Dclab Lab2 Report

Team05

- 層級架構

  DE2_115

  |- **my_qsys** (rsa_qsys)

  |- **rsa_wrapper_0** (Rsa256Wrapper)

  |- **rsa256_core** (Rsa256Core)

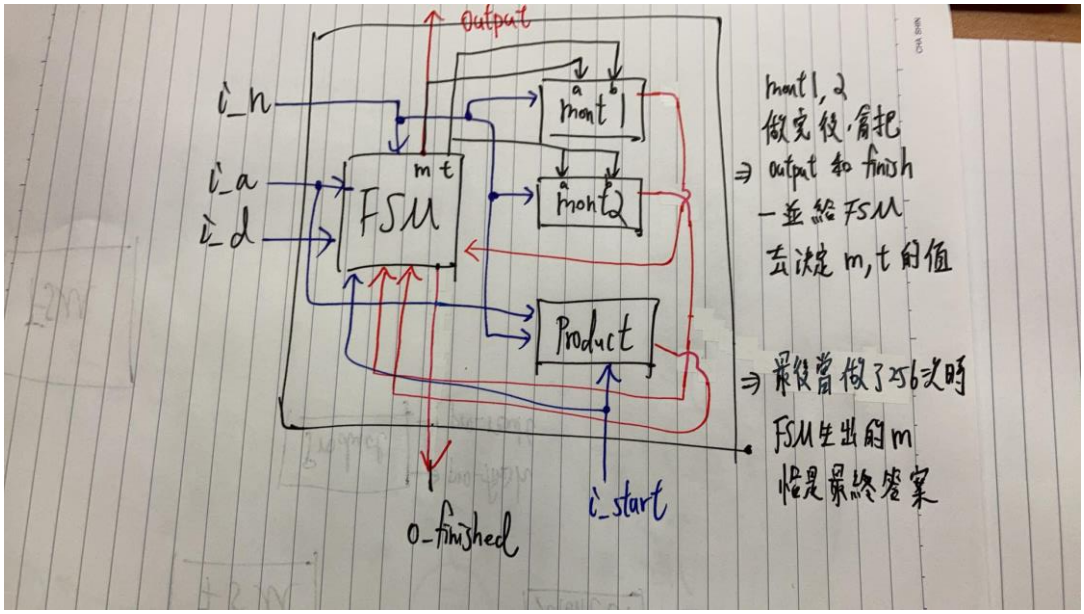
  |- **Product** (PRODUCT)

  |- **Mont_1** (Montgomery)

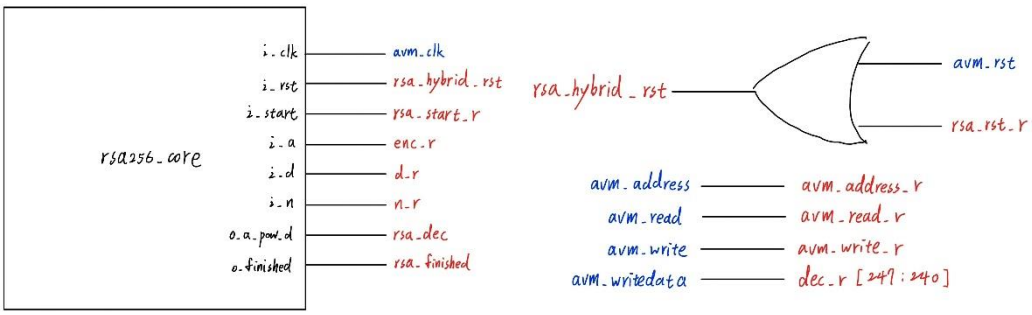
  |- **Mont_2** (Montgomery)

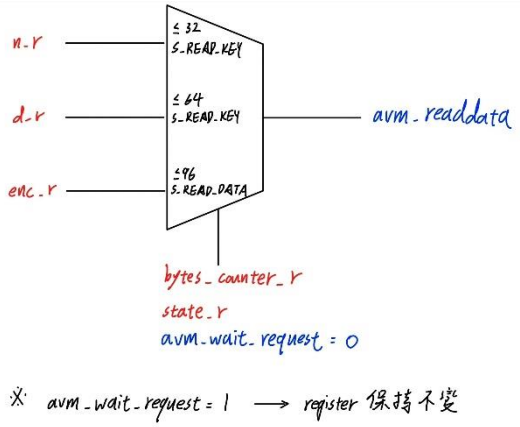
- Block Diagrams

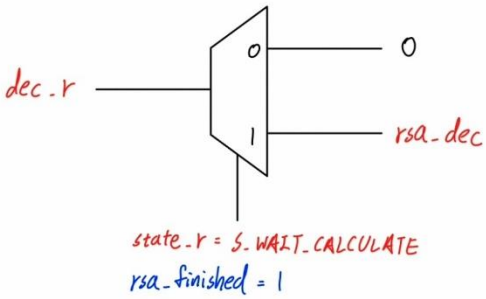
  - Rsa256Core

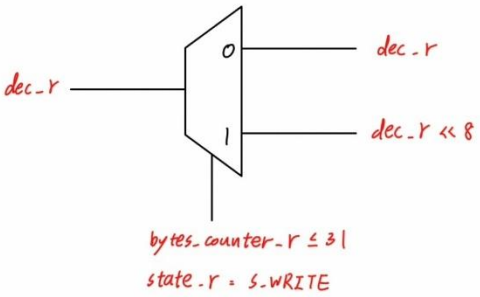


  - Rsa256Wrapper

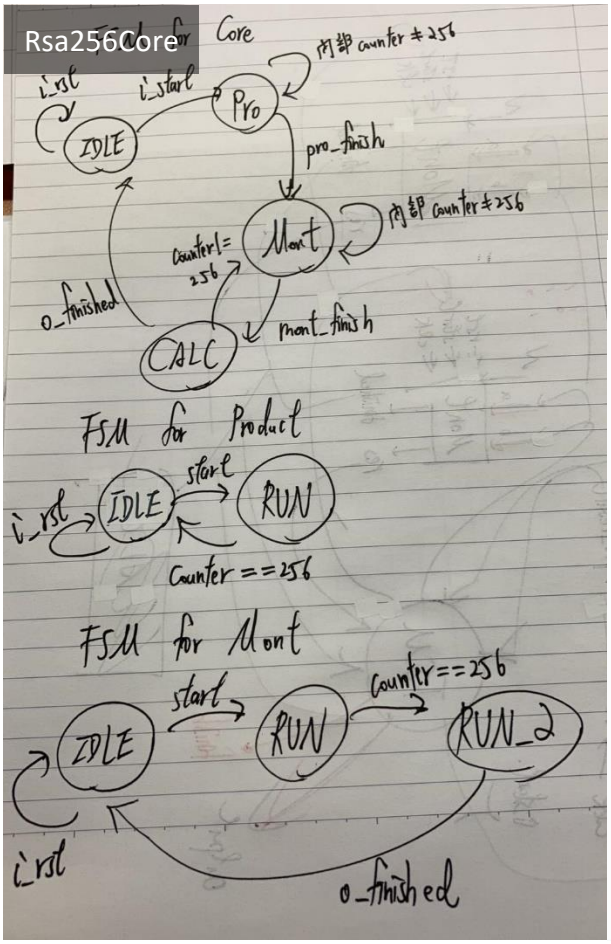


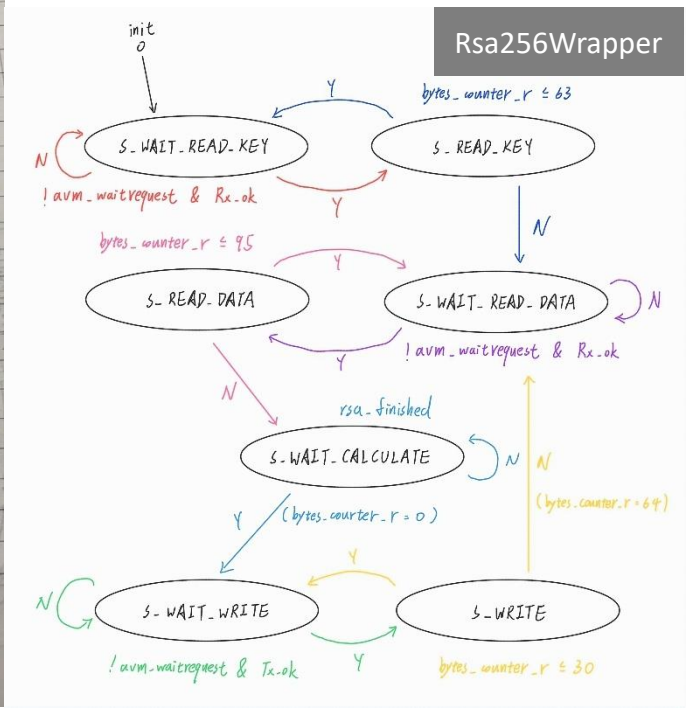




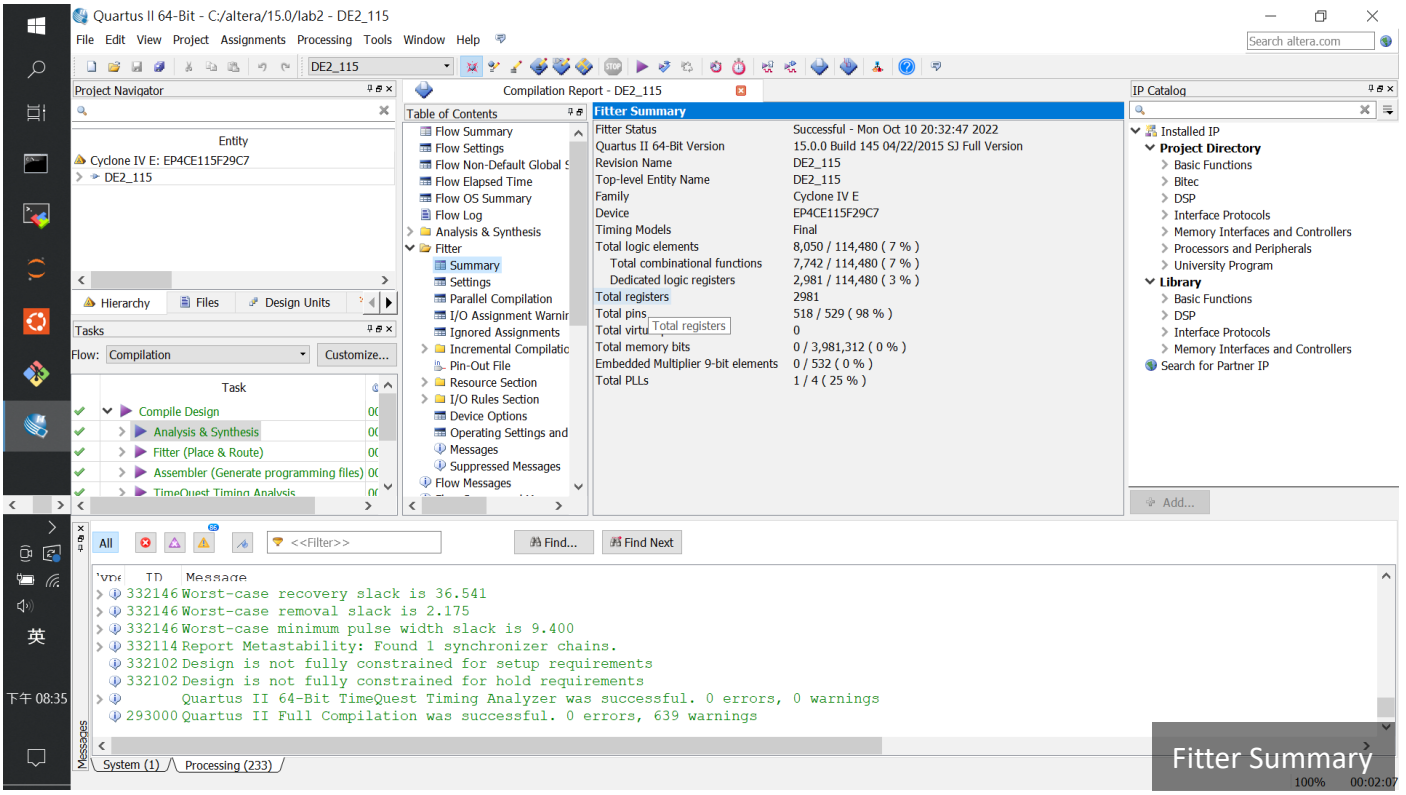


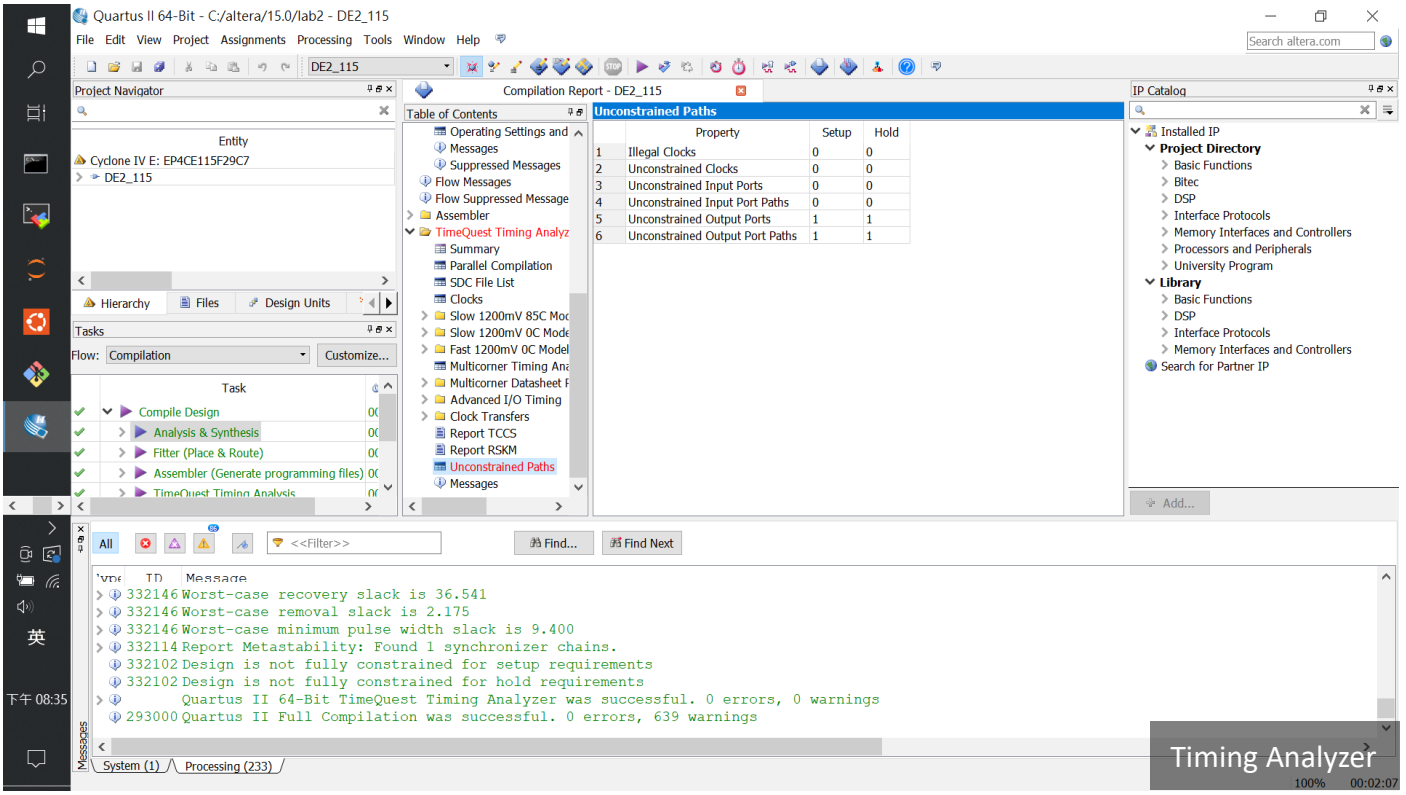
- FSM





- Fitter Summary & Timing Analyzer





- 遇到的問題與解法
  - Rsa256Wrapper
    1. 改成 7 個 States:

```
// States
localparam S_WAIT_READ_KEY = 0;  // query Rx (key)
localparam S_WAIT_READ_DATA = 1;  // query Rx (encrypted data)
localparam S_READ_KEY = 2;  // Read key (n, d)
localparam S_READ_DATA = 3;  // Read encrypted data (enc)
localparam S_WAIT_CALCULATE = 4;  // Calculate
localparam S_WAIT_WRITE = 5;  // query Tx
localparam S_WRITE = 6;  // Write
```

2. 為了實現 continuous decoding，write data 完都要 reset Rsa256Core

- 解法：增加 rsa_rst_r, rsa_rst_w 從 Rsa256Wrapper 內部和 Rsa256Wrapper 外部的 avm_rst 共同控制 Rsa256Core 的 i_rst 訊號，assign rsa_hybrid_rs = rsa_rst | avm_rst 作為 i_rst 的輸入訊號。

```
// .i_rst(rsa_hybrid_rst)
logic rsa_hybrid_rst;
assign rsa_hybrid_rst = rsa_rst_r | avm_rst;
```

- Rsa256Core
  1. 把 i_rst 當成 i_rst_n，導致寫錯。
  2. 在計算 m 的時候有可能會 overflow，debug 很久才發現這個問題，後來在每個 loop2 都多判斷一次有沒有大於 n 就不會了。

- 心得
  - 施伯儒：我負責的是 core 的部分，經過這次 lab 我對如何把 software 轉成 hardware 的 coding 寫法越來越熟能生巧。
  - 廖昶翔：我原本是跟吳宣逸一起負責 wrapper 的部分，但我想練習自己寫 verilog 的能力所以兩個 module 都寫了，然後 debug 用很久，還有詢問其他人該怎麼改進，最後自己寫出來的也可以成功跑完，滿有成就感的。
  - 吳宣逸：我這次和廖昶翔共同負責 wrapper 的部分，我遇到最大的問題是看到模板只給 4 個 state 時就慌了，寫出來邏輯一堆錯誤，最後改到共有 7 個 state 才漸漸進入狀況。另一個阻礙模擬的原因是我用 VScode 的 SSH 擴充套件連 workstation，不知道為何每次都需要跑超級久才能模擬出來，但 MobaXterm 就特別快，假如一開始有發現就會節省很多時間了，wrapper 從開始到完成整整花了約 15hr，希望之後開發能更有效率。