

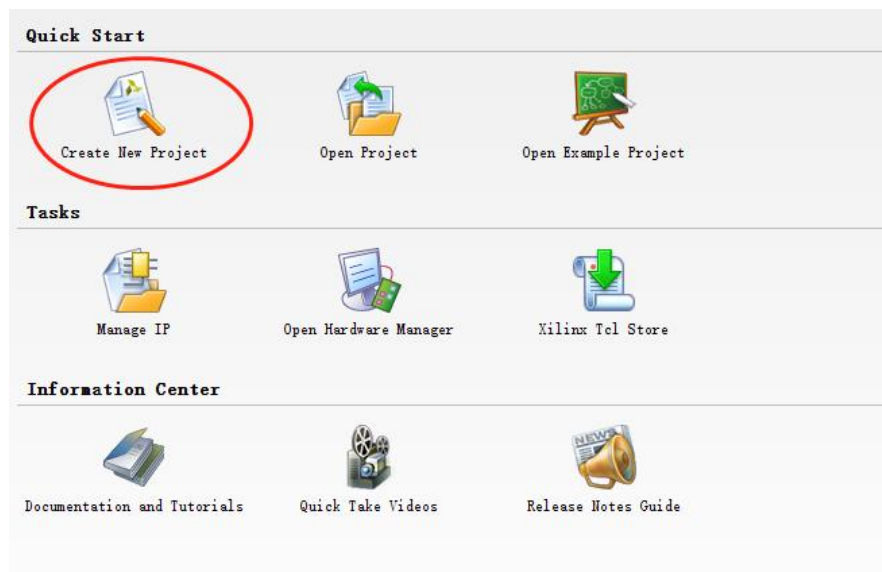
Vivado的使用

1 Vivado 设计流程

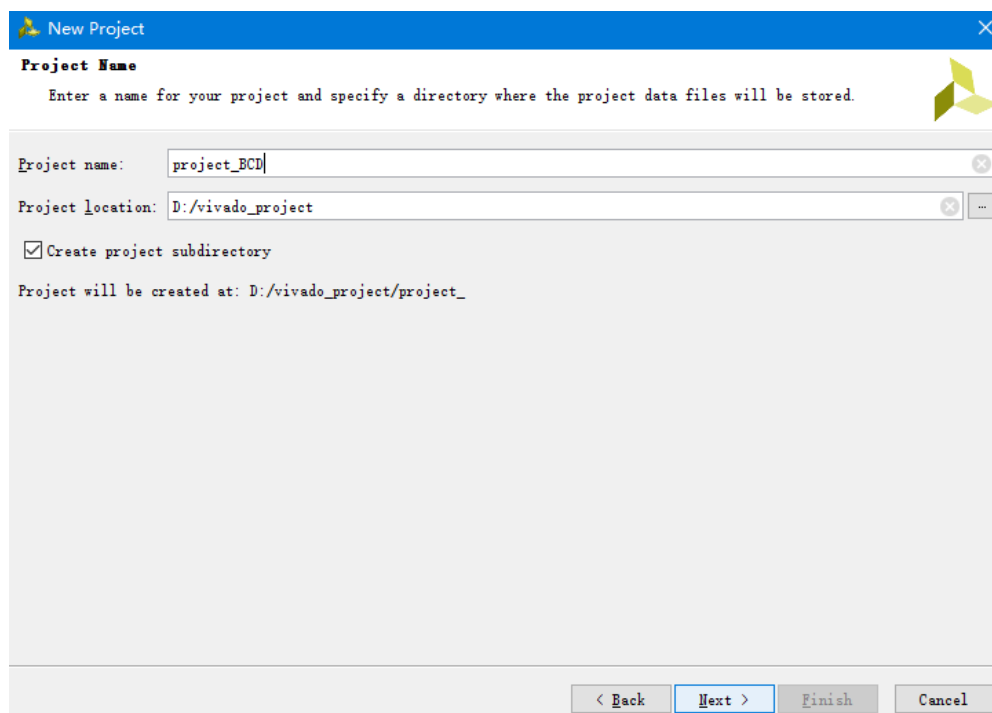
本章以 Vivado 2014.3.1 版本为例，对 Vivado 进行演示实验。

1.1 新建工程

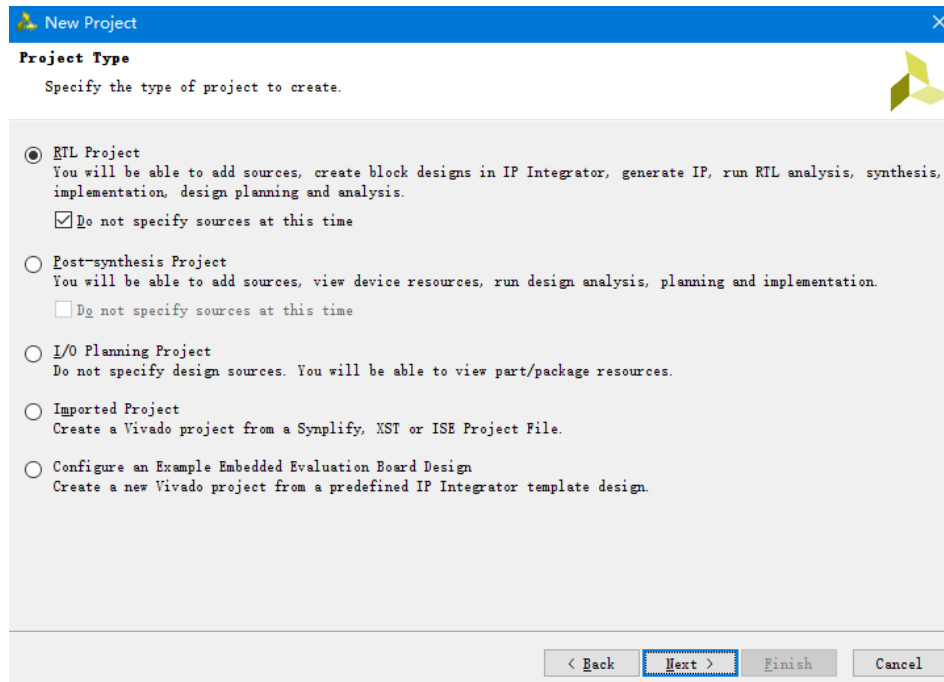
打开 Vivado，选择 Create New Project



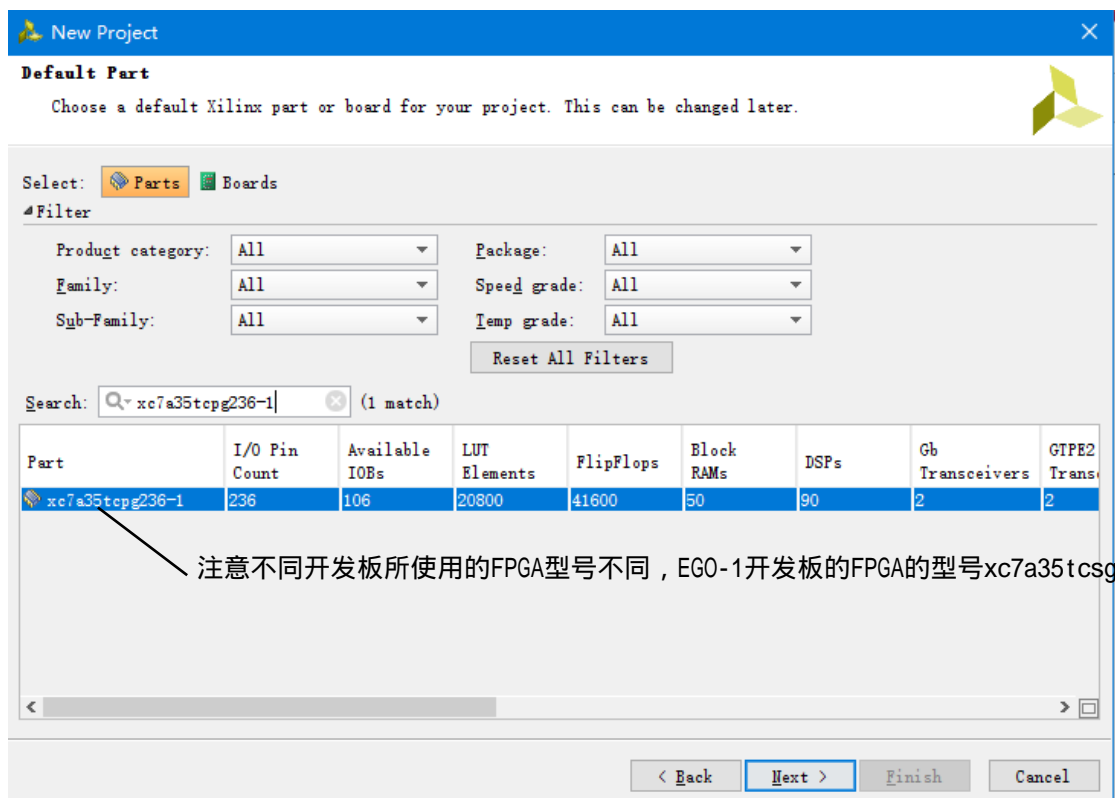
选择 next，进入界面如下：



对工程命名为 project_BCD，路径为 D:/vivado_project。注意工程名和路径名不要有中文和空格。选择 next，进入界面如下



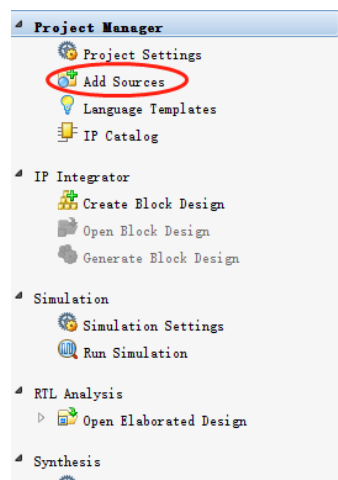
选择 RTL project，并勾选 do not specify sources at the time. 点击 next 进入如下界面



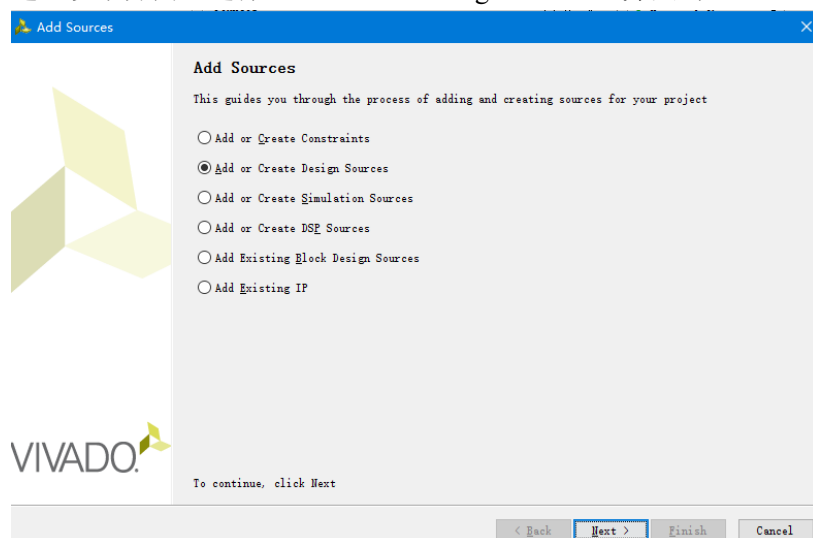
选择 FPGA 的对应型号。对于 EGO1 实验平台，选择型号 xc7a35tcsg324-1. 点击 Next，并点击 Finish 完成工程创建。

1.2 添加源文件（以七段译码器为例）

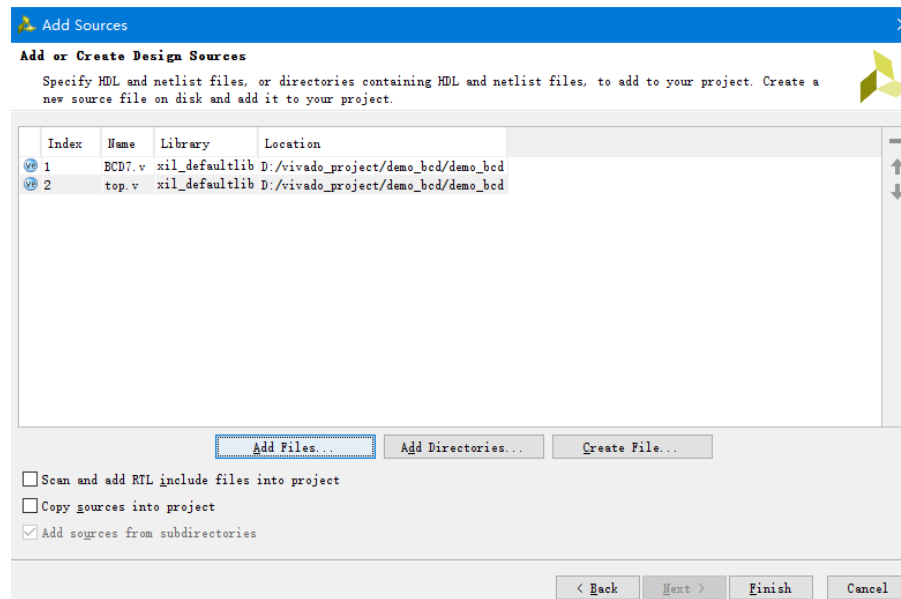
创建工程以后，在界面左侧的 Add Sources 处添加源文件



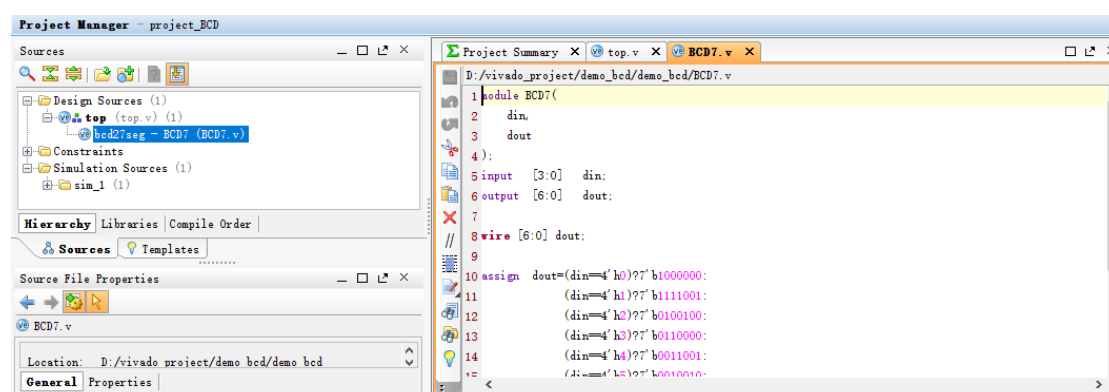
进入以下界面，选择 Add or create design sources。并点击 Next



进入以下界面。这里将已经写好的源文件 BCD7.v 与 top.v 加入如下。点击 Finish

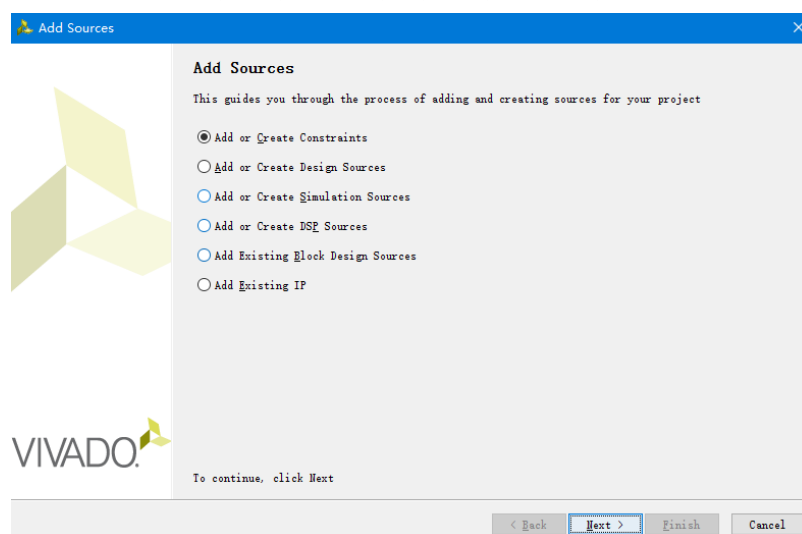


此时，可以通过主界面看到，已经添加的 BCD7.v 和 top.v 在右侧显示。

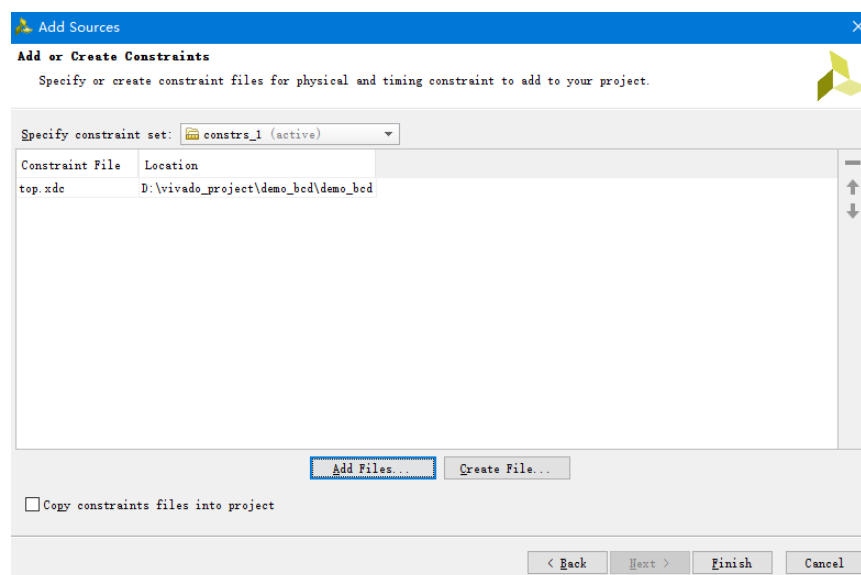


1.3 添加约束文件

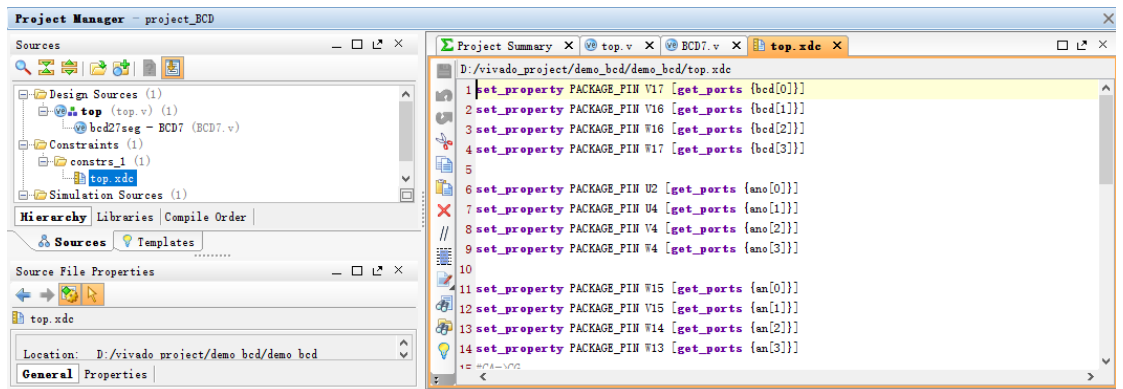
然后需要添加约束文件来绑定管脚。再次点击左侧的 Add Sources，进入如下界面



选择 Add or Create Constraints，点击 Next 进入如下界面



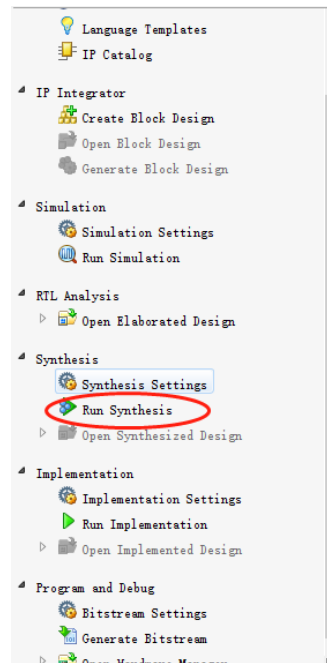
通过 Add Files 添加 top.xdc 文件。选择 Finish，即完成约束文件添加。此时可以在主界面看到已经添加的 top.xdc 文件如下



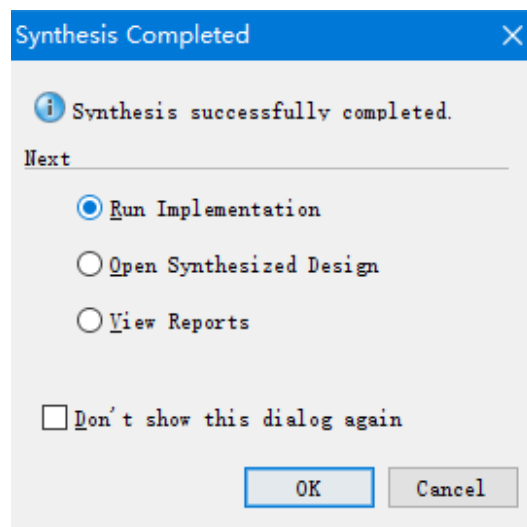
可以直接在该界面内修改不同输入输出端口与 FPGA 管脚的绑定关系。

1.4 工程综合实现

以上文件添加完成后，需要对工程进行综合和实现。首先，在左侧 Synthesis 下拉菜单中选择 Run Synthesis。



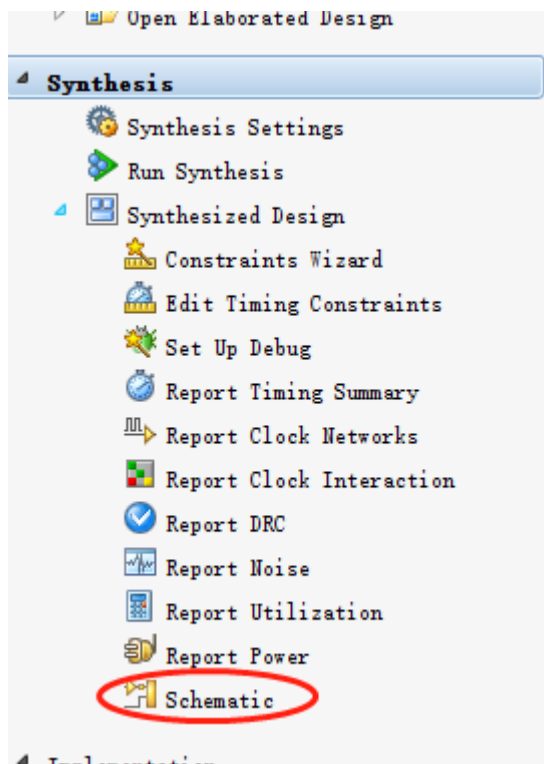
综合完成，弹出如下界面



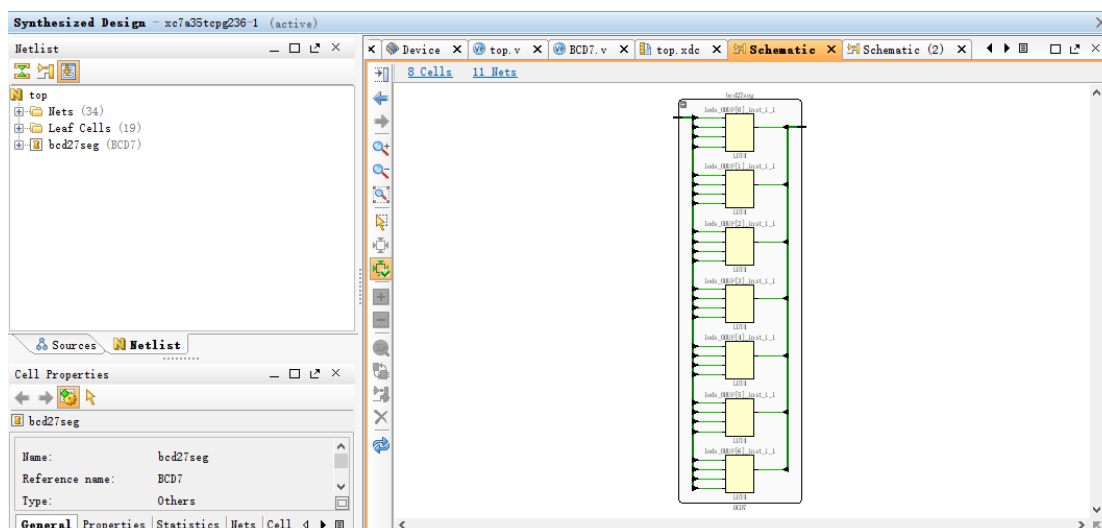
选择 Run Implementation，并点击 OK，将已经综合好的工程变成可以在硬件上实现的代码。综合完成以后，弹出界面，并选择 **Generate Bitstream** 等待生成比特流完成。然后连接 FPGA，可以将比特流下载至 FPGA 运行。

2 读 LUT

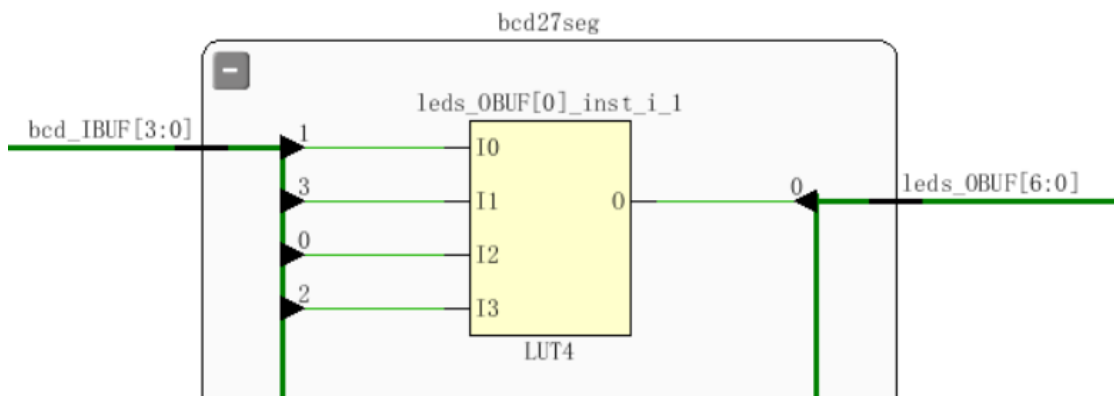
综合以后，可以通过选择 Synthesis 下的 Schematic 查看该代码综合后的电路结构图。



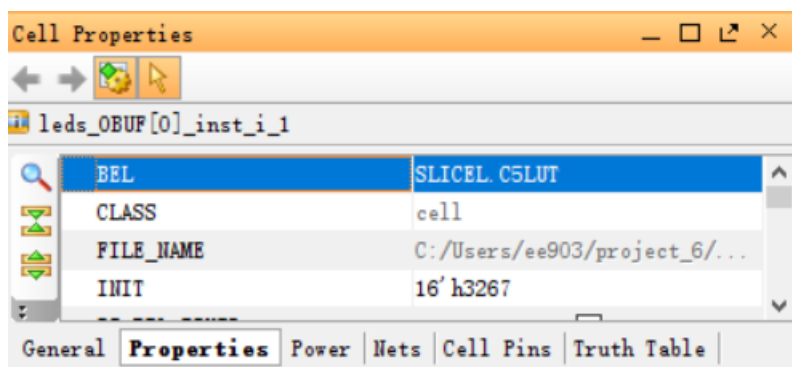
如下所示：



BCD7 内部由 7 个 LUT 构成。



放大发现，每个 LUT 均为一个 4 输入 1 输出的查找表。通过 `bcd_IBUF[3:0]` 进行输入，作为 RAM 的地址，然后输出该地址对应的内容 0 或 1。通过 0 和 1 不同的排列，可以在 FPGA 的七段译码器上显示不同的数字。



以上述 LUT 为例，右键选中该 LUT，选择 Cell Properties 选项，看到 16 位的对应地址输出为 `16'h3267`，转换为二进制即 `16'b0011_0010_0110_0111`。

例如，当 `din=4'h4` 时，其对应的 `bcd_IBUF[3:0]=2'b0100`。由于该 LUT 输入时，有对应 `I0=bcd_IBUF[1]`，`I1=bcd_IBUF[3]`，`I2=bcd_IBUF[0]`，`I3=bcd_IBUF[2]`，故其地址输入值 `I=4'b1000`，即应该查找 `16'h3267` 从最低位（第 0 个）开始的第 8 个输出值，即输出为 0。该 LUT 的输出作为最终 7 位输出的最低位，与 verilog 代码 `(din==4'h4)?7'b1100110` 一致。

例如，当 `din=4'h6` 时，其对应的 `bcd_IBUF[3:0]=2'b0110`。此时有 `I=4'b1001`，即应该查找 `16'hABA4` 从最低位（第 0 个）开始的第 9 个输出值，为 1，与 verilog 代码 `(din==4'h6)?7'b1111101` 一致。

同学们可采用不同的输入数字，对每个 LUT 的输入输出进行验证。