第 27 卷 第 4 期
2001 年 7 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol.27,No.4
July, 2001

Fuzzy Inference Coprocessor Chip[1)]

SHEN Li[1], ZHU Ya-Jiang[2], XU Hui-E[1], CHEN Xiao-Dong[2]

[1](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)
(E-mail: lshen@gsc.ict.ac.cn)

[2](*Microelectronics R&D Center, Chinese Academy of Sciences, Beijing*, 100029)

Abstract A fuzzy inference coprocessor VLSI chip, F200, was designed as a fuzzy logic controller and fabricated with 0.8 μm CMOS technology. The chip is mainly used for the real-time process control and other suitable applications, such as robot control, classification, expert systems, and so forth. F200 can work with multiple fuzzy knowledge bases and support two types of most useful fuzzy models, Mamdani and Takagi-Sugeno model. The chip has the precision of 12 bits. Its inference speed is about 1.2 M fuzzy rules per second with 20 MHz frequency.

Key words Fuzzy logic control, fuzzy reasoning, fuzzy coprocessor, VLSI chip.

1 Introduction

In the last decade, the fuzzy logic has become increasable widespread. The most successful application is the fuzzy logic control (FLC). It has many advantages as follows. For complex or ill-defined processes (nonlinear, time variant, large lag, big disturbance, etc.), FLC can be used for realizing an intelligent control with fewer rules or program codes than the traditional one. FLC has the fast control performance and good robustness with energy saving. The fuzzy reasoning has the property of parallel processing, so it is suitable for implementing a high-speed fuzzy control inference with VLSI. Since Mamdani's landmark work[1], FLC has rapidly become a significant technology and has been very successful in a variety of industrial and commercial applications[2] Furthermore, the theoretical work of fuzzy systems, such as fuzzy model, stability analysis and design technique also made a remarkable progress. It will certainly promote FLC applications.

Fuzzy logic controllers are usually implemented with software. However, for complex or fast real-time systems, the hardware implementation is required. Since middle of 1980's, several well-known chips were developed[3~6]. They can be classified into digital (binary) and non-digital (non-binary) systems. The later includes circuits of the intrinsic fuzzy logic[4], multi-value, analogue and so forth. Though the non-digital circuits may have some advantages, such as high-speed and fewer devices by use of intrinsic properties of the fuzzy logic, the binary digital VLSI is still in the dominant position due to the mature design and fabrication technology today. Actually, commercial fuzzy chips are mainly based on binary digital circuits. In other hand, as the semiconductor technology continues to improve, chip designers can put more functionality on a single chip. That is so called as "System on Chip (SoC)." Therefore, the fuzzy logic control chip can be designed as a core and integrated with a standard CPU, memories, AD/DA, etc. to realize a whole fuzzy control system.

We implemented an experiment fuzzy chip, F100[7], under the 863 National High Technology Research and Development Program. F100 was applied to several control systems. Among these, a combustion control system was implemented for the industrial boiler[8]. The result was very encouraging with the stable control under a big disturbance. In this paper, we present a new fuzzy chip, F200, with more general and powerful functionality. The main applications of the chip are the real-time process control and other suitable cases, such as robot control, classification, expert systems, and so forth. The new chip has the following features.

[1)]Supported by the 863 National High Technology R&D Program(863-306-ZT01-07-1).
Received 11 July, 2000; in revised form 2 November, 2000
收稿日期 2000-07-11 收修改稿日期 2000-11-02

1) Multiple fuzzy knowledge bases, such as on-line working base, on-line tuning base and back up bases.

2) Support the hierarchy control structure. Outputs of a high-lever controller can be directly used as inputs of a low-level controller.

3) Two types of most useful fuzzy models, Mamdani model[1] and Takagi-Sugeno (TS) model[9].

4) Multiple types of fuzzy rules, membership functions and fuzzy inferences.

5) Scale factors are integrated on chip.

6) High precision. 16 bits integer for variables, 12 bits integer for discourse and 12 bits float-point for membership grade.

7) Chip is designed as a coprocessor.

In the rest of the paper, the next section will describe fuzzy control inference methods of the chip. In the Section 3, the architecture of the fuzzy coprocessor F200 is given. Section 4 concerns the chip design. Finally, Section 5 is the conclusion.

2 Fuzzy Control Inference

The fuzzy logic controller is a rule based fuzzy system. Fig.1 shows the typical configuration of FLC. It consists of four main components as follows.

1) Fuzzy knowledge base. It determines all control operations. The knowledge base contains fuzzy control rules, membership functions of I/O variables and other necessary definitions.

2) Fuzzy inference engine. It executes the fuzzy approximate reasoning. That is based on the knowledge base to calculate output values from a given set of input values.

3) Fuzzifier. It converts crisp input data into fuzzy values (linguistic values).

4) Defuzzifier. It converts fuzzy values into crisp output values as the non-fuzzy control actions.

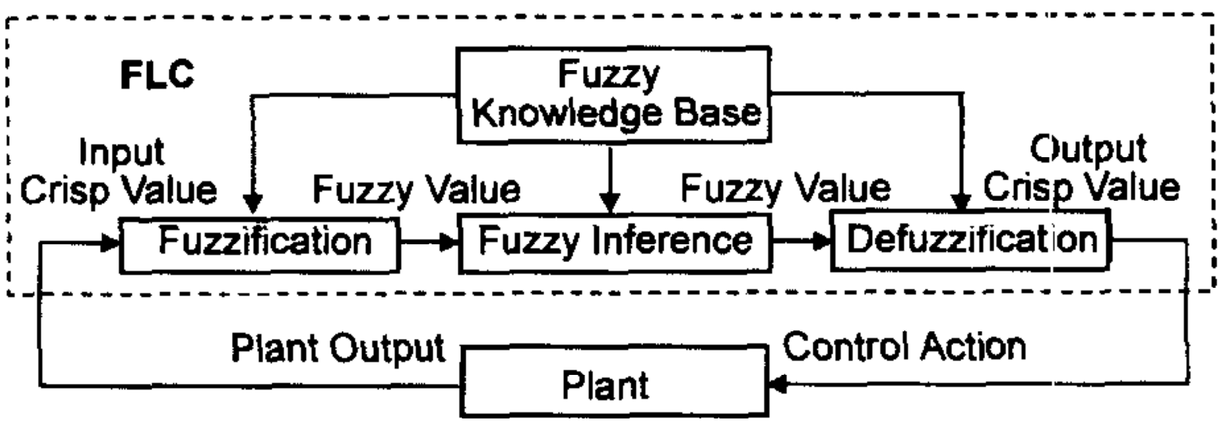


Fig.1 Configuration of FLC

Currently, from the enormous literatures, we know that the Mamdani model and TS model have become the dominant ones for FLC systems. Therefore, F200 chip will be designed to support these two fuzzy models. In this section, we will concentrate on the fuzzy control inference methods of F200 chip, which includes the fuzzy inference, fuzzification and defuzzifivation. Note that two fuzzy models have the same fuzzification method.

2.1 Fuzzification

Assume the antecedent of a fuzzy rule, "X is A," and the real input condition, "X is A'," where A, A' are linguistic values (or fuzzy labels), i.e. membership functions. For FLC, A' often uses the form of a fuzzy singleton $\mu_{A'}$, that is the membership grade $\mu_{A'}(x_0) = 1$, and $\mu_{A'}(x) = 0, x \neq x_0$, where x_0 is the real crisp input value. If μ_A is the membership function A, then the fuzzification means that takes the intersection of $\mu_A(x_0)$ and $\mu_{A'}(x_0)$ as the fuzzy value of input x_0. Usually the intersection is the min operation. Therefore, $\min(\mu_A(x_0), \mu_{A'}(x_0)) = \mu_A(x_0)$, which represents the degree of similarity of A' to A.

The fuzzification operation depends on the description of membership functions. For the table description, the function is described by a set of two coordinate values. Thus the fuzzification simply looks up the table with a small overhead. However, it needs more memory capacity to store the functions. So here, we adopt the parameter description. F200 chip supports 12 shapes of input membership functions, such as singleton, step-up, step-down, rectangle, S-shape, Z-shape, triangle, and trapezoid. For the latter four shapes, their bevel edges may be replaced by broken ones in order to form their extension shapes. These can be used to represent more

kinds of functions, such as the approximate Gauss distribution (bell shape). Each membership function is described by a set of parameters, x-coordinate values and slopes of bevel edges. Then the input fuzzification is simply to evaluate the membership function value for a given input value. For an example of a trapezoid extension shape (Fig.2), the parameters will be coordinate values $p_0, p_1, ..., p_5$ and slopes s_0, s_2, s_3, s_5. Given the input value x, the corresponding fuzzy value μ can be obtained by the following procedure.

if $(x - p_0 \geqslant 0)$ μ=0
else if $(x - p_1 \leqslant 0)$ $\mu = 0$
else if $(x - p_2 \leqslant 0)$ $\mu = (x - p_1) * s_2$
else if $(x - p_3 \leqslant 0)$ $\mu = 1 - (p_3 - x) * s_3$
else if $(x - p_4 \leqslant 0)$ $\mu = 1$
else if $(x - p_5 \leqslant 0)$ $\mu = 1 - (x - p_4) * s_5$
else $\mu = (p_0 - x) * s_0$

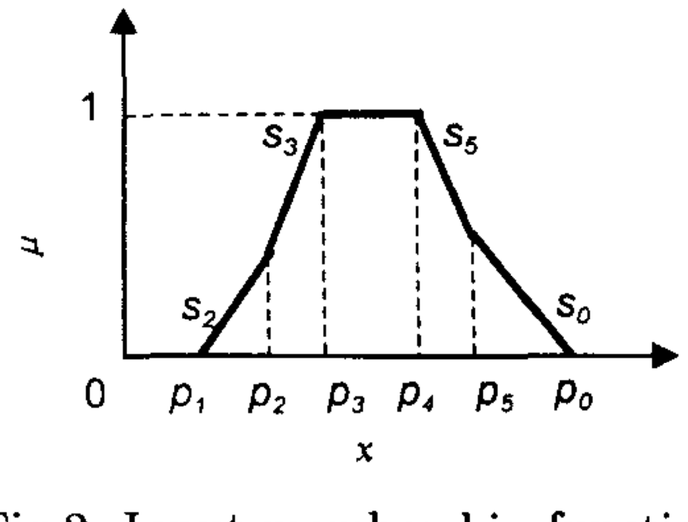


Fig.2 Input membership function with trapezoid extension shape

For each input variable, usually one can define several membership functions on the universe of discourse of the input. Since membership functions can be overlap, so one may obtain more than one fuzzy value μ for a given input value x. Thus, the fuzzification need search all membership functions on the x-coordinate axis to calculate multiple fuzzy values. The simplest method is searching from the primary point of the x-axis, which is adopted by F200. However, if the given input value x is very large, it will take more time to search. A possible improvement is that divide the x-axis into multiple regions to have more start points for searching.

2.2 Fuzzy Inference and Defuzzification

2.2.1 Mamdani Model

Consider MISO (Multiple Inputs and Single Output) fuzzy rule as follows.

If $(X_1$ is $A_1)$ and $(X_2$ is $A_2)$ and ... Then Y is C.

where X_i, Y are linguistic variables, A_i, C are fuzzy labels (linguistic values). Usually, different rules may have the same consequence label C. Therefore, we can divide the whole rule set into several rule groups according to the value C. Then the fuzzy inference process consists of four steps (Fig. 3)[7].

1) Evaluate the input firing strength α_j of the rule j

$$\alpha_j = (\wedge_i \mu_i) * RW_j,$$

where μ_i is the fuzzy value (membership grade) of the input i, RW_j is the weight of the rule j,

2) Evaluate the input firing strength α_{gk} of the rule group k

$$\alpha_{gk} = \vee_j \alpha_j,$$

3) Evaluate the output inference result C'_{gk} of the rule group k

$$\mu_{C'_{gk}}(w) = \alpha_{gk} \wedge \mu_{C_{gk}}(w),$$

where w is the universe of discourse of the output (crisp), C_{gk} is the correspoding output membership function (label),

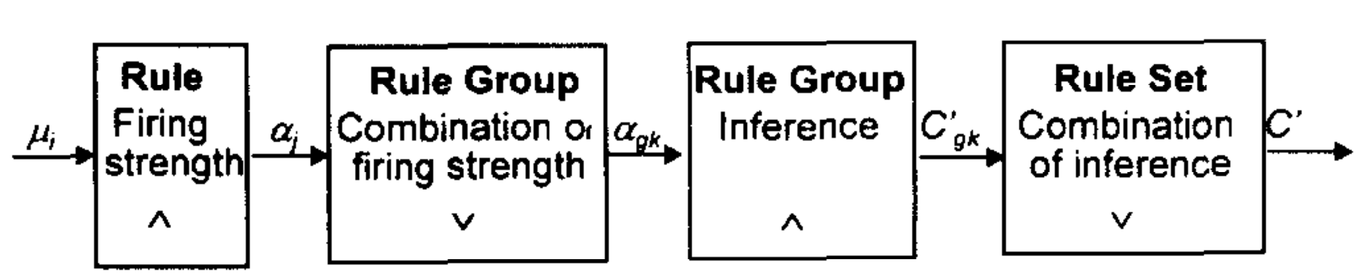


Fig.3 Fuzzy inference process

4) Evaluate the combination of output inference results C' of the rule set

$$\mu_{C'}(w) = \vee_k \mu_{C'_{gk}}(w).$$

The first two steps of the inference process correspond to the condition part of a rule. Its fuzzy operator $(\vee - \wedge)$ often use (min-max) for the simplicity of the hardware implementation.

In F200 chip, except of (min-max), we adopt one more important operator (product-bounded sum). The membership grade μ has the value of range [0, 1], usually defined as a fixed-point integer. However, for the product operation of μ, we have to use a float-point variable.

The last two steps correspond to the implication part of a rule. How to implement is related to the defuzzification. And they often performed with the output defuzzification. One possible method is that in above formulas, the value w takes 1 as its increment each time[5]. In this case, output membership functions are usually described by tables of coordinate values. Though it has more flexibility for the defuzzification, it needs the large overhead of the memory and performing time. In fact, for FLC applications, the defuzzification only need to consider the main parameters of a output membership function shape, its center location and area. We define two shapes, the singleton and isosceles triangle for F200 chip, by the parameter description. Then the last two steps can be directly evaluated based on the formulas in [10] with a small overhead. Furthermore, the chip supports two defuzzification methods as follows.

A) Center of Gravity (COG)

We can obtain the output crisp value y using the following formulas. Note that for each output, all membership functions have the same shape.

i) For singleton function $y = \dfrac{\sum_k \alpha_{gk} * p_k}{\sum_k \alpha_{gk}}$, where p_k is the location of singleton,

ii) For isosceles triangle function with $(\vee - \wedge)$ as (min-sum) implication

$$y = \frac{\sum_k \alpha_{gk} * (2 - \alpha_{gk}) * b_k * p_k}{\sum_k \alpha_{gk} * (2 - \alpha_{gk}) * b_k},$$

iii) For isosceles triangle function with $(\wedge - \vee)$ as (product-sum) implication

$$y = \frac{\sum_k \alpha_{gk} * b_k * p_k}{\sum_k \alpha_{gk} * b_k},$$

where p_k is the center location of the triangle and b_k is the bottom edge length of the triangle.

B) Mean of Maximums (MOM)

The output crisp value takes the center location of the membership function C_{gk} corresponding to the maximum α_{gk}. If there are n same maximums α_{gk}, then $y = \sum_k p_k/n$. Note that for MOM defuzzification, the output membership function only needs the singleton.

2.2.2 TS Model

Assume MISO fuzzy rule as follows.

If $(X_1$ is $A_1)$ and $(X_2$ is $A_2)$ and ... Then $z_j = \sum_i p_i * x_i$.

In the consequence, the output z_j is represented by a function of crisp input x_i. The coefficient p_i has the value of the range [-1, +1]. During the inference process, for each rule, it evaluates the input firing strength $\alpha_j = \wedge_i \mu_i$ and output z_j, where the fuzzy operator $\wedge$ can be min or product. Then the combination of output inference results y is calculated by the weighted mean method.

$$y = \frac{\sum_j \alpha_j * z_j}{\sum_j \alpha_j}.$$

3 Fuzzy Coprocessor F200

3.1 Chip Structure

The fuzzy coprocessor F200 is a general-purpose fuzzy controller mainly for the industry real-time process control. Since the performance is not a main goal, the chip is more simply designed as a coprocessor. It will be easy and flexible to construct a control system with a standard CPU chip. And the knowledge base memory is separate from F200. The block diagram of the chip is shown in Fig. 4. It includes an arithmetic unit, on-chip I/O memory and two interfaces to outside. The arithmetic unit has an adder and a multiplier to perform all

fuzzy operations. The I/O memory is implemented with latches for storing input/output variables, and working cells. The F200 chip has the following design parameters.

1) number of input variables: 16,
2) number of output variables: 8,
3) number of outputs directly to inputs: 7,
4) number of input fuzzy labels : 15,
5) overlap degree of input labels: 3,
6) number of output fuzzy labels : 16,
7) number of antecedents of a rule: 1∼16,
8) number of consequences of a rule: 1∼ 2,
9) number of knowledge bases: 1∼ 256,
10) clock frequency: 20 MHz.

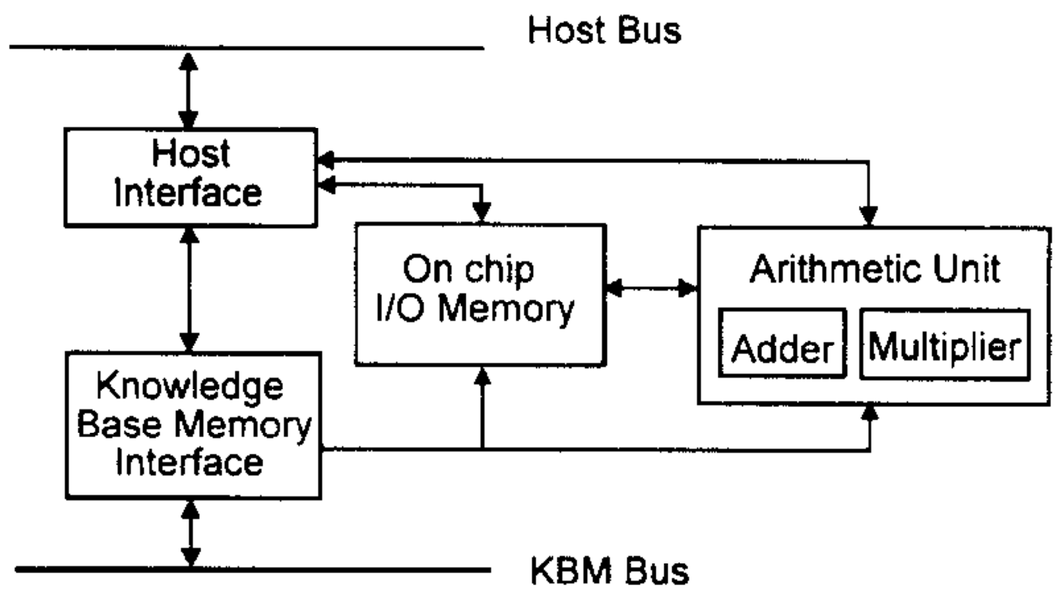


Fig.4 Block diagram of F200 chip

3.2 Variable and Scale Factor

In the F200 chip, the fuzzy inference process actually shows a series of conversions and reverse conversions of variables (Fig. 5). The purpose of scale factors integrated is that while uses the fixed word length inside the chip, we can define different lengths for external I/O variables. So the higher resolution will be guarantee. The types of variables are defined as follows.

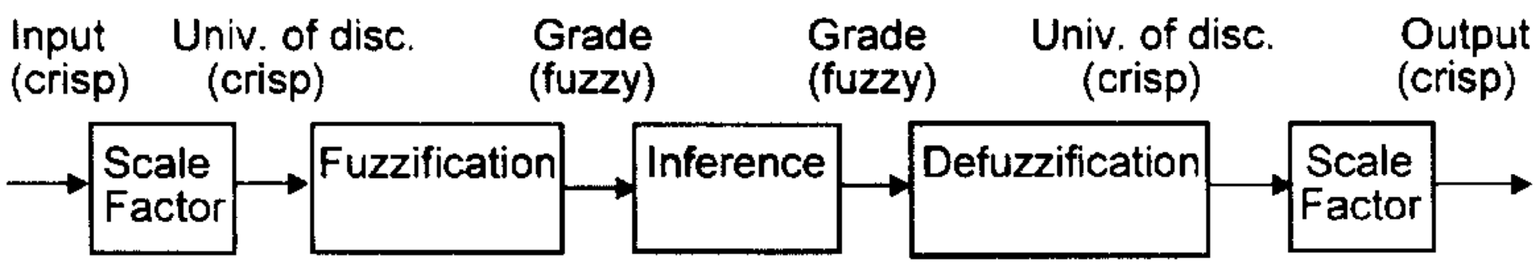


Fig.5 Variable conversions

1) External I/O variable: 16 bits, support the unsigned and signed integer (complement code).

2) Universe of discourse (U): 12 bits, unsigned integer.

3) Membership grade: 12bits, unsigned float-point with 8bits mantissa and 4bits exponent.

Membership grade	Mantissa (8 bit)	Exponent (4 bit)
1	255	20
0	0	2^0 or 2^{-52}

Exponent	1111	1110	1101	...	0000
Value	2^8	2^4	2^0	...	2^{-52}

There is a mapping between the external variable (V) and the universe of discourse (U). By use of scale factors, we can map the concerned part of V to U or vice-versa.

1) Unsigned integer mapping (Fig.6a)

$$V \text{ to } U: \quad U = \frac{D}{V_{\max} - V_{\min}} * (V - V_{\min}) = KI * (V - V_{\min}),$$

$$U \text{ to} V: \quad V = \frac{V_{\max} - V_{\min}}{D} * U + V_{\min} = KO * U + V_{\min},$$

where, KI is input scale factor, KO is output scale factor, $V_{\min}$ is lower bound of V, $V_{\max}$ is upper bound of V, D is maximum of U, $D = 2^{12} - 1$.

2) Signed integer mapping (Fig.6b)

$$V \text{ to } U: \quad U = \frac{Ds}{V_{\max}} * V = KI * V,$$

$$U \text{ to } V: \quad V = \frac{V_{\max}}{Ds} * U = KO * V,$$

where, maximum of U, $Ds \approx 0.5 * D$.

Since we use the complement code for signed integers of I/O variables, for simplicity, F200 will convert the signed integer to the unsigned one or vice-versa. That is to complement the sign bit or the most significant bit respectively. In that case, first, we define two parameters.

Base value:

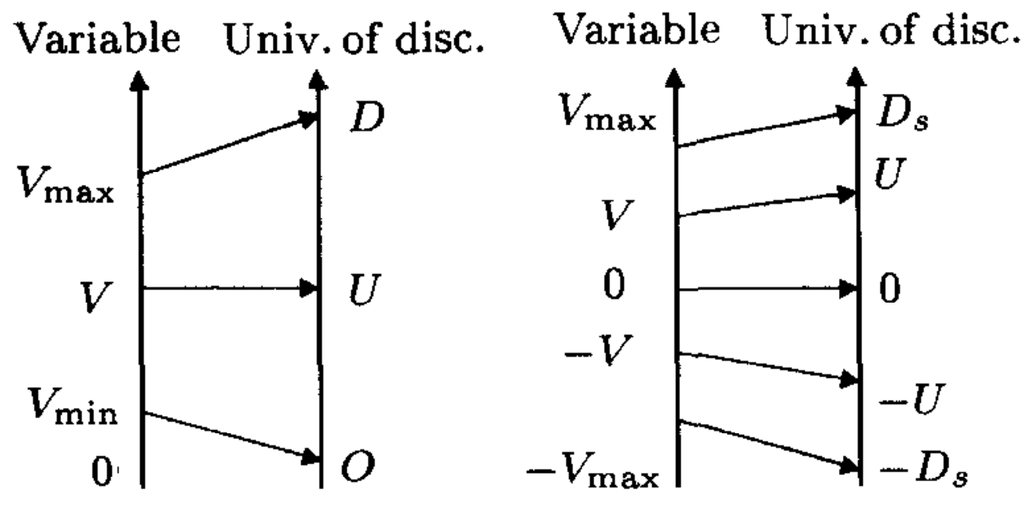
$$VB = \begin{cases} \min, & \text{for unsigned variables.} \\ 2^{15} - \max, & \text{for signed variables.} \end{cases}$$

Range of value: $$VR = \begin{cases} \max - \min, & \text{for unsigned variables.} \\ 2 * V_{\max}, & \text{for signed variables.} \end{cases}$$

Then the mapping between the converted internal I/O variable (V) and the universe of discourse (U) can be uniformly proceeded.

$$V \text{ to } U: \quad U = KI * (V - VB), \quad KI = (2^{12} - 1) * (1/VR),$$

$$U \text{ to } V: \quad V = KO * V + VB, \quad KO \approx 2^{-12} * VR.$$



(a) unsigned variable (b) signed variable

Fig.6 Variable mapping

3.3 Fuzzy Knowledge Base

The fuzzy knowledge base defines the process of the fuzzy control inference. The knowledge base memory is separated from F200 chip. The memory can be ROM or RAM. F200 supports 256 knowledge bases with one working base selected. Fig.7 shows the organization of the knowledge base memory, in which,

KBA——start addresses of knowledge bases.

MFA——start addresses of I/O membership function parameter tables.

RSA——start addresses of rule sets corresponding to outputs.

SF——scale factor table which includes scale factor parameters, variable types, and connections from outputs to inputs.

RSD——rule set description words.

MF——I/O membership function parameter tables.

RS—— rule sets.

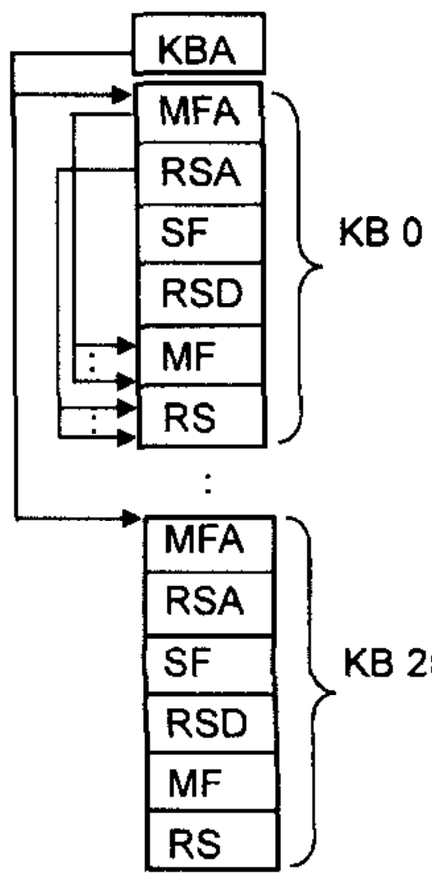


Fig.7 Organization of the fuzzy knowledge base memory

3.3.1 Rule Set Description Word

Each output or each pair of outputs corresponds to a rule set with the following description word (16bits).

4	4	1	1	2	2	2
-	NA	NC	MOD	FOP	DEFUZ1	DEFUZ2

1) NA——number of antecedents of a rule, 1∼16 for Mamdani model, and 1∼8 for TS model.

2) NC——number of consequences of a rule, 1∼2 for Mamdani model, and only one for TS model.

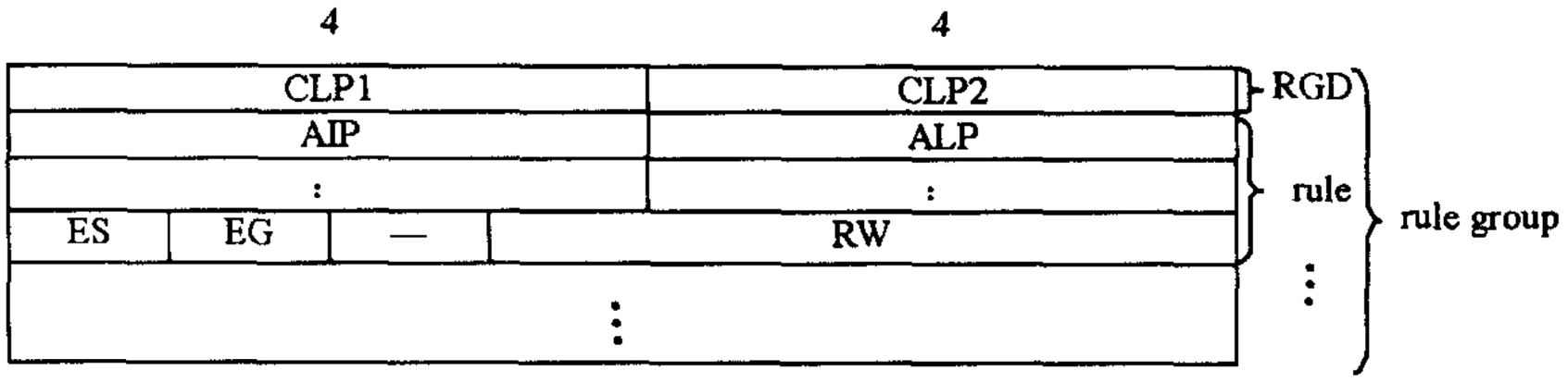
3) MOD——fuzzy model.

4) FOP——combination of fuzzy operators.

5) DEFUZi—— defuzzification method of 1∼2 outputs for Mamdani model only.

3.3.2 Rule Format of Mamdani Model

For Mamdani model, each rule set consists of several rule groups shown as follows.

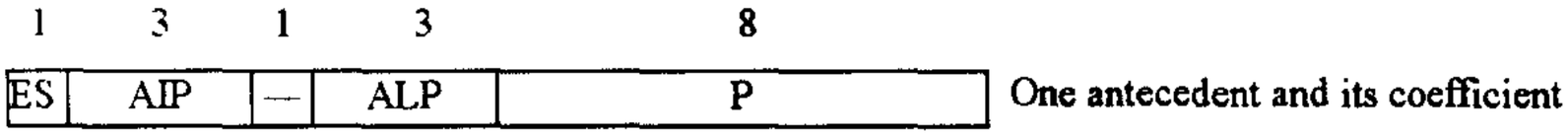


1) RGD——rule group description word, i.e. label pointers of consequences (CLP) of 1∼2 outputs.

2) rule——it includes antecedents (input pointer, AIP, and label pointer, ALP), rule weight RW, and end flag of the rule group, EG, and end flag of the rule set, ES.

3.3.3 Rule Format of TS Model

For TS model, no group is divided in the rule set. Each rule consists of several antecedents (AIP and ALP) and their corresponding coefficients, P, of consequence as follows.



3.4 Memory Addressing

F200 chip integrates two small memories by use of latches. One is the input memory (IMEM) with 16 words * 29 bits for storing input variables and their fuzzy values. Another is the output memory (OMEM) with 16 words * 16 bits for storing output variable and working cells. CPU can access the part of IMEM/OMEM, i.e. I/O variables, through the host interface. Furthermore, the chip has two ports to communicate with the host. The port 0 is used as the command/status register, and the port 1 is the data register. Therefore, CPU accesses the memory of F200 chip by a 6 bits address as follows.

Byte Address	Contents
00 ∼ 1F	input variables (16 words * 2 bytes)
20 ∼ 2F	output variables (8 words * 2 bytes)
30	port 0
31	port 2

3.5 Command and Status Word

F200 chip is a coprocessor. CPU accesses F200 by a set of commands.

1) RST Reset Chip,
2) SKB Select Knowledge Base,
3) SSFI Start Single Fuzzy Inference,
4) SMFI Start Multiple Fuzzy Inference,
5) PTI Put Input,
6) GTO Get Output,
7) WKB Write Knowledge Base,
8) RKB Read Knowledge Base,
9) TEST Testing,
10) HAIOM Host Access I/O Memory (implicit command).

where, WKB/RKB is used for CPU to down/up load the knowledge base memory, TEST is designed for chip testing. Furthermore, F200 chip has two status words for communication between CPU and F200.

3.6 Operation Mode

The main functions of F200 chip are the fuzzy inference and knowledge base management. The typical fuzzy control process is shown as follows.

1) Each set of sample data is preprocessed by CPU, then put it into F200 chip.

2) Start the fuzzy inference. First, all input data are fuzzified, then perform single or multiple output inferences and defuzzification one by one.

3) CPU gets the inference result from F200 chip.

F200 chip has two operation modes.

i) Command Mode. CPU accesses F200 chip through two ports only. All functions of the chip are performed by commands.

ii) Shared Memory Mode. In Section 3.4, all accessible addresses of F200 chip can be also seen as the CPU memory. That is so called the shared memory. In this mode, CPU accesses the F200 chip through all accessible addresses. For example, write/read I/O variables directly by the implicit command HAIOM, and call fuzzy inference engine by three commands, SKB, SSFI, SMFI, only. This mode may simplify the call for the F200 chip.

4 Chip Design

F200 was designed in a CMOS semi-custom style with Cadence. After schematic design entry and logic simulation, the test vectors for F200 were extracted from the simulation result and the netlist was transferred for automatic placement and routing. The standard cell library for 0.8 μm process of the Huajing Corporation in Wuxi was generated using the scalable CMOS cell-based design technology of the Microelectronics R&D Center, CAS. Therefore, the chip can be easily upgraded to 0.6 μm or 0.35 μm CMOS process. The main design features of F200 are given as in Table 1.

Table 1 The main design features

Process	Huajing 0.8 μm CMOS
Transistors	65,000
Die size	7.2 mm * 7.2 mm
Package	PLCC84
Signal pins	47
Clock	20 MHz
Power supply	5V

5 Conclusion

There are different techniques for the implementation of the fuzzy logic control, such as standard microcontrollers, general-purpose processor with specialized instructions for fuzzy tasks, dedicated fuzzy coprocessors, and dedicated fuzzy ASIC[11]. These approaches provide different degree of flexibility, speed and complexity. Basically, the first two approaches fall in the software implementation of FLC, which is simple and flexible, but slow speed. For the high-speed and complex systems, the hardware implementation will certainly be needed.

In this paper, we proposed the fuzzy inference coprocessor F200. It is a general-purpose fuzzy chip with powerful functionality. It can be used for construct fuzzy rule based systems, especially in fuzzy logic controllers. F200 supports multiple fuzzy knowledge bases and hierarchy control structure that one can construct more complex systems. The chip supports both Mamdani and TS models. As we know that it was the first time to implement the Takagi-Sugeno model on a fuzzy chip. F200 has multiple types of fuzzy rules, membership functions and fuzzy inferences. In addition, the input/output scale factors are also integrated on chip. It will be convenient to design controllers. F200 has the precision of 12 bits and its inference speed is 1.2 M fuzzy rules per second with 20 MHz frequency. Furthermore, the fuzzy chip can be used as a core and integrated with standard CPU, memories, etc., to realize the more powerful fuzzy system on a chip.

For the applications of F200, the corresponding fuzzy system development environment is needed. We have implemented a fuzzy control development system FDS2 on PC windows. It is an integrated development environment with the friendly user interface. Its function includes the membership function and fuzzy rule edit, fuzzy control system construction and simulation, design compilation to F200 object knowledge base, etc.

References

1 Mamdani E H. Application of fuzzy algorithms for the control of a dynamic plant. *Proc. IEE*, 1974, **121**(12): 1585~1588

2 Lee C C. Fuzzy logic in control systems: fuzzy logic controller —— Parts I, II. *IEEE Trans. on Systems, Man, and Cybernetics*,1990, **20**(2): 404~435

3 Togai M, Watanabe H. Expert system on a chip: an engine for real-time approximate reasoning. *IEEE Expert*, 1986, **1**(3): 55~62

4 Yamakawa T. Stabilization of an inverted pendulum by a high-speed controller hardware system. *Fuzzy Sets and Systems*, 1989, **32**(2): 161~180

5 Watanabe H, Dettloff W D, Yount K E. A VLSI fuzzy logic controller with reconfigurable cascadable architecture. *IEEE J. Solid-State Circuits*, 1990, **25**(2): 376~382

6 Eichfeld H, Kunemund T, Menke M. A 12b general-purpose fuzzy logic controller chip. *IEEE, Trans. on Fuzzy Systems*, 1996, **4**(4): 460~475

7 Shen L. Fuzzy logic control ASIC chip. *J. Computer Science and Technology*, 1997, **12**(3): 263~270

8 Pei S, Shen L, Ye L. Hierarchical fuzzy controller and its industrial application. In: Proceedings of the 1997 IEEE International Conference on Intelligent Processing Systems, Beijing China, 1997. 266~270

9 Takagi T, Sugeno M. Fuzzy identification of fuzzy model and its applications to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics*, 1985. **15**(1): 116~132

10 Shen L. Defuzzification method in fuzzy logic control. (in Chinese), In: Proceedings of the 1995 China Intelligent Automation Conference, Tianjing, 1995. 460~465

11 Costa A, De Gloria A, Faraboschi P, Pagni A, Rizzoto G. Hardware solutions for fuzzy control. *Proceedings of the IEEE*. 1995, **83**(3): 422~434

模糊推理协处理器芯片

沈 理[1] 朱亚江[2] 徐慧娥[1] 陈晓东[2]

[1](中国科学院计算技术研究所 北京 100080)
(E-mail: lshen@gsc.ict.ac.cn)

[2](中国科学院微电子中心 北京 100029)

摘 要 模糊推理协处理器 VLSI 芯片 F200 采用 0.8 μm CMOS 工艺，作为一种模糊控制器，主要用于实时过程控制和其它适合的应用场合，例如机器人控制、分类器、专家系统等. F200 芯片支持多个模糊知识库工作，支持最常用的两种模糊模型， Mamdani 和 Takagi-Sugeno 模型. 芯片精度 12 位，主频 20MHz, 推理速度约为每秒 1.2M 条模糊规则.

关键字 模糊控制，模糊推理，模糊协处理器， VLSI 芯片.

沈 理 中国科学院计算技术研究所研究员，博士生导师， IEEE Senior Member. 1959 年毕业于浙江大学电机工程系. 研究方向：计算智能，模糊系统，故障测试，容错计算， SOC 设计.

朱亚江 中国科学院微电子中心研究员. 1985 年获北京大学计算机系学士学位， 1988 年获中国科学院半导体研究所硕士学位. 研究方向：专用集成电路、数字模拟混合电路设计.

徐慧娥 中国科学院计算技术研究所高级工程师.

陈晓东 中国科学院微电子中心研究员，中国电子学会高级会员. 1966 年毕业于中国科技大学技术物理系. 研究方向： DSP 、深亚微米百万门级门阵和标准单元集成电路设计.