Advanced VLSI System Design (Graduate Level)

Fall 2024

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: \_\_\_\_劉益祥\_\_\_\_

Student ID: \_\_\_\_\_P76131084\_\_\_\_

**Summary**

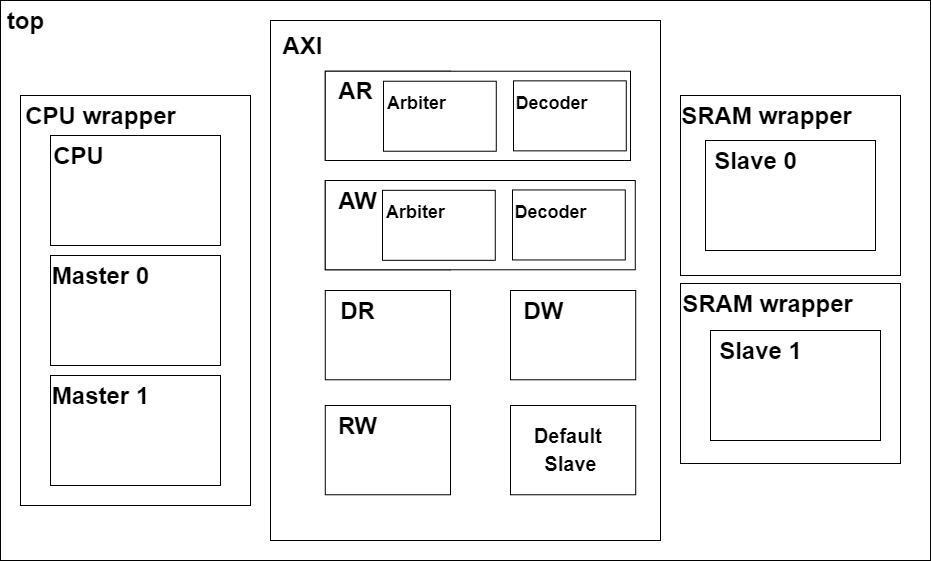
1. 完成hw2要求的module

* AXI
* Sram wrapper
* CPU wrapper

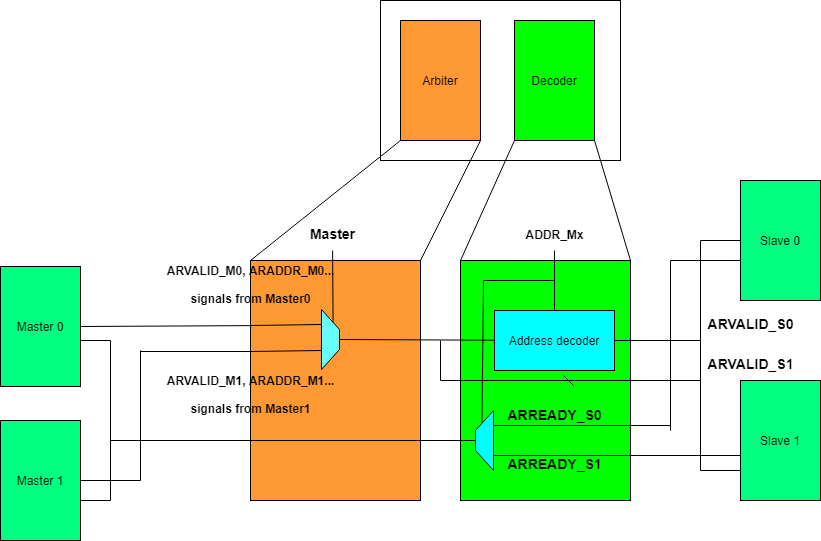
1. 將hw1的CPU加上stall使其合併AXI後能正確運行
2. 合併hw1的CPU及hw2的AXI

**Overview**

整體架構示意圖

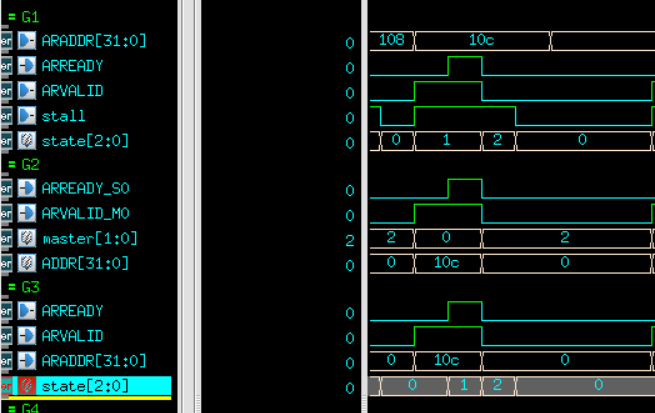


AW及AR中之Arbiter及Decoder示意圖



**Waveform verification**

**Read Address**

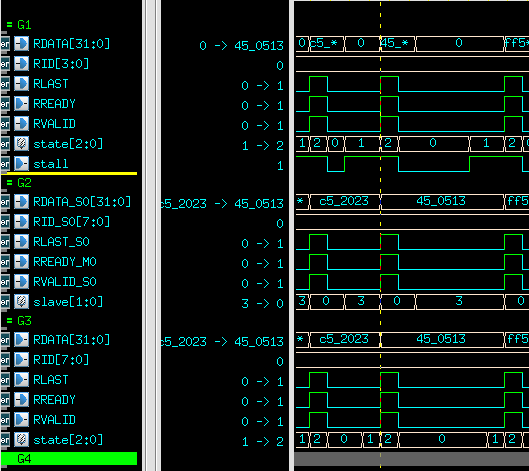


1. 在接收到CPU傳來的地址後，master會從idle(state0)進入transfer\_read\_address(state1)，開始進行地址傳送，向CPU傳送stall請求並將ARVALID拉為1(圖中紅框)
2. AXI在接收到master傳來的ARVALID訊號後會將master signal 設為傳送要求的master值，以圖中為例，由於是master0傳來的所以master值會為0。之後根據ADDR向對應的slave發送ARVALID訊號。(圖中紫框)
3. Slave收到ARVALID訊號後會從idle(state0)進入transfer\_read\_address(state1)，並傳送ARREADY訊號給master。(圖中黃框)
4. AXI收到ARREADY訊號後會直接傳給master，master收到ARREADY後會進入transfer\_read\_data (state 2)開始等待接收資料。(圖中灰框)

**Write Address:**

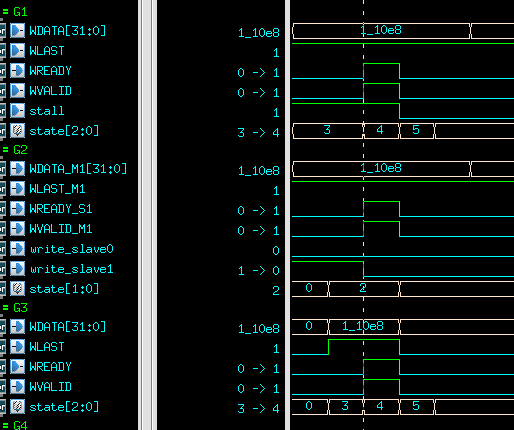
與Read address基本相同，故省略。

**Read Data:**



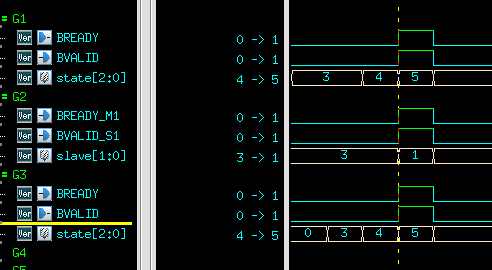
1. 在read address結束後，slave的state由transfer\_read\_address(state 1)進入transfer\_read\_data(state 2)。並同時將RVALID拉為1。(圖中紅框)
2. AXI收到RVALID訊號後，會將slave設為對應的salve值，以圖中為例，由於傳送RVALID訊號的是slave0，故slave值會被設為0。之後根據RID傳送DATA到對應的master ( RID[4] = 0傳到master0，RID[4] = 1 傳到master1)。(圖中紫框)
3. Master在此時應為transfer\_read\_data state(state 2)，並於收到RVALID後把RREADY拉為1，若從slave傳來的RLAST為1，則結束此次資料傳輸並把stall拉為0。(圖中黃框)
4. AXI及slave收到從master傳來的RREADY訊號時，若此時RLAST同時為1，則結束此次傳輸。

**Write Data:**



1. 在write address結束後，master的state由transfer\_write\_address(state 3)進入transfer\_write\_data(state 4)。並同時將WVALID拉為1。(圖中紅框)
2. AXI收到WVALID訊號後，會將state設為對應的master值，以圖中為例，由於傳送WVALID訊號的是master1，故state值會被設為1。之後根據從AW channel傳來的write\_slave0、write\_slave1訊號決定要寫入的slave，以圖中為例，write\_slave1為1故要寫入的slave為slave1。(圖中紫框)
3. slave在此時應為transfer\_write\_data state(state 4)，並於收到WVALID後把WREADY拉為1，若從master傳來的WLAST為1，則結束transfer\_write\_data state並進入write\_response state(state 5)。(圖中黃框)
4. AXI收到從slave傳來的WREADY訊號時，若此時WLAST同時為1，則結束此次write data傳輸。
5. Master收到slave傳來的WREADY訊號時，若此時WLAST為1，則進入write\_response state(state 5)並將stall拉為0。(圖中灰框)

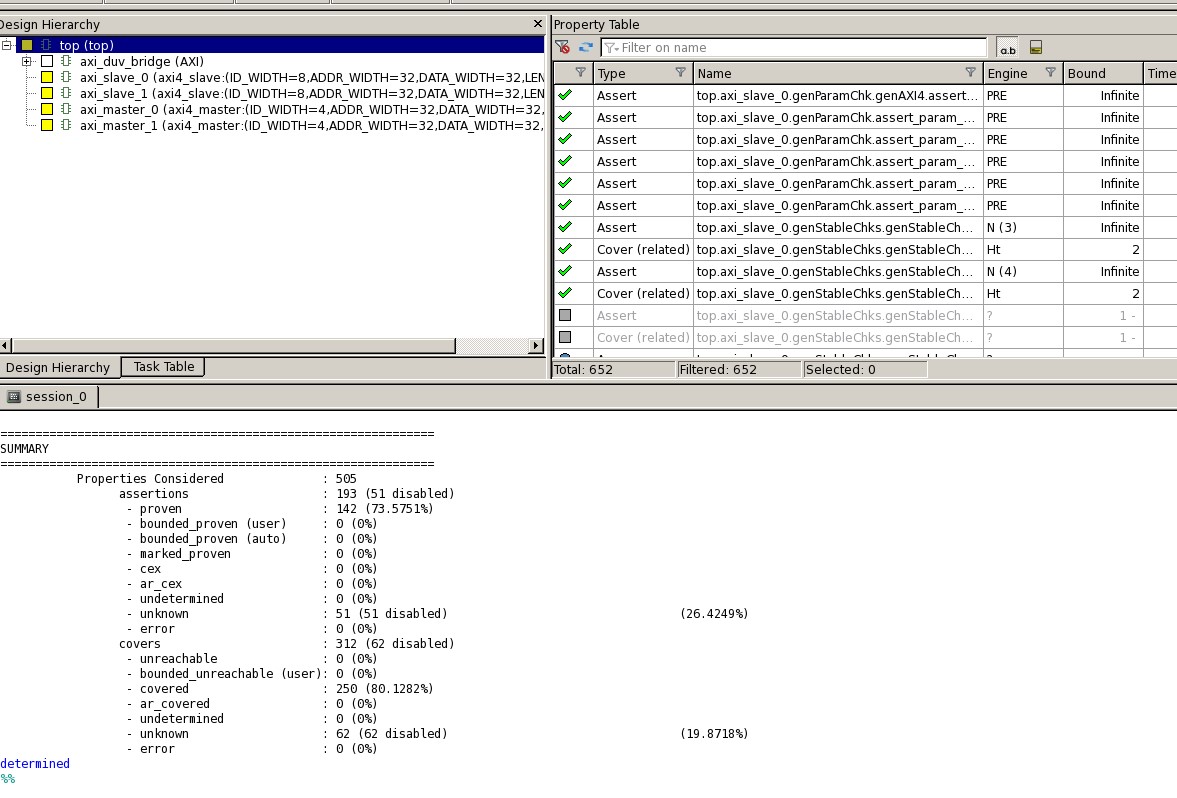
**Write Response:**



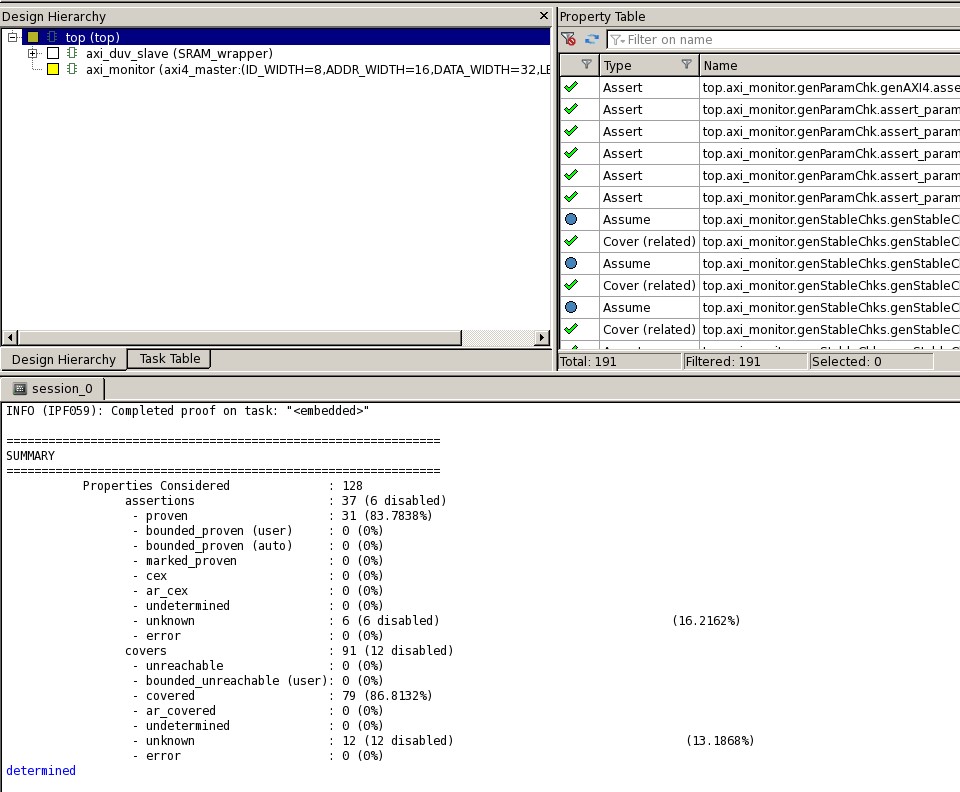
1. 結束write data後，slave會進入write\_response state(state 5)並將BVALID拉為1。(圖中紅框)
2. AXI收到BVALID訊號後，會將slave訊號設為對應的slave值，以圖中為例，傳送BVALID訊號的為slave1，故slave值為1。(圖中紫框)
3. Master此時應為write\_response state(state 5)，並於收到BVALID後將BREADY設為1。(圖中黃框)
4. AXI和slave收到BREADY後結束write response傳輸。

**Verify JasperGold VIP**

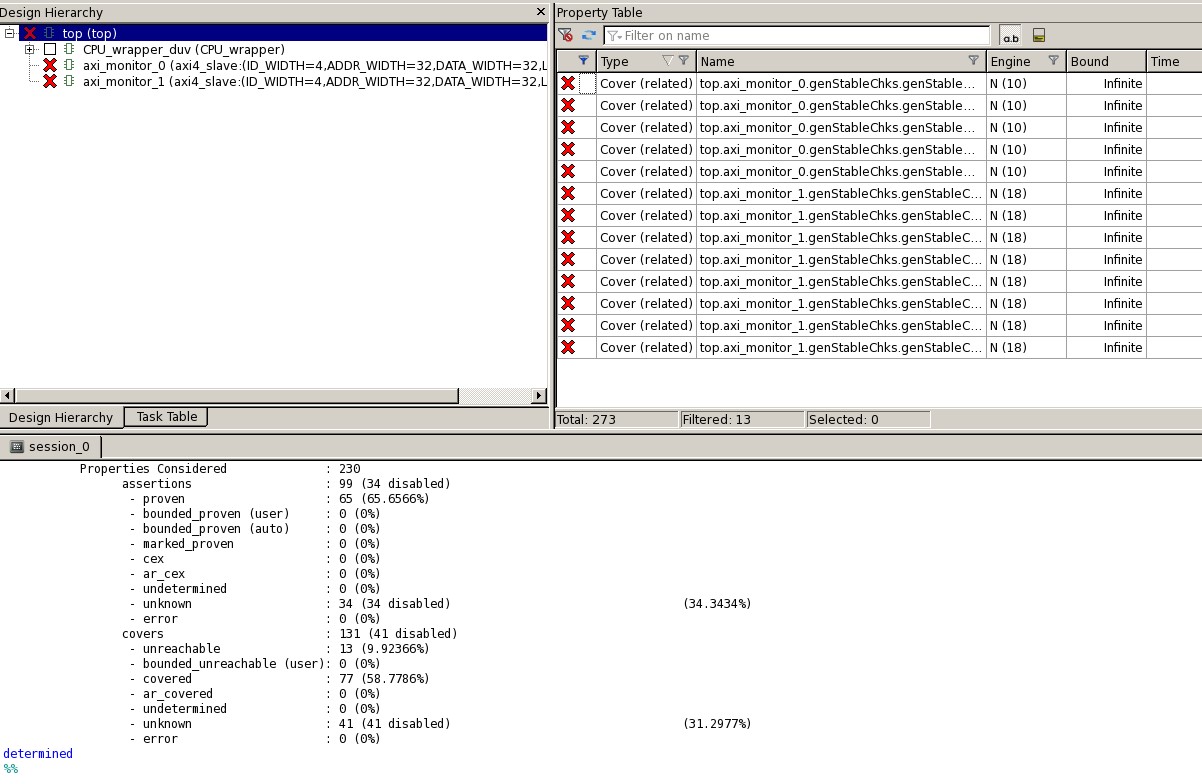
1. **Bridge**



1. **Slave**



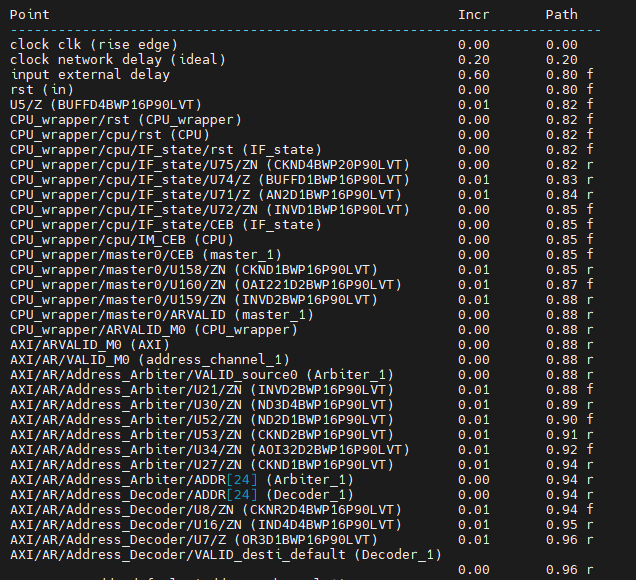
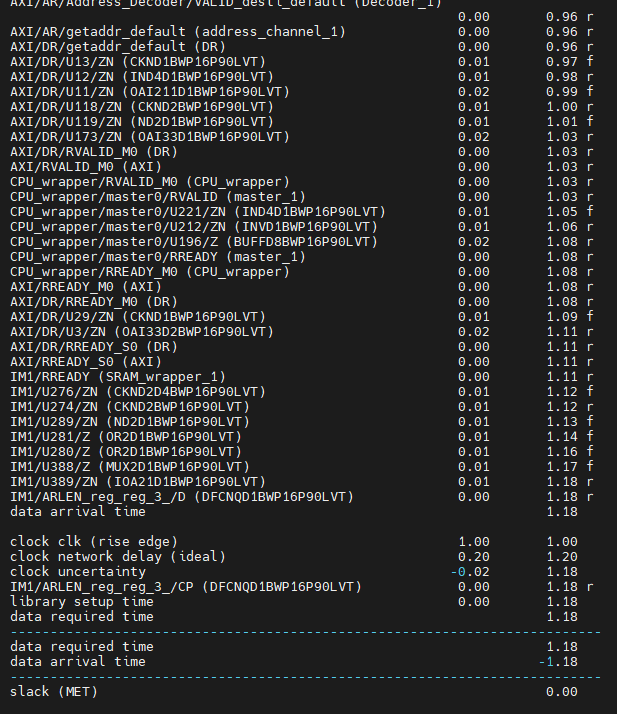
1. **Master**



在master驗證中有些cover沒有通過，其中分為2類，其一為RVALID & ~RREADY情況下的stable violation，其二為 BVALID & ~BREADY情況下的stable violation。會出現這兩種violation的原因為在我的設計中master一收到RVALID和BVALID訊號時會立即將RREADY和BREADY拉為1，故不會出現 RVALID & ~RREADY和BVALID & ~BREADY的條件。

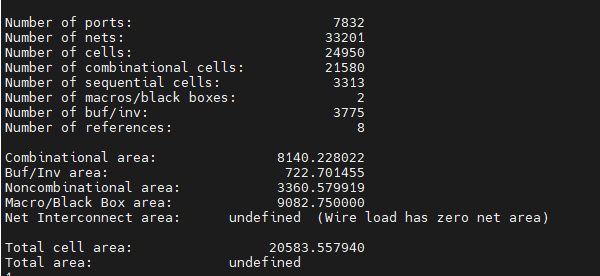
**Verify Synthesization**

**Timing**

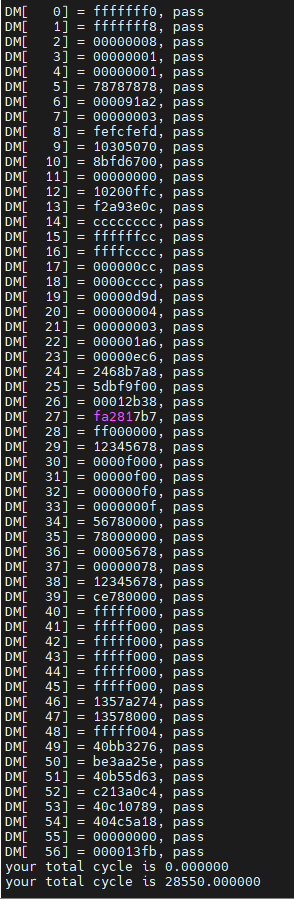
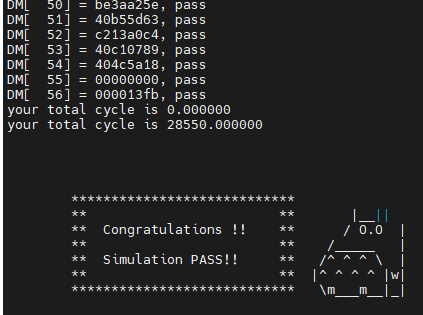
Cycle = 1.0

**Area**

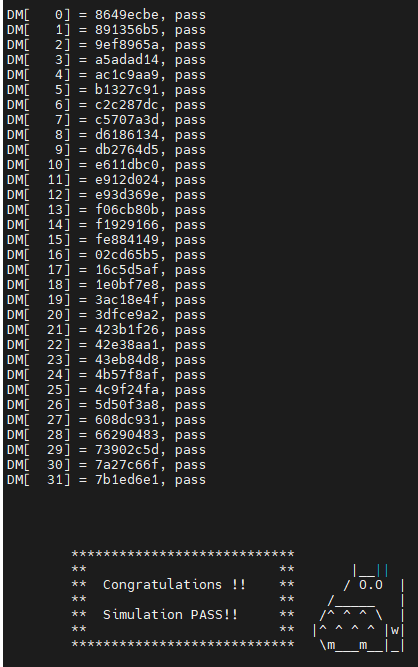


Area = 20583.557940

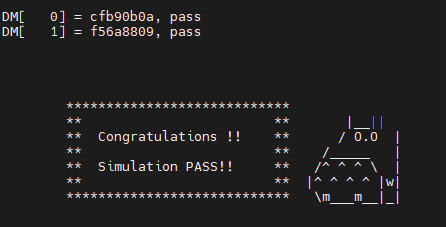
1. **Syn0**

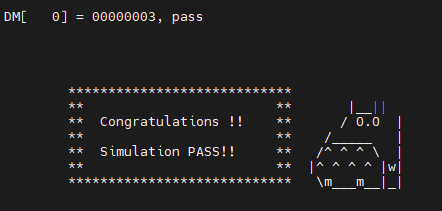
1. **Syn1**



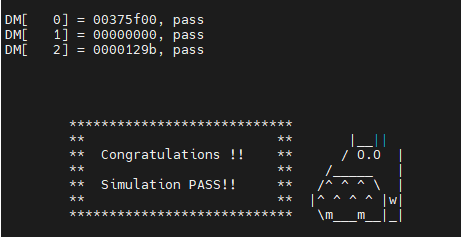
1. **Syn2**



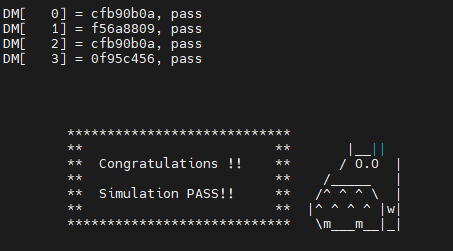
1. **Syn3**



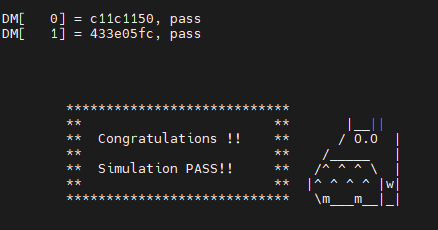
1. **Syn4**



1. **Syn5**



1. **Syn6**



**Lesson Learned**

1. 理解AXI運作原理，知道AXI中每個channel的功用。
2. 了解AXI如何進行handshake、資料如何傳輸，以及其中每個signal的具體作用。
3. 知道CPU如何合併AXI並成功運行。
4. 了解CPU在等待memory回應時如何進行stall。
5. 熟悉JasperGold VIP使用，了解cover、assert差別，知道如何透過檢查規範修正code。