Status, Capabilities, and Use of the ATA Beamformer

Billy Barott
July 8, 2008
Outline

- Introduction
- Basic Use
- Hardware
- Firmware
- Calibration Theory
- Software
- Concluding Thoughts
Major Milestones

- **2007 August**: Prototype controls shown
- **2007 October**: 4-antennas
- **2007 November**: 24-antennas
- **2008 March**: First obs with BAPP
- **2008 April**: 42-antenna, two-pol, 1-beam
- **2008 June**: Second dual-pol b.f.
- **Current**: Upgrades, engineering tests...
System Layout

- **The ATA Synchronous Time-Domain Beamformer**

- **Beamformer**: The whole thing, with one master software controller

- **Beam Processor**: Independent FPGA tree

- **BEE2**: FPGA processing machine

- **FPGA**: Element of the BEE2 (one of 4)

- **iBob**: Processor with 4 ADC inputs
Current Capabilities

- Two Independent Dual-pol beamformers
  - BF1: 42-ants, 18 FPGA (two x9 FPGA b.p.)
  - BF2: 36-ants, 16 FPGA (two x8 FPGA b.p.)
- Beam processors independently steered
- Wired for X and Y beam processors
  - Calibration on X/Y – No CP yet
- In Progress...
  - Let user modify coefficients
  - Better solar/RFI handling in calibration
Using the Beamformer

*Detailed usage information on log.hcro.org
The Very Basic:

- Initialize
- Calibrate
- Observe

Scripts should be run on tumulus or user1
Initializing the Beamformer

1. Verify ATA engineering status
   - Only observe with good feeds
   - Verify LNA / PAM / Focus settings

2. Configure iBobs: bfibob
   - Walsh, Rearm, Sky, Autoatten

3. Configure beamformer: bfini.rb
   - b [beamformer] -a [antlist]
   - bfini.rb -b 1 -a 1ax,1bx,1cx
Calibrating the Beamformer

1. **Strong Source Delay** [--caldelay]
   - Flattens instrumental delays
   - Suggested: Cas-A, 3c274 (high SNR desired)

2. **Unresolved Phase** [--calphase]
   - Aligns instrumental phase offsets
   - Suggested: 3c84, 3c48, 0927+390

3. (Optional) **Wideband Cal** [--calfreq]
   - Calculates derivative of phase with frequency
   - Same source requirements as calphase
   - *Slowly change* --freq over wide range
When To Calibrate

<table>
<thead>
<tr>
<th></th>
<th>--caldelay</th>
<th>--calphase</th>
<th>--calfreq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power cycle or iBob rearm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call to bfinit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Major Link Glitch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minor Link Glitch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changing --freq (&lt; 500 MHz)</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changing --freq (&gt; 500 MHz)</td>
<td></td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>Elapsed time (hours?)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*: Only if --calfreq is not used
**: Not tested, but recommended

- **Calibrate phase:**
  - After potential drift
  - When moving “out of calibration” range

- **Calibrate everything:**
  - After potential link disruptions
Is it good? (calphase report)

Both phase and RMS error should be small
(Phase < 5 degrees, RMS < 30 degrees)

Overflow should always be zero,
but might go above zero during initial calibrations

Peak voltage output clips at 4096.

Link diagnostics: Not fully implemented here,
but will indicate trouble with XAUI links
Steering a Beam (bftrackephem)

1. **Pointing Information**
   - Requires fixed az/el (--azel) or ATA-style ephemeris (--ephem)

2. **Observing Information**
   - Duration (-d), frequency (-f), and beamformer (-b)

3. **Everything Else...**
   - More details in logbook... Offset pointing, sideband selection, observing with 16k-channel spectrum analyzer

- *After obs, beam is “left on” with last values*
- *Pointing is not constrained by primary beam*
Accuracy

- **Delay:**
  - 5°-10° RMS typical with caldelay
  - F.D. FIR 0.02-smp steps (< 2° over 104 MHz)

- **Phase:**
  - 10°-30° RMS typical with calphase
  - F.D. worst error is 4° across 70% bandwidth

- **Amplitude:**
  - F.D. worst error is 8% (0.7 dB), most < 0.4 dB
  - Currently no balancing for sensitivity
  - Autoatten error is about 0.5 dB
Bandwidth

- Filters limit bandwidth
- iBob DDC Passband
  - 78% to 3-dB points
- Fine Delay FIR Filter
  - Designed for 70% (Phase / amp degradation point)
  - 72 MHz (Digital)
    - Discard DC channel
  - 50 MHz (Analog, shifted)
    - Discard DC, 26.2144 MHz
Output Specifications

- **Digital:** 10-Gbps XAUI
- **Analog:** XAUI into iDAC
  - 50 MHz .... Peak-peak max
  - Ordinary +/- 1.0 V pk (2v pp)
  - Clipping at about 1.25V peak
- **Calibration:** Isolated to B.P.
  - No guaranteed delay/phase between X and Y beam processors
  - No guaranteed cal between different beam formers

<table>
<thead>
<tr>
<th>Bits [lsb0]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..7</td>
<td>SubBeam #1 Real</td>
</tr>
<tr>
<td>8..15</td>
<td>SubBeam #1 Imag</td>
</tr>
<tr>
<td>16..23</td>
<td>Refant Real</td>
</tr>
<tr>
<td>24..31</td>
<td>Refant Imag</td>
</tr>
<tr>
<td>32..63</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Noise Output: 0.5 Volts per div.
Things to Watch For

- High RMS on calibration
  - Antenna problems or solar interference
- Very low input and AGC
- Update rates
  - ~5 sec: Software coefficient update
  - ~1024 * fringe rate = fringe update rate
  - ~13.5 MHz fringe table
User Scripts

(More Detail)
User-Level Scripts

- **bfibob**
  - Configures beamformer iBobs
  - Sets walshing, source select, attemplifier, etc.

- **bfinit**
  - Initializes & alters hookup
  - Loads firmware into FPGAs

- **bftrackephem**
  - Loads data for pointing and obs-time config
  - Used for calibration
bfibob

- Controls the beamformer ibobs for testing and observing
- Walshing (on/off)
- Source (Sky/sinewave/noise)
- Reset (1pps, seed)
- Attemplifiers (autoatten)
bfinit

- Initializes and creates hookup
- Sets default peak headroom
- Common Switches
  - `-a`: Antenna list
  - `-b`: Beamformer
  - `-h`: Headroom (defaults to 0.75)
bftrackephem

- General pointing and settings
- Calibration (delay, phase, agc)
- Returns diagnostic information
- *Sets atasetskyfreq*
- Software-synchronous register updates

**Pointing Switches:**
- **--ephem:** Loads ephemeris file
- **--azel:** Uses fixed az,el
- **--offset[az,el]:** Offsets ephemeris file
bftrackephem

- **Common Switches:**
  - -b: Beamformer select
  - -f: Sky frequency (MHz)
  - -d: Data table time (seconds)
  - -n: Update cycles (req. for cal)

- **Other Switches:**
  - -i: Integration time (seconds)
  - --cal[xxxx]: Calibration cycle
  - --write: Correlates without calibration
  - --sideband: Hilbert transform setting
  - --agc[db]: Automatic gain controller level
Exit Status Reporting

- Implemented:
  - Most script failures

- Coming soon:
  - Calibration status (Succeeded, failed, marginal)
  - Automatic weighting to eliminate trial-and-error dropping of inputs from hookup
Bftrackephem Startup

1. Connect to User Server [-b beam]
2. Sky Freq [-f MHz]
3. Stop Obs Timers
4. Pointing Offsets [--offsetaz/el]
5. Load Pointing for [-d seconds]
6. Process Pointing Data
7. AzEl → GeoPoint (1s resolution)*
8. **GeoPoint → Pointing Tables
9. Load Tables in BEE Servers
10. Synchronize Clocks
11. Hilbert Mode [--sideband sb]
12. Adjust Gain [--gainadjustXX]
13. Start Obs Timers [--timer sec]
14. Set Data Directory
15. Observing Loop
16. Exit

If Timers were set
Bftrackephem Observing Loop

Wait 2 Seconds

Normal Observing

Look-Ahead Write (10s)

Write “Future” Register Values

Load at “Future” Time +/- 0.1s

Integrate [-i seconds]

Write output snapshots

-n used?

Loop Until Count [-n counts]

Loop Until Time [-d seconds]

Loop Or Exit

Script Action
Set Data
Internal Action
Other
Bftrackephem Calibration Loop

Wait 2 Seconds

Calibration Cycle

Load Pointing for 
[-d seconds]

Process Pointing Data

Adjust AGC

Write output snapshots

Loop Until Count 
[-n counts]

Exit

Look-Ahead Write

Integrate 
[-i seconds]

Apply Calibration Corrections

Script Action

Set Data

Internal Action

Other
Hardware
One Beamformer

- 24 ADC iBobs
  - 96 inputs at 104 MS/s
- 5 BEE2s
  - 4 FPGAs per BEE2
  - Clocked from 52 MHz iBob output
- All hardware is synchronously clocked
The CASPER BEE2

- 5 FPGAs (4 used)
- Embedded PowerPC
  - Running BORPH Linux
- Supports four 10 Gbps XAUI links per FPGA
## FPGA Use

<table>
<thead>
<tr>
<th></th>
<th>Initial</th>
<th>RV13</th>
<th>RB13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antennas</td>
<td>12</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Beams</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FIR Taps</td>
<td>2</td>
<td>8*</td>
<td>6</td>
</tr>
<tr>
<td>Autorotator</td>
<td>0</td>
<td>4</td>
<td>8 (Muxed)</td>
</tr>
<tr>
<td>FFT Bins</td>
<td>0</td>
<td>256</td>
<td>128</td>
</tr>
<tr>
<td>I/O Registers</td>
<td>127</td>
<td>112</td>
<td>60</td>
</tr>
<tr>
<td><strong>Slices Used</strong></td>
<td><strong>84%</strong></td>
<td><strong>99%</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

- I/O registers also affect load time
- 12 antennas is the absolute maximum per-BEE2-FPGA
iBobs

- 24 ADC iBobs (each b.f.)
  - Using 84 of 96 inputs
  - Up to 48 antennas without any additional hardware

- Overheating
  - Heat sink at 80C+ w/o fans
  - 3m XAUI links fail, but 1m ok
  - Use 1m copper, 3m fiber, speed rate...?
  - Link problem only – the FPGAs are not in danger.
Synchronization

- Heat-induced link errors cause “slip”
- Formerly-calibrated data streams lose synchronization!
Synchronized (example)

Time Series: Synchronized

X-Corr Phase Spectra
5-Sample Slip (example)

Time Series

X-Corr Phase Spectra
Synchronizer Solves (most) Slips

- iBob: 10ms and 1pps out-of-band signals
- BEE2: Self-adjusting circuits keep o.o.b. pulses aligned
  - Reference link coupled to Ref-ant
- Available Diagnostics
  - Link Offset... Times Adjusted...
  - XAUI “Down” and “Empty” status counts
  - Excessive increases indicate problems!
Firmware Architecture
Cascaded Data Flow

- **Leaf Nodes (input from iBobs)**
  - Apply beam-forming corrections
  - Form sub-beam
  - Correlate antennas with reference antenna

- **Branch Nodes (input from leaf nodes)**
  - Form sub-beam
  - Correlate leaf-node reference antennas
Why Correlate?

- Early design only maximized antennas & beams
  - 12 antennas per leaf node, 2 beams
- Calibration method undefined
  - Iterative software adjustments (slow, inaccurate)
  - Copy data from external correlator
Separate Signal Paths

- ATA Beam-former and Correlator use separate signal paths
  - Different instrumental offsets in down converter
  - Different power-up clock offsets in ADCs
- Calibration must be contained in Beam-former
- Current limits: 8 ants x 1 beam (and stuffed!)
Inside the FPGAs

- Reference antenna selected from inputs
- Later stages correlate reference antennas (not sub beams)
- ATA beamformer has 3 stages for 42 antennas
Leaf Node Signal Path

- Bulk Delay: 1024-sample buffer memory
- Fine Delay: 6-tap Real FIR Filter (12-bit coefficients)
- Fringe Rotator: Programmable phase angle and rate
Branch Node Signal Path

- Fs/4 Shifter: Moves usable spectrum to sideband
- Hilbert Transform: Selects real sideband output
Output Data Format

<table>
<thead>
<tr>
<th>Bits [lsb0]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..7</td>
<td>SubBeam #1 Real</td>
</tr>
<tr>
<td>8..15</td>
<td>SubBeam #1 Imag</td>
</tr>
<tr>
<td>16..23</td>
<td>Refant Real</td>
</tr>
<tr>
<td>24..31</td>
<td>Refant Imag</td>
</tr>
<tr>
<td>32..63</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

- Upper half of channel is reserved for future use
  - More beams?
  - Correlator Output?
- Passes iADC out-of-band signals from reference channel
  - 1pps / 10ms tick
- Signal is compatible with iBob output
Fringe Rotator
Fringe Rotator

- 10-bit starting angle, 32-bit fringe rate
- 0..6400 Hz in 1.49 uHz steps
  - Register = Fringe(Hz) * 2^{46} / 104.8675e6
- One multiplexed table per leaf
Fractional Delay
Sub-Sample (Fractional) Delay

- 6-Tap, 12-Bit FIR Filter
- GLS Coefficients
  - Good for small filters
- Accuracy Depends On Filter Design!
  - Taps, Bandwidth, Delay
Phase Error vs. Design Bandwidth

Phase Error for Fine Delay Filter

Operating Bandwidth (Normalized to 104 MHz)

Maximum Phase Error (Degrees)

- 6 Taps
- 8 Taps
- 10 Taps
- 12 Taps

We Are Here
Amplitude Error vs. Design Bandwidth

Amplitude Error for Fine Delay Filter

- Maximum Voltage Amplitude Error (Ratio)

- Operating Bandwidth (Normalized to 104 MHz)

- We Are Here

- 6 Taps
- 8 Taps
- 10 Taps
- 12 Taps
Amplitude Errors Depend on Delay

Fluctuations Depend on Delay
In practice...
FIR Summary: 6-Taps / GLS

- Worst errors near the band edge
- 70% Bandwidth Design
  - Worst gain error of 8% (0.7dB)
  - Worst phase error of 4 degrees
- 50% Bandwidth Design (possible)
  - Worst gain error of 1.5% (0.13dB)
  - Worst phase error of 1 degree
  - With Spectrum Shifter and Hilbert Transform: Approach full-50 MHz output to Prelude
F_{s}/4 Spectrum Shifter

Without shift: Fine delay bandwidth limits useful analog output range

With shift: Non-useful spectrum moved to discarded sideband

* “DC” at 26.2144 MHz must be discarded
Sideband Select Notes

- Spectrum shifter is always on with sideband select (--sideband)
- Sky frequency (--freq) appears at 26 MHz
  - With (--sideband upper), frequencies above 26 MHz are at sky frequencies above (--freq).
  - With (--sideband lower), the spectrum is reversed so frequencies above 26 MHz are at sky frequencies below (--freq).
Calibration
Three Calibration Modes

- **Delay (--caldelay)**
  - Strong source (casa)
  - Solves instrumental delay

- **Phase (--calphase)**
  - Unresolved source (3c84)
  - Assumes “zero-delay”, solves instrumental phase

- **Frequency (--calfreq)**
  - Assumes “zero-delay”, “zero-phase” (at $F_0$)
  - Solves $d\phi/dF$
Iterative Calibrations

- Calibration is refined with each iteration
- Zero-slope forcing on weak calibrations eliminates false-turns from noisy spectra
  - Delay is assumed to already be zero from delay-cal
Delay Calibration

- GeoPoint Delay and Delay Offset produce the total FPGA delay
- Delay error is estimated by phase slope of cross correlation spectra
Phase Calibration

- Fringe corrections determined from GeoPoint Phase, Phase Offset, Phase rate vs. Frequency, and current sky frequency
- \( F_0 \) set to \( F_{\text{sky}} \) on each calphase (but not obs. or calfreq)
- Phase error is vector mean phase
Frequency Calibration

- Phase vs. Freq is assumed to be linear.
- Major component is “unknown delay” in system.
  - Delays at skyfreq contribute to fringe angle, delays at IF do not.

![Diagram showing phase vs. frequency relationship and data flow]

- GeoPoint Phase
- Phase Offset
- dΦ/dF
- F_{sky} – F_0
- Phase Error
- Fringe Starting Angle
- Fringe Rate

- Cal Term
- Data Flow
- Pointing Data
- Cal Feedback
- To FPGA
- From FPGA
Amplitude Calibration (in progress)

- **Goal:** Weight all inputs for best beam
- **Problem:** Attempts to set amplifier is noise only
- **Requires Sensitivity:** (from correlator)
- **Two modes planned:**
  - Equal Signal: To show “Ideal” beam pattern
  - MRC: To realize the best output SNR
In Software

Pick One:
- `--caldelay` (Delay Calibration Cycle)
- `--calphase` (Phase Calibration Cycle)
- `--calfreq` (Frequency Calibration Cycle)

Also Use:
- `-d` (Ephemeris to pre-load)
- `-i` (Integration time, seconds)
- `-n` (Number of iterations)
- `-f` (Sky frequency, MHz)
- `--ephem` (Ephemeris file)
Example

```
bftrackephem -b 1 -f 1420 -i 20 -d 300 --ephem "casa.ephem" -n 4 --caldelay
```

Sets beamformer 1 to a skyfreq of 1420 MHz, and starts a delay calibration cycle on CasA. The cycle uses four iterations of 20-second integrations and preloads 5 minutes worth of ephemeris before each cycle.
Potential Limitations

- During integration:
  - Fringe angle is updated by hardware
  - Delay is not updated by software

- Very long integrations will smear results
  - Phase offset “ok”

- “Worst:” for 300m e-w baseline w/ refant
  - 1 turn in 130 seconds
Software Design
Written for Portability

- A generic, cascaded-FPGA beam processor
- Uses Ruby for fast development cycle
- Only user scripts are ATA specific
- (Hopefully) Reusable architecture
Abstraction Layers

- **User Scripts**
  - Observing level, ephemeris, frequency, etc.

- **Master User Server**
  - Coordinate user requests
  - Organize antennas by hookup and by hardware
  - Create data tables
Abstraction Layers

- **BEE User Servers**
  - Hardware only: BEE2, FPGAs, XAUIs, ADCs
  - Data tables (register vs. time)
  - Originally designed to be “Fast” on power-PC

- **BEE Process Servers**
  - Single data pipe from user server
  - Allows multiplexed access to corner FPGAs
  - Most writes are buffered to limit DRb calls
Concluding Thoughts and Engineering Tests
Simulated Beam Shape (iBob Noise)

Beam pattern using iBob noise (ATA-35), Az = 40 El = 80

- Measured
- Predicted
Simulated Array Gain (iBob Noise)

Total Output power vs. Array Size, iBob Noise

- Ideal Gain
- Peak Detector
- 15 MHz Marker
- 25 MHz Marker

Signal Power (dB) vs. Single Input

Number of Enabled Antennas
Solar Interference in Crosses
Remaining...

- Speed and performance (always)
- Very long observation response
- Circular Polarization
- Null beam
- RFI rejection
- Amplitude balancing
Software is NOT Real-Time!

- 2 sec preload allowance
- Variation in load task affects fringe accuracy
- *Recently improved, but illustrates vulnerability
Thoughts for the future

- Power PC is slow and not real-time
  - C is faster, but Ruby shortens development
  - For 0.1 Hz fringe rate, 3.6 deg / 100ms
  - Include “data tables” and RTC in firmware?
- FFTs required for correlator, and 16 multiplies per time-domain path
  - In retrospect, a frequency domain beamformer makes a lot of sense here...
Questions...?