A Fuzzy-Logic based Voltage-Frequency Controller for Network-on-Chip Routers

Hai-Phong Phan, Xuan-Tu Tran
SIS Laboratory, VNU University of Engineering and Technology – 144 Xuan Thuy road, Cau Giay, Hanoi, Vietnam
phongph.de12@vnu.edu.vn; tutx@vnu.edu.vn

***Abstract*—Low power design can be easily achieved by scaling the voltage and frequency of the target systems. The most concerning issue is how to make the voltage-frequency scaling adaptable to the required performance of the system at run-time. In this paper, we present the design of a voltage-frequency controller for network-on-chip routers based on fuzzy-logic processing. The communication traffic of a network router will be predicted by a fuzzy-logic algorithm. Then the voltage and frequency of the router will be scaled according to the predicted results in order to get power consumption optimal for the network router. The most important part of the proposed controller, the fuzzy-logic processor (FLP), is modeled and verified using VHDL and then implemented on FPGA devices.**

***Keywords*—Low power, Network-on-Chip, DVFS, fuzzy logic.**

I. INTRODUCTION

The fuzzy set theory was first proposed by L.A Zadeh in 1965 [1]. It has been widely applied in many application fields, from control theory to artificial intelligence. The typical fields in which the fuzzy logic theory has been successfully applied include automation control, power saving, data processing, signal processing…, especially robotic design. Many researches focus on the design of robots with human behaviors based on fuzzy logic. By using fuzzy logic algorithms, the behavior of robots can be implemented through the decision "if – then" which is similar to human thinking. Therefore, robots can have "thinking" and actions as humans than the previous robots generation [2].

Previously, fuzzy logic systems are usually implemented by software. The advantages of this method are the ability of quick deployment, easy modification, time reduction in development, and low cost. However, one of the most disadvantages is that the processing and computation speed is too slow. This affects the implementation of fuzzy logic in real-time systems.

Fuzzy logic theory has prompted the researchers to develop fuzzy logic systems with faster processing and calculating speed, and more power efficiency. One of the recent trends for developing fuzzy logic systems is to deploy the system as a hardware core to increase the processing speed of the system [3], [4], [5]. Besides that, the power consumption is also an important factor in designing complex systems, especially in designing Network-on-Chip (NoC) based systems. The NoC paradigm has been recently known as an emerging solution for designing large, complex system-on-chips [6]. In the NoC based systems, computing units (i.e. Intellectual Properties or IPs) communicates with each other using a micro network that is composed of network routers and network links.

In this paper, we focus on designing a voltage-frequency controller for NoC routers. The role of the proposed controller is to adjust the frequency and voltage of the target router according to its workload (the communication traffic going through). Therefore, the power consumption of each router as well as the whole system can be reduced while keeping system performance suitable. To do that, we apply a fuzzy logic algorithm using Sugeno model [7] in the controller. The design is then modeled and implemented using VHDL at RTL level.

The remaining part of the paper is organized as follows. Section II presents the model of the voltage-frequency controller. The processing principle of the proposed fuzzy logic processor (FLP), which is the most important part of the voltage-frequency controller, is shown in Section III. The simulation and experimental results of the FLP with two inputs and 8-bit resolution are given in Section IV. Finally, conclusions will be provided in Section V.

II. THE PROPOSED VOLTAGE-FREQUENCY CONTROLLER

In this paper, we assume that the traffic going through a router is also a quantity that reflects the activities of this router. If the router has a large communication traffic, it must be supplied a higher frequency, as well as a higher voltage, to meet the high data transmission rate and vice versa.

Therefore, we propose to use a voltage-frequency controller to scaling voltage and frequency of the router according to the activities of it in order to reduce the power consumption of a router used in a network node of a NoC system. The controller will monitor the communication traffic through the router, and then predict the change of traffic to make a decision to increase or decrease the values of voltage and frequency accordingly.

To simplify the structure and reduce hardware resources of the system, we propose a voltage-frequency controller as shown in Fig. 1. In this controller, we use a fuzzy logic processor to predict the communication traffic and take the decision about the values of frequency and voltage.

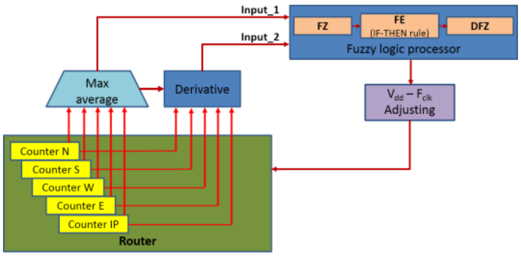

Fig. 1. The proposed voltage-frequency controller.

In this architecture, each router input port will be equipped with a traffic counter. These counters count the data flits passing through the router in certain clock cycles (average traffic) based on the corresponding response signals from the router. Since the router normally has 5 input/output ports [8], there will be 5 communication traffic values from the router. The maximum traffic value passing through the router is then decided by the Max Average (MA) block. In fact, the MA will compare and find the maximum value of the five average traffic values given by the router..Finally, this information will be sent to the *Input_1* of the FLP for being processed.

The Derivative (DER) block calculates the derivative of traffics obtained from the counters. To do that, it receives the traffic values from the counters and store these values to buffers. The derivation of traffic will be calculated by the present value and the previous value. The DER determines the derivative value of traffic according to the maximum traffic value decided by MA block and then gives it to *Input_2* of the FLP for further processes.

The Fuzzy Logic Processor (FLP) will process the given information (maximum traffic value and derivative value) to predict the next communication load passing through the router and decide the suitable voltage and frequency supplied to the router. It is constructed from the three sub-blocks: the Fuzzification (FZ), the Fuzzy Engine (FE) and the Defuzzification (DFZ) as described in Fig. 1. As mentioned above, the operation of FLP is based on Sugeno model to simplify the process of modeling and calculation. As a result, this leads to the reduction of hardware resources required for implementing the whole voltage-frequency controller.

- The FZ receives data from two inputs: the maximum traffic value passing through the router (from MA block) and the derivative of traffic (from DEV block). Then, it converts these data into blur-value based on the membership functions (MSF). At the output of FZ, we get the degree of membership function.

- The FE will be implemented based on evaluation rules (IF-THEN rules) to determine the firing strength of each rule. Because the FLP is built using Sugeno model, the FE block also calculates the levels of output of each rule. This issue will be discussed more detail in the next section.

- Based on the firing strength value and the level of output of each rule, the DFZ will predict the next situation of traffic load of the router. This predicted value is then used to adjust the frequency and the voltage.

The Voltage-Frequency Adjusting (VFA) block controls the voltage and the frequency supplied to the router. In this design, the router is supplied by three pairs of frequency – voltage values (low, medium, high). When the frequency is changed, the voltage will be also adjusted to new level corresponding to the new frequency. The change of frequency is determined by a control signal at the output of FLP.

III. Modelling the Fuzzy-Logic Processor

The proposed Fuzzy-Logic Processor (FLP) is a fuzzy logic system with two inputs and an output based on Sugeno model. The FLP is implemented by three blocks as depicted in Fig. 2.

- The Fuzzification (FZ) block is implemented by two sub-blocks: input_MSF1 and input_MSF2. Each sub-block is a process used to calculate the degree of each input (*input_1* and *input_2*) based on the membership functions.

- The Fuzzy Engine (FE) block is composed of two sub-blocks: the AND-rule is used for calculation of the firing strength w_i and the Zi is used to calculate the output level z_i of each rule.

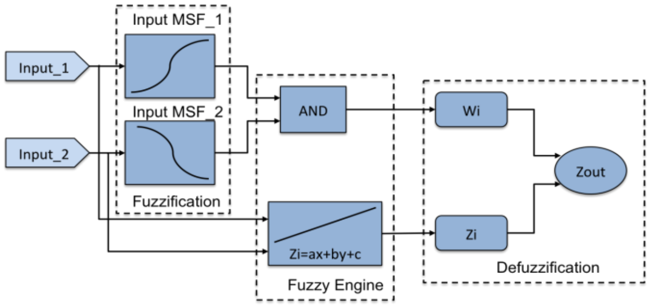

Fig. 2. Fuzzy-logic processor (FLP) model.

- The Defuzzification (DE) block is a process to calculate the final output of system based on the weighted average of all rule outputs.

The processing diagram of the FLP is described in Fig. 3.

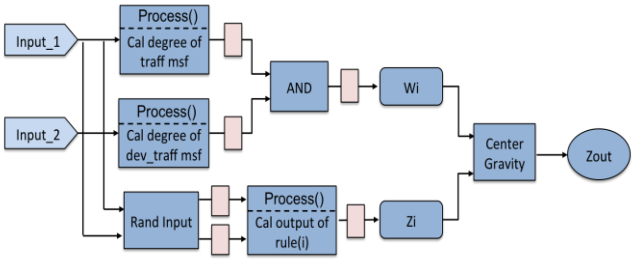

Fig. 3. Processing diagam of the fuzzy-logic processor.

A. Fuzzification

The fuzzification converts a clean value of input to a fuzzy value. This value is characterized by the degree of MSF - *μ(x)* and depends on the shape of MSF, the number of partitions of MSF and the correlation between membership functions.

Different shapes of membership functions have been already proposed. Trapezoidal, Gaussian, and triangular are some common shapes for membership functions. To simplify the explanation of MSF's mathematical formulation, without loss of generality, let us consider only triangular/trapezoidal MSF.

A trapezoidal MSF is described by a set of parameters (*point_1*, *b*, *c*, *point_2*) as illustrated in Fig. 4. The triangle MSF is the simplification of trapezoidal MSF with parameters *point_2*, and *b* is the same.

To describe the triangle MSF easily, the set of parameters becomes (*point_1*, *slope_1*, *point_2*, *slope_2*). The degree of membership is calculated as below.

- If $input_1 \in [0, a]$ or $input\_1 > d$ then $\mu(x) = 0$

- If $input\_1 \in [b, c]$ then μ(x) = 1
- If $input\_1 \in [a, b]$ or $input\_1 \in [c, d]$ then

$$\mu(x) = \begin{cases} (input\_1 - point\_1) \times slope\_1, \\ \quad if\ input\_1 < point\_2. \\ 0xFF - (input\_1 - point\_2) \times slope\_2, \\ \quad if\ input\_1 > point\_2. \end{cases}$$

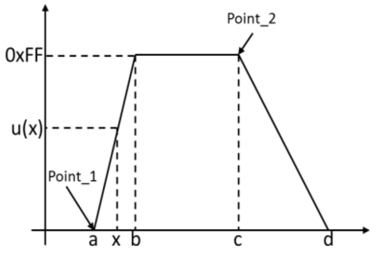

Fig. 4. The description of trapezoidal membership functions.

The degree of membership is discrete with 8-bit resolution. Therefore, $\mu(x) = 1$ equals to 0xFF in hexadecimal. In VHDL, the MSF is described as a set of parameters by a Record data type as below:

```
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;
```

The fuzzification of the traffic's value at *input_1* is implemented through five membership functions named: *vlow, low, medium, high, vhigh*.

The shape of membership functions used in this model is the triangular. The values of parameters are described in Fig. 5. The maximum value of the membership function *vhigh* is 0xC0, corresponding to the maximum value of traffic is 192*Mflits/s*. This value is selected in accordance with the maximum communication speed of a router has been designed in [8], approximately 180*Mflits/s*. The value of parameters *slope_1* and *slope_2* are 0x08 for both slopes of all membership functions. The purpose is to ensure that the calculation will be done with minimum error possibility.

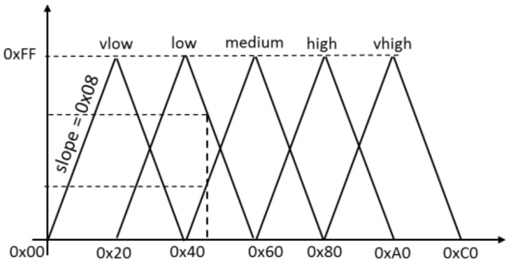

Fig. 5. The membership functions of *input_1*.

The value of *input_2* is the derivative of maximum traffic through the router. This value is defined as an absolute value of the traffic's change in a unit of time. The degree of membership at *input_2* is calculated through three membership functions with linguistic variables respectively: *slow, normal, fast*. The membership functions are also triangular functions, which are described in Fig. 6. The maximum value of the MSF *high* is 0x3C. This value corresponds to the maximum value of traffic variability of 60*Mflits/s*^{2}. The value of *slope_1* and *slope_2* of the membership functions are 0x11.

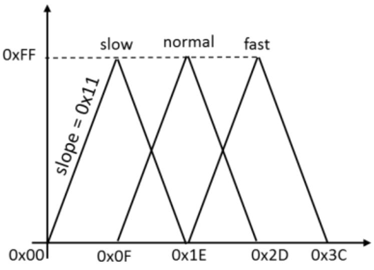

Fig. 6. The membership functions of *input_2*.

The output Z_{out} of FLP is a constant value, corresponding to the operating frequency of the router. Therefore, the membership functions of output are singletons. The output is described by three membership functions: *low, normal and high*. The membership functions of output are described as 8-bit constants in VHDL.

B. Evaluation rule

By applying the Sugeno model, an evaluation rule usually takes the "IF-THEN" statement as follows:

"IF *input_1=x* AND *input_2=y* THEN *Output=ax+by+c".*

In the case *a=b=c=0* and the output is a constant, we have a Sugeno model with zero order. In this model, the value of output z_i is characterized by the firing strength w_i of each rule. This firing strength is based on the rule evaluations of the combination of linguistic variables. Assuming that, we apply the AND rule with *input_1=x* and *input_2=y* then this value is:

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

With the membership functions are described in Section III.A, we have total 5×3 evaluation rules as in TABLE I.

TABLE I. EVALUATION RULE

	traffic	**der_traffic**	**frequency**
1	vlow	slow	slow
2	low	slow	slow
3	medium	slow	slow
4	high	slow	normal
5	vhigh	slow	normal
6	vlow	normal	slow
7	low	normal	slow
8	medium	normal	normal
9	high	normal	fast
10	vhigh	normal	fast
11	vlow	fast	normal
12	low	fast	normal
13	medium	fast	fast

14	high	fast	fast
15	vhigh	fast	fast

C. Defuzzification

The defuzzification is a process to calculate the exact values of FLP's output. With the output value of each rule z_i and the firing strength value of the rule w_i, the final output of the FLP is the weighted average of all rule outputs, which is computed as:

$$Z_{out} = \frac{\sum_{i=1}^{n} w_i . z_i}{\sum_{i=1}^{n} w_i}$$

Where *n* is the number of rules. The VHDL code of this calculation is described as below:

```
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);
```

Where *n* and *m* are the numbers of MSFs of at *input_1* and *input_2*.

IV. Simulation and Implementation Results

After modeling the whole FLP at register-transfer level, we simulated and verified the design by a developed testbench and using ModelSim tool. The testbench generates the data for inputs: *traff_val_in* and *dev_traff_in* (corresponding to *input_1* and *input_2*). Those data are also used by a separate calculation process to calculate the value of the output. By comparing the simulation results and the calculation results, we can conclude the correctness of the FLP. A simulation waveform obtained from ModelSim is presented in Fig. 7. In the circle mark, we can see when the value of each input is 0x28 then the output has a value of 0x68 (after one circle). This result is accordant with the calculation result. All testing results have proved that the operations of FLP are in accordance with the proposed model.

After being successfully modeled and verified, the designed FLP has been implemented on FPGA devices (Spartan 3E-xc3s500e-5vq100) by using Xilinx ISE tool suite. The implementation results are described in TABLE II.

TABLE II. Implementation results on FPGA Devices (Spartan 3E-xc3s500e-5vq100)

Logic Utilization	Used	Available	Utilization
Number of Slices	711	4656	15%
Number of Slice Flip Flops	197	9312	2%
Number of 4 input LUTs	1325	9312	14%
Number of bonded IOBs	26	66	39%
Number of MULT 18X18	19	20	95%

V. Conclusions

In this paper, a voltage-frequency controller using fuzzy logic algorithm for NoC routers has been presented. The controller analyzes the communication traffic passing through the targeted router and the variation of these traffics, then makes decision to increase/ decrease the frequency and voltage supplied to the router in order to reduce the power consumption. The design and implementation of the fuzzy-logic processor (FLP) – the most important part of the proposed controller – are presented and discussed in details.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2013.17.

References

[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] 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.

[4] Deliparaschos, K. M. et al., "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.

[5] 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 Informatio n Processing Society (NAFIPS), pp.1-6, 14-17 June 2009.

[6] W.~Dally and B.~Towles, "Route packets, not wires: On-chip Interconnection Networks", in Proc. of the DAC, 2001, pp. 684--689.

[7] M. Sugeno, "An Introductory Survey of Fuzzy Control",' Information Science, Vol. 36, pp. 59-83, 1985.

[8] Nam-Khanh Dang, et al., FPGA Implementation of a Low Latency and High Throughput Network-on-Chip Router Architecture. In Proceedings of the ICDV 2011, Hanoi, August 2011.

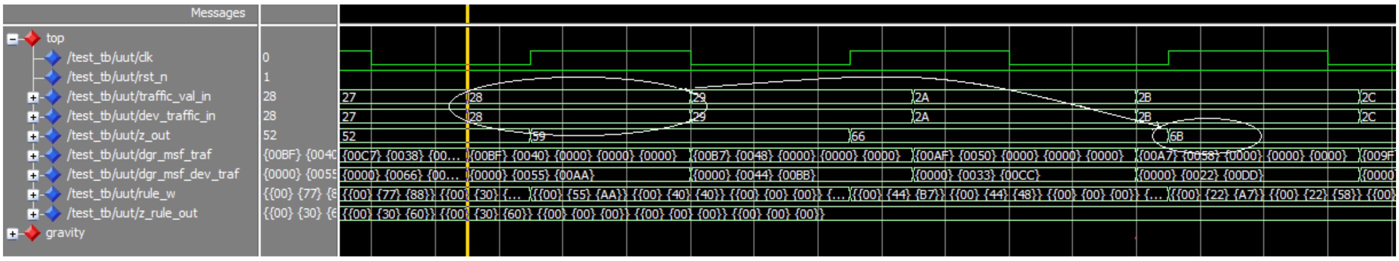
Messages
top
/test_tb/uut/clk 0
/test_tb/uut/rst_n 1
/test_tb/uut/traffic_val_in 28 | 27 | 28 | 29 | 2A | 2B | 2C
/test_tb/uut/dev_traffic_in 28 | 27 | 28 | 29 | 2A | 2B | 2C
/test_tb/uut/z_out 52 | 52 | 59 | 66 | 6B
/test_tb/uut/dgr_msf_traf {00BF} {0040 | {00C7} {0038} {00... | {00BF} {0040} {0000} {0000} {0000} | {00B7} {0048} {0000} {0000} {0000} | {00AF} {0050} {0000} {0000} {0000} | {00A7} {0058} {0000} {0000} {0000} | {009F
/test_tb/uut/dgr_msf_dev_traf {0000} {0055 | {0000} {0066} {00... | {0000} {0055} {00AA} | {0000} {0044} {00BB} | {0000} {0033} {00CC} | {0000} {0022} {00DD} | {0000
/test_tb/uut/rule_w {{00} {77} {8 | {{00} {77} {88}} {{00} {30} {... | {{00} {55} {AA}} {{00} {40} {40}} {{00} {00} {00}} {... | {{00} {44} {B7}} {{00} {44} {48}} {{00} {00} {00}} {... | {{00} {22} {A7}} {{00} {22} {58}} {{00}
/test_tb/uut/z_rule_out {{00} {30} {6 | {{00} {30} {60}} {{00} {30} {60}} {{00} {00} {00}} {{00} {00} {00}} {{00} {00} {00}}
gravity

Fig. 7. The simulation waveform of the FLP.