UHD Four-O

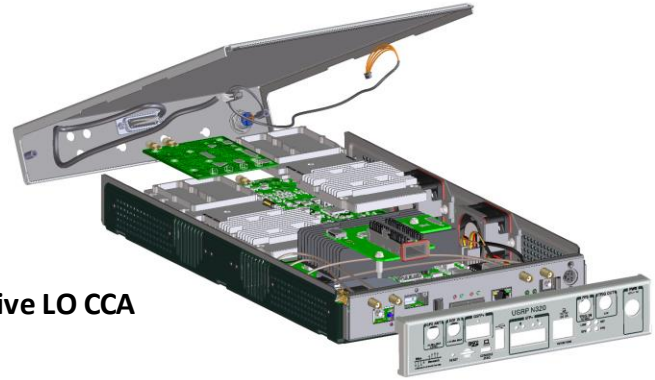
Features, Features, Features

Martin Braun

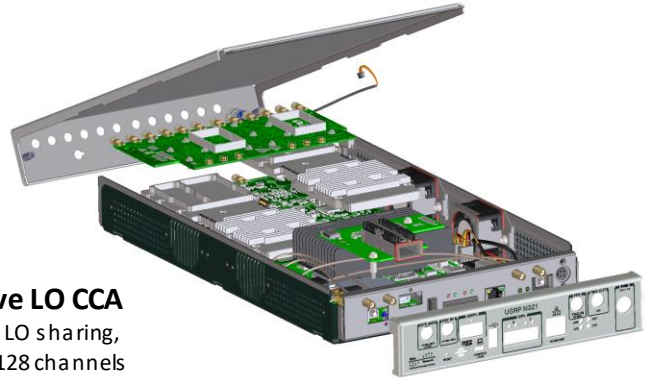
Ettus Research / National Instruments SDR

2019 Products

N320/N321: Announced May 2019



Passive LO CCA



Active LO CCA
N321: LO sharing,
up to 128 channels

1 MHz to 6 GHz
2 RF (TX/RX) Channels: 200 MHz BW

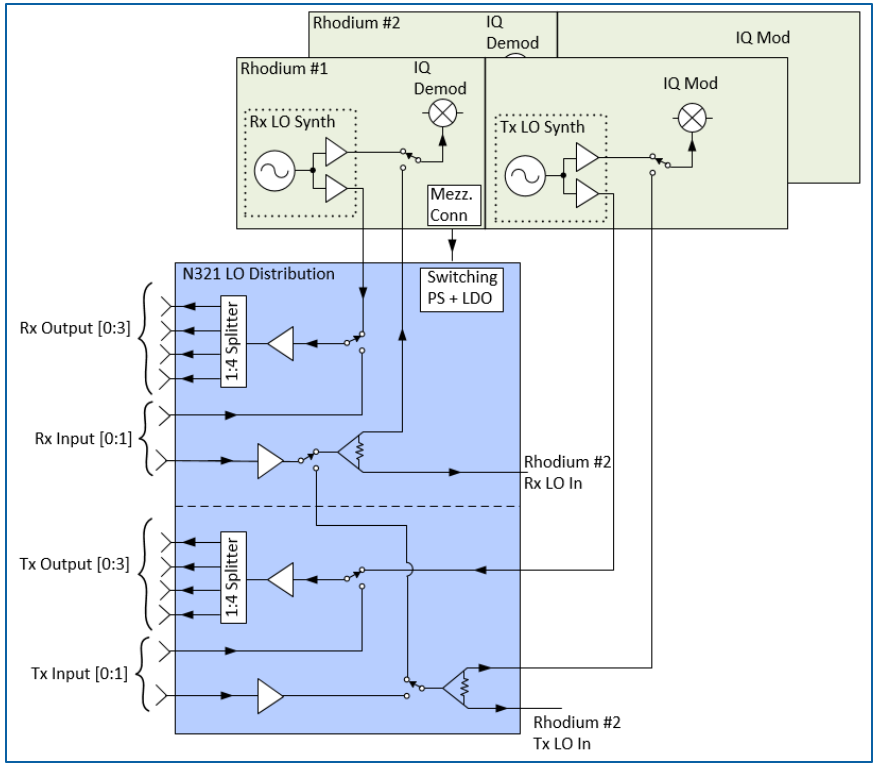
N320

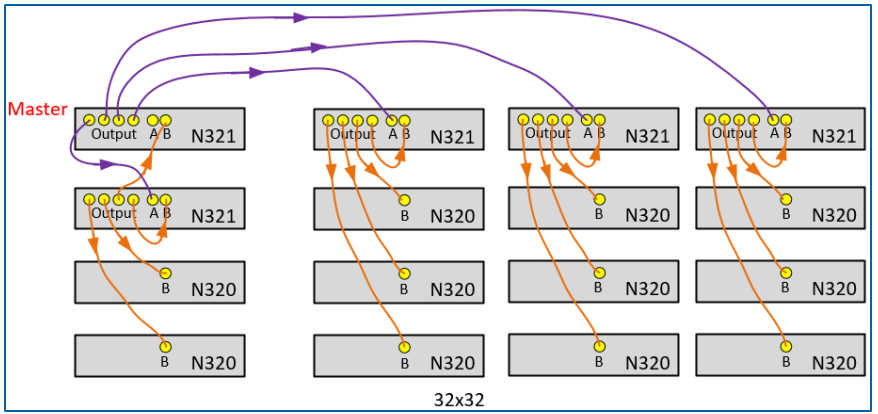
N321

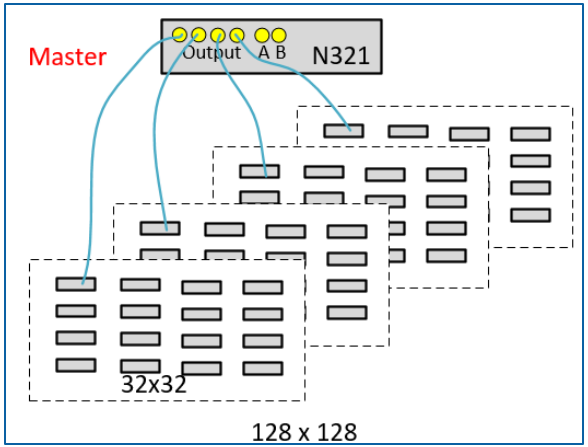
NATIONAL INSTRUMENTS

LO Distribution with the N321

- Star LO Distribution
- Support up to 128x128 MIMO







High Level Features Specs

Description	Spec
Frequency range	1 MHz – 6 GHz
Number RF boards/channels per module	2 (Tx/Rx and Rx2)
Instantaneous bandwidth	200 MHz (250 Msps)
Available Sampling Rates	200, 245.76, 250 Msps
Tx maximum output power	+18 to +13 dBm
Rx maximum input power	+10 dBm
Rx and Tx gain range / resolution	60 dB / 1 dB
High-Speed Digital Interfaces	2xSFP+, 1xQSFP
Customizable File System, RASM Features	Provided by OE-based OS

Overall Product Overview

B-Series	E-Series	X-Series	N300 Series	N320 Series
• Low Cost • USB-Powered	• Embedded • Small Form Factor • Ruggedized Enclosures	• High Rate • First device with RFNoC support • Modular (TwinRX, Basic, UBX, GPSDO, etc.)	• High Channel Density • Embedded • RASM Features	• High Rate • High RF Fidelity • LO Sharing • RASM Features

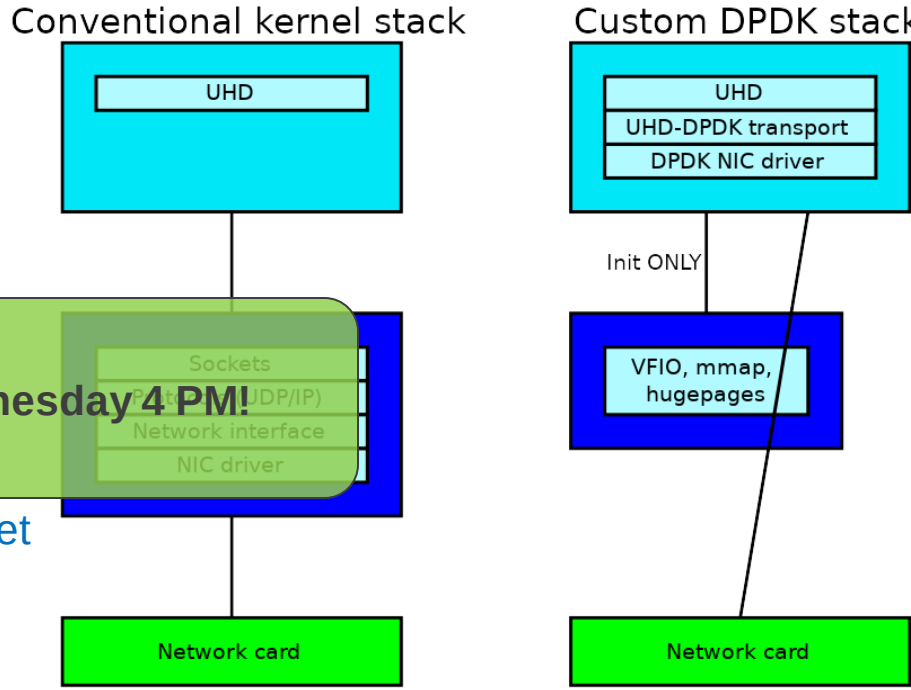
Come to our Booth!

NATIONAL INSTRUMENTS

UHD-DPDK

- DPDK-based transport for UHD
  - UHD network traffic completely bypasses the kernel
  - Much-reduced latency for transmission/reception
- Minimal network stack
  - UDP / IPv4, ARP
  - No IPv4 fragmentation
- Zero-copy operation
  - Queues contain pointers to DMA-able packet buffers
- Available for X-Series, N3XX-Series

See talk on Wednesday 4 PM!



Software Roadmap

UHD 3.15 LTS & UHD 4.0

UHD 3.15 LTS

- Long-Term Support
- Timeline: Q4 2019
- Focus on stabilization and bugfixes
- RFNoC now enabled by default
- Current master branch

UHD 4.0

- A new era!
- Timeline: Q1 2020
- Python 2 support is EOL
- Available as a preview on master-next branch (bleeding-edge!)
- > 50000 LoC changes for FPGA and UHD repos, respectively
- **Major overhaul of underlying RFNoC architecture**

Proto-RFNoC ⟺ **RFNoC**

NATIONAL INSTRUMENTS

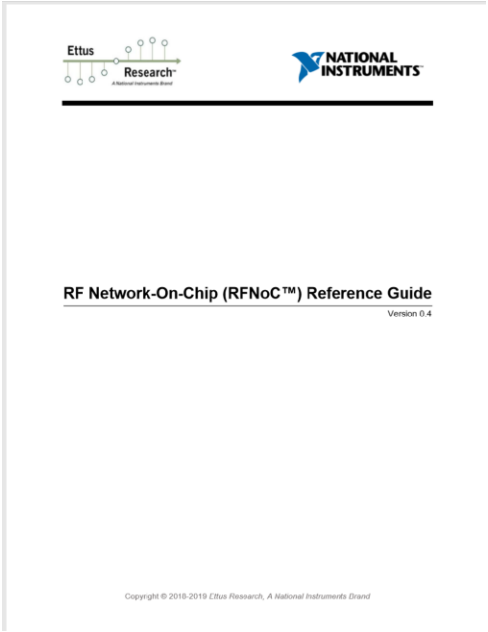
Motivations for Evolving RFNoC

- Scale Total System Bandwidth
  - Increase bandwidth per channel beyond 250 Msps
  - RTL and Software APIs to achieve bandwidth
- Scale Number of Blocks
  - Framework has minimal area overhead in the FPGA
  - Better graph features in Software
- Allow Users to Make more Design Tradeoffs
  - Throughput vs Latency
  - Flexibility vs Area
- Improve User Experience

Documentation Updates

Specification

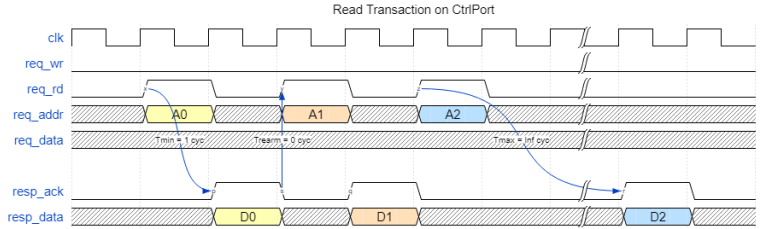
- Detailed specification for all FPGA and Software interfaces
- Timing Diagrams
- API Reference
- General Framework Info



http://files.ettus.com/app_notes/RFNoC_Specification.pdf

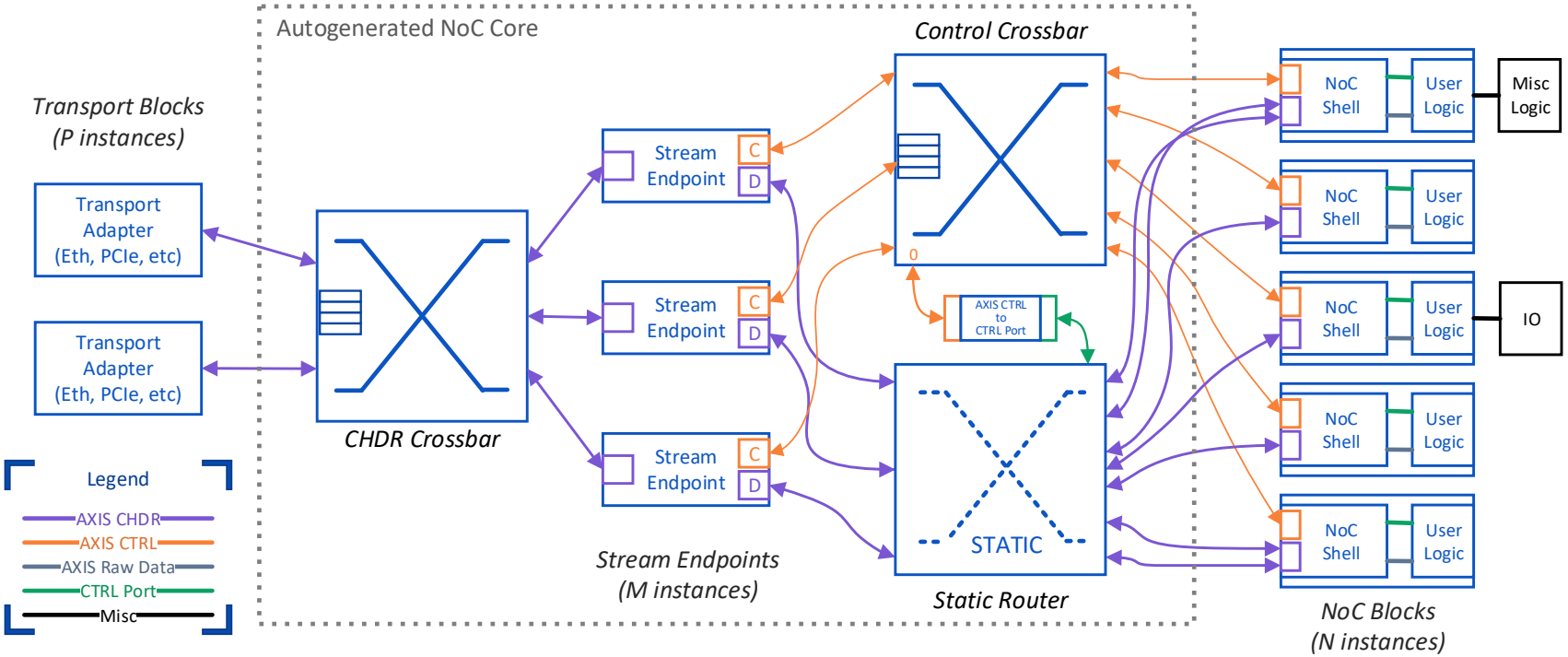
#	Memory Layout <------ CHDR_W = 64 bits ------>						Required?
0	Flags (6)	PktType (3)	NumMData (7) *Value=M*	SeqNum (16)	Length (16) *Value=L*	DstEPID (16)	Y
1	Timestamp (64)						N
2	Metadata[0] (CHDR_W)						N
.	...						.
M+1	Metadata[M-1] (CHDR_W)						N
M+2	Payload[0] (CHDR_W)						Y
.	...						.
M+N+1	Payload[N-1] (CHDR_W)						N

Table 1: Memory layout of a CHDR packet



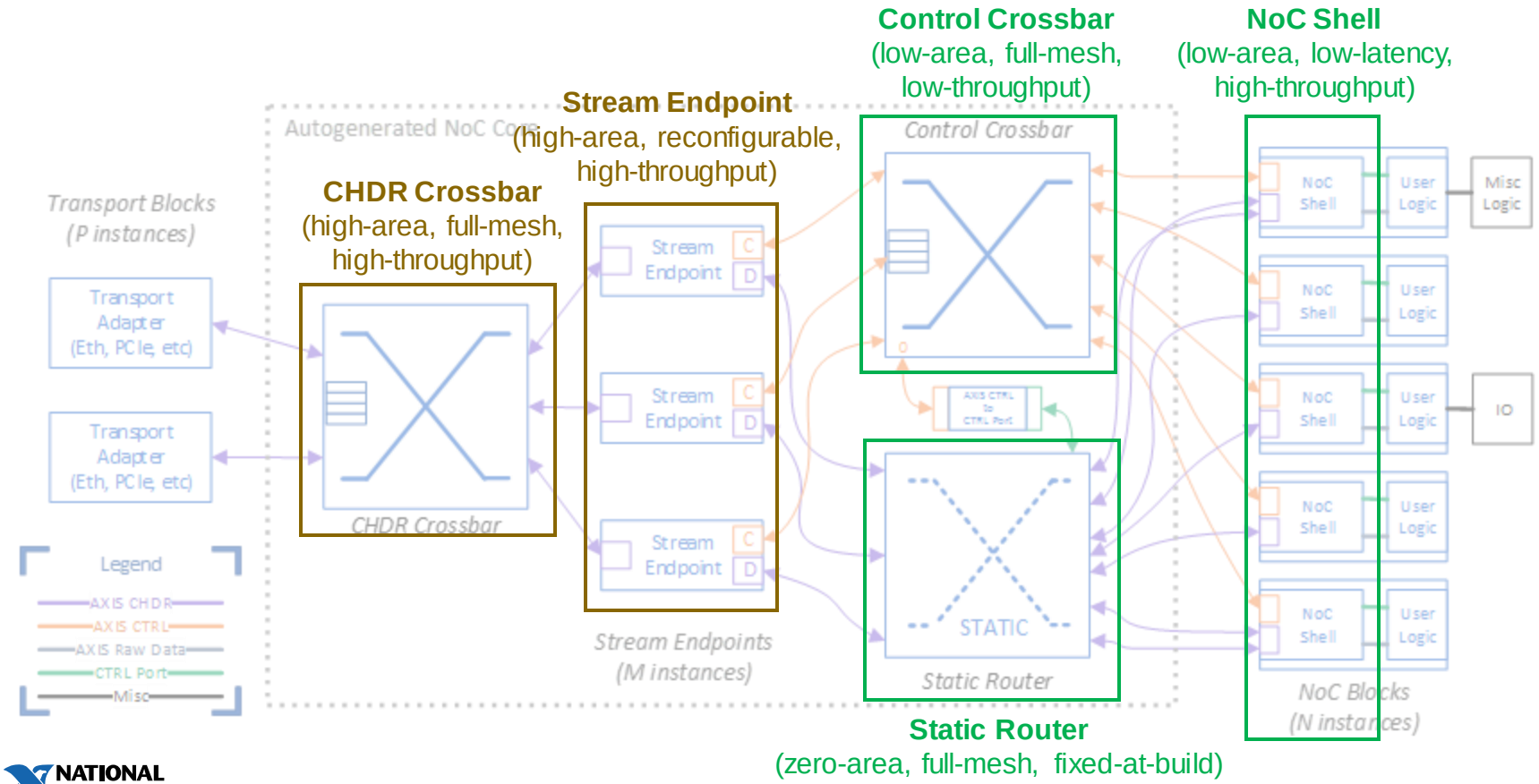
NATIONAL INSTRUMENTS

RFNoC Dataflow Updates

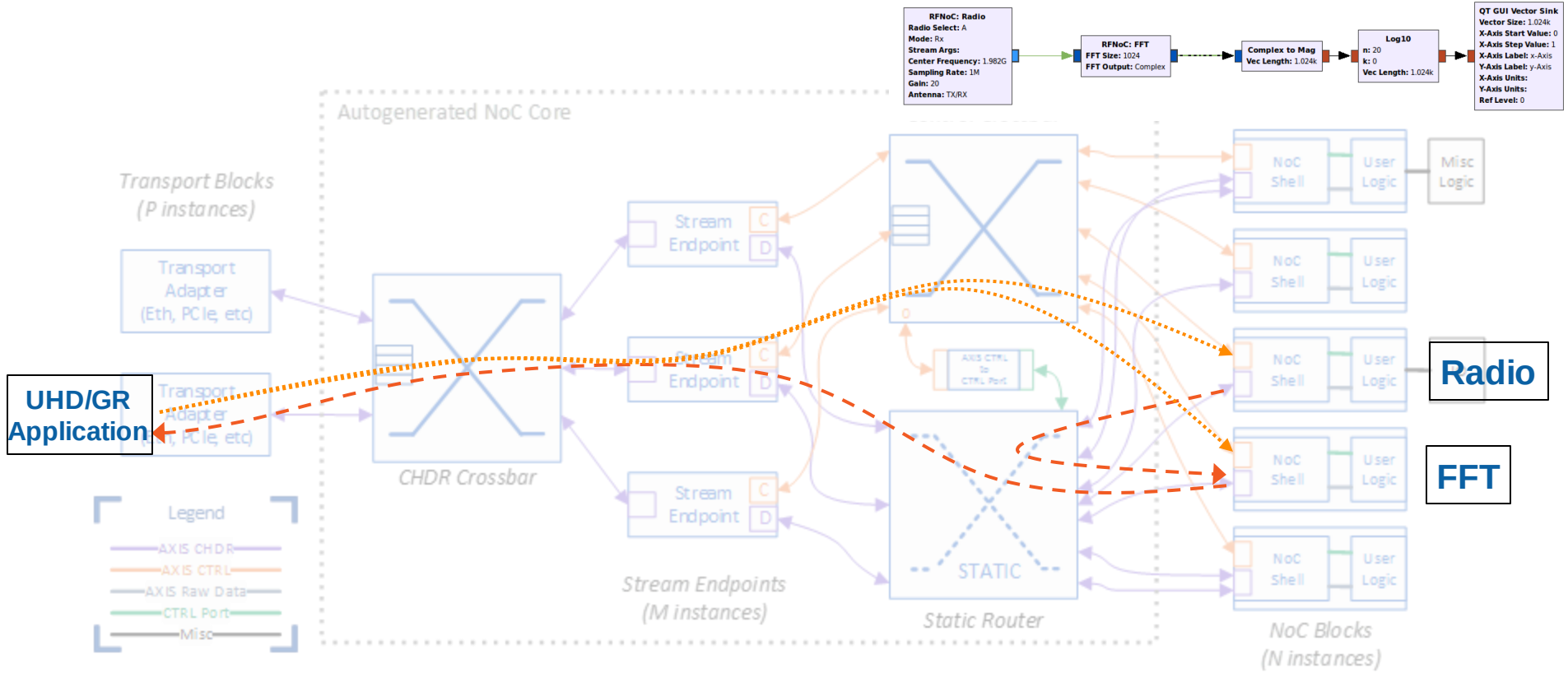


NATIONAL INSTRUMENTS

RFNoC Dataflow Updates

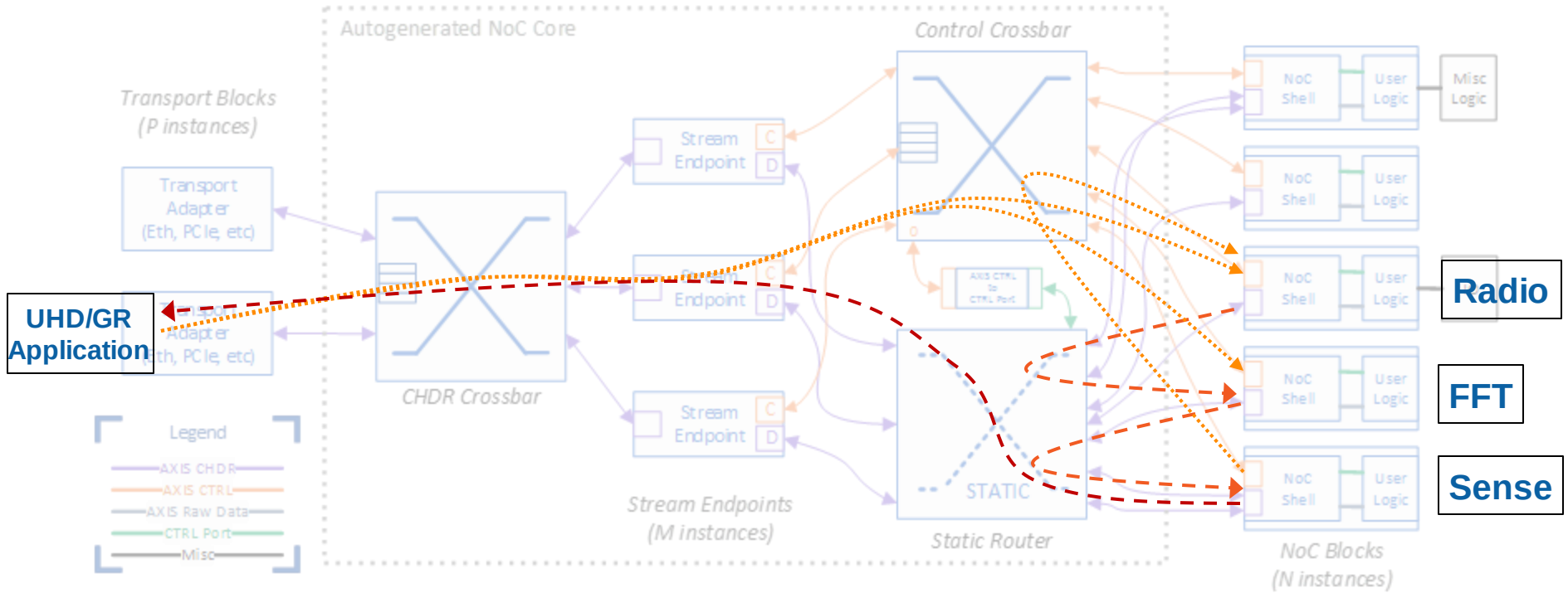


RFNoC Dataflow Updates: Spectrum Analysis Example



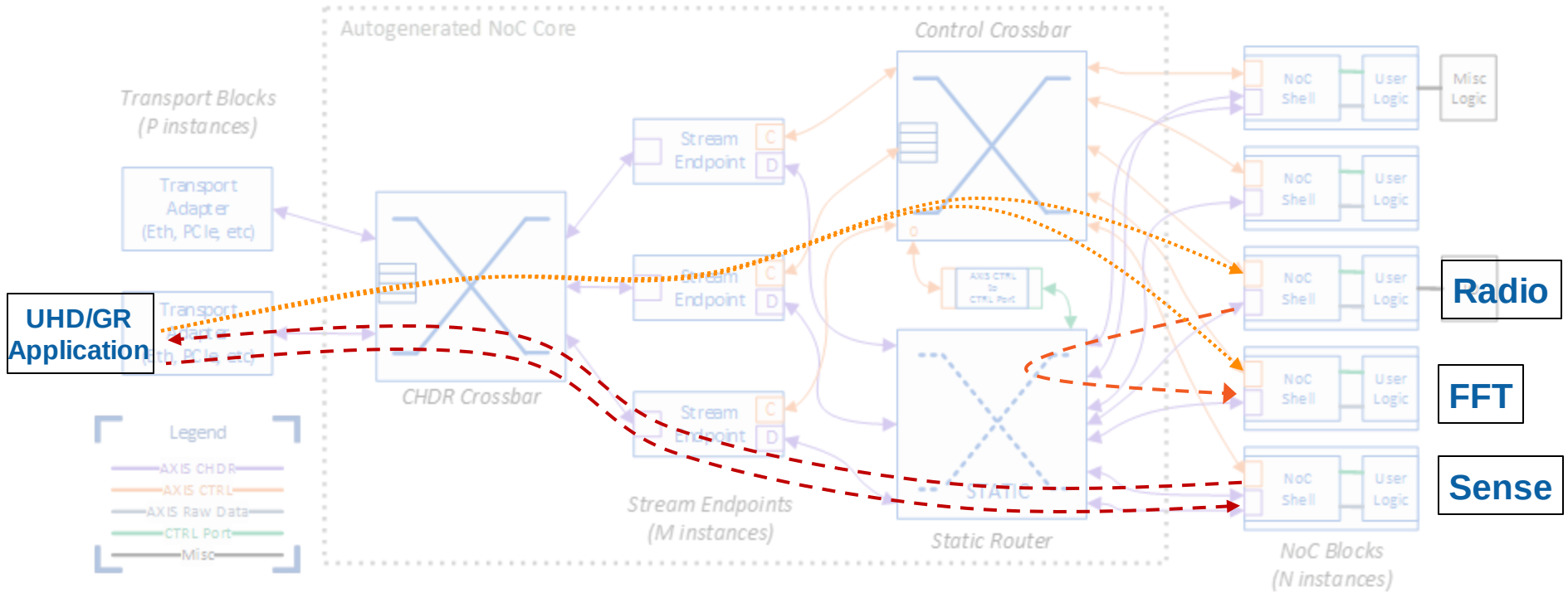
NATIONAL INSTRUMENTS

RFNoC Dataflow Updates: Spectrum Sensing Example



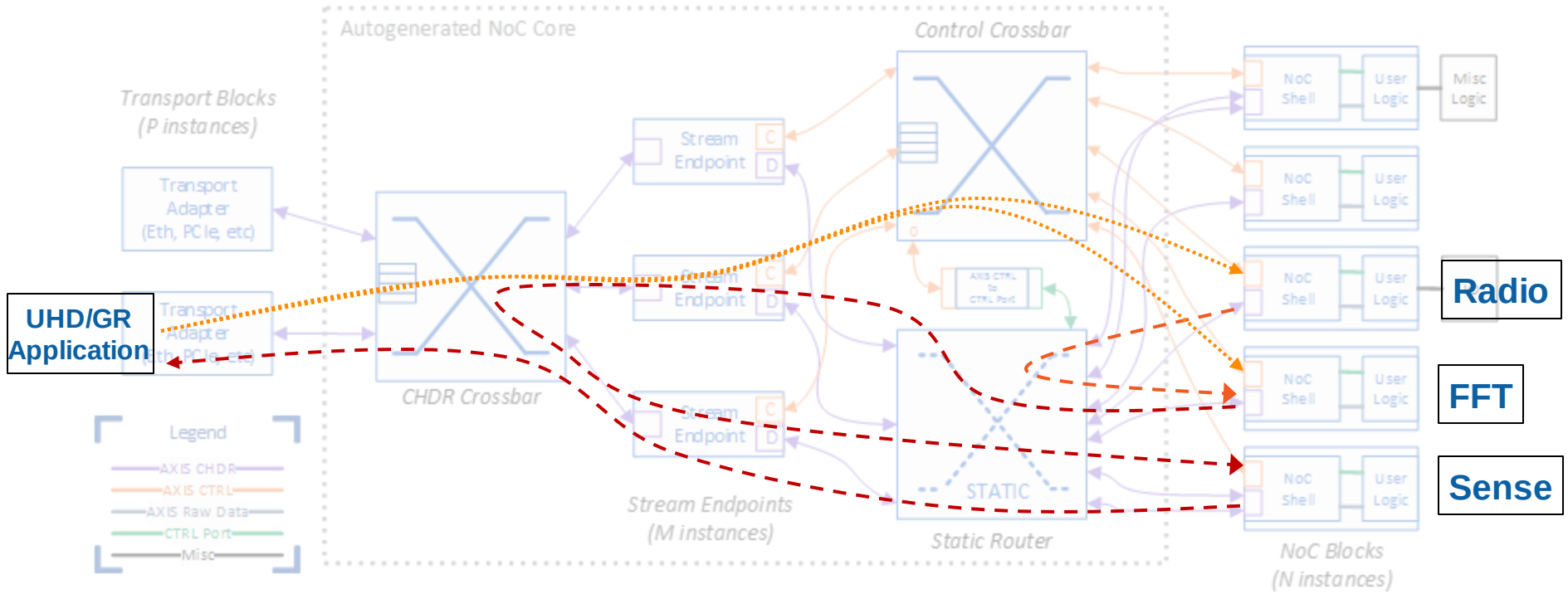
NATIONAL INSTRUMENTS

RFNoC Dataflow Updates: Spectrum Sensing (Debug)



NATIONAL INSTRUMENTS

RFNoC Dataflow Updates: Spectrum Sensing (Debug)



NATIONAL INSTRUMENTS

RFNoC Software Graph Updates

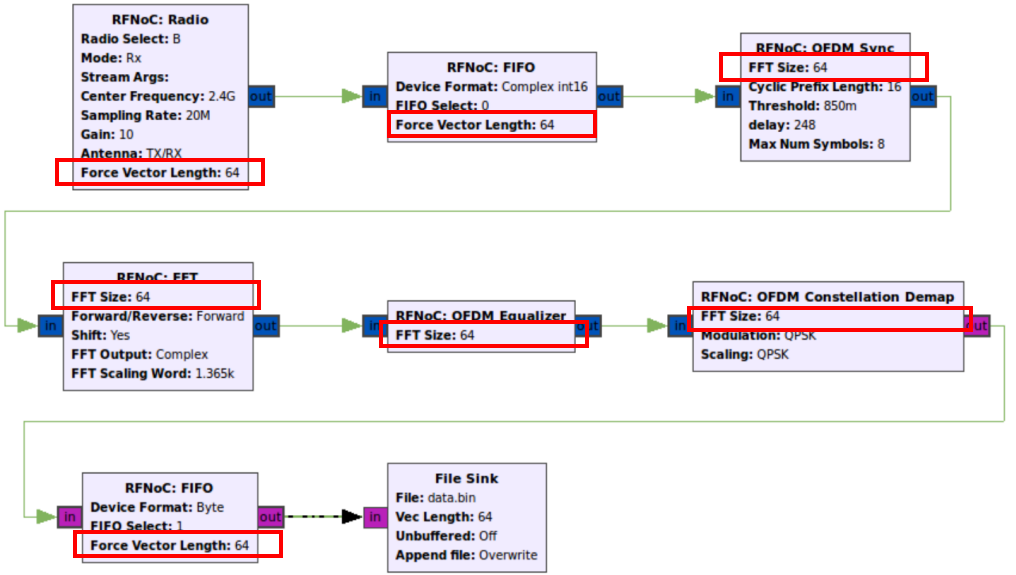
Problem

- Parameter dependencies between blocks
- Example: FFT Size

Solution

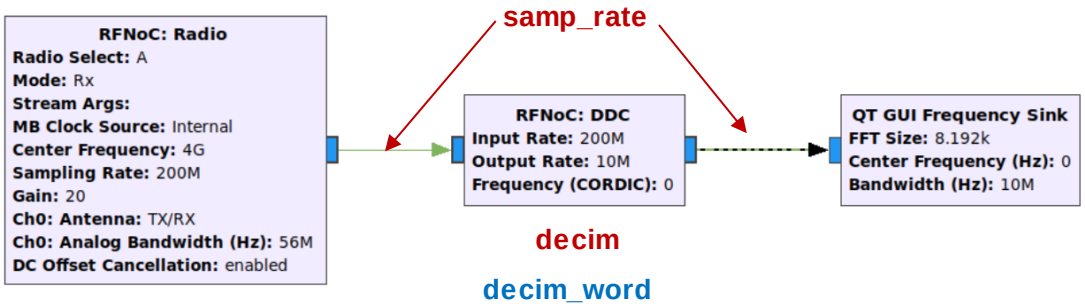
Brand new graph-propagation framework to handle user-specified parameter dependencies

- Sample Rate
- Sample Format
- Vector Width / FFT Size
- …



NATIONAL INSTRUMENTS

RFNoC Software Graph Updates

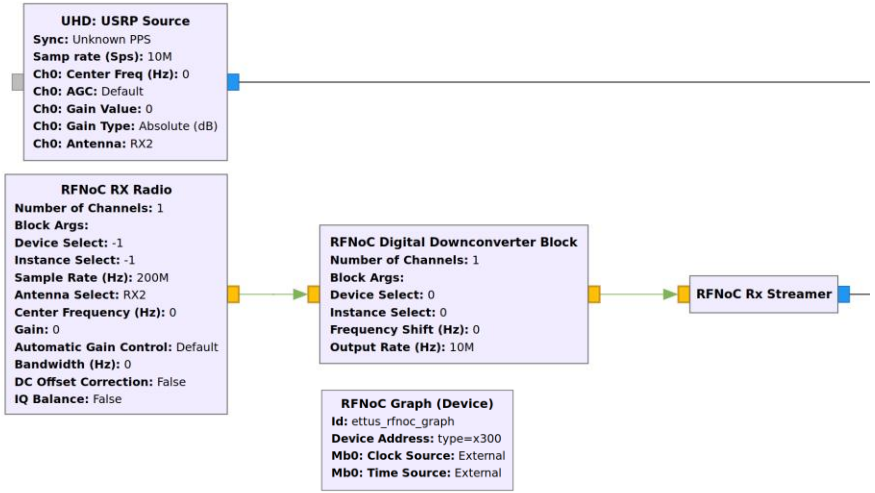


DDC Block Controller Software Example

- Define Registers: decim_word
- Define Properties: samp_rate, decim
- Define Resolver Functions:
  - decim => decim_word
  - Input samp_rate => Output samp_rate => decim property

Other RFNoC Software Updates

- Full multi_usrp compatibility
- Less property tree, more C++ APIs
- Separate access to Block Controllers & Motherboard Controllers can be accessed separately
  - Example: Setting clock/time reference is a motherboard feature, not a block feature
- FPGA Topology Detection
- Overhauled streaming subsystem



NATIONAL INSTRUMENTS

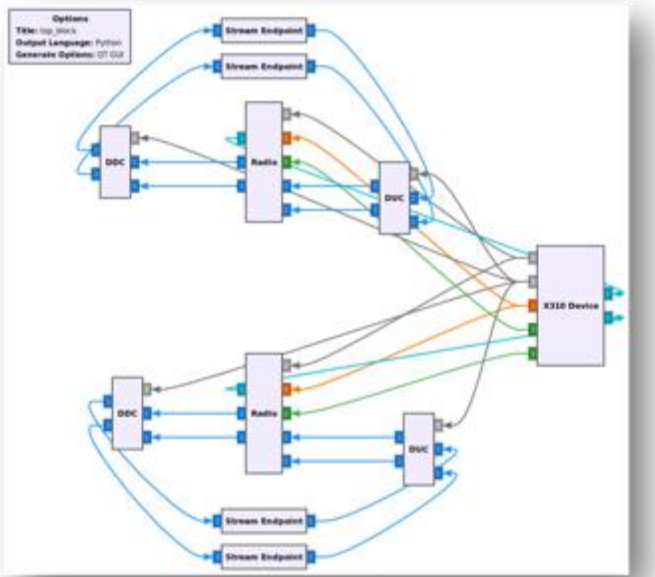
Toolflow Updates

RFNoC Image Builder

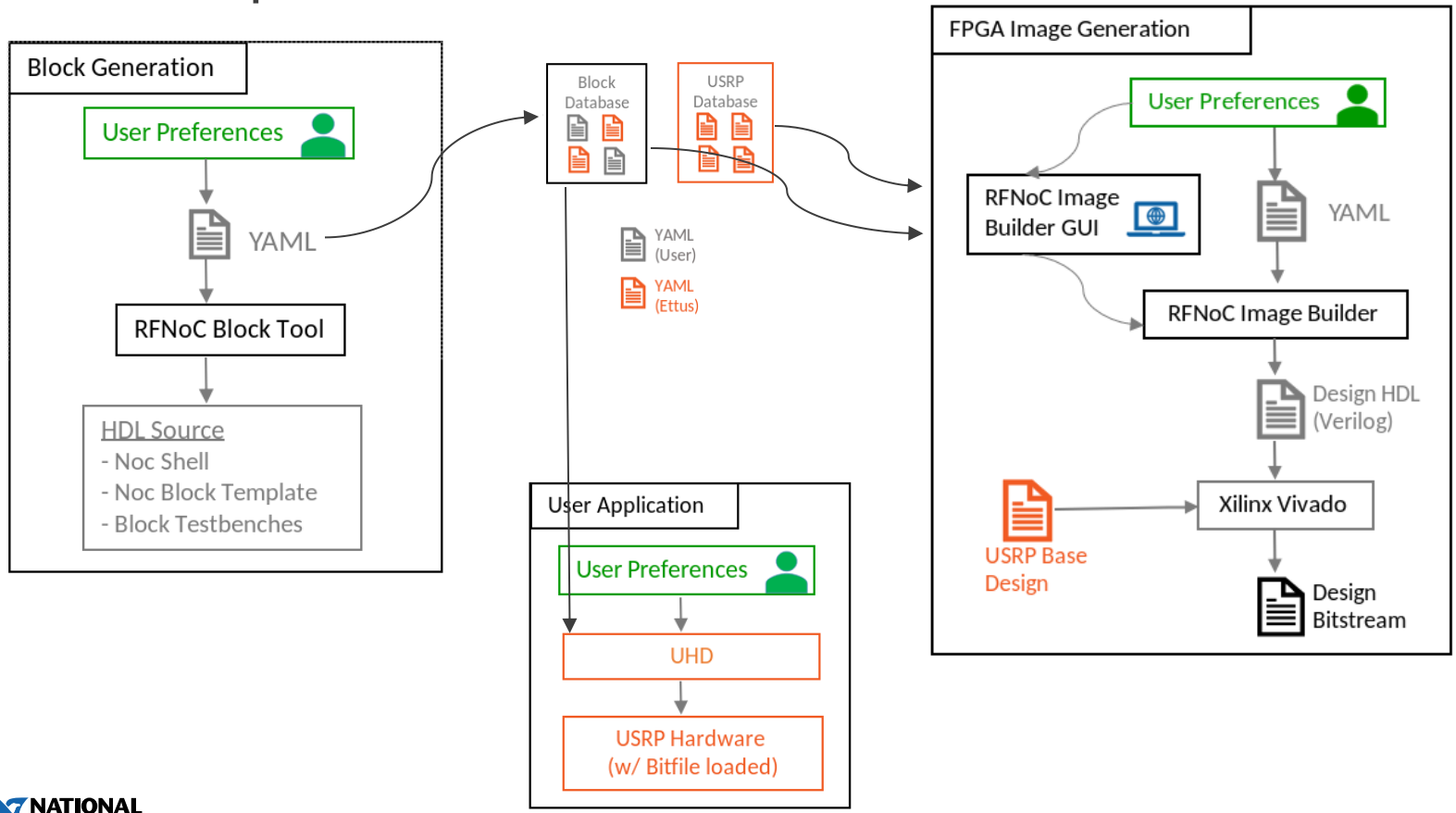
- Python based design assembly tool to help with FPGA image creation
- Create a synthesizable NoC Core instance based on user preferences
  - Stream Endpoints
  - Blocks and Static Topology
  - Additional IO connections
- Generates FPGA bitfile with topology and preference info embedded in the design
- Command line and GUI (GRC) support

Blocktool

- Utility for creating new RFNoC blocks
- Now ships with UHD



Toolflow Updates



NATIONAL INSTRUMENTS

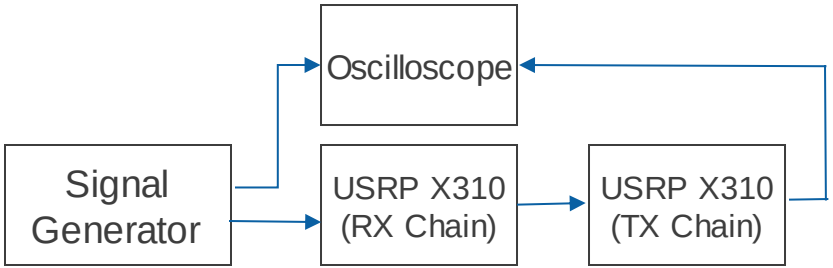
GRC Demo

Two RFNoC Example Flowgraphs in GNU Radio Companion

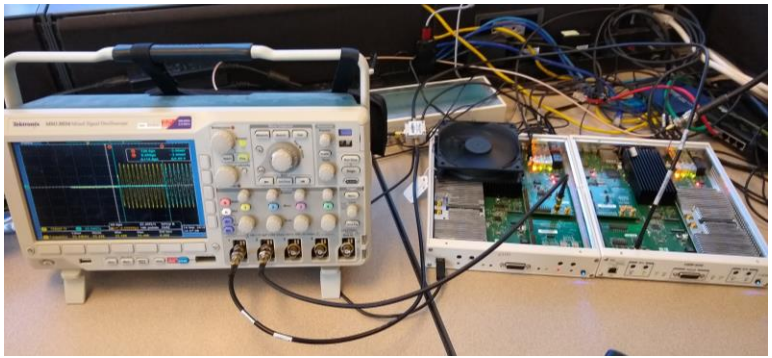
NATIONAL INSTRUMENTS

RFNoC Loopback with all Peripherals

- GRC Flowgraph to receive a tone on one motherboard, and transmit it out another motherboard
- Sent over an Aurora link from RX to TX; no host involved in the data path
- Minimizes latency
- Can run at full sample rate regardless of your host computer



Loopback Block Diagram

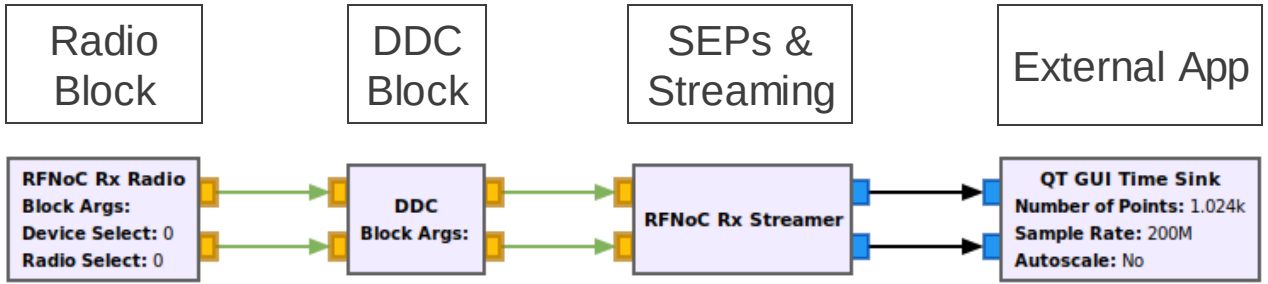


Loopback Setup

NATIONAL INSTRUMENTS

GNU Radio Integration

- GRC integration matches data flow changes
- RX/TX Streamers now become explicit Egress/Ingress
- Removes issue with single block in GR Graph
- Enables phase/time alignment for arbitrary configurations

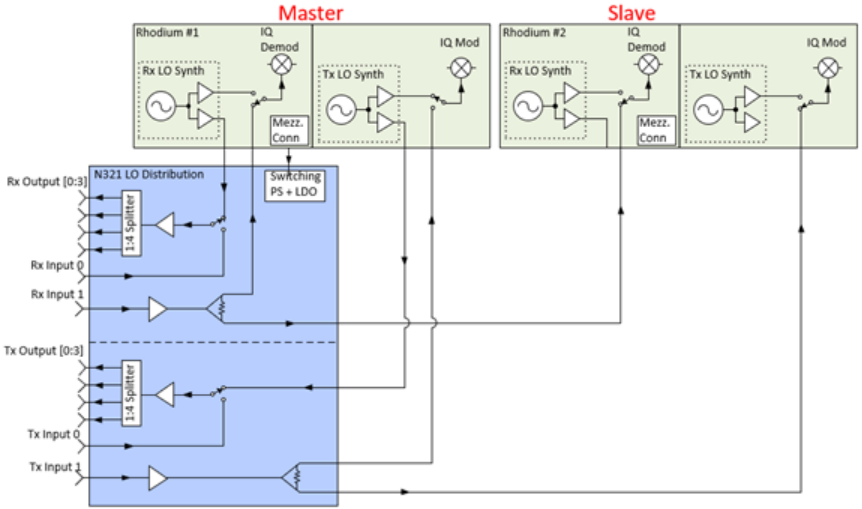


NATIONAL INSTRUMENTS

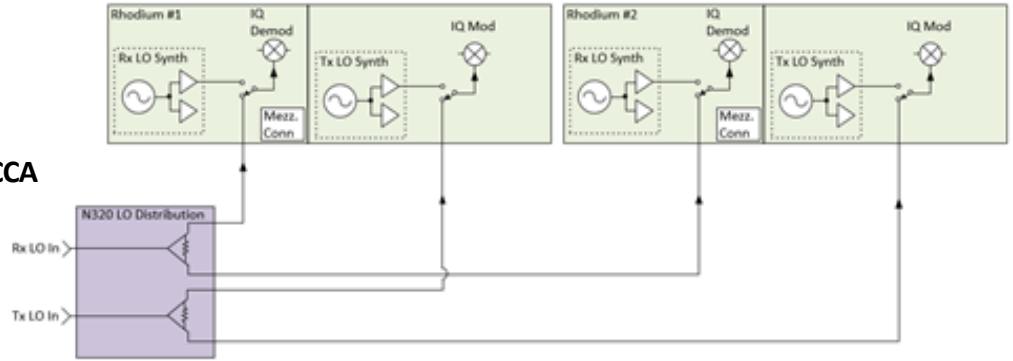
Thank You!

N320/N321 Overview

Active LO CCA (N321)

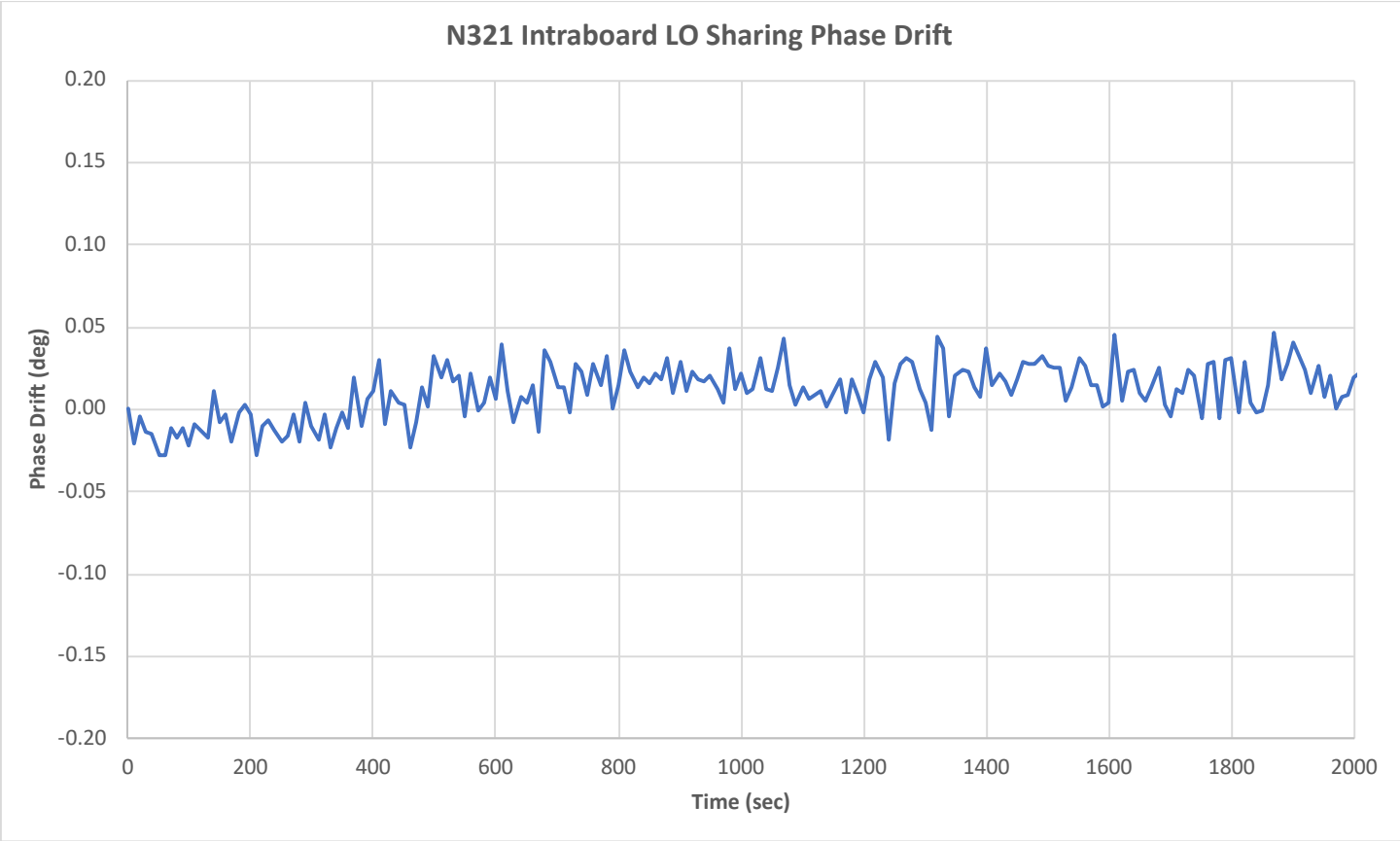


Passive LO CCA (N320)

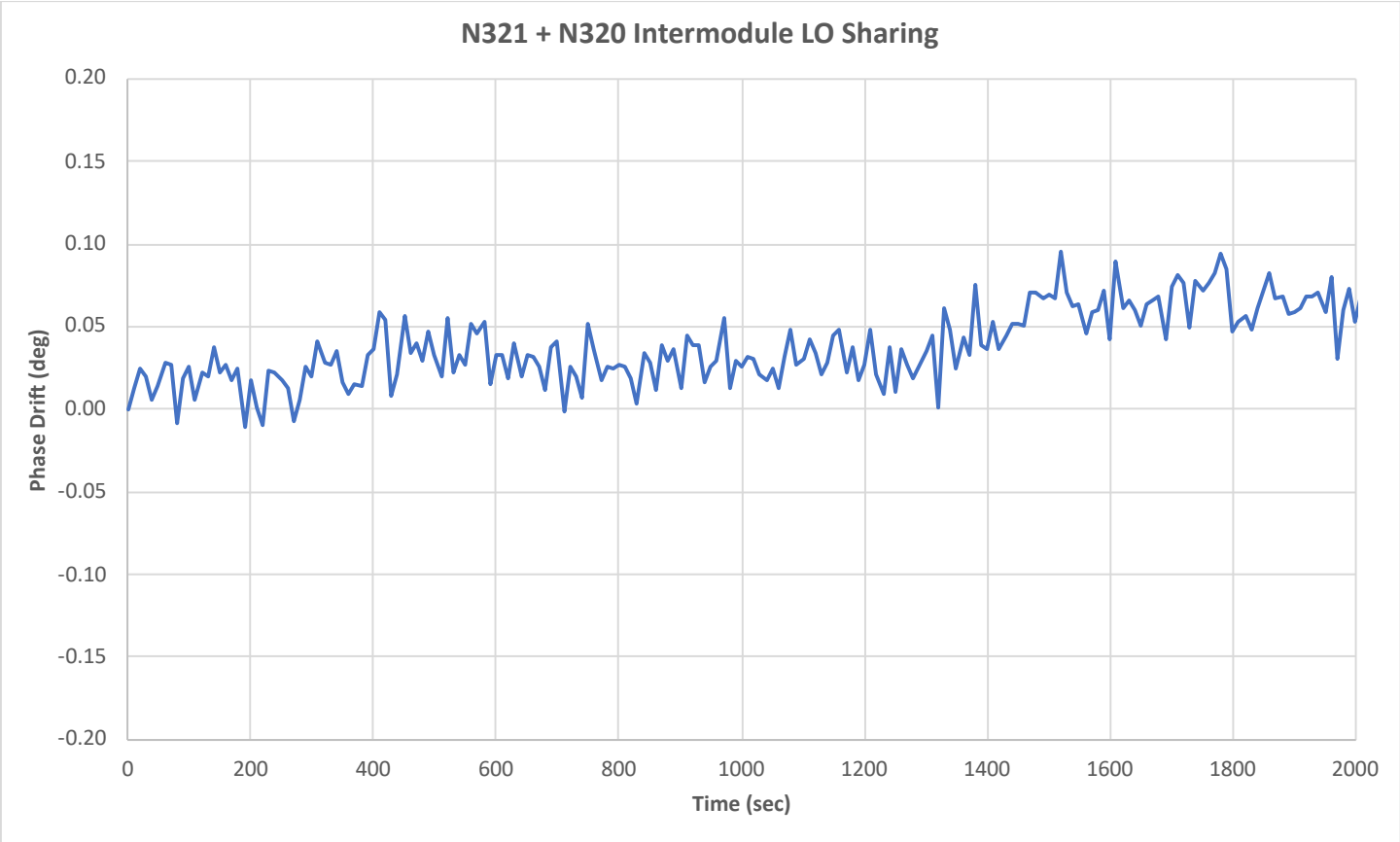


NATIONAL INSTRUMENTS

Phase Drift with LO Sharing (at 4 GHz): Within 1 Module



Phase Drift with LO Sharing (at 4 GHz): Across Two Modules



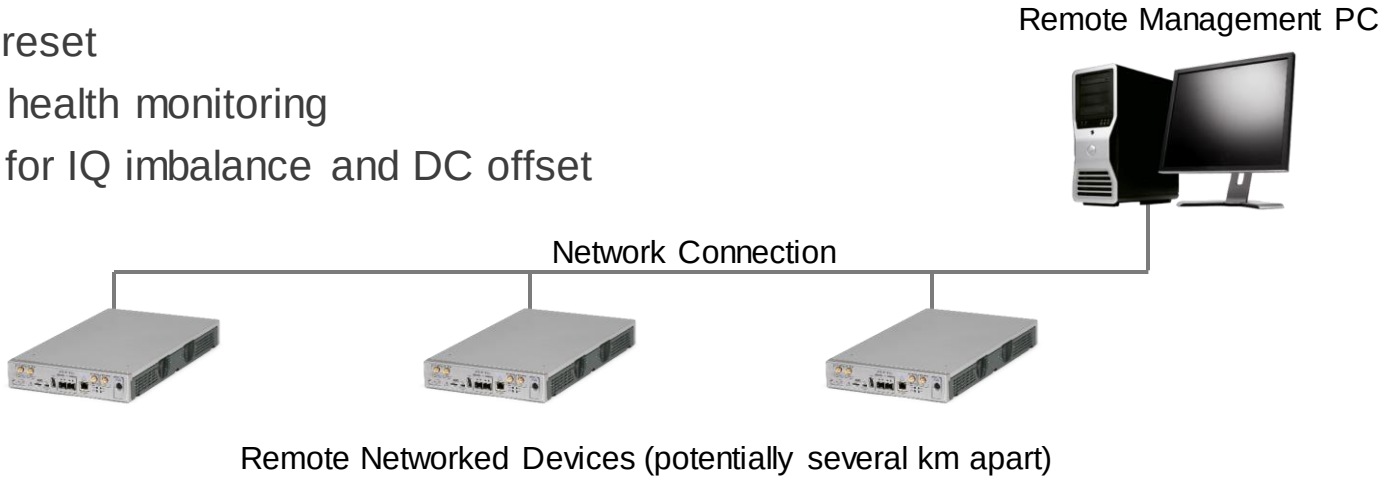
Centralized Remote Management

- Open source software development (no LV support)
- Single host application remotely manages entire
- Manage radios while streaming/processing data
- Remote firmware & OS updates
- Remote host PC and ARM application debugging
- Remote reboot
- Remote factory reset
- Remote system health monitoring
- Self corrections for IQ imbalance and DC offset

Central Management

Remote Administration

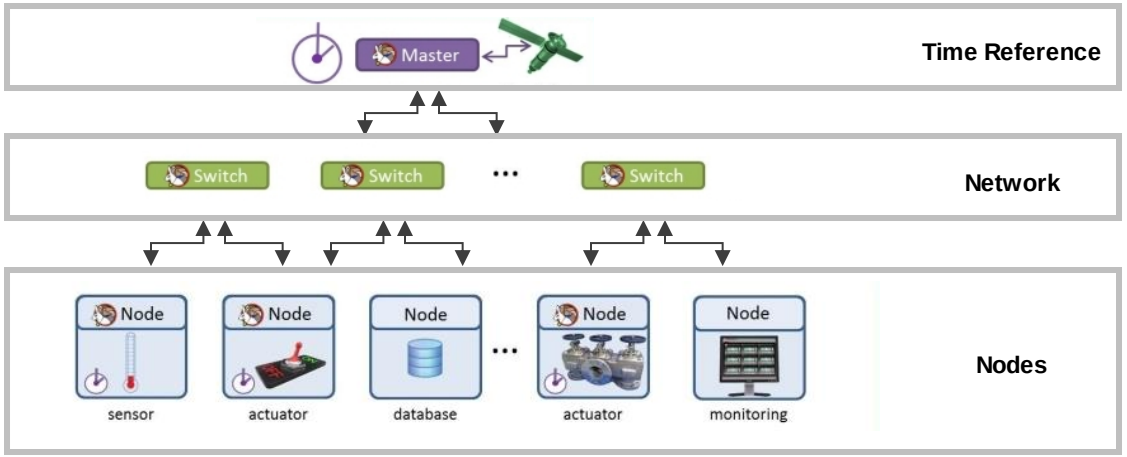
System Diagnostics



NATIONAL INSTRUMENTS

Ethernet-Based Synchronization

- White Rabbit implementation
- Geographically distributed systems
- GPS denied environments
- Based on IEEE 1588 and SyncE
- Sub ns skew, sub ps jitter
- Up to 10 km
- Scalable beyond 2000 nodes
- Open source

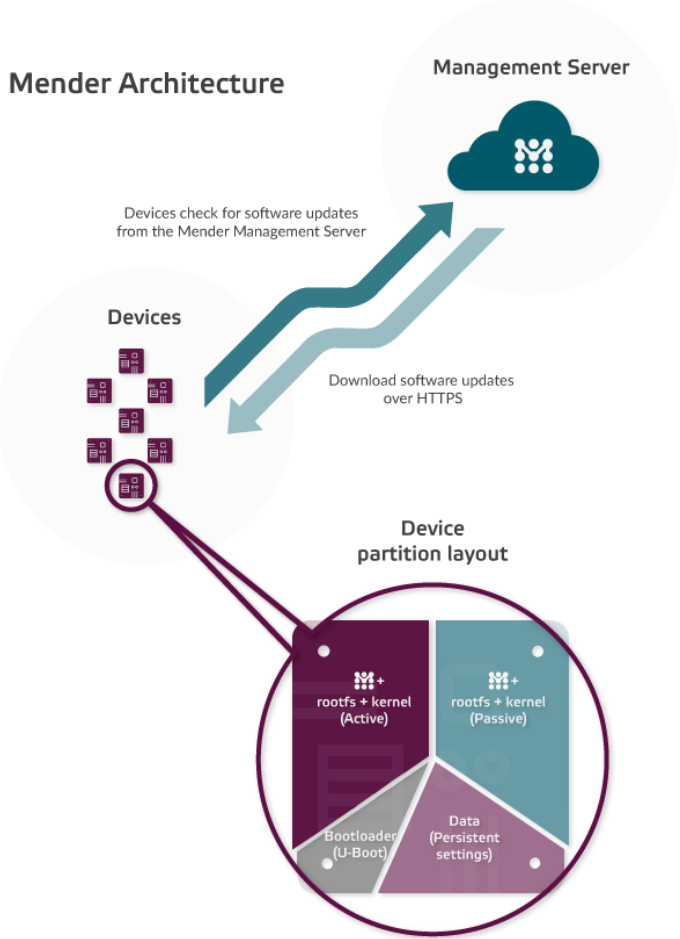


NATIONAL INSTRUMENTS

Remote Software Updates

- Open source license under Apache 2.0
- Modularity and maintainability
- Software updates for embedded Linux devices over the air or over ethernet
- Client and management server
- Designed for Yocto images such as Open Embedded Linux on USRP devices

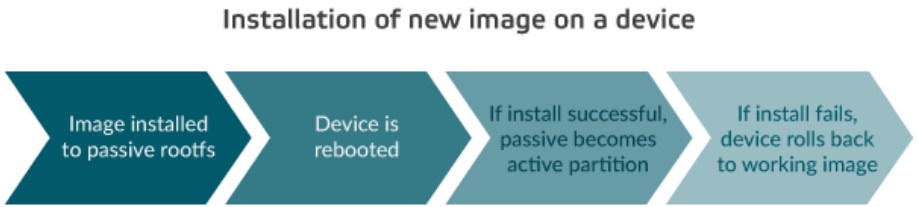
MENDER.io



NATIONAL INSTRUMENTS

Robust and Secure Updates

- Dual redundant partition scheme
- “Active partition” in use, “inactive partition” as backup
- Bootloader switches partitions after update
- Boot failure determined by checksum
- Commit or rollback
- Sign and verify updates with cryptographic keys (RSA, ECDSA256)
- Authentication between device and server



NATIONAL INSTRUMENTS