

## cuter

## Vivado实现纯逻辑开发——从最简单的开始

发表于 2014/7/20 0:51:14 阅读 ( 38376 ) 评论 ( 26 )

0  
赞

## 作者



cuter

关注



文章:126 篇

阅读:612454 次

评论:712 次

## Vivado实现纯逻辑开发

## ——从最简单的开始

硬件平台：ZedBoard

开发工具：Vivado 2014.2

## 1. 前言：

## 1.1. 关于软件的安装

本来不想提这一环节的，因为去官网下载安装包，安装，到网上找个license就搞定了，结果在这一环还走了弯路，不得不记录下来，作为教训和供他人借鉴的经验。

Xilinx官网有ISE+Vivado合体的安装包，也有网友推荐这个安装包，一次装俩，何乐而不为呢？本着这种贪小便宜的想法，我下载了这个安装包，下载的时候发现这个安装包和ISE的安装包差不多大，当时也没太在意。

我的破电脑哼哼唧唧下了N小时，完了安装，发现是Vivado版本是2012.4，当时觉得版本太低了，不太爽，想想再下载又要N久，先凑合用吧。破解，打开Vivado发现这个界面太熟悉了，和PlanAhead、XPS的欢迎界面很像，有点不好的预感啊。

新建工程，发现不支持zynq系列芯片，只支持v7和k7两个系列，太坑了！既不支持以前的低端系列，也不支持后来的高端系列。

无奈啊，回过头去看启动Vivado时弹出的命令行窗口，根据一些提示信息可以推测，这个版本的Vivado只是做了一个界面，真正工作时是跑去调用ISE的组件，这也是为什么这个安装包的大小和ISE安装包大小差不多的原因。

在这里告诉后来人，不要贪小便宜，果断不要安装这个合体包，老老实实下载比较新的版本，最好不是最新的，有可能没破解。

## 1.1. 本来应该是前言的前言

在接触新东西的时候，我总会带有一丝敬畏，好像未知的东西总是不如熟悉的东西让人感觉踏实。所以我喜欢用自己掌握的很熟练的或者是很简单的东西去验证未知的东西，加快自己学习的进度。现在我选择上手Vivado，也打算从最简单的纯逻辑设计开始。

## 2. 规划：

在设计之初就能够进行良好地规划，可以节省后续的设计时间，当项目复杂度增加时，项目的规划和方案设计更加重要。

- 1) 项目功能：流水灯
- 2) 编程语言：Verilog
- 3) 设计流程：建立工程→代码编辑→功能仿真→综合、实现→板级测试（本文是现象的观察，后续会给出ChipScope的使用方法）

项目的功能十分简单，主要目的是为了Vivado开发流程，也就是上述第3个步骤。

## 3. 设计

功能简单，上个示意框图，简单描述一下吧。

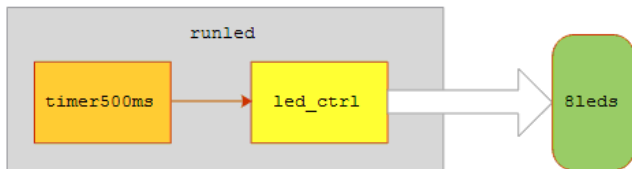


图1 总体框图

## 标签

Vivado ZedBoard

## 相关文章

大型多媒体中央控制系统的原理及应用  
简述 LED显示屏系统的基本构成  
西门子中国淡出消费电子 转型能源项目  
嵌入式PROFIBUS-DP从站通信接口设计  
研究项目致力于改善智能电网的能源分配  
紫金桥监控组态软件为南压浅冷站的LED

## 相关资料

ADE7762  
ADG602  
ADM8618ZACYKS-RL  
AD8091AR-REEL7  
AD8109  
NAF点乘算法的并行计算研究

## 相关视频

【视频】模拟器件AFE5805 用于超声波  
【视频】2011面向智能控制的嵌入式系  
【视频】什么是Haptics? 第二部分  
【视频】低功耗电视展示  
【视频】IPTV展示  
【视频】TLV320AIC3254配有mini-DSP的

timer500ms模块:

功能是产生2hz的时钟，为led\_ctrl模块提供时钟。

led\_ctrl模块:

功能是实现循环移位，在硬件上的体现就是流水灯。输入时钟为2hz，时钟上升沿进行一次移位操作，每个led点亮0.5s。

## 4. 实现:

### 4.1. 建立工程

下载了最新版的Vivado2014.2，找了个license，工具可以用了，IP可不可以还没有验证。欢迎界面看起来和之前的PlanAhead、XPS的欢迎界面差别比较大了，可以说是焕然一新吧。

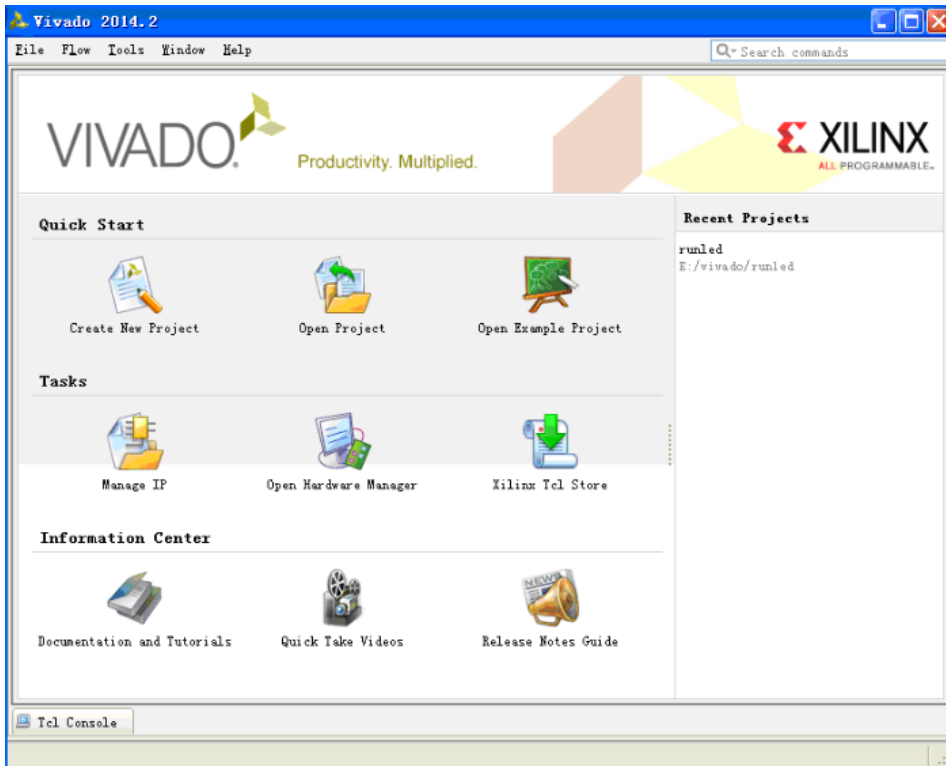


图2 欢迎界面

点击Create New Project，进入新建工程的向导，逐步next即可。



图3 新建工程向导step1

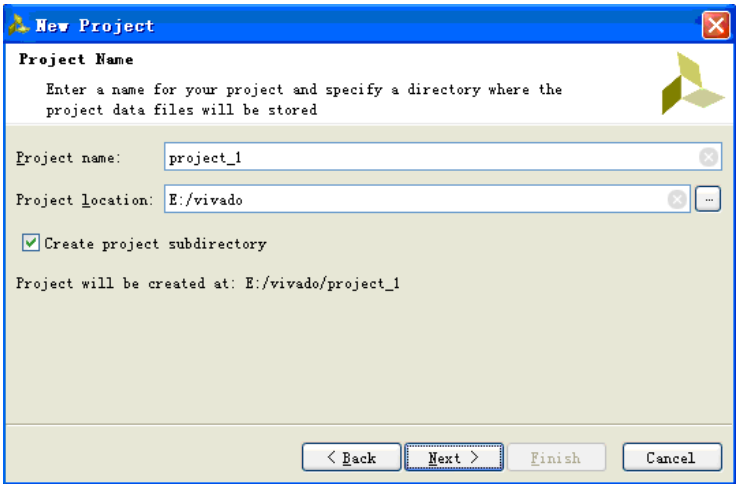


图4 新建工程向导step2 指定工程名和存储路径

勾选图5椭圆内的选项，表明不在此时指定源文件。

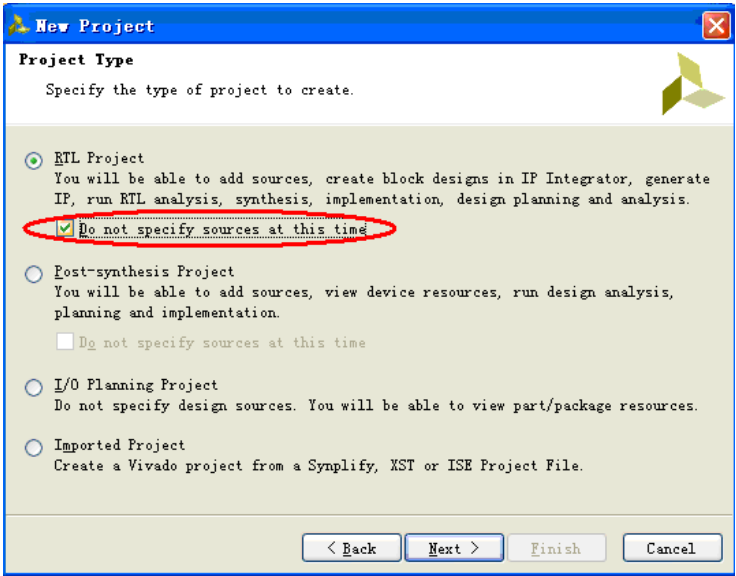


图5 新建工程向导step3

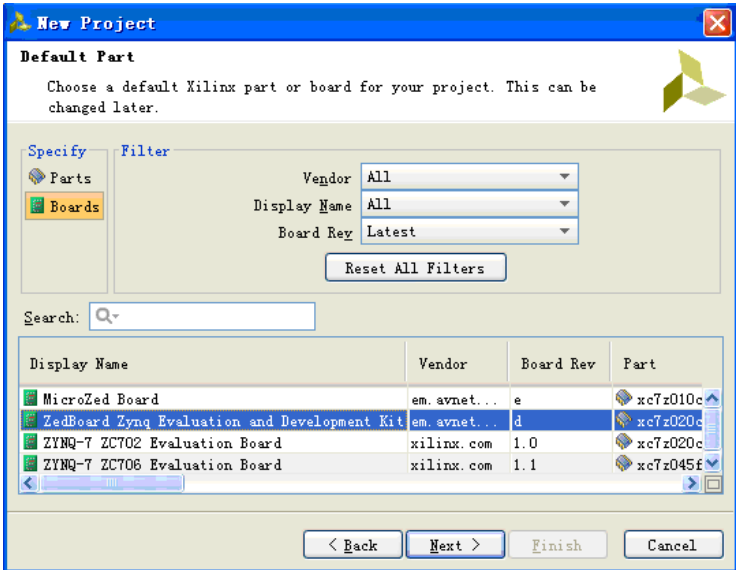


图6 选择器件/开发板

最后一步：所建立工程的概要信息。

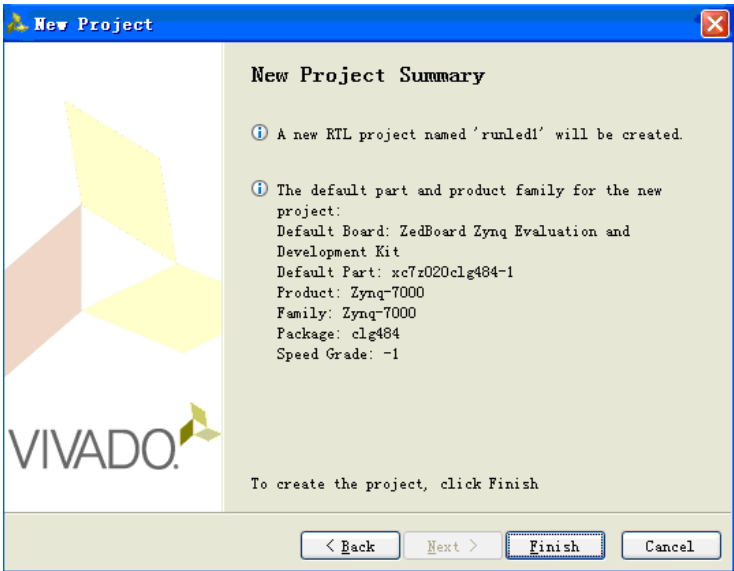


图7 新建工程信息总览

点击Finish之后，就进入Vivado主界面了，如图8所示。

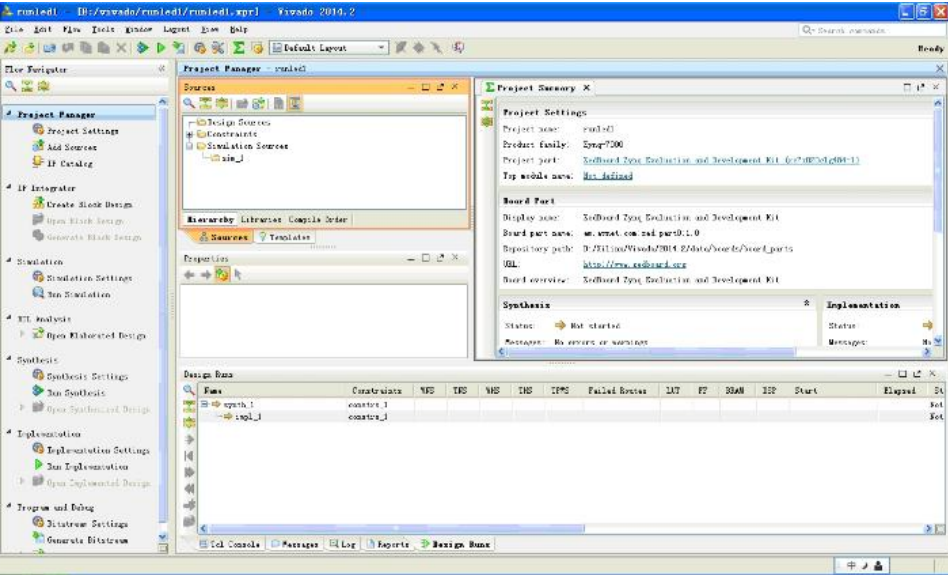


图8 Vivado主界面

主界面主要包括：

- Flow Navigator: 在此可以找到整个设计过程中涉及到的所有流程
- Sources: 工程包含的设计源文件，源码、约束等
- Properties: 所选中对象的属性信息
- Project Summary: 在此可以查看工程信息
- Design Runs: 在此可以查看提示信息、警告、错误、也可以输入tcl命令

4.2. 流程控制子窗口

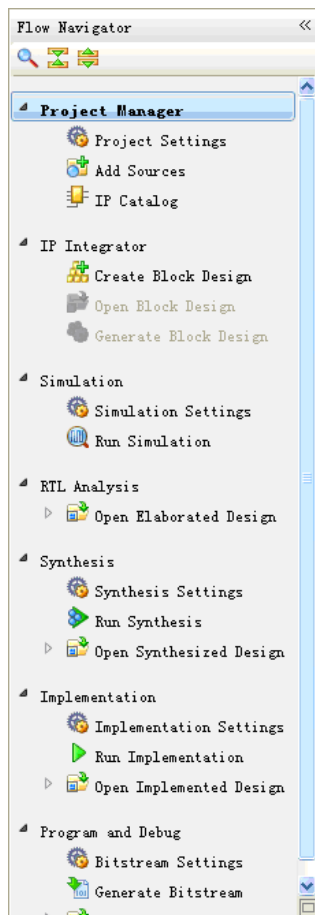


图9 流程子窗口

后续的“Add Sources”、“Simulation”、“Synthesis”、“Implementation”以及“Generate Bitstream”，都可以在该窗口找到执行菜单。

#### 4.3. 代码编辑

顶层：

```
module runled_top(  
    input iClk,  
    input iRst_n,  
    output [7:0] oLed  
);  
  
wire    clk2hz_sig;  
  
timer500ms timer500ms_inst(  
    .iClk100mhz(iClk),  
    .iRst_n(~iRst_n),  
    .oClk2hz(clk2hz_sig)  
);  
  
led_ctrl led_ctrl_inst(  
    .iClk(clk2hz_sig),  
    .iRst_n(~iRst_n),  
    .oLed(oLed)  
);  
  
endmodule
```

时钟发生：

```

/*
oFreq = iFreq/(2N)
=>
N=iFreq/(2oFreq)=100,000,000/(2*2)=25,000,000=0x17d7840
*/
module timer500ms(iClk100mhz,iRst_n,oClk2hz)
input  iClk100mhz;
input  iRst_n;
output oClk2hz;          // period = 1/2s

reg    [31:0] cnt2hz;
always @(posedge iClk100mhz) begin
if(iRst_n==1'b0) begin
    oClk2hz <= 1'b0;
    cnt2hz <= 32'b0;
end
else begin
    if(cnt2hz==32'd25000000) begin
        cnt2hz <= 32'b0;
        oClk2hz <= ~ oClk2hz;
    end
end
end
end
endmodule

```

### led控制模块:

```

module led_ctrl(
    input iClk,
    input iRst_n,
    output [7:0] oLed
);
reg [7:0] led;

/*
\ 每个时钟上升沿, 左移1bit, 写法是为了实现循环左移
*/
always @(posedge iClk) begin

    if(iRst_n==1'b0)
        led <= 8'b1;
    else
        led <= {led[6:0],led[7]};

end

assign oLed = led;

endmodule

```

添加代码源文件方法: 点击Add Sources按钮, 有两处有该按钮, 如图10所示。

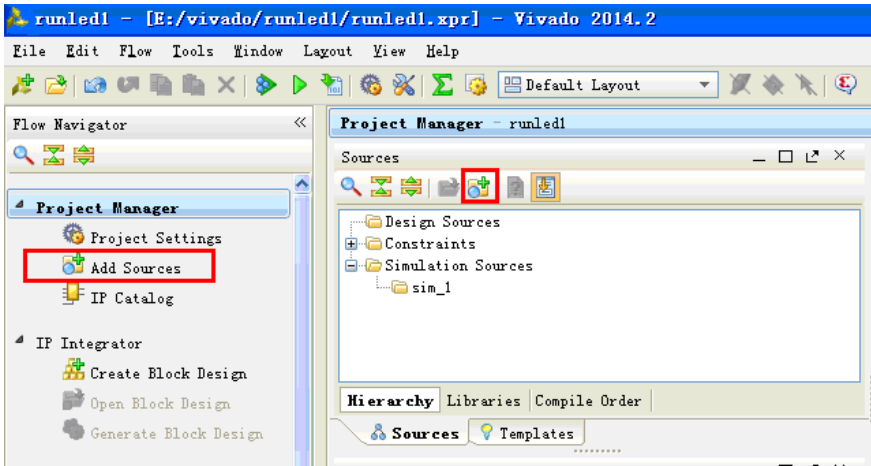


图10 add source(1)

另外，还可以在Design Sources文件夹上单击鼠标右键，弹出的菜单中有“Add Sources”选项，如图11所示。

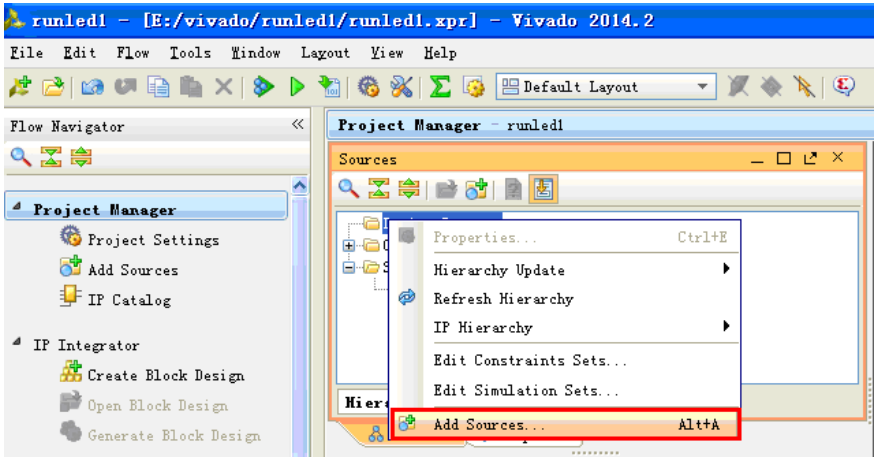


图11 add source (2)

弹出图12所示的向导。

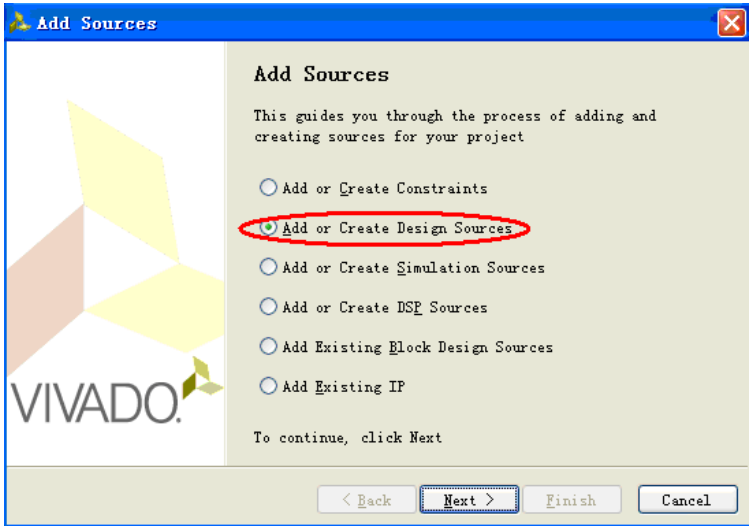


图12 选择资源类型

此处添加的是HDL设计文件，所以选择椭圆内的“Add or Create Design Sources”，NEXT，弹出图13所示的界面，选择创建或者添加源文件。

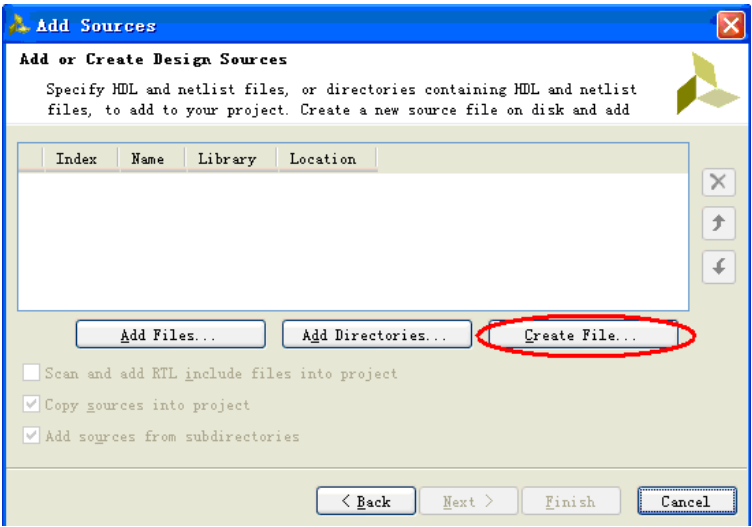


图13 创建/添加源文件

单击Create File按钮，进入图14所示的界面。

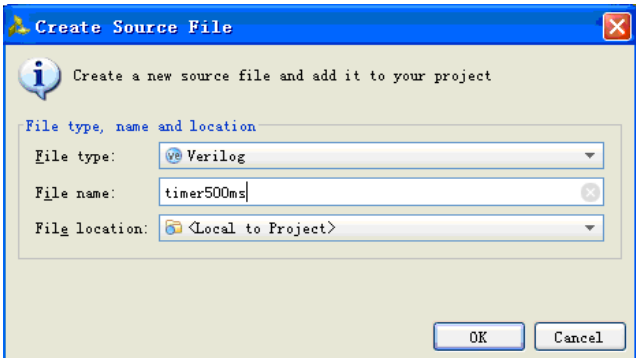


图14 新建源文件

指定文件名和存储位置后点击OK，返回上一界面，界面发生变化，如图15所示。

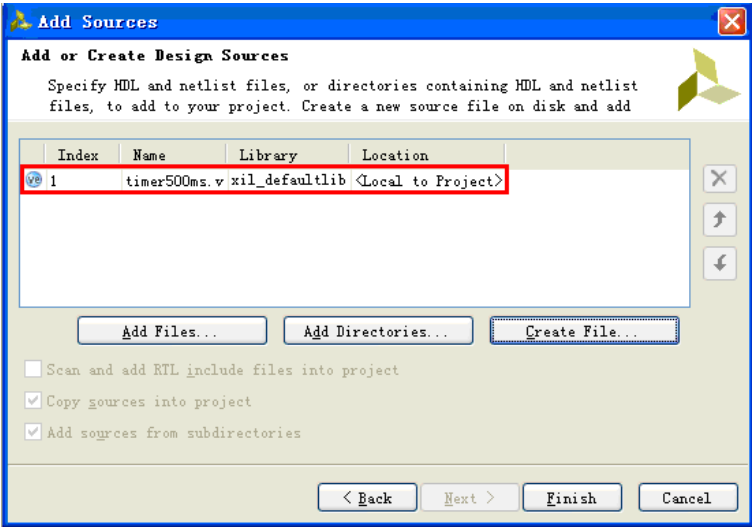


图15

点击Finish，弹出新建文件的助手，如图16所示。



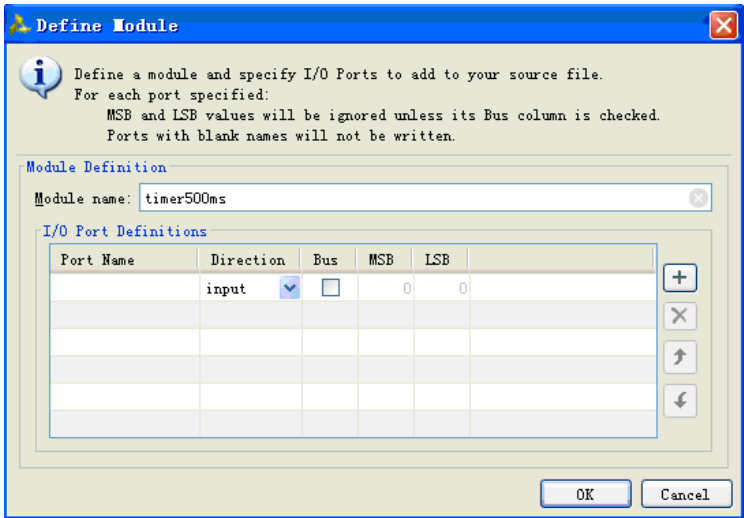


图16 源文件编辑助手

这里应该和ISE添加源文件的向导一样，我直接点击OK略过，然后自己编辑的代码。

4.4. 功能仿真

添加testbench，方法和4.3所述的类似，只是文件类型为“Add or Create Simulation Sources”（图12）。

Run Simulation，会弹出仿真结果。此处以timer500ms模块的仿真为例，给出仿真结果：

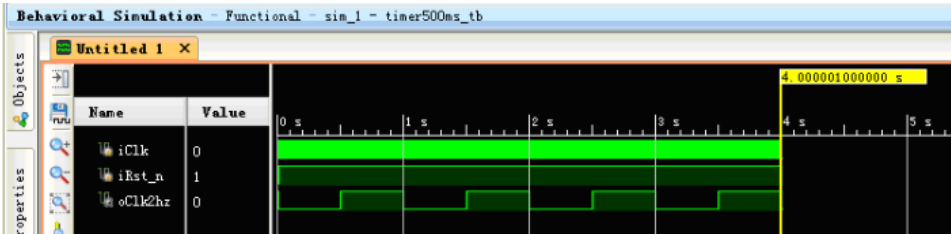


图17 timer500ms模块仿真结果

体会：在ISE中新建Testbench文件时，可以选择被仿真模块，然后自动实例化待仿真的模块；Vivado中需要自己手动编写所有代码。

4.5. 综合、实现、烧写

4.5.1. 约束文件的编写

ISE约束文件格式为ucf，Vivado约束文件为xdc，这是Vivado又一改进。xdc文件是我的约束文件为：

```
#In the following the XDC constraint is matched to the original UCF constraint, XDC
above, UCF below # Commented

set_property PACKAGE_PIN Y9 [get_ports {iClk}]
set_property IOSTANDARD LVCMOS33 [get_ports {iClk}]
#NET GCLK      LOC = Y9 | IOSTANDARD=LVCMOS33; # "GCLK"
# Bank 33, Vcco = 3.3V

#set_property IOSTANDARD LVCMOS33 [get_ports -filter { IOBANK == 33 } ]
set_property PACKAGE_PIN T22 [get_ports {oLed[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {oLed[0]}]
#NET LD0      LOC = T22 | IOSTANDARD=LVCMOS33; # "LD0"
set_property PACKAGE_PIN T21 [get_ports {oLed[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {oLed[1]}]
#NET LD1      LOC = T21 | IOSTANDARD=LVCMOS33; # "LD1"
set_property PACKAGE_PIN U22 [get_ports {oLed[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {oLed[2]}]
```

```
#NET LD2      LOC = U22 | IOSTANDARD=LVCNMOS33; # "LD2"

set_property PACKAGE_PIN U21 [get_ports {oLed[3]]}

set_property IOSTANDARD LVCNMOS33 [get_ports {oLed[3]]}

#NET LD3      LOC = U21 | IOSTANDARD=LVCNMOS33; # "LD3"

set_property PACKAGE_PIN V22 [get_ports {oLed[4]]}

set_property IOSTANDARD LVCNMOS33 [get_ports {oLed[4]]}

#NET LD4      LOC = V22 | IOSTANDARD=LVCNMOS33; # "LD4"

set_property IOSTANDARD LVCNMOS33 [get_ports {oLed[5]]}

set_property PACKAGE_PIN W22 [get_ports {oLed[5]]}

#NET LD5      LOC = W22 | IOSTANDARD=LVCNMOS33; # "LD5"

set_property PACKAGE_PIN U19 [get_ports {oLed[6]]}

set_property IOSTANDARD LVCNMOS33 [get_ports {oLed[6]]}

#NET LD6      LOC = U19 | IOSTANDARD=LVCNMOS33; # "LD6"

set_property IOSTANDARD LVCNMOS33 [get_ports {oLed[7]]}

set_property PACKAGE_PIN U14 [get_ports {oLed[7]]}

#NET LD7      LOC = U14 | IOSTANDARD=LVCNMOS33; # "LD7"

# Bank 34, Vcco = Vadj

#set_property IOSTANDARD LVCNMOS18 [get_ports -filter { IOBANK == 34 } ]

set_property IOSTANDARD LVCNMOS18 [get_ports {iRst_n}]

set_property PACKAGE_PIN P16 [get_ports {iRst_n}]

#NET BTNC      LOC = P16 | IOSTANDARD=LVCNMOS18; # "BTNC"
```

注释写的很清楚了，和ZedBoard的ucf文件是一一对应的，很容易理解。

不过关于约束的编写也不是这么简单能说的清的，后续还需要进一步学习。如果快速入门的话，可以参考模版文件，模版文件如图18所示：

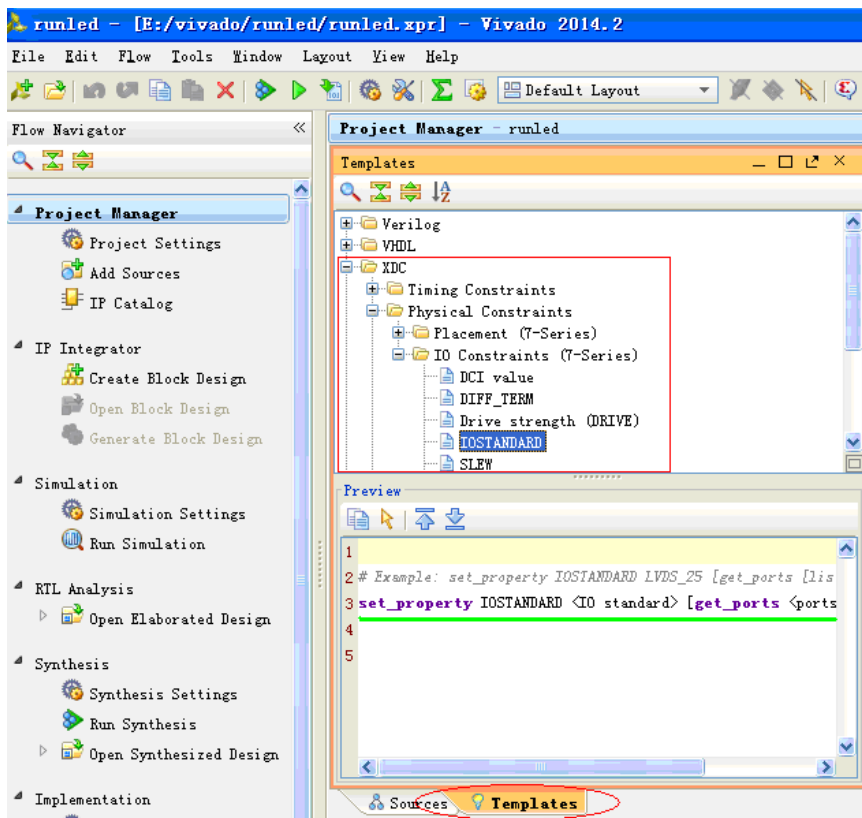


图18 模板文件的使用

注意要选中Templates子选项卡（位于Sources旁边），然后点开红色方框内的xdc文件夹，找到自己想要的约束，然后可以看到参考语句，预览结果在上图绿色直线上方。

同样的，还有verilog、vhdl、Testbench等文件的参考语句，需要时可以参考，从而提高代码编辑效率。

4.5.2. 综合、实现、烧写

综合、实现的菜单在Flow Navigator中，如果想分开执行，分别点击即可。为了方便，可以直接选择“Generate Bitstream”。除错，直到成功为止。

烧写有两种方式，一种是通过impact，一种是通过SDK Tools菜单下的Progress FPGA指令实现

4.5.3. 本节体验：

从主观感觉上来说，对于简单的设计，ISE在综合、实现的时间上要比Vivado短。

4.6. 实验现象

传个视频玩一玩。

5. 小结

通过本次学习，基本掌握了Vivado做纯逻辑开发的流程，Vivado做逻辑开发较ISE而言稍微，在操作上稍微有点繁琐，但也有方便之处，可谓各有千秋吧。

6. 隐藏的小问题

设计存在两处小问题，我没有改，有兴趣的童鞋可以看看在哪里。

版权声明：

本文由博主“cuter”发布。欢迎转载，但不得擅自更改博文内容，也不得用于任何盈利目的。转载时不得删除作者简介和版权声明。如有盗用而不说明出处引起的版权纠纷，由盗用者自负。

博客官方地址：

ChinaAET：http://blog.chinaaet.com/cuter521

EDN China：http://bbs.ednchina.com/BLOG\_cuter521\_356737.HTM

» 下一篇：【笔记】Excel辅助Modelsim调试记录

« 上一篇：CPU+FPGA实现边标志填充算法（一）

代码语言

字数统计

提交



xsomuns  
2018-11-06 11:27:35

可以考虑写一篇 关于约束文件的博客

回复



xiaoniuxxq  
2016-10-09 18:29:51

厉害了我的哥

回复

xiaohua207384390



2016-03-31 14:57:46

编程为什不使用C语言啊，verilog语言不太懂

回复



cloud2008

2016-01-15 18:08:13

代码有问题，不能直接移植，包括：等于判断写成了赋值，同步复位仿真现象错误，改成异步复位后，仿真结果正确。

回复



校园土人

2015-12-11 16:42:52

跟着学习了一下，对于熟悉vivado真不错，支持一下

回复



yukine

2015-12-08 16:39:39

博主，您好！  
4.4的仿真那块弄不出来，能否将的再详细一些。谢谢

回复



2015-11-30 11:35:00

\*\*\*此内容已被管理员屏蔽\*\*\*



dupf

2015-08-10 09:21:51

试试

回复



匿名用户

2015-08-10 09:21:10

o

回复



jtshan

2015-06-19 18:15:30

博主  
我在PS中编写c代码来控制LD9(MIO7),请问在PL中需要做什么设置吗？与PL有没有关系哪？谢谢！

回复



匿名用户

2015-04-13 18:06:55

博主，您好！  
仿真那块弄不出来，你能否再来个详细一点的，谢谢啦！！

回复



jay778778

2015-03-25 20:56:44

好例程哈 ，感谢楼主的分享，刚好需要哈

回复



匿名用户

2014-10-25 04:55:17

Vivado的威力是在大型電路, 在大型電路的合成上, 會比ISE快很多

回复



匿名用户

2014-10-11 15:02:36

约束文件这块能讲的细一点吗

回复



anchoret

2014-09-17 17:04:50

一步步学习！

回复

[«](#)

[1](#)

[2](#)

[»](#)

网站相关

- [关于我们](#)
- [联系我们](#)
- [投稿须知](#)

广告及服务

- [内容许可](#)
- [广告服务](#)
- [杂志订阅](#)

会员与积分

- [积分商城](#)
- [会员等级](#)
- [会员积分](#)
- [VIP会员](#)

关注我们

