

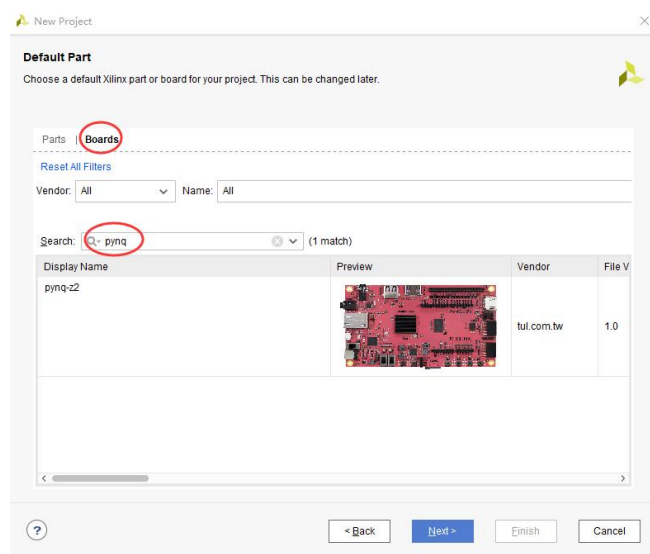
实验前准备：将板子正确连接电脑的 USB 接口，并且打开板子电源开关，注意两个 HDMI 接口一个为 HDMI IN,一个为 HDMI OUT，HDMI OUT 是连接显示器的，HDMI IN 是连接摄像头的，一定不能接反。板子中的“Xilinx 大学计划”文字下面的跳线连接为 USB，板子上的 HDMI IN 接口下方的跳线连接为 JTAG

1. 添加板卡文件。由于本次实验采用的是 PYNQ Z2 板子，Vivado 中默认库中是没有此板子的型号的，所以第一次打开时，需要先添加板卡文件。（只要没重置电脑，后续就不需要再添加了）

操作：首先找到 Vivado 的安装目录，将 pynq-z2 文件夹放入到“PATH\data\boards\board_parts\zynq”，如图

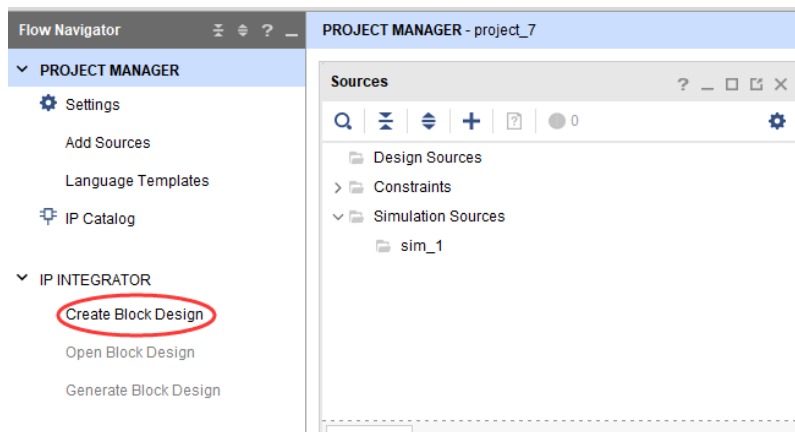


2. 打开 Vivado，依次选择 Create Project > Next ,修改项目名称和项目位置，点击 Next，选择 RTL Project，并打勾 Do not specify sources at this time>Next,点击上面的 Boards，如图



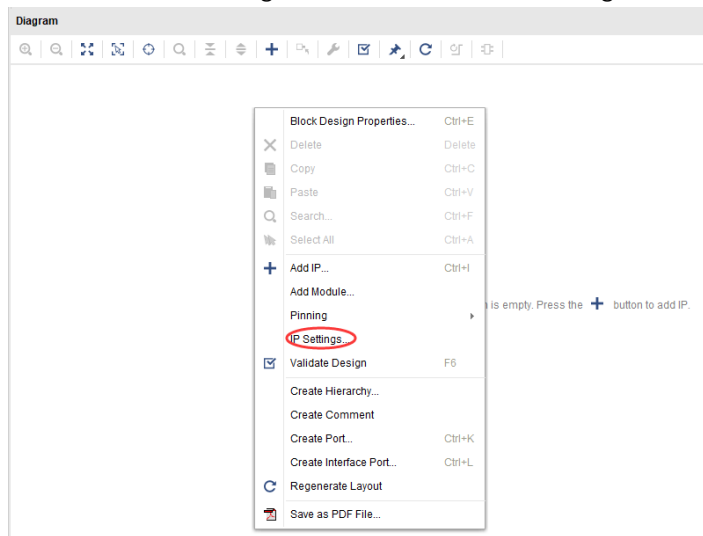
点击 pynq-z2>Next>Finish。

- 3.点击左侧 IP INTERATOR/Create Block Design

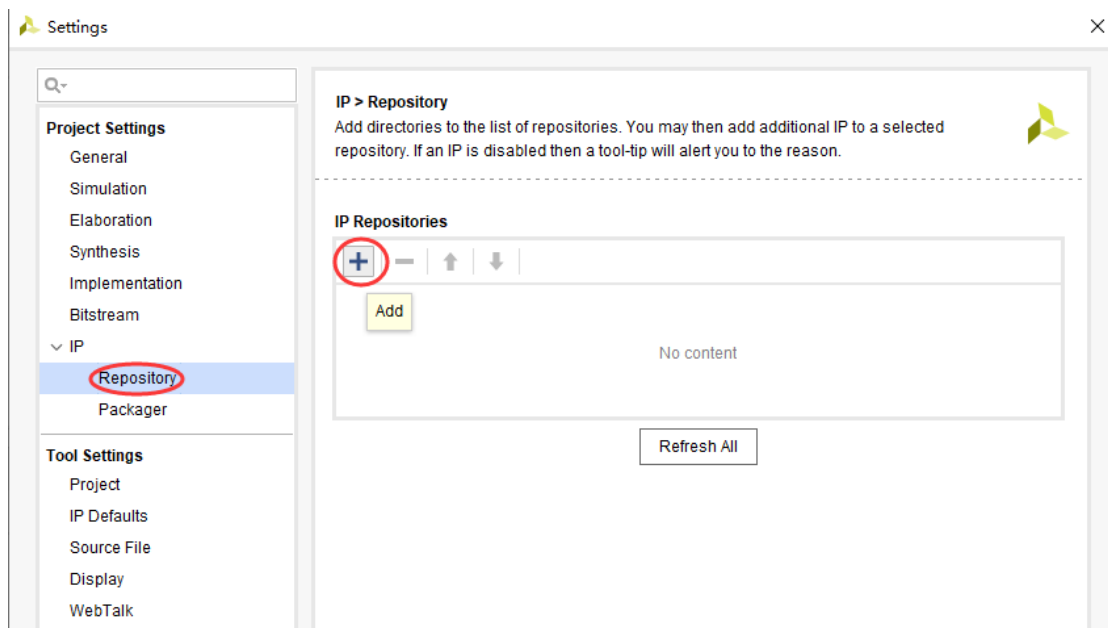


修改模块名称，点击 OK。

3. 添加 IP 核。右键 Diagram 空白处，点击 IP setting。

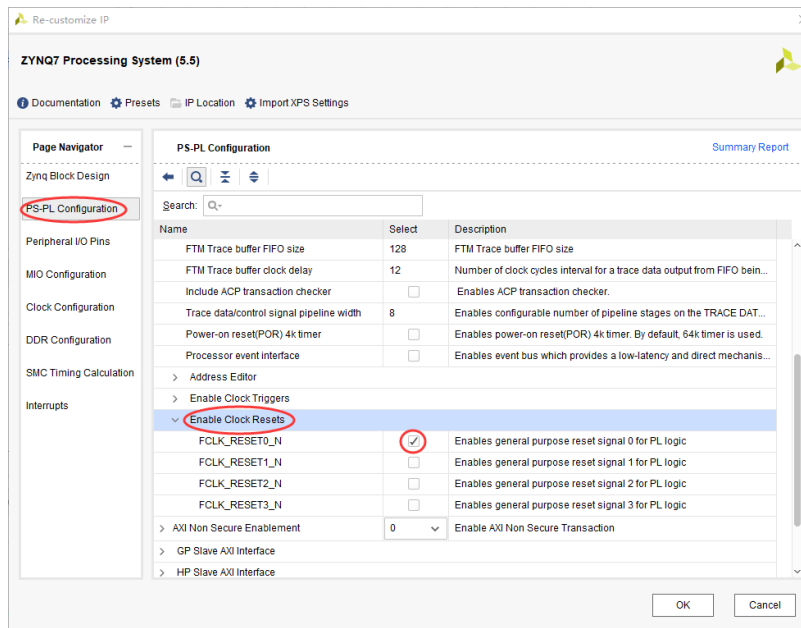


点击 IP/Repository/Add

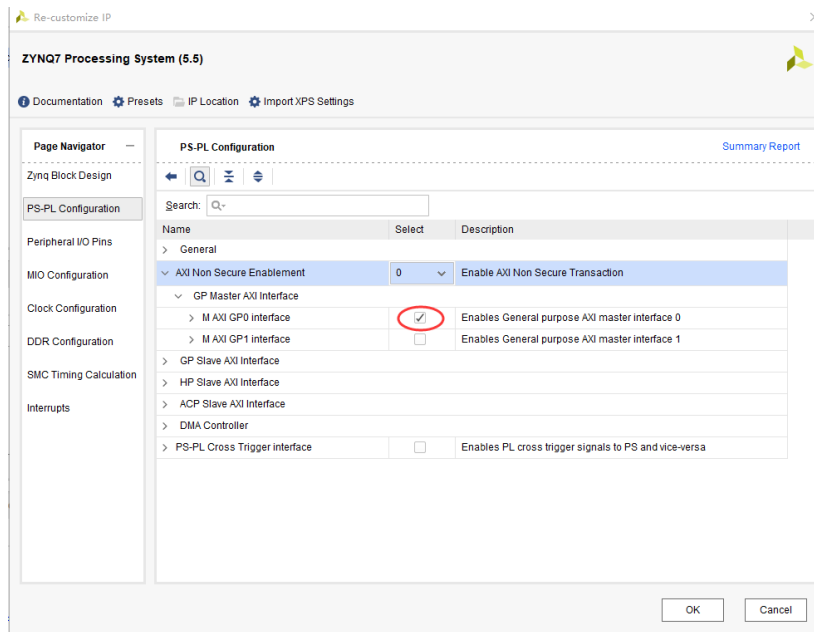


选中 HDMI/IP/vivado-library，点击 Select> OK>Apply>OK

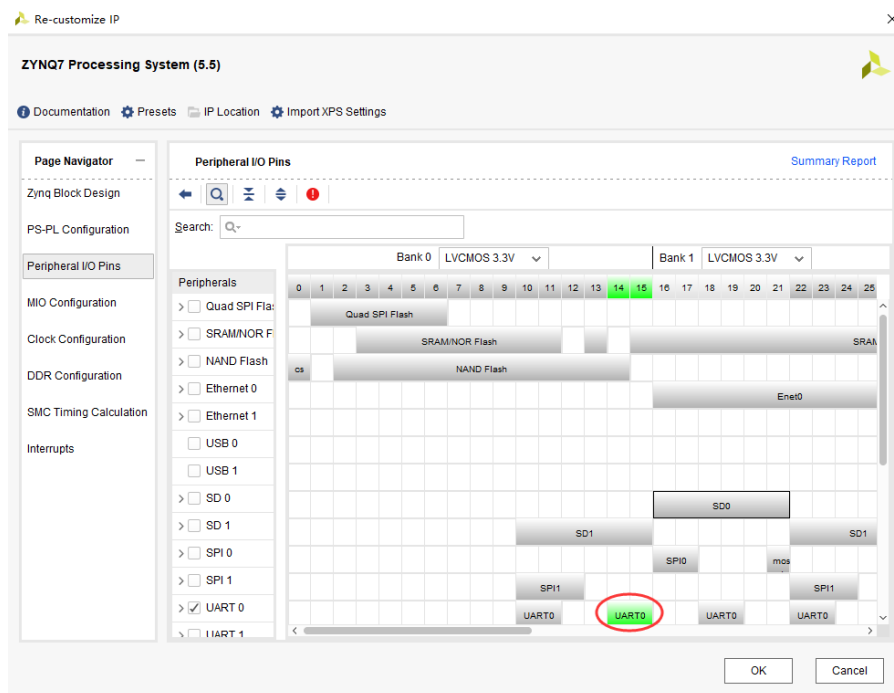
- 右键 Diagram 空白处, Add IP, 搜索 zynq7, 双击 ZYNQ7 Processing System 添加 IP。
- 双击刚刚添加的 ZYNQ7 Processing System 模块进行配置。
5.1 点击左侧的 PS-PL Configuration>General>Enable Clock Resets,将 FCLK_RESET0_N 的 ✓ 取消。



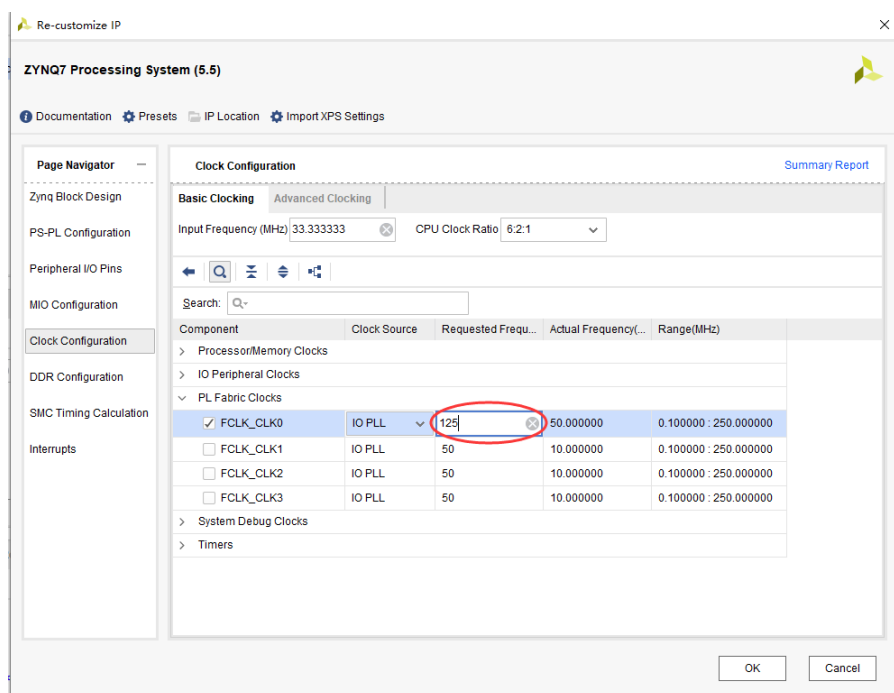
- 5.2 同样在 PS-PL Configuration 中, AXI Non Secure Enablement>GP Master AXI Interface>M AXI GP0 interface, 将其对勾取消。



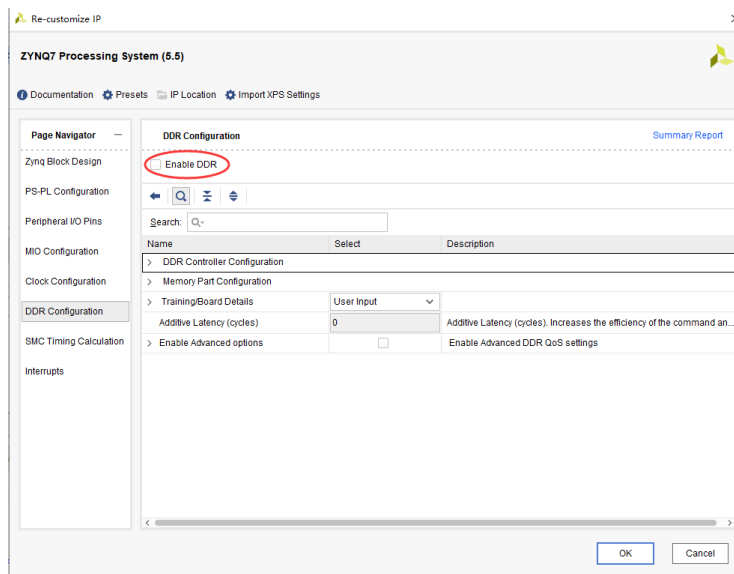
- 5.3 点击左侧 Peripheral I/O Pins, 点击表格中的 UART0



5.4 点击左侧 Clock Configuration>PL Fabric Clocks,将 FCLK_CLK0 的 Request Frequency 由 50 修改为 125.

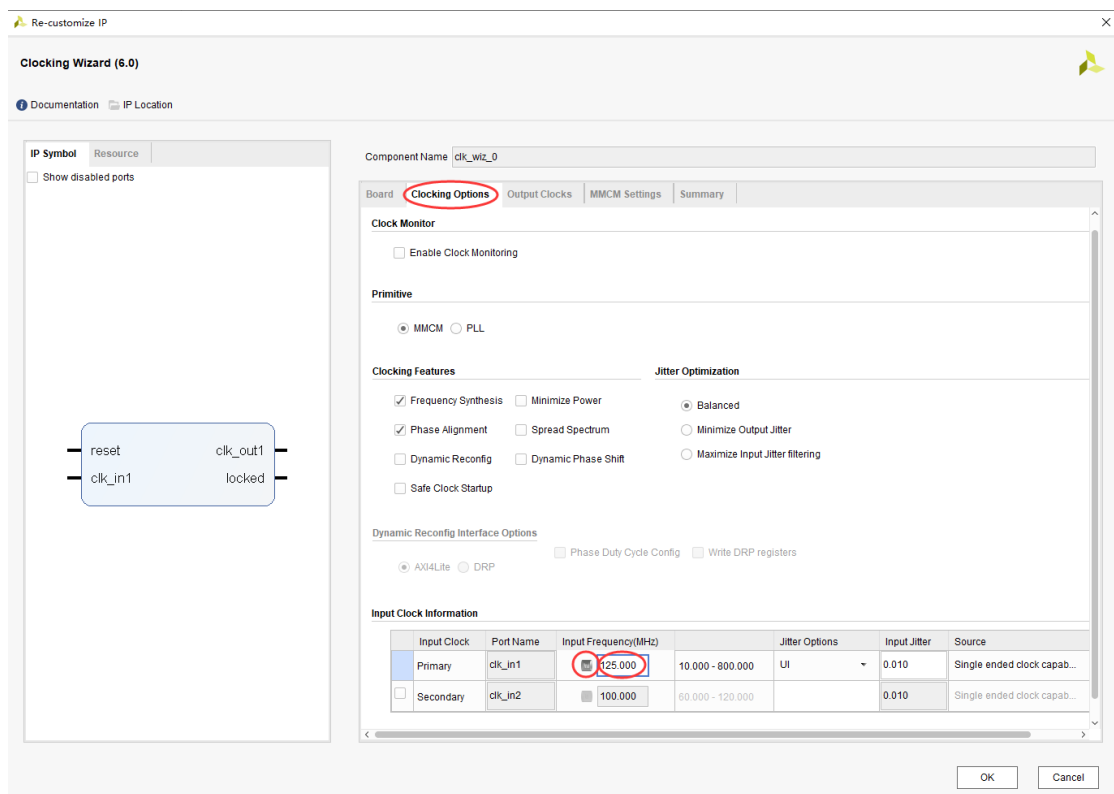


5.5 点击左侧 DDR Configuration, 将 Enable DDR 的√ 取消。



点击 OK，此模块配置完毕。

6. 右键 Add IP，搜索 clock，双击选择 Clocking Wizard。
- 6.1 双击生成的 Clocking Wizard 模块进行配置。点击 Clocking Options，点击下方的 Input Frequency 下面的按钮，并对后面的数据进行更改，将 100 修改为 125。



6.2 点击 Output Clock，将 clk_out1 的 Output Freq 中的 Requested，将 100 修改为 200

Component Name: clk_wiz_0

Board | Clocking Options | **Output Clocks** | MMCM Settings | Summary

The phase is calculated relative to the active input clock.

Output Clock	Port Name	Output Freq (MHz)		Phase (degrees)		Duty Cycle (%)		Drives
		Requested	Actual	Requested	Actual	Requested	Actual	
<input checked="" type="checkbox"/> clk_out1	clk_out1	200.000	200.000	0.000	0.000	50.000	50.0	BUFG
<input type="checkbox"/> clk_out2	clk_out2	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out3	clk_out3	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out4	clk_out4	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out5	clk_out5	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out6	clk_out6	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
<input type="checkbox"/> clk_out7	clk_out7	100.000	N/A	0.000	N/A	50.000	N/A	BUFG

点击 OK，此模块配置完成。

7. 右键 Add IP，搜索 Utility，双击选择 Utility Vector Logic。

7.1 双击生成的 Utility Vector Logic 进行配置.C_SIZE 设置为 1，C_OPERATION 设置为 not，点击 OK。

Re-customize IP

Utility Vector Logic (2.0)

Documentation | IP Location

☐ Show disabled ports

Op1[0:0] Res[0:0]

Component Name: util_vector_logic_0

C_SIZE: 1

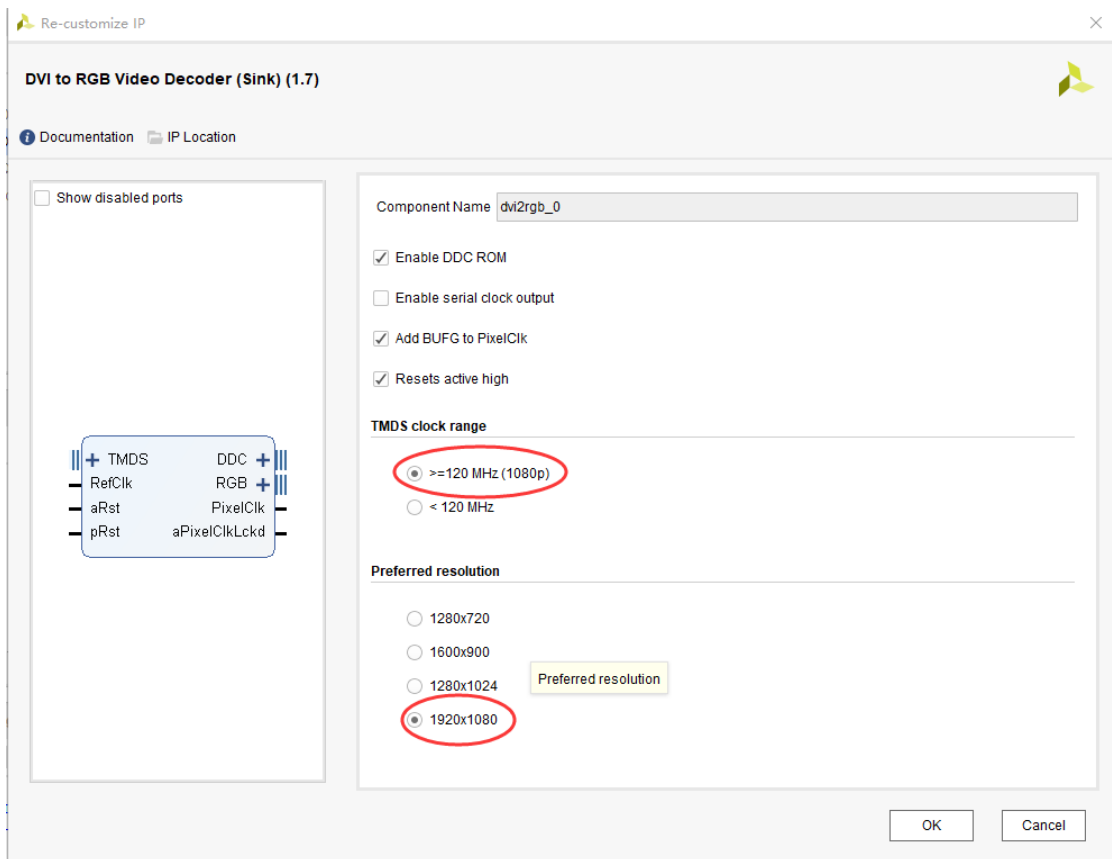
C_OPERATION

☐ and
☐ or
☐ xor
☒ not

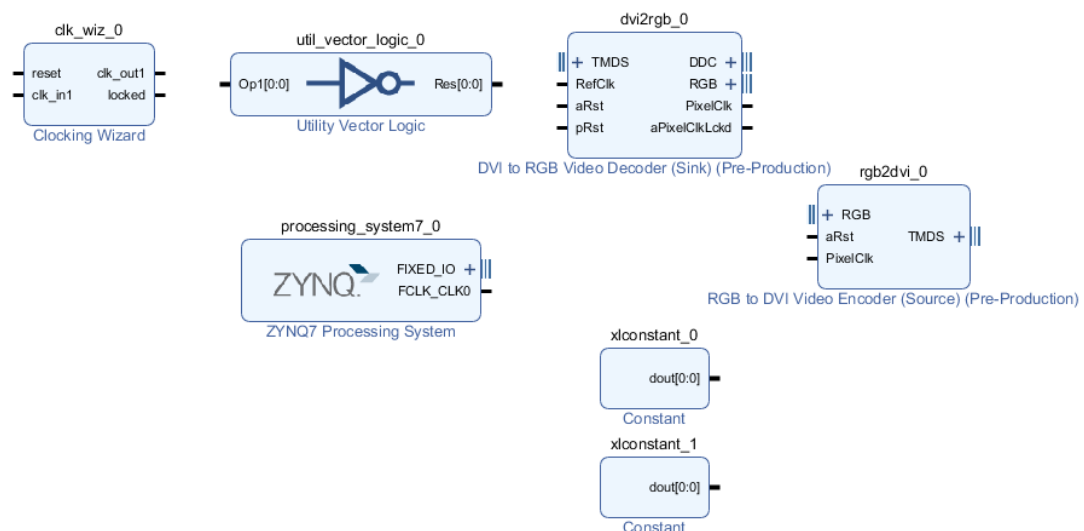
OK Cancel

8. 右键 Add IP，搜索 DVI，双击选择 DVI to RGB Video Decoder(Sink)。

8.1 双击生成的 DVI to RGB Video Decoder(Sink)进行配置。将 TMDS clock range 设置为 $\geq 1200\text{MHz}(1080p)$, Preferred resolution 设置为 1920×1080 ，点击 OK。

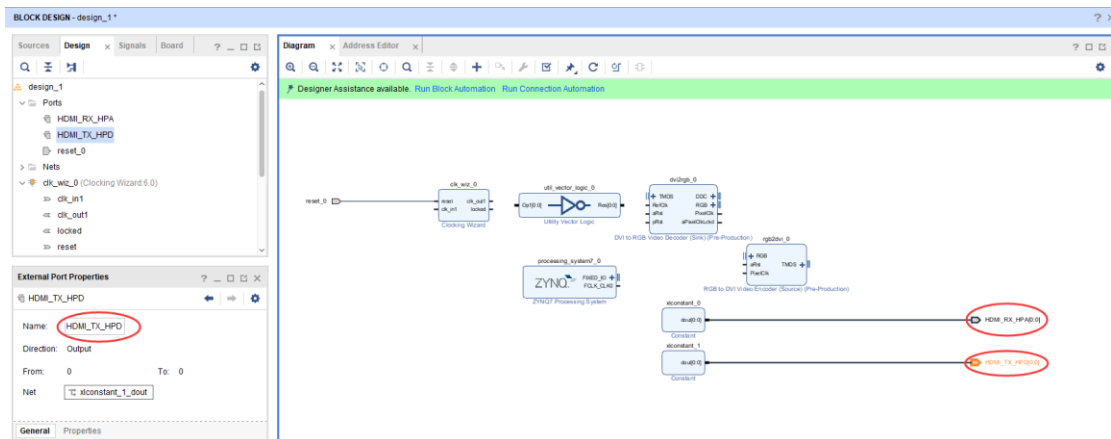


9. 右键 Add IP, 搜索 RGB, 双击选择 RGB to DVI Video Encoder(Source).
10. 右键 Add IP, 搜索 Constant, 双击选择 Constant
11. 重复步骤 10, 再生成一个 Constant。
12. 将 7 个模块按如图位置摆好 (大体相当就可, 方便后续连线)



13. 接下来开始连线。首先生成外部管脚。
 - 13.1 左键点击 clk_wiz_0 模块的 reset 名称, (注意是仅 reset 引脚变橙色才正确, 如果是整个模块都变橙色说明你选中的是整个模块, 需要重选, 如果太小了可以 Ctrl+鼠标滚轮控制缩放), reset 和其引脚选中变橙色后, 按住 Ctrl+T, 生成管脚。
 - 13.2 同样的方法, 在两个 Constant 模块的 dout[0:0]引脚处生成管脚, 并点击生成的管

脚，可以对其重命名，将两个管脚分别重命名为 HDMI_RX_HPA 和 HDMI_TX_HPD，如图



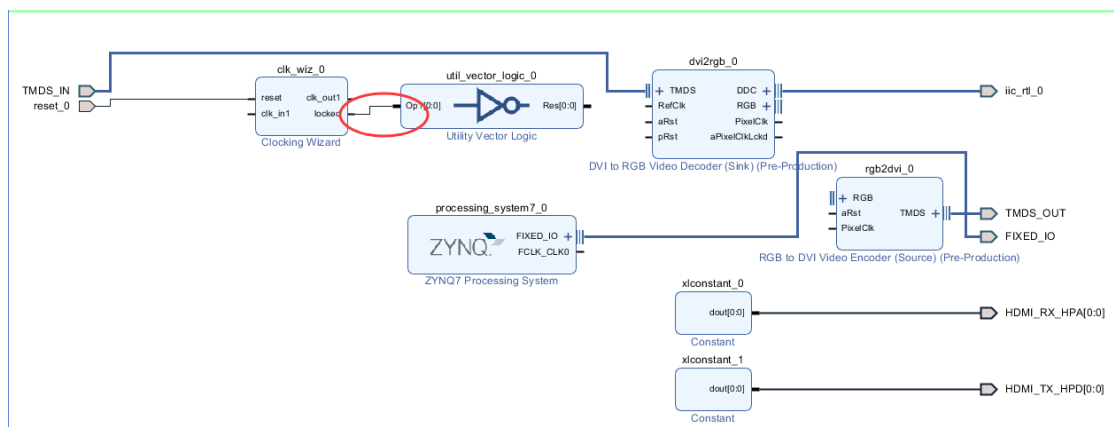
13.3 在 rgb2dvi_0 模块中的 TMDS 引脚处生成管脚，并重命名为 TMDS_OUT

13.4 在 dvi2rgb_0 模块中的 DDC 引脚处生成管脚，并重命名为 iic_rtl_0

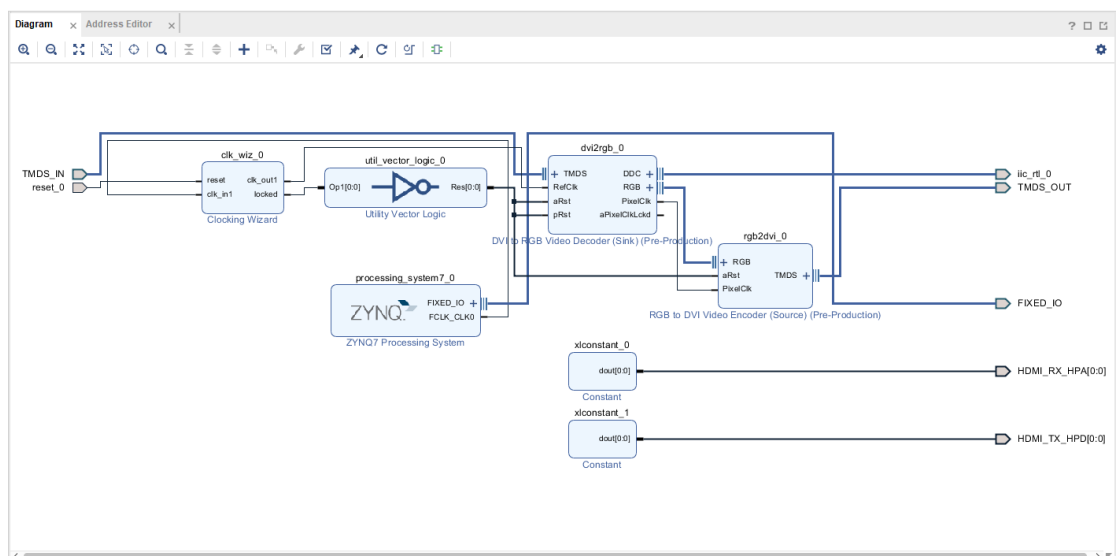
13.5 在 dvi2rgb_0 模块中的 TMDS 处生成管脚，并重命名为 TMDS_IN

13.6 在 ZYNQ7 Processing System 模块中的 FIXED_IO 处生成管脚，并重命名为 FIXED_IO

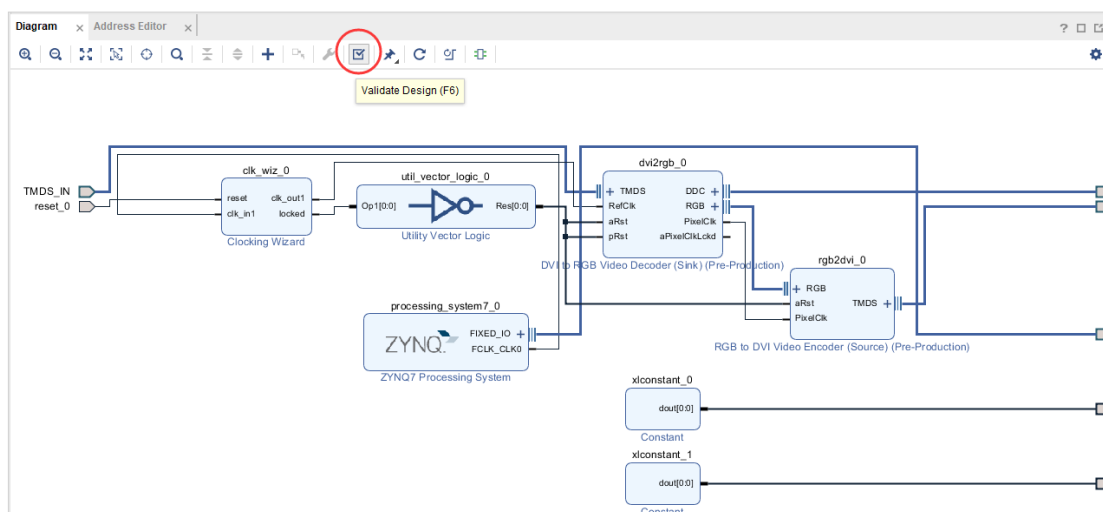
13.7 点击 clk_wiz_0 的 locked 的引脚按住鼠标连接到 util_vector_logic_0 的 Op1，生成连线，如图



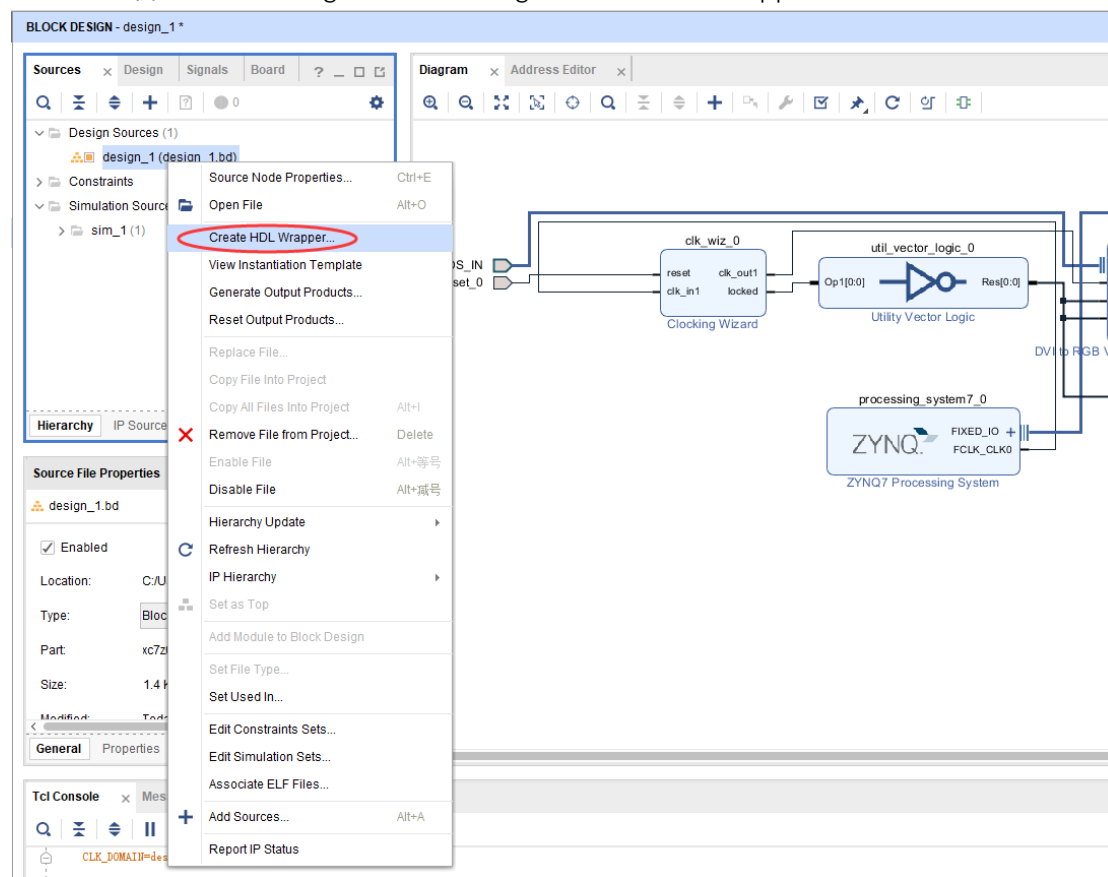
13.8 按照最终图形完成后续连线，最终图形如下（可以看 Block Design.pdf 文件）



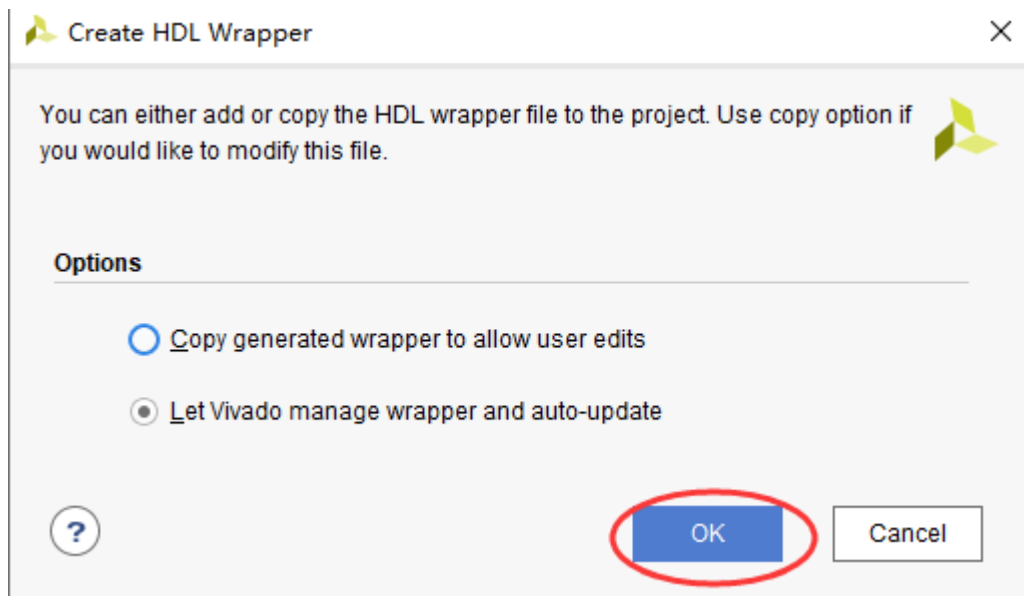
13.9 点击上方验证按钮，若出现成功，则完成验证



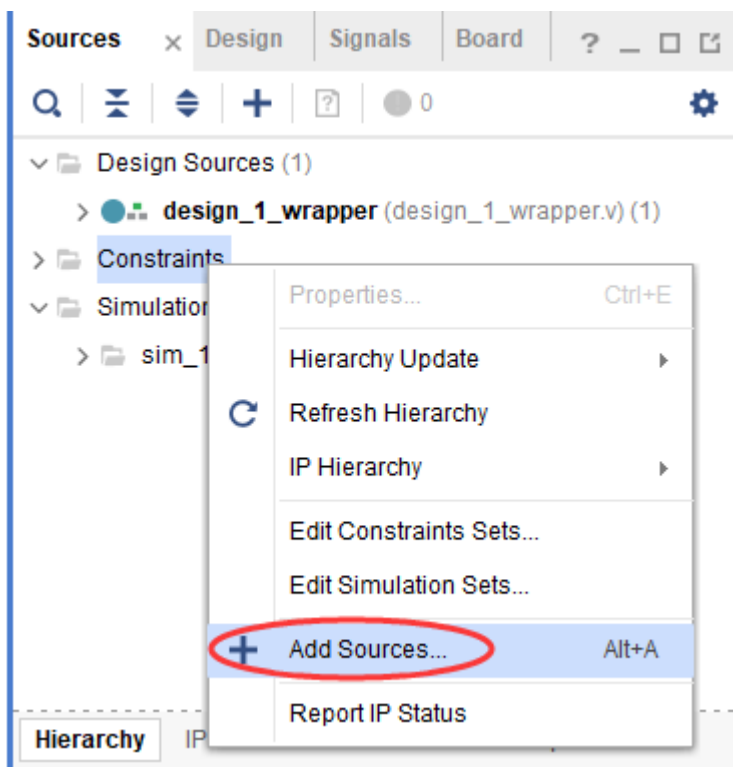
14 右键左侧 Sources/Design Sources/design>Create HDL Wrapper



14. 点击 ok

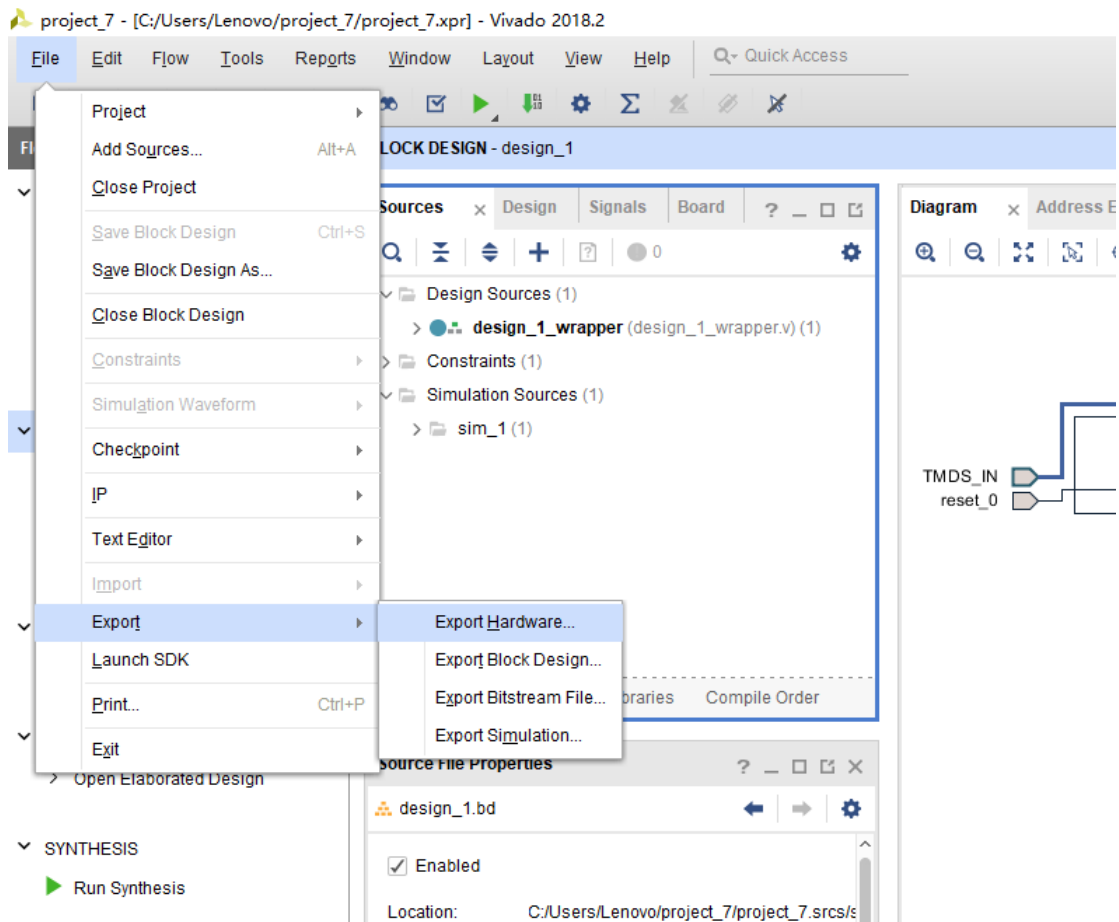


15. 右键 Sources/Constraints>Add Sources, 添加约束文件。

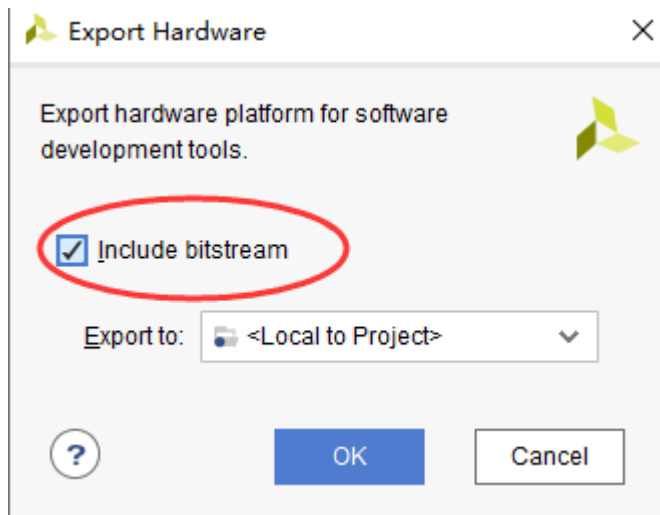


选择 Add or create constraints, 点击 Next, 点击 Add Files, 选择 HDMI/HDMI.xdc, 点击 OK, 点击 Finish, 完成添加约束文件。

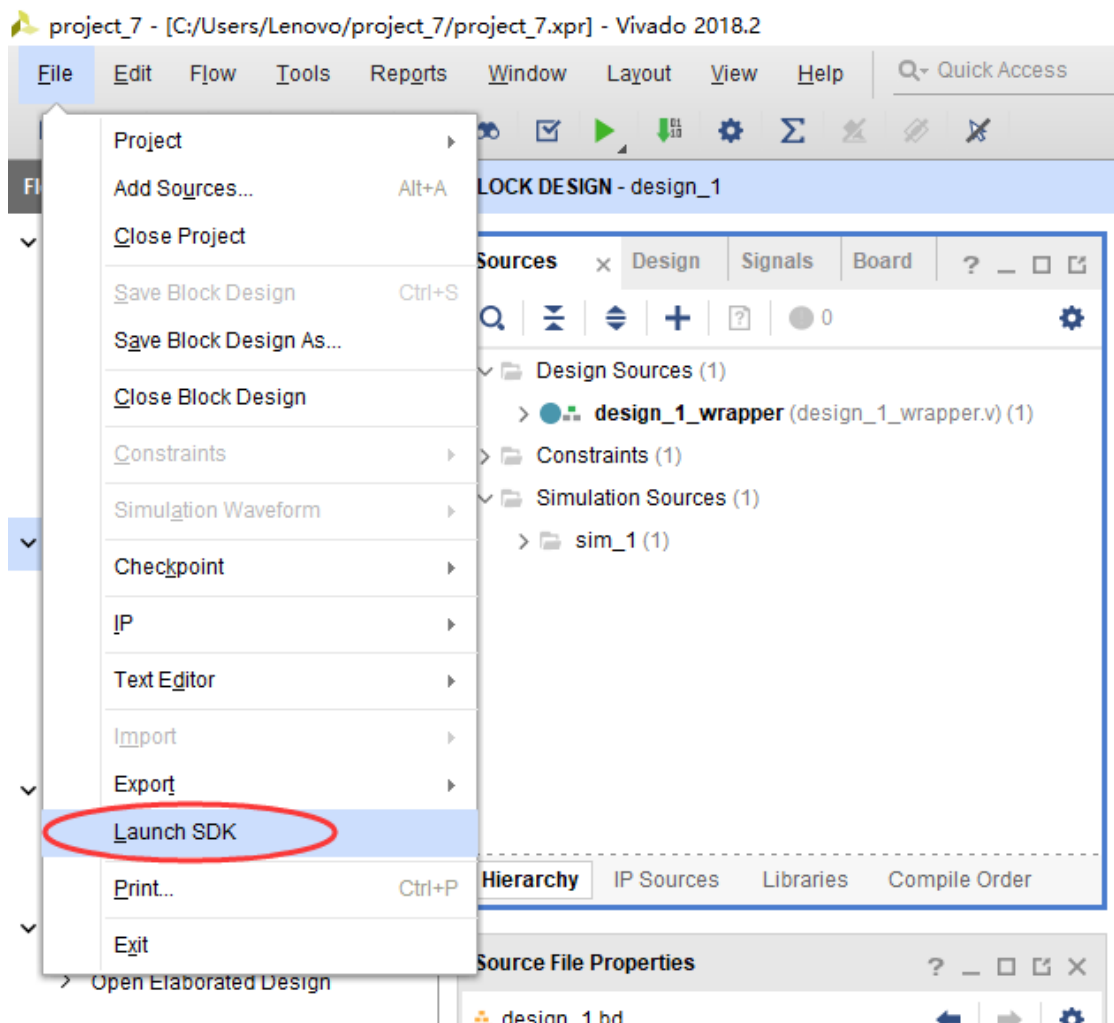
16. 点击左侧的 PROGRAM AND DEBUG/Generate Bitstream, 点击 OK>OK, 进行生成比特流。
(这一步耗费的时间比较长, 可以看右上角完全 ready 了, 再进行下一步)
17. 跳出生成比特流成功的对话框后, 点击 Cancel。点击左上角的 File/Export/Export Hardware



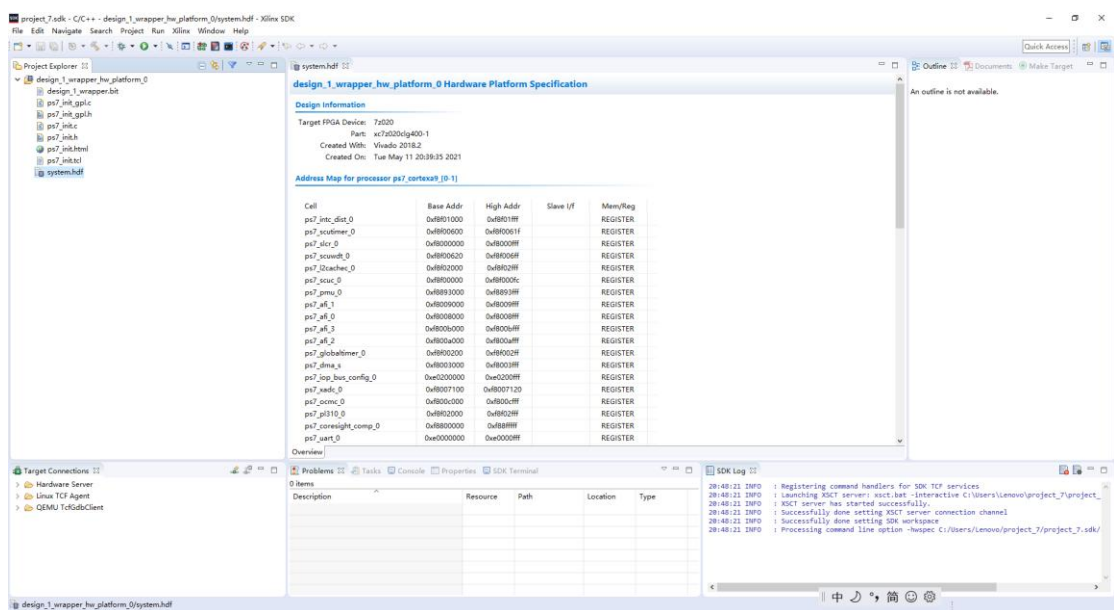
18. 勾选 Include bitstream, 点击 OK



19. 点击 File/Launch SDK, 对话框选择 OK。(这一步需要启动另一个软件 SDK, 需要一点时间)



20. 成功打开 SDK 软件界面如图



21. 点击左上 File/New/Application Project, 对话框中 Project name 写入 Hello world, 点击 Next。

SDK New Project

Application Project

Create a managed make application project.

Project name:

☒ Use default location

Location:

Choose file system:

OS Platform:

Target Hardware

Hardware Platform:

Processor:

Target Software

Language: ☒ C ☐ C++

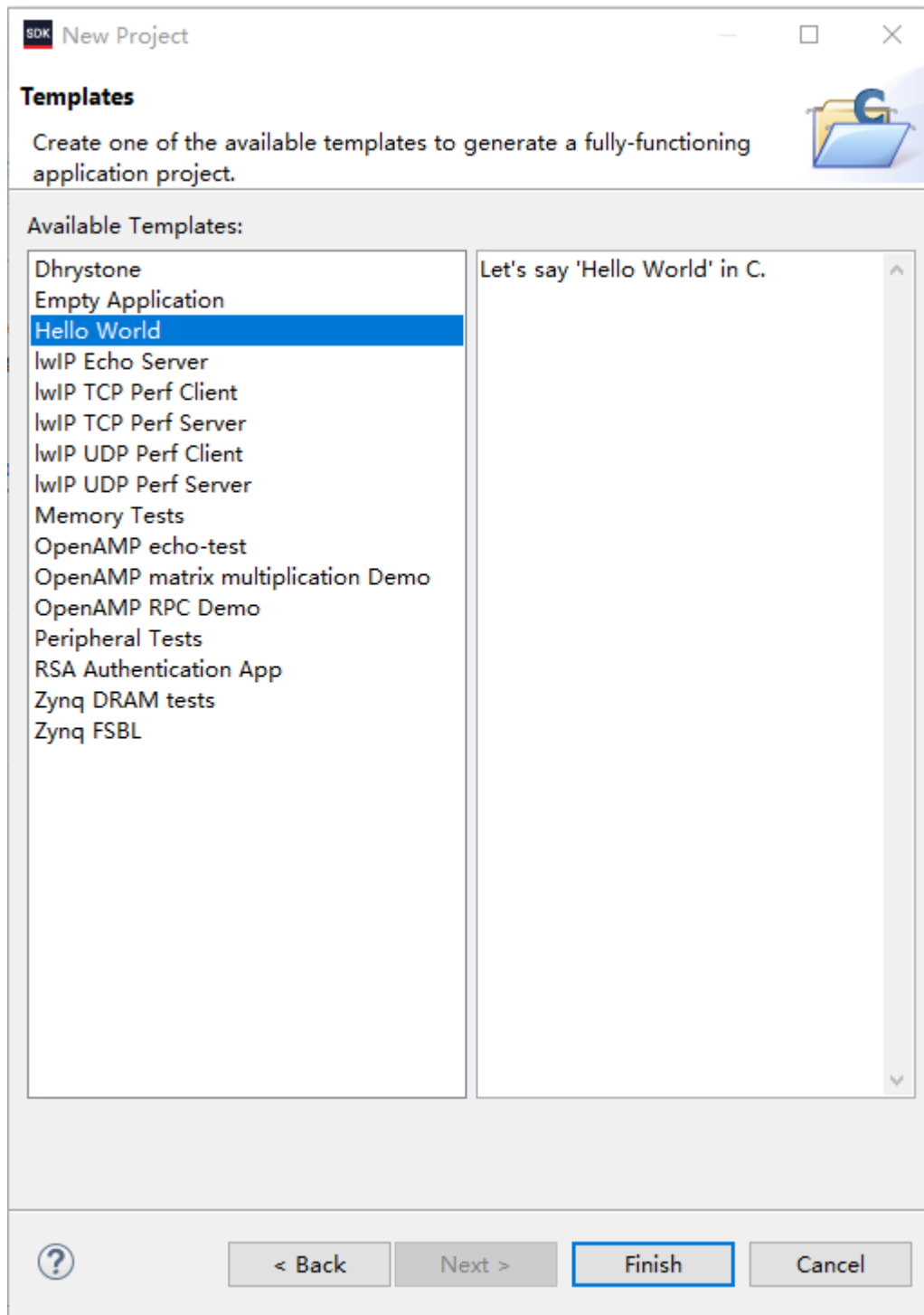
Compiler:

Hypervisor Guest:

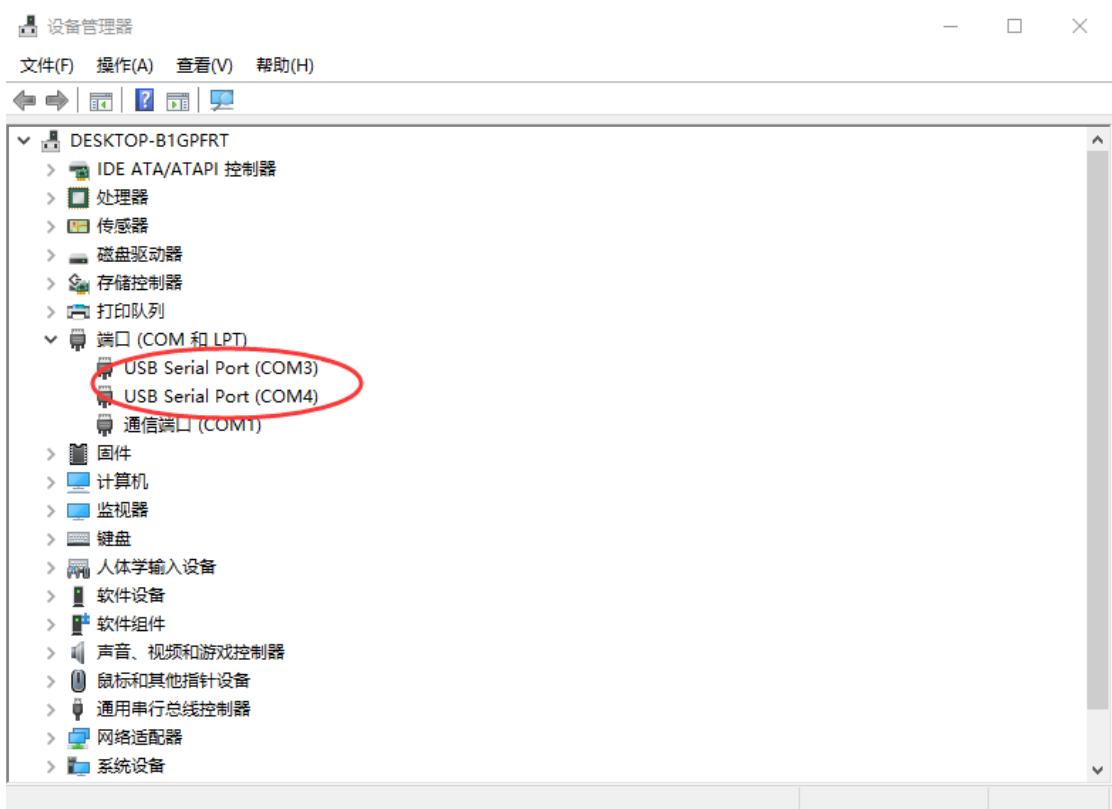
Board Support Package: ☒ Create New

☐ Use existing

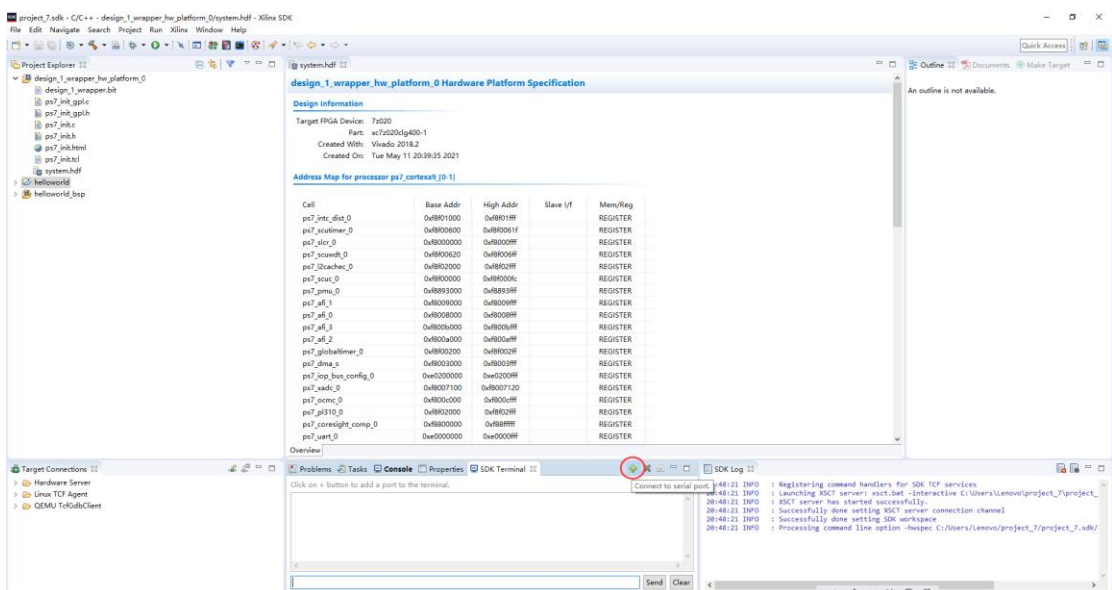
22.选择 Hello World, 点击 Finish。



22. 按住 win+r 键，输入 devmgmt.msc，按回车调出设备管理器，点击端口，查看板子的端口号



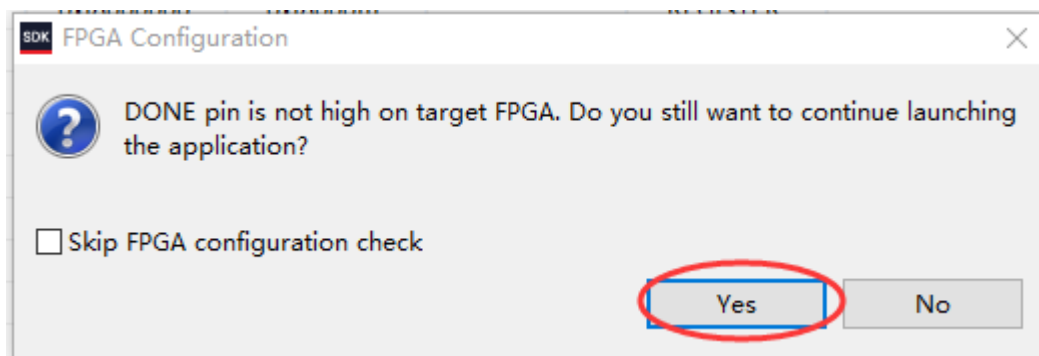
23. 返回 SDK 软件中，点击下方的 '+' 按钮



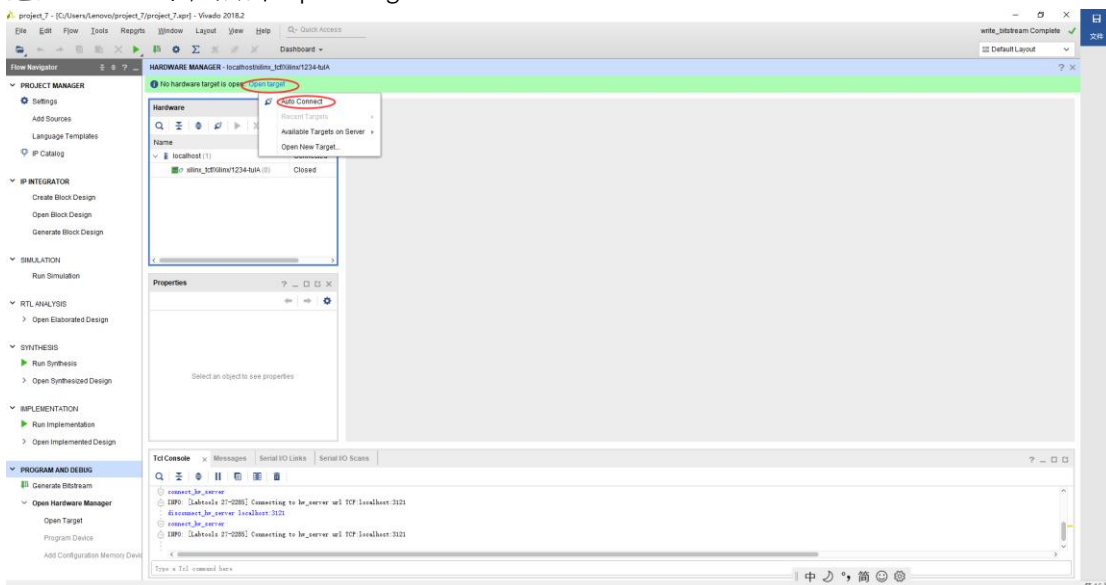
24. 对话框中 Port 选择为刚刚设备管理器中的 USB 串行接口端口，点击 OK。

25. 右键左侧的刚刚你建立的 helloworld 工程文件夹，Run As/Launch on Hardware (System Debugger)

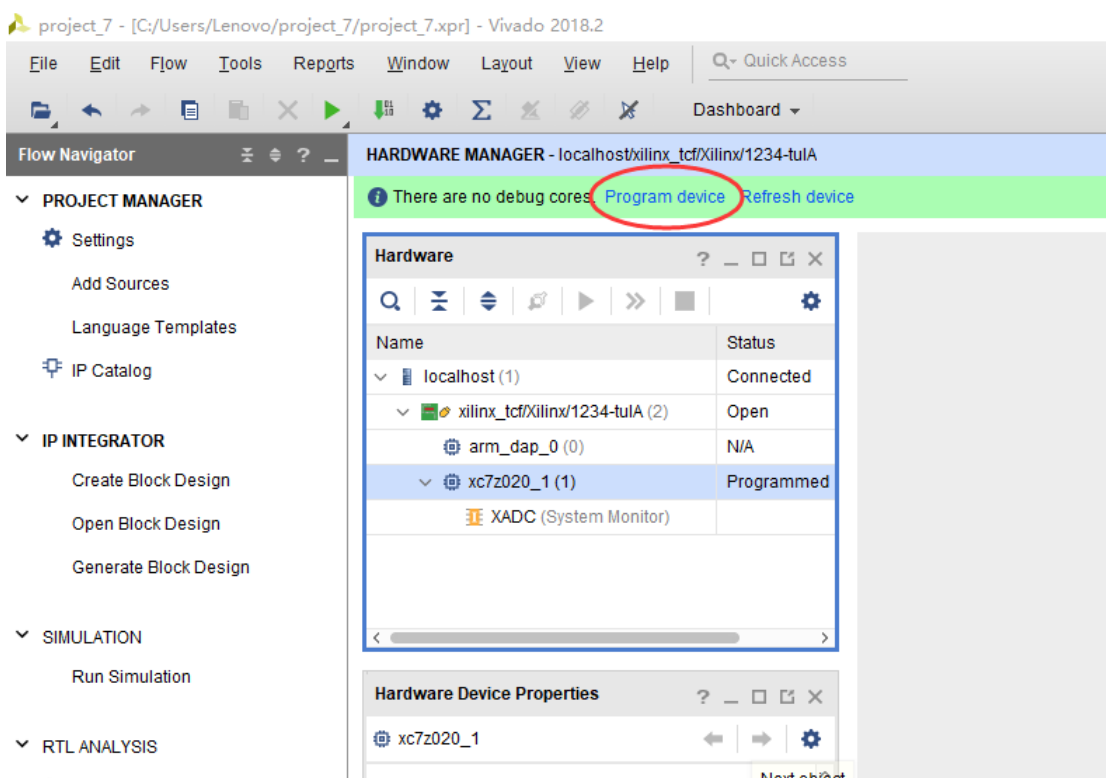
26. 跳出的对话框选 Yes



27. 返回 Vivado 中，点击 Open target



28. 点击 Program device, 点击 OK



29. 观察显示器输出，如果能够正常输出图像，则表示 HDMI IN2OUT 实验成功。你可以自行添加代码完成对图像的处理，然后从显示器输出。