

FPGA 入门系列实验教程——LED 闪烁灯

1. 实验任务

让实验板上的 8 个 LED 周期性的闪烁。通过这个实验，熟悉并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 Quartus II 软件的使用方法和开发流程。

2. 实验环境

硬件实验环境为艾米电子工作室型号 EP2C8Q208 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

艾米电子工作室开发套件板载 50MHz 的时钟源，假如我们直接把它输入到发光二极管 LED，由于人眼的延迟性，我们将无法看到 LED 闪烁，认为它一直亮着。如果我们期望看到闪烁灯，就需要将时钟源的频率降低后再输出。本实验采用计数与判断的方式来实现降低时钟源的频率。计数电路可用计数器实现，每来一个时钟脉冲 CLK，计数器就加 1，而每当判断出计数器达到某个数值时，把输出状态求反，就使得 8 个 LED 的亮灭反转一次，即：周期性的输出高电平“1”和低电平“0”。这样设计相当于把 50MHz 的时钟源分频后输出。如果最终要使得 LED 1s 闪烁一次，即输出 1Hz 的时钟脉冲，让计数器计到 25000000 便可以让 LED 亮 0.5 秒、灭 0.5 秒。

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; //时钟输入设置
      dout:out std_logic_vector(7 downto 0)); //数据输出设置
end LED;
```

```
architecture behave of LED is
signal p:std_logic_vector(31 downto 0); //信号变量设置
signal t:std_logic_vector(7 downto 0):="00000000"; //变量 t 赋初始值
begin
process(clk)
begin
if(clk'event and clk='1')then //
    p<=p+1;
    if(p=25000000)then
        p<="00000000000000000000000000000000";
        t<=t XOR "11111111"; //t 值取反
    end if;
end if;
end process;
dout<=t;
end behave;
```

代码分析:


先定义信号变量 P, t。当时钟每产生一个上升沿 P 值加 1, 当 P 加到 25000000 时 t 取反同时 P 复位置 0. 在程序执行过程中始终将 t 赋值给数据输出端口 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 led1。


(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 led.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→EP2K8Q208C8, 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。

(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

第二个实验完成了, 看到实验板上 8 个 LED 灯周期性的闪烁, 是不是觉得这个实验比上个实验有意思呢? 回想一下我们通过这个实验学会了什么? 采用计数与判断的方式来实现分频的 VHDL 的编程方法, 你学会了没? 应该有那么一点点收获吧!