Open Architecture For Radar and EW Research

1. REDIRECT Multichannel RF Reference Architecture

Contents

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 Open Architecture for Radar and EW Research. A PDF version of this AppNote is available for those who need or prefer that document style.

This user manual describes how to design a system that enables synchronization between Receive (Rx) 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. The purpose of this document is to provide a starting point in the creation of your specific system configuration.

The Open Architecture for Radar and EW Research 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 L-band, S-band, and C-band radar applications, enabling coherent operation through shared local oscillator (LO) , PPS, and 10 MHz reference clock signals.

Figure 1. System Overview
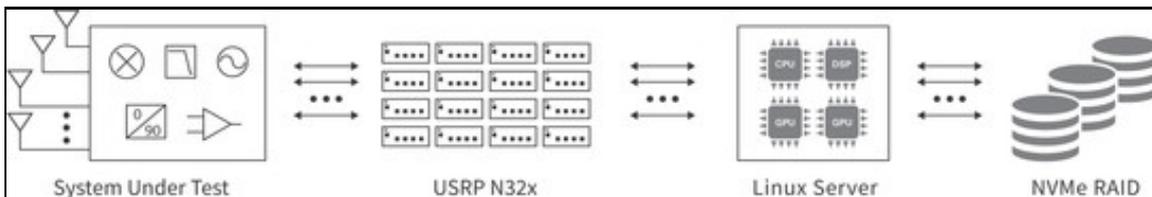
The architecture features the following components:

- User manual, bill of materials, and interconnect diagrams that describe how to set up a system with up to 32 channels of Rx-Rx and Tx-Rx 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, best practices, and known issues encountered during development and validation of an 32x32 system. *nxn* refers to systems where *n* is the number of transmit and 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 and 32 Tx-Rx channels.
  o Streaming IQ data from 32 channels to a server for further processing.

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

The reference architecture demonstrates waveform generation from file using RFNoC replay blocks.

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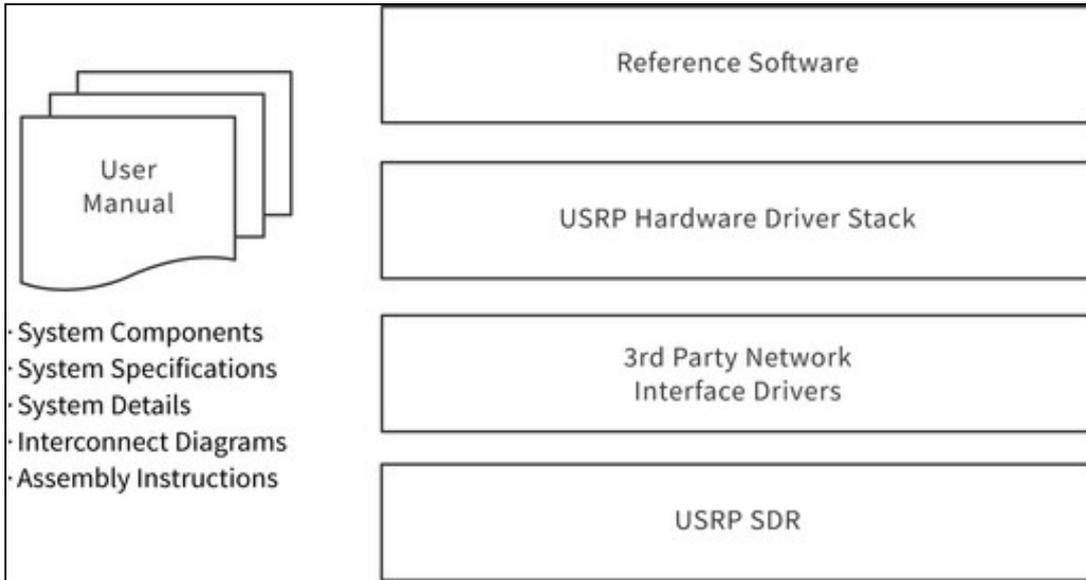
The reference architecture uses USRP N321s and USRP N320s that scale from 2 to 32 phase-coherent transmit and receive channels.

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

Figure 3. Hardware Overview
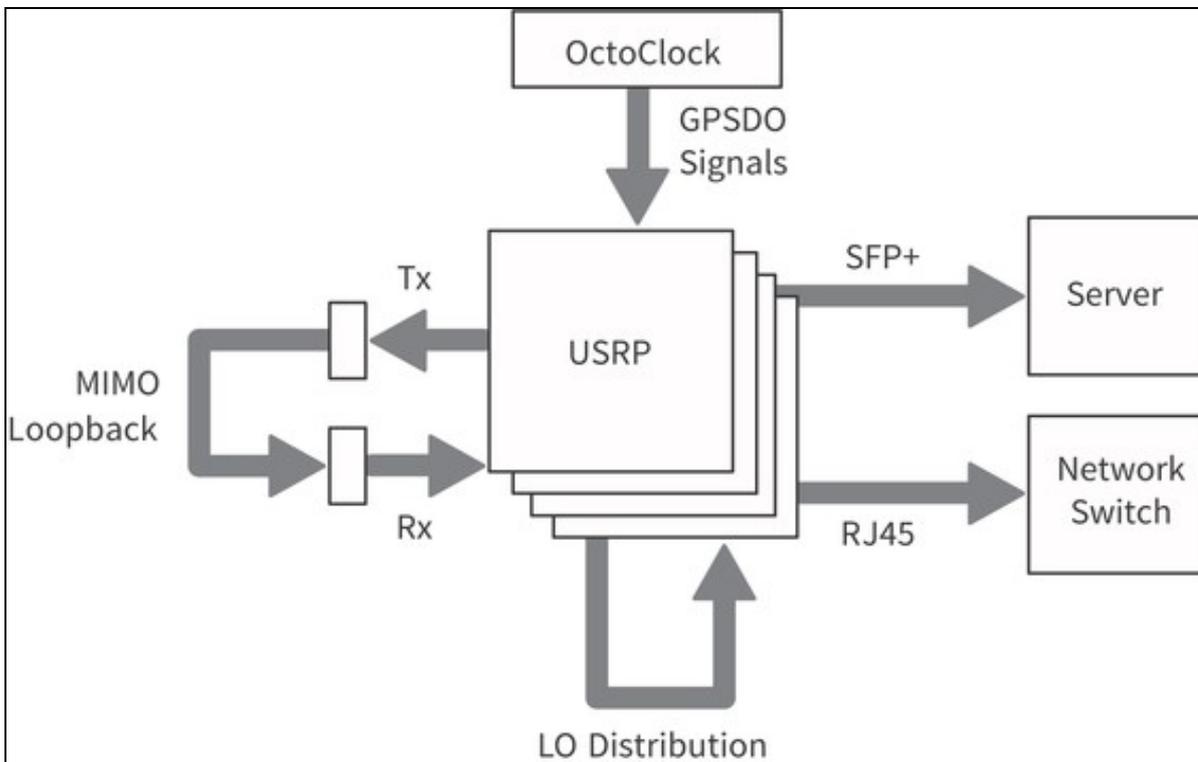
For a complete list of required hardware for the Open Architecture for Radar and EW Research, refer to the bill of materials in the *<REFARCH>/docs/* folder, where *<REFARCH>* is an alias for the following GitHub file folder location: https://github.com/EttusResearch/refarch-multich

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 10 MHz reference clock and PPS signals and features a management port.
- **NI RF Torque Screwdriver and SMA Driver Bit** (NI part number 780895-01) ? Torque screwdriver 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).
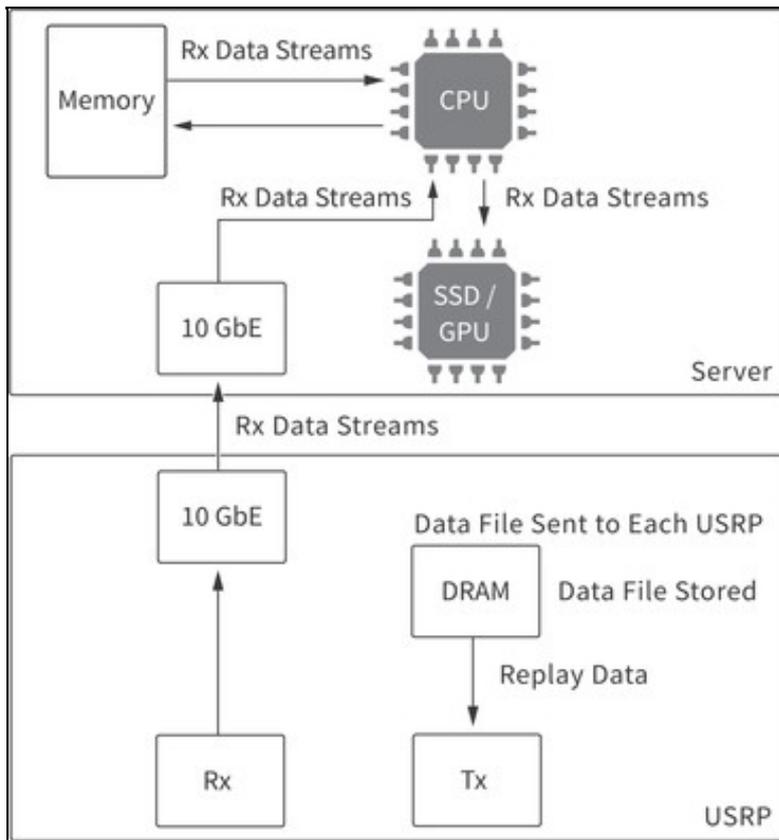
Testing and deploying a system requires hardware from both NI and third-party vendors. The following component options have been validated and are recommended for most Open Architecture for Radar and EW Research systems:

- 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 BAM3U** ? Server with Intel dual 3rd generation Xeon Ice Lake processors and 11 Gen 4.0 PCIe slots (5?x16, 6?x8). 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.

- 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

  o **Tripp Lite Rackmount Rack Extension Bracket** (B018-000-1P5) ? Bracket required to mount the USRPs in a rack.
  o **Z-Bracket** ? Bracket required to mount the USRPs in a rack to meet environmental specifications. Contact NI for more information about how to procure Z-brackets.

The reference software, shown in Figure 2, consists of a reference library and examples, written in C++, using USRP Hardware Driver (UHD) version 4.1. 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 so it is scalable for systems beyond the validated 32x32 system.

The software demonstrates a Tx-Rx loopback across a system of USRPs. To maximize Rx bandwidth, the replay block is used to reduce host processing and network traffic for Tx. Data moves through the system as shown in the following figure.

Figure 4. Data Movement

Rx Data Streams

Memory

CPU

Rx Data Streams

Rx Data Streams

SSD / GPU

10 GbE

Server

Rx Data Streams

10 GbE

Data File Sent to Each USRP

DRAM   Data File Stored

Replay Data

Rx

Tx

USRP

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

Figure 5. Rack Layout for 32x32 System

#	Left	Middle	Right	#
32		Basic PDU		32
31		Network Switch		31
30				30
29		Server		29
28				28
27				27
26		KVM		26
25			USRP N321 1	25
24	USRP N320 0			24
23			USRP N321 3	23
22	USRP N320 2			22
21		OctoClock CDA-2990 Unit 1		21
20	Shelf for Cables Unit 0 / 8-way Splitter Unit 0	8-way Splitter Unit 1	8-way Splitter Unit 2 / 8-way Splitter Unit 3	20
19			USRP N321 5	19
18	USRP N320 4			18
17			USRP N321 7	17
16	USRP N320 6			16
15		OctoClock-G CDA-2990 Unit 0		15
14	Shelf for Cables Unit 1 / 4-way Splitter Unit 0		4-way Splitter Unit 1	14
13			USRP N321 9	13
12	USRP N320 8			12
11			USRP N320 11	11
10	USRP N320 10			10
9		OctoClock CDA-2990 Unit 2		9
8	Shelf for Cables Unit 2 / 8-way Splitter Unit 4	8-way Splitter Unit 5	8-way Splitter Unit 6 / 8-way Splitter Unit 7	8
7			USRP N321 13	7
6	USRP N320 12			6
5			USRP N320 15	5
4	USRP N320 14			4
3				3
2				2
1		Switched PDU		1

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

- While running the system in loopback, attach 30dB 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. Contact NI for more information about how to procure the Z-brackets.

- 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.

- Number and label the USRPs with serial numbers. Visit the *<REFARCH>/docs/32_Channel_Template_layout_BOM_and_wiring.ods* for a customizable template.

- 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 signal path of all Rx, Tx, LO, PPS, and 10 MHz signals identical, 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 system connections.

> **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).

Warning: 30dB attenuators must be used on every TX channel to ensure safe use of the MIMO Loopback. Using the system without attenuators will overload the RX and will permanently damage the USRPs

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 the splitter to the USRP RX2 connectors (used for receiving signals).

> **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 combiner and an 8-way splitter. All cables connecting between tiers, shown in the following figure, must be the same length and type.

Figure 6. MIMO Loopback (8x8)
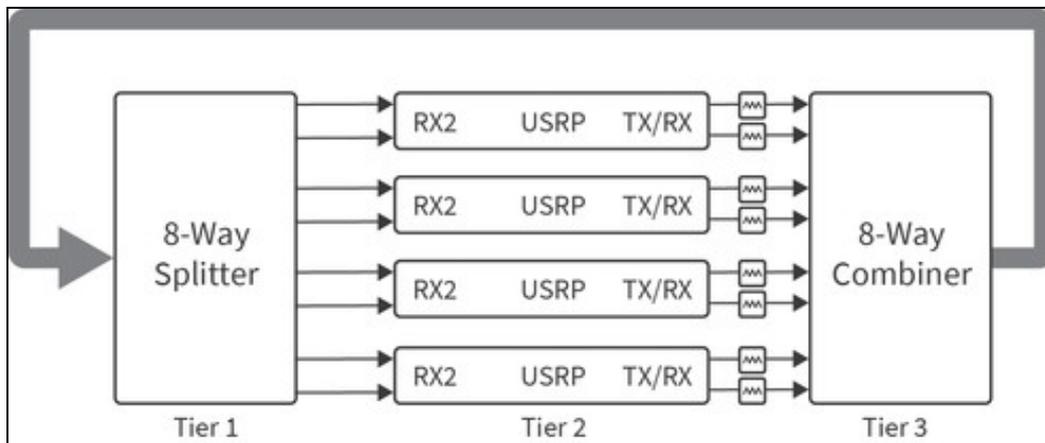
The 16x16 system requires two 8-way splitters, one 2-way splitter, two 8-way combiners, and one 2-way combiner. To set up the MIMO loopback, connect the RF 0/1 TX/RX and RX2 connectors on the USRP to an 8-way combiner and an 8-way splitter. Then connect the two 8-way combiners to a 2-way combiner and 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 7. MIMO Loopback (16x16)
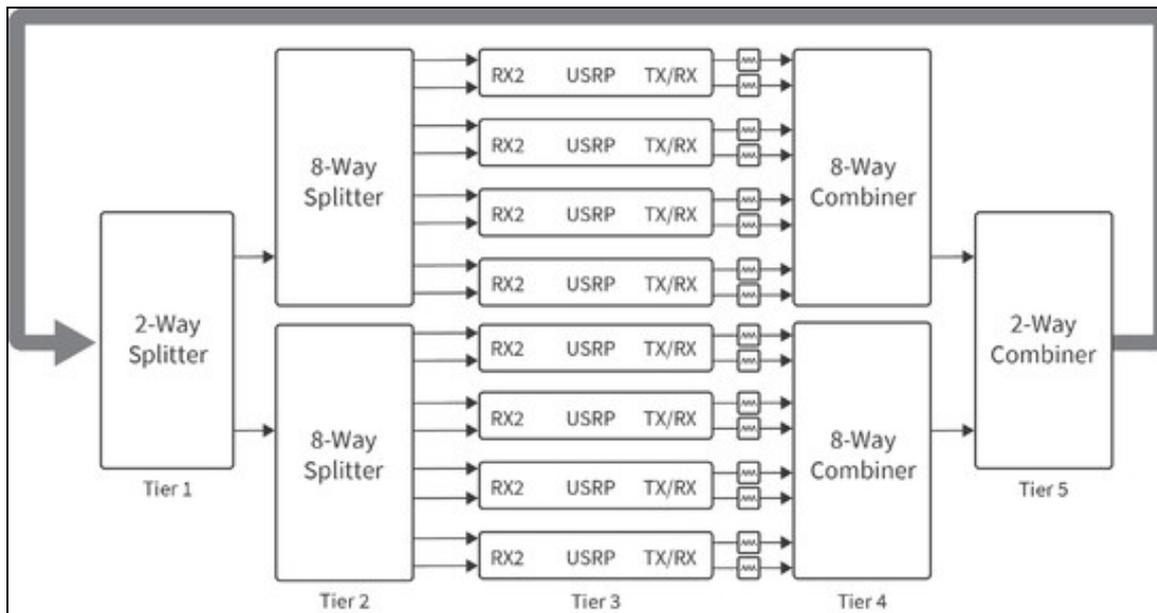
The 32x32 system requires four 8-way splitters, one 4-way splitter, four 8-way combiners, and one 4-way combiner. To set up the MIMO loopback, connect the RF 0/1 TX/RX and RX2 connectors on the USRP to an 8-way combiner and an 8-way splitter. Then connect the four 8-way combiners to a 4-way combiner 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.
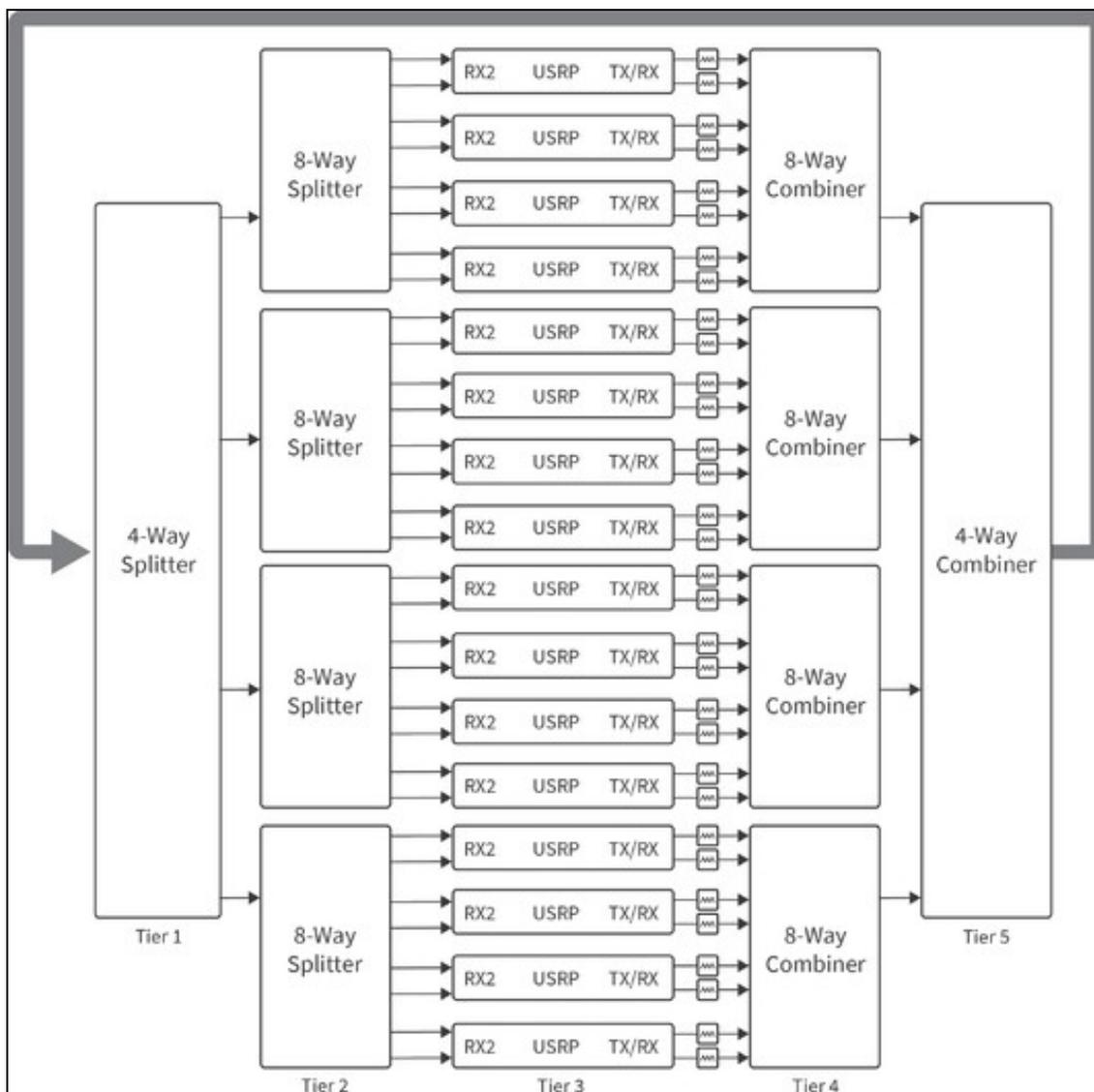
Figure 8. MIMO Loopback (32x32)

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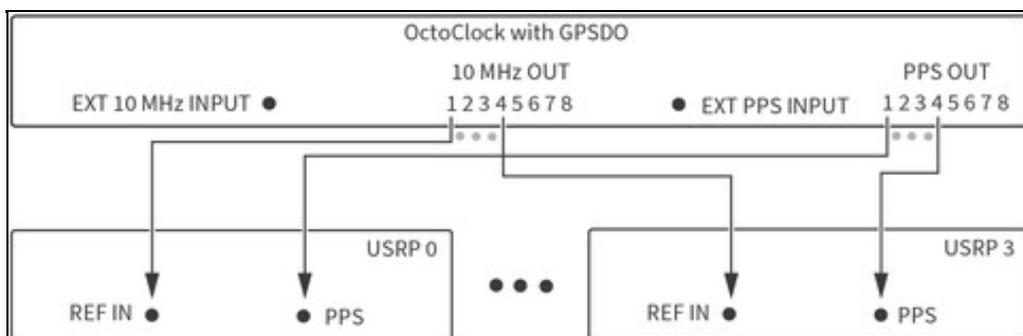
The 4x4 system requires one OctoClock with GPSDO, two USRPs, and uses two 10 MHz and two PPS OUT SMA connectors. Ensure that the source selection switch is set to internal on OctoClock 0 prior to power up.

Figure 9. 10 MHz and PPS Connections (4x4)
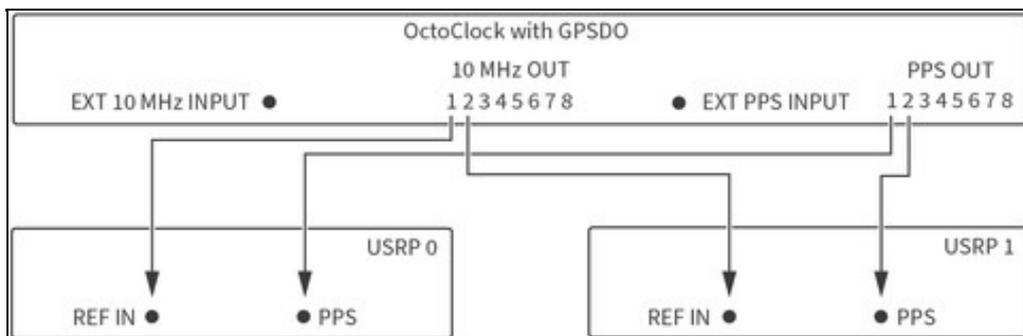
The 8x8 system requires one OctoClock with GPSDO, four USRPs, and uses four 10 MHz and four PPS OUT SMA connectors. Ensure that the source selection switch is set to internal on OctoClock 0 prior to power up.

Figure 10. 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 11. 10 MHz and PPS Connections (16x16)
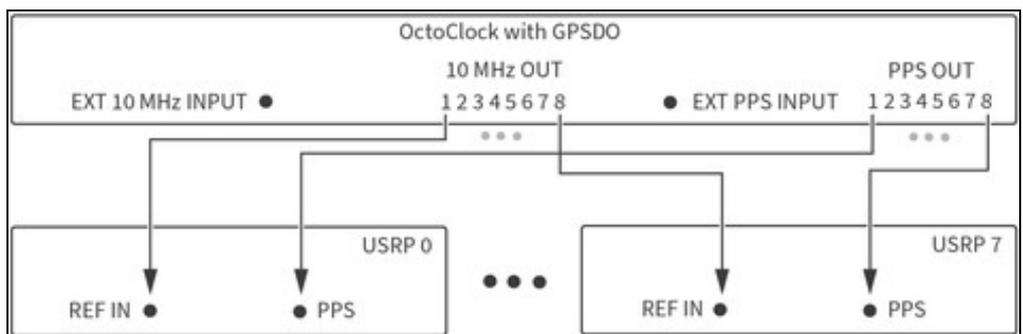
Scaling to a 32x32 system requires one OctoClock with GPSDO (OctoClock 0, shown in the following figure), two additional OctoClocks (OctoClocks 1 and 2), and 16 USRPs. The source 10 MHz and PPS signals originate from OctoClock 0, which drives the OctoClock 1 and 2 inputs. OctoClock 1 and 2 each distribute 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 12. 10 MHz and PPS Connections (32x32)
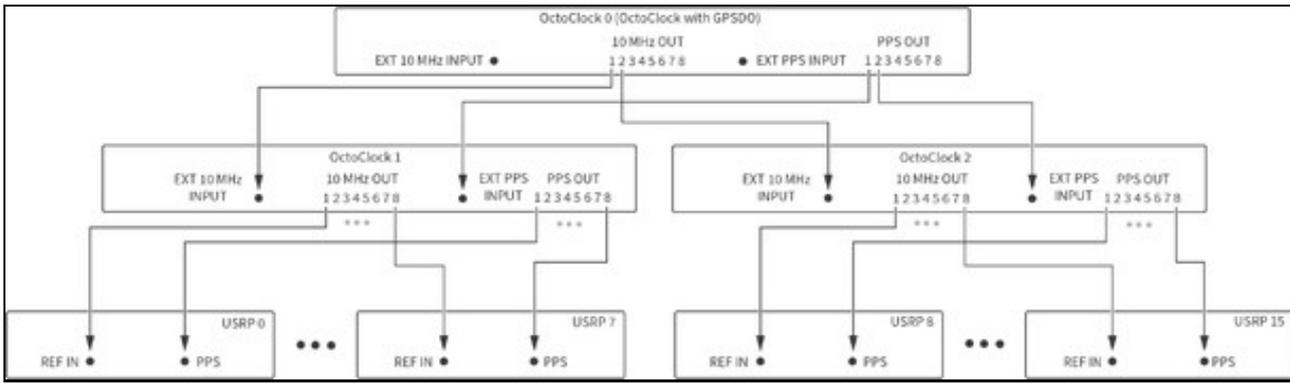
Connect the TX LO OUT, RX LO IN, and TX LO IN connectors on the USRPs for LO distribution. While connecting the LO distribution, adhere to the following:

- 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 minimal identical cable length between tiers.

The following table lists the tiers for all system configurations.

Table 1. LO Sources 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 13. LO Sharing (2x2)
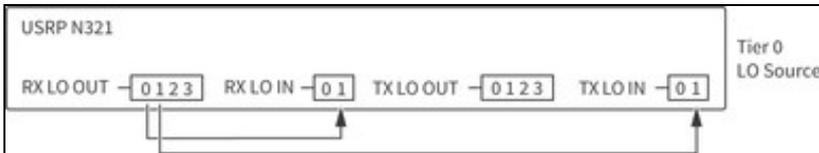
The 4x4 system requires four cables of identical length. Note that you cannot combine LO signals. NI recommends that all LO distribution cables originating from the same tier are the same length.

Figure 14. LO Sharing (4x4)
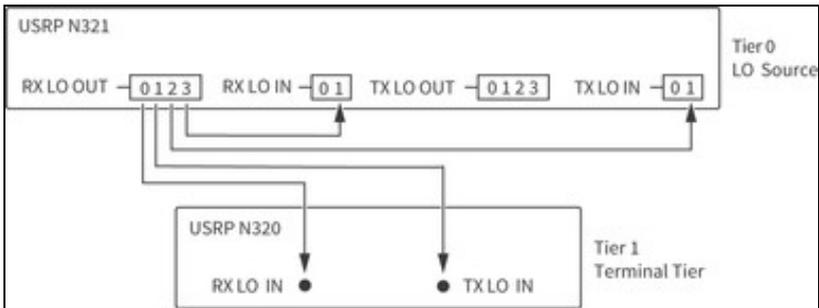
For the 8x8 system, both RX LO OUT cables originating from the LO Source USRP N321 must be the same length. All RX LO OUT and TX LO OUT cables originating from the Tier 1 USRP N321 must be the same length. Note that you cannot combine LO signals.

Figure 15. 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 16. LO Sharing (16x16)
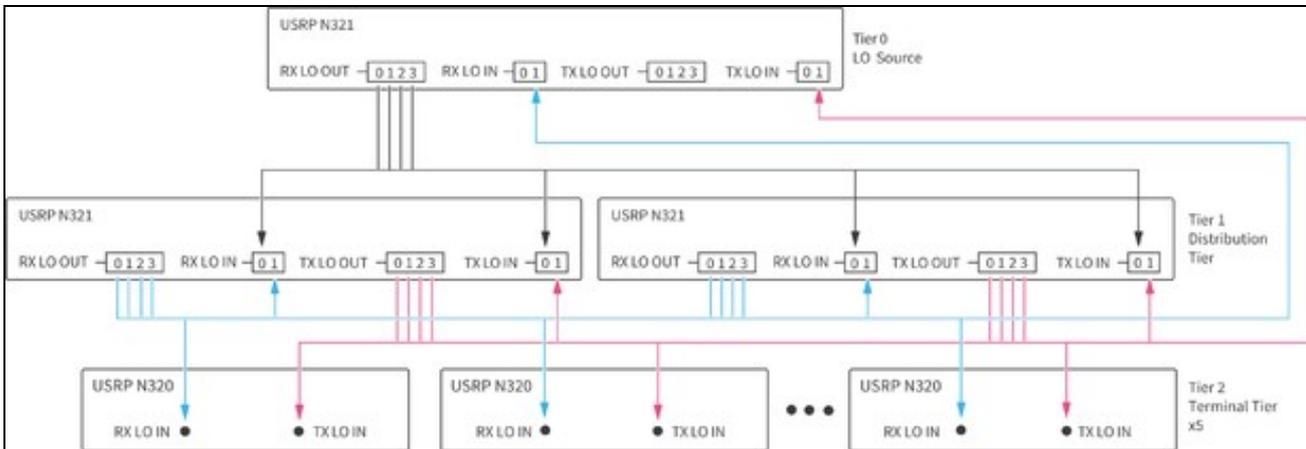
Table 3. LO Sharing Cabling (16x16)

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.
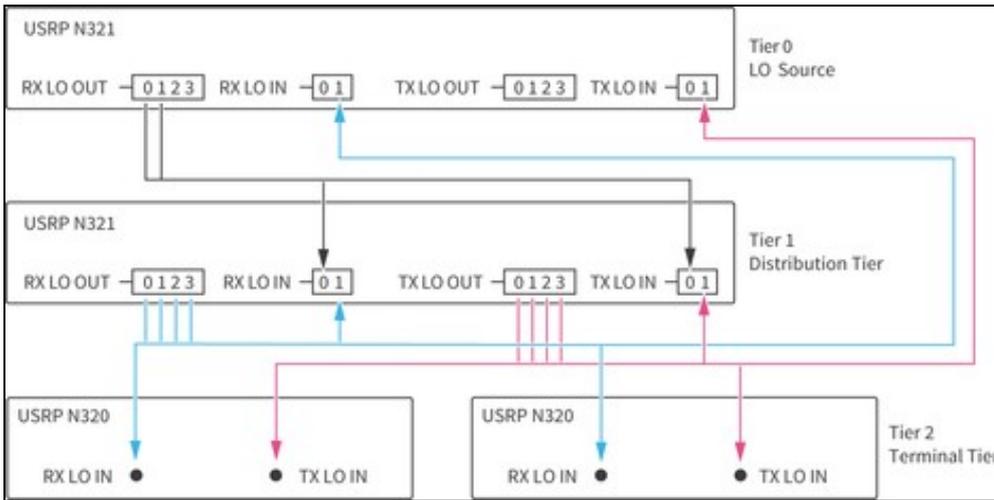
Figure 17. LO Sharing (32x32)
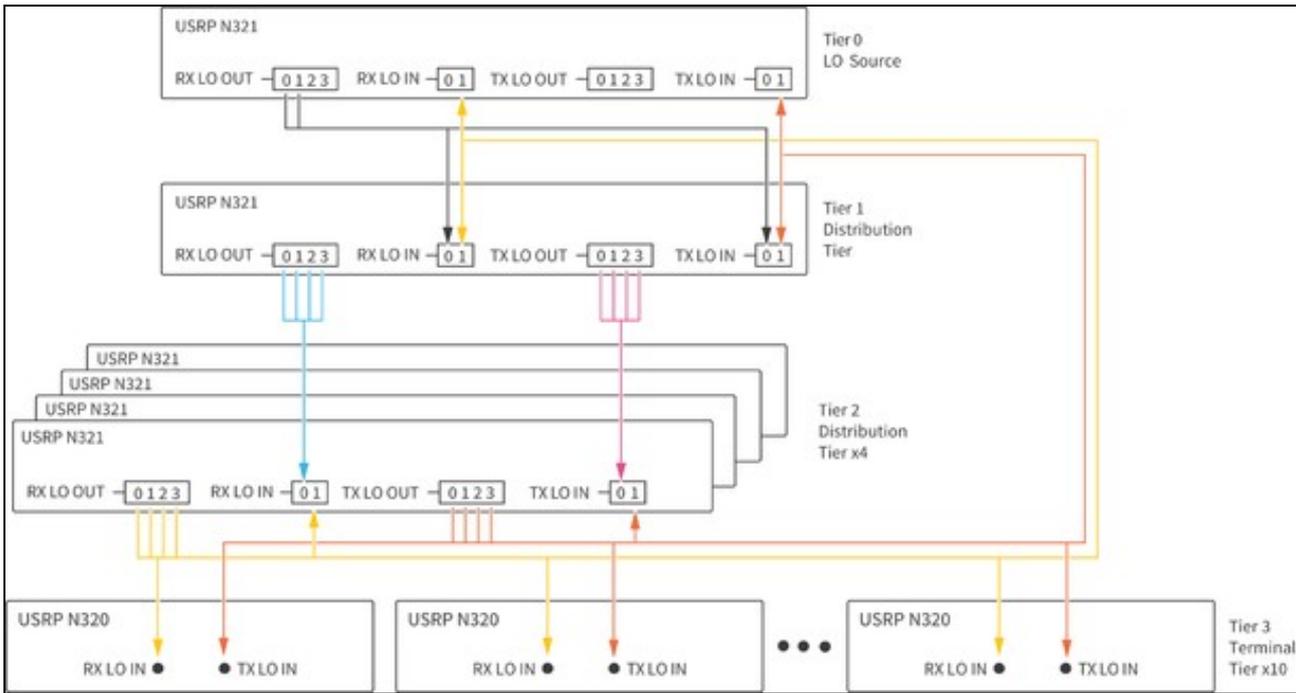
Table 4. LO Sharing Cabling (32x32)

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 configure and update the file system on the USRPs. 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 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 18. Overview of Dataflow between the USRPs and the Server
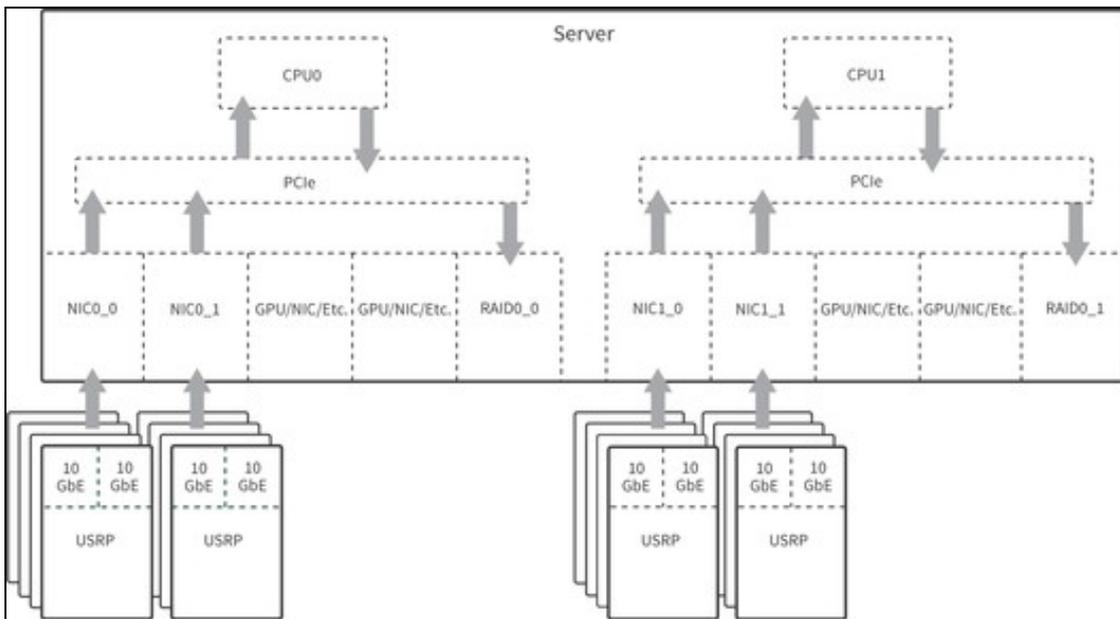
Connect an SFP+ cable between the SFP+ 0 port on each USRP and a port on one of the NICs in the server.

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.

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

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 example 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* at https://kb.ettus.com/USRP_N300/N310/N320/N321_Getting_Started_Guide. Follow the procedure with the specific steps listed in this document.

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 will update all USRPs connected to the system, see refarch-multich-dev/config/usrp_filesystem.sh on the github repository.
3. Follow the procedure in *Updating the Network Configurations*. Set the MTU to **9000**. If you make any changes to the network configurations, NI recommends that you restart the USRP for those changes to take effect.
   Note NI strongly recommends that you create a document for tracking USRP serial numbers with assigned network addresses. Visit the *<REFARCH>/docs/* folder for a customizable template. NI recommends that you assign static IP addresses for the management port IPs. This procedure is not included in this documentation.
   For a script to update the sfp1 port on all USRPs automatically, see refarch-multich-dev/config/usrp_ip_config.sh on the github repository. The host will still need to be configured manually as shown 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, see refarch-multich-dev/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 (other operating systems may configure the network interfaces differently). Visit the *<REFARCH>/config/* folder to find a customizable template .yaml file. The ports on the host require an IP address that matches the USRP it connects to. Once 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 complete procedure in *Setting Up the First USRP* for the remaining USRPs with the specific details listed here.

- 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:

> **Note** Increasing an interface's ring buffer results in more memory reserved. When setting the ring buffer make sure to monitor your memory usage. High number of interfaces can result in hundreds of GB of memory usage. A restart might be needed to clear the memory allocation.

- Increase the ring buffer for each interface by calling the following command:

```
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 calling the command `ifconfig`.

> **Note** This command is executed once for each interface being used. For example, a system with 16 USRPs with 16 interfaces runs 16 times and results in ~65GB of memory usage. A script has been provided that can 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*.

- Enable hyperthreading on the server. Refer to server BIOS settings for information.
- Enable C states on the server. Refer to server BIOS settings for information.

Open Architecture for Radar and EW Research is built using the RFNoC API that ships with the USRP Hardware Driver (UHD).

- **UHD RFNoC Examples** ? NI recommends that users who are new to USRP Hardware Driver or the RFNoC API begin with the following UHD RFNoC examples, located in the UHD installation folder, `uhd/host/examples`:

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

  o `iterative_loopback_singlethread.cpp` ? This example cycles through each Tx channel to all Rx channels.
  o `multifreq_loopback.cpp` ? This example cycles through different frequencies and calls `rfnoc_txrx_loopback.cpp`.
  o `rfnoc_txrx_loopback.cpp` ? This example demonstrates a single Tx to all Rx loopback file using a multithreaded implementation.
  o `rfnoc_txrx_loopback_mem.cpp` ? This example demonstrates a single Tx to all Rx loopback to memory using a multithreaded implementation.

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` configuration 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. This requires user configuration to `receivefunctions.cpp` in the *<REFARCH>/lib/* folder before building the examples. Refer to `receivefunctions.cpp` 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.

1. Open a terminal in the build folder.
2. Run an example using the `--cfgFile=runconfig.cfg` command. For example, run `rfnoc_txrx_loopback.cpp` by calling
   `./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 2 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.
- Use the replay example to generate a waveform. Refer to the *Using the RFNoC Replay Block* Knowledge Base article at https://kb.ettus.com/Using_the_RFNoC_Replay_Block for more information.

Received data files consist of interleaved int16 IQ data (`IQIQI...`). Refer to the `readDatFile.py` example in the *<REFARCH>/tools/* folder for more information about viewing data files.

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	3 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 Open Architecture for Radar and EW Research. 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:

    o Single device: Stimulus simultaneously received at two Rx channels on the same USRP
    o System: Stimulus simultaneously received at two Rx channels on arbitrary separate USRPs

- 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

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

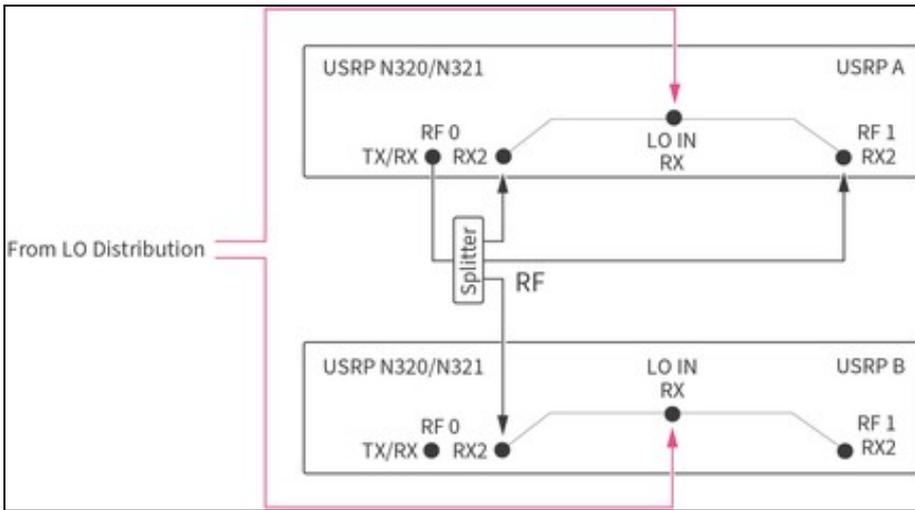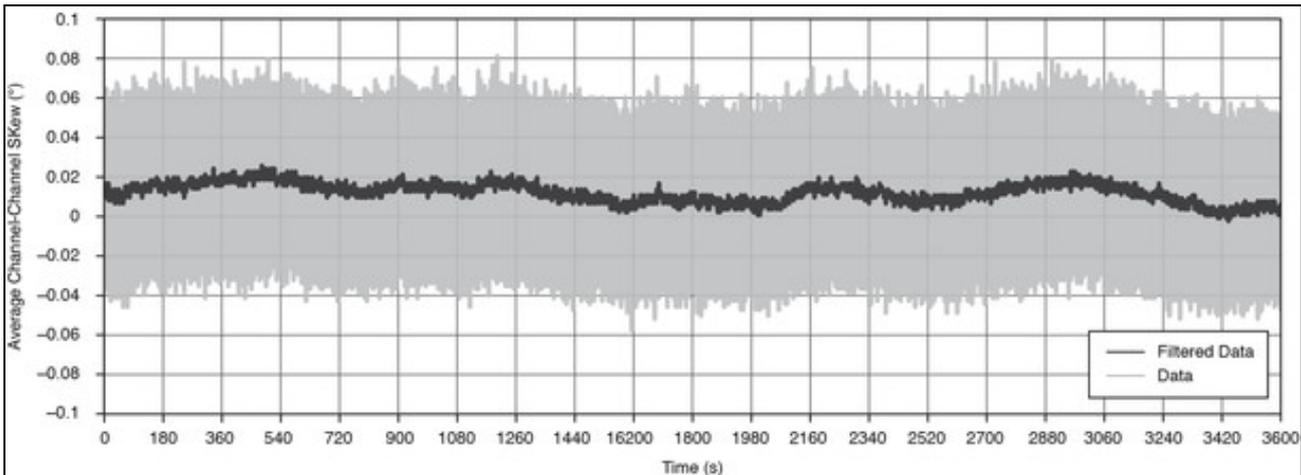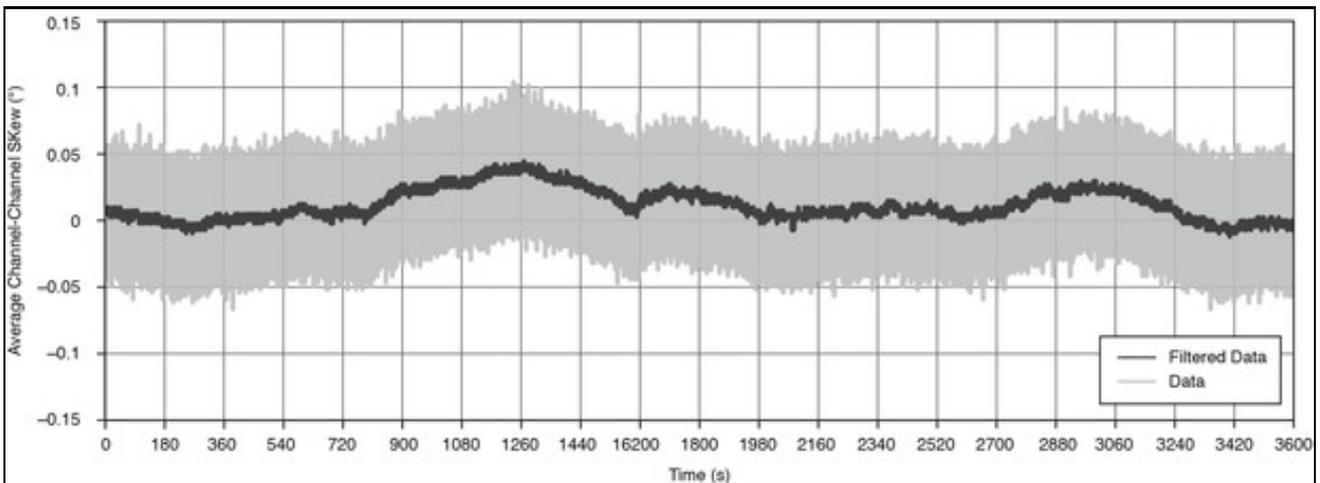Figure 19. RF Configuration for Stability Measurements

Figure 20. 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 21. 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:

  o Where μ_{nx} is the average skew between channel n and reference channel in measurement window x
  o $sd_n = ?\{\mu_{n0}, \mu_{n1}, ?, \mu_{n6}\}$ is the standard deviation of average skew for channel n using all measurement windows
  o Average over 32 channels is $\mu\{sd_1?\}$
  o Maximum of 32 channels is $\max\{sd_1?\}$

Figure 22. Standard Deviation in Short-Term Rx-Rx Phase Skew as a Function of Carrier Frequency
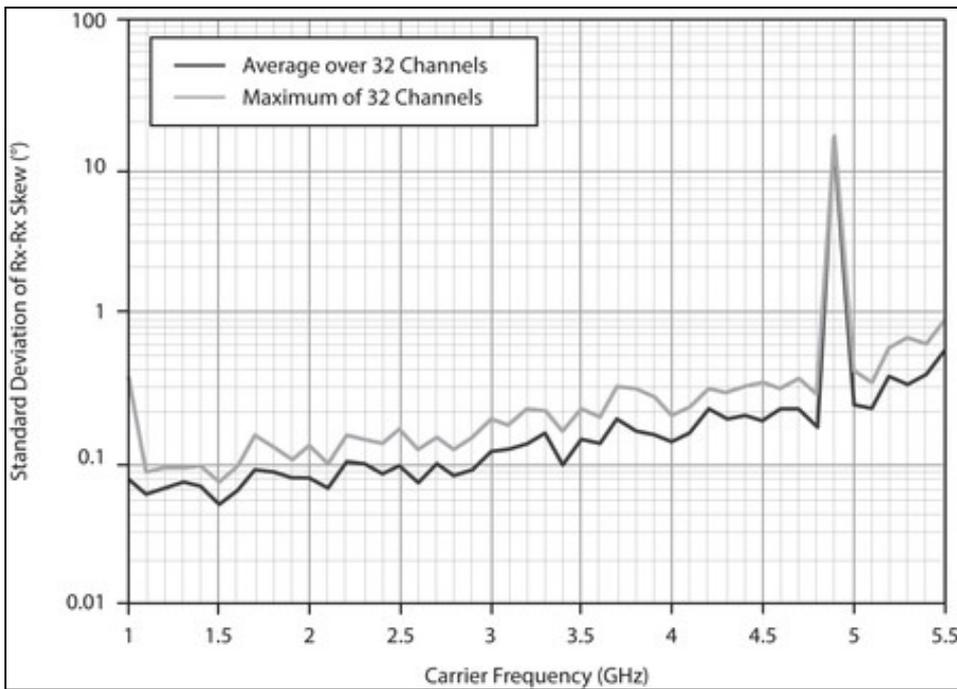
Measurements are valid under the following conditions:

- 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:

  o Where m_{nx} is Rx n-to-Rx 0 average skew in run x
  o $sd_n = ?\{m_{n0}, m_{n1}, ? m_{n9}\}$ is the standard deviation from run to run of Rx-Rx skew for channel n
  o $r_n = Max(Abs(\{m_{n0}, m_{n1}, ? m_{n9}\} - m_{n0}))$ is the range from run to run of Rx-Rx skew for channel n
  o $\mu\{sd_1, sd_2, ?, sd_{32}\}$ is the system average standard deviation of Rx-Rx skew
  o $Maximum\{sd_1, sd_2, ?, sd_{32}\}$ 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. This was tested using a 32x32 MIMO loopback and sweeping carrier frequency from 1.0 GHz to 5.5 GHz.

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 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* at https://kb.ettus.com/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
- Duration: Single run, 10 s

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

NI tested and configured the Open Architecture for Radar and EW Research 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.

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

Table 7. 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 OUT cables originating from the Tier 2 USRP N321s must be the same length. Note that you cannot combine LO signals.
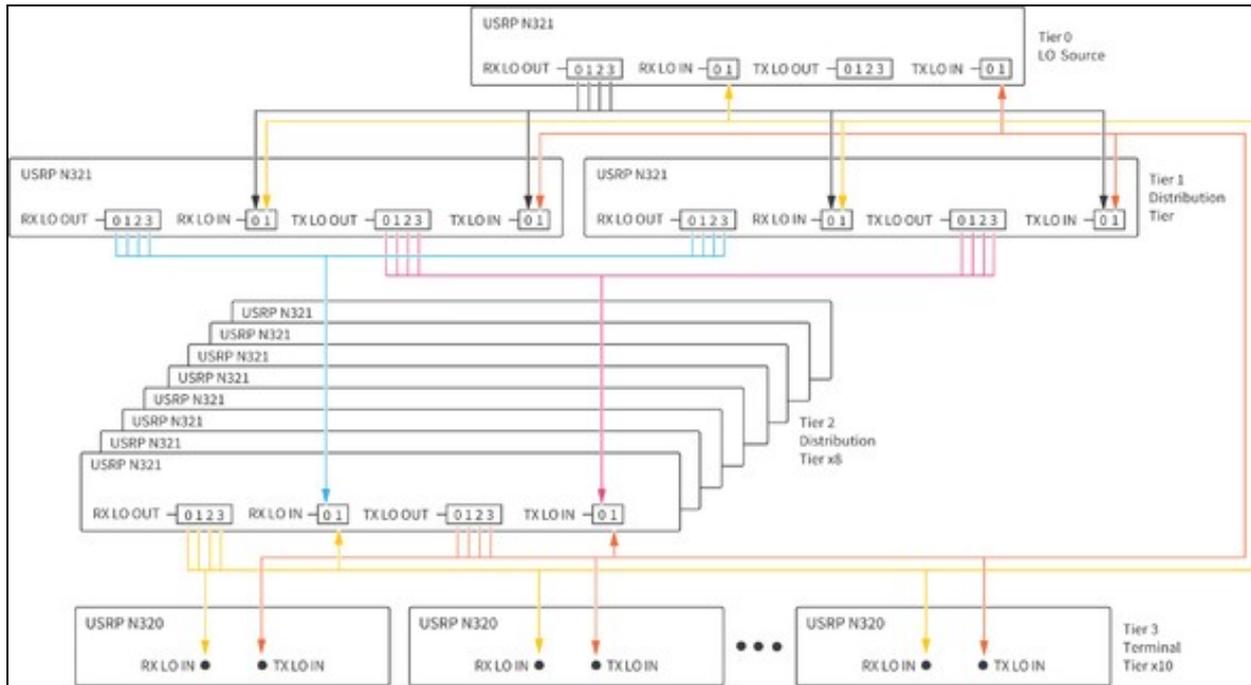
Figure 23. LO Sharing (64x64)

Table 8. 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 Open Architecture for Radar and EW Research, 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 Open Architecture for Radar and EW Research 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 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 LO Distribution Knowledge Base article at https://kb.ettus.com/USRP_N320/N321_LO_Distribution 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. ? *Streaming to memory* is measured as the sustained data transfer rate without any overflows.
7. ? *Streaming to disk* is measured as the sustained data transfer rate without any overflows.