Download and update imperix IP for FPGA sandbox

PN117 | Posted on September 5, 2024 | Updated on July 4, 2025

Benoît STEINMANN
Software Team Leader
imperix • in

Table of Contents

This page provides the imperix IP and other source files required for FPGA development on imperix controllers.

To learn how to use the imperix IP, please refer to the getting started with FPGA page and the imperix IP user guide page.

To find all FPGA-related notes, you can visit FPGA development homepage.

Download

The following table lists the different imperix IP version available. The minimal Vivado version required is 2022.1.

When upgrading an existing Vivado project, please refer to the **upgrade** section below.

C++ or ACG SDK	**imperix IP version**	**Download**
2024.3	3.10 Rev. 0	**FPGA_Sandbox_template_3.10rev0.zip**

2025.1 2025.2	3.10 Rev. 0 3.10 Rev. 1*	**FPGA_Sandbox_template_3.10rev1.zip**

New features of the imperix IP version 3.10 are shown here.

*An upgrade of the imperix IP from 3.10 Rev. 0 to 3.10 Rev. 1 is only required for users who need to configure the excitation frequency and resolution of the resolver.

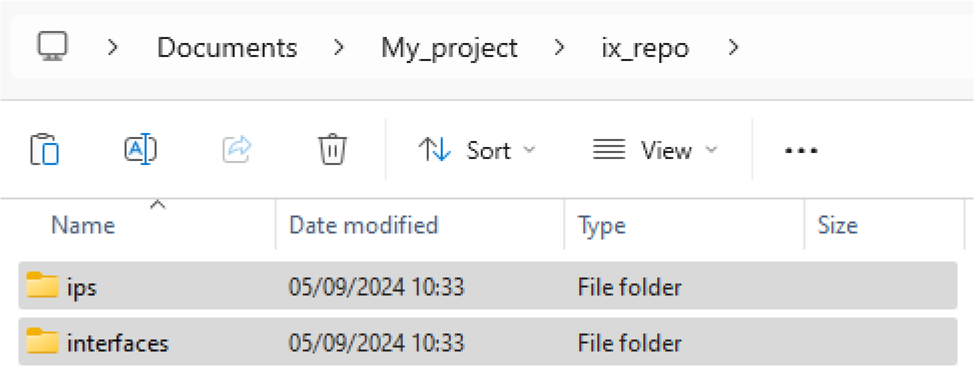
Legacy IPs for older SDKs

C++ or ACG SDK	**imperix IP version**	**Minimal Vivado version required**	**Download**
3.4.x.x 3.5.x.x	3.4 Rev. 1	2019.2	sandbox_sources_3.4_3.5.zip
3.6.x.x	3.6 Rev. 1	2019.2	sandbox_sources_3.6.zip
3.7.x.x	3.7 Rev. 1	2021.1	sandbox_sources_3.7.zip
3.8.x.x	3.8 Rev. 1	2022.1	sandbox_sources_3.8.zip
internal only	3.9 Rev. 1 to 3	2022.1	*internal only*
2024.1	3.9 Rev. 4	2022.1	FPGA_Sandbox_template_3.9rev4.zip ⚠️ Read upgrade section below
2024.2	3.9 Rev. 5	2022.1	FPGA_Sandbox_template_3.9rev5.zip

Upgrade procedure

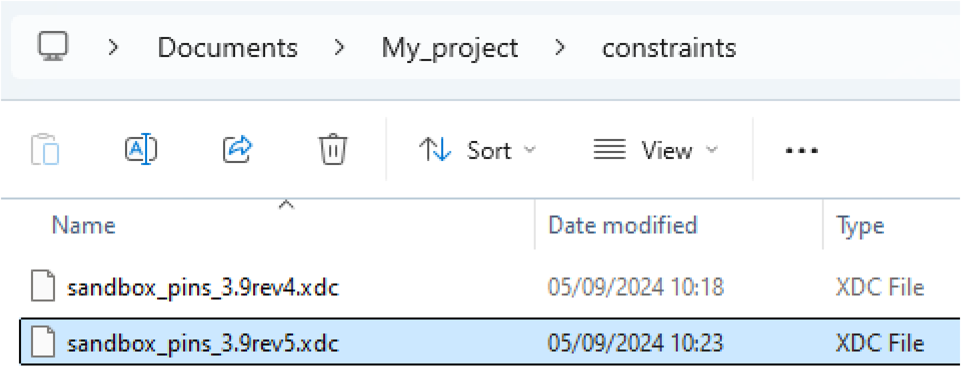
This section describes how to upgrade the imperix IP from an existing Vivado project.

1. Download the FPGA_Sandbox_template zip file for the targeted SDK version and unzip it.
2. In the ***<project_dir>/ix_repo/*** directory, replaces the ***interfaces*** and ***ips*** folders with the ones from the freshly downloaded sandbox template.



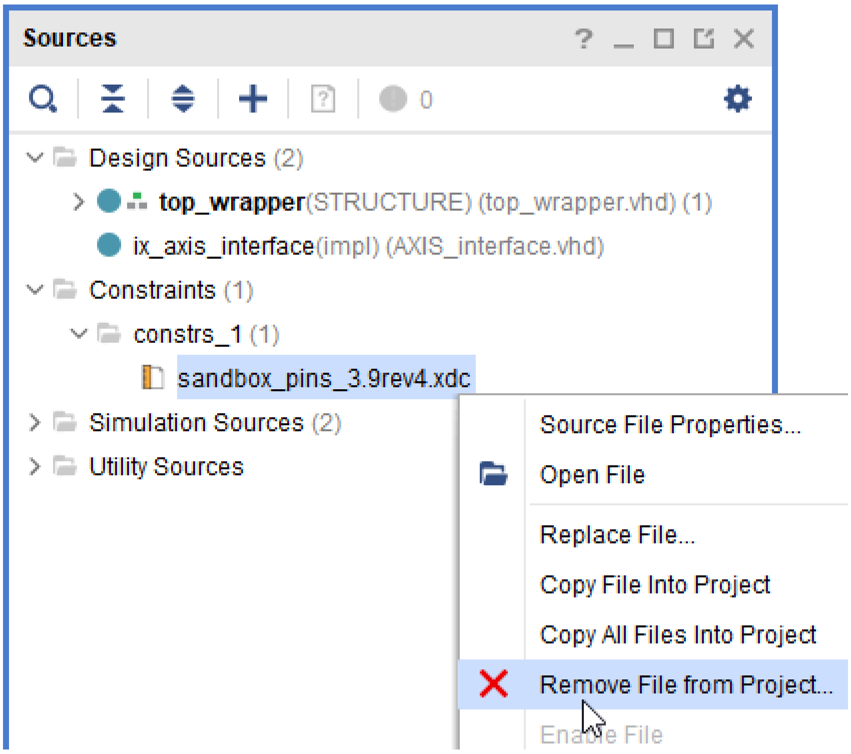
Name	Date modified	Type	Size
ips	05/09/2024 10:33	File folder	
interfaces	05/09/2024 10:33	File folder	

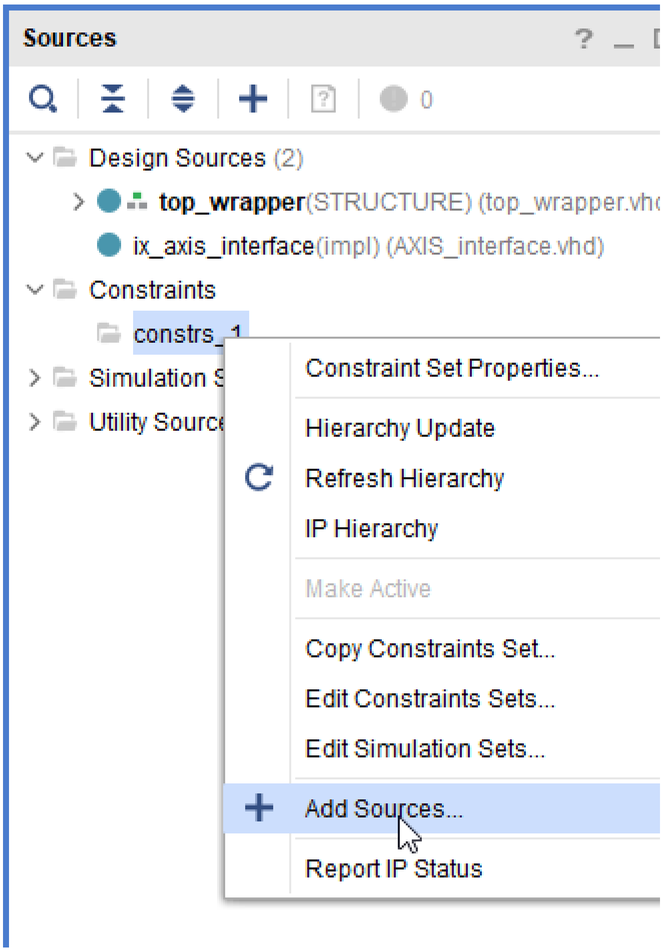
4. Copy the new constraint file to ***<project_dir>/constraints/***.
   ⚠ If modifications were made to the constraint file (e.g. to use USR pins), these modifications need to be reported to the new constraint file.

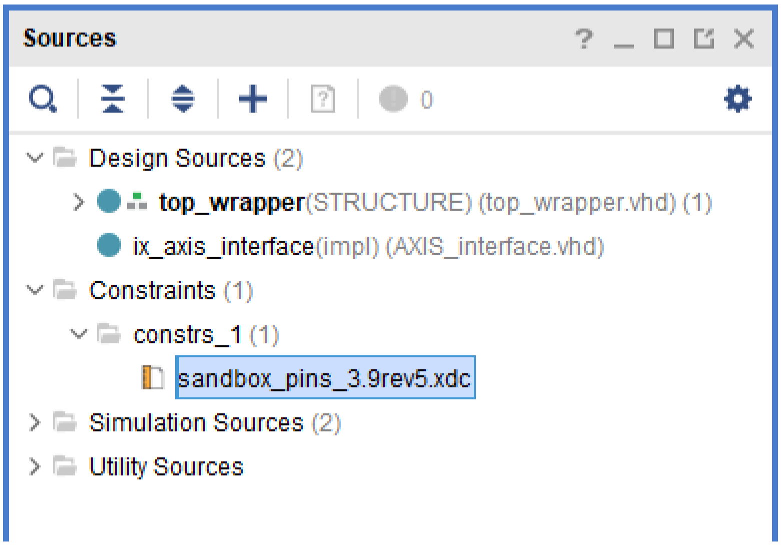


Name	Date modified	Type
sandbox_pins_3.9rev4.xdc	05/09/2024 10:18	XDC File
sandbox_pins_3.9rev5.xdc	05/09/2024 10:23	XDC File

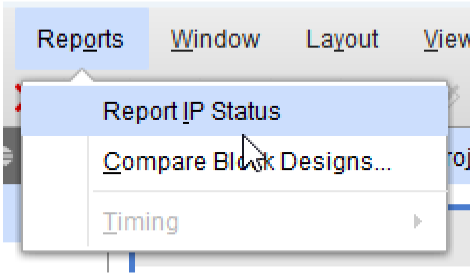
5. Open the Vivado project. Remove the old constraint file and add the new one.

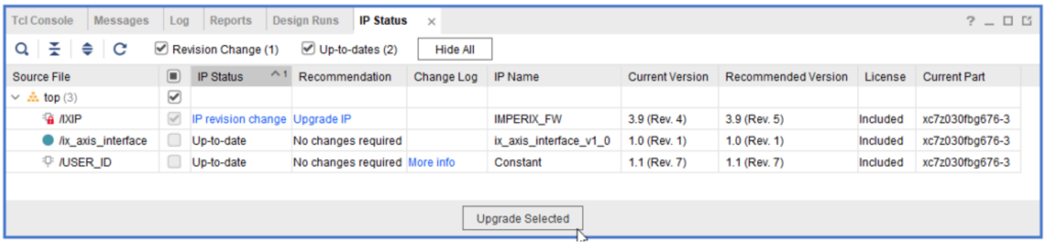






6. Upgrade the imperix sandbox IP. It may generate warnings, which is expected.





Tcl Console | Messages | Log | Reports | Design Runs | IP Status

☑ Revision Change (1) ☑ Up-to-dates (2) Hide All

Source File	■	IP Status	Recommendation	Change Log	IP Name	Current Version	Recommended Version	License	Current Part
top (3)	☑								
/IXIP	☑	IP revision change	Upgrade IP		IMPERIX_FW	3.9 (Rev. 4)	3.9 (Rev. 5)	Included	xc7z030fbg676-3
/ix_axis_interface	□	Up-to-date	No changes required		ix_axis_interface_v1_0	1.0 (Rev. 1)	1.0 (Rev. 1)	Included	xc7z030fbg676-3
/USER_ID	□	Up-to-date	No changes required	More info	Constant	1.1 (Rev. 7)	1.1 (Rev. 7)	Included	xc7z030fbg676-3

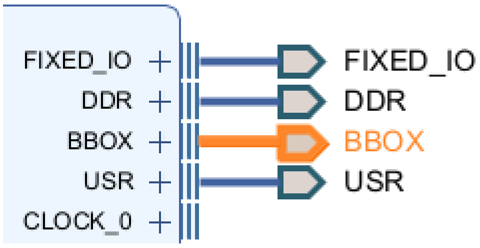
Upgrade Selected

7. Follow the sections below depending on the used imperix IP version.

Upgrading to 3.9 Rev. 4

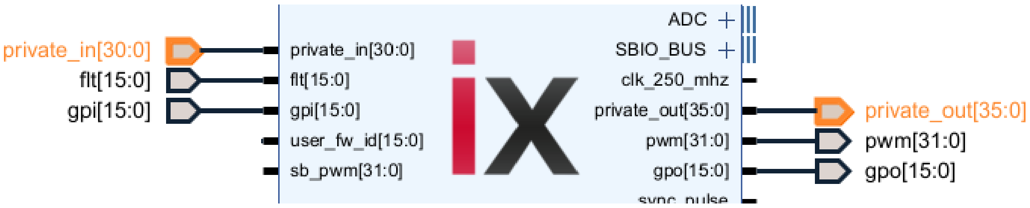
When upgrading the imperix IP to **version 3.9 Rev. 4 or later**, the following changes must be made

1. The interface **BBOX** was added, make sure to connect it to the top-level interface



Added BBOX interface

2. **`private_in`** and **`private_out`** size changed, make sure to update the top-level ports accordingly



The private_in and private_out size changed

3. After these changes, re-generate the top-level HDL wrapper. To make sure Vivado apply the change properly, we recommend deleting the `top_wrapper.vhd` and re-create the HDL wrapper.

In SDK 2024.1, a new feature called the synchronous averaging has been introduced. ADC channels with synchronous averaging enabled only output a new value once per CLOCK_0 period, regardless of the oversampling ratio set in the CONFIG block. When using an oversampling ratio, we recommend disabling the synchronous averaging option in the ADC blocks.

Upgrading to 3.9 Rev. 5

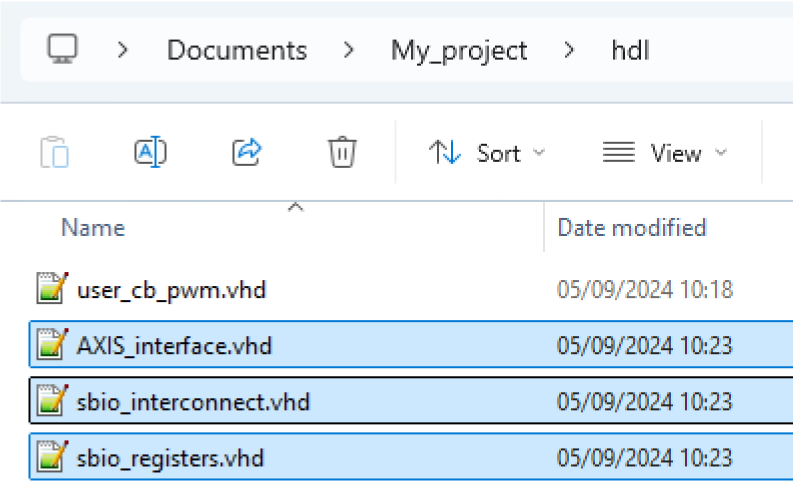
If upgrading from an older version, do not miss the **upgrading to 3.9 Rev.4** section above.

The imperix IP version 3.9 Rev. 5 brings the following changes. Additional information are provided after the procedure.

- The **SBI** and **SBO** interfaces were replaced by the memory-mapped **SBIO_BUS**. The provided VHDL module **sbio_registers.vhd** makes the bridge between the SBIO_BUS and the traditional SBI and SBO interfaces that were present in the imperix IP 3.9 Rev. 4 and earlier
- The **sbio_interconnect .vhd** was added for convenience

Upgrade procedure

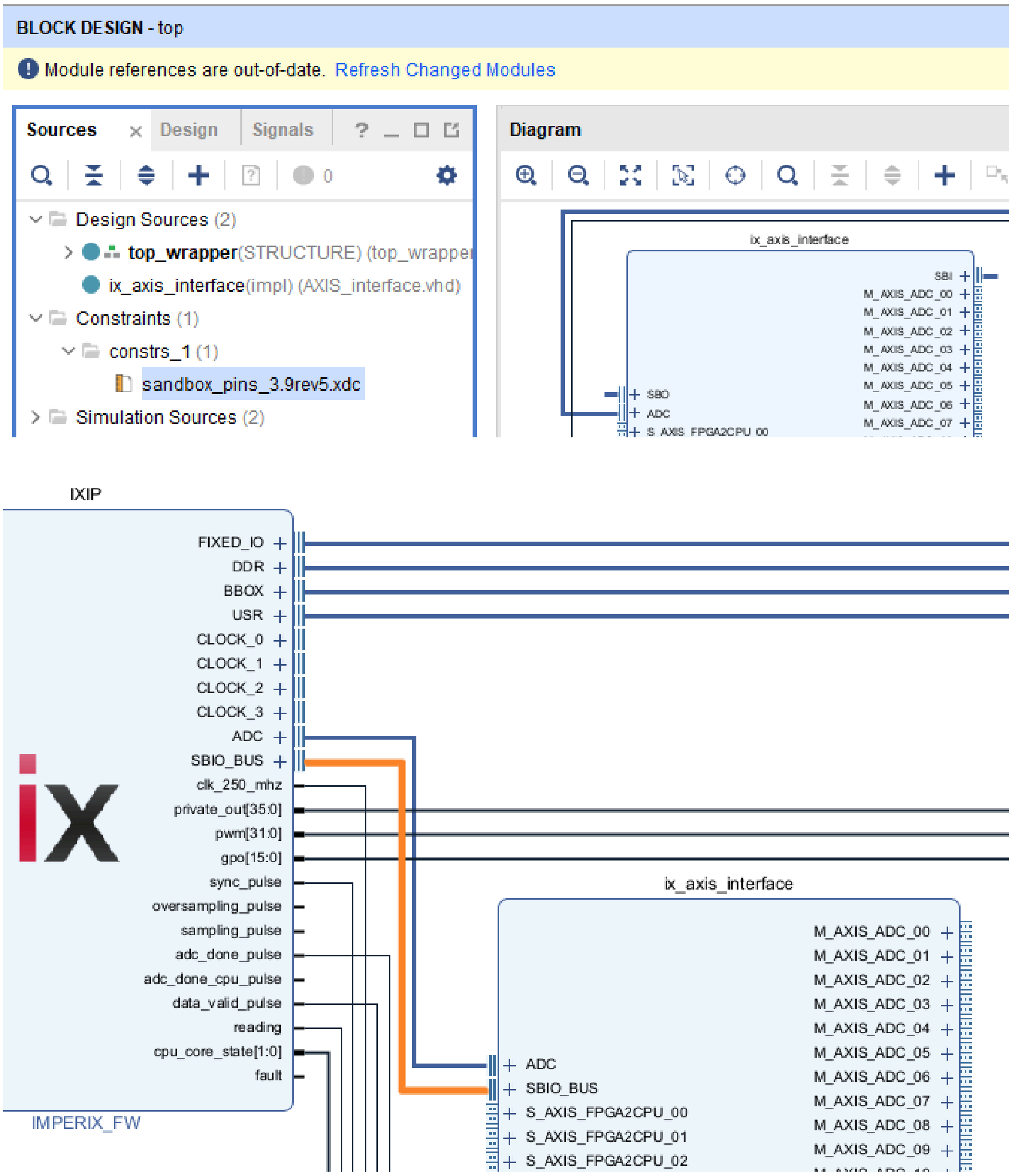
1. In ***<project_dir>/ix_repo/***
   – replace **AXIS_interface.vhd**
   – add **sbio_interconnect.vhd** and **sbio_registers.vhd**



Name	Date modified
user_cb_pwm.vhd	05/09/2024 10:18
AXIS_interface.vhd	05/09/2024 10:23
sbio_interconnect.vhd	05/09/2024 10:23
sbio_registers.vhd	05/09/2024 10:23

2. **If using the AXI-Stream interface (AXIS_interface.vhd)**

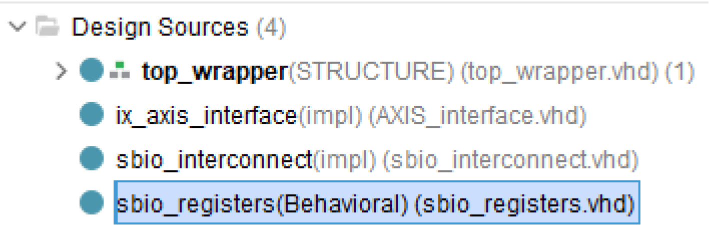
   In Vivado, click Refresh Changed Modules then reconnect the SBIO_BUS interface.



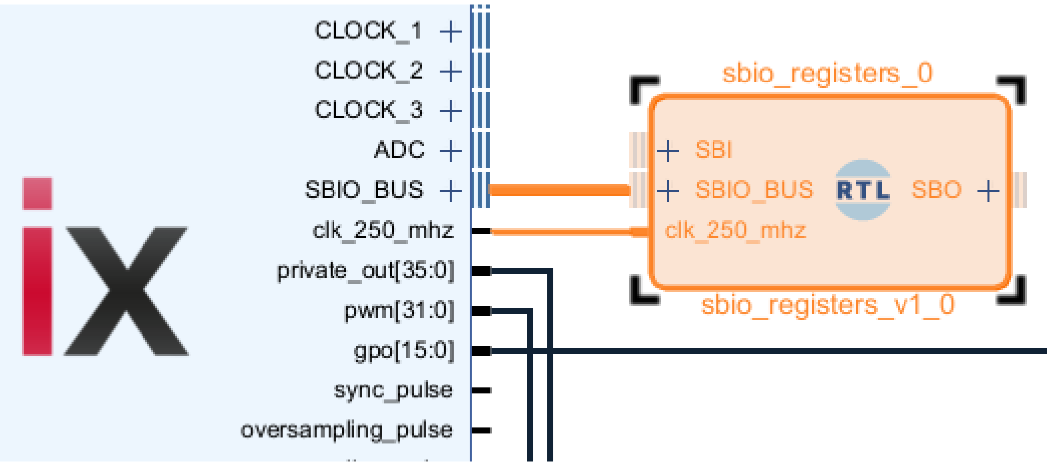
SBIO_BUS between the imperix IP and the AXI4-Stream interface

3. **If using SBI and SBO registers**

   The provided VHDL module `sbio_registers.vhd` makes the bridge between the `SBIO_BUS` and the traditional `SBI` and `SBO` interfaces that were present in the imperix IP 3.9 Rev. 4 and earlier.



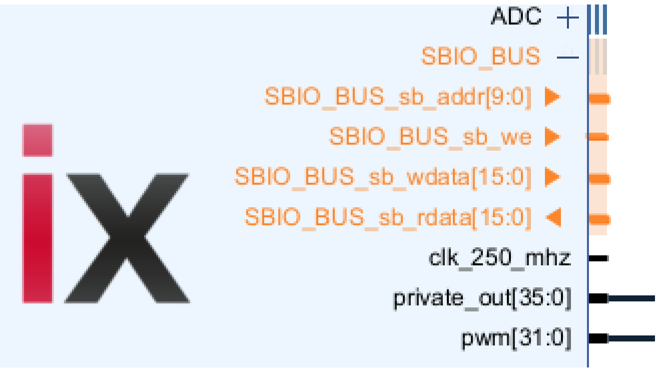
Design Sources (4)
- top_wrapper(STRUCTURE) (top_wrapper.vhd) (1)
- ix_axis_interface(impl) (AXIS_interface.vhd)
- sbio_interconnect(impl) (sbio_interconnect.vhd)
- sbio_registers(Behavioral) (sbio_registers.vhd)



sbio_register module that converts SBIO_BUS to SBI and SBO

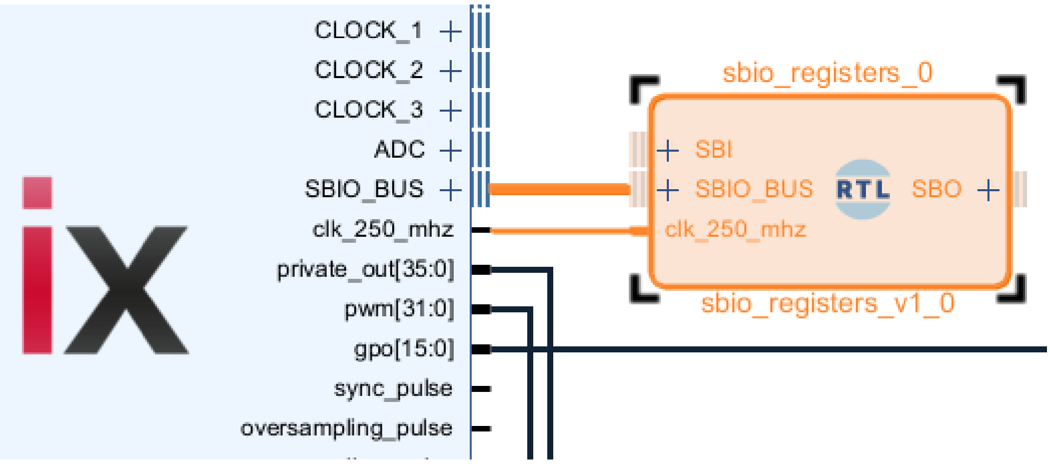
What’s new in 3.9 Rev. 5 (SDK 2024.2)

The `SBI` and `SBO` interfaces were replaced by the memory-mapped `SBIO_BUS`.
In the future, this bus allows addressing up to 1024 registers as well as using interconnects for an increase in flexibility.



Memory-mapped `SBIO_BUS` signals

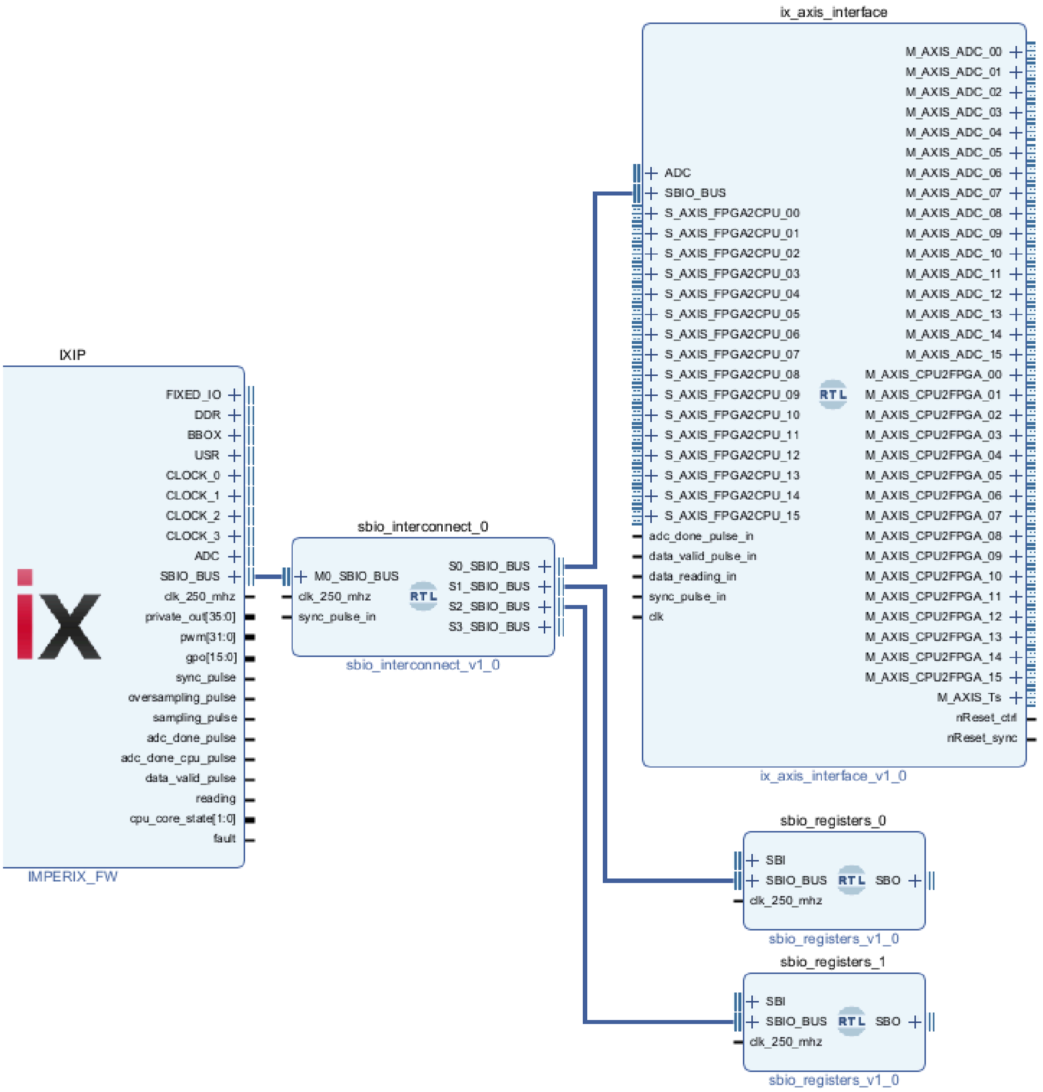
The provided VHDL module `sbio_registers.vhd` makes the bridge between the `SBIO_BUS` and the traditional `SBI` and `SBO` interfaces



sbio_register module that converts SBIO_BUS to SBI and SBO

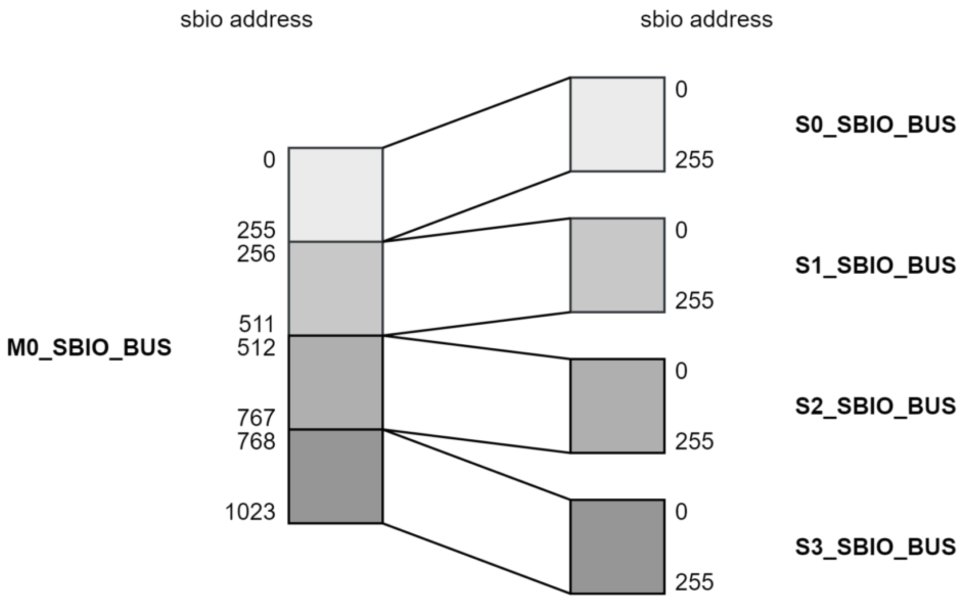
The `sbio_interconnect` was added for convenience

The SBIO interconnect increases the number of SBIO_BUS interfaces, allowing to connect multiple SBIO modules as illustrated below.



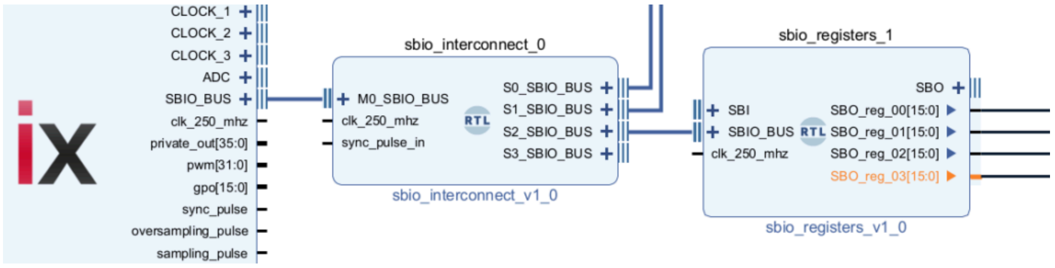
sbio_interconnect

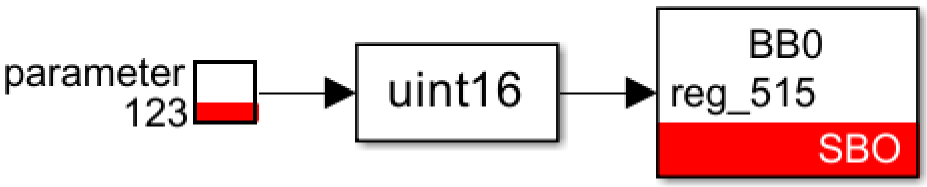
The address mapping of the SBIO interconnect is shown below, it divides the SBIO addressable range in 4 smaller areas.



sbio_interconnect memory mapping

As an example, to write to **SBO_reg_03** of an `sbio_registers` block connected to **S2_SBIO_BUS**, The user has to use an SBO block to register number 512+3=**515**.





What’s new in 3.10 (SDK 2024.3 and SDK 2025.1)

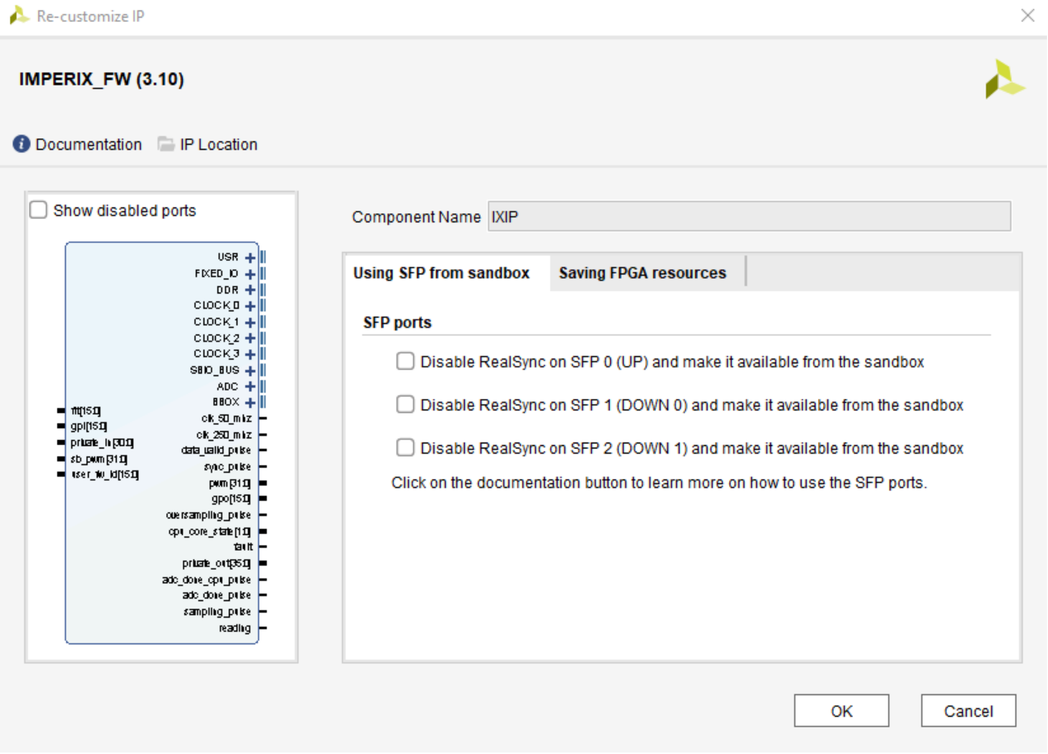
In version 3.10, the IP becomes configurable:

- **The SFP port can be repurposed by the user, to use the Aurora protocol for instance.**
- **Unused resources can be removed, to save up FPGA resources.**

Repurposing the SFP ports

Checking the box “Disable RealSync on SFP” will remove the imperix RealSync logic for the selected SFP and free the GTX transceiver. The SFP can then be repurposed by the user to implement its own communication, using the Aurora 8B10B IP for instance.

An example of SFP port repurposing is available on the page Example of FPGA-based Aurora 8B/10B communication.



Re-customize IP

IMPERIX_FW (3.10)

Documentation IP Location

□ Show disabled ports

Component Name: IXIP

Using SFP from sandbox | Saving FPGA resources

SFP ports

□ Disable RealSync on SFP 0 (UP) and make it available from the sandbox

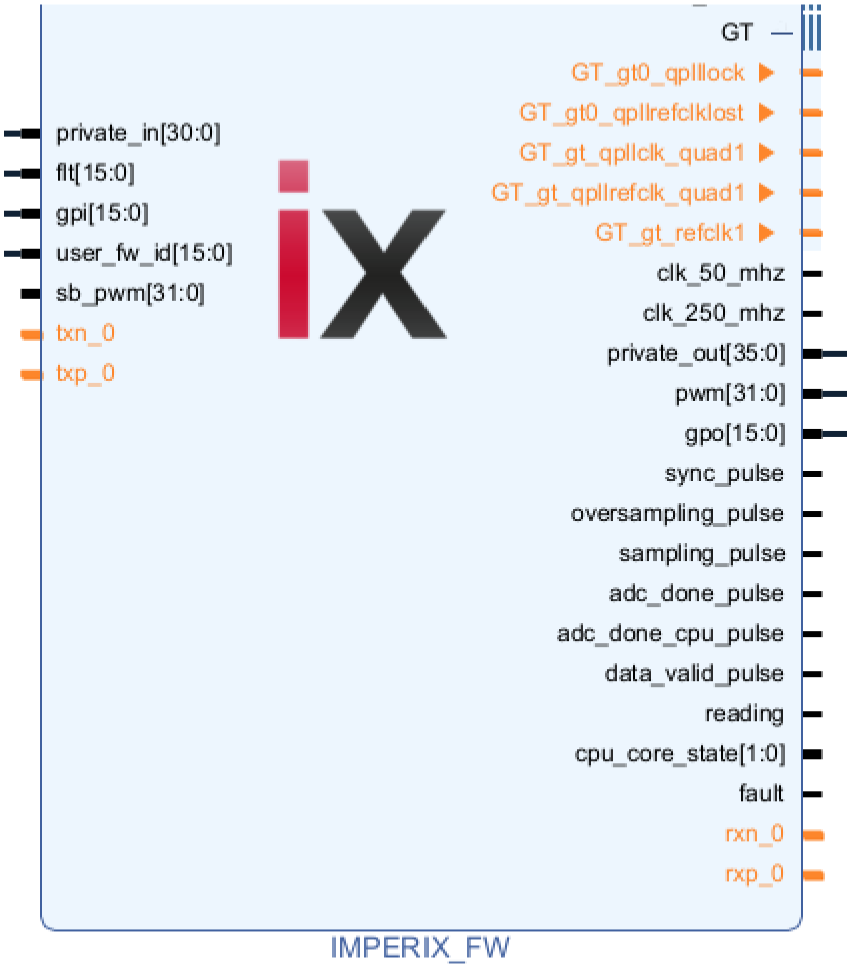
□ Disable RealSync on SFP 1 (DOWN 0) and make it available from the sandbox

□ Disable RealSync on SFP 2 (DOWN 1) and make it available from the sandbox

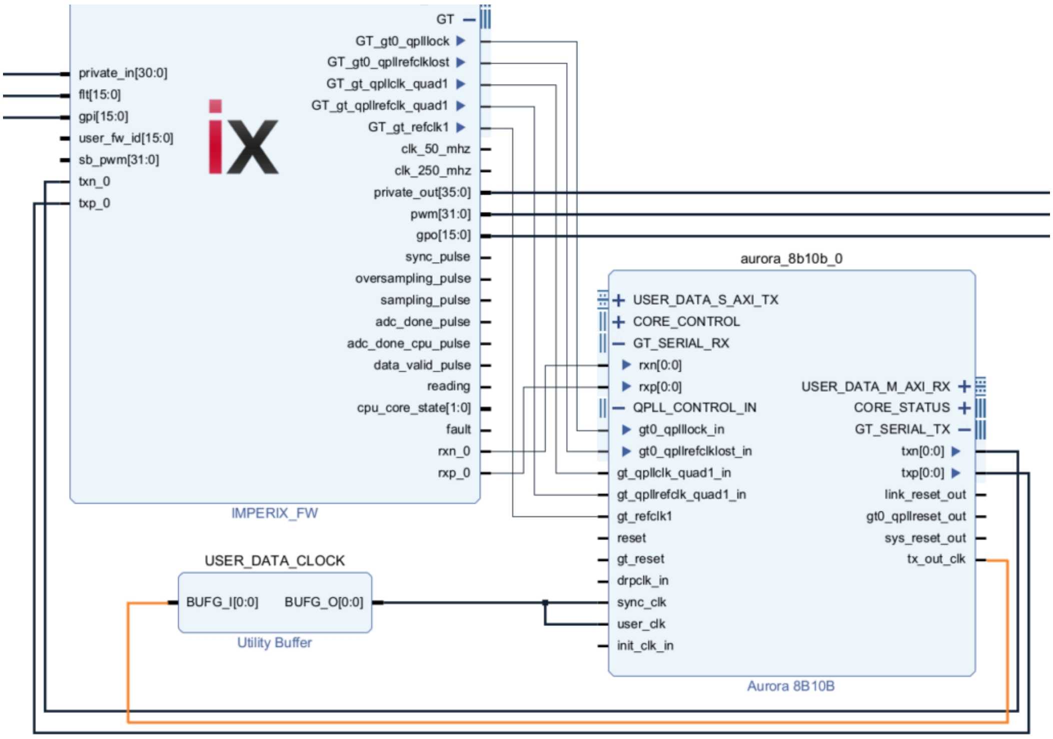
Click on the documentation button to learn more on how to use the SFP ports.

Settings to disable RealSync on specific ports

Checking the box “Disable RealSync on SFP 0” will reveal the **txn**, **txp**, **rxn**, and **rxp** ports, as well as the **GT** interface, which contains the *Shared Logic* ports.



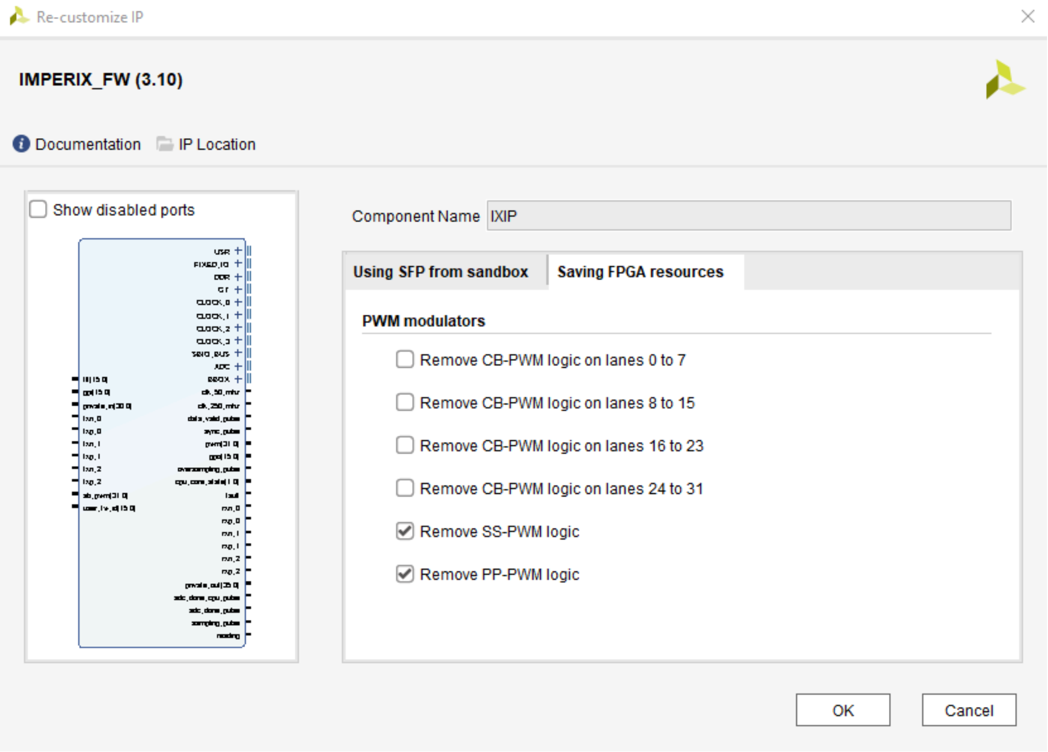
The image below shows how an Aurora 8B10B IP can be used on the SFP port. Make sure that the "**GT Refclk**" parameter is set to 250 MHz and that the "**include Shared Logic in example design**" is checked. The other parameters can be changed freely.



An example of block design showing how to set up communication with Aurora is available here.

Saving FPGA resources

Some modules can be removed from the design to free FPGA resources. For now, only PWM modulators can be removed, but in the future we intend to allow disabling additional modules (for instance modules only used by the TPI8032) to make even more FPGA resources available to the user.



□ Show disabled ports

Component Name IXIP

PWM modulators

□ Remove CB-PWM logic on lanes 0 to 7

□ Remove CB-PWM logic on lanes 8 to 15

□ Remove CB-PWM logic on lanes 16 to 23

□ Remove CB-PWM logic on lanes 24 to 31

☑ Remove SS-PWM logic

☑ Remove PP-PWM logic

Settings to remove unused PWM modulators to save resources in the FPGA sandbox

The table below summarize the resources saved for each option. The percentages indicate the percentage of resources used by the component compared to the total resources available in the FPGA. Checking all the boxes frees around 18% of FPGA resources. With all the boxes ticked, the imperix IP takes around 30% of the FPGA resources available in the FPGA.

Resource usage	Slice LUTs	Slice registers
CB-PWM lane 0 to 7	1619 (2.39%)	12165 (1.38%)
CB-PWM lane 8 to 15	1619 (2.39%)	12165 (1.38%)
CB-PWM lane 16 to 23	1619 (2.39%)	12165 (1.38%)
CB-PWM lane 24 to 31	1619 (2.39%)	12165 (1.38%)
SS-PWM	2732 (3,48%)	3305 (2,10%)
PP-PWM	4303 (5,47%)	9162 (5,83%)

FPGA resources saved for each option

Back to FPGA development homepage