

浙江大学实验报告

专业：电气工程及其自动化

姓名：严旭铨

学号：3220101731

日期：2023 年 10 月 30 日

地点：东三 406 教室

课程名称：电路与电子技术实验 I 指导老师：祁才君 成绩：

实验名称：仪器使用练习 实验类型：电学实验 同组学生姓名：褚玘铖

实验 6 智能小车导航控制系统

一、实验目的

- 了解智能小车的结构和工作原理
- 初步学习智能小车导航控制系统的控制策略
- 复习巩固 ModelSim、Quartus 等软件的使用和 VHDL 语言的编写
- 利用 FPGA 开发板实现小车自动寻迹功能

二、实验内容，

- 了解智能小车导航控制的基本原理
- 实现跟随轨道控制策略
- 实现纠偏轨道控制策略
- 实现寻迹控制策略（一对光电管）
- 实现改进跟随轨道控制策略（正反转）
- 实现寻迹轨道控制策略并让小车完成寻迹跑道

三、实验数据记录、处理与分析

内容一：了解智能小车导航控制的基本原理

智能小车导航控制系统能够实现对小车在寻迹时的闭环控制。智能小车控制系统主要由电子部分和机械部分组成。本实验主要针对控制电路和轨道检测部分进行设计。控制电路部分主体是写入了导航程序的 FPGA 开发板，轨道检测部分主要是由车前的光电对管组成。

本次实验的轨道是地上的黑色胶带，宽度约为 4cm。在小车寻迹过程中，由车前的四对红外光电对管检测小车与轨道的相对位置，并将信号传递到控制电路，控制电路再根据检测信号调整对电机的驱动，从而达到改变小车运动状态（或保持不变）的目的。

在本次实验中，统一约定：

（1）当小车的光电对管位于轨道（黑色胶带）的正上方时，光电对管输出高电位（3.3V），逻辑值记为 1；反之，输出低电位，逻辑值记为 0。

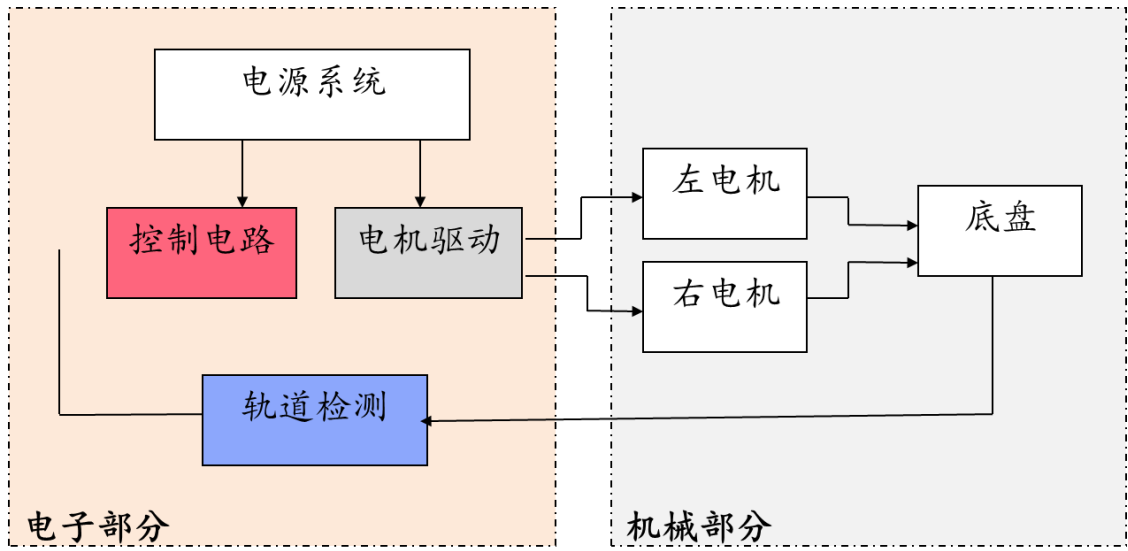
（2）当输入电机的信号为“10”，电机正转；“01”反转；“00”或“11”均不转。

（3）规定小车两个电机和传感器 PCB 板对应的一侧为正向，当小车正向前进，面向小车前进方向，左手边为左侧，右手边为右侧。对光电对管进行编号命名时，从左到右依次命名为 LS2, LS1, RS1, RS2，其中 LSX 位于 1-6 号管，RSY 位于 7-12 号管。

(4) 在控制电路中，输入逻辑值与电机对应状态如下表：

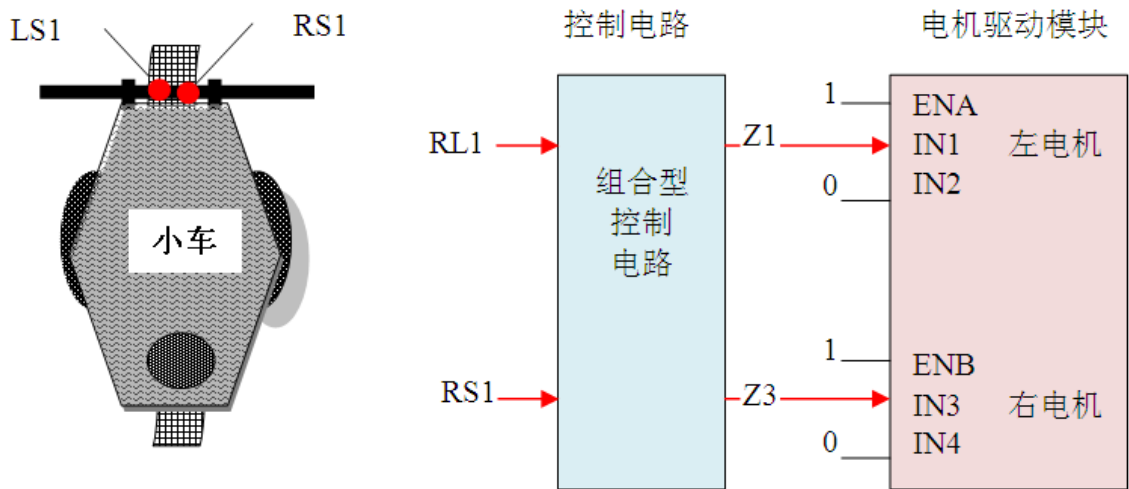
电机输入和输出值对应表

左电机 ENA					右电机 ENB				
IN1	0	0	1	1	IN3	0	0	1	1
IN2	0	1	0	1	IN4	0	1	0	1
OUT1	0	-1	1	0	OUT2	0	-1	1	0



控制系统原理图

内容二：实现跟随轨道控制策略



1. 实验要求

- 1) 只用 2 对光电传感器

- 2) 仅控制左右电机前行和停止 2 种状态
- 3) 实现前进、左转、右转三种运动模式

2. 设计思路

- 1) 选用位于小车 PCB 板中间位置的 6 号和 7 号光电对管。
- 2) 两个对管都位于轨道正上方时，说明小车正沿着轨道直线前进，两个传感器给开发板的输入信号均为“11”，要求两个电机都正转使其前进，则开发板对电机驱动的输出信号均为“10”。
- 3) 当左侧对管位于轨道正上方而右侧对管不在轨道正上方时，说明小车需要左转，两个传感器给开发板的输入信号为“10”，要求左电机不转，右电机正转使其左转弯，则开发板对左右电机驱动的输出信号分别为“00”“10”。
- 4) 当右侧对管位于轨道正上方而左侧对管不在轨道正上方时，说明小车需要右转，两个传感器给开发板的输入信号为“01”，要求右电机不转，左电机正转使其右转弯，则开发板对左右电机驱动的输出信号分别为“10”“00”。

3. 实验过程

- 1) 根据设计思路列出输入输出的逻辑真值表

跟随轨道控制真值表

检测状态	LS1	RS1	IN1 (L)	IN3 (R)	小车运动趋势
在轨	1	1	1	1	前进
左在轨	1	0	0	1	左转
右在轨	0	1	1	0	右转

- 2) 根据真值表编写 VHDL 代码，直至其编译通过

```

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  entity FC is
7      port (  LS1,RS1: in std_logic;
8              IN1,IN3: out std_logic);
9  end FC;
10 architecture BEHAV of FC is
11 begin
12     IN1<=RS1;
13     IN3<=LS1;
27 end BEHAV;

```

- 3) 将代码导入 Quartus，分析并设置管脚。

out	IN1	Output	PIN_Y4	3	B3_N0	2.5 V (c
out	IN3	Output	PIN_Y5	3	B3_N0	2.5 V (c
in	LS1	Input	PIN_AA5	3	B3_N0	3.3-V L
in	RS1	Input	PIN_AA6	3	B3_N0	3.3-V L

- 4) 在小车上按照管脚和输入值对应关系连接好电路，其中 LED6 – AA5, LED7 – AA6, Y4 – (PCB) IN1, Y5 – (PCB) IN3。在 PCB 控制板上，ENA – 5V，ENB – 5V。最后连接地线，使 PCB 板和 FPGA 板共地。
- 5) 分析后生成程序，无误，用 USB 接口将其导入 FPGA 板。
- 6) 将小车连接电源后在测试轨道（无岔路）上进行测试。

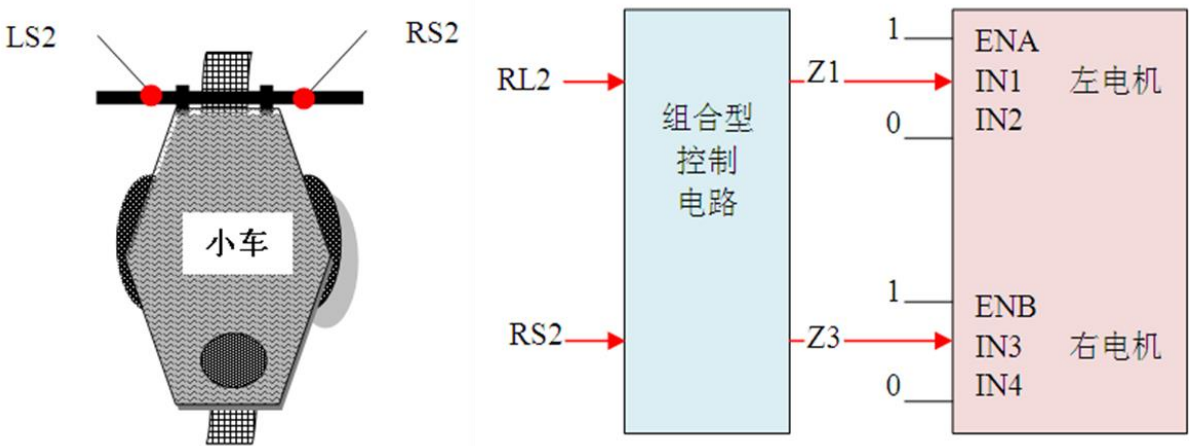
4. 实验数据处理与分析

小车能够按照要求进行前进（两灯均在轨）、左转和右转。

内容三：实现纠偏轨道控制策略

1. 实验要求

- 1) 只用 2 对光电传感器
- 2) 仅控制左右电机前行和停止 2 种状态
- 3) 实现前进、左转、右转三种运动模式



2. 设计思路

- 1) 选用位于小车 PCB 板两侧位置的 3 号和 10 号光电对管。
- 2) 两个对管都不位于轨道正上方时，说明小车正沿着轨道直线前进，两个传感器给开发板的输入信号均为“00”，要求两个电机都正转使其前进，则开发板对电机驱动的输出信号均为“10”。
- 3) 当左侧对管位于轨道正上方而右侧对管不在轨道正上方时，说明小车需要左转，两个传感器给开发板的输入信号为“10”，要求左电机不转，右电机正转使其左转弯，则开发板对左右电机驱动的输出信号分别为“00”“10”。
- 4) 当右侧对管位于轨道正上方而左侧对管不在轨道正上方时，说明小车需要右转，两个传感器给开发板的输入信号为“01”，要求右电机不转，左电机正转使其右转弯，则开发板对左右电机驱动的输出信号分别为“10”“00”。

3. 实验过程

- 1) 根据设计思路列出输入输出的逻辑真值表

纠偏轨道控制真值表

检测状态	LS2	RS2	IN1 (L)	IN3 (R)	小车运动趋势
均不在轨	0	0	1	1	前进
左在轨	1	0	0	1	左转
右在轨	0	1	1	0	右转

2) 根据真值表编写 VHDL 代码，直至编译通过

```

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  entity FC is
7      port ( LS2,RS2: in std_logic;
8             IN1,IN3: out std_logic);
9  end FC;
10 architecture BEHAV of FC is
11 begin
12     IN1<=not LS2;
13     IN3<=not RS2;
27 end BEHAV;

```

3) 将代码导入 Quartus，分析并设置管脚。

out	IN1	Output	PIN_Y4	3	B3_NO	2.5 V (
out	IN3	Output	PIN_Y5	3	B3_NO	2.5 V (
in	LS2	Input	PIN_AB3	3	B3_NO	3.3-V I
in	RS2	Input	PIN_AA7	3	B3_NO	3.3-V I

4) 在小车上按照管脚和输入值对应关系连接好电路，其中 LED3 (LS2) – AB3, LED10 (RS2) – AA7, Y3 – (PCB) IN1, Y5 – (PCB) IN3。在 PCB 控制板上，ENA – 5V, ENB – 5V。最后连接地线，使 PCB 板和 FPGA 板共地。

5) 分析后生成程序，无误，用 USB 接口将其导入 FPGA 板。

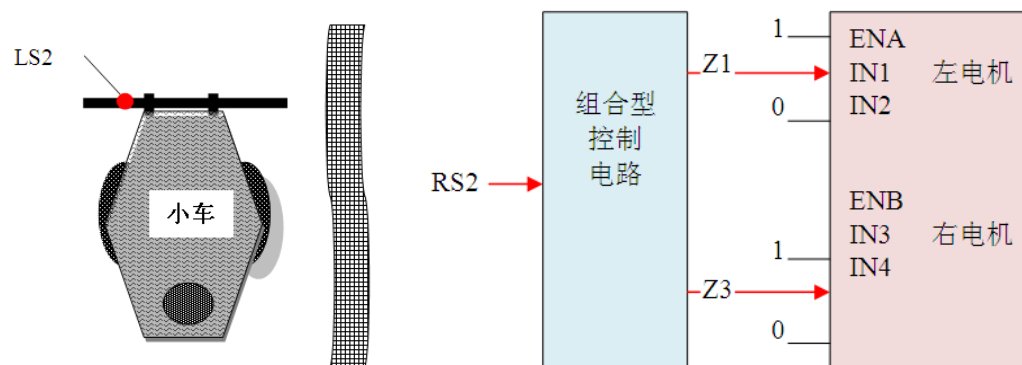
6) 将小车连接电源后在测试轨道（无岔路）上进行测试。

4. 实验数据处理与分析

小车能够按照要求进行前进（两灯均不在轨）、左转和右转。

内容四：实现寻迹轨道控制策略（一对光电管）

1. 实验要求



- 1) 只用 1 对光电传感器
- 2) 仅控制左右电机前行和停止 2 种状态
- 3) 实现前进、左转、右转三种运动模式

2. 设计思路

- 1) 选用位于小车 PCB 板两侧位置的 3 号光电对管 (LS2)。
- 2) 3 号管不位于轨道正上方时, 可以让小车右转, 直到 3 号管回到轨道正上方, 此时开发板的输入信号为“0”, 要求右电机不转, 左电机正转使其右转弯, 则开发板对左右电机驱动的输出信号分别为“10”“00”。
- 3) 3 号管位于轨道正上方时, 让小车左转, 传感器给开发板的输入信号为“1”, 要求左电机不转, 右电机正转使其左转弯, 则开发板对左右电机驱动的输出信号分别为“00”“10”。

```

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  entity FC3 is
7  port ( LS2: in std_logic;
8         IN1,IN3: out std_logic);
9  end FC;
10 architecture BEHAV of FC3 is
11 begin
12     IN1<=not LS2;
13     IN3<=LS2;
27 end BEHAV;

```

- 4) 这样一来, 小车预计是扭动着前进。

3. 实验过程

- 1) 根据设计思路列出输入输出的逻辑真值表

寻迹控制真值表（一对光电管）

检测状态	LS2	IN1 (L)	IN3 (R)	小车运动趋势
左在轨	0	1	0	左转
左不在轨	1	0	1	右转

- 2) 根据真值表编写 VHDL 代码, 直至编译通过

```

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  entity FC3 is
7  port ( LS2: in std_logic;
8         IN1,IN3: out std_logic);
9  end FC;
10 architecture BEHAV of FC3 is
11 begin
12     IN1<=not LS2;
13     IN3<=LS2;
27 end BEHAV;

```

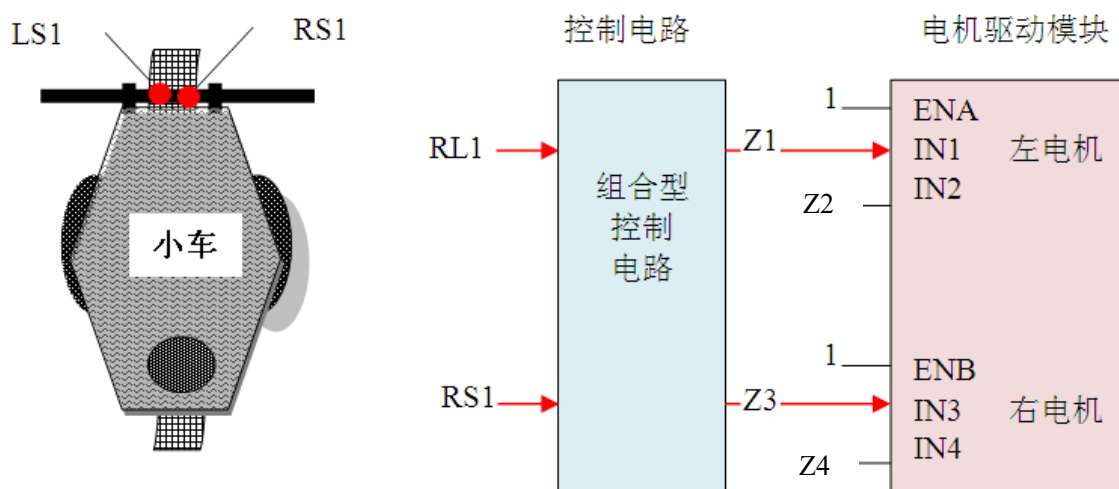
- 3) 将代码导入 Quartus, 分析并设置管脚。

out	IN1	Output	PIN_Y4	3	B3_N0	2.5 V (c
out	IN3	Output	PIN_Y5	3	B3_N0	2.5 V (c
in	LS2	Input	PIN_AB3	3	B3_N0	3.3-V L

- a) 在小车上按照管脚和输入值对应关系连接好电路, 其中 LED3 – AB3,Y3 – (PCB) IN1,Y5 –

(PCB) IN3。在 PCB 控制板上，ENA – 5V，ENB – 5V。最后连接地线，使 PCB 板和 FPGA 板共地。

- 4) 分析后生成程序，无误，用 USB 接口将其导入 FPGA 板。



- 5) 将小车连接电源后在测试轨道（无岔路）上进行测试。

4. 实验结果

- 1) 小车在直道上能够弯弯扭扭前进
- 2) 在 90° 转弯处因为速度不足在原地扭动
- 3) 在终点处不断扭动。

内容五：改进跟随控制策略

1. 实验要求

- 1) 只用 2 对光电传感器
- 2) 仅控制左右电机前行、停止和后退 3 种状态
- 3) 实现前进、左转、右转三种运动模式

2. 设计思路

- 1) 选用位于小车 PCB 板中间位置的 6 号和 7 号光电对管。
- 2) 两个对管都位于轨道正上方时，说明小车正沿着轨道直线前进，两个传感器给开发板的输入信号均为“11”，要求两个电机都正转使其前进，则开发板对电机驱动的输出信号均为“10”。
- 3) 当左侧对管位于轨道正上方而右侧对管不在轨道正上方时，说明小车需要左转，两个传感器给开发板的输入信号为“10”，要求左电机反转，右电机正转使其左转弯，则开发板对左右电机驱动的输出信号分别为“01”“10”。
- 4) 当右侧对管位于轨道正上方而左侧对管不在轨道正上方时，说明小车需要右转，两个传感器给开发板的输入信号为“01”，要求右电机反转，左电机正转使其右转弯，则开发板对左右电机驱动的输出信号分别为“10”“01”。

3. 实验过程

- 1) 根据设计思路列出输入输出的逻辑真值表

改进跟随轨道控制真值表

检测状态	LS1	RS1	IN1 (L)	IN2 (L)	IN3 (R)	IN4 (R)	小车运动趋势
在轨	1	1	1	0	1	0	前进
左在轨	1	0	0	1	1	0	左转
右在轨	0	1	1	0	0	1	右转

2) 根据真值表编写 VHDL 代码，直至其编译通过

```

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  entity FC4 is
7      port (  LS1,RS1: in std_logic;
8              IN1,IN2,IN3,IN4: out std_logic);
9  end FC4;
10 architecture BEHAV of FC4 is
11 begin
12     IN1<=RS1;
13     IN2<=not RS1;
14     IN3<=LS1;
15     IN4<=not LS1;
29 end BEHAV;

```

3) 将代码导入 Quartus，分析并设置管脚。

out	IN1	Output	PIN_y4	3	B3_NO
out	IN2	Output	PIN_Y3	3	B3_NO
out	IN3	Output	PIN_Y5	3	B3_NO
out	IN4	Output	PIN_Y6	3	B3_NO
in	LS1	Input	PIN_AA5	3	B3_NO
in	RS1	Input	PIN_AA6	3	B3_NO

4) 在小车上按照管脚和输入值对应关系连接好电路，其中 LED6-AA5,LED7-AA6,Y4-(PCB) IN1,Y3-(PCB) IN2,Y5-(PCB) IN3,Y6-(PCB) IN4。在 PCB 控制板上，ENA-5V，ENB-5V。最后连接地线，使 PCB 板和 FPGA 板共地。

5) 分析后生成程序，无误，用 USB 接口将其导入 FPGA 板。

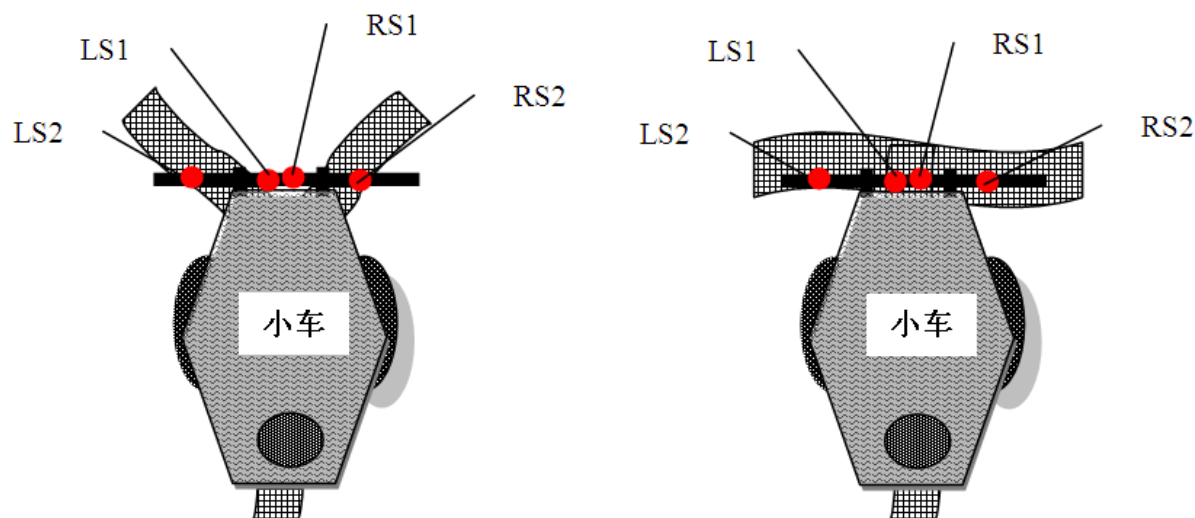
6) 将小车连接电源后在测试轨道（无岔路）上进行测试。

4. 实验数据处理与分析

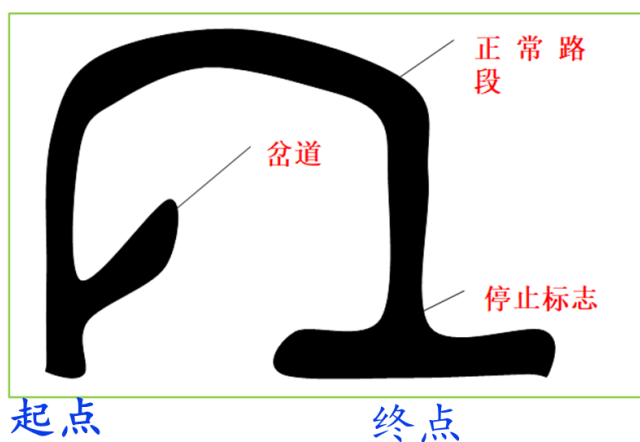
小车能够按照要求进行前进（两灯均在轨）、左转和右转，并且在大弯时更容易转过去。

内容六：实现复杂轨道导航控制

1. 实验要求



- 1) 用 4 对光电传感器
- 2) 实现前进、左转、右转、（到终点处）停止四种运动模式
- 3) 能辨别岔路



2. 设计思路

- 1) 选用 3 号、6 号、7 号和 10 号光电对管，且将 IN2 和 IN4 置零。
- 2) 6、7 对管都位于轨道正上方，且 3、10 对管不在轨道上方时，说明小车正沿着轨道直线前进且没有遇到岔路，传感器给开发板的输入信号分别为“0110”，此时要求两个电机都正转使其前进，则开发板对电机驱动的输出信号均为“10”。
- 3) 3、6、7 对管都位于轨道正上方，10 对管不在轨道上方时，说明小车正沿着轨道直线前进且左侧遇到岔路，传感器给开发板的输入信号分别为“1110”，此时要求两个电机都正转使其前进，则开发板对电机驱动的输出信号均为“10”。
- 4) 6、7、10 对管都位于轨道正上方，3 对管不在轨道上方时，说明小车正沿着轨道直线前进且右侧遇到岔路，传感器给开发板的输入信号分别为“0111”，此时要求两个电机都正转使其前

进，则开发板对电机驱动的输出信号均为“10”。

- 5) 当 6 对管位于轨道正上方而 7 对管不在轨道正上方时，说明小车需要左转，两个传感器给开发板的输入信号为“10”，左电机不转，右电机正转使其左转弯，则开发板对左右电机驱动的输出信号分别为“00”“10”。
- 6) 当 7 对管位于轨道正上方而 6 对管不在轨道正上方时，说明小车需要右转，两个传感器给开发板的输入信号为“01”，右电机不转，左电机正转使其右转弯，则开发板对左右电机驱动的输出信号分别为“10”“00”。
- 7) 当 3、6、7、10 对管均在轨道上方或均不在轨道上方时（冲出终点），说明已经到终点了，小车要停下来。四个传感器给开发板的输入信号分别为“1111”，要求左右电机均不转，则开发板对左右电机驱动的输出信号均为“00”。

3. 实验过程

- 1) 根据设计思路列出输入输出的逻辑真值表

复杂轨道控制真值表

LS2	LS1	RS1	RS2	IN1 (L)	IN3 (R)	小车运动趋势
0	0	0	0	1	1	前进
1	0	0	0	0	1	左转
0	1	0	0	0	1	左转
0	0	1	0	1	0	右转
0	0	0	1	1	0	右转
1	1	0	0	0	1	左转
0	1	1	0	1	1	前进
0	0	1	1	1	0	右转
1	0	1	0	1	1	前进
1	0	0	1	1	1	前进
0	1	0	1	1	1	前进
1	1	1	0	1	1	前进
1	1	0	1	1	1	前进
1	0	1	1	1	1	前进
0	1	1	1	1	1	前进
1	1	1	1	0	0	停止

- 2) 根据真值表编写 VHDL 代码（这里采用直接列真值表法，方便特殊情况定制修改），直至其编

```

port ( LS1,LS2,RS1,RS2: in std_logic;
        IN1,IN3: out std_logic);
end CAR4;

architecture BEHAV of CAR4 is
begin
P1:process(LS1,LS2,RS1,RS2)
variable tmp:std_logic_vector(3 downto 0);
begin
tmp:=LS1&LS2&RS1&RS2;
case tmp is
    when "0000"=> IN1<='1';IN3<='1';
    when "1000"=> IN1<='0';IN3<='1';
    when "0100"=> IN1<='0';IN3<='1';
    when "0010"=> IN1<='1';IN3<='0';
    when "0010"=> IN1<='1';IN3<='0';
    when "0100"=> IN1<='0';IN3<='1';
    when "0010"=> IN1<='1';IN3<='0';
    when "0001"=> IN1<='1';IN3<='0';
    when "1100"=> IN1<='0';IN3<='1';
    when "0010"=> IN1<='1';IN3<='0';
    when "0001"=> IN1<='1';IN3<='0';
    when "1100"=> IN1<='0';IN3<='1';
    when "0110"=> IN1<='1';IN3<='1';
    when "0011"=> IN1<='1';IN3<='1';
    when "1110"=> IN1<='1';IN3<='1';
    when "1101"=> IN1<='1';IN3<='1';
    when "1011"=> IN1<='1';IN3<='1';
    when "0111"=> IN1<='1';IN3<='1';
    when "1111"=> IN1<='0';IN3<='0';
    when others => IN1<='X';IN3<='X';
end case;
end process;
end architecture BEHAV of CAR4;

```

译通过

out IN1	Output	PIN_Y4	3	B3_N0
out IN3	Output	PIN_Y5	3	B3_N0
in LS1	Input	PIN_AA5	3	B3_N0
in LS2	Input	PIN_AB3	3	B3_N0
in RS1	Input	PIN_AA6	3	B3_N0
in RS2	Input	PIN_AA7	3	B3_N0

3) 将代码导入 Quartus，分析并设置管脚。

4) 在小车上按照管脚和输入值对应关系连接好电路，其中 LED3-AB3, LED6-AA5, LED7-AA6, LED10-AA7, Y4- (PCB) IN1, Y5- (PCB) IN3。在 PCB 控制板上，ENA-5V, ENB-5V。最后连接地线，使 PCB 板和 FPGA 板共地。

5) 分析后生成程序，无误，用 USB 接口将其导入 FPGA 板。

6) 将小车连接电源后在测试轨道（无岔路）上进行测试。

4. 实验数据处理与分析

1) 正反向测试时，小车均能辨别岔路，在岔路处能够直行。

2) 第一次测试时，真值表中标黄的一组，当输入为 0011 时，我们采用的是“11”，两轮正转前进。其他地方都能正常实现功能，但是在正向过第一个大弯的时候我们发现转不过去了。观察发现，在过弯时候传感器输入已经成为“0011”，而我们原先设计中这个弯的输入是“0010”。修改代码和真值表后再次测试，可以顺利过弯。

3) 反向测试时，对称的“1100”采用“11”却可以直接过弯，这可能和赛道路面以及车身平衡性有关。

4) 到终点处均能停下。

四、实验拓展内容&思考题

查阅并简单描述无人驾驶的硬件框架，软件框架和系统构成

1. 硬件框架

无人驾驶系统的硬件主要以下几个模块组成

*自动驾驶计算机：负责进行自动驾驶计算工作，一般包含冗余度。

*定位模块：无人车定位通过很多方式实现，他们之间也是冗余度备份的，例如激光雷达定位、惯导系统定位等。

供电系统：负责给全车设备供电

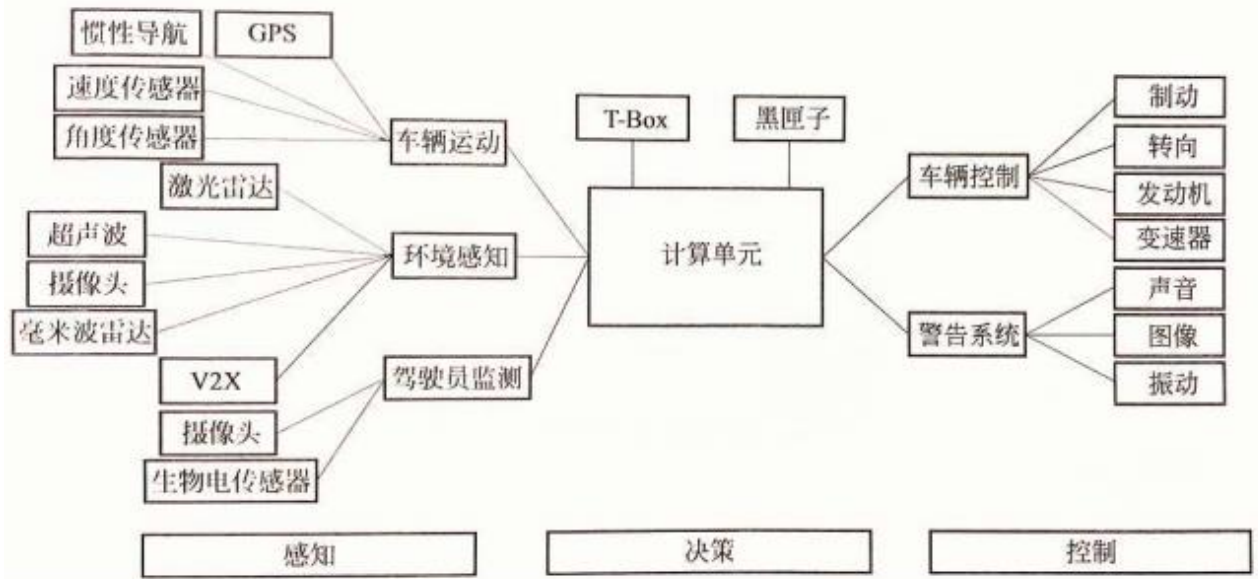
信号通信模块：负责计算机、传感器与执行机构间通信。

操纵及制动模块：包括电机、刹车等。

*感知模块：采用多种传感器，这使得无人（自动）驾驶系统可以检测、跟踪和识别周围物体，及时某一个传感器无法正常工作，也能够提供全方位的视角。这一部分一般通过摄像头系统或激光雷达、毫米波雷达系统实现。

*辅助避障模块：若其他模块失效，可提供紧急情况下的避障。

▪ 硬件架构



硬件架构示意图

1

2. 软件框架

感知层：以多传感器融合技术为主。将多种传感器得到的信息进行初步处理和融合后交送给决策层进行决策和判断。

决策层：处理感知层反馈的信息，可能涉及机器识别、卷积神经网络、深度学习融合算法等技术的运用。同时根据这些信息进行路径规划和行为决策。现在正尝试将人工智能技术引入这一部分。

3. 系统构成

由感知层、决策层、控制层组成。

感知层包括车上的各类传感器，以及多传感器融合技术。

决策层主要由无人驾驶计算单元（计算机）及其内部的算法构成。

控制层将决策层的决策传递给机械硬件部分，使车辆完成决策动作。

另外个人兴趣，找了一些具有自动驾驶功能的车型及其配套的自动驾驶技术解决方案：

北汽-极狐阿尔法 S HI：搭载华为鸿蒙 OS 系统，摄像头+毫米波雷达+激光雷达多种异构传感器的融合感知。

吉利-极氪 001：视觉感知为主（超级鹰眼系统 – VIDAR）

小鹏 P5：摄像头+毫米波雷达+激光雷达多种异构传感器的融合感知

奥迪 A8（L3 级）：超声波传感器+（红外夜视）摄像头+激光扫描仪+激光雷达

蔚来 ET7：摄像头+毫米波雷达+激光雷达多种异构传感器的融合感知（这款可以停车场全自动泊车，我坐过）（打开辅助驾驶的时候，可以在车机上看到前后约 200 米左右的车辆方位并判断其大致类型）

特斯拉：纯视觉感知方案，利用摄像头和机器识别、感知神经网络等技术

¹ 本图来源于浙江大学控制科学与工程学院冀海峰副教授开设的通识课程《传感技术导论》，是课件中的一部分

五、实验心得和感想

本次实验进行的较为顺利，一个非常重要的原因是，我们的小车传感器和电机都是正常的。试验全程只有在最后跑复杂轨道正向大转弯的时候出现一些问题，没能一把过弯。在检修排查过程中我们发现，真值表录入成代码时候有一行录错了。改了这行代码之后再试，依然没能过弯，在弯道处直接停下。这个时候我和队友仔细观察小车传感器和轨道的相对位置，发现在此处实际的传感器输入情况与我们预设的情况不符合。更改之后，小车能够过弯了。也就顺利跑完全程。这一次经历让我更深刻领会到，实验设计可能与现实有大的差距，我们要做的是在发现差距之后仔细分析然后修改设计，使之能够更贴近现实。并且不能想当然，因为实验情况下可能有很多偶然性。比如我们正向测试，修改了一行之后可以过弯；反向测试的时候，对应的对称的那种情况，输出不用修改就能过弯。

本次实验中，我对 ModelSim 和 Quartus 软件的使用更加得心应手，在改写 VHDL 代码时报错的概率也大大降低。在写 VHDL 代码时，简单的情况我尝试使用代入逻辑语句，而最后比较复杂的情况我采用了 case 语句，一是因为最后那个我转化不出来逻辑语句，二是复杂条件下可能会出现一些非规律性的情况，直接用 case 语句列真值表修改更便捷（有一组同学用逻辑语句，稍作修改就是一长串很难读懂的逻辑语句，还很难直观看对对错）。

在我们组结束实验后，我们还帮其他组一起分析解决问题，我在第三次实验时遇到的 Quartus 驱动安装的问题，在另一位同学的电脑上也遇到了，我很顺利地帮他解决了这一问题，这让我很有成就感，也让我感到，之前我在一些错误上耗费的时间并不是毫无意义的，在解决错误的尝试中，我积累了更多的经验，并且这些经验很可能在之后会派上用场。