Users Guide for ADC16x250-8 coax rev 2 and ADC16x250-8 RJ45 rev 1 boards

The Hittite HMCAD1511 ADC chips on the ADC16 board are somewhat different from other ADC chips used on other CASPER ADC boards. The ADC chips on the ADC16x250-8 board do not fully power up when power is applied. They need to be turned on via software and then the FPGA input delay blocks need to be calibrated to account for the design-specific clock skew at the FPGA SERDES inputs and to properly frame the de-serialized bytes.

Here is a high level overview of the startup sequence:

* Supply clock(s), 1PPS, network connections to the hardware
  The clock rate must be consistent with the mode the FPGA is to be used as well as the application's FPGA gateware allowable clock range.

* Power up and netboot the Roach2 from a boot server with the required Roach2 tcpborphserver software environment and support software.
  A good place to start the process of setting up the Roach2 to netboot is this CASPER mailing list thread:
  http://www.mail-archive.com/casper@lists.berkeley.edu/msg03918.html
  see also
  https://casper.berkeley.edu/wiki/Tcpborphserver
  https://casper.berkeley.edu/wiki/KATCP

  Once setup netboot is very convenient. Many users consider netboot essential if one has more than one ROACH or ROACH2, but even for just one board it makes things more pleasant since you can copy the bof file directly to the NFS server instead of to the ROACH and you have a "real" Linux distribution instead of busybox.

  Standalone (soloboot) operation is possible but only after additional system administration arrangements are made. Soloboot operation won't be discussed further in this users guide.

* Program FPGA with ADC16 design

* Turn on ADC chips via software

* Re-program FPGA with same design

* Calibrate FPGA input delays

* Bitslip FPGA SERDES blocks until data is properly framed

There is software already written to do this. It is written in Ruby. You need to have Ruby (1.9 or later recommended) and Rubygems installed. You can then install the ADC16 software onto the boot server with:

```
gem install --source http://astro.berkeley.edu/~davidm/gems adc16
```

```
This will give you a command "adc16_init.rb" that you can use to
program the FPGA and perform all the calibration steps required.  Here
is a sample command line session using adc16_init.rb where
is an sample of a bof file.

$ adc16_init.rb pf1 roach2_fengine.bof
Connecting to pf1...
Programming pf1 with roach2_fengine.bof...
Design built for ROACH2 rev2 with 8 ADCs (ZDOK rev2)
Resetting ADC, power cycling ADC, and reprogramming FPGA...
ZDOK0 clock OK, ZDOK1 clock OK
Calibrating SERDES blocks...ABCDEFGH
SERDES calibration successful.
Selecting analog inputs...
Done!

The adc16_init.rb script expects/requires the bof file to be in the
tcpborphserver's boffiles directory (i.e. already on the ROACH2
filesystem).  The adc16_init.rb script does NOT support dynamically
uploading the BOF via the "?uploadbof" command.

The bof files may be in the compressed bof.gz format.

The initial "released" version of the yellow block didn't
automatically handle the data if the ADC chips were not
configure into the 4 input / chip (16 input / board) mode.
Later versions did handle the demux of the data streams
if the chips were programmed into the 2 or 1 input / chip mode.

The gateware and software has been written with the expectation that
all ADC chips are programmed into the same 4,2 or 1 input per chip
mode.  There are apt to be significant problems if the ADC chips are
not all programmed into the same basic mode.

The "User IP Clock Rate" to specify in the XSG block's dialog
for the ADC16  must be in the range of
a) 135 to 240 MHz for sample clock of 135 to 240 MHz for 16 inputs/board mode.
b) 270 to 480 MHz for sample clock of 270 to 480 MHz for  8 inputs/board mode.
c) 540 to 960 MHz for sample clock of 540 to 960 MHz for  4 inputs/board mode.

For the ADC16 yellow block one must specify the "User IP Clock Rate"
as shown above and demux factor in the XSG block.  "adc0_clk" must be
selected as the "User IP Clock Source".  The demux factor is due to
how the Hittite ADC chip supplies the data to the FPGA as a function
of the number of analog inputs to sample.

demux = 1 for 16 inputs/board  (aka 4 inputs/chip)
demux = 2 for  8 inputs/board  (aka 2 inputs/chip)
demux = 4 for  4 inputs/board  (aka 1 input /chip)

inputs
per chip       ADC sample clock
4            1 * "User IP Clock Rate" = 1 * FPGA fabric clock rate
2            2 * "User IP Clock Rate" = 2 * FPGA fabric clock rate
1            4 * "User IP Clock Rate" = 4 * FPGA fabric clock rate
```

```
The ADC16 yellow block is currently available via:
  git clone http://astro.berkeley.edu/~davidm/mlib_devel.git

  As of 2013apr12 the designs have been compiled for 200 MHz clock
  as required by the first application (PAPER128 128 dual poln F Engines).
  The ZDok+ pinout revision 2 must be selected!


One sample bof file is adc16_test_rev2x8.bof.gz
  https://casper.berkeley.edu/wiki/images/b/b3/Adc16_test_rev2x8.bof.gz

  Engineering test model.  Has 64KB data buffers for each of the 32 possible
  analog inputs provide by 2 ADC16s.  These 64KB buffers are
  in additional to the yellow block's built-in 1K buffers and are
  used to collect raw ADC values for software postprocessing
  such as FFT, histograms, ...
  Compiled for maximum Roach2 rev2 FPGA fabric clock of 200 MHz.
  It is for ADC16x250-8 boards with rev2 ZDok+ pinout :
      coax rev 2
      RJ45 rev 1
  This gateware has no knowledge of demux=2 and demux=4 modes.  Thus
  if the chips are programmed in those modes the data must be post
  processed to make sense.  But, even without reordering the non-demuxed
  data the boffile can still be used for collecting and plotting
  of histograms of raw ADC values (time order doesn't matter).

Sample user session :
# configure the ADC16(s)
  adc16_init.rb -v -i 2 roach_name adc16_test_rev2x8.bof.gz
# histogram plot
  adc16_plot_chans.rb -l 65536  -H -c a1  roach_name
# FFT plot
  adc16_plot_chans.rb -l 65536  -F -c h4  roach_name
# check status of high speed ADC chip to FPGA digital links
  adc16_status.rb -c roach_name

The plotting commands above assume the pgplot utilities are installed.
The pgplot gem can be installed with:
sudo gem install pgplot

The pgplotter gem can be installed with:
sudo gem install --source http://astro.berkeley.edu/~davidm/gems pgplotter

The above two installs are all that's required for the histogram plots.
To also plot spectra of the inputs the gsl gem package is required.
sudo gem install gsl
This requires the Debian/Ubuntu "libgsl0-dev" package or equivalent.
```