Design of Chip Security - Spring 2024

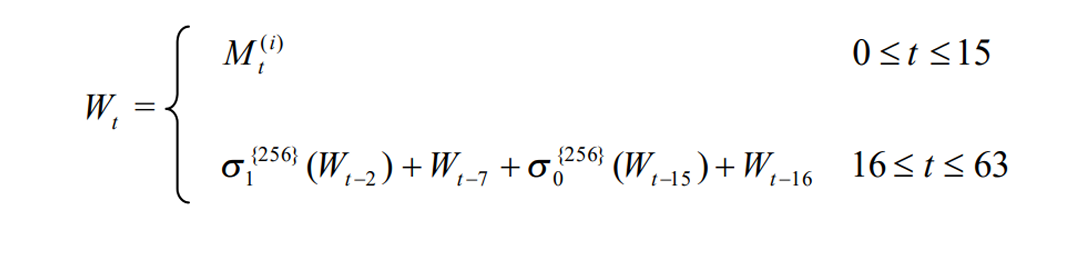
Lab01 Report

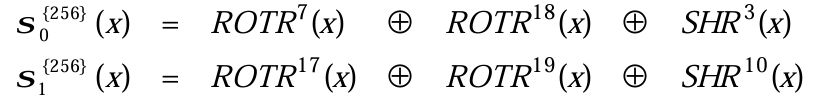
Name: 王品然

Student ID: 113063572

1. Design architecture and throughput
2. Algorithm

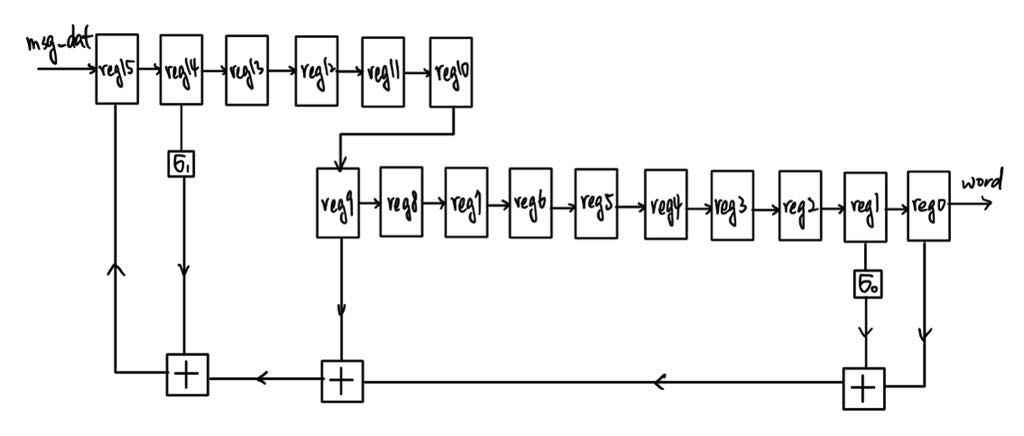
The “Message Schedule” algorithm in SHA2:



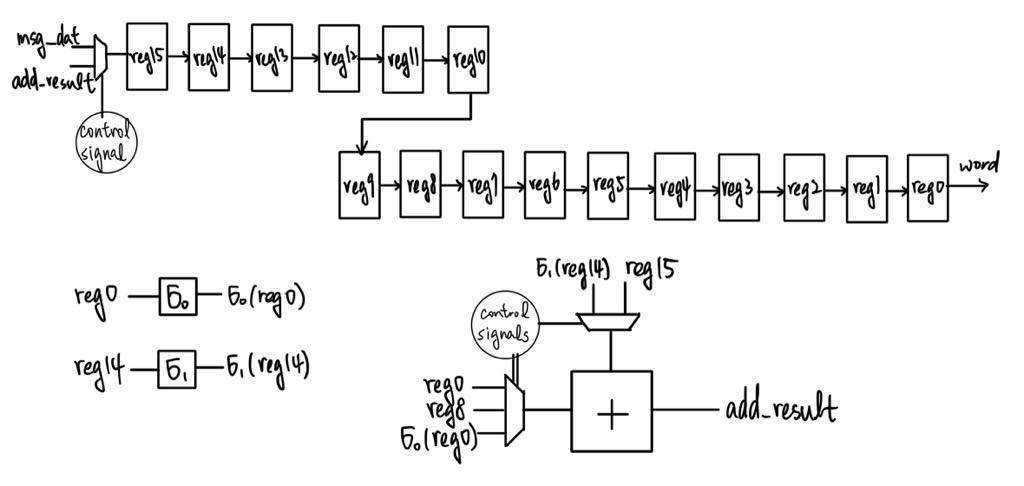


(note that “” stands for “” and “” stands for “”)

下圖的block diagram可以描述此演算法：



填入一個message block之後，即可一直從reg[0]得到message schedule的結果，結構簡單易懂且能夠一個clock cycle就能算完一個output值，但需要用到三個32-bit加法器，較佔面積，不符合這次想要小面積的要求，因此想辦法在register數量不變的情況下，只用一個32-bit加法器，如下：



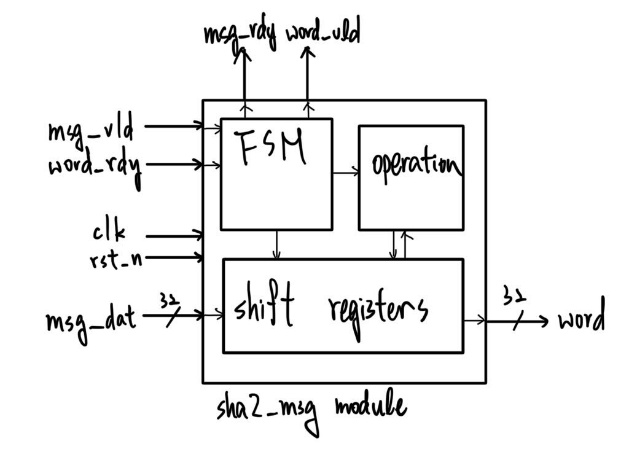
這樣簡化的block diagram以如下的步驟運算

* 1. 填入16個words(一個message block )到reg[0]~reg[15]，使得此時reg[n]暫存，也就是。
  2. 將reg[0]與相加(也就是)並存入reg[15]，同時shift reg[n+1] 到reg[n]，並且output reg[0](也就是)。
  3. 將reg[15](暫存著的結果)與reg[8](此時存著)相加並存入reg[15]。
  4. 將reg[15](暫存著與(此時reg[0]存著)相加得到也就是演算法的，並將此存入reg[15]。
  5. 繼續重複(2)到(4)62次，再回到(2)完成第64次output。

這樣便完成一個massage block的expanding。這樣的結構下的throughput會是1word per 3 cycle。接下來要加上flow control後，即可完成此演算法的電路實現。

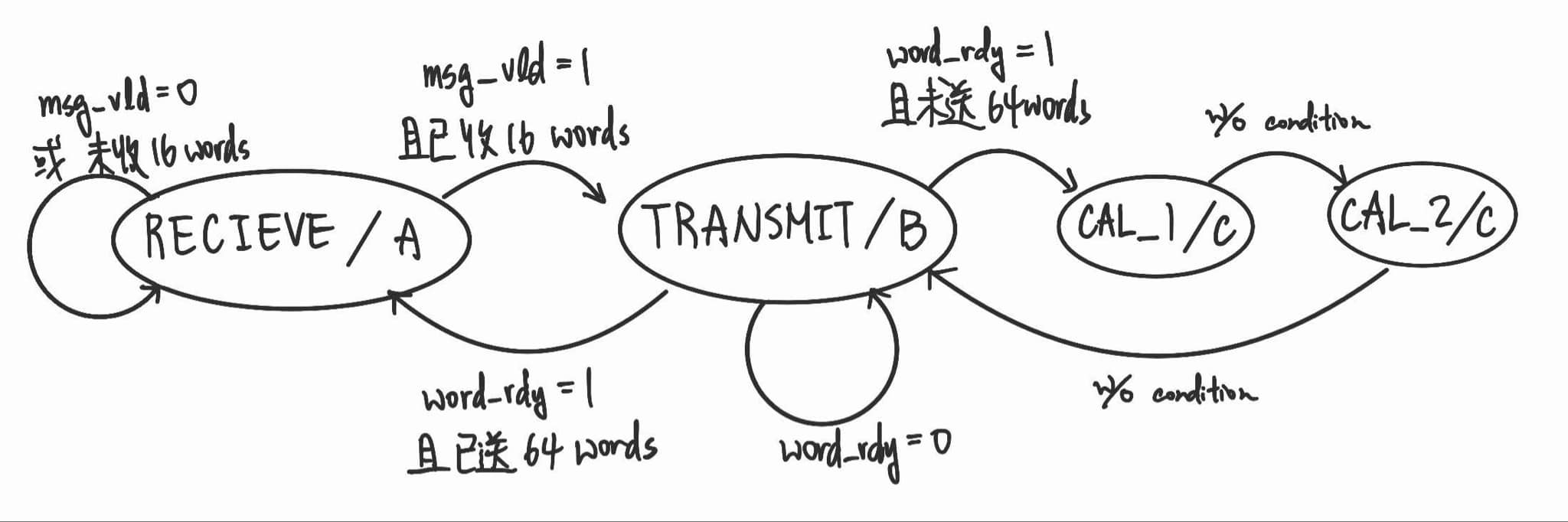
1. sha2\_msg module structure

以上述的block diagram可以初步建構出sha2\_msg的架構：



以FSM作為control unit，負責I/O protocol和運算的flow control。

1. FSM



以下解釋各個state時做的事(state transition還有output如上圖所示)

RECEIVE: 使msg\_rdy=1，在msg\_vld=1時，使reg[15]接收msg\_dat，同時shift register，並讓計算輸入了幾個word的counter+1。

TRANSMIT: 使word\_vld=1，在word\_rdy=1時，shift register，並讓計算輸出了幾個word的counter+1，同時運算reg[0]+，並存入reg[15]。

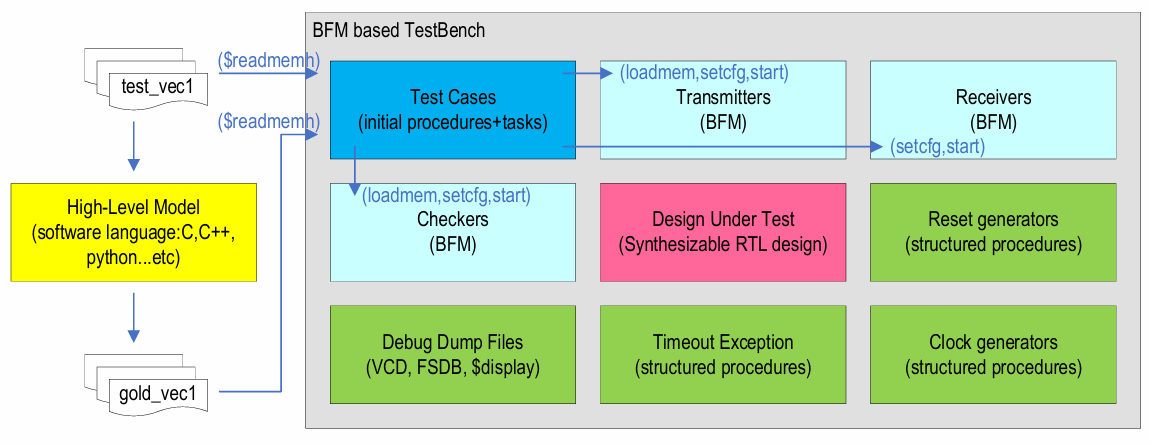
CAL\_1: 運算reg[15]+ reg[8]，並存入reg[15]。

CAL\_2: 運算reg[15]+ ，並存入reg[15]。

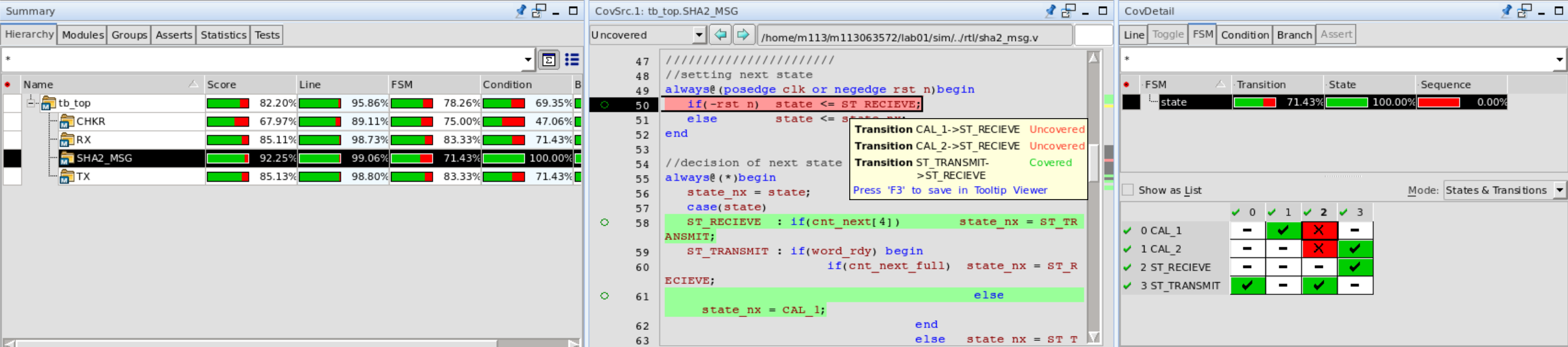
值得注意的一點是，因為這個架構下，事先input完一個block才要開始output，因此計算已input數量與計算已output數量的情況之時間不重疊，所以為了節省面積，可以使兩個功能共用一個6-bit counter，因此需要在RECEIVE state多做一件事，在counter\_next等於16時，將counter清零，才能交給output端記數。

1. Testbench

這次的testbench大多參考lecture 5的講義以及所附的例子，使用的是BFM的模式，架構如講義中的圖：



High level model的部分是使用python寫“Message Schedule” algorithm並且使其隨機生成指定長度的test vector與其對應之golden vector，輸出兩份.dat檔，在test case中讀取。驗證的情況包含msg\_vld和word\_rdy可能連續或不等間隔拉高，code coverage的部分，FSM transition缺了兩個，如下圖：



但其實uncovered的狀況不會發生，因為如lab01作業內容“rst\_n will be active at the beginning of the test.”所說，reset是在DUT還沒接收data時就來了，因此其實不會有CAL\_1或CAL\_2轉換到RECEIVE的狀況發生，除非reset在CAL\_1或CAL\_2時才來來。至於line和branch的coverage情況則是少了下圖中的那行default，因此也不重要。

