# SOC Design hw2

108061207 陳伯丞

## Part1: AXI-Master Interface

1. VITIS

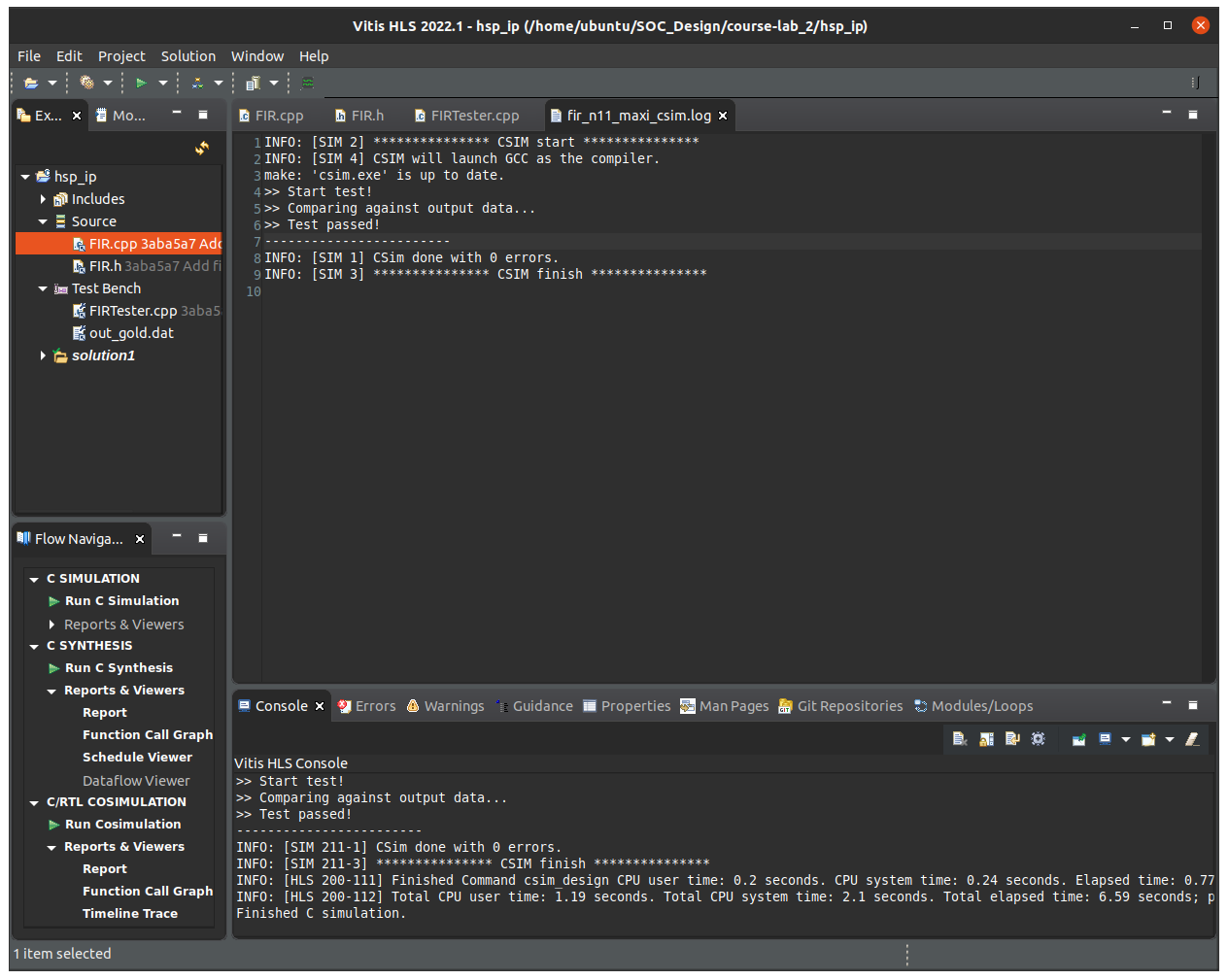


Figure1. C simulation

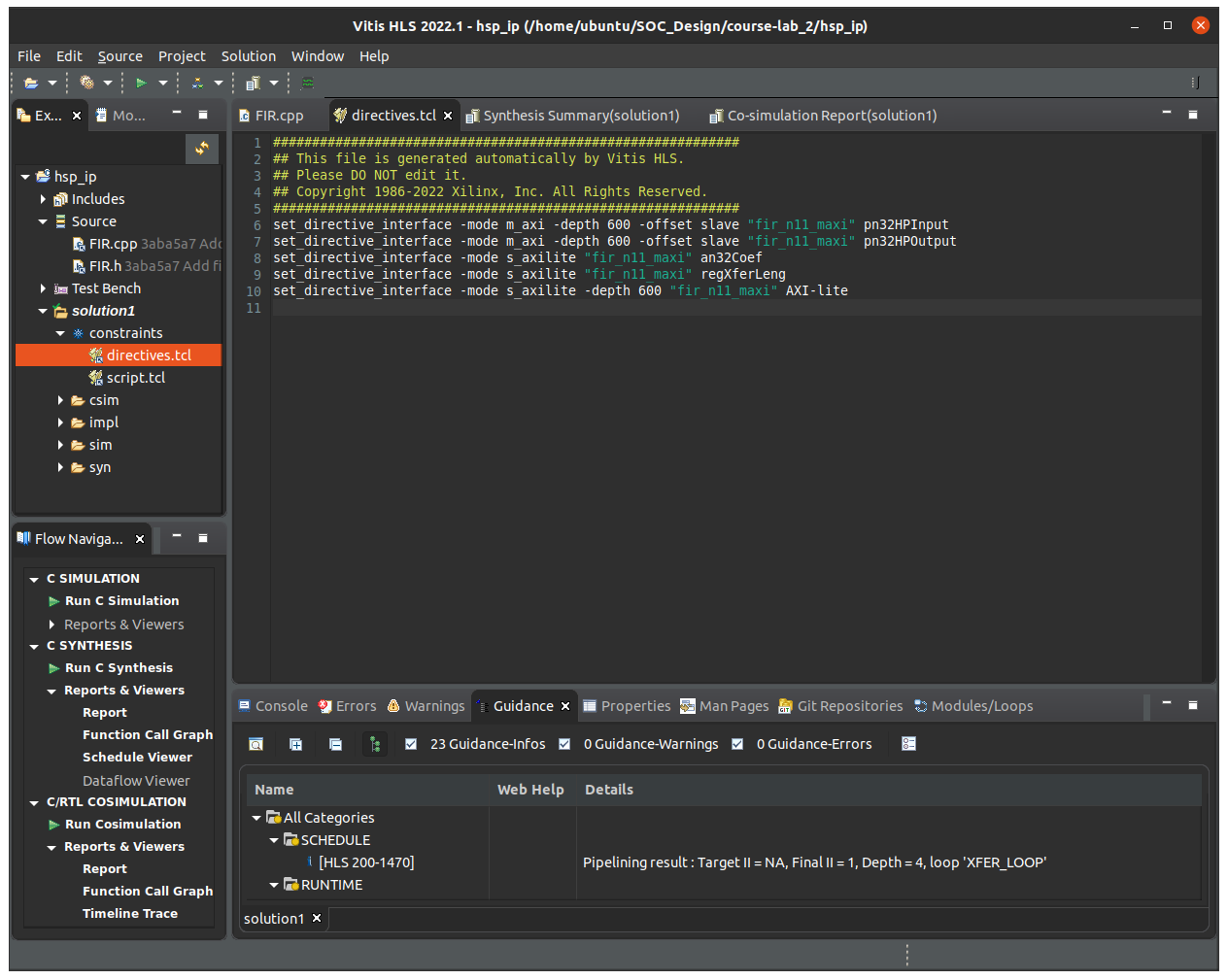


Figure2. directives

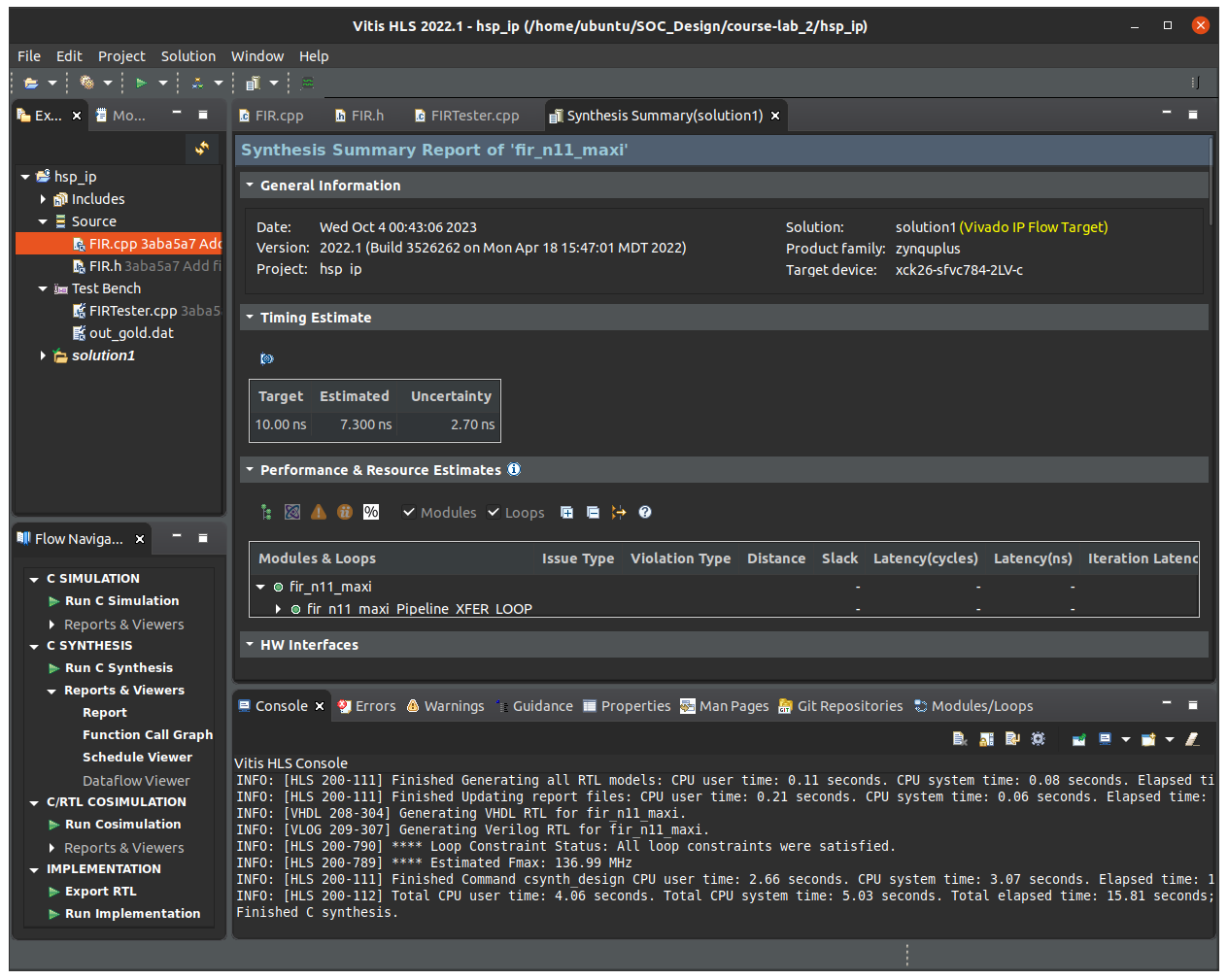


Figure3. C synthesis

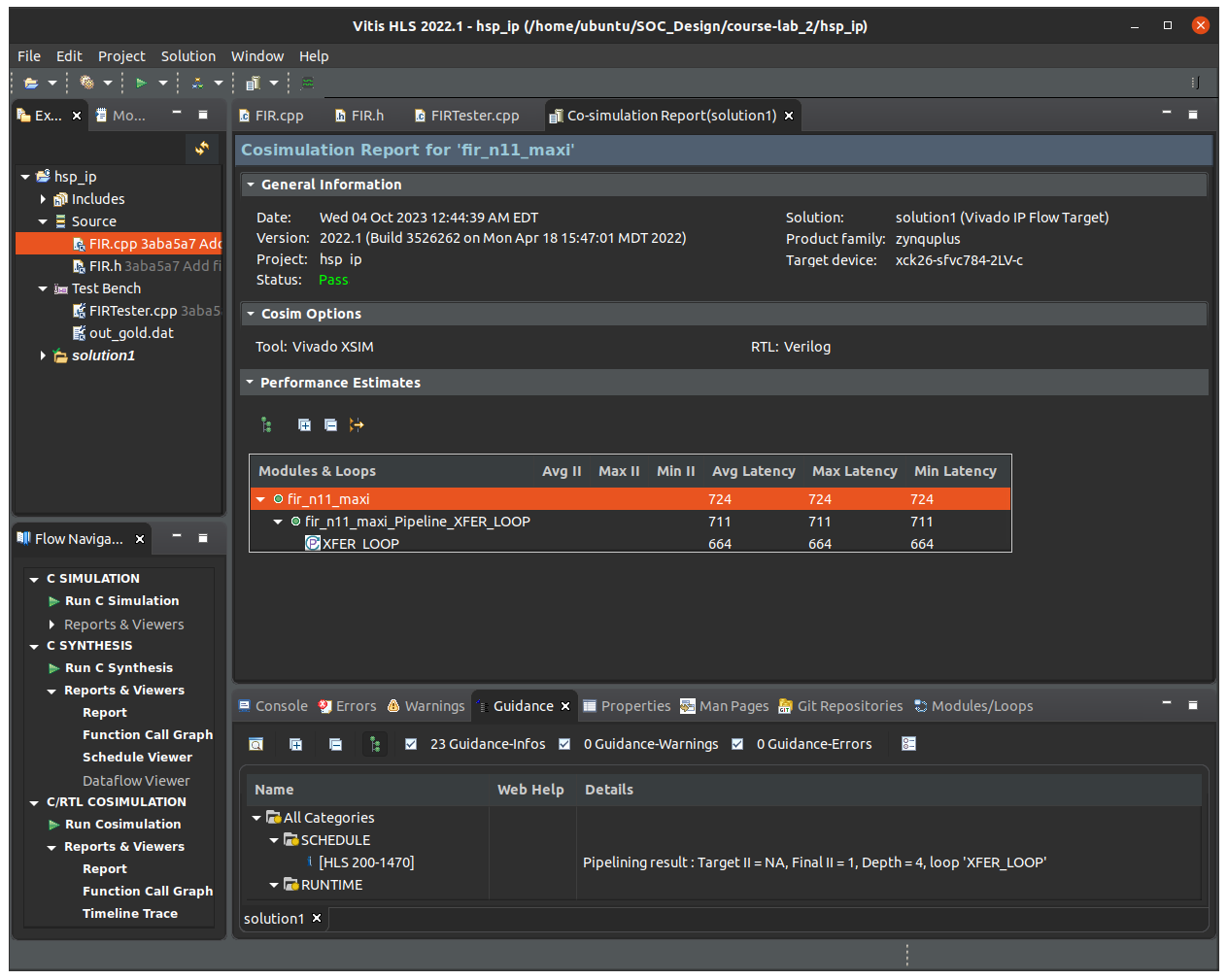


Figure4. Co-simulation

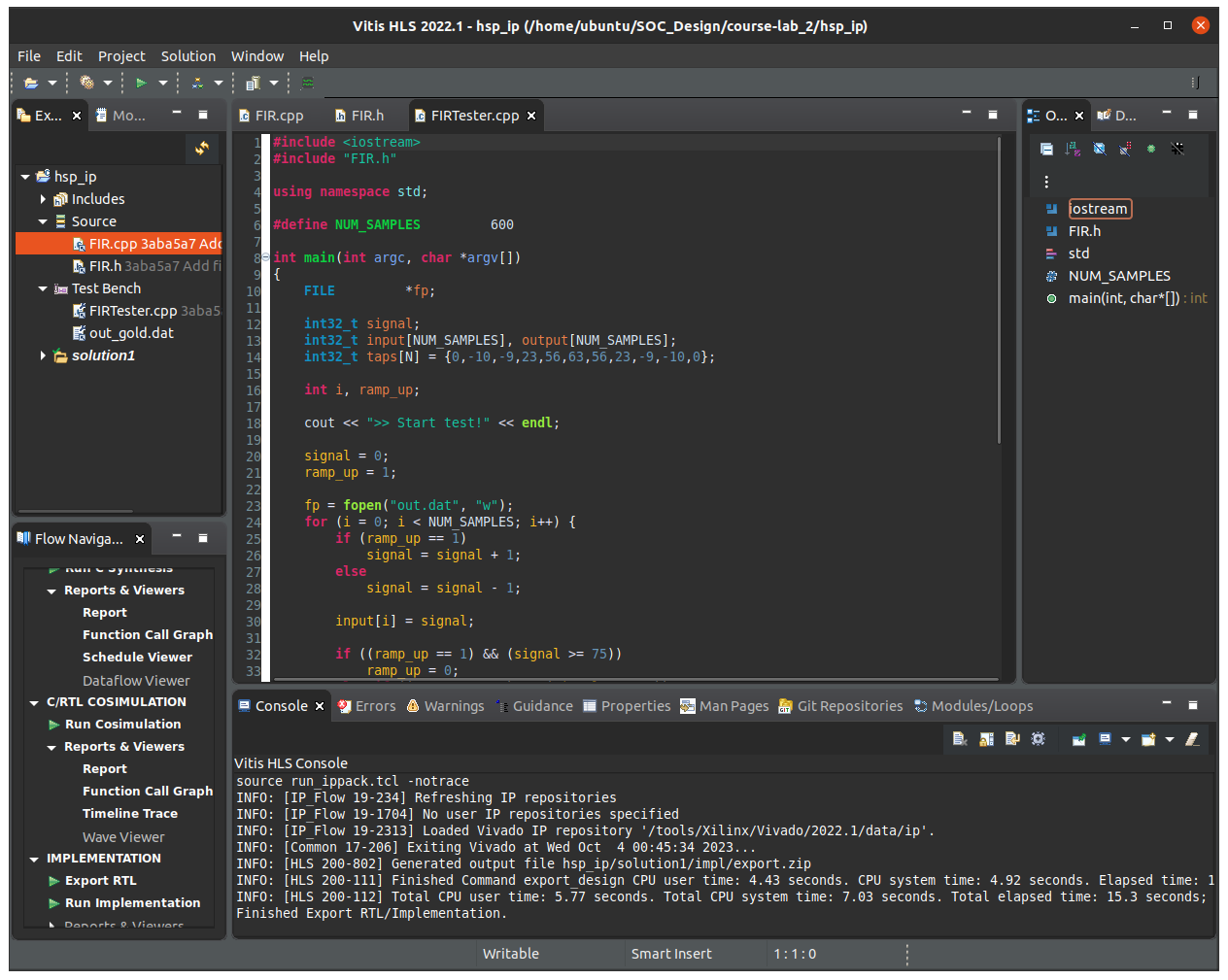


Figure5. Export RTL

1. VIVADO

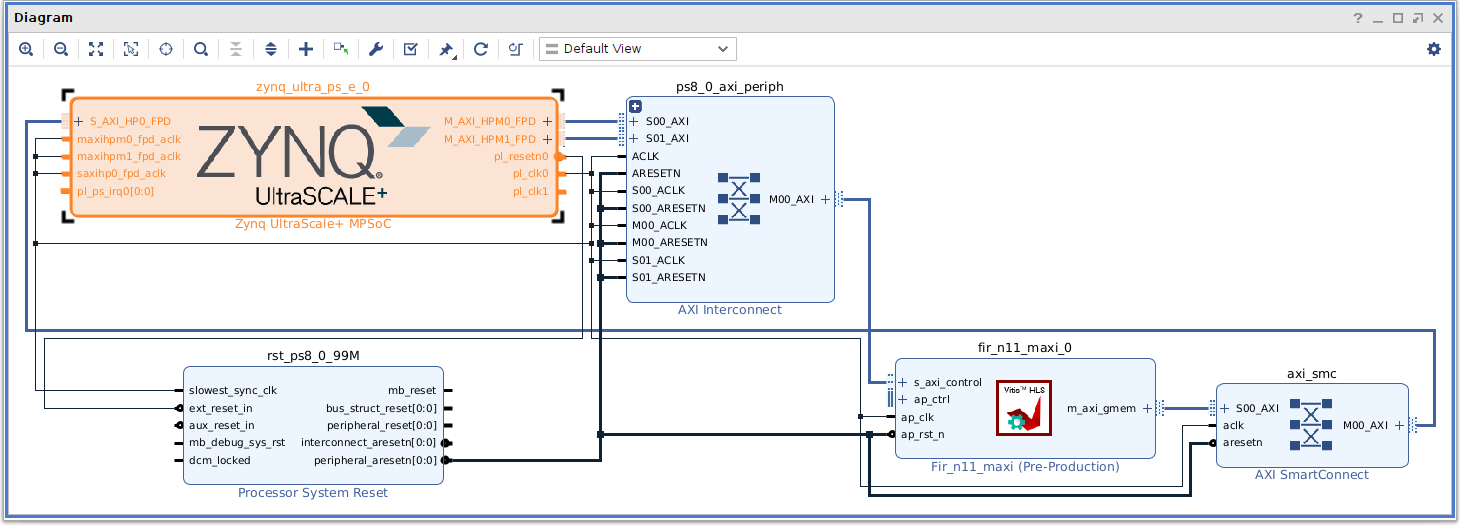


Figure6. Block Diagram

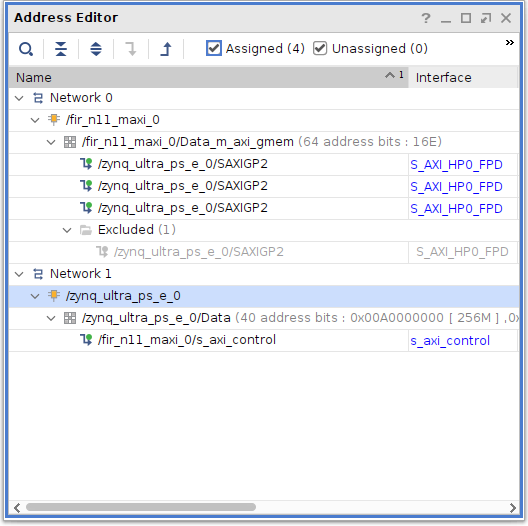


Figure7. Address Editor

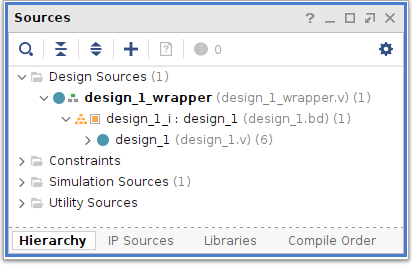


Figure8. Create Wrapper

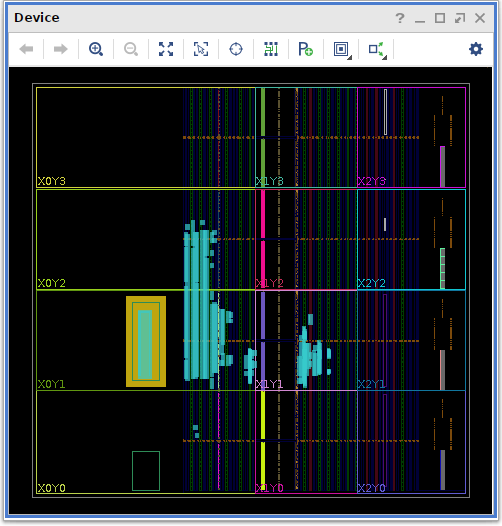


Figure9. Output of Bitstream Generating

依照Lab1和Lab2 Workbook的步驟完成了VITIS和VIVADO的操作，接著將產生的bitstream file上傳至kv260板進行模擬，但是卻無法產生預期的結果。

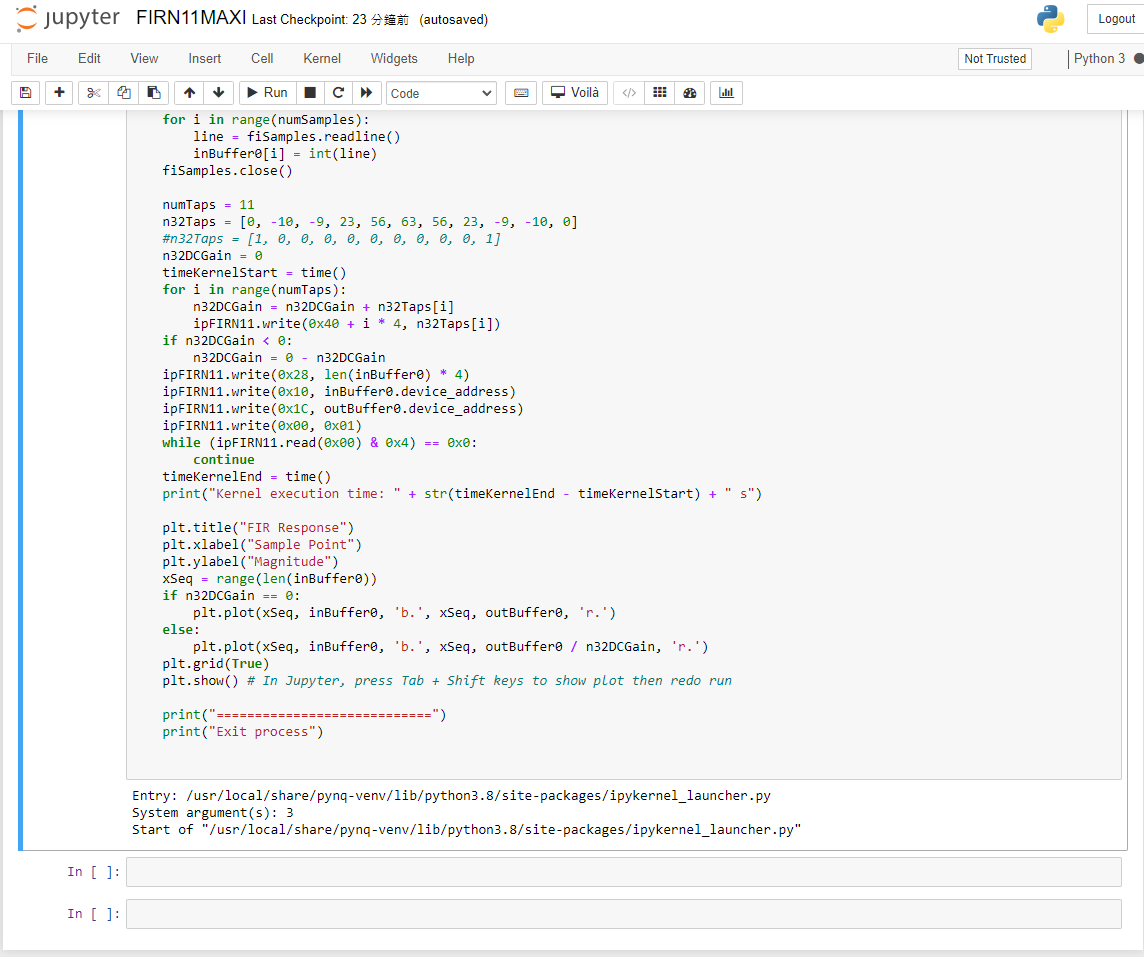


Figure10. Failed Jupyter output

經過反覆查看以及在討論區上討論，發現Block Diagram中多出了一個名叫”ap\_ctl”，而最後的解決方法是更改interface的宣告方式，不再使用directives而是直接寫在cpp檔中。

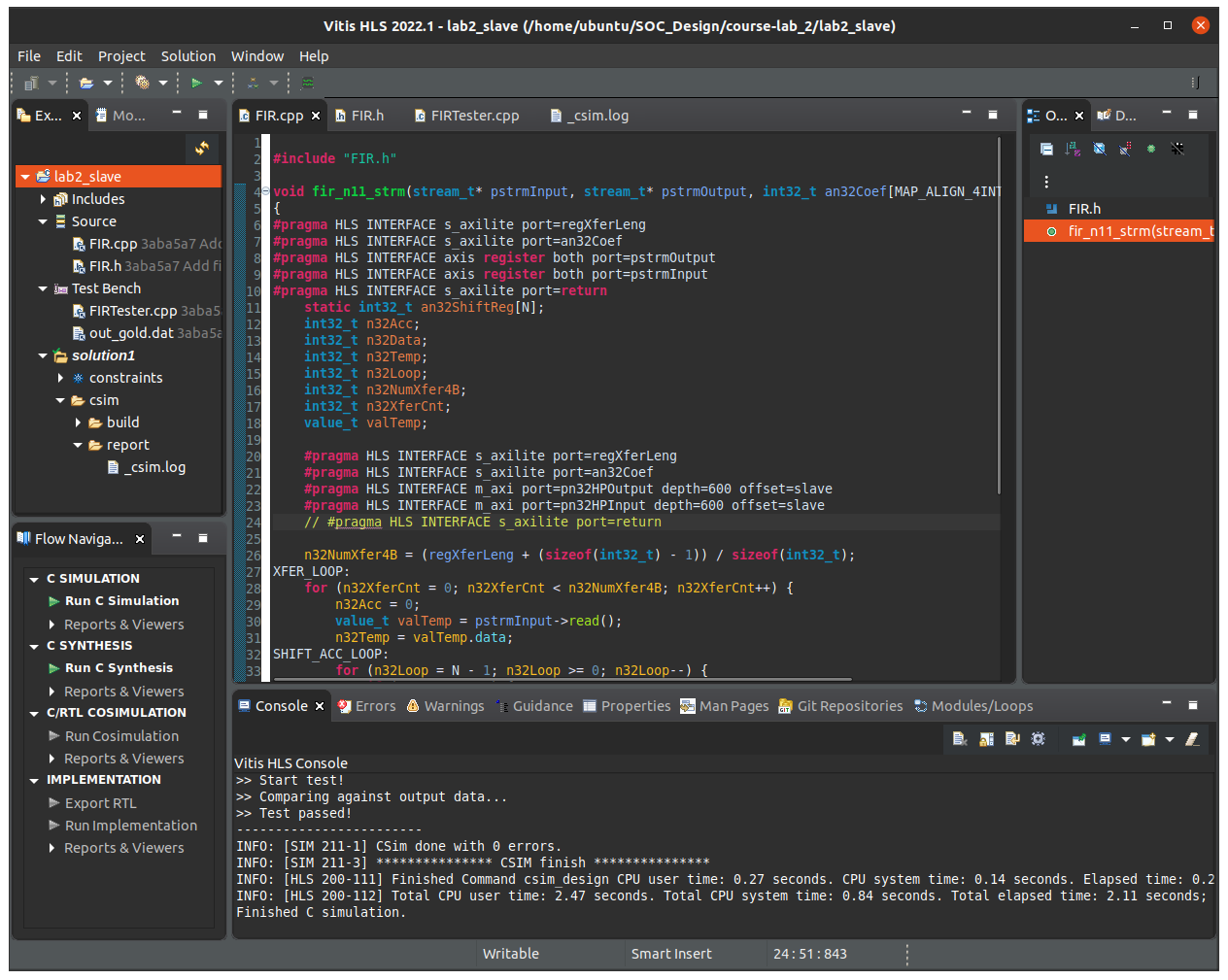


Figure11. Using pragma to define interface

更改後再重新完成的步驟。

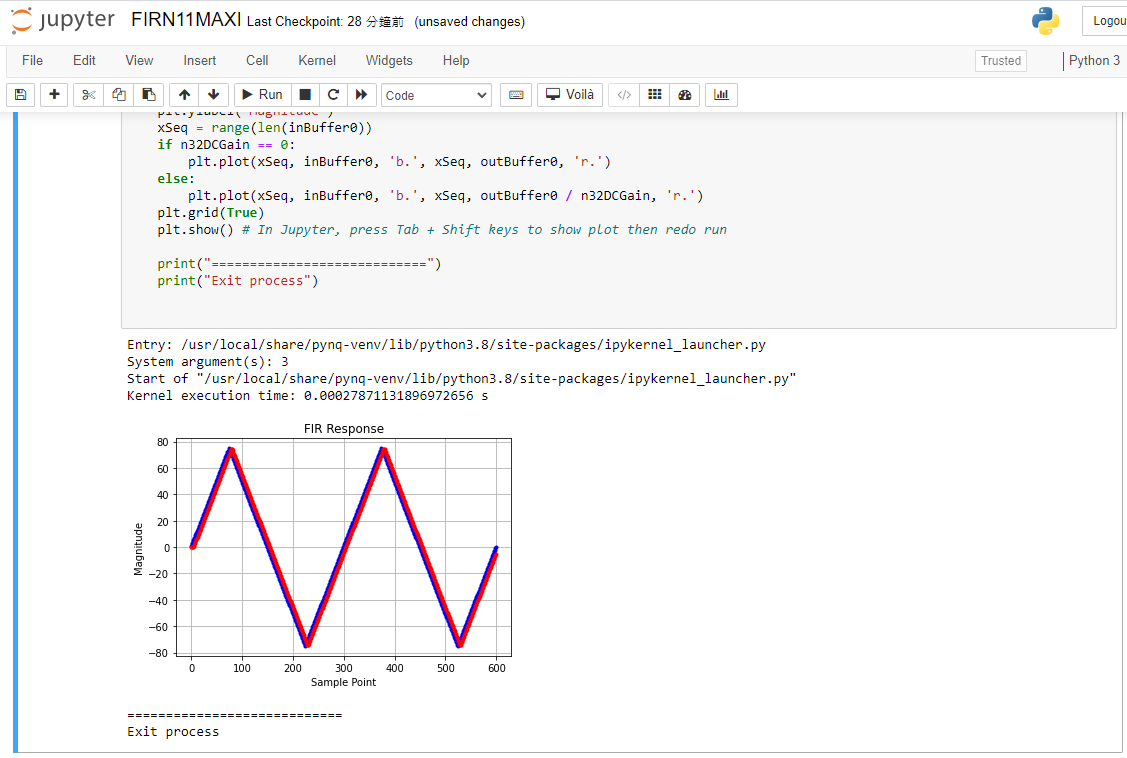


Figure12. Jupyter Output

## Part2: Stream Interface

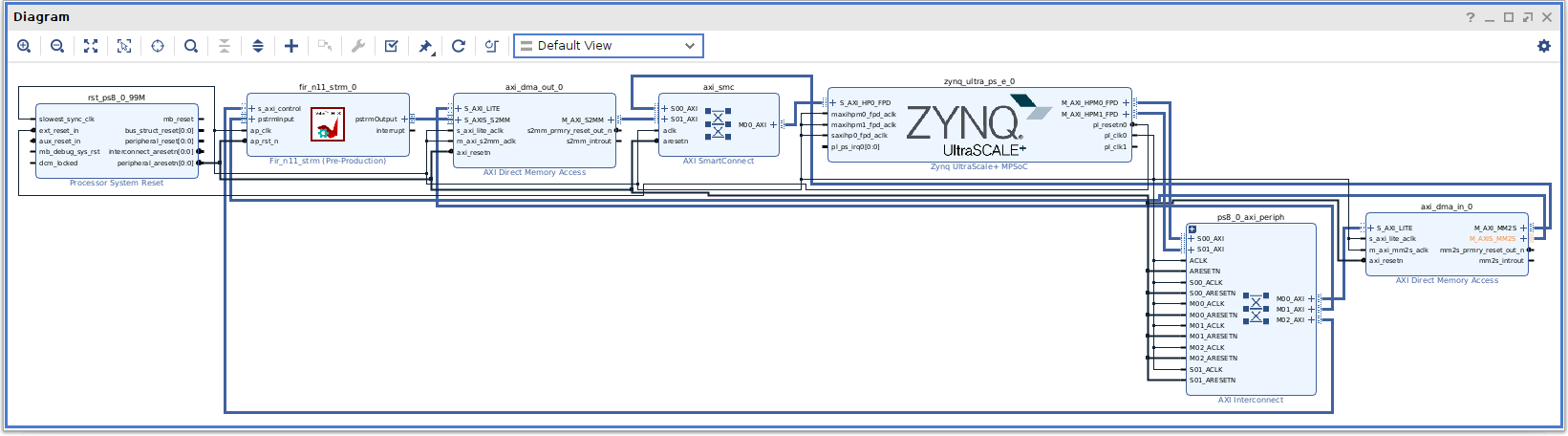


Figure13. Block Diagram

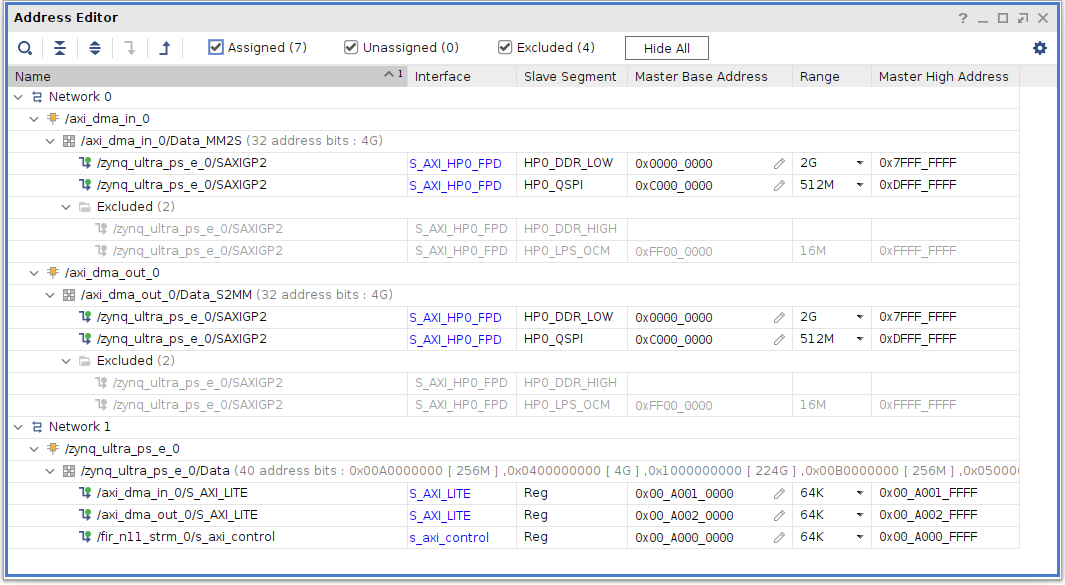


Figure14. Address Editor

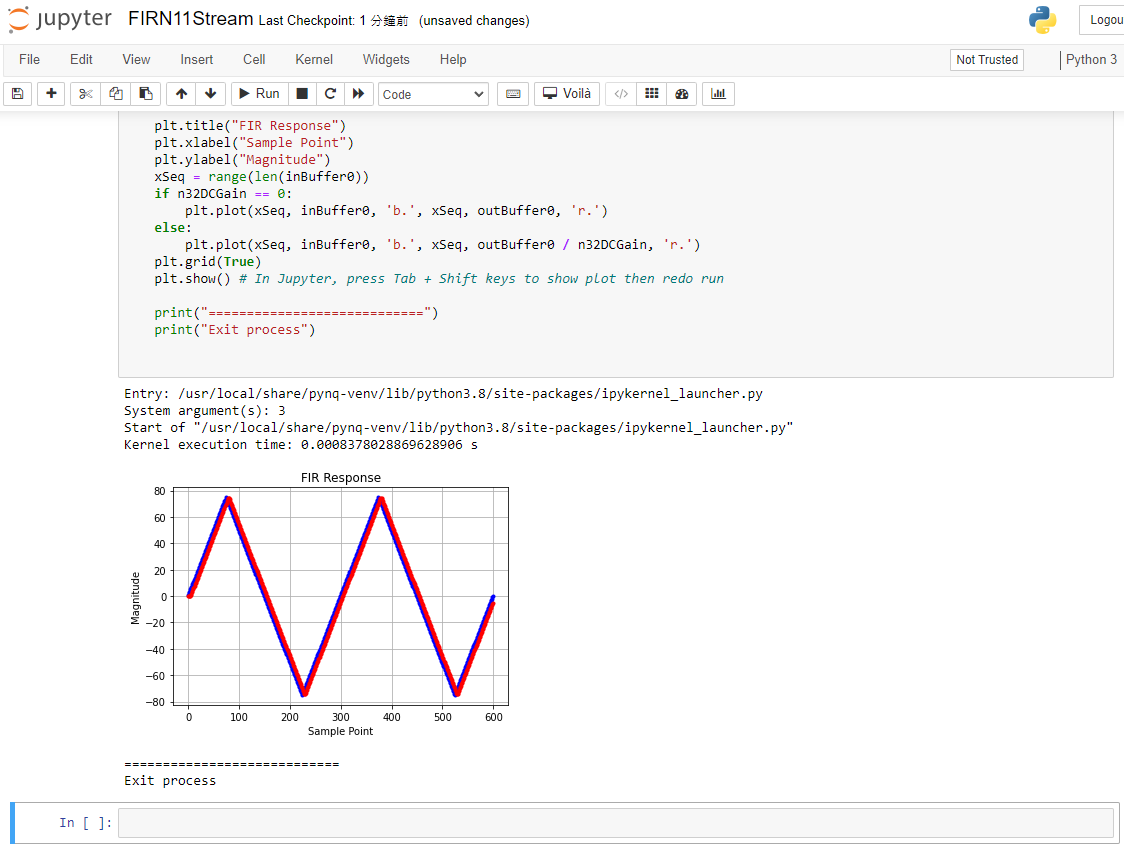
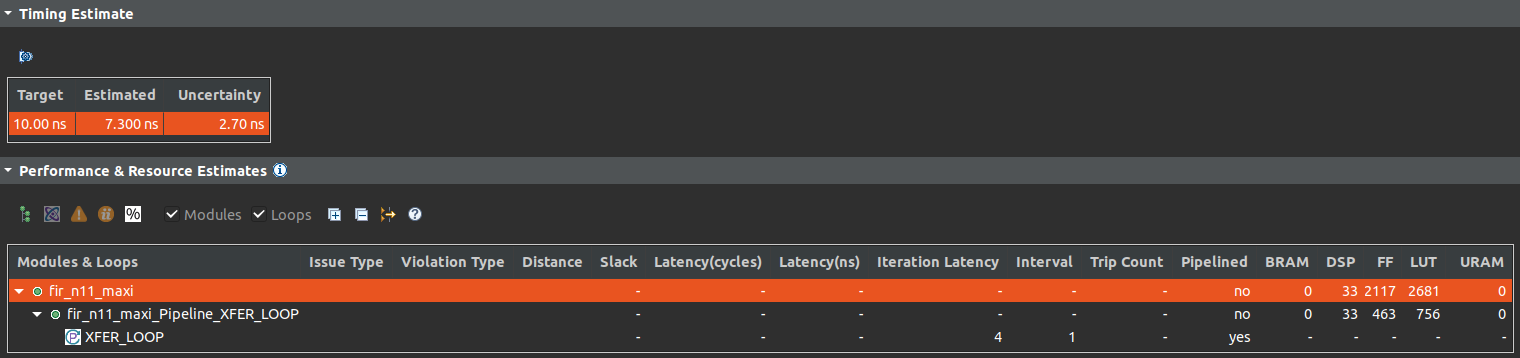
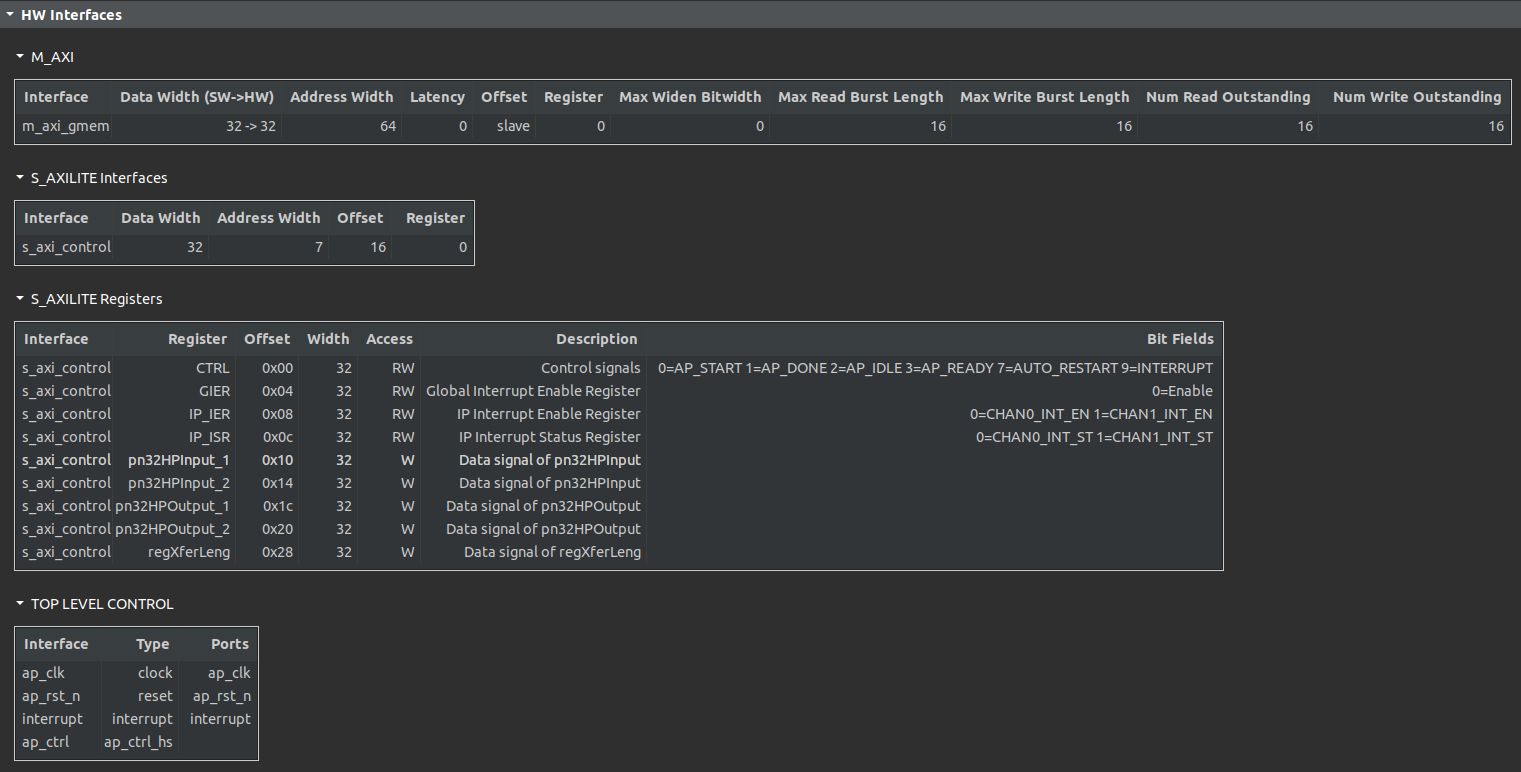
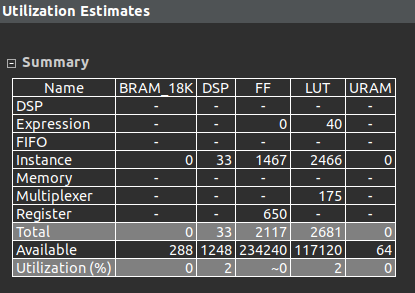


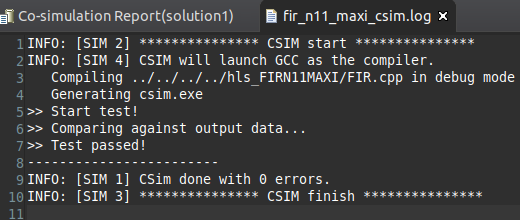
Figure15. Jupyter output

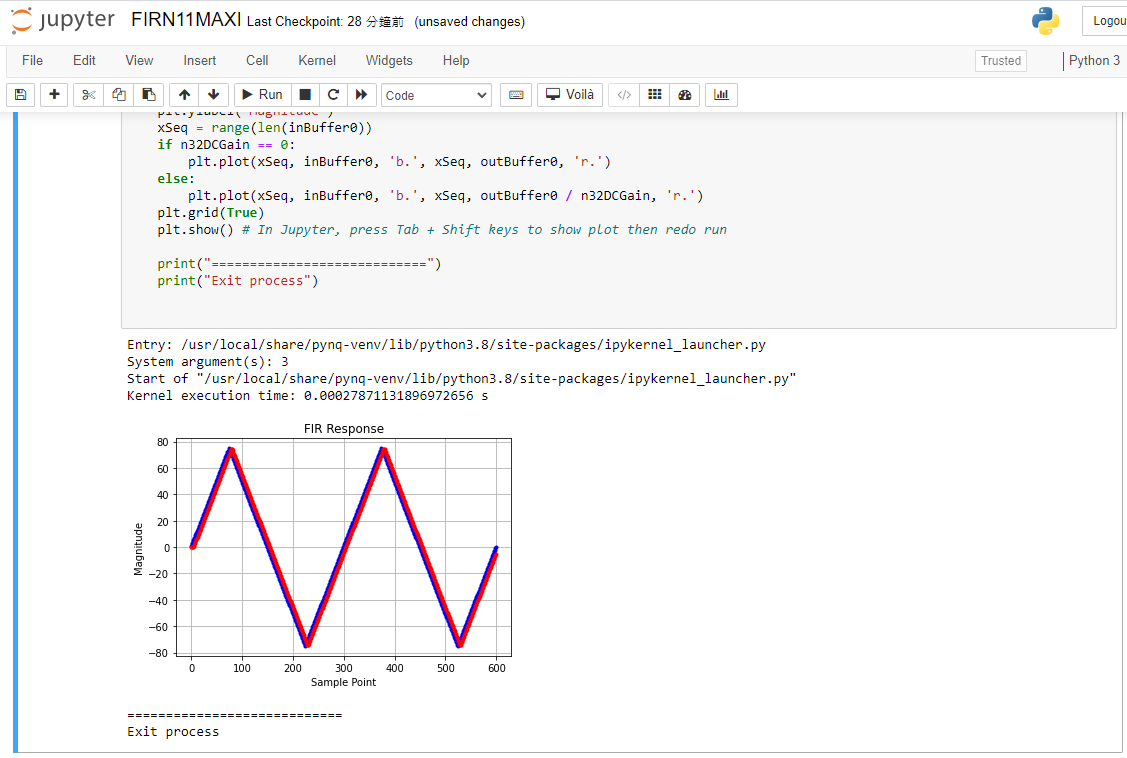
## Part3: Performance

1. Master Interface









1. Stream Interface

