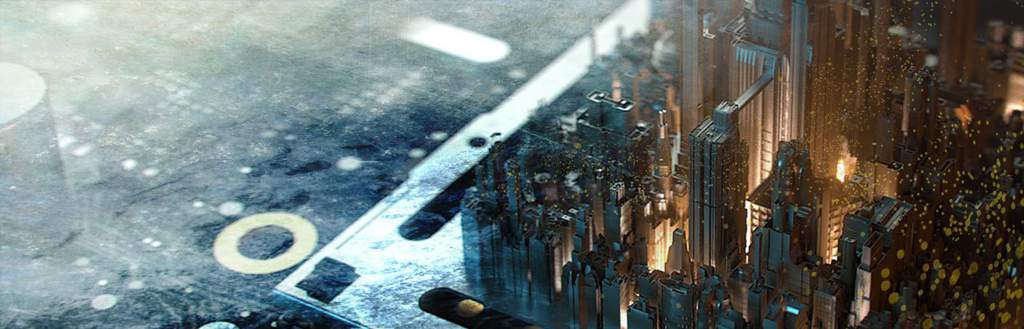
****

**Lab 4 report**

**2020/11/12**

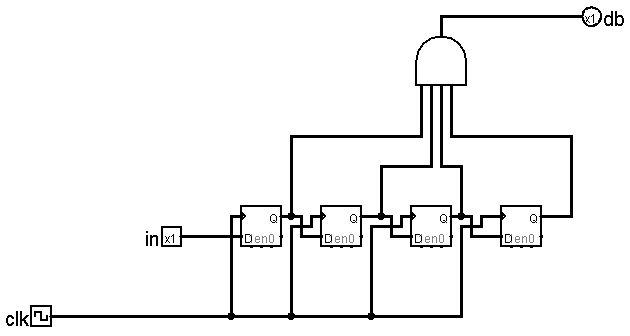
**108062125 高敦晉**

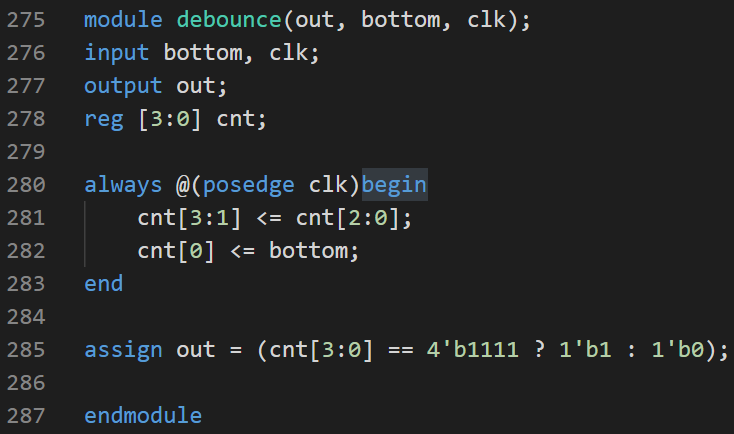
**108062229 陳皇佑**

**FPGA**

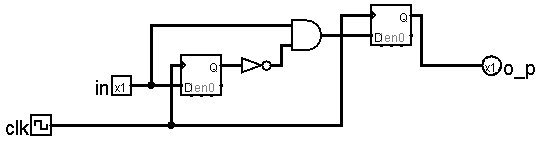
**設計構想：**

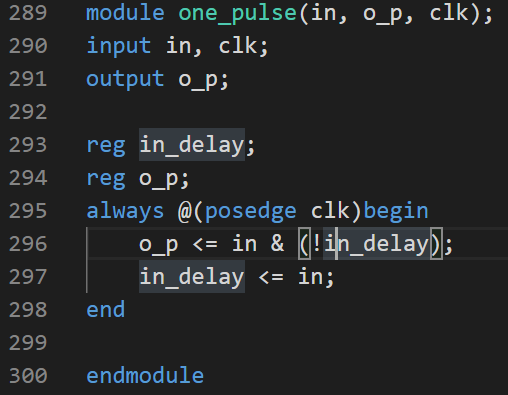
Onepluse、debounce以及for 7-segment顯示頻率的clock divider 基本都沿用上次的code

****

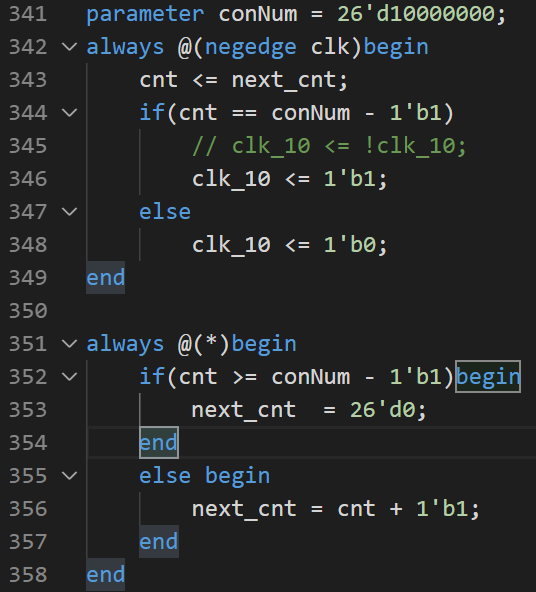
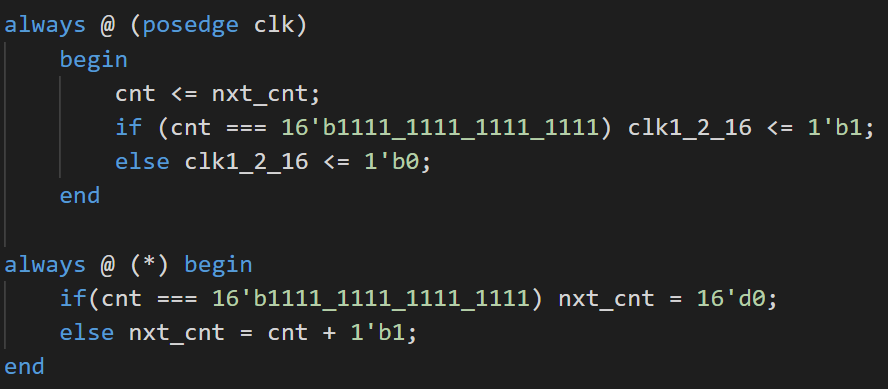


Debounce是讓input經過4個dff，確保要四個clk cycle連續為1。

****

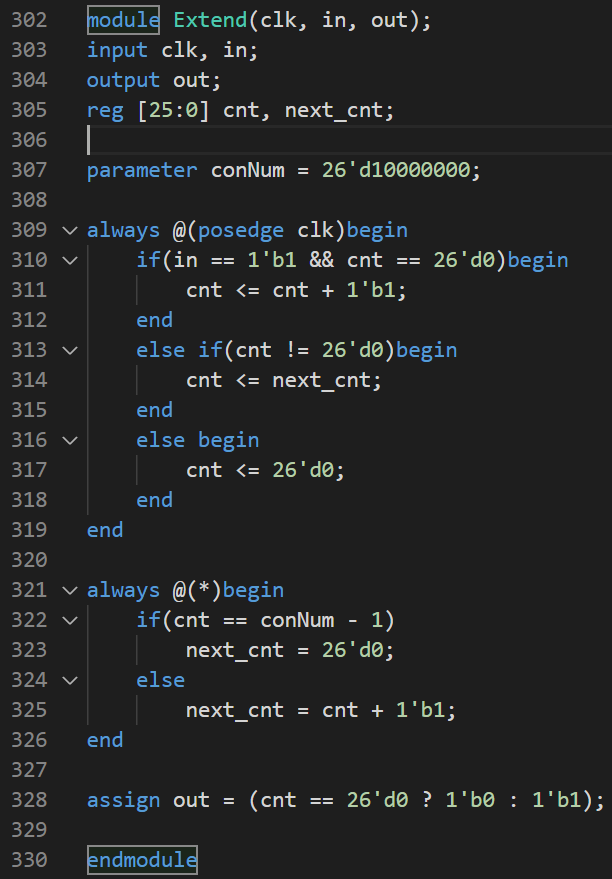


One\_pulse是in跟in\_delay的相反做and，結果用dff輸出，這樣確保訊號1只會維持一個clk cycle。

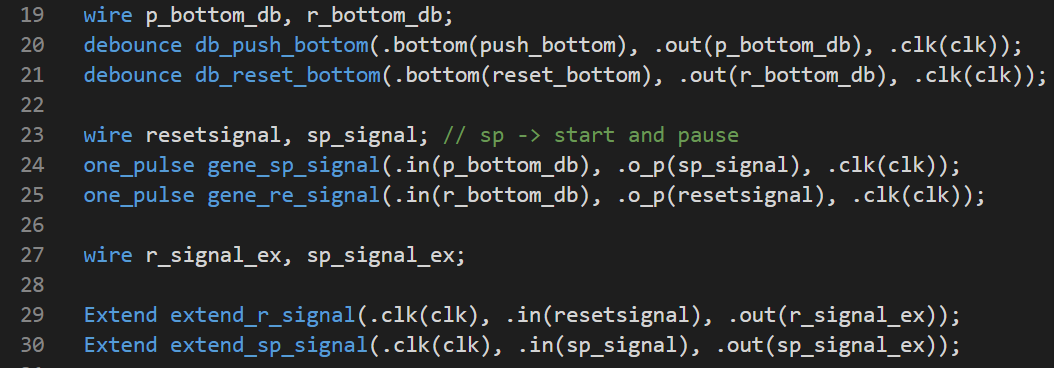
 

Clock divider，用一個dff做計數器，clksecdiv\_10是數到用來模擬現實中的秒用，注意一下這裡是用negedge trigger，後面會提到為何如此做。Clock divider 216是數到，之後counter歸0，output轉1。用來做顯示頻率。Clock divider 2是測試用的module就不列進來。

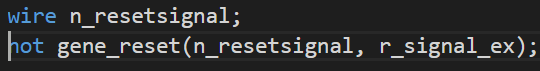
Extend module：

 用來延長one\_pluse的訊號直到能夠被0.1秒的clk吃到，解決的上次的FPGA按鈕timing要按在奇特時間點的問題。

按鈕處理單元：



對push\_bottom、reset\_bottom兩個按鈕做debounce和one\_pulse，debounce傳入clk，這樣只要一按db\_output會馬上跳一，one\_pulse傳入，為了讓one\_pulse長度達到0.1sec clk長度，再將signal接入 Extend module。



因為reset是等於0初始化，所以把處理完的reset取反。

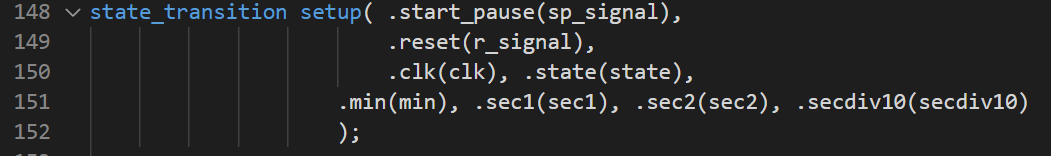


因為按鈕有點不靈敏，用來觀察按鈕到底由沒有按下去的訊號燈。



把參數放進counterup內。

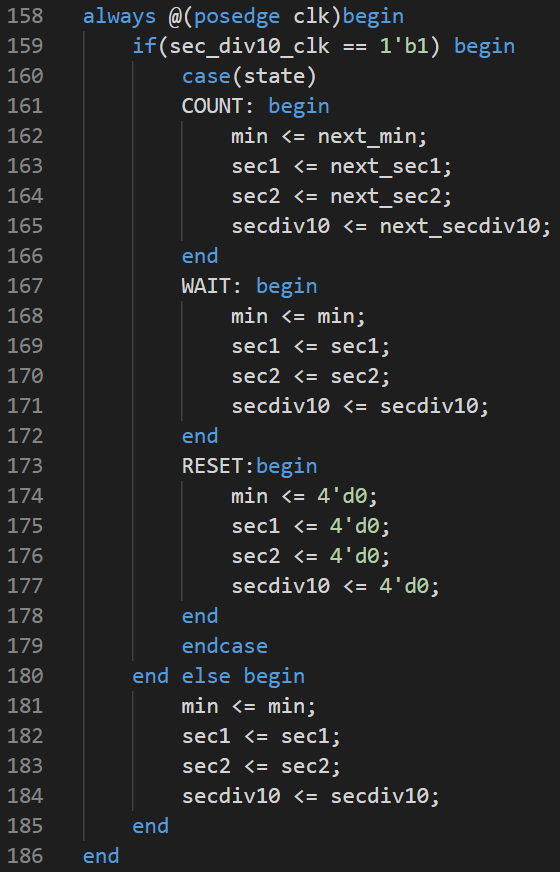
**Counterup Module：**



先將跟state有關的信號接近去state\_transition module裏頭。



這次為了遵守上課聽教授講的 ”所有的reg都應該用同一個clk trigger”，因此在有需要用到10Hz clk 的地方都用clksecdiv\_10這個negedge trigger的module，用一個10Hz升起的信號來跟posedge的clk做if。

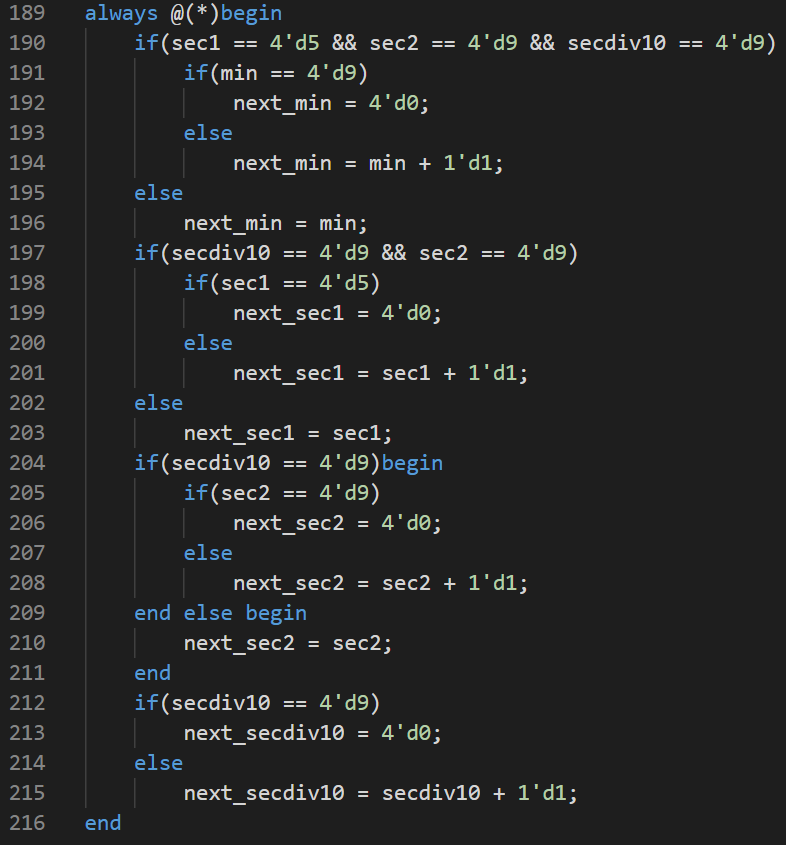
內部就看三個state ( COUNT, WAIT, RESET)

來依據要做什麼動作

如果是COUNT，那就往下數0.1秒。

WAIT，維持現在的值不動。

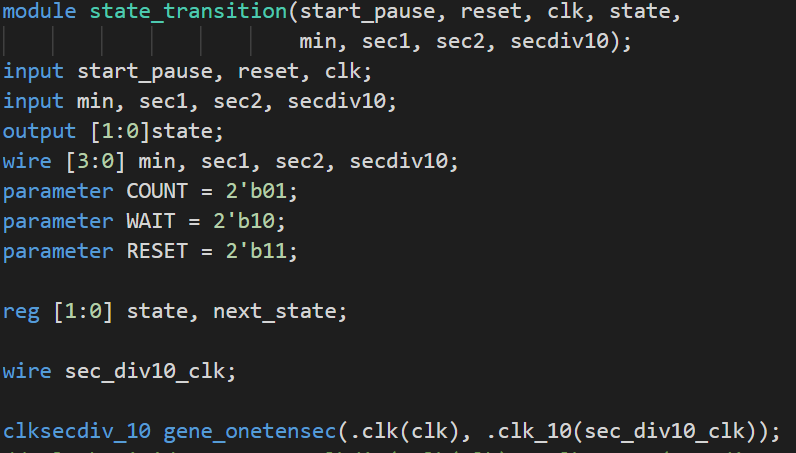
RESET重整回去0:00.0

Else這邊我是放不動。

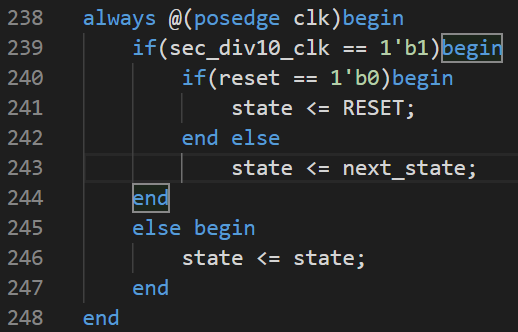
用來算下一個秒數的combinational電路。順帶一提，為了顯示方便我在second的2位數是用兩個4bit的reg去做的。

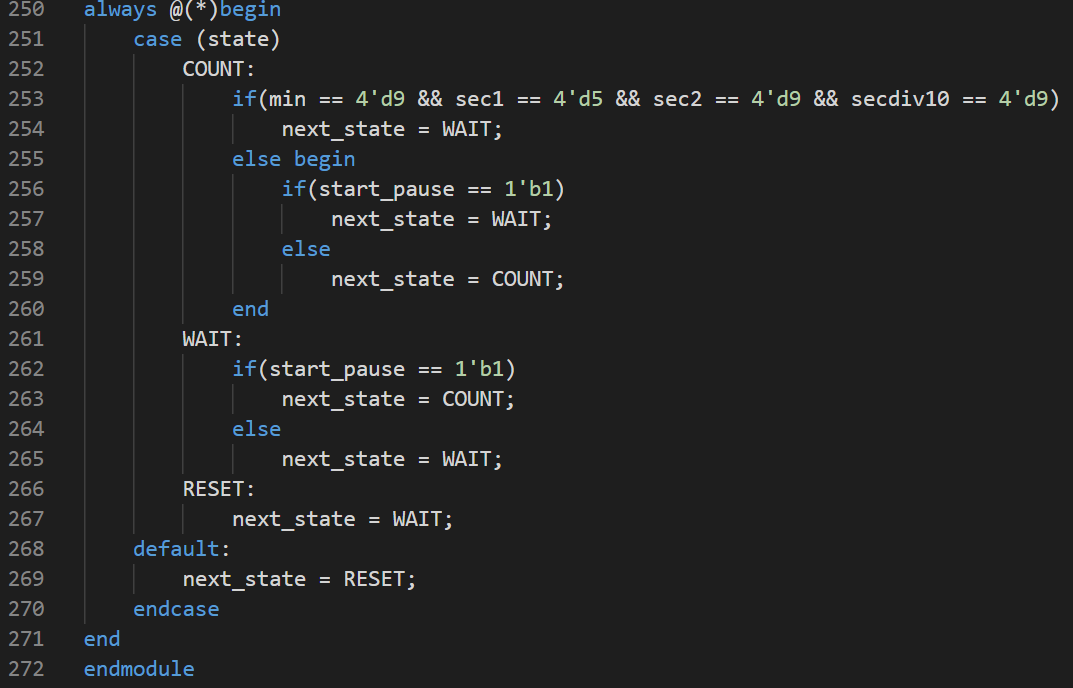
基本上就是預先判別下一秒會不會進位，決定next值。

**State Transition module：**



定義三個state的2bit值，不用00是因為我希望可以接出去信號燈觀察state的變換有沒有正確。下面一樣生一個10Hz的clk。

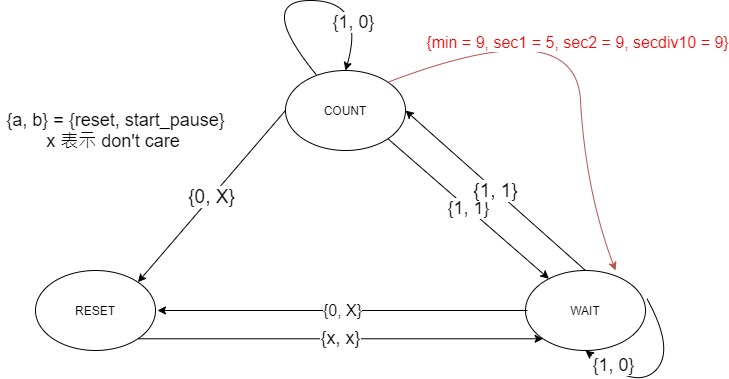
Sequential電路，就reset==1’b0的話直接跳到RESET state。不然就去next\_state。



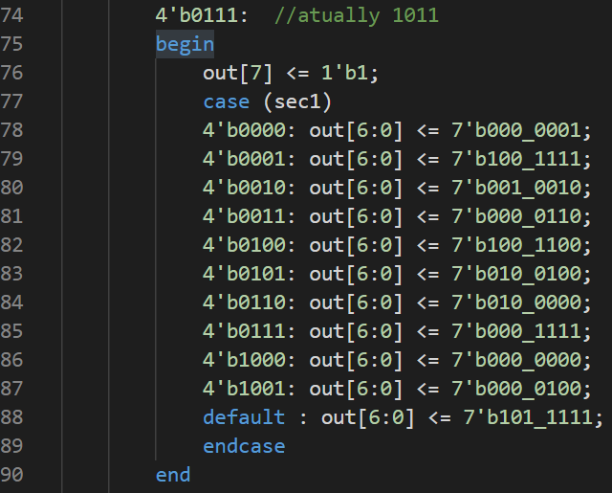
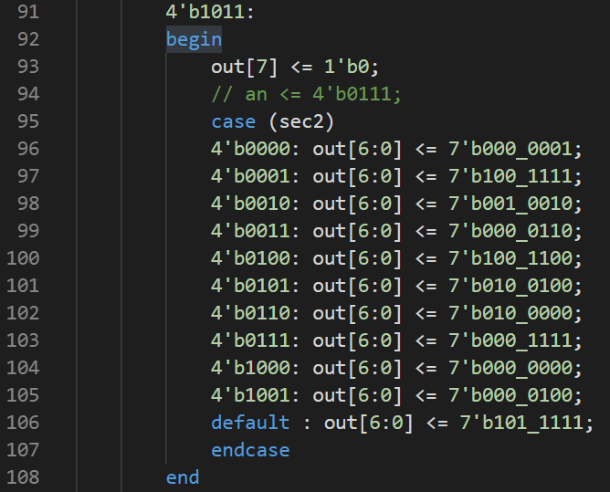
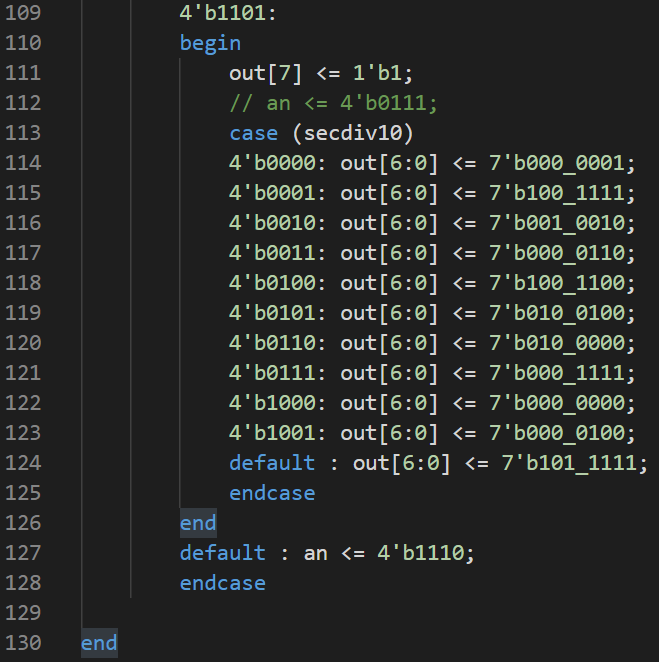
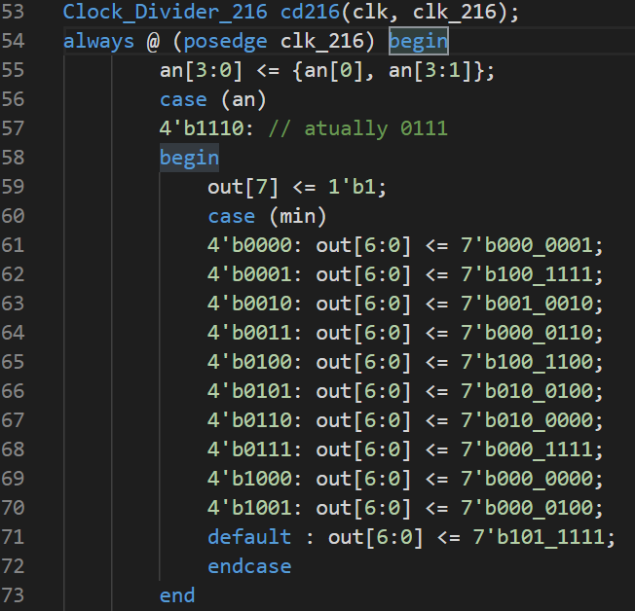
決定next\_state的combinational電路，根據現在state的值以及start\_pause這個按鈕input，決定下一個state是什麼。

意外的是，在COUNT state，是有多判斷現在的時間來決定，因為overflow之後要停在0:00.0的緣故，所以其實input看了4個東西。只是時間這個input在其他state被省略進else裏頭了。

下方是state transition diagram。



基本上如果寫的{a,b}的部分，就是將{min, sec1, sec2, secdiv10}當成don’t care。



用2的16次方clk，讓肉眼觀察不到各個digit跳動。

若reset，an從3跑到0，分別在an = 0跑到 3, 2, 1, 0時，輸出各個min, sec1, sec2, secdiv10的num and direction output，小差別只在於an = 1時。要將out[7]改成0，for 小數點的顯示。

**開發過程中的問題/學習：**

1. 本來一開始規畫的時候是將sec做成一個6bit的計數器，差點沒把自己搞死，後來決定改成分成兩個4bit位數，雖然多用幾個register，卻讓code變的整潔易維護許多。
2. 下次可能要改進的地方是，將7-segment寫成獨立的module，不然每次都手動土炮輸出，非常花時間。這份的7-segment輸出，大體上也還是沿用上次的FPGA code，clk還是用 trigger而不是原生clk加上if判別，算是一點小缺點。
3. 本來一開始寫的時候，不懂為何要用FSM，這明明就一個module，判斷input來改數字就可以解決的事情。但是越寫越體會到FSM的好處，把state分隔開來，要臨時更改state或是更改state的行為、next\_state的方向時，能夠快速簡潔的更改少部分的code，每個part各自獨立，減少出錯的機會。第一次體會到FSM設計的好處。