|  |
| --- |
| **EECS 2070 02 Digital Design Labs 2019**  **Lab 6** |
| **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rst, cheat, start做debounce及one pulse的處理。

A close up of text on a white background

Description automatically generated

* 設一個clk\_used的wire，當state為SUCCESS時，就跟著clk\_led，否則就跟著Basys3的clk跑。



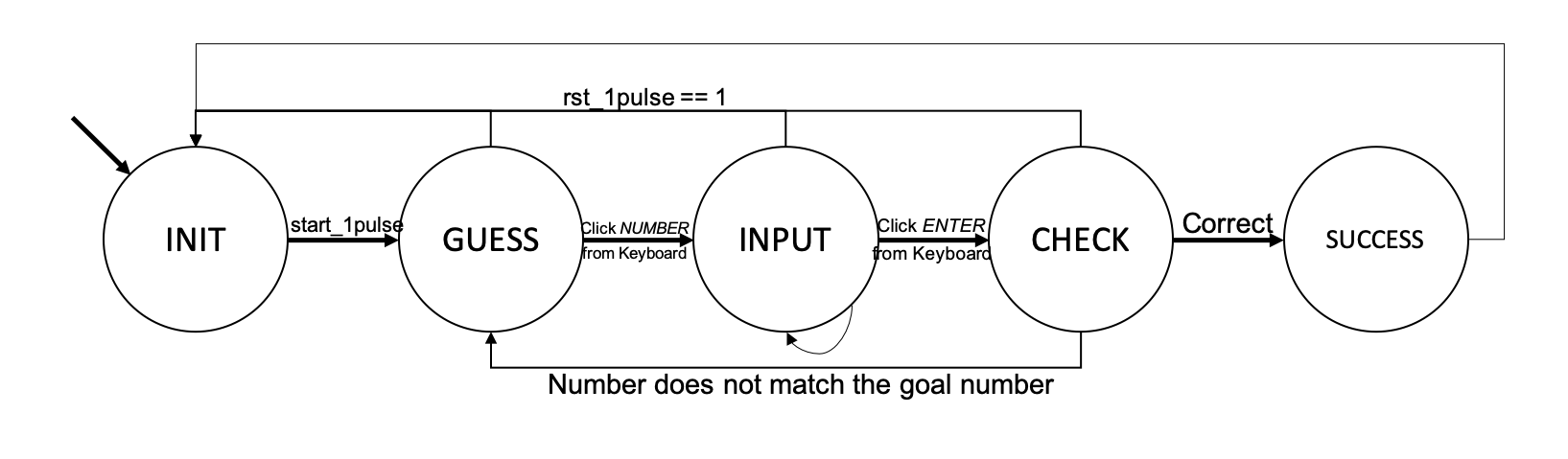
* 把posedege clk\_used及posedge rst\_1pulse當作一個always block被觸發的條件，在裡面會將state, min1, min0, max1, max0, goal1, goal0, D0, D1, D2, D3這些signal做 reset或是update。

其中{min1,min0}表示目前小於goal猜測的最大整數；{max1,max0}表示目前大於goal猜測的最小整數。

A close up of text on a black background

Description automatically generated

* 這次總共用到五個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時把min設為00，max設為99，並把7-segment Display的直設成 **- - - -**。此時會利用LFSR的module來生成亂數，在INIT 時如果按下start\_1pulse，會把random的值儲存到next\_goal，並會將state切換到**GUESS**。

A close up of a clock

Description automatically generated

* 在**GUESS**時，顯示器會顯示目前已猜測到的數字範圍，左邊兩個Digits是min，右邊兩個Digits是Max。開始進行猜數字時，意即當收到Keyboard數字鍵的Signal時，就會把state切到**INPUT**。
* 在**INPUT**時，只要使用者鍵入任何數字，個位數都會替換掉十位數，而鍵入的數字會放在個位數。當使用者按下Enter時，即會將state切入**CHECK**。
* 在**CHECK**時，如果使用者鍵入的數字只有一個位數，則不更改min和max值，直接重回**GUESS**的state。而其他情況會跟goal做比較，如果跟goal的數字相同則進入**SUCCESS**的state；其餘，如果大於goal，小於max，則把next\_max設為鍵入的數字。如果小於goal，大於min，則把next\_min設為鍵入的數字。其他情況則不更新next\_min、next\_max，直接回到**GUESS**的state。

A screen shot of a computer

Description automatically generated

* 在**SUCCESS**時，把左邊兩個digits和右邊兩個digits都顯示goal的numbers。

1. 學到的東西與遇到的困難

這次作業做起來都蠻順利的，但是後面在測試的時候出現很奇怪的bug，就是如果輸入的值不是goal，而會更新min的值時，漸入的值會蓋在max的值上面，造成max值顯示的很奇怪。後來發現是clk設定的問題，把clk換成一樣的就解決了。

1. 想對老師或助教說的話

謝謝老師與助教!

在demo時，助教幫我找出上面那個bug可能發生的原因，所以才能解決並順利完成demo。還在思考期末有沒有有趣的idea可以做出來，想做娃娃機和拉霸這些有趣的project，但是做一樣的沒什麼意思，希望有點創新，不知道助教有沒有有趣的想法。