

Acoustic Modem FPGA Overview

Daughter Board Front End

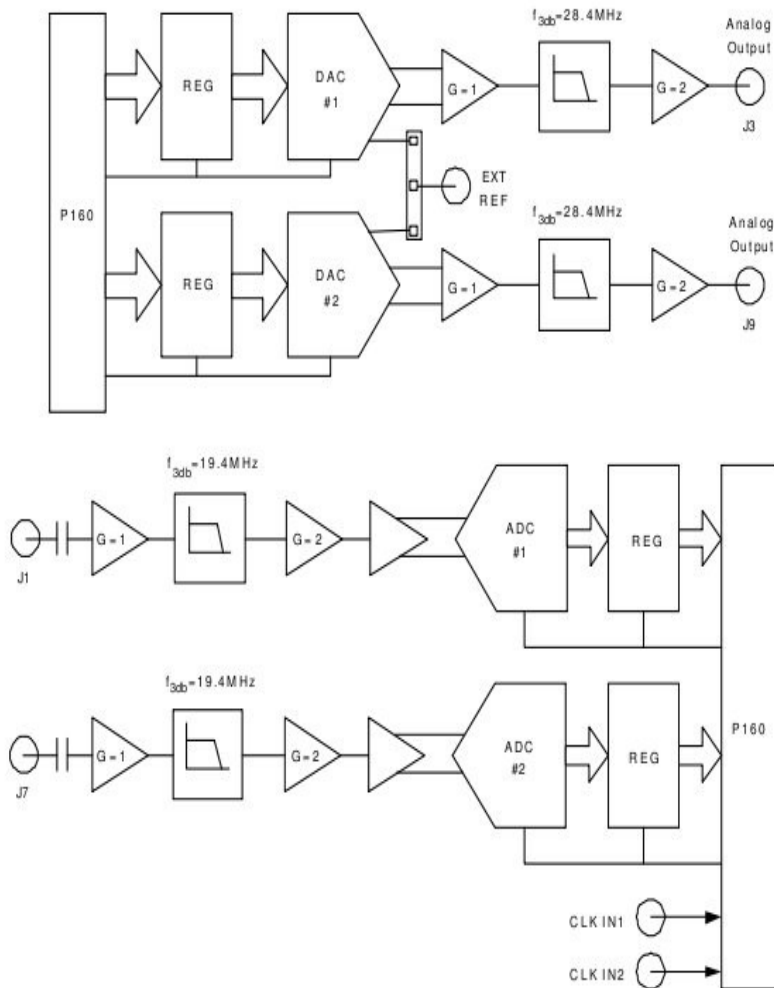
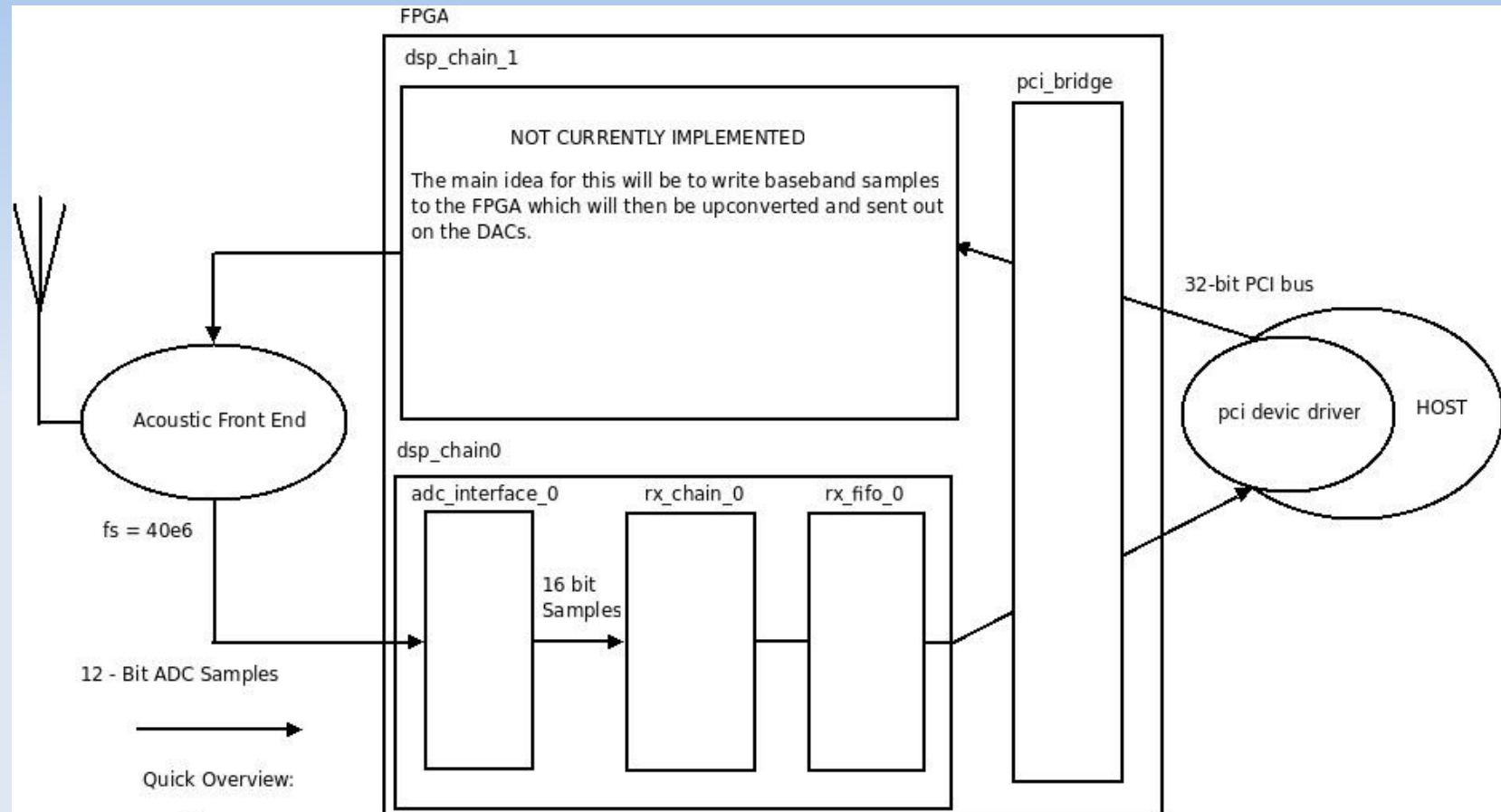


Figure 2 – P160 Analog Module Block Diagram

- Some brief information...
- Rx and Tx side of daughterboard
- Anti-aliasing filter cutoff frequency is 19.4 Mhz
- Rx sampling frequency, $F_s = 40 \text{ Mhz}$
- See the doc/memec folder for more in depth information

FPGA Block Diagram



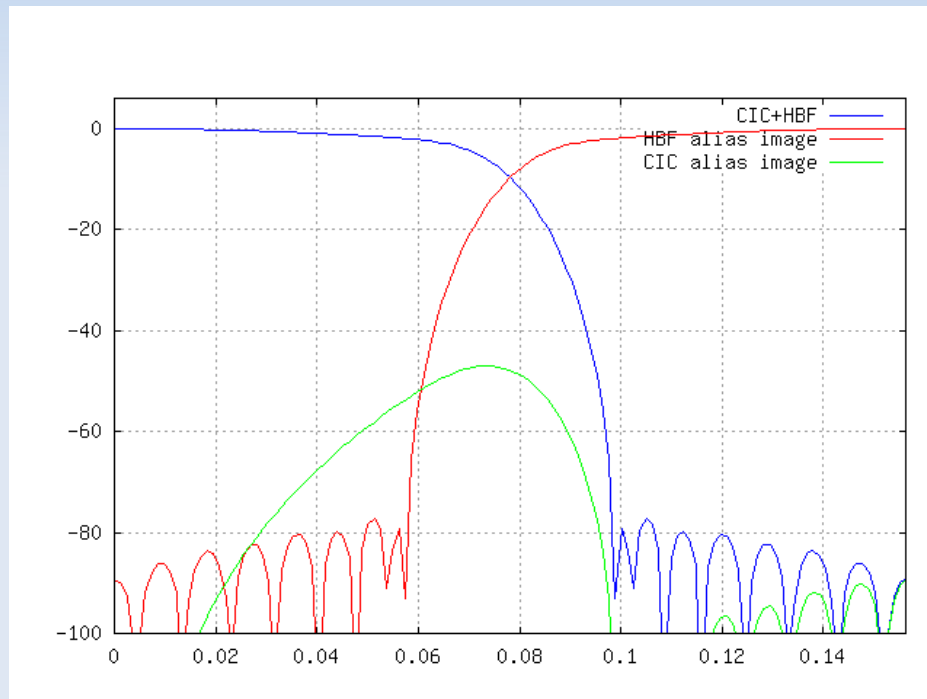
RX Side --

The `adc_interface_0` module instance takes the 12-bit ADC samples and zero pads them to 16-bits. It also has a running average dc offset corrector to remove temperature fluctuation induced dc offsets. The 16-bit samples coming from the `adc_interface_0` module instance are then passed onto the `rx_chain_0` module instance which contains the four stage CIC filter which decimates by 128 followed by the halfband filter which lastly decimates by 2 to give a decimation factor $M=256$. Since the sampling frequency of the ADCs are set to 40 MHz to avoid any aliasing (anti-aliasing filter has cutoff of 19.4 MHz) we end up with a bandwidth of $40e6/256 = 156250 \rightarrow 156.25$ kHz. The frequency response of these filters can be seen in the `cic-octave` folder. The decimated samples are then sent to the `rx_fifo_0` module instance where they are written to a FIFO. This FIFO is then read from by the Host computer via the device driver.

TX Side --

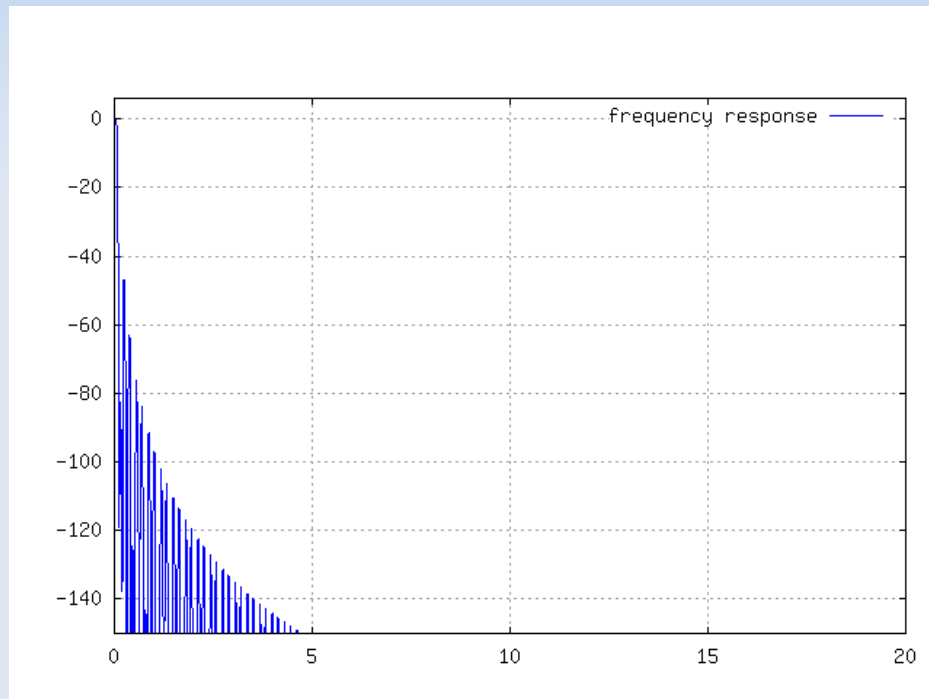
To be implemented...

DDC Filter Transfer Function (0-1 MHz)



- DDC filter transfer function, CIC+Halfband filter total decimation: 256 (CIC: 128, HBF: 2)

DDC Filter Transfer Function (0-32 MHz)



- DDC filter transfer function, CIC+Halfband filter total decimation: 256 (CIC: 128, HBF: 2)
- These plots were created with an Octave script which is located in the doc/cic-octave/cic-octave-acoustic folder.

Open Source Cores

- All modules used in this project are open source including the ones that were written in house for the project.
- The two main open source projects that are being used in this project are the pci project from opencores and the cic and halfband filter cores from the GNU Radio project. You can download and get more information about these projects from <http://www.opencores.org> and <http://gnuradio.org/trac>, respectively.

Testbench

- For simulation purposes, a testbench was created which used several different testbenches that came with the opencores pci project. These testbenches were modified so that we could test our particular design. This testbench can be found in the bench folder.
- For the Rx side of the design, we created a task that writes random numbers to the ADC register. These random numbers act as samples from the ADC which are then passed through the rx pipeline.
- Open source tools such as gtkwave and icarus verilog were used in the simulation. Simulation scripts and information can be found in the sim folder.
- The Rx side testbench has been fully implemented and our design has been verified.

PCI Device Driver

- Fortunately, the majority of the work for the pci device driver is already done. The opencores pci project comes with a driver for a VGA monitor controller that they were implementing which used the pci bridge. However, this driver is only implemented for versions 2.2 and 2.4 of the Linux kernel. We need a device driver that is compatible with the major OS changes that were introduced in version 2.6.
- Another hacker took the above mentioned device driver and ported it for kernel version 2.6. We are using this device driver and will be making any needed modifications if needed. This driver and its related documentation can be found in doc/pci_bridge_driver_1_1 folder.

Current Status...

- The majority of the work for the Rx side of the acoustic modem is completed. Simulation has verified that the design works. Current work is on getting the device driver to work properly.