How the Other Half Lives

Developing SDR Software for the Macintosh Platform

Jeremy C. McDermond, J.D. (NH6Z)

About the Author

About the Author

- Licensed in 1986 at age 14 as KB7AKH
- Began to program on the Apple II at age 5
- Learned C in 6th grade (on a 512k Mac)
- Spent 15 years in systems and network engineering
- Numerous industry certifications such as MCSE, Sun, and IBM AIX
- Avoids MS Windows like the plague

The OpenHPSDR Platform

The OpenHPSDR Platform

- Direct Downconversion Receiver (Mercury)

The OpenHPSDR Platform

- Direct Downconversion Receiver (Mercury)
- Direct Upconversion Transmitter (Penelope)

The OpenHPSDR Platform

- Direct Downconversion Receiver (Mercury)
- Direct Upconversion Transmitter (Penelope)
- USB 2.0 Connected
  - Future connection via gigabit ethernet

Mercury


ALTERA
Cyclone III
EP3C25Q240C8N
J AADA00825A
MERCURY DDC Rt Rev A12
http://hpsr.org
http://tapr.org/DHL
http://amsat.org
A3
007 R
A 6921 58

Mercury

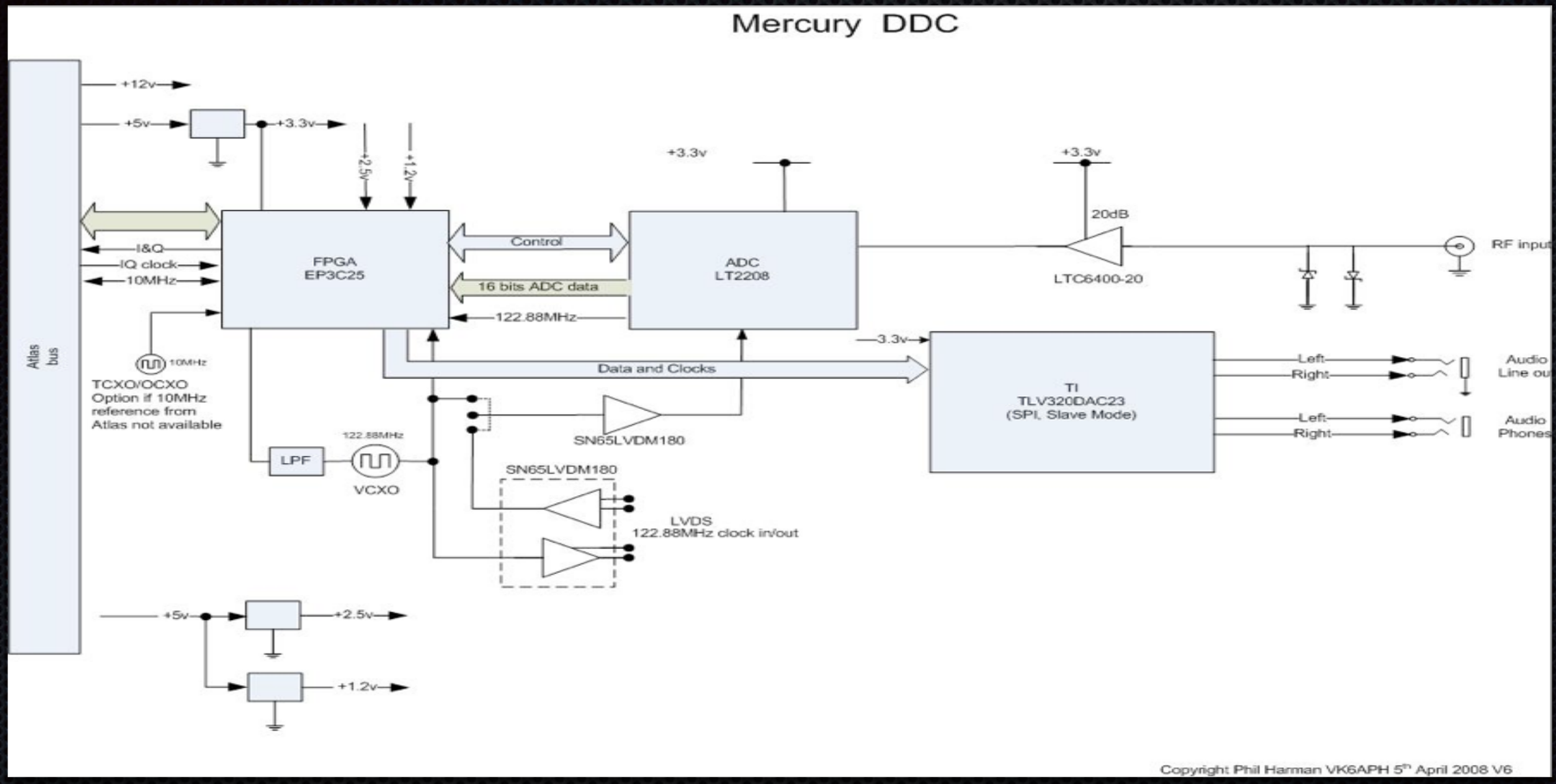
- LTC2208 ADC as close to antenna as possible
  - 130Msps 16-bit

Mercury

- LTC2208 ADC as close to antenna as possible
  - 130Msps 16-bit
- Altera Cyclone III FPGA

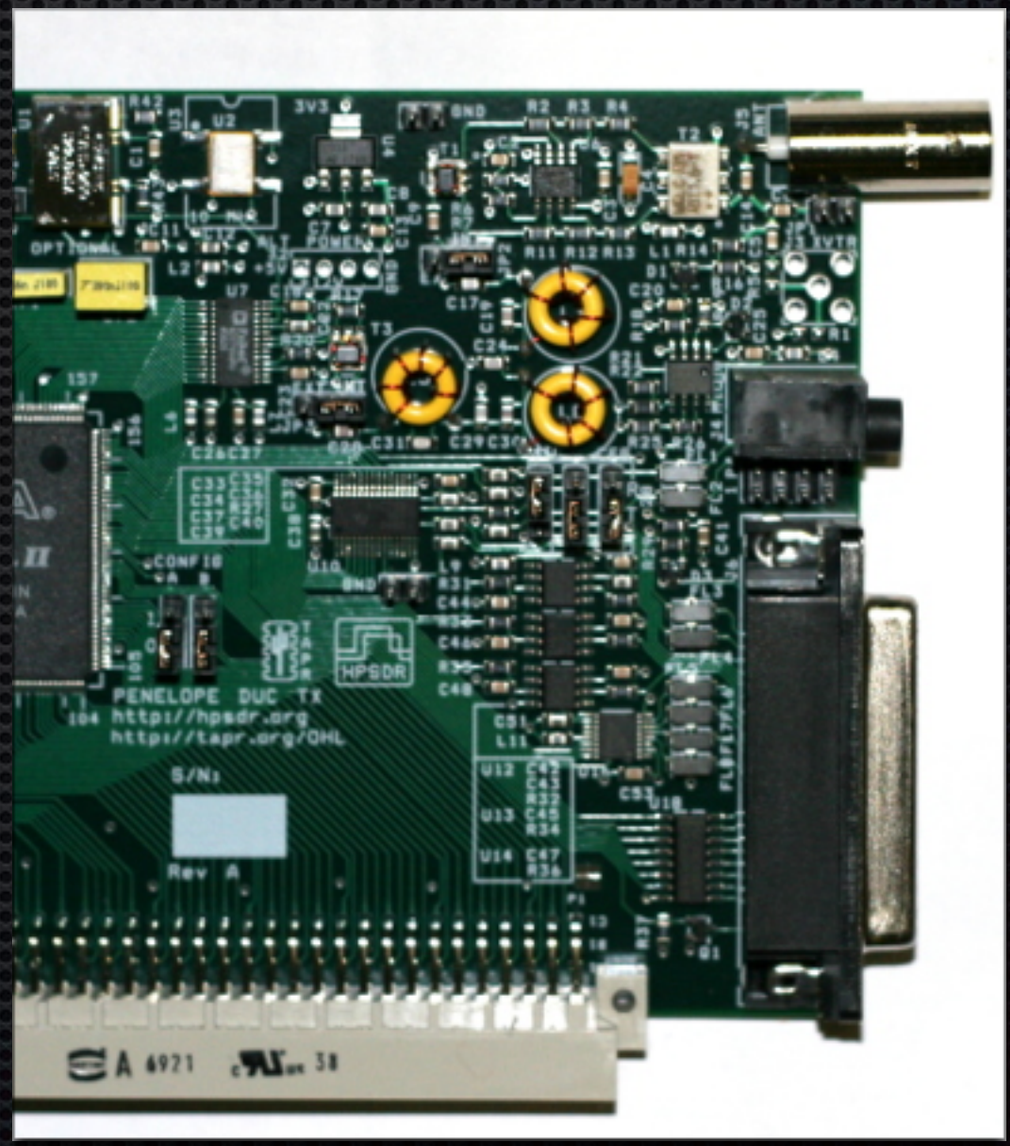
Mercury

- LTC2208 ADC as close to antenna as possible
  - 130Msps 16-bit
- Altera Cyclone III FPGA
- Current firmware allows 4 receivers at 192kHz sample rate



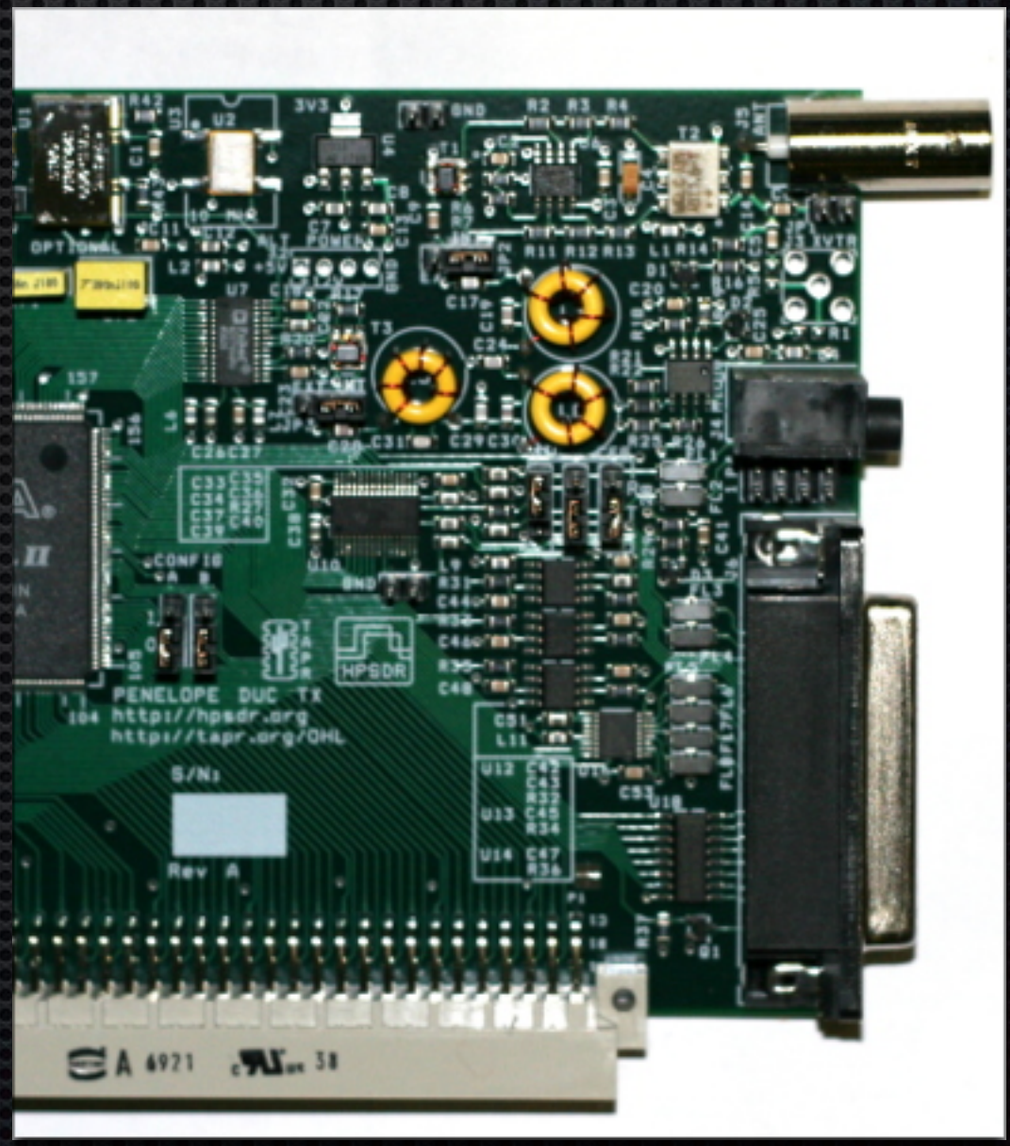
Mercury Block Diagram

Penelope


PENELOPE DUC TX
http://hpsdr.org
http://tapr.org/OHL
S/N:
Rev A
HPSDR

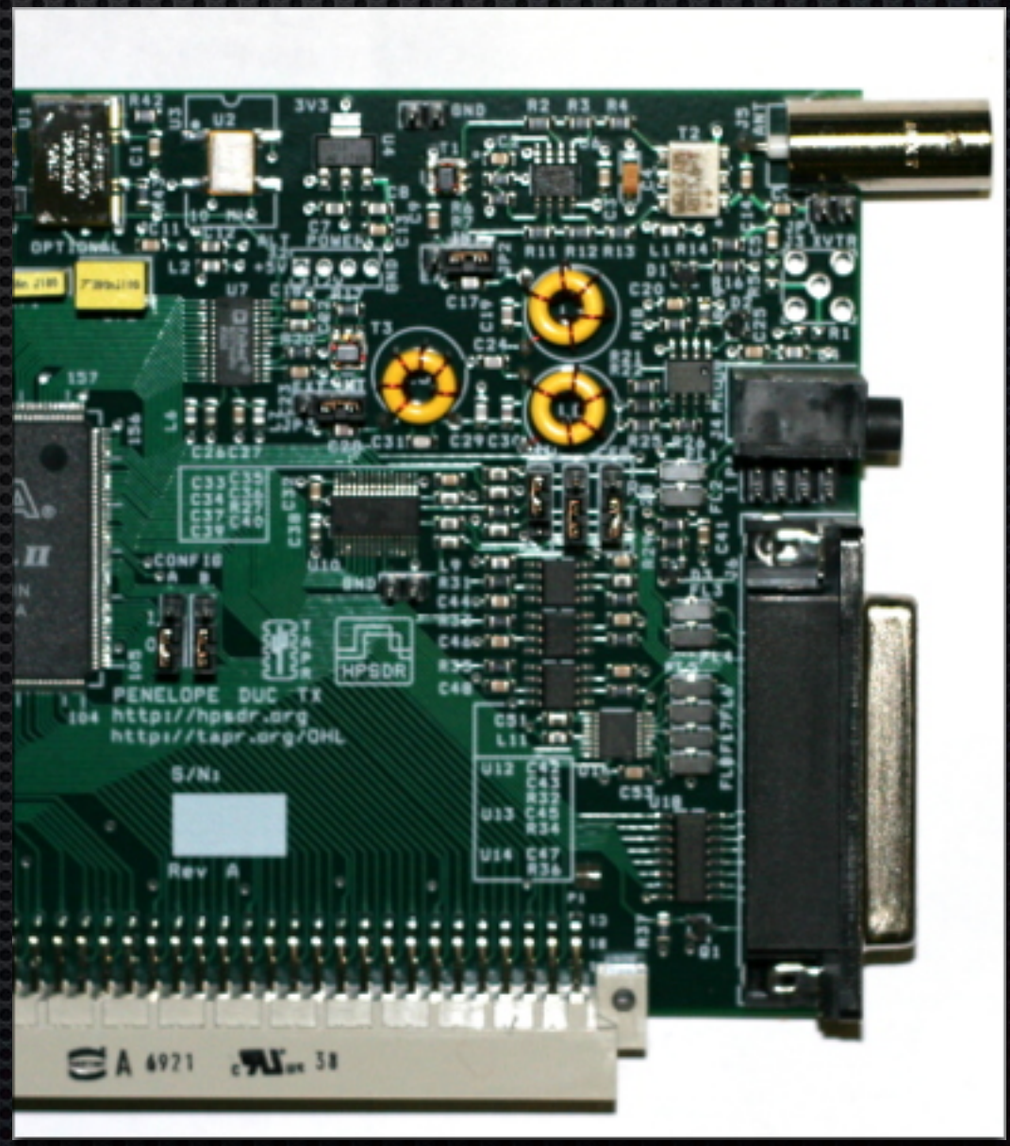
Penelope

- AD9744 DAC as close to antenna as possible
  - 210 Msps 14-bit



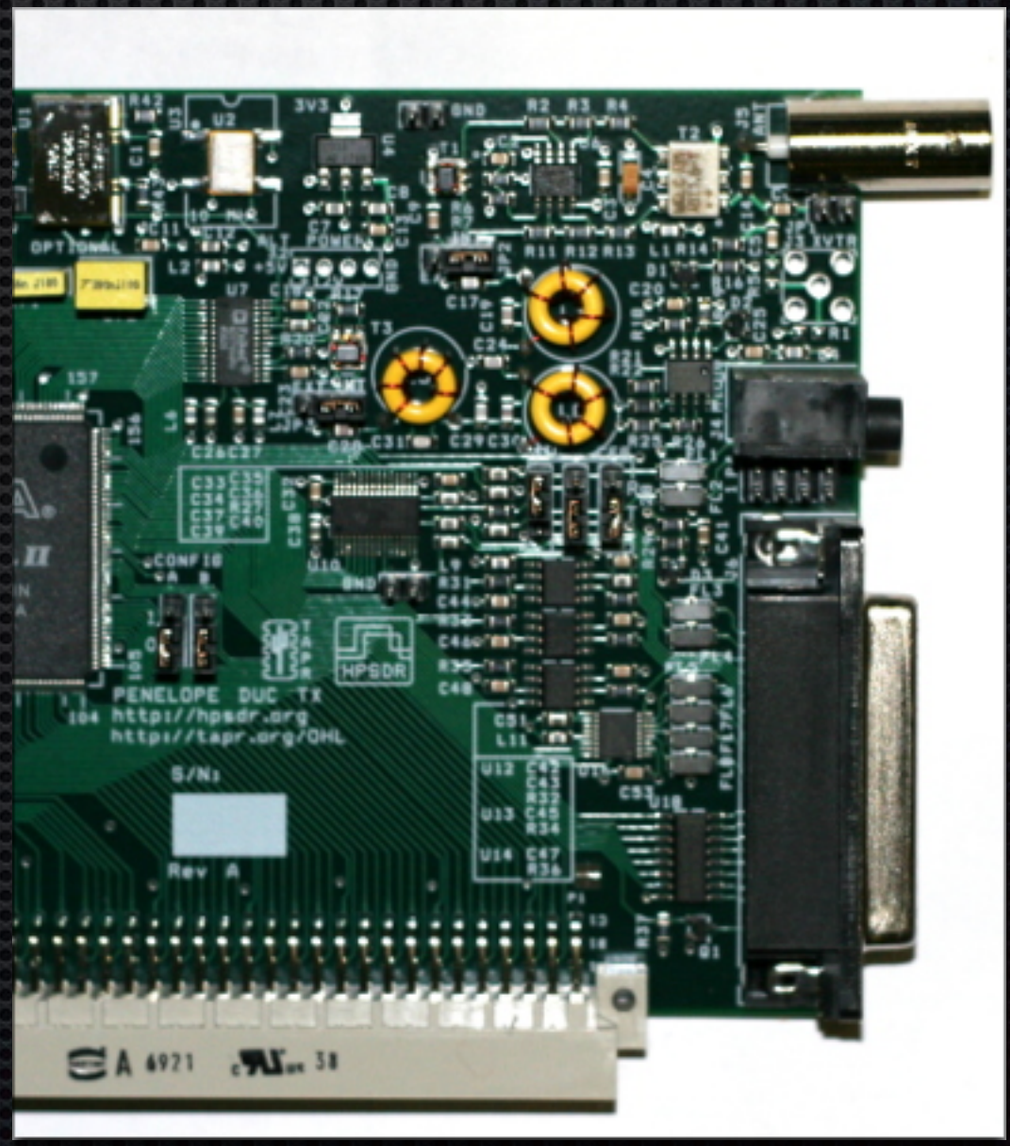
Penelope

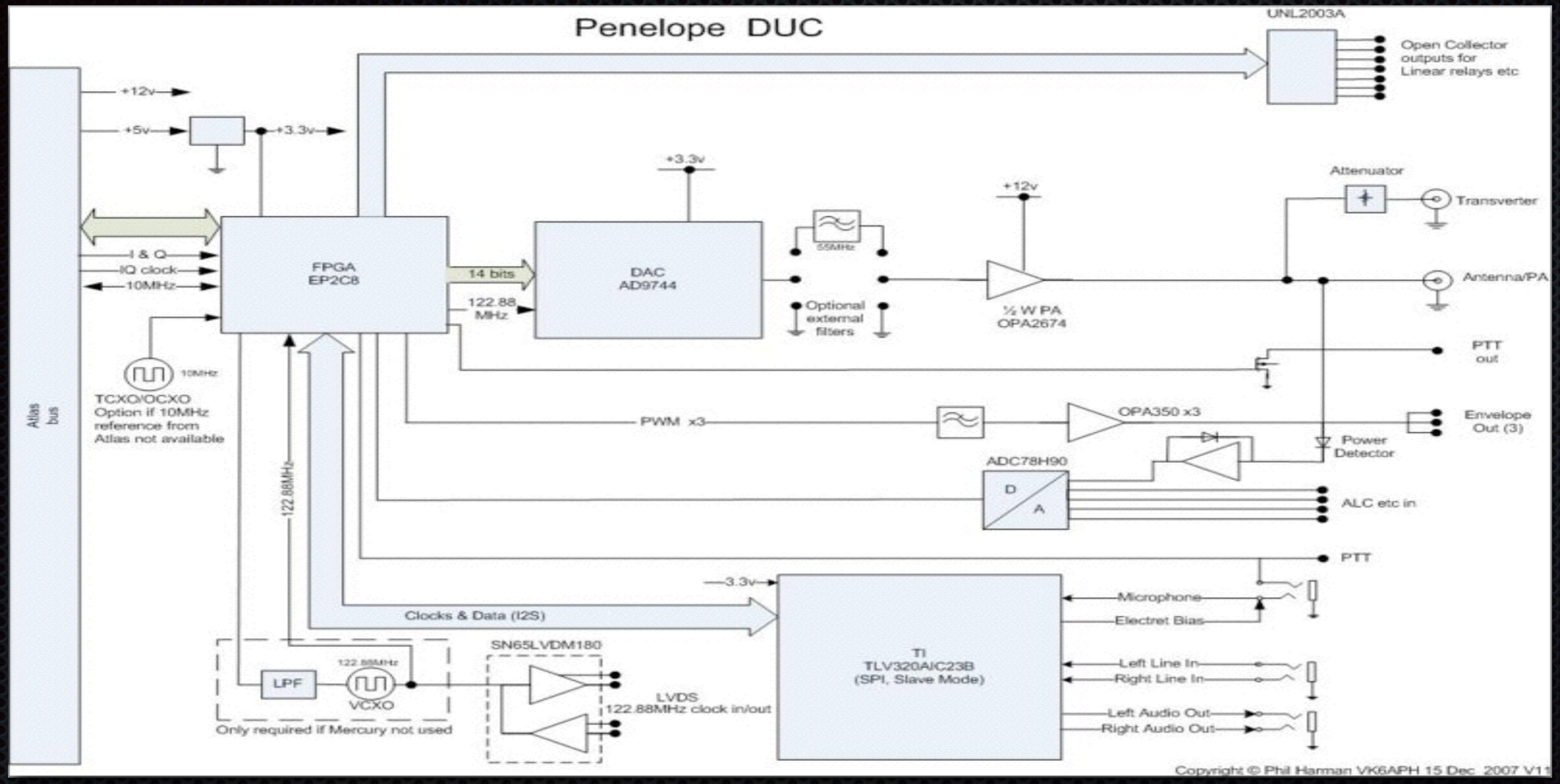
- AD9744 DAC as close to antenna as possible
  - 210 Msps 14-bit
- Altera Cyclone II FPGA



Penelope

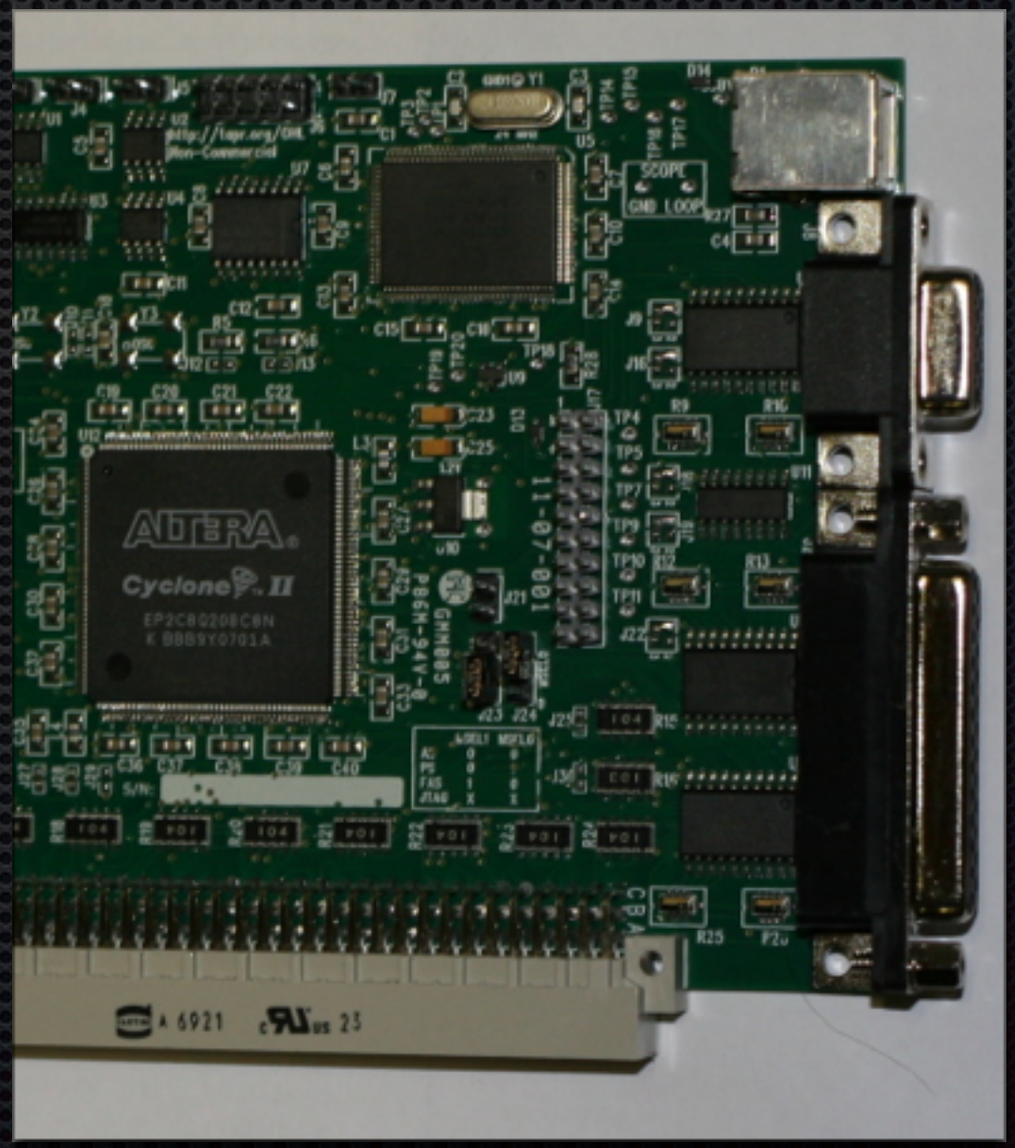
- AD9744 DAC as close to antenna as possible
  - 210 Msps 14-bit
- Altera Cyclone II FPGA
- Capable of transmitting multiple signals simultaneously





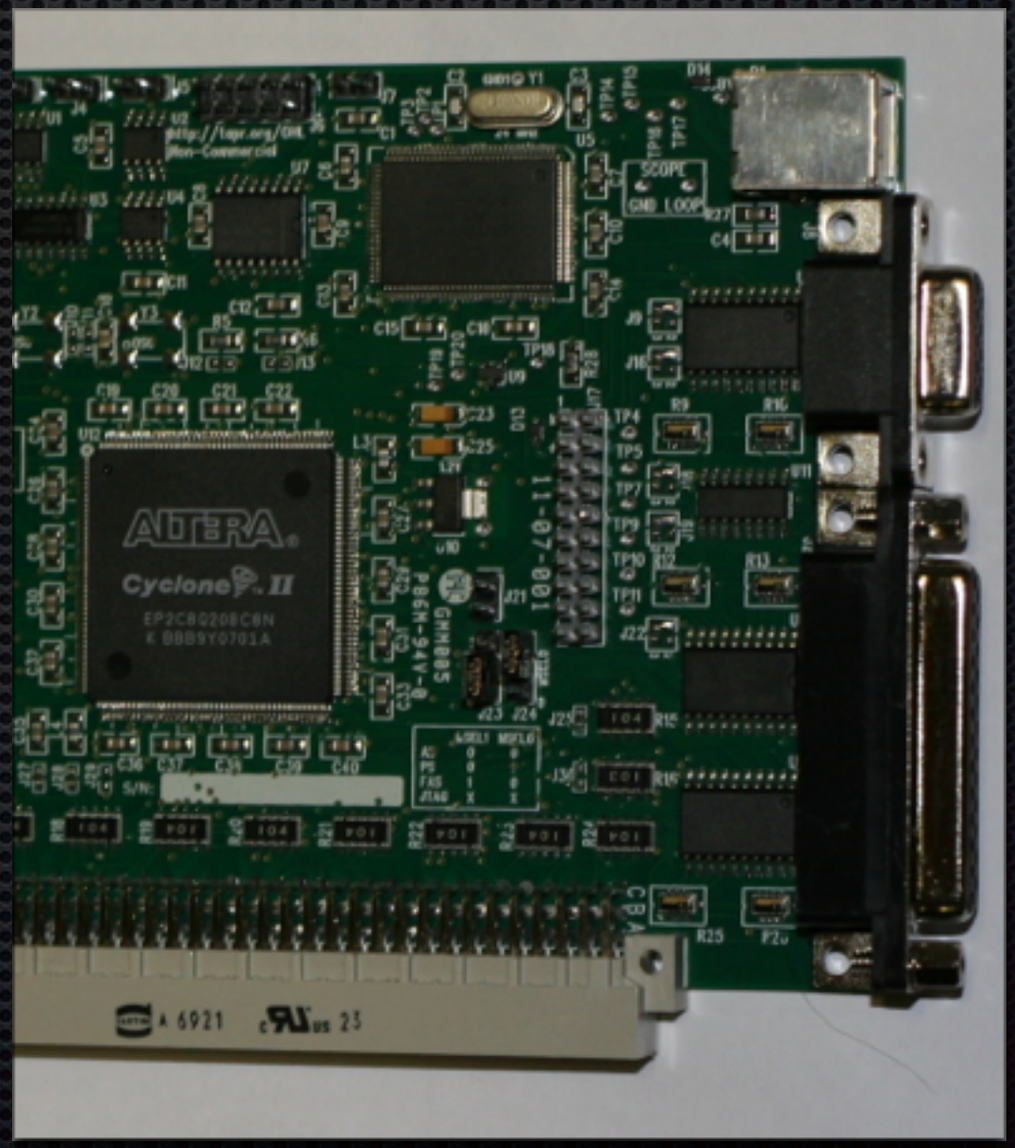
Penelope Block Diagram

Magister/Ozymandias



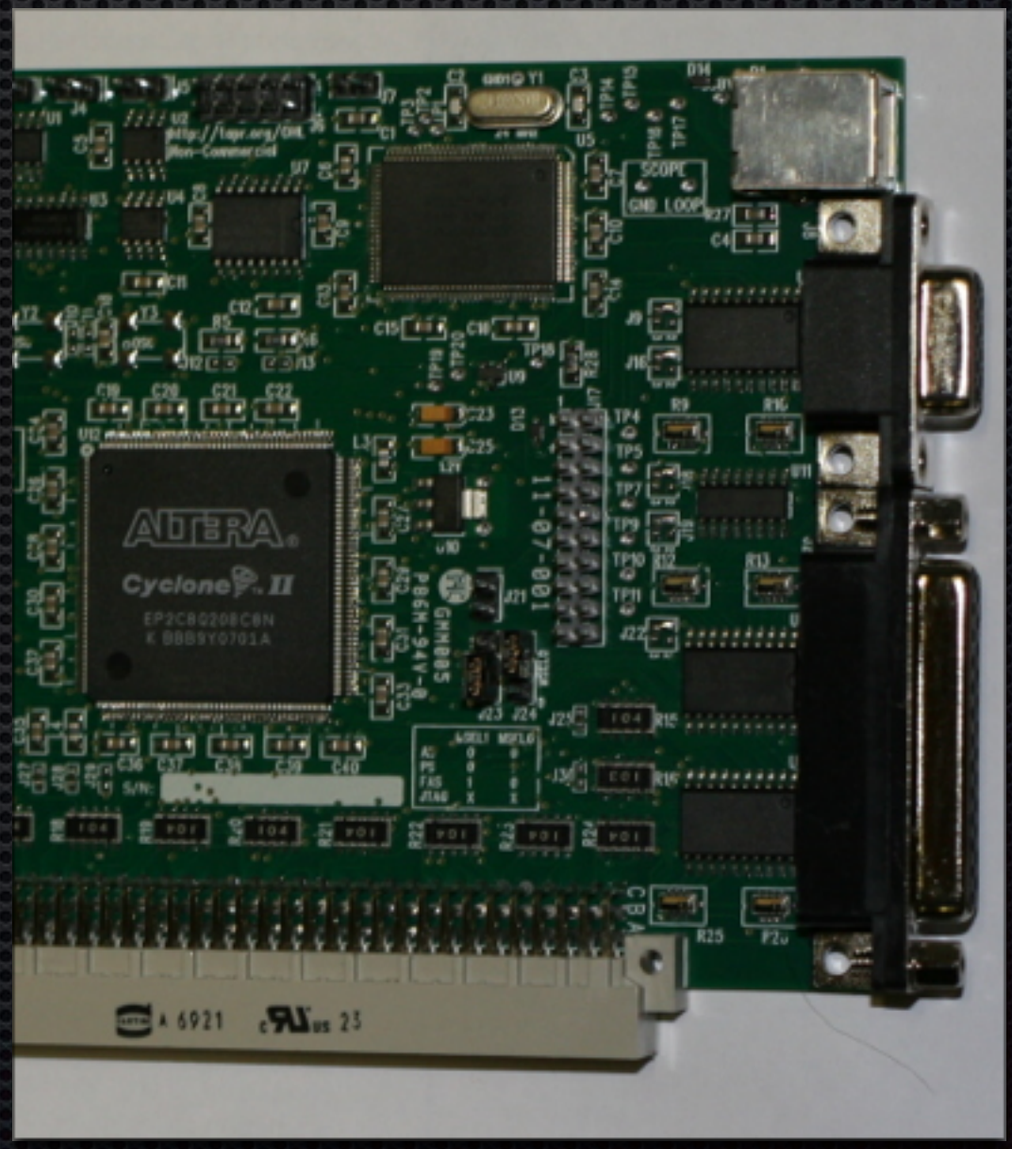
Magister/Ozymandias

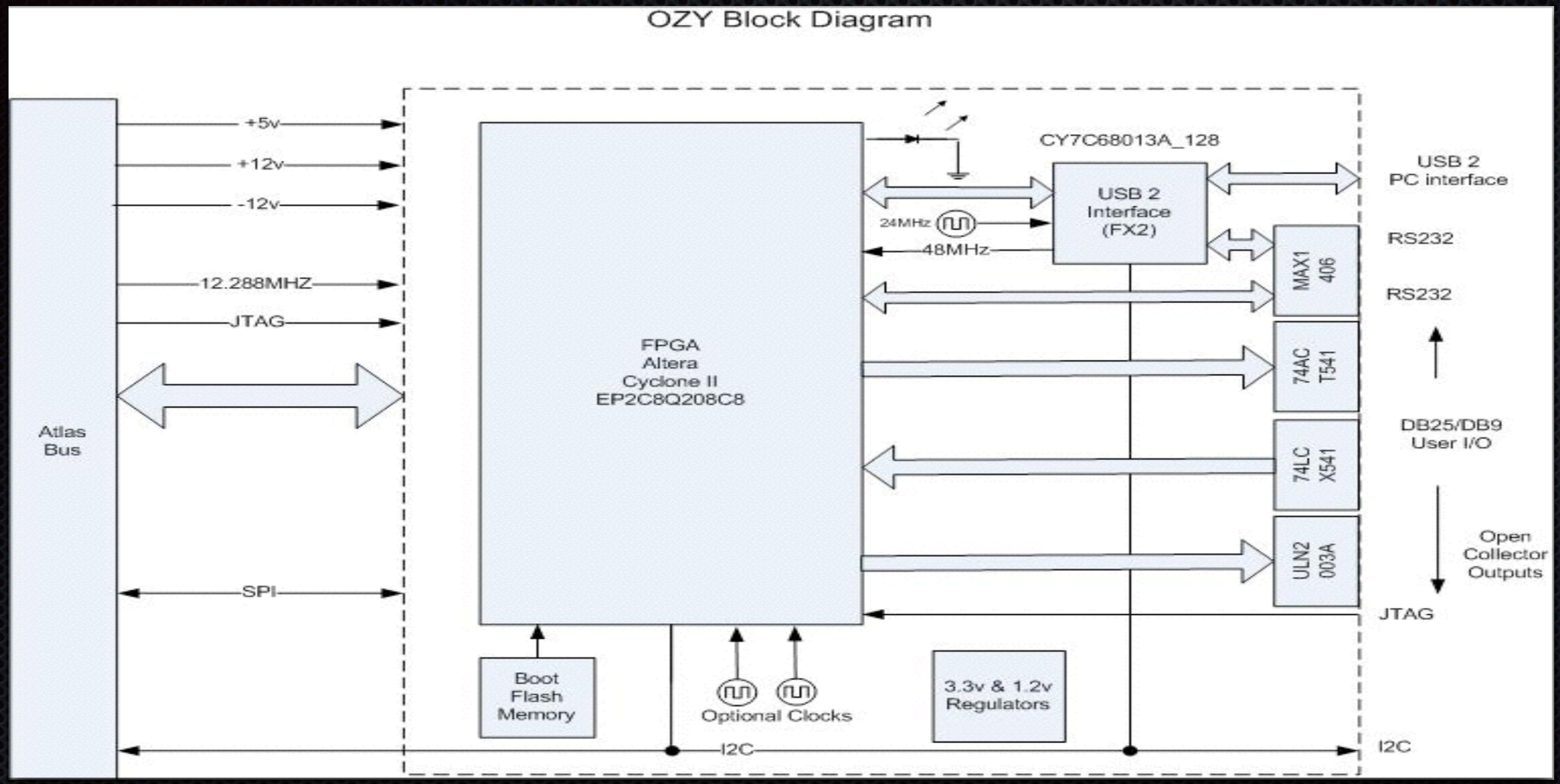
- USB 2.0 connection with Cypress FX2



Magister/Ozymandias

- USB 2.0 connection with Cypress FX2
- Altera Cyclone II FPGA





Ozymandias Block Diagram

Myths About the Mac

Myths About the Mac

- The kernel is just FreeBSD

Myths About the Mac

- The kernel is just FreeBSD
- Because it's UNIX®, all Linux software works

Myths About the Mac

- The kernel is just FreeBSD
- Because it's UNIX®, all Linux software works
- Everything's that proprietary Apple stuff

Goals of MacHPSDR

Goals of MacHPSDR

- Be a “First Class” piece of OpenHPSDR software

Goals of MacHPSDR

- Be a “First Class” piece of OpenHPSDR software
- Have a “Mac like” interface

Goals of MacHPSDR

- Be a “First Class” piece of OpenHPSDR software
- Have a “Mac like” interface
- Fully leverage available Mac technologies

Goals of MacHPSDR

- Be a “First Class” piece of OpenHPSDR software
- Have a “Mac like” interface
- Fully leverage available Mac technologies
- Have a simple installation process

Goals of MacHPSDR

- Be a “First Class” piece of OpenHPSDR software
- Have a “Mac like” interface
- Fully leverage available Mac technologies
- Have a simple installation process
- Make the source code fully available

Mac-GHPSDR—A First Cut

Mac-GHPSDR—A First Cut

- A port of John Melton's ghpsdr to the Mac

Mac-GHPSDR—A First Cut

- A port of John Melton's ghpsdr to the Mac
- Reworked the libusb code to use IOKit

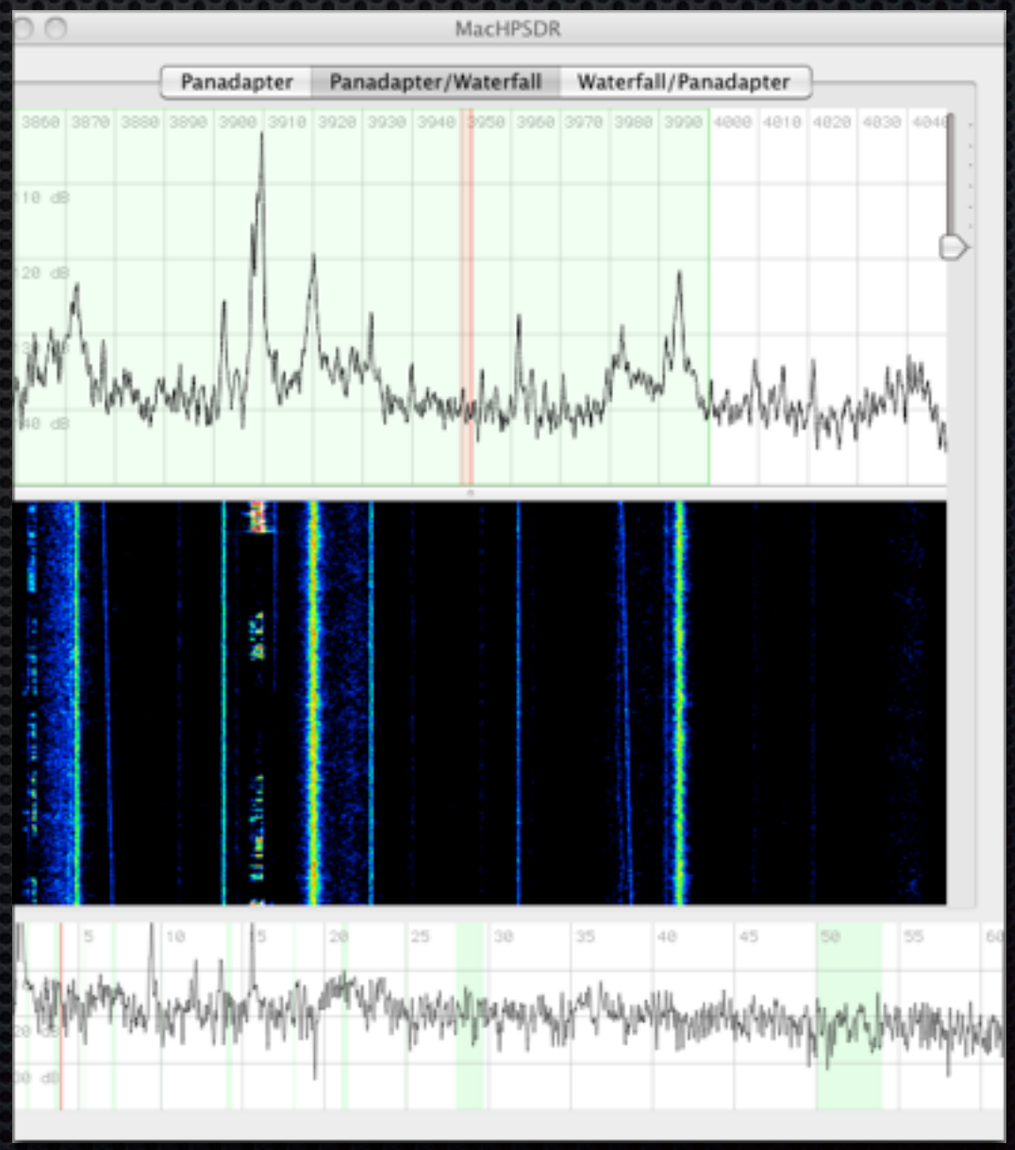
Mac-GHPSDR—A First Cut

- A port of John Melton's ghpsdr to the Mac
- Reworked the libusb code to use IOKit
- Fixed performance problems with threading

Mac-GHPSDR—A First Cut

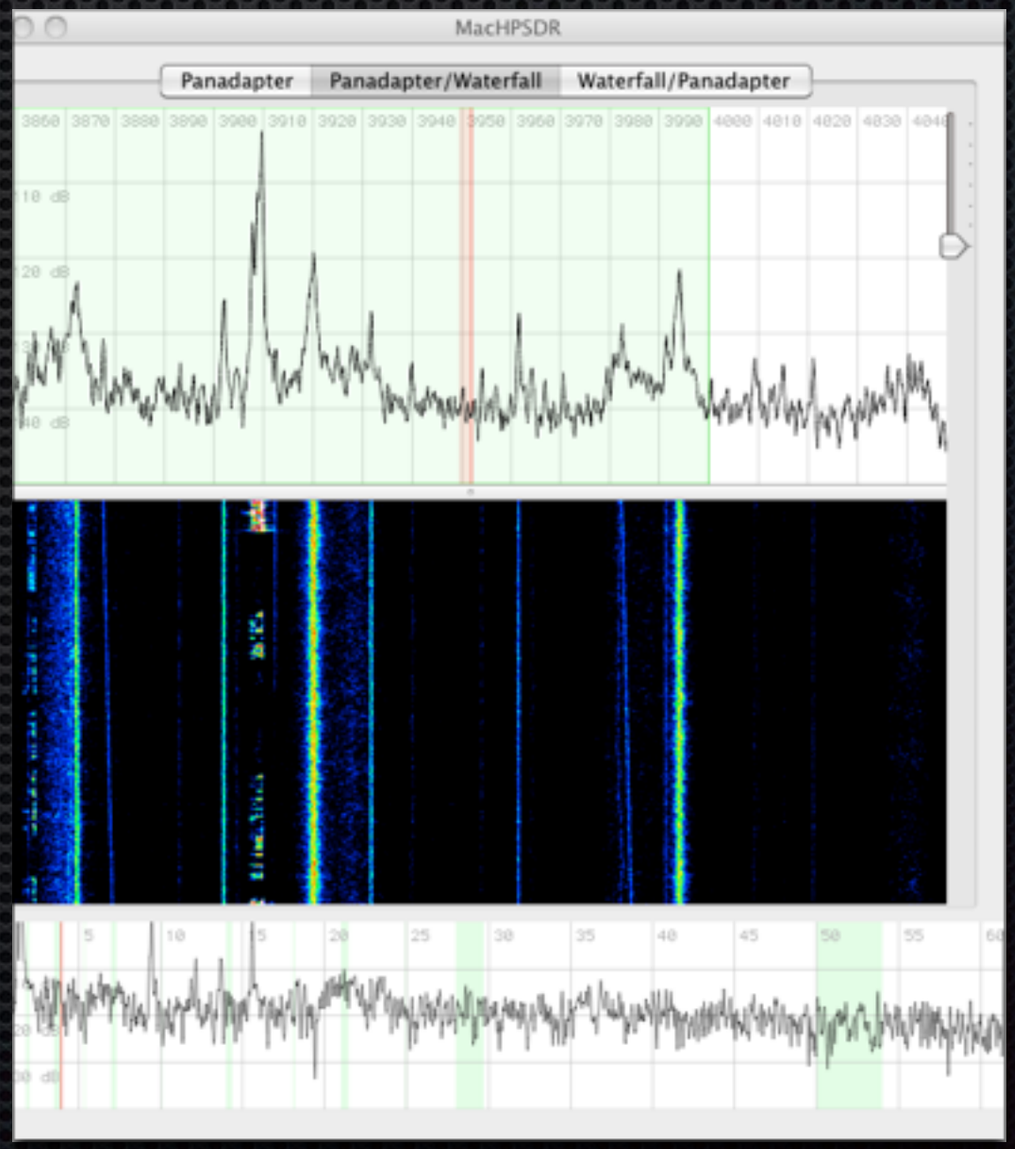
- A port of John Melton's ghpsdr to the Mac
- Reworked the libusb code to use IOKit
- Fixed performance problems with threading
- Available at https://www.xenotropic.com/ham-svn/mac-ghpsdr

MacHPSDR


MacHPSDR
Panadapter
Panadapter/Waterfall
Waterfall/Panadapter

MacHPSDR

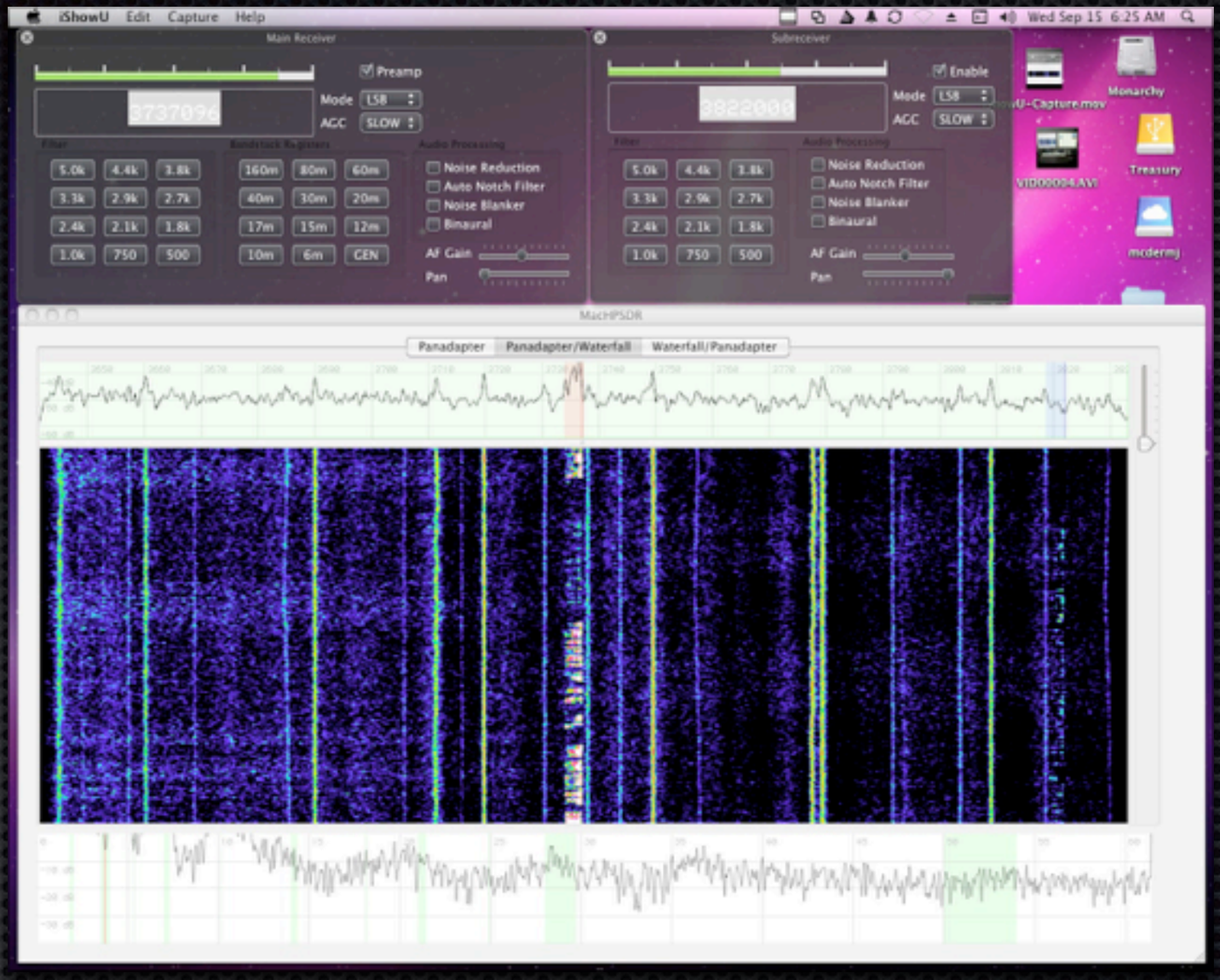
- Native Cocoa user interface
- Written in a mixture of Objective C and C
- Uses an improved version of the IOKit base from Mac-GHPSDR



MacHPSDR

Example Video

80m from Corvallis, OR 6:22 A.M. 9/15/2010



MacHPSDR

Example Video

80m from Corvallis, OR 6:22 A.M. 9/15/2010

Apple Technologies In Use

Apple Technologies In Use

- Objective C
- IOKit
- Cocoa
- Core Animation
- OpenGL
- Grand Central Dispatch
- vDSP
- Sparkle Updater

Objective C

Objective C

- Required to use Cocoa

Objective C

- Required to use Cocoa
- Is a dialect of C that grafts on Object Oriented concepts

Objective C

- Required to use Cocoa
- Is a dialect of C that grafts on Object Oriented concepts
- Has a different syntax than C++

Objective C

- Required to use Cocoa
- Is a dialect of C that grafts on Object Oriented concepts
- Has a different syntax than C++
- Isn’t bad once you get used to it

Objective C

- Required to use Cocoa
- Is a dialect of C that grafts on Object Oriented concepts
- Has a different syntax than C++
- Isn’t bad once you get used to it
- Readability is really nice

Objective C

- Required to use Cocoa
- Is a dialect of C that grafts on Object Oriented concepts
- Has a different syntax than C++
- Isn’t bad once you get used to it
- Readability is really nice
  - Ex: [transceiver changeFrequency:3500000 onReceiver:4]

IOKit

IOKit

- A generic framework for utilizing devices in the OS from userland

IOKit

- A generic framework for utilizing devices in the OS from userland
- Supports not only USB, but Firewire and other devices

IOKit

- A generic framework for utilizing devices in the OS from userland
- Supports not only USB, but Firewire and other devices
- No driver installation necessary

IOKit Performance

IOKit Performance

- Asynchronous calls are greatly preferred

IOKit Performance

- Asynchronous calls are greatly preferred
- Try to keep at least one transaction in the pipeline at a time

IOKit Performance

- Asynchronous calls are greatly preferred
- Try to keep at least one transaction in the pipeline at a time
- Use realtime threads for reading and writing SDR data

IOKit Performance

- Asynchronous calls are greatly preferred
- Try to keep at least one transaction in the pipeline at a time
- Use realtime threads for reading and writing SDR data
- POSIX semaphores don’t work quite right, use mach

IOKit Performance

- Asynchronous calls are greatly preferred
- Try to keep at least one transaction in the pipeline at a time
- Use realtime threads for reading and writing SDR data
- POSIX semaphores don’t work quite right, use mach
- Larger block sizes help reduce kernel ⟺ userland transitions

Cocoa

Cocoa

- Based on the NextStep tech (you can tell by the NS prefix to classes like NSView)

Cocoa

- Based on the NextStep tech (you can tell by the NS prefix to classes like NSView)
- Standard application toolkit

Cocoa

- Based on the NextStep tech (you can tell by the NS prefix to classes like NSView)
- Standard application toolkit
- C-based Carbon toolkit will not be ported to 64-bit

Cocoa

- Based on the NextStep tech (you can tell by the NS prefix to classes like NSView)
- Standard application toolkit
- C-based Carbon toolkit will not be ported to 64-bit
- Interface is built with a tool that creates a “nib” file that defines the objects on screen

Core Animation

Core Animation

- A technology that provides animation layers that can be composited

Core Animation

- A technology that provides animation layers that can be composited
- Includes many default animations that happen automatically

Core Animation

- A technology that provides animation layers that can be composited
- Includes many default animations that happen automatically
- Stock image filters can be applied to layers

Core Animation

- A technology that provides animation layers that can be composited
- Includes many default animations that happen automatically
- Stock image filters can be applied to layers
- Layers can be updated from any thread

Core Animation

- A technology that provides animation layers that can be composited
- Includes many default animations that happen automatically
- Stock image filters can be applied to layers
- Layers can be updated from any thread
- Layers can use a variety of drawing technologies

OpenGL

OpenGL

- 3D rendering standard

OpenGL

- 3D rendering standard
- A Core Animation layer can provide an OpenGL context

OpenGL

- 3D rendering standard
- A Core Animation layer can provide an OpenGL context
- MacHPSDR's waterfall uses an OpenGL texture that scrolls

OpenGL

- 3D rendering standard
- A Core Animation layer can provide an OpenGL context
- MacHPSDR's waterfall uses an OpenGL texture that scrolls
- MacHPSDR's panadapter draws the signal line with OpenGL into a Core Animation layer

Grand Central Dispatch

Grand Central Dispatch

- New technology for “Snow Leopard”

Grand Central Dispatch

- New technology for “Snow Leopard”
- Provides a way to manage threading and parallelism

Grand Central Dispatch

- New technology for “Snow Leopard”
- Provides a way to manage threading and parallelism
- The programmer creates queues, and the OS figures out the optimal number of threads as the program runs

Grand Central Dispatch

- New technology for “Snow Leopard”
- Provides a way to manage threading and parallelism
- The programmer creates queues, and the OS figures out the optimal number of threads as the program runs
- Optimizes based on the hardware available

Grand Central Dispatch

- New technology for “Snow Leopard”
- Provides a way to manage threading and parallelism
- The programmer creates queues, and the OS figures out the optimal number of threads as the program runs
- Optimizes based on the hardware available
- Not used extensively in MacHPSDR because you can’t make “real time” GCD queues

vDSP

vDSP

- DSP processing functions that are a part of the Accelerate framework

vDSP

- DSP processing functions that are a part of the Accelerate framework
- Shipped on every Mac

vDSP

- DSP processing functions that are a part of the Accelerate framework
- Shipped on every Mac
- Provides a set of functions that operate on arrays of floating point numbers optimized by using the processor's vector instructions (SSE/Altivec)

vDSP

- DSP processing functions that are a part of the Accelerate framework
- Shipped on every Mac
- Provides a set of functions that operate on arrays of floating point numbers optimized by using the processor's vector instructions (SSE/Altivec)
- Functions include FFT, convolutions, complex number manipulation, decibel calculations

The “DSP Rework” Branch

The “DSP Rework” Branch

- Current trunk uses hacked up DttSP from John Melton’s ghpsdr

The “DSP Rework” Branch

- Current trunk uses hacked up DttSP from John Melton’s ghpsdr
- DSP Rework is a redesign of the DSP functions that DttSP performs

The “DSP Rework” Branch

- Current trunk uses hacked up DttSP from John Melton’s ghpsdr
- DSP Rework is a redesign of the DSP functions that DttSP performs
- Rewritten using OO concepts with Objective C

The “DSP Rework” Branch

- Current trunk uses hacked up DttSP from John Melton’s ghpsdr
- DSP Rework is a redesign of the DSP functions that DttSP performs
- Rewritten using OO concepts with Objective C
- Extensively leverages the vDSP functions

Sparkle Updater

Sparkle Updater

- 3rd party freely available package

Sparkle Updater

- 3rd party freely available package
- Allows for automatic updating of software

Sparkle Updater

- 3rd party freely available package
- Allows for automatic updating of software
- Uses an RSS feed containing data about your software revisions

Sparkle Updater

- 3rd party freely available package
- Allows for automatic updating of software
- Uses an RSS feed containing data about your software revisions
- Also can be configured to canvas anonymous data on user's computer

Sparkle Updater

- 3rd party freely available package
- Allows for automatic updating of software
- Uses an RSS feed containing data about your software revisions
- Also can be configured to canvas anonymous data on user's computer
- Available at: sparkle.andymatuschak.org

Future Technologies

Future Technologies

- OpenCL
- Core Audio
- Distributed Objects

OpenCL

OpenCL

- “Open Computing Language”

OpenCL

- “Open Computing Language”
- Designed to ease construction of massively data parallel tasks

OpenCL

- “Open Computing Language”
- Designed to ease construction of massively data parallel tasks
- Computing units include the CPU, GPU, and even some custom designed IBM Cell boards

OpenCL

- “Open Computing Language”
- Designed to ease construction of massively data parallel tasks
- Computing units include the CPU, GPU, and even some custom designed IBM Cell boards
- Included in Snow Leopard

OpenCL

- “Open Computing Language”
- Designed to ease construction of massively data parallel tasks
- Computing units include the CPU, GPU, and even some custom designed IBM Cell boards
- Included in Snow Leopard
- GPGPU is difficult for SDR because of small block sizes

OpenCL + OpenGL

OpenCL + OpenGL

- Both standards maintained by the Khronos Group

OpenCL + OpenGL

- Both standards maintained by the Khronos Group
- OpenGL can use the same buffers as OpenCL

Core Audio

Core Audio

- Standard audio framework on the Mac

Core Audio

- Standard audio framework on the Mac
- Allows the use of “Audio Units”

Core Audio

- Standard audio framework on the Mac
- Allows the use of “Audio Units”
- Audio Units are the same plugins that Logic Pro, Garageband, and Logic Express use

Core Audio

- Standard audio framework on the Mac
- Allows the use of “Audio Units”
- Audio Units are the same plugins that Logic Pro, Garageband, and Logic Express use
- Audio Units can be embedded in any program

Core Audio

- Standard audio framework on the Mac
- Allows the use of “Audio Units”
- Audio Units are the same plugins that Logic Pro, Garageband, and Logic Express use
- Audio Units can be embedded in any program
- Stock audio units include a 31 band equalizer, a compressor/limiter

Distributed Objects

Distributed Objects

- Allows Cocoa objects in different programs to communicate
- Similar to remote procedure calls, but on the same machine
- Hope to use this to integrate with logging software and other external programs

Future Directions

Future Directions

- Integration with PSK31 and other digital mode software

Future Directions

- Integration with PSK31 and other digital mode software
- Finish the “DSP Rework” branch

Future Directions

- Integration with PSK31 and other digital mode software
- Finish the “DSP Rework” branch
- Integration with logging software

Future Directions

- Integration with PSK31 and other digital mode software
- Finish the “DSP Rework” branch
- Integration with logging software
- Put Audio Unit support in transmit chain

Future Directions

- Integration with PSK31 and other digital mode software
- Finish the “DSP Rework” branch
- Integration with logging software
- Put Audio Unit support in transmit chain
- Ichabod — the headless MacHPSDR

How Do I Get MacHPSDR

How Do I Get MacHPSDR

- Binary Distribution at www.nh6z.net/MacHPSDR/MacHPSDR.zip
- Source in Subversion: https://www.xenotropic.com/mac-svn/MacHPSDR

Documentation

Documentation

- There is none!
- Could use an intrepid soul to help with the help files

How to Contribute

How to Contribute

- Patches are accepted by e-mail and reviewed
- As time goes on, and if there is a critical mass of developers, SVN write access will be granted
- Bugs are tracked on Lighthouse at mcdermj.lighthouseapp.com
- The OpenHPSDR wiki is always a good place to contribute