林承毅 102062501

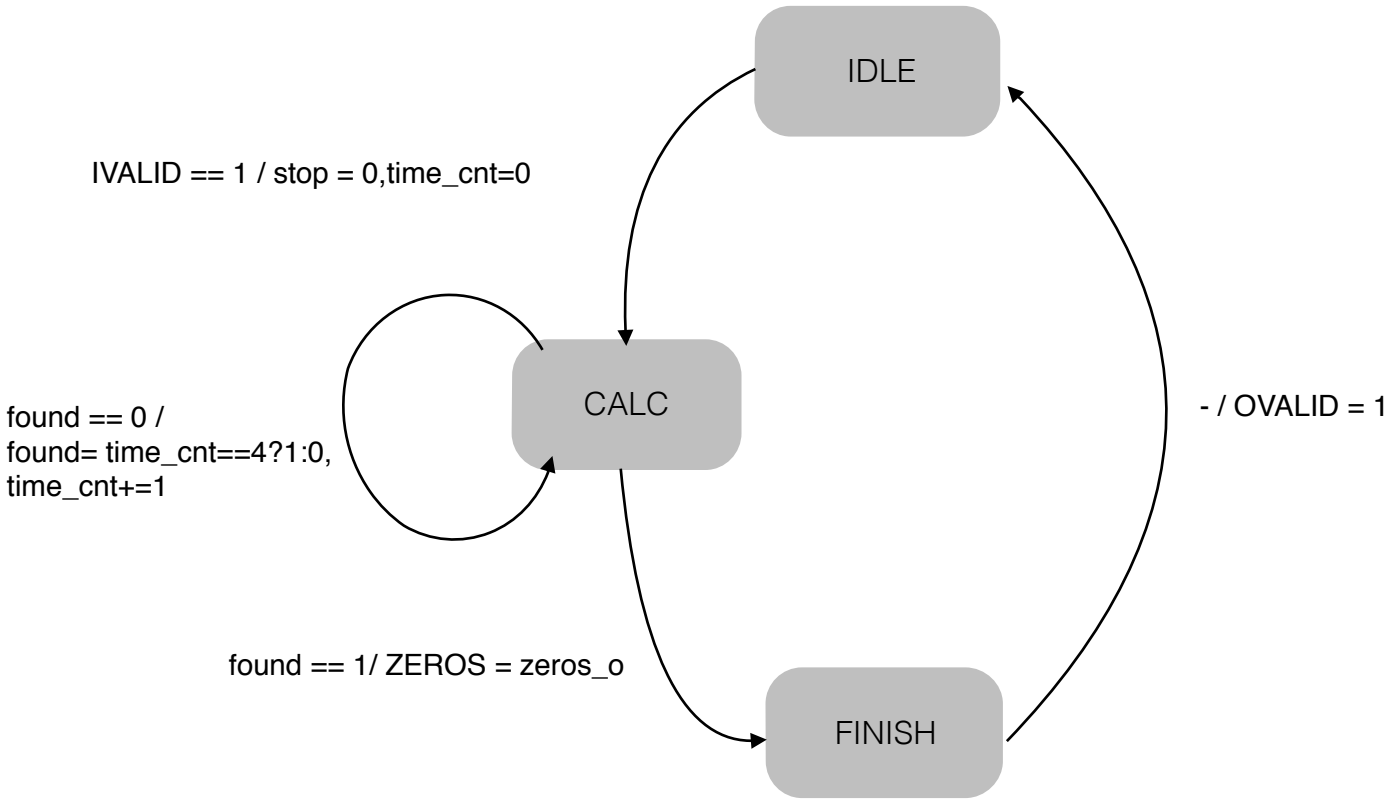
超大型積體電路設計實習

2014年10月13日

Lab2 Report

一、設計架構

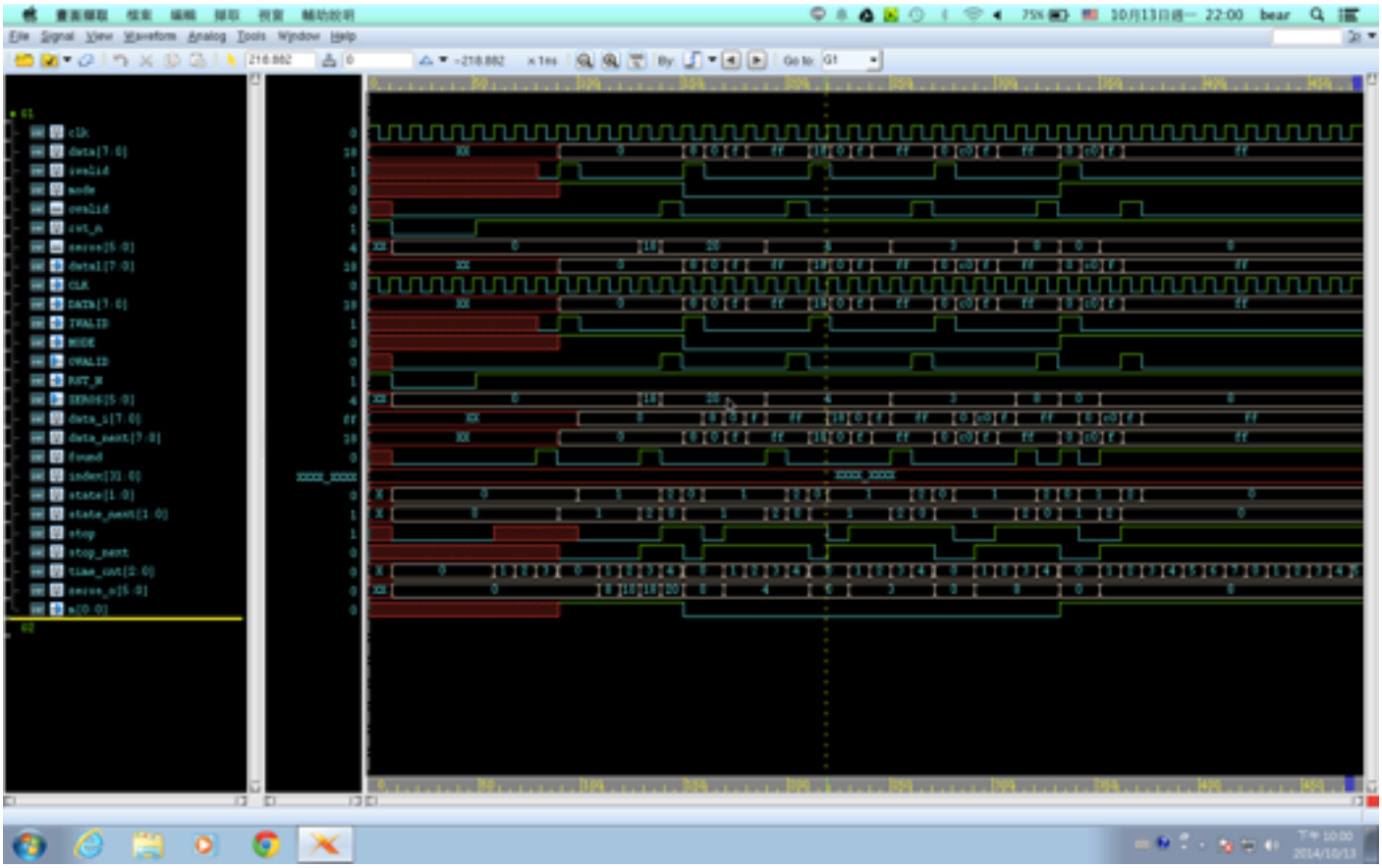
這次的實驗設計採用FSM來完成。簡單的狀態機圖例如下（Normal mode）：



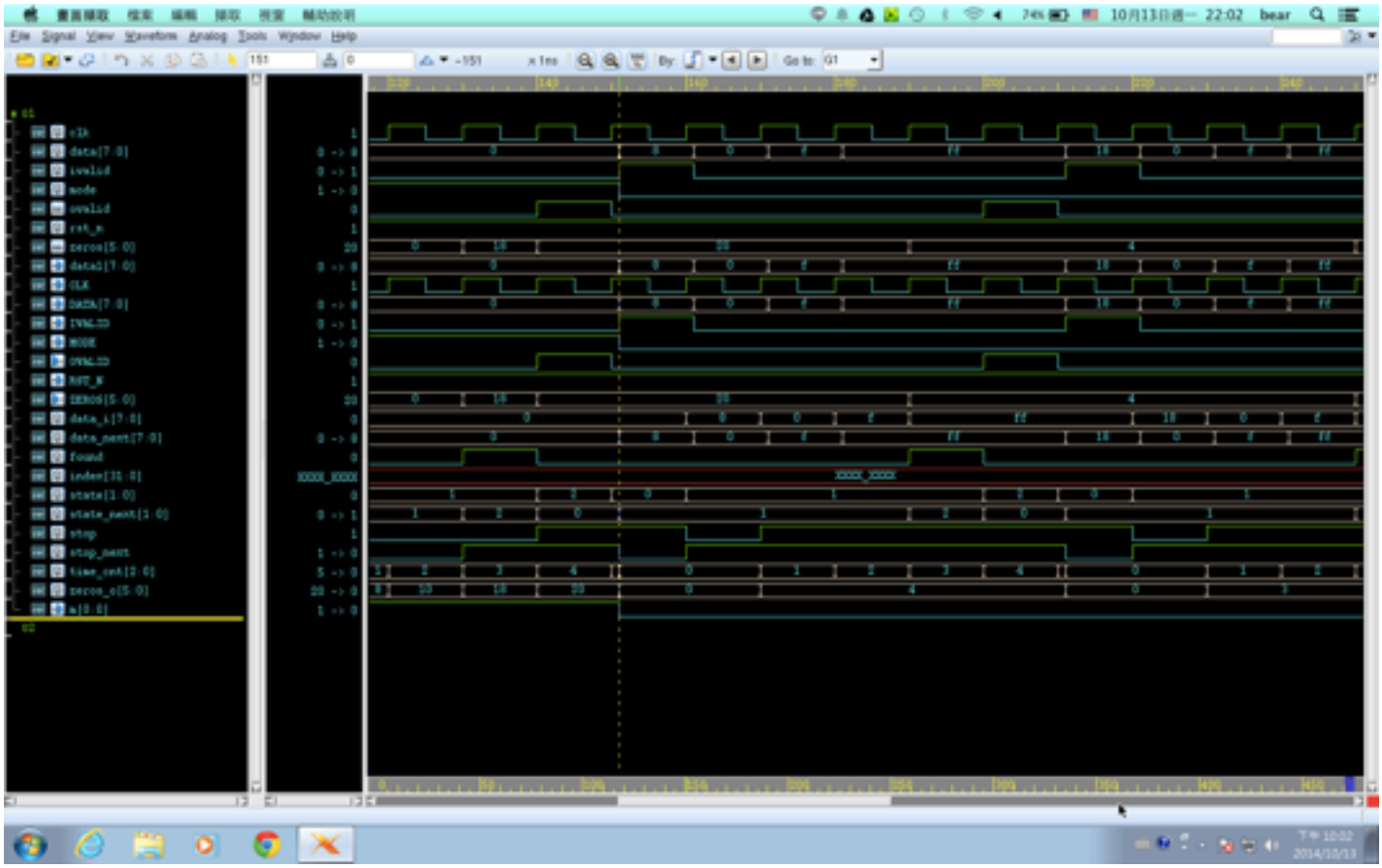
電路從idle狀態開始，直到 input 訊號 IVALID 被拉起來。此時 DATA 被存到 FF：data_i，以供下一個 state 來計算 leading zero。若電路的模式是設在normal mode，則假定接下來的3個 clock cycle會把 DATA 的另外24bits送過來。在 CALC state，每一個 clock cycle 都會把 time_cnt 增加一，來計算是否 DATA是否結束。每一個 clock cycle，會有 block 負責將 leading zero 的數目更新至 zeros_o。若發現這個 cycle 的 data_i 已經不再有連續的leading zero，則將 stop 訊號升起。若 stop 訊號升起，則下一個 clock cycle 便不會更新 zeros_o 的值。最後當 time_cnt 為3（DATA input 結束），則將 found 拉

起，並把 zeros_o 設給 output ZEROS 後移至下一個 state。在 Finish sate，會把 OVALID 拉起，通知結果出來。上述流程在 turbo mode 大至相同，唯一的區別是 found 訊號會提早在 stop 被拉起是一併設為1，並且提早進入 Finish state。

二、實驗結果



以上是電路測試的結果，顯示的範圍包含了5組input。我們放大其中一組 input （DATA = 08000FFF, mode = 0）來驗證程式的正確性。在下頁的圖例中，黃線的部份是 IVALID 被拉起的時間點。可以看到 stop_next 被設為 0，zeros_o 和 time_cnt 被清空。在下一個clock cycle，data_i 接到了上一個 clock cycle 收到的DATA，並正確地更新了 zeros_o。由於第一筆DATA為08，表示接下來的DATA不能再更新 leading zero。因此 stop_next 被拉起，並且在下一個 clock cycle 檢查到 stop 為1，代表停止更新 zeros_o。最後當 time_cnt == 3，可以結束計算時，found 訊號被拉起，進入 FINISH state 結束整個計算。最後 ZEROS == 4，顯示電路運作正確。



三、額外功能

這次的額外功能是實作不連續輸入的 DATA，亦即可以不用在連續的4個clock cycle把資料輸入完全。實作的方法是只在IVALID為1的時候輸入 DATA。並嚴格定義 IVALID 和 DATA 輸入的時間需和 clock 對齊。如此一來便能透過FSM來計算輸入的次序，達成不連續輸入的目的。

四、結論

這次的實驗最大的挑戰是，熟悉 zero-delay simulation。由於對於 zero-delay 還未完全習慣，因此剛開始的時候有許多訊號拉起的時間點不合乎預期。但在幾次的修改之後，便能校正時間點的問題，讓電路正確的運作。除此之外，在最後的修改之中，發現許多最近新教的 bad design pattern。在最後上繳前給予了修正，算是記取了不少教訓。希望能在下一個實驗中減少這一類型的錯誤。