Software Defined Radio with GNU Radio

Tom Rondeau

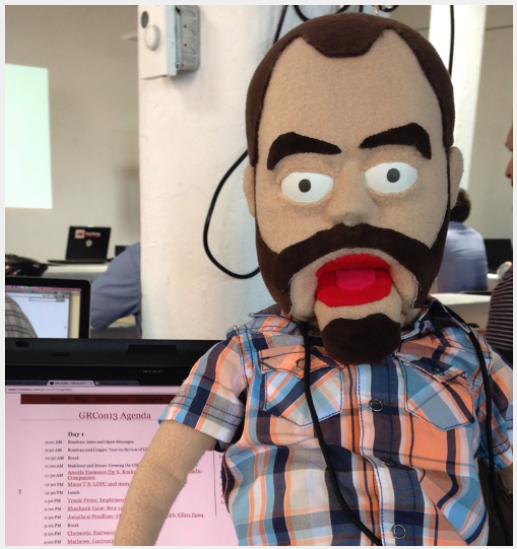
(tom@trondeau.com)

2014-08-13

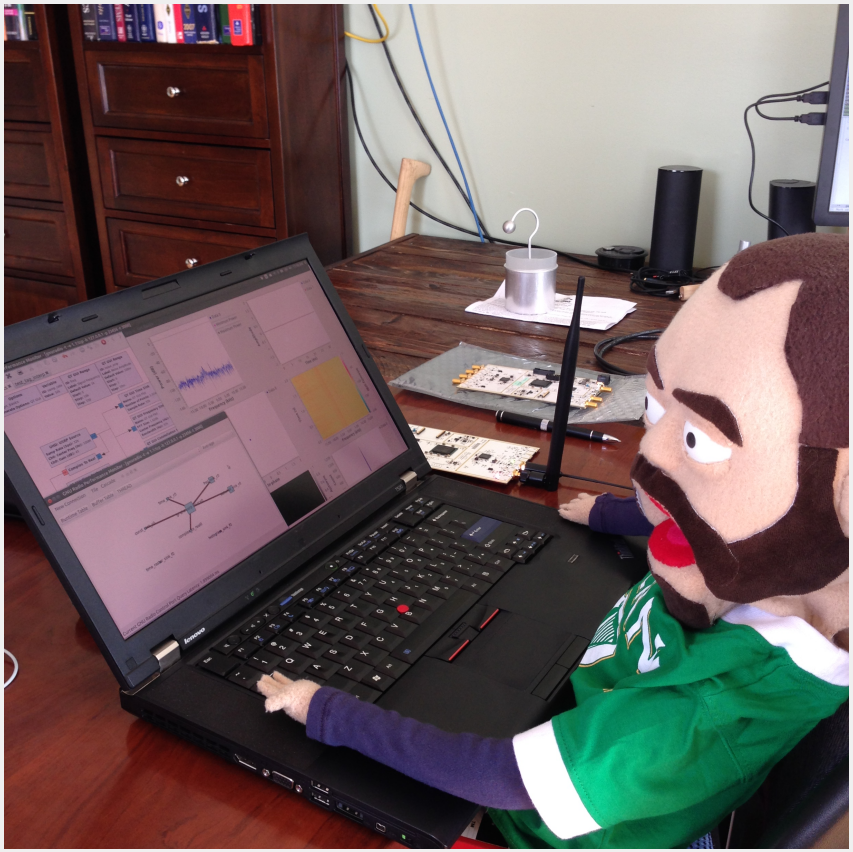
Intoduction

Tom Rondeau

- Maintainer and lead developer for GNU Radio
- Do you see this being a problem? I don't know anything about how schools do things. Consults through Rondeau Research
- Visiting Researcher at UPenn
  - (working with Jonathan Smith)
- www.trondeau.com
- gnuradio.org



What is SDR? Or is it SR?



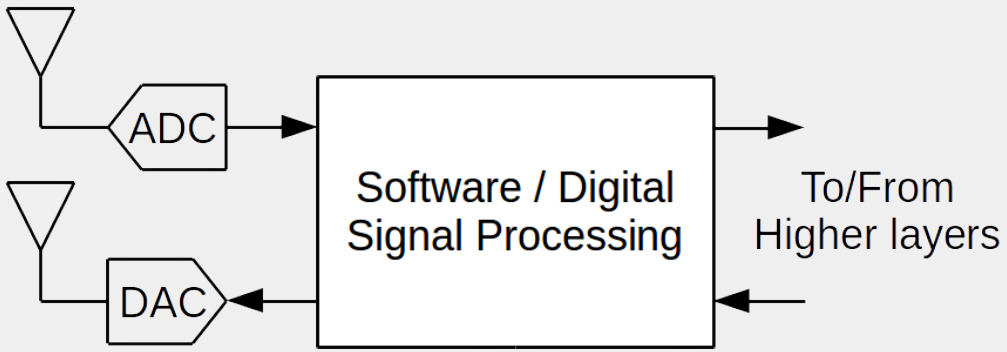
SDR Basics

What does SDR do for us?

- flexibility
- visibility
- rapid research/development/prototyping
- Improve our math and algorithms
- Future-proof?
- Science applications

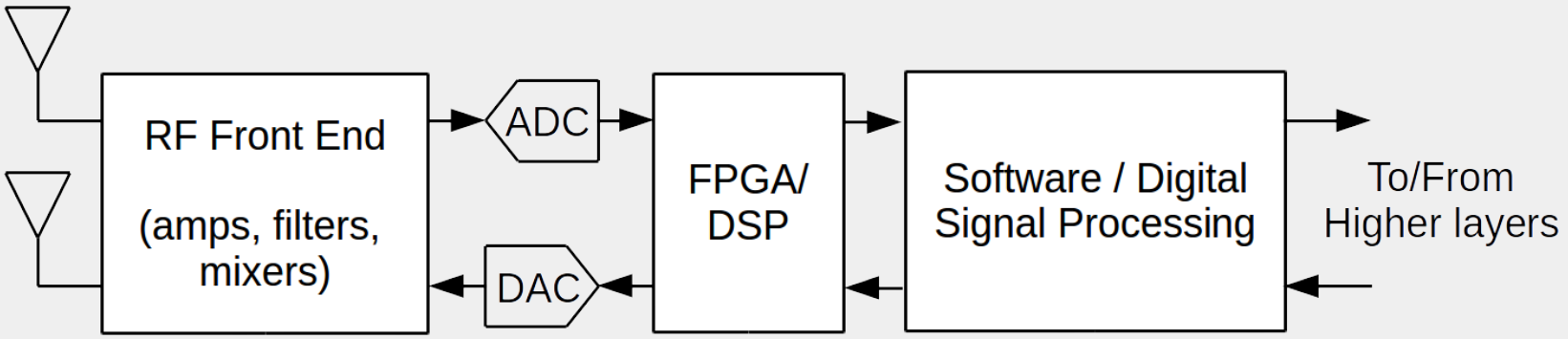
Defining SDR

Is this SDR?



Defining SDR

How about this?



Defining SDR

Does the processing domain matter?

- FPGAs through General Purpose Processors.
- How do we work with DSPs?
- Are applications-specific processing units (APUs) necessary/useful?
- Hardware co-processors:
  - FFTs
  - Viterbi / Turbo decoders
- Graphics Processing Units (GPUs)?

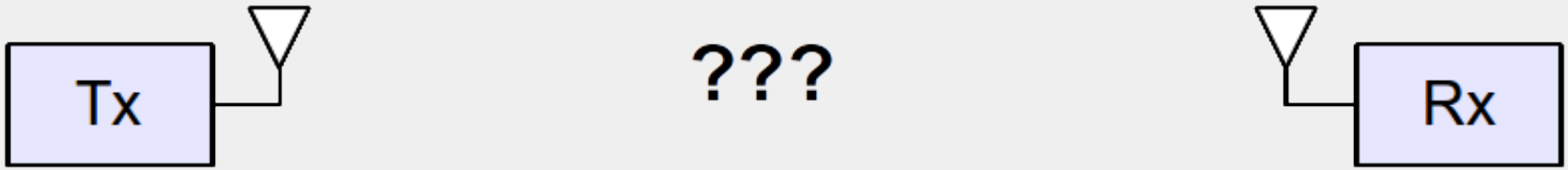
Defining SDR

Does it really matter?

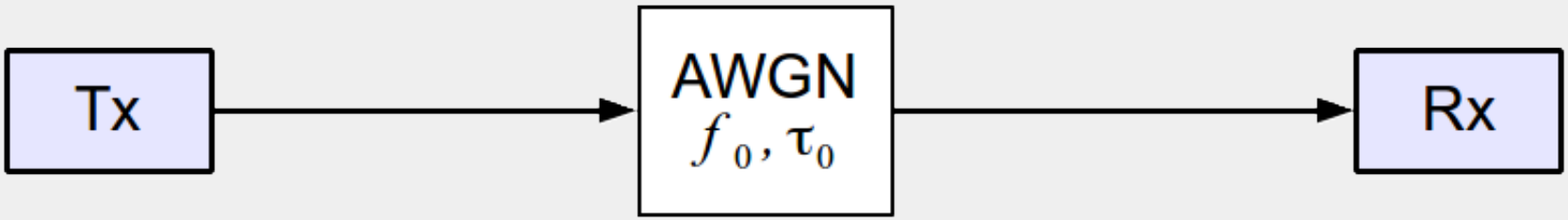
- What's the job you need to accomplish?
- What are the restrictions (size/weight/power/form factor/environment)?
- What's possible and what, if possible, is available?
- What kind of performance conditions do we require?

→ As hard as we've tried, we can't abstract away the RF layers.
→ Frequencies aren't "fungible".

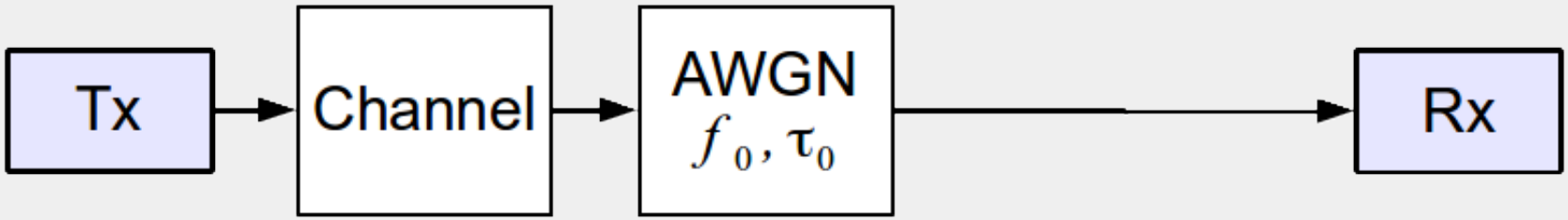
Using SDR



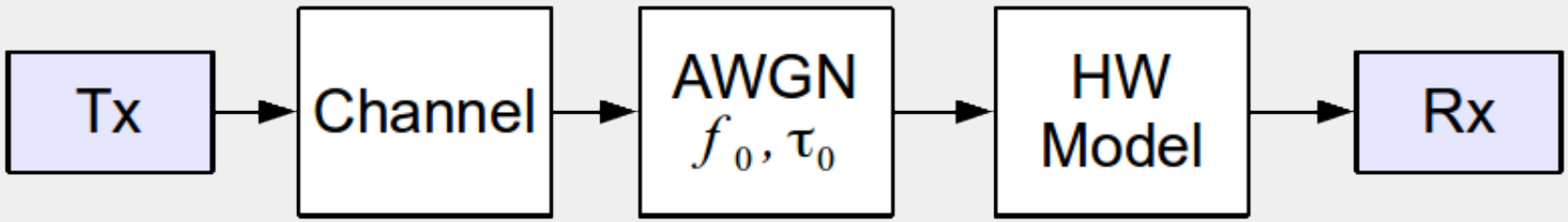
Using SDR



Using SDR



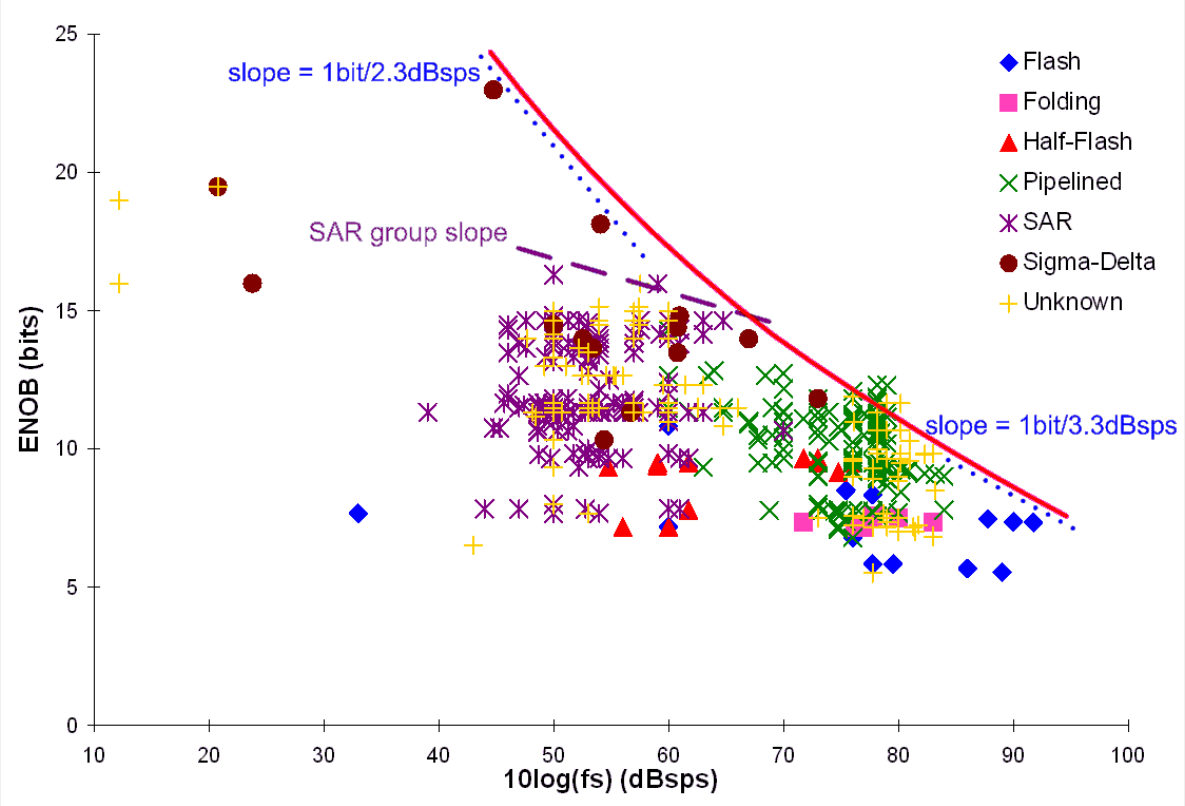
Using SDR



Using SDR

Analog to Digital Converters

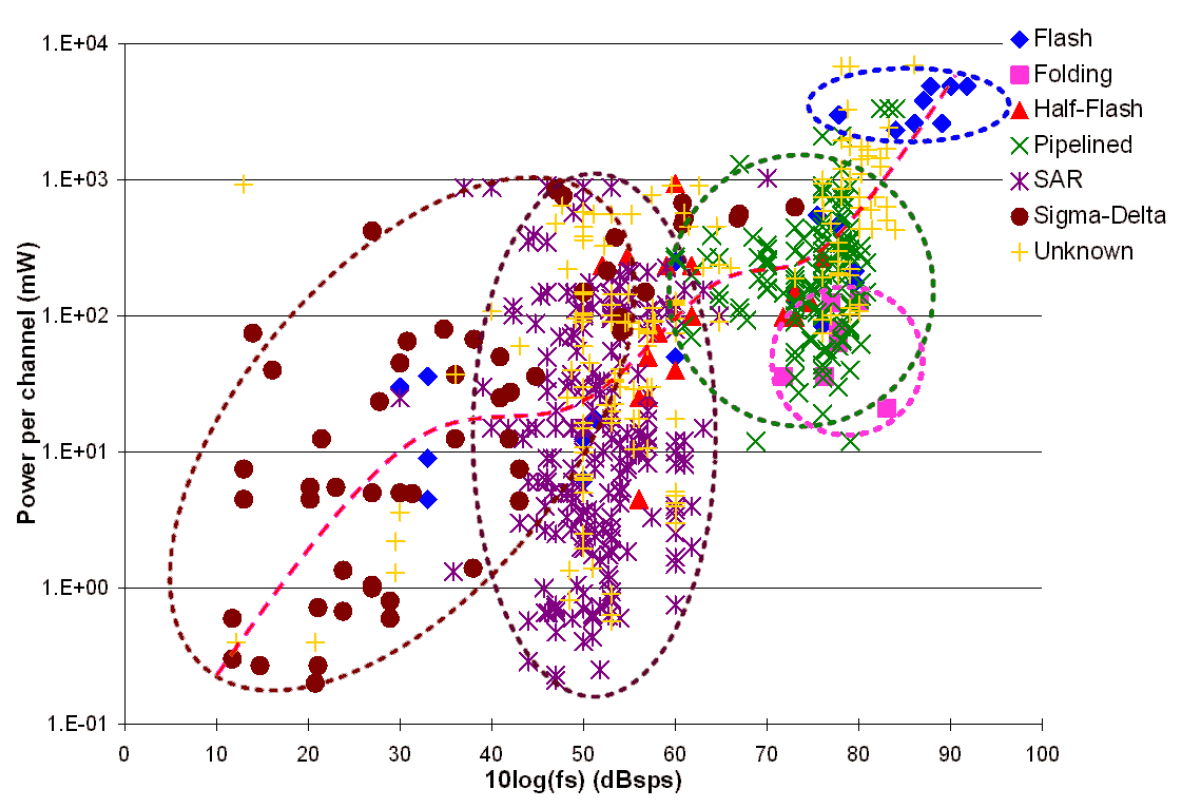
Effective Number of Bits vs. Sampling Rate (log x scale)



B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, Analog-to-Digital Converters: Past, Present, and Future, IEEE Signal Processing Magazine, pp. 69-77, Nov. 2005.

Analog to Digital Converters

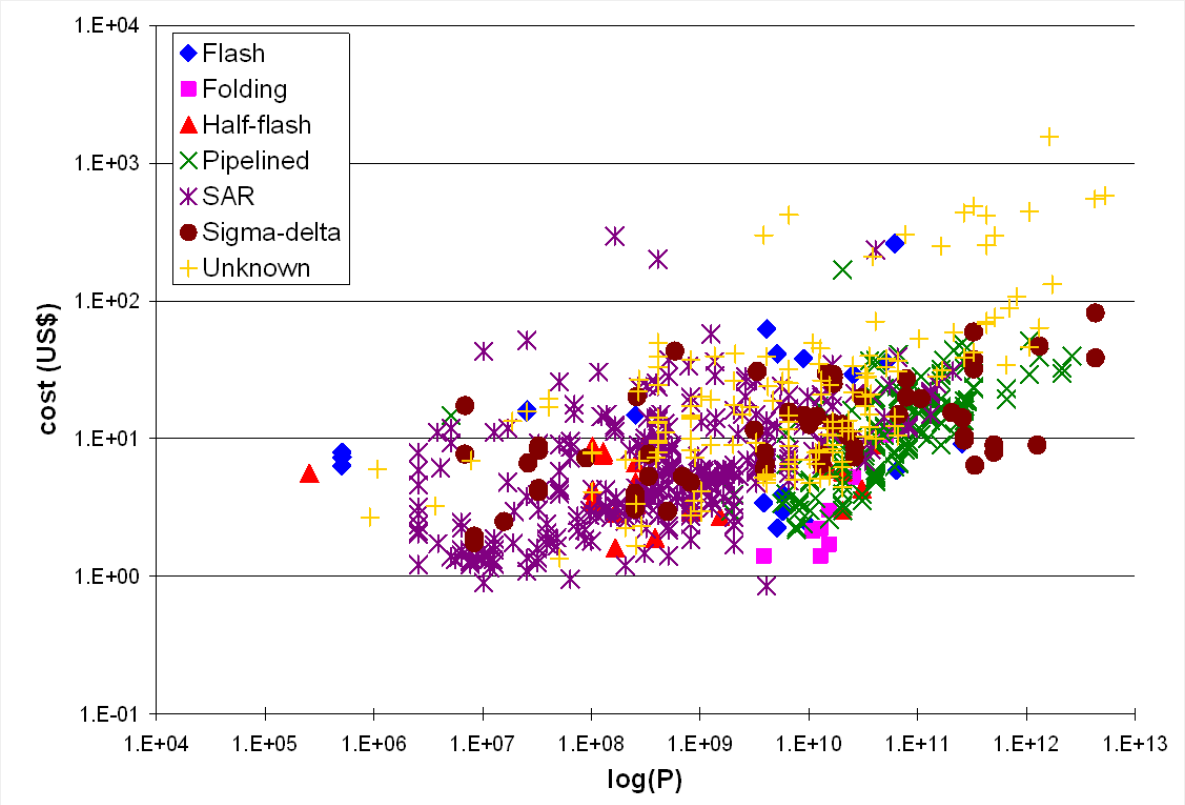
Power vs. Sampling Rate (log-log scale)



B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, Analog-to-Digital Converters: Past, Present, and Future, IEEE Signal Processing Magazine, pp. 69-77, Nov. 2005.

Analog to Digital Converters

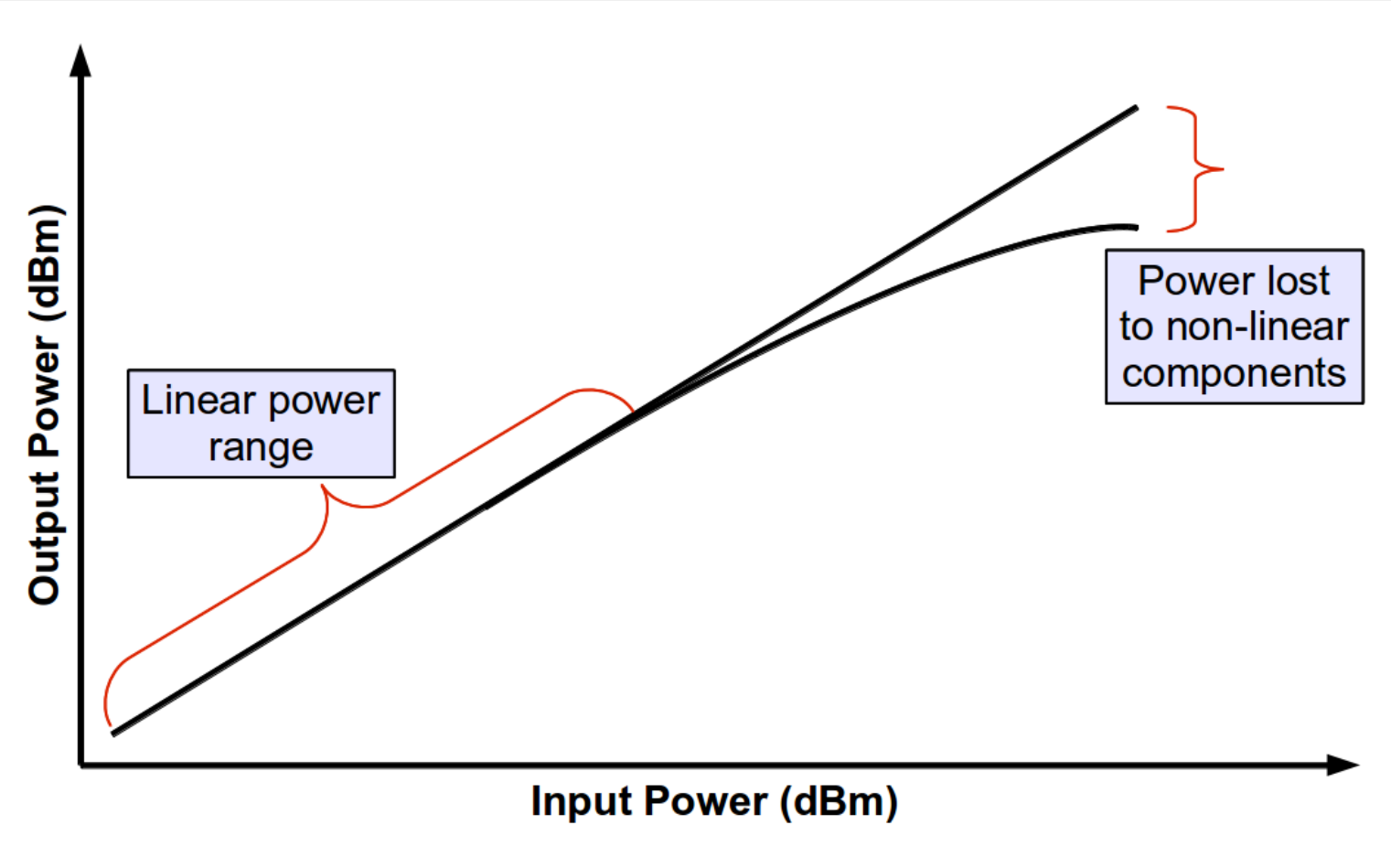
Cost per $P = 2^B \cdot f_s$ factor (log-log scale)



B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, Analog-to-Digital Converters: Past, Present, and Future, IEEE Signal Processing Magazine, pp. 69-77, Nov. 2005.

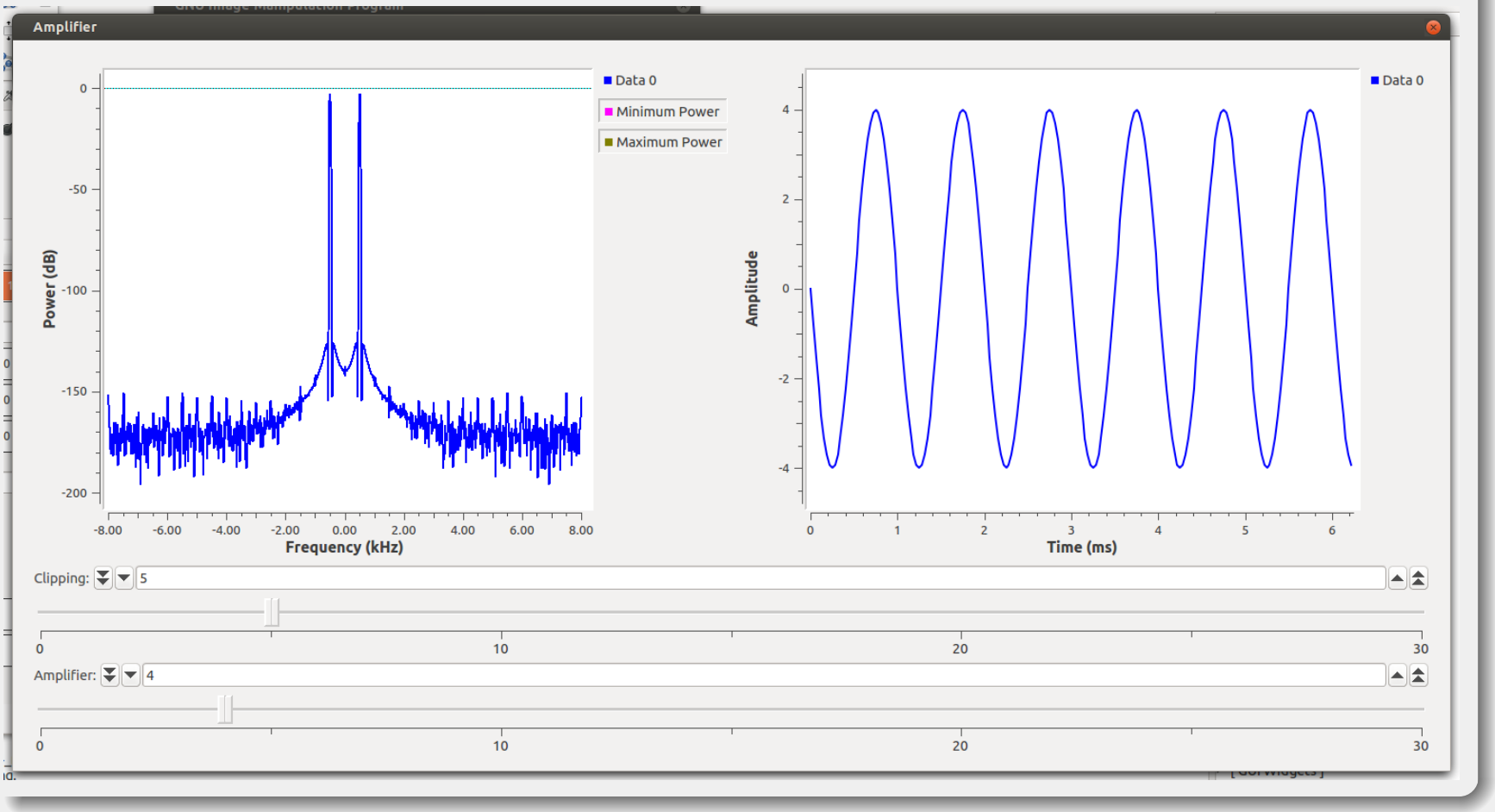
Analog Non-linearities

Linear and non-linear range of an amplifier



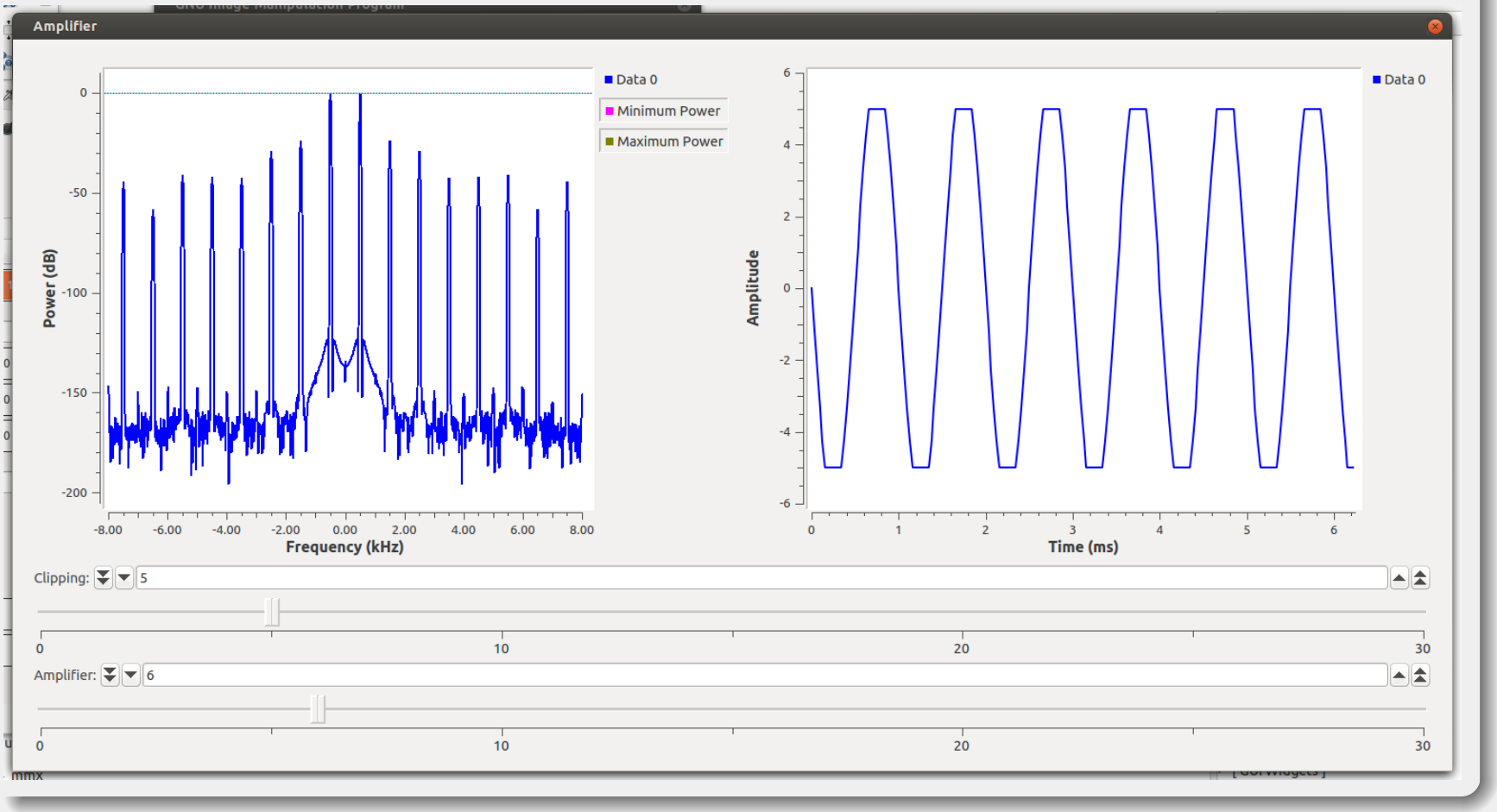
Analog Non-linearities

Clipping: Clean input sine wave



Analog Non-linearities

Clipping: Clipped sine wave



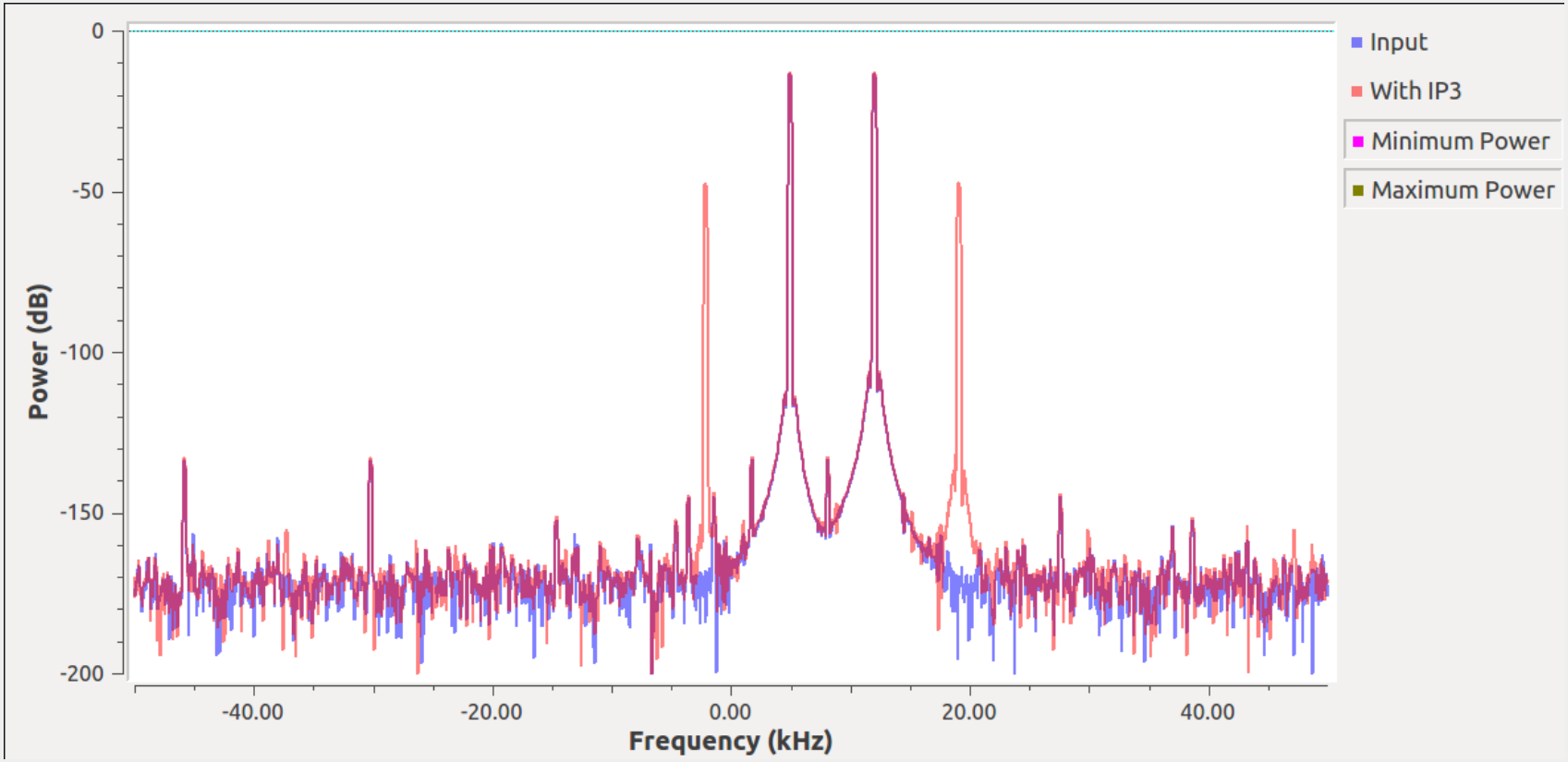
Non-linearity

- Output not a multiple of the input
  - Transfer function depends on amplitude
- Primary mechanism in semiconductor amps is clipping behavior as you approach maximum output
- $V_{out}=k_1 V_{in} + k_2 V_{in}^2 + k_3 V_{in}^3 + ...$
  - $cos^2 x = \frac{1+cos2x}{2}$
  - $cos^3 x = cos(3x) + 3cosxsin^2 x$
  - Output contains frequencies not in the input (harmonics and mixing products)
- Not just amplifiers (mixers, capacitors, inductors, even connectors)
- More complex models
  - Volterra Series
  - AM-AM and AM-PM

Third order non-linearity

- Third order the most important
- Typically modeled with third order intercept (IP3, IIP3, OIP3)
  - Intercept point is the [extrapolated] point at which intermod products would equal desired products
  - Typically ~10dB above P1dB
  - Don't actually operate at that point!
- $P_{IMD3} = 3P_{signal} - 2IP_3$
  - IMD3 products increase 3x as fast as input
  - IMD products appear at $2f_1 \pm f_2$, $2f_2 \pm f_1$, $3f_1$, $3f_2$

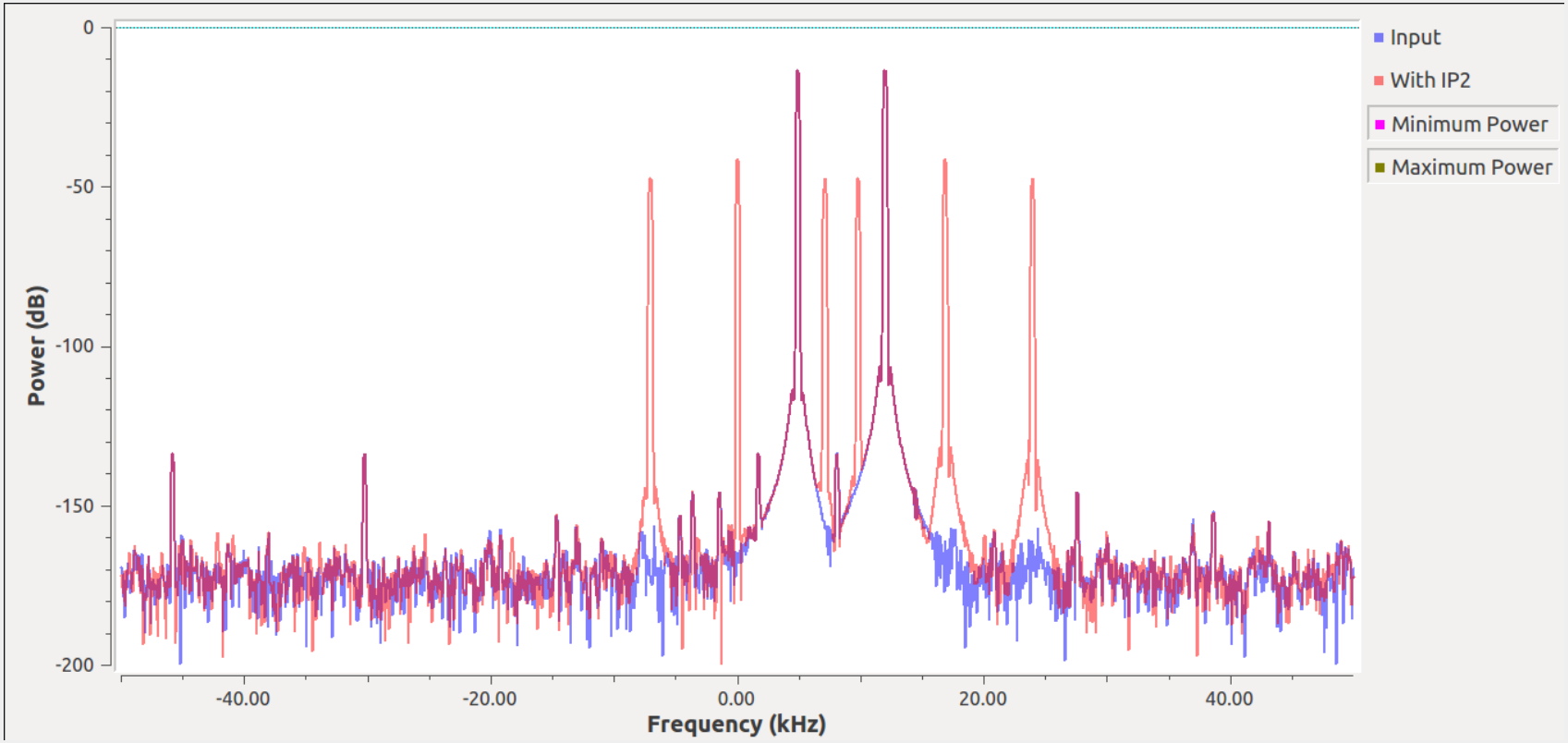
IP3 of Non-Harmonically Related Signals



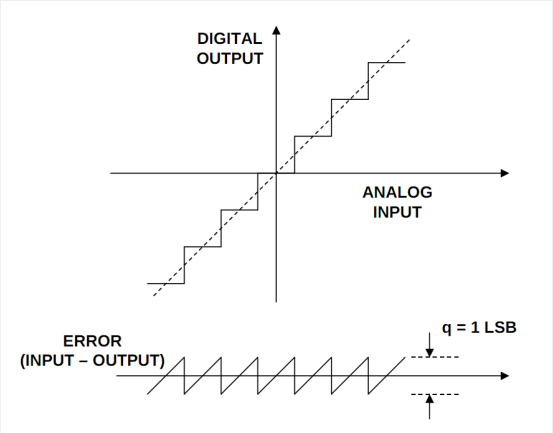
Second order non-linearity

- 2nd order products fall at DC, $f_1 \pm f_2, 2f_1, 2f_2$
- DC IMD2 product often mistaken for DC offset
- Only a problem in certain situations
  - Band of interest is greater than 1 octave
  - Band of interest includes DC
  - Direct Conversion receivers

IP2 of Non-Harmonically Related Signals



Quantization

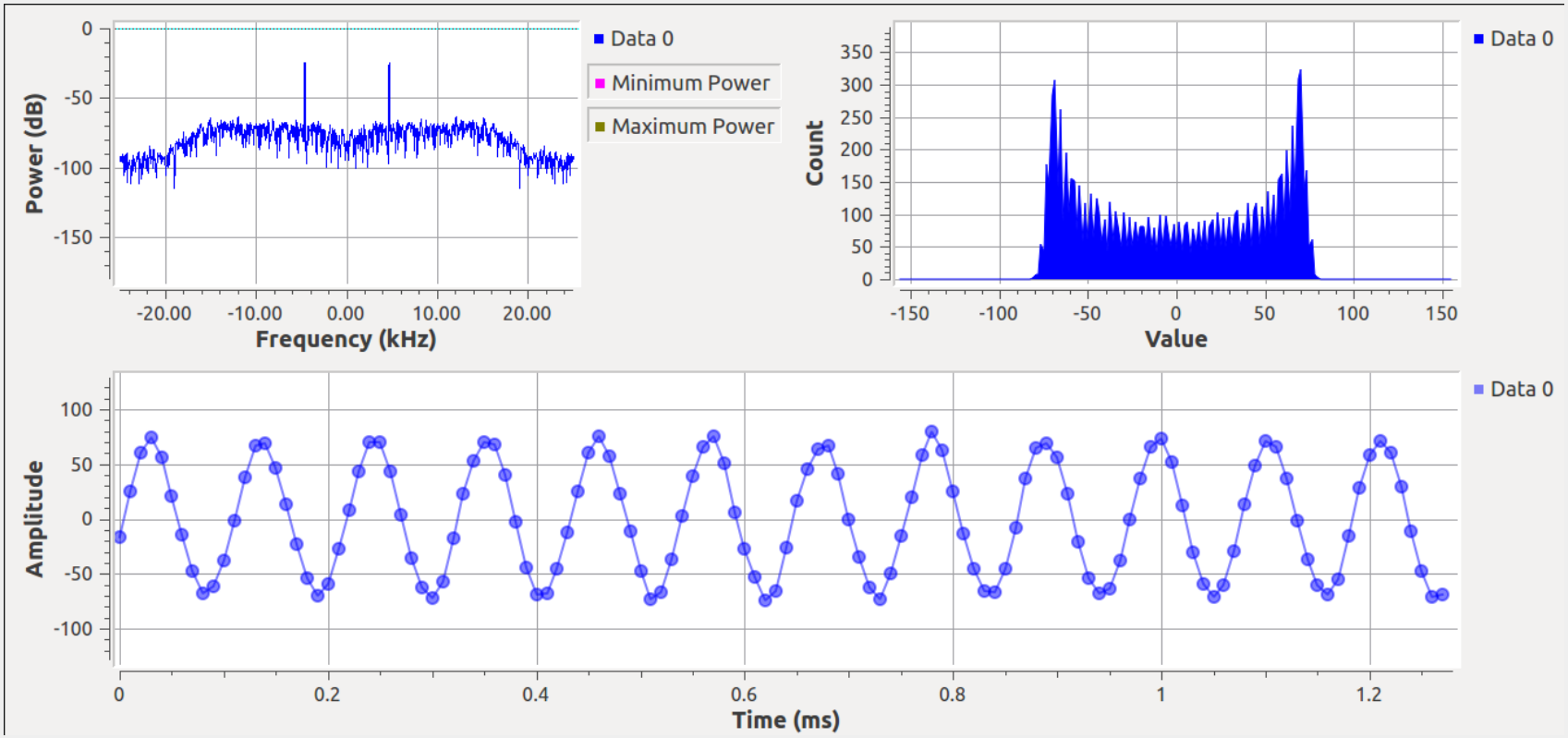


- Not to be confused with discretization (i.e. time steps)
- Inherent in digital systems
  - Finite bit widths in ADC, DAC
  - Costs of digital processing, storage, transmission
    - Cost of a Multiply operation is proportional to bits^2
  - Even floating point numbers are quantized

Quantization, cont'd

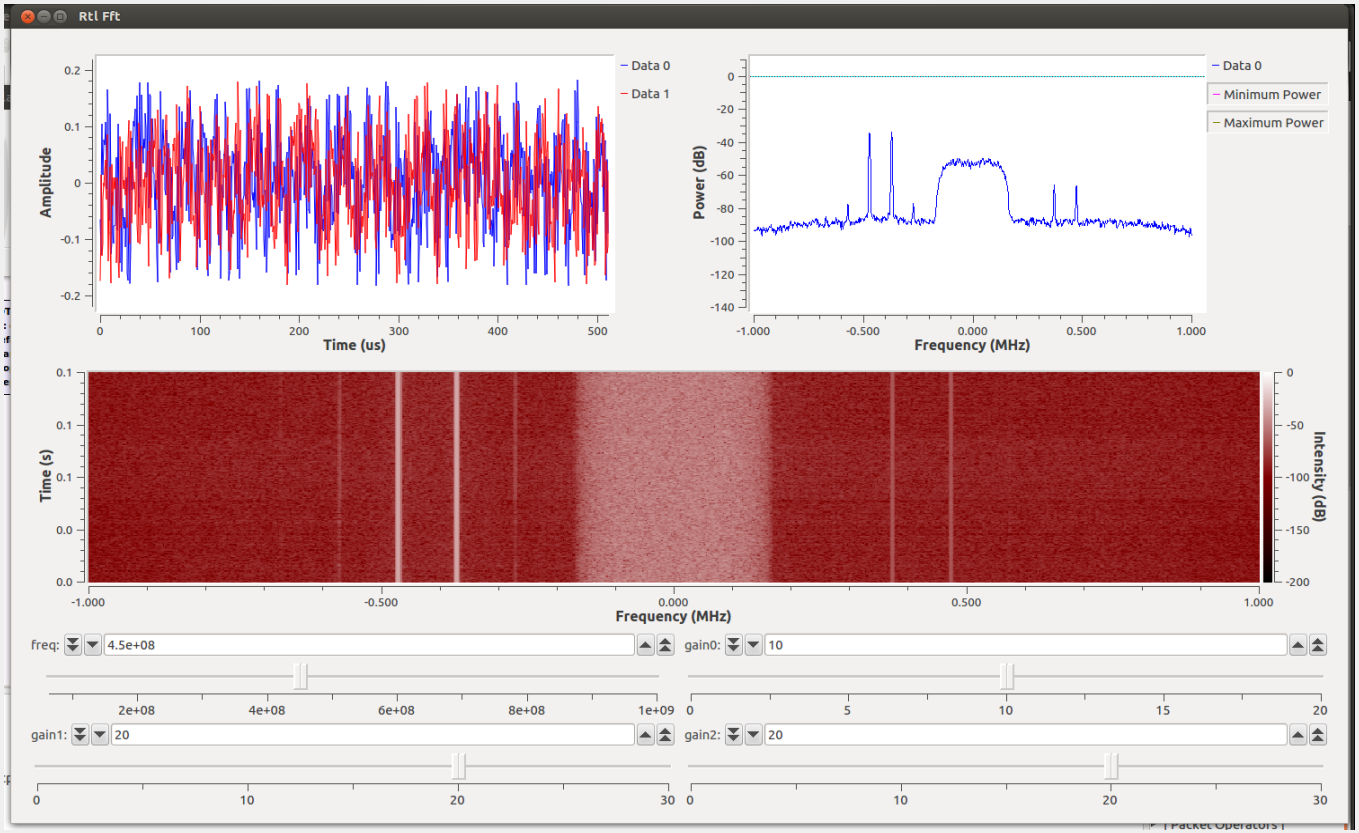
- Quantization results in noise
  - Often modeled as AWGN
    - Beware of correlated quantization noise ($f/f_s \backsimeq M/N$)
  - $SNR = 6.02N + 1.76dB$
- Non-ideal ADC/DAC behavior causes similar problems to correlated noise

Quantization Demo



Receiver Overloading and IQ Imbalance

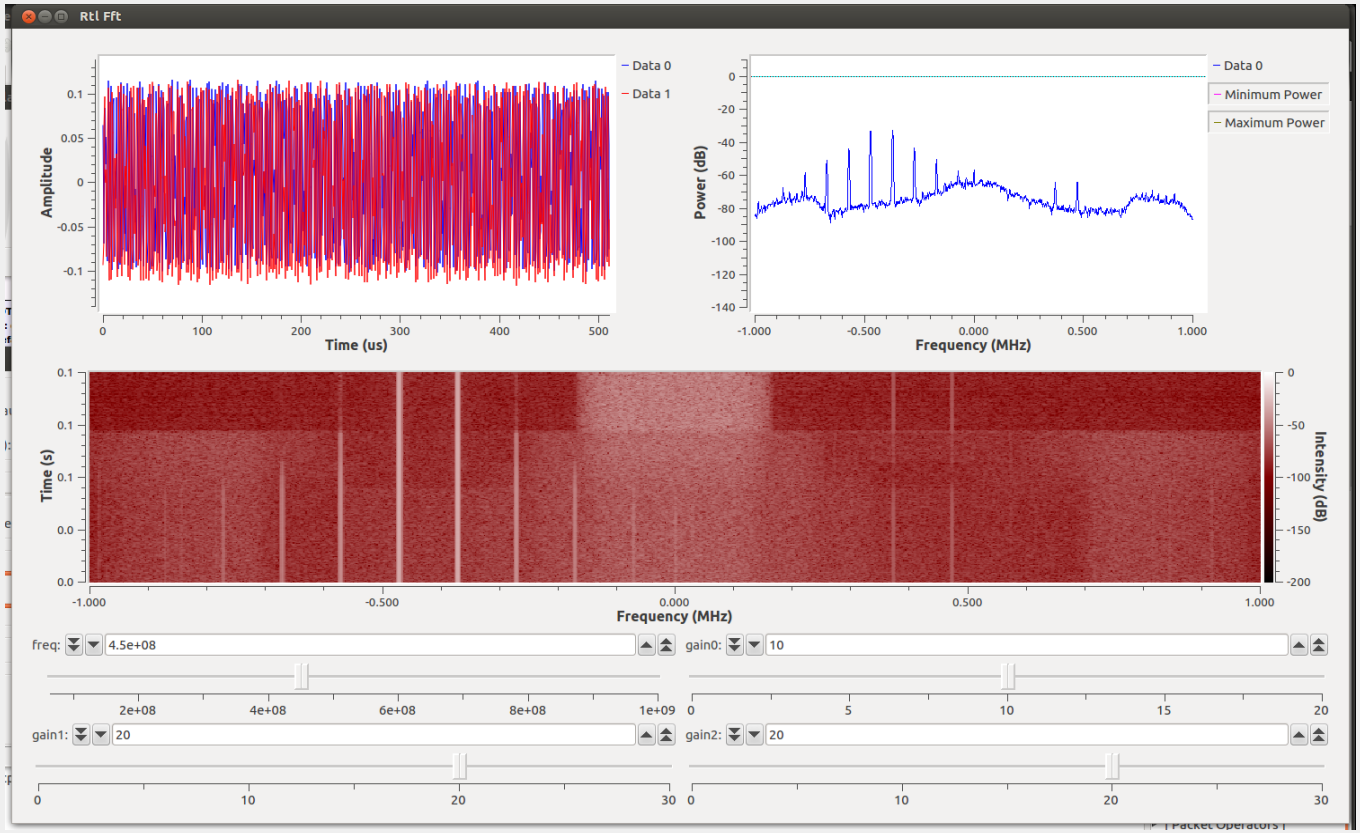
Rx'd PSK with interference tones: within receiver's dynamic range



- Images on the right due to IQ imbalance in the receiver.

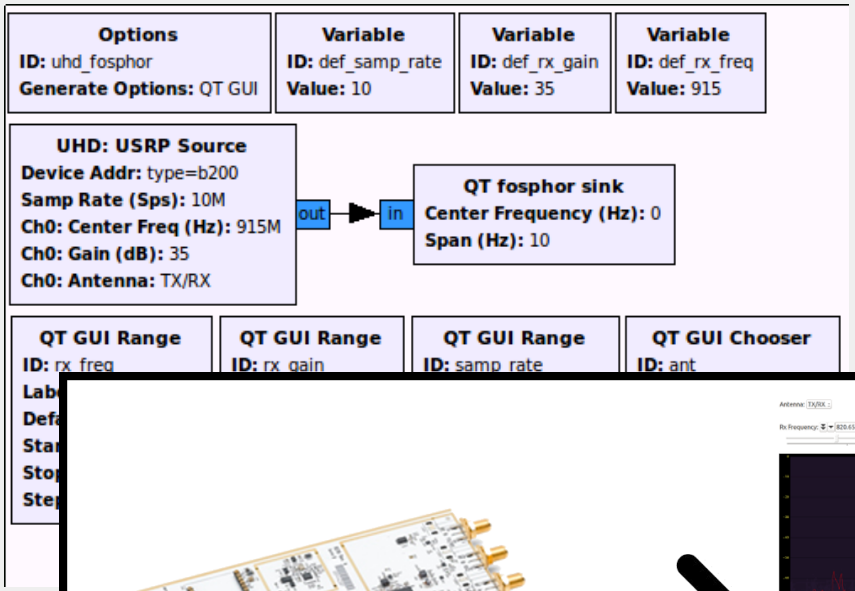
Receiver Overloading and IQ Imbalance

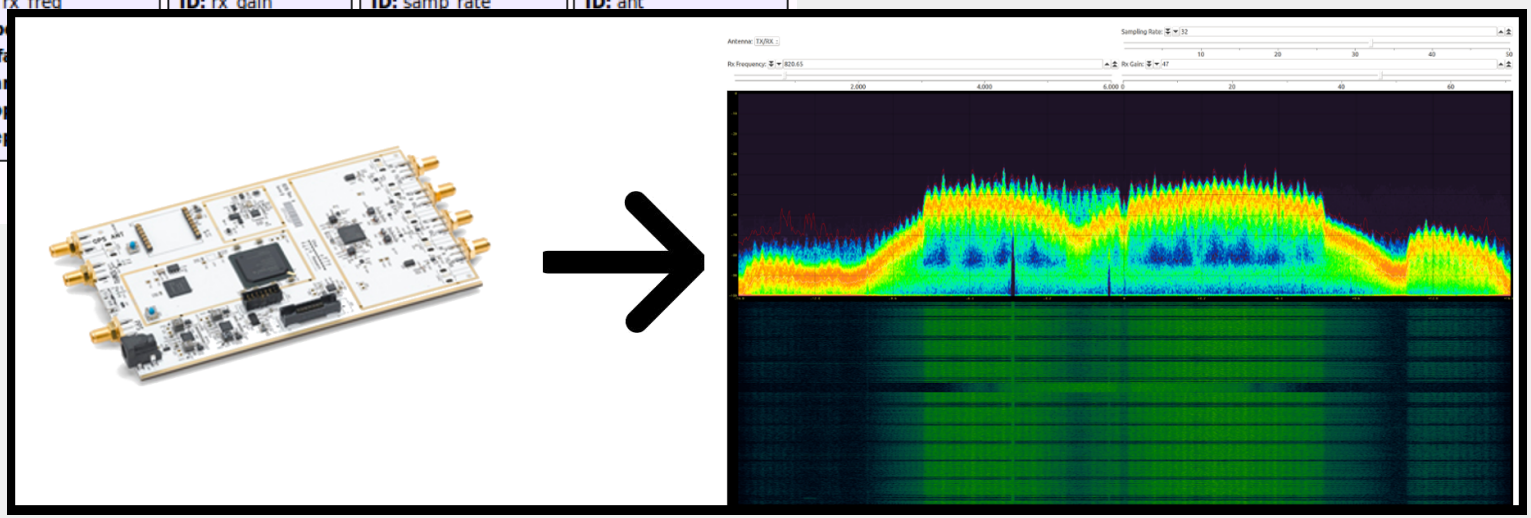
Rx'd PSK with interference tones: beyond receiver's dynamic range



- Increased power completely distorts the received signal.

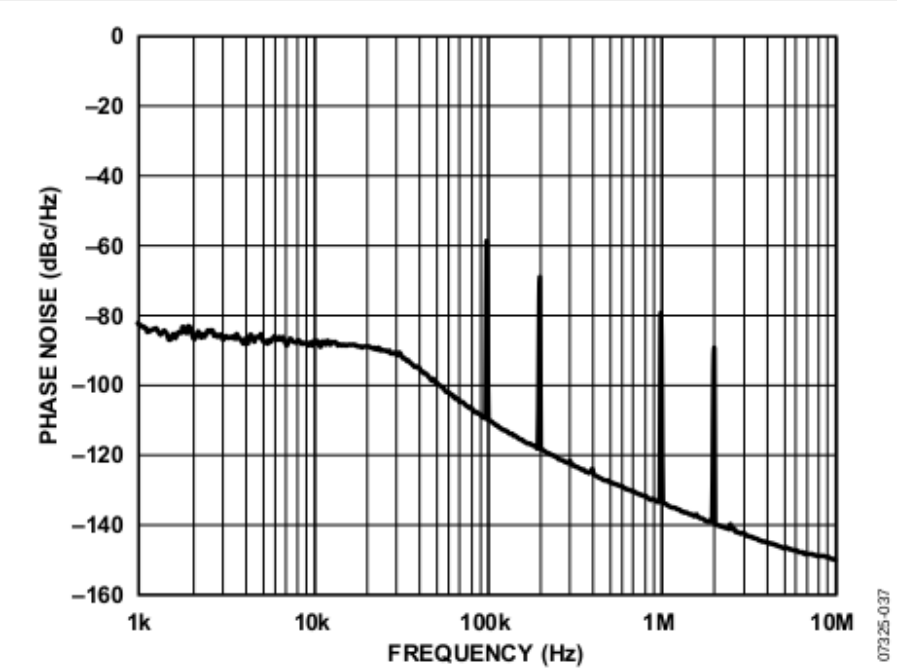
Poor Front-end Filtering: Wifi at 820 MHz?





Phase Noise

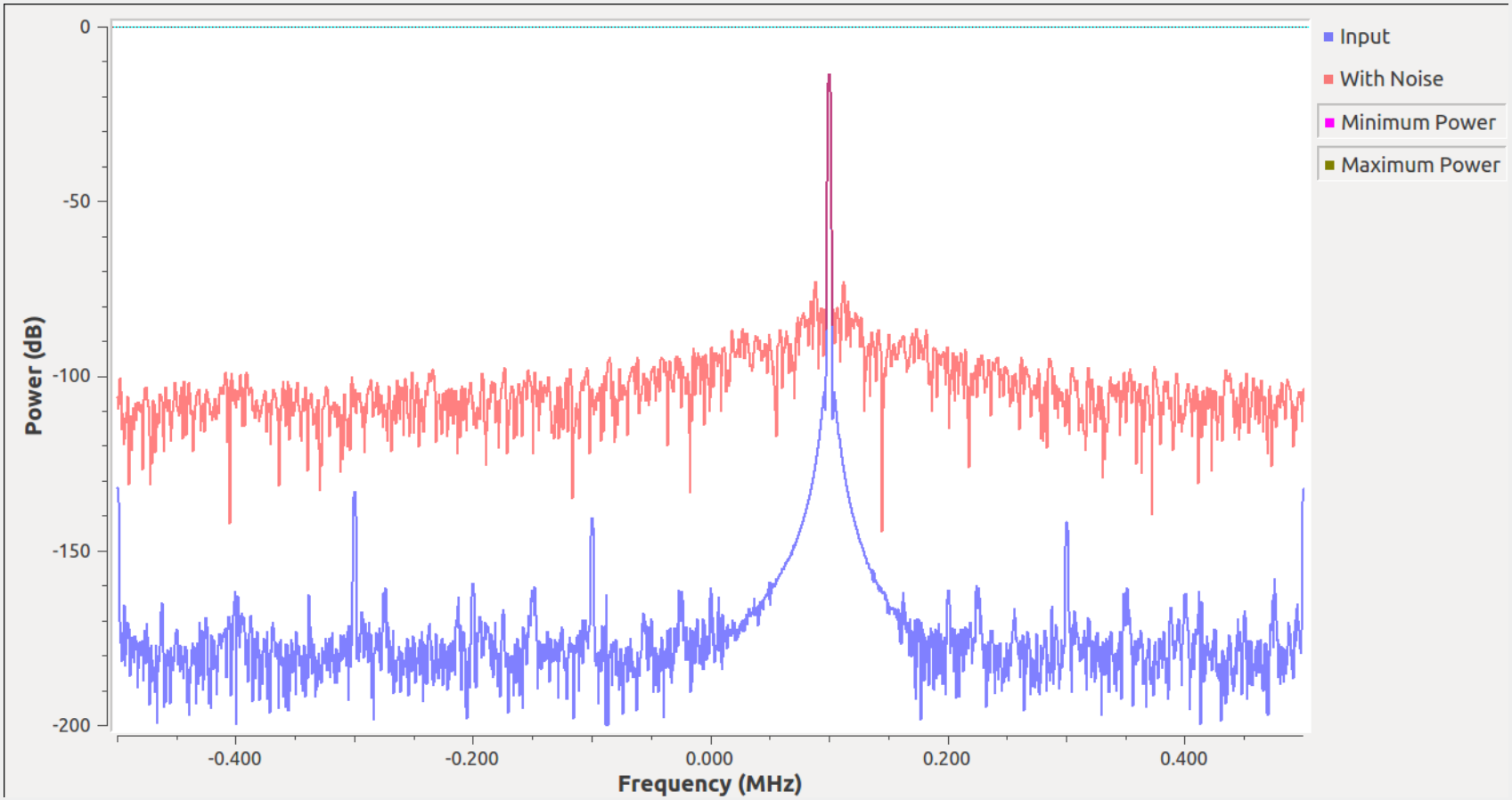
- Random phase perturbations on an oscillator
- Specified as dBc/Hz at an offset from carrier
  - i.e. -100dBc/Hz at 100kHz offset
- Modeled by the Leeson phase noise equation
- Spurs are a related phenomenon with similar symptoms



Phase Noise, cont'd

- Always causes self noise
  - increasing signal doesn't help
- -100dBc/Hz doesn't sound like much
  - Over a 10 MHz BW signal that equates to -30dBc
    - No QAM 256 for you!
- Total integrated phase noise often specified
  - I.e. 1.5 degrees RMS in a 20kHz to 80 MHz BW
- On TX causes adjacent channel emissions, broadband noise floor
- On RX mixes strong adjacent signals onto desired signal

Phase Noise Simulation



Architecture Specific

- DC Offset
- IQ Balance

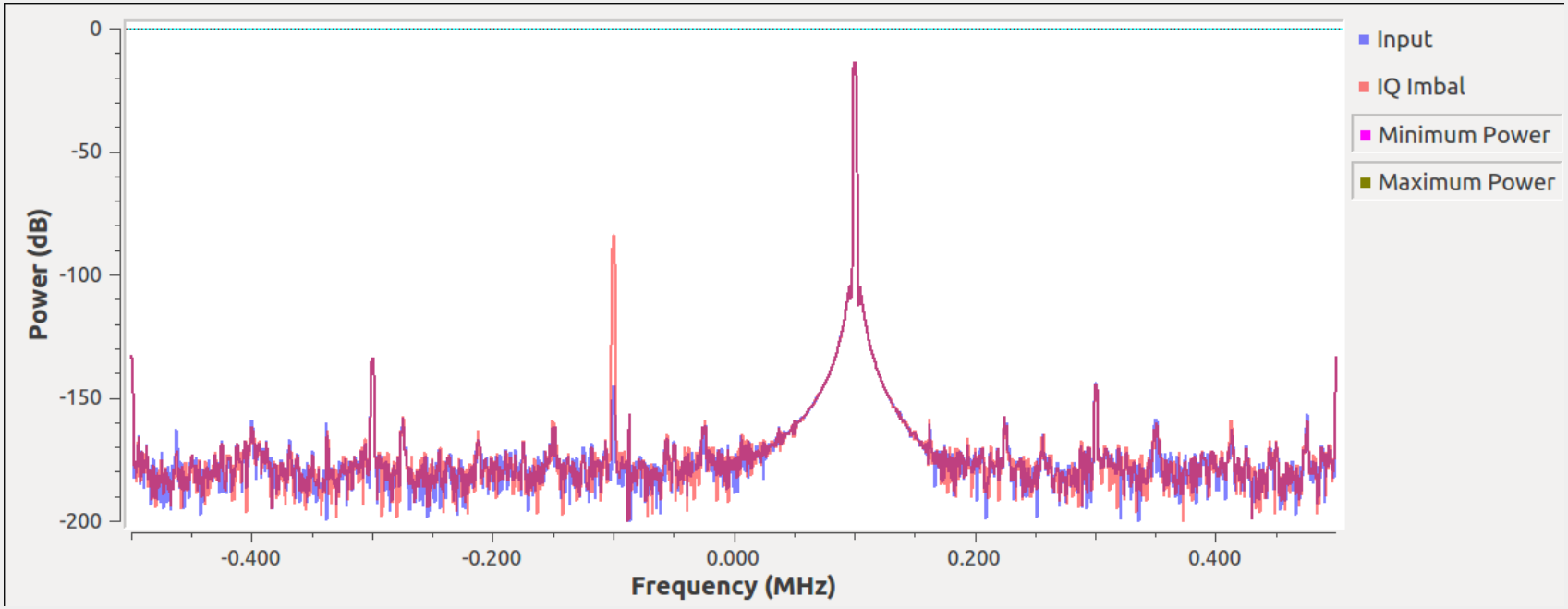
DC Offset

- Causes
  - Component mismatch
  - LO leakage
  - 2nd order distortion
- Varies with time, temp, frequency, voltage, moon phase, etc.
- Produces self interference
- Remedies
  - Ignore DC (use low-IF or ignore DC bin in OFDM)
  - AC-couple
  - Highpass filter (receiver only)
  - Estimate and subtract in either analog or digital domains
    - must be done at true baseband
    - much easier on the receiver

IQ Imbalance

- Magnitude imbalance caused by gain mismatch between paths
- Phase imbalance caused by
  - imperfect 90 degree phase shift in LO
  - mismatched phase or group delay between I and Q paths
- Varies with time, temp, frequency, voltage, moon phase, etc.
- Effects
  - Self interference
  - Out of channel leakage on transmit
  - Susceptibility to out of channel interference on receive
  - Inherently non-LTI since it generates new frequencies

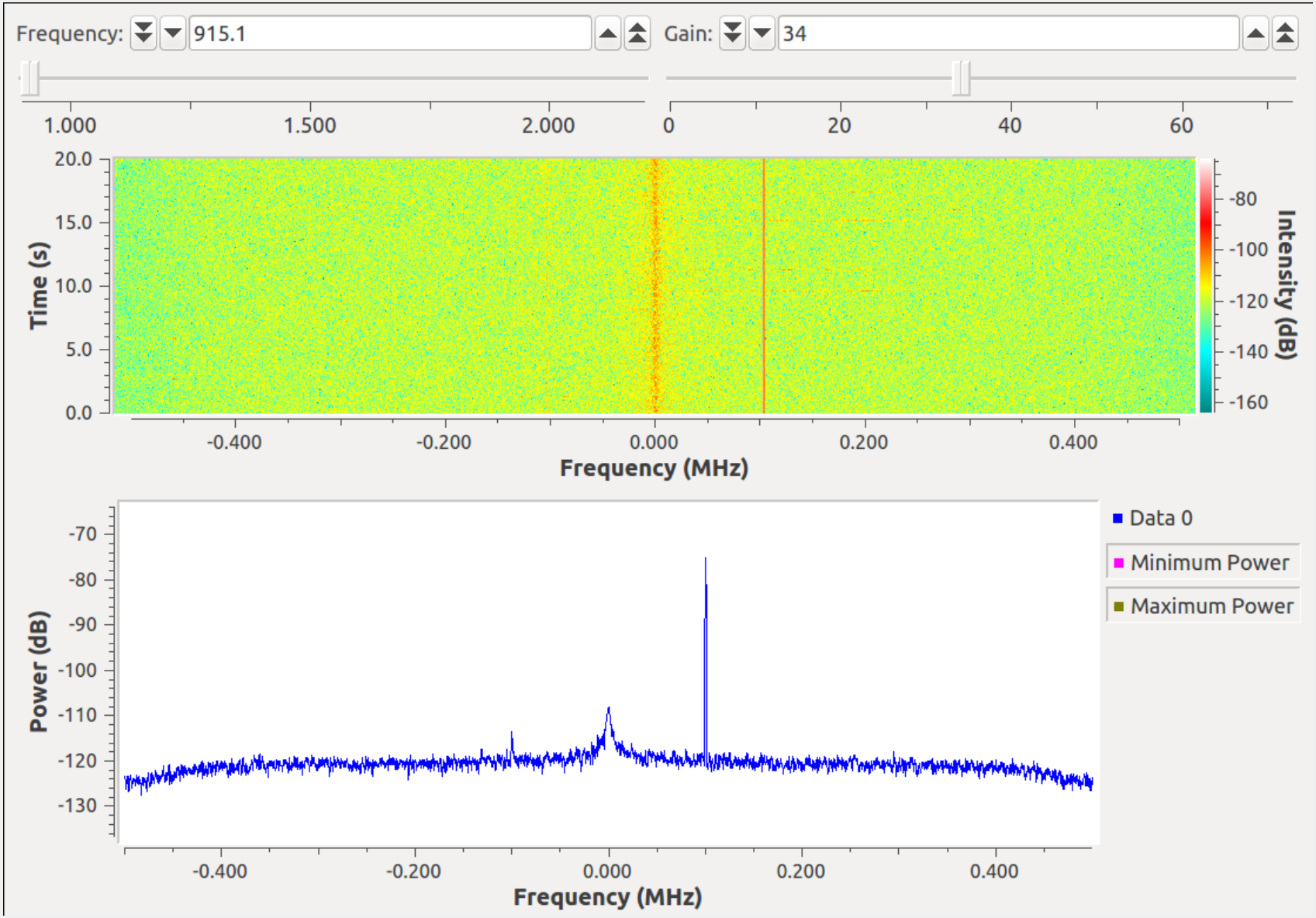
IQ Imbalance Simulation



Fixing IQ Imbalance

- Remedy is
  - estimate the relative I and Q magnitude error and scale appropriately
  - estimate the relative phase and rotate components appropriately
  - Must be done at true baseband
  - Much easier on the receiver
- May be baseband frequency selective
  - Must scale magnitude and phase differently for different frequencies
    - Requires multi-tap filter

DC Offset and IQ Imbalance in Action



Introduction to GNU Radio

"He liked the GNU. They thought in a refreshingly different way."
- Terry Pratchett, *Going Postal*

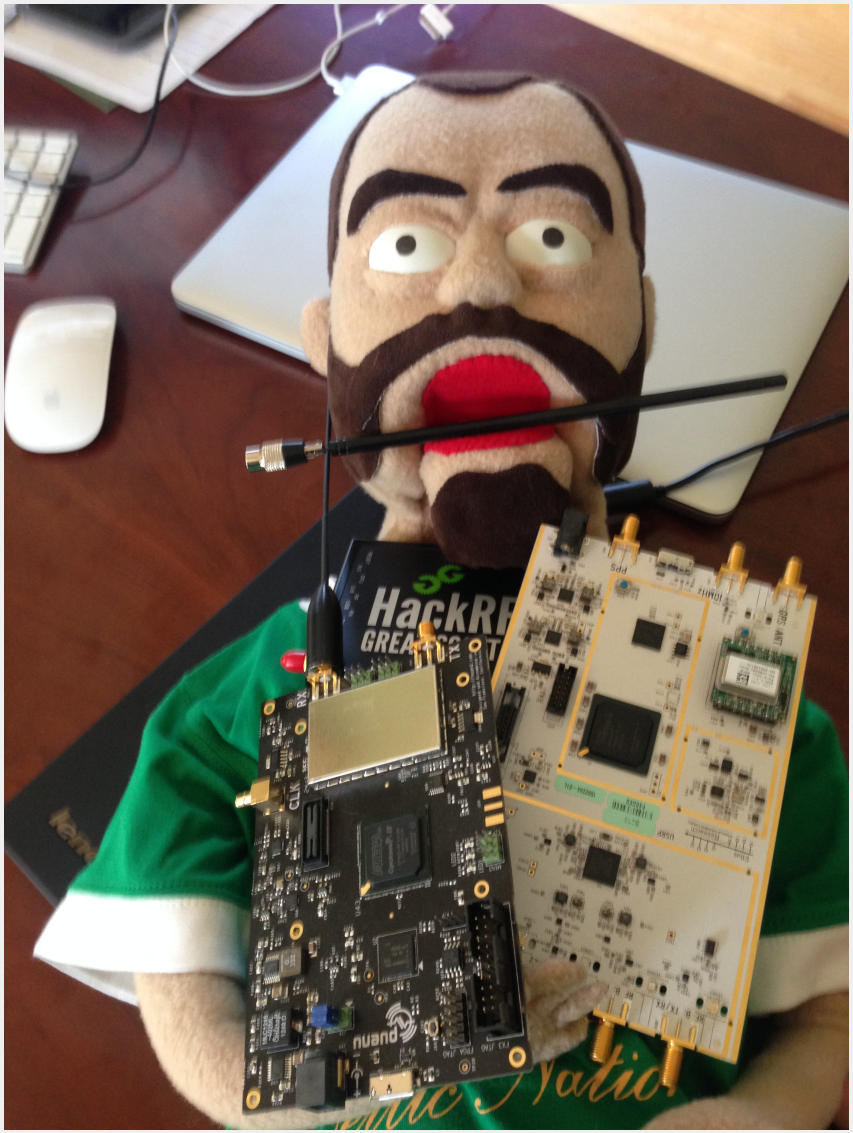
GNU Radio

- The GNU Radio Framework
- Library of DSP Blocks
  - Filters
  - Analog modulations
  - Digital Modulations
  - Visualization and interaction tools
  - etc, etc, etc...
- Hardware interfaces
- A community of experts/enthusiasts
- Growing list of 2nd party projects based on GR

GNU Radio Resources

- Website:
  - gnuradio.org
- Manual and documentation:
  - gnuradio.org/docs/doxygen
- Mailing list:
  - gnuradio.org/redmine/projects/gnuradio/wiki/MailingLists
- My blog/announcements:
  - www.trondeau.com
- GRCon Conferences:
  - www.trondeau.com/grcon14
- IRC: #gnuradio on chat.freenode.net

SDR Front Ends



General Overview

- Moves data from the analog domain to digital domain.
- Generally does the bare minimum amount of work
  - Tasks common to all signals:
    - up/down conversion of sampling rates
    - modulate signals to/from RF frequency
    - amplification, filtering
  - Often equipped with an FPGA and can be programmed for other things
- Connects to software host system over a bus:
  - USB, Ethernet, PCIe, etc.
  - Trade-offs in bandwidth, latency, complexity, and cost.

Standard RFE Ratings

- Typically stated by manufacturer:
  - Frequency range of operation (DC to Daylight)
  - Number of bits in Rx and Tx converters (resolution)
  - Dynamic range (variable gain + ADC resolution)
  - Instantaneous bandwidth (converters/bus)
  - TX/RX duplexing (full/half)
  - Power source/requirements (USB bus powered?)
- Not typically mentioned
  - Phase noise
  - IQ Imbalance
  - Noise figure
  - 1-dB and/or 3-dB compression points

Transfer Rates

- Assume: 16-bits I&Q $\rightarrow$ 32 bits/sample over the wire
- We have numbers for different buses
  - Only expect to achieve some fraction of the real throughput
- Following lists are given as: "theoretical max [practical max]"
- Other issues come in to play with these, too, like latency.

Transfer Rates

- USB
  - 2.0 @ 480 Mbps → 15 MHz [8 MHz]
  - 3.0 @ 3.2 Gbps → 100 MHz [56 MHz]
  - Examples: USRP B100/B150, Funcube Dongle, RTL-SDR
- Ethernet
  - GigE @ 1 Gbps → 31.25 MHz [25 MHz, up to 30 observed]
  - 10 GigE @ 10 Gbps → 312.5 MHz [~120 - 200 MHz]
  - Examples: USRP N2x0, X3x0
- PCIe
  - Desktop: 200 MHz, 10 μs latency
  - Laptop (ExpressCard): 50 MHz, 10 μs latency
  - Examples: USRP X3x0

RTL-SDR (rtlsdr.org)

- Varies depending on the part
- Frequency range: ~25 MHz - ~2100 MHz (break from 1100-1250)
- Resolution: 8 bits
- Dynamic range: not given (not great, either)
- Instantaneous bandwidth: 2.4 MHz max
- TX/RX duplexing: Receive only
- USB 2.0, bus powered
- $15 - $30

USRP: B200 and B210 (ettus.com)

- Frequency range: 70 MHz - 6 GHz
- Resolution: 12-bit ADC/DAC
- Dynamic range: 78 dBc (SFDR)
- Instantaneous bandwidth: 56 MHz (practically 32 MHz)
- TX/RX duplexing: full duplex
  - B210 is dual channel for 2x2 MIMO
- USB 3.0
  - B200: bus powered
  - B210: requires external power for both channels
- $675 / $1100

USRP: N200 and N210 (ettus.com)

- Frequency range: 0 MHz - 6 GHz
  - Depends on daughterboard used
- Resolution: 14-bit ADC, 16-bit DAC
- Dynamic range: depends on daughterboard
- Instantaneous bandwidth: 25 MHz (50 MHz at 8-bit samples)
- TX/RX duplexing: full duplex
- Gigabit Ethernet
- External power required
- $1,515 / $1,717

USRP: X300 and X310 (ettus.com)

- Frequency range: 0 - 6 GHz
  - Depends on daughterboard
- Resolution: 14-bit ADC, 16-bit DAC
- Dynamic Range: depends on daughterboard
- Instantaneous bandwidth: 120 MHz (up to 200 MHz possible)
- TX/RX duplexing: full duplex
  - Support 2 daughterboards for 2-channel support
- PCIe x4, ExpressCard, or 10 GigE
- External power required
- $3,900 / $4,800

USRP: E100 and E110 (ettus.com)

- Frequency range: 0 - 6 GHz
  - Depends on daughterboard
- Resolution: 12-bit ADC, 14-bit DAC
- Dynamic Range: depends on daughterboard
- Instantaneous bandwidth: $<$ 8 MHz
- TX/RX duplexing: full duplex
- Embedded OMAP Overo processor (800 MHz ARM Cortex A8)
- Bus from FPGA to OMAP
- External power required
- \$1,313 / \$1,515

USRP: E300 (ettus.com)

- Frequency range: 70 MHz - 6 GHz
- Resolution: 12-bit ADC/DAC
- Dynamic Range: 78 dBc (same as B210?)
- Instantaneous bandwidth: unknown
- TX/RX duplexing: full duplex
- Embedded Xilinx Zynq-7000 (1 GHz ARM Cortex A9)
- Bus from FPGA to Zynq
- Battery powered
- $$$ (unknown)
- Form-factor: bulky cell phone
- Release: unknown – later this year (?)

Great Scott Gadgets: HackRF (greatscottgadgets.com)

- Frequency range: 10 MHz - 6 GHz
- Resolution: 8 bits
- Dynamic range: unknown
- Instantaneous bandwidth: 8 to 20 MHz
- TX/RX duplexing: half duplex
- USB 2.0, bus powered
- $299 (to ship in a month or so)

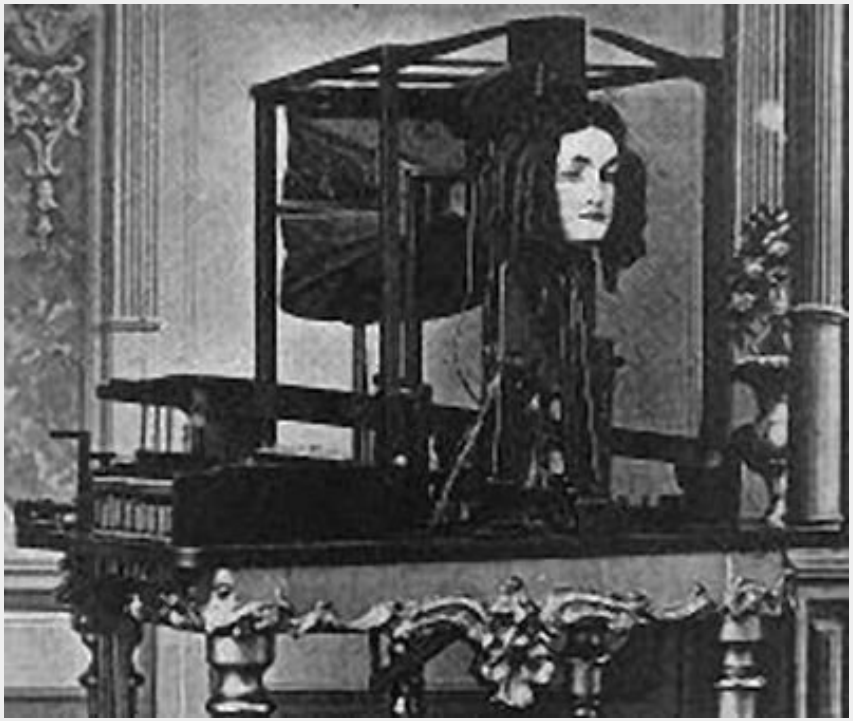
Nuand: BladeRF x40 and x115 (www.nuand.com)

- Frequency range: 300 MHz - 3.8 GHz
- Resolution: 12-bit ADC/DAC
- Dynamic range: unknown (claims to be excellent)
- Instantaneous bandwidth: 28 MHz
- TX/RX duplexing: full duplex
- USB 3.0, bus powered
- $420 / $640

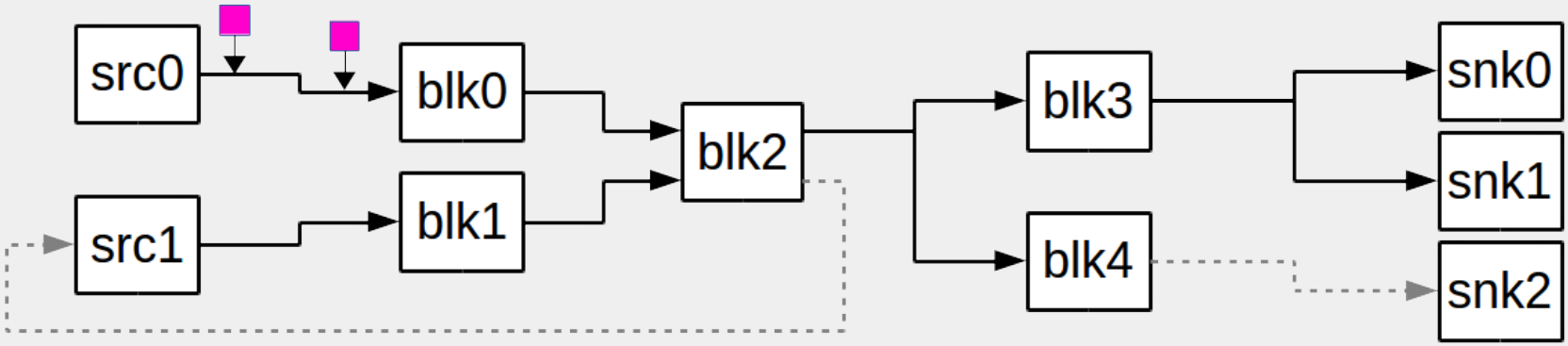
Nutaq PicoSDR (www.nutaq.com)

- Frequency range: 300 MHz - 3.8 GHz
- Resolution: unknown (12-bit ADC/DAC?)
- Dynamic range: unknown
- Instantaneous bandwidth: 1.5 - 28 MHz
- TX/RX duplexing: full duplex
  - Up to 4 channels
- 1 GigE and/or PCIe x4
- $$$? (unknown; need to get a quote)

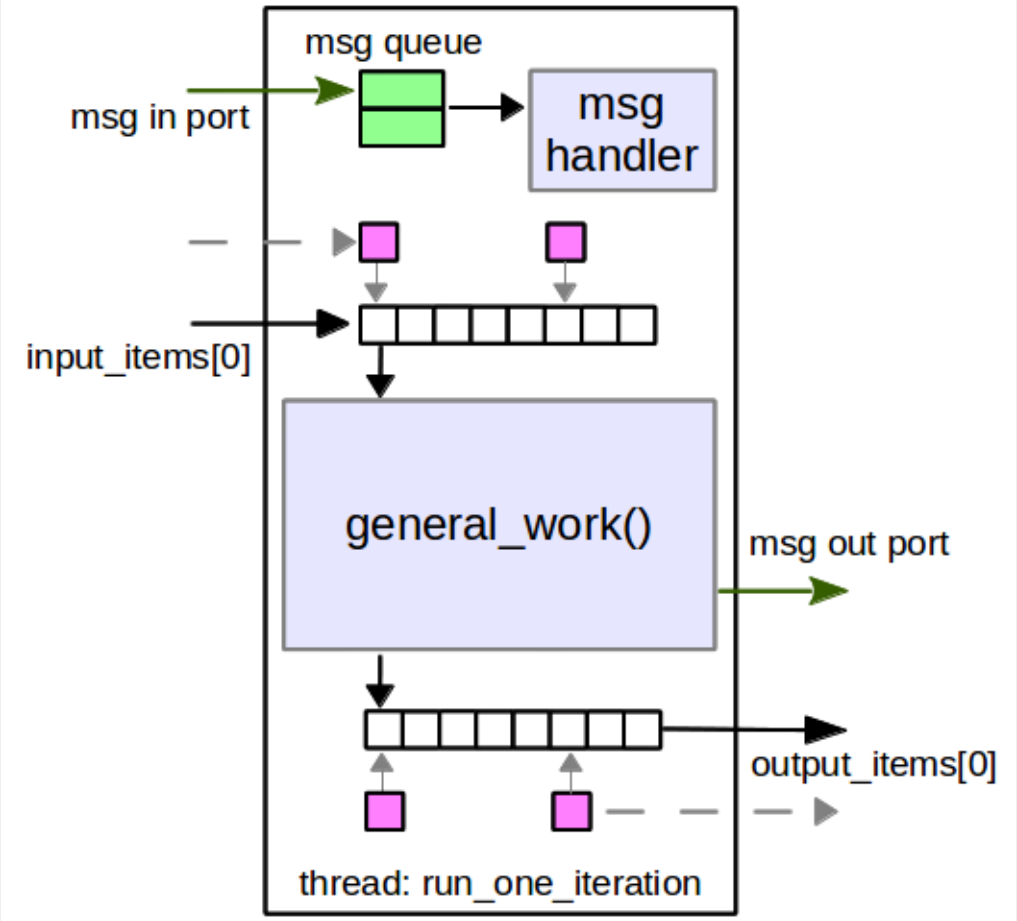
GNU Radio Data Streams



Flowgraph



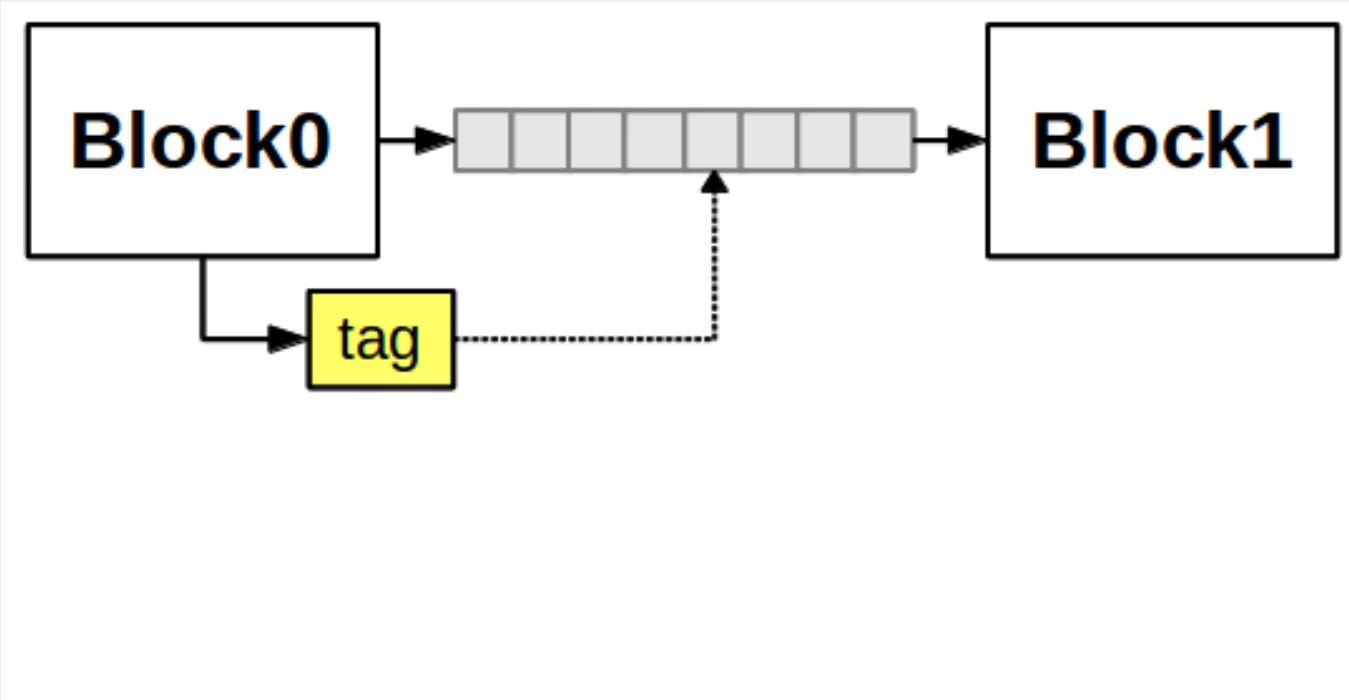
Block Model


msg queue
msg in port
msg handler
input_items[0]
general_work()
msg out port
output_items[0]
thread: run_one_iteration

Data Stream

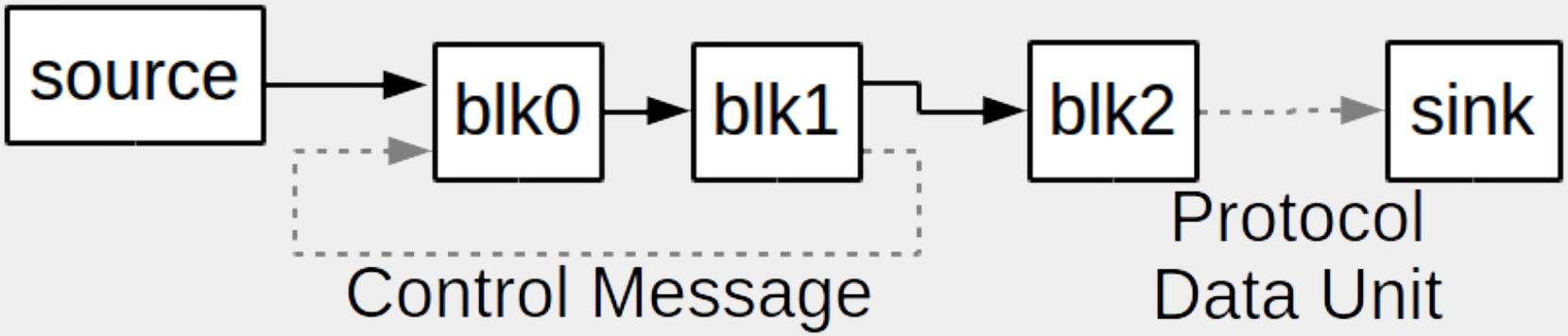
- Synchronous stream of data
- Only moves data in one direction
  - No loops!
  - Build loops internally into a block
- Natural expression for samples at the PHY layer
- For packet/frame data, this model starts to break down farther away from the antenna
  - At some point, generally useful to move to PDU/message passing model

Stream Tags for Annotating Samples



- Adds a control/logic/synchronous message interface to the data flow layer.
- Tags are associated with a specific item in the stream.
- Tags are "key: value" pairs.
- Moves downstream with the data; resampled with data

Asynchronous Message Passing



- Asynchronous messages from and to any block.
- A Publish-Subscribe model.
- Can directly post a message into or out of a block.
  - leads to direct interfaces in/out of GNU Radio.
- Message only Protocol Data Unit (PDU) blocks useful for frame/packet/segment work.