



XD-LAB-ARCH-006

Lab6: Pipelined CPU supporting interrupt

Joseph Xu

2019-1-23

修改记录

版本号.	作者	描述	修改日期
1.0	Joseph Xu	初稿	2019-01-23

审核记录

姓名	职务	签字	日期

XINGDENG	标题	文档编号	版本	页
	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	1 of 34
	作者	修改日期		
	Joseph Xu	2019/1/23		公开

目录

修改记录	0
审核记录	1
1. 实验准备	5
1. 1 实验目的	5
1. 2 实验内容	5
1. 3 实验资料	5
2. 实验步骤	7
2. 1 新建工程	7
2. 1 修改文件	9
2. 2 功能仿真	22
2. 3 生成 bitstream 文件	29
2. 4 下载验证	30

标题	文档编号	版本	页
XINGDENG Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	2 of 34
作者 Joseph Xu	修改日期 2019/1/23	公开	

图目录

图 1-1 扩充后的 3 条中断指令	5
图 1-2 实验资料网站	5
图 1-3 实验文档和工程源文件	6
图 2-1 运行 Vivado	7
图 2-2 打开工程	7
图 2-3 打开 Lab05 工程	8
图 2-4 工程另存为	8
图 2-5 工程另存为 lab06	9
图 2-6 需要修改的 lab06 工程	9
图 2-7 点击 Expand All 快捷按钮	10
图 2-8 mips_top 展开后的文件层次	10
图 2-9 移除 controller.v 文件	11
图 2-10 确认移除文件	11
图 2-11 移除 CONTROLLER 后的视图	11
图 2-12 移除所有旧文件后的源码层次	12
图 2-13 添加新文件	12
图 2-14 增加 design sources	13
图 2-15 add files	13
图 2-16 切换目录	14
图 2-17 选择文件	14
图 2-18 lab06 添加的文件	15
图 2-19 继续添加 hex 文件	15
图 2-20 Next	15
图 2-21 添加文件	16
图 2-22 指定全部类型文件可见	16
图 2-23 切换目录	17
图 2-24 选择 hex 文件	17
图 2-25 增加到工程	18
图 2-26 设置文件类型	18
图 2-27 设置类型为 Data Files	19
图 2-28 设置文件类型	19

XINGDENG	标题	文档编号	版本	页
	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	3 of 34
作者	修改日期			
	Joseph Xu	2019/1/23	公开	

图 2-29 设置类型为 Data Files	20
图 2-30 Refresh Hierarchy	20
图 2-31 Data Files	20
图 2-32 添加所有文件后的文件列表.....	21
图 2-33 设置仿真顶层.....	22
图 2-34 开始仿真.....	22
图 2-35 仿真视图.....	23
图 2-36 仿真所需要观察的所有信号.....	24
图 2-37 菜单中选择“Open Waveform Configuration”.....	25
图 2-38 切换目录.....	25
图 2-39 选择 wcfg 文件.....	25
图 2-40 加载波形配置文件后的波形列表视图	26
图 2-41 Restart Simulation.....	26
图 2-42 Run for...	27
图 2-43 仿真结果 1.....	27
图 2-44 仿真结果 2.....	28
图 2-45 关闭仿真视图.....	28
图 2-47 生成 bitstream.....	29
图 2-48 自动运行中间步骤.....	29
图 2-49 生成 bitstream 成功	30
图 2-50 连接 SWORD4.0.....	30
图 2-51 主要控制 IO	31
图 2-52 进入 Hardware Manager	31
图 2-53 自动连接到 SWORD4.0	32
图 2-54 Program Device	32
图 2-55 下载 bit 文件	33
图 2-56 运行效果.....	33

XINGDENG	标题	文档编号	版本	页
	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	4 of 34
作者	修改日期			
Joseph Xu	2019/1/23			公开

1. 实验准备

1.1 实验目的

在实验 5 的基础上，增加了协处理器 CPO，支持中断指令及中断相关机制。增加的指令如下图所示：

MIPS Instructions							
Bit #	31..26	25..21	20..16	15..11	10..3	2..0	Operations
mfc	010000	00000	rt	rd	0000 0000	sel	GPR[rt] = CPR[0, rd, sel]
mtc	010000	00100	rt	rd	0000 0000	sel	CPR[0, rd, sel] = GPR[rt]
Bit #	31..26	25	24..6		2..0	Operations	
eret	010000	1	000 0000 0000 0000 0000		0000	PC <= EPC	

图 1-1 扩充后的 3 条中断指令

1.2 实验内容

本实验内容包括：

- 设计协处理器 CPO；
- 修改设计 CPU 的数据通路 Datapath 和控制单元 Controller；
- 对设计的 CPU 进行行为仿真，检查功能正确性；
- 上板验证并观察 CPU 的执行过程。

1.3 实验资料

实验资料可以从网站 <https://gitee.com/SWORDfpga/ComputerArchitecture> 获得。

图 1-2 实验资料网站

XINGDENG	标题	文档编号	版本	页
	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	5 of 34
作者	Joseph Xu	修改日期	公开	
		2019/1/23		

为方便使用，建议建立 c:\archlabs 目录，放置实验资料和实验工程。

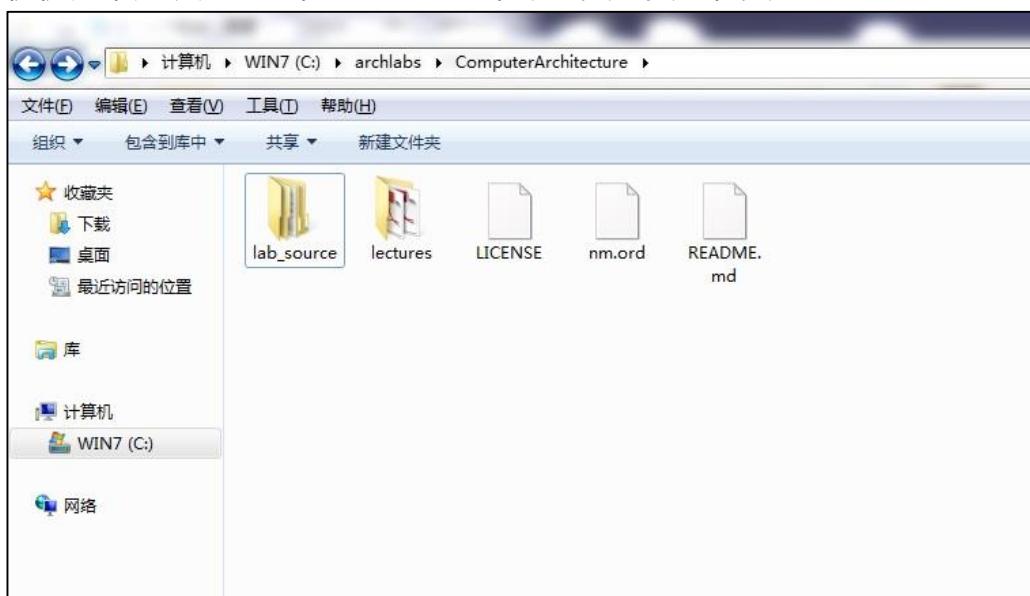


图 1-3 实验文档和工程源文件

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	6 of 34
作者 Joseph Xu	修改日期 2019/1/23	公开	

2. 实验步骤

2.1 新建工程

首先启动 Vivado 2014.4 开发工具，如图 2-1 所示：



图 2-1 运行 Vivado

点击 Open Project，如图 2-2 所示：

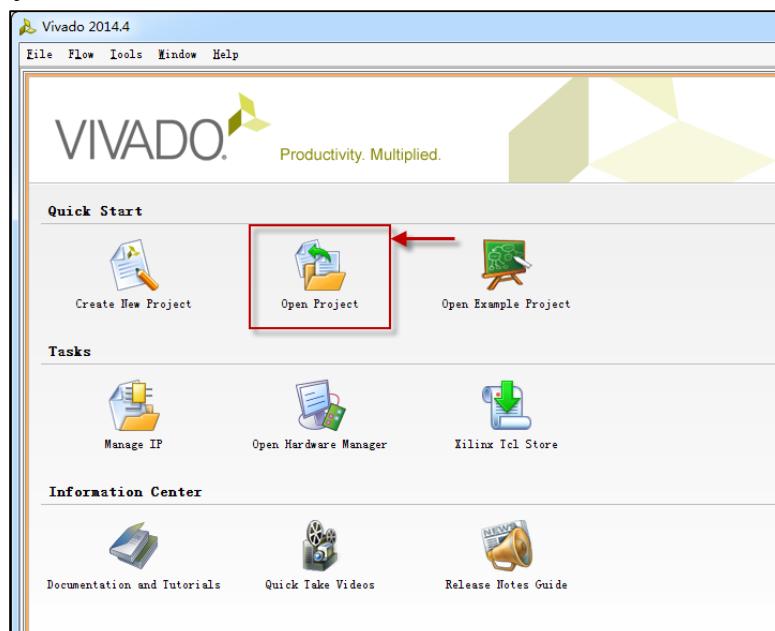


图 2-2 打开工程

 XINGDENG	标题 Lab6: Pipelined CPU supporting interrupt	文档编号 XD-LAB-ARCH-006	版本 1.0	页 7 of 34
	作者 Joseph Xu	修改日期 2019/1/23		公开

选择之前已经完成的 Lab05 工程，点击 OK 打开工程，如图 2-3 所示：

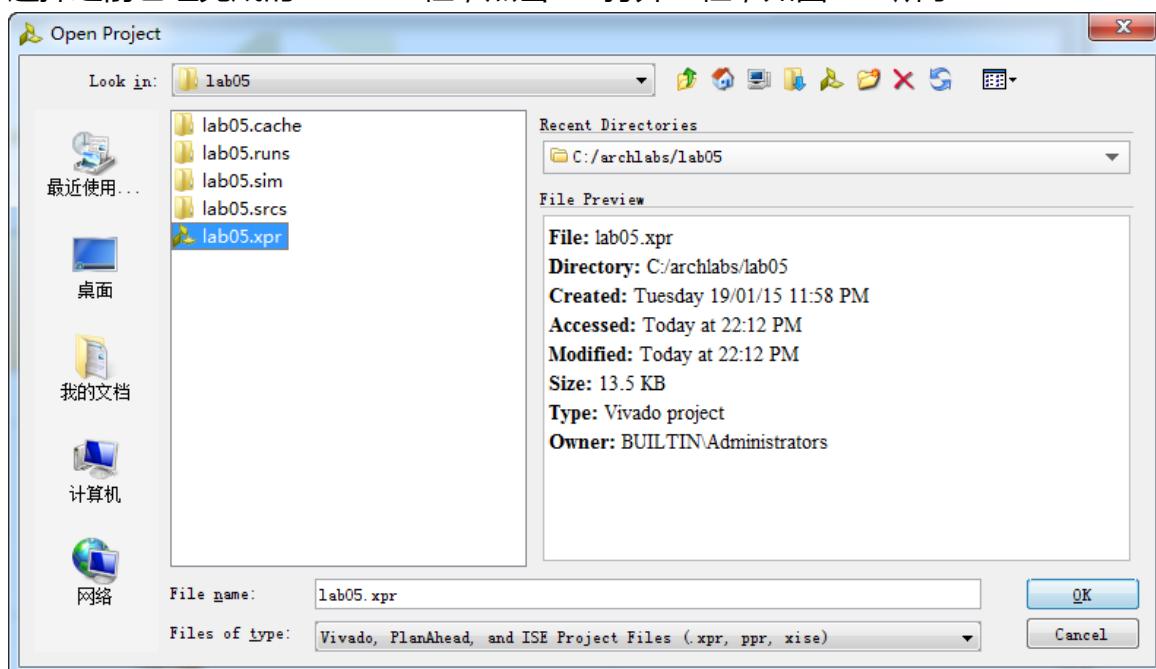


图 2-3 打开 Lab05 工程

如图 2-4 所示，本实验以 Lab05 工程为基础，在 File 菜单点击 Save Project As 将工程另存为。

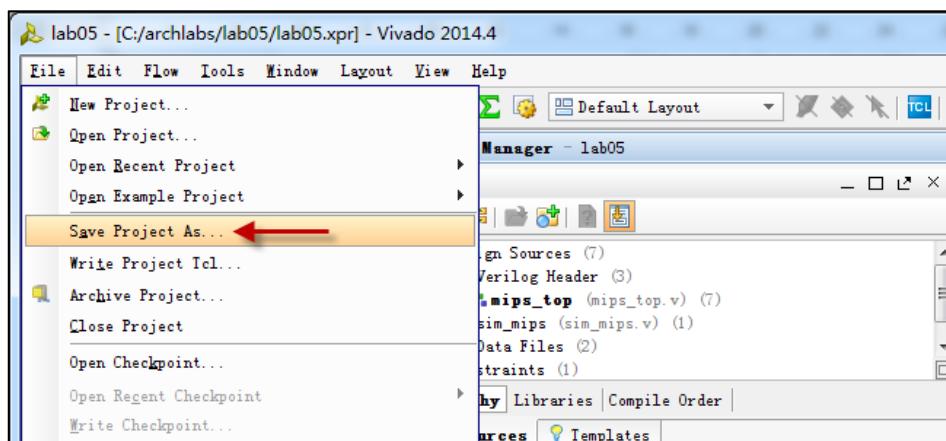


图 2-4 工程另存为

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	8 of 34
作者	修改日期		
Joseph Xu	2019/1/23		公开

输入新工程名 lab06 , 选中 Create project subdirectory , 点击 OK 继续。

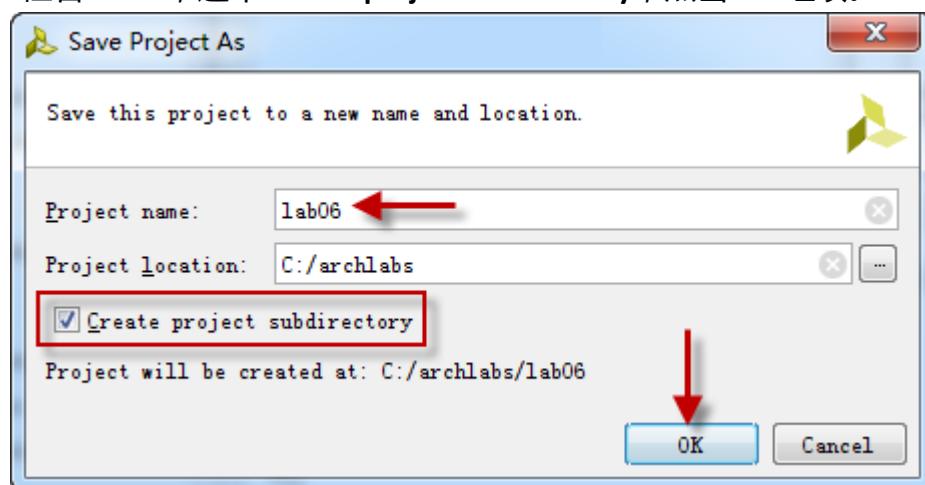


图 2-5 工程另存为 lab06

如图 2-6 所示 , 工程被另存为 lab06 , 本实验将以此工程为基础进行修改。

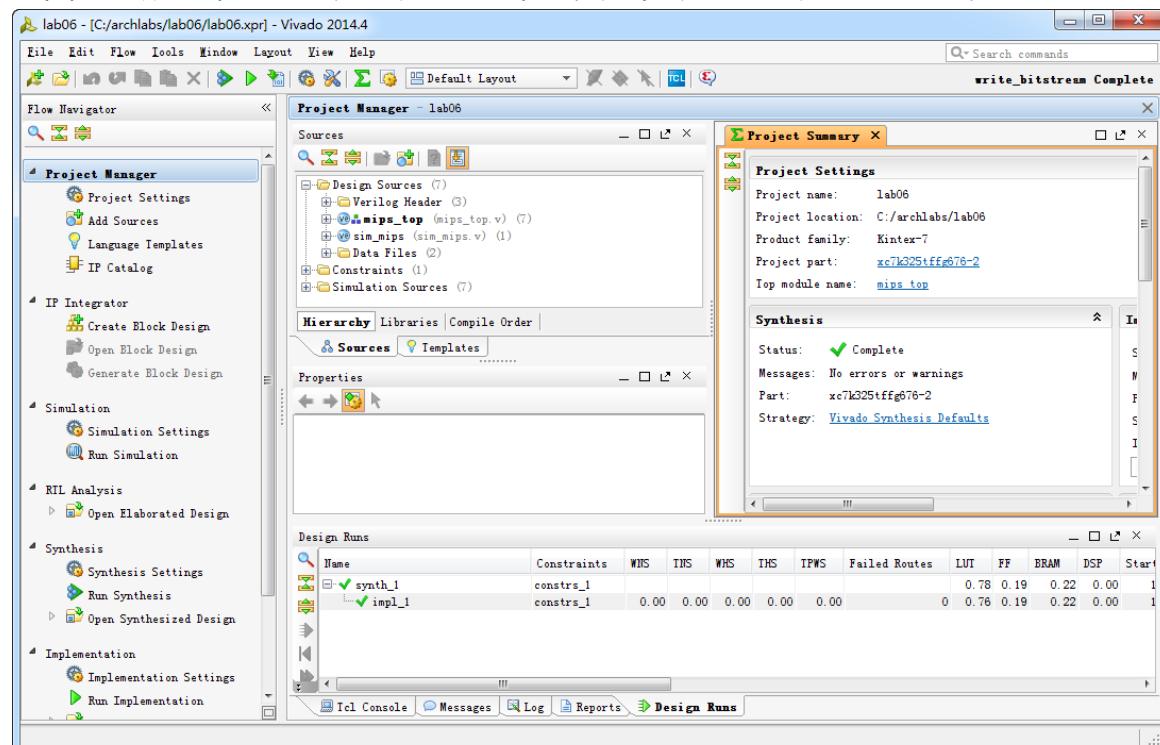


图 2-6 需要修改的 lab06 工程

2.1 修改文件

在 Project Manager 窗口下 , 首先点击 mips_top , 然后点击如图 2-7 所示的 Expand All 快捷按钮展开 mips_top 下的所有文件 :

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	9 of 34
作者	修改日期		
Joseph Xu	2019/1/23		公开

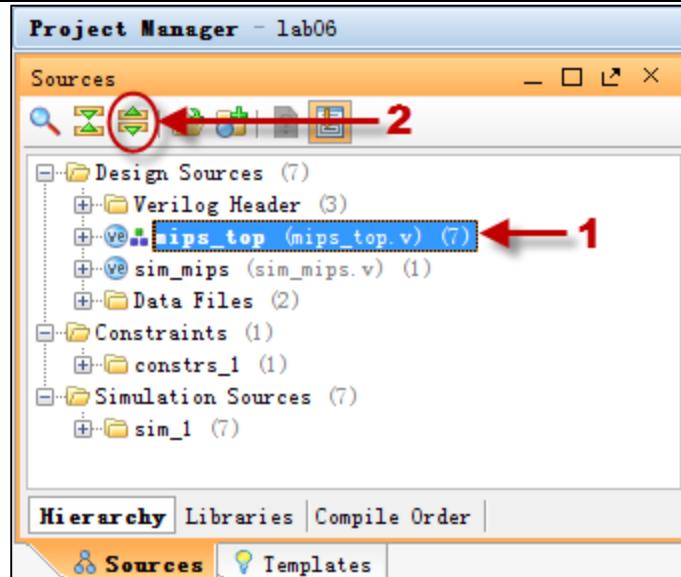


图 2-7 点击 Expand All 快捷按钮

展开后的层次如所示：

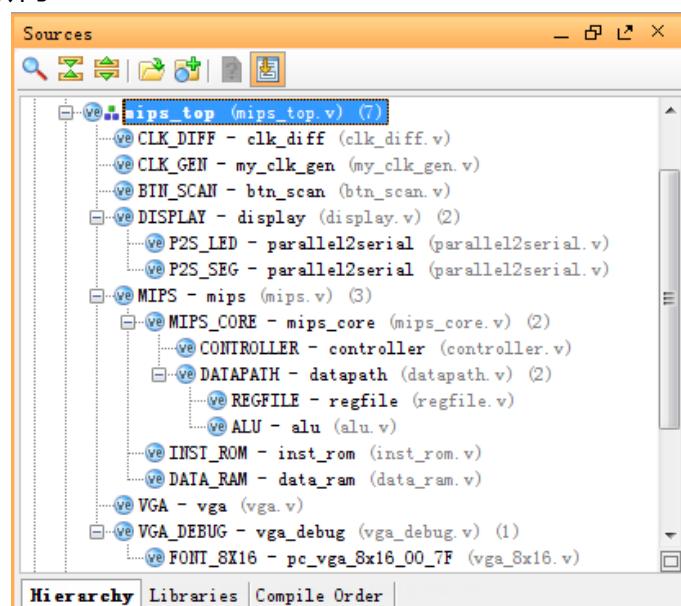


图 2-8 mips_top 展开后的文件层次

由于本实验在 lab05 的基础上需要修改和新增一些文件，所以我们进行如下一些移除旧文件，添加更新文件的操作：

首先移除 controller.v 文件：右键点击 **CONTROLLER**，点击 **Remove File from Project** 继续，如图 2-9 所示：

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	10 of 34
作者 Joseph Xu	修改日期 2019/1/23	公开	

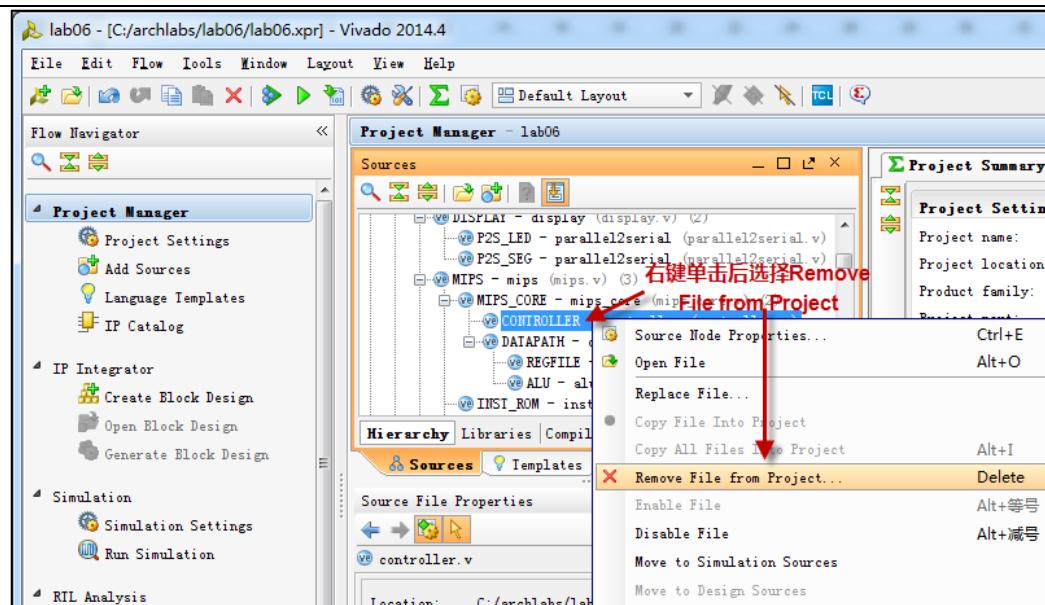


图 2-9 移除 controller.v 文件

选中 **Also delete the project local file/directory form disk**，点击 **OK** 继续，如图 2-10 所示：

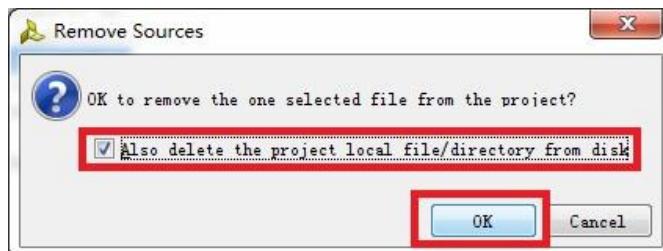


图 2-10 确认移除文件

移除后的视图如图 2-11 所示：

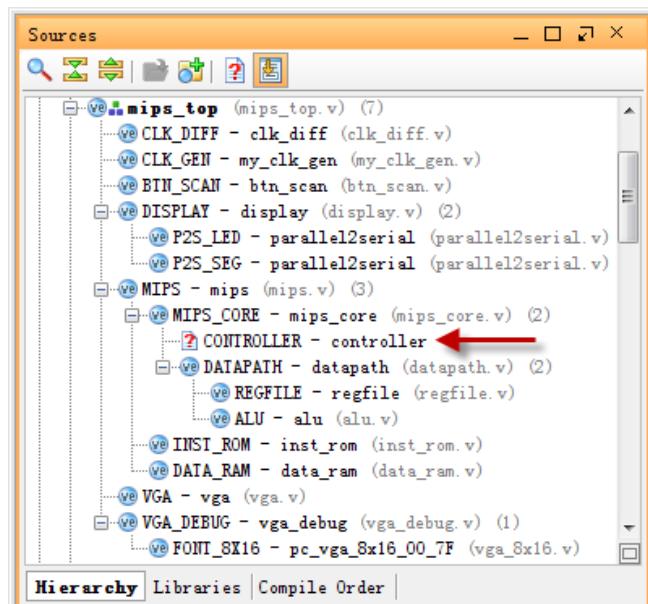


图 2-11 移除 CONTROLLER 后的视图

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	11 of 34
作者 Joseph Xu	修改日期 2019/1/23		公开

接着按照上面的移除文件方式，将下列文件依次移除：

- 1) inst_mem.hex
- 2) data_mem.hex
- 3) mips.v
- 4) alu.v
- 5) datapath.v
- 6) mips_core.v
- 7) mips_define.vh
- 8) sim_mips.v

移除后的文件列表应如图 2-12 所示，请仔细比对该图，确保所有旧文件都已经移除！！

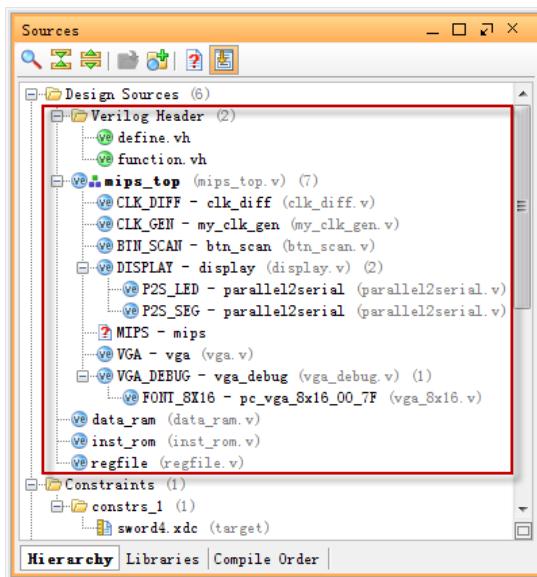


图 2-12 移除所有旧文件后的源码层次

接着在 Sources 窗口内点击 Add Sources 快捷按钮，添加新文件，如图 2-13 所示。

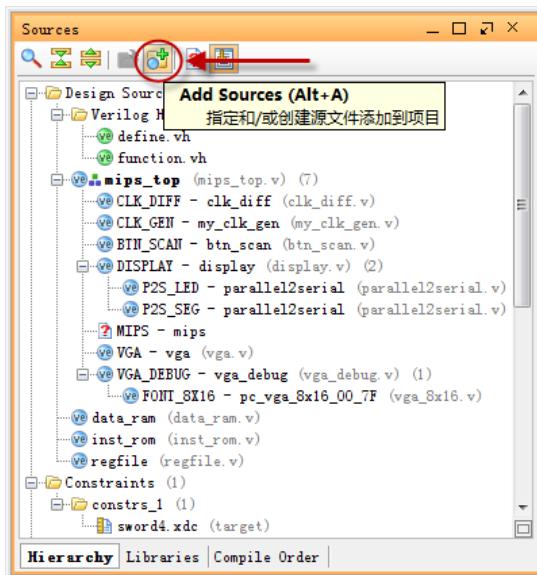


图 2-13 添加新文件

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	12 of 34
作者	修改日期		
Joseph Xu	2019/1/23	公开	

选中 Add or create design sources , 点击 Next , 如图 2-14 所示 :



图 2-14 增加 design sources

点击 Add Files , 如图 2-15 所示 :

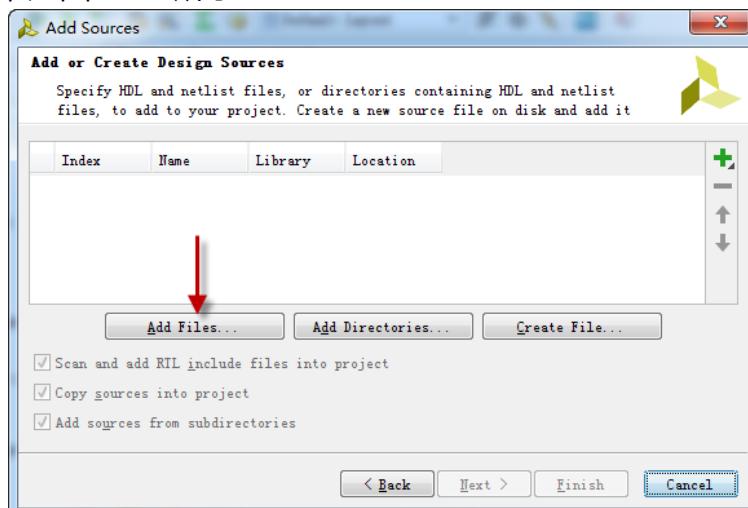


图 2-15 add files

进入 C:\archlabs\ComputerArchitecture\lab_source\lab06\sources\cpu 目录 , 如图 2-16 所示 :

标题	文档编号	版本 页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0 13 of 34
作者	修改日期	
Joseph Xu	2019/1/23	公开

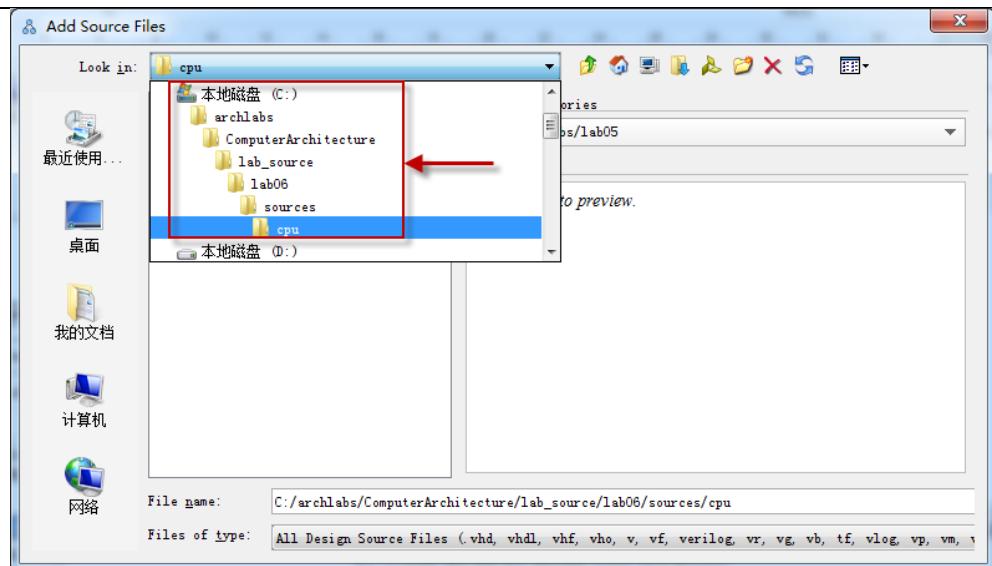


图 2-16 切换目录

选中所有文件，点击回车确认添加，如图 2-17 所示。

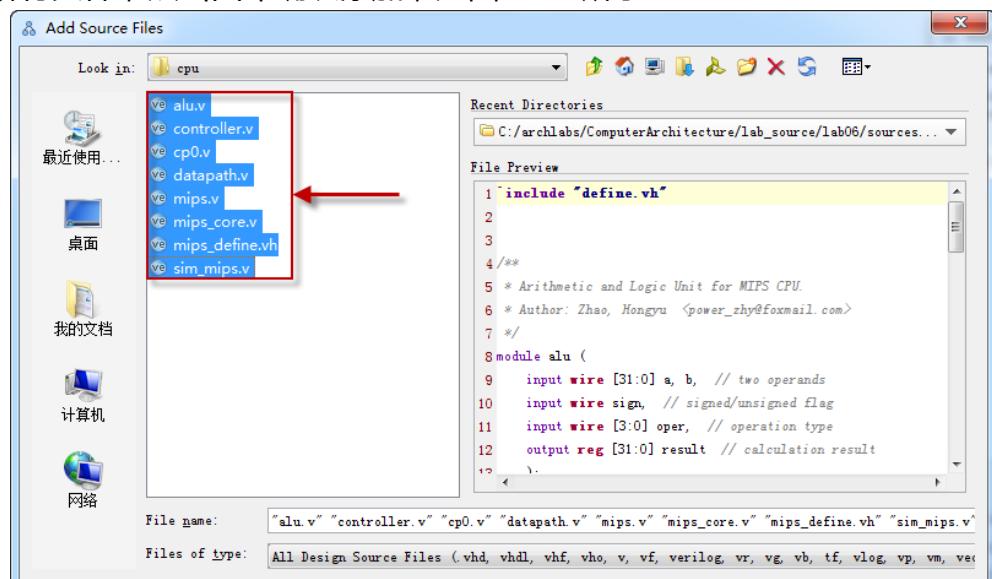


图 2-17 选择文件

如图 2-18 所示，列表中会显示要添加到工程中的文件，本次操作添加了 7 个文件，请确认添加文件的数量正确。选中 Scan and add RTL include files into project 和 Copy sources into project，点击 Finish。

本次操作新加入了一下文件：

协处理器文件 cp0.v 用来实现对中断指令的支持；

增加的 alu.v、controller.v 和 datapath.v 修改了对 CP0 的响应控制和数据通路；

mips.v 和 mips_core.v 作为连接 controller 和 datapath 的模块也做了对应修改；

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	14 of 34
作者 Joseph Xu	修改日期 2019/1/23		公开

mips_define.vh 里增加了对 mtc、mfc 和 eret 指令的定义；

sim_mips.v 里增加了用于仿真中断的激励信号。

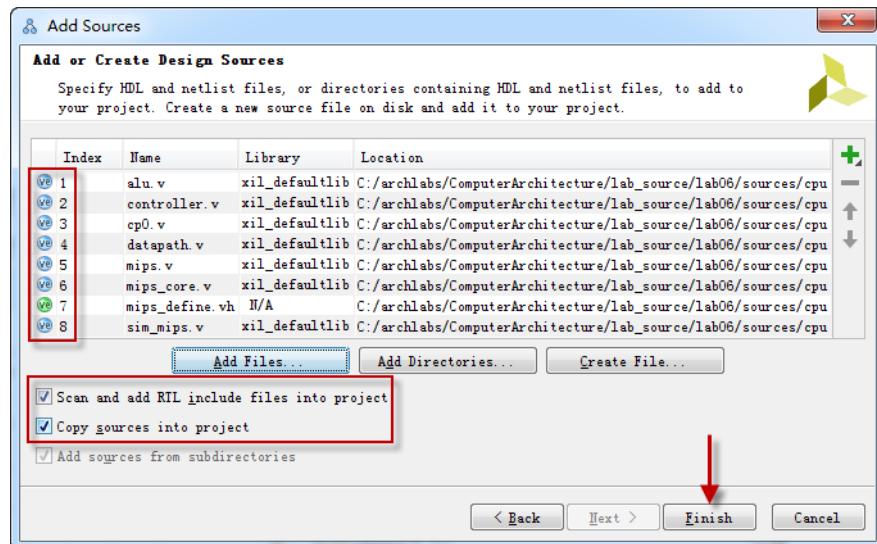


图 2-18 lab6 添加的文件

接着我们添加 hex 文件，同样点击如图 2-19 所示的 Add Sources 快捷按钮：

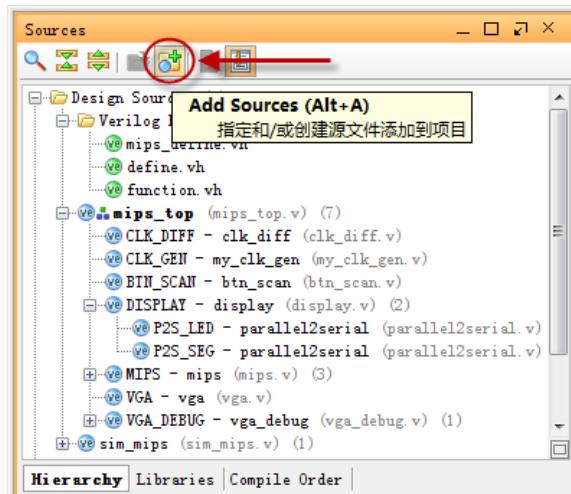


图 2-19 继续添加 hex 文件

选中 Add or create design sources，点击 Next，如图 2-20 所示：

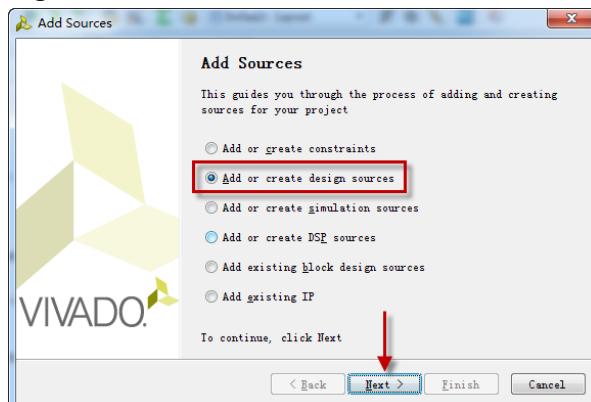


图 2-20 Next

标题	文档编号	版本
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0 15 of 34
作者	修改日期	页
Joseph Xu	2019/1/23	公开

用 Add Files 添加文件，如图 2-21 所示：

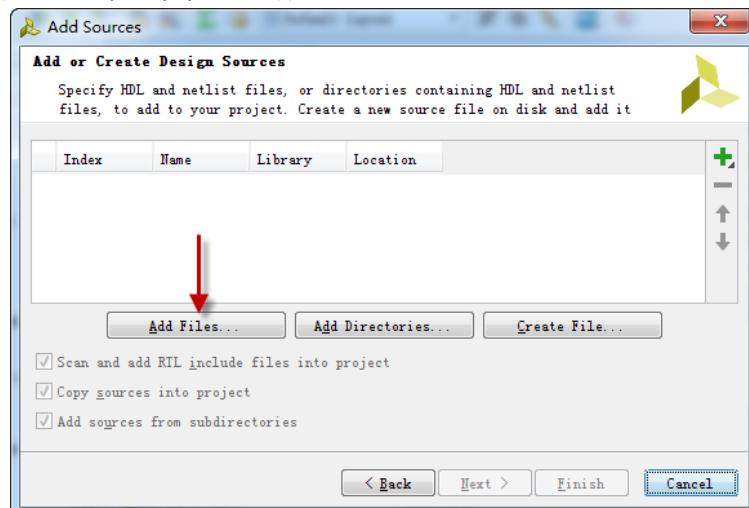


图 2-21 添加文件

指定文件类型为 All Files，如图 2-22 所示：

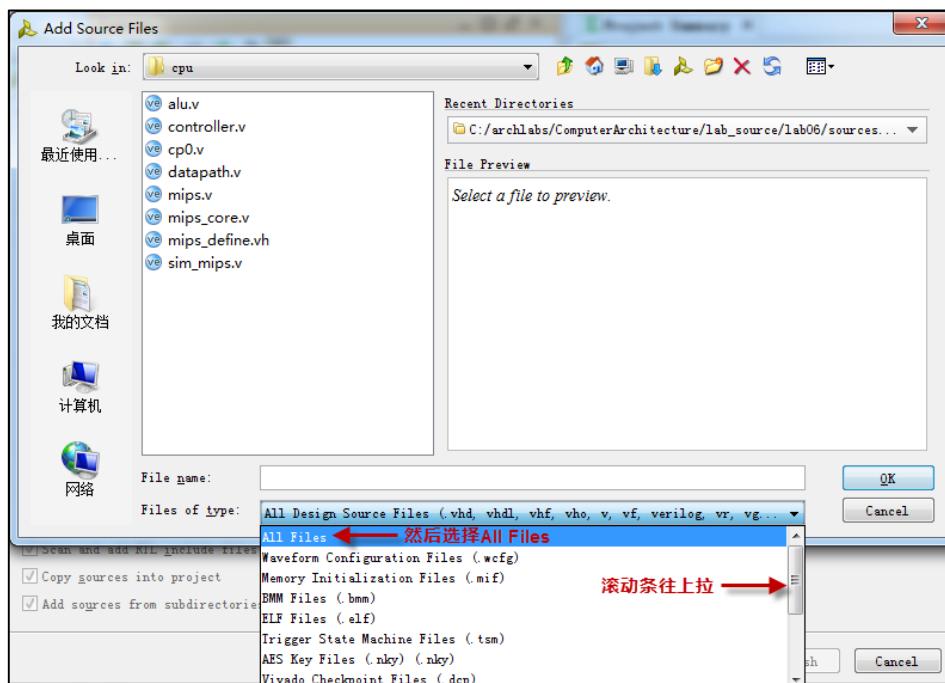


图 2-22 指定全部类型文件可见

切换到 C:\archlabs\ComputerArchitecture\lab_source\lab06\data 目录，如图 2-23 所示：

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	16 of 34
作者	修改日期		
Joseph Xu	2019/1/23		公开

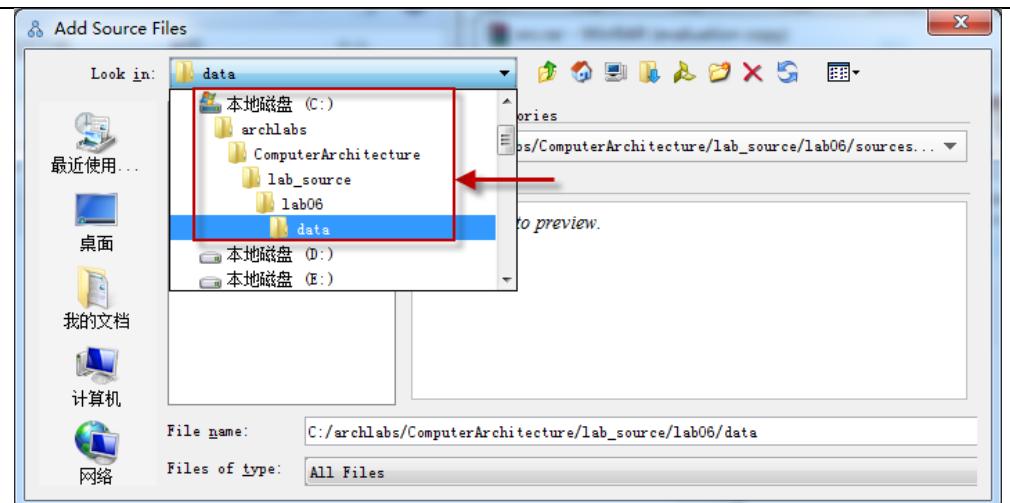


图 2-23 切换目录

选择 data_mem.hex 和 inst_mem.hex 文件，如图 2-24 所示：

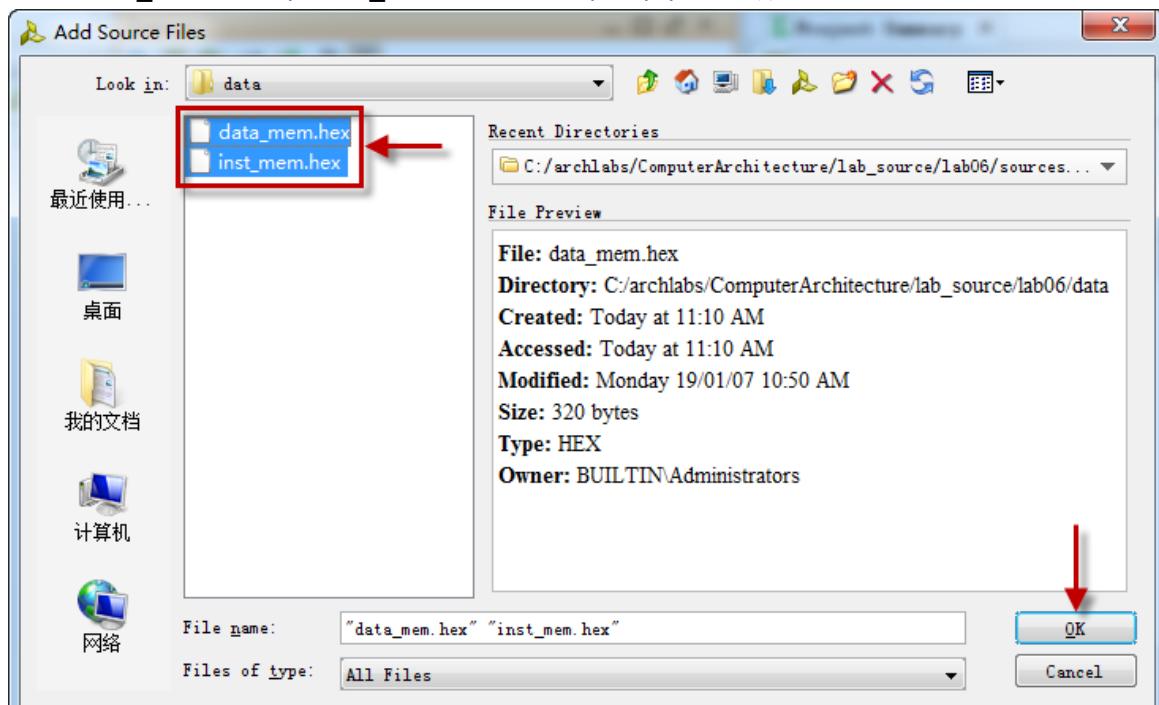


图 2-24 选择 hex 文件

选中 Scan and add RTL include files into project 和 Copy sources into project，点击 Finish 把文件加入到工程，如图 2-25 所示：

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	17 of 34
作者	修改日期		
Joseph Xu	2019/1/23	公开	

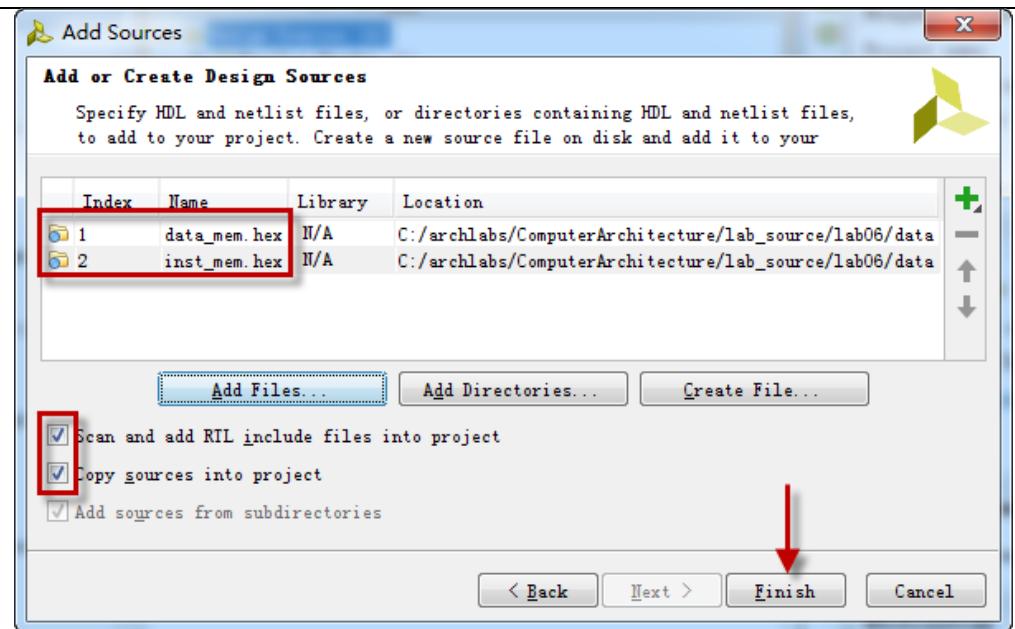


图 2-25 增加到工程

点击“+”展开 Unknown，在 data_mem.hex 上点击右键，点击 Set File Type，如图 2-26 所示：

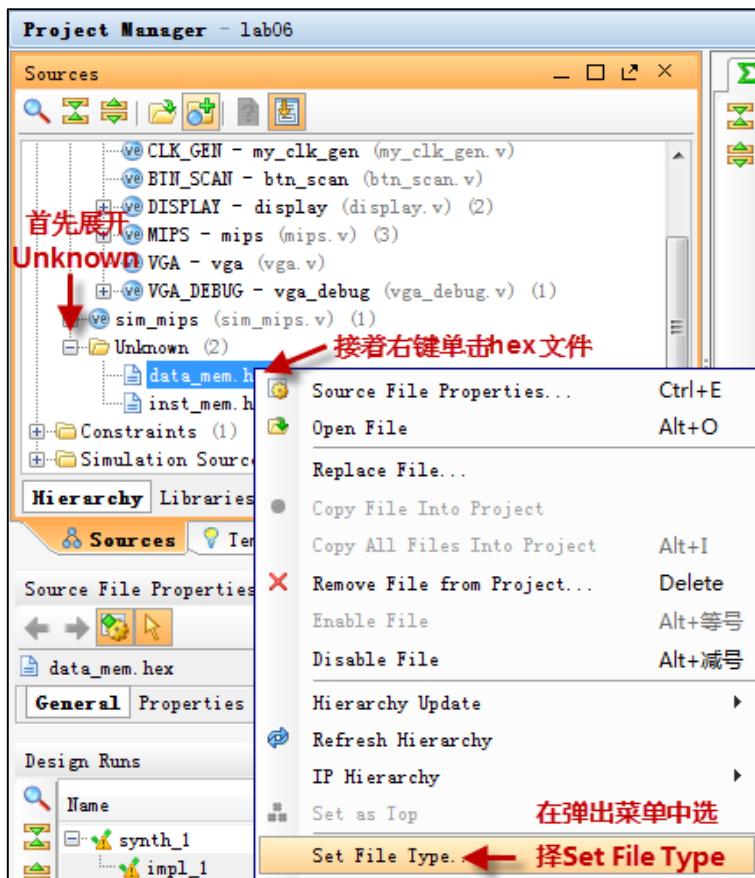


图 2-26 设置文件类型

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	18 of 34
作者	修改日期		
Joseph Xu	2019/1/23		公开

选择 Data Files , 点击 OK , 如图 2-27 所示 :

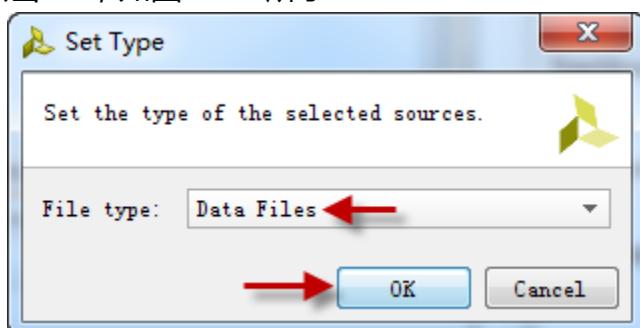


图 2-27 设置类型为 Data Files

如图 2-28 所示 , 同样设置 inst_mem.hex 的类型。

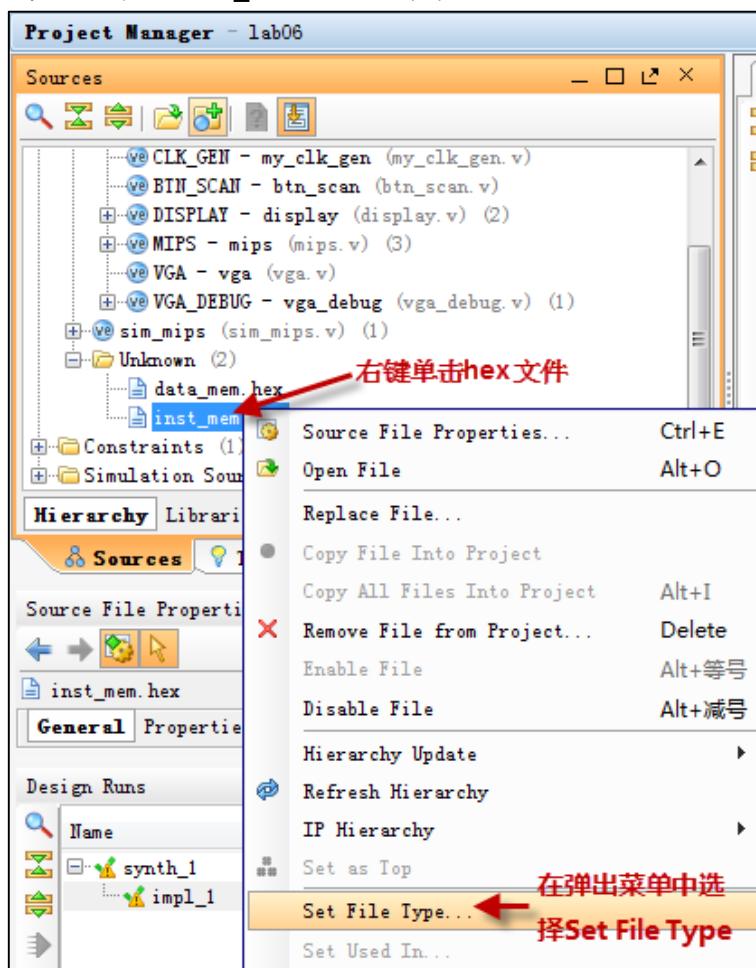


图 2-28 设置文件类型

设置为 Data Files , 如图 2-29 所示 :

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	19 of 34
作者	修改日期		
Joseph Xu	2019/1/23	公开	

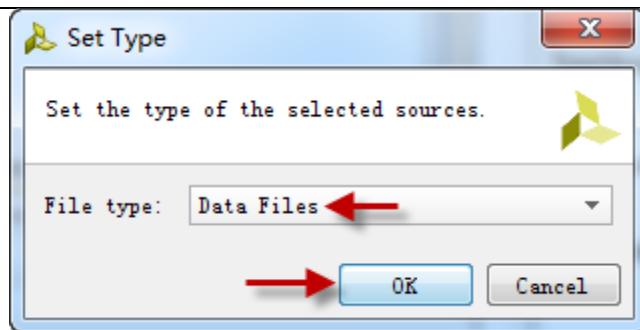


图 2-29 设置类型为 Data Files

在 data_mem.hex 上点击右键，点击 Refresh Hierarchy 更新视图，如图 2-30 所示：

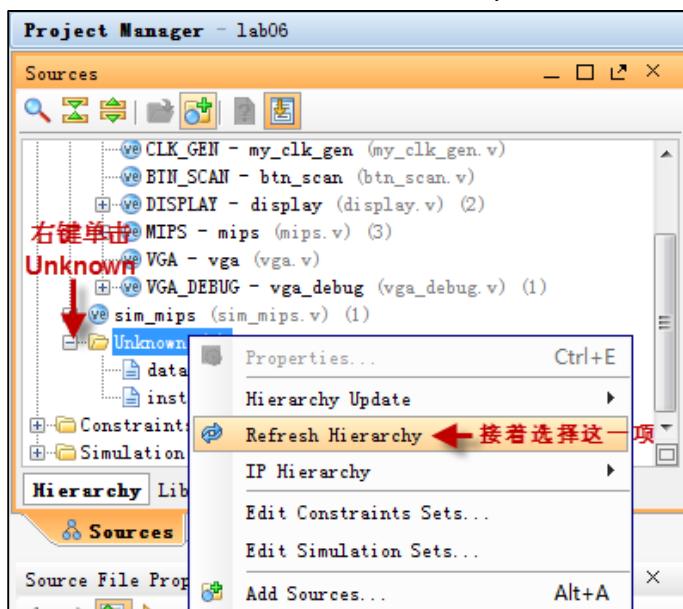


图 2-30 Refresh Hierarchy

如图 2-31 所示，现在 data_mem.hex 和 inst_mem.hex 都在 Data Files 下了，这两个文件包含指令存储器和数据存储器的初始化信息。

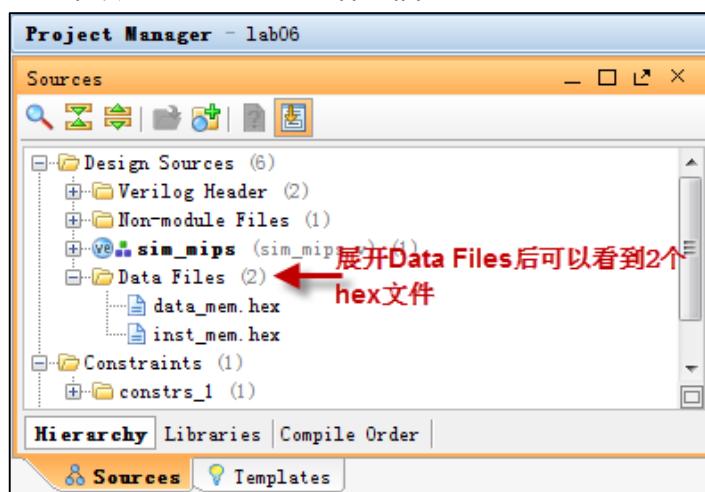


图 2-31 Data Files

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	20 of 34
作者	修改日期		
Joseph Xu	2019/1/23		公开

添加的文件列表应如图 2-32 所示，**请仔细比对该图，确保所有新文件都已经添加 !!**

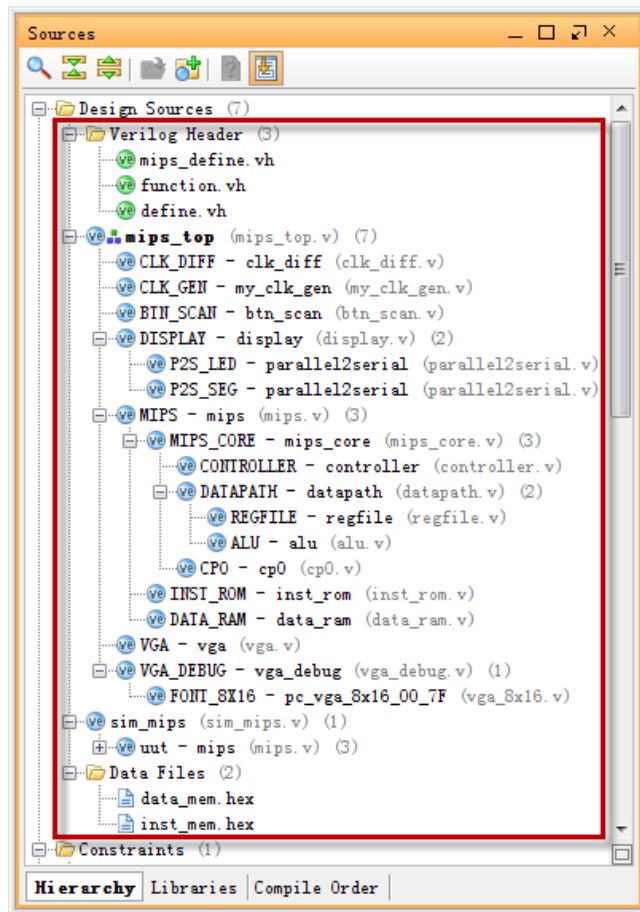


图 2-32 添加所有文件后的文件列表

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt XINGDENG	XD-LAB-ARCH-006	1.0	21 of 34
作者 Joseph Xu	修改日期 2019/1/23	公开	

2.2 功能仿真

如图 2-33 所示，在 Vivado 的 Sources 窗口的 Simulation Source 子项依次展开，然后将 sim_mips 设置为仿真顶层：

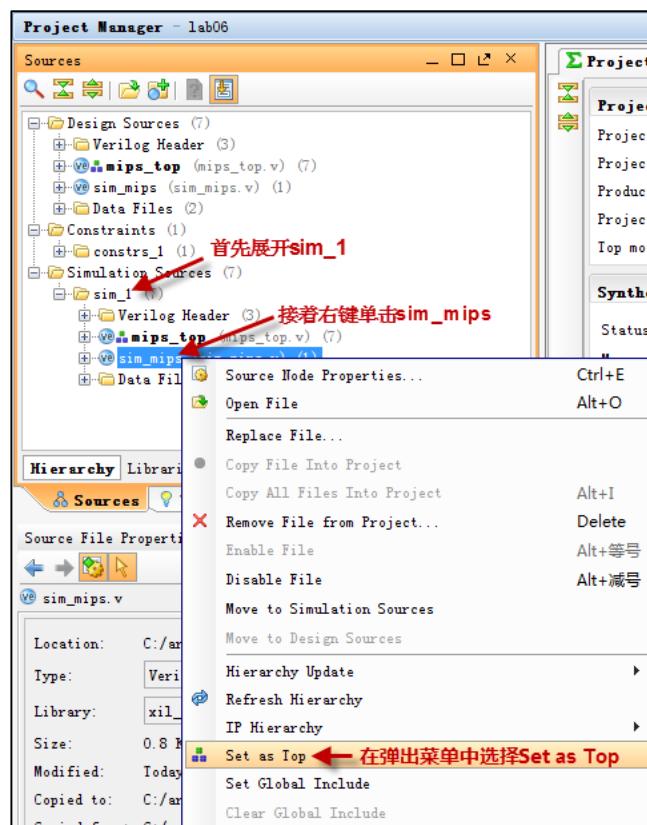


图 2-33 设置仿真顶层

点击 Run Simulation 下的 Run Behavioral Simulation，如图 2-34 所示：

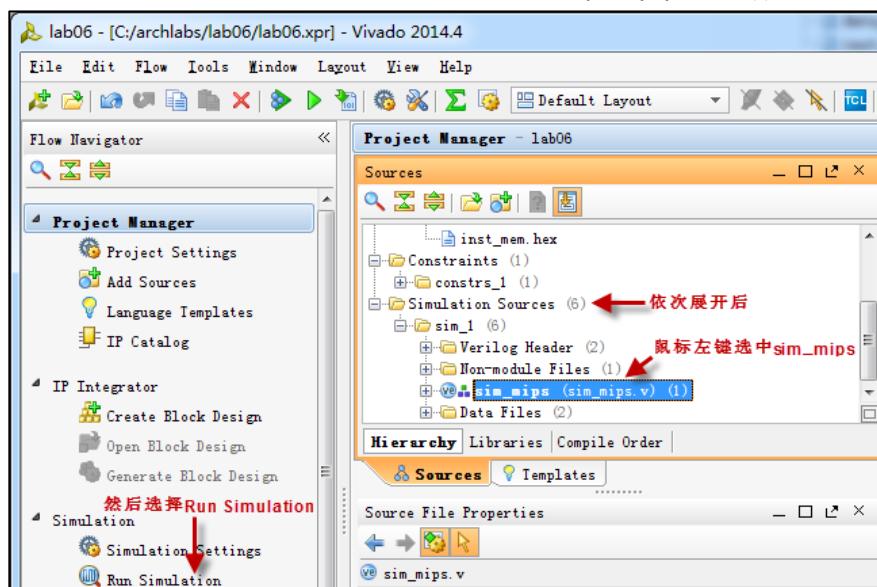


图 2-34 开始仿真

标题	文档编号	版本
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0 22 of 34
作者	修改日期	
Joseph Xu	2019/1/23	公开

仿真编译成功后，进入仿真视图，如图 2-35 所示：

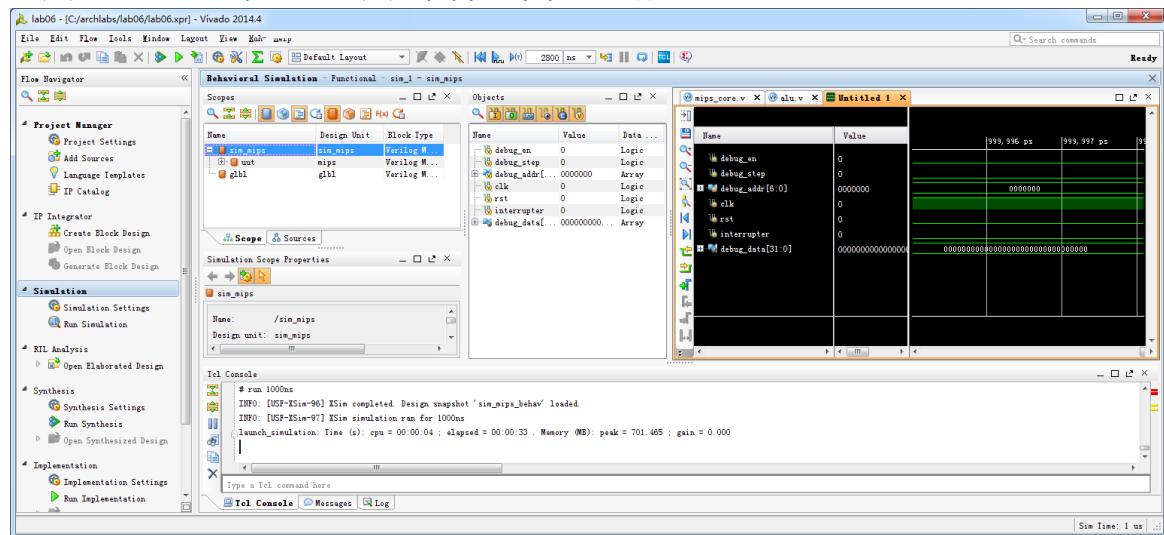


图 2-35 仿真视图

按照之前实验的仿真操作，将如下信号添加到波形列表中：

1. /sim_mips/clk
2. /sim_mips/rst
3. /sim_mips/uut/MIPS_CORE/DATAPATH/if_rst
4. /sim_mips/uut/MIPS_CORE/DATAPATH/if_valid
5. /sim_mips/uut/MIPS_CORE/DATAPATH/inst_addr[31:0]
6. /sim_mips/uut/MIPS_CORE/DATAPATH/inst_data[31:0]
7. /sim_mips/uut/MIPS_CORE/DATAPATH/inst_ren
8. /sim_mips/uut/MIPS_CORE/DATAPATH/id_rst
9. /sim_mips/uut/MIPS_CORE/DATAPATH/id_valid
10. /sim_mips/uut/MIPS_CORE/DATAPATH/inst_addr_id[31:0]
11. /sim_mips/uut/MIPS_CORE/DATAPATH/inst_data_id[31:0]
12. /sim_mips/uut/MIPS_CORE/DATAPATH/exe_rst
13. /sim_mips/uut/MIPS_CORE/DATAPATH/exe_valid
14. /sim_mips/uut/MIPS_CORE/DATAPATH/inst_addr_exe[31:0]
15. /sim_mips/uut/MIPS_CORE/DATAPATH/inst_data_exe[31:0]
16. /sim_mips/uut/MIPS_CORE/DATAPATH/fwd_a_ctrl[1:0]
17. /sim_mips/uut/MIPS_CORE/DATAPATH/fwd_b_ctrl[1:0]
18. /sim_mips/uut/MIPS_CORE/CONTROLLER/reg_stall
19. /sim_mips/uut/MIPS_CORE/CONTROLLER/wb_rst
20. /sim_mips/uut/MIPS_CORE/CONTROLLER/wb_valid
21. /sim_mips/uut/MIPS_CORE/DATAPATH/wb_wen_wb
22. /sim_mips/uut/MIPS_CORE/DATAPATH/regw_addr_wb[4:0]
23. /sim_mips/uut/MIPS_CORE/DATAPATH/regw_data_wb[31:0]
24. /sim_mips/uut/MIPS_CORE/CPO/ir_in
25. /sim_mips/uut/MIPS_CORE/CPO/oper[1:0]
26. /sim_mips/uut/MIPS_CORE/CPO/addr_r[4:0]

标题	文档编号	版本	页
XINGDENG	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0 23 of 34
作者	修改日期		
Joseph Xu	2019/1/23	公开	

27. /sim_mips/uut/MIPS_CORE/CP0/data_r[31:0]
28. /sim_mips/uut/MIPS_CORE/CP0/addr_w[4:0]
29. /sim_mips/uut/MIPS_CORE/CP0/data_w[31:0]
30. /sim_mips/uut/MIPS_CORE/CP0/jump_en
31. /sim_mips/uut/MIPS_CORE/CP0/jump_addr[31:0]
32. /sim_mips/uut/MIPS_CORE/CP0/epcr[31:0]
33. /sim_mips/uut/MIPS_CORE/CP0/ehbr[31:0]
34. /sim_mips/uut/MIPS_CORE/DATAPATH/REGFILE/regfile[1:31][31:0]

添加后将所有 4 位以上宽度的信号都设置成 16 进制显示。添加并设置后的波形列表如图 2-36 所示：



图 2-36 仿真所需要观察的所有信号

另外为了方便操作，可以直接加载预先提供的波形列表配置文件，具体操作如下：
在仿真视图下，选择 File→Open Waveform Configuration，如图 2-37 所示：

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	24 of 34
作者	修改日期		
Joseph Xu	2019/1/23	公开	

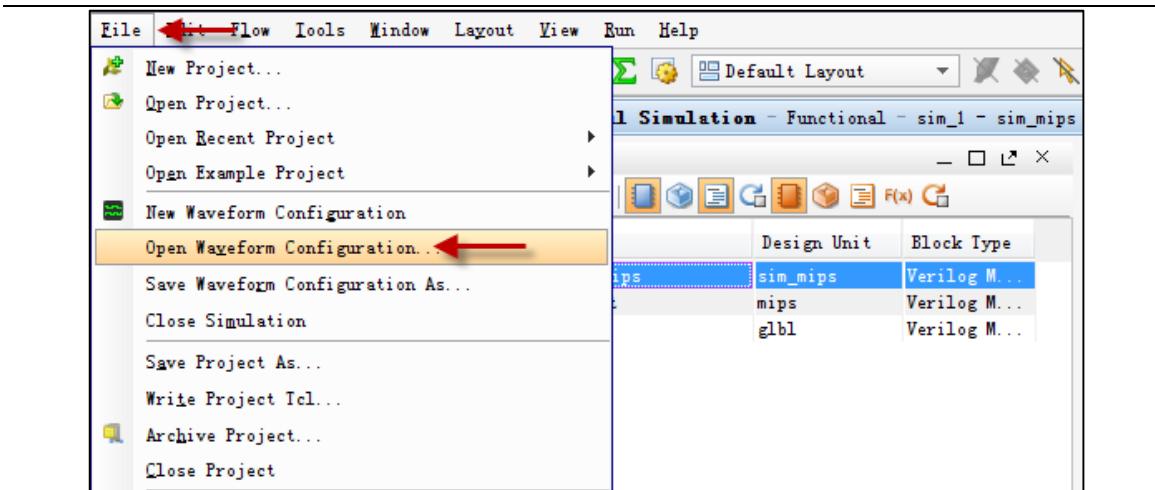


图 2-37 菜单中选择“Open Waveform Configuration”

切换到 C:\archlabs\ComputerArchitecture\lab_source\lab06\simulation 目录，如图 2-38 所示：

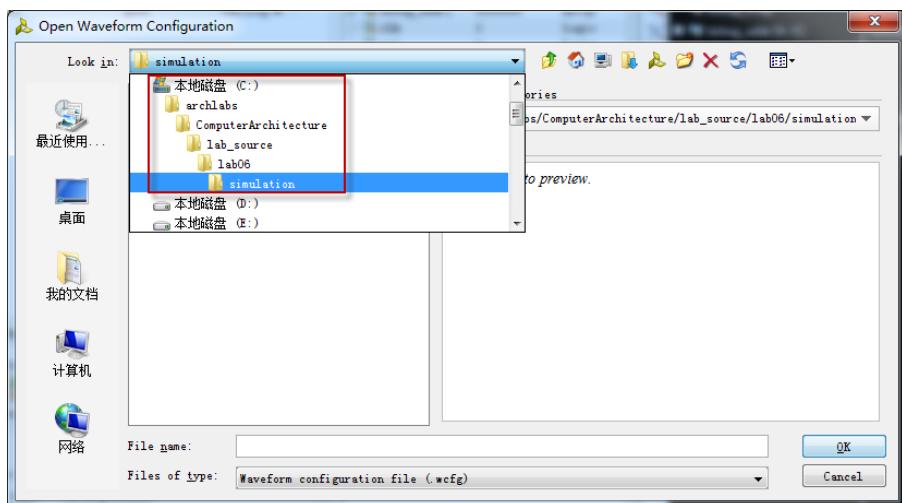


图 2-38 切换目录

选择 lab6_sim.wcfg 文件，然后点击 OK。如图 2-39 所示：

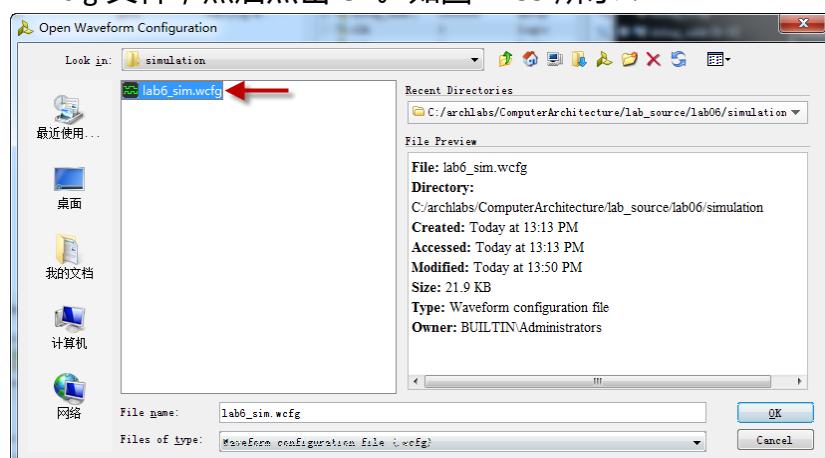


图 2-39 选择 wcfg 文件

添加后的信号如图 2-40 所示：

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	25 of 34
作者	修改日期		
Joseph Xu	2019/1/23	公开	

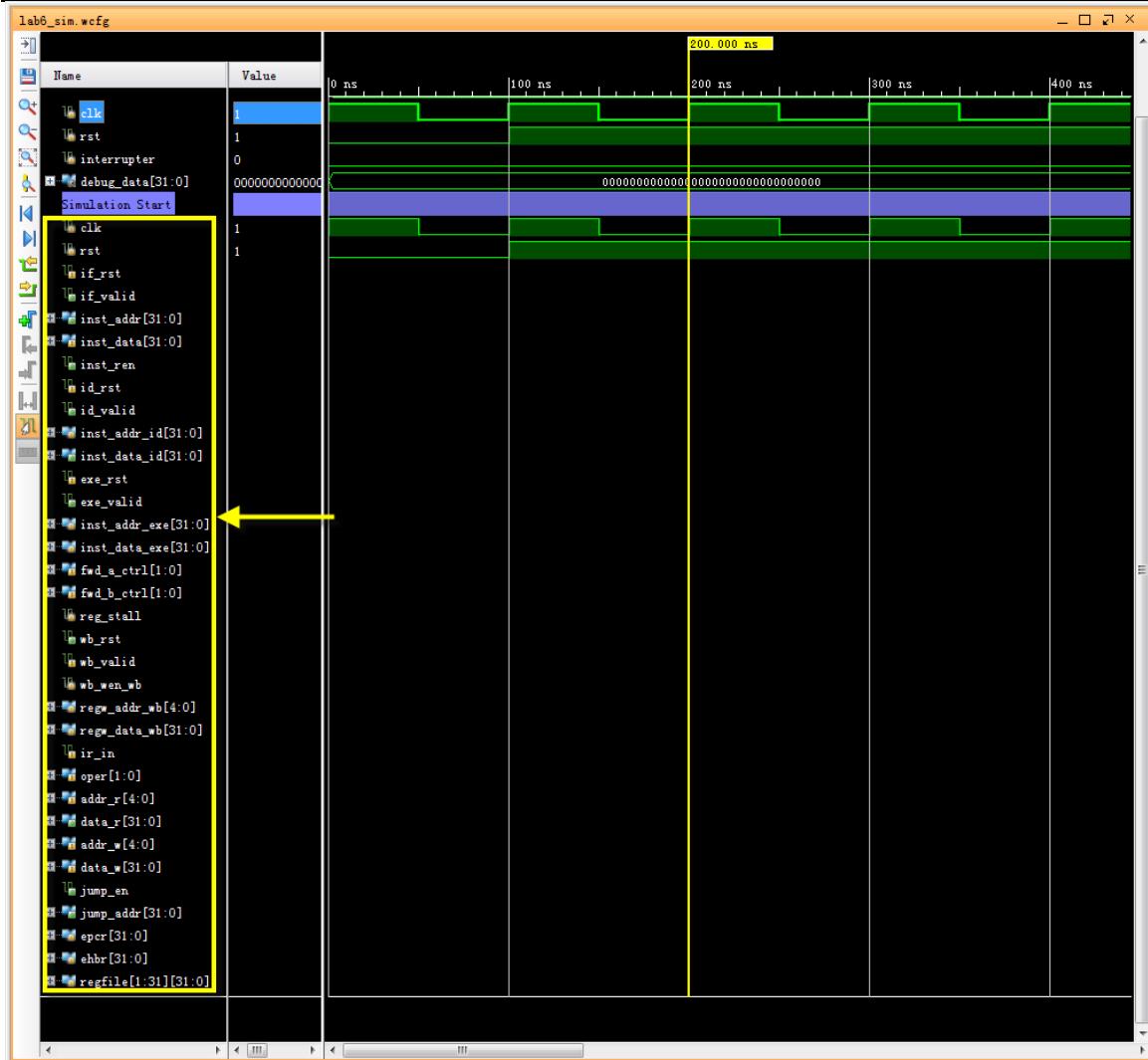


图 2-40 加载波形配置文件后的波形列表视图

增加信号后需要使用 Run 菜单下的 Restart , 重新开始仿真 , 如图 2-41 所示 :

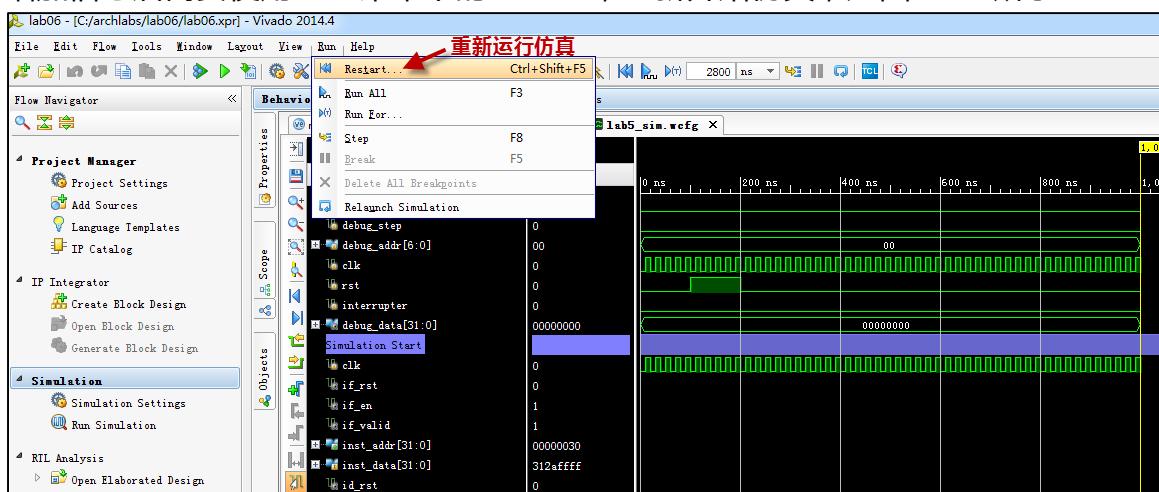


图 2-41 Restart Simulation

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	26 of 34
作者	修改日期		
Joseph Xu	2019/1/23		公开

使用 Run for... 仿真指定时间长的波形，如图 2-42 所示：

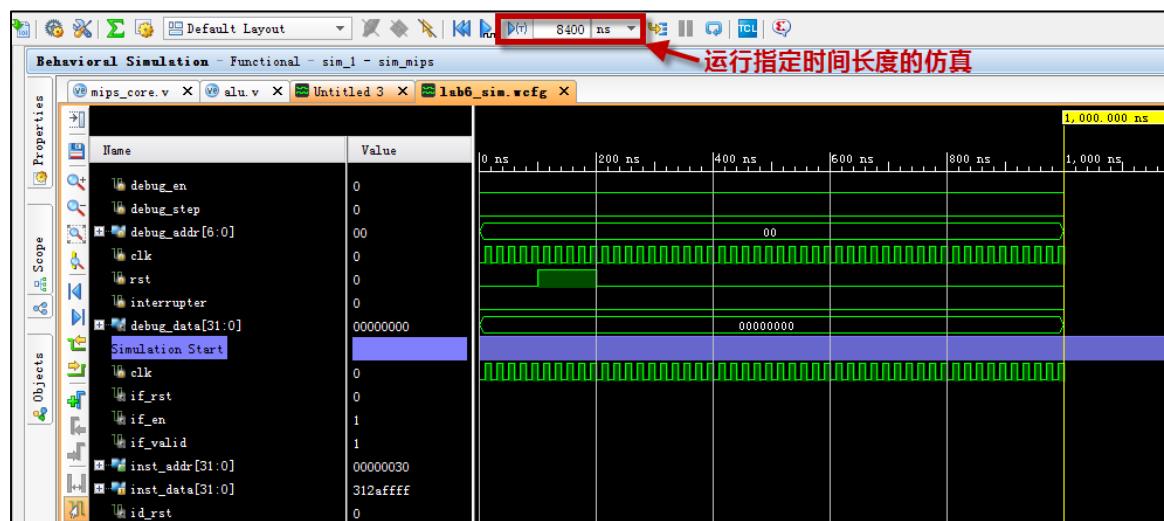


图 2-42 Run for...

通过键盘 Shift+ “-” 和 Shift+ “+” 可以对波形图进行缩放。

将仿真出的波形和下面 2 张图对比，可以知道设计的电路是否正常工作。

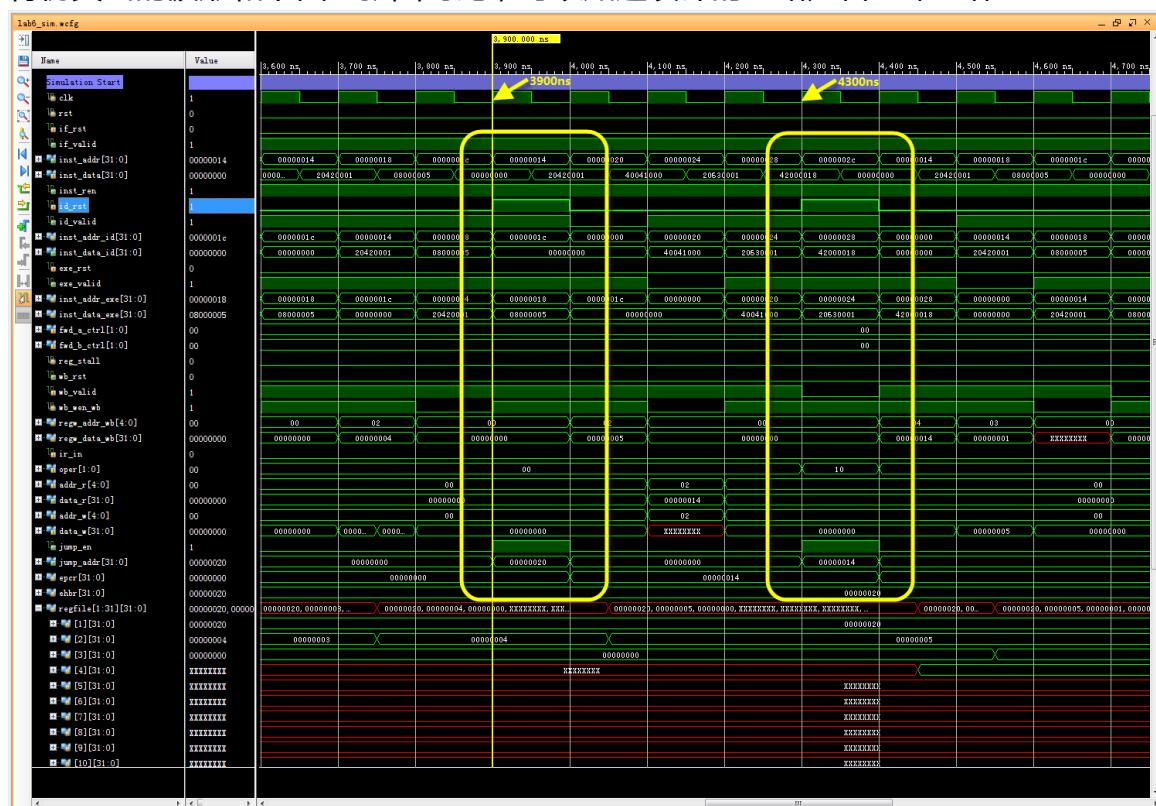


图 2-43 仿真结果 1

标题	文档编号	版本	页	
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	27 of 34	
作者	修改日期			
Joseph Xu	2019/1/23	公开		

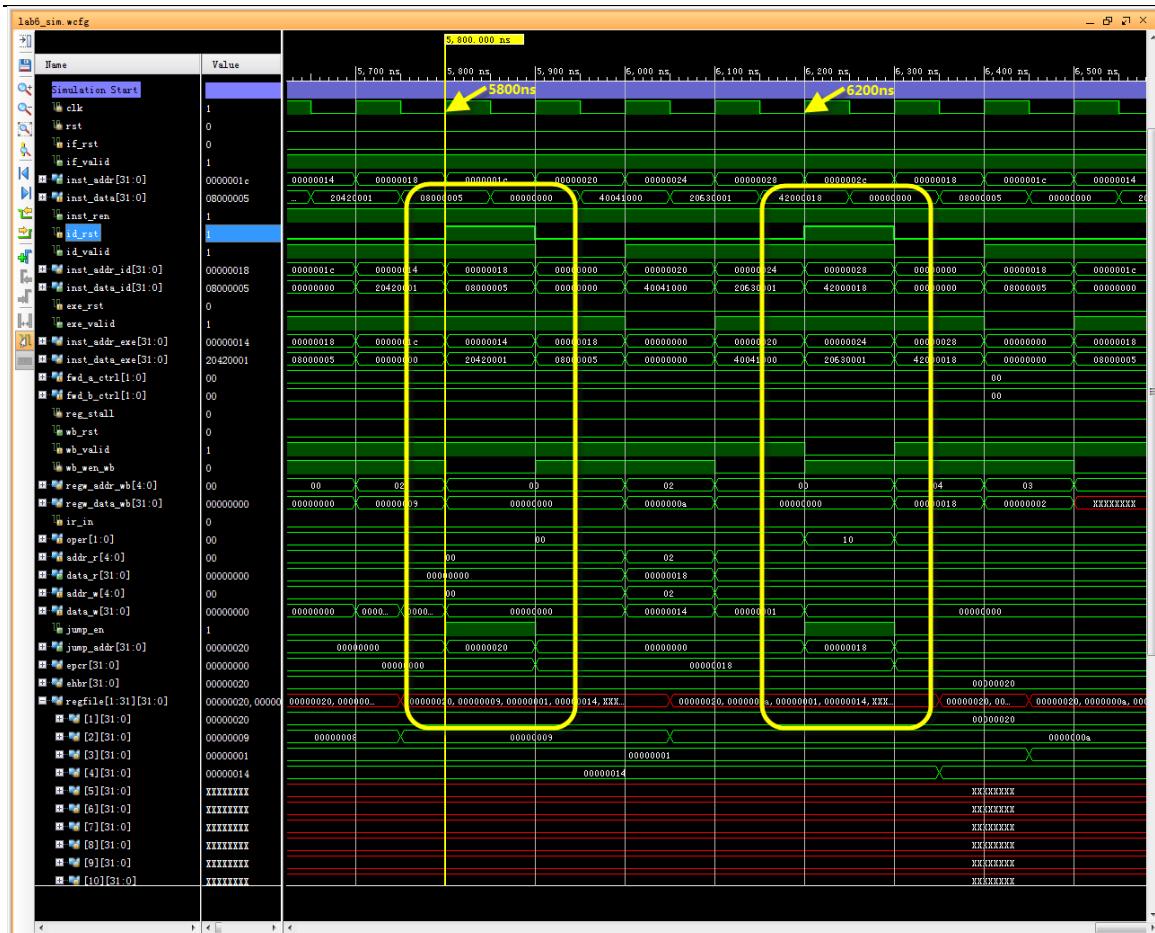


图 2-44 仿真结果 2

仿真结束后，在菜单中选择 File→Close Simulation，关闭仿真视图。如图 2-45 所示：

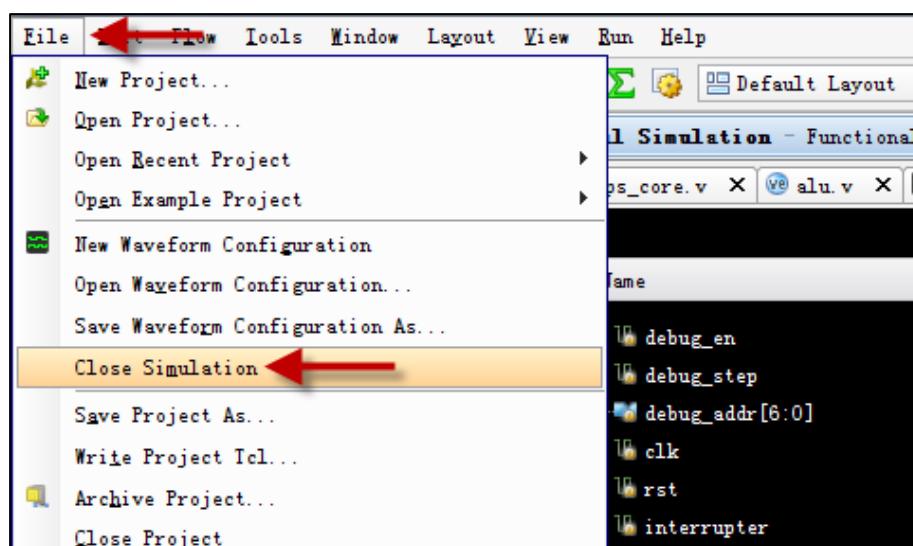


图 2-45 关闭仿真视图

XINGDENG	标题 Lab6: Pipelined CPU supporting interrupt	文档编号 XD-LAB-ARCH-006 1.0 28 of 34	版本	页
			作者 Joseph Xu	修改日期 2019/1/23

2.3 生成 bitstream 文件

回到 Vivado 主界面后，确认 mips_top.v 被设置为顶层文件后。点击 Generate Bitstream 生成 FPGA 配置文件。如图 2-46 所示：

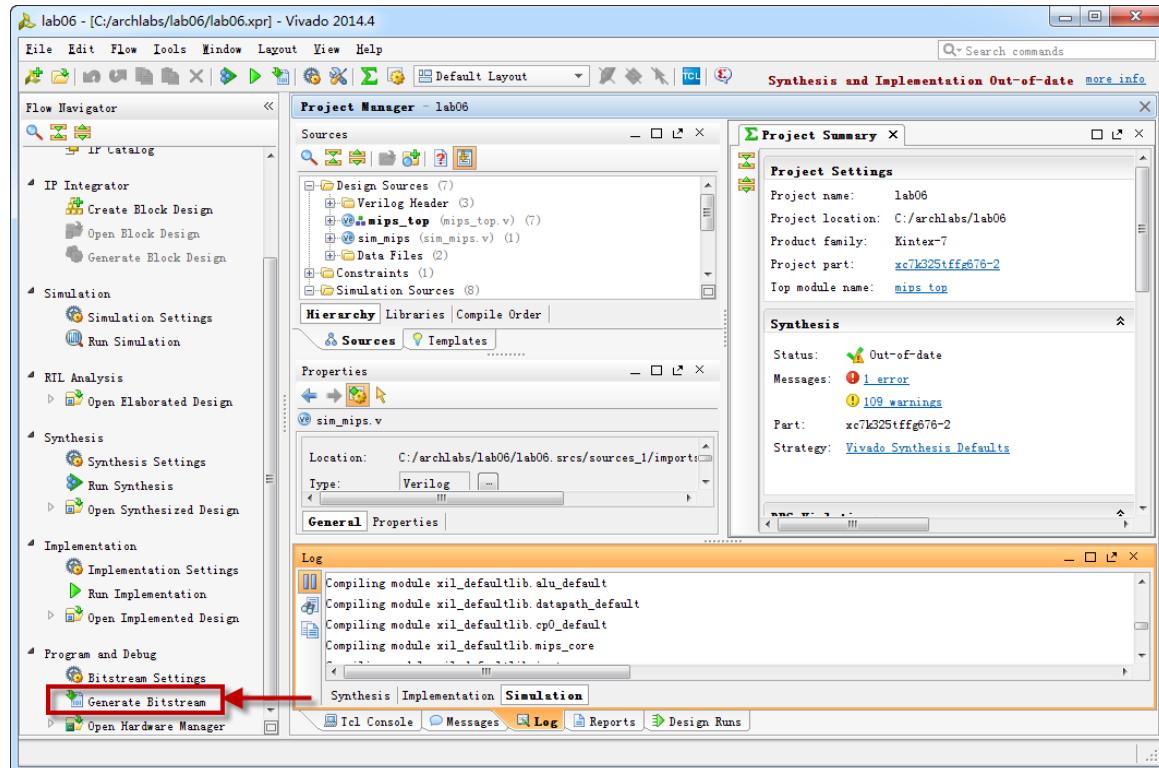


图 2-46 生成 bitstream

在询问对话框中选择 Yes，自动运行 synthesis 和 implementation，如图 2-47 所示：

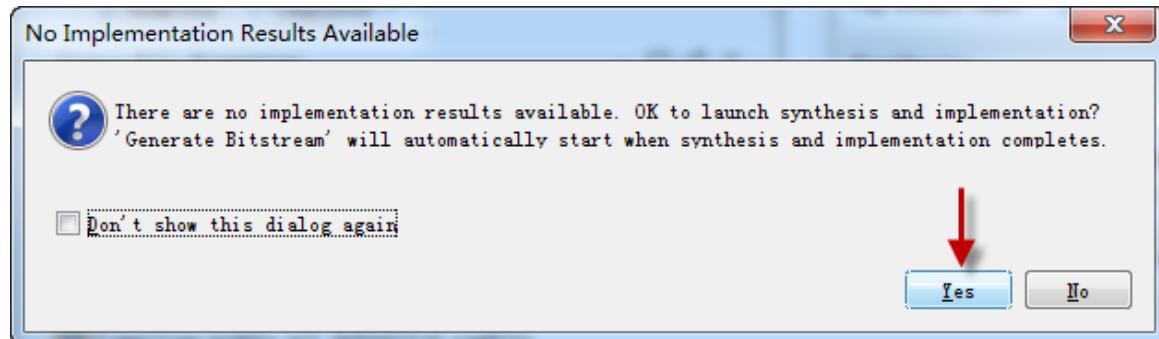


图 2-47 自动运行中间步骤

如图 2-48 所示，当显示 Bitstream Generation Completed 时表示文件成功生成，点击 Cancel 结束。

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	29 of 34
作者	修改日期		
Joseph Xu	2019/1/23	公开	

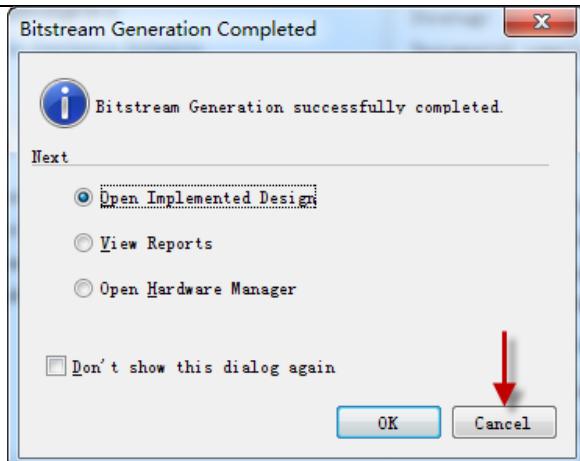


图 2-48 生成 bitstream 成功

2.4 下载验证

仿真完成后，下载 bit 文件到 SWORD4.0 开发平台查看真实效果。按图连接 SWORD4.0 的 12V 电源和 JTAG，然后打开开关。

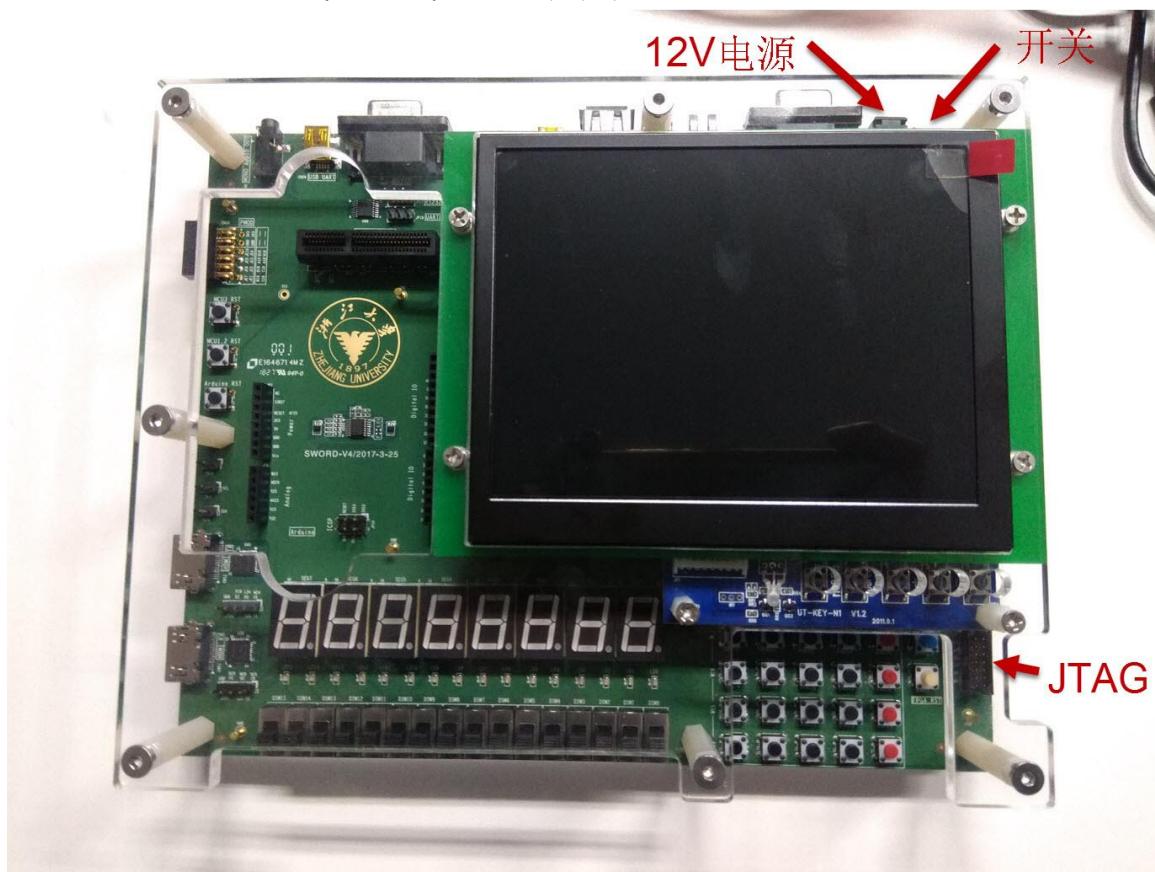


图 2-49 连接 SWORD4.0

XINGDENG	标题	文档编号	版本	页
	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	30 of 34
作者	修改日期			
Joseph Xu	2019/1/23			

SWORD4.0 的控制开关主要有拨码开关、键盘矩阵、复位按钮等。

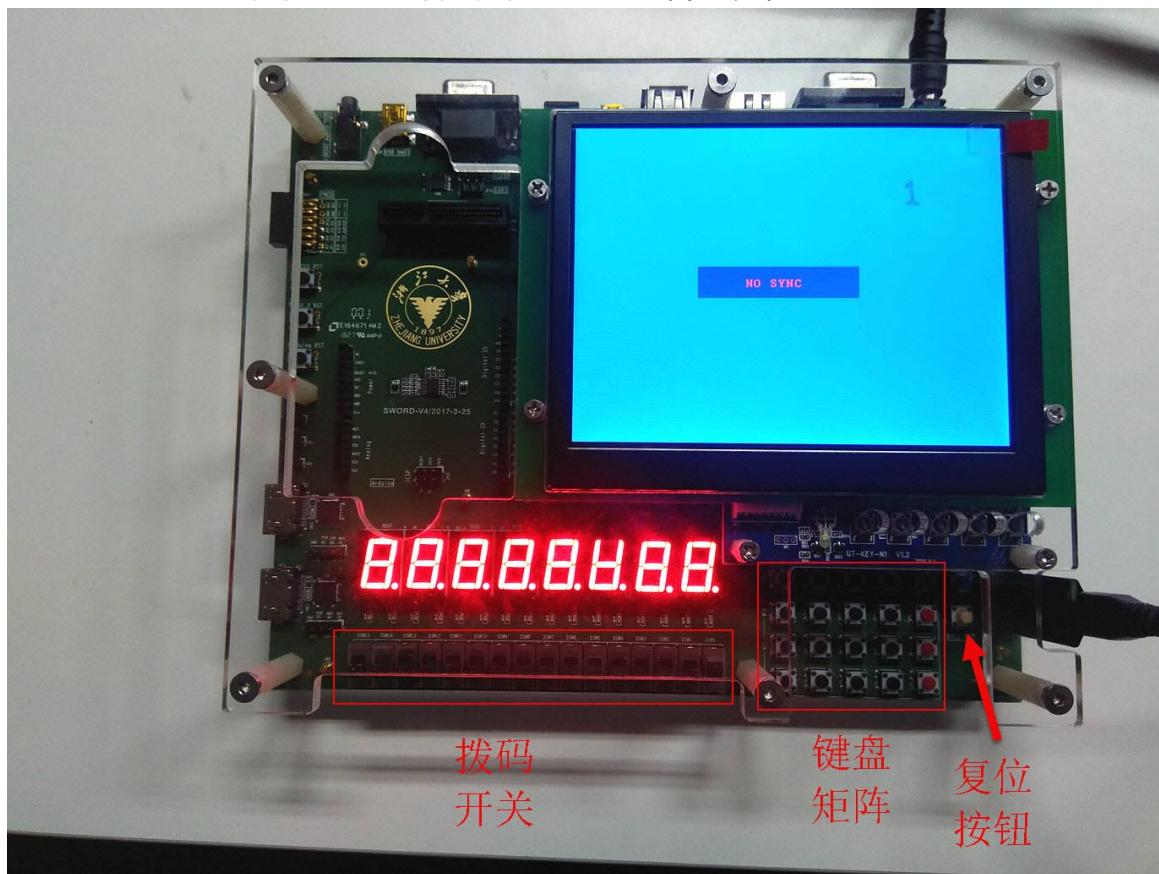


图 2-50 主要控制 I/O

在 Vivado 中点击 Open Hardware Manager。

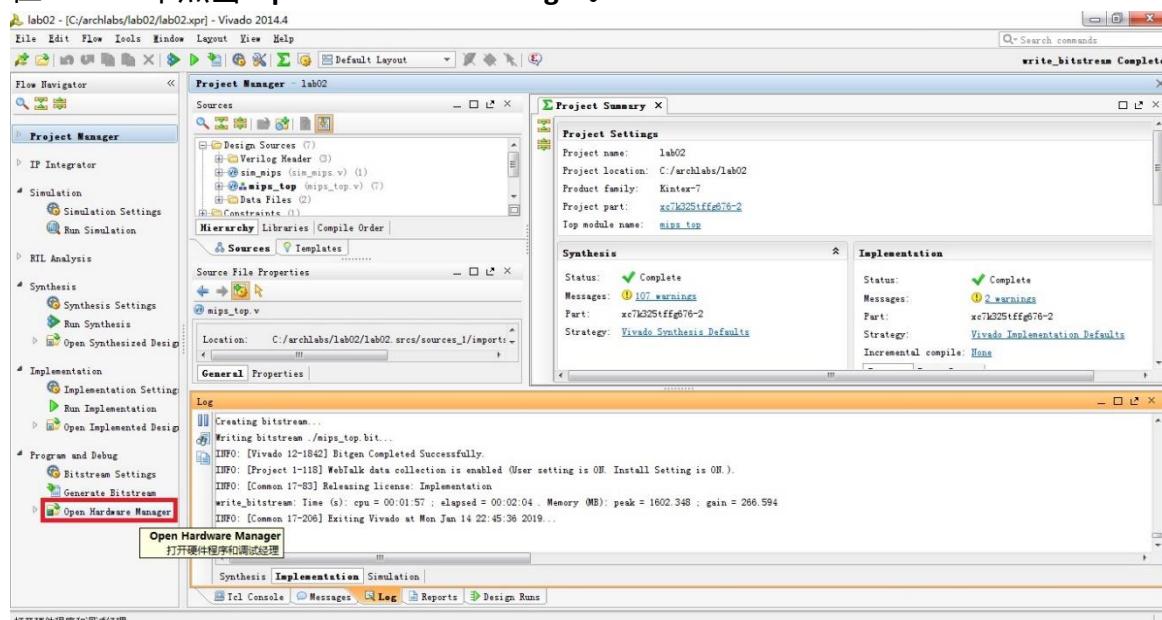


图 2-51 进入 Hardware Manager

XINGDENG	标题	文档编号	版本	页
	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	31 of 34
作者	修改日期			
	Joseph Xu	2019/1/23	公开	

点击 Open target , 然后点击 Auto Connect。

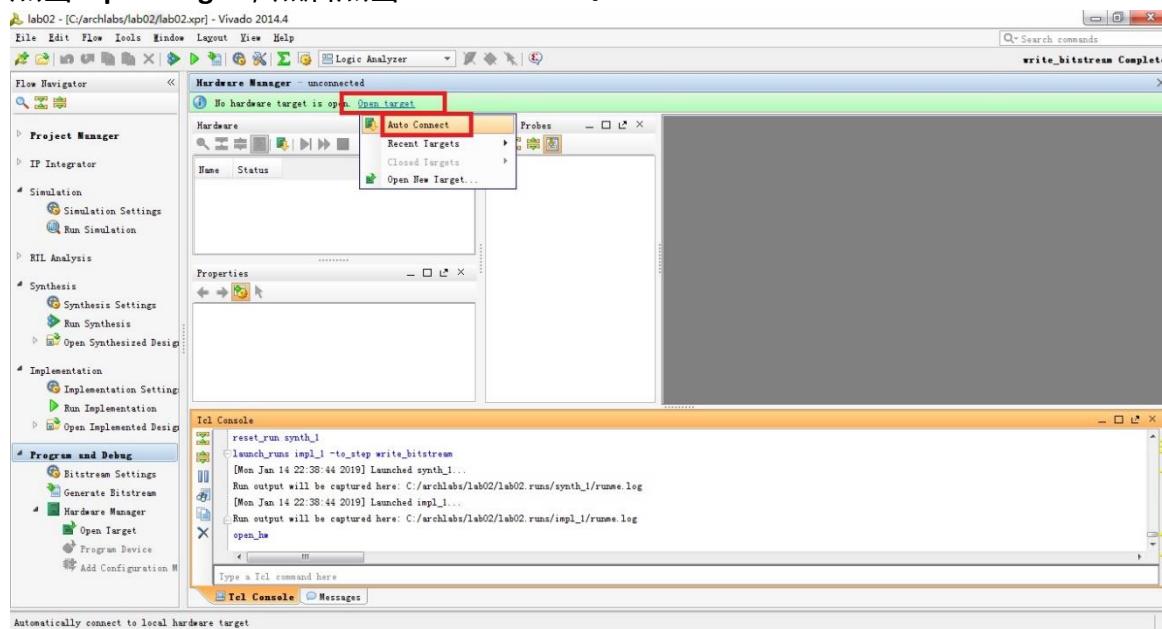


图 2-52 自动连接到 SWORD4.0

右键点击找到的 xc7k325t_0 , 选择 Program Device。

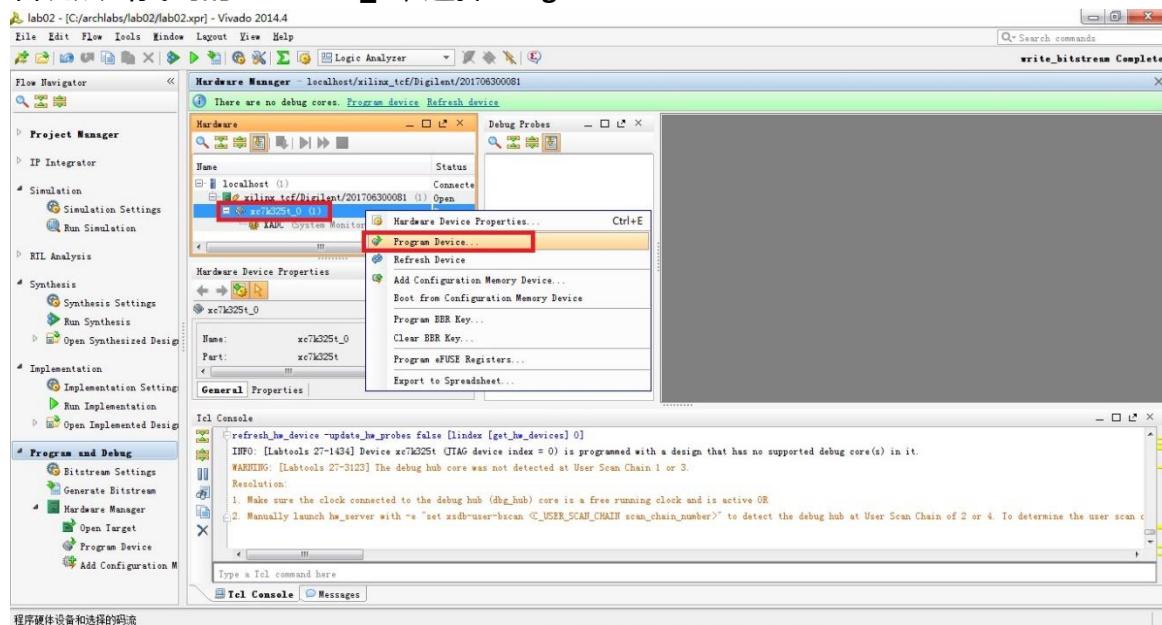


图 2-53 Program Device

点击 Program 下载 bit 文件。

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	32 of 34
作者 Joseph Xu	修改日期 2019/1/23	公开	

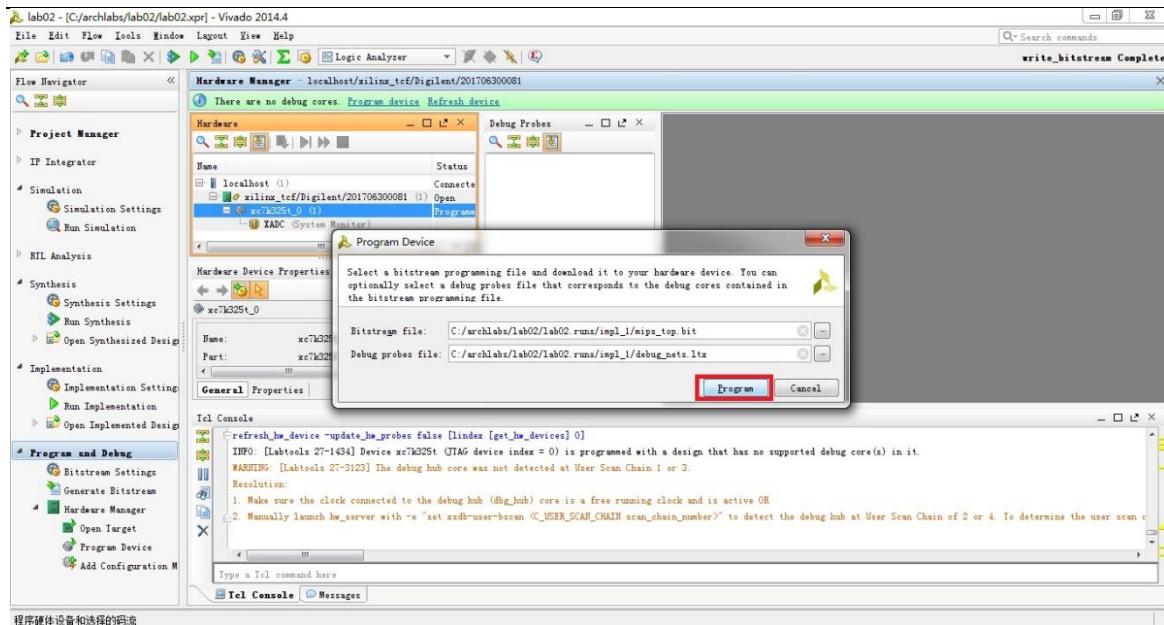


图 2-54 下载 bit 文件

下载完成后 SWORD4.0 上的反应，液晶屏上会显示调试信息。

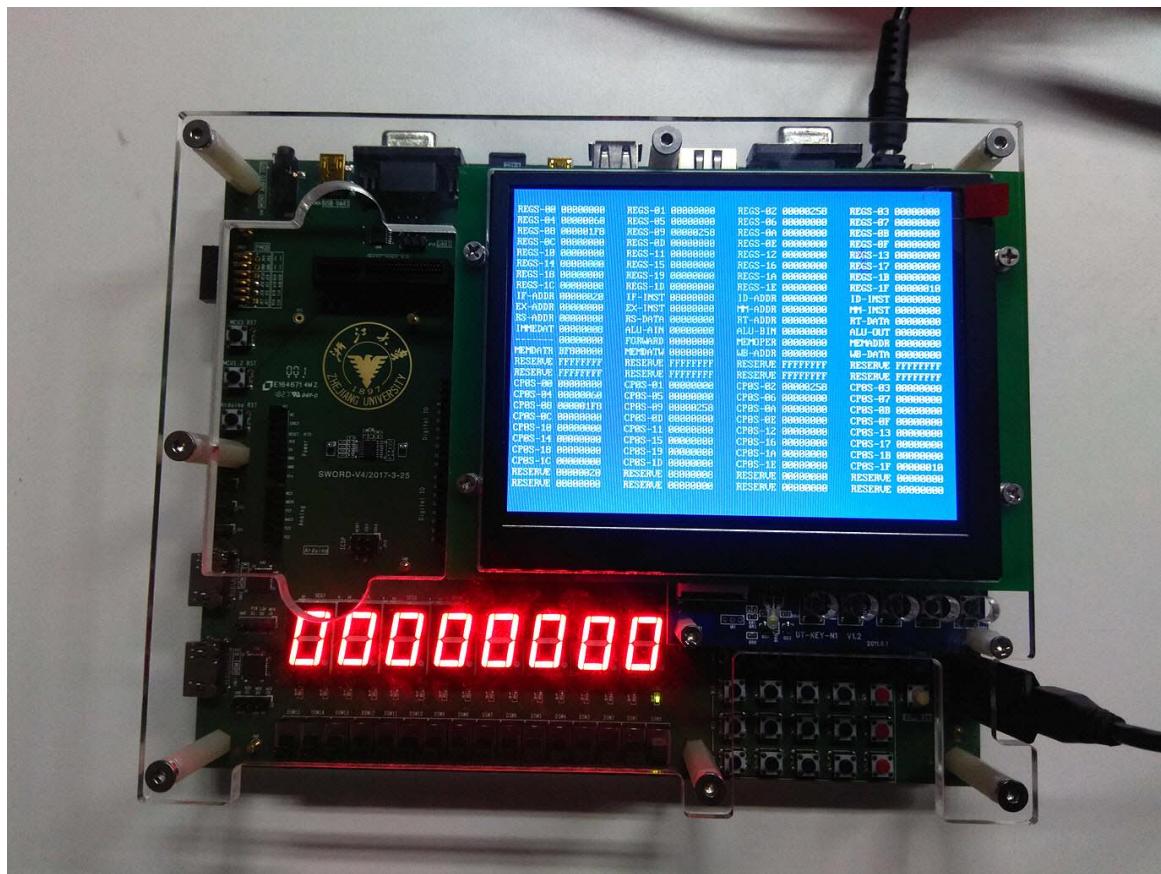


图 2-55 运行效果

通过将拨码开关 DSW0 拨向上，可以开启单步模式。之后每按一下键盘矩阵左下角的 BTNX4YO，CPU 就执行一条指令。按复位键 FPGA RST 按钮回到初始状态。

标题	文档编号	版本	页
Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	33 of 34
作者 Joseph Xu	修改日期 2019/1/23		公开

在单步模式下，通过每一步液晶屏显示的数据与仿真波形的对比，查看流水线cpu是否正常工作。

	标题	文档编号	版本	页
	Lab6: Pipelined CPU supporting interrupt	XD-LAB-ARCH-006	1.0	34 of 34
作者	Joseph Xu	修改日期	2019/1/23	公开