IC Lab Formal Verification Bonus Quick Test

2022 Spring

**Name: 林哲宇 Student ID: A101085 Account: iclab108 y**

1. What is Formal verification?

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

And list the pros and cons for each.

**Ans:**

Formal verification is the process of verifying the correctness of the design, using formal methods of mathematics.

Formal verification doesn’t need to write any test vector, it uses various algorithms to verify the design without performing any timing checks. While Pattern based verification requires user-defined stimulus to check design properties.

|  |  |  |
| --- | --- | --- |
|  | **Pros** | **Cons** |
| **Formal verification** | Cost less time  Reliable  Faster  More detailed | Don’t care or duplicate covers |
| **Pattern based verification** | Noise free | Requires planning, coding, and debug  May not cover corner case  Time consuming |

1. What is glue logic?

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

**Ans:**

Glue logic is the custom logic circuitry used to interface several off-the-shelf integrated circuits.

Since modeling complex behaviors, SVA may be complicated. It may simplify coding with no extra price.

1. 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?

**Ans:**

Functional coverage is where you have verified all the scenarios that the design is to be used. While Code coverage means that you have verified all the lines of code in the design.

100% code coverage means the code is 100 percent bugless. No error indicates that test cases have covered every criterion and requirement of the software application. No, since code coverage in simulation means that every line in the RTL is executed during the test. But executed does not necessarily mean correct.

1. 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.

**Ans:**

COI coverage is to find the region which each assertion affected by some cover items. While proof coverage is to find the region cannot truly influence assertion status.

COI doesn’t require a proof to take place, while proof coverage does.

Proof coverage is a subset of the COI, COI represents the maximum potential of proof coverage.

Proof coverage identify more unchecked code than COI measurement, with greater tool effort.

1. What are the roles of ABVIP and scoreboard separately?

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

**Ans:**

The Assertion Based Verification Intellectual Properties (ABVIPs) are a comprehensive set of checkers and RTL that check for protocol compliance. It treats like pattern that check interface signal to make sure the design satisfies protocol specification. Unlike logic simulation that uses test sequences to stimulate a desig, ABVIP no test creation is required and easy to debug.

Scoreboard behaves like a monitor, it observes input data and output data of DUV and check the input and the output is same or not.

1. List four bugs in Lab Exercise

What is the answer of the Lab Exercise?

**4 bugs:**

* 1. AR\_VALID should remain stable if AR\_VALID and !AR\_READY
  2. AW\_VALID should remain stable if AW\_VALID and !AW\_READY
  3. W\_DATA should be the same as C\_data\_w when C\_in\_valid and !C\_r\_wb
  4. AW\_ADDR is wrong addr

**Ans:**

Line 90: if(n\_state == AXI\_AR) inf.AR\_VALID <= 1'b1;

Line 124: if(n\_state == AXI\_AW) inf.AW\_VALID <= 1'b1;

Line 145: if(inf.C\_in\_valid && !inf.C\_r\_wb) inf.W\_DATA <= inf.C\_data\_w;

Line 134: inf.AW\_ADDR <= {8'b1000\_0000, inf.C\_addr, 2'b0};