

THỰC HIỆN NETWORK-ON-CHIP (NoC) TRÊN NỀN FPGA

IMPLEMENTATION OF A NETWORK-ON-CHIP (NoC) ON FPGA PLATFORM

PHẠM NGỌC NAM, *Trường ĐHBK Hà Nội*
HUỠNH VIỆT THẮNG, *Trường ĐHBK, ĐHQĐN*

TÓM TẮT

Bài báo sẽ giới thiệu mô hình mạng trên chip NoC (Network-on-Chip) ứng dụng trong thiết kế một hệ thống trên chip SoC (System-on-Chip), từ đó đề xuất một kiến trúc mạng trên chip cơ sở (đặt tên là BASIC-NoC), tổng hợp mạng bằng VHDL để thực hiện trên FPGA, việc mô phỏng và đánh giá hiệu năng mạng được thực hiện bằng phần mềm ModelSIM.

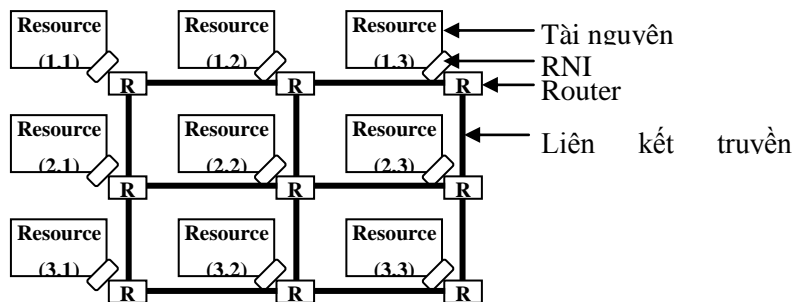
ABSTRACT

This article introduces the Network-on-Chip (NoC) model used for System-on-Chip (SoC) applications. A basic NoC structure (BASIC-NoC) is proposed to synthesize the network using VHDL and implemented on the FPGA platform. Simulation and performance results are also presented.

1. GIỚI THIỆU

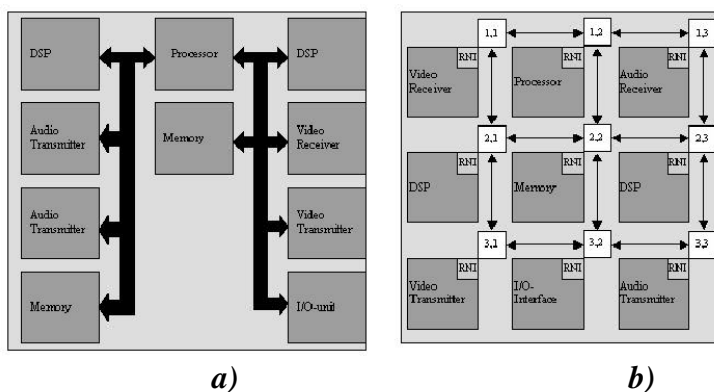
Trong qui trình thiết kế chip, bốn khía cạnh cần quan tâm đó là: quá trình tính toán (*computation*), bộ nhớ (*memory*), I/O và truyền thông (*communication*), [4]. Khi công suất xử lý tăng và các ứng dụng tập trung nhiều dữ liệu ngày càng phổ biến thì việc giải quyết các thách thức của quá trình truyền thông (*communication*) bên trong một hệ thống trên chip SoC (*System-on-Chip*) đang được đặt ra hơn bao giờ hết. Hai giải pháp thiết kế truyền thông thường dùng cho SoC là sử dụng các đường dây riêng biệt hoặc sử dụng các bus chung cho toàn hệ thống, tuy nhiên các giải pháp này bộc lộ các hạn chế như kém linh hoạt và khó thực hiện khi nhu cầu truyền thông hoặc số lượng tài nguyên gia tăng.

Mạng trên chip, Network on Chip (NoC), ra đời với ý định sẽ giải quyết các khó khăn này bằng cách thực hiện một mạng truyền thông (*communication network*) bao gồm các micro-router/switch liên kết các tài nguyên hệ thống [1], [3], [4]. Hầu hết các khái niệm cơ sở và các thuật ngữ áp dụng cho NoC đều được bắt nguồn từ mạng máy tính. Cách thức liên lạc này mang lại tính mềm dẻo, cho phép khả năng chuẩn hóa và sử dụng lại các kiến trúc truyền thông đã được thiết kế và tối ưu, giúp giảm chi phí và thời gian thiết kế, cũng đồng nghĩa với việc thời gian đưa sản phẩm ra thị trường (*time to market*) được rút ngắn. NoC cho phép chia sẻ tài nguyên, nâng cao băng thông bằng việc chia sẻ các kênh truyền trên mạng, giảm nhỏ năng lượng tiêu thụ do các dây dẫn được thu ngắn, việc phân lớp khi thiết kế và khả năng sử dụng lại các tài nguyên là những đặc điểm thuận lợi khi thiết kế dựa trên NoC.



Hình 1. Kiến trúc một NoC đơn giản 3 x 3 tài nguyên

Hình 1 trình bày một kiến trúc NoC với topology dạng lưới (*mesh*) 3×3 gồm 9 tài nguyên liên kết với nhau, dữ liệu trao đổi giữa các tài nguyên được đóng thành các packet (gói) và chuyển đi nhờ các router thông qua các liên kết truyền thông, mỗi tài nguyên được gắn chỉ với một router và khối RNI (*Resource Network Interface*) đóng vai trò giao diện giữa tài nguyên và router. Hình 2 minh họa hai giải pháp thiết kế SoC khác nhau của cùng một sản phẩm: thiết kế dựa trên kiến trúc bus (*bus-based*) và thiết kế dựa trên kiến trúc NoC (*NoC-based*).



Hình 2. Hai giải pháp thiết kế SoC: a) Thiết kế truyền thống sử dụng bus chung b) Thiết kế sử dụng kiến trúc NoC

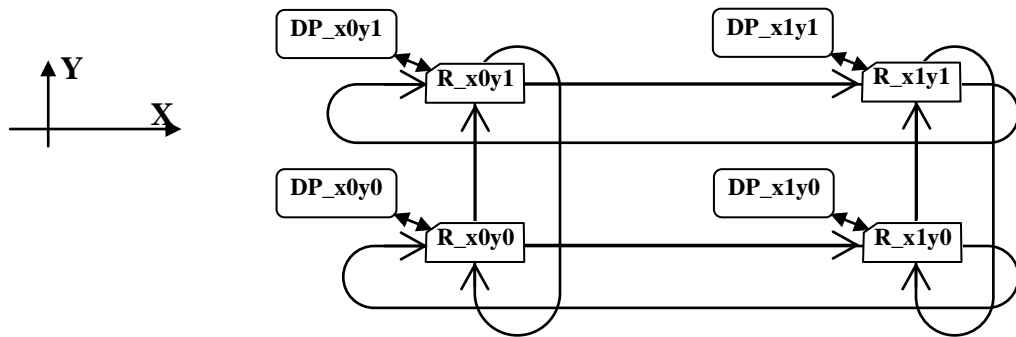
Hiện nay, các nghiên cứu về NoC đang được thực hiện bởi nhiều nhóm nghiên cứu của các công ty, trường đại học, viện nghiên cứu trên thế giới theo nhiều hướng khác nhau [3],[4]. Việc ứng dụng mô hình mạng máy tính cho SoC đã mở ra hướng phát triển đầy hứa hẹn trong thiết kế các hệ thống nhúng. Tại Việt Nam hướng nghiên cứu về NoC vẫn còn rất mới và chưa được khai phá, chính vì vậy nghiên cứu về NoC và thực hiện NoC trên nền phần cứng FPGA sẽ là một trong những bước đi tiên phong về nghiên cứu và thiết kế mạng trên chip.

2. XÂY DỰNG BASIC-NoC

Dựa trên cơ sở lý thuyết về *Interconnection Networks* [1], [2], [3], nghiên cứu này sẽ tiến hành xây dựng một kiến trúc mạng trên chip cơ sở, tổng hợp mạng bằng ngôn ngữ VHDL sử dụng *ISE Foundation 9.1i* của hãng Xilinx, mô phỏng hoạt động của mạng bằng *ModelSIM* và đánh giá kết quả. Việc thực hiện mạng trên phần cứng FPGA sẽ được tiến hành trong thời gian đến. Mạng được xây dựng sẽ là một kiến trúc cơ sở cho việc phát triển các hệ thống trên chip dựa trên FPGA (*FPGA-based SoC*) và có thể được mở rộng trong những nghiên cứu tiếp theo, vì vậy mạng được đặt tên là **BASIC-NoC**.

2.1. Topology mạng

BASIC-NoC có topology dạng **Torus 2 x 2** [1],[2], bao gồm 4 tài nguyên nối với 4 router theo sơ đồ khối được trình bày trên *Hình 3*.



Hình 3. Sơ đồ khối BASIC-NoC

Trong kiến trúc này các router lần lượt được ký hiệu là R_{x0y0} , R_{x0y1} , R_{x1y0} và R_{x1y1} với các tọa độ tương ứng theo chiều ngang và chiều đứng của các router lần lượt là $(0,0)$, $(0,1)$, $(1,0)$ và $(1,1)$. Các tọa độ này cũng chính là địa chỉ của các router trong mạng. Mỗi router được gắn với một tài nguyên hay còn gọi là một đơn vị xử lý dữ liệu **Data Processor (DP)** được ký hiệu tương ứng là DP_{x0y0} , DP_{x0y1} , DP_{x1y0} và DP_{x1y1} , địa chỉ các bộ xử lý này cũng chính là địa chỉ của router.

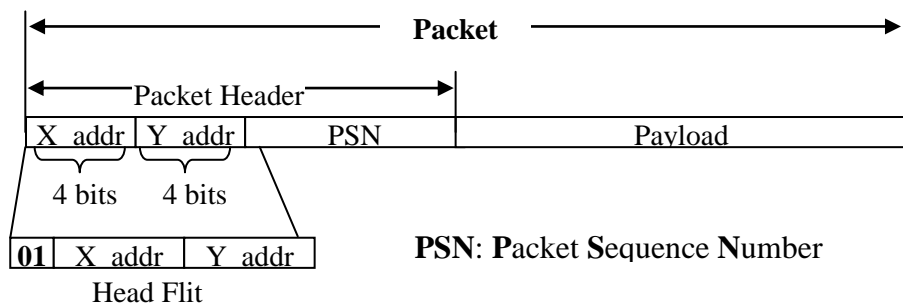
2.2. Cơ chế định tuyến (Routing)

Các DP trao đổi dữ liệu với nhau bằng cách truyền packet thông qua các router. Mỗi packet có chứa địa chỉ đích trong phần header và địa chỉ này được router sử dụng trong quá trình định tuyến để chuyển packet đến đích chính xác. Vì đây là mạng đơn hướng, nếu tính theo phương X hay theo phương Y các router chỉ chuyển packet theo một hướng xác định nên giải thuật định tuyến được áp dụng trong mạng này giải thuật định tuyến XY (*XY Routing*) [1], [2], [3], đây là một giải thuật thuộc lớp giải thuật định tuyến theo chiều (*Dimension Ordered Routing*) được sử dụng với mục đích đảm bảo dễ dàng mở rộng mạng khi có yêu cầu gia tăng số lượng tài nguyên.

2.3. Cơ chế điều khiển luồng (Flow Control)

Cơ chế điều khiển luồng đảm bảo việc cấp phát các tài nguyên sao cho các packet có thể đi qua mạng từ nguồn đến đích một cách chính xác. Packet được phân chia thành các đơn vị truyền tải nhỏ hơn gọi là các Flit và được chuyển qua mạng. Flit là đơn vị cơ bản để thực hiện việc cấp phát băng thông và bộ đệm trong router, có 3 loại flit được sử dụng:

Hình 4. Khuôn dạng của packet được sử dụng trong BASIC-NoC

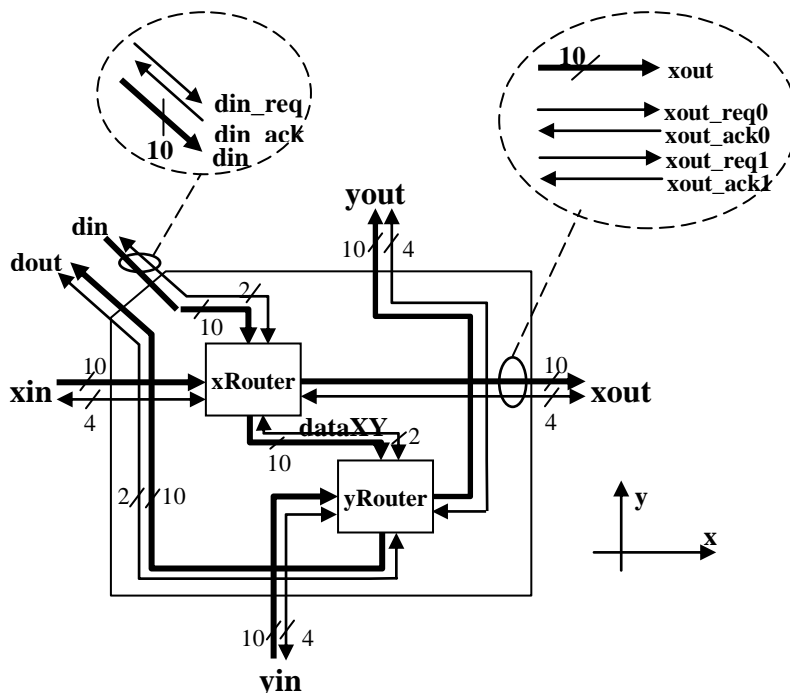


Head Flit, Body Flit và Tail Flit hoặc kết hợp của các loại flit này. Khuôn dạng của một packet được cho trên *Hình 4*. Trong nghiên cứu này sử dụng cơ chế điều khiển luồng Wormhole Routing kết hợp với việc sử dụng 2 kênh ảo Virtual Channel (VC) cho kiến trúc router [1], [2].

Với cơ chế điều khiển luồng Wormhole Routing việc truyền thông giữa các router trong mạng được tiến hành tuần tự theo từng Flit. Độ rộng của bus dữ liệu phải tương ứng với kích thước của Flit. Để tiết kiệm tài nguyên BASIC-NoC sử dụng các đơn vị dữ liệu có kích thước nhỏ nhất là 8 bit, để phân biệt giữa các loại Flit cần sử dụng thêm 2 bit điều khiển, suy ra kích thước một Flit là 10 bit (bao gồm 8 bit dữ liệu và 2 bit điều khiển), đây cũng chính là độ rộng bus dữ liệu trong BASIC-NoC.

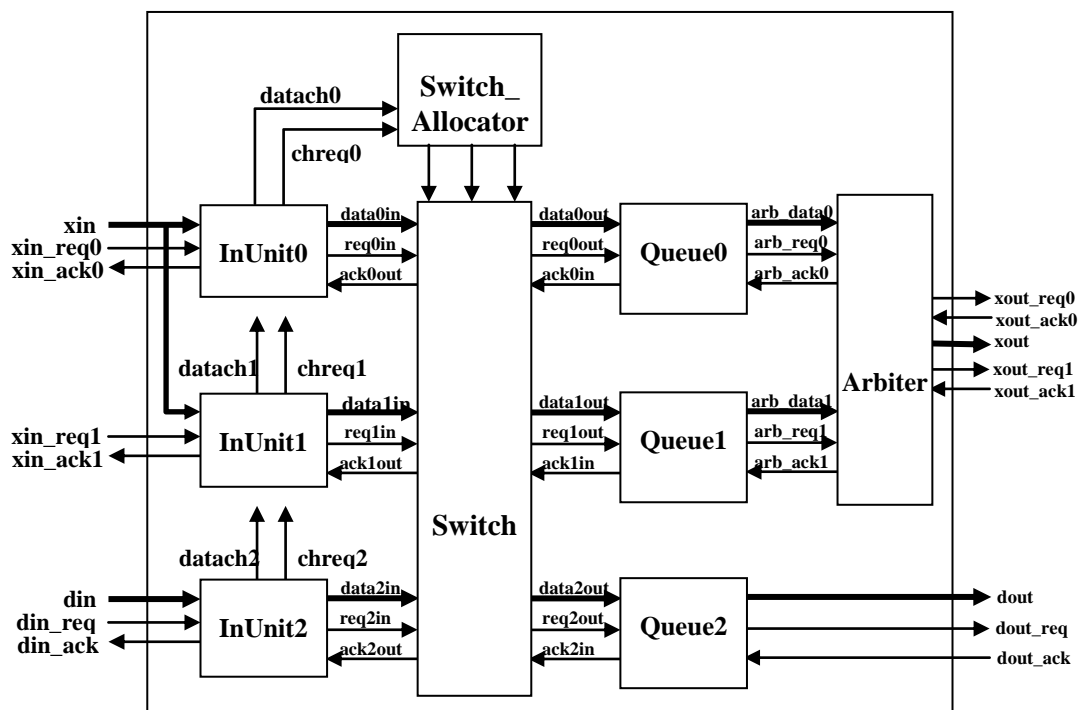
2.4. Kiến trúc Router

Trên cơ sở lựa chọn topology mạng, cơ chế định tuyến và điều khiển luồng, kiến trúc Virtual Channel Router cho BASIC-NoC được xây dựng bao gồm 2 Sub_Router có kiến trúc giống hệt nhau là **xRouter** và **yRouter** với sơ đồ khối được trình bày trên *Hình 5*.



Hình 5. Kiến trúc Virtual Channel Router cho BASIC-NoC

Giải pháp xây dựng router từ các Sub_Router giúp quá trình thiết kế trở nên dễ dàng, đặc biệt thuận lợi khi tổng hợp thiết kế BASIC-NoC bằng ngôn ngữ VHDL. Với giải pháp này, mỗi Sub_Router đảm nhận chức năng định tuyến và điều khiển luồng cho packet theo một phương xác định: xRouter chuyển packet theo phương X còn yRouter chuyển packet theo phương Y. Sơ đồ khối của Sub_Router được



trình bày trên Hình 6.

Hình 6. Sơ đồ khối Sub_Router

3. TỔNG HỢP BASIC-NoC BẰNG VHDL

Trên cơ sở mô hình mạng được xây dựng, việc tổng hợp BASIC-NoC sử dụng ngôn ngữ mô tả phần cứng VHDL được thực hiện bằng bộ công cụ ISE Foundation 9.1i của hãng Xilinx, sử dụng FPGA Xilinx Spartan 3E XC3S500E với tài nguyên khoảng 500 ngàn cổng. Sau khi chạy chương trình tổng hợp, mạng chiếm khoảng 61% tài nguyên phần cứng của chip, kết quả chi tiết được trình bày ở Bảng 1.

Device Utilization Summary (estimated values)				
No#	Logic Utilization	Used	Available	Utilization
1	Number of Slices	2873	4656	61 %
2	Number of Slice Flip Flops	2793	9312	29 %
3	Number of 4 input LUTs	4076	9312	43 %
4	Number of bonded IOBs	26	232	11 %
5	Number of GCLKs	1	24	4 %

Bảng 1. Kết quả tổng hợp BASIC-NoC sử dụng FPGA Spartan3E XC3S500E của Xilinx

4. KẾT QUẢ MÔ PHỎNG

Sau khi tổng hợp phần cứng BASIC-NoC, quá trình mô phỏng và đánh giá hiệu năng mạng được thực hiện bằng cách truyền một số lượng lớn packet qua mạng, đo tổng thời gian từ lúc bắt đầu truyền packet đầu tiên cho đến khi nhận được hoàn toàn packet cuối cùng, từ đó tính được trễ trung bình và thông lượng trung bình (*Average Throughput*) của mạng.

Nghiên cứu này thực hiện hai mô phỏng truyền dữ liệu đồng thời trên mạng:

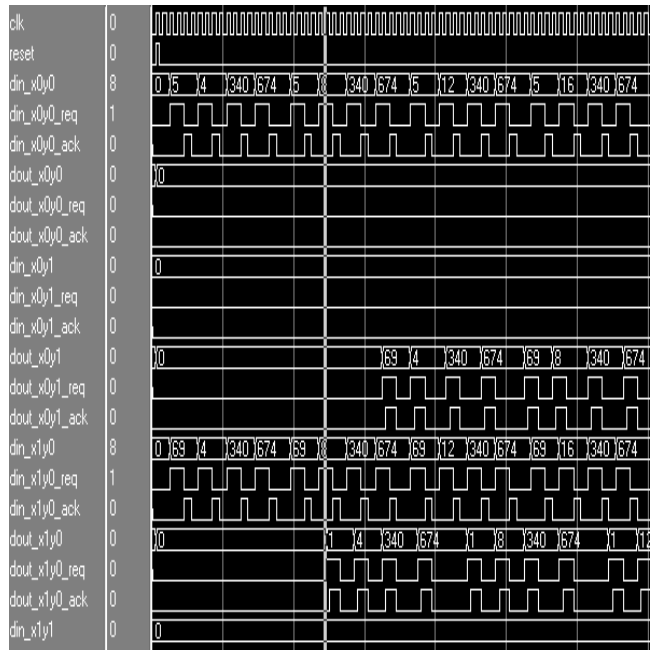
- **Mô phỏng 1:** Truyền dữ liệu từ router R_x0y0 đến router R_x1y0, các packet sẽ chỉ đi từ router R_x0y0 đến router R_x1y0.
- **Mô phỏng 2:** Truyền dữ liệu từ router R_x1y0 đến router R_x0y1, các packet sẽ phải đi từ router R_x1y0 qua router R_x0y0, sau đó được chuyển đến router R_x0y1.

Trong cả hai mô phỏng được thực hiện đều không có sự tranh chấp về tài nguyên đường truyền với mục đích tính toán giá trị thông lượng lớn nhất của mạng.

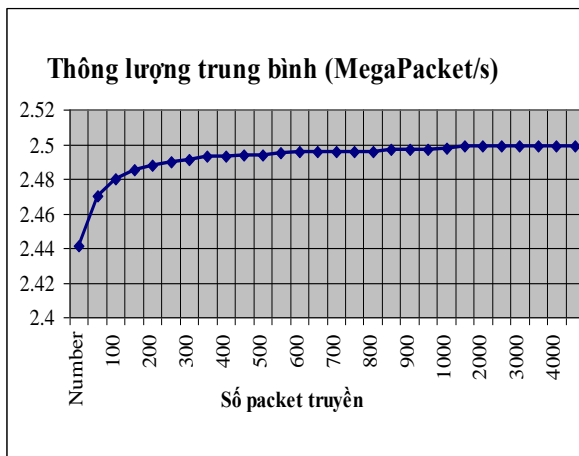
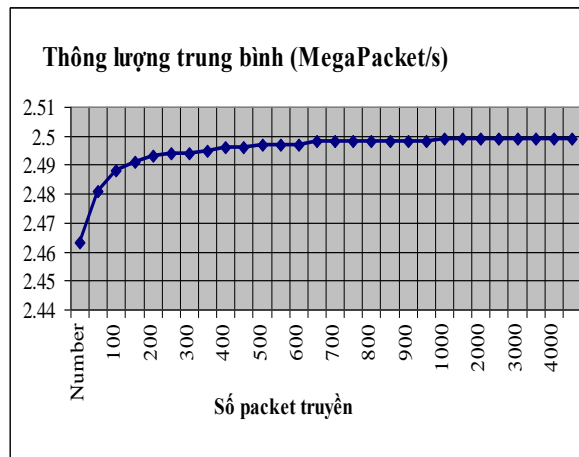
Trong cả hai mô phỏng: thực hiện việc ghi lại thời điểm bắt đầu truyền packet từ router nguồn và thời điểm nhận thành công packet tại router đích, từ đó tính được thời gian truyền-nhận trung bình của một packet, nghĩa là tính được giá trị thông lượng trung bình của mạng.

Các tham số dùng mô phỏng: truyền liên tục 5000 packet, mỗi packet gồm 4 Flit, mỗi Flit có độ rộng 10 bit, tần số xung clock cho toàn mạng $f = 50$ MHz, thời gian nghỉ giữa 2 Flit trong một packet là 1 chu kỳ xung clock, thời gian nghỉ giữa 2 packet liên tiếp là 2 chu kỳ xung clock. Hình 7 biểu diễn dạng sóng mô phỏng trong cửa sổ của chương trình ModelSIM.

Hình 8 trình bày kết quả mô phỏng biểu diễn sự phụ thuộc của giá trị thông lượng trung bình theo số packet được truyền của hai mô phỏng.



Hình 7. Dạng sóng mô phỏng trong



a) Mô phỏng 1: Truyền từ R_{x0y0} đến R_{x1y0} b) Mô phỏng 2: Truyền từ R_{x1y0} đến R_{x0y1}

Hình 8. Quan hệ giữa thông lượng trung bình (MegaPacket/s) theo số packet được truyền

Từ kết quả trên Hình 8 chúng ta rút ra những nhận xét sau đây:

- Trong cả hai mô phỏng: ban đầu khi số lượng packet truyền nhỏ thông lượng của mạng nhỏ, sau đó thông lượng của mạng tăng khi số packet truyền tăng, thông lượng đạt giá trị ổn định khi số packet truyền đủ lớn từ 1000 packet trở lên.
- Khi mạng ở trạng thái ổn định thông lượng trung bình của hai liên kết truyền thông trong hai mô phỏng đạt giá trị xấp xỉ bằng nhau vào khoảng **2.5 MegaPacket/s** ($=2.5 \times 10^6$ packet/s) = **2.5 x 4 x 10 (Mbps) = 100 (Mbps)**.
- Nếu toàn mạng có N liên kết truyền thông diễn ra đồng thời giữa các tài nguyên thông lượng của toàn mạng sẽ là: **N x 2.5 (MegaPacket/s) = N x 100 (Mbps)**.

5. KẾT LUẬN

Kết quả mô phỏng cho thấy kiến trúc mạng trên chip rất thích hợp cho các ứng dụng đa phương tiện (*multimedia*) yêu cầu tốc độ xử lý cao, lượng dữ liệu trao đổi nhiều và băng thông lớn, cũng như rất thích hợp cho các ứng dụng dựa trên xử lý song song. Kiến trúc mạng trên chip cơ sở BASIC-NoC sẽ là nền tảng cho việc triển khai các ứng dụng cho các hệ thống trên một chip SoC thực hiện trên nền phần cứng FPGA.

TÀI LIỆU THAM KHẢO

- [1].William James Dally and Brian Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann Publishers, San Francisco, 2004
- [2].Nikolay Kavaldjiev and Gerard J.M. Smit, *A Survey of Efficient On-Chip Communications for SoC*, Department of EEMCS, University of Twente, 2002.
- [3].José C. Prats Ortiz, *Design of Components for A NoC-Based MPSoC Platform: Adding a shared memory node to the mNoC*, Master of Science Thesis, Eindhoven University of Technology, Department of Electrical Engineering, 2005.
- [4].Tobias Bjerregaard and Shankar Mahadevan, *A Survey of Research and Practices of Network-on-Chip*, ACM Computing Surveys, Vol. 38, March 2006.
- [5].Pong P. Chu, *RTL Hardware Design Using VHDL – Coding for Efficient, Portability and Scalability*, John Wiley and Sons, Inc. Publication, 2006.