SKYLARK WIRELESS®

# FAROS-HUB-FPGA

## Faros Hub FPGA Compilation Guide

## SUMMARY

Skylark Wireless's Faros® solution enables scalable software-defined Massive MIMO base stations by allowing one or more remote radio heads (RRHs) to be connected to a Faros Hub. As part of this solution, Skylark provides a software-defined radio (SDR) FPGA reference design for the hub. This document is a build guide for this Hub FPGA reference design.

The reference design is a Xilinx Vivado 2018.3 project that enables software-defined radio (SDR) streaming via SoapySDR. The design supports 6 RRHs and 1 reference Iris-030 each connected via two bonded Xilinx Aurora 6.25 Gbps links (12.5 Gbps), as well as two 10 GbE SFP+ backhaul links, and a 1 GbE BaseT management port. Skylark provides projects for both Xilinx XCZU6EG and XCZU9EG system on modules (SoMs). Proprietary IP cores are included as design checkpoints (".dcp", i.e., blackboxes). A reference root filesystem (rootfs) is included, or users can develop their own.

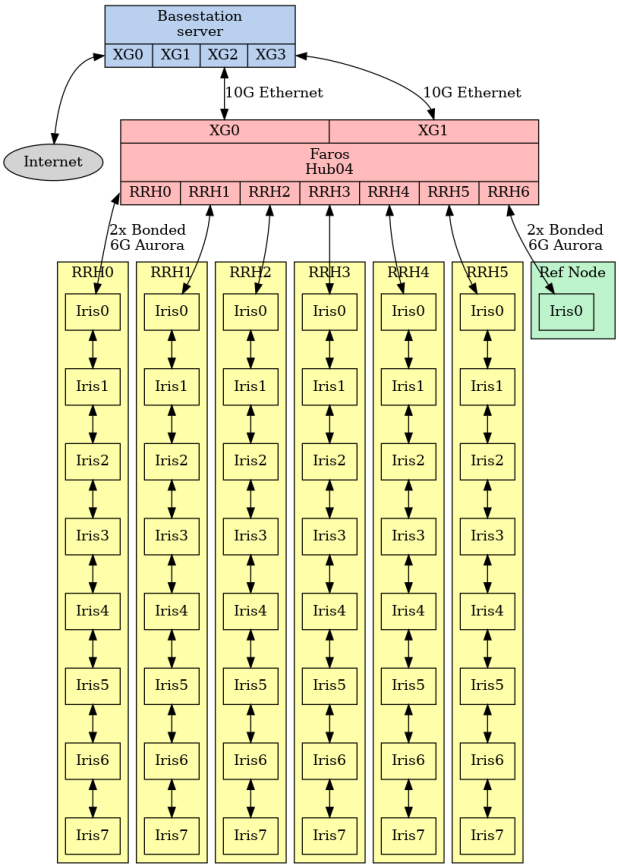


**Figure 1:** High-level Faros base station system overview, with the hub design covered by this document in red.

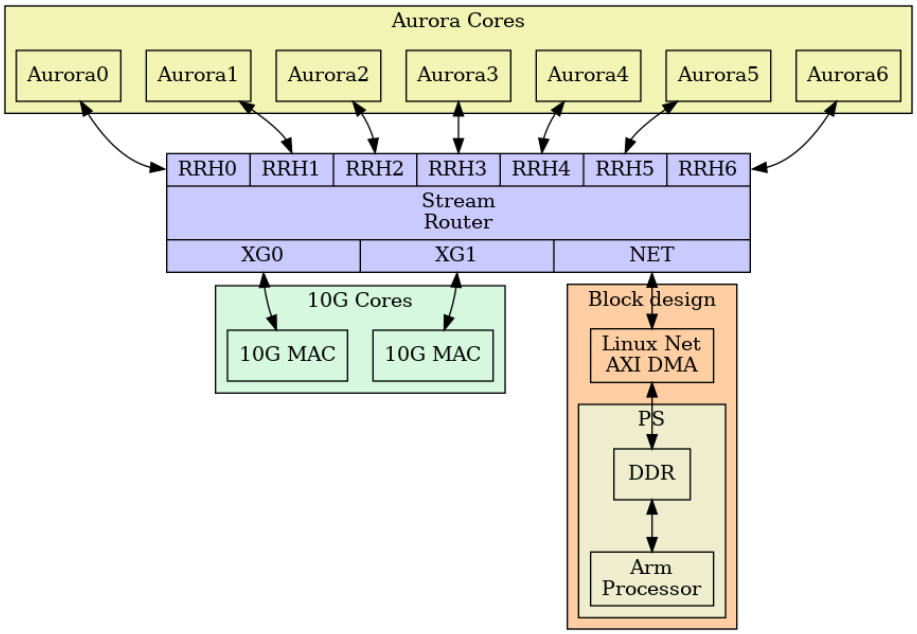


**Figure 2:** Faros HUB FPGA design block diagram.

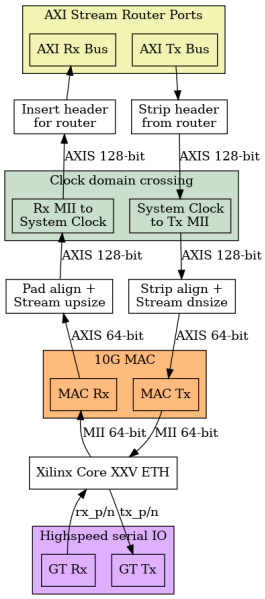


**Figure 3:** Block diagram of 10 GbE design.

# 1 DESIGN OVERVIEW

The top-level system diagram of the Faros Hub FPGA design is shown in Figure 2. The Faros Hub reference design provides seven bonded 6.25 Gbps (12.5 Gbps) Xilinx Aurora connections to chains of Irises; six of these are intended for RRHs, and the seventh as a reference node.

For backhaul, the design implements two 10 GbE Ethernet connections, both capable of providing providing a reference clock with Synchronous Ethernet (SyncE). When both are plugged in, XG0 is selected as the source by default. The 10 GbE Ethernet design is illustrated in Figure 3. Notably, for deterministic performance and simplicity, the stream router divides and isolates these Ethernet interfaces among the attached Iris chains. With one 10 GbE connection, it is split evenly among the RRHs. When both 10 GbE interfaces are connected, each are dedicated to half of the RRHs; for example, with six RRHs, each interface is dedicated to three RRHs.

The software interface for SoapySDR is provided with SoapyRemote:

`https://github.com/pothosware/SoapyRemote/wiki`

The HDL design is primarily documented as comments within the code.

# 2 BUILD INSTRUCTIONS

Building the Faros Hub SDR reference design is straightforward; the steps are:

1. Download the hub export design from `https://files.sklk.us/`
2. Extract the archive
3. Open Xilinx Vivado 2018.3, sourcing faroshub04_auto.tcl
4. Generate the bitstream in Vivado
5. Create the boot files with bootgen and provided rootfs

Download the correct export design for your FPGA SoM, e.g., somzu6eg_sdr for XCZU6EG or somzu9eg_sdr for XCZU9EG. The example commands to build in Linux are provided at the end of this document.

Never deploy a custom hub image to a remote site without testing it first. This could brick your remote hub.

# 3 LINUX BUILD EXAMPLE

Example Linux build commands (where MYVARIANT is the archive for your FPGA, e.g., somzu6eg_sdr for XCZU6EG or somzu9eg_sdr for XCZU9EG, and * is the version):

```
#variant, e.g., somzu6eg_sdr or somzu9eg_sdr
MYVARIANT=somzu6eg_sdr

#make a build directory for the vivado project
mkdir build && cd build

#unpack the export project into the build directory
tar --strip-components=1 -xzvf path/to/faroshub04_${MYVARIANT}_export-*.tar.gz

#put vivado tools (2018.3 into the path)
source /opt/Xilinx/Vivado/2018.3/settings64.sh

#create and launch the project
vivado -source faroshub04_auto.tcl

#run generate bitstream in the vivado GUI

#create bootfiles
bootgen -arch zynqmp -image linux/faroshub04.bif -o BOOT.BIN -w
sha256sum BOOT.BIN linux/image.ub > manifest.txt

tar --transform='s,^.*/,,gx' -czvf bootfiles-faroshub04_${MYVARIANT}_export-*.tar.gz BOOT.BIN linux/image.ub manifest.txt

#load firmware and reboot hardware
python3 -m "pyfaros.updater" --file bootfiles-faroshub04_${MYVARIANT}_export-*.tar.gz <serial number> <credentials>
```

# DOCUMENT REVISION HISTORY

| Date | Revision | Revision Description |
|---|---|---|
| 2020/11/10 | 1.0 | Initial document creation. |

# NOTICE OF DISCLAIMER