Motor_Control

(Based On KD240)

AMD

together we advance_

FOC Motor Control Application
Quick Start

Install Ubuntu on AMD

PYNQ currently only supports ZYNQ Based and Versal series(include KRIA , Alveo).

PYNQ is a framework designed for the Ubuntu environment, so we must first install Ubuntu on the development board.

[Install Ubuntu on AMD | Ubuntu]

CHOOSE A BOARD

- Kria™ K24 SOMs
- Kria™ K26 SOMs
- Zynq™ UltraScale+™ MPSoC Development Boards
- Versal™ Adaptive SoC Evaluation Kit

Kria™ K24 SOMs (KD240)

Ubuntu Server 22.04

The version of optimised Ubuntu Server 22.04 is beta for now, the certified version is coming soon.

Works on:

- AMD Kria™ KD240 Drives Starter Kit

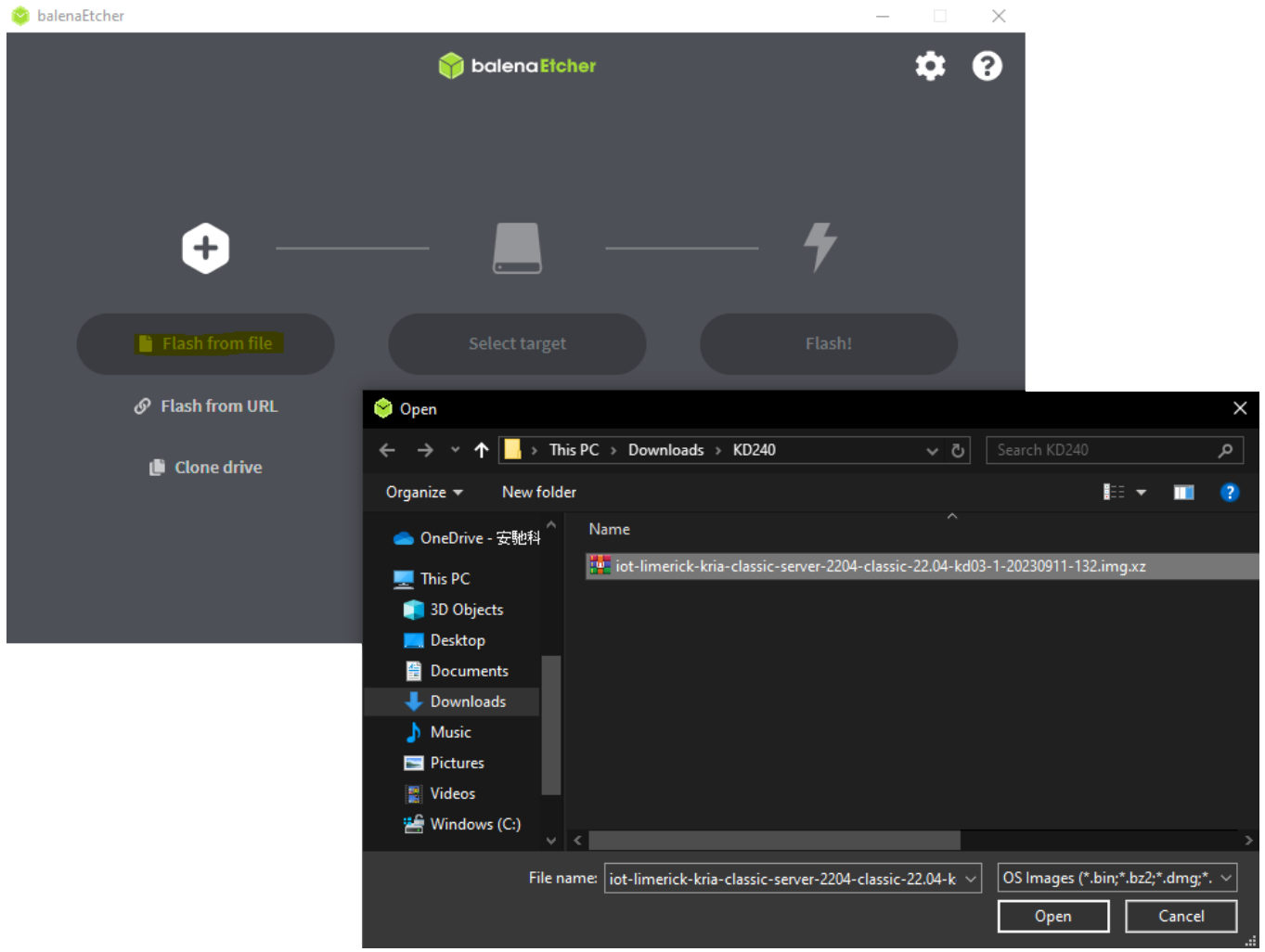
Please check the AMD Kria™ Wiki for the platform's latest boot firmware, technical documentation, and the Ubuntu for AMD-Xilinx Devices Wiki for known issues and limitations.

Download 22.04

Setting Up the SD Card Image (Ubuntu)

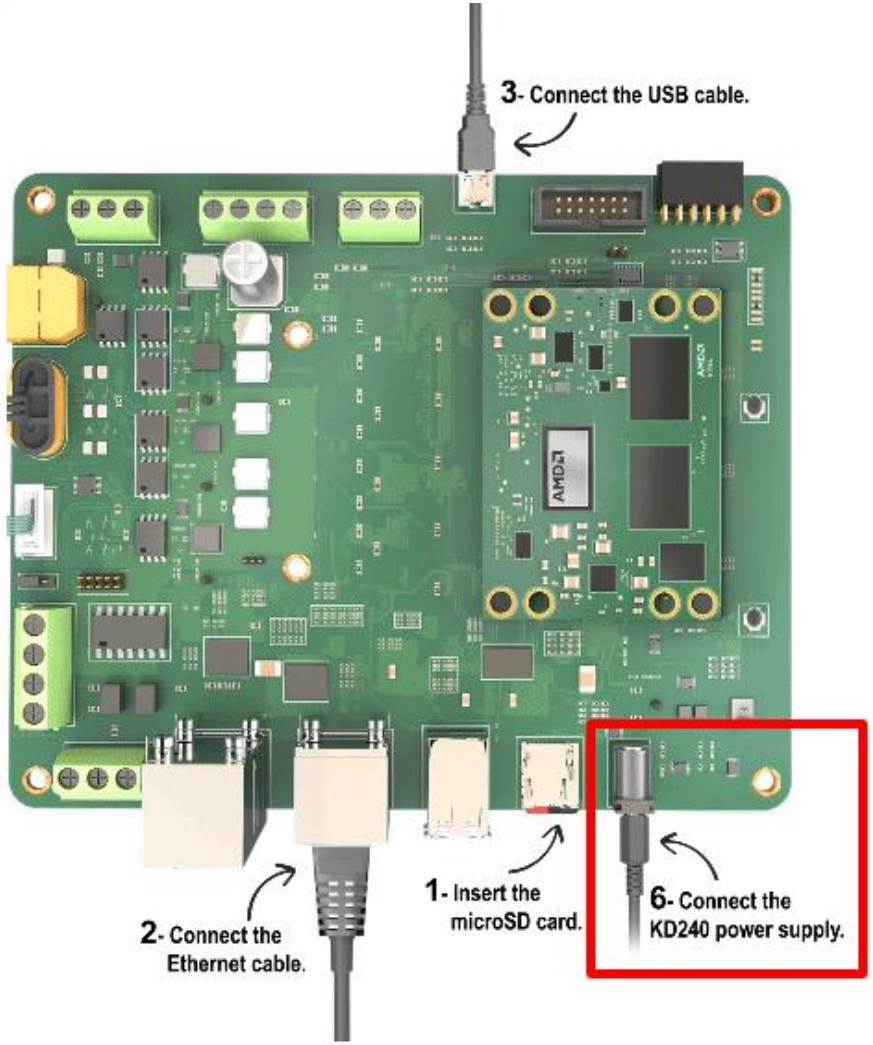
Follow the instructions in the tool and select the downloaded image to flash onto your microSD card.

[Setting up the SD Card Image (xilinx.com)]



Board Setup

Power on and communicate with the development board through UART (choose a preferred serial port debugging app.




```
2. COM17 (USB Serial Port (COM1
kria login: ubuntu
Password:
You are required to change your password immediately (administrator enforced).
Changing password for ubuntu.
Current password:
New password:
Retype new password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-9002-xilinx-zynqmp aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Dec 21 05:15:06 UTC 2023

  System load:  0.11962890625     Processes:             122
  Usage of /:   6.2% of 28.21GB   Users logged in:       0
  Memory usage: 10%               IPv4 address for eth0: 10.8.3.232
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
```

Install Ubuntu on KD240

The login username and password are both "ubuntu" After the first login, you will be prompted to change your password.

After logging in, be sure to execute the following command:

```
sudo apt-get update
sudo apt-get upgrade
```

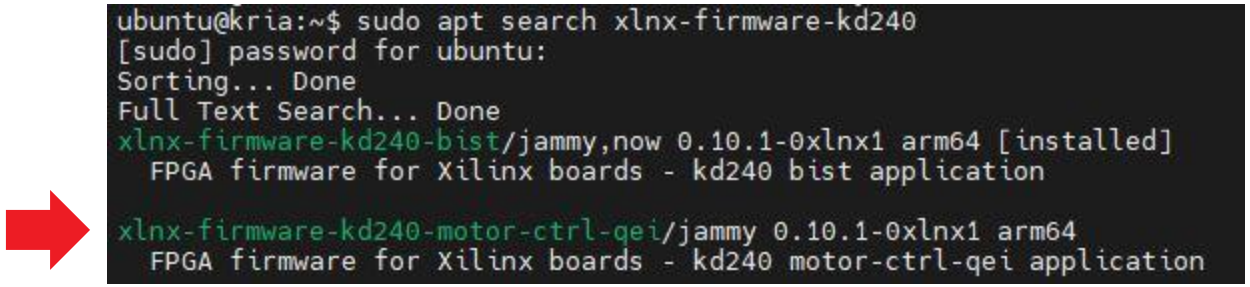
It will take about one hour.

Initial Setup

- Download the firmware

  Search package feed for packages compatible with KD240

```
sudo apt search xlnx-firmware-kd240
```



  Install firmware binaries

```
sudo apt install xlnx-firmware-kd240-motor-ctrl-qei
```

- Install motor control application

```
sudo apt install xlnx-app-kd240-foc-motor-ctrl
```

Run the motor control application

- Load the firmware

  Show the list and status of available application firmware

```
sudo xmutil listapps
```

  Load the desired application firmware

```
sudo xmutil unloadapp
```

```
ubuntu@kria:~$ sudo xmutil listapps
                  Accelerator         Accel_type                      Base         Base_type      #slots(PL+AIE)        Active_slot

                   kd240-bist           XRT_FLAT                 kd240-bist          XRT_FLAT             (0+0)                 -1
              k24-starter-kits           XRT_FLAT            k24-starter-kits          XRT_FLAT             (0+0)                 -1
          kd240-motor-ctrl-qei           XRT_FLAT        kd240-motor-ctrl-qei          XRT_FLAT             (0+0)                 -1
```

```
sudo xmutil loadapp kd240-motor-ctrl-qei
```

```
ubuntu@kria:~$ sudo xmutil listapps
                  Accelerator         Accel_type                      Base         Base_type      #slots(PL+AIE)        Active_slot

                   kd240-bist           XRT_FLAT                 kd240-bist          XRT_FLAT             (0+0)                 -1
              k24-starter-kits           XRT_FLAT            k24-starter-kits          XRT_FLAT             (0+0)                 -1
          kd240-motor-ctrl-qei           XRT_FLAT        kd240-motor-ctrl-qei          XRT_FLAT             (0+0)                  0,
```

Run the motor control application

- Run the bokeh server

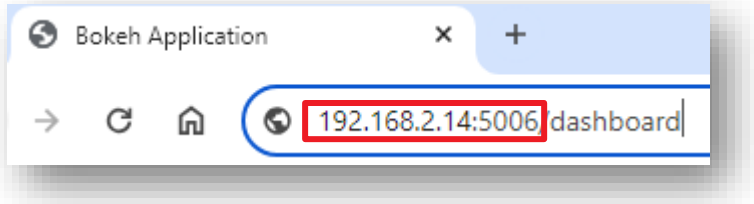
  Run the application to launch bokeh server for the dashboard

```
export PATH=${PATH}:/opt/xilinx/xlnx-app-kd240-foc-motor-ctrl/bin

start_motor_dashboard
```

```
ubuntu@kria:~$ sudo xmutil unloadapp
remove from slot 0 returns: 0 (Ok)
ubuntu@kria:~$ sudo xmutil loadapp kd240-motor-ctrl-qei
kd240-motor-ctrl-qei: loaded to slot 0
ubuntu@kria:~$ export PATH=${PATH}:/opt/xilinx/xlnx-app-kd240-foc-motor-ctrl/bin
ubuntu@kria:~$ start_motor_dashboard
Please enter the password for sudo access
Firmware kd240-motor-ctrl-qei is loaded
To the access the Application, enter '192.168.2.14:5006" in the host machine's browser.
ubuntu@kria:~$ 2023-09-14 16:21:29,693 Starting Bokeh server version 2.4.3 (running on Tornado 6.1)
2023-09-14 16:21:29,702 User authentication hooks NOT provided (default user enabled)
2023-09-14 16:21:29,723 Bokeh app running at: http://localhost:5006/dashboard
2023-09-14 16:21:29,724 Starting Bokeh server with process id: 2485
```

- On the host PC

  Open <ip>:5006 in a web browser

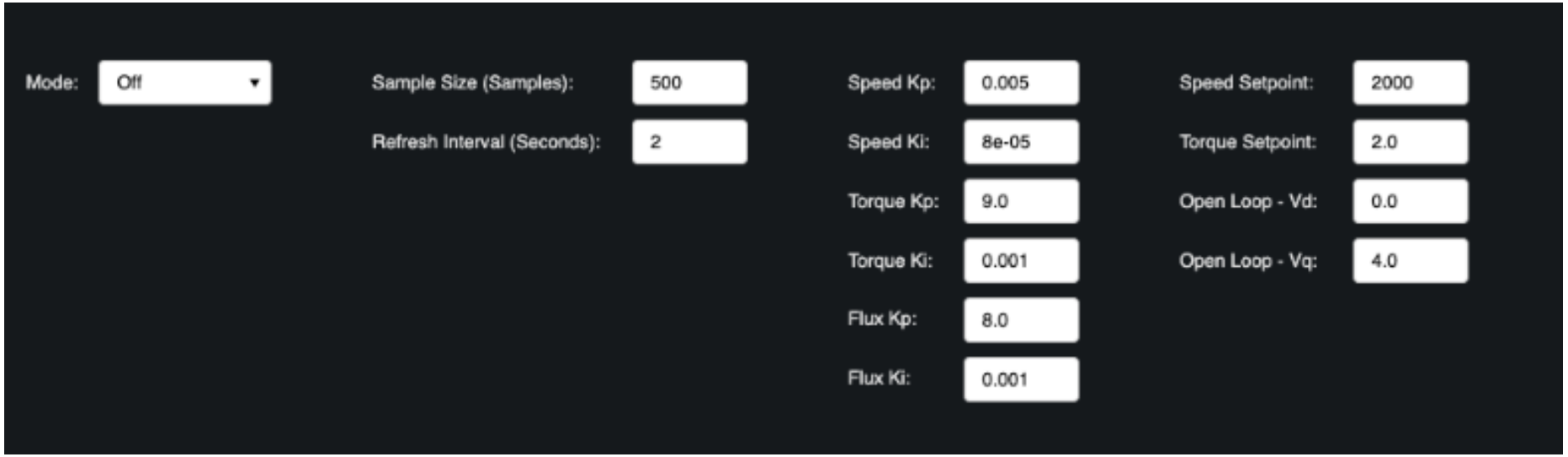


Dashboard

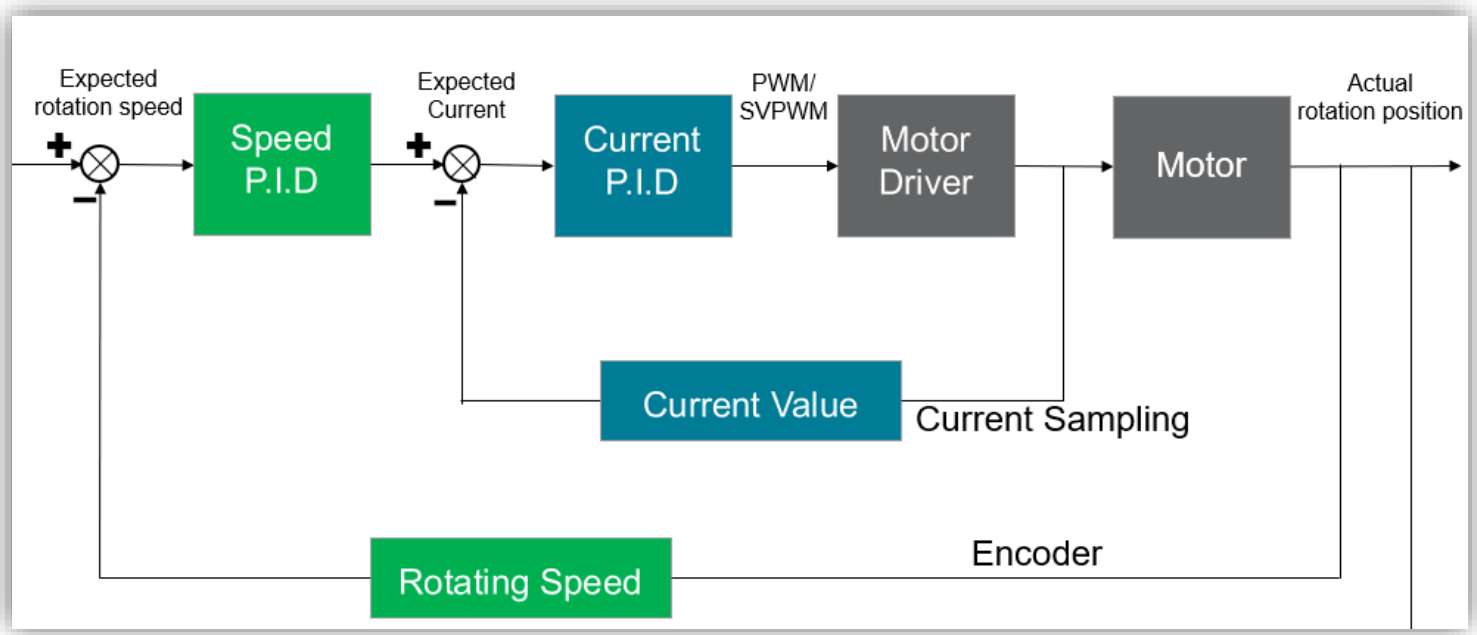
Dashboard Features

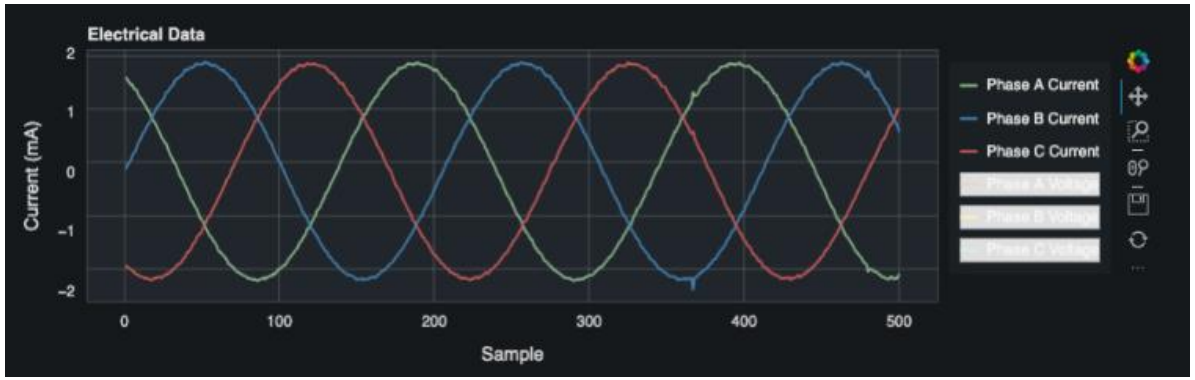
- Mode : Speed , Torque , Open Loop
- Samples are collected at 100 microsecond intervals. The maximum number of samples is limited to 3000.
- The valid range of speed setpoints is -10000 to 10000 rpm in Speed Mode.
- The valid range of torque setpoints is -2.5 to 2.5 amps in Torque Mode.
- The Open Loop - Vd text box is used to set the direct voltage (Vd) and quadrature voltage (Vq).

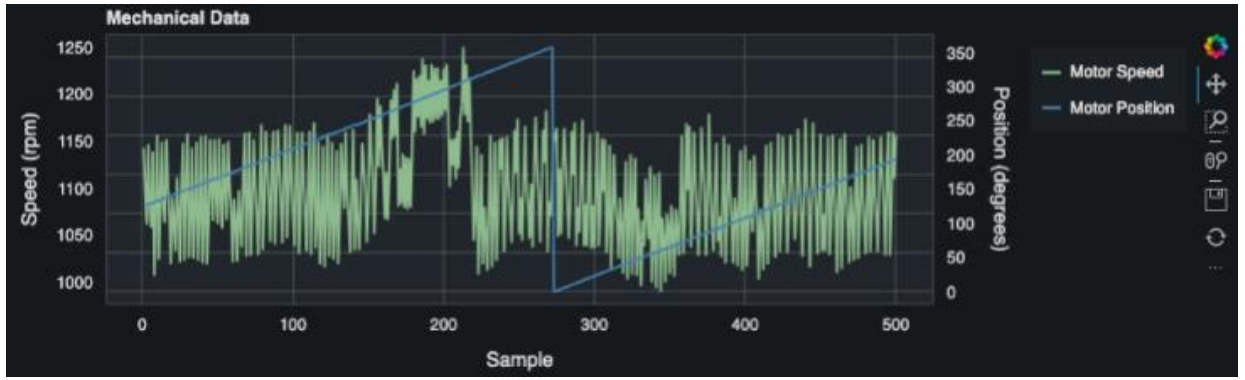
  The valid range for Vd is -24 to 24 volts & Vq is -24 to 24 volts.(Normally Vd should be set to ~0V.)



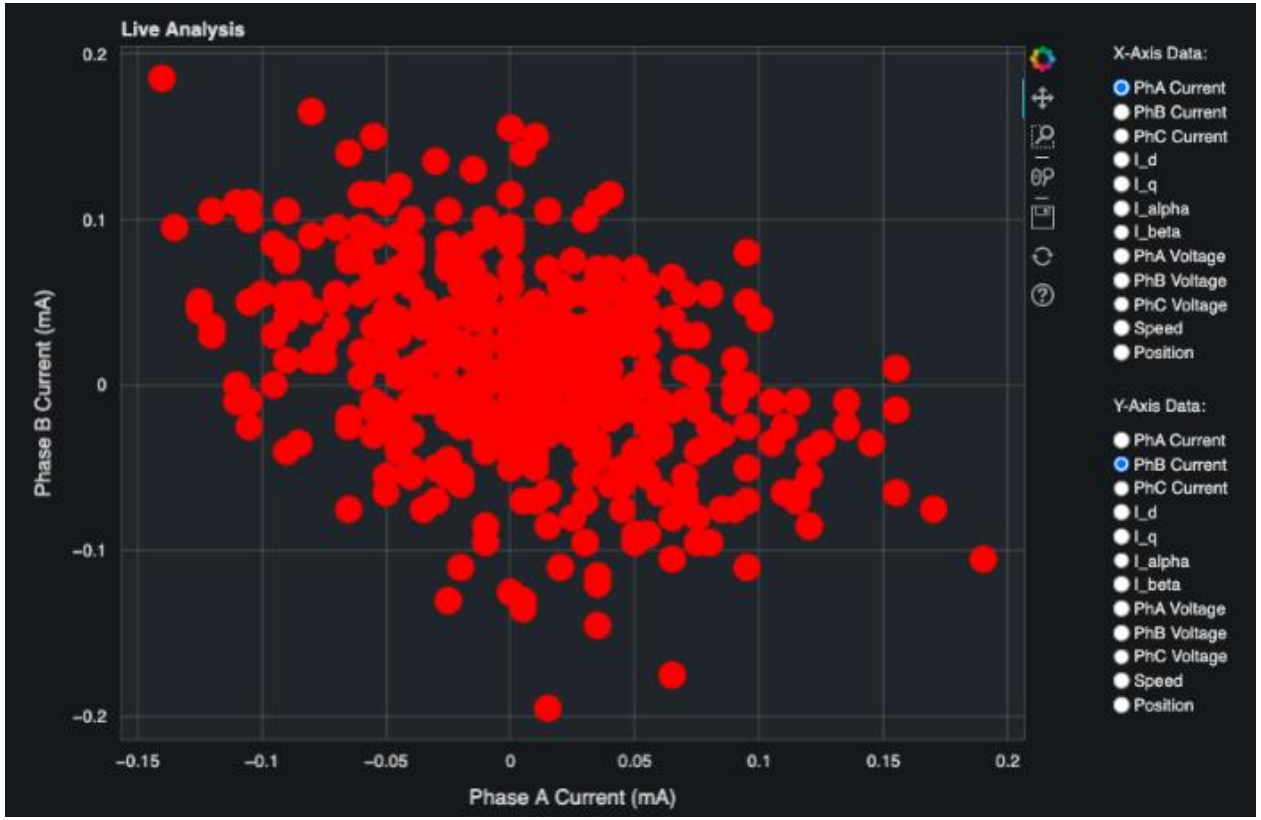
Dashboard Features

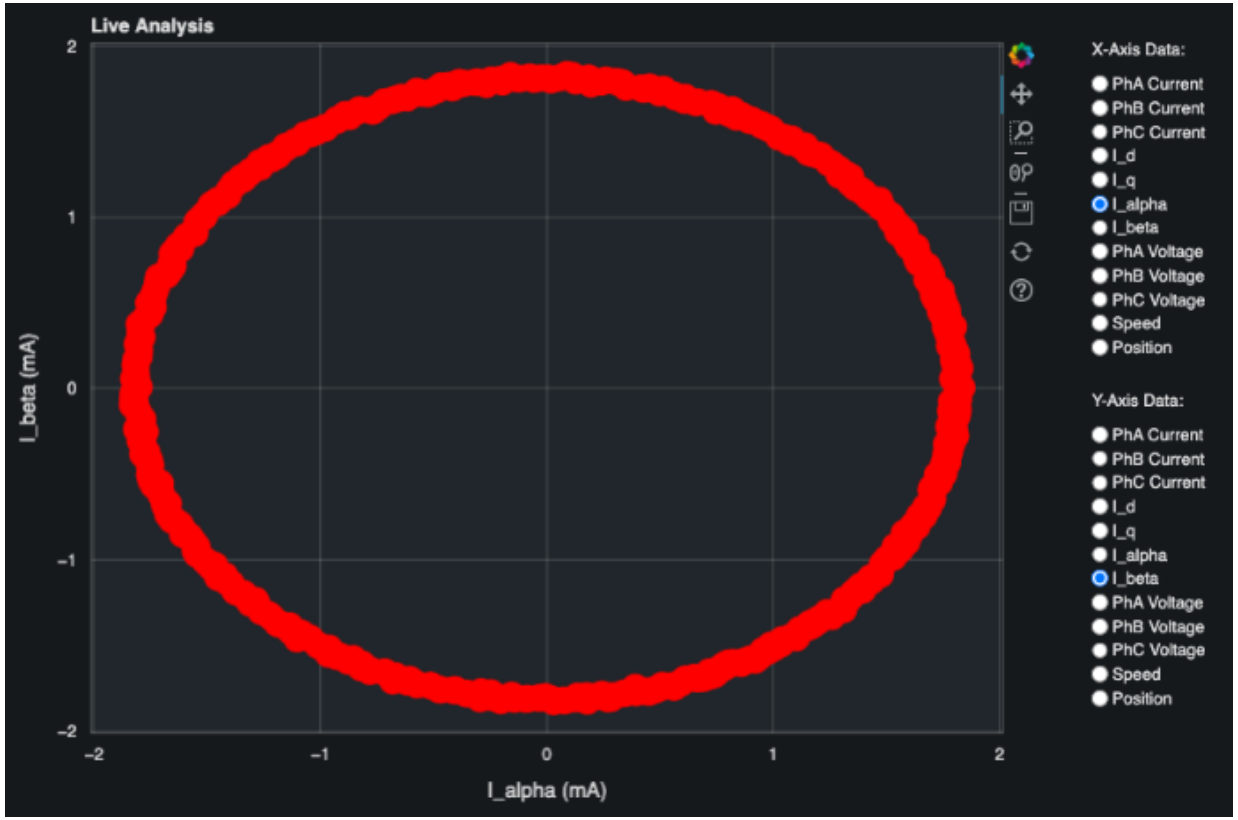






Dashboard





Using Vivado to Build the Hardware Design

Build the Hardware Design

Generating an Extensible XSA

(The following commands need to be executed on a PC where Vivado is installed.)

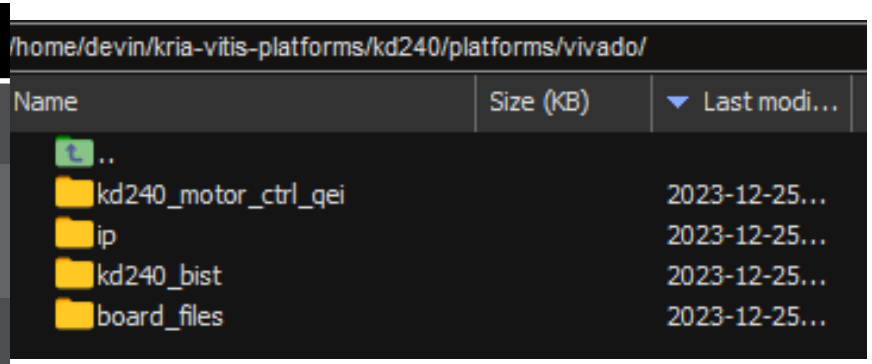
```
git clone --branch xlnx_rel_v2023.1 --recursive https://github.com/Xilinx/kria-vitis-platforms.git
```

Navigate to the kria-vitis-platforms/kd240, which is the working directory.

Go to the platform directory specific to the application.

```
cd $working_dir/platforms/vivado/<platform_name>
```

Application	Platform Name
Built-in self test (BIST)	kd240_bist
FOC motor control with position sensor	kd240_motor_ctrl_qei
ROS TSN Pub Sub	kd240_motor_ctrl_qei



Build the Hardware Design

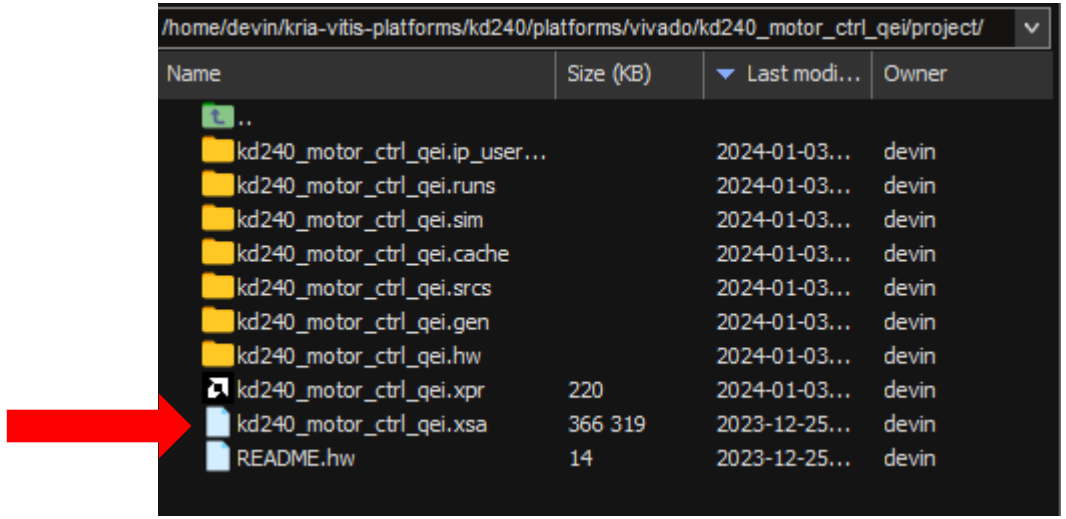
To build the XSA, source Vivado, and run the following command.

The XSA generation can take some time depending on the system specification.

```
source $HLS_install_path/Vvitits_HLS/2023.1/settings64.sh

make xsa
```

The generated XSA is located at:

```
working_dir/platforms/vivado/<platform_name>/project/<platform_name>.xsa
```

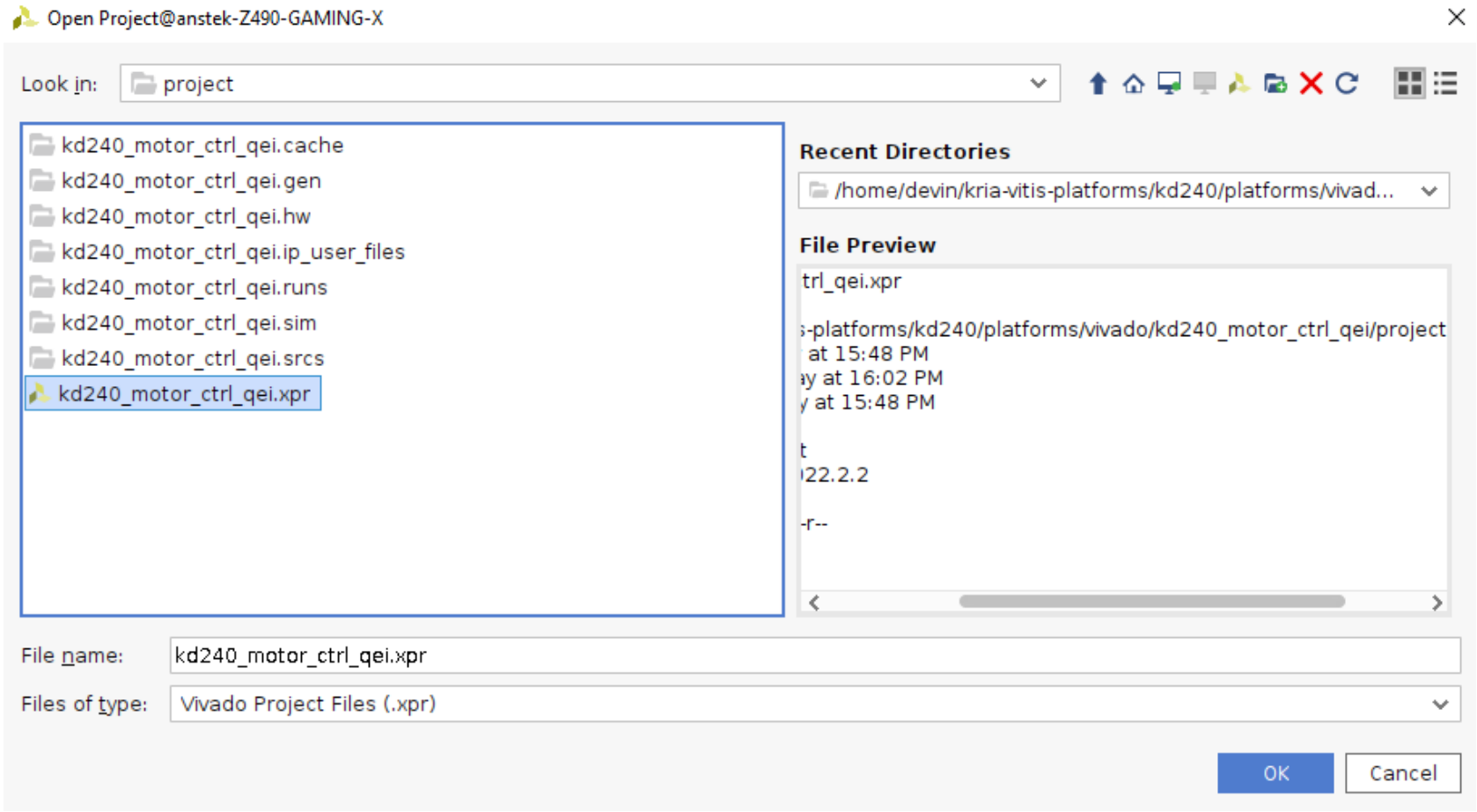


Modifying the Vivado Design and Creating a New XSA

Open the Vivado GUI, then run the following command from the Vivado Tcl console:

```
open_project ./project/<platform_name>.xpr
```

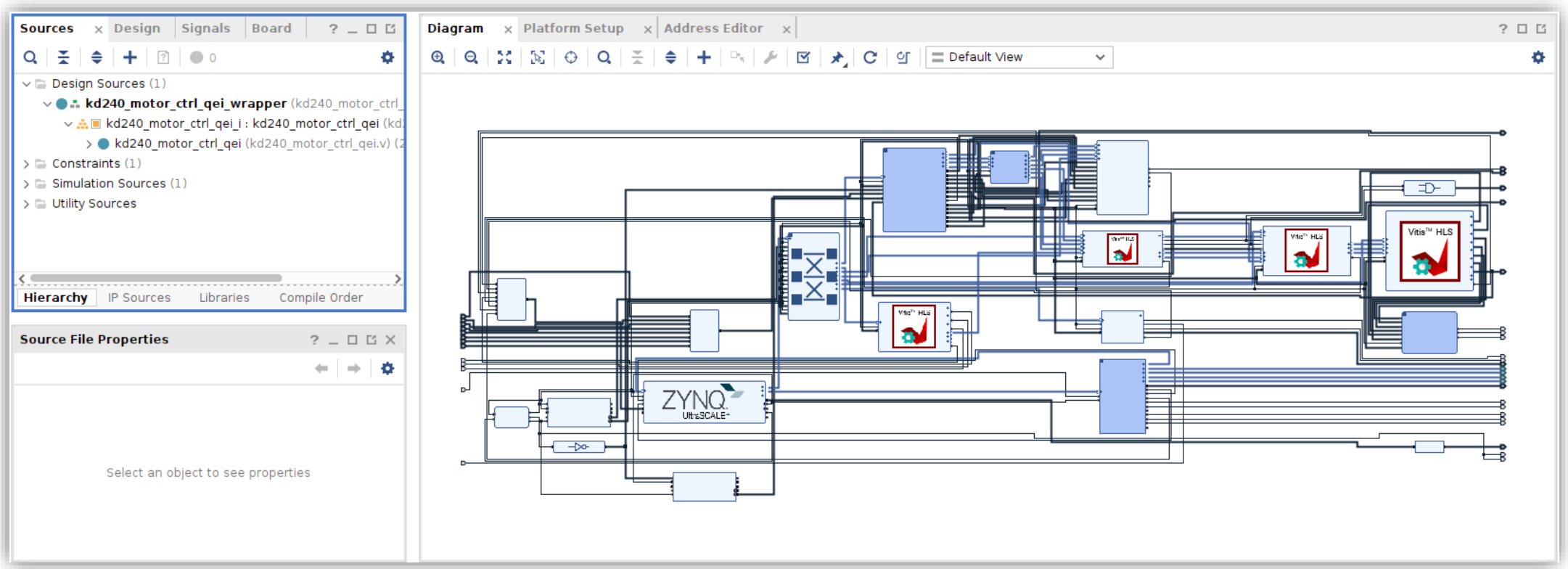
Or:



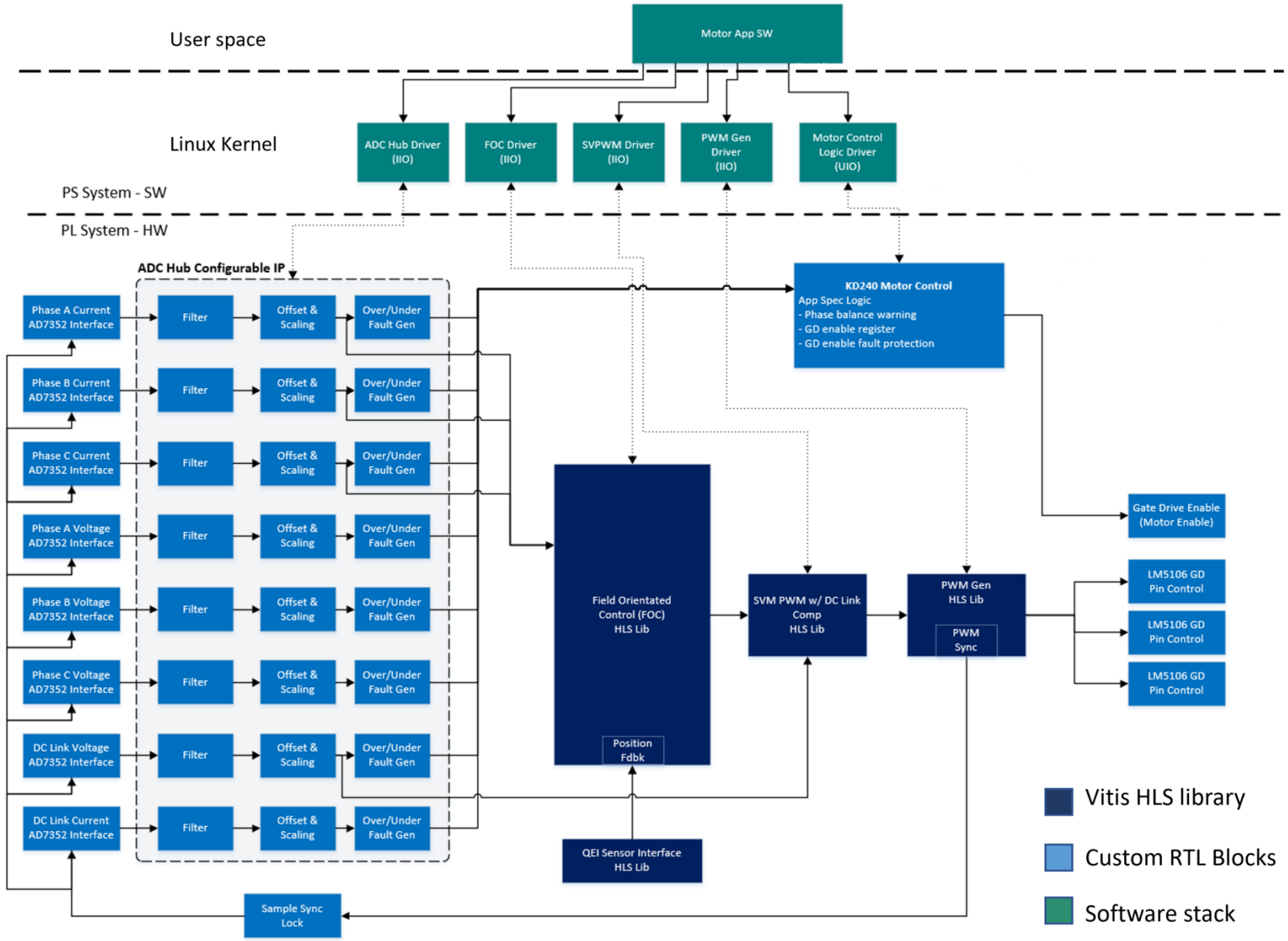
Modifying the Vivado Design and Creating a New XSA

Open the block design, and you can see the components of the platform IP.

Some IPs are generated by Vitis HLS.

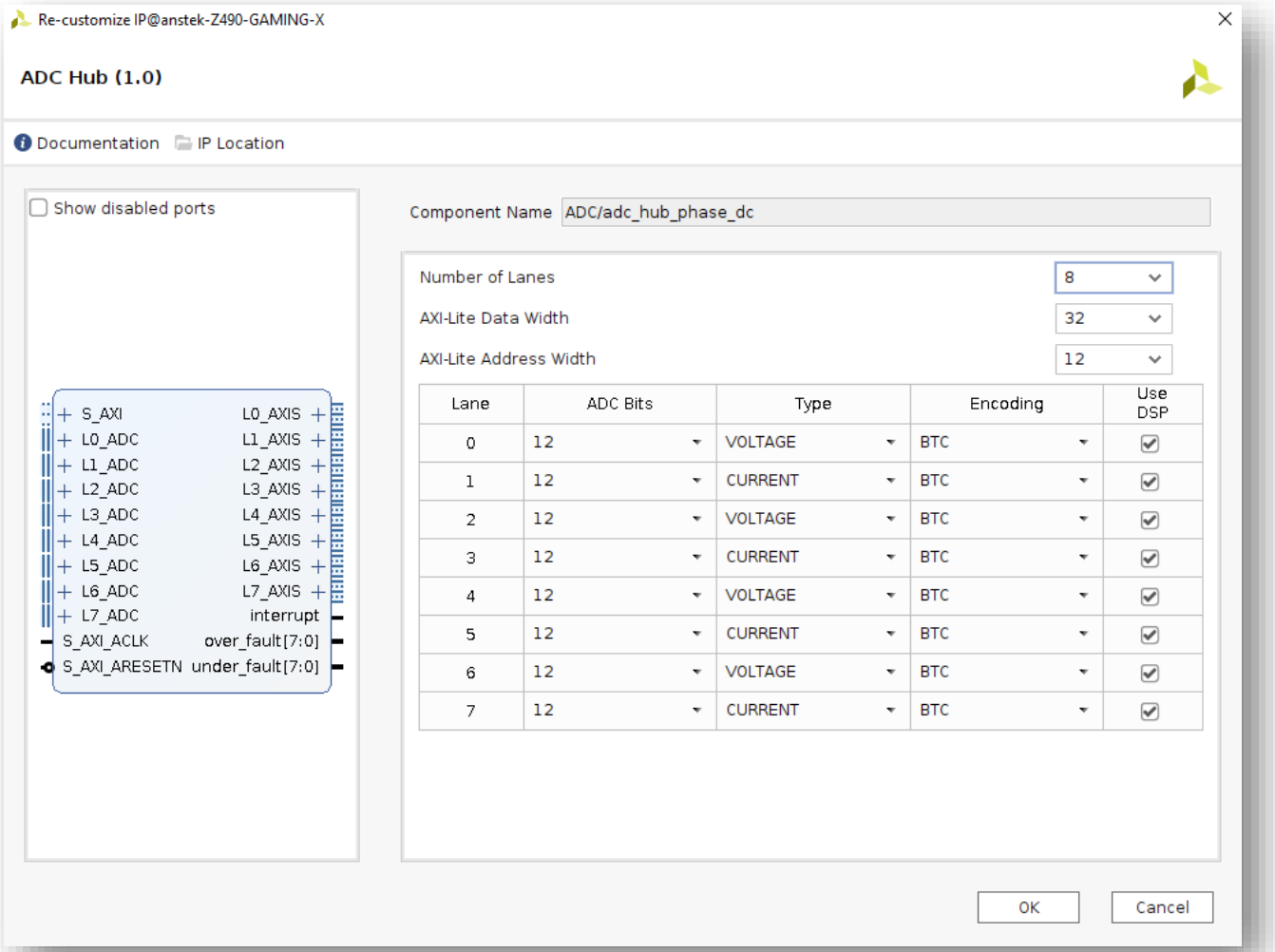


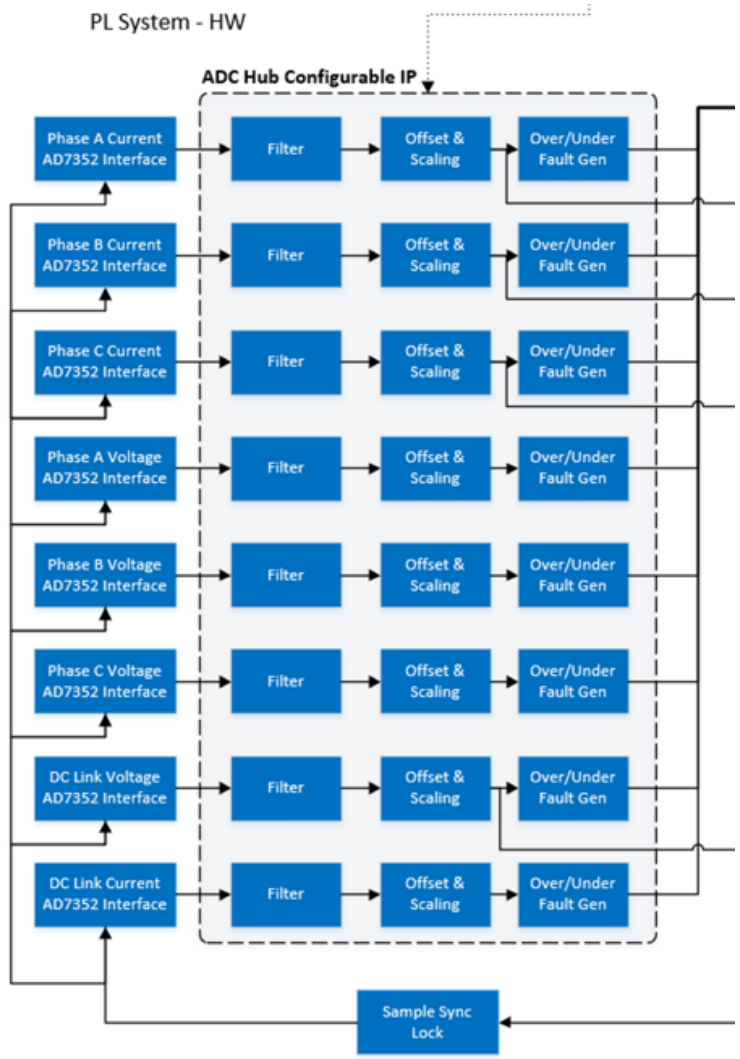
Sensored FOC Block Diagram



ADC Hub Configurable IP

The KD240 Motor Control Kit uses 8 channels, 12-bits, BTC encoding, and all channels are using the DSP for multipliers.(ADC Hub is parameterizable up to 16 channels)

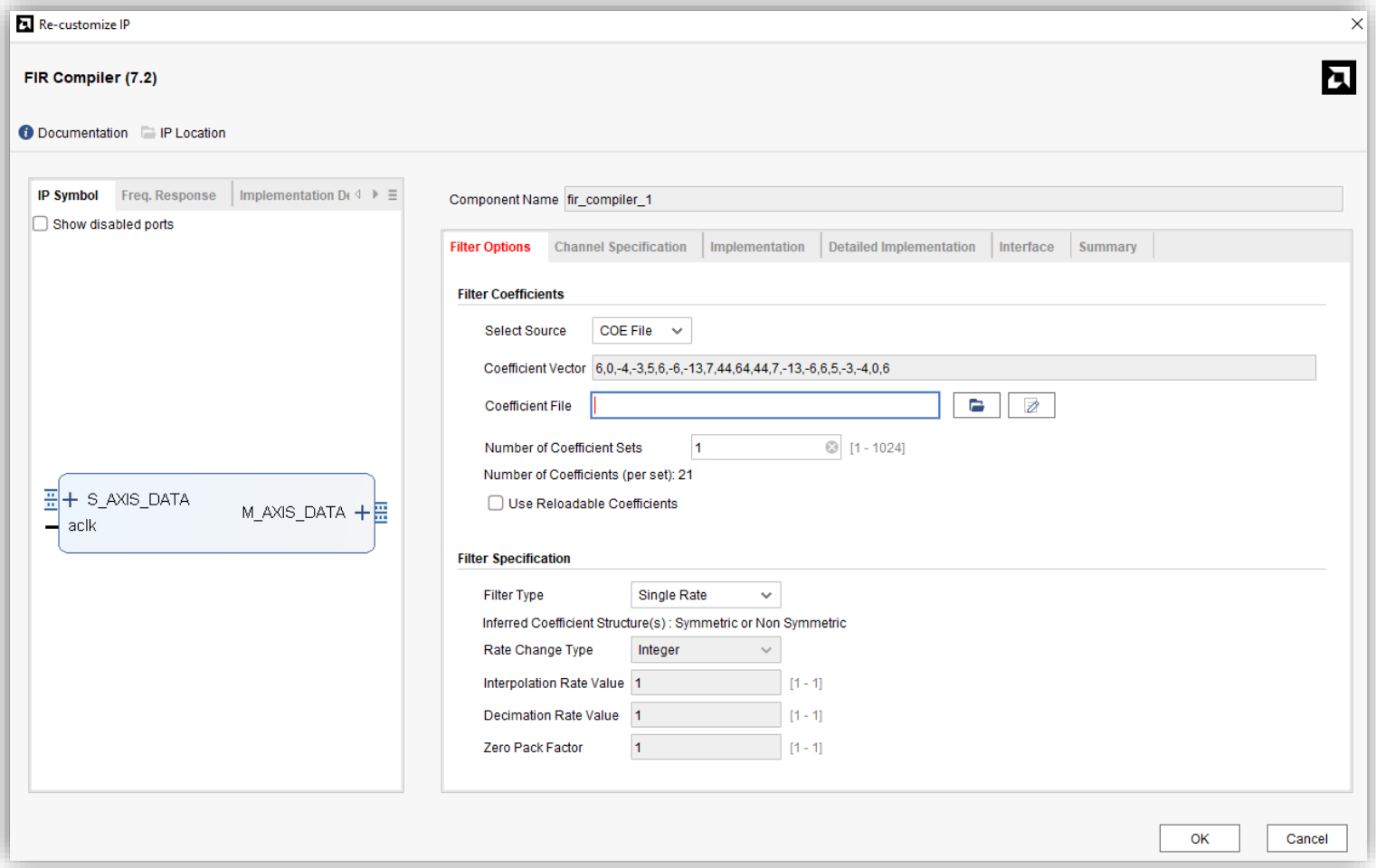


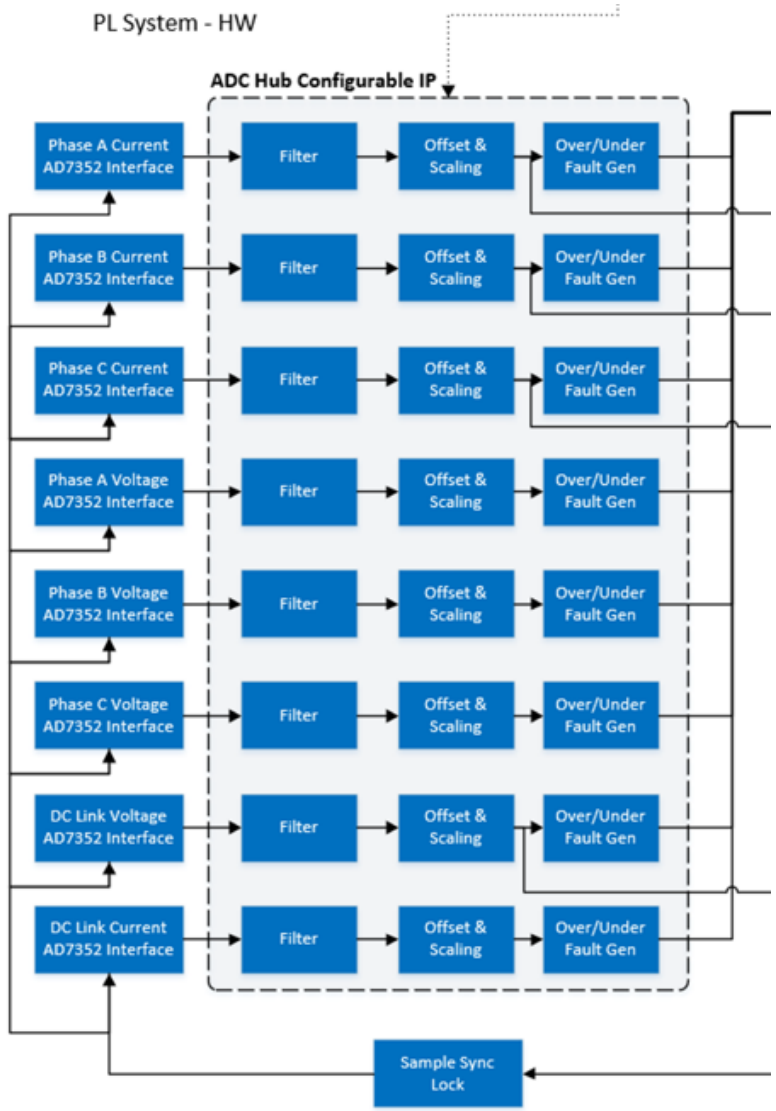


FIR Compiler IPs

For detailed instructions, please refer to the following URL

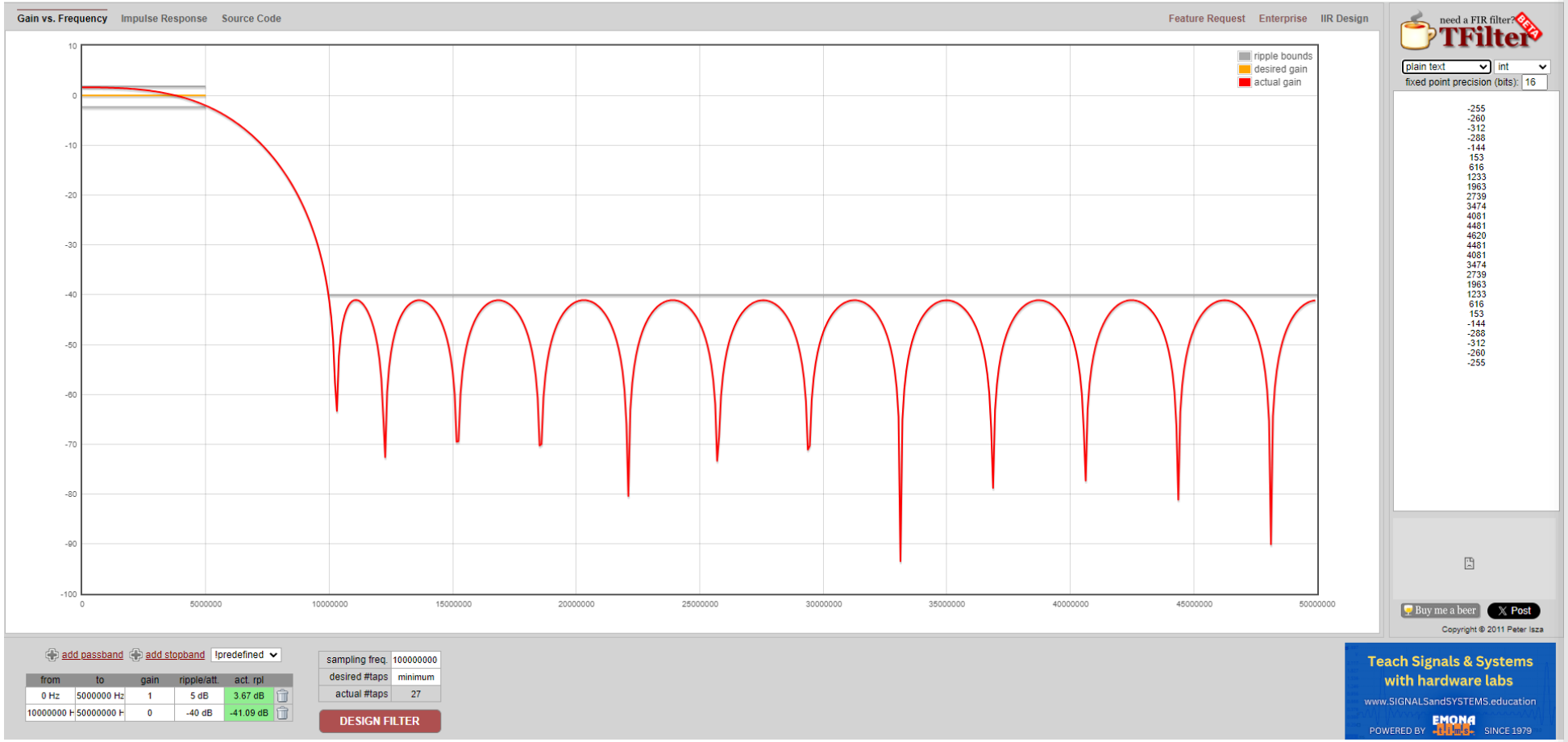
[DSP for SOMs: Getting Started with DDS & FIR Compiler IPs]





FIR Compiler IPs

Free Online FIR Filter Design

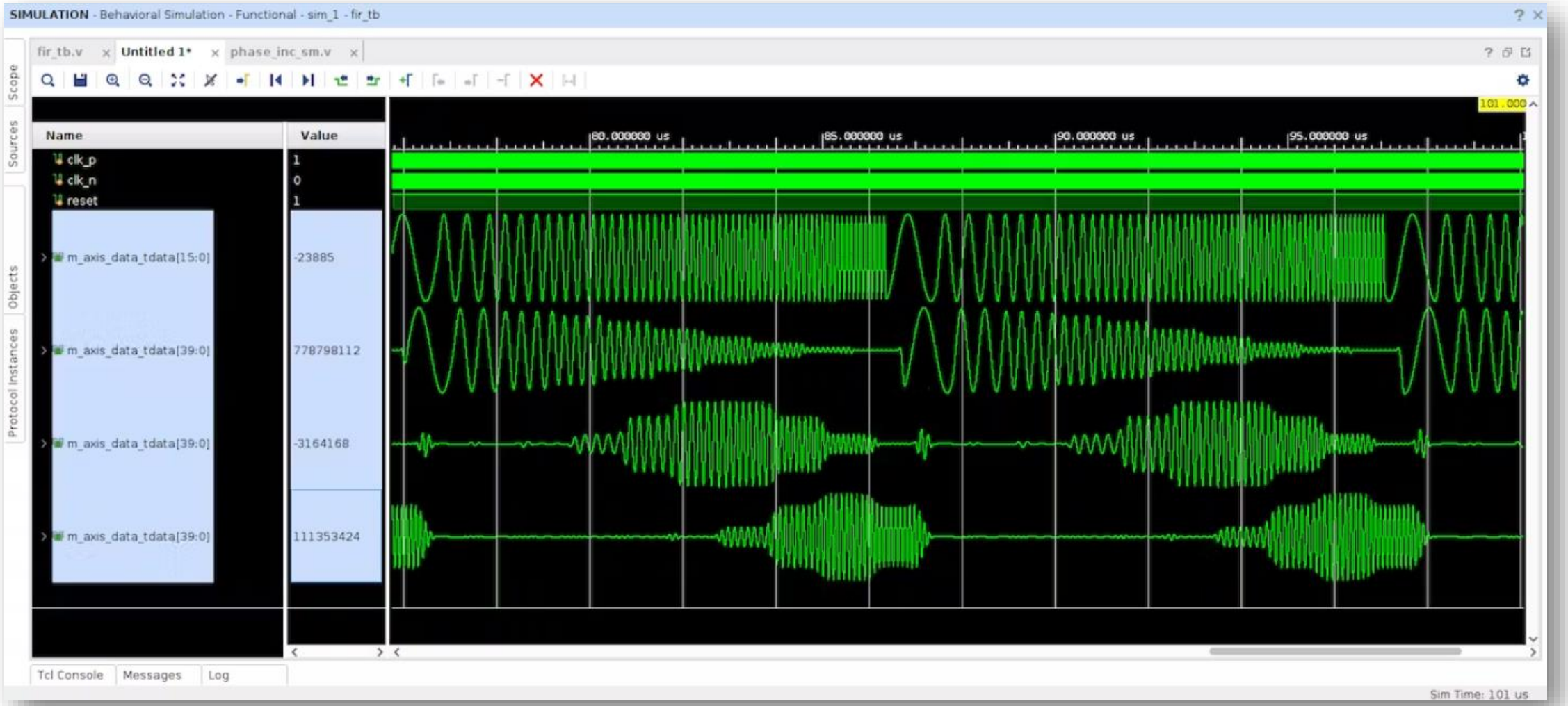


lpf_fir.coe

```
Radix=10;
CoeData=
-255,
-260,
-312,
-288,
-144,
153,
616,
1233,
1963,
2739,
3474,
4081,
4481,
4620,
4481,
4081,
3474,
2739,
1963,
1233,
616,
153,
-144,
-288,
-312,
-260,
-255
```

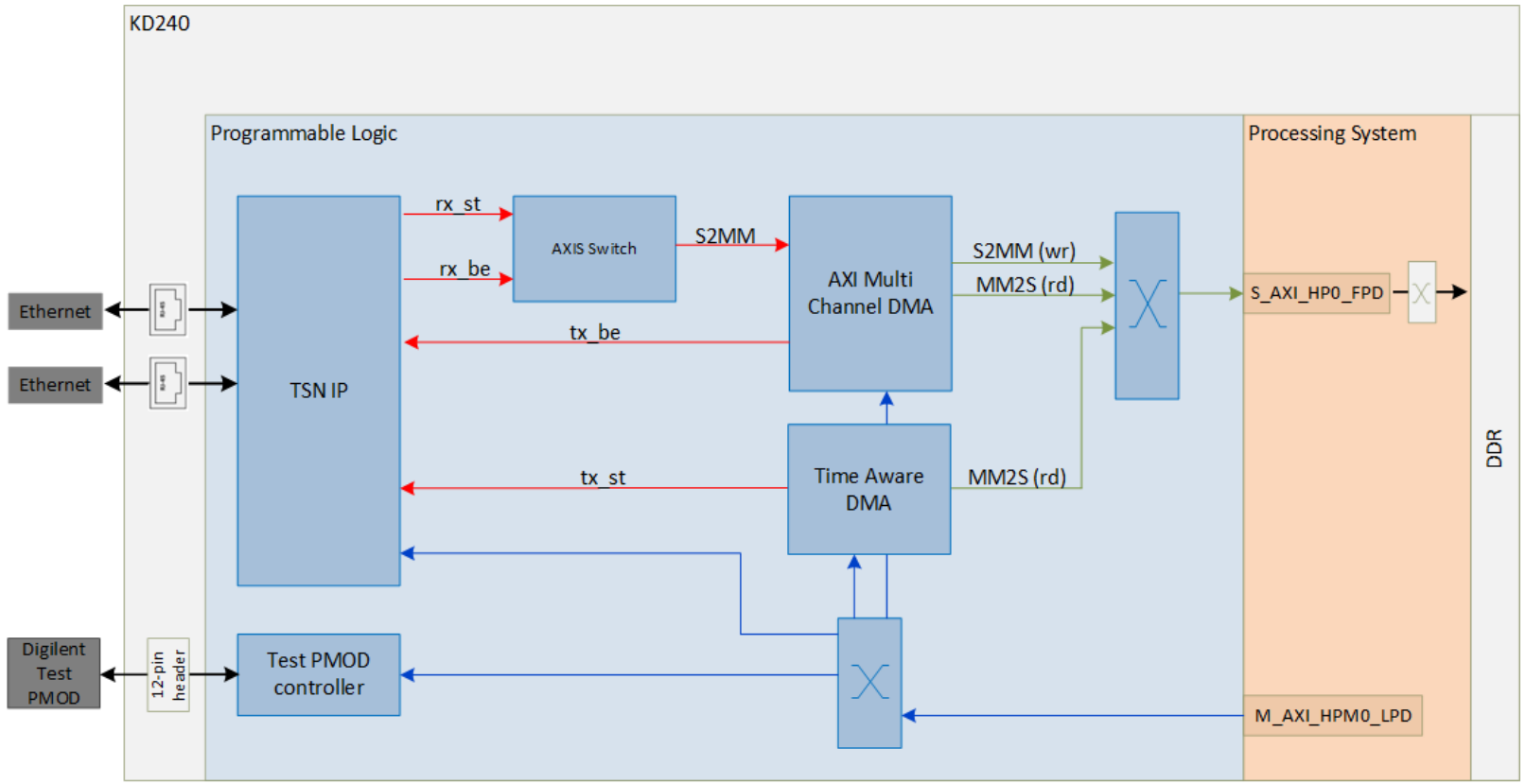
FIR Compiler IPs

Debugging with Simulation & ILA



TSN System

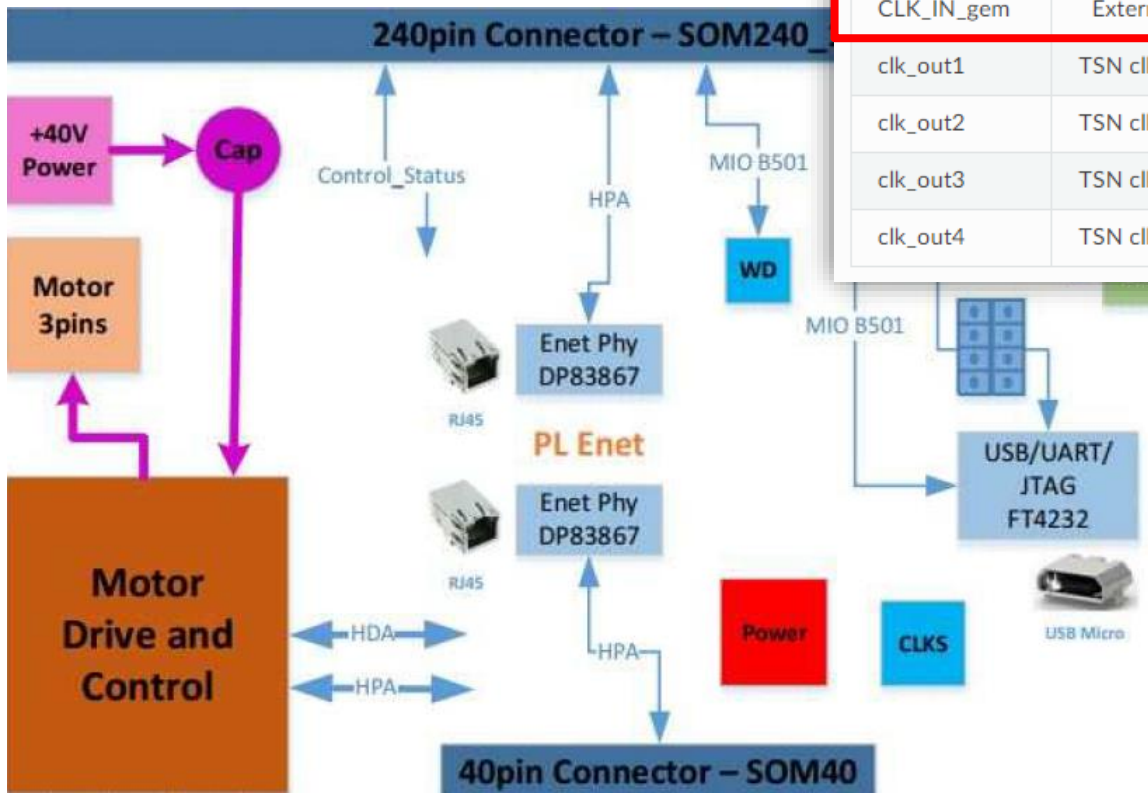
- Network Time Synchronization- 1588 Precision Time Protocol (PTP)
- Scheduled and Best Effort traffic types
- Time aware scheduling (IEEE 802.1 Qbv)
- Frame replication and elimination (IEEE 802.1 CB)
- Per Stream Filtering and Policing (IEEE 802.1 Qci)



Clocks

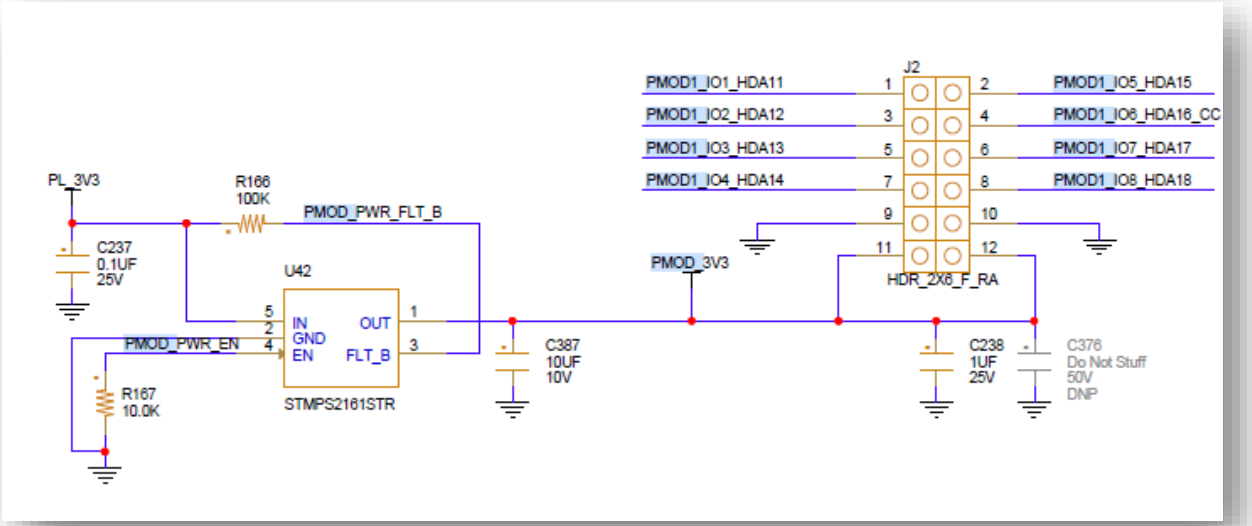
The following table identifies the main clocks of the PL design, their source, their clock frequency, and their function.

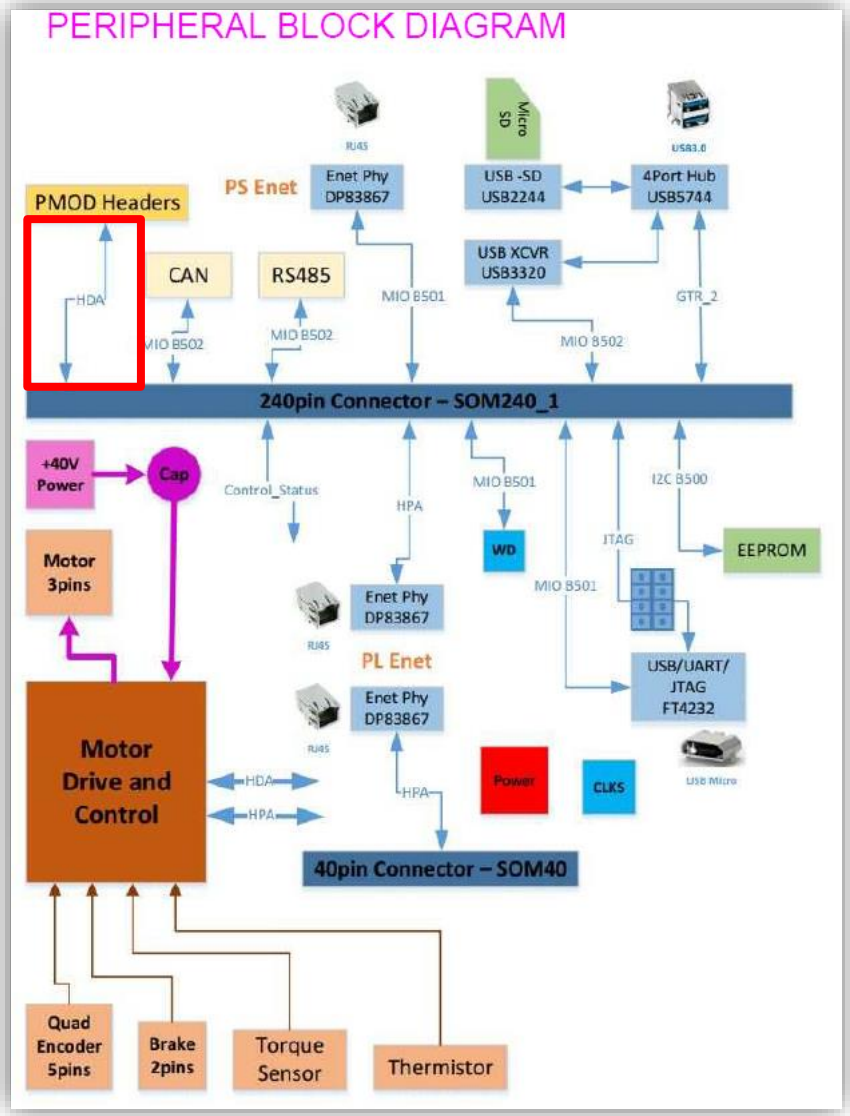
Clock	Clock Source	Clock Frequency	Function
pl_clk0	PS	100 MHz	Clock source for clocking wizard (clk_wiz_0)
clk_out_100M	clk_wiz_0	100 MHz	Clock for data on CIPS port M_AXI_HPM0_FPD and most of the system
clk_out_48M	clk_wiz_0	48 MHz	Clock for SCLK to ADC and ADC7352 Interface Module
CLK_IN_gem	External Pin	25 MHz	Clock for TSN Subsystem clocking wizard (clk_wiz_0) generating clocks
clk_out1	TSN clk_wiz_0	200 MHz	TSN IP fifo clock, TADMA IP and MCDMA IP data transactions clock
clk_out2	TSN clk_wiz_0	125 MHz	TSN IP and TADMA IP Real Time clock (RTC) for internal timers for time sensitivity
clk_out3	TSN clk_wiz_0	300 MHz	reference clock for IDELAY control block for PHY operation on TSN IP
clk_out4	TSN clk_wiz_0	100 MHz	AXI-Lite clock to configure TSN IP, MCDMA IP, and Register interface IP



PMOD Extension

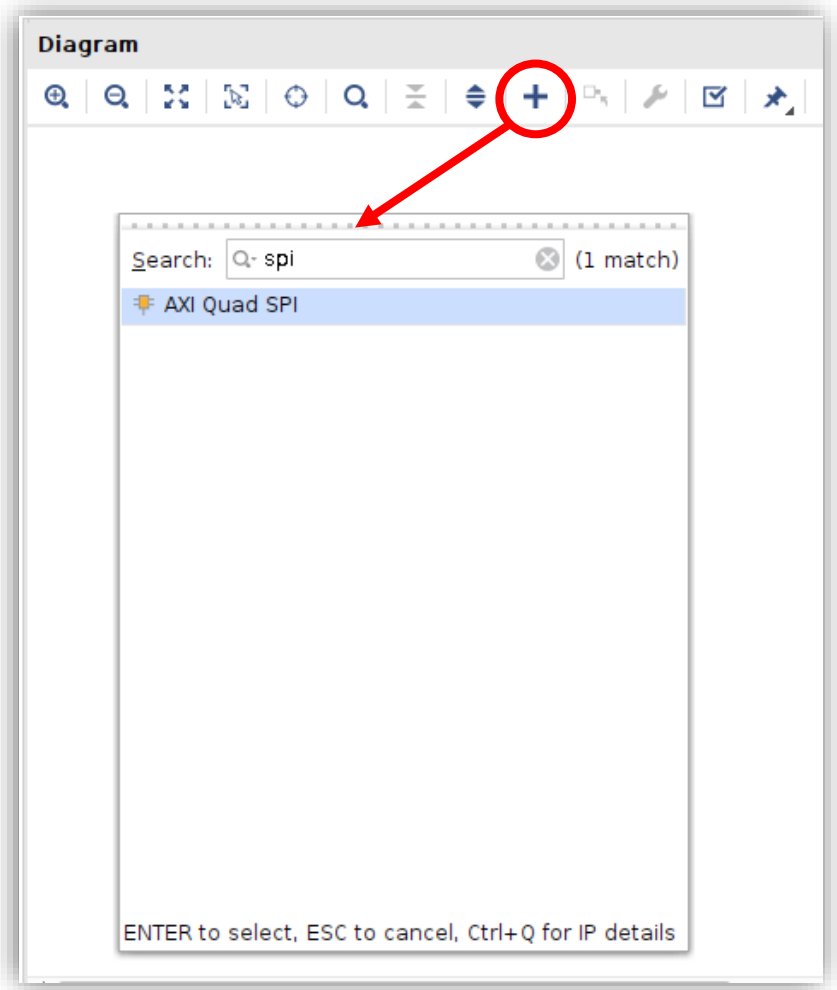
PMOD is not directly connected to the PS (Processing System), so if you want to use PMOD, you need to modify the hardware configuration yourself.

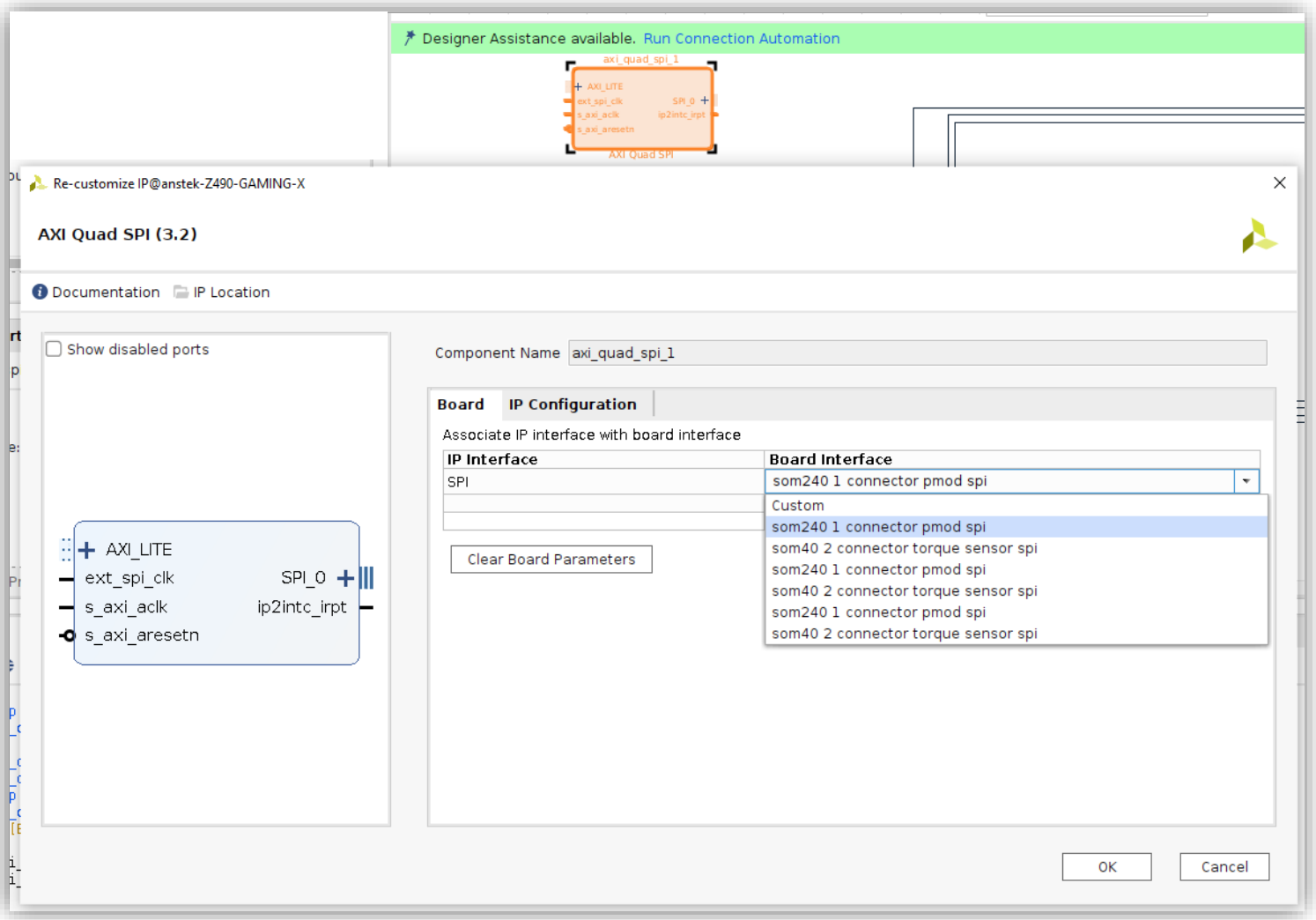




PMOD Extension

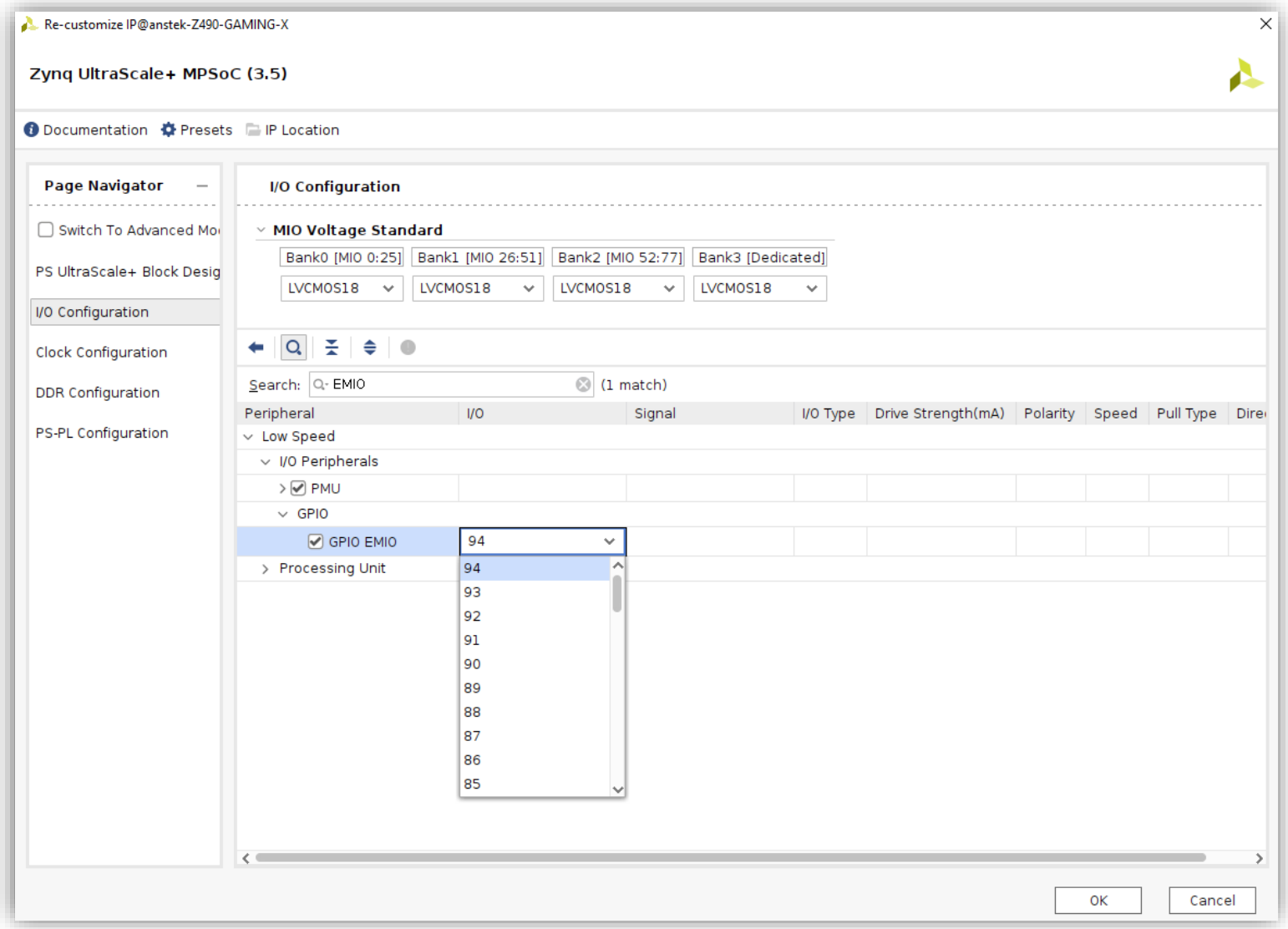
Here, using SPI as an example, you can directly establish a connection between the PMOD and SPI graphically.





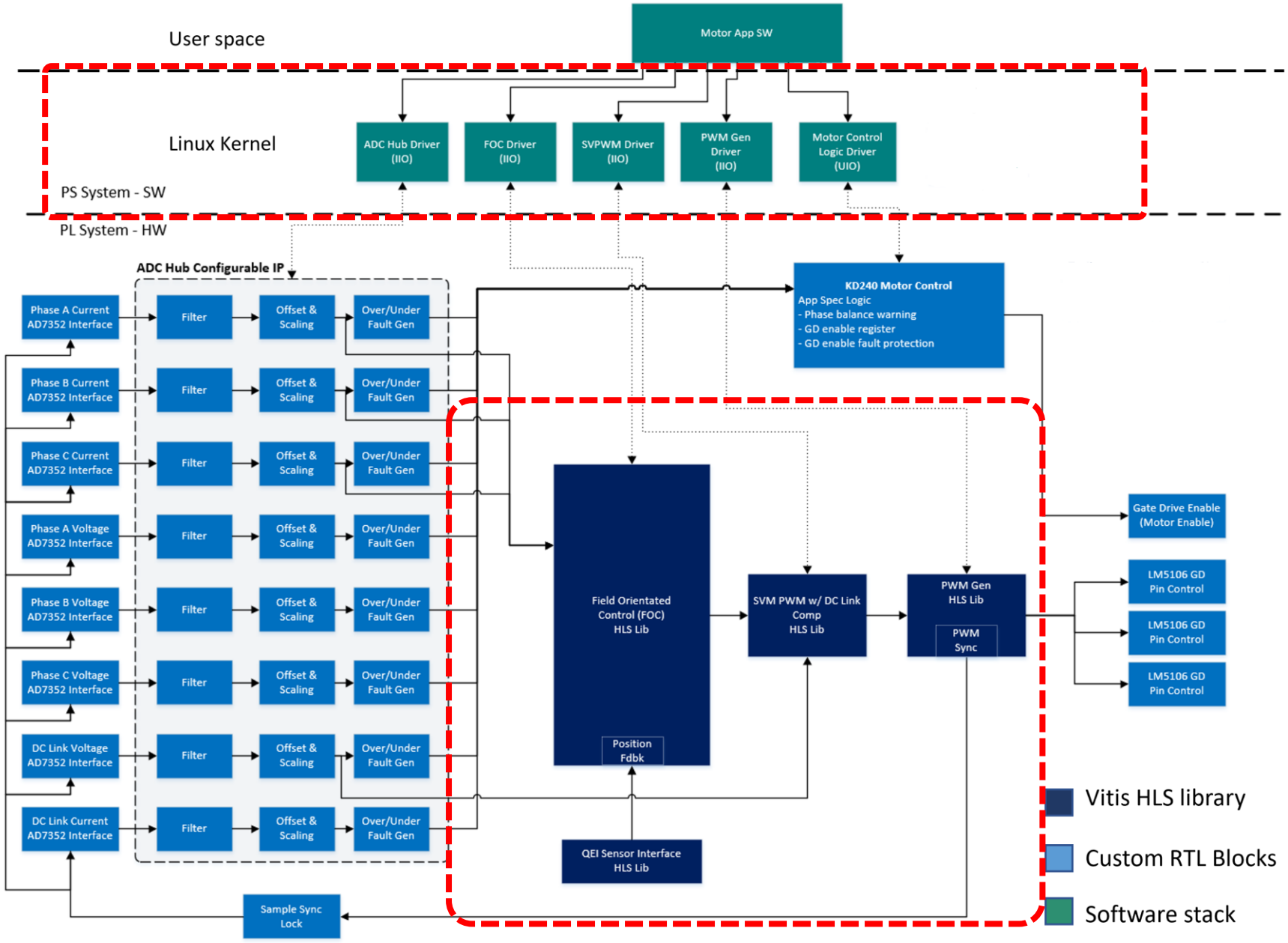
PMOD Extension

Alternatively, you can connect it to the PMOD via EMIO.



Vitis Library

Vitis Library



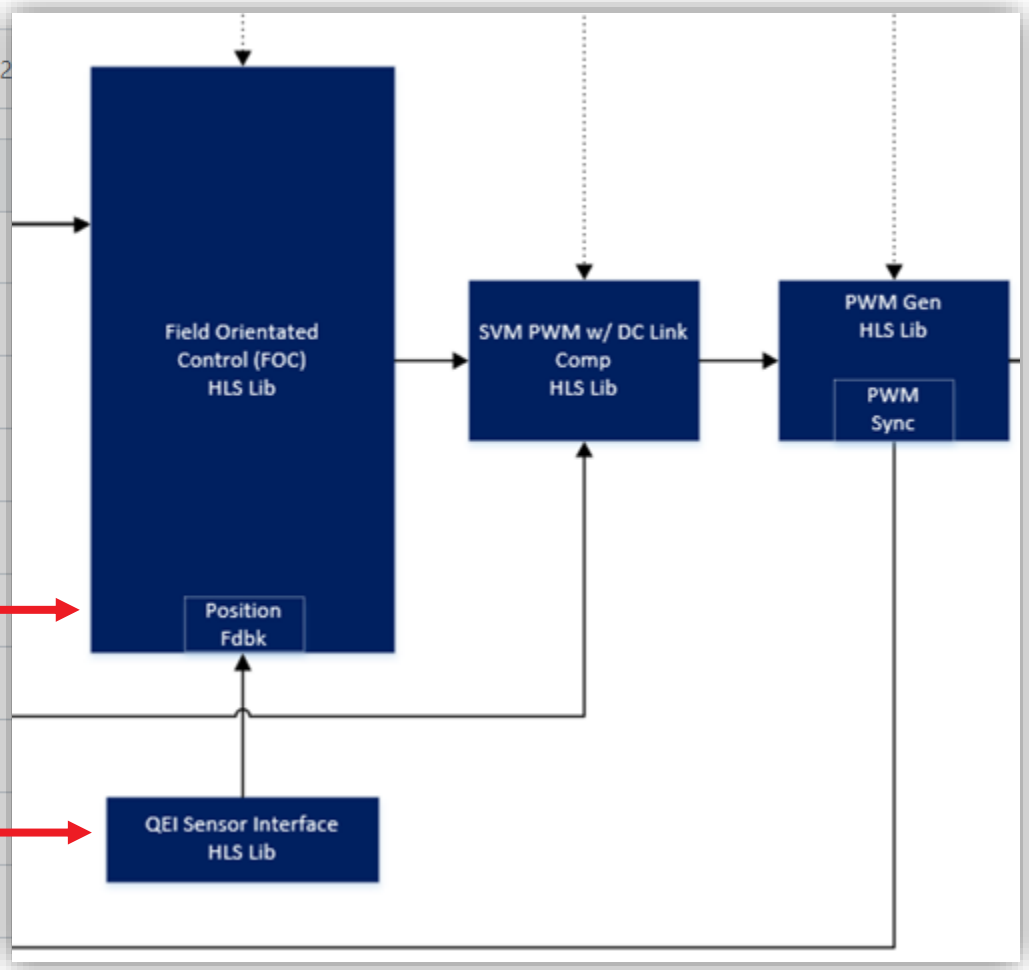
Vitis Library Source Code

You can check the HLS IP source code on GitHub.

main Vitis_Libraries / motor_control / L1 / include / hw /

2 people and GitHub Enterprise Squashed 'motor_control' changes from 88e5f89..ba7baa0 (#92

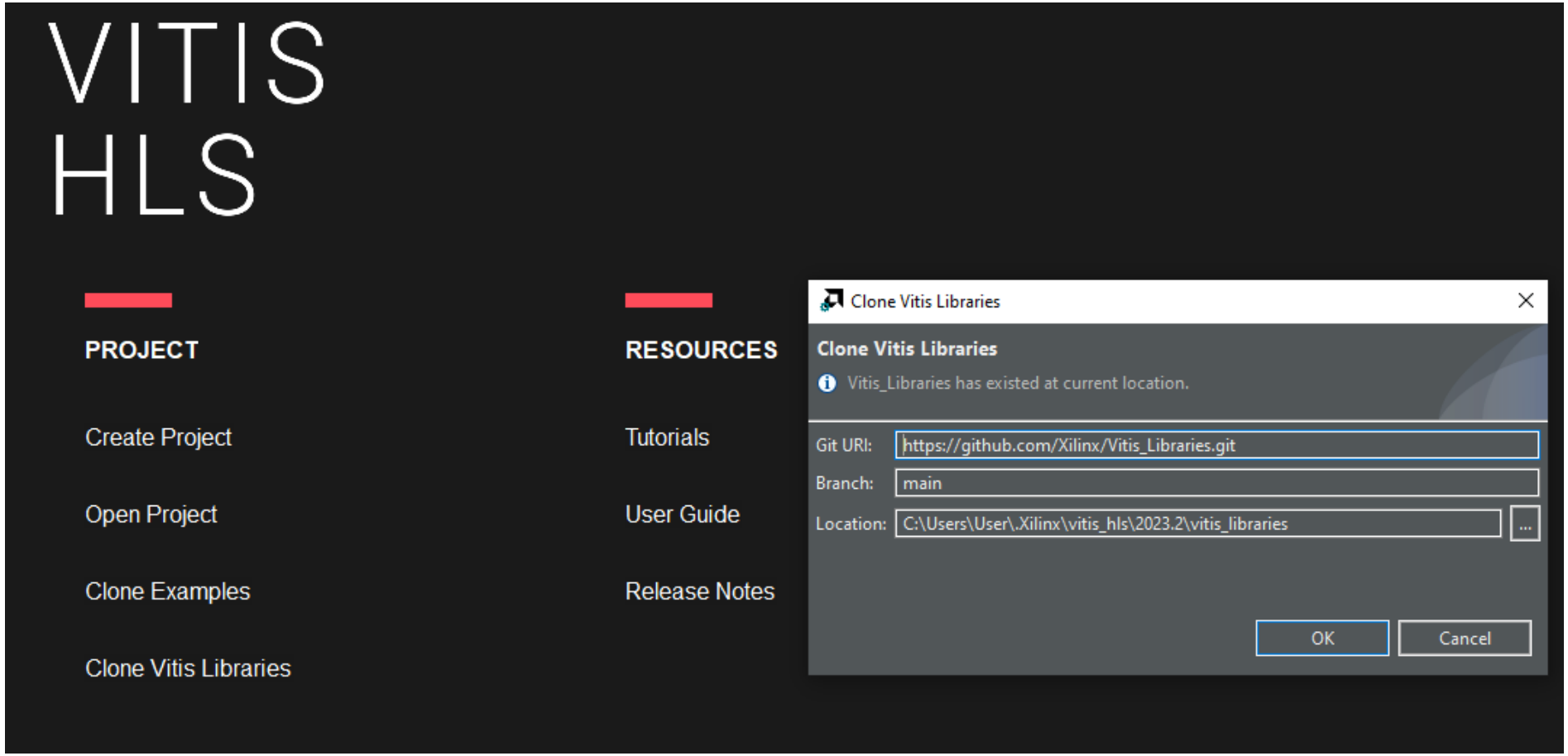
Name
..
clarke_2p.hpp
clarke_3p.hpp
common.hpp
field_weakening.hpp
foc.hpp
park.hpp
pid_control.hpp
qei.hpp
svpwm.hpp
utils.hpp



Verify HLS IP.

If you need to make modifications to the HLS IP, you can import it from the Vitis Library into HLS for verification.
Download Library from [GitHub - Xilinx/Vitis_Libraries: Vitis Libraries]

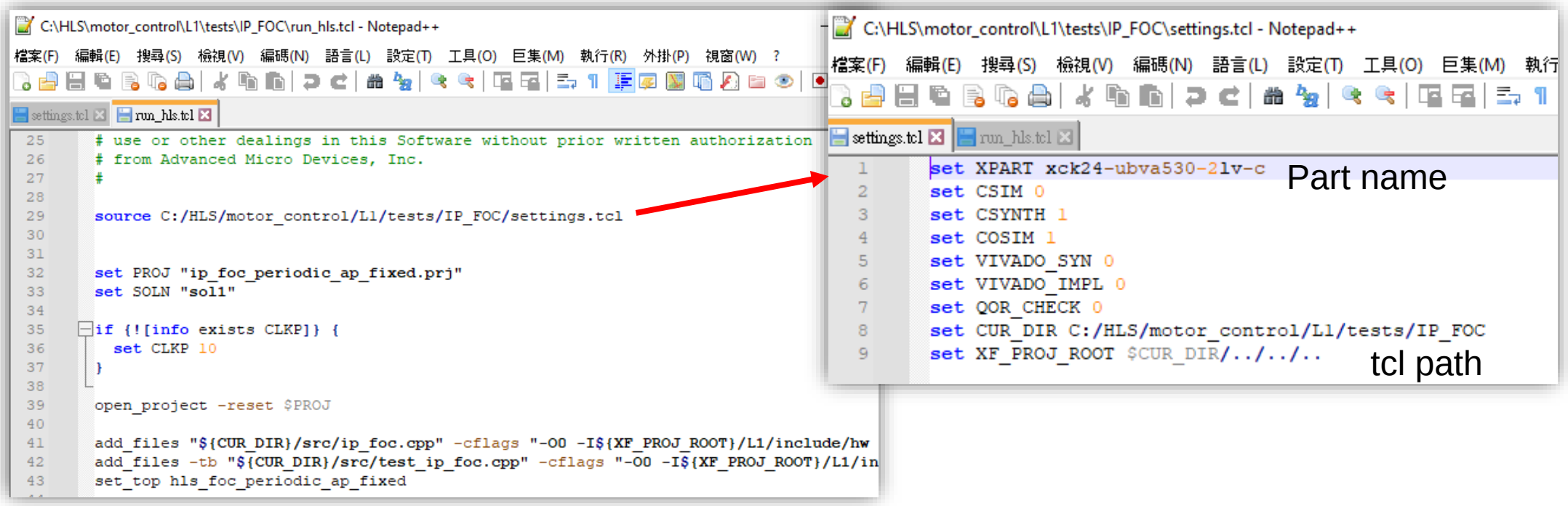
Or you can download the entire library via VITIS_HLS.



Verify HLS IP.

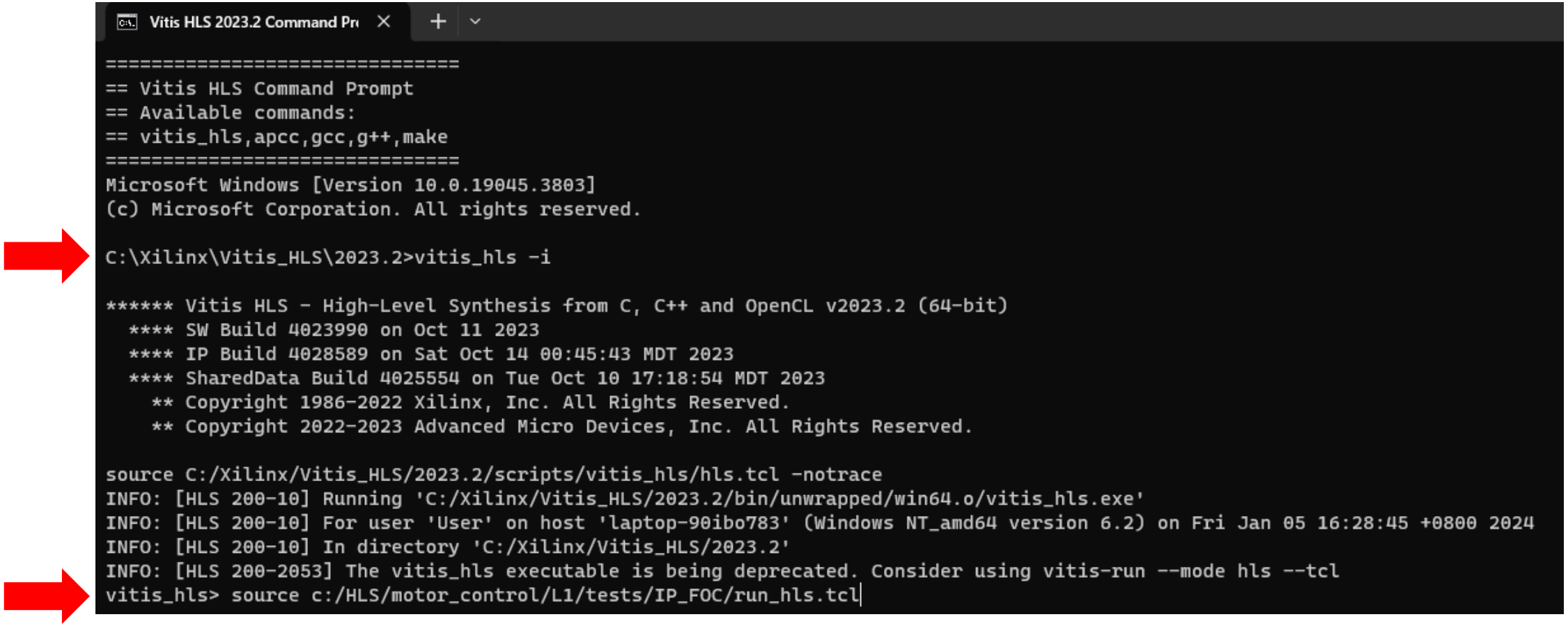
Modify the TCL files to generate HLS IP.

You need to create the settings.tcl file yourself.



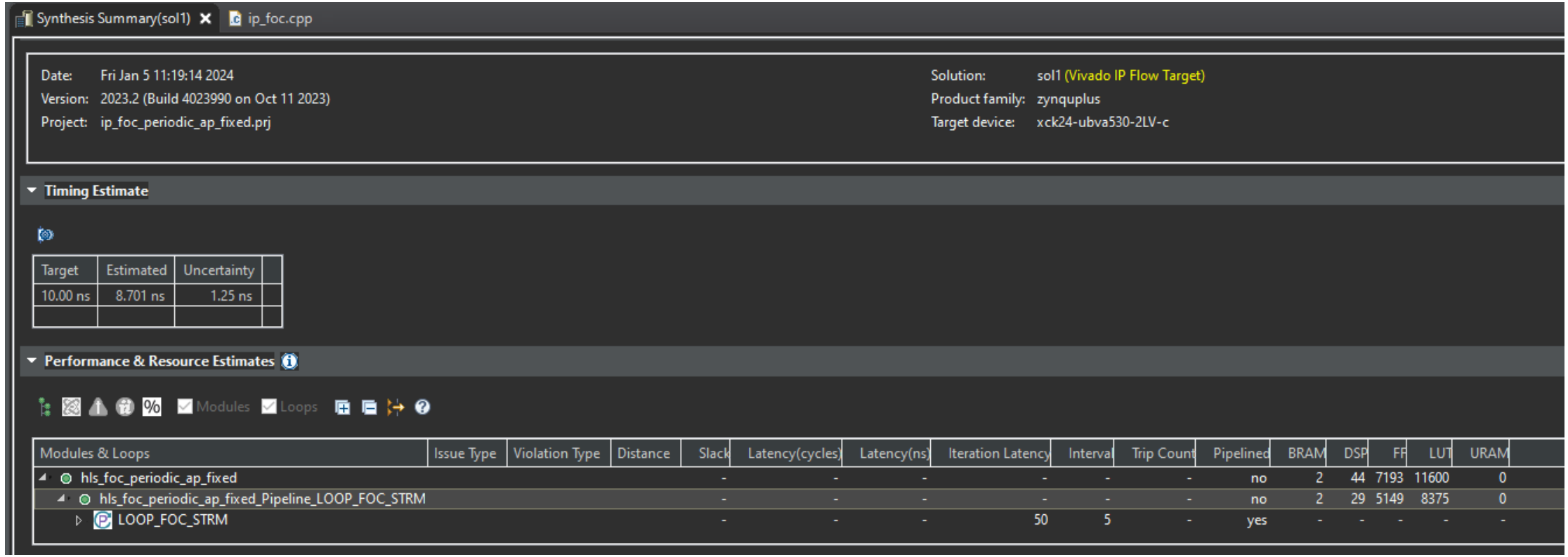
Verify HLS IP.

Open Vitis HLS Command Prompt and source run_hls.tcl



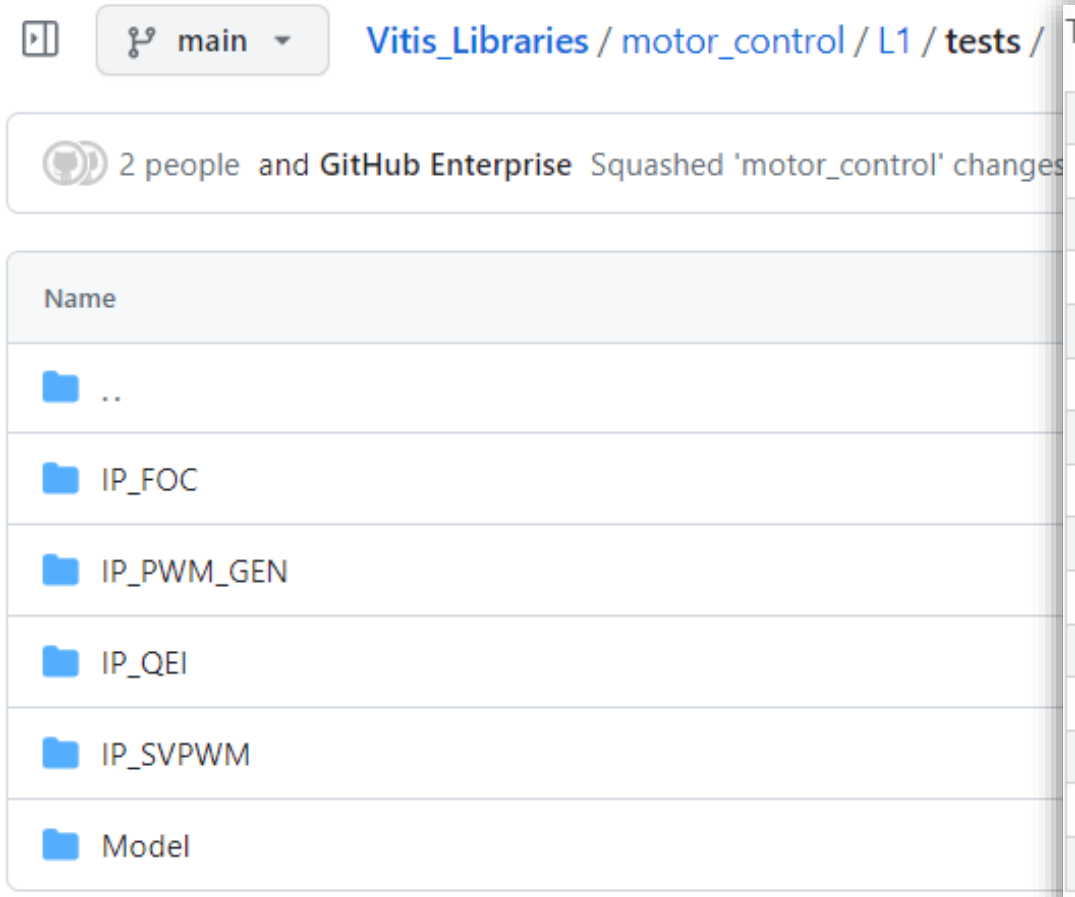
Verify HLS IP.

Use Vitis HLS to view the Performance & Resource report.



Using a virtual motor for simulation testing

These parameters could be set in the header file common.hpp in ./motor_control/L1/include/hw/ folder, the virtual motor model could be found in ./motor_control/L1/tests/Model/model_motor.hpp.

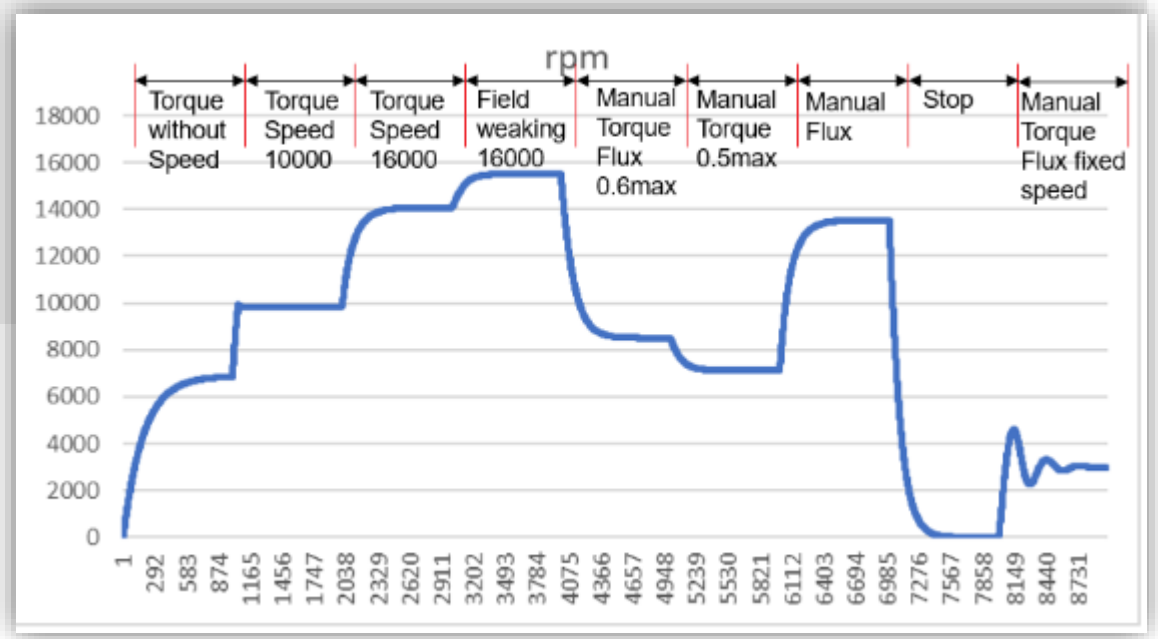


The datasheet below shows parameters used in motor model.

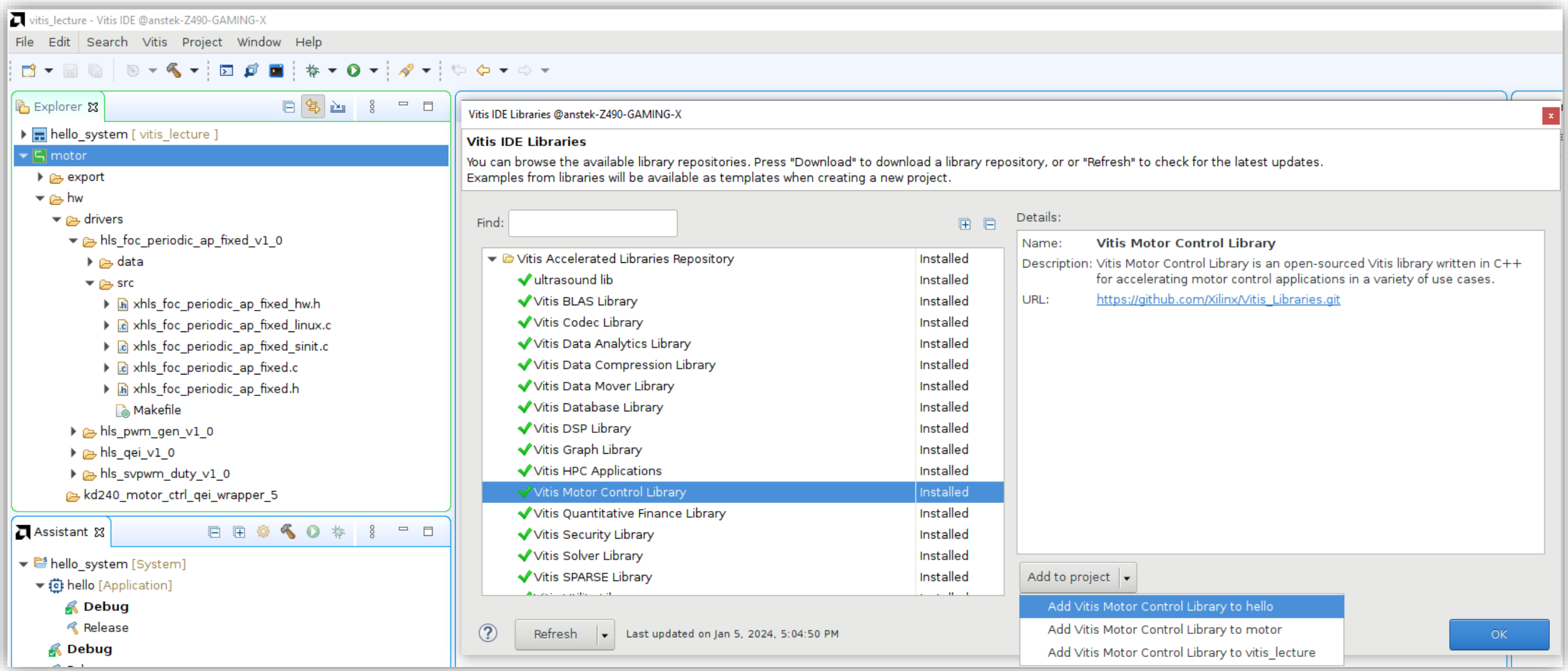
Parameter	Unit	Value
Torque Constant (Kt)	oz-in/A	2.27
Rotor Inertia (Ir)	oz-in-sec2	3.3x10^-5
Rated Voltage	V	24
Rated Torque	oz-in	2.0
Rated Speed	RPM	10,000
Rated Power	W	15
Rated Current	A	0.88
Poles pairs (N)	#	2
Permanent magnet flux linkage	A/Nm	0.008015
Peak Torque	oz-in	6.0
No Load Speed	RPM	12,800
Line-to-Line Resistance	Ω	4.63
Line-to-Line Inductance	mH	1.69
Ld = 0.5 * line-to-line inductance	mH	0.845

Simulation and verification of FOC_sensor

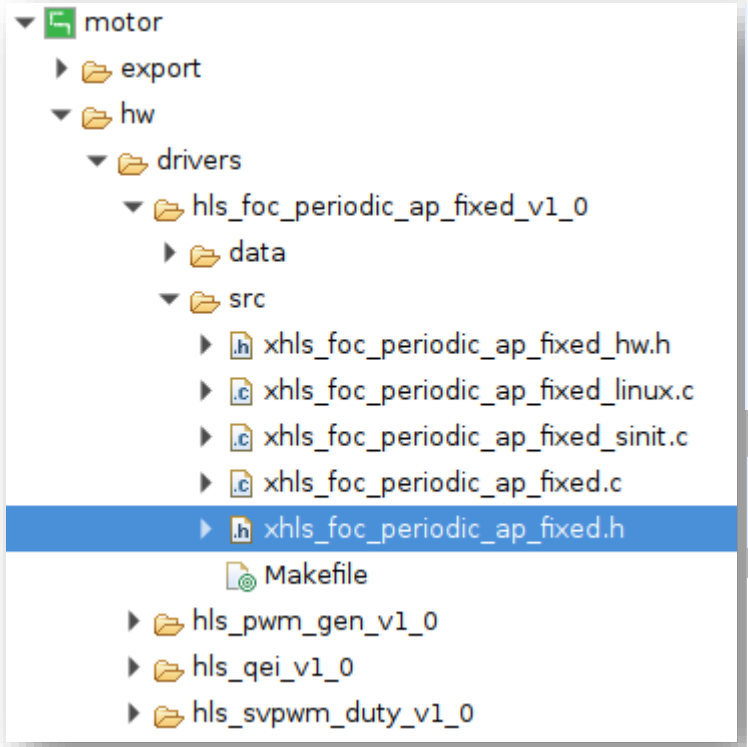
```
vim ./motor_control/L1/tests/IP_FOC/ip_foc_periodic_ap_fixed_sim.prj/sol1/csim/report/hls_foc_periodic_ap_fixed_csim.log

SIM_FOC_M:********** Simulation parameters ************ Motor parameter ******************* Log files************************************
SIM_FOC_M:  Timescale  :     10 (us)    |  motor.w     : 718.0615      (rad/s) |  Log of parameters : sim_torqueWithoutSpeed.para.foc
SIM_FOC_M:  Total step :   3000         |  motor.theta :  5.014019     (rad)   |  Log of FOC inputs : sim_torqueWithoutSpeed.in.foc
SIM_FOC_M:  Total time : 0.030000 (s)   |  motor.Id    : 0.334735      ( A )   |  Log of FOC outputs: sim_torqueWithoutSpeed.out.foc
SIM_FOC_M:  Inteval    :      3         |  motor.Iq    : 1.253668      ( A )
SIM_FOC_M:  FOC MODE   : MOD_TORQUE_WITHOUT_SPEED
SIM_FOC_M:  FOC CPR    :   1000         |  FOC PPR:      2
SIM_FOC_M:************ PID Final Status *********
SIM_FOC_M:  SPEED SP   : 10000.0        |  FLUX SP: 0.0000
SIM_FOC_M:  SPEED KP   : 2.7000         |  FLUX KP: 1.0000
SIM_FOC_M:  SPEED KI   : 0.0033         |  FLUX KI: 0.0000
SIM_FOC_M:  SPEED ERR  : 3144.000       |  FLUX ERR:  -0.332
SIM_FOC_M:  SPEED ACC  : 23853.000      |  FLUX ACC: -862.297
```



Motor Control IP Driver



Motor Control IP Driver




```
int XHls_foc_periodic_ap_fixed_Initialize(XHls_foc_periodic_ap_fixed *InstancePtr, u16 DeviceId);
XHls_foc_periodic_ap_fixed_Config* XHls_foc_periodic_ap_fixed_LookupConfig(u16 DeviceId);
int XHls_foc_periodic_ap_fixed_CfgInitialize(XHls_foc_periodic_ap_fixed *InstancePtr, XHls_foc_per
#else
int XHls_foc_periodic_ap_fixed_Initialize(XHls_foc_periodic_ap_fixed *InstancePtr, const char* Ins
int XHls_foc_periodic_ap_fixed_Release(XHls_foc_periodic_ap_fixed *InstancePtr);
#endif

void XHls_foc_periodic_ap_fixed_Start(XHls_foc_periodic_ap_fixed *InstancePtr);
u32 XHls_foc_periodic_ap_fixed_IsDone(XHls_foc_periodic_ap_fixed *InstancePtr);
u32 XHls_foc_periodic_ap_fixed_IsIdle(XHls_foc_periodic_ap_fixed *InstancePtr);
u32 XHls_foc_periodic_ap_fixed_IsReady(XHls_foc_periodic_ap_fixed *InstancePtr);
```

```
void XHls_foc_periodic_ap_fixed_Set_control_mode_args
u32 XHls_foc_periodic_ap_fixed_Get_control_mode_args(
```

```
u32 XHls_foc_periodic_ap_fixed_Get_torque_ki_args(X
void XHls_foc_periodic_ap_fixed_Set_torque_kd_args(
u32 XHls_foc_periodic_ap_fixed_Get_torque_kd_args(X
void XHls_foc_periodic_ap_fixed_Set_speed_sp_args(X
u32 XHls_foc_periodic_ap_fixed_Get_speed_sp_args(XH
void XHls_foc_periodic_ap_fixed_Set_speed_kp_args(X
u32 XHls_foc_periodic_ap_fixed_Get_speed_kp_args(XH
```

```
void XHls_foc_periodic_ap_fixed_InterruptGlobalEnable(X
void XHls_foc_periodic_ap_fixed_InterruptGlobalDisable(
```


AMD