數位邏輯設計-HW7 鄭凱文 104062223

HW7-1

Moore machine與Mealy machine的差異：

以此題為例，Moore machine的output，rd & ds都是在只看state決定，因此執行是在該state之下；Mealy machine的output則依state & input決定，因此執行是在input輸入時，如果符合條件就直接賦予值。

所以若要使兩者表現出的答案相同，則要將Mealy machine的output變化延後一個cycle。

→rd = rd\_next ; ds = ds\_next

問題1: Testbench的pattern是否足夠？

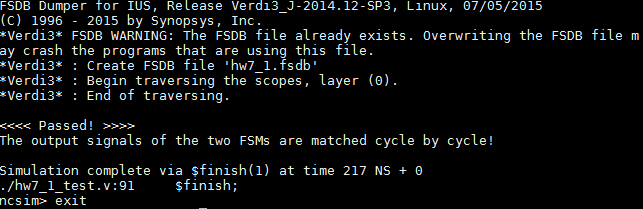
已經做到在state = DLY時，ws=0 or 1的差別判斷，應該足夠。

問題2: error和error\_count的用意？

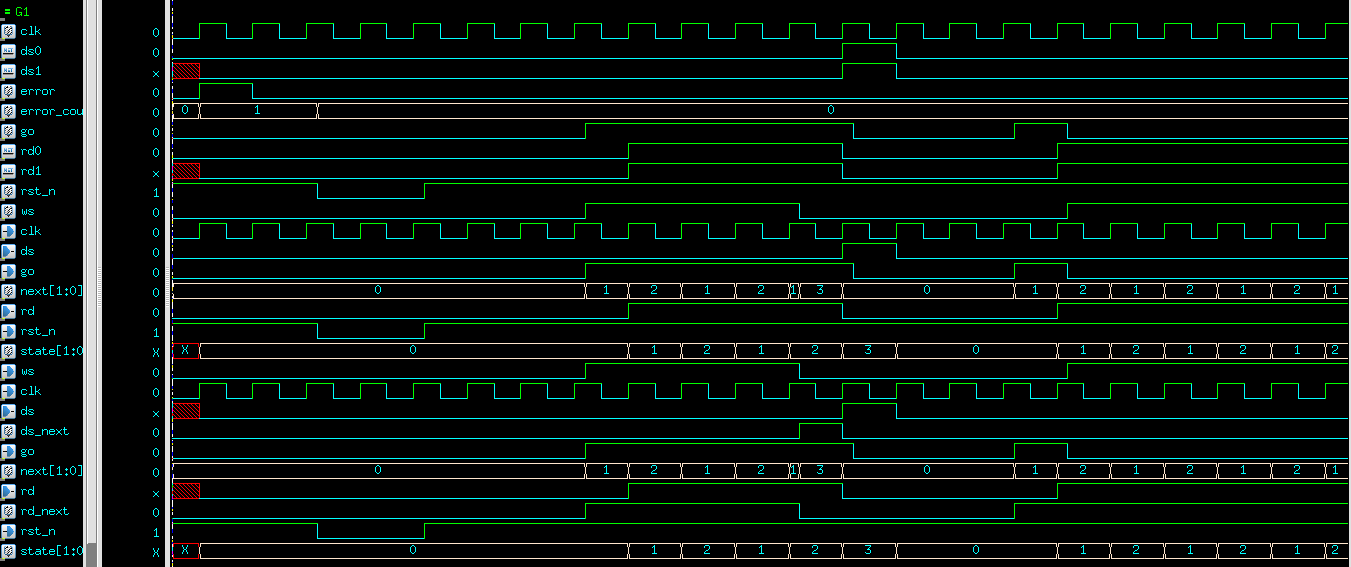
error代表執行過程中，Moore與Mealy的結果有沒有不同，若有，error = 1。

error\_count用來記錄執行過程中，出錯(即結果不同的情形)出現多少次。

Simulation:



波形圖：



HW7-2

GCD運行規則：

當讀到START = 1，開始執行運算。

藉由ERROR = error\_next，延後ERROR一個cycle更新。

若found = 1，在下一個cycle中DONE = 1，此pattern結束。

一旦ERROR = 1，下一個cycle中DONE也 = 1，此pattern亦結束。

每一個cycle執行state = state\_next, ERROR = error\_next, reg\_a/b = next\_a/b,

Y = data\_a，由DFF來控制。

計算過程利用”輾轉相減法”，每一個cycle執行一次運算。

關於Y的問題：

根據gcd\_golden.fsdb，範例的寫法是每一個cycle中均將Y = data\_a，最後得到GCD(A,B)。但依據筆記的Timing Diagram，Y的值應該要只有在found = 1，即得到最後的最大公因數，才將Y = data\_a(即GCD(A,B))。

為尊重範例，採用第一種寫法。

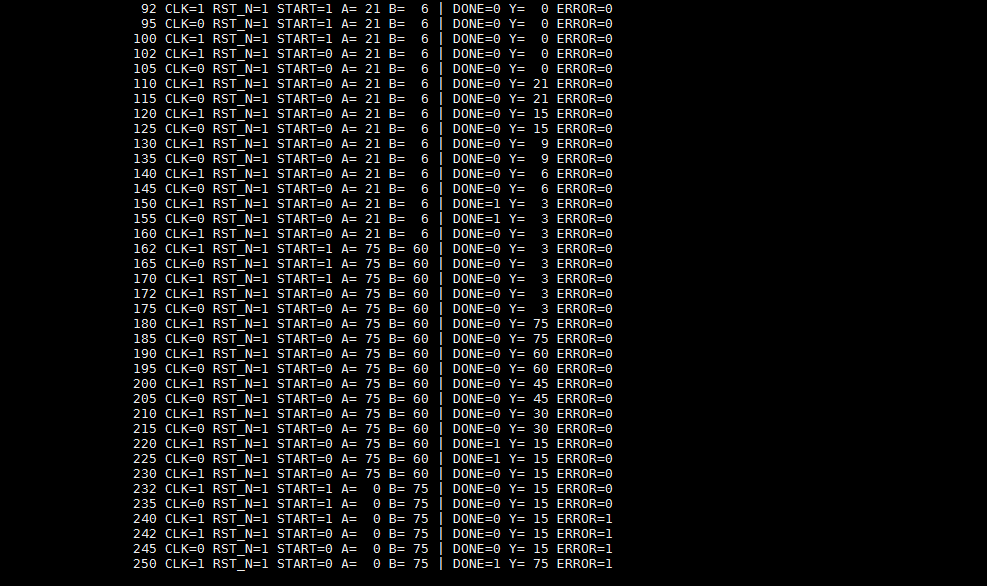
過程遇到問題：

一開始以為DFF的trigger條件可以用CLK或RST\_N以外的值，因此被筆記上的found騙到，讓程式無法正確執行。另外在同一個always block內前後更變過同個變數，造成迴圈無法順利停止。以上兩問題都在修正後解決。

Pattern的增加：

補齊A!=0 & B=0 (10,0)，以及A<B的情形 (14,21)，用來測試一開始若A<B會不會出錯。

Simulation(修改後的Testbench):



波形圖：

