

THE METSÄHOVI SOLUTION FOR GIGABIT VLBI

J. Ritakari, A. Mujunen¹

¹Metsähovi Radio Observatory, Helsinki University of Technology

Metsähovintie 114, 02540 KYLMÄLÄ, FINLAND

E-mails: jr@kurp.hut.fi, Ari.Mujunen@hut.fi

Abstract

Metsähovi Radio Observatory has developed a PC-based data acquisition system for VLBI. The system is based on the VSI standard interface [1] and is capable of recording any VSI-H compliant data source, including both new Japanese gigabit VLBI sampler data and existing VLBA or Mark IV data. Because of board simplicity and low cost, and because of the use of standard PC hardware and low-cost IDE hard disks, even high-volume, high-throughput data acquisition systems can be realized at exceptionally low cost.

1 INTRODUCTION

VSIB data I/O PCI board makes it possible to use conventional Linux PCs in continuously streaming high-throughput data acquisition and playback applications at data rates of 512 Mbits/s/PC and beyond. Furthermore, it is well-suited to data acquisition systems consisting of multiple PCs running in parallel and/or chained, resulting in both aggregate data rates of multi-Gbps and total on-line data volumes of several terabytes.

We have made special effort to design the system to be compatible with the Mark5 system that is being developed at Haystack Observatory.

2 DESIGN PHILOSOPHY

We are using a fundamentally different design philosophy than the other teams that are developing disk-based data acquisition systems.

Early in our project we noticed that the common PC technology has several bottlenecks that limit the performance in data acquisition and transfer:

- 32-bit PCI bus speed limits the performance to 600-700 Mbit/s
- Disk subsystem speed limits the performance to 500-700 Mbit/s
- Gigabit Ethernet speed limits the performance to 600-700 Mbit/s

While the other teams are designing single-PC solutions for Gbit/s data acquisition, we try to get the maximum performance from a commodity PC and build a scalable system. If the performance of a single unit is not one Gbit/s, that is no problem. We can use two units or three.

3 THE VSI I/O BOARD FOR PC (VSIB)

The VSI input board (illustrated in accompanying Figure) is a low-cost 32bit/33MHz PCI interface that stores the incoming VSI data stream to microcomputer main memory with bus-master DMA.

Data acquisition uses huge circular buffers in the PC main memory and can operate without main processor intervention.

The most important feature is that the VSIBs have two bidirectional VSI ports and can be daisy-chained to capture high-speed data. In this way the maximum sustainable speed of one computer does not limit the speed of the system.

VSIB has been designed as a standard half-size 32-bit, 33 MHz PCI expansion board. It is compatible with both 5V and 3.3V bus signalling with its “universal” dual-slot PCI card edge connector and it can thus be used also in most 64-bit, 64 MHz PCI slots.

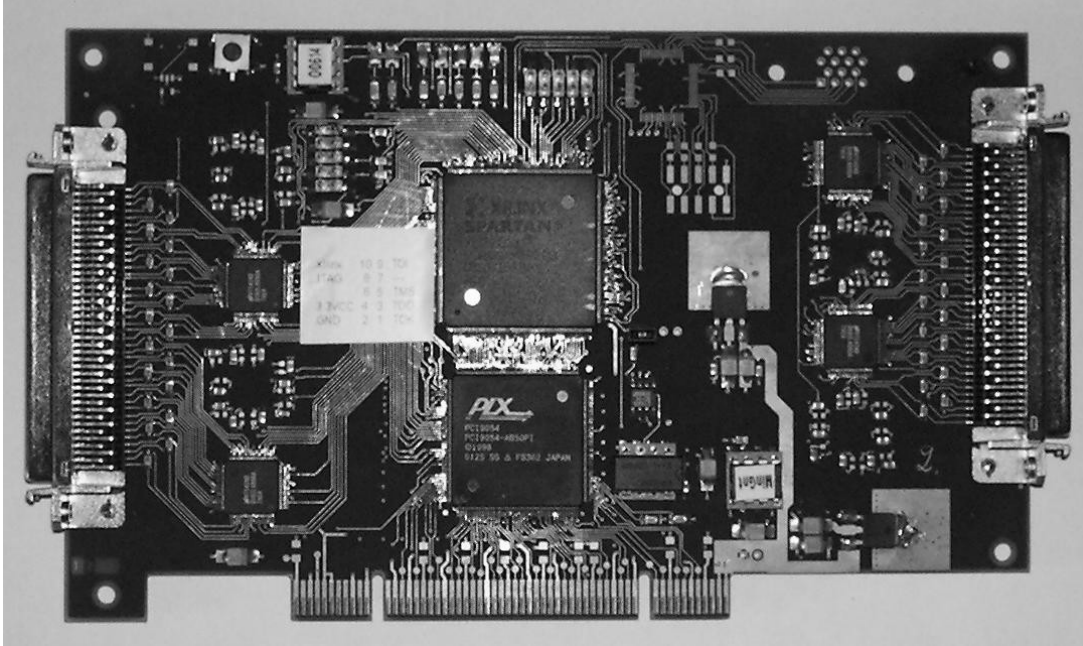


Figure 1: The VSIB data I/O PCI expansion board.

3.1 On-Board Logic

Differential LVDS signals are processed with bus LVDS transceivers terminated with 100Ω resistors at both ends, ensuring that cable connections are always correctly terminated. A Xilinx FPGA provides signal routing and processing, and a 4 kB buffer FIFO memory between VSI and PCI bus interface, ensuring continuous data flow regardless of PCI bus latencies.

The Xilinx logic allows selection of all 32 VSI data bits, or a subset of 16 or 8 (VLBI “Mark 5A” compatibility). It also allows skipping VSI input data words with a counter in the range of 1..65535. Both of these features can be used to direct a single VSIB to process a subset of the whole VSI data stream. For instance, four VSIB boards can be set up to repeatedly “demultiplex” four consecutive VSI data words to four VSIB boards in a chain, effectively reducing the data rate requirements for a single PC host to a quarter of the original data stream.

3.2 PCI Bus Interface

For connecting to any 32-bit, 33 MHz PCI expansion bus, VSIB utilizes a PLX bus-mastering interface chip. The chip enables high-throughput scatter/gather bus-master DMA transfers to and from PC main memory. Depending on motherboard chipset performance, VSIB has been demonstrated to sustain data transfers in excess of 700 MBits/s.

PC main memory, arranged in ring buffer fashion, is used to allow VSIB to operate in conventional Linux operating environment without the need for any special real-time extensions. PCI bus interface hardware takes care of depositing data in large main memory ring buffer where Linux software can find it at its own pace.

4 UNIVERSAL DUAL 40-PIN CABLE TO VSI CONVERTER (VSIC)

When we designed the VSI input board we needed a test vector generator and a signal level converter to capture real VLBI data.

We noticed that VLBI uses lots of cables with different pinouts and data polarity, but almost all cables use the same 40-pin flat cable connectors and differential RS-422 or ECL signalling. The new RS-422 line receivers have a common mode voltage range that is large enough to capture ECL signals, so we designed a universal VLBI cable to VSI converter that can interface to most of the cables used in VLBI.

The VSIC can be connected to Mark IV or VLBA formatter outputs or directly to VLBA sampler outputs. If the 1PPS marker that is needed in VSI is not available (for example in the formatter output cables) the VSIC can

regenerate the 1PPS signal from the information in the frame headers.

5 PROJECT STATUS

At this moment prototype batches of both VSIBs and VSICs have been manufactured and tested.

Data capture has been extensively tested and 512 Mbit/s speed (for one unit) seems to be fully operational, which means that sustained 1 Gbit/s operation is possible with only two microcomputers.

Data playback has been tested at the JIVE correlator and seems to be fully working at speed of 256 Mbit/s per PC.

The documentation (schematics, frequently asked questions and PostScript files of the board layout) are available at "<http://kurp-www.hut.fi/vlbi/instr>".

REFERENCES

[1] <http://web.haystack.edu/vsi/index.html>