EXPReS JRA1 FABRIC
Data Acquisition Design

Guifré Molera, Ari Mujunen, Jouko Ritakari, Jan Wagner
Metsähovi Radio Observatory, Helsinki University of Technology TKK

Bryan Anderson
Jodrell Bank Observatory, University of Manchester

1st February 2007

Abstract

This document describes the overall design of the data acquisition work package within FABRIC. It aims at a demonstratable 4 Gbps network-based system which is scalable beyond the initial 4 Gbps goal and which can be used for both real-time (broadband Onsala—eMERLIN—JIVE tests) and non-real-time VLBI data acquisition (DAQ prototype demonstration).[1]

Contents

[1]This work has received financial support from the European Commission (DG-INFSO), within the Sixth Framework Programme (Integrated Infrastructure Initiative contract number 026642, EXPReS).

1 Purpose of this Document and Background

The purpose of this document is to provide a relatively easy growth path from the eVLBI systems we use now to future multi-gigabit/s eVLBI systems.

We propose to change the existing paradigm of making VLBI data available in a commodity microcomputer via a special interface into making VLBI data directly available on the network via dedicated devices. Two factors are the driving force behind this reasoning.

1. The available bandwidth of networking (both local and global) is increasing beyond expectations, with 10 Gbps Ethernet proliferating rapidly and 40 Gbps and more already on the horizon.
2. The tools and available template hardware for FPGA-based hardware/firmware implementations of high-speed RF-to-net applications have improved and become more readily available.

1.1 Existing Data Acquisition Systems

At the moment two production computer-based data acquisition systems (DAS) with disk storage medium are in use in the EVN: the Mark5 system designed by the MIT Haystack Observatory [1], and the PCEVN system designed by the Metsähovi Radio Observatory [2].

The basic designs of both systems are five years old and mature. The Mark5 has been in more widespread use and has been recently upgraded from Mark5A to Mark5B. This upgrade consists of swapping a newer PCI-based I/O card and drops the requirement of external data formatter hardware and provides standard VSI connectors. During the same time the PCEVN's PCI VSIB board has not been upgraded since it has been using VSI connectors from the beginning. Instead PCEVN has successfully made use of the rapid consumer PC development to bring all performance improvements. Currently the main difference between PCEVN and the Mark5 is that the Mark5 system is able to record 1 Gbps with one expensive special computer, while the the PCEVN system provides the same scalability with two low-cost commodity computers. At slower speeds both systems need one computer.

There are also a number of other DAQ systems that have been in experimental use for some time but are not yet fully out of their prototyping or near completion stages. None of the systems contain disk storage. One system is the "Digital Baseband Converter" system (dBBC) [3] developed over years at the Istituto di Radioastronomia in Italy. The dBBC is a compact digital remake of the legacy analog baseband converter, sampler and data formatter systems, and attempts to be a drop-in replacement for these systems with data output over a VSI connector. Another similar system is the Haystack "Digital Back End" (DBE) [4]. All of the above mentioned systems have been able to achieve VLBI fringes.

1.2 State of the Art of eVLBI

Both existing systems described above have demonstrated 512 Mbps real-time transfer, the Mark5 system in the iGrid2005 event[2] and the PCEVN system in the "Month7 demo" [5].[3] The Mark5 system has achieved this by using TCP/IP, jumbo frames and dedicated light paths. The PCEVN system has used normal Internet, normal frames and the more efficient UDP/IP-based Tsunami protocol.

It is obvious to arrive to a conclusion that in both cases we get in a bottleneck when we approach 1 Gbps. Same problem, different reasons: hardware limitations, connectivity interfaces or network congestions. There is the necessity to open a new research path to adapt the old data acquisition systems to the new requirements.

[2]http://www.haystack.mit.edu/tech/vlbi/evlbi/igrid2005.html

[3]PCEVN tests between Jodrell Bank, Onsala, Metsähovi and JIVE.

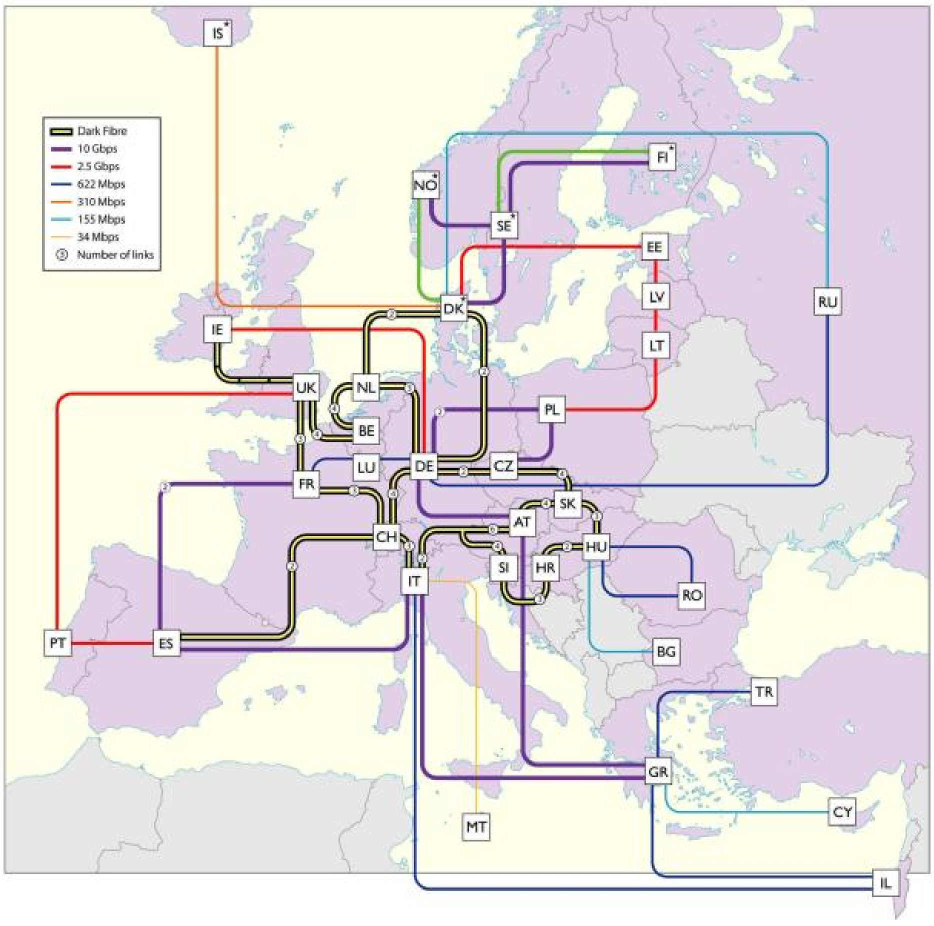

Figure 1: The GEANT2 fibre paths spread around Europe.

Our point of view and consequently the path to follow the research drives us to the so called “sample RF signal directly to Ethernet” approach applied to the eVLBI. As we know, nowadays, there is a great communications infrastructure spread around Europe, the GEANT2 network[4] which provides transfer rates at 10 Gbps to almost every European country, see detailed European connections in Figure 1. Should we not take maximal advantage of it?

According to our perspective, the receiver stations will need only to acquire radio astronomical data and packetize in situ to UDP packets to send it over the fibre cable. All the process of capturing and packetizing can be performanced by a single high-tech board. Using the multi-gigabit network and the concept of grid-based correlator

[4]GEANT2 is the seventh generation of Pan-European Research and Education Network, successor to the pan-European multi-gigabit research network GEANT.

(see the block diagram in Figure 2), a large-scale computation system can be designed for using to correlate the data outcoming from each station.

The grid-based correlator have a clear concept to use clusters and multiprocessor computers on a various grid nodes in different local positions. It exists two different levels of implementations, using distribution to the grid nodes or over the CPUs in multiprocessor and/or multi-core-CPU computers. We will not get further in the topic as the document elaborated by the PSNC "eVLBI—Grid Design Document" [14] is a perfect report about the possible ways to develop the concept.

In that sense, there is not need of hard design on the receiving stations, because the existing network can solve our storing and computational problems. So, we come to the conclusion that there is no need to invest in upgrading or new developments of old receiver systems as Mark5 or VSIB.

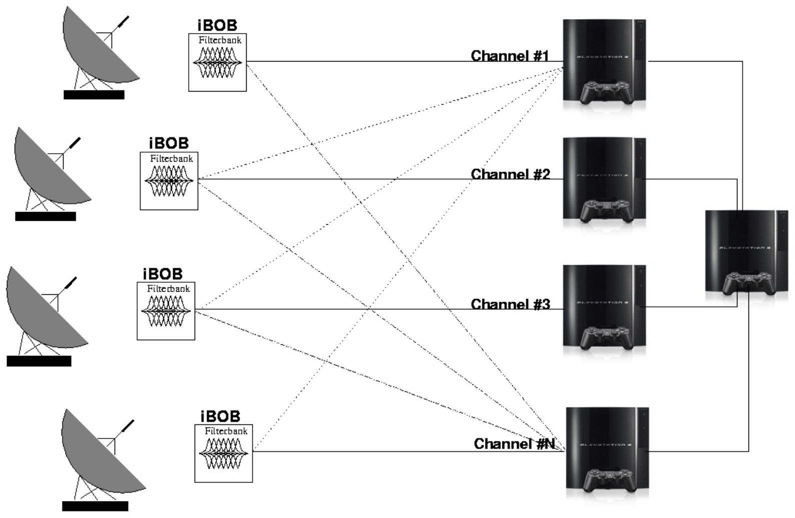

Figure 2: Design of grid-based correlation using iBOBs to acquire data and transmit at 10 Gbps to PS3 Cell processor computing nodes to run the software correlator.

2 Design Alternatives and Recent New Developments

Without building larger telescopes or altering the existing telescopes of observatories, which both are highly impractical and expensive, the best way to increase VLBI sensitivity is to observe broader bands in the radio spectrum. Currently a typical maximum band slice covers 16 MHz in the radio spectrum. Sampling with a higher bandwidth increases VLBI sensitivity but also leads to an increased data rate. From the network point of view, increased rates are not a problem, as many observatories today have access to 10G links to fast networks such as GEANT2, Figure 1. So far most eVLBI experiments have been mostly done with 128 Mbps. A move to for example 4 Gbps, possible with a 10G link, would result in a six-fold increase in sensitivity. To achieve high rates an aim of new eVLBI development is to overcome the current 512 Mbps bottlenecks of Mark5 and PCEVN and allow much wider bands to be digitized, transferred and correlated in real time, significantly improving VLBI in the EVN. A currently set goal is at 4 Gbps transfers. Any new Data Acquisition System (DAS) developed should be capable of such data rates in addition of being highly scalable.

2.1 VSIB Upgrade Path

One possible way to achieve high DAS data rates was mentioned in the previous May deliverable [7], where it was amongst others considered to adapt the VSIB board to the new PCI Express computer expansion bus standard. Upgrading the VSIB board to PCI Express in the most straightforward way would require new PLX chips that are still not available, and are not a straight drop-in replacement of the currently used chip, thus would require a larger redesign of the board. Furthermore, the maximum standardized bandwidth of a VSI connector is only 2 Gbps (32 bits times 64 Mbps/MHz, “double rate”), with some speculation that VSI might work at 4 Gbps (32 x 128 Mbps/MHz, “quadruple rate”). This would lock us into our initial goal rate of 4 Gbps for as long as we choose to use VSI connectors.

Instead of a VSIB board remake, a CameraLink PCI Express[5] board could be the basis for fast data rate handled by COTS computers and Camera Link cards. The actual commercially available boards use x2 PCI Express lanes which translates to a data processing rate of 2x340 Mbytes/s (2.5 Gbps). The LVDS chip set used on these cards limits the rate to somewhat above 2 Gbps,[6] while also the MDR type of connector in the VSI standard has a limit at 2 Gbps. Even though this can be 4 times faster than the current VSIB board the disadvantages become obvious. There is no chance to upgrade up to the target of 4 Gbps, the Camera Link boards are still unattainable for a COTS PC and implementations at higher lane (4x, 8x or 16x) are not yet available and sets out a complex level of design.

As a conclusion, the reasons above reaffirm the decision of Metsähovi not to develop further research in the VSIB board area. In addition we consider that even though the VSI interface and connector could be an acceptable short term solution, it is not a good solution in the long term and should be avoided in new DAS designs.

[5]http://sine.ni.com/nips/cds/view/p/lang/en/nid/14518

[6]Camera Link Interface Standard Specifications, http://www.silicon-software.com/download/CameraLink.pdf

2.2 iBOB and BEE2

The SETI Institute of the University of California, Berkeley (UCB) has developed a very interesting new hardware platform [9]. This iBOB/BEE2 platform is a general-purpose one and it has been already used in several astronomy-related projects. It is also very well suited for both normal VLBI and eVLBI. The development has been very quick because of the use of high-level simulation tools combined with commercially available intellectual property, for example the 10 Gbps Ethernet controller.

The main part of this platform is the BEE2 (Berkeley Emulation Engine 2) board that contains four large Xilinx chips for data processing and a fifth for controlling them. The control chip has on-chip PowerPC processor running the Linux operating system. For data communications the board has twenty 10 Gbps Ethernet ports.

Originally the iBOB board was intended to be a simple I/O board for data acquisition. The board has two connectors for a 1 Gsample per second (1 Gsps) dual (or 2 Gsps single) ADC converter board each, dual 10 Gbps Ethernet ports, a VSI port and a medium-size Xilinx FPGA chip. The iBOB board lacks some of the flexibility of the BEE2, since it has no on-board Linux and the on-board memory size is limited to 4 MB.

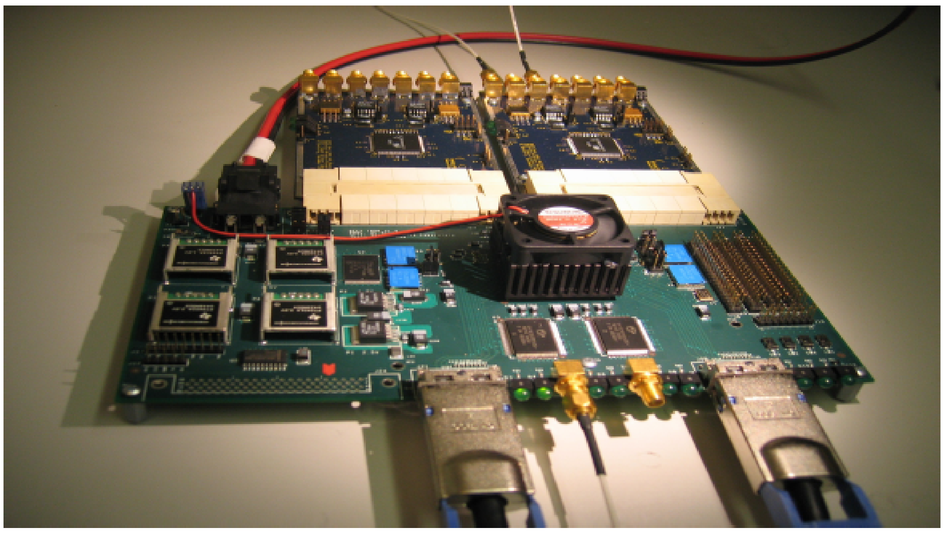

Figure 3: Picture of an iBOB board under testing. The board can accept two ADC boards with a total of four RF inputs. The FPGA core is cooled by a small fan and the two chips in front provide four megabytes of ZBT SRAM memory.

MIT Haystack observatory has already done some work in collaboration with the UCB SETI Institute to adapt the board for VLBI use. The Haystack/SETI instrument is called DBE (digital backend) and contains a 32-channel polyphase filter. Prototype of this design exists but it outputs only VSI.

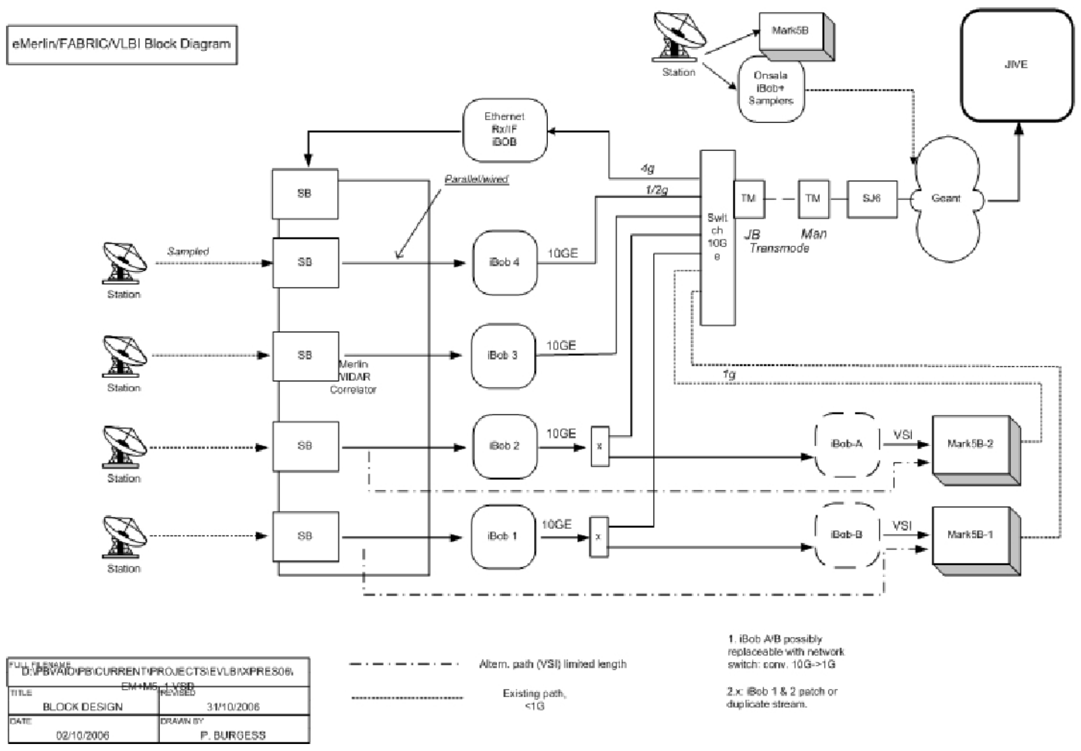

Figure 4: Possible configuration using iBOB planned by Jodrell Bank Observatory. (Diagram by Paul Burgess, JBO.)

3 Possible Improvements to the iBOB/DBE Design

The Haystack/SETI DBE suffers from a few limitations: It uses the 1 Gbps VSI port for output, it does not use the more powerful 10 Gbps Ethernet ports at all and it does not have Linux support for development nor run-time. This makes the design less flexible than it could be.

Fortunately we can use several tricks to increase the bandwidth and simplify the use of the iBOB.

If we want to increase the bandwidth we can move the output to the 10 Gbps Ethernets. This would increase the bandwidth to at least 4 Gbps and allow the use of all four 1 Gsps A/D converters simultaneously. A prototype design for a hardware UDP/IP packetizer has already been developed in UCB and could be used as a basis for further development work.

The BEE2 system has a simple programming interface and a straightforward development environment that could significantly speed up iBOB development. The existing BEE2 code-base could be partly moved to the iBOB allowing it to be more versatile. For development, UCB has used the Matlab and Simulink combination for a high level implementation of BEE2 functions, and Xilinx System Generator to automatically generate VHDL code for the BEE2 FPGAs. As shown in Figure 5 the BEE2 contains a control FPGA to distribute the VHDL or Matlab Simulink generated programs to the four data processor FPGAs. The control FPGA is interfaced over standard JTAG to the development PC, and in addition it runs a Linux system for controlling the BEE2 FPGAs.

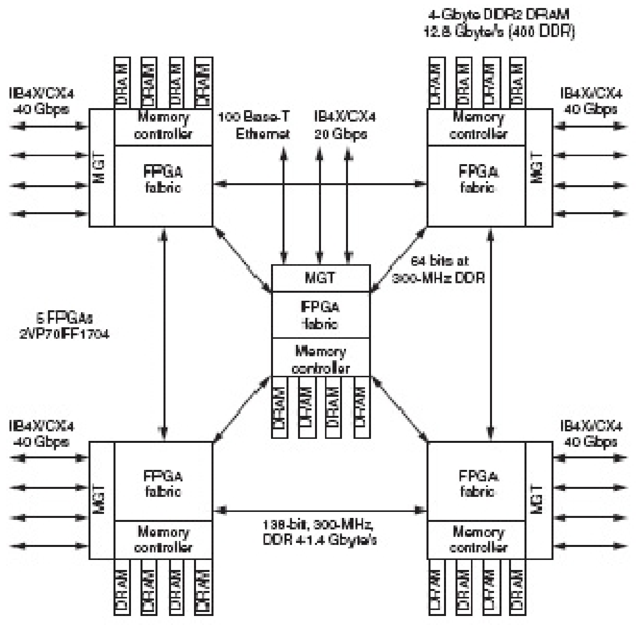

Figure 5: Block diagram of the BEE2.

The iBOB has only one FPGA with a built-in PowerPC processor core which could be used to run a similar control system as on the BEE2 control FPGA. However, it is very possible that we cannot use this on-board PowerPC because of memory constraints. While the iBOB does have 4 MB of fast ZBT SRAM, a large part of it is required for buffering the data that would be sent to the Ethernet, and not much memory remains for a Linux system.

One possible workaround to this problem is that we leave the PowerPC core unused and instead emulate the Linux control FPGA of the BEE2 by interfacing an external Linux PC to the iBOB VSI port using a VSIB board. The VSI port available on the iBOB would then be a bi-directional parallel control bus very similar to that on the

BEE2. This would have the possible advantage that we would be able to reuse most of the existing BEE2 Linux software and Matlab and VHDL code. We have nicknamed this approach “FrankeniBOB” and discuss it further below.

3.1 FrankeniBOB—a good idea or bad?

“FrankeniBOB” is the working name for an iBOB augmented with a stolen brain (e.g. an external Linux computer) performing run-time control of the (sole) FPGA of iBOB. There are good arguments both for and against this approach. The arguments against the FrankeniBOB are essentially the same as the arguments for it.

Why reinvent the BEE2 if we can buy it? The price may look a bit high (about 20.000 USD) but if we can speed up a significant radio astronomy instrumentation design project with it, it is definitely worth paying. In any case, radio astronomy equipment is never going to be produced in large quantities. Usually the best approach is to use existing off-the-shelf hardware “as is” and not to save in hardware unit cost.

The BEE2 approach would also be better for protocol design with its massive amounts of available DRAM memory and the embedded Linux computer. The FrankeniBOB possibly may not contain enough memory to allow for retransmissions, forcing us to settle for protocols with some degree of potential packet loss. In that sense, we conclude that the future of the eVLBI observations should move around the new iBOB or the upgraded version BEE2, in case the first presents handicaps for low memory or other features. Our point of view currently is that there is no need to implement a totally new and upgraded version of neither iBOB nor BEE2 designs and/or Printed Circuits Boards (PCBs). Instead, it will become enough challenge to attempt to integrate the realisations of majority of requirements needed in the VLBI data acquisition. Should we exhaust the resources available in the iBOB we can readily connect the existing prototyping iBOBs to a BEE2 board via the 10GE links and continue with adding functionality on the BEE2 FPGAs.

3.2 Moving RF Processing to Silicon and to Software

RF front-ends are used in the path down from the analog signal received by the antenna to the final analog-to-digital converter (ADC). The front-ends can be tuned to some RF center frequency within the observed RF band, after which they mix the lower and upper sidebands around the center frequency down to baseband for further analog processing. Other typical tasks of the RF front-end are gain control of the RF signal, measuring the RF power, and applying a low-pass anti-aliasing filter to the baseband signal before it is sampled by the ADC. Currently the major RF front-end equipment used in the EVN is Haystack’s analog Baseband Converter

(BBC). BBCs can no longer be manufactured, mainly because of the unavailable obsolete components used in the design, completed in 1987. This also makes them a maintenance disaster and thus there is a real need for new RF front-ends to replace the BBCs. Several new prototype front-ends have already been developed at different institutes.

While the traditional analog processing of the RF signal has its benefits, mainly the large dynamic range, there are several problems and shortcomings to the analog processing approach that have been acknowledged by the communications industry for already a decade. The mobile communications and broadcasting industry have heavily driven the development of software defined radio equipment that uses minimal analog RF signal processing. Several projects for software radio exist, including many supported by EC, such as SORT.

In the new digital radio equipment, the RF signal is directly sampled with ADCs. Functions previously implemented in analog electronics are moved into the digital signal processing domain, onto silicon and into software. The obvious benefits of digital RF processing are small size, low cost, easy reconfigurability, an easier design process with support from libraries and intellectual property (IP) building blocks, and shortened time to market.

Examples of current projects in radio astronomy where digital radio is being developed are the Haystack Digital Backend, the Italian digital BBC, LOFAR distributed radio telescope, ALMA digital BBC, and many others. UCB SETI Institute iBOB design too has all necessary capabilities to be used as a RF front-end with its own signal processing.

The iBOB was specifically developed for digital RF preprocessing in the UCB SETI project. The project has additionally developed ADC boards that plug into the iBOB ADC connector. The ADC board is capable of quadrature-sampling RF signals up to 512 MHz in bandwidth with 8-bit resolution, making it suitable for radio astronomy and VLBI.

The maximum sampling rate of the ADC board is 2.048 Msps i.e. it is capable of capturing a wide band of the RF signal. This wide band can be split into narrower sub-bands or bands of interest can be extracted by applying a simple digital filter to the sampling data stream. If quadrature sampling is used, that is, if the ADC board is fed with a local oscillator (LO) clock that has both sine (I) and cosine (quadrature, Q) outputs and two ADC channels are used to sample the same RF signal into I and Q data streams, then the large band or sub-bands can be split into upper and lower sidebands (USB, LSB) by using a polyphase filter—a normal type I or II filter that has a large structure, as shown in the polyphase filter example block diagram, Figure 6.

Summation of the I and Q signals gives the desired sideband as shown in Figure 7. Several polyphase filters in VHDL or other code suitable for FPGAs already ex-

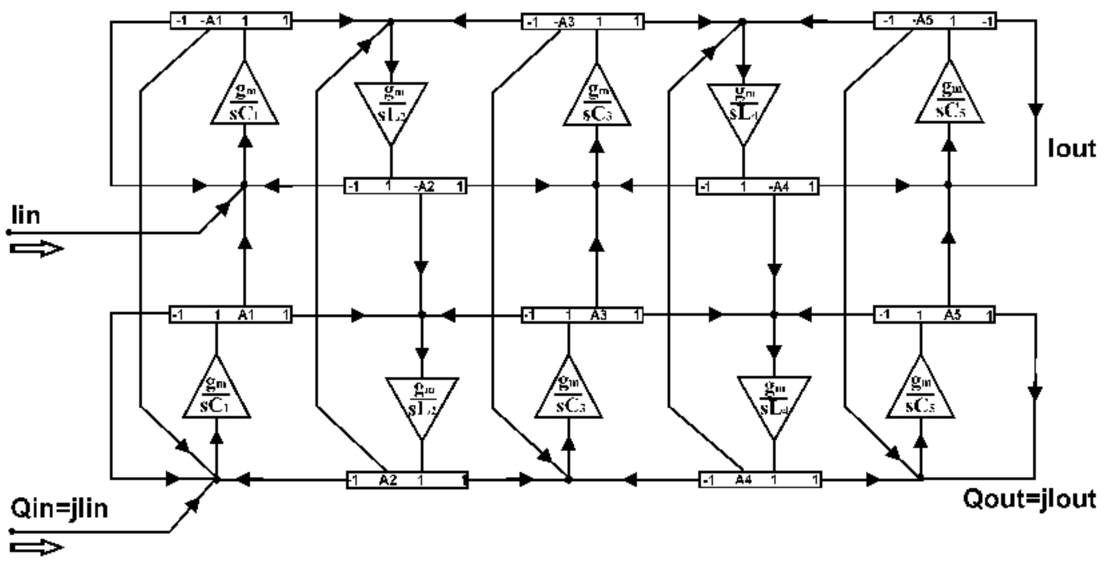

Figure 6: Example of a polyphase filter. Extracted from [12].

ists, and it is trivial to generate custom filters in for example Matlab. UCB has an existing 1024 sub-channel (as well as a 32 sub-channel) polyphase filter bank implementation available for the iBOB, based on Matlab and VHDL. Due to the old analog baseband converters used in VLBI, the use of USB and LSB image selection has been a standard requirement. Such legacy may not be required in digital processing and new correlators—when only the concept of sub-bands is used, a much simpler filter implementation without quadrature sampling becomes possible, the implementation needs less FPGA space and allows more bands to be extracted using the same FPGA.

A final Haystack Digital Backend (DBE) may have the same iBOB/ADC board setup that we might end up using, but the DBE is implementing a different functionality since they attempt to maintain interface backwards compatibility with their Mark5 systems, their legacy analog baseband converter, and VSI cabling, instead of using the standard 10 Gbps Ethernet to distribute the data. It is unlikely that the Haystack DBE will be extended for distributed software correlation. However, code derived from the DBE's RF processing functions may find applications also in the Metsähovi FrankeniBob. This will have to be evaluated, too.

The current goal for iBOB ADC evaluation and digital RF processing development at Metsähovi is to find out how well the iBOB can be adapted to VLBI use, for example how many sub-bands the current FPGA is capable of generating and processing, and how well this will extend to the data transfer needs in future grid correlation. In grid correlation, the iBOB would split the RF band into sub-bands and stream individual bands to different client computers on a fast network.

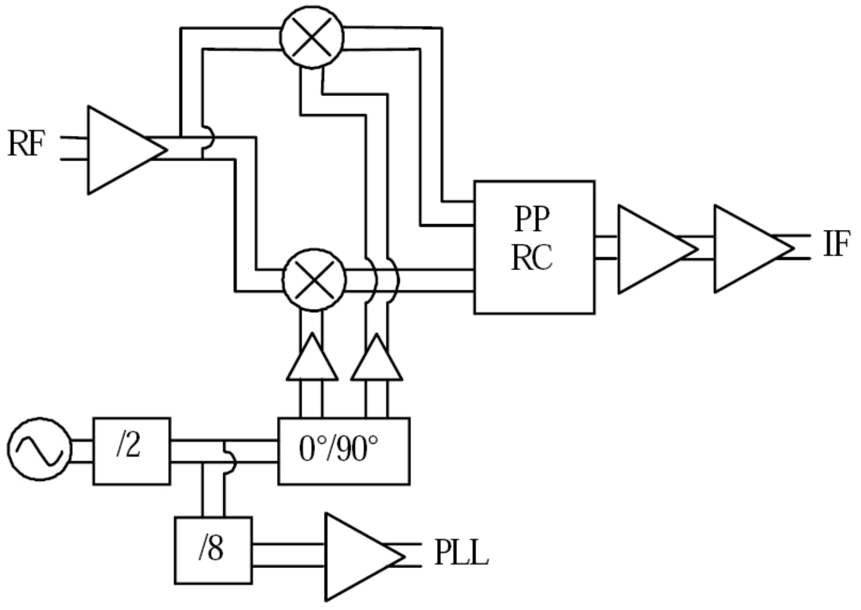

Figure 7: Example implementation of the iBOB and ADC quadrature receiver with a polyphase filter (PP RC). From [10].

3.3 Moving Data Communications Protocols to Silicon

The Tsunami approach [11] uses a relatively simple UDP/IP protocol to stream packets through Internet at fixed rate. It is using POSIX code thus runs on Linux, Unix and others, but the main inconvenience in an iBOB-based implementation could be the limited amount of memory available on iBOBs to solve retransmission requirements if we desire to have a zero packet loss system. Upgrading iBOB with extra memory path is possible through e.g. the second ADC port, though hardly convenient. This leads us to the question of retransmission necessity.

The actual average of data loss at 0.5 Gbps rate is around 1% demonstrated during Month7 [6]. It rises considerably when using higher transfer rates due to limited low-bandwidth and congestion problems (i.e. when approaching the nominal bandwidth of a given connection). In a sense is not harmful to have certain packet loss.

The relation between the Signal-to-Noise (S/N) of the correlation is determined by the equation: $S/N = \sqrt{BW(1-f)}$ according to the article Spencer [13]. It is demonstrated that response for UDP transmission is smoother than TCP up to 20% of packet loss. Even with high packet loss we can establish a correct S/N level in the correlation if we can omit the lost data from correlation calculations. This opens a discussion about the the necessity to build a bit more complex iBOB but ensuring 100% of data received or the possibility to allow (with detection) a certain loss which in reality has minimal impact on the correlation result.

3.4 UDP/IP Frames

If we examine the UDP and IP RFCs, we notice that the UDP and IP packet headers are constant for each UDP/IP source/destination pair, if the packet size is constant. This means that the headers need to be computationally calculated only once for each pair. This simplifies the FPGA implementation of data streaming: the header calculation can be done in software and the resulting header simply downloaded to FPGA header registers before starting the bulk high-speed transmission of UDP packets.

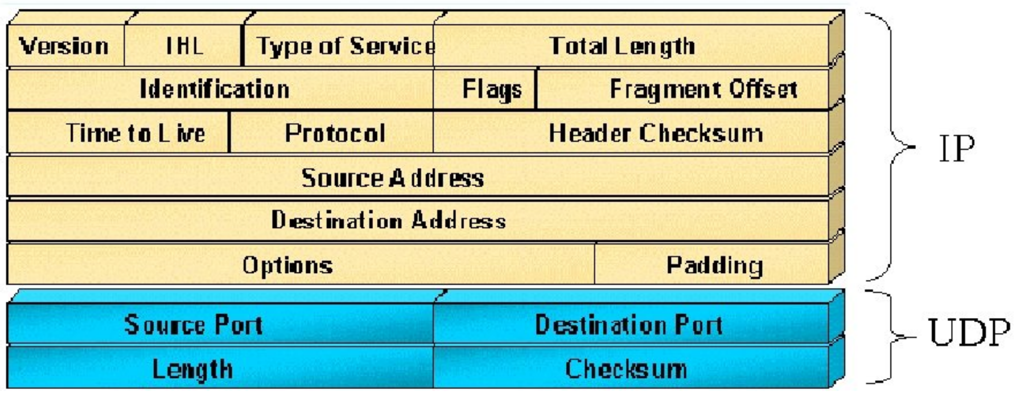

Figure 8: UDP/IP basic block diagram

Sending timewise equally-spaced (that is, controlled data rate) packets becomes also trivial, because the inter-packet delay can be produced with counters. An initial prototype implementation could use simple UDP streaming configured from the control PC. Packets can then be captured using standard tools like “netcat”[7] on the receiving computer or the UDP receiving script developed by MRO in coordination with CERN which is involved in the AMS02 project. If it turns out to be better to use Tsunami, a simple but longer upgrade from simple UDP to Tsunami on silicon (Tsunami only in parts on silicon, rest in software/firmware) could be done.

As mentioned in a report posted by Chris Phillips [8], in case of packet loss tolerance, it might become necessary to implement software correlation with “fill pattern”. The software correlator would have to be modified to detect this fill pattern and ignore the missing data, in order to not interfere to the correlation.

[7]http://netcat.sourceforge.net

When we want to move to grid-based correlation, it is relatively easy to change the UDP/IP packetizer to send the different polyphase filter channels to different correlating computers. This would be a big help in avoiding data communication bottlenecks.

4 eMERLIN Connectivity

This section is a follow up to what has been previously described in [15] on the EXPReS-FABRIC Wiki and in several FABRIC emails. It refines some of the arguments and is more specific in the recommendations.

4.1 Requirements

There is a requirement to be able to export astronomical data from eMERLIN to a VLBI data processor and a requirement to be able to import astronomical data from a *foreign* telescope into eMERLIN. In the both cases, the astronomical data formats should be as compatible as possible with the formats used by the intended destinations. The methods and means should be as flexible as possible so that future enhancements are not ruled out. The transport should be done over the Internet using components and protocols that are available and compatible with public switched networks.

The two applications have some features in common and some differences. The native eMERLIN telescopes provide (with their dedicated fibre links) astronomical data in dual polarisations at 24 Gbps in the case of 3-bit data and 16 Gbps in the case of 8-bit data. The VLBI correlators currently support operations at up to 1 Gbps. eMERLIN internally processes the data at the correlator into up to 18 pairs of bands each up to 128 MHz wide and it correlates, natively, either 4-bit or 7-bit data. VLBI band-processing is done at the telescope into up to 16 bands but each only up to 16 MHz wide. Data are transported and correlated at only 1 or 2 bits of precision. The *solutions* to the requirements should compromise joint eMERLIN-VLBI operations as little as possible.

Within the timescale of the EXPReS/FABRIC project, it is expected that the eVLBI interfaces of the JIVE correlator will support the VSI-E network standards. The target for data import to eMERLIN is to support operations at up to 4 Gbps and the target for export is to be capable of meeting and exceeding current maximum rates of 1 Gbps. Rates of 4 Gbps both in and out would be adequate for the moment.

The devil is in the detail. There are a number of factors that complicate possible solutions to these requirements and, inevitably, that might lead to some compromises.

4.2 eMERLIN Station Board

The principles on which the eMERLIN correlator, a WIDAR correlator, operate and its broad characteristics are described in references [16] and [17]. The correlator is being developed by Peter Dewdney's group with design leader Brent Carlson at the Dominion Radio Observatory at Penticton, Canada. Prototypes of all the major components exist and are being debugged. The correlator functionality is a subset of that of the much larger correlator that they are building for the EVLA. It normally accepts two IF bands 2.048 GHz wide sampled at 4.096 GHz to 3 bits of precision or two IF bands 512 MHz wide sampled at 1.024 GHz to 8 bits of precision, splits each of the bands into 18 sub-bands up to 128 MHz wide and down to Hz wide, resamples the results with either 4-bit or 7-bit precision and then correlates them. Note that there is no fixed relation between input and output data precisions: any combination can be justified and there are no restrictions on which one can be chosen. The signal processing flows in a *station board* (SB) are shown in Figure 9.

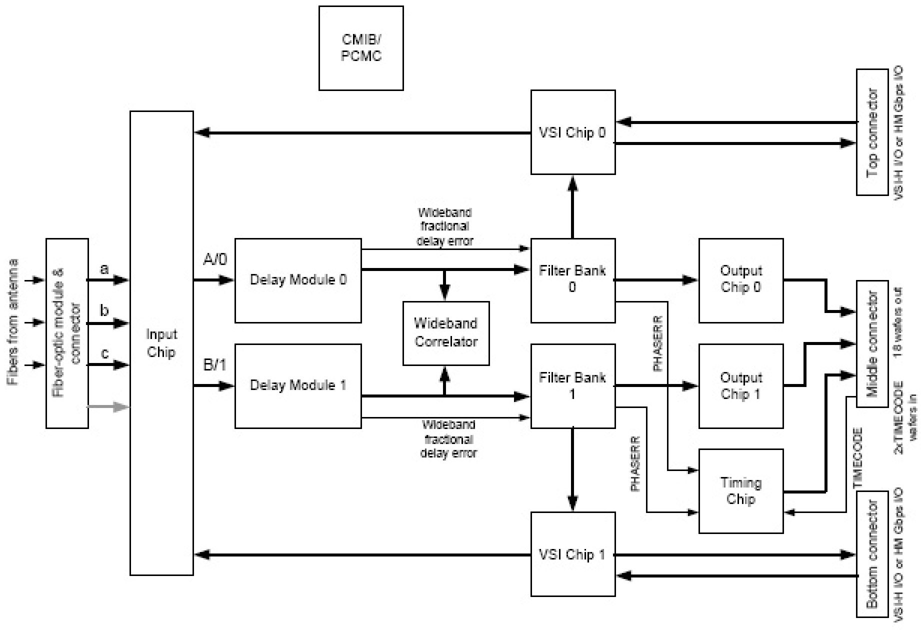

Figure 7-1 Simplified Station Board block diagram.

Figure 9: Simplified diagram of an SB from Carlson's *Programmers Guide*.

The main signal paths are horizontally through the centre of the diagram for each of the two input bands. An input selector is followed by pairs of system delays of up to 0.25 seconds and then 18-channel FIR filterbanks through which the input signals are chained from filter to filter (chip to chip). Each filter output is available at the middle connector for correlation. The astronomical data stream continuously through the system and are accompanied by *data-validity* and time-tick streams. The data-validity streams ensure that only valid data are correlated.

Things to note are the horizontal paths at the top and bottom of the figure. *VSI chips,* Xilinx Virtex-4 SX35 devices, can accept data from each filter bank, process the data in some way and route the results to a pair of connectors at the right-hand side. Likewise a path exists back from the connectors through the VSI chips to the input selector. Whilst the signal connectors are not directly VSI-H compatible, simple wiring harnesses could make them so.

The sub-band connections between the filterbanks and the VSI chips are six differential signals clocked at up to 256 MHz. The six signals are a clock, a time-tick and 4-bit data samples. In 7-bit mode, the least-significant and most-significant nibbles are transmitted alternately. The total data bandwidth from the filterbanks to the VSI chips is thus 2x16x4x256 (polarisations/subchannels/bits/clock), i.e. 32 Gbps. The *VSI connectors* can support 2x32bits at 64 MHz for 4 Gbps and have the potential, in non-VSI mode, at up to 256 MHz clock rate, for 16 Gbps depending on the quality of the pcb traces and connecting cable. The combination of these features give lots of scope for export and import of data.

4.2.1 Export

To export both hands of polarisation, it is necessary to use both VSI connectors because there are no cross-over links between the bottom filterchain and the top VSI chip or vice versa. This feature may make a direct connection from an SB to a VSI-H-compatible external device less attractive in that two external devices would be required.

Note that signals enter the filter chains after passing through the delay RAMs. If eMERLIN is concurrently making observations and trying to export data then the exported data will have the eMERLIN delay model applied, i.e. the delays would vary with time with a peak-to-peak delay range of about 1 millisecond. The VSI chips do not have sufficient internal RAM to correct for the eMERLIN delay so an external solution is needed. Since the applied delay is in steps of the original sampling clock, corrections made at the lower sub-band resampled clocks may need to be augmented by FIR interpolation filters. These might logically be part of the filtering that would make the exported data conform to VLBI spectral channelisations. There may also be differences of a few clock cycles in timing between sub-bands due to pipeline delays in a filterbank chain. These too could be corrected as part of the general delay correction.

There is a lot of signal-processing capability in the VSI chips that could be used to provide whatever VLBI channelisation is desired. The WIDAR principle requires the LO frequencies of each telescope to be offset from each other in distinct multiples of 10 kHz. For narrow-band spectral observations, it is desirable for these offsets to be removed before the final frequency channelisation is performed. A suitable place to do this would also be in the VSI chips.

4.2.2 Import

The major difficulties in importing data into eMERLIN are

1. Delaying eMERLIN signals to match the worst-case remote signal delay.
2. Matching the eMERLIN spectral channelisation.
3. Converting the network data flow from packets to streaming.
4. Removing disjoints in timing caused by the network.
5. Providing timing data to the SB.
6. Dealing with network health.

Since all data into a SB go through delay units and onto the main FIR filters, it is possible for the data from all native telescopes to be delayed by up to 0.25 seconds to align them with latest-arriving imported data and to do all the frequency channelisation in the SB. An external buffer would be required to absorb the worst of the timing jitter since there is not enough RAM in the VSI chips and control of the SB delays cannot be done quickly enough. It is assumed that the WIDAR frequency offsets would be introduced in an LO at the external telescope.

4.3 Proposals

The major features of the proposals are:

1. Solutions are based on iBOB hardware plus a UCB, 8-bit, dual-channel ADC for Onsala. Figure 10 is a block diagram of an iBOB.
2. As much commonality as possible
   (a) VSI-E compatibility
   (b) 10 GbE+UDP and 10/100 Ethernet+TCP network interfaces implemented in an iBOB

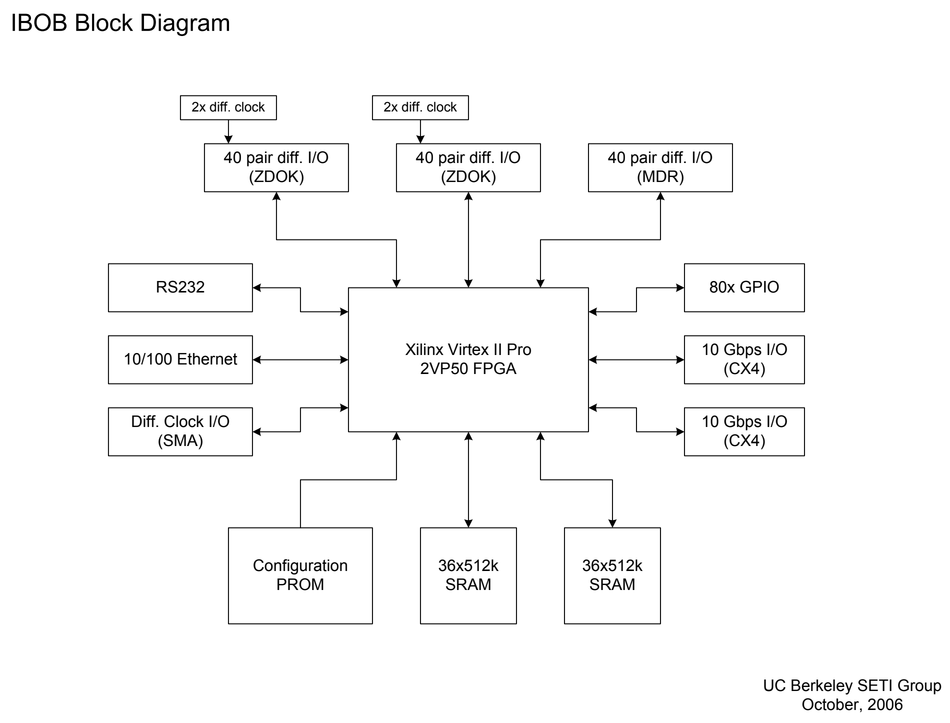

Figure 10: iBOB block diagram.

(c) Band and other processing done at the eMERLIN end

(d) Frequency, band, timing and delay processing distributed between the SB proper, the VSI chips and the SB-interface iBOB.

(e) Network buffering in the iBOB RAM.

Given that the development of the networking aspects needs to be done in any case, data import is particularly simple to implement.

4.3.1 Network Connections

The VSI-E standard requires 2 network channels to be supported: a high-bandwidth data channel which could be unidirectional and a lower bandwidth control channel which must be bidirectional. The former can use UDP and the latter TCP protocols.

Standard network practice is very flexible in that, for example, a 10 GbE interface can talk to a 1 GbE interface through an industry-standard switch or router provided that the net rates are supportable. Since the net rates are determined by the rates at which data packets are sent, it is easy to design a system that would function, in principle, at rates from zero all the way up to 7.2 Gbps in a very flexible way.

The Xilinx Virtex-2 Pro FPGA on an iBOB is interfaced to CX-4 ports allowing 10 GbE to be supported through standard media converters for the main data channel. The iBOB has 2 ZBT SRAMs that support efficient, flow-through data transport for the main signal path. The FPGA, with an added memory card, supports an embedded PowerPC core that could manage the main data channel and use the 10/100 Ethernet port for the control channel.

The interfaces inside the iBOB FPGAs between the signal processing segments and the 10 GbE MACs should be in wide FIFOs so that the signal and networking clock regimes are clearly separated. Bandwidth control is by data selection and packet rates.

The shapes and small sizes, 2x36-bits by 512k and 2x18 Mbits respectively, and the relatively-slow speed of the ZBT SRAM devices (200 MHz) impact on the achievable flow-through data rates and the amount of signal delay that can be achieved. The former is 72 bits by 100 MHz (alternating reads and writes) which is 7.2 Gbps and the latter is 36 Mbit / 4 Gbps or 9 milliseconds. In our intended applications the flow-through rate is OK and the delay range is fine for export delay correction but, perhaps, somewhat limiting for packet jitter correction. NEC have listed devices with twice the capacity and *might* be useable in an iBOB.

4.3.2 eMERLIN Data Import

The remote sender needs a dual, 1.024 Gsps ADC, a reference frequency related to the sampling clock, an iBOB and the network interface. The inputs would be the astronomical data sampled at 1.024 GHz to 8-bits of precision and a TBD timing reference, e.g. a seconds tick. The raw, input data rate would be 16 Gbps and the target output rate is 4 Gbps. Something has to give and the possibilities are

1. Truncate the astronomical data to just 2 bits and send the result leaving further processing to the JB end.
2. *Decimate* the data in time. For example, send only one packet in every four (though preserving the packet numbering as if packets had been sent and lost).
3. *Decimate* in frequency by implementing FIR filters in the iBOB to select parts of the input bands.

The first and second possibilities are the simplest to implement but the beauty of using FPGAs is that the third possibility could be tried if someone wanted to implement the code. Likewise, a DBE personality could be loaded to give Onsala a Haystack-compatible front end at no extra cost. In all cases, timing information from the ticks would be used to tag the Ethernet packets.

At JB, assuming the simplest implementation, the packets would arrive with some timing jitter, be lost or be unreadable. The receiving iBOB has to make sense of all this. Assuming that each packet is identifiable, a packet can be saved at a specific address in cyclically-addressed SRAM, ideally, with the address time related, so that it can be read out at the sending end in proper sequence with the packets adjacent to it.

Missing packets can be handled as follows. If it is not too old, a received packet should be written to its target location in SRAM, a *packet_received* flag for that packet should be set in a FPGA register and the *good_packet_received* counter should be incremented. On read out from SRAM, the appropriate *packet_received* flag should be copied to the data-valid bits. On completion of the read out for a packet, the READ process should clear its *packet_received* flag so that fresh data from a subsequent packet have to be written to those RAM locations for the data read from them next time to be valid.

Reading from the SRAM should only start some time after traffic flow has established some history of arrival times. The read process should start just ahead of where the next packet should go if it were to arrive with minimal delay. The timing to be passed on to the SB is related to the location of the read address in the SRAM. It would help to simplify the implementation if the size of the astronomical data in a packet were to fit nicely into the SRAM without the need for significant reshaping. For example, since the RAM is 72 bits, 9 bytes wide, it would help if the astronomical content of a packet contained an integer multiple of 72 bits.

The format of astronomical data passing from the iBOB to the VSI chips should be *source-synchronous*, i.e. accompanied by its clock, in a format convenient to the implementors. The VSI chips need do very little or nothing to these data so they could be passed straight through to the SB input multiplexer.

4.3.3 eMERLIN Data Export

The case of data export is a little more difficult. The steps involved are

1. Optional data stream selection.

2. Optional frequency conversion.

3. Optional filtering.
4. Delay correction both coarse and fine.
5. Optional data stream selection.
6. Packet formation and transmission.

Fortunately, most of these are straight forward and have been done in the SB, the DBE and the DBBC and help is available in the Xilinx System Generator tool. Since the two VSI chips have more processing capability than the FPGA in the iBOB, the filtering operations and the fine-delay FIR interpolation filters should be implemented in those. The iBOB FPGA can handle the coarse delay, control of the fine delay in the VSI chips and packet handling.

Control should be done by the embedded PowerPC. It can receive the delay model over Ethernet, handle the control network channel and control the hardware. The coarse delay would be handled in the ZBT SRAM which can cope with delays up to 9 milliseconds for data at 4 Gbps. Delay correction would be done by delaying the signals by, say, 5 milliseconds minus the already-applied model delay. The net result would be signals delayed by a constant 5 milliseconds which the correlator would handle as a fixed clock error.

5 Conclusions

With the iBOB concept of driving 4 Gbps (or more) of data over 10GE links we can match the multiple iBOB data sources with commodity networking technology to send the data in various ways:

1. In real-time "connected" fashion as required by the Onsala—eMERLIN—JIVE 4 Gbps demonstration.
2. In connection to commodity disk subsystems, for non-real-time storage of VLBI recording for later networked transmission to a correlator center.
3. In connection with commodity 10GE/1GE local networking equipment and/or 10GE-equipped commodity computers to explore the boundaries of commodity PC nodes in VLBI data processing.
4. In real-time "distributed" fashion for distributing the VLBI data over a set of software correlator processing nodes which may be physically residing in either a single central location or multiple locations, distributed in grid fashion.

This “mixing and matching” allows us to adapt the same VLBI DAQ technology (direct RF-to-net) to different post-processing schemes. Combined with capability to fully use emerging commodity networking technologies to scale bandwidth by adding iBOBs, network connections and equipment, and processing nodes, this adaptability will keep the direct RF-to-net concept useful and appealing for a longer time than architectures based on special interconnects.

References

[1] http://www.haystack.mit.edu/tech/vlbi/mark5/index.htm

[2] http://kurp.hut.fi/vlbi/instr/

[3] “dBBC: Status Report”, Gino Tuccari, http://www.mpifr-bonn.mpg.de/div/vlbicor/tog_chair/togreps06/DBBCrep.pdf

[4] http://www.haystack.mit.edu/tech/vlbi/digital/index.html

[5] “Report on FABRIC Month 7 Demonstration ‘eVLBI Fringes with PCEVN’”, John Conway, Michael Lindqvist, Jan Wagner, Guifré Molera, Jouko Ritakari, Ari Mujunen, Paul Burgess, Nico Kruithof, Huib van Langevelde, Zsolt Paragi.

[6] http://kurp.hut.fi/vlbi/month7/

[7] “EXPReS JRA1 FABRIC Data Acquisition Requirements”, Ari Mujunen and Jouko Ritakari, EXPReS Wiki at: http://www.jive.nl/dokuwiki/doku.php/expres:fabric

[8] “Haystack trip report”, Chris Phillips, http://www.atnf.csiro.au/lists/vlbiobs/2006/07/0577.html

[9] BEE2 datasheet, http://BEE2.eecs.berkeley.edu/

[10] “An Image-Reject Downconverter with Sideband Selection for Double-Conversion Receiver”, Kari Stadius, Petri Järviö, Kari Halonen, Petteri Paatsila, http://ieeexplore.ieee.org/iel5/9920/31529/01471327.pdf

[11] “Tsunami: A High-Speed Rate-Controlled Protocol for File Transfer”, Mark R. Meiss, Indiana University, http://steinbeck.ucs.indiana.edu/ mmeiss/papers/tsunami.pdf

[12] “Power, accuracy and noise aspects in CMOS mixed-signal design”, Sanduleanu, Mihai Adrian Tiberiu, 1999, ISBN 90-3651265-4.

[13] “Packet Loss in High Data Rate Internet Data Transfer for eVLBI”, R. Spencer, R. Hughes-Jones, A. Matthews and S. Toole

[14] “eVLBI—Grid Design Document”, PSNC, EXPReS Wiki at: http://www.jive.nl/dokuwiki/doku.php/expres:fabric

[15] “Export/Import of VLBI Data from/to eMerlin”, Bryan Anderson, EXPReS Wiki at: http://www.jive.nl/dokuwiki/doku.php/expres:fabric

[16] “Efficient wideband digital correlation”, Brent R. Carlson and Peter E. Dewdney, Electronics Letters, vol. 36, no. 11, 25 May 2000, pp. 987-988.

[17] “A proposed WIDAR correlator for the EVLA Project”, Brent Carlson, http://www.aoc.nrao.edu/evla/geninfo/memoseries/nrc_evla_memo001.pdf