

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỀ TÀI TIẾNG VIỆT:
TÊN ĐỀ TÀI TIẾNG ANH:
Cán bộ hướng dẫn (Học vị, họ và tên, nơi công tác cán bộ hướng dẫn và cán bộ hướng dẫn phụ nếu có): ThS. Trương Văn Cường, Giảng viên ĐHQG TPHCM - Trường Đại học Công nghệ Thông tin
Thời gian thực hiện: Từ ngày 11/03/2024 đến ngày28/06/2024
Sinh viên thực hiện: Ngô Thái Anh Hào - 16520347
<p>Nội dung đề tài:<i>(Mô tả chi tiết về tổng quan đề tài, mục tiêu, phạm vi, đối tượng, phương pháp thực hiện, kết quả mong đợi của đề tài,...)</i></p> <p>Tổng quan đề tài <i>(sinh viên cần nêu rõ các đề tài, sản phẩm liên quan đã được nghiên cứu hoặc đã có trên thị trường trước thời điểm hiện tại và nêu thực trạng của chúng, để từ đó nêu bật được lý do thực hiện nghiên cứu trong KLTN này):</i></p> <p>Trong quá trình thiết kế một mạch tích hợp kỹ thuật số, bên cạnh bước lên ý tưởng và thiết kế thì một trong những bước quan trọng và chính yếu đó chính là quá trình kiểm tra chức năng của mạch. Kiểm tra chức năng của mạch là việc cần làm ngay sau khi người kỹ sư đã hoàn thành việc hiện thực thiết kế thông qua ngôn ngữ đặc tả phần cứng. Ở bước này, mục tiêu của việc kiểm tra đó chính là tìm lỗi, người kiểm tra sẽ phải tiến hành xây dựng những kịch bản khác nhau để kiểm tra chức năng của mạch có thực hiện đúng theo mong muốn ban đầu đã được đề ra hay không. Và đối với các thiết kế càng phức tạp thì việc kiểm tra cũng trở nên khó khăn hơn, vì lí do đó mà ngôn ngữ SystemVerilog được ra đời vào năm 2002 với khả năng đưa lập trình hướng đối tượng testbench hỗ trợ một cách hiệu quả hơn trong quá trình kiểm tra cho người kiểm tra thiết kế. Sau đó, các phương pháp kiểm tra thiết kế dựa trên ngôn ngữ SystemVerilog lần lượt ra đời và phát triển để tăng hiệu quả của quá trình kiểm tra như eRM, RVM, VMM, AVM, OVM và UVM. Trong đó, UVM (Universal Verification Methodology) là một phương pháp kiểm tra thiết kế ra đời vào năm 2009 và được chuẩn hóa vào năm 2011 bởi Accellera. Sử dụng ngôn ngữ SystemVerilog, UVM cung cấp một thư viện các lớp có khả năng tự động hóa và tích hợp các tính năng tiện ích hỗ trợ người kiểm tra trong quá trình viết Testbench. Không như các phương pháp tiền thân được phát triển độc lập bởi các công ty cung cấp phần mềm mô phỏng, UVM là một phương pháp kiểm tra thiết</p>

kế đã được chuẩn hóa bởi Accellera - một tập đoàn được hỗ trợ bởi nhiều công ty khác nhau như: Aldec, Cadence, Mentor Graphics, Synopsys, Xilinx Simulator. Nhờ lý do này mà UVM được sử dụng một cách rộng rãi trên khắp thế giới. Điểm mạnh chính của UVM đó chính là các thư viện tích hợp sẵn, giúp người kỹ sư có thể gọi trực tiếp để sử dụng trong quá trình xây dựng môi trường kiểm tra thiết kế, đồng thời các thư viện này cũng tối ưu hóa việc tự động hóa và tái sử dụng các thành phần cũng như môi trường Testbench trong tương lai. Từ đây, người kỹ sư kiểm tra thiết kế có thể dễ dàng đọc và hiểu được Testbench do một người khác xây dựng, giúp tăng hiệu quả trong công việc. Đồng thời, người kỹ sư sẽ không cần phải xây dựng lại môi trường Testbench từ đầu mỗi khi bắt đầu việc kiểm tra một thiết kế mới. UVM cũng cung cấp chức năng ghi đè lên các lớp trong môi trường kiểm tra, giúp tăng hiệu quả việc tái sử dụng trong quá trình kiểm tra thiết kế, đặc biệt đối với việc kiểm tra các thiết kế phức tạp như SoC (System on Chip). Nhờ các điểm mạnh trên mà UVM đã trở thành một trong những phương pháp kiểm tra thiết kế mạnh mẽ hàng đầu và được sử dụng một cách rộng rãi trên thế giới từ khi nó ra đời đến nay.

Mục tiêu của đề tài (Nêu cụ thể mục tiêu của KLTN, đặc biệt phải nêu được mục tiêu cải tiến sẽ là gì so với thực trạng nêu trong phần tổng quan đề tài):

Thực hiện kiểm tra thành công một thiết kế CNN IP có sẵn sử dụng phương pháp UVM. Xây dựng và viết một môi trường UVM Testbench hoàn chỉnh để thực hiện việc kiểm tra thiết kế, độ hoàn chỉnh của UVM Testbench được đánh giá qua các tiêu chí:

- Có đầy đủ các thành phần kiểm tra tiêu chuẩn (uvm_sequence, uvm_driver, uvm_sequencer, uvm_monitor,...)
- Các phần tử trong môi trường kiểm tra có khả năng tái sử dụng
- Có thể phân chia để kiểm tra từng phần tử trong thiết kế, mỗi submodule của thiết kế CNN đều có phần tử kiểm tra riêng trong môi trường UVM Testbench
- Có khả năng đánh giá độ chính xác của IP thông qua môi trường UVM Testbench đã viết với nhiều các trường hợp khác nhau (bao gồm trường hợp thông thường và trường hợp góc) và thực hiện thu thập functional coverage.

Phương pháp thực hiện (Nêu tổng quan phương pháp thực hiện):

Sử dụng công cụ mô phỏng và thư viện UVM có sẵn của phần mềm Vivado để kiểm tra một thiết kế CNN IP được cung cấp bởi giảng viên hướng dẫn, thiết kế được viết bằng ngôn ngữ Verilog HDL và UVM Testbench được viết bằng SystemVerilog.

Xây dựng môi trường UVM riêng dành cho thiết kế CNN bằng cách tạo các lớp

con từ thư viện tự động của UVM. Các lớp trong một môi trường kiểm tra UVM bao gồm:

- `uvm_sequence_item`: là các gói tin mang thông tin di chuyển trong môi trường kiểm tra thiết kế.
- `uvm_sequence`: là một chuỗi các gói tin, mỗi chuỗi thường mang một hành vi riêng được quy định bởi người thiết kế môi trường kiểm tra với một mục đích cụ thể, Testbench có thường có nhiều chuỗi.
- `uvm_sequencer`: là thành phần có chức phân xử các chuỗi trong môi trường kiểm tra thiết kế.
- `uvm_driver`: là thành phần có chức năng lái các gói tin có ý nghĩa vào DUT thông qua giao thức interface.
- `uvm_monitor`: là thành phần có chức năng lấy kết quả xử lý của DUT sau khi được lái gói tin vào để kiểm tra, kết quả lấy được thông qua giao thức interface, sau đó sẽ được đóng gói lại thành một gói tin và gửi tới các thành phần khác trong môi trường kiểm tra.
- `uvm_agent`: là một thành phần chứa các thành phần khác bao gồm: `uvm_sequencer`, `uvm_driver` và `uvm_monitor`. Đối với một thiết kế có nhiều submodule cần được kiểm tra thì việc kiểm tra có thể được chia ra các agent khác nhau tương ứng với từng submodule cần được kiểm tra.
- `uvm_scoreboard`: là thành phần có chức năng so sánh kết quả mong đợi với kết quả thực tế trong quá trình kiểm tra, cũng là thành phần thu thập functional coverage.
- `uvm_env`: là thành phần đóng gói tất cả các thành phần con của môi trường kiểm tra thiết kế.

Các nội dung chính và giới hạn của đề tài (Nêu nội dung thực hiện và phương pháp dự kiến để đánh giá hệ thống, ví dụ, nếu KLTN có chạy demo thì hệ thống demo sẽ như thế nào):

Các nội dung thực hiện dự kiến:

- Đọc thiết kế CNN bao gồm chức năng của thiết kế, các input và output của thiết kế.
- Xây dựng các thành phần cơ bản cho một môi trường UVM Testbench bao gồm: sequence, sequencer, driver, monitor, agent, scoreboard, env, test.
- Kết nối các thành phần trên với nhau theo quy ước của UVM.
- Kết nối các thành phần trong testbench với nhau, bao gồm UVM Testbench,

Interface và DUT (Design Under Test).

- Tiến hành chạy mô phỏng với các trường hợp khác nhau, bao gồm các trường hợp ngẫu nhiên được tạo ra bởi UVM Testbench và các trường hợp đặc biệt và trường hợp góc do người dùng cung cấp.
- Kiểm tra kết quả và thu thập functional coverage bằng UVM Testbench.
- Đánh giá kết quả thu tập được.
- Toàn bộ quá trình mô phỏng được thực hiện bởi trình mô phỏng và thư viện UVM của phần mềm Vivado.

Kế hoạch thực hiện: *(Mô tả kế hoạch làm việc và phân công công việc cho từng sinh viên tham gia)*

- **19/02/2024 - 25/02/2024 (tuần 1):** Nghiên cứu về kiến thức cơ sở của UVM: UVM Factory, UVM object and component, Giao tiếp TLM, Reporting Mechanism, UVM Macros,...
- **26/02/2024 - 10/03/2024 (tuần 2-3):** Thực hành xây dựng UVM Testbench cho mạch tổ hợp và mạch tuần tự, các giao thức bus thông dụng APB, AXI và các giao thức bus UART, SPI, I2C
- **11/03/2024 - 17/03/2024 (tuần 4):** Nghiên cứu về chức năng RAL - Register Absaction Layer của UVM. RAL được sử dụng để kiểm tra các phần tử nhớ như registers và memories.
- **18/03/2024 - 24/03/2024 (tuần 5):** Nghiên cứu về SystemVerilog Assertion
- **25/03/2024 - 08/04/2024 (tuần 6-7):** Đọc Specification của thiết kế CNN IP bao gồm chức năng và các cổng output input (các submodule nếu có), chuẩn bị kế hoạch kiểm tra cho thiết kế bao gồm các trường hợp ngẫu nhiên sử dụng ràng buộc, các trường hợp đặc biệt và các trường hợp góc.
- **09/04/2024 - 28/04/2024 (tuần 8 - tuần 11):** Tiến hành xây dựng UVM Testbench cho thiết kế CNN IP.
- **29/04/2024 - 12/05/2024 (tuần 8-9):** Chuẩn bị báo cáo tiến độ và slide thuyết trình khóa luận tốt nghiệp giữa kỳ trước Hội đồng
- **13/05/2024 - 16/06/2024 (tuần 10 - tuần 14):** Hoàn thiện Testbench và tiến hành kiểm tra thiết kế CNN IP.
- **17/06/2024 - 23/06/2024 (tuần 15):** Viết báo cáo và slide thuyết trình Khóa luận tốt nghiệp trước Hội đồng

Xác nhận của CBHD

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày....thángnăm.....

Sinh viên

(Ký tên và ghi rõ họ tên)