On the Instrumentation of the Omniscope

by

Nevada J. Sánchez

S.B., E.E.C.S., Massachusetts Institute of Technology (2010)

S.B., Mathematics, Massachusetts Institute of Technology (2010)

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

June, 2011

Author _____
Department of Electrical Engineering and Computer Science
May 17, 2011

Certified by _____
Max Tegmark
Thesis Supervisor

Accepted by _____
Dr. Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

On the Instrumentation of the Omniscope
by
Nevada J. Sánchez

Submitted to the
Department of Electrical Engineering and Computer Science

May 20, 2011

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

An emergent technique known as 21-cm tomography has the potential to become the most sensitive tool we have for probing the early universe. It is expected to shed light on some of the most pressing questions in modern physics—such as the nature of dark matter. However, there are significant technical challenges involved in developing an instrument capable of 21-cm tomography. Radio telescopes are particularly well suited to the task. However, the cost of scaling a traditional radio telescope to achieve the necessary sensitivity is prohibitive. The Omniscope is an elegant solution to this problem. It is a new type of radio telescope that scales significantly better than traditional large array telescopes by using an innovative computational framework. I detail the implementation of many of the major subsystems of one of the very first Omniscopes ever built—including the digital correlator, the direct sequence spread spectrum channel encoding system and powerful system and data analysis software.

Thesis Supervisor: Max Tegmark
Title: Professor, MIT Department of Physics

Acknowledgments

I would like to thank the entire Omniscope team for all of their hard work in making the Omniscope what it is today. My supervisor, Max Tegmark, has been an invaluable source of guidance, not only in this project, but also in my career.

I give much thanks to Jon Losh and Hrant Gharibyan for spending late nights in the lab helping me produce plots and results for this thesis. Jon has been a remarkably efficient teammate and helped in the development of the important digital systems. Hrant provided most of the plots that I've used to characterize the performance of my phase switching system.

I would like to thank Andy Lutomirski for being a major source of technical support and for major contributions to the project, not the least of which being his primary authorship of the corner turning paper. I would like to thank Eben Kunz for his diligence in producing our very own analog components. I would also like to acknowledge, Adrian Liu, for his work in calibrating all of our Omniscope data, Ashley Perko, for her efforts in diagnosing the major problems with the Omniscope, Jack Hickish, for his unflinching willingness to provide us with technical support, Katherine Schutz, for helping us setup the Omniscope in the freezing cold, and everyone else who dedicated their time to produce the Omniscope we have today.

Last, but not least, I would like to thank my family, who has provided me with much support as I worked my way through MIT—including my wife, Ashley, who never hesitated to support me through all of my hardships. Finally, I would like to dedicate this thesis to my son, Hayden Sánchez, who has inspired me to reach far and enjoy life.

Contents

List of Figures

List of Tables

Chapter 1

Background

1.1 Introduction

An emerging technique known as 21-cm tomography, which maps neutral hydrogen in the distant universe via its emission of radio waves, has the potential to become the most sensitive tool in observational cosmology and is expected to help us better understand dark matter, dark energy, neutrinos and the origin of our universe. However, to observe the 21-cm background, we need a different type of telescope than the traditional radio interferometers of today.

Unlike the radio astronomy of the past, the goal of 21-cm tomography is not having the ability to discern stellar objects (which requires very fine angular resolution), but having the sensitivity to detect neutral hydrogen *in spite of* foreground objects. While current designs could be scaled to increase sensitivity, doing so would not be cost effective. This has prompted the need for a different type of telescope that can sacrifice angular resolution for increased sensitivity while maintaining a reasonable cost.

The Omniscope is precisely such a solution. By restricting array elements to lie within regular hierarchical grids, the Omniscope can replace the pairwise correlation step in a traditional radio interferometer (which multiplies the signals from all antenna pairs) with a more efficient Fast Fourier Transform (FFT). This reduces the computational cost (i.e. the cost of computational hardware and resources) from a scaling of $O(N^2)$ to a scaling of $O(N \log N)$ compared to a traditional radio interferometer (N is the number of antennas).

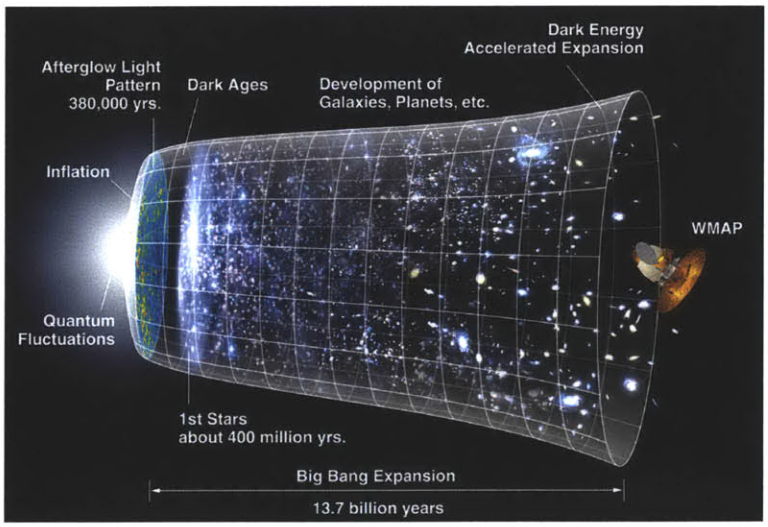


Figure 1-1: Time-line for the evolution of the universe [15]. On the far left is the Big Bang, marking the creation of the universe almost 14 billion years ago. The period between 300,000 and 150 million years marks the cosmic dark ages that we hope to observe using 21-cm tomography.

1.2 21-cm Tomography

1.2.1 Inflationary Cosmology

Figure 1-1 shows a rough picture of the evolution of our universe as we understand it today. Astronomy has so far been one of most important tools we have for probing the early universe. When we view a star that is 3 billion light-years away, we are not observing it as it is today, but as it was 3 billion years ago, when the light we receive was first created. Much of radio astronomy for the last half century has been done for the purpose of resolving these light sources (which individually occupy a very small fraction of the sky), so that we can probe the evolution of the universe after the formation of the first stars.

In recent years, the more pressing questions in physics have been focused on the state of the universe in its much younger stages. Moments after the Big Bang, the universe was filled with hot hydrogen plasma, in which matter (hydrogen) and energy (photons) were roughly in thermal equilibrium. Since the photons were scattered by the free electrons, the universe was opaque at this time. As the universe expanded, it cooled so that this balance was eventually broken. The photons no longer possessed enough energy to remove electrons from the hydrogen atoms. Thus, all matter became a transparent gas, through which we can observe the photons that were previously obscured. This period is known as *recombination.*

Because of recombination, we observe the *Cosmic Microwave Background* (CMB) at the furthest depths of our observable universe. The observation of the CMB ([13, 5]) has been hailed as one of the most important accomplishments in modern astronomy.

There remains, however, one period in the time-line of the universe that has been relatively difficult to accurately observe. Following recombination, the universe was filled with neutral hydrogen and remained so for about 150 million years until the period of *reionization* (in which the first large objects were formed that could reionize the hydrogen). This period of time between recombination and reionization is known as the *cosmic dark ages* and is so named because matter in the universe did not give off visible light at this time.

1.2.2 Hyperfine Splitting

Fortunately, due to a quantum mechanical process in neutral hydrogen called *hyperfine splitting*, there is a way to probe the cosmic dark ages. In quantum mechanics, *spin* is an elementary vector quantity of a particle (similar to charge or mass) that characterizes the magnetic dipole moment generated by the particle. Just as classical dipoles favor aligning with a surrounding magnetic field, particles favor aligning their spins with surrounding magnetic fields. For this reason, it is energetically favorable for the electron to have a spin that is anti-parallel to that of the proton.

The energy difference between the two configurations (parallel and anti-parallel spin alignment) is relatively small. Therefore, when an electron in a neutral hydrogen atom flips its spin to enter a more favorable configuration, it emits a photon with a relatively large wavelength. The wavelength associated with the electron's spin flipping in neutral hydrogen is approximately 21 cm. It is this spectral emission line that we will use to observe the distribution of matter in the cosmic dark ages. The process of reconstructing the distribution of matter by observing the 21-cm emission line is known as *21-cm tomography.*

Because the universe is constantly expanding, photons emitted at time periods in the past undergo stretching as the space they occupy expands. This causes them to be *red-shifted*, or reduced in frequency. If we know the precise frequency of the photon when it was emitted, then we can use the amount of red-shift to determine *when* the photon was emitted (equivalently, how distant the source is). Photons emitted at earlier times (greater distances) will have experienced more expansion and will therefore experience more red-shift. Thus, by observing particular narrow frequency bands, we can localize specific time

periods during the cosmic dark ages. A 21 cm wavelength photon will have a frequency of approximately 1.4 GHz. By the time it reaches current-day Earth, it will have been red-shifted to anywhere from 9 MHz to 200 MHz, depending on what period within the dark ages it was emitted. Observing spectral emissions near 200 MHz, therefore, will give us the distribution of matter in the universe as the first stars were beginning to form.

1.2.3 Why do we care?

There are many other important questions in modern physics (other than the origins of the universe) on which 21-cm tomography can shed light. For example, the origin and nature of dark matter is one of the most puzzling open questions in modern physics. Dark matter is an exotic form of matter that can so far only be observed by its (significant) gravitational influence on stellar objects. It played a large role in driving reionization, since reionization occurs when the gravitational pressure in clumped regions of matter generates enough energy to reionize hydrogen. Thus, it is useful to observe the behavior of the distribution of matter during the time periods leading up to reionization. The dependence of this cosmic clustering on both scale and time encodes information not only about dark matter, but also about dark energy and the inflation of our early universe. It is this clustering, (specifically, the spatial and temporal correlations of fluctuations in the neutral hydrogen signal) that we wish to measure with 21-cm tomography. Other applications include measuring the cosmological constant [8].

1.2.4 Technical Considerations for Performing 21-cm Tomography

Because the red-shifted 21-cm emission line lies within the radio frequency range, radio astronomy is an important tool for preforming 21-cm tomography. However, it provides a unique set of technical challenges when compared to radio astronomy of the past. In particular, the signals coming from the cosmic dark ages are as many as four orders of magnitude smaller than the foreground "noise" (stars, galaxies and nebulae).

1.3 Radio Interferometry

Because 21-cm wavelength radiation falls within the radio frequency band, it is natural to turn to radio telescopes to observe it. For this reason, it is helpful to develop the principles

behind how radio telescopes function—particularly, the radio interferometer. I describe the basic mathematics behind radio interferometry and how it allows us to form images of the sky. This development is largely based on the exposition in [16]. I then use this as a building block for describing the principles behind Fast Fourier Transform Telescopes.

1.3.1 From Baselines to Pixels

Radio astronomy is concerned with measuring the electromagnetic emissions of astronomical sources and synthesizing images or maps of them. Because of the nature of the processes that generate this radiation, astronomical sources are spatially incoherent (with the exception of certain bodies, such as pulsars and masers). This allows for the use of a very powerful relationship in coherence theory—the Van Cittert-Zernike theorem.

The Van-Cittert Zernike Theorem

Suppose we wish to observe a source $\mathcal{S}$ at a narrow frequency band centered at ν with bandwidth $\Delta\nu \ll \nu$. Furthermore, assume that $\mathcal{S}$ subtends a small enough angle in the sky that the sky can be considered essentially flat within the extent of the source (the flat-sky approximation). We wish to recover the intensity map $I(l, m)$ of the electromagnetic radiation as a function of the direction cosines l and m. The direction cosines are defined with respect to a coordinate system perpendicular to the line of sight. If a point has spherical coordinates θ (with respect to the zenith) and φ (the azimuth), then

$$\begin{pmatrix} l \\ m \end{pmatrix} = \begin{pmatrix} \sin(\theta)\cos(\varphi) \\ \sin(\theta)\sin(\varphi) \end{pmatrix}.$$

Suppose we measure the electric field induced by the source at two points P_1 and P_2 in the plane perpendicular to the line of sight to the source. Let $\mathbf{r}_1 = (x_1, y_1)$ and $\mathbf{r}_2 = (x_2, y_2)$ be the positions of P_1 and P_2 on this plane. Define

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} (x_2 - x_1)/\lambda \\ (y_2 - y_1)/\lambda \end{pmatrix}. \tag{1.1}$$

The numbers u and v represent the separation in wavelengths between P_1 and P_2, and we will see that these represent spatial frequencies. Let $E_1(t)$ and $E_2(t)$ be the measured

electric fields at P_1 and P_2 represented as phasors (i.e. $E(t) = |E|e^{i(\omega t+\phi)}$, where $|E|$ is the magnitude of the field, $\omega = 2\pi\nu$ is the frequency and ϕ is the phase). Then the mutual coherence function $\Gamma_{12}(u, v, \tau)$ is defined as the cross-correlation of $E_1(t)$ and $E_2(t)$,

$$\Gamma_{12}(u, v, \tau) = \langle E_1(t)E_2^*(t-\tau)\rangle = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} E_1(t)E_2^*(t-\tau)\, dt. \tag{1.2}$$

It turns out that we can use $\Gamma_{12}(u, v, \tau)$ to obtain the 2D Fourier components of $I(l, m)$, which are known as the *complex visibilities*. We can now state the Van Cittert-Zernike theorem as follows.

Theorem 1 (Van Cittert-Zernike Theorem). *The mutual coherence function for an incoherent source is equal to the complex visibility of the source when observed in the far field. That is,*

$$\Gamma_{12}(u, v, 0) = \Gamma_{12}(u, v) = \iint_{\mathcal{S}} I(l, m)e^{-2\pi i(lu+mv)}\, dl\, dm. \tag{1.3}$$

For a proof of this theorem, see [16]. The theorem relates the mutual coherence function to the two-dimensional Fourier transform of the intensity map and provides us with a powerful means for reconstructing $I(l, m)$. In particular, if we can sample $\Gamma_{12}(u, v)$ at sufficiently many points in the u-v plane, we can recover $I(l, m)$ by inverse Fourier methods.

Traditional Radio Astronomy

Traditional radio astronomy attempts to reconstruct $\Gamma_{12}(u, v)$ as accurately as possible so that (1.3) may be inverted and $I(l, m)$ can be faithfully recovered. Radio interferometers are instruments for computing (1.2). The basic building block of a radio interferometer is shown in Figure 1-2. Each baseline measured by the instrument represents a sample point (u, v) of the complex visibility function of a patch of sky—equivalent to the mutual coherence function of the two antennas as shown by the van Cittert-Zernike theorem. The visibility function is computed by sending the signals from each pair of antennas through a *correlator*, which is a subsystem that multiplies and integrates the two signals over time.

In order to measure many baselines at once, most radio telescopes are formed by a collection or array of antennas. Furthermore, as the Earth rotates, the baseline vectors change orientation with respect to the sky, so that even more baselines can be measured by

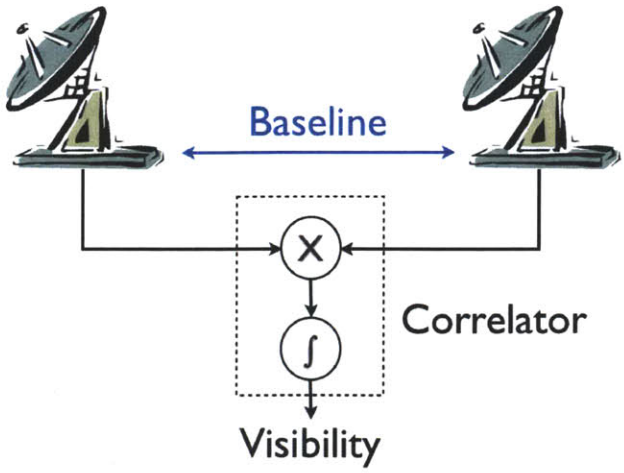


Figure 1-2: Correlator for a pair of antennas. The signal from the two antennas (separated by a vector known as the *baseline*) is multiplied and integrated over a sufficiently large time τ to produce a *visibility*.

combining measurements at different times of day.

Radio astronomy has traditionally been concerned with resolving stellar objects that subtend very small solid angles on the celestial sphere. In order to resolve them accurately, a large number of unique baselines are necessary. Furthermore, since the visibility function is the spatial Fourier transform of the intensity map, each baseline represents a particular spatial frequency component of the sky map. In order to resolve fine structures (such as distant sources within small solid angles), large baselines are necessary. Hence, the resolution of a radio interferometer is a function of the baseline distribution that it can measure.

Mapping Principles

In order to reveal the basic principles behind how one can recover images from radio interferometer measurements, we will simplify the problem by restricting to a one dimensional array. We will also assume that the measured visibilities are regularly spaced on the u axis with M baselines that are integer multiples of a distance d (which can be accomplished by a regular linear array with separation d). By the conjugate symmetry of the mutual coherence function, each visibility can be conjugated to produce another visibility in the opposite position with respect to the origin. We thus have $2M + 1$ points, $-M, \ldots, M$.

The Nyquist sampling theorem ([9]) can tell us what spatial information can be recovered

from the visibilities. Specifically, it is the spacing between samples (the sampling rate) that specifies the span of direction cosines that are measured, and it is the longest baseline that determines how finely this interval is sampled. Under the assumptions, (1.1) allows us to write (1.3) as

$$\Gamma_{12}(u) = \int_{-L}^{L} I(l) e^{-2\pi i l u} \, dl. \tag{1.4}$$

The fact that we are sampling $\Gamma_{12}(u)$ discretely implies that the intensity function $I(l)$ that we can recover is of finite duration L. Really, this is only an issue when we are under-sampled, because L must necessarily be less than 1 by the fact that it is a direction cosine. The sampling period of the angular spatial frequencies v is

$$\Delta v = \frac{2\pi\lambda}{d}.$$

The finite extent of $\Gamma_{12}(u)$, terminated by the longest baseline at $M\lambda/d$, means that $I(l)$ is also a discrete signal. Hence, we can rewrite (1.4) as a discrete Fourier transform,

$$\Gamma_{12}[n] = \sum_{m=-M}^{M} I[m] e^{-im}.$$

From the sampling theorem, we know that the width L of the image (the "bandwidth") is inversely related to the sampling period,

$$L = \left(\frac{2\pi}{\Delta v}\right) = \frac{d}{\lambda}. \tag{1.5}$$

Furthermore, we can see that if we apply a inverse discrete Fourier transform to (1.4), we recover M samples of the interval $[-L, L]$. Thus,

$$\Gamma_{12}[n] = \Gamma_{12}(n\Delta v) = \Gamma_{12}\left(\frac{2\pi n\lambda}{d}\right) \quad \text{and}$$
$$I[m] = I\left(\frac{mL}{M}\right) = I\left(\frac{md}{M\lambda}\right).$$

This expression makes clear the relationship between the baseline distribution and the data recoverable by the instrument. Baseline distributions with longer tails reveal more fine structure, and finer baseline distributions cover a larger area of the sky. There is a limit,

of course, to how finely distributed the baselines need to be. Indeed, the horizon lies at direction cosines $|l| = 1$. Thus, by (1.5) we gain maximum coverage whenever $d \leq \lambda$. For a radio telescope, this means separations on the order of meters.

In practice, traditional radio telescopes have been willing to sacrifice sky coverage for resolution. That is, long baselines have been preferred, even if they could not be placed finely enough to cover the whole sky. These baseline distributions are coarser than λ, and as a result, suffer from the problem of aliasing. Since $L < 1$, the points in $I[n]$ represent ambiguous points in the sky. This is analogous to the phenomenon by which higher frequencies become ambiguous with lower frequencies when a signal is sampled below the Nyquist rate. This can be described in terms of the *beam pattern* of the array, which quantifies the sensitivity of the instrument to signals coming from different directions. The presence of aliasing implies the presence of grating lobes in this beam pattern. To overcome the problem of aliasing, long baseline interferometers utilize parabolic dish receivers that are highly directional and mechanically track their sources —significantly suppressing the grating lobes. This is analogous to applying an aliasing filter to a signal, except that the "filter" selects an interval in the *space* domain, rather than the frequency domain.

1.3.2 Shortcomings of Traditional Radio Telescopes

With current radio telescopes, there are many difficulties in doing 21-cm tomography. Traditional radio telescopes were originally designed with the purpose of resolving objects that occupy a small angular fraction of the sky. To observe the cosmic dark ages, we need to observe a faint 21-cm background *in spite of* these stellar objects in the foreground. As one can imagine, it requires a remarkably sensitive instrument to be able to see 21-cm emissions through almost 13 billion years of collective matter formation.

The sensitivity of a radio interferometer is a measure of the effect of the instrument on the signal to noise ratio of the measurements. Specifically, it is a measure of the detector noise with respect to the desired signal. It has been shown that the detector noise is inversely proportional to the total collecting area of the array [17]. The form of the relationship is

$$C^{noise} \propto \frac{\lambda^2 T_{sys}^2}{A f^{cover} \tau \Delta\nu}, \tag{1.6}$$

where T_{sys} is the system temperature, A is the total collecting area of the array, f^{cover} is

the fraction of the sky that is covered and τ is the length of the integration. The sensitivity is inversely proportional to the detector noise.

The relationship in (1.6) reveals several challenges for the traditional approach to radio astronomy with respect to performing 21-cm tomography. In order to reach the level of sensitivity required to discern the 21-cm spectral line, very large collecting areas are necessary. The fact that f^{cover} is typically very small in traditional radio interferometers (as we saw in the last section) worsens this yet. These problems can be solved by drastically increasing the number of antennas—both to increase A and f^{cover}. However, this brings about another problem in computation.

For very large N (number of antennas), the computational costs of the correlators will dominate. This is because traditional radio interferometers are designed with the intent to correlate all $O(N^2)$ pairs of antennas. Thus, the cost of the computational hardware needed to build a radio interferometer scales as $O(N^2)$, which is not cost effective for the scales that will be needed to perform 21-cm tomography ($N \sim 10^4 - 10^6$). Because of this, we must direct our efforts towards modifying the techniques that we use to do radio astronomy. This is the inspiration behind the Omniscope.

1.4 Fast Fourier Transform Telescopes

We saw in the last section that radio telescopes can be great instruments for resolving stellar objects in the distant universe. However, as described, traditional radio telescopes face a significant cost barrier if they are naïvely re-purposed for performing 21-cm tomography. Specifically, we need enough antennas to supply a collecting area large enough for the required sensitivity. To accomplish this, we need a reasonable way to surmount the $O(N^2)$ computational cost. A fast Fourier transform telescope, as described in this section, provides an elegant solution to this problem that reduces the computational cost to $O(N \log N)$ while maintaining increased sensitivity.

1.4.1 Principles

The concepts behind the operation of a fast Fourier transform telescope (FFTT) are similar to those used in traditional radio telescopes in that there is a Fourier relationship between the sky intensity map and the measured correlations. However, we will see that we can

lift many of the approximations imposed by the traditional designs. We shall follow the development of the FFTT algorithm given in [15] for completeness.

For a fixed frequency $\omega = ck$, we can encode the information about the electric field at the position of a receiver (the sky signal) in a vector $\mathbf{s}(\hat{\mathbf{k}})$, called the Jones vector. This vector is two-dimensional and complex, so that it encodes the electric field components in two orthogonal directions (polarizations). It is a quantity that is specified for each direction (encoded in the direction of $\hat{\mathbf{k}}$ from which the radiation is traveling). We can characterize the response of an antenna element n to radiation from the direction $\hat{\mathbf{k}}$ (the *primary beam*) as an operator $\mathbf{B}_n(\hat{\mathbf{k}})$, a 2×2 complex matrix, acting on $\mathbf{s}(\hat{\mathbf{k}})$ to produced the measured signal. Furthermore, if the antenna element is at position $\mathbf{r}_n$, then there is a time dependent phase factor of $e^{-i[\mathbf{k}\cdot\mathbf{r}_n+\omega t]}$. Thus, the data measured by the receiver n is,

$$\mathbf{d}_n = \int e^{-i[\mathbf{k}\cdot\mathbf{r}_n+\omega t]}\mathbf{B}_n(\hat{\mathbf{k}})\mathbf{s}(\hat{\mathbf{k}})\ d\Omega_k.$$

This relationship is quite general. For a FFTT, we make the assumption that the antenna responses are equivalent, $\mathbf{B}_n = \mathbf{B}$, and that they lie on a plane, $z = 0$, so that $\hat{\mathbf{z}} \cdot \mathbf{r}_n = 0$. We can define

$$\mathbf{q} = \begin{pmatrix} \mathbf{k}\cdot\hat{\mathbf{x}} \\ \mathbf{k}\cdot\hat{\mathbf{y}} \end{pmatrix} \qquad \text{and} \qquad \mathbf{x}_n = \begin{pmatrix} \mathbf{r}_n\cdot\hat{\mathbf{x}} \\ \mathbf{r}_n\cdot\hat{\mathbf{y}} \end{pmatrix},$$

so that, with a suitable change of variables, we can write

$$\mathbf{d}_n = \int e^{-i[\mathbf{q}\cdot\mathbf{x}_n+\omega t]}\frac{\mathbf{B}(\mathbf{q})\mathbf{s}(\mathbf{q})}{k\sqrt{k^2-q^2}}\ d^2q. \tag{1.7}$$

We now define the function

$$\mathbf{s}_\mathbf{B}(\mathbf{q}) \equiv \frac{\mathbf{B}(\mathbf{q})\mathbf{s}(\mathbf{q})}{k\sqrt{k^2-q^2}},$$

so that it is equivalent to $\mathbf{s}(q_x, q_y, -[k^2 - q_x^2 - q_y^2]^{1/2})$ when $q = |\mathbf{q}| < k$ and zero otherwise (analogously for $\mathbf{B}(\mathbf{q})$). This is effectively windowing, beam weighting and zero-padding the sky signal. If indeed the beam pattern $\mathbf{B}(\hat{\mathbf{k}})$ vanishes below the horizon (which can be accomplished with a ground screen), then when we limit our integration over the upper

hemisphere, (1.7) becomes a two dimensional Fourier transform of $\mathbf{s}_B(\mathbf{x}_n)e^{-i\omega t}$. That is,

$$\mathbf{d}_n = \int e^{-i[\mathbf{q}\cdot\mathbf{x}_n+\omega t]}\mathbf{s_B}(\mathbf{q})\ d^2q = \tilde{\mathbf{s}}_\mathbf{B}(\mathbf{x}_n)e^{-i\omega t}.$$

As we did before in developing the van Cittert-Zernike theorem, we assume that the signals we are observing are spatially uncorrelated. Thus, when we correlate the sky signals,

$$\left\langle \mathbf{s}(\hat{\mathbf{k}})\mathbf{s}(\hat{\mathbf{k}}')^\dagger \right\rangle = \delta(\hat{\mathbf{k}},\hat{\mathbf{k}}')\mathbf{S}(\hat{\mathbf{k}}),$$

where $\delta(\hat{\mathbf{k}},\hat{\mathbf{k}}')$ is the spherical δ-function satisfying

$$\int \delta(\hat{\mathbf{k}},\hat{\mathbf{k}}')g(\hat{\mathbf{k}}')\ d\Omega'_k = g(\hat{\mathbf{k}}),$$

for any function g. The quantity $\mathbf{S}(\hat{\mathbf{k}})$ is known as the 2×2 sky intensity Stokes matrix and is the quantity of interest for radio astronomy applications of FFTTs. Therefore, when we compute the mutual coherence function (as seen through the antenna response), we find that

$$\left\langle \mathbf{d}_m\mathbf{d}_n^\dagger \right\rangle = \int e^{-i\mathbf{q}\cdot(\mathbf{x}_m-\mathbf{x}_n)}\frac{\mathbf{B}(\mathbf{q})^\dagger\mathbf{S}(\mathbf{q})\mathbf{B}(\mathbf{q})}{k\sqrt{k^2-q^2}}\ d^2q.$$

This is just the two-dimensional Fourier transform of the quantity

$$\mathbf{S_B}(\mathbf{q}) = \frac{\mathbf{B}(\mathbf{q})^\dagger\mathbf{S}(\mathbf{q})\mathbf{B}(\mathbf{q})}{k\sqrt{k^2-q^2}},$$

so that

$$\left\langle \mathbf{d}_m\mathbf{d}_n^\dagger \right\rangle = \tilde{\mathbf{S}}_\mathbf{B}(\mathbf{x}_m-\mathbf{x}_n). \tag{1.8}$$

Hence, we have again recovered a spatial Fourier relationship between the measured correlations and the sky signal. However, this time the relationship is exact—lacking the flat-sky approximations that were made in the previous sections when developing traditional radio astronomy. We can recover $\mathbf{S}(\mathbf{q})$ simply by inverse Fourier transforming (1.8) and

computing

$$\mathbf{S}(\mathbf{q}) = k\sqrt{k^2 - q^2}\mathbf{B}(\mathbf{q})^{-\dagger}\mathbf{S}_{\mathbf{B}}(\mathbf{q})\mathbf{B}(\mathbf{q})^{-1}.$$

What this analysis has shown us is that computing the sky signal of the whole hemisphere is not significantly more challenging than analyzing small patches. One simply needs enough u-v plane coverage to obtain usable resolution at this scale.

1.4.2 The FFTT Correlator

The hallmark feature of an FFT telescope is its correlator. The elements in an FFTT are arranged in a regular rectangular grid. If we imagine having a one-dimensional grid (evenly spaced points on a line), we can imagine placing the antenna measurements in a vector $\mathbf{f} = (f_1, f_2, \ldots, f_N)$. Since the elements are regularly spaced, their relative positions within the vector correspond directly to their physical separations up to a single scale factor D_0. Because of this, we can compute all of the correlations via a convolution of $\mathbf{f}$ with the reversal of itself

$$(\mathbf{f} \star \mathbf{f}^-)_i = \sum_{j=\max\{1,i+1\}}^{\min\{N,N+i\}} f_j f_{j-i}.$$

Thus $(\mathbf{f}\star\mathbf{f}^-)_m$, defined for $m = -(N-1)$ to $m = N-1$, is the sum of the products of those antenna signals separated by a distance m in $\mathbf{f}$, which is precisely what we want. By the convolution theorem, this can be efficiently computed by multiplying the Fourier transform of $\mathbf{f}$ by its complex conjugate and inverse Fourier transforming. By using an FFT, we can therefore compute the full correlations in $O(N\log(N))$ computations rather than $O(N^2)$.

Conceptually speaking, the reason that we can perform this computation so much more efficiently is that there are fewer data points being computed. A rectangular grid necessarily has many baselines that are redundant, and the FFT correlator aggregates correlations measuring the same baseline into a single data point. Traditional radio telescopes tend to layout the elements to maximize the number of *unique* baselines. Based on our earlier discussions on synthesis, this means that with an FFTT we are sacrificing resolution (by having fewer unique baselines) for cost. However, because of the cost savings, we can have vastly more elements in the array for the same computational cost—which could give us

the collecting area necessary to perform 21-cm tomography. Redundancy in baselines also gives us a novel way to calibrate the instrument that could make FFT Telescopes the most precise instruments of their kind.

1.5 Omniscopes

Omniscopes are fast Fourier transform telescopes that expand the class of baseline distributions that can be used. They accomplish this by weakening the requirement that the elements be located on a regular grid. As we will see, hierarchies of regular grid arrays, even those with arbitrary shearing and rotation, are still amenable to the $O(N \log N)$ FFT correlation algorithm. This concept is introduced in [14], and we shall follow this development to introduce the principles of operation behind the Omniscope.

1.5.1 Principles

We can generalize the development of the FFTT correlator given in Section 1.4.2. Suppose we have m unique lattice basis vectors $\mathbf{a}_i \in \mathbb{R}^2$. In the one-dimensional case that was developed, there was only one such vector $\mathbf{a}_1 = (D_0, 0)$, and as long as the elements were regularly spaced, their positions could be uniquely expressed as $\mathbf{r}_i = i\mathbf{a}_1$. Thus, i determined its location in the vector $\mathbf{f}$. If we generalize this to two-dimensions, we can introduce another vector $\mathbf{a}_2 = (0, D_0)$ and uniquely express the positions of the elements as $\mathbf{r}_{ij} = i\mathbf{a}_1 + j\mathbf{a}_2$, if the elements lie on a regular rectangular grid. If we place the measurements of the antennas labeled as such in a two-dimensional vector $\mathbf{f}$, the baseline between element (i, j) and (k, ℓ), for example, maps directly to the relative positions of the elements within $\mathbf{f}$ via a scale factor. We can therefore use a two dimensional convolution to compute the correlations,

$$(\mathbf{f} \star \mathbf{f}^-)_{ij} = \sum_{\ell=\max\{N,j+1\}}^{\min\{N,N-j\}} \sum_{k=\max\{N,i+1\}}^{\min\{N,N-i\}} = f_{k,\ell} f_{k-i,\ell-j}.$$

Thus, $(\mathbf{f} \star \mathbf{f}^-)_{ij}$ is the sum of the products of elements separated in $\mathbf{f}$ by (i, j)—hence, physically separated by the baseline (iD_0, jD_0).

In general, we do not require the vectors $\mathbf{a}_i$ to be the same magnitude or even linearly independent (indeed, we can't if we want more than two). As long as we can uniquely

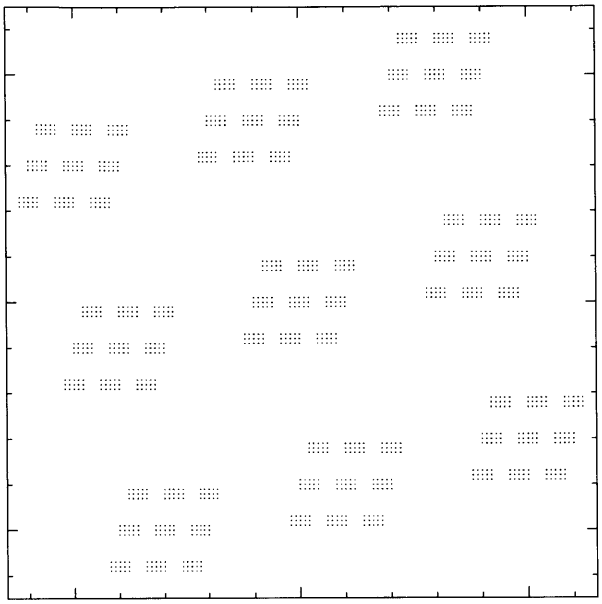


Figure 1-3: An example of a three-dimensional hierarchy. It is composed of 5×3 lattices arranged in a 3×3 lattice. The lattice basis vectors describing this layout are $\mathbf{a}_1 = (1, 0)$, $\mathbf{a}_2 = (8, 0)$, $\mathbf{a}_3 = (40, 10)$, $\mathbf{a}_4 = (0, 1)$, $\mathbf{a}_5 = (2, 8)$ and $\mathbf{a}_6 = (-10, 40)$, with $i_1 = 1, \ldots, 5$ and $i_2, \ldots, i_6 = 1, \ldots, 3$. (Figure from [14]).

express the position of each element in the form

$$\mathbf{r}_{i_1,i_2,\ldots,i_d} = \sum_{j=1}^{d} i_j \mathbf{a}_j, \tag{1.9}$$

where i_j are integers spanning a finite range of values, then we can use a convolution correlator. By the uniqueness of (1.9), each element can be uniquely placed in a d-dimensional vector $\mathbf{f}$ and computing a d-dimensional convolution will recover the correlations. That is, $(\mathbf{f} \star \mathbf{f}^-)_{m_1,m_2,\ldots,m_d}$ represents the sum of the products of the measurements made by elements separated by the baseline

$$\Delta\mathbf{r}_{m_1,m_2,\ldots,m_d} = \sum_{j=1}^{d} m_j \mathbf{a}_j.$$

The class of layouts for which (1.9) is unique for some set of $\mathbf{a}_i$ can be seen to be those which form hierarchical lattices. Figure 1-3 shows an example of such a layout. From the figure, it can easily be seen that the class of layouts which fit this requirement is quite large.

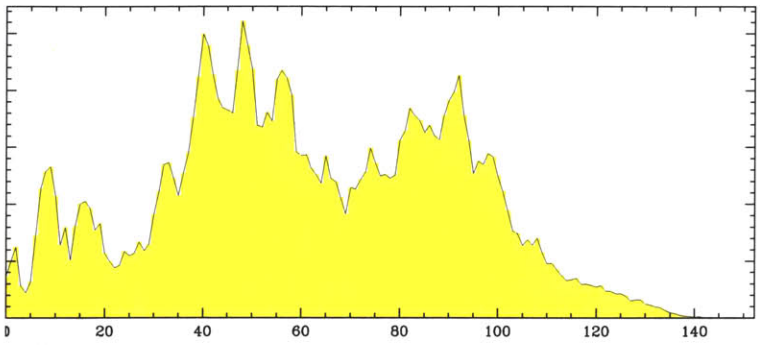


Figure 1-4: Baseline distribution of the antenna layout shown in Figure 1-3. Baselines of the same magnitude are binned together. (Figure from [14]).

1.5.2 Using Earth's Rotation

Despite the rather large class of antenna layouts that can use an $O(N\log(N))$ FFT correlator, we cannot escape the fact that we are making a sacrifice in the number of unique baselines that are present. However, it is possible to use the instrument to construct images that are composed of far more baselines than are present in the static array. Because the Earth is constantly rotating throughout the day, the baselines measured by the instrument at one time are not the same as those measured an hour afterward (they are rotated).

Using this fact, we can cover a significantly larger part of the u-v plane by combining measurements over a 12-hour period. Indeed, one baseline alone will effectively cover a circle within the u-v plane. What matters, then, is the *distribution* of baselines that can be provided by a particular antenna layout that is important. Figure 1-4 shows the baseline distribution of the antenna layout in Figure 1-3. A traditional layout, looking to maximize the number of unique baselines, may be modeled as a randomized distribution with a particular density function. Figure 1-5 compares the distributions attainable in this manner with those obtainable using a hierarchical layout.

1.5.3 Relevance to 21-cm Tomography

From Figures 1-5, we can see that the sacrifice made by reducing the number of unique baselines for a hierarchical layout is not prohibitive when Earth's rotation is used to fill in the u-v plane, and is in fact especially worthwhile when considering the vast computational benefits that can be reaped with such a layout. This makes the Omniscope a worthy candidate for performing 21-cm tomography. There is a surprising amount of flexibility in

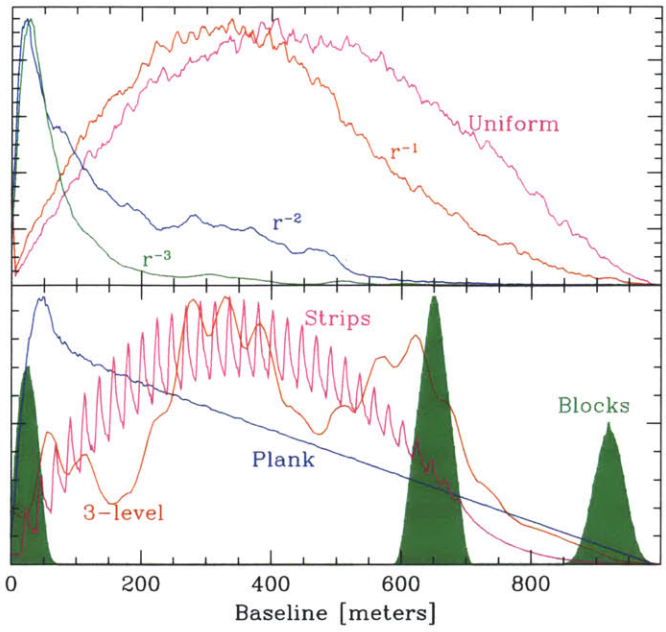


Figure 1-5: Comparison of the baseline distributions of random antenna layouts (top) and hierarchical layouts (bottom). The named layouts in the bottom figure are those described in [14], from which this figure was taken.

the kinds of baseline distributions that can be achieved by an Omniscope layout (e.g. one may want shorter baselines to generate large-scale hydrogen maps and longer baselines for foreground elimination). The $O(N\log(N))$ computational cost of an Omniscope correlator means that the instrument can feasibly be constructed on much more massive scales than other radio telescopes with the same purpose. This is precisely what is needed for 21-cm tomography.

1.6 A Guide to This Thesis

This thesis covers a great deal of material about the Omniscope system, including the theory (as developed in the current chapter), design (developed in Chapter 2) and implementation (developed Chapter 3). In Chapter 2, I provide a high level functional overview of the Omniscope and its components. This includes a description of how the Omniscope will produce the measurements necessary to achieve the science goal outlined in Chapter 1. Chapter 2 also includes a comprehensive treatment of the poly-phase and overlap add filter banks as they are applied to the problem of spectral leakage—a treatment rather difficult to find in current literature (see [1, 12, 10]). Finally, I provide a brief overview of the butterfly

corner-turner, which solves a very important networking problem that will arise when large scale Omniscopes are constructed. For a more complete description of the corner turner, the reader is referred to [7], on which I am a co-author.

In Chapter 3, I detail the implementation of the subsystems outlined in Chapter 2 as developed for a small 2×4 element Omniscope array. Many of these components were developed by other members of the Omniscope team, but are described here for completeness. To begin, the analog chain was designed and built by Eben Kunz, with help from Kris Zarb-Adami along with his group at Oxford and from MIT graduate student Andy Lutomirski. As an addition to this analog chain, I designed and built a Walsh modulation phase switching system for suppressing cross-talk. I provide a mathematical treatment of the technique and establish some interesting results on its application to the problem of cross-talk. Furthermore, I design a physical implementation of the switching system, describe how I built it and provide promising results on its performance. Finally, I detail a set of software tools that I developed for real-time monitoring of the Omniscope and for comprehensive and sophisticated presentation of the data measured by the Omniscope. In Chapter 4, I provide a summary of these results and discuss the future directions of the Omniscope.

Chapter 2

System Overview

In this chapter, we present a detailed overview of the Omniscope system. The system architecture described in the current chapter will be generally scalable to very large Omniscopes. In the next chapter, we focus on the instrumentation of the very first Omniscope prototypes. We begin by introducing the F-X correlator pattern and describing how we can implement it to compute the data necessary to form images, as described in the previous section. We start with a high-level functional schematic of the Omniscope, including the major subsystems and signal paths that connect them. We then describe, in detail, the purpose, composition, and functional behavior of each subsystem so that we have a working specification of its operation.

2.1 The F-X Correlator Pattern

Recall from the previous chapter, that the central purpose of a radio interferometer, is to compute $\langle x(t)y(t)\rangle$, for all pairs of antenna signals $x(t)$ and $y(t)$—knowing that we can reconstruct the sky signal from these values by Fourier inversion methods. By definition,

$$\langle x(t)y^*(t)\rangle = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)y^*(t)\ dt. \tag{2.1}$$

Define $x_T(t)$ and $y_T(t)$ to be $x(t)$ and $y(t)$, respectively, truncated to zero outside of $[-T, T]$ and $\overline{y}_T(t) = y_T(-t)$. Then (2.1) can be rewritten as

$$\langle x(t)y^*(t)\rangle = \lim_{T\to\infty} \frac{1}{2T} \int_{-\infty}^{\infty} x_T(t)\overline{y}_T^*(-t)\ dt. \tag{2.2}$$

The integral is easily recognized as the convolution of $x(t)$ and $y(t)$ evaluated at offset 0. Thus, we define more generally

$$\begin{aligned}\gamma(\tau) = \langle x(t)y^*(t-\tau)\rangle &= \lim_{T\to\infty}\frac{1}{2T}\int_{-\infty}^{\infty} x_T(t)\overline{y}_T^*(\tau-t)\,dt \\ &= \lim_{T\to\infty}\frac{1}{2T}\left[x_T(\tau)\star\overline{y}_T^*(\tau)\right].\end{aligned}$$

Noting that

$$\mathscr{F}\left\{\overline{y}_T^*(t)\right\} = \int_{-\infty}^{\infty} y_T^*(-t)e^{-i\omega t}\,dt = \left[\int_{-\infty}^{\infty} y_T(t)e^{-i\omega t}\,dt\right]^*,$$

we can use the convolution theorem to see that

$$\Gamma(\omega) = \lim_{T\to\infty}\frac{1}{2T}\left[X_T(\omega)\cdot Y_T^*(\omega)\right],$$

where $\Gamma(\omega)$, $X_T(\omega)$ and $Y_T(\omega)$ are the Fourier transforms of $\gamma(\tau)$, $x_T(t)$ and $y_T(t)$, respectively. By Fourier inversion, then

$$\begin{aligned}\langle x(t)y^*(t)\rangle = \gamma(\tau)|_{\tau=0} &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\Gamma(\omega)\,d\omega \\ &= \lim_{T\to\infty}\frac{1}{4\pi T}\int_{-\infty}^{\infty} X_T(\omega)\cdot Y_T^*(\omega)\,d\omega.\end{aligned}$$

If we suppose that the signals are emitting at a narrow band-width $\Delta\nu$, then this reduces to

$$\langle x(t)y^*(t)\rangle = \lim_{T\to\infty}\frac{\Delta\nu}{2T}X_T(2\pi\nu)Y_T^*(2\pi\nu). \tag{2.3}$$

The above discussion shows that we can obtain the complex visibilities by first Fourier transforming the antenna signals, and then conjugate multiplying them. This is the principle behind an F-X correlator. We first transform the signals into the frequency domain and then correlate them in the space domain. The subsystems that perform these two tasks are called the "F-Engine" and the "X-Engine", respectively.

2.1.1 Practical Considerations

To implement (2.3) in a digital setting, we must make some practical considerations. To begin, we cannot in reality compute the expression in (2.3) in the limit as $T \to \infty$, for obvious reasons. Rather, we must compute for a finite time T long enough to adequately approximate the true mutual coherence function. Furthermore, since the Omniscope is a digital instrument, the signals $x(t)$ and $y(t)$ will be discrete signals $x[n]$ and $y[n]$. While the finiteness of the limit T places a limit on the spectral resolution, the discreteness places a limit on the observable band-width via the Nyquist theorem.

The expression in (2.3) assumes that the sources of the signals are narrowly band-limited. In fact, by invoking the principle of superposition and the linearity of the Fourier transform, we can use the same formula to construct maps of the sky at several different frequency bands. Since the multiplication correlates each frequency independently, we can think of $\langle x(t)y^*(t)\rangle$ as being the superposition of complex visibilities at many frequencies. Thus, we can synthesize a map for each sub-band. Section 2.4.1 will shed some light on this matter.

While the finite limit of T places a limit on the spectral resolution obtainable by a practical F-X correlator (though for large T, this limit is insignificant), we can drastically reduce the length of the Fourier transform computation by sacrificing spectral resolution. Specifically, we can think of the data point representing a (coarser) sub-band as an "average" of the frequency components contained within the sub-band. We can do this by breaking the observation time into chunks, each of length N (the length of discrete Fourier transform used to compute $X(e^{i\omega})$ and $Y(e^{i\omega})$), and then summing per-component (vector accumulating) the result. The length of this average serves to better the approximation of (2.3) and increase the signal to noise ratio, but has the added side-effect of reducing the temporal resolution of the maps.

2.1.2 The Omniscope as an F-X Correlator

In this thesis, we will focus on the implementation of the Omniscope as a digital F-X correlator. The analog data from the antenna receiver system is first digitized and then passed into the F-Engine (which performs the temporal Fourier Transforms). Since each frequency is correlated independently, the frequency-domain is subdivided and transferred to the corresponding X-Engine. In this design, the X-Engine subsystem is actually a collection

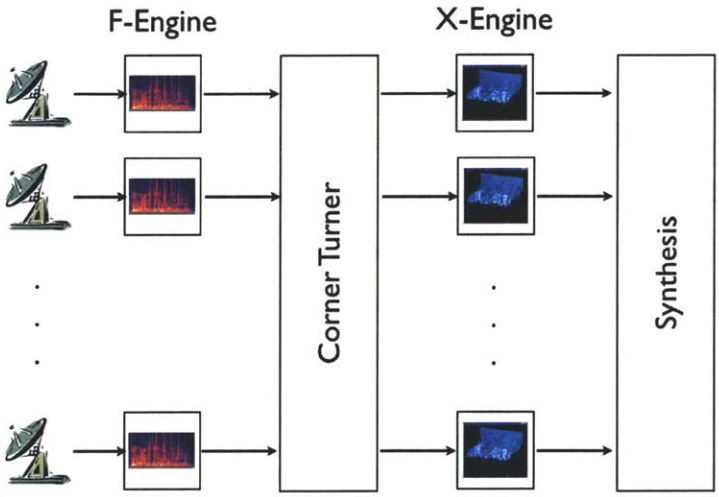


Figure 2-1: Conceptual overview of the Omniscope.

of X-Engines, each performing correlations for a particular frequency or group of frequencies. The correlations are then used for synthesis and map-making.

An important part of the chain was overlooked in the description above, and that is the means of communicating the measured frequency data with the X-Engines. Each X-Engine requires data from all of the antenna channels. For large numbers of antennas, this task can be prohibitive (it is known as the 'corner-turning problem,' since the operation resembles a matrix transpose). However, we will see in section 2.5 that there is an elegant solution to this for the Omniscope. Figure 2-1 shows a conceptual diagram of the Omniscope as an F-X correlator.

2.2 Functional System Schematic

The rest of this chapter is devoted to explaining the function of each of the subsystems in Figure 2-2. In this section, we give a high level summary of the function of each component and its role within the Omniscope. The main systems shown in the figure are the analog chain (which filters and amplifies the measured antenna voltages and then digitizes it), the F-Engine (which applies some signal processing and converts the digitized data to the frequency domain), the calibration system (which continuously calibrates the whole system), the X-Engines (which perform the spatial correlation) and the post-processing step (which synthesizes maps and analyzes the data). Many of the signal paths between these systems

require very high data rates, while others require considerably lower data rates. This is important when considering the implementation platform for each system.

The electric field at each receiver position is measured by the antenna elements. This signal is incredibly weak and therefore must be amplified significantly to become usable. Furthermore, there are many practical considerations that compel us to apply analog filters to the incoming signal. Quadrature demodulation will be applied to allow us to measure a wider range of frequency ranges. The analog chain encompasses the analog amplification, filtering and demodulation, as well as the analog to digital converter. Once through the ADC, the data moves on to the digital components.

The F-Engine is the first recipient of the digitized data. Each F-Engine is associated with an antenna or group of antennas and computes the discrete Fourier transform of the received signal as described in 2.1.1. As we will see in Section 2.4.1, the F-Engine uses efficient signal processing techniques to perform this spectral analysis while combating undesired effects such as spectral leakage.

An instrument as sensitive as the Omniscope requires careful calibration. Fortunately, certain features of the Omniscope's design (namely, the presence of redundant baselines), enable the use of powerful techniques for precisely calibrating the instrument in real-time. The calibration system is responsible for this task. In [6], MIT graduate student and current Omniscope team member Adrian Liu provides a thorough development of the methods by which one can calibrate an instrument like the Omniscope using its redundant baselines. Because the calibration algorithm relies on the separate measurements of the redundant baselines, we cannot employ the FFTT algorithm for supplying these correlations. Fortunately, calibration does not need to happen at the same rate as the imaging computation. Thus, we can afford to build a much lower data-rate full-correlator for the calibration system.

The X-Engines are the heart of the Omniscope. The corner-turner is responsible for transporting data from the F-Engines to the X-Engines. Each X-Engine operates on a particular frequency or group of frequencies, to produce visibilities that can be used to form maps at the associated frequency. So far, all of the systems (except the calibration system) are processing data at the full sampling rate of the ADC. Data leaving the X-Engine, however, has been accumulated for a relatively long time. As a result, the data rate is significantly lower when it reaches the post-processing stage. Within the post-processing

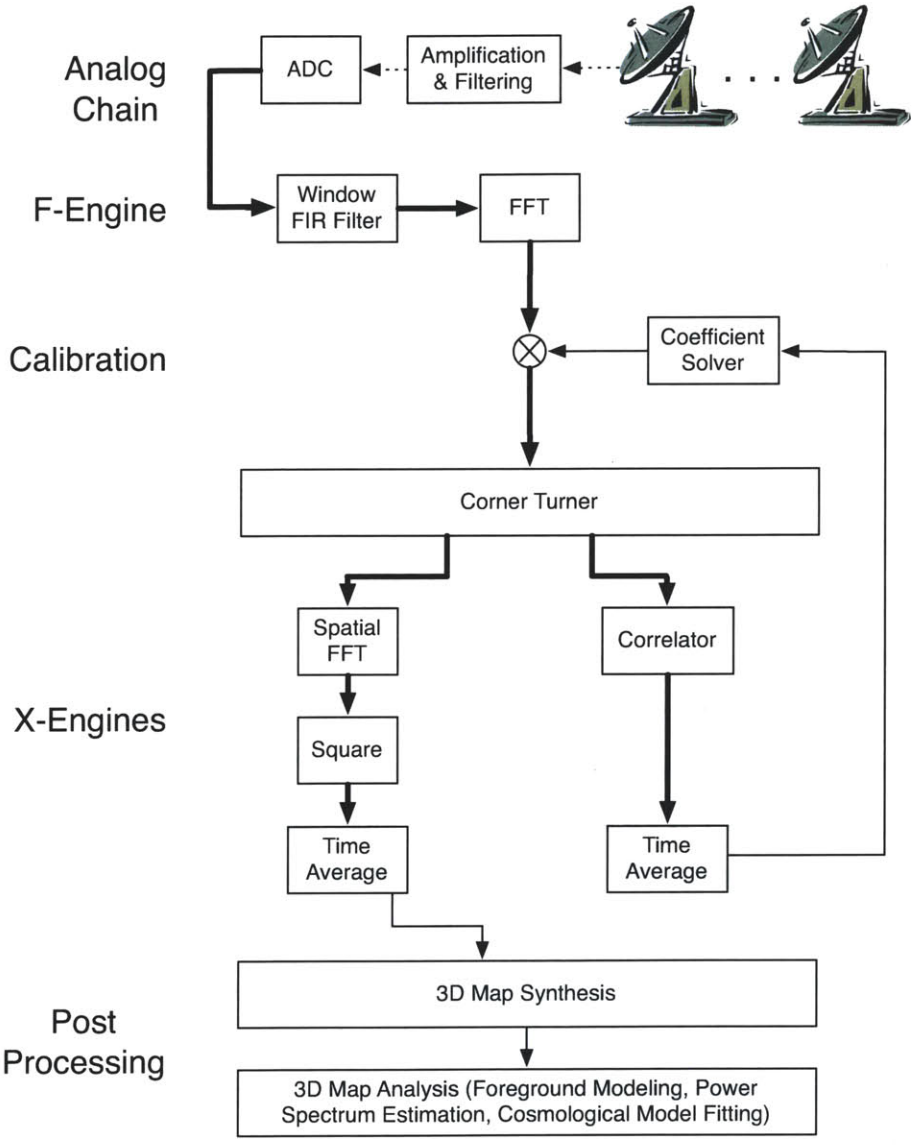


Figure 2-2: Diagram of the signal path for a single channel of the Omniscope. Thick lines indicate high-data rate digital signal paths (needing real-time computation) while thinner lines correspond to digital signal paths of much lower data rates (reduced by a factor of up to $\sim 10^9$). The dotted lines indicate analog signal paths.

stage, most of the relevant science is performed. This includes map synthesis, modeling of foregrounds, cosmological model fitting and other forms of scientific data analysis.

2.3 Analog Chain

The first element in the analog chain is the antenna. The antenna used in the Omniscope is undirected and measures the electric field components along two orthogonal directions (polarizations). Since these polarizations are measured and correlated completely independently, we will focus on the composition of an analog signal channel (of which each antenna has two). The signal measured by the antenna is quite weak and must be amplified. Because the signal levels are so low, it is important that the first amplifier be a low-noise amplifier (LNA) and that it be applied as early as possible. Furthermore, additional filters (including low-pass and notch filters) are desired at this stage to ensure that unwanted signals do not saturate the amplifiers. A line driver is then used to reliably transmit the signal between systems.

The phenomenon of aliasing that occurs when discretely sampling a signal generally limits the range of observable frequencies. However, with clever signal processing techniques, this turns out to only be a limitation on the observable *bandwidth*. Given a signal proportional to $e^{i\omega t}$, we may produce a lower frequency signal with the same information by multiplying by $e^{-i\omega_0 t}$ (producing a signal proportional to $e^{i(\omega-\omega_0)t}$). If one is interested in a band at a frequency higher than the Nyquist rate of the ADC, but with a lower width, it can be obtained by shifting the center of the band to 0 frequency and then sampling. Careful use of bandpass filters ensures that only the desired band is measured. The complex multiplication required to shift the received signal can be performed with quadrature demodulation, as discussed in the next subsection. The very last step in the analog chain is the analog to digital converter. Figure 2-3 summarizes the components in the analog conditioning system between the antennas and the ADC.

2.3.1 Quadrature Demodulation

Quadrature demodulation is a means for realizing the complex multiplication by $e^{i\omega_0 t}$ needed to translate signals in the frequency domain. The principles of its operation are as follows. Suppose a signal $x(t)$ has been band-pass filtered so that it occupies a band $\Delta\omega$ with center

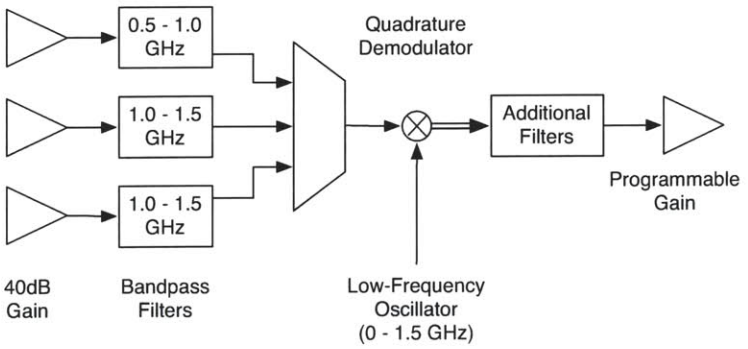


Figure 2-3: Analog signal conditioning sub-system.

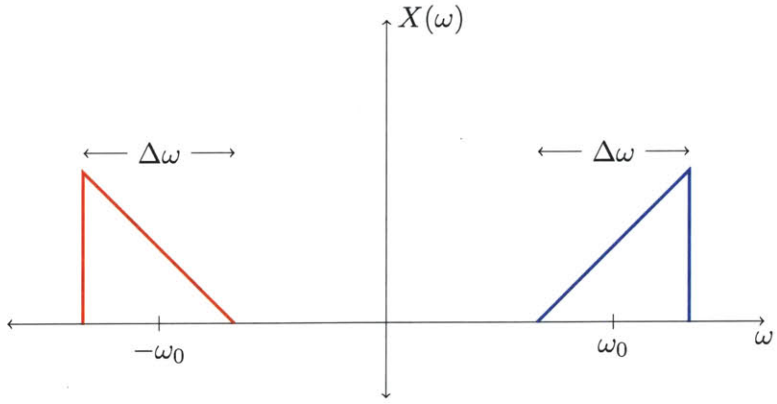
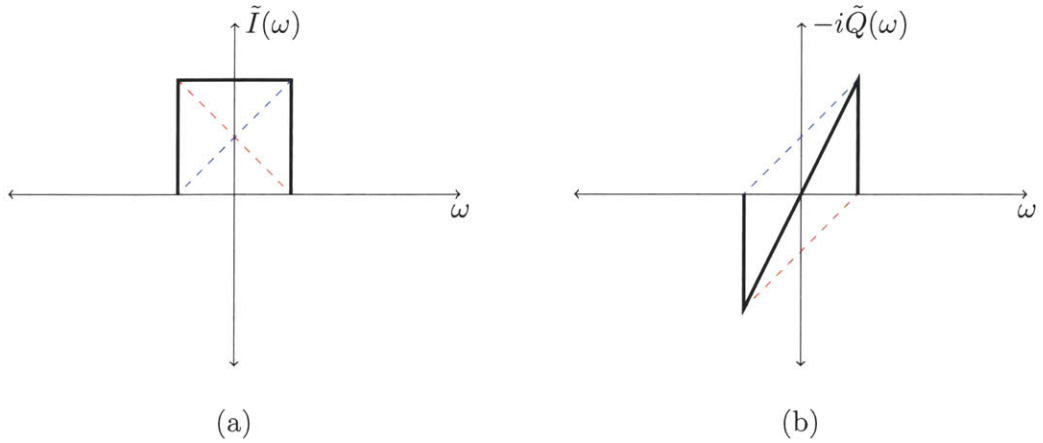


Figure 2-4: Spectrum of $x(t)$.



(a) (b)

Figure 2-5: (a) Spectrum of $I(t)$ and (b) spectrum of $-iQ(t)$.

frequency ω_0 (see Figure 2-4). Define

$$I(t) = \text{LPF}_{\Delta\omega}(\cos(\omega_0 t)x(t))$$
$$Q(t) = \text{LPF}_{\Delta\omega}(\sin(\omega_0 t)x(t)).$$

By the convolution theorem, we can construct the spectra of $I(t)$ and $Q(t)$ by convolving $X(\omega)$ with $\frac{1}{2}\left[\delta(\omega+\omega_0)+\delta(\omega-\omega_0)\right]$ and $\frac{1}{2i}\left[\delta(\omega+\omega_0)-\delta(\omega-\omega_0)\right]$, respectively, in the frequency domain. $\tilde{I}(\omega)$ is thus sum of the upper and lower copies of the frequency bands in $X(\omega)$, as shown in Figure 2-5(a). Similarly, $\tilde{Q}(\omega)$ is $-i$ times the *difference* between the upper and lower copies of the frequency bands in $x(t)$, as shown in Figure 2-5(b). By the linearity of the Fourier transform, it follows that we can reproduce the upper sideband spectrum $X_+(t)$ and the lower sideband spectrum $X_-(t)$ with

$$X_+(\omega) = \tilde{I}(\omega) + i\tilde{Q}(\omega)$$
$$X_-(\omega) = \tilde{I}(\omega) - i\tilde{Q}(\omega).$$

Since $X_+(\omega) = X_-^*(-\omega)$ if $x(t)$ is real, this simply means that we can make either sideband appear in the interval $[0,\infty)$ (i.e. we can measure above and below the center frequency ω_0). This becomes important when we discuss the biplex FFT that is used to compute the discrete Fourier transform, since it discards the negative frequencies in computing the Fourier transform.

2.4 F-Engine

The F-Engine is one of the two core digital components. It is responsible for taking the digitized antenna data and computing the Fourier transform of the signal. The structure of choice for performing this is not a simple windowed DFT, but rather, a more sophisticated structure known as a polyphase filter bank. Once the data has been transformed into frequency domain, the F-Engine divides the spectrum into individual or groups of frequencies, to be transmitted to each X-Engine unit.

2.4.1 Polyphase and Overlap-Add Filter Banks

Spectral analysis involves determining the frequency content of a signal. Ordinary windowed DFTs are useful tools for performing spectral analysis on a sampled signal. However, the phenomenon of spectral leakage, in which one frequency generates spectral components in neighboring frequency bins, makes them unattractive for many applications. Filter banks provide a method of performing spectral analysis that significantly reduces spectral leakage by design.

We begin by defining Short Time Fourier Transform (STFT) and then describe the phenomenon of spectral leakage that follows from straightforward use of it. We then introduce the filter bank as a solution to the problem of spectral leakage and describe implementations of the filter bank using DFTs. We develop two alternate representations for efficient realizations of the DFT filter bank

2.4.2 Discrete Fourier Analysis and Spectral Leakage

The Discrete Fourier Transform

Ideal spectrum analysis for a sampled (discrete) signal $x(n)$ can be performed by the Discrete Time Fourier Transform (DTFT) [9],

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}. \tag{2.4}$$

However, for signals that have infinite support (are non-zero for an infinite number of points), (2.4) cannot be physically realized. To approximate (2.4), we give up perfect frequency resolution for a finite length calculation. Instead, we use the N-point Discrete

Fourier Transform (DFT),

$$\tilde{X}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-jk(2\pi/N)n} \tag{2.5}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} x(n)W_N^{-kn}, \tag{2.6}$$

where we have defined $W_N = e^{j(2\pi/N)}$ for convenience. If $x(n) = 0$ for $n \notin [0, N-1]$, then (2.5) is just (2.4) sampled at N equally spaced intervals in frequency space and scaled by $1/N$. We can see this by plugging $\omega = 2\pi k/N$ into (2.4).

In general, if $x(n)$ is nonzero for $n < 0$ or $n \geq N$, then (2.5) is equivalent to forcing $x(n)$ to be zero outside of $[0, N-1]$ and sampling the DTFT of the result. We can think of this as applying a "window" to $x(n)$ and Fourier transforming the result. As it stands, (2.5) is equivalent applying a rectangular window to $x(n)$. That is,

$$\tilde{X}(k) = \frac{1}{N}\sum_{n=0}^{N-1} w(n)x(n)W_N^{-kn}, \tag{2.7}$$

where $w(n)$ is given by,

$$w(n) = \begin{cases} 1 & \text{if } 0 \leq n < N \\ 0 & \text{otherwise.} \end{cases}$$

The function $w(n)$ is called a *window function.* Any function that is zero outside of the range $[0, N-1]$ is a valid window function and we will see later why we might want to choose a non-rectangular window function.

In consideration of the above discussion, we can summarize the meaning of $\tilde{X}(k)$ as.

$$\tilde{X}(k) = \textsc{Sample}_N(\text{DTFT}(x(n)w(n))). \tag{2.8}$$

That is, we window $x(n)$ by $w(n)$, compute the DTFT, and then take N equally spaced samples from $[0, 2\pi)$ in the frequency domain.

Spectral Leakage

There is an alternate way of viewing the action of a DFT than that given in (2.8). Suppose we force $x(n)$ to be periodic with period N by defining

$$\tilde{x}(n) = x(n \bmod N).$$

That is, we copy the first N values repeatedly to the left and right of the origin. The DFT can be thought of as the discrete time Fourier coefficients of $\tilde{x}(n)$. Consider, for the moment, a sinusoidal $x(n)$. That is,

$$x(n) = e^{i\omega n}. \tag{2.9}$$

If $\omega = k(2\pi/N)$ for some integer k, then $\tilde{x}(n) = x(n)$ (the sinusiod will "fit" perfectly inside the window) and we will reproduce a spike in frequency bin k as expected. However, if this is not the case, then $\tilde{x}(n) \neq x(n)$ and we will clearly have several other Fourier harmonics in the DFT spectrum due to the discontinuity at the period boundaries. That is, the spectrum that would otherwise be a spike in a DTFT with infinite spectral resolution will "leak" into the other frequency bins as we instead perform a DFT.

The nature of this spectral leakage depends of the window function used in the DTFT. By the convolution theorem, we can see that the DTFT of $x(n)w(n)$ is the convolution of the DTFT of $x(n)$ and the DTFT of the window function $W(\omega)$. Thus, if $x(n)$ is given by (2.9) then the DFT is a sampled version of the DTFT of $w(n)$ offset by ω.

In Figure 2-6, we show the zero-padded 16-point DFT of a sinusiod in the 8th frequency bin using a rectangular window (see Appendix A for information on zero-padding). Because the signal falls directly in frequency bin 8, the zeros of $W(\omega)$ precisely coincide with the other frequency bins. However, if the frequency of the signal lies between bins, the other bins will sample the nonzero side-lobes of $W(\omega)$, leading to the spectrum in Figure 2-7. The spread in the sampled signal is spectral leakage.

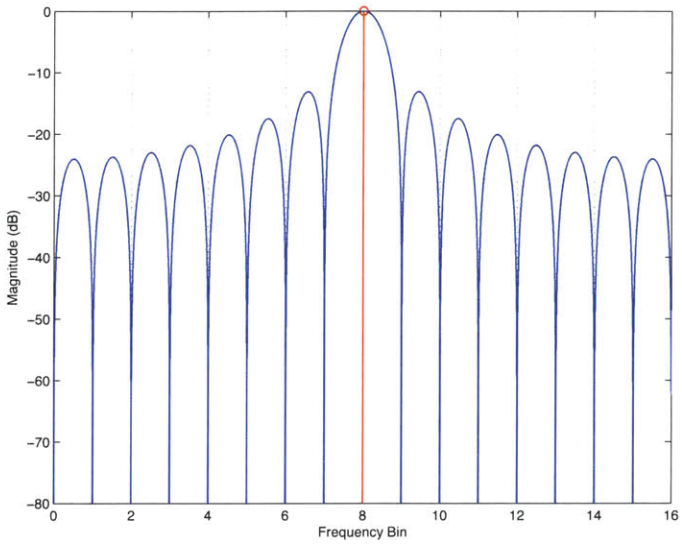


Figure 2-6: DFT of a sinusiod in bin 8 through a rectangular window (red). Note that the zeros of zero-padded spectrum (blue) align with bins of DFT.

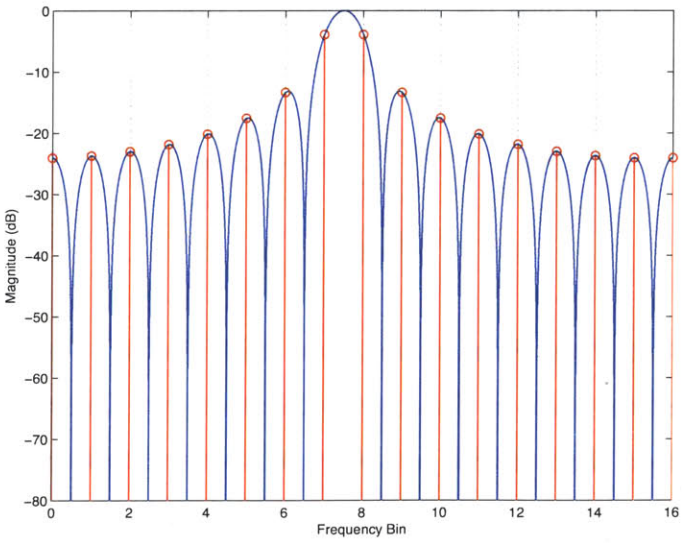


Figure 2-7: DFT of a sinusiod in bin 7.5 through a rectangular window (red). Note that the bins of the DFT now align with the peaks of the lobes in the zero-padded spectrum (blue).

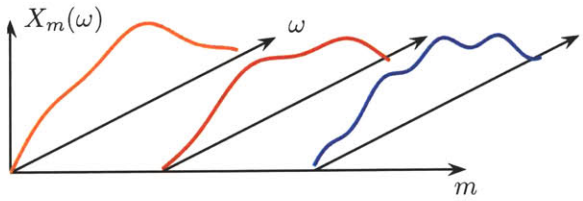


Figure 2-8: Conceptual view of the short time Fourier transform in three dimensions (transform versus time and frequency). Each $X_m(\omega)$ is a windowed Fourier transform of the signal at frame m.

2.4.3 Short Time Fourier Transform (STFT) Analysis

The Short Time Fourier Transform

Often, we may want to analyze more than just the first N points of a signal $x(n)$. We can imagine "sliding" our reference frame along $x(n)$ by a hop size R so that we window $x(n)$ at regularly spaced locations. That is, we compute

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} w(mR - n)x(n)e^{-j\omega n}$$

This is a time varying function that tells us the DFT of $x(n)$ at the mth hop. For real-time spectral analysis, we keep the window fixed in time and shift $x(n)$ in time. By an appropriate change of variables, we can write

$$X_m(\omega) = e^{-j\omega mR} \sum_{n=-\infty}^{\infty} w(-n)x(n + mR)e^{-j\omega n} \tag{2.10}$$

$$= e^{-j\omega mR}\tilde{X}_m(\omega). \tag{2.11}$$

$X_m(\omega)$ is the *Short Time Fourier Transform* (STFT) of $x(n)$ and $\tilde{X}_m(\omega)$ is the STFT with respect to the fixed frame of the observer (as in the real time case). Figure 2-8 shows a conceptual depiction of the STFT as a series of discrete Fourier spectra at regular time samples.

2.4.4 Discrete Filter Banks

A filter bank provides a technique to overcome the problem of spectral leakage that occurs when performing ordinary discrete time Fourier analysis. It works by dividing the frequency

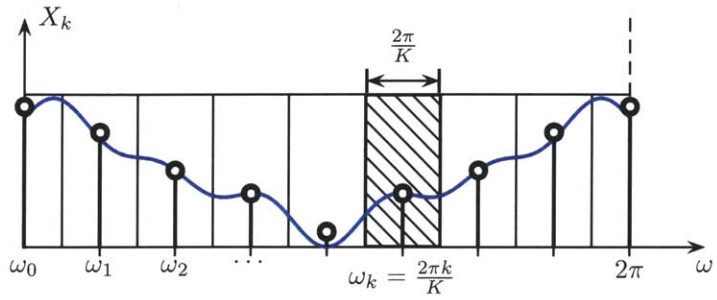


Figure 2-9: Frequency bin division for a discrete filter bank. Bin k is filled. The solid curve represents the actual spectrum of $x(n)$, whereas the stem plot shows the approximate values of X_k within each bin (i.e. the collective frequency content).

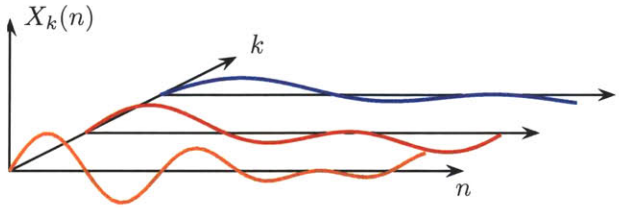


Figure 2-10: Conceptual view of a discrete filter bank. Each $X_k(n)$ is a time varying signal representing the frequency content of bin k.

domain into a series of K frequency bins (analogous to the discrete samples we get from a DFT). The value in bin k represents the collective contribution of all of the frequencies within a band of width $2\pi/K$ centered at the frequency $2\pi k/K$ (see Figure 2-9).

Filter banks are a common tool in the audio processing profession [12], in which each of the bins represents an equalizer channel. The signal coming from each bin, $X_k(n)$, is a time varying signal that reflects only desired frequency components. It can be thought of as the result of filtering the signal with a bandpass filter centered at the bin frequency. A conceptual representation of this process is shown in Figure 2-10. We now present several useful techniques for implementing the Discrete Filter Bank adapted from [2].

Implementation

Let $h(n)$ be the impulse response of a low-pass filter with width $\frac{2\pi}{K}$. We can obtain the signal $X_k(n)$ by first shifting $x(n)$ so that the frequency band at ω_k is centered at 0 and then filtering with $h(n)$. We do this through complex modulation by W_K^{-kn}. Thus,

$$X_k(n) = h(n) \star [x(n)W_K^{-kn}]. \tag{2.12}$$

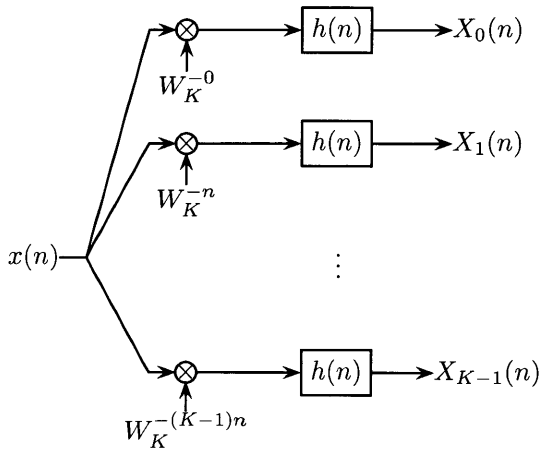


Figure 2-11: Diagram of a discrete filter bank system.

A diagram of this system is shown in Figure 2-11. In each channel of the filter bank, the signal $x(n)$ is first mixed with W_K^{-kn} to center the desired band of the signal over the low-pass filter, and then it is filtered by $h(n)$. The rest of our discussion focuses on optimizing this structure.

There is an alternative but useful way of viewing a filter bank in terms of band pass filters. From (2.12), we have

$$\begin{aligned} X_k(n) &= \sum_{r=-\infty}^{\infty} h(r)x(n-r)W_K^{-k(n-r)} \\ &= W_K^{-kn} \sum_{r=-\infty}^{\infty} h(r)W_K^{kr}x(n-r) \\ &= W_k^{-kn}\left[h_k(n) \star x(n)\right], \end{aligned} \tag{2.13}$$

where,

$$h_k(n) = h(n)W_K^{kr}, \tag{2.14}$$

is $h(n)$ shifted by ω_k in the frequency domain (i.e. it is a bandpass filter centered at ω_k with the same shape as $h(n)$). What (2.13) tells us is that we can get $X_k(n)$ by bandpass filtering $x(n)$ at the kth bin and shifting it to the origin in frequency. Figure 2-11 becomes Figure 2-12 under this perspective.

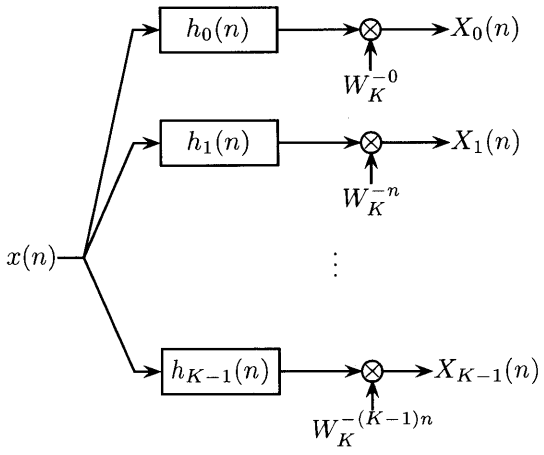


Figure 2-12: Bandpass filter view of a discrete filter bank system.

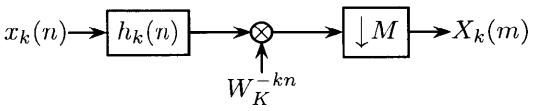


Figure 2-13: Decimating one channel of the filter bank.

The Polyphase Filter Bank

One thing to note about the system shown in Figure 2-12 is that the $X_k(n)$ are oversampled. The bandwidths of the $X_k(n)$ are a factor of K smaller than that of $x(n)$, but are sampled at the same rate. It makes sense to decimate each channel, as shown in Figure 2-13. Unfortunately, we are still filtering in the undecimated time n. We could save computational bandwidth by commuting the decimation with the filtering. Fortunately, we can do this by using a *polyphase* structure to perform the decimated filtering with $h_k(n)$. This structure is described in detail in Appendix B.

For a polyphase structure, (B.2) would transform (2.13) into

$$X_k(m) = W_K^{-kmM} \sum_{\rho=0}^{M-1} \overline{p}_{k,\rho}(m) \star x_\rho(m), \tag{2.15}$$

where

$$\overline{p}_{k,\rho}(m) = h_k(mM + \rho), \qquad x_\rho(m) = x(mM - \rho),$$

are the polyphase filters for $h(n)$ and the polyphase components of $x(n)$, respectively. If we

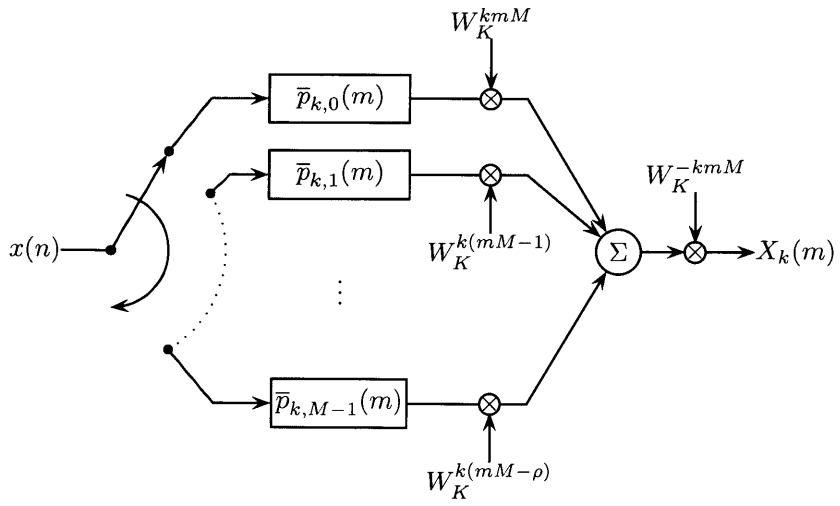


Figure 2-14: Decimating one channel of the filter bank using a polyphase decimator.

expand (2.15) using (2.14), we get

$$X_k(m) = W_K^{-kmM} \sum_{\rho=0}^{M-1} \left[\overline{p}_\rho(m) W_K^{k(mM-\rho)}\right] \star x_\rho(m), \tag{2.16}$$

Thus, using a polyphase decimator turns Figure 2-13 into Figure 2-14.

If we critically sample each channel of the filter bank so that $M = K$, then (2.16) reduces to

$$\begin{aligned} X_k(m) &= \sum_{\rho=0}^{K-1} \left[\overline{p}_\rho(m) W_K^{-k\rho}\right] \star x_\rho(m) \\ &= \sum_{\rho=0}^{K-1} \left[\overline{p}_\rho(m) \star x_\rho(m)\right] W_K^{-k\rho}. \end{aligned}$$

This is precisely in the form of (2.6) for the discrete Fourier transform. Thus, it is clear that we can efficiently compute the $X_k(m)$ using a Fast Fourier Transform (FFT), so that Figure 2-11 becomes Figure 2-15.

Typically, the $h(n)$ has finite length L, and if it as an integer multiple N_p of K, then each of the polyphase filters $\overline{p}_\rho(m)$ have length N_p—referred to as the number of *taps* in the polyphase filter. A network diagram of a polyphase filter with four taps is shown in Figure 2-16.

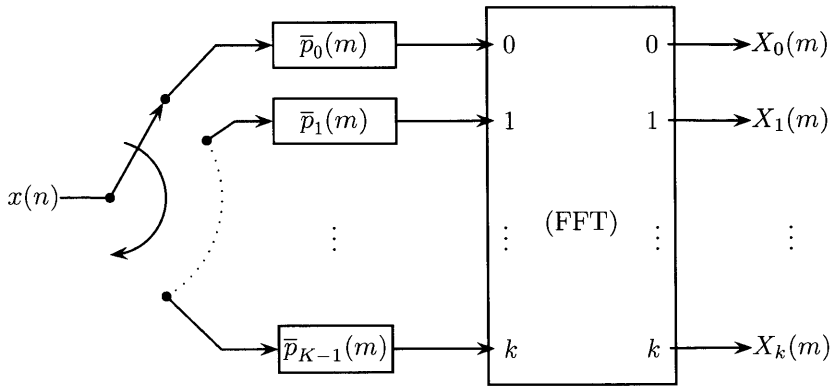


Figure 2-15: Efficient realization for a critically sampled polyphase filter bank using an FFT.

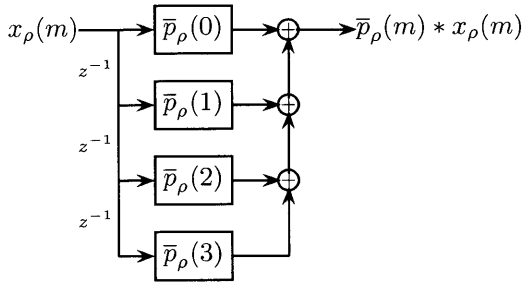


Figure 2-16: Digital network diagram of a polyphase filter with four taps.

Weighted Overlap-Add Filter Bank

Another implementation of a discrete filter bank (also adapted from [2]) can be reached if we begin by rewriting (2.13) as

$$\begin{aligned} X_k(m) &= \sum_{n=-\infty}^{\infty} h(mM-n)x(n)W_K^{-kn} \\ &= W_K^{-kmM} \sum_{n=-\infty}^{\infty} h(-n)x(n+mM)W_k^{-kn} \\ &= W_K^{-kmM} \tilde{X}_k(m). \end{aligned}$$

Comparing this to (2.10), we see that this is just like a discretized STFT of $x(n)$. Using a trick similar to that used to derive the polyphase filters in Appendix B, we can break summand in $\tilde{X}_k(m)$ up into blocks of K samples by changing variables to $n = lK + r$

$$\begin{aligned} \tilde{X}_k(m) &= \sum_{n=-\infty}^{\infty} h(-n)x(n+mM)W_K^{-kn} \\ &= \sum_{r=0}^{K} \sum_{l=-\infty}^{\infty} h(-r-lK)x(r+mM+lk)W_K^{-kr} \\ &= \sum_{r=0}^{K} \tilde{x}_m(r)W_K^{-kr}. \end{aligned}$$

This is just the DFT of the signal $\tilde{x}_m(r)$. The signal $\tilde{x}_m(r)$ is obtained by sliding the window $h(n)$ in time in hops of M (analogous to R in the Section 2.4.3). Then at each step, we multiply the translated $x(n)$ by $h(n)$, break it up into blocks of K samples, overlap them, and add them. The last step is carried out by the sum over l. Viewing $h(n)$ as a weighting function, we can see why this is called a *weighted overlap-add filter bank*. Figure 2-17 represents this process graphically where the filter $h(n)$ is finite and of length $4K$ (i.e. it has four taps).

The weighting function $h(n)$ for applied filter banks is usually a sinc function (corresponding to the rectangular filter shape) with nodes at the overlapping window boundaries (because we are critically sampling the channels). Since the sinc must be finite (because we can only have a finite number of taps), we also often apply a windowing function to this to taper the side-lobes beyond the cutoff frequency. The hamming window is a common choice since it minimizes the magnitude of the nearest (largest) side lobe [11].

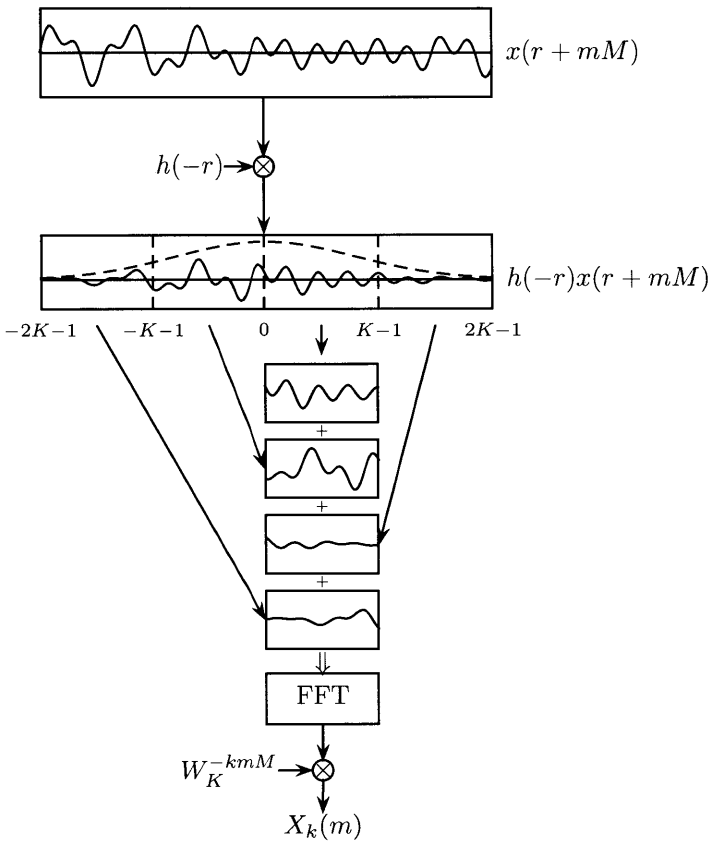


Figure 2-17: Graphic representation of a weighted overlap-add filter bank with four taps.

Comparison of Polpyhase and Overlap-Add Filter Banks

There is a noteworthy relationship between the polyphase implementation of a discrete filter bank and the overlap-add implementation. If we imagine expanding each one of the polyphase filters in Figure 2-15 using the digital network model in Figure 2-16 (letting the taps lie on the axis coming out of the page), there is a subtle relationship to the overlap-add filter bank in Figure 2-17.

If we consider the special case where the filter bank channels are critically sampled (i.e. $M = K$), then each of the polyphase filters in Figure 2-15 corresponds to a single r value in the overlapadd windows in Figure 2-17. The four windows that are accumulated then correspond to the four taps in each polyphase filter. Thus, if $M = K$ and the width of $h(n)$ is N_pK, then the polyphase filter bank and overlap-add filter bank are conceptually and functionally equivalent.

In general, the overlap-add filter bank can handle general sampling rates M of the filter bank channels, whereas the polyphase filter bank is limited to the $M = K$ case—although it can easily be adapted to the case where M is an integer multiple of K by upsampling before each polyphase filter.

Engineering The Spectral Leakage

The polyphase filter bank is well-suited for the task of suppressing spectral leakage because it eliminates it by design. Indeed, we have *defined* each bin to be the representative value of the frequencies within a certain band and no others. The shape of the filter $h(n)$ specifies how well we can faithfully uphold this definition. Indeed, a polyphase filter bank with only one tap could not possibly be more effective than a windowed DFT. This is because the finite approximation of $h(n)$ is highly inadequate with only one tap. By increasing the number of taps, we can increase the quality of our spectral leakage suppression. However, taps cost hardware resources. Thus, the polyphase filter bank gives us a way of engineering away spectral leakage by tuning the balance between leakage and hardware resources.

2.5 Corner Turner

The corner turner solves an challenging and interesting engineering problem. We must be able to have each array element (i.e. each F-Engine) communicate data to each one of the

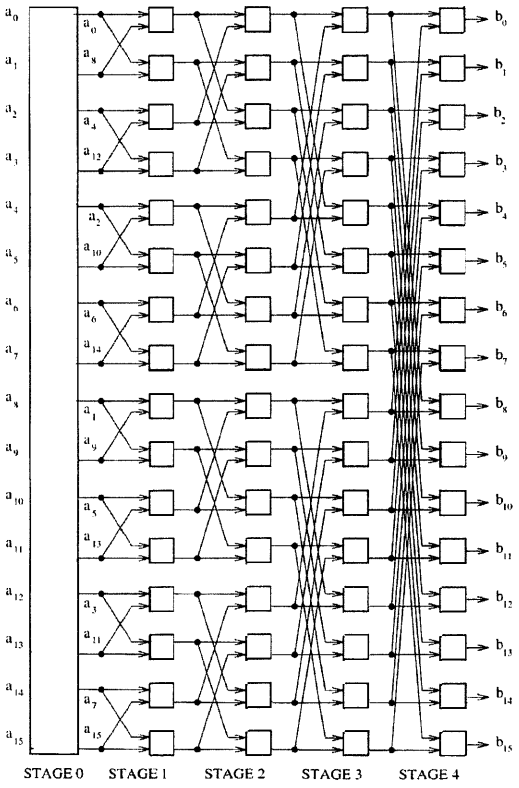


Figure 2-18: An example of a butterfly network.

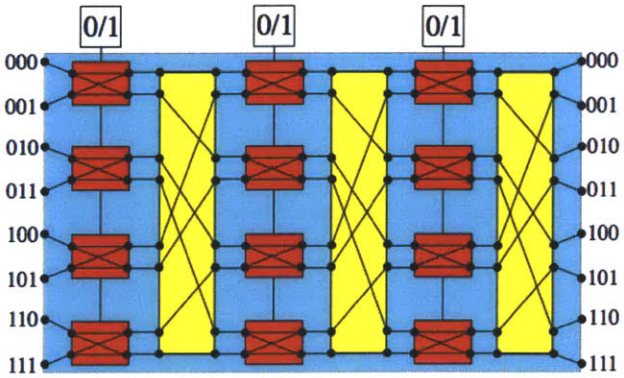


Figure 2-19: A butterfly corner-turner for 8 nodes. The small boxes are the 2×2 switches while the larger ones are the perfect shufflers. The control signals for each layer are shown on the top. (Figure from [7]).

X-Engine elements (each of which performs the correlation step for a particular frequency band). The naïve way to do this would be to either build a complete network, $O(N^2)$, or use several large (expensive) general purpose switches. Neither of these solutions are scalable to Omniscope sized arrays.

Both of the naïve solutions described above are not scalable because they do far more than we need. In a complete graph, each edge will be utilized at $1/N$ times the communication bandwidth. A general switch is unnecessary because the communication pattern is deterministic and synchronous. This allows us to use more efficient communication networks like the butterfly network.

An example of a butterfly network is shown in Figure 2-18. The purpose of the first stage in the butterfly network is to control communication across each half of the network. The purpose of the second stage is to control communication within these halves, etc. Thus, we can see that this network allows us to have each array element communicate with each X-Engine system. The main advantage of this network is that its cost also scales as $O(N \log N)$.

We can efficiently build a butterfly corner turner using simple identical modules (see [7]). We can imagine having many simple switches that have two inputs and two outputs and a single bit control signal. When the control signal is 0, input 1 routes to output 1 and input 2 routes to output 2. When the control signal is 1, the outputs are swapped. We lay N of these switches in a row for each of the $\log N$ layers (stages). Each layer shares a single control bit. We can create a network with the same overall functionality as that

Figure 2-18 by connecting the layers with a series of identical "perfect shufflers." The end result is shown in Figure 2-19.

To complete a corner turning operation, we must cycle through all of combinations of the control signals (collectively represented by a $\log N$-bit number i). The source node number and destination node number are related to i by a determined permutation (similar to an xor). Thus, as we run i through all possible values, every source node will have at some point had a direct connection with each of the destination nodes.

The butterfly network corner-turner gives us precisely the desired functionality needed for the Omniscope, without paying for the extra functionality in a general purpose switch. We can imagine combining several 2×2 switches onto a single device to create a larger module and optimize for cost.

2.6 X-Engine

The X-Engine is the second of the core digital components. As described in the introduction, the X-Engine system is actually composed of several X-Engine units. Each X-Engine computes the correlations of the instrument at a particular frequency, or group of frequencies. The first step in an X-engine is to perform a small transpose to the incoming data. The nature of this transpose depends on whether a full correlator or an FFT correlator is being used. In the full implementation of an Omniscope, the correlation will be a simple FFT over the antenna channels. However, in early prototypes of the Omniscope, it is practical to use a full correlator. This is because the array size is small enough that the cost is not prohibitive. Also, it allows us to more easily debug the Omniscope as its core components are implemented. Furthermore, it obviates the need for a second calibration correlator.

Chapter 3

Instrumentation

3.1 Introduction

The instrumentation details of the Omniscope are important, as it is the advances in semiconductor technology since early traditional radio telescopes that enable the FFTT correlator pattern to be used in interferometers, especially at the scale of the Omniscope. Specifically, programmable logic is able to operate at clock-speeds high enough to perform pipelined digital signal processing at megahertz bandwidths. In this section we justify the use of programmable hardware to implement the core digital signal processing tasks in the Omniscope and introduce the CASPER hardware that enables quick prototyping of programmable logic-based radio astronomy designs. The remainder of the chapter is involved with the detailing the implementation of the major Omniscope subsystems.

3.1.1 Field Programmable Gate Arrays

The core computational burden of the Omniscope lies with the F-Engine and X-Engine. To get an idea of the computational bandwidth required, we describe a moderately sized Omniscope with $N \sim 10^4$. We will assume a bandwidth $\Delta\nu$ of 25 MHz (requiring a 50 MHz ADC), an 8-bit ADC and a $N_{FFT} = 1024$ point FFT in the F-Engine. The amount of raw time domain data flowing into the system is

$$2\Delta\nu N = (5 \times 10^7\,\text{MHz}) \cdot (10^4) = 500\,\text{GB/s}.$$

Over the course of a typical 12-hour measurement, this leads to over 2 petabytes of raw data. A record and post process solution is both taxing in storage and computation and also lacks the real time computational data required by the calibration system. Fortunately, the data we are actually interested in is the accumulated correlations. Thus, a dedicated DSP system can significantly reduce the amount of data to a manageable size—the reason that the Omniscope is an F-X correlator.

The computational bandwidth required for real-time performance of the F-Engine is quite significant as well. Implementation of a polyphase filter bank requires on the order of 10^4 operations per frame for 1024 points. There are approximately $\Delta\nu/N_{FFT} = 5 \times 10^5$ frames per second. Thus, each channel requires at on the order of 5×10^9 operations per second. This is quite taxing for a sequential processor, especially since we haven't factored in memory and network overhead.

Fortunately, the F-Engine computations are easily pipelined and operate at continuous data flow. It is precisely these kinds of calculations that are ideally suited for field programmable gate arrays (FPGAs). When the computation is properly pipelined, nearly every clock cycle of the FPGA produces an output data point. Thus, even though the clock-speed of a complex design is typically much lower than a sequential processor, it is much more efficient in the number of data points computed per clock-cycle. Indeed, the computation described is taxing for even the fastest sequential processors for just a single channel, but we have successfully implemented precisely such a design in a moderate FPGA to process 32 channels in parallel.

3.1.2 CASPER Hardware

It is worthwhile to introduce the work of the Center for Astronomy Signal Processing and Electronics Research (CASPER) at Berkeley, since their integrated FPGA systems are heavily utilized in the early Omniscope prototypes implemented in this thesis. The CASPER group openly develops hardware and complementary software tools specifically tailored to the implementation of radio astronomical instruments. The advantage of using CASPER hardware is that early prototypes can be produced far more rapidly than would be possible otherwise.

ROACH

The ROACH (Reconfigurable Open Architecture Computing Hardware) is an integrated system, featuring at its core a Virtex-5 FPGA and a PowerPC processor. The embedded processor is the primary interface with which we program the FPGA, set the parameters of the design and retrieve data from the computation. The FPGA itself has access to a number of important hardware peripherals—including four 10 Gigabit Ethernet ports, two $2\,\mathrm{M} \times 18$-bit Quad Data Rate II (QDR) SRAMs, one DDR2 DRAM DIMM (populated with a 1GB RAM module), two Z-DOK connectors (featuring 40 differential pairs) and several General Purpose Input-Output (GPIO) ports. A system diagram of the ROACH is shown in Figure 3-1.

The main interface to the embedded PowerPC is a standard 10/100/1000T Ethernet port. Through this, we can interact with the ROACH via a customized Linux kernel. Most importantly for our purposes, however, there exist Python packages (namely, the 'corr' package) that provide programmatic access to the important features of the device—including the ability to program the FPGA, set and read software registers that can be read and written to by the FPGA and retrieve computational data stored in shared Block RAMs (BRAMs).

The CASPER DSP Tool-set

CASPER also openly develops a set of tools for designing and implementing DSP systems for the FPGA. The tools are built upon the Xilinx System Generator tool, which enables designers to create systems as block flow diagrams in MATLAB's Simulink package and directly compile them to a hardware description language (HDL, such as Verilog or VHDL), which can then be synthesized into a netlist by the standard Xilinx tools and finally converted into a bitstream for configuring the FPGA.

In addition to the many useful DSP blocks provided by Xilinx (including basic components like adders, multipliers and accumulators, as well as FIR filters and more complicated subsystems), CASPER provides several blocks that are useful for radio astronomy, such as efficient biplex FFTs, transposes and data reorders. In addition, there are several constructs implemented especially for CASPER hardware, such as a QDR vector accumulator, a 10GbE transmitter/receiver and drivers for certain ADC daughter boards.

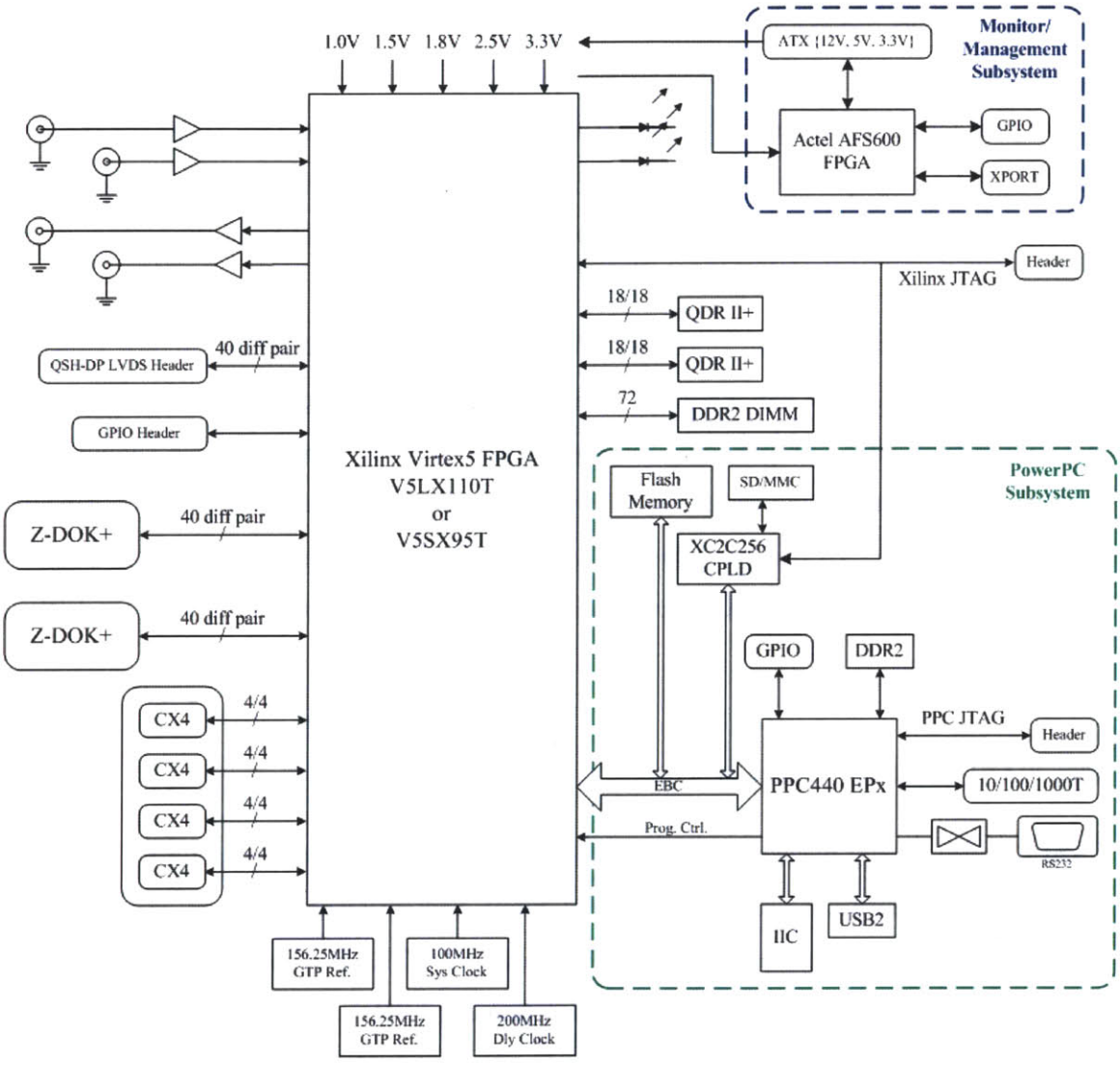


Figure 3-1: System diagram of CASPER's ROACH board from the CASPER documentation. The ZDOK connections on the left interface to the ADCs and the CX4 ports are the 10Gb Ethernet links.

Figure 3-2: MWA Antenna.

3.2 Analog Chain

The core analog chain consists of four major components: the antennas, the filtering and amplification stages, the receiver board and the ADC. The filtering, amplification and receiver stages (including quadrature demodulation) were designed by Eben Kunz, a fellow member of the Omniscope team, with help from Kris Zarb-Adami and his group from Oxford University and from MIT graduate student Andy Lutomirski. The ADC, though technically part of the analog chain, is described in the F-Engine section. The antennas that we use in the Omniscope prototype have a design borrowed from the MWA experiment. They behave roughly like dipole antennas and measure two orthogonal polarizations. They are shown in Figure 3-2. The antennas contain a low-noise amplifier (LNA) that is powered by a 5 V DC bias inserted into the antenna cables. Following the antenna is a low-pass filter (that eliminates frequencies beyond the range we are interested in, which is up to about 200 MHz).

Following the LNAs, the signal is still relatively weak and does not contain enough power to be reasonably transported from the field to the F-Engine. Each channel of the antenna therefore has a line-driver local to it. Figure 3-3 shows a schematic of the line-driver. We can see that it delivers a DC bias (cleaned through an RF choke coil) to power the antennas and then amplifies the RF signal using three broadband monolithic 19 dB amplifiers—amounting approximately 51 dB of gain after resistive losses. Also included in the line-driver, though not shown in the diagram, is a notch filter that suppresses the FM radio band, since these signals are generally very large compared to the signals we want and may end up saturating the amplifiers.

The signal is then delivered through 50 m of 75 Ω coaxial cable to the receiving equip-

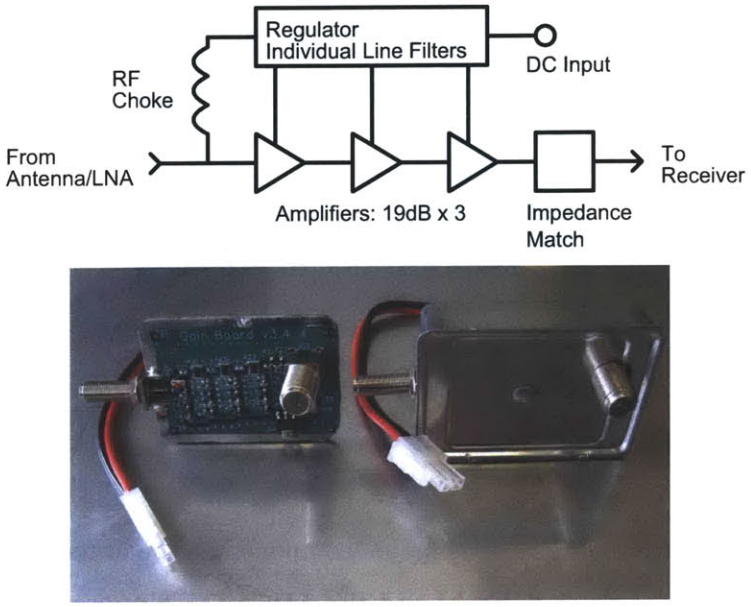


Figure 3-3: High level schematic of the line driver (above) and the hardware implementation of the same (below). The circuit is enclosed in a metal case to minimize RFI leakage.

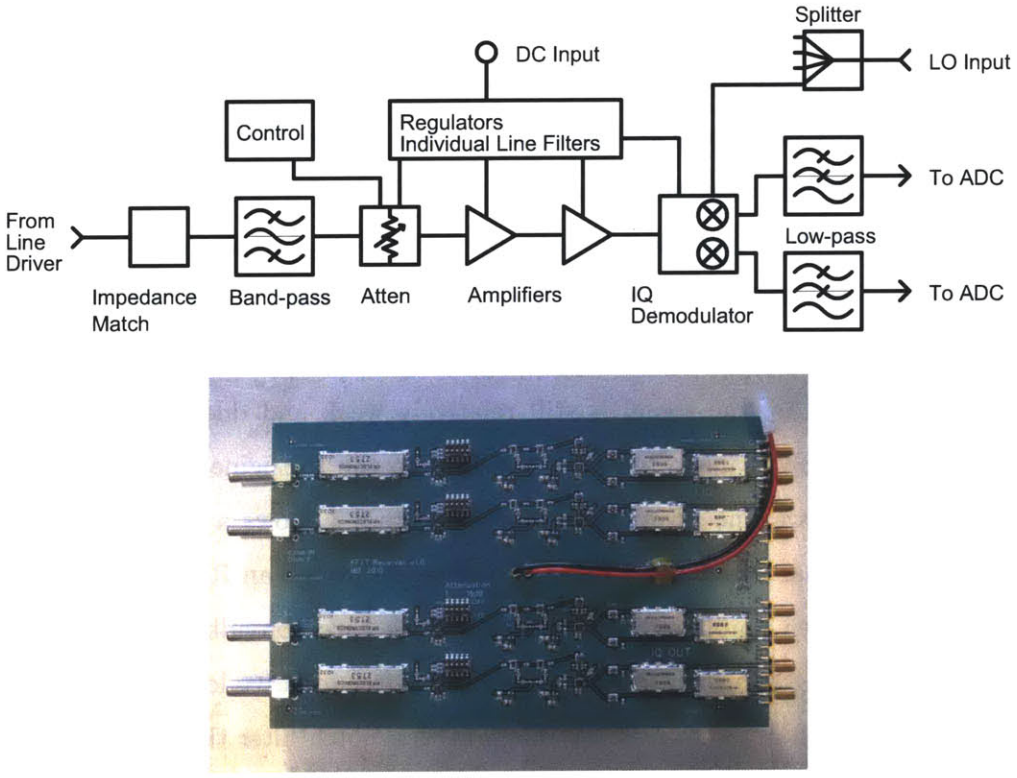


Figure 3-4: High level schematic of the receiver board (above) and the hardware implementation of the same (below).

ment. A schematic of the receiver board is shown in Figure 3-4. This board uses a band-pass filter to eliminate frequencies outside of our band of interest (in this case 110-190 MHz). The signal then passes through a variable attenuation stage and undergoes additional amplification. Finally, we reach the IQ-demodulator. A tunable local oscillator (LO) is externally supplied, which allows us to select the center frequency of the band we wish to observe. The IQ-demodulator multiplies the incoming signal by two sinusoids (at half the LO frequency) that are $\pi/2$ radians out of phase. Thus, following the low-pass filters on the output, we form the I and Q components of the signal as described in Section 2.3.1.

3.3 Phase Switching

One phenomenon that has plagued radio telescopes for a long time is cross-talk. Cross-talk occurs when the analog signals from one channel leak into another before digitization. Cross-talk is important because we are most interested in the correlation between different channels. Since the leaked signals are certainly correlated with the channels that induce them, cross-talk introduces the problem of distinguishing the correlations caused by analog electronics (which we don't want) from the correlations resulting from the sky signal itself (which we do want).

A powerful technique for combating cross-talk is to modulate each channel with a distinct Walsh function as close to the antenna as possible and then demodulate after digitization. This technique has been implemented in major radio telescopes in the past, and in this section, we describe our implementation. Before we do this, we give a formulation of the cross-talk problem and a mathematical exposition of Walsh modulation, including its application to the cross-talk problem.

3.3.1 Cross-Talk

Consider an N element array. Denote by X_i the signal that would be received by the F-Engine on channel i if the channel were perfectly isolated and free of system noise and let H_i be the additive system noise on channel i. Then the signal actually received on channel i can be modeled as

$$Y_i = \sum_{j=0}^{N-1} C_{ij} X_j + H_i.$$

The coefficients C_{ij} represent the relative strength of the cross-coupling onto channel i of signals from channel j (normalized so that $C_{ii} = 1$). In the interferometer, the measured cross-correlation is therefore

$$\langle Y_i Y_j \rangle = \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} C_{ik} C_{j\ell} \langle X_k X_\ell \rangle + \sum_{k=0}^{N-1} C_{ik} \langle X_k H_j \rangle + \sum_{\ell=0}^{N-1} C_{j\ell} \langle X_\ell H_i \rangle + \langle H_i H_j \rangle . \quad (3.1)$$

Thus, we can explicitly see the effect of cross-talk in computing visibilities. This cross-talk is generally frequency dependent, so we can imagine (3.1) as applying to a single frequency band.

3.3.2 Walsh Modulation

A Hadamard matrix is a square matrix consisting only of the values ± 1 with the property that the rows are mutually orthogonal. These matrices are known to exist for all powers of 2 and they can be constructed iteratively in the following fashion,

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad H_{2^n} = \begin{pmatrix} H_{2^{i-1}} & H_{2^{i-1}} \\ H_{2^{i-1}} & -H_{2^{i-1}} \end{pmatrix}.$$

Each row of a Hadamard matrix is a Walsh function. We let $W_i[n]$ denote the nth component of the ith Walsh function (the ith row of the Hadamard matrix). Harmuth demonstrated the applicability of Walsh functions in communications ([4]), and literature has adopted much of his notation—including the names Cal_N and Sal_N for the even and odd Walsh functions. In this section, we show that we can use Walsh functions to suppress the effects of cross-talk.

It is known that the set of Walsh functions of length $N = 2^n$ form a mathematical group, G_N under component-wise multiplication, where the Walsh function with all values unity is the identity. Hence, the product of two Walsh functions is another Walsh function and each Walsh function has a unique inverse (itself). We denote by $w_i \in G_N$ the group element corresponding to W_i. An important subgroup of G_N is the set of even Walsh functions, Cal_N—that is, those Walsh functions that are symmetric about their midpoint. This is a subgroup because the product of two even Walsh functions is both even and Walsh function and therefore an even Walsh function.

Lemma 2. *Let G_N be the group of Walsh functions of order N, where $N = 2^n$ for some*

integer n, and let H be a subgroup of G_N. For each (i,j) such that $w_i, w_j \in G_N$ and $w_i w_j \in H$, define

$$\mathcal{A}_{ij}^H = \left\{(k,\ell) \in \mathbb{Z}^2 \mid w_k, w_\ell \in G_N \wedge w_k w_\ell = w_i w_j\right\}.$$

Then the order of $\mathcal{A}_{ij}^H$ is the order of H for all i and j. Furthermore, any two $\mathcal{A}_{ij}^H$ and $\mathcal{A}_{k\ell}^H$, are either disjoint or equal, with equality holding if and only if $w_i w_j = w_k w_\ell$.

Proof. For a given i, j let $w_s = w_i w_j$. Because H is a group, for any k such that $w_k \in H$, there is exactly one ℓ such that $w_k w_\ell = w_s$, namely $w_\ell = w_k^{-1} w_s = w_k w_s$. Since there are $|H|$ such values of k, each with a unique ℓ such $(k,\ell) \in \mathcal{A}_{ij}^H$, it follows that $\mathcal{A}_{ij}^H$ has order $|H|$.

For the last result, suppose $(a,b) \in \mathcal{A}_{ij}^H$ and $(c,d) \in A_{k\ell}^H$. If $w_i w_j = w_k w_\ell$, then $w_e w_f = w_k w_\ell = w_i w_j$ for all $(e,f) \in \mathcal{A}_{k\ell}^H$ so $\mathcal{A}_{k\ell}^H \subset \mathcal{A}_{ij}^H$. By symmetry, we also have $\mathcal{A}_{ij}^H \subset \mathcal{A}_{k\ell}^H$ so that $\mathcal{A}_{ij}^H = \mathcal{A}_{k\ell}^H$. Otherwise, if $w_i w_j \neq w_k w_\ell$, then $w_e w_f = w_k w_\ell \neq w_i w_j$ for all $(e,f) \in \mathcal{A}_{k\ell}^H$ and $w_g w_h = w_i w_j \neq w_k w_\ell$ for all $(g,h) \in \mathcal{A}_{ij}^H$, so that $\mathcal{A}_{ij}^H \cap \mathcal{A}_{k\ell} = \emptyset$. □

The last result simply states that the distinct A_{ij}^H form a partition of $H \times H$. We can now form a useful lemma for the orthogonality of the Walsh functions.

Lemma 3. *Let $N = 2^n$ and $w_i, w_j, w_k, w_\ell \in \mathrm{Sal}_N$, where Sal_N is the set of odd Walsh functions. Then*

1. $\displaystyle\sum_{n=0}^{N-1} W_i[n] W_j[n] = N\delta_{ij}.$
2. $\displaystyle\sum_{n=0}^{N-1} W_i[n] W_j[n] W_k[n] = 0.$
3. $$\sum_{n=0}^{N-1} W_i[n] W_j[n] W_k[n] W_\ell[n] = \begin{cases} N & \text{if } (k,\ell) \in \mathcal{A}_{ij}^{\mathrm{Cal}_N} \\ 0 & \text{otherwise.} \end{cases}$$

where δ_{ij} is the Kronecker delta function.

Proof. The first sum follows from the orthogonality of the Walsh functions. Since the product of two Walsh functions is another Walsh function, say $W_i[n] W_j[n] = W_s[n]$, if the product is not equal to the identity Walsh function, then it is orthogonal to it. Hence, $\sum_{n=0}^{N-1} 1 \cdot W_s[n] = 0$. For the second expression, note that since $w_i, w_j \in \mathrm{Sal}_N$, $w_i w_j \in \mathrm{Cal}_N$

(since the product of two odd Walsh functions must be even). Hence, $w_i w_j$ is distinct from and therefore orthogonal to w_k for all $w_k \in \mathrm{Sal}_N$.

For the last expression note that for all $w_k, w_\ell \in \mathrm{Sal}_N$, $w_k w_\ell = w_i w_j$ if and only if $(k,\ell) \in \mathcal{A}_{ij}^{\mathrm{Cal}_N}$. By orthogonality, the sum is non-zero for fixed i,j if and only if $(k,\ell) \in \mathcal{A}_{ij}^{\mathrm{Cal}_n}$. □

Suppose that we are measuring N channels, where $N = 2^n$ for some n. We assign to each channel i, a particular Walsh function $u_i \in \mathrm{Sal}_{N+1}$. Suppose that we modulate the signals X_i by multiplying by $U_i[n]$. Suppose further that we demodulate the signals as they are received by multiplying again by the same Walsh function. That is,

$$Y_i[n] = U_i[n] \sum_{k=0}^{N-1} C_{ik} U_k[n] X_k + U_i[n] H_i.$$

Then

$$\begin{aligned} Y_i[n]Y_j[n] = {} & U_i[n]U_j[n] \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} C_{ik}U_k[n]X_k C_{j\ell}U_\ell[n]X_\ell \\ & + U_j[n]H_j \sum_{k=0}^{N-1} C_{ik}U_k[n]X_k + U_i[n]H_i \sum_{\ell=0}^{N-1} C_{j\ell}U_\ell[n]X_\ell \\ & + U_i[n]H_iU_j[n]H_j. \end{aligned}$$

The analog of (3.1) is

$$\begin{aligned} (N+1)\langle Y_iY_j\rangle = {} & \sum_{n=0}^{N}\sum_{k=0}^{N-1}\sum_{\ell=0}^{N-1}(U_i[n]U_j[n]U_k[n]U_\ell[n])C_{ik}C_{j\ell}\langle X_kX_\ell\rangle \\ & + \sum_{n=0}^{N}\sum_{k=0}^{N-1}(U_i[n]U_j[n]U_k[n])C_{ik}\langle X_kH_j\rangle \\ & + \sum_{n=0}^{N}\sum_{\ell=0}^{N-1}(U_i[n]U_j[n]U_\ell[n])C_{j\ell}\langle X_\ell H_i\rangle \\ & + \sum_{n=0}^{N}(U_i[n]U_j[n])\langle H_iH_j\rangle. \end{aligned}$$

In each of these terms, we can move the sum over n to the inside. Therefore, from Lemma 3

we have

$$(N+1)\langle Y_i Y_j\rangle = \sum_{(k,\ell)\in\mathcal{A}_{ij}^{\mathrm{Cal}_{N+1}}} (N+1)C_{ik}C_{j\ell}\langle X_k X_\ell\rangle + (N+1)\delta_{ij}\langle H_i H_j\rangle,$$

or

$$\langle Y_i Y_j\rangle = \sum_{(k,\ell)\in\mathcal{A}_{ij}^{\mathrm{Cal}_{N+1}}} C_{ik}C_{j\ell}\langle X_k X_\ell\rangle + \delta_{ij}\langle H_i H_j\rangle.$$

Since $(i,j)\in\mathcal{A}_{ij}^{\mathrm{Cal}_{N+1}}$, we can write this as

$$\langle Y_i Y_j\rangle = \langle X_i X_j\rangle + \eta_{ij},$$

where the "noise" η_{ij} is

$$\eta_{ij} = \sum_{(k,\ell)\in\mathcal{A}_{ij}^{\mathrm{Cal}_{N+1}}\setminus\{(i,j)\}} C_{ik}C_{j\ell}\langle X_k X_\ell\rangle + \delta_{ij}\langle H_i H_j\rangle.$$

This result shows quite clearly why Walsh modulation is advantageous. Note that the systematic additive noise has been completely removed from all cross-correlations. Furthermore, the cross talk matrix C_{ij} is generally 1 along the diagonal, and between 0 and ϵ, where $\epsilon \ll 1$, everywhere else. Note that if $(k,\ell)\in\mathcal{A}_{ij}^{H}$ and $k\neq i$, then $\ell\neq j$ (by the group properties). Therefore, all of the cross terms in η_{ij} are second order (i.e. less than ϵ^2), and from Lemma 2, there exactly N such terms. Hence, we can bound the cross-talk noise in cross-correlations by $N\epsilon^2$. This is far better than the $O(N\epsilon + N^2\epsilon^2)$ bound before phase switching. Furthermore, by choosing our Walsh functions from Sal_N, we can not only suppress cross-talk terms, but also completely eliminate systematic noise from the correlations.

3.3.3 Implementation

To implement the Walsh phase switching technique described above, we use a biphase modulator. A biphase modulator is a 4 port (including ground) analog device that inverts the polarity of an analog signal. A functional circuit diagram of the device is shown in Figure 3-5. As described in the figure caption, a control current selects which diodes turn

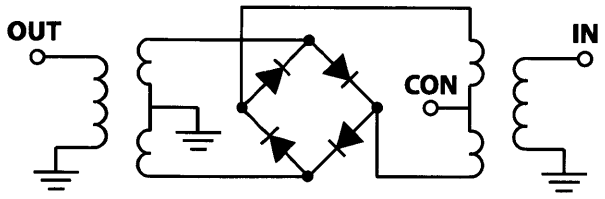


Figure 3-5: Electrical schematic of a biphase-modulator. A control current is sent into the CON terminal, which determines which diode path is active. If the current from CON to ground is positive, the upper left and lower right diodes are on. If it is negative, the upper right and lower left diodes are on. (Figure from the data sheet of the Mini-Circuits ZMAS-1).

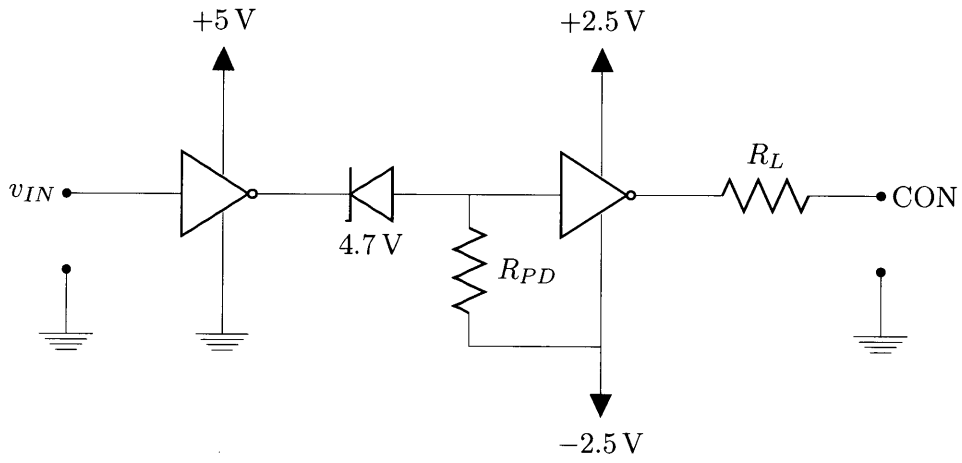


Figure 3-6: Circuit schematic for CMOS voltage to control current converter.

on. A positive (negative) current through the control terminal, CON, electrically connects the top of the input coil to the top (bottom) of the output coil and the bottom of the input coil to the bottom (top) of the output coil. Assuming that the diodes are identical and that the input/output baluns are electrically symmetric about the control taps, the current will flow symmetrically into or out of the center taps of the inner coils, effectively isolating the control current from the input and output. The input current, however, flows unilaterally and so is transferred through both baluns to the output.

The biphase modulator performs the multiplication of the analog signal from a channel with its associated Walsh function. The F-Engine demodulates each channel the analog signals with the appropriate Walsh function and is therefore also responsible for controlling the biphase modulators. Thus, we need a circuit that will translate the FPGA logic levels into inverted and non-inverted control currents. Figure 3-6 demonstrates such a circuit for

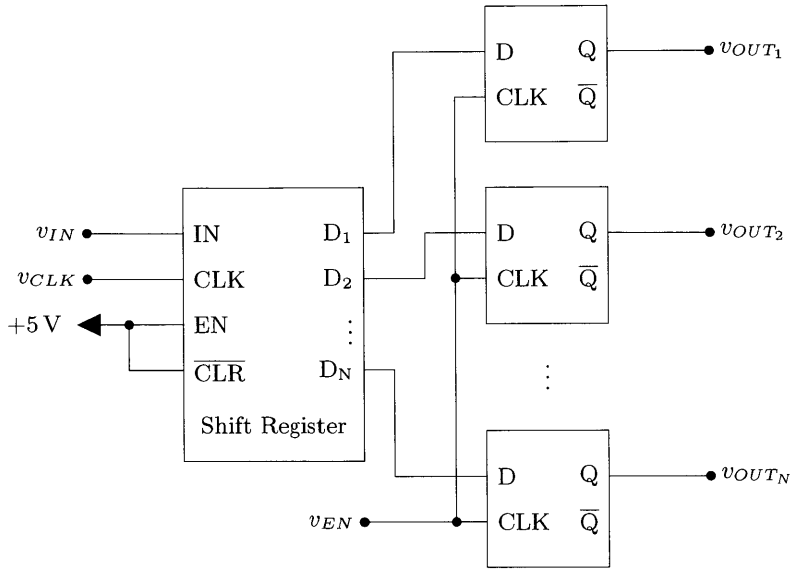


Figure 3-7: A simple deserializer for managing control outputs of the swapping system. The shift register performs the de-serialization, while the latches store the current Walsh function values in between switches.

a single channel.

Because the Walsh switching is happening several orders of magnitude slower than the FPGA clock speed, we can minimize the number of output pins coming from the FPGA by using a simple deserializer to manage the control signals. Figure 3-7 shows how this can be done with a shift register and a latch bank. The FPGA sends the sequence of Walsh function values for each antenna at the current sample time into v_{IN}. The clock at which these values are sampled is also sent in through v_{CLK}. When the full sequence has been pushed through the shift register (i.e. after N values), the FPGA pulses v_{EN} high to latch the new control values simultaneously across all channels. Each latch output v_{OUT_i} then goes to a converter circuit like that shown in Figure 3-6. Thus, the F-Engine need only send out three signals: v_{IN}, v_{CLK} and v_{EN} for an arbitrary number of channels. Because the F-Engine defines the control signals, digitally demodulating them is straightforward.

This circuit has been implemented for 8 channels on a protoboard, which is shown in Figure 3-8. The board is designed to drive a Minicircuits ZMAS biphase modulator, which takes a nominal control input of up to $\pm 20\,\text{mA}$. The control currents are delivered through CAT-5 cables in this prototype and are fed into the analog stack shown in Figure 3-9. The behavior of the swapper at the point of a Walsh function value change is shown in Figure 3-10. In this figure, we can see a transient effect on the order of $1\,\mu\text{s}$. For this reason, the

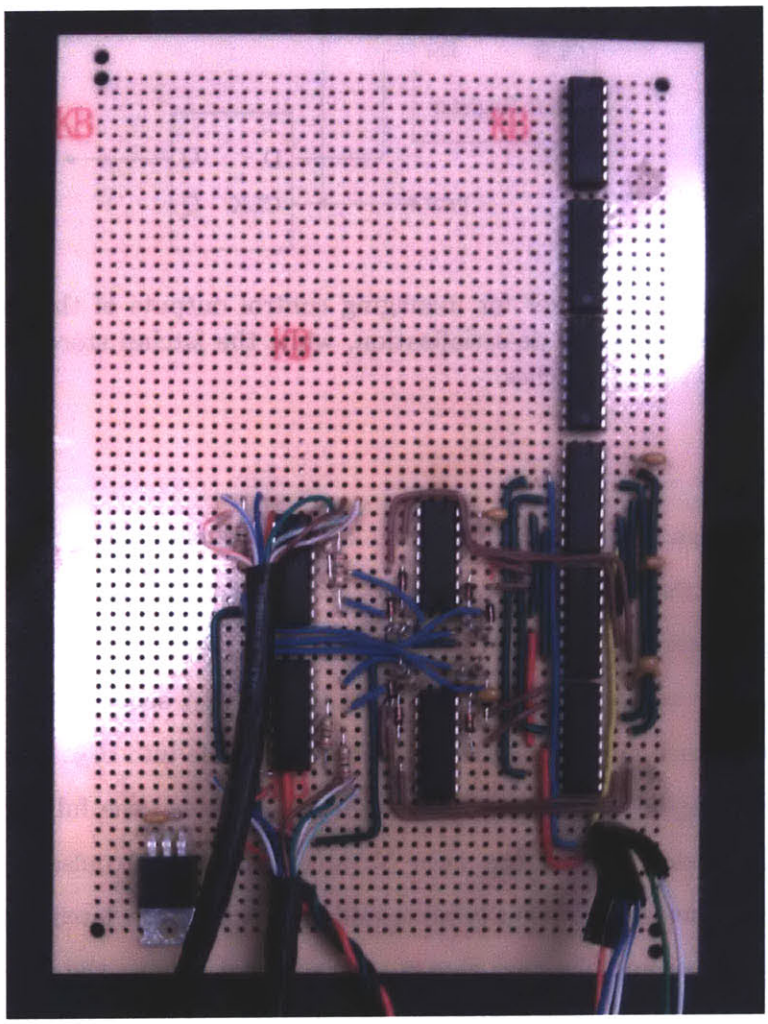


Figure 3-8: Prototype of the swapper system for 8 channels.

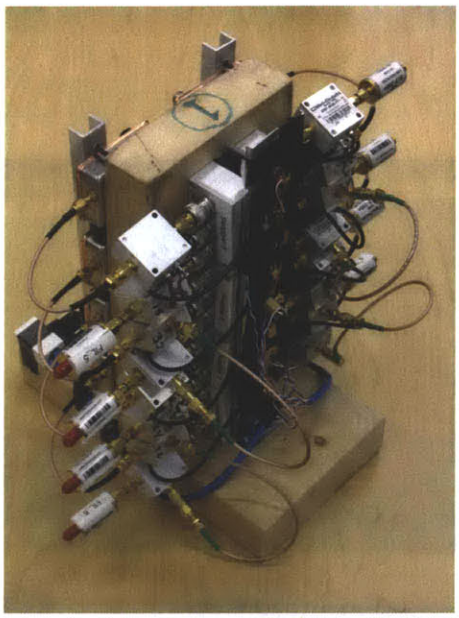


Figure 3-9: Analog stack for 8 channels, containing the line-drivers (back) and the biphase modulators (front). Bias-tees are used to send the DC bias around the biphase modulators to the antenna.

F-Engine will discard frames that encompass a phase switch. Since the frames are approximately 20 μs long, this means one frame per phase switch will be discarded. Discarding a frame simply means zeroing the values, since the frames are simply accumulated at the end. The resulting output, also clearly depicting the phase inversion, is shown in Figure 3-11.

3.3.4 Results

We tested the performance of Walsh modulation in cross-talk suppression in one channel. Specifically, we sent a heavily amplified wide-band noise source into one of the channels and left the other channels undriven. We can characterize the effect of cross-talk by observing the signal present on the other inputs and correlating it with the input signal. For this simple test, we used the alternating Walsh function $U_1 = (1, -1, 1, -1, \dots)$ to modulate the driven input, and the unit Walsh function $U_0 = (1, 1, 1 \dots)$ to modulate the rest. After demodulation, the driven channel should produce the original driving signal, whereas the other channels should see an alternating phase flipped version of the driving signal.

Let $X(t)$ be the amplified signal and $Y(t)$ be one of the undriven inputs. If we produce a scatter plot of $X(t)$ versus $Y(t)$ (i.e. a parametric plot over t with $x(t) = X(t)$ and $y(t) = Y(t)$), correlation between X and Y should manifest itself as a skew with non-zero

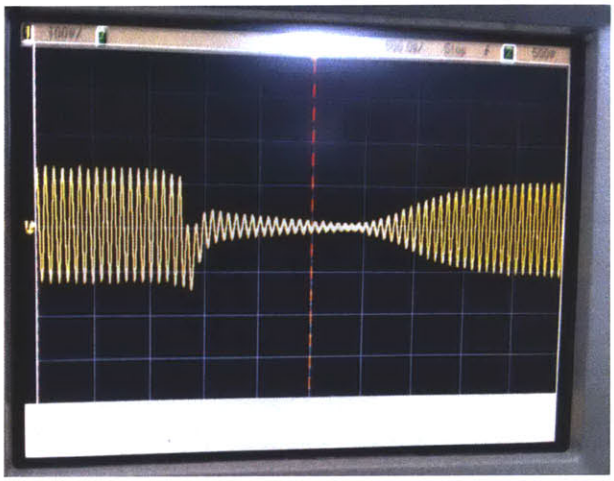


Figure 3-10: A swapped sine wave at the point of a Walsh phase flip.

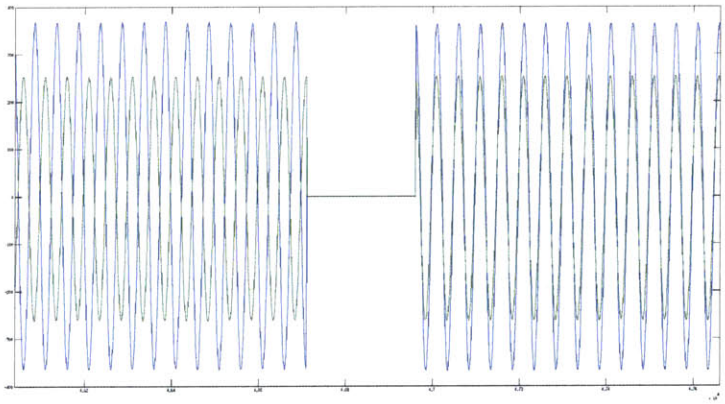
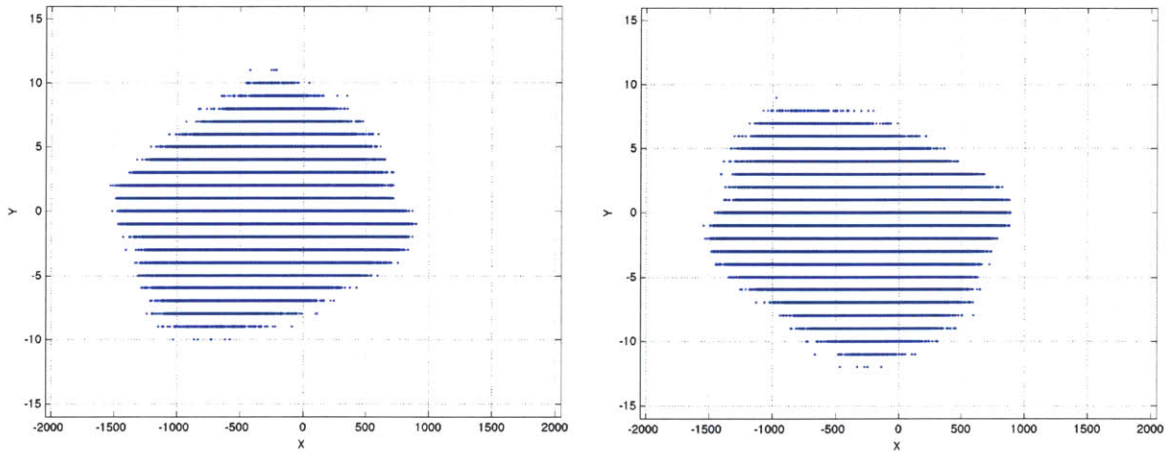


Figure 3-11: Discarding (zeroing) a frame during a Walsh phase flip.



(a) (b)

Figure 3-12: Scatter plot of $X(t)$ versus $Y(t)$ where each is modulated with a fully non-inverting (a) and fully inverting (b) Walsh function. Note that each plot is skewed with an opposite slope.

slope. Figure 3-12(a) shows such a scatter plot when all channels are modulated with U_0 (no inversion). There is a noticeable skew in the distribution with positive slope. This is more easily seen when comparing it to Figure 3-12(b), which is produced when all channels are modulated with $-U_0$. Applying a linear regression to the data shows that the first plot has a slope of 0.0010, while the second has a slope of -0.0011. If we then modulate and demodulate $X(t)$ with U_1, we get Figure 3-13, which has a fitted slope of 0.0000. This indicates that the correlation has been suppressed by the Walsh modulation.

We can more fully characterize the cross-talk performance by performing a frequency analysis on the cross correlations. We do this by estimating the correlation as

$$C = \frac{\langle X, Y \rangle}{\langle X, X \rangle}.$$

By computing the Fourier transform of C, we can obtain the frequency dependent cross-talk. Figure 3-14(a) shows the frequency dependent cross talk within a 25 MHz band measured before correlation. Figure 3-14(b) shows this same quantity measured when $X(t)$ is modulated with U_1 and $Y(t)$ is modulated with U_0. The latter plot was produced with only 20 components of the Walsh function and already shows almost a 40 dB reduction in cross-talk magnitude. Furthermore, the spectrum after modulation appears to now be noise dominated—which implies that the spectrum in Figure 3-14(b) can be integrated down by

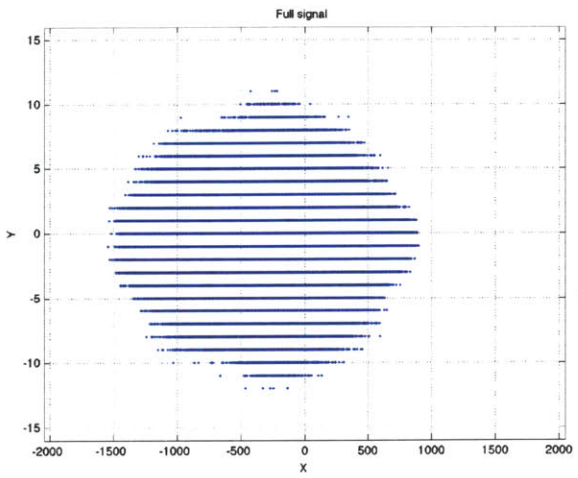


Figure 3-13: Scatter plot of $X(t)$ versus $Y(t)$, where $X(t)$ is modulated with U_1 and $Y(t)$ is modulated with U_0. Note that the skewing seen in Figure 3-12 is no longer present.

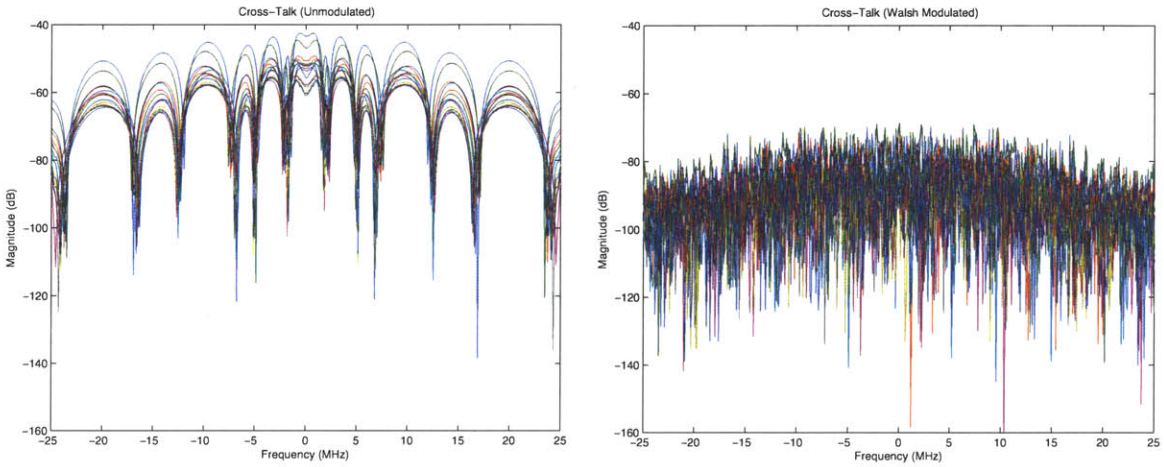
accumulating longer samples.

3.4 F-Engine

The device that was implemented as the "F-Engine" consists of an FPGA implementation of the overlap-add filter bank described in Section 2.4.1, but also encompasses the interface to the analog chain, the Walsh demodulator and to a certain degree, the corner-turner. It was implemented as described by Jonathan Losh, a current Omniscope team member, but is described here for completeness, since it is a vital complementary component to the X-Engine and also plays an important role in phase switching. Figure 3-15 shows the major functional components of the F-Engine that is described in what follows.

3.4.1 Analog to Digital Conversion

Because the ADC is a daughter board to the ROACH that houses the F-Engine and because it interfaces directly with the FPGA, it is described here rather than with the analog chain. The particular device that was used in the Omniscope prototypes is a 50 Msps 12-bit 64-channel ADC board, of which 32 are currently being used (8 antennas $\Rightarrow$ 16 polarizations $\Rightarrow$ 32 I and Q channels). The board itself operates at 200 MHz and is clocked by the FPGA. An HDL block to control and interface with this device through the FPGA was created by Jack Hickish (a former member of the Omniscope team), and provides a black-box that time-multiplexes the ADC channels into 16 signal buses (4 per wire). It is able to do this



(a) (b)

Figure 3-14: Frequency dependent cross-talk magnitude within a 25 MHz band before (a) and after (b) Walsh modulation.

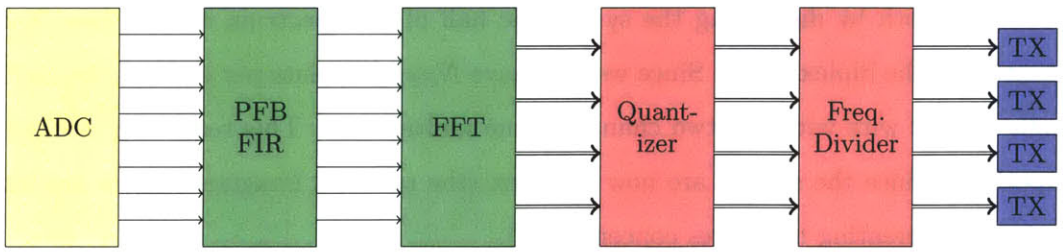


Figure 3-15: Functional diagram of the implemented F-Engine.

because the FPGA is clocked four times faster than the sample rate. The fact that the signals are time-multiplexed means that we can use the higher clock rate to reduce the amount of computational bandwidth required.

3.4.2 Polyphase Filter Bank

The data coming from an ADC channel is of the form

$$x_0(0), x_1(0), x_2(0), x_3(0), x_0(1), x_1(1), \ldots$$

Applying a reorder to the data (one of the standard DSP blocks provided by CASPER), we can effectively perform a transpose so that the data is available in the order

$$x_0(0), x_0(1), \ldots, x_0(N_{FFT} - 1), x_1(0), \ldots$$

However, it is practical to do this after the PFB FIR filter (which weights, overlaps and adds frames of N_{FFT} points for a specified number of taps). The reason is that we can perform a smaller transpose (N_{FFT} points instead of $N_{taps}N_{FFT}$ points) and the FIR filter simply needs to have frames that are four times larger and hold each coefficient four times longer (practically, this means clocking the coefficient ROM four times slower).

When the data is reordered after the PFB, a single FFT block can compute four channels in series that would otherwise be in parallel at a slower clock rate. Furthermore, a biplex FFT is included in the CASPER DSP library. This particular FFT implementation can take two signals that are assumed real, and use a single radix two butterfly to compute the FFTs of both by discarding the symmetric half of the spectrum of each (see [3] for a description of the biplex FFT). Since we now have $N_{FFT}/2$ points per channel, we can have a single output wire with the two channels time multiplexed. This reduction in data rate is superficial, since the signals are now complex (the real and imaginary parts are bitwise concatenated, creating two wires conceptually).

3.4.3 Quantization

It is worthwhile to take a digression on the issue of data representation, since it is important for considering the constraints of possible designs. In the DSP designs, data is represented

in fixed point binary. The data types are expressed as N_{bits}_BP, where N_{bits} is the bit width and BP is the binary point. Thus, 12_11 means a twelve bit value with the binary point at the 11th bit. The value represented by a 12_11 fixed point value is equal to the unsigned (or signed two's complement) binary value, divided by 2^{BP}. For example, 01101 = 1.625 and 10011 = −1.625 when interpreted as signed 5_3 fixed point values.

The data measured by the ADC is signed 12_11. The PFB FIR filter uses 18_17 coefficients and scales the output so that it is 18_17. Finally, the FFT truncates and parametrically scales each stage so that its output is also 18_17, though it is complex (hence, two concatenated data buses). This turns out to be too many bits for the downstream systems, so a quantization block is used to reduce the output to be 8_7.

Exactly how to perform this quantization, is important. In particular, since we are working with fixed point and not floating point, the magnitude of the value becomes important, since we are effectively reducing the dynamic range of the data bus. A signal that is very low, may end up only occupying a few low bits, while a signal that is very large will saturate. Thus, we would like to ensure that the signal is occupying the full dynamic range of the final data format by shifting it to the left or right. We in fact expect large values to appear at certain points in the spectrum (due to external interference or other signals), so this shift should be frequency dependent and tunable. The quantizer block does just this.

3.4.4 Packetization

The data is now ready to leave the F-Engine. Currently, the data is leaving the wires in the form

$$A_0(0), A_1, (0), \ldots, A_{K-1}(0), B_0(0), \ldots, C_0(0), \ldots$$

where $A_k(t)$ is the kth frequency bin of channel A at time sample t. The next step is to divide the frequency bands amongst the X-Engines. In the current prototype, there are four X-Engines (partially due to the fact that there are four 10GbE ports), so the bandwidth is divided into four and transmitted to the X-Engines over the 10GbE ports.

The current prototype has 32 channels (as mentioned previously), which implies 8 ADC buses containing 4 real time-multiplexed channels. After the quantizer, we have 4 signal buses, each containing one fourth of the frequencies for 32 complex time-multiplexed signals.

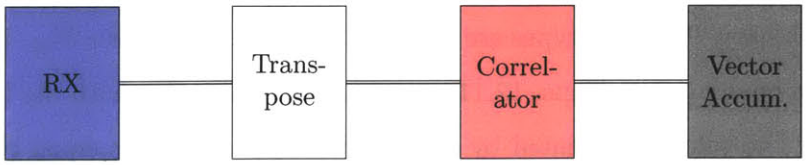


Figure 3-16: Functional diagram of the implemented X-Engine.

Thus, each of the four 10GbE transmitters is sending 16 bits/cycle (each data point is a complex 8-bit value). This amounts to 3.2 Gb/s of network bandwidth, leaving plenty of network bandwidth for UDP headers and other overhead. Since we only have four X-Engines and one F-Engine, a butterfly corner-turner is unnecessary, since the 10GbE ports can be routed directly.

3.4.5 Walsh Controller

The F-Engine module also plays a dual role as the controller of the Walsh phase switcher. This makes practical sense because the F-Engine is in the best position to perform the Walsh demodulation. Thus, by generating the control signals, it knows how to demodulate each channel, and also knows when to discard (zero) a frame in between a switch. This component, also implemented in the FPGA by Jonathan Losh, stores a set of Walsh functions in a ROM and periodically sends them to the swapper after a configurable number of frames. It does this by generating a clock signal (via a counter), that also addresses the ROM. The values are therefore being moved through the shift register in the swapper's deserializer. Once all of the values are sent, the enable output is pulsed, securing the new Walsh function values. The F-Engine can then directly invert (or not invert) the output of each ADC channel based on the channel's known Walsh function.

3.5 X-Engine

Each X-Engine is responsible for performing the correlations across all channels for its allotted band of frequency bins. Since N is small for this prototype Omniscope, a full correlator, rather than an FFT correlator, will be used. This has a number of advantages in the early stages, since we need the full correlations anyway for calibration (obviating then need for a separate, low-bandwidth full-correlator) and since calibrated quadrature demodulation can be performed in post-processing. A functional diagram of the X-Engine

is shown in Figure 3-16.

3.5.1 The Correlator

The core of the X-Engine is the correlator. Logically, the correlator is a block that takes in the channel values $A, B, C, \ldots$ on a serial data bus, and produces the values

$$\begin{array}{ccccc} AA^* & AB^* & AC^* & \ldots & AX^* \\ & BB^* & BC^* & \ldots & BX^* \\ & & CC^* & & CX^* \\ & & & \ddots & \vdots \\ & & & & XX^* \end{array}$$

Note that the omitted lower half triangle of the output matrix is the complex conjugate of the upper half triangle. An interesting feature of this computation, as compared to the other FPGA systems described, is that the number of outputs far exceeds the number of inputs (N input data points produce $\frac{N(N+1)}{2}$ output data points). Rather than solving this with N parallel outputs, the best way to handle this is to include some internal accumulation of length T within the correlation algorithm so that a single output can be used to serially transfer the $\frac{N(N+1)}{2}$ values. Thus, by the time one full block of correlations is read, N frames of channel values will have been fed into the correlator. Hence, each correlation has actually been accumulated over N frames.

Figure 3-17 shows a system that can compute the correlations using this idea. For it to work, assume that the data enters in the order

$$A(0), A(1), \ldots, A(T-1), B(0), \ldots, D(T-1)$$

By the time that $A(0)$ reaches node A, $B(0)$ will be at node B, $C(0)$ will be at node C and $D(0)$ will be at node D. Thus, the products, read from right to left, are AA^*, AB^*, AC^* and AD^* at time 0. As the system runs, it accumulates such correlations up to time $T-1$. When the frame finishes, EOF is pulsed and the muxes register these correlations in the delays. As the system continues to run, the channels present at nodes A-D have been shifted, so that B is at node A, etc. Thus, the new correlations being fed into the accumulators are BB^*, BC^*, BD^* and BA'^*, where A'^* corresponds to the first channel

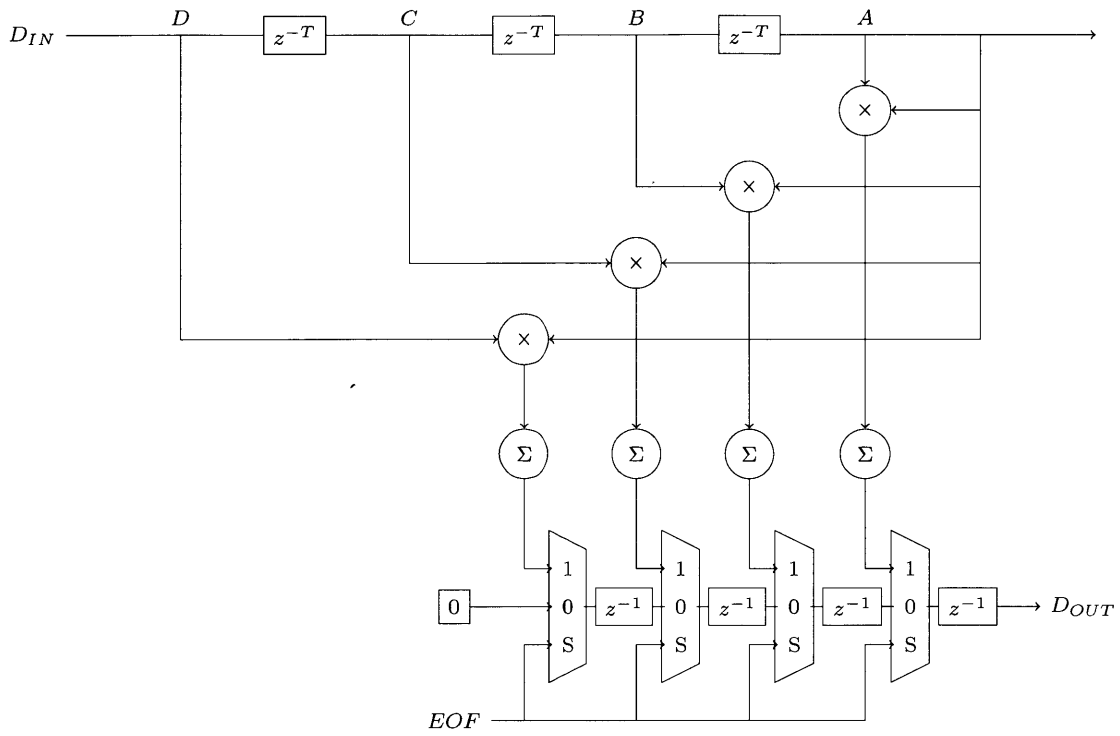


Figure 3-17: Full correlator system for four channels. The EOF signal also resets the accumulators. The multipliers are conjugate multipliers.

of the next frame (we ignore it and others like it in the output). While this correlation is being accumulated, the last accumulations can be read from the delay pipeline. Hence, as long as $T \geq N$, we will be able to read off all of the correlations by the time the next frame enters the delay pipeline.

3.5.2 Receiver and Transposer

The receiver module wraps the functionality of the 10GbE receiver and provides a continuous stream of data points. The data coming out has the form

$$
N \cdot T \left\{ \begin{array}{cccc}
A_0(0) & A_1(0) & \dots & A_{K-1}(0) \\
B_0(0) & B_1(0) & \dots & B_{K-1}(0) \\
\vdots & & & \vdots \\
A_0(1) & A_1(1) & \dots & A_{K-1}(1) \\
\vdots & & & \vdots \\
A_0(T-1) & A_1(T-1) & \dots & A_{K-1}(T-1) \\
\vdots & & & \vdots
\end{array} \right.
$$

This in not the format expected by the correlator. The arrangement into a matrix is motivation for performing a $(N \cdot T) \times K$ transpose on the data. The resulting data is ordered as

$$
\begin{array}{l}
T \left\{ \begin{array}{cccc}
A_0(0) & B_0(0) & \dots & X_0(0) \\
A_0(1) & B_0(1) & \dots & X_0(1) \\
\vdots & & & \vdots \\
A_0(T-1) & B_0(T-1) & \dots & X_0(T-1)
\end{array} \right. \\
T \left\{ \begin{array}{cccc}
A_1(0) & B_1(0) & \dots & X_1(0) \\
A_1(1) & B_1(1) & \dots & X_1(1) \\
\vdots & & & \vdots \\
A_1(T-1) & B_1(T-1) & \dots & X_1(T-1)
\end{array} \right. \\
\vdots
\end{array}
$$

The data is still not quite in the correct form. This arrangement reflects the fact that we consider correlations of different frequencies to be wholly distinct. Hence, we can produce the ordering expected by the correlator through another (considerably smaller) $T \times N$ transpose.

Because the first transpose is so large (131072 elements), it is favorable to use the ROACH's QDR SRAM to perform it. Unfortunately, as described in the next section, the

vector accumulator needs to use both QDR's. For this reason, we have implemented the transpose within the F-Engine, which is not using its QDR's. Given the 16-bit data lines, a single QDR can transpose two data buses, so that two QDRs are enough for the four X-Engines.

3.5.3 Vector Accumulator

The final stage of the X-Engine is the accumulator. The internal accumulation in the correlator was only to handle the quadratic output growth. The long accumulations must be performed in QDR. The (filtered) output of the correlator has the form

$$\begin{array}{cccccc} A_0A_0^* & A_0B_0^* & A_0C_0^* & \cdots & B_0B_0^* & B_0C_0^* & \ldots \\ A_1A_1^* & A_1B_1^* & A_1C_1^* & \cdots & B_1B_1^* & B_1C_1^* & \ldots \\ \vdots & & & & & & \\ A_{k-1}A_{k-1}^* & A_{k-1}B_{k-1}^* & A_{k-1}C_{k-1}^* & \cdots & B_{k-1}B_{k-1}^* & B_{k-1}C_{k-1}^* & \ldots \end{array}$$

Thus, we can view the data as a sequence of $\left[\frac{N(N-1)}{2}\right] K$ vectors, storing the correlations for the K frequencies allocated to the X-Engine. For the prototype, this is 67584 numbers. The internal accumulation on the correlator is set to 2^6 to avoid tight constraints on reading QDR vector accumulator provided by CASPER. The 8 bit data entering the correlator, leaves as $2 \cdot 8 + 1 + 6 = 23$ bit data. If we want to accumulate for a length of time on the order of a seconds, the bit growth is at least 15 bits. Since the QDRs are 36-bit, this means that we need a separate QDR vector accumulator for the real and imaginary parts (we cannot concatenate them).

3.5.4 Reading the X-Engine

To read the accumulations out of the QDR from the processor, we must use a shared BRAM (a Block RAM that can be accessed by the processor). The reason for using the QDR in the first place is that there isn't quite enough BRAM on the Virtex-5 to stare all of the data. Thus, at the end of a full correlation, the embedded processor continually attempts to fill the BRAM with QDR data, read and reset. This is accomplished by a Python script running on the PowerPC. The script gathers the correlation data, and than packages it into UDP to be sent via the 1000BASE-T connection to a listening computer. The receive scripts running on the listening computer then decode the UDP packets and append it to the data

already collected during the current observation—writing it to disk when the observation is over.

3.6 Software

3.6.1 Introduction

A significant part of my contribution to the Omniscope project involves the development of powerful software tools for monitoring the system in real-time, efficiently storing and organizing observation data and analyzing large data sets from many perspectives. Almost every part of these software components are written in Python, and they are separated into three major software packages. The ROACHViewer is an application framework and a collection of classes for quickly implementing real-time interfaces to inspect data shared by the ROACH. The OmniViewer is an application that enables researchers to quickly gain access to a wide variety of visualizations of the data captured during an observation. Finally, the Omniscope Data Format is a custom data format for storing and managing the data acquired by the Omniscope from an observation in a way that organizes the data as it goes through various steps of post-processing and manages the meta-data of the observation.

3.6.2 The ViewerCore Framework

Two of the major software packages described in the introduction have a similar purpose—the real-time graphical display of data. All of the software is heavily object oriented, so it makes sense to abstract away the functionality common to the two programs into a self-contained framework. This is the purpose of the 'ViewerCore' framework. The ViewerCore framework is built upon the wx-Python windowing library (a Python interface for the wx windowing system), and it contains two components: the Application classes and the `PlotCanvas` class.

Composing the Application classes are `App` (a wrapper for the top-level application), `NotebookPanel` (the top-level window that contains tabs for different functions and views) and `ViewerPanel` (which contains one or more `PlotCanvases` as well as relevant UI controls). To create an application using the ViewerCore package, one needs to create a custom application frame through wx (including menus, titles, etc.) containing a subclass of `NotebookPanel`, insert the relevant tabs into it and then instantiate `App` with the main

frame. This part is rather generic, as most of the interesting development comes with building the `ViewPanel`s.

The `PlotCanvas` class delivers the core functionality of a ViewerCore application. A plot canvas is a wx element that supports rendering from the matplotlib python package (a package that provides MATLAB-like graphing capabilities). The class encapsulates much of the functionality that might be desired in a ViewerCore application. Particularly, fast live graphics and the potential to update continuously in real-time. To use `PlotCanvas`, one simply subclasses it and overrides the `update_plot` method to provide the desired plotting functions. The class includes handles to indicate when the plot should be updated (so that plot updates can be triggered by user input) and can optionally be configured to continuously update in a separate thread.

An application built atop the ViewerCore framework is expected to contain several relevant categories of information (e.g. time-domain data, spectral data, sky maps, etc.). Each of these view categories corresponds to a different `ViewPanel`. Within a `ViewPanel`, one or more `PlotCanvas`es can be inserted. Below these views, a set of controls relevant to the panel can be inserted. It is useful to have a global application state that reflects the user input, so the framework provides a means for sharing state amongst `ViewPanel`s and routines for keeping UI controls consistent with this state.

3.6.3 The ROACHViewer

An immediate application of the ViewerCore framework is the development of real-time system inspection of a running Omniscope. Fortunately, there is a Python package (corr) that facilitates programmatic access to data on the ROACH through the PowerPC interface. For a particular FPGA design, one can create a ViewerCore application. By creating a threaded, continuous-update `PlotCanvas`es, data retrieval from the ROACH can be separated from data plotting and UI control, which preserves UI smoothness without costs in performance.

A basic ROACHViewer for an F-Engine has been implemented that consists of several major tasks: inspecting time-domain data, spectral data and the histograms of the same. The time-domain data provided in real-time by the ROACHViewer has been an indispensable tool for debugging the system during the setup and operation of a real field test of the Omniscope, as it can immediately show which channels are not working and which have too

much/too little gain. A quick glance at the histograms for any channel will immediately show whether the system gains are making sufficient use of the dynamic range of the ADC. The real-time spectral data provides quick glimpse of the composition of the signals being measured by the instrument, while the spectral histograms show whether we are truncating the data properly.

3.6.4 The Omniscope Data Format

Discussion

An observation with the Omniscope consists of a sequence of integrations. Each integration represents a single computation of the complex visibilities measured by the instrument (i.e. a single approximation of (2.3)). For our field tests, this integration time was about 5.37 seconds. An hour of observation, therefore, contains about 670 integrations. Each integration contains the correlations of the pairs of antenna channels. Each of these correlations contains the visibilities for a set of $N_{FFT}/2$ frequencies. Finally, each visibility consists of a real and imaginary floating point value. We can store all of the raw data from this observation into a multi-dimensional array D, indexed as follows,

`D[time][baseline][frequency][real/imag]`.

Here, `time` represents the time of the integration. Knowing that the visibility measured by the baseline (i, j) is the complex conjugate of that measured by that of (j, i), we can record the baselines measured by a full correlator by indexing them in the order

$$(0,0),(0,1),...,(0,N),(1,1),\ldots,(1,N),(2,2),\ldots,\ldots,(N-1,N),(N,N),$$

so that `baseline` is well-defined integer index corresponding to a particular baseline. After quadrature modulation has been performed, the signal spectrum measured by the ADC represents the band of interest shifted to the origin, which is then divided into $N_{FFT}/2$ frequency bins by the FFT. Thus, `frequency` represents a particular frequency band within the band of interest. Finally, `real/imag` is 0 or 1 for the real or complex part of the visibility, respectively.

This forms the basis of what we call a DataObject. Specifically, this is the data part of a VisibilityDataObject. Post-processing of this data yields many collections of data that

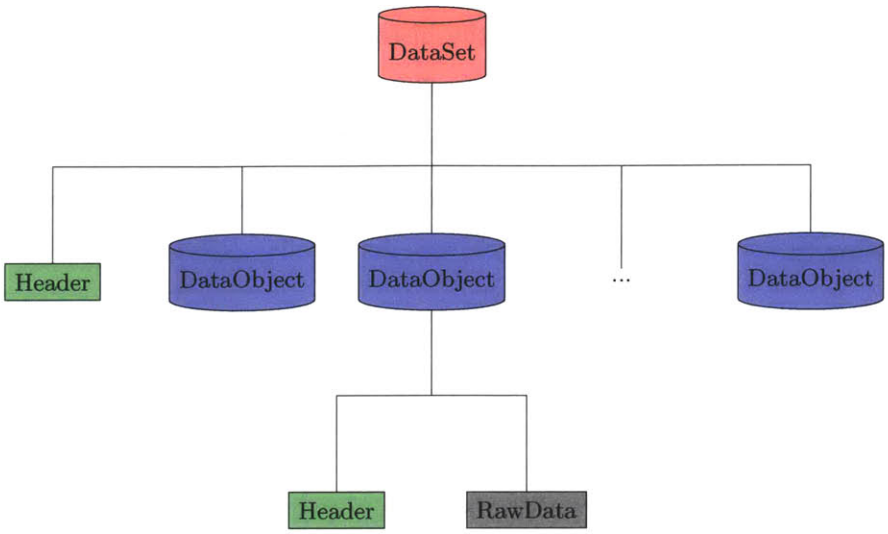


Figure 3-18: Class diagram of the core structures in the Omniscope Data Format.

have a similar format. These include estimations of the error bars for each data point and calibrated versions of the visibilities. Because it is so useful to analyze these collections of data alongside each-other and because the same post-processed files are to be created for a vast number of observations, it makes sense to collect the data for an observation into a unified data structure. This is the basis for the Omniscope Data Format (ODF).

The DataSet Structure

To unify and contain the above mentioned data collections, we introduce a high level data structure that we call a DataSet. Each DataSet corresponds to a particular observation with the instrument. As such, it contains the raw data, the post-processed data and meta-data about the observation. More concretely, a DataSet contains a header and one or more DataObjects. The header stores meta-information about the observation and DataSet, including the start time of the observation, the number of integrations in the data, the duration of each integration, the number of antennas in the instrument, the coordinates of the antennas, the number of frequencies (unfortunately given the name channels in the header, a remnant name not to be confused with the raw antenna signal channel), the start and end frequencies of the band measured, the latitude, longitude, and elevation of the observation point and finally the ODF version with which the file was written. An example such a header is shown in Table 3.1. The header is stored in plain text, so that it may be easily inspected and edited if necessary.

Table 3.1: Example values for a DataSet header.

Parameter Name	Example Value
`startDate`	Date('April 3, 2011, 07:56:60.0 UTC')
`nIntegrations`	668
`nAntennas`	8
`antennaPositions`	[[0.0, 0.0], [6.0, 0.0], [12.0, 0.0], [18.0, 0.0], [0.0, 6.0], [6.0, 6.0], [12.0, 6.0], [18.0, 6.0]]
`integrationTime`	5.3687090873700001
`nChannels`	512
`startFreq`	130.0
`endFreq`	155.0
`latitude`	45.330770000000001
`longitude`	-69.967253999999997
`elevation`	200.0
`format_version`	'0.2'

The DataObject Structure

A DataObject, generically represents a raw data array (stored in a raw binary file), along with a plain text header, similar to the DataSet header. The parameters in the DataSet header are generally those that apply to all of the DataObjects within the file. Information that is specific to a particular DataObject or differs from the DataSet header is provided in the DataObject's header. For example, the raw visibilities typically include the listing of polarizations that were correlated (e.g. 'xx,' 'xy,' 'yx,' 'yy'). A post-processed DataObject may only contain information about the most important 'xx' polarization correlation. This information would be placed in the DataObject's header.

A DataObject is an abstract structure and does not impose any particular structure its data array. Inheritance is used to create new and interesting DataObjects. A VisibilityDataObject, for example, is what is described in the introduction. An SdevDataObject represents the error bars on the visibilities and has the exact same structure as a VisibilityDataObject. However, since it is treated differently in analysis, it is treated as a different type of DataObject. A LinCalibrationsDataObject also has a similar structure to a VisibilityDataObject (it represents the calibrated visibilities), but also includes statistical data representing the quality of the calibrations. The type of a DataObject is stored within its header. Figure 3-18 summarizes the layout of a DataSet.

Each DataObject type corresponds to a Python class in the OmniReader framework.

The class defines the structure of the data array and how the data is to be interpreted. Inheritance can be used to build up a comprehensive collection of DataObject types by building on incremental changes between DataObject types. Because the framework described is so generic, the Omniscope Data Format can easily be extended to contain new DataObject formats.

Advantages

There are other formats for storing astronomical data, such as MIRIAD, but the Omniscope Data Format is well suited for the Omniscope and particularly well suited for the OmniViewer, described in the next section. To begin, ODF attempts to be very generic but does not attempt to provide extensive 'out of the box' functionality—a daunting task that is unfitting for a project with rapidly evolving requirements. Rather, ODF provides a framework in which robust data structures can be developed and unified. This has two major benefits. It places very few constraints on the types of DataObjects that can be created, since its structure and interface is given by the designer in the form of a Python subclass. Furthermore, since the structure and interface is controlled by the designer, the data can be stored straightforwardly as raw binary data, which can provide several orders of magnitude in performance enhancement for reading and writing large data collections. Finally, since the ODF version is explicitly stored in a file, the format specification can evolve rapidly while the framework can easily stay backward compatible. Finally, the format is easily portable to other languages. In fact, a C++ version of the ODF framework exists for integration with our synthesis tools.

3.6.5 The OmniViewer

Introduction

The OmniViewer is by far the most sophisticated application of the ViewerCore framework. It is a powerful tool that provides a unique perspective into the data acquired during an observation. It is specifically designed for easy and intuitive inspection of a DataSet. The need for such a tool easily follows from the dimensionality of the data in a DataSet. A DataSet contains a number of DataObjects, which contain a number of integrations, that compute a number of baseline correlations, which represent the visibilities for a number of

frequencies, which represent a pair of real and imaginary values. Thus, if one separates the polarization correlations described when discussing the VisibilityDataObject, a DataSet is effectively a 6-dimensional function. Many different projections of this function have very different scientific significance and should be displayed in such a way that is useful in each situation.

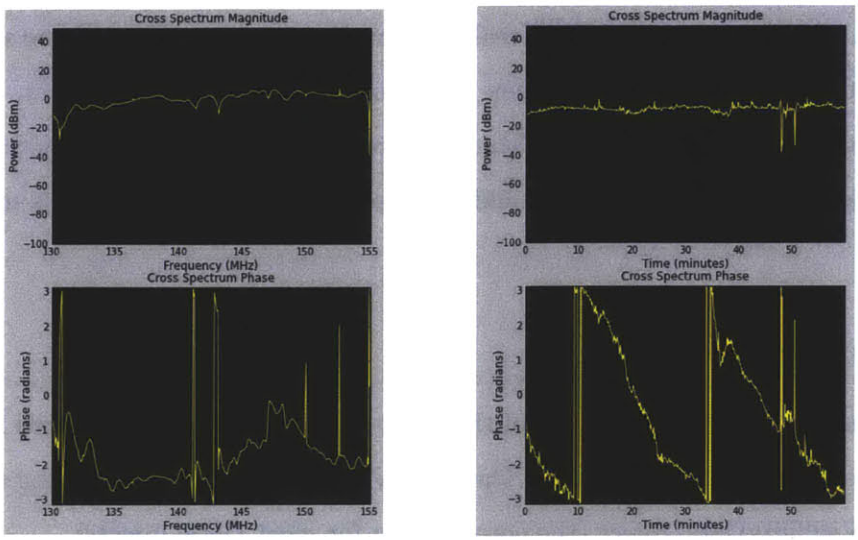
Viewing a DataSet

There are many different projections of the many-dimensional function stored in a DataSet that provide meaningful information. The OmniViewer is capable of providing the following views of a VisibilityDataObject (the core pattern for most DataObjects)

- *Spectrum:* A projection of the data onto the frequency axis for a particular baseline and integration time. Because it is complex, it is composed of a 1-dimensional magnitude and phase plot (see Figure 3-19a).
- *Time Trace:* A projection of the data onto the time axis for a particular baseline and frequency—also composed of a magnitude and phase plot (see Figure 3-19b).
- *Complex Plane:* A projection of the data onto the baseline axis for a particular time and frequency. The values are represented as points (each corresponding to a baseline) on the 2-dimensional complex plane (see Figure 3-19c).
- *Dirty Map:* A crude sky map derived from the complex plane data for a particular time and frequency. The data is used to construct (via Fourier inversion) a real-valued 2-dimensional intensity map (see Figure 3-19d).
- *Waterfall:* A projection of the data onto the frequency-time plane for a particular baseline. This is represented as a pair of 2-dimensional intensity maps, depicting the magnitude and phase (see Figure 3-19e).

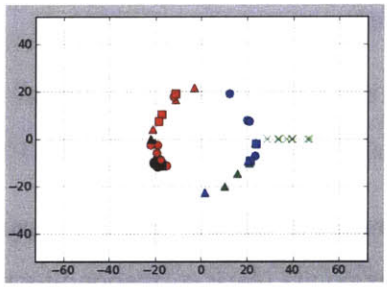
Features

In addition to providing several robust perspectives of data collected from an observation, the OmniViewer has several features that aid in analyzing the data. Multiple tabs organize relevant plots together with UI controls for selecting the parameters of the different projection plots. The main tabs of the OmniViewer are:

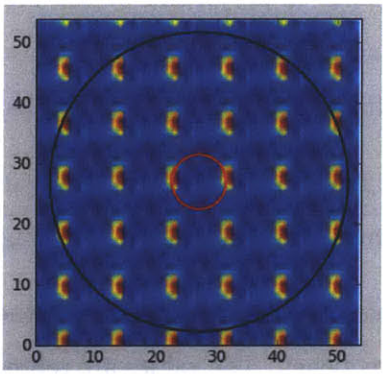


(a)

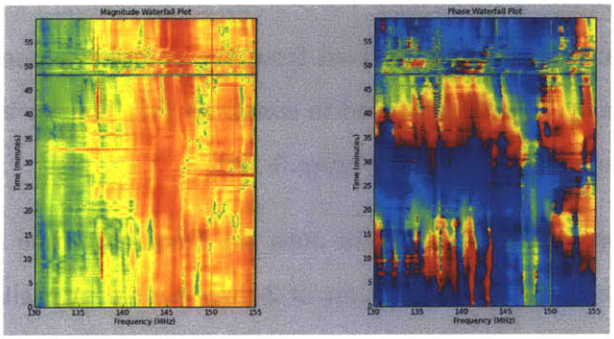
(b)



(c)



(d)



(e)

Figure 3-19: The five main perspectives of a VisibilityDataObject. (a) The spectrum, (b) the time trace, (b) the complex visibilities, (c) the Dirty Map, and (c) the waterfall plot.

- *Correlation:* This tab combines the spectrum, trace and complex plane plots to provide a detailed overview of the data.

- *Map Viewer:* This tab adds a dirty map to the plots feature in the Correlation tab

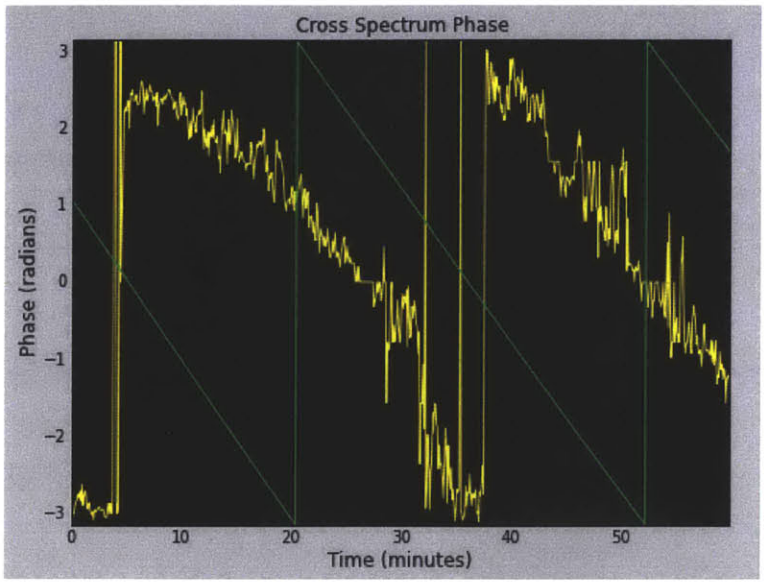


Figure 3-20: Phase fringes of the sun (yellow). As the sun rises, the phase difference between two antennas separated east to west will decrease. The simulated phase fringes (green) of the sun at that time, location and baseline are superposed. The offset is most likely due to cable length errors, but the important part is that the slope is the same.

and augments the UI with simulation functionality to compare the data to simulated sources.

- *Calibration:* This tab operates specifically on CalibrationDataObjects and displays the calibration factors and the goodness of fit of the calibration.

- *Waterfall:* This tab contains only the waterfall plot.

The simulation capabilities of the OmniViewer are very useful in identifying features that stand out in the data. Upon loading a file, the OmniViewer loads a database of ephemeris data for satellites and other man-made bodies, various stellar objects and the sun. Using this information, along with the latitude, longitude and elevation from the DataSet header, one can reconstruct the signals expected from these sources. The OmniViewer allows the user to select a particular object of interest and superpose a simulated signal atop the actual data. Figure 3-20 shows example of this where we can very clearly see the fringes of the sun as it rises. The data matches very nicely with the simulation data.

Chapter 4

Conclusions

4.1 Summary of Results

This thesis has provided a synopsis of the entire Omniscope system as well as the important software components. In Chapter 1, I presented the background material necessary to understand the science application of the Omniscope and the motivation for attacking the challenges set forth by 21-cm tomography. I also provided an exposition of traditional radio telescopes and Omniscopes that allowed us to understand the governing principles behind each and identify the key properties of both.

Chapter 2 provided a functional overview of the Omniscope, detailing the function and requirements for each of the components. It also provided a comprehensive treatment of the poly-phase and overlap add filter banks as applied to spectral leakage. Chapter 3 provided a detailed account of the implementation of each of the components with a focus on the phase switching, the X-Engine correlator and the software—the components that I was fully responsible for. I showed that Walsh modulation could be used to significantly reduce cross-talk and showed how one can choose a particular set of Walsh functions to eliminate almost all unwanted signals from them measurements. I also built a successfully functioning phase switching system and provided promising test results for its application to the problem of cross-talk. Finally, I detailed the architecture and implementation of sophisticated Python software applications built atop a custom visualization framework (ViewerCore) and an original data format for storing Omniscope measurements (ODF). The first was the ROACHViewer, a tool for real-time monitoring of the ROACH systems, which has been immensely helpful for diagnosing the Omniscope during our field tests.

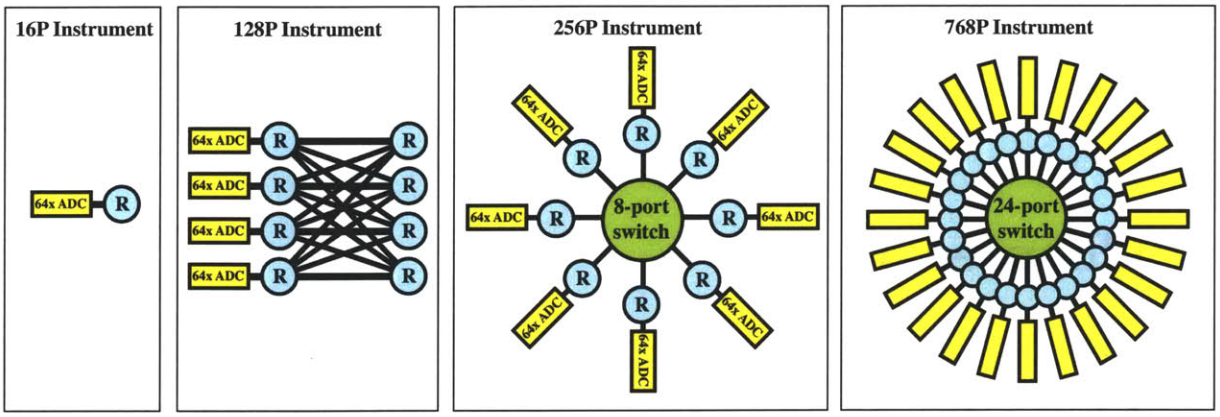


Figure 4-1: Possible network topologies for future Omniscopes. In the figure each circle "R" corresponds to a ROACH device and the yellow boxes correspond to the 64-input ADC boards. In many of the networks, adjustments may need to be made to the data bit width so that the F-Engines can also serve as X-Engines. If F-Engines are also X-Engines, then the different topologies simply express all of the ways that each ROACH can simultaneously communicate with the others. Since the ROACH is limited by 4 10GbE ports, general purpose switches will be needed once we exceed four ROACHes. Beyond a 24-port switch, a butterfly corner-turner will be required.

The second was the OmniViewer, a sophisticated data presentation tool for analyzing the data collected by the Omniscope. This tool has also been immensely helpful for diagnosing problems that manifest on longer time scales.

4.2 Future Directions

The 2×4 Omniscope prototype that we implemented was indeed successful and lays the crucial groundwork that is needed to begin scaling the Omniscope to larger arrays. The next Omniscope will feature 64 antennas—128P compared to the 16P we have now. There are several increments in scale that can be achieved with the hardware that we have now without requiring a butterfly corner turner. Figure 4-1 depicts these topologies. The topologies described in the figure can all be tested in West Forks, ME, just as the current Omniscope has been. However, very large Omniscopes will be deployed in Australia, where the MWA currently resides. This location is radio quiet and is also very flat, making it ideal for very large scale Omniscopes.

From a grander perspective, the Omniscope, as it will be developed in future generations, is a truly revolutionary instrument. Only recently has it been possible to even consider building radio telescopes at the scale of a million receiving elements. By surmounting the

problem of prohibitive cost scaling, the Omniscope opens the door to a new kind of radio astronomy—one that promises to unleash the power of 21-cm tomography. The idea of producing a piece of technology that can answer questions about the origin of the universe is one that I find immensely fascinating. It is the reason that I have devoted my thesis to elaborating the technical foundations on which future Omniscopes can be produced.

Appendix A

Zero-Padding

Zero-padding is the practice of appending zeros symmetrically to the beginning and end of a signal before Fourier transforming it. It has the effect of interpolating the spectrum of the signal. Clearly, adding zeros cannot possibly add information to the spectrum, so we do not gain any more 'detail' than that which was already present. Instead, zero-padding is a tool that can help us better understand what happens to discrete signals when they are translated by non-integer amounts.

To understand the effect of zero padding, it perhaps better to explain its effect in the reverse direction—that is, describing how zero-padding a spectrum affects the resulting time domain signal. Suppose we have a discrete signal $x[n]$ with a DTFT spectrum $\hat{x}(\omega)$. The spectrum will be periodic with period 2π. If we then convert the discrete signal $x[n]$ into a continuous time impulse train $x_c(t)$, the resulting spectrum remains unchanged. If we then apply an aliasing filter (low-pass with passband up to $\pm\pi$), the spectrum will be truncated and the signal will be interpolated via a convolution with a sinc function. This "smoothed" version of $x_c(t)$ contains the same information that $x[n]$ does, but has a smooth continuous shape.

If we sample the smoothed signal at a larger sample rate than $x[n]$ (i.e. the period of the sampling impulse train is larger than the one used to form $x_c(t)$ from $x[n]$) to form $x_2[n]$, we will have a "smoothed" version of $x[n]$ that is made up of samples from interpolation. The resulting spectrum $\hat{x}_2(\omega)$ is still periodic, but has a larger bandwidth. Since we truncated the spectrum of $x[n]$ when we applied the aliasing filter, $\hat{x}_2(\omega)$ will consist of $\hat{x}(\omega)$ surrounded by zeros on the left and right hand side. Thus, $x_2[n]$ is an interpolated discrete signal

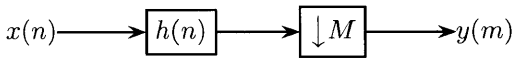
formed by *zero-padding* $\hat{x}(\omega)$.

In this paper, we discuss zero-padding done in the opposite direction (i.e. by padding the time-domain signal before Fourier transforming to get an interpolated Fourier spectrum). The zero-padded spectrum is important, because it tells us how DFT bins respond to a tone whose frequency lies between bins. It is proportional to the interpolated spectrum shifted by a non-integer multiple of the bin width.

Appendix B

The Polyphase Decimator

Consider decimating a signal $x(t)$ by a factor of M. Often, we wish to filter the signal before decimating it (e.g. with an anti-aliasing filter). The whole process is shown below,



Mathematically,

$$\begin{aligned} y(m) &= h(mM) \star x(mM) \\ &= \sum_{n=-\infty}^{\infty} h(n)x(mM-n). \end{aligned} \tag{B.1}$$

We can reduce the computational bandwidth of this structure by decimating *before* filtering (i.e. transposing the two blocks in the figure). Consider breaking the sum in (B.1) into two variables r and ρ such that $n = rM + \rho$. Then

$$\begin{aligned} y(m) &= \sum_{\rho=0}^{M-1} \sum_{r=-\infty}^{\infty} h(rM+\rho)x((m-r)M-\rho) \\ &= \sum_{\rho=0}^{M-1} \sum_{r=-\infty}^{\infty} \overline{p}_{\rho}(r)x_{\rho}(m-r) \\ &= \sum_{\rho=0}^{M-1} \overline{p}_{\rho}(m) \star x_{\rho}(m), \end{aligned} \tag{B.2}$$

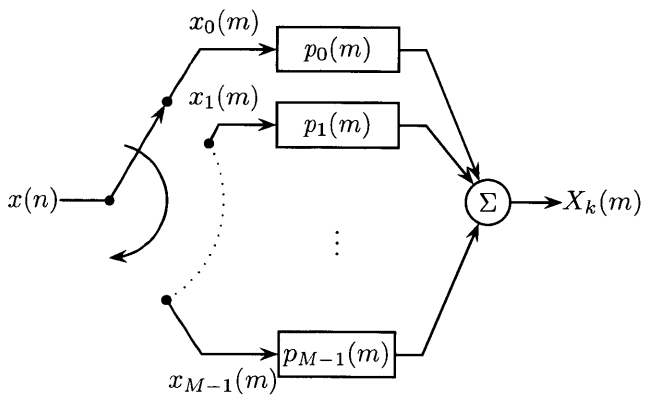


Figure B-1: The polyphase decimator. The $p_\rho(m)$ are the polyphase filters of $h(n)$. As the commutator rotates clockwise, $x(n)$ is downsampled at increasing phase factors to produce the $x_\rho(m)$.

where we have defined,

$$x_\rho(m) = x(mM - \rho), \qquad \overline{p}_\rho(m) = h(mM + \rho).$$

The $\overline{p}_\rho(m)$ are the *polyphase filters* of $h(n)$. They are downsampled versions of $h(n)$ taken at different offsets $\rho = 0, \ldots, M-1$. The $x_\rho(m)$ are the analogs for $x(n)$. Note that in (B.2), we are convolving downsampled versions of $h(n)$ and $x(n)$ at the same offset ρ. This structure can be simply realized by taking M different polyphase filters and connecting them to $x(n)$ via a commutator (a switch that "rotates" to connect a series of outputs to a single input in sequential order). This structure is shown in Figure B-1. As the commutator rotates, it picks out a different phase offset ρ, until it gets to the last polyphase filter, where it returns to the beginning (at which time the sum is computed).

If $h(n)$ has finite support L, then the polyphase filters $p_\rho(m)$ are of length $\left\lceil \frac{L-\rho}{M} \right\rceil$ and thus also have finite support. Furthermore, if L is a multiple of M, all of the polyphase filters have the same length.

Bibliography

[1] Ronald E. Crochiere and Lawrence R. Rabiner. *Multirate Digital Signal Processing.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[2] Ronald E. Crochiere and Lawrence R. Rabiner. *Multirate Digital Signal Processing.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[3] R. F. Emerson. Biplex Pipelined FFT. *Deep Space Network Progress Report*, 34:54–59, May 1976.

[4] Henning F. Harmuth. Applications of walsh functions in communications. *Spectrum, IEEE*, 6(11):82 –91, Nov. 1969.

[5] D. Larson et al. Seven-year Wilkinson microwave anisotropy probe (WMAP) observations: Power spectra and WMAP-derived parameters. `arXiv:1001.4635 [astro-ph.CO]`, 2010.

[6] A. Liu et al. Precision calibration of radio interferometers using redundant baselines. `arXiv:1001.5268 [astro-ph.IM]`, 2010.

[7] A. Lutomirski et al. Solving the corner-turning problem for large interferometers. `arXiv:0910.1351 [astro-ph.IM]`, 2009.

[8] M. McQuinn, O. Zahn, M. Zaldarriaga, L. Hernquist, and S. R. Furlanetto. Cosmological parameter estimation of reionization. *Astrophysical Journal*, 653:815–830, 2006.

[9] Alan V. Oppenheim and Alan S. Willsky. *Signals & Systems.* Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1996.

[10] M. R. Portnoff. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-24:243–248, June 1976.

[11] M. R. Portnoff. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-24:243–248, June 1976.

[12] Julius O. Smith. *Spectral Audio Signal Processing, March 2010 Draft.* `http://-ccrma.stanford.edu/~jos/sasp/`, accessed 4/29/2010. online book.

[13] G. Smoot et al. Structure in the COBE differential microwave radiometer first-year maps. *Astrophysical Journal*, 396(1):L1–L5, 1992.

[14] M. Tegmark and M. Zaldarriaga. Omniscopes: Large area telescope arrays with only $N \log N$ computational cost. `arXiv:0910.1351 [astro-ph.CO]`, 2009.

[15] M. Tegmark, M. Zaldarriaga, et al. Fast Fourier transform telescope. *Physical Review D*, 79, 2009.

[16] R. Thompson, J. Moran, and G. Swenson. *Interferometry and Synthesis in Radio Astronomy.* John Wiley and Sons, New York, 1986.

[17] Matias Zaldarriaga, Steven R. Furlanetto, and Lars Hernquist. 21 centimeter fluctuations from cosmic gas at high redshifts. *The Astrophysical Journal*, 608(2):622, 2004.