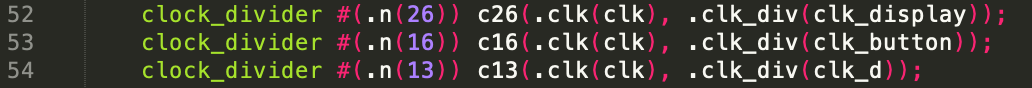
|  |
| --- |
| **EECS 2070 02 Digital Design Labs 2019**  **Lab 5** |
| **學號：104021215 姓名：熊磊** |

1. 實作過程

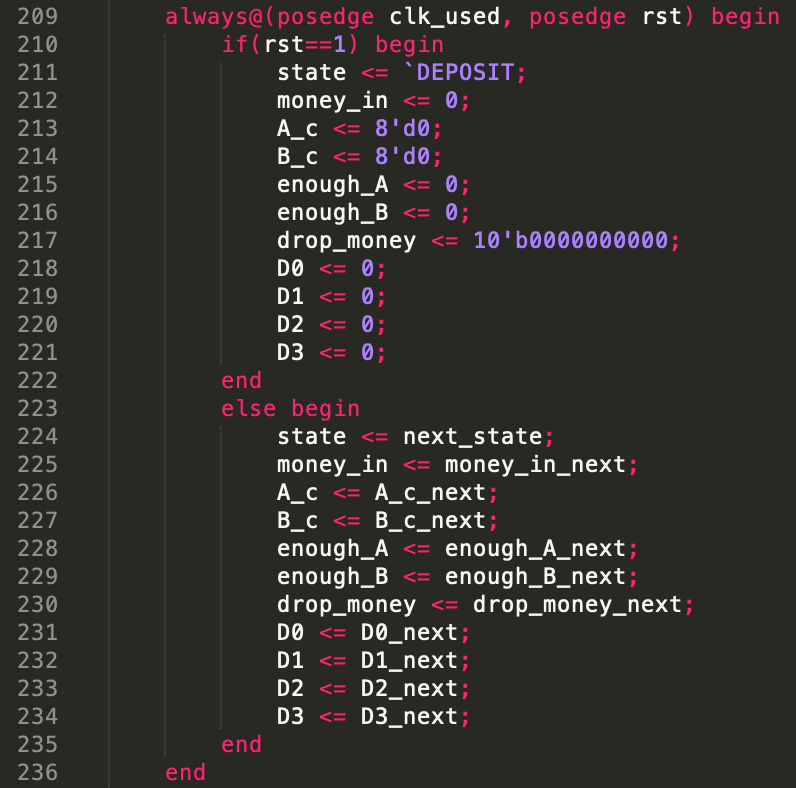
* 先利用clock\_divider做出需要的三種頻率，然後將input中的cancel, money\_10, money\_5, drink\_A, drink\_B做debounce及one pulse的處理。



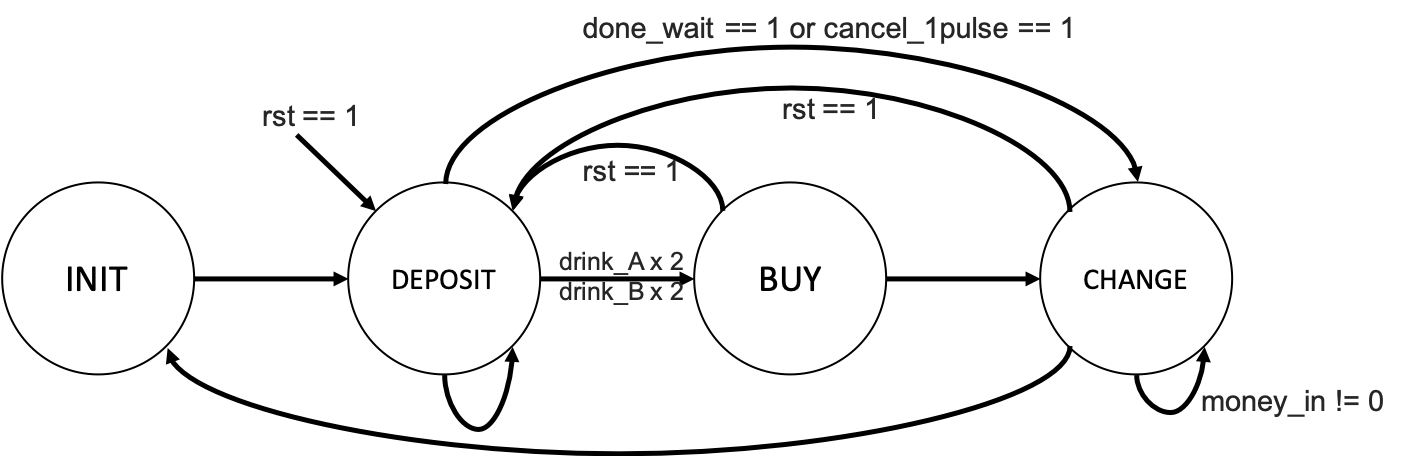
* 設一個clk\_used的wire，當state為DEPOSIT時，就跟著clk\_button，否則就跟著clk\_display。



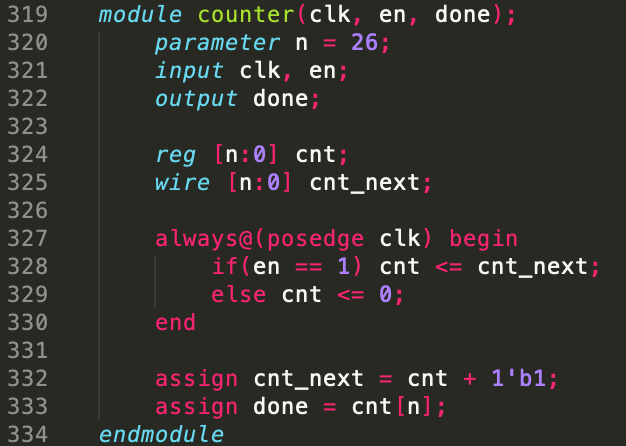
* 把posedege clk\_used及posedge rst當作一個always block被觸發的條件，在裡面會將state, 使用者投入的總金額, drink\_A的計數器, drink\_B的計數器, enough\_A, enough\_B, 飲料名稱reset, 餘額 reset或是update。



* 這次總共用到四個state，大致的FSM切換trigger 如圖所示



* 用一個@\*的always block來維護所有我需要用到的值，包含上述8個值，以下將針對各個state討論。
* 在INIT時，在進到這個state時為一次新的交易，應將上述8個值歸零並可以直接進入DEPOSIT。
* 在DEPOSIT時，如果使用者有選擇飲料，將對應的計數器設為1，另一個飲料的計數器設為0，並設定飲料的名稱。如果使用者投錢，增加money\_in。如果money\_in的金額足以買drink\_A 或 drink\_B，將enough\_A, enough\_B亮起。若使用者按了cancel，就將下一個state設為CHANGE。在這裡說明為什麼使用飲料的計數器，如此可以幫助我判斷使用者是否按兩次飲料的button。我將進入BUY的條件設為判斷投入的錢是否大於drink的金額：A數目\*20 + B數目\*25，如果使用者按了其中一個，另一個就會是0，那我要比較的金額就得到了，不用判斷他過去按的是A還是B。
* 在BUY時，將對應的飲料釋出後，飲料計數器歸零，算出要還的錢，顯示飲料的名稱，然後將下一個state設為CHANGE。
* 在CHANGE時，若有錢要還，則判斷要還的錢有沒有大於10元，若有，則將要還的錢減10元，以及將顯示的金額及Led燈更新，要還的錢為5元時，做類似的事。每次還錢都會判斷還有多少錢要還，如果已經還完，就回到INIT，否則留在CHANGE。
* 還有一個default的情形如下，state設為INIT，其他值設為0。
* Bouns的做法如下: 我每次在DEPOSIT有動作後，都將等待時間歸零，並開始計算5秒。做法是做一個counter，用226來除頻，並用en來作為計時器的開始按鈕。而en在DEPOSIT的state下，只要有任何button被按就會在該clock中設成0，而此時計時器會歸零。若沒有下個動作，下個clock，en又會被設成1，並開始counting。



1. 學到的東西與遇到的困難

這次作業讓我對finite state machine更熟悉了一些，要設計好什麼時候要改變state以及每個值在哪個時間點會改變，否則很難做出題目要的樣子，過程中有發生錯誤也很難知道錯誤在哪裡。本來沒有妥善規劃每一個值造成發生錯誤了，我也看不出來哪裡有錯，後來規畫好之後，直接重寫一遍，才比較容易找到問題在哪，也才順利寫出這次作業。

1. 想對老師或助教說的話

謝謝助教的幫忙☺