

Thiết kế bộ điều khiển đồng bộ cache cho hệ thống đa vi xử lý dựa trên lõi CPU RISC-V RV32I

Nguyễn Quốc Trường An, 21521810

Dương Hoàng Tuấn, 21522750

Khoa Kỹ thuật Máy tính, Trường Đại học Công nghệ Thông tin

ThS. Phạm Minh Quân, KS. Trần Đại Dương

Khoa Kỹ thuật Máy tính, Trường Đại học Công nghệ Thông tin

Tóm tắt— Khóa luận thực hiện thiết kế phần cứng bộ điều khiển Cache Coherence theo kiến trúc 4-Way Set Associative trong hệ thống đa vi xử lý hỗ trợ 2 lõi, sử dụng giao thức Cache Coherence MOESI Snoopy để giải quyết vấn đề đồng nhất dữ liệu giữa các lõi trên kênh giao tiếp chung AXI Bus mở rộng ACE. Thiết kế lõi xử lý RISC-V RV32I pipeline 5 tầng tích hợp I-Cache và D-Cache để thực thi chương trình một cách hiệu quả. Bên cạnh đó, xây dựng mô hình kiểm thử hệ thống 2 vi xử lý bằng phần mềm, gồm 2 lõi xử lý RISC-V RV32I, 2 I-Cache, 2 D-Cache, AXI Bus với mở rộng ACE và bộ nhớ chính.

Từ khóa—Cache Coherence, 4-Way Set Associative Cache, RISC-V RV32I, Shared Memory Multiprocessors.

I. GIỚI THIỆU

Bộ nhớ đệm (Cache) là vùng lưu trữ dữ liệu tạm thời nhằm tăng tốc độ truy xuất dữ liệu từ CPU [1]. Trong hệ thống đa vi xử lý, mỗi lõi xử lý đều có một bộ nhớ đệm cho riêng mình và cùng chia sẻ một bộ nhớ chính. Khi các lõi cùng truy cập dữ liệu ở vùng nhớ chia sẻ chung, việc chỉnh sửa dữ liệu có thể dẫn đến sự khác biệt về góc nhìn đối với dữ liệu giữa những bản sao tại các bộ nhớ đệm riêng. Để giải quyết vấn đề này, Cache Coherence là giải pháp cấp thiết giúp đồng nhất dữ liệu giữa các bộ nhớ đệm trong hệ thống đa vi xử lý.

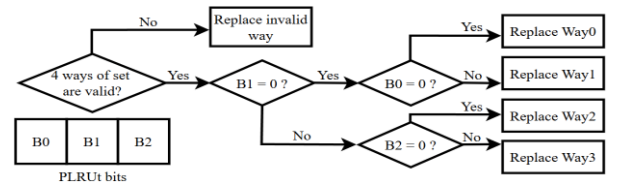
Cho đến nay, Cache Coherence trong hệ thống đa vi xử lý là vấn đề phổ biến với nhiều phương pháp giải quyết khác nhau. Trong đó có thể kể đến như phương pháp Directory-based giúp xây dựng bộ danh mục lớn có thể lưu trữ trạng thái của toàn bộ dữ liệu với lượng lớn lõi xử lý, hay phương pháp sử dụng các giao thức Snooping có thể cung cấp kênh giao tiếp chung nhằm xử lý và cập nhật trạng thái dữ liệu kịp thời với số lượng nhỏ các lõi xử lý. Nhận thấy hệ thống phù hợp với việc sử dụng giao thức Snooping với 2 vi xử lý cũng như hiện nay giao thức AXI đã có hỗ trợ nhóm kênh snooping [2], khóa luận này lựa chọn thiết kế giao thức MOESI Snoopy để giải quyết Cache Coherence dùng kênh giao tiếp chung là AXI Bus với mở rộng ACE. Bên cạnh đó, đề tài cũng thiết kế các thành phần, bao gồm: bộ nhớ đệm lệnh với kiến trúc 4-Way Set Associative, lõi xử lý RISC-V RV32I với kiến trúc pipeline 5 tầng có tích hợp AXI Wrapper, AXI Bus mở rộng ACE giúp giao tiếp giữa các thành phần và bộ nhớ chính là một IP BRAM có sẵn của Xilinx. Để hoàn thiện kiểm thử thiết kế, khóa luận còn xây dựng thêm mô hình kiểm thử bằng phần mềm với chức năng tương tự như phần cứng, tạo ra kết quả mong đợi phục vụ việc kiểm thử thiết kế. Sau cùng, thiết kế được hiện thực trên FPGA bằng kit Virtex-7 VC707 và tiến hành đánh giá kết quả.

II. GIẢI PHÁP

A. Bộ nhớ đệm (Cache)

Bộ nhớ đệm được thiết kế theo kiến trúc Multiway Set Associative với 4-Way và 16-Set riêng biệt cho từng lõi [1].

Cài đặt giải thuật thay thế PLRUt (Tree-based Pseudo Least Recently Used), sử dụng 3-bit theo dõi trạng thái truy cập của một Set với bộ nhớ đệm có cấu trúc 4-Way Set Associative [1].

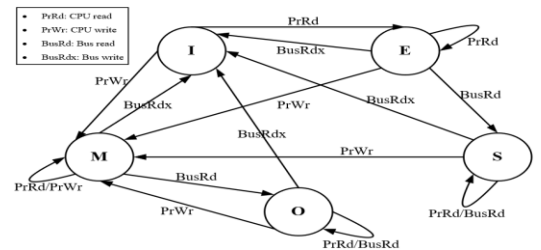


Hình 1. Giải thuật thay thế PLRUt [3]

Cài đặt chính sách ghi Write-back, chính sách cấp phát phát Read & Write Allocate để hoàn thiện bộ nhớ đệm [1].

B. Cache Coherence

Thiết kế bộ điều khiển Cache Coherence sử dụng giao thức MOESI Snoopy dùng kênh giao tiếp chung AXI mở rộng ACE hỗ trợ quản lý và theo dõi các truy cập dữ liệu. Giao thức gồm 5 trạng thái: Modified (đã chỉnh sửa), Owned (sở hữu), Exclusive (độc quyền), Shared (chia sẻ), Invalid (vô hiệu) [4].

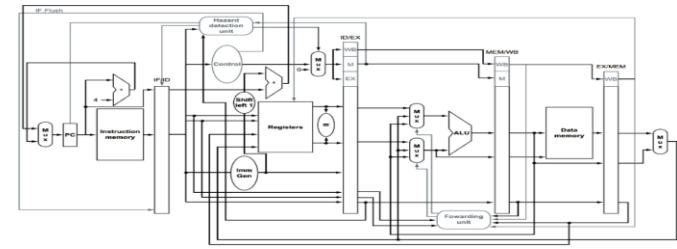


Hình 2. Giao thức Cache Coherence MOESI Snoopy

C. Lõi xử lý RISC-V RV32I

Lõi xử lý được thiết kế với kiến trúc pipeline 5 tầng bao gồm: IF (lấy và nạp lệnh), ID (giải mã lệnh, thực hiện điều khiển các tín hiệu đi qua các khối), EX (thực hiện tính toán các lệnh rẽ nhánh, ALU thực thi các tính toán số học), MEM (thực hiện các truy cập đọc/ghi vào bộ nhớ dữ liệu), WB (kết quả sau khi được tính toán sẽ được quyết định để ghi trở lại vào bộ thanh ghi hay không). Vấn đề phát hiện và xử lý Hazard [1]

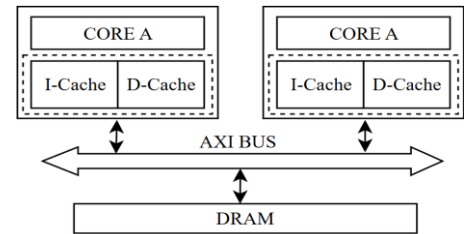
cũng được xử lý thông qua khối Hazard Unit. Bên cạnh đó, AXI Wrapper cũng được tích hợp để giao tiếp các thành phần khác.



Hình 3. Lối xử lý RISC-V RV32I pipeline 5 tầng

D. Hệ thống đa vi xử lý

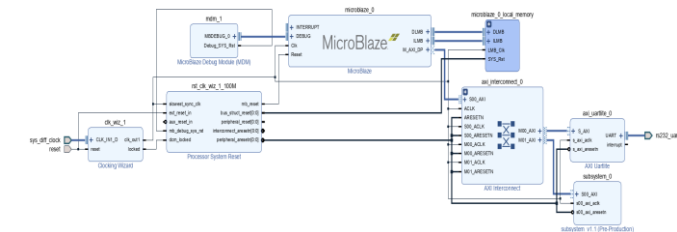
Để xác minh tính chính xác của bộ điều khiển Cache Coherence được thiết kế, một hệ thống 2 vi xử lý sẽ được xây dựng như Hình 4. Nội dung thiết kế chi tiết được chia ra thành hai phần gồm thiết kế phần cứng và mô phỏng bằng phần mềm.



Hình 4. Sơ đồ khối tổng quan của hệ thống

1. Thiết kế phần cứng trên FPGA bằng Verilog

- Bộ nhớ đệm với kiến trúc 4-Way Set Associative, cài đặt giải thuật thay thế PLRUt, chính sách ghi Write-back và chính sách cấp phát Read & Write Allocate.
- Bộ điều khiển Cache Coherence sử dụng giao thức MOESI Snoopy hỗ trợ 2 lối xử lý.
- AXI Bus mở rộng ACE hỗ trợ 4 Master (2 D-Cache & 2 I-Cache) và Slave (bộ nhớ chính).
- Lối xử lý RISC-V RV32I được pipeline 5 tầng, có giải quyết Hazard và tích hợp AXI Wrapper.
- Thiết kế Block Design tích hợp thêm IP BRAM của Xilinx làm bộ nhớ chính.



Hình 5. Block design tích hợp IP BRAM của Xilinx

- Tổng hợp và hiện thực thiết kế trên FPGA với kit Virtex-7 VC707.
2. Thiết kế phần mềm mô hình kiểm thử bằng Python
- Xây dựng mô hình kiểm thử là hệ thống 2 vi xử lý với các thành phần và chức năng giống như thiết kế phần cứng, có thể cho kết quả tính toán mong đợi.

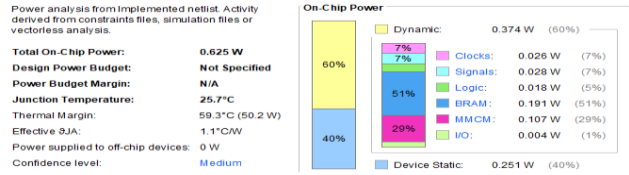
III. KẾT QUẢ

Kết quả thiết kế tổng hợp được trên phần mềm Vivado:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 1.956 ns	Worst Hold Slack (WHS): 0.058 ns	Worst Pulse Width Slack (WPWS): 1.100 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 25615	Total Number of Endpoints: 25615	Total Number of Endpoints: 7505

All user specified timing constraints are met.

Hình 6. Kết quả tổng hợp timing của hệ thống



Hình 7. Báo cáo năng lượng tiêu thụ của hệ thống

Tần số tối đa toàn hệ thống có thể hoạt động là 124MHz và tiêu thụ 0.625W.

IV. KẾT LUẬN

Sau khi hoàn thành thời gian thực hiện khóa luận, nhóm đã đạt được những kết quả như sau:

- Thiết kế phần cứng và kiểm thử thành công hệ thống 2 vi xử lý, các thiết kế thành phần cũng như toàn hệ thống hoạt động đúng chức năng, thiết kế được giao thức MOESI Snoopy và việc xử lý vấn đề Cache Coherence đúng như mong đợi.
- Thiết kế được mô hình kiểm thử bằng phần mềm, có thể tính toán ra kết quả mong đợi chính xác.
- Thực hiện đối chiếu, so sánh kết quả giữa thiết kế phần cứng và mô phỏng bằng phần mềm, kết quả cho ra trùng khớp, thiết kế hoạt động chính xác đúng theo danh sách testcase đã lập ra trong kế hoạch kiểm thử (test plan).
- Tần số hoạt động của hệ thống đạt mức 124MHz.
- Thành công hiện thực thiết kế lên FPGA với kit Virtex-7 VC707, chạy thử và cho ra kết quả chính xác với các testcase tiêu biểu trong test plan về Cache Coherence.

Để tiếp tục phát triển đề tài, nhóm đề xuất tinh chỉnh thiết kế để hỗ trợ cho hệ thống N vi xử lý (N > 2), tích hợp module MMU, DMA và các I/O để tạo thành hệ thống hoàn chỉnh.

V. TÀI LIỆU THAM KHẢO

[1] David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, RISC-V Edition, Morgan Kaufmann, 2018.

[2] ARM, AMBA AXI and ACE Protocol Specification: AXI3, AXI4, and AXI4-Lite; ACE and ACE-Lite, ARM IHI 0022E, Appendix B: Accelerator Coherency Port Interface Restrictions, 2013.

[3] Safaa Omran and Ibrahim A. Amory, "Implementation of LRU Replacement Policy for Reconfigurable Cache Memory Using FPGA", Kurdistan Region, Iraq, 2018.

[4] Daman Preet Kaur and V. Sulochana "Design and Implementation of Cache Coherence Protocol for High-Speed Multiprocessor System", 2018 International Conference on Advanced Science and Engineering (ICOASE), Iraq, 2018.

