

Thiết kế và mô hình hoá bộ xử lý lô-gic mờ trong điều khiển tần số - điện áp

Phan Hải Phong^{1,2}, Trần Xuân Tú¹

¹)PTN mục tiêu Hệ thống tích hợp thông minh, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội
144 đường Xuân Thủy, Cầu Giấy, Hà Nội, Việt Nam

²)Trường Đại học Khoa học Huế, Đại học Huế – Thành phố Huế, Việt Nam
Email: phongph@husc.edu.vn; tutx@vnu.edu.vn

Tóm tắt—Trong bài báo này, chúng tôi trình bày việc thiết kế một bộ điều khiển sử dụng thuật toán lô-gic mờ có khả năng dự đoán lưu lượng truyền thông của một bộ định tuyến trong mạng trên chip (NoC: Network-on-Chip). Từ đó, bộ điều khiển này sẽ tác động đến tần số và điện áp hoạt động của bộ định tuyến một cách phù hợp nhằm giảm thiểu năng lượng tiêu thụ của bộ định tuyến này theo phương pháp điều khiển tỷ lệ điện áp – tần số động (DVFS: Dynamic Voltage and Frequency Scaling) trong khi vẫn đảm bảo được hiệu năng hoạt động theo yêu cầu của bộ định tuyến. Bộ xử lý lô-gic mờ, phần lõi của thiết kế, đã được mô hình hoá bằng ngôn ngữ mô tả phần cứng VHDL. Hoạt động của bộ xử lý mờ được mô phỏng và kiểm chứng bằng phần mềm mô phỏng ModelSim của hãng Mentor Graphics. Một số kết quả thực thi phần cứng ban đầu của bộ xử lý mờ trên công nghệ FPGA của hãng Xilinx cũng được trình bày trong bài báo.

Keywords— *Low-power; Network on Chip; DVFS; Fuzzy logic.*

I. MỞ ĐẦU

Kể từ khi được đề xuất lần đầu tiên vào năm 1965 bởi giáo sư L. A. Zadeh [1], lý thuyết tập mờ đã được ứng dụng trong rất nhiều lĩnh vực khác nhau. Những lĩnh vực mà lý thuyết lô-gic mờ đã được áp dụng thành công có thể được kể đến như: điều khiển tự động, tiết kiệm điện, xử lý dữ liệu, xử lý tín hiệu... Đặc biệt là trong lĩnh vực thiết kế robot, các loại robot có khả năng hành xử như con người dựa trên lý thuyết lô-gic mờ đã và đang được nghiên cứu phát triển. Với việc áp dụng lý thuyết lô-gic mờ thì cách hành xử của robot có thể được thực hiện thông qua những quyết định “nếu - thì” tương tự như suy nghĩ của con người. Do đó, những robot này có thể có các “suy nghĩ”, hành động giống với con người hơn so với những thế hệ robot trước đây [2]. Trong lĩnh vực y khoa, lý thuyết lô-gic mờ cũng đã và đang được nghiên cứu và áp dụng vào nhiều chuyên ngành khác nhau, đặc biệt là trong lĩnh vực xử lý tín hiệu y sinh. Trong công trình [3], Y. C. Yeh và cộng sự đã giới thiệu

một phương pháp đơn giản và hiệu quả để nghiên cứu về tín hiệu ECG sử dụng lý thuyết lô-gic mờ. Công trình [4] cũng đã giới thiệu một kỹ thuật dựa trên lý thuyết lô-gic mờ để xử lý các ảnh MRI một cách hiệu quả.

Trước đây, các hệ thống lô-gic mờ thường được thực thi bằng phần mềm. Ưu điểm của việc thực thi hệ thống lô-gic mờ bằng phần mềm đó là khả năng triển khai hệ thống nhanh, dễ dàng sửa đổi, thời gian phát triển nhanh và tốn ít chi phí. Tuy nhiên nhược điểm của những hệ thống này đó là tốc độ xử lý và tính toán chậm. Điều này đã làm ảnh hưởng đến việc triển khai các hệ lô-gic mờ trong những hệ thống cần xử lý thời gian thực.

Cùng với việc ngày càng nhiều các ứng dụng của lý thuyết lô-gic mờ được áp dụng vào trong nhiều lĩnh vực khác nhau đã thúc đẩy các nhà nghiên cứu phát triển những hệ thống lô-gic mờ có tốc độ xử lý, tính toán nhanh và hiệu quả hơn. Một trong những hướng phát triển các hệ thống xử lý lô-gic mờ đó là thực thi những các hệ thống này dưới dạng các lõi xử lý bằng phần cứng để cho phép tăng tốc độ xử lý của hệ thống [5]–[7]. Với hướng tiếp cận này, các hệ thống xử lý lô-gic mờ có thể đạt đến hàng triệu phép tính toán/suy luận lô-gic mờ trong một giây (FLIPS: fuzzy logical inference per second) [8]. Trong công trình [9], K. Nakamura và các cộng sự cũng đã xây dựng được một lõi xử lý lô-gic mờ với 12 đầu vào có độ phân giải 16 bit và tốc độ tính toán cũng đạt đến 2.10^5 FLIPS.

Bài báo này tập trung vào việc thiết kế một khối điều khiển tần số - điện áp dành cho bộ định tuyến của mạng trên chip (NoC: Network-on-Chip [10]) nhằm điều chỉnh tần số hoạt động cũng như điện áp của bộ định tuyến mạng một cách phù hợp tùy theo lưu lượng truyền thông qua bộ định tuyến đó. Do đó, ta có thể giảm bớt năng lượng tiêu thụ của từng bộ định tuyến, cũng như năng lượng tiêu thụ chung cho toàn hệ thống, trong khi vẫn đảm bảo hiệu năng hoạt động của hệ thống đáp ứng yêu cầu của ứng dụng. Bộ điều khiển này sẽ được áp dụng thuật toán lô-gic mờ theo mô hình Sugeno [11] để xử lý và được thực thi dưới dạng một lõi cứng mô hình

hoá bằng ngôn ngữ VHDL ở mức chuyển dịch thành ghi (RTL: Register-Transfer Level).

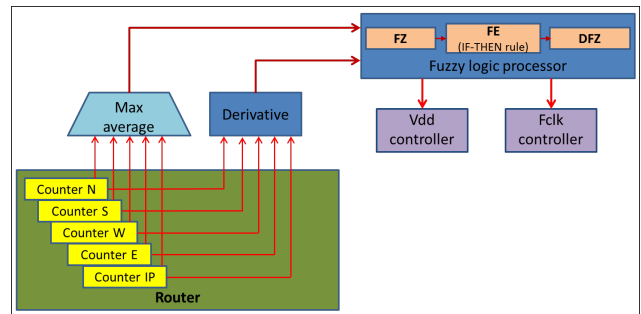
Ngoài phần Mở đầu, cấu trúc của bài báo này được chia thành các phần sau: Phần II trình bày mô hình đề xuất cho khối điều khiển tần số - điện áp; Phần III đề cập đến quá trình mô hình hoá bộ xử lý lô-gic mờ – phân lõi quan trọng của bộ điều khiển tần số - điện áp – bằng ngôn ngữ mô tả phần cứng VHDL; Phần IV cung cấp các kết quả mô phỏng và thực thi phần cứng trên công nghệ FPGA; và cuối cùng, Phần V đưa ra một số kết luận ban đầu về hoạt động cũng như khả năng ứng dụng bộ điều khiển đề xuất.

II. ĐỀ XUẤT MÔ HÌNH BỘ ĐIỀU KHIỂN TẦN SỐ - ĐIỆN ÁP

Trong bài báo này, chúng tôi xem rằng lưu lượng truyền thông qua một bộ định tuyến cũng chính là đại lượng phản ánh mức độ hoạt động của bộ định tuyến đó. Nếu lưu lượng truyền thông lớn, bộ định tuyến cần phải được cấp tần số cũng như điện áp hoạt động cao hơn để đáp ứng tốc độ truyền dữ liệu cao và ngược lại. Do đó, để giảm thiểu năng lượng tiêu thụ của các bộ định tuyến được sử dụng tại các nút mạng trong một hệ thống có kiến trúc truyền thông dạng mạng trên chip, chúng tôi đề xuất sử dụng một khối điều khiển tần số - điện áp để thay đổi tần số hoạt động của bộ định tuyến theo mức độ hoạt động của chính bộ định tuyến đó. Khối điều khiển này có nhiệm vụ giám sát lưu lượng truyền thông qua bộ định tuyến, từ đó dự đoán lưu lượng truyền thông qua bộ định tuyến trong tương lai để đưa ra quyết định tăng hoặc giảm tần số một cách phù hợp với hoạt động của bộ định tuyến. Bên cạnh đó, các mức điện áp cung cấp cho bộ định tuyến cũng được khối điều khiển điều chỉnh phù hợp với tần số hoạt động của bộ định tuyến này nhằm giảm tối đa công suất tiêu thụ của bộ định tuyến.

Để đơn giản hoá cấu trúc của bộ giám sát lưu lượng, đồng thời để giảm thiểu được tài nguyên thực thi hệ thống, chúng tôi đề xuất sử dụng mô hình lô-gic mờ để áp dụng cho quá trình dự đoán lưu lượng và đưa ra quyết định điều khiển tần số - điện áp. Hệ thống được đề xuất sẽ bao gồm các khối sau: khối tính toán giá trị lưu lượng trung bình cực đại (**MA: Max Average**), khối tính toán biến thiên lưu lượng (**DER: Derivative**), bộ xử lý lô-gic mờ (**FLP: Fuzzy Logic Processing**) và các khối điều khiển tần số (**FC: Fclk controller**), điện áp (**VC: Vdd controller**) và được mô tả như sơ đồ khối ở Hình 1. Chức năng của từng khối trong sơ đồ này được mô tả như sau:

Giá trị lưu lượng từ các cổng truyền thông của bộ định tuyến sẽ được gửi đến khối **MA**. Khối này sẽ thực hiện việc so sánh các giá trị để chọn ra giá trị lưu lượng lớn nhất đã được truyền qua bộ định tuyến và chuyển giá



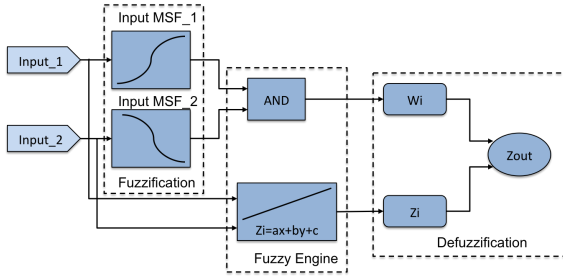
Hình 1. Sơ đồ khối của mô hình đề xuất.

trị này đến đầu vào thứ nhất của bộ xử lý lô-gic mờ **FLP**. Mục đích của quá trình này chính là để cho bộ **FLP** có cơ sở để đưa ra quyết định điều chỉnh tần số, điện áp của bộ định tuyến phù hợp với yêu cầu truyền thông cao nhất qua bộ định tuyến đó.

Khối **DER** được thiết kế để có thể xác định sự biến thiên về lưu lượng truyền thông của một cổng. Các giá trị lưu lượng qua mỗi cổng sẽ được gửi đến các bộ **DER** tương ứng với cổng đó. Các khối **DER** sẽ thực hiện việc lưu hai giá trị lưu lượng: một giá trị của thời điểm hiện tại và một giá trị của thời điểm trước đó một khung thời gian. Từ đó, khối **DER** sẽ xác định được sự biến thiên về mặt lưu lượng truyền thông qua từng cổng. Kết hợp với thông tin về lưu lượng cực đại từ khối **MA**, khối **DER** sẽ đưa ra giá trị biến thiên của lưu lượng để làm đầu vào cho thông số biến thiên lưu lượng của bộ xử lý lô-gic mờ.

Bộ xử lý lô-gic mờ **FLP** được xây dựng từ ba khối con đó là: khối mờ hoá (**FZ: Fuzzification**), khối “động lực” (**FE: Fuzzy Engine**) và khối giải mờ (**DFZ: Defuzzification**). Như đã đề cập ở trên, hoạt động của **FLP** dựa trên mô hình Sugeno [11] nhằm đơn giản quá trình mô hình hoá và các quá trình tính toán và nhờ đó, giảm thiểu tài nguyên phần cứng cần sử dụng.

- Khối **FZ** nhận dữ liệu từ hai đầu vào đó là dữ liệu về lưu lượng truyền thông lớn nhất đi qua bộ định tuyến và dữ liệu về sự biến thiên của lưu lượng đó trong một đơn vị thời gian. Nhiệm vụ của khối này là làm mờ hoá dữ liệu dựa trên các hàm liên thuộc (**MSF: Membership Function**) đã được đề xuất. Kết quả ở đầu ra của khối này là ta thu được các giá trị về mức độ phụ thuộc của các hàm liên thuộc (gọi là giá trị hay cấp độ liên thuộc (degree of membership function) – $\mu(x)$).
- Khối **FE** sẽ dựa trên bảng quy tắc hợp thành (quy tắc IF-THEN) được đề xuất để xác định ra các giá trị trọng số của các quy tắc hợp thành này - w_i . Ngoài ra, vì được xây dựng trên mô



Hình 2. Mô hình của bộ FLP được đề xuất.

hình Sugeno nên khối FE cũng có nhiệm vụ tính toán giá trị ra z_i cho từng quy tắc hợp thành đã được đề xuất.

- Khối **DFZ** sẽ dựa trên giá trị liên thuộc của các hàm liên thuộc - $\mu(x)$, giá trị trọng số w_i của các quy tắc hợp thành và giá trị ra z_i của mỗi quy tắc để xác định giá trị đầu ra của bộ điều khiển. Giá trị này chính là giá trị cần thiết để điều chỉnh tần số hoạt động của bộ định tuyến phù hợp với lưu lượng và sự biến thiên của lưu lượng qua bộ định tuyến đó.

Khối **Vcc controller** và **Fclk controller** là các khối điều khiển điện áp và tần số hoạt động của bộ định tuyến. Trong mô hình được đề xuất, bộ định tuyến sẽ sử dụng ba cặp giá trị của tần số - điện áp. Sự thay đổi của tần số hoạt động đồng thời cũng sẽ điều chỉnh điện áp hoạt động về mức tương ứng với tần số mới. Sự thay đổi tần số hoạt động sẽ được xác định thông qua giá trị điều khiển ở đầu ra của bộ xử lý lô-gic mờ.

III. MÔ HÌNH HOÁ BỘ XỬ LÝ LÔ-GIC MỜ BẰNG NGÔN NGỮ MÔ TẢ PHẦN CỨNG VHDL

Mô hình bộ FLP được đề xuất là một xử lý lô-gic mờ với hai đầu vào, một đầu ra, sử dụng mô hình Sugeno bậc 0 [11] (Hình 2). Mô hình này được thực thi với từng khối con như sau:

- Khối mờ hoá **FZ** được thực thi với hai khối con **input_MSF**. Mỗi khối con là một quá trình tính toán giá trị liên thuộc của các đầu vào $input_1$ và $input_2$ dựa trên các hàm liên thuộc được đề xuất.
- Khối “động lực” **FE** bao gồm hai khối con, khối **AND-rule** xác định giá trị trọng số cho từng quy tắc hợp thành và khối **Zi** được dùng để tính toán giá trị đầu ra của mỗi quy tắc hợp thành tương ứng.
- Khối giải mờ **DZ** được thực thi như một quá trình tính toán giá trị đầu ra cuối cùng của bộ

xử lý lô-gic mờ dựa trên các giá trị trọng số w_i và giá trị đầu ra của từng quy luật hợp thành z_i .

Toàn bộ các khối này sau đó được mô hình hoá bằng ngôn ngữ mô tả phần cứng VHDL dưới dạng các **Process**. Các thanh ghi được chèn vào sau mỗi **Process** để đảm bảo luồng dữ liệu được thực hiện đúng. Việc mô hình hoá bộ **FLP** bằng ngôn ngữ VHDL ở mức RTL được mô tả như trong Hình 3.

A. Quá trình mờ hoá

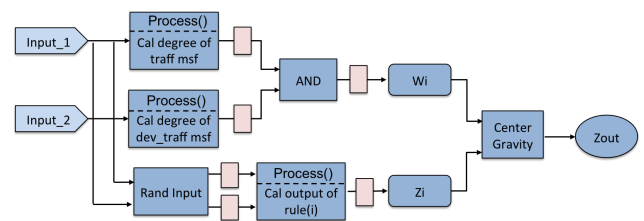
Quá trình mờ hoá là quá trình chuyển đổi từ một giá trị rõ của đầu vào, thành một giá trị liên thuộc đối với từng hàm liên thuộc. Giá trị liên thuộc này sẽ phụ thuộc vào hình dạng của hàm liên thuộc và tập xác định của nó. Thông thường, các hàm liên thuộc thường có dạng hình tam giác, hình thang, hoặc hình chuông (phân bố Gauss)... Tuy nhiên, để việc mô hình hoá hệ thống được đơn giản thì trong bài báo này chúng tôi đã lựa chọn dạng hình của hàm liên thuộc là hình tam giác.

Để mô hình hoá quá trình mờ hoá, một hàm liên thuộc dạng hình thang sẽ được mô tả với một bộ các tham số như ở Hình 4. Hàm liên thuộc dạng tam giác sẽ là sự đơn giản hoá của hàm dạng hình thang với tham số $point_2$ trùng với điểm b. Bộ tham số cần mô tả khi xây dựng một hàm liên thuộc dạng tam giác sẽ bao gồm: $point_1$, $slope_2$, $point_2$, $slope_2$. Giá trị liên thuộc $\mu(x)$ theo đầu vào theo hàm liên thuộc sẽ được tính toán dựa trên ba phân đoạn chính:

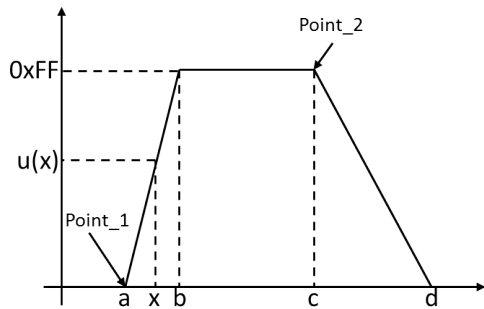
- Nếu $input_1 \in [0, a]$ hoặc $input_1 > d$ thì $\mu(x) = 0$.
- Nếu $input_1 \in [b, c]$ thì $\mu(x) = 1$
- Nếu $input_1 \in [a, b]$ hoặc $input_1 \in [c, d]$ thì $\mu(x)$ sẽ được tính theo công thức:

$$\mu(x) = \begin{cases} (input_1 - point_1) * slope_1, & \text{if } input_1 < point_2 \\ 0xFF - (input_1 - point_2) * slope_2, & \text{if } input_1 > point_2 \end{cases} \quad (1)$$

Giá trị liên thuộc $\mu(x) \in [0, 1]$. Tuy nhiên, để thuận tiện trong việc tính toán thì khoảng xác định này sẽ được rời



Hình 3. Mô hình hoá ở mức RTL của bộ FLP.



Hình 4. Mô tả hàm liên thuộc dạng hình thang.

rạc hoá thành 2^8 khoảng nhỏ có giá trị rời rạc, bắt đầu từ 0x00 (tương ứng với giá trị 0) đến 0xFF (tương ứng với giá trị 1). Với việc mô tả một hàm liên thuộc bằng bộ bốn thông số như trên thì trong ngôn ngữ VHDL, các hàm liên thuộc sẽ được mô tả dưới dạng một tập dữ liệu dạng **Record** như sau.

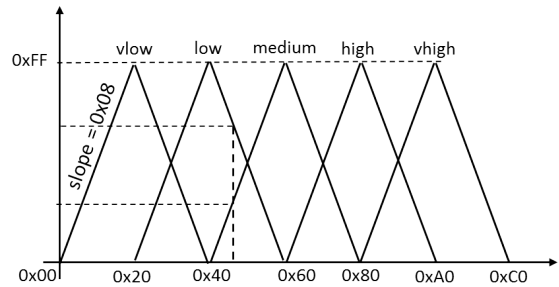
```

type traffic_type is (term_of_mfs);
type traffic_membership is
record
term: traffic_type;
point_1: std_logic_vector(7 downto 0);
slope_1: std_logic_vector(7 downto 0);
point_2: std_logic_vector(7 downto 0);
slope_2: std_logic_vector(7 downto 0);
end record;

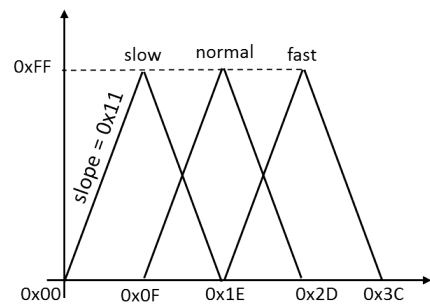
```

Quá trình mờ hoá giá trị lưu lượng ở đầu vào *input_1* sẽ được thực hiện thông qua năm hàm liên thuộc, tương ứng với các biến ngôn ngữ tự nhiên là: “*vlow, low, medium, high, vhigh*”. Các hàm liên thuộc được sử dụng trong mô hình này là các hàm dạng tam giác với các giá trị của tập dữ liệu được mô tả như trong Hình 5. Giá trị cực đại của hàm liên thuộc *vhigh* là 0xC0, tương ứng với giá trị lưu lượng cực đại là 192Gflits/s. Giá trị này được lựa chọn phù hợp với tốc độ truyền thông cực đại của bộ định tuyến đã được chúng tôi thiết kế (vào khoảng 180Gflits/s). Hệ số *slope_1* và *slope_2* được chọn là 0x08 cho cả hai sườn của hàm liên thuộc, mục đích là đảm bảo việc tính toán sẽ được thực hiện với sai số ít nhất có thể. Quá trình mờ hoá giá trị lưu lượng sẽ tương ứng với một khối **Input_MSF** trong mô hình được đề xuất và được thực thi như một **Process** trong ngôn ngữ VHDL.

Đầu vào *input_2* là giá trị của biến thiên lưu lượng cực đại qua bộ định tuyến. Giá trị của biến thiên lưu lượng được xác định là lượng thay đổi về lưu lượng, mà không phụ thuộc vào giá trị đó là tăng hay giảm. Với đầu vào *input_2*, quá trình mờ hoá sẽ được thực hiện thông qua ba hàm liên thuộc với các biến ngôn ngữ tương ứng là: “*slow, normal, fast*”. Các hàm liên thuộc này cũng là các hàm tam giác và giá trị của tập xác định được



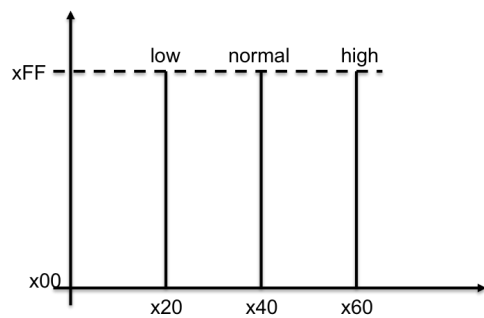
Hình 5. Các hàm liên thuộc của đầu vào *input_1*.



Hình 6. Các hàm liên thuộc của đầu vào *input_2*.

mô tả như trong Hình 6. Giá trị cực đại của hàm liên thuộc *high* là 0x3C tương ứng với lưu lượng biến thiên 60Gflits/s. Giá trị *slope_1* và *slope_2* của các hàm liên thuộc đều bằng nhau và được chọn bằng 0x11.

Đầu ra Z_{out} của **FLP** là một giá trị hằng số, tương ứng với tần số hoạt động của bộ định tuyến. Vì vậy hàm liên thuộc của đầu ra sẽ là các hàm đơn trị (singleton). Đầu ra sẽ được mô tả bằng ba hàm liên thuộc với các biến ngôn ngữ là: *low, normal* và *high* (Hình 7). Các hàm liên thuộc ở đầu ra sẽ được mô tả trong VHDL như là các hằng số 8-bit.



Hình 7. Các hàm liên thuộc của đầu ra *output*.

B. Quá trình xác định quy luật hợp thành

Bằng việc áp dụng mô hình Sugeno, một quy luật hợp thành trong mô hình có thể được phát biểu dưới dạng quy luật IF-THEN như sau:

“**IF** $input_1 = x$ **AND** $input_2 = y$ **THEN**
 $Output = ax + by + c$ ”

Trong trường hợp $a = b = 0$ và đầu ra là một hằng số thì ta gọi mô hình này là mô hình Sugeno bậc 0. Trong mô hình này, với mỗi giá trị đầu ra z_i của từng quy luật hợp thành sẽ được đặc trưng bởi một giá trị trọng số w_i . Giá trị trọng số này sẽ phụ thuộc vào quy luật hợp thành mà ta đặt ra khi tổ hợp các biến ngôn ngữ tự nhiên. Giả sử, nếu ta áp dụng quy tắc AND với đầu vào $input_1=x$ và $input_2=y$ thì giá trị này sẽ là:

$$w_i = MIN(\mu(x), \mu(y)) \quad (2)$$

Quá trình tính toán giá trị trọng số w_i sẽ tương ứng với khối **AND-rule** trong mô hình được đề xuất. Khi thực thi trong ngôn ngữ VHDL, quá trình là một **Process** và được thực hiện thông qua đoạn mã tìm kiếm giá trị cực tiểu như sau:

```
rule_weight: process (u_x, u_y) is
  variable i, j: integer;
begin
  for i in 1 to n loop
    for j in 1 to m loop
      if (u_x(i) < u_y(j)) then
        rule_w(i, j) <= u_x(i);
      else
        rule_w(i, j) <= u_y(j);
      end if;
    end loop;
  end loop;
end process rule_weight;
```

Trong đó, u_x và u_y chính là các giá trị liên thuộc của các đầu vào $input_1$ và $input_2$. Với các hàm liên thuộc được mô tả như trong Phần III-A, ta sẽ có tất cả 5×3 quy luật hợp thành. Thông thường, trong tổng số các quy luật hợp thành được đề xuất, chúng ta có thể bỏ qua những quy luật ít phổ biến hoặc ít xuất hiện. Tuy nhiên, trong bài báo này vì số lượng quy luật hợp thành không lớn nên chúng tôi vẫn tiến hành thực thi toàn bộ 15 quy luật này. Các quy luật hợp thành được mô tả như trong Bảng I.

C. Quá trình giải mờ

Quá trình giải mờ là quá trình tính toán lại giá trị chính xác ở đầu ra của **FLP**. Với các giá trị đầu ra của từng quy luật hợp thành z_i và giá trị trọng số w_i của nó, quá trình giải mờ sẽ sử dụng phương pháp giá trị trọng tâm để tính ra giá trị đầu ra Z_{out} . Giá trị này sẽ

Bảng I. CÁC QUY LUẬT HỢP THÀNH CỦA KHỐI LÔ-GIC MỜ

Lưu lượng	Biến thiên lưu lượng	Tần số
vlow	slow	low
low	slow	low
medium	slow	low
high	slow	normal
vhigh	slow	normal
vlow	normal	low
low	normal	low
medium	normal	normal
high	normal	high
vhigh	normal	high
vlow	fast	normal
low	fast	normal
medium	fast	high
high	fast	high
vhigh	fast	high

được tính theo công thức sau:

$$Z_{out} = \frac{\sum_{i=1}^n w_i \cdot z_i}{\sum_{i=1}^n w_i} \quad (3)$$

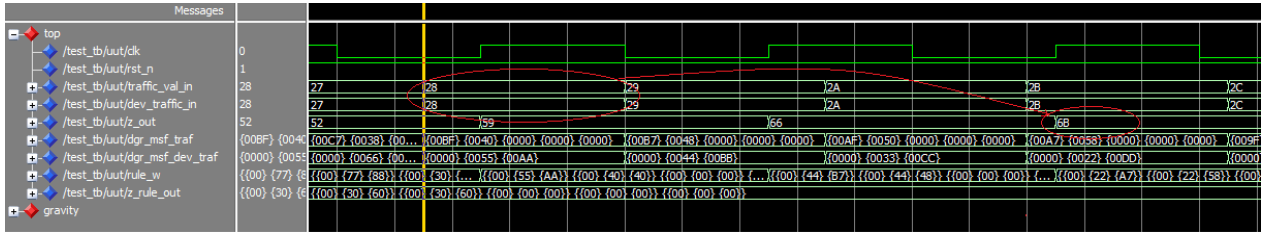
Trong đó, n là tổng số các quy tắc hợp thành đã được thiết lập. Đoạn mã VHDL sau được dùng để mô tả quá trình tính toán này:

```
for i in 1 to n loop
  for j in 1 to m loop
    w_t:= unsigned(w(i, j));
    z_t:= unsigned(z(i, j));
    upper:= upper + (w_t*z_t);
    lower:= lower + w_t;
  end loop;
end loop;
z_out:= divide(upper, lower);
```

Trong đó, n và m là số hàm liên thuộc của các đầu vào $input_1$ và $input_2$. $w(i, j)$ và $z(i, j)$ chính là các giá trị trọng số và giá trị đầu ra của các quy tắc hợp thành được tính ra trong những khối trước đó.

IV. KẾT QUẢ MÔ PHỎNG VÀ THỰC THI

Sau khi tiến hành thực thi toàn bộ các khối của bộ xử lý lô-gic mờ **FLP** bằng ngôn ngữ VHDL ở mức RTL, chúng tôi đã tiến hành mô phỏng hoạt động của các khối này bằng phần mềm mô phỏng ModelSim để kiểm chứng các hoạt động của từng khối con cũng như hoạt động của toàn bộ **FLP**. Quá trình mô phỏng kiểm tra có thể được mô tả bằng sơ đồ khối như ở Hình 8. Các dữ liệu đầu vào sẽ được tự động sinh ra và đưa vào các đầu vào $input_1$ và $input_2$ (tương ứng với các đầu vào $traffic_val_in$ và $dev_traffic_in$ trong chương trình). Giá trị của dữ liệu vào sẽ được giới hạn trong khoảng giá trị cực đại của các hàm liên thuộc. Bằng cách quan sát dạng sóng của các tín hiệu ra và giá trị của đầu ra cuối



Hình 9. Giải đồ dạng sóng của bộ FLP.

Bảng II. KẾT QUẢ TỔNG HỢP PHẦN CỨNG TRÊN SPARTAN 3E-XC3S500E-5VQ100.

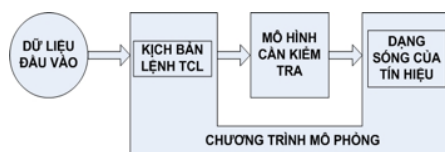
Loại cổng logic	Số lượng sử dụng	Số lượng sẵn có	Tỷ lệ
Slices	711	4656	15%
Flip flop Slices	197	9312	2%
LUTs 4 đầu vào	1325	9312	14%
Bộ nhân	19	20	95%

cũng Z_{out} (tương ứng với đầu ra z_{out} trong chương trình), ta sẽ khẳng định được hoạt động của bộ FLP là đúng hay không. Kết quả dạng sóng của quá trình mô phỏng được trình bày như ở Hình 9. Từ kết quả mô phỏng, ta có thể thấy FLP đã thực hiện đúng chức năng của mình như đã được đề xuất trong mô hình trước đó.

Chúng tôi cũng đã tiến hành thực thi phần cứng bộ xử lý lô-gic mờ FLP trên FPGA (Spartan 3E-xc3s500e-5vq100) sử dụng phần mềm Xilinx ISE để ước lượng tài nguyên phần cứng. Kết quả tổng hợp được mô tả như ở Bảng II.

V. KẾT LUẬN

Trong bài báo này, chúng tôi đã giới thiệu mô hình một khối điều khiển tần số-điện áp sử dụng thuật toán lô-gic mờ có thể áp dụng để tối ưu công suất tiêu thụ cho mạng trên chip. Bộ điều khiển phân tích lưu lượng và biến thiên lưu lượng qua bộ định tuyến mạng để từ đó phân tích và đưa ra quyết định tăng/giảm tần số và điện áp. Khối điều khiển được ứng dụng lý thuyết điều khiển lô-gic mờ để giảm thiểu độ phức tạp của hệ thống, tiết kiệm được tài nguyên phần cứng khi thực thi nhưng vẫn đảm bảo việc theo dõi lưu lượng chuyển qua bộ định tuyến và đưa ra được các lệnh điều khiển phù hợp. Mô hình của bộ xử lý lô-gic mờ – phần lõi quan trọng của bộ điều khiển tần số - điện áp – sau đó đã được mô hình



Hình 8. Sơ đồ khối của quá trình mô phỏng kiểm tra FLP.

hoá bằng ngôn ngữ VHDL ở mức RTL. Các kết quả mô phỏng cho thấy bộ xử lý lô-gic mờ đã hoạt động đúng theo yêu cầu của mô hình đề ra ban đầu.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Quỹ phát triển khoa học và công nghệ quốc gia (NAFOSTED) trong đề tài mã số 102.01-2013.17 “Reconfiguration Solution in Designing Network-on-Chip Architectures (ReSoNoC)”.

TÀI LIỆU THAM KHẢO

- [1] L. A. Zadeh, “Fuzzy Sets,” *Information and Control*, Vol. 8, No. 3, pp. 338-353, 1965.
- [2] D. H. Song and S. Jung, “Neural Compensation Technique for Fuzzy Controller Humanoid Robot Arms: Experimental Studies,” *Proc. of IEEE int. symp. on intelligent control*, pp. 424-429, Oct. 2007.
- [3] Y. C. Yeh, W. J. Wang, and C. W. Chiou, “Heart-beat Case Determination Using Fuzzy Logic Method on ECG Signals,” *Int. Journal of Fuzzy Systems*, Vol. 11, No. 4 pp. 250-261, Dec. 2009.
- [4] A. Lakdashti, M. S. Moin, and K. Badie, “Reducing the Semantic Gap of the MRI Image Retrieval Systems Using a Fuzzy Rule Based Technique,” *Int. Journal of Fuzzy Systems*, Vol. 11, No. 4, pp. 232-249, Dec-2009.
- [5] Vuong, P.T.; Madni, A.M.; Vuong, J.B., “VHDL Implementation for a Fuzzy Logic Controller,” *Automation Congress, 2006. WAC '06. World*, pp. 1-8, 24-26 July 2006.
- [6] Deliparaschos, K. M. and Nenedakis, F. I. and Tzafestas, S. G., “Design and Implementation of a Fast Digital Fuzzy Logic Controller Using FPGA Technology,” *Journal of Intelligent and Robotic Systems*, Vol. 45, pp. 77-96, 2006.
- [7] Uppalapati, S.; Kaur, D., “Design and implementation of a Mamdani fuzzy inference system on an FPGA,” *In Proc. of Annual Meeting of the North American of Fuzzy Information Processing Society (NAFIPS)*, pp.1-6, 14-17 June 2009.
- [8] S. Guo, L. Peters, and H. Surmann, “Design and Application of an Analog Fuzzy Logic Controller,” *IEEE Trans. on Fuzzy Systems*, Vol. 4, No. 4, Nov. 1996.
- [9] K. Nakamura, N. Sakashita, Y. Nitta, K. Shimomura, and T. Tokuda, “Fuzzy Inference and Fuzzy-Inference Processor,” *IEEE Micro*, Vol. 13, No. 5, pp. 37-48, Oct. 1993.
- [10] W. Dally and B. Towles, “Route packets, not wires: On-chip interconnection networks,” *In Proceedings of the DAC*, 2001, pp. 684-689.
- [11] M. Sugeno, “An Introductory Survey of Fuzzy Control,” *Information Science*, Vol. 36, pp. 59-83, 1985.