

# 浙江大学实验报告

专业： 电子信息工程

姓名： 邢毅诚

学号： 3190105197

日期： 2020-11-1

地点： 东三-406

课程名称： 电路与电子技术实验

指导老师： 祁才君

成绩：

实验名称： PWM 调速和复杂导航控制

实验类型： 验证实验

同组学生姓名： 郑冰阳

## 一、 实验目的

- (1) 熟悉掌握 VHDL 硬件描述语言
- (2) 了解 PWM 调速原理
- (3) 掌握 VHDL 智能小车速度控制原理

## 二、 实验基本内容

### 1. 实验内容

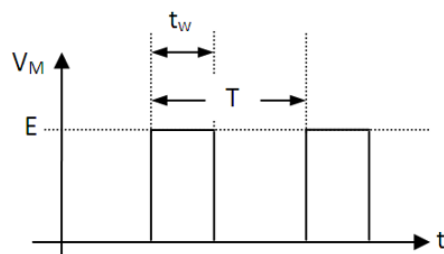
- (1) 设计并下载数字 PWM 产生电路
- (2) 至少设定二种比较值，用示波器观察并记录 PWM 的输出波形，测量其周期、幅度和占空比，并分析测试结果
- (3) 将 PWM 输出与电机驱动模块的 ENA 和 ENB 连接， $IN1=1, IN2=0$ ，比较不同占空比情况下的运动转速
- (4) 将 PWM 电路和实验 6 的寻迹导航控制电路相结合，比较占空比 50% 和 100% 两种情况下的轨道跟踪精度

### 2. 实验原理

- (1) 设计并下载数字 PWM 产生电路
- (2) 至少设定二种比较值，用示波器观察并记录 PWM 的输出波形，测量其周期、幅度和占空比，并分析测试结果
- (3) 将 PWM 输出与电机驱动模块的 ENA 和 ENB 连接， $IN1=1, IN2=0$ ，比较不同占空比情况下的运动转速
- (4) 将 PWM 电路和实验 6 的寻迹导航控制电路相结合，比较占空比 50% 和 100% 两种情况下的轨道跟踪精度

PWM 是一种通过调节电压 (输出电压占空比) 以调节电机转速的一种调节方法。相比与其他调速原理，此调节方法可以使电机两端电压导通时始终等于  $E$ ，扭矩大，同时控制所需功耗较

小。当输入电压的频率较大时, 即使占空比并不为 100%, 电机也能保持匀速直线运动的状态。由此, 根据小车所需速度, 我们可以利用间歇导通和电机运动惯性调速, 调节输出电压波形 (如下图所示)



$$\bar{V}_M = \frac{t_w}{T} \cdot E = \delta \cdot E$$

$$\delta \triangleq \frac{t_w}{T}, \delta \in [0, 1], \bar{V}_M = \delta \cdot E$$

图 1: PWM 调速法输出波形

对于模拟 PWM 电路, 我们需要产生固定频率或周期的三角波, 并将三角波与可调直流电压进行比较产生 PWM 输出, 通过改变可调直流电压, 我们就可以改变输出的脉冲宽度。

而对于数字式 PWM 电路, 其由方波发生器, N 进制计数器, 以及比较器构成, 进而输出 PWM 波。首先方波发生器产生固定方波, 而计数器根据产生的方波生成锯齿波, 而比较器通过比较锯齿波与设定电压值来输出 PWM 波,

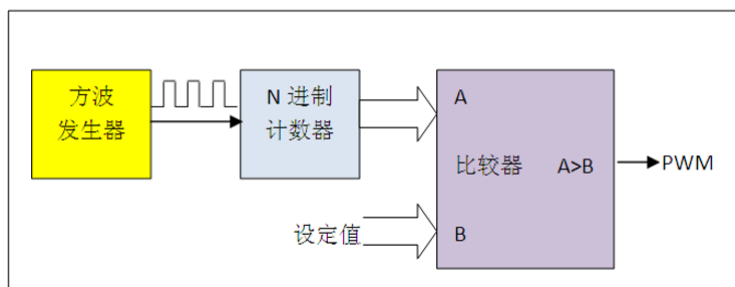


图 2: 数字式 PWM 电路原理

下面是各项参数的选取方式:

- 计数器: 根据需要, 我们认为 16 进制计数器正好可以比较完好的满足我们的需求, 同时操作也不至于过于繁琐, 因此我们选取 16 进制计数器。
- 方波发生器的频率: 将小车电机与信号发生器直接相连, 不断调节频率与占空比, 选取一个较为稳定的频率, 使得在调节占空比时, 车轮的转速会有较为明显的变化

- 比较器设定值: 根据实际情况需要进行设定

FPGA 设计 PWM 的设计框图如下图所示:

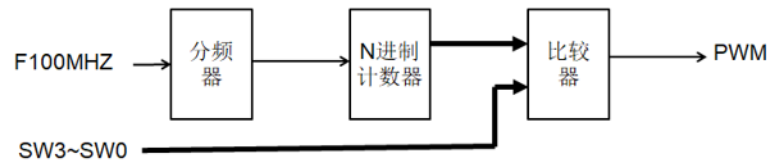


图 3: PWM 设计框图

其中,分频器的作用是将系统 100MHZ 转换为对应 PWM 要求的频率,而 N 进制计数器则用于产生锯齿波,用于与设定值进行比较,并输出 PWM 波。

正如设计框图所示, PWM 发生器所需编写的源码也由以下及部分构成:

- TOP.vhd: 顶层源码,作为主文件运行
- freq\_div.vhd: 分频器 VHDL 源码
- CNT16.vhd: 16 进制计数器源码
- COMP4BIT.vhd: 比较器源码

按照要求,编写源码如下图所示:

- **TOP.vhd**

```
1      use IEEE.std_logic_1164.all;
2      use IEEE.std_logic_unsigned.all;
3      USE IEEE.numeric_std.ALL;
4      USE IEEE.std_logic_arith.all;
5      entity TOP is
6      port (sw : in STD_LOGIC_VECTOR (3 downto 0);
7      clk_in: in STD_LOGIC;
8      L:in STD_LOGIC;
9      R:in STD_LOGIC;
10     pwm1 : out STD_LOGIC;
11     pwm2 : out STD_LOGIC);
12     end TOP ;
13     architecture Behavioral of TOP is
14     signal freqout: STD_LOGIC;
15     signal cntout : STD_LOGIC_VECTOR (3 downto 0);
16     signal pwm3:STD_LOGIC;
17     component freq_div is
18     port (
19     freq_in: in STD_LOGIC;
20     div_num: natural range 0 to 999;
21     freq_out: out std_logic);
22     end component ;
23     component CNT16 is
```

```
24     port ( CLK : in STD_LOGIC;  
25           Q : out STD_LOGIC_VECTOR (3 downto 0));  
26     end component;  
27  
28     component COMP4BIT is  
29     port (A : in STD_LOGIC_VECTOR (3 downto 0);  
30           B : in STD_LOGIC_VECTOR (3 downto 0);  
31           AGB : out STD_LOGIC);  
32     end component;  
33     begin  
34  
35     u0: freq_div port  
36         map(freq_in=>clk_in,div_num=>667,freq_out=>freqout);  
37     u1: CNT16 port map(CLK=>freqout,Q=>cntout);  
38     u2: COMP4BIT port map(A=>cntout,B=>sw,AGB=>pwm3);  
39     pwm1<=(pwm3 and R);  
40     pwm2<=(pwm3 and L);  
41     end Behavioral;
```

其中, pwm1 为左轮输出, pwm2 为右轮输出, L 为左轮输入, R 为右轮输入, clk\_in 为系统内部时钟, sw 为设定值

- freq\_div

```
1     library IEEE;  
2     use IEEE.std_logic_1164.all;  
3     use IEEE.std_logic_unsigned.all;  
4     USE IEEE.numeric_std.ALL;  
5     USE IEEE.std_logic_arith.all;  
6  
7     entity freq_div is  
8     port(  
9     freq_in: IN STD_LOGIC;  
10    div_num: natural range 0 to 999;  
11    freq_out: OUT STD_LOGIC  
12    );  
13    end freq_div;  
14  
15    architecture freq_div_behav of freq_div is  
16    begin  
17    P1:process(freq_in)  
18    variable cnt:natural range 0 to 99999999;  
19    begin  
20    if rising_edge(freq_in) then  
21    if cnt=div_num then  
22    cnt:=0;  
23    elsif cnt<=div_num/2 then  
24    freq_out<='0';  
25    cnt:=cnt+1;  
26    else
```

```
27         freq_out<='1';
28         cnt:=cnt+1;
29     end if;
30 end if;
31 end process P1;
32 end freq_div_behav;
```

- CNT16.vhd

```
1         library IEEE;
2         use IEEE.std_logic_1164.all;
3         use IEEE.std_logic_unsigned.all;
4         USE IEEE.numeric_std.ALL;
5         USE IEEE.std_logic_arith.all;
6
7         entity CNT16 is
8         port(
9             CLK: in STD_LOGIC;
10            Q:out STD_LOGIC_VECTOR(3 downto 0));
11         end CNT16;
12
13         architecture Behavioral of CNT16 is
14             signal tmp: STD_LOGIC_VECTOR(3 downto 0):="0000";
15
16             BEGIN
17             process(CLK)
18             BEGIN
19                 if(CLK'EVENT and CLK='1') then
20                     tmp<=tmp+'1';
21                 END IF;
22             END PROCESS;
23             Q<=tmp;
24             END Behavioral;
```

其中, CLK 为 FPGA 元件内置时钟, Q 为计数

- COMP4BIT.vhd

```
1         library IEEE;
2         use IEEE.std_logic_1164.all;
3         use IEEE.std_logic_unsigned.all;
4         USE IEEE.numeric_std.ALL;
5         USE IEEE.std_logic_arith.all;
6
7         entity COMP4BIT is
8         Port(A: STD_LOGIC_VECTOR(3 downto 0);
9             B: STD_LOGIC_VECTOR(3 downto 0);
10            AGB: out STD_LOGIC);
11         end COMP4BIT;
```

```
13
14     architecture behaviour of COMP4BIT is
15     begin process(A,B)
16     begin
17     if (A>B) then
18     AGB<='1';
19     else
20     AGB<='0';
21     END IF;
22     END process;
23     end behaviour;
```

其中, A 为输入的待比较值, B 为设定值, AGB 为输出值

### 三、 主要仪器设备

- (1) 智能小车
- (2) DE10 开发板
- (3) Keysight DSOX1102G 示波器
- (4) RIGOL DG1022U 信号源
- (5) GWINSTEK GPD4303S 稳压源
- (6) FPGA 开发软件 Quartus(Quartus Prime Standard 17.1)

### 四、 操作方法和实验步骤

- (1) 设计并下载数字 PWM 产生电路
  - (1) 检查实验仪器是否损坏, 有无缺失, 检查无误后开始实验
  - (2) 将智能小车的输出端 (ENA,ENB) 信号发生器直接相连, 调节信号发生器的波形为方波, 进行输出
  - (3) 调节信号发生器输出波形频率, 并选择一个频率, 使得调节输出占空比时, 电机转速也会有较为明显的变化
  - (4)
  - (5) 打开 Quartus 17.1, 新建一个空项目, 设置项目名为 **PWM**, 设置硬件主体名为 **qxor**, 选择开发设备为 **10M50DAF484C7G**, 选择模拟环境为 ModelSim—Altera 并设置语言为 VHDL 语言。
  - (6) 新建四个 vhd 文件, 分别命名为"TOP","CNT16","freq\_div","COMP4BIT", 并编写代码 (已经在实验原理中列出)
  - (7) 在 quartus 中进行编译
  - (8) 编写仿真代码, 如下图所示, 并观察仿真图象, 确认实验结果是否正确

```
1      LIBRARY ieee;
2      USE ieee.std_logic_1164.all;
3
4      ENTITY TOP_vhd_tst IS
5      END TOP_vhd_tst;
6      ARCHITECTURE TOP_arch OF TOP_vhd_tst IS
7      SIGNAL clk_in : STD_LOGIC;
8      SIGNAL L : STD_LOGIC;
9      SIGNAL pwm1 : STD_LOGIC;
10     SIGNAL pwm2 : STD_LOGIC;
11     SIGNAL R : STD_LOGIC;
12     SIGNAL sw : STD_LOGIC_VECTOR(3 DOWNTO 0);
13     COMPONENT TOP
14     PORT (
15     clk_in : IN STD_LOGIC;
16     L : IN STD_LOGIC;
17     pwm1 : OUT STD_LOGIC;
18     pwm2 : OUT STD_LOGIC;
19     R : IN STD_LOGIC;
20     sw : IN STD_LOGIC_VECTOR(3 DOWNTO 0)
21     );
22     END COMPONENT;
23     BEGIN
24     i1 : TOP
25     PORT MAP (
26     — list connections between master ports and signals
27     clk_in => clk_in,
28     L => L,
29     pwm1 => pwm1,
30     pwm2 => pwm2,
31     R => R,
32     sw => sw
33     );
34
35     always : PROCESS
36
37     BEGIN
38     clk_in<='1';L<='1';R<='1';sw<="0000";
39     wait for 1ps;
40     clk_in<='0';L<='1';R<='1';sw<="0000";
41     wait for 1ps;
42
43     END PROCESS always;
44     END TOP_arch;
```

(9) 设置管脚, 将 VHDL 代码烧入 FPGA 器件中

(2) 至少设定二种比较值, 用示波器观察并记录 PWM 的输出波形, 测量其周期、幅度和占空比, 并分析测试结果

- 设置比较值为“0001”，将 FPGA 器件与示波器相连，读出示波器显示的波形等相关信息
  - 设置比较值为“1000”，将 FPGA 器件与示波器相连，读出示波器现实的波形等相关信息
- (3) 将 PWM 输出与电机驱动模块的 ENA 和 ENB 连接，IN1=1,IN2=0，比较不同占空比情况下的运动转速
- 将 PWM 输出与电机驱动模块的 ENA，ENB 分别连接
  - 将 IN1，IN3 设置为 1，IN2，IN4 设置为 0，打开电源进行输出
  - 调节 PWM 输出的占空比，比较其转速
- (4) 将 PWM 电路和实验 6 的寻迹导航控制电路相结合，比较占空比 50% 和 100% 两种情况下的轨道跟踪精度
- 将 PWM 的占空比分别调至 50% 以及 100%(即设定 sw 分别为“1000”以及“0000”)
  - 将小车接通电源，放置于电路上进行运行
  - 观察小车运动状态并比较小车在不同占空比时运动情况的区别

## 五、实验数据记录与分析

- (1) 设计并下载数字 PWM 产生电路

按照实验步骤，进行仿真，获得图像如下图所示：

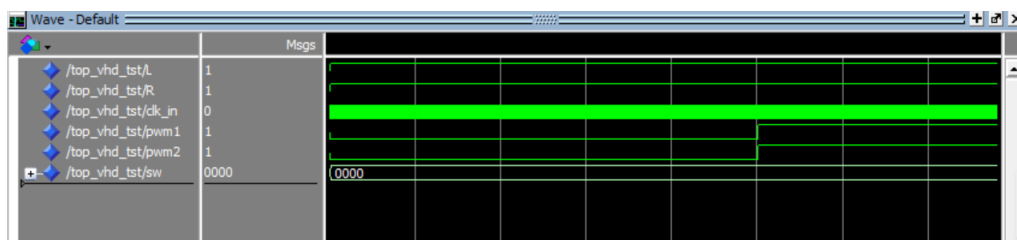


图 4: PWM 设计框图

经过观察后，我们发现，实验的结果与我们预想的结果大致相同，因此我们可以认定程序编写大致正确

- (2) 至少设定二种比较值，用示波器观察并记录 PWM 的输出波形，测量其周期、幅度和占空比，并分析测试结果

按照实验步骤进行实验，测得 FPGA 输出的波形分别如下图所示：

设定值	0001	1000
峰峰值/V	3.666	3.596
最大电平	3.216	2.826
最小电平	-0.45	-0.769
占空比	6.24%	50.00%
周期/ $\mu s$	213.74	213.8

表 1: PWM 输出相关波形



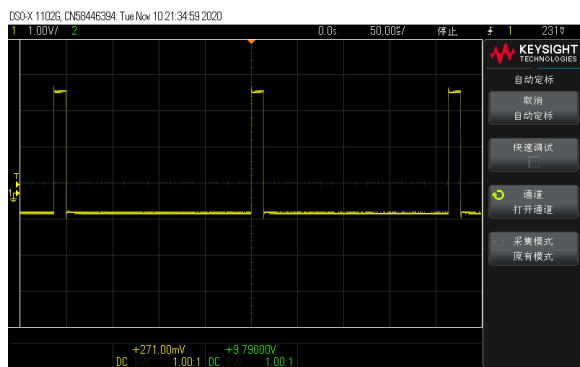


图 5: “0001” 波形

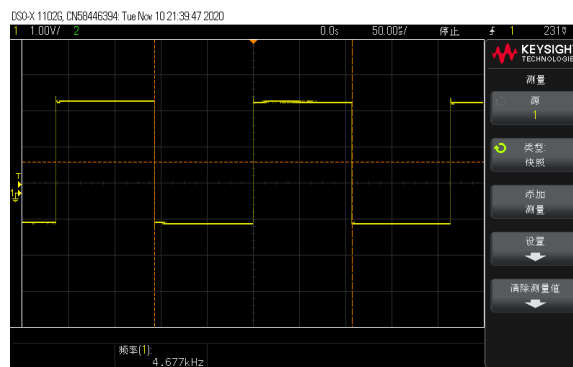


图 6: “1000” 波形

- (3) 将 PWM 输出与电机驱动模块的 ENA 和 ENB 连接, IN1=1, IN2=0, 比较不同占空比情况下的运动转速

经过观察, 我们可以发现, 在调节 PWM 的占空比时, 电机的转速也会逐渐变化。具体表现为: 当转空比减小时, 电机转速变慢, 而转空比增大时, 电机转速变快

- (4) 将 PWM 电路和实验 6 的寻迹导航控制电路相结合, 比较占空比 50% 和 100% 两种情况下的轨道跟踪精度

在经过实验后, 我们发现, 采用 50% 的占空比的 PWM 波时, 小车运行的速度不及占空比为 100% 的 PWM 波, 但同时, 在面临一些转弯弧度较大的转弯时, 小车转空比越小, 其运行轨迹便越准确, 原因在于, 小车运动速度较慢时, 便留有了较大的时间进行转弯, 在面临一些弧度较大的转弯时, 速度较小的汽车反而会顺利通过。

## 六、心得与体会

### 1. 实验感想

在本次实验中, 我较为系统地学习了 VHDL 语言以及其相关知识, 并熟练掌握了 PWM 调速方法的原理以及操作步骤。PWM 是一种同过调节方波占空比从而调节电机速率的调节方式。但同时, 在进行实验的过程中, 我们也遇到了许许多多的困难, 比如相比于其他同学的代码, 我们的代码并不能顺利运行, 最后经过反复修改, 我们才成功完成了实验。