outer	19,402	22,505	88	73	28	6
RCC Inner	4,852	8,069	0	5	58	6
Available	27,392	27,392	136	136	556	8

The efficiency of the RCC is determined according to how efficiently logic is used to implement the required processing and data transmission. While Aurora requires less logic than other serial protocols, its overhead in terms of logic usage is expected to reduce efficiency.

Furthermore, the efficiency is further decreased by the two-level architectural design of the RCC. To determine the RCC logic efficiency, we must first define a single FPGA system that can be used for comparison with the RCC as a whole. The single FPGA system is defined by its maximum capability in terms of numbers of inputs and beams formed. Using a single ML310 FPGA, two antenna streams could be combined to form a maximum of four beams. These limitations arise from the limited ML310 input connections and EDT bandwidth. It is important to note that while this is a feasible single FPGA implementation, the quality of results is vastly decreased by the use of only two antennas. The single FPGA system is only used for the purpose of evaluating logic efficiency, and it would not be implemented on an actual system.

The fully implemented RCC functions on 24 antenna inputs to form a total of 16 single pol beams. The number of antennas used is scaled by 12 and the number of beams formed is scaled by a factor of 4. To correct for this, the single FPGA system is modified to form only two beams. This increases the scaling factor of beams formed to eight. Due to multiplexing and the nature of the beamforming algorithm, the increased number of antenna streams increases the required implementation logic more rapidly than an increased number of beams formed. For this reason, the non-ideal system scaling will be ignored and the RCC functionality will be estimated to increase by a factor of 12 from the single FPGA implementation. Table 6.2 summarizes the amount of logic required by the single FPGA implementation and each of the RCC nodes in terms of logic registers per implementation.

Table 6.2: Logic registers required by each implementation.

FPGA Nodes	Synchronization Logic	Aurora Protocol Logic	DSP and System Logic	Total Required Logic
Single FPGA	-	-	20,055	20,055
RCC Outer	-	585	18,817	19,402
RCC Inner	1,858	1,756	1,238	4,852

As seen in Table 1, there is a small overhead added to the RCC from the Aurora and synchronization logic. To compute the scaled logic efficiency of the RCC, the number of logic registers for a single FPGA design can be taken from Table 1. The total parallel logic registers is:

$$\begin{aligned} \text{parallel logic registers} &= (12 \cdot \text{outer logic registers}) + (4 \cdot \text{inner logic registers}) \\ &= (12 \cdot 19,402) + (4 \cdot 4,852) \\ &= 252,232 \end{aligned} \tag{6.1}$$

The scaled logic efficiency will be calculated by dividing the single ML310 logic registers by the scaled parallel ML310 logic registers:

$$\begin{aligned} \text{Scaled Efficiency} &= \frac{\text{single logic registers}}{\text{parallel logic registers/functionality increase}} \\ &= \frac{20,055}{252,232/12} \\ &= .95 \end{aligned} \tag{6.2}$$

Hence the RCC logic is about 95% efficient which is expected given the large amount of logic required by the FFT and beamforming operations. The existing architecture could not be modified to increase logic efficiency due to the necessity of Aurora protocol and synchronization logic. However, the cost of the system could be decreased by combining the inner node logic on four of the outer RCC nodes. While this would require about the same amount of logic, it would utilize all eight of the InfiniBand connectors on the four combined node ML310s. This implementation would not affect the logic usage efficiency but would decrease the cost of the overall system by requiring the purchase of only twelve ML310 boards. Furthermore, these combined nodes would require approximately 20,055 logic registers or about 73% of the available FPGA logic on the ML310 board. Combining the inner and outer node implementations would add a small amount of design complexity but would lower equipment costs.

6.6 Scalability

The concept of scalability is important for evaluating systems which may be reproduced or expanded to incorporate more functionality. Scalability should be considered as this drastically impacts the cost effectiveness of expanding or implementing additional systems. In evaluating the scalability of the ETA radio telescope, three scalable areas will be discussed. The number of antennas used, the number of beams formed, and the duration of continuous observation are key factors in evaluating the scalability of the ETA radio telescope. Expanding the capability of the system in these three areas leads to increased noise immunity and observation precision, expanded observation area and precision, and improved likelihood of detecting astronomical events respectively. The following sections describe in greater detail how the expanding system capability in theses three areas affects the scalability of the ETA radio telescope.

6.6.1 Scalability of Antennas

As mentioned earlier, increasing the number of antennas can improve noise immunity and overall precision of a sensing device. However expanding the number of antennas requires additional connectivity and processing capability. When adding antennas to the system, the required input and output connections contribute most to the need for additional hardware. The addition of two antennas requires that an S25 digital receiver node and an ML310 outer processing node be added to the system. After adding four ML310 outer nodes, the two-level RCC hierarchy becomes saturated as all of the available FPGA gigabit transceivers and InfiniBand connections would be utilized. Scaling the system beyond 32 antennas requires an additional level in the RCC hierarchy. The combination of node and hierarchical growth means that the cost of adding more antennas to the system increases as $n \cdot \log_8(n)$, where n is the number of antennas in the system. (Note: Cost increases as the log_8 due to the fact that each ML310 board allows for eight InfiniBand connections.) This shows that while

the ETA radio telescope design is scalable in terms of increasing the number of antennas, it would be very costly and at some point become unmanageable due to the number of nodes in the system. The ETA radio telescope architecture is better suited to systems with fewer than thirty-two antennas.

6.6.2 Scalability of Beams Formed

Scaling the system in terms of forming additional beams increases the amount of sky which can be viewed and improves observation precision. Increasing the number of beams formed through system processing is another key feature that can be scaled. Forming additional beams requires more logic and system bandwidth. In modes other than FFT beamforming, only a small amount of the FPGA resources are used. As a result, logic or hardware devices would not be a limiting factor. While the FFT beamforming mode utilizes significantly more FPGA resources, room for additional beamforming components is available. Furthermore the current and added hardware instances could be multiplexed to form additional beams. This is possible since the beamforming hardware is active for only one eighth of available processing cycles. Modifications to form more beams would require a small amount of added hardware other than the expansion of the current coefficient tables. Based on this analysis, a large amount of additional processing or beamforming could be performed in the RCC nodes without adding additional nodes.

While the current RCC hardware can provide sufficient processing for additional beamforming operations, the system bandwidth does not allow for the formation of additional beams without added system components. Forming additional beams requires more data to be transmitted through the InfiniBand and EDT links. The transmission of two beams requires approximately one quarter of the InfiniBand bandwidth and half the EDT bandwidth. The current implementation of the ETA radio telescope is limited to a total of 16 single-pol beams by the EDT connection bandwidth. Additional bandwidth and therefore formed beams could be achieved by increasing the number of inner RCC and acquisition PC nodes. The addition

of one inner RCC node and one acquisition PC node allows for the generation of an additional four beams. This proportional increase would continue until a maximum of sixteen RCC inner and acquisition PC nodes were added to the system giving the capability to form 64 single-pol beams. It would not however be practical to create a system with 64 individual beams as this would cover the viewable sky many times over and yield little quality increase. The cost of increasing the number of beams is then proportional to the number of beams added to the system and results in an order m cost and hardware increase, where m is the total number of beams formed. The system is thus scalable with respect to the number of beams that can be formed within a realistic upper bound.

6.6.3 Scalability of Observation Time

The duration of continuous observing time is another important factor to consider when determining the scalability of the ETA radio telescope. As observation recordings are made over a longer duration in time, it becomes more likely that interesting phenomenon will be observed. Extended recording duration can be achieved by increasing the amount of hard disk space available in the acquisition PC nodes. Currently the ETA radio telescope acquisition PCs are equipped with three hard drives that provide approximately 600 GB of disk space on each PC. Upgrading these hard drives to 500 GB and adding a forth hard drive to each system can increase the available disk space to 2000 GB per PC. A forth disk would help sustain throughput as data is stored on inner tracks, but testing is required to establish sustained throughput measures. Assuming sustained throughput capability, the extra disks would essentially triple the amount of data that could be stored and thus increasing the acquisition duration. Beyond this, acquisition times could also be increased by adding inner RCC and acquisition PC nodes as described in the previous section. The acquisition start times could then be staggered to allow for a continuous stream of data to be stored on additional PCs. As mentioned previously, a maximum of sixteen acquisition PCs could be added to the system which is sufficient to provide continuous observation recording.

Through increasing the disk space of each PC and staggering acquisition start times, more than eight hours of continuous observation can be recorded without storing data to external tape drives. By storing the data to tape in a cyclical fashion, one set of acquisition PCs would always be available to start an acquisition as another set completes one. Through this method, observation could continue as long as the tape disks were changed frequently. While continuous recording seems ideal, in practice continuous recording would not be utilized due to large amounts of daytime radio frequency interference. The best data recordings are taken during times of little RFI which generally occur between midnight and the early morning. During this six hour window, little interference is observed in data recordings and continuous observation would add value to the system. Outside of this time frame, it is likely that a majority of data would be overrun by RFI and be unusable. For this reason, it is likely that expanding the acquisition duration to six hours would provide the maximum benefit to the ETA radio telescope. Six hour observation times could be achieved by increasing hard drive space and adding four RCC inner and acquisition PC nodes. Due to availability and low cost of hard drives, the incremental cost of extending observation time is significantly less than the cost of scaling the system in terms of antennas or beams formed.

6.6.4 Scalability Summary

The ETA radio telescope system architecture is scalable in terms of increasing the number of antennas, the number of beams, and the length of acquisitions. However the current implementation allows for sufficient antennas, beams, and observation lengths to produce accurate and viable recordings. Better noise discrimination would result from replicating this system in various locations. Using this method, multiple sets of data could then be combined to confirm observed phenomenon and reject RFI. While reproducing the ETA system is not considered scalable in the traditional sense, it would provide for additional functionality that could be used to address any of the three scaling issues presented above. Scaling the number of ETA systems as a whole could provide additional antenna inputs, beams, or observation

time and would have an infinite scalability proportional to the number of systems created. It is important to note that while this system architecture is scalable on its own; the cost of scaling it is expensive and becomes difficult to manage, as with most cluster architectures. Scaling the number of systems as a whole is a much better method to improve research observations and has a linear cost and complexity.

Chapter 7

Conclusions

7.1 Summary and Contributions

The ETA radio telescope is designed to detect transient pulses within the 29 to 47 MHz frequency range and requires a large amount of processing on the 24 antenna inputs. The ETA architecture is designed from COTS components which enables low-cost, rapid development and migration to future technologies. The processing is performed by 28 FPGA boards, sixteen of which constitute the RCC. The RCC is responsible for performing the FFT and beamforming computations in real-time as antenna data samples are streamed from antennas to storage disk. The use of FPGAs in the RCC architectures provides the high computational and communication throughput required by the ETA system. The contributions of this thesis are summarized and described below.

Thesis Contributions

- Development of a flexible, high performance FPGA cluster from COTS components.
- Partitioning of a multi-FPGA stream-processing system.

- Global and local synchronization using system, source, and self synchronous methodologies.
- High performance stream-computation in an FPGA cluster including large amounts of multiply-accumulates and parallelism.
- High throughput stream-communication in an FPGA cluster including both parallel and high-speed serial communication.

The RCC implementation demonstrates how an FPGA architecture can be created which maps directly to the processing data flow of a system. The complex computation performed by the RCC requires a large amount of processing and communication interconnect capability. With the use of COTS FPGA development boards, the RCC establishes a tiered hierarchy of processing nodes and high-bandwidth links which matches the processing data flow. The FPGA configurations also contain processing modules and structures specific to the ETA processing flow. Processors and DSPs do not allow for such customized data-paths. ASICs are also capable of providing custom processing, but are too costly and time-consuming to develop.

Another aspect of the processing performed in the ETA system is that all samples and antenna data streams must be sourced, processed, and stored synchronously. The system functions with both global and local synchronization divided between the processing nodes. A sampling clock and reset are globally distributed to the digital receivers and ensure identical sample sampling rates and identification values between boards. The identification value is then used at various point to synchronize processing through the system or data stored on disk. The FPGA architecture facilitates synchronization within the ETA system by allowing parallel processing paths and predictable latencies which differ by less than 5 cycles.

Due to the parallel processing components within the FPGAs, the RCC is capable of performing computation at 66 GMACs/s during bursts and 7.9 GMACs average. Added computation beyond that is performed in each node to synchronize, combine, send, and receive

data without impacting the computational performance. While high-performance processors and DSPs are capable of achieving this level of performance, they would required custom board design and would increase cost and development time. The FPGA-based RCC provides an architecture that can be customized and developed rapidly at a lower cost than other custom design solutions.

The computational ability of the RCC is possible only through the high communication bandwidth available in the RCC. The RCC aggregate input and output data rates of 450 MBs/s and 240 MBs/s is achieved through parallel, source-synchronous connections that are 5-bit and 16-bit parallel connections respectively. The network between RCC outer and inner nodes consists of self-synchronous serial connections, and streams data at an average rate of 2.88 GBs/s which is about half of the maximum network maximum bandwidth. To allow the links to be highly noise immune, several precautions are taken on both the parallel and the serial links. The parallel connections use LVDS signaling over shielded interconnect and the serial connection implements 8b/10b encoding and LVDS signaling with proper adjustments to the differential voltage and applied pre-emphasis for the length of cable used. The BER of the RCC and ETA system is minimal and on the order of 10^{-12}. The RCC also utilizes a custom control interface for monitoring or changing system status or operation. Through the control interface fault prone links can quickly be identified and repaired.

The RCC achieves high communication bandwidth without the use of complicated signaling protocols or overheads. Direct connections for the parallel interfaces only requires a small amount of data reformatting at transmission. The protocol used over the serial link is a light-weight Aurora protocol available from Xilinx. This protocol requires little overhead in terms of both signaling overhead, requiring only periodic clock corrections, and hardware resources, allowing the processing architecture to be 95% efficient.

It has also been shown that the RCC can be scaled to 32 antennas, 64 beams, or 24-hour observation with minor adjustments to the RCC architecture. While this is the case, better quality of results may be achieved through creating multiple ETA systems at different

locations. Creating multiple telescopes aids scalability in terms of antennas, beams, and observation time as well as creating results which are independent and thus introduces anti-coincidence into observed phenomena.

7.2 Conclusions

The ETA radio telescope is successful at employing COTS FPGA technology to performing useful, real-time computation. The RCC architecture demonstrates how multiple FPGAs can be used to create custom data processing in an application to achieve high computation and communication bandwidth. The RCC also demonstrates the use of global and local synchronization through a processing system which is highly parallel and application specific. The use of COTS FPGAs has enabled the rapid development of the ETA system while keeping development costs down. The reconfigurable architecture also allows for design flexibility and capability to create custom interfaces for monitoring or changing system operation.

7.3 Future Work

The ETA system has recorded data which is currently being processed to search for transient pulses. While no detections have been reported to date, the results from the ETA system have enabled scientific contributions to the study of primordial black holes and low-frequency radio astronomy. Additional funding has been received to create a second instrument ETA2. The upgraded instrument currently plans to use more powerful FPGA devices. ETA2 will be housed in a mobile trailer which will facilitate development and relocation of the telescope. Ongoing work includes evaluating the recorded results from the ETA system, seeking new funding opportunities, and completing a second ETA system.

Work has also been done to create a software model of the FFT operation such that test vectors can be generated to validate proper FFT and FFT beamforming operations. Com-

pleting and testing a software model of full system operation would also be beneficial to the development and validation of ETA2. System improvements to the ETA FPGA design include using processing resources more efficiently and passing the originally sourced counter through the entire system. Currently the counter is reproduced at the output of the FFT block and does not pass directly from the digital receivers to the PC acquisition nodes. Following this modification, it will be possible to rapidly demonstrate that the digital data passes correctly through the system from receivers to disks.

Bibliography

[1] Marco de Vos, "LOFAR: The First of a New Generation of Radio Telescopes," *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05).*, vol. 5, pp. v/865–v/868 Vol. 5, 18-23 March 2005.

[2] Steven W. Ellingson, "Antennas for the Next Generation of Low-Frequency Radio Telescopes," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 8, Aug. 2005.

[3] S. W. Hawking, "Black Hole Explosions?" *Nature*, vol. 248, pp. 30–31, Mar. 1974.

[4] R. Narayan, B. Paczynski, and T. Piran, "Gamma-Ray Bursts as the Death Throes of Massive Binary Stars," *Astrophysical Journal*, vol. 395, pp. L83–L86, Aug. 1992.

[5] Michael Kavic, John H. Simonetti, Sean E. Cutchin, Steven W. Ellingson, and Cameron D. Patterson, "Transient Pulses from Exploding Primordial Black Holes as a Signature of an Extra Dimension," Jan. 2008, http://arxiv.org/abs/0801.4023.

[6] Cameron D. Patterson, Brian S. Martin, Steven W. Ellingson, John H. Simonetti, and Sean Cutchin, "FPGA Cluster Computing in the ETA Radio Telescope," *2007 Int'l Conf. on Field-Programmable Technology*, pp. 25–32, Dec. 2007.

[7] B. F. Burke and F. Graham-Smith, *An Introduction to Radio Astronomy.* Cambridge University Press, 2002.

[8] "First International LOFAR Station in Effelsberg Starts Observations," Dec. 2007, http://www.lofar.org/p/news/111207.htm.

[9] "LWA Home Page," Dec. 2007, http://lwa.unm.edu/.

[10] "MWA Home Page," Dec. 2007, http://www.haystack.mit.edu/ast/arrays/mwa/.

[11] D. Lorimer and M. Kramer, *Handbook of Pulsar Astronomy*. Cambridge University Press, 2005.

[12] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid, *Signals and Systems*. Prentice Hall, 1996.

[13] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, Apr. 1965.

[14] Van Veen, B.D. and Buckley, K.M., "Beamforming: a Versatile Approach to Spatial Filtering," *ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine]*, vol. 5, no. 2, pp. 4–24, April 1988.

[15] Xilinx, http://xilinx.com.

[16] Altera Corporation, http://altera.com/.

[17] Xilinx, "Virtex-II Pro and Virtex-II Pro X FPGA User Guide," Nov. 2007, version 4.2.

[18] Xilinx , "ISE Product Brief," 2008.

[19] Xilinx, "Xilinx CORE Generator System Overview," 2008.

[20] Xilinx , "ChipScope Pro Tool Brochure," 2008.

[21] Kenneth L. Rice, Christopher N. Vutsinas, and Tarek M. Taha, "A Preliminary Investigation of a Neocortex Model Implementation on the Cray XD1," May 2008, http://sc07.supercomputing.org/schedule/event_detail.php?evid=11062.

[22] Rob Baxter and others, "Maxwell – a 64 FPGA Supercomputer."

[23] Supercomputing Online, "Xilinx Powered SGI RASC Platform Accelerates Bioinformatics Application 900X," Dec. 2007.

[24] National Semiconductor, "LVDS Owner's Manual," 1997.

[25] Xilinx, "Aurora Product Specification," 2008, version 3.0.

[26] Altera Corporation, "Stratix EP1S25 DSP Development Board," Dec. 2004, version 1.6.

[27] Steven W. Ellingson, John H. Simonetti, and Cameron D. Patterson, "Design and Evaluation of an Active Antenna for a 29–47 MHz Radio Telescope Array," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 826–831, Mar. 2007.

[28] Altera Corporation, "Stratix Device Family Overview," June 2008, http://www.altera.com/products/devices/stratix-fpgas/stratix/stratix/overview/stx-overview.html.

[29] Tyco Electronics, "BLUE RIBBON Coax Assemblies Flexible, high density, high data rate," June 2008, http://www.precisionint.com/HighSpeedData/BlueRibbon/.

[30] CS Electronics, "Infiniband Cables," June 2008, http://www.cs-electronics.com/infiniband-cables.htm.

[31] Xilinx, "RocketIO Transceiver User Guide," Feb. 2007, version 3.0.

[32] Hewlett-Packard, IBM and Quantum, "LTO Tape Drive," June 2008, http://www.lto-technology.com/default.php.

[33] Engineering Design Team (EDT), "PCI CDa," 2008.

[34] Brian S. Martin, "An Analysis of Parallel FPGAs in a DSP Application," Jan. 2007, http://www.ece.vt.edu/swe/eta/rcc/070117_Martin.pdf.

[35] Lava Computer MFG, June 2008, http://www.lavalink.com/dev/index.php?id=60.

[36] Debian, "Minicom," Feb. 2008, http://alioth.debian.org/projects/minicom/.