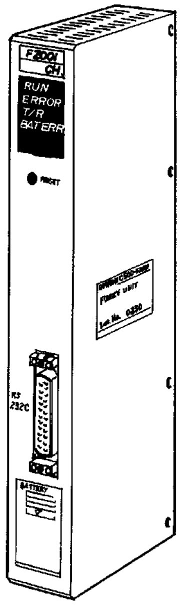
C500-FZ001 Fuzzy Logic Unit

Operation Manual

Cat. No. W209-E1-1A



Notice:

OMRON products are manufactured for use according to proper procedures by a qualified operator and only for the purposes described in this manual.

The following conventions are used to indicate and classify warnings in this manual. Always heed the information provided with them.

Caution Indicates information that, if not heeded, could result in minor injury or damage to the product.

DANGER! Indicates information that, if not heeded, could result in loss of life or serious injury.

OMRON Product References

All OMRON products are capitalized in this manual. The word “Unit” is also capitalized when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation “Ch,” which appears in some displays and on some OMRON products, often means “word” and is abbreviated “Wd” in documentation in this sense.

The abbreviation “PC” means Programmable Controller and is not used as an abbreviation for anything else.

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3... 1. Indicates lists of one sort or another, such as procedures, checklists, etc.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

About this Manual:

This manual describes the operation of the C500-FZ001 Fuzzy Logic Unit and includes the sections described below. The C500-FZ001 is programmed using the Fuzzy Support Software. Refer to the *Fuzzy Support Software Operation Manual* for details on Fuzzy Support Software.

Please read this manual completely and be sure you understand the information provide before attempting to operate the C500-FZ001.

Section 1 Introduction introduces the features of the Fuzzy Logic Unit and explains the fuzzy logic processing that takes place during operation.

Section 2 Components and Switch Settings describes the components of the C500-FZ001 and their functions.

Section 3 Memory Allocation explains how words in the PC are allocated to the C500-FZ001 and their functions as flags, control bits, and I/O words.

Section 4 Program and Knowledge Base explains briefly how to prepare the sub-program used in the PC and the knowledge base used in the C500-FZ001.

Section 5 Example Application describes the steps in the development of an actual control system that uses a C500-FZ001.

Section 6 Maintenance and Troubleshooting provides guidelines for routine maintenance and dealing with errors that might occur while using the C500-FZ001.

TABLE OF CONTENTS

SECTION 1
Introduction

This section introduces the features of the Fuzzy Logic Unit and explains the fuzzy logic processing that takes place during operation.

1-1 Features

The C500-FZ001 Fuzzy Logic Unit brings advanced, high-speed fuzzy logic technology to the C500, C1000H, C2000H, CV500, CV1000, CV2000, and CVM1 PCs. Control of some complicated systems that previously had to be supervised by a specialist can now be handled by these PCs.

Large I/O Capacity

The C500-FZ001 provides 8 inputs and 2 outputs, and each rule can have up to 5 condition and 2 conclusion parts, allowing it to be used in a wide range of applications.

Wide Variety of I/O Units

The user can select from the wide variety of Units that can be mounted on a C500, C1000H, C2000H, CV500, CV1000, CV2000, or CVM1 Backplane.

System Compatibility

The C500-FZ001 Fuzzy Logic Unit uses the I/O READ and I/O WRITE Instructions to communicate with the PC, so a fuzzy control system can be created if there is space for just one I/O Unit in the existing control system.

IBM-compatible Software

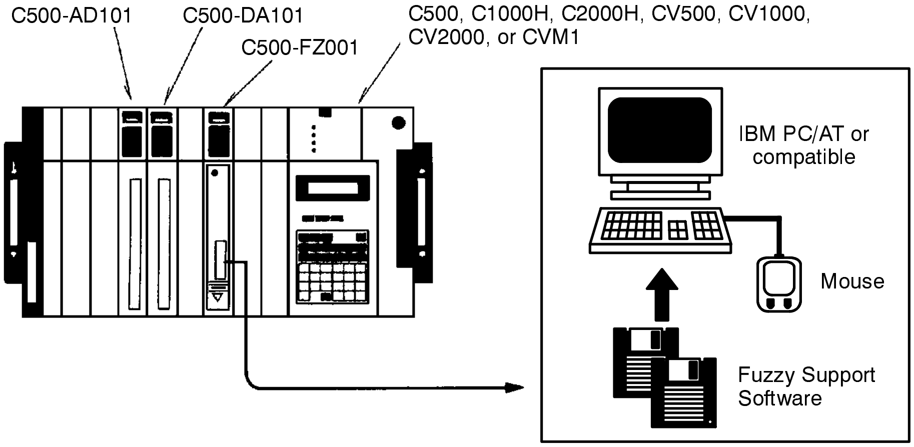
The Fuzzy Support Software (FSS), used to create and transfer the knowledge base as well as to monitor operation, runs on an IBM PC/AT or compatible computer. All of the computer operations can be controlled with pull-down menus, making the software easy to use, even for beginners.

1-2 Configuration

The C500-FZ001 can be mounted in any slot of a C500, C1000H, or C2000H CPU or Expansion I/O Rack, but not in a Slave Rack because the Unit transfers data to and from the CPU with the I/O WRITE (WRIT(87)) and I/O READ (READ(88)) instructions. The C500-FZ001 cannot be used with high-speed models of the C500 that do not support WRIT(87) and READ(88).

The Fuzzy Logic Unit can be mounted in any slot of a CV500, CV1000, CV2000, or CVM1 CPU, Expansion CPU, or Expansion I/O Rack, but not in a Slave Rack because it uses the I/O WRITE (WRIT(191)) and I/O READ (READ(190)) instructions.

The Fuzzy Logic Unit is often used together with the C500-AD101 Analog Input Unit and C500-DA101 Analog Output Unit, as shown in the following diagram.



1-3 Fuzzy Logic Theory

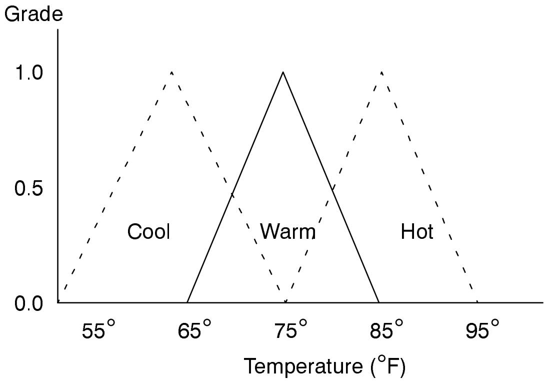
1-3-1 Introduction to Fuzzy Logic

Before fuzzy logic theory, modern control methods required a precise expression (either theoretical or numerical) of the relationship between inputs and outputs in a system. Instead of a precise mathematical formula, fuzzy logic expresses the relationship between inputs and outputs in a group of if/then statements based on human know-how and experience. In fuzzy logic theory the if/then statements are called **rules**. The “if” part of the rules describes possible system conditions, and the “then” part of the statements describes the appropriate action(s) for those conditions.

Creating a fuzzy control system with if/then statements is much, much simpler than expressing a control process using a set of mathematical equations, but the key to fuzzy logic theory isn’t the use of rules, it’s the assignment of a **grade** to those statements according to how well they fit the current system conditions. All of the rules are combined with their relative importance determined by their grade. Assigning grades to the statements allows computers to deal easily with the variations in the real world.

Let’s consider the concept of “warmth” to demonstrate how fuzzy theory assigns grades to conditions. “Warm” is an inexact, or fuzzy, concept. Most people will agree that 75° is warm, but wouldn’t say that a temperature of 70° or 80° isn’t warm. The feeling of “warm,” as opposed to “hot” or “cold,” just isn’t as strong at 70° or 80°.

Fuzzy theory assigns a grade to this feeling of warmth, and the function that relates the condition (feeling of warmth) and the grade is called the **membership function**. A set of values defined by a membership function is called a **fuzzy set**. The most accurate membership function is usually a bell curve, but triangular or trapezoidal shapes are used most often because they are easy to define and use. The following diagram shows a triangular membership function for “warmth.”



From this function, the computer can calculate the level of “warmth” for any temperature. A temperature of 75° has a grade of 1, while 70° and 80° have a grade of 0.5. In other words, a rule that uses “warmth” in its if statement (If it’s warm, then ...) will be weighted by a grade of 1 at 75°, or 0.5 at 70° or 80°.

The fuzzy logic processor calculates the grade for every rule according to the current conditions and then combines these results to yield a single result. The final result is affected by all of the rules that the programmer included.

Making If/Then Statements

The first step in creating a fuzzy control system is expressing the control process in if/then statements. The following statements express the know-how that people use unconsciously when they adjust the temperature of their bath.

If the water is very hot, turn the faucet far to the left.

If the water is hot, turn the faucet a little to the left.

If the water is warm, don’t turn the faucet.

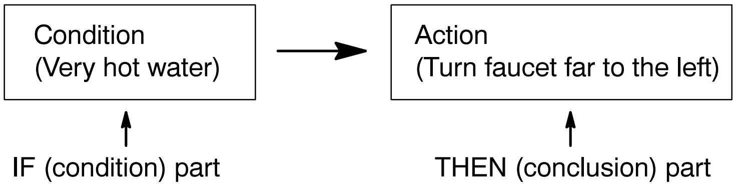
If the water is cool, turn the faucet a little to the right.

If the water is cold, turn the faucet far to the right.

A pattern emerges in these statements that shows about how much to adjust the faucet depending on the temperature of the water. The know-how that one uses when he adjusts the bath temperature has been converted to a form that can be entered into a computer.

Making the Rules

The "if" clauses are called the **condition parts** and the "then" clauses are called the **conclusion parts** of the rule.



The C500-FZ001 can have up to 128 rules with 5 condition and 2 conclusion parts each. More condition and conclusion parts allow the fuzzy control to handle more complicated control processes. For the bath example, a rule might have two condition parts such as "If the water is very hot and the air is cold," and two conclusion parts such as "turn the faucet far to the left and add a little water." The rules can be listed as shown below.

Condition parts:		Conclusion parts:
Aa1, Aa2, ...	→	Ca1, Ca2
Ab1, Ab2, ...	→	Cb1, Cb2
Ac1, Ac2, ...	→	Cc1, Cc2
⋮		⋮

Condition parts within one rule are linked by logical ANDs, i.e., all of the parts must be true. Rules are linked by logical ORs, i.e. any rule can be satisfied regardless of the status of other rules. These relationships are illustrated below.

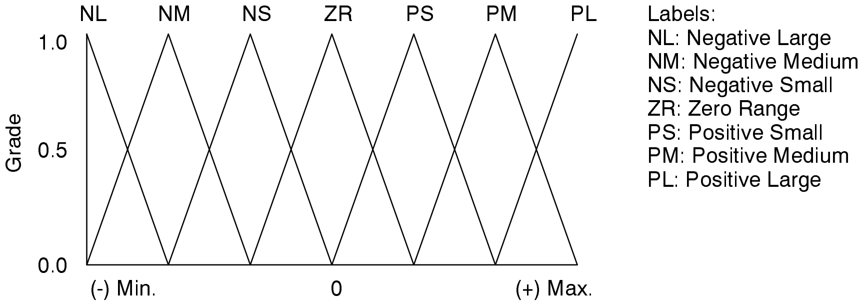
IF (Aa1 AND Aa2 AND ...) THEN (Ca1 AND Ca2)
OR
IF (Ab1 AND Ab2 AND ...) THEN (Cb1 AND Cb2)
OR
etc.

Membership Functions

The condition membership function assigns a numerical value to how well a specific value of a **fuzzy variable** (temperature, distance, speed, etc.) satisfies a condition (cool, warm, close, far, slow, fast, etc.) in a rule.

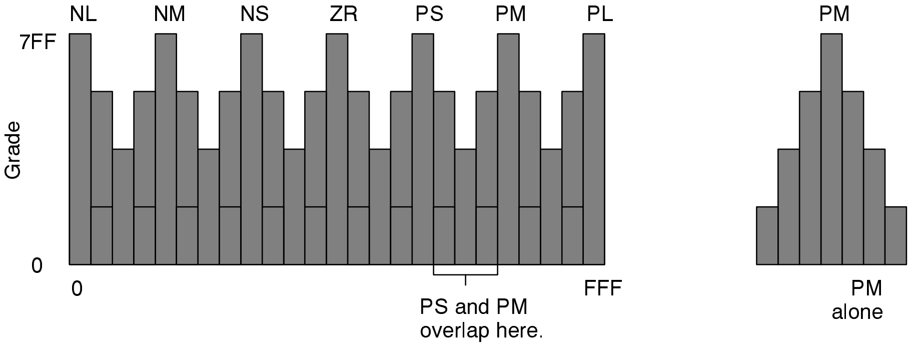
Condition membership functions can have four different shapes: S, Z, Λ, or Π. Refer to *1-3-3 Developing the Knowledge Base* for details on membership function shapes.

The C500-FZ001 can have up to 7 membership functions for each fuzzy variable. The conditions (cool, warm, close, far, slow, fast, etc.) defined by the membership functions are called **labels**. The following diagram shows a standard arrangement of Λ-type membership functions.



When creating membership functions for a particular system, modify this standard arrangement to suit the particular application. Refer to *1-3-3 Developing the Knowledge Base* for details on creating condition membership functions.

The conclusion membership function assigns a specific value to an action (turning a valve, raising voltage, etc.) indicated in the conclusion part of a rule. The shape of the conclusion membership functions is roughly triangular, made up of 4 columns for NL and PL and 7 columns for NM through PM. Each conclusion membership function overlaps the adjacent function(s), producing a total of 25 columns in the output range.

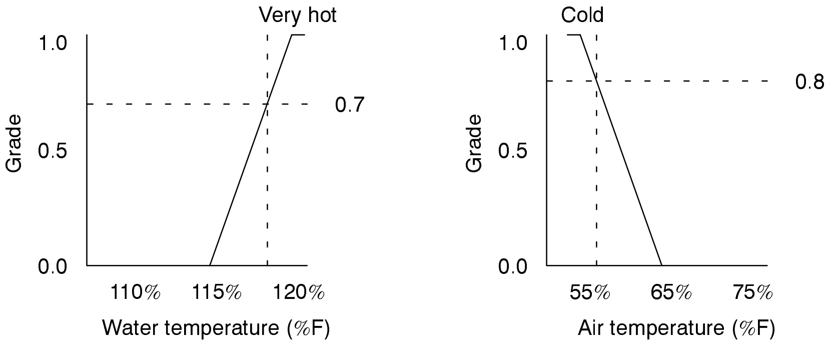


Fuzzy Logic Processing

This section briefly explains how the fuzzy logic processor calculates a result from the condition and conclusion parts of the rules that were expressed in terms of membership functions above. Refer to *1-3-2 Fuzzy Logic in the C500-FZ001* for a more detailed explanation.

First of all, the grades for the condition parts of each rule are calculated from the inputs and condition membership functions for each rule. The minimum grade of the condition parts is the **rule grade**. The minimum value is taken because condition parts of a rule are linked by logical ANDs, so all of the parts must be satisfied simultaneously.

For a rule with two condition parts such as “If the water is very hot and the air is cold,” the rule grade is the lesser of the two grades. In the case shown below, the rule grade is 0.7.



After all of the rule grades are calculated, the grade for each label is calculated. The grade for a label is called its **fuzzy output**. The fuzzy output is the maximum rule grade for that label. The maximum value is taken because the rules are linked by logical ORs. Taking the maximum value ensures that the results from all of the rules are taken into account fairly. The final result is calculated from the fuzzy outputs in an operation called defuzzification.

Defuzzification

The center of gravity method of defuzzification is used in the C500-FZ001. Each conclusion membership function is cut off above the fuzzy output and then the center of gravity of the 25 columns in the output range (on the x-axis) is calculated. The location of the center of gravity is the final result.

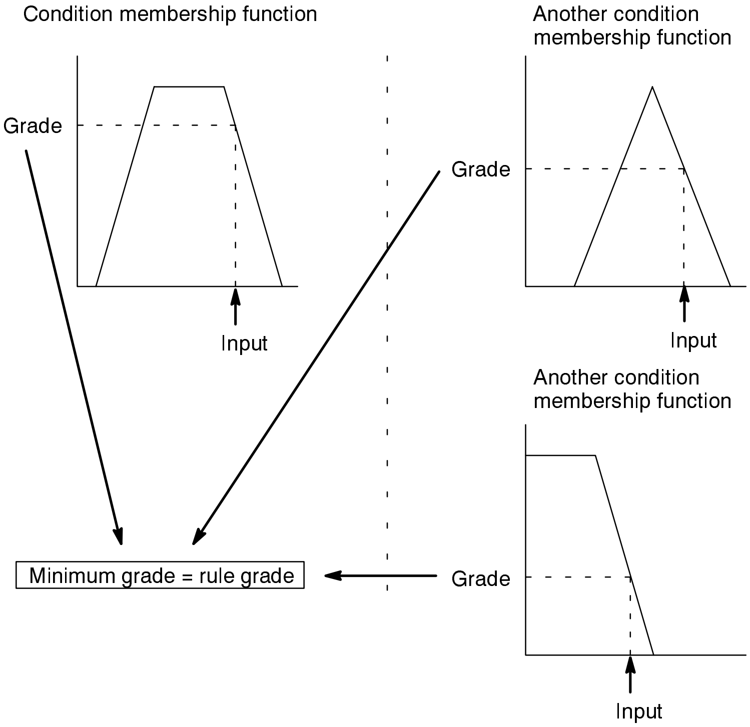
1-3-2 Fuzzy Logic in the C500-FZ001

In this section, the fuzzy logic processing that was outlined in the previous section is examined in detail. Fuzzy logic processing in the C500-FZ001 is divided into three steps: condition part processing, conclusion part processing, and defuzzification.

Condition Part Processing: Calculating Rule Grades

Condition part processing involves calculating the rule grades. This is a two-step process because one rule can have up to 5 condition parts. First, the grade for each condition part of a rule is calculated from its condition membership function and the inputs, then the minimum grade is taken as the rule grade.

The rule in the following example has 3 condition parts.

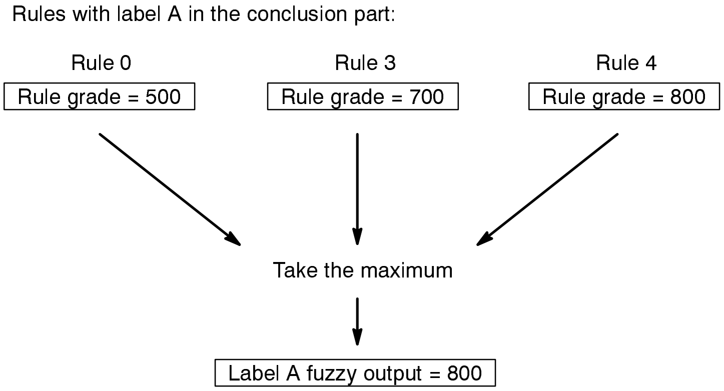

Condition membership function
Another condition membership function
Grade
Grade
Input
Input
Another condition membership function
Minimum grade = rule grade
Grade
Input

The rule grades, which are used in conclusion part processing, are stored in RAM and can be monitored with the Fuzzy Support Software.

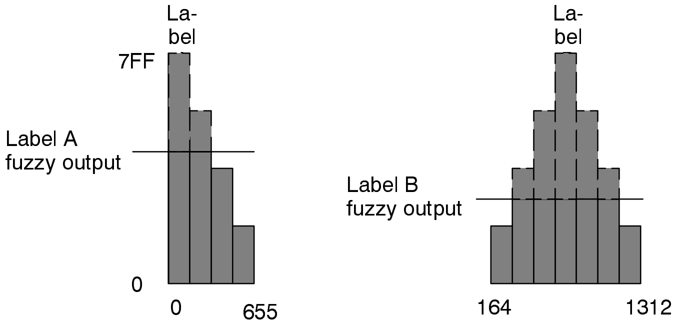
Conclusion Part Processing

Conclusion part processing involves calculating the fuzzy output for each label. The rule grades for all rules that have that label in their conclusion part are compared and the maximum is taken as the fuzzy output for that label.

In the following example rules 0, 3, and 4 affect label A.



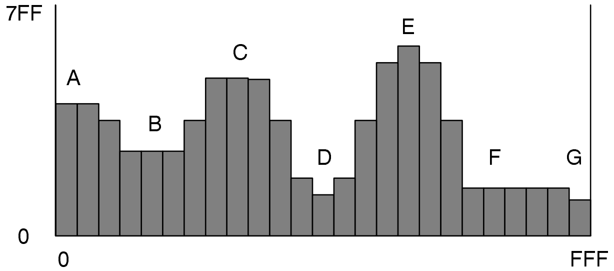
When the operation shown in the previous diagram is performed for all of the labels, each label will have a fuzzy output. The conclusion membership function for each label is cut off above the fuzzy output. Examples of this process for labels A and B are shown below.



Defuzzification

Defuzzification determines a final result by calculating the center of gravity of the conclusion membership functions cut off at the fuzzy output. The equation used to calculate the center of gravity along the x-axis, R, is shown below. The term S_n is the location of each of the 25 columns on the x-axis. F_n is the height of the column at S_n. When two columns overlap, the greater of the two is taken as F_n.

$$R = \frac{\sum_{n=1}^{25}(S_n \times F_n)}{\sum_{n=1}^{25}(F_n)}$$



If all of the fuzzy outputs are zero, the divisor in the equation above will also be zero, the division will not be performed and the Error Flag (bit 15) and No Corresponding Output Flag (bit 04) for that output will be turned ON. Refer to *3-1-2 Output Data Transfer* for details on the output data format.

1-3-3 Developing the Knowledge Base

The rules and membership functions together are known as the **knowledge base**. This section explains how to organize the rules and membership functions used in the C500-FZ001.

Determining Objectives

The first step in designing a fuzzy control system is determining the parameters that will be controlled.

1, 2, 3...

1. Decide what part of the controlled system will be controlled by the Unit.
2. One of the strengths of fuzzy logic is its capacity to operate with more than one objective. Decide what the objectives are for the fuzzy control system, the order of priority for those objectives, and the goal for each objective. As an example, the following table lists three objectives for a manufacturing process in order of importance: increasing manufacturing precision, speed, and energy efficiency.

Item	Objective	Goal
Manufacturing precision	Maximize precision	x_1 mm
Manufacturing speed	Maximize speed	x_2 m/s
Energy efficiency	Minimize power consumption	x_3 kilowatts

Determining I/O

1, 2, 3...

1. Organize the information about the controlled system and decide what information is necessary to achieve the objectives.

2. Divide the information into inputs and outputs.
3. Organize the information into groups by criteria such as those below:
   - Grouped according to objective.
   - Grouped according to output.
   - Grouped according to input.
   - Grouped according to situation.
   - Grouped according to steps (e.g., condition A goal A output value).

   These groups can be used as they are when creating rules.
4. Review the groups to verify that they satisfy the objectives.

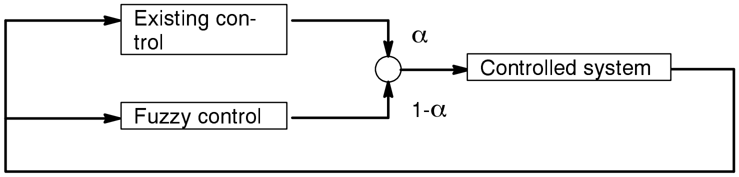
System Configuration

The Fuzzy Logic Unit can be used alone or in a conventional-fuzzy hybrid control system. The best control system will depend on the characteristics of the controlled system.

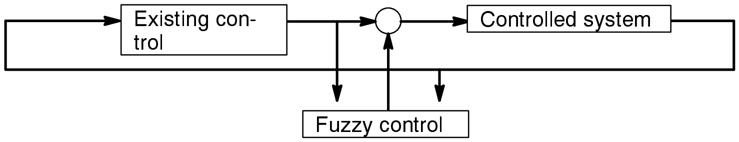
1, 2, 3...

1. The following diagrams show two versions of conventional-fuzzy hybrid control systems. The first uses fuzzy logic in parallel with the existing control system and the second uses fuzzy logic to refine the output of the existing control system.

Parallel control:

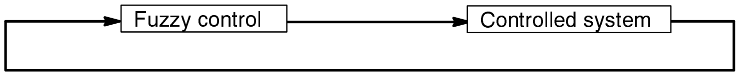


Fuzzy logic-refined control:



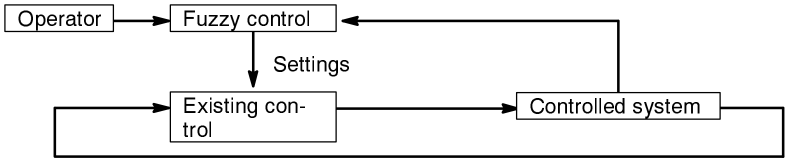
2. Fuzzy logic alone can be used to control the system, as shown below.

Fuzzy logic control:



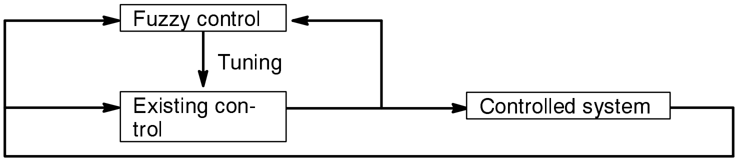
3. A Fuzzy Logic Unit can be installed to act as an interface between the human operator and existing control system in systems that require an operator to adjust settings.

Fuzzy logic interface:



4. A Fuzzy Logic Unit can be used to tune the output of the existing control system. This configuration is similar to the fuzzy logic-refined control system in configuration 1.

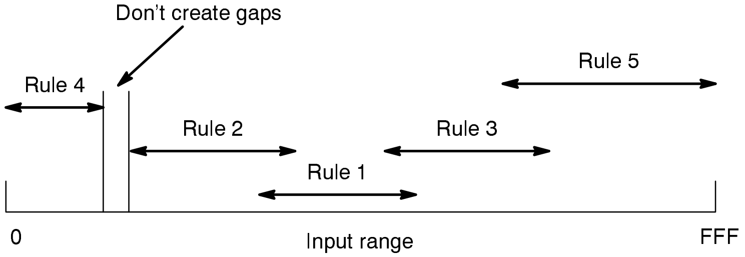
Fuzzy logic-tuned control:



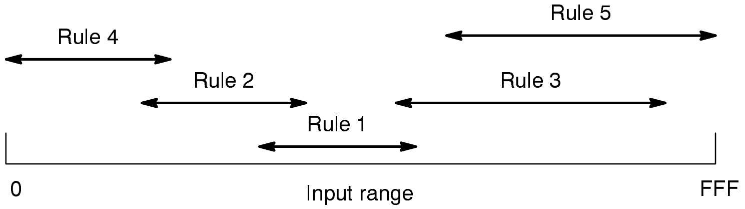
Making the Rules

Make enough rules so that there are outputs from at least two rules in input ranges where precision and smooth operation are required. However, having 3 or more rules with a grade of more than 50% at the same time is redundant.

There must be at least one output from a rule at all times. An error will occur if there isn't an output, so be sure not to allow gaps in input coverage such as the one shown below.



In cases where precise response is not required, make rules that cover a broad section of the input range, as shown below.

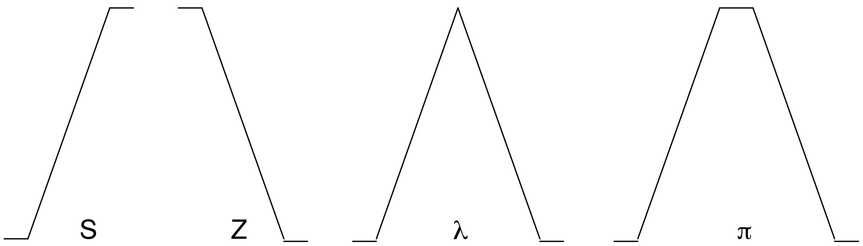


The time required to write and execute the program can be reduced by making a single rule with two conclusion parts instead of two separate rules with just one conclusion part where possible. If the response time is too slow, try grouping rules according to situation, reducing the number of rules as much as possible, and broadening the input range that each rule covers.

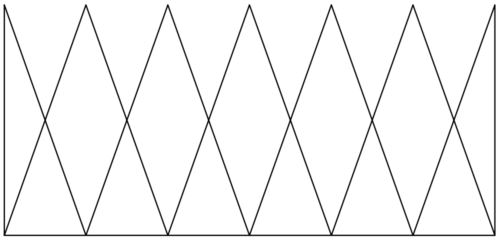
We recommend saving to the floppy disk frequently to avoid losing data, and keeping a journal of changes and the reasons for making the changes as an educational resource for the programmer himself and others.

Condition Functions

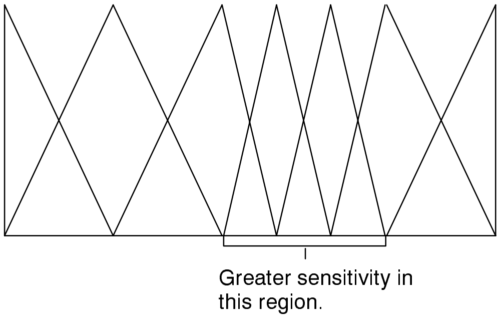
The four standard shapes for condition membership functions are shown below.

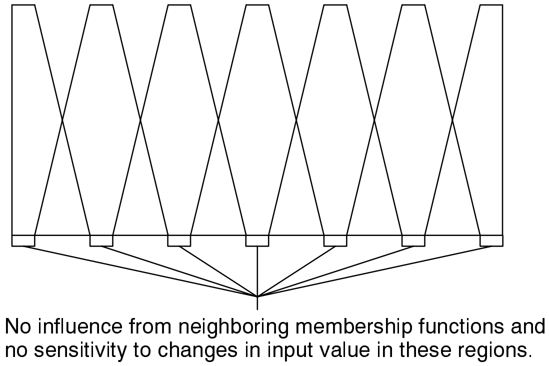


In a closed-loop system, we recommend starting with triangular (λ) membership functions that are spaced one-half their width, as shown below. The input range has a large effect on control performance, so proper I/O signal conversion (gain adjustment) is very important.



The following diagrams show two effects of modified membership functions. In the diagram on the left, the rules in one section of the input range cover a narrower range for greater sensitivity. In the diagram on the right, the central section of each membership function has no sensitivity to changes in the input value.

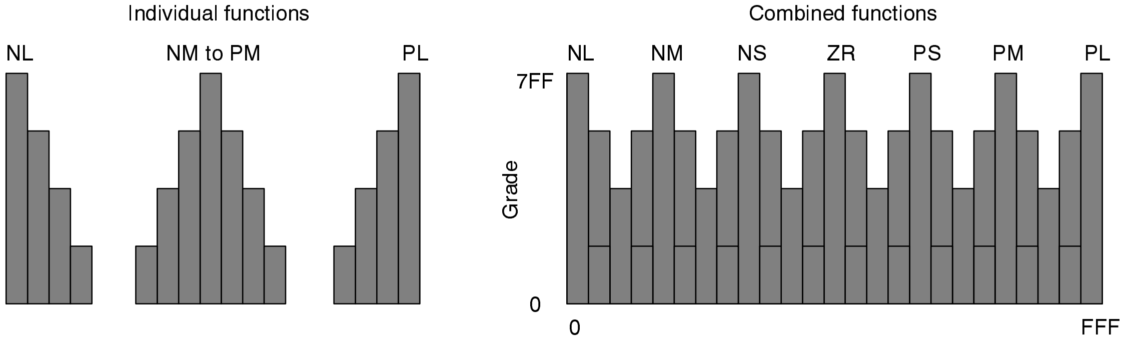



No influence from neighboring membership functions and
no sensitivity to changes in input value in these regions.

The construction of the membership functions will depend on the characteristics of the controlled system. The arrangement of membership functions can become very asymmetrical and complicated, particularly with open-reel control systems and recognition/identification systems.

Conclusion Function

The shape and location of the conclusion membership functions is fixed. The functions are roughly triangular, made up of 4 columns for NL and PL and 7 columns for NM through PM. Three columns of the membership functions overlap the adjacent function(s), producing a total of 25 columns in the output range.



Integrated Development of the Rules and Membership Functions

Follow the procedure below when modifying the knowledge base for improved performance.

1, 2, 3...

1. The first step in designing the knowledge base is to list the rules. Use a standard shape for the membership functions.

   After listing the rules, we'll modify the knowledge base based on the results of simulations.
2. Modify the rules, add new rules when needed.

3. Before modifying the membership functions, adjust the I/O gain.
4. Modify the membership functions. Begin by adjusting the width and position of the condition parts. For the conclusion parts, position is the only setting that can be adjusted.

After completing this procedure, test for unusual or unexpected outputs using a variety of input combinations.

1-3-4 Downloading the Knowledge Base

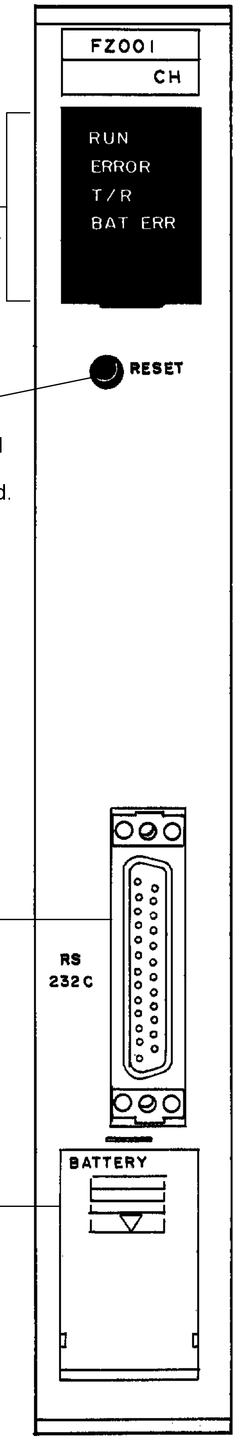
After the knowledge base has been written in the computer, it must be transferred (downloaded) to the Fuzzy Logic Unit. Refer to *4-2 Making and Downloading the Knowledge Base* for details.

SECTION 2
Components and Switch Settings

This section describes the components of the C500-FZ001 and their functions.

2-1 Components

Front



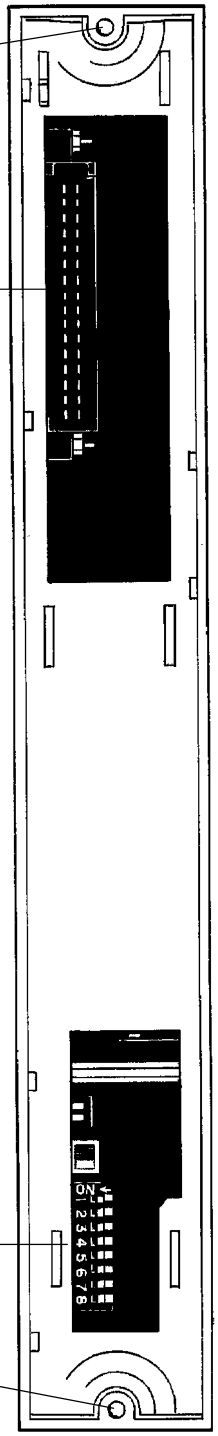
LED indicators
Indicate operational status. (The front DIP switch is under the LED indicator cover. Refer to *3-1 Switch Settings* for details.)

Reset switch
Used to restart the Unit. All I/O status is reset to zero when this switch is pressed.

RS-232C port
Used to connect to the computer.

Battery compartment

Back



Mounting screw
Attaches the Unit to the PC's Backplane.

Connector
Connects the Unit to the PC's CPU and power supply.

Back DIP switch
Refer to *3-1 Switch Seti*
for details.

Mounting screw
Attaches the Unit to the PC's Backplane.

2-2 Indicators

The LED indicators on the front of the Fuzzy Logic Unit provide visual information on the general operation of the Unit. The function of the indicators is described in the following table.

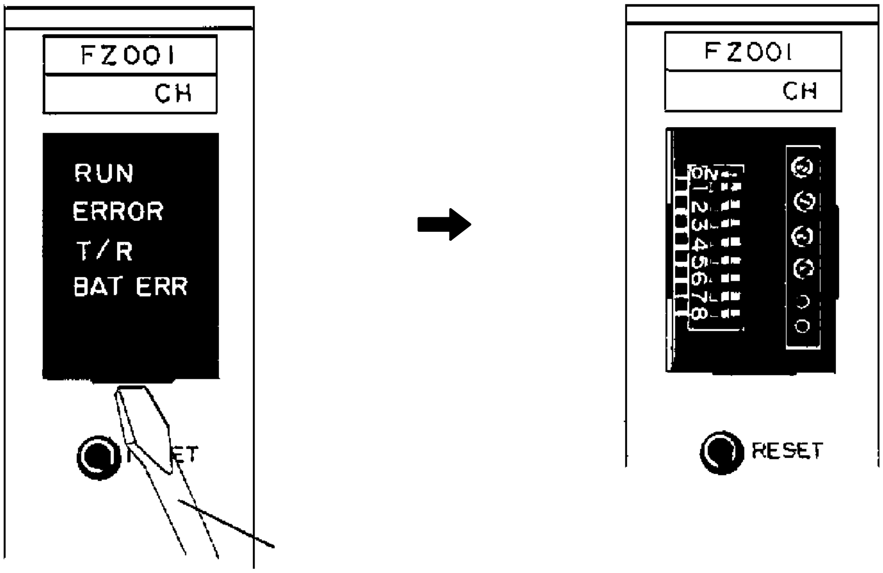
Indicator	Function
RUN	Lights when the Unit is operating normally.
ERROR	Lights when an error is discovered in self-diagnosis operations. Flashes when the knowledge base hasn't been entered or has been damaged.
T/R	Lights when data is being transmitted to or from the Unit via the RS-232C port.
BAT ERR	Lights when the battery voltage is below the minimum.

2-3 Switch Settings

The settings of the DIP switches on the front and back of the Unit determine how the Unit will communicate with a programming device through the RS-232C port. Always turn the PC power off before changing the DIP switch settings.

Front DIP Switch

The front DIP switch is located under the LED indicator cover and can be accessed by lifting off the cover with a standard screwdriver, as shown in the illustration below.



Pins 1 to 3 set the baud rate for communications through the RS-232C port, as shown in the table below. (The baud rate is set to 9,600 at the factory.)

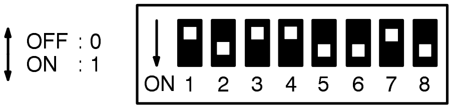
Pin settings			Baud rate
1	2	3	
OFF	OFF	OFF	Not used.
OFF	OFF	ON	19,200
OFF	ON	OFF	9,600
OFF	ON	ON	4,800
ON	ON	OFF	2,400
ON	OFF	ON	1,200
ON	ON	OFF	600
ON	ON	ON	300

Pins 4 to 8 of the front DIP switch are used to set the unit number of the C500-FZ001 as seen from the computer. The unit number can be 0 to 31, and is set as a binary number. Pin values (when ON) are shown in the table below. (The unit number is set to 0 at the factory.)

Pin value				
Pin 4	**Pin 5**	**Pin 6**	**Pin 7**	**Pin 8**
2^4 (16)	2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)

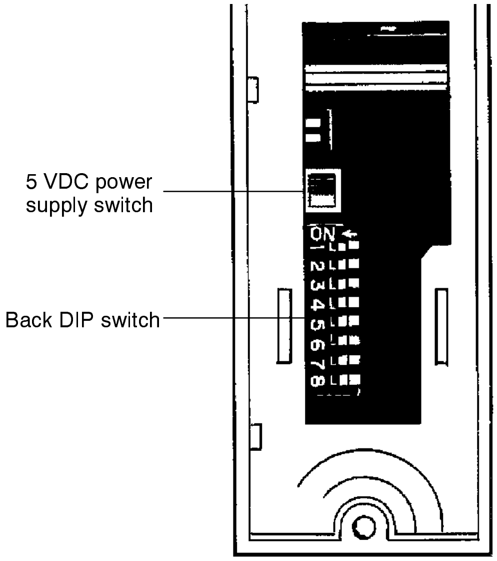
Example

The following diagram shows the front DIP switch set for 9,600 baud and unit number 13 (0 + 8 + 4 + 0 + 1 = 13).



Back DIP Switch

The back DIP switch is located on the back of the Unit near the bottom and is used to set the communications protocol for the RS-232C port.



Pins 1 to 3 are used to set the character format and pin 7 is used to set the CTS signal, as shown in the following tables. Pins 4 to 6 and 8 are not used. (These pins are all set to 0 at the factory.)

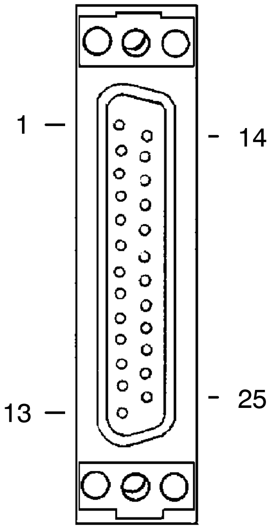
Pin settings			Character format
1	2	3	
OFF	OFF	OFF	7 data bits, even parity, 2 stop bits
ON	OFF	OFF	7 data bits, odd parity, 2 stop bits
OFF	ON	OFF	7 data bits, even parity, 1 stop bit
ON	ON	OFF	7 data bits, odd parity, 1 stop bit
OFF	OFF	ON	8 data bits, no parity, 2 stop bits
ON	OFF	ON	8 data bits, no parity, 1 stop bit
OFF	ON	ON	8 data bits, even parity, 1 stop bit
ON	ON	ON	8 data bits, odd parity, 1 stop bit

Pin 7 setting	CTS signal
OFF	The CTS signal is received from the external source.
ON	The CTS signal is ignored (connected to 0 V).

2-4 RS-232C Interface

The C500-FZ001 communicates with programming devices through the RS-232C port on the front of the Unit. The function of the pins in the RS-232C port are shown in the following table.

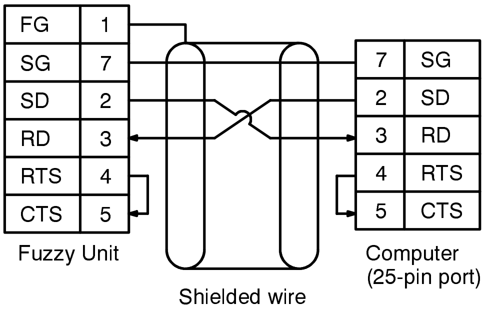
Connector Pin Numbers

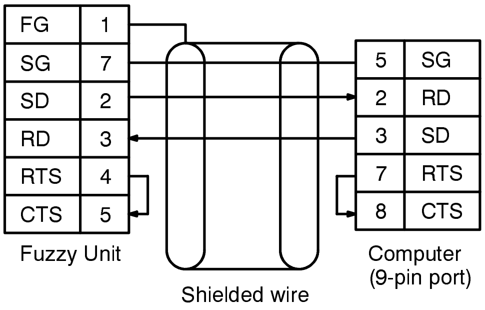


Pin No.	Abbr.	Name	I/O
1	FG	Frame ground	-
2	SD	Send data	Out
3	RD	Receive data	In
4	RTS	Request to send	Out
5	CTS	Clear to send	In
6	-	Not used	
7	SG	Signal ground	-
8 to 13	-	Not used	-
14	5 V	Optical interface power supply	-
15 to 19	-	Not used	-
20	ER	Enable receive	Out
21 to 25	-	Not used	-

C500-FZ001/Computer Connection

Connect the cable to the Unit only after turning off the computer.



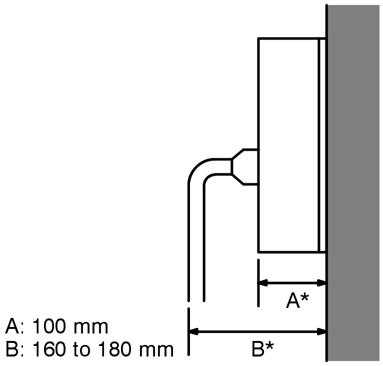


Connecting Cable

The maximum cable length between the C500-FZ001 and a computer is 15 m. The connecting cable must be purchased separately.

2-5 Mounting Depth

The depth of the Fuzzy Logic Unit is only 100 mm, including the Backplane. Allow an additional 60 to 80 mm for cable clearance, as shown below.



SECTION 3
Memory Allocation

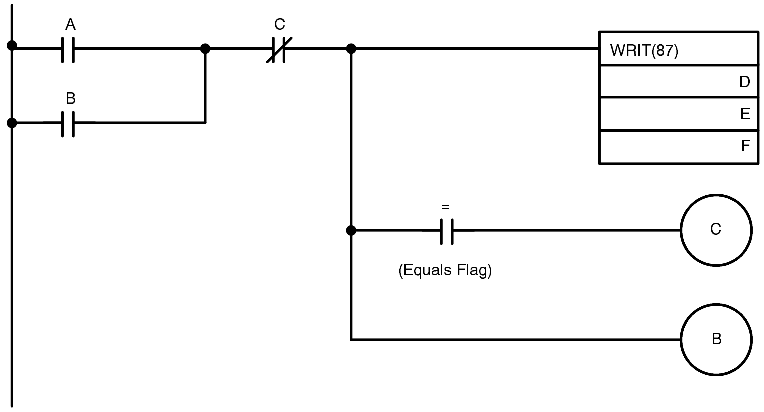
This section explains how words in the PC are allocated to the Fuzzy Logic Unit, and the function of those words as flags, control bits, and I/O words.

3-1 Memory Allocation in the PC

The Fuzzy Logic Unit is allocated 2 words in the IR Area of a C500, C1000H, or C2000H, or in the I/O Area of a CV500, CV1000, CV2000, or CVM1 when the I/O table is registered. These two words are used with the I/O READ and I/O WRITE instructions to transfer data between the Unit and the PC. The first word is used to transfer input data from the PC to the Unit and the second word is used to transfer output data from the Unit to the PC.

3-1-1 Input Data Transfer

Input data is transferred to the Fuzzy Logic Unit from the PC with the WRIT(87) instruction (WRIT(191) in CV-series PCs) via the first word allocated to the Unit in the I/O Table Registration operation, as shown below. Up to 8 words can be transferred.



A: Start condition
B: Self-holding bit
C: Stops processing when fuzzy logic processing is completed
D: Number of words (1 to 8, BCD)
E: Address of first source word in PC
F: First word allocated to the Unit

Data Format

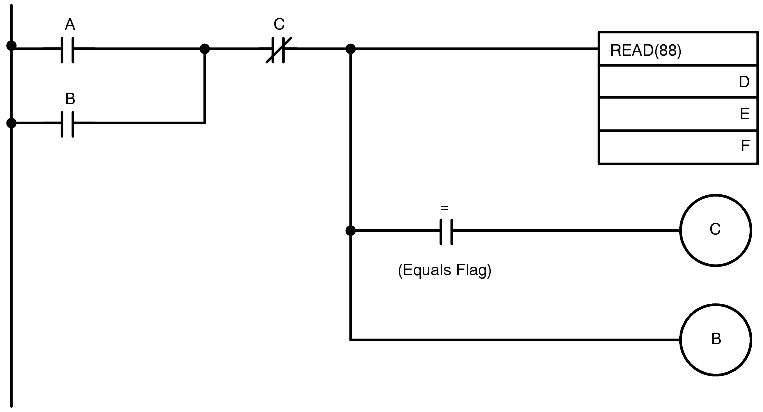
Each input is allocated one word, and the input data is contained in bits 0 to 12 of that word in 3-digit hexadecimal. The input range is thus 000 to FFF (0 to 4095 decimal). Bits 12 to 15 are not used.

The data structure for inputs is shown in the following table beginning from the first input word, I. In this case there are 8 inputs, but anywhere from 1 to 8 inputs can be used. We recommend using the DM Area to store the source data.

Word	Bits 12 to 15	Bits 0 to 11
I	Not used. Set to zero.	Input 0 data
I+1	Not used. Set to zero.	Input 1 data
⋮	⋮	⋮
I+7	Not used. Set to zero.	Input 7 data

3-1-2 Output Data Transfer

Output data is transferred from the Fuzzy Logic Unit to the PC with the READ(88) instruction (READ(190) in CV-series PCs) via the second word allocated to the Unit in the I/O Table Registration operation, as shown below. Four words are transferred.



A: Start condition
B: Self-holding bit
C: Stops loading of output data when completed
D: Number of outputs (4, BCD)
E: Address of first destination word in PC
F: Second word allocated to the Unit

Data Format

Each output is allocated one word, and the output data is contained in bits 0 to 12 of that word in 3-digit hexadecimal. The output range is thus 000 to FFF (0 to 4095 decimal). Bits 12 to 15 are not used.

The output data structure is shown in the following table beginning from the first output word, O. Two words (O and O+1) are allocated to output data regardless of the number of outputs specified. We recommend using the DM Area to store output data.

Word	Bits 12 to 15	Bits 0 to 11
O	Not used. Set to zero.	Output 0 data
O+1	Not used. Set to zero.	Output 1 data

Error codes for the outputs are contained in words O+2 and O+3, as shown in the following table.

Word	Bit 15	Bits 5 to 14	Bits 0 to 4
O+2	ER	Not used. Set to zero.	Output 0 error code
O+3	ER	Not used. Set to zero.	Output 1 error code

The Error Flag (ER) will be turned ON when any of bits 0 to 4 for that output are turned ON. The following table shows the function of the error code bits.

Bit	Flag name	Function
00	Input Range Exceeded Flag	Turned ON to indicate that the input range (000 to FFF) has been exceeded.
01	Too Many Input Words Flag	Turned ON to indicate that too many inputs have been transferred to the Unit.
02	Too Few Input Words Flag	Turned ON to indicate that not enough inputs have been transferred to the Unit.
03	Undefined Output Flag	Turned ON to indicate that the output was not defined in the Fuzzy Logic Settings. The final result will be set to EFFF.
04	No Corresponding Output Flag	Turned ON to indicate that no grade has been output for that output. The final result will be set to FFFF.

It isn't absolutely necessary to read the error codes, but we recommend loading them to find any errors in fuzzy logic processing as soon as possible. If you determine that it isn't necessary to load the error codes, set the Number of Outputs setting in the Fuzzy Logic Settings to the required number of outputs.

3-1-3 Status Flags

The two words allocated to the Fuzzy Logic Unit are also used to transfer data on PC and Unit status, as well as input and output data. The first word is used to transfer data on PC status to the Unit and the second word is used to transfer data on Unit status to the PC. These flags are refreshed during I/O refreshing at the end of each scan/cycle after the I/O READ or I/O WRITE instruction has been executed. The table below shows the status flags contained in the first word allocated to the Unit.

Bit(s)	Flag name	Function
00	PC Busy Flag	These flags are set or reset automatically when WRIT(87) or READ(88) is executed. (WRIT(191) and READ(190) in CV-series PCs)
01	PC Read Complete Flag	
02	PC Write Complete Flag	
03 to 15	---	Not used.

The table below shows the status flags contained in the second word allocated to the Unit.

Bit(s)	Flag name	Function
00	Unit Busy Flag	These flags are set or reset automatically when WRIT(87) or READ(88) is executed. (WRIT(191) and READ(190) in CV-series PCs)
01	Unit Read Complete Flag	
02	Unit Write Complete Flag	
03	Fuzzy Error Flag	Turned ON to indicate that an error has occurred during the self-diagnostic check.
04	Memory Error Flag	Turned ON to indicate that an error has been found in the rules or membership functions.
05	Battery Error Flag	Turned ON to indicate that the battery voltage has fallen below the minimum, or that the battery isn't installed or connected properly.
06	---	Not used.
07	Processing Enabled Flag	Turned ON to indicate that the Unit is ready to begin fuzzy logic processing. Turned OFF when rules and membership functions are being transferred or on command from the computer.
08 to 15	---	Not used.

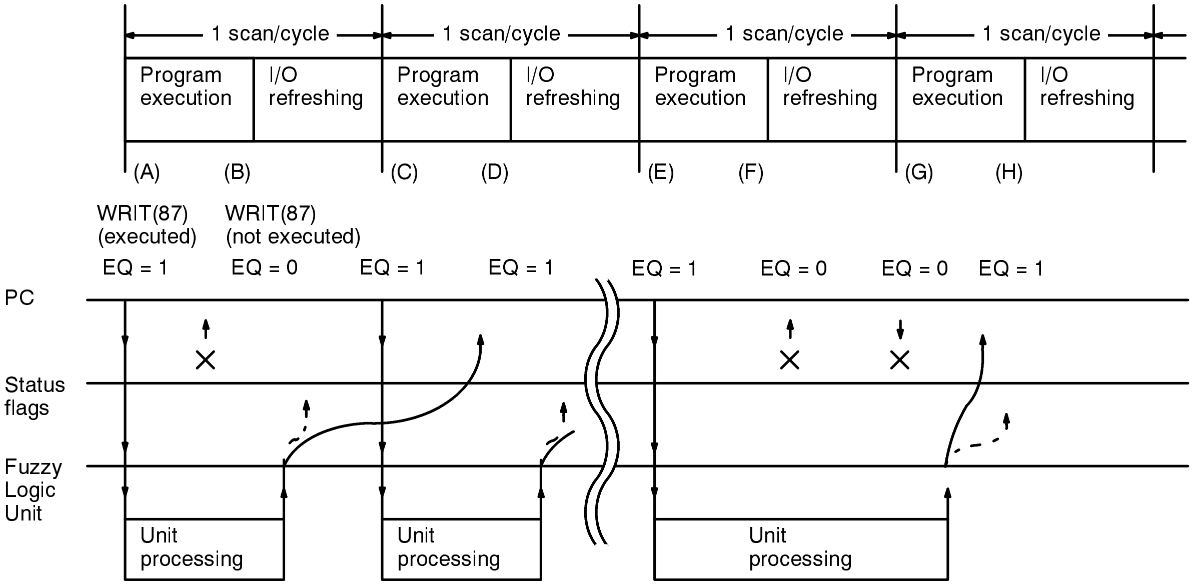
3-2 Data Transfer and Timing

The WRIT(87) instruction (WRIT(191) in CV-series PCs), which transfers the input data from the PC to the Unit, will be executed when its execution condition is ON and the Unit Busy and Unit Read Complete Flags are OFF. When WRIT(87) has been executed, the PC Write Complete Flag will be turned ON for one scan/cycle, and the Equals Flag will be turned ON. The status of the Equals Flag can be used to determine whether or not WRIT(87) has been executed.

Likewise, the READ(88) instruction (READ(190) in CV-series PCs), which transfers the output data from the Unit to the PC, will be executed when its start condition is ON and the Unit Busy and Unit Read Complete Flags are OFF. When READ(88) has been executed, the PC Write Complete Flag will be turned ON for one scan/cycle, and the Equals Flag will be turned ON. The status of the Equals Flag can be used to determine whether or not READ(88) has been executed.

While the input and output data is transferred when WRIT(87) and READ(88) are executed, the flags indicating PC and Unit status (described in *3-1-3 Status Flags*) are refreshed during I/O refreshing at the end of the scan/cycle, so these flags cannot be read until after the next I/O refresh.

The diagram below shows the timing of data transfer between the PC and the Unit.



After a WRIT(87) instruction has been executed, the next WRIT(87) instruction won't be executed until processing has been completed in the Fuzzy Logic Unit. READ(88) will be executed, transferring the results of the previous processing, even if fuzzy logic processing is in progress using new data transferred from another WRIT(87) instruction. However, if fuzzy logic processing using the new data is completed before READ(88) is executed, the new results will be transferred and the results of the previous processing lost.

SECTION 4
Program and Knowledge Base

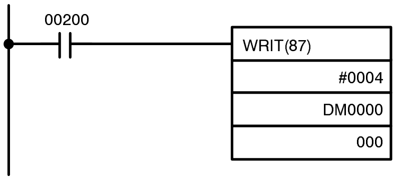
This section explains how to prepare the two basic software requirements for Fuzzy Logic Unit operation: the sub-program in the PC and the knowledge base in the Fuzzy Logic Unit itself.

4-1 PC Sub-program

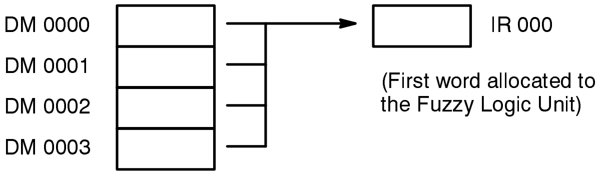
A sub-program must be prepared within the PC program to control the transfer of data between the PC and the Unit. This program can be written with standard instructions such as MOV(21), WRIT(87), and READ(88). In this section we will briefly review the use of WRIT(87) and READ(88) (WRIT(191) and READ(190) in the CV-series), because these instructions are essential to operation of the Fuzzy Logic Unit, and then provide examples of sub-programs transferring data to and from Analog Input and Output Units, which are often used with the Fuzzy Logic Unit.

4-1-1 I/O WRITE

The I/O WRITE instruction (WRIT(87) or WRIT(191)) is used to transfer input data from the PC to the Unit. In the example below, which is written for a C1000H PC, the four input words contained in DM 0000 to DM 0003 are transferred to the Unit through IR 000, the first word allocated to the Unit when the I/O table is registered.



Address	Instruction	Operands
00200	LD	00200
00201	WRIT(87)	
		# 0004
		DM 0000
		000



The I/O WRITE instruction will be executed when the Fuzzy Logic Unit is capable of receiving the data (i.e., when the Unit Busy and Unit Read Complete Flags are OFF) and the execution condition is ON. Check the status of the Equals and Error Flags to verify that the instruction has been executed. The meaning of these flags' status is described in the following table.

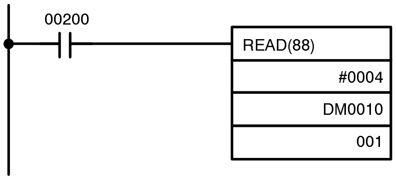
Flag status		Meaning
Equals (=)	Error (ER)	
ON	OFF	WRIT(87) executed without errors.
OFF	OFF	WRIT(87) wasn't executed because the Unit is busy or hasn't yet load the data from the previous execution of WRIT(87).
Unchanged	ON	An error has occurred during execution of WRIT(87). Refer to your PC's *Operation Manual* for possible causes of errors.

Note The I/O WRITE instruction will be executed when the Fuzzy Logic Unit is capable of receiving the data and the execution condition is ON, even if the Output OFF Bit is ON.

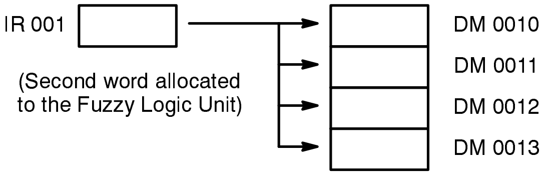
4-1-2 I/O READ

The I/O READ instruction (READ(88) or READ(190)) is used to transfer output data from the Unit to the PC's CPU. In the example below, which is written for a

C1000H PC, the four output words (two containing processing results and two containing error codes) are transferred to DM 0010 to DM 0013 through IR 001, the second word allocated to the Unit when the I/O table is registered.



Address	Instruction	Operands	
00200	LD		00200
00201	READ(88)		
		#	0004
		DM	0010
			001



The I/O READ instruction will be executed when the Fuzzy Logic Unit is sending the data and the execution condition is ON. Check the status of the Equals and Error Flags to verify that the instruction has been executed. The meaning of these flags' status is described in the following table.

Flag status		Meaning
Equals (=)	**Error (ER)**	
ON	OFF	READ(88) executed without errors.
OFF	OFF	READ(88) wasn't executed because the Unit is busy or isn't sending the data.
Unchanged	ON	An error has occurred during execution of READ(88). Refer to your PC's *Operation Manual* for possible causes of errors.

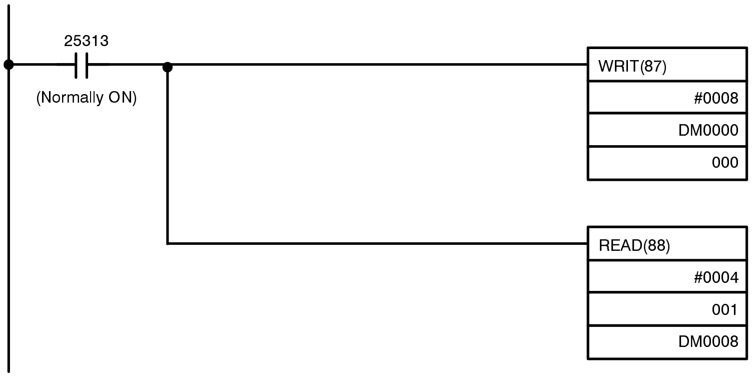
Note The I/O READ instruction will be executed when the Fuzzy Logic Unit is sending the data and the execution condition is ON, even if the Output OFF Bit is ON.

4-1-3 Basic Sub-programs

The following examples show sub-programs for controlling data transfer with various Analog Input and Analog Output Units. In all cases, the Fuzzy Logic Unit is allocated words IR 000 to IR 001.

1: Basic I/O Data Transfer

This example shows the ladder diagram and mnemonics that will transfer I/O data between the Fuzzy Logic Unit and the PC when the timing isn't important and it isn't necessary to transfer the output data once for each time the input data is transferred. The 8 input data words are contained in words DM 0000 to DM 0007, and the 4 output data words are contained in words DM 0008 to DM 0011.



Address	Instruction	Operands
00000	LD	25313
00001	READ(87)	
		# 0008
		DM 0000
		000

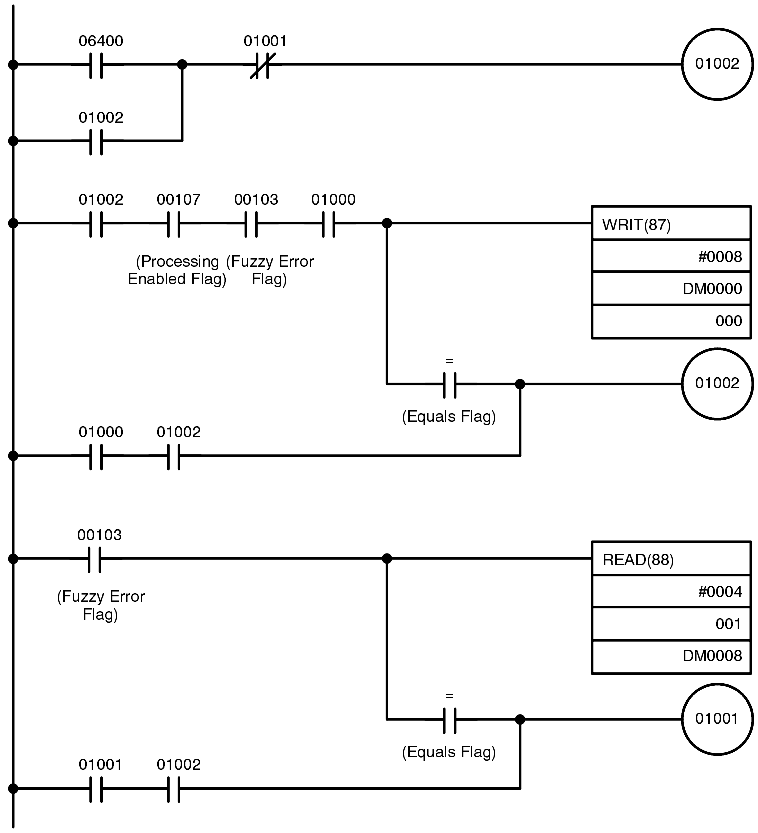
Address	Instruction	Operands
00002	WRIT(88)	
		# 0004
		001
		DM 0008

Note 1. The function codes for the I/O READ and I/O WRITE instructions are different in the CV-series PCs: READ(191) and WRIT(190).

2. The output data will not be output for several PC scans (or cycles). The delay will be twice the CPU scan time or fuzzy logic processing time, whichever is longer.

2: I/O Data Transfer

This example shows the ladder diagram that will transfer I/O data between the Fuzzy Logic Unit and the PC when the timing must be controlled and it is necessary to transfer the output data once for each time the input data is transferred. The 8 input data words are contained in words DM 0000 to DM 0007, and the 4 output data words are contained in words DM 0008 to DM 0011.

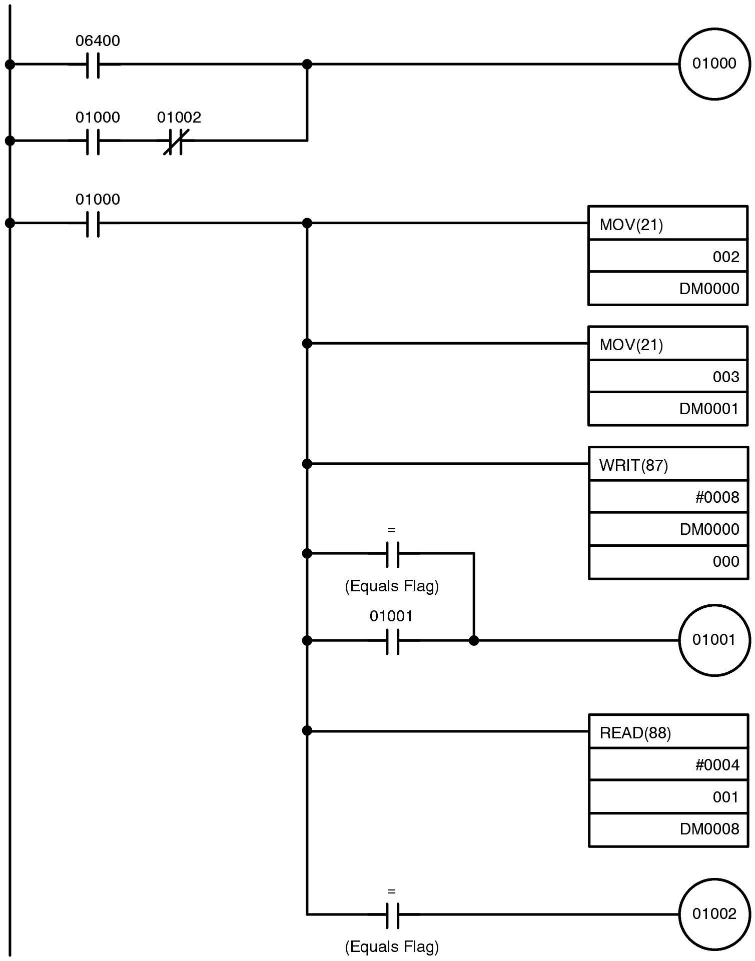


After the I/O WRITE instruction has been executed, the next I/O WRITE instruction won't be executed until processing has been completed in the Fuzzy Logic Unit. The completion of processing in the Fuzzy Logic Unit also enables execution of the I/O READ instruction. If the next I/O WRITE instruction is executed before the I/O READ instruction, the I/O READ instruction can be executed during processing, transferring the results of the previous processing. However, if the I/O READ instruction is executed after fuzzy logic processing is completed using the new data, the new results will be transferred and the results of the previous processing lost.

Check the Status Flags (See *3-1-3 Status Flags*) to see whether the I/O WRITE and I/O READ instructions have been executed.

3: Data Transfer from Unit

This example shows a ladder diagram that will transfer input data to the PC from a C500-AD001 Analog Input Unit that is allocated words IR 002 and IR 003, transfer that data to the Fuzzy Logic Unit, and then transfer the results of fuzzy logic processing from the Fuzzy Logic Unit back to the PC. Once again, the 8 input data words are contained in words DM 0000 to DM 0007, and the 4 output data words are contained in words DM 0008 to DM 0011.

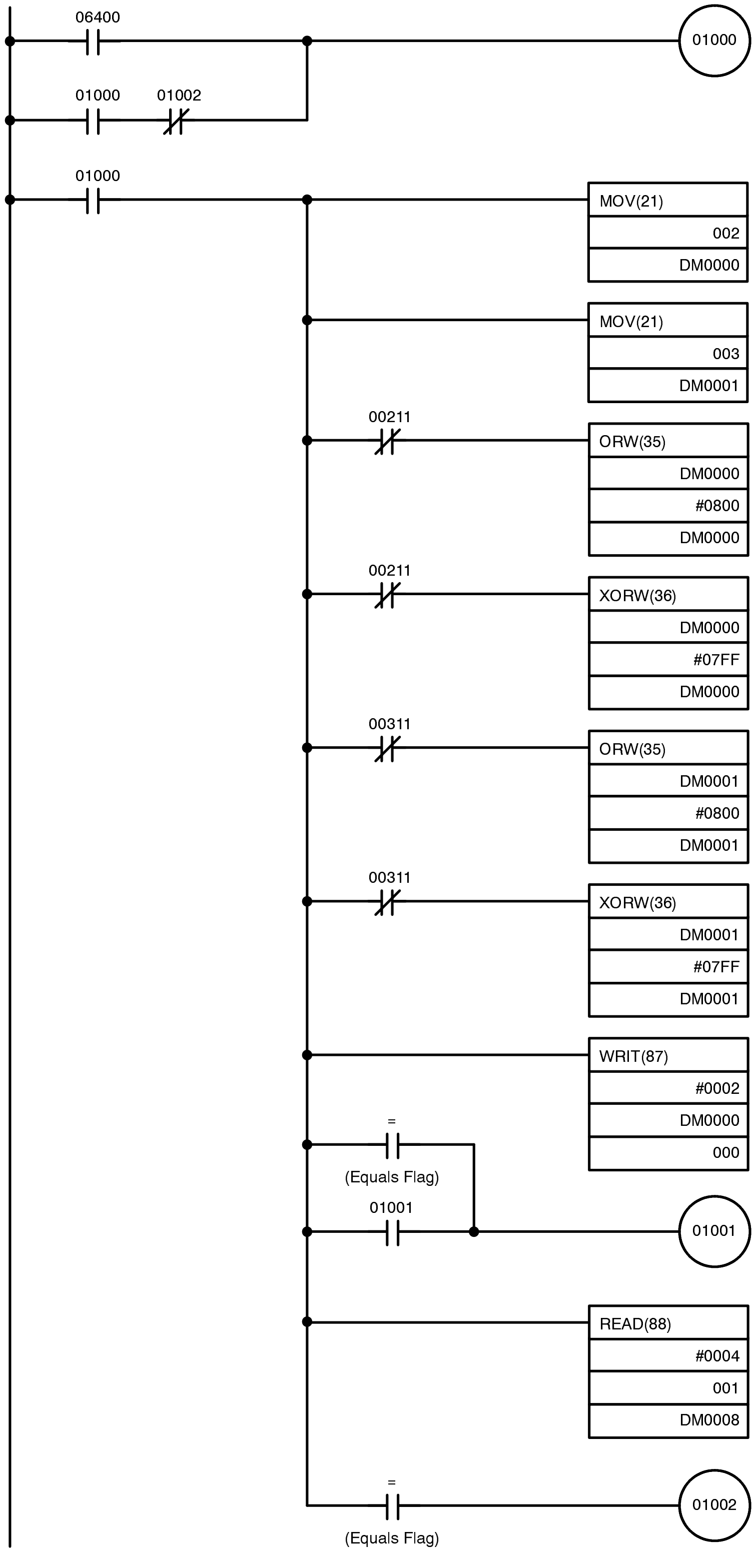


If IR 06400 stays ON for just one scan, the input data will be loaded from the Analog Input Unit, the Fuzzy Logic Unit will process that data just once and the results will be transferred to DM 0008 to DM 0011.

If IR 06400 stays ON continuously, the Fuzzy Logic Unit will continue processing data from the Analog Input Unit, and the results will change as the input data changes.

4: Data Transfer from Unit

This example shows a ladder diagram that will transfer input data to the PC from a C500-AD005 Analog Input Unit that is allocated words IR 002 and IR 003, convert the data and transfer it to the Fuzzy Logic Unit, and then transfer the results of fuzzy logic processing from the Fuzzy Logic Unit back to the PC. Once again, the 8 input data words are contained in words DM 0000 to DM 0007, and the 4 output data words are contained in words DM 0008 to DM 0011.

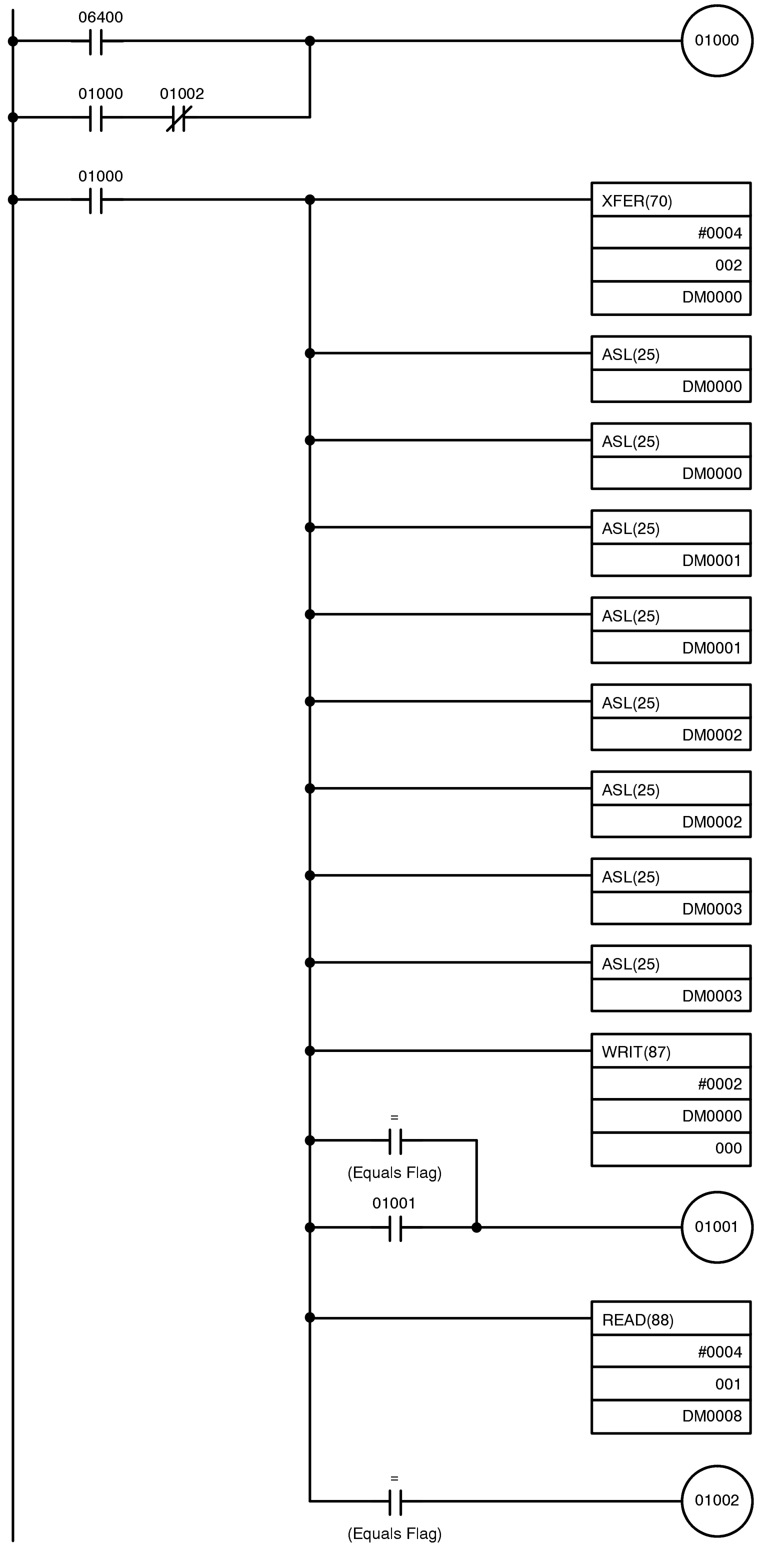


If IR 06400 stays ON for just one scan, the input data will be loaded from the Analog Input Unit and converted to an acceptable range. The converted data is then processed just once by the Fuzzy Logic Unit and the results are transferred to DM 0008 to DM 0011.

If IR 06400 stays ON continuously, the Fuzzy Logic Unit will continue processing data from the Analog Input Unit, and the results will change as the input data changes.

5: Data Transfer from Unit

This example shows a ladder diagram that will transfer input data to the PC from a C500-AD006 Analog Input Unit that is allocated words IR 002 to IR 005, convert the data and transfer it to the Fuzzy Logic Unit, and then transfer the results of fuzzy logic processing from the Fuzzy Logic Unit back to the PC. Once again, the 8 input data words are contained in words DM 0000 to DM 0007, and the 4 output data words are contained in words DM 0008 to DM 0011.

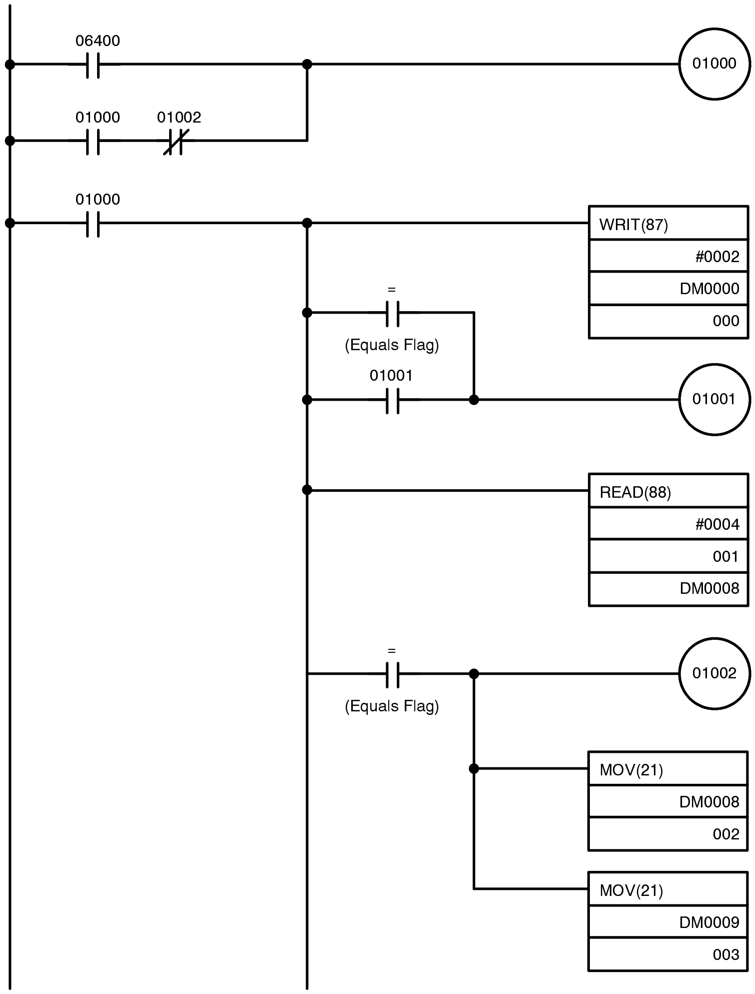


If IR 06400 stays ON for just one scan, the input data will be loaded from the Analog Input Unit and converted to an acceptable range. The converted data is then processed just once by the Fuzzy Logic Unit and the results are transferred to DM 0008 to DM 0011.

If IR 06400 stays ON continuously, the Fuzzy Logic Unit will continue processing data from the Analog Input Unit, and the results will change as the input data changes.

6: Data Transfer to Unit

This example shows a ladder diagram that will transfer input data to the Fuzzy Logic Unit, and then transfer the results of fuzzy logic processing from the Fuzzy Logic Unit to the PC and then on to a C500-DA001 Analog Output Unit that is allocated words IR 002 and IR 003. Once again, the 8 input data words are contained in words DM 0000 to DM 0007, and the 4 output data words are contained in words DM 0008 to DM 0011.

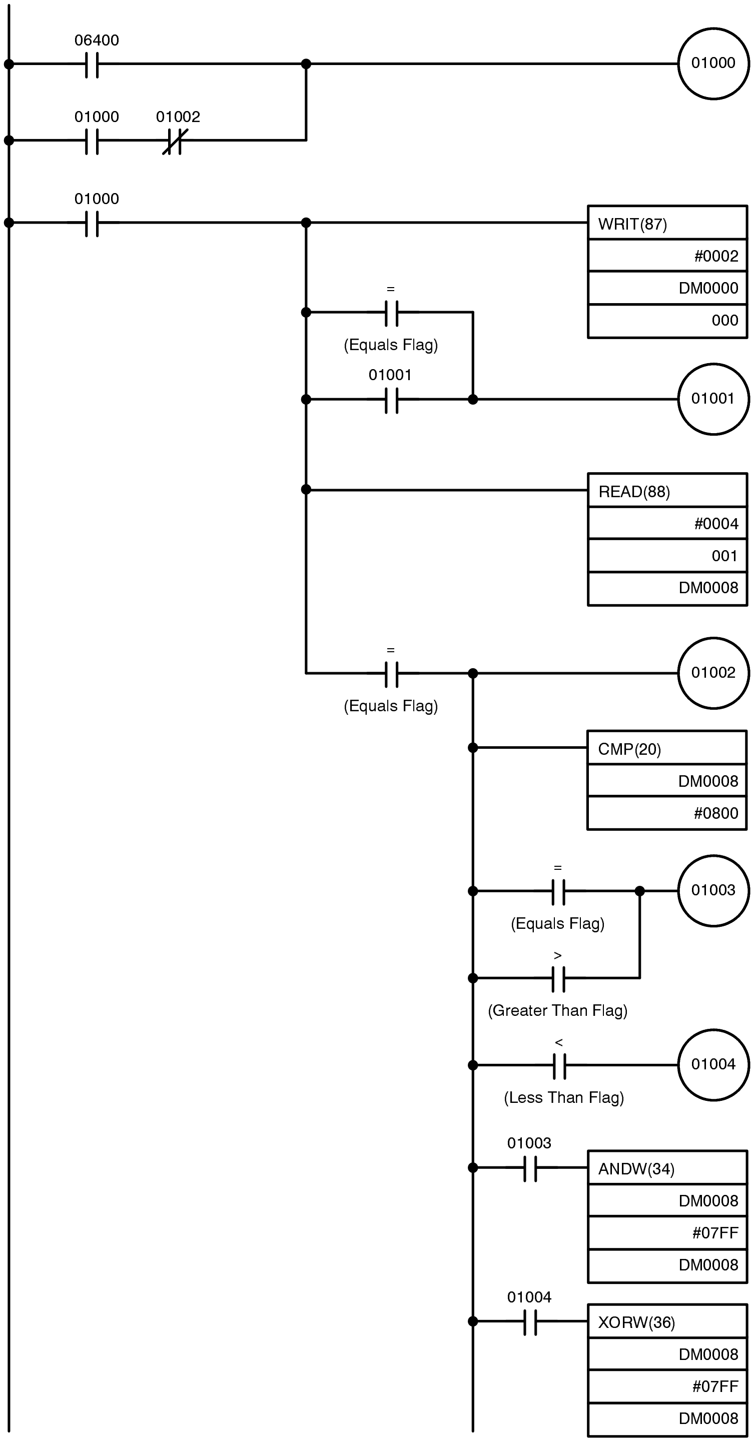


If IR 06400 stays ON for just one scan, the input data is processed just once by the Fuzzy Logic Unit and the results are transferred to the Analog Output Unit.

If IR 06400 stays ON continuously, the Fuzzy Logic Unit will continue processing the input data, and the results will change as the input data changes.

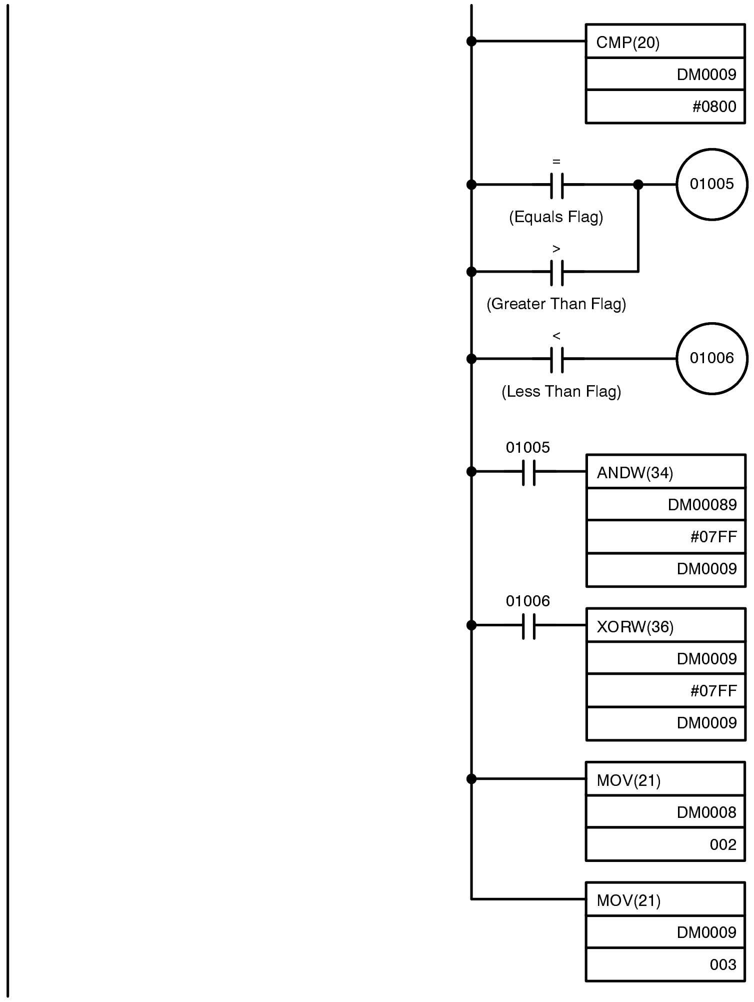
7: Data Transfer to Unit

This example shows a ladder diagram that will transfer input data to the Fuzzy Logic Unit, transfer the results of fuzzy logic processing from the Fuzzy Logic Unit to the PC, convert the data and then transfer it to a C500-DA005 Analog Output Unit that is allocated words IR 002 and IR 003. Once again, the 8 input data words are contained in words DM 0000 to DM 0007, and the 4 output data words are contained in words DM 0008 to DM 0011.



(Continued on the next page.)

(Continued from the previous page.)



If IR 06400 stays ON for just one scan, the input data is processed just once by the Fuzzy Logic Unit and the results are transferred to the Analog Output Unit.

If IR 06400 stays ON continuously, the Fuzzy Logic Unit will continue processing the input data, and the results will change as the input data changes.

4-2 Making and Downloading the Knowledge Base

The knowledge base must be created with the FSS in the computer and then downloaded to the Fuzzy Logic Unit. The basic steps involved in making and downloading the knowledge base are listed below.

1, 2, 3...

1. Make the knowledge base using the FSS.
2. Verify that the knowledge base has been written correctly
3. Download the knowledge base to the Unit.

The operations that can be performed with the FSS are described briefly in a table below. Refer to the *Fuzzy Support Software Operation Manual* for details on these FSS operations.

Operation		Function
File management	Making a new knowledge base file	Clears the knowledge base file being edited and starts making a new one.
	Loading a knowledge base file	Loads a knowledge base file from the floppy or hard disk and starts editing.
	Saving a knowledge base file	Saves the knowledge base file being edited to the floppy or hard disk.
	Recovering the Unit's knowledge base file	Recovers the knowledge base file that was set in the Unit for editing.
	Saving the Unit's processing data	Saves the Unit's file data as a file.
	Print	Prints the knowledge base file being edited and the Unit's operating status.
Making the knowledge base	Making membership functions	Edits the condition and conclusion membership functions.
	Making the rules	Edits the rules.
Verifying the knowledge base	General check	Checks whether the knowledge base can be used in fuzzy logic processing.
	Detail check	Checks for redundancies in the Knowledge base.
Online (link) operations	Affirming the Unit's knowledge base	Loads the knowledge base from the Unit and displays information such as the knowledge base name.
	Knowledge base comparison	Compares the knowledge base in the Unit to the one being edited.
	Inference	Starts or stops fuzzy logic processing.
	Knowledge base download	Downloads the knowledge base file being edited to the Unit.
	Processing status monitor	Monitors the processing status of the Unit in real-time.
	Unit initialization	Initializes the Unit.

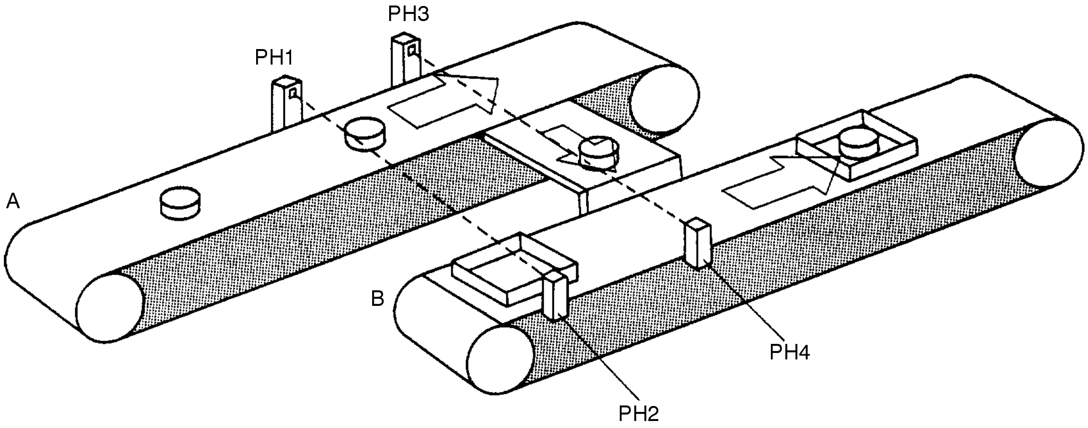
SECTION 5
Example Application

This section describes the development of an actual control system that uses a C500-FZ001 Fuzzy Logic Unit.

5-1 Conveyor Belt Control System Summary

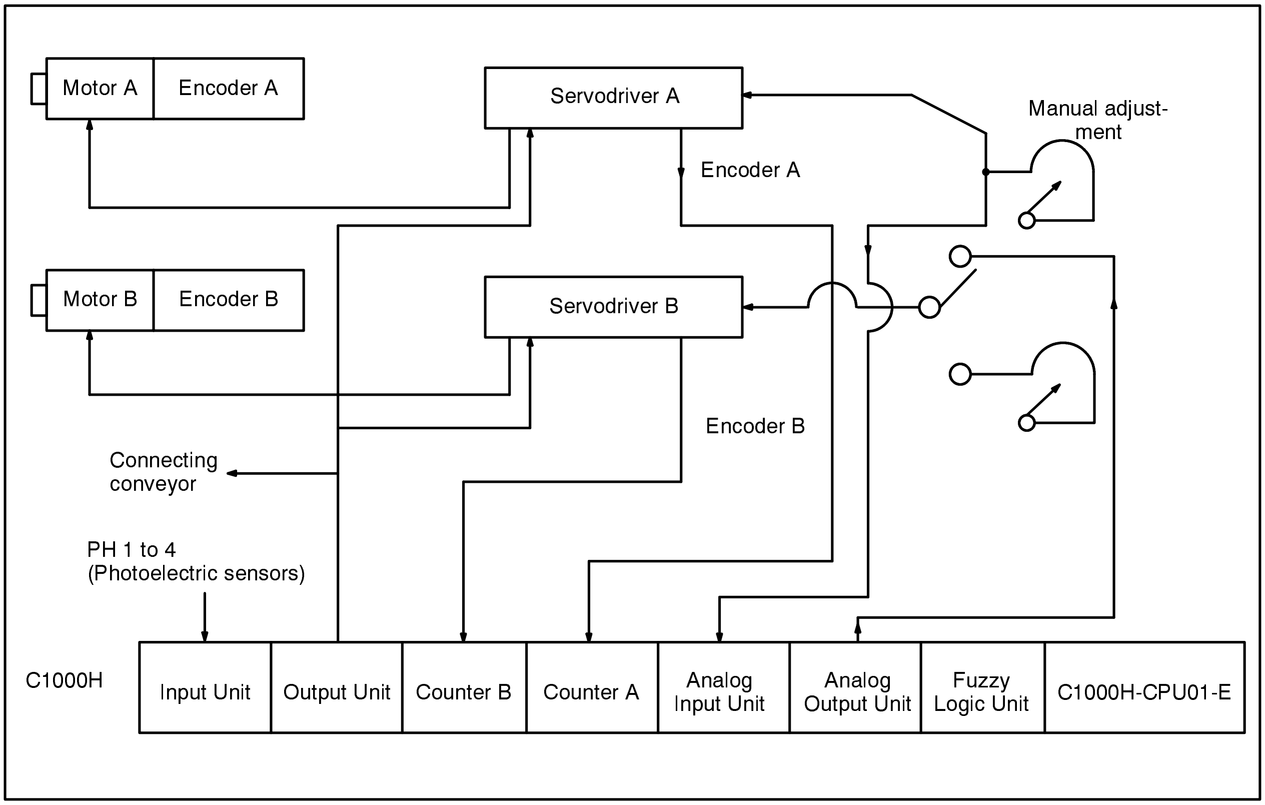
In this example application, a Fuzzy Logic Unit is used to control two conveyor belts for product packaging. The products are carried on conveyor A at random intervals, but at a fixed speed. The boxes are carried at regular intervals on conveyor B, which runs parallel to conveyor A at a speed controlled by the Fuzzy Logic Unit.

The Fuzzy Logic Unit adjusts the speed of conveyor B so that the boxes arrive at the same time as the products. The required information is the offset between the product and the box and the rate that the offset is changing. A conventional control system eliminates the offset between the product and the box, but sometimes no speed adjustment is necessary even though there is a large offset, so the system controlled by the Fuzzy Logic Unit is much more efficient.



Conveyor A:	Carries the products.
Conveyor B:	Carries the boxes.
PH 1 to PH 4:	Photoelectric sensors

System Configuration



Parts List

Part	Function	Model number
Motor A (B)	Drives conveyor A (B).	---
Rotary encoder A (B)	Encoder for motor A (B)	---
Servodriver A (B)	Servodriver controlling motor A (B)	---
Photoelectric sensors (4)	Sense passing products and boxes.	---
Input Unit	Receives photoelectric sensor inputs.	3G2A5-ID212
High-speed Counter A (B)	Counts the output of rotary encoder A (B).	3G2A5-CT001
Analog Input Unit	Converts analog speed data from servodriver A to digital data.	3G2A5-AD007
Analog Output Unit	Converts the digital output from fuzzy logic processing to analog data and outputs it to servodriver B.	3G2A5-DA002
Fuzzy Logic Unit	Handles fuzzy logic processing.	C500-FZ001

System Operation

The distance that the box travels from the time that it passes the first photoelectric sensor on conveyor B (PH 2) until the product passes the first photoelectric sensor on conveyor A (PH 1) can be calculated from the rotary encoders attached to the motors and High-speed Counter Units A and B. We will call this distance the product/box offset.

The first rules in the Fuzzy Logic Unit will be based on this offset:

> “If the product/box offset is large, then slow down conveyor B.”
>
> “If the product/box offset is small, then don’t change conveyor B’s speed much.”

Fuzzy logic processing using these rules and rules based on the rate of change of the product/box offset allow very smooth adjustment of conveyor B's speed.

Note We assume that only one product or box travels between the photoelectric sensors at a time.

5-2 Making the Knowledge Base

5-2-1 Input Data

The input data for the product/box offset is taken from the relative product/box position (E) calculated as shown below:

E = (Count from rotary encoder A) – (Count from rotary encoder B)

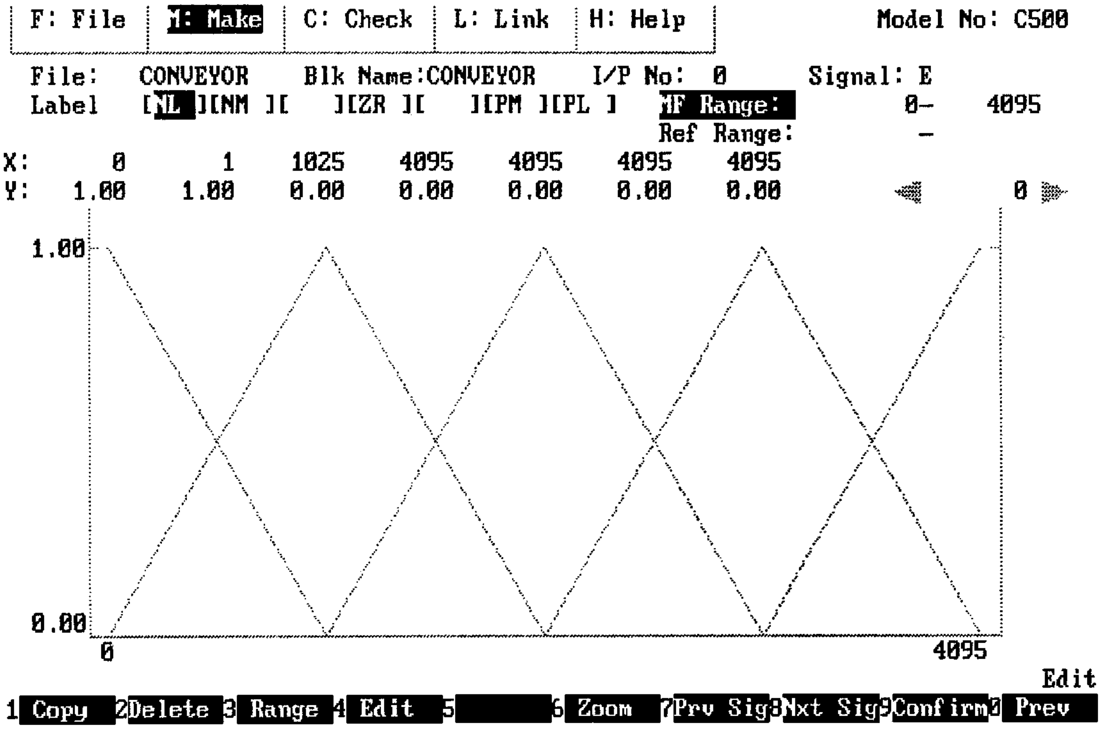
The input data for the rate of change of the offset (ΔE) is simply the difference between the most recent value of E (E_n) and the previous value of E (E_{n-1}) as shown below:

$\Delta E = (E_n) - (E_{n-1})$

5-2-2 Membership Functions

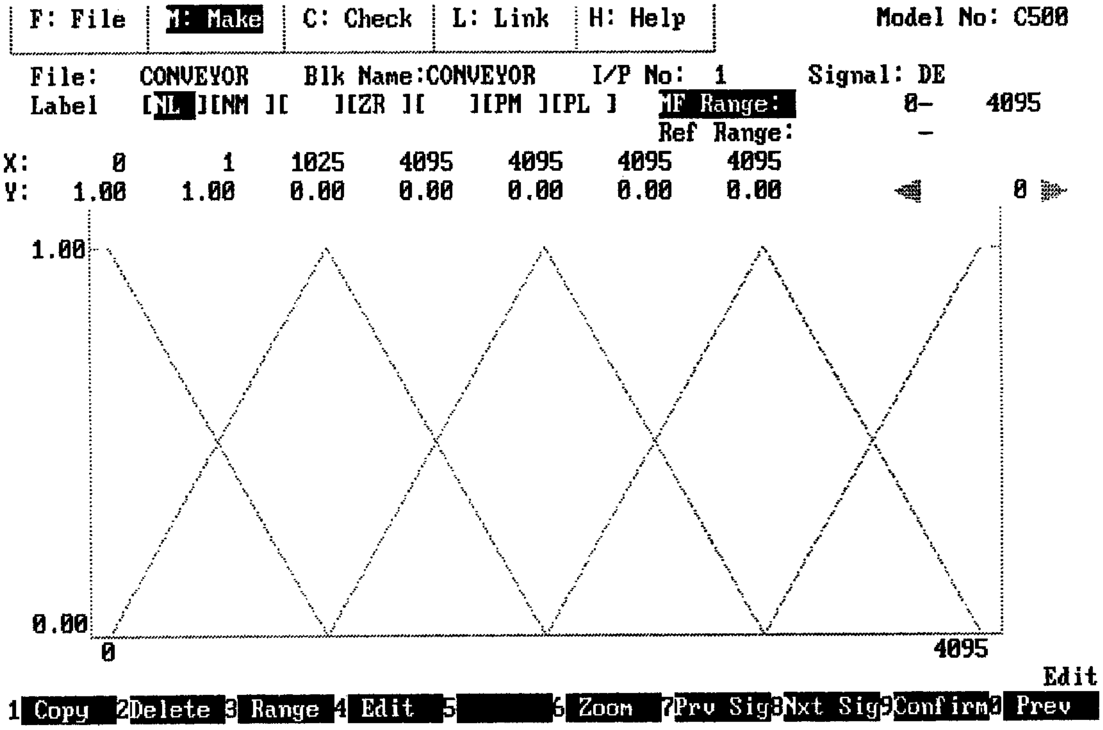
The diagrams in this section show the condition membership functions for the product/box offset and offset rate of change, as well as the conclusion membership function (conveyor B speed adjustment).

Product/Box Offset



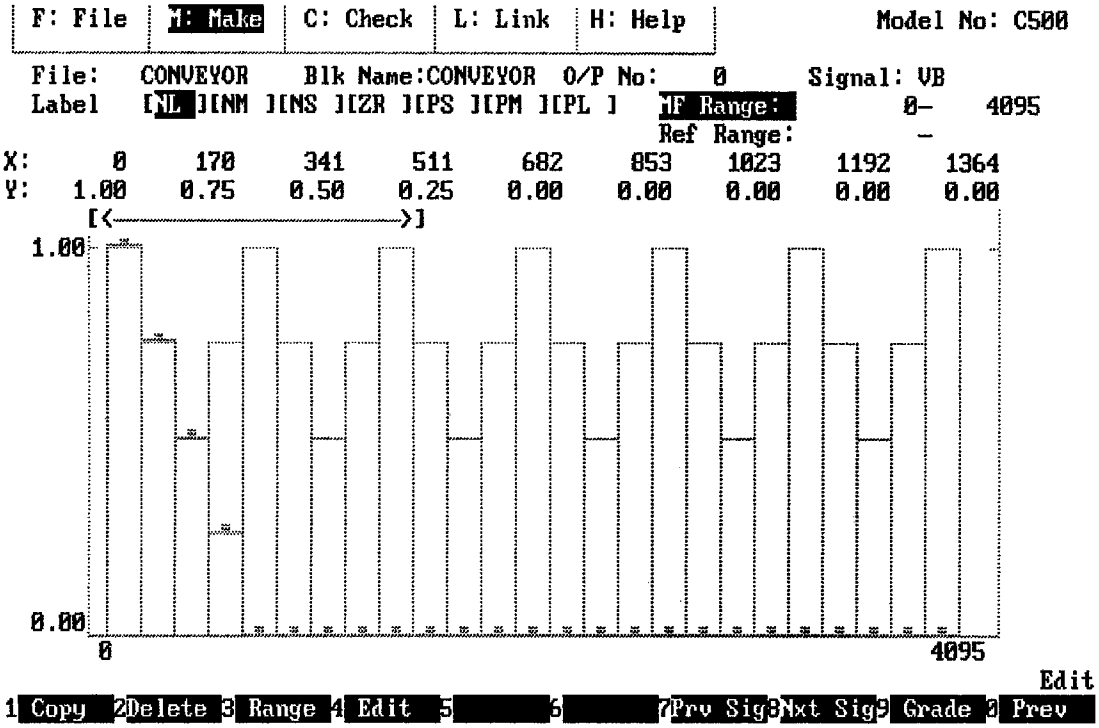
```
                                                          Model No: C500
 File:   CONVEYOR    Blk Name:CONVEYOR    I/P No:  0      Signal: E
 Label   [NL ][NM ][    ][ZR ][    ][PM ][PL ]   MF Range:        0-    4095
                                                 Ref Range:        -
X:      0       1     1025    4095    4095    4095    4095
Y:   1.00    1.00     0.00    0.00    0.00    0.00    0.00               0
```

Offset Rate of Change



```
File:    CONVEYOR     Blk Name:CONVEYOR    I/P No:  1       Signal: DE
Label    [NL ][NM ][      ][ZR ][      ][PM ][PL ]  MF Range:          0-    4095
                                                    Ref Range:           -
X:       0        1     1025     4095     4095     4095     4095
Y:    1.00     1.00     0.00     0.00     0.00     0.00     0.00          ◄       0 ►
```

Conveyor B Speed Adjustment



```
File:    CONVEYOR     Blk Name:CONVEYOR   O/P No:    0      Signal: VB
Label    [NL ][NM ][NS ][ZR ][PS ][PM ][PL ]   MF Range:          0-    4095
                                                Ref Range:           -
X:       0      170      341      511      682      853     1023     1192     1364
Y:    1.00     0.75     0.50     0.25     0.00     0.00     0.00     0.00     0.00
```

5-2-3 Creating Rules

Expressing Rules in Words

Start creating the rules by organizing your “know-how” or “past experience” of the system in everyday expressions. A good way to organize these expressions is a table like the one below.

E ΔE	Box is ahead		About even		Product is ahead
Box is faster	Slow box a lot	Slow box a lot	Slow box a little	Speed it up a little	Speed it up
	Slow box a lot	Slow the box	Slow box a little	Speed it up a little	Speed it up
About even	Slow the box	Slow box a little	Don’t change	Speed it up a little	Speed it up
	Slow the box	Slow box a little	Speed it up a little	Speed it up	Speed it up a lot
Box is slower	Slow the box	Slow box a little	Speed it up a little	Speed it up	Speed it up a lot

Converting to Labels

Next, convert the input data, output data, and the expressions from the table above to labels, as shown below.

Input data:

Product/box offset: E

Offset rate of change: DE (ΔE)

Output data:

Conveyor B speed adjustment: VB

The expressions from the table above can be converted as follows:

E DE	NL	NS	ZR	PS	PL
PL	NL	NL	NS	PS	PM
PS	NL	NM	NS	PS	PM
ZR	NM	NS	ZR	PS	PM
NS	NM	NS	PS	PM	PL
NL	NM	NS	PS	PM	PL

Converting to If/Then

Next, convert the labels from table form to if/then statement form, which can be entered into a computer.

IF E=NL AND DE=PL THEN VB=NL
IF E=NL AND DE=PS THEN VB=NL
IF E=NL AND DE=ZR THEN VB=NM
IF E=NL AND DE=NS THEN VB=NM
IF E=NL AND DE=NL THEN VB=NM
IF E=NS AND DE=PL THEN VB=NM
IF E=NS AND DE=PS THEN VB=NM
IF E=NS AND DE=ZR THEN VB=NS
IF E=NS AND DE=NS THEN VB=NS
IF E=NS AND DE=NL THEN VB=NS
IF E=ZR AND DE=PL THEN VB=NS
IF E=ZR AND DE=PS THEN VB=NS
IF E=ZR AND DE=ZR THEN VB=ZR
IF E=ZR AND DE=NS THEN VB=PS

IF E=ZR AND DE=NL THEN VB=PS
IF E=PS AND DE=PL THEN VB=PS
IF E=PS AND DE=PS THEN VB=PS
IF E=PS AND DE=ZR THEN VB=PS
IF E=PS AND DE=NS THEN VB=PM
IF E=PS AND DE=NL THEN VB=PM
IF E=PL AND DE=PL THEN VB=PM
IF E=PL AND DE=PS THEN VB=PM
IF E=PL AND DE=ZR THEN VB=PM
IF E=PL AND DE=NS THEN VB=PL
IF E=PL AND DE=NL THEN VB=PL

Entering If/Then Statements

Finally, enter the if/then statements into the computer for downloading to the Fuzzy Logic Unit.

SECTION 6
Maintenance and Troubleshooting

This section provides guidelines for dealing with routine maintenance and errors that might occur while using the Fuzzy Logic Unit.

6-1 Maintenance

The items listed in the following table should be inspected regularly to ensure correct operation.

Area	Item	Correct condition
Ambient conditions	Temperature	0° to 55°C
	Humidity	35% to 85% (no condensation)
	Dust or debris	No dust or debris should be allowed to accumulate on the Unit.
Installation	Unit mountings	Check that the Unit is mounted securely.
	Cable connectors	Check that the cable connectors are tight.
	Connecting cable	Check the connecting cable for damage.

Replacing the Battery

The C500–BAT08 battery in the Fuzzy Logic Unit has a life expectancy of 5 years at 25°C. The lifetime will be shorter at higher temperatures.

When the battery begins to fail, the BAT ERR indicator will light. Replace the battery as outlined below within 1 week after the BAT ERR indicator first lights.

1, 2, 3...

1. Turn off the power. If the power was off already, turn it on for at least one minute, then turn it off.
2. Remove the battery cover by pressing it and sliding it down.
3. Remove the old battery and its connector. Replace the old battery with a new one within 5 minutes.
4. When installing the new battery, be sure that you press the connector into its socket firmly.
5. Replace the battery cover. Be sure that the catch on the top of the cover goes into the notch in the Unit.

6-2 Troubleshooting

Error Indications

The table below lists error conditions, their probable cause, and possible remedies.

Indicator(s)	Probable cause	Remedy
No indicators lit.	The PC power supply is OFF.	Connect the PC power supply or check connectors.
	Fuzzy Logic Unit mounting screws are loose.	Tighten screws.
ERROR indicator is lit.	An error has occurred during the Fuzzy Logic Unit's self-diagnostics.	Replace the Unit.
BAT ERR indicator is lit.	The battery isn't connected.	Check the battery contacts and be sure that they are properly connected.
	Battery voltage has dropped below the minimum.	Replace the battery.
ERROR indicator is flashing.	The knowledge base hasn't been input or has been corrupted somehow.	Transfer the knowledge base to the Fuzzy Logic Unit.

Replacing the Unit

We recommend that users have a backup Unit available to make repairs and minimize down-time if a problem occurs with a Fuzzy Logic Unit. Please observe the following precautions in the event of a problem:

- Always turn the power off when replacing a Fuzzy Logic Unit.
- If a Fuzzy Logic Unit fails, replace it with a new one and immediately verify that the new Unit is working properly.

- When returning a malfunctioning Fuzzy Logic Unit for repair, please attach a detailed description of the problem to the Unit and return it to the sales office nearest you (see listing at the end of this publication).
- If you suspect that a poor connection is the cause of a malfunction, clean the connectors using a clean, soft cloth and industrial-grade alcohol. Remove any lint or threads left from the cloth, and re-mount the Fuzzy Logic Unit.

Appendix A
Standard Models

Name	Model number	Notes
Fuzzy Logic Unit	C500-FZ001	---
Battery	C500-BAT08	One Battery is included with the Fuzzy Logic Unit.
Fuzzy Support Software (FSS)	C500-SU981-E	The FSS is available on either 3.5” or 5.25” floppy disks.

Appendix B
Specifications

Item		Specifications
Fuzzy logic processor	I/O capacity	8 inputs and 2 outputs max.
	Rule format	5 condition and 2 conclusion parts max.
	Number of rules	128 rules max.
	Logic process	Forward logic
	Processing time	125 ms per rule max.
	Logic rule	MAX–MIN logical product
	Number of labels	7 max.
	Final calculation	Center of gravity method
Membership functions	Condition	7 inflection points maximum
	Conclusion	The conclusion membership function is made up of 25 columns and the vertical resolution is 2048.
I/O words	Inputs	Each input is allocated one word. Twelve bits of the word are used, so the range is 000 to FFF (0 to 4095 decimal).
	Outputs	Each output is allocated one word.Twelve bits of the word are used, so the range is 000 to FFF (0 to 4095 decimal).
RS-232C communications	Communications	Half duplex
	Synchronization	Start-stop synchronization
	Baud rate	300, 600, 1200, 2400, 4800, 9600, or 19200 (set on the front DIP switch)
	Transmission distance	15 m max. (with a serial cable)
	Interface	RS-232C port (Can be connected to a Link Unit, allowing communication by optical fiber.)
	Communications protocol	Special procedure (1:N)
Processing time		125 ms per rule + 600 ms
Data retained in a power interruption		Rules and membership functions are retained.
Battery	Life expectancy	5 years at 25°C. Shorter at higher temperatures.
	Replacement	Replace within one week after the BAT ERR indicator lights. Data is retained for 5 minutes max. without the battery.
Self-diagnostics	Program check	A “memory error” will be generated if an error occurs during the program check.

Glossary

center of gravity method A method of defuzzification. The center of gravity of the fuzzy outputs (along the x-axis) is calculated as the final result of fuzzy logic processing.

condition part The “if” clauses of a rule. A single rule can have up to 8 condition parts in the form “IF X_1 AND X_2 ... AND X_8.”

conclusion part The “then” clauses of a rule. A single rule can have up to 2 conclusion parts in the form “THEN Y_1 AND Y_2.”

defuzzification Defuzzification is the process that combines the fuzzy outputs to a single result that is output from the Fuzzy Logic Unit. The C500–FZ001 uses the center of gravity method.

fuzzy output After all of the rule grades are calculated, the grade for each label is calculated. The grade for a label is called its fuzzy output. The fuzzy output is the maximum rule grade for that label. The maximum value is taken because the rules are linked by logical ORs.

fuzzy variable A variable such as distance, temperature, or pressure, that is measured by the PC’s sensors. The measured values of the fuzzy variables are input to the Unit for fuzzy logic processing.

grade A number between 0 and 1 that indicates how well the given value of the fuzzy variable satisfies a label. A grade of 0 means the label is not satisfied at all, and a grade of 1 means the label is perfectly satisfied at the given value of the fuzzy variable.

label Labels are used to describe the state of a fuzzy variable. Labels for distance might be “close,” “OK,” and “far.” Labels for temperature might be “cold,” “cool,” “warm,” and “hot.” Labels for pressure might be “light,” “moderate,” and “hard.” The C500–FZ001 can have up to 7 labels for each fuzzy variable.

knowledge base The rules and membership functions.

membership function Defines the relationship between the grade and the fuzzy variable for each label. The most accurate shape for a membership function is usually a bell curve, but triangles and trapezoids are used most often to simplify and speed up processing.

rule An if/then statement that expresses a relationship between inputs and outputs. The C500–FZ001 can have up to 128 rules.

rule grade Membership functions assign grades to the labels of the condition parts of a rule according to the present conditions. The rule grade is the minimum of these grades.

Index

Revision History

C500-FZ001 Fuzzy Logic Unit
Operation Manual

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.

Cat. No. W209-E1-1A

Revision code

The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	March 1992	Original production
1A	July 1993	Pages 2, 20 : Minor changes to add CV2000 and CVM1.