

LAB 2 Report

Team 4

B08901006 蔡亞辰 / B08901074 尤韻嘉 / B08901174 郭尚睿

Hierarchy

→ DE2_115.sv

→ rsa_qsys.v

→ Rsa256Wrapper.sv

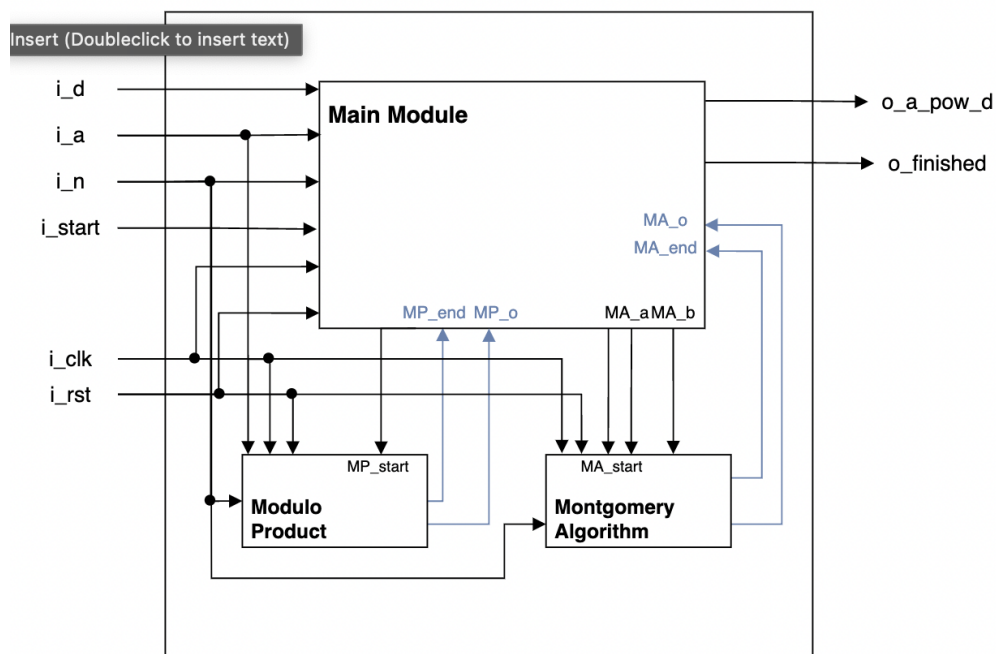
→ Rsa256Core.sv

→ ModProd (modulo product)

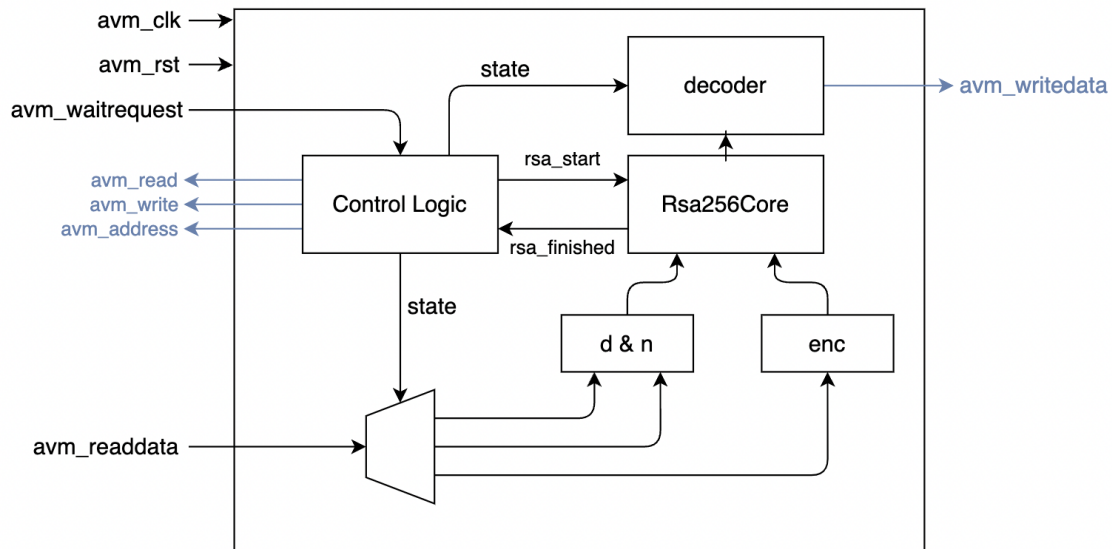
→ MontAlg (montgomery algorithm)

Block Diagram

[Core]

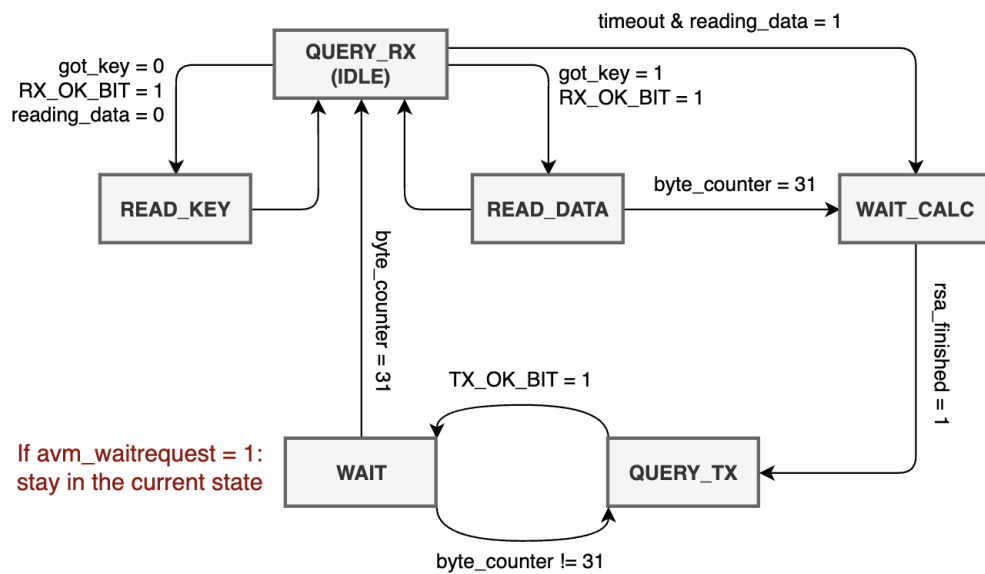


[Wrapper]

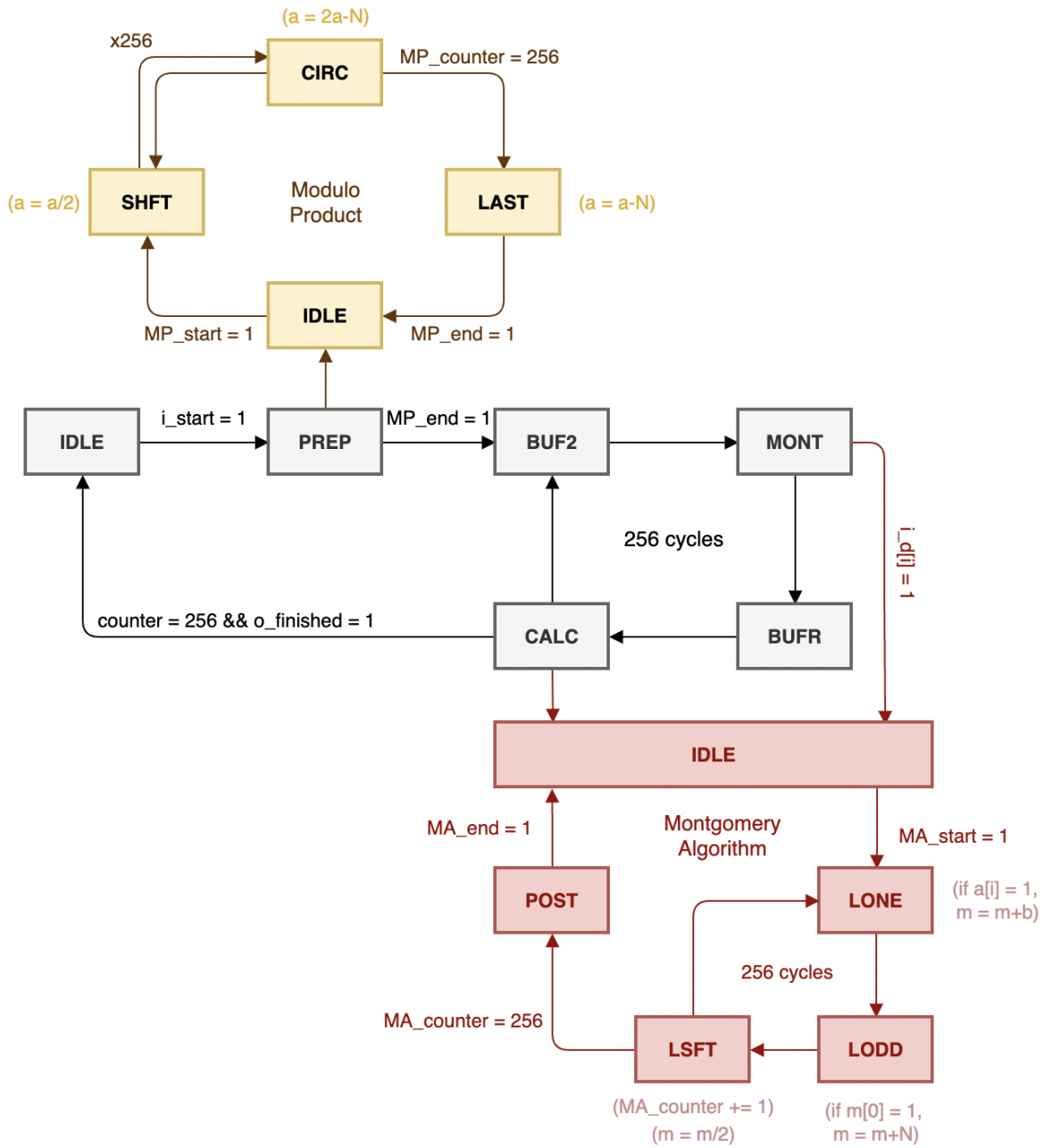


FSM

[Wrapper]



[Core]



Fitter Summary

The screenshot displays the 'Fitter Summary' window in Quartus II. The window has a title bar with four tabs: 'Rsa256Wrapper.sv', 'rsa_qsys.v', 'DE2_115.sv', and 'Compilation Report - DE2_115'. The main area is divided into two panes. The left pane, titled 'Table of Contents', shows a tree view of the compilation process, with 'Fitter' expanded and 'Summary' selected. The right pane, titled 'Fitter Summary', displays a table of compilation results.

Fitter Summary	
Fitter Status	Successful - Thu Oct 13 21:19:20 2022
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version
Revision Name	DE2_115
Top-level Entity Name	DE2_115
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	6,737 / 114,480 (6 %)
Total combinational functions	6,696 / 114,480 (6 %)
Dedicated logic registers	3,334 / 114,480 (3 %)
Total registers	3334
Total pins	518 / 529 (98 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	1 / 4 (25 %)

Timing Analyzer

Unconstrained Paths			
	Property	Setup	Hold
1	Illegal Clocks	0	0
2	Unconstrained Clocks	1	1
3	Unconstrained Input Ports	0	0
4	Unconstrained Input Port Paths	0	0
5	Unconstrained Output Ports	1	1
6	Unconstrained Output Port Paths	1	1

Difficulties & Solutions

- [Core] 在MONT state時把MA_start拉起來等於1，但整段時間MA_start都會等於1(理想上是想要他等於1一個clk cycle後就掉下來)；這會導致MA_start = 1延續到整個CALC state，Montgomery Algorithm input (a, b) 還沒assign好，MA的小module就會開始跑，而導致錯誤的計算結果：
 - 在MONT跟CALC前各加一個buffer state，準備input以及使MA_start短暫變成0
- [Core] 計算時發現overflow：
 - 把output 以及計算時用到的變數開到270 bits
- [Core] 把output開到270 bits 會不容易跟wrapper對接(應該要是256 bits)：
 - 因為雖然在計算過程中可能overflow，但最終的output應該不會overflow，所以我們另開一個270-bit register (o_a_pow_d_r)，並assign final_output =
o_a_pow_d_r [255:0]
- [Core] 一開始想跟Lab1一樣，只靠counter跟FSM實作，但這樣Code變得很複雜也難以Debug：
 - 所以改以把各個function實作成小module，靠i_start跟o_finish跟大module溝通

- [Wrapper] 如果收到的data長度不是256 bytes的整數的話，在跑python時會卡在FSM中，無法回到end state 或IDLE state：
 - 加一個判斷式，若大於一段時間沒有收到下一筆data的話，就自動回到IDLE

Bonus!