

VHDL 流水灯设计

侯晶南

中国海洋大学信息学院

2011 级通信工程

hjn910@163.com

摘 要

VHDL 主要用于描述数字系统的结构的行为,功能和接口.除了含有许多具有硬件特征的语句外,VHDL 的语言形式、描述风格以及语法是十分类似于一般的计算机高级语言.VHDL 的程序结构特点是将一项工程设计,或称设计实体(可以是一个元件,一个电路模块或一个系统)分成外部(或称可视部分,及端口)和内部(或称不可视部分),既涉及实体的内部功能和算法完成部分.在对一个设计实体定义了外部界面后,一旦其内部开发完成后,其他的设计就可以直接调用这个实体.

本文是运用 vhd1 语言基于单片机 89C51 上进行的流水灯设计.

关键词: VHDL 、 流水灯、 控制系统

1.引言

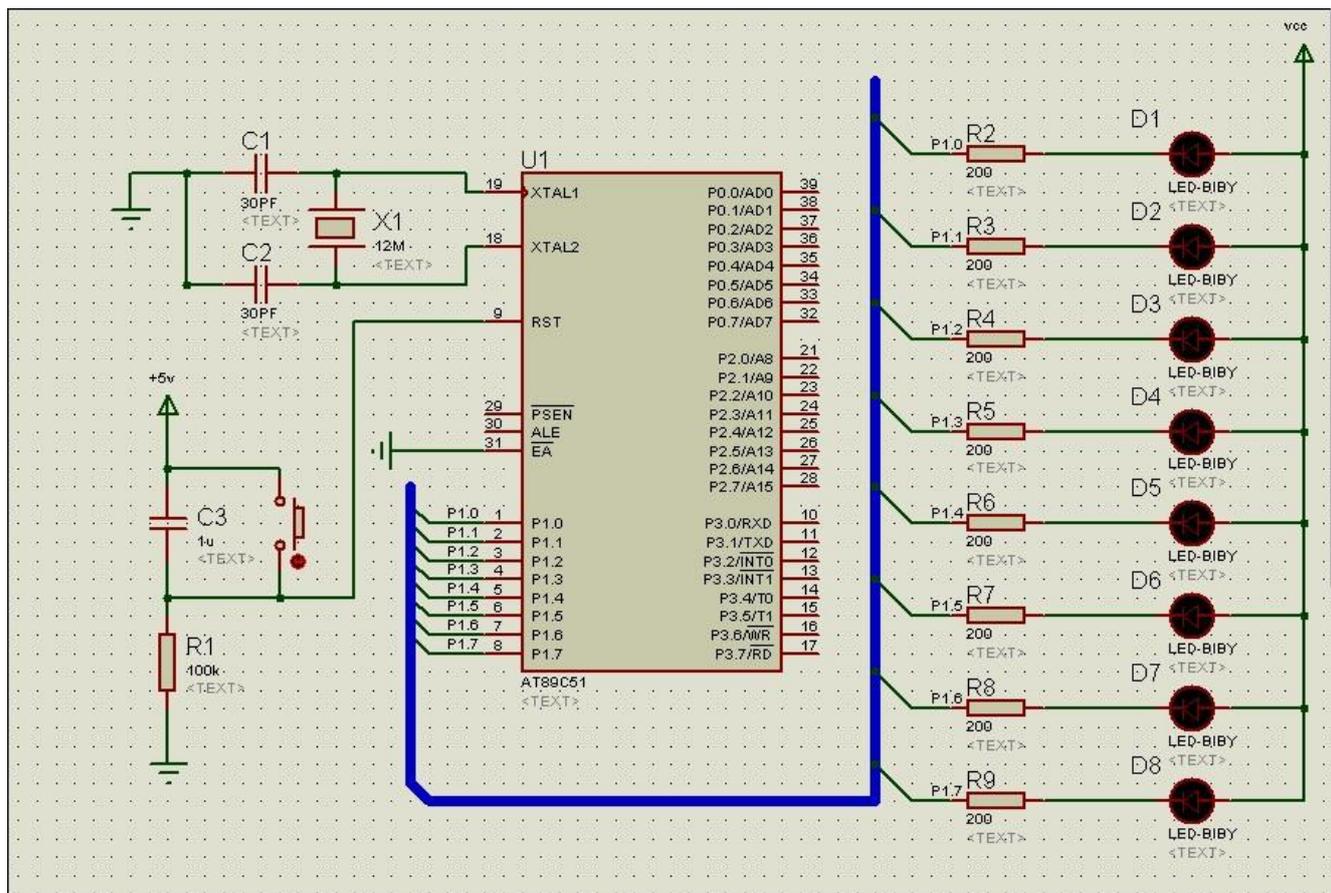
学习 VHDL 的最有效方法就是理论与实践并重,现在我用 VHDL 语言做出流水灯的效果,需要更深的去了解 VHDL 语言的具体应用,努力的去查找资料,当今时代是一个新技术层出不穷的时代,在电子领域尤其是自动化智能控制领域,传统的分立元件或数字逻辑电路构成的控制系统,正以前所未见的速度被智能控制系统所取代.灵活运用 VHDL 愈发重要,而本人能力有限,所以选择了相对简单的流水灯设计,期以此锻炼自己的实践能力,并有助于课程的学习.

2. 实验设计

2.1 设计思路

想要实验流水灯,则需借助单片机来实现.AT89C51 单片机是美国 ATMEL 公司生产的低电压、高性能 CMOS 8 位单片机,具有丰富的内部资源: 4kB 闪存、128BRAM、32 根 I/O 口线、2 个 16 位定时/计数器、5 个向量两级中断结构、2 个全双工的串行口,具有 4.25~5.50V 的电压工作范围和 0~24MHz 工作频率,使用 AT89C51 单片机时无须外扩存储器.因此,本流水灯实际上就是一个带有八个发光二极管的单片机最小应用系统,即为由发光二极管、晶振、复位、电源等电路和必要的软件组成的单个单片机.然后通过 VHDL 语言实现内部结构的编译,从而实验流水灯的设计.

2.2 流水灯硬件原理图



从原理图中可以看出,如果要把接在 P1.0 口的 LED1 亮起来,那么只要把 P1.0 口的电平变为低电平就可以了;相反,如果要把接在 P1.0 口的 LED1 熄灭,就要把 P1.0 口的电平变为高电平;同理,接在 P1.1~P1.7 口的其他 7 个 LED 的点亮和熄灭的方法同 LED1.因此,要实现流水灯功能,我们只要将发光二极管 LED1~LED8 依次点亮、熄灭,8 只 LED 灯便会一亮一暗的做流水灯了.在此我们还应注意一点,由于人眼的视觉暂留效应以及单片机执行每条指令的时间很短,我们在控制二极管亮灭的时候应该延时一段时间,否则我们就看不到“流水”效果了.

3. 实验流程

3.1 实验过程

单片机的应用系统由硬件和软件组成,上述硬件原理图搭建完成上电之后,我们还不能看到流水灯循环点亮的现象,我们还需要告诉单片机怎么来进行工作,即编写程序控制单片机管脚电平的高低变化,来实现发光二极管的一亮一灭.软件编程是单片机应用系统中的一个重要的组成部分,是单片机学习的重点和难点.下面我们以 5 个流水灯为例,简述其编译过程.

主要程序为:

```
▶ present_state is
▶ when a=>
▶ led_out<="10001";--*.*?*两边灯亮
▶ next_state<=b;
▶ when b=>
▶ led_out<="01010";--*.*?*第二个和点三个灯亮
▶ when c=>
▶ led_out<="00100";--*.*?中间灯亮
▶ next_state<=d;
▶ when d=>
▶ led_out<="11111";--全亮
▶ next_state<=a;
▶ end case;
```

分析上述程序,通过定义四个变量 a,b,c,d 来代表其四种状态,通过 0 与 1 代表灯是否点亮,又定义 next-state 语句使程序向下运行,此外还定义了延时程序,否则无法观察到流水的现象了.

3.2 实验现象与结果

实验仿真如下图：



由图我们可以看出其由 10001 到 01010 的变化,即从两边灯亮到中间灯亮的过程,所以本实验是成功的.

4. 实验总结与分析

4.1 实验分析：

本实验设计简单,通过变量定义与工作状态的一一对应来实现编译过程,并运用了 ISE 14.6 进行实验编程与仿真,由于没有实体电路板,所以本实验省去了分配引脚等过程,只有仿真,但成功的展现了灯的亮灭变化,亦可称为成功的实验.

4.2 实验总结

在实验过程中有付出,有收获,为了找到程序的错误,坚持到了电脑没电,最后发现实体声明部分出了问题,虽然这直接导致了第二天的早上没起来上

课,但还是感觉到很充实和快乐,独立完成程序的制作,事情虽小,但对我意义重大,一行行的检验程序对我这种没耐心的人是个巨大的挑战,还好我坚持了下去,总结起来就是那句话:成功就是你不能坚持的时候再坚持一下.

5. 鸣谢

首先感谢郑海永老师给我们这个展示的机会,虽然是半强迫性质的,但对我们的好处不言而喻,从中我们不仅掌握了课程知识,还有一些做人的道理,再次感谢!

然后还有魏涛同学在实验过程中给予的帮助,在仿真时,波形总是出错,后来在魏涛同学的帮助下才予以解决,还有许多同学在此过程中给予了帮助,特此感谢.

参考文献

- (1) 李欣 张海燕编著,VHDL 数字系统设计,科学出版社,2009.1
- (2) 万光毅,严义,邢春香,单片机实验与实践教程[M],北京航空航天大学出版社,2006.4

附录：

代码：

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
use IEEE.STD_LOGIC_ARITH.ALL;
```

```
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
entity led is
```

```
Port(clk:in STD_LOGIC;
```

```
led_out:out STD_LOGIC_VECTOR(4 downto 0));
```

```
end led;
```

```

architecture Behavioral of led is

signal n:integer range 0 to 6;

type state_type is(a,b,c,d);

signal

next_state,present_state:state_type;

begin

tem:process(present_state)

begin

case

present_state is

when a=>

led_out<="10001";--*...*?两边灯亮

next_state<=b;

when b=>

led_out<="01010";--.*.*?第二个和点三个灯亮

when c=>

led_out<="00100";--..*..中间灯亮

next_state<=d;

```

```

when d=>
led_out<="11111";--全亮
next_state<=a;
end case;
end process tem;

    tep:process(clk)
begin
if  rising_edge(clk)then--可通过设计 n 的值进行延时
    n<=n+1;
if  n=5  then
    n<=0;
    present_state<=next_state;

    end if;
end if;

    end process tep;

end Behavioral;

```