MASSACHUSETTS   INSTITUTE   OF   TECHNOLOGY
HAYSTACK   OBSERVATORY
WESTFORD,   MASSACHUSETTS 01886

20 January 2012

To: Mark 5 Development Group

From: Russ McWhirter

Subject: Testing of RDBE PFBG version 1.4.1

The ROACH Digital Back End (RDBE) utilizes the CASPER ROACH board [7] to perform digitization and channelization of two 512 MHz bands. Each band produces 16 channels, which are requantized to two bits. A channel selector chooses 16 channels for output. For example, the odd channels from both bands can be selected, which are then transmitted to recorders on a 10 Gigabit Ethernet interface at a total data rate of 2.048 Gbps. The channelization is performed using polyphase filter banks (PFB) and complex to real interpolation [3].

The geodetic firmware has been released as PFBG version 1.4 [9], which has been validated in testing and VLBI experiments. However, there is a phase change of 180° in some of the output channels which requires phase calibration. This was found to be a shift in the phase rotator of the interpolator, which has been corrected in release 1.4.1 [10]

To verify the correct operation of PFBG 1.4.1, several tests have been performed with analog test signals and digital test vectors.

A data capture module was added to the output of the interpolator. The interpolator has eight outputs containing the complex samples of the 16 channels from each band. Four of the outputs contain the real part of the samples and four contain the imaginary part. Four channels from each band are multiplexed onto each pair of outputs. The capture depth is 1024 samples of the eight outputs, which is demultiplexed to 256 complex samples of 32 channels.

A MATLAB simulation of the RDBE was written that produces a floating point model of the complete RDBE signal chain. The internal data capture of the interpolator output can be compared to the model. The two bit recorded data can also be compared to the ideal model.

Sinusoidal Test Results

Connecting a signal generator to the inputs of the RDBE and processing the recorded data shows the relative phase between the corresponding channels of each input band.

A signal generator (HP 8648B) was set to -9 dBm and connected with a splitter to both inputs of the RDBE, as shown in Figure 1. Note that the automatic level control (ALC) was bypassed and the signals were connected directly to the ADC inputs to simplify testing.

The signal generator was swept over two Nyquist zones, from 1 to 1021 MHz in 4 MHz increments, and the data recorded at each frequency. The recorded data were then processed to extract the signal from the relevant output channel [4].

The plot in Figure 2 shows the difference in the phases between the two RDBE inputs for PFBG

1.4.1, which is close to zero as expected. There is a residual phase difference that can be attributed to analog effects up to and within the digitizer board.

By comparison PFBG 1.4 has a phase difference that is 180° for channels 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, as shown in Figure 3.

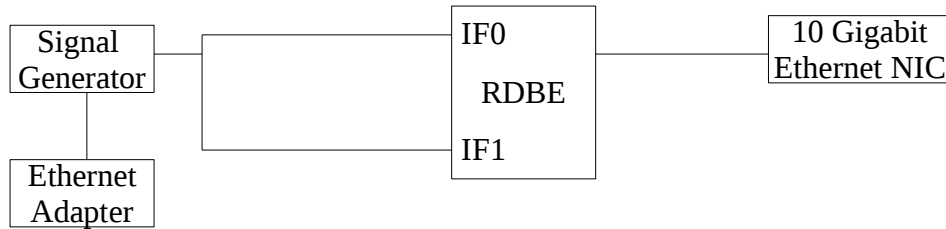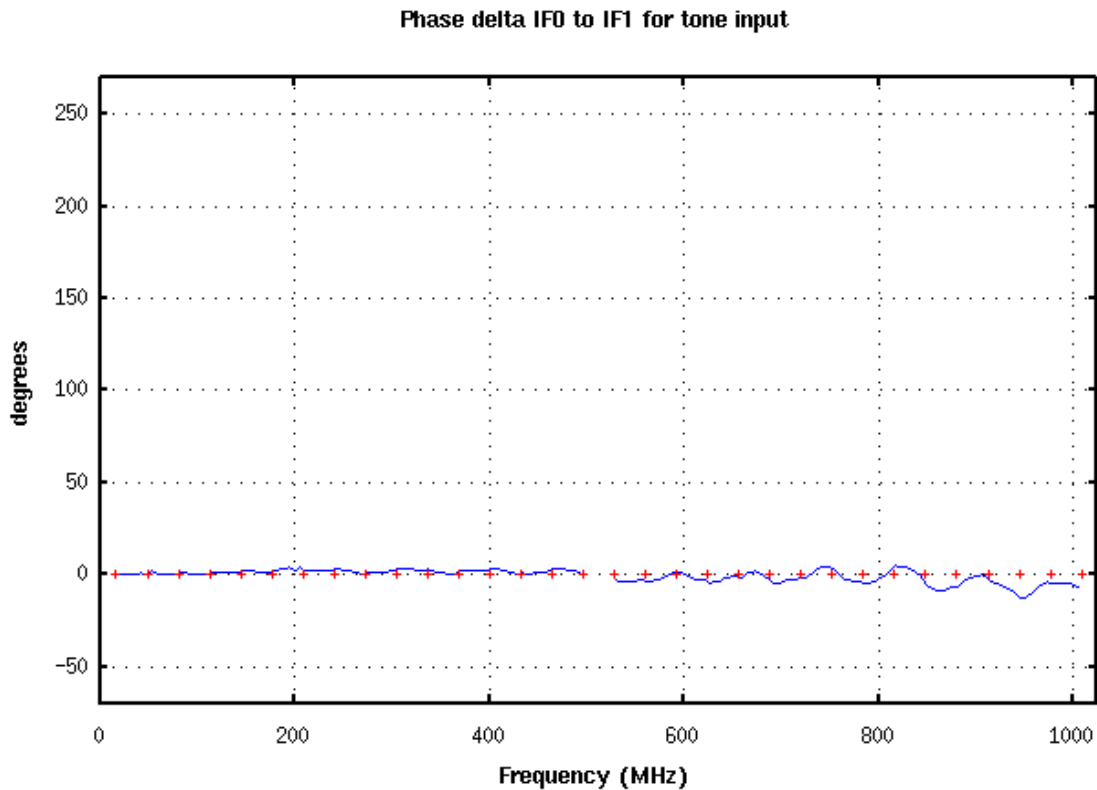Figure 1. Sinusoidal test setup

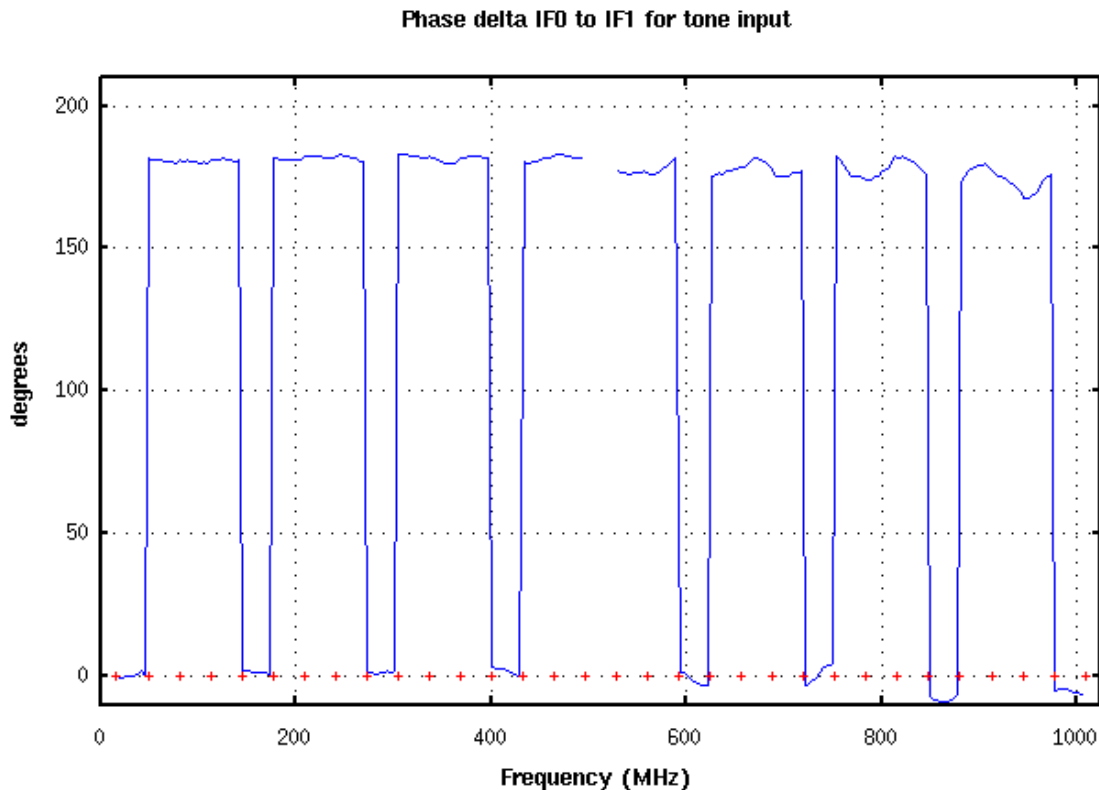Figure 2. Plot of phase between IF0 and IF1 for PFBG 1.4.1

Phase delta IF0 to IF1 for tone input

Figure 3. Plot of phase between IF0 and IF1 for PFBG 1.4

TVG Testing

A test vector generator (TVG) was added to both inputs of the RDBE, and a control register selects between input ADC data or TVG data, as shown in Figure 4. The interpolator capture module was added for testing only, and is not present in the release build of PFBG 1.4.1.

The TVG has a 4096 point RAM that can be loaded with arbitrary test vectors, such as an impulse, or a sinusoidal signal, or random noise. The TVG starts the output sequence when the pulse per second (PPS) is detected and loops through the RAM continuously. The RAM is read up to a programmable depth and then continues from the first address. A burst mode is also provided that reads out the RAM once only.

The interpolator capture data can then be compared to the floating point Matlab model to show correct operation of the RDBE signal processing chain [12].

The recorded ethernet packet data can also be compared to the MATLAB model. In addition, since the TVG is aligned to the PPS, the recorded packets can be checked for correct data alignment and timing and header information.
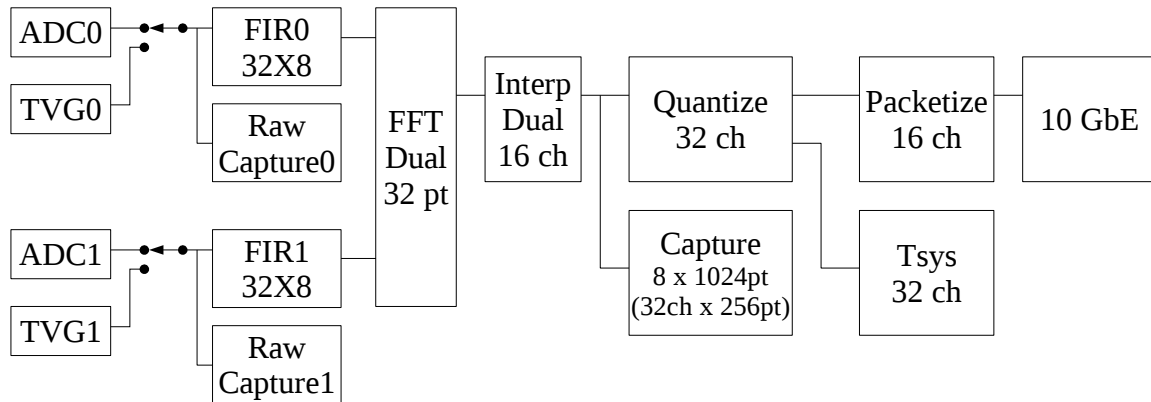
Figure 4. TVG and data capture module in PFBG 1.4.1

TVG Impulse

An impulse was loaded into both TVG RAMs with an amplitude of 64 ADC units, and the output of the interpolator captured. The captured 16 channels of IF0 and IF1 are plotted in Figure 5 and Figure 6, respectively.

The simulation data are plotted in the first subplot, followed by the captured data, and lastly the difference between captured and simulated. Both IF0 and IF1 have identical responses as expected.

The error in the captured data is approximately 8e-6, with a signal amplitude of 0.5. The internal data path has 8 integer bits and 17 fractional bits, so the error is approximately one least significant bit.
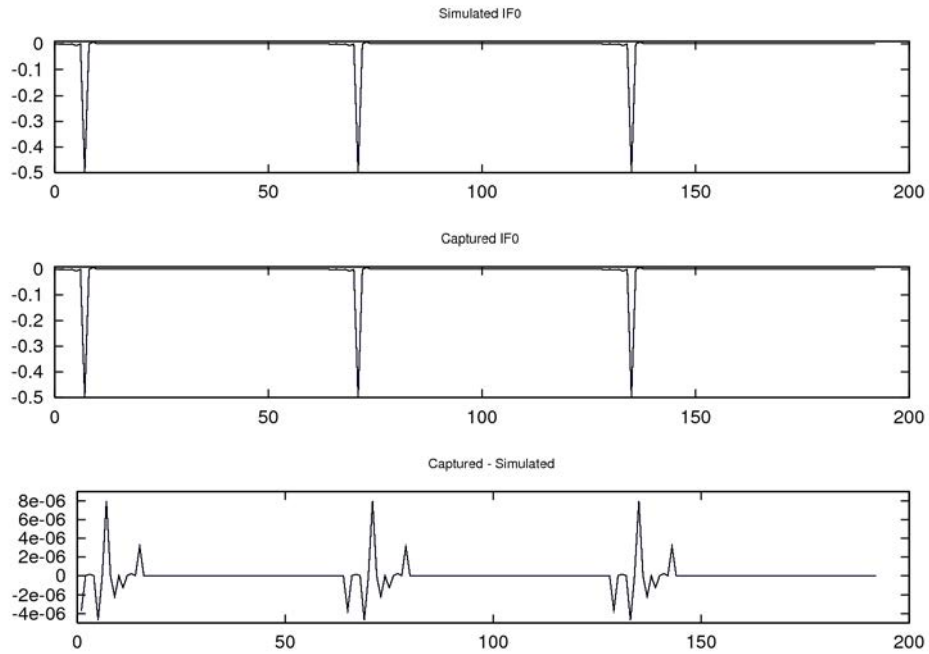
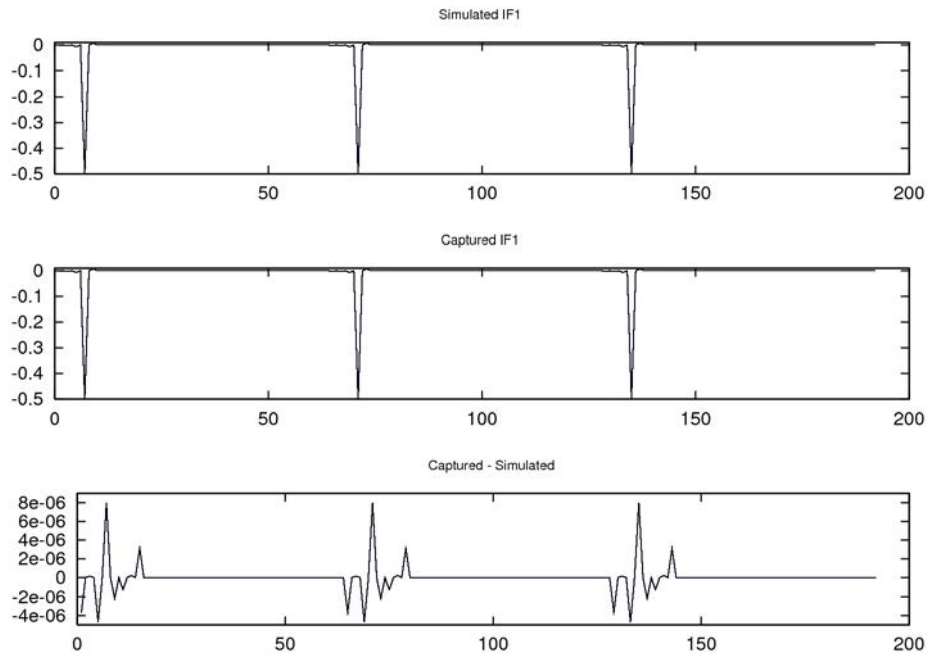Figure 5. IF0 impulse response, showing simulated, captured, and difference

Figure 6. IF1 impulse response, showing simulated, captured, and difference

TVG Tones

A signal was generated that contained the sum of 15 tones that correspond to 2 MHz in channels 1-15. In the TVG RAM this equates to 18, 50, 82, 114, 146, 178, 210, 242, 274, 306, 338, 370, 402, 434, and 466 MHz. The amplitude of each tone was 8 ADC units, to avoid saturation. The output of the interpolator was captured, and the 16 channels of IF0 and IF1 are plotted in Figure 7 and Figure 8, respectively.

The error in the captured data is approximately 2e-4, with a signal amplitude of 1.0. This equates to approximately 5 bits of noise in the internal data path. Since there are 8 integer and 17 fractional bits, the captured signal matches the ideal model exactly over a 20 bit range.
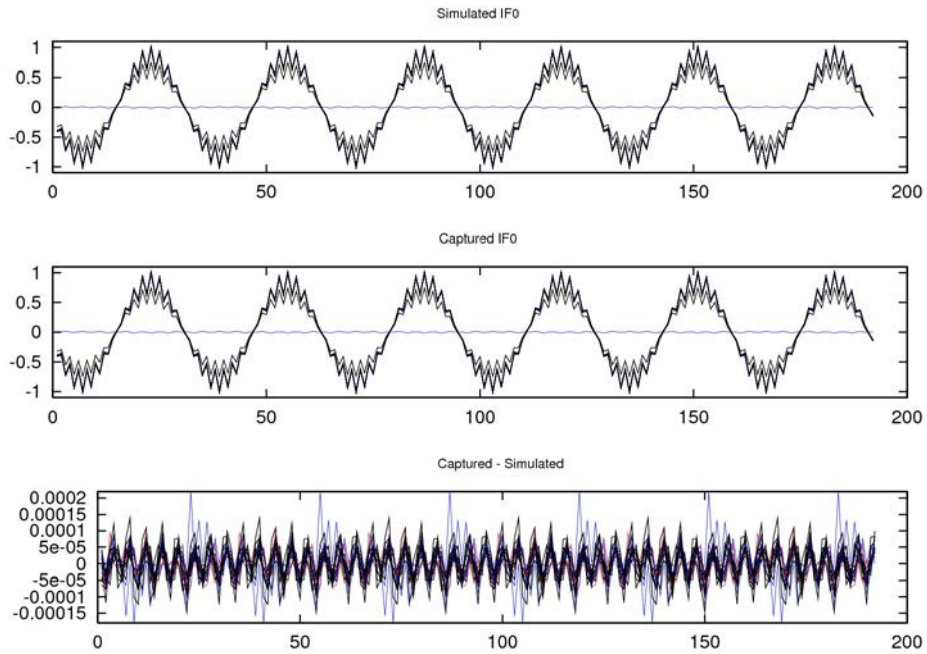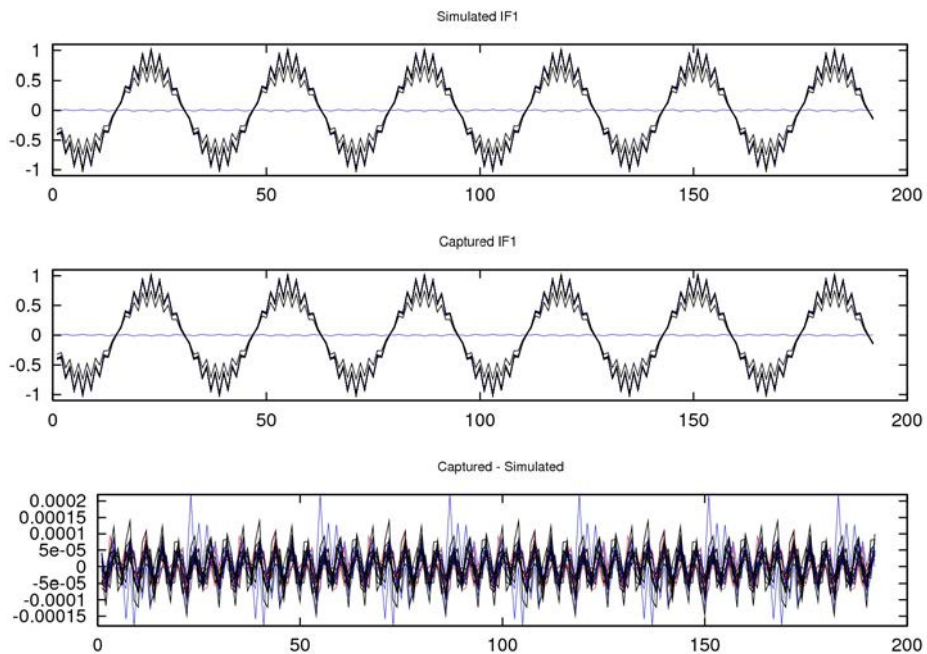

Figure 7. IF0 tone response

Figure 8. IF1 tone response

TVG Gaussian Random Noise

A gaussian random signal was generated with a variance of 32 ADC units, which approximates the expected input level during operation, generated with the MATLAB randn function. Seeds of 0 and 1 were used to create known sequences for each TVG. The output of the interpolator was captured, and the 16 channels of IF0 and IF1 are plotted in Figure 9 and Figure 10, respectively.

The error in the captured data is approximately 2e-4, similar to the error in the sinusoidal case, showing ideal operation over a 20 bit signal range.
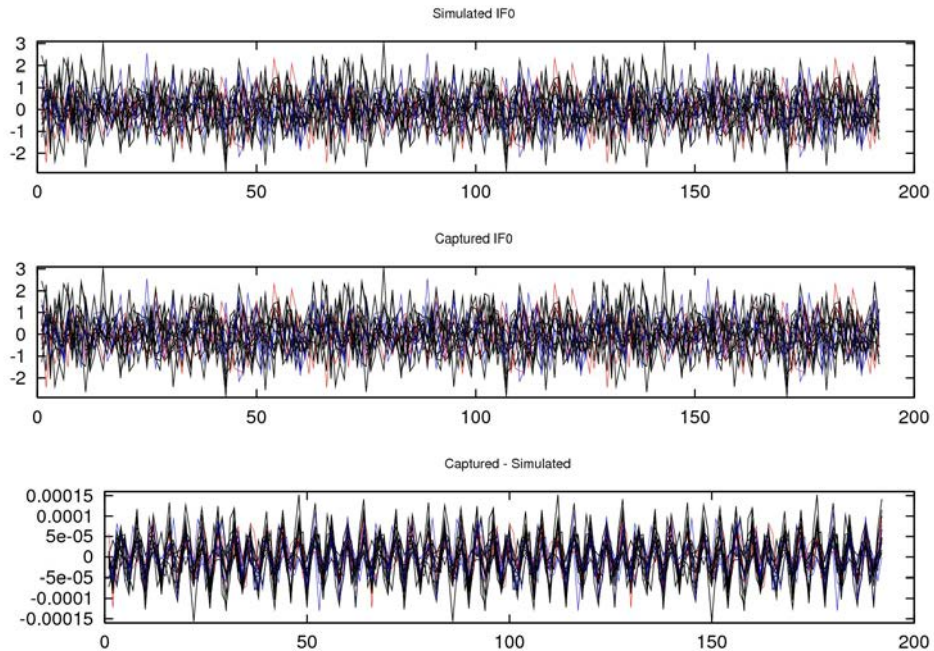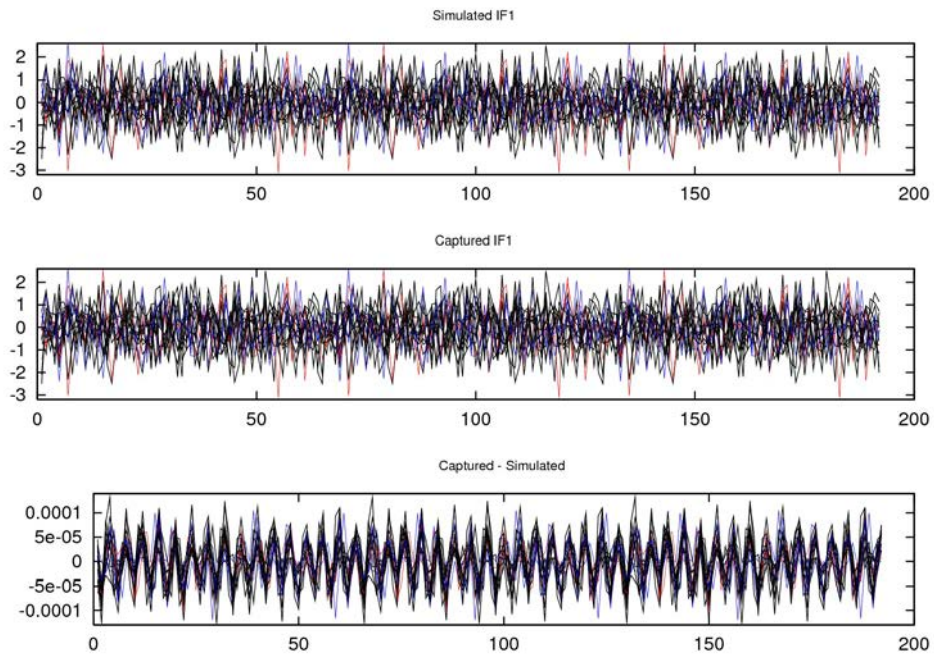

Figure 9. IF0 gaussian random noise response

Figure 10. IF1 gaussian random noise response

TVG Gaussian Random Noise Recorded Data on Disk

The gaussian random noise vectors were loaded into the TVG and quantized with the default values threshold of 0.52. The recorded data were then compared to the ideal floating point model. For both IF inputs with independent noise vectors, the two bit data agreed exactly for the entire 4096 test vector, as shown in Figure 11 and Figure 12.
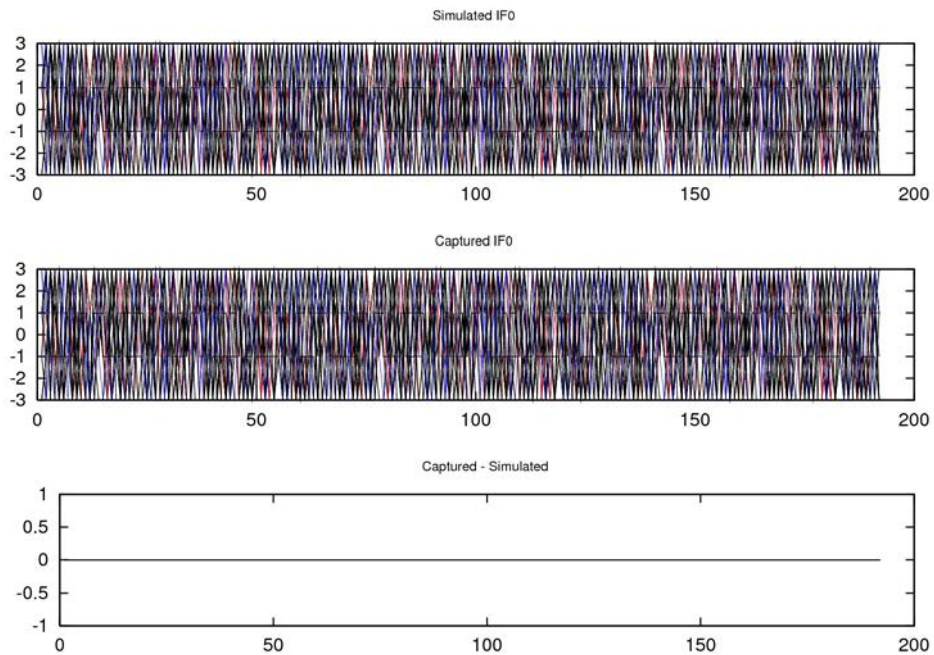

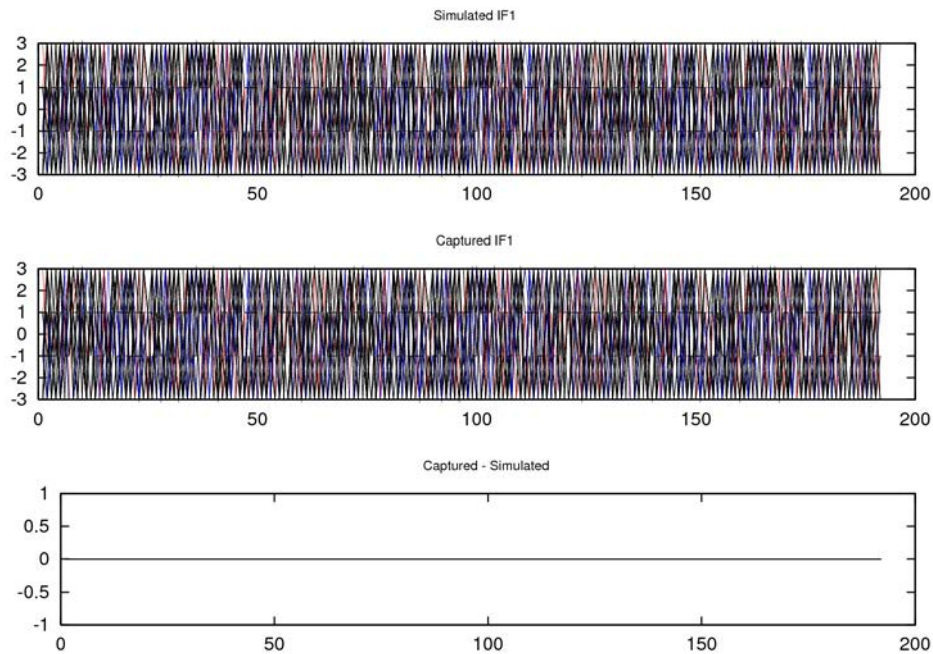Figure 11. IF0 gaussian random noise response recorded to disk

Figure 12. IF1 gaussian random noise response recorded to disk

References

[1] Xilinx, IP LogiCORE FIR Compiler v5.0 , DS534 March 1, 2011 ,
http://www.xilinx.com/support/documentation/ip_documentation/fir_compiler_ds534.pdf
[2] Fred Harris, Chris Dick, and Michael Rice, Digital Receivers and Transmitters Using Polyphase Filter Banks for
Wireless Communications, IEEE Trans. on Microwave Theory and Techniques, Vol. 51, No.4. 4 April 2003 , http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1193158
[3] Alan E.E. Rogers and Hans F. Hinteregger , Real to real polyphase filter bank , MARK 5 MEMO #018 , November 19, 2004 , MIT Haystack Observatory
http://www.haystack.mit.edu/tech/vlbi/mark5/mark5_memos/018.pdf
[4] svn+ssh://vault/svnrepos/RDBE/FPGA/trunk/M-files/RoachGUI/scanphase.sh
[5] International VLBI Service for Geodesy & Astrometry
http://ivscc.gsfc.nasa.gov/publications/memos/index.html
[6] VLBI Broad Band Development Memo Series
http://www.haystack.mit.edu/geo/vlbi_td/BBDev/memo.html
[7] CASPER Collaboration https://casper.berkeley.edu/wiki/ROACH
[8] 9th US VLBI Technical Coordination Meeting
http://www.aoc.nrao.edu/~jromney/9th_US_VLBI/
[9] svn+ssh://vault/svnrepos/RDBE/FPGA/tags/PFBG-1.4/PFBG_1_4.bin
[10] svn+ssh://vault/svnrepos/RDBE/FPGA/tags/PFBG-1.4.1/PFBG_1_4_1.bin
[11] svn+ssh://vault/svnrepos/RDBE/FPGA/trunk/M-files/RoachGUI/rdbesim.m
[12] svn+ssh://vault/svnrepos/RDBE/FPGA/trunk/M-files/RoachGUI/tvg_ram_plot.m