GReasy: HARDWARE ACCELERATED GNU Radio MADE EASY

GNU RADIO CONFERENCE, 25 SEPTEMBER 2012

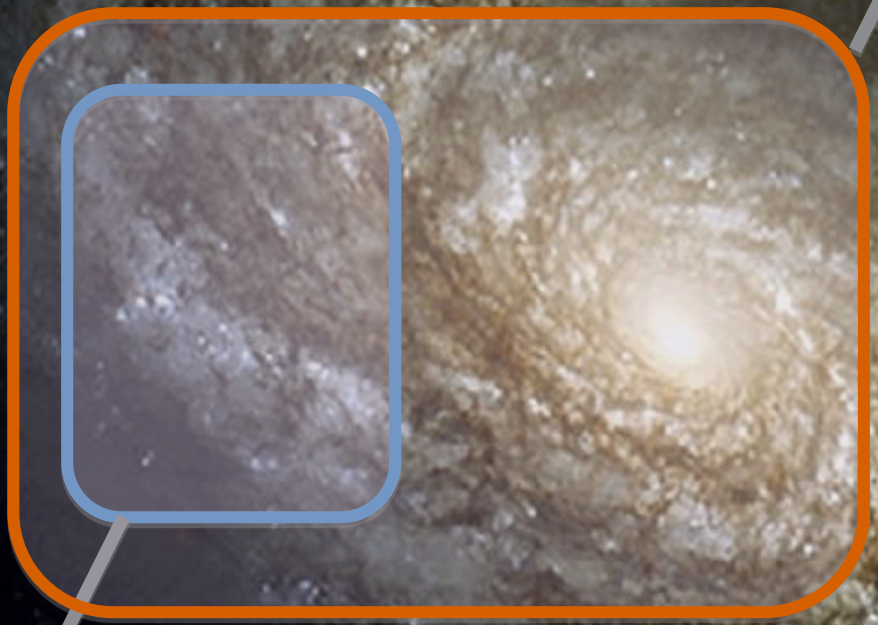
Ryan Marlow
Shaver Deyerle

Advisor: Peter Athanas

AGENDA

Rationale
Software Implementation
Hardware Platform
Demo Video

The Galaxy of Radio Prototypes



Extending GNURadio w/ HW Acceleration

Radios that can be quickly prototyped with the baseline GNURadio

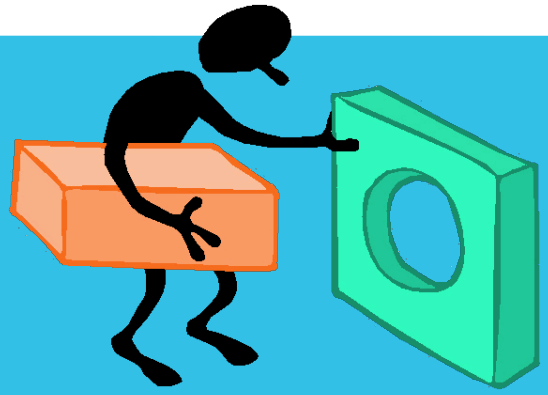
Extending GR w/ FPGAs?

GNU Radio

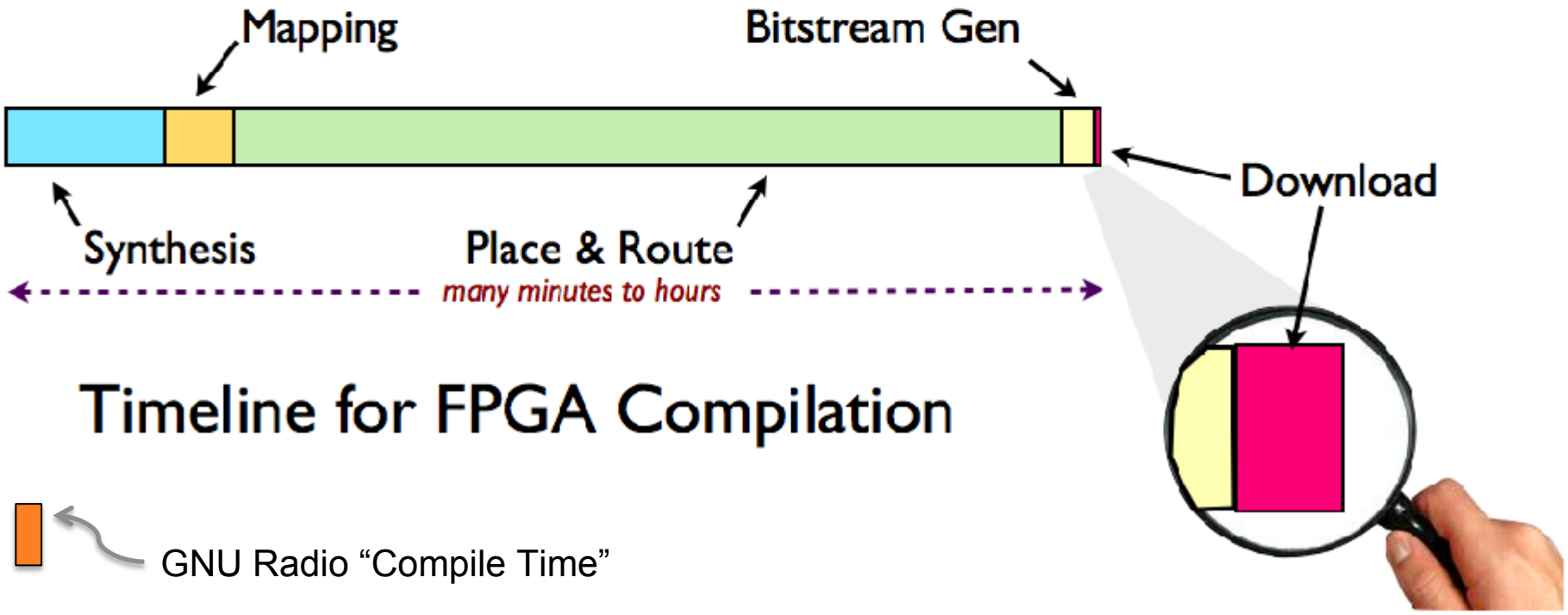
- Instant Gratification
- Intuitive to Radio Designers
- Allows Design-Space Exploration w/ Real Data

XILINX®

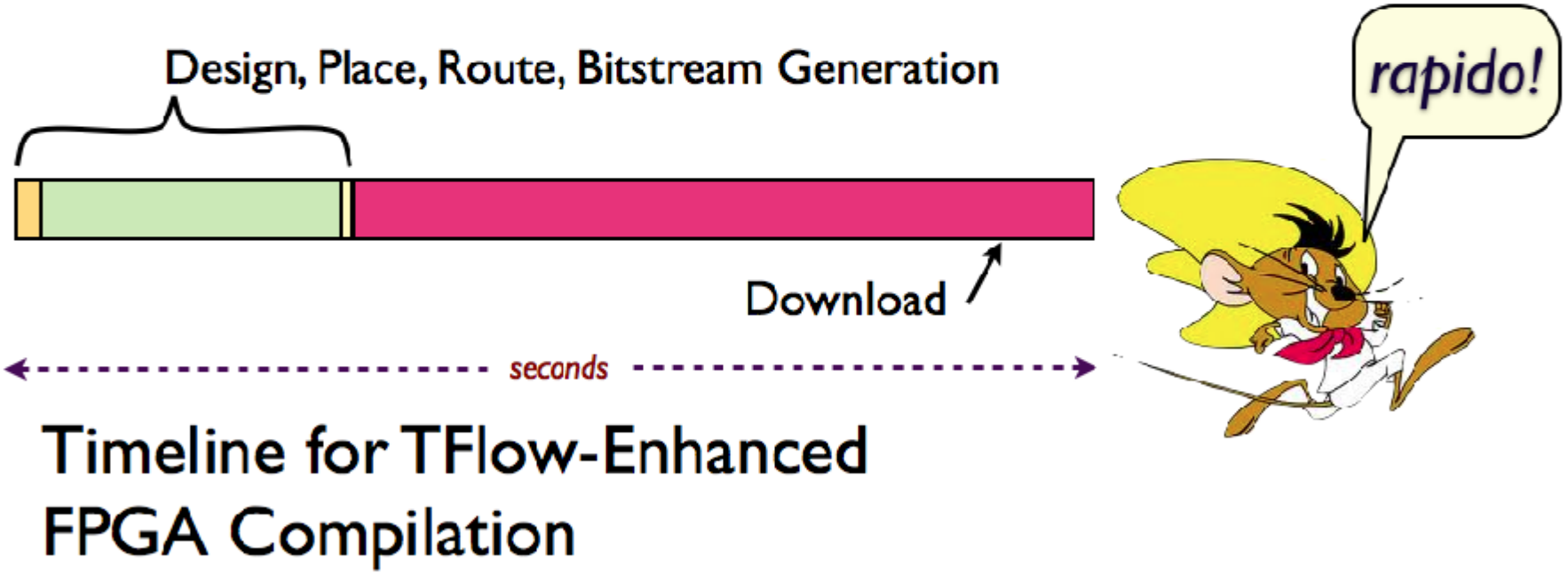
- Long Compile Times
- No standardized HDL framework
- Lengthy Simulations with Limited Data



FPGAs-in-the-Loop Ruin Productivity



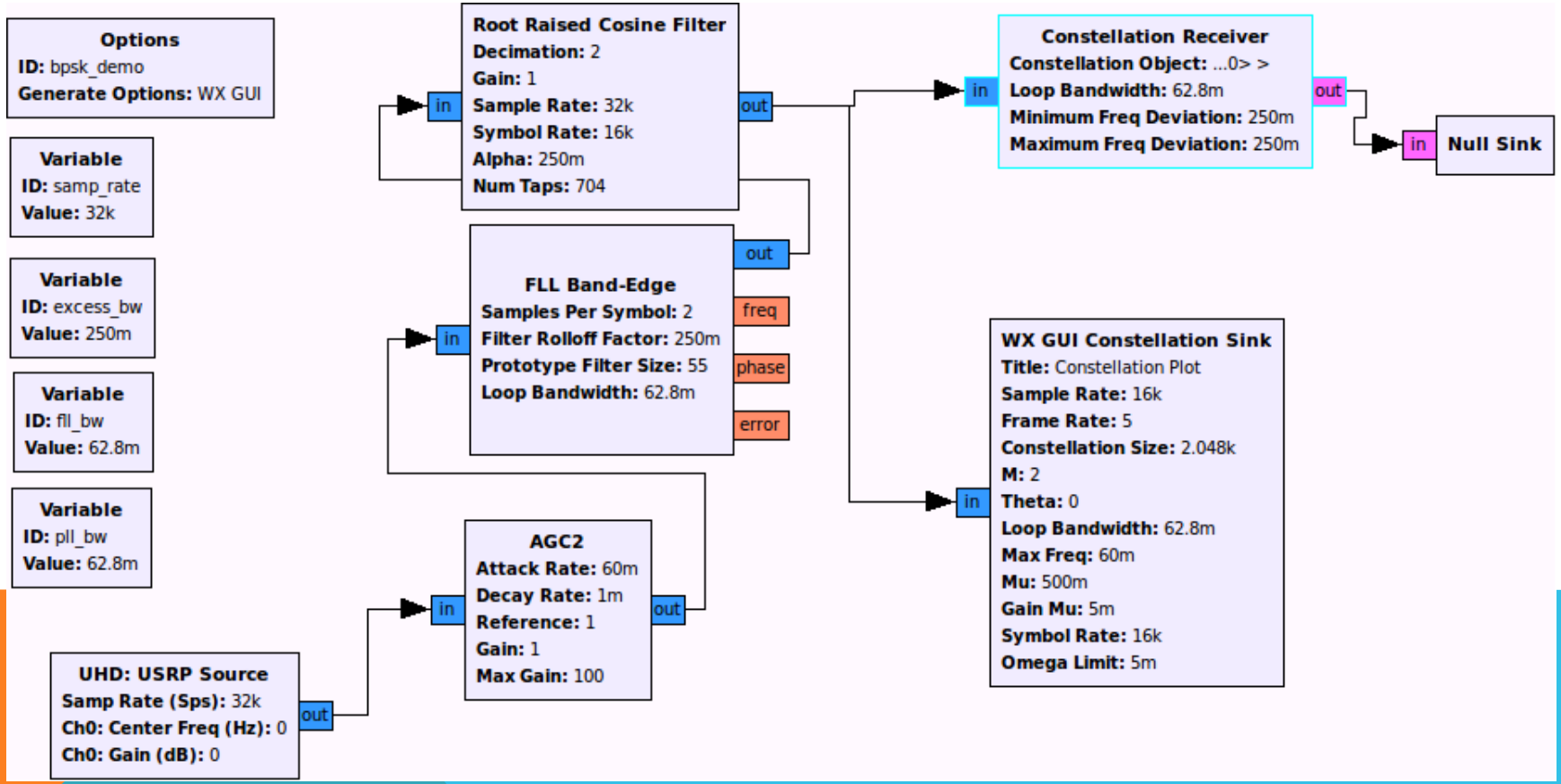
Preserving Instant Gratification



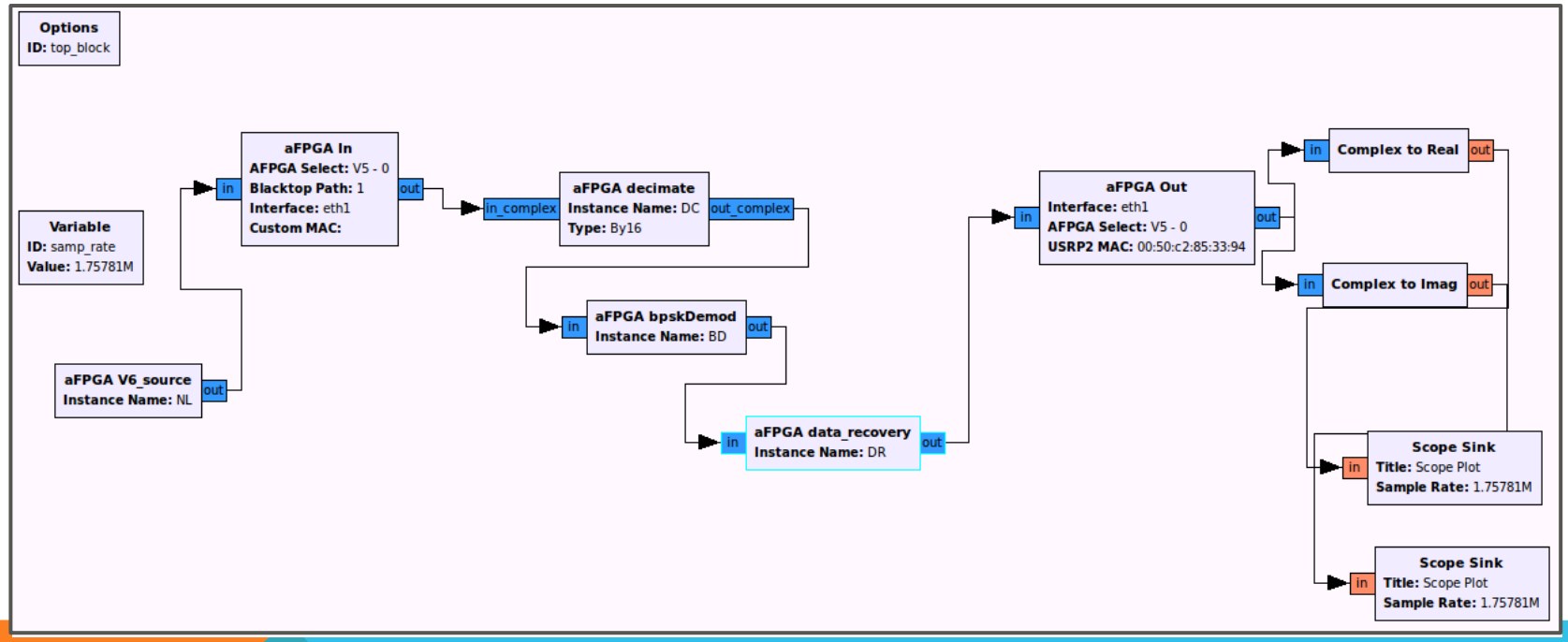
Timeline for TFlow-Enhanced FPGA Compilation

ENHANCED GNURADIO DESIGN FLOW

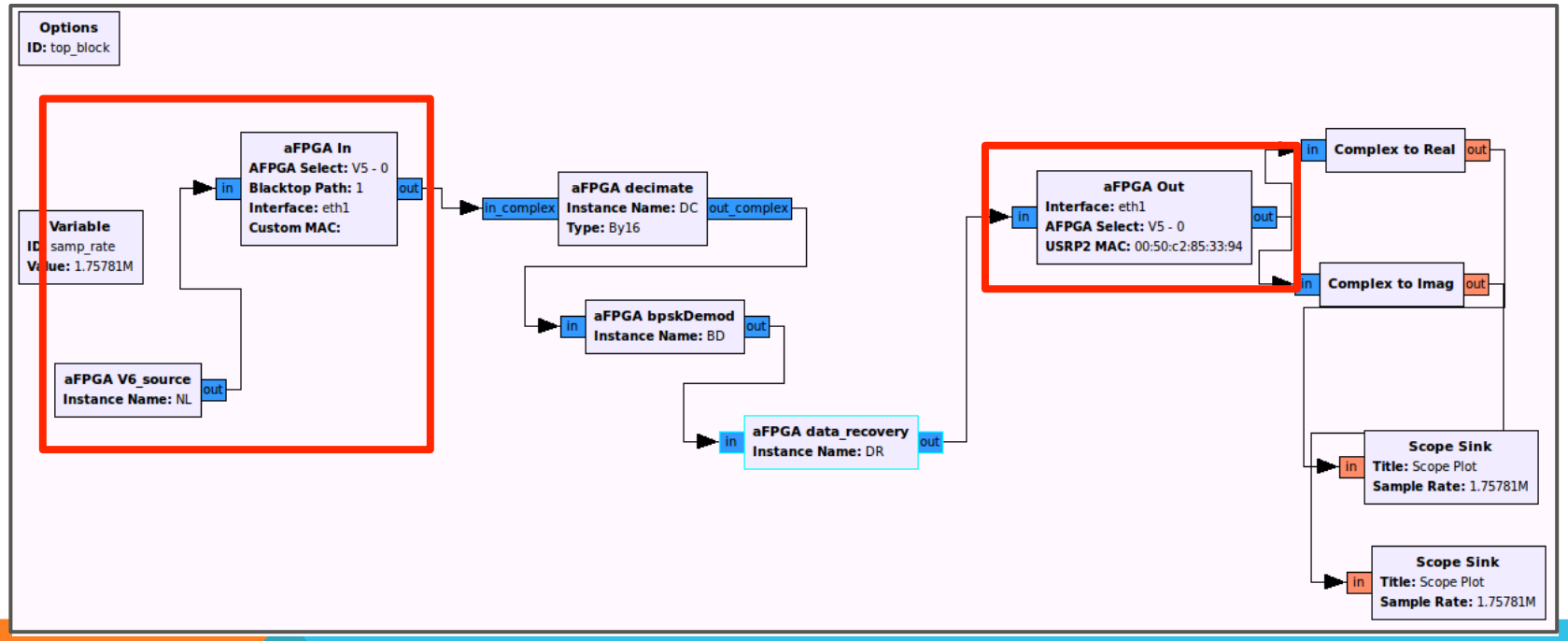
EXAMPLE SOFTWARE DESIGN IN GNU RADIO



EXAMPLE HARDWARE DESIGN IN GNU RADIO

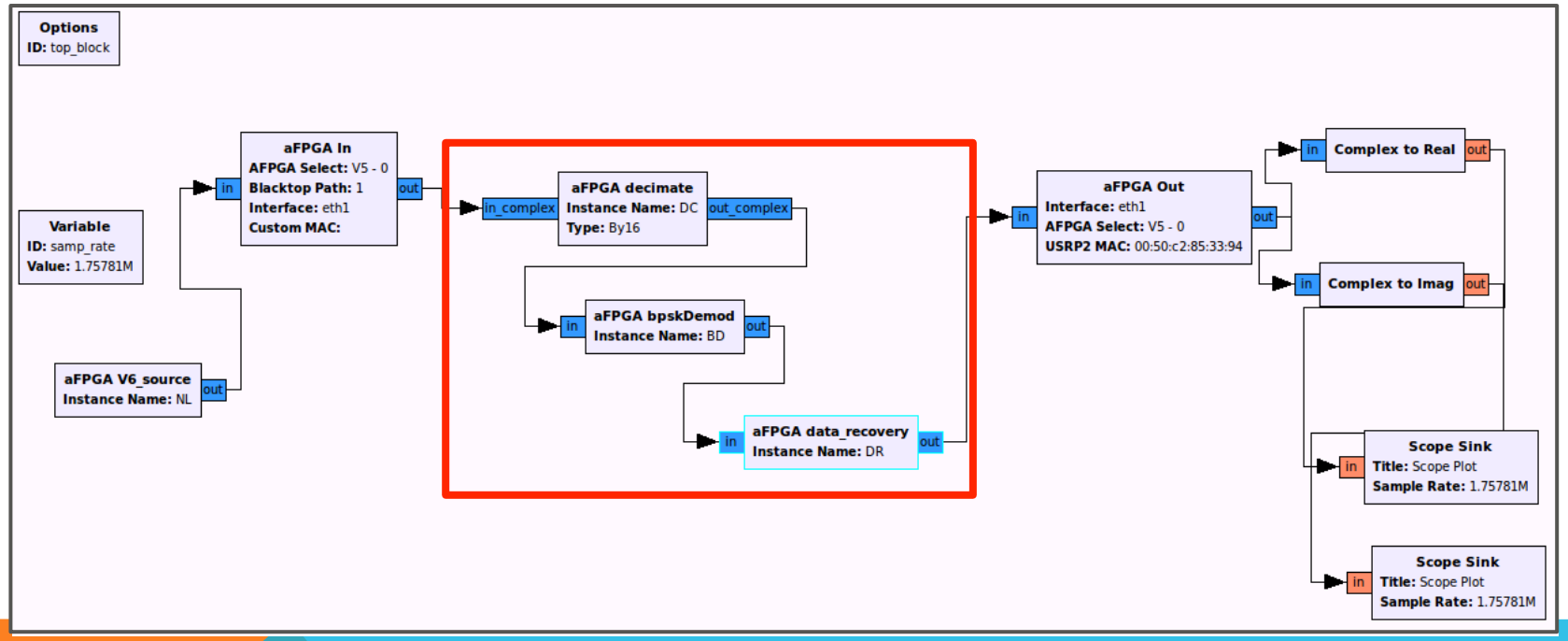


EXAMPLE HARDWARE DESIGN IN GNU RADIO



Define FPGA input and FPGA Output

EXAMPLE HARDWARE DESIGN IN GNU RADIO

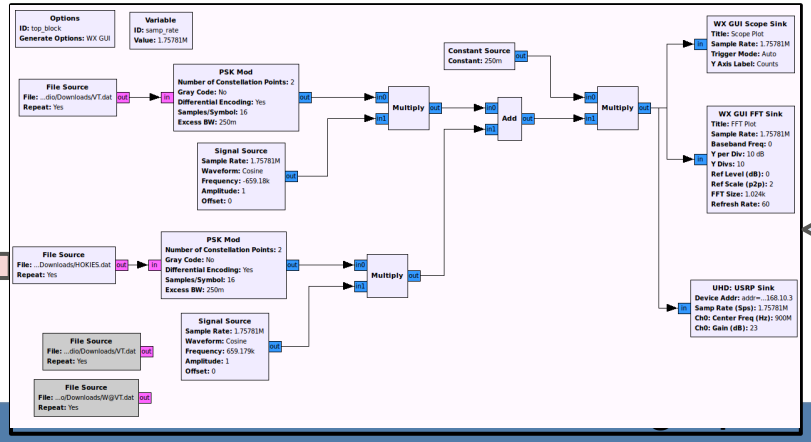


Drag and Drop Pre-Compiled FPGA Blocks

DESIGN FLOW WITH GREASY

Library Developer Flow

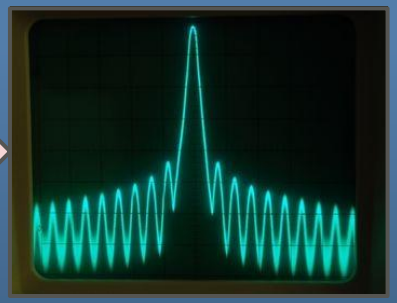
Module HDL and Cores → Individual Bitstream Generation → Module Registration → GNU Radio Block Library



Click "Go"

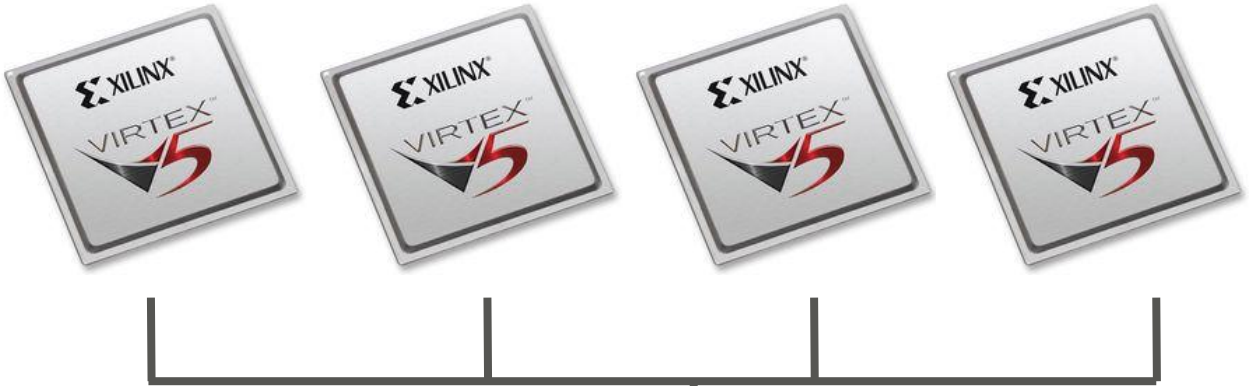
End User Flow

EDIF Creation → Combine Module Bitstreams → JTAG Programming



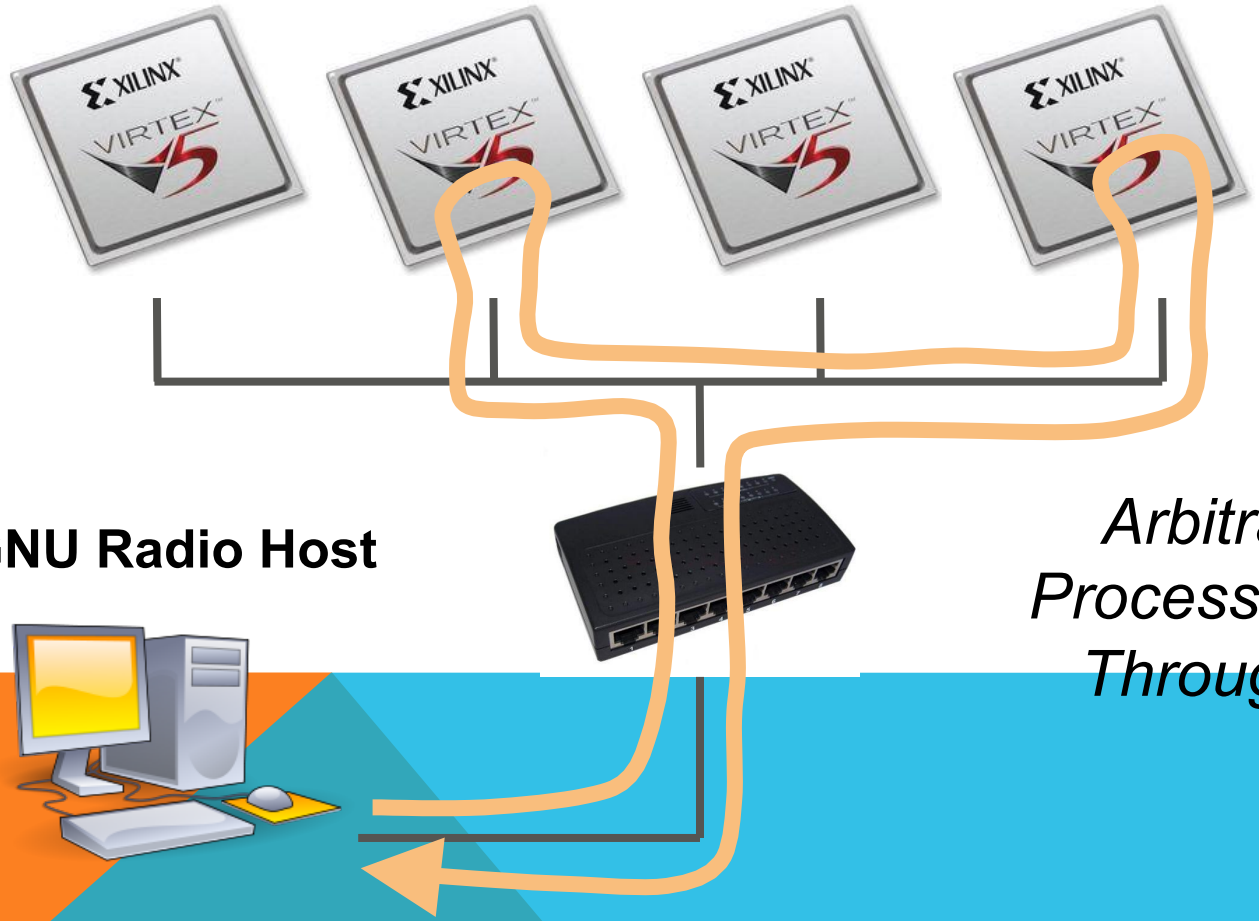
EXTENDED GNURADIO PLATFORM

COMPUTING MODEL



GNU Radio Host

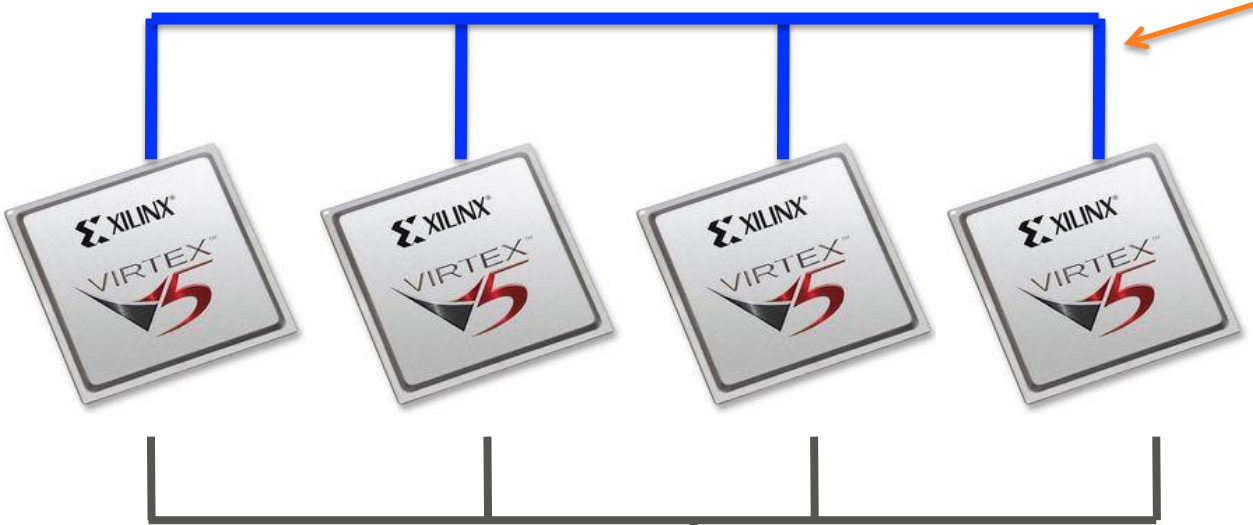
COMPUTING MODEL



GNU Radio Host

Arbitrarily Chain Processing Elements Through Ethernet

COMPUTING MODEL



High-Speed Auxiliary Networks

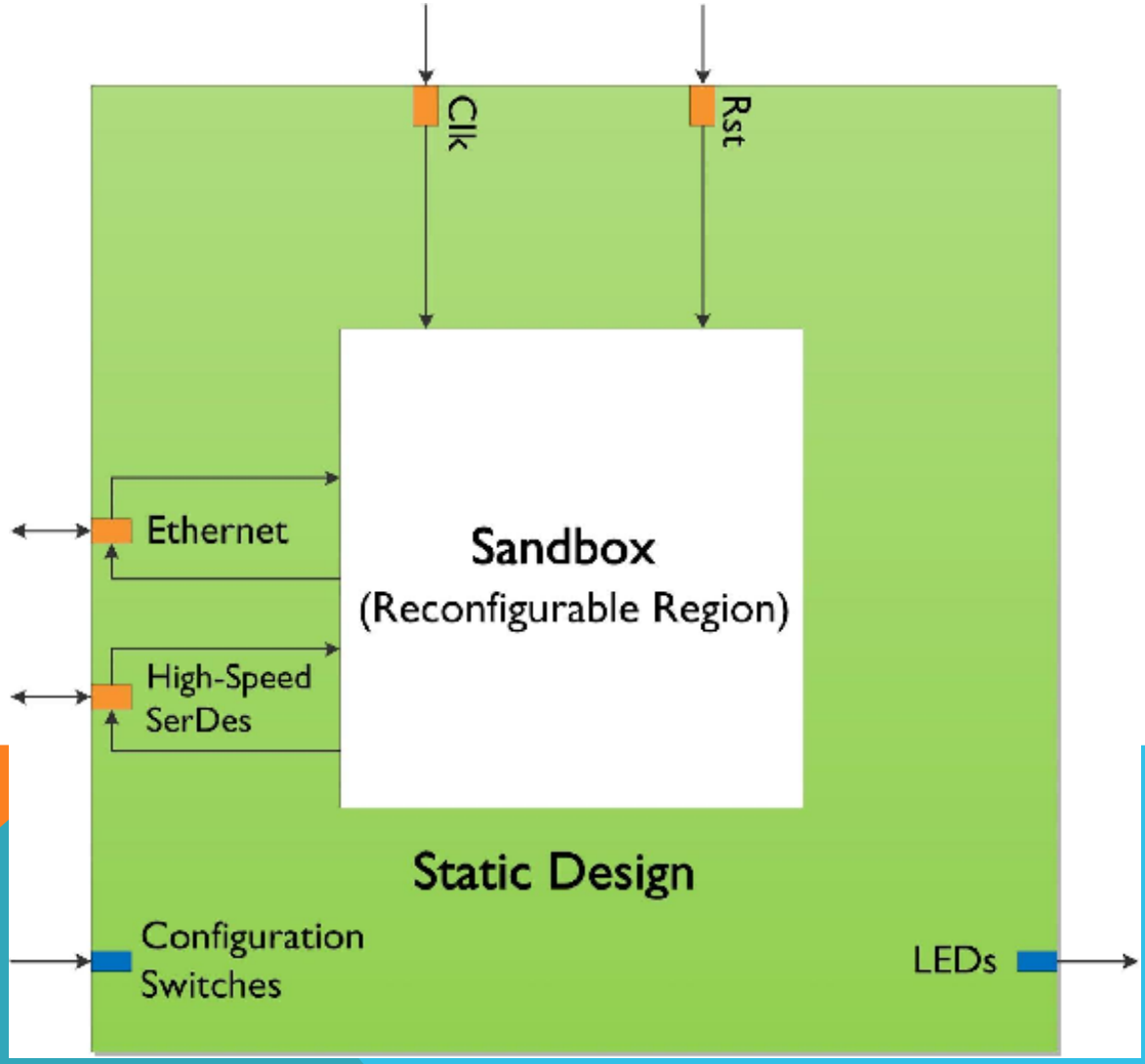
GNU Radio Host

COMPUTING MODEL

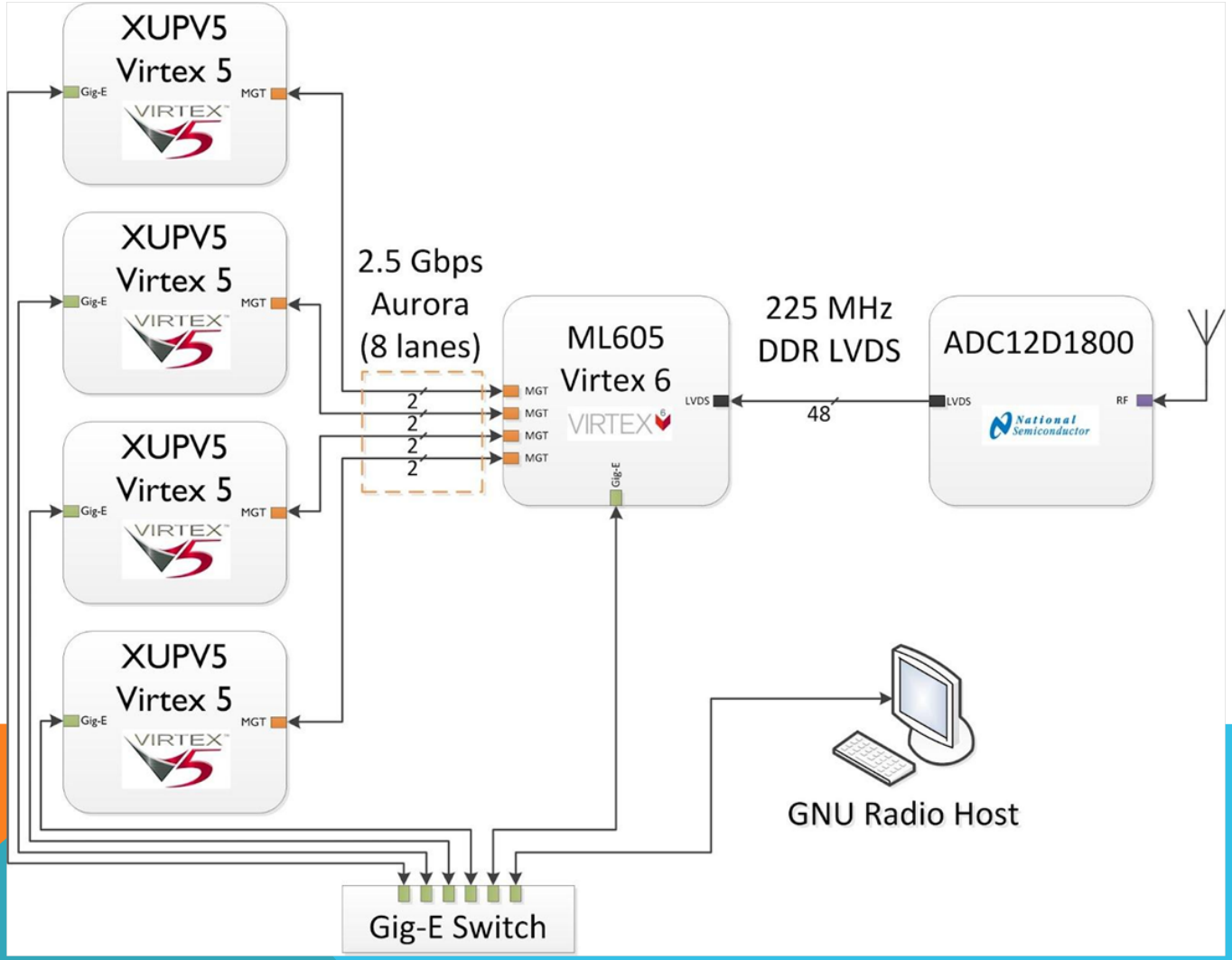
GNU Radio Host

Expansion beyond FPGAs

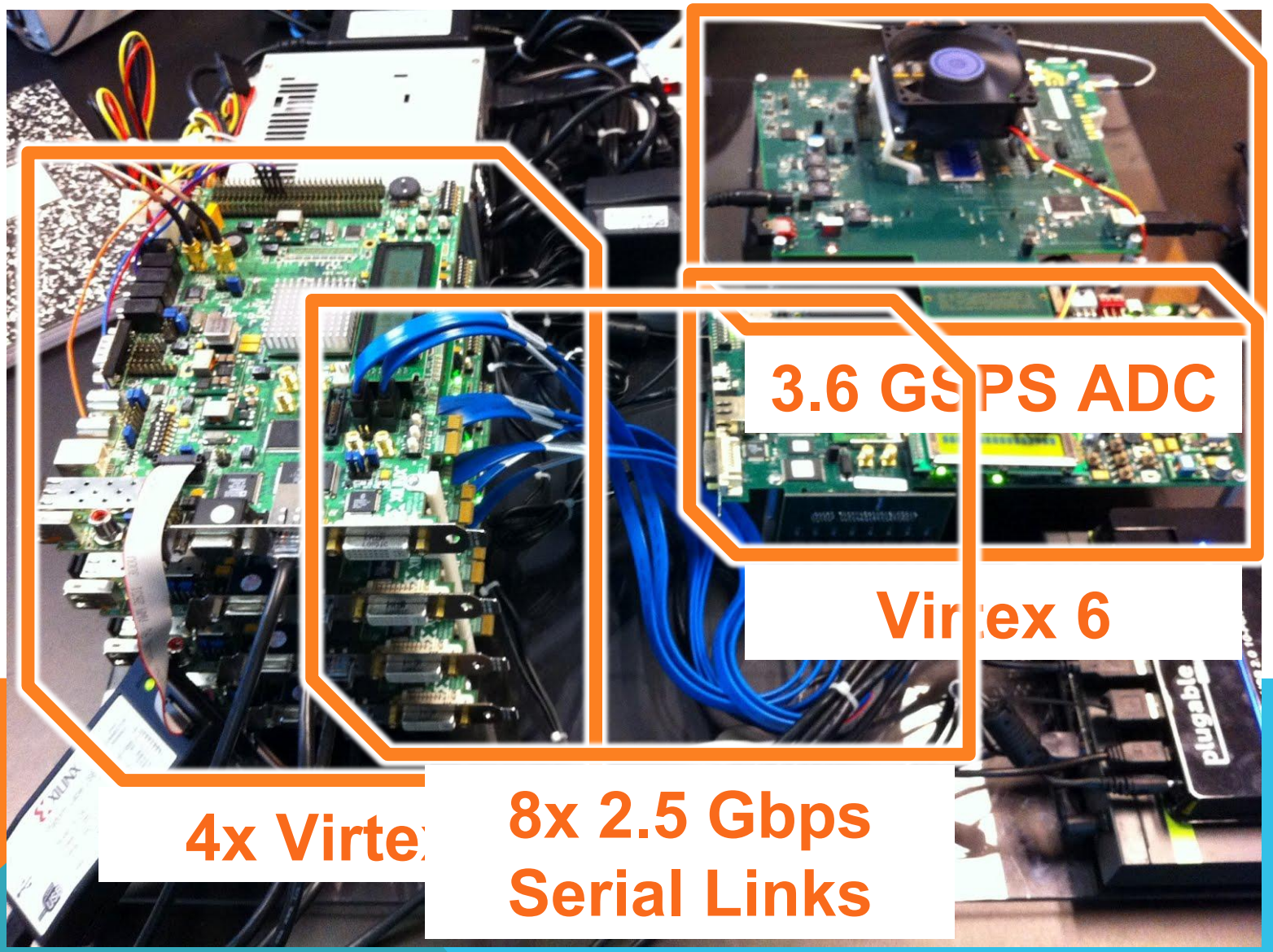
FPGA MODEL


Clk
Rst
Ethernet
High-Speed
SerDes
Sandbox
(Reconfigurable Region)
Static Design
Configuration
Switches
LEDs

HARDWARE DIAGRAM



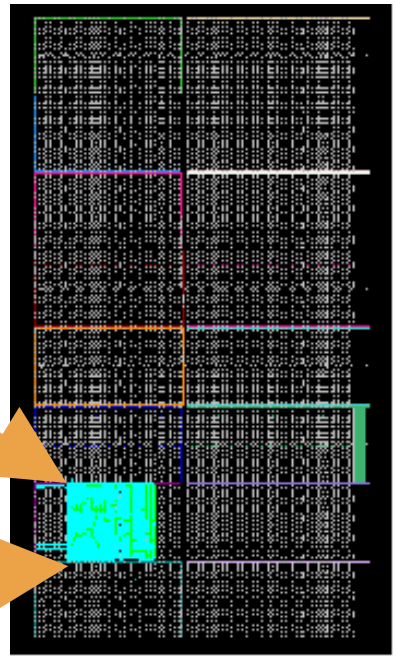
PROTOTYPE HARDWARE



LOW LEVEL DESIGN ASSEMBLY

PRECOMPILING A MODULE

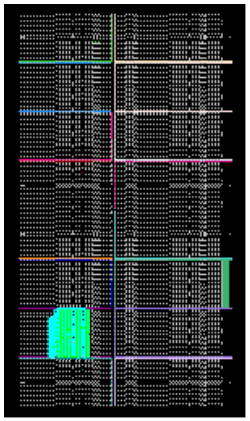
```
Module has 300 CLBs, 4 BRAMs, and 8 DSPs.
Module needs 294 CLBs, 3 BRAMs, and 7 DSPs.
Excess Capacity: 6 CLBs, 1 BRAMs, and 1 DSPs.
         8   10  12  14  16  18  20  22  24  26
100      M L M L M L M L B M M D M L M L M L M
         ------------------------------------
99       M L M L M L M L N M M N M L M L M L M
98       M L M L M L M L N M M N M L M L M L M
97       M L M L M L M L N M M N M L M L M L M
96       M L M L M L M L N M M N M L M L M L M
95       M L M L M L M L B M M D M L M L M L M
94       M L M L M L M L N M M N M L M L M L M
93       M L M L M L M L N M M N M L M L M L M
92       M L M L M L M L N M M N M L M L M L M
91       M L M L M L M L N M M N M L M L M L M
90       M L M L M L M L B M M D M L M L M L M
89       M L M L M L M L N M M N M L M L M L M
88       M L M L M L M L N M M N M L M L M L M
87       M L M L M L M L N M M N M L M L M L M
86       M L M L M L M L N M M N M L M L M L M
85       M L M L M L M L B M M D M L M L M L M
84       M L M L M L M L N M M N M L M L M L M
83       M L M L M L M L N M M N M L M L M L M
82       M L M L M L M L N M M N M L M L M L M
81       M L M L M L M L N M M N M L M L M L M
80       M L M L M L M L B M M D M L M L M L M
         ------------------------------------
79       M L M L M L M L N M M N M L M L M L M

Module constraints file (ucf) written to: bpskDemod.ucf
```



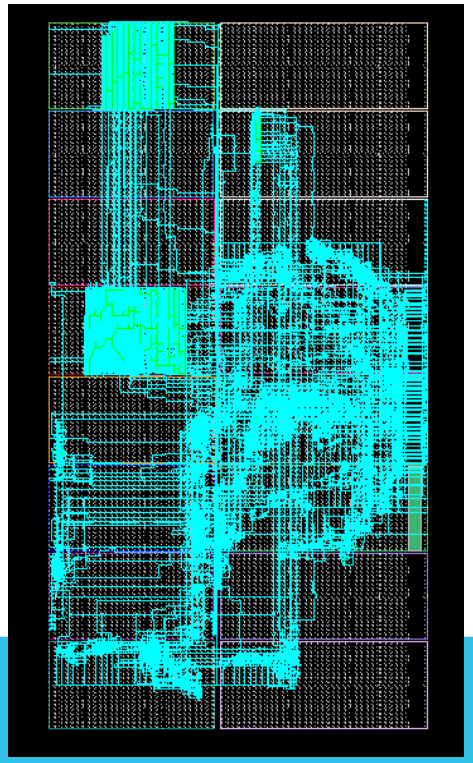
Module Size Estimation

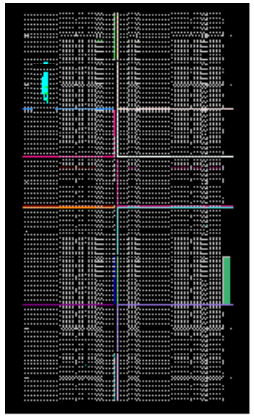
Placement on Chip

ASSEMBLING A DESIGN

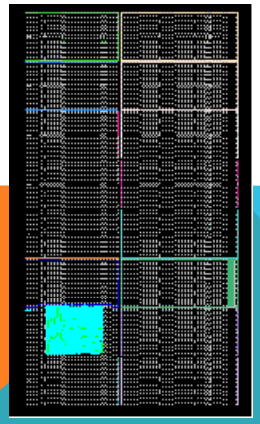


Decimator





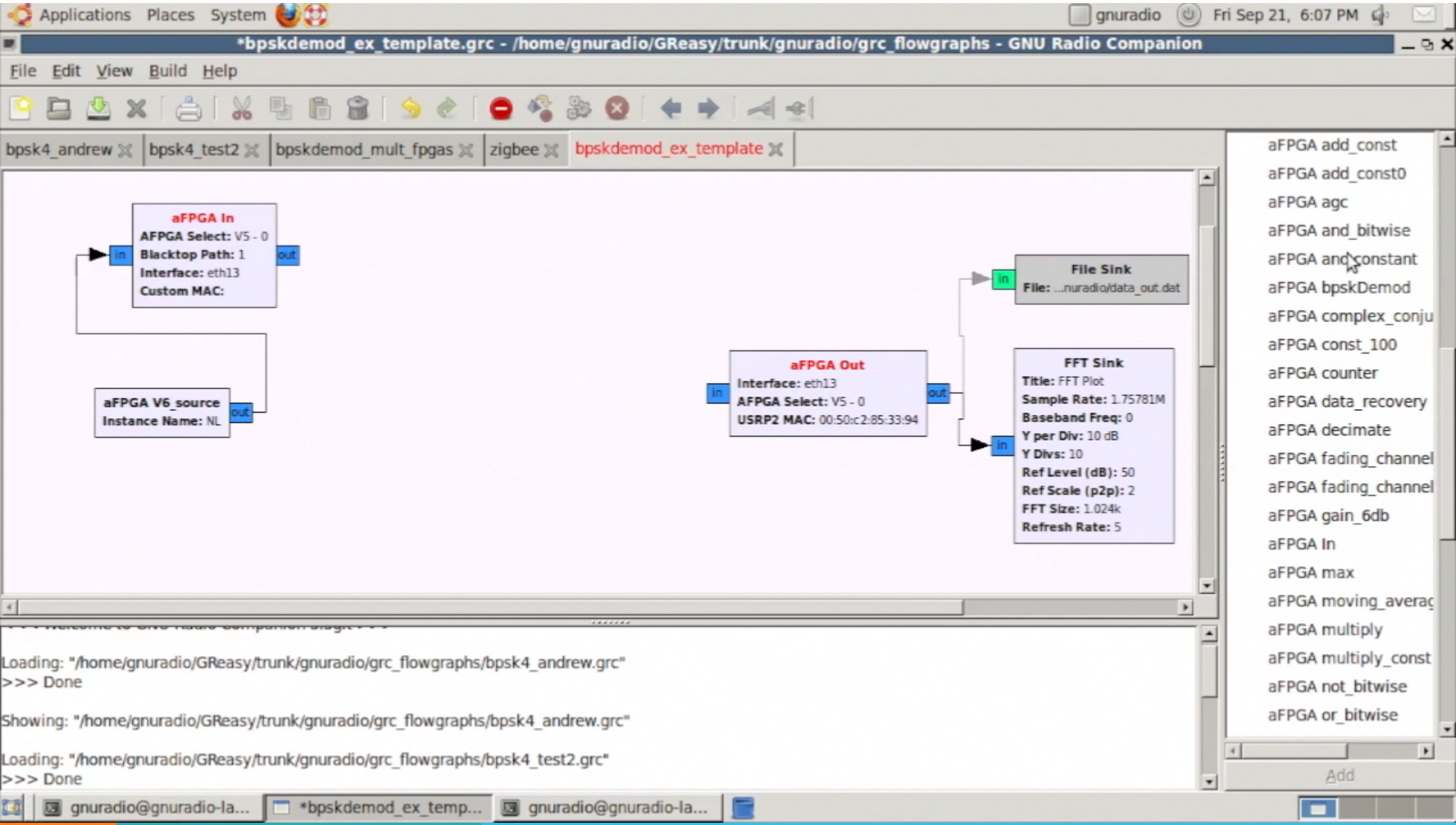
Data Recovery



BPSK Demod

DEMO: BUILDING A BPSK RECEIVER

DEMO


Applications Places System
gnuradio Fri Sep 21, 6:07 PM
*bpskdemod_ex_template.grc - /home/gnuradio/GReasy/trunk/gnuradio/grc_flowgraphs - GNU Radio Companion
File Edit View Build Help
bpsk4_andrew
bpsk4_test2
bpskdemod_mult_fpgas
zigbee
bpskdemod_ex_template
aFPGA In
AFPGA Select: V5 - 0
Blacktop Path: 1
Interface: eth13
Custom MAC:
in
out
aFPGA V6_source
Instance Name: NL
aFPGA Out
Interface: eth13
AFPGA Select: V5 - 0
USRP2 MAC: 00:50:c2:85:33:94
File Sink
File: ...nuradio/data_out.dat
FFT Sink
Title: FFT Plot
Sample Rate: 1.75781M
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 50
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 5
aFPGA add_const
aFPGA add_const0
aFPGA agc
aFPGA and_bitwise
aFPGA and_constant
aFPGA bpskDemod
aFPGA complex_conju
aFPGA const_100
aFPGA counter
aFPGA data_recovery
aFPGA decimate
aFPGA fading_channel
aFPGA fading_channel
aFPGA gain_6db
aFPGA In
aFPGA max
aFPGA moving_averag
aFPGA multiply
aFPGA multiply_const
aFPGA not_bitwise
aFPGA or_bitwise
Add
Loading: "/home/gnuradio/GReasy/trunk/gnuradio/grc_flowgraphs/bpsk4_andrew.grc"
>>> Done
Showing: "/home/gnuradio/GReasy/trunk/gnuradio/grc_flowgraphs/bpsk4_andrew.grc"
Loading: "/home/gnuradio/GReasy/trunk/gnuradio/grc_flowgraphs/bpsk4_test2.grc"
>>> Done
gnuradio@gnuradio-la...
*bpskdemod_ex_temp...
gnuradio@gnuradio-la...

IMPACT

- Expand application space for GNU Radio
- Significant reduction in radio design time
- Paradigm is extendable beyond FPGAs
- Enables designers to more readily explore "what if" radio construction

GREASY

QUESTIONS?