

SWORD

XD-LAB-IMG-001

Lab1: 基本视频 IO 实验: HDMI In & Out Passthrough

Joseph Xu

2019-2-14

修改记录

版本号.	作者	描述	修改日期
1.0	Joseph Xu	初稿	2018-4-1
1.2	Joseph Xu	文档图片微调	2019-2-14

审核记录

姓名	职务	签字	日期

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	1 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

目录

修改记录.....	1
审核记录.....	1
1. 实验简介.....	6
1.1 概述.....	6
1.2 实验目标.....	7
1.3 实验条件.....	7
1.4 实验原理.....	8
2. 实验流程.....	12
2.1 操作步骤.....	12
3. 实验结果.....	40
3.1 现象.....	40

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	2 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

图目录

图 1-1	SWORD4.0 的 HDMI 接口位置 (上为输出, 下为输入)	6
图 1-2	HDMI 视频线	7
图 1-3	实验连接示意图	8
图 1-4	HDMI 输入输出回环连接框图	8
图 1-5	DVI2RGB IP	9
图 1-6	RGB2DVI IP	10
图 2-1	Vivado 下创建新工程	12
图 2-2	创建新工程向导窗口	12
图 2-3	工程命名和保存路径	13
图 2-4	选择工程类型	13
图 2-5	选择器件型号	14
图 2-6	点击 Finish 完成工程创建	14
图 2-7	设置: 添加 IP 库	15
图 2-8	IP 库文件存放位置	15
图 2-9	选择 IP 库的位置	16
图 2-10	IP 库中的 IP 列表显示	16
图 2-11	创建模块化设计	17
图 2-12	添加 DVI2RGB IP	17
图 2-13	添加 RGB2DVI IP	18
图 2-14	添加 IP 后的视图	18
图 2-15	配置 DVI2RGB IP	19
图 2-16	DVI2RGB IP 参数设置	20
图 2-17	RGB2DVI IP 参数设置	21
图 2-18	选择 DVI2RGB IP 的 TMDS 端口	21
图 2-19	将端口引出到外部	22
图 2-20	引出端口	22
图 2-21	将 DDC 端口引出到外部	23
图 2-22	端口引出	23
图 2-23	端口如何重命名	24
图 2-24	端口重命名后的视图	24
图 2-25	连接 RGB 端口	25
图 2-26	调整框图布局	25

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	3 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

图 2-27	添加时钟向导 IP	26
图 2-28	添加时钟向导 IP 后的视图	26
图 2-29	点击 IP 的配置	27
图 2-30	时钟向导设置——Clocking Options	27
图 2-31	时钟向导设置——Output Clocks	28
图 2-32	各 IP 设置好后的视图	28
图 2-33	将时钟向导的 2 个输入端口引出到外部	29
图 2-34	各 IP 之间的连接图	29
图 2-35	重新调整布局后的视图	30
图 2-36	保存模块设计	30
图 2-37	创建顶层文件	31
图 2-38	自动更新顶层文件	31
图 2-39	查看自动生成的顶层文件	32
图 2-40	Open Elaborated Design	32
图 2-41	进行管脚映射	33
图 2-42	IO 管脚设置	34
图 2-43	保存文件	34
图 2-44	将文件命名为 lab1	34
图 2-45	打开 lab1.xdc 文件	35
图 2-46	修改 lab1.xdc 文件	35
图 2-47	Generate Bitstream	36
图 2-48	点击 Yes 确认生成 bit 文件	36
图 2-49	打开 Hardware Manager	37
图 2-50	硬件连接对应位置	37
图 2-51	实际硬件连接	38
图 2-52	Open target	38
图 2-53	Program Device	39
图 2-54	烧写目标器件	39
图 2-55	编程进度条	39
图 3-1	正常显示结果	40
图 3-2	显示器的显示分辨率	40

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	4 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

表目录

表 1-1	实验条件.....	7
表 1-2	DVI2RGB IP 端口列表.....	9
表 1-3	RGB2DVI IP 端口列表.....	10
表 2-1	端口设置表.....	33

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	5 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

1. 实验简介

该实验利用 SWORD4.0 的 HDMI 视频输入输出接口，将外部接收到的 HDMI 视频信号转换为像素数据并重新输出给 HDMI 显示器，该实验为后面的处理模块设计做准备。

- **对于初学者，整个实验预计耗时 1 小时。**
- **对于进阶者，整个实验预计耗时 10 分钟。**

1.1 概述

HDMI 英文全称 High Definition Multimedia Interface，是一种高清晰度多媒体接口，常用于视频领域，在笔记本电脑，计算机显卡，电脑显示器，平板电视等视频信号源设备或视频终端上应用非常普遍。SWORD4.0 上具有一路 HDMI 输入接口和一路 HDMI 输出接口，其中 HDMI 输入信号支持的最高分辨率为 1600x900P@60Hz，HDMI 输出信号的最高分辨率为 1920x1080P@60Hz。

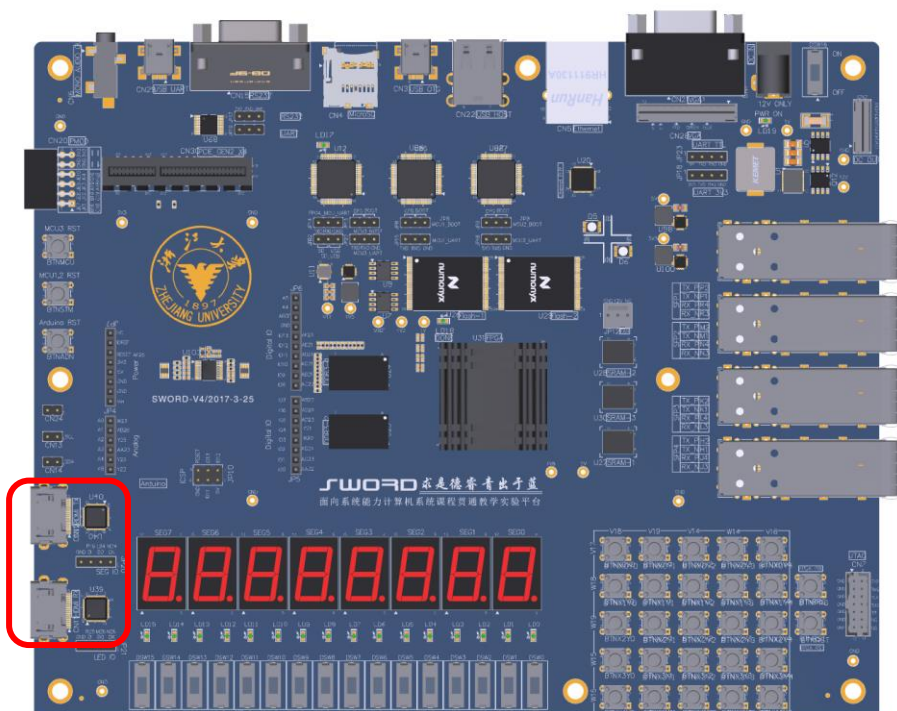


图 1-1 SWORD4.0 的 HDMI 接口位置（上为输出，下为输入）

xingdeng	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	6 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		



图 1-2 HDMI 视频线

1.2 实验目标

本实验的目标为 SWORD4.0 能够正常地在 HDMI 显示器上通过 SWORD4.0 的 HDMI 输入接收的视频画面，其中包含如下几种分辨率*：

1. 640x480@60Hz——VGA；
2. 800x600@60Hz——SVGA；
3. 1280x720@60Hz——HD；
4. 1600x900@60Hz——HD+；

*注：根据 https://en.wikipedia.org/wiki/Graphics_display_resolution

1.3 实验条件

表 1-1 实验条件

类别	名称	数量	说明
硬件	SWORD4.0	1	
	HDMI 信号源	1	如笔记本 HDMI 输出/台式计算机 HDMI 输出/带 HDMI 输出的视频机顶盒
	带 HDMI 接口的显示器	1	
	HDMI 视频线	2	
软件	Vivado Design Suite	1	版本：2014.4
	视频接口 IP 库	1	repo.zip

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	7 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

1.4 实验原理

该实验的连接方式如下图所示：

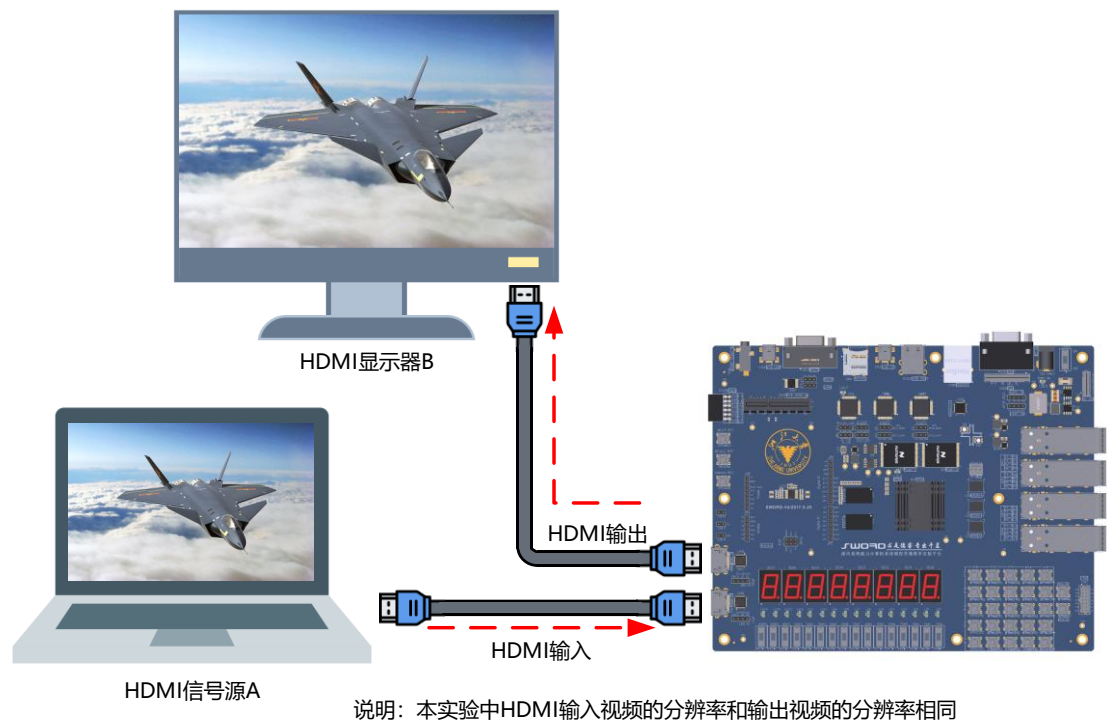


图 1-3 实验连接示意图

实验利用了 2 个 IP 来实现 HDMI 输入和 HDMI 输出的回环显示：DVI2RGB, RGB2DVI。
IP 连接框图如下图所示：

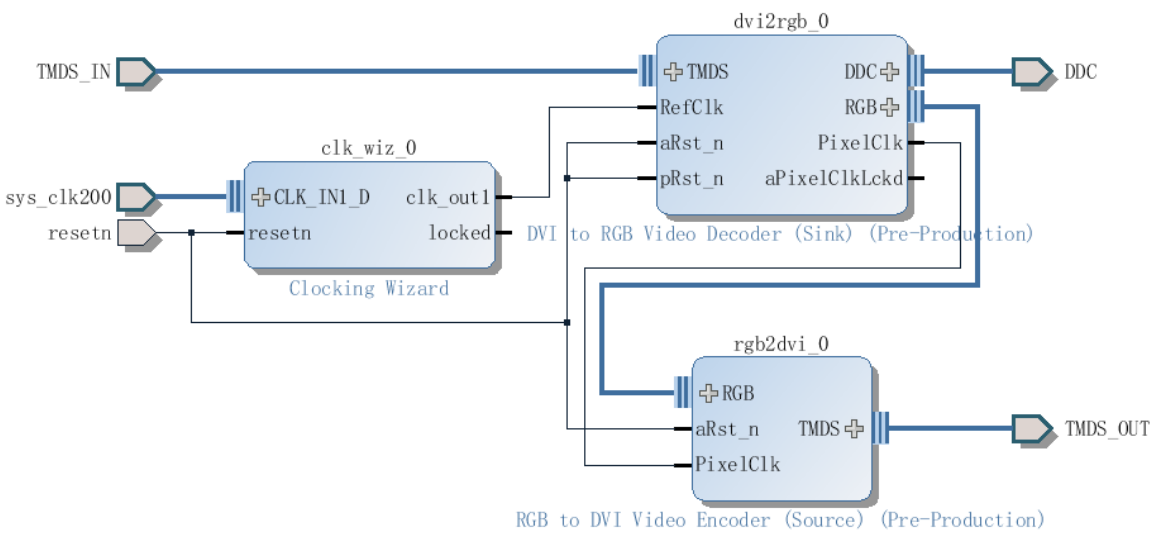


图 1-4 HDMI 输入输出回环连接框图

由于 DVI 和 HDMI 标准中的视频信号都属于 TMDS 信号，这是一种时钟速率较高的串行差分信号；而视频图像的最小基本单位为像素（Pixel）。

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	8 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

因此 DVI2RGB 这个 IP 的作用就是将经过特殊编码的串行型号转换成像素时钟域下的像素数据信号，其中单个像素用一个 24 位宽的二进制数据表示，同时还从中恢复出数据有效信号（DE）和同步信号（水平同步 Hsync，垂直同步 Vsync）。

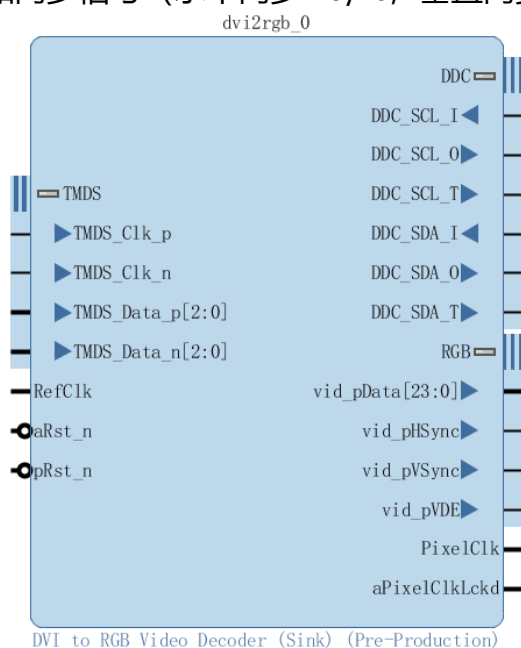


图 1-5 DVI2RGB IP

该 IP 的端口信号定义如下表所示：

表 1-2 DVI2RGB IP 端口列表

信号名	方向	宽度	含义
RefClk	I	1	200 MHz reference clock.
aRst_n	I	1	Asynchronous reset of configurable polarity. Assert, if RefClk is not within spec.
pRst_n	I	1	Active-high reset synchronous with PixelClk. Configurable polarity.
TMDS_Clk_p	I	1	TMDS Clock Channel.
TMDS_Clk_n	I	1	TMDS Clock Channel.
TMDS_Data_p	I	3	TMDS Data Channel 0.
TMDS_Data_n	I	3	TMDS Data Channel 0.
DDC_SCL	O	1	Display Data Channel clock signal. Optional.
DDC_SDA	IO	1	Display Data Channel data signal. Optional.
PixelClk	O	1	Pixel clock recovered from the TMDS clock channel. Driven by BUFR.
pVDE	O	1	Video data valid:

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	9 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

信号名	方向	宽度	含义
			• 1 = Active video. • 0 = Blanking period.
pHSync	O	1	Horizontal synchronization video timing signal.
pVSync	O	1	Vertical synchronization video timing signal.
pData	O	24	Video pixel data packed as RGB.
aPixelClkLckd	O		Active-high asynchronous locked signal for PixelClk. When low, PixelClk is lost or not within specs.
SerialClk	O		Fast clock, toggling at five times the frequency of PixelClk. It is used internally for deserialization. Provided optionally as output for advanced use, like clocking an RGB2DVI core. Driven by BUFIO.

而 RGB2DVI 这个 IP 的作用是将像素数据按照特殊编码的方式转换成 TMDS 信号。

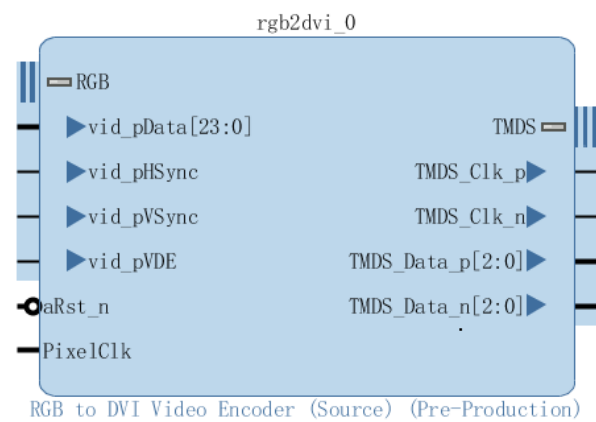


图 1-6 RGB2DVI IP

该 IP 的端口信号定义如下表所示：

表 1-3 RGB2DVI IP 端口列表

信号名	方向	宽度	含义
SerialClk	I	1	Optional fast serial clock. Has five times the frequency of PixelClk and are phase-aligned. Can be generated internally.
aRst(_n)	I	1	Asynchronous reset of configurable polarity. Assert, if PixelClk and SerialClk are not within spec.

信号名	方向	宽度	含义
Clk_p	O	1	DVI Clock Channel.
Clk_n	O	1	
Data_p	O	3	DVI Data Channel 0.
Data_n	O	3	
PixelClk	I	1	Pixel clock.
pVDE	I	1	Video data valid: • 1 = Active video. • 0 = Blanking period.
pHSync	I	1	Horizontal synchronization video timing signal.
pVSync	I	1	Vertical synchronization video timing signal.
pData	I	24	Video pixel data packed as RBG.

2. 实验流程

本章将详细描述如何在 Vivado 2014.4 的环境下完成实验。请耐心等待，仔细按照图示和文字说明进行操作。

2.1 操作步骤

1. 首先启动 Vivado 2014.4，然后在主界面点击“Create New Project”，创建工程，如下图所示：



图 2-1 Vivado 下创建新工程

2. 在弹出的向导窗口点击 Next 继续，如下图所示：

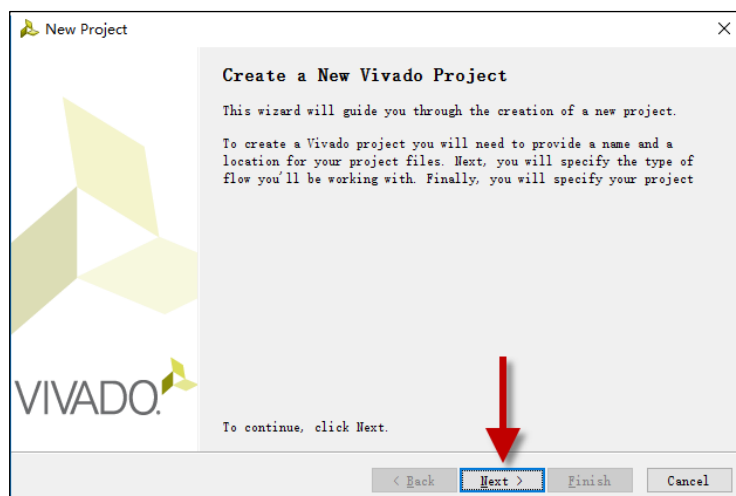


图 2-2 创建新工程向导窗口

xingdeng	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	12 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

3. 接着在窗口页面输入工程名，工程路径和相关选项，按如下信息填写（注意：为保证整个实验的流畅性，请严格按照以下信息填写）：

Project name: lab1

Project location: D:/ImageLabs

Create project subdirectory: 勾选

提示：如果本地没有 ImageLabs 这个目录，请自行创建一个

填写完成后应如下图所示，点击 Next 继续；

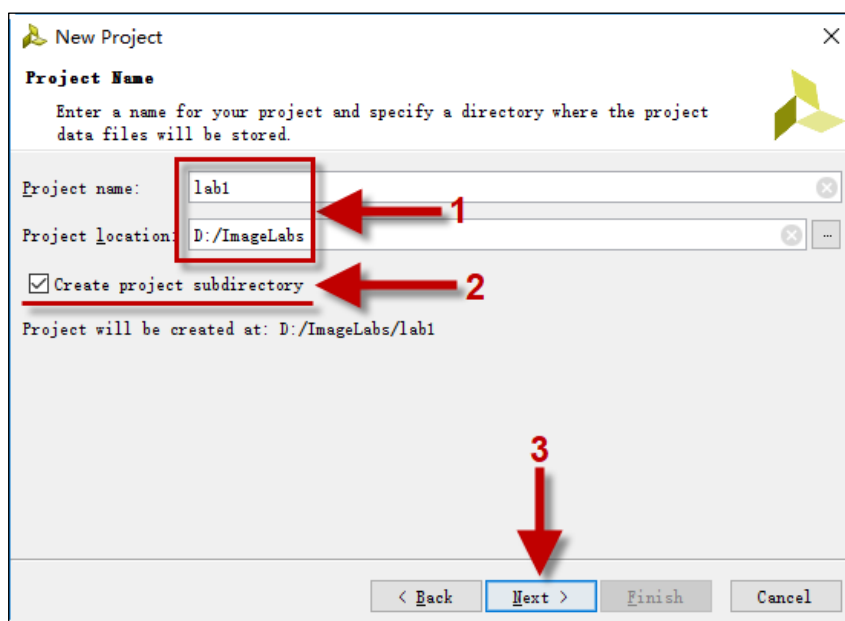


图 2-3 工程命名和保存路径

4. 接着选择工程类型，选择 RTL Project，并勾选 Do not specify sources at this time，点击 Next 继续，如下图所示：

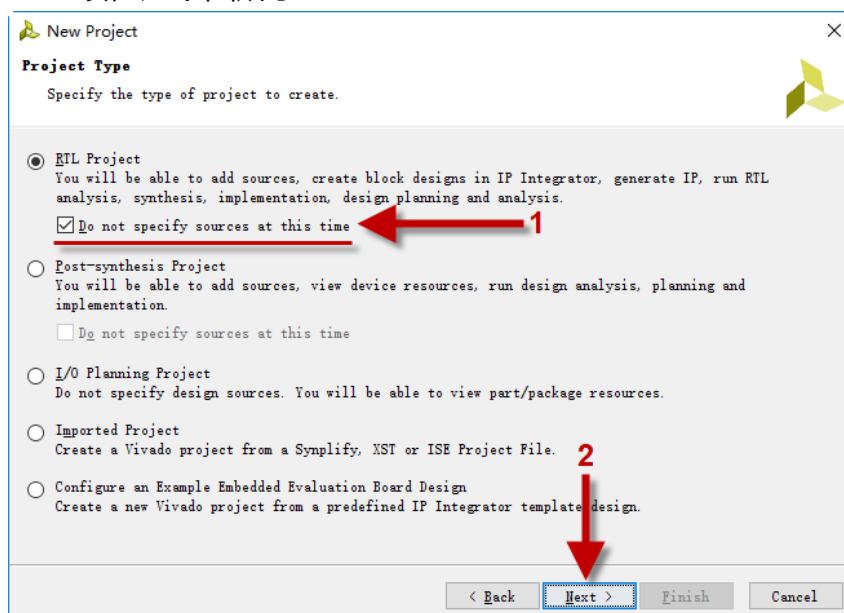


图 2-4 选择工程类型

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	13 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

5. 在 Default Part 页面按照如下信息选择目标器件:

Product category: General Purpose

Family: Kintex-7

Sub-Family: Kintex-7

Package: ffg676

Speed grade: -2

此时在器件列表中剩下的 3 个型号中选择 xc7k325tffg676-2 这个型号, 然后点击 Next 继续, 如下图所示:

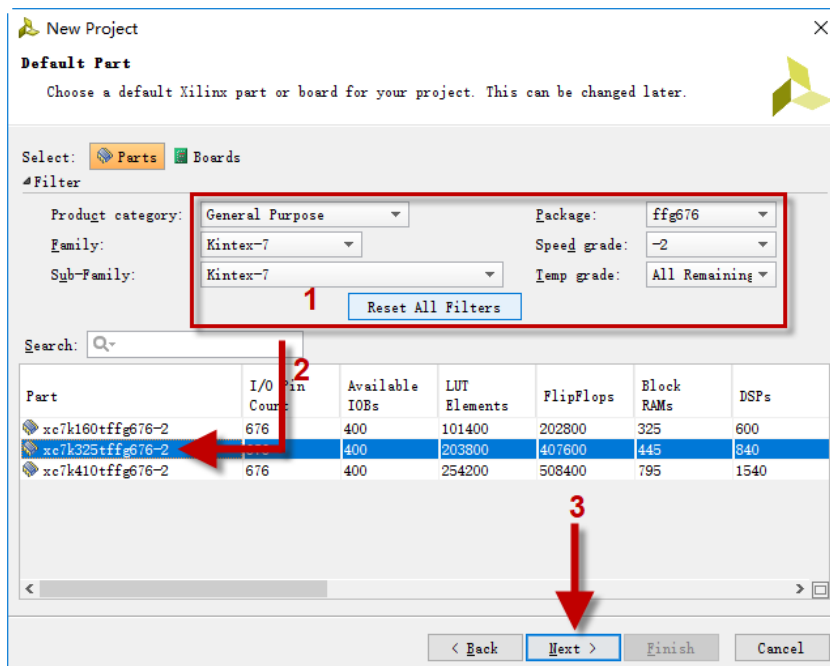


图 2-5 选择器件型号

6. 在 New Project Summary 页面直接点击 Finish 完成新工程的创建, 如下图所示:

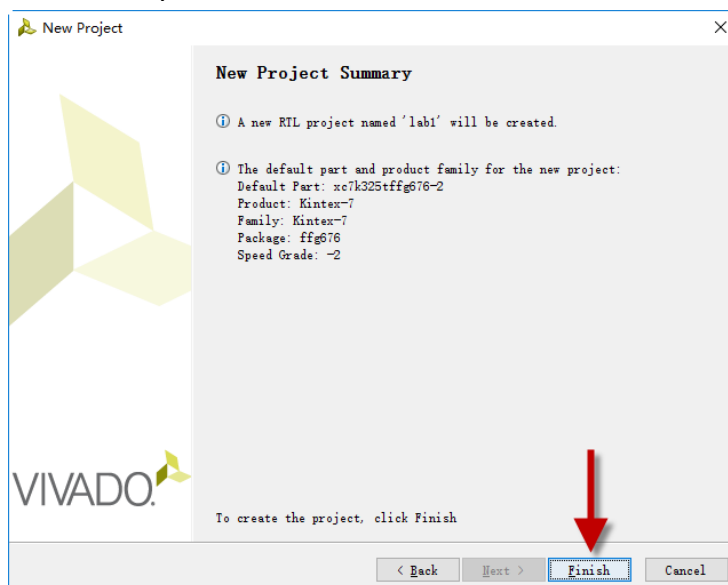


图 2-6 点击 Finish 完成工程创建

7. 接着要为新的工程添加一个 IP 库 (repo), 为此我们在 Vivado 主界面的左侧边

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	14 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

栏点击 Project Settings，然后在弹出的设置窗口中选择 IP 项，接着点击 Add Repository，整个过程如下图所示：

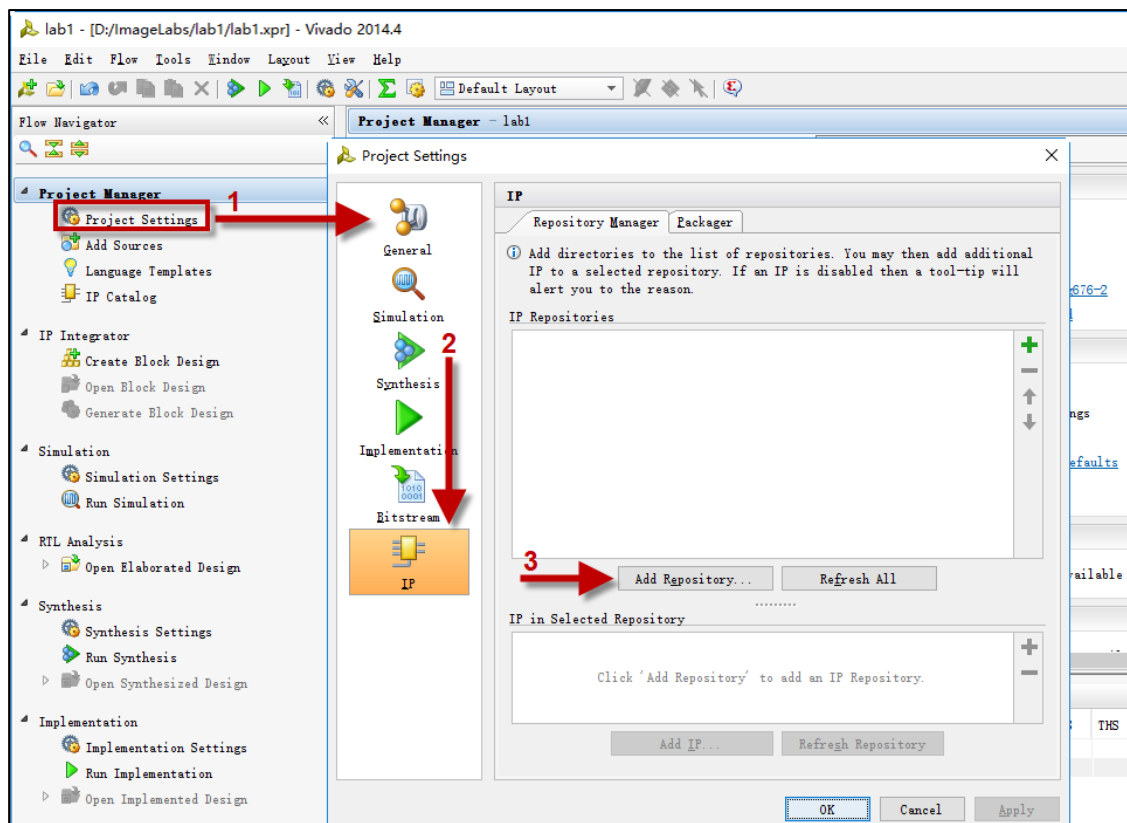


图 2-7 设置：添加 IP 库

8. 这时我们先将提前准备好的 repo.zip 压缩包解压到工程上级目录 D:/ImageLabs 中，如下图所示：

提示：如果是初学者，请按照此路径存放，如果是 Vivado 的使用熟练者，可按照自己的习惯来选择 IP 库的存放位置。



图 2-8 IP 库文件存放位置

9. 接着我们在 IP Repositories 对话框中选择刚刚保存 repo 的目录，然后点击 Select 完成 IP 库的添加，如下图所示：

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	15 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

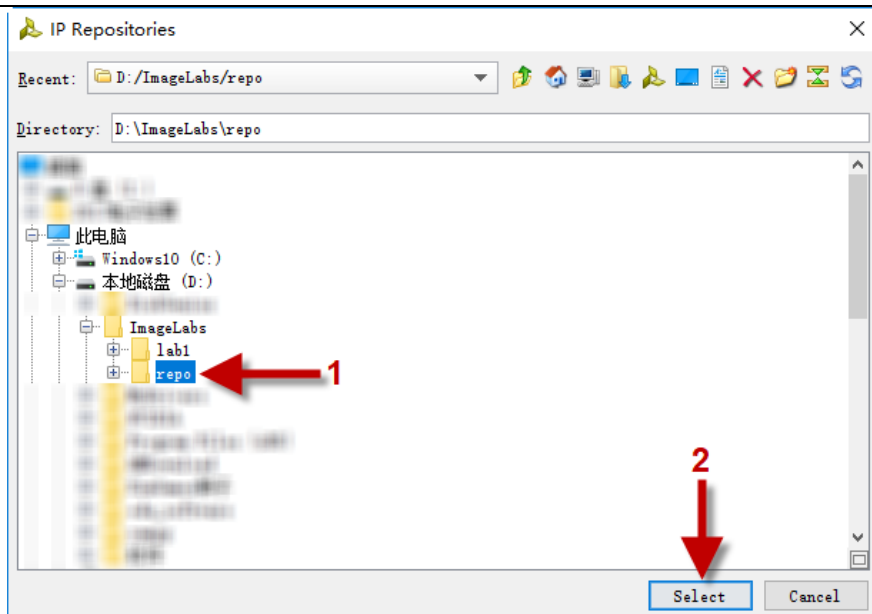


图 2-9 选择 IP 库的位置

10. 这时 Vivado 会自动扫描 repo 目录中的 IP 文件，如果正确无误，将会在 IP 列表中显示 IP 的名称，描述，和版本号，如下图所示，点击 OK 完成 IP 库的添加：

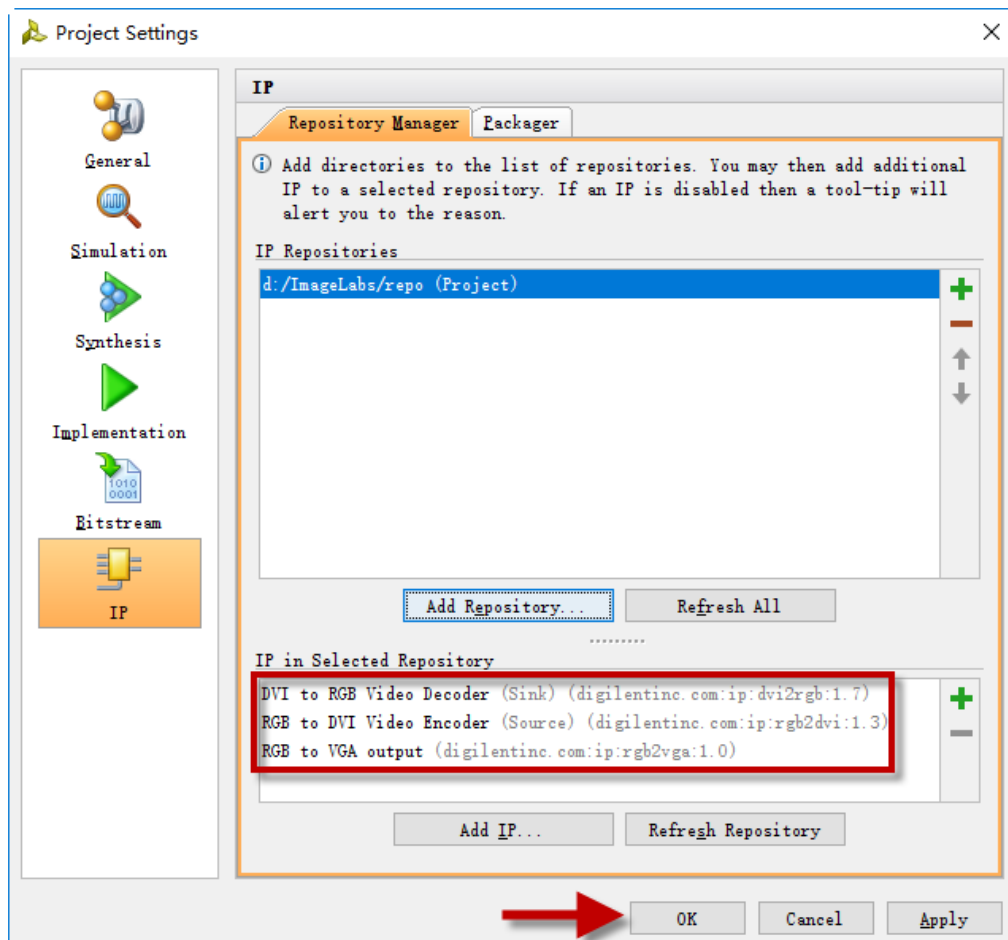


图 2-10 IP 库中的 IP 列表显示

xingdeng	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	16 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

11. 在 Vivado 主界面的左侧边栏点击 Create Block Design, 然后在弹出的窗口能够看到自动命名为 design_1, 保持不变, 直接点击 OK, 整个过程如下图所示:

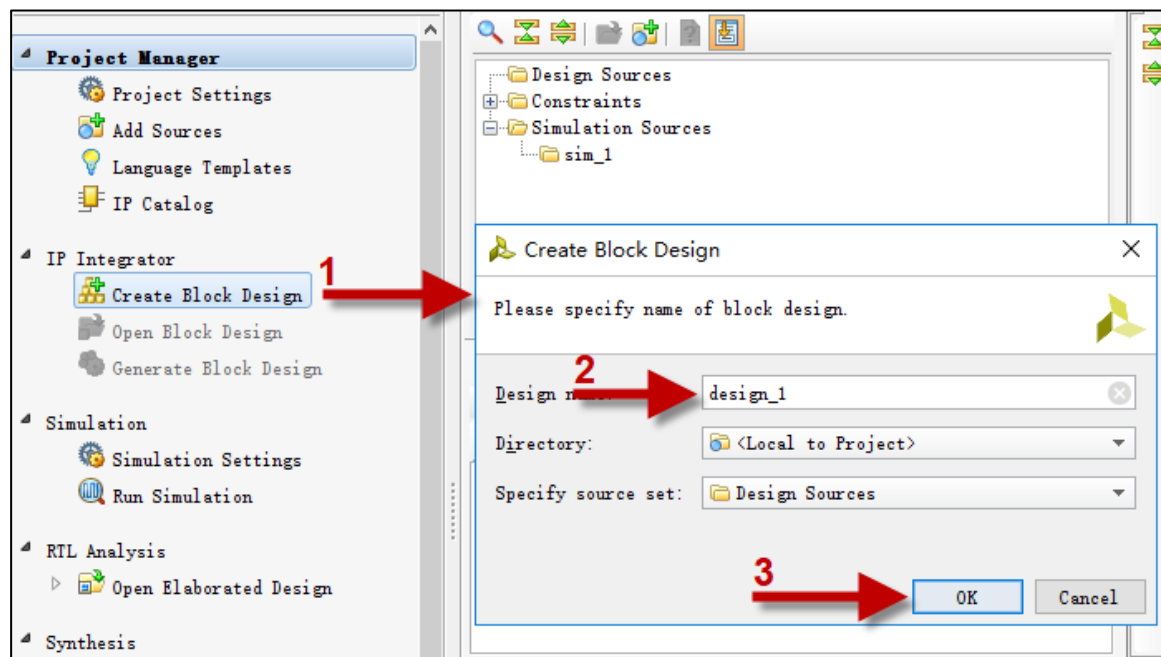


图 2-11 创建模块化设计

12. 在 Vivado 主界面的 Diagram 子窗口, 点击 Add IP, 然后在搜索栏输入: dvi2rgb, 在结果栏双击 DVI to RGB Video Decoder (Sink), 添加 DVI2RGB IP, 整个过程如下图所示:

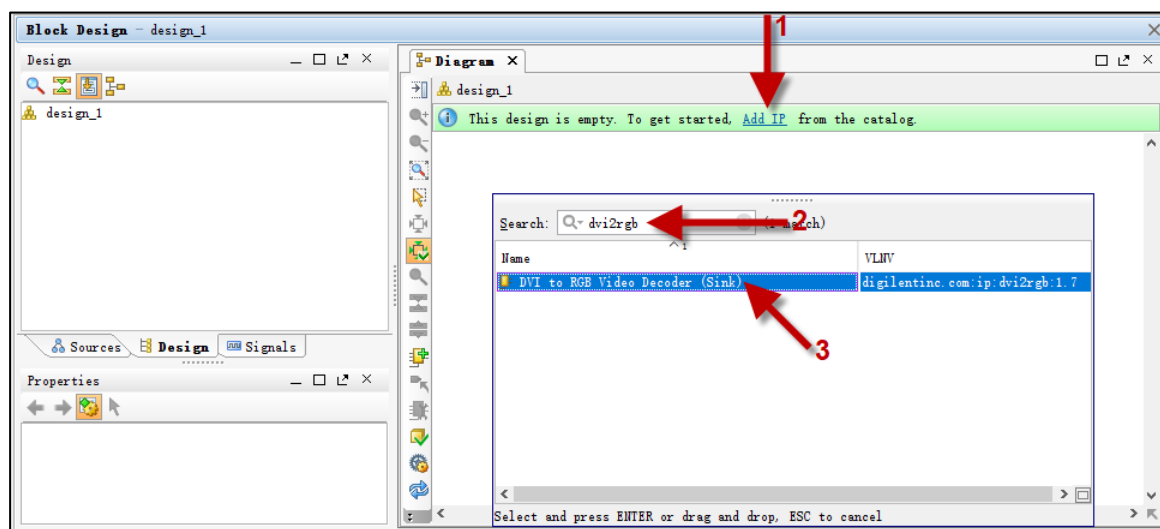


图 2-12 添加 DVI2RGB IP

13. 接着按同样的方法添加 RGB2DVI IP, 在 Diagram 子窗口的左侧边栏, 点击 Add IP 的图标, 然后在搜索栏输入: rgb2dvi, 在结果栏双击 RGB to DVI Video Encoder (Source), 添加 DVI2RGB IP, 整个过程如下图所示:

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	17 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

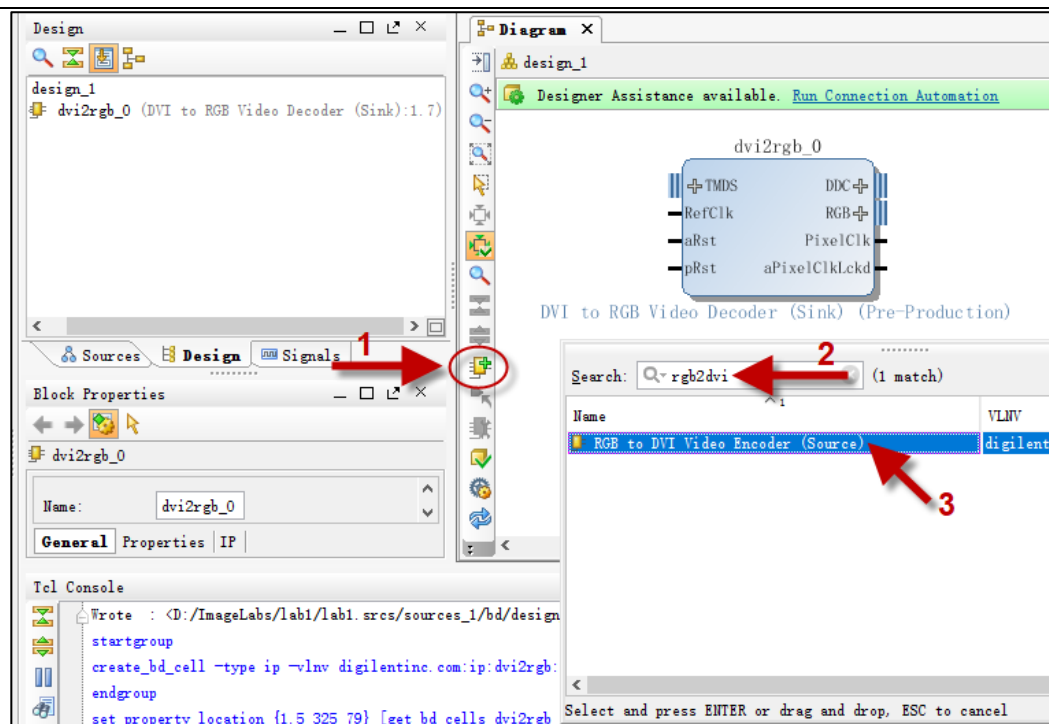


图 2-13 添加 RGB2DVI IP

14. 添加好 DVI2RGB 和 RGB2DVI 两个 IP 后的视图如下图所示：

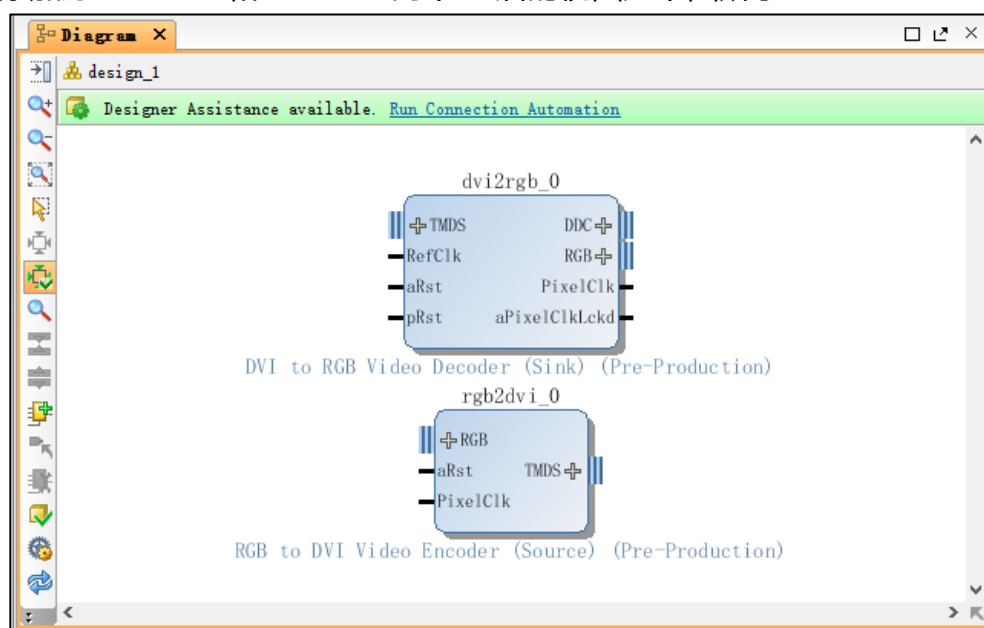


图 2-14 添加 IP 后的视图

15. 接着对这两个 IP 分别进行配置 (Customize)：

先配置 DVI2RGB：鼠标右键单击 DVI2RGB IP，然后在菜单中选择 Customize Block，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	18 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

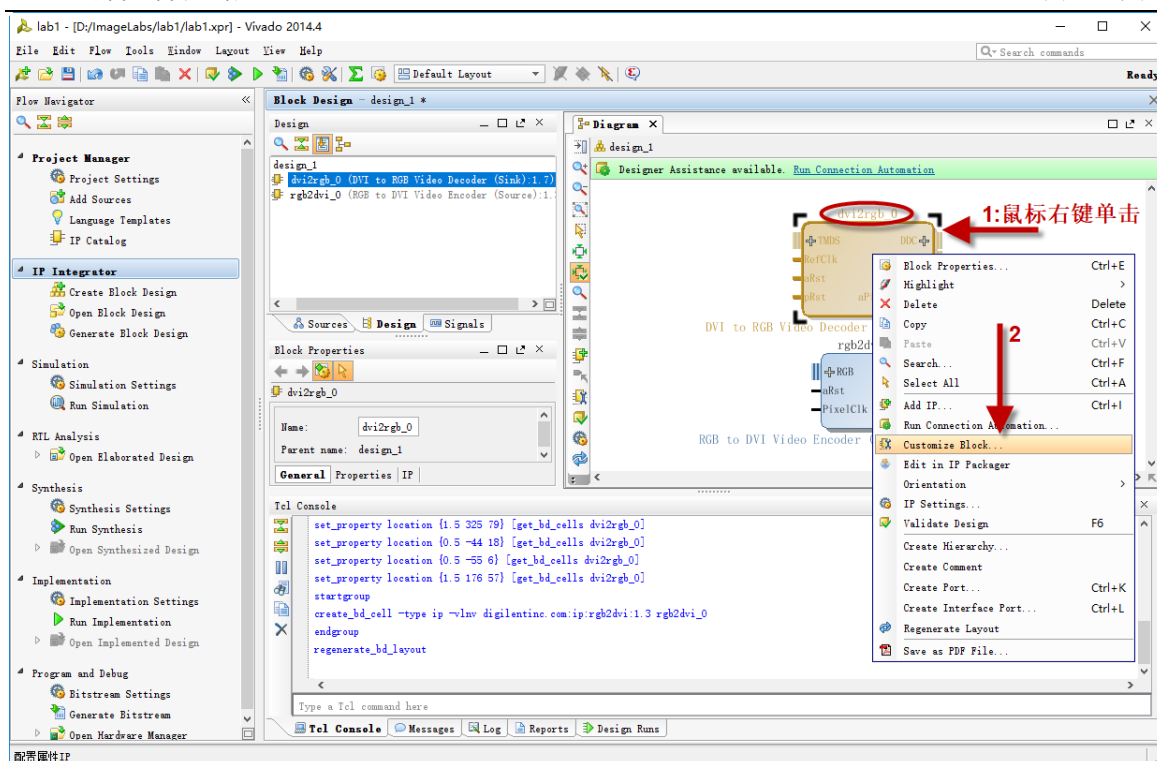


图 2-15 配置 DVI2RGB IP

在配置界面按照如下图所示进行配置，其中：

勾选 Enable DDC ROM 的作用是启用 EDID ROM，主要是用来告诉 HDMI/DVI 信号源 Sink 支持哪些显示分辨率；

由于 SWORD4.0 的开关和按键都是常为高电平的，所以将该 IP 设置为复位低电平有效，即不勾选 Reset active high；

在 Preferred resolution 中，保持默认设置即可，该项的含义是设置输入视频的默认分辨率；

其他选项保持默认不变即可，点击 OK 继续；

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	19 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

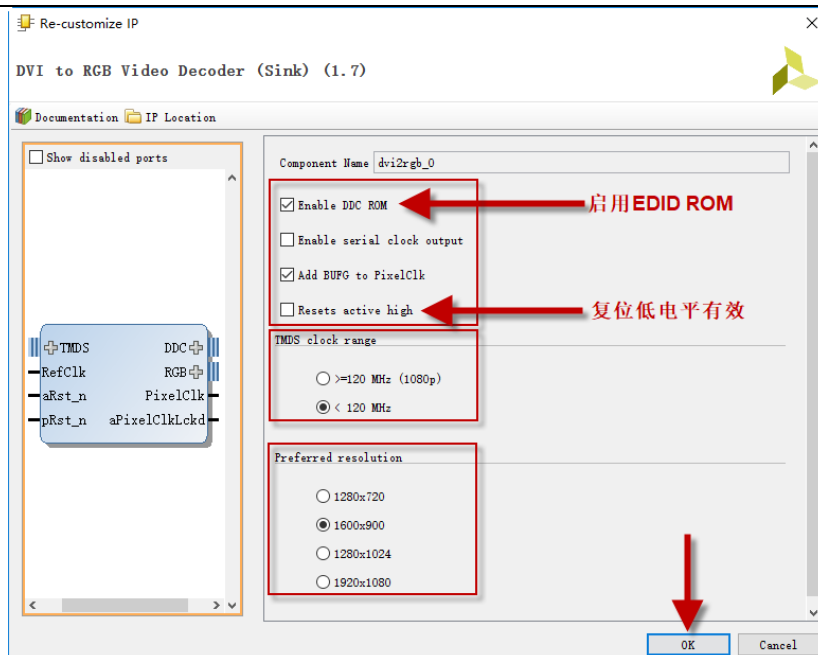


图 2-16 DVI2RGB IP 参数设置

接着配置 RGB2DVI，按照之前同样的操作，鼠标右键单击 RGB2DVI IP，然后在菜单中选择 Customize Block，在配置界面按照如下图所示进行配置，其中：

由于 SWORD4.0 的开关和按键都是常为高电平的，所以将该 IP 设置为复位低电平有效，即不勾选 Reset active high；

勾选 Generate SerialClk internally from pixel clock；

MMCM/PLL：勾选 PLL

TMD clock range：勾选 <120MHz（720p）

其他选项保持默认不变即可，点击 OK 继续；

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	20 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

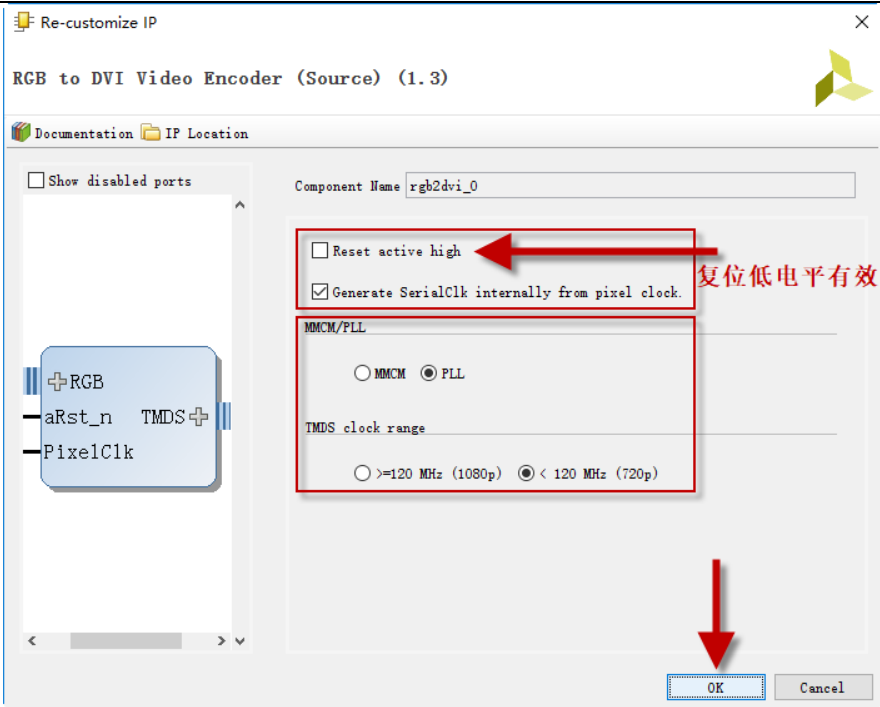


图 2-17 RGB2DVI IP 参数设置

16 . 设置好后回到 Diagram 子窗口界面，然后鼠标移动到 DVI2RGB IP 的 TMDs 端口上，此时光标会变成一个，先点击左键，此时信号名会高亮成黄色，如下图所示：

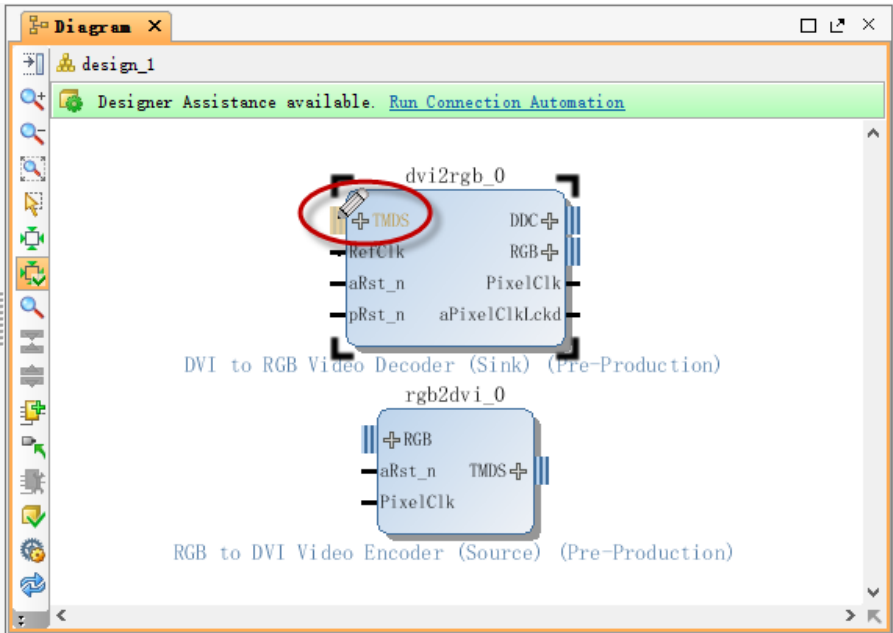


图 2-18 选择 DVI2RGB IP 的 TMDs 端口

然后鼠标右键再单击一次，在弹出的菜单中选择点击 Make External，将 TMDs 端口引出到外部，如下图所示：

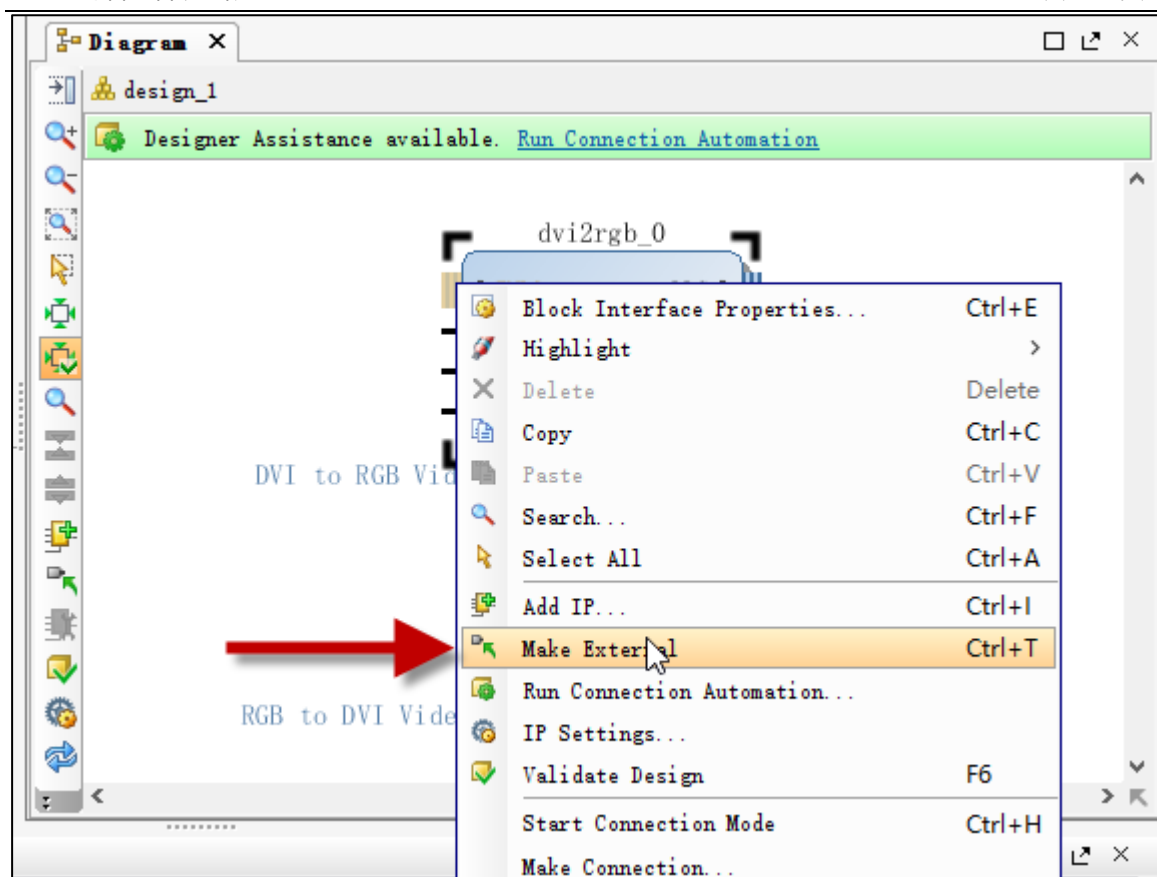


图 2-19 将端口引出到外部

17. 按照同样的方法，将鼠标移动到 DVI2RGB IP 的 DDC 端口上，此时光标变成一个，先点击左键，此时信号名会高亮成黄色，如下图所示：

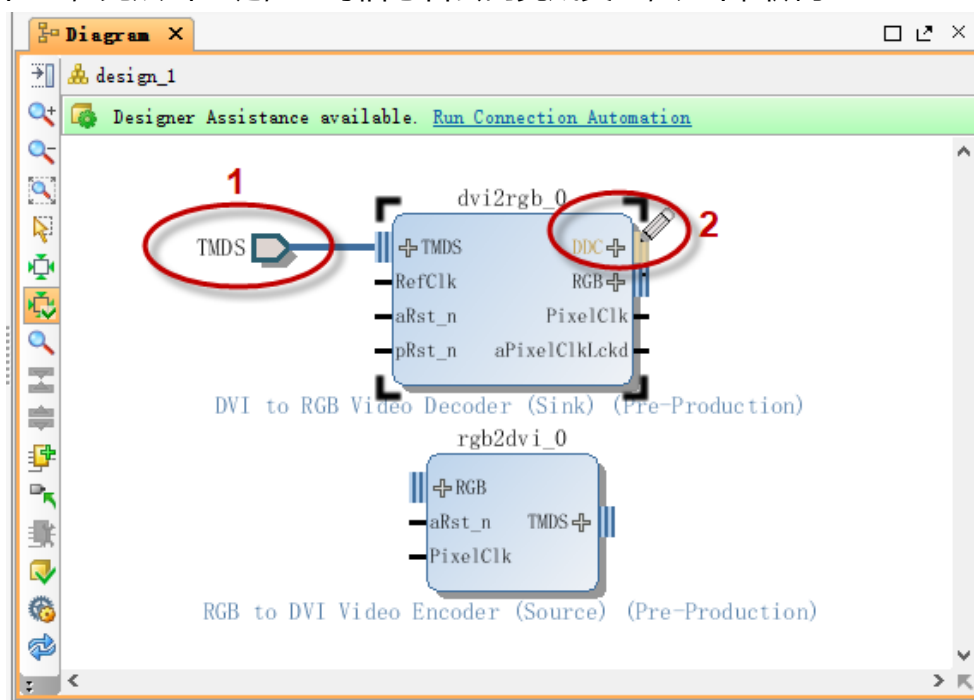


图 2-20 引出端口

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	22 of 40
	作者	修改日期	公开	
Joseph Xu		2019/2/14		

然后鼠标右键再单击一次，在弹出的菜单中选择点击 Make External，将 DDC 端口引出到外部，如下图所示：

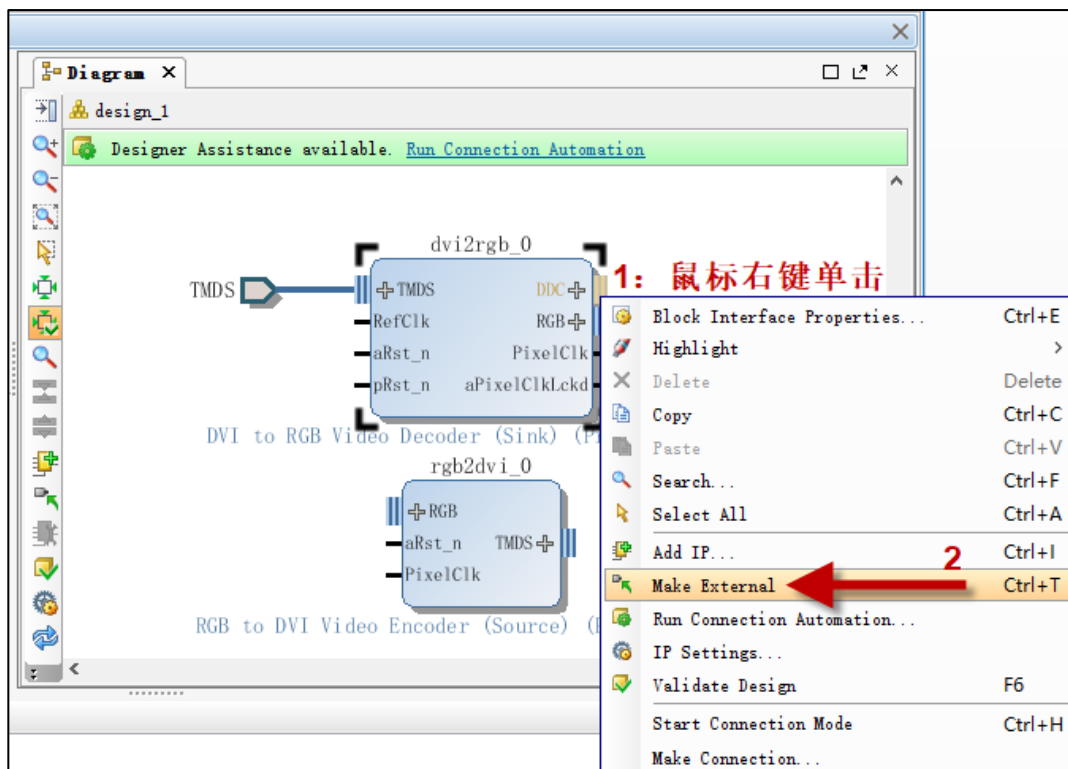


图 2-21 将 DDC 端口引出到外部

18. 按照和前面两次同样的操作方法，将 RGB2DVI IP 的 TMD5 端口引出到外部，引出的效果如下图所示：

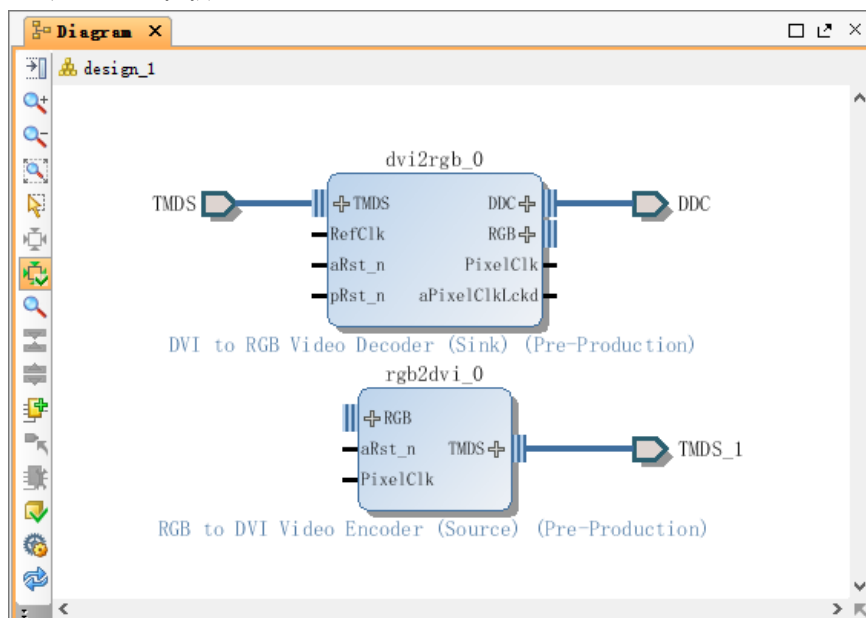


图 2-22 端口引出

19. 接着修改端口名，将其改为更容易理解的信号名：

首先鼠标左键选择 DVI2RGB IP 的 TMD5 端口，可以看到该端口符号高亮为黄

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	23 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

色，然后在 Diagram 窗口左边的属性编辑框看到 Name 文本框同步显示为 TMDS，在该文本框输入 TMDS_IN，然后回车，完成端口的重命名，整个过程如下图所示；

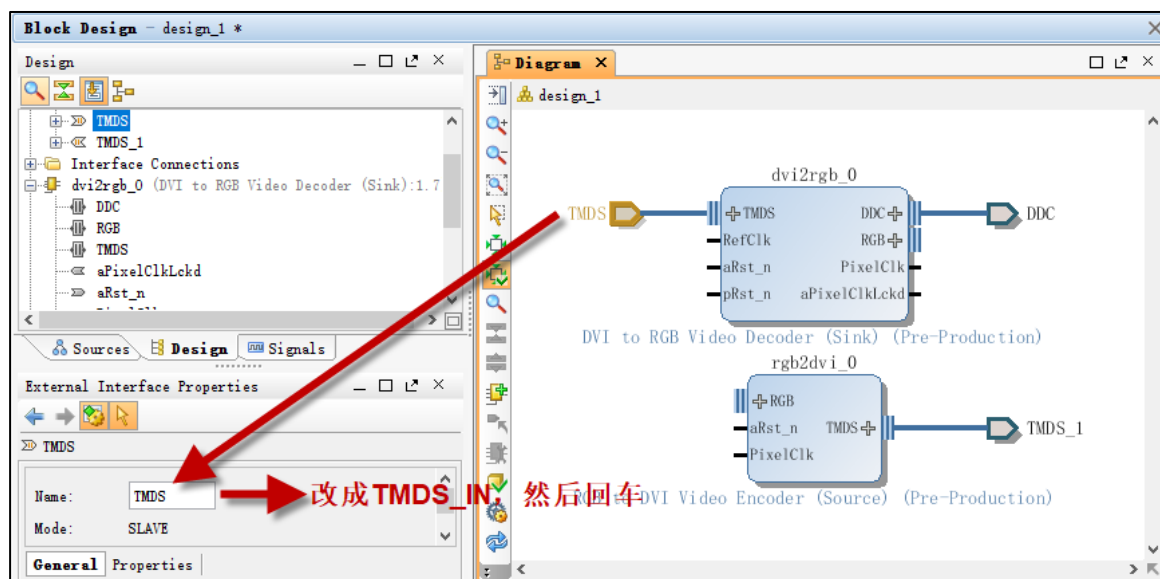


图 2-23 端口如何重命名

改好后的效果如下图所示，请按照同样的方法将上图中的 TMDS_1 信号重命名为 TMDS_OUT；

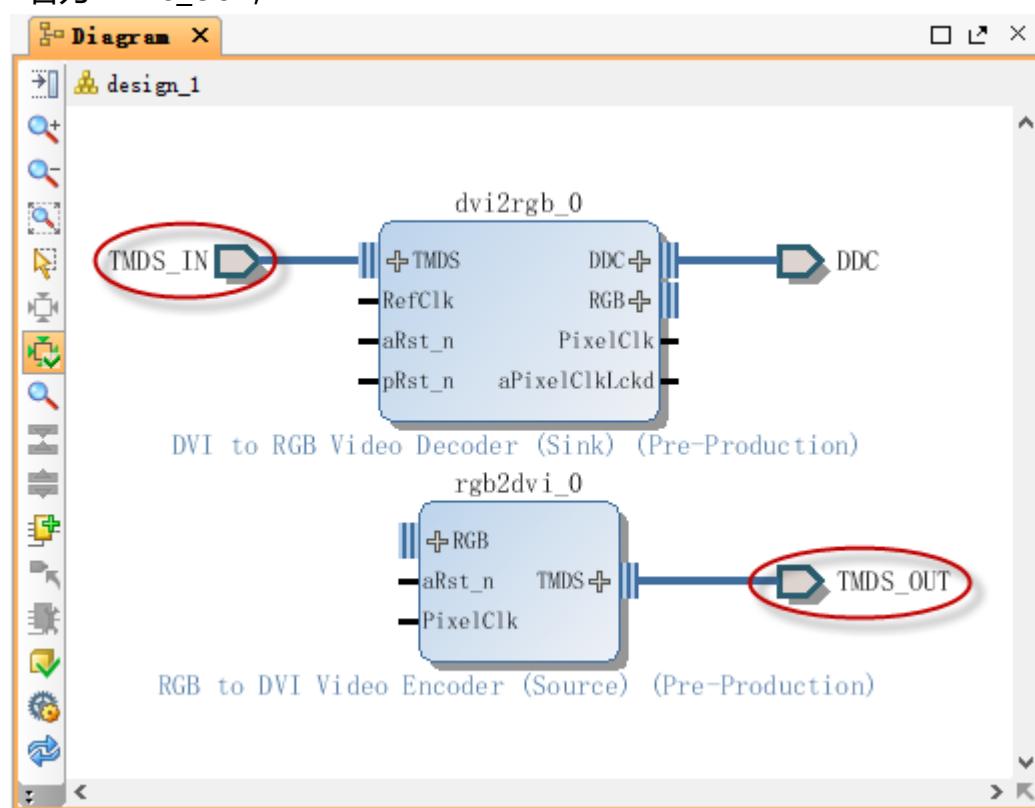


图 2-24 端口重命名后的视图

20. 接着进行 IP 之间的连接，将鼠标左键单击 DVI2RGB IP 的 RGB 端口，此时光标

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	24 of 40
	作者	修改日期	公开	
Joseph Xu		2019/2/14		

会变成一个，然后按住鼠标不放拖拽到 RGB2DVI 的 RGB 端口处，能看到有一个绿色的勾，表示两者连接符合规则，此时松开鼠标，工具会自动将 DVI2RGB 和 RGB2DVI 的 RGB 端口用连线连接，整个过程如下图所示：

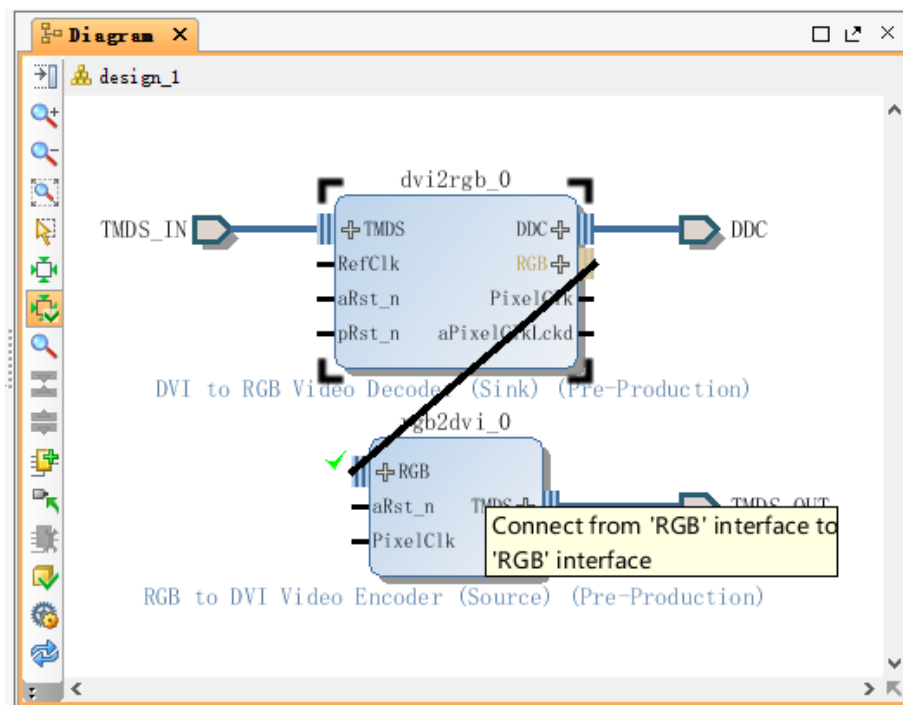


图 2-25 连接 RGB 端口

21. 按照同样的方式，连接 DVI2RGB 和 RGB2DVI 的 PixelClk 端口，连接完成后的视图应如下图所示，接着点击左边栏的 **Regenerate Layout** 图标，对视图进行布局调整，整个过程如下图所示：

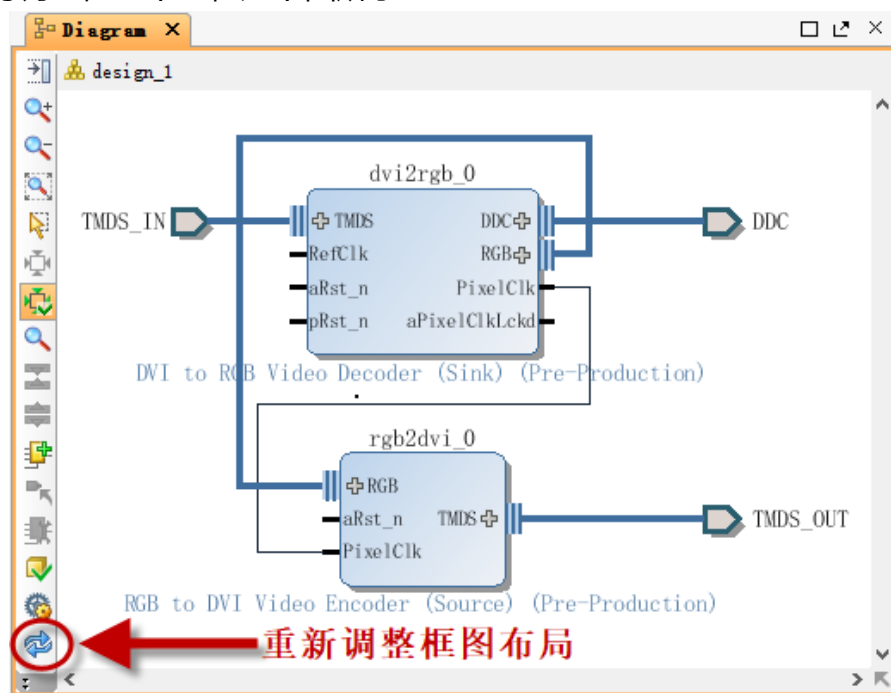


图 2-26 调整框图布局

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	25 of 40
	作者	修改日期	公开	
Joseph Xu		2019/2/14		

22. 调整后的布局如下图所示，接着点击左边栏的 Add IP 图标，然后在搜索栏中输入 clock，在结果栏中双击 Clocking Wizard，添加时钟向导模块，整个过程如下图所示：

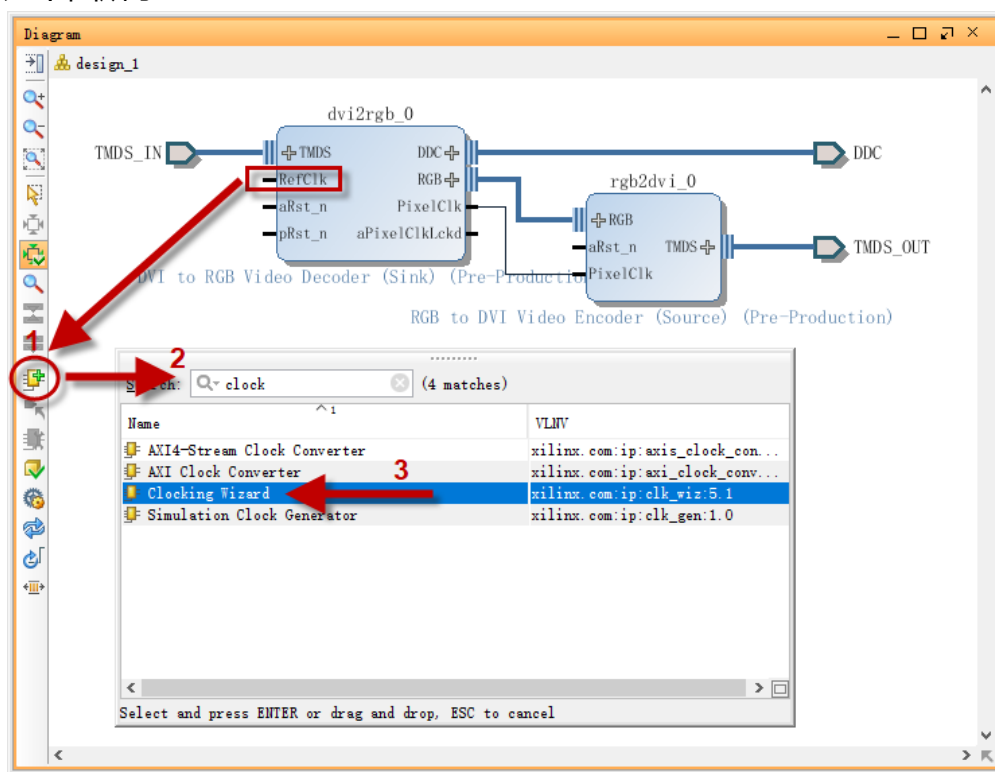


图 2-27 添加时钟向导 IP

23. 添加 Clocking Wizard IP 后的视图如下图所示：

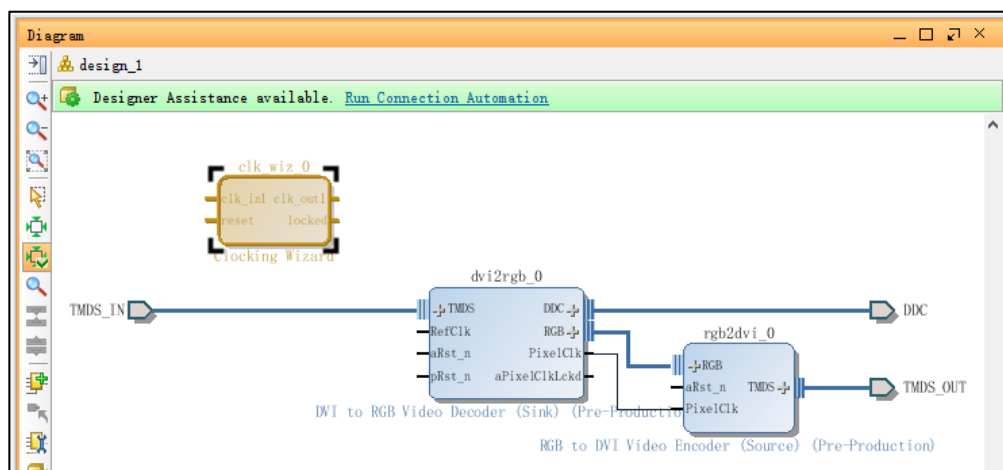


图 2-28 添加时钟向导 IP 后的视图

24. 接着鼠标右键单击 Clocking Wizard IP，然后在弹出的菜单中选择 Customize Block，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	26 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

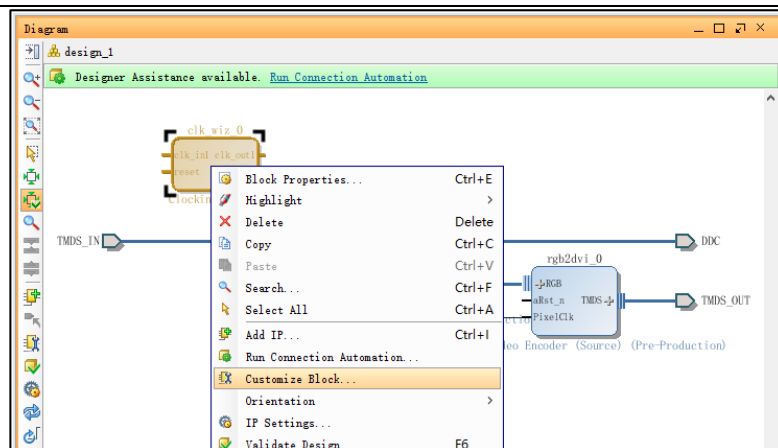


图 2-29 点击 IP 的配置

25. 在弹出的 IP 配置窗口按照如下信息配置：

在 Clocking Options 栏下：

Primitive 项：选择 PLL；

Input Frequency(MHz)：设置为 Manual，并将数值设置为 200.000

Source：选择 Differential clock capable pin

整个过程如下图所示：

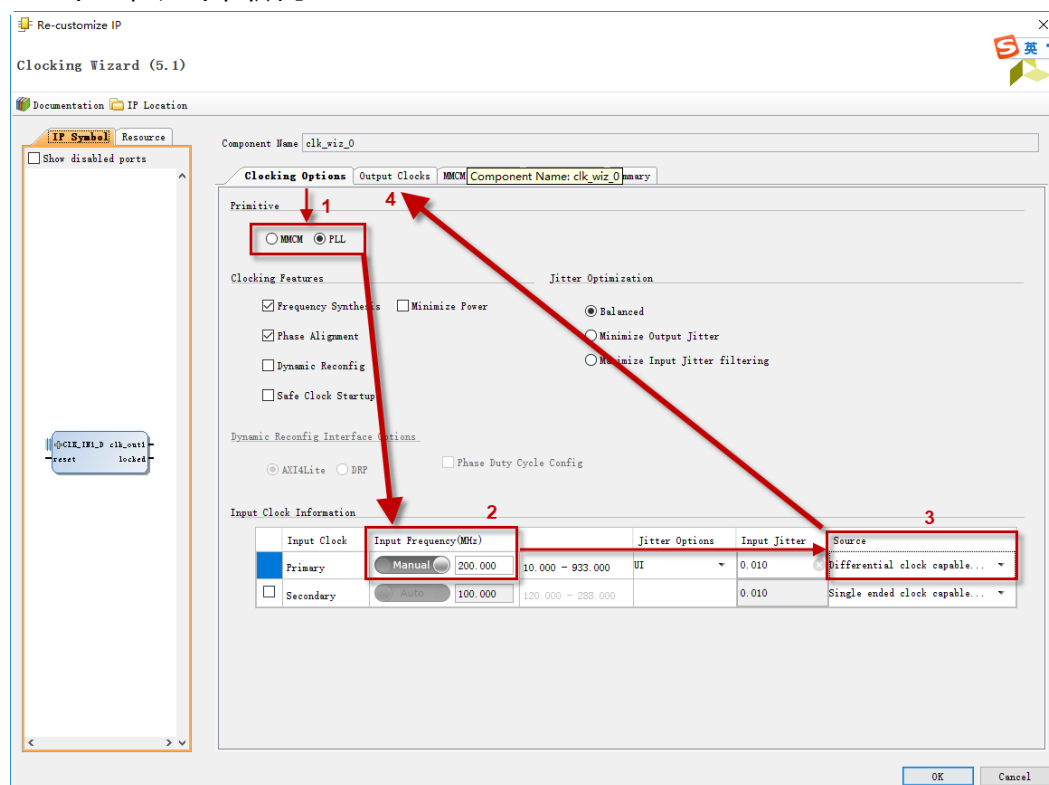


图 2-30 时钟向导设置——Clocking Options

在 Output Clocks 栏下：

Output Clock：勾选 clk_out1；

Output Freq (MHz)：Requested 处输入 200.000；

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	27 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

Enable Optional Inputs/Outputs: 勾选 reset 和 locked;

Reset Type: 勾选 Active Low

其他设置保持默认不变, 设置后的页面参数如下图所示, 点击 OK 确定:

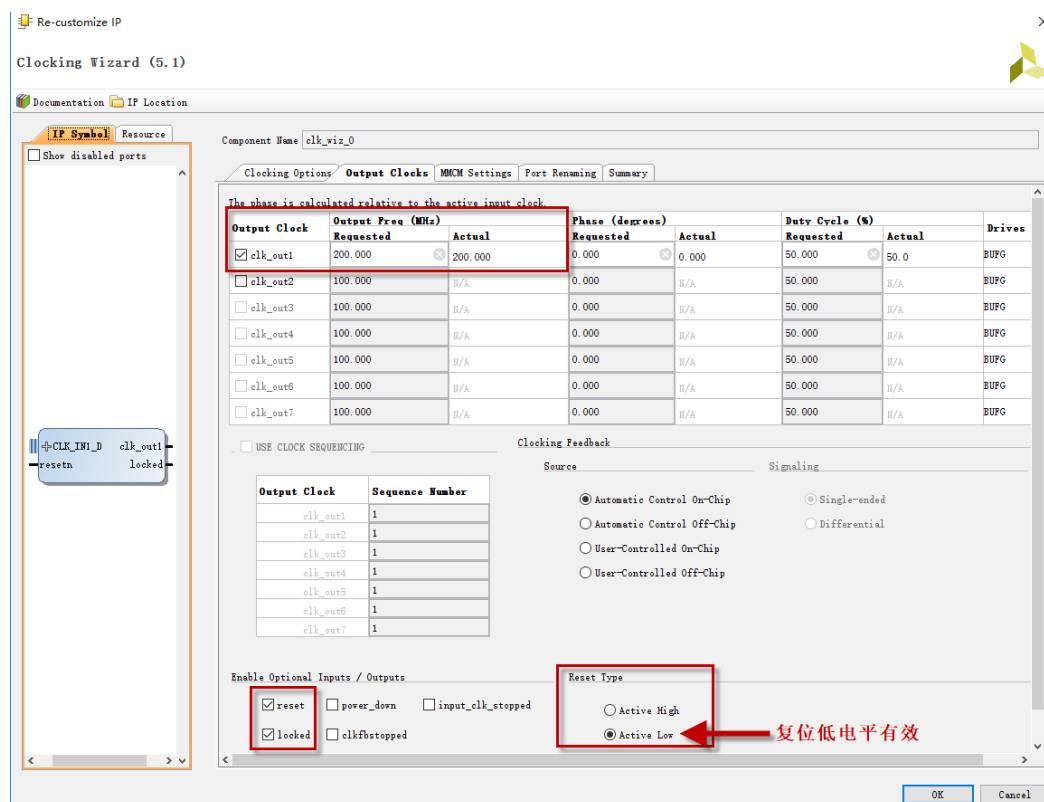


图 2-31 时钟向导设置——Output Clocks

26. 设置完成后的 IP 视图, 如下图所示, 请检查图中各处是否和你的设置一样;

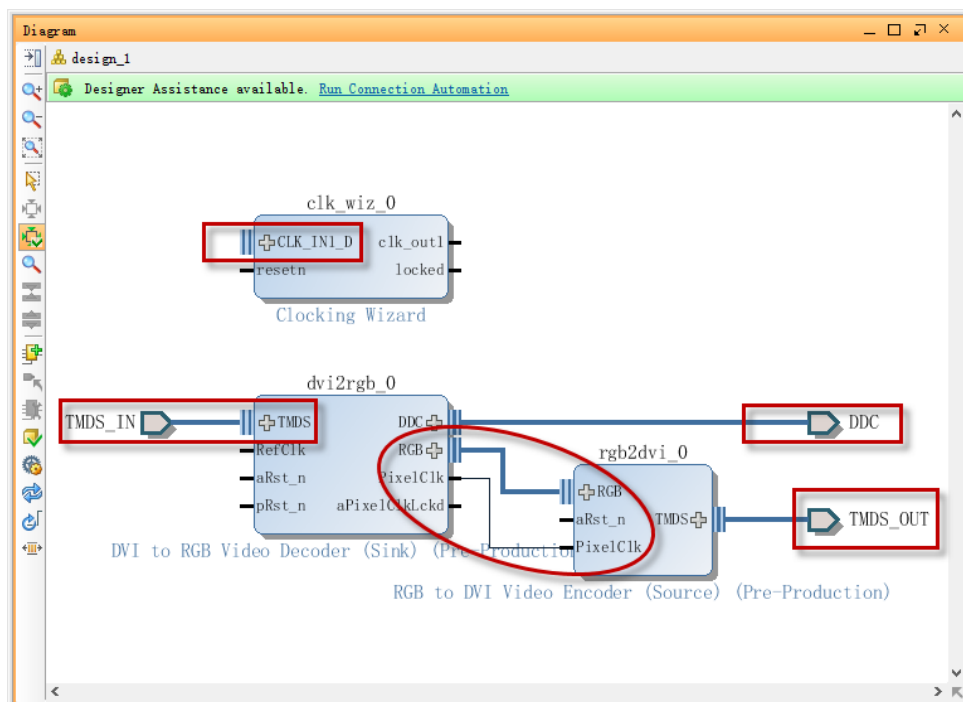


图 2-32 各 IP 设置好后的视图

XINGDENG	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	28 of 40
	作者	修改日期	公开	
Joseph Xu		2019/2/14		

27. 接着按照之前 16 和 17 的方法，将 Clocking Wizard 的 CLK_IN1_D 端口和 resetn 端口引到外面，引出后的结果如下图所示：

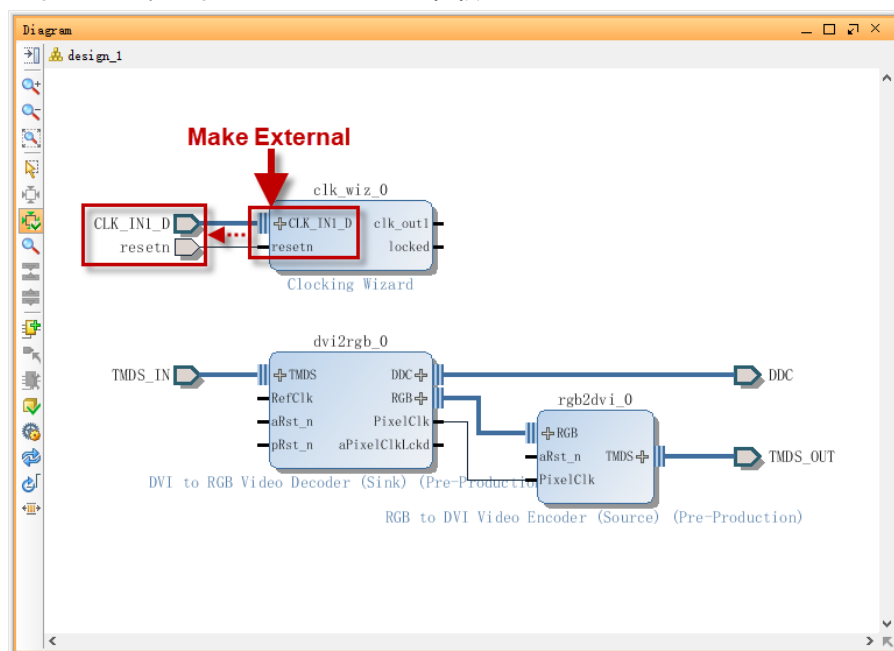


图 2-33 将时钟向导的 2 个输入端口引出到外部

28. 接着参考之前 19 的方法，将 clk_wiz_0 的 CLK_IN1_D 端口重命名成 sys_clk200，改名后的视图如下图所示；

接着进行各 IP 间的连线，参考 20 的方法，首先将 clk_wiz_0 的 clk_out1 和 dvi2rgb_0 的 RefClk 进行连接；接着将 clk_wiz_0 的 resetn 和 dvi2rgb_0 的 aRst_n, pRst_n, 以及 rgb2dvi_0 的 aRst_n, 一并连接起来，连接后的视图应如下图所示，接着点击 Diagram 子窗口左边的 Regenerate Layout 图标进行连接后的布局调整：

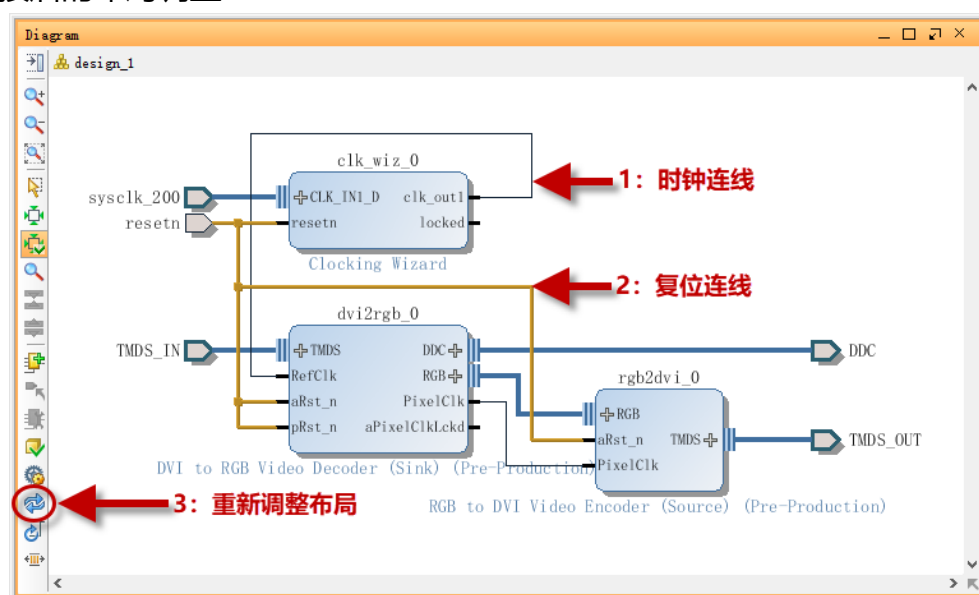


图 2-34 各 IP 之间的连接图

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	29 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

29. 重新调整布局后的视图如下图所示:

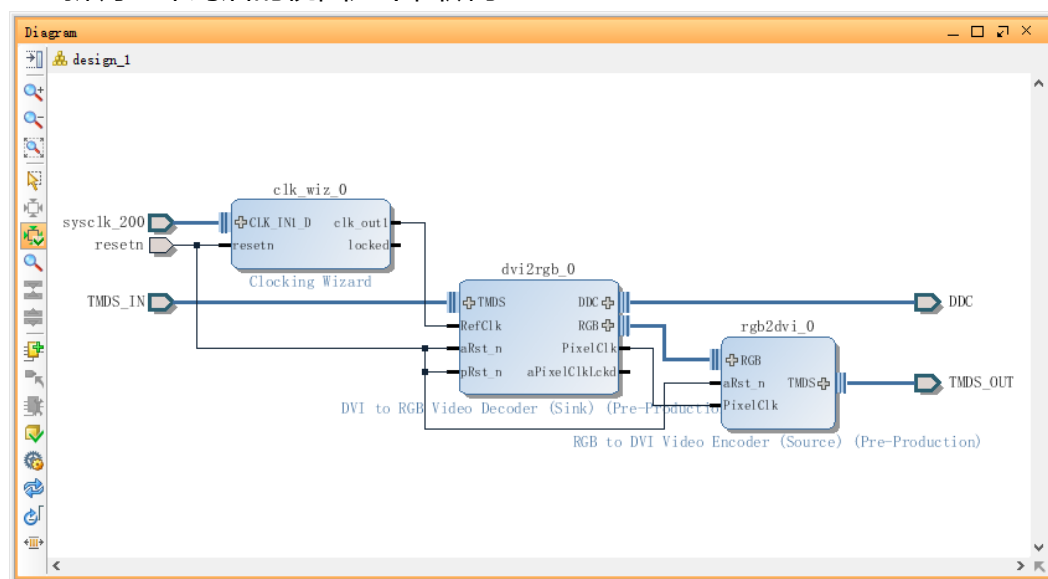


图 2-35 重新调整布局后的视图

30. 至此, 实验 1 的模块设计完成, 点击 Vivado 主界面菜单栏下的保存图标进行保存, 如下图所示:

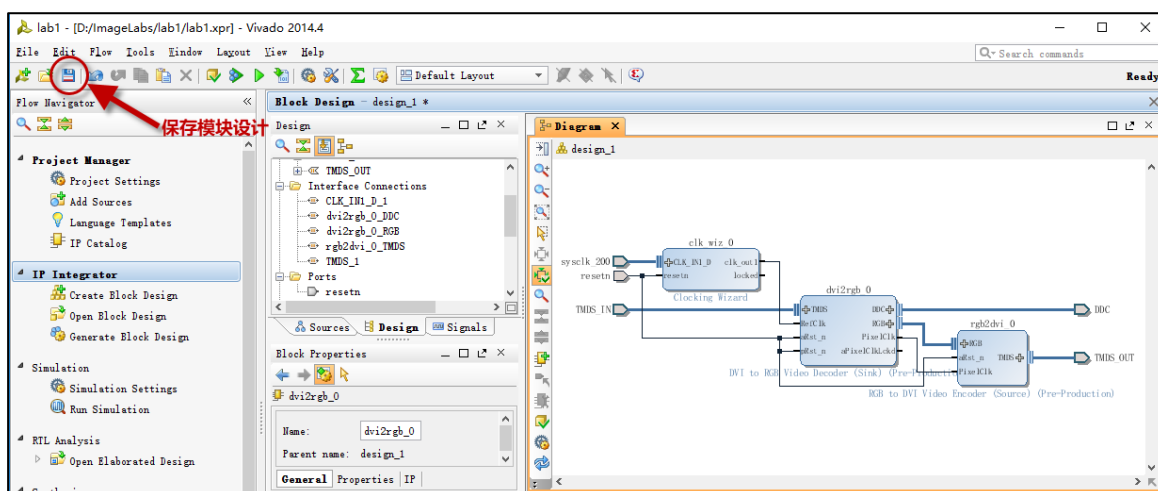


图 2-36 保存模块设计

31. 接着创建顶层文件, 点击 Vivado 主界面下的 Source 标签, 然后右键单击 design_1 图标, 在弹出的菜单中选择 Create HDL Wrapper, 整个过程如下图所示:

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	30 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

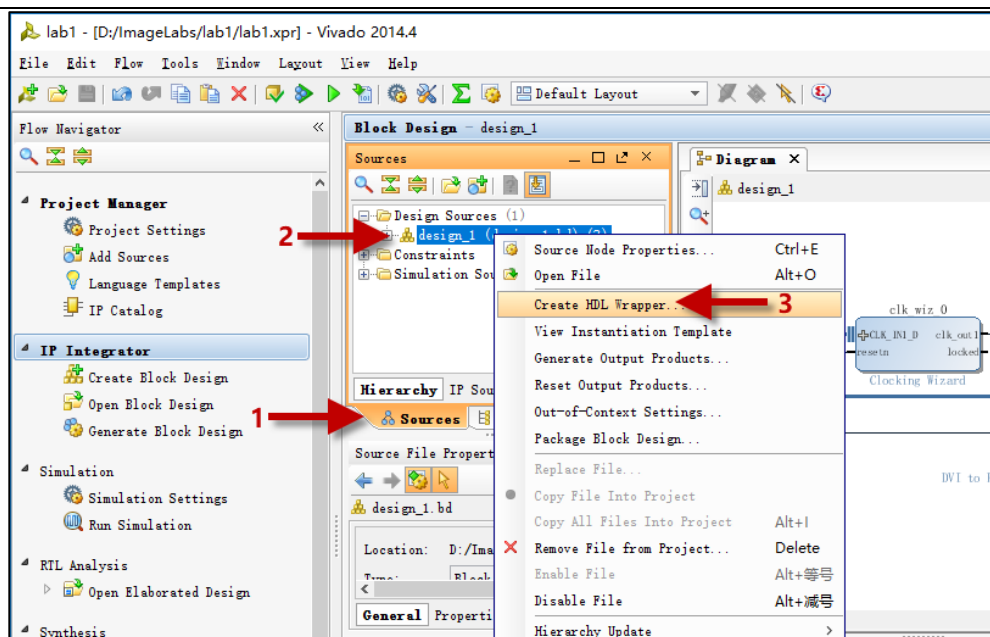


图 2-37 创建顶层文件

在弹出的窗口中，选择 Let vivado manage wrapper and auto-update，点击 OK 继续，如下图所示：

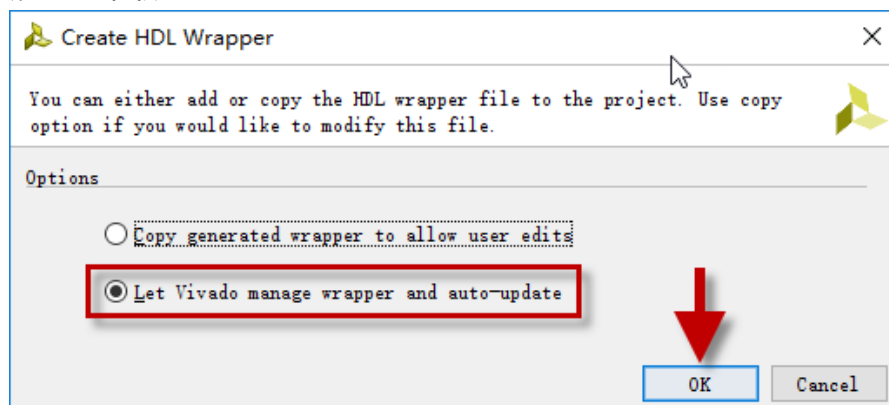


图 2-38 自动更新顶层文件

接着在 Vivado 主界面的 Source 窗口栏，我们就能看到 Vivado 自动根据 design_1.bd 模块设计文件自动生成的顶层文件了，如下图所示；

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	31 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

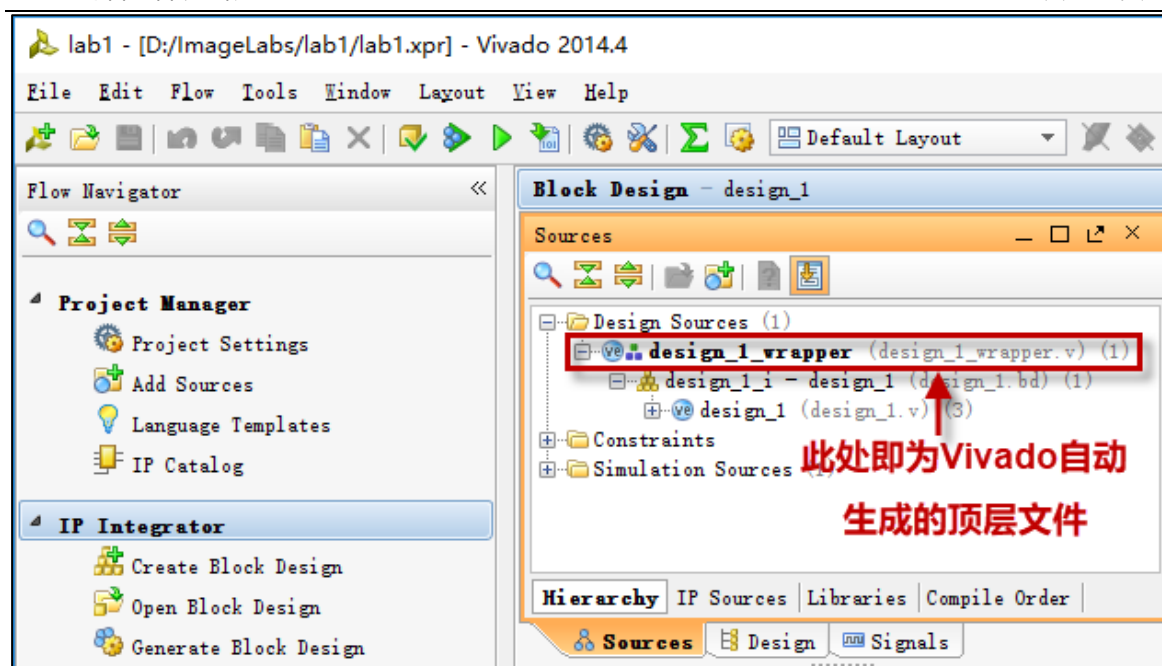


图 2-39 查看自动生成的顶层文件

32. 接着我们点击 Vivado 主界面左边的 Open Elaborated Design 项，如下图所示：

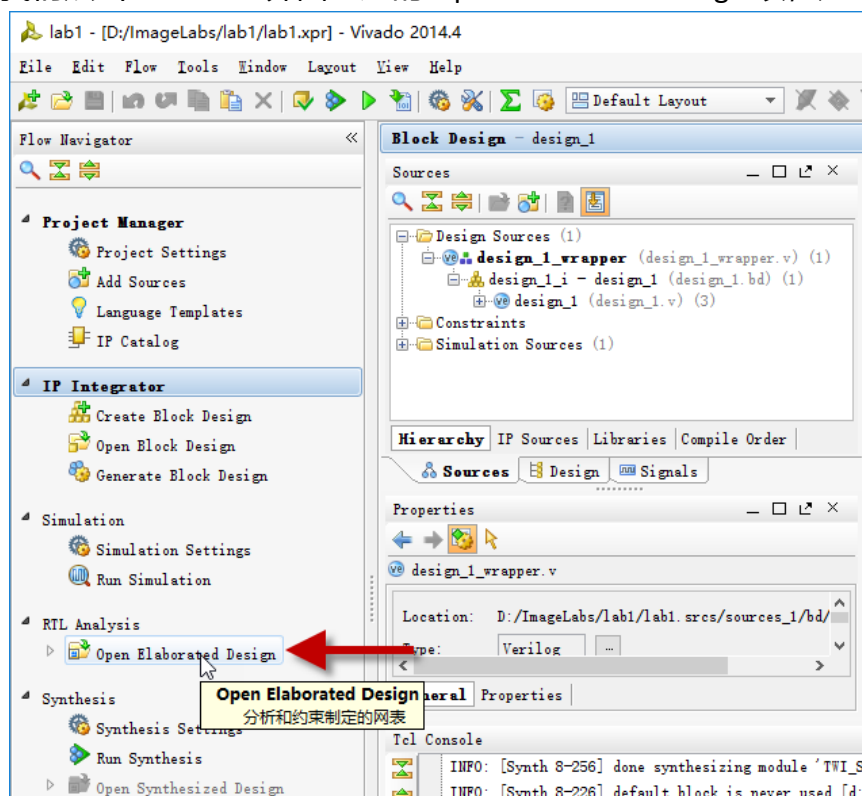


图 2-40 Open Elaborated Design

在接下来的视图中，我们首先在 Vivado 主界面点击 RTL Analysis，接着确保处于 I/O Planning 子项，然后点击主界面下方的 I/O Ports 窗口最大化按钮，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	32 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

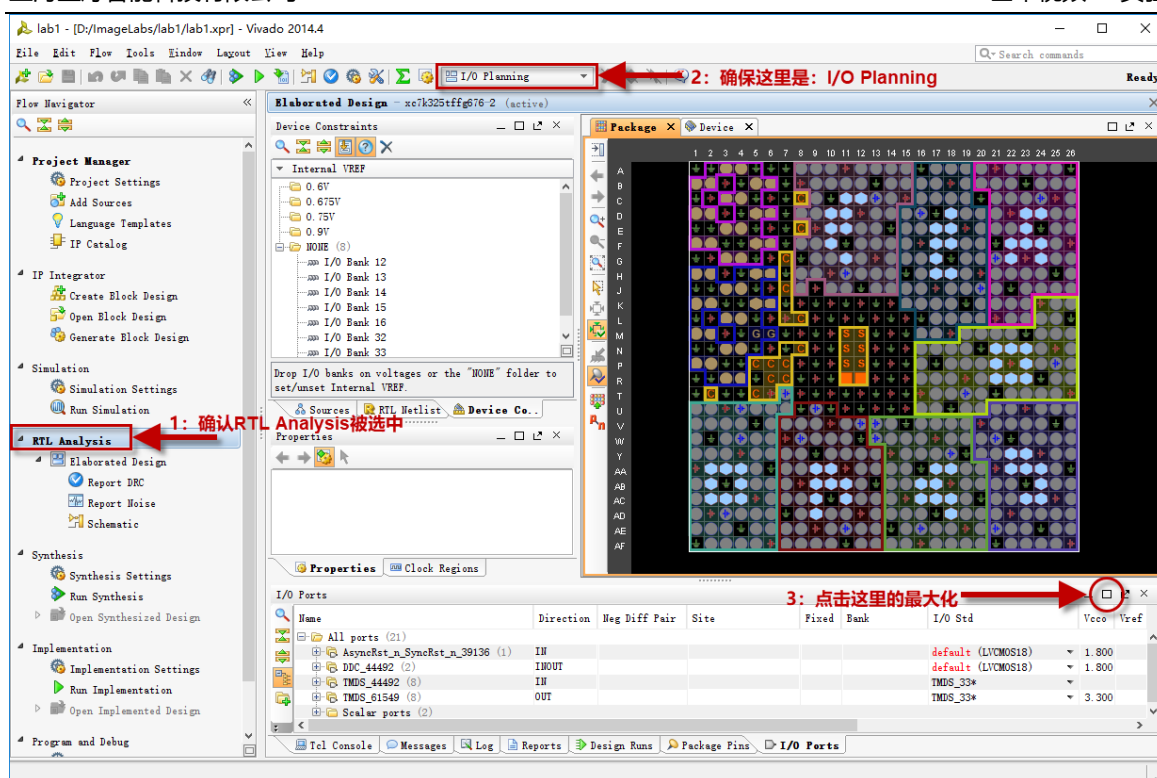


图 2-41 进行管脚映射

33. 在最大化的 I/O Ports 窗口，按照如下信息进行设置：

表 2-1 端口设置表

Name	Site	I/O Std
resethn	W13	LVCMOS18
ddc_scl_io	V24	LVCMOS33
ddc_sda_io	U26	LVCMOS33
TMDS_IN_data_p[2]	AF19	LVDS
TMDS_IN_data_p[1]	AD16	LVDS
TMDS_IN_data_p[0]	AA14	LVDS
TMDS_IN_clk_p	AA17	LVDS
TMDS_OUT_data_p[2]	AE17	LVDS
TMDS_OUT_data_p[1]	AF14	LVDS
TMDS_OUT_data_p[0]	AE18	LVDS
TMDS_OUT_clk_p	AD15	LVDS
sys_clk200_clk_p	AC18	LVDS

设置后的参数应如下图设置，请填写并核对，检查无误后，点击 Vivado 主界面菜单栏下的保存图标将管脚映射保存为文件，如下图所示：

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	33 of 40
	作者	修改日期	公开	
Joseph Xu		2019/2/14		

34. 接着我们还要对刚刚保存的文件进行一处修改。为此，先在 Vivado 的主界面选中 Project Manager，接着在右边的 Source 窗口栏中依次展开 Constraints → constrs_1，在展开后的最下层可以看到刚刚保存的 lab1.xdc 文件，双击这个文件打开，整个过程如下图所示：

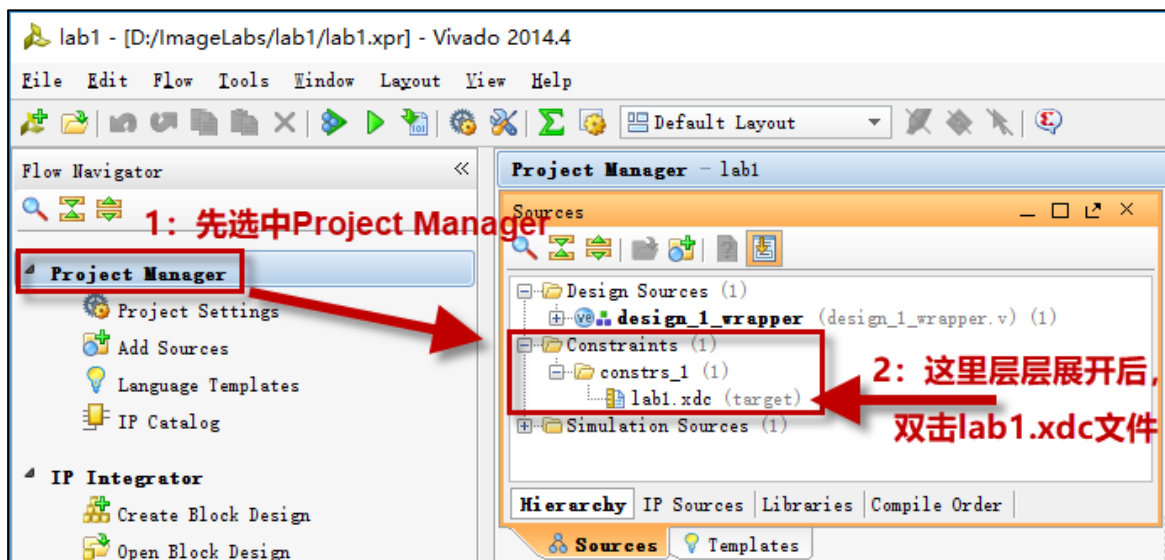


图 2-45 打开 lab1.xdc 文件

接着在相对应的文本编辑区域，首先确认打开的文件是 lab1.xdc，接着在该文件的最后一行，加上如下语句：

```
set_property CLOCK_DEDICATED_ROUTE BACKBONE [get_nets
design_1_i/clk_wiz_0/inst/clk_in1_design_1_clk_wiz_0_0]
```

添加后点击窗口左边的保存图标，保存修改。整个过程如下图所示：



图 2-46 修改 lab1.xdc 文件

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	35 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

35. 至此，整个实验的设计部分全部结束，回到 Vivado 的主界面，在左边栏点击 Program and Debug 项下的 Generate Bitstream，来生成 bit 文件。

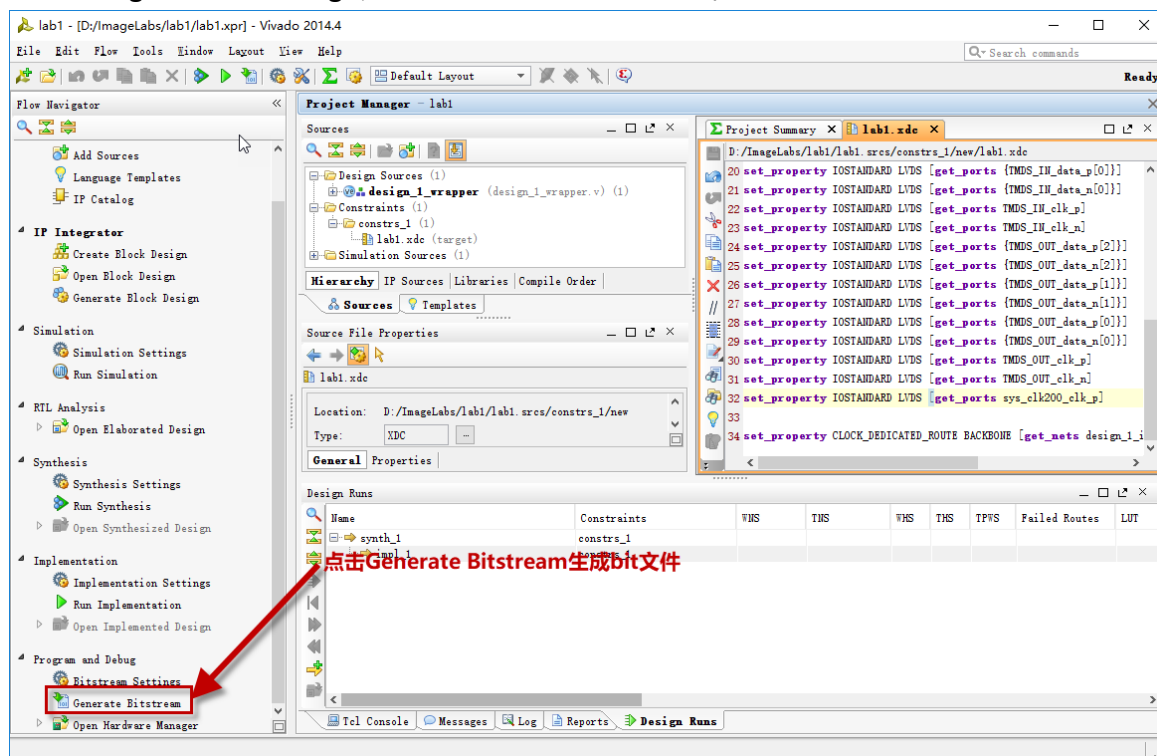


图 2-47 Generate Bitstream

在弹出的窗口，点击 Yes，如下图所示：

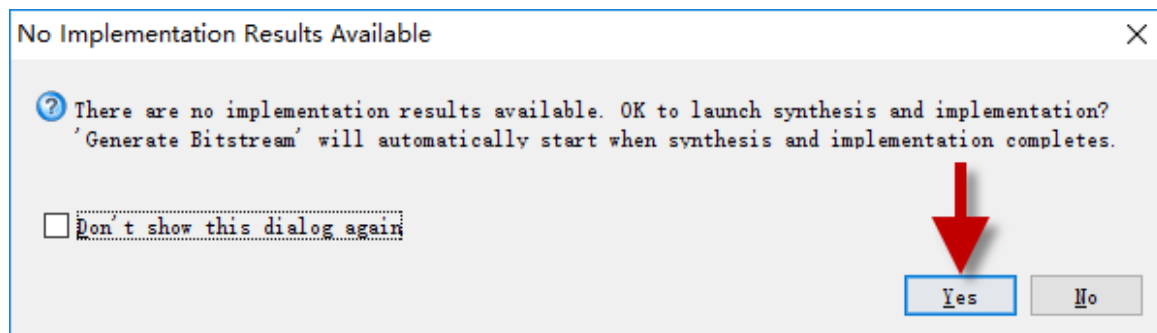


图 2-48 点击 Yes 确认生成 bit 文件

经过大约 10~15 分钟后，Bitstream 生成的提示窗口会自动弹出，表示 Vivado 成功地生成了 bit 文件，这时选择 Open Hardware Manager，并点击 OK，准备将 bit 文件下载到 SWORD4.0 硬件平台。

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	36 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

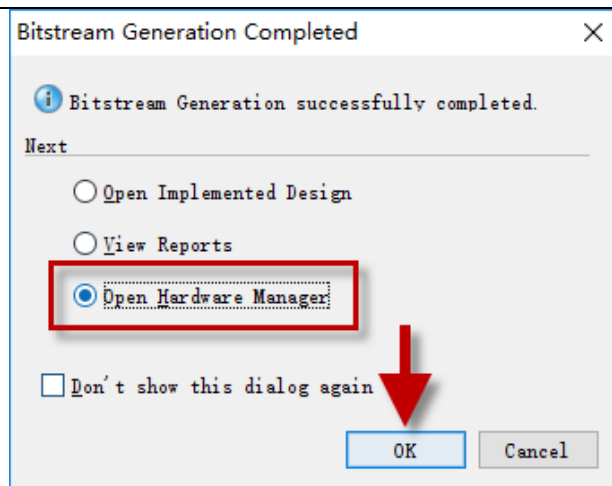


图 2-49 打开 Hardware Manager

接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关

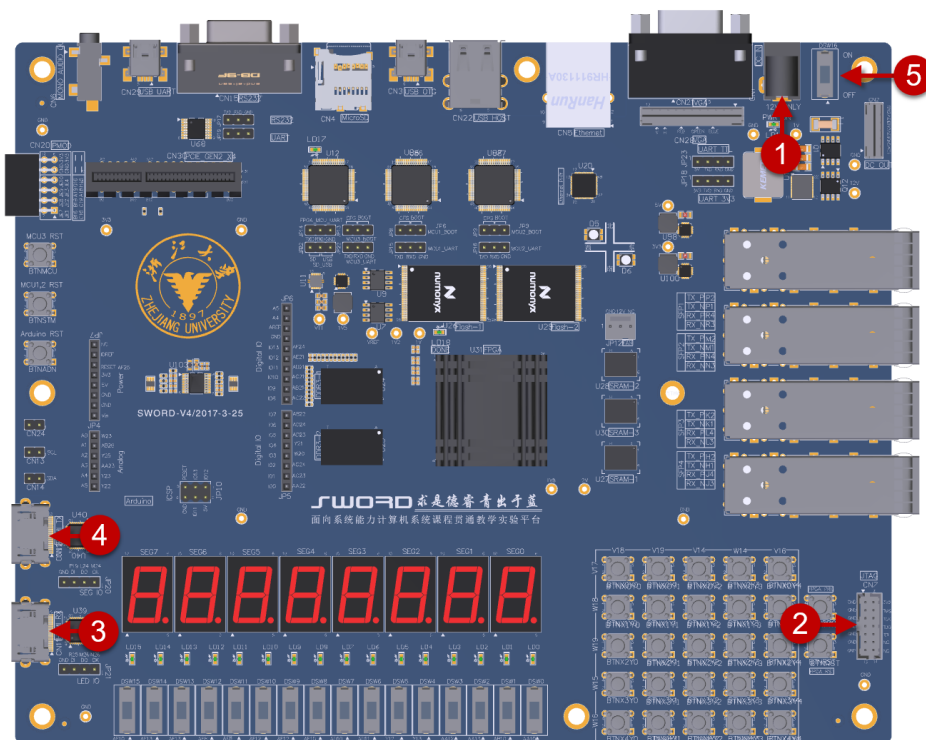


图 2-50 硬件连接对应位置

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	37 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

连接好后的效果如下图所示：



图 2-51 实际硬件连接

36 . 接着在 Hardware Manager 界面下，点击 Open target，在随之弹出的菜单中选择 Auto Connect，整个过程如下图所示：

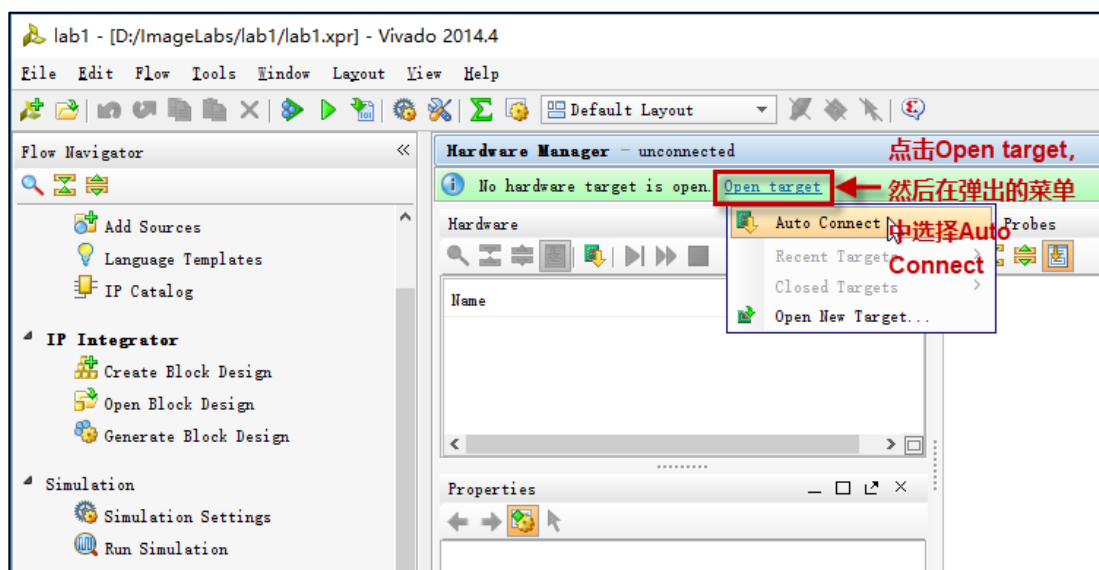


图 2-52 Open target

接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	38 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

3. 实验结果

此时我们可以将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次，以便让信号源设备重新检测 (Detect) 一下接收设备，一切正常的话，我们即可在 HDMI 显示器上看到显示画面。

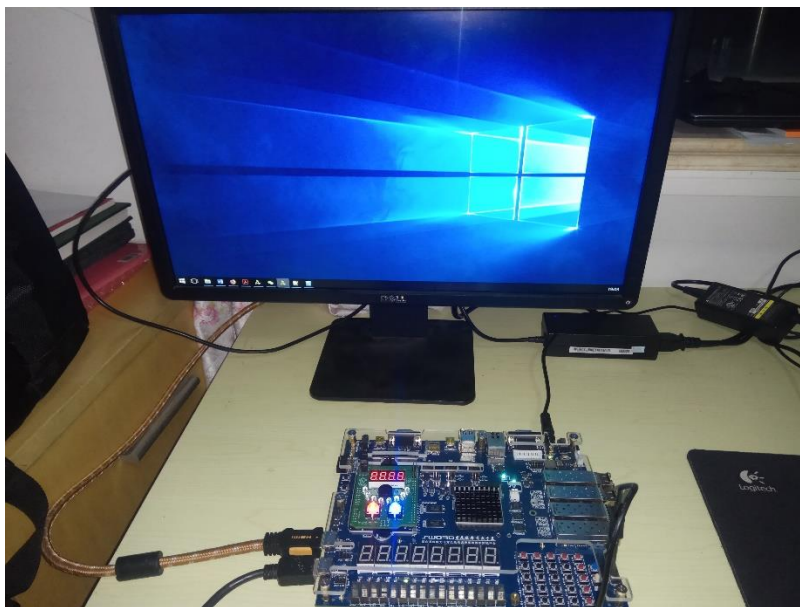


图 3-1 正常显示结果

3.1 现象

如下图所示，我们可以看到显示器当前的分辨率为 1600x900, 60Hz，这个和 DVI2RGB IP 里的参数设置是符合的。在我们没有修改 HDMI 信号源输出的情况下，HDMI 信号源自动读取 SWORD4.0 模拟的 EDID ROM，将输出分辨率设定为了 1600x900。



图 3-2 显示器的显示分辨率

	标题	文档编号	版本	页
	Lab1: 基本视频 IO 实验	XD-LAB-IMG-001	1.2	40 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		