

“实验一 计数器和序列检测器” 提示

1. 代码的风格

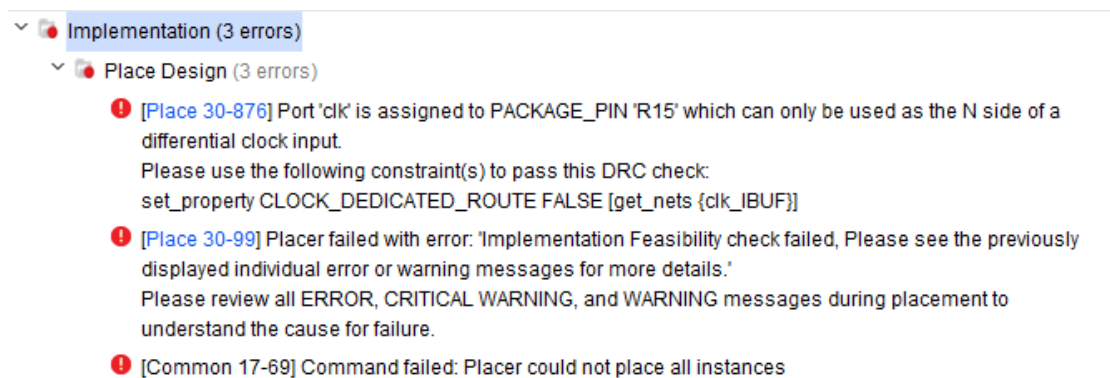
在实验过程中，有的同学已经遇到了语法正确，但是综合后烧写到板子上结果却不正确的情况。排除设计错误的前提下，其最大的可能性就是在 Verilog 的编写中，没有遵循可综合的 Verilog 代码风格。

代码风格一个很好的手册是 Xilinx 公司针对我们所用的 Vivado 工具给出的用户手册 UG901。这个文档我们已经上传到网络学堂的课程文件中，大家可以参考。移位寄存器的例子在 78 页，有限状态机的例子在 164 页。有限状态机用我们课件中的两段式也是可以的。

如果遇到问题，建议查看综合实现输出的 Message 中的 Warning 信息，有助于问题的定位与解决。

2. 时钟管脚的问题

用按键连接引脚直接作为时钟输入，Vivado 会报出如下错误：



原因正如在这个错误消息中所说，FPGA 对时钟的引入管脚有特殊的限制，按键连接的这个引脚不能够接入时钟分配网络。

解决办法在消息中，其实已经给出：

在约束文件中，加入

```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {clk_IBUF}]
```

或者

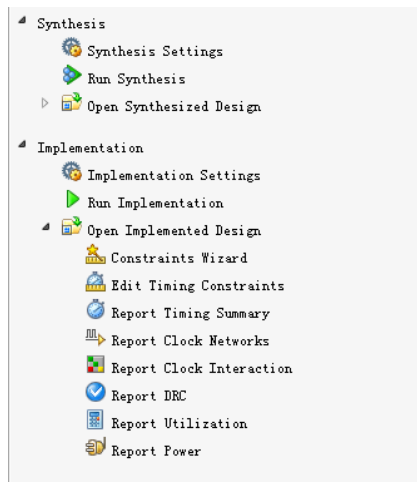
```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {clk}]
```

3. 占用资源的查看

从实验一开始，我们在实验报告中，要求报告所设计的电路的资源占用情况。以下面代码为例，顶层由设计的 FSM（有限状态机）构成。

```
fsm x fsm(.clk(clk), .reset(reset), .data(data), .result(out), .cond(code));
```

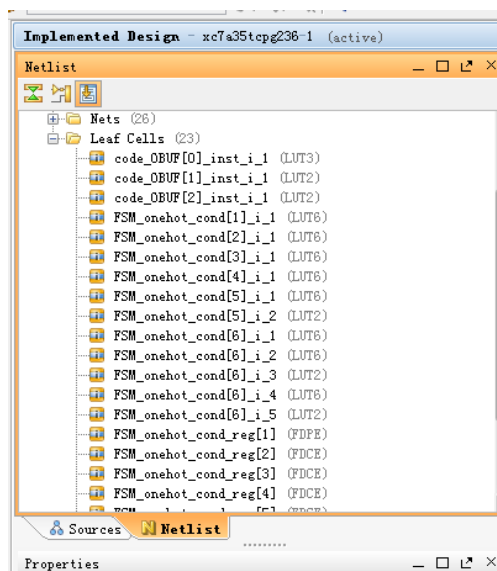
我们要分析我们设计的 FSM 的资源占用情况。在 Implementation 完成后，点击 Report Utilization 生成面积占用情况报告。



资源占用情况包括组合逻辑（在 FPGA 用查找表 LUT 实现）和寄存器。点开 Slice Logic 中的 Slice LUTs 可以看到 x fsm 占用了 13 个查找表，点开 Slice Registers 可以看到 x fsm 占用了 7 个寄存器。

Name	Used
mealy	21
x fsm (fsm)	13
xdebounce (debounce)	8

占用的寄存器比预想的多。我们 Open Implemented Design 在 Netlist 中，看到寄存器的名字后面加了 onehot，猜想 Vivado 使用 one-hot 编码优化了状态机。



我们可以查阅 Vivado Synthesis Report，确认其编码方式。

Name	Modified	Size	GUI Report
Synth Design (synth_design)			
Vivado Synthesis Report	4/21/17 5:42 PM	19.8 KB	
Utilization Report	4/21/17 5:42 PM	6.1 KB	
Design Initialization (init_design)			
Timing Summary Report			
Opt Design (opt_design)			
Post opt_design DRC Report			
Timing Summary Report			
Power Opt Design (power_opt_design)			
Timing Summary Report			
Place Design (place_design)			
Vivado Implementation Log	4/21/17 5:43 PM	20.8 KB	

Tcl Console Messages Log Reports Package Pins Design Runs Utilization

其中确有如下内容：

INFO: [Synth 8-802] inferred FSM for state register 'cond_reg' in module 'fsm'

INFO: [Synth 8-3354] encoded FSM with state register 'cond_reg' using encoding 'one-hot' in module 'fsm'