Single Board System for Fuzzy Inference

James R. Symon Hiroyuki Watanabe
Department of Computer Science
CB# 3175 Sitterson Hall
University of North Carolina
Chapel Hill, NC 27514-3175
TEL. (919) 962-1817, 962-1893

Abstract

The VLSI implementation of a fuzzy logic inference mechanism allows the use of rule-based control and decision making in demanding real-time applications such as robot control and in the area of command and control. We have designed a full custom VLSI inference engine. The chip is fabricated using 1.0 μ CMOS technology. The chip consists of 688,000 transistors of which 476,000 are used for RAM memory.

The fuzzy logic inference engine board system incorporates the custom designed integrated circuit into a standard VMEbus environment. The Fuzzy Logic system board uses TTL logic parts to provide the interface between the Fuzzy chip and a standard, double height VMEbus backplane allowing the chip to perform application process control through the VMEbus host. High level C language functions hide details of the hardware system interface from the applications level programmer. The first version of the board was installed on a robot at Oak Ridge National Laboratory in January of 1990.

1 Introduction

Fuzzy logic based control uses a rule-based expert system paradigm in the area of real-time process control [4]. It has been used successfully in numerous areas including train control [12], cement kiln control [2], robot navigation [6], and auto-focus camera [5]. In order to use this paradigm of a fuzzy rule-based controller in demanding real-time applications, the VLSI implementation of the inference mechanism has been an active research topic [1, 11]. Potential applications of such a VLSI inference processor include real-time decision-making in the area of command and control [3], and control of precision machinery.

An original prototype experimental chip designed at AT&T Bell Labs [7] was the precursor to the fuzzy logic inference engine IC that is the heart of our hardware system. The current chip was designed at the University of North Carolina in cooperation with engineers at the Microelectronics Center of North Carolina (MCNC) [8]. MCNC fabricated and tested fully functional chips.

The new architecture of the inference processor has the following important improvements compared to previous work:

1. programmable rule set memory
2. on-chip fuzzifying operation by table lookup
3. on-chip defuzzifying operation by centroid algorithm
4. reconfigurable architecture
5. RAM redundancy for higher yield

The fuzzy chips are now incorporated in VMEbus circuit boards. One of the boards was designed for NASA Ames Research Center and another board was designed for Oak Ridge National Laboratory (ORNL). The latter board has been installed and is currently performing navigational tasks on experimental autonomous robots [9].

ORNL will soon receive the second version of the board system featuring seven Fuzzy chips in a software reconfigurable interconnection network. The network provides host and inter-chip I/O in any logical configuration of the seven chips.

2 Fuzzy Inference

The inference mechanism implemented is based on the compositional rule of inference for approximate reasoning proposed by Zadeh [13]. Suppose we have two rules with two fuzzy clauses in the IF-part and one clause in the THEN-part:

Rule 1: If (x is A_1) and (y is B_1) then (z is C_1),
Rule 2: If (x is A_2) and (y is B_2) then (z is C_2).

We can combine the inference of the multiple rules by assuming the rules are connected by OR connective, that is Rule 1 OR Rule 2 [7]. Given fuzzy proposition (x is A') and (y is B'), weights α_i^A and α_i^B of clauses of premises are calculated by :

$$\alpha_i^A = \max_x(A', A_i),$$
$$\alpha_i^B = \max_y(B', B_i), \quad for \quad i = 1, 2.$$

Then, weights w_1 and w_2 of the premises are calculated by :

$$w_1 = \min(\alpha_1^A, \alpha_1^B),$$
$$w_2 = \min(\alpha_2^A, \alpha_2^B),$$

Weight α_i^A represents the closeness of proposition (x is A_i) and proposition (x is A'). Weight w_i represents similar measure for the entire premise for the i^{th} rule. The conclusion of each rule is

$$C_i' = \min(w_i, C_i), \quad for\ i = 1, 2.$$

The overall conclusion C' is obtained by

$$C' = \max(C_1', C_2').$$

This inference process is shown in Figure 1. In this example, $\alpha_1^A = 0.5$ and $\alpha_1^B = 0.25$, therefore $w_1 = 0.25$. $\alpha_2^A = 0.85$ and $\alpha_2^B = 0.5$, therefore $w_2 = 0.5$.

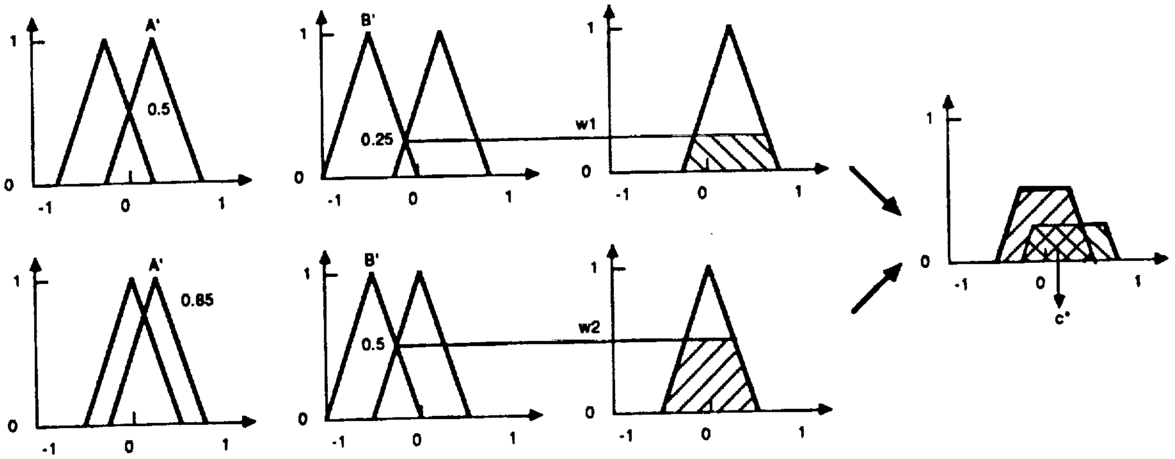

Figure 1: Inference.

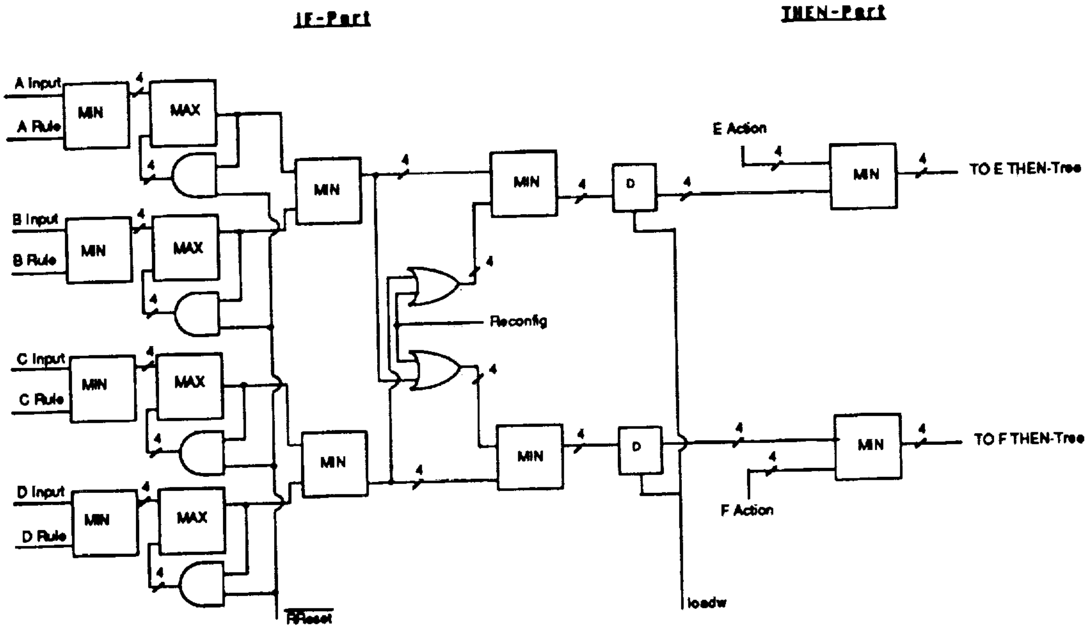

Figure 2: Fuzzy Chip Datapath.

3 Fuzzy Chip

The fuzzy logic inference engine is a fully custom designed 1.0 micron CMOS VLSI circuit of 688,000 transistors implementing a fuzzy logic based rule system. Included on chip are a programmable rule set memory, an optional input fuzzifying operation by table lookup, a minimax paradigm fuzzy inference processor, and an optional output defuzzifying operation using a centroid algorithm. The standard data path configuration is shown in Figure 2. The design has a reconfigurable architecture implementing either 50 rules with 4 inputs and 2 outputs, or 100 rules with 2 inputs and 1 output. Separately addressed status registers allow programmed control of the fuzzy inference processing and chip configuration. All the rules operate in parallel generating new outputs over 150,000 times per second.

The chip has 12 bidirectional data pins and 7 address pins for rule memory I/O. For process-control I/O, each of 4 inputs and 2 outputs has 6 pins. Each of 4 inputs has a corresponding load pin. The chip also has several control signals. Control signals **RW**(read high write low) and **CEN** (chip enable) are similar to that of a memory chip.

4 The System Boards

4.1 Single Chip Systems

The Fuzzy Logic system boards place the Fuzzy chip into a VMEbus environment to provide application process control through a VMEbus host. The single chip system designed for NASA Ames Research Center uses an off-the-self VMEbus prototyping board [10]. The overall configuration of the design is shown in Figure 3. In this design, the VMEbus interface is provided by the prototyping board system and needed a minimum of design for integration of the fuzzy chip. The fuzzy chip interface to the board is realized using discrete TTL parts and wire-wrapping. In the board system for ORNL, the VMEbus interface was designed by the first author and realized using a programmable logic device (PLD) and TTL parts. More robust printed circuit board (PCB) technology was used. The PCB architectural concept is shown in Figure 4. The UNIX device driver interfaces of these two boards are quite similar.

The ORNL board is designed to standard VMEbus specifications for a 24 bit address, 16 bit data, slave module as found in *The VMEbus Specification*, Revision C.1, 1985. It provides digital communication between the host and the Fuzzy chip. A large, UV erasable PLD generates the board control signals. VMEbus interface is through TTL parts. One Fuzzy Inference IC processes four 6-bit inputs to generate two 6-bit outputs. The interface with the host computer uses memory mapping to include the Fuzzy chip's I/O addresses in the application process storage space. All of the chip's memory as well as its inputs and outputs are accessed through addresses on the VMEbus so that the entire Fuzzy Logic board system responds like a section of memory.

The board's address space is 1024 bytes or 512 16-bit words in length. Most of the addresses in that space are not used by the board. The lower 128 word addresses of the board are mapped into the fuzzy chip. One hundred addresses are for rule memory. Another six addresses are mapped to four *fuzzification* tables and two status registers. The board has six addresses for I/O for the fuzzy chip, and addresses for hardware reset and board ID. On-board dip switches and signal jumpers allow the user to select the board base address comprised of the upper 14 bits of the 24 bit address, and the board's user privilege response characteristic determined by the VMEbus *address modifier* bits. Further design details are shown in Figure 5.

4.2 Multiple Fuzzy Chip System

The second version of the system board keeps the standard VMEbus interface of the first version but adds significant new capabilities. Seven Fuzzy chips communicate with each other and the host through a software reconfigurable interconnection network. Two Texas Instruments digital crossbar switch IC's

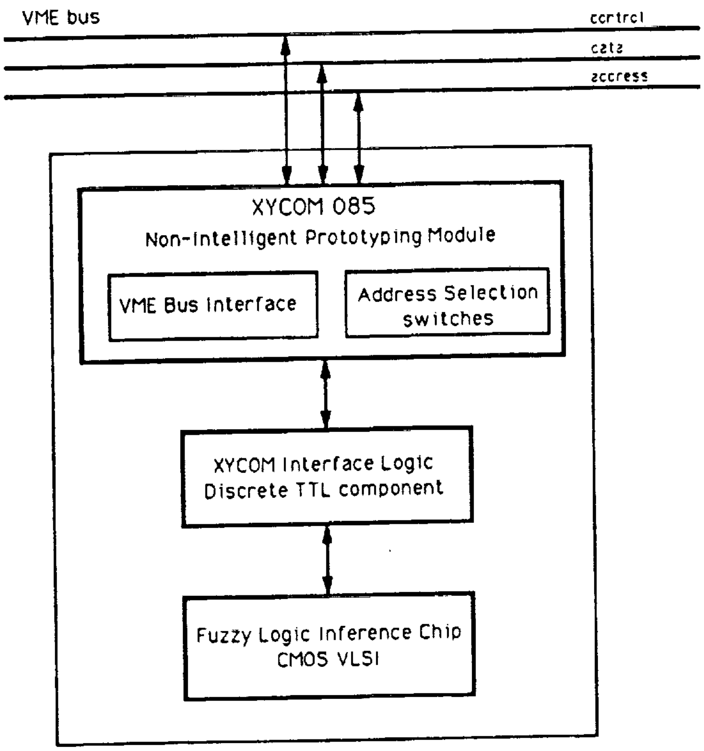

Figure 3: Single Chip System Based On Prototyping Board.

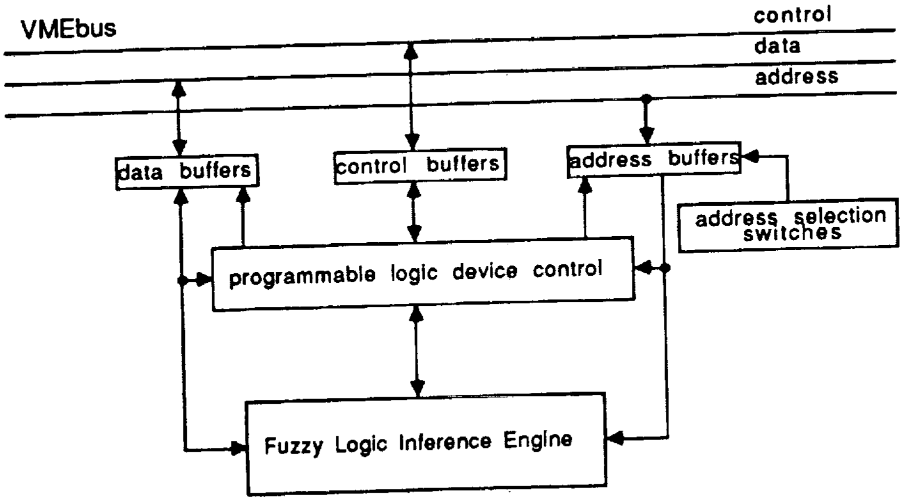
VMEbus
control
data
address
data buffers
control buffers
address buffers
address selection switches
programmable logic device control
Fuzzy Logic Inference Engine

Figure 4: Single Chip System Based On Custom PCB.

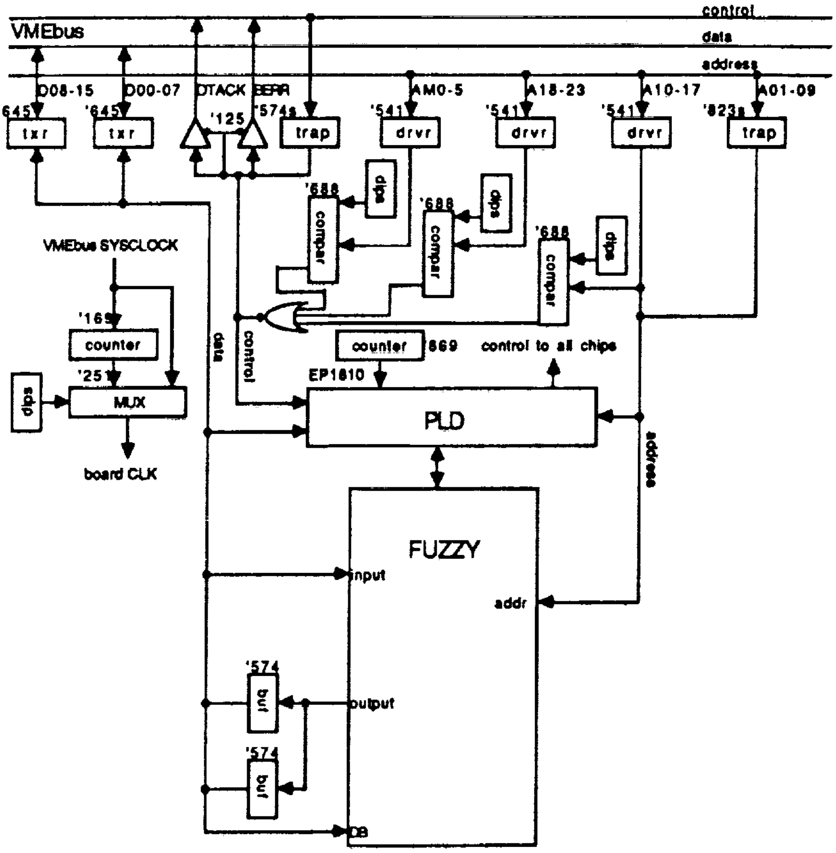

Figure 5: Details of PCB Architecture

implement the network. Any logical configuration of the seven chips may be specified in software, e.g. seven in parallel, 4-2-1 binary tree, etc. Any fuzzy output may be routed to any input. With the new board more inputs may be processed and hierarchies of rule sets may be explored. We can simulate rules with up to 16 conditions in the IF-part by using three layers of Fuzzy chips. Another application is to load multiple rule sets for different tasks in a single board. This is done by configuring multiple chips in parallel. The new printed circuit board architectural concept is shown in Figure 6.

This arrangement exploits an important feature of the Fuzzy chip. Normal input to the chip is by 6-bit integers which the chip *fuzzifies* into 64-value membership functions to be fed into the processing pipeline. The final output membership function is *defuzzified* into a 6-bit output integer. However, the chip has another mode of operation. Any input or output can bypass the [de]fuzzification process so that I/O occurs in *streaming* mode. The full 64-value input or output membership function is placed on the pins, one value per clock cycle. When an output of one chip is connected to an input of another chip (or itself), communication can be done in streaming mode without the loss of information inherent in the [de]fuzzification operations. On this system board, all inter-chip communication is done in streaming mode.

The new board also has four 64-value FIFO queues which allow final output to the host to be done in streaming mode. The application process is then free to perform its own custom operations on the full output membership functions. The final defuzzification is no longer limited to a centroid method. One can, also, generate the result in higher precision than 6 bits if necessary.

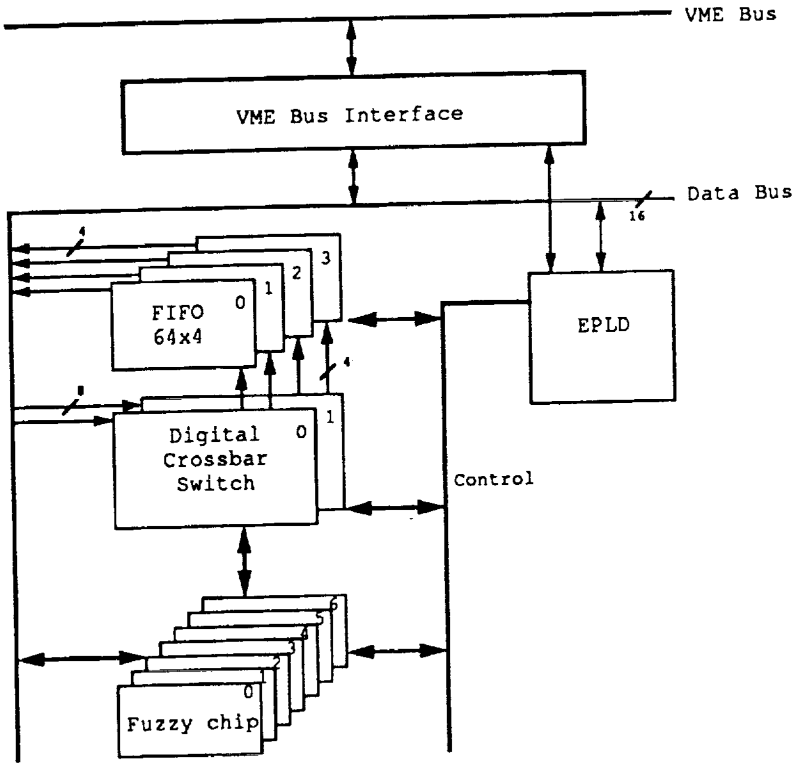

Figure 6: Seven Chip System Architecture.

The new board will be installed at ORNL in August, 1990. In addition to navigational tasks the system will be used to explore fuzzy logic control of manipulator arm functions.

5 Software Interface

High level C language functions can hide the operational details of the board from the applications programmer. The programmer treats rule memories and fuzzification function memories as local program structures passed as parameters to the C functions. Similarly, local input variables pass values to the system and outputs return in local variable function parameters. Programmers are only required to know the library procedures. Some procedures provided for the version 1 board are described in the following table.

1. *WriteRule(rulenum, ruledata)* - The rule data structure pointed to by *ruledata* is written to the board.
2. *ReadRule(rulenum, ruledata)* - Reads back into *ruledata* the rule identified by *rulenum* currently stored in the chip.
3. *WriteFuzz(fuzznum, fuzzdata)* - Fuzzification table is written to the board.
4. *StartFZIAC(inpA, inpB, inpC, inpD)* - Four inputs are sent to the fuzzy board and inference processing will be started.
5. *ReadOut(outE, outF)* - Both outputs are read from the board. Inference process will be continued.

6. *StopFZIAC(outE, outF)* - Both outputs are read from the board. Inference process will be halted.

6 Summary

We have described the architecture and associated high level software of two VME bus board systems based on a VLSI fuzzy logic chip. In addition to operating in the robot at ORNL, the single chip board is installed on a Sun-3 workstation at the University of North Carolina for further research and software development. For example, it is useful to provide an X-window based user interface to this fuzzy inference board. The complex and flexible architecture of the multiple chip board will require more sophisticated support software to facilitate exploration of various hierarchical interconnection schemes.

7 Acknowledgements

The research reported here is supported in part by Oak Ridge National Laboratory, by MCNC Design Initiative Program, and by NASA Ames Research Center.

References

[1] Corder, R. J., "A High-Speed Fuzzy Processor," *Proc. of 3nd IFSA Congress*, pp. 379-381, August 1989.

[2] Holmblad, L. P. and Ostergaard, J. J., "Control of a Cement Kiln by Fuzzy Logic," *Fuzzy Information and Decision Processes* (eds. M. M. Gupta and E. Sanchez) pp. 389-399, 1982.

[3] Kawano, K., M. Kosaka, and S. Miyamoto, "An Algorithm Selection Method Using Fuzzy Decision-Making Approach," *Trans. Society of Instrument and Control Engineers*, Vol. 20, No. 12, pp. 42-49, 1984. (in Japanese)

[4] King, P. J. and E. H. Mamdani, "The Application of Fuzzy Control Systems to Industrial Processes," *Automatica*, Vol. 13, No. 3, pp. 235-242, 1977.

[5] Maeda, Y., "Fuzzy Obstacle Avoidance Method for a Mobile Robot Based on the Degree of Danger," *Proc. of NAFIPS'90*, pp.169-172, June 1990.

[6] Shingu, T. and E. Nishimori, "Fuzzy-based Automatic Focusing System for Compact Camera," *Proc. of 3nd IFSA Congress*, pp. 436-439, August 1989.

[7] Togai, M. and H. Watanabe, "An Inference Engine for Real-time Approximate Reasoning: Toward an Expert on a Chip," *IEEE EXPERT*, Vol. 1, No. 3, pp. 55-62, August 1986.

[8] Watanabe, H., W. Dettloff and E. Yount "A VLSI Fuzzy Logic Inference Engine for Real-Time Process Control," *IEEE Journal of Solid-State Circuits*, Vol.25, No.2, pp.376-382, April 1990.

[9] Weisbin, C.R., G. de Saussure, J.R. Einstein, and F.G. Pin, "Autonomous Mobile Robot Navigation and Learning," *Computer*, Vol.22, No.6, June 1989.

[10] XYCOM, *XVME-85 Prototyping Module Preliminary Manual*, 1984.

[11] Yamakawa, T. and T. Miki, "The Current Mode Fuzzy Logic Integrated Circuits Fabricated by the Standard CMOS Process," *IEEE Transactions on Computers*, Vol. C-35, No. 2, pp. 161-167, February 1986.

[12] Yasunobu, S. and S. Miyamoto, "Automatic Train Operation System by Predictive Fuzzy Control," in *Industrial Applications Of Fuzzy Control*, M. Sugeno (Ed), pp. 1-18, 1985.

[13] Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision-Making Approach," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SME-3, pp. No. 1, pp. 28-45, January 1973.