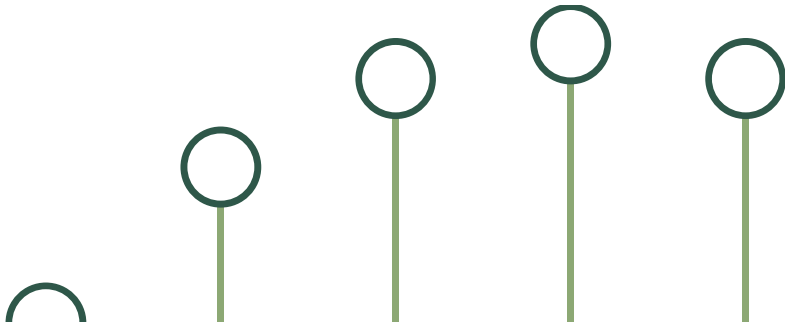
Ettus Research™
A National Instruments Company

FPGAs: **Why, When**, and How to use them (with RFNoC™) – Pt. 1

Martin Braun, Nicolas Cuervo

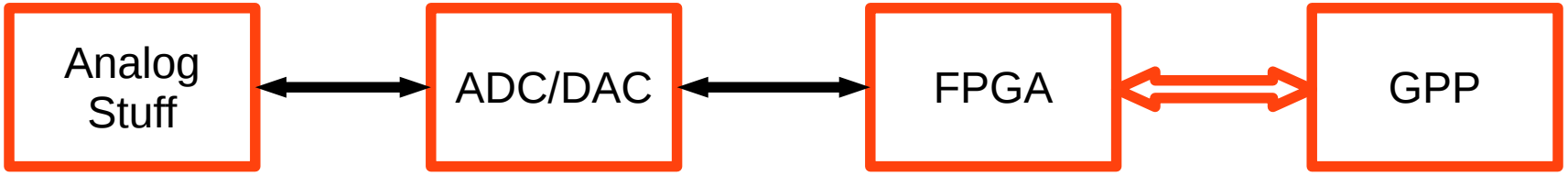
FOSDEM 2017, SDR Devroom



Schematic of a typical SDR

Ettus Research™ A National Instruments Company

- Very rough schematic:

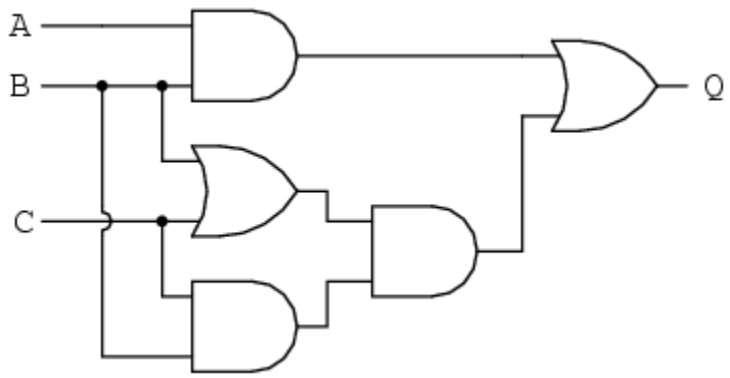


- Let's ignore the analog stuff
- FPGA sits closest to the ADC/DAC
- GPP is separated by some transport (USB, Ethernet, DMA FIFO, or maybe it's just on the same PCB)

What is an FPGA?

Ettus Research™
A National Instruments Company

- Wikipedia: ‘an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable"’
- In SDRs: Effectively a user-definable digital circuit between ADC/DAC and the software
- Can be redefined “any time”, but will take down the circuitry while doing so
- Typical clock rates: several hundred MHz (or more? Or less?)
- Remember these:



How are FPGAs programmed?

Ettus Research™
A National Instruments Company

1. Define your circuitry (shall it filter? Shall it generate UDP packets? Shall it…)

2. Encode that in a format your FPGA toolchain understands (Verilog, VHDL, graphical tools)

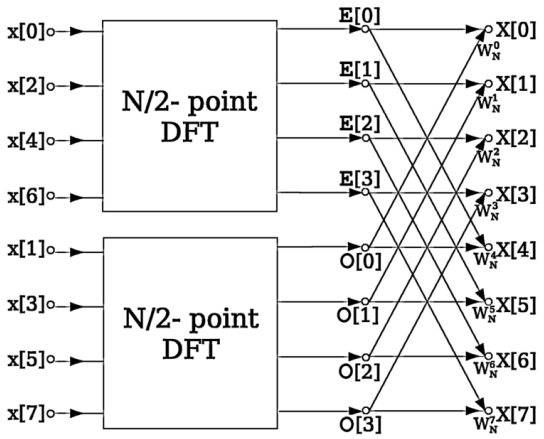
3. Synthesize to netlist + generate bitstream. A bitstream is a binary representation of how the internals of the FPGA is configured. Often proprietary formats.

4. Load bitstream onto FPGA, typically using dedicated pins.

What do we use FPGAs for?

Ettus Research™ A National Instruments Company

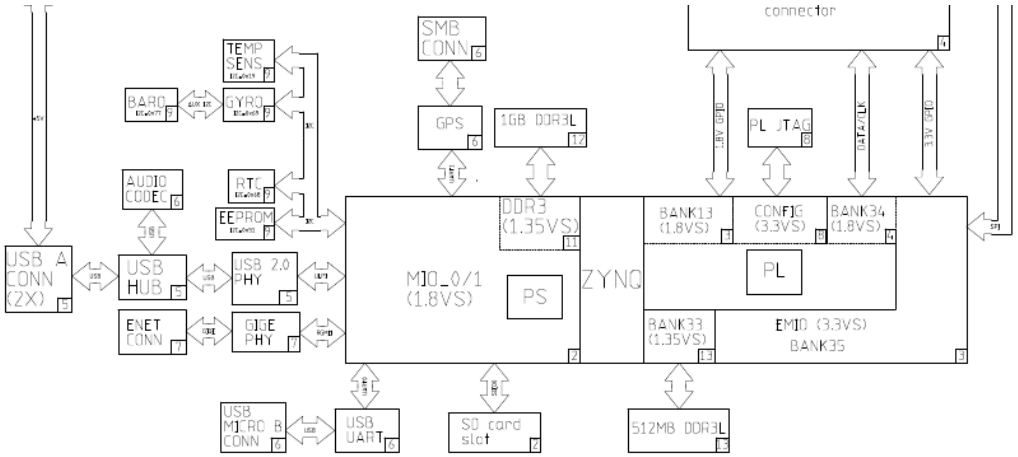
- Can an FPGA run software? Well, it can, but only if you make it look like a CPU. Let's ignore that for now.
- If you can draw a digital circuit, it'll usually work well on an FPGA
- Multiple parallel circuits are also possible, and in fact one of the strengths of FPGAs.
- Latency can be controlled on the order of clock cycles.
- These work well:
  - FIR filters, FFTs, Neural Networks
  - Control loops
- These not so much:
  - Protocol handling, complex rulesets



(Source: https://github.com/Themaister/muFFT/blob/master/doxygen/fft.md)

Flexibility (or lack thereof)

- During “runtime”, the digital circuit can’t be easily replaced
- Building bitfiles can take a long time (depending on the tools, design, and chip between a few seconds and several hours)
- If your FPGA is controlling peripherals, those will be disabled while the FPGA is reprogrammed



(Source: Ettus Research USRP E310 Schematic files.ettus.com/schematics/e310)

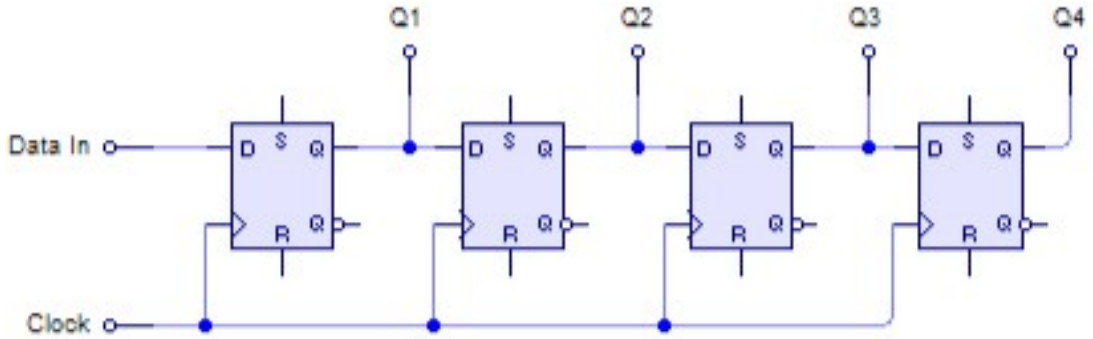
Challenges: Digital Logic

Ettus Research™ A National Instruments Company

- Did you pay attention in school?
- Quick, what's this equation as a digital circuit:

$$(f * g)[n] = \sum_{m=-M}^{M} f[n-m]g[m].$$

- Concepts may seem trivial if you're an EE major, but there's a lot of concepts worth knowing (Types of flip flops, bus arbitration, interface designs, memory architectures, …)
- What does this do?



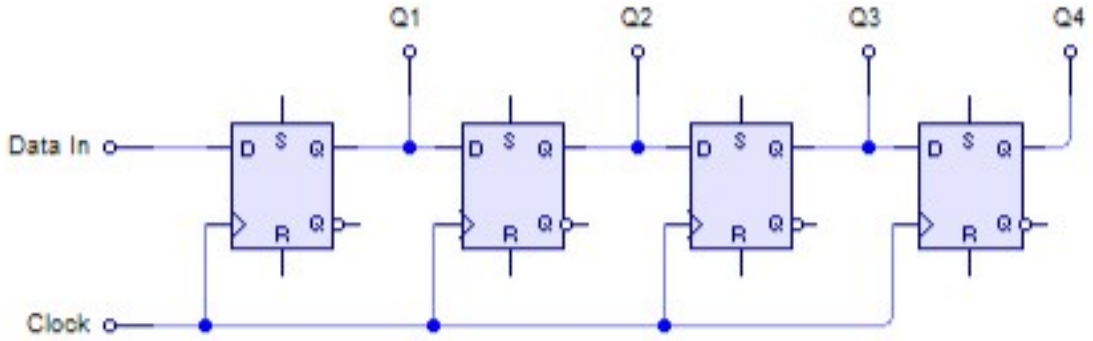
(Source: https://en.wikipedia.org/wiki/Shift_register)

Challenges: Circuit Magic

Ettus Research™
A National Instruments Company

- The digital logic is only half of it
- What kind of constraints are relevant for our SIPO?

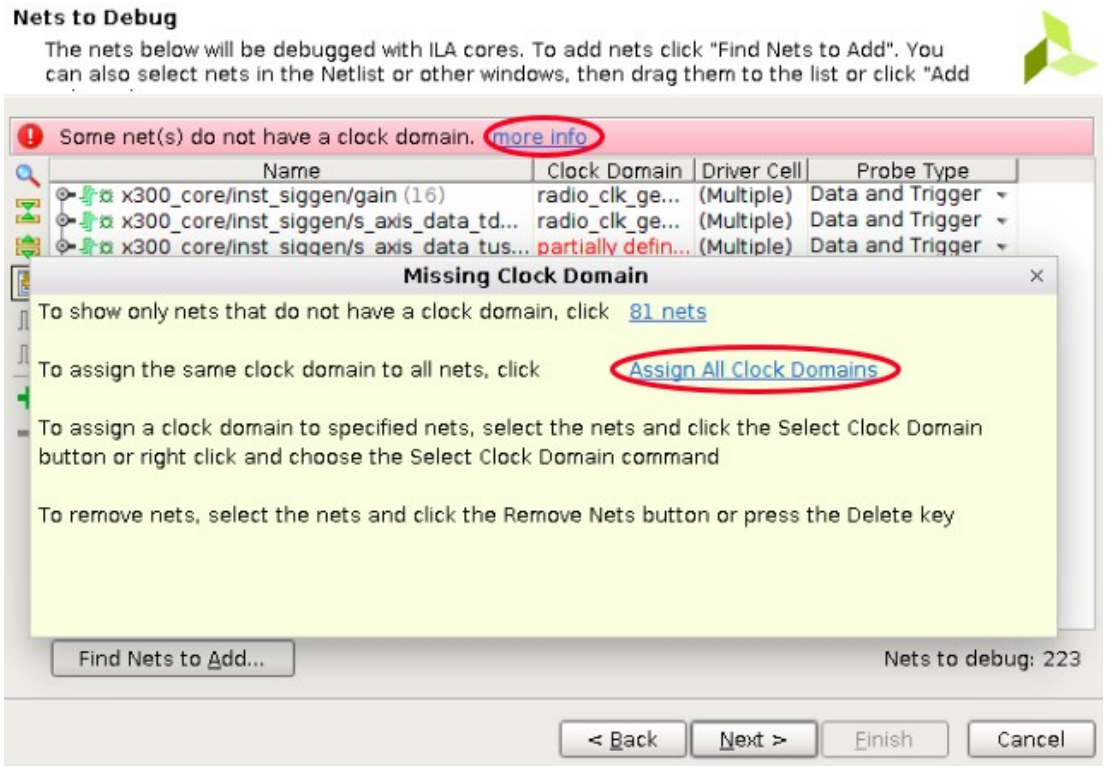
- Where did the clock come from? How fast is it?
- Will the FFs keep up?
- How long do I need to read the outputs?
- Is ‘Data In’ a pin? Are QN pins? Shouldn’t I connect reset lines?



Challenges: Tools

Ettus Research™
A National Instruments Company

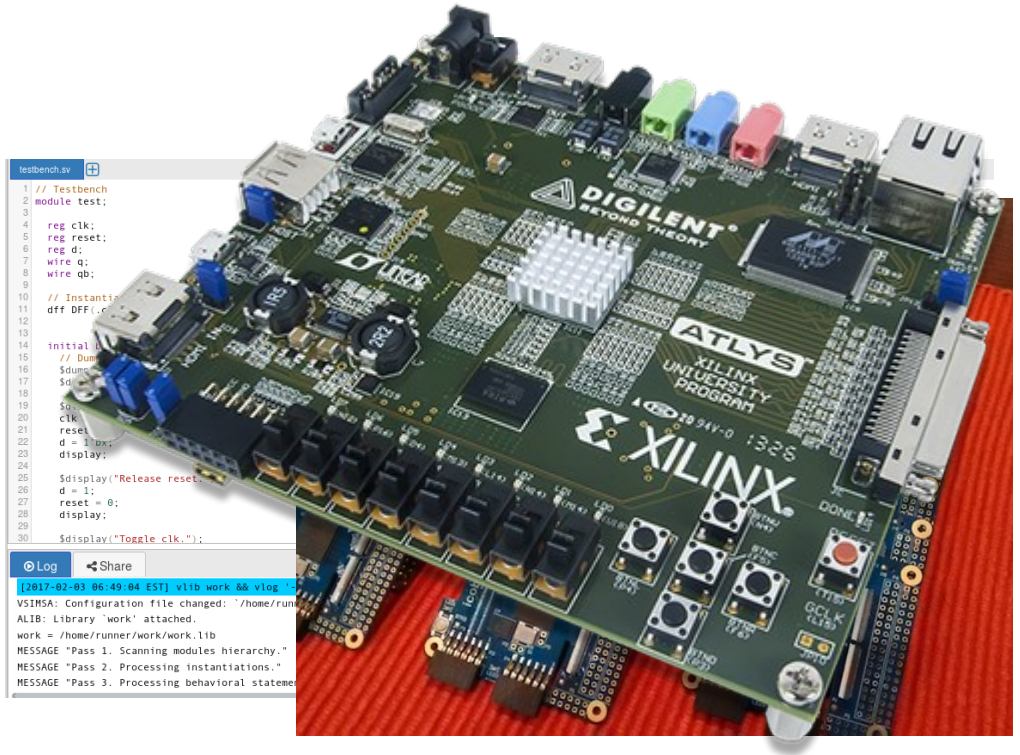
- Most likely, you're leaving the safe, easy confines of running gcc and clang
- You're in for a treat! Good luck getting Vivado running on Gentoo.
- Ever heard of TCL?



Pointers

Ettus Research™
A National Instruments Company

- EDA Playground: Play around with Verilog in your browser
- Yosys, Icoboard: RPi, free software
- Xilinx, Altera have eval kits e.g. from Digilent
- USRPs will let you do SDR



RF-Network-on-Chip (RFNoC)

If you only remember one slide…

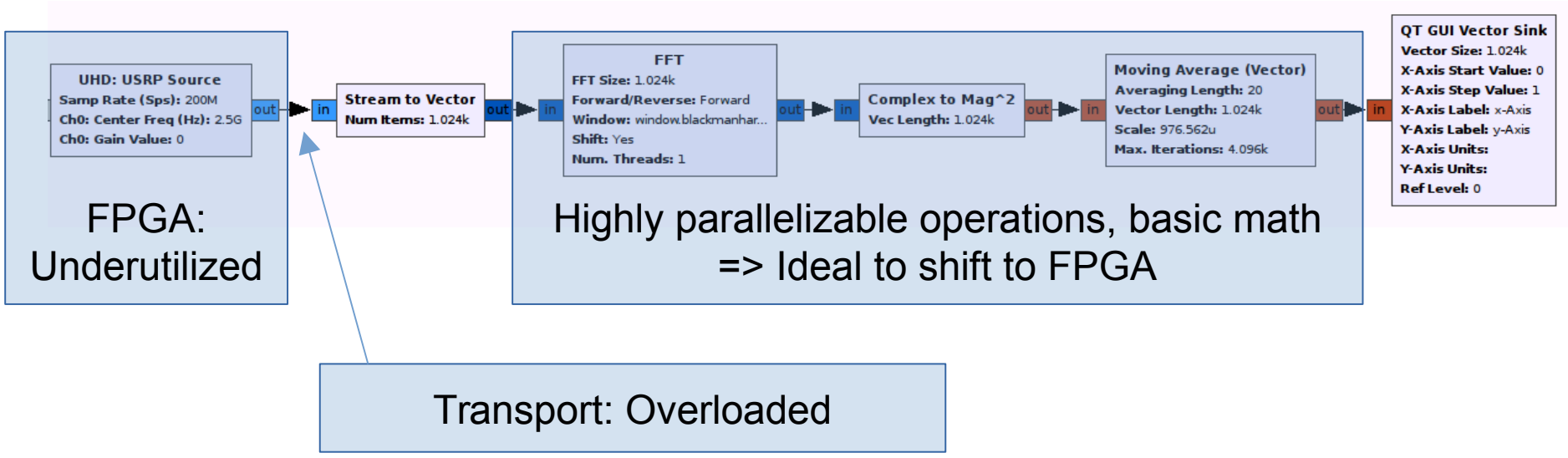
Ettus Research™ A National Instruments Company

RFNoC is for FPGAs is what GNU Radio (currently) is for GPPs.

	RFNoC	GNU Radio
Provides Easy-to-use Infrastructure for SDR applications	✓	✓
Handles Data Movement between blocks	✓ (AXI-Based)	✓ (Circular Buffers)
Takes care of boring and recurring tasks	✓ (Flow control, addressing, routing)	✓ (R/W pointer up-dating, tag handling…)
Provides library of blocks to get started	✓ (Growing)	✓ (Huge and well-tested)
Works with GNU Radio Companion	✓ (Through gr-ettus)	✓ (Built-in)
Well-documented	✓ (Right?)	✓ (Right? RIGHT?)
Writes your blocks for you	✗	✗

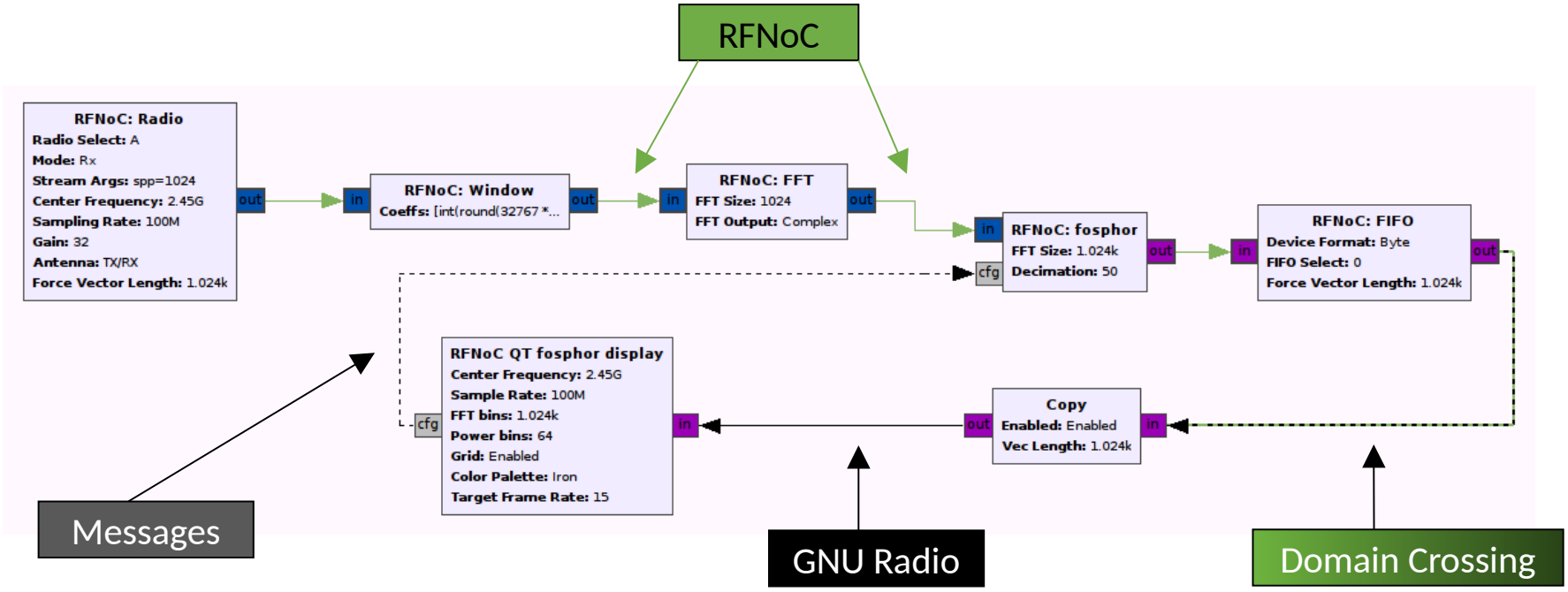
Example: Wideband Spectral Analysis

- Simple in Theory: 200 MHz real-time, Welch's Algorithm
- In practice: Several stumbling blocks. That's the problem RFNoC is trying to solve.



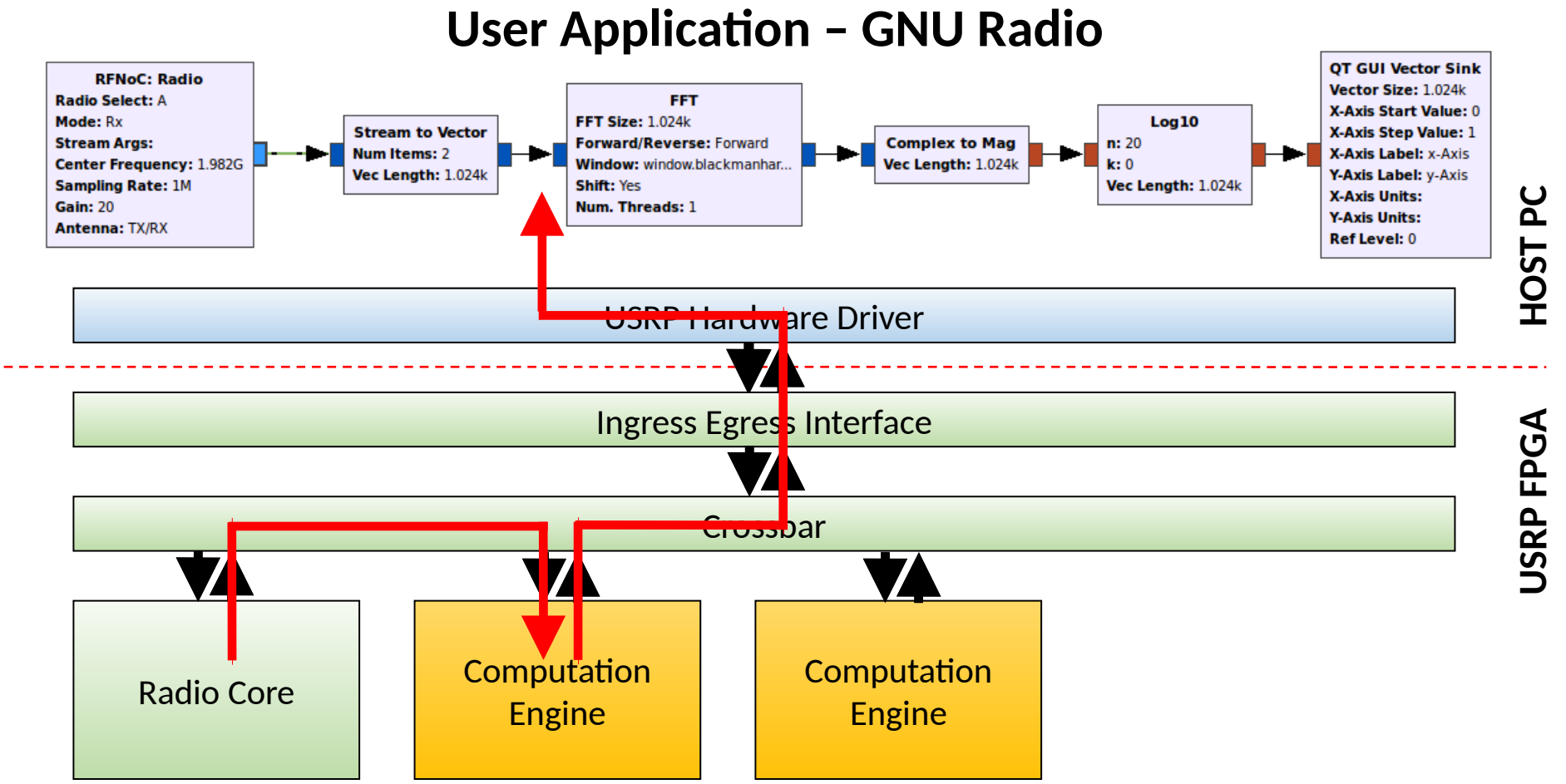
Example: E310 + fosphor

- RFNoC + GNU Radio: Work nicely together
  - Ideal way to use and test RFNoC is with GNU Radio
- Data is passed between "domains" easily

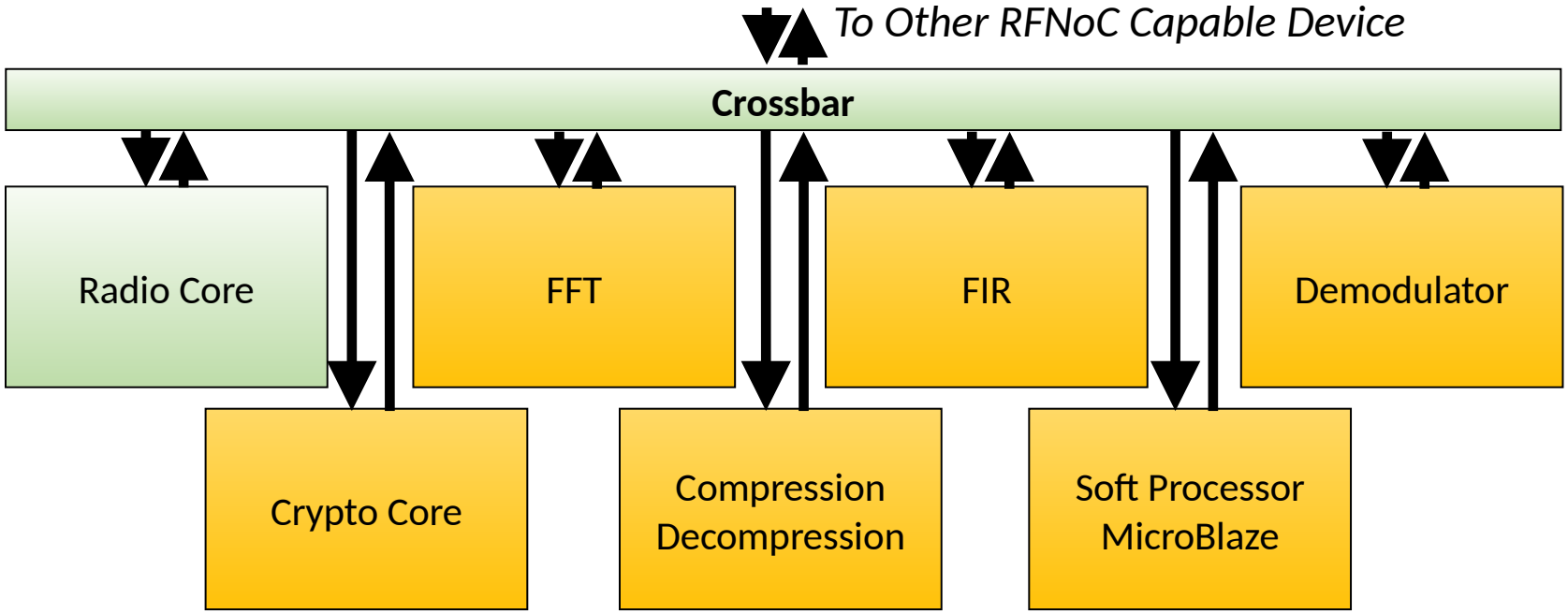


RFNoC Architecture

Ettus Research™
A National Instruments Company

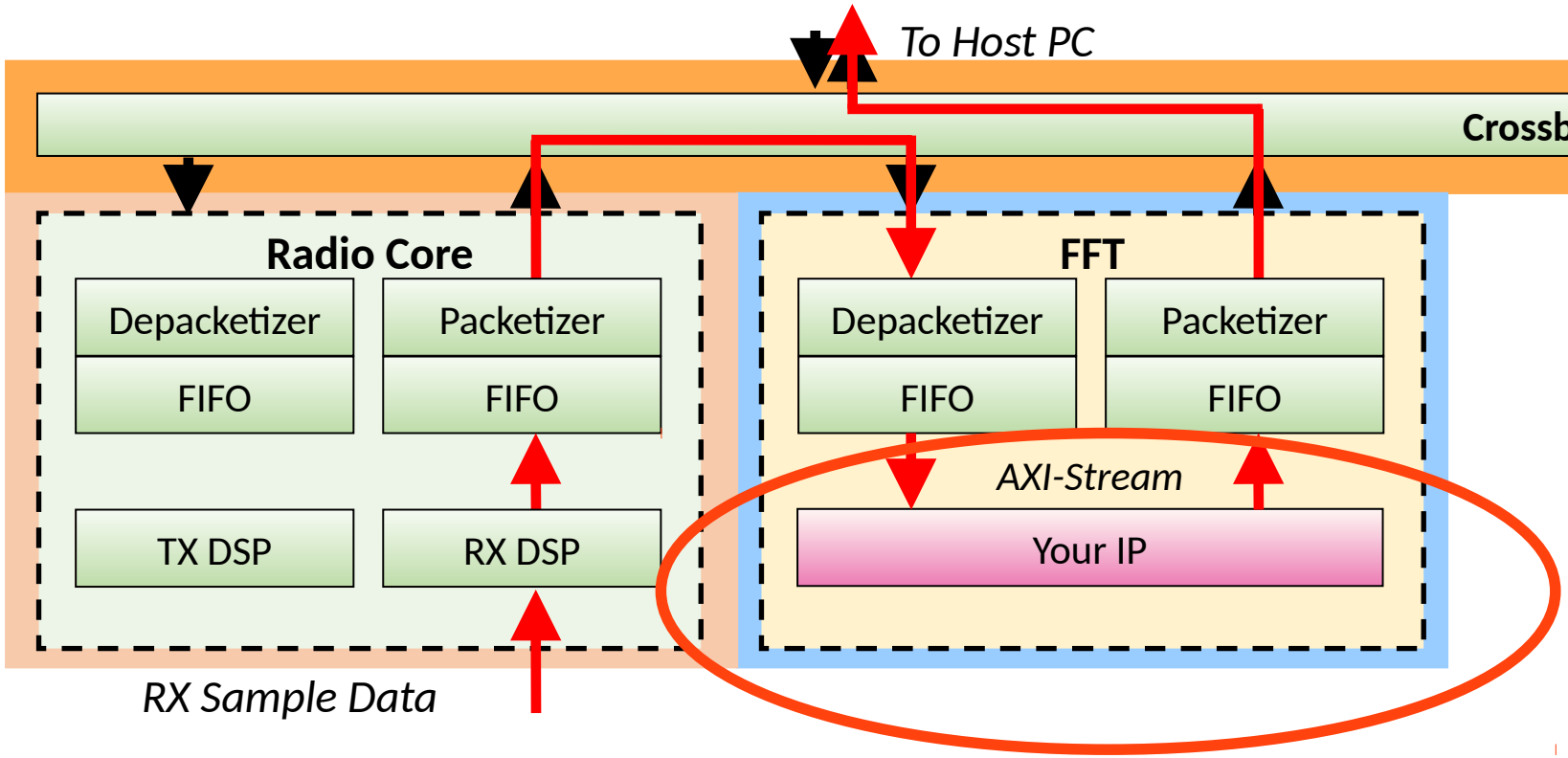


Device Configuration



- Blocks are chosen when bitfile is generated

Anatomy of an RFNoC Block



- Blocks are separate entities
  - Separate clock domain
  - Optimized for developing separately