True off-the-shelf Mark 5

First draft

Jouko Ritakari, Jouko.Ritakari@hut.fi

Metsähovi Radio Observatory

August 28, 2001

Summary

A cost-effective Mark 5 disk recording system can be built from components that are available from the nearest computer store.

It is important to respect the limitations of the current off-the shelf PC technology to obtain the simplest possible scalable system, minimize the development work and get the best price/performance ratio.

The most important advantages of this approach are scalability, flexibility and independence of the PC and disk technology. You can have a reliable 256 Mbit/s recording system with one PC only. If you want to upgrade it to gigabit per second capability you only add three PCs. You don't want to clutter the correlator room with 64 PCs, no problem. You use eleven or sixteen PCs and play back the data at quarter speed. You want to use slow speed for spectral line experiments and use Internet to transfer the data to the correlator. Again no problem, the data is in normal files in a Unix computer that is connected to the Internet.

Off-the shelf technology limitations

The office PC technology has several limitations when applied to real time data storage.

The main limitations are the speed of the PCI bus, the speed of the data path to disks (in this case the ATA 5 –interface) and the speed of the disks themselves.

Ari Mujunen has shown in his paper "The Sustained Disk Streaming Performance of COTS Linux PCs" (Mark 5 memo #004) that a normal PC with unmodified Linux operating system can sustain data storing at the speed of 256 Mbit/s.

PCI bus speed

Until now the speed of the PCI bus has been a limiting factor. The theoretical limit of the 32-bit, 33MHz bus is 132 MB/s (1056 Mbit/s) and it is a shared resource. Clearly this is too slow for sustained gigabit recording and even at the speed of 512 Mbit/s the data needs to be transferred directly from the input card to the disk controller. Normal operation (data from input card to main memory and from the main memory to disks) is limited to 256 Mbit/s at most.

However, the PCI bus speed limitation is no longer valid.

Most of the current server-class PC computers already have 64-bit PCI buses. For example, the Dell Poweredge 500SC has three separate PCI buses, two 64-bit 66 MHz buses with one slot each and one 32-bit 33 MHz legacy PCI bus with three slots. The specifications and brochures of this particular server can be found at www.dell.com. Additionally, in many new computers the ATA disk interfaces have been decoupled from the PCI bus so they do not compete for the same resources.

The new server class computers are not very expensive, for example the pricing of the Dell Poweredge 500SC starts from 699 dollars.

ATA disk interface and disk speed

Ari Mujunen has tested the streaming performance of the COTS PCs and has found out that at speeds below 300 Mbit/s the hard disk is clearly the dominating bottleneck, not the CPU, memory, operating system overhead or user software.

That means that there is no advantage in using custom formats in storing the data.

With two disks it is possible to sustain 256 Mbit/s speed and with three or four it is possible to get more than 300 Mbit/s speed, but it will be difficult to get the 512 Mbit/s speed with only one completely off-the-shelf computer.

Gigabit recording – a non-issue ?

The specifications of the Mark 5 have been to a great extent derived from buzzwords like "gigabit recording" and "24-hour challenge".

Let’s calculate what these mean in terms of the technology we have now.

- At this moment, largest ATA disks are 100 GB.
- 100 GB means 800 Gbit.
- At gigabit per second speeds 800 Gbit disk contains 800 seconds (about 12 minutes) of data.
- One station needs five disks per hour.
- An eight-station 24-hour experiment needs 960 disks.

We can clearly see that the operation of the network in the near future will be limited by available disk space.

Affordable gigabit recording

In the previous paragraphs I have shown that the current off-the-shelf technology limits the recorder to 256 Mbit/s speed and 400 GB disk capacity, if the recorder uses all the four ATA disks for data. The recorder doesn’t really need a system disk, it can boot itself from the field

system PC.

This means that one recorder can hold 3.5 hours of continuous data at 256 Mbit/s, clearly less than what is desired. Of course, next year the size of the disks will be doubled, but that's another story.

The simplest and most cost-effective way to expand the storage capability is to have two or four identical PC recorders for each station. After all, these server-class microcomputers cost only 700 dollars each.

If we have four recorder units at every station, it is very easy to use these in parallel and have gigabit per second capability. We only distribute the data to all the four computers and each computer records one fourth.

Time-multiplexed gigabit recording

However, it makes more sense to time-multiplex the data to several computers (record data with several computers that record different time-slices). If each computer has a gigabyte memory buffer, it can acquire eight seconds worth of gigabit data in real time. When the eight seconds are up, the next computer starts to acquire the data and so on. The starting and stopping of data acquisition can be controlled by the 1PPS pulse from the VSI interface so the computers need to be synchronized at one second accuracy only.

The beauty of this scheme is that the playback units at the correlator will be very simple. We just plug in all the hard disks from the four recording computers to one playback computer (that possibly has an extra ATA disk controller) and play back the data at 256 Mbit/s that the playback computer is capable of sustaining. No complicated synchronization is needed between several parallel playback computers and we avoid cluttering the correlator room.

Data and file formats

Disk-based recording has different problems and limitations than tape recorders. Most of the things the formatter does (fan-in, fan-out, parity bit insertion, CRC checks, barrel-rolling) serve no useful purpose in disk recording.

In fact, tape-specific formatting can be considered to damage data.

Clearly the best option is to record raw sampler outputs, synchronize the data-taking to 1PPS pulses and store the data into normal files in the Unix file system. This way the bookkeeping is easy and the data is available to local processing or FTP transfer to the correlator.

The disk-based recording has one considerable advantage over tape-based recording: auxiliary data (time information, station information, experiment name etc.) can be recorded into same files as data, which simplifies the logistics over the old plain-data-on-tapes and different log files –approach.

The data format could be based on the well-known FITS format, auxiliary data is recorded at the start of the file in plain old ASCII.

Optimally the file naming doesn't matter. When the disks are inserted into the playback computer, the computer reads the auxiliary information from the start of each file. If we are using gigabyte-sized files and 100GB disks, the time needed to read all the auxiliary data (a few blocks from the beginning of 100 files) is negligible.

Playback at correlator

So we have managed to record raw sampler outputs in normal Unix files and we have transported the disks to the correlator. Or transferred the files with FTP, if you like.

Of course the correlator cannot accept sampler data. We must format the data.

Fortunately formatting during playback is a relatively simple and straightforward process.

In the case of data replacement format, first we read the data into a large (gigabyte-size) circular buffer in computer main memory. Each bit of the 32-bit word contains data from different bit stream. Then we simply replace the appropriate words of the data with headers.

Fortunately, most of the header data is the same for every track, so only words containing 0x00000000 and 0xffffffff need to be written. The exception is the auxiliary data containing track numbers etc., but it can be precalculated.

When the data has been appropriately formatted it is played to the correlator with DMA. External clocking of data is preferred because it can keep even multiple PCs perfectly synchronized.

In the case of data non-replacement format the process is almost as easy. The DMA of the modern PCI chips can be programmed to a so-called scatter/gather mode (a chain of DMA descriptors tells the hardware the locations and lengths of data to be transferred). The headers can be all in one memory area and data in another area, the hardware interleaves them during the DMA transfer.

Another approach would be to precalculate all the headers needed for the experiment and store them in a NFS server. The playback computers could read the headers, build the DMA descriptor chain and we would have instant formatting.

One word of warning: it is probably best that the hardware inserts the parity bits into the playback data. The procedure of calculating the parity is very simple (XORing eight 32-bit words with each other) but the processor would need to move all the data to get space for parity words which could be time-consuming.

Development work needed

VLBA to VSI converter

We need a module to distribute the raw sampler outputs to several computers, if we want to use more than 256 Mbit/s speed or have long recording times.

For these purposes, VSI is as good an interface as any. It has 32 parallel data streams, a 1PPS signal and clock. There are several levels of VSI compliance, our converter is fully VSI-H Level

B –compliant.

I have made a preliminary study what a VLBA to VSI converter would be: a four-layer board, size 20cm*20cm, that is installed behind a panel in a 19” rack.

The schematics and preliminary routing of the board can be found at kurp.hut.fi/vlbi/instr/VSIconverter/VLBAtoVSI.html . Please note that some components (bypass capacitors, regulators etc.) are intentionally missing to speed up the experimentation.

The converter inputs the sampler outputs (differential ECL) from the VLBA samplers, reclocks the data and outputs it to two identical VSI connectors.

If we want to distribute the data to more than two computers we can either build a board that has four identical VSI connectors or use two boards of this kind. Then we would distribute the outputs from the first sampler to two VSI connectors and the outputs of the second sampler to two more computers.

A Mark IV to VSI converter is quite similar. All the same signals are available inside the Mark IV formatter and there is ample room inside for a converter board.

Using VSI as a recorder interface would have one additional benefit: we would have instant compatibility with the Japanese Gigabit VLBI equipment and could very easily upgrade to direct IF sampling if we want to.

VSI input module for 64-bit PCI bus

I have evaluated several possibilities for the design of the input module. These include modules with ping-pong buffers in dual-port RAM, ping-pong buffers in two ZBT SRAM modules and several FIFO-based architectures.

The most promising design so far uses a Plxtech PCI 9656 I/O accelerator chip with a Xilinx programmable logic chip.

The PCI 9656 is intended for 64-bit 66 MHz PCI bus cards. It is downwards compatible with the 9054 chip that has existed for several years, for example the local bus structure and the programming interfaces are the same. The chip is available in sample quantities and a rapid development kit will be available during Q2/2001.

In addition to the PCI interface the board needs some logic for the state machines and a small high-speed FIFO. Both of these can be built with a single PLD.

I have programmed a prototype VHDL program for a middle-sized Xilinx chip, model Spartan II XC2S100, that has about 2000 logic cells and 50 kB of high-speed block RAM that can be used for the FIFOs. The program used only 10% of the logic cells, all the block RAM and 75% of the global clock signals.

If the FIFOs on the Spartan II chip are not enough, it is very easy to add an external FIFO chip between the PCI 9656 and Xilinx chips.

The capabilities of the prototype VHDL program are the following:

- Data capture starting from the next 1PPS signal.
- Programmable clock divider: data can be captured at 32, 16, 8…, 0.125 Msamples/s. If even slower speeds are needed, the main processor can record only part of the samples in main memory.
- Compatibility mode for old correlators: parallel 32-bit capture from all lines. If all the data is not needed, the main processor can discard part of the data during recording.
- Data separator mode for testing or near real-time correlation: data from eight one- or two-bit data streams can be separated into separate bytes and rearranged into separate files during the recording.

Conveniently, both PLXtech and Xilinx provide prototyping boards for their products. A prototype input board was constructed by connecting these boards together with flat cables. (Please see the photo at the end of this article.) The prototype is built with the older PCI9054 (that has the same local bus as the 64-bit PCI9656 chip) and ECL input buffers for direct connection to VLBA samplers, but it demonstrates the possibilities of this approach very nicely.

Correlator output module for 64-bit PCI bus

This module shares most of the design of the VSI input module. Data is stored into a large (gigabyte size) circular buffer in computer main memory and is transferred to the correlator with scatter/gather DMA. The same Xilinx chip that was used in VSI input module to capture data is used here to insert parity bits. Xilinx block RAM is used as high-speed FIFOs to buffer the data.

Main differences between the VSI input module and the correlator output module are the level converters and connectors. Instead of VSI connector and LVDS drivers the correlator output module has something else, probably RS-422.

It could be possible to use the same module for both purposes. After all, the board has lots of room for extra components and the Xilinx chip has lots of unused I/O pins. Of course only the bidirectional VSI connector will fit in the back panel of the card, the legacy connectors must be mounted elsewhere. I can see problems only if external FIFOs are needed.