|  |  |
| --- | --- |
|  |  |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG****---------------------------------------** |
|  |
|  |
| **VƯƠNG VIẾT THAO** |
|  |
| **NGHIÊN CỨU THIẾT KẾ BỘ ĐIỀU KHIỂN BỘ NHỚ NGOÀI DDRAM3 SỬ DỤNG AXI4 TRÊN HỆ THỐNG TRÊN CHIP (SOC)****TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SỸ KĨ THUẬT***(Theo định hướng ứng dụng)* |
|  |
|  |
| **HÀ NỘI - 2025** |

|  |  |
| --- | --- |
|  |  |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG****KHOA ĐÀO TẠO SAU ĐẠI HỌC****---------------------------------------** |
|  |
|  |
| **VƯƠNG VIẾT THAO** |
| **NGHIÊN CỨU THIẾT KẾ BỘ ĐIỀU KHIỂN BỘ NHỚ NGOÀI DDRAM3 SỬ DỤNG AXI4 TRÊN HỆ THỐNG TRÊN CHIP (SOC)**

|  |  |
| --- | --- |
| **Chuyên ngành:** | **Kĩ thuật điện tử** |
| **Mã số :** | **8.52.02.03** |

 |
| **ĐỀ ÁN TỐT NGHIỆP THẠC SỸ KĨ THUẬT***(Định hướng ứng dụng)*NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN TRUNG HIẾU |
| **HÀ NỘI - 2025** |

**MỞ ĐẦU**

Trong bối cảnh ngành công nghiệp vi mạch và bán dẫn toàn cầu đang phát triển mạnh mẽ, Việt Nam cũng xác định đây là lĩnh vực chiến lược cần được thúc đẩy. Thiết kế hệ thống trên chip (SoC) đang trở thành trọng tâm nghiên cứu trong giáo dục và công nghiệp vi điện tử. Đề tài *"Nghiên cứu thiết kế bộ điều khiển bộ nhớ ngoài DDR3 sử dụng AXI4 trên hệ thống trên chip (SoC)"* được lựa chọn nhằm đóng góp vào nền tảng công nghệ bán dẫn trong nước, đáp ứng nhu cầu lưu trữ tốc độ cao và dung lượng lớn trong các hệ thống nhúng hiện đại.

Giao thức AXI4 (thuộc hệ AMBA của ARM) đã trở thành chuẩn giao tiếp phổ biến trong các SoC nhờ tính năng truyền dữ liệu hiệu quả, hỗ trợ burst, kênh độc lập và khả năng mở rộng cao. Việc xây dựng một bộ điều khiển DDR3 tương thích với AXI4 không chỉ giúp tối ưu hóa luồng dữ liệu giữa CPU và bộ nhớ, mà còn dễ dàng tích hợp vào các hệ thống xử lý tín hiệu số hoặc IoT.

Mục tiêu nghiên cứu của đề tài là thiết kế, mô phỏng và thử nghiệm thành công một bộ điều khiển DDR3 tuân thủ giao thức AXI4, có khả năng cấu hình linh hoạt và độ ổn định cao. Phương pháp tiếp cận dựa trên việc sử dụng công cụ thiết kế logic số, IP Core MIG 7 Series hỗ trợ AXI4, cùng các kỹ thuật mô phỏng – đo kiểm để đánh giá hiệu quả truyền dữ liệu và độ chính xác trong hoạt động đọc/ghi.

**NỘI DUNG**

**CHƯƠNG I: CƠ SỞ LÝ LUẬN**

1. **Tổng quan về bộ nhớ truy cập ngẫu nhiên DRAM**

 Bộ nhớ RAM (Random Access Memory) là bộ nhớ cho phép truy xuất dữ liệu tại bất kỳ vị trí nào một cách nhanh chóng, không theo thứ tự. RAM có tốc độ cao nhưng là bộ nhớ dễ bay hơi, mất dữ liệu khi mất điện.

1. **DRAM và DDR SDRAM**
2. ***DRAM***

DRAM là bộ nhớ truy cập ngẫu nhiên động, lưu trữ dữ liệu bằng điện tích trong tụ điện. Mỗi ô nhớ gồm một tụ và một transistor, cần làm mới định kỳ để tránh mất dữ liệu do rò rỉ điện.



Hình 1. 1: Cấu trúc tổng quát của DRAM

1. ***DDR SDRAM và phân loại***
2. DDR2 SDRAM

 DDR2 SDRAM nâng cấp từ DDR với tốc độ cao hơn (200–533 MHz), băng thông đến 1066 Mb/s/pin và điện áp giảm còn 1.8V giúp tiết kiệm năng lượng. Tích hợp ODT, tín hiệu vi sai và tính năng Additive Latency giúp ổn định và tối ưu hiệu suất. DDR2 hỗ trợ 8 bank và burst length đến 8, phù hợp hệ thống hiệu năng cao.

1. DDR3 SDRAM

 DDR3 còn hỗ trợ tính năng tự làm mới tự động (self-refresh) và làm mới theo vùng (partial array self-refresh) giúp tiết kiệm năng lượng ở chế độ chờ. Nhờ khả năng burst dài và truy cập xen kẽ các bank, DDR3 cải thiện đáng kể hiệu suất truyền dữ liệu. Với ưu điểm về tốc độ, điện năng và độ tin cậy, DDR3 được ứng dụng rộng rãi trong các hệ thống nhúng, SoC, máy tính cá nhân và thiết bị di động hiệu năng cao.

1. DDR4 SDRAM

 DDR4 cải thiện hiệu suất với tốc độ cao, điện áp thấp (1.2V), và mật độ lưu trữ gấp đôi so với DDR3, phù hợp cho hệ thống hiệu năng cao. Cơ chế thiết lập Vref mới và kiểm soát jitter chính xác giúp tăng độ tin cậy truyền dữ liệu. Nhờ đó, DDR4 trở thành lựa chọn chuẩn cho các ứng dụng xử lý tốc độ cao và tiết kiệm năng lượng.

1. **Cấu trúc của DDR SDRAM**
2. ***Cell storage trong DDR SDRAM***

 Trong DDR SDRAM, cell lưu trữ gồm một transistor và một tụ điện (capacitor), nơi dữ liệu được lưu dưới dạng điện tích. Transistor điều khiển truy cập đọc/ghi, còn capacitor lưu giá trị “1” (có điện) hoặc “0” (mất điện). Do tụ điện bị rò điện theo thời gian, cell cần được làm mới định kỳ để giữ dữ liệu.

****

Hình 1. 2: Cells Strorage trong DDR SDRAM

1. ***Các thành phần cơ bản của DDR SDRAM***
2. Bank

Banks trong DDR SDRAM là các đơn vị bộ nhớ con giúp tối ưu hiệu suất bằng cách cho phé Banks trong DDR SDRAM là các khối bộ nhớ độc lập, cho phép truy cập song song để tăng hiệu suất và giảm độ trễ. Mỗi bank gồm nhiều hàng và cột, tạo thành lưới ô nhớ lưu dữ liệu. DDR3 có từ 4 đến 16 bank, hỗ trợ thực thi lệnh đồng thời và tăng tốc độ xử lý.

1. Rank

 Rank trong DDR SDRAM là tập hợp các chip nhớ hoạt động như một đơn vị độc lập, chia sẻ chung lệnh điều khiển và bus dữ liệu. Có hai loại: Single Rank (1 rank) và Dual Rank (2 rank, truy cập luân phiên). Rank giúp tăng dung lượng và cải thiện hiệu suất nhờ khả năng xen kẽ truy xuất, giảm độ trễ trong xử lý.

1. Row

 Lệnh ACTIVATE trong DDRAM được gửi kèm địa chỉ hàng để chọn và chuẩn bị một hàng trong bank sẵn sàng truy cập. Bộ giải mã hàng kích hoạt đường từ tương ứng, cho phép điện tích từ các ô nhớ truyền đến đường bit, là bước đầu tiên để thực hiện thao tác đọc hoặc ghi dữ liệu.

1. Column

 Cấu trúc cột trong DDRAM gồm các đường bit nối dọc theo cột và liên kết với ô nhớ tại mỗi giao điểm hàng–cột. Mỗi ô chứa một bit dữ liệu dưới dạng điện tích. Các đường bit được kết nối với bộ khuếch đại cảm biến để tăng cường tín hiệu, giúp đọc dữ liệu chính xác từ ô nhớ.

1. **Nguyên tắc truy cập bộ nhớ DDR SDRAM**

 Lệnh activate được gửi kèm địa chỉ hàng để chọn một hàng trong bank cần truy cập. Địa chỉ này được giải mã, kích hoạt đường từ kết nối ô nhớ với đường bit. Điện tích từ ô nhớ truyền đến bộ khuếch đại cảm biến, nơi tín hiệu được khuếch đại đủ mạnh để đọc hoặc ghi dữ liệu chính xác.

1. **Kết luận chương I**

Chương 1 trình bày cơ sở lý thuyết về bộ nhớ DRAM, tập trung vào phân tích các loại như SDRAM, DDR và đặc biệt là DDR3 SDRAM. DDR3 được chọn làm đối tượng nghiên cứu chính nhờ hiệu suất cao, phổ biến và phù hợp với yêu cầu thiết kế bộ điều khiển bộ nhớ ngoài cho SoC. Phần này cung cấp nền tảng quan trọng cho các chương thiết kế và mô phỏng tiếp theo.

**CHƯƠNG II: NGHIÊN CỨU GIẢI PHÁP ĐIỀU KHIỂN BỘ NHỚ NGOÀI DDRAM3 TRÊN SOC**

1. **Giới thiệu về Soc (System on chip)**
2. ***SoC là gì?***

 SoC (System on Chip) là vi mạch tích hợp toàn bộ hệ thống trên một chip duy nhất, kết hợp phần cứng số, tương tự và hỗn hợp để tối ưu kích thước, hiệu suất và năng lượng. SoC thường gồm CPU, GPU, bộ nhớ, điều khiển ngoại vi, giao tiếp mạng và quản lý năng lượng. Nhờ tính tích hợp cao, SoC được ứng dụng rộng rãi trong thiết bị di động, hệ thống nhúng và ô tô thông minh.

1. ***Các ứng dụng của SoC***

 SoC được ứng dụng rộng rãi trong điện thoại, thiết bị IoT, ô tô thông minh và thiết bị y tế nhờ kích thước nhỏ, hiệu suất cao và tiêu thụ điện thấp. Chúng đảm bảo xử lý nhanh, tiết kiệm năng lượng và hoạt động ổn định trong nhiều môi trường.

1. ***Các giải pháp thiết kế SoC***
2. IP và công nghệ nền cho SoC

 *IP core trong thiết kế SoC:*

Star core SC3850

DSP Core

32KB L1 ICache

32KB L1 Dcache

512KB L2 Cache

Class Multiple Fabric

3MB Shared

M3 Memory

64 bit DDR3 Memory Controller 1.33MHz

I2C

SPI

UART

GPIO

DMA

Ethernet

Maple-B Baseband Accelerator

Hinh 2. 1: Kiến trúc tổng quan về lõi IP Core trong SoC

 IP core là các khối chức năng được thiết kế sẵn như CPU, GPU, bộ nhớ, bộ điều khiển I/O hay giao tiếp không dây, dùng để tích hợp nhanh vào SoC. Trong SoC, nhiều khối nhỏ cần trao đổi dữ liệu với CPU, nên cần thiết kế cầu nối (bus) để liên kết chúng hiệu quả.

1. Phương pháp thiết kế theo module

 Thiết kế SoC kết hợp chặt chẽ phần cứng và phần mềm, thường sử dụng phương pháp đồng thiết kế kiến trúc. Phần cứng được xây dựng từ các lõi IP: Hard IP (cố định, hiệu suất cao), Soft IP (mã nguồn HDL, linh hoạt), và Firm IP (dạng trung gian). Phần mềm dùng ngôn ngữ cấp cao như C++/MATLAB, chuyển thành RTL qua công cụ HLS. Sau khi xác lập kiến trúc, các thành phần được mô tả bằng HDL và kết nối thành hệ thống SoC hoàn chỉnh.

1. **Các thành phần cơ bản của SoC**

 SoC (System on Chip) là chip tích hợp toàn bộ hệ thống vào một vi mạch duy nhất, gồm các thành phần chính như CPU, GPU, bộ nhớ và thiết bị I/O. SoC được ứng dụng rộng rãi trong điện thoại, IoT, máy tính bảng và hệ thống nhúng. Bài viết tập trung tìm hiểu kiến trúc và các đặc điểm nổi bật của SoC.

Multimedia encoder/decoder

Direct Memory Access

CPU

DSP

Network interface card

Audio

USB

Video

Storage

Memory

Hinh 2. 2: Các thành phần cơ bản của SoC

1. **Các loại bộ nhớ phổ biến dùng để tích hợp trên SoC**

Một số bộ nhớ ngoài phổ biến trong SoC gồm:

SRAM: Tốc độ nhanh, không cần làm tươi, thường dùng làm cache nhưng chi phí cao và dung lượng thấp.

DRAM: Dung lượng lớn, giá rẻ hơn SRAM, cần làm tươi định kỳ nên kém hiệu quả hơn trong ứng dụng tốc độ cao.

NAND Flash: Lưu trữ không mất dữ liệu khi tắt nguồn, dung lượng lớn, giá rẻ, nhưng tốc độ đọc/ghi chậm hơn.

DDRAM: Kết hợp ưu điểm của DRAM và SRAM, cải thiện tốc độ và hiệu suất, phù hợp với SoC hiệu năng cao.

1. **Các giải pháp điều khiển bộ nhớ**
2. ***Điều khiển bộ nhớ dữ trên giao thức chuẩn***

 AXI là giao tiếp bus hiệu năng cao trong kiến trúc AMBA của ARM, hỗ trợ burst transfer, pipelining, và kênh đọc/ghi độc lập. Nó đảm bảo băng thông lớn, độ trễ thấp và quản lý QoS hiệu quả, phù hợp với các SoC phức tạp dùng vi xử lý ARM.

1. ***Bộ điều khiển bộ nhớ chuyên dùng***

 Bộ điều khiển bộ nhớ đóng vai trò quản lý truy cập và làm tươi dữ liệu trong DDR3. Nó tối ưu hiệu suất bằng kỹ thuật burst read/write và lập lịch lệnh thông minh, giúp giảm overhead và tăng băng thông đáng kể

1. **So sánh các giao thức điều khiển bộ nhớ ngoài trên SoC**
2. ***Giao thức AXI trong SoC***

 AXI là giao diện bus thuộc kiến trúc AMBA của ARM, đáp ứng yêu cầu băng thông cao và độ trễ thấp trong SoC hiện đại. Nó hỗ trợ burst transfer, pipelining, kênh đọc/ghi độc lập và QoS, giúp truyền dữ liệu hiệu quả và đảm bảo hiệu suất trong các hệ thống đa master–slave phức tạp, đặc biệt trên nền ARM.

1. ***Giao thức Avalon***

 Avalon là giao thức bus dành cho thiết kế SoC trên FPGA, hỗ trợ giao tiếp ánh xạ bộ nhớ và truyền dữ liệu luồng. Nó đơn giản, dễ tích hợp giữa các thành phần hệ thống nhưng thiếu các tính năng nâng cao như QoS hay hỗ trợ đa master phức tạp, so với AXI.

1. ***Các đặc điểm nổi bật của AXI4 trong điều khiển bộ nhớ DDR***

 AXI4 hỗ trợ burst transfers lên đến 256 beats, giúp tăng hiệu quả truyền dữ liệu trong ứng dụng tốc độ cao. Giao dịch đọc và ghi được tách biệt qua các kênh riêng, hỗ trợ pipelining, giảm độ trễ và tăng thông lượng. AXI4 phù hợp với hệ thống đa master/slave, dễ tích hợp và tối ưu khi giao tiếp với bộ nhớ ngoài như DDR hay NAND Flash.

**CHƯƠNG III: XÂY DỰNG HỆ THỐNG BỘ ĐIỀU KHIỂN DDRAM CONTROLLER.**

1. **Tổng quan về IP-Core trong thiết kế bộ điều khiển DDR SDRAM**
2. ***Tổng quan về IP-Core.***

 IP-Core là các mô-đun phần cứng thiết kế sẵn, dùng để tích hợp nhanh vào hệ thống SoC mà không cần xây dựng từ đầu. Chúng giúp tiết kiệm thời gian, đảm bảo độ ổn định nhờ được kiểm thử sẵn, và có thể tùy chỉnh linh hoạt trong nhiều dự án. IP-Core thường được cung cấp dưới dạng mã HDL và tích hợp dễ dàng qua các công cụ như Vivado.

1. ***Phân tích tổng quan lõi Ip truy xuất bộ nhớ.***

 Lõi IP điều khiển bộ nhớ là thành phần quan trọng giúp kết nối hiệu quả giữa bộ xử lý và DDR3 SDRAM trong hệ thống SoC. Nó sử dụng bus AXI4, hỗ trợ truyền burst và cho phép tùy chỉnh các tham số thời gian. Lõi IP kết nối vật lý với DDR3 qua các tín hiệu DQ, DQS, CLK… và có thể tích hợp thêm tính năng như ECC hoặc tự động làm mới. Ví dụ, IP MIG trong Vivado hỗ trợ kết nối DDR3 tối ưu và đơn giản hóa thiết kế trên FPGA như Artix-7.

1. **Giao thức AXI4 trong điều khiển bộ nhớ ngoài DDR3 SDRAM**
2. ***Cấu trúc giao thức AXI4***
3. Giới thiệu AXI4

AXI là một giao thức bus thuộc họ AMBA do ARM phát triển, được ưa chuộng nhờ tính mở và khả năng hỗ trợ hệ thống hiệu năng cao. Giao thức này nổi bật với: Phân tách địa chỉ, điều khiển và dữ liệu, hỗ trợ truy cập unaligned, write strobe và burst, tách biệt kênh đọc và ghi, cho phép nhiều transaction đồng thời và out-of-order, hỗ trợ chèn thanh ghi để cải thiện timing hệ thống.

1. Các khái niệm quan trọng sử dụng trong AXI4

Transaction: Quá trình truyền dữ liệu giữa master và slave, gồm đọc, ghi và phản hồi. Aligned: Địa chỉ dữ liệu khớp với kích thước từ (ví dụ 32-bit tại địa chỉ chia hết cho 4). Unaligned: Địa chỉ dữ liệu không khớp với kích thước từ (ví dụ 32-bit tại địa chỉ 0x03). Burst Transfer: Truyền nhiều từ dữ liệu liên tiếp trong một giao dịch để tăng hiệu suất. Burst Length: Số lượng từ dữ liệu được truyền trong một burst (1–256 từ). Multiple Outstanding Transactions: Master có thể phát nhiều giao dịch mà không cần chờ phản hồi từ các giao dịch trước.

1. Các quy định của giao thức AXI4

 Giao thức AXI định nghĩa chuẩn giao tiếp giữa master, slave và interconnect thông qua các interface – tập hợp tín hiệu truyền dữ liệu. Trong AXI, master thực hiện các transaction đọc hoặc ghi với slave, gọi chung là access.

BUS 1 (Interconnect)

BUS 0(Interconnect)

Master2

Master3

Slave3

Slave4

Master0

Master1

Slave0

Slave1

Slave2

Hình 3. 1: Cấu trúc của giao thức AXI4

1. ***Quá trình đọc, ghi và cơ chế bắt tay của AXI4***
2. Cơ chế phân kênh AXI4

 AXI sử dụng 5 kênh độc lập gồm: AR (địa chỉ đọc), R (dữ liệu đọc), AW (địa chỉ ghi), W (dữ liệu ghi), và B (phản hồi ghi). Mỗi kênh có chức năng riêng và hoạt động song song. Một transaction bao gồm 4 thành phần chính: địa chỉ, điều khiển, dữ liệu và phản hồi.

Master

Slave

Read address channel

Read data channel with read response

Write address channel

Write data channel

Write response channel

Hình 3. 2: Cơ chế đọc, ghi và phản hồi giữa master và slave trong giao thức AXI4

1. Quá trình đọc dữ liệu

 Một transaction đọc gồm hai bước xử lý:

 Phía Master gửi một thông tin địa chỉ và thông tin điều khiển để khởi động một transaction đọc trên kênh AR.

 Phía slave gửi các dữ liệu kèm thông tin response trên kênh R. Số lượng dữ liệu và thông tin response được quy định bởi thông tin điều khiển phát từ phía master trên kênh AR

Burst

Read data response 1 (A)

Read data response 0 (A)

Read data response 0 (B)

Master

Slave

Address and control A

Address and control B

Read data response 2 (B)

Read data response 1 (B)

Beat

TransactionA

Read Channel

Transaction B

Read Channel

Hình 3. 3: Ví dụ về quá trình đọc của AXI4

 Hoạt động này thể hiện rõ đặc điểm "(1) AXI tách riêng pha truyền địa chỉ (address), thông tin điều khiển (control) với pha truyền dữ liệu (data)" đã nói phía trên

1. Quá trình ghi

 Một transaction ghi gồm 3 bước xử lý:

Phía master gửi một thông tin địa chỉ và thông tin điều khiển để khởi động một transaction ghi trên kênh AW

Phía master gửi các dữ liệu ghi trên kênh W. Số lượng dữ liệu ghi được quy định bởi thông tin điều khiển phát từ phía master trên kênh AW

Phía slave gửi thông tin response khi burst ghi đã hoàn thành

Burst

Write data response 1 (A)

Write data response 0 (A)

Write data response 0 (B)

Master

Slave

Address and control A

Address and control B

Write data response 2 (B)

Write data response 1 (B)

Beat

TransactionA

Write Channel

Transaction B

Write Channel

Hình 3. 4: Quá trình ghi của giao thức AXI4

1. Cơ chế bắt tay

 AXI sử dụng cơ chế bắt tay hai chiều VALID/READY để điều phối truyền dữ liệu. Nguồn phát VALID khi có dữ liệu, đích phát READY khi sẵn sàng nhận. Mỗi kênh có cặp tín hiệu riêng, cho phép truyền độc lập, tối ưu hiệu năng và tránh xung đột.



Hình 3. 5: Quá trình yêu cầu và phản hồi trong cơ chế bắt tay của AXI4

1. ***Cơ chế burst transaction***

 Giao thức AXI4 hỗ trợ truyền dữ liệu theo cơ chế burst transaction, trong đó chỉ cần phát địa chỉ đầu tiên của burst, còn các địa chỉ tiếp theo được slave tự tính toán dựa vào thông tin điều khiển. Master phát địa chỉ và điều khiển trên kênh AR (đọc) hoặc AW (ghi), gồm: AxADDR: Địa chỉ byte đầu tiên của burst, AxLEN: Số lượng beat (transfer) trong burst, AxSIZE: Kích thước mỗi beat (ví dụ 8, 16, 32 bit…), AxBURST: Kiểu burst (incrementing, fixed...), AxVALID / AxREADY: Tín hiệu bắt tay điều phối truyền

Master

Slave

Read Address Channel

ARADDR[\*:1]

ARLEN[7:0]

ARSIZE[2:0]

ARBURST[1:0]

ARVALID

ARREADY

RDATA[\*:0]

RRESP[1:0]

RLAST

RVALID

AREADY

Read Data Channel

Hình 3. 6: Cơ chế busrt transaction

 Loại response được xác định bằng giá trị trên RRESP và BRESP. AXI hỗ trợ 4 loại response

Table 1 Bảng các kiểu trả về của slave

|  |  |
| --- | --- |
| RRESP [1:0]BRESP[1:0] | Response |
| 0b00 | OKAY |
| 0b01 | EXOKAY |
| 0b10 | SLVERR |
| 0b11 | DECERR |

1. ***Thiết kế giao diện AXI4.***
	1. AXI4 tương tác với Slaves

 AXI4 gồm 5 kênh: AW (địa chỉ ghi), W (dữ liệu ghi), B (phản hồi ghi), AR (địa chỉ đọc), và R (dữ liệu đọc). Các kênh hoạt động độc lập, giúp truyền dữ liệu hiệu quả và cung cấp phản hồi đầy đủ cho mỗi transaction.

* 1. Phản hồi của slaves về cho master thông qua AXI4

 Trong AXI4, master gửi lệnh ghi qua kênh AW và dữ liệu qua kênh W, slave phản hồi trạng thái qua kênh B. Với lệnh đọc, master gửi địa chỉ qua kênh AR, slave trả dữ liệu và trạng thái qua kênh R.

* 1. Khối điều khiển phân luồng của AXI4 (Arbiter)

 Khối arbiter trong AXI4 phân luồng và ưu tiên truy cập giữa các master, đảm bảo các kênh đọc và ghi hoạt động song song, hiệu quả và không xung đột trong hệ thống SoC.

Read Cmd Block

Write Cmd Block

Read Data Block

Write data block

Write Response block

Abiter

Hình 3. 7: Khối điều khiển phân luồng của Arbiter

* 1. Quá trình lưu dữ liệu FIFO

 FIFO là bộ nhớ tạm hoạt động theo nguyên tắc "vào trước, ra trước". Nó có hai thông số chính: độ sâu (số ô nhớ) và độ rộng (số bit mỗi ô). Dữ liệu được quản lý bằng con trỏ ghi và đọc – là các bộ đếm tuần tự để xác định vị trí ghi/đọc.

 FIFO giúp đồng bộ hóa dữ liệu giữa các thành phần hoạt động với tốc độ khác nhau bằng cách đóng vai trò như một bộ đệm tạm thời. Trong hệ thống AXI4, FIFO hỗ trợ lưu trữ dữ liệu giữa các giao dịch đọc và ghi, đảm bảo truyền dữ liệu liên tục, giảm độ trễ và tránh mất mát thông tin khi tốc độ đọc/ghi không khớp nhau giữa master và slave.

1. **Xây dựng bộ điều khiển bộ nhớ DDR3 SRAM**
2. ***Kiến trúc tổng quan bộ điều khiển DDR3 SDRAM***

 Bộ điều khiển DDR3 đóng vai trò cầu nối giữa master AXI và bộ nhớ DDR3 SDRAM, đảm nhận việc khởi tạo và xử lý các yêu cầu thời gian phức tạp của DDR3.

A

X

I

4

DDR Controller

DDR3 Memory

Write Address

Read Address

Write Data

Read Data

Write Response

Hình 3. 8: Sơ đồ khối thiết kế bộ điều khiển DDR3 SDRAM dùng giao diện AXI4

1. ***Thiết kế khối điều khiển truy cập địa chỉ***
2. Cấu trúc địa chỉ của AXI4 và DDR3 SDRAM

 AXI4 là một giao thức bus tiên tiến, hỗ trợ định dạng địa chỉ linh hoạt với độ dài 32-bit hoặc 64-bit tùy cấu hình hệ thống.

 Cơ chế chuyển đổi địa chỉ từ AXI4 sang DDR3

 Quá trình chuyển đổi địa chỉ được thực hiện bằng cách trích xuất các bit địa chỉ từ AXI4 và ánh xạ tương ứng theo định dạng của DDR3, dựa trên cấu hình kích thước của từng vùng địa chỉ (số lượng bank, số hàng, số cột). Ví dụ: nếu hệ thống DDR3 có 8 bank, 16K hàng và 1K cột, thì địa chỉ AXI4 cần được chia thành:

3 bit cho Bank (log₂ (8) = 3),

14 bit cho Row (log₂ (16K) = 14),

10 bit cho Column (log₂ (1K) = 10)

1. ***Thiết kế khối điều khiển dữ liệu***
2. Thành phần chính của khối điều khiển

 Khối điều khiển dữ liệu chịu trách nhiệm đảm bảo việc truyền tải dữ liệu giữa bus AXI4 và bộ nhớ DDRAM3. Đảm bảo việc dữ liệu được đọc và ghi chính xác, kịp thời và đáp ứng được yêu cầu về băng thông, đột rễ.

1. Cơ chế truyền tải dữ liệu từ AXI4 với DDR3

 Trong hệ thống giao tiếp giữa AXI4 và DDR3, cơ chế chuyển đổi địa chỉ và truyền dữ liệu đóng vai trò cốt lõi, đảm bảo sự phối hợp hiệu quả giữa thiết bị chủ (master) và bộ nhớ. Bộ điều khiển trung gian chính là thành phần chịu trách nhiệm điều phối hai luồng chính: ghi từ AXI4 sang DDR3 và đọc từ DDR3 về AXI4. Cơ chế này tận dụng kết hợp bộ đệm nội bộ và truyền dữ liệu theo burst nhằm tối ưu hóa băng thông và giảm độ trễ truy cập.

1. Cải thiện hiệu suất truyền tải dữ liệu

 Prefetching (dự đoán dữ liệu) là một kỹ thuật tối ưu hóa nhằm giảm độ trễ truy cập bộ nhớ DDR3 bằng cách chủ động nạp trước dữ liệu vào bộ đệm nội bộ, dựa trên các mẫu truy cập được phát hiện từ các giao dịch AXI4. Thay vì chờ đến khi thiết bị chủ gửi yêu cầu cụ thể, bộ điều khiển sử dụng cơ chế phân tích mẫu truy cập (access pattern recognition) để phát hiện các chuỗi truy cập tuần tự hoặc định kỳ, từ đó đưa ra quyết định thực hiện các lệnh đọc trước (prefetch commands) một cách chủ động.

1. **Thiết kế bộ quản lý truy cập DDR3 SDRAM**
2. ***Tổng quan về quản lý truy cập AXI***

 Nhiệm vụ quan trọng nhất của bộ quản lý truy cập AXI là chuyển đổi các lệnh AXI thành các lệnh truy cập bộ nhớ để tối đa hóa việc sử dụng băng thông DDR3. Bộ nhớ DDR3 chỉ nhận lệnh ở chế độ burst 4 hoặc 8, trong khi lệnh AXI burst nhỏ hơn hoặc dài hơn. Bộ quản lý truy cập AXI kết hợp các lệnh bất cứ khi nào có thể để cải thiện hiệu suất và chuyển lệnh cuối cùng đến bộ điều khiển DDR3. Để đảm bảo băng thông là tối đa, nó tiền nạp các lệnh từ giao diện AXI, dịch chúng thành các giao dịch bộ nhớ và lưu trữ cục bộ

Control Logic

Control Store

Burst Manage

AXI - IF

Address Control

Hình 3. 9: Tổng quan về bộ truy cập AXI

1. ***Các thành phần quản lý chính***

 Control Logic là thành phần điều khiển chính của khối truy cập AXI, chịu trách nhiệm phân tích lệnh, quyết định thứ tự thực thi và phối hợp các khối như Burst Manager và Address Control. Nó tối ưu hiệu suất bằng cách kết hợp lệnh nhỏ thành burst lớn, xử lý theo ưu tiên và tránh tắc nghẽn hệ thống.



Hình 3. 10: Sơ đồ thiết kế các phần của DDR3 controller

 AXI-IF là cầu nối giữa bộ xử lý và bộ điều khiển, nhận lệnh từ các kênh AXI (AR, AW, W, R, B), giải mã và phân tích loại lệnh, địa chỉ, dữ liệu, burst size. Nó sử dụng FIFO để lưu tạm lệnh, hỗ trợ xử lý song song nhiều lệnh, đảm bảo không tắc nghẽn. Sau khi giải mã, AXI-IF chuyển lệnh đến Burst Manager và trả phản hồi về host sau khi xử lý xong.

Khối trung tâm quản lý bank

Khối điều khiển nguồn

Khởi tạo khối điều khiển

Khối làm tươi bộ nhớ

Address Control

Bank Manager

D

D

R

A

M

3

Hình 3. 11: Sơ đồ khối chi tiết của Memory Interface generator (DDR3 Controller)

1. **Thiết kế máy trạng thái cho bộ điều khiển**
2. ***Thiết kế máy trạng thái ghi dữ liệu và ghi địa chỉ***

 Sơ đồ trên minh hoạ các hoạt động đọc ghi giữa bus master và bộ nhớ DDR3. Nó minh hoạ luồng dữ liệu dịch chuyển trong các mode đọc ghi dữ liệu. Hay nói cách khác sơ đồ trên biểu thị luồng dữ liệu (Data Path). Sau khi có tín hiệu reset.

Hình 3. 12: Máy trạng thái ghi dữ liệu và ghi địa chỉ

1. ***Thiết kế máy trạng thái cho việc ghi lệnh***

 Khi gửi lệnh để đọc và ghi, thì việc thiết kế máy trạng thái quản lý lệnh sẽ giúp quản lý lệnh một cách chặt chẽ hơn. Khi quản lý lệnh với máy trạng thái cần tuân thủ theo các tín hiệu phản hồi về từ slave tới master và ngược lại.

r=1

w=1

Hình 3. 13: Máy trạng thái cho việc ghi lệnh

1. ***Thiết kế máy trạng thái cho việc làm tười bộ nhớ***

 FSM (Finite State Machine) làm mới bộ nhớ trong hệ thống điều khiển DDR3 có chức năng duy trì tính toàn vẹn của dữ liệu bằng cách làm mới các hàng trong bộ nhớ. Bộ nhớ DDR3 là bộ nhớ động (DRAM), yêu cầu làm mới định kỳ để tránh mất dữ liệu. Quá trình này có thể được thực hiện bằng hai phương pháp chính:

Hình 3. 14: Máy trạng thái cho việc làm tươi bộ nh

**CHƯƠNG IV: MÔ PHỎNG VÀ ĐÁNH GIÁ MỨC ĐỘ HIỆU QUẢ CỦA BỘ ĐIỀU KHIỂN DDRAM**

1. **Mô phỏng bộ điều khiển bộ nhớ ngoài DDR3 SDRAM**
2. ***Cấu hình chi IP MIG7 Series.***

 Đầu tiên việc chọn board mạch hết sức quan trong, trong project của tôi đang sử dụng dòng FPGA Artix™ 7 FPGA AC701 Evaluation Kit. Tạo một sơ đồ khối (Block Diagram - BD) mới, sau đó vào danh mục IP (IP Catalog) để thêm IP vào sơ đồ. Sau đó sẽ chọn lõi “Memory Interface Generator (MIG 7 Series)”.



Hình 4. 1: Lựa chọn các board mạch FPGA cho việc mô phỏng bộ điều khiển DDR3

1. ***Viết code thực thi và thiết kế mô phỏng***
2. Thiết kế giao diện AXI RX (64bit data)

 Thiết kế một khối giao diện tuân thủ chuẩn AXI4-Stream RX dùng để tiếp nhận dữ liệu 64-bit từ các IP ngoại vi hoặc khối xử lý upstream.

1. Thiết kế khối FiFo Buffer

 Cài đặt bộ nhớ FIFO làm bộ đệm trung gian cho dữ liệu và địa chỉ truyền xuống bộ nhớ DDR3.

1. Thiết kế khối tập hợp dữ liệu

 Gom các từ dữ liệu 64-bit thành block 512-bit phục vụ cho ghi DDR3 theo burst.

**Chức năng:** Tăng bộ đếm cnt\_block sau mỗi lần ghi. Khi đủ số block cần thiết thì gán tín hiệu wlast.

1. Khối ghi dữ liệu

 Sinh các giao dịch ghi DDR3 theo chuẩn AXI4, bao gồm lệnh ghi địa chỉ và dữ liệu.

FSM điều khiển awvalid để gửi lệnh ghi địa chỉ.

Tự động tăng địa chỉ ghi awaddr.

Chuẩn bị tín hiệu điều khiển awid, awlen, awsize.

1. Khối đọc dữ liệu

 Sinh các lệnh đọc từ bộ nhớ DDR3 theo chuẩn AXI4 để lấy lại dữ liệu đã ghi: FSM điều khiển arvalid. Tự động tăng araddr. Đảm bảo các lệnh đọc được gửi khi ghi xong.

1. Khối AXI Stream TX

 Tách dữ liệu đọc được (512-bit) thành từng từ 64-bit và truyền ra ngoài qua giao diện AXI4-Stream.

Tạm thời chưa hoàn thiện trong phiên bản hiện tại.

Cần thêm bộ FSM để tách từng S\_AXI\_rdata thành 8 phần 64-bit.

Điều khiển tx\_axis\_tvalid, tx\_axis\_tdata, tx\_axis\_tkeep, tx\_axis\_tlast.

1. **Kiểm thử bộ điều khiển với các chức năng write/read**

 Dữ liệu ghi vào DDR3 là tín hiệu: S\_AXI\_wdata <= wdata;

 Và wdata được tăng dần trong process wvalid\_control:

 wdata <= wdata + 1;

*Các tín hiệu điều khiển ghi:*

S\_AXI\_awaddr: địa chỉ ghi

S\_AXI\_awvalid: bắt đầu ghi

S\_AXI\_wvalid: dữ liệu hợp lệ

S\_AXI\_wlast: kết thúc burst write

S\_AXI\_wstrb: byte mask (đang gán '1' toàn bộ, tức ghi toàn bộ dữ liệu)

S\_AXI\_bready: xác nhận phản hồi write

1. **Đánh giá mức độ hiệu quả của bộ điều khiển thiết kế**

 Dựa trên kết quả mô phỏng chức năng trong môi trường Vivado, có thể đánh giá bộ điều khiển thiết kế DDR3 như sau:



Hình 4. 2: Kết quả chạy thử nghiệm với bộ điều khiển DDR3 dùng AXI4



Hình 1: Kết quả mô phỏng với quá trình ghi dữ liệu

 Đọc dữ liệu: Các phản hồi từ phía bộ nhớ (s\_axi\_rdata) cũng xuất hiện tương ứng, khớp với trình tự đã ghi.



Hình 4. 3: Kết quả mô phỏng đọc dữ liệu từ bộ nhớ DDR3

1. Kết luận chương IV

Chương IV đã mô phỏng và kiểm thử thành công bộ điều khiển DDR3 trên nền tảng SoC, xác nhận khả năng thực thi đúng chuẩn AXI4, đảm bảo quá trình đọc/ghi ổn định và chính xác. Kết quả chứng minh thiết kế đạt hiệu năng tốt, đáng tin cậy và sẵn sàng cho triển khai hoặc mở rộng sau này.

**KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN.**

Đề tài đã hoàn thành thiết kế bộ điều khiển DDR3 trên SoC sử dụng Vivado và giao thức AXI4, hỗ trợ truyền dữ liệu burst, đảm bảo tốc độ cao và độ chính xác trong giao tiếp bộ nhớ. Kiểm thử mô phỏng cho thấy dữ liệu đọc/ghi trùng khớp, chứng minh thiết kế hoạt động chính xác.

Hướng phát triển: mở rộng giao diện AXI4 với các chế độ địa chỉ mới, bổ sung tín hiệu phản hồi nâng cao, phát triển bộ điều khiển đa kênh để tăng băng thông, và ứng dụng cho BRAM hoặc các chuẩn bộ nhớ cao hơn như DDR4, DDR5.

**DANH MỤC TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1]  | "DDR3 SDRAM Standard (JESD79-3C)," 2007. Available: https://www.jedec.org/standards-documents/docs/jesd-79-3d. |
| [2]  | V. Milovanović and a. D. Tasovac, "A Customizable DDR3 SDRAM Controller Tailored for FPGA-Based Data Buffering Inside Real-Time Range-Doppler Radar Signal Processing Back Ends," in *IEEE*, Indian, 2019.  |
| [3]  | S. Chandrasekar and a. J. Thangaraj, "Implementation and Optimization of Throughput in High Speed Memory Interface Using AXI Protoco," in *ICCCNT*, 2018.  |
| [4]  | A. Shrivatsava, V. Yadav and a. D. Puthal, "Comparative Analysis of AXI and AHB Protocols," in *ICECS*, 2010.  |
| [5]  | B. Wang, J. Du, X. Bi and a. X. Tian, "High Bandwidth Memory Interface Design Based on DDR3 SDRAM and FPGA," in *ISOCC*, Taiwan, 2015.  |
| [6]  | H. El-Hassan, K. Hachani and and Lahcen El Abdellaoui, "A Synthesizable AXI Protocol Checker for SoC Integration," in *IEEE* , 2020.  |
| [7]  | P. Rakshit and a. N. Patel, "RTL Design for Time Efficient DDR3 Memory Interfaced with RTG4 FPGA," in *in Proceedings of the ICOEI*, 2019.  |
| [8]  | H. Byun, H.-J. Park, K.-H. Kim and a. J.-D. Lee, "FPGA-based DDR3 DRAM Interface Using Bulk-Si Optical Interconnects," in *Samsung Electronics*, Korean, 2019.  |
| [9]  | A. Graham and a. J. David, "IP Solves the Increasing Challenge of Implementing an Interface to Off-Chip DDR SDRAM," in *DesignCon*, 2007.  |
| [10]  | A. Ltd, "AMBA AXI and ACE Protocol Specification, Version 2.0," 2011. [Online]. Available: https://developer.arm.com/documentation/ihi0022/latest. |
| [11]  | M. W. a. L. B. Jinhui Yi, "Design of DDR3 SDRAM read-write controller based on," *Jinhui Yi et al 2021 J. Phys.: Conf. Ser. 1846 012046,* 2021.  |
| [12]  | S. C and J. Thangaraj, "Implementation and optimization of throughput in high speed memory interface using AXI protocol," in *India*, Bengaluru, India, 2018.  |
| [13]  | C.-H. Chen, J.-C. Ju and a. I.-J. Huang, "A Synthesizable AXI Protocol Checker for SoC Intergration," in *National Science Council (NSC) of Taiwan*, Taiwan, 2010.  |
| [14]  | M. Gupta and A. K. Nagawat, "Design and Implementation of High Performance Advanced Extensible Interface(AXI) Based DDR3 Memory Controller," in *International Conference on Communication and Signal Processing*, Indian, 2016.  |