VANTEON®

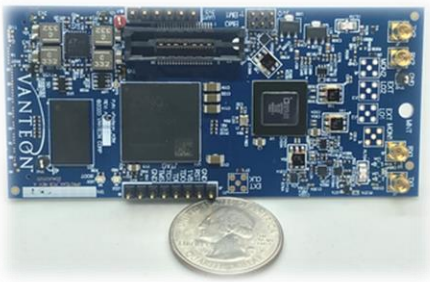
WIRELESS SOLUTIONS

Application Note

Multichip Synchronization of ADRV9002 RF channels on the Vanteon vProtean® SDR

Overview

Multichannel Radar and other Multiple Input Multiple Output (MIMO) applications require phase synchronization between the RF channels. For the ADRV9002 from Analog Devices, this channel synchronization is achieved using the Multi-Chip Synchronization process (MCS). This application note describes how to implement MCS using Vanteon's vProtean SDR.



The Vanteon vProtean® SDR is built around the Analog Devices ADRV9002 RF Transceiver. The ADRV9002 is the industry's first high performance highly integrated transceiver IC that operates from 30 MHz to 6 GHz, capable of handling narrow band and/or wideband signals from 12 kHz to 40 MHz. The ADRV9002 is a highly integrated RF transceiver that has dual-channel transmitters, dual-channel receivers, integrated synthesizers, and digital signal processing functions. This application note discusses use of the ADRV9002 Multichip Synchronization features on a Vanteon vProtean SDR.

Accelerate Your Design

Let Vanteon's engineering team help you get your products to market faster.

Call us at 888.506.5677 or email sales@vanteon.com

About Vanteon

Vanteon is a design services company offering complete electronic systems design and engineering solutions with a focus in Wireless and RF design. We offer Analog, Digital, and RF Hardware design, as well as Software and FPGA design services. To learn how Vanteon's expertise can be enlisted to help solve your most pressing challenges, visit the Vanteon website at https://vanteon.com.

Best Companies to work for NEW YORK 2022

Vanteon Corp. | 888.506.5677 | 585.419.9555 | 99 Garnsey Road, Suite 200, Pittsford, NY 14534 | www.vanteon.com

Application Details

Whether synchronizing 2 channels on a single ADRV9002 transceiver or a design using multiple ADRV9002 transceivers, the MCS process is required. This Application Note will first discuss the MCS processing on a single vProtean followed by a review of the MCS process for multiple vProtean radios. The vProtean is a Software Defined Radio (SDR) platform based on the ADRV9002 developed at Vanteon.

The ADRV9002 supports two MCS modes, MCSMODE_ENABLE and MCSMODE_ENABLED_WITH_RFPLL_PHASE.

The MCS mode MCSMODE_ENABLE guarantees the delay between the RF ports and the Synchronous Serial Interface (SSI) (as well as the reverse) are deterministic across all channels.

If MCS mode MCSMODE_ENABLED_WITH_RFPLL_PHASE is selected, then phase synchronization for the PLLs across multiple devices is also provided.

This application note discusses the use of the MCS mode MCSMODE_ENABLED_WITH_RFPLL_PHASE.

The MCS pulse timing and alignment with the reference clock are well described in the ADRV9001 Reference Manual titled “ADRV9001 System Development User Guide for the RF Agile Transceiver Family” from Analog Devices. This Application Note assumes some familiarity with the reference manual.

MCS on a single vProtean

Figure 1 Shows the vPortean reference design. The ADRV FPGA IP block is available as part of the ADRV9002 SDK from Analog devices and is used to interface with the ADRV9002. The vProtean uses LVDS for the Synchronous Serial Interface (SSI) data interface.

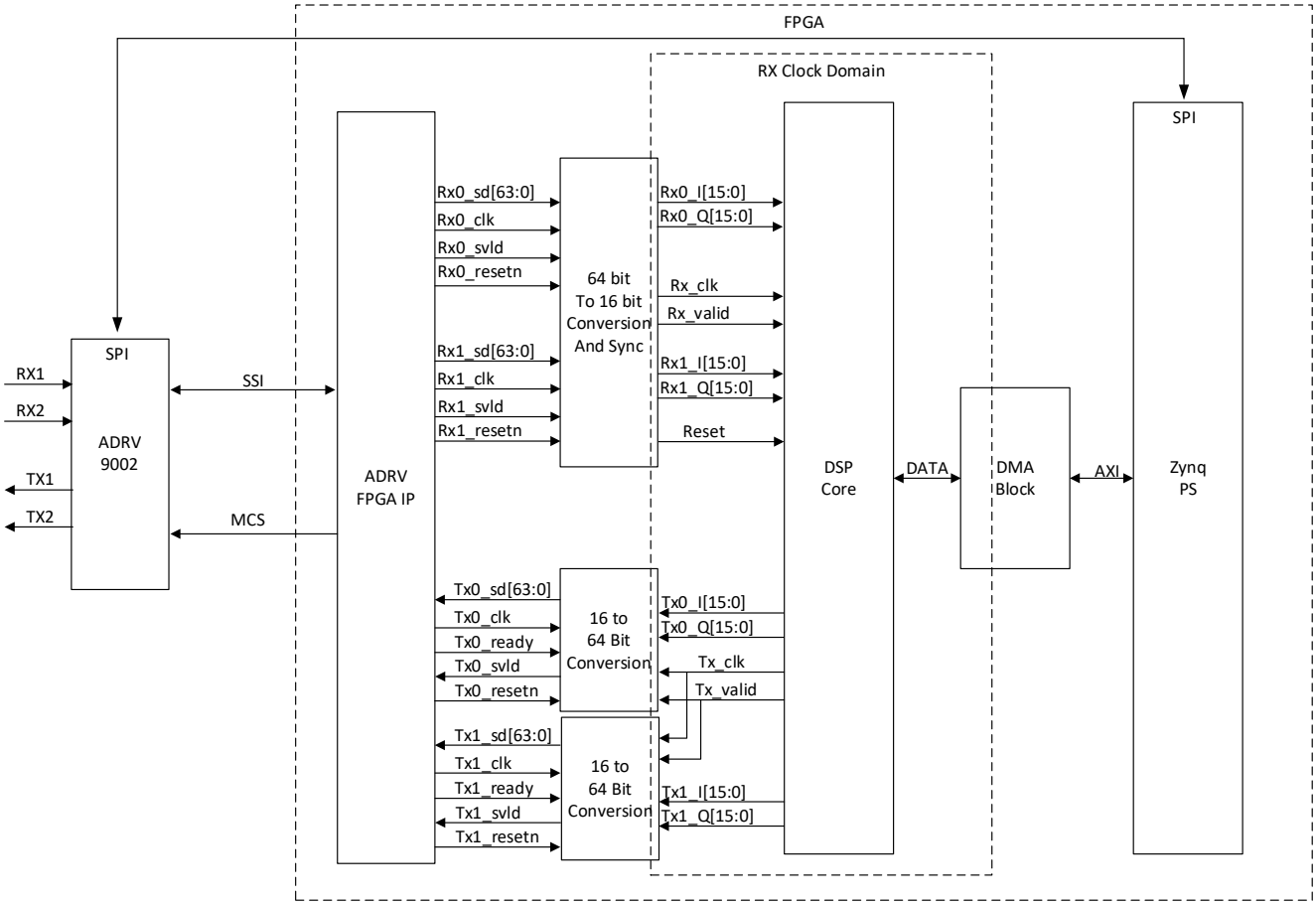


Figure 1 vProtean Reference Design

The digital data interface of the ADRV FPGA IP block for TX and RX are 64-bit buses with two full I and Q samples packed into each 64 bits. Each 64-bit bus has its own clock and data valid signal. Since the DSP core is expecting two 16-bit data buses, one for I and one for Q, the 64-bit buses are converted to 16-bit buses. Since the DSP core uses one clock for all the processing, the ADRV9002 is configured to use the RX clock for the TX clock and the DSP core uses one of the RX clocks. The 64-to-16-bit conversion handles the clock domain crossing between the 64-bit bus clocks and the DSP Core clock. The ADRV FPGA IP block uses the MCS pulses to synchronize the channel data in the 64-bit buses, but for a fully synchronized data path the 16-bit buses also must be synchronized. The use of the data path reset signals Rx1_resetn, Rx2_resetn, Tx1_resetn, and Tx2_resetn can be used to accomplish this synchronization.

When doing MCS on a single vProtean, the MCS pulses can be generated internally in the FPGA using the MCS pulse generation capabilities of the ADI supplied ADRV9002 FPGA IP Block. The MCS pulse connections are as shown in Figure 1 with the FPGA generating the MCS pulse and driving the MCS pulse input on the ADRV9002 transceiver.

The MCS sub block in the ADRV FPGA IP routes the MCS pulse internally for data path synchronization and is used to generate an external MCS pulse which is sent to the transceiver. The MCS sub block can be configured to generate an MCS Pulse or accept an externally generated MCS pulse.

For more information on MCS pulse and timing, refer the Multichip Synchronization section in the ADRV9001 Reference manual.

The Transceiver Evaluation Software (TES) provided by Analog devices can generate initialization code based on the desired configuration. The generated code for MCS can be found in the calibrate.c file near the end. MCS needs to run after calibration and before moving the RF channels into the prime state.

For a signal chip design MCS code does the following:

1. Calls **adi_fpga9001_Mcs_SsiConfigure()** to configure the SSI interface in the ADRV FPGA IP indicating which MCS pulse will use for its synchronization of the SSI. It is typically configured to sync on the last MCS Pulse.
2. Calls **adi_adrv9001_Radio_ToMcsReady()** to transition the ADRV9002 transceiver from the calibrate state into the MCS ready sub state.
3. Calls **adi_fpga9001_Mcs_Configure()** which does two things, selects the internal MCS pulse as the MCS pulse to used and configures the MCS pulse generator in the ADRV FPGA IP block.
4. Calls **adi_fpga9001_Mcs_Start()** which triggers the MCS Pulses and then checks and waits until the ADRV9002 transceiver status indicates that the MCS process is complete.

It is important to note that the first step instructs the ADRV FPGA IP to prepare for MCS pulses. When this happens the RX and TX clocks from the ADRV FPGA IP block are stopped, and the RX and TX reset signals go active. They remain in this state until the MCS sync pulse configured in step 1. The ADRV FPGA block will synchronize the TX and RX data paths in the FPGA IP based on the MCS pulse.

Since the clocks feeding the DSP core are suspended during the MCS process it is important that DSP core be held in reset until the clocks resume. The ADRV FPGA IP block provides four reset signals, one for each of TX and RX clocks. These reset signals go active when the clocks are suspended and inactive when the clocks resume. It is recommended that any downstream data processing uses these reset signals to ensure the data processing is in sync after a MCS process.

There are several ways to test if the MCS process worked. To test the data path and SSI synchronization the adi_adrv9001_Ssi_Loopback_Set() API can be used to loop the TX data back to RX channel. The loopback is implemented in the ADRV9002 transceiver so both the TX and RX channels must be in the RF enabled state for the loopback to work.

This test will only work if the baseband process supports synchronized TX and RX. If synchronized TX and RX are supported, then this loopback combined with Ramp data on the TX lines can be used to verify that the TX and RX SSI paths are Synchronized. Figure 2 shows the capture when the TX data is a RAMP and the TX to RX loopback is enabled.

Although difficult to see in the figure, inspection of individual values showed that the data paths had the same values at every sample.

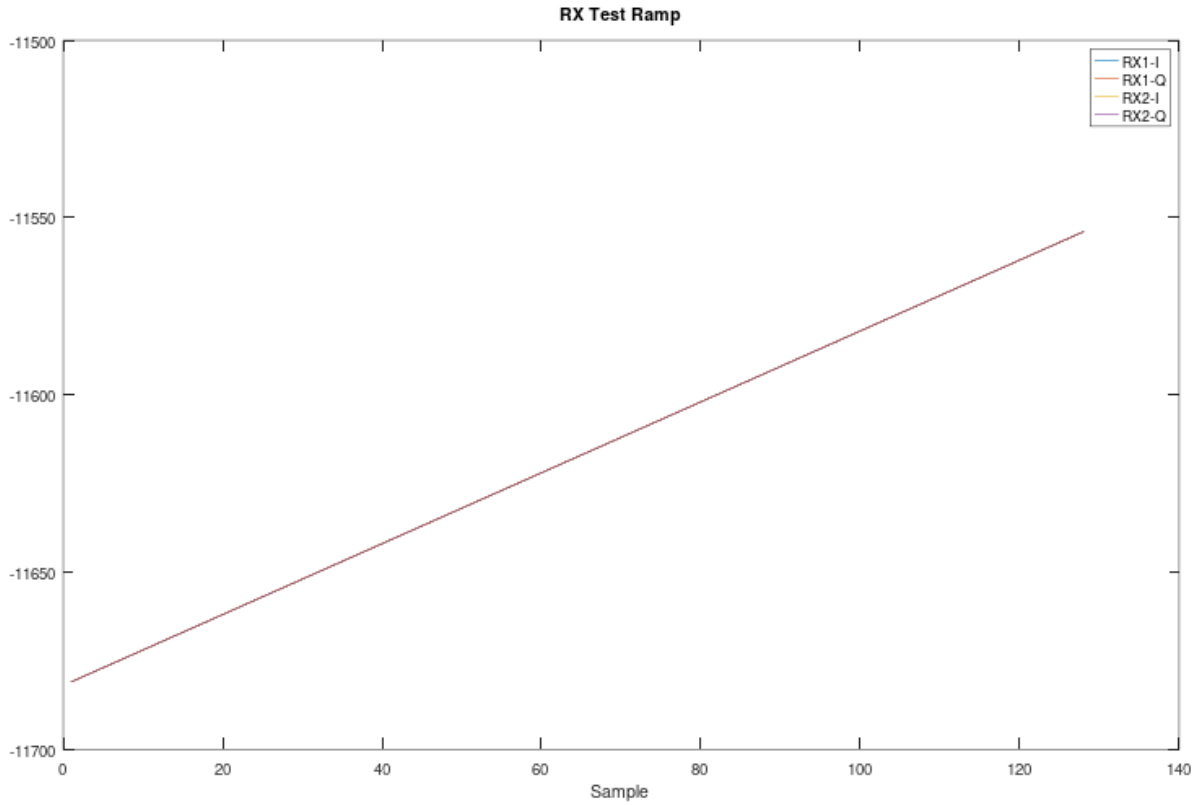


Figure 2 RX Ramp Test Data

To test the RX SSI synchronization alone, the adi_adrv9001_Ssi_Rx_TestMode_Configure() API can be used to configure the ADRV9002 to send digital ramp data on the RX SSI channels. The digital ramp data is generated in the SSI interface so this test will not test the full RX path, but rather only the SSI link. The TX loopback test is a more comprehensive test, but this test can be used to provide further diagnostics if the TX loopback test fails.

The configuration for each channel is done separately so the ramp data will not be aligned at first. The two RX channels need to be configured to send ramp data prior to the MCS pulse. It has been found in testing at Vanteon that once the MCS process is complete, the RX ramp data should be synchronized. Captured RX data should produce a plot like that shown in Figure 4.

To test the RF phase alignment, the vProtean is configured with each RX channel using a different Local Oscillator (LO), and both LOs set to 1.5 GHz. An external 1.502 GHz signal is supplied to both channels through a power splitter making sure that the cables between the splitter and the two RX inputs are of equal lengths.

Figure 3 shows the received I and Q data from two RF channels without MCS. Note that there is no data alignment between the channels.

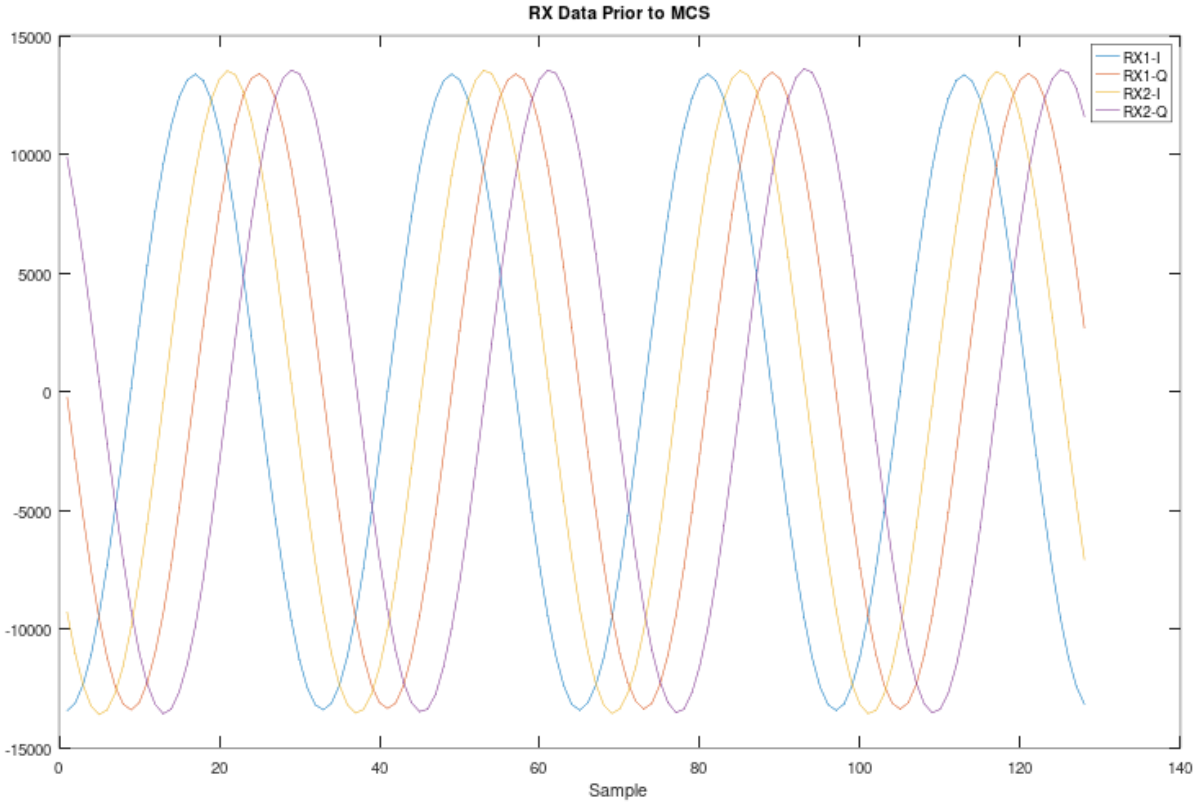


Figure 3 RX Data before MCS

Figure 4 show a capture of the same channels after performing MCS. The data between the channels are now aligned with RX2 I and Q data closely matching that of RX1.

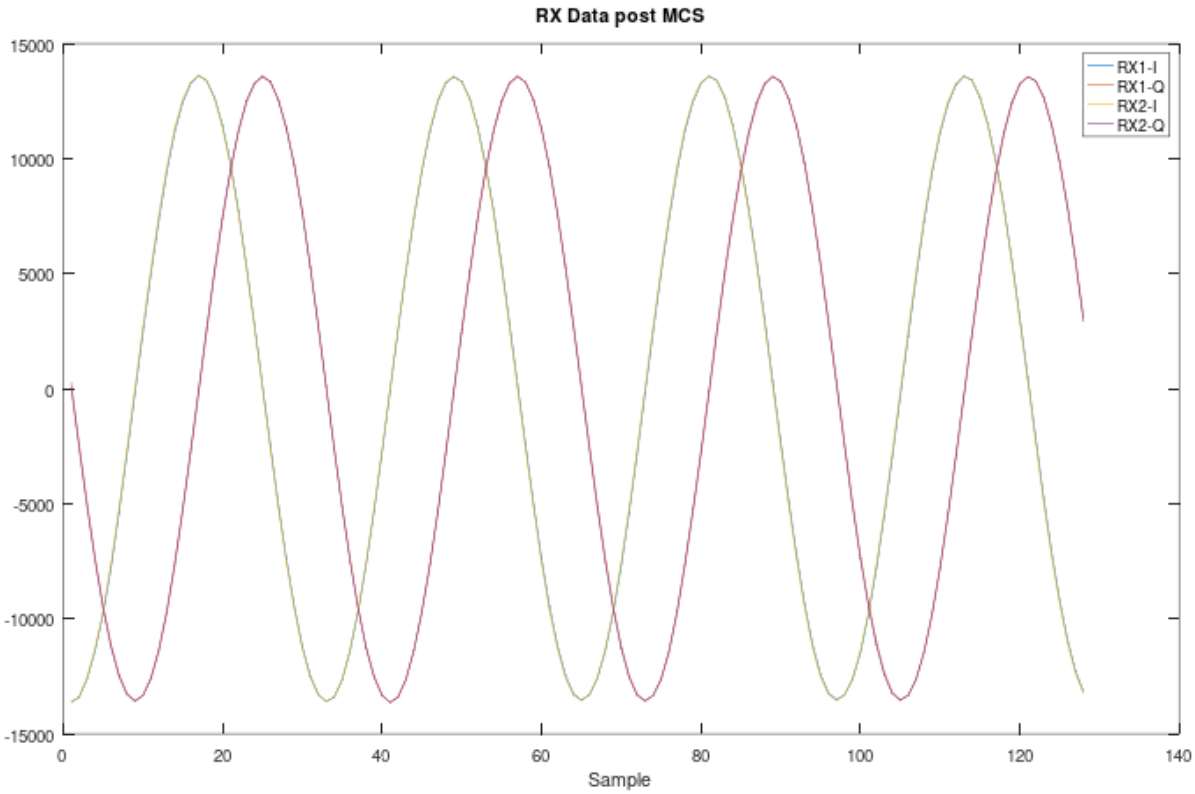


Figure 4 RX Data after MCS

Computing the phase difference between RX1 and RX2 from the RX data, the phase difference is about -1 degree. As can be seen from Figure 3 and Figure 4 MCS with phase alignment provides matched receive channels.

MCS also synchronizes the TX channels. To demonstrate this, TX1 is looped back to RX2 via RF cables and TX2 is looped back to RX1 with cables of the same length. Figure 5 shows the captured data. Measuring the phase from the RX data, the phase between channels is 2.3 degrees.

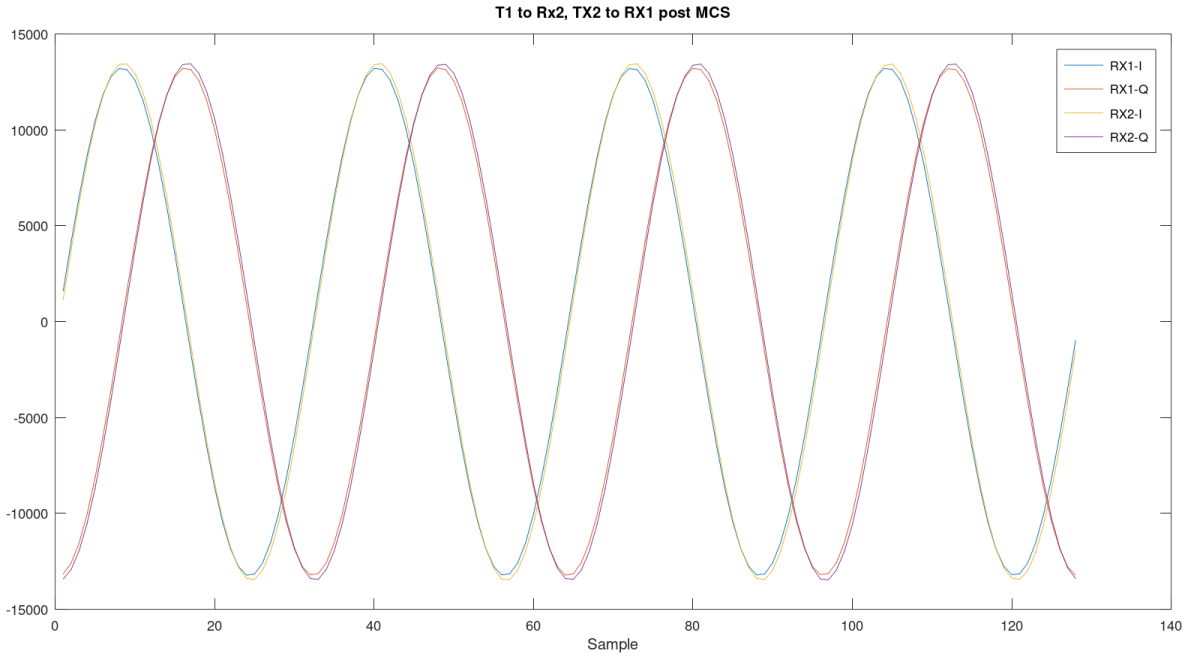


Figure 5 TX1 to RX2 and TX2 to RX1 after MCS

Besides capturing data and computing the phase difference between channels, the ADRV9002 SDK API includes the adi_adrv9001_Mcs_Status_Get() function which returns the RF PLL phase reference to the MCS synchronization point for the two internal LOs. This function can be used to measure the phase difference between the RF PLLs of two channels. In the above tests the API return phase differences slightly lower than those measured. This is most likely due to the external RF connections.

MCS in Systems using more than one vProtean

Figure 6 shows a carrier board that contains two vProteans. In this design the MCS can be generated by the base board FPGA or received from another source.

When developing a system with multiple vProteans, the ADRV9002 transceivers must not only be driven from the same dev clock, but the clock signal paths must be of equal length. The phase of the Dev clocks at all the vProteans is vital to the success of MCS. Likewise, the MCS pulse needs to be distributed to all vProteans with equal length signal paths.

Any misalignment in DEV clock and MCS between ADRV chips will contribute to the phase error.

In Vanteon's design this was achieved by placing a programable delay in MCS path within the base board FPGA. This allowed fine tuning of the MCS pulse arrival time with respect to the DEV clock.

The steps for MCS using multiple vProteans are as follows:

1. Call **adi_fpga9001_Mcs_SsiConfigure()** on all vProteans to configure the SSI interface in the ADRV FPGA IP indicating which MCS pulse will use for its synchronization of the SSI. It is typically configured to sync on the last MCS Pulse.
2. Call **axi_adrv9001_mcs_select_set ()** on all vProteans to configure the ADRV FPGA IP block to use the External MCS pulse.
3. Call **adi_adrv9001_Radio_ToMcsReady()** on all vProteans to transition the ADRV9002 transceivers from the calibrate state into the MCS ready sub state.
4. Generate the MCS pulses – This will be system dependent, for one carrier board the MCS was generated on the carrier board. When multiple carrier boards were used, one carrier board was designated as the primary, which generated the MCS pulse that was then delivered to all other carriers.

5. Call **axi_adrv9001_mcs_status ()** on all vProteans to monitor the MCS status and wait until the ADRV9002 transceiver status indicates that the MCS process is complete.

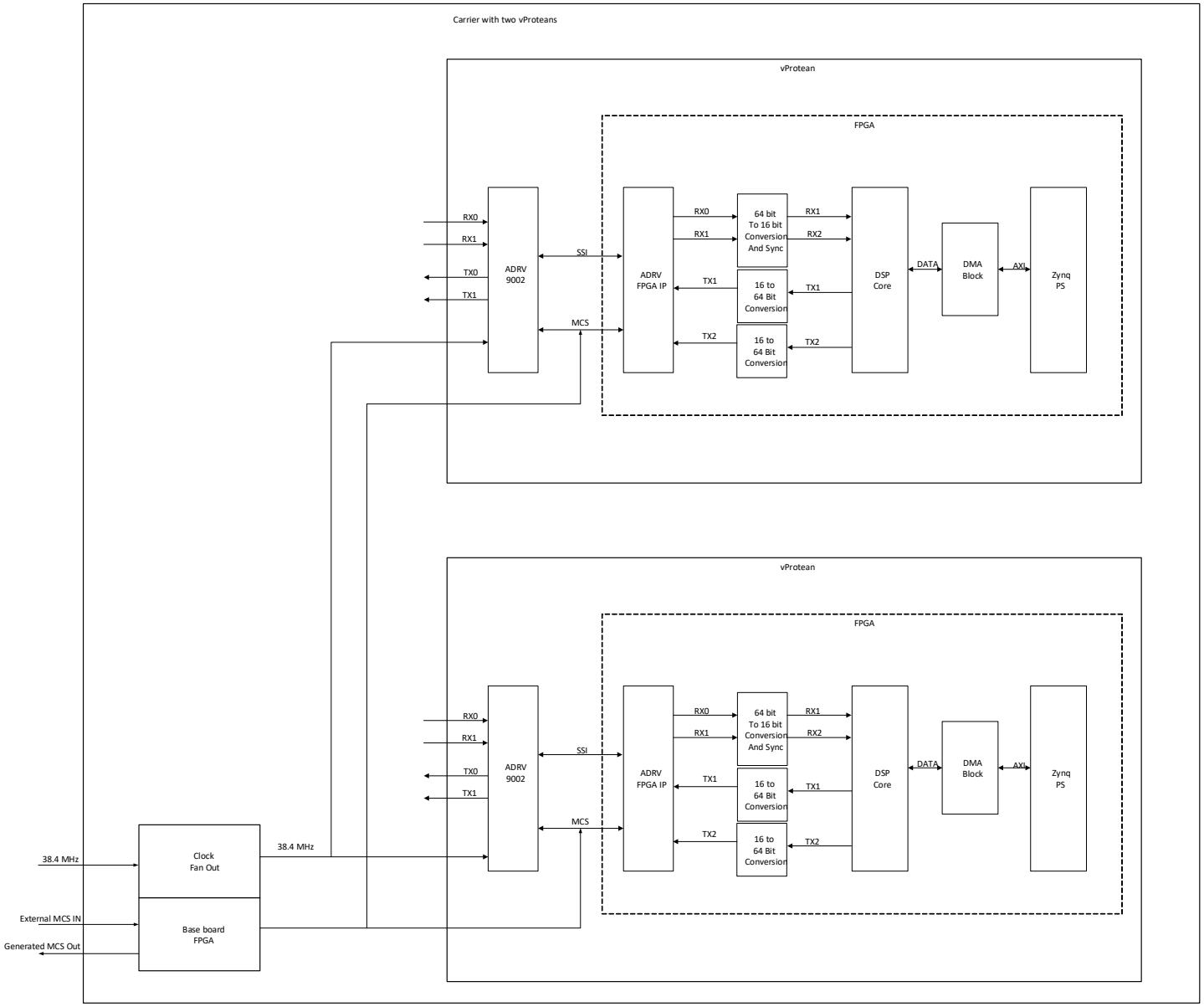


Figure 6 Carrier with Multiple vProtean showing MCS setup

Figure 7 shows RX data captured from a single carrier board. The real data from all channels from the two vProteans on the carrier are shown. Using channel 1 as the reference the measure phase are: RX1 to RX2 -0.6 Degrees, RX1 to RX3 0.1 degree, RX1 to RX4 0.2 degrees.

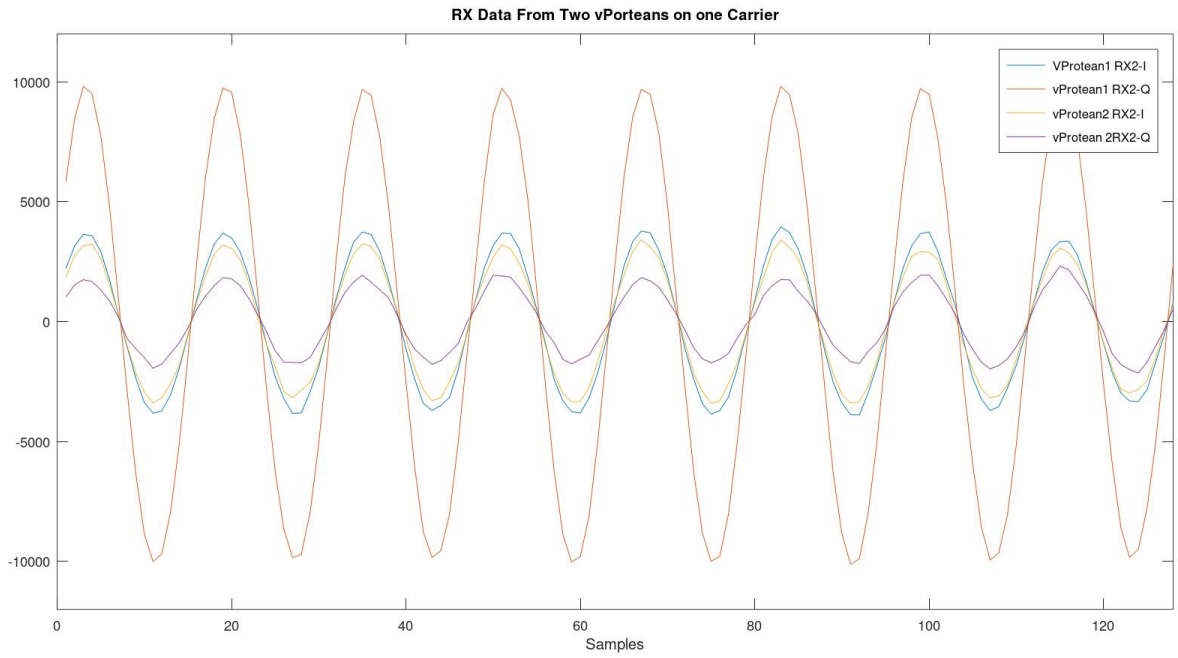


Figure 7 Data From two vProtean on a single carrier board

Vanteon has demonstrated a system of up to 4 carriers for a total of 16 channels. To achieve a synchronization in a system with multiple carriers, a single carrier was chosen as the primary carrier. One vProtean on the primary carrier was selected as a reference. All other vProteans were then measured against this reference to ensure that the DEV clock phase matched and that the MCS pulses arrived at all the vProteans on the same DEV clock edge. When required, the programmable delay in the MCS signal path was used to position the MCS pulse at each vProtean accordingly.

The SSI loopback from TX to RX was used to verify that all vPRoteans were transmitting and receiving data synchronously.

Conclusions

Whether using MCS to synchronize two channels in a single vProtean, or across multiple vProteans, it is important to make sure that the digital data path between the ADRV9002 transceiver and the baseband processor is included in the MCS process. With multiple vProteans, it is important to ensure that the MCS pulse arrives at the all the vProteans on the same DEV clock edge.

To test the RX SSI synchronization only, the RX digital Ramp can be used if it is enabled on all RX channels prior to MCS. For full TX and RX data path validation transmitting the same signal on the two TX channels synchronously and using the TX loopback will allow validation of the digital data paths.

Finally the SDK API function adi_adrv9001_Mcs_Status_Get() can be used to retrieve the internal RF PLL measured phase.

To learn how Vanteon's expertise can help, contact us at info@vanteon.com.

References

vProtean Web Page: https://vanteon.com/vprotean/

vProtean Datasheet: https://vanteon.com/wp-content/uploads/2021/10/vProtean_DS.pdf

ADI ADRV9002 Web Page: https://www.analog.com/en/applications/technology/sdr-radioverse-pavilion-home/wideband-transceivers/adrv9002-transceiver-ic.html

ADI ADRV9002 Datasheet: https://www.analog.com/media/en/technical-documentation/data-sheets/adrv9002.pdf

ADI ADRV9002 User Guide: https://www.analog.com/media/en/technical-documentation/user-guides/adrv9001-system-development-user-guide-ug-1828.pdf