

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

A. Block Diagram

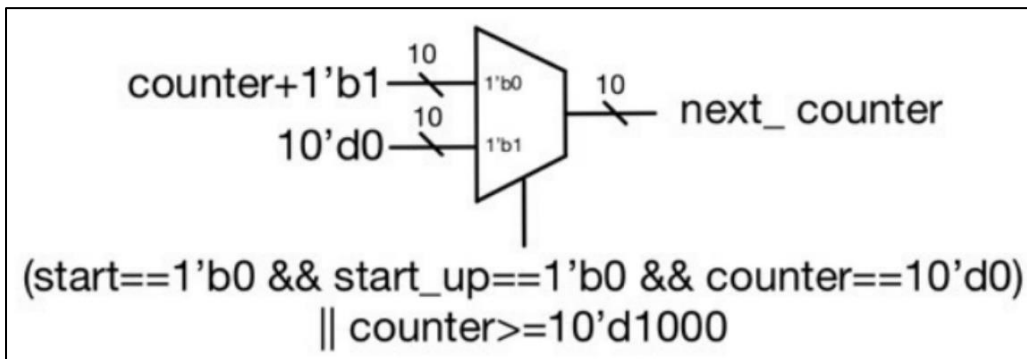


Figure 1.1

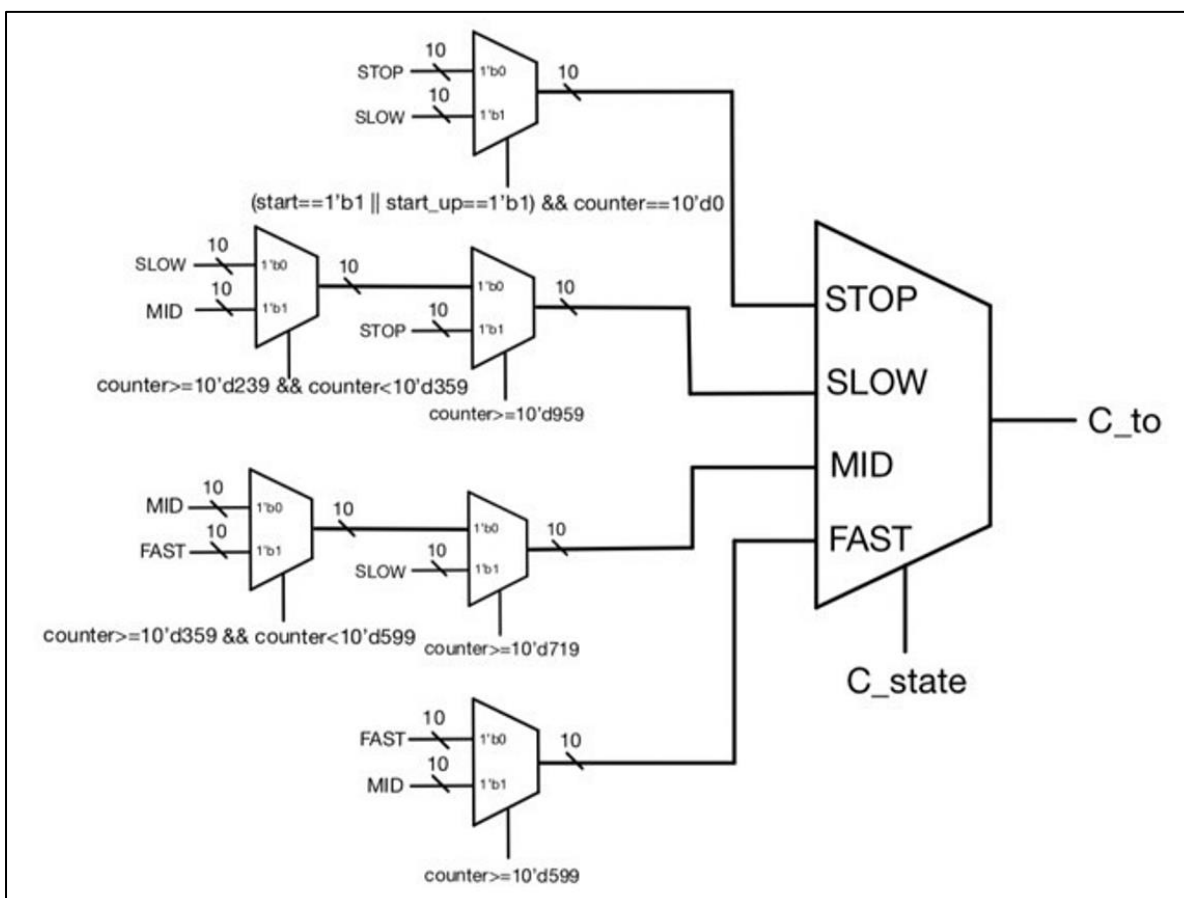


Figure 1.2

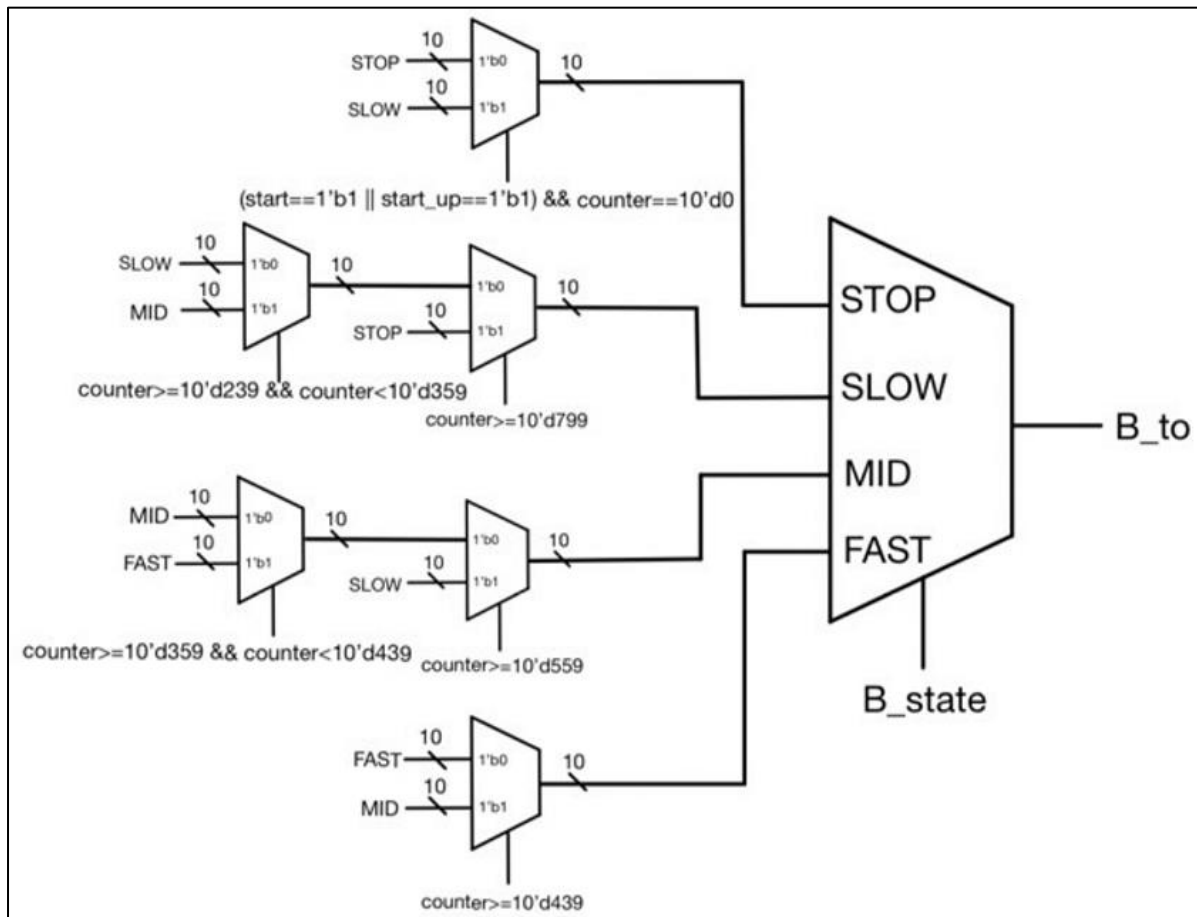


Figure 1.3

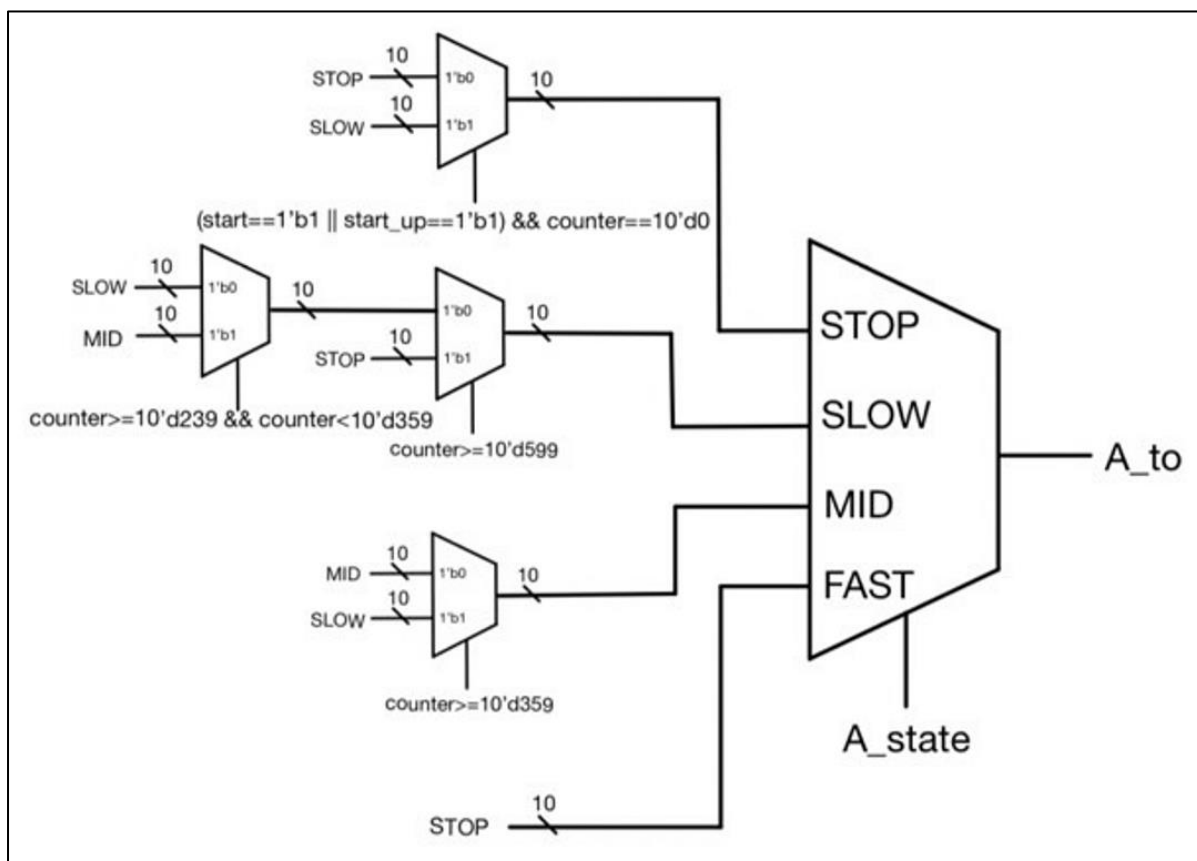


Figure 1.4

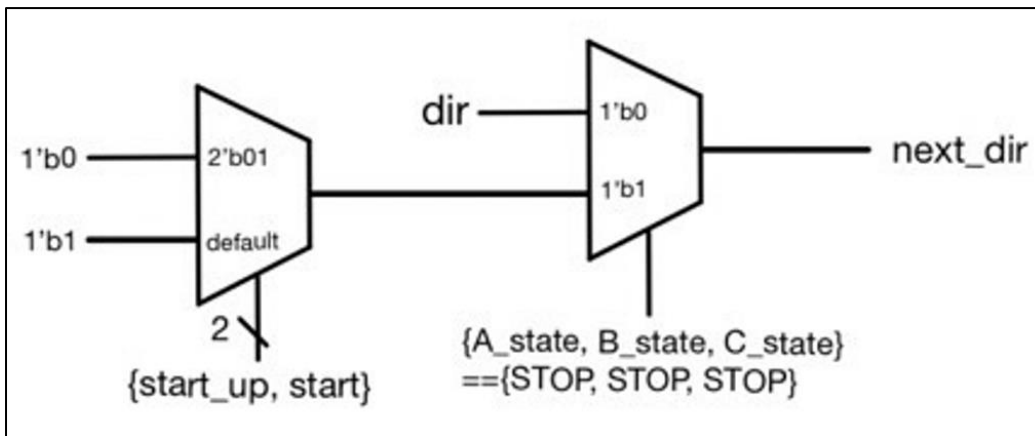


Figure 1.5

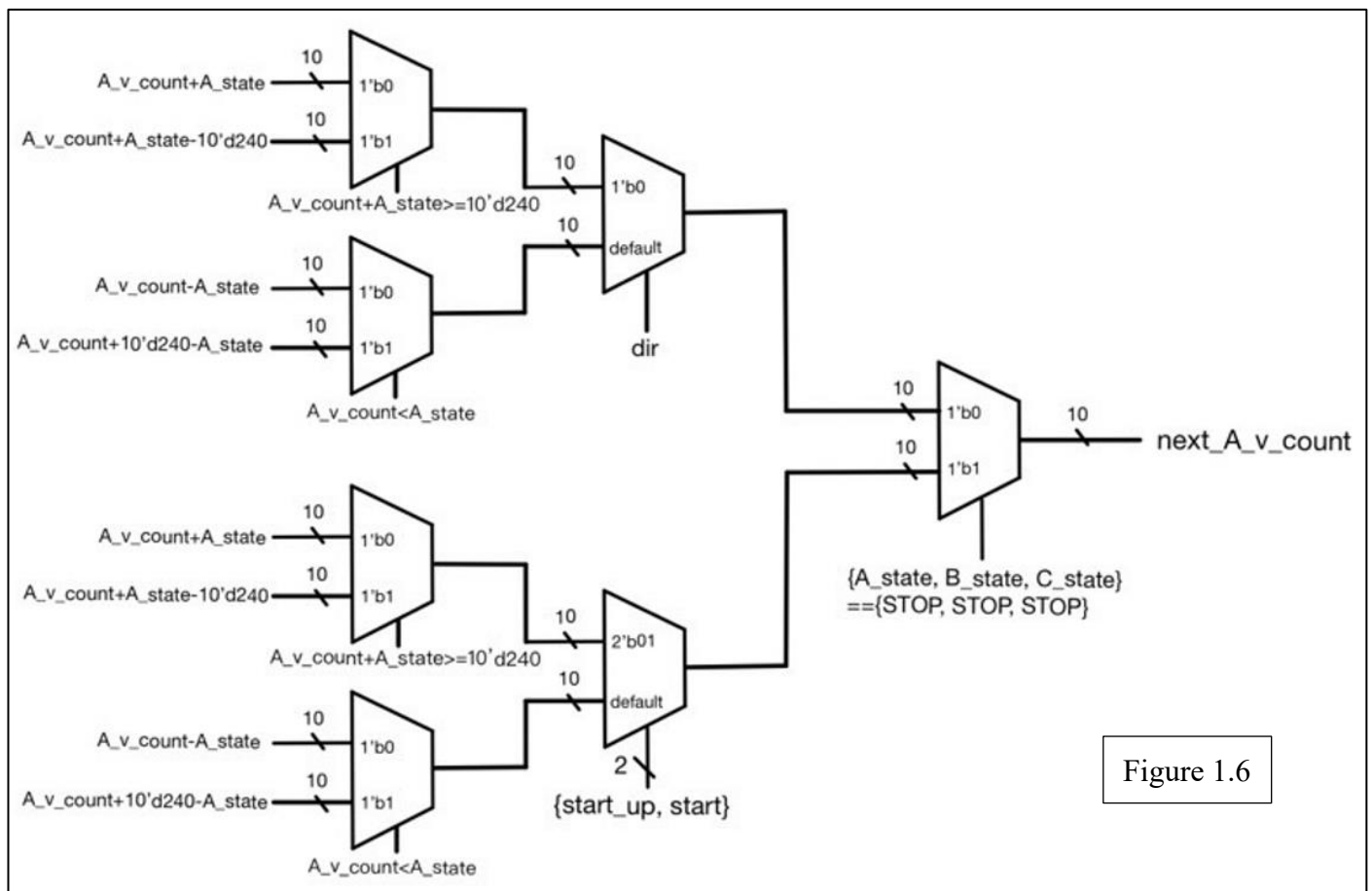


Figure 1.6

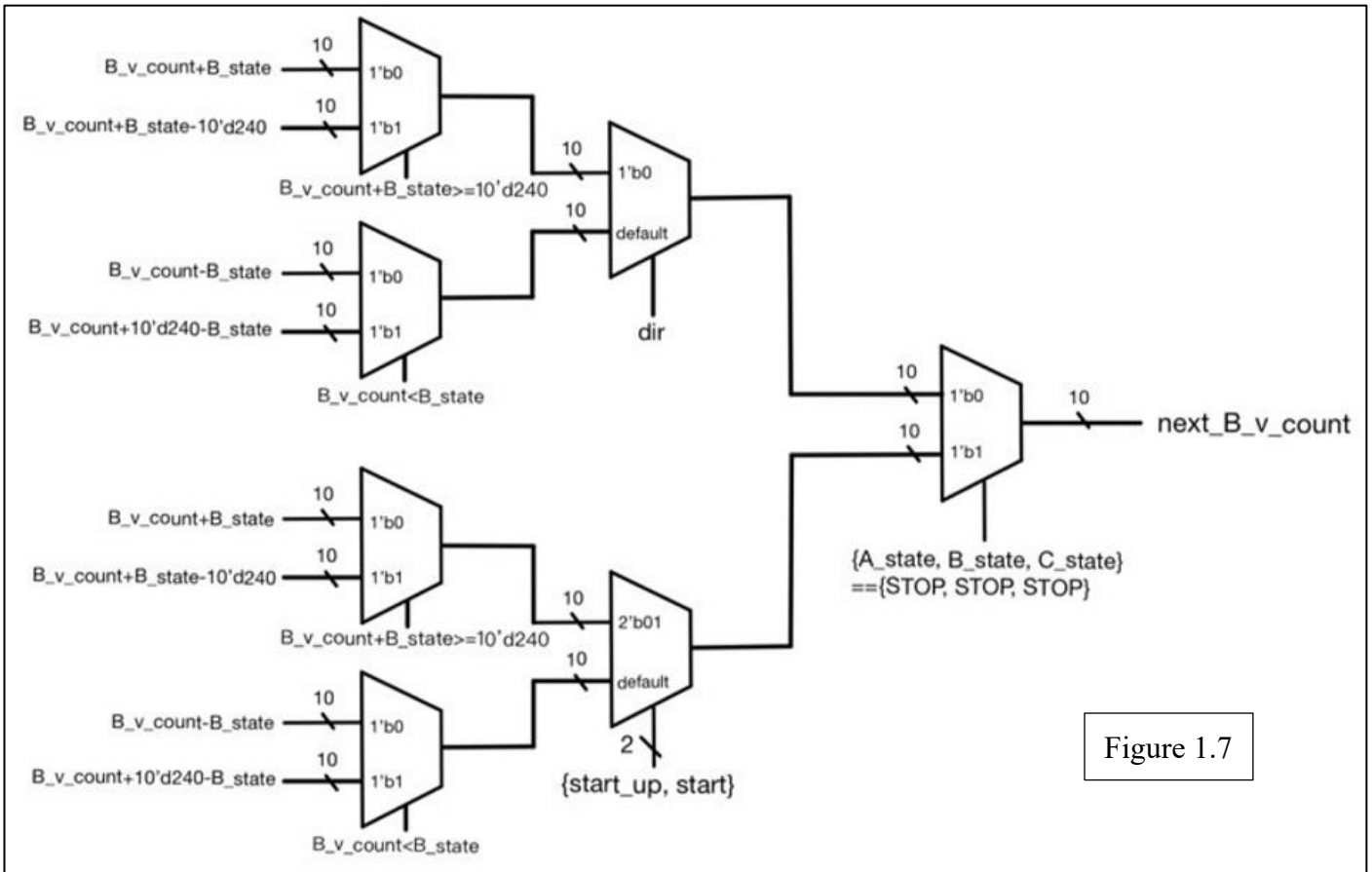


Figure 1.7

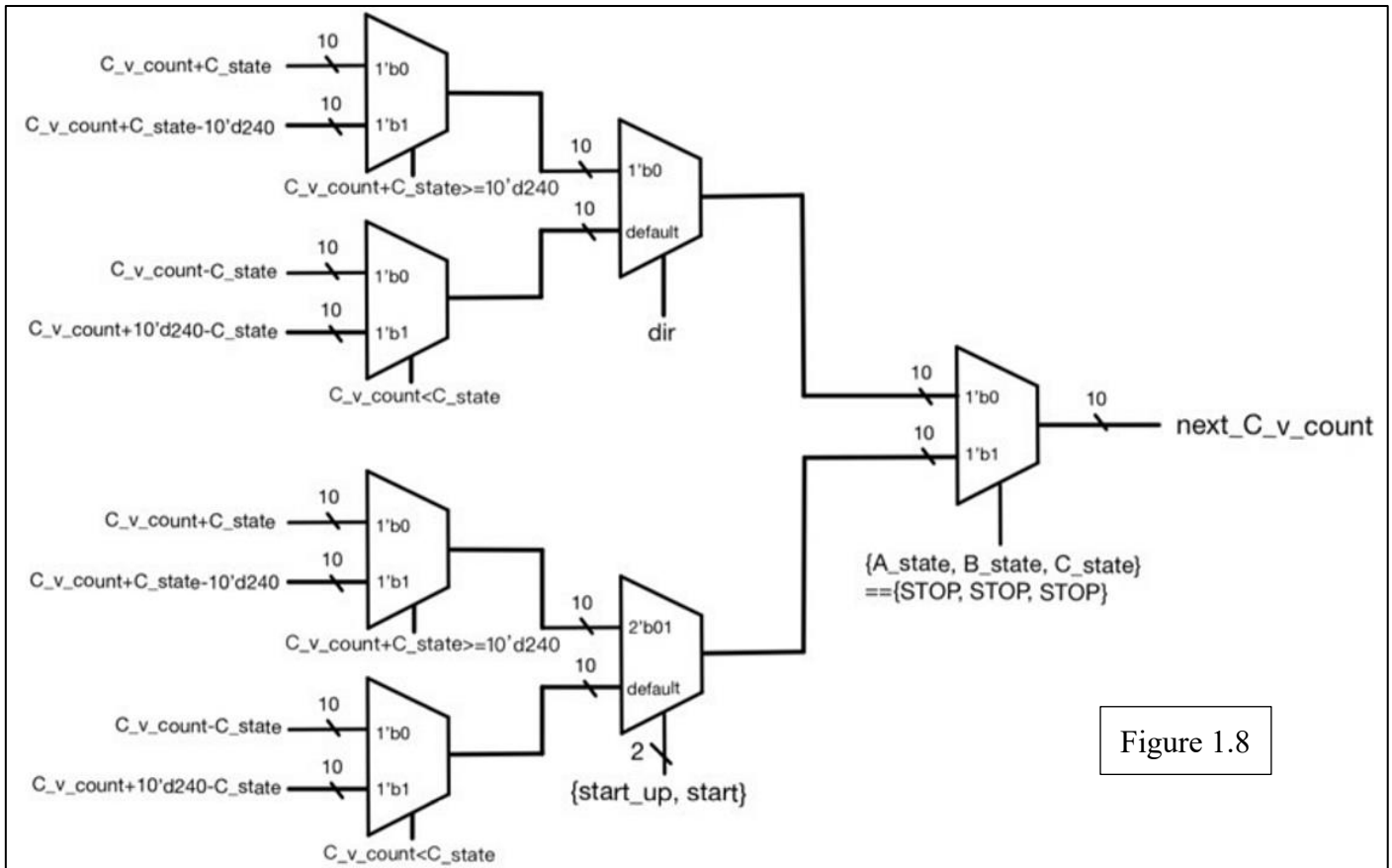


Figure 1.8

B. 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。

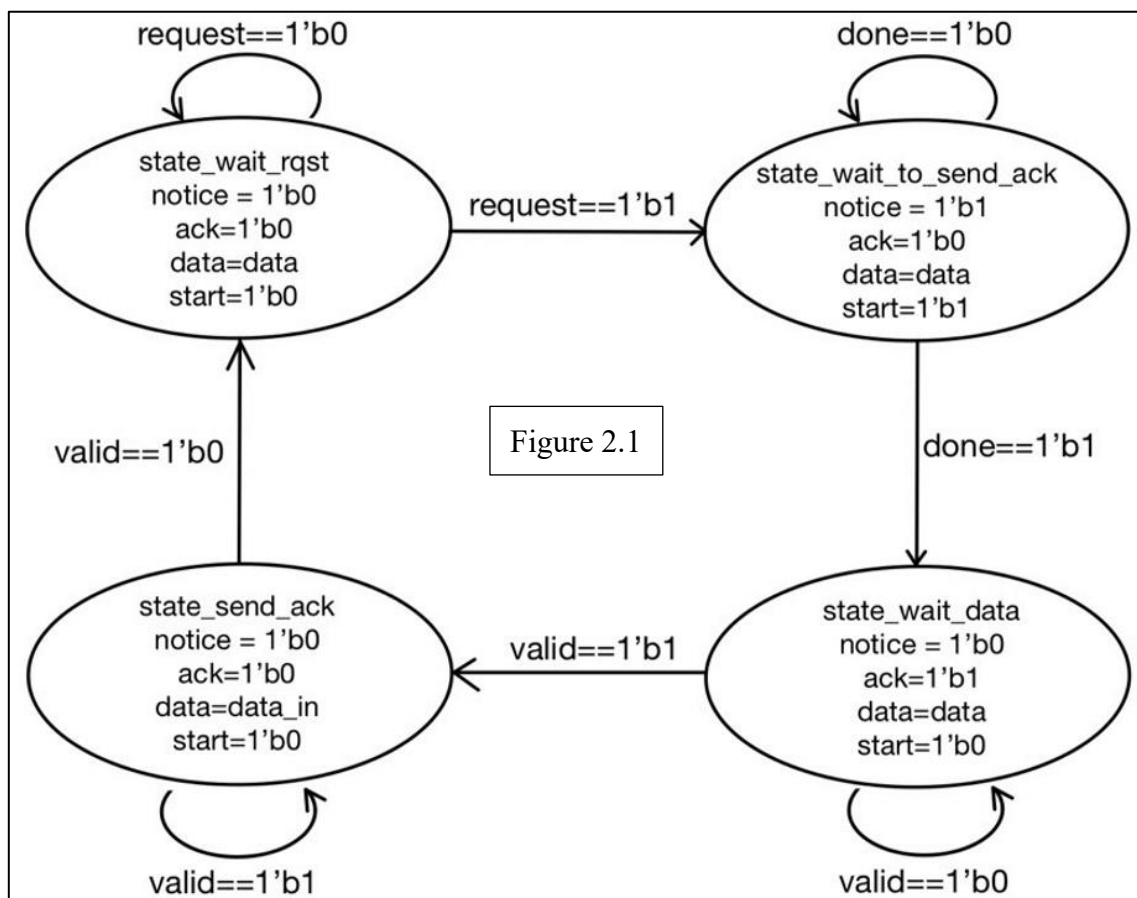
為了實現向上旋轉的功能，我們新增了一個叫做 start_up 的 input 來接受向上旋轉 button 的輸入，原本向下旋轉 button 的 input 就繼續沿用原本 start 這個名稱。因為加入了 start_up，所以對於 A、B、C 開始數數的條件我們改成了只要 start 或是 start_up 為 1 時，並且 counter 是 0 時（slot machine 目前在停止狀態），counter 才會開始數數 (Figure 1.2 ~ Figure 1.4)。

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。

2. FPGA Question: Chip2chip

A. Finite State Diagram



B. 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。

3. FPGA Question: Car

A. Finite State Diagram

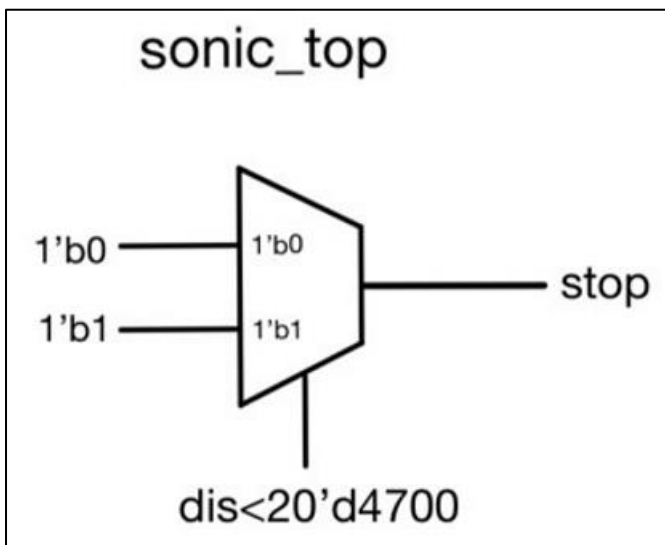


Figure 3.1

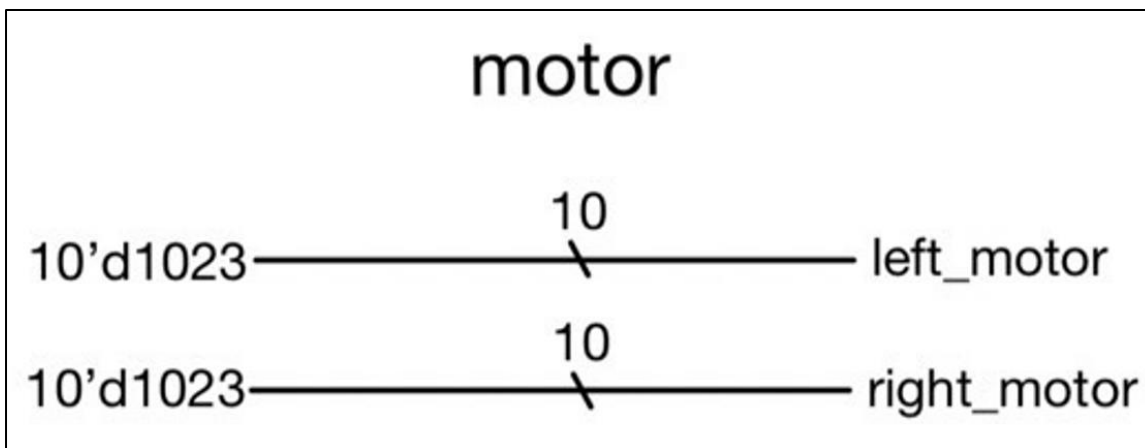
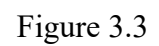


Figure 3.2



B. 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。

4. What We Have Learned

A. 李侑霖

這次 lab 我學到很多東西，除了對螢幕有更深入的了解之外，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，車子的部分也讓我了解其基本的原理，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。

B. 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%