Multichannel RF Reference Architecture

Contents

- 1 Application Note Number and Authors
- 2 Revision History
- 3 Architecture Overview
  - 3.1 Hardware Overview
    - 3.1.1 Required NI/Ettus Research Hardware
    - 3.1.2 Validated Hardware
  - 3.2 Software Overview
- 4 Building the System
  - 4.1 System Assembly Best Practices
  - 4.2 Cabling the Hardware
    - 4.2.1 (Optional) Connecting the RF RX-TX Ports for a MIMO Loopback
    - 4.2.2 Connecting 10 MHz and PPS
    - 4.2.3 Connecting the LO Distribution
    - 4.2.4 Connecting the USRP Management Port
    - 4.2.5 Connecting the SFP+ Ports
    - 4.2.6 Connecting to Power
    - 4.2.7 (Optional) Installing and Configuring DPDK
  - 4.3 Installing and Configuring the Software
  - 4.4 Setting Up the First USRP
  - 4.5 Setting Up Additional USRPs
  - 4.6 Optimizing the Host System
- 5 Using the Software
  - 5.1 Example Source Code
  - 5.2 Building Examples
  - 5.3 Configuration Files
  - 5.4 Running the Examples
  - 5.5 Generating Waveform Data Files
  - 5.6 Viewing Data Files
- 6 Measured Performance
  - 6.1 Definitions
  - 6.2 System Characteristics
  - 6.3 Synchronization
    - 6.3.1 Synchronization Terminology
    - 6.3.2 Synchronization Methodology
    - 6.3.3 Rx Phase Coherence One-Hour Stability[2]
    - 6.3.4 Short-Term Rx-Rx Phase Coherence[3]
    - 6.3.5 Repeatability of Rx-Rx Phase Coherence
      - 6.3.5.1 Resetting Session between Runs[4]
      - 6.3.5.2 Resetting Configuration between Runs[5]
    - 6.3.6 Tx Phase Coherence One-Hour Stability
      - 6.3.6.1 Repeatability of Tx-Tx Phase Coherence
        - 6.3.6.1.1 Resetting Session between Runs [6]
        - 6.3.6.1.2 Resetting Configuration between Runs
  - 6.4 Streaming
    - 6.4.1 Streaming to Memory Data Transfer Rates[7]
    - 6.4.2 Streaming to Disk Data Transfer Rates[8]
    - 6.4.3 Streaming Simultaneously to/from Disk Rates[9]
    - 6.4.4 Streaming Simultaneously with DPDK to/from Disk Rates[10]
- 7 Appendix: Creating High?Channel?Count Systems
  - 7.1 Connection Considerations for High?Channel?Count Systems
  - 7.2 LO Distribution for High?Channel?Count Systems
- 8 Additional Resources
  - 8.1 Related Documentation
  - 8.2 Training Resources
- 9 Footnotes

AN-822 by Dylan Baros, Michael Dickens, Joseph Paller, Neel Pandeya, and Jovian Wysocki

This document provides guidance for designing a system that uses the NI Multichannel RF Reference Architecture version 1.2.

This document describes how to design a system that enables synchronization between Receive (Rx) channels, Transmit (Tx) channels, and Transmit-Receive (Tx-Rx) channels, as well as high-throughput data sample transport from the software-defined radio (SDR) to a server system over 10 GbE data links. This document provides a starting point in the creation of your specific system configuration. A basic understanding of Linux, command line, C++, Git, and networking is required to build and develop on the NI Multichannel RF Reference Architecture system.

The Multichannel RF Reference Architecture allows engineers to quickly scale from software simulation to hardware demonstration. It features direct conversion RF sampling (200 MHz instantaneous bandwidth) with NI USRPs for multiple RF applications, enabling coherent operation through shared local oscillator (LO), PPS, and 10 MHz reference clock signals.

Figure 1. System Overview



The architecture features the following components:

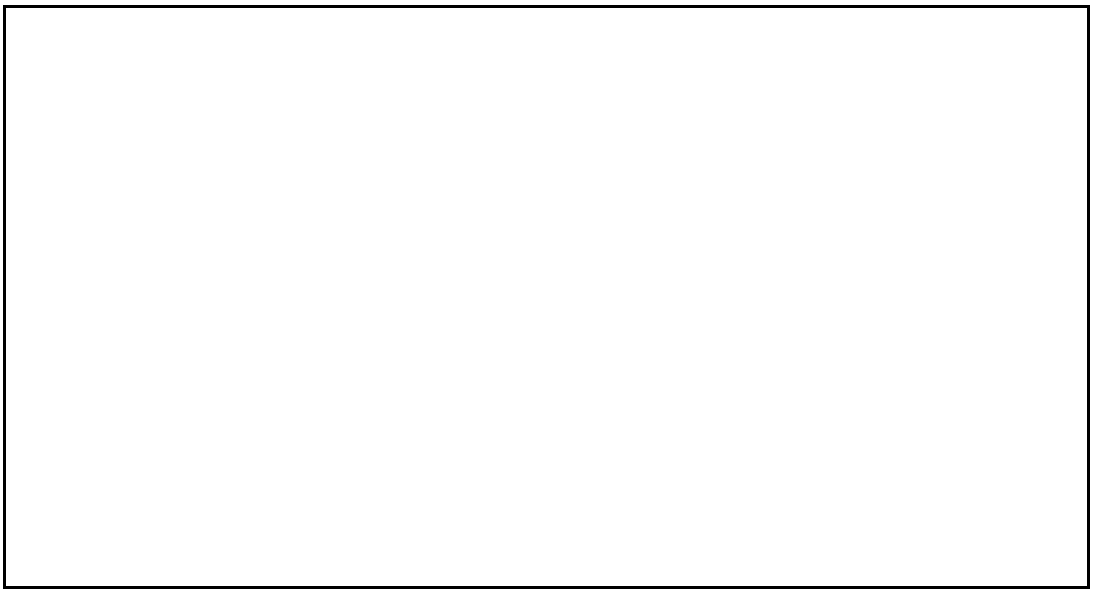
- Document, bill of materials, and interconnect diagrams that describe how to set up a system with up to 32 channels of Rx-Rx, Tx-Rx, and Tx-Tx synchronization.
- Reference library that contains functions to configure and control the USRP N321s and USRP N320s to maintain phase coherence between multiple channels.
- Reference library that contains functions to configure and control the USRP N321s and USRP N320s for high-rate data transfer from multiple channels to memory or storage.
- C++ examples that demonstrate how to use reference library functions to configure and control a multichannel system.
- Documentation that lists installation and configuration information for a 32x32 system, as well as best practices for development and validation of the system. *N*x*M* refers to systems, where *N* is the number of transmit channels and *M* is the number of receive channels.
- Documentation that lists performance benchmarks with validated results for systems with up to 32 channels for the following specifications:

  o Synchronization between 32 Rx-Rx channels, 16 Tx-Tx channels, and 32 Tx-Rx channels
  o Bidirectional streaming of IQ data between 32 channels and server

- Bill of materials for systems with up to 32 channels.

The reference architecture demonstrates using RFNoC to transmit from both the replay block as well as the host.

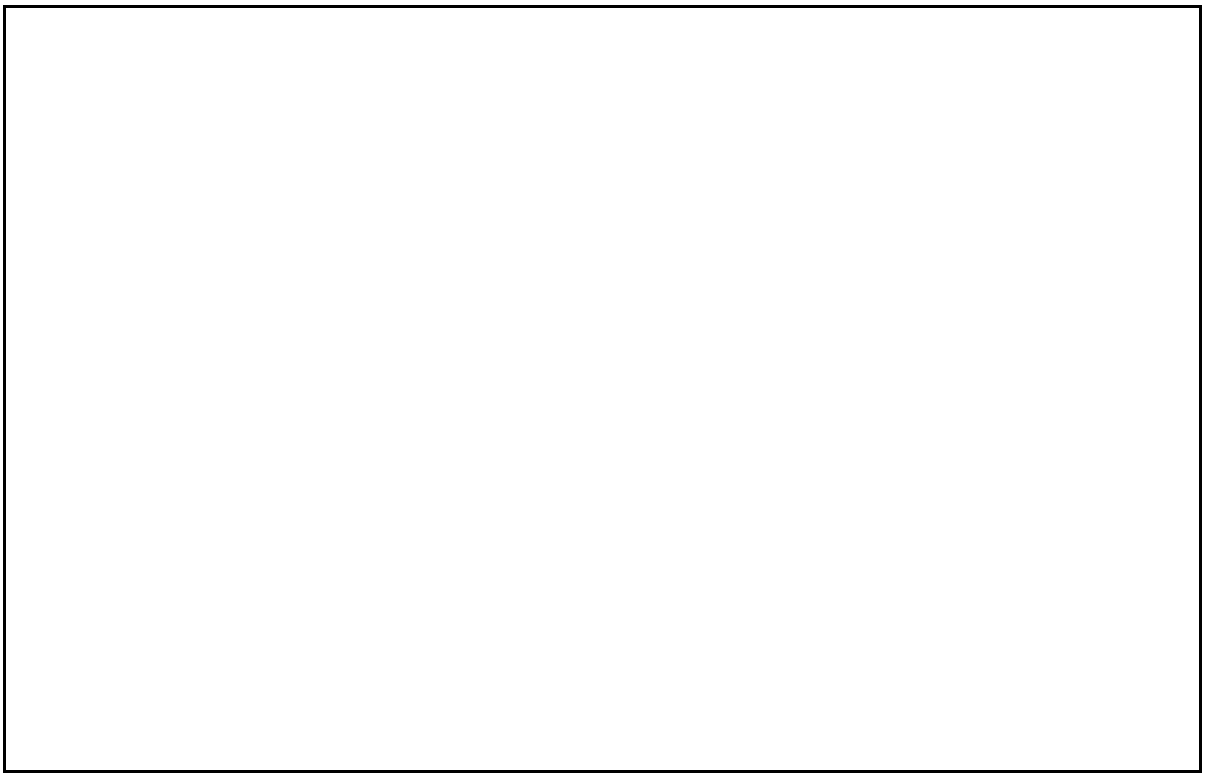
Figure 2. Reference Architecture Overview



The reference architecture uses USRP N321s and USRP N320s that scale from 2 to 32 phase-coherent transmit and receive channels.

The following diagram provides an overview of the topography of the connections and primary hardware components.

Figure 3. Hardware Overview



For a complete list of required hardware for the Multichannel RF Reference Architecture, refer to the bill of materials in the *<REFARCH>*/docs/ folder, where *<REFARCH>* is an alias for the following GitHub file folder location:
https://github.com/EttusResearch/refarch-multich

This section describes the NI/Ettus Research products required for generating and acquiring RF signals as well as the synchronization signals?PPS, 10 MHz, and Local Oscillator (LO)?shown in Figure 3:

- **USRP N321**?Networked software-defined radio with 2-channel RF front end and 250 MSample/s on both Tx and Rx. Features a power splitter for LO distribution, 10 MHz shared reference clock for coherent MIMO functionality, shared PPS for enabling shared triggering, dual 10 GbE ports for data transfer, and an RJ45 management port for device management.
- **USRP N320**?Networked software-defined radio with 2-channel RF front end and 250 MSample/s on both Tx and Rx. Features LO sharing and 10 MHz shared reference clock for coherent MIMO functionality, shared PPS for enabling shared triggering, dual 10 GbE ports for data transfer, and an RJ45 management port for device management.
- **CDA-2990 with GPSDO**?Clock distribution device called an OctoClock that generates and distributes the GPSDO signals?10 MHz reference (REF) and 1 Hz (PPS)?and features a management port.
- **NI RF Torque Screwdriver and SMA Driver Bit** (NI part number 780895-01)?A torque screwdriver is *strongly* recommended to securely tighten all connections to/from the USRPs in the system. You can also order the RF SMA Driver Bit Only (NI part number 780894-01).

Testing and deploying a system requires hardware from both NI and third-party vendors. The following component options have been validated and are recommended for most Multichannel RF Reference Architecture systems:

- Server

  o **Supermicro 4124GS-TNR**?Server with dual AMD EPYC processors and 10 PCIe 4 x16 slots. Numerous PCIe lanes from the CPU support many 10 GbE connections and the PCIe processing required for high-channel-count systems.
  o **Trenton Systems 3U BAM**?Server with Intel dual 3rd generation Xeon Ice Lake processors, 11 Gen 4.0 PCIe slots (5 x16 slots, 6 x8 slots), and 24x DDR4-3200 ECC RDIMM slots for high-channel-count systems.

- Network Interface Cards (NIC)

  o **Intel X710-DA4**?Network card that supports x4 10 GbE connections.
  o **Intel E810-CQDA2**?Network card that supports two 100 GbE connections, or, with a breakout cable, x8 10 GbE connections.
  o **NVIDIA MCX4121A-ACAT ConnectX-4 Lx EN**?Network card that supports two 25 GbE connections.

- Storage

  o **HighPoint SSD7505**?PCIe carrier board that holds x4 M.2 drives and supports either a software or hardware RAID configuration.
  o **Sabrent 2TB Rocket 4 PLUS**?NVME M.2 SSD drive that supports consecutive writes rates of over 6 GB/s.

- Power Splitter/Combiners

  o **Mini-Circuits 8-Way ZN8PD1-63W-S+**?DC pass, 8-way -0°, 50 ?, 500 MHz to 6,000 MHz power splitter/combiner.
  o **Mini-Circuits 4-Way ZN4PD1-63HP-S+**?High power, DC pass, 4-way -0°, 50 ?, 30 W, 500 MHz to 6,000 MHz power splitter/combiner.

- Cabling

  o **SMA Cable, Loop Back Cable Kit** (NI part number 782781-01)?Loopback cable kit includes 2 SMA-M to SMA-M cables (60 cm/2 ft) and 2 SMA-F to SMA-M Attenuators (30 dB, 50 Ohm, DC-6 GHz).
  o **SFP+ Cable, 10 Gigabit Ethernet Cable w/ SFP+ Terminations (1 Meter)** (NI part number 783343-01)?SFP+ cable recommended for USRPs with 10 GigE interfaces. It can be used to connect the USRP to a 10 GigE switch or network adapter.

- Rack Accessories

- **Tripp Lite Rackmount Rack Extension Bracket** (B018-000-1P5)?Bracket required to mount the USRPs in a rack.
- **Z-Bracket**?Bracket required to mount the USRPs in a rack to meet environmental specifications.

The reference software, shown in Figure 2, consists of a C++ reference library and examples, using USRP Hardware Driver (UHD) version 4.2. The reference software is supported on Ubuntu Linux 20.04. It demonstrates how to assemble multi-USRP systems through the UHD RFNoC API and replay block. It contains a Python-based binary int16 data file (.dat) generator and viewer. The reference software is written to be scalable for systems 2x2 to 32x32 and beyond, as well as Tx or Rx only (2 to 32 channels and beyond).

The software provides examples that stream data from the replay block as well as the host, as shown in the following figures.

Figure 4. Replay Block Data Movement

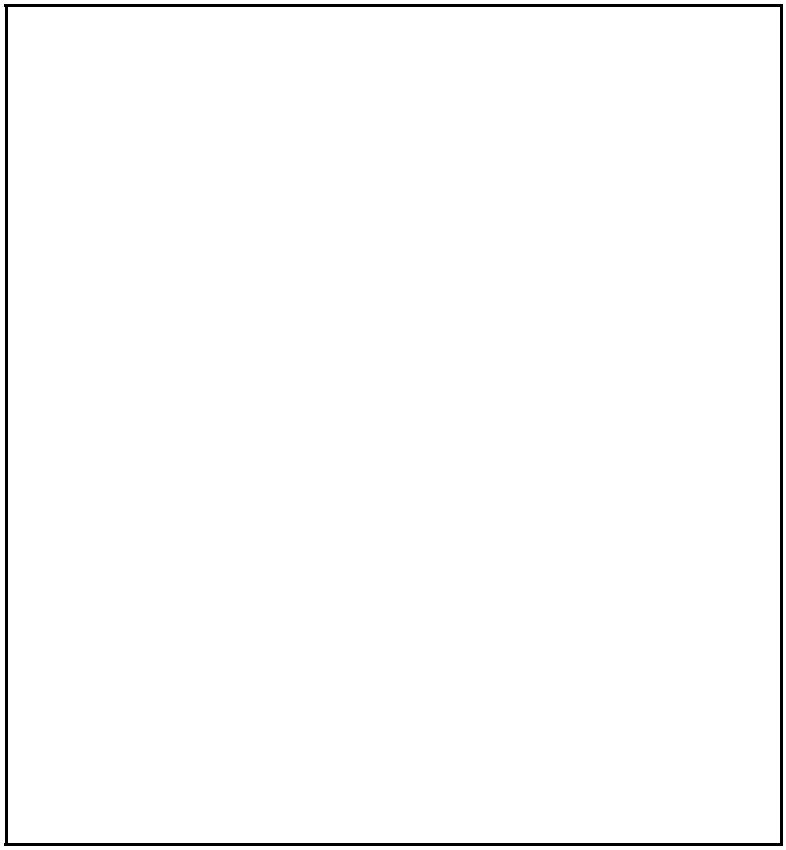
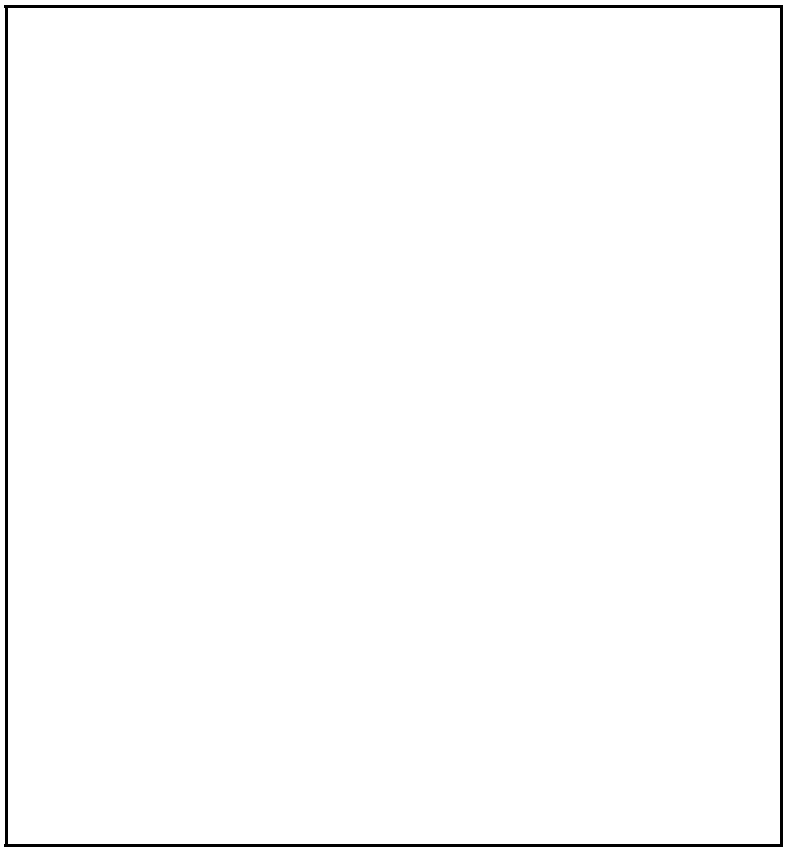


Figure 5. Stream from Host Data Movement



The following figures show a 32x32 system layout. NI recommends using a rack for this system to simplify connections between the different hardware components.

Figure 6. Multichannel RF Reference Architecture System

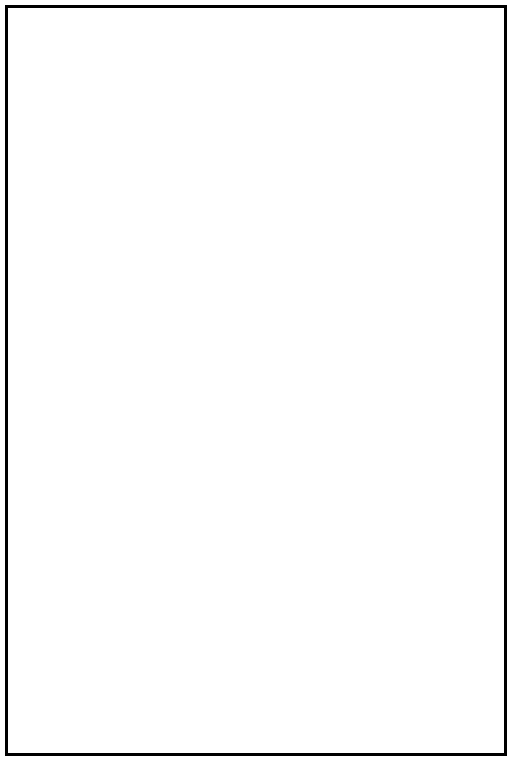


Figure 7. Rear View of System with Cabling

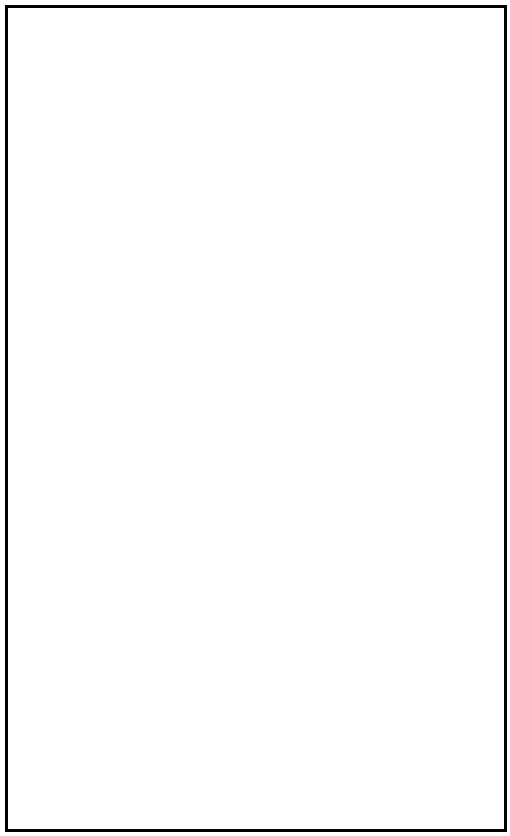
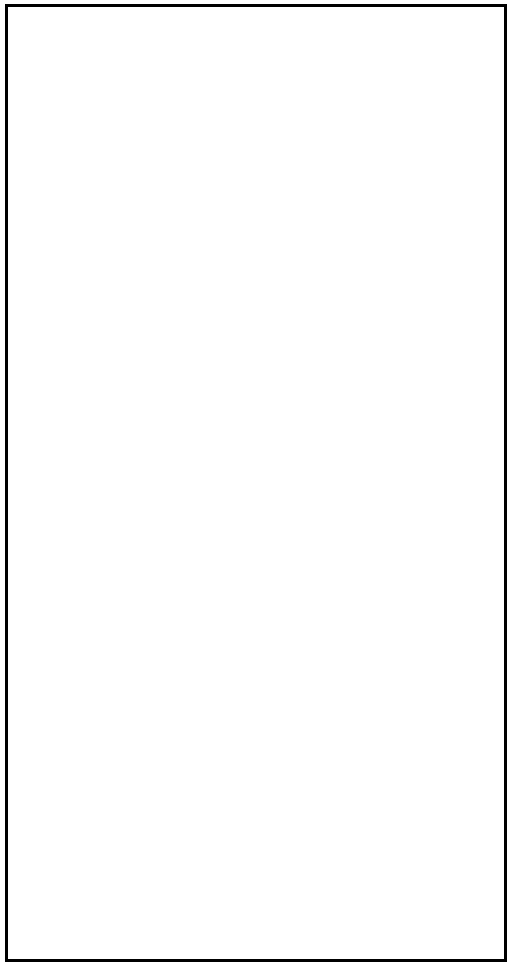


Figure 8. Rack Layout for 32x32 System



Adhere to the following best practices when installing, setting up, and connecting components to your system:

- Number and label the USRPs with serial numbers on the front of the devices. Visit *<REFARCH>/docs/32_Channel_Template_layout_BOM_and_wiring.ods* for a customizable template.
- While running the system in loopback, attach 30 dB attenuators to all output channels to prevent damage to the USRPs.
- Arrange the equipment in the rack to optimize proper airflow. USRP N321s and USRP N320s facilitate airflow from right to left, and the server facilitates airflow from front to back. NI recommends using Z-brackets to vertically stagger installation of USRPs.
- Position the lowest USRP in the rack a few units up from the floor to allow for adequate device connector accessibility.
- NI recommends positioning the USRPs halfway to 2/3 toward the rear of the rack to allow easier bundling of cables.
- Label the NICs and 10 GbE cables to identify each USRP connection, as described in *Connecting the SFP+ Ports*.
- Connect the RX LO OUT and TX LO OUT cabling from Distribution Tier of USRP N321s to the nearest USRP N320 devices, as described in *Connecting the LO Distribution*
- Keep the end-to-end path of each Rx, Tx, LO, PPS, and 10 MHz signal identical in length, using the shortest cable possible per tier, as described in *Cabling the Hardware*.
- Use a calibrated torque wrench to connect all SMA connections to industry standard torque measurements.
- Periodically check all cabling to ensure a secure connection.

The following topics describe the connections for 8x8, 16x16, and 32x32 systems.

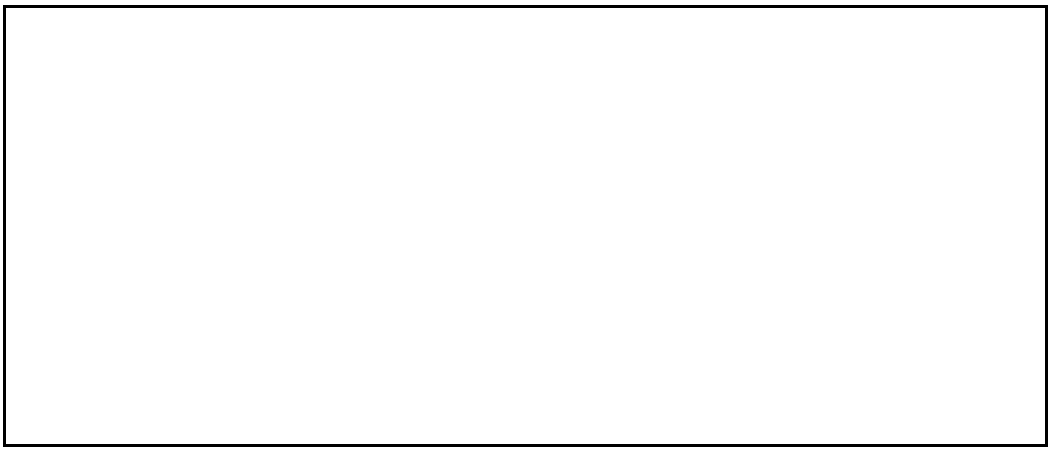
> **Note** Ensure that you use a calibrated torque wrench for all cabling connections. NI recommends the RF Torque Screwdriver and SMA Driver Bit (NI part number 780895-01).

You can use the MIMO loopback to validate the setup of the system and provide a baseline for synchronization calibration. Create the MIMO loopback by connecting the USRP TX/RX connectors (used for transmitting signals) to a combiner, connecting the combiner to a splitter, and connecting the splitter to the USRP RX2 connectors (used for receiving signals).

> **Notice** 30 dB attenuators must be used on every TX to ensure safe use of the MIMO loopback. Using the system without attenuators will overload the RX and will permanently damage the USRPs.
>
> **Note** NI recommends keeping all cables in the same tier the same length to reduce skew.

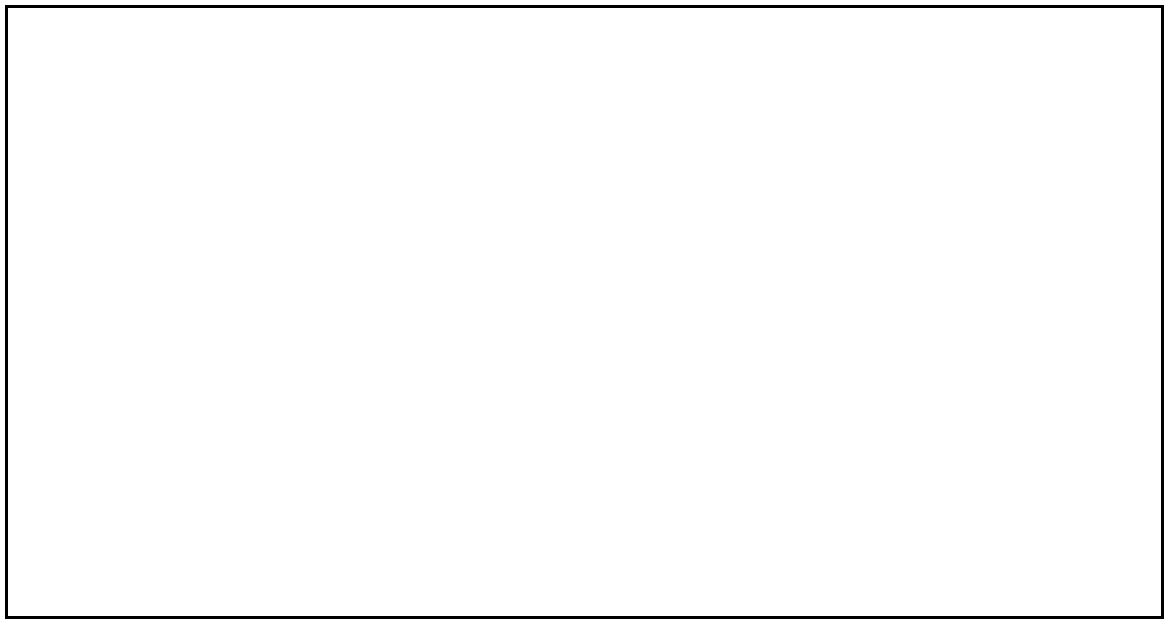
The 8x8 system requires one 8-way splitter and one 8-way combiner. To set up the MIMO loopback, connect the RF 0/1 TX/RX and RX2 connectors on the USRP to an 8-way splitter and an 8-way combiner, as shown in the following figure. All cables connecting between tiers must be the same length and type.

Figure 9. MIMO Loopback (8x8)



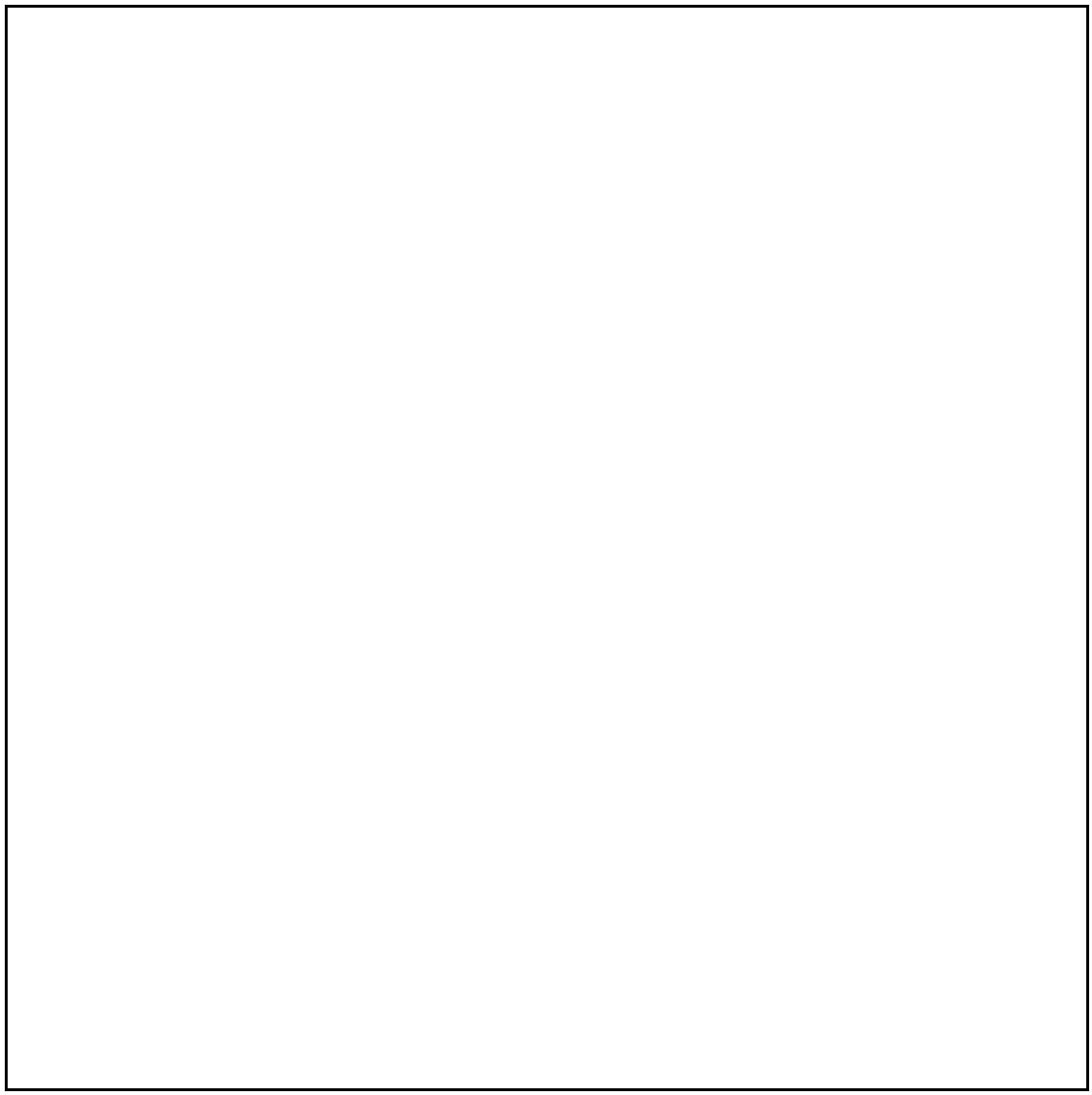
The 16x16 system requires two 8-way splitters, one 2-way splitter, two 8-way combiners, and one 2-way combiner. To set up the MIMO loopback, connect the RF 0/1 TX/RX and RX2 connectors on the USRP to an 8-way combiner and an 8-way splitter. Then connect the two 8-way combiners to a 2-way combiner and the two 8-way splitters to a 2-way splitter. All cables connecting between tiers, shown in the following figure, must be the same length and type.

Figure 10. MIMO Loopback (16x16)



The 32x32 system requires four 8-way splitters, one 4-way splitter, four 8-way combiners, and one 4-way combiner. To set up the MIMO loopback, connect the RF 0/1 TX/RX and RX2 connectors on the USRP to an 8-way combiner and an 8-way splitter. Then connect the four 8-way combiners to a 4-way combiner and the four 8-way splitters to a 4-way splitter. All cables connecting between tiers, shown in the following figure, must be the same length and type.

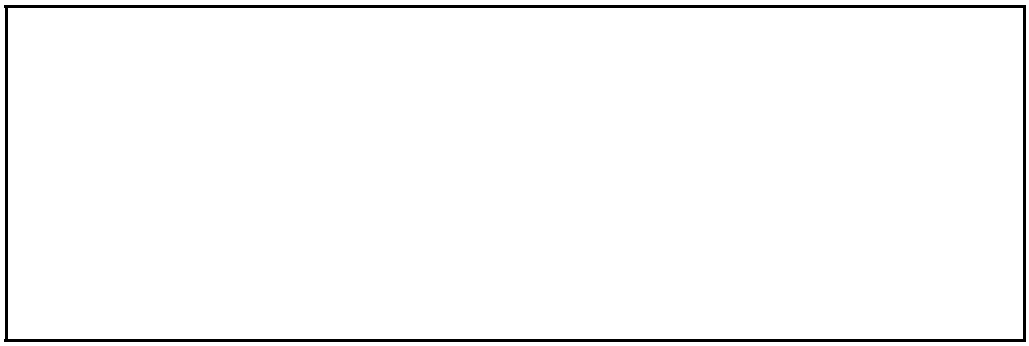
Figure 11. MIMO Loopback (32x32)



Connect a 10 MHz OUT connector on the OctoClock to the REF IN connector on the USRP. Connect a PPS OUT connector on the OctoClock to the PPS connector on the USRP.

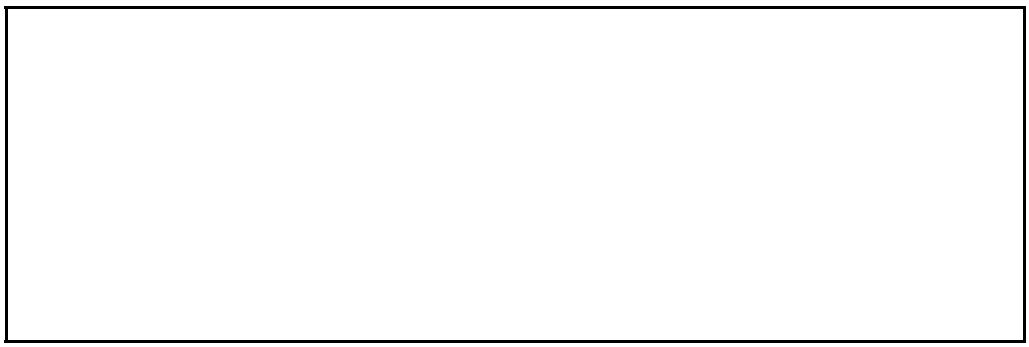
The 4x4 system requires one OctoClock with GPSDO, two USRPs, and uses two 10 MHz and two PPS OUT SMA connectors. Ensure that the source selection switch is set to internal on OctoClock 0 prior to power up.

Figure 12. 10 MHz and PPS Connections (4x4)



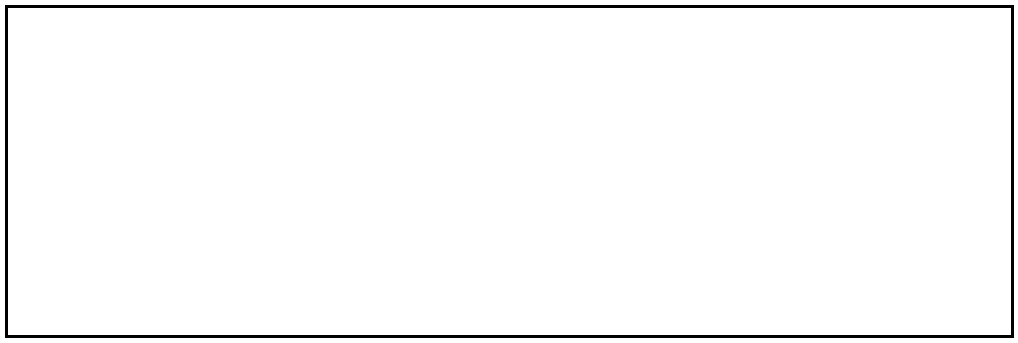
The 8x8 system requires one OctoClock with GPSDO, four USRPs, and uses four 10 MHz and four PPS OUT SMA connectors. Ensure that the source selection switch is set to internal on OctoClock 0 prior to power up.

Figure 13. 10 MHz and PPS Connections (8x8)



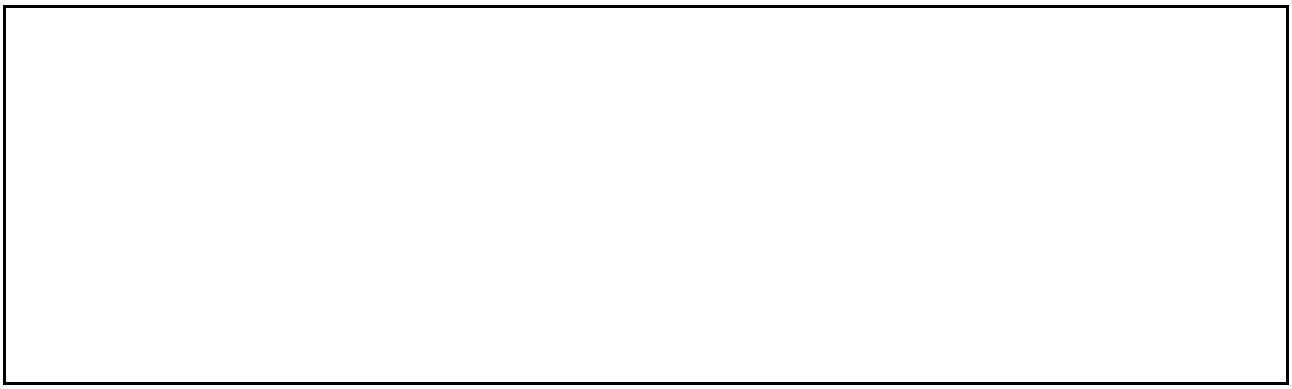
The 16x16 system setup requires one OctoClock with GPSDO, eight USRPs, and uses eight 10 MHz and eight PPS OUT SMA connectors. Ensure that the source selection switch is set to internal on OctoClock 0 prior to power up.

Figure 14. 10 MHz and PPS Connections (16x16)



Scaling to a 32x32 system requires one OctoClock with GPSDO (OctoClock 0, shown in the following figure), two additional OctoClocks (OctoClocks 1 and 2), and 16 USRPs. The source 10 MHz and PPS signals originate from OctoClock 0, which drives the OctoClock 1 and 2 inputs. OctoClock 1 and 2 each distribute all eight output 10 MHz and PPS signals to USRPs, 16 in aggregate. All cables connected from OctoClock 0 must be the same length. All cables connected to the USRPs must be the same length. Ensure that the source selection switch is set to internal on OctoClock 0, and that the source selection switch is set to external on OctoClocks 1 and 2. The selection switches must be set prior to power up.

Figure 15. 10 MHz and PPS Connections (32x32)



Connect the TX LO OUT, RX LO IN, and TX LO IN connectors on the USRPs for LO distribution.

> **Note** This configuration is used to synchronize both the Rx and Tx to the same LO resulting in the same center frequencies for both Tx and Rx. To synchronize different frequencies, connect RX LO OUT to RX LO IN and TX LO OUT to TX LO IN. Refer to *USRP N320/N321 LO Distribution* for connection diagrams.

While connecting the LO distribution, adhere to the following best practices:

- Tx and Rx must share the same LO connection. The LO sharing configuration for USRP N321s and USRP N320s in this system differs slightly from the LO sharing diagram on the Ettus Knowledge Base[1]. Distributing the LO from one channel to all other channels results in phase coherence across all channels, and there will be a constant non-zero phase offset between any two channels. This phase offset can be empirically measured and compensated for in software, thus resulting in all channels being phase aligned. The overall system is phase synchronous under a wide range of conditions; refer to *Synchronization* for more information about measured synchronization performance.
- Configure all USRPs, including the LO Source USRP, to use external LO inputs. *Refer to* `sync.cpp` in the *<REFARCH>*/lib/ folder for information about how the reference software configures the LOs on the USRPs.
- Use tiered LO signal distribution and identical cable lengths between any specific tiers.
- Connect the RX LO OUT and TX LO OUT cabling from Distribution Tier of USRP N321s to the nearest USRP N320s, as shown in the following figures.
- Use minimum identical cable length between tiers.

The following table lists the tiers for all system configurations.

Table 1. LO Source and Tiers

System	Tier(s)	USRP N321 Minimum Quantity	USRP N320 Maximum Quantity
2x2	LO Source (0)	1	?
4x4	LO Source (0), 1	1	1
8x8	LO Source (0), 1, 2	2	2
16x16	LO Source (0), 1, 2	3	5
32x32	LO Source (0), 1, 2, 3	6	10

The following diagrams represent LO distribution signal connections.

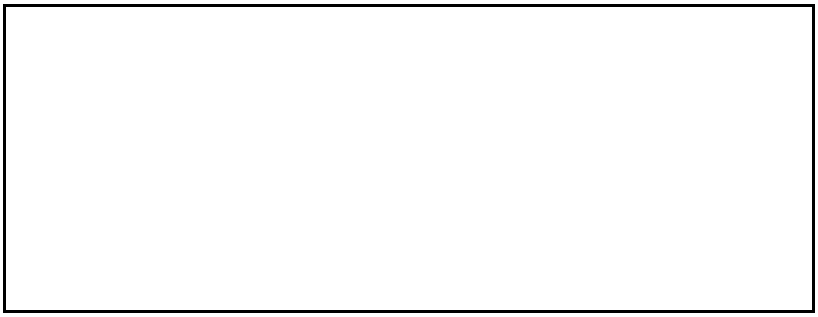
The 2x2 system requires two cables of identical length. Note that you cannot combine LO signals. NI recommends that all LO distribution cables originating from the same tier are the same length.

Figure 16. LO Sharing (2x2)



The 4x4 system requires four cables of identical length. Note that you cannot combine LO signals. NI recommends that all LO distribution cables originating from the same tier are the same length.

Figure 17. LO Sharing (4x4)



For the 8x8 system, both RX LO OUT cables originating from the LO Source USRP N321 must be the same length. All RX LO OUT and TX LO OUT cables originating from the Tier 1 USRP N321 must be the same length. Note that you cannot combine LO signals.

Figure 18. LO Sharing (8x8)

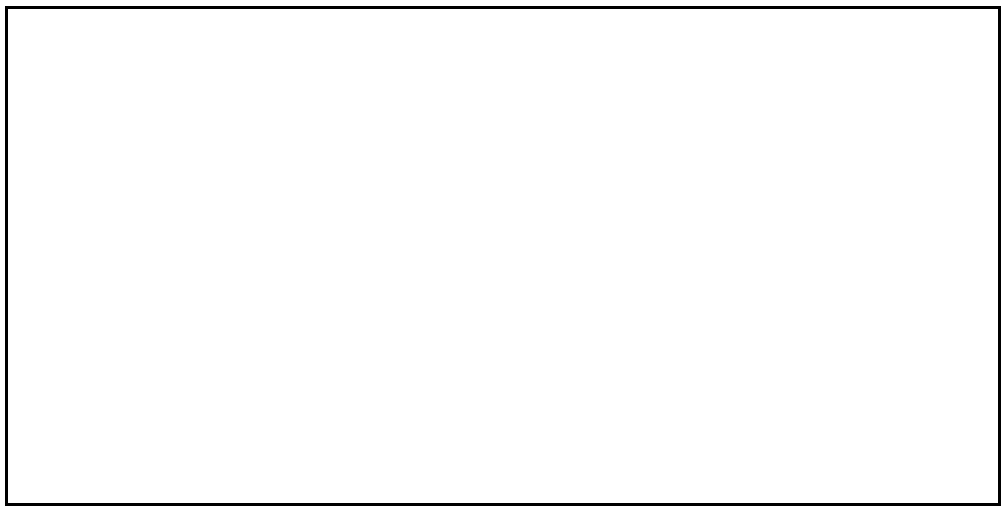


Table 2. LO Sharing Cabling (8x8)

Cable Color in Diagram	Cable Tier	Number of Cables in Tier
	LO Source Tier 0	2
	Distribution Tier 1, RX LO OUT	4
	Distribution Tier 1, TX LO OUT	4

For the 16x16 system, all RX LO OUT cables originating from the LO Source USRP N321 must be the same length. All RX LO OUT and TX LO OUT cables originating from Tier 1 USRP N321s must be the same length. Note that you cannot combine LO signals.

Figure 19. LO Sharing (16x16)

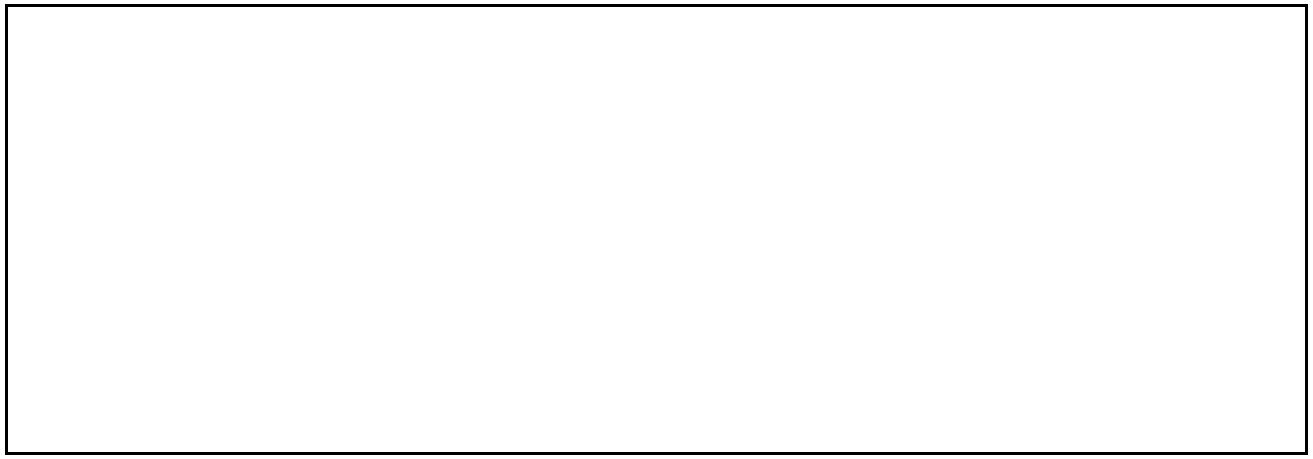


Table 3. LO Sharing Cabling (16x16)

Cable Color in Diagram	Cable Tier	Number of Cables in Tier
	LO Source Tier 0	4
	Distribution Tier, RX LO OUT	8
	Distribution Tier, TX LO OUT	8

For the 32x32 system, both RX LO OUT cables originating from the LO Source USRP N321 must be the same length. All RX LO OUT and TX LO OUT cables originating from the Tier 1 USRP N321 must be the same length. All RX LO OUT and TX LO OUT cables originating from the Tier 2 USRP N321s must be the same length. Note that you cannot combine LO signals.

Figure 20. LO Sharing (32x32)

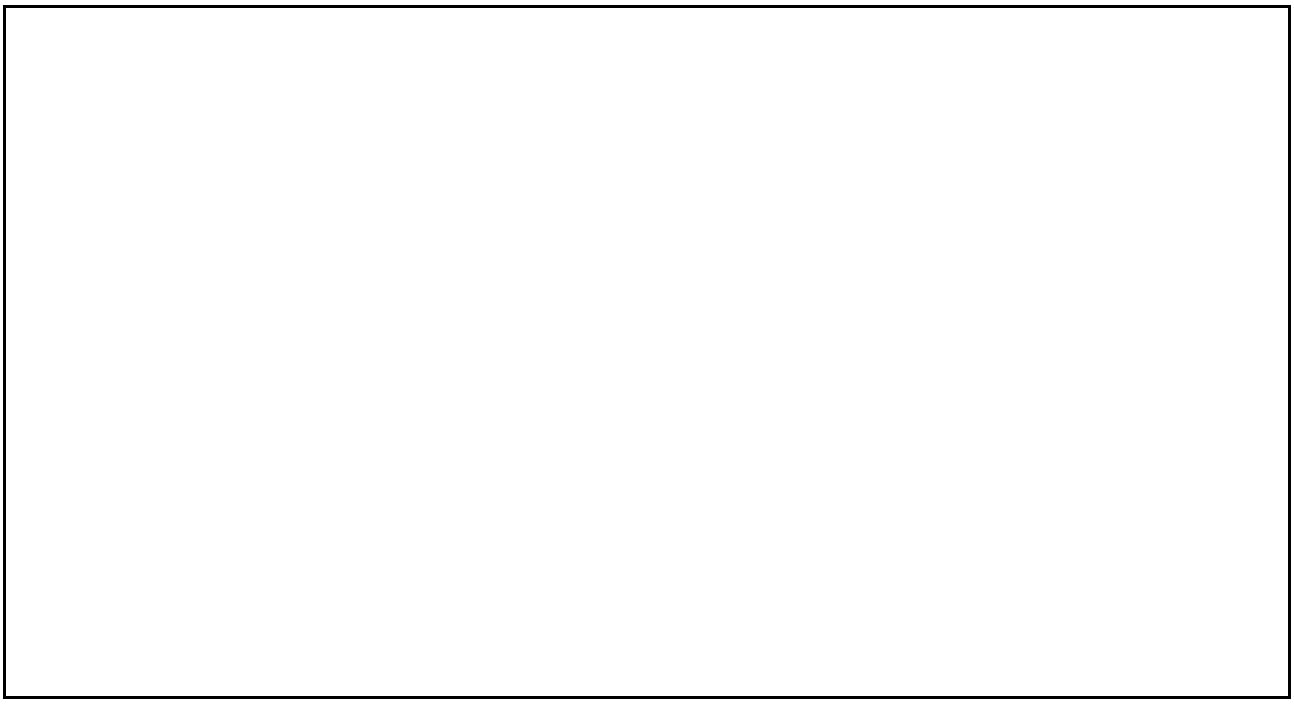


Table 4. LO Sharing Cabling (32x32)

Cable Color in Diagram	Cable Tier	Number of Cables in System
	LO Source Tier 0	2
	Distribution Tier 1, RX LO OUT	4
	Distribution Tier 1, TX LO OUT	4
	Distribution Tier 2, RX LO OUT	16
	Distribution Tier 2, TX LO OUT	16

Make the following RJ45 connections to allow USRP file system configuration and updates later in the process.

> **Note** The network port on the USRPs in this system use DHCP by default. You must either have a DHCP server running on an upstream networked device or set a static IP address for each USRP.

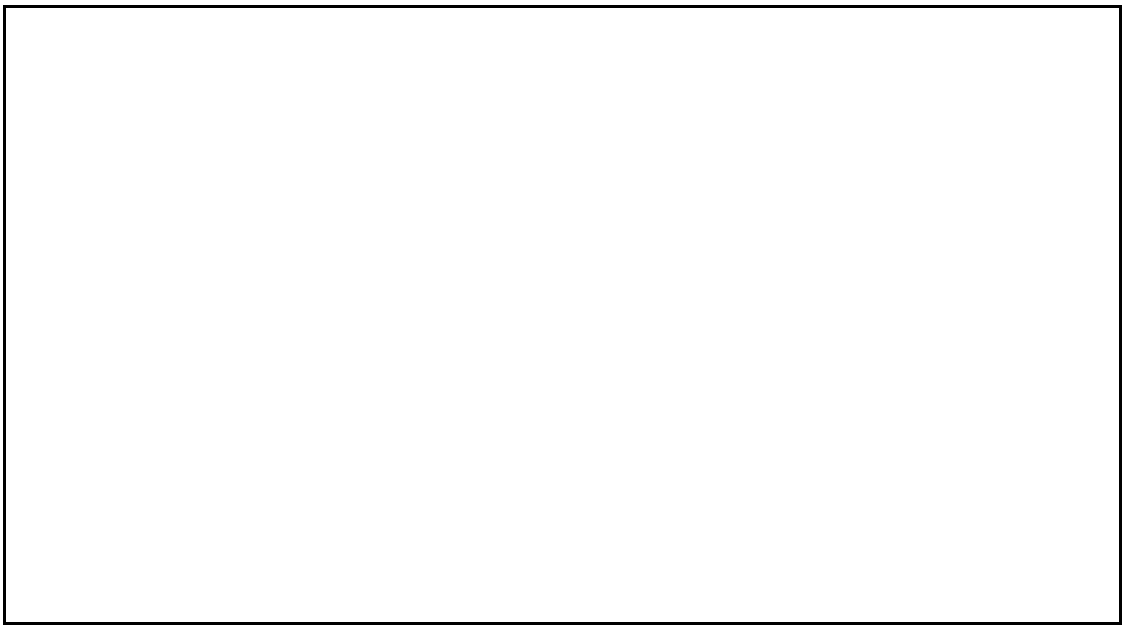
1. Connect an RJ45 cable between the server and the switch.
2. Connect an RJ45 cable between the Management Port on each USRP and a port on one of the switches.
3. (Optional) Connect a cable between the network switch and external internet for network management.

> **Note** NI recommends that you create a document for tracking USRP serial numbers with assigned network address. Visit the *<REFARCH>*/docs/ folder for a customizable template.

The server contains multiple network interface cards (NIC), each supporting multiple SFP+ ports.

Each USRP has two SFP+ ports, SFP+ 0 and SPF+ 1; however, the reference architecture assumes the use of SFP+ 0.

Figure 21. Overview of Dataflow between the USRPs and the Server



Connect an SFP+ cable between the SFP+ 0 port on each USRP and a port on one of the NICs in the server.

Assign each USRP to a particular network address. NI recommends that you organize your network in the following ways:

- Label the NICs and 10 GbE cables to identify each USRP connection.
- Create a document for tracking USRP serial numbers with assigned network address. Visit the *<REFARCH>*/docs/ folder for a customizable template.

> **Note** For maximum streaming rates with DPDK, both SFP+0 and SFP+1 must be configured and connected. Dual port streaming can also be used without DPDK for enhanced streaming capability.

Plug the USRPs, OctoClocks, server, and switch into the system PDU.

If you want to improve streaming performance for your system, you can install Data Plane Development Kit (DPDK) 20.11. Streaming rates to disk are limited by the write speed of the chosen storage medium. Refer to *Streaming with DPDK Simultaneously to/from Disk Rates* for information on the streaming rates.

> **Note** This reference architecture has been tested using DPDK 20.11 with NVIDIA MCX4121A-ACAT ConnectX-4 Lx EN Adapter Cards. The following steps may only apply to the NVIDIA ConnectX-4.

1. Ensure that you install the proper drivers for your NIC prior to installing DPDK and UHD.
2. Install and configure the DPDK software according to the *Data Plane Development Kit Getting Started Guide for Linux.*
3. Configure huge pages by opening the grub configuration file,
   `/etc/default/grub`.
4. Add the following parameters to `GRUB_CMDLINE_LINUX_DEFAULT`:
   ```
   GRUB_CMDLINE_LINUX_DEFAULT="quiet splash
   default_hugepagesz=2048M hugepagesz=2048M hugepages=10000"
   ```
5. Close `/etc/default/grub`.
6. At the command prompt, update your grub configuration with the following command:
   ```
   sudo update-grub
   ```
7. Reboot the host machine for these settings to take effect.
8. Create a UHD configuration file to interface with DPDK.
   1. Run the following command to list the MAC addresses for your NICs:
      ```
      ip a
      ```
   2. Use the following commands to create the UHD configuration file at the location `/etc/uhd/`:
      ```
      sudo su
      mkdir -p /etc/uhd
      nano /etc/uhd/uhd.conf
      ```
9. Update the following fields of your UHD configuration file.
   - Update the `dpdk_mac` field to match your NIC MAC address variables
   - Update the `dpdk_driver` field if the location is different on your system.
   - Update the `dpdk_corelist` and `dpdk_lcore` fields. In this example, a two-port NIC is used. There should be one core for the main DPDK thread (in this example, core #2), and separate cores assigned to each NIC (in this example, core #3 for the first port on the NIC, core #4 for the second port on the NIC).
   - Update the `dpdk_ipv4` fields to your IP range.

Refer to the following `uhd.conf` file for an example of how you should update the fields in your configuration file.

```
[use_dpdk=1]
dpdk_mtu=9000
dpdk_driver=/usr/local/lib/dpdk-pmds
dpdk_corelist=2,3,4
dpdk_num_mbufs=8192
dpdk_mbuf_cache_size=64

[dpdk_mac=50:6b:4b:3b:3c:d8]
dpdk_lcore = 3
dpdk_ipv4 = 192.168.50.1/24
dpdk_num_desc = 4096

[dpdk_mac=50:6b:4b:3b:3c:d9]
dpdk_lcore = 4
dpdk_ipv4 = 192.168.51.1/24
dpdk_num_desc = 4096
```

Refer to *Running the Examples* to verify successful installation of DPDK on your system.

> **Note** Refer to *Getting Started with DPDK and UHD* for more information about DPDK and UHD.

Complete the following steps to install the software and necessary dependencies on your host machine.

1. Navigate to the GitHub repository, located at https://github.com/EttusResearch/refarch-multich, to get the reference software.
2. Clone the repository to your computer using standard GitHub commands or download the repository directly.
3. In a terminal, navigate to the top-level *refarch-multich* directory and issue the following command to make the setup script executable:
   ```
   chmod +x setup_script.sh
   ```
4. Run the script with the following command:
   ```
   sudo ./setup_script.sh
   ```

The `refarch-multich` repository contains the following folders:

- `config`?Contains script files that assist with the software installation process.
- `docs`?Contains system bill of materials and reference architecture template.
- `examples`?Contains reference architecture examples and configuration files.
- `lib`?Contains the library source files.
- `tools`?Contains Python-based data file generator and viewer.

The following procedure references the *USRP N300/N310/N320/N321 Getting Started Guide*. Follow the https://kb.ettus.com/USRP_N300/N310/N320/N321_Getting_Started_Guide procedure with the specific steps below.

> **Note** When you run the scripts in this section, *all* USRPs connected to the host are updated. Run the `uhd_find_devices` example for a list of USRPs that are connected to the host.

1. Follow the *Connecting to the ARM via SSH* procedure in *Connecting the Device*. You do not need to set up the USRPs for a serial console connection.
2. Follow the procedure in *Updating the Linux File System*.

   > **Note** If the procedure in *Updating the file system with Mender* fails, you must follow the process in *Updating the files system by writing the disk image*. For a script that updates all USRPs connected to the system, refer to `<REFARCH>/config/usrp_filesystem.sh` in the GitHub repository.

3. Follow the procedure in *Updating the Network Configurations*. Set the MTU to `9000`. Every USRP SFP+ ports IP address must have a unique subnet. Change the subnet by replacing *XXX* in the address, for example, `192.168.XXX.2`. If you make any changes to the network configurations, NI recommends that you restart the USRP for those changes to take effect.

   > **Note** For maximum streaming rates with DPDK, dual port streaming must be configured. Follow the procedure in Updating the Network Configurations for configuring both SFP+0 and SFP+1 ports on the USRP. Ensure that the IP addresses are unique.

   > **Note** NI recommends that you create a document for tracking USRP serial numbers with assigned network addresses. Visit the *<REFARCH>/*docs/ folder for a customizable template. For network addresses, NI recommends that you assign static IP addresses for the management port IPs. Refer to the procedure in *Updating the Network Configurations* to update eth0.network file on the USRP.

   For a script to update the SFP ports on all USRPs automatically, refer to *<REFARCH>/*config/usrp_ip_config.sh in the GitHub repository. The script changes the sfp1 port by default but can be modified to change the sfp0 port. You still need to manually configure the host as listed in step 5.

4. Follow the *Network Mode FPGA Image Update* procedure in *Updating the FPGA Image* to update to the XG image. Use the following command to configure both SFP ports to 10 Gb links:
   ```
   uhd_image_loader --args "type=n3xx,addr=<N3xx_IP_ADDR>,fpga=XG"
   ```
   where *<N3xx_IP_ADDR>* is the actual IP address of the USRP being updated. You can find this address by correlating the serial number on the device with the IP address returned by the command `uhd_find_devices`.

   > **Note** For a script to update the FPGA images on all USRPs connected to the system, refer to *<REFARCH>/*/config/usrp_image_loader.sh on the GitHub repository.

5. Follow the *Dual 10Gb Streaming SFP Ports 0/1* procedure in *Setting Up a Streaming Connection* to configure the host Ethernet adapters for 10 GbE connection to the USRP.

   > **Note** Ubuntu Linux uses Netplan to manage IP addresses of network cards, however, other operating systems may configure the network interfaces differently. Visit the *<REFARCH>/config/* folder to find a customizable template `.yaml` file. Each host port and the USRP to which it is paired must have a matching subnet. After the `.yaml` file is configured, copy it to the `/etc/netplan` directory?overwriting `01-network-manager-all.yaml`, if necessary. Then use the following command to apply the network configuration:
   > ```
   > $ sudo netplan apply
   > ```

6. Follow the procedure in *Verifying Device Operation*.
7. (Optional) Follow the procedure in *USRP N3xx Device Specific Operations* to configure USRP settings.

After setting up the first USRP, follow the procedure in *Setting Up the First USRP* for the remaining USRPs with the specific details listed below.

- For subsequent devices, you may choose to execute the following procedures in parallel:

  o *Updating the Linux File System*
  o *Updating the Network Configurations*?Each USRP port must be configured to have a different subnet.
  o Updating the FPGA Image

- Update the `.yaml` file on the host computer after configuring each remaining USRP.

Adjust certain settings to optimize the host system. Most of these settings are completed for you through the setup script. NI recommends that you manually configure the following optimizations:

- Increase the ring buffer for each interface by executing the following command in a terminal

  ```
  sudo ethtool -G <interface> tx 4096 rx 4096
  ```
  where *<interface>* is the network interface of the specific USRP. You can find this interface through the Network Manager or by executing the following command in a terminal `ifconfig`.

  > **Note** Increasing the ring buffer of an interface results in more memory reserved. When setting the ring buffer, make sure to monitor your memory usage. A large number of interfaces can result in hundreds of GB of memory usage. You may need to restart the host system to clear the allocated memory.
  >
  > **Note** This command is executed once for each interface being used. For example, a system with 16 USRPs and 16 interfaces runs 16 times and results in around 65 GB of memory usage. A script has been provided to assist with these settings. The user is required to insert the names of the host interfaces. Refer to `nicrb.sh` in the *<REFARCH>*/config/ folder for more information. This script may need to be run after each host reboot. You can reference the interface names in the `.yaml` file configured in *Setting Up the First USRP Device*.

- Enable hyperthreading on the server to improve multi-threaded performance. Refer to the server BIOS settings for information.

Multichannel RF Reference Architecture is built using the RFNoC API that ships with the USRP Hardware Driver (UHD).

- **UHD RFNoC Example Source Code**? NI recommends that users who are new to UHD or the RFNoC API begin with the following UHD RFNoC examples, located in the `uhd/host/examples` directory:

  o `rfnoc_radio_loopback.cpp`?This example uses the RFNoC API to connect an Rx radio to a Tx radio and run a loopback.
  o `rfnoc_replay_samples_from_file.cpp`?This example uses the replay block to replay data from a file.
  o `rfnoc_rx_to_file.cpp`? This example uses the RFNoC API to demonstrate the radio receiving and writing to file.

- **Reference Architecture Example Source Code**? The following examples, located in the *<REFARCH>*/examples/ folder, demonstrate synchronized Tx-Rx operation scalable from 2x2 up to 32x32 systems:

  o `Arch_iterative_loopback.cpp`?This example cycles through each Tx channel to all Rx channels.
  o `Arch_multifreq_loopback.cpp`?This example cycles through different frequencies and calls `rfnoc_txrx_loopback.cpp`.
  o `Arch_rfnoc_txrx_loopback.cpp`?This example demonstrates a single Tx to all Rx loopback file using a multithreaded implementation.
  o `Arch_rfnoc_txrx_loopback_mem.cpp`?This example demonstrates a single Tx to all Rx loopback to memory using a multithreaded implementation.
  o `Arch_rx_to_mem.cpp`?This –example demonstrates receiving on all channels to memory.
  o `Arch_txrx_fullduplex.cpp`?This example demonstrates simultaneously transmitting to and receiving from the host.
  o `Arch_txrx_fullduplex_dpdk_mem.cpp`?This example demonstrates simultaneously transmitting to and receiving from the host memory using DPDK.
  o `Arch_txrx_fullduplex_dpdk.cpp`?This example demonstrates simultaneously transmitting to and receiving from the host storage using DPDK
  o `Arch_dynamic_tx.cpp`?This example is written for a four-channel system. It demonstrates the ability to stream different files and/or have different logic in each transmit thread. Four (potentially) different transmit threads are spawned, one to each USRP channel. Each thread can be customized as needed. You can apply this example to higher channel count systems, as well as receive threads if needed.

  > **Note** The provided examples use the same LO and Tx frequency transmitted across all channels and devices. You must modify this reference architecture if you want to transmit different frequencies across channels.

The reference architecture examples are compiled when you run the following setup script: `setup_script.sh`.

If you need to make changes to the examples, you must recompile the code.

1. Navigate to *<REFARCH>/build/* to recompile the provided examples.
2. Execute the following commands in the terminal:

   ```
   cmake ..
   make ?j
   ```

If a custom example needs to be compiled, you must modify *<REFARCH>*/CMakeLists.txt to be used with `cmake`. To build a new example titled `new_example.cpp`, complete the following steps.

1. Add the following lines to *<REFARCH>*/CMakeLists.txt:

   ```
   add_executable(new_example {relative File Location}/new_example.cpp)

   message(STATUS "New Example")

   target_link_libraries(new_example PRIVATE UHD_BOOST Arch_lib)
   ```

2. Navigate to *<REFARCH>*/build/.
3. Execute the command `make ?j` in the terminal to compile the custom examples.

The reference architecture examples use configuration files instead of input arguments to configure the application. The configuration files are located in the *<REFARCH>*/examples/ folder.

- Refer to the example `runconfig.cfg` file for a detailed explanation of each argument.
- The reference architecture uses a multiple RAID 0 system to increase the number of parallel writes and, thus, achieve higher data rates. You must configure the `rx-file-location` variable in the `runconfig.cfg` file. Refer to the configuration file for more information.

  > **Note** Ensure that the management address for each USRP is specified in the configuration file as demonstrated in the `runconfig.cfg` example. Not specifying a management address can lead to loss of performance.

Note If dual port streaming is configured, it must be enabled in the Reference Architecture configuration file. Uncomment `second_addr#` for each USRP and replace with the correct USRP number and SFP+1 IP address as configured in *Setting Up the First USRP*.

1. If you are running a DPDK example, you must alter the configuration file. Add `use_dpdk=1` to the following line in the configuration file:
   ```
   args = type=n3xx,master_clock_rate=250e6,use_dpdk=1
   ```
   Note Ensure that you remove `use_dpdk=1` when running non-DPDK examples.
2. Open a terminal in the build folder.
3. Run an example using the `--cfgFile=runconfig.cfg` command. For example, run `rfnoc_txrx_loopback.cpp` by executing the following command in a terminal:
   ```
   ./rfnoc_txrx_loopback --cfgFile=../examples/runconfig.cfg
   ```

- The replay block works as a playback buffer that uses DRAM to store samples on the USRP. Refer to the importData function in `replaycontrol.cpp` in the *<REFARCH>/lib/* folder to observe how data is sent from the host to store on the USRP DRAM.
- Sample data files should contain sc16 data samples and should be a multiple of two samples (8 bytes) in size because the replay block records and plays back in multiples of 8 bytes. Refer to samples_generator.py in the *<REFARCH>/tools/* folder for an example of a sine wave generation. Refer to the *Using the RFNoC Replay Block* Knowledge Base article for more information.

Received data files consist of interleaved int16 IQ data (IQIQI...). You can view data and helpful phase information with the python-based `readDatFile.py` example, located in the *<REFARCH>*/tools/dat_analysis folder. Refer to the comments in `readDatFile.py` for more information about the use of the program.

NI has validated the performance specifications, described in this section, using a 32x32 test system configured as described in *Building the System*. NI used the reference software to determine the specifications in this document.

Warranted?specifications describe the performance of the system under stated operating conditions and are covered by the system warranty. Warranted specifications account for measurement uncertainties, temperature drift, and aging. Warranted specifications are ensured by design or verified during production and calibration.

Characteristics describe values that are relevant to the use of the system under stated operating conditions, but are not covered by the model warranty.

Typical?specifications describe the performance met by a majority of systems.

Nominal?specifications describe an attribute that is based on design, conformance testing, or supplemental testing.

Measured?specifications describe the measured performance of a representative system.

Specifications are?*Measured*?unless otherwise noted.

Transmit channel count	Up to 32
Receive channel count	Up to 32
RF frequency coverage	450 MHz to 6 GHz
RF bandwidth	Up to 200 MHz instantaneous bandwidth per channel
Configuration plane	1 GbE
Data plane	10 GbE
Processing nodes	
On USRP	FPGA
On server	CPU
Data storage and streaming	
Format	Binary
Type	Complex interleaved U16
Acquisition mode	Streaming, finite records

In the context of this reference architecture, a phase-coherent system is interpreted as a system operating at the same frequency with a consistent phase relationship between any two channels in the system. *Phase coherence* is defined as the attribute of two or more waves where the relative phase between waves is constant during the time of the measurement.

Phase coherence performance of the test system is described using the following metrics:

- *Average channel-to-channel phase skew*?The average phase offset between two channels. Described by how it varies over time or from run to run.
- *Standard deviation of channel-to-channel skew*?One number that describes the variation of phase offset between channels around the average skew over time or from run to run.
- *Rx-Rx*?Channel-to-channel specifically referring to Rx channels.

Skew measurements are made based on IQ data files generated using the reference software provided as part of the Multichannel RF Reference Architecture. The measured system uses the MIMO loopback configuration, described in *Connecting the RF RX-TX Ports for a MIMO Loopback*, to route signals from all Tx ports to all Rx ports with a consistent path length. To measure the phase skew, the Angle-FFT method is used, wherein the IQ data generated from each channel is measured against the relevant reference signal for each measurement. The FFT method is used to extract the frequency and phase components of each signal in the vicinity of the signal?s fundamental frequency. This information is used to determine the magnitude of the phase offset between two signals in each measurement period. To compare signals throughout their entire duration, each signal is split into shorter measurement windows in which the instantaneous phase is calculated using the method described above.

Measurements are valid under the following conditions:

- Configured carrier frequency: 2 GHz (S-band/L-band)
- Configured baseband stimulus: 500 kHz sine wave, continuously transmitted for 1 hour, sampled at 6.25 MSample/s at receiver (250 MSample/s clock rate with decimation rate of 40)

- Measurement window: 4 ms (2,500 samples)
- Configuration:
  - Single device: Stimulus simultaneously received at two Rx channels on the same USRP
  - System: Stimulus simultaneously received at two Rx channels on arbitrary separate USRPs
- Analysis:
  - Where μ_n is average skew between two channels in measurement window n
  - $\mu_n - \mu_0$ is the average skew relative to t_0
- Range$\{\mu_0, \mu_1, \ldots, \mu_N\}$ is the range

Note Rx channels on the same USRP internally share the RX LO input to that device as shown in the following figure.

Figure 22. RF Configuration for Stability Measurements

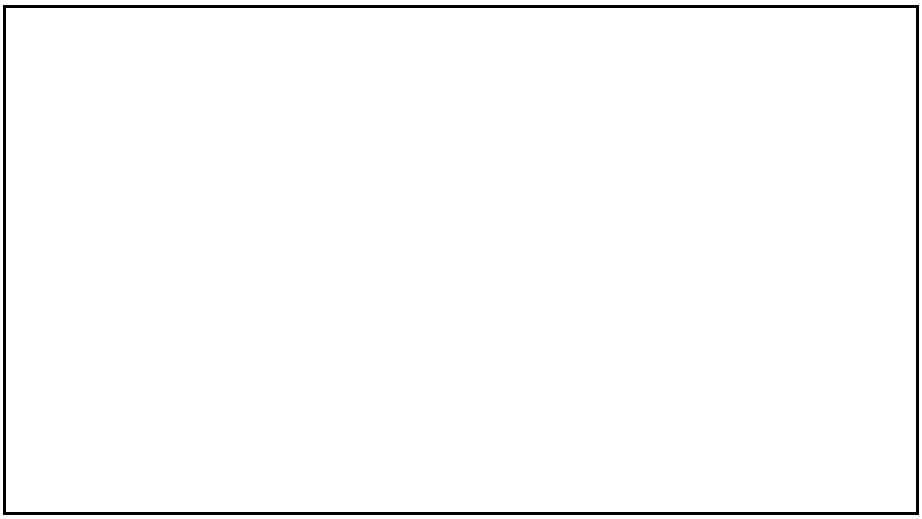
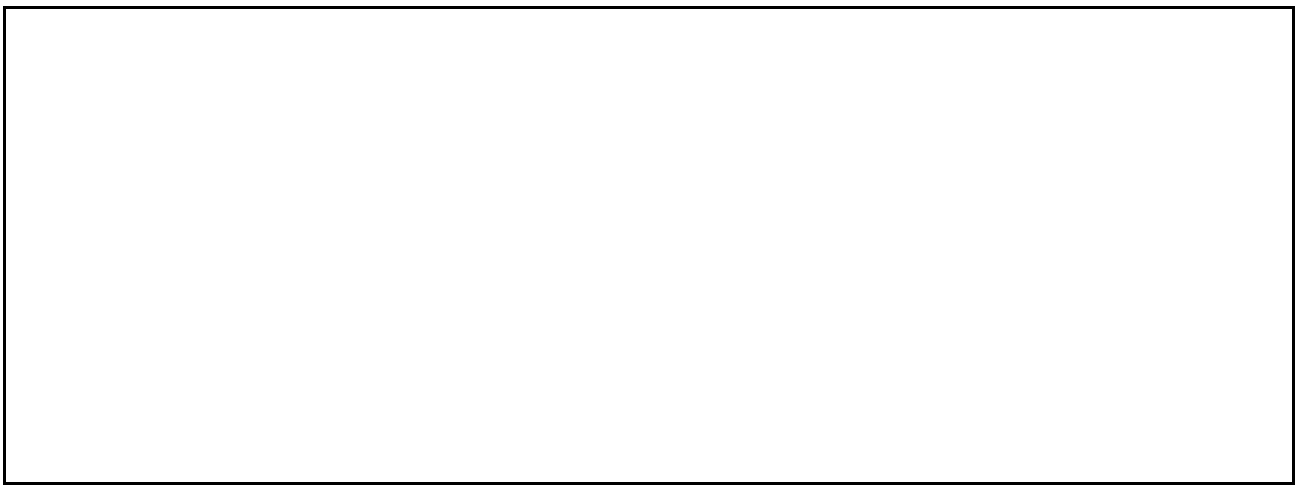
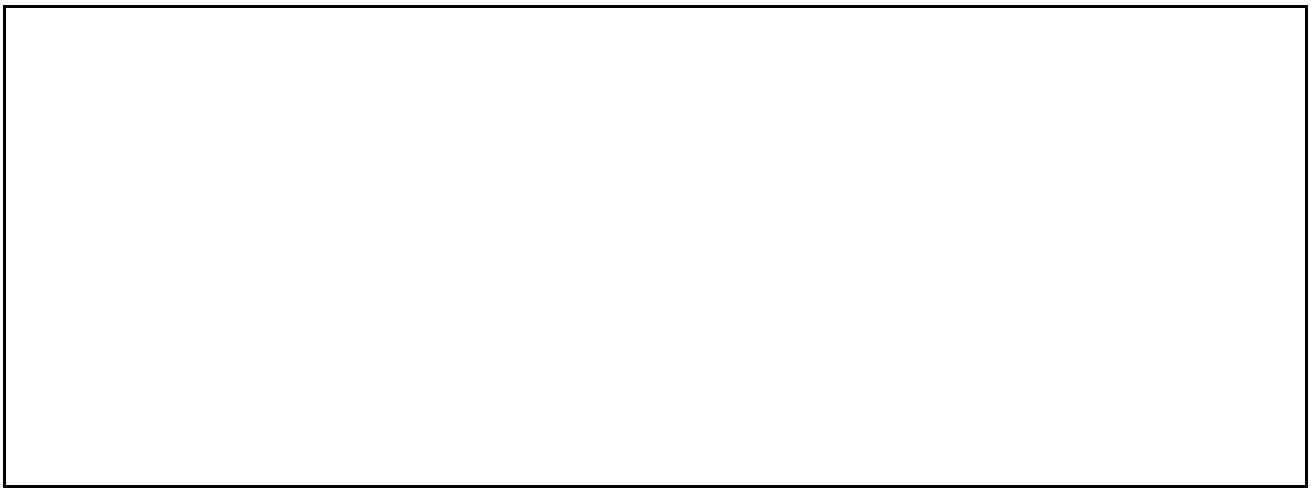


Figure 23. Average Channel-to-Channel Phase Skew between Two Rx Channels on the Same Device over One Hour, Relative to t_0



Range, single device	0.139°

Figure 24. Average Channel-to-Channel Phase Skew between Two Rx Channels on Separate Devices over One Hour, Relative to t_0

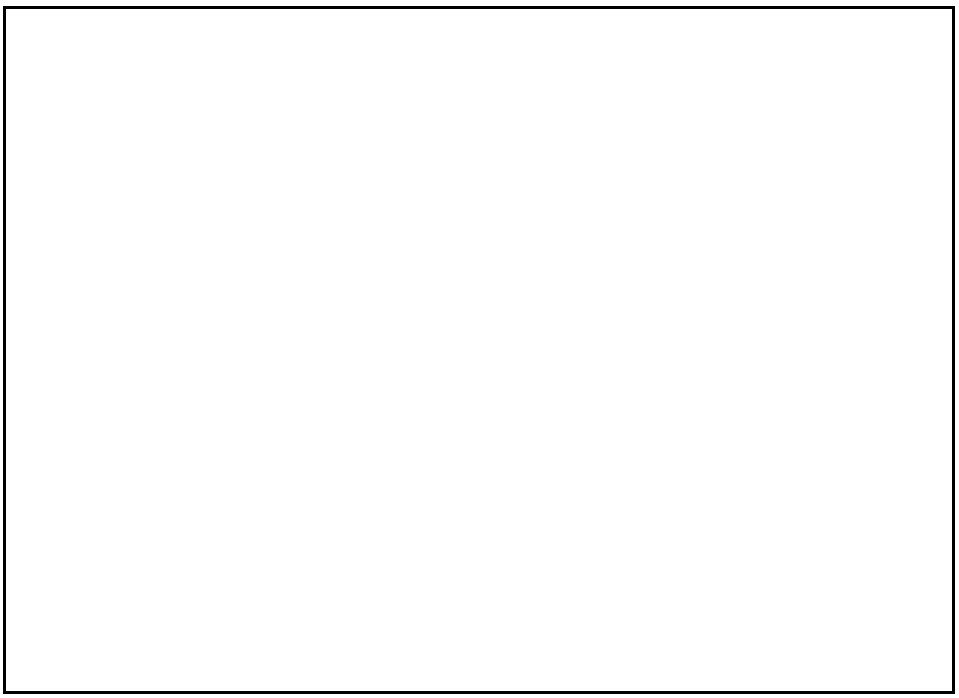


Range, system	0.167°

Measurements are valid under the following conditions:

- Sample data generation: 32 Rx receiving simultaneously at each carrier frequency
- Carrier frequency range: 1.0 GHz to 5.5 GHz, steps of 100 MHz
- Configured baseband stimulus: 500 kHz sine wave, continuously transmitted for 56 µs, sampled at 250 MSample/s
- Measurement window: 8 µs (2,000 samples at 250 MSample/s)
- Analysis:
  - Where μ_{nx} is the average skew between channel *n* and reference channel in measurement window x
  - $sd_n = ?\{\mu_{n0}, \mu_{n1},..., \mu_{n6}\}$ is the standard deviation of average skew for channel n using all measurement windows
  - Average over 32 channels is $\mu\{sd_1...\}$
  - Maximum of 32 channels is $\max\{sd_1...\}$

Figure 25. Standard Deviation in Short-Term Rx-Rx Phase Skew as a Function of Carrier Frequency



Measurements are valid under the following conditions:

- 10 runs
- Transmit from one Tx to all 32 Rx, repeat 32 times, once with each Tx
- Configured carrier frequency: 2 GHz
- Configured baseband stimulus: 56 µs, sampled at 250 MSample/s
- Measurement window: 8 µs (2,000: samples)
- Analysis:
- Where m_{nx} is Rx n-to-Rx 0 average skew in run *x*
  - $sd_n = ?\{m_{n0}, m_{n1},..., m_{n9}\}$ is the standard deviation from run to run of Rx-Rx skew for channel *n*
  - $r_n = \text{Max}(\text{Abs}(\{m_{n0}, m_{n1},..., m_{n9}\} ? m_{n0}))$ is the range from run to run of Rx-Rx skew for channel *n*
  - $\mu\{sd_1, sd_2,...,sd_{32}\}$ is the system average standard deviation of Rx-Rx skew
  - $\text{Maximum}\{sd_1, sd_2,...,sd_{32}\}$ is the system maximum standard deviation of Rx-Rx skew

o $\mu\{r_1, r_2,...,r_{32}\}$ is the system average range of Rx-Rx skew
o Maximum$\{r_1, r_2,...,r_{32}\}$ is the system maximum range of Rx-Rx skew

Run-to-run, standard deviation of channel-to-channel skew (sd_n)	
Average over 32 channels	0.021°
Maximum of 32 channels	0.145°
Range of average channel-to-channel skew over 10 runs (r_n)	
Average of 32 channels	0.070°
Maximum of 32 channels	0.483°

Note In each run analyzed here, the carrier frequency is 2.0 GHz. Users must recharacterize channel-to-channel skew when using different carrier frequencies, because average channel-to-channel skew in the system deviates by >>1° per channel.

Run-to-run, standard deviation of channel-to-channel skew (sd_n)	
Average over 32 channels	0.025°
Maximum of 32 channels	0.066°
Range of average channel-to-channel skew over 10 runs (r_n)	
Average of 32 channels	0.067°
Maximum of 32 channels	0.169°

Measurements are valid under the following conditions:

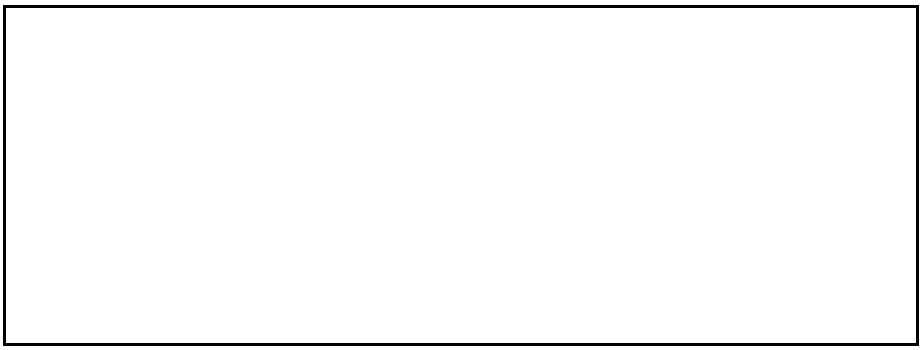
- Configured carrier frequency: 1 GHz
- Configured baseband stimulus: 500 kHz sine wave, continuously transmitted at 250 MSample/s for 1 hour, sampled at 3.2 GSample/s in finite windows
- Measurement window: 35 ms (120,000 samples)
- Configuration:

  o Transmitters: Stimulus simultaneously transmitted from any two channels
  o Receivers: Stimulus simultaneously received at two Rx channels on external acquisition device

- Analysis:

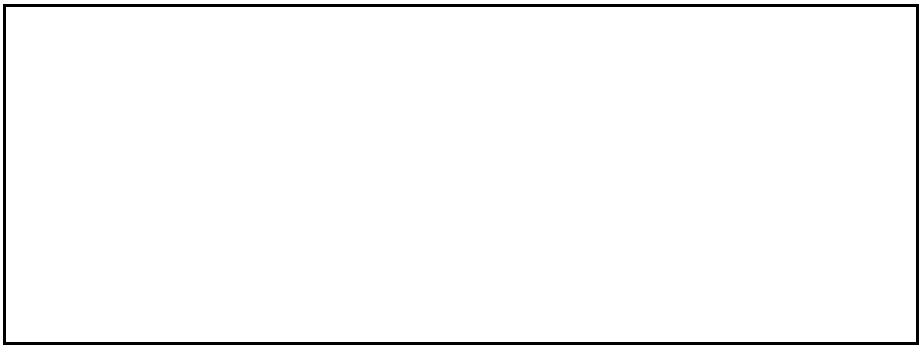
  o Where μ_n is average skew between two channels in measurement window *n*
  o μ_n - μ_0 is the average skew relative to t_0
  o Range$\{\mu_0, \mu_1, ... , \mu_N\}$ is the range

Figure 26. Average Channel-to-Channel Phase Skew between Two Tx Channels on the Same Device over One Hour, Relative to t_0



Range, single device	0.128°

Figure 27. Average Channel-to-Channel Phase Skew between Two Tx Channels on Separate Devices over One Hour, Relative to t_0



Range, system	0.115°

Measurements are valid under the following conditions:

- 50 runs
- Transmit from 16 Tx
- Configured carrier frequency: 1 GHz
- Configured baseband stimulus: 35 µs, sampled at 3.4 GSample/s
- Measurement window: 35 µs (120,000: samples)
- Analysis:
  - Where m_{nx} is Tx n-to-Tx 0 average skew in run x
  - sd_n = ?{m_{n0}, m_{n1},..., m_{n9}} is the standard deviation from run to run of Tx-Tx skew for channel n
  - r_n = Max(Abs({m_{n0}, m_{n1},..., m_{n9}} - m_{n0})) is the range from run to run of Tx-Tx skew for channel n
  - µ{sd_1, sd_2,...,sd_{32}} is the system average standard deviation of Tx-Tx skew
  - Maximum{ sd_1, sd_2,...,sd_{32}} is the system maximum standard deviation of Tx-Tx skew
  - µ{r_1, r_2,...,r_{32}} is the system average range of Tx-Tx skew
  - Maximum{ r_1, r_2,...,r_{32}} is the system maximum range of Tx-Tx skew

Run-to-run, standard deviation of channel-to-channel skew (sd_n)	
Average over 16 channels	0.009°
Maximum of 16 channels	0.019°
Range of average channel-to-channel skew over 50 runs (r_n)	
Average of 16 channels	0.034°
Maximum of 16 channels	0.070°

Run-to-run, standard deviation of channel-to-channel skew (sd_n)	
Average over 16 channels	0.008°
Maximum of 16 channels	0.018°
Range of average channel-to-channel skew over 50 runs (r_n)	
Average of 16 channels	0.033°
Maximum of 16 channels	0.070°

Measurements are valid under the following conditions:

- Configured frequency: 2.5 GHz (S-band)
- Configured stimulus: Single-tone
- Number of channels: Varied
- Duration: Single run, 10 s

Table 5. Streaming to Memory Data Transfer Rate

Streaming Rate (MSample/s per Channel)	Number of Channels	Measurement Time (s)
33.33	32	10
50	16	10
122.88	8	10

Note For more information about supported sampling and master clock rates, refer to the *USRP N300/N310/N320/N321 Getting Started Guide*.

Measurements are valid under the following conditions:

- Configured frequency: 2.5 GHz (S-band)
- Configured stimulus: Single-tone
- Number of channels: Varied

Table 6. Streaming to Disk Data Transfer Rate

Streaming Rate (MSample/s per Channel)	Number of Channels	Measurement Time (s)
33.33	32	10
50	16	10
62.5	8	10

Measurements are valid under the following conditions:

- Configured frequency: 2.5 GHz (S-band)

- Configured stimulus: Single-tone
- Number of channels: Varied

Table 7. Streaming Simultaneously to/from Disk Rate

Streaming Rate (MSample/s per Channel)	Number of Channels	Measurement Time (s)
11.11	32	10
25	16	10
62.5	8	10
62.5	4	10

Measurements are valid under the following conditions:

- Configured frequency: 2.5 GHz (S-band)
- Configured stimulus: Single-tone
- Number of channels: Varied

Table 8. Streaming Simultaneously from Host to Host Memory Rate

Streaming Rate (MSample/s per Channel)	Number of Channels	Measurement Time (s)
250	8	7200

Table 9. Streaming Simultaneously from Host to Host Disk Rate

Streaming Rate (MSample/s per Channel)	Number of Channels	Measurement Time (s)
125	8	10

NI tested and configured the Multichannel RF Reference Architecture for a 32x32 system. It is possible to build a 64x64 (or greater) system.

When the system channel count increases to 64 channels and greater, consider the following:

- The MIMO loopback, 10 MHz and PPS, and LO distribution connections require additional tiers.
- Drift increases for typical measurements.

Refer to *Connecting the LO Distribution* for information about LO distribution connections and best practices.

> **Note** This configuration is used to synchronize both the Rx and Tx to the same LO resulting in the same center frequencies for both Tx and Rx. To synchronize different frequencies, connect RX LO OUT to RX LO IN and TX LO OUT to TX LO OUT. Refer to *USRP N320/N321 LO Distribution* for connection diagrams.

The following table lists the tiers for the 64x64 system configuration.

Table 8. 64x64 LO Sources and Tiers

System	Tiers	USRP N321 Minimum Quantity	USRP N320 Maximum Quantity
64x64	LO Source (0), 1, 2, 3	11	21

The following diagram represents LO distribution signal connections.

For the 64x64 system, all RX LO OUT cables originating from the LO Source USRP N321 must be the same length. All RX LO OUT and TX LO OUT cables originating from the Tier 1 USRP N321s must be the same length. All RX LO OUT and TX LO IN cables originating from the Tier 2 USRP N321s must be the same length. Note that you cannot combine LO signals.

Figure 28. LO Sharing (64x64)

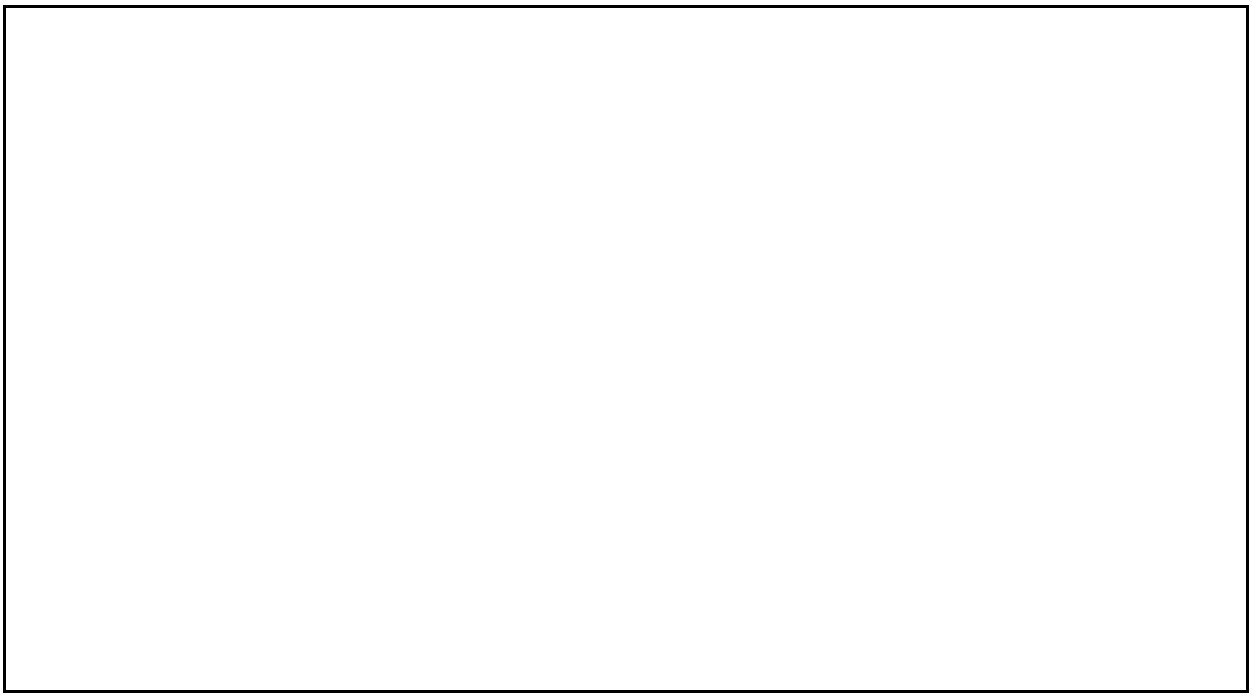


Table 9. LO Sharing Cabling (64x64)

Cable Color in Diagram	Cable Tier	Number of Cables in Tier
	LO Source Tier 0	4
	Distribution Tier 1, RX LO OUT	8
	Distribution Tier 1, TX LO OUT	8
	Distribution Tier 2, RX LO OUT	32
	Distribution Tier 2, TX LO OUT	32

Refer to the following documents to learn more about the Multichannel RF Reference Architecture, software, and hardware components:

- GitHub repository?Contains the system bill of materials and reference software example and library files needed to set up, install, and run your Multichannel RF Reference Architecture system. Visit https://github.com/EttusResearch/refarch-multich.
- *USRP N300/N310/N320/N321 Getting Started Guide*?Describes how to install software, how to set up your USRP, and confirm that your device is operating properly. Visit https://kb.ettus.com/USRP_N300/N310/N320/N321_Getting_Started_Guide.
- *N320/N321* specifications?Lists specifications for the USRP N321 and USRP N320. Visit https://kb.ettus.com/N320/N321.
- *USRP Hardware Driver and USRP Manual?*Describes how to use the USRP N321 and USRP N320 and how to use the API to connect to devices through your own software. Visit https://files.ettus.com/manual/index.html.
- *USRP N320/N321 LO Distribution*?Describes LO distribution connections and commands. Visit https://kb.ettus.com/USRP_N320/N321_LO_Distribution.
- *Using the RFNoC Replay Block*?Describes the basic use of the RFNoC Replay block in UHD and how to run the UHD Replay example. Visit https://kb.ettus.com/Using_the_RFNoC_Replay_Block.
- *Data Plane Development Kit Getting Started Guide for Linux*?Describes how to install and configure DPDK software in a Linux environment. Visit https://doc.dpdk.org/guides-20.11/linux_gsg/intro.html.

- Using Open-Source Tools with USRP Hardware for SDR Applications?Visit learn.ni.com, select **View Library**, and search for `using open-source tools with USRP`.
- Open-Source SDR Knowledge Base?Visit https://kb.ettus.com/Knowledge_Base.
- SDR-Academy Informal Getting Started Videos?Visit the SDR Academy at www.ettus.com/support/sdr-academy/.
- RFNoC Workshop from GRCon 2020?Visit https://kb.ettus.com/Training_Workshops to find RFNoC video links and workshop materials.

1. ? The *USRP N320/N321 LO Distribution* Knowledge Base article does not show one LO signal shared across Tx and Rx.
2. ? *Stability* reports variation in the value of average channel-to-channel phase skew over a long (>>1 s) duration.
3. ? *Short-term* describes variation in channel-to-channel phase skew over a short duration (<100 µs).
4. ? A *USRP session* is a period in which communication links between chips in the USRP are established and not changed. Session reset is accomplished by restarting the reference software for each run.
5. ? A *USRP configuration* is a period in which the digital path within the USRP is configured and not changed. Configuration reset is accomplished by changing carrier frequency from 2.0 GHz to 2.1 GHz and back to 2.0 GHz between runs.
6. ? Measurements were taken with a two-port direct sampling analyzer. The analyzer port 0 was directly connected to USRP 0 Channel 0. Port 1 of the analyzer was connected to the output of a 16-channel RF multiplexer. All other USRP channels were connected to the input of the 16-channel RF multiplexer. USRP 0 Channel 0 was compared to each other channel sequentially.
7. ? *Streaming to memory* is measured as the sustained data transfer rate without any overflows.
8. ? *Streaming to disk* is measured as the sustained data transfer rate without any overflows.
9. ? *Streaming simultaneously to/from disks* is defined as streaming from the host to the radio for transmission, while simultaneously receiving on the radio to a file on the host. The rates are measured as sustained data transfer rates without any overflows.
10. ? *Streaming Simultaneously with DPDK to/from Disk Rates* Streaming simultaneously with DPDK to/from disk is defined as streaming from the host to the radio for transmission, while simultaneously receiving on the radio to a file on the host. The rates are measured as sustained data transfer rates without any overflows using the Trenton Systems 3U BAM.