JTE HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

A Comparison of the Wishbone and Amba Axi Bus Architecture

Nguyen Quang Anh Tuan*, Nguyen Van Tuan, Vo Phan Man Dat, Van Ba Huy, Truong Quang Phuc, Nguyen Ngo Lam
Ho Chi Minh City University of Technology and Education, Vietnam
* Corresponding author. *Email: 18161299@student.hcmute.edu.vn*

ARTICLE INFO

Received: 4/7/2022
Revised: 19/7/2022
Accepted: 19/8/2022
Published: 30/8/2022

KEYWORDS

System-on-Chip (SoC)
IP Core
WISHBONE Bus
AMBA AXI bus
Bus Architecture

ABSTRACT

In the current trend of The Fourth Industrial Revolution, System-on-chip (SoC) design plays an important role in embedded systems and is a leading field of many industrial countries around the world. The rapid development of semiconductor technology makes it possible to integrate more and more components on a chip. Bus protocols were developed and used as a common interface for efficient interconnection of on-chip components, reducing complexity, energy consumption, and manufacturing costs. In this paper, we compare two popular bus architectures WISHBONE and AMBA AXI (Advanced Microcontroller Bus Architecture Advanced eXtensible Interface) based on the point-to-point connection system model by performing simulation and comparing their performance based on resource parameters, power consumption of each system. Specifically, the Point-to-Point connection model of both buses will use a MASTER interface which is a DMA (Direct Memory Access) connected to a SLAVE interface which is a RAM (Random Access Memory). These interfaces are IP cores that can be used and reused for different applications and purposes. The results are verified through simulation with the software Xilinx Vivado 2019.1.

So Sánh Kiến Trúc Bus Wishbone Và Amba Axi

Nguyễn Quang Anh Tuấn*, Nguyễn Văn Tuấn, Võ Phan Mẫn Đạt, Văn Bá Huy, Trương Quang Phúc, Nguyễn Ngô Lâm
Trường Đại học Sư phạm Kỹ thuật Tp. Hồ Chí Minh, Việt Nam
** Tác giả liên hệ. Email: 18161299@student.hcmute.edu.vn*

THÔNG TIN BÀI BÁO

Ngày nhận bài: 4/7/2022
Ngày hoàn thiện: 19/7/2022
Ngày chấp nhận đăng: 19/8/2022
Ngày đăng: 30/8/2022

TỪ KHÓA

Hệ thống trên chip (SoC)
Lõi IP
Bus WISHBONE
Bus AMBA AXI
Kiến trúc Bus

TÓM TẮT

Trong xu hướng hiện tại của cuộc Cách mạng Công nghiệp 4.0, thiết kế Hệ thống trên chip (SoC) đóng vai trò quan trọng trong các hệ thống nhúng và là một lĩnh vực mũi nhọn của nhiều nước công nghiệp trên thế giới. Sự phát triển nhanh chóng của lĩnh vực công nghệ bán dẫn giúp chúng ta có thể tích hợp ngày càng nhiều các thành phần trên một con chip. Các giao thức bus được phát triển và sử dụng như một giao diện chung cho việc kết nối hiệu quả giữa các thành phần trên chip, giúp giảm mức độ phức tạp, năng lượng tiêu thụ và chi phí sản xuất. Trong bài báo này, nhóm tác giả so sánh hai loại kiến trúc bus phổ biến là bus WISHBONE và AMBA AXI (Advanced Microcontroller Bus Architecture Advanced eXtensible Interface) dựa trên mô hình hệ thống kết nối điểm - điểm bằng cách thực hiện mô phỏng và so sánh hiệu năng của chúng dựa trên thông số tài nguyên, công suất tiêu thụ của mỗi hệ thống. Cụ thể, mô hình kết nối điểm – điểm của cả hai bus sẽ sử dụng một giao diện MASTER là một DMA (Direct Memory Access) kết nối với một giao diện SLAVE là một bộ nhớ RAM (Random Access Memory). Các giao diện này là các lõi IP có thể được sử dụng và tái sử dụng cho các ứng dụng và mục đích khác nhau. Kết quả được thực nghiệm thông qua mô phỏng với phần mềm Xilinx Vivado 2019.1.

Doi: https://doi.org/10.54644/jte.71B.2022.1234

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

1. Giới thiệu

Thiết kế Hệ thống trên chip là một trong những xu hướng phát triển rất mạnh mẽ trong cuộc cách mạng công nghiệp 4.0 trên toàn thế giới, khi mà các nhà sản xuất có thể tích hợp ngày càng nhiều thành phần của một hệ thống điện tử như bộ vi xử lý (CPU), bộ nhớ (ROM, RAM), các hệ thống xử lý tín hiệu số (DSP) trên một con chip duy nhất. Nhờ đó, SoC có thể trở thành một vi mạch tích hợp (IC) có thể thực hiện hầu hết các chức năng của một hệ thống điện tử hoàn chỉnh trong một kích thước nhỏ gọn hơn, hiệu suất và tính ổn định cũng cao hơn [1, 2].

Bus là hệ thống có vai trò chính trong việc giao tiếp, trung chuyển dữ liệu giữa các thành phần trong một SoC và ảnh hưởng trực tiếp đến hiệu suất của hệ thống. Một số các kiến trúc bus hệ thống phổ biến hiện nay như: AMBA (Advanced Microcontroller Bus Architecture) AHB (Advanced High-Performance Bus) APB (Advanced Peripheral Bus) hay AXI (Advanced Extensible Interface) bus phát triển bởi ARM, CoreConnect của IBM, Open Core Protocol (OCP) của OCP International Partnership, WISHBONE của OpenCores [3].

Trong số các kiến trúc bus hệ thống kể trên, bus WISHBONE và AMBA AXI được sử dụng rất phổ biến và rộng rãi trong thiết kế SoC. Bus WISHBONE được sử dụng rất phổ biến trong SoC vì nó là một giải pháp thiết kế đơn giản, linh hoạt, hỗ trợ người dùng bằng cách chuẩn hóa các giao diện lõi IP (Intellectual Property) bằng sơ đồ kết nối tiêu chuẩn với kết nối có thể thay đổi cấu trúc liên kết như điểm - điểm (Point-to-Point), luồng dữ liệu (Data Flow), bus chia sẻ (Shared Bus) hoặc chuyển mạch chéo (Crossbar Switch), tùy vào nhu cầu và loại SoC thích hợp, giúp người dùng tiết kiệm đáng kể chi phí và thời gian sản xuất [4]. AMBA AXI được sử dụng phổ biến trong các vi điều khiển ARM, hỗ trợ các hệ thống hiệu năng cao, tần số cao, phù hợp với các thiết kế có độ trễ thấp và băng thông cao mà không yêu cầu sử dụng các cầu nối phức tạp, đồng thời đáp ứng các yêu cầu giao tiếp của nhiều thành phần trên chip và cũng rất tương thích với các giao thức AHB và APB của họ AMBA.

Trong các bài viết [5, 6], tác giả đã giới thiệu tổng quan về các giao thức bus phổ biến sử dụng trong SoC hiện nay như AMBA (AXI, AHB, APB) của ARM, WISHBONE của OpenCores, CoreConnect của IBM, mô tả tóm tắt các đặc điểm và đưa ra những so sánh về ưu nhược điểm của từng loại để lựa chọn phù hợp cho các ứng dụng khác nhau. Bên cạnh đó, tác giả đã thực hiện việc mô phỏng và xác minh giao diện kết nối điểm – điểm của bus WISHBONE. Trong bài viết [7], tác giả đã thực hiện thêm mô phỏng và đánh giá hai hệ thống kết nối theo kiểu điểm – điểm và bus chia sẻ, thống kê các thông số tài nguyên sử dụng trên FPGA (Field Programmable Gate Array) Virtex-II Pro và Spartan3e.

Trong bài viết [8], tác giả đã giới thiệu tổng quan về các đặc điểm, kiến trúc và hoạt động của bus AMBA AXI, mô tả thiết kế và triển khai mô hình kết nối AXI sử dụng ngôn ngữ mô tả phần cứng Verilog trên phần cứng Xilinx Spartan 3E FPGA và mô phỏng thực hiện với phần mềm Modelsim RTL Simulation. Trong bài [9], tác giả mô tả các đặc điểm quan trọng và các kết nối trong giao diện AXI. Ở bài viết [10] tác giả đã tính toán độ trễ trong các hoạt động đọc và ghi bằng cách phát triển mã hóa RTL (Register Transfer Level) Verilog và tổng hợp nó với công cụ của Synopsys ở tiến trình 90nm.

Mặc dù có nhiều nghiên cứu và đánh giá về bus WISHBONE và AMBA AXI, tuy nhiên chưa có bài viết so sánh về kiến trúc, kết nối hay hiệu năng của chúng. Do đó, trong bài báo này chúng tôi đề xuất sử dụng mô hình kết nối điểm - điểm của hai loại kiến trúc bus WISHBONE và AMBA AXI để thực hiện so sánh và đánh giá hiệu năng của hệ thống dựa trên các thông số tài nguyên, công suất tiêu thụ. Quá trình mô phỏng và đánh giá thông qua sử dụng mô phỏng trên phần mềm Xilinx Vivado 2019.1. Kết quả sau khi thực nghiệm cho thấy hệ thống sử dụng bus WISHBONE sử dụng tài nguyên và công suất tiêu thụ ít hơn so với bus AMBA AXI. Bên cạnh đó, hệ thống sử dụng bus AMBA AXI khi thực hiện quá trình xác nhận thông tin truyền nhận dữ liệu yêu cầu nhiều tín hiệu để xác nhận hơn. AMBA AXI cho thời gian và hiệu suất truyền nhận dữ liệu tốt hơn do quá trình đọc/ ghi diễn ra trên các kênh độc lập nhau, một dữ liệu mới có thể được ghi trong khi đang thực hiện quá trình đọc miễn là không diễn ra trên cùng một địa chỉ.

Phần còn lại của bài báo được tổ chức như sau. Phần 2 giới thiệu về hai loại bus WISHBONE và AMBA AXI, trình bày về các đặc điểm, mô hình kết nối của hai kiến trúc bus. Phần 3 trình bày các kết quả thu được từ nghiên cứu, giải thích hoạt động của hai mô hình kết nối, so sánh và nhận xét thông qua

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

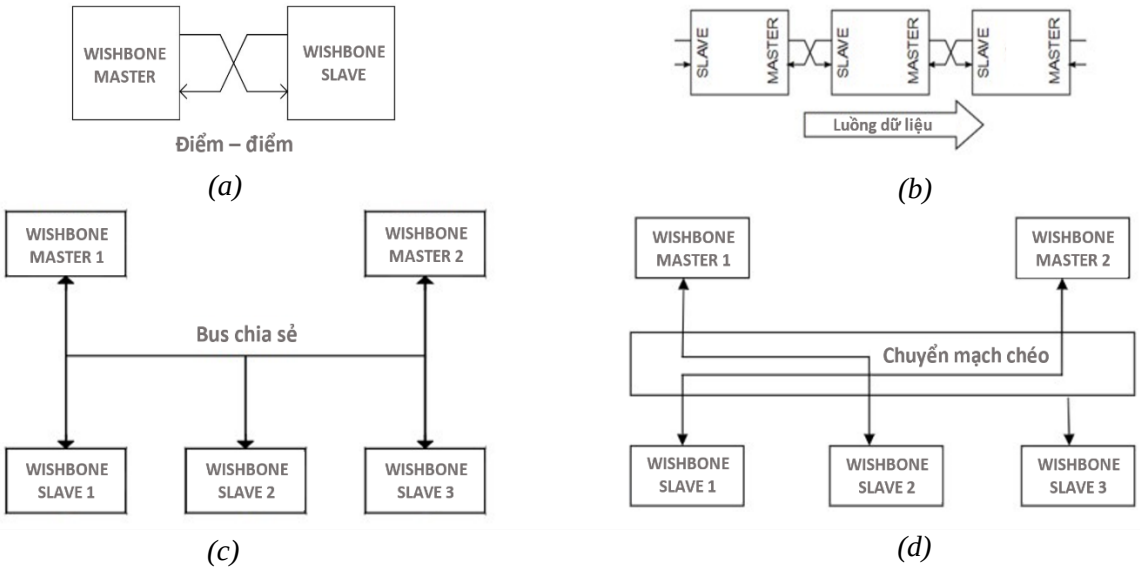
kết quả mô phỏng và các thông số tài nguyên sử dụng của cả hai hệ thống. Cuối cùng, phần 4 là phần kết luận rút ra từ những kết quả thu được trong quá trình thực hiện đề tài và đề xuất những hướng nghiên cứu, phát triển và ứng dụng từ kết quả nghiên cứu.

2. Bus WISHBONE

WISHBONE là một kiến trúc bus hệ thống thường được sử dụng trong thiết kế SoC. Đây là một phương pháp thiết kế đơn giản, linh hoạt giúp cho việc kết nối các lõi IP trong một SoC trở nên hiệu quả và dễ dàng hơn. Các lõi IP có thể được phát triển bởi các nhà sản xuất khác nhau với các chức năng và cách kết nối khác nhau tùy vào nhu cầu của người dùng. Mục đích chính của bus WISHBONE là hỗ trợ người dùng chuẩn hóa các giao diện lõi IP dưới một giao diện chung, giúp cho việc kết nối các lõi IP trở nên dễ dàng và có thể tái sử dụng thiết kế này trên các hệ thống khác [11, 12].

2.1. Kiến trúc và hoạt động của bus WISHBONE

WISHBONE sử dụng kiến trúc MASTER/SLAVE. Kết nối WISHBONE cho phép người dùng thay đổi cách mà các lõi IP kết nối với nhau giúp cho việc triển khai phần cứng có khả năng tương thích với nhiều loại cấu trúc liên kết [12]. Điều này rất quan trọng vì không có một cách duy nhất nào đúng để thực hiện cho mọi SoC. Có bốn loại kết nối WISHBONE được mô tả như trong Hình 2.1 bên dưới, bao gồm: điểm - điểm (Point-to-Point), luồng dữ liệu (Data Flow), bus chia sẻ (Shared Bus) và chuyển mạch chéo (Crossbar Switch).



Hình 2.1. *Kết nối (a): Điểm – điểm; (b): Luồng dữ liệu; (c): Bus chia sẻ; (d): Chuyển mạch chéo*

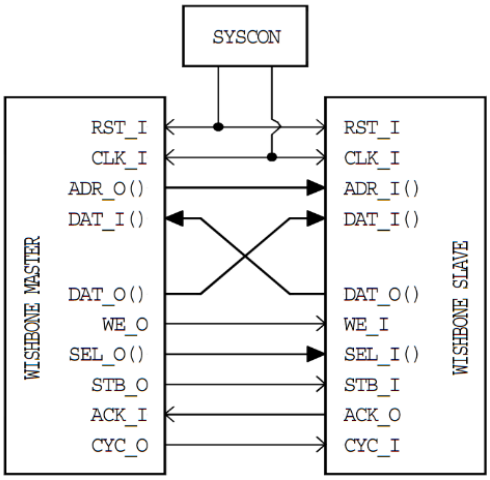
Hình 2.1a mô tả cho kiểu kết nối điểm - điểm. Kết nối điểm - điểm hỗ trợ kết nối trực tiếp hai bên tham gia truyền dữ liệu, cho phép kết nối trực tiếp từ một giao diện MASTER đến một giao diện SLAVE.

Hình 2.1b mô tả cho kiểu kết nối luồng dữ liệu. Kết nối luồng dữ liệu được sử dụng khi dữ liệu được xử lý theo cách tuần tự từ lõi này đến lõi tiếp theo. Mỗi lõi IP trong kiến trúc luồng dữ liệu có cả giao diện MASTER và SLAVE.

Hình 2.1c mô tả cho kiểu kết nối bus chia sẻ. Kết nối bus chia sẻ rất hữu ích để kết nối hai hoặc nhiều MASTER với một hoặc nhiều SLAVE. Một bộ phân xử xác định thời điểm và cách thức cho mỗi MASTER truy cập vào tài nguyên được chia sẻ.

Hình 2.1d mô tả cho kiểu kết nối chuyển mạch chéo. Kết nối chuyển mạch chéo được sử dụng để kết nối hai hoặc nhiều MASTER với một hoặc nhiều SLAVE. Khác với kết nối bus chia sẻ, kết nối chuyển mạch cho phép có nhiều hơn một MASTER để sử dụng kết nối miễn là hai MASTER không truy cập cùng một SLAVE cùng một lúc.

2.2. Mô hình hệ thống kết nối điểm – điểm của bus WISHBONE



Hình 2.2. *Giao diện kết nối điểm – điểm WISHBONE INTERCON*

Hình 2.2 ở trên mô tả mô hình kết nối điểm - điểm WISHBONE (ICN0001a). Nó bao gồm một giao diện MASTER (DMA0001a), một giao diện SLAVE (MEM0002a) và một khối SYSCON (SYC0001a). Khối SYSCON cung cấp xung nhịp Clock và tín hiệu Reset cho hoạt động của hệ thống. Khối MASTER khởi tạo chu kỳ và gửi các yêu cầu truyền nhận dữ liệu với SLAVE. Khối SLAVE tiếp nhận các thông tin, yêu cầu từ MASTER và thực hiện các yêu cầu đó.

WISHBONE thực hiện thay đổi dữ liệu tại cạnh lên của xung Clock và tất cả tín hiệu của WISHBONE đều tác động tích cực mức cao. Điều này nhằm đơn giản hóa việc tích hợp các ngoại vi WISHBONE vào bất kỳ thiết kế nào [4, 7]. Mô tả của các chân tín hiệu sử dụng trong giao diện kết nối của bus WISHBONE được thể hiện trong Bảng 1 dưới đây.

Bảng 1. *Mô tả các tín hiệu trong mô hình sử dụng bus WISHBONE [12]*

Tên tín hiệu	Độ rộng (bit)	Mô tả chức năng
CLK	1	Xung CLK đồng bộ cho hoạt động của hai giao diện.
RST	1	Khởi động lại hệ thống.
CYC	1	Yêu cầu chu kỳ bus.
STB	1	Cho biết một chu kỳ đọc/ghi hợp lệ.
WE	1	Cho phép chế độ đọc (WE = 0) hoặc ghi (WE = 1).
ADR	32	Đường địa chỉ truy xuất.
DAT_I/O	32	Đường dữ liệu vào/ra.
ACK	1	Tín hiệu bắt tay của SLAVE phản hồi lại với tín hiệu STB và CYC từ MASTER, cho biết dữ liệu sẽ được đọc/ghi tại cạnh lên xung CLK kế tiếp.

3. Bus AMBA AXI

AXI là một trong các giao thức bus trong bộ giao tiếp AMBA do hãng ARM phát triển và được sử dụng phổ biến trong các vi điều khiển ARM. Phiên bản đầu tiên của AXI được xuất hiện lần đầu trong AMBA 3.0 ra mắt vào năm 2003. Đến 2010, AMBA 4.0 ra mắt, trong đó có phiên bản thứ 2 của AXI là AXI4 [13].

JTE
HCMUTE
ISSN: 1859-1272

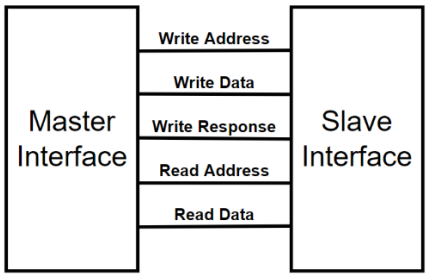
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Giao thức AXI phù hợp với các thiết kế băng thông cao và độ trễ thấp, cung cấp hoạt động tần số cao mà không cần sử dụng các cầu nối phức tạp, bên cạnh đó còn cung cấp tính linh hoạt trong việc triển khai các kiến trúc kết nối và có khả năng tương thích ngược với các giao diện AHB và APB [13, 14].

3.1. Kiến trúc và hoạt động của bus AMBA AXI

AMBA AXI hoạt động dựa trên năm loại kênh độc lập được mô tả cụ thể như Hình 3.1 bên dưới. Mỗi kênh có vai trò, nhiệm vụ khác nhau và việc hoàn thành nhiệm vụ của mỗi kênh không phụ thuộc vào các kênh khác, nhưng chúng vẫn liên quan đến nhau [13]. Năm kênh này gồm:

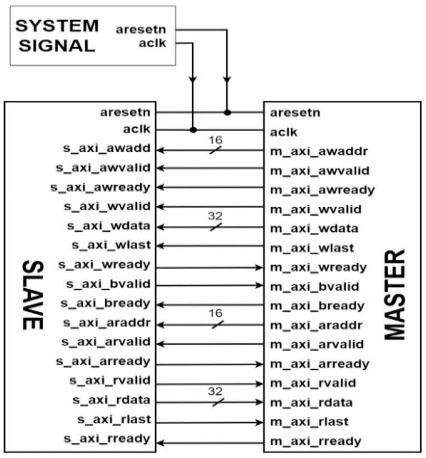
- Kênh địa chỉ đọc (Read Address): truyền thông tin địa chỉ và thông tin điều khiển của một quá trình đọc từ MASTER đến SLAVE.
- Kênh dữ liệu đọc (Read Data): truyền dữ liệu đọc và thông tin phản hồi của một quá trình đọc từ SLAVE đến MASTER.
- Kênh địa chỉ ghi (Write Address): truyền thông tin địa chỉ và thông tin điều khiển của một quá trình ghi từ MASTER đến SLAVE.
- Kênh dữ liệu ghi (Write Data): truyền dữ liệu ghi từ MASTER đến SLAVE.
- Kênh đáp ứng ghi (Write Response): truyền thông tin phản hồi của một giao dịch ghi từ SLAVE đến MASTER.



Hình 3.1. *Các kênh giao tiếp giữa AXI MASTER và SLAVE*

AXI hoạt động dựa theo cơ chế truyền liên tục (burst). Đặc kiểm của cơ chế truyền trong AXI là phía MASTER không phát các địa chỉ trung gian của từng đoạn trong một quá trình truyền mà chỉ phát địa chỉ byte đầu tiên trong một giao dịch, đây chính là địa chỉ của lần truyền dữ liệu đầu tiên. Phía SLAVE dựa trên thông tin điều khiển để xác định địa chỉ của các đoạn từ địa chỉ đầu tiên này. Điều này giúp tăng hiệu suất của hệ thống bus nhưng làm mạch logic xử lý giao tiếp AXI tại phía SLAVE phức tạp hơn vì phải tự tính toán các đoạn địa chỉ [13, 14].

3.2. Mô hình hệ thống kết nối điểm – điểm của bus AMBA AXI



Hình 3.2. *Mô hình kết nối điểm – điểm của bus AMBA AXI*

JTE HCMUTE
ISSN: 1859-1272
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Hình 3.2 trên mô tả hệ thống kết nối điểm – điểm sử dụng kiến trúc bus AMBA AXI, bao gồm một giao diện AXI MASTER, một giao diện AXI SLAVE và SYSTEM SIGNAL. Giao diện AXI MASTER mô phỏng lại các hành vi chính của một MASTER thực hiện như khởi tạo các tín hiệu để kết nối, yêu cầu các chu kỳ truyền nhận dữ liệu với SLAVE, thực hiện đọc/ghi dữ liệu. Giao diện AXI SLAVE là một khối AXI RAM, có chức năng lưu dữ liệu ghi được gửi từ MASTER, tiếp nhận và phản hồi lại các tín hiệu trong quá trình kết nối truyền nhận dữ liệu với MASTER. SYSTEM SIGNAL là khối cấp xung Clock từ hệ thống và tạo tín hiệu Reset cho hoạt động của các giao diện AXI. Mô tả của các chân tín hiệu sử dụng trong giao diện kết nối của bus AMBA AXI được thể hiện trong Bảng 2 dưới đây.

Bảng 2. *Mô tả các tín hiệu cơ bản của bus AMBA AXI [14 ,15]*

Tên tín hiệu	Độ rộng (bit)	Mô tả chức năng
aclk	1	Xung CLK đồng bộ cho hoạt động của toàn hệ thống.
aresetn	1	Tín hiệu RESET tích cực mức THẤP khởi tạo lại cho toàn hệ thống.
axi_awaddr	16	Cho biết địa chỉ của transfer đầu tiên của burst.
axi_awvalid	1	Chỉ ra rằng kênh đang báo hiệu địa chỉ ghi hợp lệ hoặc thông tin điều khiển.
axi_awready	1	Chỉ ra rằng SLAVE sẵn sàng để chấp nhận địa chỉ và các tín hiệu điều khiển liên quan.
axi_wvalid	1	Cho biết dữ liệu ghi hợp lệ và strobes có sẵn.
axi_wdata	32	Dữ liệu ghi.
axi_wlast	1	Cho biết transfer cuối của của burst ghi.
axi_wready	1	Cho biết SLAVE sẵn sàng nhận dữ liệu ghi.
axi_bvalid	1	Cho biết kênh đang thông báo một phản hồi ghi hợp lệ.
axi_bready	1	Cho biết MASTER sẵn sàng nhận một phản hồi ghi.
axi_araddr	16	Kênh địa chỉ đọc đưa địa chỉ của transfer đầu tiên.
axi_arvalid	1	Cho biết kênh đang thông báo địa chỉ đọc hợp lệ và thông tin điều khiển.
axi_arready	1	Cho biết SLAVE sẵn sàng nhận một địa chỉ và các tín hiệu điều khiển liên quan.
axi_rvalid	1	Cho biết kênh đang thông báo dữ liệu đọc hợp lệ.
axi_rdata	32	Dữ liệu đọc.
axi_rlast	1	Cho biết transfer cuối cùng trong burst đọc.
axi_rready	1	Cho biết MASTER sẵn sàng nhận dữ liệu đọc và phản hồi thông tin.

4. Kết quả mô phỏng và đánh giá

4.1. Kết quả mô phỏng

Cấu hình mô phỏng cho hệ thống sử dụng bus WISHBONE và AMBA AXI:

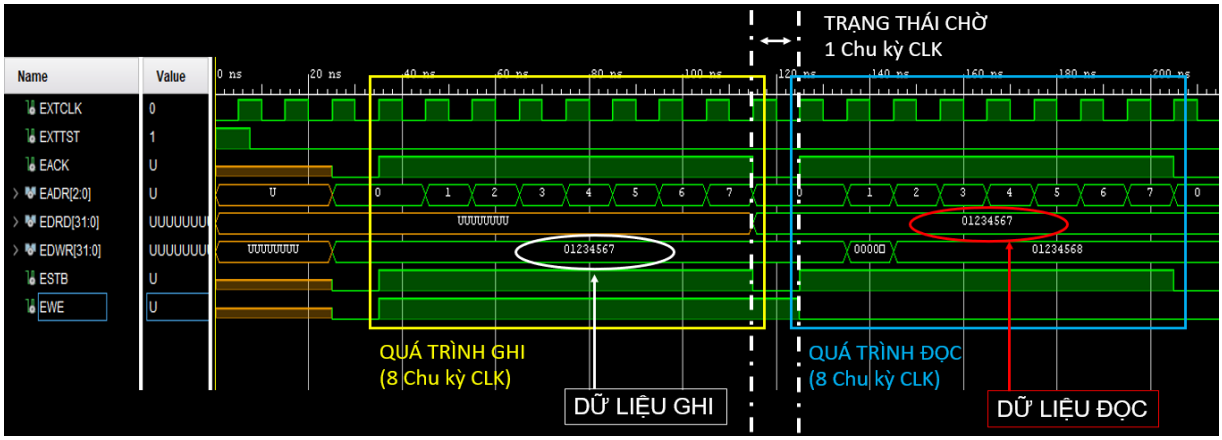
- Giá trị dữ liệu được truyền: 32’h01234567
- Độ dài dữ liệu: 32-bit
- Tần số hoạt động: 100MHz

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Mô phỏng được thực hiện trên phần mềm Xilinx Vivado 2019.1 với máy tính sử dụng hệ điều hành Windows 11, CPU AMD Ryzen 5 3500U và RAM 8GB. Thời gian để phần mềm thực hiện quá trình tổng hợp thiết kế đối với hệ thống bus WISHBONE là 22 giây, với hệ thống bus AMBA AXI là 65 giây. Bộ nhớ RAM sử dụng trong quá trình tổng hợp của hệ thống bus WISHBONE là 1371 MB và đối với hệ thống bus AMBA AXI là 1037 MB [15, 19].

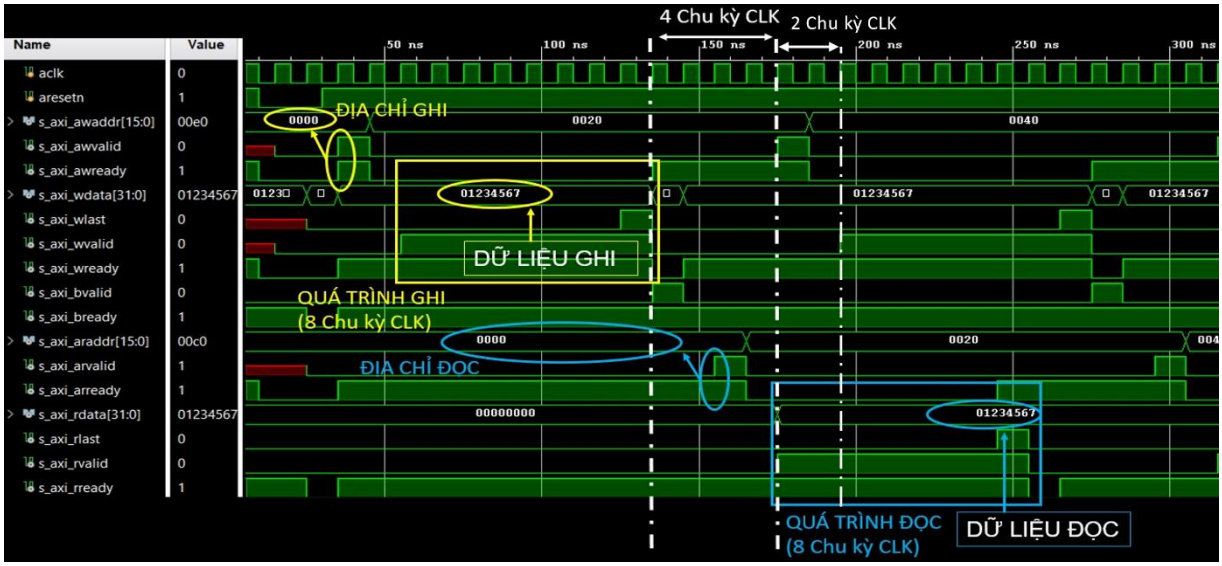
Thiết kế được xây dựng để mô phỏng hoạt động đọc/ghi của hai mô hình kết nối điểm – điểm sử dụng bus WISHBONE và AMBA AXI. Kết quả dạng sóng mô phỏng hoạt động của hệ thống bus WISHBONE và hệ thống bus AMBA AXI được lần lượt thể hiện trong Hình 3.1 và 3.2 dưới đây.



Hình 3.1. *Hoạt động đọc/ghi của bus WISHBONE*

Bus WISHBONE thực hiện một quá trình truyền dữ liệu khi có tín hiệu cho biết một chu kỳ hợp lệ [ESTB] = 1 và tín hiệu phản hồi từ SLAVE xác nhận yêu cầu từ MASTER [EACK] = 1. Tín hiệu [EWE] = 1 cho phép quá trình ghi, [EWE] = 0 cho phép quá trình đọc.

Thời gian thực hiện quá trình đọc/ghi dữ liệu 32'h01234567 của bus WISHBONE là 8 chu kỳ xung Clock cho mỗi quá trình. Thời gian để dữ liệu 32'h01234567 sau khi được ghi xong và bắt đầu được đọc là 1 chu kỳ xung Clock. Bus WISHBONE sử dụng chung kênh truyền cho cả hoạt động đọc và ghi nên sau khi hoàn tất quá trình đọc thì mới có thể tiếp nhận một quá trình ghi dữ liệu mới. Do đó, thời gian sau khi dữ liệu 32'h01234567 được ghi xong cho đến lúc một dữ liệu mới bắt đầu được ghi cần thời gian là 10 chu kỳ xung Clock (1 chu kỳ chờ sau khi ghi xong, 8 chu kỳ quá trình đọc và 1 chu kỳ chờ sau khi đọc xong).



Hình 3.2. *Hoạt động đọc/ghi của bus AMBA AXI*

JTE HCMUTE
ISSN: 1859-1272
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Giao thức AXI quy định quá trình ghi phải trải qua 3 giai đoạn gồm đưa địa chỉ, ghi dữ liệu và phản hồi ghi. Cặp tín hiệu awvalid-awready thông báo địa chỉ ghi hợp lệ. Cặp tín hiệu wvalid-wready cho biết có dữ liệu hợp lệ được truyền và tích cực cho đến khi kết thúc quá trình ghi. Tín hiệu wlast tích cực cho biết transfer cuối cùng của quá trình ghi. Tín hiệu bvalid được phía SLAVE bật để báo hiệu dữ liệu được ghi thành công.

Thời gian thực hiện quá trình đọc/ghi dữ liệu 32'h01234567 của bus AMBA AXI là 8 chu kỳ xung Clock cho mỗi quá trình. Thời gian để dữ liệu 32'h01234567 sau khi được ghi xong và bắt đầu được đọc là 4 chu kỳ xung Clock. Bus AMBA AXI sử dụng các kênh truyền độc lập nhau cho hoạt động đọc và ghi nên quá trình ghi dữ liệu mới có thể được tiếp nhận trong lúc quá trình đọc đang diễn ra miễn là không cùng trên một địa chỉ. Do đó, thời gian sau khi dữ liệu 32'h01234567 được ghi xong cho đến lúc một dữ liệu mới bắt đầu được ghi cần thời gian là 6 chu kỳ xung Clock.

4.2. Tài nguyên sử dụng của hệ thống

Thông số tài nguyên và công suất tiêu thụ trong hệ thống bus WISHBONE và AMBA AXI sử dụng cho việc tổng hợp trên phần cứng Zybo Zynq-7000 [20], mô phỏng sử dụng phần mềm Xilinx Vivado 2019.1, với tần số hoạt động là 100MHz và điện áp là 3.3V, được mô tả lần lượt trong Bảng 3, Bảng 4 và Bảng 5 dưới đây.

Bảng 3. *Thông số tài nguyên hệ thống sử dụng bus WISHBONE và AMBA AXI*

Có sẵn / Sử dụng	LUT (17600)	FF (35200)	BRAM (60)	IO (100)
WISHBONE	39 *(0.22%)*	71 *(0.20%)*	X	69 *(69.0%)*
AMBA AXI	85 *(0.48%)*	131 *(0.37%)*	16 *(26.7%)*	98 *(98.0%)*

Bảng 4. *Công suất tiêu thụ với hệ thống sử dụng bus WISHBONE*

Thành phần		Công suất (W)	Tỷ lệ
Dynamic	Signals	0.001	4%
	Clock	0.001	2%
	Logic	<0.001	1%
	IO	0.035	93%
	Tổng	**0.037**	**29%**
Static		**0.092**	**71%**
Tổng		**0.129**	**100%**

Bảng 5. *Công suất tiêu thụ với hệ thống sử dụng bus AMBA AXI*

Thành phần		Công suất (W)	Tỷ lệ
Dynamic	Signals	0.001	2%
	Clock	0.002	3%
	Logic	<0.001	1%
	IO	0.026	39%
	BRAM	0.038	55%
	Tổng	**0.068**	**42%**
Static		**0.093**	**58%**
Tổng		**0.161**	**100%**

Qua các bảng thống kê tài nguyên và công suất tiêu thụ trên, có thể thấy về ưu điểm của bus WISHBONE là giao diện kết nối yêu cầu ít tín hiệu kết nối nên sử dụng ít tài nguyên hơn và công suất tiêu thụ cũng ít hơn so với bus AMBA AXI.

4.3. Tổng kết

Dựa trên kết quả mô phỏng hoạt động truyền nhận dữ liệu trong mô hình kết nối điểm – điểm sử dụng hai loại bus WISHBONE và AMBA AXI, có thể thấy ưu điểm của giao diện kết nối của bus WISHBONE là giao diện đơn giản, yêu cầu ít tín hiệu để kết nối. Do đó, tài nguyên sử dụng và công suất tiêu thụ của hệ thống bus WISHBONE ít hơn so với của AMBA AXI. Bus AMBA AXI yêu cầu nhiều tín hiệu để kết nối hơn và quá trình xác nhận thông tin gồm nhiều tín hiệu xác nhận giúp đảm bảo cho dữ liệu truyền hợp lệ. Thời gian để một dữ liệu được truyền nhận của bus WISHBONE nhanh hơn so với AMBA AXI. Tuy nhiên, do ưu điểm về các kênh truyền độc lập của bus AMBA AXI nên khi có nhiều hơn một dữ liệu được truyền nhận thì AMBA AXI sẽ cho thời gian và hiệu suất truyền tốt hơn bus WISHBONE.

Những điểm chung trong kiến trúc bus WISHBONE và AMBA AXI:

- Đều là các chuẩn bus mở (Open standard buses), miễn phí, không yêu cầu bản quyền.
- Bộ phân xử do người dùng cuối quyết định loại phân xử (Static Priority, TDMA, Lottery, Round-Robin, Token-passing và CDMA).
- Hỗ trợ truyền theo cơ chế bắt tay (Handshaking) và Burst.
- Tốc độ hệ thống do người dùng cuối xác định, không giới hạn, chỉ phụ thuộc vào thành phần kết nối hoặc phần cứng.

Những đặc điểm khác nhau giữa hai loại kiến trúc bus WISHBONE và AMBA AXI:

Bus	WISHBONE	AMBA AXI
Quyền sở hữu	OpenCores	ARM
Độ rộng đường dữ liệu	8, 16, 32, 64 bit	8, 16, 32, 64, 128, 256, 512, 1024 bit
Độ rộng đường địa chỉ	1–64 bit	32 bit
Kiểu kết nối	Điểm – điểm (Point-to-Point) Luồng dữ liệu (Data Flow) Bus chia sẻ (Shared Bus) Chuyển mạch chéo (Crossbar Switch)	Kết nối dựa trên 5 kênh độc lập Kết nối đa MASTER đa SLAVE thông qua AXI Interconnect
Truyền dữ liệu	Cơ chế bắt tay (Handshaking) Cơ chế Burst	Cơ chế bắt tay (Handshaking) Cơ chế Burst Cơ chế Ống dẫn (Pipelined)
Hoạt động kết nối	Giao diện kết nối yêu cầu ít tín hiệu để kết nối, đơn giản.	Giao diện kết nối yêu cầu nhiều tín hiệu để kết nối.

5. Kết luận

Những so sánh về đặc điểm của hai kiến trúc bus WISHBONE và AMBA AXI đã được thực hiện trong bài viết này. Quá trình mô phỏng và thực nghiệm hoạt động truyền nhận dữ liệu trong mô hình kết nối điểm – điểm sử dụng hai loại kiến trúc bus trên đã được thực hiện trên phần mềm Xilinx Vivado 2019.1. Các kết quả mô phỏng hoạt động truyền nhận dữ liệu của hai kiến trúc bus đã cho thấy ưu và nhược điểm của mỗi loại. Mô hình kết nối sử dụng bus WISHBONE sử dụng ít tín hiệu kết nối hơn, đơn giản, sử dụng ít tài nguyên và cho thời gian truyền nhận dữ liệu tốt hơn đối với trường hợp chỉ có một dữ liệu được truyền nhận. Mô hình kết nối sử dụng bus AMBA AXI cho thời gian và hiệu suất truyền nhận dữ liệu tốt hơn trong trường hợp có nhiều hơn một dữ liệu được truyền nhận do đặc điểm có các kênh truyền độc lập. Bên cạnh đó, tính bảo mật của AMBA AXI cũng cao hơn do có nhiều tín hiệu kiểm tra xác nhận thông tin dữ liệu hợp. Bài viết cũng đã tổng kết và so sánh, chỉ ra những điểm giống và khác nhau của hai loại bus, ưu nhược điểm và ứng dụng của mỗi loại.

TÀI LIỆU THAM KHẢO

[1] Rohita P. Patil and Pratima V. Sangamkar, "A Review of System-On-Chip Bus Protocols", International Journal of Advanced Research in Electrical, *Electronics and Instrumentation Engineering*, Vol. 4, Issue 1, January 2015.
[2] Mohandeep Sharma and Dilip Kumar, "WISHBONE BUS ARCHITECTURE – A SURVEY AND COMPARISON", International Journal of VLSI design & Communication Systems (VLSICS), Vol.3, No.2, April 2012.
[3] R. Usselmann, "OpenCores SoC Bus Review Rev. 1.0", p. 12, January 9, 2001.
[4] Silicore Corporation, WISHBONE SOC Architecture Specification, Revision B.3, USA, 2002.
[5] Chandrala Brijesh A. and Mahesh T. Kolte, "Design and Verification Point-to-Point Architecture of WISHBONE Bus for System-on-Chip", International Journal of Emerging Engineering Research and Technology, Volume 2, Issue 2, PP 155-159, May 2014.
[6] S. R. M. P. P. S. S., "Design and Implementation of WISHBONE Bus Interface Architecture for SoC Integration USING VHDL ON FPGA", IJRITCC, vol. 2, no. 7, pp. 1847–1850, Jul. 2014.
[7] A. K. Swain and K. Mahapatra, "Design and verification of WISHBONE bus interface for System-on-Chip integration," 2010 Annual IEEE India Conference (INDICON), 2010, pp. 1-4, doi: 10.1109/INDCON.2010.5712616.
[8] B. Sasi Rekha, G. Ananta Divya, V. Usha Sai Jyothi, "Design of AXI bus interface modules on FPGA", International Conference on Advanced Communication Control and Computing Technologies, 2016.
[9] M Prasanna Deepu, Prof. R. Dhanabal, "Validation of Transactions in AXI Protocol," 2017.
[10] Shubhi Sharma, Vidyadhar Jambhale, Abhijeet Shinde and S.Ravi, "Transaction based AMBA AXI bus interconnect in Verilog", International Research Journal of Engineering and Technology, 2017.
[11] Silicore Corporation, WISHBONE Public Domain Library for VHDL, October 8, 2001.
[12] Richard Herveille, "WISHBONE System-on- Chip (SoC) Interconnection Architecture for Portable IP Cores Revision: B.3", Open Cores Organization, September 7, 2010.
[13] ARM, AMBA AXI and ACE Protocol Specification, June 3, 2011.
[14] Xilinx, AXI Interconnect v2.1: LogiCORE IP Product Guide, PG059 (v2.1), 2017.
[15] Xilinx, Vivado Design Suite: AXI Reference Guide, UG1037 (v4.0) July 15, 2017.
[16] Xilinx, Zynq-7000 SoC Technical Reference Manual, UG585 (v1.13) April 2, 2021.
[17] Xilinx, Zynq-7000 SoC Data Sheet: Overview, DS190 (v1.11.1) July 2, 2018.
[18] Xilinx, Zynq-7000 SoC: Embedded Design Tutorial, UG1165 (2019.2) October 30, 2019.
[19] Xilinx, Vivado Design Suite User Guide, UG896 (v2019.2) March 3, 2020.
[20] Digilent, ZYBO™ FPGA Board Reference Manual, Rev. B, February 27, 2017.

Nguyen Quang Anh Tuan is currently a student at the Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. His main research interests include wireless communication networks and FPGA-based designs for DSP applications.

Nguyen Van Tuan is currently a student at the Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. His main research interests include wireless communication networks and FPGA-based designs for DSP applications.

Vo Phan Man Dat is currently a student at the Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. His main research interests include wireless communication networks and FPGA-based designs for DSP applications.

Van Ba Huy is currently a student at Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. His main research interests include design/verify IP and layout design for ASIC/SoC at block level.

Truong Quang Phuc was born in Can Tho City, Vietnam. He received the B.Eng degree in Electronics and telecommunication engineering and the M.Eng degree in Electronics engineering from the Ho Chi Minh City University of Technology and Education, Vietnam, in 2011 and 2014, respectively. Currently, He is with the Faculty of Electrical and Electronics Engineering, Ho ChiMinh City University of Technology and Education, Vietnam as a Research Assistant and Ph.D. student. His research interests include convex optimization techniques, heterogeneous networks, Internet of Things, and Intelligent Reflecting Surfaces (IRS).

Nguyen Ngo Lam is currently a lecturer at the Faculty For High Quality Training, Ho Chi Minh City University of Technology and Education . He received his Bachelor and Master degree in radio and electronics engineering from the Ho Chi Minh City University of Technology, Vietnam in 2000 and 2004 respectively. His research interests include wireless communication, data communication, digital signal processing, computer - aided engineering.