Lab ?: ??????????

Group 3 : 109060013張芯瑜 109062328 吳邦寧

# Design Explanation

## Content-addressable memory

設計說明：

利用register創造一個可存放16個8 bit數字的空間CAM。

當ren = 1時，我們由存放位置序號較大的，也就是CAM[4’b1111] 開始找起，整個運算利用if else判斷。若有存放的數字與din相同，則將下一個clock的dout設為該數字存放位置，且hit設為1。若有兩位置皆存放與din相同的值，會因為先後順序而選擇位置序號較大的作為輸出。當跑完整個CAM仍然沒找到與din相同的值時，則輸出dout = 0 和 hit = 0。

當ren = 0且wen = 1時，將din值在下一個clock時存入CAM的第addr個位置。

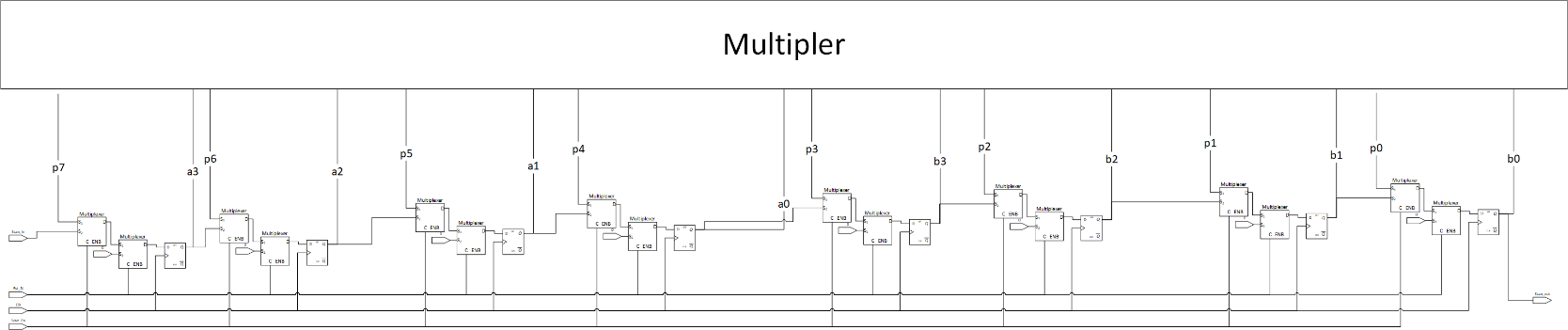
當ren = 0且 wen = 0，則輸出dout = 0 和 hit = 0。

驗證：

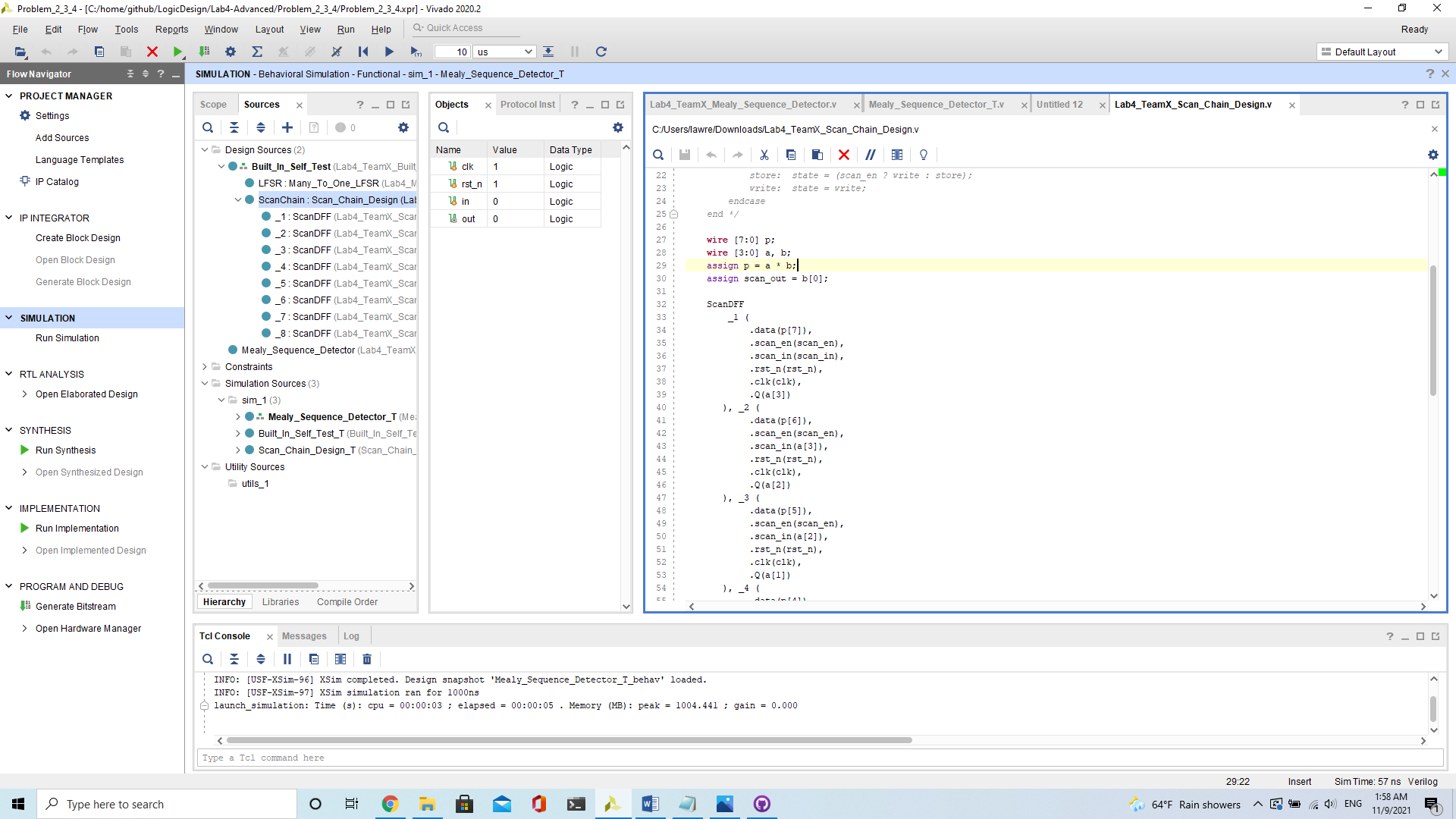


## Scan Chain Design

設計說明：

首先，跟著規格設計Scan DFF。

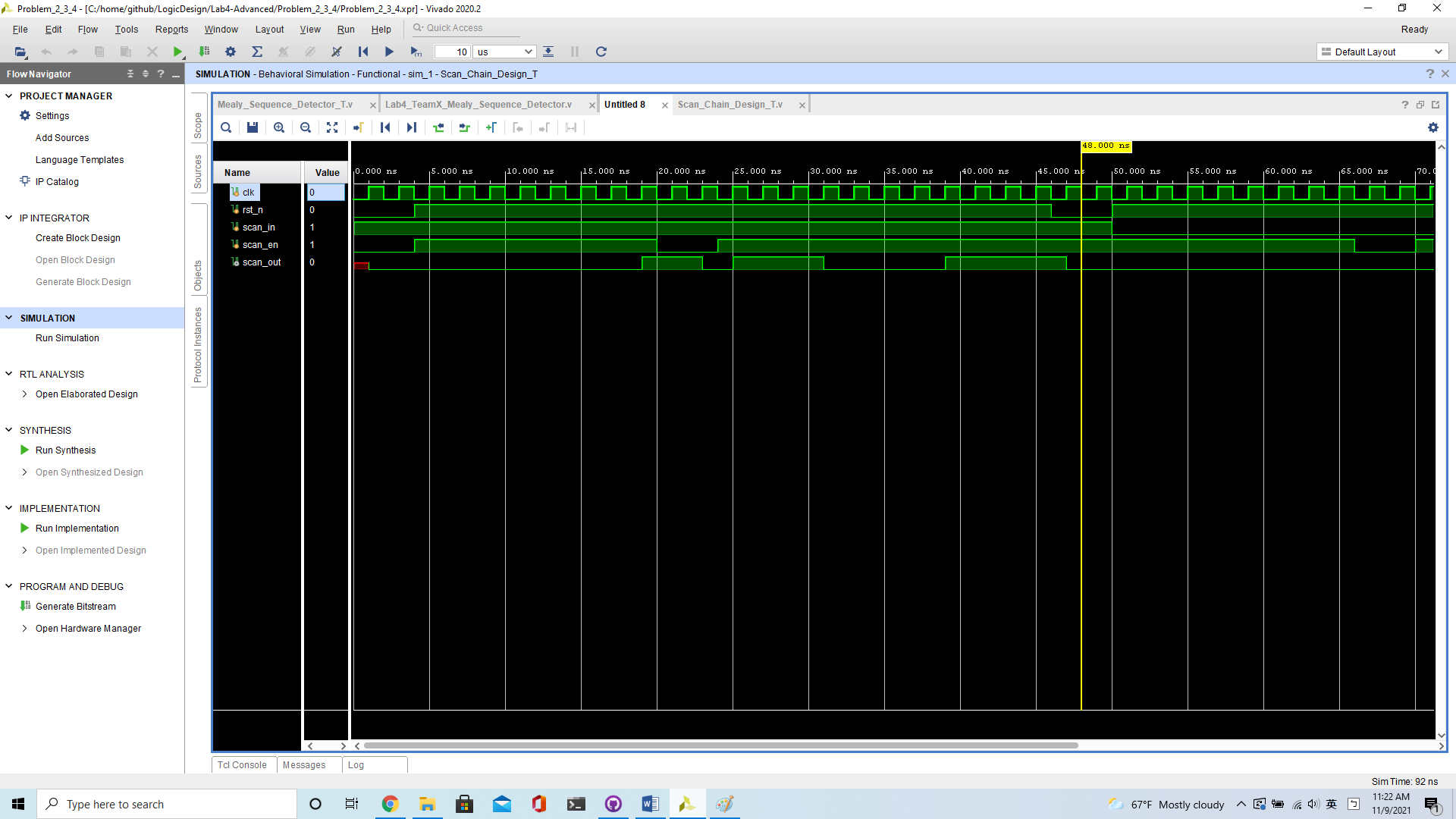
接下來，利用內建的乘法運算子實作乘法器，再將資料打回Scan DFF。



驗證：

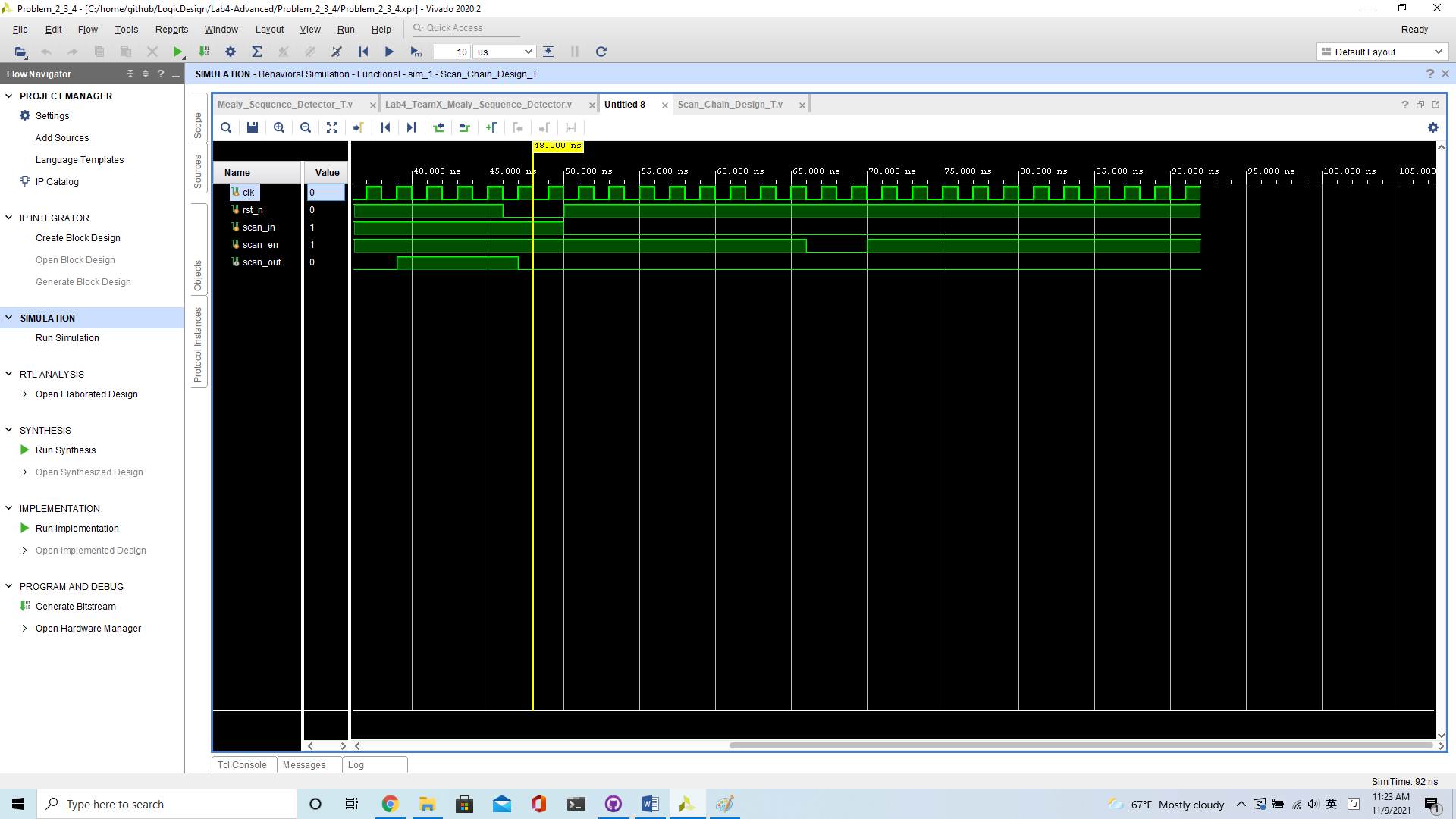
測試資料 #1

讓所有輸入都是高電位。



測試資料 #2

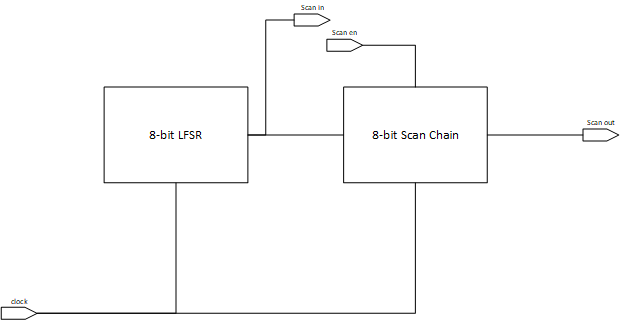
讓所有輸入都是低電位。



## Built In Self Test

設計說明：

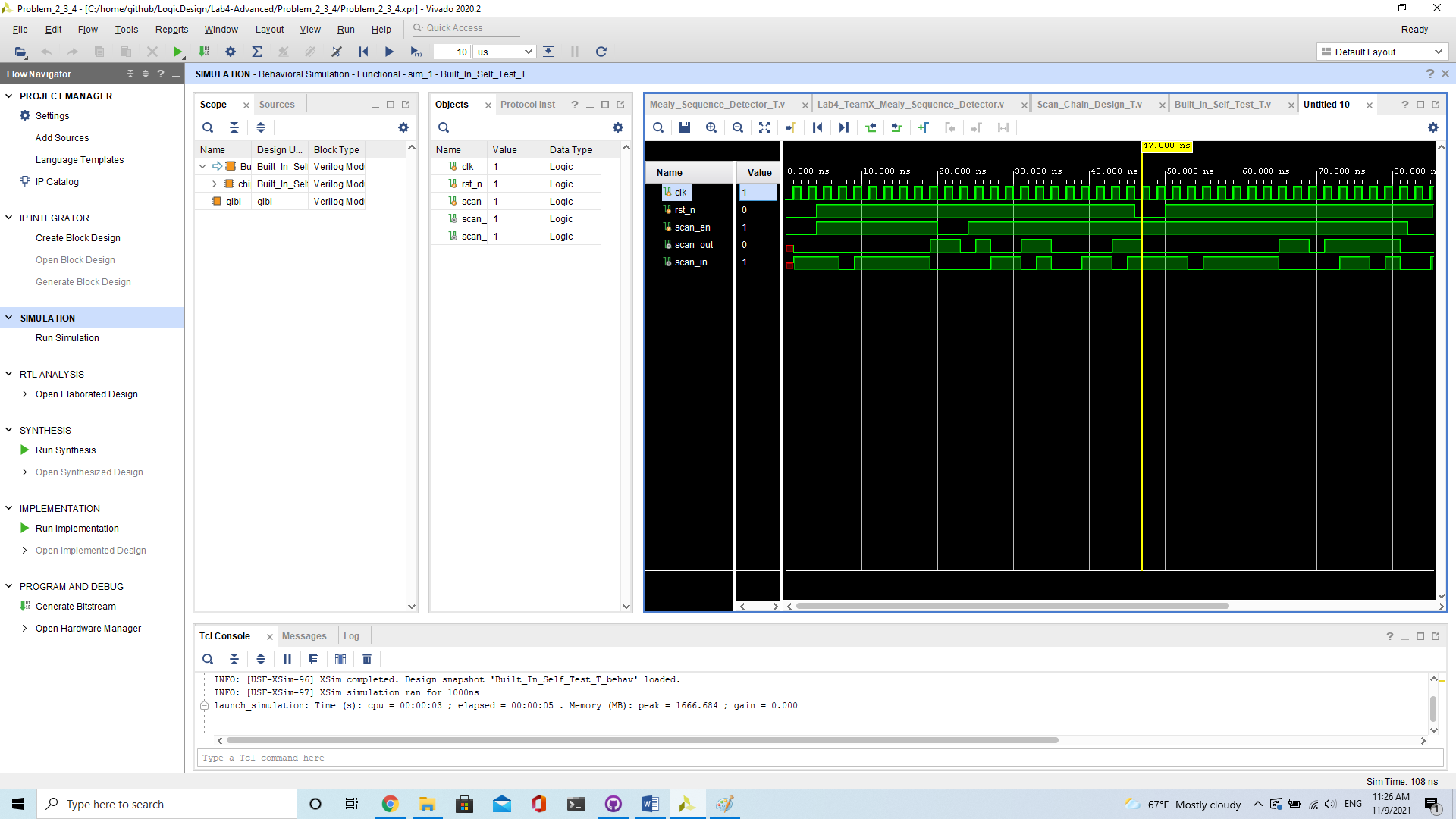
跟著規格設計BIST即可。



驗證：

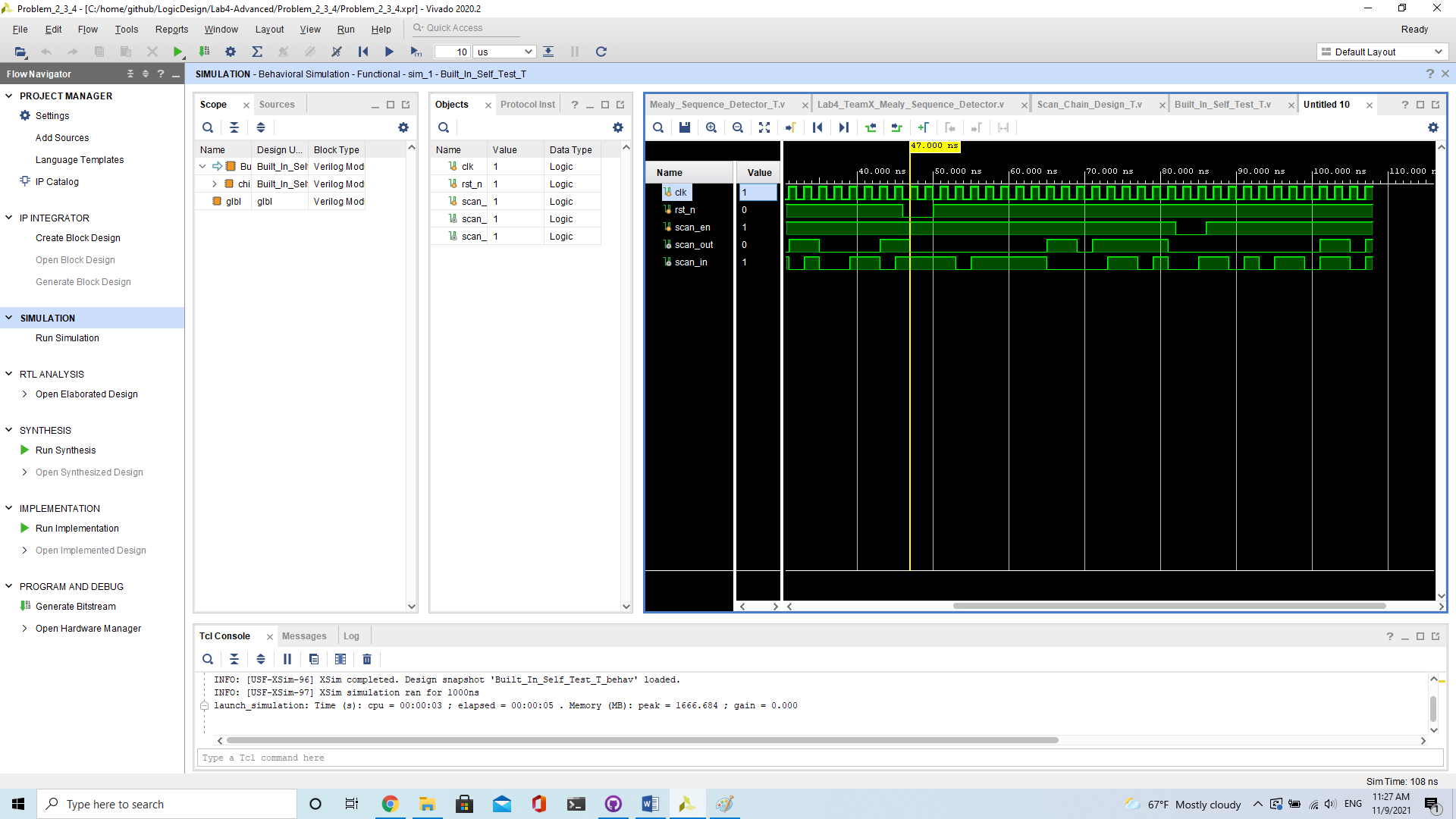
測試資料 #1

讓輸入等8個clocks



測試資料 #2

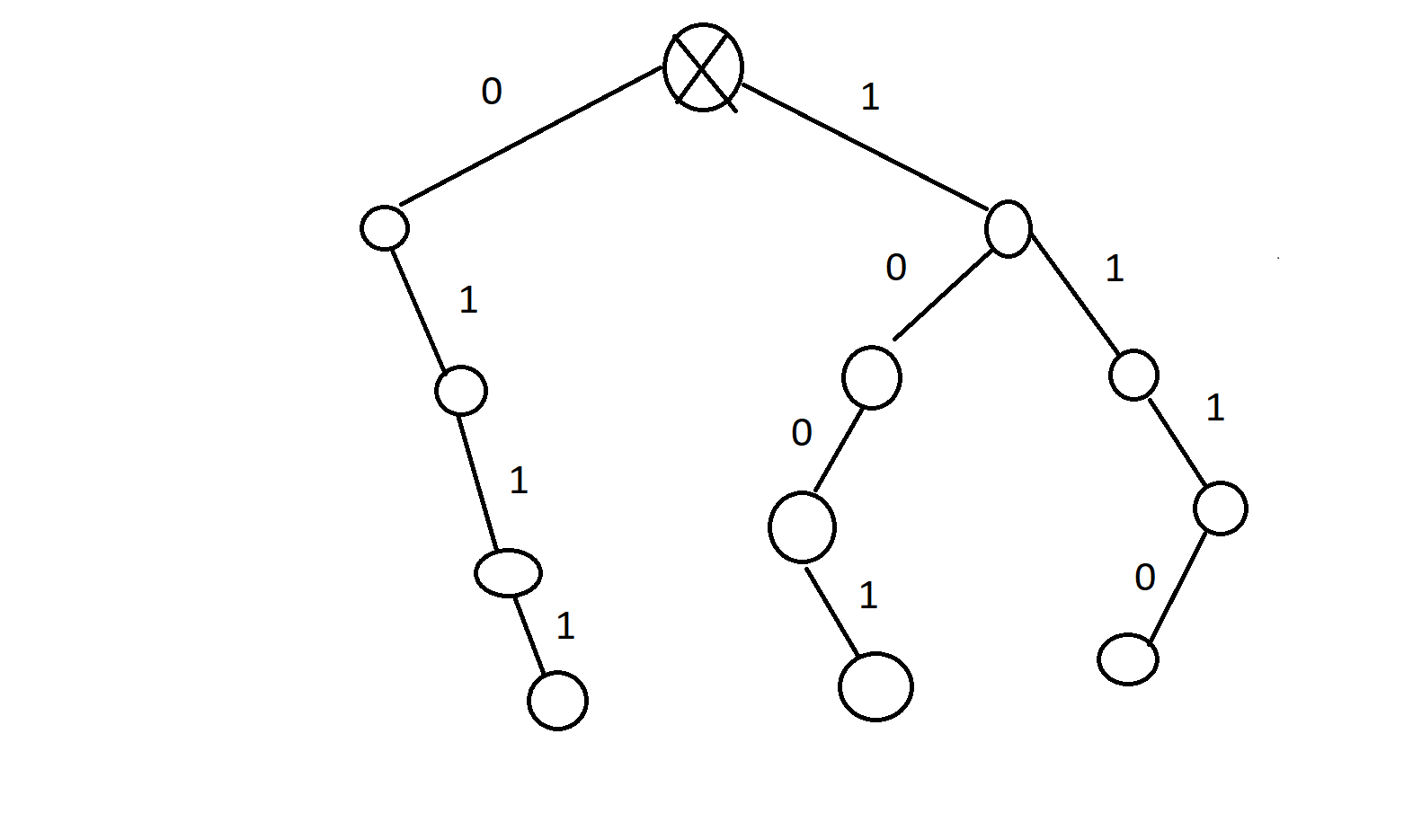
讓輸入等16個clocks



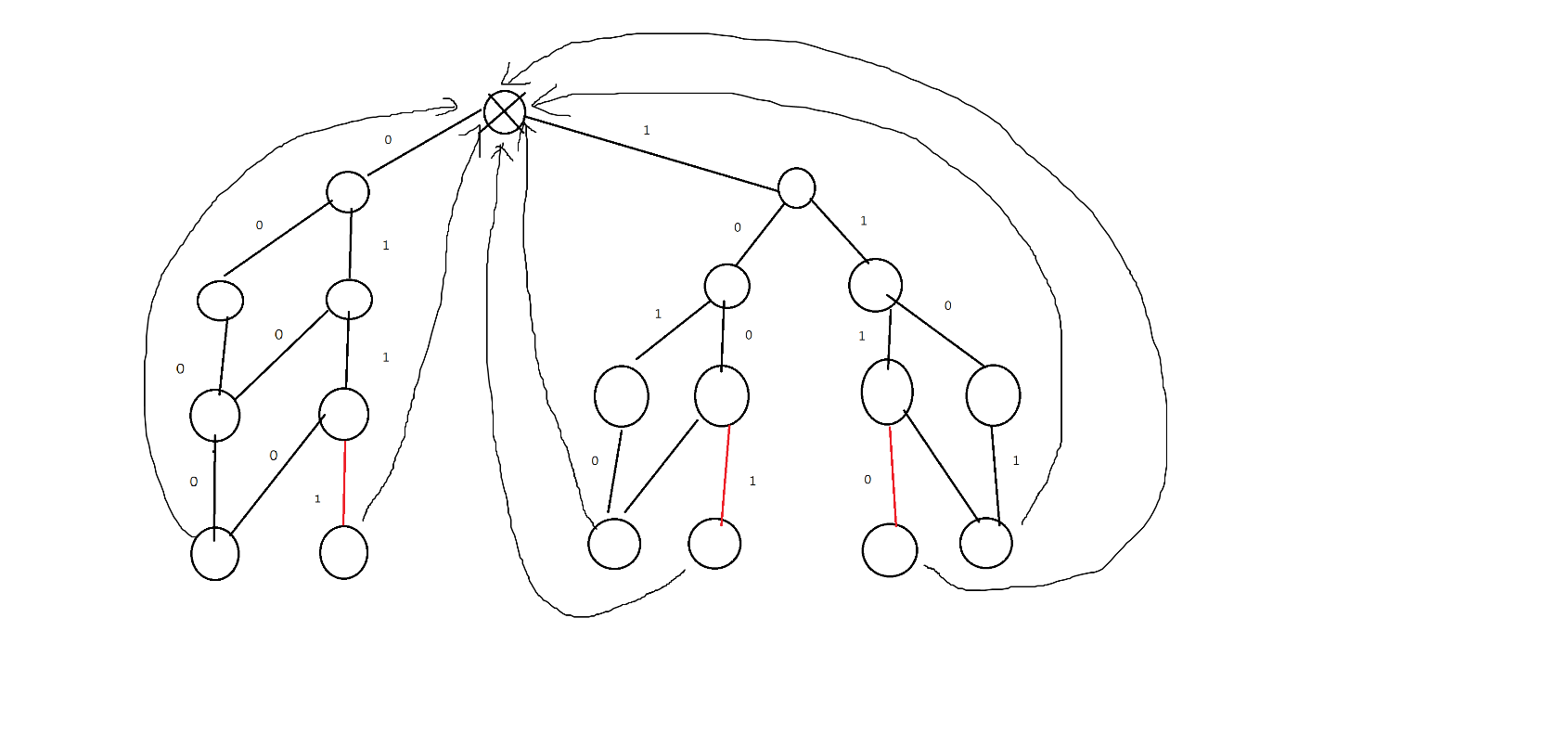
## Mealy Machine Pattern Detector

設計說明：

首先，將所有模式放入字典樹內。

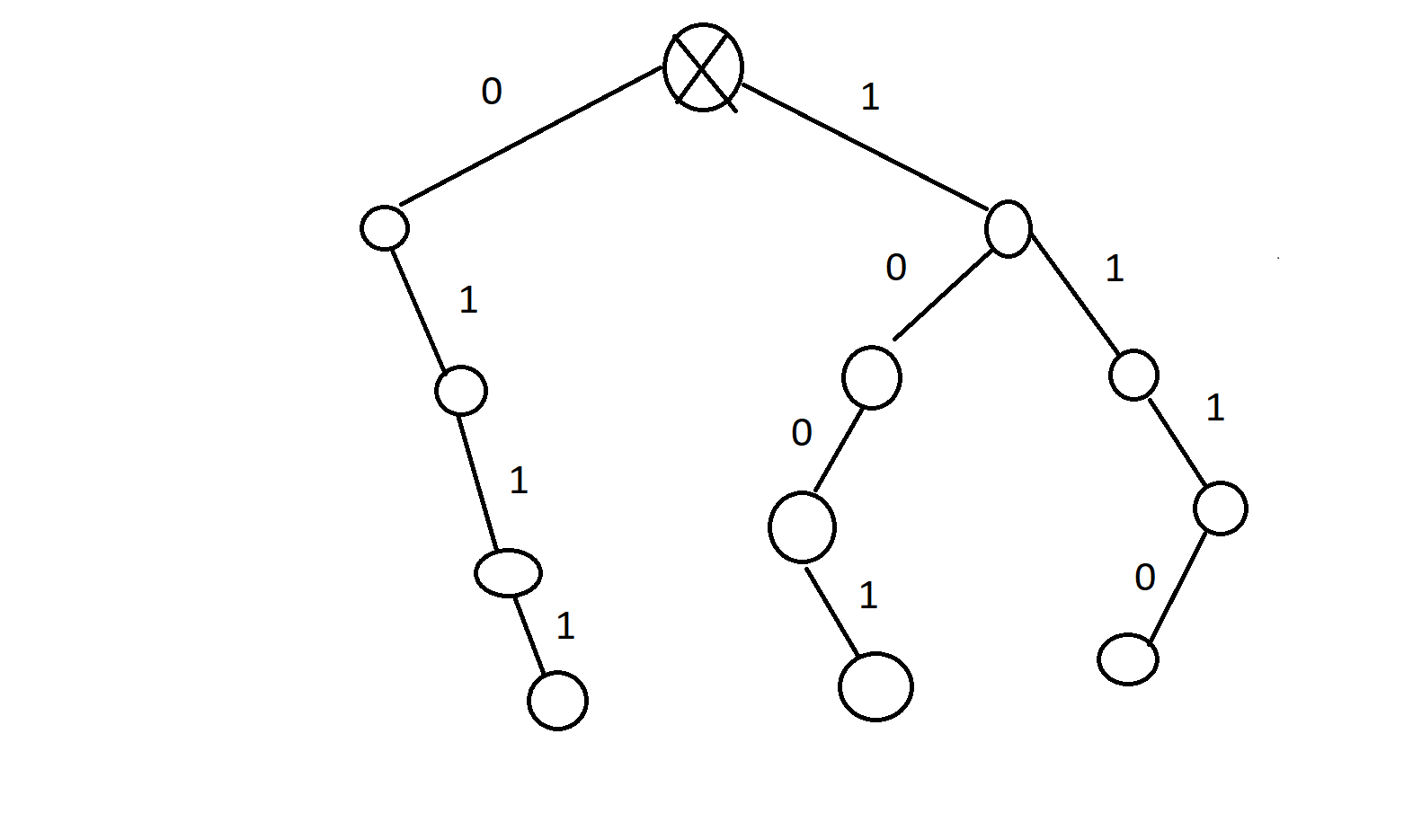


接下來，增加節點，各自代表無法匹配的狀態。紅色的邊代表找到匹配，黑色的邊代表沒有找到匹配。

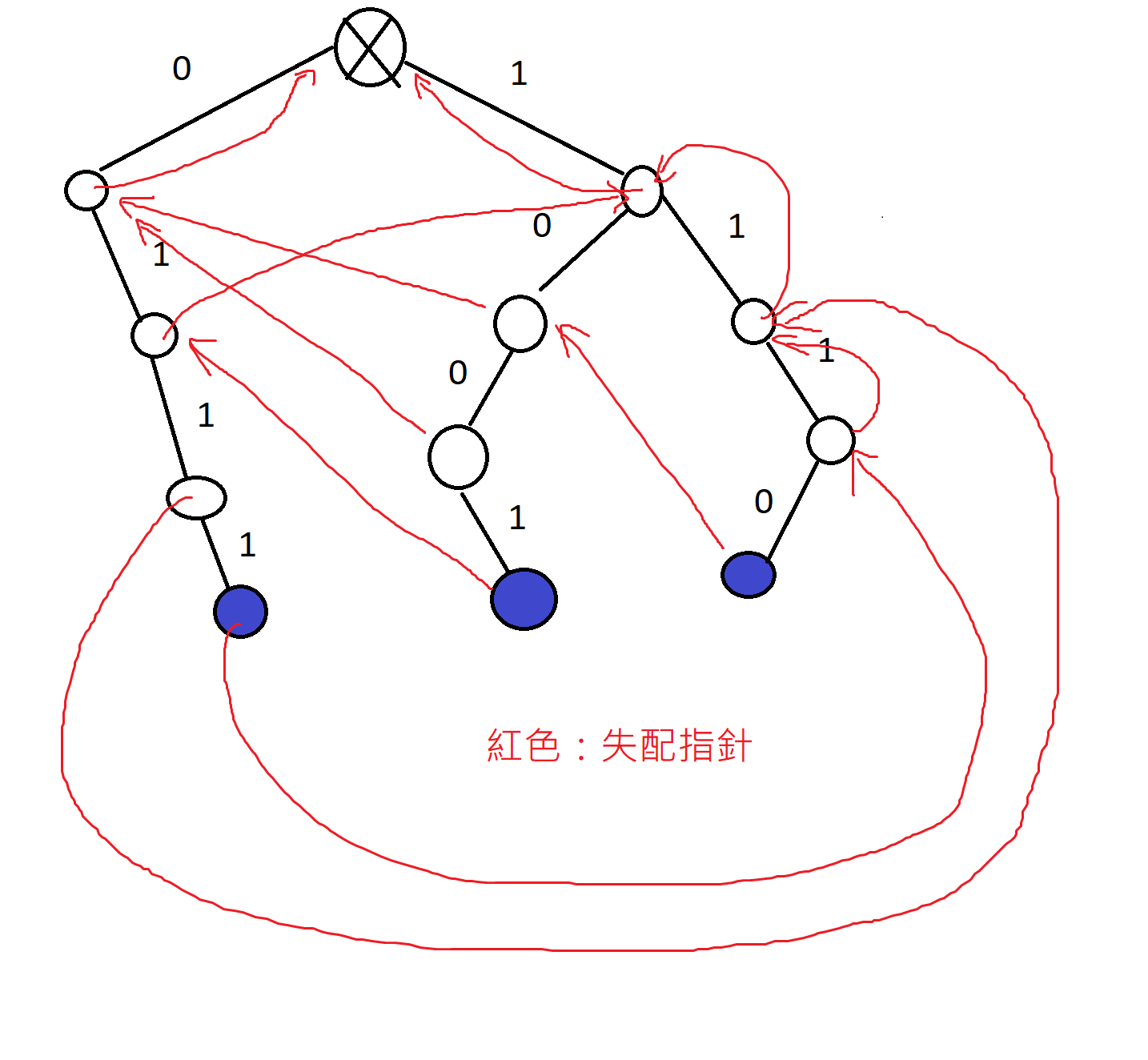


嘗試過但失敗的設計說明：

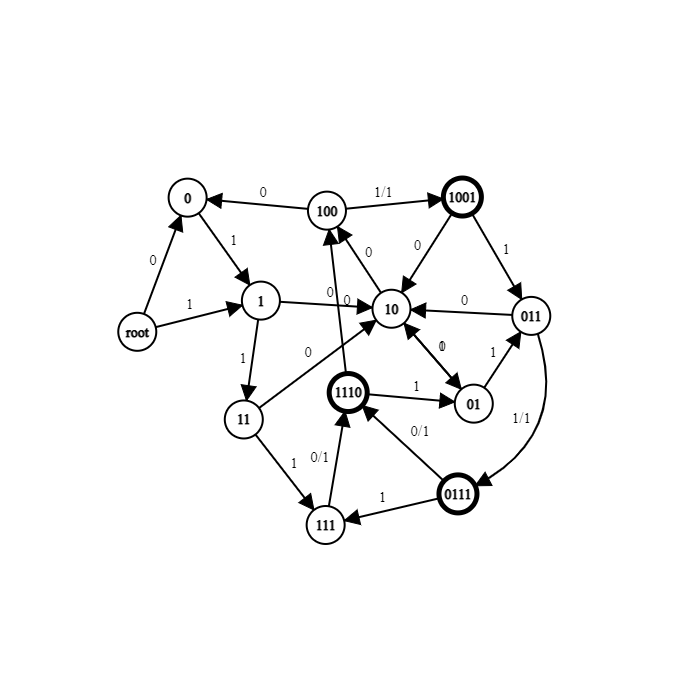
首先將所有模式放入字典樹。



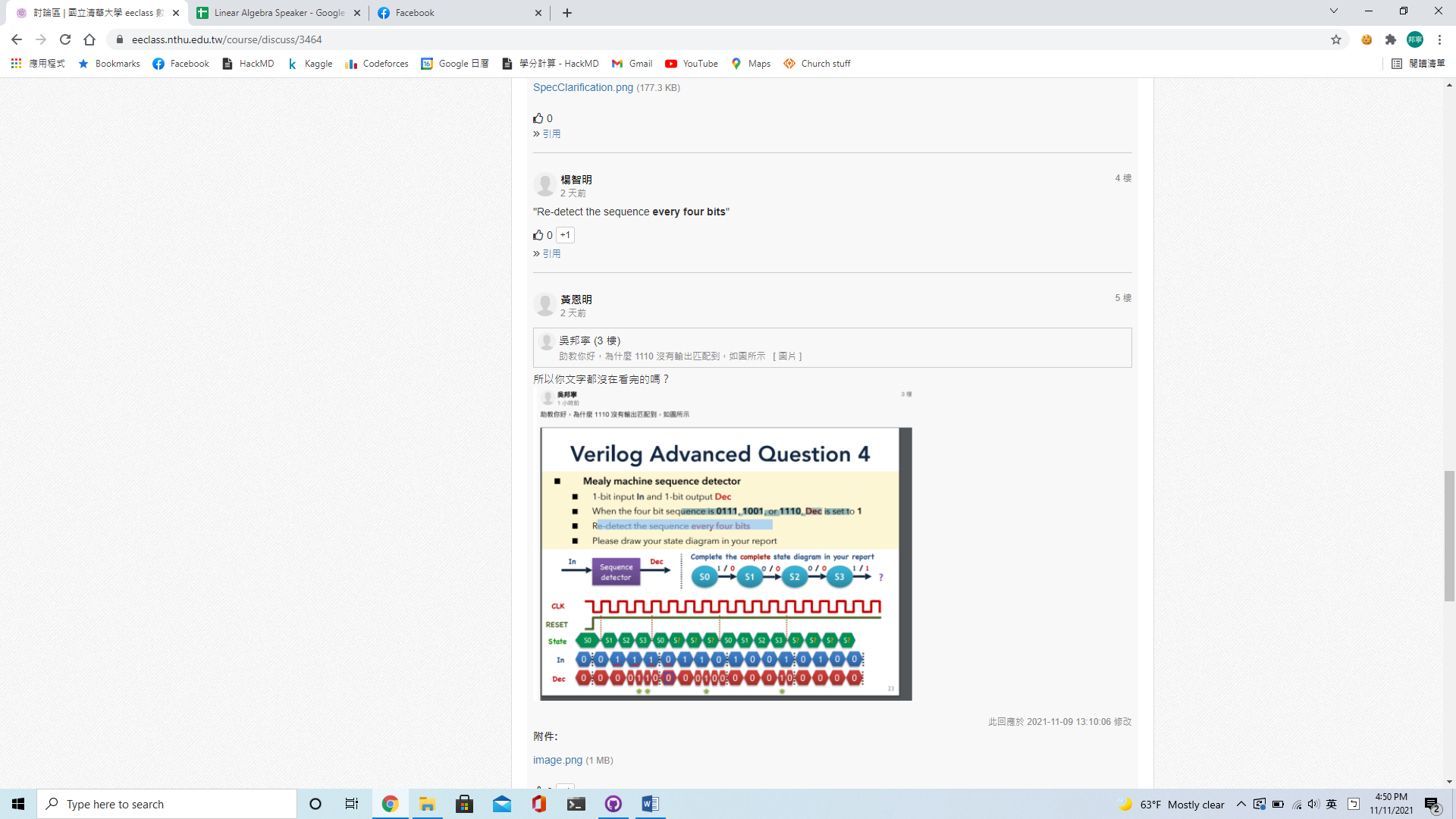
將字典樹編譯成Aho-Corasick自動機。在執行廣度優先搜索時，若是父節點的失配節點沒有對應的子節點，那就再往父節點的失配節點的失配節點搜索對應的子節點，直到走回樹根。重複上述步驟即可建造Aho-Corasick自動機。



移除走不到的失配指針，並將連續轉跳的失配指針合成為一條邊，即可轉換為Mealy Machine。



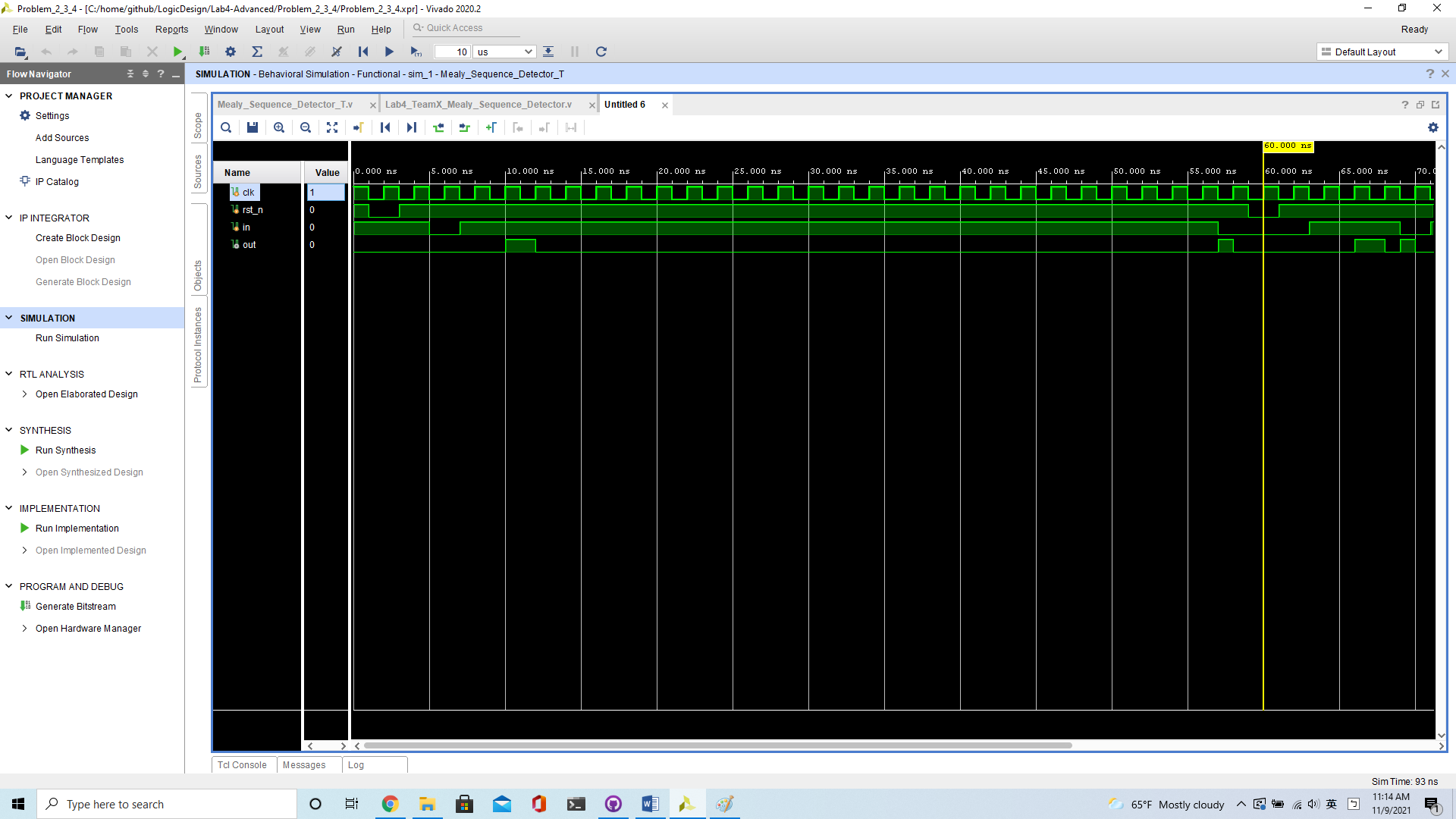
但是題目的要求是每四個位元進行一次檢驗，所以不適用Aho-Corasick自動機。



驗證：

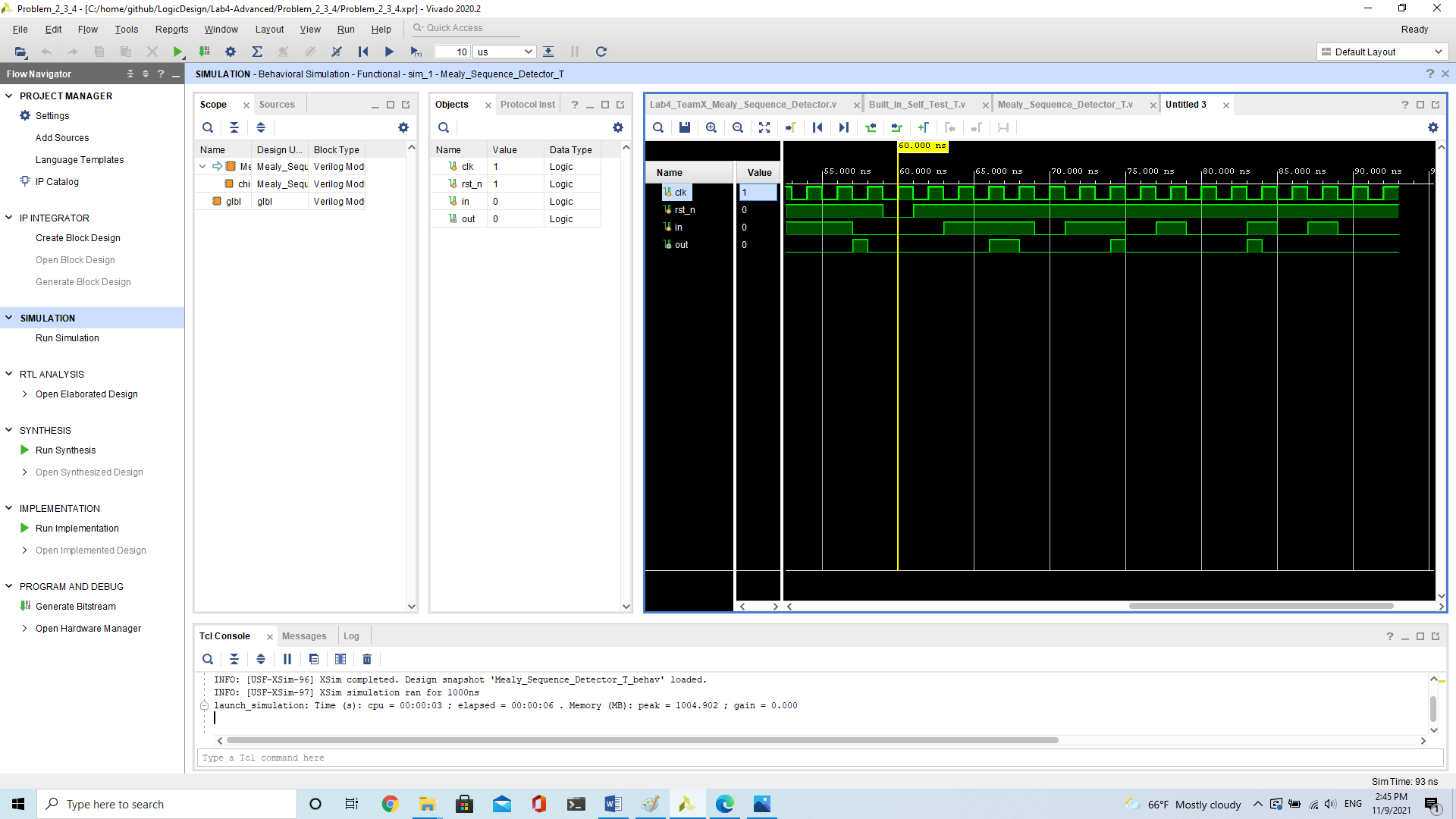
測試資料 #1

先讓資料都是高電位，時不時把電位拉下來。



測試資料 #2

同助教測試資料。



# Contribution

## Lawrence Wu

第二、三以及五題之實作以及報告撰寫。

## Ariel Chang

# What have we learned?

1. 沒有測試過的程式碼不要用