ĐẠI HỌC QUỐC GIA TP. HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO TỔNG KẾT

ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ SINH VIÊN NĂM 2023

*Tên đề tài tiếng Việt: THIẾT KẾ KHỐI QUẢN LÝ BỘ NHỚ TRÊN KIẾN TRÚC RISC-V SỬ DỤNG GIẢI THUẬT LRU TRÊN KHỐI TLBs*

*Tên đề tài tiếng Anh: DESIGN THE MEMORY MANAGEMENT UNIT ON RISC-V ARCHITECTURE WITH THE TRANSLATION LOOKASIDE BUFFER USING LRU ALGORITHM*

Khoa/ Bộ môn: Kỹ thuật máy tính

Thời gian thực hiện: 06 tháng

Cán bộ hướng dẫn: ThS. Phạm Minh Quân

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tham gia thực hiện | | | | |
| TT | Họ và tên, MSSV | Chịu trách nhiệm | Điện thoại | Email |
|  | Bùi Đăng Huy | Chủ nhiệm | 0392211585 | 21520039@gm.uit.edu.vn |
|  | Nguyễn Quốc Trường An | Tham gia | 0865179307 | 21521810@gm.uit.edu.vn |

Thành phố Hồ Chí Minh – Tháng 02/2024

|  |  |  |  |
| --- | --- | --- | --- |
| LOGO DHCNTT -hinh.jpg | ĐẠI HỌC QUỐC GIA TP. HCM  TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN | Ngày nhận hồ sơ |  |
| Mã số đề tài |  |
| *(Do CQ quản lý ghi)* | |

BÁO CÁO TỔNG KẾT

*Tên đề tài tiếng Việt:* *THIẾT KẾ KHỐI QUẢN LÝ BỘ NHỚ TRÊN KIẾN TRÚC RISC-V SỬ DỤNG GIẢI THUẬT LRU TRÊN KHỐI TLBs*

*Tên đề tài tiếng Anh: DESIGN THE MEMORY MANAGEMENT UNIT ON RISC-V ARCHITECTURE WITH THE* *TRANSLATION LOOKASIDE BUFFER USING LRU ALGORITHM*

|  |  |
| --- | --- |
| *Ngày ... tháng ...... năm ....*  Cán bộ hướng dẫn  *(Họ tên và chữ ký)* | *Ngày ... tháng ...... năm ....*  Sinh viên chủ nhiệm đề tài  *(Họ tên và chữ ký)* |

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: THIẾT KẾ KHỐI QUẢN LÝ BỘ NHỚ TRÊN KIẾN TRÚC RISC-V SỬ DỤNG GIẢI THUẬT LRU TRÊN KHỐI TLBs

- Chủ nhiệm: Bùi Đăng Huy

- Thành viên tham gia: Nguyễn Quốc Trường An

- Cơ quan chủ trì: Trường Đại học Công nghệ Thông tin.

- Thời gian thực hiện: 06 tháng

2. Mục tiêu:

2.1. Lý do chọn đề tài

2.2. Mục tiêu tổng quan

2.3. Mục tiêu cụ thể

3. Tính mới và sáng tạo:

3.1. Phân tích hiện trạng

3.1.1. Tình hình nghiên cứu trong nước

Hiện nay trong nước các nghiên cứu về lĩnh vực vi mạch bán dẫn còn hạn chế và khó tiếp cận, đặc biệt là các nghiên cứu liên quan đến cải thiện tốc độ hệ thống bộ nhớ phân cấp, chủ yếu là các đề tài tốt nghiệp, các luận văn thạc sĩ… Các thiết kế trên có hiện thực được đơn vị quản lý bộ nhớ (Memory Management Unit) và bộ nhớ đệm nhưng chỉ ở mức thực hiện được chức năng cơ bản mà chưa quá chú trọng đến mức độ tiêu tốn tài nguyên phần cứng, năng lượng và giải thuật để cải thiện hiệu suất và đơn vị quản lí bộ nhớ được xem như một phần nhỏ trong thiết kế tổng thể của họ.

Một số đề tài liên quan đáng chú ý đã được thực hiện bởi sinh viên, học viên dùng cho khóa luận tốt nghiệp và luận văn thạc sĩ tại các trường đại học lớn trong nước như:

- Luận văn thạc sĩ của học viên Nguyễn Văn Linh – Trường Đại học Bách Khoa, thiết kế bộ nhớ đệm lệnh cho vi xử lí RISC-V. Đề tài đề xuất và hiện thực bộ đệm lệnh, tích hợp vào vi điều khiển Pulpino trong các hệ thống IoT, nhằm giúp cho vi điều khiển này có thiết kế linh hoạt hơn, tốn ít tài nguyên và năng lượng hơn.

- Khóa luận tốt nghiệp của sinh viên Trường Đại học Công nghệ Thông tin Lê Phước Nhật Nam và Trần Quốc Trưởng thiết kế và thực hiện đơn vị quản lí bộ nhớ. Đề tài phân tích và thiết kế IP có thể hiện thực được một mô hình đơn vị quản lí bộ nhớ (Memory Management Unit) sử dụng ngôn ngữ Verilog và thiết kế theo kiến trúc đường ống (pipeline).

3.1.2. Tình hình nghiên cứu quốc tế

Trên thế giới hiện có rất nhiều thiết kế và ý tưởng giải thuật nhằm cải tiến tốc độ truy cập bộ nhớ. Có thể kể đến các giải thuật nổi tiếng như Round Robin (RR), Most Recently Used (MRU), Least Recently Used (LRU)… Tuy nhiên mỗi giải thuật đều có ưu nhược điểm khác nhau về mức độ tiêu tốn tài nguyên, hiệu năng và độ phức tạp. Đa số các hệ thống thông dụng giao việc quản lí bộ nhớ và dịch địa chỉ cho hệ điều hành làm việc, điều này là tăng tính phức tạp và quá tải cho hệ điều hành. Song song đó, vẫn có một số thiết kế trên thế giới thực hiện việc tách rời quản lí bộ nhớ và đưa xuống phần cứng, một số khác thực hiện đánh giá hiệu năng của các giải thuật thay thế bộ nhớ đệm nhằm phục vụ cho mục đích tối ưu thiết kế, một số nghiên cứu tiêu biểu như:

- Performance evaluation of cache replacement policies for the SPEC CPU2000 benchmark suite của hai tác giả Hussein R. Al-Zoubi và Aleksandar Milenkovic công bố tại hội nghị Proceedings of the 42nd Annual Southeast Regional Conference năm 2004. Công bố thực hiện khảo sát đánh giá hiệu suất của các giải thuật Random, Last In First Out (FIFO), Least Recently Used (LRU), Pseudo Least Recently Used Policy based on MRU bits (PLRUm), Tree-based Pseudo Least Recently Used (PLRUt). Đánh giá thực hiện trên SimpleScalar toolset với SPEC CPU2000 Benchmark Suite và sử dụng bộ nhớ cache trên nhiều kích thước và kiến trúc khác nhau và tổng quát cho thấy giải thuật PLRUm cho hiệu năng tương đối tốt ở nhiều kích thước và kiến trúc bộ nhớ đệm khác nhau trong khi vẫn đảm bảo tính đơn giản và tiêu tốn ít tài nguyên.

- The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework, thiết kế của nhóm tác giả người nước ngoài Nastaran Hajinazarõ, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo F. Oliveira, Jonathan Appavoo, Vivek Seshadri, Onur Mutluõ. Thiết kế tiếp cận theo một hướng hoàn toàn mới lạ và chưa được áp dụng vào thực tế nhưng khá hay đó là sử dụng một không gian địa chỉ cho toàn bộ hệ thống (globally-visible address space – địa chỉ ảo toàn cục). VBI dựa trên ba nguyên tắc hướng dẫn. Đầu tiên, các chương trình phải được phép chọn kích thước không gian địa chỉ ảo của chúng, để giảm thiểu chi phí dịch thuật liên quan đến không gian địa chỉ ảo rất lớn. Thứ hai, việc dịch địa chỉ phải được tách rời khỏi việc bảo vệ bộ nhớ, vì chúng tách biệt về mặt logic và không cần được quản lý ở cùng mức độ chi tiết bởi cùng một cấu trúc. Thứ ba, phần mềm phải được phép truyền đạt thông tin ngữ nghĩa về dữ liệu ứng dụng tới phần cứng, để phần cứng có thể quản lý tài nguyên phần cứng cơ bản một cách thông minh hơn.

3.2 Phân tích công nghệ

3.2.1 Công nghệ bộ nhớ và bộ nhớ ảo

Hiệu suất hệ thống máy tính phụ thuộc rất nhiều vào hệ thống bộ nhớ cũng như kiến trúc bộ xử lý. Trước đây, bộ xử lý ban đầu tương đối chậm nên bộ nhớ có thể theo kịp. Nhưng tốc độ bộ xử lý đã tăng nhanh hơn tốc độ bộ nhớ. Tố độ bộ nhớ DRAM hiện chậm hơn tốc độ bộ xử lý từ 10 đến 100 lần. Khoảng cách ngày càng tăng giữa tốc độ bộ xử lý và tốc độ bộ nhớ DRAM đòi hỏi các hệ thống bộ nhớ ngày càng khéo léo hơn để cố gắng thu hẹp khoảng cách giữa tốc độ bộ nhớ và tốc độ bộ xử lý. Hình dưới đây minh họa tốc độ phát triển của bộ xử lý và bộ nhớ qua các năm:

A graph showing the memory and processor

Description automatically generated

Hình 1. Tốc độ phát triển của bộ xử lý và bộ nhớ qua các năm

Bộ nhớ ảo là một kỹ thuật quản lý bộ nhớ trong đó bộ nhớ phụ có thể được sử dụng như thể nó là một phần của bộ nhớ chính. Bộ nhớ ảo là một kỹ thuật phổ biến được sử dụng trong hệ điều hành (OS) của máy tính. Bộ nhớ ảo sử dụng cả phần cứng và phần mềm để cho phép máy tính bù đắp sự thiếu hụt bộ nhớ vật lý, tạm thời chuyển dữ liệu từ bộ nhớ truy cập ngẫu nhiên (RAM) sang bộ lưu trữ trên đĩa. Ánh xạ các khối bộ nhớ vào các tệp đĩa cho phép máy tính xử lý bộ nhớ phụ như thể nó là bộ nhớ chính.

3.2.2 Bộ nhớ phân cấp và Translation Lookaside Buffer (TLB)

Bộ nhớ máy tính chủ yếu được xây dựng từ RAM động (DRAM) và RAM tĩnh (SRAM). Lý tưởng nhất là hệ thống bộ nhớ máy tính nhanh, lớn và rẻ. Trong thực tế, một bộ nhớ chỉ có hai trong số ba thuộc tính này; nó chậm, nhỏ hoặc đắt tiền. Nhưng hệ thống máy tính có thể gần giống lý tưởng bằng cách kết hợp bộ nhớ nhanh, nhỏ, rẻ và bộ nhớ chậm, lớn, rẻ, tạo ra một hệ thống bộ nhớ phân cấp. Thông thường một hệ thống bộ nhớ phân cấp bao gồm:

- Registers (Thanh ghi): Thanh ghi là các đơn vị bộ nhớ nhỏ, tốc độ cao nằm trong CPU. Chúng được sử dụng để lưu trữ dữ liệu và hướng dẫn được sử dụng thường xuyên nhất. Các thanh ghi có thời gian truy cập nhanh nhất và dung lượng lưu trữ nhỏ nhất, thường từ 16 đến 64 bit.

- Cache Memory (Bộ nhớ đệm): Bộ nhớ đệm là một đơn vị bộ nhớ nhỏ, nhanh nằm gần CPU. Nó lưu trữ dữ liệu và hướng dẫn được sử dụng thường xuyên được truy cập gần đây từ bộ nhớ chính. Bộ nhớ đệm được thiết kế để giảm thiểu thời gian truy cập dữ liệu bằng cách cung cấp cho CPU quyền truy cập nhanh vào dữ liệu được sử dụng thường xuyên.

- Main Memory (bộ nhớ chính, bộ nhớ vật lý): Bộ nhớ chính hay còn gọi là RAM (Bộ nhớ truy cập ngẫu nhiên) là bộ nhớ chính của hệ thống máy tính. Nó có dung lượng lưu trữ lớn hơn bộ nhớ đệm nhưng chậm hơn. Bộ nhớ chính được sử dụng để lưu trữ dữ liệu và hướng dẫn hiện đang được CPU sử dụng.

- Secondary Storage (Bộ nhớ thứ cấp): Bộ lưu trữ thứ cấp, chẳng hạn như ổ đĩa cứng (HDD) và ổ đĩa thể rắn (SSD), là đơn vị bộ nhớ cố định có dung lượng lưu trữ lớn hơn bộ nhớ chính. Nó được sử dụng để lưu trữ dữ liệu và hướng dẫn hiện không được CPU sử dụng. Bộ lưu trữ thứ cấp có thời gian truy cập chậm nhất và thường là loại bộ nhớ ít tốn kém nhất trong hệ thống phân cấp bộ nhớ.

Trong đó, Translation Lookaside Buffer (TLB) có thể được coi là bộ đệm của các mục trong bảng trang nằm trong bộ nhớ. Về cơ bản, chúng lưu trữ bản dịch của địa chỉ bộ nhớ ảo sang địa chỉ bộ nhớ vật lý. Nó được sử dụng để giảm thời gian truy cập vào vị trí bộ nhớ người dùng. Nó có thể được gọi là bộ đệm dịch địa chỉ. Nó là một phần của đơn vị quản lý bộ nhớ (MMU) của chip. TLB có thể nằm giữa CPU và bộ đệm CPU, giữa bộ đệm CPU và bộ nhớ chính hoặc giữa các cấp độ khác nhau của bộ đệm đa cấp. Phần lớn bộ xử lý máy tính để bàn, máy tính xách tay và máy chủ bao gồm một hoặc nhiều TLB trong phần cứng quản lý bộ nhớ và nó gần như luôn có trong bất kỳ bộ xử lý nào sử dụng bộ nhớ ảo phân trang hoặc phân đoạn.

Một cách phổ biến dùng để đánh giá hiệu năng của hệ thống bộ nhớ phân cấp:

Average access time = Hit ratio x Average cache access time + (1 – Hit ratio) x (Average cache access time + Average upper-level access time)

Trong đó:

- Average access time: Thời gian trung bình truy cập bộ nhớ

- Hit ratio: Tỷ lệ hit của một cấp bộ nhớ

- Average cache access time: Thời gian trung bình truy cập của một cấp bộ nhớ

- Average upper-level access time: Thời gian truy cập trung bình của cấp bộ nhớ tiếp theo

3.2.3 Giải thuật xấp xỉ LRU: Pseudo Least Recently Used Policy based on MRU bits (PLRUm)

Vì bộ đệm không thể chứa toàn bộ bộ nhớ nên các nhà thiết kế phải đối mặt với một vấn đề quan trọng: dòng nào sẽ được lưu trữ ở vị trí nào của bộ đệm, dòng nào sẽ được giữ trong bộ đệm và dòng nào sẽ bị loại bỏ? Nhiều thuật toán thay thế dòng bộ đệm cố gắng trả lời câu hỏi thay thế theo cách hiệu quả, tiêu thụ điện năng thấp, nhanh chóng, dễ thực hiện và rẻ tiền nhưng vẫn đảm bảo hiệu suất.

Giải thuật Pseudo Least Recently Used Policy based on MRU bits (PLRUm) là một giải thuật xấp xỉ giải thuật Least Recently Used (LRU). Trong giải thuật này, mỗi dòng bộ đệm được gán một bit MRU, được lưu trong bảng thẻ. Bit MRU cho mỗi dòng bộ đệm được đặt thành “1” mỗi lần xảy ra lần truy cập bộ đệm tại dòng này, cho biết dòng bộ đệm gần đây đã được sử dụng. Khi bộ điều khiển bộ đệm buộc phải thay thế một dòng bộ đệm, nó sẽ kiểm tra bit MRU cho mỗi dòng bộ đệm để tìm kiếm “0”. Khi tìm thấy số “0”, bộ điều khiển bộ đệm sẽ thay thế dòng bộ đệm đó và sau đó đặt bit MRU thành “1”. Vấn đề có thể xảy ra nếu các bit MRU cho tất cả các khối bộ nhớ đệm được đặt thành “1”. Nếu điều này xảy ra, tất cả các dòng đều không có sẵn để thay thế, gây ra Deadlock. Để ngăn chặn kiểu Deadlock này, tất cả các bit MRU trong bộ đệm sẽ bị xóa ngoại trừ bit MRU được truy cập khi phát hiện tình huống có thể gây Deadlock (tất cả các bit MRU đều là “1”).

Đã có một số nghiên cứu chứng minh rằng PLRUm và PLRUt là những thuật toán xấp xỉ rất tốt của thuật toán LRU. PLRUm vượt trội hơn PLRUt và thậm chí còn tốt hơn LRU trong một số thiết kế. Điều này được lý giải rằng tại sao PLRUm đã được sử dụng trong các máy tính IBM (ví dụ IBM3033). Do đó, PLRUm là một thuật toán khá tối ưu, đáp ứng yêu cầu: rẻ, hiệu quả, dễ triển khai hơn cho các ứng dụng nhúng.

3.3 Tính mới và sáng tạo của đề tài

Nhìn chung, đề tài thiết kế đơn vị quản lý bộ nhớ đã xuất hiện từ lâu với nhiều biến thể và sử dụng nhiều phương pháp và tích hợp nhiều giải thuật khác nhau. Tuy nhiên với mục tiêu ban đầu là phục vụ việc học tập và nghiên cứu về công nghệ bộ nhớ hỗ trợ bộ nhớ ảo và việc dịch địa chỉ hiện nay, thiết kế giúp cung cấp một cái nhìn tổng quan và cơ bản về khối quản lý bộ nhớ (Memory Management Unit).

Qua quá trình tìm hiểu các đề tài nghiên cứu đã thực hiện trước đó, đa số các thiết kế đều theo hướng phát triển dựa trên hệ thống bộ nhớ ảo thông thường. Việc tiếp tục nỗ lực để tối ưu hóa hiệu năng và chi phí trên nền kiến trúc bộ nhớ cũ hiện nay không mang lại quá nhiều cải tiến và lợi ích, thậm chí đôi khi còn làm giảm hiệu suất của hệ thống trong một số trường hợp nhất định. Thiết kế của nhóm đi theo hướng tiếp cận hoàn toàn mới là tách biệt việc bảo vệ bộ nhớ ra khỏi việc dịch địa chỉ đồng thời giao nhiệm vụ quản lý bộ nhớ xuống cho phần cứng thực hiện.

3.3.1 Tách biệt việc bảo vệ bộ nhớ ra khỏi việc dịch địa chỉ

Bộ nhớ ảo nên được tách riêng việc dịch địa chỉ khỏi việc kiểm tra bảo về quyền truy cập, vì hai nhiệm vụ nào có đặc điểm hoàn toàn khác nhau, việc tách rời một trong chúng ra không làm ảnh hưởng đến cái còn lại thậm chí còn mang đến nhiều lợi ích. Mặc dù việc dịch địa chỉ thường được thực hiện ở mức độ chi tiết của trang nhưng thông tin bảo vệ thường giống nhau đối với toàn bộ cấu trúc dữ liệu, có thể trải rộng trên nhiều trang. Hơn nữa, thông tin bảo vệ hoàn toàn là chức năng của địa chỉ ảo và không yêu cầu dịch địa chỉ. Tuy nhiên, các hệ thống hiện tại lưu trữ cả thông tin dịch và thông tin bảo vệ cho mỗi trang ảo như một phần của bảng trang. Việc tách nhiệm vụ dịch địa chỉ ra khỏi nhiệm vụ kiểm tra bảo vệ bộ nhớ có thể tạo cơ hội loại bỏ việc dịch địa chỉ ra khỏi đường găng (Critical Path). Trì hoãn việc dịch cho đến khi việc kiểm tra bảo vệ bộ nhớ được hoàn thành và truy cập bộ nhớ được cấp quyền. Nghĩa là hệ thống sẽ kiểm tra quyền truy cập bộ nhớ trước, nếu truy cập bộ nhớ là hợp lệ thì việc dịch địa chỉ mới được tiến hành, thay vì như trong các hệ thống thông thường hiện nay, thông tin bảo vệ bộ nhớ được lưu như một phần của bảng trang, việc kiểm tra và dịch địa chỉ được làm cùng nhau, làm tăng các truy cập bộ nhớ không cần thiết. Việc tách rời này mang lại lợi ích giảm chi phí hoạt động cho bộ nhớ ảo, giúp tăng tốc khi truy cập bộ nhớ vật lý.

3.3.2 Giao nhiệm vụ quản lý bộ nhớ cho phần cứng thực hiện

Việc giao nhiệm vụ quản lý bộ nhớ xuống cho phần cứng thực hiện thay vì hệ điều hành thực hiện giúp phân chia cân bằng hơn nhiệm vụ giữa hệ điều hành (phần mềm) và Memory Management Unit (phần cứng). Bởi trong các hệ thống thông thường hiện này, hệ điều hành đảm nhận quá nhiều các tác vụ liên quan đến cả phần mềm người dùng, phần mềm hệ thống và thông tin phần cứng. Việc này làm cho việc phát triển hệ điều hành ngày càng trở nên khó khăn và khó quản lý, đồng thời mất cân bằng giữa việc phân chia tác vụ phần mềm và phần cứng. Việc giao nhiệm vụ quản lý bộ nhớ xuống cho phần cứng thực hiện nhằm giải quyết vấn đề này, giảm tải cho hệ điều hành, hơn nữa, việc đưa tác vụ quản lý bộ nhớ xuống cho phần cứng thực hiện cũng giúp tăng hiệu suất hệ thống bởi tốc độ thực hiện phần cứng nhanh hơn tốc độ thực hiện phần mềm rất nhiều.

4. Tóm tắt kết quả nghiên cứu:

4.1. Đặc tả kỹ thuật

4.2. Thiết kế chi tiết

5. Tên sản phẩm: KHỐI QUẢN LÝ BỘ NHỚ TRÊN KIẾN TRÚC RISC-V SỬ DỤNG GIẢI THUẬT LRU TRÊN KHỐI TLBs

6. Hiệu quả, phương thức chuyển giao kết quả nghiên cứu và khả năng áp dụng:

7. Hình ảnh, sơ đồ minh họa chính:

|  |  |
| --- | --- |
| Cơ quan Chủ trì  *(ký, họ và tên, đóng dấu)* | Chủ nhiệm đề tài  *(ký, họ và tên)* |