Designing an Automatic Gain Control for the bladeRF

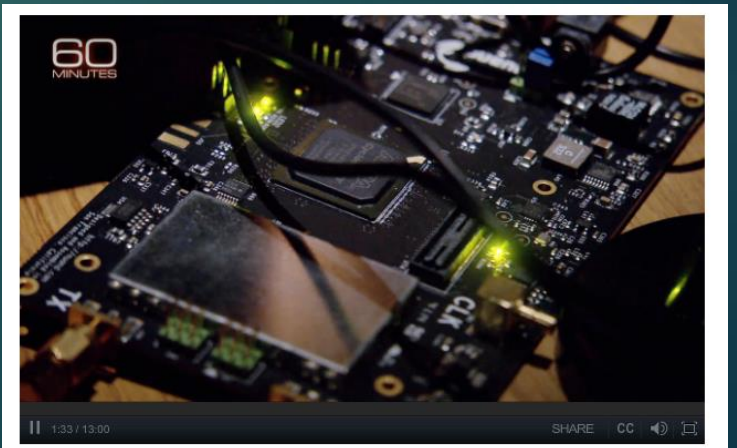
BY ROBERT GHILDUTA, NUAND LLC

Introduction

- Owner of Nuand, LLC
- Interests are software, hardware, and RF engineering
- Background in DSP (telecom, and acoustics)
- Previously: 6 years of professional experience building enterprise networking equipment
  - Worked on first wave1 and wave2 enterprise 802.11ac APs
- Long time information security researcher

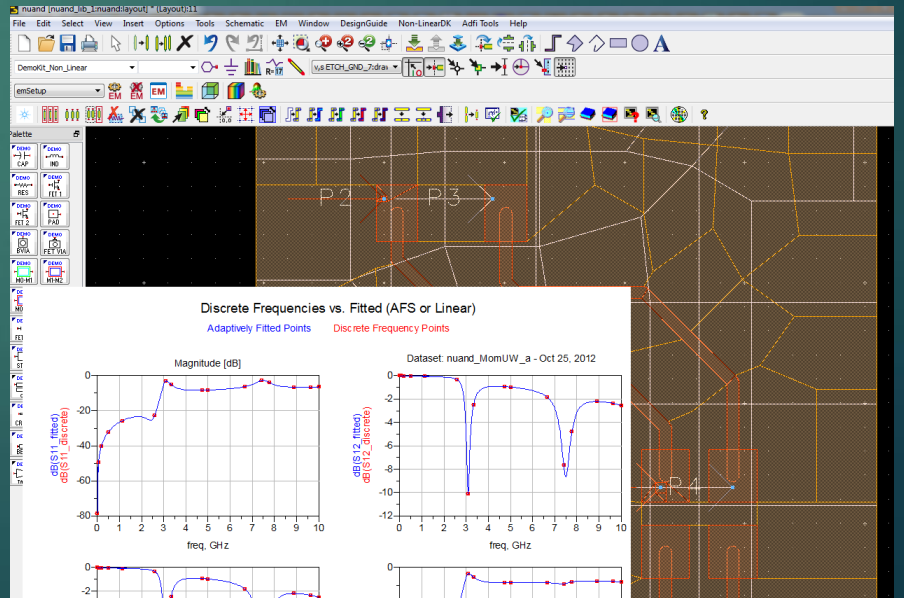
- Nuand is known for the bladeRF, a low cost USB3.0 software defined radio product line that is used by thousands of engineers, hobbyists, and researchers.
- Launched crowd-funding campaign for the original bladeRF in 2013
  - Revamped product line with bladeRF 2.0 micro
- Decades of combined engineering experience in Software Defined Radio, enterprise networking equipment, and defense
- Focusing on low-cost Software Defined Radios and applications
  - Support for GNURadio, gr-osmosdr, YateBTS, SDR#, SDRAngel, SRSLTE
  - lightweight C library and API, command line interface
  - OS support: Linux, Windows, macOS, FreeBSD

nuand

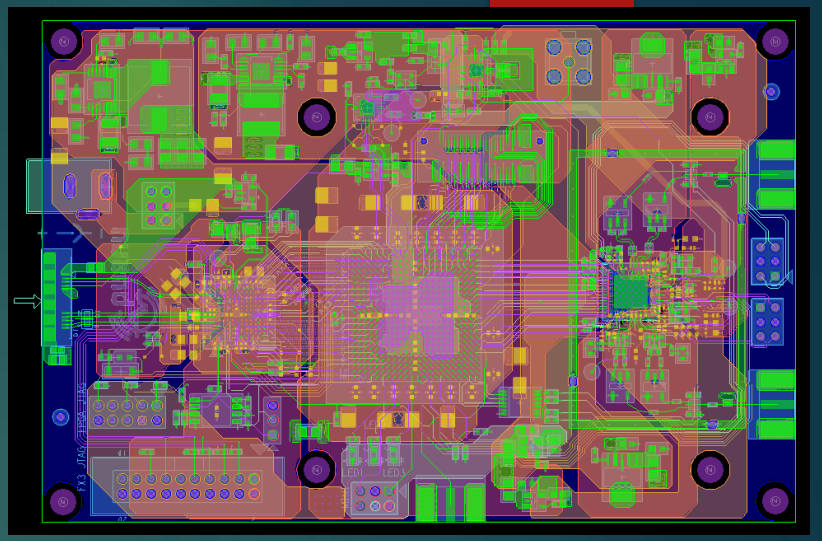


The design of bladeRF 1

- OrCad for schematics
  - 300 components
- Cadence for layout
  - 8 layer, 5/5 mil PCB
  - FR-4 TG
- Simulations done with HSPICE
  - 2.5 field solver

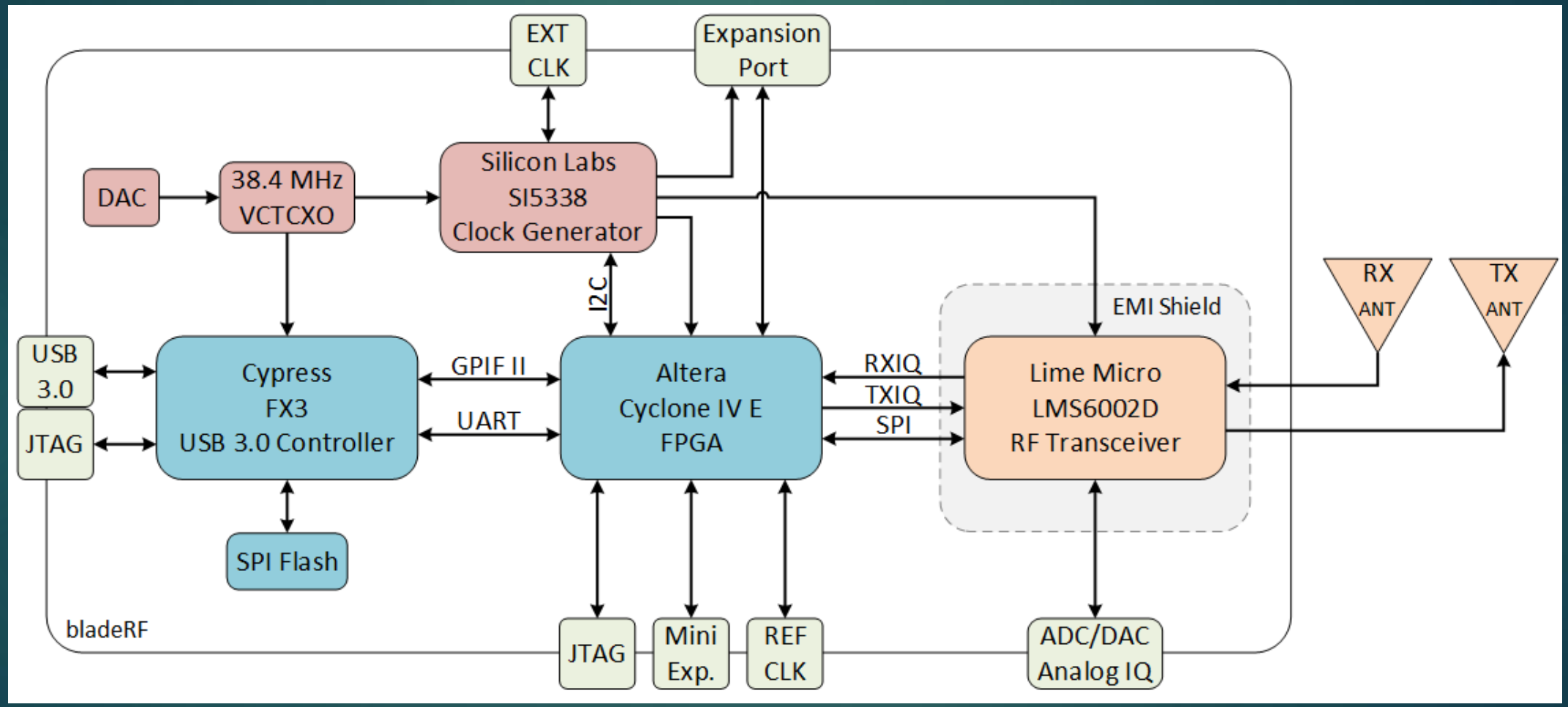


Architecture of bladeRF 1

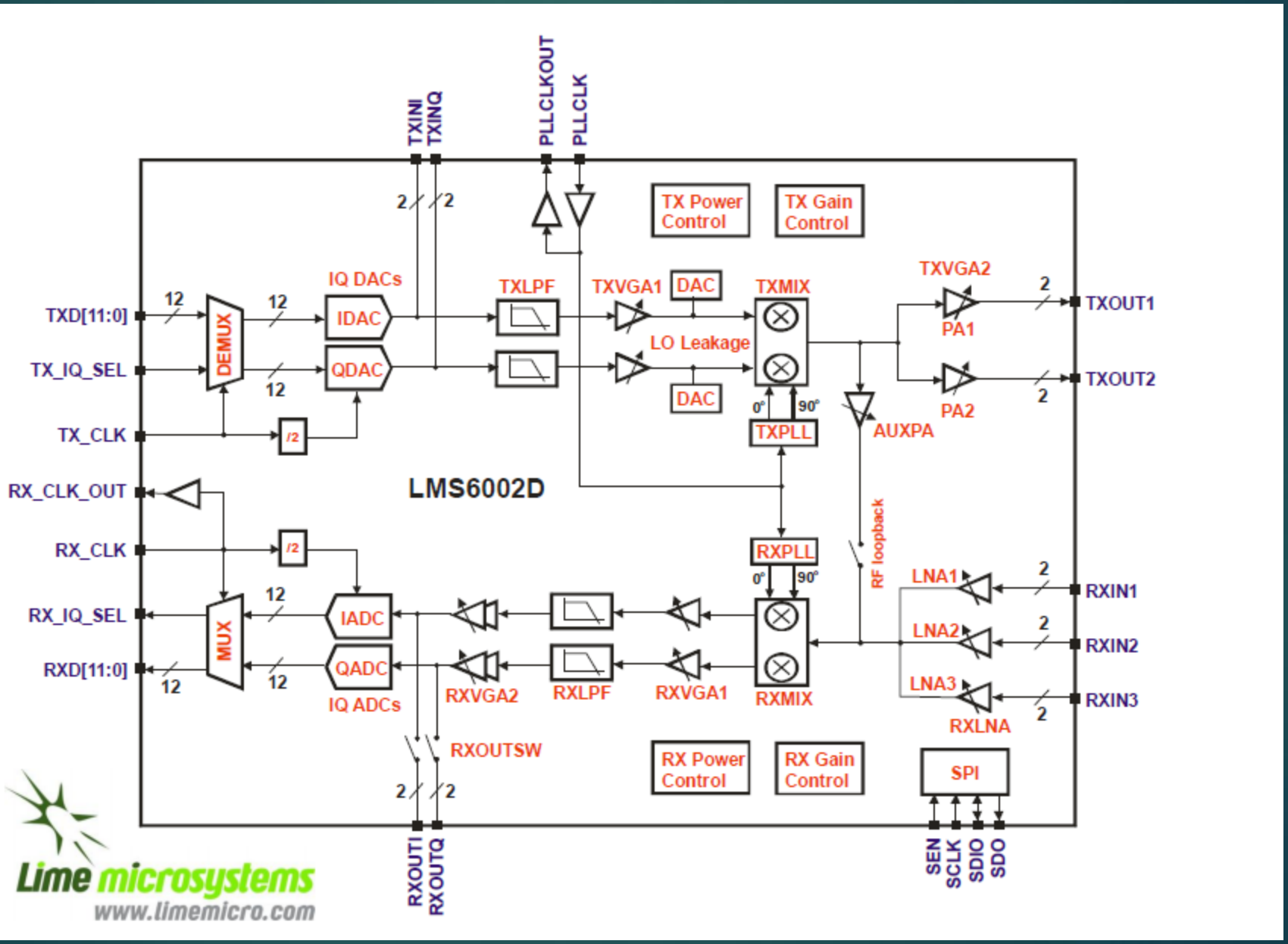


- Cypress FX3 USB 3.0 transceiver
  - Embedded 200MHz ARM7
- Altera Cyclone 4
- LimeMicro LMS6002D
  - "All in one" RF-to-bits MMIC
  - 300MHz – 3.8GHz RF range
  - Full duplex 12 bit 40MSPS quadrature sampling
- Si5338 synthesizer
  - SiLab's top of the line synthesizer
- Development board
  - GPIO, LEDs, all JTAG connector populated

bladeRF 1 block diagram



LMS6002D


TXINI
TXINQ
PLLCLKOUT
PLLCLK
TX Power Control
TX Gain Control
IQ DACs
TXLPF
TXVGA1
DAC
TXMIX
TXVGA2
TXD[11:0]
TX_IQ_SEL
TX_CLK
DEMUX
IDAC
QDAC
/2
LO Leakage
PA1
PA2
TXOUT1
TXOUT2
AUXPA
TXPLL
0°
90°
LMS6002D
RX_CLK_OUT
RX_CLK
RX_IQ_SEL
RXD[11:0]
MUX
IADC
QADC
IQ ADCs
RXPLL
RF loopback
LNA1
LNA2
LNA3
RXIN1
RXIN2
RXIN3
RXLNA
RXVGA2
RXLPF
RXVGA1
RXMIX
RXOUTSW
RX Power Control
RX Gain Control
SPI
RXOUTI
RXOUTQ
SEN
SCLK
SDIO
SDO
Lime microsystems
www.limemicro.com

Incoming signals

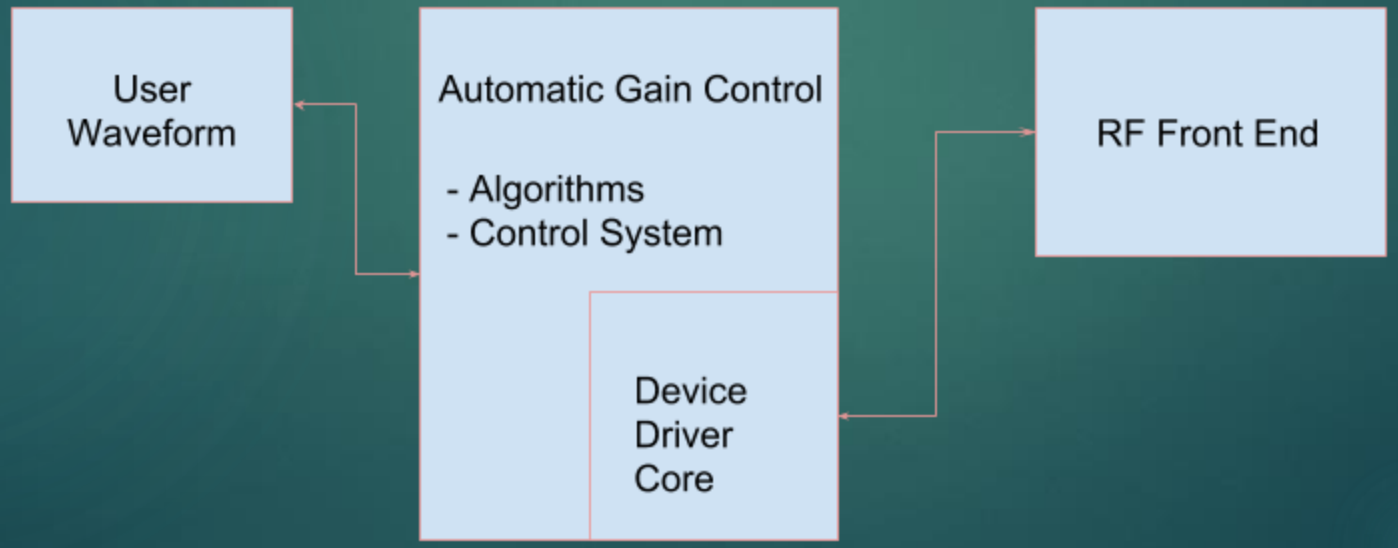
- Realistically you will see things between -110dBm to -25dBm.
  - That's a 85dB range!
- Problem….
  - Analog to Digital Converter (ADC) has a limited sensitivity range
    - Each bit is ~6.02dB
    - 12 bits is ~72.24dB
  - So a 12bit ADC does not have the ability to simultaneously listen to a -25dBm and -100dBm signal
  - This is known as static range

RF frontend

- So how do we have a frontend that can listen to a -110dBm signal as well as a -25dBm signal?
  - They cannot occur simultaneously
  - So how about packets?
    - Change gains on a per packet basis by adjusting the RF frontend's gains
      - LNA1 (0 to 6dB)
      - RXVGA1 (5dB to 30dB)
      - RXVGA2 (0dB to 30dB)
    - This is known as dynamic range
    - Total dynamic range of bladeRF 1's RF gain stages is 66dB
    - Total dynamic range of bladeRF 1's sensitivity is 66dB+72dB=138dB

Automatic Gain Control Framework

- Create a framework where user code (waveform) interacts with a device-independent AGC interface
  - Contains IIR, decisions to increase gain, etc
- Device Driver contain gain strategy tables, and all necessary logic to change gain

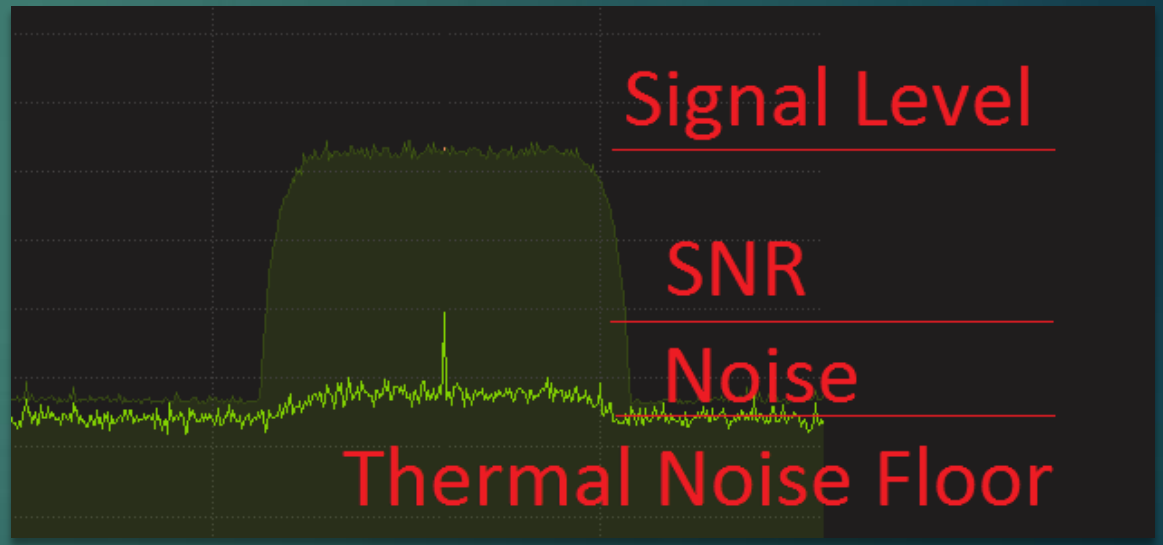


Motivations

- AGC is tightly coupled to the waveform being developed
- Unified AGC helps more easily port waveforms between devices
- Reusable RF Front End Drivers
- Unify behavior of AGC
  - Digital Interface
    - Request hold
    - Signal valid
  - Performance
    - Settling times

What order should the gains be adjusted in?

- Combat noise figure
  - Amplifiers amplify and add noise
  - Quickly separate signal from noise floor
- Friis formulas for noise basically says:
  - It's best to first turn up gain (and noise) of the LNA



Recommended gain tables

5.18 What is recommended gain table for the receiver?

Receiver gain table versus modulated input signal level 3GPP Band 1 (1950MHz)

Min. Signal (dBm)	Max. Signal (dBm)	SNR(dB) Min.	SNR(dB) Max.	Antenna Switch	LNA Gain	RxVGA1 gain	RxVGA2 gain
-117 **	-85	-17.4	14.6	-	Max.	Max.	Max.
-84 *	-53	15.6	46.6	-	Max.	Max.	Variable
-52	-28	47.6	71	-	Max.	Variable	Min.
-27	-22	69.46	74.46	-	Mid (Max - 6 dB)	Min.	Min.
-21	-13	67.98	75.98	-	Bypassed	Min.	Min.
-12	4	60.5	76.48	Switched	Bypassed	Min.	Min.

Receiver gain table versus modulated input signal level for Band 5 (840 MHz)

Min. Signal (dBm)	Max. Signal (dBm)	SNR(dB) Min.	SNR(dB) Max.	Antenna Switch	LNA Gain	RxVGA1 gain	RxVGA2 gain
-119 **	-85	-19.49	14.5	-	Max.	Max.	Max.
-84 *	-53	15.5	46.5	-	Max.	Max.	Variable
-52	-39	47.5	54.75	-	Max.	Variable	Min.
-38	-32	51.25	57.25	-	Mid (Max - 6 dB)	Min.	Min.
-31	-13	51.5	69.5	-	Bypassed	Min.	Min.
-12	4	-	-	Switched	Bypassed	Min.	Min.

bladeRF 1 implementation

- 6 different gains will take far too long to settle
  - Each gain that is changed is one separate SPI transaction
  - I wanted a fast AGC
- Solution:
  - 3 gain settings, High/Mid/Low
  - At most only 2 variables change (33% faster than changing 3 variables!)

High gain: -82dBm - -52dBm
lna_gain => 6 dB (Max)
rxvga1_gain => 30 dB
rxvga2_gain => 15 dB

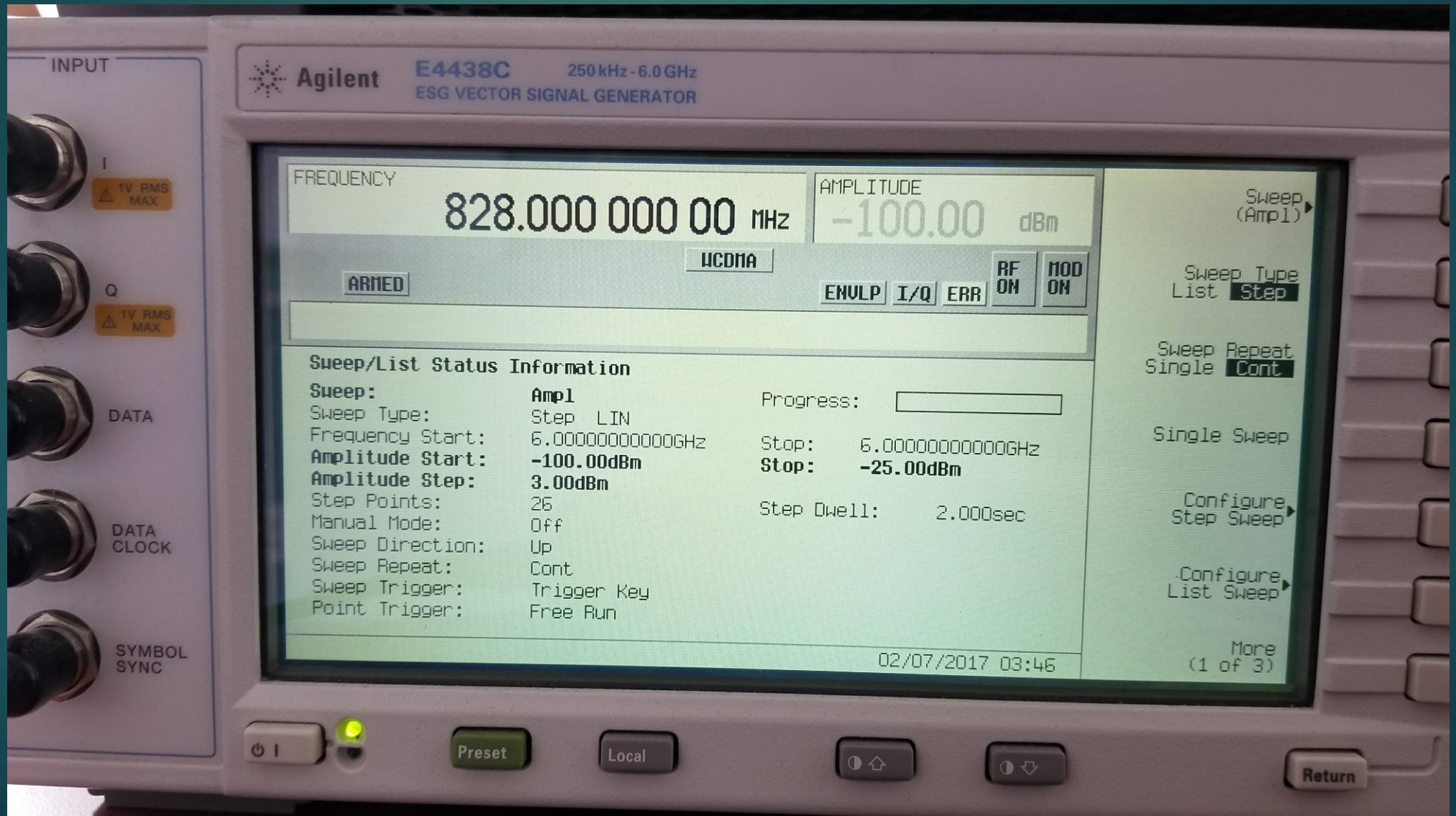
Mid gain: -52dBm - -30dBm
lna_gain => 3 dB (Mid)
rxvga1_gain => 30 dB
rxvga2_gain => 0 dB

Low gain: -30dBm - -17dBm
lna_gain => 3 dB (Mid)
rxvga1_gain => 12 dB
rxvga2_gain => 0 dB

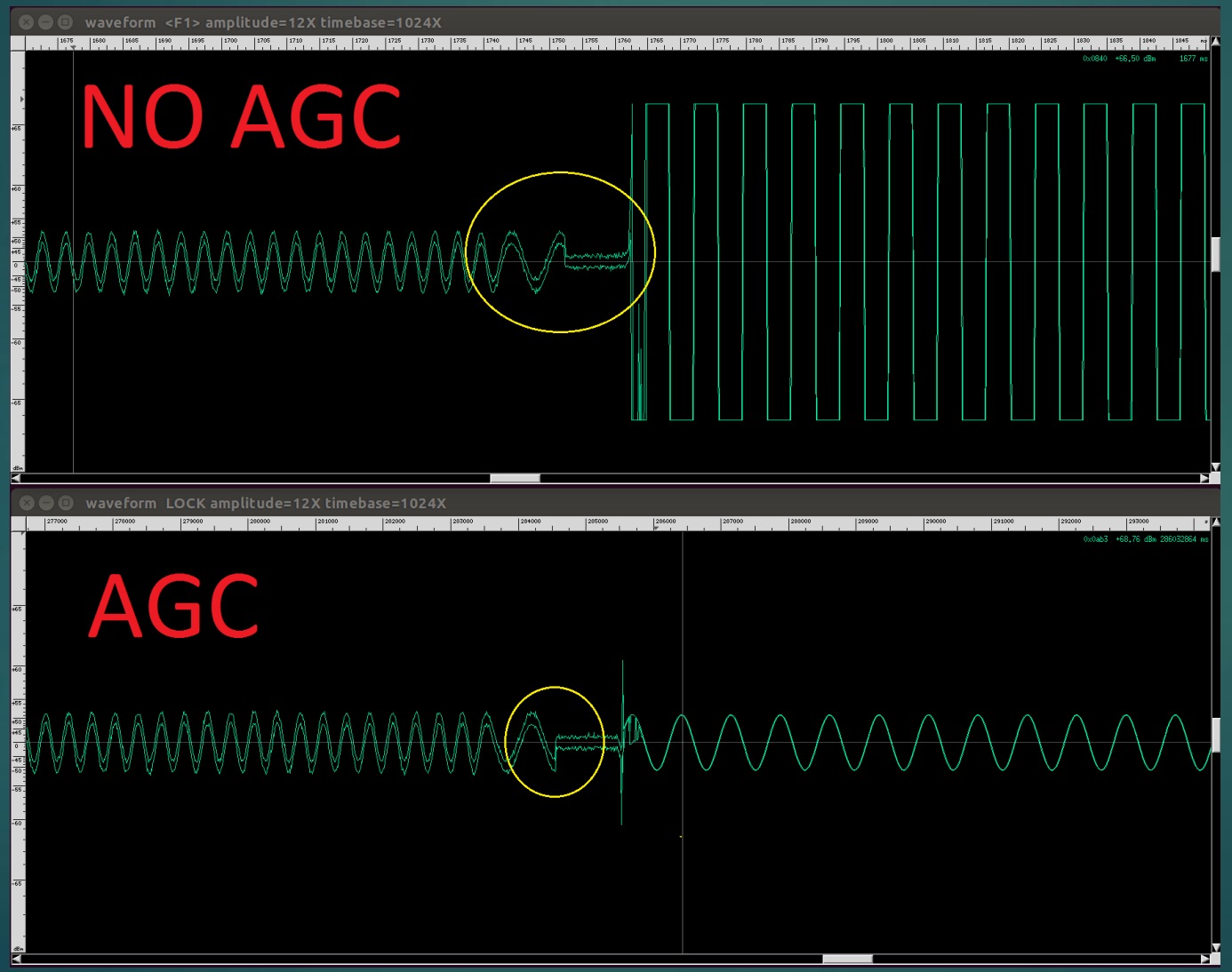
Now you know the gains – how do you make this automatic?

- Starting configuration
  - Gains should be turned up high
    - It's better to hear a signal and clip than to not hear a signal and miss it
- Gains should decrease as a "hot" signal comes in
  - 12-bit ADC means you'll have a -2047 to 2048 range on IQ samples
  - Decrease gain when within 3dB of hitting the top value (2048)
- Problems
  - Gain stages have group delay
    - You need to wait a little bit
  - Signal has a high PAPR or is very noisy
    - IIR
    - Solution: Hysteresis
  - Signal is very hot
    - Shortcut settling time

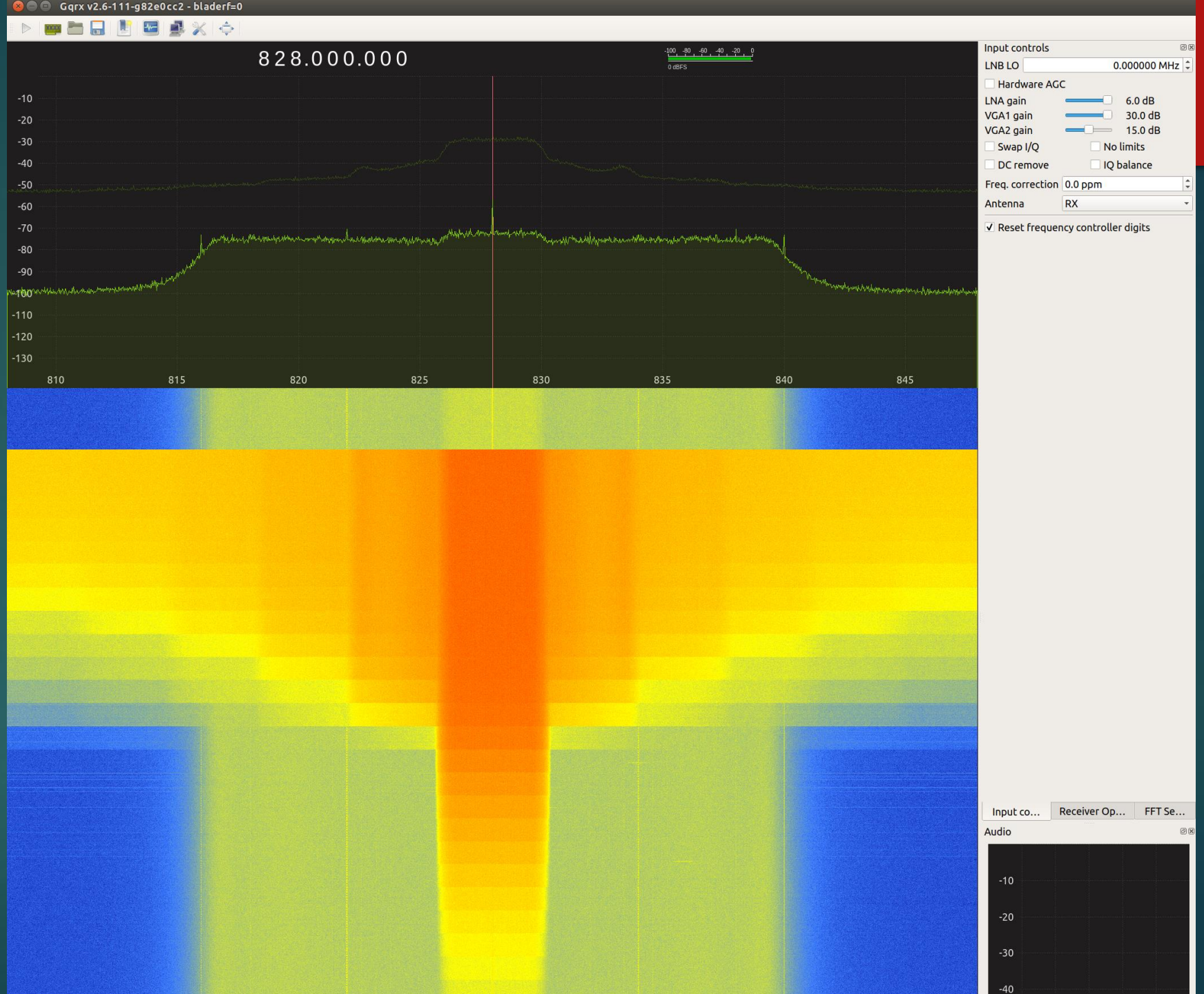
Tested using one of these

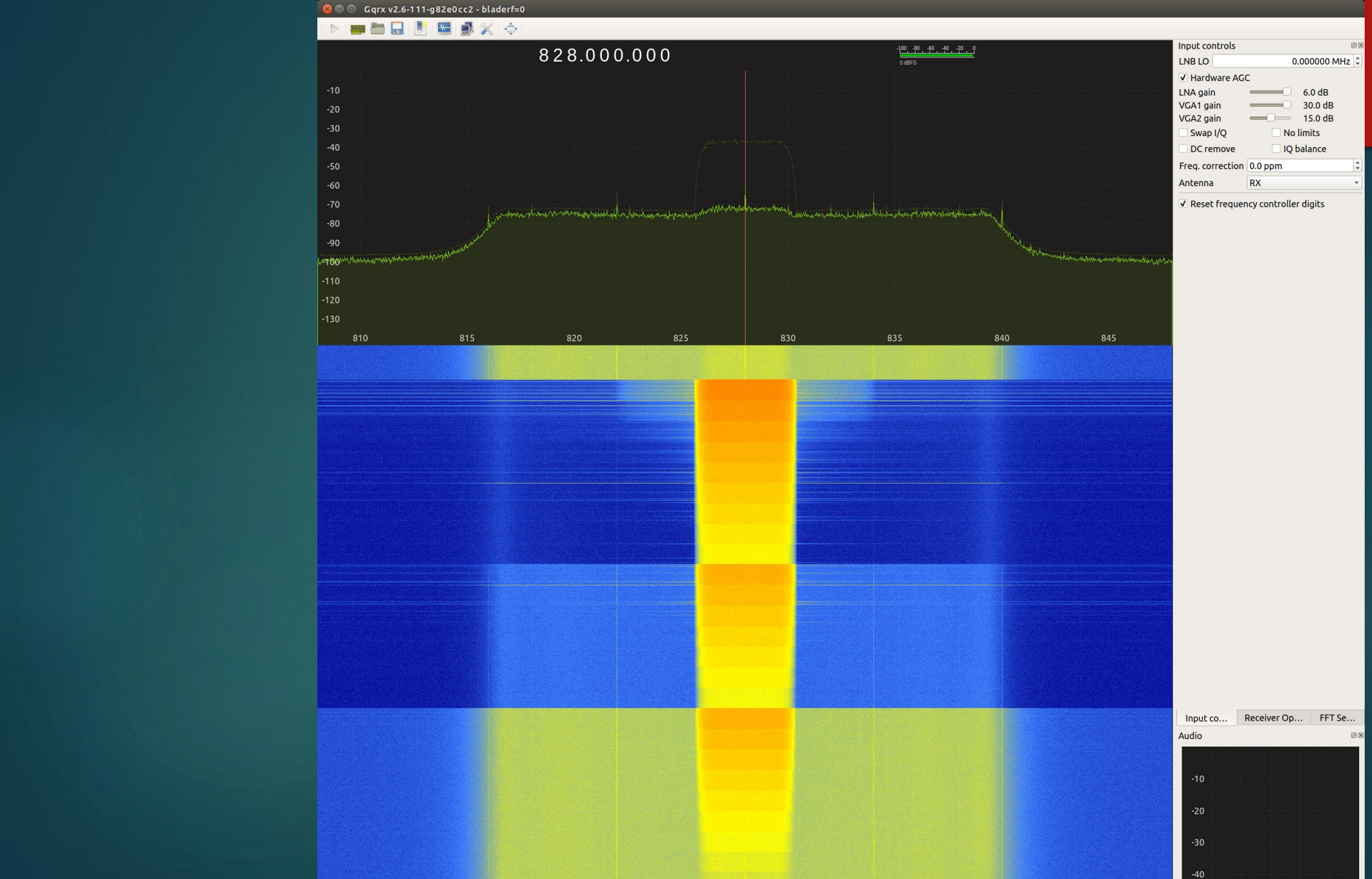


Baudline

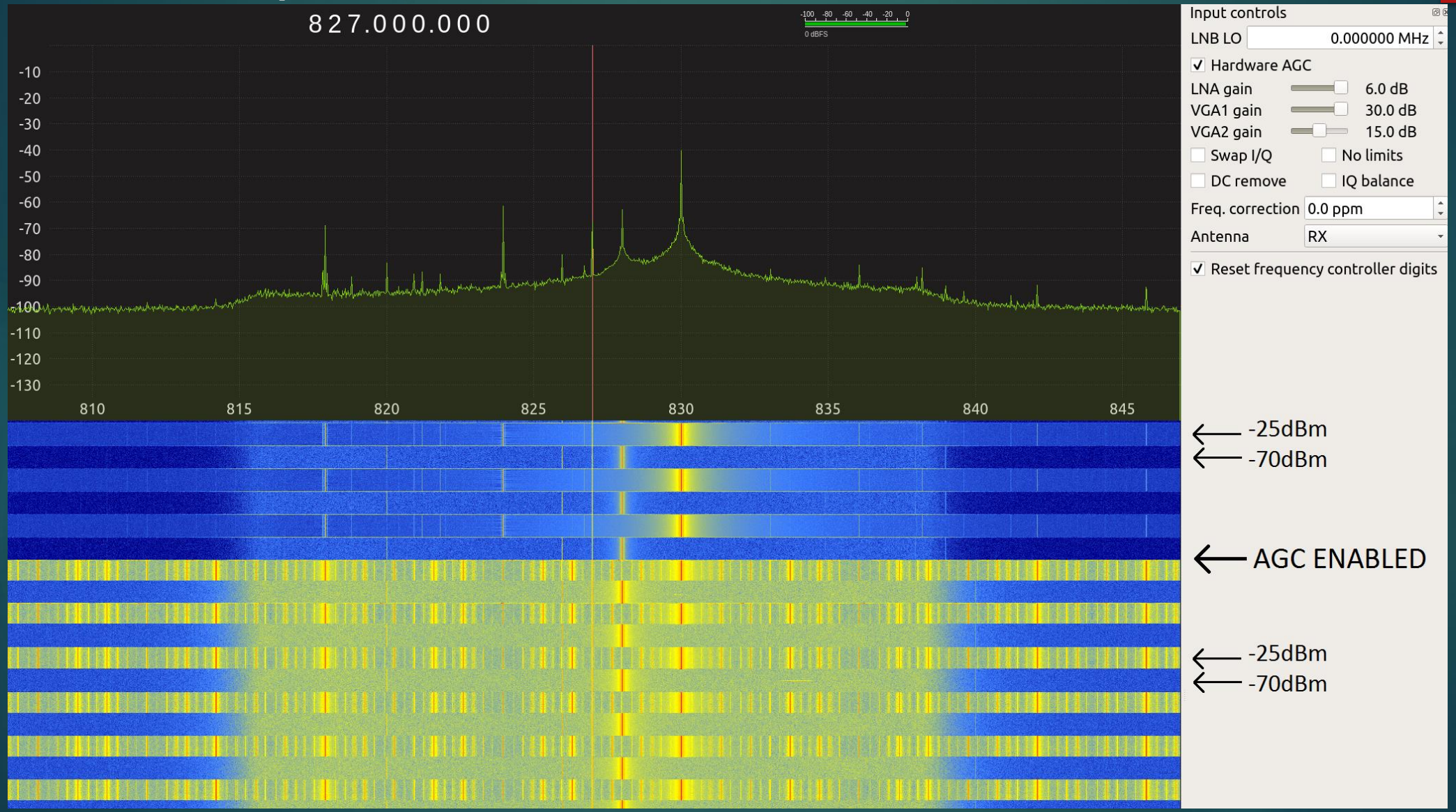

waveform <F1> amplitude=12X timebase=1024X
NO AGC
waveform LOCK amplitude=12X timebase=1024X
AGC

GQRX


Gqrx v2.6-111-g82e0cc2 - bladerf=0
828.000.000
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
-120
-130
810
815
820
825
830
835
840
845
Input controls
LNB LO
0.000000 MHz
Hardware AGC
LNA gain
6.0 dB
VGA1 gain
30.0 dB
VGA2 gain
15.0 dB
Swap I/Q
No limits
DC remove
IQ balance
Freq. correction
0.0 ppm
Antenna
RX
Reset frequency controller digits
Input co…
Receiver Op…
FFT Se…
Audio
-10
-20
-30
-40


Gqrx v2.6-111-g82e0cc2 - bladerf=0
828.000.000
-100 -80 -60 -40 -20 0
0 dBFS
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
-120
-130
810
815
820
825
830
835
840
845
Input controls
LNB LO
0.000000 MHz
Hardware AGC
LNA gain
6.0 dB
VGA1 gain
30.0 dB
VGA2 gain
15.0 dB
Swap I/Q
No limits
DC remove
IQ balance
Freq. correction
0.0 ppm
Antenna
RX
Reset frequency controller digits
Input co…
Receiver Op…
FFT Se…
Audio
-10
-20
-30
-40

GQRX pt2


827.000.000
-100 -80 -60 -40 -20 0
0 dBFS
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
-120
-130
810
815
820
825
830
835
840
845
Input controls
LNB LO 0.000000 MHz
Hardware AGC
LNA gain 6.0 dB
VGA1 gain 30.0 dB
VGA2 gain 15.0 dB
Swap I/Q
No limits
DC remove
IQ balance
Freq. correction 0.0 ppm
Antenna RX
Reset frequency controller digits
-25dBm
-70dBm
AGC ENABLED
-25dBm
-70dBm

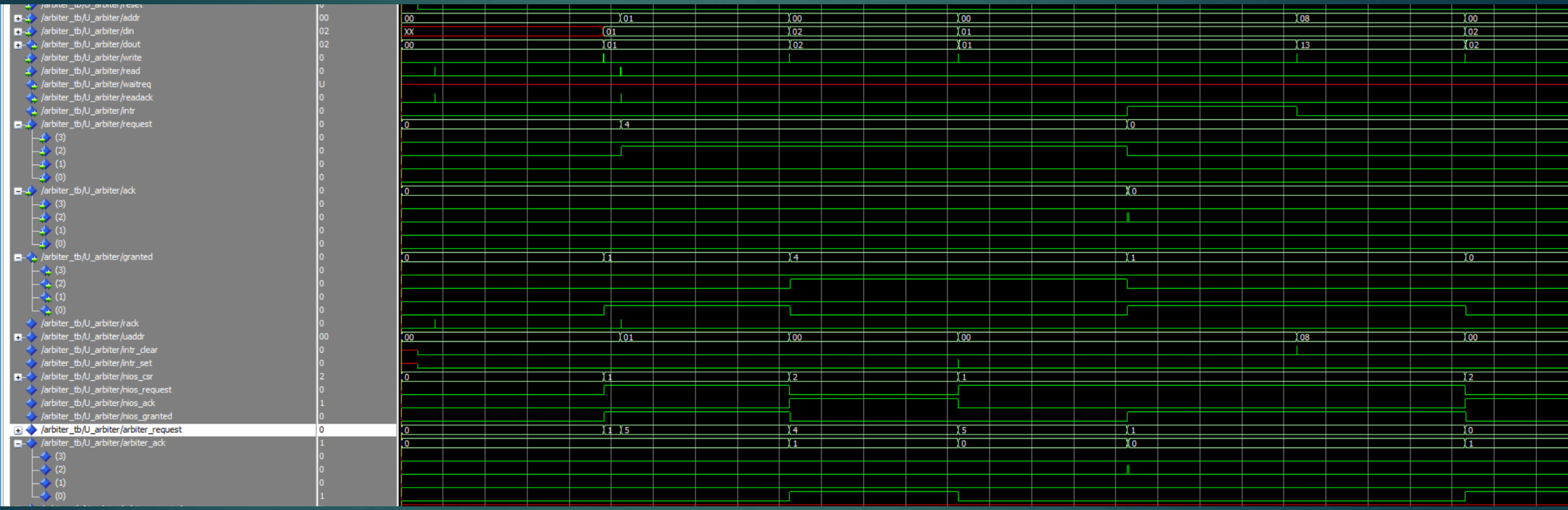
FPGA Development

- Altera Cyclone IV (40 or 115 kLE)
  - Interfaces with:
    - LMS6002D Transceiver
    - SI5338 Clock Controller
  - Timestamp support
  - (Manual) IQ Balance
    - Automated in GNU-Radio
- bladeRF FPGA support written in VHDL
- Quartus II 16.0 Web Edition ($0)
    - Can use SignalTap if you enabled TalkBack

HDL logic

- SPI Bus sharing (bus arbiter)
  - LMS controller only has one set of SPI controller lines
    - Commands can come in from NiOS (libusb)
    - AGC should have priority
  - Priority round-robin implementation

Arbiter


/arbiter_tb/U_arbiter/addr
/arbiter_tb/U_arbiter/din
/arbiter_tb/U_arbiter/dout
/arbiter_tb/U_arbiter/write
/arbiter_tb/U_arbiter/read
/arbiter_tb/U_arbiter/waitreq
/arbiter_tb/U_arbiter/readack
/arbiter_tb/U_arbiter/intr
/arbiter_tb/U_arbiter/request
(3)
(2)
(1)
(0)
/arbiter_tb/U_arbiter/ack
(3)
(2)
(1)
(0)
/arbiter_tb/U_arbiter/granted
(3)
(2)
(1)
(0)
/arbiter_tb/U_arbiter/rack
/arbiter_tb/U_arbiter/uaddr
/arbiter_tb/U_arbiter/intr_clear
/arbiter_tb/U_arbiter/intr_set
/arbiter_tb/U_arbiter/nios_csr
/arbiter_tb/U_arbiter/nios_request
/arbiter_tb/U_arbiter/nios_ack
/arbiter_tb/U_arbiter/nios_granted
/arbiter_tb/U_arbiter/arbiter_request
/arbiter_tb/U_arbiter/arbiter_ack
(3)
(2)
(1)
(0)

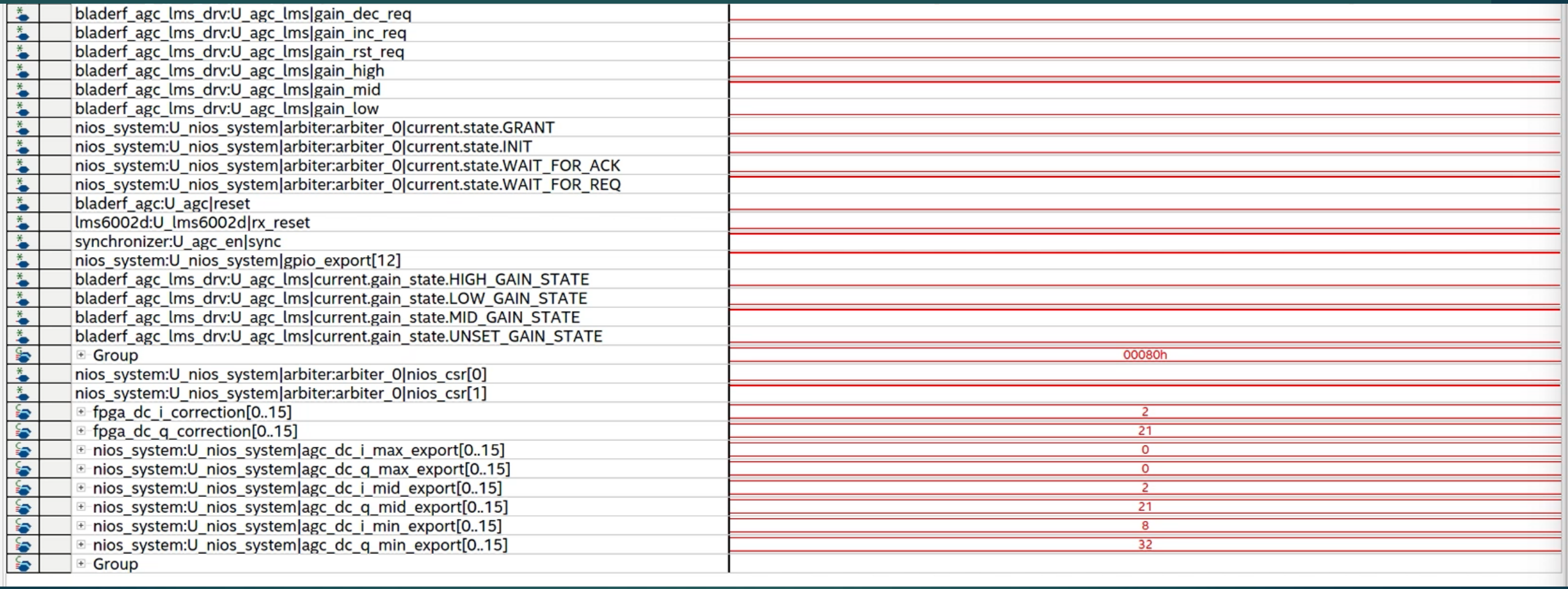
Transceiver imperfections

- AGC's additional RF problem
  - Frequency specific DC offset
    - Every time the gain is changed the DC IQ cal changes
- IQ imbalance and DC offset are an analog phenomenon in Zero IF transceivers
- Most transceivers have the ability to use analog correction methods
  - Opamps and programmable chargepumps to subtract out DC offset
- DC Offset
  - Expected in zero IF architectures
  - Analog values not "resting" at exactly 0
  - Symptoms
    - Y offset in time domain
    - "Large" spike in frequency domain
    - Uncentered constellation
  - Huge problem for AGC because power = sqrt(I^2+Q^2)

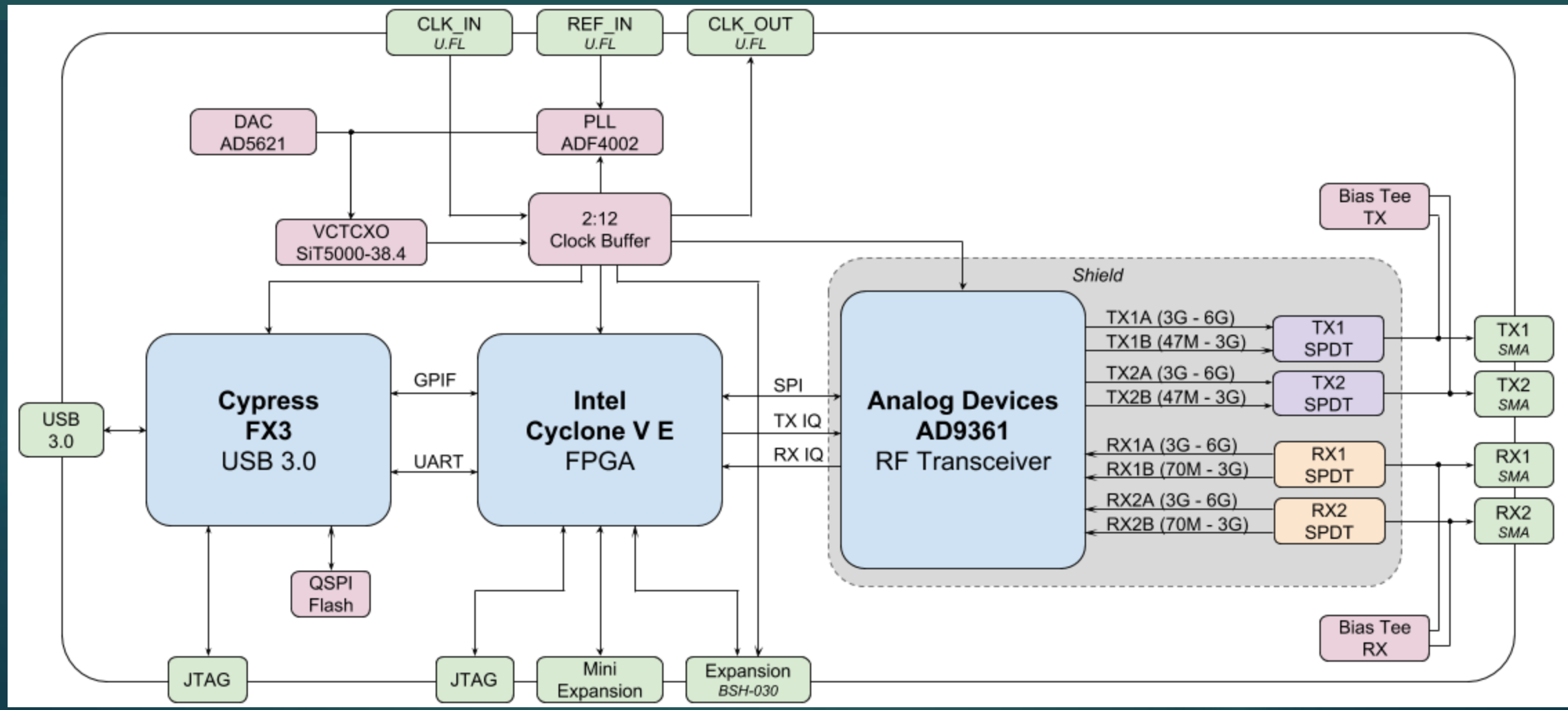
- IQ imbalance
  - IQ amplitude differences and phase offsets
  - Most noticeable as "mirror image" at $\pm F_{sig}$
  - HW/SW compensation
    - gr-iqbal
- IQ imbalance is gain setting agnostic but DC offset is not
  - The closer the gain stage is to the ADC the more of an effect it has
- libbladeRF
  - Calculates settings of analog DC cal stages
    - Too impractical to change these values when changing gain
    - Easiest solution is to keep analog compensators the same and just subtract out DC offset
      - Calculated during new DC cal

Solution?

- DC lookup table mux
- Calibrate DC offsets at max gain value across band at 100MHz increments
  - Calibrate DC offset opamps
  - Then adjust gains to mid and low values and observe DC mean error
  - Save DC correction values and mean error values for high, mid, low gain settings
- At runtime libbladeRF loads appropriate high, mid, low gain settings
  - These are exported in HDL
  - The AGC tells a mux its setting
  - The mux then selects the appropriate mean error value
  - The correction block then removes the DC offset


bladerf_agc_lms_drv:U_agc_lms|gain_dec_req
bladerf_agc_lms_drv:U_agc_lms|gain_inc_req
bladerf_agc_lms_drv:U_agc_lms|gain_rst_req
bladerf_agc_lms_drv:U_agc_lms|gain_high
bladerf_agc_lms_drv:U_agc_lms|gain_mid
bladerf_agc_lms_drv:U_agc_lms|gain_low
nios_system:U_nios_system|arbiter:arbiter_0|current.state.GRANT
nios_system:U_nios_system|arbiter:arbiter_0|current.state.INIT
nios_system:U_nios_system|arbiter:arbiter_0|current.state.WAIT_FOR_ACK
nios_system:U_nios_system|arbiter:arbiter_0|current.state.WAIT_FOR_REQ
bladerf_agc:U_agc|reset
lms6002d:U_lms6002d|rx_reset
synchronizer:U_agc_en|sync
nios_system:U_nios_system|gpio_export[12]
bladerf_agc_lms_drv:U_agc_lms|current.gain_state.HIGH_GAIN_STATE
bladerf_agc_lms_drv:U_agc_lms|current.gain_state.LOW_GAIN_STATE
bladerf_agc_lms_drv:U_agc_lms|current.gain_state.MID_GAIN_STATE
bladerf_agc_lms_drv:U_agc_lms|current.gain_state.UNSET_GAIN_STATE
Group
00080h
nios_system:U_nios_system|arbiter:arbiter_0|nios_csr[0]
nios_system:U_nios_system|arbiter:arbiter_0|nios_csr[1]
fpga_dc_i_correction[0..15]
2
fpga_dc_q_correction[0..15]
21
nios_system:U_nios_system|agc_dc_i_max_export[0..15]
0
nios_system:U_nios_system|agc_dc_q_max_export[0..15]
0
nios_system:U_nios_system|agc_dc_i_mid_export[0..15]
2
nios_system:U_nios_system|agc_dc_q_mid_export[0..15]
21
nios_system:U_nios_system|agc_dc_i_min_export[0..15]
8
nios_system:U_nios_system|agc_dc_q_min_export[0..15]
32
Group

bladeRF 2.0 micro



bladeRF 2.0 micro

- Improved RF, power, and form factor performance over original bladeRF
- Characteristics and capabilities
  - 2x2 MIMO, 61.44MSPS sampling rate, 56MHZ IBW
  - 47MHz to 6GHz frequency range, shielding and spur mitigation for high SFDR performance
  - Cyclone V FPGA – xA4 (49KLE) and xA9 (with largest-in-class 301KLE FPGA)
- Bias-Tee Modules – Powered directly by the bladeRF 2.0 micro
  - Wideband Low-Noise Amplifiers and Power Amplifiers, and (soon) Filters