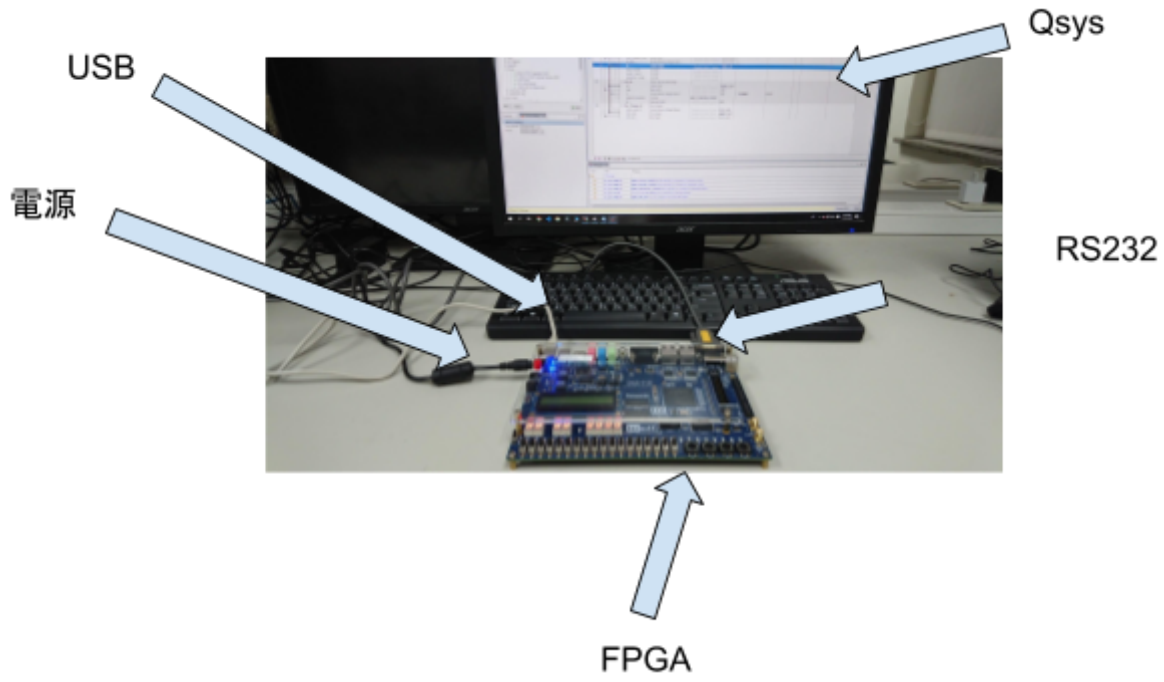
數位電路實驗 Lab2 Report

B07901056 張凱鈞、B07901100 林亮昕、B07901108 馬健凱

壹、實驗器材及架設



貳、使用方式與步驟

1. 將verilog 經由Quartus compile
2. 將compile後的code燒錄到FPGA板上
3. 燒錄成功後，按下key0 reset
4. 在command line跑rs232.py
5. 每次要decode的時候就要重新reset

參、實作設計

(note: 以下FSM中，藍色字為執行內容，紅色字為觸發條件)

一、RSA256 Core

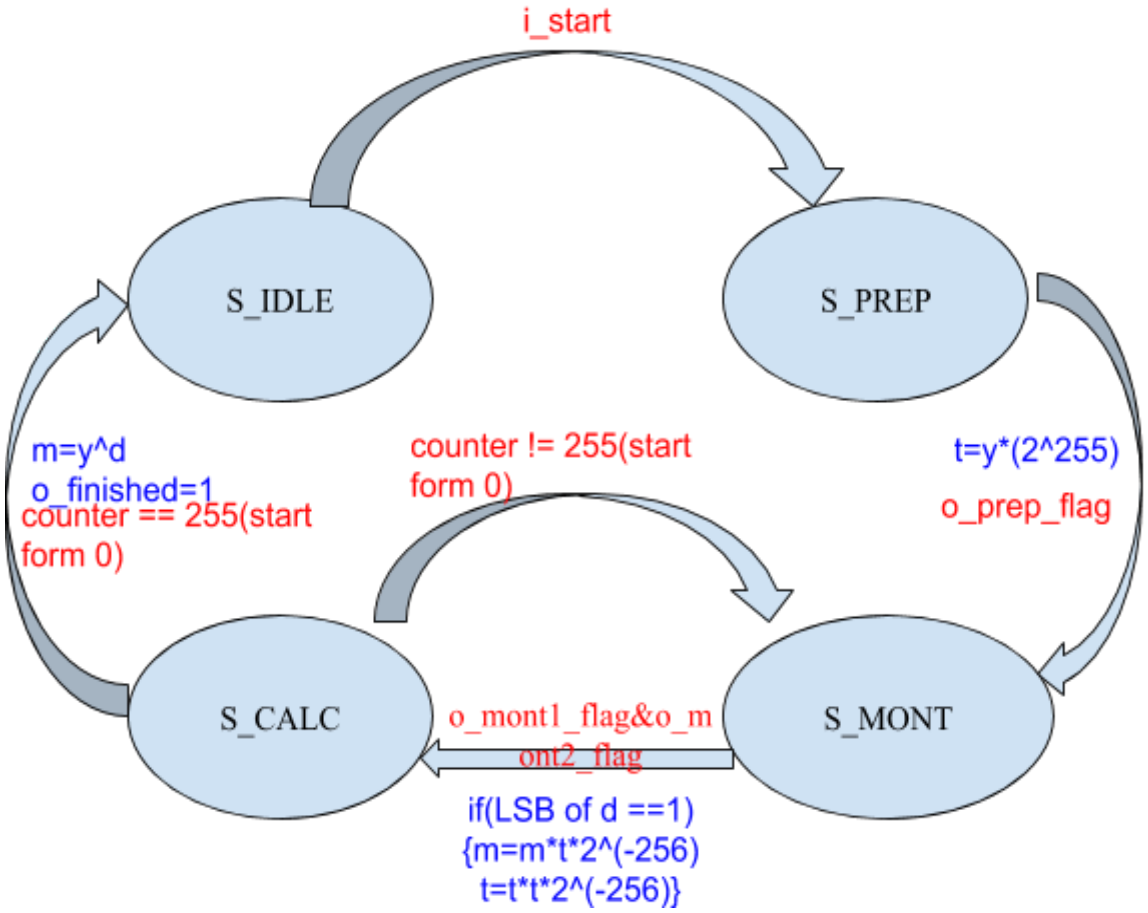


Fig.1 Core FSM

當i_start拉成1後，Core會開始計算進入PREP階段，算出t=y*2^256(modN)幫助之後的MONT方便計算，之後進入MONT階段計算藍色字的內容並將counter+1，之後進入CALC階段，當counter<255時回到MONT繼續重複計算，最後當counter=255時得到m=y^d(modN)即為所求

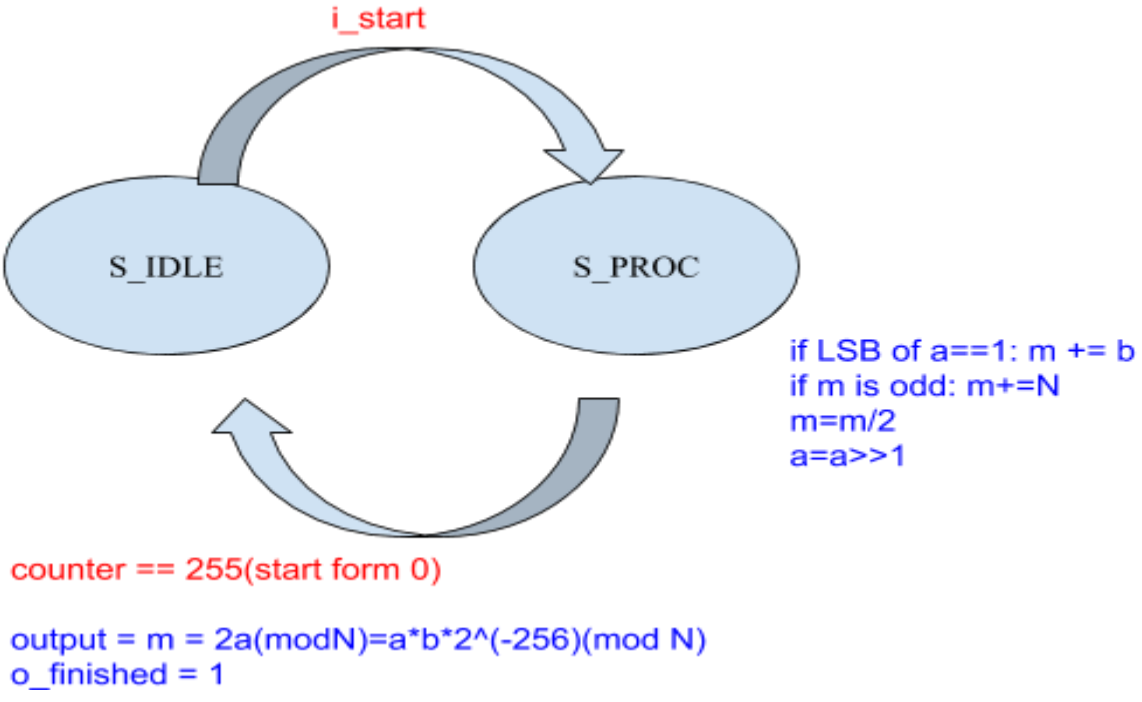


Fig.2 Mont FSM

當i_start拉成1後，Core會開始計算進入PROC階段(couter設成0)，之後每次計算藍色字的內容並把counter+1，直到counter=255時再計算最後一次並modN，最後得到output=a*b*2^(-256)(modN)即為所求

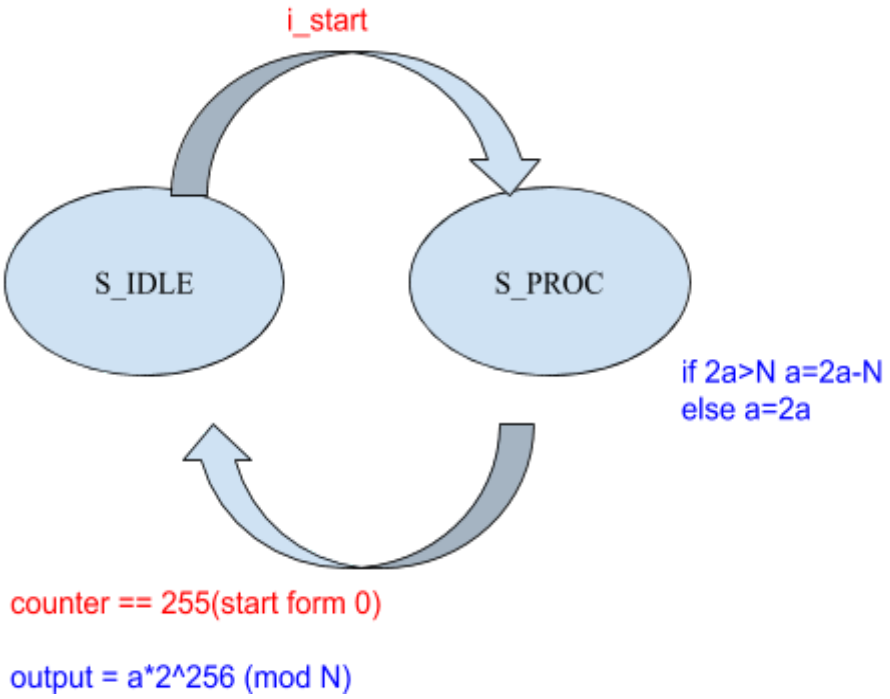


Fig.3 PREP FSM

當i_start拉成1後，Core會開始計算進入PROC階段(couter設成0)，之後每次計算藍色字的內容並把counter+1，直到counter=255時再計算最後一次並modN，最後得到output=a*2^(256)(modN)即為所求

二、RSA256 Wrapper

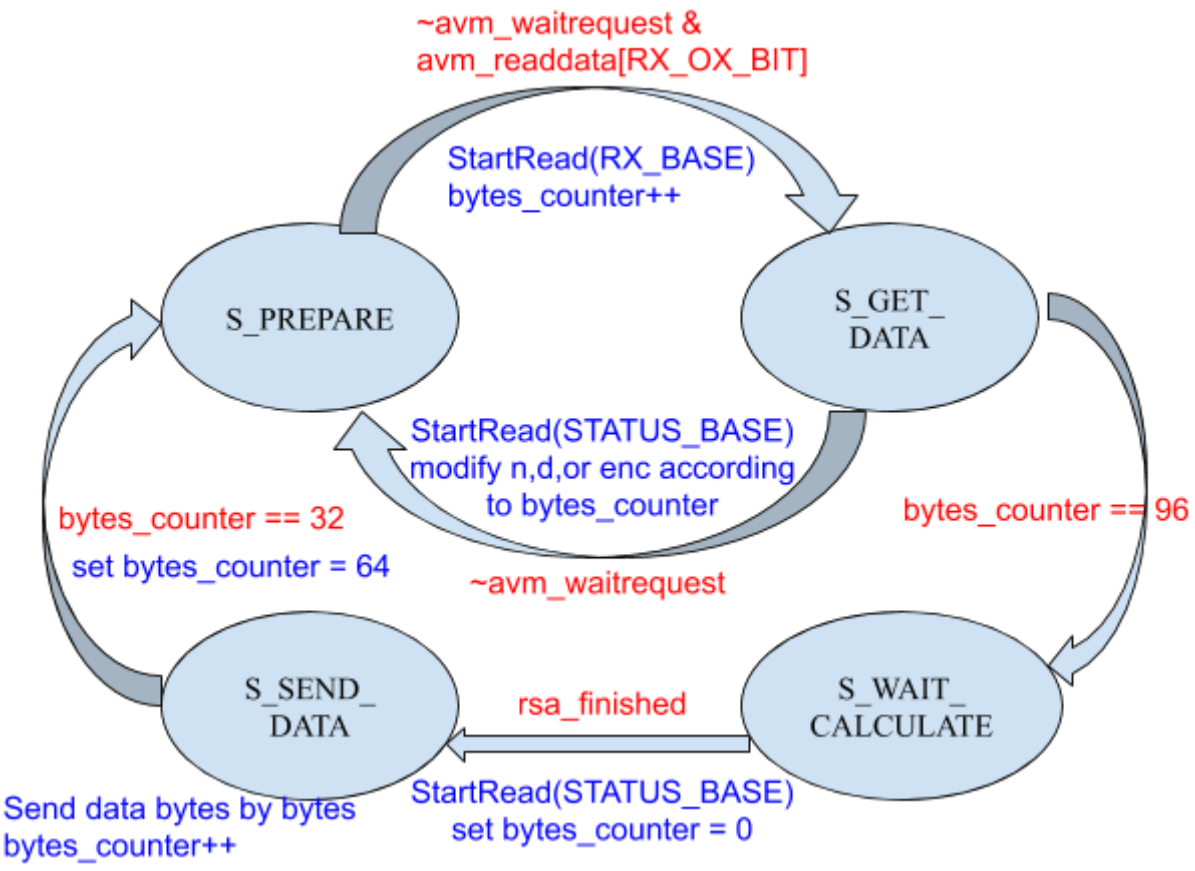


Fig. 4 Wrapper FSM

在 reset 後，程式會進入 S_PREPARE，並且歸零 bytes_counter。接著會開始接收 N,e,和 encrypted data。接收過程就是輪流讀 RX_BASE 和 STATUS_BASE，一次接收1 byte 的資料，並且根據 bytes_counter 來決定目前在讀取的資料是 N,e,還是 encrypted data (對應到的 bytes_counter 分別是1-32, 33-64, 65-96)。

當資料都接收完後，就進入S_CALCULATE，等待 Core 的運算。等收到 rsa_finished 後，就進入 S_SEND_DATA 回傳解密完的資料，流程類似於讀取資料時。當傳送完資料的時候，就回到 S_PREPARE，並設定 bytes_counter = 64 (這樣下筆資料就會是新的 encrpted data，而非 key)。

肆、問題與挑戰

過程中最大的挑戰在於 Core 的 debug，雖然演算法本身不難理解，但加密過程中的 data 是一連串的數字，所以很難在短時間內發現問題出在哪裡，需要細心的檢視中間值的傳遞才能找出癥結所在，這之間需要耗費許多精力。舉例來說，在設計 wrapper 的 bytes_counter 時，我們自認為有仔細想好數到多少的時候要跳到下個 state，但後來跑 testbench 時出錯，回頭看 nWave 才好不容易發現原來我們多數了一次，導致多讀了 1 byte 的 data 而導致錯誤。

另外，我們原本有打算嘗試 bonus 的「不 reset 就讀進下一組鑰匙進行解密」，但不知道該怎麼直接判斷接下來的這組資料到底是 data 還是新的 key。因此有想到也許可以利用 fpga 上面的開關，來表示「每讀幾組 data 就會自動換成讀 key

」，這樣雖然就不需要按 reset，但就會導致每次都只能讀同樣大小的 data set，使用上也會有所限制 (最後也就沒有實際 implement 這部分)。