**《电子线路设计、测试与实验》实验报告**

|  |  |
| --- | --- |
| 实验二名称： | 步进电机脉冲分配器 |
| 院（系）： | 电子信息与通信工程学院 |
| 专业班级： | 电磁1802 |
| 姓名： | 吴叶赛 |
| 学号：  时间： | U201813405  2020年9月17日 |

**一、实验目的**

1. 熟悉verilog分层设计；
2. 熟悉功能模块的设计与重复调用；
3. 掌握步进电机脉冲分配器的实现原理；
4. 实现步进电机的自启动和正转反转的三相六拍分配器电路。

**二、实验条件**

|  |  |
| --- | --- |
| 内容 | 型号 |
| 开发环境 | Vivado2018.1 |
| 开发语言 | Verilog HDL |
| 开发板 | NEXYS4 DDR |

**三、实验原理**

1、步进电机是一种作为控制用的特种电机, 将电脉冲信号转换成角位移或线位移的机电元件。步进电动机的输入量是脉冲序列，输出量则为相应的增量位移或步进运动。它的旋转是以固定的角度(称为"步距角")一步一步运行的, 其特点是没有积累误差(精度为100%), 所以广泛应用于各种开环控制。步进电机的运行要有一电子装置进行驱动, 这种装置就是步进电机驱动器, 它是把控制系统发出的脉冲信号转化为步进电机的角位移, 或者说: 控制系统每发一个脉冲信号, 通过驱动器就使步进电机旋转一步距角。所以步进电机的转速与脉冲信号的频率成正比。所以，控制步进脉冲信号的频率，可以对电机精确调速；控制步进脉冲的个数，可以对电机精确定位目的。正常运动情况下，它每转一周具有固定的步数；做连续步进运动时，其旋转转速与输入脉冲的频率保持严格的对应关系，不受电压波动和负载变化的影响。由于步进电动机能直接接受数字量的控制，所以特别适宜采用微处理器进行控制。；

　2、步进电机通过细分驱动器的驱动，其步距角变小了，如驱动器工作在10细分状态时，其步距角只为‘电机固有步距角‘的十分之一，也就是说：‘当驱动器工作在不细分的整步状态时，控制系统每发一个步进脉冲，电机转动1.8°；而用细分驱动器工作在10细分状态时，电机只转动了0.18° ‘，这就是细分的基本概念。 细分功能完全是由驱动器靠精确控制电机的相电流所产生，与电机无关。

3、驱动器细分有什么优点，为什么一定建议使用细分功能？

　　驱动器细分后的主要优点为：完全消除了电机的低频振荡。低频振荡是步进电机（尤其是反应式电机）的固有特性，而细分是消除它的唯一途径，如果您的步进电机有时要在共振区工作（如走圆弧），选择细分驱动器是唯一的选择。提高了电机的输出转矩。尤其是对三相反应式电机，其力矩比不细分时提高约30-40% 。提高了电机的分辨率。由于减小了步距角、提高了步距的均匀度，‘提高电机的分辨率‘是不言而喻的。

步进电机的控制

向步进电机的驱动器发送所需要的数量的脉冲，

驱动器驱动步进电机按照发送的脉冲数量旋转。具体旋转位置及速度由驱动器来设定。

通常步进电机的驱动模块要通过微处理器的GPIO电平控制，一般接口有控制驱动器睡眠或者工作状态的sleep硬件引脚，设置一个输入脉冲控制电机转动的模式的细分驱动器的model引脚，接受输入脉冲个数的step引脚，控制步进电机转动方向的direction引脚。初始化转动方向和转动细分控制器后，sleep引脚设置为active状态，输入脉冲队列，马达的转动频率跟输入脉冲的频率成正比。

**四、实验内容**

1）拟定步进器电路的组成框图，直接设计电路。

2）使用Verilog HDL设计各单元电路并进行仿真分析。

3）用FPGA开发板实现步进器，并实际测试步进器系统的逻辑功能。

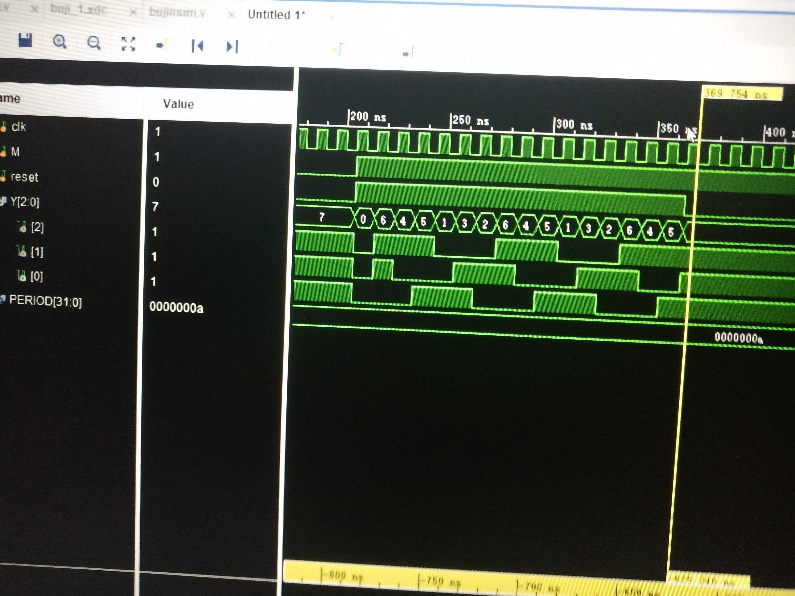
4）根据实验要求，实现步进电机的自启动和正转反转的三相六拍分配器电路。

**五、仿真结果**

## 正转对应波形图

## 

## 反转对应波形图



**六、模块代码**

module bujin(

input M,

input reset,

input clk,

output reg [2:0] Y

);

reg clk1;

reg [25:0] cnt1;

always @(posedge clk)

begin

if (cnt1 >= 49999999)

begin

cnt1 <= 26'b0;

clk1 = ~clk1;

end

else

begin

cnt1 <= cnt1+ 1;

end

end

parameter S0=3'b111,S1=3'b000,S2=3'b110,S3=3'b010,S4=3'b011,S5=3'b001,S6=3'b101,S7=3'b100;

always@(posedge clk1 or negedge reset)

begin

if(~reset)

Y<=S0;

else

case(Y)

S0:Y<=S1;

S1:Y<=S2;

S2:Y<=(M)?S7:S3;

S3:Y<=(M)?S2:S4;

S4:Y<=(M)?S3:S5;

S5:Y<=(M)?S4:S6;

S6:Y<=(M)?S5:S7;

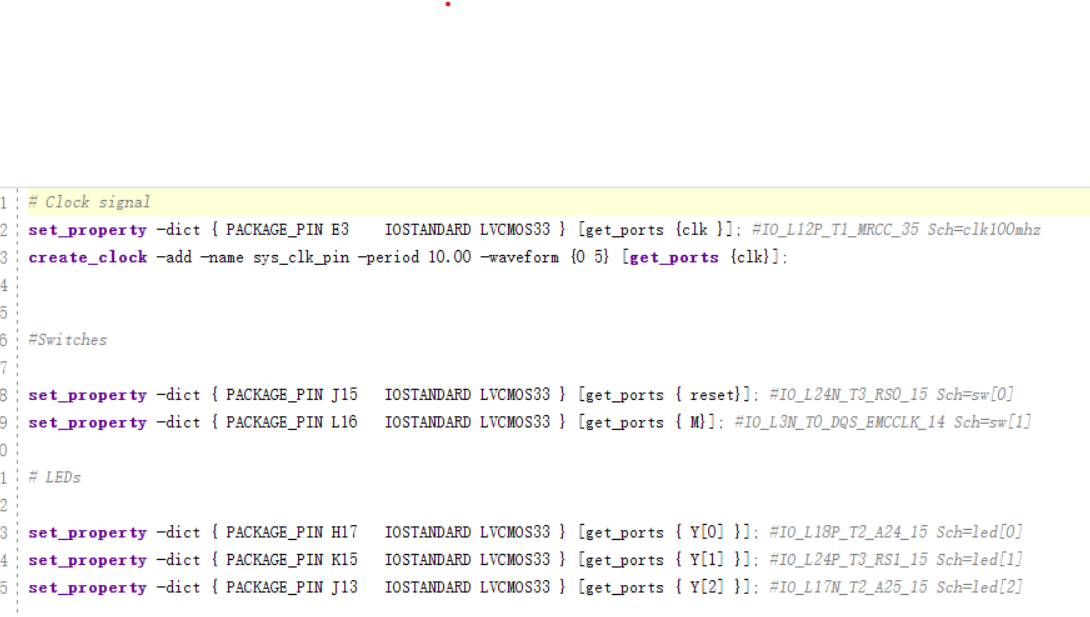
S7:Y<=(M)?S6:S2;

endcase

end

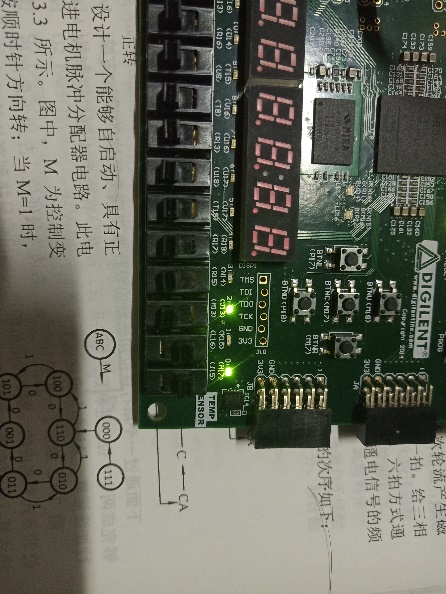
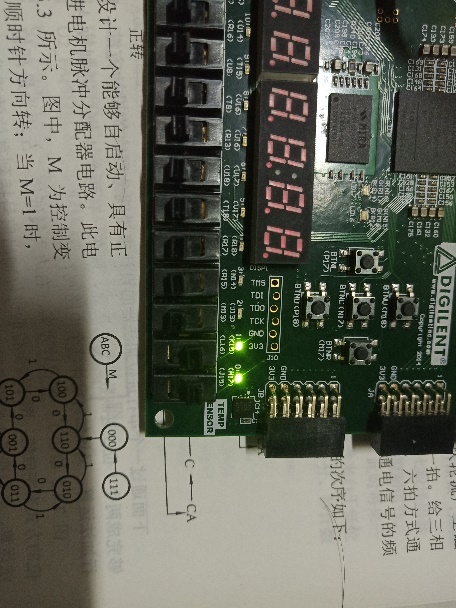
endmodule

**七、引脚约束**



**八、实验结果及分析**

将代码装载到开发板上进行实验，发现功能正确

第一个按钮为复位按钮，第二个按钮为正转反转按钮。

**九、小结**

这个步进器实验比数字钟实验稍微简单一些，所涉及到的引脚少多了，做完数字钟再做这个感觉会比之前一头雾水轻松不少。

本次试验综合性较强，参考的是mooc上的代码，自顶向下构建每个模块，最后实现了所有功能。可见，在电路设计中，一定要有目的性，要有很明细的架构设计。

而且有时候一步步照着慕课上做，最后也会有各种各样的问题出现，我们要自己多思考，真正理解了这个实验的原理和过程之后，做起来就又会有不一样的感觉，也会对EDA设计有不一样的理解。