BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT

THÀNH PHỐ HỒ CHÍ MINH

CÔNG TRÌNH NGHIÊN CỨU KHOA HỌC CẤP TRƯỜNG

XÂY DỰNG MÔ HÌNH MẠNG TRÊN CHIP (NOC-NETWORK ON CHIP) ỨNG DỤNG TRONG THIẾT KẾ VI MẠCH

MÃ SỐ: T2014-26

SKC005567

Tp. Hồ Chí Minh, 2014

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH

BÁO CÁO TỔNG KẾT

ĐỀ TÀI KH&CN CẤP TRƯỜNG

XÂY DỰNG MÔ HÌNH MẠNG TRÊN CHIP (NOC-NETWORK ON CHIP) ỨNG DỤNG TRONG THIẾT KẾ VI MẠCH

Mã số: T2014-26

Chủ nhiệm đề tài:
Ths, Phạm Văn Khoa

TP. HCM, 11/2014

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ

BÁO CÁO TỔNG KẾT

ĐỀ TÀI KH&CN CẤP TRƯỜNG

XÂY DỰNG MÔ HÌNH MẠNG TRÊN CHIP (NOC-NETWORK ON CHIP) ỨNG DỤNG TRONG THIẾT KẾ VI MẠCH

Mã số: T2014-26

Chủ nhiệm đề tài:
Ths, Phạm Văn Khoa

TP. HCM, 11/2014

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH

BÁO CÁO TÓM TẮT
ĐỀ TÀI KH&CN CẤP TRƯỜNG

XÂY DỰNG MÔ HÌNH MẠNG TRÊN CHIP (NOC-NETWORK ON CHIP) ỨNG DỤNG TRONG THIẾT KẾ VI MẠCH

Mã số: T2014-26

Chủ nhiệm đề tài:
ThS, Phạm Văn Khoa

TP. HCM, 11/2014

MỤC LỤC

MỤC LỤC **i**
DANH SÁCH CHỮ VIẾT TẮT **iv**
DANH SÁCH CÁC HÌNH **v**
DANH SÁCH CÁC BẢNG **vii**
CHƯƠNG 1 TỔNG QUAN **1**
1.1 Mô hình SoC đối với các thiết kế đa lõi 1
1.2 Xu thế phát triển và các vấn đề tồn tại về kết nối trên SoC 2
1.3 Giải pháp và sự cần thiết của nền tảng NoC cho các thiết kế SoC 4
1.4 Mục tiêu của đề tài 7
1.5 Phương pháp nghiên cứu 8
1.6 Bố cục trình bày 9
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT **10**
2.1 Mạng trên chip 10
2.2 Sự phát triển của nền tảng SoC 11
2.2.1 Mô hình kết nối trực tiếp 11
2.2.2 Mô hình bus 11
2.2.3 Mô hình mạng trên chip 12
2.3 Các vấn đề giao tiếp mạng 13
2.3.1 Mô hình truyền thông phân lớp 13
2.3.2 Topo mạng 14
2.3.3 Kỹ thuật chuyển mạch 15
2.3.4 Kênh ảo 21
2.3.5 Định tuyến 23
2.4 Các thành phần trên hệ thống ứng dụng NoC 26
2.4.1 Resource 26
2.4.2 Resource Network Interface 27

2.4.3 Router 27
2.5 Công cụ phát triển 28
2.5.1 Vai trò FPGA trong thiết kế hệ thống số.....28
2.5.2 Công cụ phát triển Quartus.....30
2.5.3 Công cụ mô tả phần cứng VHDL.....31
2.5.4 Công cụ Design Compiler 31
2.5.5 Thư viện thiết kế.....32
CHƯƠNG 3 THIẾT KẾ MÔ HÌNH ỨNG DỤNG MẠNG TRÊN CHIP 34
3.1 Thiết kế kiến trúc router 34
3.1.1 Kiến trúc bên ngoài của router 34
3.1.2 Kiến trúc bên trong của router.....39
3.2 Thiết kế thành phần giao tiếp mạng 48
3.2.1 Các biến thể của RNI 50
3.2.2 Các vấn đề đối mặt khi thiết kế RNI 50
3.2.3 Thiết kế cấu trúc slave RNI.....54
3.2.4 Thiết kế cấu trúc master RNI 57
3.3 Thiết kế tài nguyên mạng 59
3.3.1 Khối switch ngõ vào.....59
3.3.2 Khối bộ nhớ (dummy memory).....60
3.3.3 Khối xử lý (dummy processor) 61
3.3.4 Khối traffic counter và giao tiếp uart 62
CHƯƠNG 4 ĐÁNH GIÁ THIẾT KẾ 64
4.2 Đánh giá hoạt động thực nghiệm của mô hình 64
4.2 Đánh giá tài nguyên của thiết kế trên FPGA 70
4.3 Đánh giá thiết kế về mặt công suất, diện tích và thời gian 72
4.3.1 Công suất tiêu thụ 72
4.3.2 Tần số hoạt động 74
4.3.3 Diện tích 75
CHƯƠNG 5 KẾT LUẬN 79
5.1 Kết quả đạt được 79
5.2 Phần khuyến nghị 80

5.2.1 Những tồn tại trong đề tài 80
5.2.2 Các giải pháp cải tiến thiết kế 81
5.2.2.1 Giảm số lượng các thanh ghi và các cổng logic.....81
5.2.2.2 Độ rộng dữ liệu.....81
5.2.2.3 Bộ đệm trong các RNI.....82
5.2.2.4 Kích thước mạng thay đổi.....82
5.2.2.5 Bidirectional router.....82
5.2.2.6 Gói tin chứa đa flit.....83
5.2.2.7 Cơ chế Wormhole.....83
5.2.2.8 Cơ chế kênh ảo.....83
5.2.2.9 Thuật toán định tuyến.....84
5.2.2.10 Độ ưu tiên trong việc định tuyến.....84

DANH SÁCH CHỮ VIẾT TẮT

NoC	Network on Chip
SoC	System on Chip
RNI	Resource Network Interface
IP	Intellectual Property
CPU	Central Processing Unit
AMBA	Advanced Microcontroller Bus Architecture
MPSoC	Multiprocessor System-on-Chip
FPGA	Field-programmable gate array
DC	Design Compiler
FSM	Finite State Machine
DOR	Demension Ordered Routing
QoS	Quality of Service
DSP	Digital Signal Processing
ASIC	Application Specific Integrated Circuit
IC	Integrated Circuit
HDL	Hardware Description Language
RTL	Register Transfer Level
UART	Universal Asynchronous Receiver-Transmitter
IDE	Integrated Development Environment
VHDL	Very High Speed Integrated Circuit HDL
IEEE	Institute of Electrical and Electronic Engineer
EDA	Electronic Design Automation
ID	Identification
OSI	Open Systems Interconnection

DANH SÁCH CÁC HÌNH

Hình 1.1: Các thành phần trong một SoC 1
Hình 1.2: Kiến trúc hệ thống dựa trên mô hình bus AMBA [5] 3
Hình 1.3: Phương pháp thiết kế NoC đối với MPSoC có tích hợp lớn [2] 4
Hình 1.4: Mô hình SoC dựa trên nền tảng truyền thông NoC [7] 5

Hình 2.1: Truyền thông trên SoC dựa trên mô hình kết nối trực tiếp [7] 11
Hình 2.2: Truyền thông trên SoC dựa trên mô hình bus [7] 12
Hình 2.3: Truyền thông trên SoC dựa trên mô hình mạng trên chip [7] 13
Hình 2.4: Topo dạng vòng, lưới 2 chiều và dạng torus 14
Hình 2.5: Topo dạng lưới với kích thước 2*3 [12] 15
Hình 2.6: Thiết kế tổng quát bên trong một switch [2] 16
Hình 2.7: Kết nối giữa 2 router với 4 router trung gian [2] 17
Hình 2.8: Kỹ thuật chuyển mạch store-and-forward [3] 18
Hình 2.9: Virtual Cut-Through trong trường hợp không có trì hoãn [3] 18
Hình 2.10: Virtual Cut-Through trong trường hợp có trì hoãn [3] 19
Hình 2.11: Message được chia nhỏ thành các flit để di chuyển trên mạng [3] 19
Hình 2.12: Phương pháp điều khiển luồng wormhole [3] 20
Hình 2.13: 4 kênh ảo chia sẻ một kênh truyền vật lý [3] 21
Hình 2.14: Sơ đồ các phương pháp định tuyến [6] 23
Hình 2.15: Đường đi của gói tin theo cơ chế định tuyến OXY [1] 25
Hình 2.16: Gói tin di chuyển mọi hướng và chỉ hướng theo chiều X-Y 25
Hình 2.17: Kiến trúc cơ bản của router [2] 28
Hình 2.18: Thể hiện bên trong của một FPGA [20] 29
Hình 2.19: Design flow của Quartus [21] 30
Hình 2.20: Sơ đồ mô tả chức năng của Design Compiler 32

Hình 3.1: Thiết kế router với 2 kênh truyền vật lý vào/ra tại mỗi hướng 35
Hình 3.2: Topo dạng lưới 2 chiều với 16 node mạng 36
Hình 3.3: Định tuyến Oxy cho mô hình dạng lưới 2 chiều 37
Hình 3.4: Sơ đồ các bước xử lý đối với một router cơ bản [1] 40
Hình 3.5: Kiến trúc tổng quát bên trong của router [1] 41
Hình 3.6: Kiến trúc tổng quát bên ngoài của một router 42
Hình 3.7: Cấu trúc khối bộ đệm và luồng dữ liệu tại ngõ vào router 42
Hình 3.8: Độ ưu tiên các cổng trên router 45
Hình 3.9: Cấu trúc khối chuyển mạch router 47
Hình 3.10: Mô hình hoá các tín hiệu trên router 47
Hình 3.11: Vị trí của RNI trên hệ thống mạng trên chip [12] 48
Hình 3.12: Thành phần và kết nối của RNI trong hệ thống mạng trên chip 49
Hình 3.13: Giao diện giao tiếp giữa master RNI với resource 52
Hình 3.14: Giao diện giao tiếp giữa slave RNI với resource 52
Hình 3.15: Giao diện giao tiếp giữa master RNI với router 53
Hình 3.16: Giao diện giao tiếp giữa slave RNI với router 53
Hình 3.17: Cấu trúc tổng quát bên ngoài của slave RNI 54
Hình 3.18: Cấu trúc tổng quát bên ngoài của master RNI 57
Hình 3.19: Cấu trúc tổng quát bên ngoài của switch 59
Hình 3.20: Cấu trúc tổng quát bên ngoài của dummy memory 60
Hình 3.21: Cấu trúc tổng quát bên ngoài của dummy processor 61

Hình 4. 1 Mô tả kết nối giữa board DE2 và máy tính 65
Hình 4.2: Tình trạng mạng khi switch ở trạng thái 0000 67
Hình 4.3: Tình trạng mạng khi switch ở trạng thái 0100 69
Hình 4.4: Tình trạng mạng khi switch ở trạng thái 0011 69
Hình 4.5: Tình trạng mạng khi switch ở trạng thái 0010 70

DANH SÁCH CÁC BẢNG

Bảng 2.1: Tóm tắt các kỹ thuật chuyển mạch [3] 22

Bảng 3.1: Ý nghĩa của các chân tín hiệu của slave RNI 55
Bảng 3.2: Ý nghĩa của các chân tín hiệu của master RNI 58
Bảng 3.3: Ý nghĩa của các chân tín hiệu của switch 60
Bảng 3.4: Ý nghĩa của các chân tín hiệu của dummy memory 61
Bảng 3.5: Ý nghĩa của các chân tín hiệu của dummy processor 62

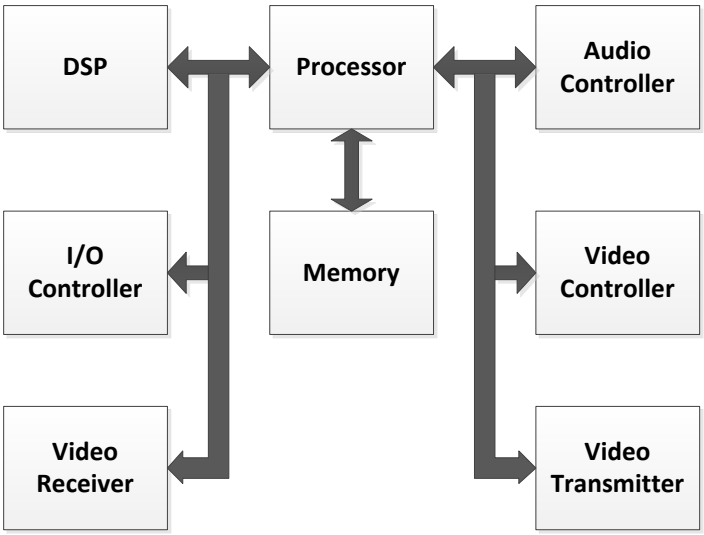
Bảng 4. 1 Phân bố màu chỉ thị đối với kênh dẫn 64
Bảng 4.2 Mô tả hành vi một số resource trên mạng 66
Bảng 4. 3: Bảng tổng hợp thiết kế trên nền tảng phần cứng FPGA 70
Bảng 4.4: Bảng thống kê tài nguyên chi tiết đối với các khối trong thiết kế 71
Bảng 4. 5 Đơn vị tính toán về mặt công suất 73
Bảng 4. 6 Công suất tiêu thụ của thiết kế đối với từng phương pháp tối ưu 73
Bảng 4. 7 Thông tin tần số hoạt động tối đa khác nhau của các họ FPGA 74
Bảng 4. 8 Thông tin về diện tích thiết kế với từng phương pháp tổng hợp 75
Bảng 4. 9 Thông tin về công suất, diện tích và thời gian với phương pháp tối ưu mức cao trong quá trình tổng hợp 77
Bảng 4. 10 Thông tin về công suất, diện tích và thời gian với phương pháp tối ưu mức trung bình trong quá trình tổng hợp 78

CHƯƠNG 1
TỔNG QUAN

1.1 Mô hình SoC đối với các thiết kế đa lõi

Trong lĩnh vực vi mạch điện tử, vấn đề quan trọng nhất là làm thế nào để giảm giá thành, tăng hiệu năng của hệ thống và tăng số lượng sản phẩm[1]. Mật độ tích hợp của hệ thống là yếu tố ảnh hưởng nhiều đến giá thành sản phẩm. Mật độ tích hợp càng cao thì sự phức tạp của các thành phần bên trong hệ thống cũng tăng theo. Để phù hợp với việc gia tăng nhanh về mật độ tích hợp (density), tần số hoạt động (performance) và yêu cầu về thời gian đưa sản phẩm ra thị trường (time to market) thì đã có ngày càng nhiều các thiết kế kiến trúc SoC (System on Chip) đa lõi xử lý (multiprocessor SoC) được ra đời[2].

Việc tích hợp một hệ thống hoàn chỉnh trên một vi mạch duy nhất được biết đến như là hệ thống trên chip. Trong thiết kế SoC, các bộ xử lý với công suất tối ưu được kết hợp với các thành phần phần cứng cần thiết khác là một sự lựa chọn tốt đối với hầu hết các thiết kế nhằm phù hợp giữa hiệu năng và công suất tiêu hao.



Hình 1.1: Các thành phần trong một SoC

Thiết kế SoC phân tách riêng biệt các tác vụ ứng dụng thành nhiều thành phần xử lý khác nhau trong đó mỗi thành phần có thể hoạt động hoặc không hoạt động

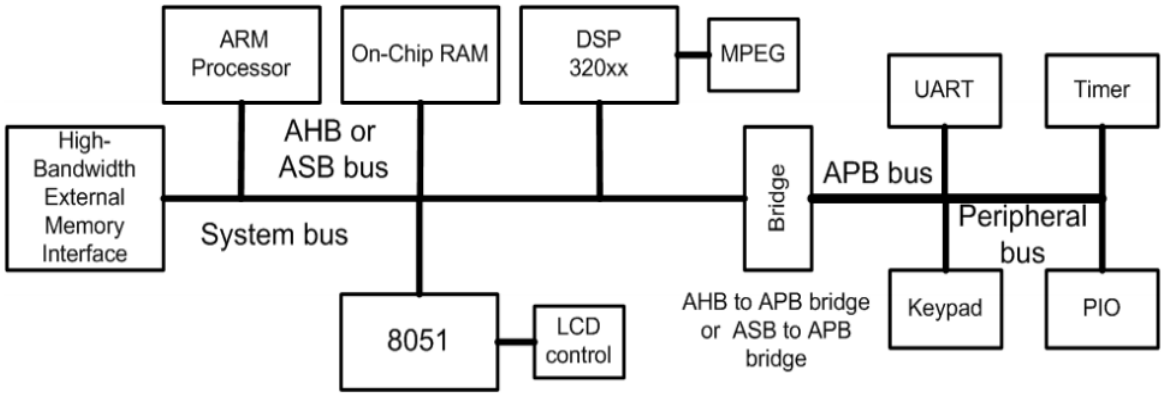
một cách độc lập nhau nhằm giảm sự tiêu hao năng lượng. Mỗi thành phần xử lý có thể hoạt động tại một mức điện áp và tần số khác nhau. Một SoC với thành phần DSP chuyên để xử lý các dữ liệu tín hiệu số, thành phần I/O controller nhằm thu thập và điều khiển các tín hiệu ngoại vi, thành phần memory và memory controller chuyên để lưu trữ dữ liệu và tất cả việc xử lý tính toán chính trên hệ thống sẽ được thực hiện trên bộ xử lý trung tâm của hệ thống.

1.2 Xu thế phát triển và các vấn đề tồn tại về kết nối trên SoC

Trong vài năm gần đây, các nhà thiết kế SoC đã có thể tích hợp với số lượng nhiều hơn 50 lõi [2] với khoảng hơn 4 tỉ transistor trên cùng một chip bao gồm các bộ xử lý, đi kèm là các khối bộ nhớ, khối xử lý tín hiệu...Toàn bộ hệ thống có thể hoạt động với tần số khoảng 10Ghz. Tuy nhiên, ngày nay với sự phát triển nhanh chóng số lượng các thành phần bên trong chip thì các nhà thiết kế SoC phải đối mặt với một thách thức mới trong việc thiết kế nền tảng kết nối các thành phần này[1-4].

Ý tưởng sử dụng các dây dẫn trực tiếp để kết nối các thành phần đã dẫn đến số lượng lớn các chân kết nối cho mỗi thành phần, làm tăng thời gian định tuyến, bên cạnh đó là thời gian trì hoãn (delay) lớn đối với việc lan truyền tín hiệu[1]. Giải quyết vấn đề tồn tại với dây dẫn kết nối trực tiếp thì SoC hiện nay hầu hết đều sử dụng mô hình bus cho việc kết nối các thành phần trên chip. Mô hình bus giúp giảm số lượng chân kết nối và làm chuẩn hóa việc kết nối giữa các thành phần trong chip với bus kết nối vì thế mô hình kết nối này có khả năng tái sử dụng đối với các thiết kế mới. Bus có tính mở rộng nhiều hơn và có thể đạt được hiệu năng gần như tương đương với kiến trúc kết nối trực tiếp.

Với các đặc tính này đã dẫn đến việc các SoC ngày nay chủ yếu sử dụng các mô hình bus cho việc truyền thông và tích hợp các thành phần chức năng trên chip[2, 5]. Các mô hình bus phân cấp đã được đề xuất để mang lại hiệu năng giao tiếp giữa nhiều thành phần giao tiếp. Hiện nay, mô hình bus đã được triển khai trên các ứng dụng SoC như vi điều khiển và các hệ thống đa vi xử lý, điển hình là bus AMBA của hãng thiết kế ARM, Avalon của Altera, CoreConnect của IBM và STBus của ST…[1, 2]



Hình 1.2: Kiến trúc hệ thống dựa trên mô hình bus AMBA[5]

Tuy nhiên, khi xét đối với các hệ thống lớn mô hình bus lại thể hiện một số hạn chế là không có tính mở rộng và không mang lại hiệu năng mong muốn bởi băng thông được chia sẻ giữa tất cả các thành phần kết nối đến bus[1, 2, 5, 6]. Ngoài ra, với hệ thống lớn có số lượng lõi IP lớn sẽ phát sinh vấn đề làm cho đường dây kết nối dài và hình thành các điện trở và điện dung kí sinh có giá trị lớn. Kết quả dẫn đến là gặp phải các vấn đề về trì hoãn liên kết và công suất tiêu thụ.

Với xu hướng thiết kế SoC hiện nay, thách thức chính mà các nhà nghiên cứu phải đối mặt hiện nay để tích hợp một số lượng lớn nhiều thành phần khác nhau trên chip là vấn đề về tính cấu trúc (structured), khả năng mở rộng (scalable), khả năng tái sử dụng (reusable), và hiệu năng cao (high performance).

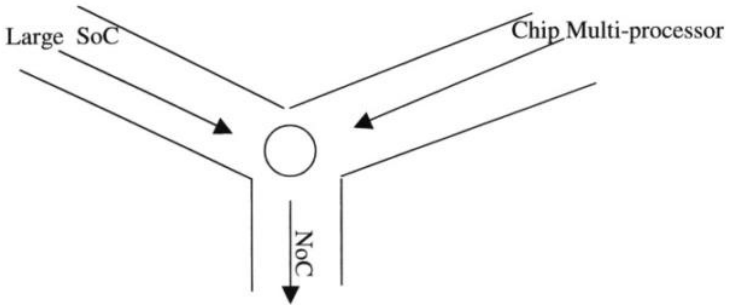
Một nền tảng kết nối được xem là có tính cấu trúc (structured) tức có khả năng phân tích về mặt toán học. Khả năng mở rộng (scalable) của nền tảng kết nối nhằm chỉ ra rằng nền tảng đó có khả năng mở rộng hệ thống với số lượng lớn các thành phần. Một hệ thống không có tính mở rộng tức tính tái sử dụng không cao trong việc kết nối các nguồn tài nguyên. Đặc tính tái sử dụng (reusability) là yếu tố chính trong tất cả các khía cạnh thiết kế SoC lớn và phức tạp. Tính tái sử dụng ảnh hưởng đến thời gian thiết kế và kiểm thử của sản phẩm. Nhờ yếu tố này mà việc xây dựng các SoC mới sẽ dễ dàng hơn bằng cách thêm các thành phần vào thiết kế đã có sẵn.

Trong nhiều ứng dụng thương mại quan trọng, đặc biệt là trong lĩnh vực truyền thông đa phương tiện và xử lý thì yêu cầu nhiều lõi SoC với băng thông lớn giữa các lõi. Do đó, hệ thống có hiệu năng cao tức có băng thông truyền dữ liệu

giữa các lõi là rất lớn[2]. Các đặc tính này đều liên quan với nhau và có sự đánh đổi (trade-off) giữa các tính chất trên trong quyết định thiết kế một SoC.

1.3 Giải pháp và sự cần thiết của nền tảng NoC cho các thiết kế SoC

Trong vài năm gần đây, các công nghệ tích hợp mới ngày càng ra đời đi kèm là yêu cầu kích thước của transistor ngày càng nhỏ hơn và thời gian trì hoãn lan truyền của dây dẫn ngày càng giảm. Vì thế, thiết kế SoC cần một kiến trúc giao tiếp có tính cấu trúc và tính mở rộng để phù hợp với các ứng dụng đòi hỏi sự phức tạp lớn. Việc thiết kế kiến trúc giao tiếp trên chip ngày càng trở nên quan trọng và dần đã có sự chuyển dời từ phương pháp thiết kế tập trung vào khả năng tính toán của hệ thống (computation centric design) sang phương pháp thiết kế tập trung vào vấn đề truyền thông giao tiếp trên hệ thống (communication centric design)[1].



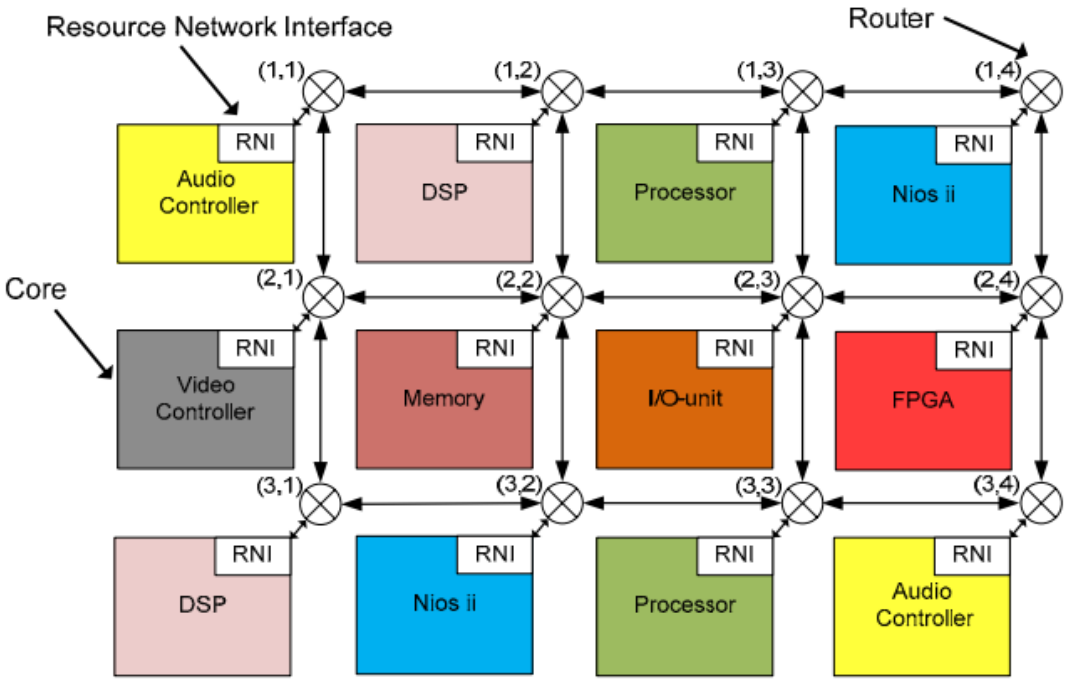
Hình 1.3: Phương pháp thiết kế NoC đối với MPSoC có tích hợp lớn[2]

Các nhà nghiên cứu trong lĩnh vực kiến trúc máy tính đã nghiên cứu nhiều kiến trúc có tính mở rộng để xây dựng các hệ thống đa bộ xử lý. Tính mở rộng và sự thành công của mô hình internet đã mang lại ý tưởng cho các mạng sử dụng thiết bị định tuyến (router). Dựa vào mô hình truyền thông giao tiếp dữ liệu bằng gói tin trên mạng máy tính để áp dụng cho lĩnh vực thiết kế SoC và các hệ thống giao tiếp thông tin trên chip.

Việc truyền thông dựa trên phương pháp chuyển mạch gói mang lại khả năng mở rộng, chuẩn hóa và sử dụng lại đối với cơ sở hạ tầng giao tiếp. Các đặc tính này rất quan trọng với các nhà thiết kế chip với mục đích đưa sản phẩm ra thị trường một cách nhanh chóng. Các nhà nghiên cứu đã đề xuất phương pháp thiết kế tập trung vào vấn đề truyền thông và mô hình mạng trên chip (Network on Chip-NoC)

được đề xuất với nhiều ưu điểm và giải quyết triệt để các vấn đề tồn tại đối với truyền thông trên các SoC hiện nay.

Các khái niệm tồn tại trong nền tảng NoC tương tự với mạng máy tính. Trong đó, thành phần router sẽ nhận chức năng liên kết các thành phần còn lại của hệ thống như bộ xử lý, bộ nhớ và ngoại vi trong một hệ thống ứng dụng NoC. Các giao thức truyền thông dựa trên mô hình phân lớp được sử dụng trong thiết kế NoC. Vì thế, phương pháp thiết kế này đem lại tính mở rộng cao (high scalability) và linh hoạt trong thiết kế các SoC.



Hình 1.4: Mô hình SoC dựa trên nền tảng truyền thông NoC[7]

Trên thế giới, đã có nhiều công trình nghiên cứu và ứng dụng về lĩnh vực NoC nhằm từng bước hiện thực hóa phương pháp thiết kế này[7-16]. Hiện nay, việc nghiên cứu về lĩnh vực mạng trên chip đang được thực hiện bởi nhiều nhóm nghiên cứu của các công ty, trường đại học, viện nghiên cứu trên toàn thế giới theo nhiều hướng khác nhau. Thuật ngữ NoC được sử dụng trong các nghiên cứu đã được công bố với một ý nghĩa rộng, từ các nghiên cứu thực hiện ở mức cổng (gate-level), mức layout, triển khai ứng dụng hệ thống dựa trên NoC, đến việc phát triển thiết kế kiến trúc NoC sao cho đạt hiệu năng cao nhất. Ngoài ra, việc phát triển các công cụ thiết

kế NoC nhằm từng bước thay thế hoàn toàn các phương pháp thiết kế được sử dụng hiện nay đối với SoC cũng được chú ý nghiên cứu.

Có thể bao quát một số công trình nghiên cứu cũng như đề xuất đã được thực hiện trong lĩnh vực NoC như sau:

- **Ethereal** được phát triển bởi Philips từ năm 2001. Mục tiêu là thực hiện một nền tảng NoC đảm bảo thông lượng truyền dẫn. Đối với Ethereal việc đảm bảo dịch vụ là yêu cầu đối với thiết kế phần cứng cũng như cơ sở cho thiết kế phần mềm. Năm 2003, router của Ethereal với 6 cổng 2 chiều độ rộng 32 bit đã được tổng hợp với công nghệ CMOS 0.13um, với diện tích tổng cộng là 0.175mm^2 và băng thông đạt là 16Gbit/giây trên mỗi cổng.
- **MiniNoC** là đề tài được thực hiện tại Đại học kỹ thuật Eindhoven Hà Lan nhằm mục đích nghiên cứu các ứng dụng cho hệ thống MPSoC. MiniNoC gồm 4 bộ xử lý được kết nối với nhau theo topo dạng torus, với cơ chế định tuyến là E-Cube. Việc phát triển các ứng dụng trên hệ thống này sử dụng ngôn ngữ lập trình C/C++. MiniNoC có khả năng mở rộng thành mạng có kích thước lớn.
- **Benini và De Micheli** trong nghiên cứu về NoC đã dự đoán rằng các mạng chuyển mạch gói sẽ là yếu tố quan trọng cần thiết để phù hợp với sự phức tạp của các thiết kế SoC trong tương lai. Họ phân các lớp giao thức đối với mạng trên chip thành 3 phần. Trong đó phần thấp nhất sẽ liên quan đến các vấn đề về dây dẫn vật lý. Phần thứ 2 sẽ phù hợp với thiết kế ở mức kiến trúc và luận lý ở mức độ liên kết dữ liệu và giao vận. Phần cao nhất sẽ xử lý các về vấn đề ứng dụng. Các nhà nghiên cứu đã kết luận rằng ảnh hưởng chính yếu trong thiết kế NoC liên quan đặc biệt đến các giao thức nhằm phù hợp với ứng dụng và hiệu năng.
- **Học viện kỹ thuật hoàng gia, VTT electronic, Oulu** đã hợp tác phát triển một nền tảng NoC cho các hệ thống SoC lớn. Nền tảng này bao gồm các bộ chuyển mạch dạng lưới 2 chiều. Các tài nguyên được đặt tại các giao diện kết nối với các bộ chuyển mạch. Mỗi bộ chuyển mạch được kết nối đến một tài

nguyên và 4 bộ chuyển mạch lân cận. Một tài nguyên có thể là bất cứ thành phần xử lý, bộ nhớ hoặc khối ngoại vi mà phù hợp với giao diện kết nối trên NoC. Để giảm thời gian trì hoãn gói tin trong mạng và giảm chi phí phần cứng của bộ chuyển mạch, các nhà nghiên cứu đã thiết kế bộ đệm dữ liệu có kích thước nhỏ song được kết hợp hiệu quả với thuật toán định tuyến động.

Tại Việt Nam, trong những năm gần đây đã có một số đề tài nghiên cứu về NoC[17, 18]. Đó là các đề tài nhằm triển khai ở cấp độ hệ thống cũng như cải tiến về mặt cấu trúc để tăng hiệu năng của một mô hình NoC. Trong đó:

- Năm 2007, với **BASIC-NoC**, **Huỳnh Việt Thắng** đã giới thiệu mô hình NoC với topo dạng torus kích thước 2x2, sử dụng giải thuật định tuyến theo chiều, cơ chế điều khiển luồng wormhole và kết hợp sử dụng hai kênh ảo. Kết quả nghiên cứu thực thi trên phần cứng FPGA. Kết quả mô phỏng đã thể hiện được ưu điểm của giải pháp sử dụng kênh ảo trong kiến trúc router nhằm tăng hiệu năng và giảm độ trễ trên mạng.
- Năm 2012, **Phạm Đăng Lâm** đã giới thiệu kiến trúc **pipeline NoC**. Với việc xây dựng một NoC với topo dạng lưới 2 chiều và kích thước 3x3, sử dụng giải thuật định tuyến cố định theo chiều, với 2 kênh ảo trên mỗi cổng và cơ chế điều khiển luồng req/ans tác giả đã trình bày mô hình bộ chuyển mạch gói tin trên mạng. Với mô hình này, băng thông đạt được là 6 megapacket/s và băng thông trung bình đạt là 240Mbps.

Tuy kết quả ở mức độ nghiên cứu và triển khai mô hình giả lập dẫn đến hiệu năng và ứng dụng vào các hệ thống thực tế chưa cao. Song đây được xem là những bước đi đầu tiên trong việc nghiên cứu và làm nền tảng để tiếp cận giải pháp NoC đối với thiết kế các SoC nói riêng cũng như lĩnh vực thiết kế vi mạch nói chung.

1.4 Mục tiêu của đề tài

Đề tài tập trung nghiên cứu để giải quyết 3 mục tiêu chính là:

Thứ nhất, nghiên cứu và phát triển mô hình SoC hoàn chỉnh nhằm minh hoạ ứng dụng nền tảng kết nối theo phương pháp truyền gói tin giữa các thành phần, bao

gồm: thiết kế bộ chuyển mạch gói tin là các router, nền tảng liên kết giữa các router để tạo thành một NoC, thiết kế thành phần kết nối giữa các tài nguyên trên SoC là network adapter, giả lập một số thành phần trên hệ thống SoC như bộ xử lý (processor), bộ nhớ (memory), ngoại vi vào/ra …

Thứ hai, mô hình minh hoạ ứng dụng NoC hoàn chỉnh sẽ được thực nghiệm và đánh giá trên nền tảng phần cứng FPGA của hãng Altera là board DE2. Từ việc thực nghiệm bằng phần cứng sẽ đánh giá chi phí của thiết kế cũng như minh họa trực quan quá trình giao tiếp và tình trạng kết nối giữa các thành phần trên mạng thông qua giao diện được xây dựng bởi công cụ Matlab.

Thứ ba, việc đánh giá các vấn đề liên quan đến công suất, diện tích cũng như tần số hoạt động của thiết kế sẽ được thực hiện dựa trên công cụ biên dịch và tổng hợp Design Compiler của hãng Synopsys.

1.5 Phương pháp nghiên cứu

Mô hình minh hoạ NoC sẽ được nghiên cứu và xây dựng dựa trên các vấn đề:

- Nghiên cứu và ứng dụng topo phù hợp cho thiết kế từ đó triển khai phương pháp định tuyến dữ liệu cũng như xác định kích thước bus dữ liệu cho kết nối trên mạng.
- Nghiên cứu các cơ chế chuyển mạch cho mạng bao gồm chuyển mạch mạch (circuit switching) và chuyển mạch gói (packet switching) từ đó chọn ra phương pháp chuyển mạch cho thiết kế.
- Nghiên cứu và ứng dụng cơ chế điều khiển luồng bao gồm điều khiển luồng dựa trên gói tin (store and forward, cut-through), và điều khiển luồng dựa trên flit (wormhole).
- Nghiên cứu và xây dựng một router hoàn chỉnh bao gồm: bộ chuyển mạch trên mạng (switch network), bộ đệm dữ liệu trên mạng (buffer), bộ phân xử dựa vào độ ưu tiên (arbiter).
- Nghiên cứu nền tảng phần cứng FPGA, công cụ phát triển thiết kế hệ thống số Quartus, công cụ mô phỏng Matlab và ngôn ngữ mô tả phần cứng VHDL

nhằm thực thi, minh họa một cách trực quan các trường hợp giao tiếp của thiết kế trên giao diện máy tính.

- Nghiên cứu công cụ biên dịch và đánh giá thiết kế Design Compiler nhằm phân tích các bản báo cáo tổng hợp về mặt diện tích, công suất và tần số theo công nghệ tích hợp chuẩn của hãng Synopsys.

1.6 Bố cục trình bày

Chương 2 mô tả về nền tảng cơ sở hạ tầng truyền thông trên SoC, một số khái niệm cơ bản và các vấn đề giao tiếp mạng gồm: topo mạng, kỹ thuật chuyển mạch, kỹ thuật định tuyến, kỹ thuật điều khiển luồng. Phần này cũng mô tả các thành phần tồn tại trong một thiết kế ứng dụng NoC cũng như chức năng chung của chúng. Ngoài ra, công cụ phát triển cho thiết kế FPGA và công cụ đánh giá thiết kế Design Compiler cũng được trình bày trong chương này.

Chương 3 mô tả phương pháp thiết kế được lựa chọn đối với thiết kế các thành phần trong NoC như: kích thước gói tin, kích thước bộ đệm dữ liệu, tín hiệu điều khiển luồng, thành phần phân xử trong trường hợp xảy ra tranh chấp, thành phần định tuyến. Mô hình thiết kế chi tiết đối với thành phần giao tiếp mạng, tài nguyên mạng như dummy processor, dummy memory, i/o cũng được trình bày trong chương này.

Chương 4 trình bày về việc đánh giá thiết kế về chi phí sử dụng tài nguyên FPGA của các khối thành phần dựa trên công cụ thiết kế Quartus của hãng Altera. Phương pháp mô phỏng tình trạng hoạt động của mạng với sự hỗ trợ của công cụ Matlab dựa trên việc giao tiếp nhận dữ liệu từ mô hình trên FPGA. Ngoài ra, trong phần này việc đánh giá kết quả tổng hợp về diện tích, công suất và tần số của thiết kế cũng được trình bày.

Chương 5 đưa ra những nhận xét về kết quả và những hạn chế trong thiết kế mô hình ứng dụng NoC. Những phương án thiết kế nhằm cải tiến hiệu năng cho mô hình ứng dụng NoC cũng được đề xuất trong phần này.

CHƯƠNG 2
CƠ SỞ LÝ THUYẾT

Chương này trình bày về các khái niệm trong thiết kế NoC, nền tảng cơ sở hạ tầng truyền thông trên SoC cũng như một số khái niệm cơ bản và các vấn đề giao tiếp mạng gồm: topo mạng, kỹ thuật chuyển mạch, kỹ thuật định tuyến, kỹ thuật điều khiển luồng. Chương này đặc tả chi tiết về cấu trúc và chức năng các khối thành phần trong một thiết kế ứng dụng NoC. Ngoài ra, các công cụ phát triển dành cho thiết kế số dựa trên nền tảng phần cứng FPGA và công cụ đánh giá tổng hợp thiết kế cũng được trình bày trong chương này.

2.1 Mạng trên chip

Mạng kết nối đa dụng trên chip, NoC, là thiết kế phù hợp với sự phát triển gia tăng của các thiết kế hệ thống đa lõi xử lý (MPSoC) hiện đại. NoC được thiết kế theo phương pháp tập trung vào hướng truyền thông trong đó nền tảng kết nối đa dụng sẽ được phát triển trước, sau đó các khối chức năng xác định như bộ xử lý, bộ nhớ, bộ điều khiển thiết bị ngoại vi sẽ được gắn kết vào NoC thông qua các giao diện giao tiếp RNI[1].

Nghiên cứu trong lĩnh vực NoC yêu cầu sự hiểu biết về nhiều lĩnh vực khoa học khác nhau từ cấp độ liên kết vật lý đến cấp độ phần ứng dụng. Do đó, các nghiên cứu NoC cần được giải quyết ở các cấp độ khác nhau và có thể được phân loại thành các lớp vật lý, lớp mạng, lớp ứng dụng dựa trên các cấp độ trừu tượng khác nhau. Mỗi phân vùng tồn tại các vấn đề và giải pháp tương ứng. Có thể chia các hướng nghiên cứu về lĩnh vực NoC như sau[18]:

- Khía cạnh cấp độ hệ thống (system level)
  - Phương pháp thiết kế: mô hình hoá và khảo sát mô hình
  - Đặc tính về mặt lưu lượng: độ trễ lan truyền, luồng dữ liệu và những yếu tố ảnh hưởng.
- Khía cạnh về thiết kế giao diện giao tiếp mạng (network adapter hay RNI)
  - Chức năng: phương pháp đóng gói, quản lý dịch vụ

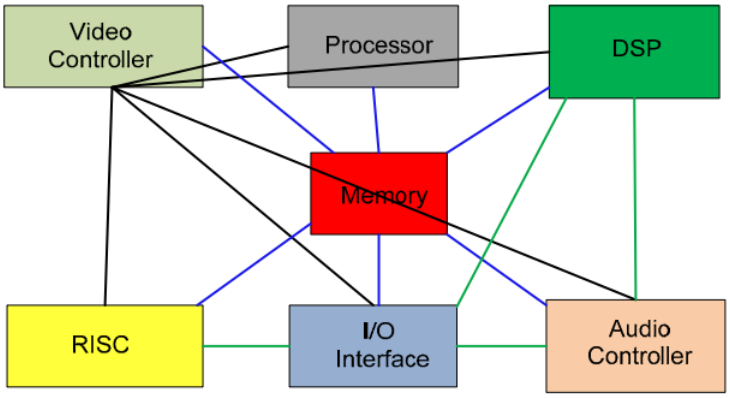
- Giao diện kết nối: dạng kết nối và hoạt động (plug and play) và khả năng linh hoạt kết nối với lõi chức năng IP khác nhau
- Khía cạnh về đặc điểm cấu trúc mạng
  - Topo mạng: các topo phổ biến và biến thể kết hợp
  - Giao thức: định tuyến, chuyển mạch và sơ đồ điều khiển
- Điều khiển luồng: giải quyết tình trạng deadclock, cơ chế kênh ảo, bộ đệm…
- Khía cạnh cấp độ kết nối
  - Đồng bộ dữ liệu, mã hoã dữ liệu, độ tin cậy của hệ thống

2.2 Sự phát triển của nền tảng SoC

2.2.1 Mô hình kết nối trực tiếp

Trong trường hợp này, các thành phần chức năng được kết nối một cách trực tiếp với nhau thông các các đường dây dẫn riêng biệt. Mặt hạn chế của phương pháp kết nối trực tiếp đó là khi số lượng thành phần trong SoC tăng lên thì số lượng dây yêu cầu, số lượng chân kết nối cũng như diện tích sử dụng cho dây dẫn tăng[7].

Hệ thống được phát triển dựa trên nền tảng dây dẫn kết nối trực tiếp sẽ thiếu tính mở rộng. Điều này làm cho hệ thống không đáp ứng được tính tái sử dụng. Ngoài ra việc ảnh hưởng về mặt điện từ của nhiễu đến dây cũng cao. Điều này có thể ảnh hưởng đến chất lượng tín hiệu của truyền thông trên chip.



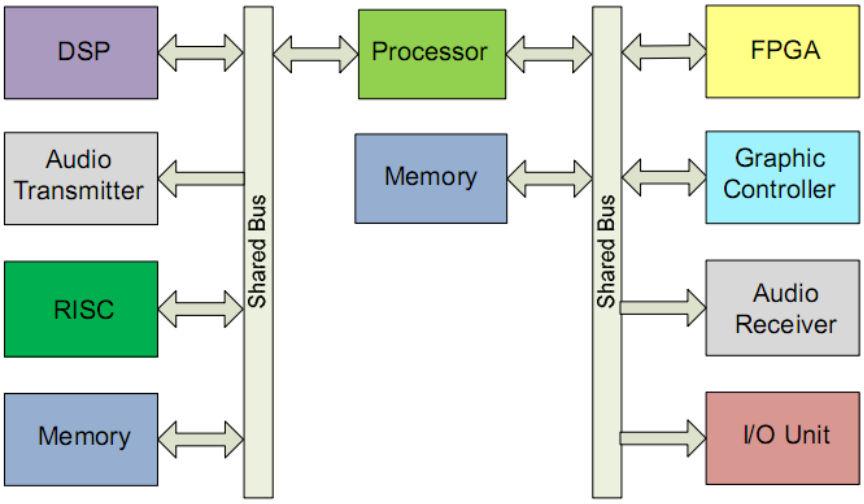
Hình 2.1: Truyền thông trên SoC dựa trên mô hình kết nối trực tiếp [7]

2.2.2 Mô hình bus

Hiện nay, hầu hết việc truyền thông tin giữa các thành phần trên các SoC đều sử dụng mô hình bus [1, 2, 5, 7, 12]. Một mô hình bus sẽ chia sẻ môi trường truyền

dẫn giữa tất cả các khối chức năng. Mỗi thành phần trong hệ thống được kết nối với bus thông qua một giao diện được thiết kế một cách chuẩn hoá. Bộ phân xử là thành phần điều khiển chính của mô hình bus. Khi một thành phần trên mạng muốn gửi dữ liệu đến một thành phần khác thông qua bus thì bộ phân xử sẽ thêm một số tín hiệu điều khiển và gửi nó đến thành phần yêu cầu.

Số lượng chân kết nối đối với các thành phần và dây dẫn được giảm bớt trong mô hình bus. Đối với hệ thống nhỏ thì mô hình bus thể hiện được hiệu quả song với các hệ thống lớn thì khả năng mở rộng và hiệu năng của hệ thống bị ảnh hưởng. Điều này bởi vì nhiều thành phần được kết nối cùng chung bus và chia sẻ cùng băng thông giao tiếp.

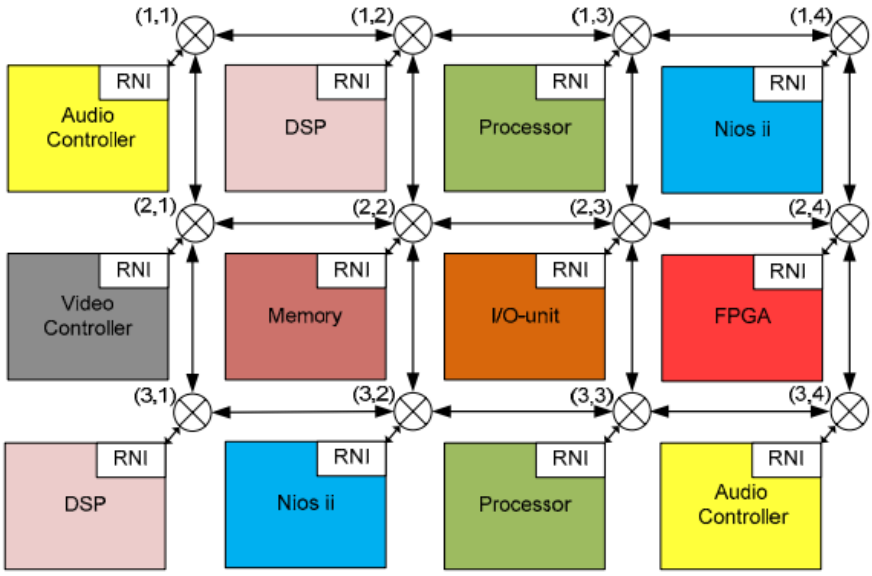


Hình 2.2: Truyền thông trên SoC dựa trên mô hình bus[7]

2.2.3 Mô hình mạng trên chip

NoC là một phương pháp thiết kế thay thế phương pháp kết nối giữa các khối thành phần nhằm giải quyết các vấn đề trong việc truyền thông trên SoC. Trong thiết kế NoC, một phương pháp thiết kế theo mô hình mạng được sử dụng. Trong đó, các thành phần trên hệ thống được kết nối đến mỗi thành phần khác thông qua một mạng gồm nhiều bộ chuyển mạch. Mô hình mạng trên chip mang lại hiệu năng, tính mở rộng và khả năng tái sử dụng cao khi so sánh với mô hình bus và mô hình kết nối trực tiếp bằng dây dẫn.

Mô hình NoC phát huy các mặt lợi điểm khi phát triển các hệ thống có số lượng thành phần xử lý lớn, khoảng trên 20 lõi. Xét về các khía cạnh công suất tiêu thụ và tần số hoạt động so với mô hình bus thông dụng được áp dụng như AMBA thì mô hình NoC không đạt được tiêu chí đạt ra. Song xét về sự trì hoãn trên mạng NoC thì đã giảm trung bình khoảng 15.8%[12] so với thời gian trì hoãn trong việc phân xử trong mô hình bus.



Hình 2.3: Truyền thông trên SoC dựa trên mô hình mạng trên chip [7]

2.3 Các vấn đề giao tiếp mạng

2.3.1 Mô hình truyền thông phân lớp

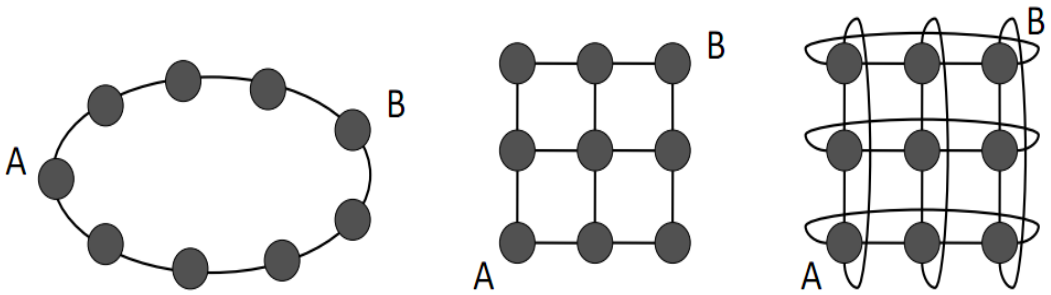
Việc truyền thông trên chip trong mô hình NoC có thể được thể hiện tại 3 lớp mức thấp trong mô hình OSI bao gồm: lớp vật lý, lớp liên kết và lớp mạng. Trong một số nghiên cứu khác cũng đề cập đến một số vấn đề về các lớp mức cao như lớp giao vận, lớp ứng dụng song trong phạm vi của đề tài không trình bày đến các vấn đề này. Chức năng của 3 lớp[1, 2] được mô tả như sau:

- Lớp vật lý sẽ định nghĩa các đặc tính về điện. Ngoài ra, trong lớp này qui định về tín hiệu điều khiển, tín hiệu xung nhịp đối với mỗi kết nối, sơ đồ chân kết nối, kích thước và số lượng dây dẫn

- Lớp liên kết dữ liệu tương ứng với việc truyền thông tin cậy của dữ liệu thông qua liên kết vật lý. Tại lớp này, thiết kế phải được đảm bảo để phù hợp với số lượng bit, kỹ thuật phát hiện lỗi và sửa lỗi.
- Lớp mạng quyết định việc định tuyến đối với các gói tin từ nguồn đến đích thông qua các bộ chuyển mạch trên mạng. Chức năng của lớp này bao gồm việc quyết định định tuyến và thiết kế bộ đệm dữ liệu trên mạng. Ngoài ra, lớp này cũng thực hiện đảm bảo một số vấn đề về chất lượng dịch vụ như thời gian trì hoãn và độ ưu tiên của dữ liệu.

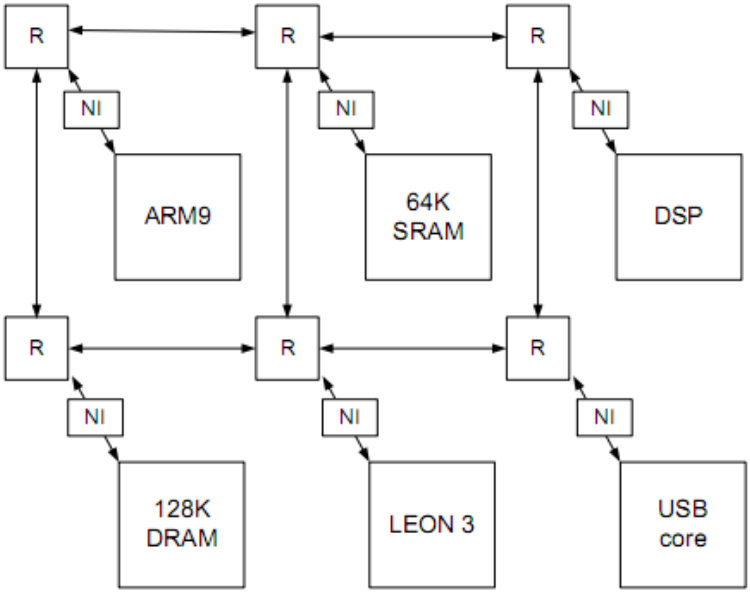
2.3.2 Topo mạng

Topo mạng quyết định thiết kế vật lý và qui định về sơ đồ kết nối giữa các router, các kênh truyền dẫn trên toàn mạng. Topo mạng ảnh hưởng đến hiệu năng trên toàn mạng. Ngoài ra topo còn quyết định số lượng router mà một khung dữ liệu sẽ đi qua cũng như chiều dài liên kết giữa các router. Do đó, topo ảnh hưởng đến độ trễ lan truyền (latency) và công suất năng lượng tiêu thụ của mạng. Chi phí thực hiện của một topo phụ thuộc vào 2 yếu tố: số lượng của các kết nối tại mỗi router và kiến trúc của topo gồm chiều dài của dây dẫn, số lượng lớp. Một số topo mạng thông dụng được áp dụng cho NoC như topo vòng, torus, lưới 2 chiều [19].



Hình 2.4: Topo dạng vòng, lưới 2 chiều và dạng torus

Topo dạng lưới (mesh topo) được xem là lựa chọn tốt cho thiết kế NoC thông qua nhiều nghiên cứu[3, 6, 20]. Dạng lưới 2 chiều có hình dáng là một lưới với chiều rộng và chiều ngang có kích thước xác định. Topo này được sử dụng phổ biến nhất vì sự đơn giản trong chế tạo, khả năng mở rộng, tính đối xứng và độ trễ giữa các router có thể dự đoán trước.



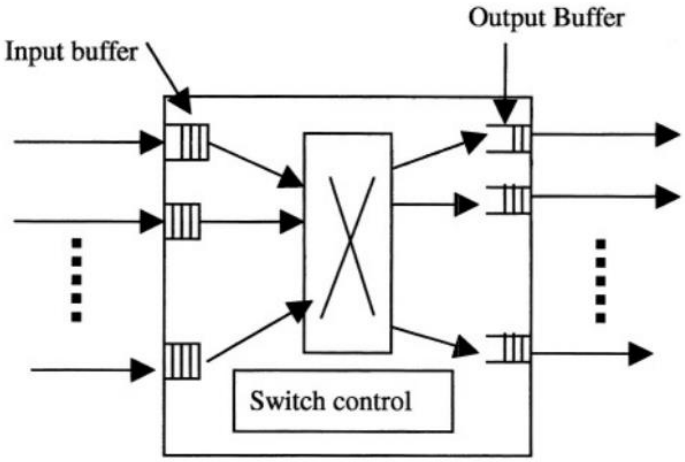
Hình 2.5: Topo dạng lưới với kích thước 2*3 [12]

Địa chỉ router là duy nhất và có thể được tính toán dựa trên vị trí theo chiều dọc và chiều ngang của router trên lưới. Hình trên minh họa cho topo dạng lưới với kích thước 2*3, mỗi router trên mạng liên kết trực tiếp với router khác, tại mỗi router đều kết nối với một lõi chức năng riêng (IP core).

2.3.3 Kỹ thuật chuyển mạch

2.3.3.1 Bộ chuyển mạch (Switch)

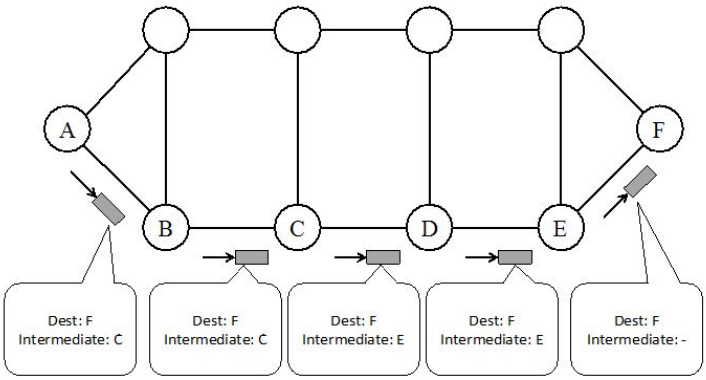
Switch là thành phần chính của kiến trúc mạng. Switch có chức năng chuyển thông tin từ một trong các ngõ vào đến bất cứ một hay nhiều ngõ ra trên switch. Số lượng ngõ vào/ra thường được giới hạn tuỳ thuộc vào topo được áp dụng cho thiết kế. Khoảng thời gian giữa thông tin đi vào ngõ vào đến thời gian khi thông tin rời switch ở ngõ ra gọi là trì hoãn chuyển mạch (switch delay). Dữ liệu cần được lưu trữ trước khi đi ra ở ngõ ra trên switch. Switch có những bộ đệm (buffer) để nhận các gói tin đầu vào và các bộ đệm để lưu trữ các gói tin trước khi chúng được đưa ra ngõ ra. Một crossbar switch hỗ trợ để kết nối bất cứ ngõ ra nào của bộ đệm ngõ vào đến bất cứ bộ đệm ngõ ra. Hiện nay, trên thế giới đã nhiều nghiên cứu tập trung vào việc thiết kế các switch để đạt được độ trễ thấp hơn, thông lượng cao hơn.



Hình 2.6: Thiết kế tổng quát bên trong một switch [2]

Một bộ chuyển mạch mạng (switch network) bao gồm một liên kết của nhiều switch cho phép một số lượng lớn các thành phần như CPU, bộ nhớ,ngoại vi… giao tiếp với nhau. Switch network được đặc trưng bởi topo của nó, đó là một sơ đồ cấp thấp về hình dáng liên kết các switch lại với nhau. Có nhiều loại topo được sử dụng và đề xuất cho việc xây dựng các switch network đối với các hệ thống đa vi xử lý. Một topo được cho là phù hợp với thiết kế SoC nếu nó có một bản thiết kế có tính hiệu quả về diện tích.

Các tham số liên quan đến kết nối đặc trưng và quyết định hiệu năng cho một switch network là network diameter, connectivity, bandwidth và latency[2]. Network diameter là số lượng router trung gian cực đại giữa nguồn phát và đích nhận. Connectivity nói đến số lượng các router lân cận trực tiếp kết nối với bất cứ router trên mạng. Bandwidth là một phép đo tốc độ cực đại tính theo bit/giây của luồng thông tin trên mạng. Trong NoC có thể tính toán bandwidth cho một đường kết nối đơn, cho một router đơn hoặc một bandwidth tổng hợp của toàn kết nối mạng. Latency là thời gian được tính bởi khi một thông điệp di chuyển từ nguồn phát đến đích nhận.



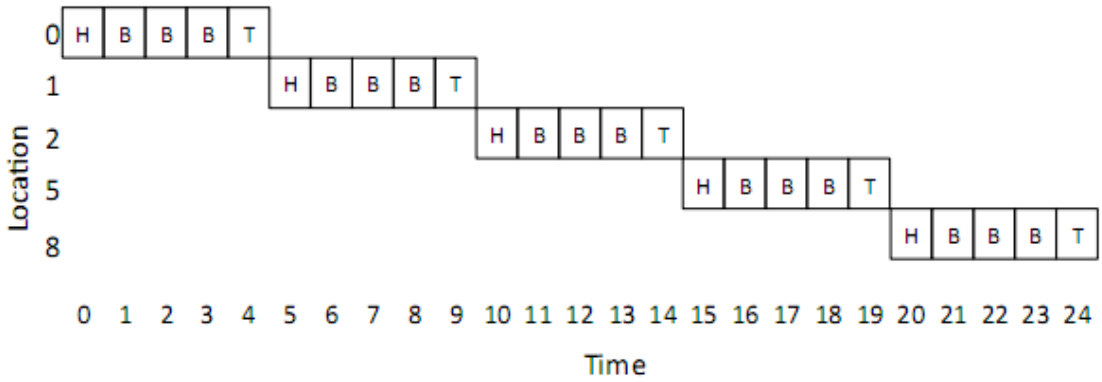
Hình 2.7: Kết nối giữa 2 router với 4 router trung gian [2]

2.3.3.2 Kỹ thuật chuyển mạch

Kỹ thuật chuyển mạch qui định các luồng dữ liệu di chuyển như thế nào qua switch trên mạng. Có 2 loại chuyển mạch cơ bản là: chuyển mạch mạch (circuit switching) và chuyển mạch gói (packet switching). Trong phương pháp chuyển mạch mạch, một mạch cố định được thiết lập giữa bên phát và bên nhận dữ liệu. Kết nối trực tiếp này được duy trì trong suốt thời gian giao tiếp.

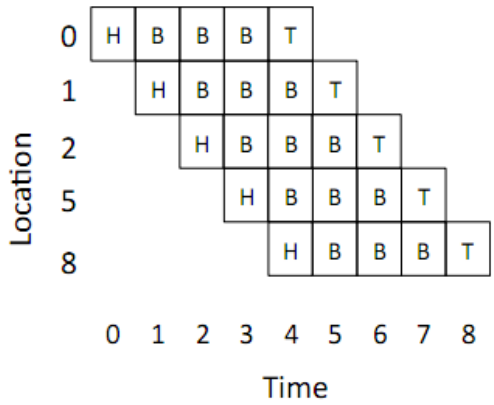
Mặc dù chuyển mạch mạch có thể đảm bảo khả năng băng thông được duy trì một cách cố định với một khối lượng dữ liệu cần truyền tải lớn. Song đối với phương pháp này tồn tại một thời gian trễ lớn để khởi tạo và cấu hình thiết lập kết nối trực tiếp. Kỹ thuật chuyển mạch gói bao gồm một số phương pháp như store and forward, virtual cut-through, và wormhole[1, 2, 6].

Trong phương pháp chuyển mạch gói store-and-forward, toàn bộ gói tin sẽ được lưu trữ trong bộ đệm khi một gói tin di chuyển qua một router trung gian. Sau khi một gói tin đến router và được được chứa trong bộ đệm, gói tin có thể được chuyển tiếp đến router tiếp theo nếu bộ đệm trên router đó cho phép để lưu trữ toàn bộ gói tin. Với kỹ thuật này thì yêu cầu một lượng lớn không gian lưu trữ trên bộ đệm sao cho phải bằng kích thước của gói tin lớn nhất. Như vậy với phương pháp này có thể làm tăng kích thước không gian thiết kế trên chip. Với không gian cần thêm vào trong phương pháp thiết kế này có thể sẽ làm tăng độ trễ lan truyền bởi một gói tin chỉ có di chuyển đến router tiếp theo khi toàn bộ gói tin được lưu trữ.



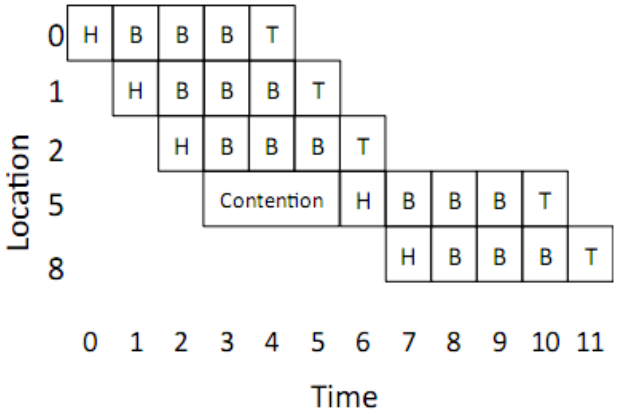
Hình 2.8: Kỹ thuật chuyển mạch store-and-forward [3]

Để giảm độ trì hoãn các gói tại mỗi router, phương pháp virtual cut-through được sử dụng. Phương pháp này cho phép truyền một gói tin đến router tiếp theo trước khi toàn bộ gói tin được nhận tại router hiện hành. Router hiện tại vẫn yêu cầu một khoảng không gian bộ đệm. Hình dưới đây minh họa sơ đồ thời gian đối với phương pháp virtual cut-through khi không có tranh chấp. Có thể thấy được rằng khi sử dụng phương pháp store-and-forward thì cần đến 25 chu kỳ xung để truyền toàn bộ gói tin song với virtual cut-through giảm xuống còn 9 chu kỳ xung.



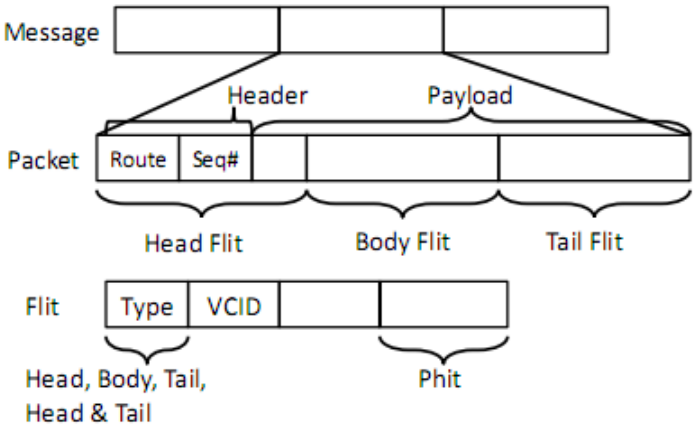
Hình 2.9: Virtual Cut-Through trong trường hợp không có trì hoãn [3]

Trong trường hợp có tranh chấp xảy ra, gói tin sẽ bị trì hoãn. Hình dưới đây minh họa cho trường hợp virtual cut-through có trì hoãn. Việc truyền phát gói tin ở router 2 đến 5 sẽ bị trì hoãn dù router vẫn có bộ đệm cho 2 trong số 5 flit. Như vậy sẽ không có flit nào được xử lý cho đến khi có đủ bộ đệm cho toàn bộ 5 flit.



Hình 2.10: Virtual Cut-Through trong trường hợp có trì hoãn [3]

Vấn đề về việc yêu cầu một khoảng không gian vùng nhớ lớn cho bộ đệm có thể được giải quyết bằng cách sử dụng phương pháp chuyển mạch wormhole. Trong phương pháp chuyển mạch wormhole, các gói tin được chia thành các flit (flow control digit). Các flit này được di chuyển theo hướng truyền theo mô hình pipeline.



Hình 2.11: Message được chia nhỏ thành các flit để di chuyển trên mạng [3]

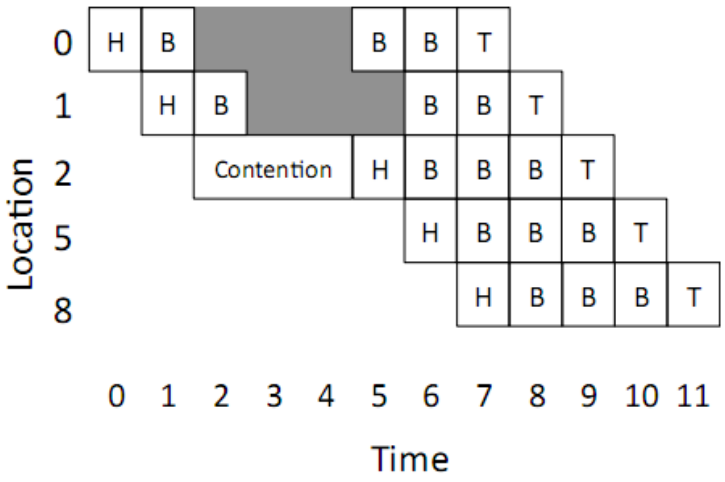
Do đó, phương pháp wormhole không cần các bộ đệm có kích thước lớn để lưu trữ toàn bộ các gói tin nhưng cũng cần có các bộ đệm nhỏ để chứa một vài flit. Flit đầu tiên là head flit sẽ tạo ra một đường định tuyến để cho phép các flit khác theo sau cùng di chuyển trên đường dẫn do nó tạo nên.

Giống như phương pháp virtual cut-through, thì wormhole cho phép các flit di chuyển đến các router tiếp theo trước khi toàn bộ gói tin được nhận tại router hiện hành. Các flit có thể rời khỏi router hiện hành ngay sau khi có đủ không gian vùng

nhớ cho flit này. Tuy nhiên không giống như phương pháp store-and-forward và virtual cut-through, wormhole sẽ phân chia việc lưu trữ và băng thông cho các flit thay vì cho toàn bộ gói tin. Điều này cho phép thiết kế sử dụng các bộ đệm flit trên mỗi router và ngay cả cho các gói tin có kích thước lớn.

Phương pháp wormhole sử dụng bộ đệm rất hiệu quả song lại không hiệu quả khi sử dụng băng thông đường truyền. Mặc dù, phương pháp này phân bố lưu trữ và băng thông dựa trên kích thước flit, một đường truyền sẽ bị chiếm giữ hoàn toàn trong suốt quá trình của một gói tin trên router. Như vậy khi gói tin bị chặn, tất cả các đường truyền vật lý mà gói tin chiếm giữ sẽ bị sẽ không sử dụng được. Vì wormhole phân bố các bộ đệm theo mỗi flit, một gói tin bao gồm nhiều flit có thể di chuyển khắp trên nhiều router và như vậy có nghĩa là sẽ có nhiều đường truyền vật lý bị rơi vào trạng thái không làm việc cho các gói tin khác. Thông lượng sẽ bị ảnh hưởng giảm đi bởi các gói tin khác nằm liền sau gói tin bị khóa và không thể sử dụng các đường truyền vật lý không làm việc.

Ở minh họa trong hình dưới đây, mỗi router có 2 bộ đệm cho flit. Khi head flit tranh chấp trong khi di chuyển từ node 1 sang 2, các body và tail flit của nó cũng bị chặn lại ở router 0 vì không có không gian bộ đệm tại router 1 cho đến khi head di chuyển đến router2. Tuy nhiên, kênh truyền thì vẫn bị giữ bởi gói tin này.

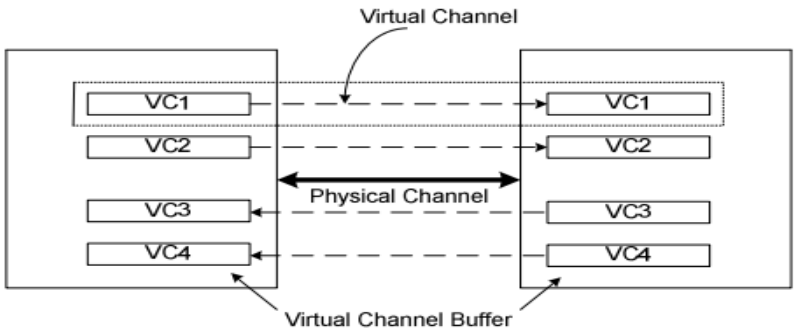


Hình 2.12: Phương pháp điều khiển luồng wormhole [3]

Như vậy, phương pháp Wormhole giảm độ trễ gói tin bằng cách cho phép một flit rời khỏi router ngay khi có không gian bộ đệm của router tiếp theo trên đường truyền. Nếu xét trong trường hợp có tranh chấp và thì độ trễ coi như bằng với phương pháp virtual cut-through. Thêm vào đó, wormhole yêu cầu một bộ đệm nhỏ hơn phương pháp store-and-forward vì thế phương pháp này phù hợp vào các giới hạn đặt ra đối với diện tích và công suất của mô hình NoC.

2.3.4 Kênh ảo

Vấn đề head flit bị chặn có thể dẫn đến các gói tin theo sau cũng bị chặn. Hiện tượng này gọi là dealock tức mạng bị chặn bởi các bộ đệm trong trạng thái đầy và sự phụ thuộc vòng tròn diễn ra giữa các router. Phương pháp kênh ảo (virtual channel) được đưa ra để giải quyết cho vấn đề giải phóng deadlock trên mạng. Phương pháp này về cơ bản sẽ tạo ra các hàng đợi riêng biệt trong một router. Với nhiều kênh ảo khác nhau sẽ chia sẻ sử dụng cùng một kênh truyền vật lý giữa 2 router một cách luân phiên. Bằng cách liên kết đa hàng đợi với mỗi cổng đầu vào, vấn đề chặn head-of-line sẽ được giảm bớt. Các kênh ảo sẽ phân xử băng thông kênh truyền vật lý trên từng chu kỳ (cycle-by-cycle). Trong thiết kế không sử dụng kênh ảo, nếu một gói tin hoặc flit đang chờ để được truyền đi từ một ngõ vào/ra của router sẽ phải chờ nếu cổng của router đó bận. Khi một gói tin trên một kênh ảo bị chặn, các gói tin khác trong tình trạng chờ vẫn có thể di chuyển thông qua kênh truyền ảo còn lại. Do đó kênh ảo tăng khả năng sử dụng của các kênh truyền vật lý và mở rộng thông lượng của toàn mạng.



Hình 2.13: 4 kênh ảo chia sẻ một kênh truyền vật lý [3]

Cơ chế hoạt động như sau: router nguồn gửi một flit di chuyển đến một router đích và sau đó một flit xác nhận (ack flit) được gửi ngược lại về router nguồn. Khi nhận tín hiệu xác nhận, router nguồn sẽ phát toàn bộ gói tin với băng thông lớn nhất cho đường dẫn. Đường dẫn xác định sẽ được giải phóng bởi router đích khi trên router này gặp flit cuối cùng (tail flit). Có thể thiết kế từ 2 đến 8 kênh ảo tồn tại trên 1 cổng của một router.

Hiện nay, chi phí dây dẫn thì hầu như ngang bằng với chi phí các cổng[1, 4, 9, 15]. Như vậy trong tương lai, chi phí về dây dẫn cần phải được xem xét hiệu quả. Do đó, việc sử dụng dây dẫn một cách hiệu quả là một việc rất quan trọng nhằm giảm chi phí của hệ thống. Một kênh ảo cung cấp một đường dẫn được xử dụng một cách luân phiên đối với luồng dữ liệu, do đó kênh ảo sẽ sử dụng dây dẫn hiệu quả hơn đối với việc truyền dữ liệu. Bởi vậy, chúng ta có thể giảm độ rộng dây dẫn trên hệ thống tức có thể chọn 32 bit hay vì sử dụng 64 bit. Do đó, chi phí về dây dẫn và hệ thống sẽ được giảm bớt.

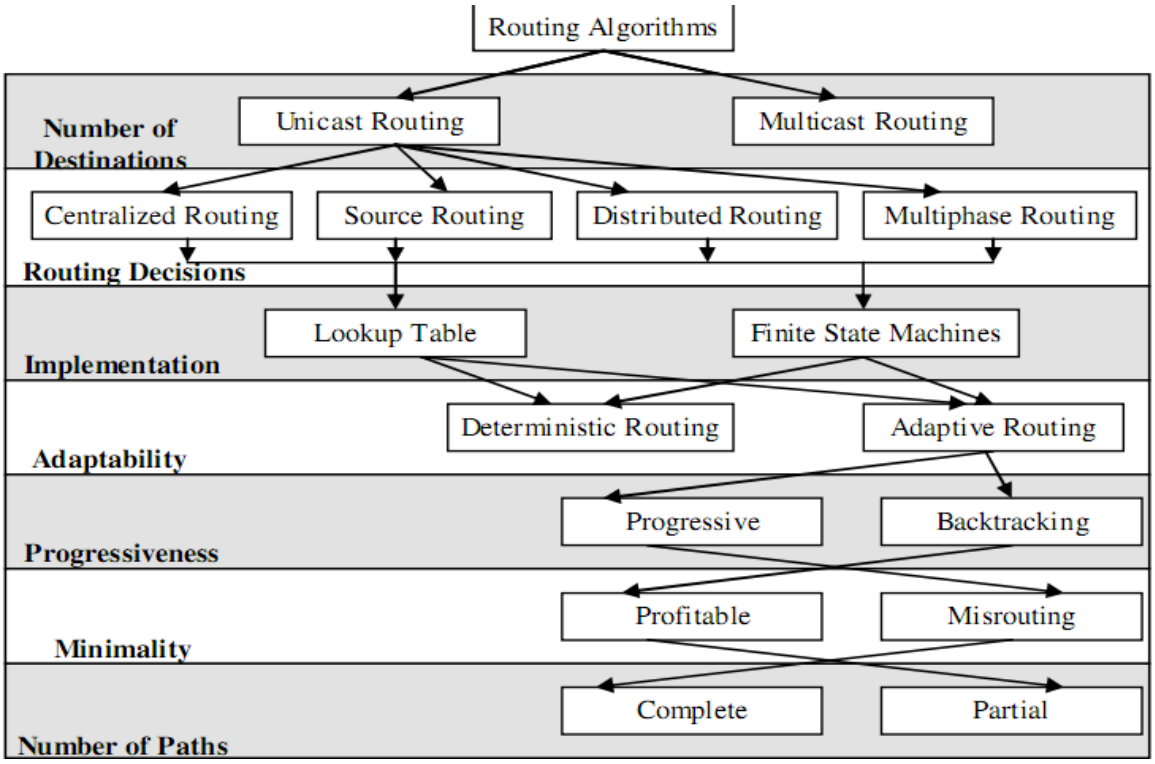
Tóm lại, việc sử dụng các kênh ảo làm giảm độ trễ của mạng, công suất tiêu thụ vào chi phí sản xuất của việc thực thi NoC[1-3]. Ngoài ra, phương pháp này còn yêu cầu ít bộ đệm hơn nên độ trễ sẽ nhỏ hơn. Phương pháp wormhole kết hợp với kênh ảo là phương pháp chuyển mạch phù hợp nhất và là sự lựa chọn của nhiều nghiên cứu về mạng trên chip.

Bảng 2.1: Tóm tắt các kỹ thuật chuyển mạch [3]

Chuyển mạch	Liên kết	Bộ đệm	Đặc tính
Chuyển mạch mạch	Thông điệp	Không	Yêu cầu việc cấu hình và xác thực
Store-and-forward	Gói tin	Gói tin	Head flit cần chờ toàn bộ gói tin trước khi tiếp tục đến đường dẫn kế tiếp.
Virtual Cut-Through	Gói tin	Gói tin	Head flit có thể di chuyển đến đường dẫn kế tiếp trước khi tail flit đến router hiện hành
Wormhole	Gói tin	Flit	Head flit của gói tin bị chặn làm giảm hiệu quả băng thông đường truyền.
Virtual Channel	Flit	Flit	Có thể xen kẽ flit của các gói tin khác nhau trên đường truyền ảo.

2.3.5 Định tuyến

Các thuật toán định tuyến được phân loại thành các dạng khác nhau. Với phương pháp định tuyến dạng unicast, các gói tin có một điểm đích đến duy nhất trong khi trường hợp định tuyến multicast thì các gói tin có nhiều điểm đích đến. Đối với việc giao tiếp trên chip, chiến thuật định tuyến unicast được xem như là phương pháp thực tế bởi việc kết nối đơn điểm giữa các thành phần bên trong chip. Định tuyến unicast được chia ra làm 4 loại là: định tuyến trung tâm, định tuyến nguồn, định tuyến phân tán, định tuyến multiphase.



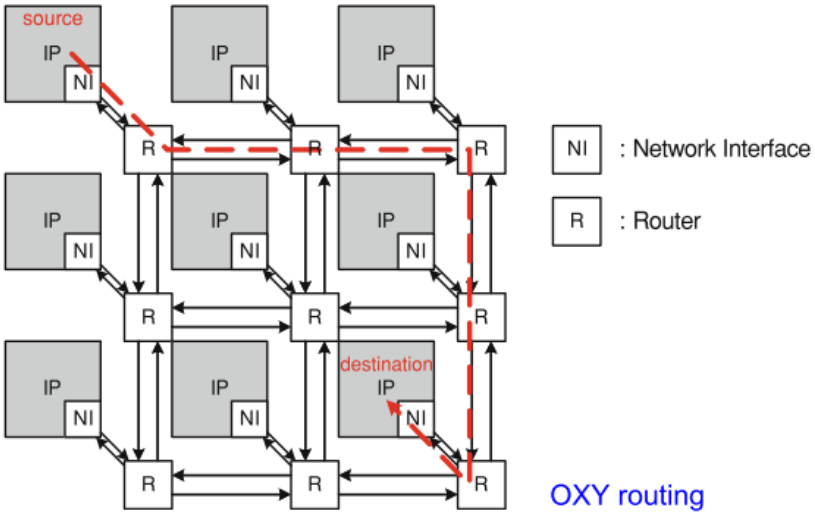
Hình 2.14: Sơ đồ các phương pháp định tuyến [6]

Trong phương pháp định tuyến trung tâm, một bộ điều khiển trung tâm sẽ điều khiển luồng dữ liệu trong hệ thống. Trong định tuyến nguồn, việc quyết định định tuyến sẽ được diễn ra tại nơi tạo dữ liệu trong khi đó định tuyến phân tán thì việc quyết định định tuyến lại được quyết định luồng gói tin/flit thông qua mạng. Sự kết hợp của 2 loại, định tuyến nguồn và phân tán được gọi là định tuyến multiphase.

Các thuật toán định tuyến còn được định nghĩa dựa trên việc thực thi của chúng và được chia ra thành dựa trên bảng tra (lookup table) hoặc máy trạng thái (FSM). Các thuật toán định tuyến dựa trên bảng tra thì phổ biến. Chúng được thực hiện bằng phần mềm trong đó bảng tra được lưu trữ trong mỗi router. Thuật toán định tuyến có thể thay đổi bằng cách thay đổi toàn bộ bảng tra. Các thuật toán định tuyến dựa trên máy trạng thái có thể được thực hiện bằng cả phần mềm hoặc cứng.

Ngoài ra, thuật toán định tuyến còn được phân loại dựa trên sự thích nghi. Định tuyến xác định dựa vào một đường dẫn xác định trên mạng như các phương pháp định tuyến XY. Các thuật toán định tuyến thích nghi cần nhiều thông tin về tình trạng của mạng để tránh các đường dẫn xung đột trên mạng. Các thuật toán định tuyến này thì khá phức tạp để thực hiện do đó nó có chi phí cao về diện tích, công suất.

Hầu hết việc thực thi NoC sử dụng thuật toán, là định tuyến theo kích thước bậc (DOR) vì tính đơn giản và sự phù hợp với topo lưới 2 chiều. Đây là phương pháp định tuyến theo kiểu định tuyến xác định. Trong đó, tất cả các thông điệp từ router nguồn A đến đích B sẽ cùng di chuyển trên cùng một đường truyền. Với phương pháp định tuyến này, một thông điệp di chuyển trên mạng dựa vào vị trí đã được xác định, tại mỗi router gói tin sẽ xác định vị trí hiện hành của nó và quyết định tiếp tục di chuyển đến router nào kế tiếp. Xét trường hợp topo mạng dạng lưới 2 chiều có kích thước 3*3, thuật toán định tuyến theo kích thước bậc X-Y sẽ gửi các gói tin theo chiều X trước sau đó theo chiều Y. Trong mô hình mạng lưới 2 lưới dưới đây, một gói tin di chuyển từ router có vị trí (0,0) đến router vị trí (3,3) sẽ di chuyển đến 2 router theo chiều X sau đó tại router (0,2) sẽ di chuyển qua 2 router theo chiều Y để đến đích.



Hình 2.15: Đường đi của gói tin theo cơ chế định tuyến OXY [1]

Trong định tuyến gói tin, một vấn đề có thể ảnh hưởng lên thông lượng của truyền dẫn là đó là vấn đề deadlock. Hình sau đây mô tả vấn đề deadlock.

Hình 2.16: Gói tin di chuyển mọi hướng và chỉ hướng theo chiều X-Y

Ở trường hợp a thể hiện tất cả các khả năng chuyển hướng có thể trong mạng lưới 2 chiều trong khi trường hợp b minh họa đặc tính chỉ chuyển hướng giới hạn ở phương pháp định tuyến DOR X-Y. Việc cho phép chuyển mọi hướng có thể diễn ra tình trạng deadlock. Trong trường hợp b, không có vòng tròn xoay vòng nào. Một gói tin di chuyển từ đông sang tây chỉ được cho phép xoay vòng sang hướng bắc hoặc nam tuy nhiên nếu di chuyển từ bắc và nam thì không được chuyển hướng.

Phương pháp định tuyến dựa trên kích thước bậc có cả 2 đặc tính là đơn giản và không dealock, tuy nhiên nó không cho phép di chuyển gói tin trên nhiều đường khác nhau trong mạng. Điều này dẫn đến thông lượng của mạng bị giảm. Khi không

có sự đa dạng về đường đi cho các gói tin, thì phương pháp định tuyến này không có khả năng tránh các vùng đụng độ, không có khả năng cân bằng tải trên mạng.

2.4 Các thành phần trên hệ thống ứng dụng NoC

Một hệ thống ứng dụng nền tảng NoC trong kết nối có thể được mô tả bao gồm các thành phần khác nhau, song tập trung gồm 3 thành phần chính là: tài nguyên (resource), giao diện giao tiếp mạng (network adapter) và bộ chuyển mạch mạng (router). Node là đơn vị giao tiếp trên mạng. Tại mỗi node quá trình truyền và nhận dữ liệu trên mạng được diễn ra. Node là sự kết hợp của 3 thành phần resource, RNI, và router với các đường tín hiệu và dữ liệu tương ứng. Trong thiết kế của đề tài chứa các resource bao gồm dummy processor, dummy memory, switch và uart. Vì thế node được phân chia thành các loại sau:

- Loại 0: chứa thành phần master IP, master RNI và router
- Loại 1: chứa thành phần slave IP, slave RNI và router
- Loại 2: chứa thành phần uart, slave RNI và router
- Loại 3: chỉ chứa router
- Loại 4: chỉ chứa thành phần ngõ vào switch

2.4.1 Resource

Một lõi (resource) là một bản thiết kế số hoặc tương tự đã được thiết kế trước. Thông thường resource đảm nhận một chức năng nhất định trong một hệ thống SoC (system on chip). Một số resource được sử dụng phổ biến như NiosII, xử lý tín hiệu số (DSP), điều khiển âm thanh (audio controller), khối điều khiển video (video controller), mã hoá (encryption), điều khiển ngoại vi (I/O controller), bộ nhớ (memory), khối cung cấp nguồn (power), khối xử lý cho ứng dụng xác định (application specific processor)…Các lõi này thường được thiết kế từ một số công ty chuyên trong lĩnh vực thiết kế. Các công ty sản xuất chip sẽ mua lại thiết kế của các lõi này và tiến hành sản xuất ra con chip riêng của công ty họ trong đó có tích hợp các lõi xử lý được mua.

Phụ thuộc vào khả năng linh hoạt trong việc tái cấu hình mà các lõi có thể được phân loại thành các loại như sau:

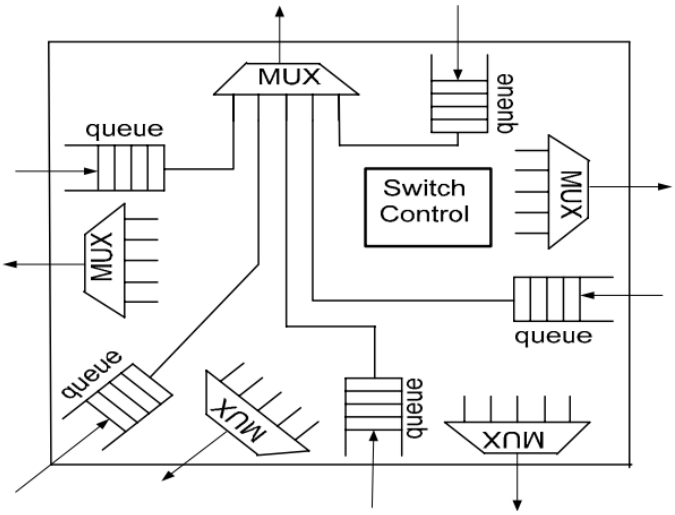
- Hard core chứa thông tin việc sắp xếp về mặt vật lý của thiết kế. Hard core có thể liên kết với soft core và các thành phần ngoại vi trên một chip. Các hard core đã được tối ưu ở mức độ cao nhất về mặt diện tích cũng như hiệu năng. Core ở dạng này có công nghệ được cố định (fixed technology) nên kém linh hoạt trong khả năng thay đổi và tối ưu. ARM, Virtex-II và Lexra là các nhà cung cấp hard core.
- Soft core là một mô tả về mặt chức năng của một IP core. Một soft core thông thường được thể hiện bằng một dạng ngôn ngữ mô tả phần cứng như là VHDL, Verilog... Dạng core này có khả năng tái cấu hình tức có thể thay đổi chức năng và cấu trúc bởi mô tả RTL vì thế khả năng linh hoạt của dạng này cao hơn so với hard core. Nios II của Altera và Microblaze của Xilinx là một số điển hình cho dạng soft core.

2.4.2 Resource Network Interface

Resource được kết nối với router thông qua các RNI (hay còn gọi là network adapter). Chức năng của RNI tương tự như của card mạng trên một hệ thống máy tính cá nhân nhằm để kết nối máy tính với nhau trên mạng máy tính. Thiết kế bên trong của RNI có thể được chia thành 2 phần: thành phần phụ thuộc và không phụ thuộc vào resource.

2.4.3 Router

Ý tưởng thực hiện của router trên nền tảng NoC được lấy từ ý tưởng thiết kế router trong mạng máy tính. Router đóng một vai trò rất quan trọng và được xem như là xương sống của NoC. Router chuyển tiếp gói tin từ một trong số các ngõ vào của nó đến một trong các ngõ ra khác. Một số nghiên cứu đã đề xuất một số kiến trúc khác nhau cho NoC router[4, 9, 10, 13, 15-18, 21]. Một mô hình router của NoC đặc trưng cho topo dạng lưới 2 chiều được thể hiện như hình dưới đây



Hình 2.17: Kiến trúc cơ bản của router [2]

Các router cần được thiết kế để phù hợp với các yêu cầu về độ trễ và thông lượng nhằm đáp ứng với các giới hạn về diện tích cũng như năng lượng tiêu thụ. Điều này là thách thức lớn mà nhà thiết kế hệ thống phải đối mặt. Độ phức tạp của router sẽ tăng khi đòi hỏi lớn về nhu cầu băng thông và độ trễ. Các router đơn giản có thể không sử dụng cơ chế pipeline, không sử dụng điều khiển luồng wormhole, không sử dụng kênh ảo, bộ đệm vào ra giới hạn chỉ phù hợp trong các ứng dụng mà không yêu cầu cao về thông lượng.

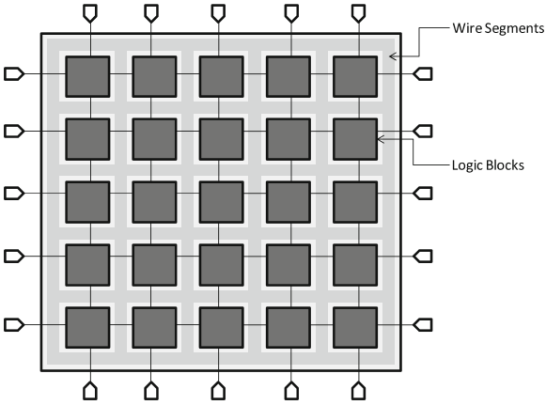
2.5 Công cụ phát triển

Công cụ phát triển hệ thống ứng dụng NoC bao gồm: ngôn ngữ mô tả phần cứng VHDL nhằm mô tả hành vi hoạt động của hệ thống, công cụ phát triển Altera Quartus trên nền tảng hệ điều hành Windows để thiết kế kiến trúc hệ thống, công cụ mô phỏng và tổng hợp cho FPGA, công cụ tổng hợp và đánh giá thiết kế Design Compiler trên nền tảng hệ điều hành Linux nhằm tạo ra các báo cáo về diện tích, công suất, tần số của thiết kế.

2.5.1 Vai trò FPGA trong thiết kế hệ thống số

Trong công nghệ thiết kế và chế tạo IC thì ASIC là một IC thực hiện cho một ứng dụng xác định. ASIC phức tạp chứa các bộ xử lý, các khối bộ nhớ và một số ASIC khác được gọi là các IP. Tất cả thành phần này tạo thành một hệ thống gọi là

SoC. Một phương pháp thiết kế tối ưu chứa nhiều ưu điểm trong những năm gần đây đối với thiết kế và kiểm thử IC là sử dụng FPGA.



Hình 2.18: Thể hiện bên trong của một FPGA [22]

FPGA có thể được sử dụng để thực hiện bất cứ chức năng nào của một ASIC, chỉ khác là nếu một IC được thiết kế như một FPGA thì có thể được lập trình bởi người dùng sau khi IC đó được sản xuất. FPGA bao gồm một dãy các khối logic và các kết nối giữa các khối thành phần này đều có khả năng lập trình.

Trong việc thiết kế và kiểm thử sử dụng phần cứng FPGA, nhà phát triển hệ thống sẽ sử dụng một trong các dạng ngôn ngữ HDL như Verilog, VHDL… để mô tả chức năng của phần cứng. HDL được xác nhận về mặt chức năng thông qua chương trình mô phỏng và sau đó được tổng hợp thành các cổng logic xuống phần cứng FPGA. Phương pháp sử dụng FPGA để thực nghiệm hoạt động hoạt động thực tế của thiết kế sẽ nhanh hơn. Ngày nay, FPGA trở thành một phương pháp phổ biến để thực hiện nguyên mẫu cho ASIC và các SoC. Phương pháp này được gọi là FPGA prototyping[22].

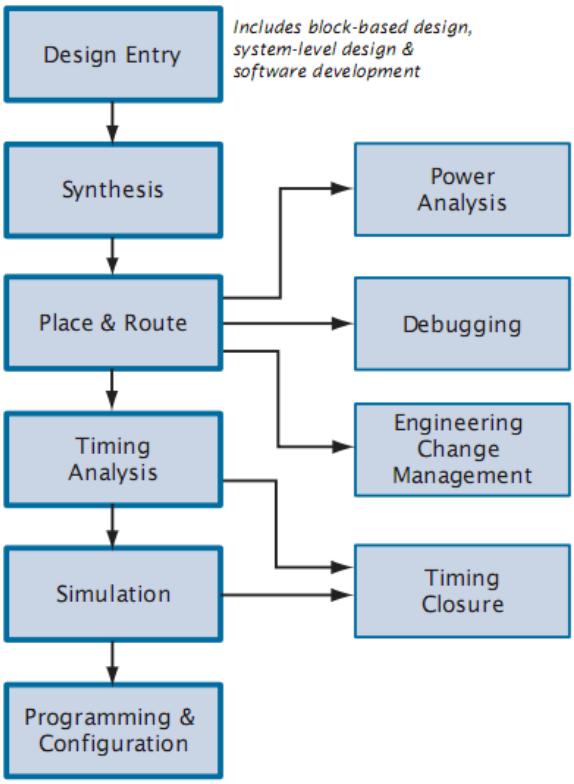
Đề tài sử dụng phương pháp kiểm thử trên FPGA của board DE2 Altera để xác nhận tính đúng đắn của thiết kế trong thực tế. Board DE2 của hãng Altera bao gồm một FPGA CycloneII với khoảng 35000 logic element được kết nối với một số thành phần ngoại vi khác nhau bao gồm: 512K bộ nhớ SRAM, 4MB bộ nhớ Flash, 8MB bộ nhớ SDRAM, kết nối ngõ ra với chuẩn VGA, chuẩn giao tiếp mạng Ethernet, ngõ vào/ra cho cổng âm thanh và các cổng host/device usb. DE2 chứa 2 bộ dao động tạo xung clock là 50Mhz và 27Mhz (sử dụng cho định thời video).

Trong thiết kế đề tài, tài nguyên trên DE2 board chỉ được sử dụng một phần chủ yếu là khối FPGA, khối dao động 50Mhz và cổng UART cũng như switch.

2.5.2 Công cụ phát triển Quartus

Phần mềm Altera QuartusII là công cụ hoàn chỉnh nhất cho việc thiết kế hệ thống trên khả trình chip (system-on-a-programmable-chip SOPC)[23] trên FPGA của Altera hoạt động trên hệ điều hành Windows. Nó cung cấp một môi trường thiết kế hoàn chỉnh và đa nền tảng và dễ dàng phù hợp với các đặc tính thiết kế riêng của nhà thiết kế. QuartusII bao gồm một môi trường phát triển tích hợp IDE và một trình biên dịch (compiler) được sử dụng để chuyển từ ngôn ngữ mô tả phần cứng HDL như Verilog, VHDL, SystemC thành dữ liệu để cấu hình dành cho FPGA.

Công cụ QuartusII chứa thành phần TimeQuest Timing Analyzer cho phép phân tích các đặc tính về thời gian của thiết kế. TimeQuest phân tích sử dụng phương pháp SDC chuẩn công nghiệp đối với các ràng buộc thiết kế và các kết quả báo cáo.



Hình 2.19: Design flow của Quartus [23]

Nhà thiết kế có thể sử dụng thông tin từ TimeQuest Timing Analyzer để phân tích, gỡ lỗi và đánh giá hiệu năng về tần số của thiết kế. Một khi hoàn tất việc biên dịch thiết kế, nhà thiết kế có thể lập trình hoặc cấu hình thiết bị FPGA của Altera sử dụng chức năng của Assembler.

Một khi việc xác nhận hoàn chỉnh, chức năng của thiết kế sẽ được nạp vào FPGA. Khi đó FPGA sẽ thực hiện chức năng như nhà thiết kế mong muốn. Sau quá trình hoàn tất biên dịch thiết kế, các bản báo cáo tổng hợp về tài nguyên được sử dụng bởi mỗi khối trong thiết kế, việc khảo sát tần số hoạt động cũng được thể hiện.

2.5.3 Công cụ mô tả phần cứng VHDL

VHDL là một trong hai dạng ngôn ngữ mô tả phần cứng theo tiêu chuẩn công nghiệp. VHDL được tổ chức US Department of Defense công bố từ năm 1980. Năm 1986, VHDL được đề xuất như một chuẩn IEEE. Nó có nguồn gốc từ ngôn ngữ ADA. VHDL được sử dụng để mô tả cấu trúc phần cứng từ mức trừu tượng đến mức độ khối. Việc sử dụng VHDL phát triển khá nhanh vì nó được khoảng 10000 kỹ sư trên khắp thế giới sử dụng để tạo ra các sản phẩm điện tử tinh vi. VHDL là một ngôn ngữ hỗ trợ mạnh mẽ với rất nhiều cấu trúc ngôn ngữ có khả năng mô tả hành vi rất phức tạp.

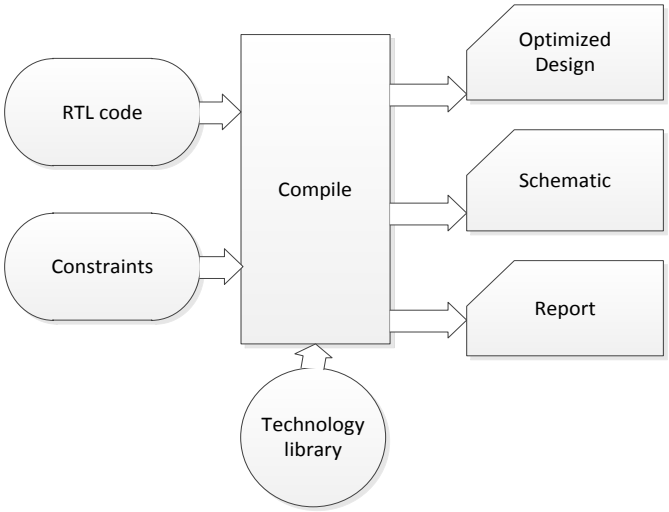
2.5.4 Công cụ Design Compiler

Design Compiler là một công cụ chuyên tổng hợp các thiết kế số của hãng Synopsy[24, 25] trên nền hệ điều hành Linux. Công cụ tổng hợp nhận đầu vào là mã nguồn RTL được viết bằng một trong các loại ngôn ngữ Verilog hoặc VHDL…và thư viện cell chuẩn của nhà sản xuất để đưa ra kết quả ngõ ra là tập tin netlist mức cổng (gate-level netlist). Đây là tập tin thể hiện cấu trúc các cell chuẩn dựa trên thư viện đầu vào.

Trong suốt quá trình tổng hợp, Design Compiler chuyển đổi mã nguồn RTL thành các thành phần lấy từ thư viện kỹ thuật và thư viện DesignWare. Thư viện kỹ thuật bao gồm các cổng logic và flip-flop cơ bản. Thư viện DesignWare chứa các cell phức tạp hơn như bộ cộng, trừ và so sánh có thể được sử dụng để xây dựng các khối thực hiện phép toán học. Design Compiler có thể tự động quyết định khi nào

sử dụng các thành phần trong thư viện DesignWare và nó có thể tổng hợp một cách hiệu quả các thành phần này thành dạng mức cổng (gate-level). Kết quả của tập tin netlist cũng phụ thuộc vào các ràng buộc được đưa ra. Các ràng buộc là đặc tính kỹ thuật của nhà thiết kế về mặt thời gian, diện tích, công suất, điện áp…

Trong thiết kế NoC, các tham số thiết kế như topo, kỹ thuật chuyển mạch, giao thức định tuyến và cơ chế điều khiển luồng đã được đề cập đến trong phần trước. Các tham số này được mô hình hoá bởi ngôn ngữ mô tả phần cứng là VHDL và sau đó được tổng hợp bằng công cụ Design Compiler để chuyển sang netlist ở mức cổng (gate level netlist). Kết quả ngõ ra là tập tin chứa một danh sách các thành phần mạch điện và các kết nối bên trong. Như vậy tập tin gate level netlist là một mô tả về mặt cấu trúc hoàn chỉnh với các cell chuẩn.



Hình 2.20: Sơ đồ mô tả chức năng của Design Compiler

2.5.5 Thư viện thiết kế

Trong đề tài, mô hình SoC ứng dụng NoC được tổng hợp bởi một thư viện cell chuẩn. Thư viện được sử dụng là saed90nm. Đây là thư viện phục vụ cho hướng giáo dục trong việc thiết kế vi điện tử IC với công nghệ 90nm[26]. Các đặc tính của thư viện này:

- Được tối ưu dành cho thiết kế tiêu thụ năng lượng thấp
- Được tối ưu đối với các công cụ EDA của hãng synopsys
- Chỉ dành cho lĩnh vực giảng dạy không dành mục đích chế tạo sản xuất

Thư viện Digital Standard Cell bao gồm 340 cell nhằm để tối ưu thiết kế IC. Nó chứa các cell logic tổ hợp và tuần tự với sức lái (drive strenght) khác nhau. Nó đáp ứng toàn bộ các yêu cầu cho việc thiết kế tiêu thụ ít năng lượng bao gồm hỗ trợ các thiết kế IC với các điện áp lõi khác nhau nhằm giảm thấp nhất vấn đề công suất. Thư viện được thiết kế với điện áp hoạt động 1.2V/2.5V và tần số là 100MHz.

Lý do lựa chọn và sử dụng thư viện saed90nm_typ_ht.lib vì đây là thư viện hỗ trợ đối với lĩnh vực giáo dục.

CHƯƠNG 3
THIẾT KẾ MÔ HÌNH ỨNG DỤNG MẠNG TRÊN CHIP

Chương này sẽ trình bày những phân tích thiết kế và phương pháp xây dựng các thành phần cơ bản của một hệ thống ứng dụng nền tảng truyền thông NoC trong giao tiếp truyền dữ liệu. Đối với thiết kế thành phần router, các phần trình bày sẽ bao gồm: lựa chọn topo mạng, bộ đệm dữ liệu gói tin, giải thuật định tuyến, cơ chế điều khiển luồng. Vai trò của RNI trong giao tiếp mạng trên chip, các vấn đề đối mặt khi thiết kế thành phần RNI, thiết kế mô hình khối slave RNI và master RNI và cơ chế hoạt động của 2 loại RNI sẽ được trình bày trong phần thiết kế thành phần giao tiếp mạng RNI. Đối với các thành phần resource được minh hoạ trong đề tài sẽ bao gồm: thiết kế mô hình master IP với dummy processor, mô hình slave IP với dummy memory và switch…

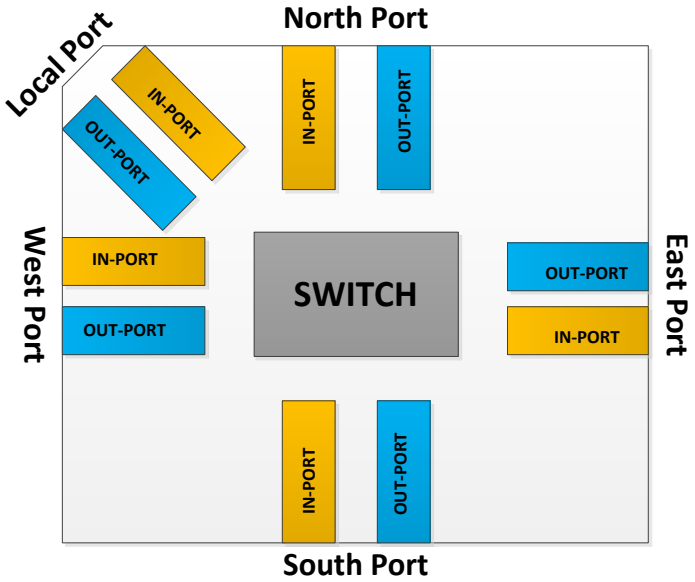
3.1 Thiết kế kiến trúc router

3.1.1 Kiến trúc bên ngoài của router

Đối với mô hình mạng trên chip với topo dạng lưới 2 chiều, thì router sẽ có 5 ngõ vào/ra tương ứng với 4 hướng kết nối với các router lân cận là North, South, East, West và 1 đường kết nối với thành phần resource thông qua thành phần RNI. Tại mỗi hướng kết nối có thể bao gồm các kênh truyền vật lý riêng biệt tương ứng cho ngõ vào, ngõ ra hoặc chỉ duy nhất 1 kênh truyền tại mỗi hướng thực hiện cả chức năng truyền và nhận dữ liệu. Router với thiết kế kênh truyền riêng biệt thực hiện cho chức năng đọc ngõ vào hoặc ghi ngõ ra được gọi là unidirectional router. Ngược lại, với kênh truyền có khả năng thực hiện cả việc truyền và nhận thông tin được gọi là bidirectional router.

Trong thiết kế của đề tài, chúng tôi lựa chọn phương pháp thiết kế các kênh truyền vật lý thực hiện chức năng riêng biệt hoặc chỉ đọc dữ liệu và chỉ ghi dữ liệu.

Việc tách biệt chức năng cho mỗi kênh truyền vật lý sẽ làm đơn giản hoá thiết kế phần cứng của các khối điều khiển hoạt động của router.



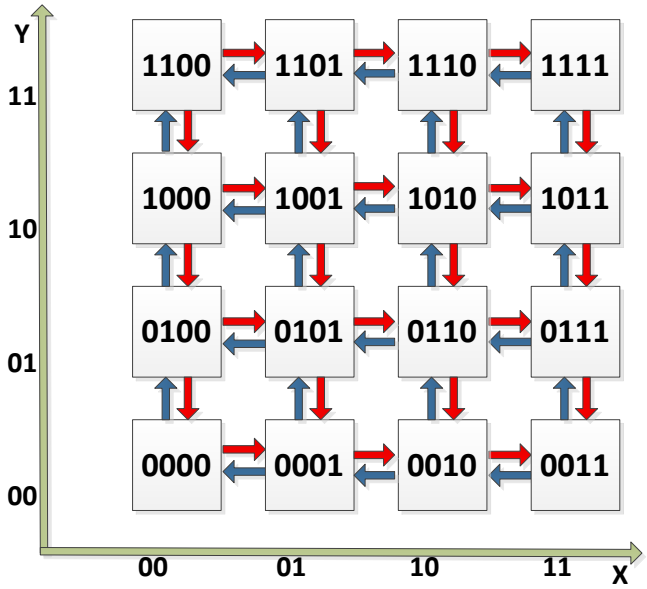
Hình 3.1: Thiết kế router với 2 kênh truyền vật lý vào/ra tại mỗi hướng

Mỗi kênh dẫn vật lý truyền flit có độ rộng n bit phụ thuộc vào định dạng kích thước của flit, đối với mỗi đầu vào và đầu ra tại mỗi hướng cũng như với network adapter thì có 2 kênh dẫn vật lý riêng biệt. Như vậy mỗi kênh dẫn chỉ có khả năng truyền theo 1 chiều duy nhất.

3.1.1.1 Topo mạng

Với một số ưu điểm ở topo dạng lưới 2 chiều đã phân tích ở phần trước, để đáp ứng tốt cho công nghệ chế tạo và đơn giản hóa trong giải thuật định tuyến cũng như cơ chế điều khiển luồng thì đề tài lựa chọn mô hình mạng theo dạng lưới 2 chiều với kích thước 4x4.

Kích thước NoC là một tham số quan trọng cần được quyết định đầu tiên trước khi thiết kế mạng. Kích thước NoC sẽ quyết định số lượng bit yêu cầu để nhận diện node trong mỗi gói tin. Giả sử rằng kích thước tối đa của mạng là 4x4 thì có khoảng 16 resource được phân biệt trên mạng. Vậy cần ít nhất có 4 bit được yêu cầu trong gói tin để thể hiện chi địa chỉ nguồn và địa chỉ đích của resource.



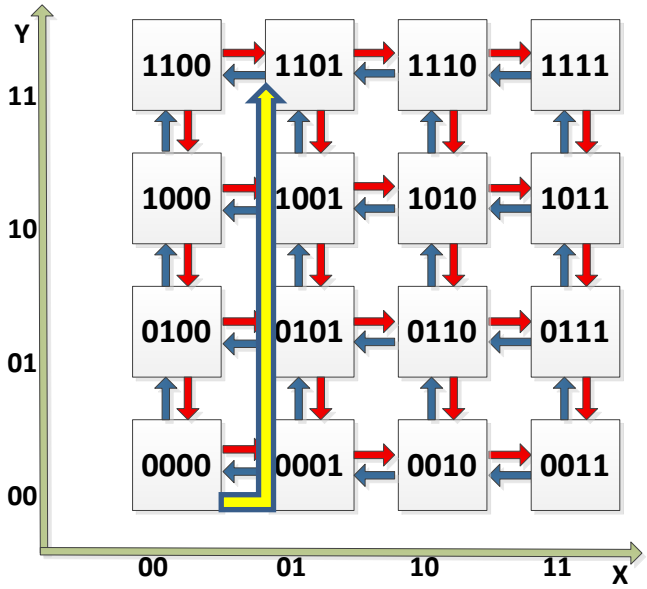
Hình 3.2: Topo dạng lưới 2 chiều với 16node mạng

Trong topo ở hình trên, trên mỗi trục X-Y sẽ có 4 router liên kết với nhau. Địa chỉ trên mạng của các router lần lượt sẽ tương ứng với vị trí theo trục Ox hoặc Oy trên mô hình lưới 2 chiều. Trong đó 2 bit đầu mã hóa vị trí của router theo trục Ox và 2 bit sau mã hóa vị trí của router theo trục Oy. Địa chỉ này là duy nhất và dùng để phân biệt các router cũng như các node mạng với nhau.

3.1.1.2 Giải thuật định tuyến

Trong các giải thuật định tuyến được trình bày ở phần trước, đối với mô hình mạng dạng lưới 2 chiều thì giải thuật định tuyến phù hợp được sử dụng là định tuyến OXY theo kích thước và bậc. Giải thuật này được chọn cho việc đơn giản hóa trong quá trình xây dựng một router.

Hình minh họa sau đây, một resource tại node có địa chỉ giả sử là 0000 gửi dữ liệu đến resource tại node có địa chỉ 1101 thì trước tiên dữ liệu sẽ được định tuyến dựa trên vị trí dữ liệu trên router hiện hành theo chiều Ox, sau khi vị trí của gói tin trùng với vị trí Ox trong địa chỉ của router đích thì gói tin tiếp tục được định tuyến theo chiều Oy. Sau đó, gói tin sẽ di chuyển theo chiều Oy cho đến khi đến được router đích là 1101.



Hình 3.3: Định tuyến Oxy cho mô hình dạng lưới 2 chiều

3.1.1.3 Cấu trúc khung dữ liệu

Nhằm đơn giản hoá trong phần thiết kế bộ đệm, bộ điều khiển chuyển mạch trong kiến trúc của router. Một gói tin trên mạng chỉ chứa một flit. Có 3 loại gói tin trong mạng là:

- Gói tin ghi (write flit)
- Gói tin yêu cầu đọc (read request flit)
- Gói tin trả lời yêu cầu đọc (read return flit)

Write flit

48-43	42	41	40	39-36	35-8	7-0
x-y routing	Read	Write	Return	Destination Node ID	Memory address	Data

Read request flit

48-43	42	41	40	39-36	35-8	7-4	3-0
x-y routing	Read	Write	Return	Destination Node ID	Memory address	Source Node ID	zeros

Return flit

48-43	42	41	40	39-36	35-28	27-0
x-y routing	Read	Write	Return	Destination Node ID	Data memory	zeros

Trong đó:

- Gói tin write được gửi bởi master IP là một dummy processor để ghi dữ liệu lên một slave IP là dummy memory
- Gói tin read request được gửi bởi master IP đến slave IP và sau đó slave IP sẽ trả lời với gói tin read return.

Tất cả các gói tin đều có kích thước cố định là 49 bit trong đó một số phần trong gói tin chứa các bit 0. Trong tất cả các gói tin có 3 loại bit cờ là write, read, return ở vị trí bit 40, 41, 42 trong gói tin. Một bit thể hiện cho một cờ khác nhau, nhằm chỉ ra rằng gói tin thuộc loại nào. Trong 3 bit cờ khác nhau đó thì chỉ 1 trong 3 bit được bật lên và nếu là gói tin rỗng (empty packet) thì 3 bit này được đặt về 0.

Ngoài ra, trong cả 3 loại gói tin đều yêu cầu thông tin của node mà các gói tin muốn gửi đến là node ID đích. Số node tương ứng trên mạng lưới kích thước 4x4 là 16 node. Như vậy cần 4 bit bao gồm 2 bit cho trục OX và 2 bit cho trục OY để mã hoá vị trí cho mỗi node. Các gói tin write và read request yêu cầu một địa chỉ trong bộ nhớ, nhằm cung cấp thông tin cần ghi và đọc từ địa chỉ bộ nhớ nào trên slave IP trong trường hợp resource là bộ nhớ.

Gói tin read request chứa node ID nguồn bởi bên nhận gói tin read request cần phải biết node nào trên mạng gửi yêu cầu đọc dữ liệu. Gói tin read return nhằm trả lời cho gói read request và gói tin này chỉ chứa node ID đích và dữ liệu đọc từ bộ nhớ. Gói tin write chứa 8 bit dữ liệu ghi trong khi gói tin read return chứa 8 bit dữ liệu đọc.

Tất cả các gói tin có 6 bit ở vị trí bit 43 đến 48 để phục vụ cho việc định tuyến: 3 bit cho mỗi chiều X và Y. Các bộ đếm này được khởi tạo với khối RNI và thay đổi bởi router. Như đã trình bày, kích thước mạng cố định là 16 node vì thế 2 bit được sử dụng để mã hoá vị trí của node trên lưới 2 chiều 4x4 và 1 bit để chỉ ra chiều di chuyển của gói tin trên trục X và Y. Nếu một mô hình lưới có kích thước lớn hơn thì số lượng bit này cần được thay đổi sao cho phù hợp.

3.1.1.3 Phân tích cơ chế điều khiển luồng

Với phương pháp thiết kế mỗi gói tin chỉ chứa một flit. Kích thước flit bằng kích thước độ rộng của kênh truyền vật lý kết nối giữa các router. Như vậy trong một phiên giao tiếp giữa node nguồn và node đích thì gói tin sẽ được lưu trữ toàn bộ sau đó lần lượt được gửi đến các router tương ứng theo luật định tuyến mà không phân tán gói tin trên các node mạng.

Trong trường hợp gói tin được chia nhỏ thành nhiều flit như head flit, body flit và tail flit thì việc phân tán các phần của một gói tin trên các router khác nhau có thể gây ra một số tình trạng tắc nghẽn trên mạng. Để giải quyết các vấn đề về tắc nghẽn này thì một số giải pháp như xây dựng bộ phân phối kênh ảo, bộ đệm với kích thước lớn để chứa toàn bộ hoặc một phần gói tin và phải xây dựng cơ chế điều khiển bộ đệm ngõ vào và ngõ ra hoạt động một cách nhịp nhàng.

Cơ chế điều khiển luồng được sử dụng trong thiết kế của đề tài là Store and Forward. Trong cơ chế này chỉ yêu cầu một bộ đệm có kích thước nhỏ, gói tin sẽ được chuyển đến router lân cận nếu router này có không gian bộ nhớ chứa gói tin.

3.1.2 Kiến trúc bên trong của router

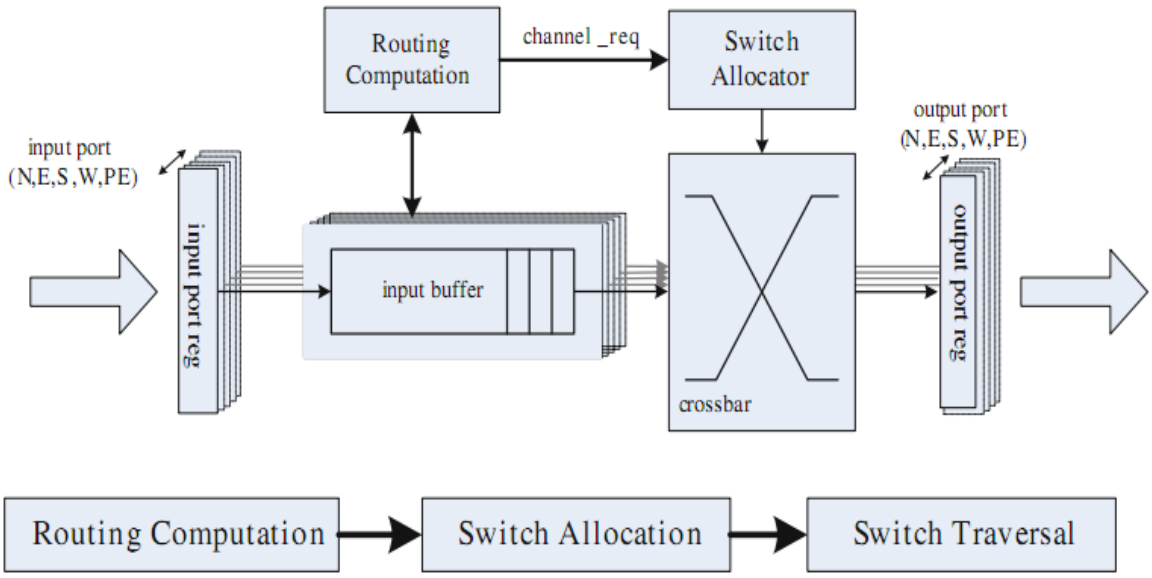
Đề tài sử dụng phương pháp tiếp cận với mô hình NoC theo hướng đồng bộ. Tức các router hoạt động đồng bộ theo nguồn clock cố định. Tần số hoạt động của router được cung cấp với xung clock toàn cục (global clock) và tín hiệu reset của hệ thống. Một gói dữ liệu truyền từ ngõ vào để đến được ngõ ra cần một số chu kỳ clock cụ thể. Các trạng thái cơ bản của gói dữ liệu đi qua router là

BW	RC	ST	LT

- BW (buffer write): flit từ router lân cận đi đến ngõ vào của router hiện tại và flit được ghi lên bộ đệm.
- RC (routing compulation): Dựa vào các bit thông tin đặc trưng được thêm vào mỗi flit, router sẽ tính toán ngõ ra tương ứng của khung này là hướng North, South, East hoặc West nếu flit được đi vào router từ một resource.
- SA (switch allocation): Flit này sẽ được đưa đến Crossbar. Flit đợi cho đến khi nó được chọn để đẩy ra ngoài.

- ST (switch traversal): Flit di chuyển qua kênh truyền vật lý.

Như vậy có 4 trạng thái cơ bản mà một flit phải trải qua khi đi qua một router. Quá trình trên tạo ra độ trễ lớn khi các khung đi qua nhiều router trung gian

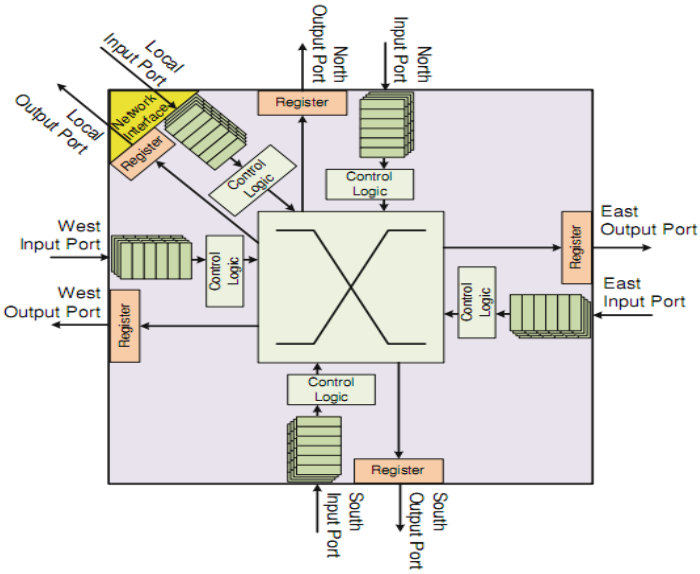


Hình 3.4: Sơ đồ các bước xử lý đối với một router cơ bản [1]

Chức năng chính của các router là định tuyến các flit từ ngõ vào để đi ra các cổng ngõ ra tương ứng trên địa chỉ trên flit để flit có thể đi đến router đích. Các thành phần chính tạo nên router là các bộ đệm ngõ vào và bộ đệm ngõ ra (buffer in/out), bộ tính toán định tuyến (route computation), bộ cấp phát kênh ảo (virtual channel allocator), bộ cấp phát chuyển mạch (switch allocator) và thành phần chuyển mạch giữa các hướng khác nhau (crossbar switch).

Trong các mạng chuyển mạch gói, resource sẽ gửi dữ liệu là các gói tin đến router. Gói tin nhận tại ngõ vào của router. Bộ đệm lưu trữ gói tin trước khi gửi nó đến ngõ ra của router. Mỗi ngõ ra của router có một bộ dồn kênh. Nó nhận các gói tin từ các hàng đợi khác nhau và sau đó chọn một gói tin để gửi ra bên ngoài để đến đầu vào của router tiếp theo. Các bộ điều khiển hướng đi và bộ đệm là thành phần chính quan trọng trong router. Các giao thức và thuật toán định tuyến được thực hiện trong khối này.

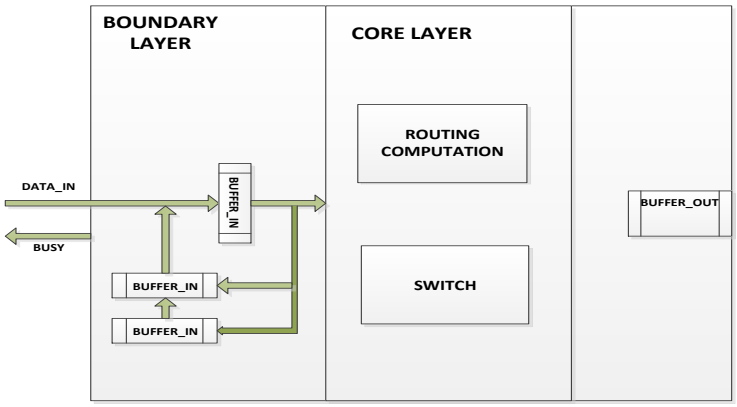
Một kiến trúc trong router sẽ quyết định chính xác độ trễ trên đường dẫn, thông số này ảnh hưởng đến độ trễ trên từng router và độ trễ trên toàn mạng. Ngoài ra kiến trúc router cũng tác động đến năng lượng tiêu thụ của mạng. Việc thực hiện định tuyến, điều khiển luồng và cơ chế pipeline sẽ ảnh hưởng đến hiệu quả sử dụng của các bộ đệm và kênh truyền.



Hình 3.5: Kiến trúc tổng quát bên trong của router [1]

Kiến trúc bên trong router bao gồm 2 lớp chính:

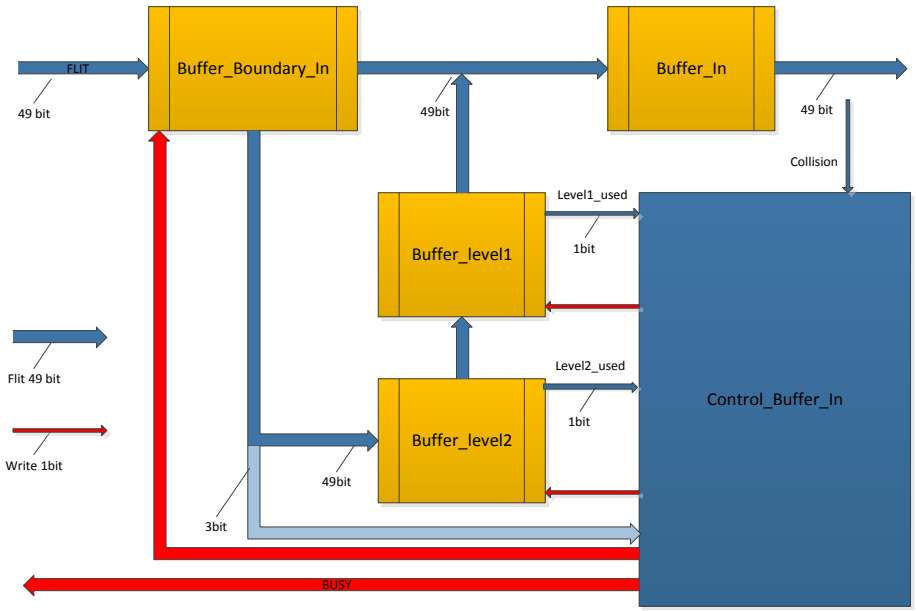
- Lớp biên (boundary layer). Ở lớp này chức năng chính là lưu giữ các flit ở các bộ đệm và bộ điều khiển bộ đệm quản lý các tín hiệu ngõ vào.
- Lớp lõi (internal layer). Chức năng chính là xử lý thông tin địa chỉ nguồn và đích của flit từ đó đưa ra quyết định định tuyến các flit đến ngõ ra kênh dẫn tương ứng hoặc được chuyển ngược lại về các bộ đệm ở lớp biên.



Hình 3.6: Kiến trúc tổng quát bên ngoài của một router

3.1.2.1 Thiết kế bộ nhớ đệm cho gói tin

Trong kiến trúc tổng quát bên trong router, mỗi cổng vào đều có một khối đệm được dùng làm vùng nhớ cho dữ liệu đi vào từ một cổng. Bên cạnh đó ở các cổng ngõ ra cũng dùng một khối đệm cho dữ liệu ở ngõ ra. Kích thước độ rộng và sâu của một bộ đệm thì tùy thuộc vào mỗi thiết kế router khác nhau. Trong đó, độ rộng của bộ đệm thì tùy thuộc vào kích thước định dạng của một flit dữ liệu được truyền tải trên kênh truyền vật lý. Độ sâu tức khả năng chứa được tối đa bao nhiêu flit của một bộ đệm thì có thể thay đổi và linh hoạt phụ thuộc vào cơ chế điều khiển luồng được áp dụng trong thiết kế.



Hình 3.7: Cấu trúc khối bộ đệm và luồng dữ liệu tại ngõ vào router

Trong phương pháp thiết kế của đề tài, mỗi gói tin chỉ bao gồm 1 flit vì thế kích thước bộ đệm sẽ được giảm bớt. Bộ đệm nhận dữ liệu ngõ vào hoạt động với nguyên tắc:

- Gói tin đi từ router lân cận khi đi vào một router khác sẽ được lưu trữ tạm thời trên một bộ đệm buffer_boundary_in. Buffer_level1 và buffer_level2 là hai bộ đệm riêng biệt mỗi bộ đệm lưu trữ được một gói tin có kích thước độ rộng 49 bit.
- Tình trạng sử dụng của mỗi buffer_level1 và buffer_level2 được thể hiện qua tín hiệu level1_used và level2_used. Tín hiệu này nhằm báo bộ đệm đó đang được sử dụng . Tín hiệu này sẽ đi đến khối control_buffer_in. Ngoài ra, trong nội dung của gói tin từ router lân cận gửi đến thì bit 42, 41 và 40 sẽ đi đến khối control_buffer_in để khối này nhận biết là có gói tin read request, write hoặc return mới đến. Từ các thông tin về tình trạng các bộ đệm, tình trạng xung đột mà control_buffer_in sẽ điều khiển trạng thái của buffer_level1, buffer_level2 và buffer_in.
- Nếu buffer_level1 được sử dụng thì giá trị này sẽ được ghi vào buffer_in của lớp bên trong để tiếp tục cho quá trình xử lý tiếp theo.

Chi tiết cơ chế hoạt động của buffer_level1

- Nếu buffer_level1 được sử dụng và có tranh chấp thì giữ nguyên trạng thái của buffer_level1 và giữ trạng thái tranh chấp.
- Nếu buffer_level1 và buffer_level2 được sử dụng và không có tín hiệu tranh chấp thì nội dung buffer_level1 được đưa vào bộ đệm ở lớp trong, nội dung buffer_level2 được ghi lên buffer_level1, tín hiệu báo buffer_level1 đã sử dụng.
- Trong trường hợp có 1 trong các loại gói tin mới đến được ghi vào bộ đệm lớp biên của router:
  - Nếu kèm theo tín hiệu cho biết buffer_level1 được sử dụng, buffer_level2 chưa sử dụng, và không xảy ra tranh chấp thì nội dung buffer_level1 được

đưa vào bộ đệm lớp bên trong của router, nội dung của gói tin mới đến sẽ được ghi vào buffer_level1, tín hiệu báo buffer_level1 được bật.

- Nếu kèm theo tín hiệu báo tranh chấp và buffer_level1 và buffer_level2 chưa được sử dụng thì nội dung gói tin mới đến sẽ được ghi vào buffer_level1 và tín hiệu báo sử dụng buffer_level1 được bật.

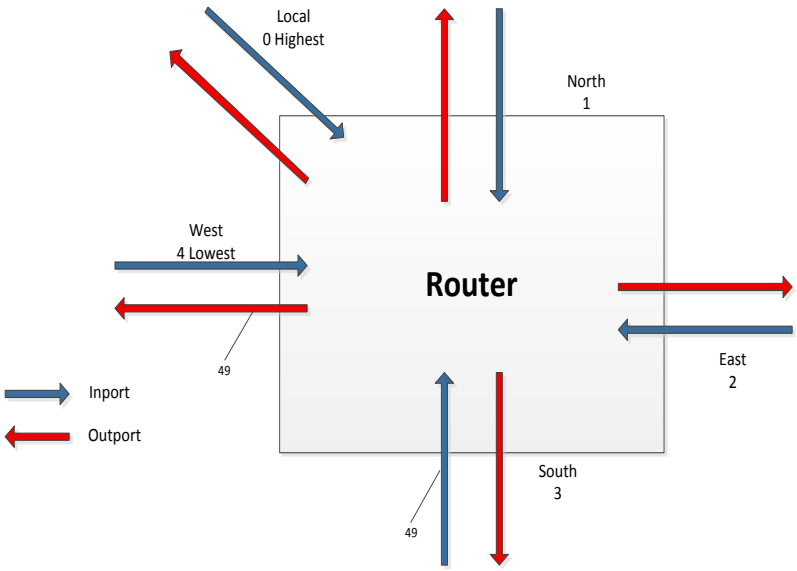
Cơ chế hoạt động của buffer_level2

- Nếu buffer_level1 và buffer_level2 được sử dụng kèm theo tín hiệu báo tranh chấp thì nội dung buffer_level2 giữ nguyên trạng thái và tín hiệu báo buffer_level2 đang được sử dụng vẫn duy trì.
- Trong trường hợp có 1 trong các dạng gói tin mới đến được ghi vào bộ đệm lớp biên của router:
  - Nếu kèm theo buffer_level1 và buffer_level2 được sử dụng và không có tín hiệu báo tranh chấp thì nội dung trên bộ đệm lớp ngoài sẽ được ghi vào buffer_level2 và tín hiệu báo buffer_level2 sử dụng được bật.
  - Nếu kèm theo buffer_level1 được sử dụng, buffer_level2 không sử dụng và tín hiệu báo tranh chấp thì nội dung trên bộ đệm lớp ngoài sẽ được ghi vào buffer_level2 và tín hiệu báo buffer_level2 sử dụng được bật.

3.1.2.2 Cơ chế xử lý tranh chấp

Trong quá trình truyền thông giữa các router trên mạng đôi khi xảy ra các trường hợp tranh chấp. Tranh chấp xảy ra khi có 2 hay nhiều gói tin ở ngõ vào muốn di chuyển đến cùng một kênh truyền ngõ ra. Để kiểm soát vấn đề tranh chấp trên mạng, thuật toán định tuyến sẽ xem xét mỗi ngõ vào một cách riêng biệt trong một vòng lặp với thứ tự ưu tiên cố định sau:

- 0 ở hướng local kết nối với resource có độ ưu tiên cao nhất
- 1 ở hướng north kết nối với router lân cận
- 2 ở hướng east kết nối với router lân cận
- 3 ở hướng south kết nối với router lân cận
- 4 ở hướng west kết nối với router lân cận có độ ưu tiên thấp nhất



Hình 3.8: Độ ưu tiên các cổng trên router

Ở mỗi ngõ vào, thuật toán định tuyến sẽ chọn một gói tin hoặc từ bộ đệm nếu kênh dẫn ngõ vào được sử dụng hoặc một gói tin empty nếu kênh dẫn trong trạng thái rãnh (idle). Nếu dữ liệu nhận được là một trong các gói tin read request, write, return thì chiều ngõ ra mong muốn được tính toán và lưu trong một biến. Mỗi đầu vào cũng có một cờ tranh chấp (collision flag) tương ứng.

Mỗi bộ đệm ngõ vào của router có 1 tín hiệu busy và nó được đặt lên 1 khi bộ đệm ngõ vào được sử dụng. Đầu tiên, thuật toán định tuyến sẽ kiểm tra tín hiệu báo busy ở ngõ vào hiện tại và nếu cờ này báo bận thì sau đó cờ tranh chấp sẽ được đặt lên 1. Một vòng lặp được thực hiện từ ngõ vào 0 tức đối với ngõ kết nối với resource đến ngõ vào hiện tại. Điều này bỏ qua các gói tin empty và nếu có bất kỳ gói tin nào muốn di chuyển đến cùng ngõ ra thì collision flag sẽ được đặt lên 1.

Sau khi kiểm tra tất cả, cờ collision flag được chuyển đến phần tiếp theo của quá trình xử lý để quyết định trạng thái tiếp theo của các kênh dẫn ngõ ra và các bộ đệm ngõ vào. Nếu một collision flag được đặt thì một số gói tin cần được lưu trữ ở bộ đệm ngõ vào. Mỗi router có 2 cấp bộ đệm cho mỗi kênh ngõ vào. Việc thiết kế 2 cấp bộ đệm là yếu tố cần thiết nhằm bù khoảng thời gian trì hoãn xảy ra khi router hiện hành muốn thông báo cho router lân cận biết về tình trạng busy hiện thời của bên nhận flit và mong muốn router không gửi bất cứ gói tin nào trên kênh dẫn này.

Đặc biệt, nếu collision flag được đặt đối với một ngõ vào thì gói tín ngõ vào tương ứng được lưu vào trong bộ đệm cấp 1 (first level buffer), một cờ có tên là First Level Buffer Used sẽ được đặt lên 1 và đồng thời một tín hiệu busy được gửi ngược lại cho phía gửi gói tin.

Tuy nhiên, tín hiệu busy này chỉ nhằm để bên gửi flit xem xét để gửi flit tiếp theo và một gói tin thứ 2 có lẽ đã sẵn sàng từ bên gửi truyền đến bên nhận. Nếu một gói tin thứ 2 được gửi trong lúc đó, thì gói tin này sẽ được lưu vào bộ đệm cấp 2 (second level buffer) và khi đó thì cờ "Second Level Buffer Used" sẽ được đặt. Nếu một gói tin được gửi trong khi một tín hiệu busy được bật thì gói tin đó sẽ bị loại bỏ nếu cả hai bộ đệm đều đã trong tình trạng đang được sử dụng. Các tín hiệu busy_in nhằm chỉ rõ tình trạng sử dụng đối với mỗi kênh dẫn.

3.1.2.3 Cơ chế định tuyến của router

Phương pháp định tuyến XY được sử dụng trong thiết kế. Để dịch chuyển các gói tin thông qua toàn mạng từ node gửi đến node nhận, router sẽ tìm tại các bộ đếm x-y tại bit 43-48 của gói tin đi đến router. Việc quyết định gói tin sẽ đi theo hướng North hay South trên chiều dọc của mạng lưới 2 chiều sẽ được dựa trên thông tin của phần y-flag và y-counter. Trước tiên, router sẽ kiểm tra thông tin ở chiều dọc. Nếu y-flag được đặt là 0 thì gói tin được di chuyển đến hớng South. Ngược lại, nếu y-flag được đặt là 1 thì gói tin sẽ được di chuyển đến hướng North. Khi gói tin được đi đến router kế tiếp, thì giá trị trong phần y-counter được giảm đi 1 đơn vị.

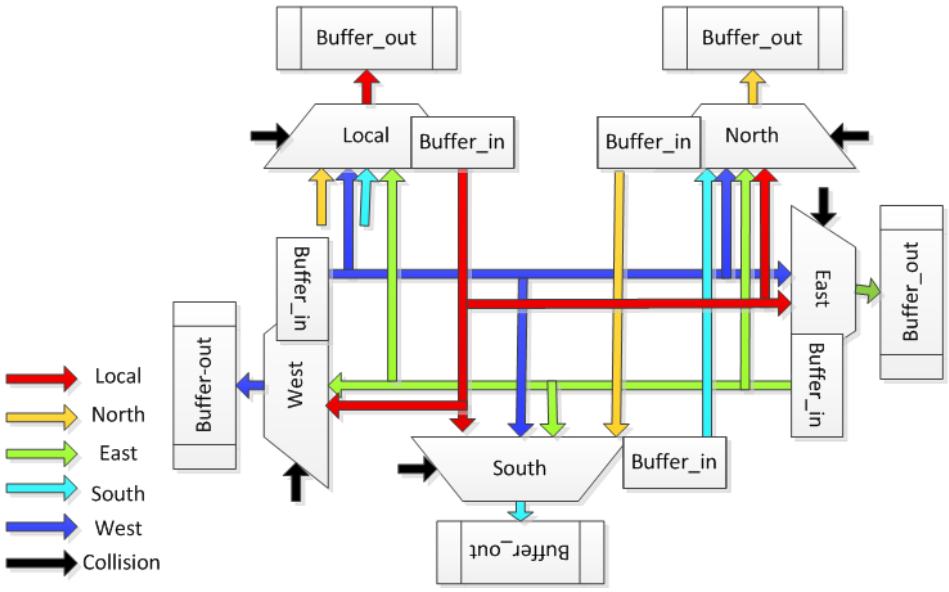
Nếu router nhận được một gói tin với phần y-counter là 0, thì thuật toán định tuyến sẽ xem xét tiếp tục ở phần x-flag và x-counter để quyết định sẽ di chuyển gói tin theo hướng nào của chiều ngang. Cơ chế xử lý trong phần này cũng tương tự như đối với xử lý quyết đi hướng đi theo chiều dọc của gói tin. Nếu x-flag đặt là 0 thì gói tin sẽ được di chuyển đến hướng East. Trong trường hợp ngược lại, nếu x-flag đặt là 1 thì gói tin sẽ được di chuyển đến hướng West.

Trường hợp, nếu router nhận được một gói tin mà thông tin trong cả x-counter và y-counter đều được đặt là 0 thì router nhận ra rằng gói tin đã đi đến được đích. Hướng tiếp theo gói tin cần đi sẽ là hướng Local thông qua RNI để đi đến resource.

3.1.2.4 Khối chuyển mạch Cross bar

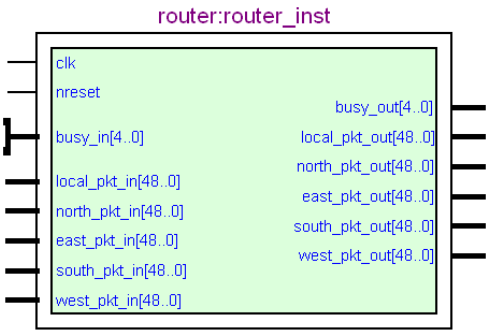
Do đặc điểm của giải thuật định tuyến XY, các gói dữ liệu sẽ được truyền theo phương X trước. Do đó, các gói dữ liệu ở ngõ vào hướng North và South (phương Y) sẽ không được phép chuyển đến ngõ ra hướng East và West (phương X) và tất cả các dữ liệu từ hướng East, hướng West (phương X) và Local sẽ được truyền đến tất cả các hướng còn lại.

Đối với ngõ kết nối với resource thì dữ liệu từ 4 hướng North, South, West, East đều có thể truyền về resource. Đặc tính truyền nhận dữ liệu giữa các hướng có thể được thể hiện bằng hình dưới đây.



Hình 3.9: Cấu trúc khối chuyển mạch router

3.1.2.5 Mô hình router tổng hợp bởi HDL

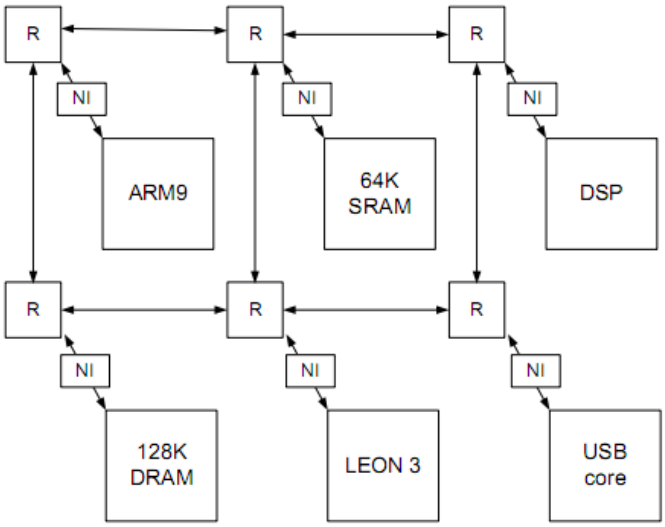


Hình 3.10: Mô hình hoá các tín hiệu trên router

- Chuyển gói tin đến các router lân cận và network adapter trên node.
- Độ ưu tiên trong định tuyến gói tin theo thứ tự ngõ vào lần lượt là: local, North, East, South, West
- Mỗi router có 2 cấp bộ đệm
- Các gói tin được lưu trữ trong các bộ đệm khi một ngõ vào có độ ưu tiên cao hơn đồng thời muốn truyền dữ liệu đến cùng một ngõ ra, dữ liệu ngõ vào được lưu trữ trong các bộ đệm nếu ngõ ra là busy.
- Một ngõ ra được đặt ở trạng thái busy khi bộ đệm ngõ vào đang được sử dụng và do đó không thể nhận bất cứ gói tin dữ liệu ngõ vào.

3.2 Thiết kế thành phần giao tiếp mạng

Một khía cạnh quan trọng trong việc truyền thông dữ liệu trên chip đó là các phương pháp để các resource là các lõi xử lý, bộ nhớ, ngoại vi được tích hợp vào hệ thống. Điều này bao gồm các phương pháp để kiểm tra các resource có phù hợp với các ràng buộc về đặc tính vật lý về các đặc tính điện và thời gian cho việc tích hợp hay không? Một giao diện kết nối, còn được gọi là một RNI, cần được thêm vào để gắn kết phần mạng trên chip với phần IP core. Chức năng của thành phần này là để phần mạng và các thành phần resource của mạng có thể truyền thông dữ liệu với nhau.



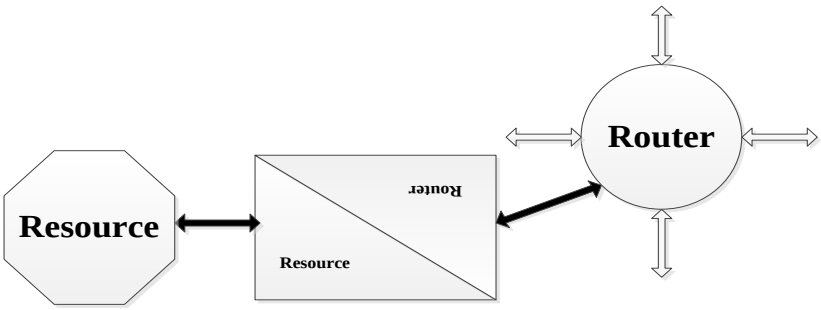
Hình 3.11: Vị trí của RNI trên hệ thống mạng trên chip [12]

Mục đích của phần này là nhằm tập trung mô tả cấu trúc thiết kế bên trong của RNI cho dạng định tuyến phân tán, mô tả cấu trúc top-level, các tín hiệu điều khiển và các giao thức giao tiếp của khối RNI.

Các resource trên mạng có thể đảm nhiệm một số nhiệm vụ khác nhau. Do đó, một resource có thể là một bộ xử lý, bộ xử lý tín hiệu số (DSP), bộ nhớ hoặc một khối ASIC nhằm thực hiện một công việc giao tiếp song song hoặc tuần tự. Như vậy, resource này cần được thiết kế sử dụng cùng công nghệ của bộ chuyển mạch trên mạng (router) và đặc tính về điện khi hoạt động của nó cần phải phù hợp với các ràng buộc về đặc tính điện áp.

Mỗi resource yêu cầu một giao diện để kết nối đến phần mạng chuyển mạch dạng gói tin. Hình dưới thể hiện một dạng RNI mẫu. Giao diện kết nối này có chức năng giống như một network card trên máy tính cá nhân, thành phần này được thêm vào nhằm để máy tính có khả năng kết nối mạng Ethernet. RNI cần được thực hiện tại lớp thấp nhất của 3 lớp dưới trong mô hình OSI. Điều này có nghĩa là RNI có khả năng cho phép một resource giao tiếp với một mạng sử dụng các gói tin.

Thiết kế bên trong của một RNI có thể được phân chia thành 2 phần. Một phần được kết nối với mạng thì độc lập với phần resource là các bộ xử lý, bộ nhớ…và phần còn lại được kết nối với phần resource thì phụ thuộc vào phần resource và độc lập với mạng. Phần độc lập với resource có thể được tái sử dụng nhằm giao tiếp với bất kỳ resource khác. Phần này chỉ phụ thuộc và thiết kế của bộ chuyển mạch (router).



Hình 3.12: Thành phần và kết nối của RNI trong hệ thống mạng trên chip

Thành phần phụ thuộc resource sẽ phụ thuộc vào các giá trị tín hiệu trên các chân I/O như các bus địa chỉ và bus dữ liệu hoặc điều khiển. Nếu nhiều resource có

cùng một chuẩn giao diện về các chân ngoại vi giao tiếp thì RNI có thể được sử dụng cho các resource này. Chức năng chính của phần phụ thuộc vào resource trên RNI là chuyển thông tin từ bus dữ liệu thành dạng gói tin và chuyển gói tin thành dữ liệu phù hợp với bus dữ liệu.

Phần độc lập với resource cần quan tâm đến các vấn đề về thời gian (timing), về bộ đệm dữ liệu và các khía cạnh về đồng bộ hoá trong suốt quá trình giao tiếp dữ liệu. Trong trường hợp định tuyến nguồn (source routing), RNI còn chứa các bảng định tuyến và chứa thêm phần thêm thông tin đường dẫn hoàn chỉnh vào trong flit đầu tiên của gói tin (head flit).

3.2.1 Các biến thể của RNI

Thiết kế RNI được phân thành theo hướng định tuyến phân tán (distributed routing) và hướng định tuyến nguồn (source routing). Đối với RNI thuộc loại định tuyến phân tán thì địa chỉ đích (destination address) được thêm vào trong nội dung của gói tin được chuyển đi. Ngược lại, nếu thuộc loại source routing, một thông tin định tuyến hoàn chỉnh sẽ được thêm vào trong nội dung của gói tin được chuyển đi. Đề tài lựa chọn phương pháp thiết kế RNI theo loại định tuyến phân tán.

3.2.2 Các vấn đề đối mặt khi thiết kế RNI

3.2.2.1 Thiết kế bộ nhớ đệm trong RNI

Các gói tin có thể có kích thước khác nhau. Kích thước gói tin phụ thuộc vào kích thước của thông điệp (message) mà resource muốn gửi đi. Với gói tin được gửi từ resource nguồn đến resource đích thì trước tiên sẽ được lưu trữ tạm thời trong bộ đệm RNI. Để lưu trữ hoàn chỉnh một gói tin bao gồm nhiều flit thì RNI cần có kích thước bộ đệm bằng với kích thước của gói tin lớn nhất. Điều này dẫn đến chi phí tăng. Trong giải pháp của đề tài, phương pháp thiết kế sử dụng 1 flit cho mỗi gói tin. Như vậy, thiết kế không sử dụng bộ đệm lớn để lưu trữ toàn bộ gói tin. Thiết kế nhằm cho phần cứng đơn giản và giảm chi phí.

3.2.2.2 Địa chỉ đối với RNI

Mỗi resource cần một địa chỉ nhất định trong mạng để giao tiếp với các resource khác trên mạng. Một resource đích cần biết địa chỉ chính xác của resource

gửi dữ liệu đến để phản hồi lại dữ liệu tương ứng. RNI được kết nối đến resource sẽ chia sẻ địa chỉ với resource, bởi trong mô hình NoC chỉ một RNI được kết nối với một resource tương ứng.

Có 2 phương pháp để xác định địa chỉ nguồn và đích của resource. Phương pháp thứ nhất là bản thân resource sẽ tự thêm địa chỉ nguồn của chính nó trong trong gói tin khi gửi nó đến RNI. Phương pháp thứ hai đó là RNI sẽ thêm địa chỉ nguồn vào trong flit đầu tiên của gói tin trong quá trình chuyển dữ liệu được gửi từ resource đến router.

Phương pháp được lựa chọn trong đề tài là phương pháp thứ 2. Tức RNI sẽ thực hiện nhiệm vụ thêm địa chỉ nguồn vào gói tin và chuyển đến router.

3.2.2.3 Giao diện giao tiếp với resource

Một resource là một khối chức năng được thiết kế trước đó, thông thường là thực hiện một chức năng xác định mà có thể được sử dụng đối với thiết kế của một SoC. Trong đề tài, các resource được gắn kết vào mạng bao gồm: dummy processor, dummy memory, uart và switch.

Trong đó, dummy processor đóng vai trong là thiết bị master và các thành phần còn lại đóng vai trò là thiết bị slave trong mối quan hệ master slave giữa các thiết bị với nhau. Thiết bị master sẽ hoạt động với vai trò gửi các gói tin write và read request đến các các thiết bị slave. Các thiết bị slave sẽ hoạt động với vai trò trả lời gói tin read request bằng gói tin read return cũng như ghi nhận dữ liệu được truyền từ master.

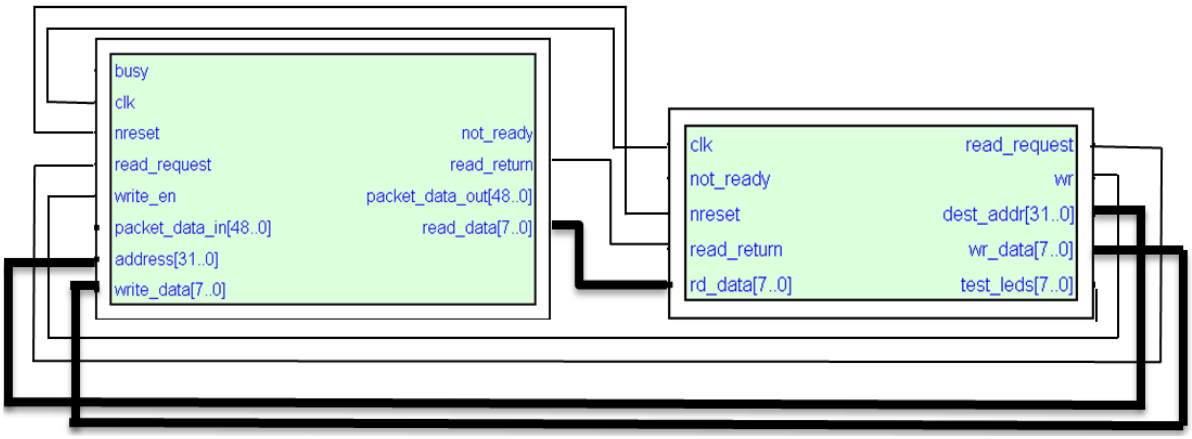
Như vậy, yêu cầu đặt ra là thiết kế 2 loại RNI riêng biệt tương ứng với 2 loại thiết bị master và slave. Trong đó:

- Network adapter master kết nối với thành phần bộ xử lý dummy processor
- Network adapter slave kết nối với thành phần: memory, uart, switch. Network adaptor kết nối với memoy, uart, switch là như nhau về các tín hiệu và nguyên lý giao tiếp.

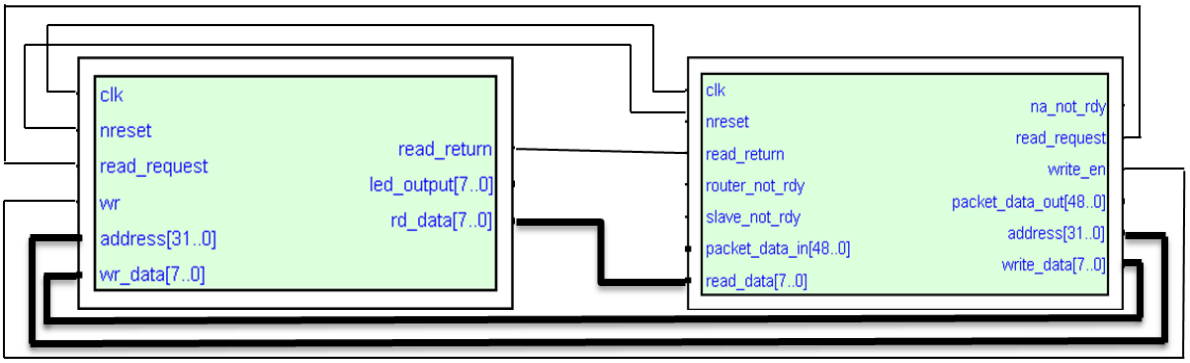
Ngoài ra đối, với thiết bị slave là memory, switch yêu cầu RNI có khả năng nhận dữ liệu từ các thiết bị này. Với việc áp dụng cơ chế req/ans trong việc điều

khiển luồng giao tiếp giữa RNI và resource đề tài sử dụng 2 tín hiệu 1 bit là request và return. Trong đó, tín hiệu request gửi từ RNI đến resource để yêu cầu đọc dữ liệu và tín hiệu return được trả về khi resource chấp nhận trả về dữ liệu tương ứng.

Độ rộng bus dữ liệu từ resource đến RNI là 8 bit cho dữ liệu ghi và 8 bit cho dữ liệu đọc. Trong đó, độ rộng của mỗi tín hiệu điều khiển là 1 bit bao gồm: read request, return, write. Việc giao tiếp giữa resource và RNI bán song công (half-duplex). RNI có thể hoặc nhận gói tin từ resource hoặc gửi gói tin đến resource .



Hình 3.13: Giao diện giao tiếp giữa master RNI với resource



Hình 3.14: Giao diện giao tiếp giữa slave RNI với resource

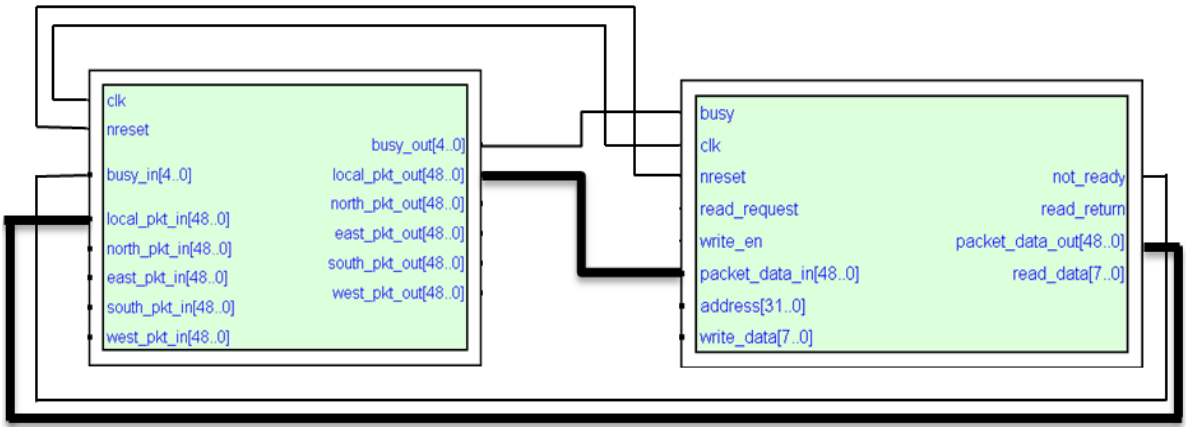
3.2.2.4 Giao diện giao tiếp với router trên mạng

Trong việc giao tiếp giữa RNI và router tồn tại 2 phương pháp. Phương pháp thứ nhất được gọi là credit based scheme và tuỳ chọn thứ hai được gọi là ready to receive based scheme. Đây là 2 phương pháp điều khiển luồng cơ bản giữa thành phần RNI với router trên hệ thống mạng. Trong phương pháp credit based scheme , router sẽ luôn gửi một tín hiệu đến RNI để thể hiện số lượng không gian trống trong

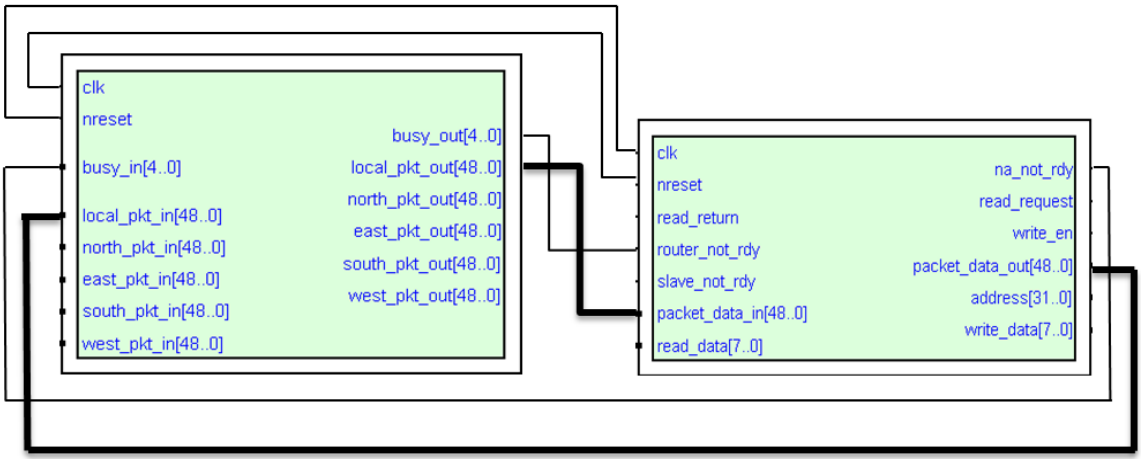
bộ đệm của nó. Tín hiệu này nhằm hỗ trợ cho việc điều khiển luồng (flow control) trên mạng.

Trong cơ chế ready to receive based, router sẽ gửi một tín hiệu ready to receive (RTR) đến RNI bất cứ khi nào nó có không gian bộ đệm trống. Đối với phương pháp này thì yêu cầu về thiết kế giao diện giao tiếp đơn giản hơn. Đề tài sử dụng phương pháp này để thực hiện. Router sẽ gửi một tín hiệu 2 bit router_not_rdy đến cho RNI trong trường hợp router không còn nơi lưu trữ flit.

Độ rộng bus dữ liệu từ RNI đến router và từ router đến RNI là 24 bit và độ rộng đối với các tín hiệu bắt tay là 1 bit bao gồm: busy, router_not_rdy. Việc giao tiếp giữa RNI và router sẽ là song công. Điều này có nghĩa là RNI có thể gửi và nhận gói tin đến router tại cùng thời điểm.



Hình 3.15: Giao diện giao tiếp giữa master RNI với router



Hình 3.16: Giao diện giao tiếp giữa slave RNI với router

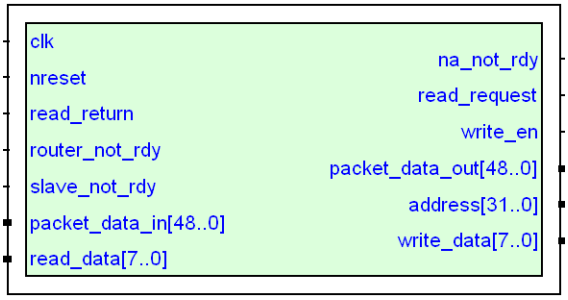
3.2.2.5 Cơ chế chuyển thông tin từ bus dữ liệu sang dạng gói tin và ngược lại

Để chuyển dữ liệu từ resource nguồn đến resource đích mà không để thất lạc dữ liệu, một số thông tin điều khiển cần được thêm vào trong gói tin. Đây là quá trình chuyển từ dữ liệu được gửi ra từ resource thành dạng các gói tin hay còn gọi là flitization. Tương tự khi một gói tin được đã đến được resource đích, thông tin điều khiển mở rộng mà được thêm vào gói tin trước đó sẽ được loại bỏ để lấy phần dữ liệu thực sự. Sau đó dữ liệu từ tất cả các flit được tổng hợp và chuyển đến phần resource. Quá trình này còn được gọi là deflitization.

Trong đề tài, vì mỗi gói tin chỉ chứa một flit duy nhất vì thế quá trình flitization và deflitization coi như được giảm bớt.

3.2.3 Thiết kế cấu trúc slave RNI

RNI được thiết kế sử dụng ngôn ngữ VHDL. Mô hình cấu trúc khối VHDL của thiết kế hệ thống hoàn chỉnh đối với slave RNI được thể hiện trong hình sau



Hình 3.17: Cấu trúc tổng quát bên ngoài của slave RNI

Khối slave RNI nhằm kết nối giữa một thiết bị slave và router trên một node đơn bao gồm: router, network adapter, slave IP. Chức năng của khối này bao gồm:

- Xây dựng một gói tin (1 flit) dựa trên dữ liệu từ slave IP.
- Giải mã gói tin đến từ router và chuyển thành dữ liệu để chuyển đến slave IP.
- Thêm thông tin định tuyến x-y

Bảng 3.1: Ý nghĩa của các chân tín hiệu của slave RNI

Chiều	Chân/bus	Đối tượng kết nối	Mục đích
Ngõ vào	**clk**	Nguồn xung nhịp 50Mhz của hệ thống.	Cung cấp xung nhịp hoạt động cho khối slave RNI
	nreset	Nút nhấn vật lý reset trên hệ thống	reset toàn bộ hệ thống
	read_return	Khối slave IP	Cho slave RNI biết các khối slave IP có trả về dữ liệu yêu cầu không.
	router_not_rdy	Khối Router	Cho slave RNI biết rằng router không sẵn sàng để nhận dữ liệu từ slave RNI gửi đến.
	slave_not_rdy	Khối Slave IP	Cho slave RNI biết rằng slave IP không sẵn sàng để nhận dữ liệu từ slave RNI gửi đến.
	packet_data_in[48:0]	Khối Router	Nhận flit dữ liệu 49 bit từ router gửi đến slave RNI và đích đến cuối là slave IP.
	read_data[7:0]	Khối Slave IP	Nhận một khung dữ liệu 8 bit được trả về từ một slave IP tương ứng với một read_request.
Ngõ ra	**na_not_rdy**	Local_busy_in	Thông báo rằng slave RNI không sẵn sàng để nhận dữ liệu vì slave IP không sẵn sàng
	read_request	Khối Slave IP	Đưa ra tín hiệu yêu cầu đọc dữ liệu từ slave IP.
	write_en	Khối Slave IP	Điều khiển slave IP cho phép slave IP (ngoại trừ switch) ghi dữ liệu nhận được từ router gửi về.
	packet_data_out[48:0]	Khối Router	Chuyển một flit dữ liệu từ slave IP đến router và truyền trên mạng.
	Address[31:0]	Khối Slave IP	Nạp địa chỉ cho dữ liệu ghi tương ứng lên slave IP.
	write_data[7:0]	Khối Slave IP	Nạp một byte dữ liệu nhận được từ router trong flit 49 bit đến slave IP

3.2.3.1 Cơ chế xử lý x-y counter của RNI

Trong gói tin mà slave RNI nhận và gửi đều chứa thông tin về node ID đích. RNI sẽ quyết định giá trị cho các bộ đếm x-y counter dựa trên giá trị node ID của chính nó và node ID đích mà gói tin muốn gửi đến để hỗ trợ cho router trong quá trình định tuyến gói tin trên mạng. Thông tin do RNI xử lý gồm các bit từ 18 đến 23 trong nội dung gói tin gửi đi có dạng như sau

48	47-46	45	44-43
y-flag	**y-counter**	**x-flag**	**x-counter**

Trong đó:

Bit y-flag được đặt là 1 hoặc 0 dựa vào vị trí y trên lưới 4x4 của node. Đây là 2 bit cao trong 4 bit mà RNI lấy từ thông tin node mà RNI gắn vào cũng như node mà RNI muốn gửi gói tin đến

- 1 nếu vị trí node đích < hơn vị trí của node mà RNI gắn kết
- 0 nếu vị trí node đích > hơn vị trí của node mà RNI gắn kết

x-flag cũng được đặt giá trị 1 hoặc 0 dựa vào vị trí trên lưới 2 chiều 4x4. Cơ chế tương tự như đối với y-flag.

y-counter là một giá trị 2 bit và thu được nhờ cơ chế xử lý sau. RNI sẽ xem xét 2 bit cao của node ID đích trong gói tin trừ đi giá trị node ID của chính RNI.

- Nếu kết quả là một số âm, thì giá trị này sẽ được đặt vào trong phần y-counter và trước y-counter sẽ được đặt giá trị 1.
- Nếu giá trị nhận được là dương thì phần trước y-counter sẽ được đặt giá trị 0.

Quá trình xử lý tương tự được áp dụng đối với x-counter của gói tin. Nhưng giá trị sử dụng để thực hiện phép trừ là 2 bit thấp trong phần node ID đích và node ID của chính RNI.

3.2.3.2 Các trường hợp xử lý của slave RNI

Slave RNI được gắn với slave IP là switch và memory nên gói tin mà slave RNI nhận về chỉ là read_request tương ứng khi master IP yêu cầu đọc dữ liệu từ slave IP và write tương ứng khi master IP muốn ghi dữ liệu lên slave IP. Khi nhận gói tin read request tức bit 18 trong gói tin nhận được nếu bit này là 1 thì:

- RNI nhận 3 bit địa chỉ bộ nhớ từ bit thứ 9 đến 11 của gói tin.

- RNI tác động read_request lên mức 1 để yêu cầu đọc dữ liệu từ slave IP.
- RNI tác động chân tín hiệu write và trên write_data(7:0) đều ở mức 0.
- RNI nhận 4 bit giá trị node ID nguồn từ 4 bit từ bit thứ 5-8 trong gói tin nhận.

Khi nhận gói tin write tức bit thứ 17 trong gói tin nhận được là 1 thì

- RNI tác động chân write mức 1,
- RNI gán giá trị cho bus write_data[7:0] bằng 8 bit dữ liệu từ bit thứ 1 đến 8 trong gói tin nhận được.

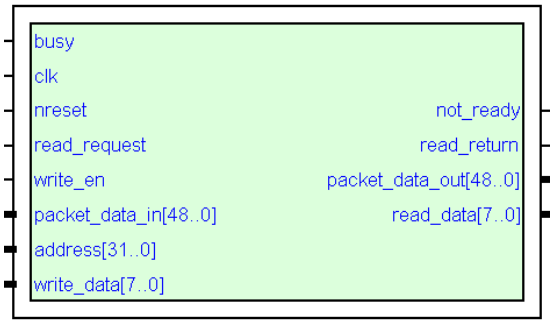
Khi RNI nhận tín hiệu read_return từ slave IP thì RNI sẽ thực hiện đóng gói gói tin và chuyển đến router với cơ chế sau:

- Thiết lập x-y counter
- Đặt bit thứ 16 trong gói tin lên 1 để chỉ ra rằng đây là gói tin read-return.
- Gán dữ liệu 8 bit nhận được từ slave ip vào các bit 4 đến 11 trong gói tin
- Gán giá trị cho node id đích chính là giá trị node id nguồn từ gói tin read_request

3.2.4 Thiết kế cấu trúc master RNI

Khối master RNI nhằm kết nối giữa một thiết bị master IP và router trên một node đơn bao gồm: router, network adapter, master IP. Chức năng khối này gồm:

- Xây dựng một gói tin (1 flit) dựa trên dữ liệu từ master IP.
- Giải mã gói tin từ router và chuyển thành dữ liệu để chuyển đến master IP.
- Thêm thông tin định tuyến x-y



Hình 3.18: Cấu trúc tổng quát bên ngoài của master RNI

Bảng 3.2: Ý nghĩa của các chân tín hiệu của master RNI

Chiều	Chân/bus	Đối tượng kết nối	Mục đích
Ngõ vào	**clk**	Nguồn xung nhịp 50Mhz của hệ thống.	Cung cấp xung nhịp hoạt động cho khối network adapter
	Busy		
	Nreset	Nút nhấn vật lý reset trên hệ thống	reset toàn bộ hệ thống
	Read_request	Khối Master IP	Cho master RNI biết master IP muốn đọc dữ liệu từ slave IP
	Write_en	Khối Master IP	Chân này bị loại bỏ
	Packet_data_in[48:0]	Khối Router	Nhận gói tin từ router gửi đến
	Address[31:0]	Khối Master IP	Nhận 32 bit địa chỉ mà master IP muốn đọc từ slave IP
	Write_data[7:0]	Khối Master IP	Chân này bị loại bỏ
Ngõ ra	**Not_ready**	Khối Router	
	Read_return	Khối Master IP	Chỉ ra rằng master RNI nhận được một gói tin read_return từ router
	Packet_data_out[48:0]	Khối Router	Chuyển một gói tin từ master IP đến router
	Read_data[7:0]	Khối Master IP	Chuyển 1byte mã hoá của gói tin mà master RNI nhận được từ gói tin read_return đến cho Master IP

Cơ chế xử lý đối với phần x-y counter trên master RNI tương tự như slave RNI. Chế độ hoạt động của master RNI là phát đi các gói tin write, read request và nhận về gói tin read return.

Khi master RNI phát đi gói tin write từ yêu cầu write của master ID tức tín hiệu write_en là 1 thì:

- RNI nạp 6 bit giá trị cho bộ x-y counter của gói tin write
- RNI đặt bit 40, 42 là 0 và bit 41 là 1 để nhằm đánh dấu đây là gói tin write
- RNI gán địa chỉ nhận từ master IP đến các bit 8 đến 39 của gói tin write.
- RNI gán dữ liệu nhận được từ master IP đến bit 0 đến 7 của gói tin write.

Khi master RNI phát đi gói tin read_request từ yêu cầu read của master ID tức tín hiệu read_request là 1 thì:

- RNI nạp 6 bit giá trị cho bộ x-y counter của gói tin write
- RNI đặt bit 40, 41 là 0 và bit 42 là 1 để nhằm đánh dấu đây là gói tin read_request
- RNI gán địa chỉ nhận từ master IP đến các bit 8 đến 39 của gói tin write
- RNI gán địa chỉ node ID nhận gói tin read_request.
- RNI đặt các bit 0 đến 3 là 0

Khi master RNI nhận gói tin read_return được trả về từ slave IP tức master RNI kiểm tra bit 40 của gói tin nhận về là 1 thì:

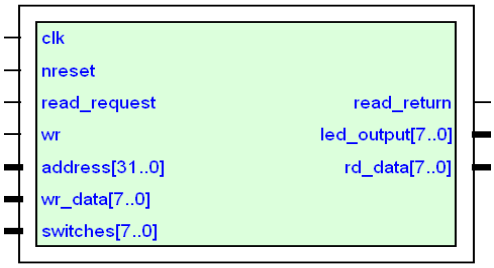
- RNI gán giá trị nhận được cho read_data với giá trị bit từ 28 đến 35 của gói tin read_return
- RNI đặt tín hiệu read_return mức 1.

3.3 Thiết kế tài nguyên mạng

3.3.1 Khối switch ngõ vào

Chức năng của khối này là nhận thông tin ngõ vào từ người dùng vì thế bất kỳ hoạt động ghi dữ liệu lên khối switch sẽ bị loại bỏ. Thông tin nhận được sẽ quyết định hành vi của một số thành phần khác trên mạng.

Khi khối switch nhận được yêu cầu đọc dữ liệu từ gói tin read request sẽ phản hồi bằng một gói tin read return chứa thông tin trạng thái của các switch tại thời điểm hiện hành.



Hình 3.19: Cấu trúc tổng quát bên ngoài của switch

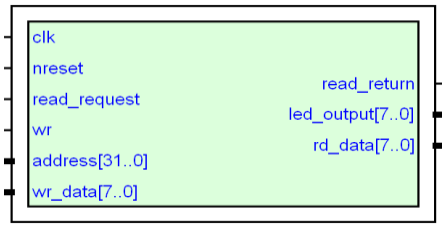
Với việc thay đổi trạng thái các switch sẽ cho phép người dùng thay đổi hành vi của mạng, cụ thể là việc giao tiếp giữa node với các node khác nhau. Sự thay đổi này được minh hoạ trực quan dựa vào giao diện Matlab trên máy tính ở phần sau.

Bảng 3.3: Ý nghĩa của các chân tín hiệu của switch

Chiều	Chân/bus	Đối tượng kết nối	Mục đích
Ngõ vào	**clk**	Nguồn xung nhịp 50Mhz của hệ thống.	Cung cấp xung nhịp hoạt động cho khối network adapter
	nreset	Nút nhấn vật lý reset trên hệ thống	reset toàn bộ hệ thống
	Read_request	Khối slave RNI	Nhận tín hiệu request yêu cầu đọc trạng thái của switch.
	wr	Khối slave RNI	Không sử dụng
	Address[31:0]	Khối slave RNI	Không sử dụng
	Wr_data[7:0]	Khối slave RNI	Không sử dụng
	Switch[7:0]	Switch vật lý trên board DE2	Nhận trạng thái của các switch tại thời điểm hiện tại
Ngõ ra	**Read_return**	Khối slave RNI	Trả về tín hiệu chấp nhận yêu cầu đọc trạng thái từ slave RNI
	Led_out[7:0]	Các led vật lý trên board DE2	Hiển thị trạng thái của các switch lên các led đơn.
	Rd_data[7:0]	Khối slave RNI	Trả về 1byte dữ liệu chứa trạng thái của switch.

3.3.2 Khối bộ nhớ (dummy memory)

Chức năng của khối này là minh hoạ hoạt động của một thiết bị slave trên hệ thống mạng. Thiết bị là một bộ nhớ với 8 thanh ghi độ rộng 8 bit có khả năng ghi và đọc dữ liệu trên 8 bit. Để đánh địa chỉ của 8 thanh ghi này cần 3 bit để mã hoá địa chỉ. Các bit mã hoá được đặt từ bit 25 đến 27 trong không gian 32 bit địa chỉ.



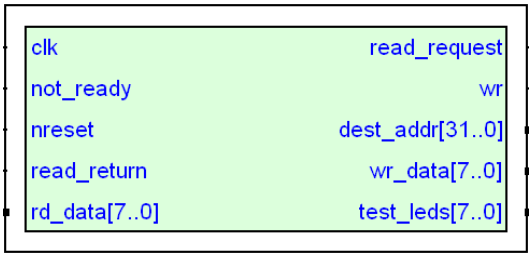
Hình 3.20: Cấu trúc tổng quát bên ngoài của dummy memory

Bảng 3.4: Ý nghĩa của các chân tín hiệu của dummy memory

Chiều	Chân/bus	Đối tượng kết nối	Mục đích
Ngõ vào	**clk**	Nguồn xung nhịp 50Mhz của hệ thống.	Cung cấp xung nhịp hoạt động cho khối RNI
	nreset	Nút nhấn vật lý reset trên hệ thống	reset toàn bộ hệ thống
	Read_request	Khối slave RNI	Nhận tín hiệu yêu cầu đọc bộ nhớ từ một thiết bị master IP.
	wr		Nhận tín hiệu cho phép ghi dữ liệu từ slave RNI
	Address[31:0]		Nhận địa chỉ mà master IP muốn ghi dữ liệu 8 bit
	Wr_data[7:0]		Nhận dữ liệu 8 bit để ghi lên địa chỉ tương ứng
Ngõ ra	**Read_return**	Khối slave RNI	Dummy memory chấp nhận yêu cầu đọc dữ liệu
	Led_out[7:0]	Khối led vật lý	dự trữ
	Rd_data[7:0]	Khối slave RNI	Trả về 1byte dữ liệu đọc từ bộ nhớ.

3.3.3 Khối xử lý (dummy processor)

Chức năng của khối này là minh hoạ hoạt động của một thiết bị master trên hệ thống mạng. Thiết bị sẽ phát các tín hiệu read_request để yêu cầu đọc dữ liệu từ một slave cũng như write để ghi dữ liệu đến slave.



Hình 3.21: Cấu trúc tổng quát bên ngoài của dummy processor

Bảng 3.5: Ý nghĩa của các chân tín hiệu của dummy processor

Chiều	Chân/bus	Đối tượng kết nối	Mục đích
Ngõ vào	**clk**	Nguồn xung nhịp 50Mhz của hệ thống.	Cung cấp xung nhịp hoạt động cho khối RNI
	nreset	Nút nhấn vật lý reset trên hệ thống	reset toàn bộ hệ thống
	Not_ready	Khối master RNI	Cho master IP biết không gửi bất cứ tín hiệu yêu cầu đọc hoặc ghi khi not_ready bật lên 1
	Read_return		Cho master IP biết có gói tin chứa dữ liệu trả về tương ứng với yêu cầu đọc.
	Rd_data[7:0]		Chứa 1byte dữ liệu đọc được
Ngõ ra	**Read_request**	Khối master RNI	Phát tín hiệu yêu cầu đọc dữ liệu của slave IP
	wr		Phát tín hiệu yêu cầu ghi dữ liệu lên slave IP
	Dest_addr[31:0]		Gán địa chỉ cần ghi dữ liệu
	Wr_data[7:0]		Dữ liệu 8bit ghi lên bộ nhớ
	Test_leds[7:0]	Khối led vật lý	Dự trữ

3.3.4 Khối traffic counter và giao tiếp uart

Bộ traffic counter nhằm thống kê tình trạng của mạng cho biết kênh truyền vật lý nào đang được sử dụng với lưu lượng lớn. Với thiết kế mô hình mạng dạng lưới 2 chiều, tại mỗi hướng có 2 kênh truyền dẫn vào/ra. Kết nối giữa các router với nhau gồm 48 kênh truyền vật lý độ rộng 49 bit. Traffic counter gồm 128 thanh ghi với độ rộng 8 bit được thêm vào thiết kế phần cứng. Trong đó:

- 48 thanh ghi tương ứng với các dây dẫn kết nối giữa các router. Việc thiết kế số lượng thanh ghi này chỉ đúng trong trường hợp kích thước mạng là 4x4 tức 16 node mạng.
  - Hướng North gồm 12 thanh ghi ở các vị trí 0,2,4,6,8,10,12,14,16,18,20,22
  - Hướng South gồm 12 thanh ghi ở các vị trí 1,3,5,7,9,11,13,15,17,19,21,23
  - Hướng East gồm 12 thanh ghi: 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46.
  - Hướng West gồm 12 thanh ghi: 25, 27, 29, 31, 33, 35, 37, 39, 41, 43,45, 47.

- 80 thanh ghi còn lại tương ứng với các tín hiệu busy trên các router. Trên mỗi router số lượng kết nối đến 5 hướng với tổng số lượng là 16 router.

Trong mỗi xung nhịp, mỗi thanh ghi trong mỗi tập 128 thanh ghi sẽ tăng lên 1 nếu như các kênh truyền vật lý tương ứng đang được sử dụng hoặc nếu tín hiệu busy trên mỗi hướng của router được đặt. Sau mỗi 500 chu kỳ xung nhịp thì các xoá một tập bộ đếm 128 bit và khởi động tập 128 bit thay thế. Vào cùng thời điểm, tiến trình khác sẽ xoay vòng giữa các thanh ghi và chuyển giá trị tổng từ các tập thanh ghi lưu trữ giá trị không tăng đến khối UART.

Khối UART hoạt động với chức năng kết nối với giao diện matlab nhằm thể hiện trực quan tình trạng mạng. Cấu trúc khối này gồm bộ đệm dữ liệu chứa 128byte lưu 128 thanh ghi tốc độ baud trong giao tiếp là 115200bps. Và chỉ có chức năng truyền dữ liệu lên máy tính. Việc xây dựng khối UART chỉ nhằm cho mục đích hỗ trợ quan sát trạng thái hoạt động mạng trên máy tính, khi thực hiện hệ thống thực tế thì không cần thiết sự tồn tại của khối này. Như vậy, cấu trúc chi tiết bên trong của khối UART sẽ không được trình bày trong đề tài.

CHƯƠNG 4
ĐÁNH GIÁ THIẾT KẾ

Chương này sẽ trình bày về mô hình khảo sát tình trạng hoạt động khác nhau của mạng. Mô hình được thực hiện thông qua giao diện trực quan hoạt động trên máy tính với sự hỗ trợ của phần mềm Matlab dựa trên việc giao tiếp truyền dữ liệu nối tiếp bất đồng bộ giữa mô hình FPGA với máy tính. Bên cạnh đó, việc đánh giá thiết kế NoC về mặt chi phí sử dụng tài nguyên trên phần cứng FPGA và tần số hoạt động tối đa dựa kết quả công cụ Quartus trên nền hệ điều hành Windows. Ngoài ra, trong chương này cũng phân tích về các bản báo cáo tổng hợp về mặt diện tích, công suất và tần số của thiết kế dựa trên công cụ Design Compiler trên nền hệ điều hành Linux kết hợp với thư viện kỹ thuật saed90nm.

4.2 Đánh giá hoạt động thực nghiệm của mô hình

Việc đánh giá hoạt động thực tế của mạng với các trường hợp tranh chấp đường truyền vật lý sẽ được thể hiện trực quan thông qua một sơ đồ mạng được vẽ bằng công cụ Matlab. Việc cập nhật trạng thái của sơ đồ được thực hiện với đặc tính thời gian thực. Chương trình được xây dựng dựa trên Matlab có chức năng đọc dữ liệu từ hệ thống xây dựng trên FPGA thông qua cổng truyền thông nối tiếp bất đồng bộ và cập nhật trạng thái của mạng bằng việc thay đổi màu sắc của các hình vẽ tương ứng đối với các kênh truyền dẫn và các bộ đệm ngõ vào trên router.

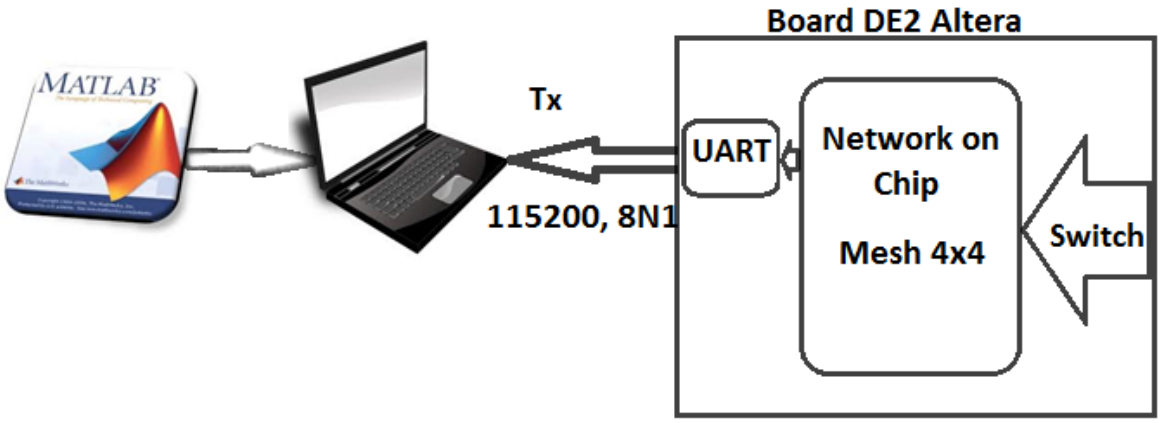
Bảng 4. 1 Phân bố màu chỉ thị đối với kênh dẫn

Màu	Mô tả trạng thái
Đỏ	Kênh dẫn có luồng dữ liệu lớn: chu kỳ gửi dữ liệu nhỏ (khoảng 4 chu kỳ), có nhiều gói tin từ các cặp node nguồn - đích khác nhau đi qua.
Cam	Kênh dẫn có luồng dữ liệu đi qua ít hơn: chu kỳ gửi dữ liệu lớn (khoảng 10 chu kỳ), có nhiều gói tin từ các cặp node nguồn - đích khác nhau đi qua.

Vàng	Kênh dẫn có luồng dữ liệu đi qua ít: chu kỳ gửi dữ liệu lớn (khoảng 10 chu kỳ), không có nhiều gói tin từ các cặp node nguồn - đích khác nhau đi qua.
Trắng	Không có truyền thông tin trên các kênh dẫn này

Phương pháp phân bố màu tương tự cũng được sử dụng đối với các tín hiệu busy để thể hiện trạng thái của các bộ đệm ngõ vào trên các router. Một hình chữ nhật bên ngoài các router được minh hoạ trên hình vẽ để chỉ ra các bộ đệm ngõ vào. Các hình chữ nhật này sẽ thay đổi màu sắc khi các gói tin ngõ vào được đi đến từ các chiều mà không thể xử lý đến ngõ ra mong muốn bởi các kênh dẫn đó đang trong trạng thái phục vụ hoặc một ngõ vào với một độ ưu tiên cao hơn muốn định tuyến đến cùng một kênh dẫn ngõ ra.

Dữ liệu được truyền một chiều từ khối UART trên board thông qua cổng RS232 đến máy tính. Tốc độ truyền dẫn dữ liệu giữa mô hình NoC trên board và máy tính là 115200 bps.



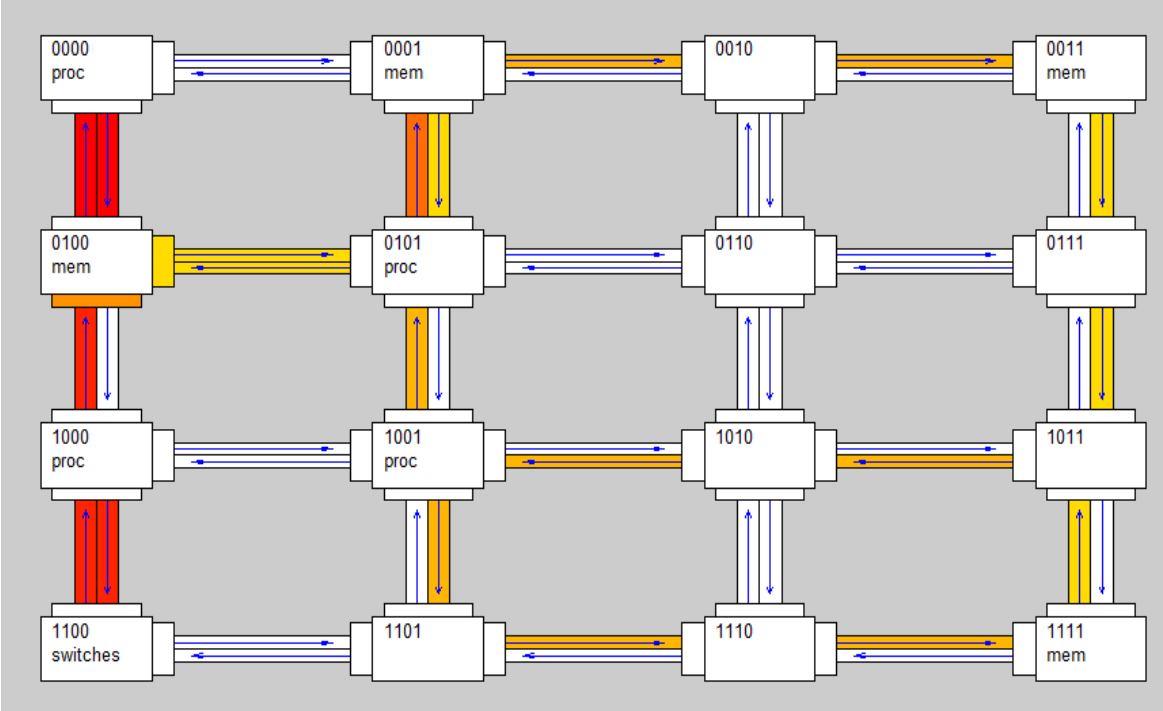
Hình 4. 1 Mô tả kết nối giữa board DE2 và máy tính

Bảng 4.2 Mô tả hành vi một số resource trên mạng

Trường hợp	Node nguồn	Node đích	Tần suất	Hoạt động
1	processor 0000	memory 0100	4 chu kỳ xung	Ghi và yêu cầu đọc
2	processor 0101	memory 0100	10 chu kỳ xung	Yêu cầu đọc
3	processor 0101	memory 0001	10 chu kỳ xung	Ghi và yêu cầu đọc
4	processor 1001	memory 0011	10 chu kỳ xung	Yêu cầu đọc và ghi lên 1111
5	processor 1001	memory 1111	10 chu kỳ xung	Yêu cầu đọc và ghi lên 0011
6	processor 1000	Switch 1100	2 chu kỳ xung	Yêu cầu đọc
7	processor 1000	Node (Switch)	4 chu kỳ xung	Ghi

Trạng thái hoạt động của mạng sẽ được minh hoạ qua các luồng dữ liệu thay đổi liên tục trên mạng. Luồng dữ liệu này được tạo ra bởi các yêu cầu ghi và đọc dữ liệu giữa các khối IP. Như đã trình bày ở phần trước, các khối IP là các dummy processor và dummy memory, i/o để tạo luồng dữ liệu thay đổi liên tục. Ngoài ra, trạng thái của Switch sẽ quyết định hành vi của dummy processor tại node 1000 từ đó luồng dữ liệu sẽ liên tục thay đổi, minh hoạ một số trường hợp tranh chấp trên mạng sẽ được trình bày dưới đây.

Trường hợp switch được cài đặt ở trạng thái 0000 tức dummy proc ở node 1000 giao tiếp với dummy proc ở node 0000. Bộ đệm ở hướng south của node 0100 nhấp nháy báo tình trạng busy. Dummy proc tại node 0000 liên tục cứ 4 chu kỳ xung sẽ gửi gói tin write và read request đến node 0100 chứa module bộ nhớ. Node 0100 chứa module bộ nhớ sẽ gửi trả lại gói tin read return. Vì thế luồng dữ liệu từ ngõ ra và ngõ vào của node 0000 tại cổng hướng south đến ngõ vào và ra của node 0100 tại cổng hướng north có lưu lượng cao nên màu hiển thị cho vùng này là ĐỎ



Hình 4.2: Tình trạng mạng khi switch ở trạng thái 0000

Dummy proc tại node 0101 cứ mỗi 10 chu kỳ xung sẽ gửi:

- Gói tin read request đến node 0100 chứa module bộ nhớ và write đến node 0001 chứa module bộ nhớ.
- Gói tin read request đến node 0001 và write đến 0001 chứa module bộ nhớ.

Dummy proc tại node 0101 nằm cùng vị trí trên trục Ox với mem tại node 0100 nên gói tin chỉ cần chuyền theo hướng Oy. Tần suất phát và nhận gói tin tại cổng ngõ ra/vào hướng west của node 0101 và tại cổng ngõ ra/vào hướng East của node 0100 là 10 chu kỳ xung thấp hơn so với 4 chu kỳ xung clock của trường hợp giao tiếp giữa node 0000 và node 0100 nên màu chỉ thị là VÀNG.

Tương tự cơ chế như trên, dummy pro tại node 0101 cùng vị trí ở trục Oy so với 0001 nên gói tin chỉ cần di chuyển theo hướng trục Ox. Luồng dữ liệu giao tiếp giữa node 0001 và node 0101 có màu chỉ thị là VÀNG song hướng dữ liệu ngõ ra cổng north từ node 0101 đến cổng south của node 0001 còn thêm các gói tin write nên luồng dữ liệu này có màu chỉ thị là ĐỎ.

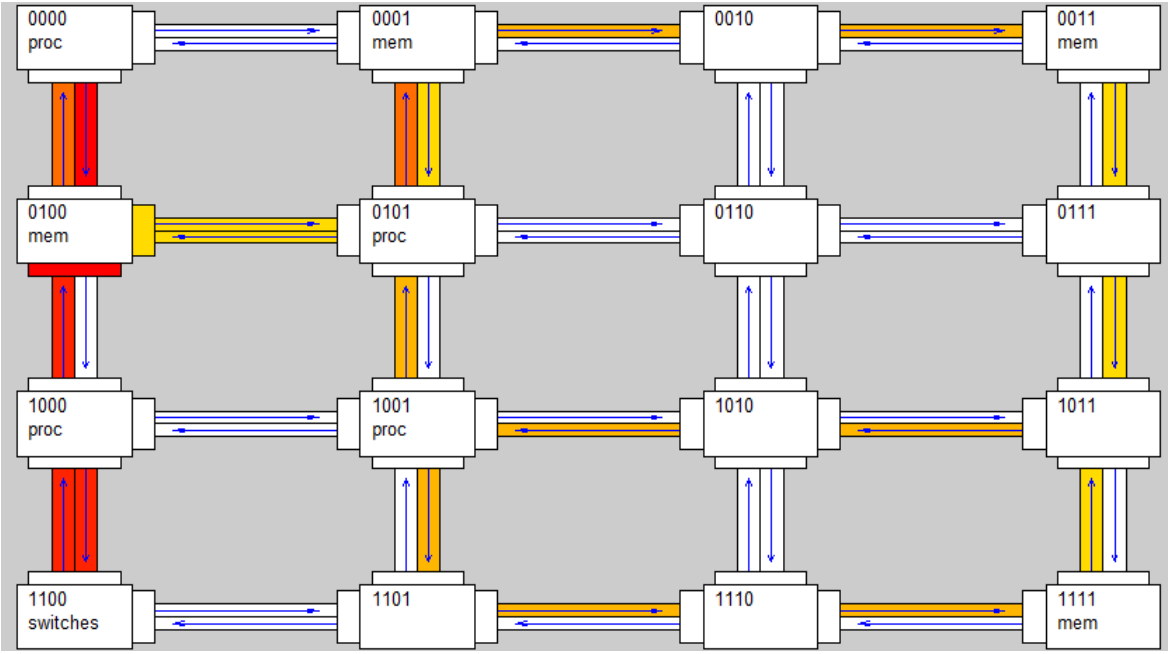
Dummy proc tại node 1001 cứ mỗi 10 chu kỳ xung sẽ gửi một gói tin read request đến node 0011 chứa module bộ nhớ và gửi gói tin write đến node 1111 chứa

module bộ nhớ. Gói tin write được gửi từ master để ghi dữ liệu đến slave. Luồng đi của gói tin read request, read return và write là khác nhau. Với gói tin read request có hướng đi từ node 1001 – 0101 – 0001 – 0010 – 0011. Gói tin read return có hướng đi từ node 0011 – 0111 – 1011 – 1010 – 1001. Gói tin write có hướng đi từ node 1001 – 1101 – 1110 – 1111. Một gói tin read request đến node 1111 chứa module bộ nhớ và write ngược lại đến node 0011.

Tương tự luồng đi dữ liệu của các gói tin read request, read return, write đều theo giao thức định hướng đi theo hướng Ox trước rồi sau đó theo Oy. Những hướng liên kết giữa các node có nhiều gói tin di chuyển thì có màu chỉ thị là màu CAM thay vì VÀNG.

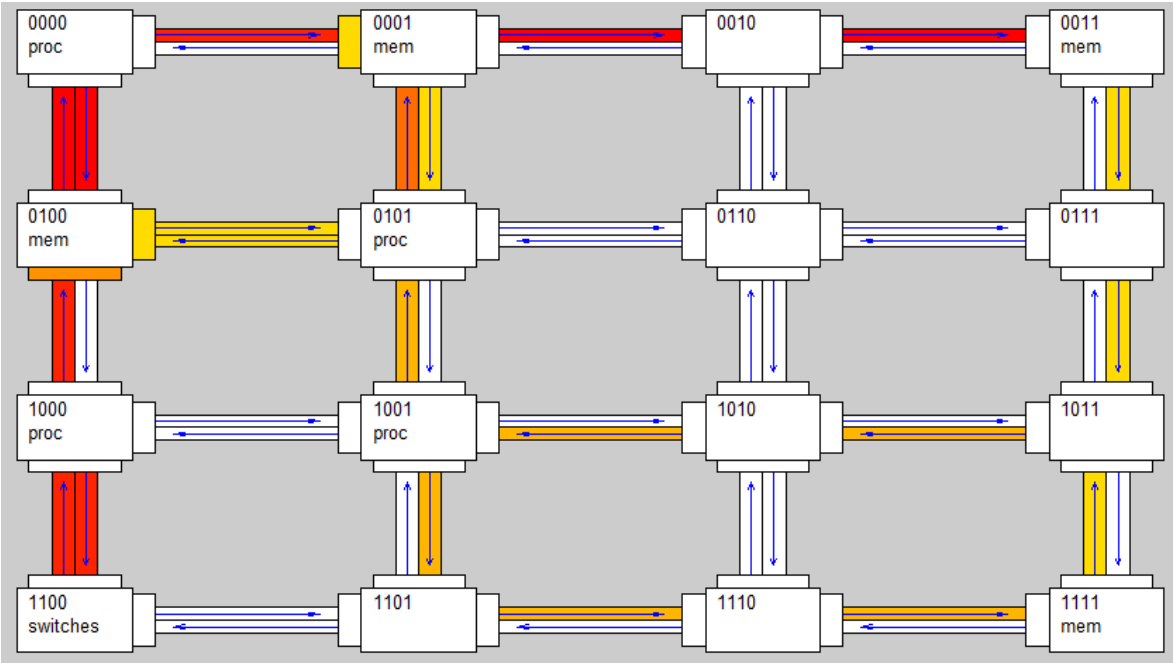
Phụ thuộc vào vị trí thay đổi switch tương ứng với trạng thái [3:0] mà dummy proc ở node 1000 sẽ thay đổi địa chỉ đích gói tin Write tương ứng trong 2 chu kỳ xung. Như vậy, đây là luồng dữ liệu có tần suất trao đổi dữ liệu lớn.

Trường hợp khi SW chuyển từ 0000 thành 0100. Trong một số xung nhịp thì có xung đột xảy ra giữa các gói tin hướng north, east, south. Tất cả đều đang gửi các gói tin đến khối bộ nhớ tại 0100. Các tín hiệu busy chỉ ra thứ tự ưu tiên luôn được đặt với ngõ vào North và do đó nó không bao giờ phải chờ. Tuy nhiên, các gói tin tại ngõ vào hướng East và South đôi khi cần phải chờ trong các bộ đệm ngõ vào vì thứ tự ưu tiên của chúng thấp hơn. Bộ đệm ở hướng South thể hiện màu ĐỎ cho thấy có tranh trấp xảy ra ở hướng này nhiều nhất vì độ ưu tiên hướng này thấp nhất so với 2 hướng còn lại.



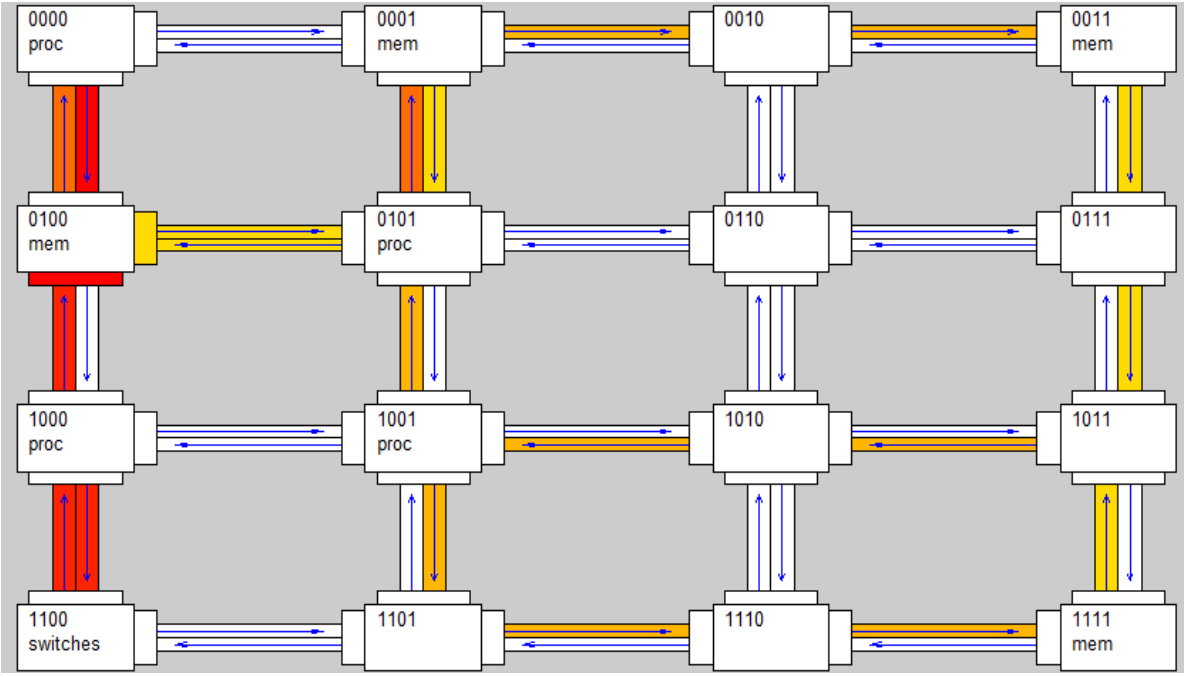
Hình 4.3: Tình trạng mạng khi switch ở trạng thái 0100

Trường hợp khi SW chuyển từ 0000 thành 0011. Switch được cài đặt là 0011, có thể thấy rằng các tín hiệu busy được bật lên tại ngõ vào hướng south của node 0100 (các ngõ vào hướng south và local đều đang ghi đến ngõ ra hướng north) và tại ngõ vào hướng west của node 0001 (ngõ vào hướng west và south đều muốn ghi đến ngõ ra hướng east)



Hình 4.4: Tình trạng mạng khi switch ở trạng thái 0011

Trường hợp khi SW chuyển từ 0000 thành 0010. Tương tự bộ đệm ở west của node 0001 nhấp nháy khi dummy proc ở node 1000 gửi gói tin write đến node 0010. Luồng dữ liệu từ east của node 0001 đến west của node 0010 chuyển từ màu CAM sang ĐỎ cho thấy lưu lượng sử dụng của đường truyền này đã gia tăng.



Hình 4.5: Tình trạng mạng khi switch ở trạng thái 0010

4.2 Đánh giá tài nguyên của thiết kế trên FPGA

Thiết kế được phân tích và tổng hợp từ công cụ Quartus trên nền Windows sử dụng phần cứng FPGA Cyclone-II cho kết quả về phương diện sử dụng tài nguyên phần cứng như bảng sau

Bảng 4. 3: Bảng tổng hợp thiết kế trên nền tảng phần cứng FPGA

Total logic elements	18,383/33,216 (55%)
Total combinational functions	17,158/33,216 (52%)
Dedicated logic registers	11,177/33,216 (34%)
Total registers	11,177
Total pins	36/475 (8%)
Total memory bit	192/483,840 (1%)

Bảng 4.4: Bảng thống kê tài nguyên chi tiết đối với các khối trong thiết kế

Các thành phần trên NoC (Network-on-chip elements)	Tổng số mạch tổ hợp (Total combinational functions)	Tổng số thanh ghi chuyên dụng (Dedicated logic registers)
All node in NoC	**17,158**	**11,177**
Master Network Adapter	12	22
Slave Network Adapter	26	32
Corner Router	283	191
Edge Router	325	240
Internal Router	633	302
Corner Node + dummy processor	401	263
Edge Node + dummy processor	570	313
Internal Node + dummy Processor	873	412
Dummy Processor	62	52
Dummy Memory	51	73

Từ kết quả tổng hợp phần cứng CyCloneII, chúng tôi nhận thấy rằng số lượng thành phần logic được yêu cầu đối với mỗi module trong thiết kế là khác nhau. Điều này đúng với cả số lượng các khối mạch tổ hợp (combinational logic) và số lượng mạch tuần tự (sequential logic). Quartus đã thống kê chi tiết số lượng tài nguyên được sử dụng đối với mỗi khối thành phần trong thiết kế.

Trong đó, router chiếm nhiều thành phần logic nhất và kích thước của chúng phụ thuộc vào vị trí của các router trên mạng. Điều này bởi vì hầu hết các yêu cầu phần cứng để xử lý và quyết định định tuyến gói tin đều được thực hiện trên router. Ngoài ra, mỗi router trên mạng có số lượng hướng kết nối là khác nhau và số lượng bộ đệm tương ứng ở các router này khác nhau.

Như vậy phần cứng trên mỗi router đã được tối ưu hoá và các hướng không có kết nối với thành phần nào thì các kênh dẫn vật lý và các bộ đệm ngõ vào sẽ bị loại bỏ. Đối với thiết kế mạng theo topo lưới 2 chiều thì

- Các router được đặt tại 4 góc tức router 0000, 0011, 1100, 1111 thì có 6 kênh dẫn tương ứng với 3 chiều và tương ứng là các tín hiệu busy.

- Các router ở biên tức các router 0100, 1000, 1101, 1110, 1011, 0111, 0001, 0010 thì có tổng cộng 8 kênh dẫn đối với 4 chiều và các tín hiêu busy.

Đối với các RNI có số lượng thành phần logic ít hơn nhiều lần so với router. Bởi hầu hết việc tính toán phức tạp của việc định tuyến, điều khiển các bộ đệm đã được đặt trên các router trong khi đó các RNI thì chỉ có chức năng xây dựng và giải mã các gói tin và tính toán giá trị cho thành phần định tuyến được thêm vào trong gói tin gửi đi và nhận về.

Trong phương án thiết kế của đề tài, một phần tài nguyên của FPGA cũng được sử dụng bởi khối thống kê lưu lượng (traffic counter) và giao tiếp máy tính (UART). Hai khối này có thể được loại bỏ khi thực hiện hệ thống thực tế. Trong hệ thống thực tế thì kích thước của các lõi IP có thể sử dụng nhiều thành phần logic hơn so với các dummy processor và memory được sử dụng trong thiết kế này.

4.3 Đánh giá thiết kế về mặt công suất, diện tích và thời gian

4.3.1 Công suất tiêu thụ

Công suất tiêu thụ trong một thiết kế có thể được chia thành 2 dạng: công suất tĩnh (static power) và công suất động (dynamic power).

Tổng công suất tiêu thụ = công suất tĩnh + công suất động

Trong đó:

Công suất tĩnh hay công suất đóng gói (leakage power) bị tiêu thụ bởi các cổng vật lý khi ở trạng thái không chuyển mạch. Nguyên nhân do có các dòng điện đi qua các transistor khi transistor ở trạng thái không dẫn. Từ phương diện chức năng của hệ thống thì công suất tĩnh được xem như năng lượng tiêu hao một cách lãng phí, không được sử dụng cho bất cứ tác vụ nào.

Công suất động (dynamic power) được tiêu hao khi mạch ở trạng thái chuyển mạch tức khi thực hiện tác vụ. Công suất động được phân ra: công suất chuyển mạch (switching power) và công suất bên trong (internal power). Công suất chuyển mạch bị tiêu thụ khi nạp và xả dung kháng tải tại ngõ ra của cell. Dung kháng tải bao gồm dung kháng của thành phần dây dẫn (net) và dung kháng của thành phần cổng (gate) mà thành phần dây dẫn kết nối đến. Hầu hết công suất chuyển mạch phụ

thuộc vào hoạt động chuyển mạch (liên quan đến tần số hoạt động) của cell. Nếu càng có nhiều cell thay đổi trạng thái ngõ ra thì công suất chuyển mạch càng tăng. Công suất bên trong được tiêu thụ bên trong một cell đối với việc nạp và xả cả các điện dung bên trong cell.

Tóm lại, công suất tiêu thụ động (dynamic power consumption) phụ thuộc chính vào hoạt động chuyển mạch trong các khối logic bất đồng bộ. Mặt khác, công suất tĩnh tương ứng với diện tích của thiết kế.

Phần đánh giá thiết kế về mặt công suất sẽ được thực hiện với sự hỗ trợ của công cụ Design Compiler của Synopsys từ các báo cáo về công suất. Design Compiler nhận đầu vào là tập tin mô tả phần cứng RTL sử dụng ngôn ngữ mô tả phần cứng VHDL, thư viện cell chuẩn saed90nm với đặc tính công nghệ 90nm, công suất thấp, và các ràng buộc khác nhau về thiết kế. Đối với từng ràng buộc thay đổi thì phương pháp tổng hợp sẽ khác nhau và sẽ ảnh hưởng đến năng lượng tiêu thụ của thiết kế. Ngoài ra, với các thư viện cell khác nhau được sử dụng để tổng hợp thiết kế sẽ cho ra kết quả công suất tiêu thụ khác nhau.

Bảng 4. 5 Đơn vị tính toán về mặt công suất

Đơn vị công suất	
Đơn vị điện áp (Voltage Unit)	1V
Đơn vị dung kháng (Capacitance Unit)	1.000000ff
Đơn vị thời gian (Time Unit)	1ns
Đơn vị công suất động (Dynamic Power Unit)	1uW
Đơn vị công suất tĩnh (Leakage Power Unit)	1pW

Bảng 4. 6 Công suất tiêu thụ của thiết kế đối với từng phương pháp tối ưu

Công suất	**Tối ưu mức cao**	**Thấp**
Cell Internal Power	3.4978 mW	3.7003 mW
Net Switching Power	215.2984 uW	226.0009 uW
Total Dynamic Power	3.7131 mW	3.9263 mW
Cell Leakage Power	28.3452 mW	29.8259 mW

Dựa vào bảng tổng hợp trên, có thể thấy rằng giá trị của tổng công suất động (total dynamic power) bằng tổng của công suất nội bên trong cell (cell internal power) và công suất chuyển mạch trên dây dẫn (net switching power). Vì sự hạn

chế về thời gian thực hiện mà kết quả của đề tài chưa đánh giá công suất động khác nhau đối với mỗi tình trạng hoạt động khác nhau của mạng.

4.3.2 Tần số hoạt động

Trong hệ thống số, tần số hoạt động tối đa có thể được đánh giá ở nhiều khía cạnh khác nhau và phụ thuộc vào các điều kiện hoạt động, các ràng buộc cũng như tài nguyên vật lý để xây dựng thiết kế. Các tài nguyên vật lý này có thể là nền tảng phần cứng FPGA hoặc là thư viện cell chuẩn của nhà máy để sản xuất ASIC.

Yếu tố chính ảnh hưởng đến tần số hoạt động của thiết kế là kiến trúc của thiết kế được xây dựng từ các bản mô tả hành vi thiết kế (hay RTL). Từ bản mô tả hành vi thiết kế này sẽ xây dựng thiết kế thành các mạch tổ hợp, tuần tự, kết hợp kết nối với các flip-flop. Giá trị của tần số sẽ do giá trị setup-time và hold-time của các flip-flop và giá trị trì hoãn của mạch tổ hợp quyết định bên cạnh đó còn có thêm giá trị trì hoãn trong quá trình định tuyến của các kết nối sẽ các mạch.

Ở khía cạnh tài nguyên vật lý, nếu một thiết kế được prototype trên nền tảng FPGA thì tần số hoạt động tối đa sẽ phụ thuộc phần nhiều vào loại FPGA được sử dụng. Ngoài ra, một số yếu tố khác cũng có thể ảnh hưởng đến tần số hoạt động của một thiết kế như nhiệt độ, diện tích…Đối với cấu hình xác định được cài đặt dựa trên công cụ Quartus nhằm đánh giá tần số hoạt động của thiết kế trong đề tài như:

- Điện áp hoạt động: 1.2V
- Nhiệt độ: 85°C

Bảng 4. 7 Thông tin tần số hoạt động tối đa khác nhau của các họ FPGA

Họ FPGA (Altera)	Tần số hoạt động tối đa
Cyclone	90 Mhz
Cyclone II	122 Mhz
Stratix II	150 Mhz

Chúng ta có thể dễ dàng nhận thấy các FPGA khác nhau sẽ cho phép thiết kế có thể hoạt động ở tần số khác nhau như thể hiện ở bảng trên. Nếu sử dụng FPGA họ CycloneII thì tần số hoạt động tối đa cho phép là 122Mhz. Ở tần số này các giá trị setup-time và hold-time được đảm bảo, không có đường kết nối nào bị vi phạm. Nếu ở 125Mhz các giá trị setup-time và hold-time không được đảm bảo và xuất

hiện 32 đường kết nối bị vi phạm. Kết quả đề tài được khảo sát trên phần cứng FPGA họ CycloneII cụ thể là trên board DE2.

4.3.3 Diện tích

Với phương pháp tổng hợp và đánh giá dựa trên công cụ Design Compiler của Synopsys đi kèm sử dụng thư viện saed90nm. Việc ánh xạ từ RTL sang thư viện cụ thể kết hợp với điều kiện môi trường hoạt động khác, các ràng buộc thiết kế khác nhau sẽ ảnh hưởng trực tiếp đến tần số hoạt động của thiết kế. Một trong những yếu tố có thể thay đổi trong quá trình tổng hợp sử dụng Design Compiler là diện tích. Ngoài ra tuỳ vào phương pháp ánh xạ trong quá trình tổng hợp mà sẽ đưa ra sự thay đổi khác nhau về tần số hoạt động.

Nếu một thiết kế trong quá trình tổng hợp yêu cầu tối ưu mức cao về mặt diện tích thiết kế thì sẽ ảnh hưởng đến vấn đề timing của thiết kế, cụ thể là giảm tần số hoạt động tối đa. Bởi tuỳ thuộc vào yêu cầu tối ưu về diện tích mà số lượng các thành phần cell bên trong xây dựng thiết kế sẽ thay đổi. Bảng sau minh hoạ điều này

Bảng 4. 8 Thông tin về diện tích thiết kế với từng phương pháp tổng hợp

Thành phần	Tối ưu mức cao	Thấp
Number of ports:	36	36
Number of nets:	13600	12847
Number of cells:	10340	9730
Number of combinational cells:	8125	7532
Number of sequential cells:	2100	2099
Number of macros:	0	0
Number of buf/inv:	2840	1960
Number of references:	173	163
Combinational area:	645525.503150	695995.083915
Noncombinational area:	459278.435207	459248.022409
Net Interconnect area:	150677.076265	174088.557009
Total cell area:	1104803.938358	1155243.106324
Total area:	1255481.014623	1329331.663333

Diện tích của thiết kế được chia ra thành diện tích đối với cell và diện tích đối với thành phần kết nối. Thông tin về diện tích trong thiết kế NoC có thể được thể hiện bằng cách sử dụng tuỳ chọn báo cáo về diện tích (report_area) khi sử dụng công cụ Design Compiler. Vì số lượng cổng độc lập với công nghệ của thư viện.

Diện tích thiết kế phụ thuộc vào thư viện tổng hợp thiết kế. Ở các thư viện sản xuất các cell được tối ưu vì thế cho kết quả diện tích thiết kế được tối ưu. Ngoài ra, diện tích thiết kế còn phụ thuộc vào phương pháp tổng hợp một thiết kế và các ràng buộc khi tổng hợp thiết kế. Diện tích thiết kế và tần số hoạt động có mối liên hệ. Nếu tối ưu hoá về diện tích thiết kế thì xem như không quan trọng vấn đề tần số hoạt động và ngược lại.

Bảng sau sẽ thể hiện kết quả báo cáo về mặt tần số, công suất và diện tích đối với các trường hợp tổng hợp khác nhau. Trong đó, nếu kết quả về mặt tần số được ghi chú là VIOLATED thì cho biết đã vi phạm và mặt thời gian khi tổng hợp thiết kế với ràng buộc xác định. Một thiết kế thoả mãn về các yêu cầu ràng buộc khi bản báo cáo về mặt thời gian được ghi chú là MET.

Bảng 4. 9 Thông tin về công suất, diện tích và thời gian với phương pháp tối ưu mức cao trong quá trình tổng hợp

Tần số	Công suất/ diện tích	Công suất	Diện tích	Thời gian
Mức cao **76 Mhz**	Tối ưu mức cao về mặt diện tích	Total Dynamic Power= 3.6877 mW (100%) Cell Leakage Power= **28.6877** mW	Combinational area: 462888.340837 Noncombinational area: 459092.272005 Net Interconnect area: 151561.696162 Total cell area: 921980.612843 Total area: **1073542.309004**	data required time 13.29 data arrival time -14.21 slack **(VIOLATED)** -0.91
	Không tối ưu về mặt diện tích	Total Dynamic Power= 3.9263 mW (100%) Cell Leakage Power= **29.8259** mW	Net Interconnect area: 174088.557009 Total cell area: 1155243.106324 Total area: **1329331.663333**	data required time 13.29 data arrival time -13.29 slack **(MET)** 0.00

Bảng 4. 10 Thông tin về công suất, diện tích và thời gian với phương pháp tối ưu mức trung bình trong quá trình tổng hợp

Tần số	Công suất/ diện tích	Công suất	Diện tích	Thời gian
Mức trung bình **75Mhz**	Tối ưu mức cao về mặt diện tích	Total Dynamic Power= 4.0453 mW (100%) Cell Leakage Power= **30.1973** mW	Combinational area: 459592.699212 Noncombinational area: 459124.528008 Net Interconnect area: 150983.224717 Total cell area: 918717.227220 Total area: **1069700.451937**	data required time 13.53 data arrival time -13.53 - Slack **(MET)** 0.00
	Không tối ưu về mặt diện tích	Total Dynamic Power= 3.8421 mW (100%) Cell Leakage Power= **30.9792** mW	Net Interconnect area: 173778.879205 Total cell area: 1171879.829968 Total area: **1345658.709174**	data required time 13.59 data arrival time -13.59 - Slack **(MET)** 0.00

CHƯƠNG 5
KẾT LUẬN

Chương này sẽ trình bày về những kết quả mà đề tài đã đạt được, những hạn chế còn tồn tại trong thiết kế mô hình minh hoạ ứng dụng NoC. Ngoài ra những đề xuất về phương án thiết kế nhằm cải tiến hiệu năng cho mô hình ứng dụng NoC cũng được trình bày trong phần này.

5.1 Kết quả đạt được

Mạng trên chip là một phương pháp xây dựng kết nối trên các SoC hiện đại, đặc biệt là đối với các hệ thống đa lõi. Với một số kết quả nghiên cứu so sánh mô hình truyền thông NoC trên chip so với mô hình bus như ở [12] cho thấy công suất và tần số hoạt động của hệ thống ứng dụng NoC không thể hiện hiệu quả so với hệ thống ứng dụng bus trong kết nối. Song về mặt trì hoãn phân xử (arbitration delay) và khả năng mở rộng hệ thống thì mô hình truyền thông NoC thể hiện được các ưu điểm vượt trội.

Đề tài thực hiện minh hoạ một SoC cơ bản sử dụng nền tảng giao tiếp NoC với phương pháp thực nghiệm (experimental approach). Phương pháp này sử dụng các công cụ mô phỏng, giả lập hoặc nền tảng phần cứng FPGA để đánh giá hoạt động và hiệu năng của thiết kế.

Mô hình này không nhằm để đưa ra một thiết kế chi tiết cho một ứng dụng cụ thể như xử lý ảnh hoặc xử lý tín hiệu mà chỉ nhằm xây dựng mô hình tổng quát minh họa một SoC cơ bản sử dụng nền tảng NoC trong giao tiếp. Kết quả của đề tài đã đạt được một số vấn đề như:

- Xây dựng thành công một mô hình minh hoạ SoC ứng dụng NoC trong việc truyền thông dữ liêu. Mô hình mạng trên chip được thiết kế với các đặc tính như: topo mạng dạng lưới 2 chiều, kích thước mạng 4*4 gồm 16 node, giao

tiếp bằng các đơn vị gói tin (flit), cơ chế điều khiển luồng request-return, cơ chế điều khiển tắc nghẽn với tín hiệu busy giữa các router

- Thực nghiệm thành công mô hình SoC ứng dụng NoC trên nền tảng phần cứng FPGA cụ thể là trên board DE2 Altera, nhằm xác nhận hoạt động thực tế của mô hình mạng NoC lưới 4*4. Bên cạnh đó, việc trực quan hóa hoạt động của mô hình mạng với giao diện Matlab cũng đã hoàn thành.
- Đánh giá thiết kế về tần số hoạt động, diện tích thiết kế, công suất tiêu thụ của thiết kế với công cụ hỗ trợ về thiết kế vi mạch Design Compiler và thư viện saed90nm của Synopsys.

Như vậy, kết quả đề tài có thể làm nền tảng cơ sở để:

- Làm cơ sở nghiên cứu, từ đó cải tiến và phát triển các đặc tính về kiến trúc nhằm tăng hiệu năng cho thiết kế.
- Làm cơ sở trong giảng dạy về lĩnh vực Design for Configurable Architecture trong trường đại học.

5.2 Phần khuyến nghị

5.2.1 Những tồn tại trong đề tài

Bên cạnh các kết quả đã được, mô hình triển khai của đề tài vẫn còn tồn tại nhiều điểm hạn chế như:

- Gói tin chỉ bao gồm 1 flit.
- RNI thực hiện chức năng kết nối các IP với thành phần router của mạng. Chưa triển khai chức năng chuyển đổi dữ liệu từ gói tin thành các flit đối với mô hình gói tin lớn chứa nhiều flit thành phần và ngược lại.
- Chưa tối ưu phương pháp định tuyến, đường đi của gói tin cố định. Vì thế chưa linh hoạt trong việc chia sẻ đường truyền các gói tin trên mạng nhằm nâng cao hiệu năng mạng. Đây là các vấn đề nhằm cân bằng tải (load balancing) trên mạng.
- Chưa triển khai cơ chế nâng cao hiệu quả kênh truyền dẫn vật lý bằng cách sử dụng cơ chế kênh ảo nhằm xử lý trong trường hợp gói tin với nhiều flit đi kèm và tốc độ chuyển giao dữ liệu giữa các IP core lớn.

- Thiết kế một SoC với các lõi IP thực tế chưa được đề cập. Các IP core trong đề tài như dummy proc, dummy mem, uart, switch…chỉ nhằm mục đích minh họa cho hoạt động của SoC.
- Chưa đánh giá hiệu năng mạng với các thông số throughput và latency.
- Mỗi hướng bắt buộc gồm 2 kênh truyền vật lý riêng biệt tương ứng 1 kênh cho chiều ngõ vào và 1 kênh cho chiều ngõ ra của dữ liệu.

5.2.2 Các giải pháp cải tiến thiết kế

Với thiết kế của được lựa chọn trong đề tài, có một số cải tiến cần thực hiện nếu thiết kế được đưa vào các ứng dụng thực tế

5.2.2.1 Giảm số lượng các thanh ghi và các cổng logic

Thiết kế NoC có một số lượng lớn các thanh ghi được sử dụng cho các bộ đệm và hầu hết thì chúng không được sử dụng cùng một thời điểm. Như vậy có nhiều phương pháp để giảm số lượng này mà không cần thay đổi thuật toán định tuyến. Một trong các phương pháp đó là xây dựng bộ đệm tập trung cho các thanh ghi dành cho bất cứ router nào và có các nhãn để chỉ ra router nào hiện đang sử dụng mỗi thanh ghi. Điều này có thể cho phép số lượng thanh ghi sử dụng là ít hơn. Tuy nhiên, cần phải xây dựng một bộ điều khiển linh hoạt để kiểm soát việc sử dụng của bộ đệm tập trung này.

5.2.2.2 Độ rộng dữ liệu

Trong thiết kế sử dụng độ rộng cho kênh truyền dẫn là 49 bit tuy nhiên lượng thông tin hữu dụng của mỗi flit chỉ là 8 bit. Việc chọn số lượng 8 bit trên mỗi flit bởi vì việc kiểm thử trong đề tài có liên quan đến khối uart. Khối này chỉ yêu cầu dữ liệu truyền dẫn là 8 bit. Tuy nhiên, đối với các hệ thống thực tế thì nên có số lượng dữ liệu lớn hơn được truyền dẫn trên mỗi gói tin.

Một lượng dữ liệu lớn trên mỗi flit được truyền đi sẽ làm giảm bớt chi phí phải phân tách nhỏ gói tin và làm giảm các tình trạng tắc nghẽn và xung đột trên mạng. Ngoài ra, việc thiết kế số lượng bit dữ liệu thay đổi linh hoạt phải đối đầu với thách thức là xây dựng một cơ chế linh hoạt hơn so với việc cố định số lượng bit.

5.2.2.3 Bộ đệm trong các RNI

Tất cả các router có 2 cấp bộ đệm trên mỗi ngõ vào nhằm giải quyết vấn đề tranh chấp với các độ ưu tiên khác nhau đối với mỗi kênh truyền. Tuy nhiên, với lựa chọn của thiết kế trong đề tài thì các RNI chưa xây dựng các bộ đệm bên trong. Như vậy cần lưu ý đến khả năng xây dựng các bộ đệm trong mỗi RNI nhằm giải quyết các vấn đề nghẽn dữ liệu trong mạng khi có lượng lớn thông tin được truyền dẫn.

Tóm lại, RNI cần thiết phải xây dựng 2 bộ đệm khác nhau để lưu trữ các gói tin trong suốt quá trình truyền dữ liệu từ node nguồn đến đích. Bộ đệm dữ liệu thứ nhất được xây dựng để lưu trữ tạm thời các gói tin được truyền từ IP core đến router trong khi đó bộ đệm thứ hai được xây dựng để lưu trữ các gói tin được truyền từ router đến IP core.

5.2.2.4 Kích thước mạng thay đổi

Với thiết kế được lựa chọn thì kích thước mạng cố định là 2 chiều mỗi chiều gồm 4 node. Tuy nhiên, đối với một số ứng dụng thực tế có thể yêu cầu linh hoạt nhiều hơn ít hơn hoặc nhiều 16 node mạng. Như vậy cần có cơ chế có thể thay đổi kích thước mạng một cách linh hoạt. Trong một số cấu trúc router khác nhau, yêu cầu này có thể thay đổi kích thước của gói tin với sự thay đổi x/y counter và số node ID

5.2.2.5 Bidirectional router

Phương án thiết kế trong đề tài sử dụng 2 kênh truyền riêng biệt nhằm tách biệt 2 ngõ vào và ra tại mỗi cổng. Tuy nhiên phương án này tạo ra một số vấn đề như gia tăng số lượng dây dẫn, diện tích và công suất của thiết kế. Một giải pháp khắc phục đó là kiến trúc BiNoC được đề xuất cho phép truyền dẫn dữ liệu 2 chiều trên cùng một kênh truyền vật lý nhằm cải thiện hiệu năng truyền.

Đối với thiết kế của mô hình BiNoC thì mỗi kênh truyền dẫn vật lý cho phép tự nó có thể cấu hình một cách linh hoạt chiều truyền dẫn ở cả hai chiều. Điều này làm tăng tính linh hoạt nhằm làm tăng băng thông, giảm độ trễ trong việc phân phát các gói tin trên mạng. Chiều truyền dẫn của mỗi kênh được kiểm soát bởi một giao thức điều khiển chiều kênh dẫn (channel direction control protocol). Giao thức này được sử dụng làm tăng hiệu năng mạng, giải quyết vấn đề deadlock và starvation.

Như đã trình bày ở phần trước, nếu xét về khía cạnh chi phí dây dẫn thì hầu như ngang bằng với chi phí các cổng. Như vậy trong tương lai, chi phí về dây dẫn đáng để xem xét. Do đó, việc sử dụng dây dẫn hay kênh truyền vật lý một cách hiệu quả là một việc rất quan trọng nhằm giảm chi phí của hệ thống

5.2.2.6 Gói tin chứa đa flit

Với các phương án được lựa chọn nhằm xây dựng mô hình NoC của đề tài, thiết kế đơn giản hoá phần cứng nên mỗi gói tin chỉ chứa duy nhất 1 flit. Tuy nhiên, trên thực tế thì gói tin có kích thước khá lớn và vì thế phải có phương pháp để chia gói tin ra thành nhiều flit. Bởi kích thước kênh dẫn không thể có độ rộng bằng kích thước của gói tin. Việc cho phép chia gói tin ra làm nhiều flit sẽ dẫn đến kích thước gói tin có thể lớn hơn và giảm độ rộng kênh dẫn.

Để giải quyết yêu cầu này thì việc tạo nhiều gói tin trên một flit sẽ làm tăng độ phức tạp của thuật toán định tuyến và có thể xảy ra vấn đề nghẽn mạng và giải quyết tranh chấp giữa các gói tin từ các node gửi nhận khác nhau. Ngoài ra RNI cũng phải đảm nhận chức năng xử lý nhiều hơn bởi gói tin có thể được tạo với nhiều xung clok. Thiết kế bộ đệm nhận dữ liệu tại mỗi ngõ vào và ngõ ra của mỗi router cũng là một vấn đề cần lưu ý.

5.2.2.7 Cơ chế Wormhole

Nếu một gói tin gồm nhiều flit thì phương pháp chuyển mạch cần thiết triển khai theo wormhole bởi trong phương pháp này sẽ làm giảm đáng kể lượng bộ nhớ đệm để lưu trữ gói tin tại mỗi router. Việc thiết kế bộ đệm có kích thước nhỏ sẽ làm giảm chi phí và một số vấn đề đã được trình bày ở trên. Tuy nhiên, đối với phương pháp này cần xây dựng một cơ chế quản lý việc sử dụng bộ đệm của các flit. Việc sắp xếp các flit sao cho chính xác với từng gói tin là một thách thức trong thiết kế.

5.2.2.8 Cơ chế kênh ảo

Khi triển khai phương pháp chuyển mạch gói wormhole thì vấn đề gặp phải là deadlock. Việc chia sẻ sử dụng cùng chung một kênh truyền vật lý giữa 2 router là một phương pháp tăng hiệu năng và giảm diện tích thiết kế, bên cạnh đó là giảm bớt tình trạng head-of-line của gói tin.

5.2.2.9 Thuật toán định tuyến

NoC sử dụng phương pháp đơn giản để định tuyến gói tin trên một đường dẫn cụ thể. Tuy nhiên, đường định tuyến của mỗi gói tin có thể thay đổi bởi router dựa trên tình trạng lưu lượng của mỗi kênh dẫn. Một thuật toán định tuyến linh hoạt sẽ giảm thời gian định tuyến của gói tin nhưng ngược lại cũng làm tăng vấn đề phức tạp trong thiết kế router vì router phải xử lý linh hoạt hơn.

5.2.2.10 Độ ưu tiên trong việc định tuyến

Việc sử dụng cơ chế độ ưu tiên cố định làm giảm độ phức tạp của phần định tuyến. Tuy nhiên, thiết kế cần thiết xây dựng cơ chế xử lý độ ưu tiên định tuyến linh hoạt như là độ ưu tiên của ngõ vào thay đổi theo cơ chế round robin. Điều này có thể mang lại lợi ích khi một IP core đưa ra một độ ưu tiên cho mỗi gói tin bởi thế các gói tin quan trọng hơn có thể được định tuyến trên mạng nhanh hơn.

TÀI LIỆU THAM KHẢO

1. Chen S-J, Lan Y-C, Wen-ChungTsai, Hu Y-H. Reconfigurable Networks-on-Chip: Springer; 2011.
2. Jantsch A, Tenhunen H. Networks on Chip: Springer; Kluwer Academic Publishers; 2004.
3. Hill M, Wisconsin Uo, Madison. On-Chip Networks, Synthesis Lectures on Computer Architecture. 2006-2009. p. 141.
4. Karlsruhe G. 2011 22nd IEEE International Symposium on Rapid System Prototyping. IEEE Reliability Society. 2011 (2011 22nd IEEE International Symposium on Rapid System Prototyping):212.
5. MileStojcev M. An Overview of On-Chip Buses. Facta Universitatis 2009.
6. Agarwal A, Iskander C, Shankar R. Survey of Network on Chip (NoC) Architectures & Contributions Journal of Engineering, Computing and Architecture. 2009.
7. Mahmood ZAMA. Design And Prototype Of Resource Network Interfaces For Network On Chip. 2009.
8. Hung-Chih Lai RM, Marios Savvides, and Tsuhan Chen. Communication-Aware Face Detection Using Noc Architecture. 2006 (Apps of NoC) 10.
9. Mello A, Tedesco L, Calazans N, Moraes F. Virtual Channels in Networks on Chip: Implementation and Evaluation on Hermes NoC. 2005:6.
10. Park S. A Veriog-Hdl Implementation Of Virtual Channels In A Network-On-Chip Router [Computer Engineering]: Texas A&M University; 2008.
11. Qian Y, Lu Z, Dou W. Analysis of Worst-case Delay Bounds for Best-effort Communication in Wormhole Networks on Chip. IEEE. 2009:10.
12. S. Kurup1 TM. A comparison of traditional on-chip interconnects with Network-on-Chip architecture. California State University, Long Beach, California, USA 2006 (maybe).
13. Seyyed Amir Asghari HP, Mohammad Khademi, and Pooria Yaghini Amirkabir University of Technology, dér Engineering and Information Technology, Tehran, Iran Designing and Implementation of a Network on Chip Router Based on Handshaking Communication Mechanism. 2009.
14. VilleRantala, TeijoLehtonen, JuhaPlosila. Network on Chip Routing Algorithms. 2006.
15. Xu Y, BoZhaoy, YoutaoZhangz, JunYang. Simple Virtual Channel Allocation for High Throughput and High Frequency On-Chip Routers. IEEE. 2009:11.
16. ZhonghaiLu. Using Wormhole Switching for Networks on Chip: Feasibility Analysis and Microarchitecture Adaptation. 2005.
17. Pham D, Le T. Reduced Clock Cycle Latency Router Architecture In Network-On-Chip (NoC). 2012:6.

18. Thang HV.Nghiên cứu và thực thi NoC trên nền tảng FPGA:BK Đà Nẵng; 2007.
19. Graeme Best MB, Oscar Rahnama, Wojciech Pawlak. Design and implementation of a simple mesh Network-on-Chip 2012.
20. Malviya S. Five Port Router for Network on Chip. 2004:6.
21. Pui R, Pau H. A Configurable Router for Embedded Network-on-Chip Support in Field-Programmable Gate Arrays: Queen's University; 2008.
22. Sridhar Gangadharan SC. Constraining Designs for Synthesis and Timing Analysis Spinger; 2013. 245 p.
23. Altera. Introduction to the Quartus® II Software. Altera: Version 10.0 Altera; 2010.
24. Bhatnagar H. Advanced Asic Chip Synthesis Using Synopsys® Design Compiler™ Physical Compiler™ and PrimeTime®: Kluwer Academic Publishers 2002.
25. Synopsys. Design Compiler Tutorial Using Design Vision. SynopsysVersion B-2008.09, June 2009.
26. Synopsys. Synopsys 90nm Generic Library for Teaching IC Design. Synopsys. (Synopsys 90nm Generic Library):2.

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

www.hcmute.edu.vn