Hardware Design

**Lab 6 Report**

Peripheral Components:

VGA, Mouse, and Dual FPGA

**Team 01**

112062122 謝佳晉 112062144 范升維

Table of Contents:

[1. Dual FPGA Communication 1](#_Toc183443841)

[2. Slot Machine 1](#_Toc183443842)

[3. The Car 1](#_Toc183443843)

[What have we learned from Lab 6? 1](#_Toc183443844)

[Contributions 2](#_Toc183443845)

# 1. Dual FPGA Communication

# 2. Slot Machine

# 3. The Car

We are asked to modify the sample code to make the car complete the tracks. The sample code are divided into four parts, which are sonic, motor, tracker sensor, and top.

In the sonic part, we modified the sonic\_top module. In order to let the car stop when the distance between it and obstacle in front of it is less than 40cm, we set *stop* to (dis < 20’d4000).

The overall design of sonic part is shown in Figure 3.1.

一張含有 文字, 螢幕擷取畫面, 字型, Rectangle 的圖片

自動產生的描述

▲ Figure 3.1: sonic part

In the tracker sensor part, we modified the tracker\_sensor module. We design our *state* to be 4bit to include in the *STAY\_L* and *STAY\_R* states, which we use when the sensors are all out of track. In addition to *state*, we also add signal *dir* to record which direction it should turn before the car goes out of track. It can helps us decide on whether *STAY\_L* and *STAY\_R* the car should switch to.

一張含有 文字, 螢幕擷取畫面, 寫生, Rectangle 的圖片

自動產生的描述

▲ Figure 3.2: diagram of *dir*

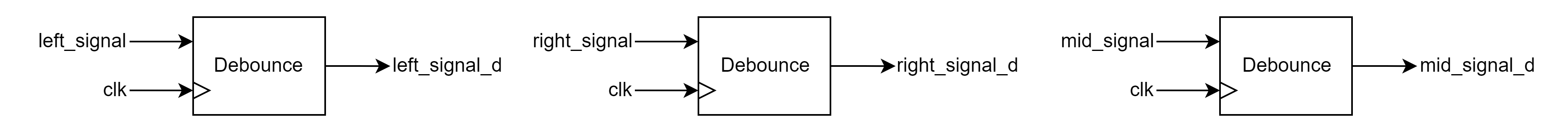
At first, our state transition is simply based on sensors’ output signals. However, we discover that the car will randomly turn its direction before going back to track from the black area. It’s because of the scratching on black areas.

一張含有 圖形, 字型, 設計 的圖片

自動產生的描述

▲ Figure 3.3: scratching the black area

When the sensors passes the scratching, they will output *1’b1*. The not-wanted outputs are like glitches. Therefore, after a long term of testing, we add debounce on sensors’ outputs and let the car keep its state on *STAY\_L* and *STAY\_R* until the sensors are all on track.



一張含有 螢幕擷取畫面, 寫生, Rectangle, 正方形 的圖片

自動產生的描述

▲ Figure 3.4: sensor signals debounce and *state* design

In the motor part, we modified the motor module. We design different pwm combinations according to *mode*(state). For states of one sensor out, we set one of the wheel’s pwm to *10’d1023*(100%) and the other wheel’s to *10’d700*(68.4%). For states of two sensors out, we set one of the wheel’s pwm to *10’d1023*(100%) and the other wheel’s to *10’d600*(58.7%). For states of all sensors out, we set both of the wheels’ pwms to *10’d1023*(100%) and invert one of the wheel’s direction. In this way, the car can turn back to track faster.

一張含有 寫生, Rectangle, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述一張含有 文字, 寫生, 螢幕擷取畫面, Rectangle 的圖片

自動產生的描述

▲ Figure 3.5: motor\_part

In the top part, we combine the above three parts, add some signals(*LED\_L*, *LED\_M*, *LED\_R*, *stop\_b*) for debugging, and change left motor’s and right motor’s direction according to the states and stop signals. When *stop* and *stop\_b* are positive, {*left*, *right*} is changed to *4’d0*.

When the car is in state *STAY\_L* and *STAY\_R*, {*left*, *right*} is changed to *4’b0110* and *4’b1001*.

Under other conditions, {*left*, *right*} is set to *4’b1010*. The overall design is shown in Figure 3.6.

一張含有 螢幕擷取畫面, 正方形, Rectangle, 圖表 的圖片

自動產生的描述

▲ Figure 3.6: top part

# What have we learned from Lab 6?

Haha

# Contributions

謝佳晉：

范升維：