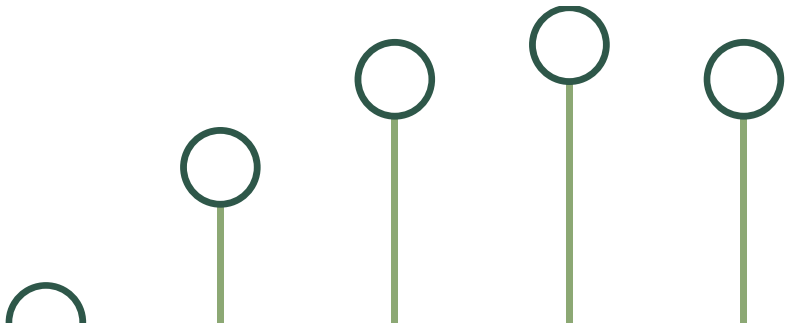
Ettus Research™
A National Instruments Company

RFNoC 3 Workshop

Part 1

Jonathon Pendlum – Ettus Research
Neel Pandeya – Ettus Research

August 2020



Schedule

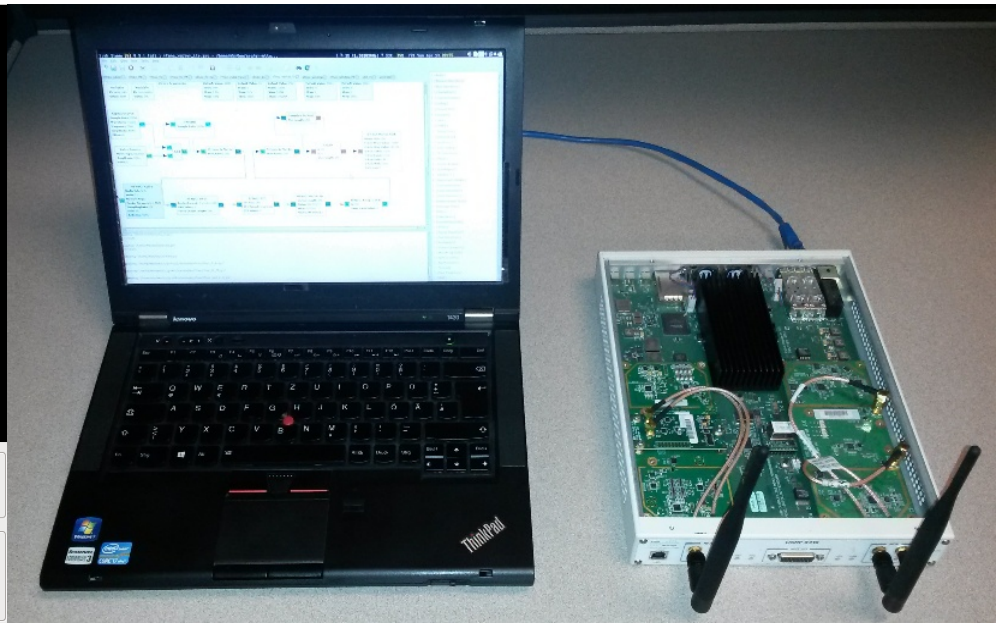
Ettus Research™
A National Instruments Company

- Part 1
  - RFNoC Framework Overview
  - Hands on Demos
- Part 2
  - RFNoC FPGA & Software Architecture
  - Hands on Computation Engine Development
  - Advanced RFNoC Topics

Host-Based SDR – Current Situation

Ettus Research™
A National Instruments Company

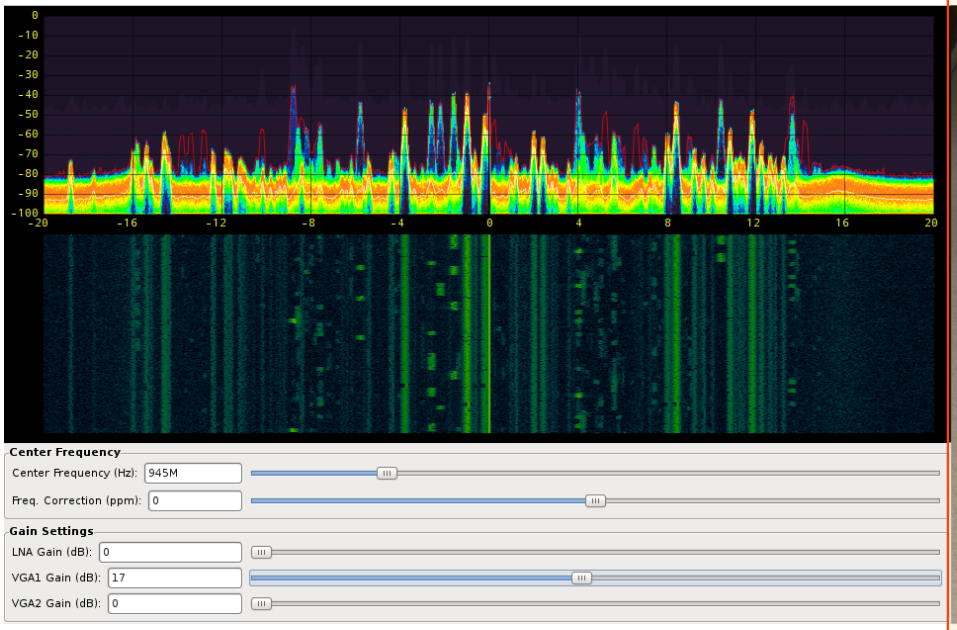
- PC + Flexible RF Hardware + SDR Framework



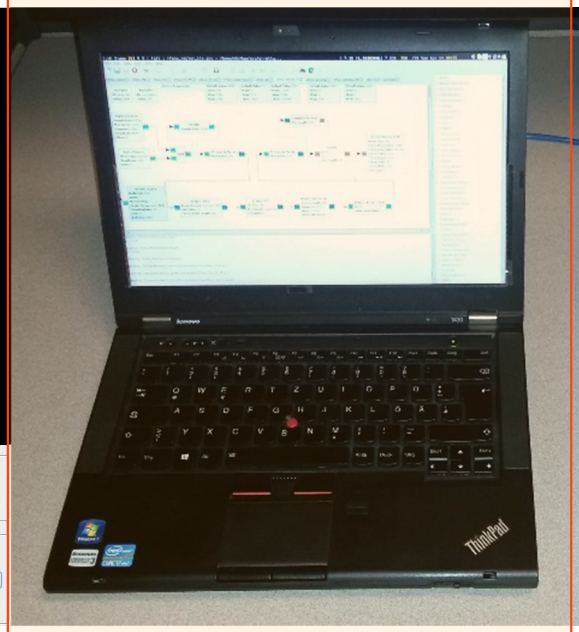
Host-Based SDR – Current Situation

Ettus Research™
A National Instruments Company

- PC + Flexible RF Hardware + SDR Framework
  - GPP: Multi-core + SIMD -- GNU Radio



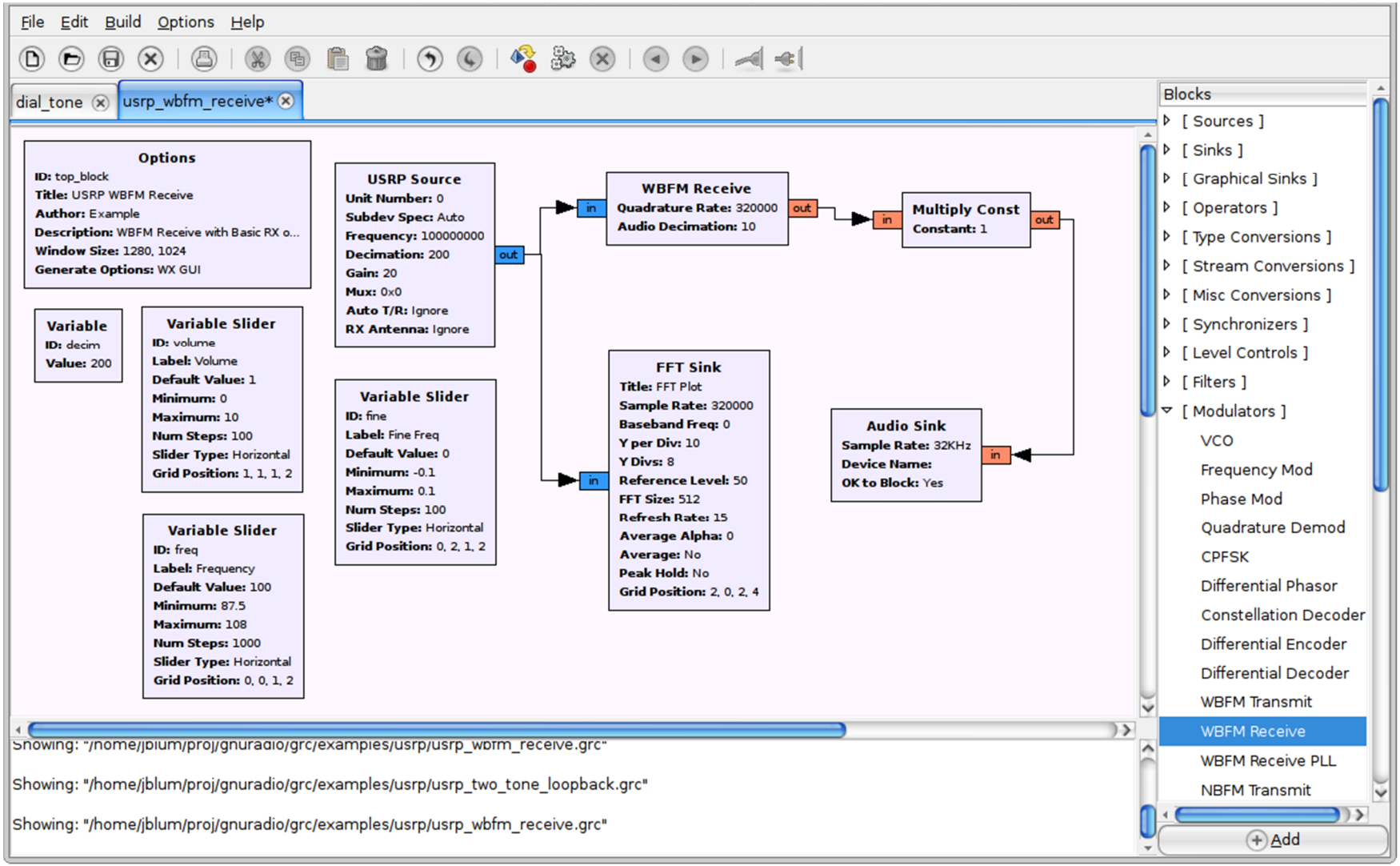
GPP



GNU Radio

Ettus Research™
A National Instruments Company

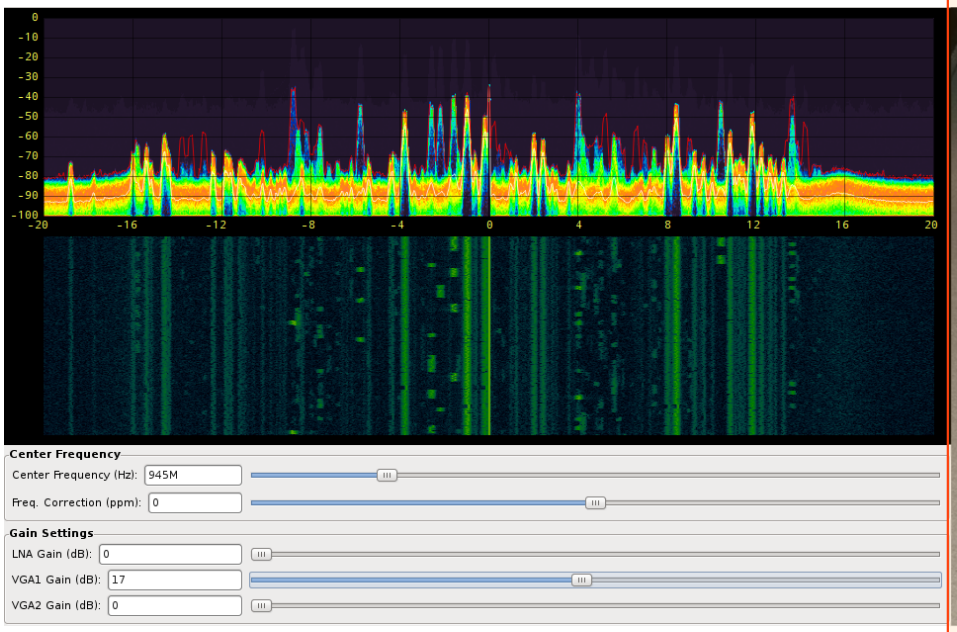
Open source toolkit for developing software radios



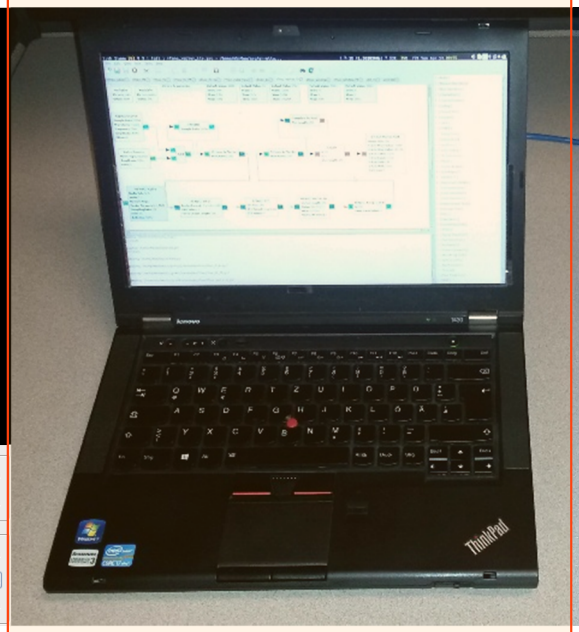
Host-Based SDR – Current Situation

Ettus Research™ A National Instruments Company

- PC + Flexible RF Hardware + SDR Framework
  - GPP: Multi-core + SIMD -- GNU Radio



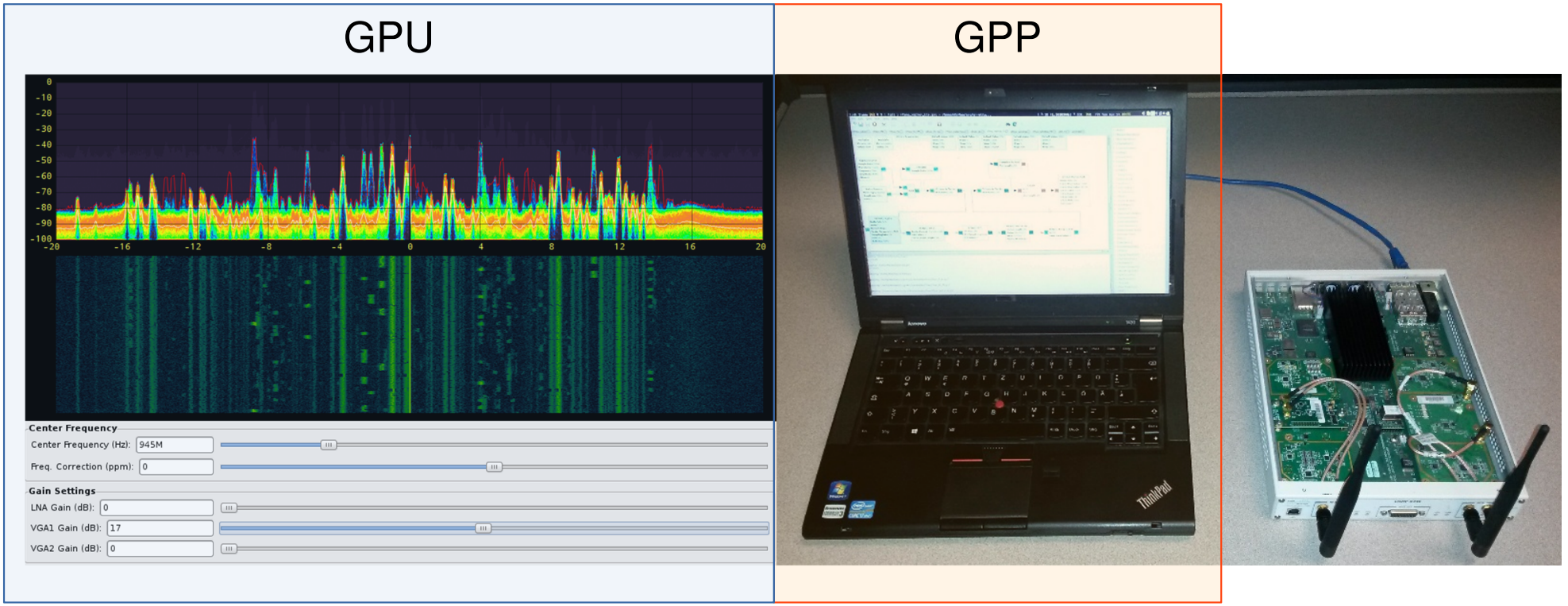
GPP



Host-Based SDR – Current Situation

Ettus Research™ A National Instruments Company

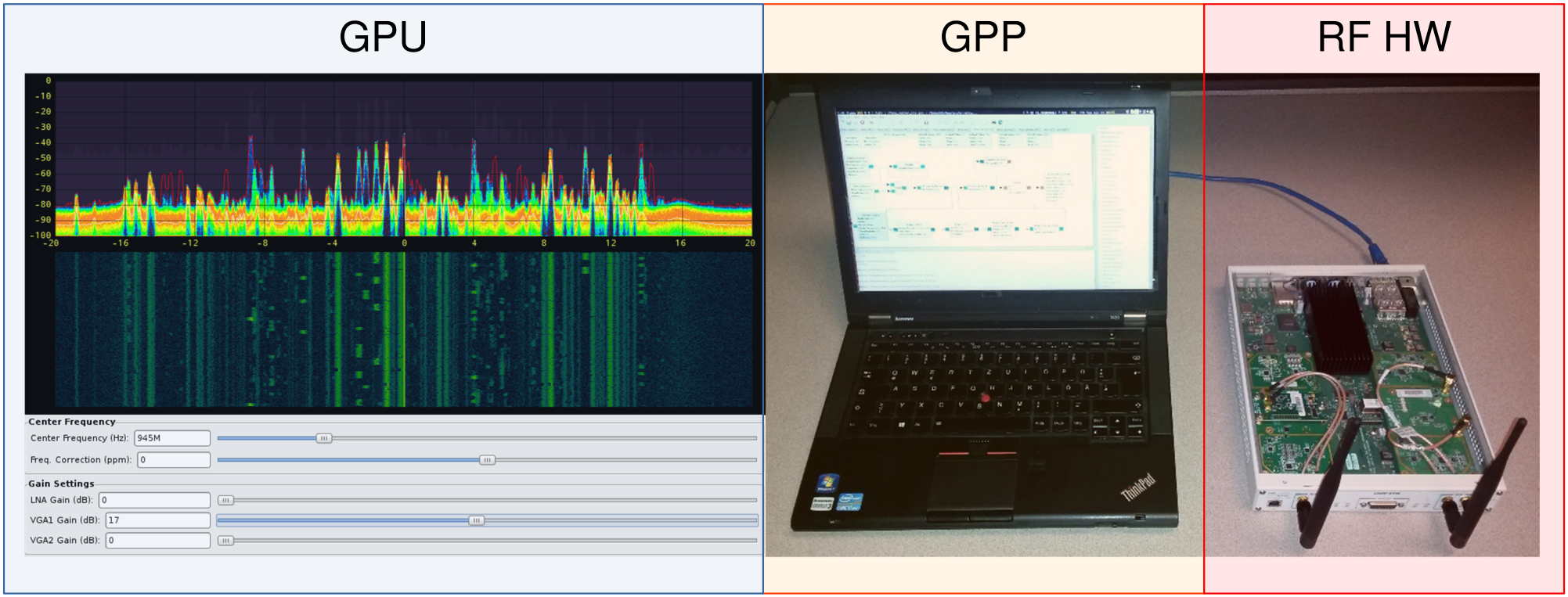
- PC + Flexible RF Hardware + SDR Framework
  - GPP: Multi-core + SIMD -- GNU Radio
  - GPU: High performance FP -- OpenCL, gr-fosphor



Host-Based SDR – Current Situation

Ettus Research™ A National Instruments Company

- PC + Flexible RF Hardware + SDR Framework
  - GPP: Multi-core + SIMD -- GNU Radio
  - GPU: High performance FP -- OpenCL, gr-fosphor
  - RF HW: Wide bandwidth, large FPGA -- Rate change DSP



Universal Software Radio Peripheral

Ettus Research™
A National Instruments Company

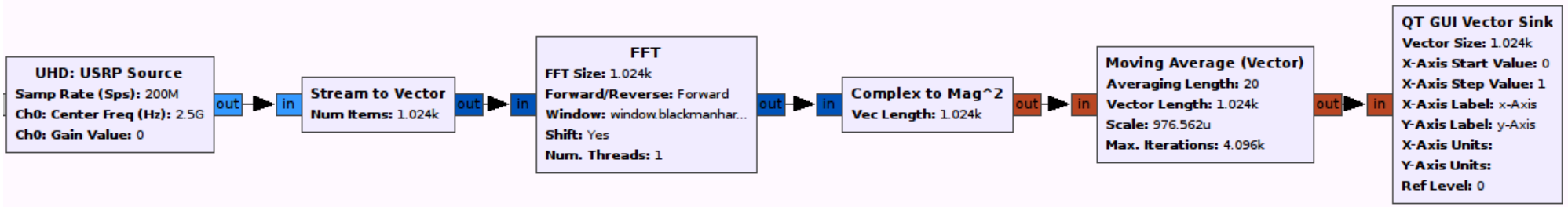
	Gen 1	Gen 2	Gen 3 (E310)	Gen 3 (X310)
FPGA	Cyclone 1	Spartan 3	Zynq	Kintex 7
Logic Cells	12K	53K	85K	406K
Memory	26KB	252KB	560KB	3180KB
Multipliers	NONE!	126	220	1540
Clock Rate	64 MHz	100 MHz	200 MHz	250 MHz
RF Bandwidth	8 MHz	50 MHz	128 MHz	640 MHz
Free Space	NONE!	~50%	~60%	~75%




Challenges

Ettus Research™
A National Instruments Company

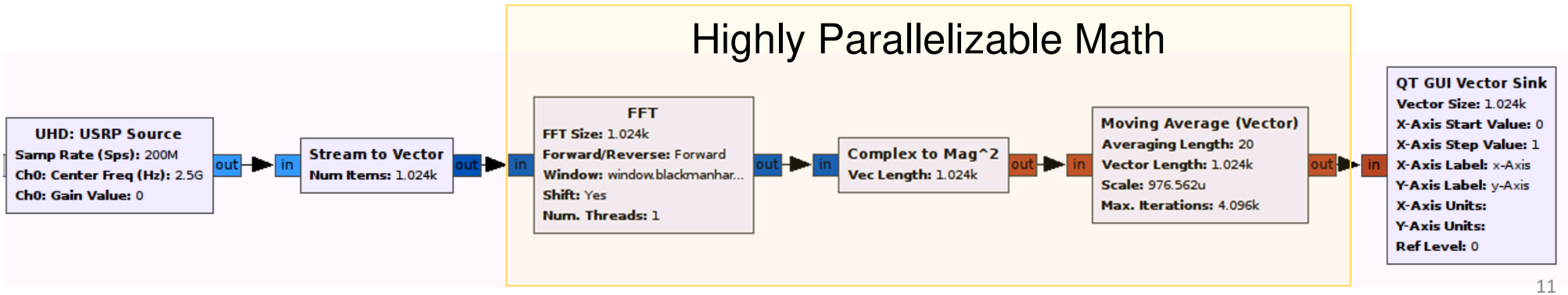
- Massive processing requirements
  - Welsh's algorithm for Power Spectrum Estimation
  - 1024 FFT + $|X|^2$ + Moving Average at 200 MSPS



Challenges

Ettus Research™
A National Instruments Company

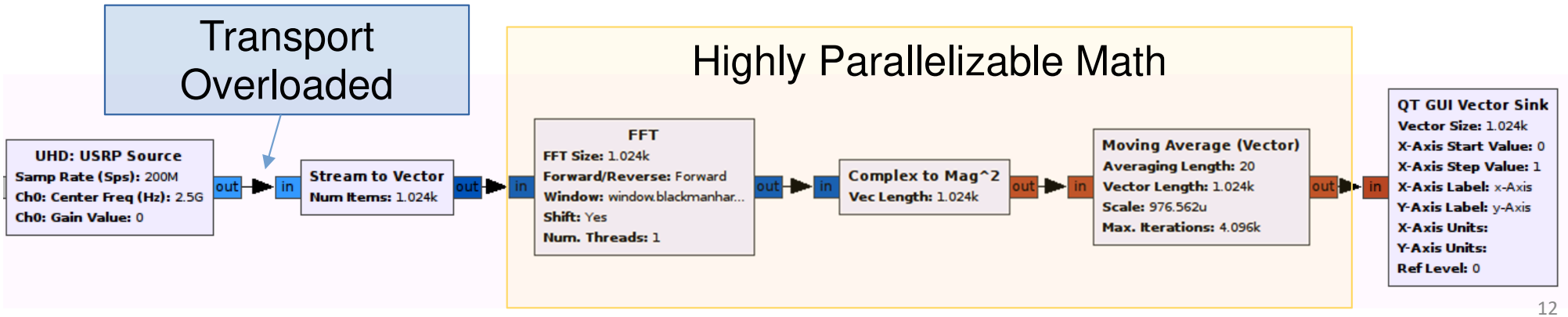
- Massive processing requirements
  - Welsh's algorithm for Power Spectrum Estimation
  - 1024 FFT + $|X|^2$ + Moving Average at 200 MSPS



Challenges

Ettus Research™
A National Instruments Company

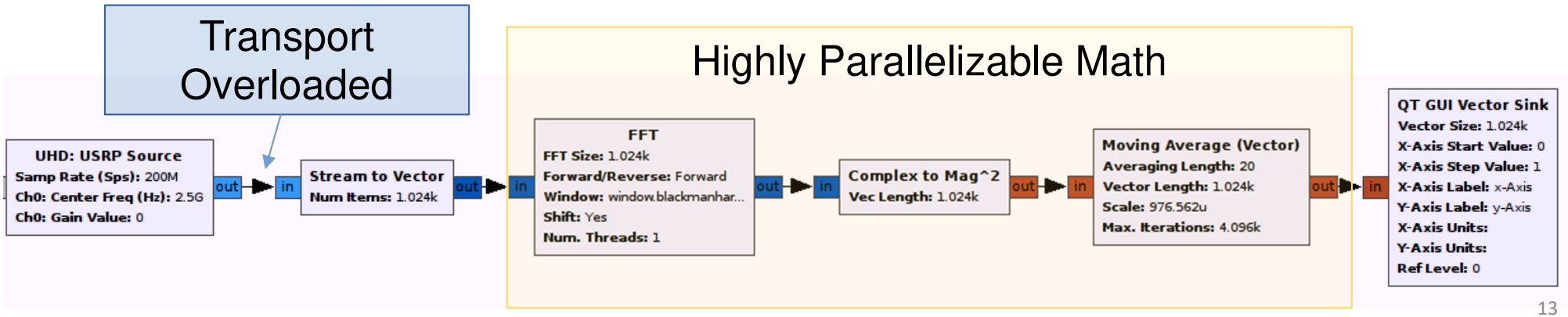
- Massive processing requirements
  - Welsh's algorithm for Power Spectrum Estimation
  - 1024 FFT + $|X|^2$ + Moving Average at 200 MSPS
- Overloaded transport
  - 200e6 samp/sec * 32 bits/samp => **6.4 Gb/sec**



Challenges

Ettus Research™
A National Instruments Company

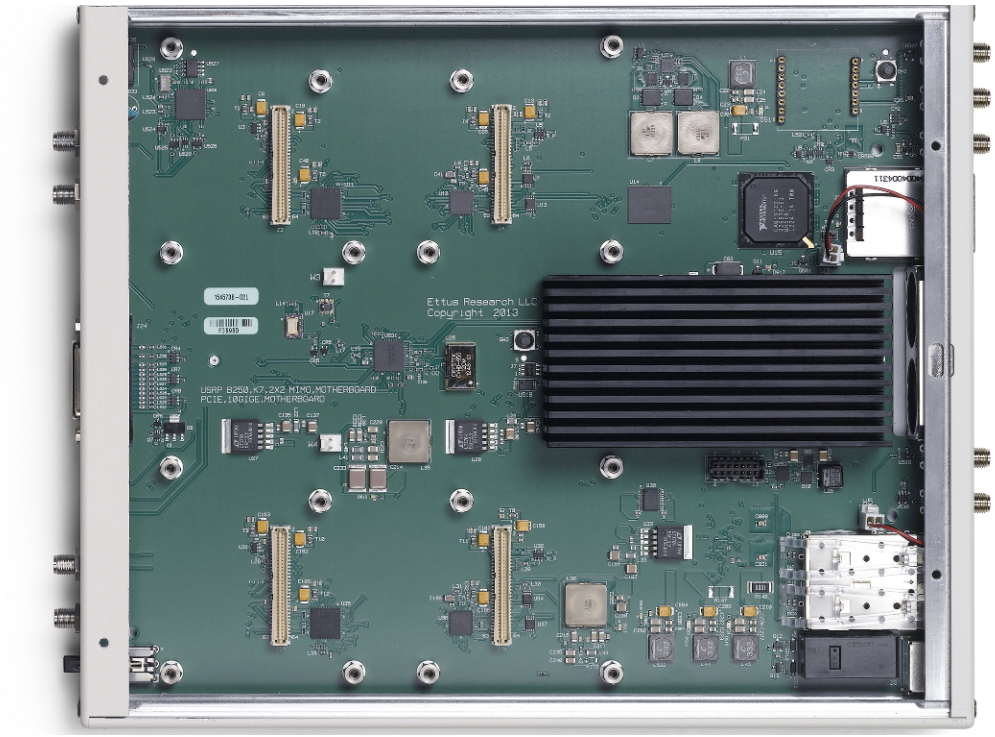
- Massive processing requirements
  - Welsh's algorithm for Power Spectrum Estimation
  - 1024 FFT + $|X|^2$ + Moving Average at 200 MSPS
- Overloaded transport
  - 200e6 samp/sec * 32 bits/samp => **6.4 Gb/sec**
- Latency and Determinism
  - Ethernet latency, OS scheduling, precise timing



Opportunity: Use the FPGA!

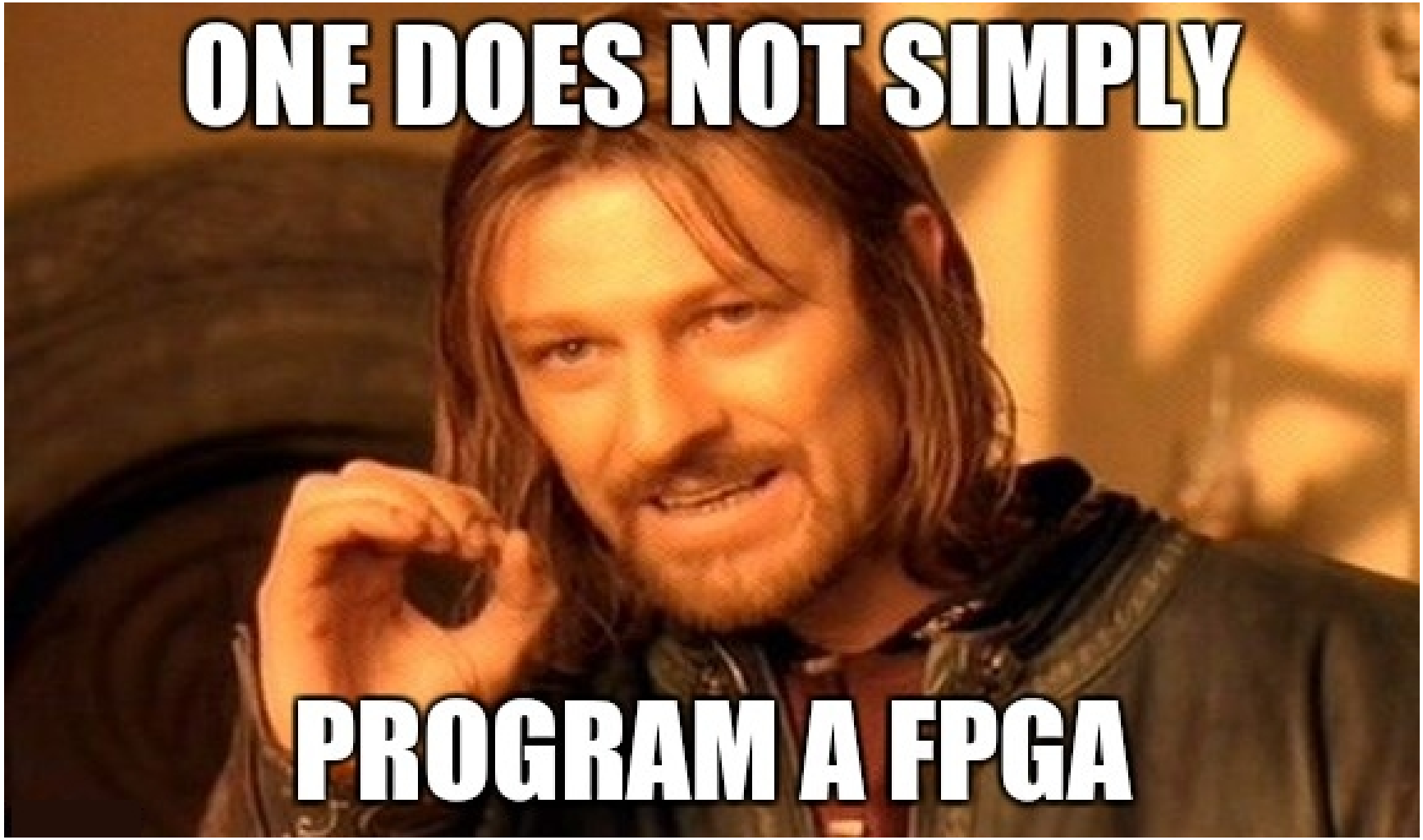
Ettus Research™
A National Instruments Company

- Everything USRP is open source, available online (code, firmware, schematics)
- Contains big and expensive FPGA!
- Why do customers not use it?



FPGAs: Hard to use… slow to develop

Ettus Research™
A National Instruments Company



Domain vs FPGA Experts

Ettus Research™
A National Instruments Company

- FPGA development is not a requirement of a communications engineering curriculum
- Math in FPGAs is hard
- Complicated system architecture

almost pure-noise channels. This intuition is clarified more by the following inequality. It is shown in [1] that for any B-DMC W,

$$1 - I(W) \leqslant Z(W) \leqslant \sqrt{1 - I(W)^2} \tag{2}$$

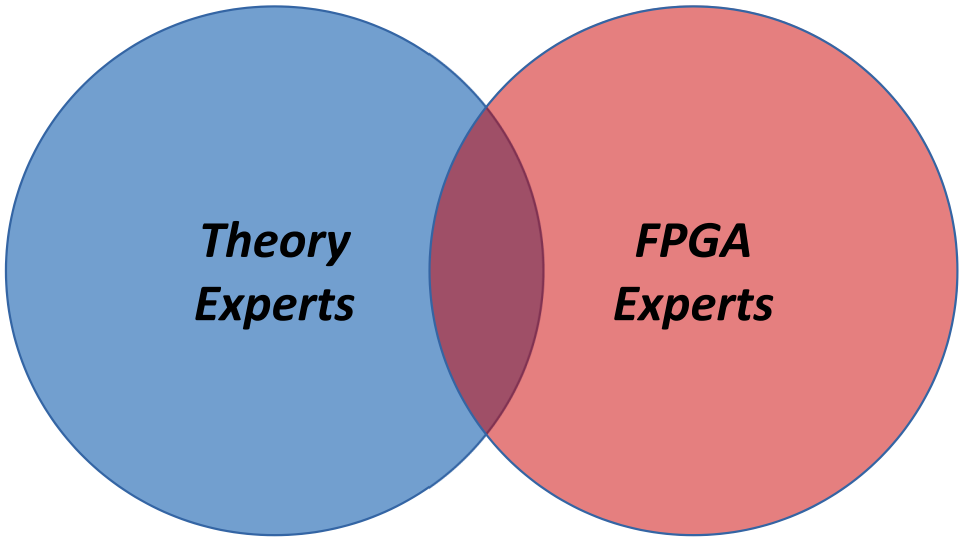
where $I(W)$ is the symmetric capacity of W.

Let W^N denote the channels that results from N independent copies of W i.e. the channel $\langle \{0,1\}^N, \mathscr{Y}^N, W^N \rangle$ given by

$$W^N(y_1^N | x_1^N) \stackrel{\text{def}}{=} \prod_{i=1}^{N} W(y_i | x_i) \tag{3}$$

where $x_1^N = (x_1, x_2, \ldots, x_N)$ and $y_1^N = (y_1, y_2, \ldots, y_N)$. Then the *combined* channel $\langle \{0,1\}^N, \mathscr{Y}^N, \widetilde{W} \rangle$ is defined with transition probabilities given by

$$\widetilde{W}(y_1^N | u_1^N) \stackrel{\text{def}}{=} W^N(y_1^N | u_1^N G_N) = W^N(y_1^N | u_1^N R_N G^{\otimes n})$$



RFNoC: RF Network on Chip

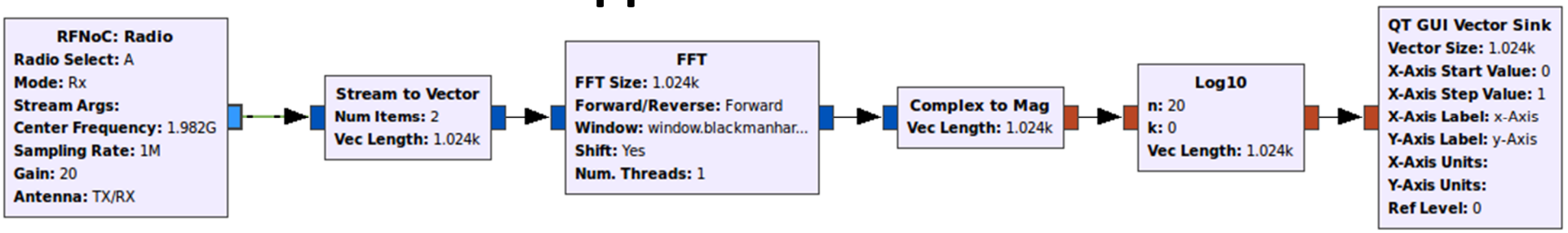
Ettus Research™
A National Instruments Company

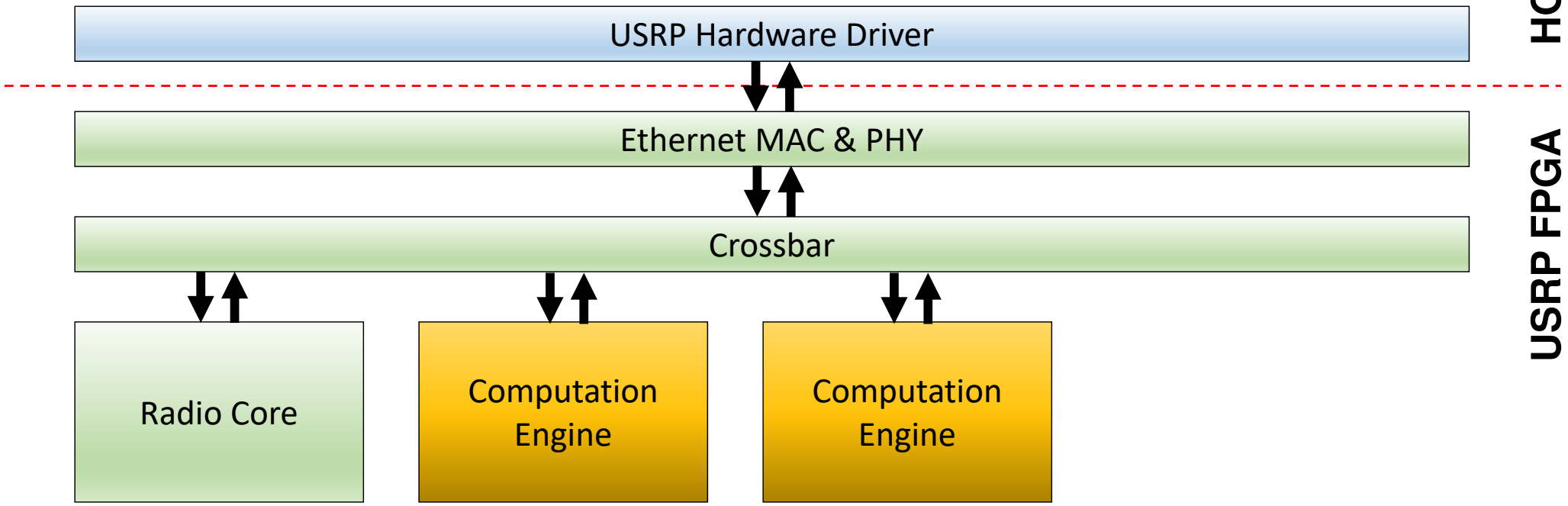
- Make USRP FPGA acceleration more accessible
- Software API + FPGA infrastructure
  - Handles FPGA – Host communication / dataflow
- Provides users simple software and HDL interfaces
  - Infrastructure transparent to user -- reusable code
- Open source
- Fully supported in GNU Radio
  - Modularity and composability

RFNoC Architecture

Ettus Research™
A National Instruments Company

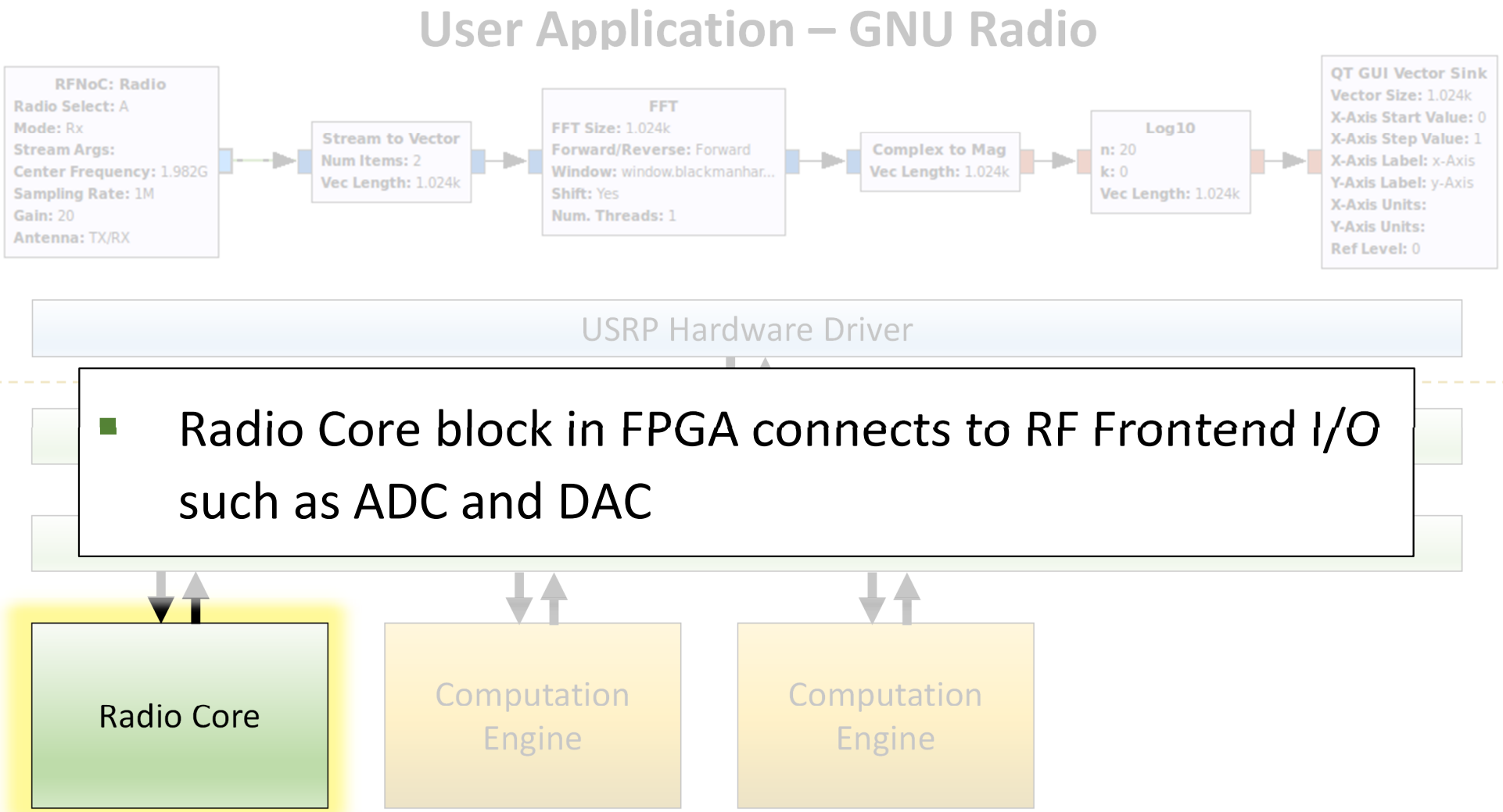
User Application – GNU Radio





RFNoC Architecture

Ettus Research™
A National Instruments Company

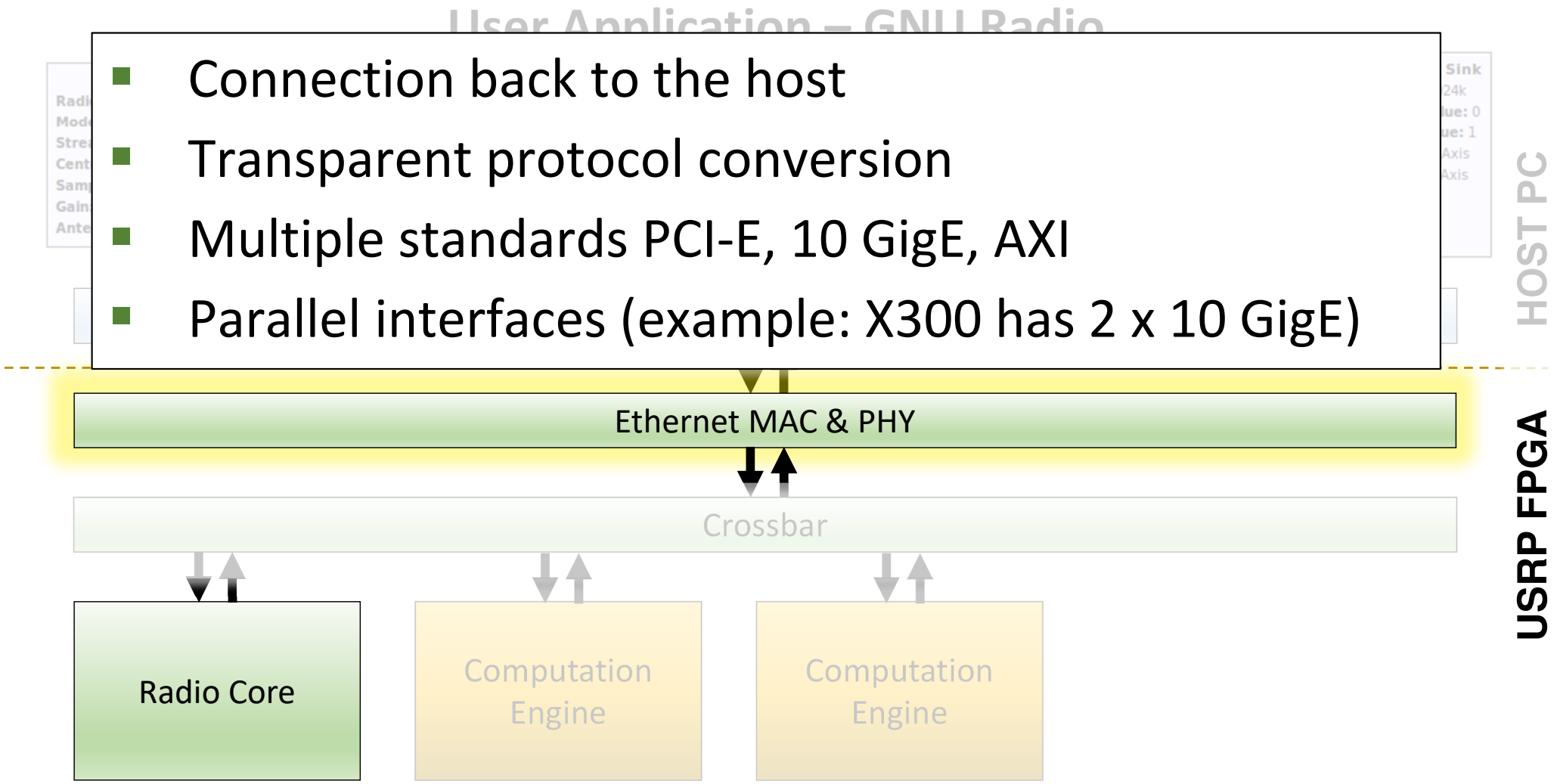


- Radio Core block in FPGA connects to RF Frontend I/O such as ADC and DAC

RFNoC Architecture

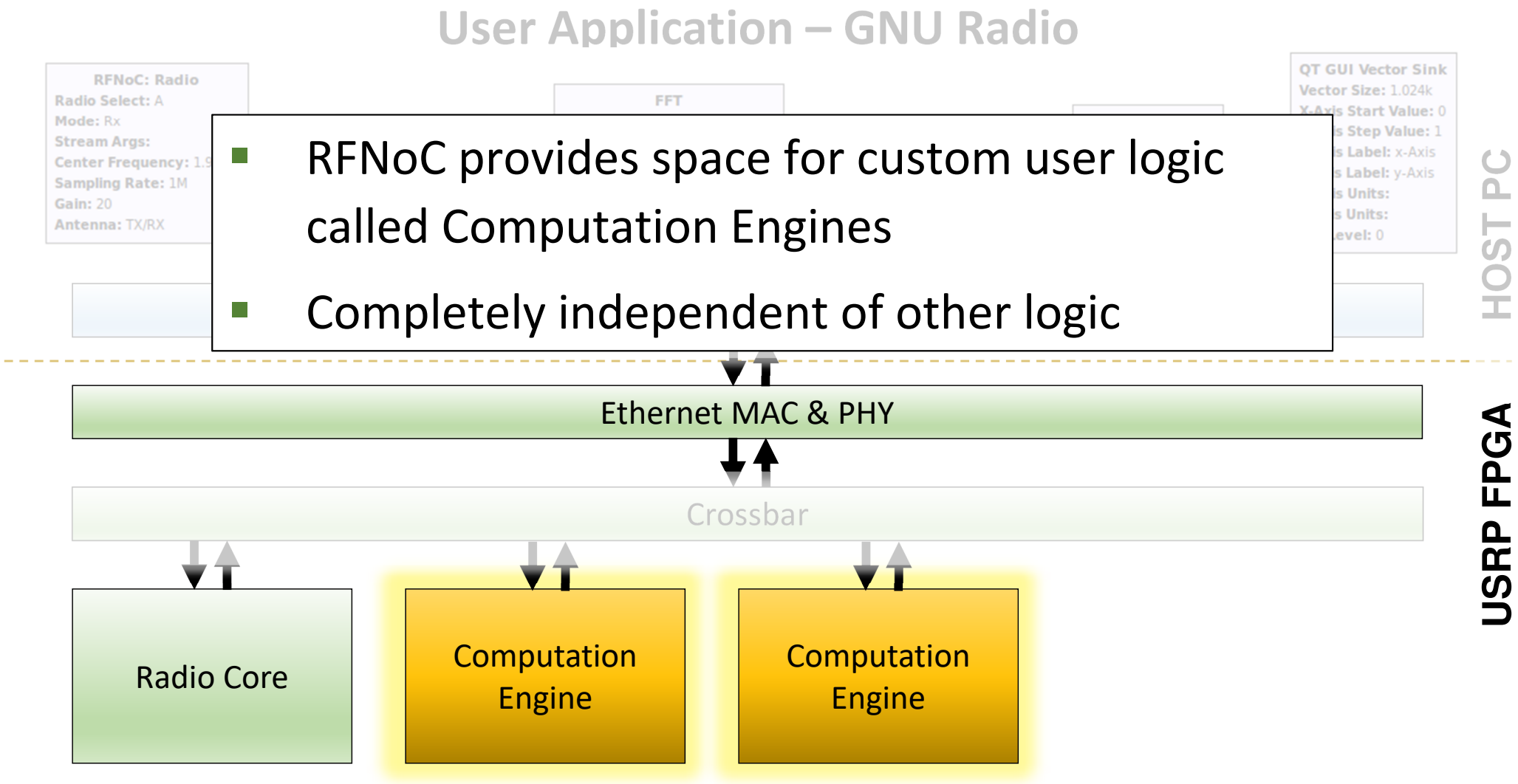
Ettus Research™ A National Instruments Company

- Connection back to the host
- Transparent protocol conversion
- Multiple standards PCI-E, 10 GigE, AXI
- Parallel interfaces (example: X300 has 2 x 10 GigE)



RFNoC Architecture

Ettus Research™
A National Instruments Company

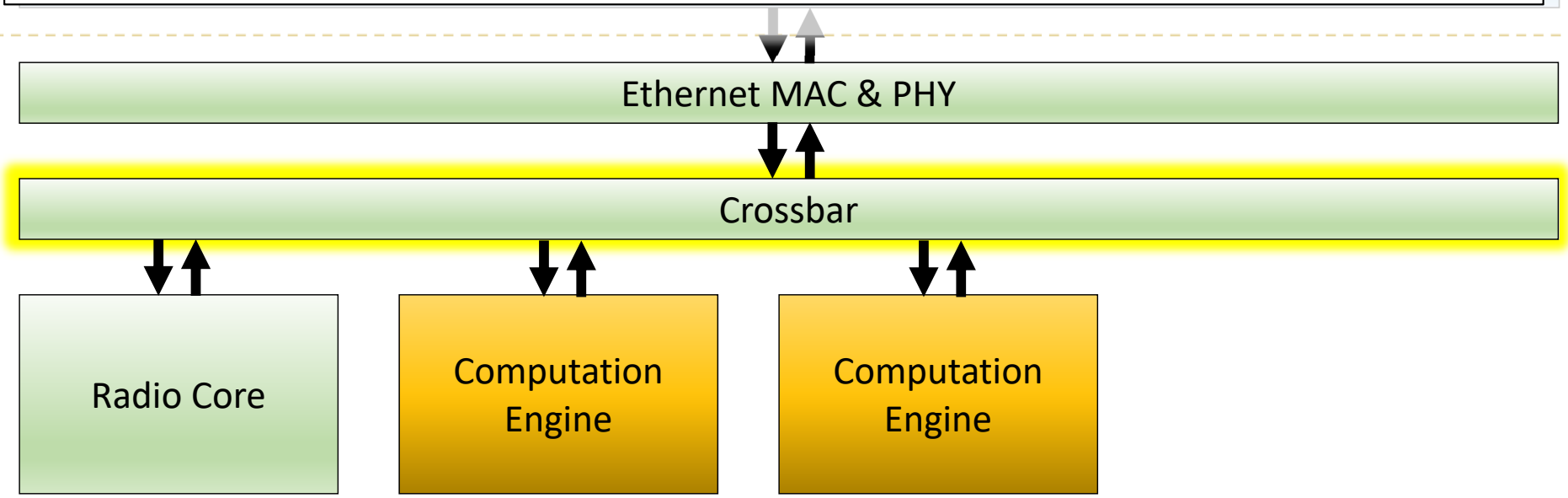


RFNoC Architecture

Ettus Research™ A National Instruments Company

User Application – GNU Radio

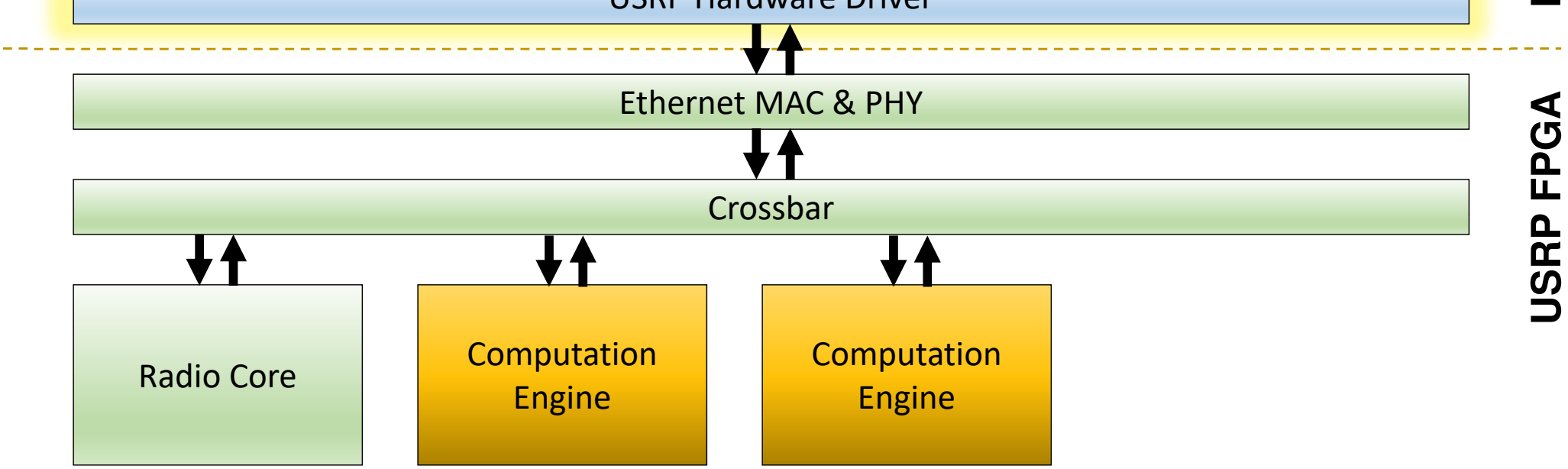
- Crossbar connects Computation Engines, Radio Core, and I/O interfaces (e.g. Ethernet)
  - Packet switched
  - Fully connected crossbar



RFNoC Architecture

Ettus Research™ A National Instruments Company

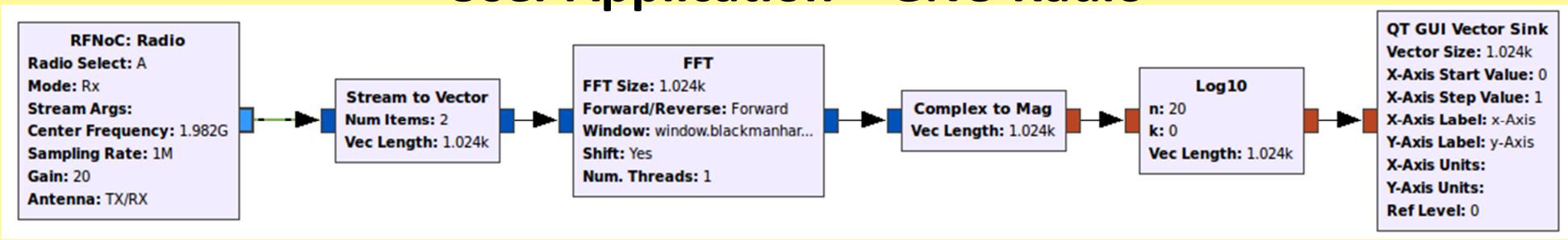
- Software API to:
  - Configure USRP hardware & RFNoC FPGA infrastructure
  - Provide user sample data (r/w buffers) & control (r/w regs) interfaces



RFNoC Architecture

Ettus Research™
A National Instruments Company

User Application – GNU Radio



- User application
  - Standalone: C, C++, Python
  - Framework: GNU Radio

HOST PC

USRP FPGA

Crossbar

Radio Core

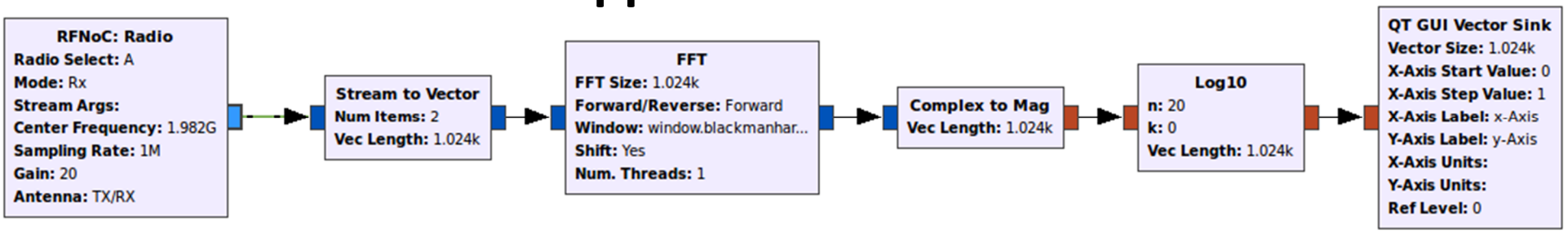
Computation Engine

Computation Engine

RFNoC Example

Ettus Research™
A National Instruments Company

User Application – GNU Radio



HOST PC

- Example: Plotting Frequency Spectrum

Crossbar

USRP FPGA

Radio Core

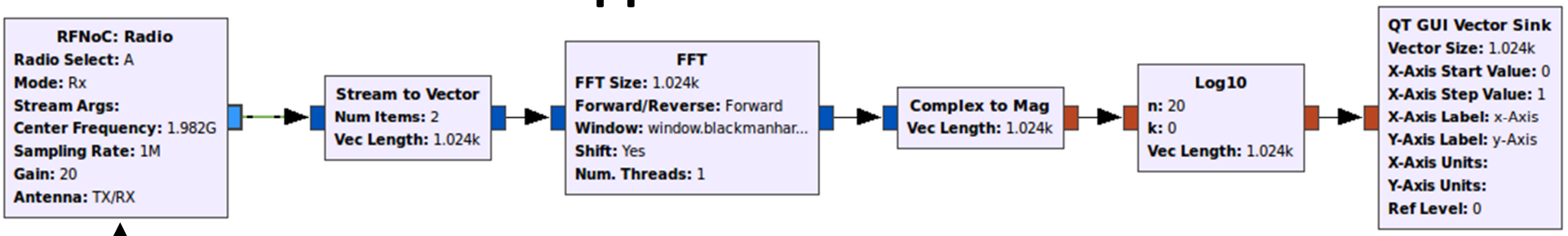
Computation Engine

Computation Engine

RFNoC Example

Ettus Research™
A National Instruments Company

User Application – GNU Radio



- Radio block in GNU Radio represents the Radio Core RFNoC block in FPGA

HOST PC

USRP FPGA

Crossbar

Radio Core

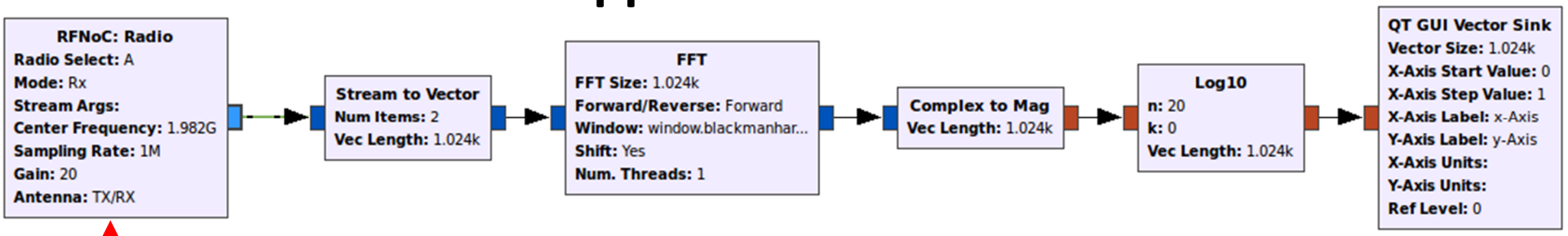
Computation Engine

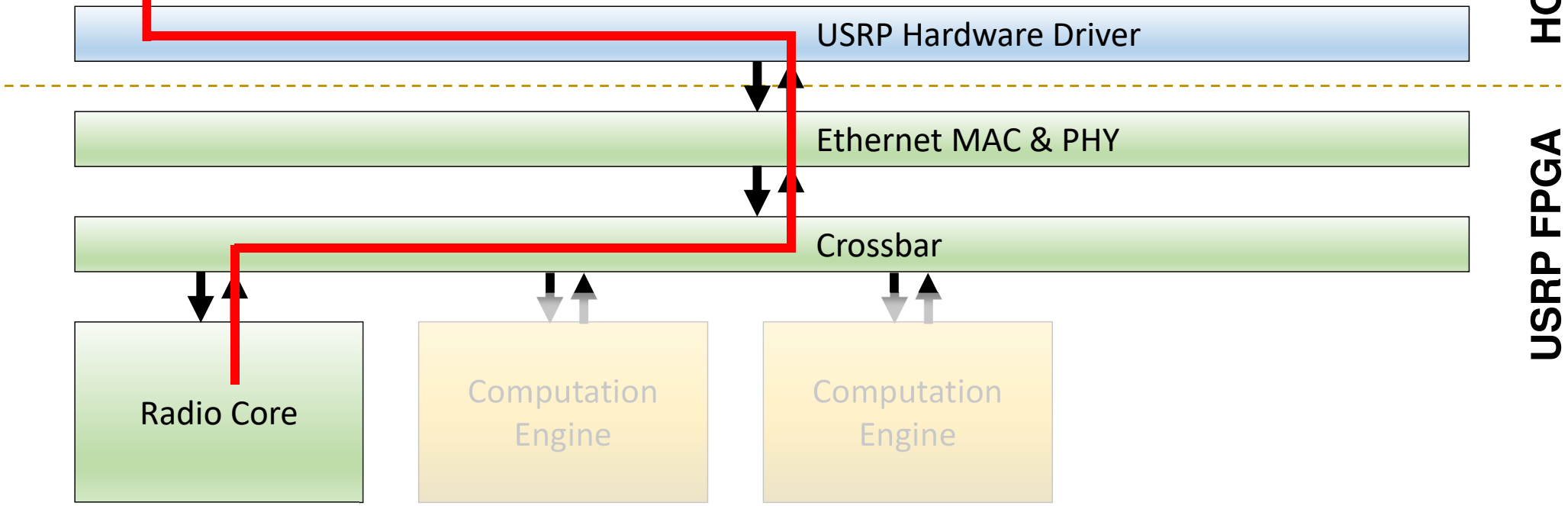
Computation Engine

RFNoC Example

Ettus Research™
A National Instruments Company

User Application – GNU Radio

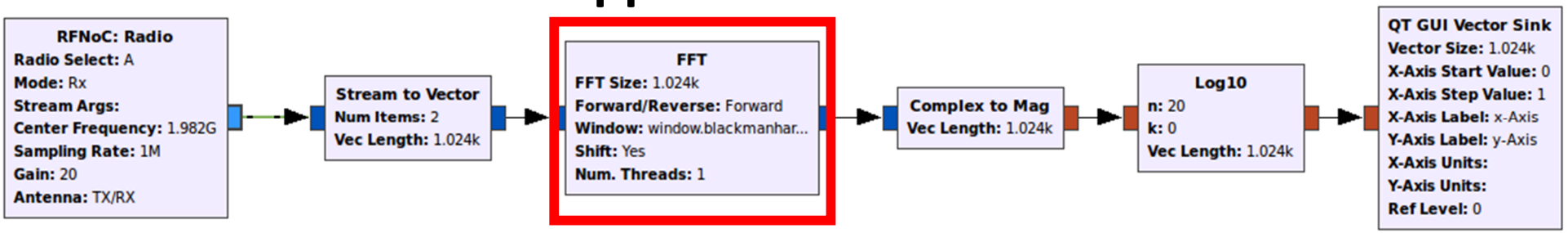




RFNoC Example

Ettus Research™
A National Instruments Company

User Application – GNU Radio



HOST PC

USRP FPGA

- Profiling reveals FFT is a hotspot

Crossbar

Radio Core

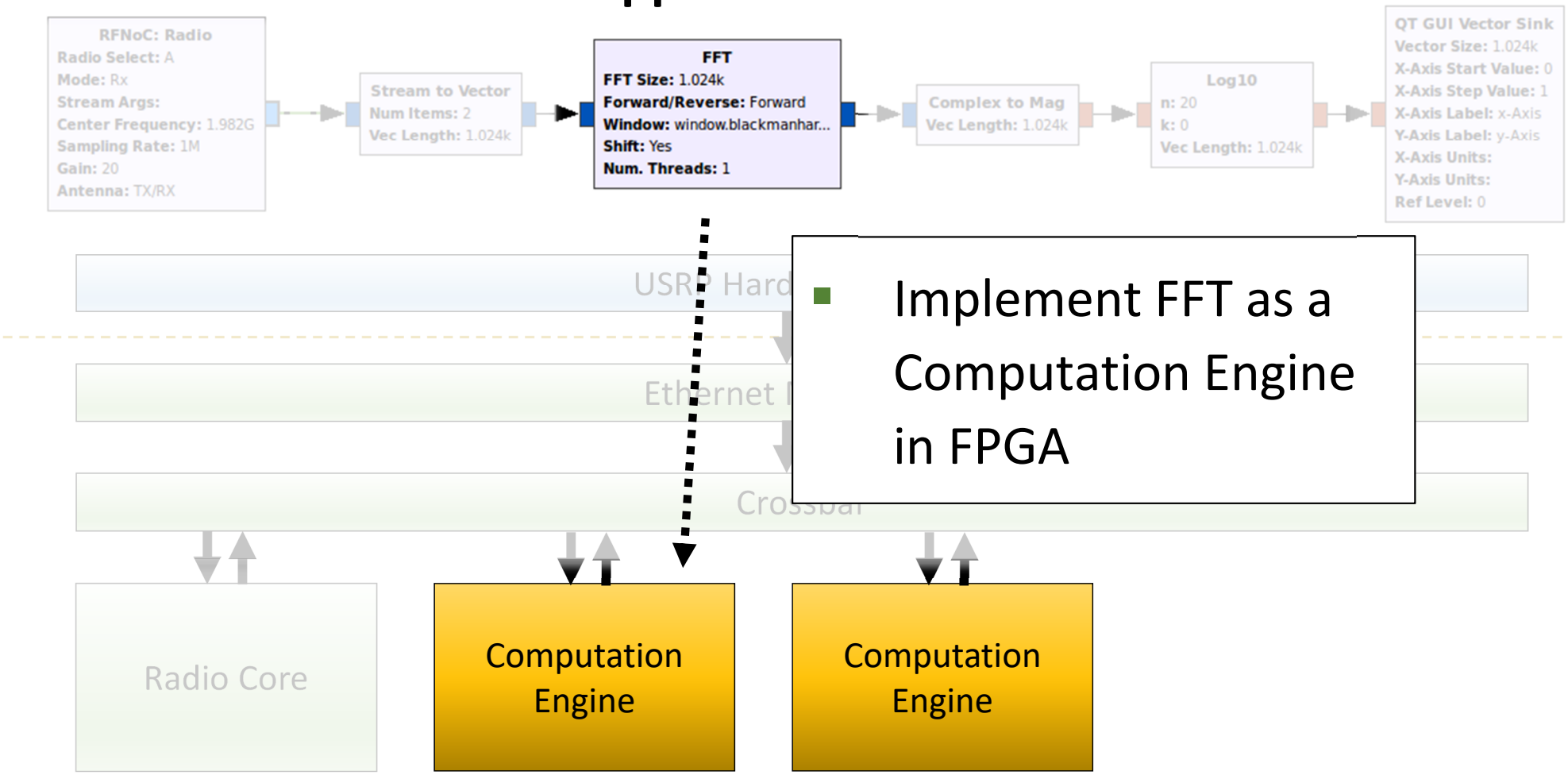
Computation Engine

Computation Engine

RFNoC Example

Ettus Research™
A National Instruments Company

User Application – GNU Radio

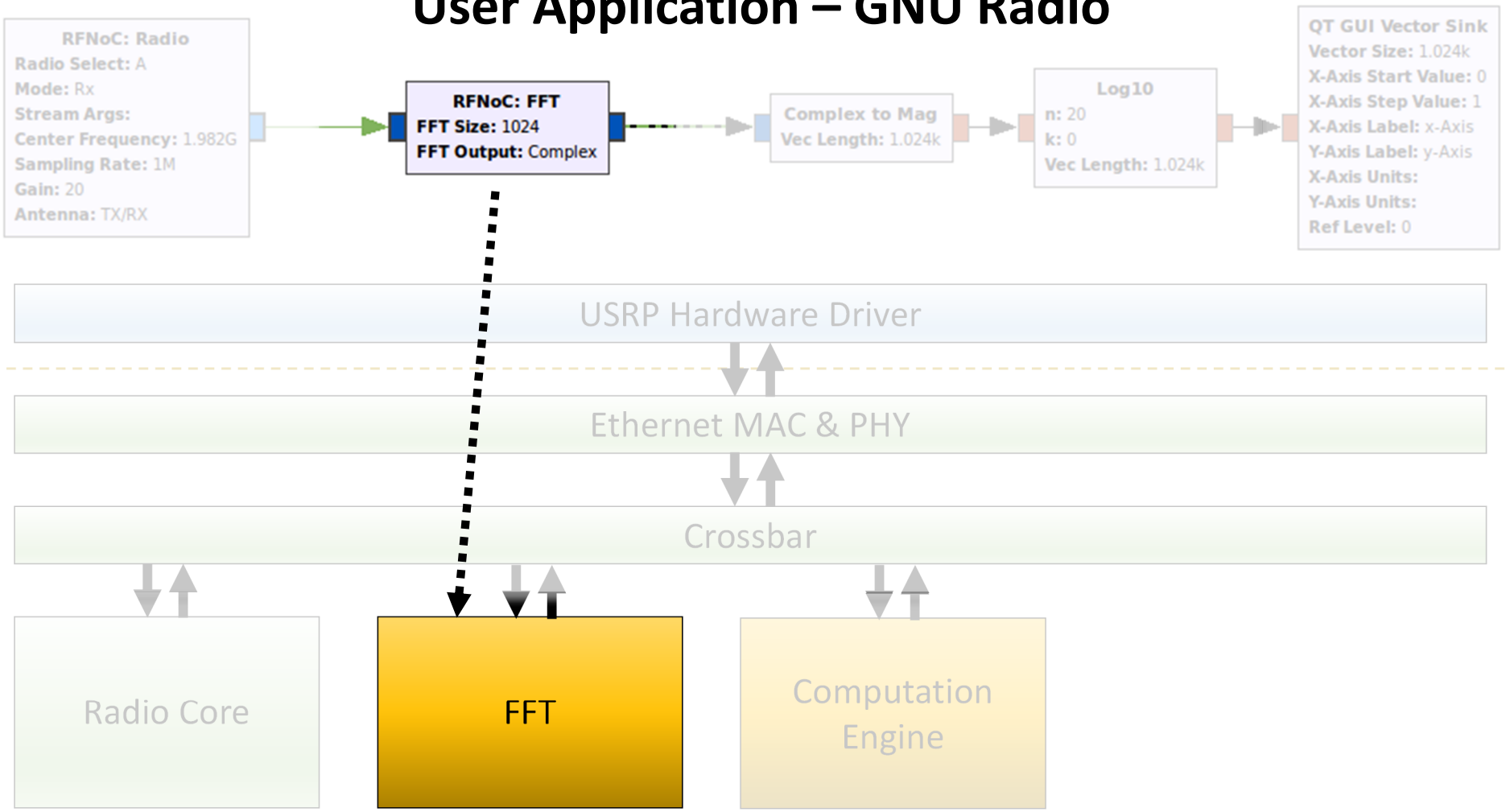


- Implement FFT as a Computation Engine in FPGA

RFNoC Architecture

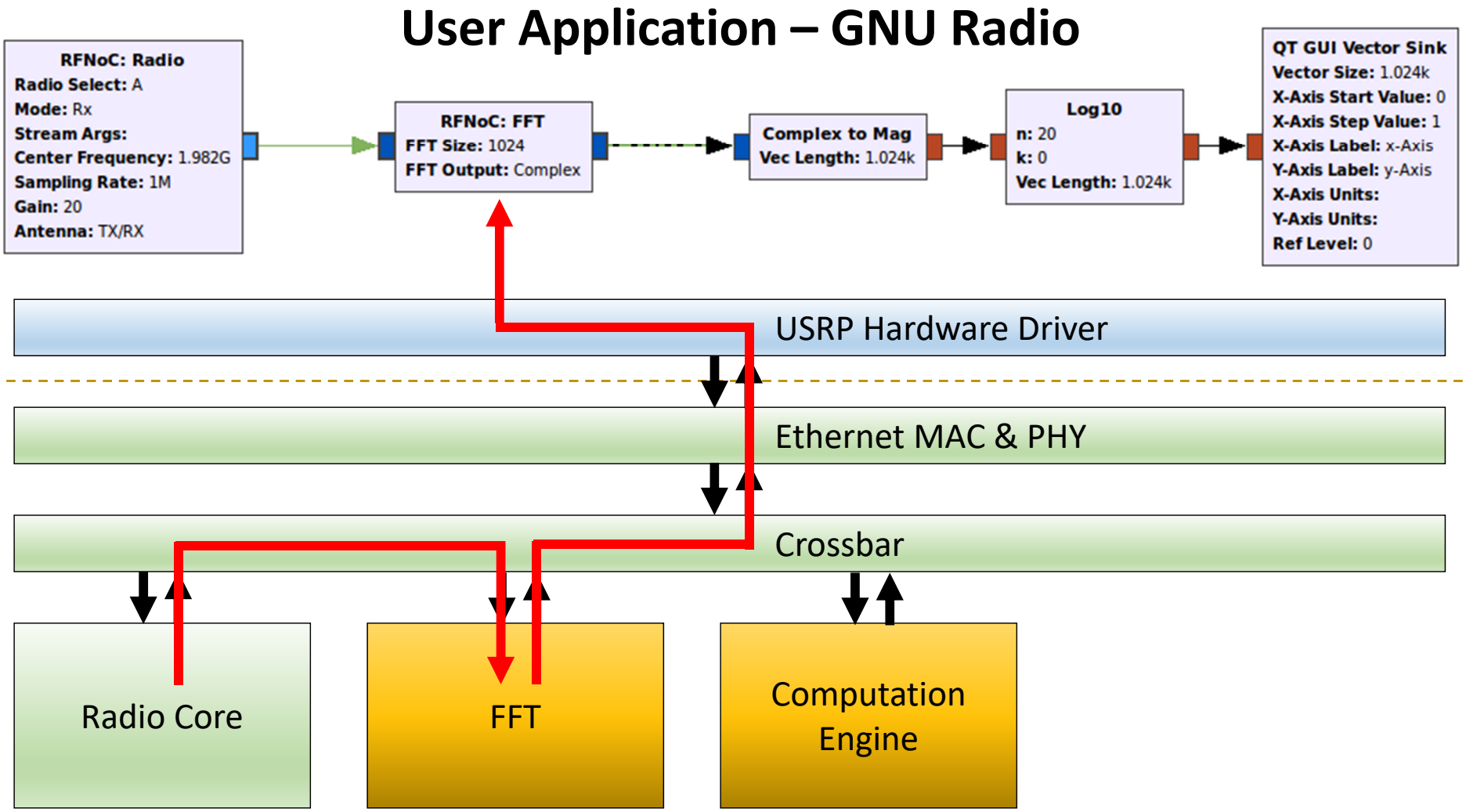
Ettus Research™
A National Instruments Company

User Application – GNU Radio



RFNoC Architecture

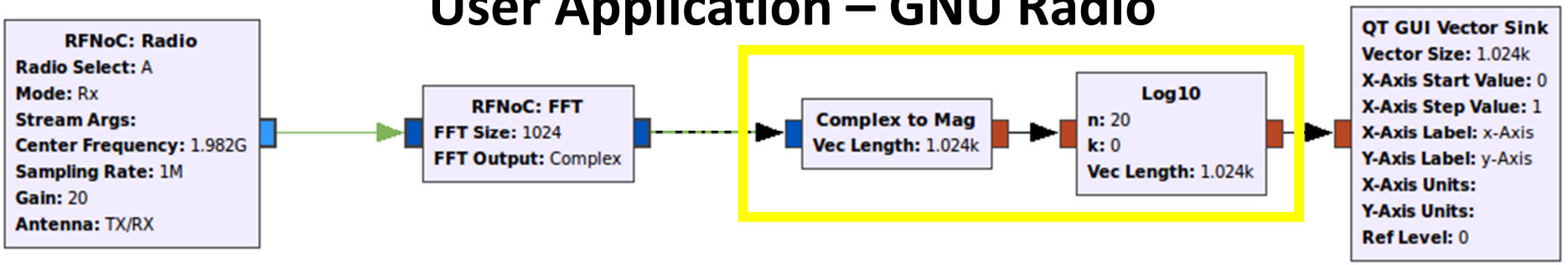
Ettus Research™
A National Instruments Company



RFNoC Architecture

Ettus Research™
A National Instruments Company

User Application – GNU Radio



HOST PC

USRP FPGA

- Profile again and decide if more FPGA acceleration is needed
- Note: Already have these blocks in RFNoC

Crossbar

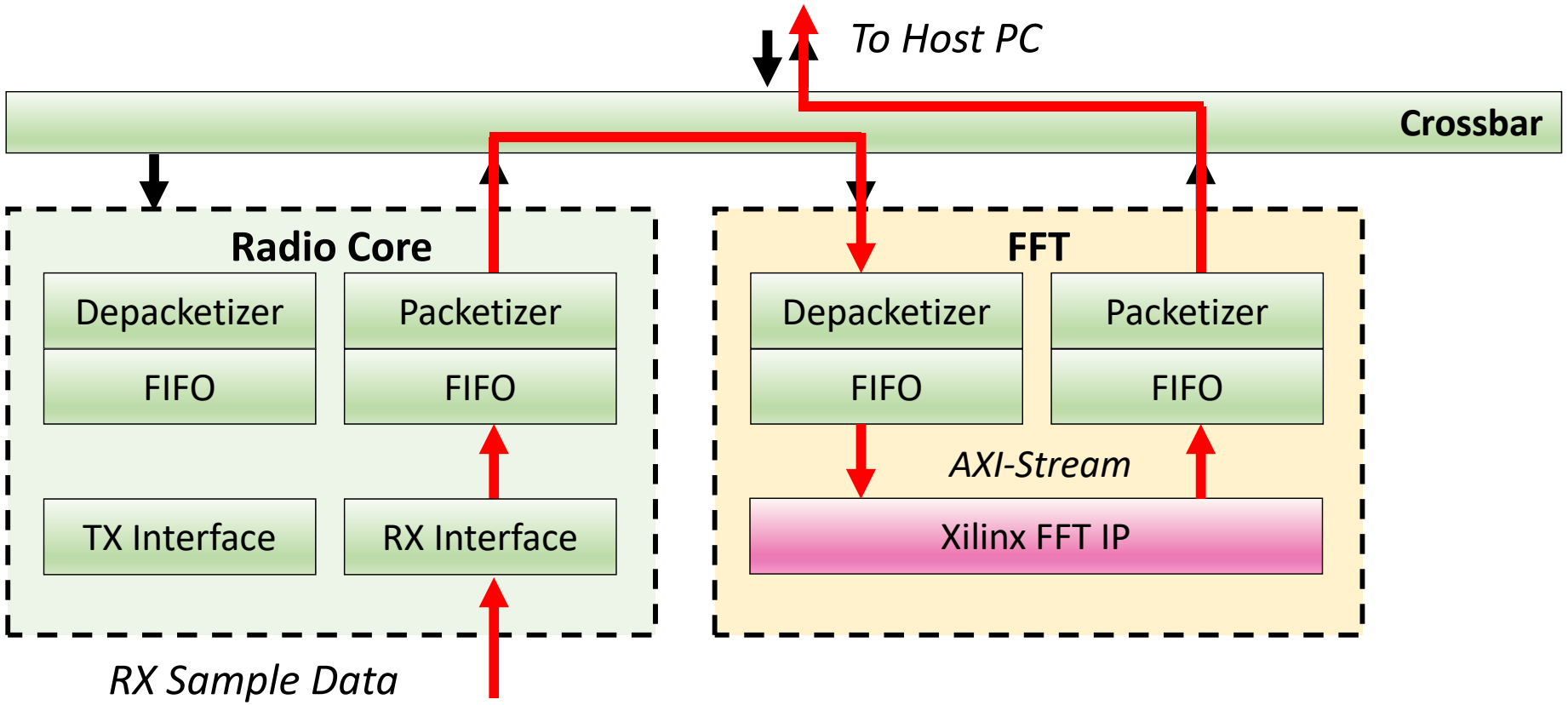
Radio Core

FFT

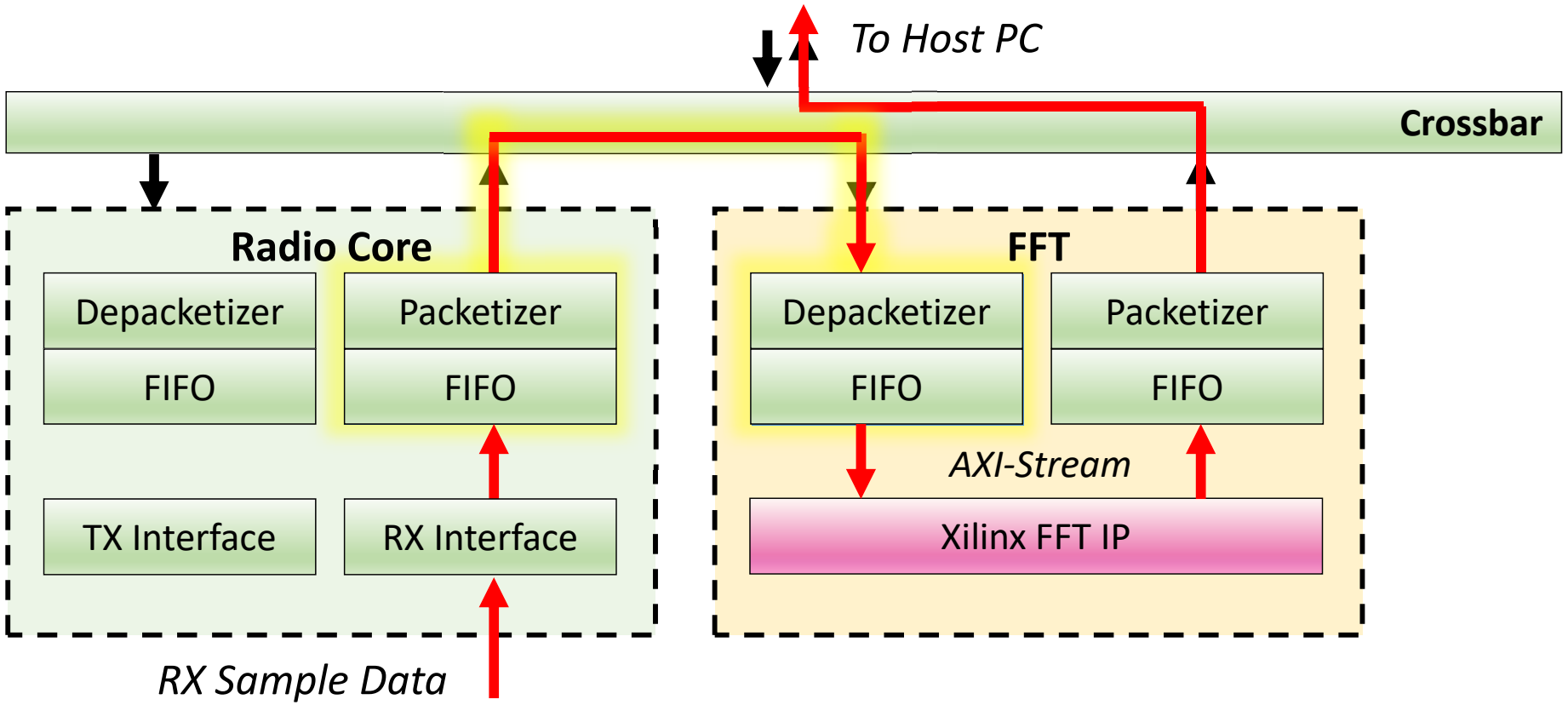
Computation Engine

Computation Engine

Ettus Research™ A National Instruments Company

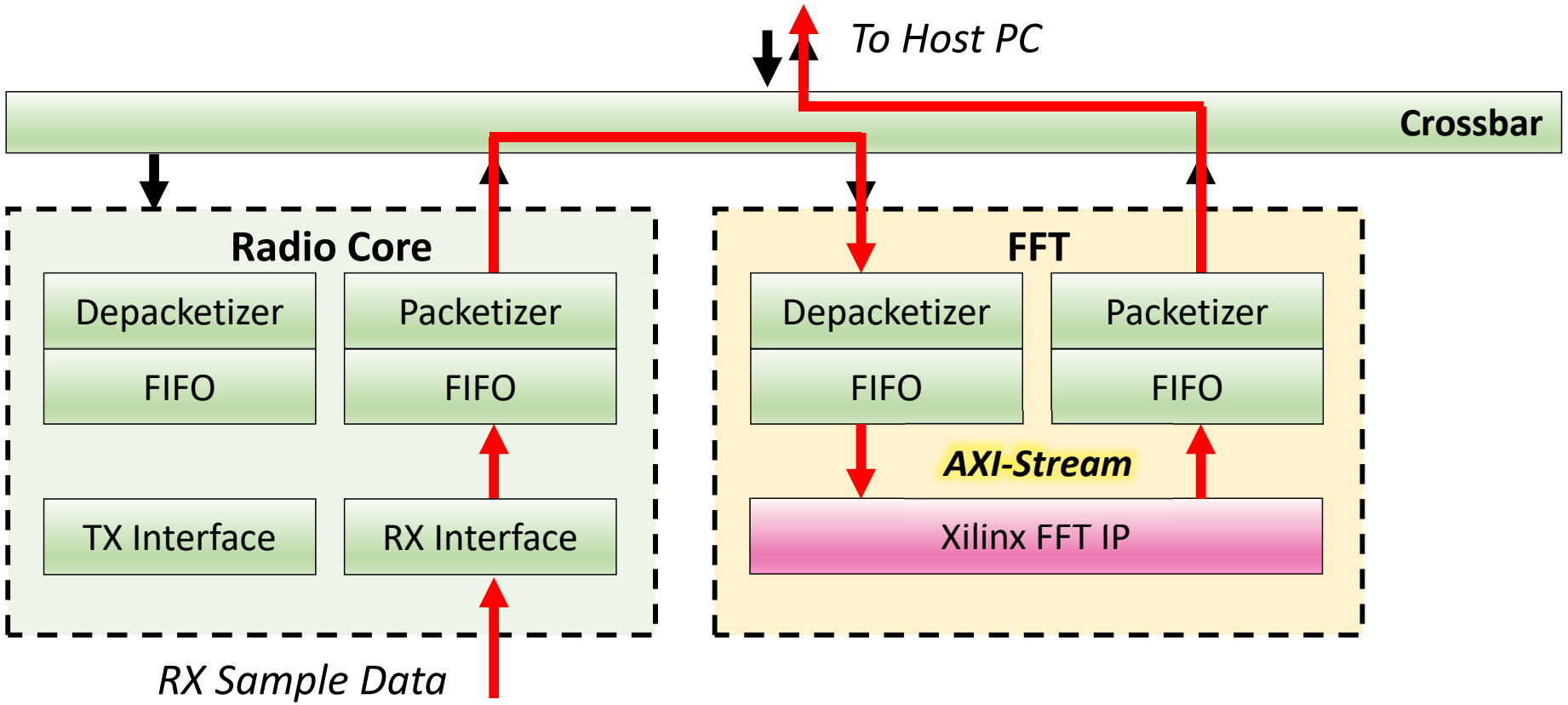


Computation Engine



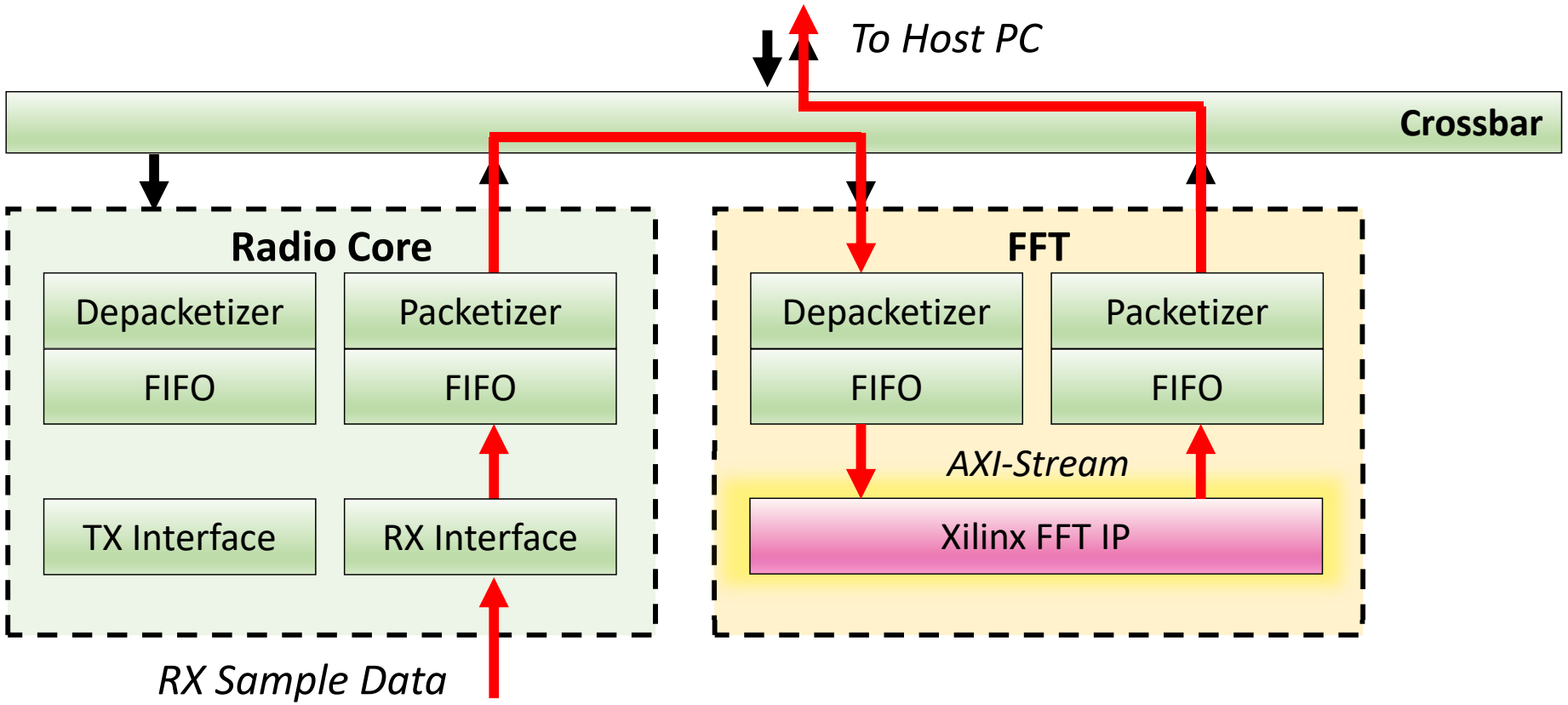
- FIFO to FIFO, packetization, flow control
- Provided by RFNoC infrastructure

Computation Engine



- User interfaces to RFNoC via AXI-Stream
  - Industry standard (ARM), easy to use
  - Large library of existing IP cores

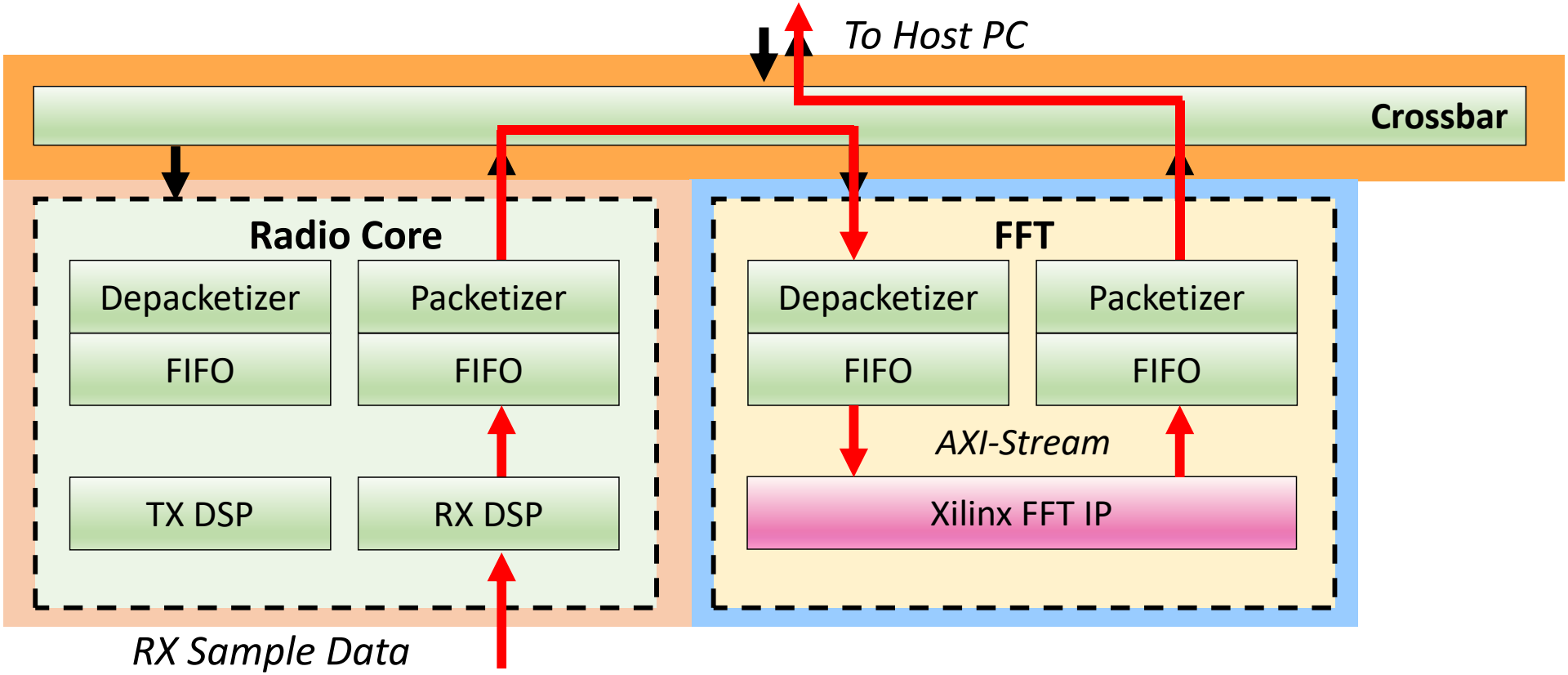
Computation Engine



- User writes their own HDL or drops in IP
  - Xilinx IP, opencores.org, third party IP
  - Vivado HLS, HDL generator software

Computation Engine

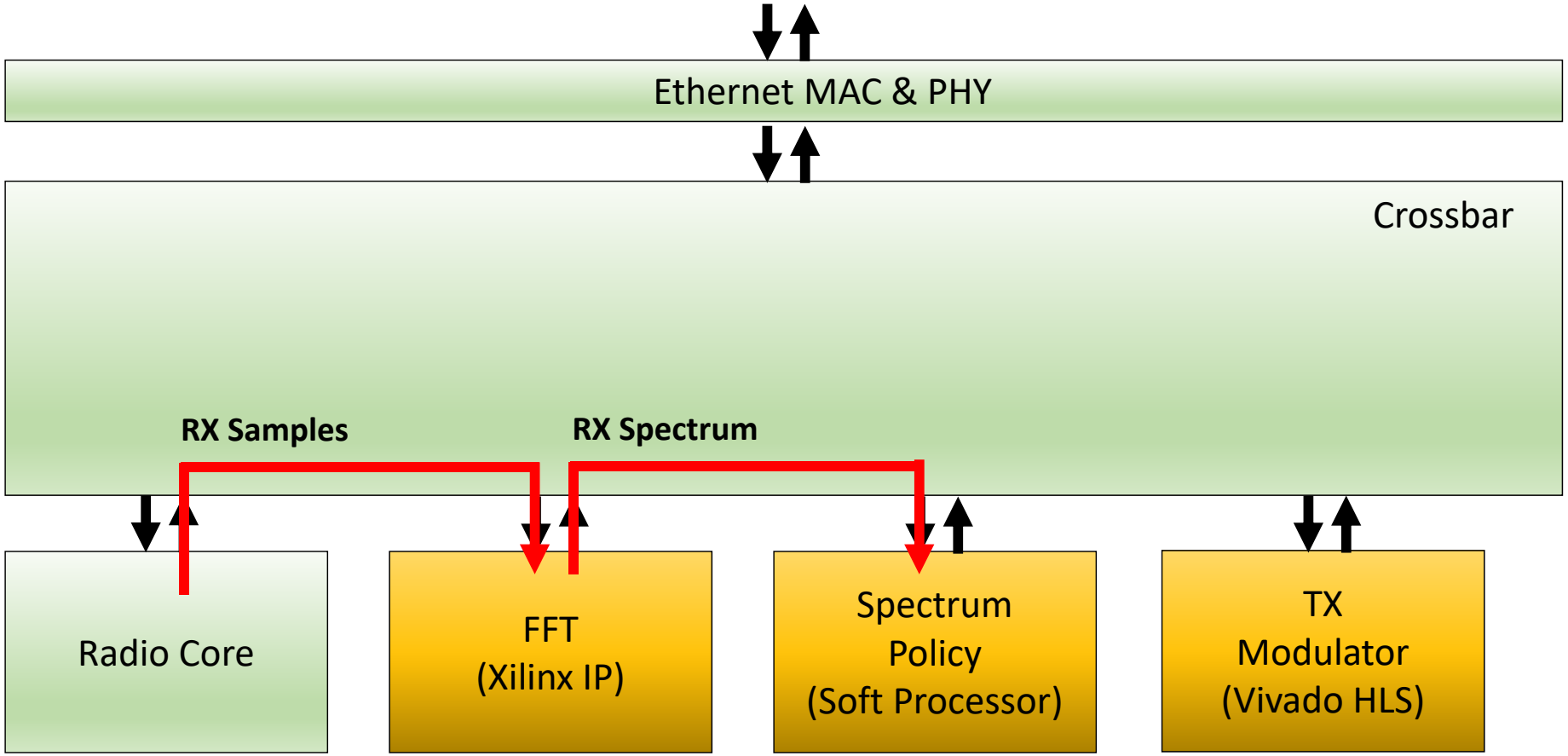
Ettus Research™ A National Instruments Company



- Each block is in their own clock domain
  - Improve block throughput, timing
  - Interface to Crossbar has clock crossing FIFOs

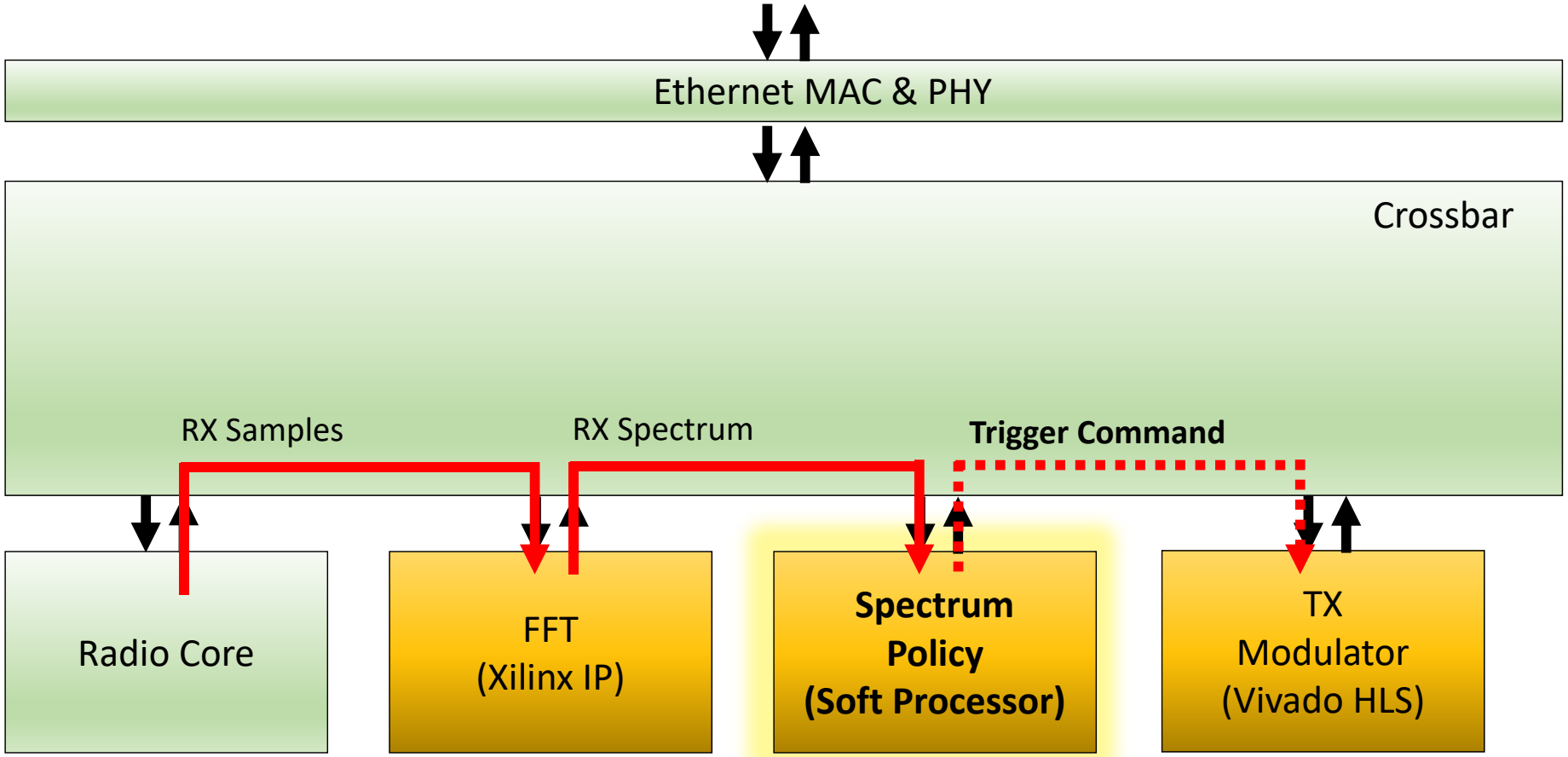
Cognitive Radio

Ettus Research™
A National Instruments Company



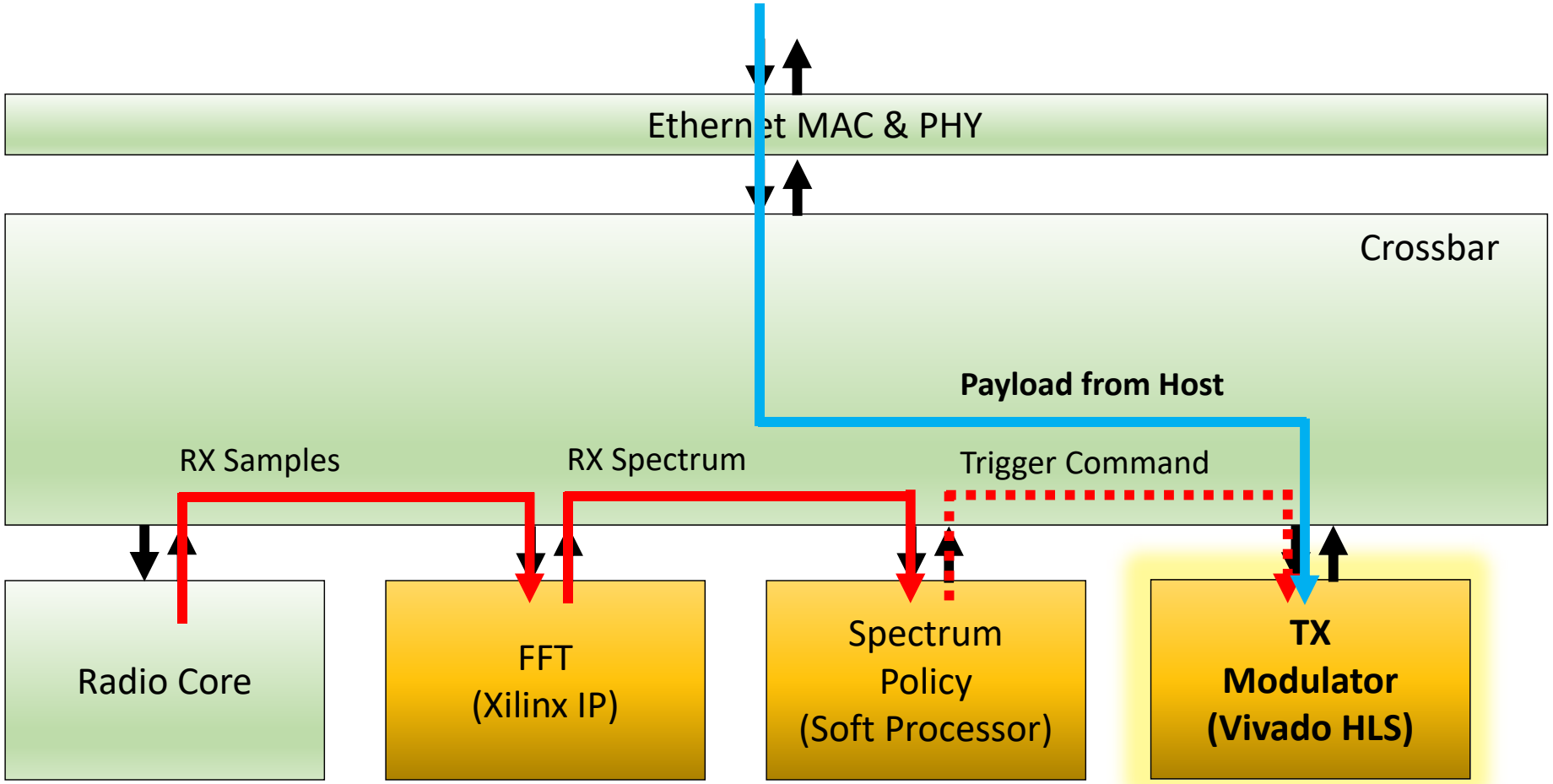
Cognitive Radio

Ettus Research™
A National Instruments Company



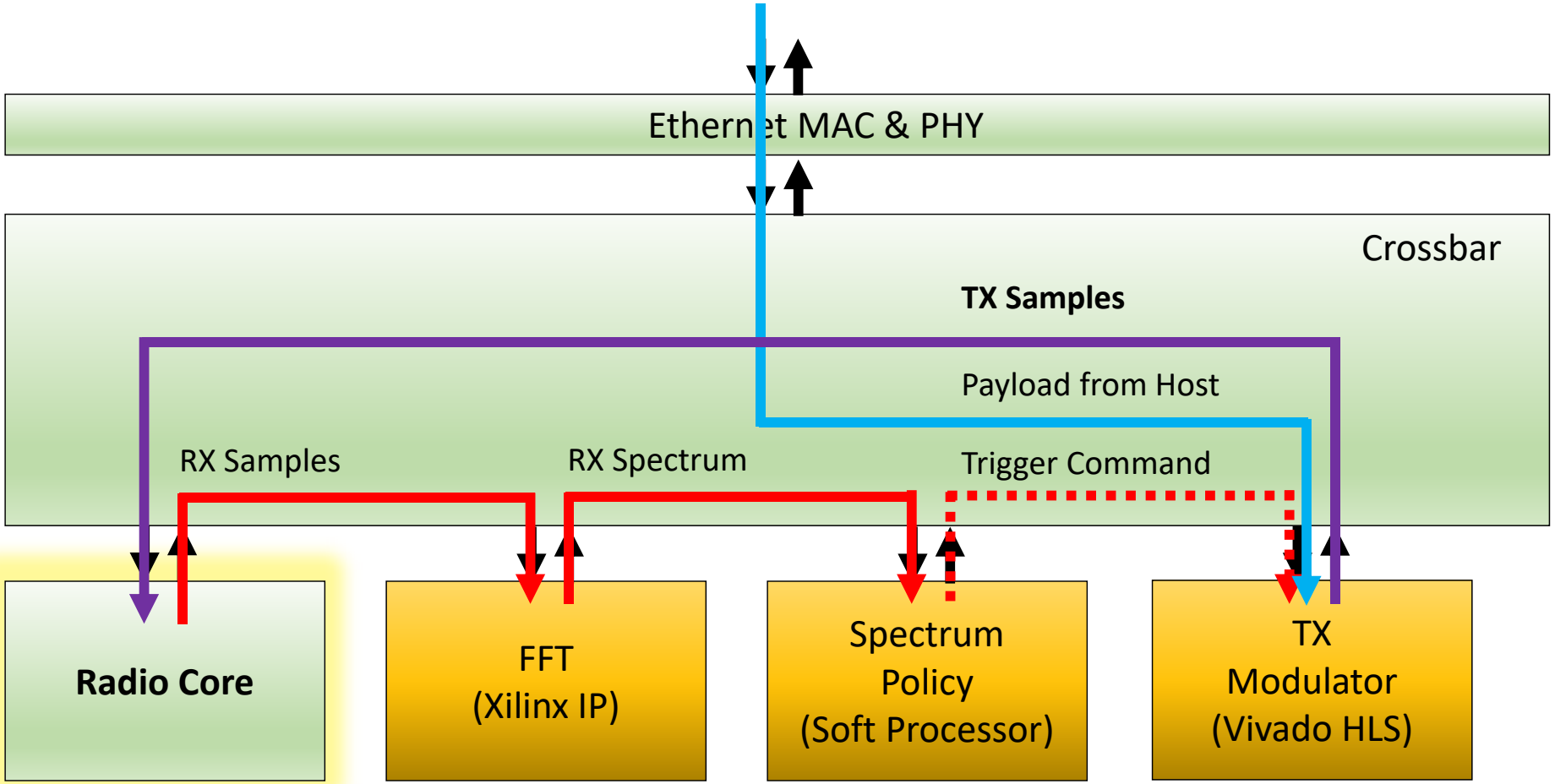
Cognitive Radio

Ettus Research™
A National Instruments Company



Cognitive Radio

Ettus Research™
A National Instruments Company



Summary

Ettus Research™
A National Instruments Company

- Make FPGA acceleration more accessible on USRP
- Tightly integrated with GNU Radio
- Implemented several interesting CEs
  - OFDM, FFT, FIR, Signal Generator, Fosphor
- Portable between all third generation USRPs
  - X3x0, E3xx, N3xx
- Completely open source
- kb.ettus.com/RFNoC_Getting_Started_Guides
- After the break: FPGA & Software Development

Hands on Demos

Ettus Research™
A National Instruments Company

- Open new terminal and run:
  - **source ~/rfnoc-workshop/setup_env.sh**
  - **gnuradio-companion**
- **rfnoc_fosphor**: Wideband Spectrum Display
- **rfnoc_window_fft**: FFT with selectable window
- **rfnoc_fir**: FIR filter with reconfigurable taps
- **rfnoc_ddc**: Receive with radio and DDC block
- Flowgraph directory:
  ~/rfnoc-workshops/src/gr-ettus/examples/