

IC LAB FORMAL VERIFICATION

BONUS QUICK TEST

2022 FALL

Name: 黃旻濤Student ID: 311510156Account: iclab002

1. Bonus:

(a) What is Formal verification?

Formal verification 是使用數學方法驗證 design 正確性的過程，使用各種演算法來驗證設計，想辦法覆蓋所有 state space 並找出 bug。其透過 Property 檢查，assertion 確認正確性，coverage 確認 state 的覆蓋程度。

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

Formal based verification 與 pattern based verification 的差異：

- (i) Pattern based 需要 testbench 來驗證電路行為而 formal based 不用
- (ii) Pattern based 一次測試一種 state path，而 formal based 會窮舉所有可能的 state path 進行測試

And list the pros and cons for each.

	Formal based verification	Pattern based verification
Pros	驗證完整	快速覆蓋 common case
Cons	如果 design 太大，覆蓋所有 state space 會很耗時	較難覆蓋到 corner case

(b) What is glue logic?

Glue logic 是運用額外的邏輯描述來追蹤與紀錄事件和狀態，使原本複雜難懂的邏輯以簡化的形式寫出來的通稱。

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

使用 glue logic 的原因，是因為它可以大幅簡化 coding 的複雜度，使 code 變得好讀好理解。我們可以將複雜的 property 透過 glue logic 表示出來，取代原先的 SVA

operator，增加 code 可讀性。

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

以下列出 functional coverage 與 code coverage 的差異：

Functional coverage 包含了 property 與 covergroup 的驗證，是人為設定的，所以會耗時，而且有可能會漏掉 corner case。Code coverage 包含 branch coverage、statement coverage、expression coverage 等，可以自動產生並執行，負責檢查每行 code/每個 branch/每個 statement 是否被執行，可以排除人為因素的 error，但也有可能有 false alarm 的出現(沒用的 code)。

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

不行。這只是說明每行 code 都有執行到，但是結果沒有被檢查，所以並不代表 function 上沒有 error。所以仍需進行 functional coverage 的驗證才能確保 assertion 的完整性。

(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.

Meaning perspective:

那些在 design 中影響 assertion 結果的 code，被稱為 COI，是一個在 DUT 裡面的範圍。Proof coverage 則是利用 formal engine 來驗證，只會包含實際影響到的部分。

Relationship perspective:

Proof coverage 是 COI 的 subset。

Tool effort perspective:

Proof coverage 需要進行 formal proof 而 COI 則不用。因此 proof coverage 需要花費較長的時間與較多的資源。

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

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

Definition:

ABVIP: The Assertion Based Verification Intellectual Properties(ABVIPs) are a comprehensive set of checkers and RTL that check for protocol compliance(e.g. ARM AMBA AXI based systems and designs).

Scoreboard: like a monitor, which observes input data and output data of DUV.

Objective:

ABVIP: 提供一個完整的 checker set，對於現有的 protocol 進行驗證與檢查。

Scoreboard: 檢查 design 的輸入與輸出是否一致。

Benefit:

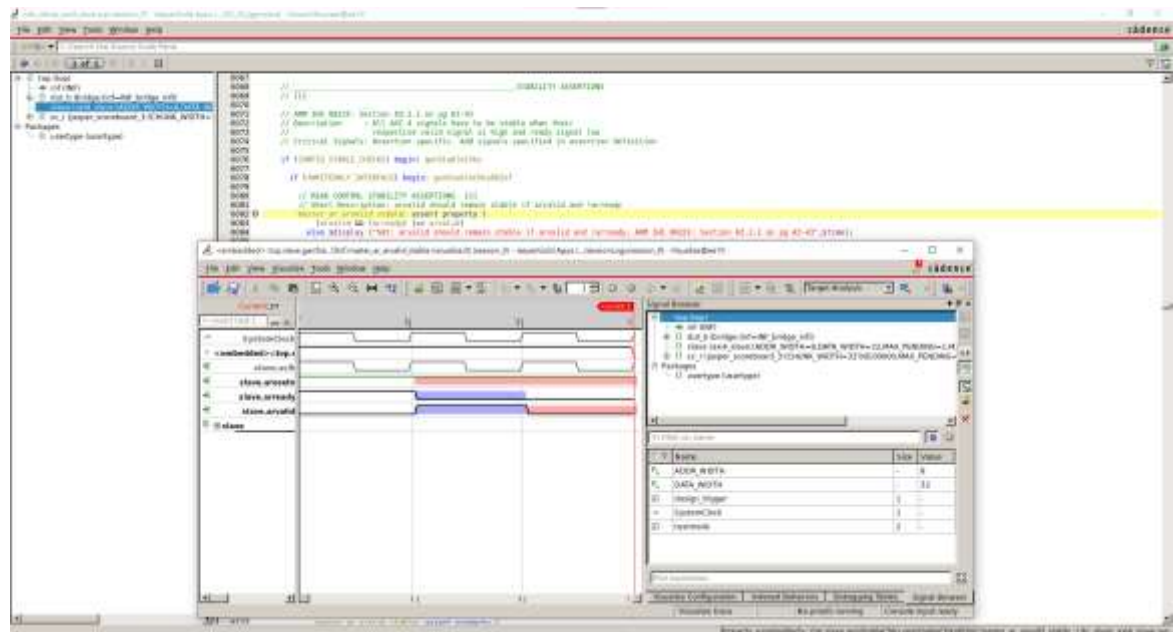
ABVIP: 節省 designer 寫 assertion 的時間，並提供正確且完整的驗證工具。

Scoreboard: 減少 state space 複雜度，減少 adoption 障礙。

(f) List four bugs in Lab Exercise

What is the answer of the Lab Exercise?

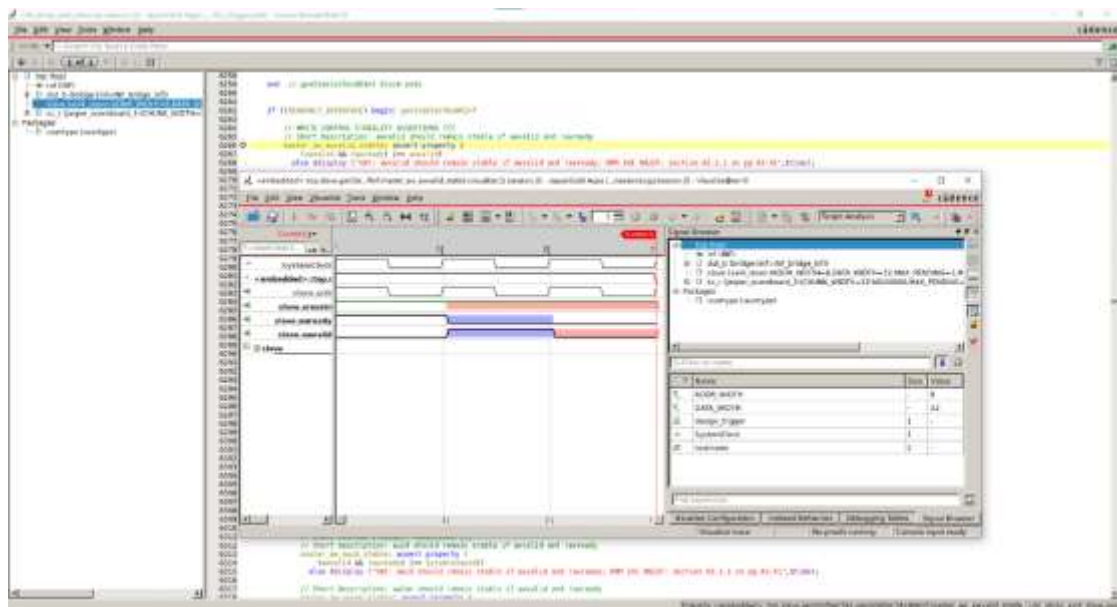
Bug1: inf.AR_VALID



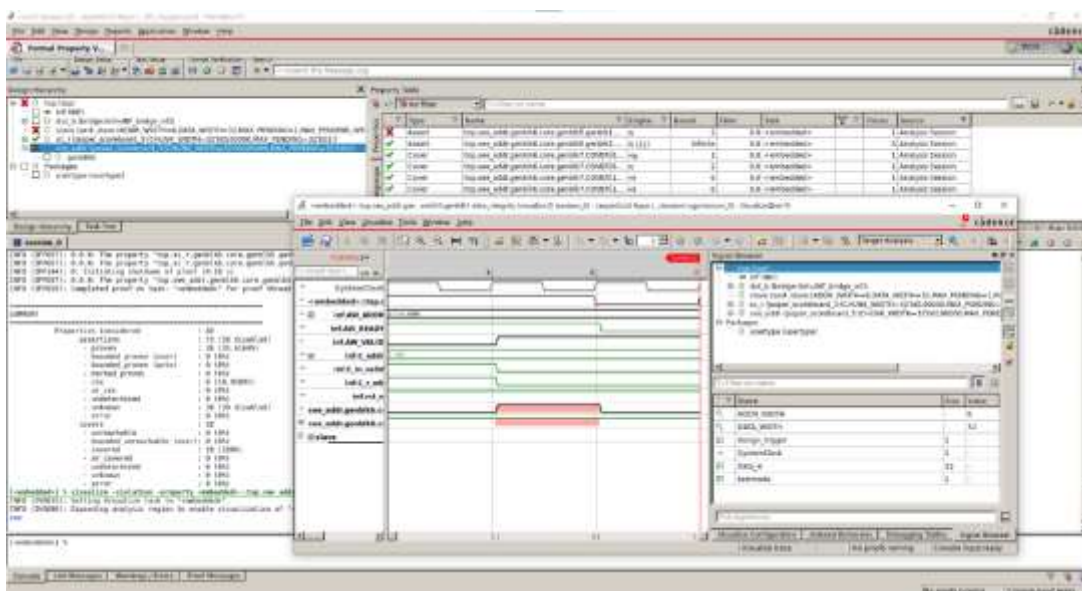
修正之後的 code 如下：

```
always_ff@(posedge clk or negedge inf.rst_n) begin
    if(!inf.rst_n)begin
        inf.AR_VALID <= 'b0;
    end
    else begin
        //if(inf.AR_READY) inf.AR_VALID <= 1'b1;
        if(n_state == AXI_AR) inf.AR_VALID <= 1'b1;
        else
            inf.AR_VALID <= 1'b0;
    end
end
```

Bug2: inf.AW_VALID



Bug3: inf.AW_ADDR



利用 scoreboard 找出 bug 後的修正版本的 code 如下：

```
always_ff@(posedge clk or negedge inf.rst_n) begin
    if(!inf.rst_n)begin
        inf.AW_ADDR <= 'b0;
    end
    else begin
        //if(n_state == AXI_AW && c_state != AXI_AW) {inf.AW_ADDR <= {8'h1000_0000, inf.C_addr, 2'b0};
        if(n_state == AXI_AW && c_state != AXI_AW) inf.AW_ADDR <= {1'b1, 7'b0, inf.C_addr, 2'b0};
        else
            inf.AW_ADDR <= inf.AW_ADDR ;
    end
end
```

```
jasper_scoreboard_3 #(
    .CHUNK_WIDTH(32),
    .SINGLE_CLOCK(1),
    .ORDERING(`JS3_IN_ORDER),
    .MAX_PENDING(1)
)see_addr(
    .clk(SystemClock),
    .rstN(inf.rst_n),
    .incoming_vld(inf.C_in_valid && ~inf.C_r_wb),
    .incoming_data({8'b1000_0000, inf.C_addr, 2'b0}),
    .outgoing_vld(inf.AW_VALID && inf.AW_READY),
    .outgoing_data(inf.AW_ADDR)
);
```

Bug4: inf.W_DATA



利用 glue logic+SVA 找出 bug 後的修正版本的 code 如下：

```
always_ff@(posedge clk or negedge inf.rst_n) begin
    if(!inf.rst_n)begin
        inf.W_DATA <= 'b0;
    end
    else begin
        //if(inf.C_in_valid && inf.C_r_wb)      inf.W_DATA <= inf.C_data_w;
        if(inf.C_in_valid && !inf.C_r_wb)      inf.W_DATA <= inf.C_data_w;
        else
            inf.W_DATA <= inf.W_DATA ;
        end
    end
end
```

```
//method 1
logic [31:0] data_w;
always_ff@(posedge SystemClock) begin
    if(inf.C_in_valid && ~inf.C_r_wb) begin
        data_w <= inf.C_data_w;
    end
end

assert_w_data: assert property(p_w_data)
else begin
    $display("assert_w_data is violated");
end

property p_w_data;
    @(posedge SystemClock) inf.W_VALID && inf.W_READY |-> data_w == inf.W_DATA;
endproperty: p_w_data
```

(g) What is your opinion about this class ? (at least 50 words) (you can write anything !!!!)

謝謝助教們對於這門課程的付出與努力，讓我們可以通過紮實的課堂講解與每周 lab 的訓練，得到充分的對於 IC 設計的知識與應有的技能。雖然這學期過得非常辛苦，每周一次的 lab 輪迴搞得我壓力山大，但通過這學期的考驗與磨練我覺得我不只是得到學業方面的增長，在心理層面、抗壓的能力也有所提升。最後也謝謝李鎮宜老師幾次的課前提點與勉勵，讓我有繼續前進的動力。