

IC Lab Formal Verification
Bonus Report 2024 Fall

Name: 李曉旻

Student ID: 313552025

Account: iclab162

(a) What is Formal verification?

What's the difference between **Formal** and **Pattern** based verification?

And list the pros and cons for each.

Formal verification 是一種基於數學方法的驗證技術，用於檢查 Design 是否滿足規範，無需測試數據或模擬，通過工具使用窮舉法來驗證每一個 state 是否都正確來進行驗證，它可靠度較高但需要更多的運算，相較費時。

而 Pattern verification 則是依賴手動生成測試模式或自動生成的測資，進行模擬驗證，執行速度較快，但是有可能會有 Corner Case 無法測出來的情況，可靠度較低。

(b) Explain SVA (SystemVerilog Assertions) and the roles of Assertion, Cover, and Assumption.

What is glue logic?

Why will we use **glue logic** to simplify our SVA expression?

SystemVerilog Assertions 是一種用於設計驗證的語言，主要用來描述和檢查數位電路設計的時序與功能行為。SVA 提供以下三個 roles:

1. Assertion : 用於驗證設計是否滿足指定的條件或約束。
2. Cover : 用於檢查測試是否涵蓋了所有預期的情況，像是 cover property (c); 確保 c 的條件至少被觸發一次。。
3. Assumption: 用於為形式驗證或模擬提供假設條件，限制輸入信號的範圍或行為。

Glue logic 指在 design 當中為了簡化驗證而加入的 auxiliary logic，通常與結果輸出無關所以並不會被合成。如果 SVA 的條件過於複雜，直接編寫會難以閱讀或維護，Glue Logic 可以將複雜條件分解為幾個簡單的中間信號，然後使用這些信號撰寫 Assertion，透過 Glue Logic，SVA 表達式能更加直觀簡潔，且在驗證過程中容易追蹤信號行為和分析錯誤原因。

(c) What is the difference between **Functional coverage** and **Code coverage**?

What's the meaning of 100% code coverage, could we claim that our assertion is well enough for verification? Why?

Functional coverage 通常用於檢查 design functionality，且須由設計者自行定義 coverage 的規則，而 Code coverage 通常為 tool 自動產生，可以檢查我們的 RTL code 裡面是否都有成功被執行。

100% Code Coverage 表示 RTL 程式碼中的每一行、每一條分支和條件都在模擬過程中被執行過。但是即使有 100 % code coverage 也不代表驗證已經完備，因為 Code Coverage 只反映程式碼執行的情況，並不檢查執行結果是否正確，Assertion 的完備性需進一步確認，透過功能覆蓋確保設計行為符合需求。

(d) What is the difference between **COI coverage** and **proof coverage** for realizing checker's completeness? Try to explain from the meaning, relationship, and tool effort perspective.

COI Coverage 包括所有可能與assertion相關的cover items，而 Proof Coverage 聚焦於真正影響assertion的cover items，是 COI Coverage 的子集合。且Proof Coverage 需要執行 Formal Verification，所以tool effort需求較高，但結果更準確，有助於確認檢查器的完整性。

(e) What are the roles of **ABVIP** and **scoreboard** separately?

Try to explain the definition, objective, and the benefit.

ABVIP 是 Agent-Based 的驗證知識產權 (VIP, Verification IP), 為 checker 組成的 IP, 常用於 protocol 的檢查，由於一個 protocol 可能需要大量的 assertion 來檢查，而且相同 protocol 也可以重複使用相同的 assertion，所以就產生了 ABVIP 可以加速設計者驗證確保數據的傳輸符合協議規範。

scoreboard 就像是一個 monitor，提供準確的結果比較機制，即時檢查輸入、輸出的訊號是否正確，可以加速設計者驗證自己的電路功能是否有誤。

(f) Among the JasperGold tools (Formal Verification, SuperLint, Jasper CDC, IMC Coverage), which one do you think is the most effective based on its functionality and typical application scenarios? Please explain your reasoning by describing a hypothetical scenario where this tool would be particularly beneficial, and discuss any potential challenges or limitations that might arise when using it.

我認為 Formal Verification 是一個令我印象深刻且具啟發性的驗證工具。在修課過程中，常常需要自己撰寫 PATTERN 來進行驗證，這雖然能幫助我熟悉設計邏輯，但偶爾會因忽略 Corner Case 而導致 2 demo。這讓我體會到，傳統模擬雖然透過數百萬次測試激勵可以覆蓋設計的大部分狀態，但仍難以避免漏掉一些 Corner Case 或潛在問題。而 Formal Verification 也特別在於，透過數學方法全面分析所有可能狀態，不僅省去 stimulus 生成的繁瑣過程，更能精確捕捉設計中的潛在缺陷，大幅提升驗證的完整性與效率。