**Hardware Design and Lab: Lab6**

**Team 28**

**111060013 EECS 26' 劉祐廷**

**111060002 EECS 26' 李侑霖**

**Catalog**

**1. FPGA Question: Slot Machine………………………….……P3**

**2. FPGA Question: Chip2chip……………………....………….P8**

**3. FPGA Question: Car………………………………...…….P9**

**4. What We Have Learned…………………………………P12**

1. **FPGA Question: Slot Machine**
2. **Block Diagram**

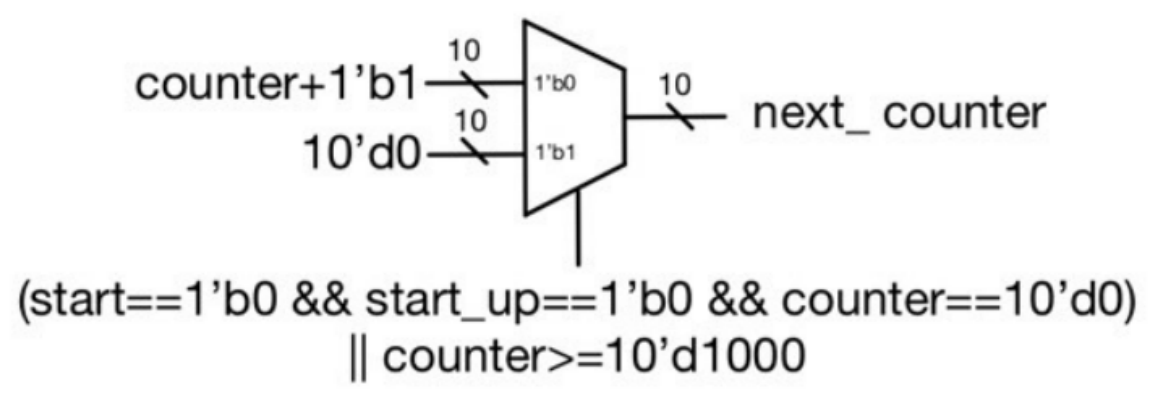
****

Figure 1.1

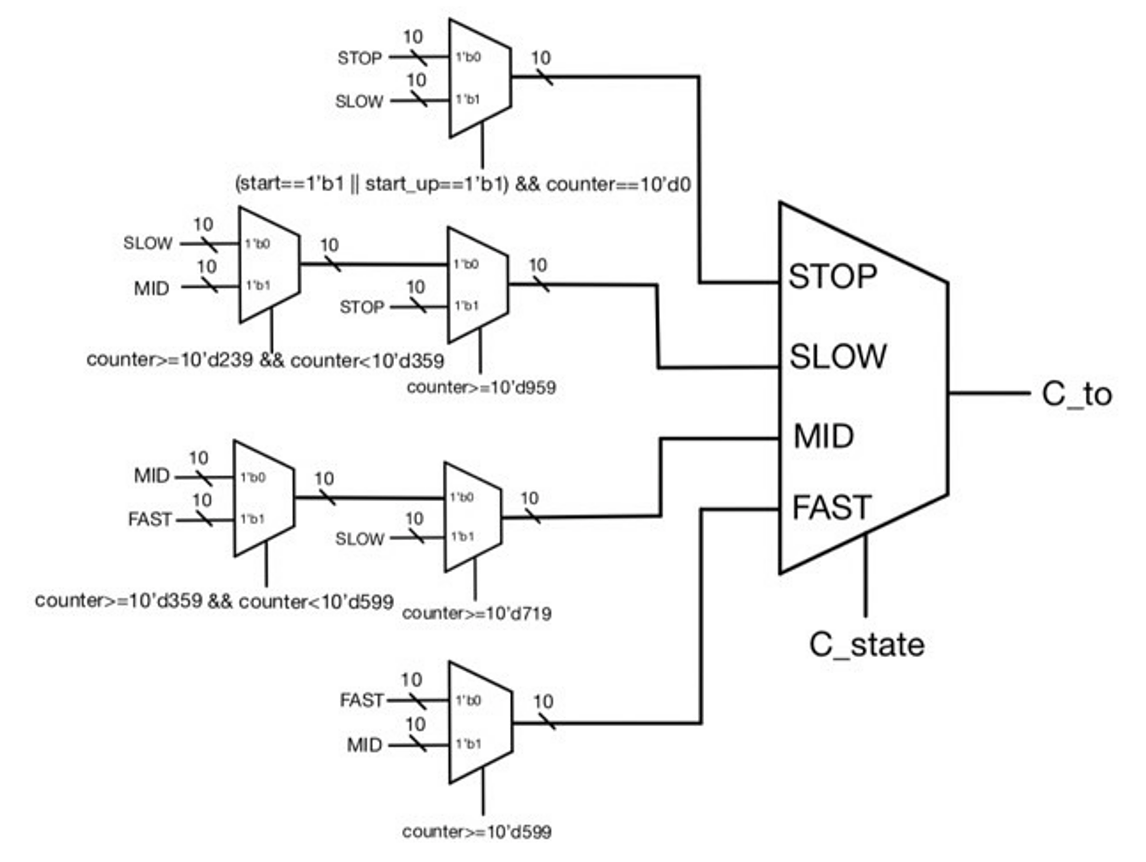
****

Figure 1.2

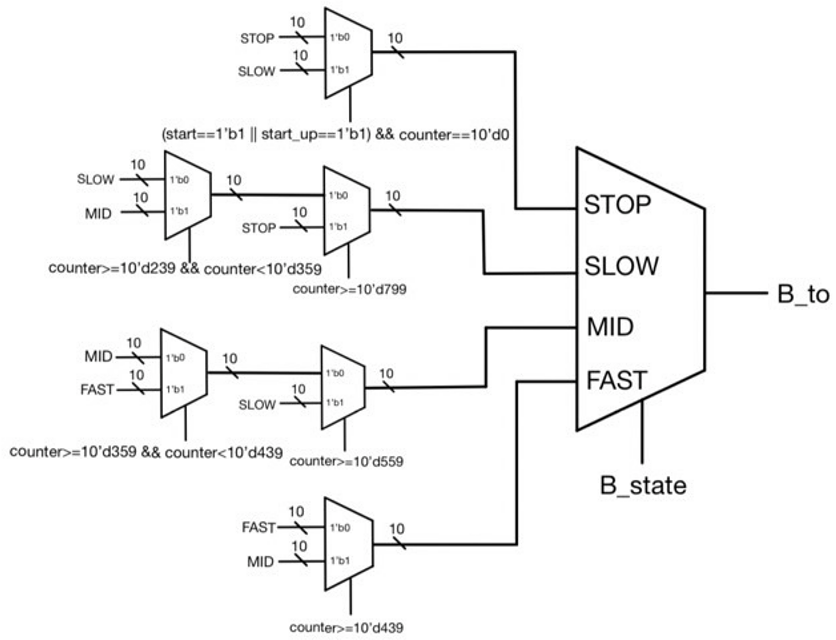
****

Figure 1.3

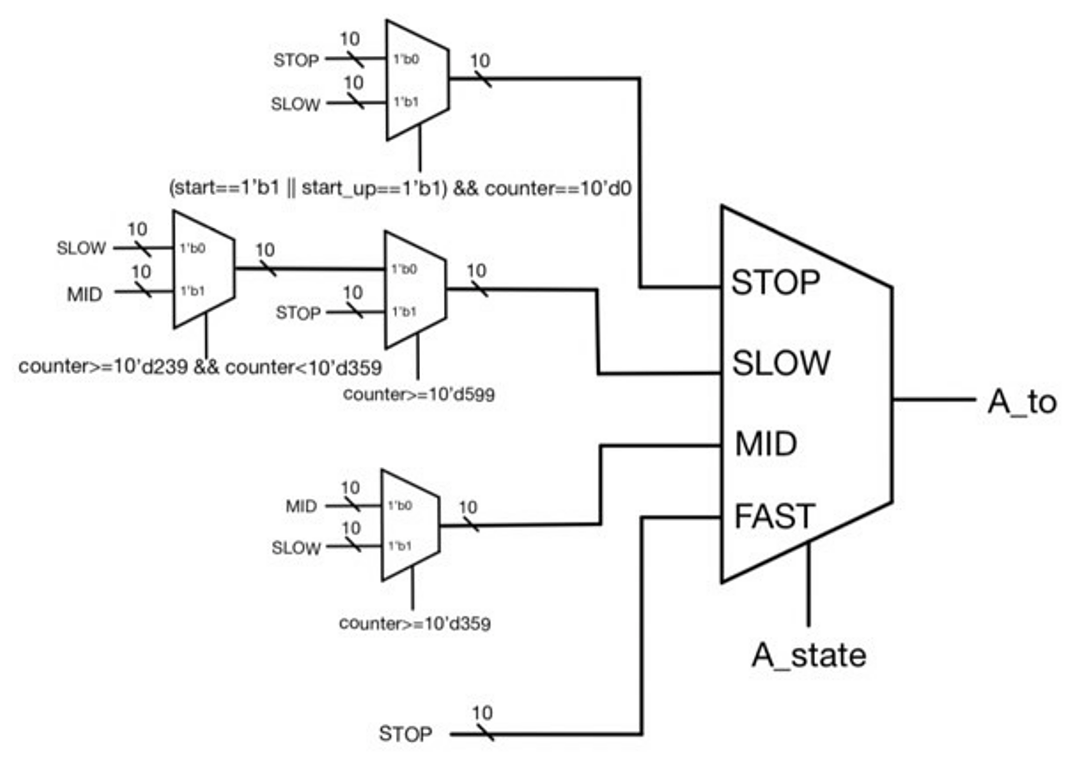
****

Figure 1.4

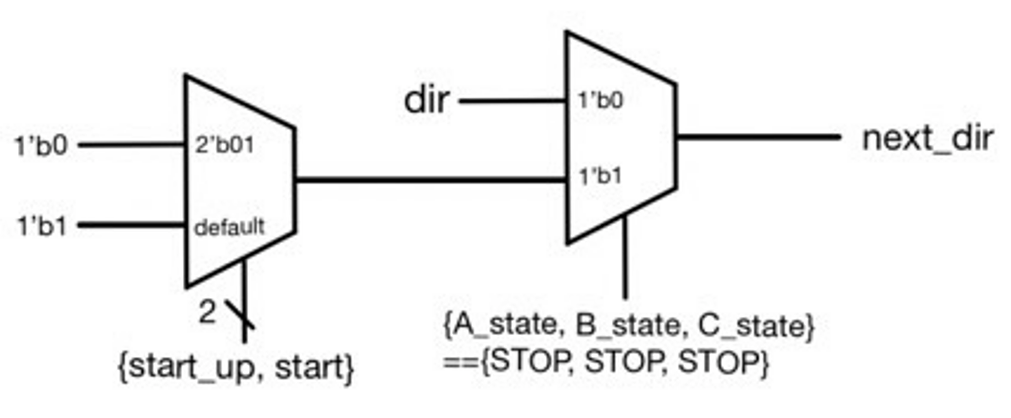
****

Figure 1.5

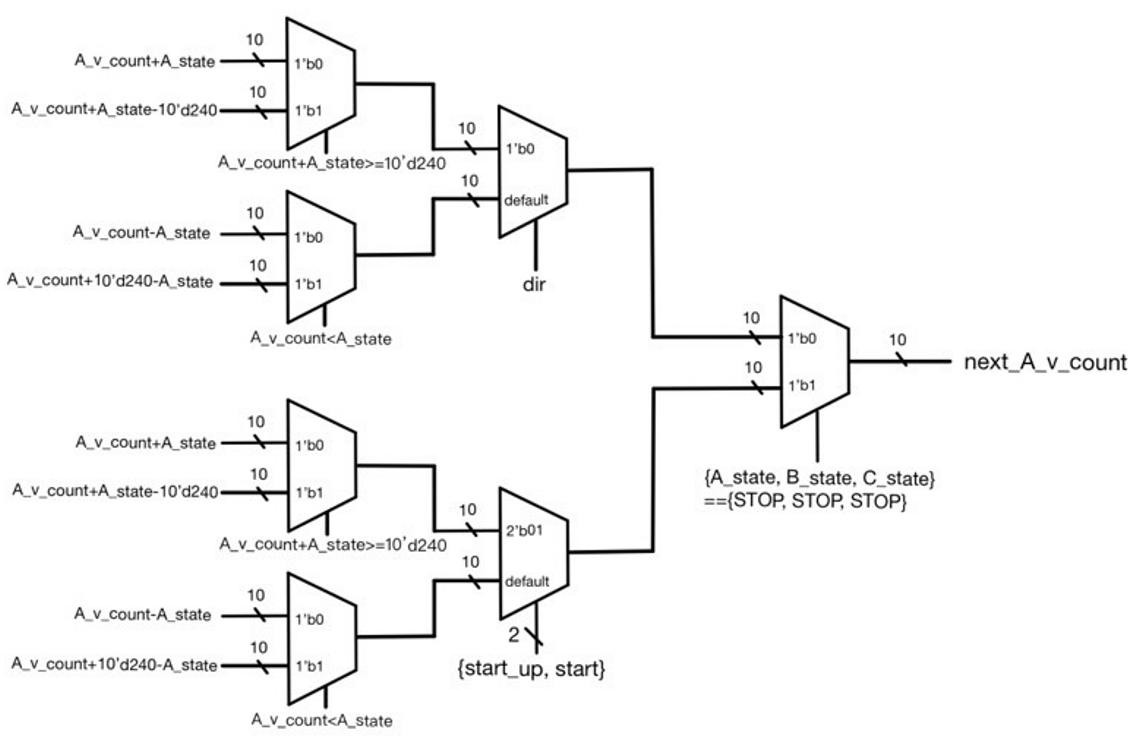
****

Figure 1.6

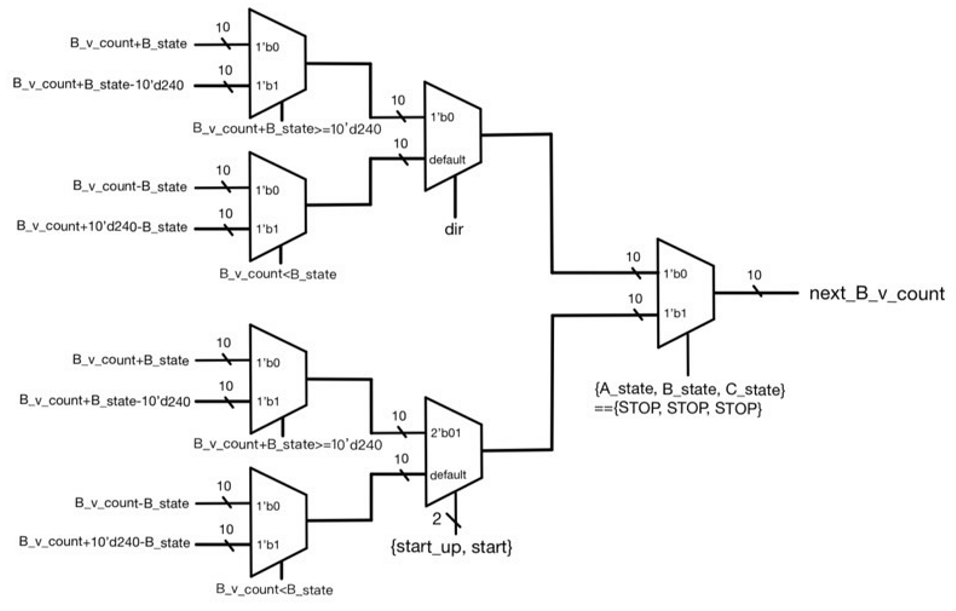
****

Figure 1.7

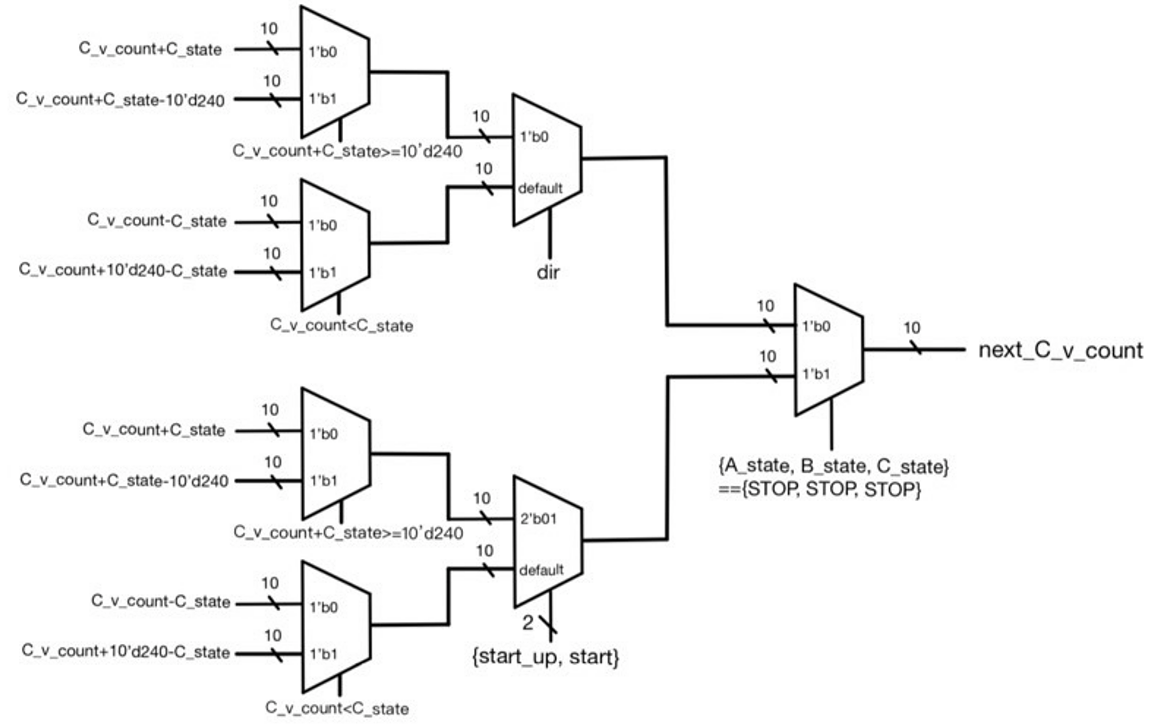
****

Figure 1.8

1. **Explanation**

在Slot Machine這題一開始的時候，我們注意到了在每次向下運轉後，都需要再按一次reset才能重新向下旋轉，經過trace code之後，我們發現 (Figure 1.1) counter 原本的運作邏輯有問題，因此我們將它改成當沒有任何start button被按下且counter為0時 (代表slot machine目前在停止狀態)，或是當counter數到1000時 (代表slot machine剛運轉完)，counter會被重新設為0，也就是原本等待button被按下時的狀態，以此達成不需要reset就可以連續使用向上旋轉或向下旋轉的功能。

為了實現向上旋轉的功能，我們新增了一個叫做start\_up的input來接受向上旋轉button的輸入，原本向下旋轉button的input就繼續沿用原本start這個名稱。因為加入了start\_up，所以對於A、B、C開始數數的條件我們改成了只要start或是start\_up為1時，並且counter是0時 (slot machine目前在停止狀態)，counter才會開始數數 (Figure 1.2 ~ Figure 1.4)。

為了要避免在旋轉狀態時，按下反方向的button導致slot machine出錯，我們多設了一個register叫做dir，用來儲存目前的狀態，如Figure 1.5所示，在slot machine為停止狀態時，才會依照按下的button去改變旋轉的方向，以此避免前述提到的問題。

Figure 1.6 ~ Figure 1.8展示了我們如何實現向上或向下旋轉，我們將原本的遞增改成了遞減，並稍微修改了A\_v\_count、B\_v\_count、C\_v\_count超出範圍的條件，並透過dir來保持運作時的方向，已完成向上旋轉的功能。

1. **FPGA Question: Chip2chip**
2. **Finite State Diagram**

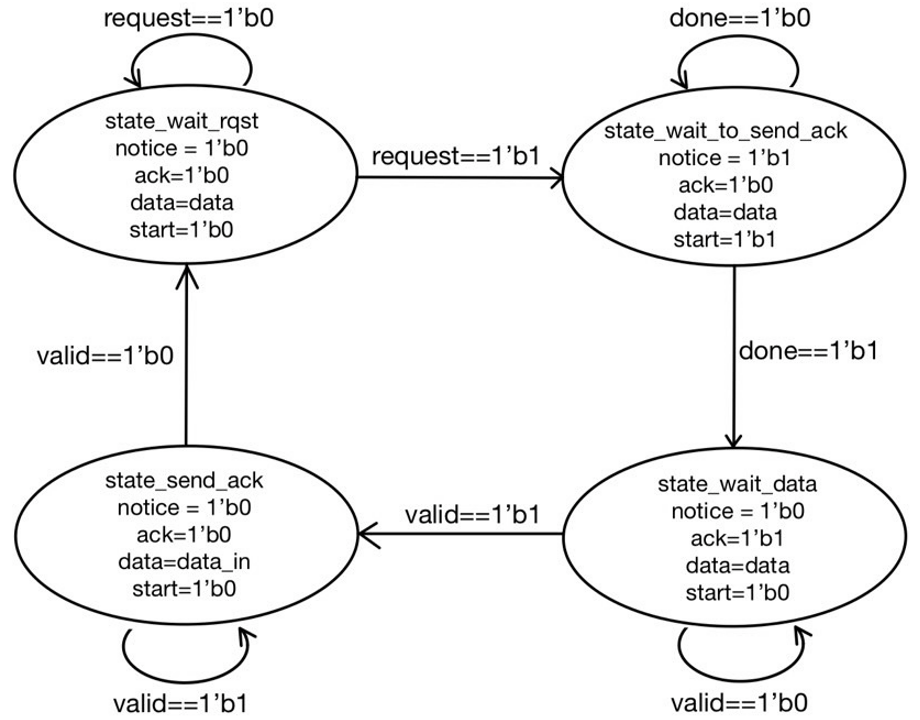
****

Figure 2.1

1. **Explanation**

在這題中我們使用moore machine來完成slave端的設計(Figure 2.1)，依照簡報中提供的finite state graph完成這次的code，其中done用來表示燈是否亮完1秒，亮完之後slave端會把ack設成1，代表slave端有收到request，接著進入等待傳data的狀態，當master把valid設成1時，slave會去抓data\_in並把ack設成0，此時master會將valid設成0，然後slave就會回到等待request的狀態。

1. **FPGA Question: Car**
2. **Finite State Diagram**

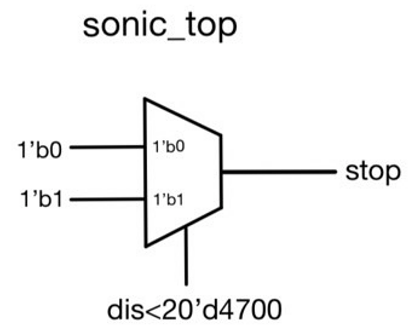
****

Figure 3.1

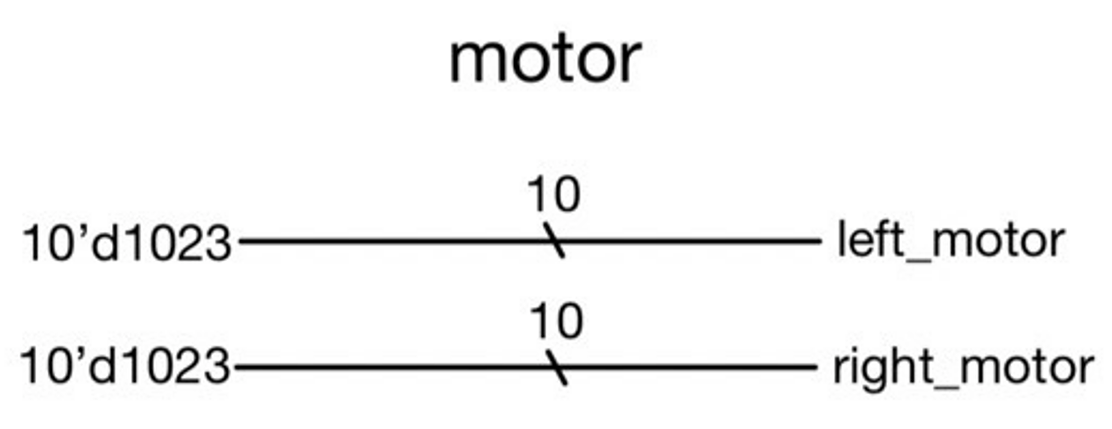


Figure 3.2

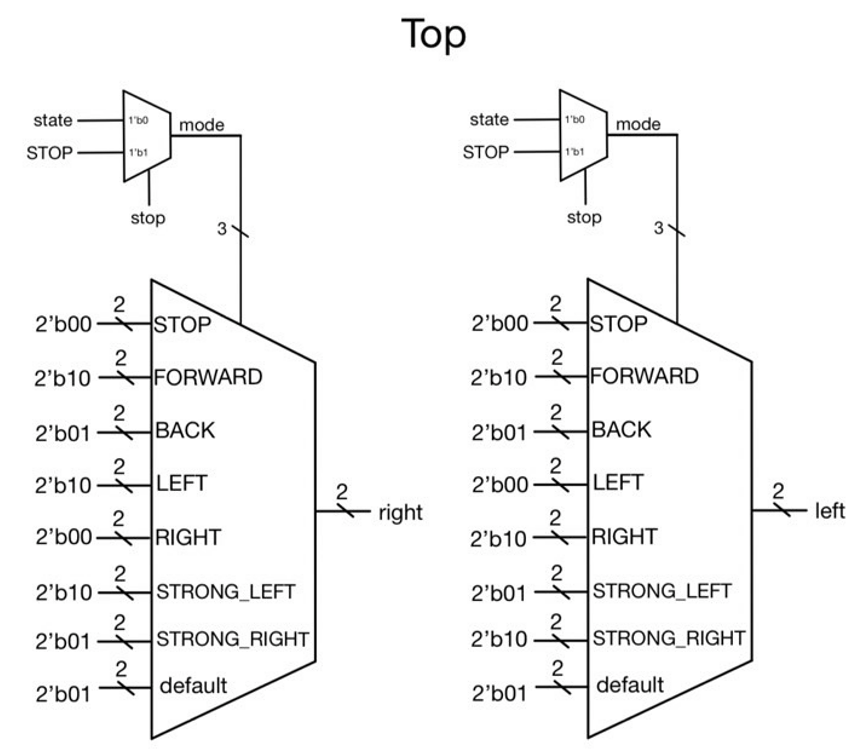


Figure 3.3

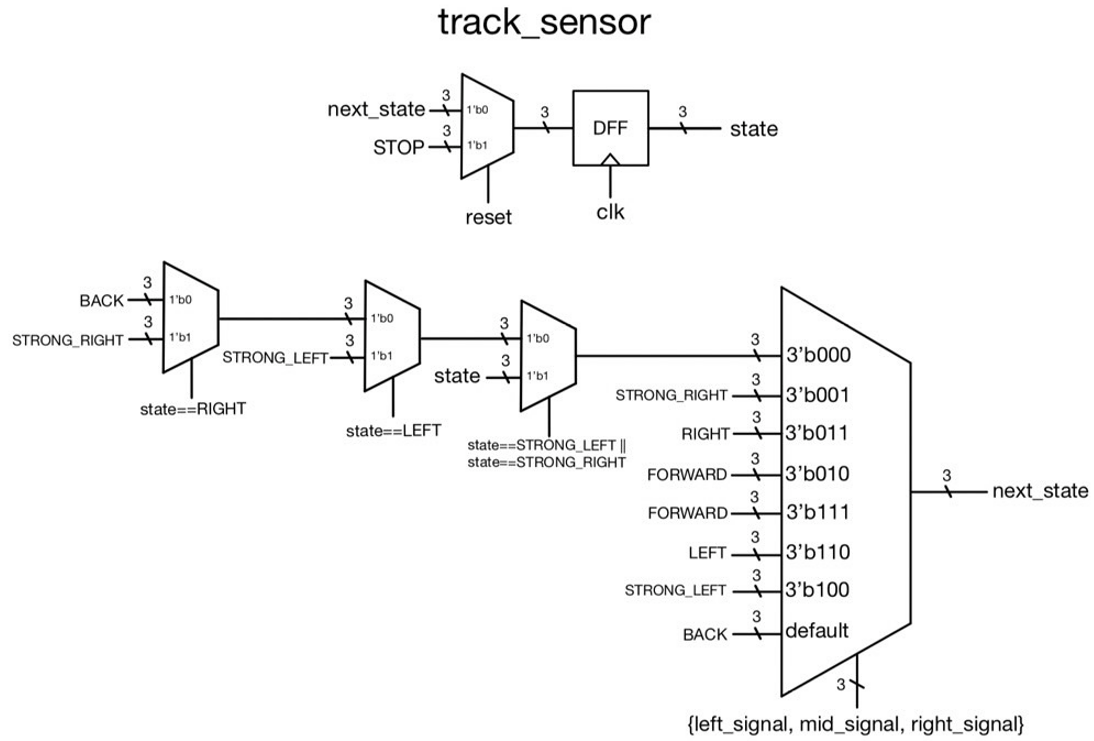


Figure 3.4

1. **Explanation**

Figure 3.1展示我們超聲波模組的判斷條件，經過計算，回傳的值若小於約4700時，車車就必須停下(小於40cm)。

Figure 3.2為我們對於馬達速度的設計，由於我們拿到的馬達動力有問題，因此測試的時候都是把left\_motor與right\_motor設成1023，沒有另外設計不同狀態下的速度，前後轉彎都是使用top module裡面的判斷去完成的。

Figure 3.3展示了我們如何在top module裡面進行行進狀態的判斷，如果超聲波模組接出來的stop為1時，代表有障礙物距離<40cm，因此無論track sensor接出來的state為何，都需要將狀態切成STOP。

Figure 3.4展示了我們車車運行的策略，當track sensor三顆感應器均為1或是只有中間的sensor為1時，state會被設成FORWARD。若只有右邊的sensor為0時，state會設成LEFT，此時左邊的馬達會關掉，剩右側的馬達前運作，以達成左轉修正的目的，而若只有左邊sensor感測到0的話則反之。若只有右邊的sensor為1時，此時我們認為需要做急轉彎，因此state會設成STRONG\_RIGHT，此時左邊的馬達會向前轉，右側的馬達會向後轉，以達成急右轉修正的目的，而若只有左邊sensor感測到1的話則反之。最後一個情況是三個sensor都是0的時候，此時我們會依照上一個clock的state來做判斷，若是FORWARD的話，車車會到退修正，尋求能回到路線上的機會，若上一個state是LEFT或STRONG\_LEFT的話，則會把state設成STRONG\_LEFT，避免不必要的倒退增快通過急彎的速度，上一個狀態是RIGHT或STRONG\_RIGHT的話則反之。

1. **What We Have Learned**
2. **李侑霖**

這次lab我學到很多東西，除了對螢幕有更深的了解之外，也學到了利用2個板子互傳資料的方法，也學到如何處理丟失的資料和建立連線，此外，車子的部分也讓我知道其基本的原作原理，並且這次lab需要與隊友的大量溝通與磨合，互相激勵思考也是我在這次lab學習到的東西。

1. **劉祐廷**

這次的lab我學到了如何使用螢幕以及，兩個板子的傳輸，這學期剛好在修計網概，能自己實作一個傳輸的模組真的很酷，感覺有把所學的東西運用進實作，而這次的車車其實暑假再辦營隊的時候就有做過相似的東西了，所以在思考跟打code的時候非常順暢，然而我們測試時車車的馬達很弱而且倒退會卡住，暑假那兩個營隊的經驗告訴我，這大概率是硬體的問題，因為這種東西的品質很不穩定，後來去跟朋友借車測試，確定了我們的馬達是有問題的，而從這件事我也意識到人脈的重要性，在未來的路上，若能有朋友的相互扶持，應該能夠更順利的度過每個難關。

**此次Lab都是一起做的，所以工作分配為分別各占50%**