AMRAD Charleston SDR Project
Quick Start
4 April. 2010

Welcome to the AMRAD Charleston SDR Project. This project springs from the work of John Schwacke and Terry Fox, WB4JFI, both in Charleston SC. John Schwacke started the receiver design as part of a project to track multiple dolphins simultaneously when their attached transmitters only come above the ocean surface for short periods. The receiver can process and detect across the bandwidth of all the transmitters as a single receive rather than a bank of individual receivers, each tuned to one dolphin's frequency.

At AMRAD we were impressed with the simplicity and potential for very high performance signal processing with the FPGA. John was kind enough to pass the details of the receiver design to Terry Fox and AMRAD for this project. Some of us at AMRAD have worked for years with signal processing. With modern FPGAs, the number of gates available for use is 500,000 for the bottom end chip and 1,200.000 gates for the higher end of the Xilinx Spartan 3E chips.

Early FPGAs featured a couple of thousand gates and found a lot of use as an attractive alternative to individual gate chips for glue logic in system designs. These FPGAs prepared the technology base for building chips with many times the number of gates. With more gates, these chips become attractive for implementing signal processing blocks. A whole Fourier transform can be implemented on a single chip. Repetitions of the FFT butterfly block can be run in parallel yielding impressive speeds compared with those of a single DSP processor since each butterfly runs at the same time as all the rest. In a processor based system, butterflys are executed one at a time. This gives rise to what some people say is a 32 fold speed-up of an FPGA over a DSP processor. This assumes a lot about speeds inside the overall processes and processors so we must use this and other such numbers with caution. Nonetheless, the FPGA will hold a significant speed advantage over the DSP processor and that is why AMRAD is pursuing the Charleston project.

We built 10 receiver boards on a Saturday using a stencil to deposit the solder paste on the pads of the PC board, a magnifying glass and tweezers to place parts and finally, a toaster over to heat and reflow the solder paste into solder at each joint. We withheld placement of the larger LPF and transformer parts and the connectors. Those were placed and soldered by hand with a soldering iron after the toaster oven heating. A few of the very close leads on the ICs had solder bridges between adjacent leads and these were cleaned up with a soldering iron and solder wick. In the end, all 10 receivers worked. We started with each builder doing his own board. After 3 boards we switched to having a group of boards aligned and going over them all, one component type at a time.

A second group build has been done but debugging has not yet been finished.

We made arrangements with Digilent to buy a group of the Nexsys II boards so on the same weekend, everyone bought a Nexsys II for their receiver.

Charleston PCB Guided Tour

For a quick tour of the receiver, go to the file blockdiag.png.

The antenna enters on the left at RF In.

From there it can go one of two ways.

The upper path goes through the LPF and the Variable Gain Amplifier. This is the path normally used for HF radio and includes the low pass filter and the VGA amplifier.

The lower (bypass) path goes into the AFEDRI8201. This is the chip where some serious work gets done. Refer to the data sheet for the AFEDRI8201. By using this path, the Charleston receiver can be used on an alias in the VHF band while you provide an external filter that removes any baseband signals.

The signals go through a transformer and then to the AFEDRI8201 chip where the signal goes into the A/D converter. After that, all the signals are processed as digital signals.

One part of the chip to understand is the CIC filter. This little box is like magic in that it takes a high data stream in and ends up with a smaller data stream out. It is done digitally and an analog version would use a bandpass filter and another A/D converter. We included two files on this part of the design. First is the file CIC1.pdf and the second is CIC Understanding.pdf. If you get very deep into programming the FPGA you should understand this chip and the parameters sent to it to set the center frequency and filter settings and how the data rate to the FPGA is reduced.

The chip also includes several FIR filters that are programmed to provide the bandwidth of the signal path going to the FPGA. Just keep in mind these filters do their job digitally and do not operate on the analog signals. These filters are pretty well covered in the data sheet.

Once the AFEDR18201 finishes its processing, the data is sent over to the FPGA on the Nexsys II board. The Nexsys II board includes a USB interface to the FPGA. The USB carries commands from the PC to the Nexsys II and to the Charleston receiver board and to the FPGA. The USB carries the output from FPGA to the PC. The Auxiliary Digital to Analog converter in the AFEDRI8201 is used to control the gain of the Variable Gain Amplifier in the upper path.

An extensive amount of data and files on the Nexsys II is included in the Digilent Items sub-directory.

Building the Charleston Receiver Board

You will find the data for building the board in the BOARD sub-directory. The PC board was laid out using FreePCB. The files needed to run that are in the FreePCB sub-directory. The FreePCB file for the PC board is in the CharlestonBoard sub-directory. Included are the Gerber files and .png files to see each part of the PCB. Also, the zip file CharlestonV1.0 can be sent to your favorite board maker to make PCBs. We used this file with PCB-POOL with good results. Included with the order was a free solder paste stencil which really helped with the board assembly.

We bought almost all the parts from Digikey. The exceptions were the 4 voltage regulator chips which we bought from Mouser, the LPF and input transformer which we bought direct from Mini-Circuits. If you are going to build only one board, you will run into minimum quantity limits for many of the parts that will require you to buy more parts than needed for one receiver board.

We bought Amtech SynTech-N solder paste from Solder and More at solderandmore.com. We found this to work fine and it lists it as not needing refrigeration. Of course, it is listed as lasting twice as long if it is so we keep it in our refrigerator. This flux in this paste stays tacky for 8 hours so you have time to manually apply the parts, inspect etc. before the flux loses its tackiness and the parts start falling off before the reflow cycle, We used a toaster oven as a reflow oven for this first group of boards. For our next group, we plan to set up a techFX temperature controller from

http://thesiliconhorizon.com/store/silicon-horizon-techfx-reflow-making-your-reflow-oven-p-48.html

While the toaster oven alone did a satisfactory job, we were concerned about getting repeatable control over each reflow cycle. We think this controller will help us achieve this.

We took some pictures of our group build of the receiver boards and they are included along with board pictures in the Pix sub-directory.

Once built, we fabricated a power test fixture using an old floppy drive cable. We had some colored spaghetti that we slipped over the wires so we would clearly see black for minus and red for plus. This is no time for a reversed polarity. Check your work carefully at this point. We then ran up the lab supply which was current limited to around 500 ma. Observe the receiver current draw is about 230 ma. We did see a couple that were much lower and we needed to touch up the soldering of the voltage regulators. We did not see any shorted ones but now was the point to find and fix that.

First Signals on Your Charleston Receiver

Install the Charleston receiver board on the Nexsys II on the two of four connectors closest to the side with the USB, RS232 and VGA connectors. Connect the Nexsys II to one end of the USB cable and the computer to the other end. Now the computer will send power to the Nexsys II board and the Charleston receiver board and transfer data in both directions with the Nexsys II.

At this point the receiver parameters have not been set and the FPGA is not programmed. To start this process, the Nexsys II needs to be started up using the software in the Nexsys II/Adept sub-directory. We have had problems using the version 2 of Adept so run Adept Ver 1.10. on Windows. It is the ***DASV1-10-0.msi*** file that when run will install Adept to your PC. This will install the program ***Export*** as part of Adept.

> THE PROBLEM
> Adept Ver 1.10 sets the serial number of your Nexsys II board somewhere into a file. If another Nexsys II board is plugged into the computer, it does not recognize the new board and will not work.
>
> THE FIX
> Power on and plug in your Nexys2 500k, and run ***Export***. Under the ***Edit*** pull down menu select ***Configure Communication Module***. A window will open and you should see your modules listed. Click on and ***Delete*** all module entries from this table. Next click on ***Add Module*** and go to the ***USB*** tab, click on your ***Nexys2 500k*** from the list, click ***Add***, then click ***Done***. This should have your module identified so you can ***Close*** the ***Communication Module*** window. Once this is done, unless you connect up another Nexsys II board to this computer, this should remain correct.

Now, to install the firmware onto the Nexsys II board run ***Export.*** Then click on ***Initial Chain***. You should see two boxes appear in the upper right window. The upper box should say FPGA and the lower box ROM. Click on the box next to FPGA to indicate that it will be bypassed for loading and that the data load will go to the ROM. Click on Browse and find and open the file ***sdrif500kIQ.bit*** (An earlier version works fine but the I and Q channels are reversed.) You should be back at the ***ExPort*** window with the file ***sdrif500kIQ.bit*** showing in the window next to ROM. Now the button ***Program Chain*** should no longer be grayed out. Press this button and the ***Export Chain Programmer*** window should pop up to indicate the file is beeing programmed into the ROM on the Nexsys II board. Another window should come up indicating the programming was successful. This process has placed the file ***sdrif500kIQ.bit*** into the ROM. Now whenever the Nexsys II board it powered up, the ROM contents will be loaded into the FPGA.

You should be able to run the program ***specview.*** First, the dll files ***cygwub1.dll***, ***cygfltknox-1.1.dll*** and ***cygfftw3-3.dll*** need to be copied into ***Windows/System32***. Then you can run the program ***specview.exe***. The dlls and exe are found in SpecView sub-directory. A window on the PC should open and the spectrum should be seen. Pull downs are available to set the input direct or through the LPF and LNA. Other pull downs allow you to set ADC Gain, LNA Range, LNA Mode and LNA Gain. Now you are up and running.

Programming the FPGA for the Charleston Receiver

Once you have built the Charleston receiver board, have it connected to the Nexsys II board you will need to program the FPGA to make a working receiver.

A basic book on Digital Signal Processing is provided in the DSP Book sub-directory. It can provide a useful understanding of the underlying concepts of DSP and a place to go to when looking into some area. Of course, Google on the internet can turn up all sorts of information. We usually try to find a concept described at least in two places to fill in the bits and pieces need to understand an idea.

We have been using the free FPGA programming software ISE Webpack on the Xilinx web site. It is currently in Revision 11.1 for the DVD from Xilinx and then upgrades to 11.5 over the Internet. We are concerned that upgrading may spoil interoperability between Charleston developers. For now, we do not use the upgrade when it asks for it and stay with version 11.1. If you have a high speed connection, you can download ISE Webpack without getting the dual layer DVD. If you end up with version 11.5 or later, it will probably not be a problem, just be aware of it.

Xilinx includes some helpful videos on using the ISE Webpack at their web site. They are in the Videos sub-directory of this CD. A copy of the Windows media player is included that does play these videos. One video is an ISE Webpack Quick Start, good to view before starting your programming. The video FPGAs for Signal Processing is good to get an idea about signal processing but keep in mind that some of the functions and capabilities may not be part of the free version of ISE Webpack. Another video is on using a Partition capability in ISE Webpack to take functions and pack them up into a separate partition and avoid reinventing and testing that function again.

Xilinx would like to see everyone using their biggest and best software tools and they include several of these but then they time out after 30 days of “Free Evaluation”. If you steer clear of using these tools, we think you can program the Charleston receiver with just the free portion. However, at this point, it is only speculation as we have not completed anything beyond some simple FPGA functions using it.

We have included several courses on FPGA design using the Xilinx FPGA and Verilog. An important file on using the ISE Webpack is in the Verilog Classes/SJSU/lab1. The course instructor has written up a good quick start description of ISE Webpack.

We have also been using ModelSim PE Student Edition. (There is a version for Xilinx and it is not free). By using a text editor to prepare a file of Verilog to be input and tried. This free version is not for commercial use but should meet their constraint so long as it is used for the Charleston project.

Finally, we would like everyone working on the Charleston project to join AMRAD. You will get our newsletter which will include articles on the progress with Charleston. As a member you get access to our email reflector. Send an email to

mailman-request@amrad.org

containing just the word "help" in the body and you will get instructions about our email reflector. From there you can see how to subscribe, unsubscribe, etc. with the reflector.

Frank Gentges K0BRA, AMRAD Director