Overview of the HISPEC 8192 spectrometer firmware for the Parkes 64 m telescope

April 18, 2013

1 Introduction

In most modern radio telescopes, signals are digitized and processed in the digital domain. The bandwidth of the analogue to digital converter, data transport bandwidths, and the processing power of the hardware, all limit the instantaneous bandwidth achievable. This memo gives a detailed overview of the HISPEC 8192-channel firmware that is in use for wide-band observations with HIPSR. Readers familiar with digital spectrometers and FPGAs can skip straight to section 5.

2 Digital spectrometer implementations

In radio astronomy, a spectrometer is a device that computes the power spectral density of a signal. The power spectral density of a waveform is related to its autocorrelation function by the Wiener-Khinchin theorem. The theorem states that the relationship between a stationary ergodic signal $x(t)$, its power spectrum $S_{xx}(\nu)$, and its autocorrelation $r_{xx}(\tau)$ is given by

$$S_{xx}(\nu) = \int_{-\infty}^{\infty} r_{xx}(\tau) e^{-2\pi i \nu \tau} d\tau. \tag{1}$$

That is, the autocorrelation

$$r_{xx}(\tau) = \langle x(t) x(t-\tau) \rangle \tag{2}$$

is related to power spectral density by a Fourier transform. In the discrete case, the relationship becomes

$$S_{xx}(k) = \sum_{k=-\infty}^{\infty} \langle x(n) x(n-k) \rangle \, e^{-2\pi i k \tau}, \tag{3}$$

which is a discrete convolution. It follows from the convolution theorem that

$$S_{xx}(k) = \left\langle |X(k)|^2 \right\rangle, \tag{4}$$

where $X(k)$ denotes the Discrete Fourier Transform (DFT) of $x(t)$:

$$X(k) = \sum_{n=0}^{N} x(n) e^{-2\pi i n k / N} \tag{5}$$

There are two distinct classes of digital spectrometers: ones that approximate $S_{xx}(k)$ through numerical approximation to Equation 3, and those that evaulate Equation 4. The first class are called lag-autocorrelators, or auto-correlation spectrometers (ACS). The second class are known as a discrete Fourier transform filterbanks (FTF).

ACS and FTF architectures differ in both their spectral response and the number of computations required to implement them. For regularly sampled data, the Fast Fourier Transform (FFT) algorithm may be used to compute the DFT (see, for example, Brigham, 1988), which reduces the number of computations required from $O(N^2)$ for to $O(Nlog(N))$. In Chapter 4 of Taylor et al. (1999), it is shown that the ratio of multiplies for the two architectures is

$$R_{\frac{ACS}{FTF}} = \frac{n_t}{2log_2(n_t)}, \tag{6}$$

where n_t is the number of samples per FFT or lag correlation. So in general, FTF architectures require fewer computations than their equivalent ACS counterpart. Romney also shows that the spectral response of FTF and ACS architectures differ, with ACS architectures having a *sinc* reponse, and FTF architectures posessing a $sinc^2$ response. The result is that interchannel isolation is better in FTF architectures (first sidelobe at ~ -13.6 dB), than in ACS architectures (~ -6.8 dB).

2.1 Windowing functions

The spectral response of an FTF spectrometer can be improved by windowing the data before performing the DFT; in ACS architectures a windowing function may be applied after the auto-correlation step. An in-depth discussion of windowing functions is given in Gade & Herlufsen (1987).

For filterbanks, the Hamming and Hann windows are commonly applied windowing functions. The Hamming window has coefficients given by

$$w(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right), \tag{7}$$

which are optimized to minimize the level of the first sidelobe. These coefficients are similar to those of the Hann window[1]:

$$w(n) = 0.5(1 - cos\left(\frac{2\pi n}{N-1}\right)), \tag{8}$$

which offers faster sidelobe rolloff, but a higher first-sidelobe level. These should be compared to the Dolph–Chebyshev window

$$w(n) = \frac{cos\{Ncos^{-1}[\beta cos(\frac{\pi k}{N})]\}}{cosh[Ncosh^{-1}(\beta)]}, \tag{9}$$

where $\beta = cosh(N^{-1}cosh^{-1}(10^{\alpha}))$, which sets the average sidelobe level to be -20α dB, but offers no sidelobe rolloff.

A more dramatic improvement may be achieved by using an lowpass filter frontend before the DFT (Bellanger et al., 1976), to form what is known as a polyphase filterbank (PFB). This approach is detailed further in the following section. Figure 1a compares the spectral response of a standard DFT ('vanilla'), Hamming windowed DFT, and polyphase filterbank, for a single channel. An example showing a 16-channel, 4-tap Hamming window-based polyphase filterbank is shown in Figure 1b.

2.2 Spectrometers for radio astronomy

The first digital spectrometer used for radio astronomy was developed by Weinreb (1963). This 1-bit ACS was used to observe the 18-cm wavelength hydroxyl (OH) absorption line in the spectrum of Cassiopeia A, providing the first evidence of OH in the interstellar medium (Weinreb et al., 1963). The first reference to FTF spectrometers for radio astronomy can be found in Chikada et al. (1987); however FTF spectrometers did not enjoy widespread adoption until much later. This is due to implementation issues, such as increased data output rates as compared to ACS implementations; see Bunton (2000), for a discussion of these issues.

[1] Named after Julius von Hann, but often erroneously referred to as 'Hanning'.

In comparison to both ACS and FTF architectures, polyphase filterbanks offer vastly improved interchannel isolation. For radio astronomy purposes, high interchannel isolation is important so that spectral features are not smeared out, and so the spectrometer is more resilient to the high levels of radio frequency interference (RFI) emitted from terrestrial sources. Polyphase filterbanks are therefore the best candidate for radio astronomy spectrometers, if computationally affordable.

3 Polyphase filterbanks

A filterbank is simply a collection of filters, a simple example being a highpass and lowpass filter pair. I discuss here filterbanks where each filter has identical passband characteristics and evenly spaced central frequencies, such as the 16 channel filterbank shown in Figure 1b. This equally spaced form is the most commonly implemented for radio astronomy applications.

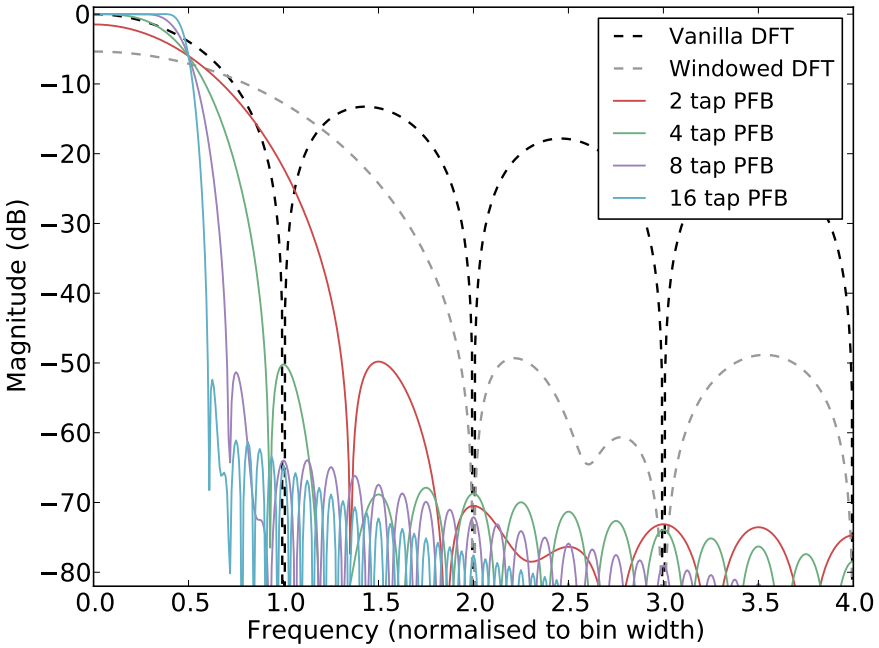
This section begins with a short overview of digital filtering techniques; I refer the reader unfamiliar with digital signal processing to the excellent Lyons (2004) and Smith (2002). A comprehensive overview of polyphase filters is given by Vaidyanathan (1990), and in Chapter 2 of Harris (2004); I will give a brief introduction here. In the diagrams that follow, the symbol $\otimes$ denotes multiplication of time samples, and $\oplus$ denotes addition. The symbol z^{-n} is used to denote a time delay of n units, and the symbol $\downarrow D$ is used for downsampling by a factor D and $\uparrow U$ for upsampling by a factor U.

3.1 Finite impulse response filters

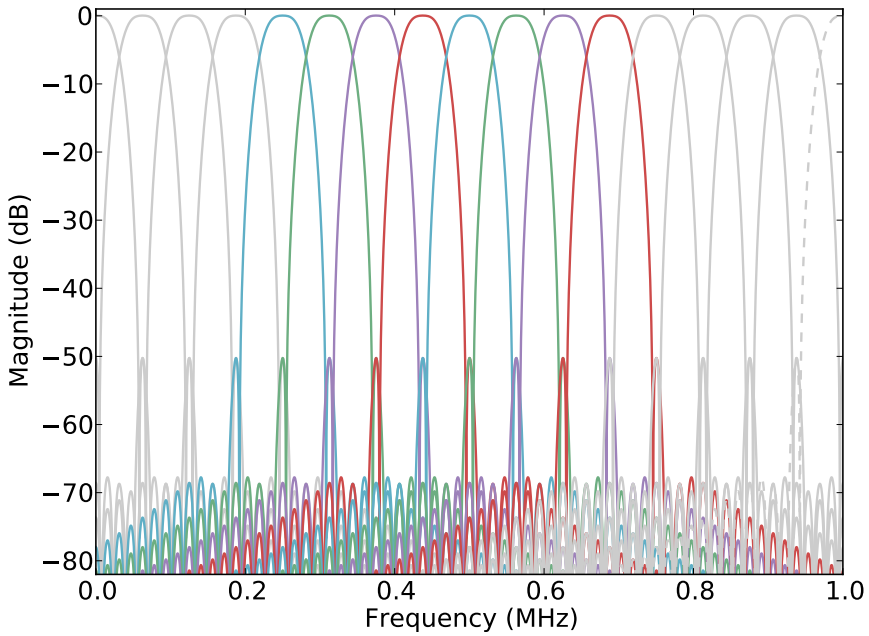
A finite impulse response (FIR) filter is the windowed moving average of an input sequence $x(t)$. An FIR filter computes the sum

$$y(t) = \sum_{k=0}^{K-1} h(k)x(t-k), \tag{10}$$

where $y(n)$ is the output sequence, and $h(k)$ is a set of K coefficients used for weighting. The upper summation bound, K, is called the number of taps. A streaming implementation of an FIR filter is shown in Figure 2.



(a) Affect of number of taps on the frequency response of a single polyphase filter bin. Hamming window coefficients are used in these data.



(b) A 4-tap, 16 channel polyphase filterbank, spanning a 1 MHz band. The middle eight channels are highlighted to indicate the overlap between channels.

Figure 1: Polyphase filter bank (PFB) spectrometer characteristics.

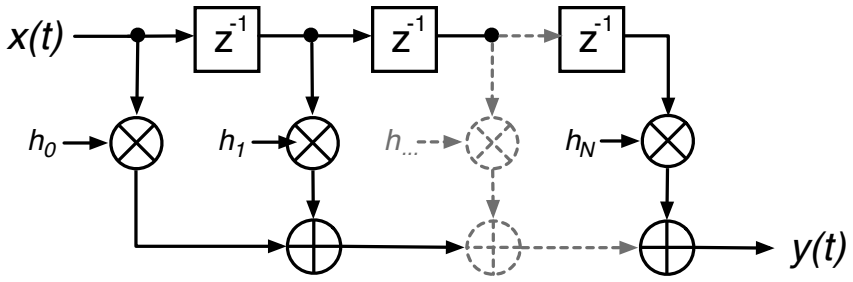


Figure 2: N-tap FIR filter block diagram. An FIR filter applies a weighted sum to the input sequence $x(t)$, outputting the filtered signal $y(t)$.

If we downsample after an FIR by $\downarrow D$, we only keep the outputs $t = mD$, $m \in \mathbb{Z}^+$. A $\downarrow D$ downsampled filter will alias spectra centred at any multiple of the output sample rate to baseband. In such cases it is more efficient to only compute the terms we wish to keep:

$$y(mD) = \sum_{k=0}^{K-1} h(k)x(mD - k). \tag{11}$$

One way we can accomplish this is to use a polyphase decimating filter, which is discussed below.

3.2 Polyphase FIR filters

A common technique in DSP is to decompose an input sequence $x(t)$ into a set

$$\mathbb{P} = \left\{ x_k(t) \quad k \in (0, P-1) \right\} \tag{12}$$

of P sub-sequences, $x_k(t)$, each of which is given by

$$x_k(t) = (\downarrow P)(z^{-k})x(t). \tag{13}$$

This is known as polyphase decomposition. Even and odd decomposition of the signal $x(t)$ is achieved when $P = 2$:

$$x_0(t) = \{x(0), x(2), x(4), ...\} \tag{14}$$

$$x_1(t) = \{x(1), x(3), x(5), ...\}. \tag{15}$$

More generally, an input stream may be decomposed into P different 'phases'.

Polyphase filter structures are often more efficient than standard finite impulse response filters when used in sample rate conversion. A $\downarrow P$ decimating FIR filter of length $K = MP$ can be constructed from P discrete FIR filter 'branches', each acting upon a different phase; that is

$$y(t) = \sum_{p=0}^{P-1} \sum_{m=0}^{M-1} h_p(m) x_p(t-m), \tag{16}$$

This is known as a decimating polyphase filter. For example, a $P = 4$ branch polyphase filter with $M = 7$ taps per sub-filter would compute the sum

$$\begin{aligned} y(t) &= \sum_{m=0}^{7} h_0(m) x_0(t-m) + \sum_{m=0}^{7} h_1(m) x_1(t-m) \\ &+ \sum_{m=0}^{7} h_2(m) x_2(t-m) + \sum_{m=0}^{7} h_3(m) x_3(t-m), \end{aligned} \tag{17}$$

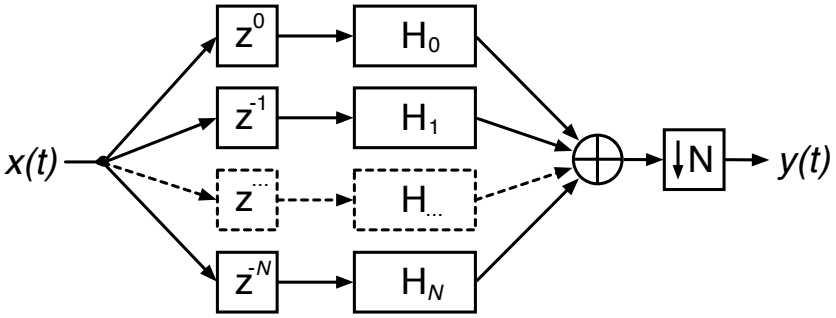
and has an output equivalent to a $4 \times 7 = 28$ tap standard FIR filter with 4:1 downsampling.

Decimating polyphase filter structures are more efficient than standard FIR filter based downsampling techniques. If $\downarrow D$ downsampling occurs after the moving average of Equation 10, we compute D sums, but keep only 1 in D of these. This is inefficient. In comparison, Equation 16 only computes the output values that are of interest. A comparison of two decimating filters is shown in Figure 3. Figure 3b is a polyphase type structure, but decimation occurs after summation. Conversely, Figure 3b shows a more efficient implentation where decimation occurs before summation. By the noble identities (see Vaidyanathan, 1990), the output of this second filter is identical to that in Figure 3b.

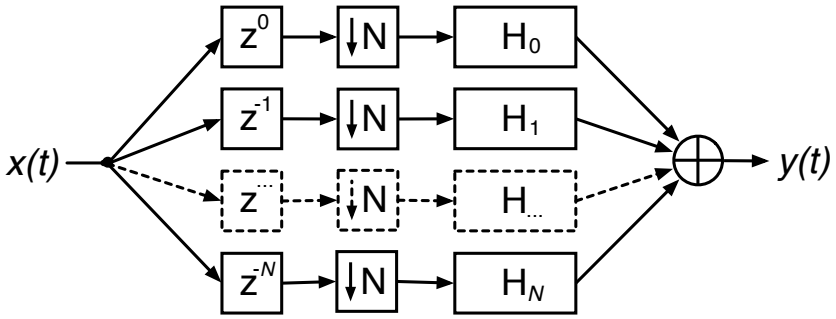
3.3 DFT BASED POLYPHASE FILTERBANKS

A computationally efficient filterbank with high interchannel isolation can be constructed from an FFT preceded by a prototype polyphase FIR filter frontend. Such an implementation exploits the fact that a lowpass filter with coefficients $h(k)$, can be converted into a quadrature (complex) bandpass filter with central frequency ω_k by multiplying the coefficients by $e^{i\omega_k t}$. That is,

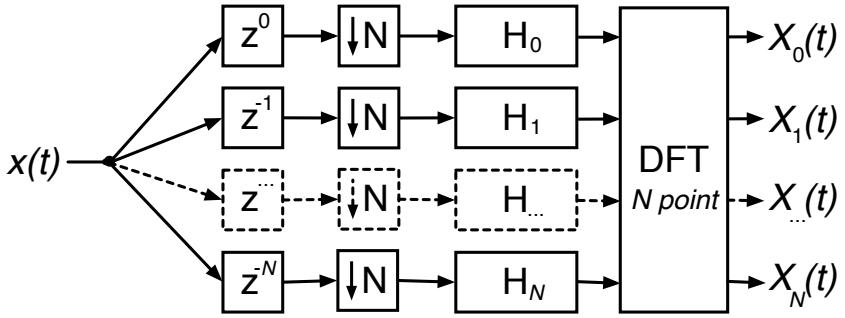
$$h_{bpf}(k) = h(k) e^{i\omega_k t}. \tag{18}$$



(a) An inefficient implementation of a decimating filter.



(b) N:1 decimating polyphase filter.



(c) N-channel polyphase filter bank.

Figure 3: Development of a polyphase filterbank from a prototype filter (a). In (b), the down-sampling step is moved using the noble identities for filters. In (c), the adder is replaced with an N- point DFT, creating a filterbank.

Now, suppose we have a decimating lowpass polyphase filter, such as that in Figure 3b. The output of each branch is

$$y_p(t) = \sum_{m=0}^{M-1} h_p(m) x_p(t-m), \tag{19}$$

where $h_p(m)$ are coefficients from our prototype lowpass filter. If we follow this by a DFT, as in Figure 3c, we then have

$$\begin{aligned} Y(k) &= \sum_{p=0}^{P-1} y_p(t) e^{-2\pi i k p/P} && (20) \\ &= \sum_{p=0}^{P-1} \sum_{m=0}^{M-1} h_p(m) e^{-2\pi i k p/P} x_p(t-m). && (21) \end{aligned}$$

Comparing this form to Equation 11 and Equation 18, we recognise that the output of this structure — Figure 3c — is equivalent to a set of $\downarrow P$ downsampling polyphase filters:

$$\mathbb{F} = \Big\{ h_k(m), \quad k \in (0, P-1) \Big\} \tag{22}$$

where the central frequency of each filter is shifted by an amount $2\pi i k/P$. For data sampled at the Nyquist rate f_s, this filterbank consists of N filters spanning $-f_s/2$ to $f_s/2$, with each filter separated by $f_s/2N$. For real sampled data the negative frequencies contain no extra information, so they need not be computed. This leaves a a bank of $N/2$ filters, evenly spaced over a bandwidth $f_s/2$. This is the structure implemented in the HISPEC spectrometer detailed in this memo.

This polyphase filterbank (PFB) structure was first proposed by Bellanger et al. (1976), and has been previously used in radio astronomy applications, see for example Zimmerman & Gulkis (1991) and Parsons et al. (2006). The rest of this memo discusses the implementation of a 8192 channel polyphase filterbank for radio astronomy, on Field Programmable Gate Array based digital hardware.

4 Field programmable gate arrays

A Field Programmable Gate Array (FPGA), is a reconfigurable integrated circuit that is used for digital signal processing. FPGAs consist of programmable logic components and memory elements with reconfigurable interconnections. Complex signal processing structures, such as dig-

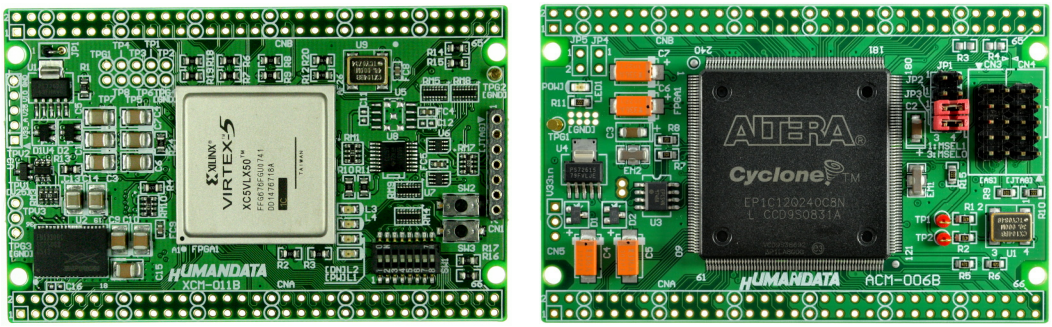


Figure 4: Example FPGA prototyping boards. The Xilinx Virtex-5 FPGA is shown on the left, and an Altera Cyclone series FPGA is shown on the right.

ital filters and FFTs, can be made by interconnecting these multiple logic components. Some example FPGA boards are shown in Figure 4.

FPGAs have several advantages over CPUs and GPUs in certain applications. Firstly, they are more energy efficient, in some cases orders of magnitude so. For example, Kestur et al. (2010) report that FPGAs achieve between 2.7 to 293 times better energy efficiency for certain basic linear algebra subroutines (BLAS) than CPU and GPU based BLAS implementations. In addition, FPGAs are better suited to high bandwidth applications, as the I/O (input/output) bandwidth on CPUs and GPUs are limited by the speed of their host's PCIe bus, whereas FPGA bandwidth is limited by the number and speed of I/O pins upon the chip. FPGAs are not without their disadvantages: they are significantly harder to program, as algorithms must be mapped into a physical circuit within the FPGA's fabric. Each FPGA has a limited amount of logic and memory available, so algorithms which require large amounts of data to be stored in memory are hard to implement. The cost of the FPGA chip and FPGA design software is also a factor which may limit their applicability.

For cases where energy efficiency is the prime concern, FPGAs are outclassed by application specific integrated circuits (ASICs). ASICs are chips designed for one particular use; they trade off the flexibility of reconfigurability for smaller footprints and energy consumption. Kuon & Rose (2006) report that an FPGA is on average 40 times larger and 3.2 times slower than an ASIC implementation of logic-only circuits, and that FPGAs consume on average 12 times more dynamic power than equivalent ASICs. The disadvantages of ASICs are their lack of reconfigurability; their significant fabrication cost (of order US$1M); and, longer lead times due to their bespoke nature.

If energy efficiency and bandwidth are not limiting factors, and a desired algorithm can be

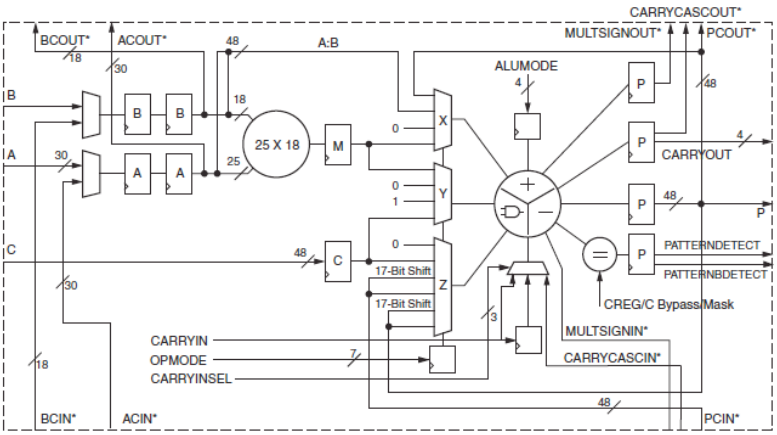


Figure 5: Schematic of the Xilinx DSP48E slice, from DPS48E usage guide (Xilinx, 2012).

performed on a CPU or GPU, then it should be implemented on one. This is as the development time is invariably lower, the code is easier to read, and debugging is far easier. The choice between FPGAs, CPUs, GPUs, and ASICs is therefore not a simple one: it is highly dependent on the application, the timescale, and how much money one has to spend.

5 DSP HARDWARE FOR RADIO ASTRONOMY

For high bandwidth radio astronomy applications, FPGAs are an attractive technology, and have received much attention in recent years. See for example the Altera-powered Uniboard project (Szomoru, 2010), and the Xilinx-powered CASPER iBOB, BEE2 and ROACH FPGA boards (Parsons et al., 2006). A recent addition to the CASPER family is the ROACH-2, a Virtex-6 based board. The spectrometer design detailed in this memo is designed to run upon the CASPER ROACH board, which is introduced below. Central to the ROACH is the Virtex-5 FPGA.

5.1 VIRTEX-5 FPGA

The Virtex-5 FPGA is a high perfomance FPGA designed by Xilinx Incorporated[2]. It is constructed using a 65 nm CMOS manufacturing process. Since my DPhil. began in 2009, Xilinx have released the Virtex-6 and Virtex-7 FPGAs, which are based respectively on 40 nm and 28 nm manufacturing processes.

The basic logic elements for Xilinx FPGAs are Configurable Logic Blocks (CLBs). These provide

[2] http://www.xilinx.com/support/documentation/virtex-5.htm

combinatorial and synchronous logic as well as distributed memory and shift registers. In addition to CLBs, each Virtex-5 contains block RAMs (BRAMs); these RAM modules are arranged into columns on the chip fabric and are 36 kb each.

Much of the DSP versatility of the Virtex-5 comes from elements called DSP48E slices. These slices consist of a 25×18 bit multiplier followed by data routing multiplexers. This mulitplex is followed by an element which can be used as an adder or two-input logic unit. A circuit schematic of the DSP48E is shown in Figure 5.

Peak performance of the Virtex-5 is achieved when every slice is being used constructively every clock cycle. In reality, this level of utilization is never achieved; nevertheless, if one managed to utilize all 640 DSP48E slices of a SX95T Virtex-5 at the maximum clock rate of 550 MHz, 352 billion multiply accumulate operations each second (MACs) could be performed. Hickish et al. (2012) shows that each DSP48Es can be used to compute several 4-bit multiplications per clock cycle, so perfomance up to 1TMAC is theoretically achievable on one Virtex-5 FPGA.

5.2 The CASPER ROACH board

The FPGA boards produced by the Collaboration for Astronomical Signal Processing and Electronic Research (CASPER) are based around the Xilinx Virtex series of FPGA. The ROACH, or Reconfigurable Open Architecture Computing Hardware board, is a SX95T Virtex-5 FPGA based signal processing board designed for radio astronomy applications. A simplified block diagram of the board is shown in Figure 6.

Input/Output interfaces Four CX4 connectors allow up to 40Gb/s of bandwidth for data transfer over 10GbE or XAUI protocols. Daughter boards, such as analogue to digital converters, may be connected via the two Z-DOK interfaces, each with 40 differential pairs which connect to the FPGA I/O pins. High speed data transfer may also be done through the Q-Strip® connector (QSH). General Purpose Input/Output (GPIO) pins are also provided for control of miscellaneous sensors and external devices. For temporary storage of large vectors, the FPGA is connected to quad data rate random access memory (QDRII+ RAM) and double data rate memory (DDR2).

PowerPC The Virtex-5 FPGA is connected to a PowerPC chip by an On-Chip Perhipheral Bus (OPB) controller. The PowerPC runs a customized Linux kernel and provides command and control for the board. It has its own dedicated DDR2 and Flash memory, 1Gb Ethernet port, USB

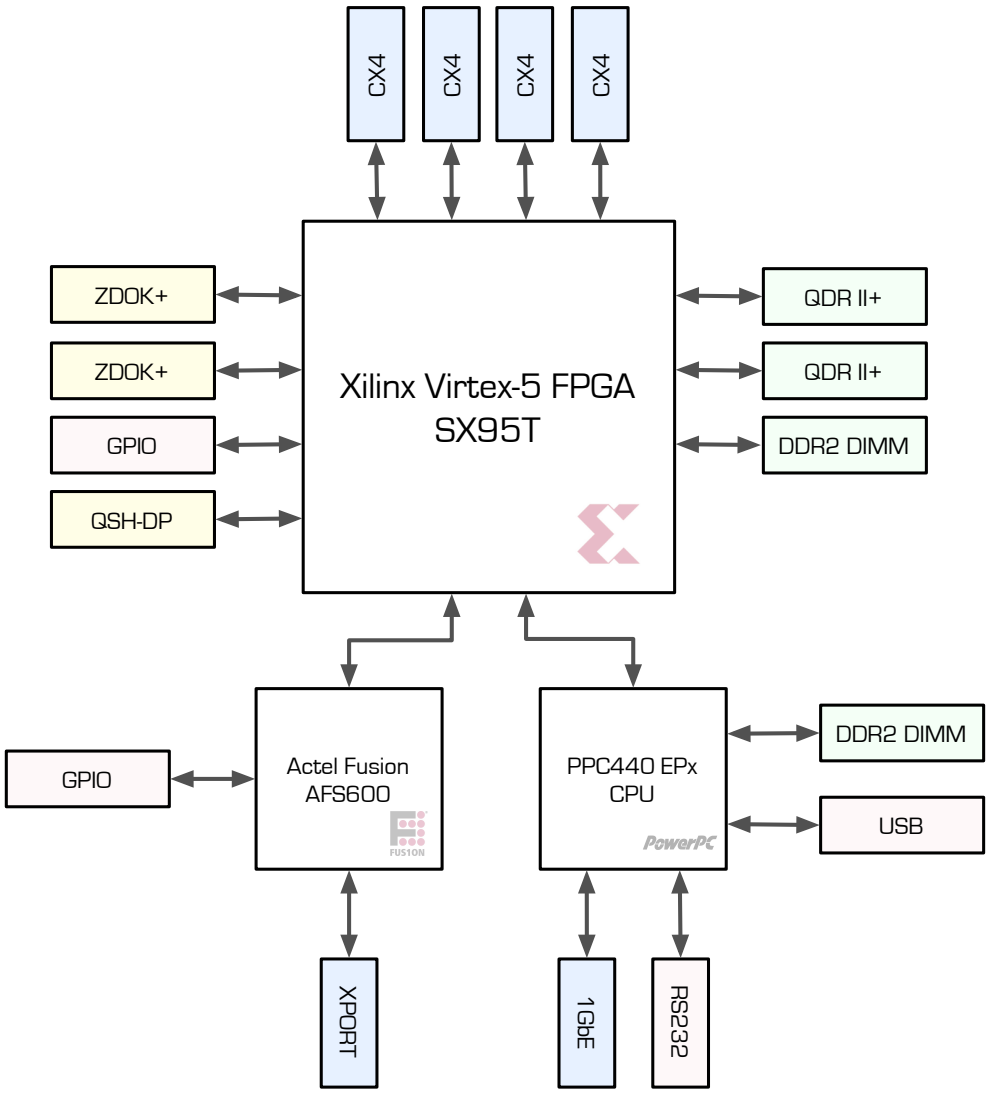


Figure 6: Block diagram of the CASPER ROACH board, showing input/output interfaces and peripheral memories.

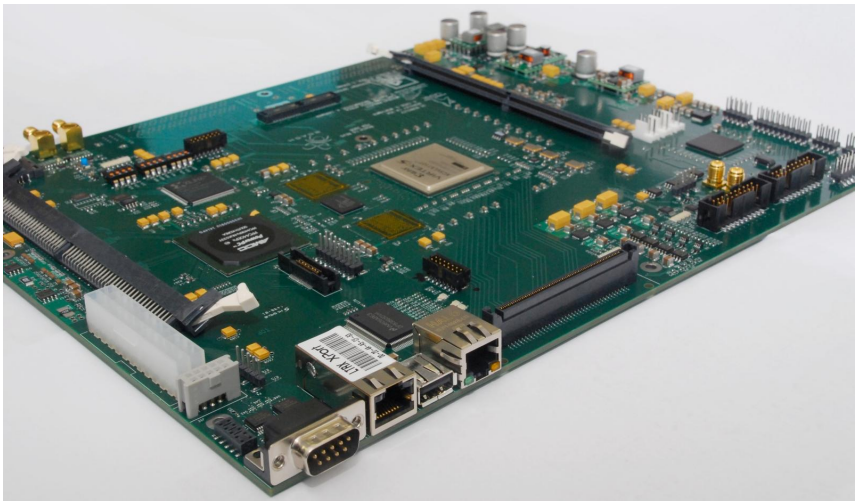


Figure 7: The CASPER ROACH board, revision 1.00. Image credit: CASPER collaboration

port, and SD memory card socket.

ACTEL FUSION An Actel Fusion FPGA controls the ATX power supply's soft power toggling and monitors the voltage and current draw of key power supplies. It also monitors fan speeds and component temperatures. Remote access to the Actel FPGA is enabled by an XPORT Ethernet adaptor, which encapsulates serial I/O with the Actel into an Ethernet format.

5.3 iADC DIGITIZER CARD

The iADC, or ADC2x1000-8, is an 8-bit, 2-input analogue to digital converter (ADC). The iADC is a Eurocard-sized daughter board which connects to the ROACH via the Z-DOK connector. Its centrepiece is an Atmel/e2V AT84AD001B digitizer chip, which runs at speeds of up to 1Gsample/sec. Inputs are connected to the iADC's SMA jacks; a MiniCircuits ADTL2-18 balun (30 MHz-1.8 GHz bandwidth) converts the single-ended transmission line to differential, as is required by the digitizer chip. When in operation, an external synthesized clock signal must be presented as a clock source. A pulse per second (1PPS) may also be connected; this signal is passed through to the FPGA so multiple ROACH boards may be synchronized to within a clock cycle. Using this ADC, two signals with a bandwidth of 512 MHz may be digitized. Alternatively, the AT84AD001B offers an interleave mode which sacrifices one input but is then able to digitize 1024 MHz from a single input.

Two important characteristics for spectrometers are crosstalk levels between signals, and the

Table 1: Atmel AT84AD001B performance characteristics (Atmel, 2009).

Sampling rate	1Gsample/sec, 2GS/s interleaved
Number of bits	8
Effective number of bits (ENOB)	6.8
Differential analogue input voltage (full scale)	500mVpp
Spurious free dynamic range	-54 dBc
Differential non-linearity	0.25 LSB
Integral non-linearity	0.5 LSB
Bit error rate	10^{-13} at 1GSample/sec

linearity of the digitizer's response. Crosstalk of the iADC has been measured by Parsons (2009) to be under 28 dB, and was found that $\sim$ 60 dB of isolation can be achieved by subtracting the stable crosstalk component. Linearity was tested by Lipsey (2012, personal communication), and found to be in accordance with the manufacturer's quoted specification (differential 0.25 LSB, integral 0.5 LSB). Further specifics are listed in Table 1.

5.4 The Xilinx toolflow

All the FPGA firmware detailed in this thesis was designed using the Xilinx ISE toolflow[3]. The toolflow has been integrated into MATLAB Simulink[4], which is a high level graphical programming environment.

The CASPER collaboration provide Simulink based component libraries designed specifically for radio astronomy instrumentation. These libraries include implementations of DSP structures such as FFTs, FIR filters, vector accumulators, digital mixers and complex multipliers. These components are parameterized, so that their specifications can be changed by the user.

The CASPER libraries are built upon the Xilinx System Generator blockset for Simulink. Each Xilinx block has a set of configurable parameters, input ports and output ports. Simulink allows systems to be developed by connecting blocks together, and provides a simulation environment for analysing signal properties. Once a model has been constructed, an HDL (hardware description language) netlist can be compiled through Xilinx System Generator. From this point, the Xilinx toolflow can be used to synthesize and implement FPGA specific firmware.

The Xilinx toolflow maps Simulink designs into physical circuits upon the FPGA's fabric, using a heuristic place-and-route algorithm. This process can take several hours for designs with

[3] http://www.xilinx.com/products/design-tools/ise-design-suite/
[4] http://www.mathworks.com.au/products/simulink/

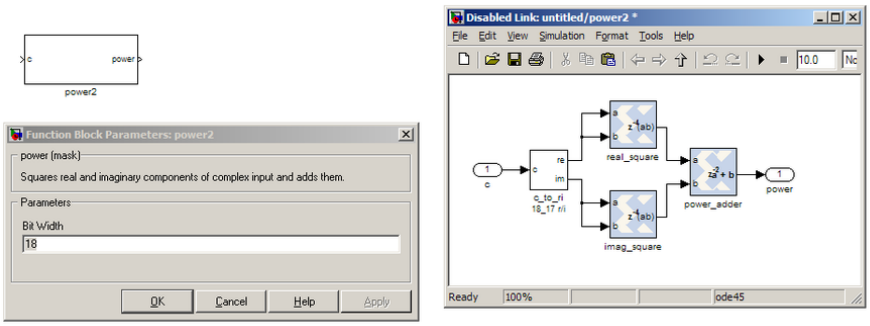


Figure 8: Example of a CASPER simulink block. This particular block computes the power of a complex valued input signal.

high resource utilization (using a computer with a 2.66 GHz Intel Core-2 Duo CPU, 8 GB RAM). After the design is placed and routed, a bitstream is then generated with which to program the FPGA. Loading a pre-compiled bitstream onto an FPGA takes less than a second, hence FPGAs are considered 'field programmable'.

6 THE HISPEC SPECTROMETER

In this section, I introduce an FPGA based wide-bandwidth spectrometer called HISPEC 8192, designed for redshifted HI line spectroscopy. This spectrometer is used in the HIPSR project. HISPEC is based on an 8192-channel, 4-tap polyphase filterbank. It digitizes both inputs from a dual-polarization antenna and applies a PFB to each input before computing all four Stokes parameters and accumulating the signal. The spectrometer can digitize and process up to 512 MHz of bandwidth instantaneously.

The HISPEC spectrometer runs on an SX95T Virtex-5 FPGA equipped ROACH board. I designed the FPGA gateware using the Xilinx ISE toolflow, and the MATLAB Simulink block libraries provided by the CASPER collaboration (Parsons et al., 2006); in addition, I developed new blocks whenever the CASPER provided blocks were unsuitable or simply did not exist.

6.1 FIRMWARE OVERVIEW

A block diagram of the HISPEC firmware is shown in Figure 9. Only the top-level blocks are shown; each of these blocks is created from Xilinx primitives. The data can be visualized as flowing from down the arrows and through the processing blocks, from left to right. In the

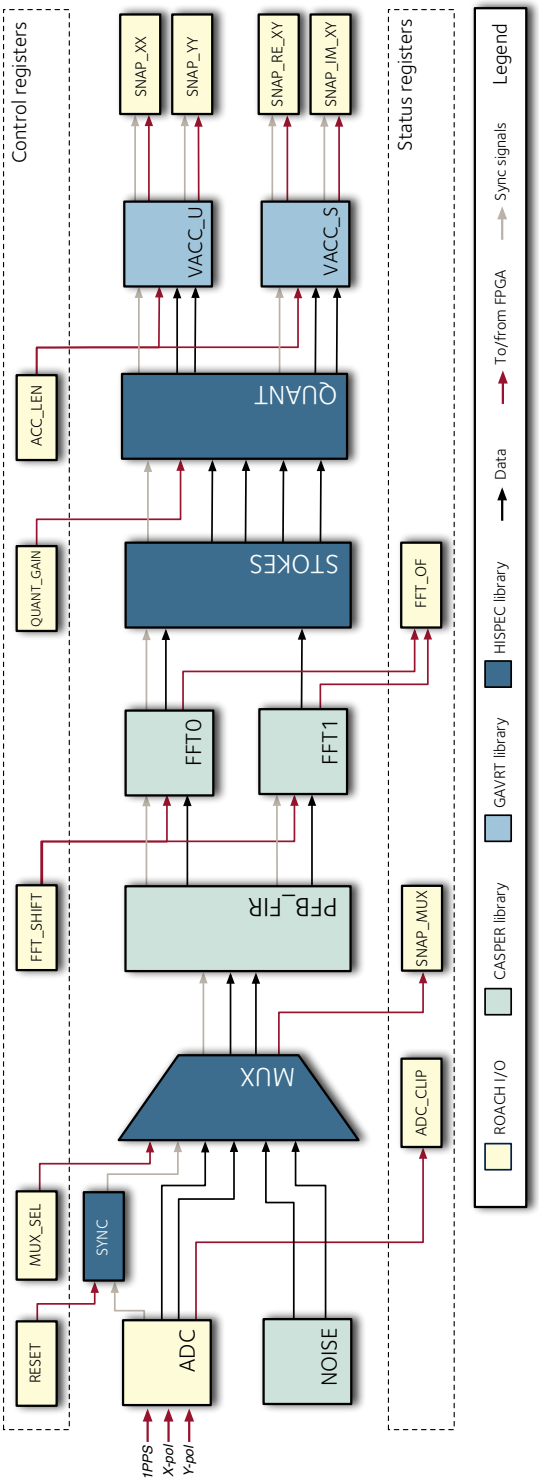


Figure 9: Simulink diagram of the HISPEC spectrometer. Only the top-level blocks are shown. The control and status registers are software accessible for real-time configuration via the PowerPC. Spectral data is stored in the shared BRAMs snap_xx, snap_yy, snap_re_xy and snap_im_xy, which is also software accessible.

diagram, all Input/Output interfaces are shown in light yellow. In the firmware description that follows, I will use the notation *a_b* to refer to *a* bit numbers with a decimal point preceding bit *b*. For example, 8_7 is an 8-bit signal, with 7 fractional bits. All arithmetic in the FPGA is fixed point, and is represented in 2's complement form when signed. From left to right, Figure 9 shows:

- Many of the CASPER library blocks use a vector warning system to initialize, synchronize and reset blocks. The warning signal is a single boolean 'pulse' of one cycle. Once generated, this propagates through the design, and care is taken to ensure that the first valid datum arrives at each block one clock cycle after the warning. This is commonly referred to as a 'sync pulse'. In the HISPEC design, this pulse is generated by the *sync* block.

- The first block, *adc*, links the iADC digitizer to the Simulink model as an external HDL defined peripheral. The *adc* block presents four time samples from the iADC in parallel per FPGA clock cycle. These samples are fixed point, 8_7 bit signed 2's complement numbers in the range [-1,1). For diagnostics, a digital noise source *noise* of identical bitwidth can be used as a test vector for downstream blocks. The selection between *adc* and *noise* is achieved using the multiplexer block *mux*, which is controlled by the register *mux_sel*.

- The signals pass through the 4-tap, 16384-branch, Hamming-windowed *pfb_fir* polyphase filterbank frontend. Each filter coefficient 18_17 bits, and the number of bits representing the signal grows from 8_7 to 18_17 bits through this filter. The *pfb_fir* is provided by the CASPER toolflow.

- After this, a 16384 point Radix-2 biplex pipelined FFT is applied, $fft0$ and $fft1$, to each signal. 18-bit fixed point arithmetic is used throughout, with the data cast back down to 18-bit after each multiply. The output is 18-bit real, 18-bit imaginary, but is represented as a 36_0 quantity. This block is a core component of the CASPER library and has been optimized by many members of the collaboration.

- The *stokes* block then computes auto- and cross-correlations for the two signals, from which Stokes parameters can be determined. Denoting the two input signals as X and Y, the required correlations are the real-valued XX^* and YY^*, and the complex-valued XY^*. As the FFT uses 18-bit complex fixed point arithmetic, the dynamic range is $10log_{10}(2^{2\times18})$ =108.37 dB. The heart of the *stokes* block is a DSP48E slice based complex multiplier.

The multiplication yields 36_34 bit numbers. The autocorrelations are computed using unsigned arithmetic, and the cross correlations with signed.

- The *quant* block converts the signal down to 18_0 bits in preparation for vector accumulation. Bit selection is achieved by multiplication with a 16_0 bit gain term, growing the signal to 52_0 bits, from which a 18_0 bit slice is taken. As the data output from the *stokes* block is above 25 bit, this multiply is conducted using distributed logic, not DSP48E slices.

- The *vacc_s* and *vacc_u* blocks (signed for cross correlations, unsigned for autocorrelations), are vector accumulators with 32_0 bit output which uses BRAMs for vector storage. Data are accumulated by feeding the output of the BRAM back into a preceding adder with an appropriate time delay. Up to 2^{16} accumulations cycles can be run before there is risk of signal overflow. These blocks are from the GAVRT blockset, which is a sub-library supplied by CASPER, originally designed for the Goldstone Apple Valley radio telescope.

- Finally, the four *snap* blocks on the right hand side of the diagram store the signal in BRAM, which is shared between the FPGA and Power PC.

Along with these signal processing blocks are a set of control registers, which allow the user to set the accumulation length, FFT shift schedule (which determines at which stages the signal is requantized), and to select between the digital noise source and the iADC. These are read-writable from the PowerPC, and read-only from the FPGA. A number of status registers are also included; these are read-only by the PowerPC and read-writable by the FPGA. The status registers are used for monitoring the data quality: boolean flags are raised for ADC clipping and signal overflows, and a count of successful accumulations is kept. A number of diagnostic shared BRAMs are interspersed throughout the design for debugging and signal monitoring.

Several of the pins of the FPGA are connected to a bank of GPIO (general purpose input/output) headers. Eight of these headers are connected to a panel of diagnostic LEDs. In the HISPEC design, boolean flag is raised whenever ADC clipping and signal overflow occurs. This flag is passed to the GPIO pins which creates a voltage and causes the LEDs to flash. The 1PPS signal and the accumulation count also trigger an LED to flash, akin to a 'heartbeat' expressing board health.

Table 2: HISPEC resource utilization, as a fraction of total resources available on the Virtex 5 FPGA.

BRAMs	202/244	82%
DSP48Es	222/640	34%
Slice logic	11203/14720	76%

6.2 Firmware compilation

One of the challenges of FPGA firmware design is ensuring an implementation uses less circuit logic than the amount available on the FPGA fabric. More specifically, the number of nets in the HDL netlist may not exceed the number of available components on the FPGA chip. To fit dual 16384-point FFTs onto the Virtex-5 required balancing BRAM and distributed memory usage. The CASPER FFT block is parameterized, so that coefficients and delays can be stored in distributed memory if they are under a given size; this decreases the BRAM usage at the expense of slices and routing complexity. To find a suitable balance, I compiled the FFT multiple times with different memory storage parameters until I found an appropriate implementation.

Along with such physical constraints are timing constraints: the output of each logic element must reach the input of the next element within a clock cycle. If the Xilinx place-and-route tool cannot find a solution in which all timing constraints are satisfied, the implementation is said to 'not meet timing'. As the FPGA clock speed increases, the period of time available to route a signal decreases, and as such the likelihood of meeting timing also decreases.

To digitize 512 MHz of bandwidth, the FPGA must be clocked at 256 MHz. While the HISPEC design compiles at clock speeds up to 200 MHz, attempting to compile the design at 256 MHz fails. To find which nets were the most troublesome to route, I analysed the HISPEC design using the Xilinx Planahead software package[5]. Among the features of Planahead is a floor-planning tool, which highlights which nets are not meeting constraints. Once identified, issues can then be removed by modifying the design, or by asserting placement constraints.

The majority of timing issues for HISPEC arose from the *pfb_fir* frontend. In the *pfb_fir* block, filter coefficients are stored in BRAM. Routing the coefficients from the BRAM to the filter taps (which use DSP48E slices as multipliers), required routing over path distances that were unacheivable at the high clock speed. The solution I implemented was to force the coefficients to be routed through an intermediate register, which required modification of the *pfb_fir*. These

[5] http://www.xilinx.com/tools/planahead.htm

modifications increase the overall latency of the block without altering the overall implementation.

An alternative strategy to meeting timing at high clock speeds is to place constraints on the position of nets. While I experimented with this approach, solving the underlying timing issues within the *pfb_fir* by adding latency proved adequate and constrained placement was ultimately not required. Figure 10 shows the final mapping of the HISPEC spectrometer into the FPGA's logic blocks.

Table 2 shows the resource utilization of the final design. The BRAM usage of both the filter frontend, FFT and vector accumulator grow in proportion to the number of channels; as such, a design similar in implementation but with a larger number of channels would not fit into the FPGA fabric, as too many BRAMs would be required. Adding additional taps to the filter frontend also increases BRAM usage.

In order to build larger full Stokes spectrometers on a single FPGA, one would have to optimize their designs to use fewer BRAMs, or use a different FPGA with a larger number BRAMs. This design could be ported to the ROACH-2, which has a Xilinx Virtex-6 XC6VSX475T FPGA as its centrepiece. This chip has 1064 BRAMs[6], so one would expect that the number of channels could be quadrupled to 2^{16} without major changes to the design. The recently released Virtex-7 series sports up to 1880 BRAMs and 3360 DPS48Es[7], and this upward trend will continue in future FPGAs.

7 Results

To measure the response of the spectrometer to a reference signal, I connected a TTi TGR2050 synthesized RF signal generator to the spectrometer's inputs. This signal generator operates over 150 kHz to 2000 MHz, with a setting resolution of 10 Hz. The TG2050 has an output power of −127 to +7 dBm, with $\pm$ 2 dBm accuracy. Harmonically related signals are specified to <−25 dBc (decibels relative to the carrier). Further specifications can be found in TTi (2006).

In the paragraphs that follow, the response of the spectrometer to a single tone, the spectral isolation between channels, and the noise floor of the spectrometer are discussed.

[6] http://www.xilinx.com/support/documentation/virtex-6.htm
[7] http://www.xilinx.com/products/silicon-devices/fpga/virtex-7/index.htm

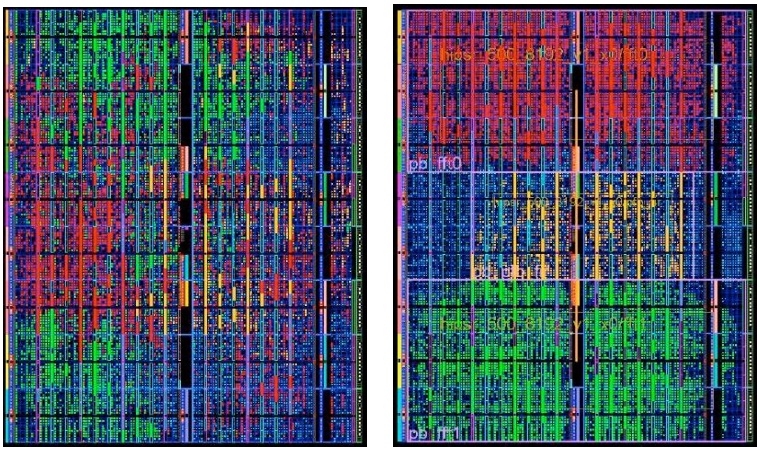


Figure 10: Xilinx Planahead circuit diagram of HIPSR spectrometer compiled at a clock speed of 256MHz, before floor planning (left), and after (right). Components associated with the PFB FIR frontend are highlighted in gold, and the two FFTs in red and green.

SINGLE TONE RESPONSE The response of the digitizer to a 1 MHz sine wave is shown in Figure 11. These 8-bit data were extracted from a shared BRAM on the FPGA fabric, which captures 4096 FPGA clock cycles worth of data. This is useful for debugging and ensuring the power level of the input signal is suitable. The filterbank's response to a -30 dBm, 10 MHz signal is shown in Figure 12. The first harmonic is $\sim$12 bits below the carrier, or -36.1 dBc — well below the manufacturer quoted <-25 dBc.

SPECTROMETER NOISE FLOOR Figure 13 shows the response of the spectrometer with no input signal; i.e. with terminators on the signal inputs. In the figure, the lowest 18 bits of the 36-bit autocorrelations were selected, and an accumulation length of 2^{14} clock cycles was used. There is a clear structure apparent in the data, particularly around multiples of 25 MHz.This structure arises to the imperfect nature of the digitizer: the sample and hold process introduces noise, distortion and aperture jitter.

There are also some self-induced interference sources apparent in the spectrum of Figure 13. At 133 MHz the PCI bus clock is visible, and at 4096 (200 MHz), the derived FPGA clock can be seen. The DDR2 clock can also be seen, at 333 MHz. The non-linear response of this noise floor can be mitigated by setting the gain coefficients of the *quant* block such that the least significant bit is not selected. A requisite for this is that the RMS power of the digitized signal is much greater than the 1-bit toggle level.

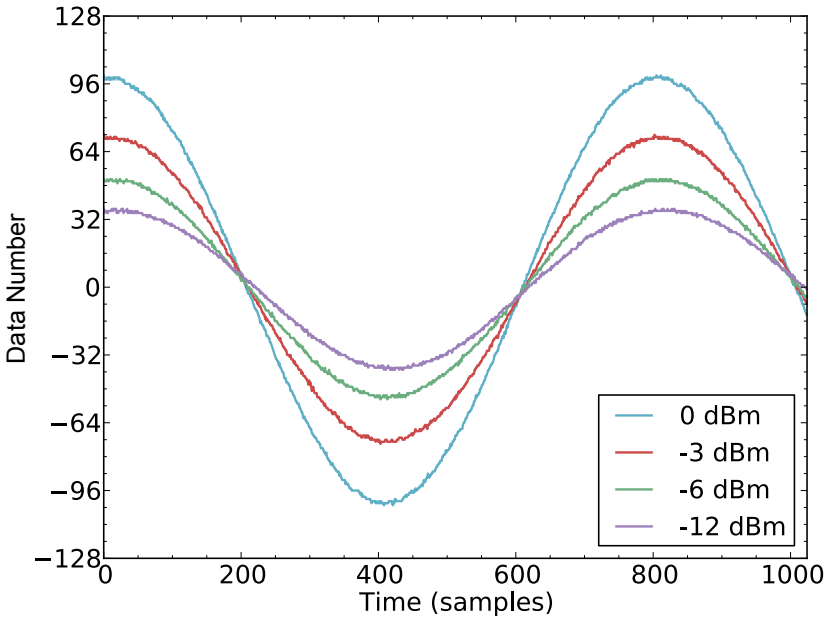


Figure 11: iADC response for a 1 MHz sine wave. The dBm power value neglects transmission loss of ∼3 dB over the coaxial cable's length

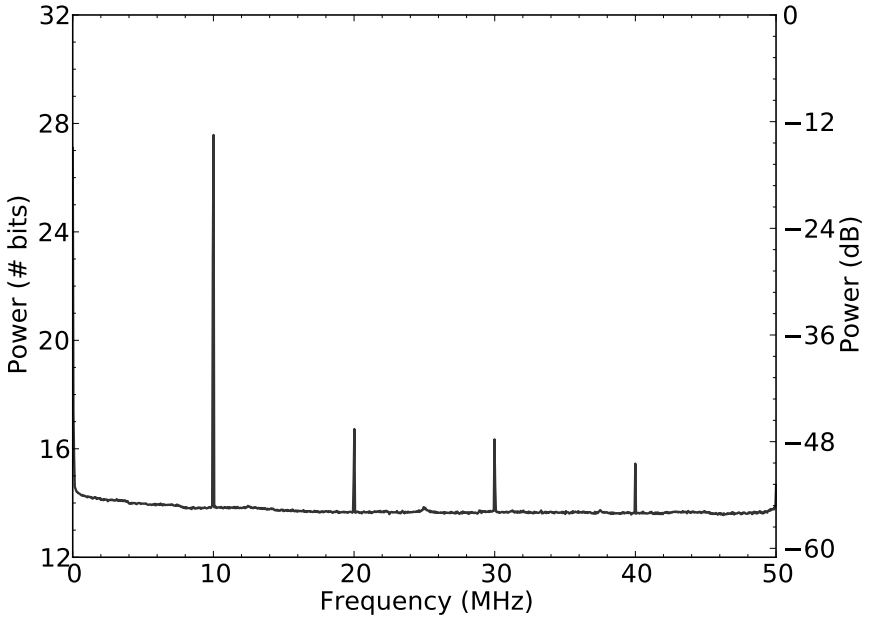


Figure 12: Spectrometer response to a 10 MHz tone. The harmonics of the signal can be seen at a lower level; the first harmonic is -36.1 dBc. Power is expressed on a log_2 based scale (equivalent to number of bits) on the left axis, and as power in decibels on the right axis ($10log_{10}$ values).

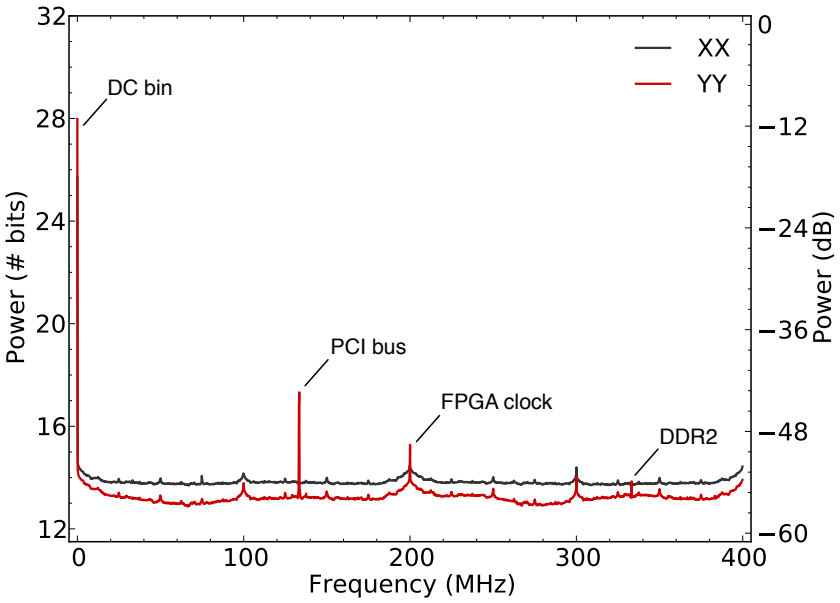


Figure 13: Noise floor of the spectrometer (integration time of 2 s). The reference clocks of the PCI bus, FPGA, and DDR2 memory can be seen in the data. The autocorrelation XX^* is shown in black, and YY^* is in red.

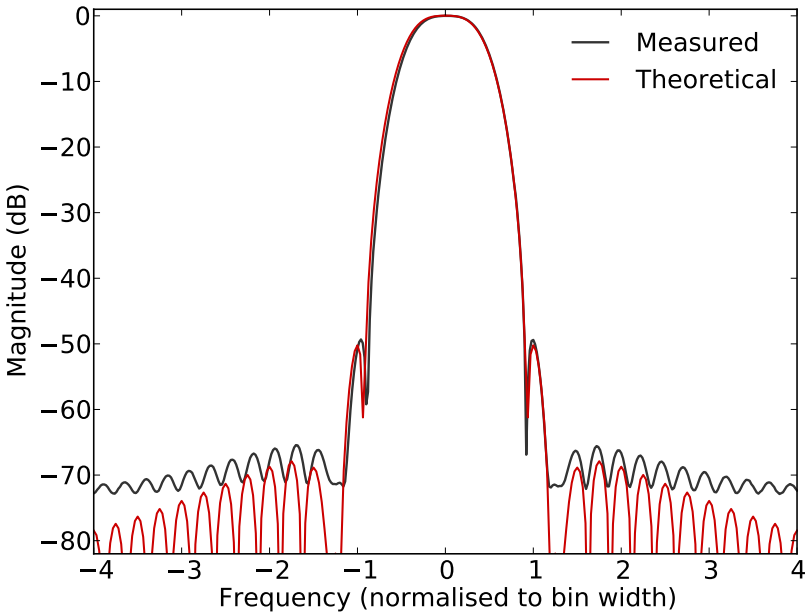


Figure 14: Measured response of a single polyphase filterbank channel of HISPEC, compared to the theoretical response of a 32768-tap Hamming windowed FIR filter with 32-bit floating point coefficients.

SPECTRAL ISOLATION In order to probe the response of a single polyphase filterbank channel, I measured the response of a single channel as the input frequency was scanned. So that high frequency resolution could be achieved, data capture was automated with a Python script that controls the signal generator via GPIB and collects data from the spectrometer via Ethernet. An input signal power of -15 dBm was used to measure the sidelobes, and decreased to -30 dBm to measure the main lobe. The measured filter response is shown in Figure 14, along with the response of an equivalent 32768-tap Hamming windowed filter. As can be seen, the responses match well over the main lobe and first sidelobe, but deviate after this. This is due to rounding errors within the *pfb_fir* and FFT, which use 18-bit coefficients. Nevertheless, these sidelobes lie below the 54 dB of dynamic range available after the *quant* block, so cannot be seen unless the signal is clipping elsewhere, which is unsatisfactory.

8 DISCUSSION

The HISPEC spectrometer has been designed for wide-bandwidth spectroscopy at L-band wavelengths. Consequently, it is illuminating to compare its performance and specifications with existing instruments in use at major science facilities.

8.1 COMPARISON TO EXISTING INSTRUMENTS

Table 3 presents the specifications of three L-band spectrometers currently in use at major facilities, and contrasts them against HISPEC. MBCORR is the is main spectrometer in use at the Parkes 64 m telescope (Staveley-Smith et al., 1996); GBTS is the lag correlator in use at the Robert C. Byrd 100 m dish at Green Bank, West Virginia (GBT, 2011); and FFTS is the L-band spectrometer in use at the Effelsberg 100 m telescope (Klein et al., 2006).

QUANTIZATION EFFICIENCY Signal quantization affects the signal-to-noise ratio (SNR) of a signal. The fractional loss in SNR is known as the efficiency factor, n_Q; Thompson et al. (2007) gives convenient formulae to calculate n_Q. The GBT Spectrometer and MBCORR use 3- and 4-level quantization, and corresponding efficiency factors of 0.8812 and 0.8098, respectively. In comparison, HISPEC and FFTS use 256-level quantization (8-bit), which yields efficiency factors close to unity. As shown by Thompson et al. (2007), the efficiency factor depends upon the root mean square (RMS) value of the input signal. If one sets the input power to be 6 dB

below the full-scale level, then an efficiency factor of 0.9999 is achieved. However, for L-band applications it is desirable to have some headroom so that intermittent RFI sources do not cause clipping. A more suitable choice is to set the RMS power of the signal to 30 dB below the full-scale level; this yields an efficiency factor n_Q=0.9984 while allowing 24 dB of headroom for RFI sources.

MAXIMUM CHANNELS AT FULL BANDWIDTH These three spectrometers are all flexible: bandwidth may be traded off for increased spectral resolution. HISPEC is also bandwidth agile as the reference clock signal can be set as low as 200 MHz, resulting in 100 MHz of bandwidth. However, this is not an efficient use of the FPGA's resources because the number of operations performed per second drops accordingly. For this reason, the table compares maximum channels at maximum instantaneous bandwidth, as opposed to maximal spectral-resolution modes.

INTERCHANNEL ISOLATION An ideal filter channel would have a width equal to the frequency spacing, with unity response over its bandwidth $\Delta\nu$ and zero response elsewhere. As a perfect FIR filter would require an infinite number of coefficients, all filterbank implementations suffer from imperfect responses: neighbouring channels overlap, and there is always some level of spectral leakage between channels.

The interchannel isolation of a spectrometer is dependent upon the windowing function used. FFTS uses a 3-term Blackmann-Harris window, while a Hann window is routinely applied to data from MBCORR and GBTS. For a Hann-windowed ACS, the first sidelobe level is -16 dB. In contrast, the Blackman-Harris window of FFTS gives -60 dB sidelobe suppression. HISPEC uses a Hamming window, a version of the Hann window modified to minimize the first sidelobe level, which gives a sidelobe level of -50 dB. Note that the first sidelobe of a Hamming-windowed FFT lies 6 dB higher, at -44 dB.

Applying a windowing function increases the width of the main lobe. The first null in the response of a non-windowed FFT occurs at ± 1 channel (see Figure 1a), giving a first null bandwidth (FNBW) of $2\Delta\nu$. For a Hann window, the main lobe width is doubled; for a Blackman-Harris window, it is quadrupled to $8\Delta\nu$. In comparison, a 4-tap Hamming-windowed PFB has a FNBW of $1.88\Delta\nu$. This means that neighbouring channels overlap less, so spectral resolution is higher. At the location of the first sidelobe of the Blackman-Harris window, the 4-tap Hamming window is -80 dB, a full 20 dB lower.

Table 3: Comparison of HISPEC with current L-band spectrometers. Note that all of these spectrometers have multiple modes of operation; I have listed here maximum channels at maximum instantaneous bandwidth.

	MBCORR Parkes 64 m	GBTS Green Bank 100 m	FFTS Effelsberg 100 m	HISPEC This work
Sampling	4 level (n_Q=0.8812)	3 level (n_Q=0.8098)	256 level (n_Q=0.9948)	256 level (n_Q=0.9948)
Max bandwidth	64 MHz	800 MHz	500 MHz	512 MHz
Max chans (full BW)	1024	8192	16384	8192
Implementation	ACS (1024 lag)	ACS (8192 lag)	FTF (32768 point)	PFB (4 tap, 16384 point)
Windowing function	Hann	Hann	Blackman-Harris	Hamming
First sidelobe level	-16 dB	-16 dB	-60 dB	-50 dB
Main lobe FNBW	$4\Delta\nu$	$4\Delta\nu$	$8\Delta\nu$	$1.88\Delta\nu$
Polarimetry	I, Q	I, Q	I, Q	I, Q, U, V

POLARIMETRY As MBCORR, GBTS, and FFTS do not compute the cross correlations of two input signals, only the I and Q Stokes parameters may be formed. HISPEC does compute the cross correlations, so the U and V Stokes parameters may also be formed. as such, HISPEC is suitable for polarimetric observations, where FFTS, MBCORR and GBTS are not.

8.2 NEXT GENERATION SPECTROMETERS

HISPEC is one of many planned modes of the HIPSR project, which will likely replace the aging MBCORR. There is similar upgrade work taking place at Green Bank and Effelsberg.

XFFTS A successor to FFTS, named XFFTS, has recently been installed at Effelsberg (Klein et al., 2012). XFFTS is a polyphase filterbank implemented on a Virtex-6 based board with an integrated e2v 4×1.25GS/s, 10-bit digitizer. The Virtex-6 chip in XFFTS is more powerful than the ROACH Virtex-5 and as such each XFFTS board is capable of digitizing and more bandwidth and processing more channels than a HISPEC programmed ROACH.

While both XFFTS and ROACH are Xilinx FPGA based, the design of the two are markedly different. The ROACH does not have an on-board digitizer, but instead has a ZDOK connector with which to connect daughter boards. ROACH boards are designed to be interconnected by up to four 10GbE/XAUI connections, whereas XFFTS board only offers 1GbE (they are not designed to be interconnected). The QDR, DDR and PowerPC of the ROACH are also absent from the XFFTS. The absence of such features allows the XFFTS board to fit onto a eurocard — the same

size as a single ROACH daughter board.

VEGAS The CASPER collaboration have also designed a Virtex-6 based instrument, ROACH-2. The ROACH-2 board is similar in design to ROACH, but has a more powerful FPGA and increased I/O throughput and memory bandwidth. A number of ROACH-2 boards have been installed at Green Bank for the VEGAS project (Versatile GBT Astronomical Spectrometer), which will replace the GBTS spectrometer and add new functionality (Werthimer, 2012). The ROACH-2 boards are paired with a National ADC08300 based 3Gsample/s digitizer card. In total, 8 ROACH-2 boards will be installed, and will be connected by 10Gb Ethernet to a CPU/GPU based high performance processing cluster.

This heterogeneous architecture is expected to greatly increase the flexibility of the system, by allowing algorithms not suitable for the FPGA to be run on the CPU/GPU nodes, and vice-versa. The HIPSR system also uses a heterogeneous architecture; however, the HISPEC firmware does not require any CPU/GPU based signal processing as all necessary tasks are conducted on the FPGA. In contrast, the pulsar and high time resolution modes of operation use the CPU/GPUs extensively.

9 Conclusions

Polyphase filterbanks have attractive characteristics for radio astronomy applications. They offer excellent interchannel isolation and can be implemented using efficient FFT based structures. The HISPEC 8192 spectrometer is implemented on CASPER ROACH boards, using the Xilinx toolflow and CASPER block libraries. The HISPEC spectrometer is one of many planned modes of operation for the HIPSR system.

References

Atmel (2009). AT84AD001C Dual 8-bit 1 Gsps ADC. Technical report, Atmel.

Bellanger, M., Bonnerot, G., & Coudreuse, M. (1976). Digital filtering by polyphase network:application to sample-rate alteration and filter banks. Acoustics, Speech and Signal Processing, IEEE Transactions on, 24(2), 109 – 114.

Brigham, E. O. (1988). The FFT and its applications. Prentice-Hall Inc.

Bunton, J. (2000). An improved fx correlator. Alma Memorandum Series, (342).

Chikada, Y., Ishiguro, M., Hirabayashi, H., Morimoto, M., & Morita, K.-I. (1987). A 6 x 320-mhz 1024-channel fft cross-spectrum analyzer for radio astronomy. IEEE, 75, 1203.

Gade, S. & Herlufsen, H. (1987). Use of weighting functions in dft/fft analysis (part i). Brüel & Kjær, Windows to FFT Analysis (Part I) Technical Review, (3).

GBT (2011). Observing with the Green Bank Telescope. observer guide, NRAO.

Harris, F. (2004). Multirate Signal Processing for Communication System. Prentice-Hall Inc.

Hickish, J., Foster, G., Adami, K. Z., & Jones, M. (2012). Multipliers, multipliers, multipliers. in prep.

Kestur, S., Davis, J. D., & Williams, O. (2010). Blas comparison on fpga, cpu and gpu. In Proceedings of the 2010 IEEE Annual Symposium on VLSI, ISVLSI '10 (pp. 288–293). Washington, DC, USA: IEEE Computer Society.

Klein, B., Hochgürtel, S., Krämer, I., et al. (2012). High-resolution wide-band fast Fourier transform spectrometers. A & A, 542.

Klein, B., Philipp, S. D., Gusten, R., Kramer, I., & Samtleben, D. (2006). A new generation of spectrometers for radio astronomy: Fast fourier transform spectrometer. In J. C. Mather, H. A. MacEwen, & M. W. M. de Graauw (Eds.), Space Telescopes and Instrumentation I: Optical, Infrared, and Millimeter, volume 6275 (pp. 627511).: SPIE.

Kuon, I. & Rose, J. (2006). Measuring the gap between fpgas and asics. In Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays, FPGA '06 (pp. 21–30). New York, NY, USA: ACM.

Lyons, R. G. (2004). Understanding digital signal processing. Prentice Hall, 2nd edition.

Parsons, A., Backer, D., Chang, C., & Chapman, D. (2006). Petaflop/second fpga signal processing for seti and radio astronomy. Signals.

Parsons, A. R. (2009). Low-frequency interferometry: Design, calibration, and analysis towards detecting the epoch of reionization. CASPER Theses, (pp. 1–127).

Smith, S. W. (2002). Digital Signal Processing: A Practical Guide for Engineers and Scientists. Newnes.

Staveley-Smith, L., Wilson, W. E., Bird, T. S., et al. (1996). The parkes 21 cm multibeam receiver. Publications Astronomical Society of Australia, 13, 243.

Szomoru, A. (2010). The UniBoard. In 10th European VLBI Network Symposium and EVN Users Meeting: VLBI and the New Generation of Radio Arrays.

Taylor, G. B., Carilli, C. L., & Perley, R. A., Eds. (1999). Synthesis Imaging in Radio Astronomy II, volume 180. Astronomical Society of the Pacific.

Thompson, A. R., Emerson, D. T., & Schwab, F. R. (2007). Convenient formulas for quantization efficiency. Radio Science, 42, 3022.

TTi (2006). TGR2050 Instruction Manual. Technical report, Thurlby Thandar Instruments.

Vaidyanathan (1990). Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial. Proceedings of the IEEE, 78(1), 56 – 93.

Weinreb, S. (1963). A digital spectral analysis technique and its application to radio astronomy. MIT Research Laboratory of Electronics Technical report.

Weinreb, S., Barrett, A. H., Meeks, M. L., & Henry, J. C. (1963). Radio observations of oh in the interstellar medium. Nature, 200, 829.

Werthimer, D. (2012). personal communication.

Xilinx (2012). Virtex-5 FPGA XtremeDSP Design Considerations. Technical report, Xilinx Incorporated.

Zimmerman, G. A. & Gulkis, S. (1991). Polyphase-discrete fourier transform spectrum analysis for the search for extraterrestrial intelligence sky survey. The Telecommunications and Data Acquisition Progress Report, 107, 141.