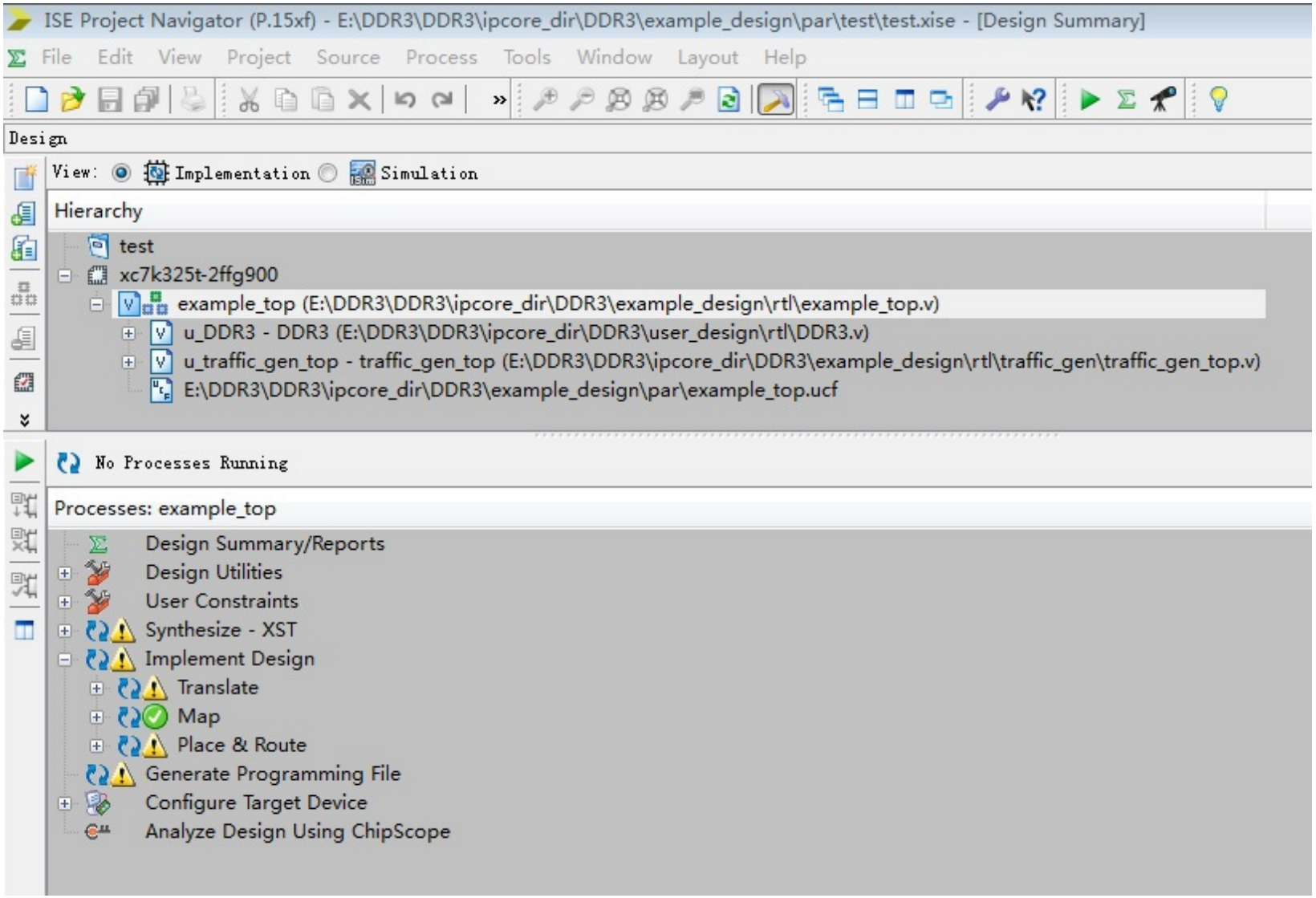
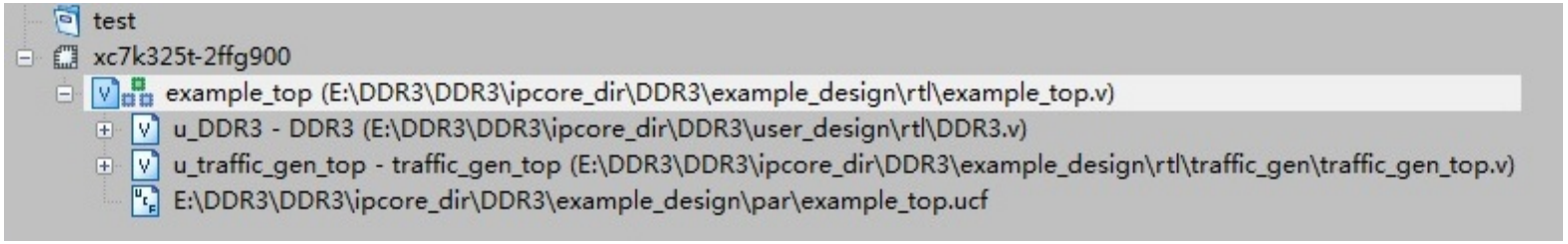
你已经看过我写的仿真教程，并且按照步骤都做成功了吧？
现在要做综合？来个一图流就行了——看明白了吗？编译已经成功，bit文件生成了。
什么？没看明白？下面分开讲吧~





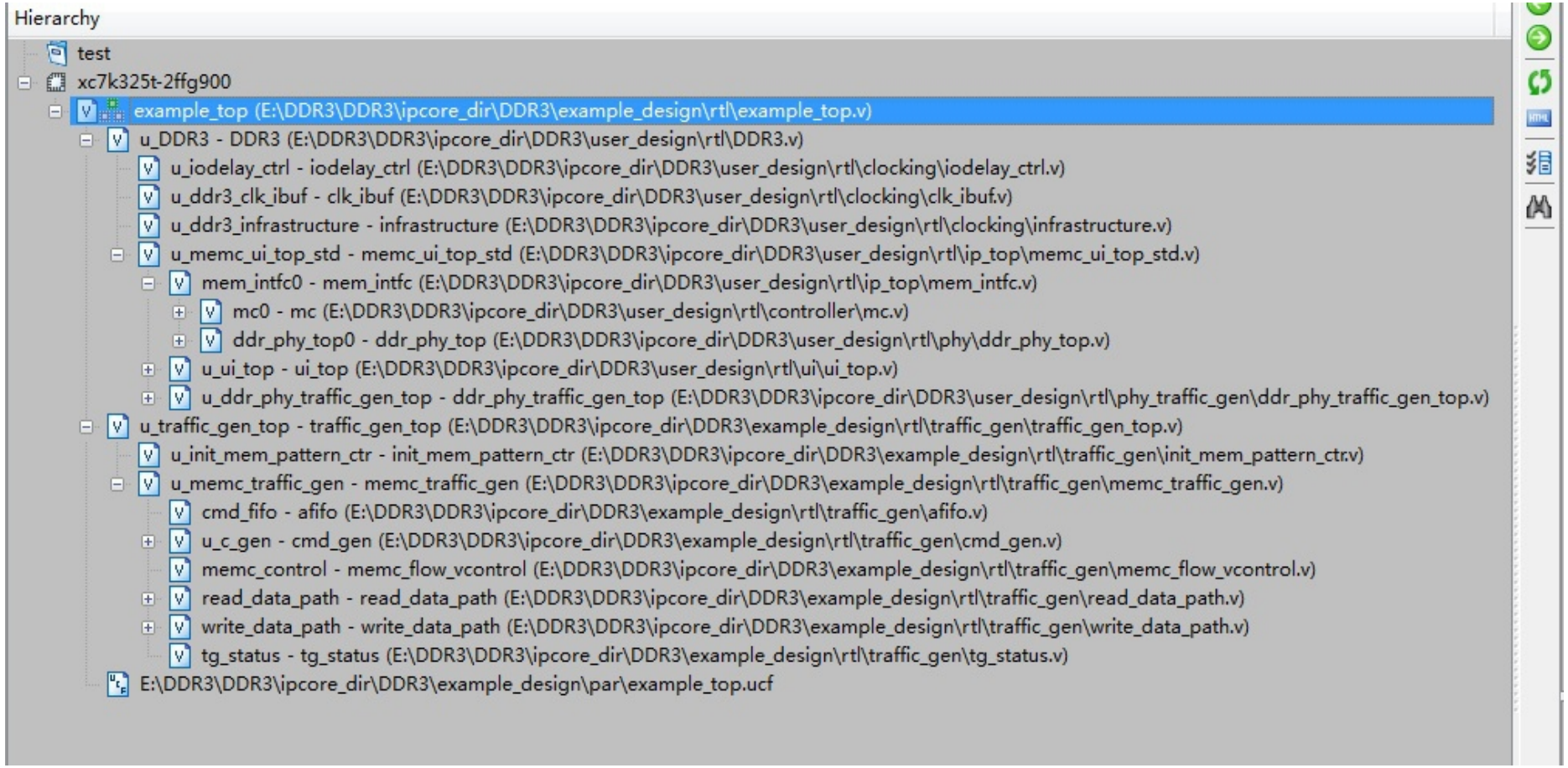
之前仿真教程里面讲过过traffic_gen，现在派用场了

看这个工程里面的顶层文件，不是我写的
其实这工程里面所有的文件都不是我写的

只要你生成好了IP core，管脚分配好（你也可以让core gen自动分配，用默认值就行，还是参见仿真教程）

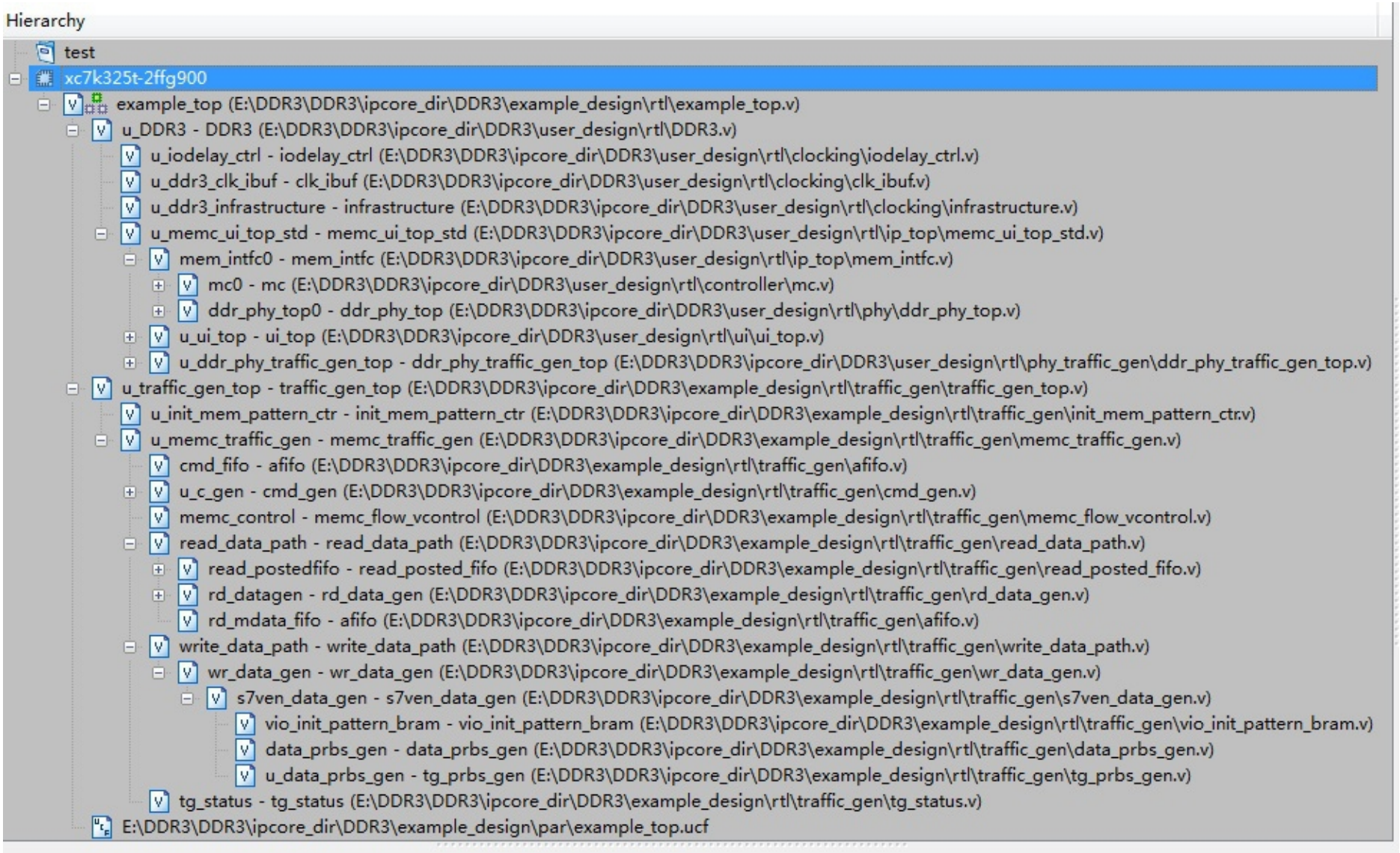
你要做的事，就是这么区区几步：
1，建立一个ISE工程，FPGA型号和封装选对就可以了
2，加入example_design/rtl下面的 example_top.v 以及其他所有.v文件
（什么，你写VHDL的？乖乖，不知道现在写VHDL的很难找工作？）
3，加入user_design/rtl下面的rtl下面的所有.v文件
4，加入example_design/par 下面的example_top.ucf文件

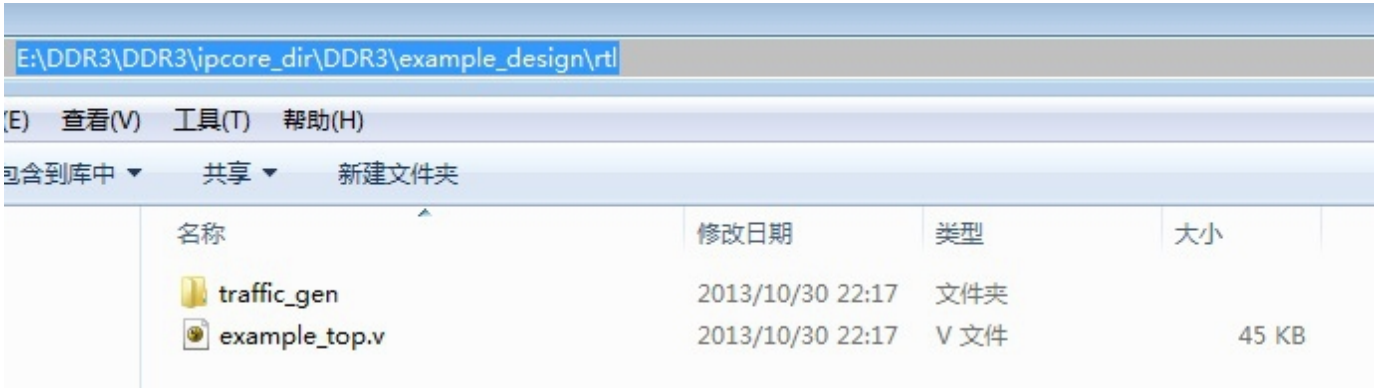
然后你就开始跑编译吧~

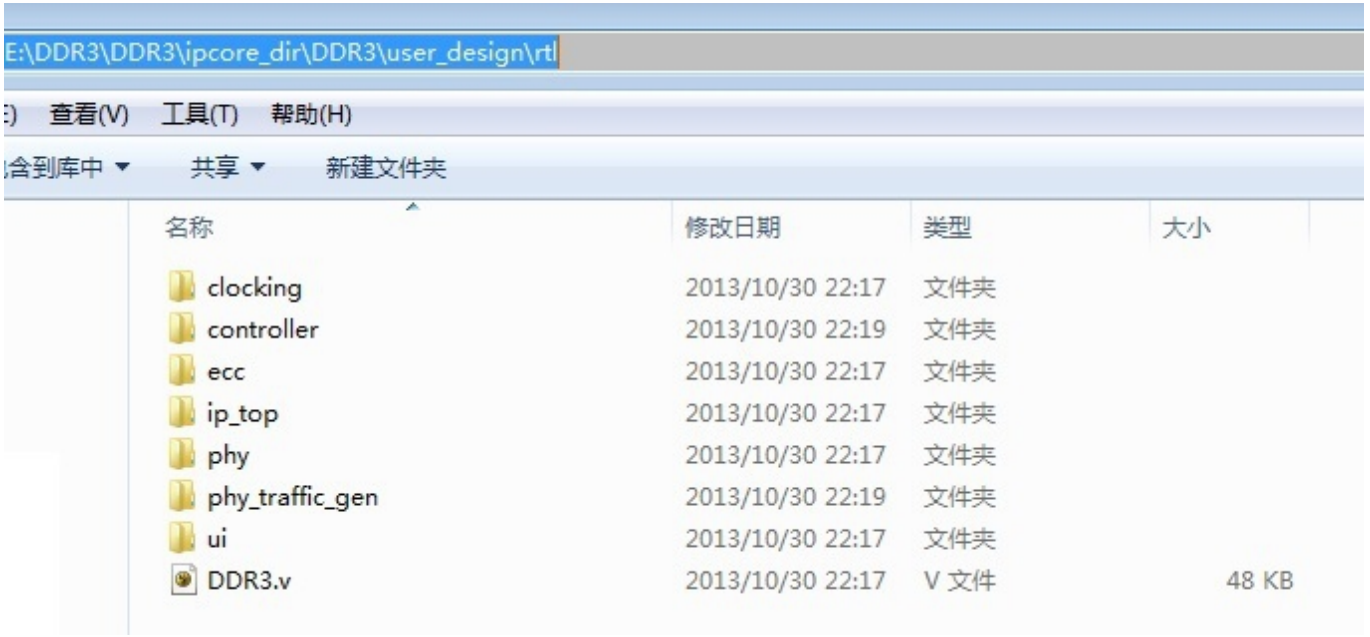


文件全部加完了之后就是这样的，都是绝对路径显示

多展开几级，来个壮观点儿的



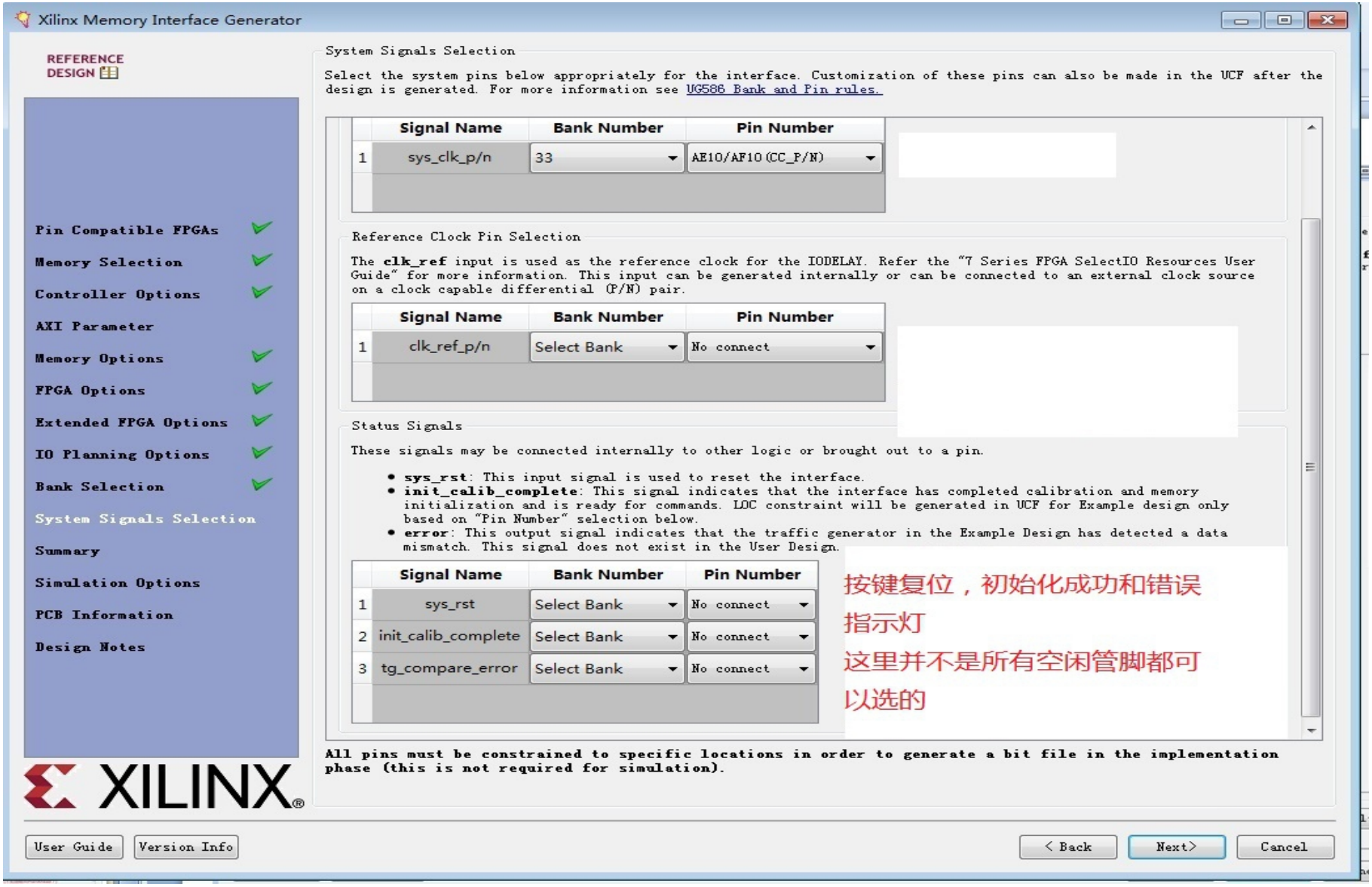




其实就是把这两个目录下的*.v都加进去，最后别忘了加上example_design/par 下面的example_top.ucf文件，就可以了

只要你在core gen的时候分配好了管脚，就不可能编译不成功。

这张图还记得吗？
万一做板子的人把按键和LED灯的管脚给你分配到了这里不能选的地方，
你就得留意下一页的内容了。



http://www.xilinx.com/support/documentation/ip_documentation/mig_7series/v1_5/ug586_7Series_MIS.pdf

在上面这个文档，也就是xilinx MIG的用户手册里面的132页，
有这么一段：

I/O Standards

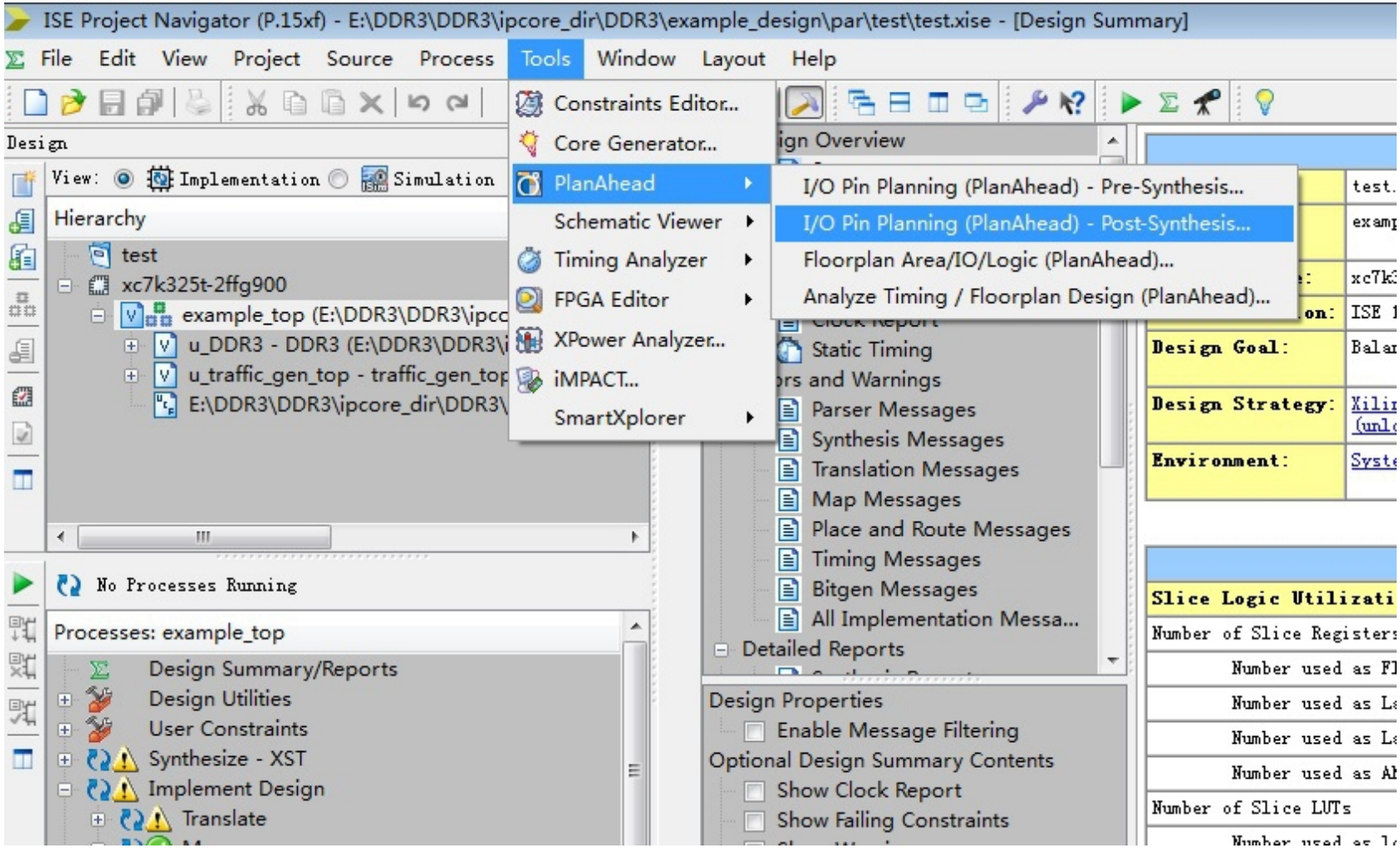
These rules apply to the I/O standard selection for DDR3 SDRAMs:

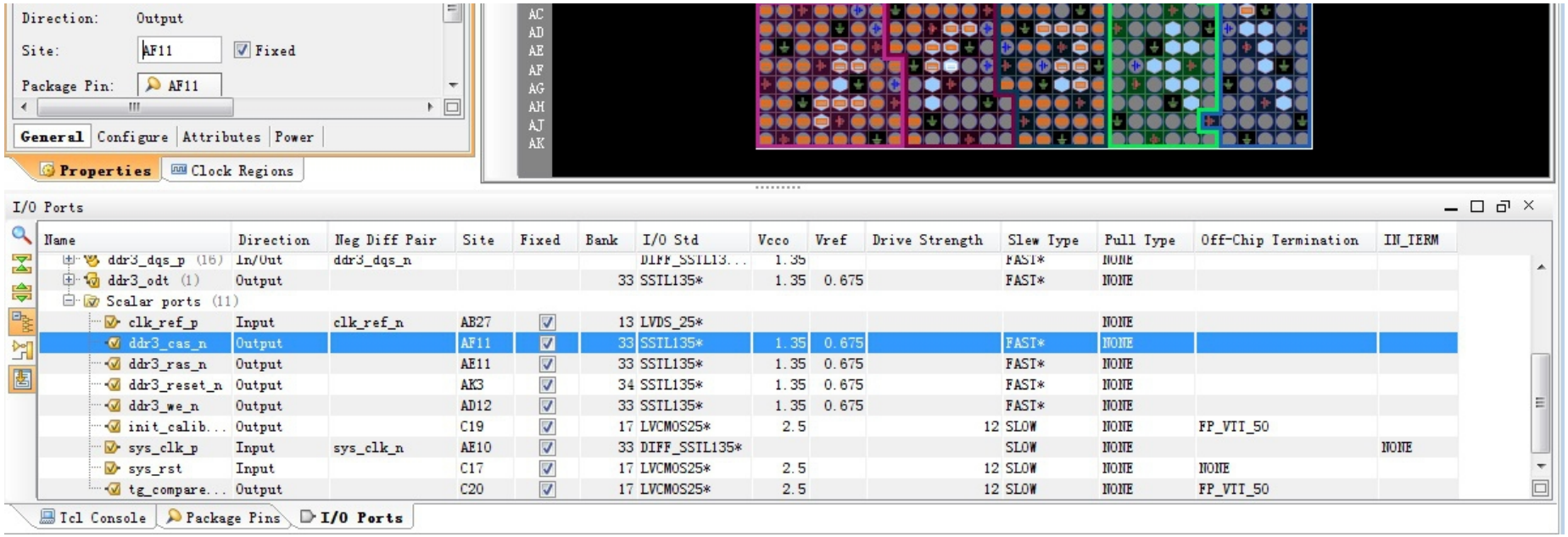
- Designs generated by the MIG tool use the SSTL15_T_DCI and DIFF_SSTL15_T_DCI standards for all bidirectional I/O (DQ, DQS) in the High-Performance banks. In the High-Range banks, the tool uses the SSTL15 and DIFF_SSTL15 standard with the internal termination (IN_TERM) attribute chosen in the GUI.
- The SSTL15 and DIFF_SSTL15 standards are used for unidirectional outputs, such as control/address, and forward memory clocks.
- LVCMOS15 is used for the RESET_N signal driven to the DDR3 memory.

The MIG tool creates the UCF using the appropriate standard based on input from the GUI.

简单说一下要点：
数据管脚应该是SSTL15系列的电平，而复位管脚应该是LVCMOS15系列的电平
其中15表示1.5V

管脚电平可以用planAhead来设置。
推荐你综合完了之后用post-synthesis（综合后）的选项来重设管脚属性，
这个比pre-synthesis（综合前）有说服力，行不行还是得看你综合完了之后不是？





但是实际上你打开了之后会发现不是那么回事儿
这为啥LVCMOS都是2.5V，SSTL都是1.35V，编译也通过了？
可能待会儿上电了下载还正常跑。

那我告诉你，这个例子里SSTL都是1.35V，跟core gen的时候选的sodimm条子款式有关，有些条子就是1.35V的，比如我这里选的这根儿就是。

电压或许可以和1.5V有所区别，但是SSTL和LVCMOS的区别不能搞混，否则可能map不过去。

SSTL和LVCMOS有啥区别的呢

SSTL是针对DDR之类的高速管脚的电平

LVCMOS就是常用的默认电平

只要知道，DDR的高速打数据，尤其是双沿动作的管脚，都得是SSTL的电平；而按键指示灯之类的慢速控制管脚，应该是lvcmos电平，就可以了。

如果你在planAhead里面选择好管脚属性保存之后没有立即提示错误，也不代表你后面map和par就能通过。

DDR设计的原则之一就是，别去改哪些你想当然认为肯定没事的东西。

先全部用默认值生成一个DDR工程，编译通过，保存好，然后再在这个工程里改动。在编译出问题的时候，仔细核对你改动的部分和原先的初始工程有什么区别，项目才能比较顺利的做下去。

最后大致讲一下traffic gen

这个貌似DDR2里面没有
只有DDR3有

功能之前讲过了，就是写数据进DDR，读出来看和写进去的是不是一致，不一致的话就报错，拉高tg_compare_error

很不幸的是，这东西虽然用了DDR core用户接口，但是代码写得基本没有通用性，你不可能在这个基础上修改修改就拿来自己用。

所以要做DDR设计，你得自己重新写一个控制DDR用户接口的代码。

最典型的例子就是从一个fifo里面读数据，然后写进DDR

这个例子我之后再讲了。

现在，先动手建一个工程吧。

在ISE14.2和以上版本，可以直接进example_design\par文件夹，
运行create_ise.bat，然后工程会自动生成。
其实这个过程和你手动建立工程是一样一样的，聊胜于无吧，嘿嘿~