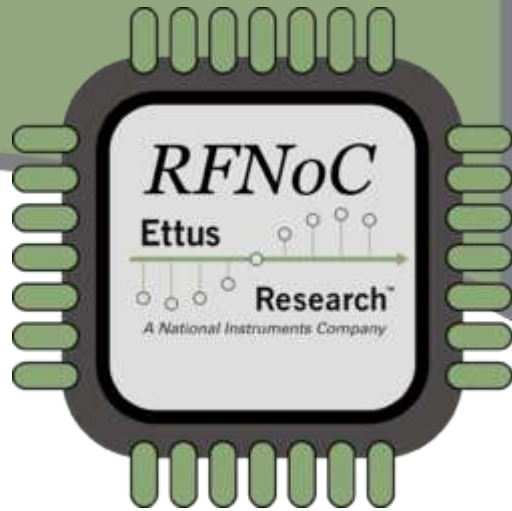
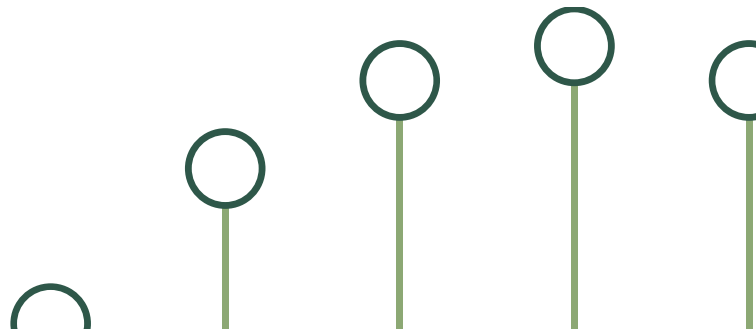


WinnComm Europe 2016
RFNoC™ Tutorial: Application Development

Tim Fountain
e: tim.fountain@ni.com
p: +1-503-720-2456



Agenda

Ettus Research™
A National Instruments Company

- Tool flow
- FPGA Development
- Simulation
- Host Development
- Discussion – Digital Gain

Getting Started with RFNoC

Ettus Research™ A National Instruments Company

- PyBOMBS - http://pybombs.info/
  - Python Build Overlay Managed Bundle System
  - GNU Radio install management system for resolving dependencies and pulling in out-of-tree projects.
- Install UHD with pybombs or manually build from source

  $ git clone --recursive -b rfnoc-radio-redo https://github.com/EttusResearch/uhd.git
- Install GNU Radio (optional)

  $ git clone --recursive -b maint https://github.com/gnuradio/gnuradio.git
- Install GR-ETTUS (optional)

  $ git clone -b radio-redo https://github.com/EttusResearch/gr-ettus.git

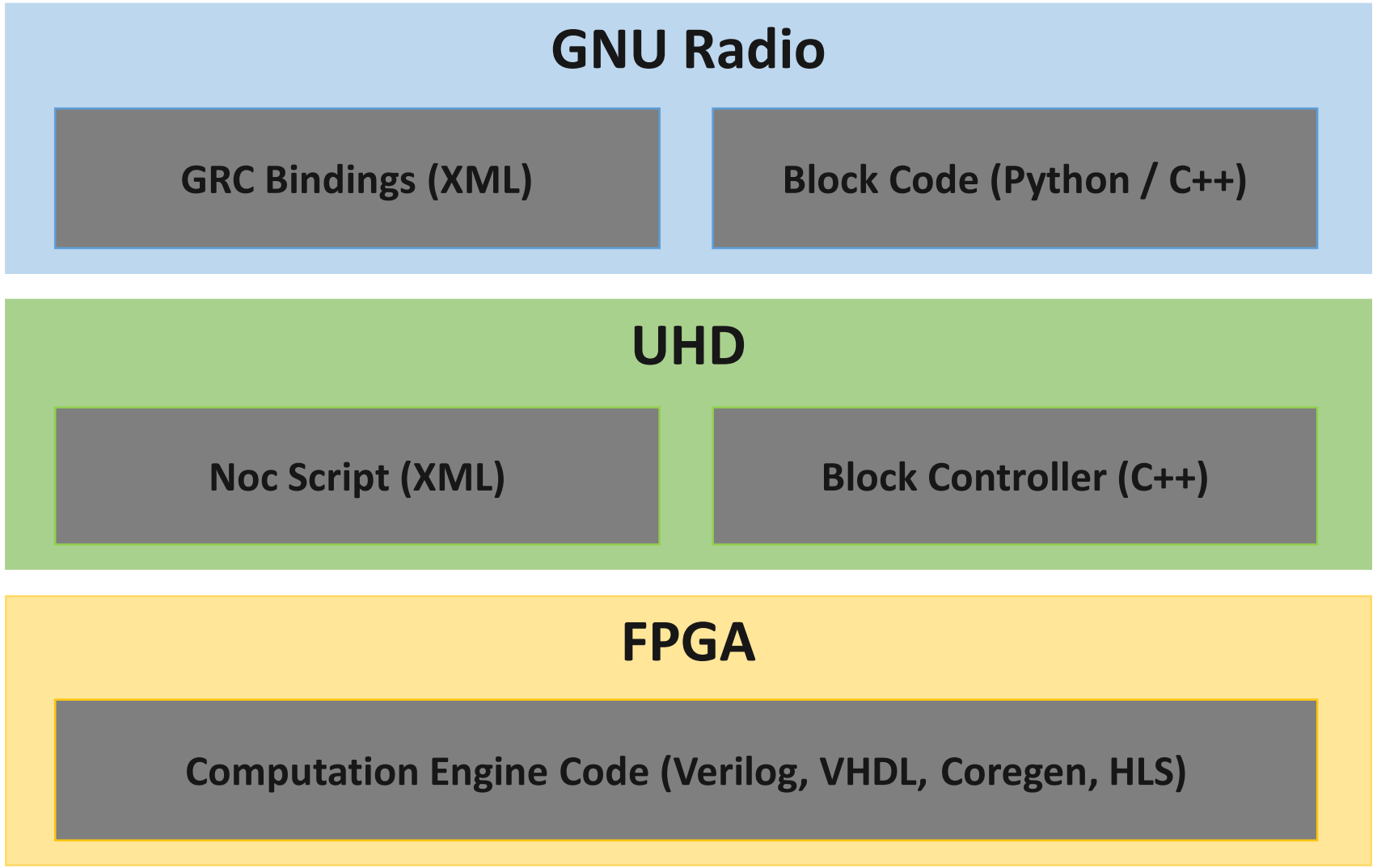
- Detailed getting started guide:

  https://kb.ettus.com/Getting_Started_with_RFNoC_Development
- See UHD Manual for UHD prerequisites:

  http://files.ettus.com/manual/page_build_guide.html

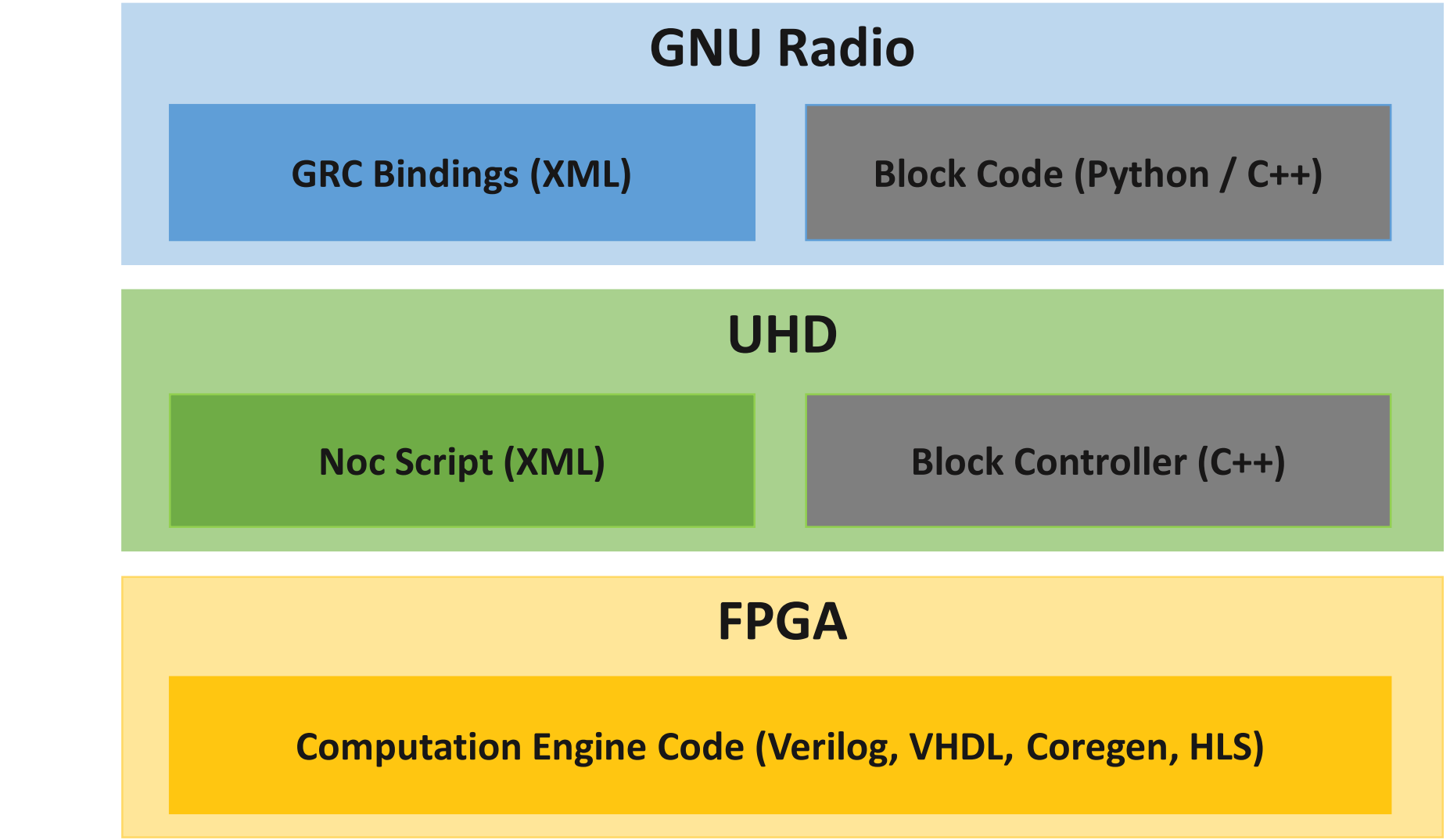
Use Existing Computation Engine(s)

Ettus Research™ A National Instruments Company



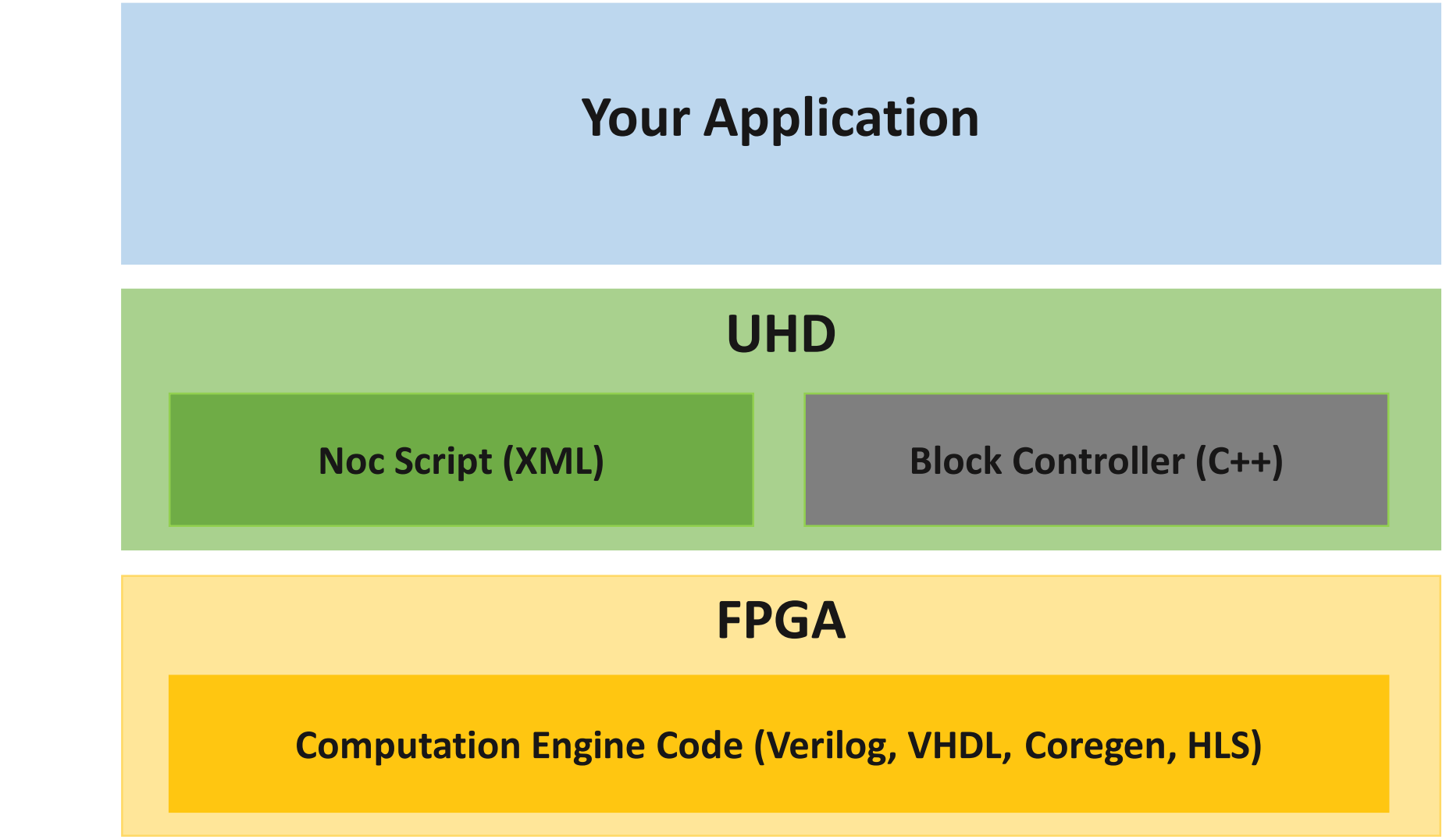
Integration of FPGA IP Into GNU Radio

Ettus Research™ A National Instruments Company



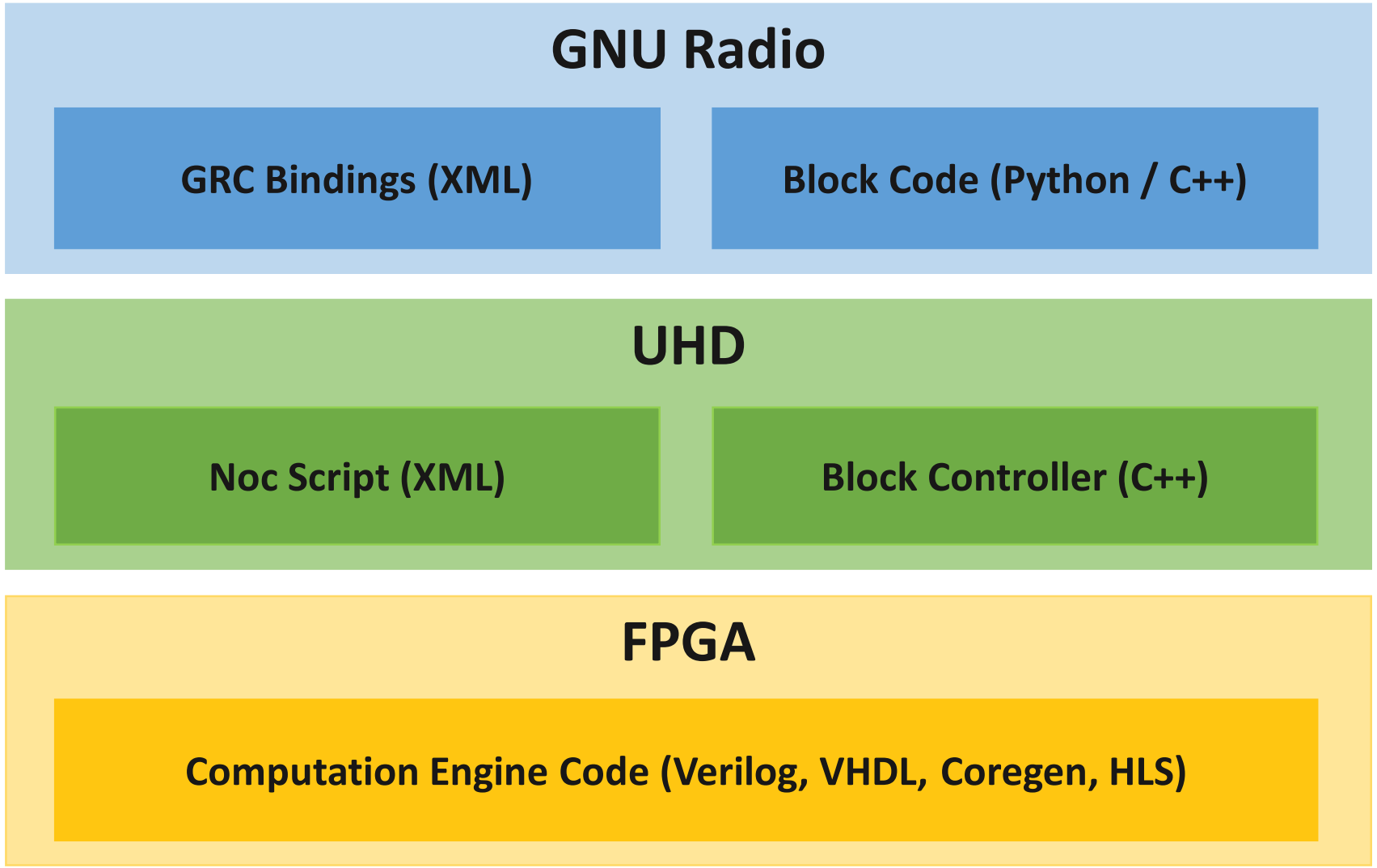
Integration of FPGA IP with custom host application

Ettus Research™ A National Instruments Company



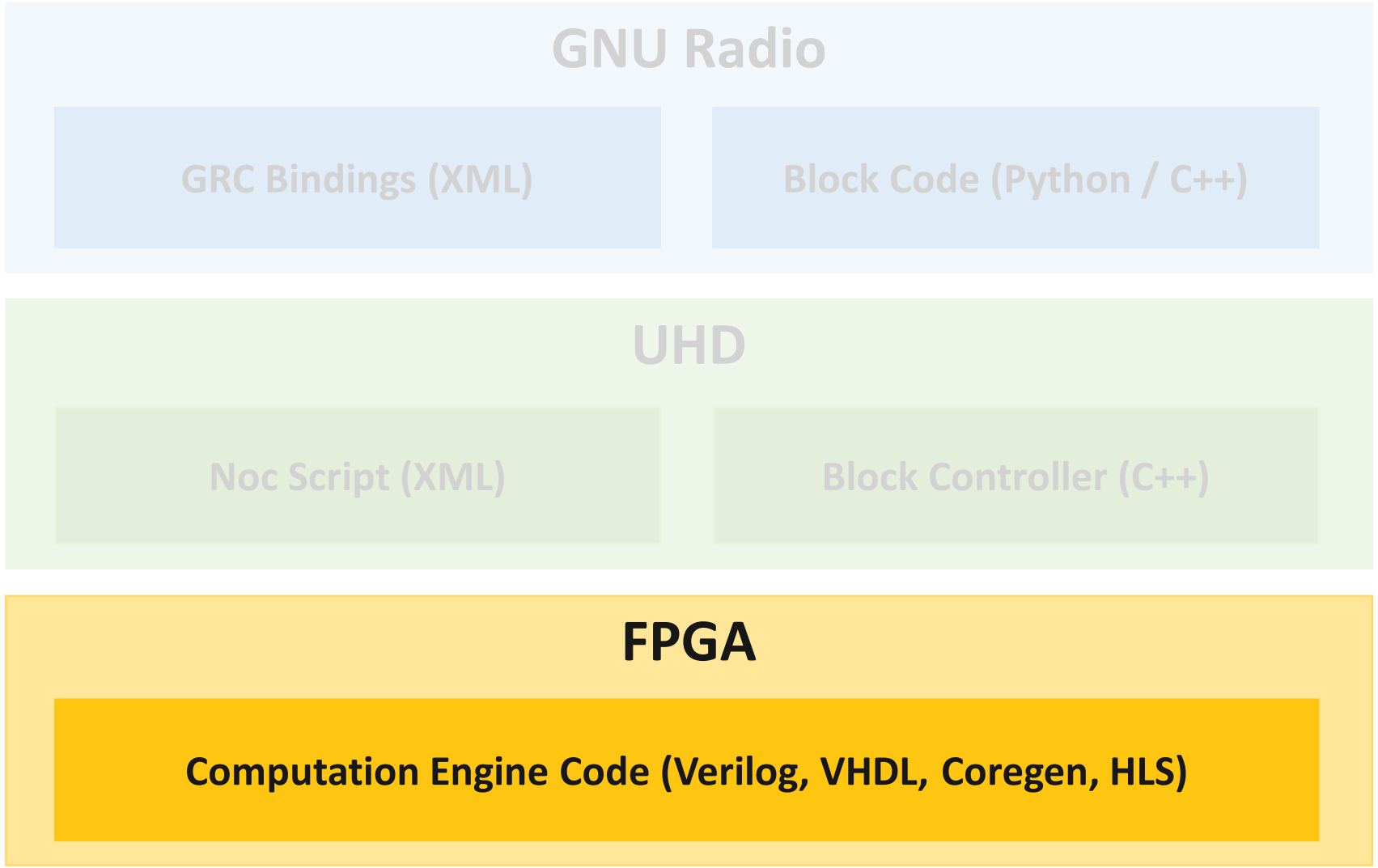
Complete Customization of RFNoC Framework with GNU Radio

Ettus Research™ A National Instruments Company

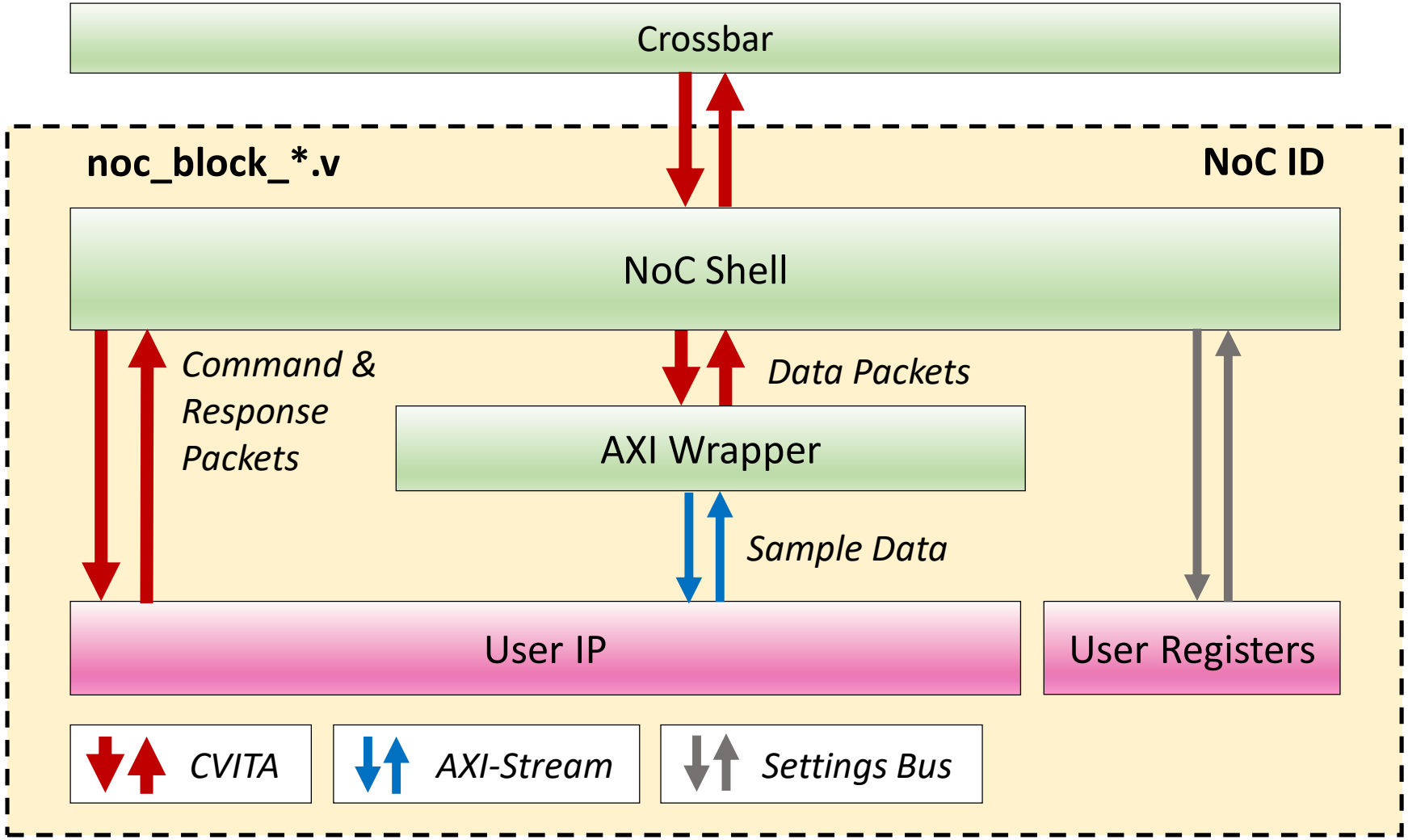


RFNoC Framework

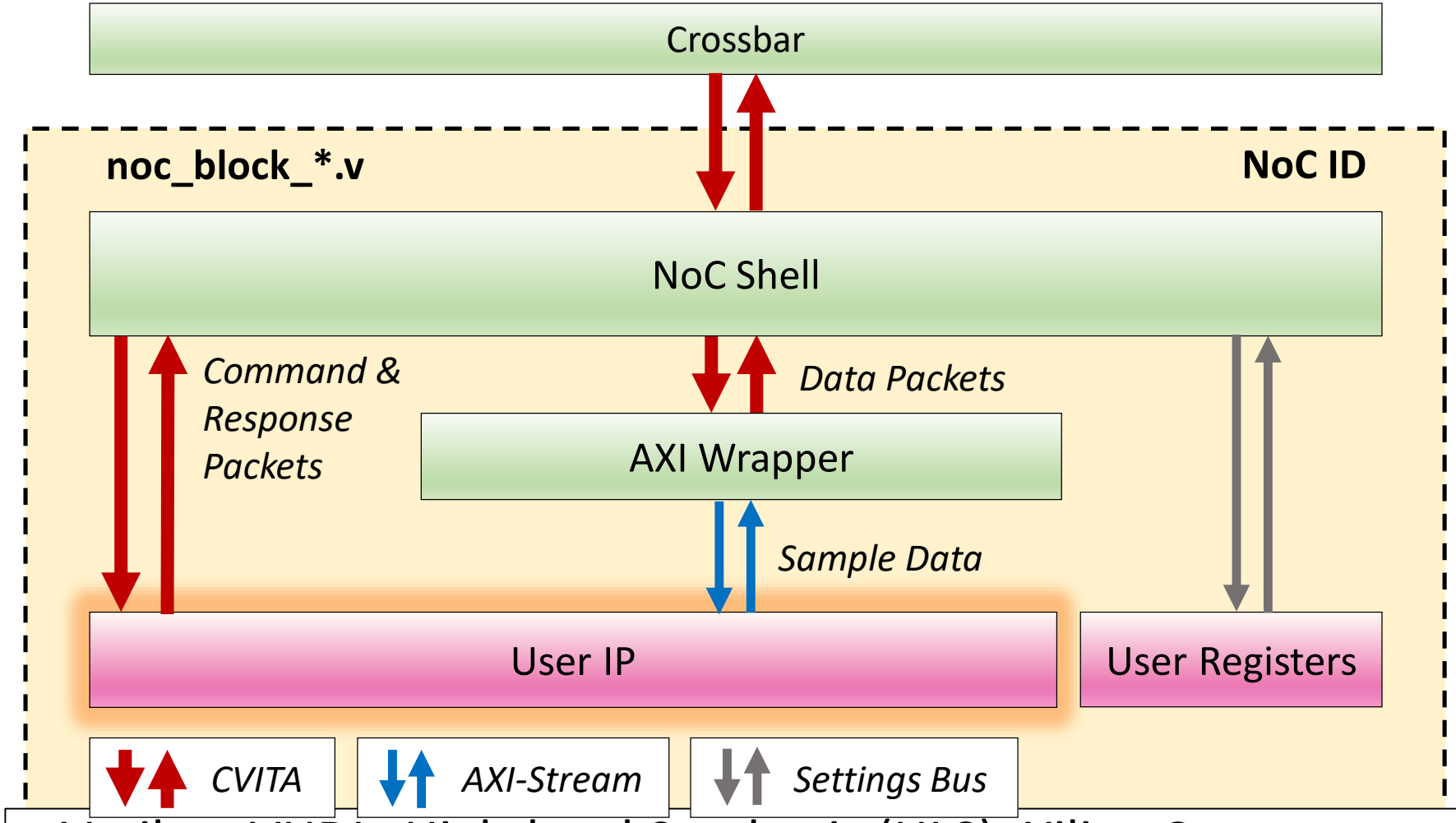
Ettus Research™
A National Instruments Company



Anatomy of a Computation Engine



Anatomy of a Computation Engine



- Verilog, VHDL, High-level Synthesis (HLS), Xilinx Coregen, Third Party IP

CVITA Packet Protocol

Ettus Research™ A National Instruments Company

63 - 62	61	60	59 - 48	47 - 32	31 - 16	15 - 0
Pkt Type	Has Time	EOB	Seq #	Length (in bytes)	SRC SID	DST SID
Fractional Time (Optional, Has Time = 1)						
Payload						
…						

- Packet type based on bits 63, 62, & 60

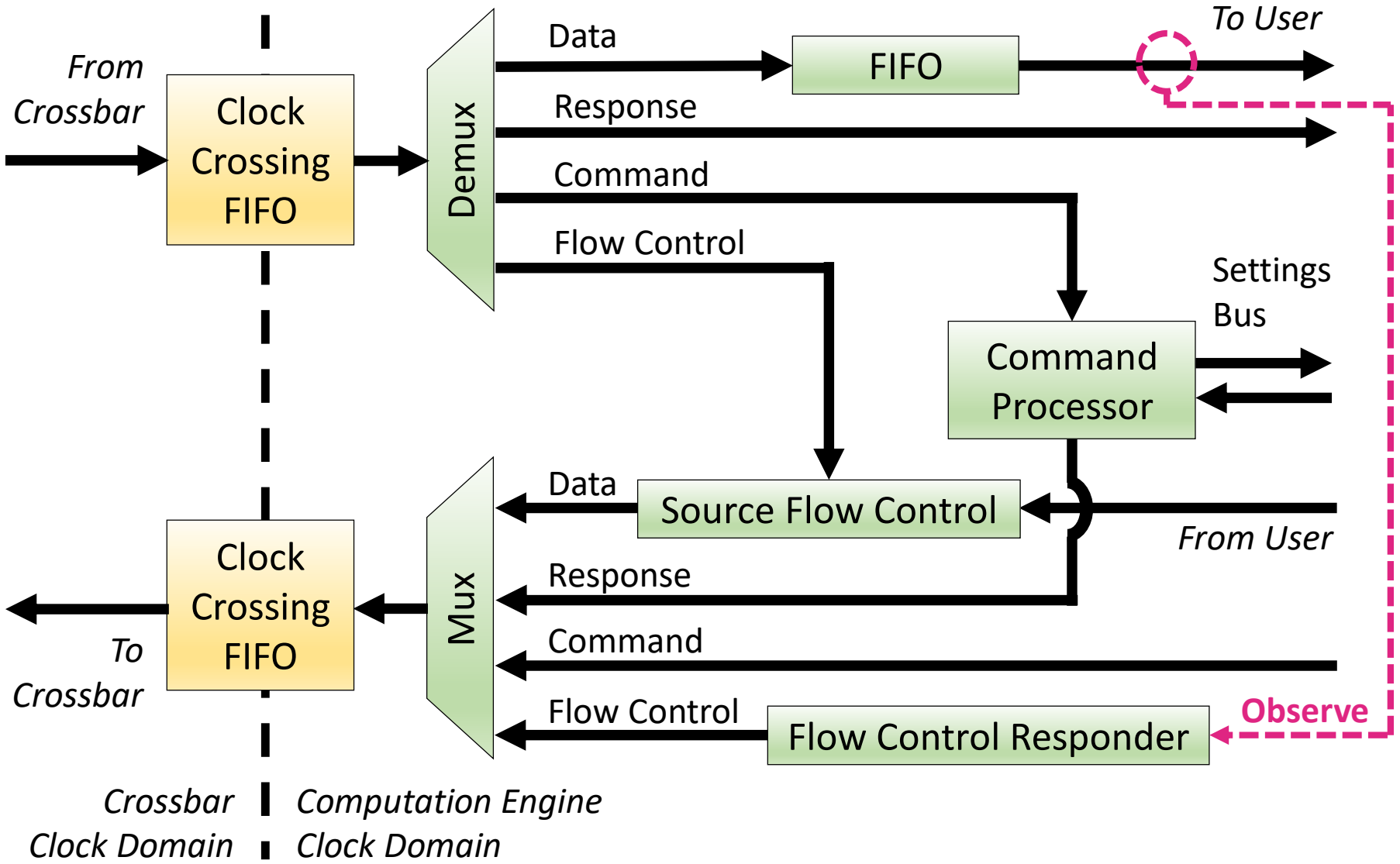
63	62	60	Packet Type
0	0	0	Data
0	0	1	Data (End of Burst)
0	1	0	Flow Control
1	0	0	Command
1	1	0	Response
1	1	1	Response (Error)

Stream IDs

SID		
Crossbar	Crossbar Port	Block Port
15 - 8	7 - 4	3 - 0

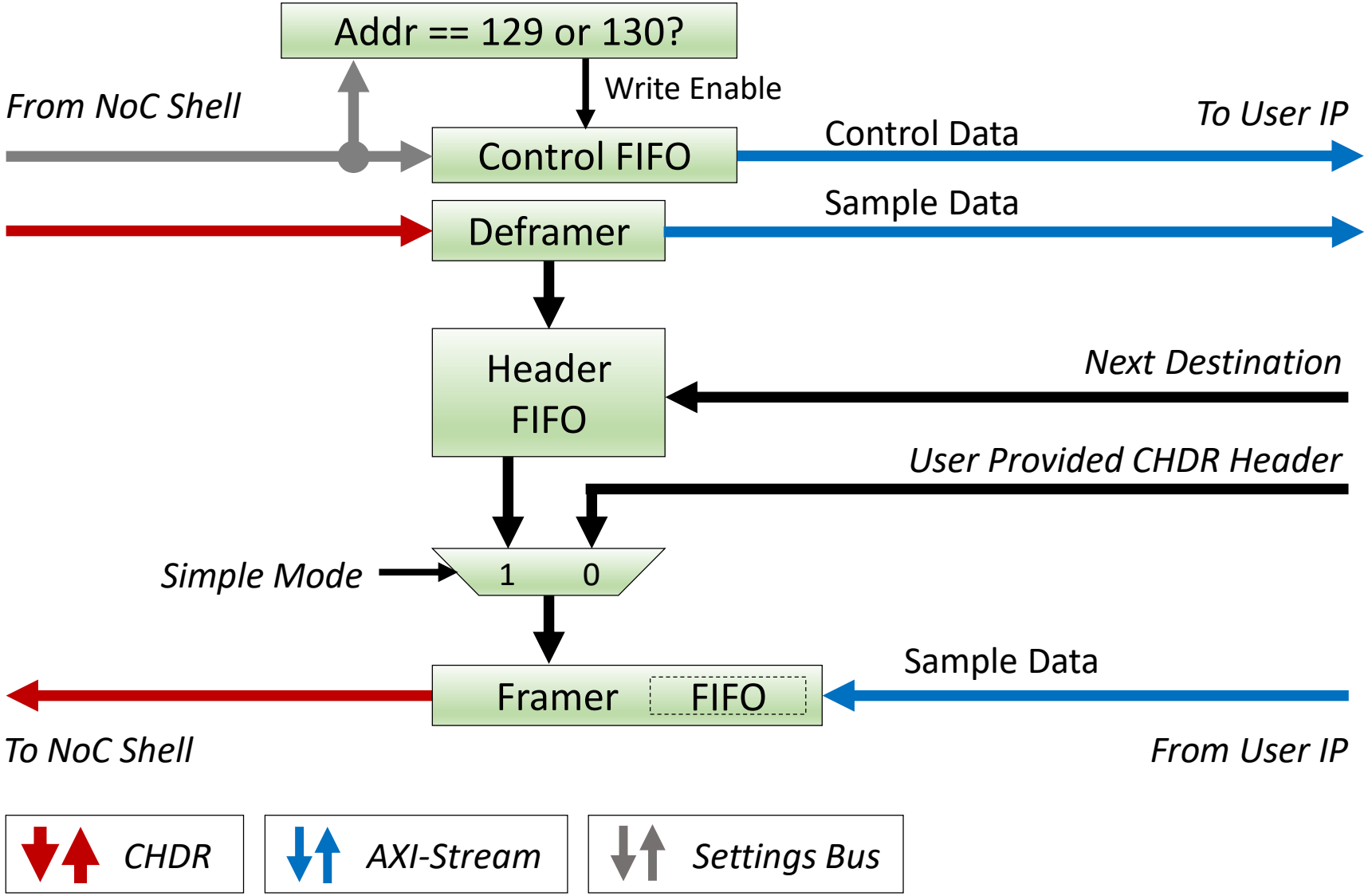
- 16 bit Stream ID
  - 256 unique crossbar (or device) IDs
  - 16 ports per crossbar
  - 16 (logical) ports per block
- Example Crossbar ID: 2, Port: 1, Block Port: 0
  SID: 2.1.0

NoC Shell


From Crossbar
Clock Crossing FIFO
Demux
Data
FIFO
To User
Response
Command
Flow Control
Settings Bus
Command Processor
Data
Source Flow Control
From User
Clock Crossing FIFO
Mux
Response
To Crossbar
Command
Flow Control
Flow Control Responder
Observe
Crossbar Clock Domain
Computation Engine Clock Domain

AXI Wrapper

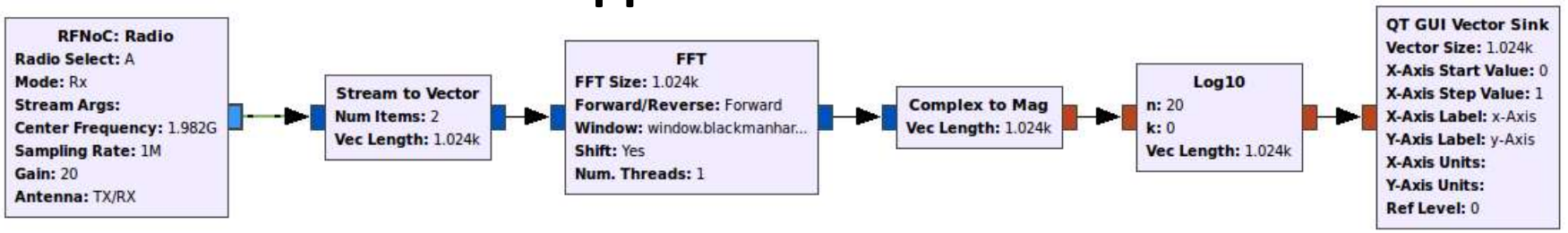
Ettus Research™ A National Instruments Company

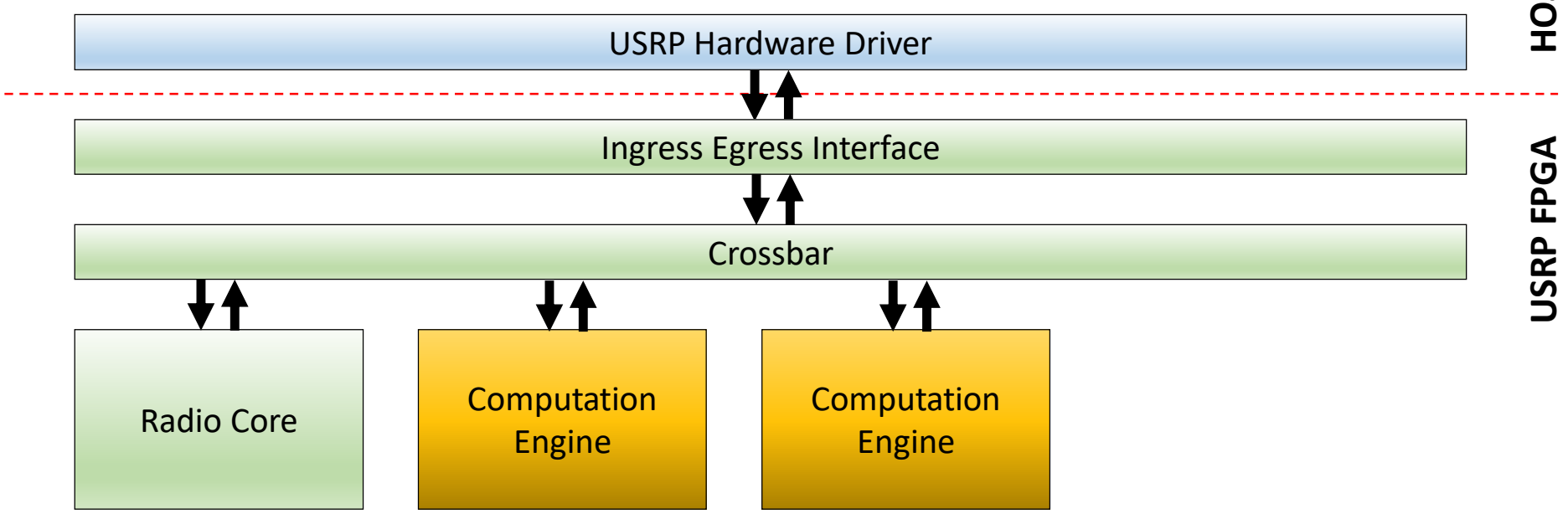


FFT Computation Engine

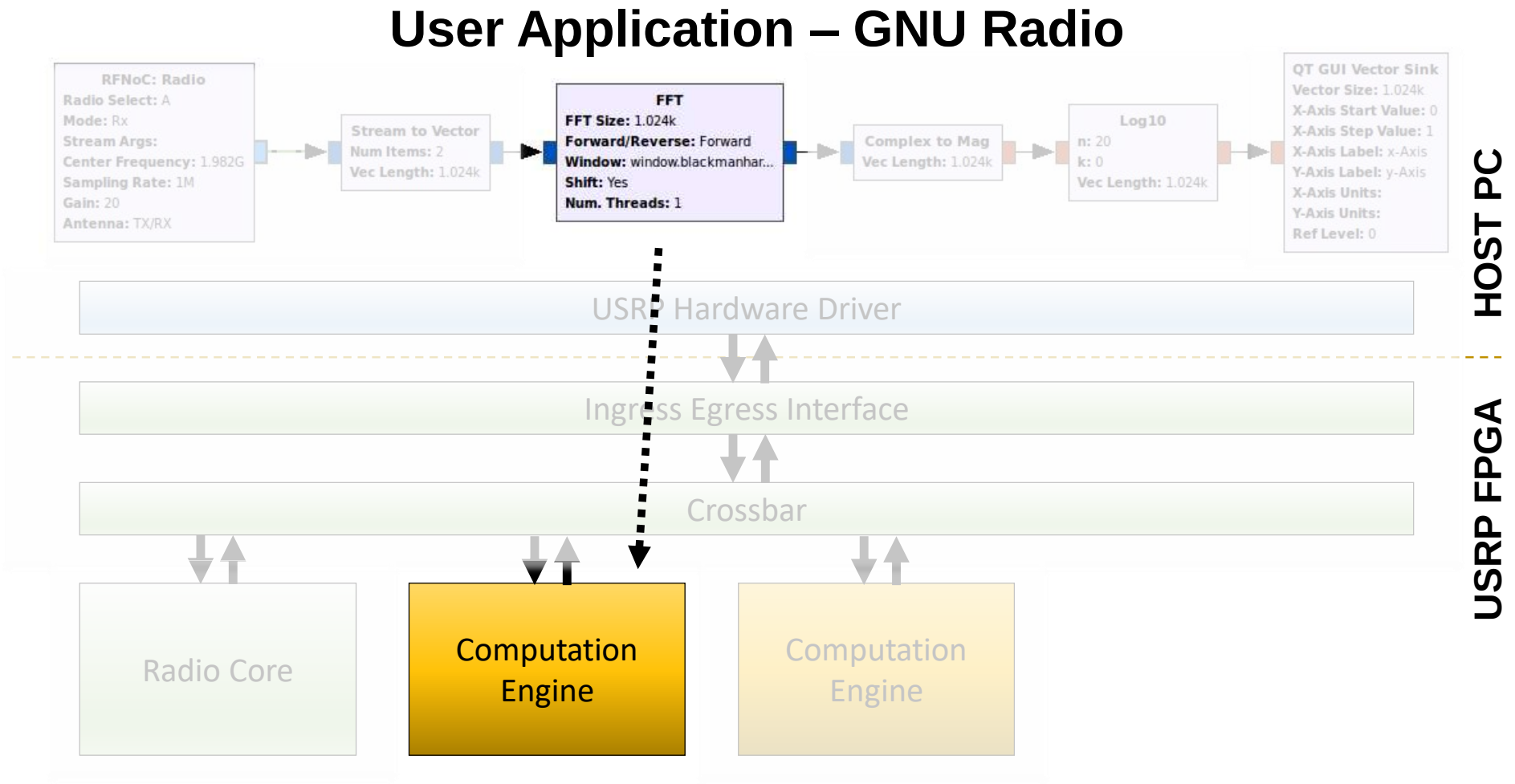
Ettus Research™ A National Instruments Company

User Application – GNU Radio





FFT Computation Engine

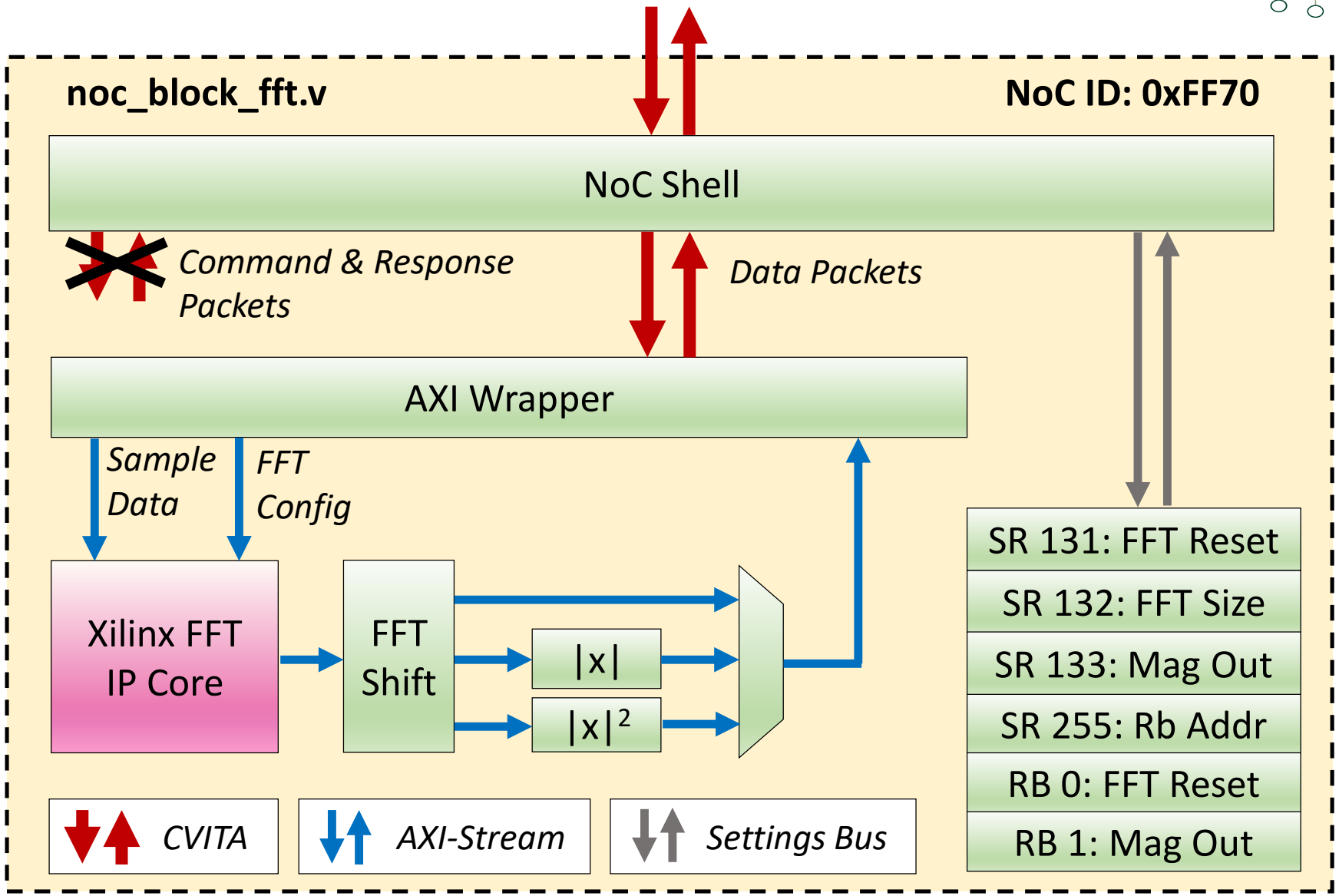


NoC Block FFT Walkthrough

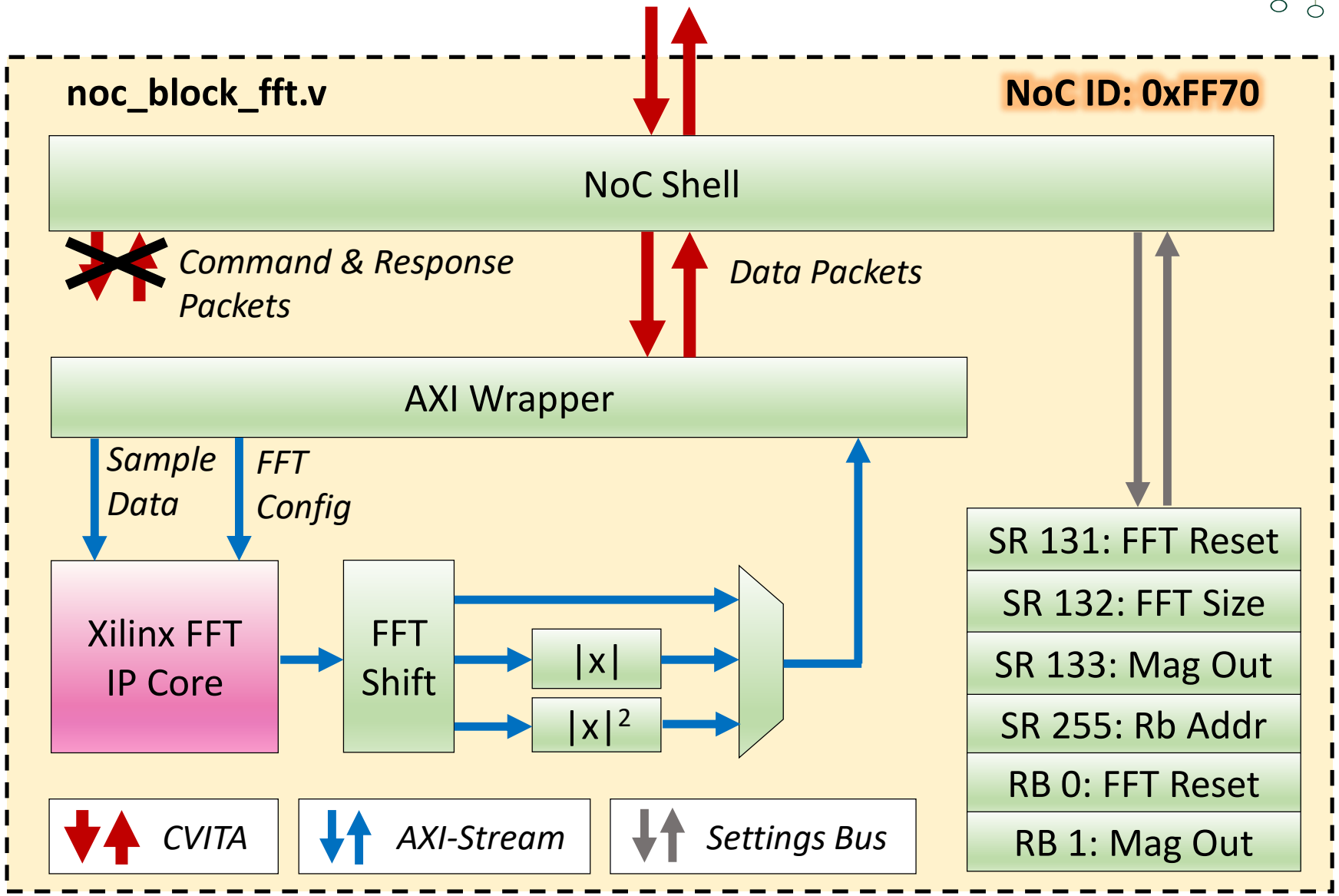
Ettus Research™
A National Instruments Company

- <UHD>/fpga-src/usrp3/lib/rfnoc/noc_block_fft.v
- Testbench: noc_bloc_fft_tb/noc_block_fft_tb.sv
- Based on Xilinx AXI-Stream FFT core
- FFT size 32 – 4096
- Configurable output
  - Complex, Magnitude, Magnitude Squared
- Built in FFT shift

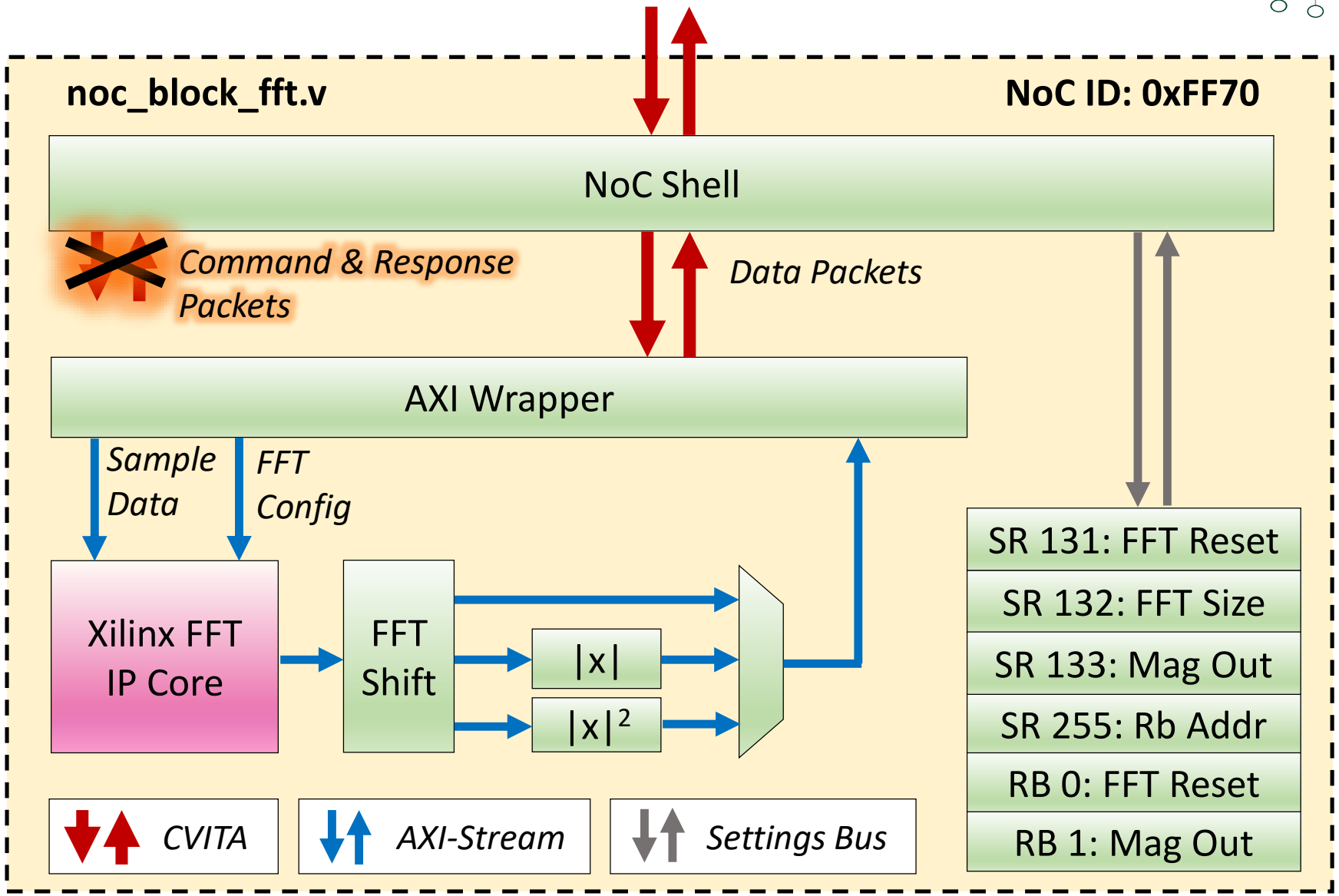
NoC Block FFT



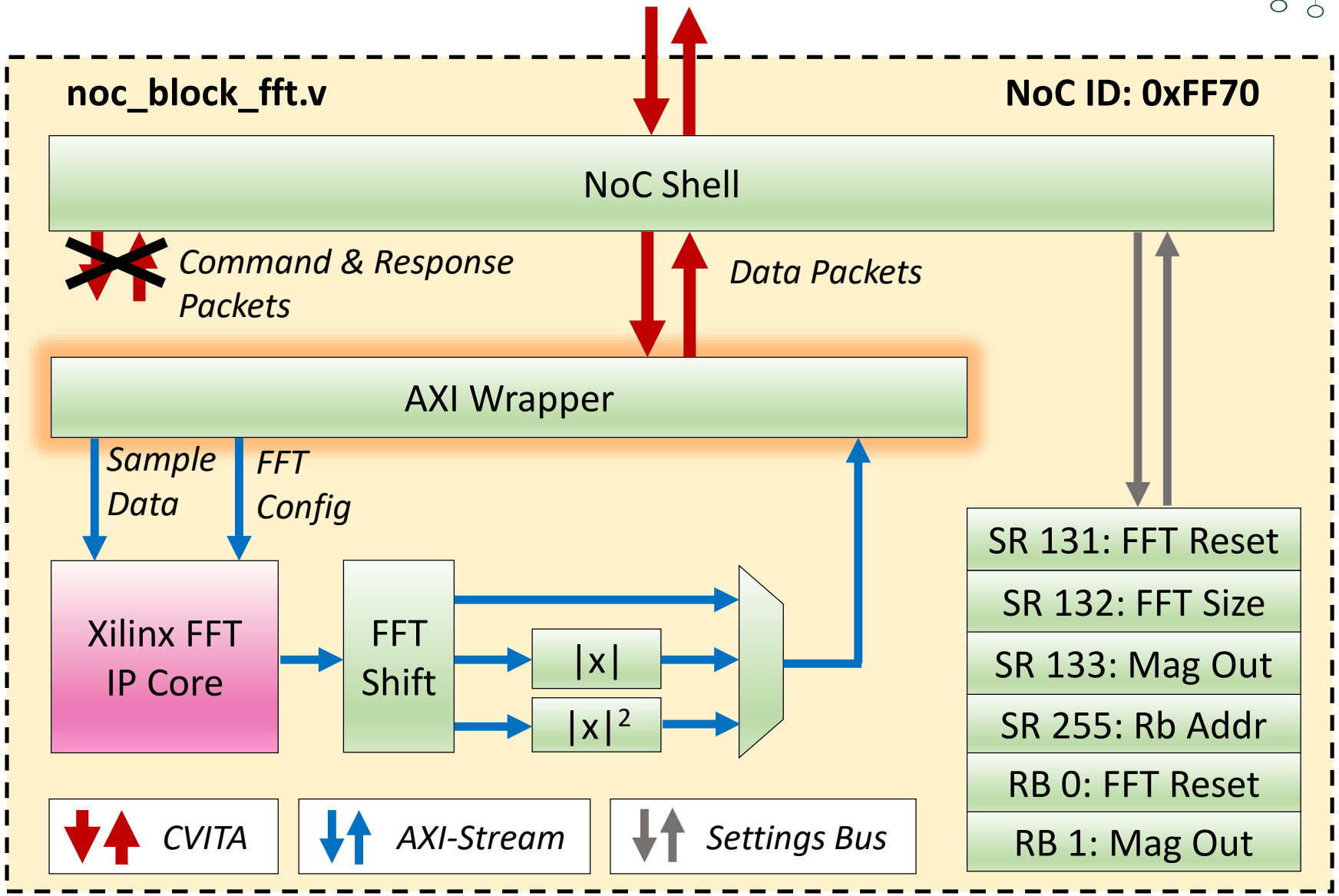
NoC Block FFT



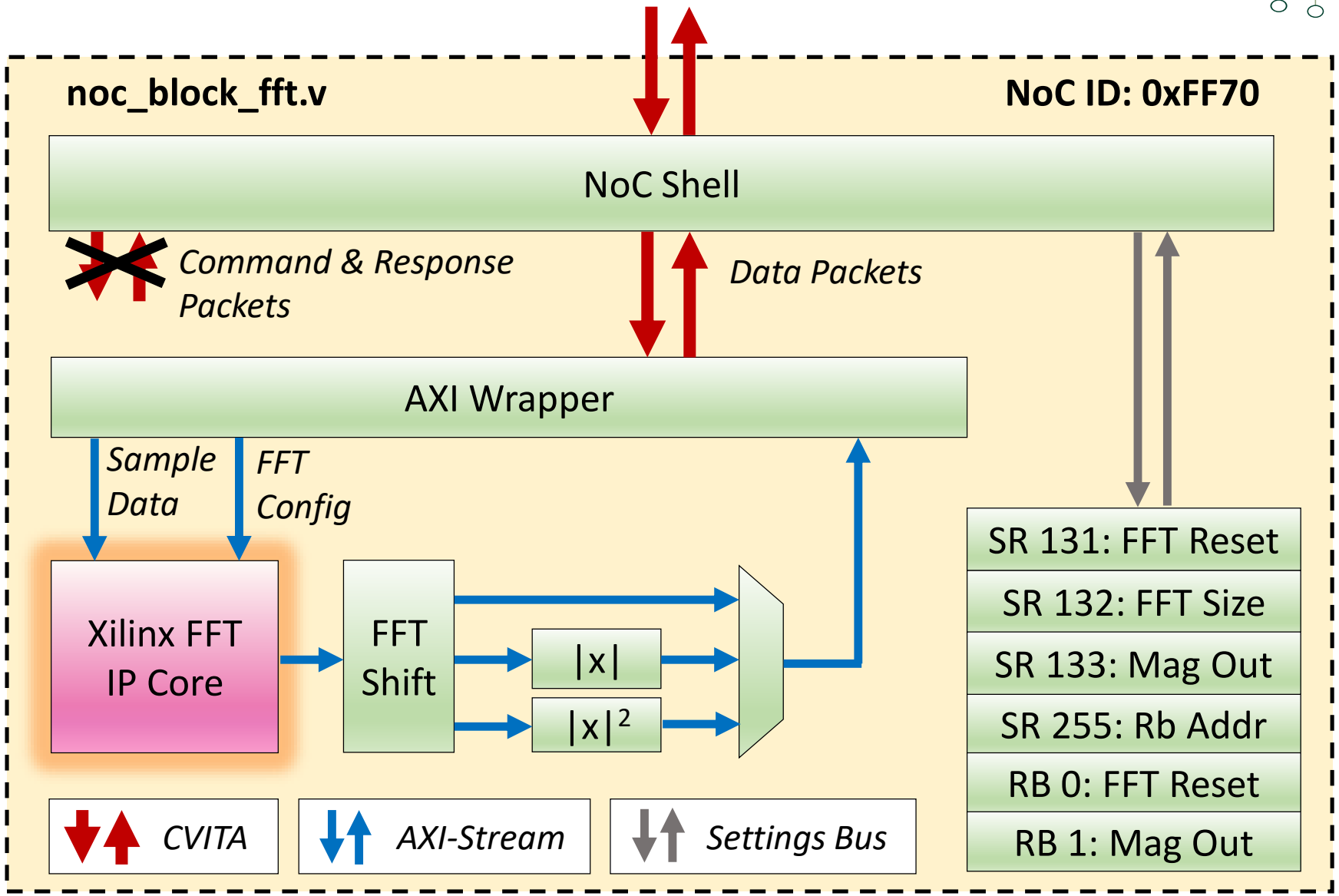
NoC Block FFT



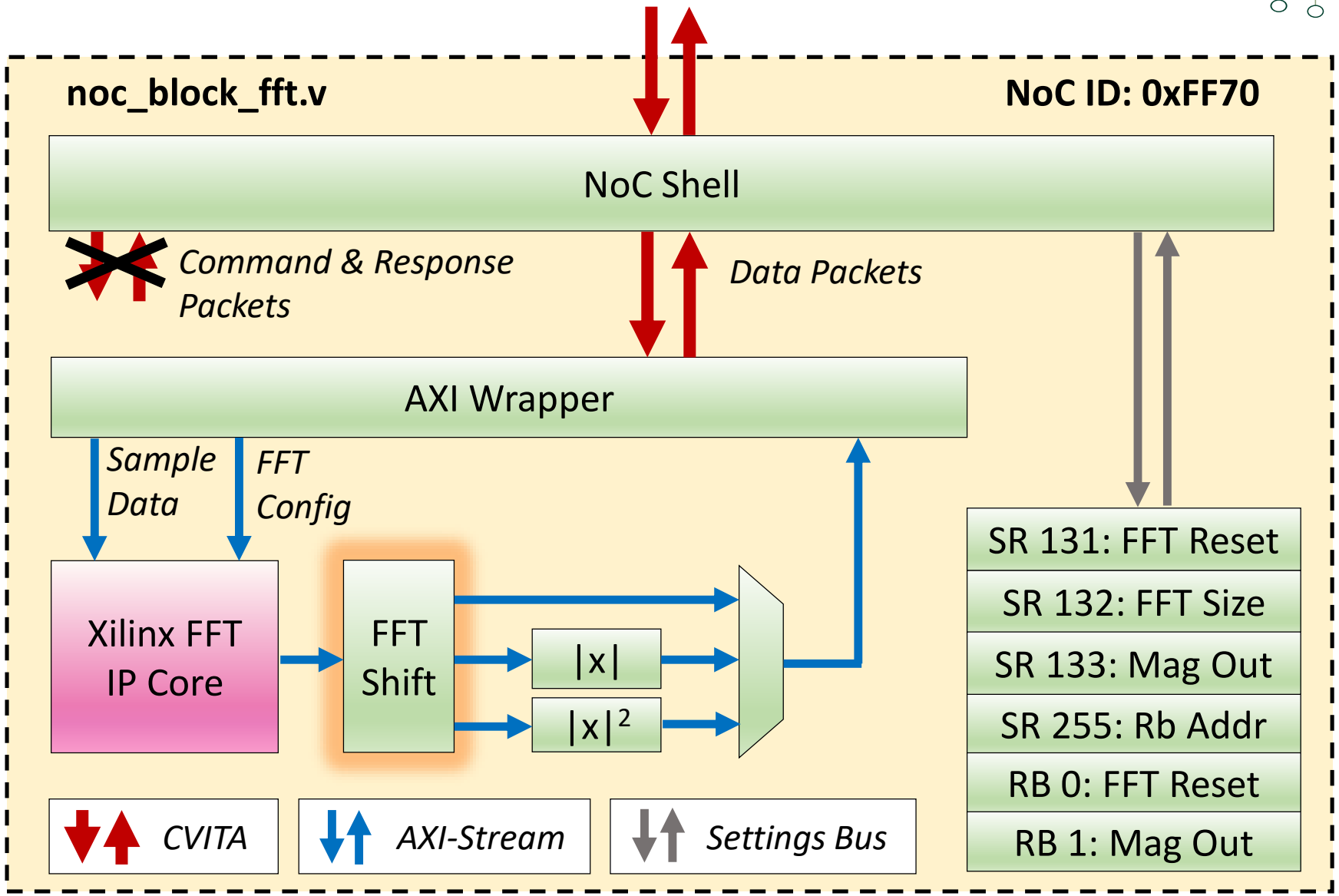
NoC Block FFT



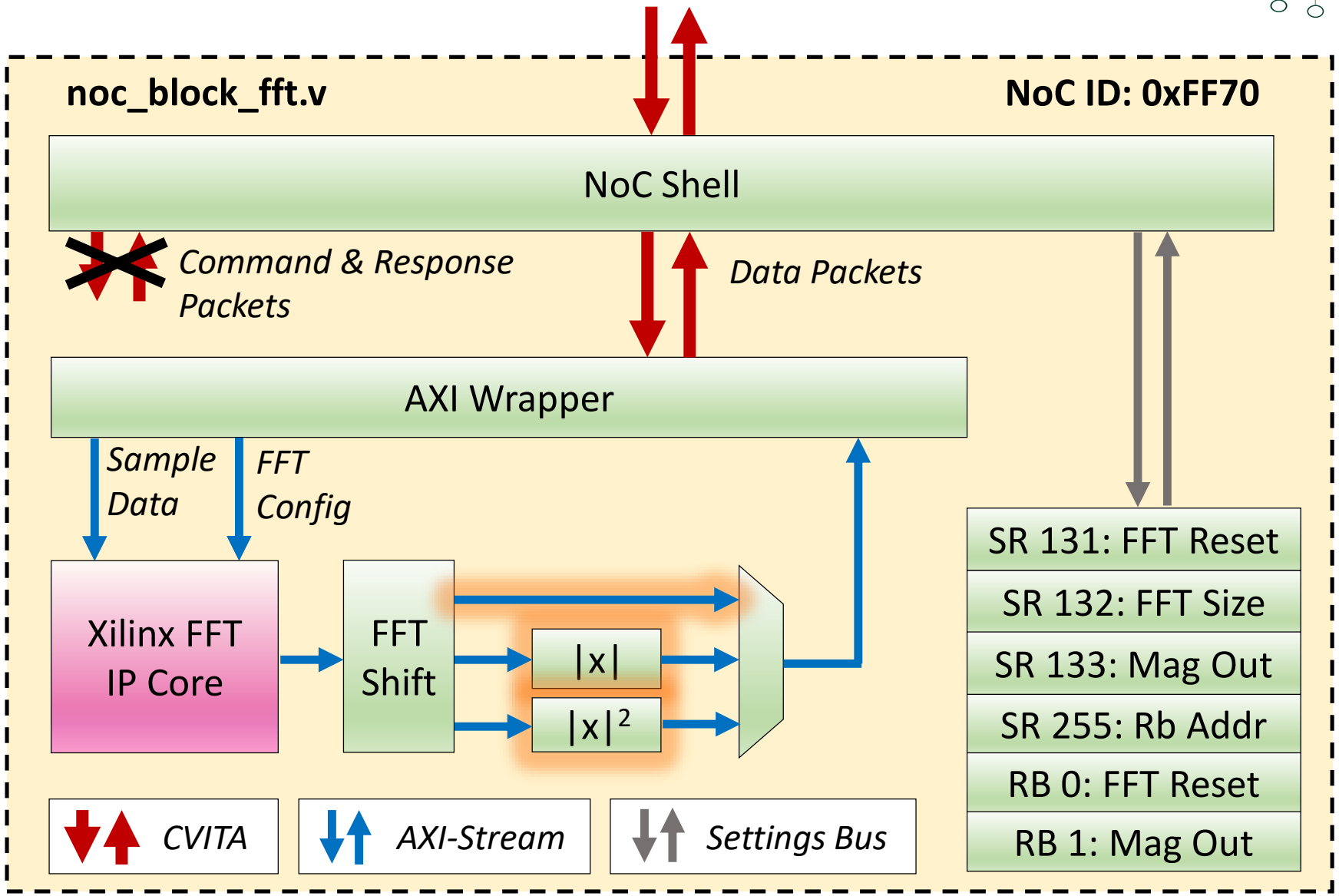
NoC Block FFT



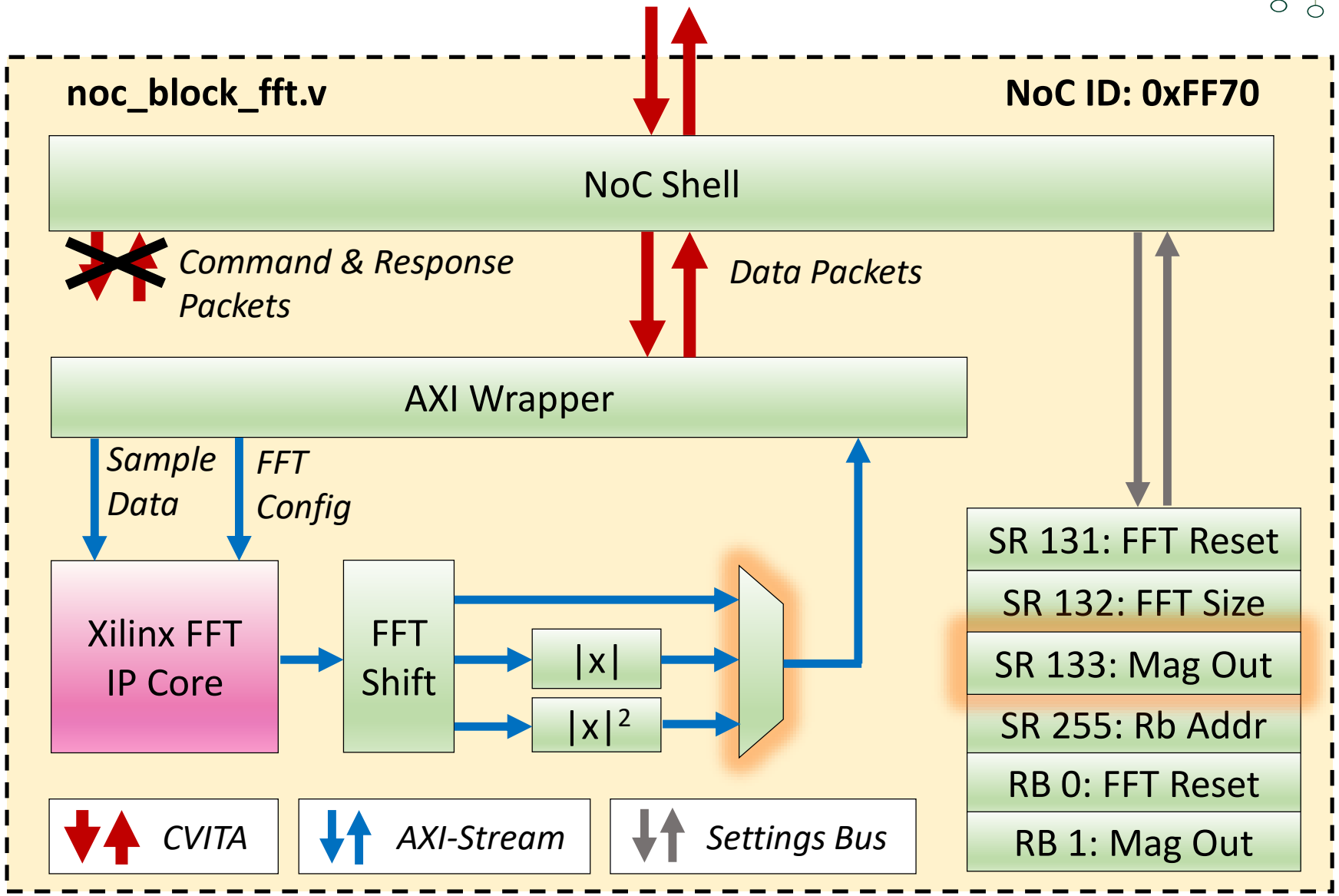
NoC Block FFT



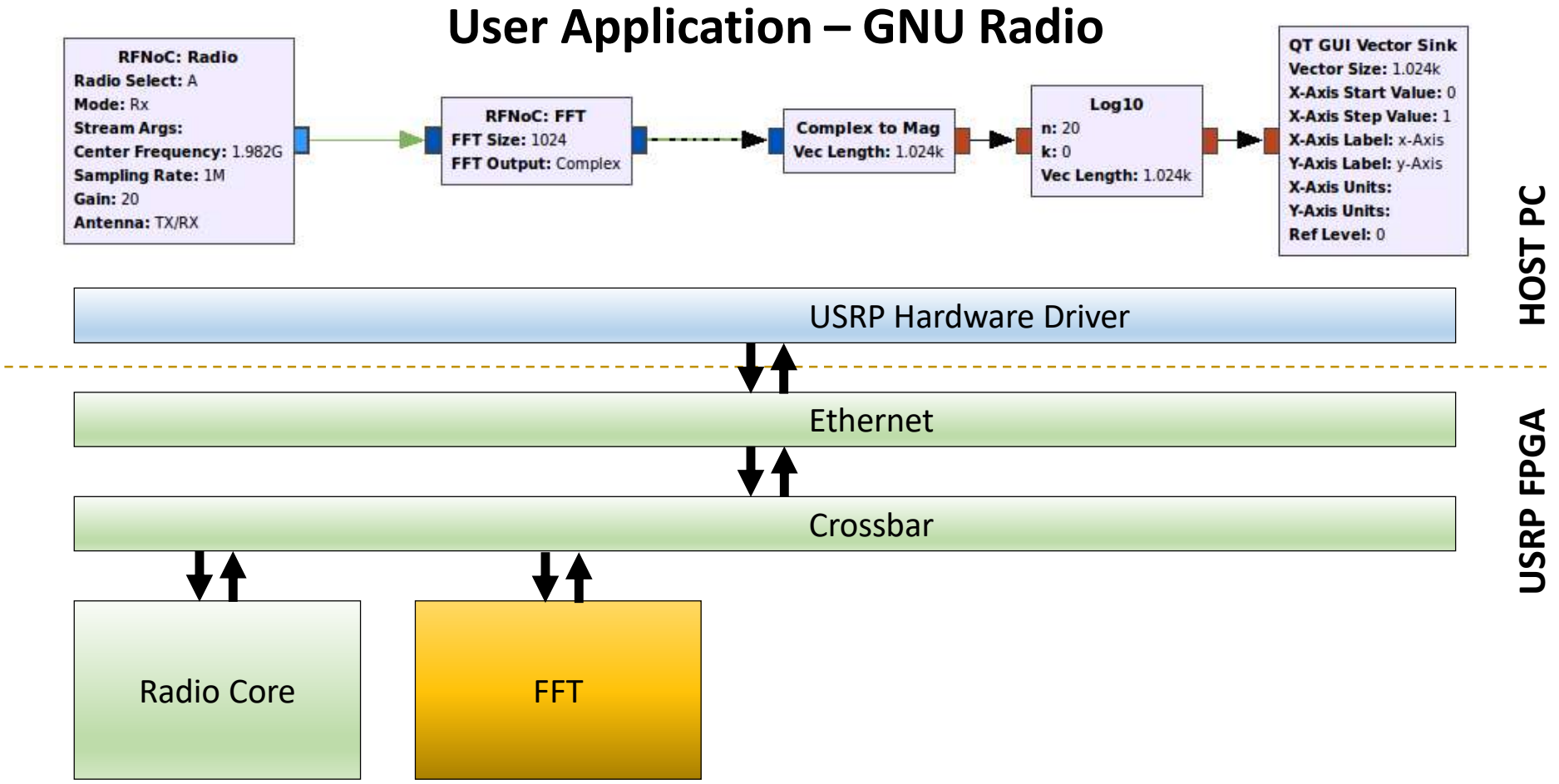
NoC Block FFT



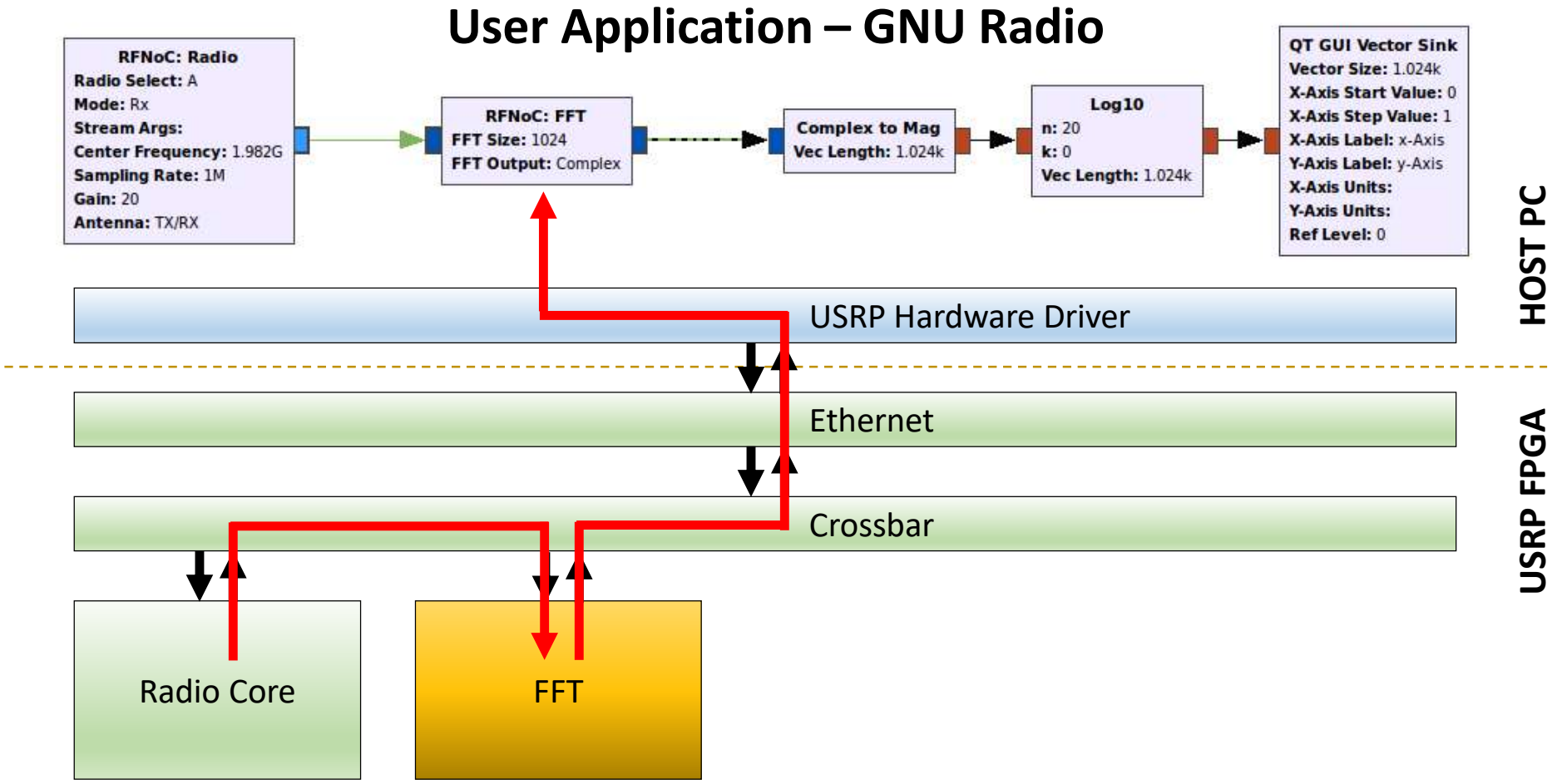
NoC Block FFT



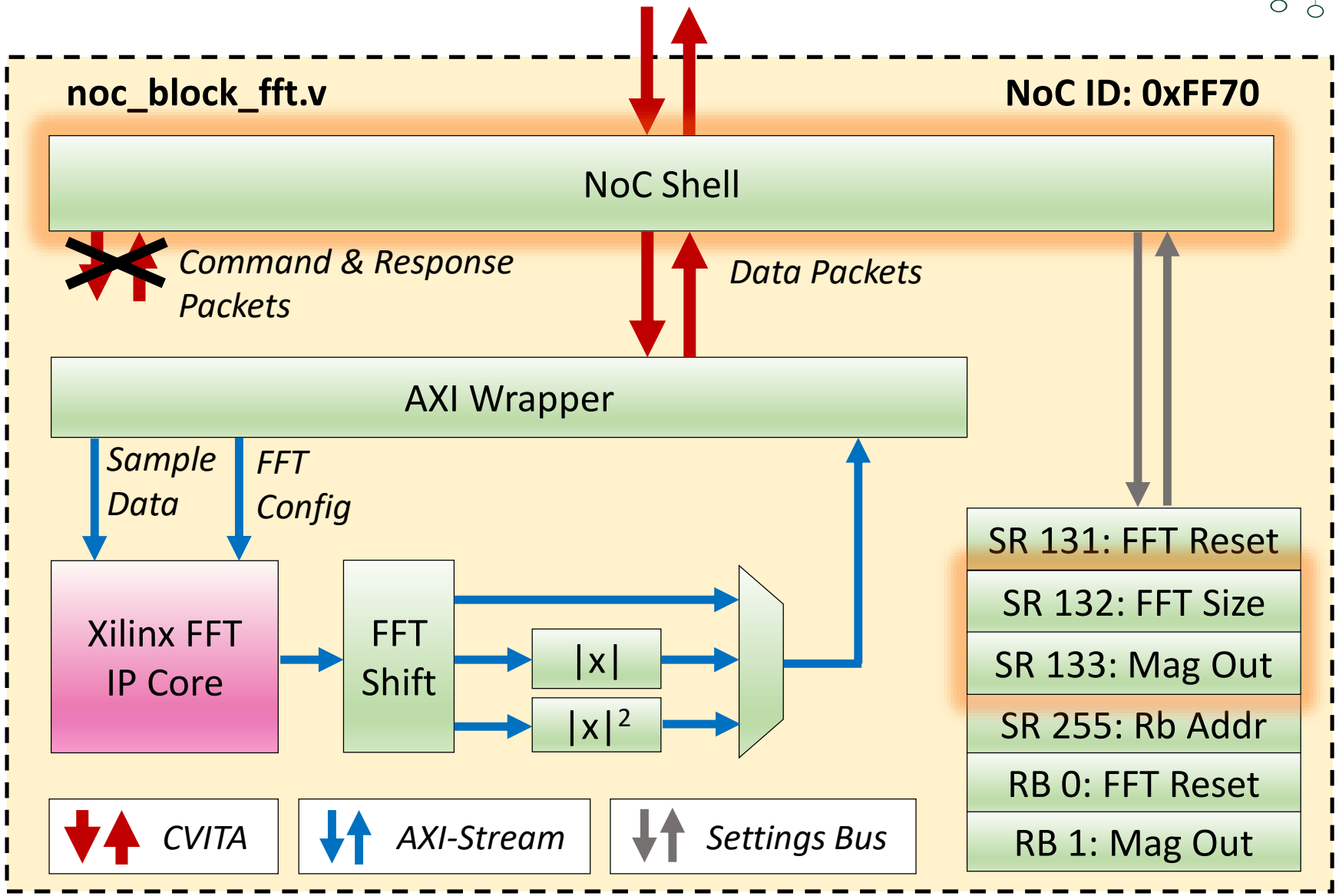
FFT Data Flow Step by Step



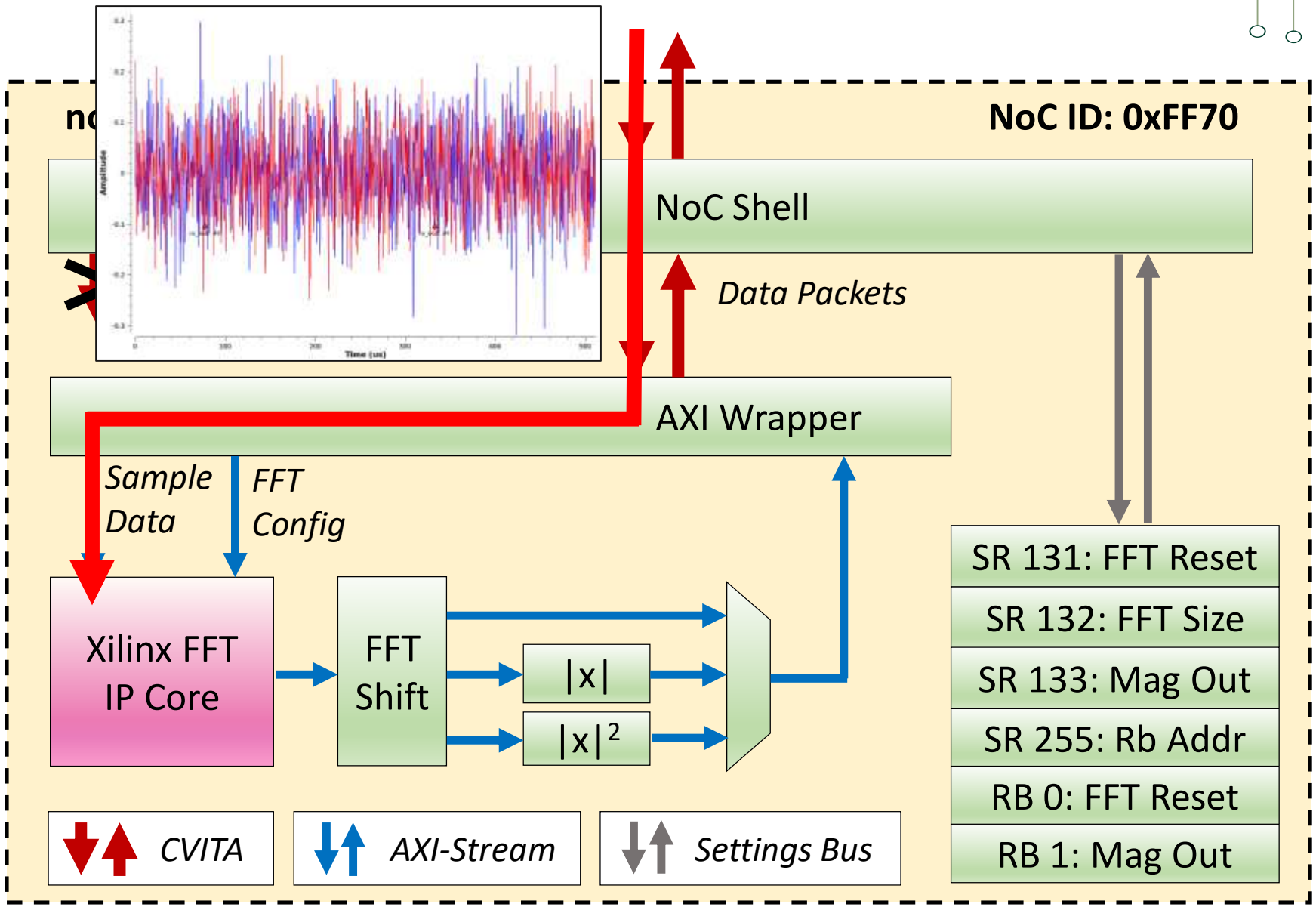
FFT Data Flow Step by Step



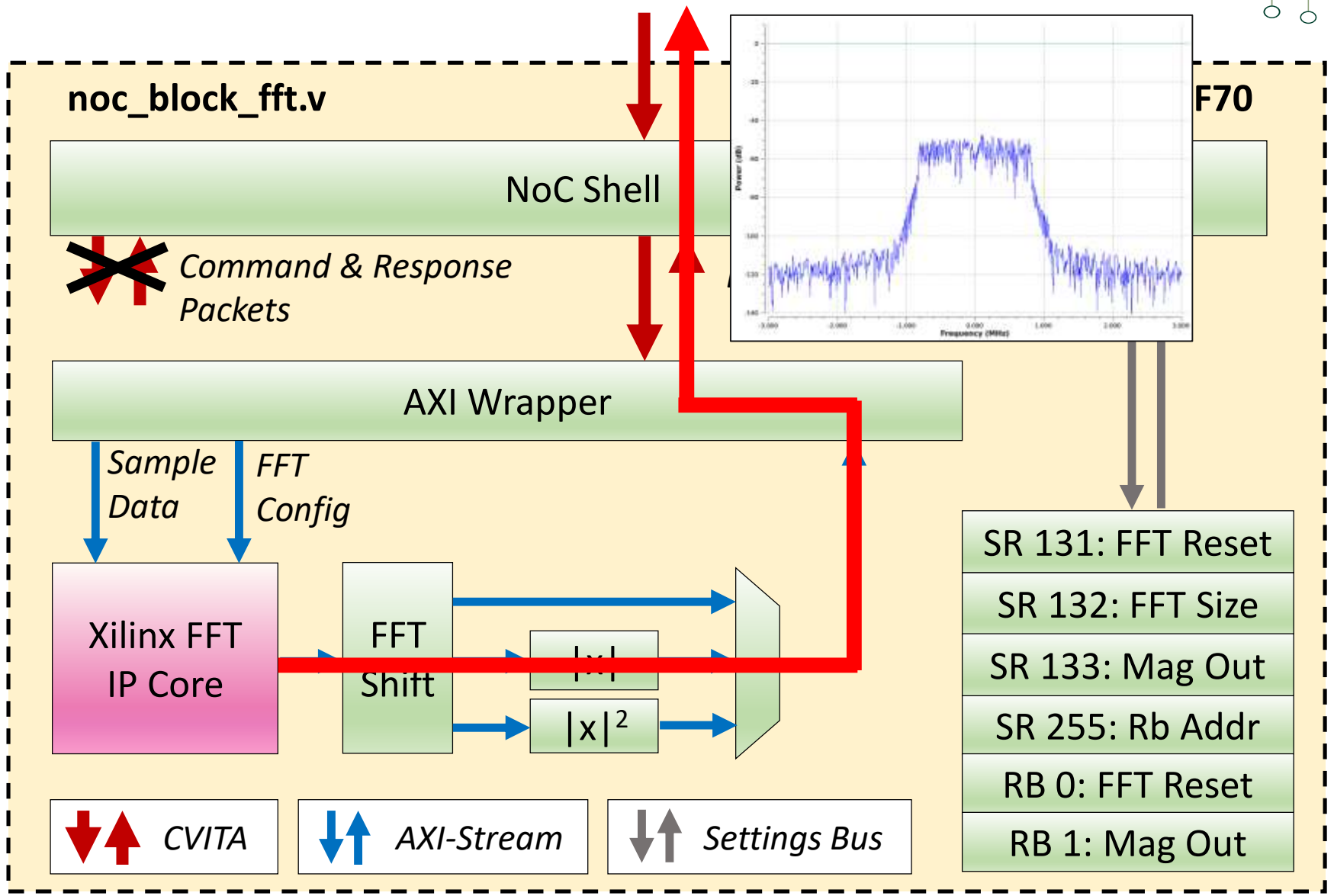
Initialization



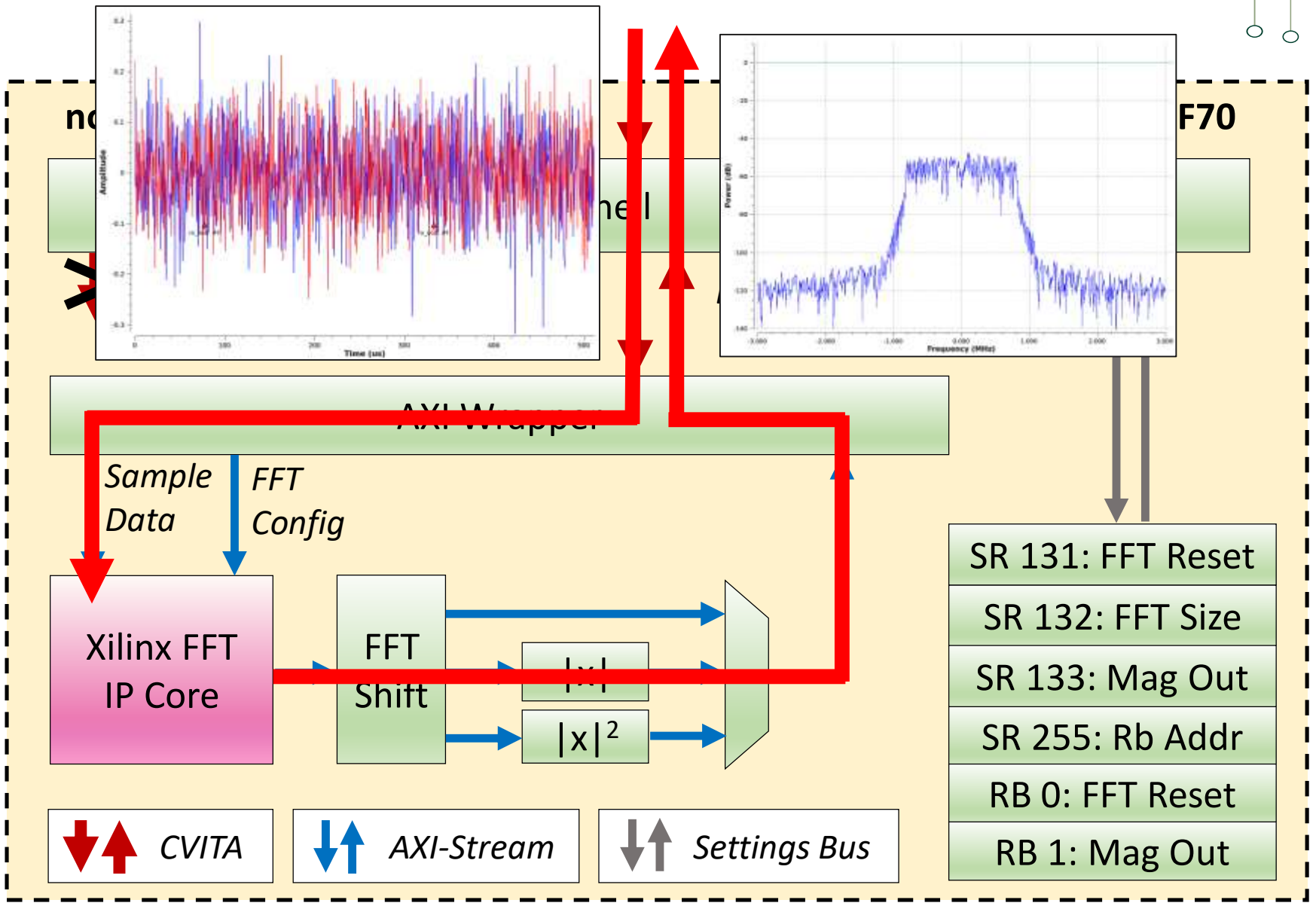
Radio Streams Data



FFT Processes and Outputs Packet

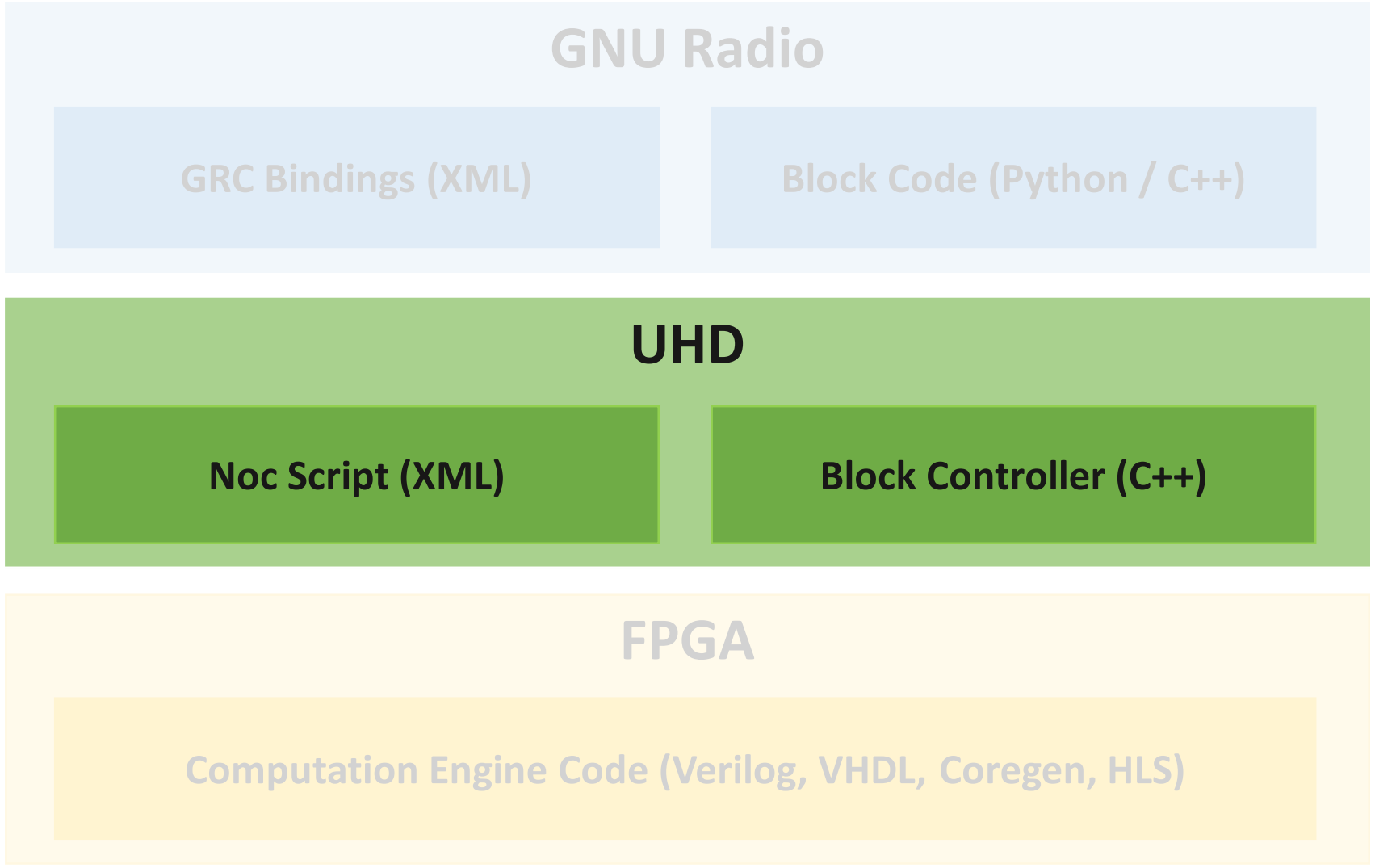


Continuously

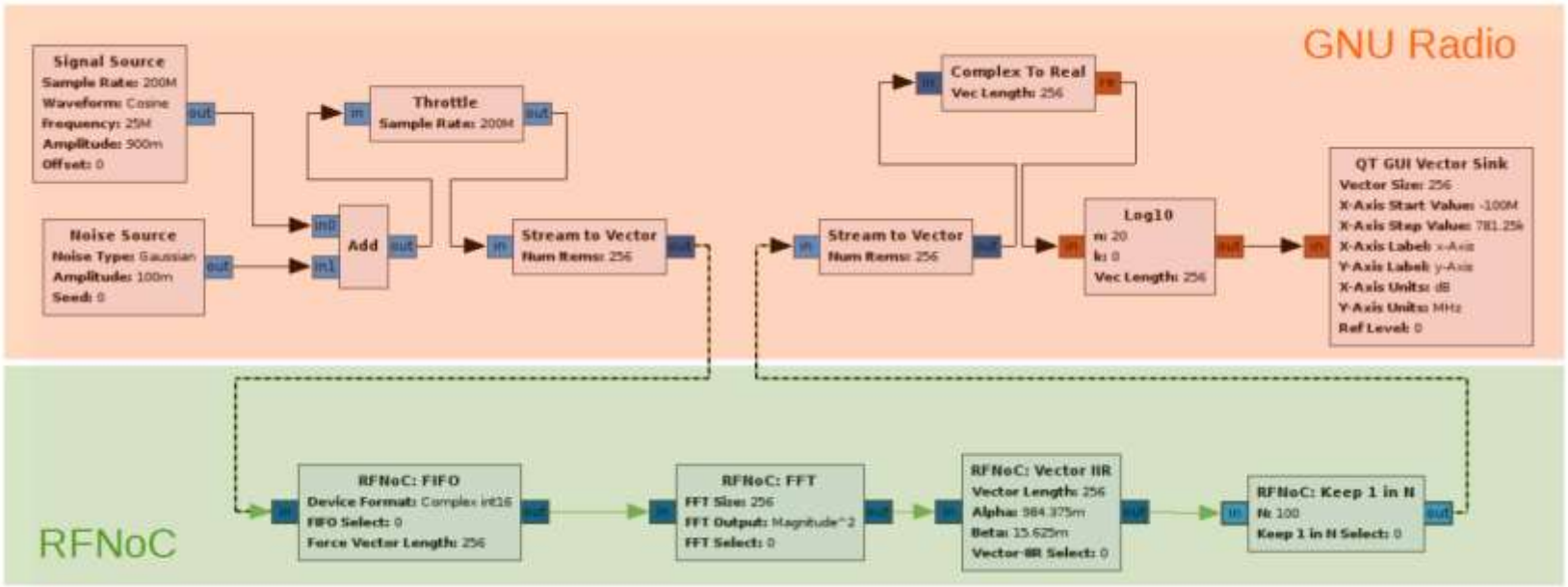


RFNoC Framework

Ettus Research™
A National Instruments Company

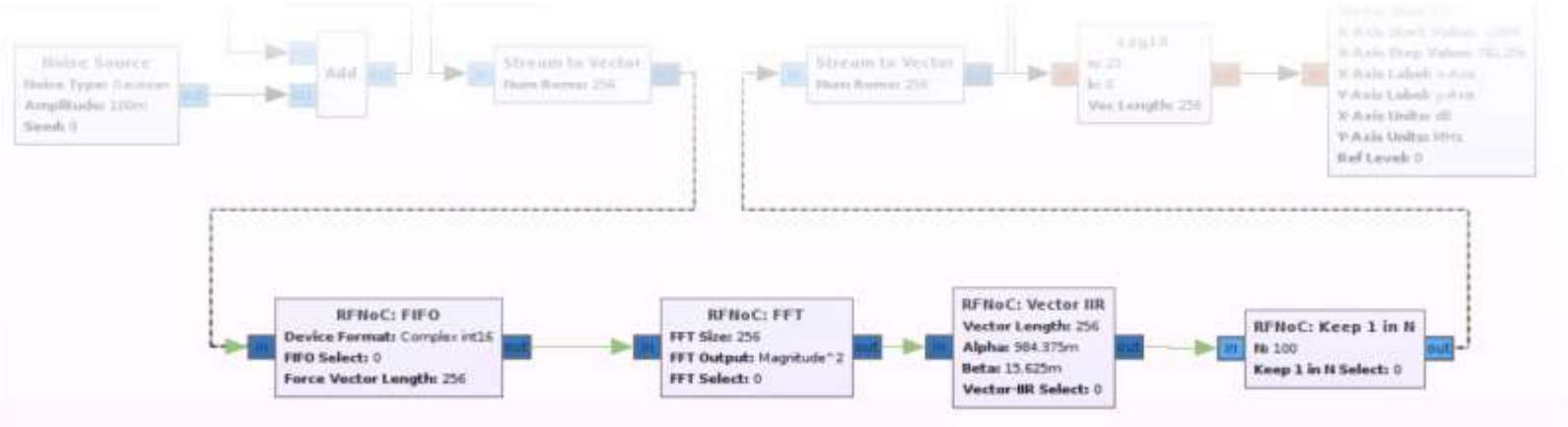


Host Development



- Application runs heterogeneously on host process such as
  - GNU Radio
  - MATLAB
  - C++
  - Any application that can call the UHD library
- Host needs access to all RFNoC CE settings

What does the Host do?



- Configure connections between blocks (e.g. configure stream IDs, set up flow control
- Configure block-specific operations and settings registers (e.g. FFT size, FIR taps…)
- Host-side checking of data type matching
- Abstracts the transport type and initiates streaming (Ethernet, PCIe, AXI)

Block Declaration

Ettus Research™ A National Instruments Company

XML file that describes the input and output data types & registers of a Computation Engine to UHD

- Blocks are identified by their NoC-ID
- Description of block for UHD
  - Argument List (e.g. FFT size)
  - Input- and output ports (data types, vector length, packet size)
  - Settings and read back registers
- NocScript: helps provide decision making, math, logic capability to the inputs and outputs of blocks
- Examples: <UHD>/host/include/uhd/usrp/rfnoc/blocks/

Noc Script

Ettus Research™ A National Instruments Company

Simplified programming language (Pseudo Language) designed for RFNoC framework

- No recompilation!
- Basic types: Integers, Strings, Doubles, Vectors
- Basic arithmetic and logic operations available
- Basic access to block arguments and settings registers
- See under the hood: <UHD>/host/lib/rfnoc/nocscript/

```
 <registers>
  <setreg>
    <name>GAIN</name>
    <address>128</address>
  </setreg>
</registers>
<args>
  <arg>
    <name>gain</name>
    <type>int</type>
    <value>1</value>
    <check>GE($gain, 0) OR LE($gain, 65535)</check>
    <check_message>Gain must be in range [0,65535].</check_message>
    <action>SR_WRITE("GAIN", $gain)</action>
  </arg>
</args>
```

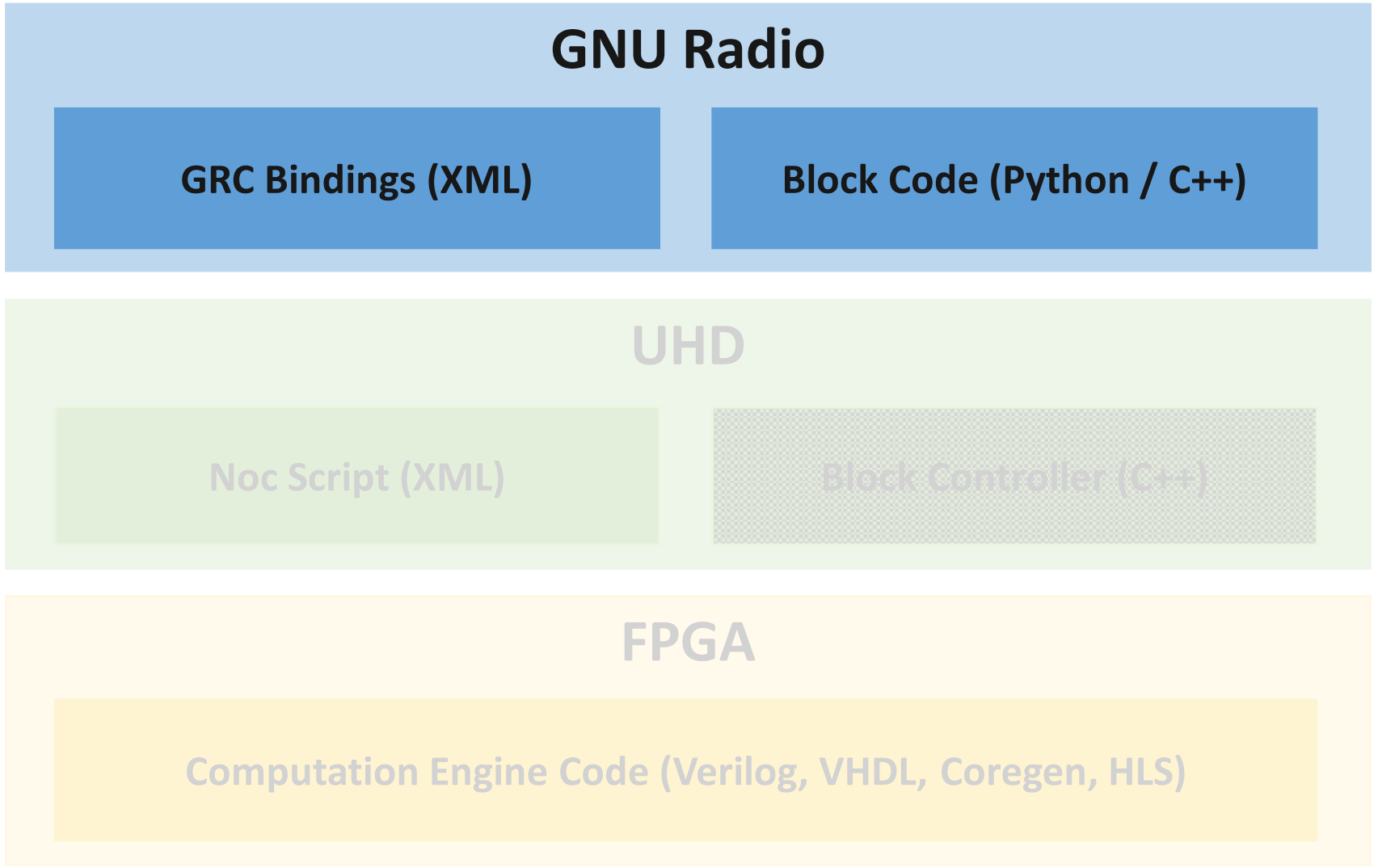
Block Controller

If Noc Script does not provide sufficient functionality use block controller (often not required)

- Example use cases
  - Independently perform operation(s) on a data stream
  - Examples: FFT, FIR, Signal Generator, Fosphor
- Written in C++
- Tells UHD about block capabilities and configuration
- Exposes methods to access and control block
- Default block controller class covers most cases

RFNoC Framework

Ettus Research™
A National Instruments Company



GR-ETTUS and GR-UHD

Ettus Research™
A National Instruments Company

- gr-uhd: mainline GNU Radio module interfacing with UHD
- gr-ettus: Out-of-tree module for experimental code, provides example RFNoC blocks in GRC
- Available online: http://github.com/EttusResearch/gr-ettus.git

GNU Radio Companion Bindings

Ettus Research™ A National Instruments Company

- XML file that describes GNU Radio blocks to GNU Radio Companion
- No recompilation!
- Not strictly necessary when using GNU Radio directly with python
- Tutorial on how to write them:
  www.gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion

GNU Radio Block Code

Ettus Research™ A National Instruments Company

- Written in C++ or Python
- How does GNU Radio interface to RFNoC?
  - via C++ infrastructure code in gr-ettus
  - Gr-ettus provides a base RFNoC block class
  - Users extend base class for their RFNoC blocks
  - Many blocks can use base class “as is”
    - No need to write C++ or Python if using base class (most use cases)

RFNoC Prerequisites

Ettus Research™ A National Instruments Company

1. Build/install UHD from rfnoc-devel
2. Build/install GNU Radio
3. Build/install GR-ETTUS OOT from rfnoc-radio-redo branch
4. Install Xilinx VIVADO
5. Latest compatible version documented in UHD manual
   - Webpack license for E Series
   - Full license needed for X Series
6. Setup VIVADO build environment
   - Run <UHD>/fpga-src/usrp3/top/e300/setupenv.sh
7. Useful Resource: RFNoC
   - https://kb.ettus.com/Getting_Started_with_RFNoC_Development

Final Takeaway

Ettus Research™ A National Instruments Company

RFNoC is for FPGAs is what GNU Radio is for GPPs

	RFNoC	GNU Radio
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	✓ (Growing)	✓ (Huge and well-tested)
Has a graphical front end	✓ (gr-ettus)	✓ (GRC)
Open Source	✓	✓
Writes your blocks for you	✗	✗

www.kb.ettus.com/RFNoC_Getting_Started_Guides

Ettus Research™
A National Instruments Company

Questions & Answers