

Sail River Gen III O-RU (WS-RLN-00098)

Hardware User Manual

Last updated: **2025-05-22**

Altera Confidential

© Altera Corporation. Altera, the Altera logo, the 'a' logo, and other Altera marks are trademarks of Altera Corporation.

Altera and Intel warrant performance of its FPGA and semiconductor products to current specifications in accordance with Altera's or Intel's standard warranty as applicable, but reserves the right to make changes to any products and services at any time without notice.

Altera and Intel assume no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera or Intel.

Altera and Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

Altera Confidential

Contents

1.0	Introduction	6
1.1	Terminology	6
1.2	Reference Documents	6
1.3	Scope.....	6
1.4	Prerequisites	6
2.0	GSRD Boot Flow	8
3.0	Hardware Setup	9
3.1	Hardware Modifications.....	9
3.2	Hitek Devkit & RF EVM Setup	13
3.3	Clocks	19
3.4	LLS-C1	19
3.5	LLS-C3	21
4.0	Software Environment Setup	22
4.1	Ubuntu 22.04 Setup.....	22
4.2	Quartus Software Installation	22
4.3	Environment Script	24
5.0	Quartus Project Compilation	25
5.1	Quartus IP Patches	25
5.2	Compile Directives	25
5.2.1	Short/Long PRACH	25
5.2.2	Enable One or Two Component Carriers (CC).....	25
5.3	Generating the .sof	26
5.3.1	GUI Method	26
5.3.2	Script Method	26
5.4	Pre-synthesized .sof(s).....	26
6.0	Creating the SD Card Image	27
6.1	Environment Setup	27
6.2	Yocto Image Build.....	27
6.3	Injecting U-Boot into the FPGA .sof (RBF/JIC)	27
6.4	Insert .rbf into SD Card Image	28
7.0	Loading the SD Card Image	29
8.0	O-RU Bring Up.....	30
8.1	Trimble GM200 Configuration	30
8.2	Cisco Nexus 93180YC-FX3 Switch Configuration	31
8.2.1	Link/VLAN Configuration	31
8.2.2	G.8275.1 PTP Configuration.....	31
8.2.3	SyncE Configuration	32
8.3	Keysight S5040A (ORAN Studio) Configuration	32
8.4	Programming the FPGA Flash (JIC)	34
8.5	Serial Configuration	35
8.6	Resizing the Root Filesystem	35
8.7	cAD9528 Clock Configuration	36
8.8	IP Configuration	36
8.9	Source MAC Address Configuration	37
8.10	Running ptp4l.....	38

Altera Confidential

8.11	Loading the O-RU Drivers & User Application	38
8.12	Koror Configuration	39
8.13	Test Case Configuration	40
Appendix A. Updating the Skyworks Si5518 Programming .bin Files		47
A.1	Si5518 Programming via U-Boot	48
A.2	Si5518 Programming via HPS	48
9.0	Revision History	50

List of Figures

Figure 2-1.	GSRD Boot Flow	8
Figure 3-1.	Hitek ADI Carrier Board - TEE Sites Post Rework	11
Figure 3-2.	RF EVM - AD9528 Master (Default)	12
Figure 3-3.	Hitek Devkit & RF EVM Reference Setup (Front)	13
Figure 3-4.	Hitek Devkit & RF EVM Reference Setup (Rear)	14
Figure 3-5.	Hitek Devkit & RF EVM Reference Setup (Top)	15
Figure 3-6.	FMC Connector	16
Figure 3-7.	Timing Module - U.FL to RF EVM Ref. Clock & 1PPS Header	17
Figure 3-8.	RF EVM - Si5518 Ref. Clk Input	18
Figure 3-9.	Hardware Setup for LLS-C1	20
Figure 3-10.	Hardware Setup for LLS-C3	21
Figure 4-1.	Quartus Prime Pro Version Selection Example for Linux	22
Figure 6-1.	Loop Device Creation	28
Figure 8-1.	Trimble GM200 Front Panel	30
Figure 8-2.	Trimble GM200 Login Prompt	30
Figure 8-3.	ORAN Studio - Port Settings	33
Figure 8-4.	ORAN Studio - Time Sync Configuration	34
Figure 8-5.	Configuring AD9528 Clocks	36
Figure 8-6.	Configuring Source MAC Address	37
Figure 8-7.	ORAN Studio - Configured Source MAC Address	37
Figure 8-8.	AGX CLI - ptp4l Converged Output	38
Figure 8-9.	AGX CLI - O-RU Driver Load (SSH)	38
Figure 8-10.	AGX CLI - ToD Reset via O-RU User App (SSH)	39
Figure 8-11.	AGX CLI - MATLAB Connection	39
Figure 8-12.	AGX CLI - Koror Configuration Output for Digital Loopback	40
Figure 8-13.	MATLAB GUI - Connection Established with Hitek Devkit	40
Figure 8-14.	MATLAB GUI - Test Case Selection	41
Figure 8-15.	MATLAB GUI - eCPRI & ORAN Register Configuration	42
Figure 8-16.	MATLAB GUI - DXC_ss Registers	42
Figure 8-17.	MATLAB GUI - PRACH Register	43
Figure 8-18.	ORAN Studio - Load and Play the PCAP Stimulus	44
Figure 8-19.	ORAN Studio - ORAN TX Packets (DL)	44
Figure 8-20.	MATLAB GUI - Radio Config Register Configuration	45
Figure 8-21.	ORAN Studio - Rx/UL Incrementing Counters	46

List of Tables

Table 1-1.	Terminology	6
Table 1-2.	Reference Documents	6
Table 2-1.	Description of Boot Flow Stages	8

Altera Confidential

Table 3-1. Hitek ADI Carrier Board - TEE Site Requirements	9
Table 3-2. Clocks	19
Table 9-1. Document Revision History	50

1.0 Introduction

The objective is to generate and bring-up the software and hardware of the Sail River Gen III design with F-Tile 1588 on the Hitek eSOM board.

1.1 Terminology

Table 1-1. Terminology

Term	Description

1.2 Reference Documents

Table 1-2. Reference Documents

Document	Document No./Location
WS-VTR-00115 Sail River Gen III	WS-VTR-00115

1.3 Scope

The scope of this user manual is to provide an insight into the bring up of hardware and software of the Sail River Gen III design.

1.4 Prerequisites

The following list is the BOM for Sail River Gen III validation.

- **Hitek Agilex eSOM7-2F + ADI Carrier Board + Timing Module**
 - **FPGA:** AGFB027R24C2E2VR2
 - **DPLL:** Hitek Si5518 Timing Module
 - USB-C/Micro USB cable for UART
 - Dependency on carrier board revision
 - Micro USB cable for USB Blaster II (JTAG)
 - AC Power adapter
 - CAT5e cable with RJ45 terminations (HPS MGMT)
 - U.FL to SMA cable
- **ADRV904x EVM (Koror)**
 - AC Power Adapter
 - SMA to SMA coax cable
- **SFP Modules & Fiber Cabling:**
 - Cisco SFP-25G-SR-S: <https://www.fs.com/products/67991.html>

Altera Confidential

1.0 Introduction

- Cisco QSFP-100G-SR4-S: <https://www.fs.com/products/48354.html>
- 3m (10ft) MTP® (Female) to 4 LC UPC Duplex 8 Fibers Type B Plenum (OFNP) OM4 50/125 Multimode Elite Breakout Cable, Magenta:
<https://www.fs.com/products/68053.html?attribute=6193&id=2069586>
- **PTP Grandmaster:** Trimble GM200
- **ORAN O-DU Emulator:** Keysight S5040A
- **PTP T-BC Switch:** Cisco 93180YC-FX3
- **Oscilloscope:** Tektronix MDO3054
- **Build/Host Machine OS:** Ubuntu 22.04

Note: The above optics and cable have been used for validation. Users may use other manufacturers.

2.0 GSRD Boot Flow

Figure 2-1. GSRD Boot Flow



The following table presents a short description of the different boot stages:

Table 2-1. Description of Boot Flow Stages

Stage	Description
SDM	Secure Device Manager boots first
U-Boot SPL	Configures IO, FPGA, brings up SDRAM
ATF	Arm Trusted Firmware, provides SMC handler
U-Boot	Loads Linux kernel
Linux	Operating system
Application	User application

For more information, please refer to Intel Agilex SoC Boot User Guide and Intel Agilex Hard Processor System Technical Reference Manual (Booting and Configuration chapter).

3.0 Hardware Setup

3.1 Hardware Modifications

Prior to setting up, the Hitek devkit and RF EVM may require solder reworks to ensure proper clock routing is achieved. The table and figures below represent the correct configuration and provide a visual aid.

The TEE04 site controls the incoming DSP clock to the FPGA. To ensure that the design can run standalone, the DSP clock is currently transmitted via the Hitek Si5518 Timing Card. To reroute this incoming clock so it is received via the FMC, the resistors will need to be moved.

Table 3-1. Hitek ADI Carrier Board - TEE Site Requirements

eSOM TEE #	PCB Ref	Populated?	Value	Size	Description
TEE 01	C125	NO			JESD XCVR reference clock from AD9528 (RF EVM) via FMC
	C124	NO			
	C121	YES	0.1uF	0402	
	C120	YES	0.1uF	0402	
TEE 02	C118	NO			Unused
	C123	NO			
	C119	YES	0.1uF	0402	
	C122	YES	0.1uF	0402	
TEE 03	R166	NO			Unused
	R175	NO			
	R173	YES	0Ω	0402	
	R174	YES	0Ω	0402	
TEE 04	R336	YES	0Ω	0402	DSP Clock from Si5518 to FPGA via Timing Card
	R339	YES	0Ω	0402	
	R337	NO			DSP Clock from FMC. Resistor rework required. Reference details above.
	R338	NO			
TEE 05	R160	NO			
	R169	NO			
	R159	YES	0Ω	0402	

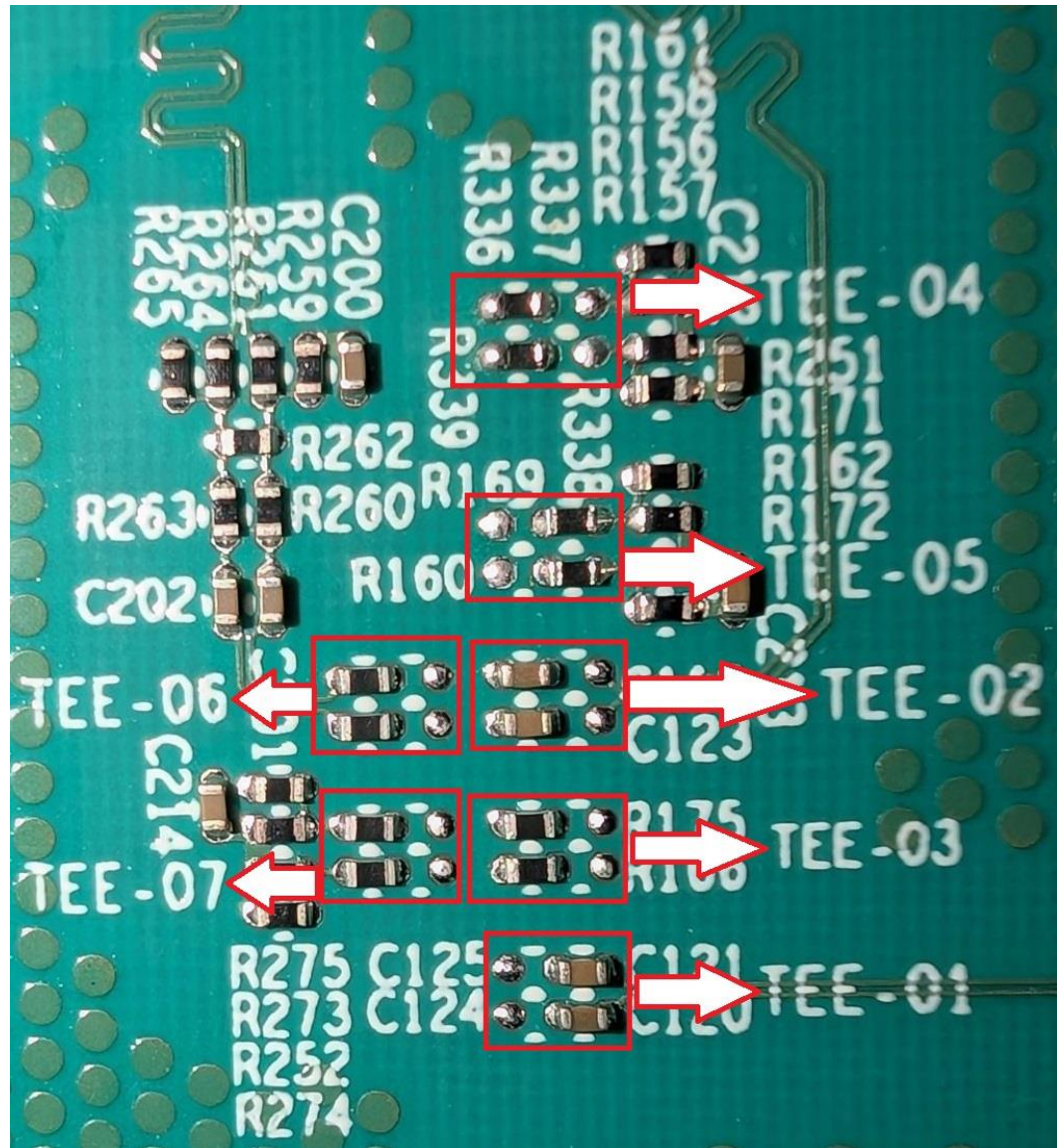
Altera Confidential

3.0. Hardware Setup

eSOM TEE #	PCB Ref	Populated?	Value	Size	Description
	R170	YES	0Ω	0402	Sysref from AD9528 (RF EVM) via FMC
TEE 06	R167	NO			
	R178	NO			
	R168	YES	0Ω	0402	Unused
	R177	YES	0Ω	0402	
TEE 07	R140	NO			
	R165	NO			
	R163	YES	0Ω	0402	Unused
	R164	YES	0Ω	0402	

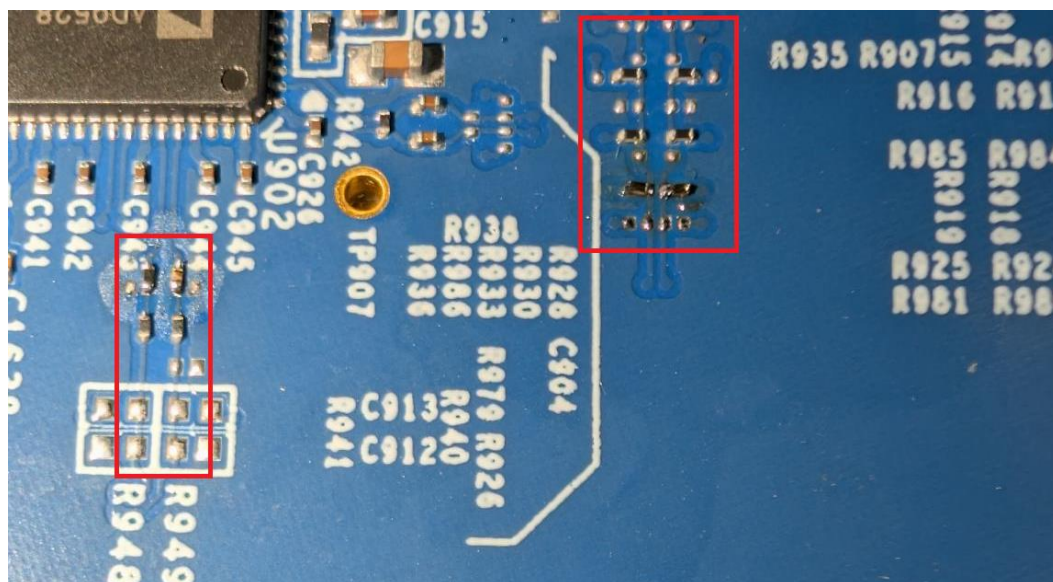
Altera Confidential

Figure 3-1. Hitek ADI Carrier Board - TEE Sites Post Rework



Sail River Gen III O-RU (WS-RLN-00098): Hardware User Manual

Figure 3-2. RF EVM - AD9528 Master (Default)



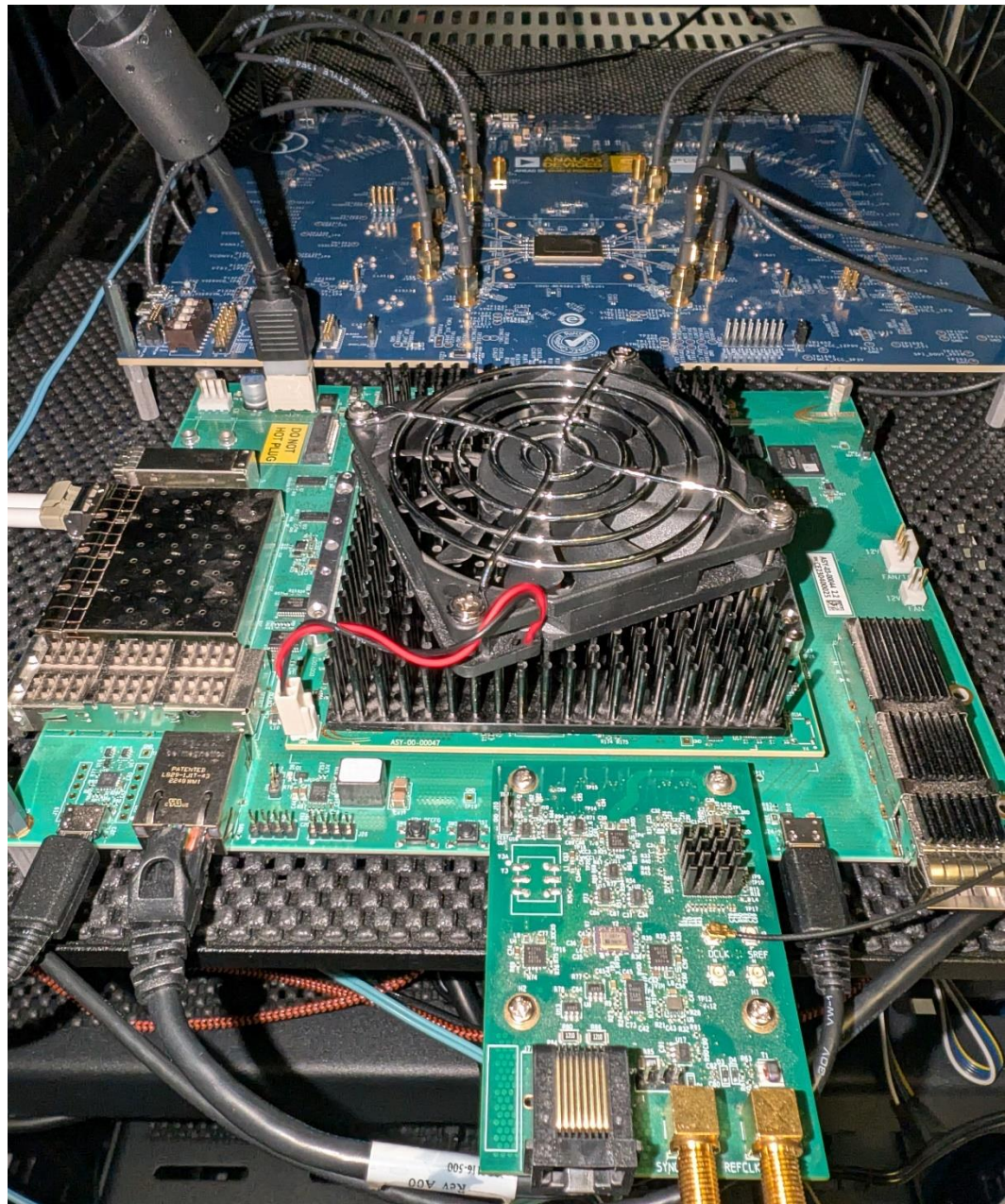
Altera Confidential

3.0 Hardware Setup

3.2 Hitek Devkit & RF EVM Setup

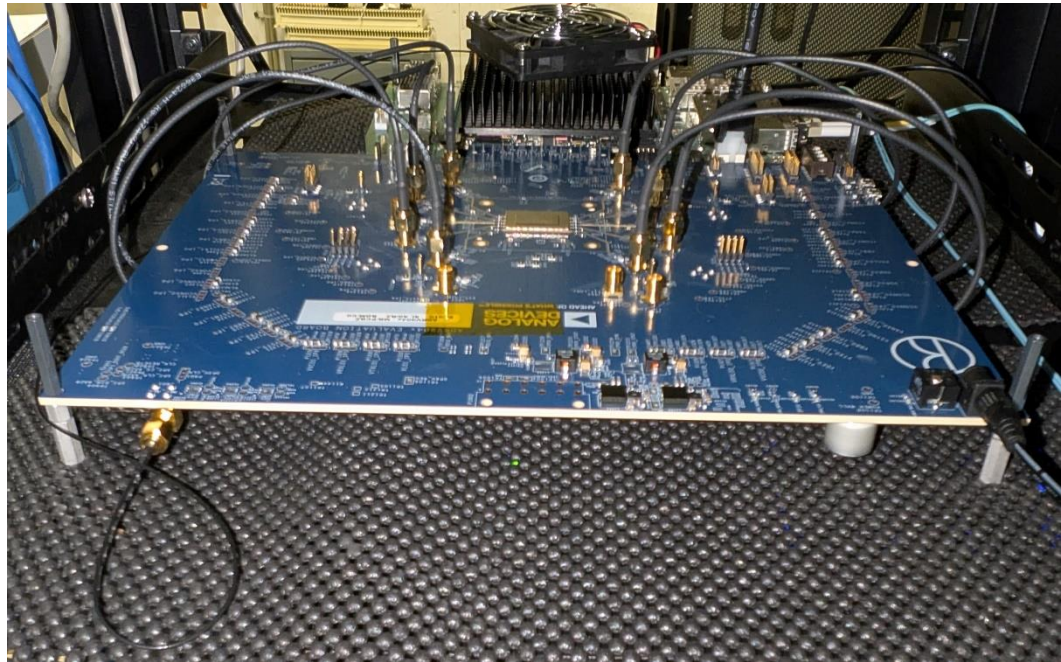
The figures in this section are reference photos displaying connections between the Hitek Devkit, Hitek Timing Module, and RF EVM. Block diagrams will be provided in the following subsections LLS-C1 and LLS-C3.

Figure 3-3. Hitek Devkit & RF EVM Reference Setup (Front)



Altera Confidential

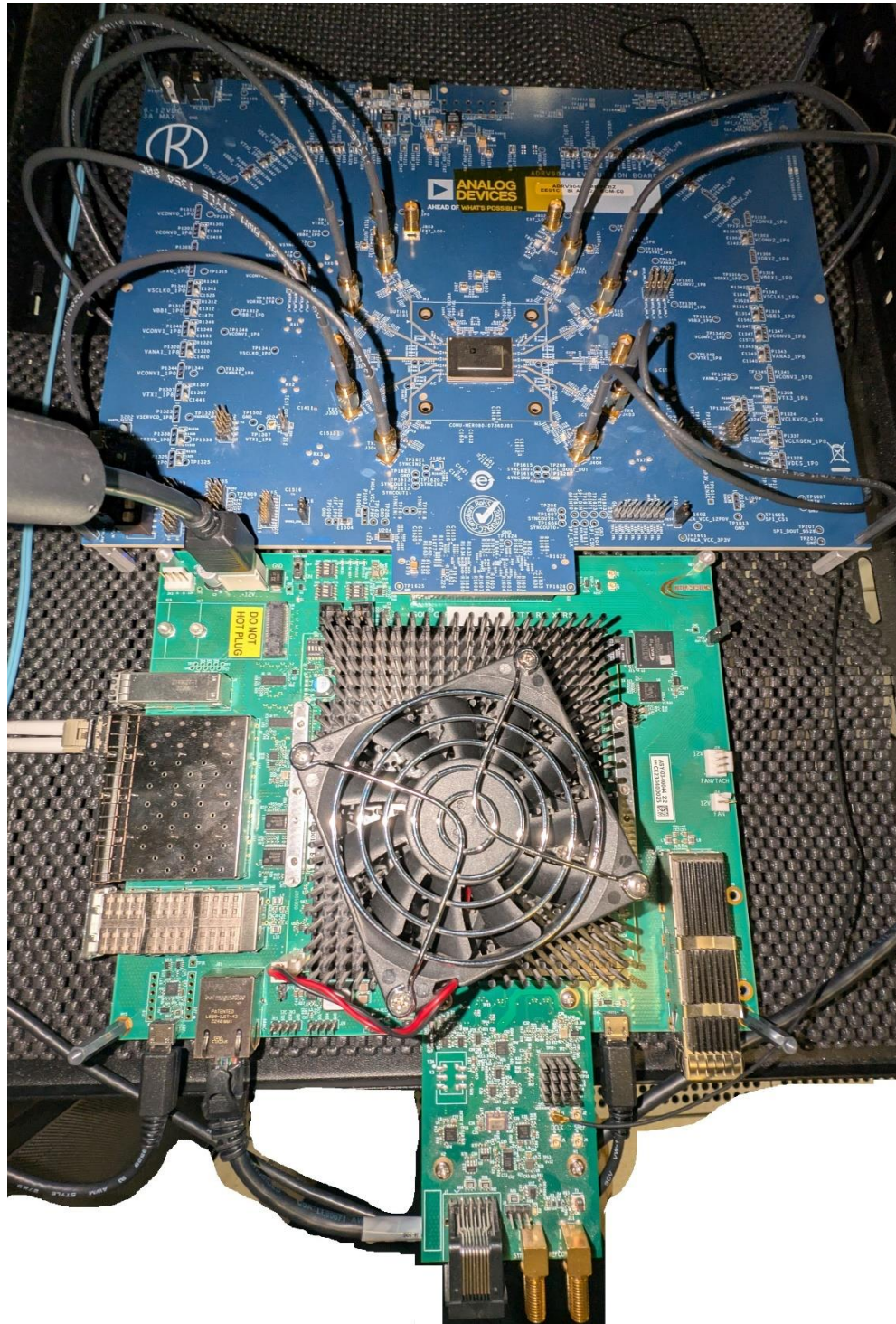
Figure 3-4. Hitek Devkit & RF EVM Reference Setup (Rear)



Altera Confidential

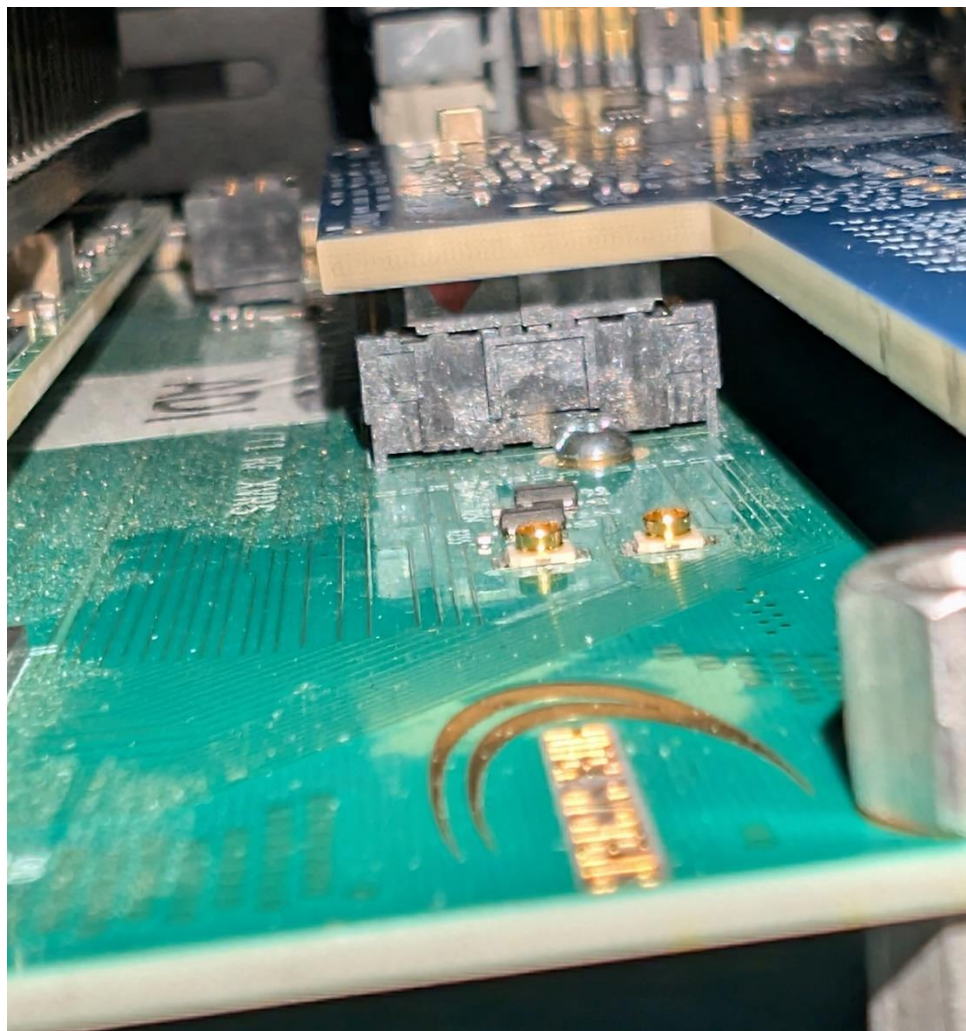
3.0 Hardware Setup

Figure 3-5. Hitek Devkit & RF EVM Reference Setup (Top)



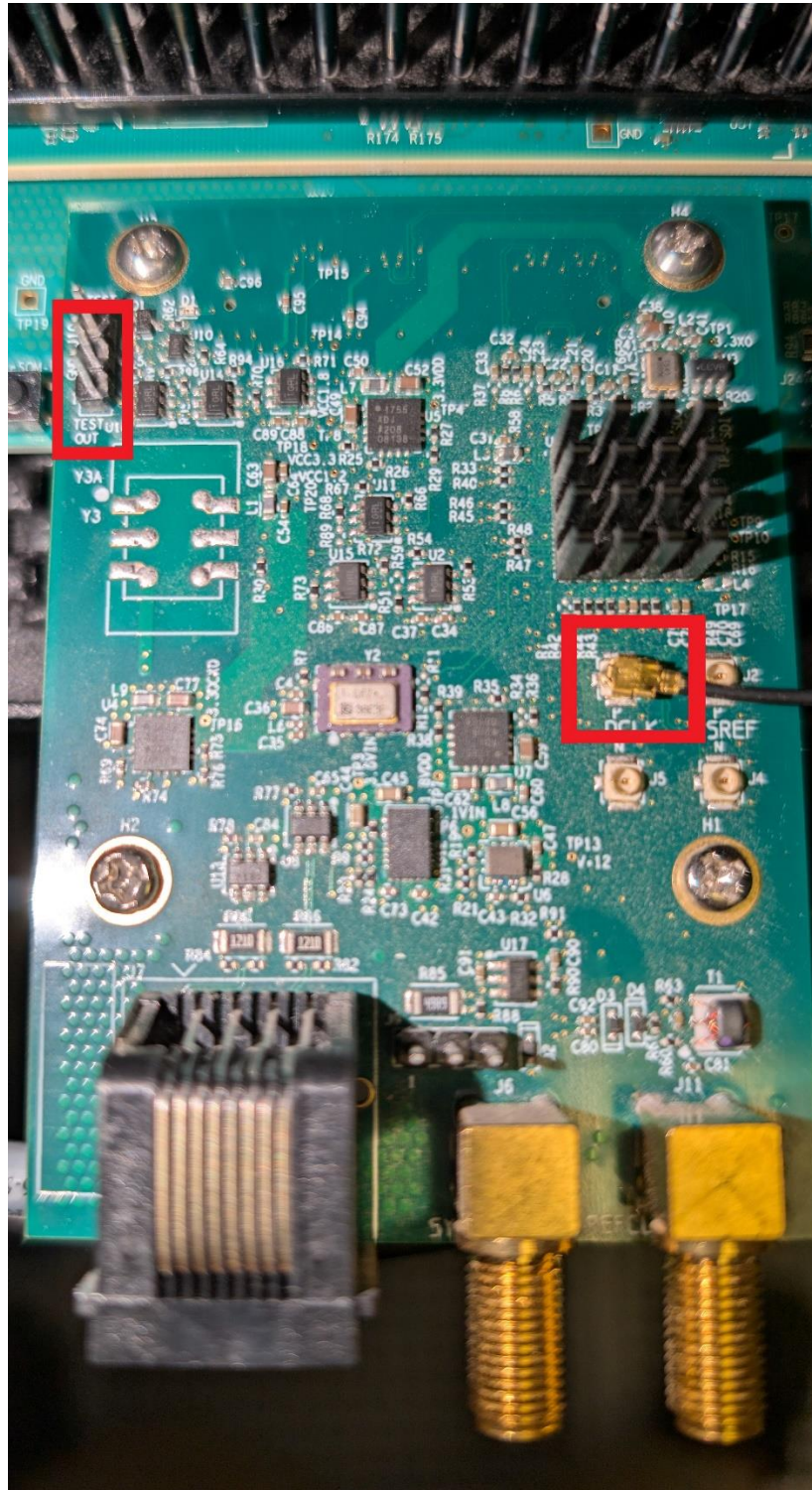
Altera Confidential

Figure 3-6. FMC Connector



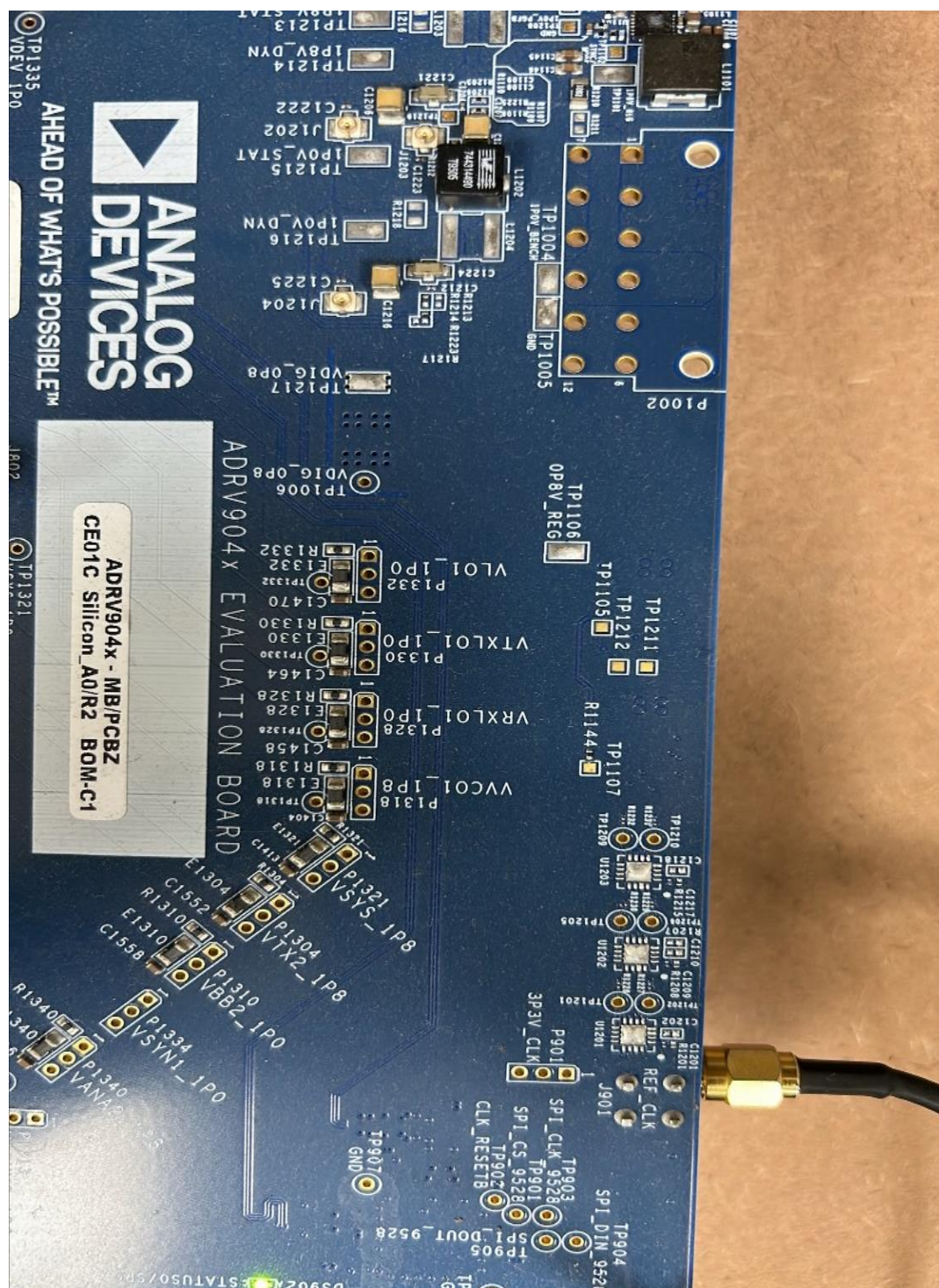
Altera Confidential

Figure 3-7. Timing Module - U.FL to RF EVM Ref. Clock & 1PPS Header



Altera Confidential

Figure 3-8. RF EVM - Si5518 Ref. Clk Input



Altera Confidential

3.0 Hardware Setup

3.3 Clocks

The table below describes clocks required by the design as reflected in the Quartus project.

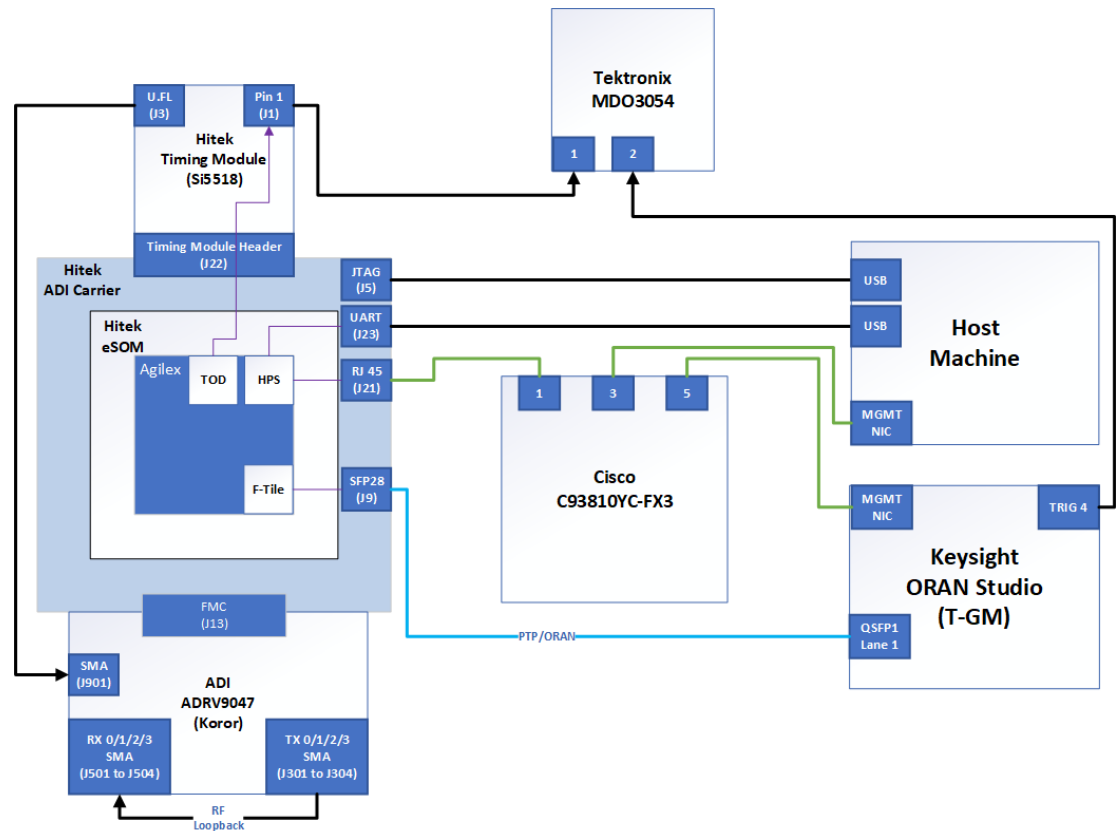
Table 3-2. Clocks

Clock name	Description	FPGA Direction	Clock Frequency	FPGA Pin
hssi_cdr_clk_out	RX recovery clock (SyncE downstream) to Timing Module	Output	390.625 MHz	CY32/DA33
ftile_clk_ref	F-Tile HIP reference clock (SyncE upstream) from Timing Module	Input	156.25 MHz	BC49/BE49
ftile_master_todclk_ref	ToD 1588-steered clock from Timing Module	Input	156.25 MHz	CT42/CR43
master_tod_top_0_pulse_per_second	1PPS debug clock	Output	1 Hz	Y42
jesd_xcvr_clk	JESD XCVR reference clock from AD9528 (RF EVM) via FMC	Input	122.88 MHz	AJ49/AH48
dsp_clk_491m	DSP reference clock from SI5518	Input	491.52MHz	DF34/DE35
sysref	JESD Sysref from AD9528 (RF EVM) via FMC	Input	3.84 MHz	CP40/CN41

3.4 LLS-C1

The below figure represents a diagrammatic test environment needed for synchronization configuration LLS-C1.

Figure 3-9. Hardware Setup for LLS-C1



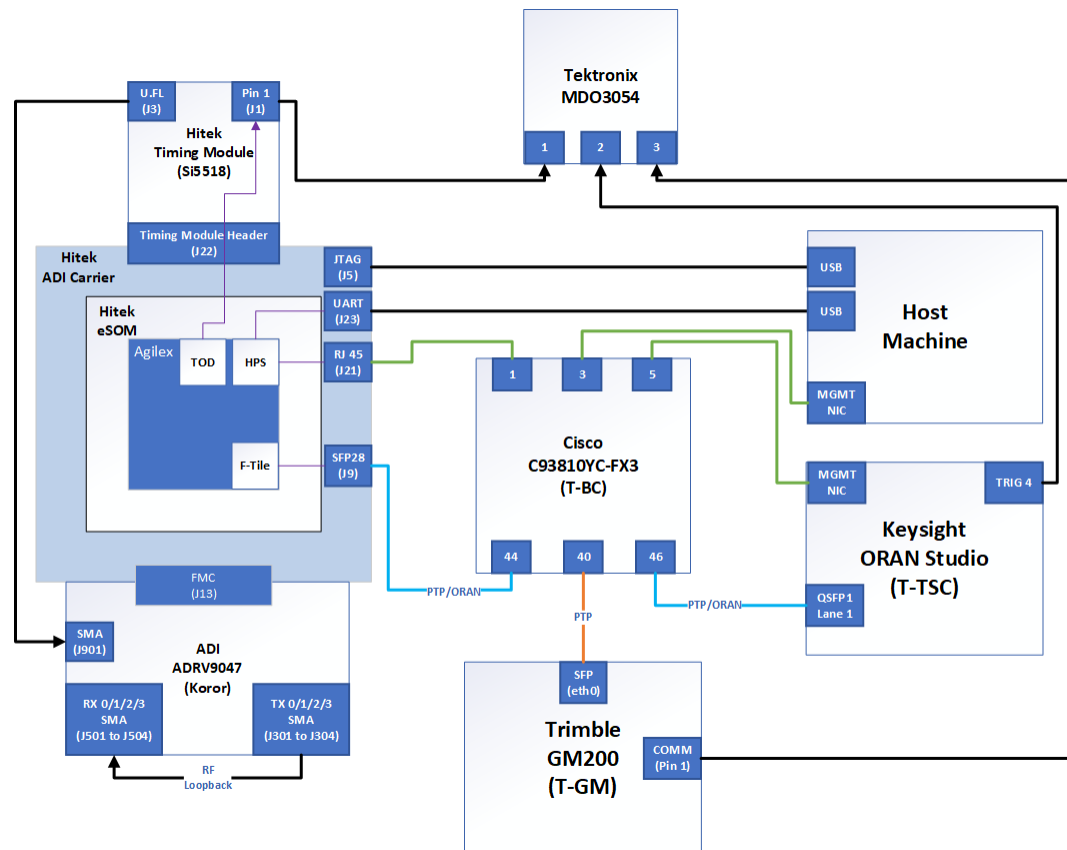
Altera Confidential

3.5 LLS-C3

The below figure represents a diagrammatic test environment needed for synchronization configuration LLS-C3.

This will be the configuration assumed throughout the document.

Figure 3-10. Hardware Setup for LLS-C3



Altera Confidential

4.0 Software Environment Setup

This build process assumes an **Ubuntu 22.04 LTS** host machine. The provided **env.sh** script will set up the environment variables used throughout this document.

4.1 Ubuntu 22.04 Setup

Install the following packages:

```
$ sudo apt install libncurses-dev libncurses5 flex bison libssl-dev swig u-boot-tools gawk
wget git diffstat unzip texinfo gcc build-essential chrpath socat cpio python3 python3-dev
python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2
libegl1-mesa libstdl1.2-dev pylint xterm python3-subunit mesa-common-dev zstd liblz4-tool
minicom file locales libacl1 libpcap-dev cmake
```

Generate the en_US.UTF-8 locale files:

```
$ sudo locale-gen en_US.UTF-8
```

Due to changes from Ubuntu 20.04 to Ubuntu 22.04, **pylint** will need to be symlinked for portability:

```
$ cd /usr/bin
$ sudo ln -s pylint pylint3
```

4.2 Quartus Software Installation

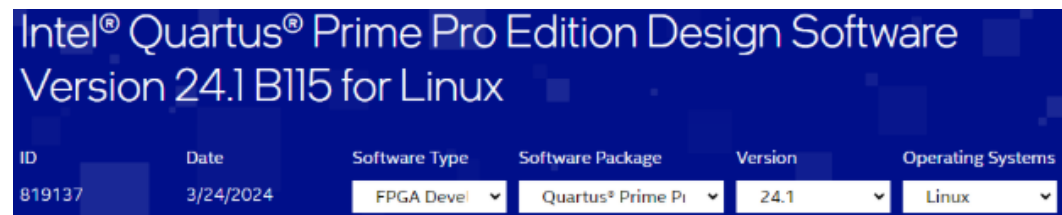
The Quartus Prime Pro software can be found at this link:

<https://www.intel.com/content/www/us/en/products/details/fpga/development-tools/quartus-prime/resource.html>

Install the following:

- **Software Package:** Quartus Prime Pro
- **Version:** 24.1
- **Operating System:** Linux

Figure 4-1. Quartus Prime Pro Version Selection Example for Linux



4.0 Software Environment Setup

Install the following required IPs. It is intended to install the 24.3 version of the ORAN IP.

Fronthaul Compression IP:

https://downloads.intel.com/akdlm/software/webcores/24.1/122.5/fh_comp_top-24.1.122-linux.run

eCPRI:

<https://downloads.intel.com/akdlm/software/webcores/24.1/122.5/ecpri-24.1.122-linux.run>

ORAN:

<https://downloads.intel.com/akdlm/software/webcores/24.3/124.1/oran-24.3.124-linux.run>

4.3 Environment Script

The **env.sh** script's purpose is to set environment variables necessary for the Quartus installation path and Yocto build. The following variable(s) should be modified machine to machine:

```
QUARTUS_PATH_PREFIX=/path/to/intelFPGA_pro
```

All other paths are relative to the release's directory structure. Below is a mapping between the directory/file structure and environment variable used in this user manual and environment script:

- release/ - \$REL_DIR
- \$REL_DIR/software/yocto - \$YOCTO_SRC_DIR
- \$REL_DIR/software/fpga/sofs - \$SOF_DIR

The **env.sh** script will be run during the [Creating SD Card Image \(Yocto\)](#) section of this document.

5.0 Quartus Project Compilation

The following are the prerequisites for **SOF** generation:

- **OS:** Windows/Linux
- **Tools/Applications:** Quartus Prime Pro 24.1

5.1 Quartus IP Patches

In this section, the IPs will be patched. The patches are provided in \$REL_DIR/software/patches/quartus and the below command will recursively transfer the files to their proper locations:

```
$ cd $REL_DIR/software/patches/quartus
$ cp -rv * $QUARTUS_PATH
```

5.2 Compile Directives

There are compiler directives that require modification prior to compiling the design. See below sections for details on modifying the contents of the following file:

```
$REL_DIR/fpga/src/ip/custom/common/def_param.vh
```

5.2.1 Short/Long PRACH

To enable Short PRACH:

```
`define SHORT_PRACH_FORMAT
```

To enable Long PRACH:

```
//`define SHORT_PRACH_FORMAT
```

5.2.2 Enable One or Two Component Carriers (CC)

To enable one CC only:

```
`define NUM_CC_ONE
//`define NUM_CC_TWO
```

To enable two CCs:

```
//`define NUM_CC_ONE
`define NUM_CC_TWO
```

5.3 Generating the .sof

There are two methods available to generating the .sof: GUI and script. Note that pre-synthesized .sof(s) have been provided as per the next section.

5.3.1 GUI Method

Open the Quartus Project File located here:

```
$REL_DIR/fpga/quartus/ag_esom_top.qpf
```

To start the compilation, click the Play button next to **Start Compilation**

The following file(s) will be generated:

```
$REL_DIR/fpga/quartus/output_files/ag_esom_top.sof
```

5.3.2 Script Method

Change to the following directory and run the **generate_build.sh** script:

```
$ cd $REL_DIR/fpga/quartus/build_script  
$ ./generate_build.sh stp_disabled
```

The sysid will be automatically incremented by 1 and the compilation will start.

5.4 Pre-synthesized .sof(s)

Pre-synthesized validated .sof(s) have been provided with the release.

Long PRACH + 1CC:

```
$REL_DIR/fpga/sofs/long_prach_1cc.tgz
```

Long PRACH + 2CC:

```
$REL_DIR/fpga/sofs/long_prach_2cc.tgz
```

Long PRACH + 1CC will be used during the Yocto build process.

6.0 Creating the SD Card Image

This section describes the process to generate an SD card image for the Agilex platform using the Yocto build system. The image will then be updated to include the .rbf necessary to program the FPGA's fabric.

As a base, the following components are used:

1. **Yocto:** Kirkstone
2. **ARM Trusted Firmware:** 2.11.0
3. **U-Boot:** 2024.04
4. **Linux Kernel:** 6.1.38-lts

6.1 Environment Setup

This section assumes that the prior section [Software Environment Setup](#) has been followed.

6.2 Yocto Image Build

To start the Yocto build process:

```
$ cd $YOCTO_SRC_DIR  
./yocto.sh
```

This script will perform the following tasks:

- Clones the gsr-d-socfpga meta-layer from GitHub
- Copies and sets up custom build scripts and layers (meta-altera-oru, meta-intel-fpga, and meta-intel-fpga-refdes).
- Runs the Yocto build environment setup script (**agilex-gsr-d-build.sh**).
- Builds the image with BitBake.
- Generates the SD Card image **console-image-minimal-agilex.wic**.
- Moves the generated SD Card image and u-boot-spl-dtb.hex files to \$BIN_DIR for the next subsection.

6.3 Injecting U-Boot into the FPGA .sof (RBF/JIC)

To inject U-Boot into the .sof, the u-boot-spl-dtb.hex generated from the prior subsection is required. Once the yocto build process completes, the following script can be run as follows:

```
$ cd $YOCTO_SRC_DIR  
./fpga.sh
```

A directory will be generated in \$BIN_DIR with a timestamp that includes the source .sof/.hex and generated .jic/.rbf pair.

6.4 Insert .rbf into SD Card Image

The SD Card image file (.wic) can be updated by mounting the image file in the Ubuntu host machine. The following procedure will be used to include the .rbf into the SD Card image.

```
# Create a mount point
sudo mkdir /mnt/sdcard

# Generate a loop device
sudo partx -a -v $BIN_DIR/console-image-minimal-agilex.wic
```

Figure 6-1. Loop Device Creation

```
Trying to use '/dev/loop8' for the loop device
/dev/loop8: partition table type 'dos' detected
range recount: max partno=2, lower=0, upper=0
/dev/loop8: partition #1 added
/dev/loop8: partition #2 added
```

A device labeled loopX will be created where X will be a number (e.g. /dev/loop0). Assuming the device is loop0, mount the loop device's first partition which will be the FAT partition used by U-Boot:

```
# Mount the FAT partition of the SD Card image
sudo mount /dev/loop0p1 /mnt/sdcard

# Transfer the generated .rbf:
sudo cp $BIN_DIR/ag_esom_top.core.rbf /mnt/sdcard

# Unmount the loop device:
sudo umount /dev/loop0p1

# Remove the loop device:
sudo partx -d -v /dev/loop0
```

7.0 Loading the SD Card Image

The process below will assist with loading the image onto a SD card in a Linux environment.

1. Insert the SD card into the Linux host machine using an SD Card adapter.
2. Run the following command to see the last connected device:

```
$ sudo dmesg | tail | grep sd
```

Attention: DATA LOSS OF HOST MACHINE CAN OCCUR IN THE NEXT STEP IF IMPROPER DEVICE IS SELECTED

3. Use the **dd** utility to write the SD card image to the SD card using the last connected device. The device name should be in the format of sdx where "x" will vary:

```
$ sudo dd if=$BIN_DIR/console-image-minimal-agilex.wic of=/dev/<DEVNAME> bs=1M
```

4. Use the **sync** utility to flush the changes to the SD card:

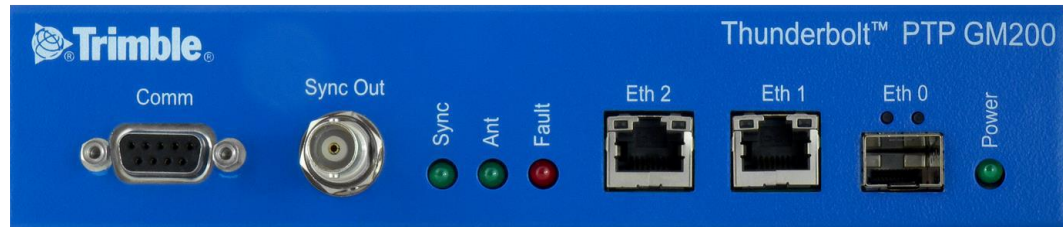
```
$ sudo sync
```

8.0 O-RU Bring Up

This next chapter will go over communicating with the devkit, adjusting U-Boot, programming the Si5518 via HPS, and initializing the O-RU.

8.1 Trimble GM200 Configuration

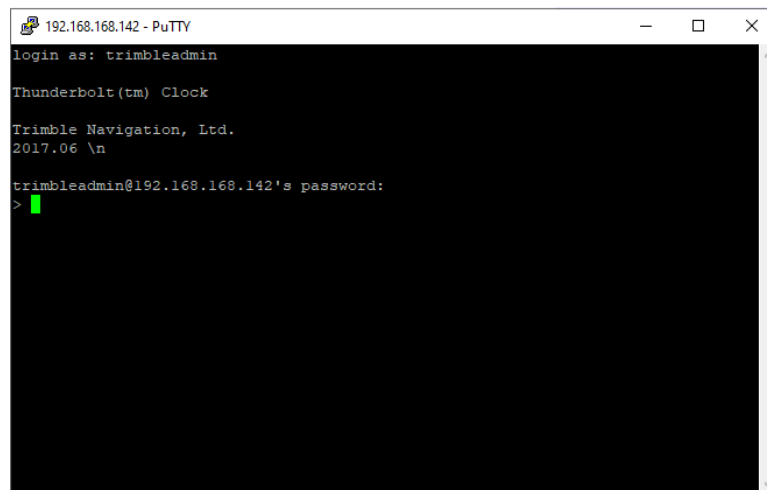
Figure 8-1. Trimble GM200 Front Panel



The Trimble GM200 can receive a GNSS signal from a GPS antenna via the back panel SMA connector. Grandmaster clocks receive UTC-based time information from a GNSS satellite source.

Log in to the GM200 command line interface (CLI) in PuTTY, with the IP assigned to the eth2 network port (Management port). Refer to the below figure for reference.

Figure 8-2. Trimble GM200 Login Prompt



Execute the following commands in the terminal, to set the PTP configuration:

```
> set ptp eth0 disable
> set ptp eth0 mode master profile G8275 domain 24 sm 2 transport Eth
> set ptp eth0 enable
```

Altera Confidential

8.0 O-RU Bring Up

For more information, refer to the Trimble GM200 User Manual.

8.2 Cisco Nexus 93180YC-FX3 Switch Configuration

The Cisco Nexus 93180YC-FX3 switch is capable of being a T-BC in a 1588 aware network. It is also capable of running the G.8275.1 ITU profile. For quick reference, the ports being configured belong to the following device mapping:

Switchport - Device

Eth1/40 - Trimble GM200

Eth1/46 - Keysight S5040A (ORAN Studio)

Eth1/44 - Hitek MXL Carrier + eSOM (SFP28)

Login to the switch using your set user ID and password. For additional details, please refer to the Cisco Nexus 9000 documentation:

<https://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/products-installation-and-configuration-guides-list.html>

- System Management Configuration Guide
 - Configuring Frequency Synchronization (SyncE)
 - Configuring PTP
 - Configuring PTP Telecom Profile

8.2.1 Link/VLAN Configuration

Execute the below commands:

```
# config terminal
# int Eth1/40
# fec auto
# int Eth1/46
# fec rs-ieee
# mtu 9216
# switchport mode trunk
# int Eth1/44
# fec rs-ieee
# mtu 9216
# switchport mode trunk
# exit
# configure terminal
# vlan 2
# state active
# no shutdown
# copy running-config startup-config
# exit
```

8.2.2 G.8275.1 PTP Configuration

Execute the below commands to enable PTP globally:

Altera Confidential

```
# conf t
# feature ptp
# ptp profile 8275-1
# mode hybrid
# exit
# exit
```

Execute the below commands to enable PTP on each interface:

```
# int Eth1/40
# ptp
# int Eth1/46
# ptp
# int Eth1/44
# ptp
# exit
# copy running-config startup-config
# exit
```

Note: This section may require changing the settings of QoS TCAM region depending on the NXOS version. This section assumes that this has been completed.

8.2.3 SyncE Configuration

Cisco requires that SyncE is enabled when the G.8275.1 profile is used as mentioned by the "hybrid" mode. Execute the below commands to enable SyncE sources and to use the Trimble GM200 (Eth1/40) as the primary source:

Enable SyncE globally:

```
# conf t
# feature frequency-synchronization
# exit
```

Enable SyncE on interfaces:

```
# int Eth1/40
# frequency synchronization
# selection input
# quality receive exact itu-t option 1 PRC
# wait-to-restore 0
# exit
# int Eth1/46
# freq sync
# wait-to-restore 0
# exit
# int Eth1/44
# freq sync
# wait-to-restore 0
# exit
# copy running-config startup-config
# exit
```

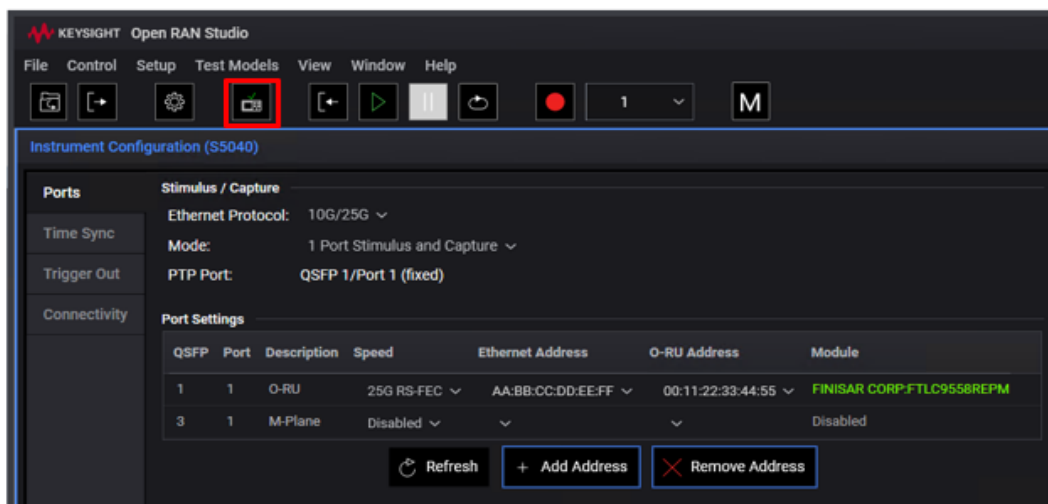
8.3 Keysight S5040A (ORAN Studio) Configuration

The next steps will configure the Keysight S5040A to send and receive to the proper MAC Addresses as well as enable PTP as a T-TSC device for LLS-C3 configuration.

1. Open Keysight ORAN Studio and click on the instrument configuration button as shown in the below figure. The **Instrument Configuration (S5040A)** window will open.

8.0 O-RU Bring Up

2. In the opened window, select the **Ports** tab and click the **Add Address** button to add Ethernet and O-RU addresses.
3. Create two new addresses by inputting the following MAC addresses and then clicking OK:
 - 00:11:22:33:44:55
 - AA:BB:CC:DD:EE:FF
4. In **Ethernet Address** and **O-RU Address** dropdown select the respective addresses as shown in the below figure for QSFP 1 Port 1.

Figure 8-3. ORAN Studio - Port Settings

5. Click on the Time Sync tab and match the settings as shown in the below figure

Figure 8-4. ORAN Studio - Time Sync Configuration

Ports	Clock Settings	
	Reference Clock:	Internal ▼
Time Sync	PTP	
Trigger Out	Mode:	Slave ▼
Connectivity	Multicast:	Forwardable ▼
	Domain:	24
	Priority2:	128
	UTC Offset:	37 seconds
	Master Link Up Delay:	0 seconds
	Master Link Down Delay:	0 seconds
	System Frame Number	
	Beta:	0
	Stimulus	
	<input type="checkbox"/> Sync on PPS-In	Sync to Frame 0 ▼
	<input type="checkbox"/> Play Stimulus on Trigger-In	
	<input type="checkbox"/> Check to disable SyncE	

At this stage, the Keysight S5040A should be converging over PTP to the T-BC and is ready to send ORAN traffic over the link if in LLS-C3 configuration.

8.4 Programming the FPGA Flash (JIC)

This section will go over programming the **.jic** file produced by **Section 7.4 Injecting Uboot into the FPGA image (RBF/JIC)**. This will ensure from a cold boot that the board will boot into U-Boot/HPS.

To determine the <board-JTAG-index> number to use:

```
$ jtagconfig -d
```

The JTAG interface will report as "HTK USBII"

To program the JIC:

```
$ jtagconfig --setparam <board-JTAG-index> JtagClock 16M
$ jtagconfig --setparam <board-JTAG-index> JtagClockAutoAdjust 0
$ quartus_pgm -c <board-JTAG-index> -m jtag -o "pvi;ag_esom_top.hps.jic"
```

Example:

```
$ jtagconfig --setparam 1 JtagClock 16M
$ jtagconfig --setparam 1 JtagClockAutoAdjust 0
$ quartus_pgm -c 1 -m jtag -o "pvi;$BIN_DIR/ag_esom_top.hps.jic"
```

8.0 O-RU Bring Up

This is a one-time process and is only necessary to perform again if the Quartus project's I/O ring (e.g. pin assignments) and/or U-Boot .hex (e.g. U-Boot version) is updated.

Note:

AFTER PROGRAMMING OF .JIC, POWER CYCLE BOARD

8.5 Serial Configuration

The minicom utility can be used to communicate with the devkit. To determine the device name on the Host PC, connect the mini-USB cable from J7 to the Host machine. Then perform the following command:

```
$ sudo dmesg | tail | grep tty
```

This will output the most recent tty connected device. Typically, the device will be /dev/ttyACM0.

To configure minicom:

```
$ sudo minicom -s
```

In Serial Port Setup choose the following:

- **Serial Device:** /dev/ttyACM0 (edit to match the system as necessary)
- **Bps/Par/Bits:** 115200 8N1
- **Hardware Flow Control:** No
- **Software Flow Control:** No

Press **[ESC]** to return to the main configuration menu.

Select **Save Setup as dfl** to save the default setup. Then select **Exit**.

To enter the serial connection minicom give the following command:

```
$ sudo minicom
```

To run minicom without needing sudo access, the user account must be added to the **dialout** group:

```
sudo usermod -aG dialout USERNAME
```

8.6 Resizing the Root Filesystem

During the Yocto process, the **meta-intel-fpga/wic/sdimage-stratix10-agilex.wks** file had been modified to generate a reduced SD Card image size from 2GB to 600MB. This was done by reducing the FAT partition size to 100MB and the root filesystem to 500MB.

The root filesystem can be expanded in runtime by using the following process:

1. Run the resize.sh script:

Altera Confidential

```
~/misc/resize.sh
```

2. Power cycle the devkit
3. Run the resize.sh script again after logging in:

```
~/misc/resize.sh
```

Note: This is a one-time process and is persistent across power cycles.

8.7 cAD9528 Clock Configuration

Follow the steps below to configure the AD9528's clocks:

1. After the initial booting process, login as root.
2. Go to the **koror** directory and run the **run.sh** script:

```
$ cd koror
$ ./run.sh
```

3. Reboot the board

```
$ reboot
```

Figure 8-5. Configuring AD9528 Clocks

```
agilex login: root
Last login: Fri May  9 13:03:31 +0000 2025 on pts/2 from 192.168.168.4.
root@agilex:~# [ 10.857592] socfpga-dwmac ff802000.ethernet eth1: Link is Up - 1Gbps/Full - flow control rx/tx
[ 10.866632] IPv6: ADDRCONF (NETDEV_CHANGE): eth1: link becomes ready
cd koror/
root@agilex:~/koror# ./run.sh
Transceiver API Version: 2.0.0 (build 11)
===== Completed Koror AD9528 Programming =====
root@agilex:~/koror# reboot
```

8.8 IP Configuration

The ethernet ports can be configured with IPs. Below are the assigned Linux interfaces:

eth1: Ethernet RJ-45 HPS MGMT Port

eth0: SFP28 Interface (J9)

To set the IP (e.g. for SSH access on eth1), modify the ~/ethsetup/sethpsip.sh script with the proper IP and run:

```
$ ~/ethsetup/sethpsip.sh
```

IP settings will be lost on reboot.

8.0 O-RU Bring Up

8.9 Source MAC Address Configuration

The source MAC address of the SFP28 interface must be reconfigured. To reconfigure, perform the following command:

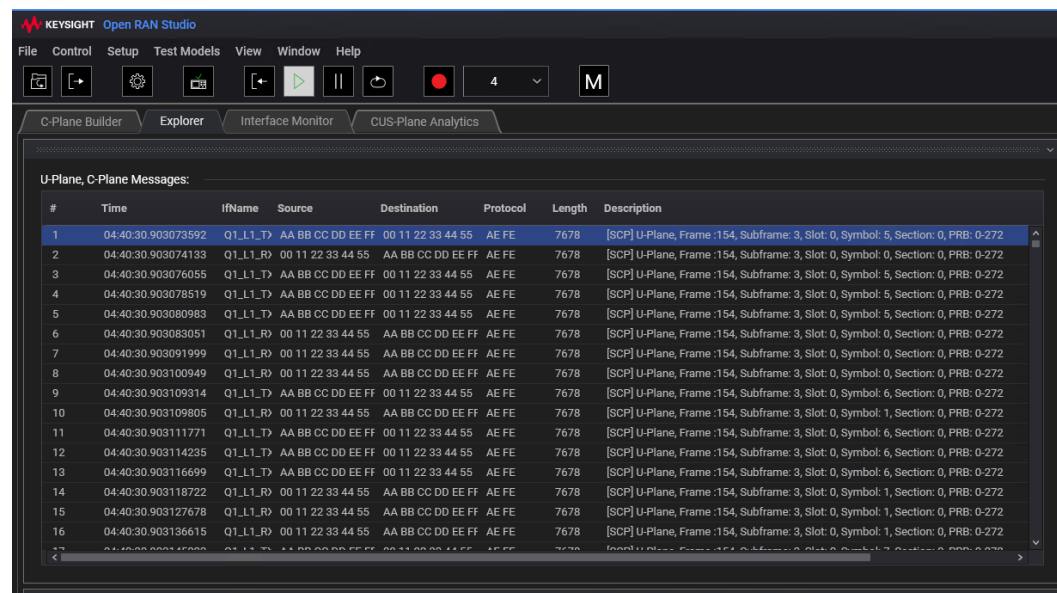
```
~/ethsetup/setsfpmac.sh
```

The assigned MAC address will be reconfigured and appear in ORAN Studio as shown in the below figures.

Figure 8-6. Configuring Source MAC Address

```
root@agilex:~# ethsetup/setsfpmac.sh
[67324.681599] intel_fpga_eth soc:hssi_0_eth eth0: Link is Down
[67326.796905] intel_fpga_eth soc:hssi_0_eth eth0: Device MAC address 00:11:22:33:44:55
[67328.817622] intel_fpga_eth soc:hssi_0_eth eth0: msgdma_pref_initialize: RX Desc mem at 0x7aac0000
[67328.826557] intel_fpga_eth soc:hssi_0_eth eth0: msgdma_pref_initialize: TX Desc mem at 0x7aad0000
[67328.835929] intel_fpga_eth soc:hssi_0_eth eth0: msgdma_pref_initialize: RX Desc mem at 0x7aae0000
[67328.844876] intel_fpga_eth soc:hssi_0_eth eth0: msgdma_pref_initialize: TX Desc mem at 0x7aaf0000
[67328.854345] intel_fpga_eth soc:hssi_0_eth eth0: msgdma_pref_initialize: RX Desc mem at 0x7ab00000
[67328.863218] intel_fpga_eth soc:hssi_0_eth eth0: msgdma_pref_initialize: TX Desc mem at 0x7ab10000
[67328.872112] intel_fpga_eth soc:hssi_0_eth eth0: device MAC address 00:11:22:33:44:55
[67328.883621] intel_fpga_eth soc:hssi_0_eth eth0: Device MAC address 00:11:22:33:44:55
[67328.891405] intel_fpga_eth soc:hssi_0_eth eth0: Ethernet link monitoring thread started
root@agilex:~# [67334.366226] intel_fpga_eth soc:hssi_0_eth eth0: F-tile rx_flow_ctrl: 0x00000003
[67334.373795] intel_fpga_eth soc:hssi_0_eth eth0: F-tile tx_flow_ctrl: 0x00000003
[67334.381259] intel_fpga_eth soc:hssi_0_eth eth0: F-tile: pause_quanta0: 0x0000ffff
[67334.388748] intel_fpga_eth soc:hssi_0_eth eth0: DBG: eth_ftile_tx_rx_user_flow speed=25000 num_vl=1 num_fl=1 num_1
[67334.400031] intel_fpga_eth soc:hssi_0_eth eth0: DBG: eth_ftile_tx_rx_user_flow ETH_TX_PTP_READY - tx_ref_pl:0 tx_7
[67334.416579] intel_fpga_eth soc:hssi_0_eth eth0: DBG: eth_ftile_tx_rx_user_flow ETH_RX_PTP_READY - rx_ref_pl:0 rx_1
[67334.430514] intel_fpga_eth soc:hssi_0_eth eth0: Link is Up - 25Gbps/Full - flow control rx/tx
[67334.439103] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
root@agilex:~#
```

Figure 8-7. ORAN Studio - Configured Source MAC Address



The screenshot shows the ORAN Studio interface with the 'Interface Monitor' tab selected. The 'U-Plane, C-Plane Messages' table displays network traffic data.

#	Time	IfName	Source	Destination	Protocol	Length	Description
1	04:40:30.903073592	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 5, Section: 0, PRB: 0-272
2	04:40:30.903074133	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 0, Section: 0, PRB: 0-272
3	04:40:30.903076055	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 5, Section: 0, PRB: 0-272
4	04:40:30.903078519	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 5, Section: 0, PRB: 0-272
5	04:40:30.903080983	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 5, Section: 0, PRB: 0-272
6	04:40:30.903083051	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 0, Section: 0, PRB: 0-272
7	04:40:30.903091999	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 0, Section: 0, PRB: 0-272
8	04:40:30.903100949	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 0, Section: 0, PRB: 0-272
9	04:40:30.903109314	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 6, Section: 0, PRB: 0-272
10	04:40:30.903109805	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 1, Section: 0, PRB: 0-272
11	04:40:30.903111771	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 6, Section: 0, PRB: 0-272
12	04:40:30.903114235	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 6, Section: 0, PRB: 0-272
13	04:40:30.903116699	Q1_L1_T	AA BB CC DD EE FF	00 11 22 33 44 55	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 6, Section: 0, PRB: 0-272
14	04:40:30.903118722	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 1, Section: 0, PRB: 0-272
15	04:40:30.903127678	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 1, Section: 0, PRB: 0-272
16	04:40:30.903136615	Q1_L1_R	00 11 22 33 44 55	AA BB CC DD EE FF	AE FE	7678	[SCP] U-Plane, Frame :154, Subframe: 3, Slot: 0, Symbol: 1, Section: 0, PRB: 0-272

Altera Confidential

8.10 Running ptp4l

In this section, ptp4l will run in a separate SSH connection to the Hitek devkit instead of via the serial connection.

Set the IP address of the Hitek board and connect to the board via SSH (e.g. PuTTY)

Then, log in and run the **runptp.sh** script:

```
$ ~/ptp/runptp.sh
```

For more details, refer to the contents of the above script(s).

Do not close the SSH connection.

Figure 8-8. AGX CLI - ptp4l Converged Output

```
ptp4l[769.369]: master offset      3 s2 freq      +3 path delay      313
ptp4l[769.432]: master offset      0 s2 freq      +1 path delay      318
ptp4l[769.495]: master offset      5 s2 freq      +4 path delay      311
ptp4l[769.558]: master offset     -6 s2 freq      -3 path delay      320
ptp4l[769.621]: master offset      6 s2 freq      +5 path delay      306
ptp4l[769.683]: master offset     -3 s2 freq      -1 path delay      321
ptp4l[769.740]: master offset      3 s2 freq      +3 path delay      313
ptp4l[769.808]: master offset     -7 s2 freq      -4 path delay      318
ptp4l[769.871]: master offset     -9 s2 freq      -5 path delay      321
ptp4l[769.934]: master offset     -1 s2 freq      -0 path delay      315
ptp4l[769.997]: master offset     -1 s2 freq      -0 path delay      315
ptp4l[770.060]: master offset     -2 s2 freq      -1 path delay      315
ptp4l[770.123]: master offset     -8 s2 freq      -5 path delay      315
ptp4l[770.186]: master offset      1 s2 freq      +1 path delay      309
ptp4l[770.249]: master offset      5 s2 freq      +4 path delay      314
ptp4l[770.312]: master offset     -5 s2 freq      -3 path delay      322
ptp4l[770.375]: master offset      3 s2 freq      +3 path delay      306
ptp4l[770.437]: master offset      1 s2 freq      +1 path delay      306
```

8.11 Loading the O-RU Drivers & User Application

In this section the O-RU driver will be loaded, the ToD reset via the user application, and communication will be established to connect with the MATLAB GUI.

Open a new SSH connection, log in, and perform the following:

```
$ ~/o_ru/o_ru_driver_load.sh
```

Figure 8-9. AGX CLI – O-RU Driver Load (SSH)

```
root@agilex:~/o_ru# ./o_ru_driver_load.sh
Installing SYSID driver
Installing WRAPPER_TOP driver
Installing CAP_BUF driver
```

Now reset the ToD using the O-RU user application:

Altera Confidential

```
$ ~/o_ru/tod.sh
```

Figure 8-10. AGX CLI - ToD Reset via O-RU User App (SSH)

```
root@agilex:~# ~/o_ru/tod.sh
alpha --> 0
beta --> 0
Frame resetting
Frame reset doneroot@agilex:~#
```

Finally, edit the `~/o_ru/server.sh` script and input the MATLAB host machine's IP and Port. Then, start the MATLAB connection by running the script:

```
$ ~/o_ru/server.sh
```

Figure 8-11. AGX CLI - MATLAB Connection

```
root@agilex:~# ~/o_ru/server.sh
main : 357 Setting server IP address
Waits for Server connection
main : 363 Server address --> 192.168.4.3
main : 335 Setting connection port for 192.168.4.3
main : 342 Port address --> 50001
FIFO monitor thread created
Server acquired for creating 0 IP 192.168.4.3 Port 50001
Thread Run in server 1
Result command thread priority 0
Server waiting 0 IP 192.168.4.3 Port 50001
Server connected 0 IP 192.168.4.3 Port 50001
Server acquired for reading 0 IP 192.168.4.3 Port 50001
Command received
Command received
Command received
```

At this stage, the Hitek devkit will be running `ptp4l` and retrying a connection to the MATLAB GUI which will be started in the next section. The above figure displays a successful connection.

Do not close the SSH connections.

8.12 Koror Configuration

Koror Configuration is done after selecting loopback points in matlab GUI. Koror configuration is only applicable to digital and RF loopback.

```
$ cd koror
```

For digital loopback, use the command below

Altera Confidential

```
$ ./config_koror --config --dig_lpbk
```

For RF loopback, use the command below

```
$ ./config_koror --config --status
```

Figure 8-12. AGX CLI - Koror Configuration Output for Digital Loopback

```
root@agilex:~# cd koror/
root@agilex:~/koror# ./config_koror --config --dig_lpbk

hps_i2c_rw /dev/i2c-0 0x6 16 32 0x8 0x1
hps_i2c_rw /dev/i2c-0 0x6 16 32 0xC 0xA5A5EFEF

PhaselPrepCfgFiles()

      Firmware Binary File:      ./resources/ADRV9040_FW.bin
      Device Profile Binary File: ./resources/DeviceProfileTest.bin
      Stream Processor Binary File: ./resources/stream_image.bin
      Rx Gain Table CSV File:     ./resources/RxGainTable.csv
      DFE Binary File:           ./resources/ADRV9040_DFE_CALS_FW.bin
      Radio Sequencer Binary File:
```

8.13 Test Case Configuration

In this section, MATLAB will interface with the Hitek devkit over a LAN connection. The GUI will be used to configure the registers in the FPGA, interpret tx/rx ORAN traffic in the design, and more.

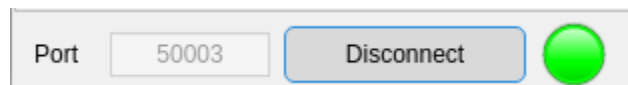
Please ensure **MATLAB 2023b** or later is installed and directories **\$REL_DIR/fpga** and **\$REL_DIR/testcase** are available.

1. In MATLAB, set the working directory to **\$REL_DIR/software/matlab_gui**
2. Run the following command in the command window to open the GUI:

```
> O_RU_design
```

3. Enter the Port number and click the **Connect to Soc** button to connect MATLAB to the Hitek devkit.

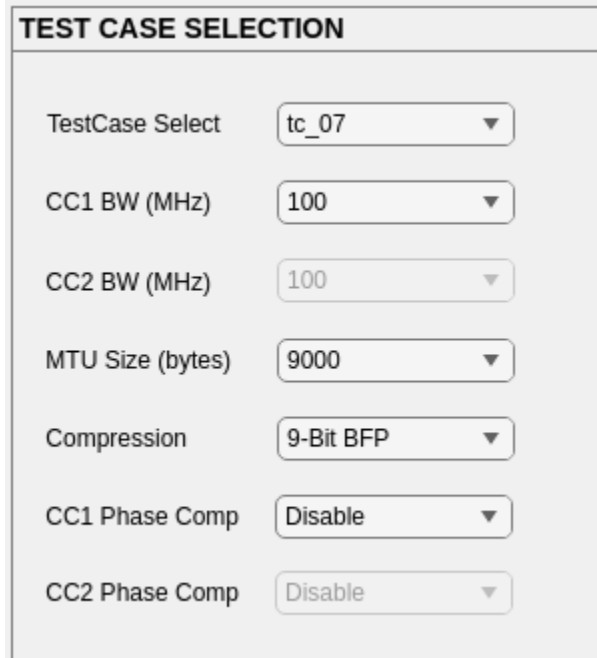
Figure 8-13. MATLAB GUI - Connection Established with Hitek Devkit



4. Go to the **eCPRI_ORAN_ss** tab.

8.0 O-RU Bring Up

- Click the dropdown box next to **TestCase Select** and select the proper test case. This will automatically update the corresponding CC1/CC2 BW, MTU size, and Compression dropdown fields.

Figure 8-14. MATLAB GUI - Test Case Selection

TEST CASE SELECTION	
TestCase Select	tc_07 ▼
CC1 BW (MHz)	100 ▼
CC2 BW (MHz)	100 ▼
MTU Size (bytes)	9000 ▼
Compression	9-Bit BFP ▼
CC1 Phase Comp	Disable ▼
CC2 Phase Comp	Disable ▼

- Click the **Configure** button. This will configure the eCPRI, ORAN, FH-Compression, and EAXC-ID registers as shown in the below figure:

Figure 8-15. MATLAB GUI - eCPRI & ORAN Register Configuration

O_RU_Setup	eCPRI_ORAN_ss	DXC_ss	Low_phy_ss	Power_meter	O_RU_Diagnostic	O_RU_Control																																																																		
<div> <div> TEST CASE SELECTION <p>Test Case Select: <input type="text" value="tc_07"/></p> <p>CC1 BW (MHz): <input type="text" value="100"/></p> <p>CC2 BW (MHz): <input type="text" value="100"/></p> <p>MTU Size (bytes): <input type="text" value="9000"/></p> <p>Compression: <input type="text" value="9-Bit BFP"/></p> <p>CC1 Phase Comp: <input type="text" value="Disable"/></p> <p>CC2 Phase Comp: <input type="text" value="Disable"/></p> </div> <div> eCPRI Configs <p>SRC Address(in HEX): <input type="text" value="001122334455"/></p> <p>DEST Address(in HEX): <input type="text" value="aabbccddeeff"/></p> <p>VLAN Enable: <input type="text" value="1"/></p> <p>VLAN tag VID: <input type="text" value="2"/></p> <p>VLAN tag PCP: <input type="text" value="7"/></p> <p>DFI MATCH: <input type="text" value="0"/></p> <p>DFI VID addr: <input type="text" value="2"/></p> </div> </div>																																																																								
<div> <div> ORAN Configs <p>t2a_max_cp_dl: <input type="text" value="457000"/></p> <p>t2a_min_cp_dl: <input type="text" value="150000"/></p> <p>t2a_max_cp_ul: <input type="text" value="457000"/></p> <p>t2a_min_cp_ul: <input type="text" value="150000"/></p> <p>t2a_max_up: <input type="text" value="206000"/></p> <p>t2a_min_up: <input type="text" value="80000"/></p> <p>ta3_max_up: <input type="text" value="150000"/></p> <p>ta3_min_up: <input type="text" value="40000"/></p> <p>ta3_max_up_1p25khz: <input type="text" value="1500000"/></p> <p>ta3_min_up_1p25khz: <input type="text" value="40000"/></p> <p>window offset: <input type="text" value="0"/></p> <p>rx_window_enable: <input type="text" value="1"/></p> <p>tx_window_enable: <input type="text" value="1"/></p> <p>Functional mode: <input type="text" value="0"/></p> <p>Comp method: <input type="text" value="1"/></p> <p>Comp Bitwidth: <input type="text" value="9"/></p> <p>Fragment Size: <input type="text" value="273"/></p> </div> <div> Eaxc ID Configs <table border="1"> <thead> <tr> <th colspan="2">DL eAxC_ID</th> <th colspan="2">CC1</th> <th colspan="2">CC2</th> </tr> </thead> <tbody> <tr><td>TRX-1</td><td><input type="text" value="0"/></td><td>TRX-1</td><td><input type="text" value="4"/></td></tr> <tr><td>TRX-2</td><td><input type="text" value="1"/></td><td>TRX-2</td><td><input type="text" value="5"/></td></tr> <tr><td>TRX-3</td><td><input type="text" value="2"/></td><td>TRX-3</td><td><input type="text" value="6"/></td></tr> <tr><td>TRX-4</td><td><input type="text" value="3"/></td><td>TRX-4</td><td><input type="text" value="7"/></td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">UL eAxC_ID</th> <th colspan="2">CC1</th> <th colspan="2">CC2</th> </tr> </thead> <tbody> <tr><td>TRX-1</td><td><input type="text" value="0"/></td><td>TRX-1</td><td><input type="text" value="4"/></td></tr> <tr><td>TRX-2</td><td><input type="text" value="1"/></td><td>TRX-2</td><td><input type="text" value="5"/></td></tr> <tr><td>TRX-3</td><td><input type="text" value="2"/></td><td>TRX-3</td><td><input type="text" value="6"/></td></tr> <tr><td>TRX-4</td><td><input type="text" value="3"/></td><td>TRX-4</td><td><input type="text" value="7"/></td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">PRACH eAxC_ID</th> <th colspan="2">CC1</th> <th colspan="2">CC2</th> </tr> </thead> <tbody> <tr><td>TRX-1</td><td><input type="text" value="8"/></td><td>TRX-1</td><td><input type="text" value="12"/></td></tr> <tr><td>TRX-2</td><td><input type="text" value="9"/></td><td>TRX-2</td><td><input type="text" value="13"/></td></tr> <tr><td>TRX-3</td><td><input type="text" value="10"/></td><td>TRX-3</td><td><input type="text" value="14"/></td></tr> <tr><td>TRX-4</td><td><input type="text" value="11"/></td><td>TRX-4</td><td><input type="text" value="15"/></td></tr> </tbody> </table> </div> </div>							DL eAxC_ID		CC1		CC2		TRX-1	<input type="text" value="0"/>	TRX-1	<input type="text" value="4"/>	TRX-2	<input type="text" value="1"/>	TRX-2	<input type="text" value="5"/>	TRX-3	<input type="text" value="2"/>	TRX-3	<input type="text" value="6"/>	TRX-4	<input type="text" value="3"/>	TRX-4	<input type="text" value="7"/>	UL eAxC_ID		CC1		CC2		TRX-1	<input type="text" value="0"/>	TRX-1	<input type="text" value="4"/>	TRX-2	<input type="text" value="1"/>	TRX-2	<input type="text" value="5"/>	TRX-3	<input type="text" value="2"/>	TRX-3	<input type="text" value="6"/>	TRX-4	<input type="text" value="3"/>	TRX-4	<input type="text" value="7"/>	PRACH eAxC_ID		CC1		CC2		TRX-1	<input type="text" value="8"/>	TRX-1	<input type="text" value="12"/>	TRX-2	<input type="text" value="9"/>	TRX-2	<input type="text" value="13"/>	TRX-3	<input type="text" value="10"/>	TRX-3	<input type="text" value="14"/>	TRX-4	<input type="text" value="11"/>	TRX-4	<input type="text" value="15"/>
DL eAxC_ID		CC1		CC2																																																																				
TRX-1	<input type="text" value="0"/>	TRX-1	<input type="text" value="4"/>																																																																					
TRX-2	<input type="text" value="1"/>	TRX-2	<input type="text" value="5"/>																																																																					
TRX-3	<input type="text" value="2"/>	TRX-3	<input type="text" value="6"/>																																																																					
TRX-4	<input type="text" value="3"/>	TRX-4	<input type="text" value="7"/>																																																																					
UL eAxC_ID		CC1		CC2																																																																				
TRX-1	<input type="text" value="0"/>	TRX-1	<input type="text" value="4"/>																																																																					
TRX-2	<input type="text" value="1"/>	TRX-2	<input type="text" value="5"/>																																																																					
TRX-3	<input type="text" value="2"/>	TRX-3	<input type="text" value="6"/>																																																																					
TRX-4	<input type="text" value="3"/>	TRX-4	<input type="text" value="7"/>																																																																					
PRACH eAxC_ID		CC1		CC2																																																																				
TRX-1	<input type="text" value="8"/>	TRX-1	<input type="text" value="12"/>																																																																					
TRX-2	<input type="text" value="9"/>	TRX-2	<input type="text" value="13"/>																																																																					
TRX-3	<input type="text" value="10"/>	TRX-3	<input type="text" value="14"/>																																																																					
TRX-4	<input type="text" value="11"/>	TRX-4	<input type="text" value="15"/>																																																																					
<div> <div> FH compr Configs <p>Compression mode: <input type="text" value="145"/></p> <p>decompr fsoffset: <input type="text" value="5"/></p> <p>compr fsoffset: <input type="text" value="5"/></p> </div> <div> <input type="button" value="Configure"/> <input type="button" value="Default"/> </div> </div>																																																																								

- Go to the **DXC_ss** tab
- Click **Configure** button. This will configure the DUC/DDC subsystem registers as shown in the below figure.

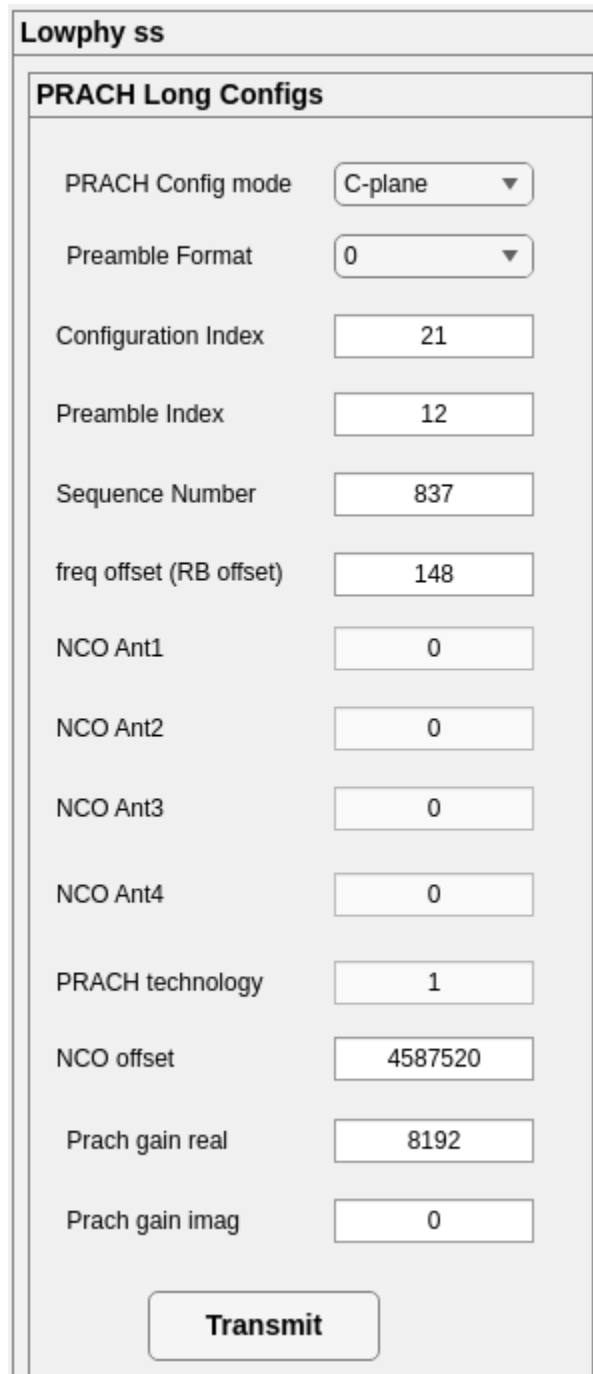
Figure 8-16. MATLAB GUI - DXC_ss Registers

O_RU_Setup	eCPRI_ORAN_ss	DXC_ss	Low_phy_ss	Power_meter	O_RU_Diagnostic	O_RU_Control
<div> <div> DXC SS Configs <input checked="" type="checkbox"/> <p>DDC mux sel L1: <input type="text" value="0"/></p> <p>DDC mux const L1 ant1 re: <input type="text" value="0"/></p> <p>DDC mux const L1 ant1 im: <input type="text" value="0"/></p> <p>DDC mux const L1 ant2 re: <input type="text" value="0"/></p> <p>DDC mux const L1 ant2 im: <input type="text" value="0"/></p> <p>DDC mux const L1 ant3 re: <input type="text" value="0"/></p> <p>DDC mux const L1 ant3 im: <input type="text" value="0"/></p> <p>DDC mux const L1 ant4 re: <input type="text" value="0"/></p> <p>DDC mux const L1 ant4 im: <input type="text" value="0"/></p> <p>DDC mux sel L2: <input type="text" value="0"/></p> <p>DDC mux const L2 ant1 re: <input type="text" value="0"/></p> <p>DDC mux const L2 ant1 im: <input type="text" value="0"/></p> <p>DDC mux const L2 ant2 re: <input type="text" value="0"/></p> <p>DDC mux const L2 ant2 im: <input type="text" value="0"/></p> <p>DDC mux const L2 ant3 re: <input type="text" value="0"/></p> <p>DDC mux const L2 ant3 im: <input type="text" value="0"/></p> <p>DDC mux const L2 ant4 re: <input type="text" value="0"/></p> <p>DDC mux const L2 ant4 im: <input type="text" value="0"/></p> </div> <div> DUC Configs <input checked="" type="checkbox"/> <p>DUC NCO 1 ant1: <input type="text" value="50"/></p> <p>DUC NCO 1 ant2: <input type="text" value="-50"/></p> <p>DUC NCO 1 ant3: <input type="text" value="50"/></p> <p>DUC NCO 1 ant4: <input type="text" value="-50"/></p> <p>DUC NCO 2 ant1: <input type="text" value="-50"/></p> <p>DUC NCO 2 ant2: <input type="text" value="50"/></p> <p>DUC NCO 2 ant3: <input type="text" value="-50"/></p> <p>DUC NCO 2 ant4: <input type="text" value="50"/></p> </div> <div> CA Interp Configs <input checked="" type="checkbox"/> <p>DUC ant1 gain: <input type="text" value="8192"/></p> <p>DUC ant2 gain: <input type="text" value="8192"/></p> <p>DUC ant3 gain: <input type="text" value="8192"/></p> <p>DUC ant4 gain: <input type="text" value="8192"/></p> <p>DUC 1 ant1 delay: <input type="text" value="0"/></p> <p>DUC 1 ant2 delay: <input type="text" value="0"/></p> <p>DUC 1 ant3 delay: <input type="text" value="0"/></p> <p>DUC 1 ant4 delay: <input type="text" value="0"/></p> <p>DUC 2 ant1 delay: <input type="text" value="0"/></p> <p>DUC 2 ant2 delay: <input type="text" value="0"/></p> <p>DUC 2 ant3 delay: <input type="text" value="0"/></p> <p>DUC 2 ant4 delay: <input type="text" value="0"/></p> </div> <div> dec dly comp Configs <input checked="" type="checkbox"/> <p>DDC ant1 gain: <input type="text" value="8192"/></p> <p>DDC ant2 gain: <input type="text" value="8192"/></p> <p>DDC ant3 gain: <input type="text" value="8192"/></p> <p>DDC ant4 gain: <input type="text" value="8192"/></p> <p>DDC 1 ant1 delay: <input type="text" value="0"/></p> <p>DDC 1 ant2 delay: <input type="text" value="0"/></p> <p>DDC 1 ant3 delay: <input type="text" value="0"/></p> <p>DDC 1 ant4 delay: <input type="text" value="0"/></p> <p>DDC 2 ant1 delay: <input type="text" value="0"/></p> <p>DDC 2 ant2 delay: <input type="text" value="0"/></p> <p>DDC 2 ant3 delay: <input type="text" value="0"/></p> <p>DDC 2 ant4 delay: <input type="text" value="0"/></p> </div> <div> DDC Configs <input checked="" type="checkbox"/> <p>DDC NCO 1 ant1: <input type="text" value="-50"/></p> <p>DDC NCO 1 ant2: <input type="text" value="50"/></p> <p>DDC NCO 1 ant3: <input type="text" value="-50"/></p> <p>DDC NCO 1 ant4: <input type="text" value="50"/></p> <p>DDC NCO 2 ant1: <input type="text" value="50"/></p> <p>DDC NCO 2 ant2: <input type="text" value="-50"/></p> <p>DDC NCO 2 ant3: <input type="text" value="50"/></p> <p>DDC NCO 2 ant4: <input type="text" value="-50"/></p> </div> </div>						
<div> <input type="button" value="Default"/> <input type="button" value="Configure"/> </div>						

Altera Confidential

8.0 O-RU Bring Up

9. Go to the **Low_phy_ss** tab.
10. Configure PRACH Registers by clicking **Transmit** as shown in the below figure.

Figure 8-17. MATLAB GUI - PRACH Register


The screenshot shows the 'Lowphy ss' tab in the MATLAB GUI. Under the 'PRACH Long Configs' section, there are several configuration parameters, each with a corresponding input field or dropdown menu. At the bottom of the configuration area is a 'Transmit' button.

Parameter	Value
PRACH Config mode	C-plane
Preamble Format	0
Configuration Index	21
Preamble Index	12
Sequence Number	837
freq offset (RB offset)	148
NCO Ant1	0
NCO Ant2	0
NCO Ant3	0
NCO Ant4	0
PRACH technology	1
NCO offset	4587520
Prach gain real	8192
Prach gain imag	0

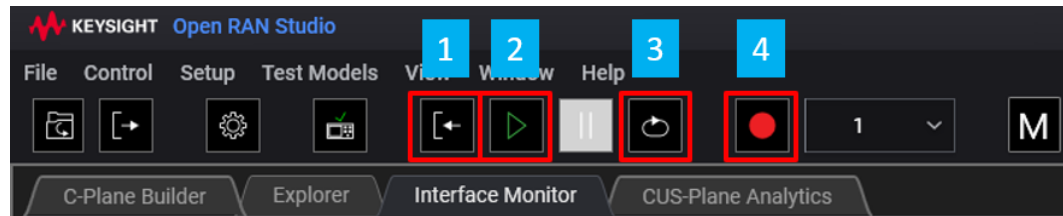
Transmit

Altera Confidential

At this stage, Keysight ORAN Studio will need to begin playback of an appropriate test case .pcap. The below list refers to the below figure's ORAN Studio icons. Hovering over the icons will also display their names:

- Load Stimulus (1)
- Play Stimulus (2)
- Single/Continuous Sweep (3)
- Record O-RAN Stimulus/Response between O-DU and O-RU (4)

Figure 8-18. ORAN Studio - Load and Play the PCAP Stimulus

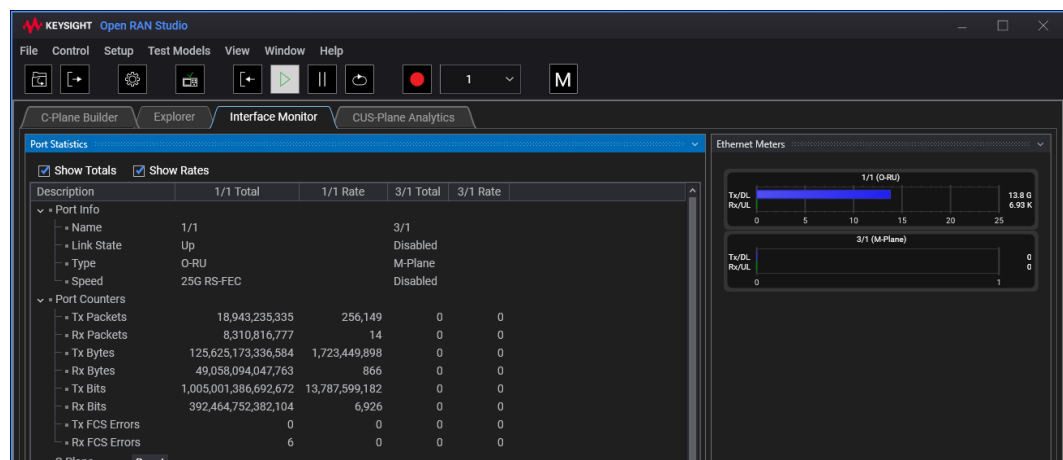


11. Go to Keysight ORAN Studio and click the Load Stimulus icon (1) to load the corresponding test case PCAP from the following directory:

\$REL_DIR/testvector/input/hardware/<TC_NAME>/<TC>.pcap

12. Ensure that the **Single/Continuous Sweep (3)** icon is set to Continuous Sweep as represented in the above figure.
13. Click on the **Play Stimulus (2)** icon to start transmission of ORAN packets from ORAN Studio to the Hitek Devkit. Tx/DL bandwidth will increase per the appropriate test case as shown in the below figure.

Figure 8-19. ORAN Studio - ORAN TX Packets (DL)



14. Go back to the MATLAB GUI
15. Go to the **Low_Phy_ss** tab

Altera Confidential

8.0 O-RU Bring Up

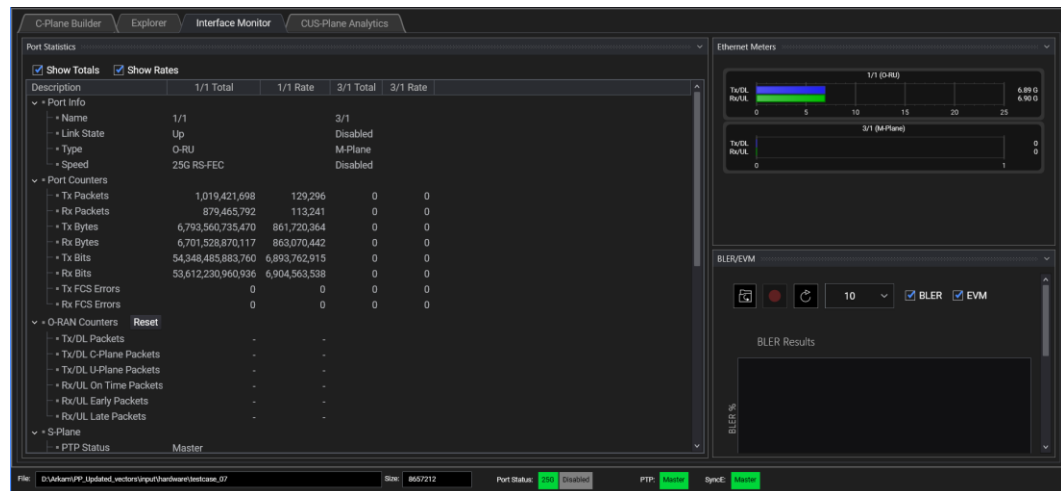
16. Change **DL start** and **UL start** to 1.
17. Click **Configure** to configure the IFFT, FFT, and Low phy subsystem registers as shown in the below figure.

Figure 8-20. MATLAB GUI - Radio Config Register Configuration

IFFT Configs <input checked="" type="checkbox"/>	FFT configs <input checked="" type="checkbox"/>	Low phy SS configs <input checked="" type="checkbox"/>
IFFT I1 gain <input type="text" value="16384"/>	FFT HCS bypass L1 <input type="text" value="1"/>	CC1 Disable <input type="text" value="0"/>
IFFT I2 gain <input type="text" value="16384"/>	FFT HCS bypass L2 <input type="text" value="1"/>	BW config (MHz) <input type="text" value="100"/>
IFFT I1 shift <input type="text" value="12"/>	FFT I1 gain <input type="text" value="16384"/>	CC2 Disable <input type="text" value="1"/>
IFFT I2 shift <input type="text" value="12"/>	FFT I2 gain <input type="text" value="16384"/>	BW config (MHz) <input type="text" value="100"/>
IFFT I1 mux const <input type="text" value="0"/>	FFT I2 gain <input type="text" value="16384"/>	DL Input Config:
IFFT I2 mux const <input type="text" value="0"/>	FFT I1 shift <input type="text" value="0"/>	DL start <input type="text" value="1"/>
IFFT I1 mux select <input type="text" value="0"/>	FFT I2 shift <input type="text" value="0"/>	DL config (in ns) <input type="text" value="-51858"/>
IFFT I2 mux select <input type="text" value="0"/>	DC_SC_I1_enable <input type="text" value="1"/>	sample <input type="text" value="9583"/>
DC_SC_I1_enable <input type="text" value="1"/>	DC_SC_I2_enable <input type="text" value="1"/>	symbol <input type="text" value="26"/>
DC_SC_I2_enable <input type="text" value="1"/>	FFT I1 F1 freq(in GHz) <input type="text" value="3.7497"/>	subframe <input type="text" value="9"/>
IFFT I1 F1 freq(in GHz) <input type="text" value="3.7497"/>	FFT I2 F1 freq(in GHz) <input type="text" value="3.7497"/>	radio frame <input type="text" value="1"/>
IFFT I2 F1 freq(in GHz) <input type="text" value="3.7497"/>	FFT I1 phase Enable <input type="text" value="0"/>	UL Input Config:
IFFT I1 phase Enable <input type="text" value="0"/>	FFT I2 phase Enable <input type="text" value="0"/>	UL start <input type="text" value="1"/>
IFFT I2 phase Enable <input type="text" value="0"/>	FFT I1 RC Enable <input type="text" value="1"/>	UL config (in ns) <input type="text" value="1735"/>
IFFT I1 RC Enable <input type="text" value="1"/>	FFT I2 RC Enable <input type="text" value="1"/>	sample <input type="text" value="853"/>
IFFT I2 RC Enable <input type="text" value="1"/>		symbol <input type="text" value="0"/>
		subframe <input type="text" value="0"/>
		radioframe <input type="text" value="0"/>
		<input type="checkbox"/> PRB Blanking Disable

18. After configuring the Low Phy SS configs registers, ORAN studio will start receiving data (Rx/UL) and RX counters will begin to increment.

Figure 8-21. ORAN Studio - Rx/UL Incrementing Counters



- Click the **Record O-RAN Stimulus/Response between O-DU and O-RU (4)** icon to capture the .pcap to use for data validation in reference to this release's VTR.

Note: For test case changes or changes in PRACH configurations, a power cycle of the devkit will **not** be required. Please refer to the VTR document for reconfiguration procedure.

Appendix A. Updating the Skyworks Si5518 Programming .bin Files

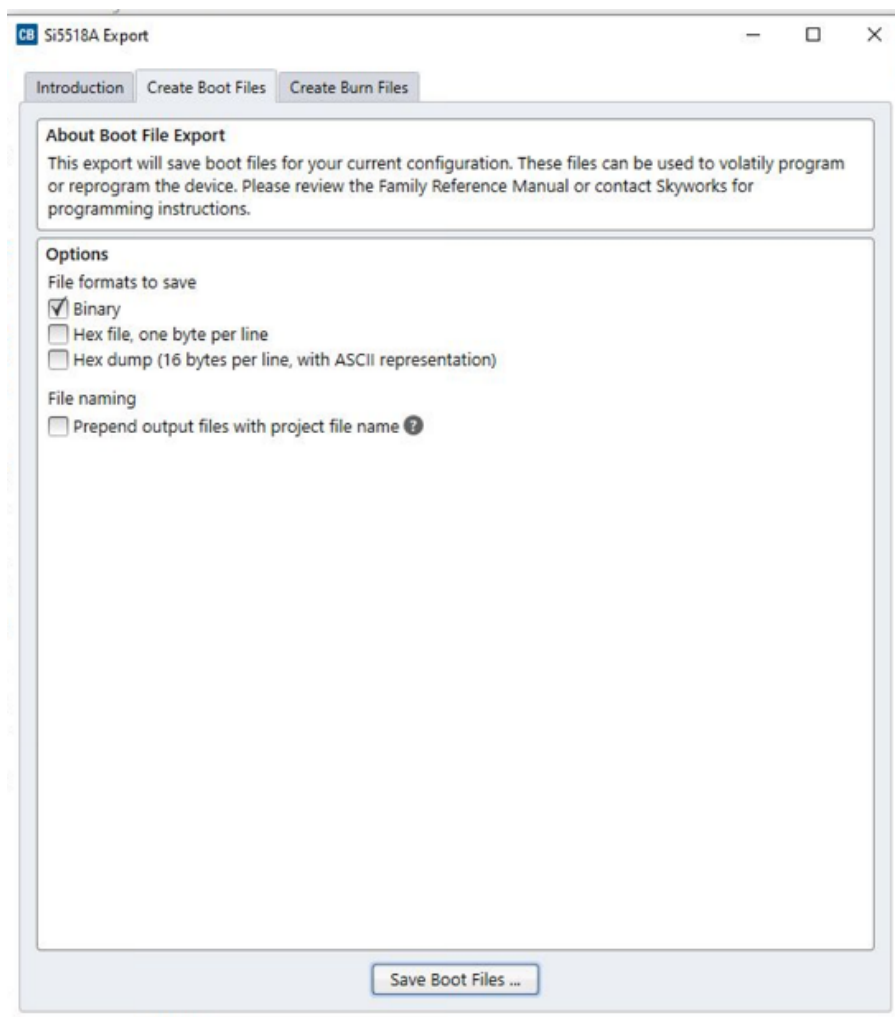
This section will go over generating a new set of binary files for programming the Si5518 and is provided as information only. This will need to be performed any time the ClockBuilder Pro project has been modified. **You do not need to generate the .bin files as this has been provided.**

1. Download the Skyworks ClockBuilder Pro software for Windows:

<https://www.skyworksinc.com/Application-Pages/Clockbuilder-Pro-Software>

2. Click **Export** on the **Project** page which will then open another window seen in the following figure.

Figure A-1 ClockBuilder Pro Export Page



Altera Confidential

3. Change to the **Create Boot Files** tab.
4. Checkmark **Binary**.
5. Click **Save Boot Files...** at the bottom of the window.

The next two subsections will describe two ways of programming the Si5518 over SPI: U-Boot and HPS. The default method of reprogramming is done via U-Boot to ensure the clocks are available before the FPGA fabric is configured. The HPS programming method is provided as a reference.

This will bring up the Hitek Si5518 Timing Module connected to the Hitek Carrier and enables the SFP28 link to be established in this design. This will also enable the linuxptp servo to frequency tune the Si5518 DCO with the required ppb steps.

A.1 Si5518 Programming via U-Boot

Please refer to the following u-boot script:

```
$REL_DIR/software/yocto/meta-intel-fpga-refdes/recipes-bsp/u-boot/files/uboot.txt
```

The parameter **clkprog** takes the combined generated prod_fw.boot.bin and user_config.boot.bin files (si5518.bin) and programs the Si5518 in chunks. To generate an updated si5518.bin file from an updated project, run the following command on an Ubuntu host machine:

```
$ cat prod_fw.boot.bin user_config.boot.bin > si5518.bin
```

Then, replace the **si5518.bin** file in the FAT partition of the SD Card. Once the board is power cycled, the **clkprog** parameter will run in U-Boot and configure the Si5518.

A.2 Si5518 Programming via HPS

The generated split **.bin** files are already included in this project.

To be able to stream the files using the SPI interface, they will need to be split into smaller chunks each prepended by 0xC005. This value represents the command for the Si5518 to initiate the load. The chunks are already provided and can be regenerated with the included script **\$REL_DIR/software/si5518/split_files.sh** and **\$REL_DIR/software/si5518/split_files.sh** on the Ubuntu host machine.

To run the script, move the generated .bin files to the **\$REL_DIR/software/si5518** directory and perform the following:

```
$ cd $REL_DIR/software/si5518
$ ./split_files.sh
```

The binary for the Skyworks board is loaded with **si5518_load.sh** which will send the binary files across the SPI interface. Below is the process on the Agilex HPS:

```
$ cd /home/root/si5518
$ ./si5518_load.sh /dev/spidev2.0
```

This will load the split files found under /home/root/si5518/split_files.

Then perform a soft reboot to trigger the FPGA fabric to reconfigure in U-Boot:

Altera Confidential


```
$ reboot
```

Note: For further information please refer to the Skyworks API documentation.

9.0 Revision History

Table 9-1. Document Revision History

Date	Version	Changes
2025-05-22	0.0.1	Gen III Release