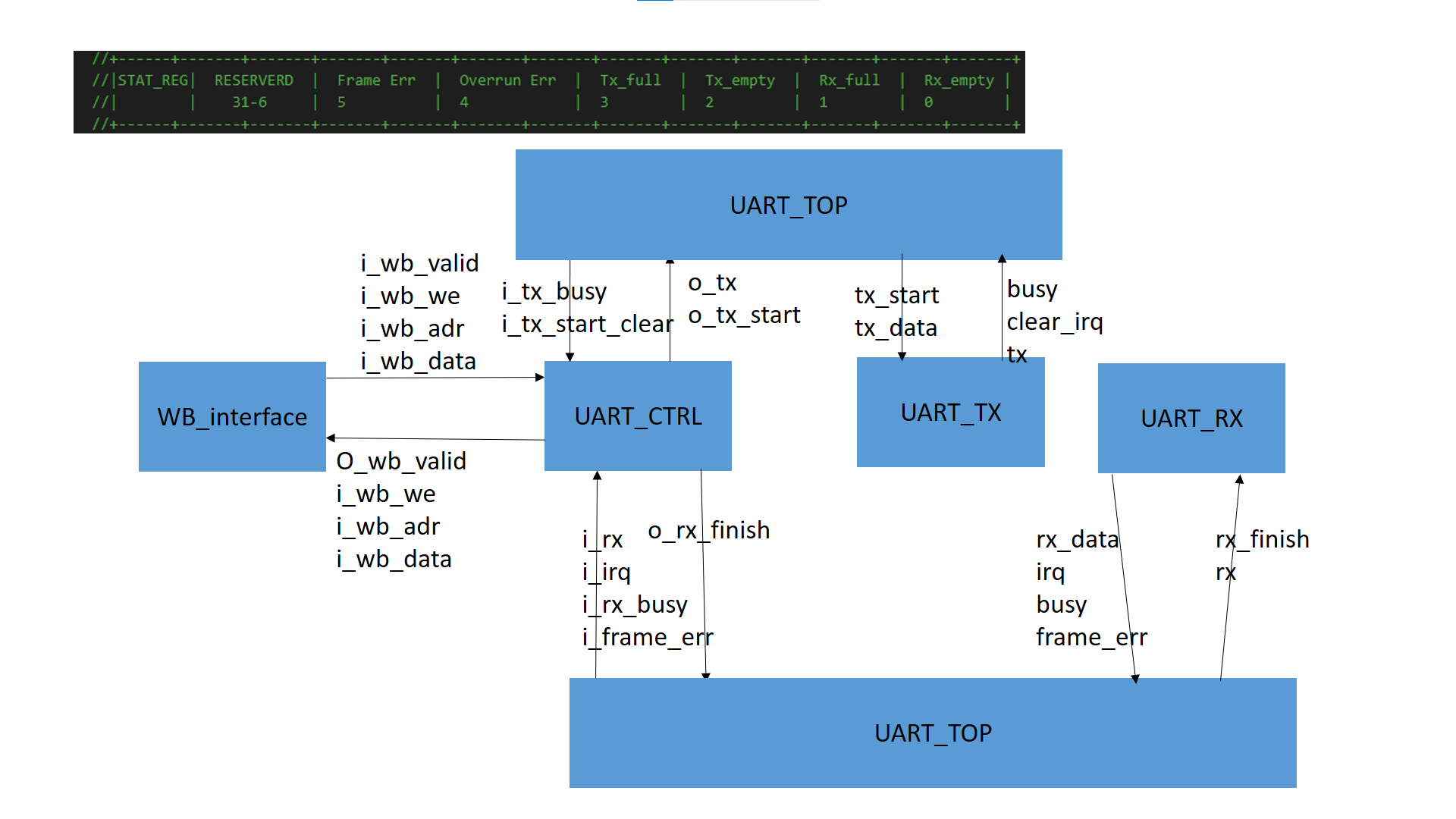
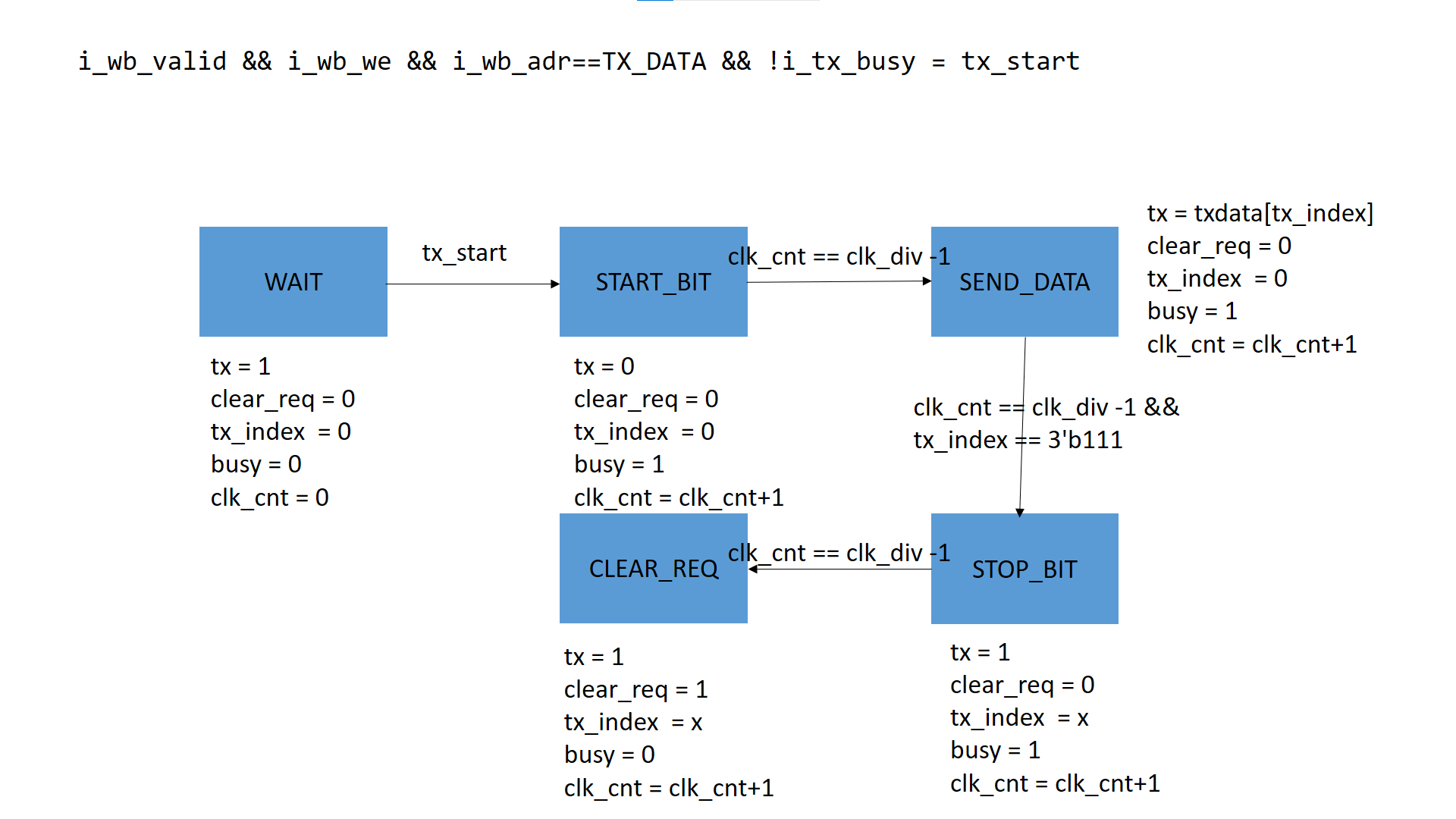
**SOC Lab6 report 第六組**

Block Diagram for UART：

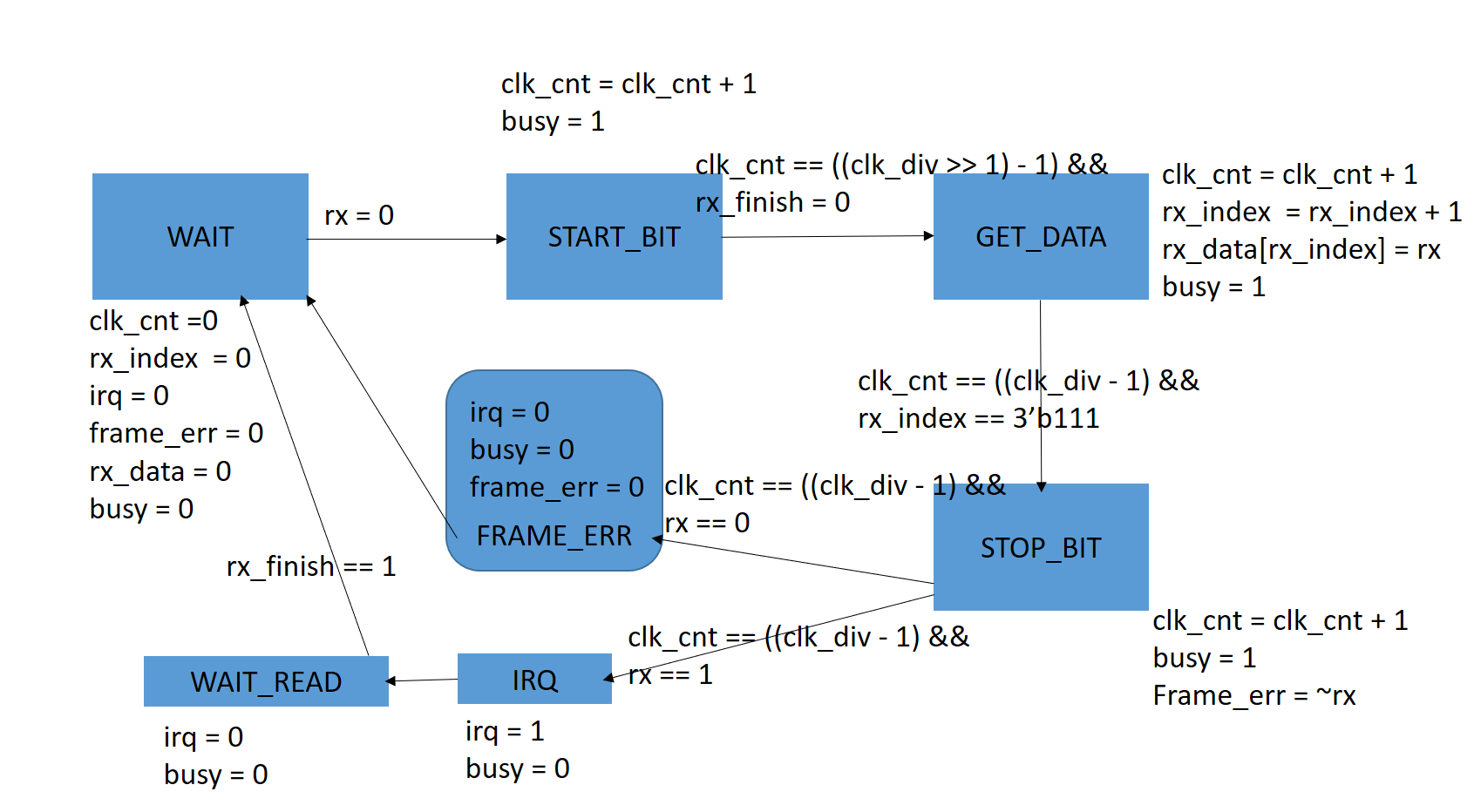


可以觀察到ctrl的輸出入剛好和tx、rx是相反方向的

FSM Diagram for uart\_tx：

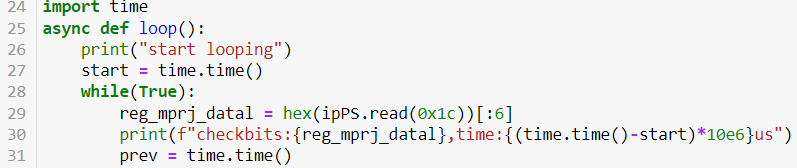


FSM Diagram for UART\_rx

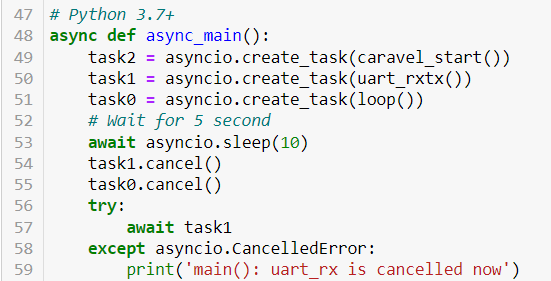


Method to verify our answer from notebook：

參考了github討論區#175的做法：

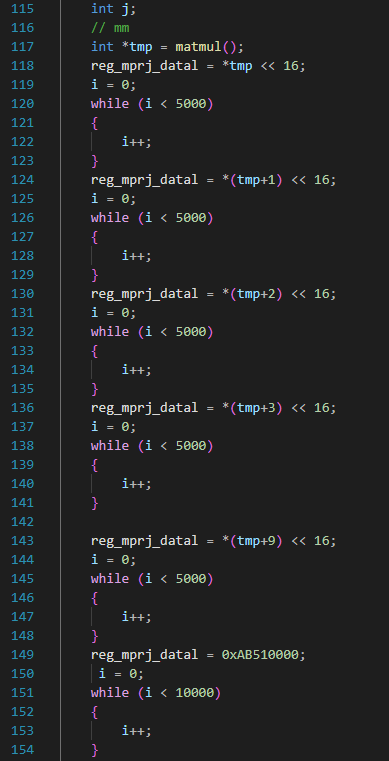


卡while loop去print checkbit和對應時間

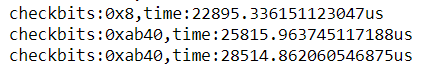


Top.c的部分參考同個討論區另一位同學的方法：

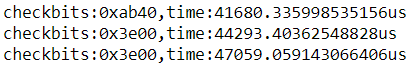
在每個reg\_mprj\_datal輸出的間隔塞while loop，使韌體的運行速度變慢，就能夠在ipynb上確認正確性。

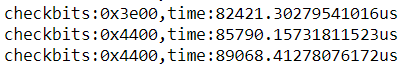


Firmware啟動：



mm開始：







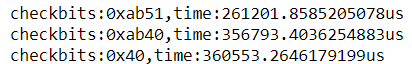




Mm結束：



Fir開始：



Fir結束：



這邊fir跟qs應該是同時並行的，所以中間有一些data和ab71可能被蓋掉了，每次重跑kernal的輸出都會有些不一樣，多跑幾次這個kernal就可以看到是ab71結尾的情況，也就是qs是最後完成的operation，符合top.c的behavior。

Qs開始：





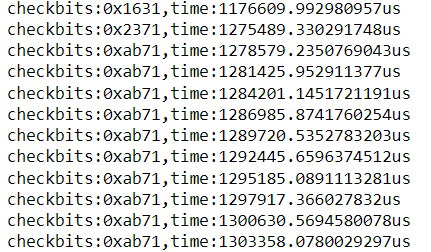




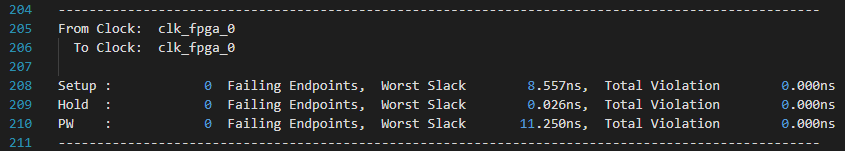


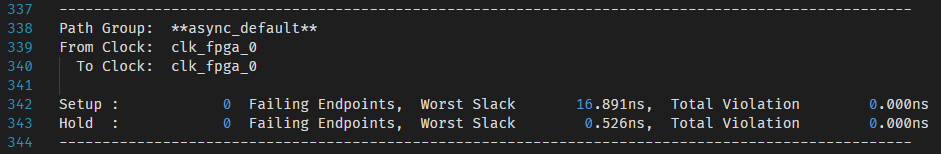


Qs結束：



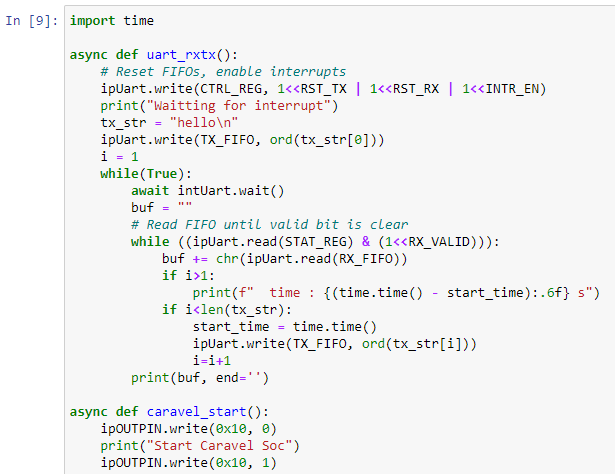
Timing report/ resource report after synthesis：

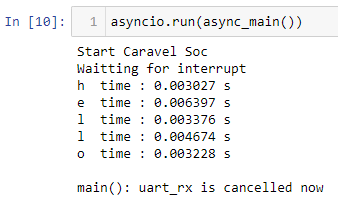




合成後的delay確實都有MET（slack > 0）

Latency for a character loop back using UART：





平均下來Latency為4.17 ms左右

Suggestion for improving latency for UART loop back：

1. Baud Rate：

在此次實驗中是採用40Mhz，或許我們可以嘗試看看更高的頻率。而較高的頻率通常具有較低的延遲。我們選擇發送器和接收器都可以支援且不會出現錯誤的頻率。

1. DMA：

改採用DMA而不使用UART的話，我們就不須透過processer發送address，可以獨立地直接讀寫memory，從而減少延遲。