

FPGA 入门系列实验教程 V1.0

前言

目前市场销售 FPGA 开发板的厂商很多,但大多只提供些简单的代码,没有详尽的文档和教程提供给初学者。经验表明,很多学生在学习 FPGA 设计技术的过程中,虽然刚开始学习热情很高,但真正坚持下来一直到“学会”的却寥寥无几,除了学生个人毅力的因素外,另外一个更主要的原因就是所选的开发板缺少配套的说明文档和手把手系列的入门教程。原本 FPGA 的学习门槛相对于单片机来说就高了不少,再加上缺少手把手系列教程资料,这就给初学者学习 FPGA 增添了更多的困难,很多初学者因为找不到入门的方法而渐渐失去了学习 FPGA 的兴趣和信心。

作者从接著到系统学习 FPGA 有两年多的时间了,学习 FPGA 的时间不长,期间因为没有专业的老师指导,自己摸索学习 FPGA 走了很多的弯路。有过问题迎刃而解的快乐,也有过苦苦寻求结果和答案的痛苦历程,回想起自己学习 FPGA 的历程,从开始的茫然,到后来的疯狂看书,购买开发板,在开发板上练习各种 FPGA 实例,到最后能独立完成项目,一路走来,感受颇多。发觉学习 FPGA 只要选择正确的方法是有捷径可走的,有很多人学习 FPGA 很长时间,因为没有找到正确的方法还是停留在入门阶段。

针对现状,作者从初学者的角度出发,结合作者学习 FPGA 的经验和亲身体会,遵循“宁可啰嗦一点,也不放过细节”的原则编写了详尽的实验教程作为艾米电子工作室开发套件的配套教程使用,主要面向 FPGA 初学者。FPGA 的学习只有通过大量的操作与实践才能很好并快速的掌握。为此本实验教程从点亮 LED 灯写起,深入浅出,以图文并茂的方式由易到难地列举了很多实例,采用手把手、Step by Step 的方式让初学者以最快的方式掌握 FPGA 技术的开发流程以及 Quartus II 软件的使用,从而激起初学者学习 FPGA 的兴趣。在教程中作者采用“授人以渔”的方式,努力做到不仅讲述怎样进行试验,而且分析为什么这样做,以便初学者深刻理解并快速掌握 FPGA 的学习方法。

本实验教程的所有实例均在艾米电子工作室开发套件上验证通过,本教程虽然简单但编写也花费了作者大量的时间和精力,对于转载需要注明出处:<http://www.amy-studio.com> (艾米电子工作室 作者:静水流深),并未经艾米电子工作室同意不得用于其他商业用途。

FPGA 技术是不断发展变化的,要掌握 FPGA 技术的精髓,需要设计者在实践中不懈地摸索与积累,逐步提高自己的设计水平,本实验教程试图对初学者起到快速入门的作用。但由于作者学习 FPGA 时间不长,水平有限,错漏和不严谨之处在所难免,欢迎大家批评指正。并请您将阅读中发现的错误或建议发送到作者 Email:zhoujie9220@163.com,以使本教程不断地完善。

FPGA 入门系列实验教程——LED 跑马灯

1. 实验任务

让实验板上的 8 个 LED 实现跑马灯的功能。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 Verilog HDL 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 EP2C5T144 开发套件。
软件实验环境为 Quartus II 8.1 开发软件。

3. 实验原理

从LED[0]到LED[7]依次点亮，第二个灯点亮时第一个熄灭，每个灯交换的时间为0.5S；看上去的效果就象一个亮点从LED[0] 跑向LED[7],然后重复此循环，故命名跑马灯。初始状态时，LED[0]亮，其余熄灭。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LED[1]，同时LED[0]熄灭，并进行移位。这样，依次点亮所有的灯，就形成了跑马灯。而当LED[7]点亮时，需要一个操作使得恢复为初始状态，即：LED[0]亮，其余熄灭。然后，再进行一次跑马灯，重复此循环。如果是右移位，就出现向右跑马的现象；反之，向左跑马。

4. 实验程序


```
module ledwalk(led,clk);
input clk;
output [7:0] led;
reg[7:0]led_out;
reg[25:0]buffer;
always@(posedge clk)
begin
    buffer<=buffer+1'b1;
    if(buffer==26'd25000000)
    begin
        led_out=led_out<<1;
        if(led_out==8'b00000000)
            led_out=8'b00000001;
    end
end
assign led=led_out;
endmodule
```

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 ledwalk。


(2) 建立文本编辑文件：

点击 File→New..在该项目下新建 Verilog HDL 源程序文件 ledwalk.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 EPCS1, 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 LED[0]---LED[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

实验应该很轻松就成功了吧！看到 LED[0]到 LED[7]依次点亮，移位运算符的使用应当非常熟练了吧！你有没有想过用其他方式实现跑马灯的功能，比如用 case 语句。下面给出一个用 case 语句实现的跑马灯程序供大家学习。大家可以比较一下两种方式实现跑马灯的资源占用情况。

```
module ledwalk(sys_clk, led);
input sys_clk;
output [7:0] led;
reg [7:0] led;
reg [24:0] count;
reg [2:0] state;
wire clk;

always @ (posedge sys_clk)
count=count+1;
assign clk=count[24];

always @ (posedge clk)
begin
case(state)
3'b000: led=8'b10000000;
3'b001: led=8'b01000000;
3'b010: led=8'b00100000;
3'b011: led=8'b00010000;
3'b100: led=8'b00001000;
3'b101: led=8'b00000100;
3'b110: led=8'b00000010;
3'b111: led=8'b00000001;
endcase
state=state+1;
end
endmodule
```

用 case 语句实现跑马灯的方法应该很容易理解吧!简单的介绍一下,先把系统时钟进行分频,分频后的时钟 clk 作为 case 语句的触发时钟,每来一个 clk 脉冲,计数器加 1,通过 case 语句的选择实现跑马灯的功能。大家试着修改一下程序让它实现流水灯的功能或者其他方式的花样灯。应该很简单吧!