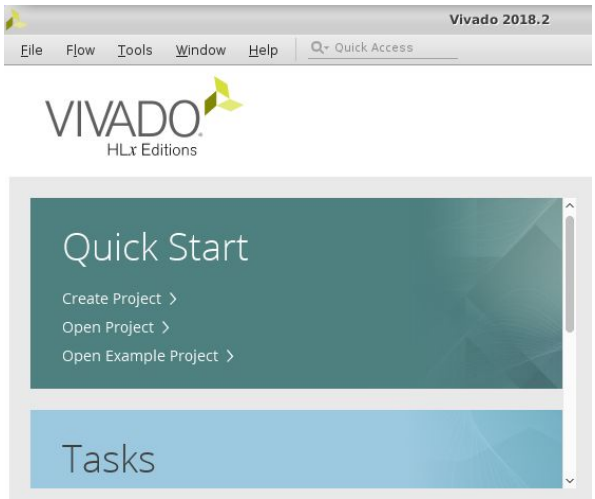
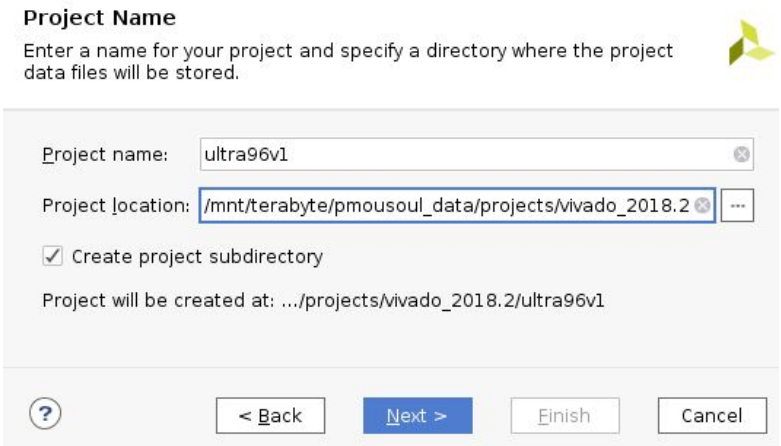
Platform Creation for Ultra96 v1.2 board using Xilinx Vivado, PetaLinux, and SDSoC

1. Create new Vivado project

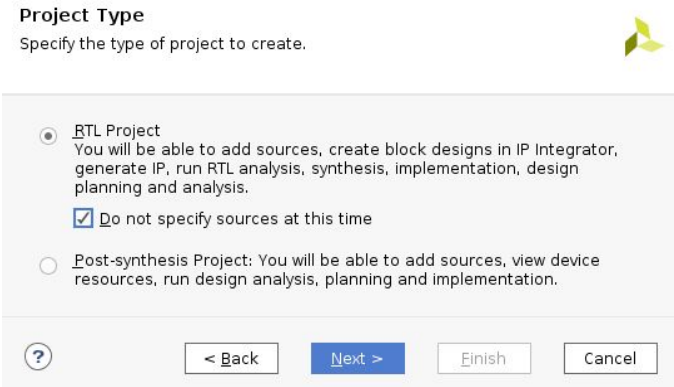
Open Vivado IDE and select **Create Project** in Quick Start.



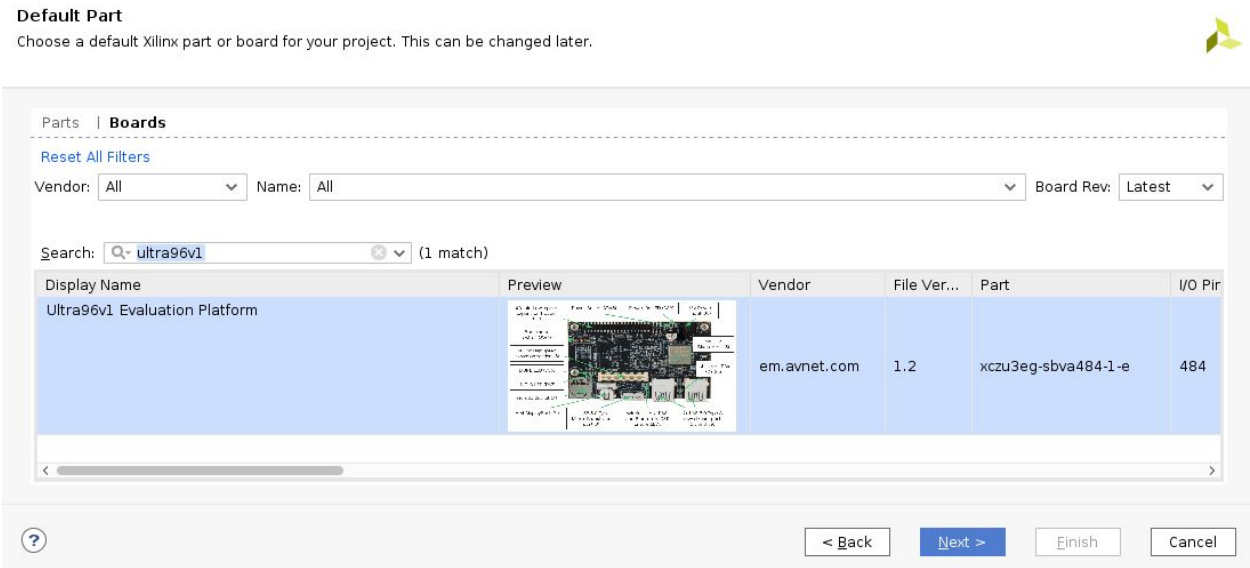
Click **Next.** In the New Project window, enter the name of the project (e.g. use **ultra96v1)** and select the location to be saved on the hard disk.



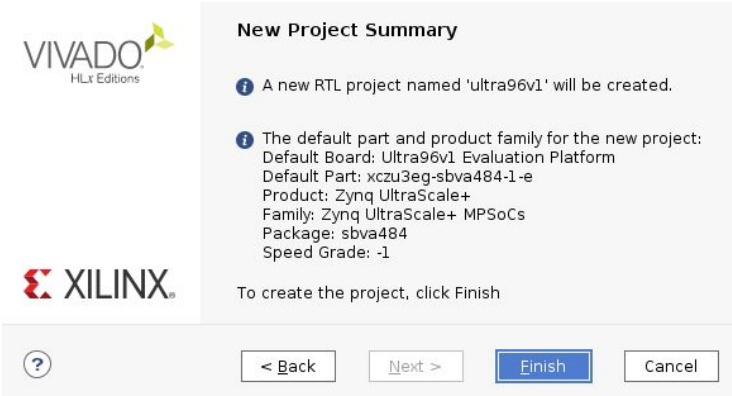
In the Project Type window, select **RTL Project**, select **Do not specify sources at this time**, and click on **Nex**t to create a new project using IP blocks (IP Integrator) without selecting any predefined source files.



In the Default Part window, select **Boards**, and type **ultra96v1** in the search field. Select the **Ultra96v1 Evaluation Platform**.

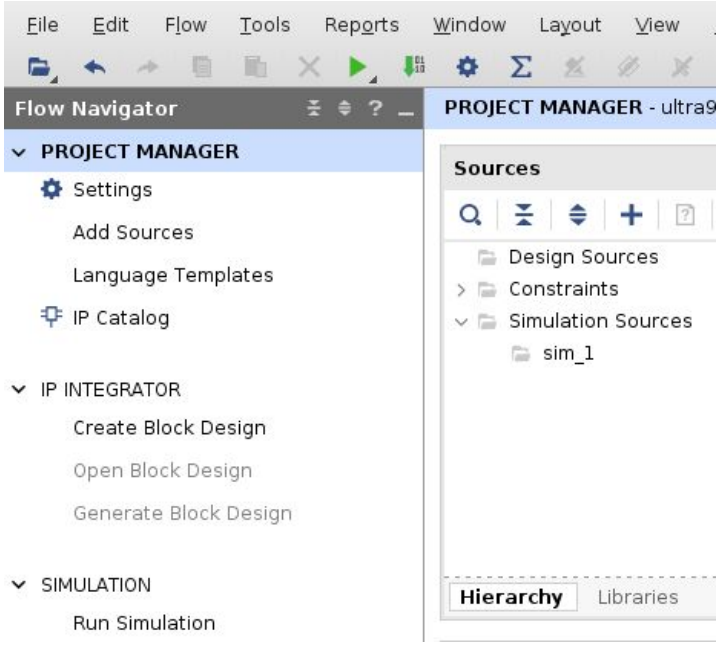


The New Project Summary window should contain the same information as below.

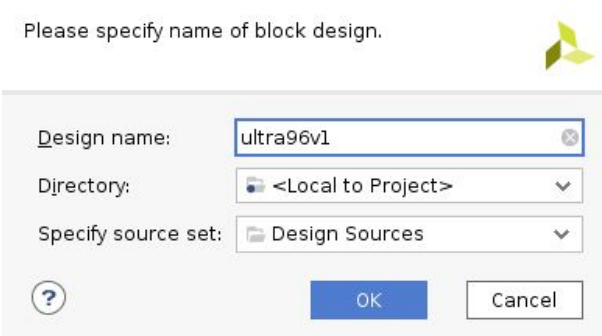


2. Create design using IP blocks

Goto **Flow Navigator**, expand **IP INTEGRATOR** and select **Create Block Design**.



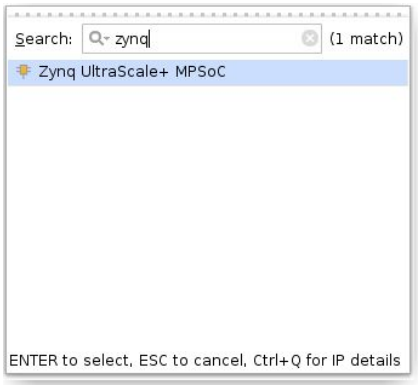
In the window that opens, we type the name of our design.
CATION: The design name we enter here must be the same as the SDSoC platform name that we will create later.
Enter **ultra96v1** and click **OK**.



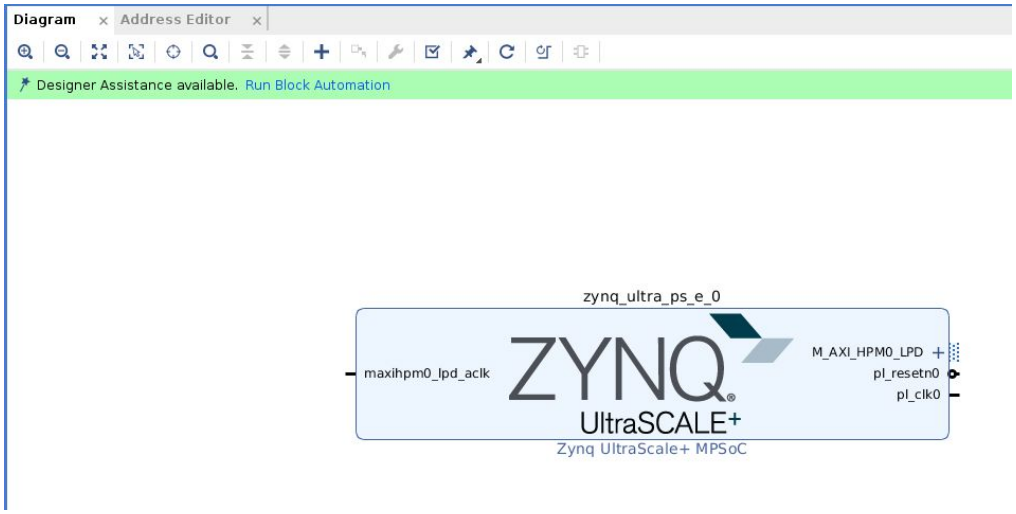
The following table gives a brief description of the IP blocks that will be included in our design.

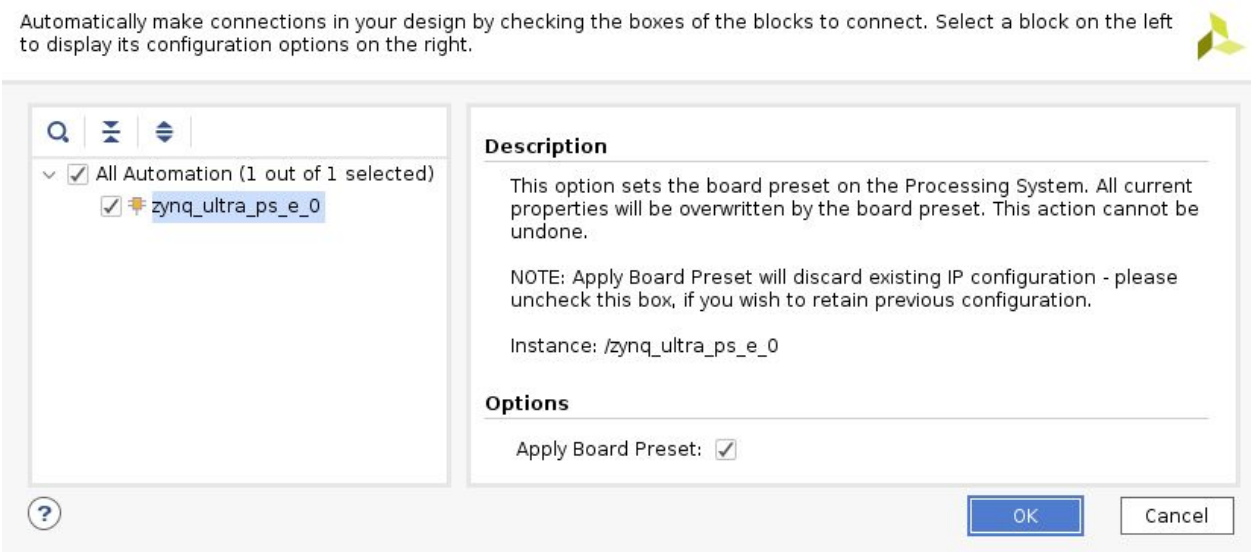
IP Block	Usage Summary
Zynq UltraScale+ MPSoC Processor System (PS)	• Dual-core Arm processor with cache hierarchy • Integrated I/O peripherals • DDR memory controller with external memory interface • PS to Programmable Logic (PL) interconnects • PL to PS interconnects
Processor System Reset Block (PL)	• Reset sequencing and synchronization block for PL logic
Clocking Wizard (PL)	• Multiple output clock generator to drive PL logic
Concat Block (PL)	• PL interrupt structure that feeds Zynq UltraScale+ MPSoC PS interrupt request input

In the **Diagram** section of the BLOCK DESIGN sub-window, we click on the **+** symbol to add IP blocks to our design. Or we can right-click in the **Diagram** section and select **Add IP**. We add a **Zynq UltraScale+ MPSoC** IP in our design.

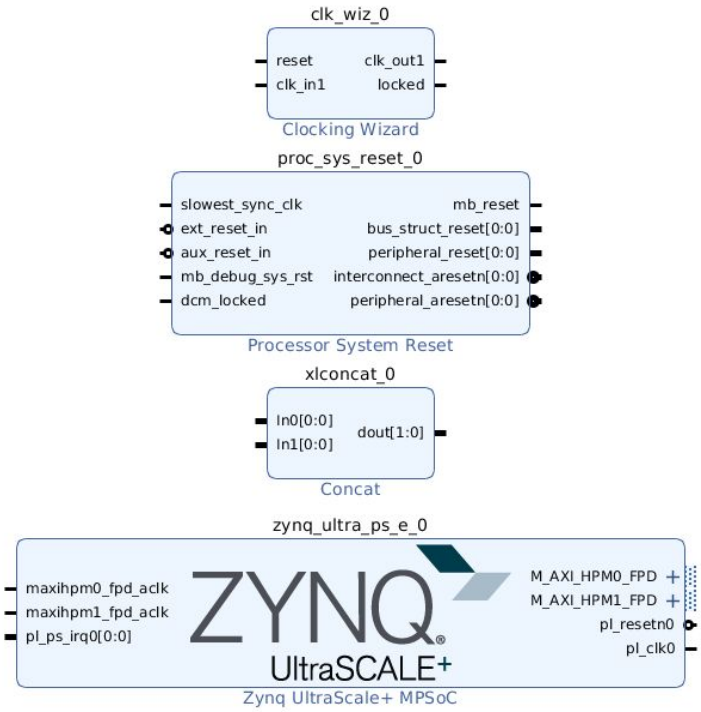


Click on the **Run Block Automation** of the **Designer Assistance** to **Apply Board Preset**. Click **OK**.

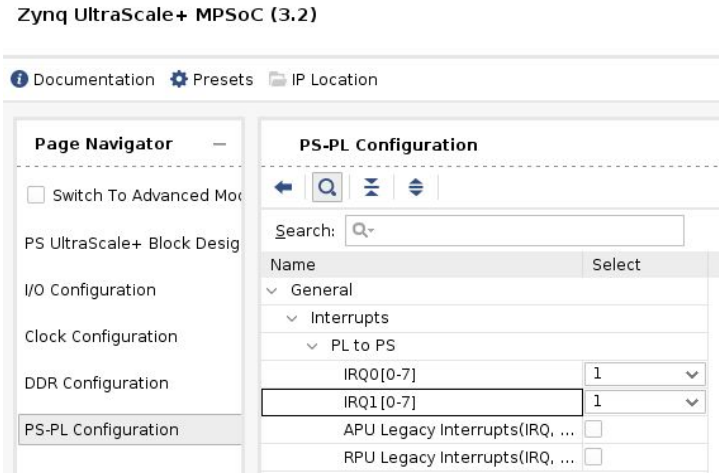




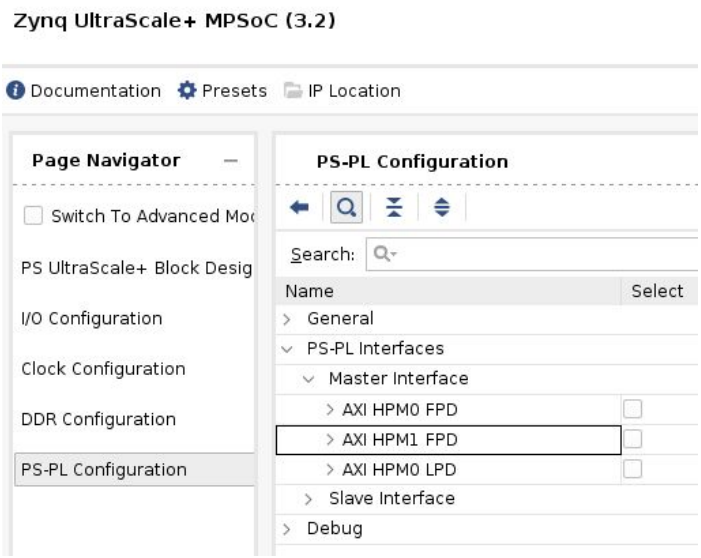
We add the **Processor System Reset**, **Clocking Wizard**, and **Concat** IPs.



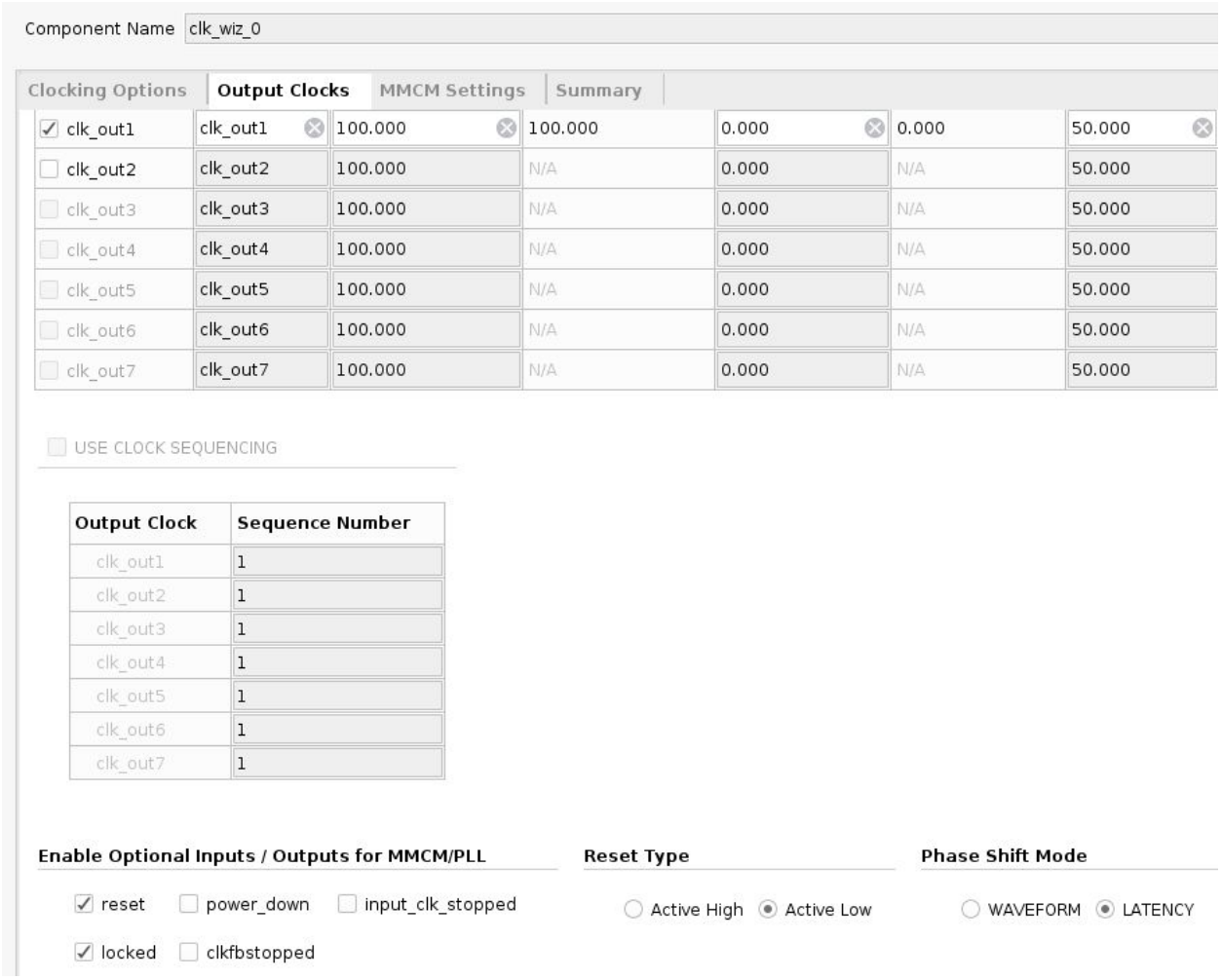
We configure the **Zynq UltraScale+ MPSoC** IP block by double-clicking on it. To enable interrupts from PL to PS we select **PS-PL Configuration** in the **Page Navigator** section of the **Re-customize IP** and we expand at the right of the **Page Navigator** the **General** -> **Interrupts** -> **PL to PS** and we enter **1** in **IRQ0[0-7]** and **IRQ1[0-7]**.



We select again the **PS-PL Configuration** in the **Page Navigator** section of the **Re-customize IP** window and we expand at the right of the **Page Navigator** the **PS-PL Interfaces** -> **Master Interface** and we de-select the **AXI HPM0 FPD** and **AXI HPM1 FPD**. We click on **OK** to close the **Re-customize IP** window.

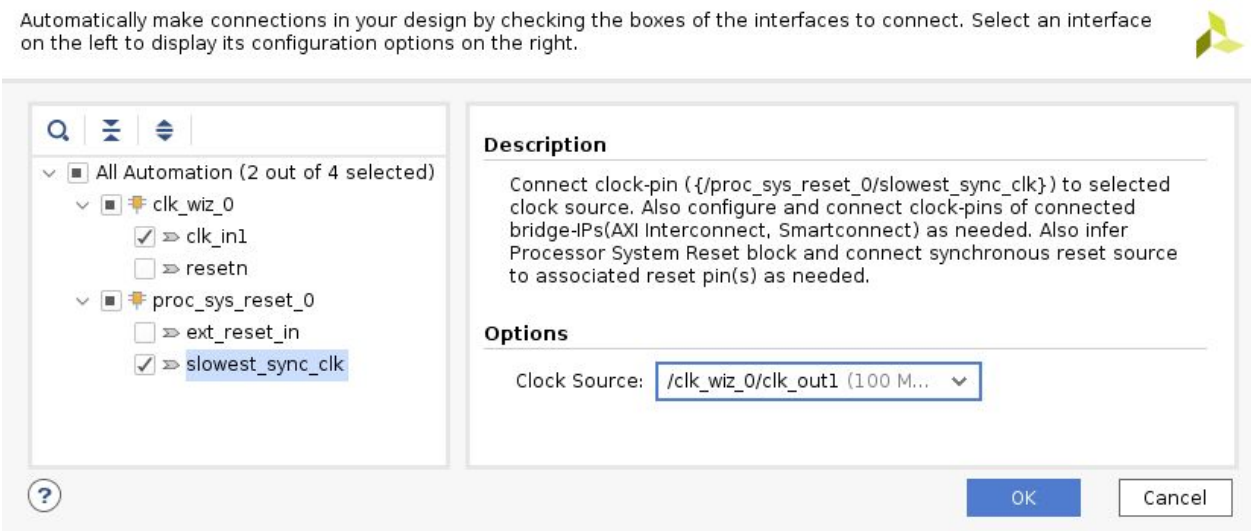


We double-click the **Clocking Wizard** IP. Make sure that in the **Output Clocks** tab the **clk_out1** signal has the value 100MHz. Also make sure that the **Reset Type** signal is selected as **Active Low**. Click **OK** to close the **Re-customize IP** window.



Double click the **Concat** IP and set a value equal to **1** in the **Number of Ports** field. Click **OK** to close the **Re-customize IP** window.

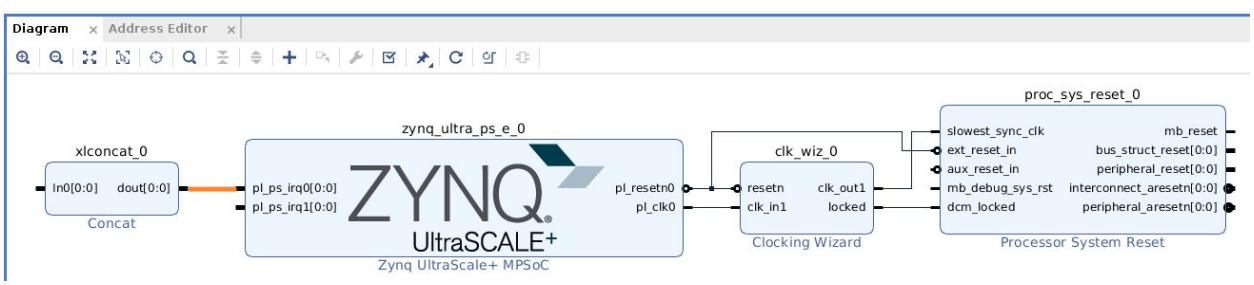
Select the **Run Connection Automation** of the **Designer Assistance** to connect the **Clocking Wizard** and the **Processor System Reset** IP blocks with the **Zynq UltraScale+ MPSoC** IP block. The **clk_in1** port of the **Clocking Wizard** IP must be connected to the **pl_clk0** port of the PS and the **slowest_sync_clk setting** port of the **Processor System Reset** IP must be connected to the **clk_out1** port of the **Clocking Wizard** IP.



We make “by-hand” the rest of the connections - using “drag-n-drop” begining from the port of one IP block and ending to the port of another IP block. We must make the following connections:

- Connect the **pl_resetn0** port of **Zynq UltraScale+ MPSoC** IP to the **resetn** port of the **Clocking Wizard** IP
- Connect the **pl_resetn0** port of **Zynq UltraScale+ MPSoC** IPto the **ext_reset_in** port of the **Processor System Reset** IP
- Connect the **locked** port of **Clocking Wizard** IP to the **dcm_locked** port of the **Processor System Reset** IP
- Connect the **dout[0:0]** port of the **Concat** IP to the **pl_ps_irq0[0:0]** port of the **Zynq UltraScale+ MPSoC** IP

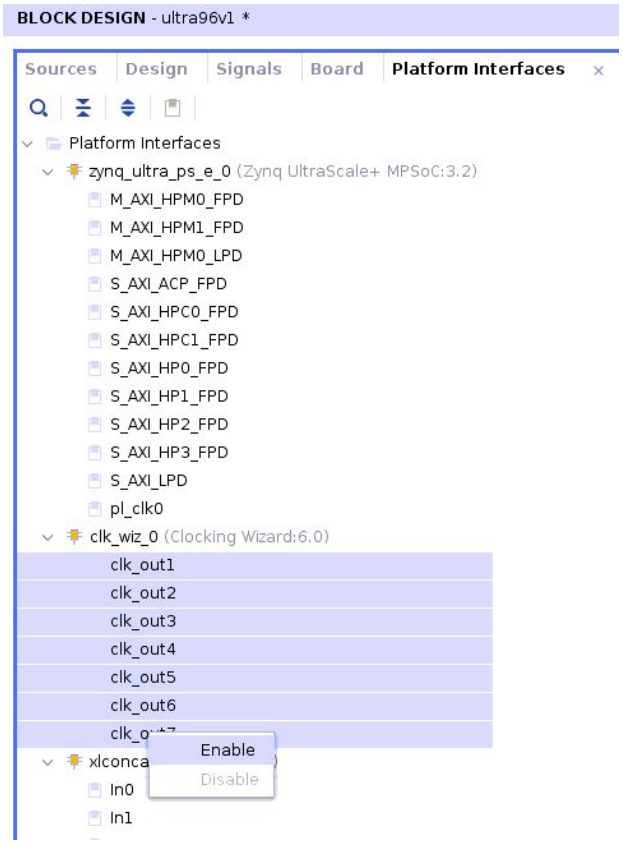
Clicking on the **Regenerate Output** our design must look like the following diagram.



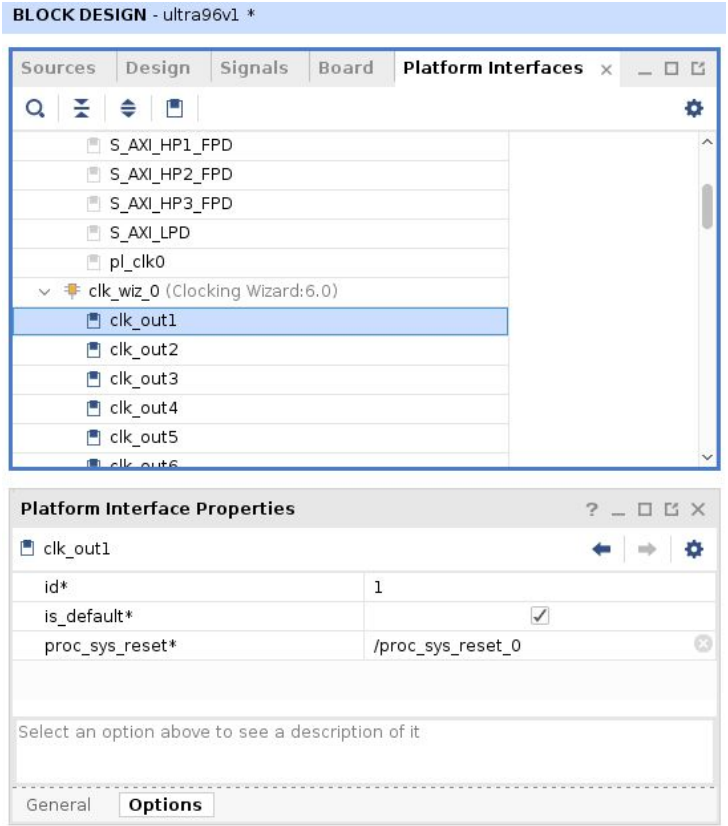
Finally, we click on the **Validate Design** to ensure that our design is error-free.

3. Setting the hardware interfaces

We select **Window** -> **Platform Interfaces** and click on **Enable platform interfaces**.
The **Platform Interfaces** section shows all the port that could be available to our platform. We can choose the ports we would like to include be right-clicking on them and selecting **Enable**.



Using the section **Platform Interface Properties** we can set the default clock signal for use by the SDSoC. We select **clk_out1** (100MHz) and in the **Platform Interface Properties** we select **is_default***.

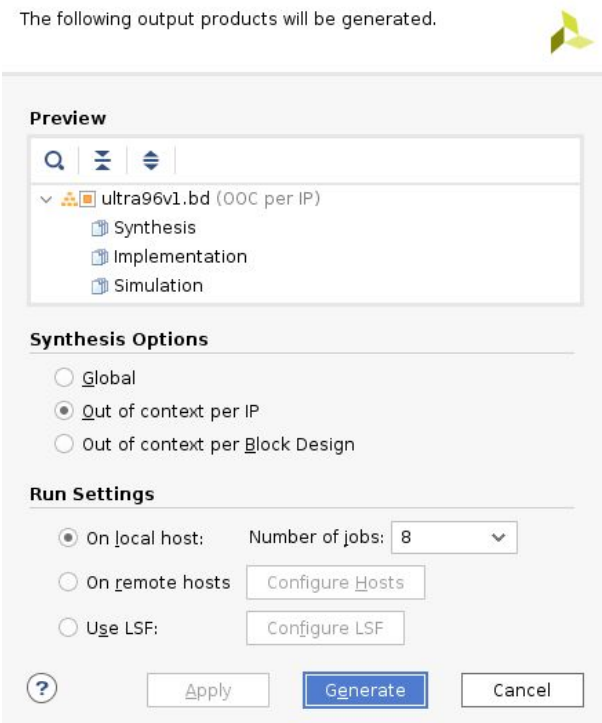


Enable all the AXI interfaces except **pl_clk0**, **S_AXI_ACP_FPD**, και **S_AXI_LPD**.

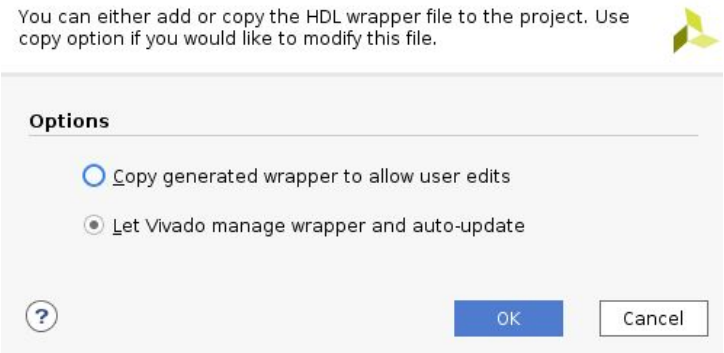
Finally, enable the interrupt ports 0 to 7 of the **Concat** IP.

4. HDL and DSA file generation

In the **Sources** section, we right-click the **ultra96v1 (ultra96v1.bd)** and select **Generate Output Products**. In the window that opens we select **Generate**.

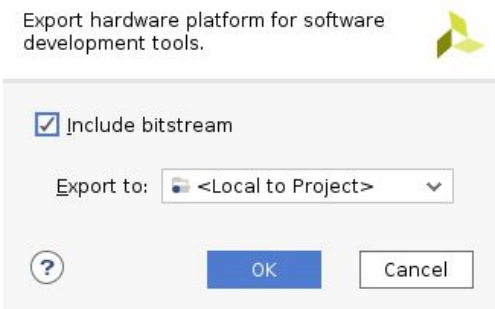


After the previous action's end, we right-click on the **ultra96v1 (ultra96v1.bd)**, select **Create HDL Wrapper**, and click **OK**.



In the **Flow Navigator** section we select **PROGRAM AND DEBUG** -> **Generate Bitstream**. We select **Yes** for executing **synthesis and implementation** and in the next window we click **OK**.

After the end of the previous action, we select **Cancel** in the window that opens. We select **File** -> **Export** > **Export Hardware** from the main menu and check the **Include Bitstream**.



Finally, we execute in the **Tcl Console** the following commands for generating and validating the dsa file:

write_dsa -force -include_bit <τοποθεσία_αρχείου>/όνομα_αρχείου.dsa

validate_dsa <τοποθεσία_αρχείου>/όνομα_αρχείου.dsa

e.g.

write_dsa -force -include_bit /mnt/terabyte/pmousoul_data/projects/vivado_2018.2/ultra96v1/ultra96v1.dsa

validate_dsa /mnt/terabyte/pmousoul_data/projects/vivado_2018.2/ultra96v1/ultra96v1.dsa

Close Vivado.

5. PetaLinux build

1. `petalinux-create -t project -n <project_name> -s <path_to_base_BSP>` , [e.g. `petalinux-create -t project -n ultra96v1 -s ultra96v1_full_2018_2.bsp`]

2. `petalinux-config -p <project_name> --get-hw-description=<HDF_path>` , [e.g. `petalinux-config -p ultra96v1 --get-hw-description=./`]

3. `petalinux-config -p <project_name> -c kernel`
   - Device Drivers → Generic Driver Options → Size in MB(1024)
   - Device Drivers → Staging drivers (ON)
   - Device Drivers → Staging drivers → Xilinx APF Accelerator driver (ON)
   - Device Drivers → Staging drivers → Xilinx APF Accelerator driver → Xilinx APF DMA engines support (ON)
   - CPU Power Management → CPU idle → CPU idle PM support (OFF)
   - CPU Power Management → CPU Frequency scaling → CPU Frequency scaling (OFF)

4. `petalinux-config -p <project_name> -c rootfs`
    - Filesystem Packages → misc → gcc-runtime → libstdc++ (ON)

5. `nano ultra96v1/project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi`

```
/{
    xlnk {
    compatible = "xlnx,xlnk-1.0";
    };
};
```

6. `petalinux-build -p <project_name>`, για να κάνουμε build την PetaLinux διανομή.

6. Platform Creation

Prepare the PetaLinux files:

```
cd <petalinux_project>/images/linux
mkdir boot
mkdir image
cp u-boot.elf boot/u-boot.elf
cp *fsbl.elf boot/fsbl.elf
cp bl31.elf boot/bl31.elf
cp pmufw.elf boot/pmufw.elf
cp image.ub image/image.ub
```

Create file:

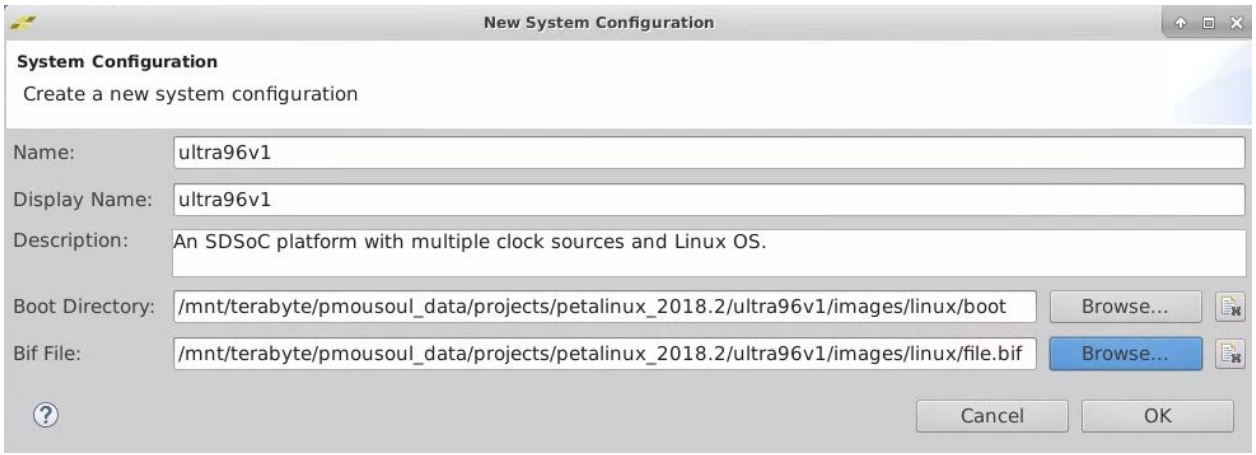
```
nano file.bif
```

```
the_ROM_image:
{
    [fsbl_config] a53_x64
    [bootloader]<fsbl.elf>
    [pmufw_image]<pmufw.elf>
    [destination_device=pl] <bitstream>
    [destination_cpu=a53-0, exception_level=el-3, trustzone] <bl31.elf>
    [destination_cpu=a53-0, exception_level=el-2] <u-boot.elf>
}
```

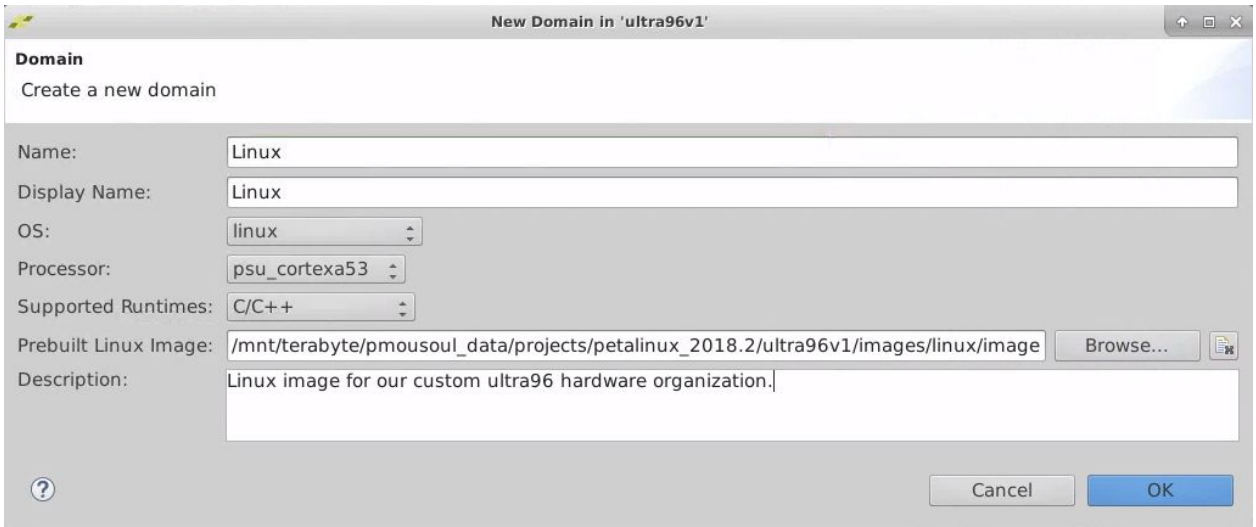
Open SDSoC and select **File -> New SDx Project**. We select **Platform** and click **Next**.

In the next window we set the location of the DSA file we created. We also select **Import Software Platform Components** and then we select **Finish**.

We click on the **Define System Configuration** and we type the information displayed in the next figure.



We click **OK** to close the window and we click on the **Add Processor Group/Domain**. we type the information displayed in the next figure.



Next, we select the **Generate Platform** and click **OK** in the window that opens. Finally we select the **Add to Custom Repositories** in order to make the platform available in SDSoC and click **OK**.