

# Lab 6: Peripheral Components: VGA, Mouse, and Dual FPGA

Group 21: 陳克盈 (112062205)、蔡明妍 (112062224)

## Table of Contents

<b>1</b>	<b>Chip to Chip</b>	<b>2</b>
<b>2</b>	<b>Slot Machine</b>	<b>4</b>
2.1	Motification . . . . .	5
<b>3</b>	<b>Car</b>	<b>7</b>
3.1	Sonic . . . . .	7
3.2	Tracking Strategy . . . . .	8
3.3	Overall . . . . .	10
3.4	What We Have Learned . . . . .	11
3.5	分工 . . . . .	11

## 1 Chip to Chip

這題需要補齊 slave 接收與傳送訊號的部分，完成 master 和 slave 之間的溝通協議。

master 會傳送 request，slave 收到 request 後會回傳 ack 給 master，master 端接收到 ack 後再將 data 傳送給 slave，slave 收到 data 後 decoder 會處理並輸出給 seven\_segment 呈現在板子上。

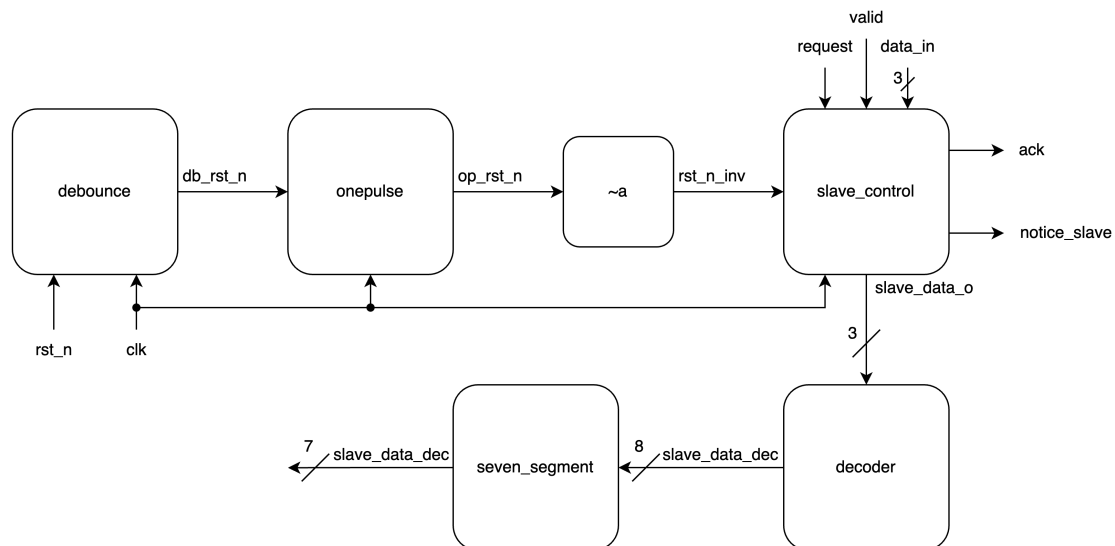
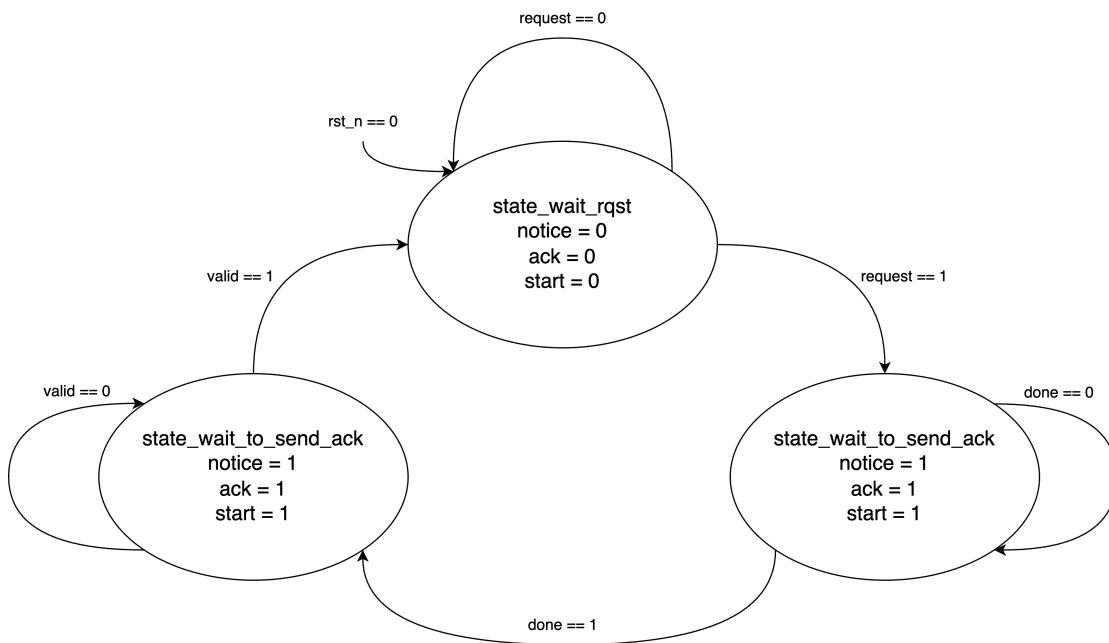


Fig. 1 Q1 Slave FPGA block diagram

對 **slave\_control** 的 FSM 設計如下：

- **state\_wait\_rqst**：等待 master 傳送 request
  - state：收到 master 傳來的 request 後，切換至 **state\_wait\_to\_send\_ack**
  - 收到 request 後，切換成 1 開始溝通，此時 counter 會開始運作，直到 `count == 27'd100000000` 時，將 **done** 設定為 1
  - notice：收到 request 後，使 LED[0] 亮起
  - ack：在這個 state 不做動作保持為 0
  - data：在這個 state 不做動作保持原本的数据
- **state\_wait\_to\_send\_ack**：等待 `done == 1` 並且傳送 ack
  - state：當 `done == 1` 時，切換至 **state\_wait\_data**
  - start：當 `done == 1` 時，start 切換回 0 使 counter 停止運作
  - notice：LED[0] 持續亮起直到 `done == 1`
  - ack：當 `done == 1` 時，將 ack 設成 1 傳送給 master
  - data：在這個 state 不做動作保持原本的数据

- state\_wait\_data：等待 master 傳送 data
  - state：收到 master 傳來的 valid 後，切換回 state\_wait\_rqst
  - start：在這個 state 不做動作保持為 0
  - notice：在這個 state 不做動作保持為 0
  - ack：直到收到 master 傳來的 valid 保持為 1
  - data：收到 valid 後，切換成從 master 傳送來的 data\_in



**Fig. 2** Q1 Slave Control State diagram

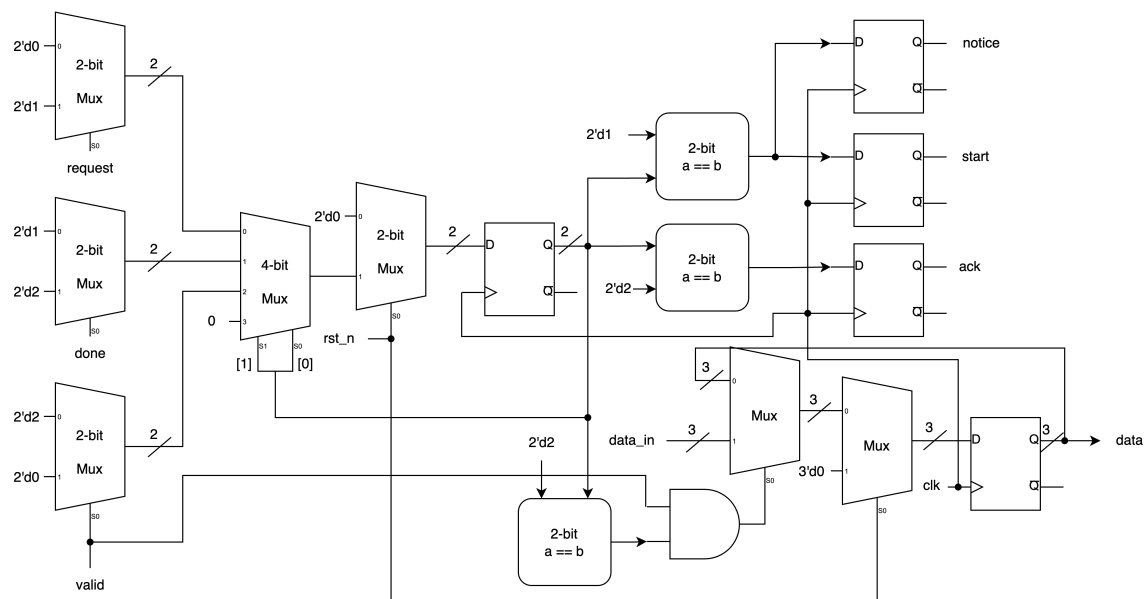


Fig. 3 Q1 Slave Control Circuit

## 2 Slot Machine

這題要在 FPGA 上實作使螢幕呈現往上的 777 動畫，板子上的 Top Button 為 Reset 訊號，Left Button 按下時要顯示往上的動畫，Right Button 按下時要顯示往下的動畫。

首先在原本的 sample code 上加入 Left Button 的 input (start\_up)，當 start\_up 為 1 時表示要顯示一個往上的動畫，當 start 為 1 時表示要顯示一個往下的動畫。

state\_control 處理完每幀畫面位置的 X\_v\_count 後，交給 mem\_addr\_gen 產生圖片與螢幕對應的正確記憶體位置，blk\_mem\_gen\_0 再根據輸出的 pixel\_addr 產生圖片 7 對應的 RGB，vga\_controller 則用於給予 mem\_addr\_gen 必要的輸入項，整體流程圖如下：

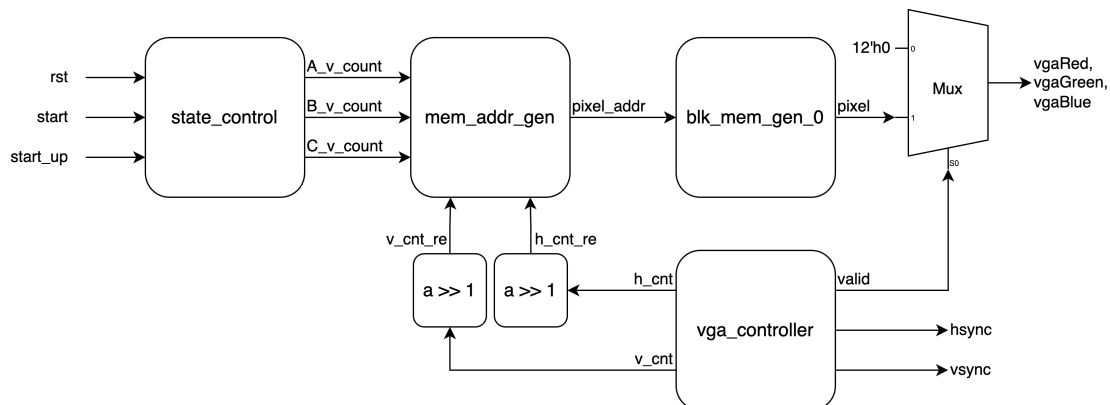


Fig. 4 Q2 Slot Machine FPGA block diagram

## 2.1 Motification

在 state\_control 中，原先的 A\_to, B\_to, C\_to 在 counter 等於 0 時，透過 start 決定是否開始動畫，加入 start\_up 後，將 A\_to, B\_to, C\_to 修改成當 start\_up 為 1 時也會開始動畫。

(為了避免簡報過於冗長，後面以 X 替代 A, B, C)

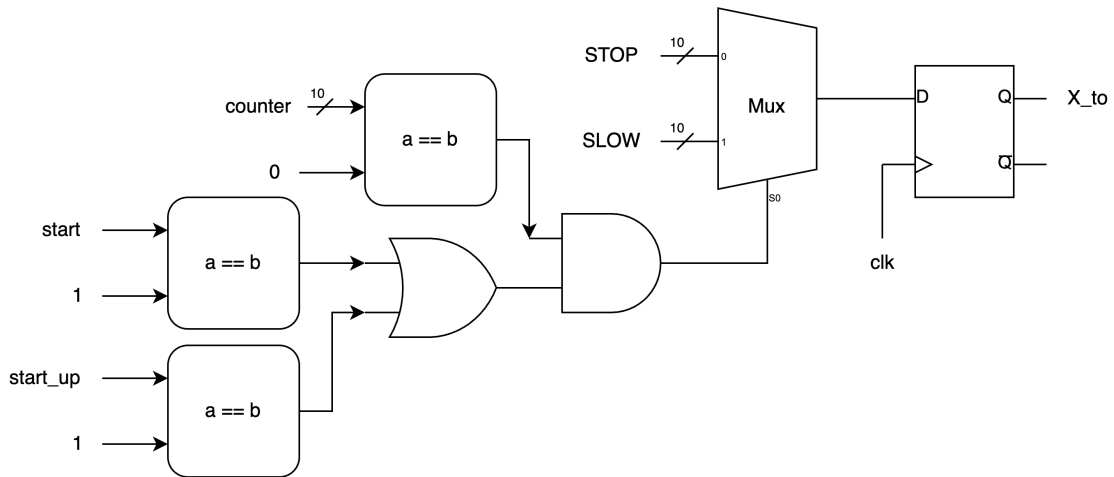


Fig. 5 Q2 Slot Machine X\_to Circuit

對於 next\_counter，因為加入了 start\_up，將原本 (start==1'b0 && counter==10'd0) 修改為 (start==1'b0 && start\_up==1'b0 && counter==10'd0)，使 start 和 start\_up 未按下並且 counter 為 0 時，讓 counter 歸 0 不再持續運作。

為了使 start\_up 按下後不需要經過 reset，直接按下 start 也可以繼續運作，將 counter >= 10'd1000 的情況發生時，使 counter 歸 0。

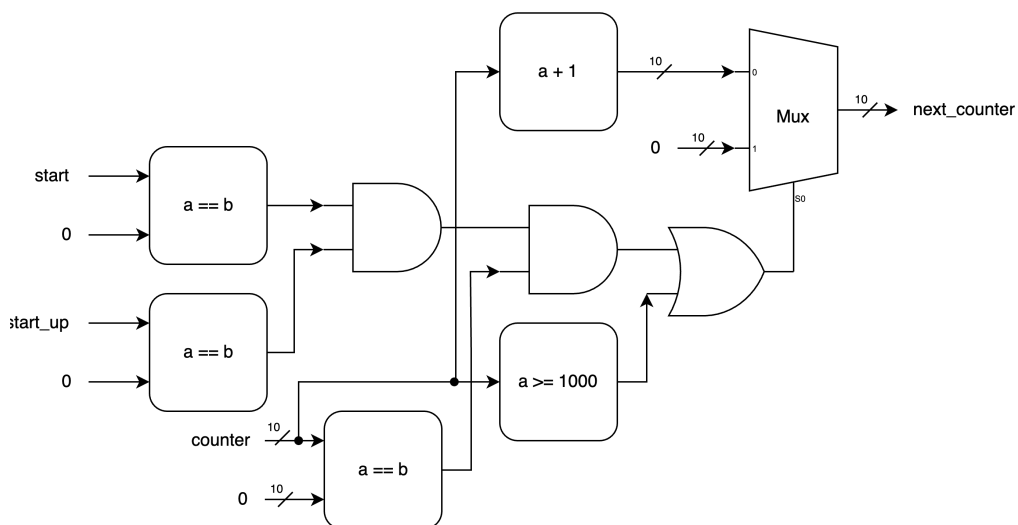
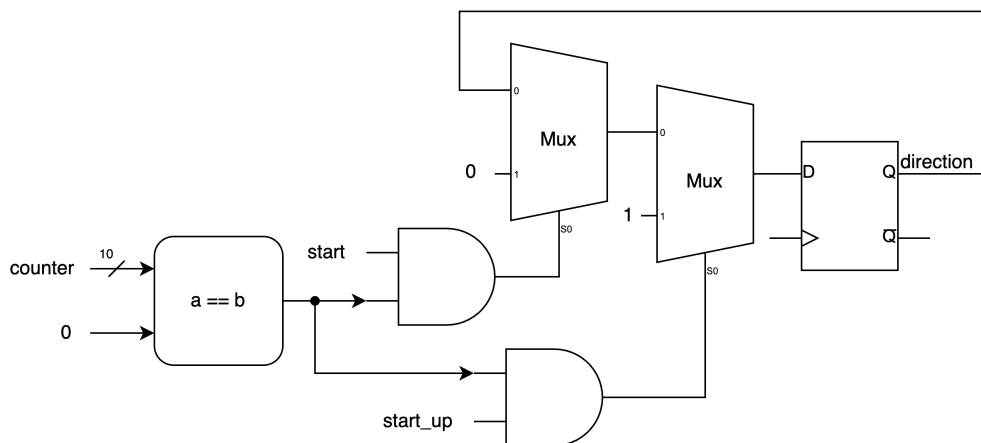


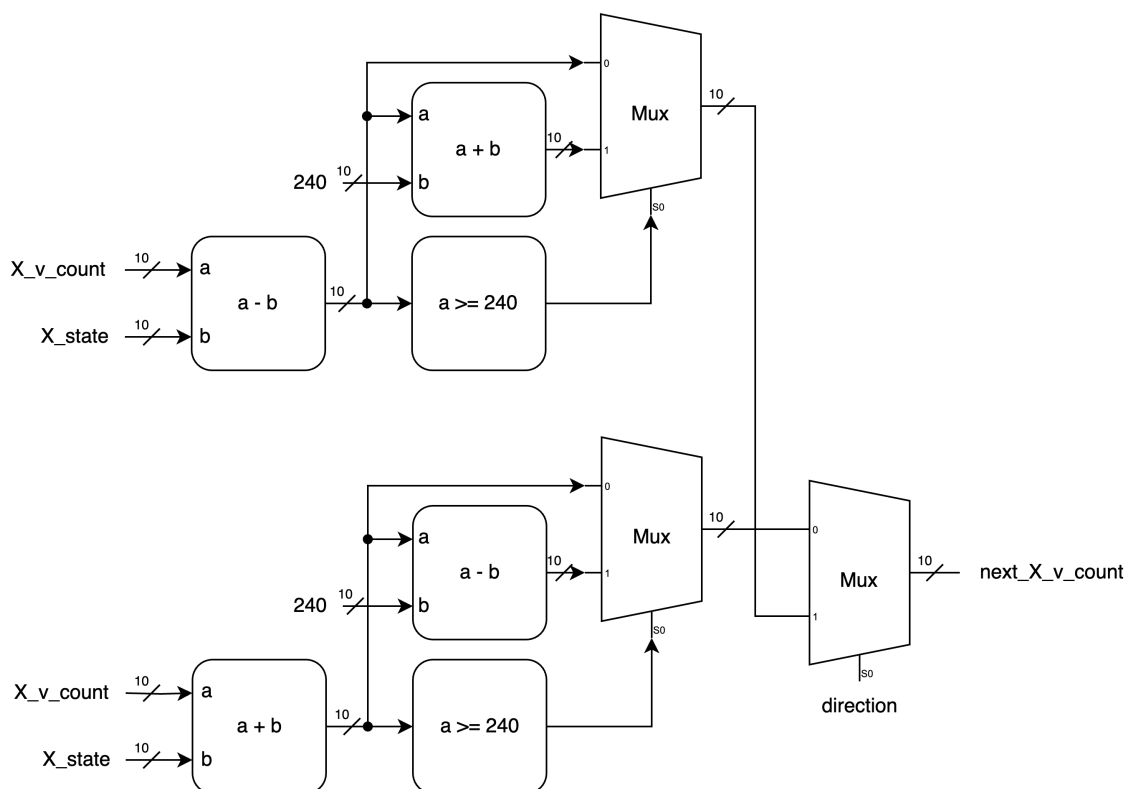
Fig. 6 Q2 Slot Machine next\_counter Circuit

我們加入一個 direction，當 counter 等於零且 start 被按下時，direction 設定為 0，且當 counter 等於零且 start\_up 被按下時，direction 設定為 1，其他情況 direction 維持原樣。



**Fig. 7** Q2 Slot Machine direction Circuit

接著新增了 up\_X\_v\_count 用來計算往上的 count，利用先前的 direction 變數用來判斷當下的 next\_X\_v\_count 會是往上還是往下的 count。



**Fig. 8** Q2 Slot Machine up\_X\_v\_count Circuit

### 3 Car

這部分我們需要使用 FPGA、超音波感應器、線路追蹤器以及馬達來組成一個自走車，使其能夠在寬度為 12 公分的白色賽道上自動行駛。

#### 3.1 Sonic

超音波感應器的部分由四個 module 組成：sonic\_top, PosCounter, TrigSignal, div：

div

將板子 100 MHz 的 clock 轉換為 1 MHz

TrigSignal

產生 10 us 的 Trig 信號，週期為 10ms

PosCounter

這個 module 會負責計算超音波回波的時間並轉換成距離。運作過程中分成了三個狀態：

- S0: 等待回波開始，當偵測到 Echo 的 posedge 時，就會進到 S1
- S1: 計算回波時間，每個 clk cycle 會將 counter 加一，直到偵測到 Echo 的 negedge，就會進到 S2
- S2: 計算距離，單位為 0.01 公分

sonic\_top

接收其他 module 計算出來的距離，判定如果距離小於 40 公分，就輸出停止訊號。不過由於資電館疑似有不乾淨的東西，導致超音波感應器在某些地方會偵測到不存在的物體，因此額外加了一個 **enable\_stop** 訊號，使其能夠透過板子上的開關來決定是否要開啟超音波感應器的功能。

## Block Diagram

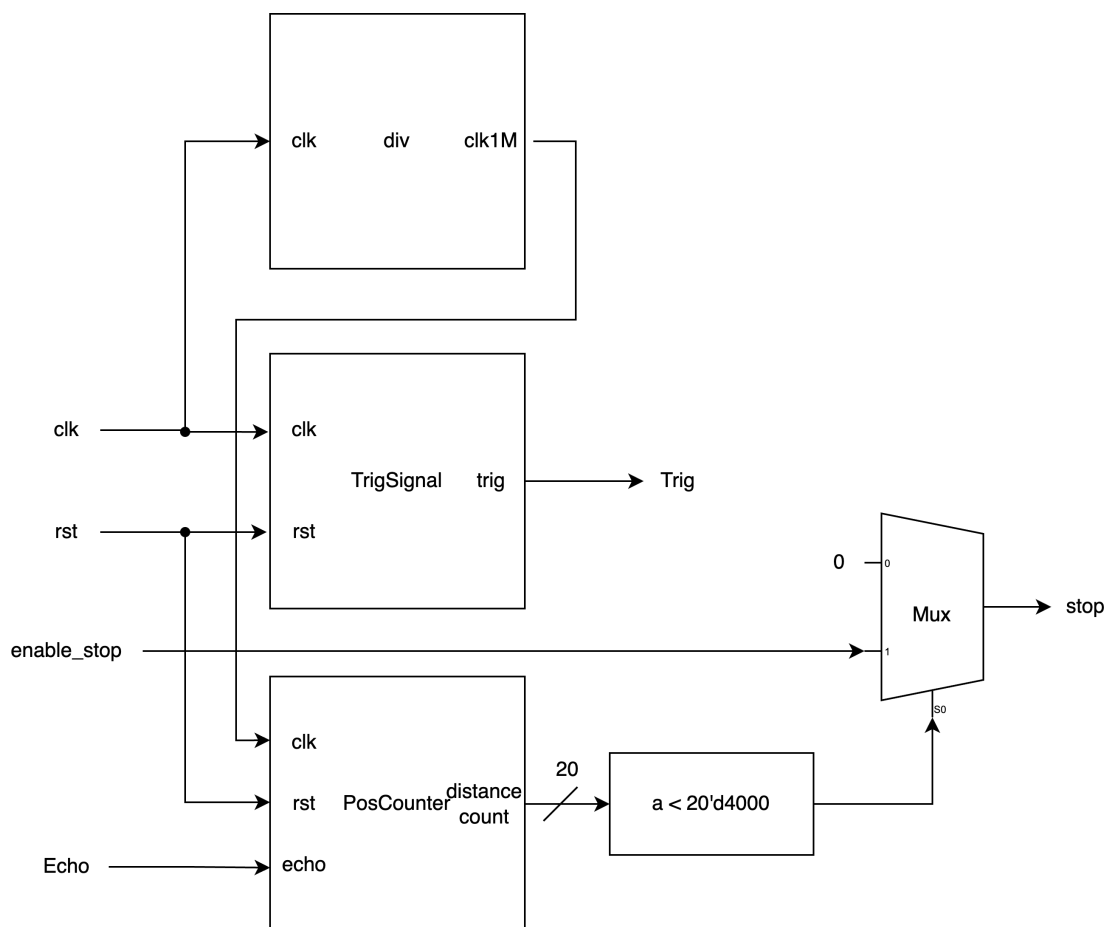


Fig. 9 Sonic

## 3.2 Tracking Strategy

我們利用三個 bit 的 state 來表示感應器的狀態，state[2] 代表 left, state[1] 代表 middle, state[0] 代表 right，針對不同的策略調整馬達的運轉方向，而無論如何，馬達的速度都設定為 1023（最大值）：

111

這代表三個感應器都偵測到黑線，此時車子會直行。

001

這代表車子已經向左偏移了不少，此時左輪會往前，右輪會往後，使車子更快的向右轉。

011

如果車子是從 111 的狀態轉變過來的，則右輪會停止，左輪會往前，使車子緩慢的向右轉。反之如果車子是從 001 的狀態轉變過來的，那代表還沒有完全轉回來，則會繼續保持左輪往前、右輪往後的狀態。

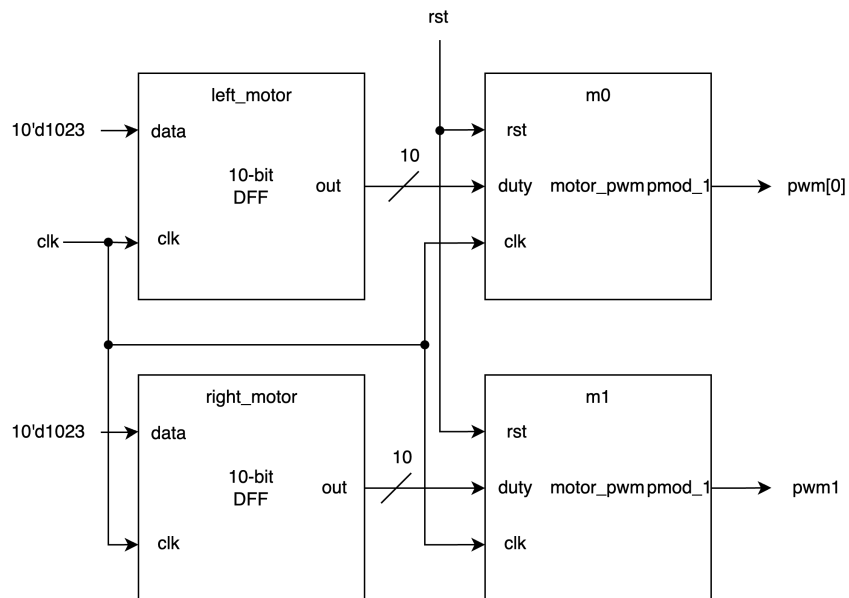


100, 110

這兩個狀態與 001, 011 相似，只是方向相反。

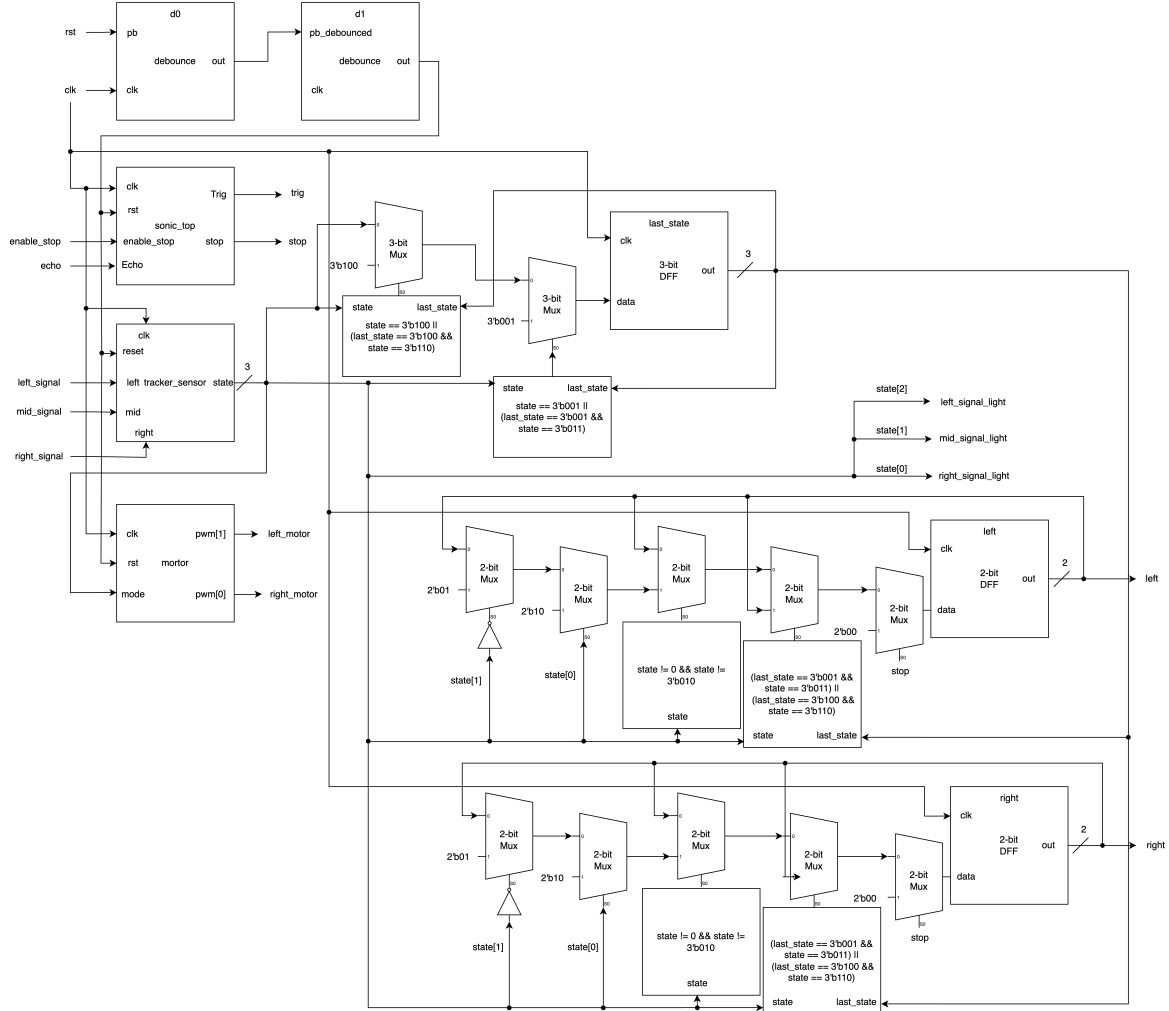
其它

保持不變。



**Fig. 10** Tracking Strategy

### 3.3 Overall



**Fig. 11 Overall**

### 3.4 What We Have Learned

- 學習到了如何實際應用 Handshaking protocol 在兩塊 FPGA 板上做數據傳輸，以及將圖片動畫透過 VGA 傳輸到電腦螢幕上顯示
- 超音波感應器的原理以及距離計算
- 如何使用線路追蹤器來控制馬達的運轉方向
- 如何設計一個簡單的 state machine 來控制車子的運行

### 3.5 分工

- 陳克盈：Car
- 蔡明妍：Chip-to-Chip, Slot Machine