

Nghiên cứu thiết kế và hiện thực mô hình kiểm tra cho một thiết kế Mạng nơ-ron tích chập

Ngô Thái Anh Hào, 16520347

Khoa Kỹ thuật máy tính, Trường Đại học Công Nghệ Thông tin

Trương Văn Cường

Hồ Ngọc Diễm

Khoa Kỹ thuật máy tính, Trường Đại học Công Nghệ Thông tin

Tóm tắt— UVM (Universal Verification Methodology) là một trong các phương pháp kiểm tra thiết kế mạnh mẽ nhất và được sử dụng rộng rãi trong quá trình thiết kế mạch tích hợp kỹ thuật số. Việc áp dụng UVM vào quá trình kiểm tra thiết kế giúp tăng hiệu quả kiểm tra và giảm thời gian xây dựng Testbench, đặc biệt đối với các thiết kế phức tạp. Nhóm quyết định hiện thực quá trình kiểm một thiết kế CNN (Convolution Neural Network) áp dụng UVM, thực hiện xây dựng môi trường kiểm tra theo hướng đối tượng, các phần tử của môi trường được xây dựng và liên kết với nhau thông qua thư viện tích hợp của UVM. Đồng thời quá trình kiểm tra được tối ưu về mặt tự động hoá bằng việc sử dụng Shell và TCL (Tool Command Language) Script, cũng như tính năng DPI (Direct Programming Interface) được tích hợp của SystemVerilog cũng được nhóm áp dụng nhằm xây dựng mô hình tin cậy từ ngôn ngữ C/C++ có khả năng giao tiếp với SystemVerilog Testbench trong quá trình mô phỏng.

Từ khóa—UVM, CNN, TCL, DPI

I. GIỚI THIỆU

Trong quá trình thiết kế một mạch RTL, quá trình đánh giá và kiểm tra là một trong những nhân tố then chốt trong việc xác định thiết kế có hoạt động như mong đợi hay không, đồng thời xác định được mức độ hoàn thiện, tính đúng đắn và độ tin cậy của thiết kế.

Mục tiêu chính của việc kiểm tra thiết kế là tìm lỗi của thiết kế đó, quá trình kiểm tra được hiện thực bằng cách đưa các trường hợp đầu vào khác nhau và xác định kết quả đầu ra có chính xác hay không. Độ tin cậy của thiết kế phụ thuộc vào số lượng mẫu thử đầu vào và mức độ chính xác của đầu ra. Do độ phức tạp của quá trình kiểm tra tỉ lệ thuận với độ phức tạp của thiết kế, đặc biệt với các thiết kế như SoC, các phương pháp kiểm tra thiết kế được các tập đoàn lớn lần lượt

được phát triển sau sự ra đời của ngôn ngữ SystemVerilog: eRM, RVM, VMM, AVM, OVM và UVM [6]. Các phương pháp này tận dụng điểm mạnh của ngôn ngữ SystemVerilog là áp dụng lập trình hướng đối tượng vào quá trình xây dựng Testbench, hỗ trợ người kiểm tra thiết kế có thể tối ưu tính tự động hoá và khả năng tái sử dụng các phần tử có sẵn bên trong Testbench, giúp tăng hiệu quả và giảm thiểu thời gian kiểm tra. Trong các phương pháp kiểm tra trên, nổi bật nhất chính là UVM (Universal Verification Methodology) được xem như phương pháp kế thừa điểm mạnh của các phương pháp tiền thân [5], đồng thời được chuẩn hoá và liên tục phát triển bởi Accellera từ 2011 tới nay và được sử dụng rộng rãi trên thế giới ở thời điểm hiện tại. UVM cung cấp cho người dùng một thư viện các lớp có khả năng tự động hoá và tích hợp các tính năng tiện ích hỗ trợ người kiểm tra trong quá trình xây dựng Testbench.

Bên cạnh việc sử dụng UVM vào việc xây dựng môi trường kiểm tra, áp dụng Shell và TCL Scripts cũng hỗ trợ tăng hiệu quả và giảm thời gian cho quá trình kiểm tra thông qua khả năng tự động hoá của Scripts. Đồng thời, một mô hình tin cậy của thiết kế được xây dựng dựa trên ngôn ngữ C và có khả năng giao tiếp với UVM Testbench thông qua DPI-C của SystemVerilog giúp việc xác định tính đúng đắn và độ tin cậy của thiết kế được tối ưu hơn.

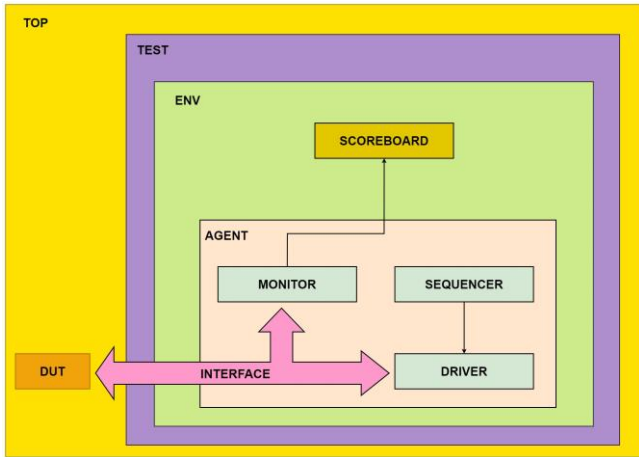
II. GIẢI PHÁP

A. Xây dựng môi trường kiểm tra UVM

Một môi trường UVM cơ bản bao gồm các phần tử được xây dựng bởi các lớp tích hợp sẵn bên trong thư viện UVM, mỗi phần tử bên trong môi trường có một chức năng riêng biệt.

Các thành phần chính của UVM Testbench bao gồm: `uvm_sequence_item`, `uvm_sequence`, `uvm_driver`, `uvm_monitor`, `uvm_scoreboard`, `uvm_agent`, `uvm_env`,

uvm_test và interface. Các thành phần này có vai trò chính là tạo ra các gói tin để đưa vào DUT (Design Under Test), vận chuyển các gói tin đầu vào và thu thập các gói tin chứa kết quả đầu ra, đánh giá kết quả nhận được trong quá trình mô phỏng và xuất báo cáo kết quả thu được sau khi mô phỏng. Cấu trúc vào liên kết của các phần tử được thể hiện qua hình bên dưới:



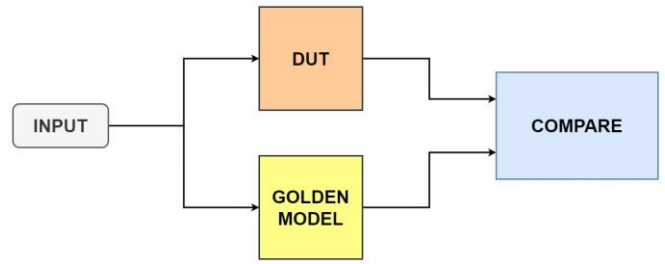
Hình 1: Mô hình UVM Testbench

Các phần tử bên trong môi trường UVM Testbench được liên kết và giao tiếp với nhau thông qua giao thức TLM (Transaction Level Modeling), đây cũng là một giao thức tích hợp của thư viện UVM. UVM Testbench và DUT được liên kết với nhau thông qua Interface.

B. Xây dựng mô hình tin cậy cho thiết kế

Một mô hình tin cậy của thiết kế được xây dựng dựa trên ngôn ngữ C, đồng thời việc tiền xử lý dữ liệu cho mô hình được xây dựng trên C++ có sử dụng thư viện OPENCV. Mô hình tin cậy này được dùng để so sánh với kết quả có được từ quá trình mô phỏng để đánh giá tính đúng đắn và mức độ tin cậy của thiết kế. Để ngôn ngữ C/C++ có thể giao tiếp với SystemVerilog, DPI-C được sử dụng, đây là một tính năng của SystemVerilog cho phép SystemVerilog Testbench gọi các hàm viết bằng ngôn ngữ C/C++ trong quá trình mô phỏng. Code C/C++ sẽ được trình biên dịch tổng hợp để có được một thư viện dưới dạng tệp dpi.so, cho phép UVM Testbench có thể gọi hàm C/C++ trong quá trình mô phỏng.

Cùng truyền một tập đầu vào tới DUT và mô hình tin cậy, sau khi có được kết quả từ DUT, kết quả này sẽ được so sánh với kết quả từ mô hình tin cậy để đánh giá kết quả có đạt hay không. Đồng thời kết quả của quá trình mô phỏng sẽ được xử lý và xuất ra file báo cáo.



Hình 2: Đánh giá kết quả mô phỏng

III. KẾT QUẢ

Nhóm đã xây dựng thành công môi trường kiểm tra hoàn chỉnh cho dựa trên phương pháp và cấu trúc UVM cho thiết kế tích chập của LeNet5 và ShuffleNet IP. Môi trường kiểm tra chứa đầy đủ các thành phần và chức năng cơ bản của một UVM Testbench. Nhóm cũng tích hợp thành công chức năng kiểm tra hành vi tín hiệu bên trong DUT sử dụng SVA và ứng dụng DPI vào việc xây dựng mô hình tin cậy cho thiết kế CNN IP. Đồng thời, các thành phần bên trong môi trường kiểm tra cũng được tham số hóa để có thể tối ưu tính tái sử dụng của môi trường.

IV. KẾT LUẬN

UVM là một phương pháp kiểm tra mạnh mẽ, nhờ việc ứng dụng UVM vào xây dựng môi trường kiểm tra cho một thiết kế phức tạp như CNN IP sẽ hỗ trợ người kiểm tra thiết kế tăng hiệu suất và kết quả kiểm tra, với điểm mạnh chính đó là thư viện tích hợp, tối ưu khả năng tự động hóa và tính tái sử dụng môi trường. Cùng với các chức năng và cơ chế như DPI và SVA của SystemVerilog mà quá trình kiểm tra thiết kế trở nên khách quan và chính xác hơn.

V. TÀI LIỆU THAM KHẢO

- [1] Universal Verification Methodology (UVM) 1.2 Class Reference; Accellera Systems Initiative (Accellera); June 2014
- [2] Universal Verification Methodology (UVM) 1.2 User's Guide; Accellera Systems Initiative (Accellera); June 2014
- [3] SystemVerilog for Verification: A Guide to Learning the Testbench Language 3rd Edition; Chris Spear, Synopsys, Inc; 2012
- [4] C-based Predictor for Scoreboard in Universal Verification Methodology; Srikanth Konale, N.Bheema Rao; IEEE International Conference on Advances in Engineering & Technology Research (ICAETR – 2014); 2014
- [5] An Introduction to Universal Verification Methodology for the digital design of Integrated circuits (IC's); A Review; N B Harshitha, Praveen Kumar Y G, M Z Kurian; 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS); 2021
- [6] Universal Verification Methodology – UVM Cookbook; SEIMEN Verification Academy
- [7] UVM for Verification; Kumar Khandagle (Udemy)