6 December 1999

# SHARC II Data Acquisition Computer

C. Darren Dowell, Walter Collins, Attila Kovacs

*Caltech*

## ABSTRACT

The SHARC II acquisition system consists of a VME chassis, an UltraSPARC CPU running Solaris, a quad TMS320C62x DSP board, and several I/O cards. The computer interfaces with the 256-channel A/D system of Goddard heritage and uses the DSP's for real-time demodulation of the data. Communication with the antenna computer is via Ethernet, with a reflective memory card as a possible real-time option. Control and monitoring of the chopping secondary are accomplished via simple digital and analog I/O. Low-level DSP access uses the SwiftNet routines and C code. High level acquisition software is written in JAVA or possibly C. The Goddard-developed IRC software is used as much as possible. To observe, the user connects via Ethernet to the acquisition computer to access a scriptable command line. Real-time monitoring of the data occurs via X Windows display on a remote computer. Data is stored on a local hard drive during acquisition and is backed up on a CD-R drive nightly. The dewar is monitored and controlled via a GPIB or serial interface to a Neocera LTC-21.

## 1. REQUIRED OBSERVING MODES

Three observing modes are required for optimum SHARC II usage: pointed on-off observations with telescope tracking (CHOP-SLEWY), controlled scans of the telescope with tracking (On-The-Fly), and drift scanning. Each observing mode makes different requirements of the telescope and antenna computer, but in general what is required is a method by which accurate positions and times can be recorded with each data frame. An interface with the chopping secondary is also required.

### 1.1. On-Off Integration

For the on-off (CHOP-SLEWY) mode, the telescope tracks a source on the sky automatically, and the acquisition computer need only be concerned with occasional adjustments (”offsets”) commanded by the acquisition computer. The basic observing strategy consists of two offset positions, usually with an azimuth separation, observed in a Left-Right-Right-Left sequence. The separation distance is equal to the throw of the chopping secondary or the distance between differenced pixels in the bolometer array.

The observing procedure for one **nodpair** cycle is:

1) The acquisition computer commands the antenna computer to move the telescope to the Left offset position. During the slew time, samples are recorded, but are flagged as invalid either in real time or following the integration. The invalid samples are saved, however, for possible data filtering or frequency analysis.

2) Valid samples in Left beam are acquired for roughly 5 seconds. (Integration time is programable.)

3) The telescope is slewed to the Right beam while the acquisition computer saves invalid samples.

4) Valid samples are acquired for *two* integration periods in Right beam.

5) The telescope is slewed to Left beam.

6) Vaild samples are acquired in Left beam for one integration period.

7) The recorded samples are flagged (valid or invalid) and written to disk.

For convenience, a specified number of on-off cycles may be written to one file. An automatic 'dither map' procedure should be implemented.

### 1.2. OTF Mapping

For On-The-Fly observing, the telescope tracks the source in an average sense. However, a continuous stream of offsets are applied to the telescope in order to sweep the telescope across a source. The scans are constant speed in a single direction, usually AZ, but could also be some other axis. Prior to starting the continuous scan, the telescope 'overshoots' the initial position so that it can ramp up to the target speed. Typical speeds are of order $10''$ /second. On-The-Fly mapping can be performed with and without chopping.

The observing procedure for one **OTF** scan is:

1) The acquisition computer commands the antenna computer to initiate a scan with the telescope. During the slew time, samples are recorded, but are flagged as invalid either in real time or following the integration. The invalid samples may be omitted from the data file.

2) Valid samples are acquired for the duration of the scan.

3) Some invalid samples are acquired following the scan. They may be omitted from the data file.

4) When the acquisition computer receives a flag from the antenna computer that the scan is complete, samples are no longer recorded.

5) The valid samples are associated with positions and written to disk.

Each individual scan will be written to its own file. Multiple-row On-The-Fly maps will be controlled via observing scripts.

### 1.3. Drift Scanning

For drift scanning, the telescope is parked at a fixed position, and the sky drifts by at a rate of $15cos(DEC)$ arcseconds per second of time. This in principle gives very good scan accuracy and also is the method with the least pickup of unwanted signals. Scans sweep across arcs in RA (constant DEC), so it may be desirable to rotate the dewar to align the detector array with RA and DEC.

During a drift scan:

1) The acquisition computer commands the antenna computer to acquire the AZ/ZA that a specified source will pass through at a specified time. A duration for the scan is also sent to the antenna computer after which source tracking resumes. Recording of samples begins at the time the request is sent.

2) Samples, first invalid, then valid, then invalid are acquired until the acquisition computer receives a flag from the antenna computer that the scan is complete.

3) The valid samples are associated with positions and written to disk.

Each individual scan will be written to its own file. Multiple-row drift maps will be controlled via observing scripts.

## 2. CHASSIS AND CPU HARDWARE

The computer **enclosure** is Elma Electronic, Inc. part number 12V-0716-RV12J12A-P750-A, which is a 12-slot VME backplane in a rackmount enclosure to accomodate 6U $\times$ 160 mm VME boards. The connectors are 96-pin. The power supply is 750 watts, oversized due to the altitude. Integrated fans provide bottom to top airflow. Space is provided for two full-height or four half-height peripherals. The AC power has a switch and fuse. The cost was $2804, and it should be delivered by Feb. 1, 2000.

The **CPU** is a Themis Computer model USPIIi-1V/1-1024-256-440, which contains a 440 MHz UltraSPARC IIi processor, 256 Mbytes DRAM, 1 Mbyte second level cache, a single 10/100BaseT Ethernet port, dual Ultra Fast/Wide SCSI ports, two serial ports, a keyboard/mouse port, and a

VME64 interface. The cost was $9860. We have ordered a PS2-type **keyboard** and **mouse** at a cost of $47. We have also purchased an **I/O kit**, Themis model INT-KIT-USPIIi-1V/1-VME, at a cost of $340. This mainly sounds like cabling. The expected delivery date for this hardware is by Feb. 1, 2000.

We have not ordered a **monitor** to accompany the CPU. We have been told by Themis that we can connect a dumb terminal to one of the serial ports, so we will use that option for CPU debugging. **QUESTION: How exactly do we make this work? Are there spare dumb terminals around compatible with the CPU card, or do we need to order something? Graphical feedback to the SHARC II user will be via remote X Windows displays.

## 3. OPERATING SYSTEM

The CPU will run **Solaris**. We have ordered Themis Computer part number SOL-7.0-USPIIi1V-DU-PKG at a cost of $485, which is Solaris 7 in a 64-bit environment. VME drivers are included.

The CPU needs to support X Windows and JAVA.

The CPU clock should be regularly synchronized with the accurate antenna computer clock. **QUESTION: Via Network Time Protocol?

## 4. DATA STORAGE

We have purchased a 18Gbyte **SCSI hard drive** from Insight Direct, their model number ST318275LW. It is a Seagate Barracuda 7200 rpm Ultra-2 SCSI unit with a low profile (1"), 80 Mbyte/sec maximum transfer rate, and 6.9 msec average seek time. The cost was $480, and it was delivered in November 1999.

A **CD-R** (writable CD drive) is desired for data backup. At this time, we have not ordered a dedicated drive since one is available at the CSO. It is a Yamaha CRW4416SX (4X write, 16X read) and is attached to the Sun called hapuna. CD-R drives with 8X speed are under $500, and the media costs $2/650 MB. It would take about 10 minutes to write a full CD. DVD writers are $800, and the media is $35/2.6 GB.

## 5. A/D AND DSP

Goddard designed a 256-channel A/D and DSP system for now-defunct Kuiper Airborne Observatory projects which has been copied by U. Chicago and Caltech for ground-based projects. The DSP card uses 8 TI TMS320C50 DSP's running at 20 MHz, but unfortunately it is NuBus form factor, so have decided to replace it. The **A/D** design is salvageable; we have built up an A/D system of the Goddard design to 128 channels currently, expandable to 256 channels.

The replacement **DSP** system is a Pentek, Inc., model number 4290-244-304, which contains four TI TMS320C6201 DSP's on a VME64 board. The processor chips are Revision 3 running at 200 MHz. The software that we bought to accompany the DSP board are Pentek Swiftnet (drivers), part number 4945-004, and Pentek Readyflow (C routines for applications), part number 4999-290, both running under Solaris. We also purchased TI Code Composer for Windows NT, Pentek part number 4977-100, to write code for the DSP's. The cost of the DSP card and software were $15995 and $5495, respectively. Delivery is expected by March 1, 2000.

The A/D chassis contains logic circuitry to stack the serial streams from 32 A/D channels onto a single fiber optic cable. (8 total fibers are needed for the 256 channel version.) The NuBus DSP card contains an FPGA chip (Actel A1020) to decode the fiber signals to (three-conductor) serial data fed directly into the DSP chips. A fourth signal is sent to an interrupt pin to signal the start of a data block. The Actel chip was programmed by Goddard so is from our perspective a 'black box'; however, we use this same chip to interface the fiber optic signals to the new DSP's since they have a direct serial interface. The decoder card plugs into the VME backplane for power, but the fiber interface and serial interface to the DSP card are connectors on the front panel.

## 6. SIGNAL DEMODULATION

## 7. ANTENNA COMPUTER INTERFACE

Ethernet will be a required interface between the antenna and acquisition computers. Although not real time, Ethernet permits a simple interface with potentially complex commands. The **obs** commands to the antenna computer will be sent over Ethernet.

Another possibility for the interface is reflective memory, although we may not be able to take full advantage of the real-time capability due to our choice of operating system. **QUESTION: Do we want the reflective memory?

## 8. IMPLEMENTATION OF OBSERVING MODES

On-off observing is the easiest observing mode to implement, the reason being that the telescope is pointed at a fixed position on the sky during the integration. At the sacrifice of a small amount of observing time, Ethernet could be used to signal the valid and invalid samples. (See Section 1.1.) Valid samples begin when the acquisition computer receives a reply from the antenna computer that the telescope is in position. Valid samples end when the acquisition computer requests a new offset position. For greater efficiency (and accuracy), one can use the method that will be employed for on-the-fly observing and drift scanning.

On-the-fly and drift scanning are more difficult to implement with accuracy since the telescope is constantly in motion with respect to the sky. Since we do not have a real-time operating system for the CPU, we should not rely on it for coordinating samples and positions. That leaves the DSP card and/or off-line association of samples and positions as the available possibilities.

To encode the telescope position with digital I/O to the DSP card would require of order 46 bits (the number of bits in the telescope encoder) which is unwieldy. An alternate solution would be for the antenna computer to send instead a counter value to the DSP, where the counter value corresponds to an index in a list of telescope positions stored by the antenna computer. Assuming 100 second integrations and 100 Hz resolution, 10000 counter states – i.e., 14 bits – would be necessary to avoid counter wrap around. This is a more reasonable number of bits. The number of bits could be reduced somewhat by occasional status checking over Ethernet to break degeneracies. A third solution is to synchronize the DSP and antenna computer clocks and to again use the list of telescope position. **QUESTION: What procedure is used to synchronize the clocks? A digital line from the antenna computer to the DSP that ticks, say, every 10 seconds? Network Time Protocol to synchronize the CPU clock, and then a statistical approach to synchronize the DSP?

The list of telescope positions vs. time or index that we will use can be transfered from the antenna computer to the acquisition computer in several ways. One is to use reflective memory if available and to construct a queue. Another is to save the list of telescope positions to a disk file to be read by the acquisition computer at the end of the integration.

**Recommended strategy**: synchronized clocks between the antenna computer and DSP computer, accomplished by a digital line. Antenna computer records a list of position vs. time stored on disk, available to acquisition computer.

**QUESTION: What information regarding position do we actually want? For precise on-off integrations, a pair of offsets (FAZO/FZAO or RAO/DECO) showing the actual position of the telescope with respect to the nominal boresight would do, or alternatively, the requested offset pair plus the error terms (EAZ or EZA). For on-the-fly, the actual telescope offsets from the boresight should be recorded. These offset are most likely AZO/ZAO but could be RAO/DECO. For drift scanning, perhaps the most useful information would be the RAO relative to the nominal source currently pointed to by the nominal boresight of the telescope. Or perhaps the time until transit of the nominal source through the nominal boresight. We should also record the EAZ and EZA.

## 9. CHOPPING SECONDARY I/O

It is desirable to have the A/D sampling synchronized with other modulations occuring in the system, such as secondary mirror chopping or AC biasing. Since the A/D only generates a frequency reference, modulation frequencies are necessarily controlled by it. Both the chopping secondary and amplifier electronics allow external frequency references.

The typical chopping frequency is 3 Hz, and the typical AC bias frequency is 30 Hz. Therefore, the 1 kHz frequency reference from the A/D needs to be divided down, either with direct hardware on the fiber optic decoder card or in DSP 'software'. The duty cycle of the divided down signal must be 50%, which probably means that the available frequencies are given by 1000 Hz/2N, where N is a positive integer. **QUESTION: How do we want to do this?

## 10. AMPLIFIER PROGRAMMING I/O

The SHARC II amplifiers require a fairly hefty amount of digital control. Each amplifier has an associated DAC channel for subtraction of large common mode signals. Each DAC is a quad chip, so it takes approximately 12 digital lines per 32-signal amplifier module for DAC programming. Additional control lines include a gain select common to all amplifiers, and a few digital lines for remote integration control near the instrument. For the full 12×32 array, of order 150 digital I/O lines will be required.

## 11. GENERAL OBSERVING PROCEDURE

The acquisition computer is located in the AOS room of the observatory. The observer sits at a console in the control room, probably a SUN, and logs into the acquisition computer with **ssh**. To discourage using the acquisition computer for other purposes, there is only common user account. At the Solaris prompt, the user starts the acquisition program. One of the very first actions is a query for a user name, which loads custom configurations such as catalogs, window layout, observing scripts, and analysis preferences. The user name is a sticky parameter, so the previous user name can be entered simply with a *return*. The custom configurations can be edited at any time during the operation of the program. **QUESTION: Where will source catalogs actually reside – UIP, antenna computer/OBS, or acquisition computer?

The acquisition program uses X Windows to display the control and status information on the user's console. The X Windows should display automatically and not require the user to 'setenv DISPLAY', etc. If possible, color table problems should be avoided. (For example, the heterodyne backend processes fail if the host they are displaying to is running Netscape.)

Source acquisition can be done either with the acquisition program or with the UIP. Pointing adjustments should also be allowed from both platforms.

The basic control interface to the acquisition computer is a command line. However, common commands should also be available in GUI form. GUI commands should also print in the command line. The commands are distributed to the various hardware components as necessary, using **OBS** for antenna computer instructions.

Data files are stored in one master directory containing subdirectories organized by date. File prefixes are selected by the user, but extensions (suffixes) are set to incrementing numbers.

## 12. OBSERVING LOG

### 12.1. Data Files

As much information as possible should be stored with the data files – time, source position, offsets, tilts, secondary focus, weather, pointing model, etc.

### 12.2. Observing Record

A copy of the commands executed during each observing session should be recorded. This log file should be stored in the same directory as the data files. For one coherent record, exclusive use of the SHARC II software for telescope control (as opposed to the UIP) is advantageous.

### 12.3. Global Log

A sparse log of observations with a fixed file location should be recorded. This will be useful for searching for sources observed, pointing, focus, etc. among several nights and/or users. This is similar to the CSO heterodyne database. Two new entries should be made for each source observed – one for the source left and another for the new source acquired.

## 13. ANALYSIS PIPELINE

There are at least three analysis modes: pointing, quick look, and full. Following is a description of each:

1) Pointing. The data are reduced just enough to generate a map in the region where the source is expected. Most likely, pointing information will be an on-off map consisting of one or more nodpairs. The pointing analysis software should include a graphical interface which shows the map and performs a centroid with a mouse click.

2) Quick look. The map is reduced to the point that the user gets some idea of the signal-to-noise. CPU intensive tasks such as filtering are skipped. We might even consider skipping the NOD2 reconstruction. Quick look mode for on-off integrations may be essentially the same as pointing mode, so they may not be two separate modes.

3) Full analysis. Sophisticated techniques are used for the analysis – NOD2, Emerson2, iterative

sky noise subraction, etc. This may take a while, so it should be done off line while the observer is acquiring more data. Hopefully, we will be in the situation that it takes no longer to analyze data than it takes to acquire it.

The analysis pipeline should be as automatic as possible so that the user does not have to type extra commands and then wait for the software to execute.

## 14. DEWAR MONITORING AND CONTROL

The SHARC II Dewar must periodically be prepared for observing. It is also occasionally necessary to monitor the internal temperatures of the Dewar to verify correct operation. Therefore, two programs are necessary:

**1) Automatic $^3$He fridge recycling software**

**2) Temperature monitoring and logging software**

The SHARC II $^3$He fridge has no moving parts, which lends itself to automatic control. A JAVA program has been written to interface with a Neocera LTC-21 temperature controller to do the recycling. It runs under Solaris and uses the serial port. The program can be set to recycle at a specified time of day. There are three recycling options: full $^4$He/$^3$He cycle (for unpumped liquid helium bath), $^3$He only (for pumped bath), and $^4$He only (for cryogenic debugging and tests). **QUESTION: How can we assure this program will execute even if the host with the X Windows graphics display fails? Do we want separate control and monitoring programs?

Remote monitoring of the SHARC II internal temperatures is highly desirable. This will be accomplished with separate software. The temperatures to be monitored are: coldhead (GRT), $^3$He pump (diode), $^4$He pump (diode), liquid helium coldplate (diode), vapor-cooled coldplate (diode), and liquid nitrogen coldplate (diode). In total, 1 GRT readout and 5 diode readouts are needed. However, the Neocera only supports 2 inputs. **QUESTION: How can we read that many diodes? Multiplexer (relay) box, either homemade or from Keithley? Dedicated readout hardware and analog I/O?