VLSI System Design (Graduate Level)

Fall 2023

HOMEWORK II

REPORT

Must do self-checking before submission:

Compress all files described in the problem into one tar

All SystemVerilog files can be compiled under SoC Lab environment

All port declarations comply with I/O port specifications

Organize files according to File Hierarchy Requirement

No any waveform files in deliverables

Student name: \_\_\_林晉宇\_ \_張正德\_\_\_\_\_

Student ID: \_\_P76121673\_ \_P76121089\_\_

Summary

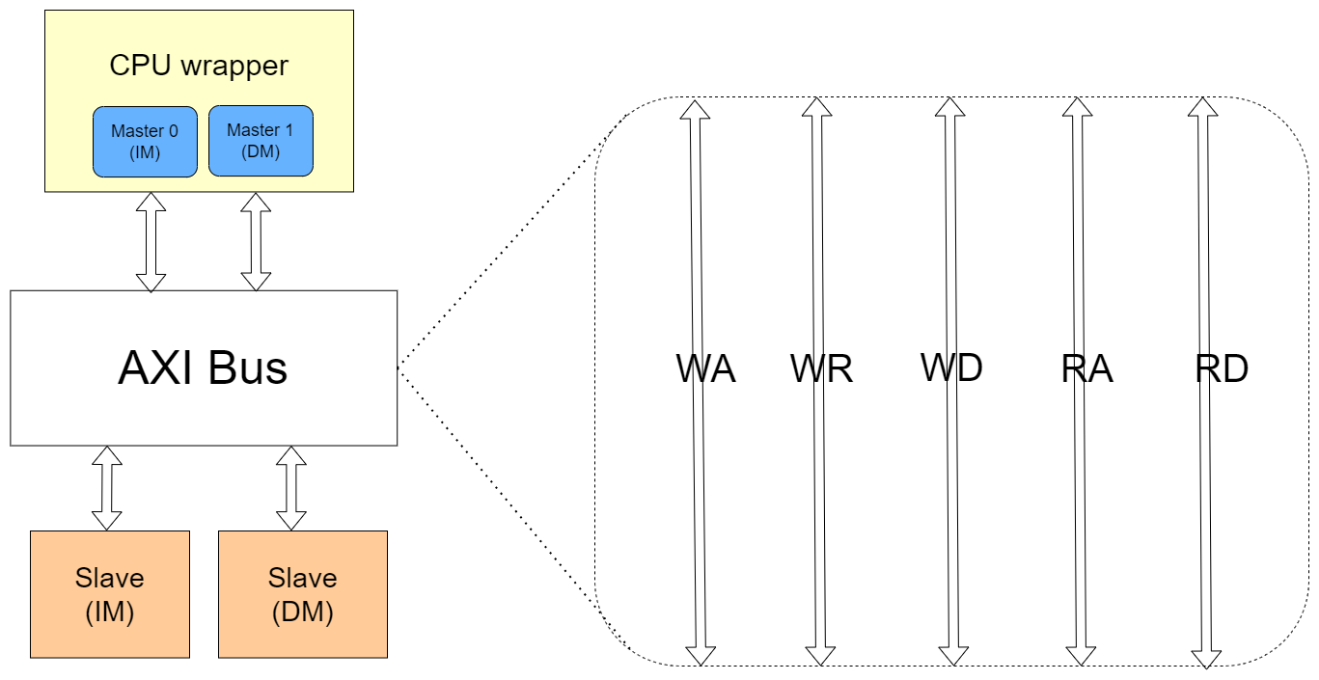
* Module :
* AXI ( AR, DR, AW, DW, RW, Arbiter, Decoder, DefaultSlave )
* Sram\_wrapper
* CPU\_wrapper ( CPU, Master )
* 將HW1的CPU增加IM\_Stall以及DM\_Stall線路使得在連接上AXI後能夠將CPU stall，讓CPU運作正確

貢獻度

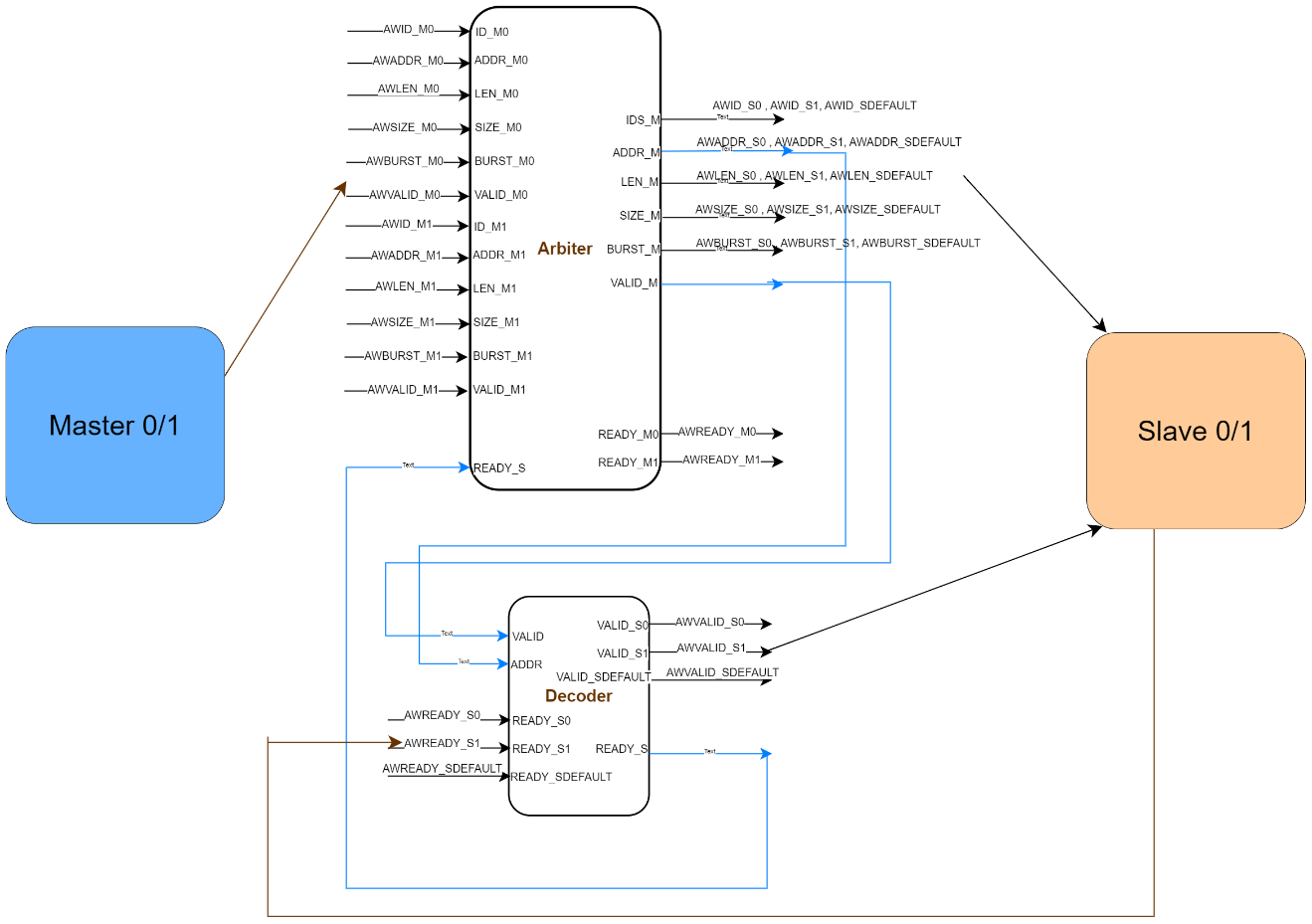
|  |  |
| --- | --- |
| 林晉宇 | 張正德 |
| P76121673 | P76121089 |
| 50% | 50% |

Overview

Architecture of Pipeline CPU with AXI Bus :



Arbiter & Decoder :



Waveform verification

本 次作業主要是實作介於CPU和Instruction Memory 以及 Data Memory之間的AXI Bus，因此波型驗證將著重於LW、SW指令以及從IM讀取instructions，驗證CPU (Master) 和Memory (Slave)之間的關係與溝通。

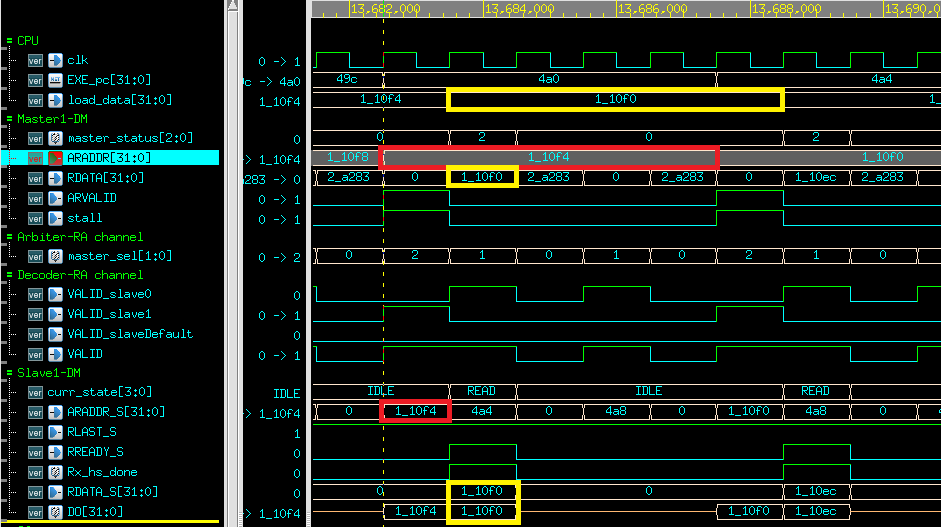
1. Fetch instructions from IM via AXI

一張含有 文字, 螢幕擷取畫面, 鮮豔, 陳列 的圖片

自動產生的描述

* 1. master\_status : 0→read\_addr，2→read\_data ；
  2. master\_sel : 0→default ，1→master0，2→master1。
  3. 上圖紅框處可以看到instr\_addr訊號線在master0-ARVALID以及Slave0-ARREADY同時為1的時候，透過Arbiter以及Decoder選擇master0與slave0為目的地，將instr\_addr從master端正確傳送到slave端。
  4. 上圖黃框處可以看到，在master0-RREADY以及Slave0-RVALID同時為1的時候(Rx\_hs\_done=1)，將iSRAM輸出的DO給RDATA並傳送到Master端。
  5. Slave0(IM)在IDLE-state時接收AXI傳送過來的addr，並在ARx\_hs\_done後跳到READ-state。

1. LW instruction



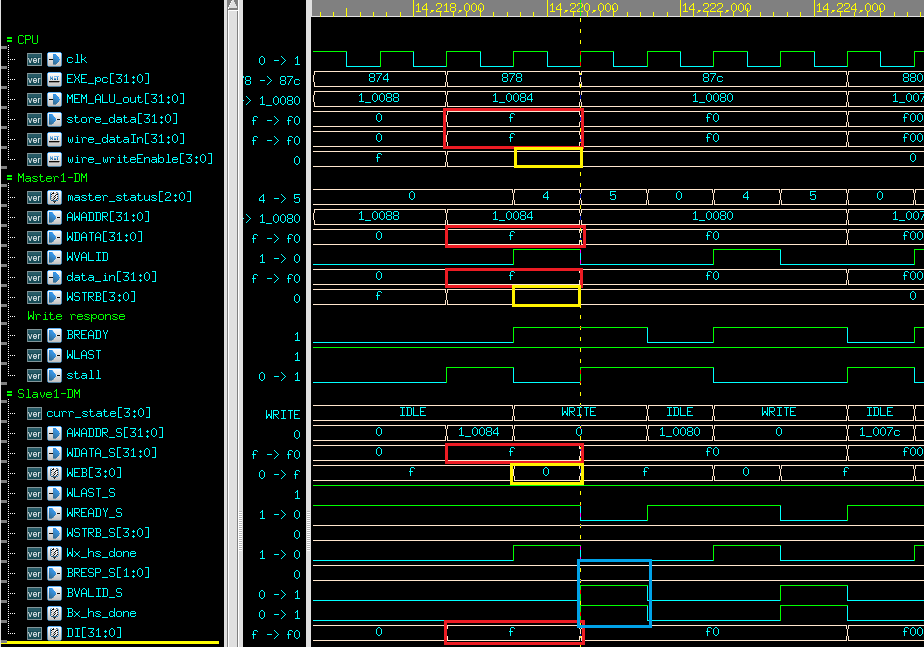
* 1. master\_status : 0→init state(send address)，2→read\_data state；
  2. master\_sel : 0→default ，1→master0，2→master1。
  3. 見prog0中編譯後的main.log可知pc為49c位址的指令是lw t0, 0(t0)。
  4. 該指令在下一個clk進入MEM-stage，開始讀取Data memory。
  5. 上圖紅框處可以看到ARADDR訊號線在master1-ARVALID以及Slave1-ARREADY同時為1的時候，透過Arbiter以及Decoder選擇master1與slave1為目的地，將ARADDR從master端正確傳送到slave端。
  6. 上圖黃框處可以看到，在master1-RREADY以及Slave1-RVALID同時為1的時候(Rx\_hs\_done=1)，將iSRAM輸出的DO給RDATA並傳送到Master端，傳入CPU的load\_data中。
  7. Slave1(DM)在IDLE-state時接收AXI傳送過來的addr，並在ARx\_hs\_done後跳到READ-state。
  8. 可以發現到此指令在DM master取得data後，CPU仍被stall，是因為IM master及DM master同時需要AXI bus取得data，但DM master會先於IM master獲得Arbiter鎖定，因此在DM master完成transfer後由IM接著使用AXI並stall CPU，並且可以發現到因為ARREADY一直為拉高的情形，因此IM slave在read\_data的同時可以完成與IM Master的AR handshake。

1. SW instruction
   1. Write address channel (Master → Slave)

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述

* + 1. master\_status : 0→init state(send address)，4→write data state，5→write response；
    2. master\_sel : 0→default ，1→master0，2→master1。
    3. 見prog0中編譯後的main.log可知pc為874位址的指令是sw t0, -4(s0)。
    4. 該指令在下一個clk進入MEM-stage，開始存取Data memory。
    5. 上圖紅框處可以看到AWADDR訊號線在master1-AWVALID以及Slave1-AWREADY同時為1的時候，透過Arbiter以及Decoder選擇master1與slave1為目的地，將AWADDR從master端正確傳送到slave端。
    6. 上圖黃框處可以看到Arbiter中的master\_sel訊號以及Decoder中的VALID訊號分別選擇到了正確的目標master1、slave1。
    7. 並且Slave1-DM在AW\_hs\_done後轉換state成WRITE-state，開始寫入資料(傳送WDATA)。
  1. Write data channel & Write response channel (Slave → Master)



* + 1. master\_status : 0→init state(send address)，4→write data state，5→write response；
    2. master\_sel : 0→default ，1→master0，2→master1。
    3. 承上一個write address channel，接續介紹下一個clk進入write data state。
    4. 上圖紅框處可以看到WDATA訊號線在master1-WVALID以及Slave1-WREADY同時為1的時候，，將要寫入的資料從CPU-store\_data傳到Master端的WDATA再正確傳送到Slave端(DI)。
    5. 並且master\_status也正確轉換到write\_data state，
    6. 上圖黃框處可以看到writeEnable也在此時從CPU傳入Slave端(DM內的WEB)
    7. 可以發現到完成W channel handshake後即回到init state。繼續寫入下一條SW指令的address。

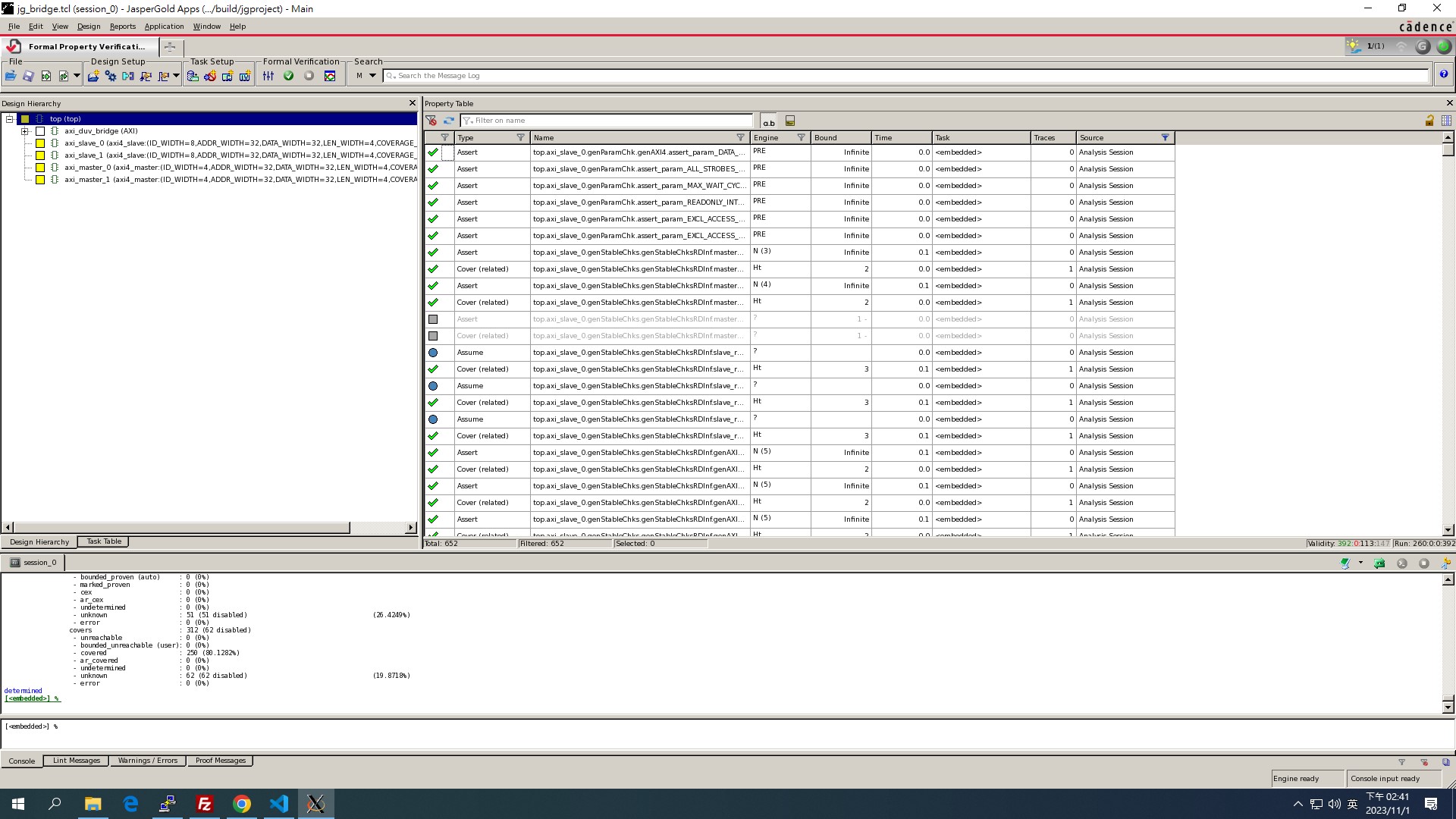
Lesson learned

* **處理jaspergold 問題:**

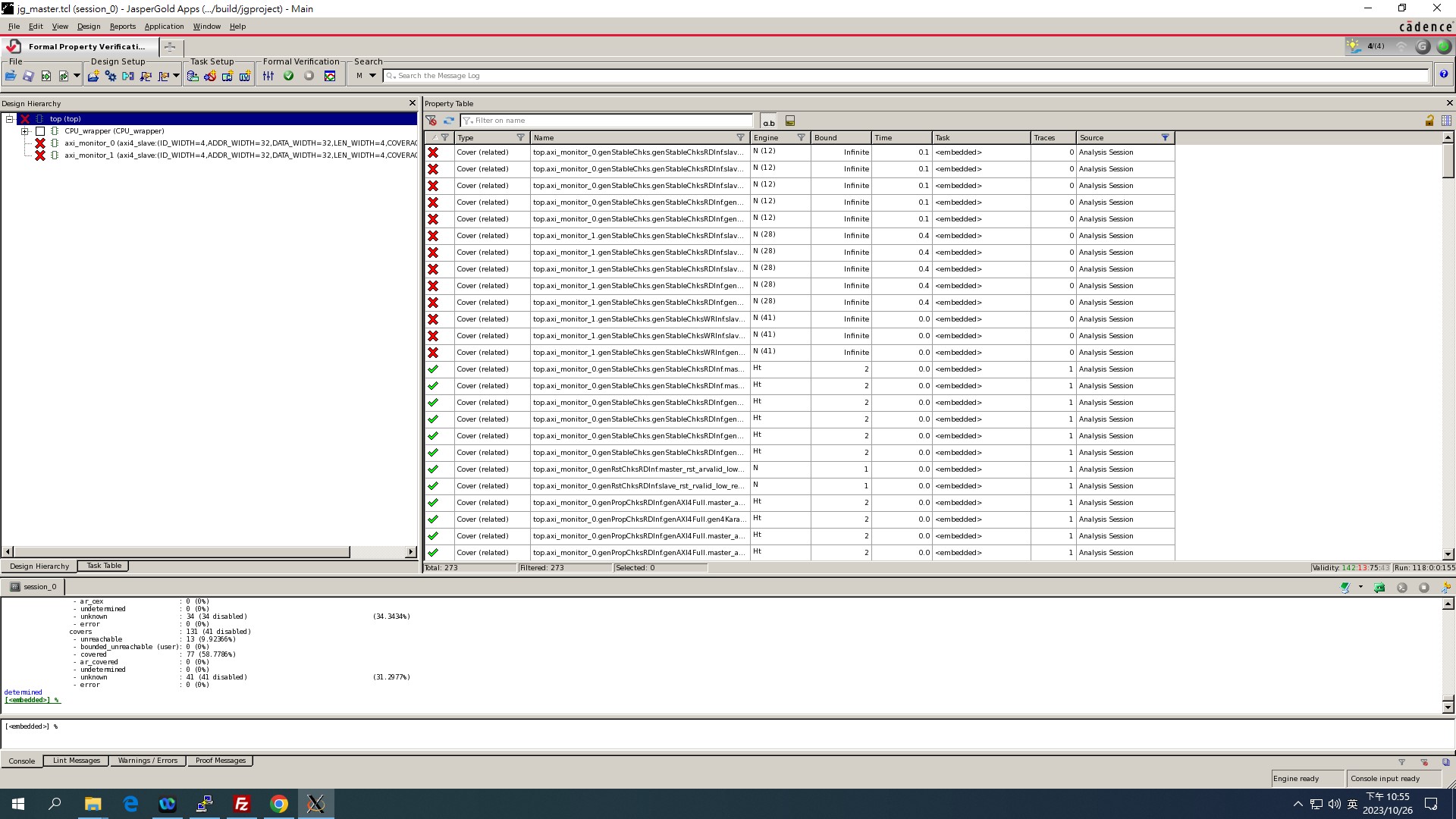
在驗證vip的時候會碰到很多程式沒寫好的情況，導致在jaspergold驗證的時候出錯。學到解決jaspergold的問題時，可以先從程式邏輯去下手。其中最難解的問題就是bridge部分的wlast assert，因為在CPU中只有Master1會寫入DM中，所以在rtl和syn中並沒有報錯，最後發現是WD中對應的slave設置錯誤。

Verify AXI architecture

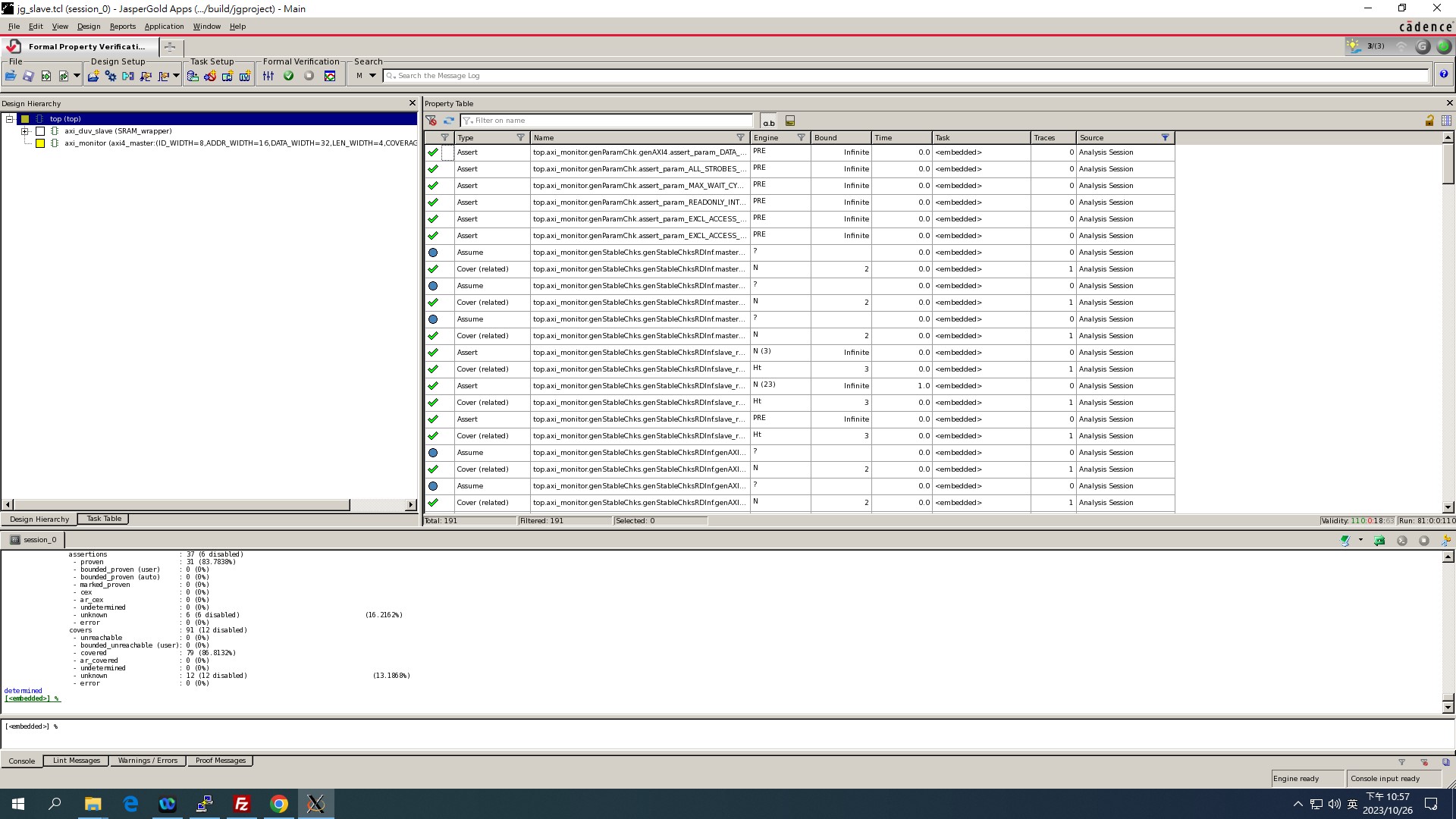
1. Bridge



1. Master
   * 在Master有些cover並沒有辦法通過驗證，這是因為在write response與read response，我的Master只有當Slave發出valid訊號時才會同時將ready拉起來，所以導致某些cover會沒有辦法驗證出來。
   * 在我們的master中一收到Slave端傳來的valid訊號就會馬上將ready拉起來，所以並不會產生valid為1, ready為0的狀況，所以才會導致cover產生。

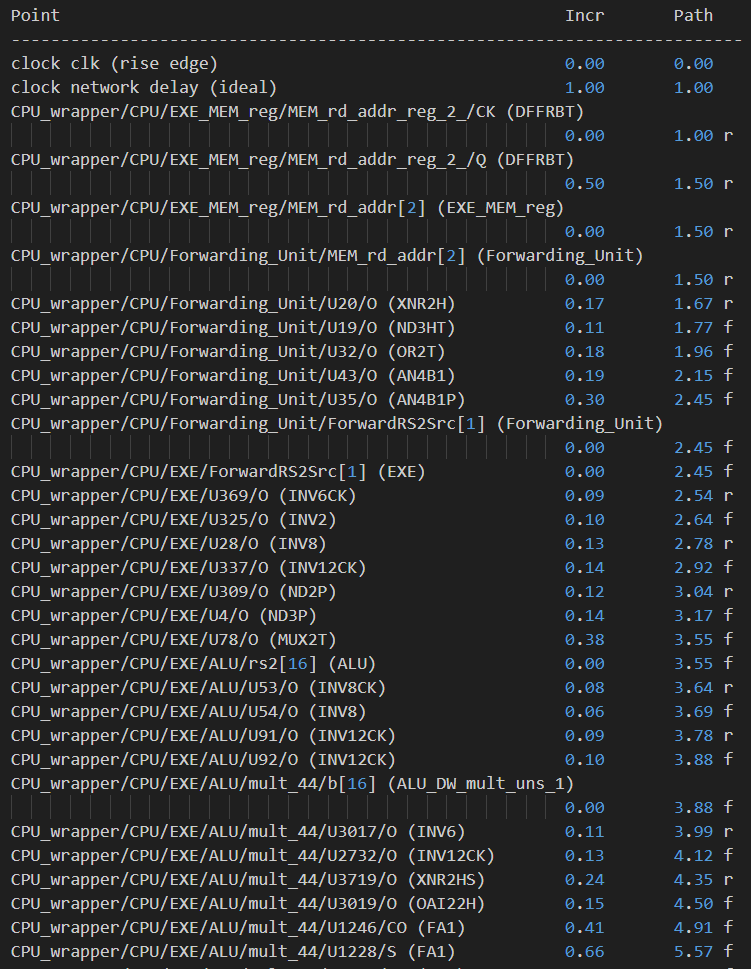
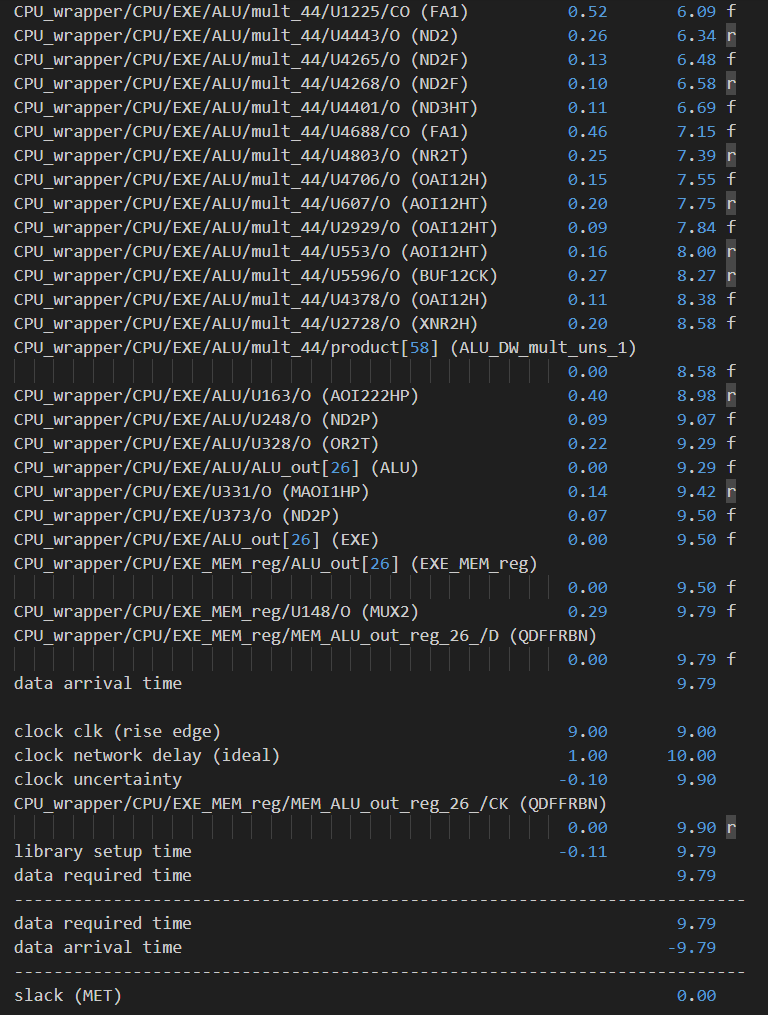


1. Slave

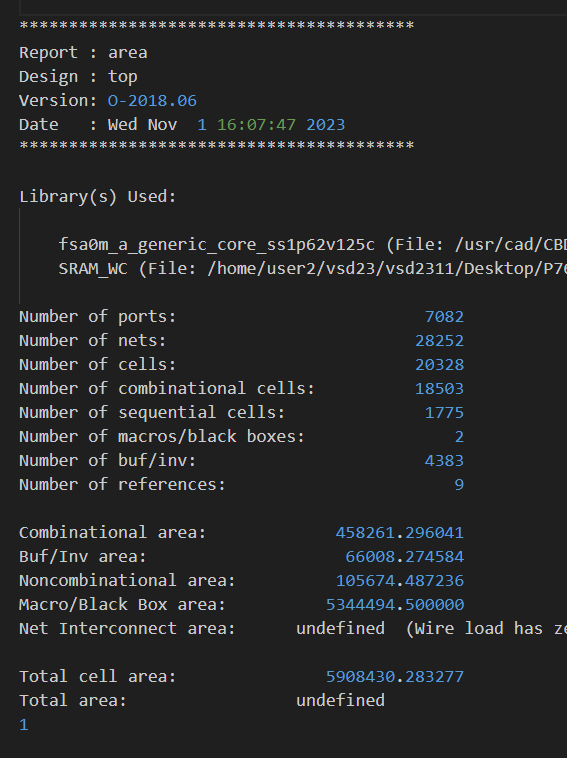


Report

* Timing report

* Area report



Simulation results

|  |  |  |
| --- | --- | --- |
|  | rtl | syn |
| Prog0 |  |  |
| Prog1 |  |  |
| Prog2 |  |  |
| Prog3 |  |  |
| Prog4 |  |  |
| Prog5 |  |  |

|  |  |  |
| --- | --- | --- |
|  | rtl | syn |
| Prog0 | Pass | Pass |
| Prog1 | Pass | Pass |
| Prog2 | Pass | Pass |
| Prog3 | Pass | Pass |
| Prog4 | Pass | Pass |
| Prog5 | Pass | Pass |

Performance

|  |  |
| --- | --- |
| Cycle time | 9ns |
| Total Cell Area | 5908430.283277 |