Lab 5: Finite State Machines

Group 3 : 109060013張芯瑜 109062328 吳邦寧

# Design Explanation

## Sliding Window sequence detector

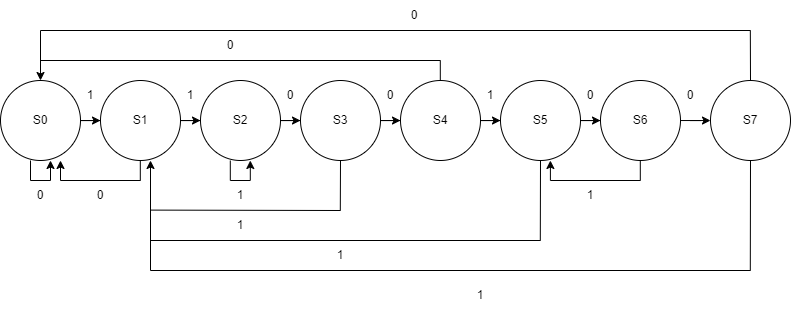
1. 設計說明：

利用finite state machine實作，偵測輸入進來的數串是否符合題目需求。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

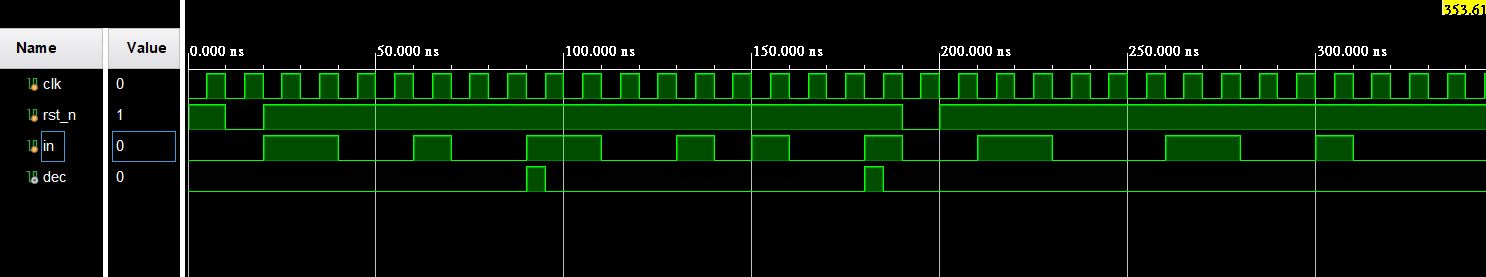
當state達到S7切in=1，dec被拉起，並在下一個posedge clk時歸零。

1. State Diagram：



1. 驗證：

嘗試所有可能的狀況，跑過所有的state並能夠得到正確的dec訊號。



## Traffic light controller

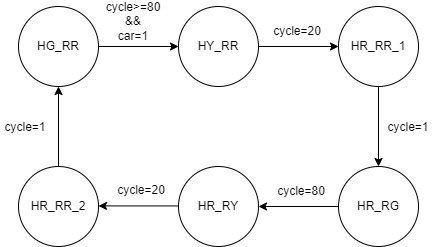
1. 設計說明：

按照題目要求設計，並分成下面六個state。

* HG\_RR：Highway綠燈/Local Road紅燈。
* HY\_RR：Highway黃燈/Local Road紅燈。
* HR\_RR\_1：Highway紅燈/Local Road紅燈第一次。
* HR\_RG：Highway紅燈/Local Road綠燈。
* HR\_RY：Highway紅燈/Local Road黃燈。
* HR\_RR\_2：Highway紅燈/Local Road紅燈第二次。

利用每一clk加一的方式計算cycle數， 當cycle數符合跳到下一state的條件時則將next\_state設為下一個對應的state。

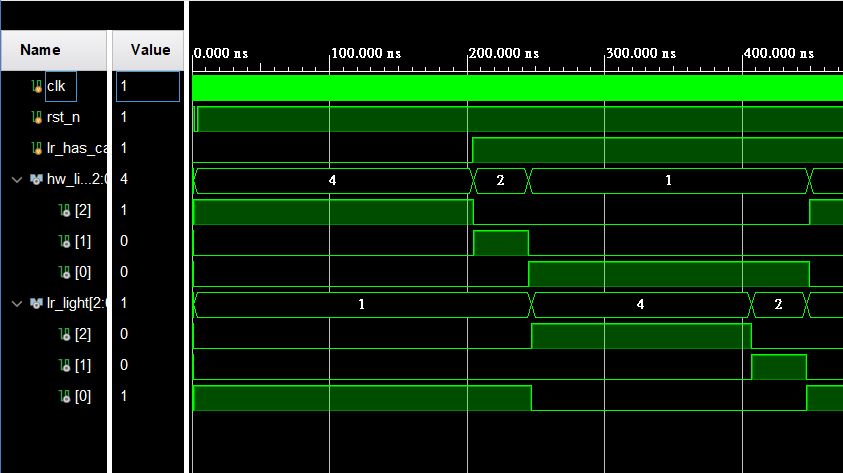
1. State Diagram：



1. 驗證：

測試兩種狀況

* Highway綠燈先過80cycle，Local Road 才有車。



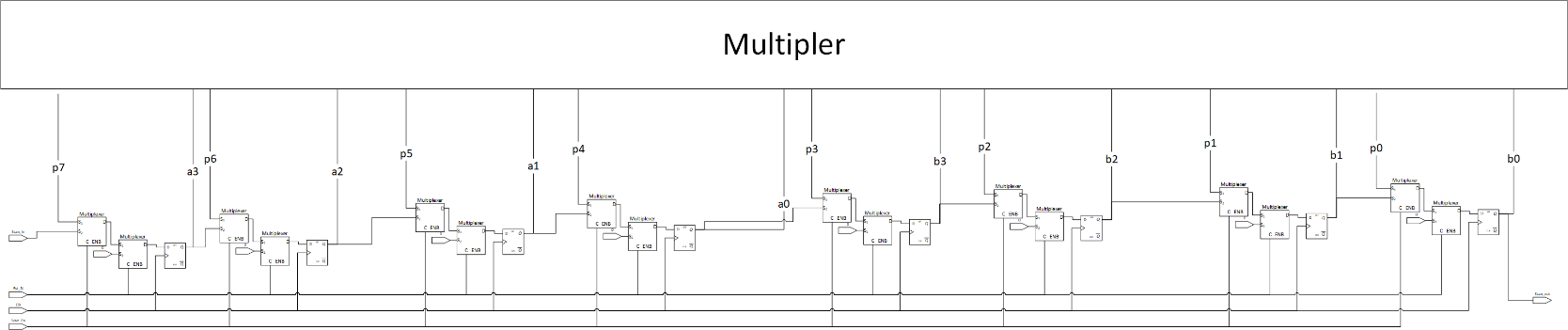
* Highway綠燈未過80cycle，Load Road 就有車。



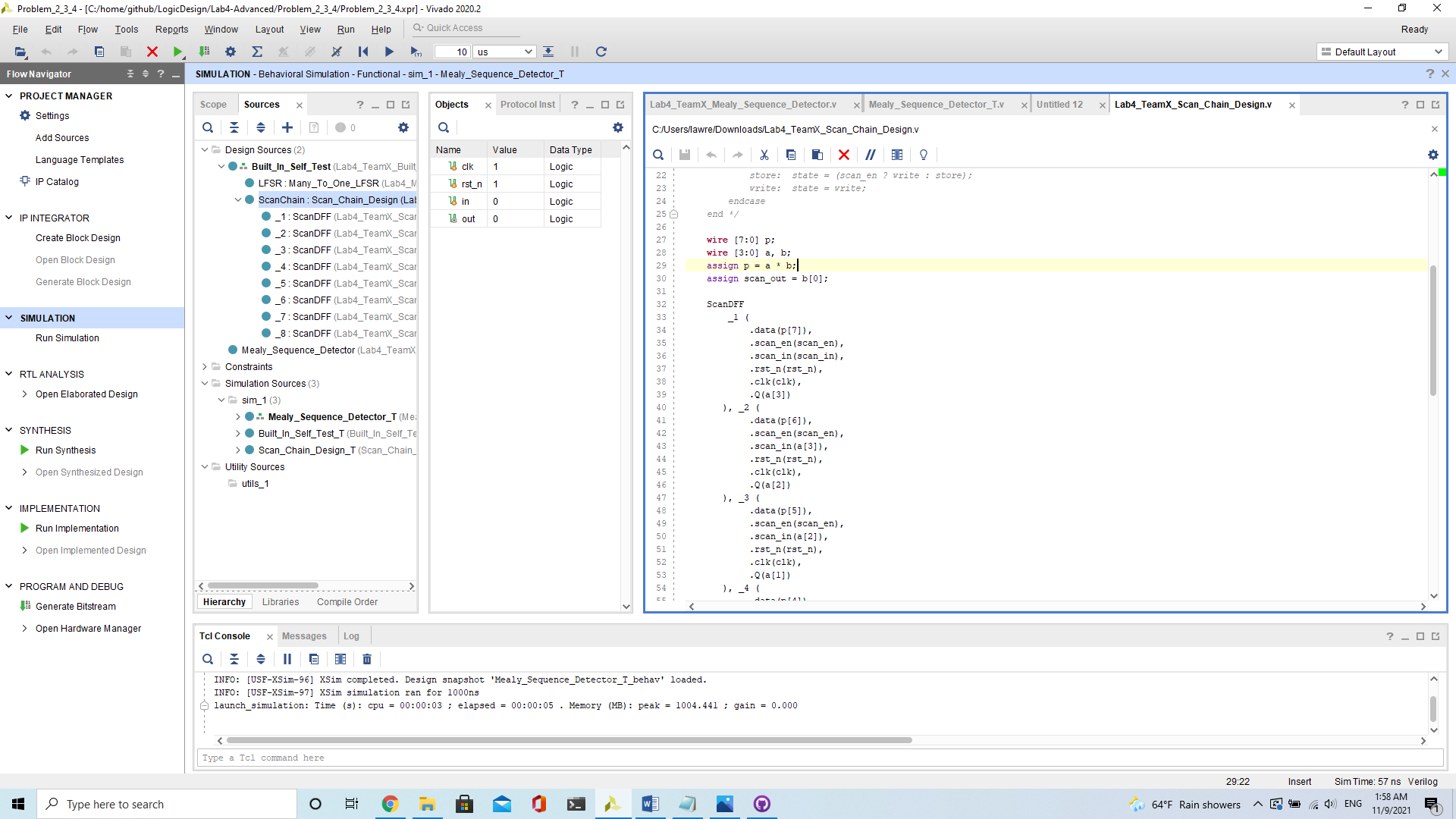
## Scan Chain Design

1. 設計說明：

首先，跟著規格設計Scan DFF。



接下來，利用內建的乘法運算子實作乘法器，再將資料打回Scan DFF。

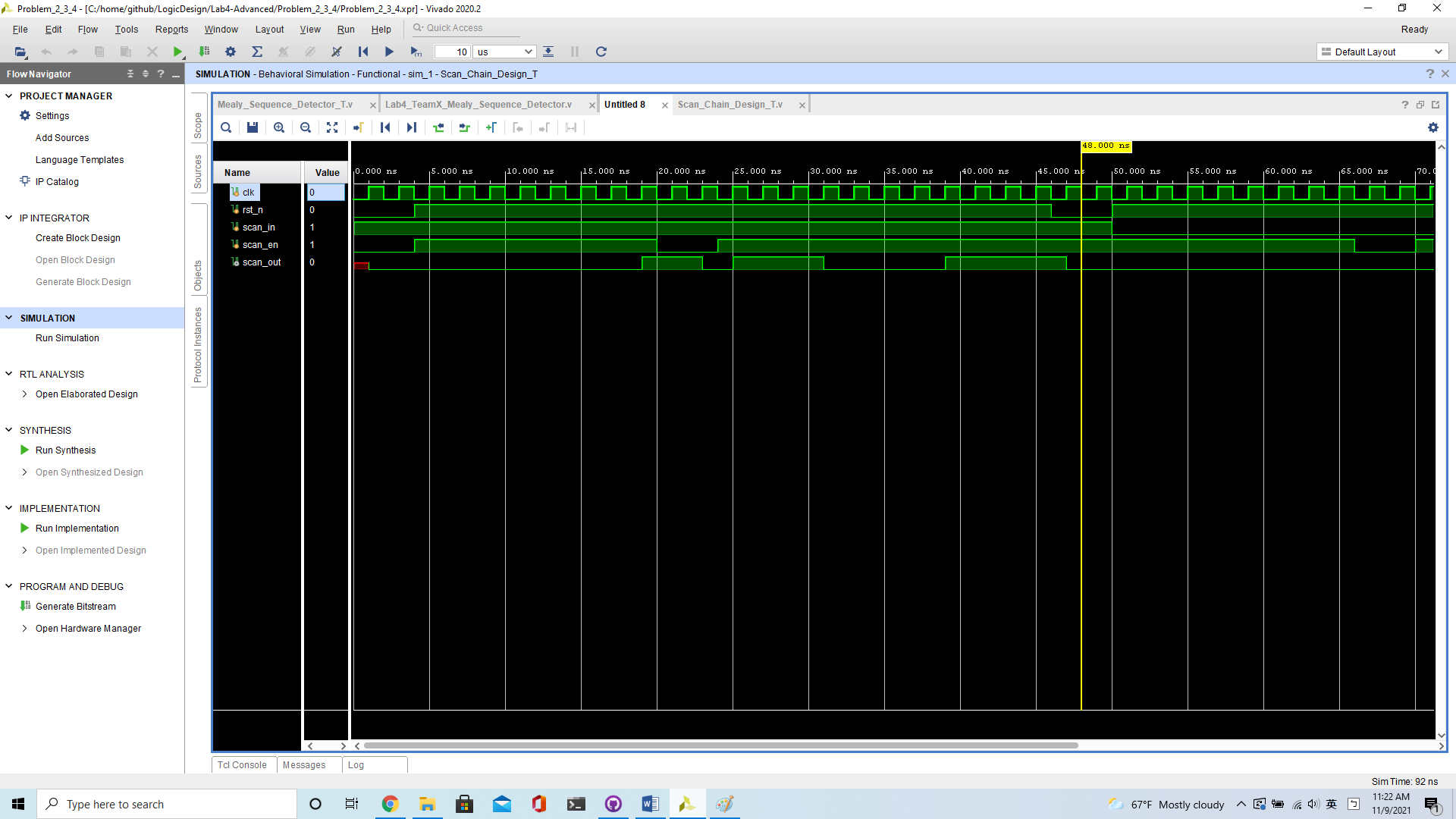


利用上述兩個設計，即可完成題目所需之要求。

1. 驗證：

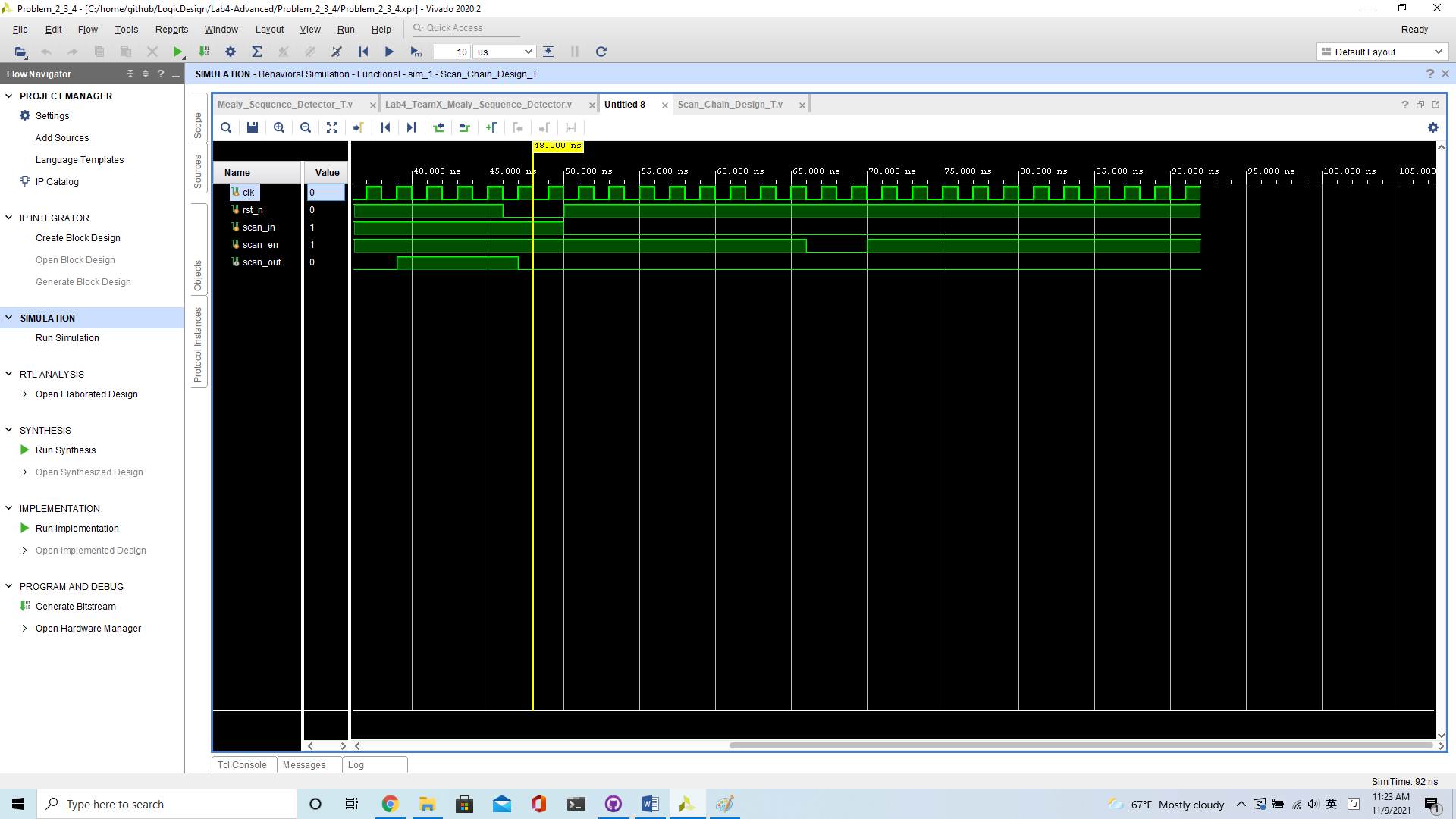
測試資料 #1

讓所有輸入都是高電位。



測試資料 #2

讓所有輸入都是低電位。



## Greatest Common Divisor

1. 設計說明：

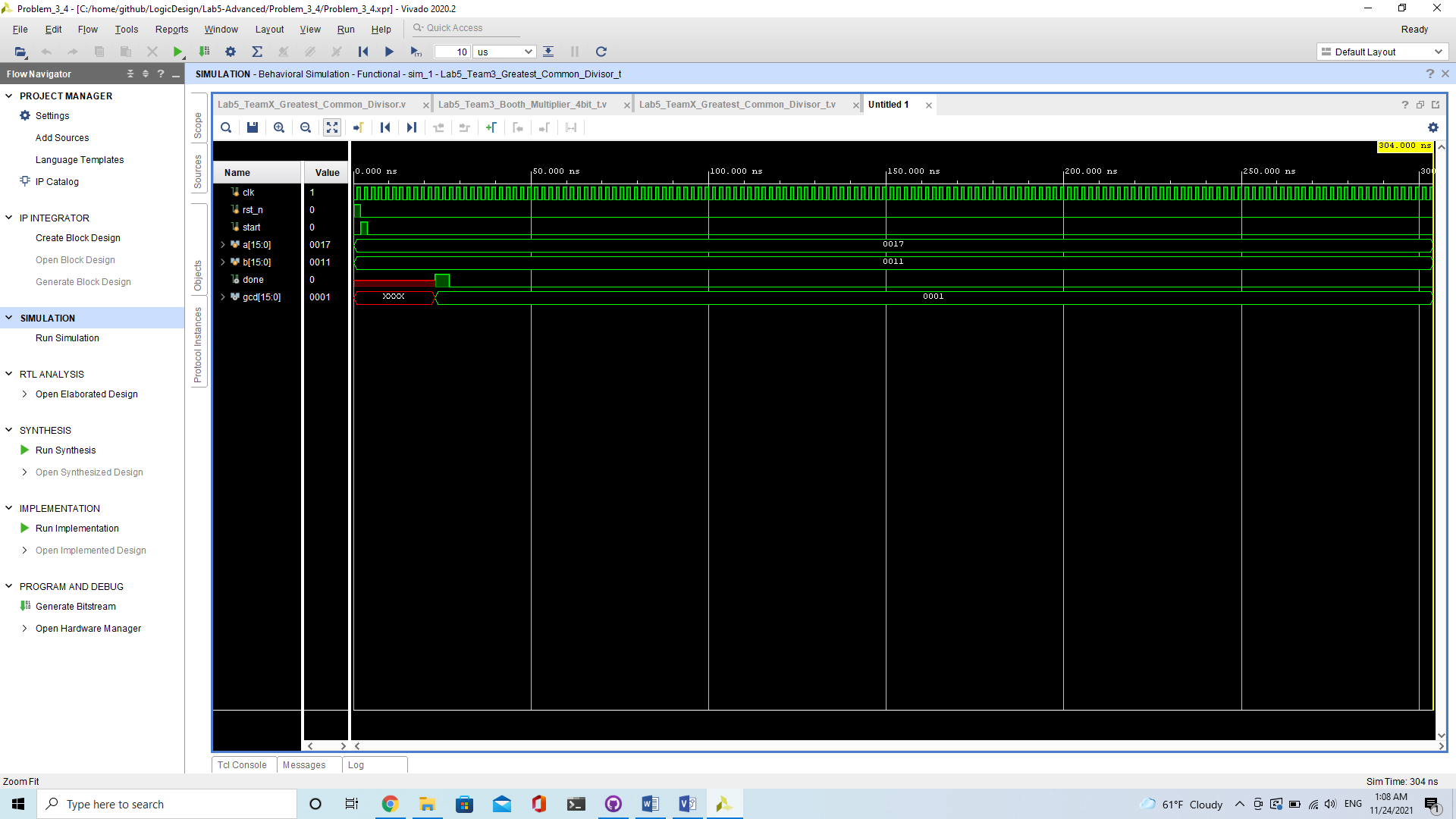
依照助教給定的虛擬碼進行實作，在重設時改變狀態到WAIT，在啟動時改變狀態到CAL，完成運算時（也就是A,B其中之一歸零時）改變狀態到FINISH。

為了維持有兩個時脈的Done，在完成運算時，我們將Done設為高電位，並且設立一個Dummy State，在FINISH狀態等候一個時脈後，轉移到Dummy State，並且將Done設為低電位。藉由設立一個新的Dummy State，就能輕鬆延長Done的維持時間。

1. 驗證：

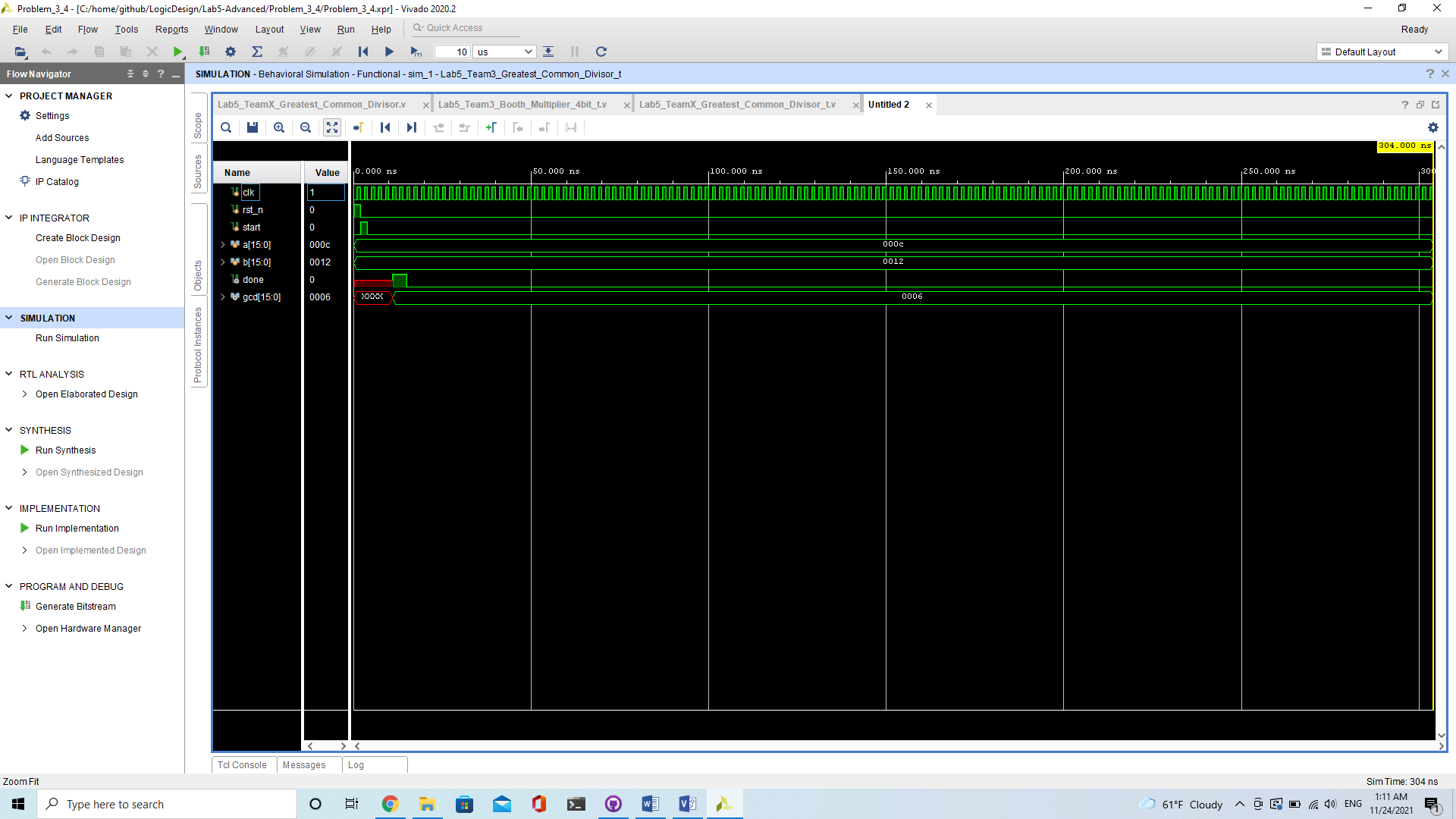
測試資料 #1

A=23, B=17



測試資料 #2

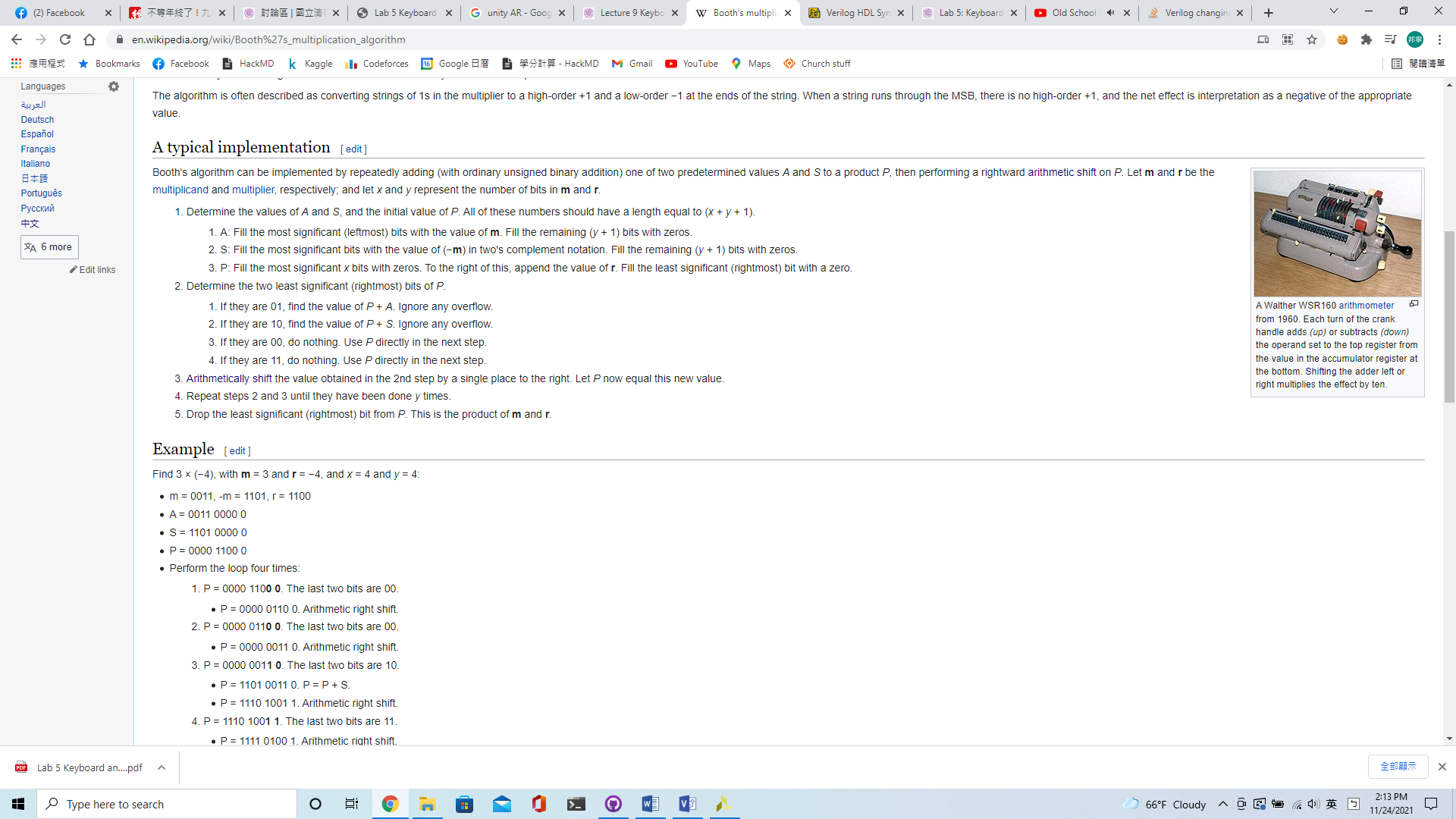
A=12 B=18



## Booth Multipler

1. 設計說明：

參考自維基百科



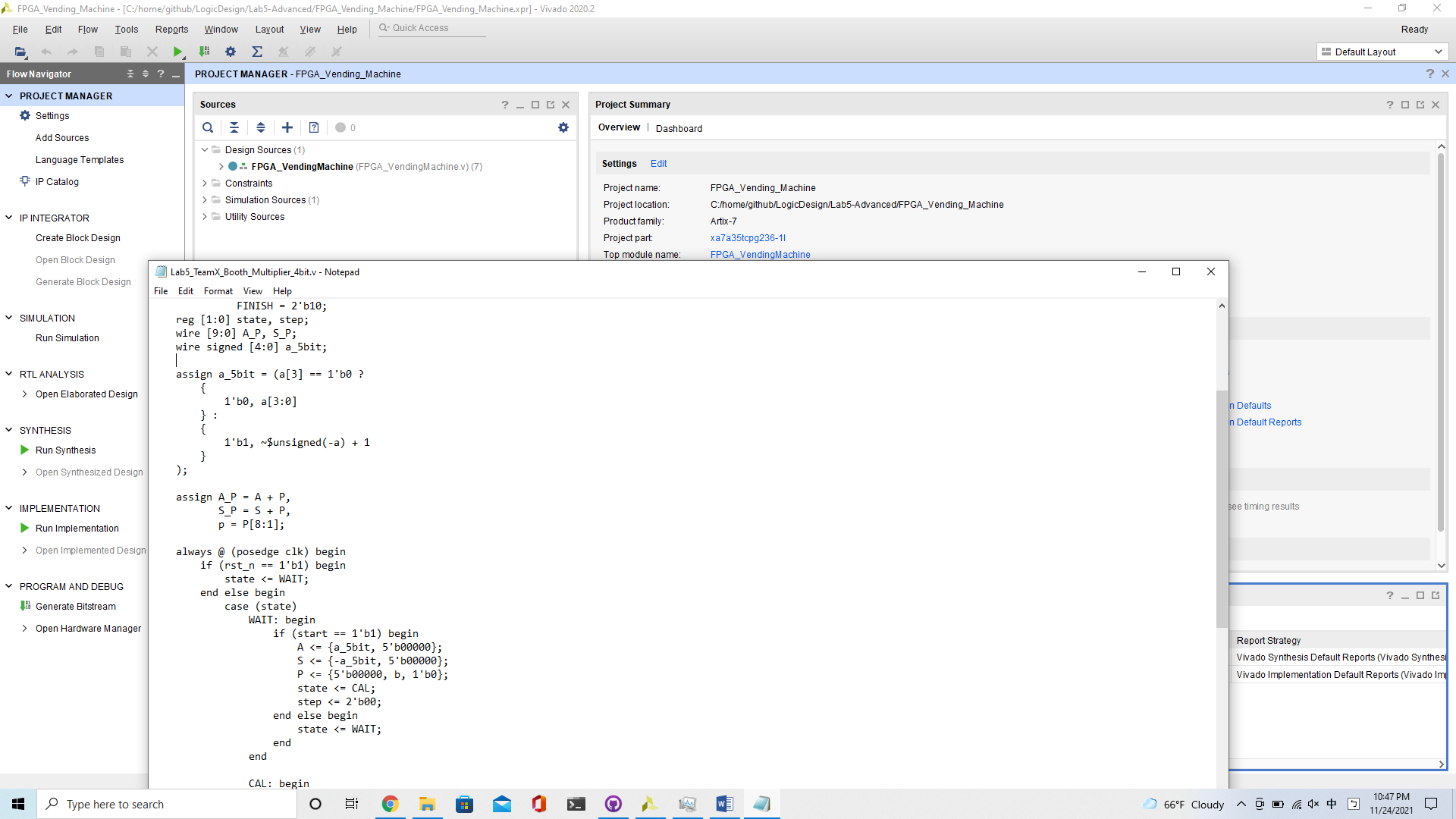
該算法的核心精神為「利用二進位的性質，將一堆加法組合成少量的加法與減法」，如果要計算一個位數布林乘法，該演算法需要個時脈，並且需要個電晶體。與暴力法相較，暴力法需要個時脈，但是需要個電晶體，由此可知該算法有「以時間換取空間」的特性。

假定為最負整數，也就是在位數二補數系統下的，的值沒辦法在位數二補數系統下表示，因為。為了避免溢位，我們將轉為位數二補數系統，並在前面append一個空格，藉此放入額外的位元。

從位數二補數系統轉換到位數二補數系統的流程如下，我們分成兩個情況來討論，第一種情況是待轉換的數字。在該情況下，僅需複製原本的數字，並在最高位補上0即可。

第二種情況是待轉換的數字，首先，我們要計算在位數二補數系統的，並將執行位元反轉後再加一，最後在最高位補上1即可。

上述流程可以用這段簡潔的虛擬碼表示。

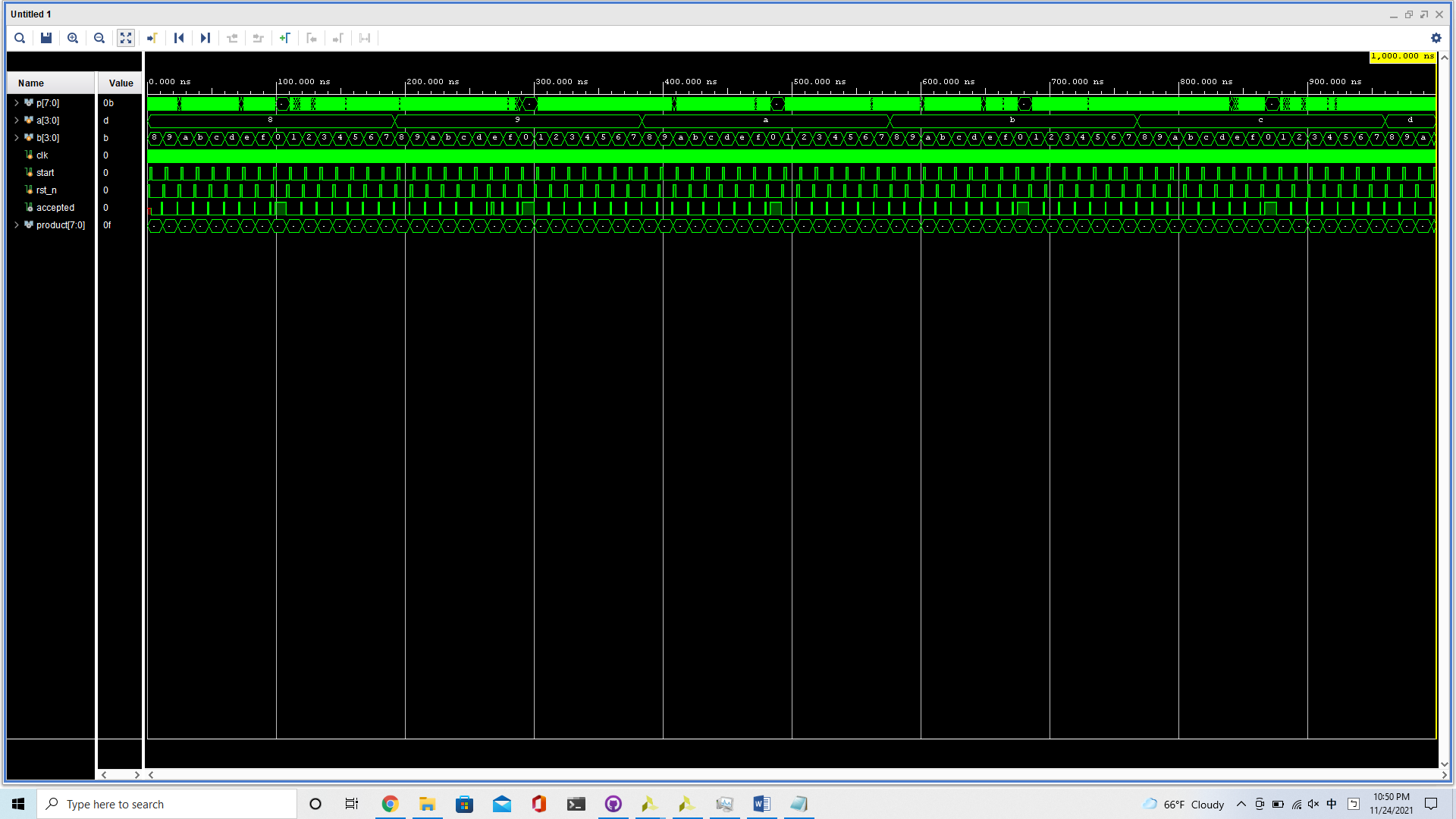


最後，跟著維基百科給定的方法，即可完成晶片設計。

1. 驗證：

測試資料 #1

枚舉所有可能，並用肉眼觀察是否相等，Accept為兩者是否相等。



# Contribution

## 吳邦寧

第三、四題之實作及報告之撰寫、販賣機之實作。

## 張芯瑜

第一、二題之實作及報告撰寫、音階盒之實作。

# What have we learned?

1. 多去看討論區！
2. 瀑布開發的效率很高，非常適合趕工用。
3. 不要寫髒Code，未來的自己會討厭你。
4. 盡量維持模組的可讀性，未來的自己會很感謝你。