國立台灣海洋大學資訊工程學系專題報告

題目

Keypro

作者

00857035　戴芷柔 taijustina0807@gmail.com

00857047　邱莘瑜 sheenachiu.pub@gmail.com

報告編號:NTOUCSE 111 年度 – 競賽組第 29 組

指導教授：嚴茂旭 博士

中華民國 111 年 12 月 05 日

目錄

專題分工及貢獻度說明

專案管理系統

摘要

簡介

理論推導

架構與演算法則

模組設計描述

實驗結果

討論

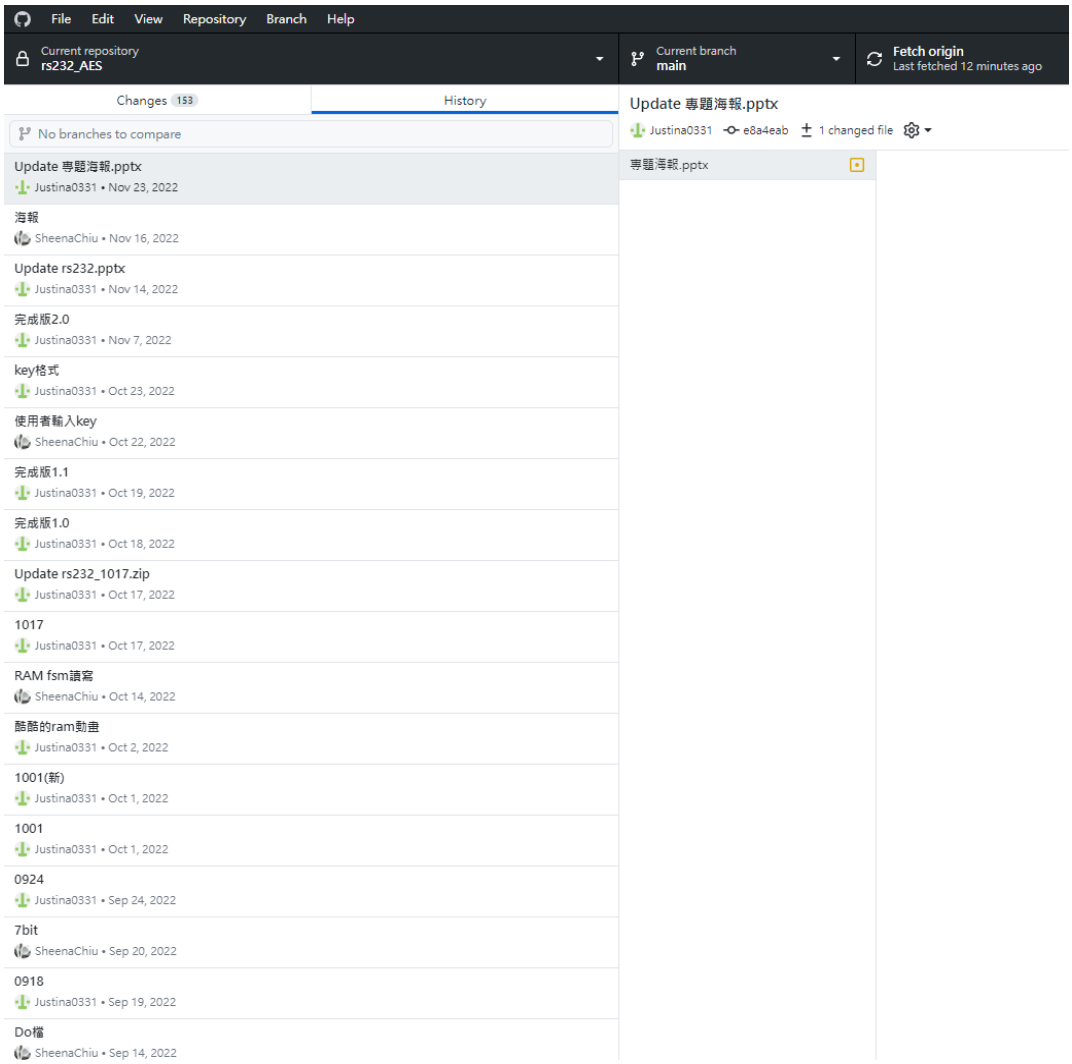
結論

相關資源

專題分工及貢獻度說明

編號	姓名	主要工作內容	專題貢獻度(100%)
1	戴芷柔	rs232 傳接收控制，結合 aes 加密實作，上機燒版測試，報告撰寫。	55%
2	邱莘瑜	ram 資料存取，金鑰輸入，test bench 撰寫，海報設計，do 檔撰寫。	45%

專案管理系統截圖



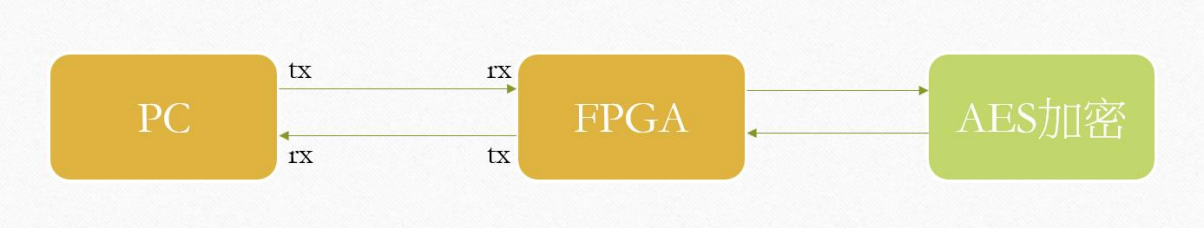
摘要:

目標為可傳輸加解密資料的 FPGA 設計，使用 verilog 實現。傳輸規格為 RS232，加密技術使用 AES。可透過 verilog 程式實現 PC 傳送的資料到 DE0 板子後，進行 AES 加密及解密，使用的 AES 模式為 ECB 模式，加解密完成的資料亦可回傳到 PC 端，PC 可透過 RS232 軟體進行測試結果。

一、 簡介:

使用 verilog 實現 AES 加密的程式，在 github 上已經有人撰寫出來，過去也已經有前人使用過 SPI 的傳輸規格撰寫類似項目，因此這項專題希望能藉由不同的傳輸規格去做到 FPGA 的加密設計，專題主要為專研 rs232 的傳輸規格，以及理解 AES 加密規則並使用。

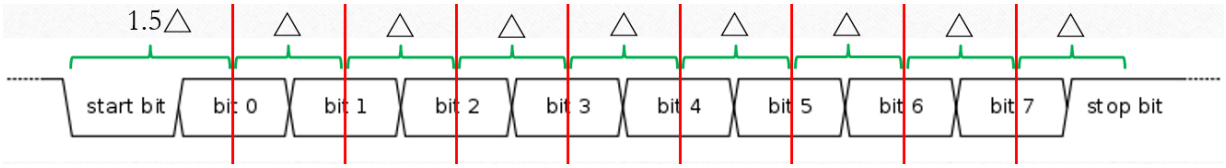
二、 理論推導:



- 首先要了解 rs232 的傳輸規格，rs232 傳輸線是一種 UART，UART 主要 pin 腳為 RX、TX、GND 及 VCC，VCC 為供電用 pin，在專題中沒有使用，只有連接前三個 pin 腳，RX 控制接收，TX 控制傳輸，GND 則是接地線，RS232 傳輸的資料均由邏輯 0 跟 1 組成，在訊息尚未開始傳送時，不管是 RX 或 TX 均是停留在高電平等待，當要開始傳送或接收時，訊號將從高電平掉到低電平，以表示開始傳接收訊息。

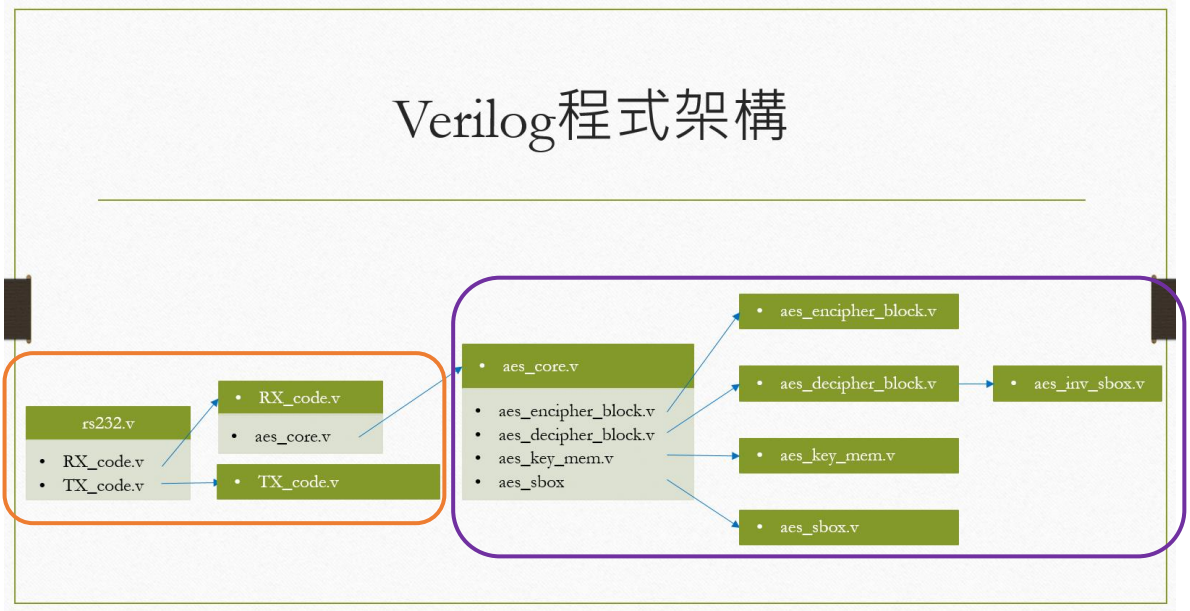
- 在這過程中會需要處理 rs232 傳輸頻率與 DE0 板子接收訊息頻率的問題，需要藉由 DE0 板子使用的 clock 及 rs232 測試軟體傳輸速率去計算並使兩者傳接收可以同步，DE0 板子使用的 clock 為 50MHz，rs232 測試軟體選用 baud rate 19200，DE0 的 clock 會比 rs232 傳輸速度快很多，因此在撰寫程式時，需要計算每過多少 clock 才需要接收一次 rs232 訊息，而這個 clock 的值在這邊用△代稱。
- △ 的計算方式：

  △ = 資料傳輸 1bit 的時間 = 1/19200 ≈ 52.08 μs，DE0 傳輸 1 clock 的時間 = 1/50MHz = 0.02 μs ，故每△ = 52.08/0.02 ≈ 2600 個 clock。
- 當一接收到訊號 0，便開始計數經過的 clock，為了更精確地接收訊息，因此避免在訊息傳送來的瞬間接收，而是在中心點接收，如下圖的紅線位置，在接收 bit 0 時需計數 1.5 個△才會是接收到中心線的資料，後續每接收一個資料均是計數 1 個△。



三、 架構與演算法則:

- Verilog 程式架構說明：這項專案主要撰寫橘色框內 rs232.v、RX_code.v、TX_code.v 三組程式碼，這三組程式碼控制了 rs232 傳接收資訊及資料存取，而紫色框內為 AES 架構，使用secworks的 source code，只需要詳讀 aes_code.v 規格後便可方便的使用 aes 加解密，此程式碼有支援 128 bits 及 256 bits 兩種金鑰模式。



- RX 接收格式：

資料一次接收 8 組 bytes，以下介紹每組 byte 的作用，由左向右開始接收，如下圖所示，此外注意 addr 及讀寫共用一組 byte。

功能	結束	check byte	data	data	data	data	addr	讀寫	開始
bit	8	8	8	8	8	8	7	1	8
說明	03 結束	check byte (未使用)	為了避免 02 及 03 資料無法輸入，因此第 1 個 bit 不使用，ex:若需要輸入 02 資料則輸入 82(1000 0010)來代替。				ram 128*32	1 寫 0 讀	02 開始

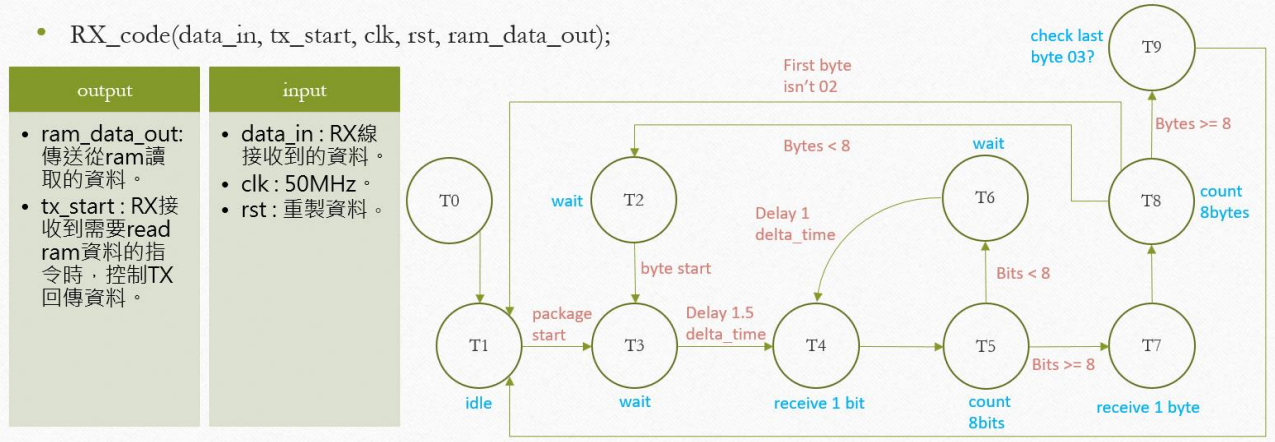
- TX 傳送檔案格式:

功能	data	data	data	data
bit	8	8	8	8
說明	直接輸出從 RAM 選取位置的資料，寬度為 32bits，但每個 byte 的第 1 個 bit 資料不使用。			

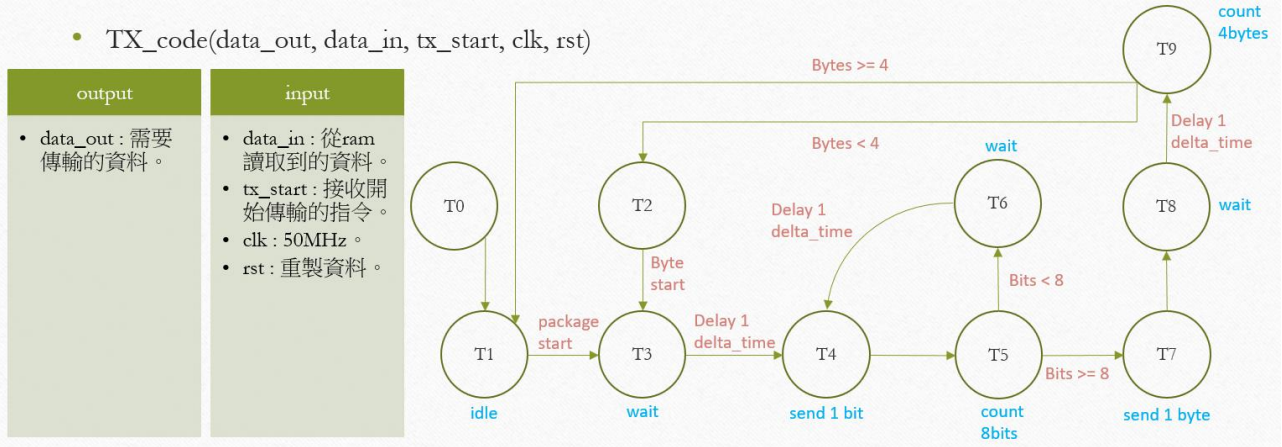
四、 模組設計描述:

- 設計環境：
  - Altera Tool：ModelSim-Altera 10.1d (Quartus II 13.1)
  - Quartus Tool：Quartus II 13.1 (64-bit) Web Edition
  - FPGA Device：Cyclone III (EP3C16F484C6)
  - AES S-box：參考secworks
- rs232.v：為模組的 top 檔案，括號內接連接 FPGA 的 pin 腳。

- RX_code.v：FPGA 接收資訊的主要處理檔案，右半部為 FSM。

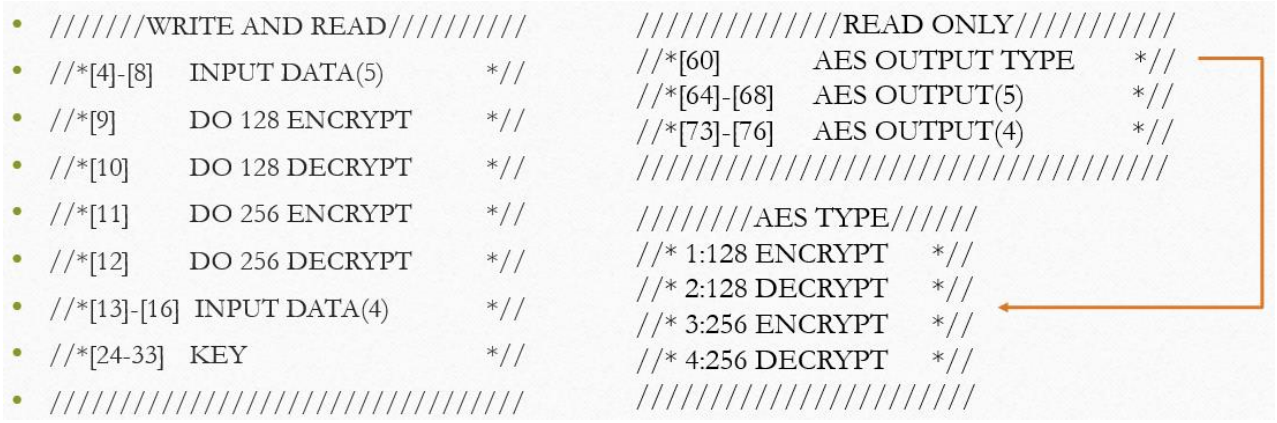


- TX_code.v：FPGA 傳送資訊的主要處理檔案，右半部為 FSM。



- aes_core.v：AES 加解密的主要檔案，各個接腳用途說明如下。

- RAM 資料存取格式及 AES 接收格式說明：RAM 分配了 7bits 因此有 128 個位址，以下為有使用到的位址分別功能。

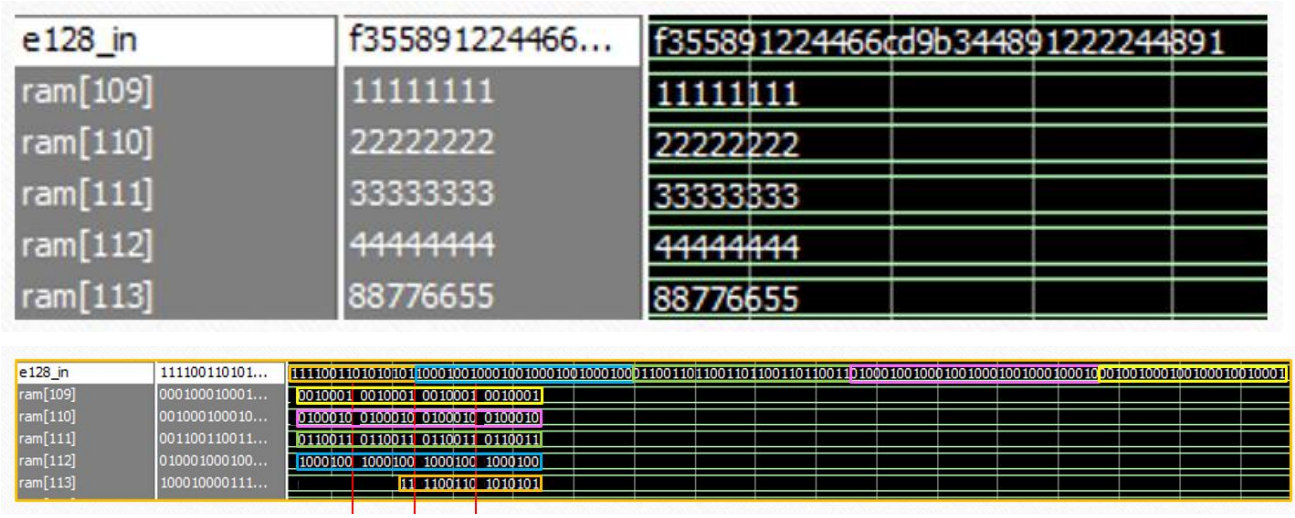


```
//////WRITE AND READ/////////
//*[4]-[8]    INPUT DATA(5)          *//
//*[9]        DO 128 ENCRYPT         *//
//*[10]       DO 128 DECRYPT         *//
//*[11]       DO 256 ENCRYPT         *//
//*[12]       DO 256 DECRYPT         *//
//*[13]-[16]  INPUT DATA(4)          *//
//*[24-33]    KEY                    *//
////////////////////////////////////

//////////////READ ONLY//////////
//*[60]         AES OUTPUT TYPE     *//
//*[64]-[68]    AES OUTPUT(5)       *//
//*[73]-[76]    AES OUTPUT(4)       *//
////////////////////////////////////

////////AES TYPE//////
//* 1:128 ENCRYPT    *//
//* 2:128 DECRYPT    *//
//* 3:256 ENCRYPT    *//
//* 4:256 DECRYPT    *//
/////////////////////
```

- [4]-[8]：每個 ram 存取一次 rx 接收資料，包含 4 組 data，也就是 32 bits，由於每組 data 的第 1 個 bit 不使用，因此每個 ram 實際使用到資料只有 28 bits，而 AES 一次加解密資料需要使用 128 bits，128/28 = 4...16，因此需要 5 組 ram 資料做一次 AES 加解密，其中四筆資料取 28 bits，而其中一筆只取 16bits，這五組待加解密資料存在[4]-[8]。
- [9]-[12]:而當 RX 接收到的 addr 為[9]-[12]，執行上述的加解密指令。
- [13]-[16]:由於[4]-[8]是為了排除接收問題而存取的格式，在檢查 AES 加解密是否正確時相當不值觀，因此在[13]-[16]的位址另外存放扣除掉每個 data 的 1 個 bit 之後合併的資料。
- [24]-[33]：存放接收的 key，當 key 正確，[9]-[12]指令才可以執行。
- [60]：顯示 AES 的輸出是什麼模式，各個模式如下圖 AES TYPE。
- [64]-[68]：執行完 AES 的結果，同上述[4]-[8]，若需要將資料重新丟回 input 執行，為了符合 5bytes 的輸入格式因此提供 5bytes 的輸出。

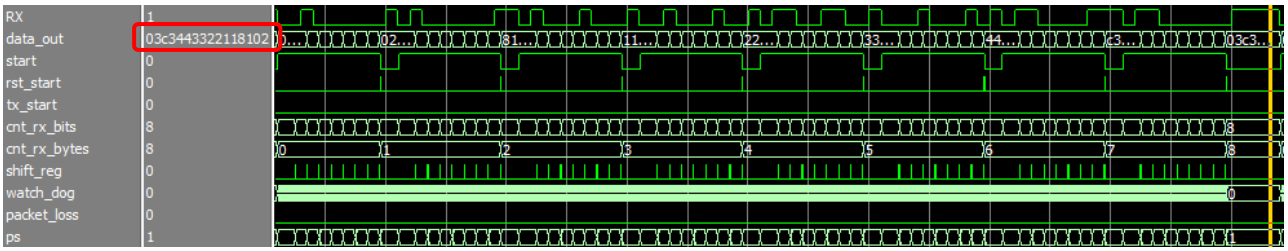
- [73]-[76]：執行完 AES 的結果，同上述[13]-[16]原因，可供直觀的方式查看加解密原始結果。

- 下圖為[4]-[8]傳送給 aes 的輸入範例圖，由於截圖當時版本不同，這裡請將[109]-[113]當作[4]-[8]位置理解即可，第一張圖由十六進制顯示，第二章圖由二進制顯示。



五．實驗結果:

- RX 接收模擬結果：

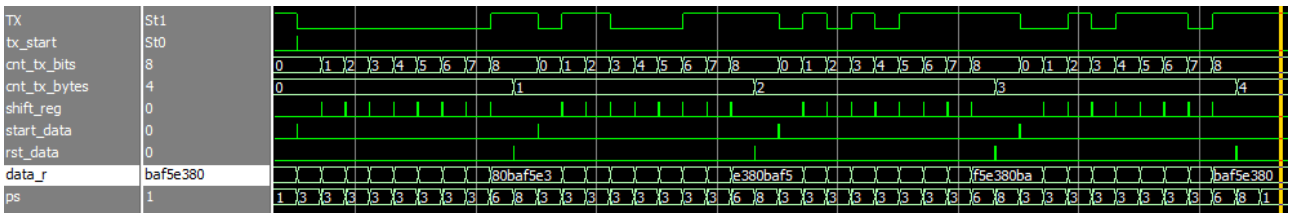


此筆測試資料為

結束	x	data	write/read	addr	開始
03	c3	44332211	1(write)	1	02

- RX：一開始為1，到0開始接收資料。
- start：開始接收一個 package。
- rst_start：結束接收一個 package。
- tx_start：若接收到 read 指令，則令其變1，使 tx 開始傳送資料。
- cnt：用來記錄目前接收了幾筆 bit、幾筆 byte。
- shift_reg：接收 rx 資料並存到 data_out。
- watch_dog：在02開始後跟著開始計數，若接收 package 的時間超出預期，則結束 package 接收並將接收一半的資料清除，回到 idle 狀態。

● TX 傳送模擬結果：



此筆測試資料為

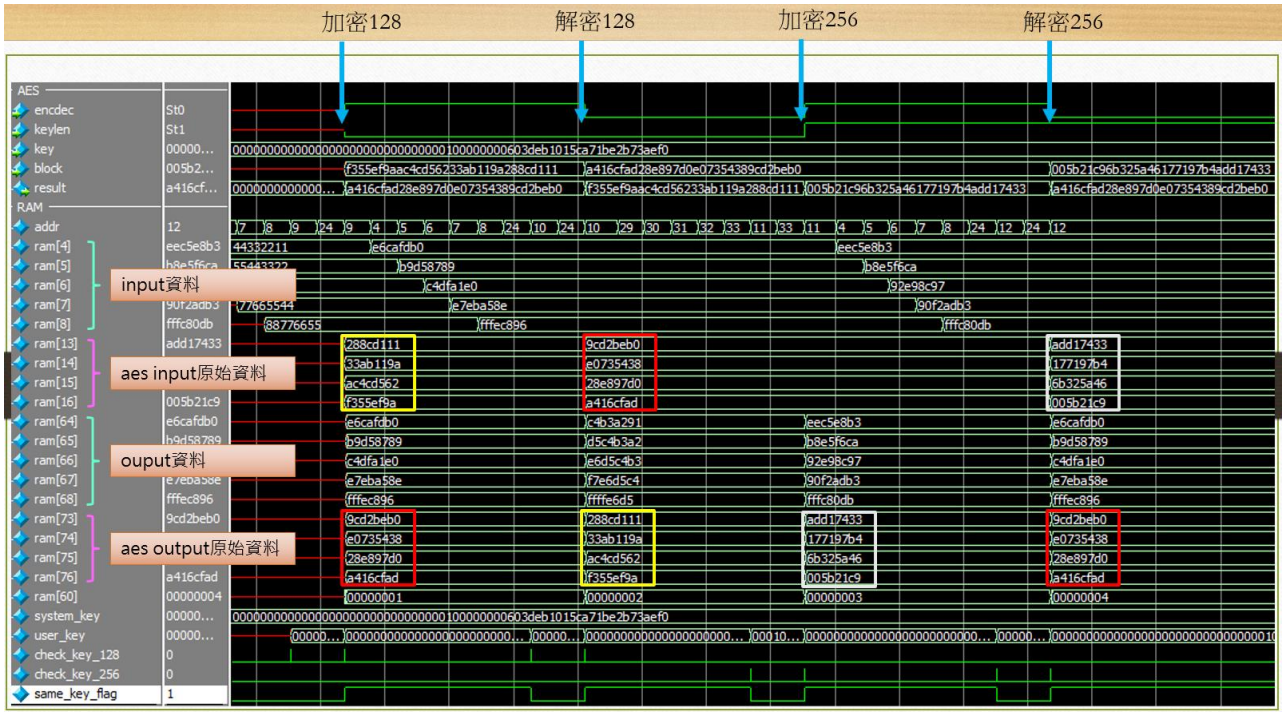
data	baf5e380

- TX ：一開始為1，掉到0開始傳送資料。
- tx_start：判斷一個 package 開始傳送。
- cnt：用來記錄目前接收了幾筆 bit、幾筆 byte。
- shift_reg：傳送 tx 資料。
- start_data：開始傳送一個 package。
- rst_data：結束傳送一個 package。

- data_r：需要傳送的資料，tx 每次都將傳送 data_r[0]的位置資料，因此 data_r 的資料會不斷向右 shift。

- 整套模組包含 AES 加密模擬結果：

  aes output 原始資料必須等於 aes input 原始資料，亦為加解密正確。



- 使用 RS232 測試軟體測試結果，這裡使用 128bits key：

六、討論:

- 優點：在 RS232 傳送資料時，多了一種可選擇的加密方式，使用規則簡單，資料輸出直觀明確，加解密需透過 FPGA 設備進行，FPGA 本身就是一層鑰匙，可以更有效的保護資料。
- 缺點：本項專案在 RX 接收資料的部分，check bit 未使用到，因此在大量資料連續傳送時，有可能會有資料接收錯誤卻未發現的情況。
- 改進方向：

  1. 添加 check bit 使用，以確保資料正確性。
  2. 目前 key 使用為預設 key，希望能添加令使用者更改自訂 key。
  3. baud rate 目前預設為 19200，可分配一組 ram 存取 baud rate，提供多種不同 baud rate 選擇，確保完整性。

- 未來發展：前項專案制做得還不夠成熟，在小型測試尚可使用，但未在大筆測資上進行實作，這項技術可有效的對資料傳輸進行加密，若能完善其正確性、完整性，相信未來會是個有用的技術。

七・結論:

- 本專題最終可令使用者自行輸入 128 或 256 bits 的 key 並比較是否與內建的 key 相同，以此達到實作簡易的 KEYPRO 功能，但還有許多地方值得改進。一方面是預設內建 key 啟用過一次後如何產生下一組隨機的內建 key ，以及更人性化一點的構想 : 如何實作可令使用者於加密時自行設定內建 key；另一方面則是本專題如何與生活中的應用結合，這些問題都值得再去探討。

八、相關資源：

aes 程式碼來源：https://github.com/secworks/aes

製作的 github 紀錄，目前為 private：

https://github.com/Justina0331/rs232_AES

UART 傳輸器說明：https://zh.wikipedia.org/zh-tw/UART

測試結果影片：https://www.youtube.com/watch?v=QLBHetRdFaM