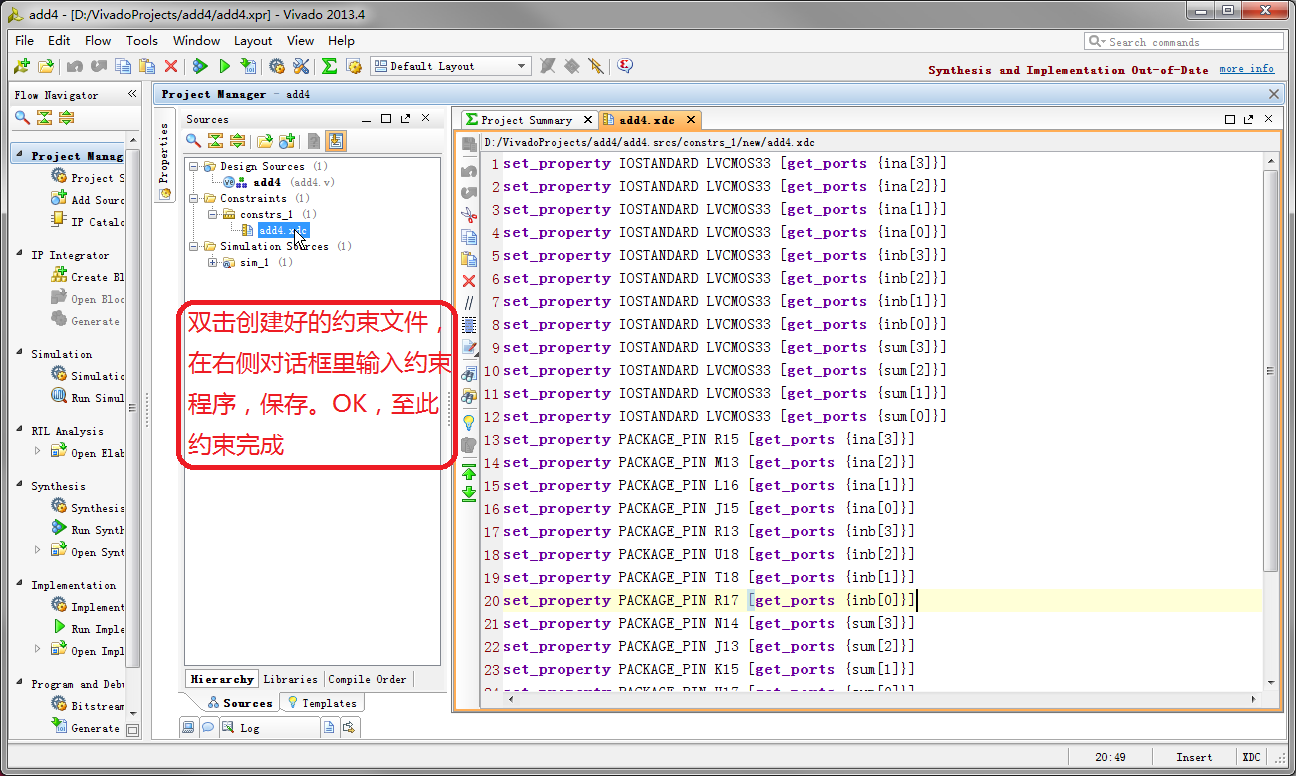
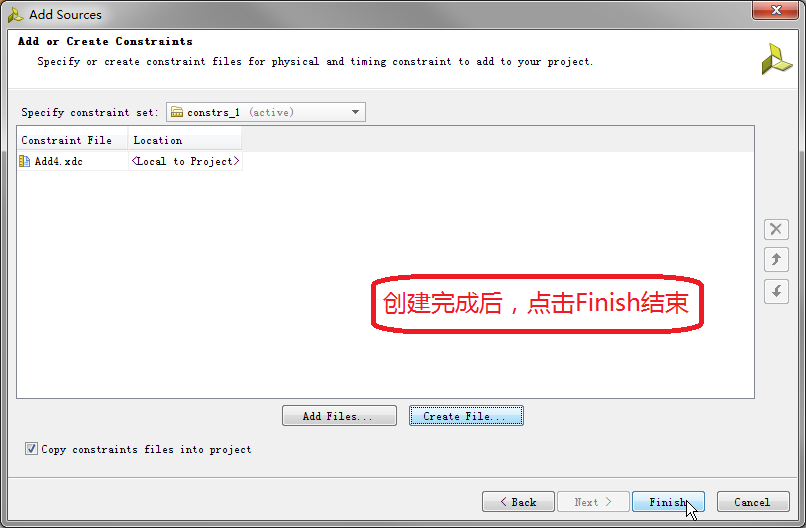
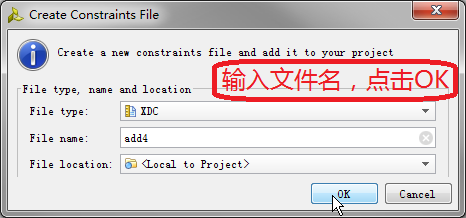
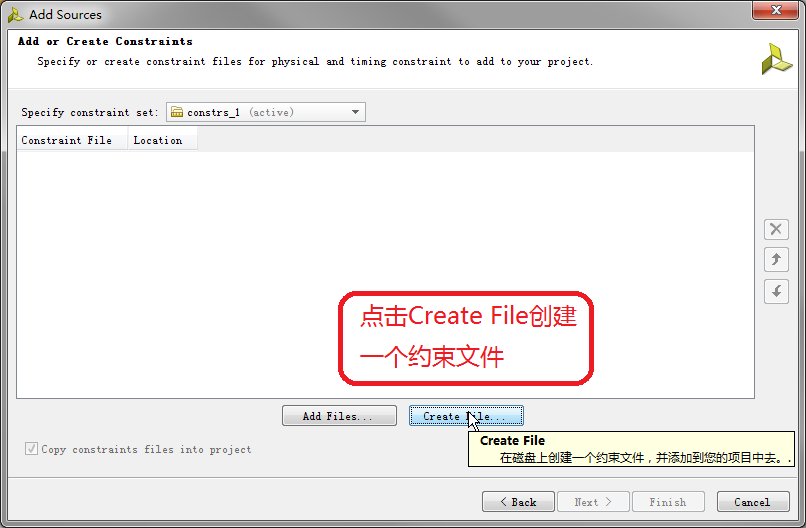
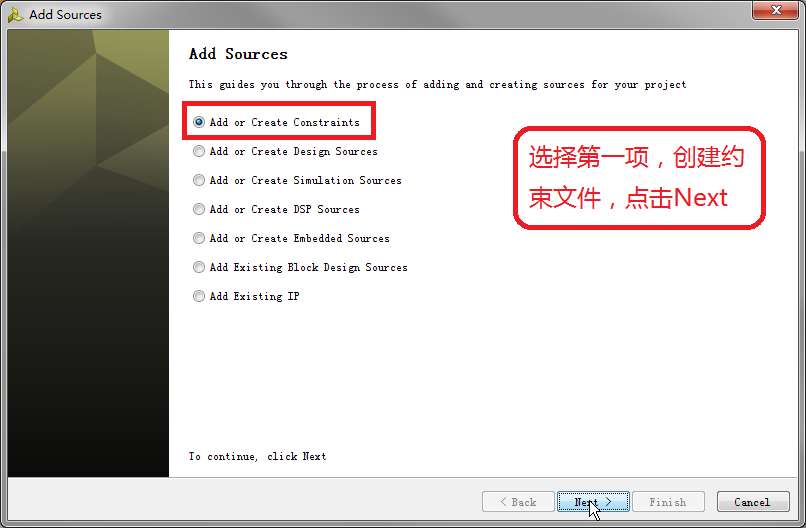
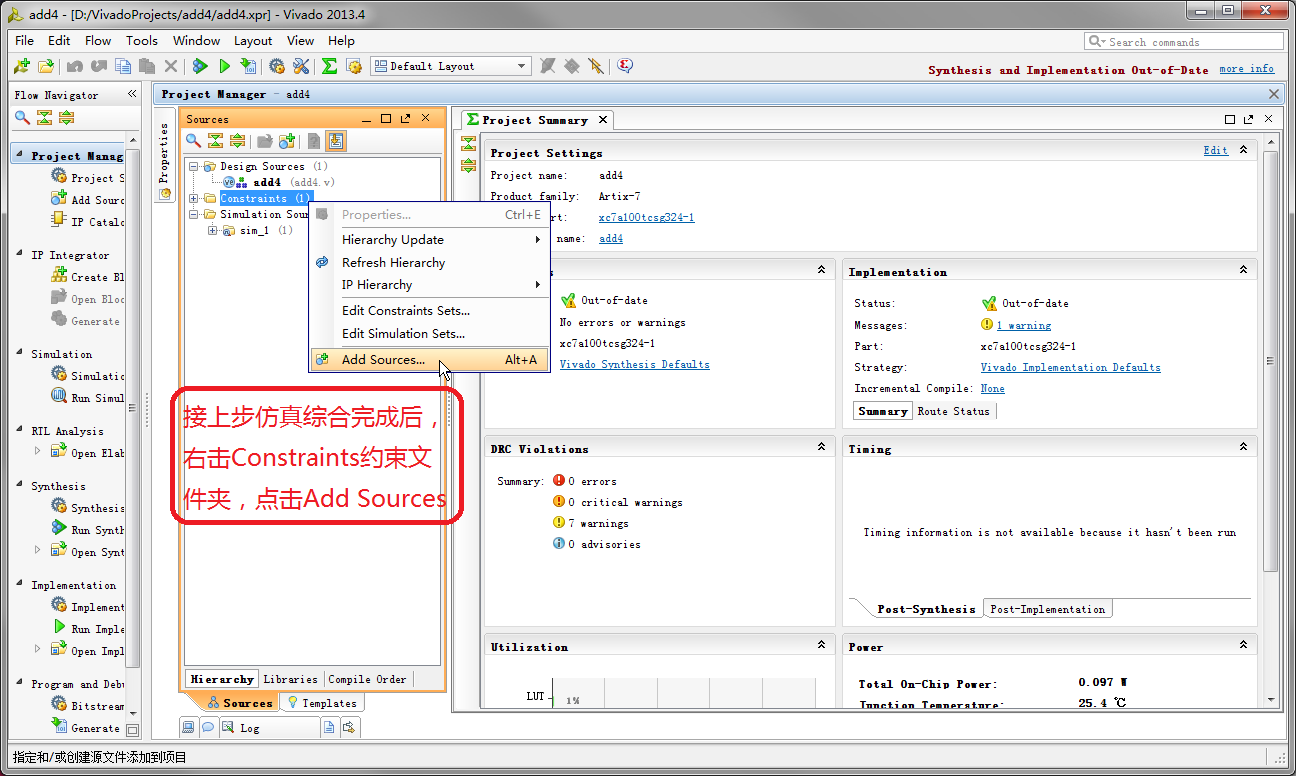
PS: 由于下面用到约束管脚的另外一种方法，故开头介绍，所有步骤如下图：（下板操作才需要这一步）



1. 程序文件源码+注释分析

module add4(

input [3:0] ina, //定义输入数据接口

input [3:0] inb, //定义另一个输入数据接口

input cin, //定义初始进位C0，即上一个数的进位

output cout, //定义进位Cn

output [3:0] sum //定义输出数据接口

);

assign {cout, sum}=ina+inb+cin; //计算公式表示为：XXXXX（第一个二进制数为进位，后四个为计算结果） = XXXX（数据A） + XXXX（数据B） + X（上一个数的进位，即初始进位控制端）

endmodule

1. 测试文件源码+注释分析

`timescale 1ns / 1ps

module add4\_text(); //测试文件提供模拟的输入数据和输出

reg [3:0] ina; //将输入定义为寄存器型变量

reg [3:0] inb;

reg cin;

wire cout; //将输出定义为连线型变量

wire [3:0] sum;

initial begin

ina = {$random} % 16; //提供0~15之间的随机数作为输入数据

inb = {$random} % 16;

cin = {$random} % 2; //提供随机0或1作为输入数据

repeat(10) begin //将begin和end之间的语句重复执行10次

#100

ina = {$random} % 16;

inb = {$random} % 16;

cin = {$random} %2;

end

#1000 $stop;

end

add4 ADD4(ina,inb,cin,cout,sum); //需要测试的程序文件的调用，注意参数顺序应与程序文件保持一致

endmodule

1. 约束文件源码+注释分析（下板操作才需要这一步）

set\_property IOSTANDARD LVCMOS33 [get\_ports {ina[3]}] #提供输入/输出标准和电压信息

set\_property IOSTANDARD LVCMOS33 [get\_ports {ina[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {ina[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {ina[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {inb[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {inb[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {inb[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {inb[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sum[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sum[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sum[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sum[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports cin]

set\_property IOSTANDARD LVCMOS33 [get\_ports cout]

set\_property PACKAGE\_PIN R15 [get\_ports {ina[3]}] #提供程序定义的输入/输出和电路板上的引脚关联的信息

set\_property PACKAGE\_PIN M13 [get\_ports {ina[2]}]

set\_property PACKAGE\_PIN L16 [get\_ports {ina[1]}]

set\_property PACKAGE\_PIN J15 [get\_ports {ina[0]}]

set\_property PACKAGE\_PIN R13 [get\_ports {inb[3]}]

set\_property PACKAGE\_PIN U18 [get\_ports {inb[2]}]

set\_property PACKAGE\_PIN T18 [get\_ports {inb[1]}]

set\_property PACKAGE\_PIN R17 [get\_ports {inb[0]}]

set\_property PACKAGE\_PIN N14 [get\_ports {sum[3]}]

set\_property PACKAGE\_PIN J13 [get\_ports {sum[2]}]

set\_property PACKAGE\_PIN K15 [get\_ports {sum[1]}]

set\_property PACKAGE\_PIN H17 [get\_ports {sum[0]}]

set\_property PACKAGE\_PIN V10 [get\_ports cin]

set\_property PACKAGE\_PIN V11 [get\_ports cout]

1. 仿真实验结果如下图，可以看出sum的结果为ina、inb、cin之和，若数据大于16，则产生进位，cout用以存储进位

