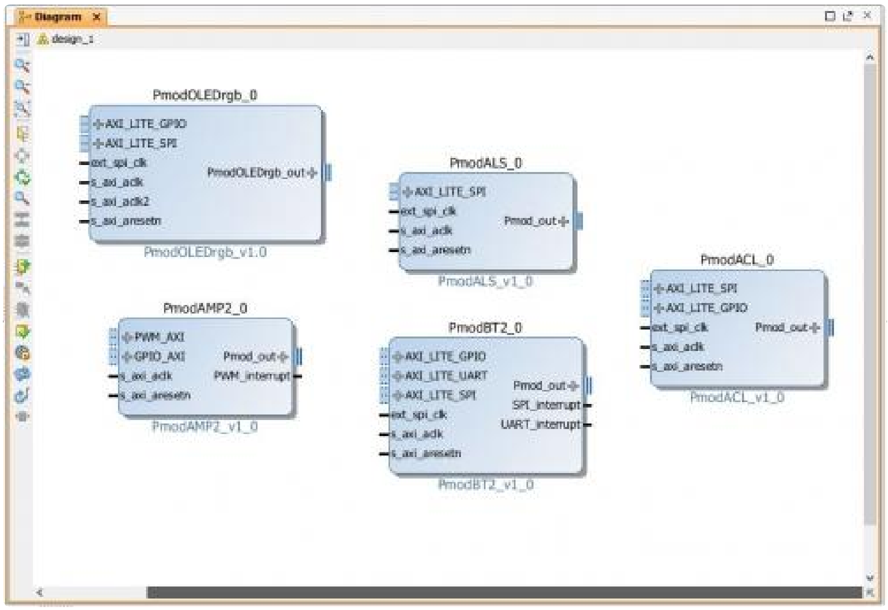
Getting Started with Digilent Pmod IPs



(https://reference.digilentinc.com/_media/vivado/pmods.jpg)

Overview

Digilent provides several IPs that are designed to make implementing and using a Pmod on an FPGA as straightforward as possible. This guide will describe how to use a Pmod IP core in Vivado Microblaze or Zynq design.

At the end of this tutorial you will have a Vivado design and demo for your FPGA or Zynq platform that uses a Digilent Pmod IP core.

The following two dropdown tables show which Digilent FPGA system boards and Pmods are supported by this tutorial, as well as some details about each one that you will need to know to complete this tutorial.

Platforms Supported

Platform	Processor Type
Arty A7 (https://reference.digilentinc.com/reference/programmable-logic/arty-a7/start)	Microblaze
Arty S7 (https://reference.digilentinc.com/reference/programmable-logic/arty-s7/start)	Microblaze
Arty Z7 (https://reference.digilentinc.com/reference/programmable-logic/arty-z7/start)	Zynq
Basys3 (https://reference.digilentinc.com/reference/programmable-logic/basys-3/start)	Microblaze
Cmod A7 (https://reference.digilentinc.com/reference/programmable-logic/cmod-a7/start)	Microblaze
Cmod S7 (https://reference.digilentinc.com/reference/programmable-logic/cmod-s7/start)	Microblaze
Cora Z7 (https://reference.digilentinc.com/reference/programmable-logic/cora-z7/start)	Zynq
Genesys2 (https://reference.digilentinc.com/reference/programmable-logic/genesys-2/start)	Microblaze
Nexys4 (https://reference.digilentinc.com/reference/programmable-logic/nexys-4/start)	Microblaze
Nexys4-DDR (https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/start)	Microblaze
Nexys Video (https://reference.digilentinc.com/reference/programmable-logic/nexys-video/start)	Microblaze
Sword (https://reference.digilentinc.com/reference/programmable-logic/sword/start)	Microblaze
Zybo (https://reference.digilentinc.com/reference/programmable-logic/zybo/start)	Zynq
Zybo Z7 (https://reference.digilentinc.com/reference/programmable-logic/zybo-z7/start)	Zynq

Pmods Supported

Pmod	Interface Type	Reference clock frequency (MHz ())	Reference Clock signal name	Interrupt pin name/s	Uses PmodGPIO IP
8LD	GPIO ()	-	-	-	Yes
ACL ()	SPI	80	ext_spi_clk	-	-
ACL2	SPI	50	ext_spi_clk	-	-
AD1	SPI	-	-	-	-
AD2	IIC	-	-	-	-
AD5	SPI	50	-	-	-
ALS	SPI	50	ext_spi_clk	-	-
AMP2	GPIO ()	-	-	timer_interrupt	-
BB ()	GPIO ()	-	-	-	Yes
BLE	UART	-	-	-	-
BT2	UART	-	-	-	-
BTN	GPIO ()	-	-	-	Yes
CAN	SPI	100	ext_spi_clk	SPI_interrupt, GPIO ()_interrupt	-
CLS	SPI	50	ext_spi_clk	-	-
CMPS2	IIC	-	-	-	-
COLOR	IIC	-	-	-	-
DA1	SPI	50	ext_spi_clk	-	-
DHB1	PWM/GPIO ()	-	-	-	-
DPG1	SPI	50	ext_spi_clk	-	-
ENC	GPIO ()	-	-	-	-
ESP32	UART	-	-	-	-
GPS ()	UART	-	-	gps_uart_interrupt	-
GYRO	SPI	50	ext_spi_clk	-	-
HYGRO	IIC	-	-	-	-
JSTK	SPI	16	ext_spi_clk	-	-
JSTK2	SPI	16	ext_spi_clk	-	-
KYPD	GPIO ()	-	-	-	-

Pmod	Interface Type	Reference clock frequency (MHz ())	Reference Clock signal name	Interrupt pin name/s	Uses PmodGPIO IP
LED ()	GPIO ()	-	-	-	Yes
MAXSONAR	GPIO ()	-	-	-	-
MicroSD	Use the Pmod SD IP core				
MTDS	SPI	-	-	-	-
NAV	SPI/GPIO ()	50	ext_spi_clk	-	-
OLED ()	SPI/GPIO ()	-	-	-	-
OLEDrgb	SPI/GPIO ()	50	ext_spi_clk	-	-
R2R	GPIO ()	-	-	-	-
RTCC ()	IIC	-	-	-	-
SD	SPI	-	-	-	-
SF3	SPI	50	ext_spi_clk	QSPI_INTERRUPT	-
SSR	GPIO ()	-	-	-	Yes
SWT	GPIO ()	-	-	-	Yes
TC1	SPI	50	ext_spi_clk	-	-
TMP3	IIC	-	-	-	-
WIFI	SPI	-	-	WF_INTERRUPT	-

Prerequisites

Hardware

- **Supported Digilent 7-Series FPGA System Board**
- **MicroUSB Cable/s**
- **One or More Supported Digilent Pmods**

Software

- **Xilinx Vivado 2018.2 with Xilinx SDK and Digilent Board Files**
  - *Other versions of Vivado may work, but functionality is not guaranteed*
  - *See the "Installing Vivado and Digilent Board Files" (https://reference.digilentinc.com/vivado/installing-vivado/2018.2) tutorial for more information.*
- **Digilent Vivado IP Library**
  - *Step 2 of this tutorial covers how to download and extract these files.*

Important

If the Pmod IP to be used has a README file, be sure to review it before starting this tutorial. This file can be found in the **vivado-library/ip/Pmods/"your pmod"** directory.

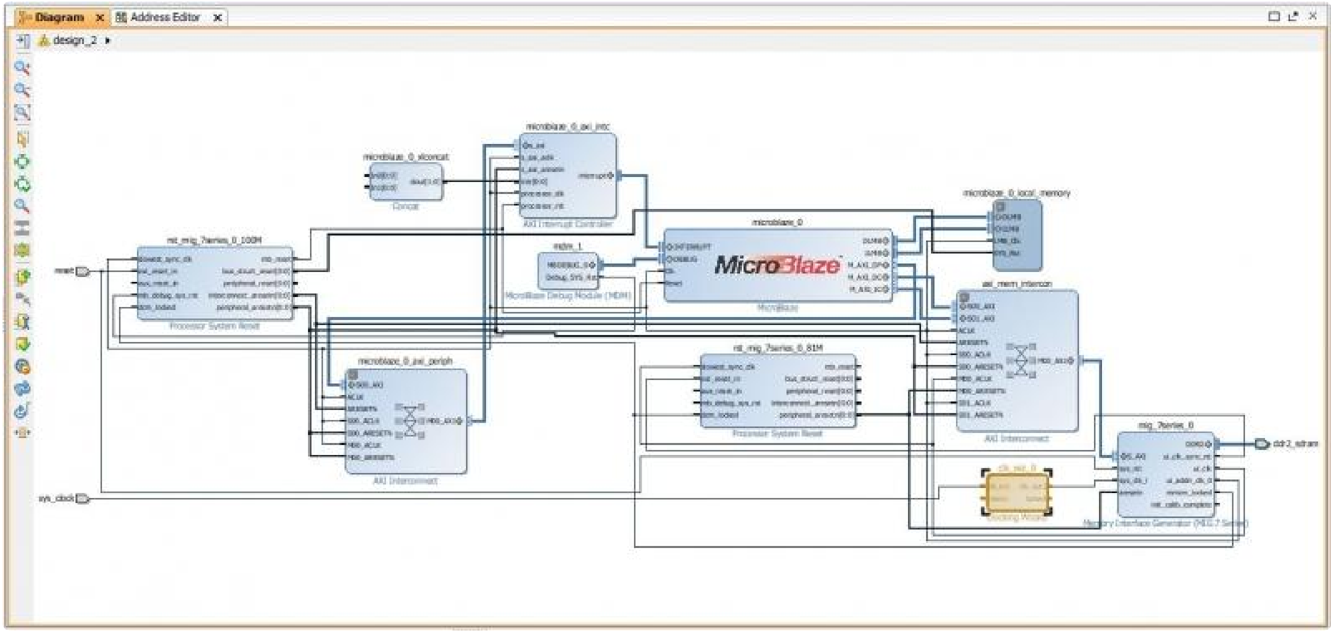
Tutorial

1. Create a New Microblaze/Zynq Block Design

To determine whether you need to use Microblaze or Zynq for this tutorial, refer to the entry in the **Platforms Supported** dropdown table found in the Overview Section of this tutorial. Alternatively, navigate to the resource center for your platform here (https://reference.digilentinc.com/reference/programmable-logic/start).

Microblaze

Follow the Getting Started with Vivado IP Integrator (https://reference.digilentinc.com/vivado/getting-started-with-ipi/2018.2) tutorial to obtain a basic MicroBlaze block design.

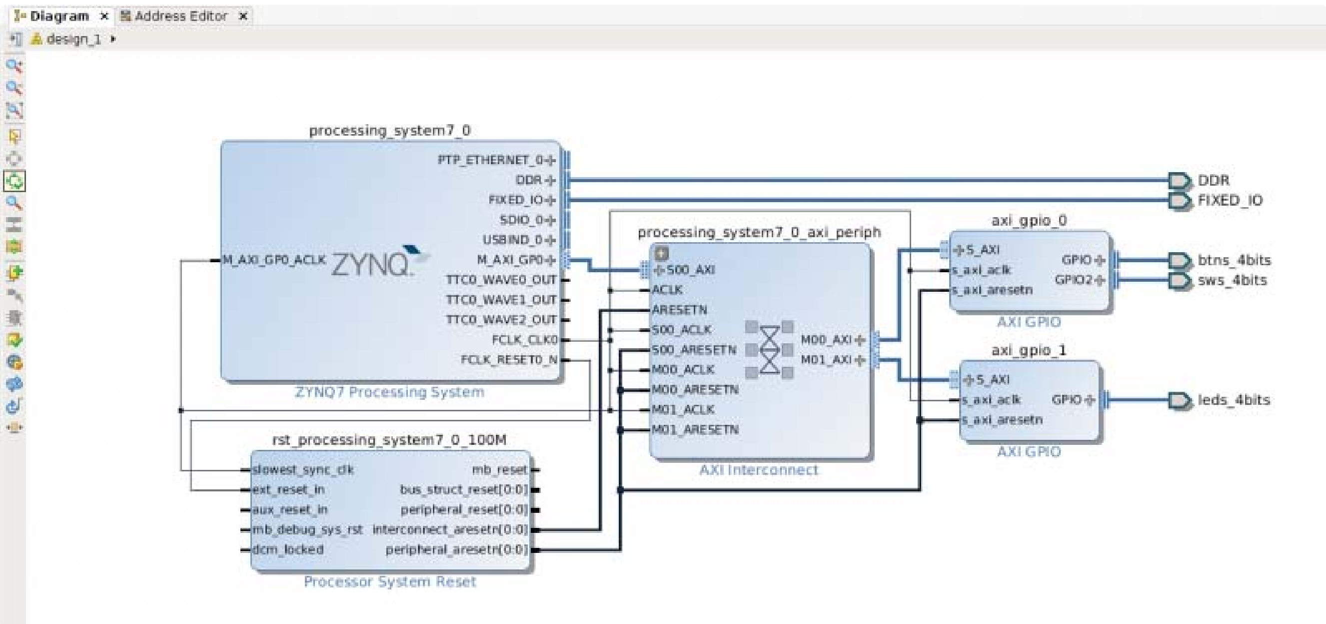


(https://reference.digilentinc.com/_media/vivado/basedesign.jpg)

An example of a MicroBlaze block design for a board that has external DDR memory.

Zynq

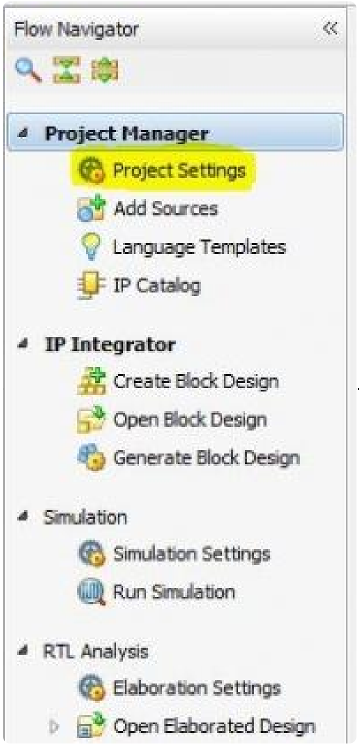
Follow the Getting Started with Vivado IP Integrator (https://reference.digilentinc.com/vivado/getting-started-with-ipi/2018.2) tutorial to obtain a basic Zynq block design.



An example of a Zynq block design.

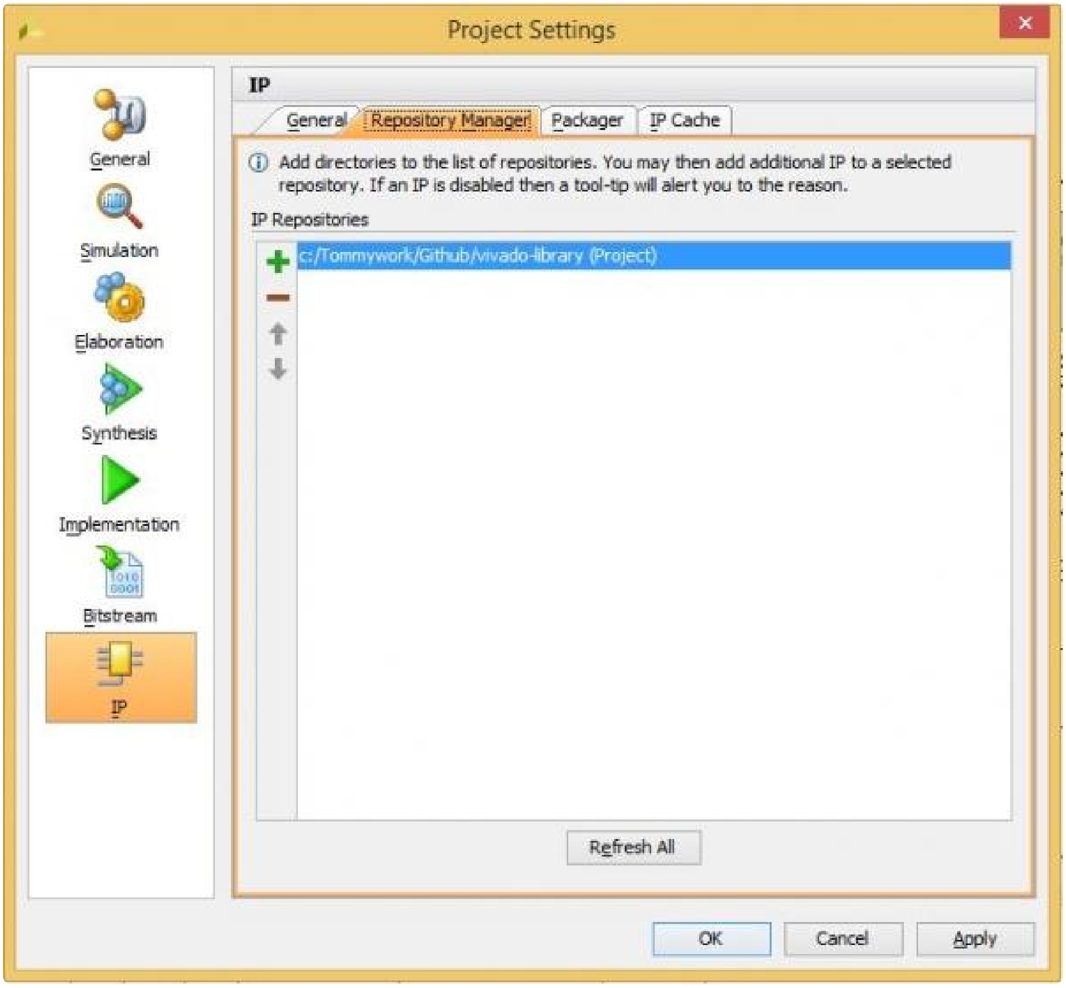
2. Add the Digilent Library Repository

2.1) Find the latest release of Digilent's vivado-library (https://github.com/Digilent/vivado-library/releases) repository where the version number matches the version of Vivado being used (example: “v2016.4-1” is the first release for Vivado 2016.4). Download the **vivado-library-<version>.zip** file (*NOT one of the source code archives!*), then extract this archive in a memorable location. This GitHub repository contains a large number of IP cores intended for use with Digilent boards, including all of Digilent's Pmod IP cores and Pmod interface description.
2.2) Click **Project Settings** under Project Manager.



(https://reference.digilentinc.com/_media/arty/projsetting.jpg)

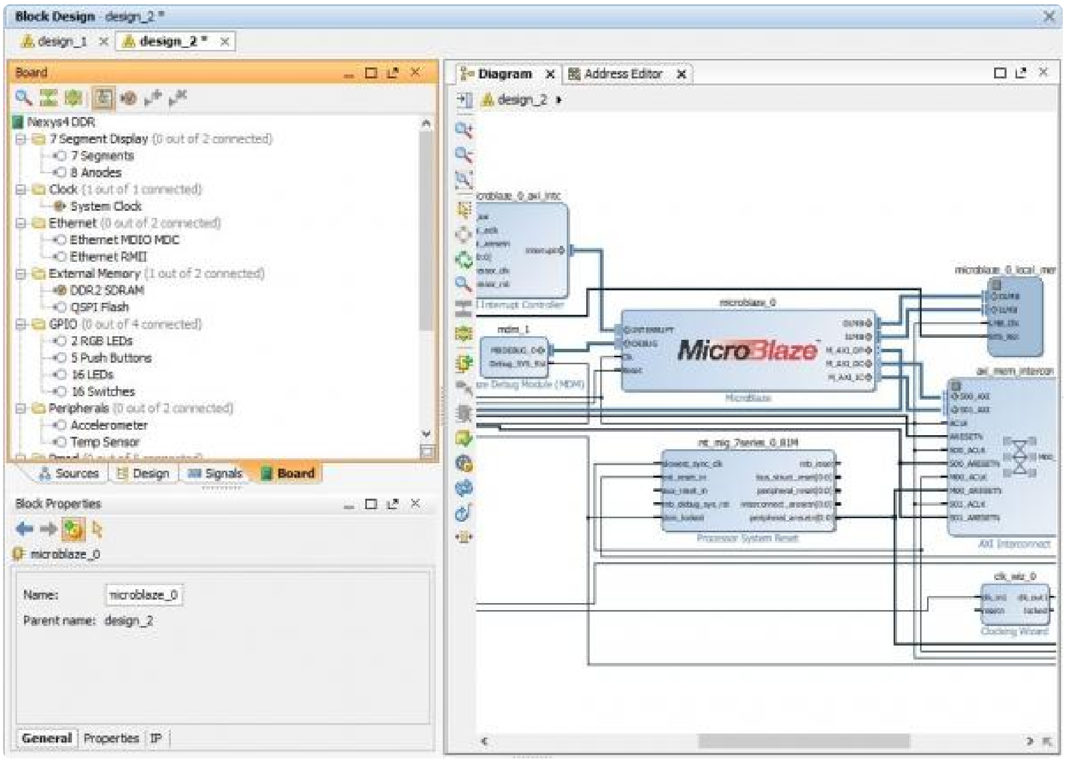
2.3) Click **IP** then open the **Repository Manager** tab. Click the (https://reference.digilentinc.com/_media/genesys2/plus.jpg) **Add** button and select the **vivado-library** folder from where the ZIP archive was extracted to. Click **OK**.



(https://reference.digilentinc.com/_media/vivado/iprepo.jpg)

3. Add the Pmod to Your Block Design

3.1) Click the **Board** tab (Highlighted in orange below)

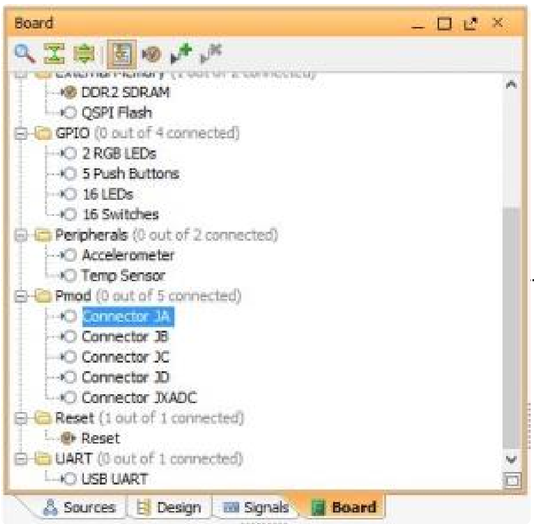


(https://reference.digilentinc.com/_media/vivado/boardtab.jpg)

Info

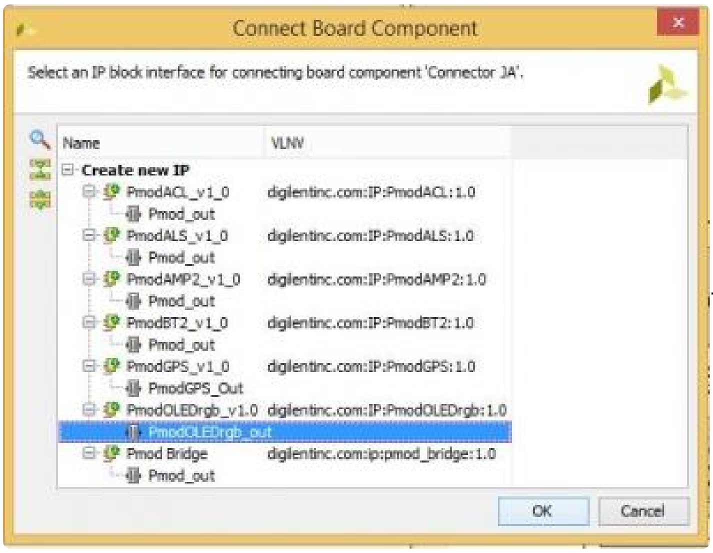
This list contains all of the components defined in the board file for your platform. You can use it to easily insert an IP block that will work with a piece of hardware found on your platform, for example an Ethernet port or general purpose LED (). Several of these should have already been selected when you created your initial design in step 1.1.

3.2) Scroll down to the Pmod section of the board components. Double click the connector that you want to set up.



(https://reference.digilentinc.com/_media/vivado/pmodconnecter.jpg)

3.3) Select the Pmod IP for your specific Pmod and click **OK**. The photo below shows the IP core for PmodOLEDrgb being selected.



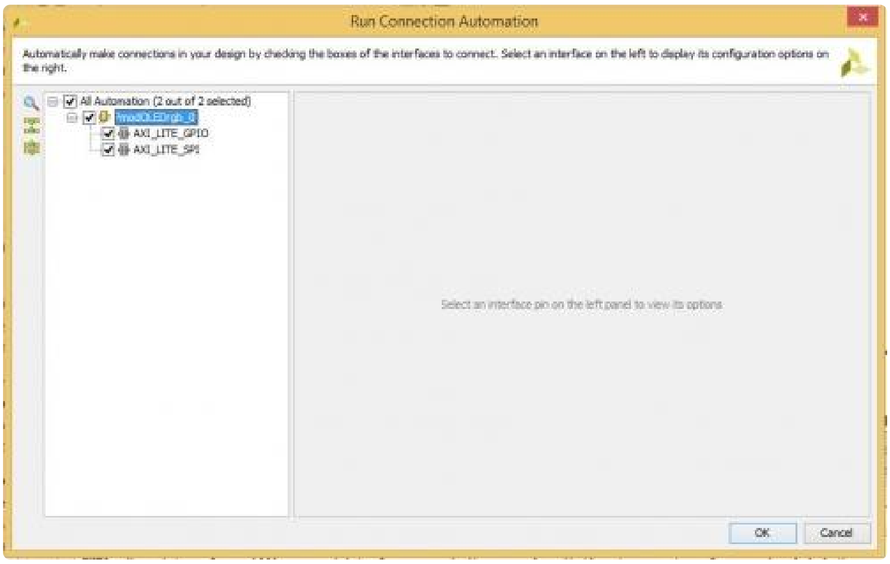
(https://reference.digilentinc.com/_media/vivado/pmodip.jpg)

Tip

Several of the simpler GPIO () Pmods can be used with the PmodGPIO IP Core. To see if your Pmod is supported with this IP core consult the Pmod compatibility table found in the Overview Section of this tutorial.

4. Run Connection Automation

4.1) Click **Run Connection Automation** then check the box next to the name of your Pmod IP core and click **OK**.



(https://reference.digilentinc.com/_media/vivado/auto.jpg)

5. Connect Reference Clocks

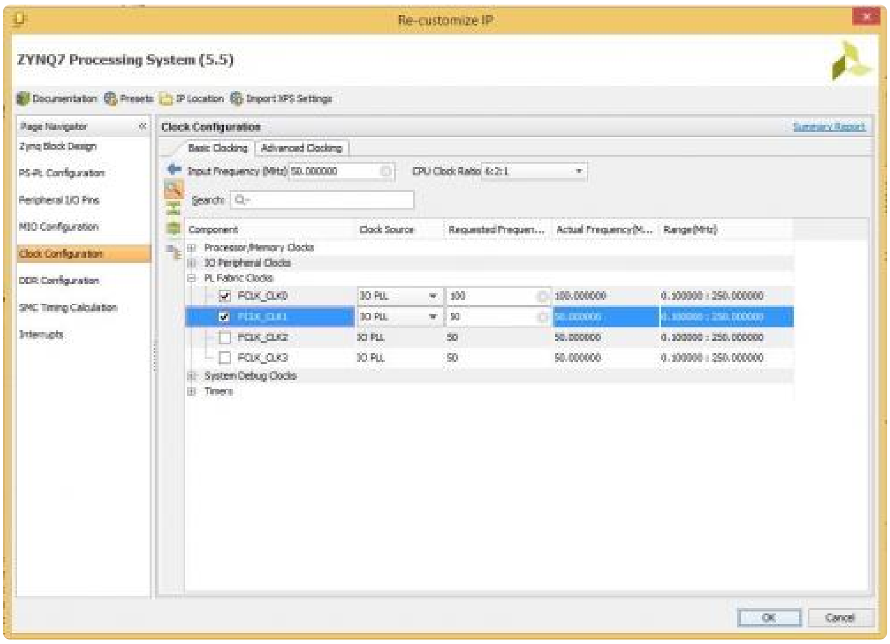
Important

Some Pmod IP cores require a reference clock to function properly. To see if your Pmod requires a reference clock consult the Pmod compatibility table found in the Overview Section. If your Pmod does not require a reference clock then skip to Step 6.

Attaching a reference clock to the Pmod IP core is different depending on which platform you are using. Select the tab that describes your platform best (consult the platform compatibility table in the Overview Section if you don't know).

Zynq

5.1) Double click the **ZYNQ Processing System** block to re-customize it. In the menu to the left, click **Clock Configuration**. Expand the **PL Fabric Clocks** drop down and check the first FCLK_CLK that is not already checked to activate it. Set the requested frequency to the frequency required for your Pmod. This frequency can be found in the Pmod compatibility chart in the Overview Section of this tutorial. Click **OK**.

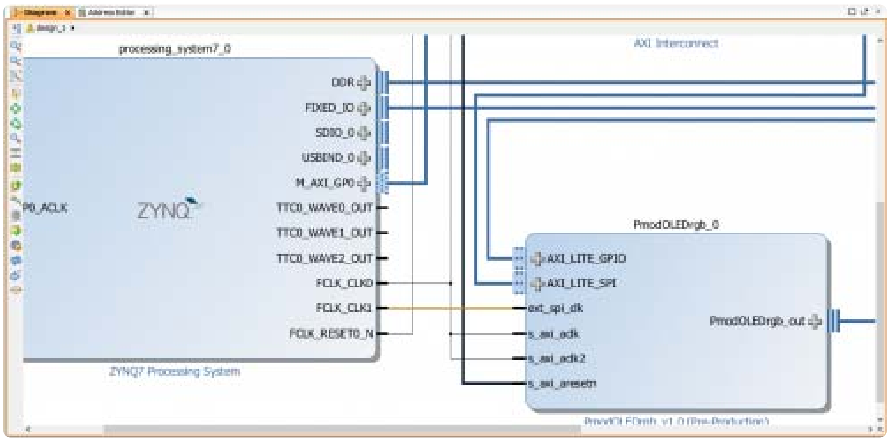


(https://reference.digilentinc.com/_media/vivado/zynqclock.jpg)

Tip

If one of the FCLK_CLK's that is already checked has a frequency that matches the frequency needed for your pmod, you may use that clock. It is possible to connect a single clock to multiple destinations.

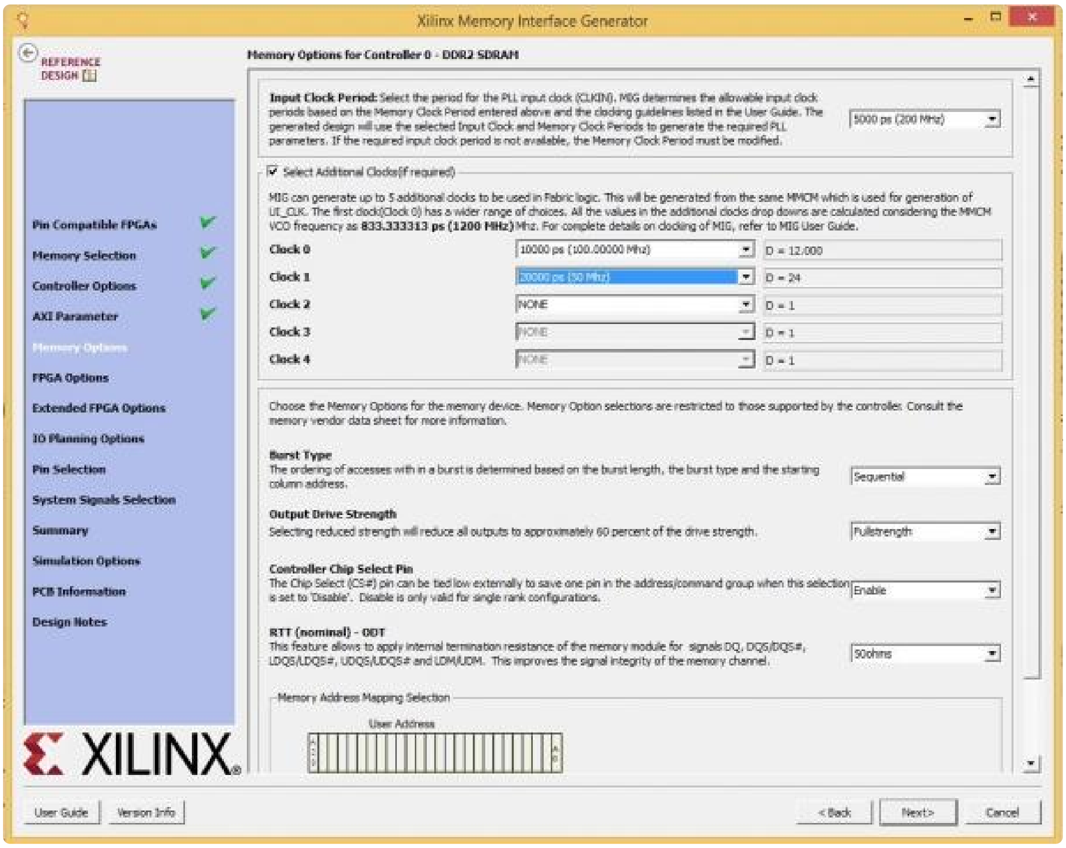
5.2) Connect this new clock to the clock input on your Pmod IP Core. The name of the clock input for your Pmod IP core can be found in the Pmod compatibility chart in the Overview Section of this tutorial.



(https://reference.digilentinc.com/_detail/playground/zynq-clock.png?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

MicroBlaze with MIG

5.1) Double click the mig_7series block to re-customize it. In the Xilinx Memory Interface Generator window, keep clicking **Next** until you see **Select Additional Clocks** (shown below). Click this box and select the frequency required for your Pmod or the closest available slower frequency. The required frequency can be found in the Pmod compatibility chart in the Overview Section of this tutorial.



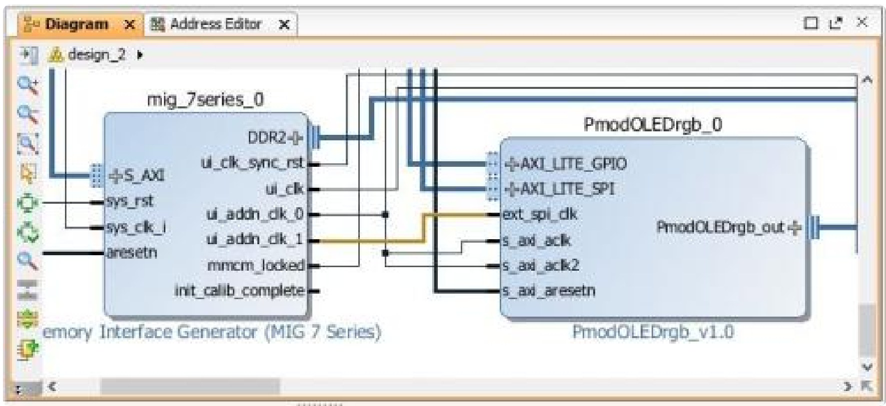
(https://reference.digilentinc.com/_media/vivado/migclock.jpg)

Tip

If one of the other clocks is already being used and has a frequency that matches the frequency needed for your Pmod then you may use that clock. It is possible to connect a single clock to multiple destinations. If this is the case, you may **Cancel** out of the MIG configuration dialog.

5.2) Keep clicking **Next**. When you reach the pin selection screen, click **Validate** and then **OK**. Keep clicking **Next**. Click **Accept** on the license agreement screen, then continue to click **Next**. Once you've reached the end, click **Generate** to regenerate your MIG block with your additional clocks.

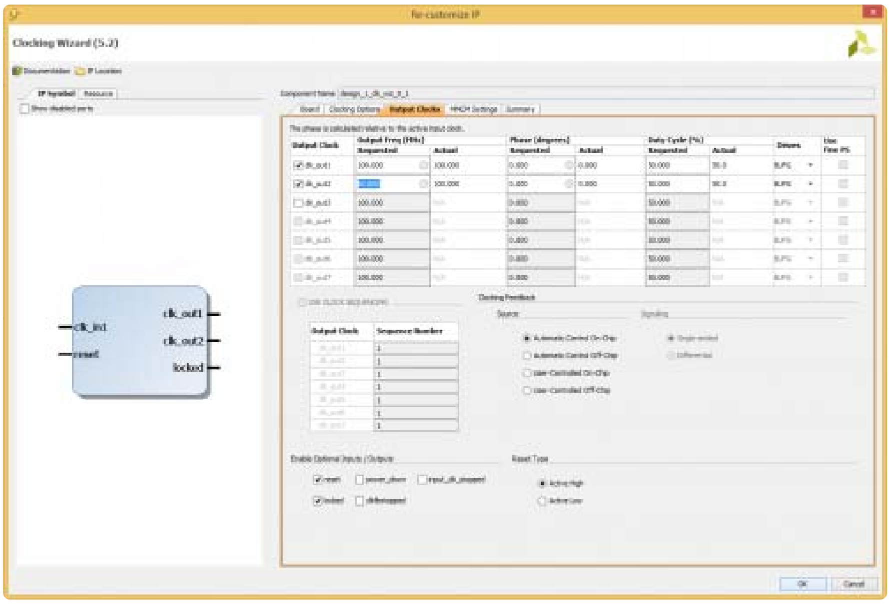
5.3) Connect this new clock to the clock input on your Pmod IP Core. The name of the clock input port for your Pmod IP core can be found in the Pmod compatibility chart in the Overview Section of this tutorial.



(https://reference.digilentinc.com/_media/vivado/clkconect.jpg)

MicroBlaze without MIG

5.1) Double click the clocking wizard IP block to re-customize it. In the customization dialog, select the Output Clocks tab. Check the next clock that is not already checked to activate it. Set the requested output frequency to the frequency required for your Pmod. This frequency can be found in the Pmod compatibility chart in the Overview Section of this tutorial. Click **OK**.

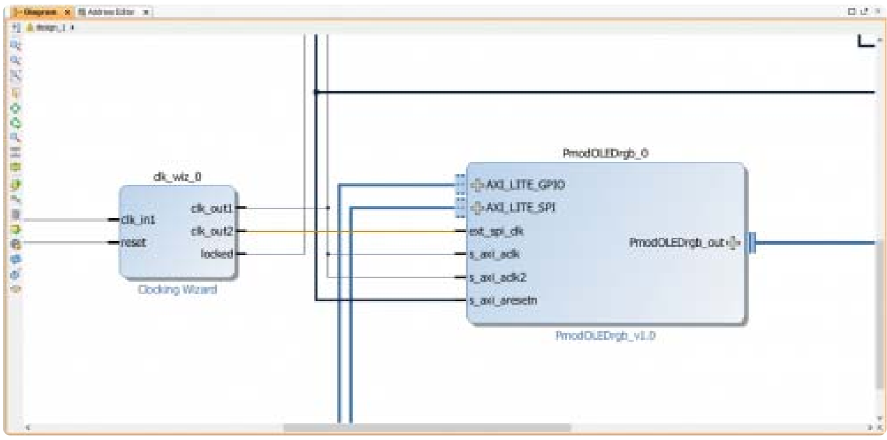


(https://reference.digilentinc.com/_detail/playground/clk-wiz-nomig.png?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

Tip

If one of the other clocks is already being used and has a frequency that matches the frequency needed for your Pmod then you may use that clock. It is possible to connect a single clock to multiple destinations.

5.2) Connect this new clock to the clock input on your Pmod IP Core. The name of the clock input for your Pmod IP core can be found in the Pmod compatibility chart in the Overview Section of this tutorial.



(https://reference.digilentinc.com/_detail/playground/clk-nomig.png?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

6. Connect Interrupts

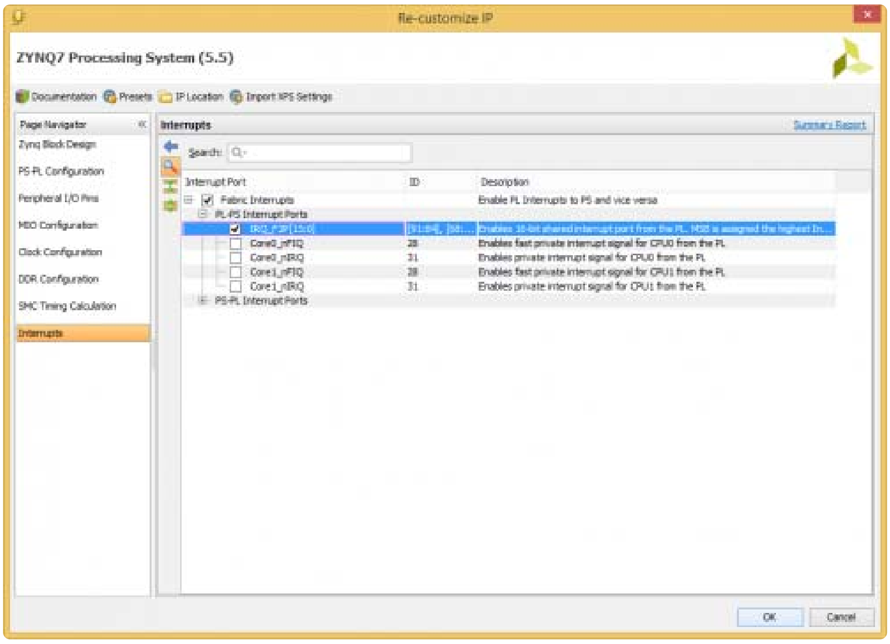
Important

Some Pmod IP cores require an interrupt to function properly. To see if your Pmod requires an interrupt consult the Pmod compatibility table found in the Overview Section. If your Pmod does not require an interrupt then skip to Step 7.

Attaching a Pmod IP core interrupt to the processor is different depending on which platform you are using. Select the tab that describes your platform best (consult the platform compatibility table in the Overview Section if you don't know).

Zynq

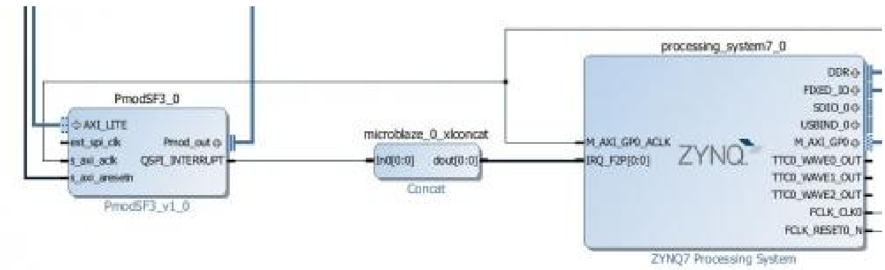
6.1) If a port named *IRQ_F2P* does not exist on your Zynq Processing System block, double click on the block to re-customize it. In the menu to the left, click **Interrupts**. Check and then expand Fabric Interrupts, then expand *PL-PS Interrupts* and check the box next to IRQ_F2P.



(https://reference.digilentinc.com/_detail/playground/zynq-intr.png?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

6.2) Add a **Concat** IP core to the block design. Re-customize the concat block to make sure that the number of inputs matches the number of interrupts you need to connect to your Zynq Processor - probably only one. Click **Ok**.

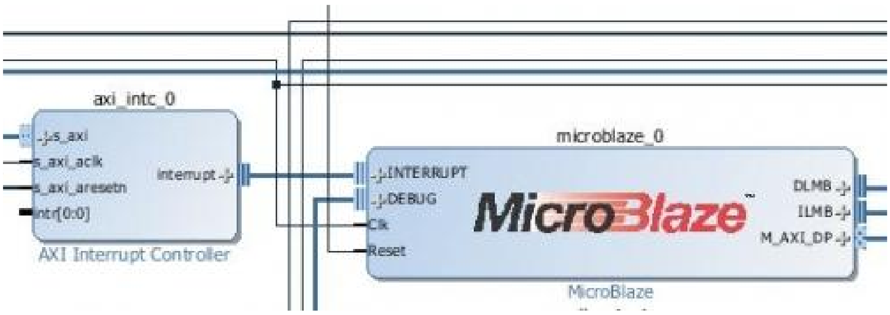
6.3) Connect your Pmod IP core's interrupt port to the input port to the concat block, and the concat's output port to the IRQ_F2P port of your Zynq Processing System.



(https://reference.digilentinc.com/_detail/playground/zynq-concat.jpg?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

Microblaze

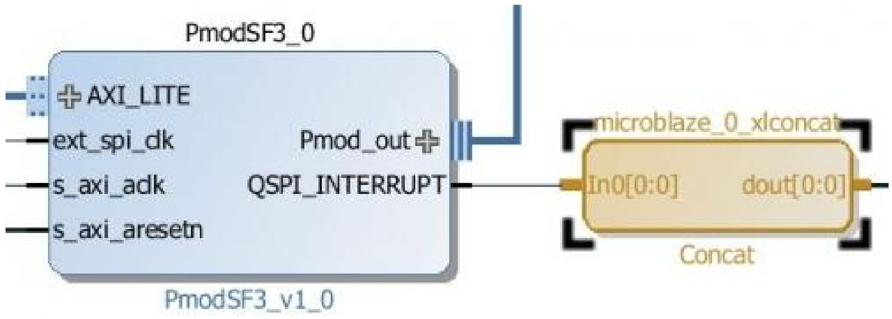
6.1) If your MicroBlaze block doesn't have an AXI Interrupt Controller connected to it's INTERRUPT port, add an AXI Interrupt controller to your block design. Manually connect the interrupt port of the axi_intc_0 to the INTERRUPT port of your MicroBlaze block. Use **Connection Automation** to connect it's AXI interface to the rest of the system.



(https://reference.digilentinc.com/_detail/playground/ub-intr.jpg?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

6.2) Add a Concat IP core to the block design. Re-customize the concat block to make sure that the number of inputs matches the number of interrupts you need to connect to your Microblaze Processor - probably only one. Click **Ok**.

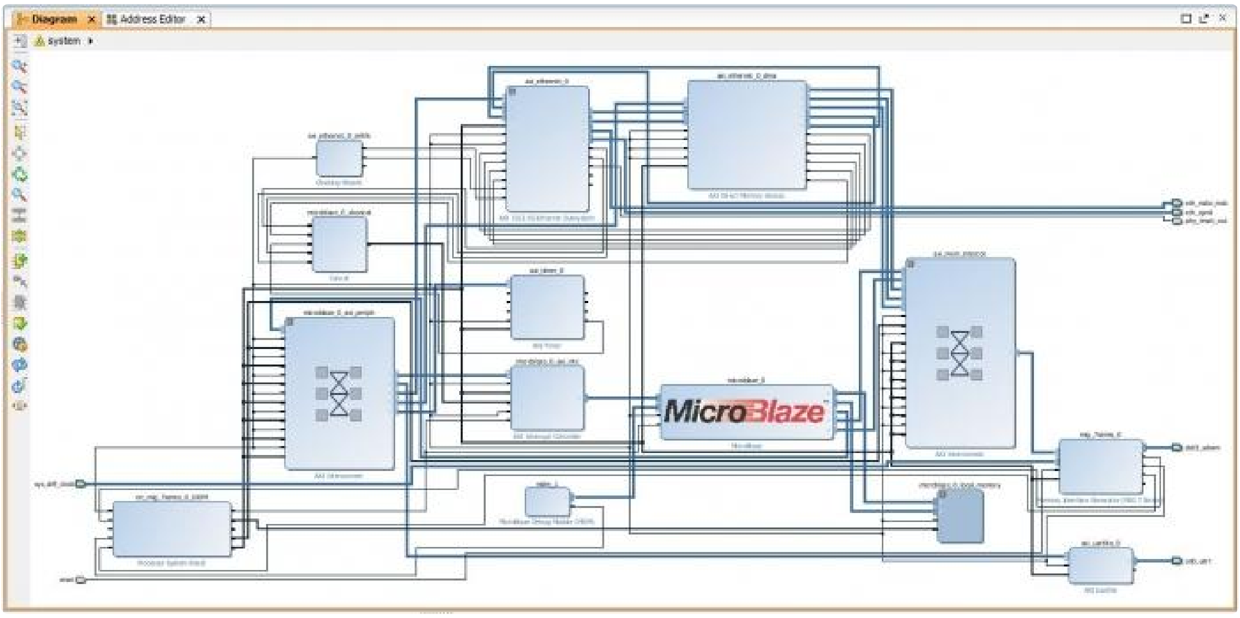
6.3) Connect your Pmod IP core's interrupt port to the input port to the concat block, and the concat's output port to the *intr* port of your AXI Interrupt Controller.



(https://reference.digilentinc.com/_detail/playground/ub-concat.jpg?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

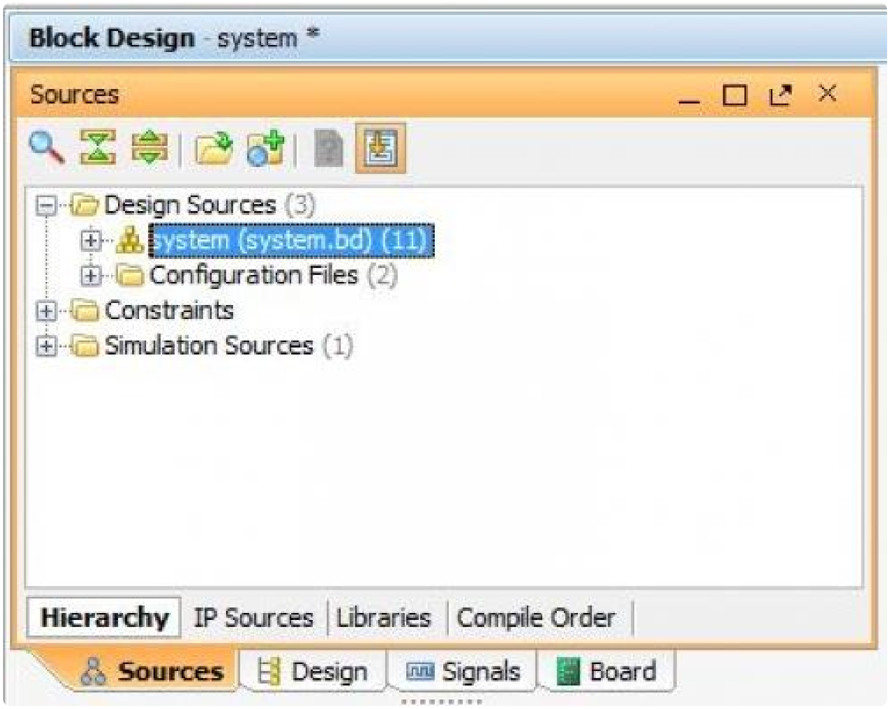
7. Validate the Design

7.1) Click the **Regenerate Layout** button to rearrange your block design.

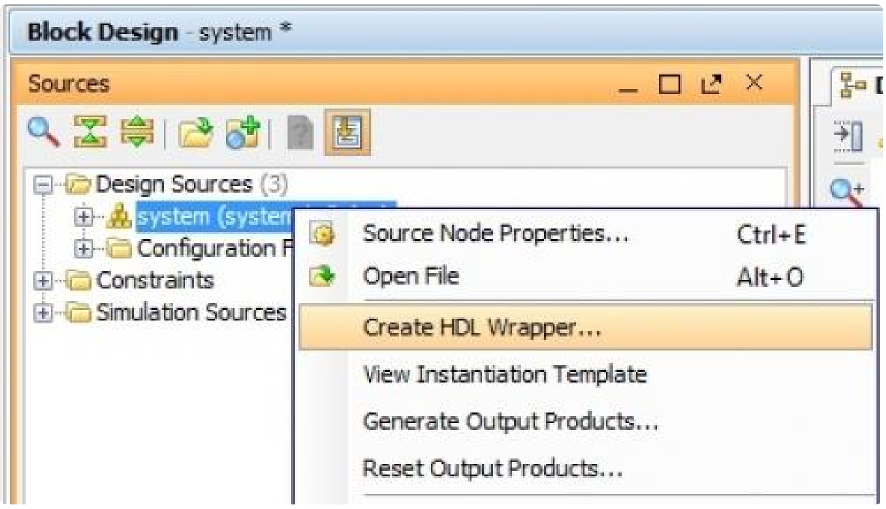


7.2) Select **Validate Design**. This will check for design and connection errors.

7.3) If you have already created an HDL wrapper for your block design as part of the board specific **Getting Started With…** tutorial, skip the remainder of this step. Otherwise, after the design validation step we will proceed with creating a HDL System Wrapper. Click on the **Sources** tab and find your block design.



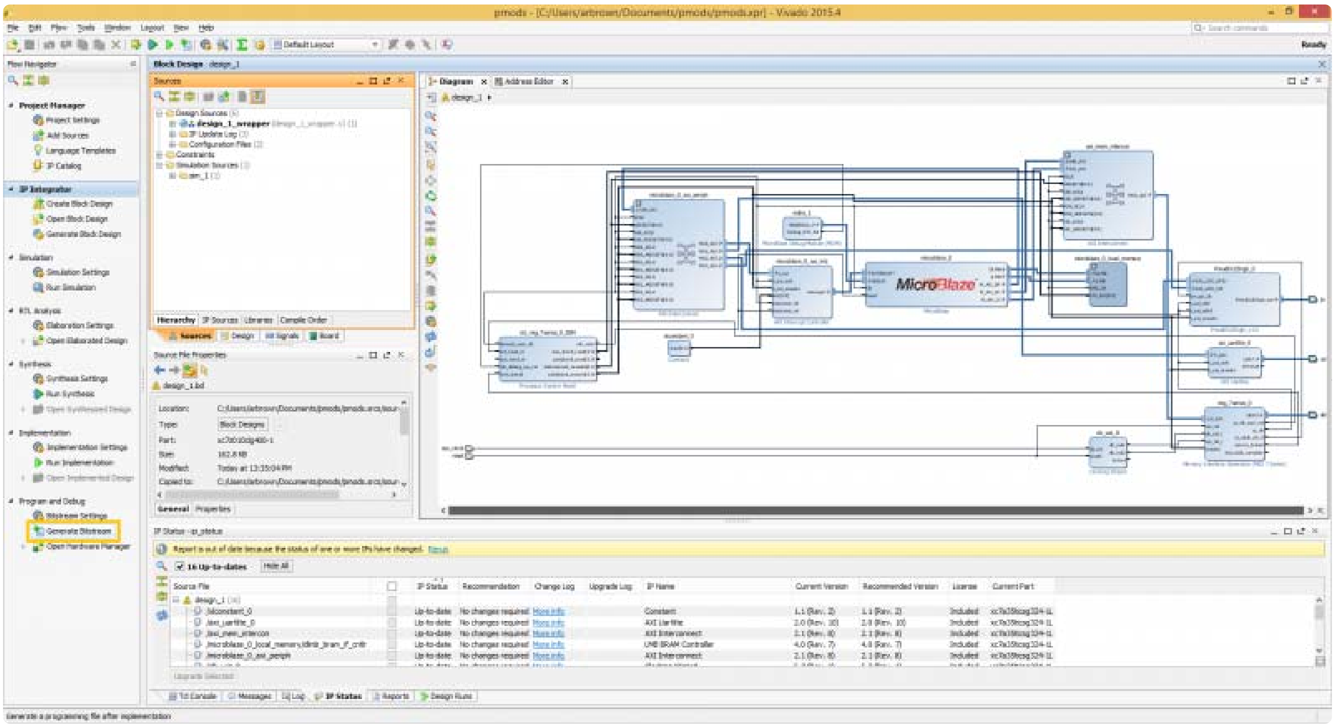
7.4) Right click on your block design and click **Create HDL Wrapper**. Let Vivado manage the wrapper and automatically update it and click **OK**.



This will create a top module in VHDL and will allow you to generate a bitstream.

8. Generate the Bit File

8.1) In the top toolbar in Vivado, click **Generate Bitstream**. This can also be found in the *Flow Navigator* panel on the left, under *Program and Debug*. If you haven't already saved your design, you will get a prompt to save the block design.



(https://reference.digilentinc.com/_detail/playground/gen-bit.png?id=learn%3Aprogrammable-logic%3Atutorials%3Apmod-ips%3A2018.2)

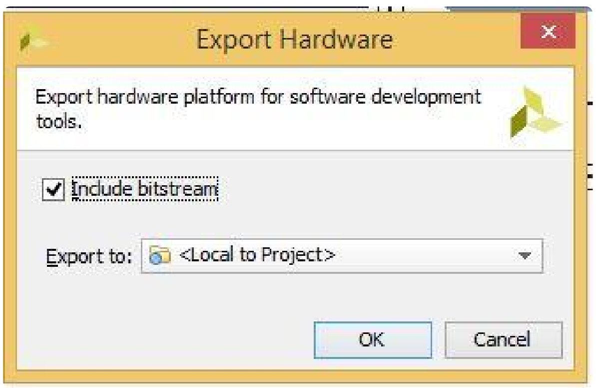
8.2) The bit file generation will begin. The tool will run *Synthesis* and *Implementation*. After both have been successfully completed, the bit file will be created. You will find a status bar of Synthesis and Implementation running on the top right corner of the project window.

This process can take anywhere from **5 to 60 minutes** depending on the computer Vivado is running on and the size of the target FPGA.

8.3) After the bitstream has been generated, a message prompt will pop-up on the screen. You don't have to open the Implemented Design for this demo. Just click **Cancel**.

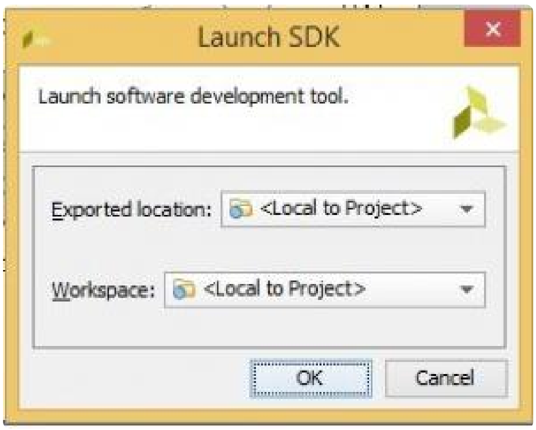
9. Export the Hardware Design to SDK

9.1) At the top of the Vivado window, click **File → Export → Export Hardware**. Check the box to **Include Bitstream** and click **OK**. This will give Xilinx SDK all of the information it needs to know about the hardware design, as well as the files needed to program the hardware onto a target FPGA system board.



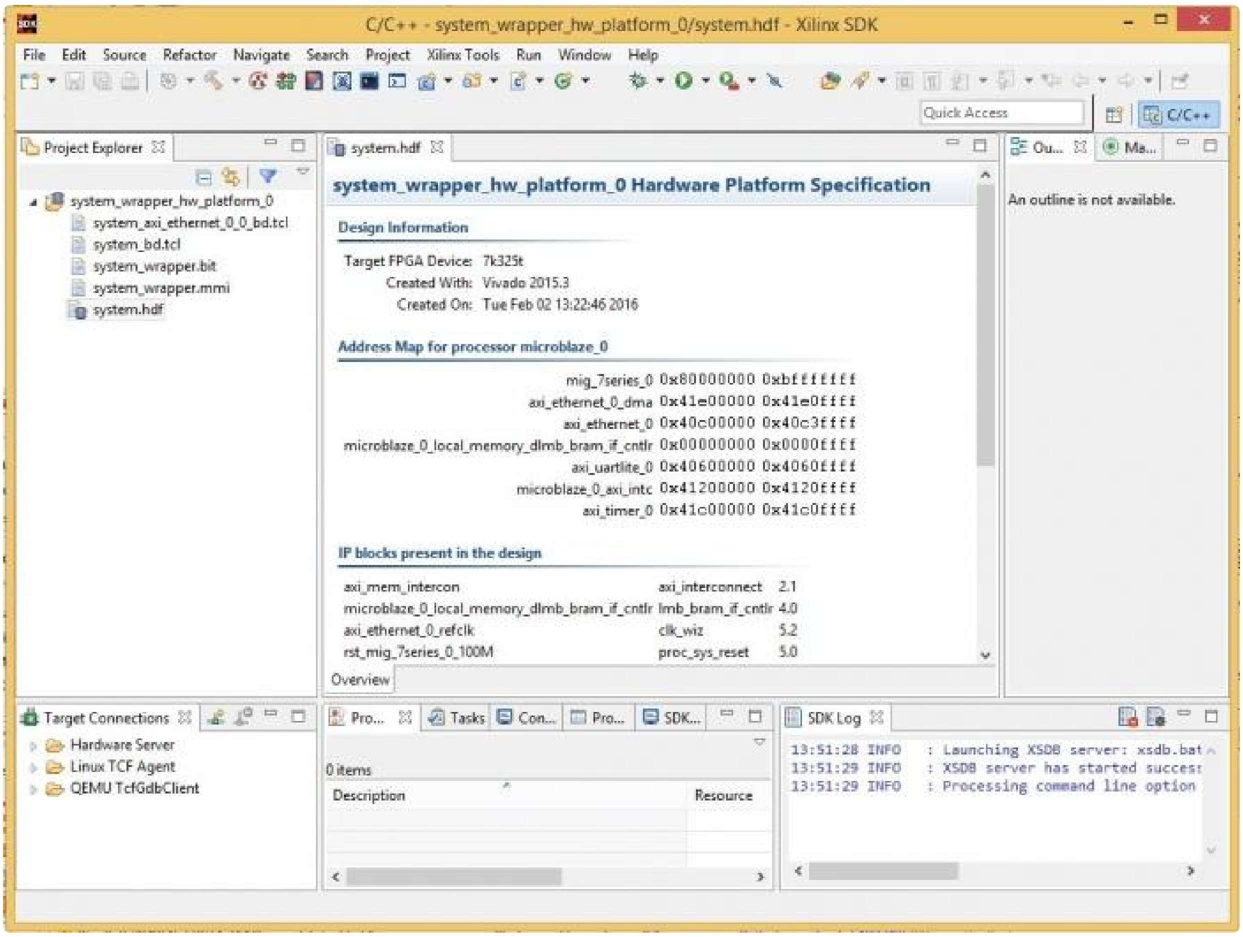
A new file directory will be created in the project directory under **echo_server.SDK** similar to the Vivado hardware design project name. Two other files, *.sysdef* and *.hdf* are also created. This step essentially creates a new SDK Workspace.

9.2) On the main toolbar, click **File → Launch SDK**. Leave both of the dropdown menus as their default *<Local to Project>* and click **OK**. This will open Xilinx SDK and import the exported hardware.



10. Xilinx SDK

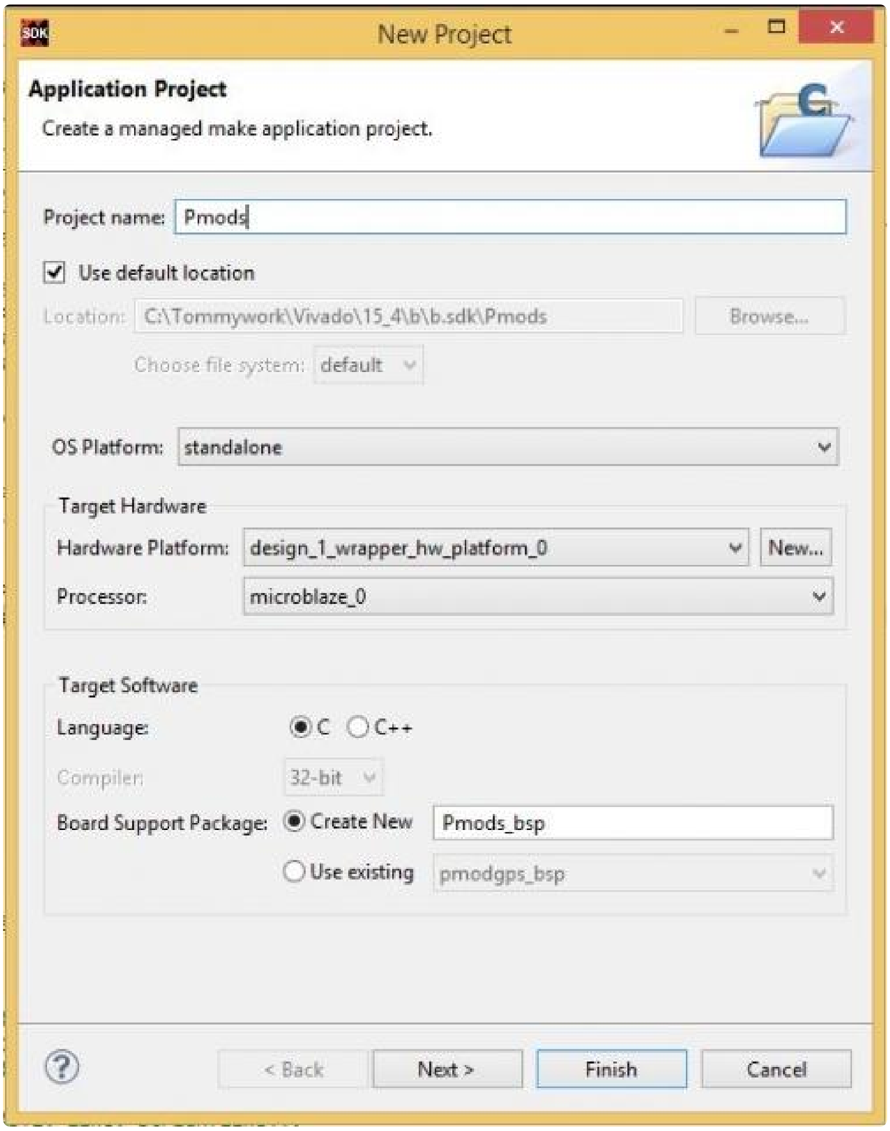
The HW design specification and included IP blocks are displayed in the "system.hdf" file. Xilinx SDK is independent of Vivado, i.e. from this point, you can create your SW project in C/C++ on top of the exported HW design. If necessary, SDK can be launched directly with the <PROJECT>.sdk folder in the main Vivado project directory as the workspace. From this point, if you need to go back to Vivado and make changes to the HW design, then it is recommended to close the SDK window and make the required HW design edits in Vivado. After this you must follow the sequence of saving design, allowing Vivado to regenerate the HDL wrapper, and generating a new bit file. This new bit file and modified hardware design must then be exported to SDK.



Within the *Project Explorer* tab on the left, you can see the hardware platform project. The name of the hardware platform follows the name of the block design wrapper created in Vivado. This hardware platform has all the HW design definitions, IP interfaces that have been added, external output signal information and local memory address information.
The drivers for any Pmod IPs in the design can be found in the appropriate folder in the hardware platform, under **/drivers**. If you want to edit these drivers, use the versions found in the board support package project under **libsrc/**. If you do modify the drivers, keep in mind that any changes to your hardware will overwrite these changes, as well as any use of the **Regenerate BSP Sources** option.

11. Create a New Application Project in SDK

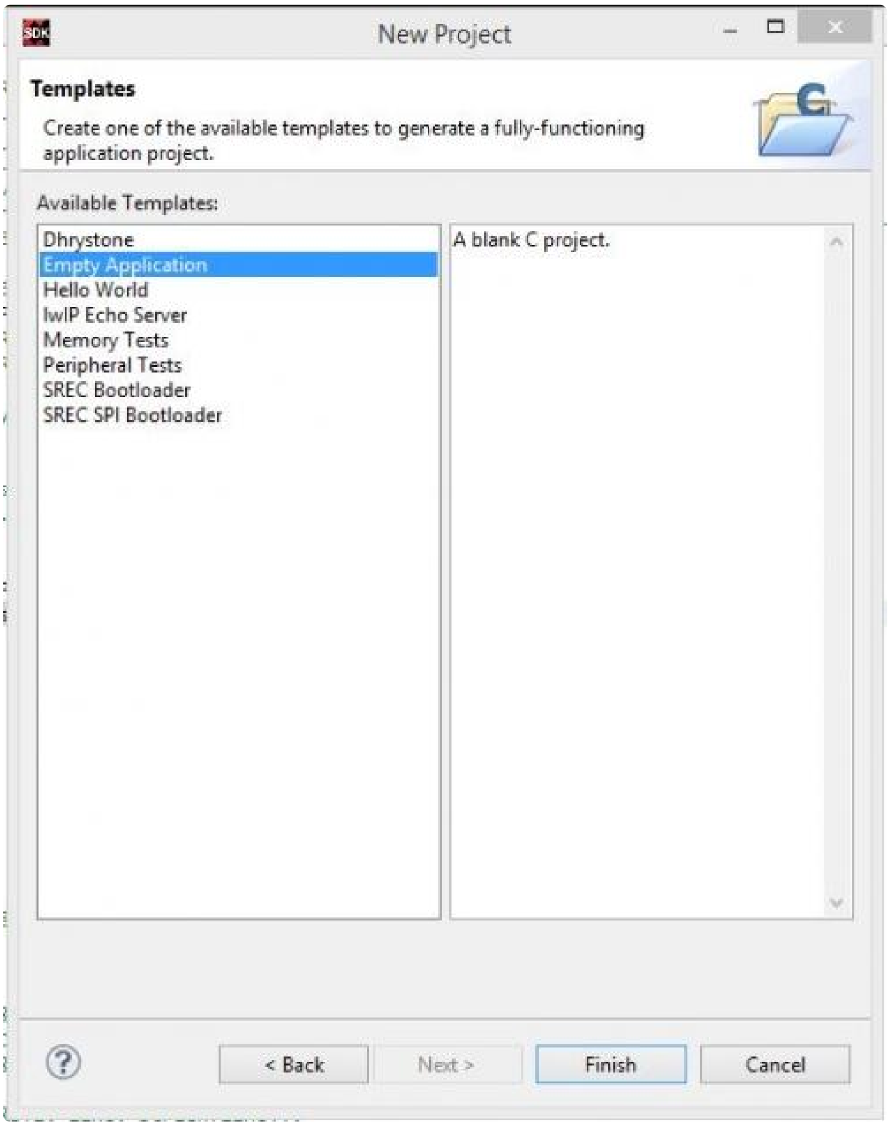
11.1) Click the **New** dropdown arrow and select **Application Project**.



(https://reference.digilentinc.com/_media/vivado/newproj.jpg)

Give your project a name that has no empty spaces and click **Next**.

11.2) Select **Empty Application** from the list of templates and click **OK**.



(https://reference.digilentinc.com/_media/vivado/empty.jpg)

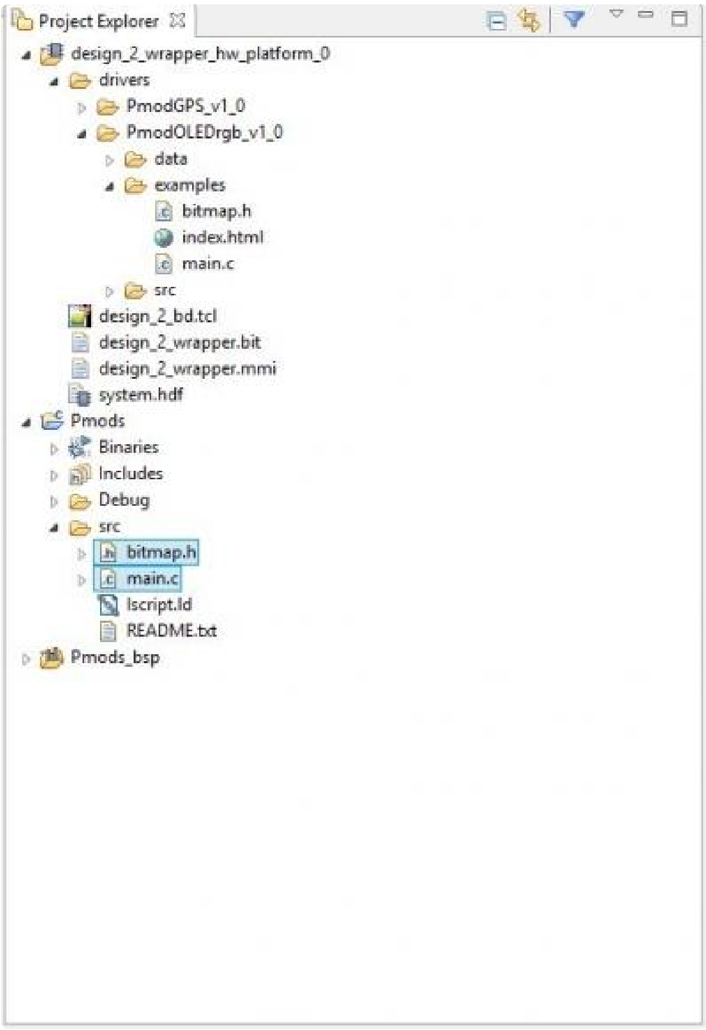
You will see two new folders in the **Project Explorer** panel.

- Your application project which contains all the binaries, .C and .H (Header) files
- The board support package for your project, which contains driver source files that your project may include

Our main working source folder also contains an important file shown here which is the "lscript.ld". This is a Xilinx auto generated linker script file and includes information about memory addresses for different IP components of your block design, as well as the sizes of other memory regions.

12. Import the Example Project

Expand the **design_1_wrapper_platform_0** folder. Within the **drivers** folder, you will find the list of Pmods that you are using in your design. Expand the **Pmod…/examples** folder and copy all of the files found here into the **project_name/src** folder.

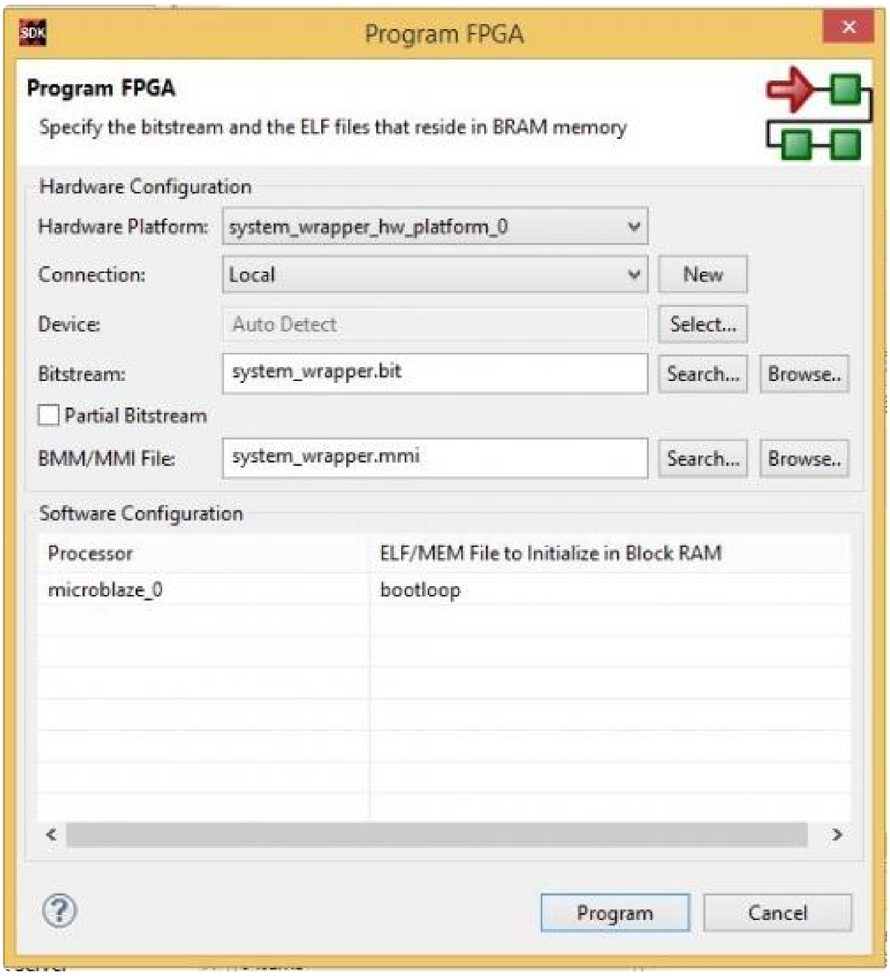


(https://reference.digilentinc.com/_media/vivado/example.jpg)

13. Program the FPGA with the Bit File

13.1) Make sure that your board is turned on and connected to the host PC with micro USB cables for UART and programming. On some boards, you will only need to connect a single PROG/UART port while on others you will need to connect your PC to two different ports typically named UART and PROG or JTAG. On the top toolbar, click the **Program FPGA** button. Some boards will also require that they be connected to a separate power source.

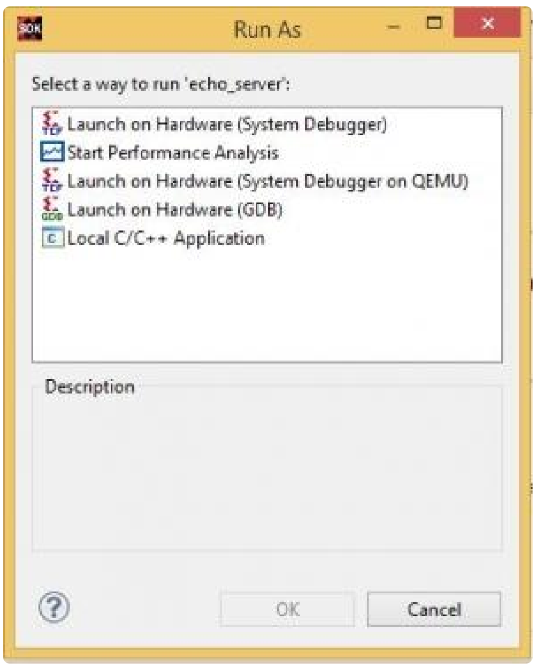
13.2) Click **Program** to program your FPGA with your hardware design.



14. Program the Microblaze/ZYNQ Processor

14.1) The majority of demos require that you use a serial terminal on your PC to read messages printed by the demo. Settings for the terminal will vary depending on your board, but typically you will need to use a baud rate of 115200 or 9600, 8 bit data, no parity bit, and one stop bit. Zynq projects will use 115200 baud, while the baud rate for MicroBlaze projects will depend on the configuration of the Uartlite IP in Vivado.

14.2) Select your application project and click the **Run As…** button. Select **Launch on Hardware (System Debugger)** and click **OK**.



> 14.3) Xilinx SDK will run the program starting in main.c on your Microblaze/Zynq processor. Review the comment header at the top of the example main file to get more information on what the demo does, as well as any additional setup requirements.

---

learn (https://reference.digilentinc.com/tag/learn?do=showtag&tag=learn), programmable-logic (https://reference.digilentinc.com/tag/programmable-logic?do=showtag&tag=programmable-logic), tutorial (https://reference.digilentinc.com/tag/tutorial?do=showtag&tag=tutorial), software (https://reference.digilentinc.com/tag/software?do=showtag&tag=software), vivado (https://reference.digilentinc.com/tag/vivado?do=showtag&tag=vivado), in-work (https://reference.digilentinc.com/tag/in-work?do=showtag&tag=in-work), arty (https://reference.digilentinc.com/tag/arty?do=showtag&tag=arty), basys-3 (https://reference.digilentinc.com/tag/basys-3?do=showtag&tag=basys-3), cora-z7 (https://reference.digilentinc.com/tag/cora-z7?do=showtag&tag=cora-z7), cmod-s7 (https://reference.digilentinc.com/tag/cmod-s7?do=showtag&tag=cmod-s7), zybo (https://reference.digilentinc.com/tag/zybo?do=showtag&tag=zybo), genesys-2 (https://reference.digilentinc.com/tag/genesys-2?do=showtag&tag=genesys-2), nexys-4-ddr (https://reference.digilentinc.com/tag/nexys-4-ddr?do=showtag&tag=nexys-4-ddr), nexys-video (https://reference.digilentinc.com/tag/nexys-video?do=showtag&tag=nexys-video), zybo (https://reference.digilentinc.com/tag/zybo?do=showtag&tag=zybo), arty-a7 (https://reference.digilentinc.com/tag/arty-a7?do=showtag&tag=arty-a7), arty-s7 (https://reference.digilentinc.com/tag/arty-s7?do=showtag&tag=arty-s7), arty-z7 (https://reference.digilentinc.com/tag/arty-z7?do=showtag&tag=arty-z7), zybo-z7 (https://reference.digilentinc.com/tag/zybo-z7?do=showtag&tag=zybo-z7), sword (https://reference.digilentinc.com/tag/sword?do=showtag&tag=sword)

Our Partners

- Xilinx University Program (https://store.digilentinc.com/partners/xilinx-university-program/)
- Technology Partners (https://store.digilentinc.com/technology-partners/)
- Distributors (https://store.digilentinc.com/our-distributors/)

Help

- Technical Support Forum (https://forum.digilentinc.com)
- Reference Wiki (https://reference.digilentinc.com)
- Contact Us (https://store.digilentinc.com/contact-us/)

Customer Info

- Videos (https://youtube.com/user/digilentinc)
- FAQ (https://resource.digilentinc.com/verify/faq)
- Store Info (https://store.digilentinc.com/store-info/)

Company Info

- About Us (https://store.digilentinc.com/ pageid=26)
- Shipping & Returns (https://store.digilentinc.com/ returns/)
- Legal (https://store.digilentinc.com/
- Jobs (https://store.digilentinc.com/
- Internships (https://store.digilentinc.com/

Connect With Us

- (https://twitter.com/digilentinc)
- (https://www.facebook.com/Digilent)
- (https://www.youtube.com/user/DigilentInc)
- (https://instagram.com/digilentinc)

- (https://github.com/digilent)
- (https://www.reddit.com/r/digilent)
- (https://www.linkedin.com/company/1454013)
- (https://www.flickr.com/photos/127815101@N07)