

实验六、VGA 显示

实验介绍:

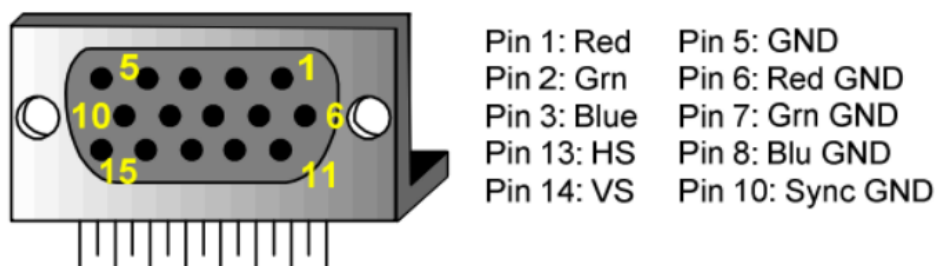
本实验在 EGo 数模混合口袋实验平台上通过 VGA 接口显示图片，VGA 控制器采用 $640 \times 480 @ 60\text{Hz}$ 的标准，要显示的 Logo 图案预先存储在 ROM 中。最终显示效果为 Logo 图片在显示器上游动，到达边界自动反弹。

实验目的:

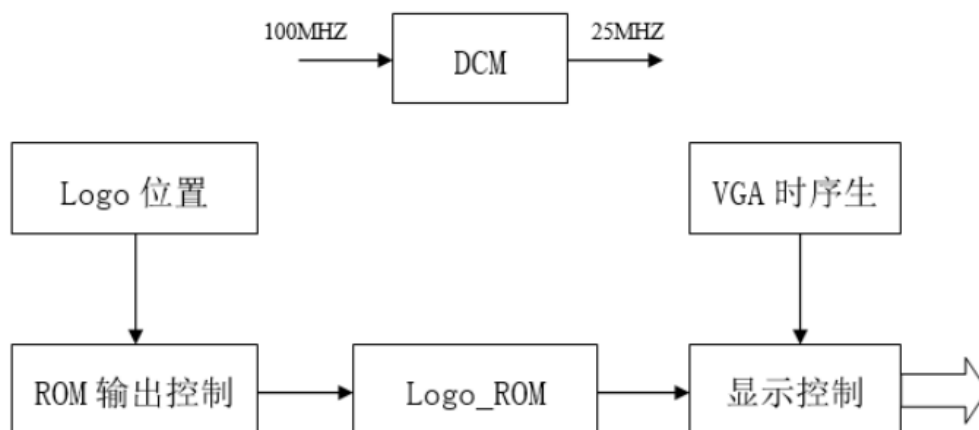
1. 学习使用 Xilinx Vivado 开发工具完成 FPGA 设计
2. 学习 VGA 显示原理、VGA 时序等相关知识
3. 熟悉 EGo 数模混合口袋实验平台上的 VGA 等接口外设的使用

实验原理:

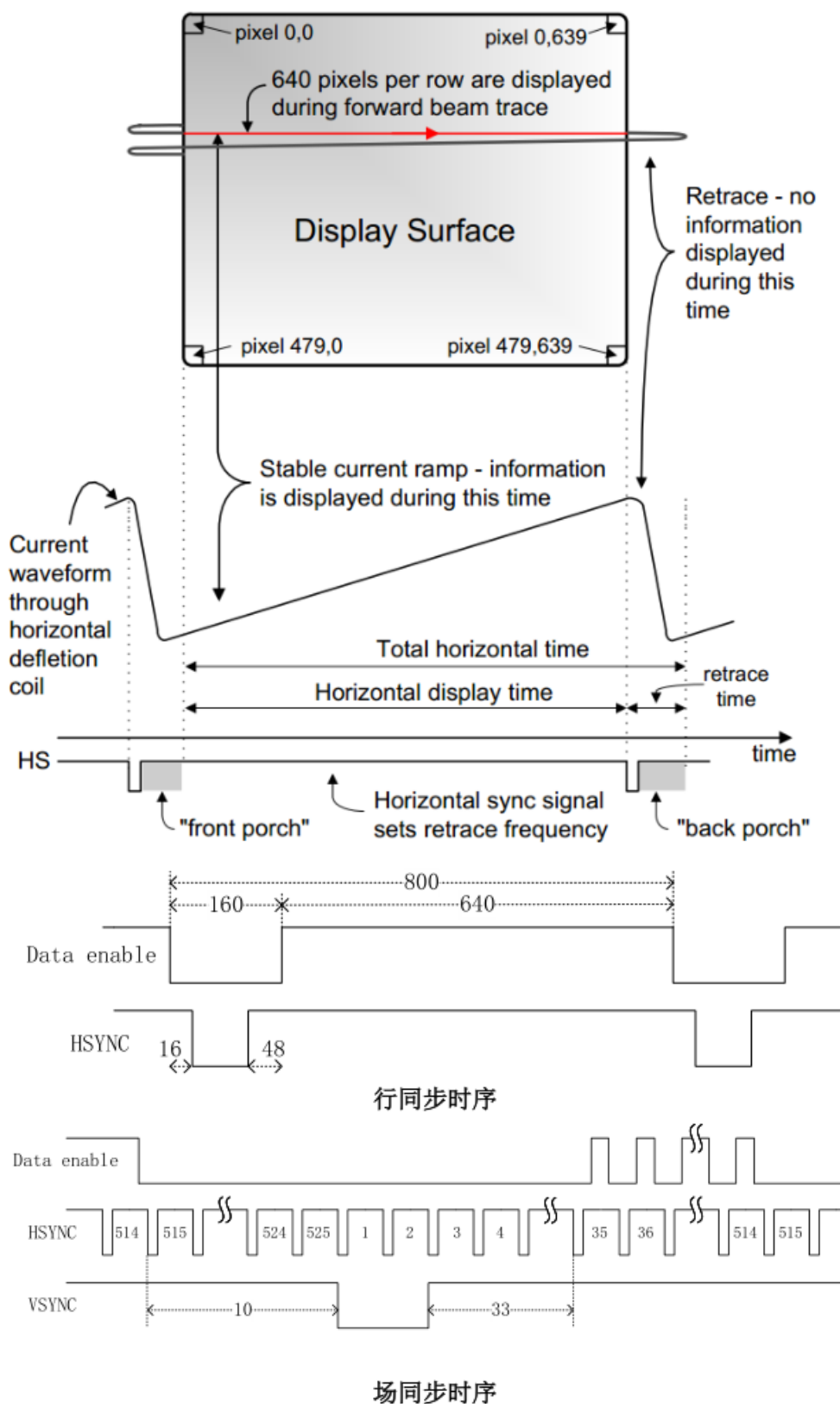
VGA 接口是显示器上应用最为广泛的接口类型，它是一种 D 型接口，上面共有 15 个针孔，信号连接如下图所示：



通用 VGA 显示卡系统主要由控制电路、显示缓存区和程序 BIOS 程序三个部分组成。控制电路主要完成时序发生、显示缓冲区数据操作、主时钟选择和 D/A 转换等功能；显示缓冲区提供显示数据缓存空间；视频 BIOS 作为控制程序固化在显示卡 ROM 中。本实验的系统架构如下图所示：



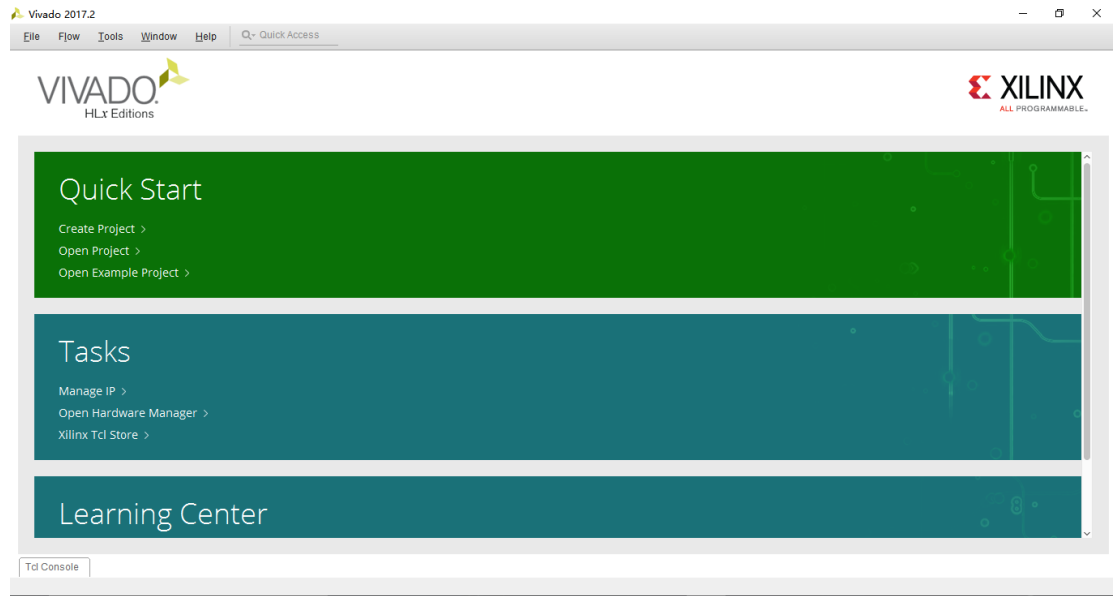
要实现 VGA 显示，主要还是实现 VGA 时序的控制，时序标准参见下图。行时序和帧时序都需要产生同步脉冲 sync、显示后沿 back porch、显示时序段 active 和显示前沿 front porch 四个部分。



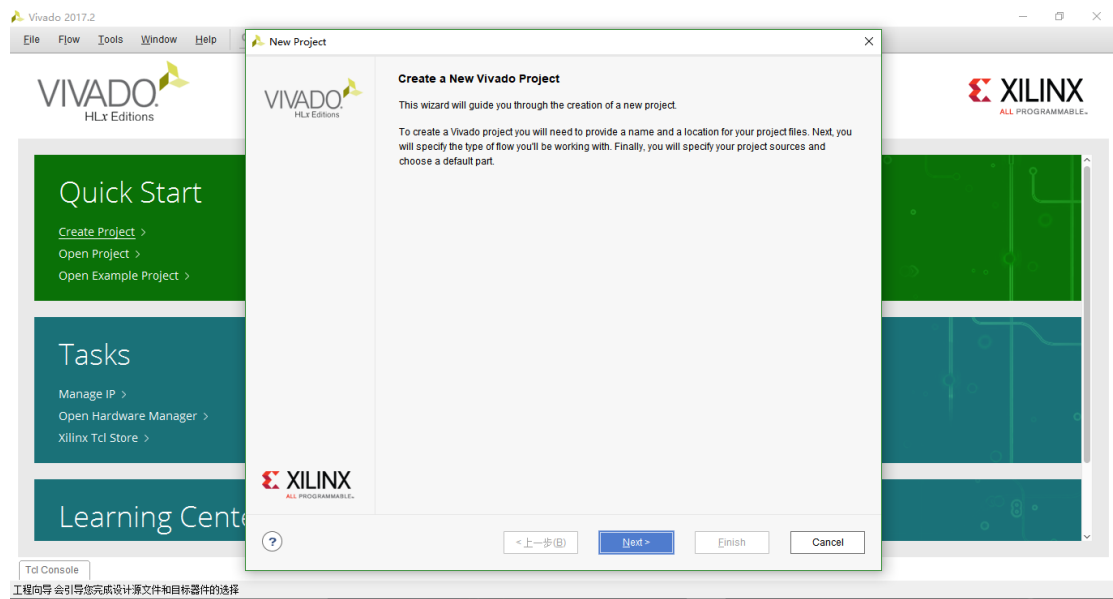
实验步骤:

一、创建工程

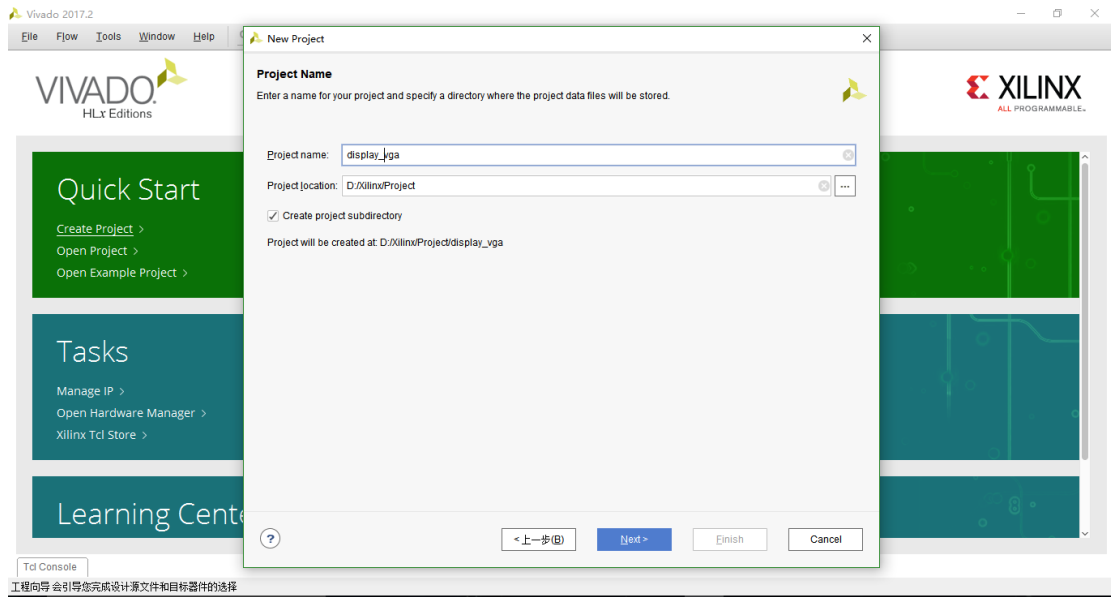
1、打开 Vivado2017.2，选择 “Create New Project”，创建一个新的工程。



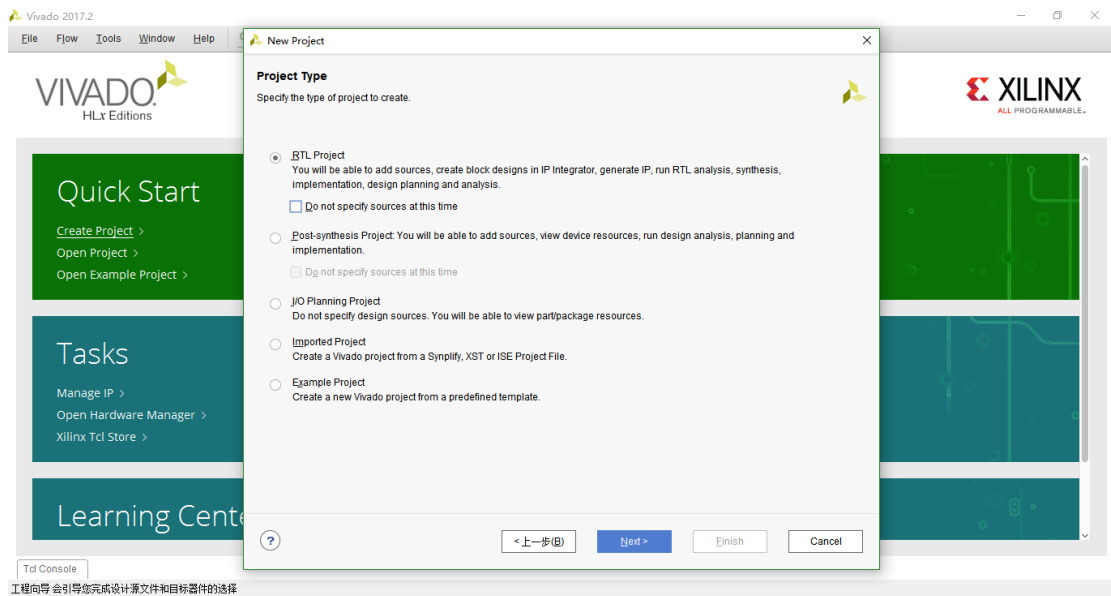
2、进入新建工程向导。



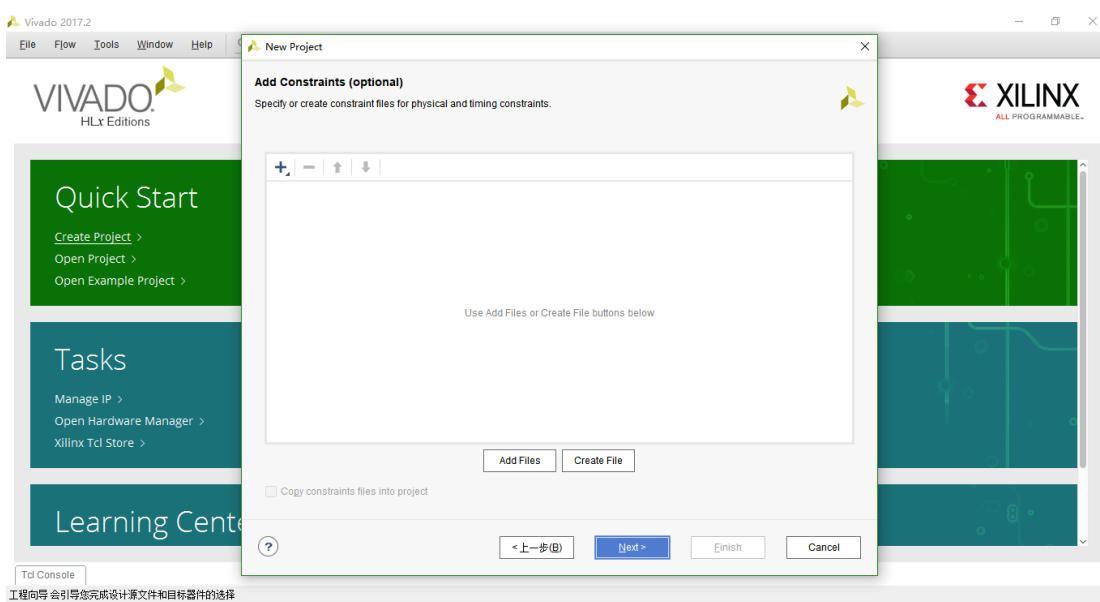
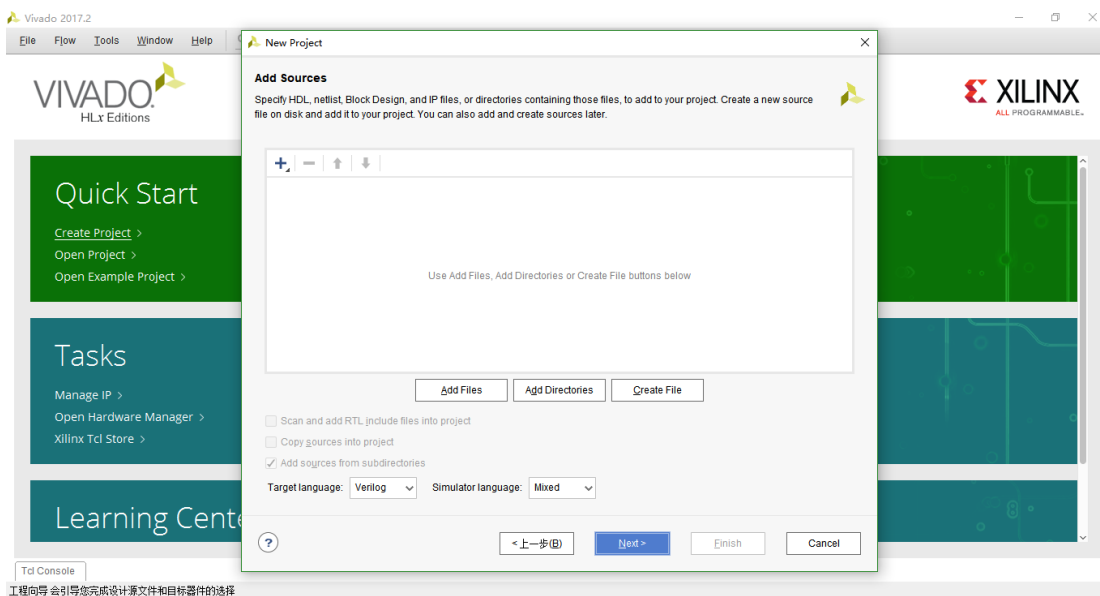
3、点击 “Next”，输入工程名并指定工程所在的目录，确认勾选 “Create project subdirectory”。注意工程名及工程所在的路径中只能包括数字、字母及下划线，不允许出现空格、汉字以及特殊字符等。这里我们将工程名命名为 `display_vga`，工程目录选择 `E:/lab`。



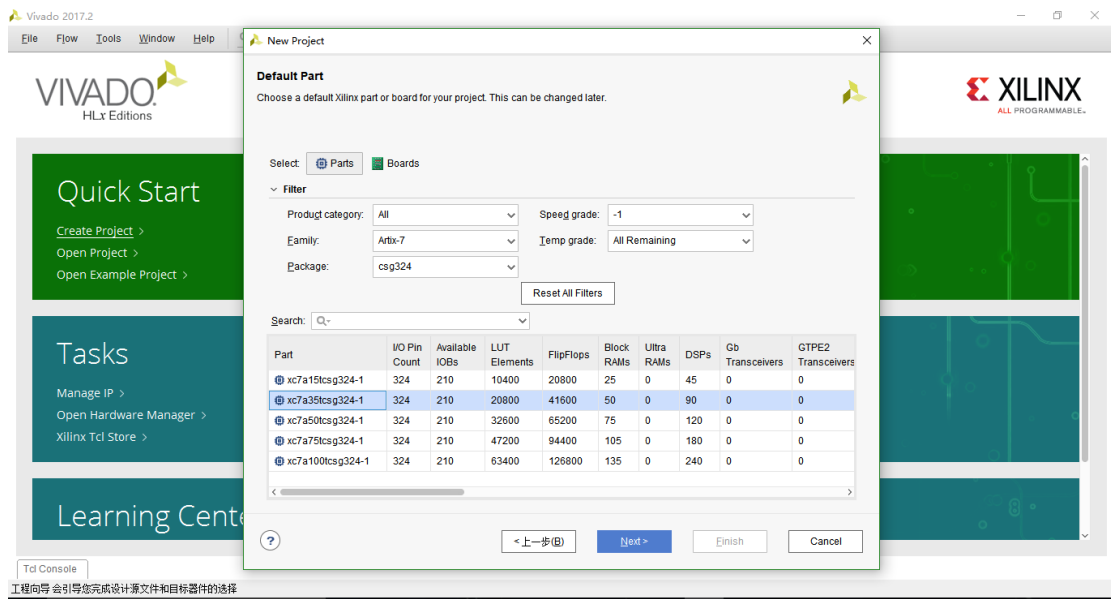
4、点击 “Next”，指定创建的工程类型，选择 “RTL Project”。



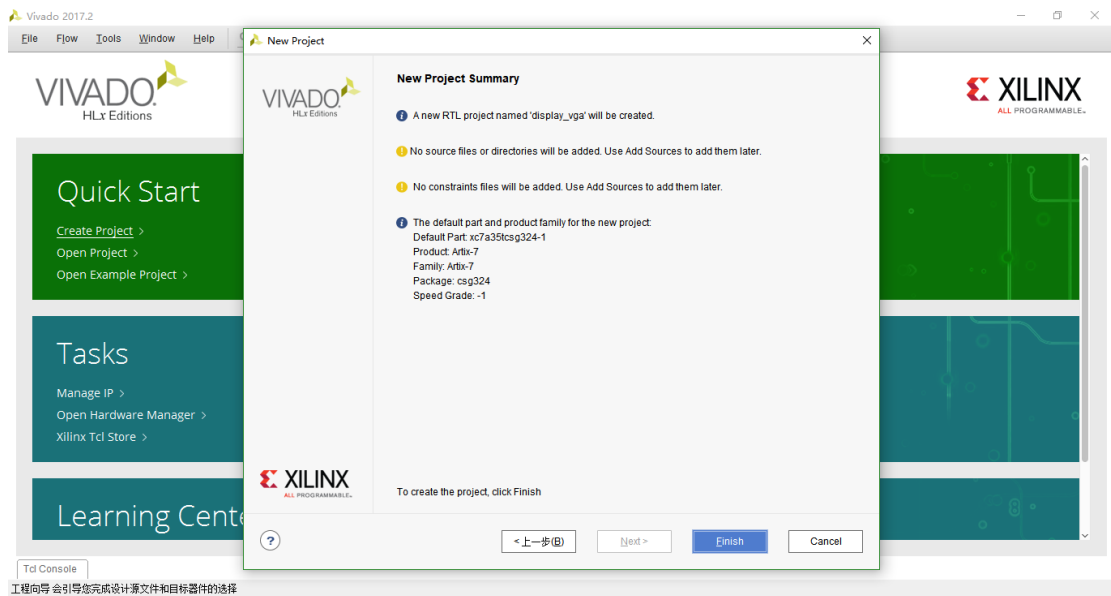
5、Target language 选择 “Verilog” 语言，Simulator language 选择 Mixed，接着连续点击 “Next”。



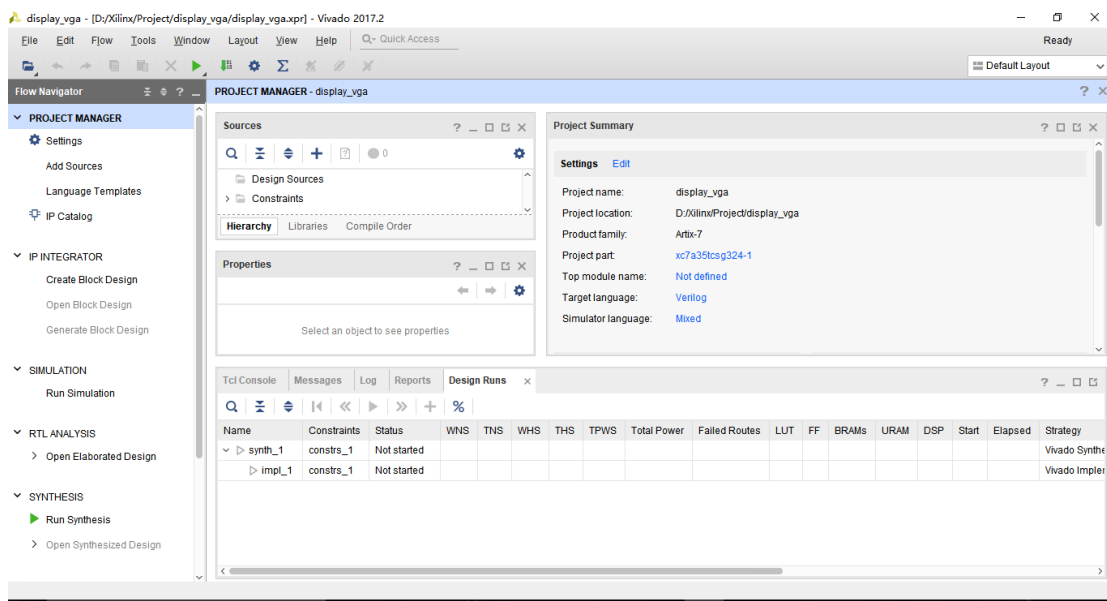
6、进入器件选择界面，通过下拉按钮选择器件的系列、封装形式，速度等级和温度等级，在符合条件的器件中选中板卡对应的芯片。Ego1 上的芯片型号为 XC7A35TCSG324-1。



7、点击 “Next”，进入 Summary 界面查看所创建工程的相关信息。

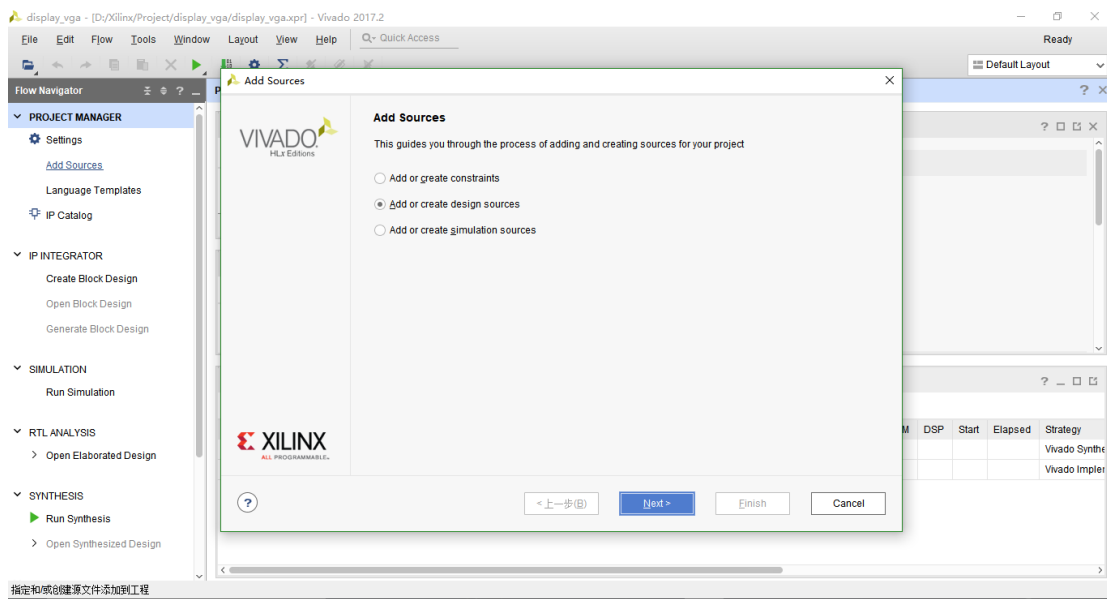


8、点击 “Finish”，打开创建的工程。

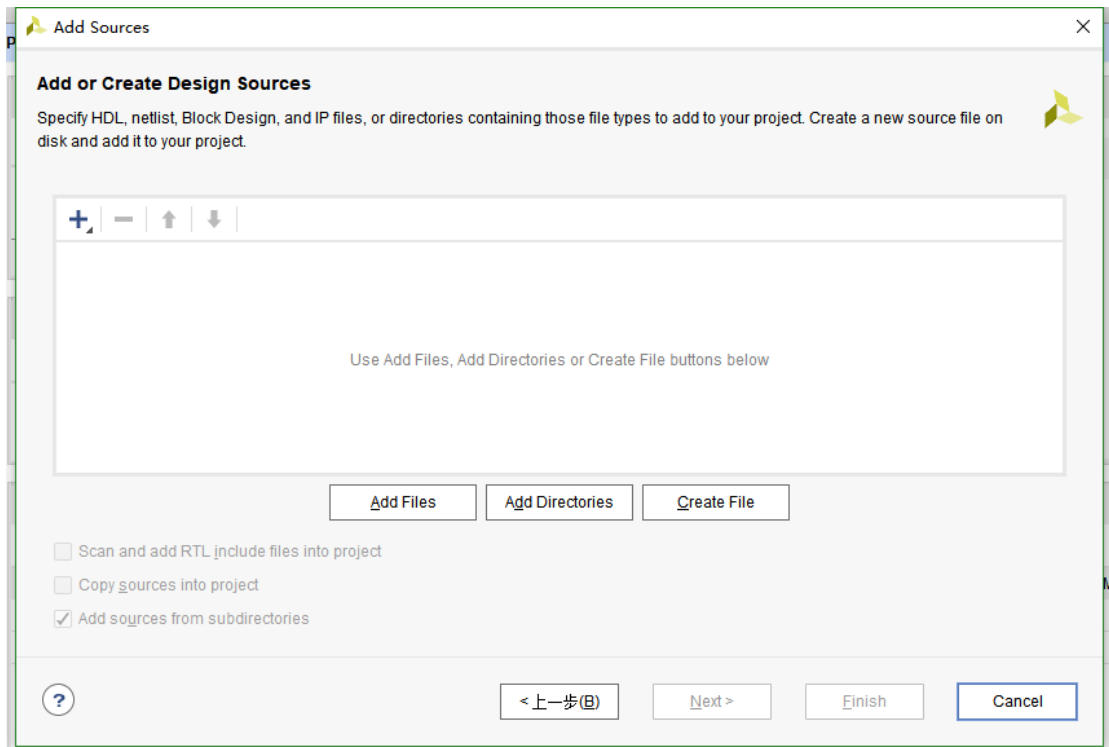


二、添加源文件

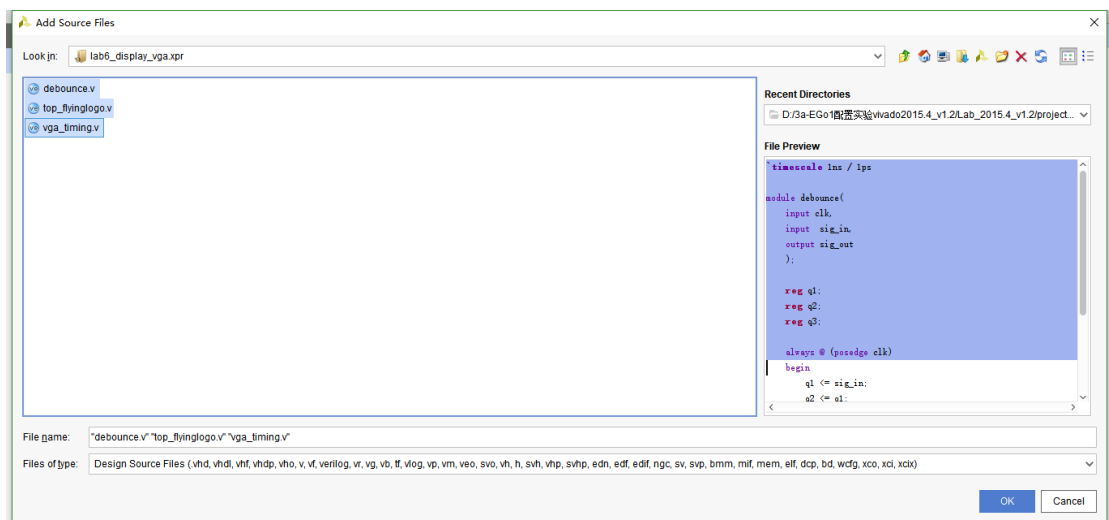
9、在左侧 “Flow Navigator” 栏中的 “Project Manager” 下点击 “Add Sources”，在弹出的窗口中选择 “Add or create design sources”，点击 “Next”。



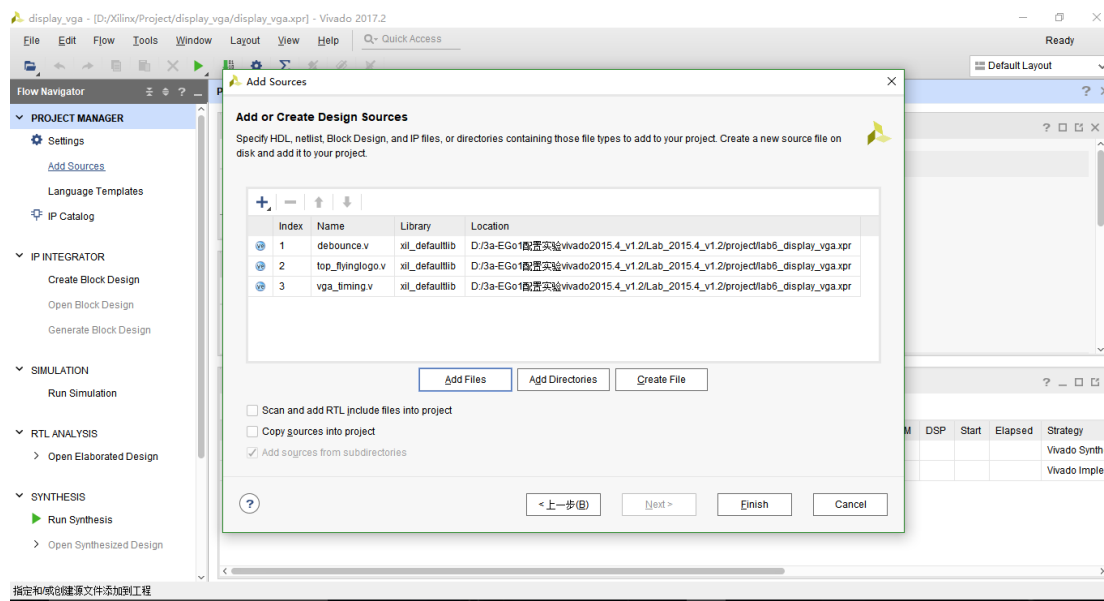
10、选择 “Add Files”。



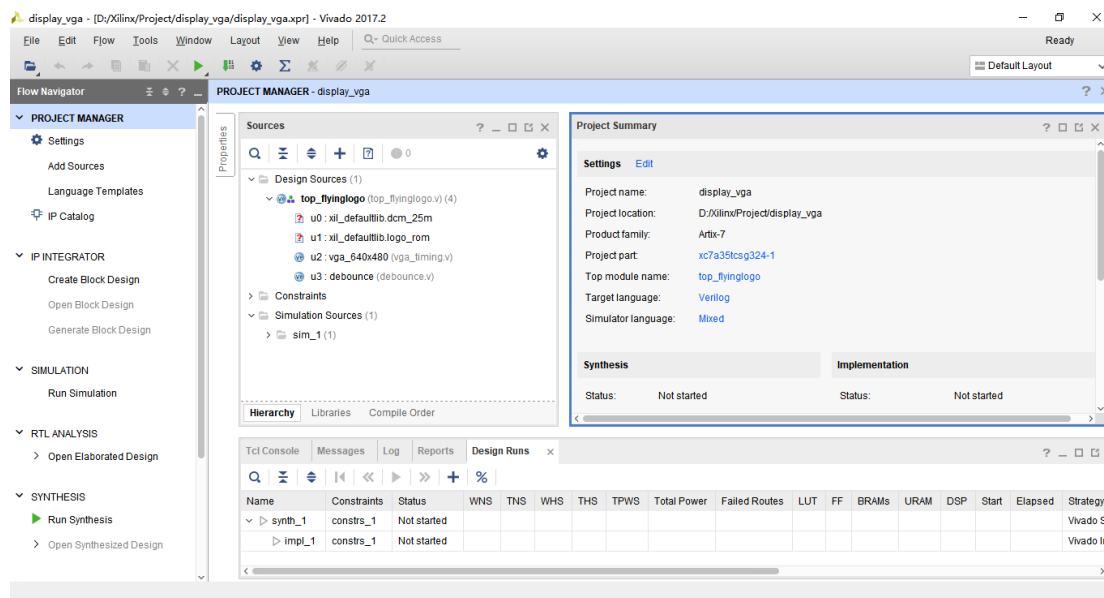
11、进入本实验源文件目录（D:/project/sources），选中各模块的源文件并点击 “OK”。



12、点击 “Finish”。



13、在 Sources 窗口可查看添加的源文件。

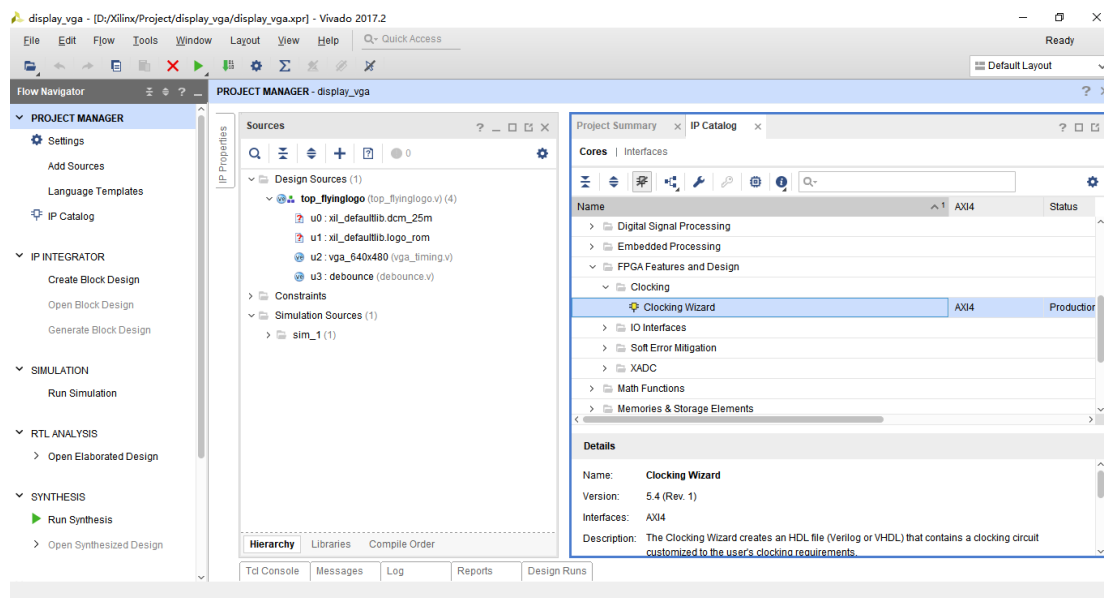


三、例化 IP 核

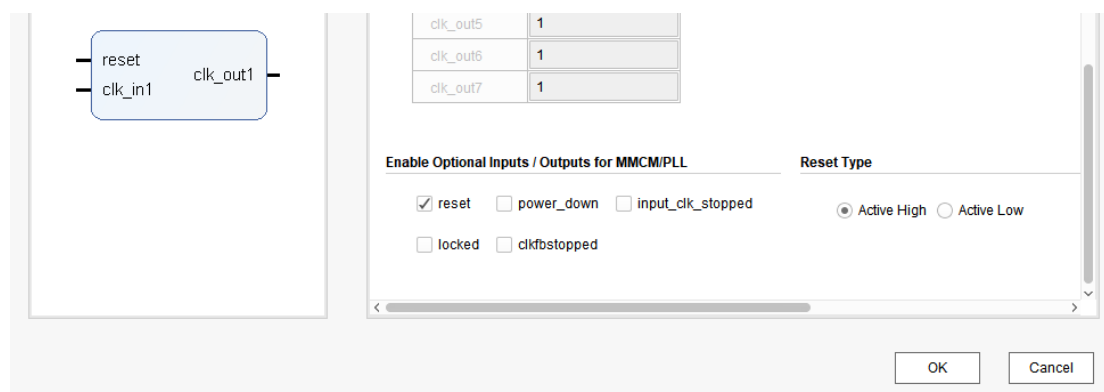
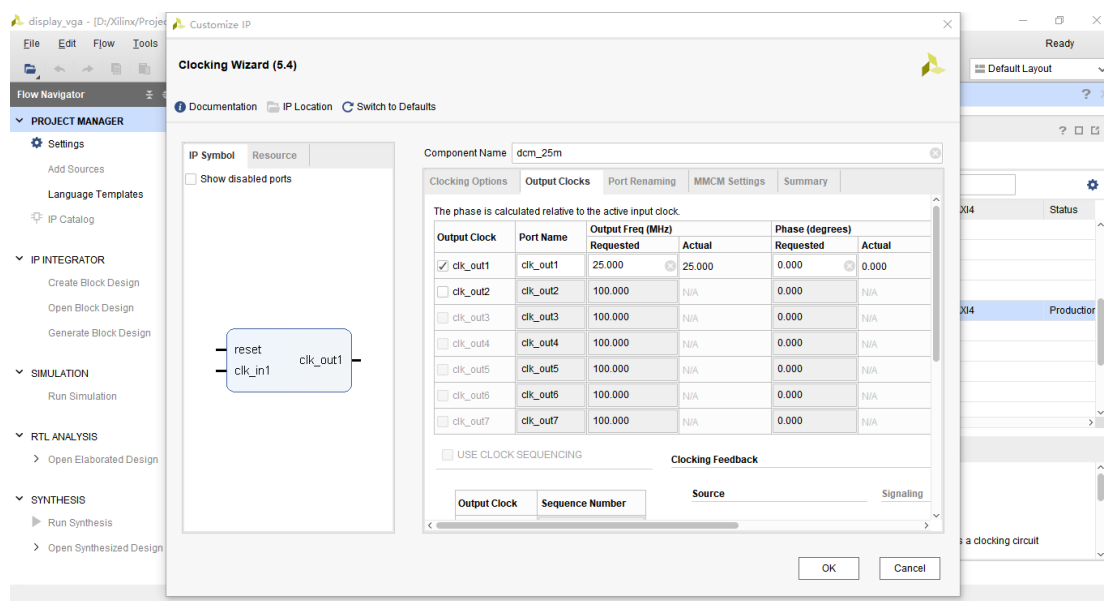
14、在上图可以看到有两个模块缺失，一个是时钟模块，另一个是 ROM 模块。我们将通过例化 IP 核来完成这两个模块的设计。

本实验中需要用 25MHz 的时钟驱动 VGA 控制器，接下来的步骤通过例化时钟 IP，将 EGo 上的 100MHZ 板载时钟分频得到 25MHz 时钟。

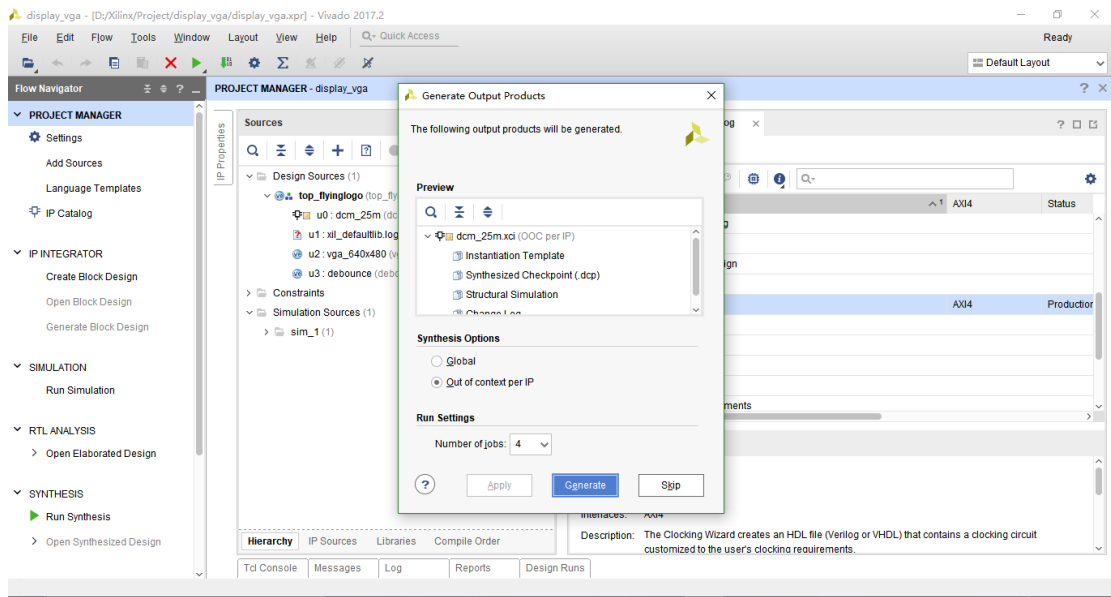
在左侧“Flow Navigator”栏中的“Project Manager”下点击“IP Catalog”。在界面右侧弹出的“IP Catalog”窗口中选择“FPGA Features and Design→Clocking→Clocking Wizard”。



15、双击打开 “Clocking Wizard”，在 “Customize IP” 窗口中将 Clocking Wizard 的 Component Name 改为 “dcm_25m”。同时在 “Output Clocks” 标签页下将 “clk_out1” 改为 25MHZ，然后下拉到界面的下方，取消 locked 选项。

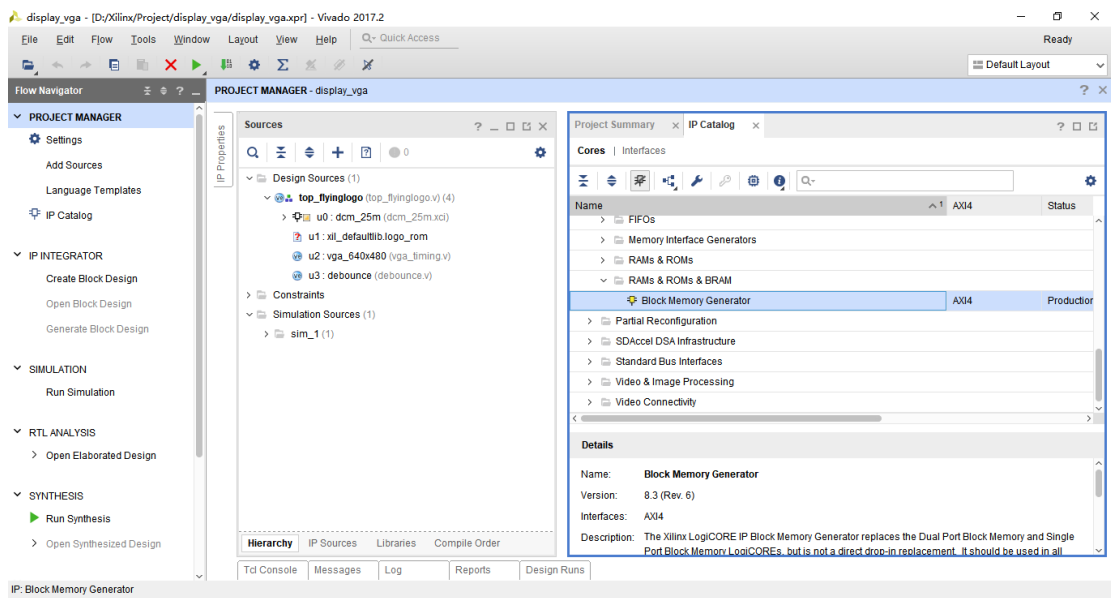


16、点击 OK，在弹出的窗口中选择 Out of context per IP，然后点击 “Generate”。该过程结束后在弹出的窗口中点击 OK 确认。

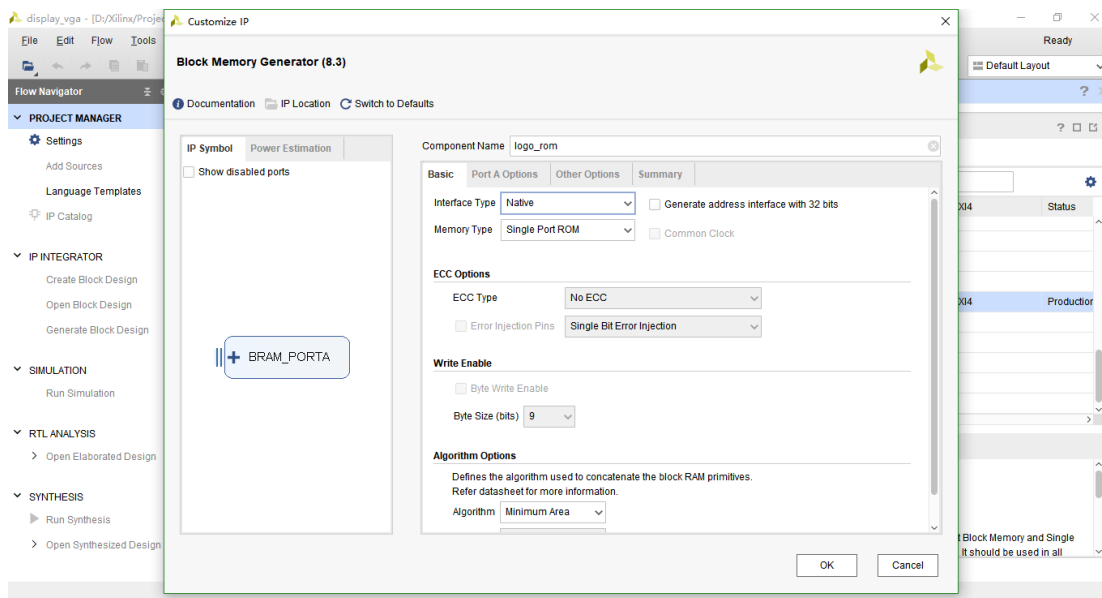


17、至此已经完成时钟 IP 核的例化，接下来我们通过例化 ROM 的 IP 核并对其进行初始化，将我们要显示的图片数据存放在 ROM 中。

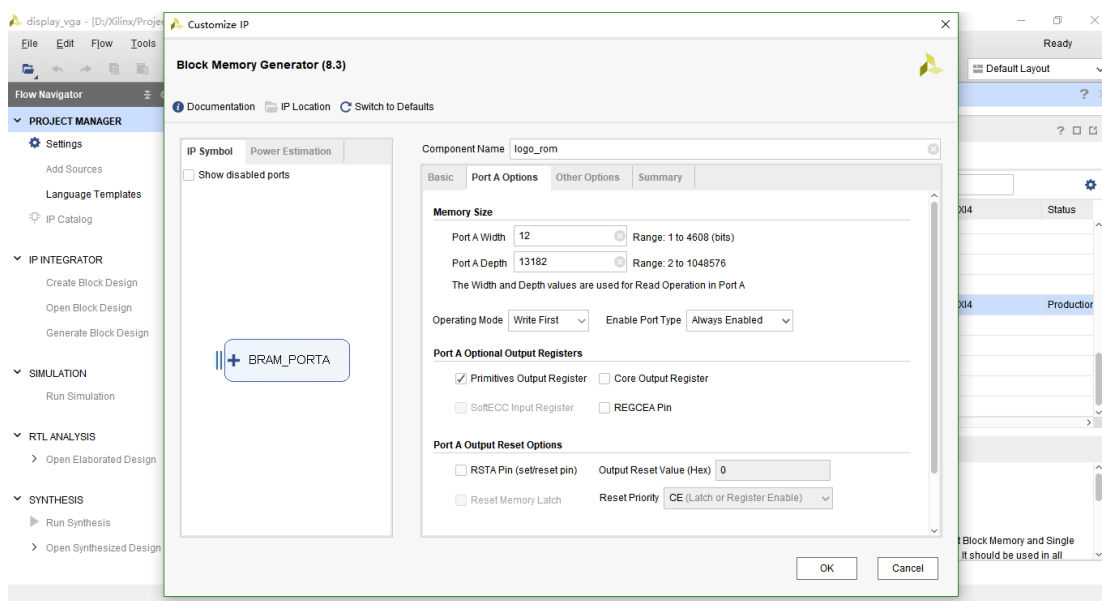
同样，在 “IP Catalog” 窗口中选择 “Memories & Storage Elements→RAMs & ROMS & BRAM→Block Memory Generator”。



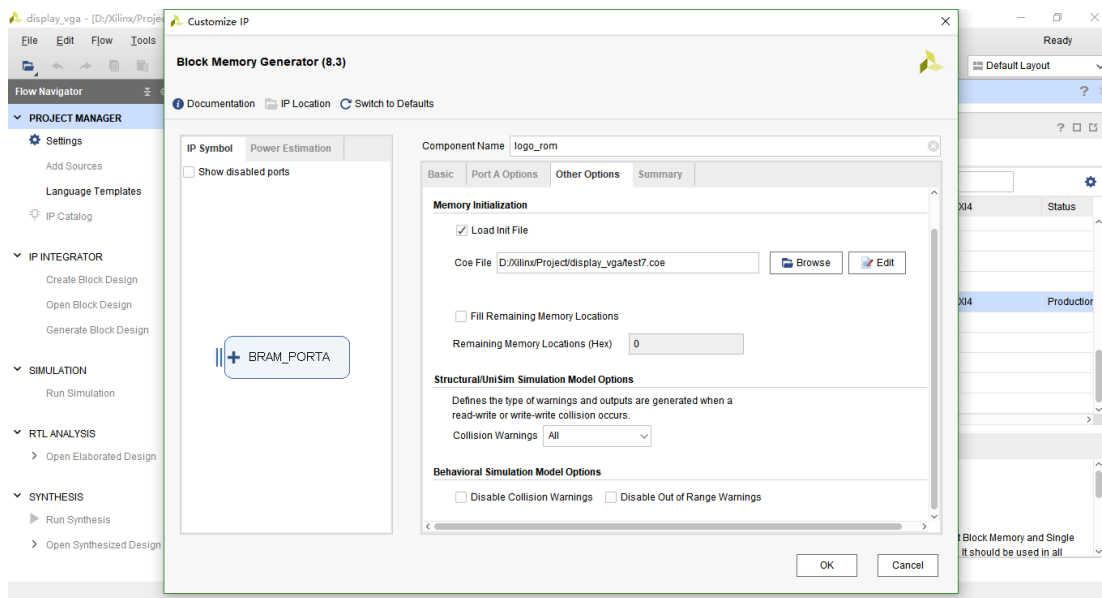
18、双击打开 Block Memory Generator，在 Customize IP 窗口中，将 Component Name 改为 logo_rom，在 Basic 标签页下，Memory Type 选择 Single Port ROM。



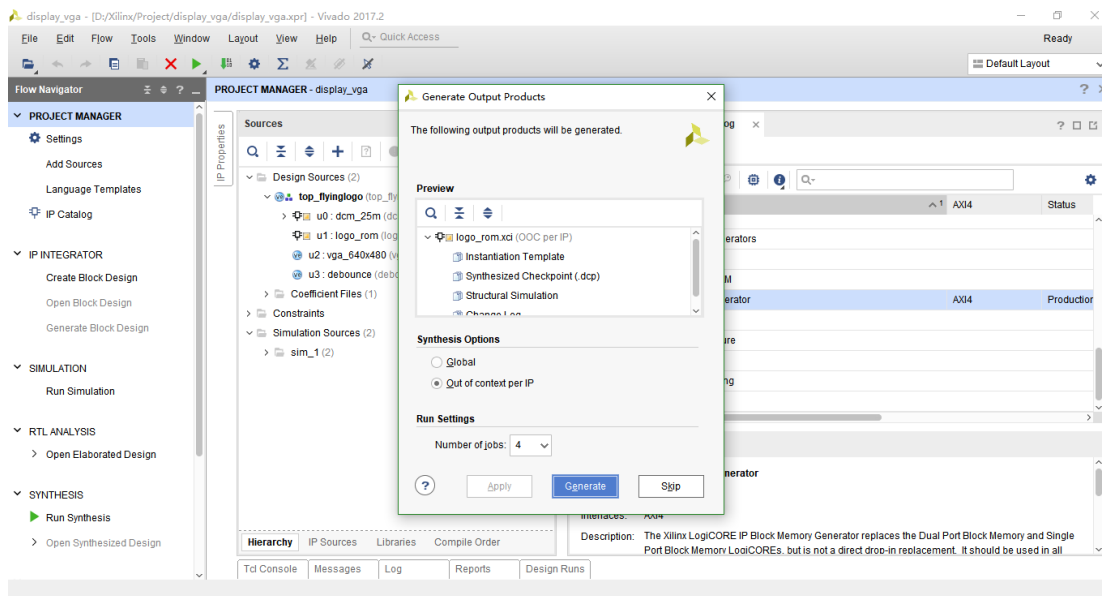
19、在 Port A Options 标签页下，将 Port A Width 设置为 12，Port A Depth 设置为 13182（图片为 169×78），Enable Port Type 设置为 Always Enable。



20、将 lab 文件夹中的 coe 文件夹拷贝到当前工程目录下（由图像生成 coe 文件的方法参见附录）。在 Other Options 标签页中勾选 Load Init File，然后通过点击右侧的文件夹图标，将目录指向刚刚拷贝的 coe 文件夹，双击其中的初始化文件 test7.coe，将其加载到 ROM 中。



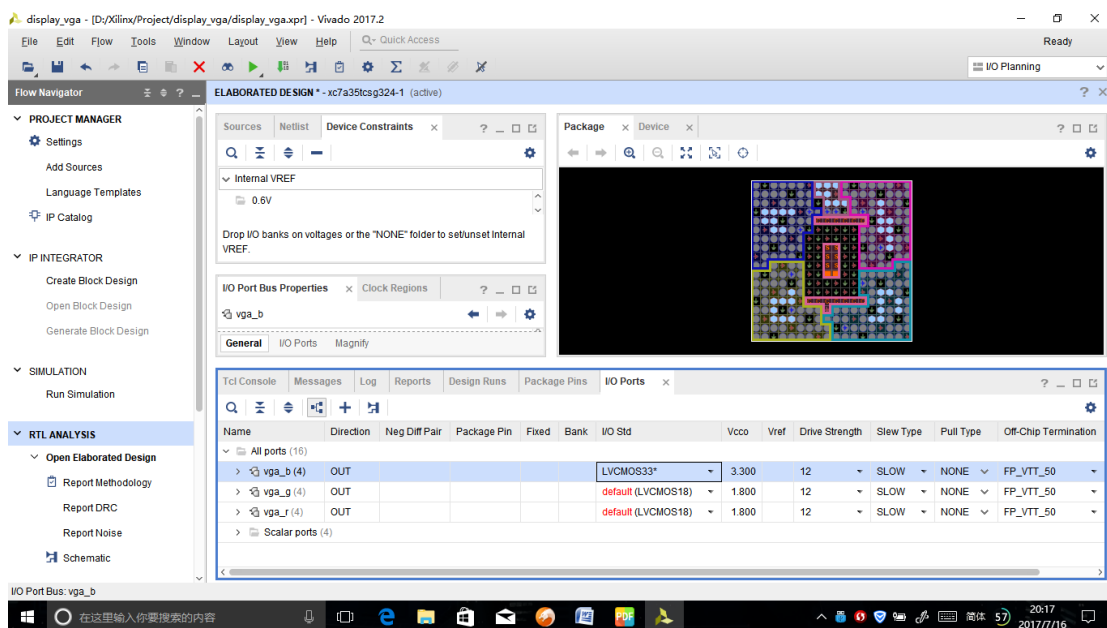
21、点击 OK，在弹出的窗口中选择 Out of context per IP，然后点击 “Generate”。



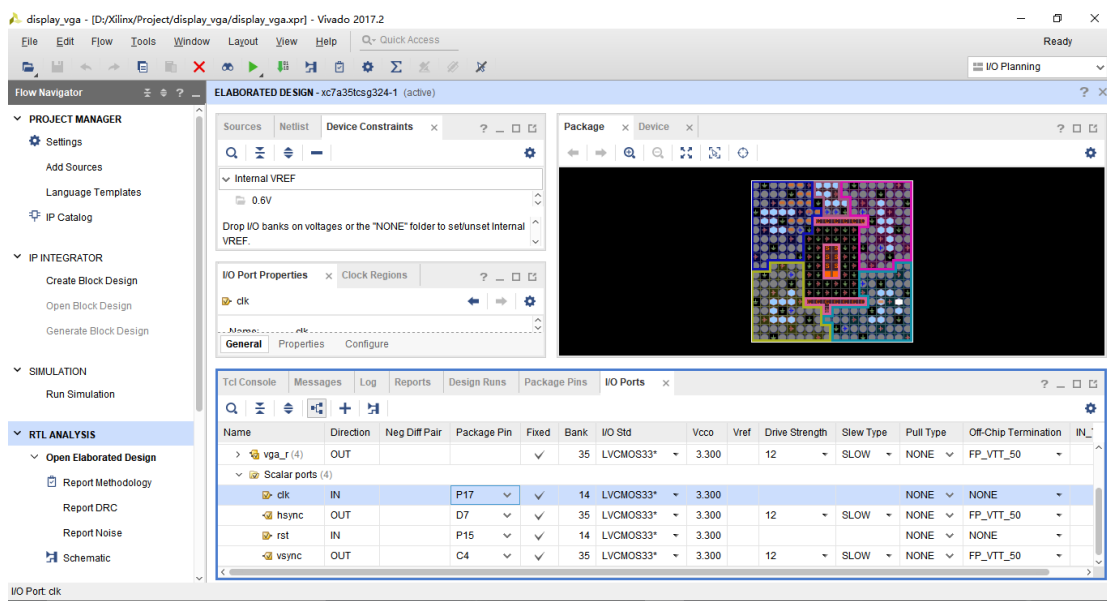
四、添加约束文件

22、在界面左侧 Flow Navigator 一栏中点击 RTL Analysis 下的 Open Elaborated Design，然后在菜单栏中选择 Layout→I/O Planning。

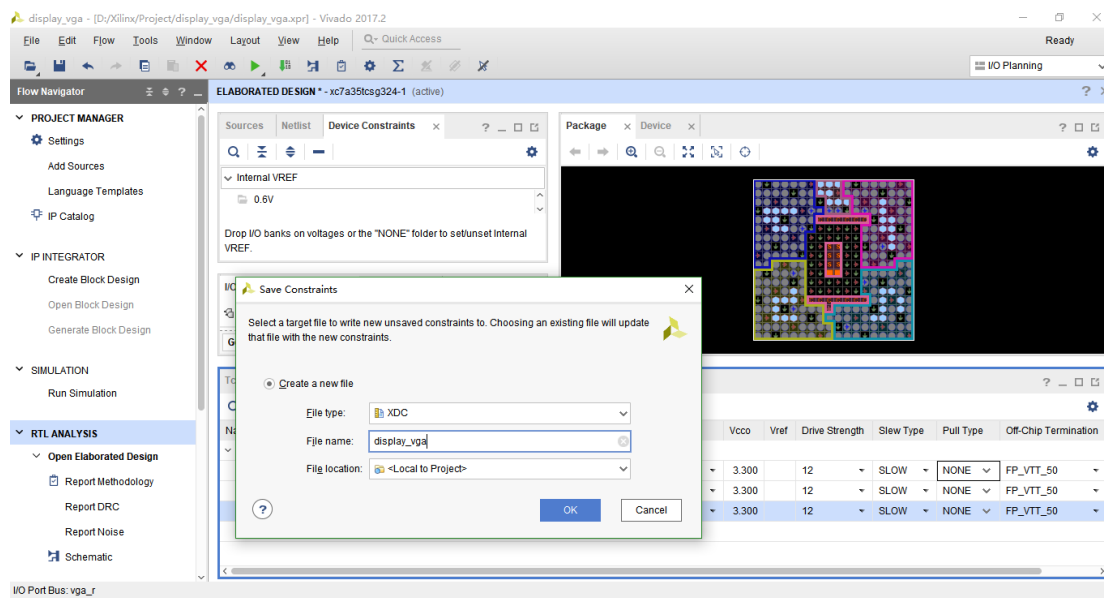
在 “I/O Ports” 窗口中对输入输出信号添加约束。首先在 “I/O Std” 一栏通过下拉按钮选择 “LVCOMS33”，将所有信号的电平标准设置 3.3V。



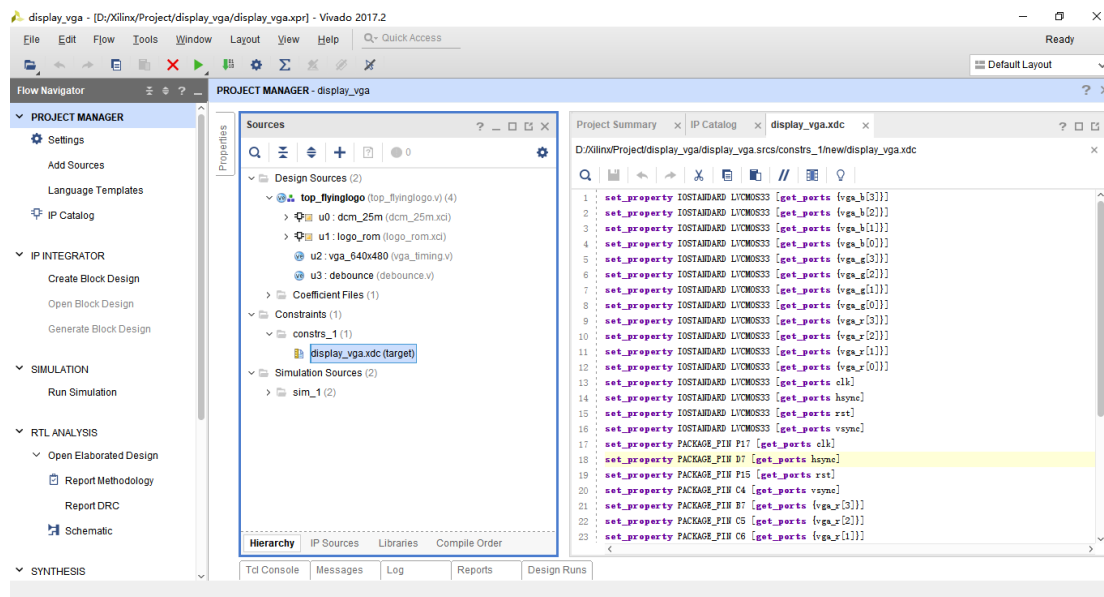
23、在 “Site” 一栏分配各个信号在 FPGA 芯片上引脚的位置，各信号的具体位置可查看板卡的原理图。



24、管脚分配完成后点击左上角的保存按钮，在弹出的对话框中 “File Name” 一栏输入约束文件的名称 “display_vga”，点击 “OK”。

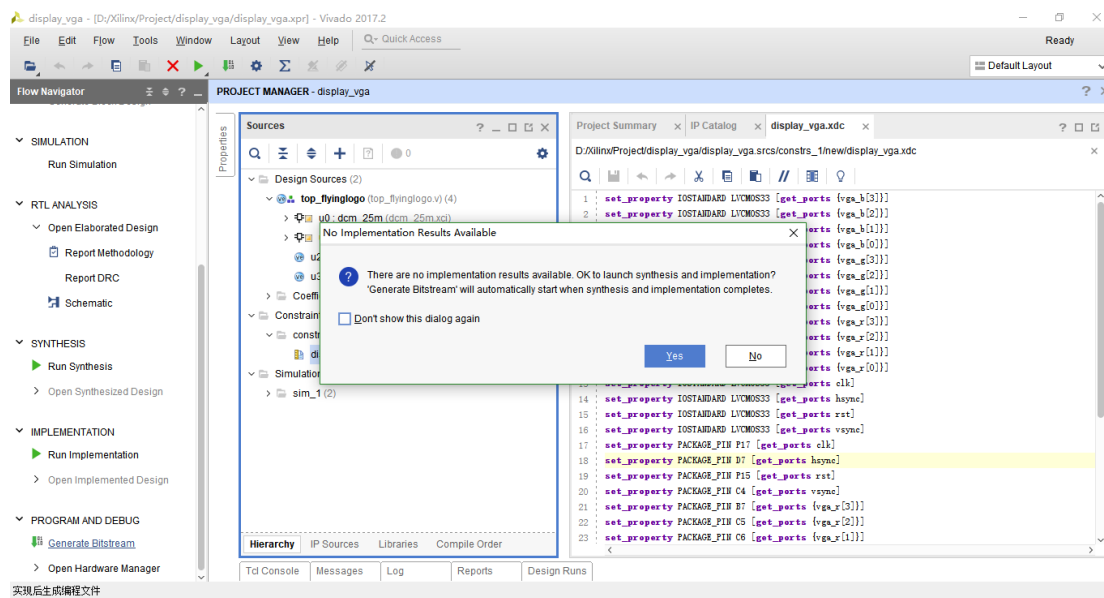


25、在右上角浅蓝色区域点击叉号，点击“OK”确认关闭“Elaborated Design”。在“Sources”窗口中的“Constraints/ constrs_1”下双击“display_vga.xdc”，打开刚刚创建的约束文件并查看内容。也可以通过直接添加约束文件的方式来对设计进行管脚分配。

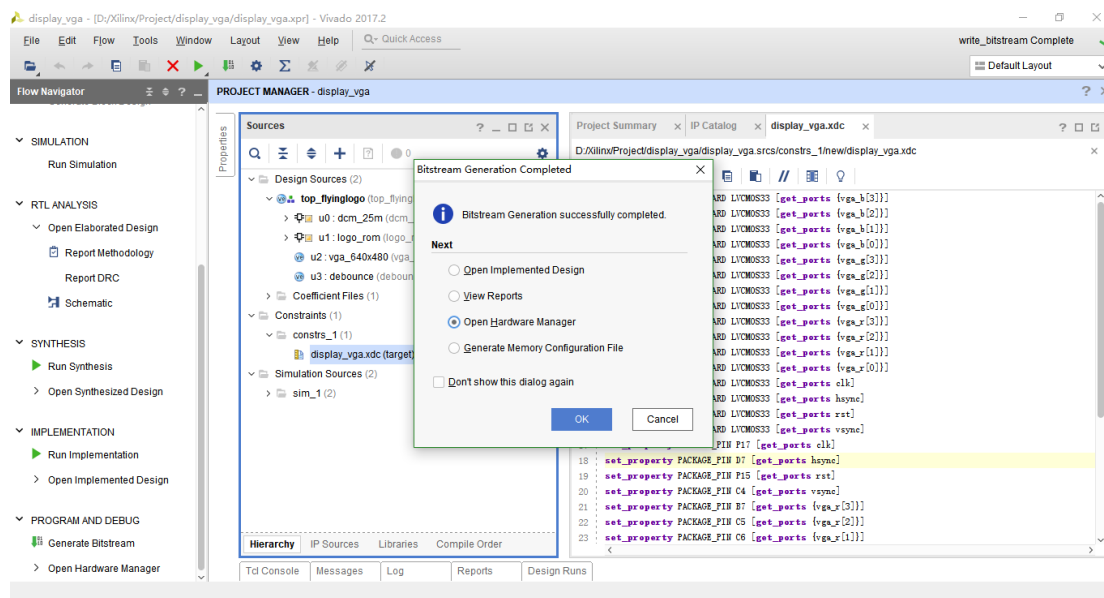


五、生成 bit 文件

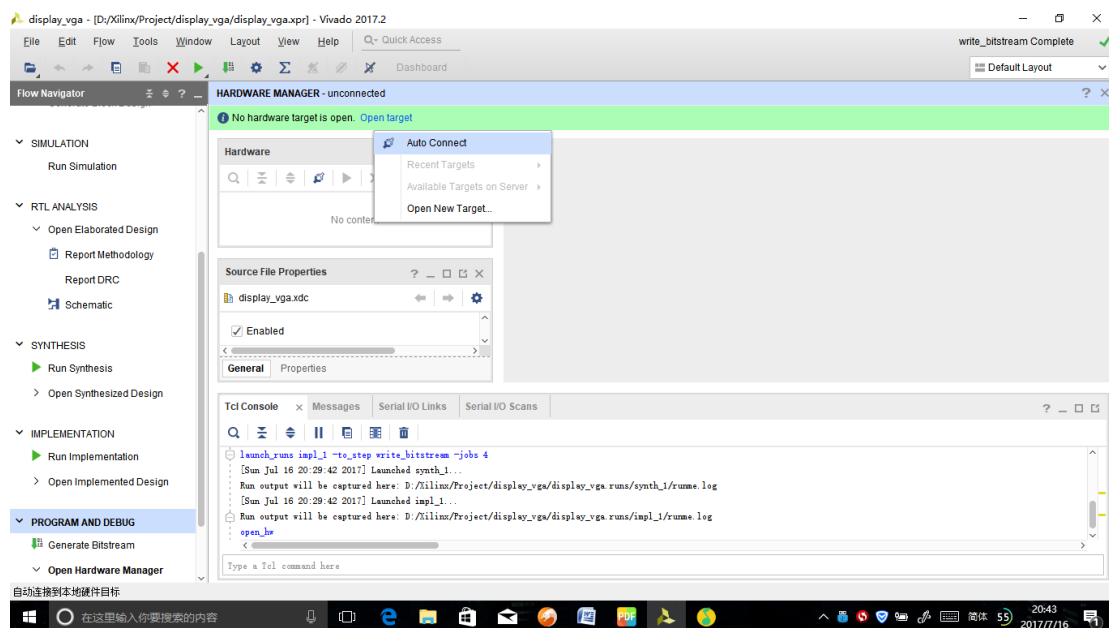
26、在“Flow Navigator”一栏中的“Program and Debug”下点击“Generate Bitstream”，此时会提示工程没有实现，点击“Yes”会自动执行综合及实现过程。



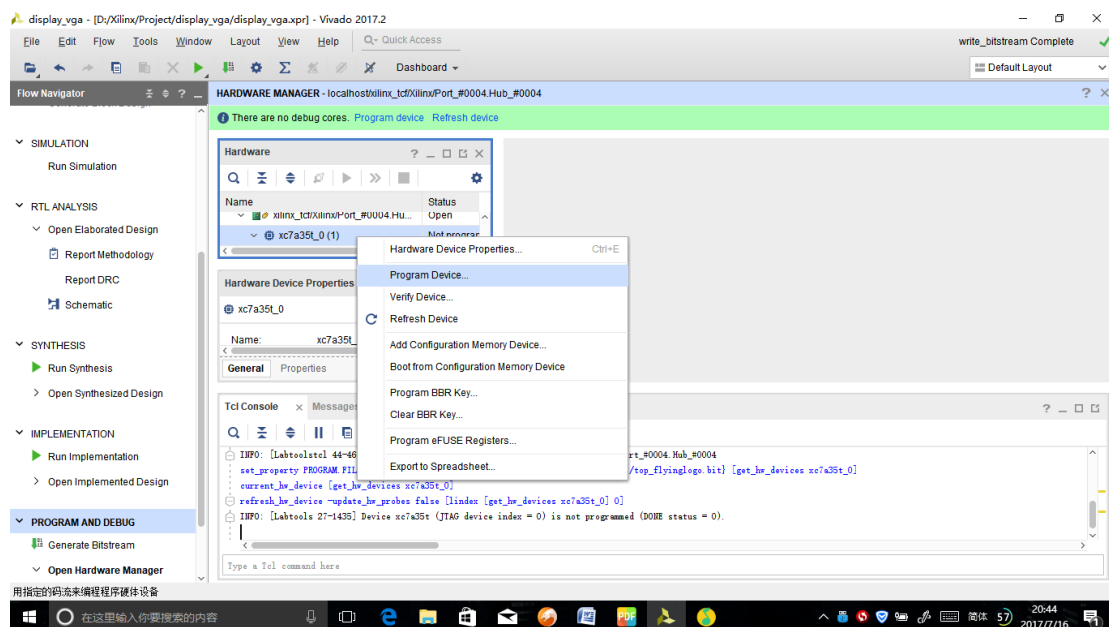
27、生成比特流文件完成后，选择 “Open Hardware Manager” 并点击 “OK”。



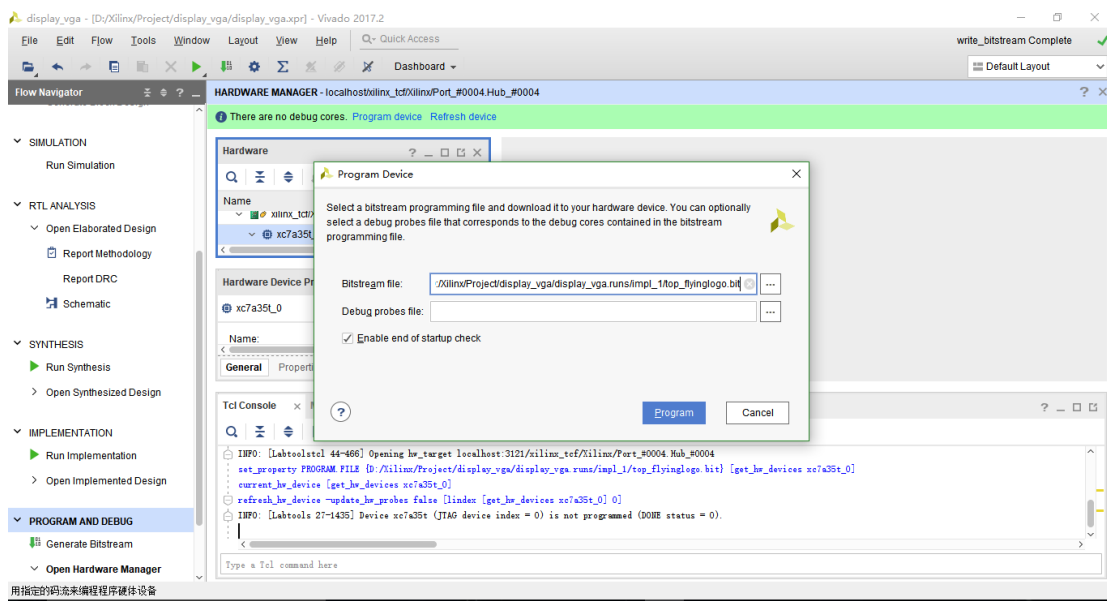
28、将显示器与板卡上 VGA 接口连接，然后用 Micro USB 线连接电脑与板卡上的 JTAG 端口，并打开电源开关。在 “Hardware Manager” 界面点击 “Open target”，选择 “Auto Connect”。



29、连接成功后，在目标芯片上右击，选择 “Program Device”。



30、在弹出的对话框中 “Bitstream File” 一栏中已经自动加载本工程生成的比特流文件，点击 “Program” 对 FPGA 芯片进行编程。



31、比特流文件下载完成后，在 VGA 显示器上验证实验结果。

