TAPR #113 AUTUMN 2010

PACKET STATUS REGISTER

President's Corner

By Steven Bible, N7HPR, President, TAPR

Did you get to attend the 29th Annual ARRL and TAPR Digital Communications Conference? If you did not, please do yourself a favor and check out the DVD recordings now on sale at the TAPR web site. There are six DVDs in all. They are only $12 each, $15 for the Sunday Seminar including the handout CD, or $65 for the whole set. A big thank you to Ed Mellnik, WB2QHS, and volunteers to record, edit and author the DVDs.

We also extend a hardy thank you to our local hosts, Dennis Bourassa, W7IME, and Mark Walker, W7CLU. These guys did a lot of foot work, searching, and all-around jockeying to help us with the conference. We could not have done it without you!

And thank you to Rick Muething KN6KB, for putting together an excellent four-hour Sunday seminar on DSP Short Course. Rick created a very professional course that included a CD handout to all who attended. This CD is also available with the DVD DCC set.

Thank you to Dr. Nathan 'Chip' Cohen, W1YW, CEO of Fractal Antennas Systems Inc., for being our keynote speaker at the Saturday banquet. Chip gave a very insightful talk not only about antennas, but how amateur radio plays a big part in experimentation.

The DCC DVDs do not give justice to the whole conference. The DVDs are just of the talks. What is missing is the face-to-face interaction, the demos displayed in the demo room, and the camaraderie. You just have to be there. We hope that you can attend next year, September 2011. We are working on the Baltimore, Maryland area. As plans firm up we'll announce the date and place. Until then, start writing about your project, submit a paper, and come talk about it at the next DCC.

Until then, 73,

Steve, N7HPR

###

DVB-S and DVB-T Using a USRP2

(Or What Have I Been Up To In the Shack?)

By Charles Brain, G4GUO

At the beginning of the year a USRP2 and WBX were obtained from Ettus research with the idea of experimenting with wideband data modes on UHF and above. Looking around for things to do it was soon obvious that most of the ideas I had were already being done by others, eventually I settled on implementing Digital TV. This short article describes what has been achieved so far.

Figure 1. Inside the USRP2 with WBX board fitted.

The USRP2 for those of you not familiar with it is a Gigabit Ethernet interface hooked up to a XILINX Spartan 3 FPGA with both a LTC2284 14 bit 105 Msps A/D and a AD9777 16 bit 160 Msps D/A. The WBX is a 50MHz – 2.2GHz transceiver board.

The USRP2 generates an analogue I & Q signal at baseband and that is translated by the WBX to the final transmit frequency. On receive the WBX translates the input I&Q signal to a zero IF.

There are other transceiver boards for other frequency ranges.

The DATV system at G4GUO consists of a Hauppague PVR-150 hardware MPEG 2 encoder PCI card, a 4 core Pentium 4 P.C running Fedora Linux 12, the USRP2 and RF bricks for 70 cms and 23 cms. For reception a standard Fortec Star satellite set top box (STB) is used for DVB-S and for DVB-T reception a regular U.K digital TV with a home-brew receive converter are used.

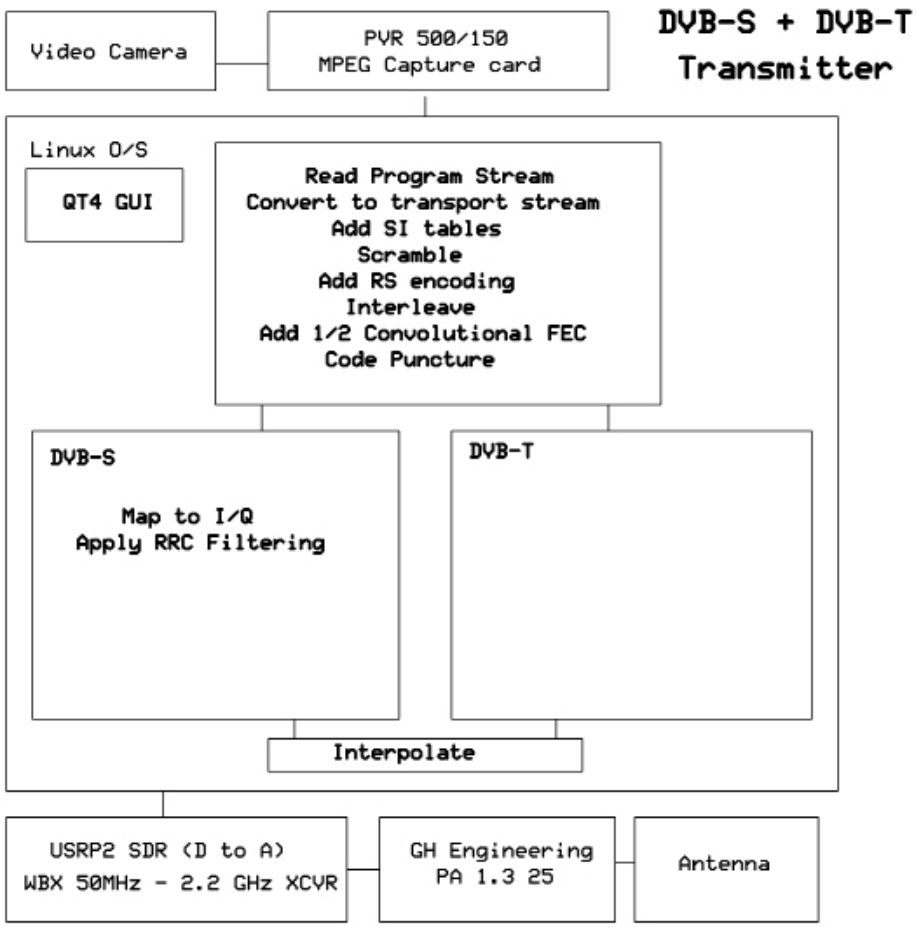


Figure 2. Block diagram of DVB Software and System)

The QT4 software development environment is being used. The GUI is a simple QT4 application which consists of a number of tabs, each tab allows the configuration of a set of related DVB parameters. It is all very basic.

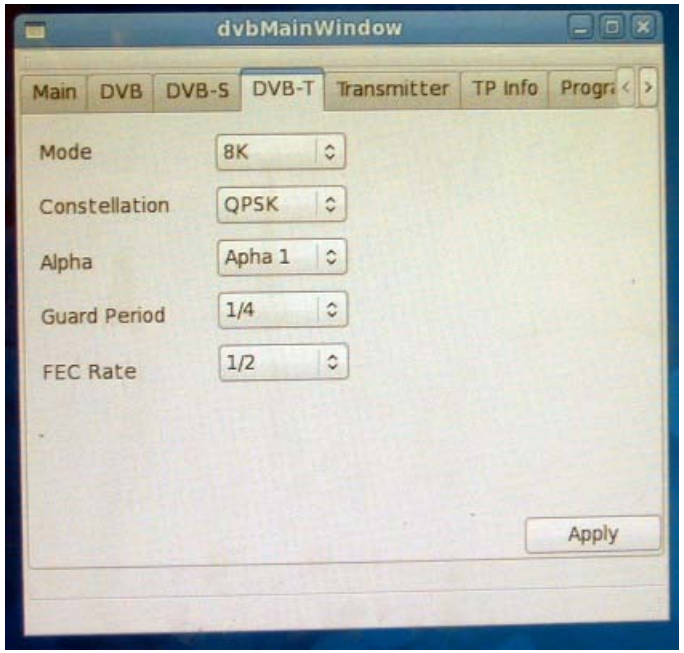


Figure 3. Example of basic GUI, DVB-T parameter tab)

A lot has been written recently in the PSR about digital TV so we won't go into any great detail on the various intricacies of the various DVB

protocols. Please look back at recent PSR articles for further information.

The software that has been developed over the last 8 months, formats all the various System Information (SI) tables required to make the signal a valid DVB signal (these contain program information, elementary stream ids and the like). It also pads out the transmit stream to a constant bit-rate using NULL packets, NULL packets are valid transport stream packets that contain no information. Every few seconds the software retransmits the program guide and the other mandatory SI tables. Transport stream packets are 188 bytes long.

As well as generating SI tables the software adds Reed Solomon encoding of the transport packets and does convolutional FEC, the mother code is the standard K=7 ½ rate code used by just about everything. For code rates greater than ½, code puncturing is used, finally interleaving of the bit stream is done by the software. In the case of DVB-S the bit-stream is then mapped into QPSK symbols before being pulse shaped and interpolated to the required sample rate.

In the case of DVB-T OFDM modulation is used, extra interleaving is added and the addition of BCH encoding for some of the pilot symbols in the OFDM waveform. The pilot symbols act as references to the receiver and also signal the format being transmitted. There is a choice of QPSK, 16 QAM or 64 QAM modulation in this mode. The software supports both the 2k and 8k mode, all the specified guard intervals and all the various FEC rates mentioned in the DVB-T standard. Currently only 7 MHz channels are supported as the sample rate conversion for that mode is fairly easy to do. Interpolate by 5 and then decimate by four. This converts the 8 MHz sample rate to 10 MHz. Code within the USRP2 then interpolates the 10 MHz up to the final 100 MHz rate used by the USRP2s D/A converters. Both 6 MHz and 8 MHz channels could be added if new sample rate conversion code was added. This is one major failing of the USRP2 as far as this application is concerned. It would be preferable to be able to change slightly the final sample rate. The required up / down conversion to get a suitable fixed final rate wastes considerable processing cycles.

It was necessary to carry out pre-compensation of the FFT bin magnitudes in the DVB-T code to compensate for the roll off of the filters both in the program itself and in the FPGA code inside the USRP2.

The sample rate conversion filters use fixed point MMX instructions. Various of the libraries that come with GNURadio were tried first but they were simply not fast enough. For the OFDM modulator MMX instructions were again used to implement the FFT. The code came from an Intel application note, naturally the Intel assembly code was in one format and that required for the GAS assembler in a different one.

The program is quite memory hungry as wherever possible lookup tables have been used rather than doing the actual computations during transmission. This was done to reduce the CPU load.

GNU Radio was not used in this project. The software talks directly to the Ettus Universal Hardware Driver (UHD). I started off writing the code in C but it soon became evident that it would be easier to use C++ because of the required standard libraries and the UHD library.

The source code for this project is available to anyone that wants it. It is not in a state to compile straight out of the box as I am constantly changing it, but it may prove interesting to anyone that wants to see how it was done or has suggestions for improvements. Some of the code modules also need renaming to more precisely reflect what they actually do.

So far experiments with standard definition video at 576i and more recently with high definition video at 1080i have been undertaken. The camera used is a JVC GZ-HD3EK HD Camcorder. Originally a standard CCTV camera was used but the very much improved picture quality of the 3 CCD JVC camera meant the CCTV camera was ditched. For the HD video a Hauppauge PVR-HD USB capture device which takes component video from the camera and encodes it into MPEG4 was used.

Interestingly the MPEG2 card uses a program stream format and the MPEG4 HD capture device uses a transport stream. The software has to be manually told what stream type to expect, however it would not be too difficult to select the correct stream type automatically. The software communicates with the cards using V4L. The video streams appear as /dev/video0 /dev/video1 etc.

For DVBT a domestic U.K HD TV had no difficulty in decoding and displaying the signal both for SD and HD formats. The sound is transmitted in stereo. As yet all tests have been trans-shack. Antennas are ready for 70 cms and 23 cms operation but as yet not time has been found to erect them.

Figure 4. High Definition Signal being received on a domestic digital TV.

For portable use a dual core Lenovo laptop is used along with a Hauppauge USB MPEG2 encoder. The camera and encoder boards were all obtained in used condition on eBay as was the DVB-T television. With 4MSamples/sec the CPU load is about 30% on a 2 GHz dual core machine.

Figure 5. Laptop based "portable" setup.

The system output power is about 5 watts on 23 cms and 20 watts on 70 cms. The output power is limited by the linearity of the power amplifiers. The power amplifiers are external to the USRP2 and are based on Mitsubishi modules and an experimental phempt amplifier that was built for 23 cms, PUFF was used to design the matching circuit. The extra amplifier is needed for 23 as the output of the USRP2 is somewhat lower on that band.

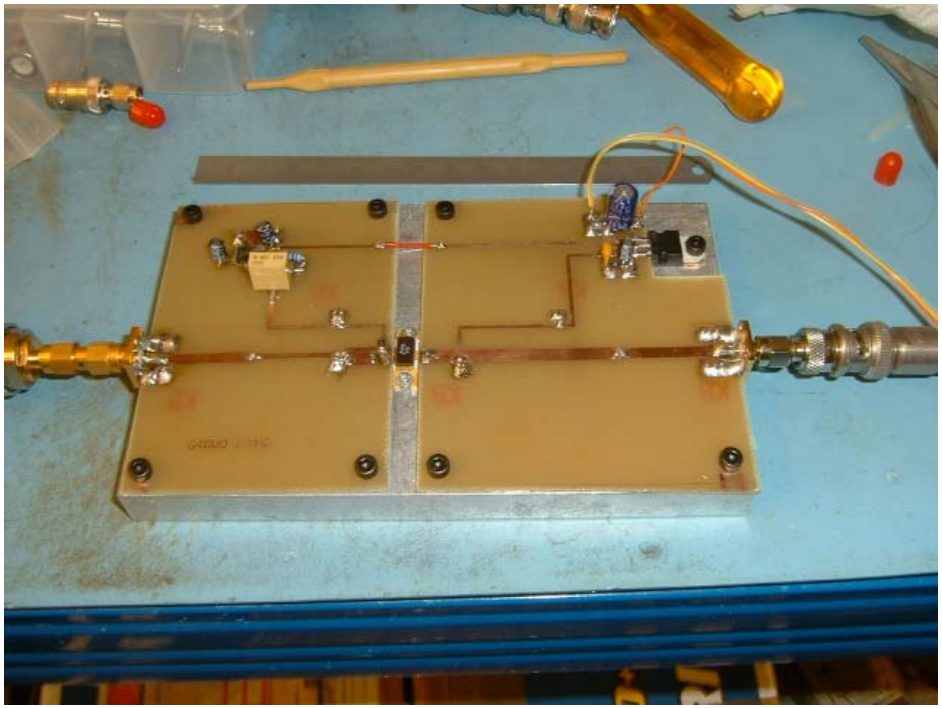


Figure 6. phemt amplifier under test.

As an alternative to using a domestic TV it is possible to use one of the Linux media players like Xine and either a satellite receiver card or a terrestrial receiver card. This has not been tried with a receive stream but the DVB transmit stream has been fed into Xine using a fifo and it has been able to decode and display the signal (both MPEG2 video and MPEG4 video).

For receiving the DVB-T HD a small receive converter was built that mixes the receive 23 cms signal with an 800 MHz LO which has the effect of translating the 23 cms band down into the domestic broadcast band. The LO for the receive converter was a re-tuned microwave xtal multiplier from an abandoned project of yesteryear. There is not enough space on 70cms to transmit DVB-T, which is a shame as the domestic TV will easily tune down into the band.

For 70cms DVB-S reception a modified surplus DirecTV adapter is being used, these can be picked up on eBay for around $5. You need a preamp in front of them but at 70 cms that is not much of a problem.

The highest data rate that has been transmitted so far on DVB-T has been around 15 Mbits/s. This appears to be limited by the maximum encoding rate of the MPEG encoder. To achieve a higher rate multiple video streams would be needed. This would involve modifying the code slightly. The highest rate on DVB-S that the software is capable of is 6 M Symbols/sec.

As well as video and sound DVB supports data services but these have not been implemented.

In the future it would be interesting to implement DVB-S2 and DVB-T2, the low density parity check (LDPC) codes look interesting. I have made a start on some of the DVB-S2 protocol but I am a bit busy with other things so I have not progressed very far.

Of course this project has only covered the transmit side, the receive side is somewhat more difficult, however some of the basic building blocks (like the Viterbi decoder) have already been written for other projects. With the low cost of receivers for digital TV it does not seem time effective to write code for the receive side at present. That may change if non standard TV modes start to be deployed.

It may have been better to have written this software to run on an FPGA but that would have been rather too complex a project for a novice, maybe when more experience has been gained with FPGAs an attempt at moving the code to an FPGA will be made.

The Internet, Linux community, GNURadio community and fellow digital ATVers have provided immense help in providing tools and encouragement for this project. There is a lot of non Amateur video software available out on the Net if you look for it. The transmission aspect of digital TV is only one very small part of course so it is an aspect of the hobby that has wide appeal.

One of the biggest problems encountered was understanding the format of some of the MPEG2 descriptors. Eventually this was solved by comparing tables created with the project code against known working binary files created by other people (who knew what they were doing).

The next step will probably be to get the antennas up. The station is located only a few feet above sea level and as there is little local digital TV activity so it might be a while before a proper two way QSO takes place. I have no plans for 3D TV (yet)! No attempt has been made to write ATSC code either.

For further pictures and diagrams please look at the blog http://www.g4guo.blogspot.com/

You will need to look at the earlier posts to see the whole story.

Further information on the USRP2 can be found here: http://www.ettus.com/

###

Join TAPR on Facebook & Twitter

By Mark Thompson, WB9QZB, e-mail wb9qzb@tapr.org

We created a new Facebook Fan page called *TAPR* at www.facebook.com/pages/TAPR/116614778354245. The TAPR fan page is linked to TAPR's Twitter account, *TAPRDigital*, so when you post an announcement on the *TAPR* Facebook Wall, a link to the announcement is automatically posted on Twitter.

We encourage everyone to follow TAPR on Facebook and Twitter to learn about:

- What's new at TAPR.
- Upcoming events at the Dayton Hamvention and the Digital Communications Conference.
- Other updates like *PSR*, projects, etc.

By the way, you can access the TAPR Twitter account at www.twitter.com/taprdigital

###

Counting TNCs on the Wall

By Darryl Smith, VK2TDS

I have been doing more cleaning up in my house, and came across my TNC-1 and TNC-2. The TNC-1 has not been used since 1997, and the TNC-2 since before that [The TNC-1 had a better unscrambled 9600 bps output I was using for testing]. Anyway, they were sitting here in my garage collecting dust. So I decided to talk to the local picture framing company who put the units into picture frames. The results are attached. They look great, IMHO.

So, for those who have an old TNC, or three, and are wondering what to do with it, turn it into a piece of art.

###

2010 DCC Notes

By Christopher Poda, AE6HL

Some data and web links on a few presentations made at the conference.

I have not yet taken the time to write more detailed notes about my observations at DCC. Here are a few links to web sites that might have more details about the topics.

Some of this data should be in the Proceedings, but I do not know when that will be available from TAPR or ARRL (*Editor's Note: DVDs are available at* http://www.tapr.org/pub_dvd.html *and the printed proceedings are available at* http://www.lulu.com/product/paperback/2010-digital-conference/12297269). I do not have my copy of the Proceedings with me as I write this, and apologize for any inaccuracies.

Codec2: an Open Future for Digital Voice; Bruce Perens, K6BP

http://codec2.org/

From the codec2 web site: Codec2 is an Open Source and patent-free audio codec, currently in alpha testing. It provides good voice rendition at 2550 bits per second, and we expect that its bandwidth will be reduced with continuing development. Its voice quality and latency are an improvement over Speex, when Speex is run at a similarly low bandwidth rate.

The original motivation for its development was to provide a completely open codec to replace AMBE+, MELP, and other proprietary codecs for use in Amateur ("Ham") Radio communications. However, the codec is potentially useful for commercial two-way radio, telephony, and other applications.

http://www.rowetel.com/blog/?page_id=452

Open Source Low Rate Speech Codec Part 1

http://www.rowetel.com/blog/?p=128

SDR Cube: A Portable Software Defined Radio

Utilizing An Embedded DSP Engine for Quadrature Sampling Transceivers; George L. Heron, N2APB, and Juha Niinikoski, OH2NLT

SDR in a 4-inch cube, with LCD display; and it does not require a PC!

http://www.sdr-cube.com/

- See the note on Availability for a possible Early Bird Special ordering incentive (in October, 2010, before release in November).

http://www.dh1tw.de/sdr-cube-interview

A Simple SDR Receiver; Michael Hightower, KF6SJ

http://www.simplecircuits.com/SimpleSDR.html

Bidirectional Low Frequency Transverter (Bi-LIF) Computer Interface for Demodulation and Modulation of Radio Signals; Alex Schwarz ,VE7DXW

http://groups.yahoo.com/group/mdsradio/

http://users.skynet.be/myspace/mdsr/DADP/dadp%20hardware.htm

See http://softsynth.com for JSyn tools & SDK; this project uses Java.

WINMOR Phase 2: Demonstration to Deployment

V4 and V4Chat: A Protocol and Client for Keyboard Radio QSOs

RMS Express - A Multimode Winlink 2000 User Client Program

Several talks covered technology, coding, protocols, & client s/w for

WinMOR (WinLink Message Over Radio), introduced at DCC 2008. Significant progress has been made since then.

"The goal of the WINMOR project is to provide a sound card ARQ mode (error free delivery of data) that approaches Pactor 2 and 3

in performance on a modern PC using standard PC sound cards."

V4 refers to Viterbi encoding, 4FSK (not version 4) a sound card protocol.

The client s/w is RMS Express, a multimode user clilent for accessing the Winlink 2000 network. It was used for on-air beta testing, and enhanced since then.

Rick Muething, KN6KB / AAA9WK

Victor Poor, W5SMM / AAA9WL

http://www.winlink.org

Other

Some other data that I found useful.

AMSAT has an arrangement with some s/w vendors that permits developers to use the tools via Internet licenses; and TAPR members can also use these tools, eg, schematic capture and PCB layout.

A good DSP book, recommended by several people at the DSP short course is *Understanding Digital Signal Processing* by Rick Lyons, ISBN 978-0131089891 http://isbn.nu/9780131089891 The third edition is due to be published soon (late 2010).

A few presentations were not listed in the Summer PSR:

D-Star for the Technically Curious

john@hays.org

http://dstarUsers.org

APRS on the Road

Barry, ve7vie/ wv2j

Very informative presentation by an APRS user. Not seen in the printed proceedings.

http://www.nwaprs.info/

Thanks

Thanks to all the conference organizers and volunteers, especially to TAPR, ARRL, local members, the Heathman Lodge; and to all authors and attendees.

###

Shorts

CODEC2 Slashdotted

Mel Whitten, K0PFX, reported that Slashdot has news on CODEC2 from Bruce Perens, K6BP. Read all about it here: http://news.slashdot.org/story/10/09/21/0428259/Codec2-mdash-an-Open-Source-Low-Bandwidth-Voice-Codec

###

Tickets: Open-Source Computer-Aided-Dispatch Application

Arnie Shore wrote that *Tickets*, a free open-source Computer-Aided-Dispatch application, has gained popularity in the RACES/ARES communities, especially for use in special events. The most recent version has a 'no-internet' option, targeted to those teams wanting a CAD capability for use when the balloon has gone up.

This version is available for download at SourceForge, at http://sourceforge.net/projects/openises/ - as file tickets_10_15.zip. Among the recently added capabilities is better support for mobile terminals and smart phones. APRS, LocateA, Instamapper, and other interfaces are included for location data, as well as geo-location (including reverse), driving directions, Street-view, and more.

And yes, it's free.

(Earlier this year, an article about us appeared in EMS World at http://www.emsworld.com/features/article.jsp?id=11988&siteSection=7.)

###

HInternet

Jason Spence, KF6RGF,, announced, "I'm working on a project to build a big packet data network. You can read more about it here: https://www.noisebridge.net/wiki/HInternet"

###

Packet Node Project Seeks Web Site Guru

The Packet Node Project has been around since the early 1990's, and is composed of hams who teamed up to map the amateur packet radio node network of North America. We have over 16,400 nodes in our database, representing all 50 US states, all provinces and territories of Canada except Labrador and Nunavut, and many states of Mexico. We are currently building a web site to share our data and maps with all.

One of our key team members has to leave the project because his day job is demanding so much of his time that he can no longer help us. We have an immediate opening for a ham who is a web site design and development guru. Packet experience is a plus, but is not required. This, of course, is a volunteer (non-paid) position, but those of us who are involved are having fun withit. The candidate can live anywhere in the world that has Internet access.

Interested candidates should contact Dick Sisson, w5onl@hotmail.com.

###

Metis PCB Design Challenges

By Kevin Wheatley, M0KHZ

Background

What follows is an insight into the development of the HPSDR Metis PCB – A Gigabit Ethernet card for the Atlas backplane, the design suite used and some of the tools within the suite.

Prior to this project my only experience with PCB CAD design was with Cadence Eagle, and over the years I had become reasonably proficient at a hobby level using this software, producing equipment for the shack.

Due to the HPSDR / TAPR / AMSAT collaboration agreements, 'big boys' CAD systems are available to developers and after some discussions Mentor Graphics 'PADS' was chosen as the design suite for the development of Metis.

The initial learning curve was steep but happily rapid, thanks to the tutorials and web support, and I must admit I have thoroughly enjoyed the experience.

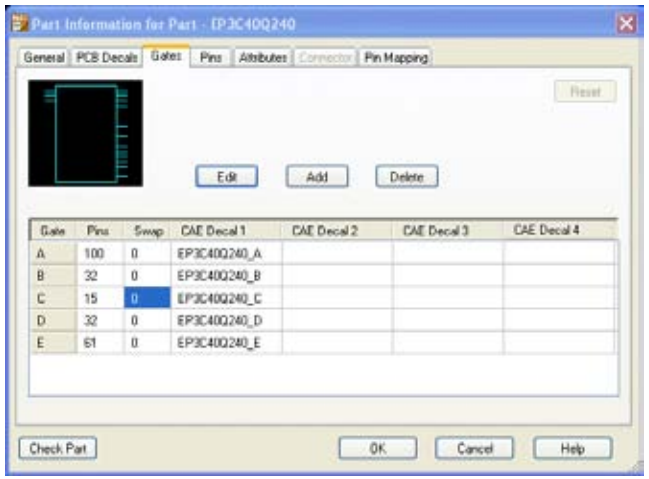
Schematic Capture

Phil Harman, VK6APH, took the lead in schematic capture (PADS Logic) and our daily e-mail exchange provided a rapid climb up the learning curve for us both, within a few weeks the schematic was taking shape nicely. It must be noted that within modern CAD systems the schematic IS the fundamental basis of design.

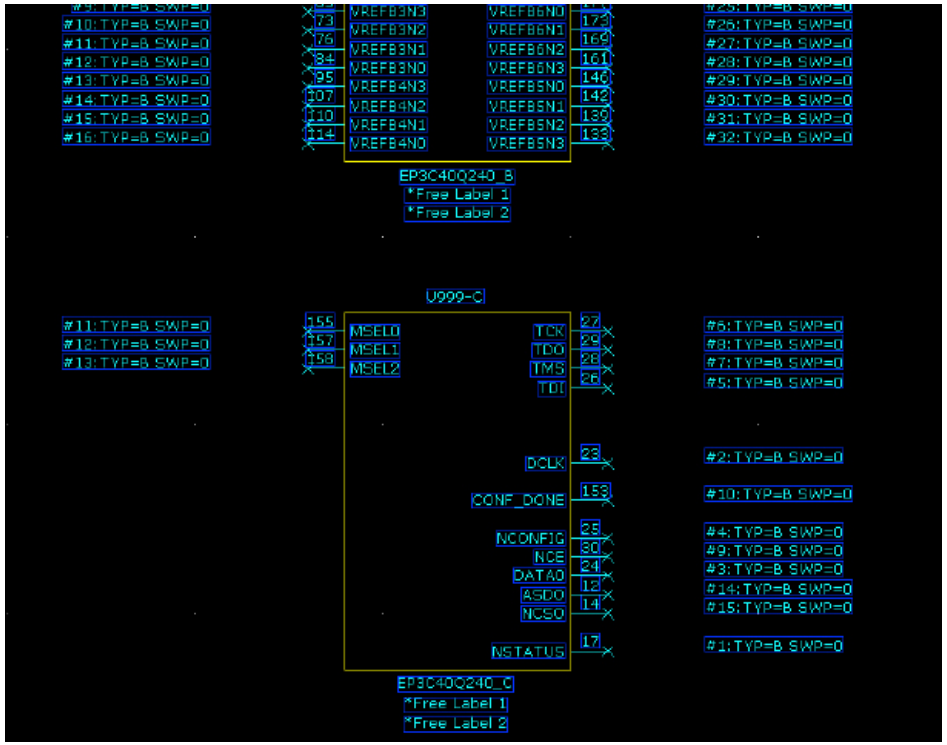
OK this statement is obvious however lets look a bit closer. Creating parts for inclusion into the schematic includes a full part definition for example PCB footprint, vendor, vendor part number, and cost. This then greatly simplifies the rest of the design, BOMs are easily created, PCB footprints are defined etc.

Within PADS a number of graphical tools are available to simplify this process, throughout this document a number of screen shots are presented providing an insight into the PADS design suite, so let's start by looking at parts creation.

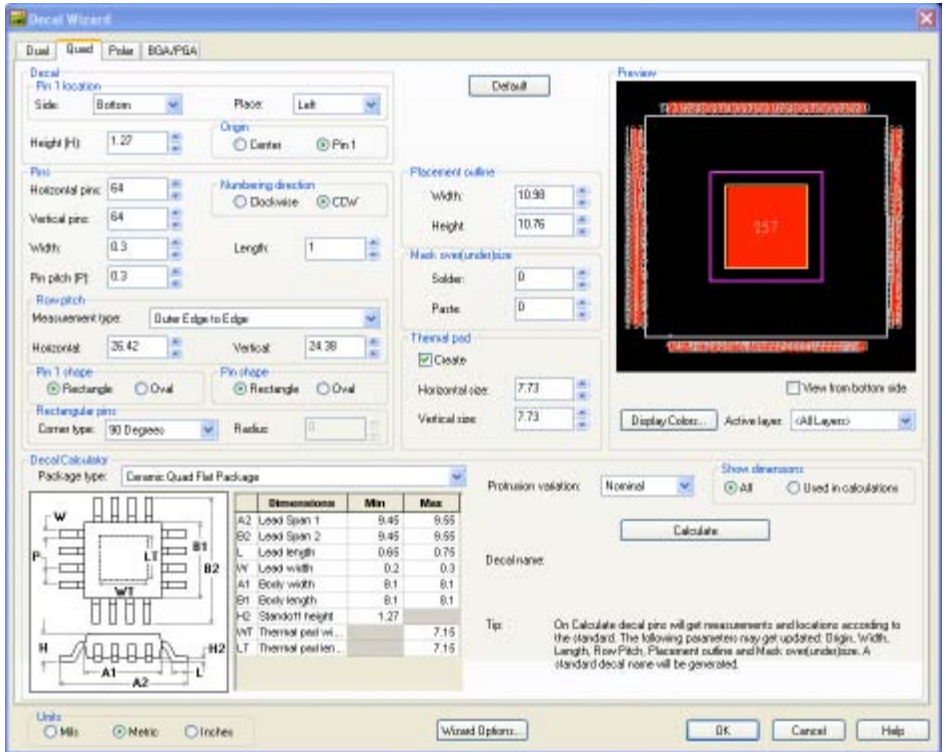
Many modern components have a high pin count, to make the schematic easier to interpret these high pin count components are often split out into multiple gates, grouped into logical blocks, this methodology has been adopted with Metis. Using the FPGA as an example, the first step for the designer is to analyze the data sheet and decide the number of gates, then create definitions for these gates:



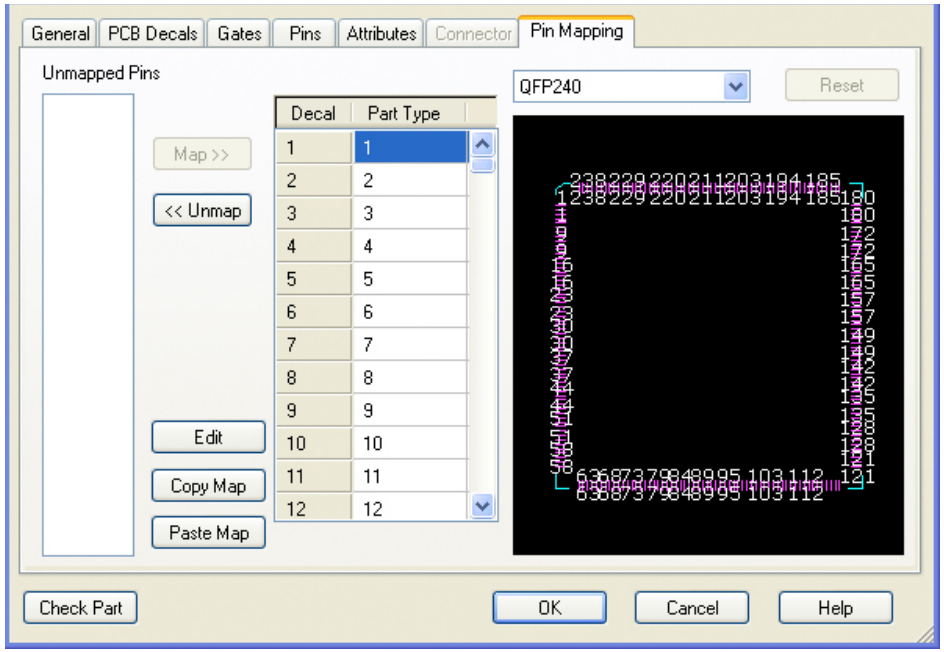
Following gate definition, each gate schematic 'picture' is created:



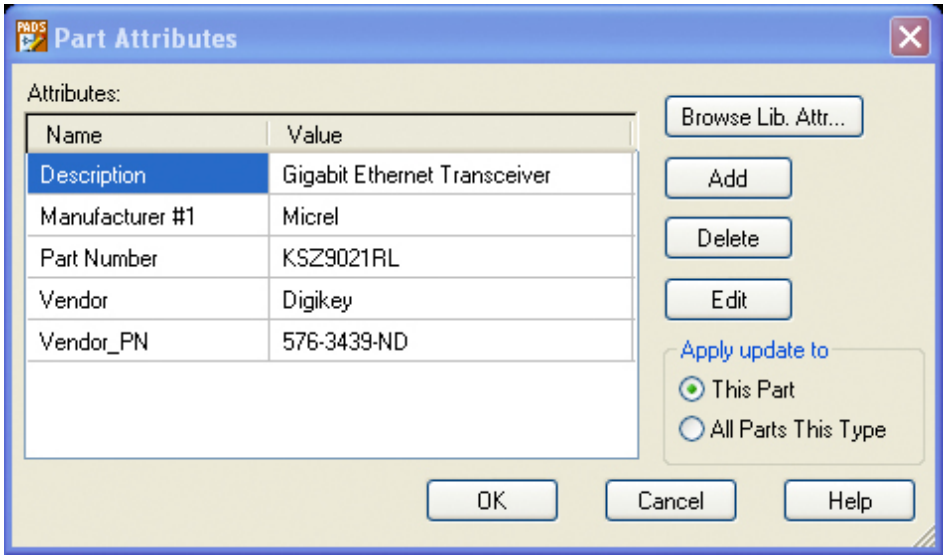
The schematic part now needs a PCB footprint to be assigned, as our designs frequently use cutting edge technology, this usually requires a new footprint to be created, as they are not available from the standard footprint libraries. Fortunately within PADS version 10.2 a new footprint wizard is included which greatly simplifies this task:



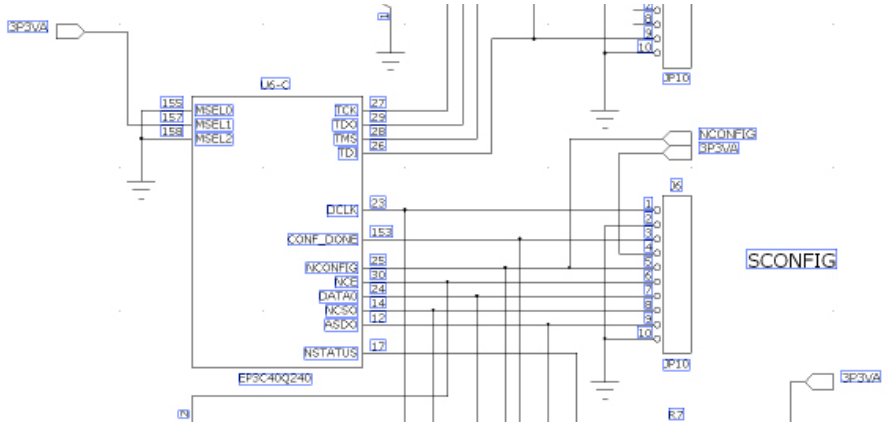
The schematic 'picture' and the PCB footprint now require to be mapped, this process ensures each schematic pin is assigned to a physical pin on the PCB:



The final step in part creation is the addition of the part attributes, this sounds a simple task but requires hours searching the web, for various vendors, availability, price and price breaks need analyzing:

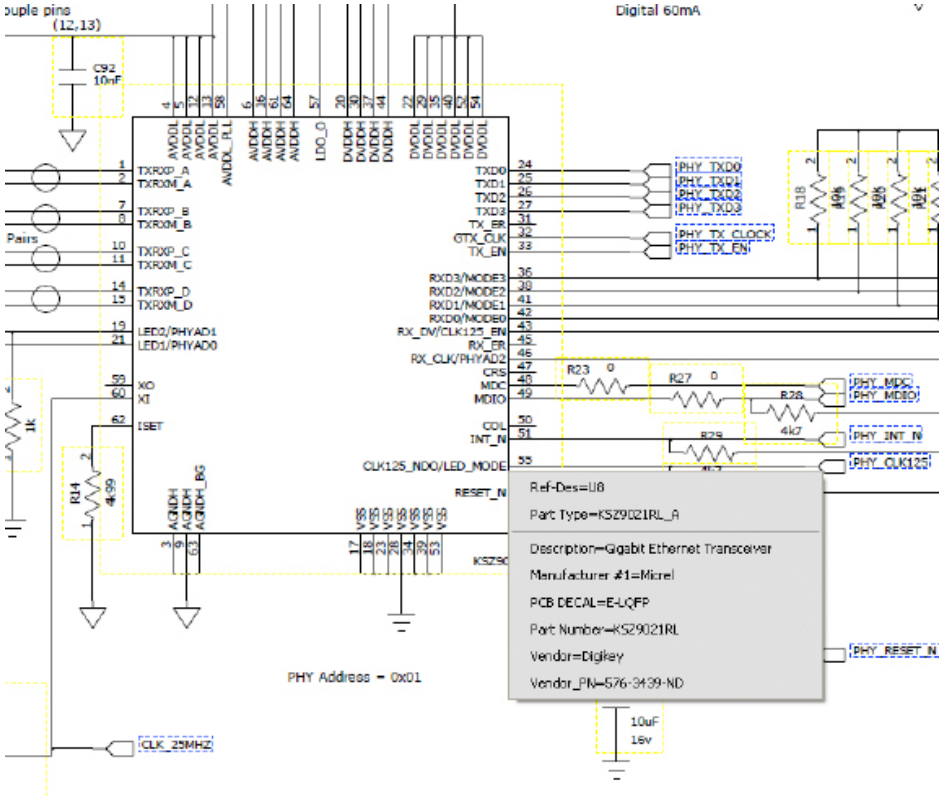


Once the part has been created and saved to a library of parts for future reuse, it can then be used within the schematic:



For complex designs such as HPSDR boards this process is very time consuming so please don't get frustrated when a concept block diagram is released and it takes months before a schematic is available for peer review!

A very nice feature of PADS is the ability to create an intelligent PDF of the schematic, where you can point and click at a component and the attributes are displayed.



During peer review and subsequent modifications to the design, a number of additional tasks are performed. The main ones being the decision for the number of layers (and layer stack-up) the design rules, for example which layers can support data tracks, clock tracks etc., which

layers are ground planes or power distribution planes, what signal lines are differential pairs or high speed data tracks, the minimum and maximum track width etc. There are a large number of options and decisions here and I was making decisions to be made based on information 'gleamed' from the web and data sheets for the components within the design. I'm sure once I become more familiar with the environment I will develop a set of 'HPSDR Rules' that we can use across future projects, and foreshorten this exercise.

Once complete the design is ready for transition into the PCB layout tool.

PCB Creation

I took the lead on PCB design (PADS Layout) with Phil chasing my heels on the learning curve, once again our daily e-mail exchange provided a rapid climb up the learning curve for us both, and within a few weeks the board was taking shape nicely. What follows is a brief insight into the boards development.

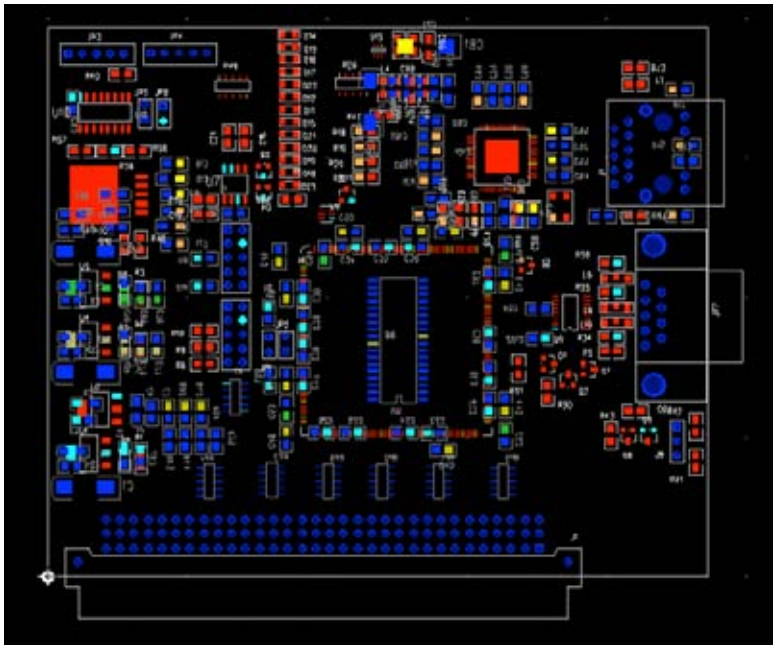
The initial opening of Layout from within schematic editor (the preferred method) presents the budding designer with a perfect picture of chaos! Every component is stacked on top of each other at a point source (x 0.0000, y 0.0000). Once the panic has subsided and the realization sinks in that nothing is actually wrong, the first task is to draw the board outline and have the layout tool distribute the components around the edge of the board, to enable parts placement, fortunately this is a very easy and quick step taking minutes.

The PADS Layout tool is a separate executable, however, it is fully integrated with the schematic tool, what this means, in simplistic terms, is that if you select a component within 'Layout' the appropriate schematic page will be opened and the component identified and visa versa. This feature is VERY useful especially when placing the components on the PCB, unfortunately to use the feature a large screen is required or a second monitor, in my case I opted for a second monitor and I squandered (my XYL's words not mine, as I considered this a sound and worthwhile investment) two months of hobby budget in one single purchase!

Now the fun starts, will the design 'fit' the board? With an appropriate grid defined within the options and 'snap to grid' checked, together with not displaying unrouted traces, initial parts placement can commence.

There are some obvious part placements for example connections to the outside world and the Atlas connector, but pretty much everything else is 'up for grabs'. I find it best not to be too critical here as once pin to pin tracking starts, components often need to be moved, and sometimes whole sections of the design require relocation.

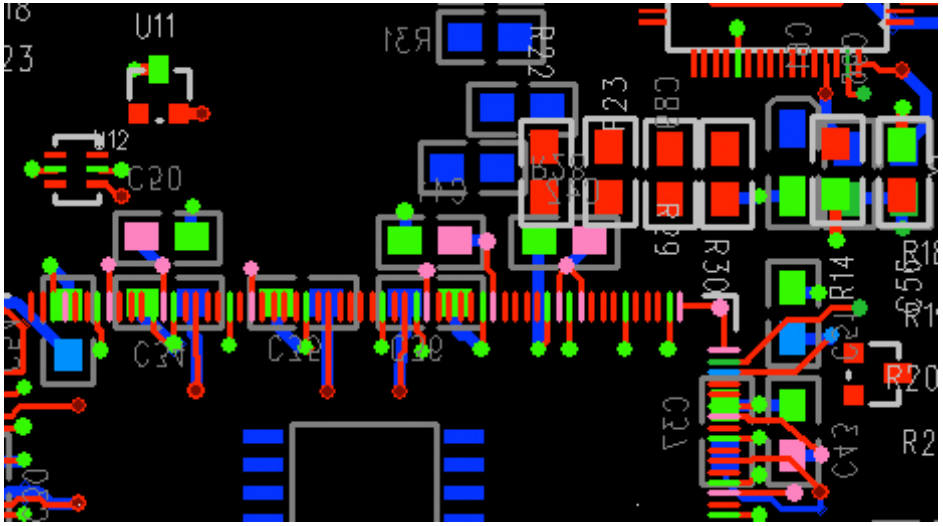
With Metis a choice was made to mount all de-coupling capacitors on the bottom of the board together with the RAM mounted directly under the FPGA, following this the remaining components were placed.



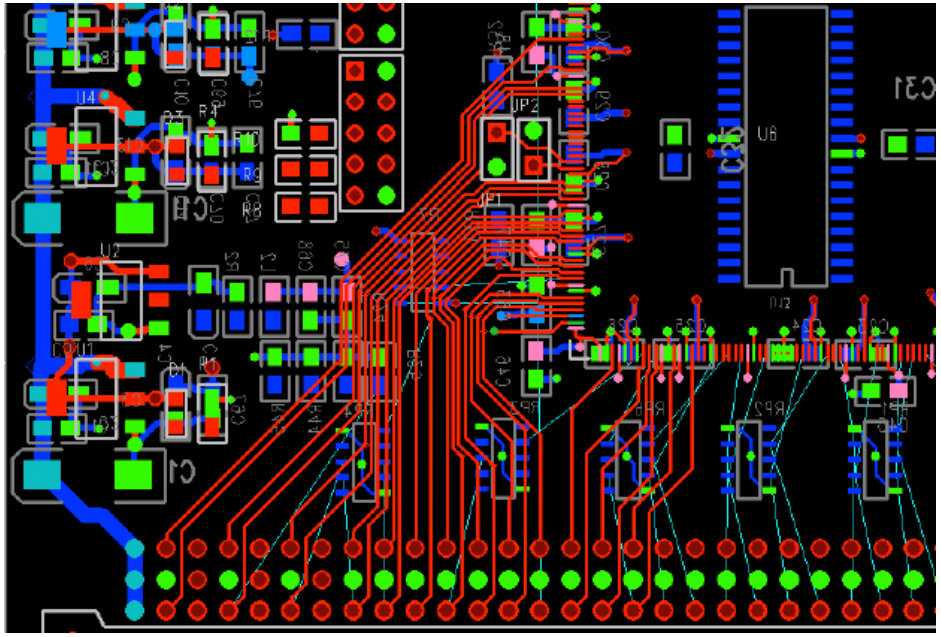
The next step is laborious but definitely necessary, as during my first attempt at laying out the PCB I naively missed out this step, only to return two weeks later, after deleting two weeks worth of effort - it's a good job I'm not doing this professionally.

This phase of the design involves tracking and punching through an appropriate 'via' for all components (top and bottom) having either a ground (analog or digital) or power supply connection. The exact position isn't vital although there are some 'norms' that assist in noise reductions and these were used. These 'vias' will eventually be connected to the inner layers previously defined, more on this later. The important point here is that when tracking is started the ground and power vias are in place and you 'track' around them, this is where I initially went wrong, and having spent hours most weekday evenings and at least two weekends laying down the tracks, I realized my mistake within minutes of attempting to connect ground signals! Did I mention this before?

To assist in this exercise, there is a very useful tool within PADS, where you can color code individual nets or groups of nets, thus DGND, AGND, 5V, 3V3, 2V5 & 1V2 nets were given different color codes, together with not displaying unrouted traces and the appropriate component pads are displayed with the required color, this accelerates this process.



Laying down tracks between components can now begin in earnest, this is where the fun really starts. Once again rather than being presented with a complete 'rats nest' of unrouted paths, the net display color tool is used to only show the nets of interest. Below is a snapshot of the Atlas bus connections being tracked.



Within PADS Layout there are three options to route a track. Manual, Manual assist and Auto. Manual is the method I am most used to using, this requires each 'stretch of the road' to be placed individually, bend by bend, you have complete control of where tracks are positioned where vias are positioned and which board layers are used. Manual assist is a method new to me and one I will use extensively from now on, basically you start from one component pad and draw a path with your mouse toward the destination pad, PADS automatically adds the track, the 'bends in

the road', checks that the design rules are still valid and even 'pushes' previously laid tracks around while preventing errors! In this mode you still control where vias are positioned and which layer to use, in a nut shell, this mode is FANTASTIC. The last mode is auto tracking where the whole board is automatically tracked following the design rules created earlier. I've spent hours playing with the design rules and auto tracking but have never been really satisfied with the end result, so I've abandoned this mode, but who wouldn't with manual Assist!

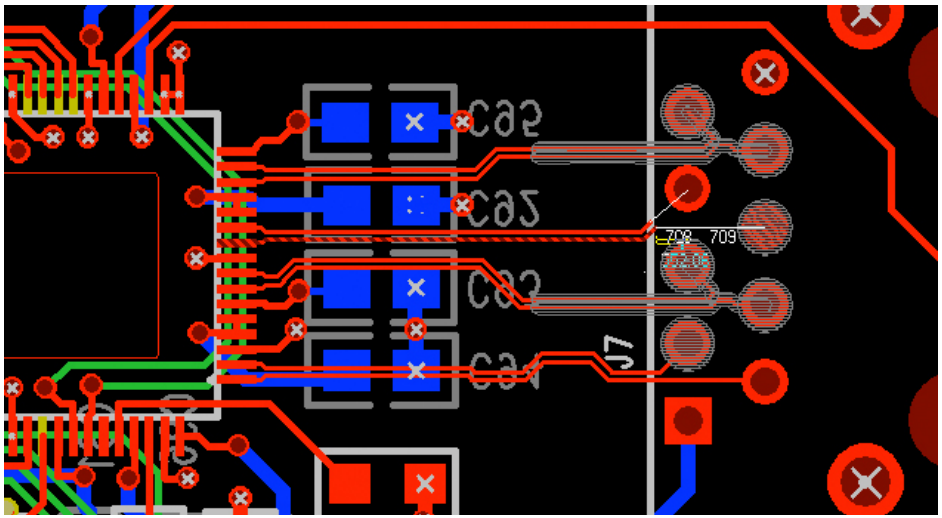
The PCB starts to look like a 'proper' board as the number of unrouted tracks start to decrease, however the Metis design required a few 'special' tracks to be laid down, namely:

<> the differential pairs between the PHY chip and the RJ45 connection to the outside world.

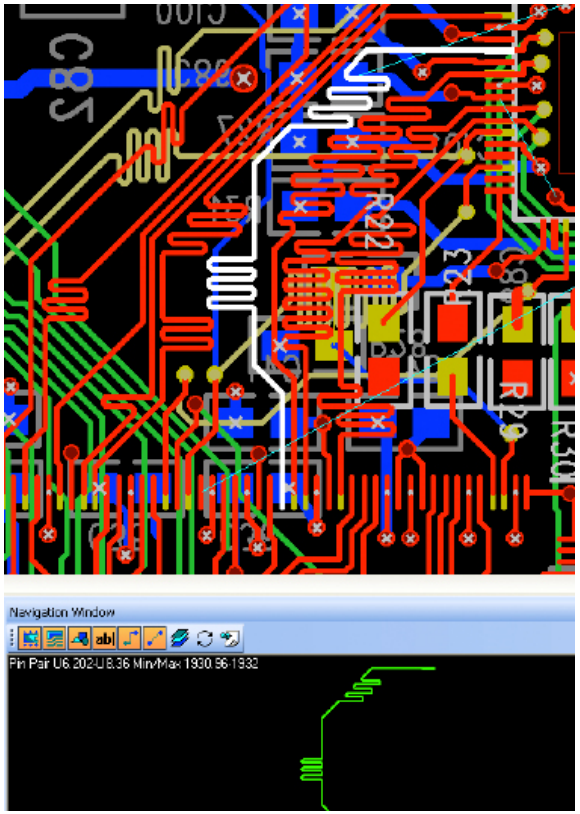
<> the Tx & Rx tracks between the PHY and FPGA, where these need to match lengths to reduce timing skew errors.

Fortunately the data sheets and PADS come to the rescue. The data sheets define the track requirements, track width and space between tracks, and third executable within the PADS suite, 'Router', provides the perfect tool for the task. This is a fun tool to use, within Layout you defined which tracks are differential pairs including track width and spacing requirements, groups of nets and length requirements, you then move the entire design out of Layout and into the Router at the push of a button.

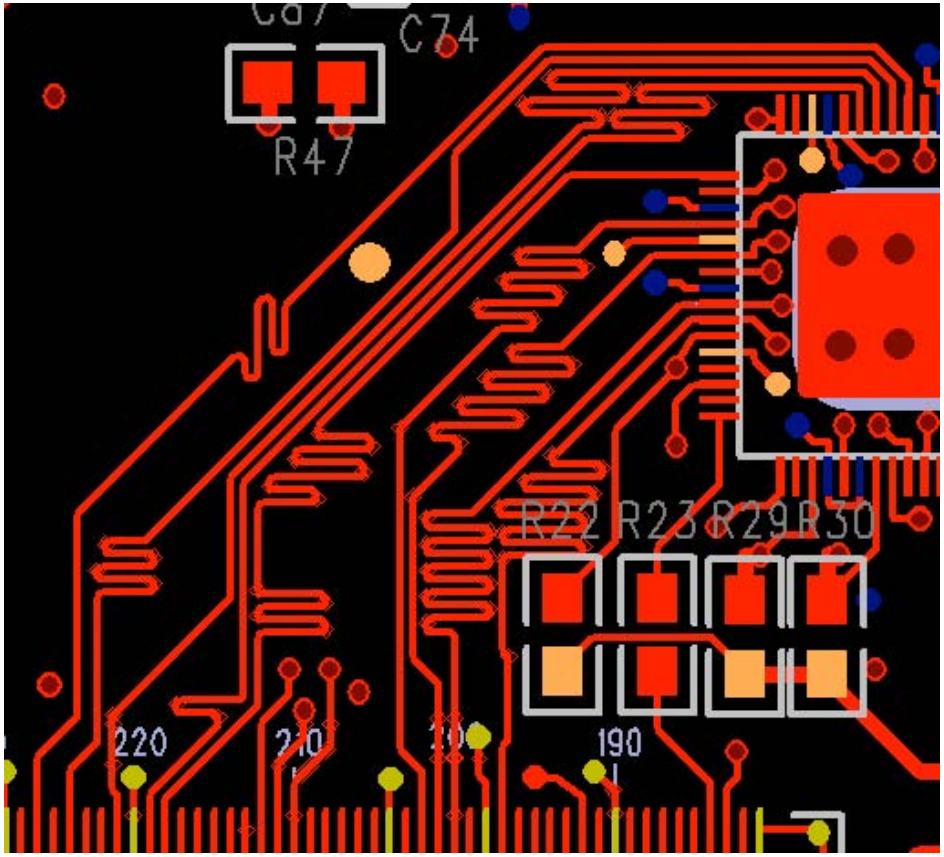
PADS Router opens and the differential pairs are tracked, in pairs by a similar manner to the manual assist method within Layout but the rules defined are applied.



Matching the track lengths for the PHY to FPGA is achieved by highlighting in individual track (or group of tracks) and selecting the tune tool. The router then adds 'accordions' to the track to ensure the correct length is achieved.



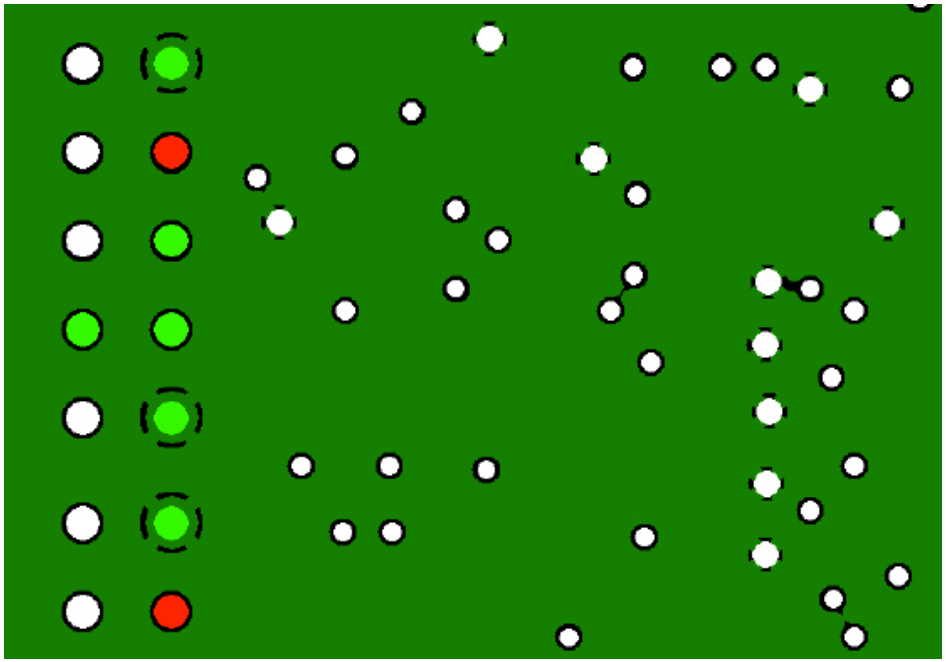
After careful selection and tuning of individual tracks, the design can be returned to the Layout environment. I've only scratched the surface of the capabilities within the PADS Router application and am keen to learn more, the tool definitely did what was required for Metis.



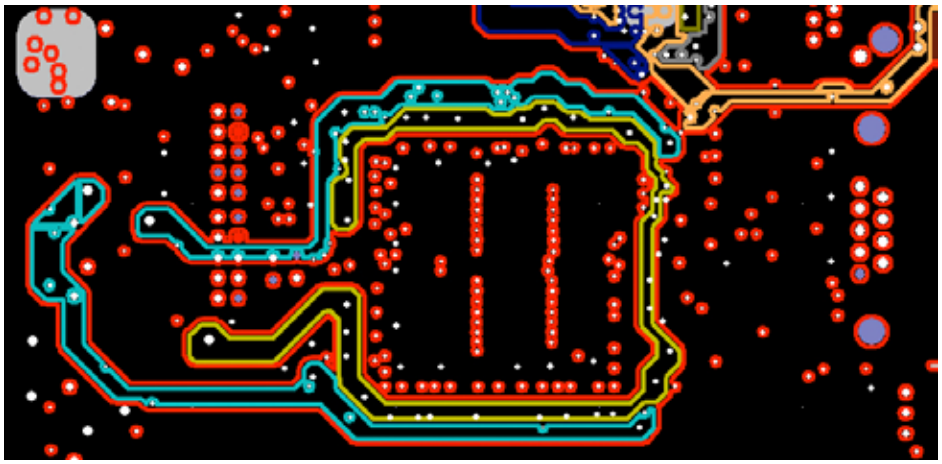
The design is really taking shape now with only a few steps in the dance remaining, once again naively I thought the job was almost complete!

Final Steps

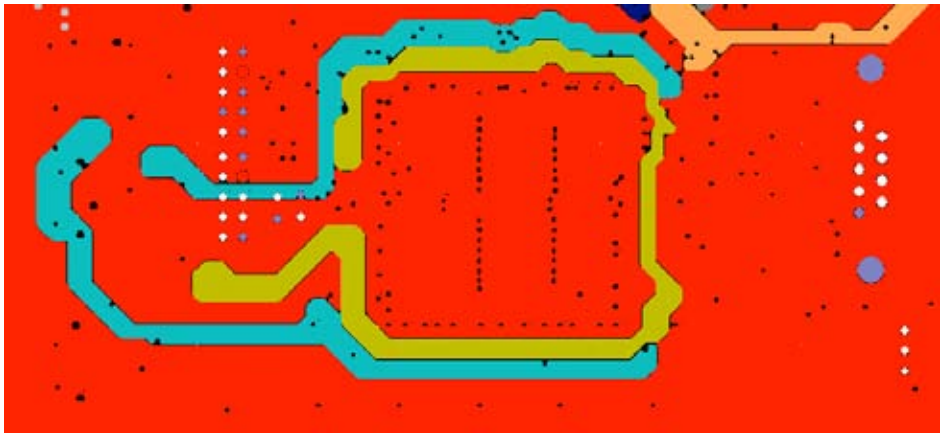
Metis has two ground planes, these are layers of copper buried within the inner layers of the PCB, analog ground and digital ground. The next step is to connect each of the previously placed vias, hundreds of them, to the required ground plane, sounds complicated but extremely straight forward. Layer selected, board outline drawn and a copper pour initiated and the PADS sorts everything out.



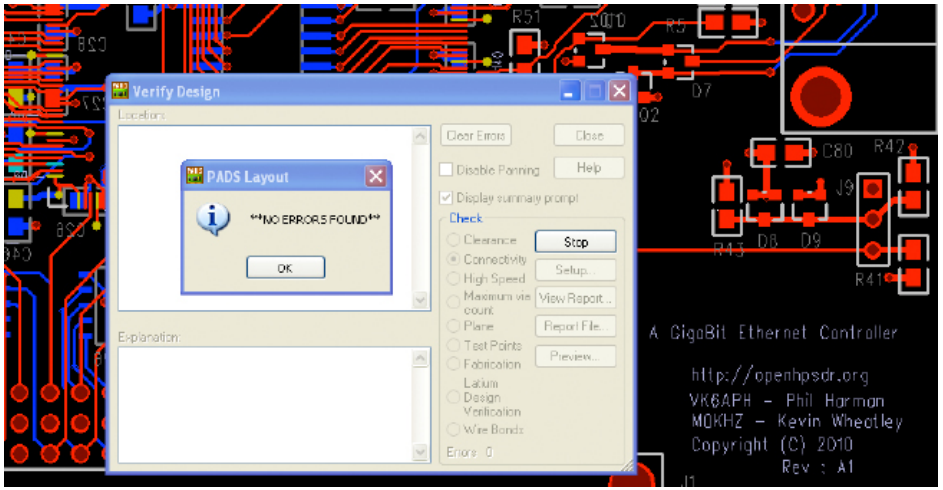
The last plane requiring attention is the power plane, once again buried within the PCB. This plane is a little different as it has a number of distinct areas for separation, so once the plane is selected, you create a polygon grouping a series of vias associated with a voltage, 1V2 for example. This polygon is then given a pour sequence number, polygon placement is continued until you have lassoed all the vias for each power supply distribution.



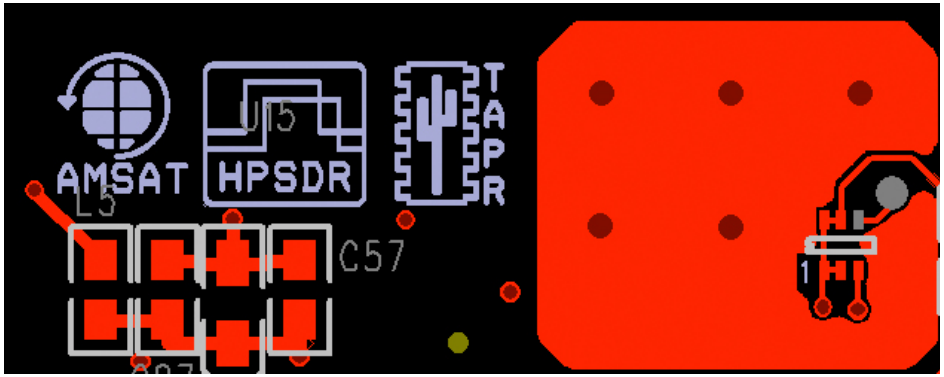
The whole plane can then be flooded with copper, each area being flooded in accordance with it's pre defined pour sequence number. This is important, as if you forget to set a sequence number, as I did on my first attempt, when the pour is initiated the last pour takes precedence and obliterates all the other polygons!



In theory if the PCB was now manufactured you'd have a working board, fortunately, once again PADS has a very useful tool to illuminate some of the guess work, a verification tool. This tool runs a series of test against the design and prompts for anything that looks suspicious, it's surprising what it finds.



If you've followed me this far, you'd be thinking the design is finished, so did I, however there are a couple of final 'house keeping' tasks required, that of ensuring the silk screen text is the same font and size, the text orientation is readable from once direction and not covering any solder pads. This a straight forward task, the very last task was to place the TAPR / AMSAT / HPSDR logos on the board, mistakenly I thought this was going to be an easy, WRONG, this step took nearly a week of 'shack time'! You would expect an expensive package like PADS to have the ability to import a bitmap (or similar) and allow placement, unfortunately this was not the case. After spending days trying different things with partial success, Phil came to the rescue with a free 3rd party tool called BMB2ASC.exe, full details can be found here:
http://www.ashevillecommunity.org/hawker/pcb/pads/bmp2asc.html

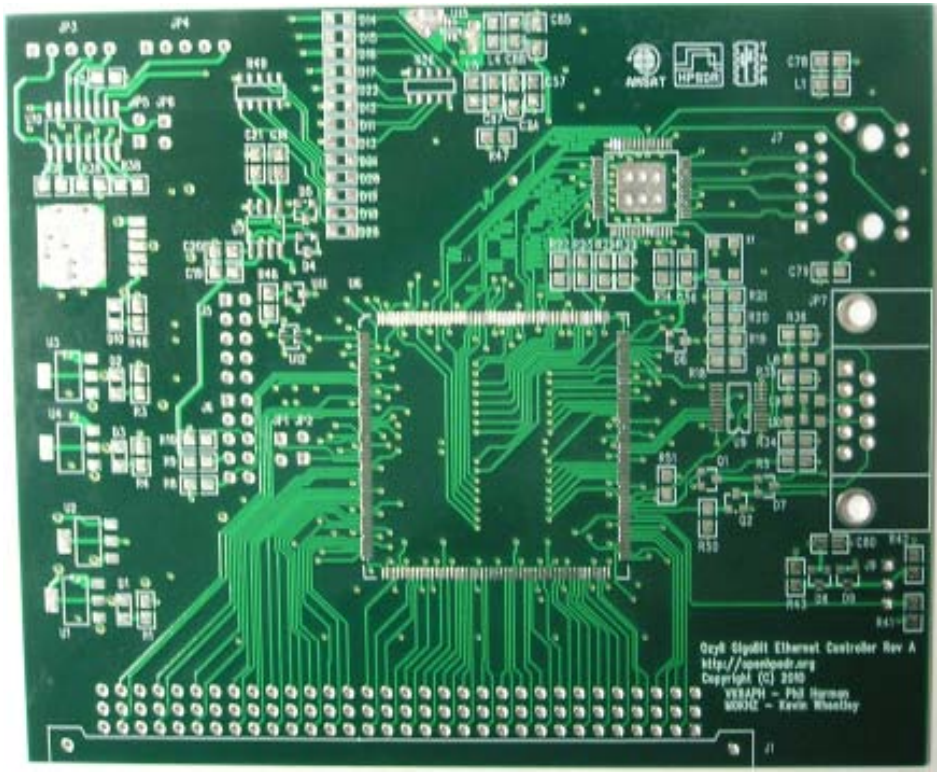


Once placed, the design was finished, a set of gerbers generated for manufacture. Alpha PCB order placed, followed by two weeks of nervous anticipation before one morning the boards arrive in the post, great joy viewing the final product.

Update

At the time of writing, the alpha board is a fully functional and stress testing proves sustained data transfers in excess of 500Mbps!

###



TAPR Election Results

The following officers were elected at the TAPR Board of Directors meeting in Portland, OR, on 23 September 2010:

President: Steve Bible, N7HPR

Vice President: Scotty Cowling, WA2DFI

Secretary: Stan Horzepa, WA1LOU

Treasurer: Tom Holmes, N8ZM

In the general election for board members, the following individuals were elected to three-year terms:

John Ackermann, N8UR

Dan Babcock, N4XWE

Jeremy McDermond, NH6Z

###

Write Early and Write Often

Packet Status Register (*PSR*) is looking for a few good writers, particularly ham radio operators working on the digital side of our hobby, who would like to publicize their activities here.

You don't have to be Vonnegut to contribute to *PSR* and you don't have to use Microsoft Word to compose your thoughts. The *PSR* editorial staff can handle just about any text and graphic format, so don't be afraid to submit whatever you have to wa1lou@tapr.org.

The deadline for the next issue of *PSR* is January 15, so write early and write often.

###

TAPR is a community that provides leadership and resources to radio amateurs for the purpose of advancing the radio art.

Packet Status Register

#113 Autumn 2010, ISSN: 1052–3626
Published by
TAPR
phone 972–671–TAPR (8277)
e–mail taproffice@tapr.org
URL www.tapr.org
Facebook
www.facebook.com/pages/TAPR/116614778354245
Twitter www.twitter.com/taprdigital
TAPR Office Hours: Monday – Friday, 9 AM – 5 PM Central Time

Postmaster: Send address changes to TAPR, P. O. Box 852754, Richardson, TX 75085–2754. *Packet Status Register* is published quarterly by TAPR. Membership in TAPR, which supports the electronic publication of the *Packet Status Register*, is $25 per year payable in US funds.

Submission Guidelines

TAPR is always interested in receiving information and articles for publication. If you have an idea for an article you would like to see, or you or someone you know is doing something that would interest TAPR, please contact the editor (wa1lou@tapr.org) so that your work can be shared with the Amateur Radio community. If you feel uncomfortable or otherwise unable to write an article yourself, please contact the editor for assistance. Preferred format for articles is plain ASCII text (Microsoft Word is acceptable). Preferred graphic formats are PS/EPS/TIFF (diagrams, black and white photographs), or TIFF/JPEG/GIF (color photographs). Please submit graphics at a minimum of 300 DPI.

Production / Distribution:

Packet Status Register is exported as Adobe Acrobat and distributed electronically at www.tapr.org

PSR Packet Status Register Editor:

Stan Horzepa, WA1LOU
One Glen Avenue, Wolcott, CT 06716–1442 USA
phone 203–879–1348
e–mail wa1lou@tapr.org

TAPR Officers:

President: Steve Bible, N7HPR, n7hpr@tapr.org
Vice President: Scotty Cowling, WA2DFI, wa2dfi@tapr.org
Secretary: Stan Horzepa, WA1LOU, wa1lou@tapr.org
Treasurer: Tom Holmes, N8ZM, n8zm@tapr.org

TAPR Board of Directors:

Board Member, Call Sign, Term Expires, e–mail address
John Ackermann, N8UR, 2013, n8ur@tapr.org
Steve Bible, N7HPR, 2011, n7hpr@tapr.org
Dan Babcock, N4XWE, 2013, n4xwe@tapr.org
Scotty Cowling, WA2DFI, 2012, wa2dfi@tapr.org
Stan Horzepa, WA1LOU, 2011, wa1lou@tapr.org
John Koster, W9DDD, 2012, w9ddd@tapr.org
Jeremy McDermond, NH6Z, 2013, mcdermj@xenotropic.com
Darryl Smith, VK2TDS, 2011, vk2tds@tapr.org
Mark Thompson, WB9QZB, 2012, wb9qzb@aol.com

TAPR is a not–for–profit scientific research and development corporation [Section 501(c)(3) of the US tax code]. Contributions are deductible to the extent allowed by US tax laws. TAPR is chartered in the State of Arizona for the purpose of designing and developing new systems for digital radio communication in the Amateur Radio Service, and for disseminating information required, during, and obtained from such research.