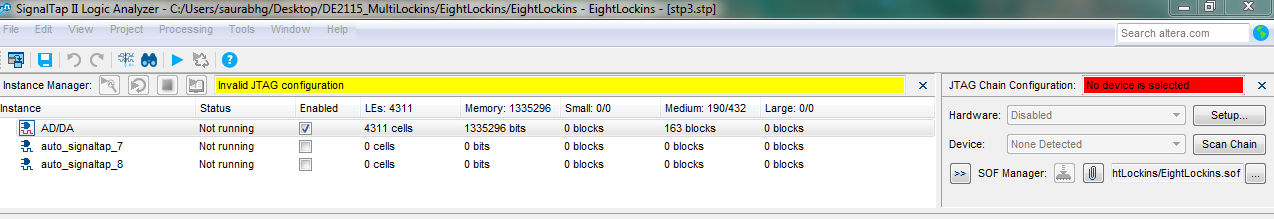
# MAFLIA User-Guide

## By: Bill & Saurabh

### Things to consider for FPGA-RPi operation.

* Connection diagram: <https://github.com/wahudson/fpgadata/blob/master/trunk/doc/rpixel.text>
* How to program the FPGA using the .SOF file:

Open Quartus -> tools -> Signal tap II and then use the panel on the right side in the image below to program the FPGA. Make sure the SOF file path is correct.



* Scripts to tune the o/p frequencies (comb and custom spacing) and how to use them through system console:

Open Quartus -> tools -> System debugging tools -> system console. Then use ‘execute script’ to run the script. You will need to adjust the lock-in frequencies prior to executing the script. The ‘TCL console’ will show the exact frequencies at which the lock-ins are running.

* Amplitude adj [LEDG3]
  + ADC has a full scale range of 512 mVpp, AC coupled
  + LEDG[3] glows when the voltage is beyond the full scale range. Run a sample scan to ensure no overflow through the scan area of the sample
  + Use amps / attenuators as necessary such that the signal stays within range
* DAC (14-bit) amplitude adjustment
  + Use appropriate amplifiers to tune the drive signal amplitude
  + For multi-lockin operation the DAC amplitude needs to be adjusted in the following script: <https://github.com/eefer/DE2115_MultiLockins/blob/master/EightLockins/scripts/mli_comb.tcl>
* CIC adjustment (overflow)
  + LEDR[0] glows when a CIC overflow is encountered
  + Adjust the parameters in the script here to avoid overflow: <https://github.com/eefer/DE2115_MultiLockins/blob/master/EightLockins/scripts/mli_comb.tcl>
* AFM sync into 74AHC14 Schmitt trigger on breadboard (protection)
  + The i/p to the FPGA must be in the range of 0-3.3 V max
  + We apply that part (as a buffer) on a breadboard and is powered by 3.3 V supply of the FPGA to provide protection. Note: the i/p is 5 V tolerant
* Sync mark (line and frame triggers) conditioning for RPi capture on the FPGA.
  + A µs duration pulse from the AFM would not get captured (due to being very narrow) by RPi and so this logic makes the pulse width longer.
    - SW[17] allows you to select if you’d like to stretch/toggle the AFM trigger (‘on’) OR send it to the FIFO as is (‘off’).
    - SW[16] allows you to then select if you’d like to stretch (‘on’) OR toggle the pulse (‘off’)
    - While collecting data at NIST, we kept SW[16] and SW[17] in ‘on’ position and this allowed us to stretch the pulse by 100,000 clk ticks.
    - <https://github.com/eefer/DE2115_MultiLockins/blob/master/EightLockins/DE2115_EightLockins_top.v>
* Data collection
  + start RPI then start AFM scan, Pi waits for Y sync mark edge with ‘rpixel –frame=0’ command
  + see rpixel man-page for details: <https://github.com/wahudson/fpgadata/blob/master/trunk/man/man1/rpixel.txt>
* Figure out the number of data points based on the AFM scan rate
  + Haha, this one is on you!
* How do you know when the scan is done?
  + Rpixel will post an exit message after data collection
* RPI debug
  + <https://github.com/wahudson/fpgadata/blob/master/trunk/man/man1/rpixel.txt>
  + <https://github.com/wahudson/fpgadata/blob/master/trunk/doc/rpixel.text>
* Data transfer
  + Upload to cloud from RPi using firefox web browser
  + Copy to a flash drive
  + Use linux rsync command to copy across network
* Data analysis
  + Run pre-processing matlab script to reassemble image