

FPGA 入门系列实验教程——LED 流水灯

1. 实验任务

让实验板上的 8 个 LED 实现流水灯的功能。通过这个实验，进一步掌握采用计数与判断的方式来实现分频的 VHDL 的编程方法以及移位运算符的使用。

2. 实验环境

硬件实验环境为艾米电子工作室型号 EP2C8Q208C8 增强版开发套件。

软件实验环境为 Quartus II 8.1 开发软件。

3. 实验原理

流水灯，顾名思义就是让 LED 象水一样的点亮。如果把流水做慢动作播放，可以想象到其实就是移动，即：把水块不断地向同一方向移动，而原来的水块保持不动，就形成了流水。同样，如果使得最左边的灯先亮；然后，通过移位，在其右侧的灯，由左向右依次点亮，而已经亮的灯又不灭，便形成了向右的流水灯。初始状态时，8 个灯都不亮。每来一个时钟脉冲 CLK，计数器就加 1。每当判断出计数器中的数值达到 25000000 时，就会点亮一个灯，并进行移位。FPGA 输出的数据就应该首先是 10000000，隔 1 秒钟变成 11000000.....一直变化到 11111111，这样，依次点亮所有的灯，就形成了流水灯。而当 8 个灯都点亮时，需要一个操作使得所有的灯恢复为初始状态，即：灯都不亮。然后，再一次流水即可。如果是右移位，就出现向右流水的现象；反之，向左流水。

4. 实验程序

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity LED is
port(clk:in std_logic; //设置时钟输入
      dout:out std_logic_vector(7 downto 0)); //8 位端口输出
end LED;

architecture behave of LED is
signal p:std_logic_vector(31 downto 0); //信号变量
signal t:std_logic_vector(8 downto 0);
begin
```

```
process(clk)
begin
if(clk'event and clk='1')then //判断上升沿信号
    p<=p+1;
    if(p=25000000)then
        t(8 downto 1)<=t(7 downto 0); //将低 8 位移至高 8 位
        t(0)<='1'; //最地位置 1
        p<="00000000000000000000000000000000";
    end if;
    if(t(8)='1')then
        t<="000000000";
    end if;
end if;
end process;
dout<=t(7 downto 0); //将低 8 位赋给输出端口
end behave;
```

代码分析:


先定义一个二进制 32 位的信号变量 P 和一个 9 位的信号变量 t，时钟每产生一个上升沿的时钟跳变，P 自动加 1。当 P 加到 25000000 时将 t 的低 8 位移到 t 的高 8 位，同时将 t 的最地位置 1，P 复位置 0，P 又将从 0 开始计数。当 t 的最高位变为 1 时将 t 复位置 0 从头开始。在程序执行过程中始终将 t 的低 8 位赋值给数据输出端 dout。

5. 实验步骤

(1) 建立新工程项目:

打开 Quartus II 软件，进入集成开发环境，点击 File→New project wizard 建立一个工程项目 ledwater。


(2) 建立文本编辑文件:

点击 File→New 在该项目下新建 VHDL 源程序文件 ledwater.v，输入试验程序中的源程序代码保存后选择工具栏中的  按钮启动编译，若在编译中发现错误，则找出并更正错误，直到编译成功为止。

(3) 选择器件型号及引脚的其他设置:

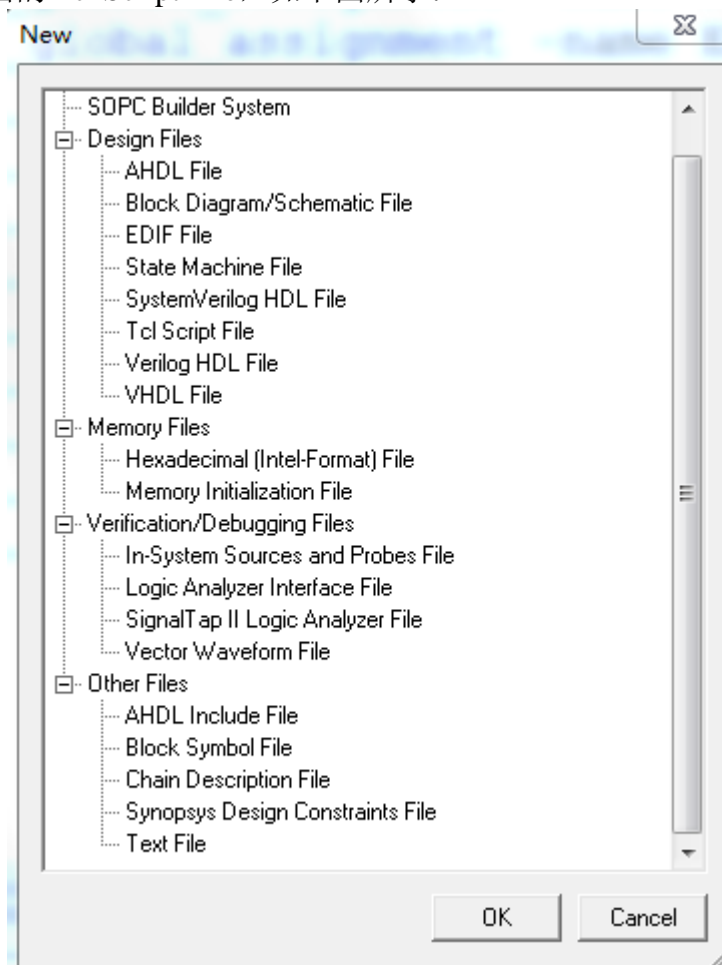
选择所用的 FPGA 器件→ EP2C8Q208C8, 以及进行一些配置。选择配置器件 EPCS4, 设置不需要使用的 IO 功能为 As inputs, tri-stated。点击两次 ok, 回到主界面。

(4) 配置 FPGA 引脚:

在 Quartus II 软件主页面下, 选择 Assignments→ Pins 或选择工具栏上  按钮, 配置 dout[0]---dout[7]以及 clk 的引脚。

上述配置引脚的方法适合引脚比较少的项目, 对于引脚比较多的工程项目, 分配起来就比较麻烦了, 下面介绍一种方便快捷的引脚分配方法。

- a. 首先建立一个 TCL Script 文件: 点 File→ New 选择 Design Files 选项卡里面的 Tcl Script File, 如下图所示:



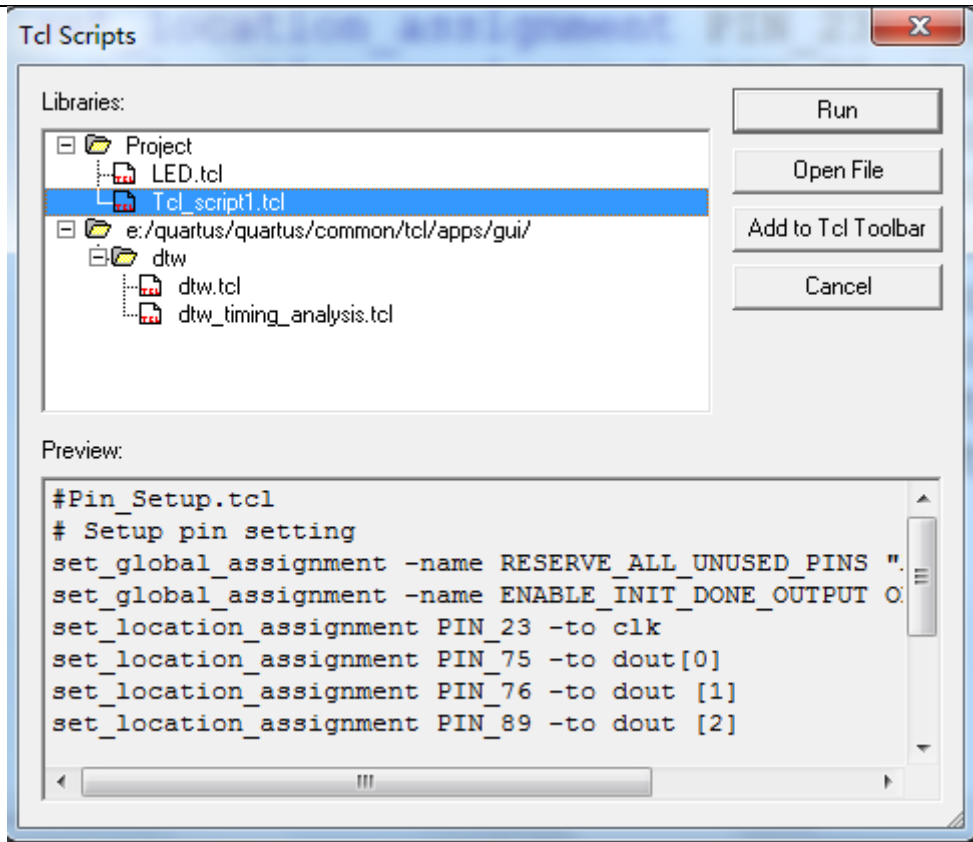
b. 点击 ok，输入下面的代码：

```
#Pin_Setup.tcl
# Setup pin setting
set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT
TRI-STATED"
set_global_assignment -name ENABLE_INIT_DONE_OUTPUT ON
set_location_assignment PIN_23 -to clk
set_location_assignment PIN_75 -to dout[0]
set_location_assignment PIN_76 -to dout [1]
set_location_assignment PIN_89 -to dout [2]
set_location_assignment PIN_94 -to dout [3]
set_location_assignment PIN_92 -to dout [4]
set_location_assignment PIN_97 -to dout [5]
set_location_assignment PIN_102 -to dout [6]
set_location_assignment PIN_101 -to dout [7]
```


比较简单的代码，简单的讲解一下，第一行和第二行是注释，第四行的意思是设置不用的引脚为三态输入，第五行的意思是打开 INIT_DONE 输出。后面的几行分别是锁定 clk 和 8 个 led 的引脚。

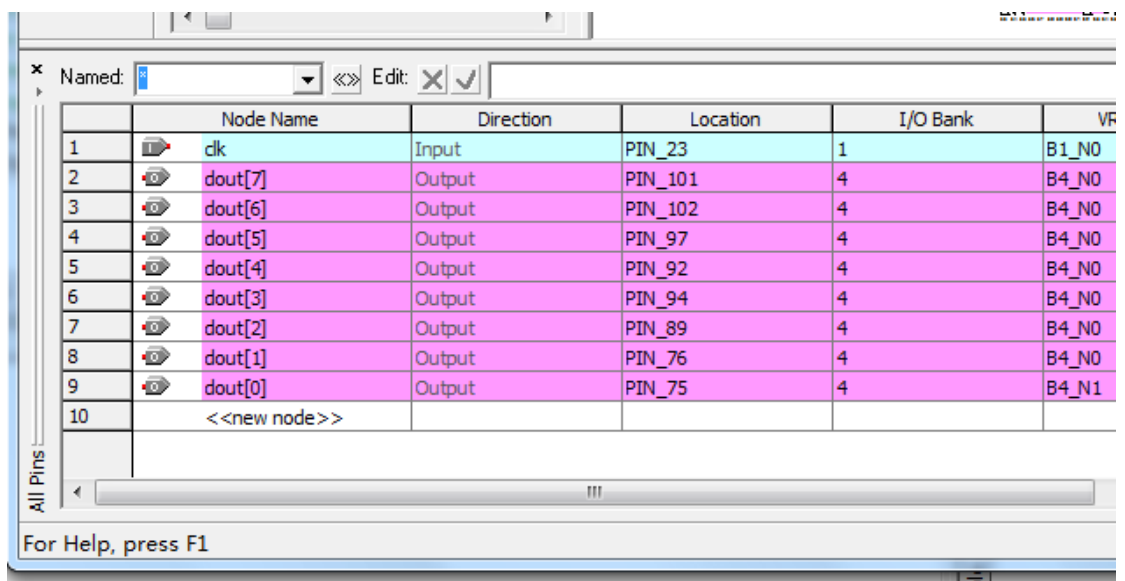
c. 在 Quartus II 主界面中选择 File→Save As 保存这个文档，软件会自动把它加到项目里面。

d. 在 Quartus II 主界面中选择点 Tools->TCL Scripts...，出现下图所示这个界面：




e. 选中你刚才建好的 Tcl 文件，点击 Run。

f. 你可以选择 Assignments → Pins 或选择工具栏上  按钮确认一下是否正确分配了，分配的情况如下图：



怎么样这种方法是不是比以前的方法快多了!这个文件你可以适当修改用于别的项目,如果引脚一样也可以直接在别的项目中加入这个文件然后执行,可以省很多事情。

(5) 编译工程项目:

在 Quartus II 主页面下,选择 Processing —Start Compilation 或点击工具栏上的  按钮启动编译,直到出现“Full Compilation Report”对话框,点击 OK 即可。

(6) 波形仿真: 由于本次试验比较简单,波形仿真将在后面实验详细讲解。

(7) 下载设计程序到目标 FPGA

6. 实验现象

经过前两个实验的训练,第三个实验应该很轻松地就做完了吧!看到实验板上 LED 实验了流水灯的功能,你是否掌握了移位运算符的使用,你可以试着自己编写其他花样的流水灯,比如左流水或其他花样的流水灯,这就要看你的想象力了。