DEVELOPMENT AND IMPLEMENTATION OF A FUZZY PROGRAMMABLE LOGIC CONTROLLER

AREIF AZREE BIN MOHD AZAHARI

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

DEVELOPMENT AND IMPLEMENTATION OF A FUZZY PROGRAMMABLE LOGIC CONTROLLER

By

AREIF AZREE BIN MOHD AZAHARI

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

DEVELOPMENT AND IMPLEMENTATION OF A FUZZY PROGRAMMABLE LOGIC CONTROLLER

by

Areif Azree Bin Mohd Azahari

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

____

Assoc. Prof. Dr. Nordin Saad
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

JUNE 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Areif Azree Bin Mohd Azahari

ABSTRACT

Programmable Logic Controller (PLC) has been used in many type of automation for many years. PLC provides a compatible input and output device that easily find in the market. Although PLC has been introduced in many years, it is not an intelligent device that can control things without other factor to control the decision. Although the arithmetic operations are available however it still not considered to be intelligent. Thus the introduction of Fuzzy logic in PLC may provide the intelligent. These could produce the decision making facility that can be programmed into PLC. Fuzzy logic is a superset of Boolean logic. The purpose of control is to influence off to dictate the behavior of a system by changing any input to the system according to rules of the system. With this Fuzzy controller, the control signal to the final element is the outcome of the produced by the PLC based on Fuzzy reasoning. The Fuzzy-PLC is utilized to control a process which is in this case a level process plant.

ACKNOWLEDGEMENT

I would like to thank my supervisor Assoc. Prof. Dr. Nordin Saad for giving support and guidance throughout the duration of this project. He motivates and coaches me for the entire time under his supervision. I would like to thank Mr Arrofiq, postgraduate student for guiding me and never refused upon answering any question regarding my project. To En Azhar, the lab technologist, thank you for helping me on the PLC lab and the moral support you have given. Not to forget Ms Siti Hawa Hj Mohd Tahir, thank you for the correcting formats of my project. Finally to both of my parents and friends, thank you for all the advices, moral support and guidance which will not be forgotten.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATION

PLC	Programmable Logic Controller
DCS	Distributed Control System
FYP	Final Year Project
BCD	Binary Coded Decimal
HR	Holding Relay

CHAPTER 1
INTRODUCTION

1.1 Background of study

Programmable Logic Controller (PLC) is a special form of microprocessor-based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes. [1]. The functions of PLC have evolved over the years which include process control, sequential relay control and many more.

However, PLC only restricted to control and cannot make a decision on the system. In recent years, the Distributed Control System (DCS) is the one that make the decision for PLC e.g. Open or close valve if any alarm occur. Therefore by including Fuzzy in PLC will produce the decision as instructed in the PLC.

1.2 Problem statement

Fuzzy Control has been one of the important methodology / approach for controlling and monitoring. However, implementation this approach in real application is difficult due to interfacing issues between the Fuzzy algorithm and real world. PLC is limited to Boolean logic. With Fuzzy logic in the PLC, it can act as intelligent controller.

1.3 Objective

The objective of this project is design and implement Fuzzy control algorithm onto a basic PLC. The developed Fuzzy PLC then applied to the control of a single process.

1.4 Scope of study

The scopes of study of this project are:

- The concept of Programmable Logic Controller (PLC) and the programming.
- The Fuzzy Logic concept.
- Interoperate the Fuzzy in PLC.
- Implement Fuzzy PLC in-real devices.

CHAPTER 2
LITERATURE REVIEW

2.1 Fuzzy Logic

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth or truth values between completely truths and completely false. Fuzzy logic is also a form of multi-valued logic that derived from the Fuzzy set theory that dealing with reasoning which is approximate not precise. The binary logic or Boolean logic in other words in Fuzzy logic terms, crisp logic, the Fuzzy logic variables may have the membership values from 0 to 1. This mean the Fuzzy set theory, the membership values can be range from 0 to 1 itself. Moreover the degree of truth statement can be between 0 and 1 which is not restricted to only true (1) or false (0).

2.1.1 Fuzzy set

The degrees of membership are the elements in the Fuzzy set. The Fuzzy set was introduced by Lotfi A. Zadeh. Fuzzy set is the extension of the normal mathematical set. In the theory of Fuzzy set, the characteristic is generalized to a membership function that assigns to every $u \in U$ a value form the unit interval [0, 1] instead two element set {0, 1}. [2]. It is a process to convert from crisp input to the Fuzzy values. There are few membership function shape, examples triangular, trapezoidal and bell shape.

2.1.2 Fuzzy Logic Controller

The purpose of control is to influence the behavior of a system by changing the inputs to that system according to a rule or set of rules that model how the system operates. There are several ways the system can be operates such as mechanical, electrical and etc. The below figure is the simple structure of Fuzzy logic control.

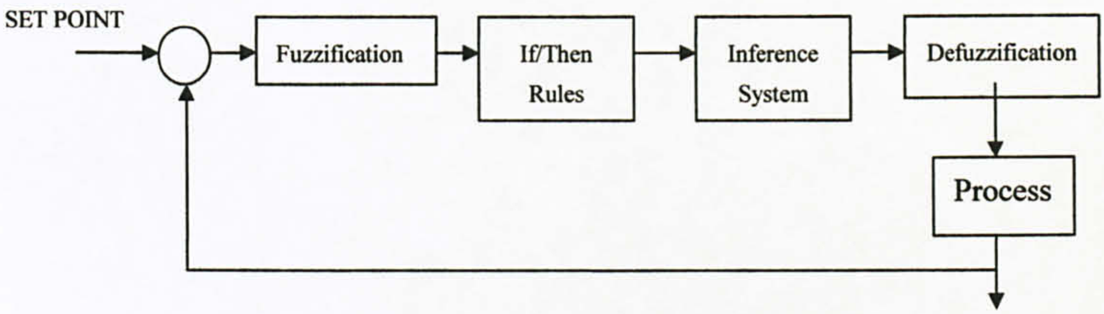


Figure 1: Fuzzy Logic Controller Structure

2.1.2.1 ***Fuzzification***

Fuzzification is transforming a crisp input set into Fuzzy set. This will determine the degree to which the inputs belong to each of the appropriate Fuzzy sets through a membership function. The data manipulation in a Fuzzy logic controller is based on Fuzzy set theory, fuzzification is necessary at an early stage of Fuzzy logic controller. The fuzzification block has to match the input data together with the conditions of the rules to measure the how well the condition of each rule match to that particular input. There is a degree of membership for each linguistic term that applies to that input variable. [4]

2.1.2.2 ***If-Then Rule***

Several variables maybe used in both the conditions and the conclusion. The controller can be applied in both multi input multi output and single input (MIMO) and single input and single output (SISO). The normal SISO is to regulate the control signal from the error signal. The controller may need the error, the change in error, and the accumulated error as inputs. These errors are from the error measurement. The if-then rule statements are use to make the conditional statements that comprise Fuzzy logic. A single Fuzzy if-then rules: *if path is left then steer is positive.* The left and positive are linguistic values defined by Fuzzy sets on the ranges left and steer. The if – part of the rule "path is left "is called the antecedent or premise, while the then-part of the rule "steer is positive is called the consequent or conclusion.

2.1.2.3 ***Inference system***

The inference system block is where the Fuzzy process the formula mapping from an input to an output. The formula mapping provides a basis from which decision can be made or patterns choose. The process of inference all of the previous described section which is the membership function, Fuzzy logic and if-then rules.

2.1.2.4 ***Defuzzification***

Defuzzification is a mapping from the Fuzzy control. Many practical application crisp control action defined over an output universe of discourse into space of crisp control action. There are five types of defuzzification which are Centre Of Gravity (COG), Centre of Gravity Method for Singleton (COGS), Bisector of Area (BOA), Mean of maxima (MOM) and Leftmost maximum (LM) and Rightmost maximum (RM). In order to do defuzzification, the author has to choose from one of it. These methods will produce different shape of Fuzzy set but the relative of complexity is small.

$$U = \frac{\sum_{i=1}^{l} u_i . max\mu(u_i)}{\sum_{i=1}^{l} max\mu(u_i)}$$

2.2 Programmable Logic Controller (PLC)

Programmable Logic Controller (PLC) is an industrial purpose computer designed for use in control of a wide variety of manufacturing machines and system. PLC control application varies from one another. The on/off of a pump motor using level switch to control of a conveyor system. The figure 2 shows the structure of a PLC.

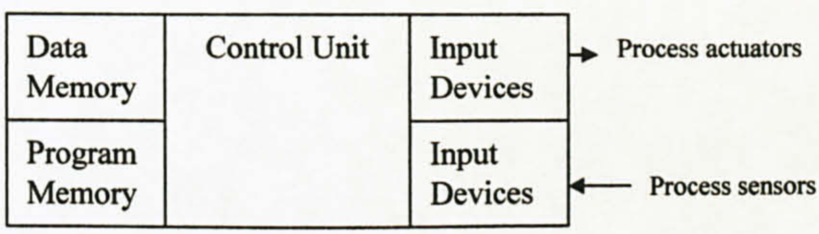


Figure 2: Structure of PLC

2.2.1 Programming Language

There six types of programming language in PLC standard. They are Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL) and Sequential Function Charts (SFC). This programming language depends on the type of PLC will be use in the author project.

2.2.2 *Ladder Diagram*

The programming language that the author will be using is ladder diagram. Ladder diagram is graphical representation of relay logic in the hardware. The author will try to incorporate between the Fuzzy logic and the ladder diagram in PLC. It is basically based on the figure 3.

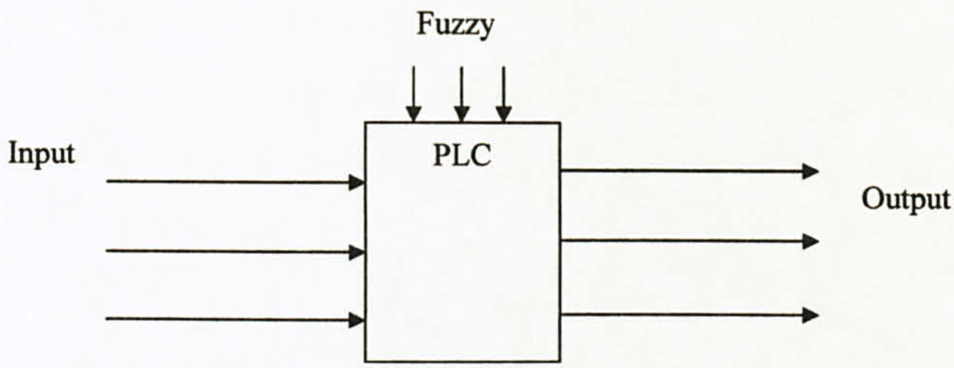


Figure 3: Fuzzy PLC structure

2.2.3 *Instruction set*

2.2.3.1 *Compare – CMP (20)*

Ladder Symbol

CMP (20)
Cp1
Cp2

Operand Data Areas

Cp1: First compare word
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
Cp2: Second compare word
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #

Each of the instruction has their own limitation for each of them. As for the compare instruction CMP (20), the limitation is when comparing values to the PV of a timer or counter, the value must be in Binary Coded Decimal (BCD). The compare instruction description is when the execution is OFF or the rung is not ON, CMP (20) will not execute. When the rung is ON, CMP (20) compares the

Cp1 and Cp2 and the result to the GR, EQ and LE flags in the SR area. However, Flags may change the status when placing other instruction between CMP (20) and the operation on assesses the EQ, GR and LE.

Example:

EQ : ON if Cp1 equals Cp2

LE : ON if Cp1 is less than Cp2

GR : ON if Cp1 is greater than Cp2

Table 1: Flags Condition Table

Flag	Address	CP1 < Cp2	Cp1 = Cp2	Cp1 > Cp2
GR	25505	OFF	OFF	ON
EQ	25506	OFF	ON	OFF
LE	25507	ON	OFF	OFF

The above table the result of compare between Cp 1 and Cp2. The flags based on the condition.

2.2.3.2 Move – MOV (21)

Ladder Symbol

MOV (21)
S
D

Operand Data Areas

S: Source word
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
D: Destination word
IR, SR, AR, DM, EM, HR, LR

The limitation for this instruction is the memory area of DM is it cannot be used the memory from DM6144 to DM6655 for D (destination word). The description is when the rung is not activated or execute, MOV (21) will not execute. When the rung is execute, the content of S copies to D.

2.2.3.3 BCD Add – ADD (30)

Ladder Symbol

ADD (30)
Au
Ad
R

Operand Data Areas

Au: Augend word (BCD)
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
Ad: Addend word
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
R: Result word
IR, SR, AR, DM, EM, HR, LR

In this instruction, the limitation is the memory area DM6144 to DM6655 cannot be used for R. When the execution condition is ON, ADD (30) adds to contents of Au and Ad and places the results in R. The operation is as below

$$Au + Ad = R$$

2.2.3.4 BCD Subtract – SUB (31)

Ladder Symbol

SUB (31)
Mi
Su
R

Operand Data Areas

Mi: Minuend word (BCD)
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
Su: Subtrahend word
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
R: Result word
IR, SR, AR, DM, EM, HR, LR

In this instruction, the limitation is the memory area DM6144 to DM6655 cannot be used for R. As the execution is ON, SUB (31) subtracts the contecnt Su from Mi and place the result at R

$$Mi - Su = R$$

2.2.3.5 BCD Multiply – MUL (32)

Ladder Symbol

MUL (32)
Md
Mr
R

Operand Data Areas

Md: Minuend word (BCD)
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
Mr: Subtrahend word
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
R: Result word
IR, SR, AR, DM, EM, HR, LR

When the execution condition is ON, MUL (32) multiplies Md by the content of Mr and places the result in R and R + 1. The limitation is on the memory area in DM6143 to DM6655.

$$Md \times Mr = R$$

2.2.3.6 BCD Divide – DIV (33)

Ladder Symbol

MUL (32)
Md
Mr
R

Operand Data Areas

Md: Minuend word (BCD)
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
Mr: Subtrahend word
IR, SR, AR, DM, EM, HR, TIM/CNT, LR, #
R: Result word
IR, SR, AR, DM, EM, HR, LR

When execution condition is ON, DIV (33) by the content of Dr and Dd places the result in R. The limitation is on the memory area in DM 6143 to DM6655

$$Dd \div Dr = R$$

2.2.4 Memory Area

There are many types of memory area in PLC. Each memory area has own functionality and purposes in order for a program to work. In this project the author will use the DM memory area. Data assess in memory in word units. There is certain address can be used as read/write that part of DM area can be freely read and written from the program. Refer to the appendix for reference of the address memory area.

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

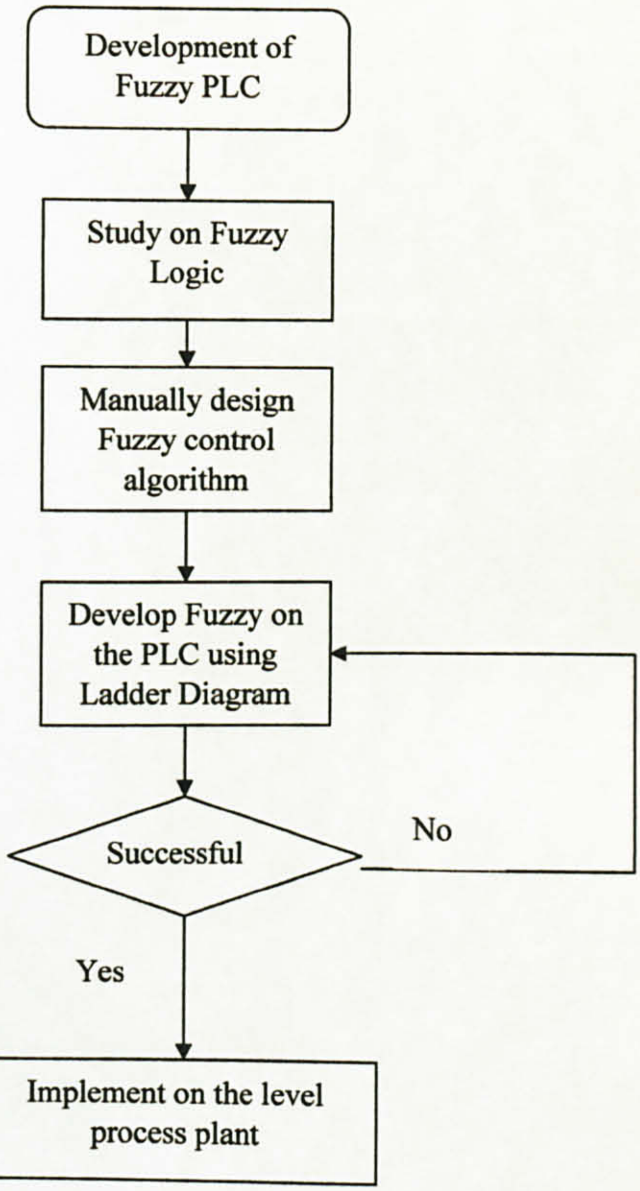


Figure 4 : Flow chart

3.2 Tool and Equipment

Hardware:

- Omron PLC

Software:

- Automation Studio 5.0
- CX- Programmer

3.3 Description of the plant

The process to be control by the Fuzzy PLC is a level control system. The main component involve in this system is, Level Transmitter signal (LT1 and LT2), the controller (PLC) and the control elements control valve (Va and Vb).

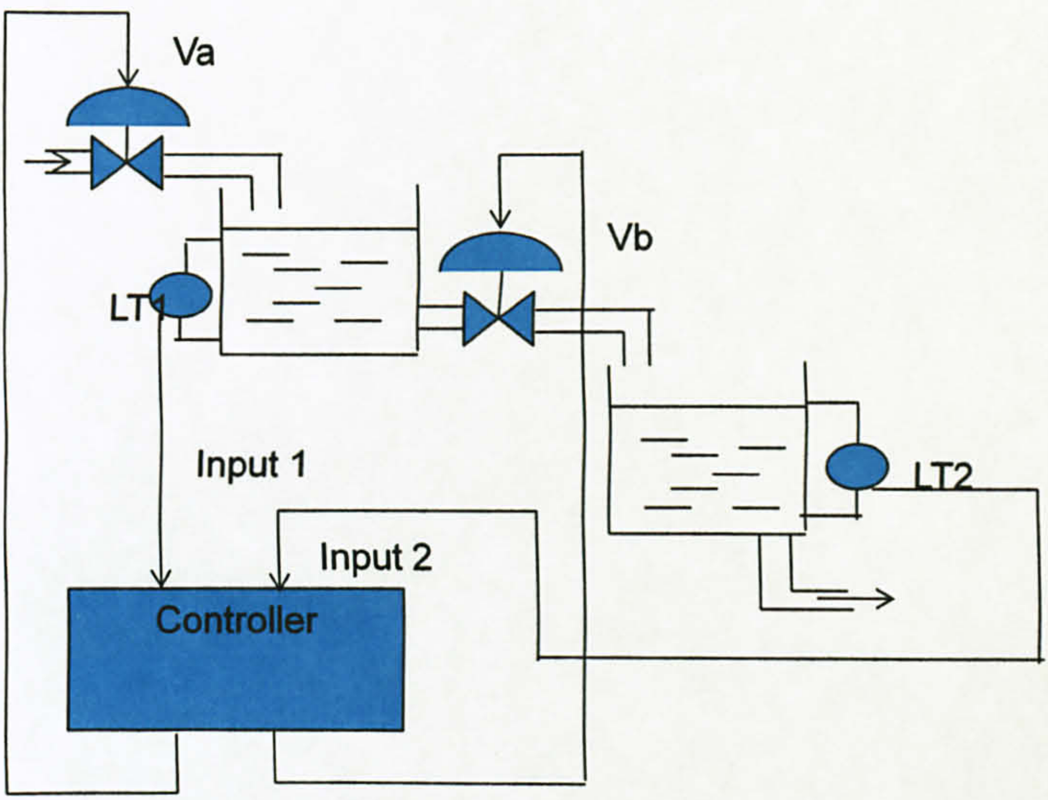


Figure 5 : Level plant system

The variable involve is the water level in the tank and is called the process variable of the process variable (PV) or the controlled variable (CV), i.e., the variable of the process that needs to be controlled or maintained at some desired value or set point.

In order to maintain the water level at the desired value, the Va and Vb current signal (4-20mA) is adjusted to adjust the level in the both tank. This current is the manipulated variable that is to be controlled via the PLC to control the valve and give the set-point to the tank.

As can be seen in the diagram, the water level will be maintained as long as water level from both tanks remains unchanged or equal. If any disturbance occurs that can result to change in water level in tanks, thus this will make the level in different steady – state level. Thus, Fuzz-PLC will be programmed to adjust the Va and Vb so that the water level will be maintained or settle at the required set – point.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Result

Example

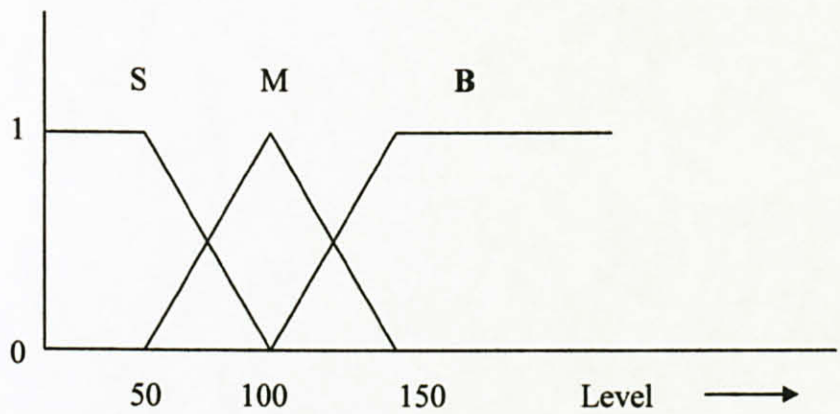


Figure 6 : Membership Input

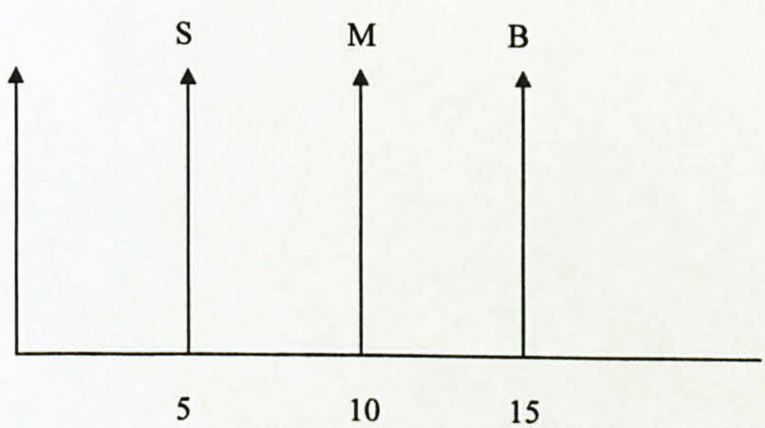


Figure 7 : Membership output

Table 2 : Rules

		INPUT 1		
INPUT 2		**S**	**M**	**B**
	S	S	S	M
	M	S	M	B
	B	M	B	B

Fuzzification

Given input 1 = 30, input 2 = 90

Let μ_S = membership function for small

μ_M = membership function for medium

μ_B = membership function for big

Input1 = 30

$$\mu_s\,(In1);\ \frac{50-30}{50-0} = \frac{1-x}{1}$$

$$\frac{20}{50} = 1 - x$$

$$2 = 5 - 5x$$

$$x = \frac{3}{5}$$

$$= 0.6$$

$\mu_M\,(In1) = 0\,;\ \mu_B\,(In1) = 0$

Input2 = 90

$$\mu_s\,(In2);\ \frac{100-90}{100-50} = \frac{0-x}{0-1}$$

$$\frac{10}{50} = x$$

$$x = 0.2$$

$$\mu_M\,(In2);\ \frac{100-90}{100-50}=\frac{1-x}{1}$$

$$\frac{10}{50}=1-x$$

$$x=0.8$$

$\mu_M\,(In2) = 0$

$\mu_s\,(In1) = 0.6$	$\mu_s\,(In2) = 0.2$
$\mu_M\,(In1) = 0$	$\mu_M\,(In2) = 0.8$
$\mu_B\,(In1) = 0$	$\mu_B\,(In2) = 0$

Min Max Rule

Rule 1	min (μ_{1S}, μ_{2S}) =	min (0.6, 0.2)	= 0.2 (S)
Rule 2	min (μ_{1M}, μ_{2S}) =	min (0, 0.2)	= 0 (S)
Rule 3	min (μ_{1B}, μ_{2S}) =	min (0, 0.2)	= 0 (M)
Rule 4	min (μ_{1S}, μ_{2M}) =	min (0.6, 0.8)	= 0.6 (S)
Rule 5	min (μ_{1M}, μ_{2M}) =	min (0, 0.8)	= 0 (M)
Rule 6	min (μ_{1B}, μ_{2B}) =	min (0, 0.8)	= 0 (B)
Rule 7	min (μ_{1S}, μ_{3S}) =	min (0.6, 0)	= 0 (M)
Rule 8	min (μ_{1M}, μ_{3M}) =	min (0, 0)	= 0 (B)
Rule 9	min (μ_{1B}, μ_{3B}) =	min (0, 0)	= 0 (B)

Rule	Output	μ_{out}
Rule 1	S	0.2
Rule 2	S	0
Rule 3	M	0
Rule 4	S	0.6
Rule 5	M	0
Rule 6	B	0
Rule 7	M	0
Rule 8	B	0
Rule 9	B	0

Defuzzification

Centre of Gravity (COG) , $U = \frac{\sum_{i=1}^{l} u_i . max\mu(u_i)}{\sum_{i=1}^{l} max\mu(u_i)}$

S = max (0.2, 0, 0.6) = 0.6

M = max (0, 0, 0) = 0

B = max (0, 0, 0) = 0

$$\sum_{i=1}^{3} \frac{U_i . max\mu(U)}{max\mu(U)} = \frac{(5\times0.6)+(10\times0)+(15\times0)}{0.6}$$

$$= 5$$

4.2 Discussion

This Fuzzy-PLC has been developing successfully. Before the Fuzzy-PLC develops, the Fuzzy controller has to develop manually. From the trend, the Fuzzy is developing to Fuzzificaiton. Fuzzification is to convert the input to the membership function. This transfer then has to be in expression or mathematic algorithm.

Then for the "if – then" rules, the rules need to be set as the specific output. After that table them so that can do the if then rules. For thi, a flow chart will help in for the ladder diagram.

Then after the rules, the selection for the max and min based on the rules so that can do the defuzzification. The defuzzification is a process to convert vack to value of the input to the process but output for Fuzzy controller. The selection type of defuzzification is the center of gravity and uses the formula to get the output.

Then after does the Fuzzy controller manually, convert all the process to the Ladder Diagram in CX-programmer. Use the instruction based on the respective characteristics.

Pre-process of Fuzzy PLC is a process before the process of Fuzzy is done. Hence using aflowchart will make the understanding better. Pre-process is to set the value if in the case of level less than 50 and more than 150. These values tend to have the membership function equal to 1. The region of interest for the Fuzzy process is in 50 to 150 regions. The flow chart of pre-process is as below

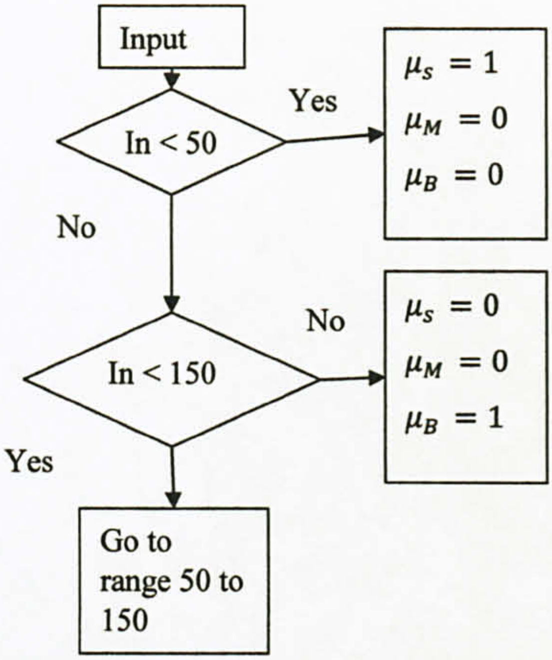


Figure 8: Flow chart for Pre - process

After the pre-prcess, the fuzzification process will be done. Based on the flow chart you will see the step to understand when it converts to PLC.

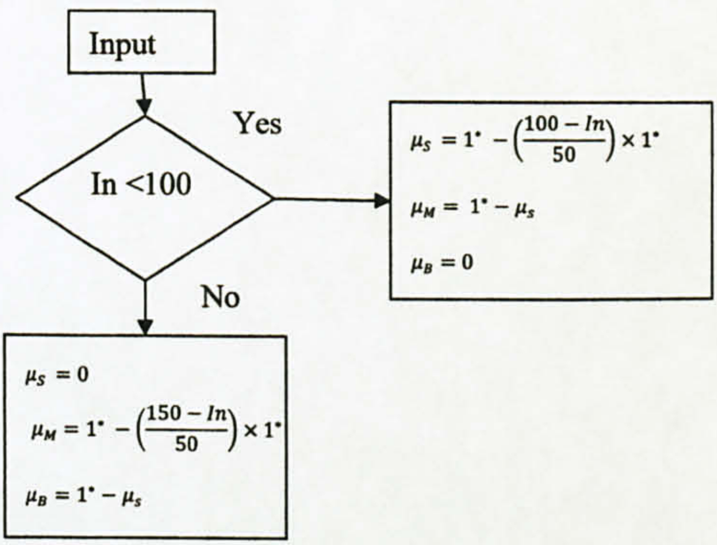


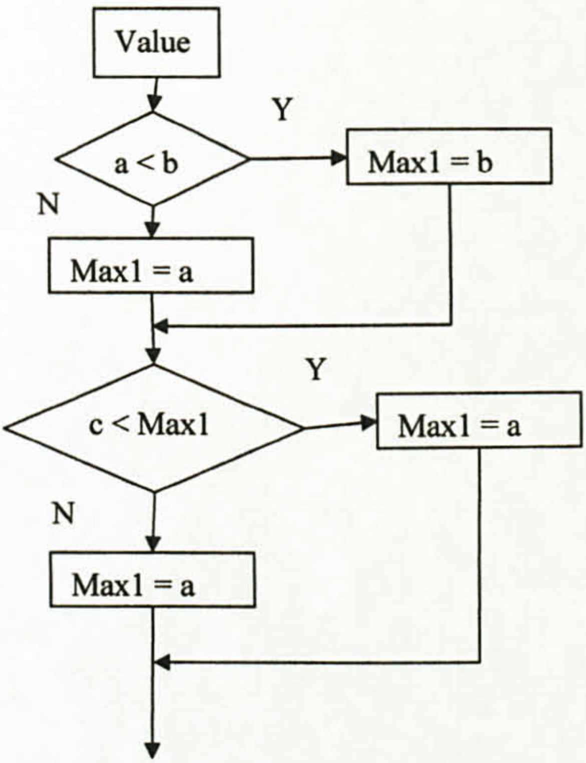
Figure 9 : Flowchart for Fuzzification

The min and max rule is one of the types in the rule for Fuzzy process. In this rule to convert to ladder just have to use the compare instruction then plays around with the memory.

After compare the min and max rule then have to choose the max for the defuzzification process used centre of gravity equation (2)

Centre of Gravity (COG) , $U = \frac{\sum_{i=1}^{l} u_i . max\mu(u_i)}{\sum_{i=1}^{l} max\mu(u_i)}$ **(2)**

The flowchart is as follows:



CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The understanding of Fuzzy is in progress. It is a challenging task to understand it in a short period. As for current understanding, Fuzzy is a logic that can extend the Boolean logic understanding. Previously Boolean logic can define 0 to 1 but with the Fuzzy logic, the output is extending to a range from 0 to 1.

PLC is a tool for the industrial application. Most of it used for control purposes. PLC is a microcontroller-based which can store memory for the programming. The programming language used is ladder diagram. Ladder diagram is easy to understand because it uses the graphical representation.

The Fuzzy-PLC algorithm already successfully done in CX-programmer. The real application still needs to be done in the future. The program that has been done is based on the current case study. Different case study needs different algorithm to be written in manual way and in the CX-programmer.

5.2 Future work

In future, the program for implementing logic into PLC can still be improved. The reliability of the PLC to implement the Fuzzy control should be verified and validated through real case studies. Other control techniques for example adaptive control algorithm should be exploited.

Further work on Fuzzy-PLC to control a simple plant should be conducted based on the design outcome of this project.

REFERENCES

[1] W. Bolton, 2006 , *Programmable Logic Controller Fourth Edition* , Newnes

[2] James A Rehg, Glenn J. Sartori, 2009, *Programmable Logic Controller Second Edition*, Pearson Prentice Hall

[3] D. Driankov, H Hellendoorn, M. Reinfrank, 1993, *An Introduction to Fuzzy Control Second Edition*, Springer

[4] Jan Jantzen, PDF file, Design of Fuzzy Controllers

[5] Omron programming manual, *W364-E1-05-CGM1H Series*

APPENDICES

APPENDIX A

PROJECT MILESTONE

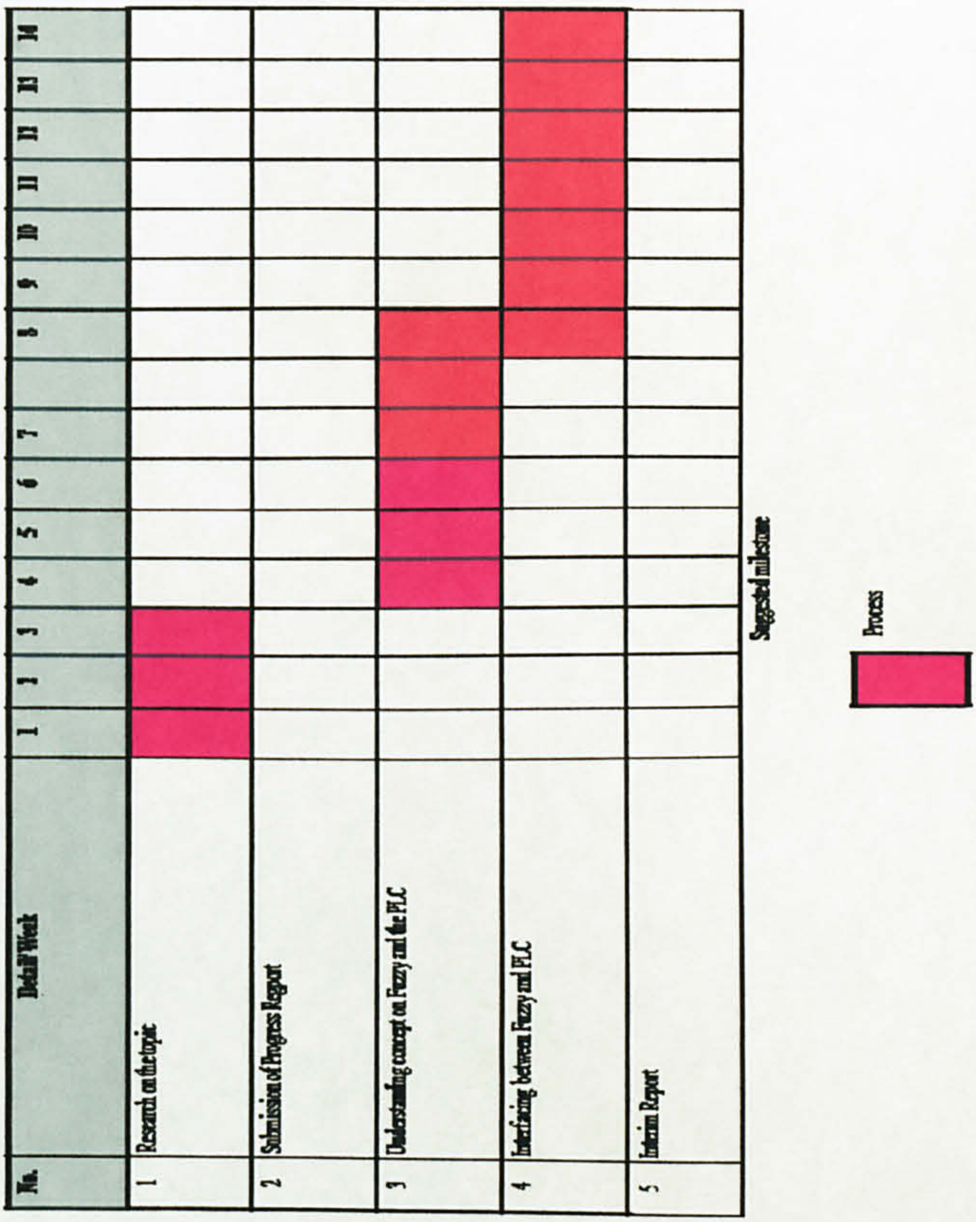


Figure 10 : Project Milestone FYP1

Work	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Understanding the Instruction Set in PLC														
Progress Report 1														
Convert Fuzzification to PLC														
Progress Report 2														
Pre-Process														
Convert Rules to PLC														
Convert Defuzzification to PLC														
Combine all Fuzzy to PLC														
Draft Report														
Final Draft Report														

Figure 11: Project milestone for FYP2

APPENDIX B

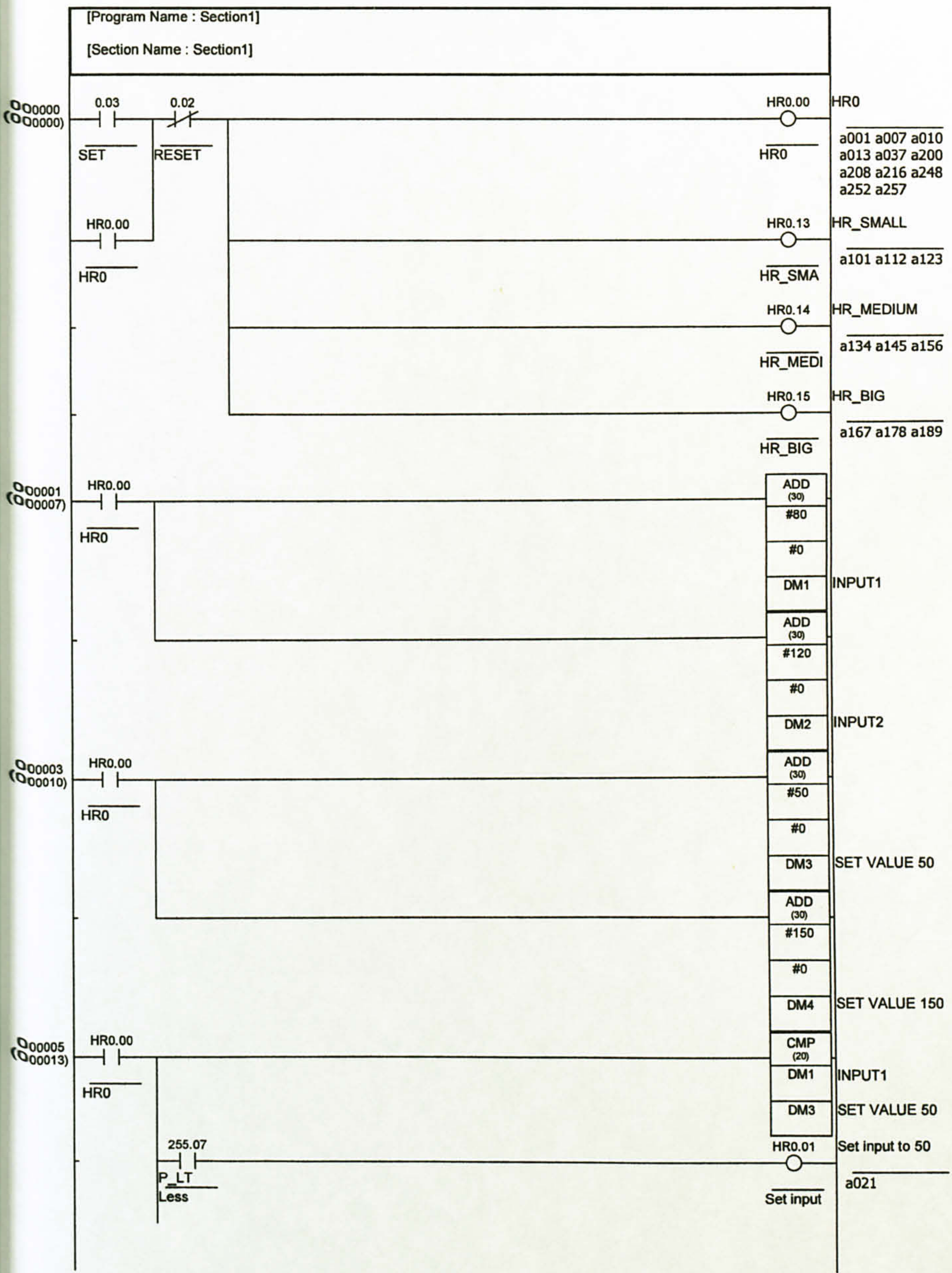
LIST OF MEMORY AREA

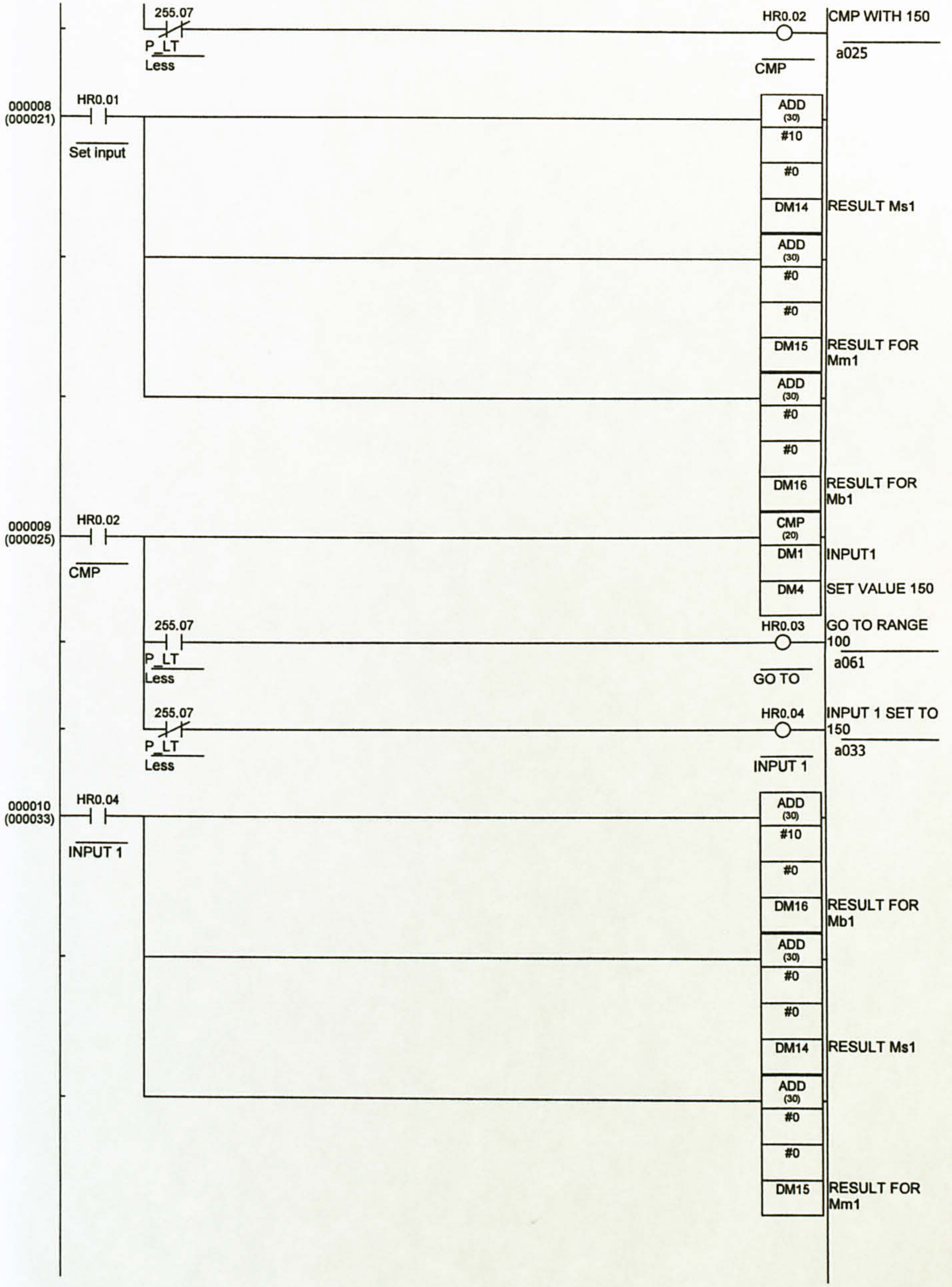
HR0	1.HR for input 1 and input 2 2.HR for initial value for 50 and 150
DM1	Input 1
DM2	Input 2
DM3	Initial value set 50
DM4	Initial value set 150
HR0.01	If the input 1 less than 50 set the input to 50
HR0.02	If the input 1 not less than 50, the input then compare with 150
HR0.03	If the input 1 less than 150, got to range 100
HR0.04	If the input 1 not less than 150 then set the input 1 to 150
HR0.05	If the input 2 is less than 50 then set input 2 to50
HR0.06	If the input 2is not less than 50 then compare with 150
HR0.07	If the input 2 is less than 150, go to range 100
HR0.08	If the input 2 is not less than 150 set the input 2 to 150
HR0.09	Form HR0.03 compare with 100
DM12	Result for SUB(input 1)
DM13	Result for DIV(input1)
DM14	Result for Ms1
DM15	Result for Mb1
DM18	Result for SUB2
DM19	Result for DIV2
HR0.11	HR for less than 100 input 2
HR0.12	HR for not less than 100 input 2

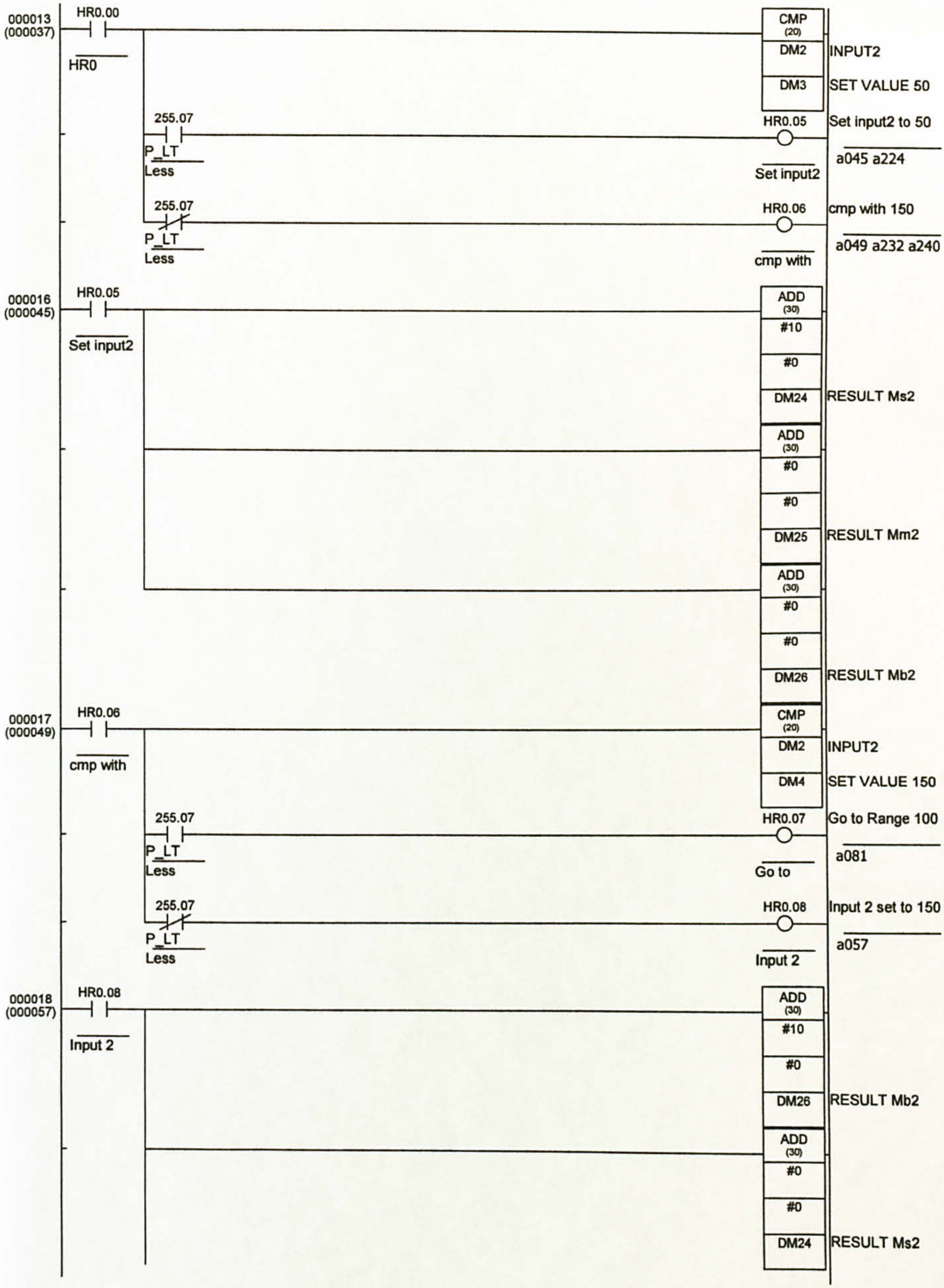
DM22	Result for SUB1(input 2)
DM23	Result for DIV1 (input2)
DM24	Result for Ms2
DM25	Result for Mm2
DM26	Result for Mb2
DM28	Result for SUB2
DM29	Result for DIV2
DM32	R_SMALL1
DM33	R_SMALL2
DM35	R_SMALL4
DM36	R_MEDIUM3
DM37	R_MEDIUM5
DM38	R_MEDIUM7
DM39	R_BIG6
DM40	R_BIG8
DM41	R_BIG9
DM44	CMP WITH C (SMALL)
DM45	MAX_SMALL
DM46	CMP WITH C (MEDIUM)
DM47	MAX_MEDIUM
DM48	CMP WITH C (BIG)
DM49	MAX_BIG
DM50	DEFUZ_SMALL
DM51	DEFUZ_MEDIUM
DM52	DEFUZ_BIG
DM53	DEFUZ_S+M
DM54	DEFUZ_S+M+B
DM55	MAX_+M
DM56	MAX_S+M_B
DM57	RESULT DEFUZ

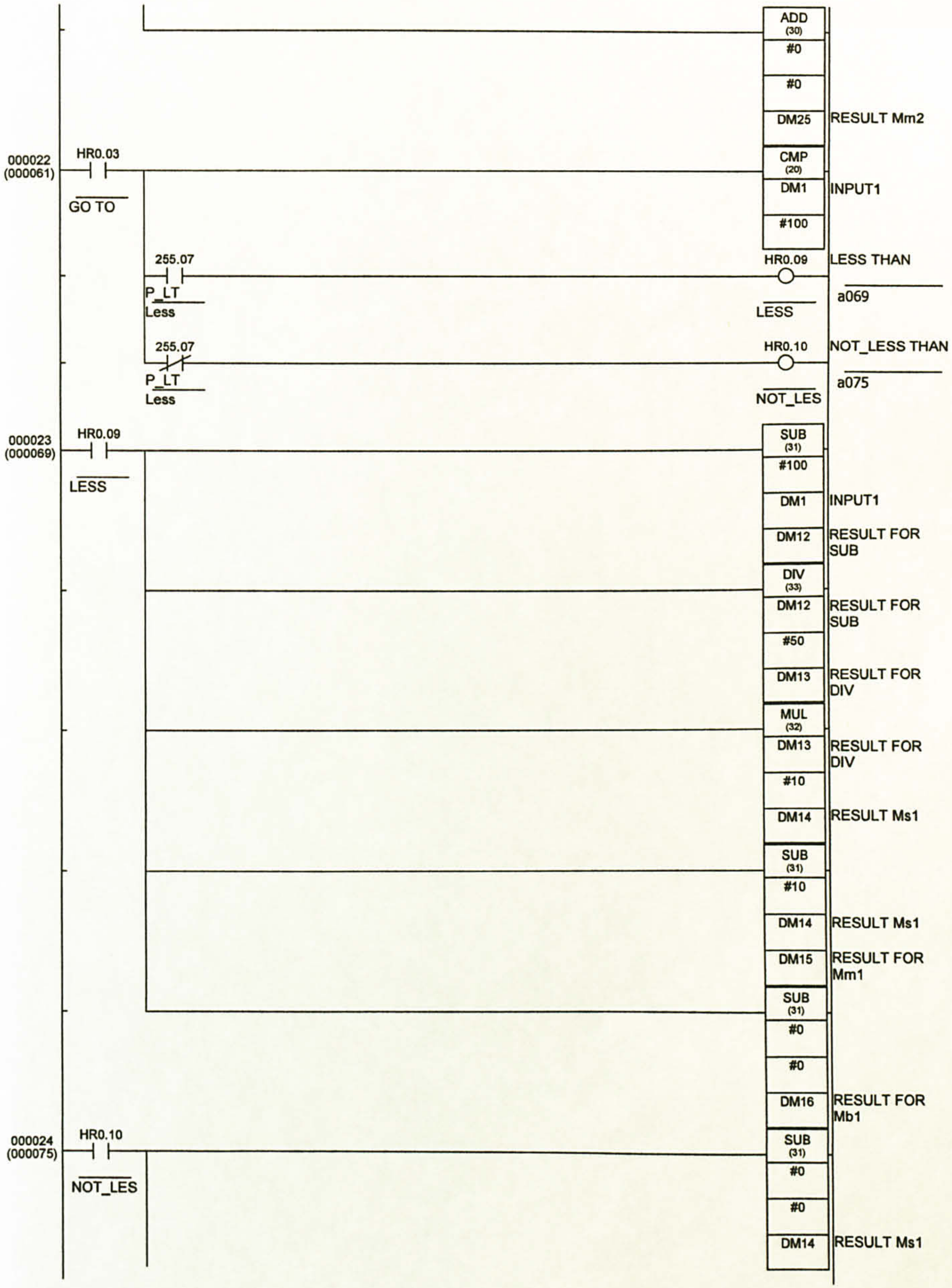
APPENDIX C

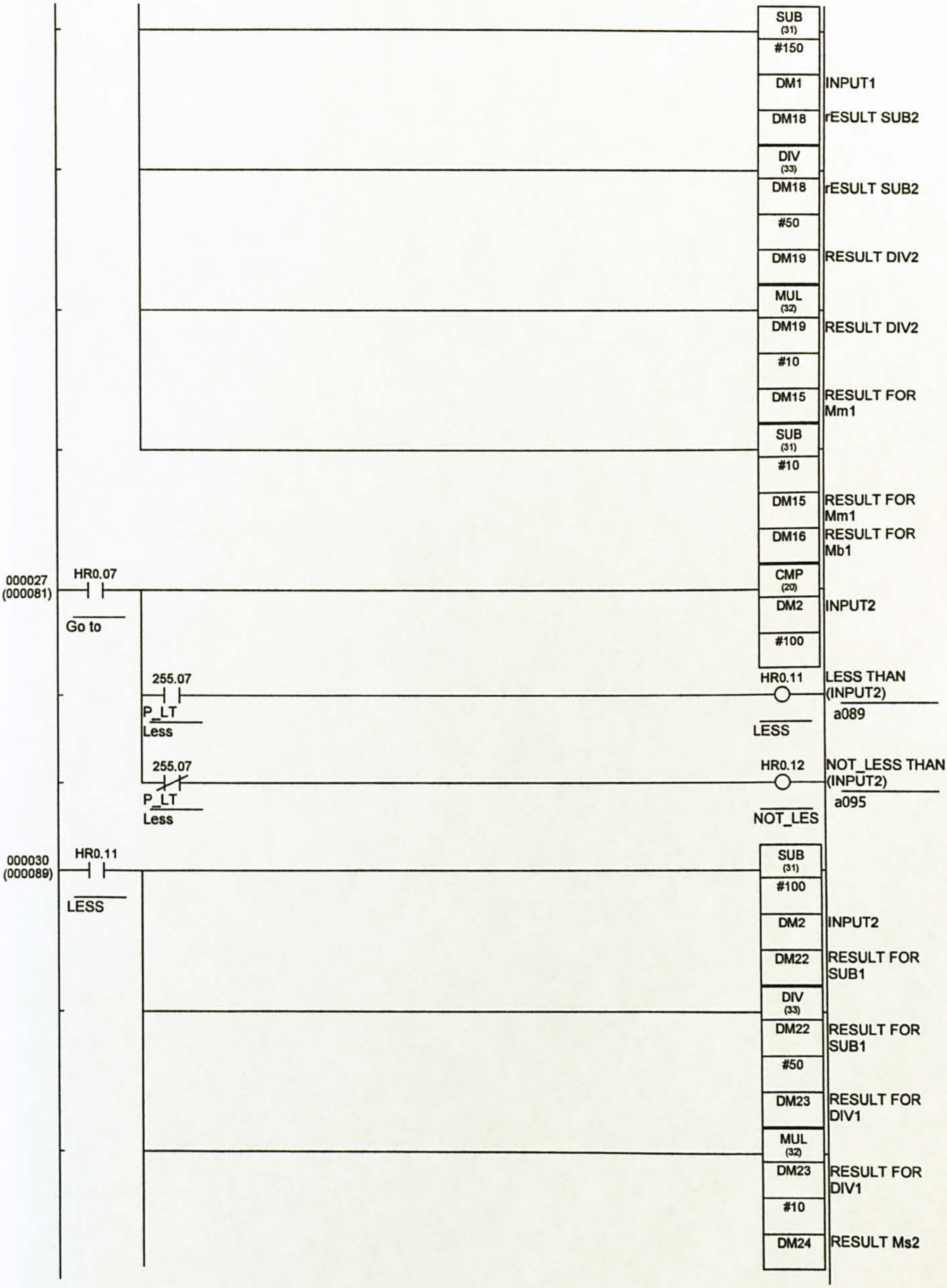
LADDER DIAGRAM FUZZY PLC

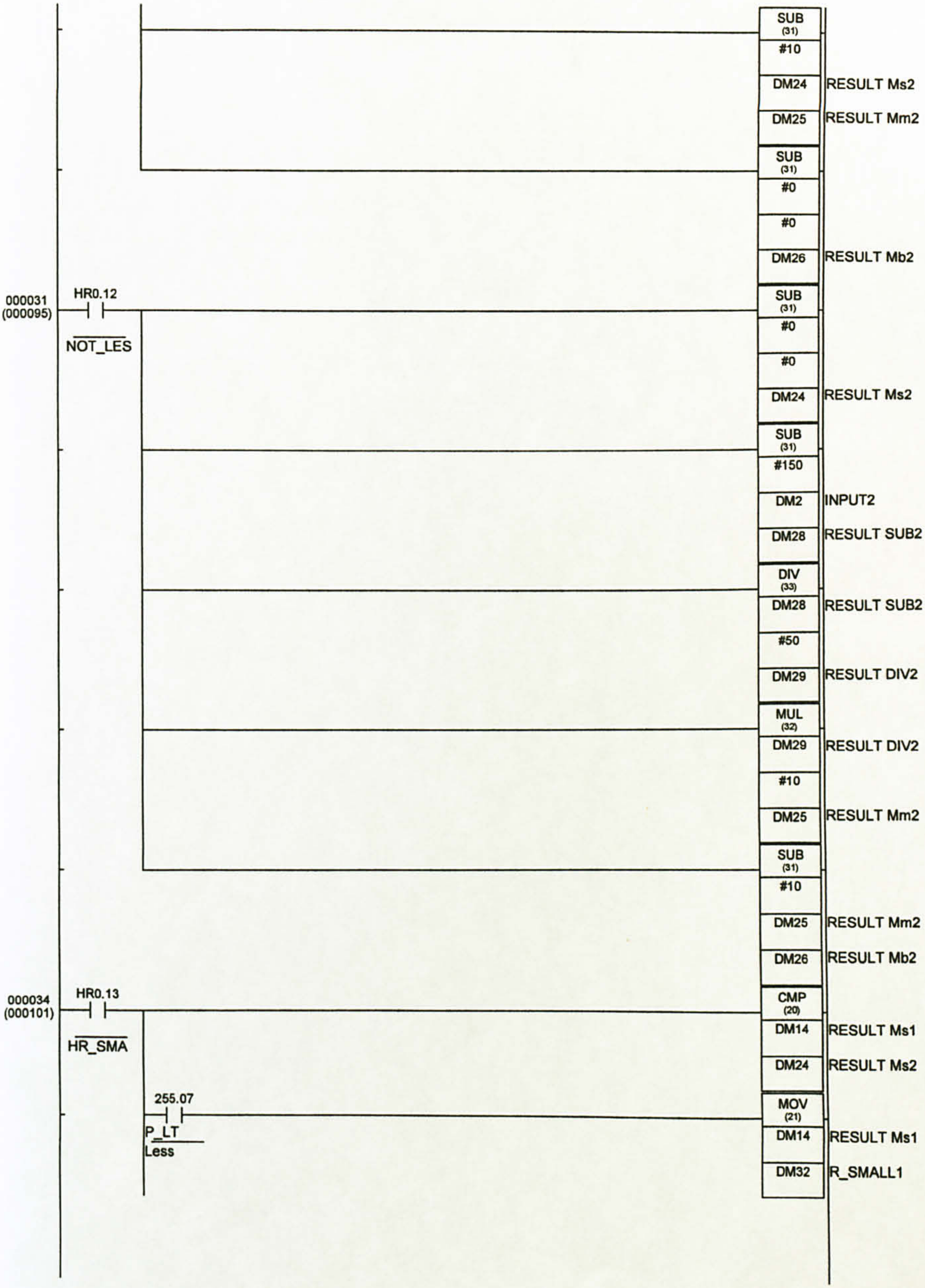

[Program Name : Section1]
[Section Name : Section1]
000000
(000000)
0.03
SET
0.02
RESET
HR0.00
HR0
HR0.00
HR0
a001 a007 a010
a013 a037 a200
a208 a216 a248
a252 a257
HR0.13
HR_SMA
HR_SMALL
a101 a112 a123
HR0.14
HR_MEDI
HR_MEDIUM
a134 a145 a156
HR0.15
HR_BIG
HR_BIG
a167 a178 a189
000001
(000007)
HR0.00
HR0
ADD
(30)
#80
#0
DM1
INPUT1
ADD
(30)
#120
#0
DM2
INPUT2
000003
(000010)
HR0.00
HR0
ADD
(30)
#50
#0
DM3
SET VALUE 50
ADD
(30)
#150
#0
DM4
SET VALUE 150
000005
(000013)
HR0.00
HR0
CMP
(20)
DM1
INPUT1
DM3
SET VALUE 50
255.07
P_LT
Less
HR0.01
Set input
Set input to 50
a021

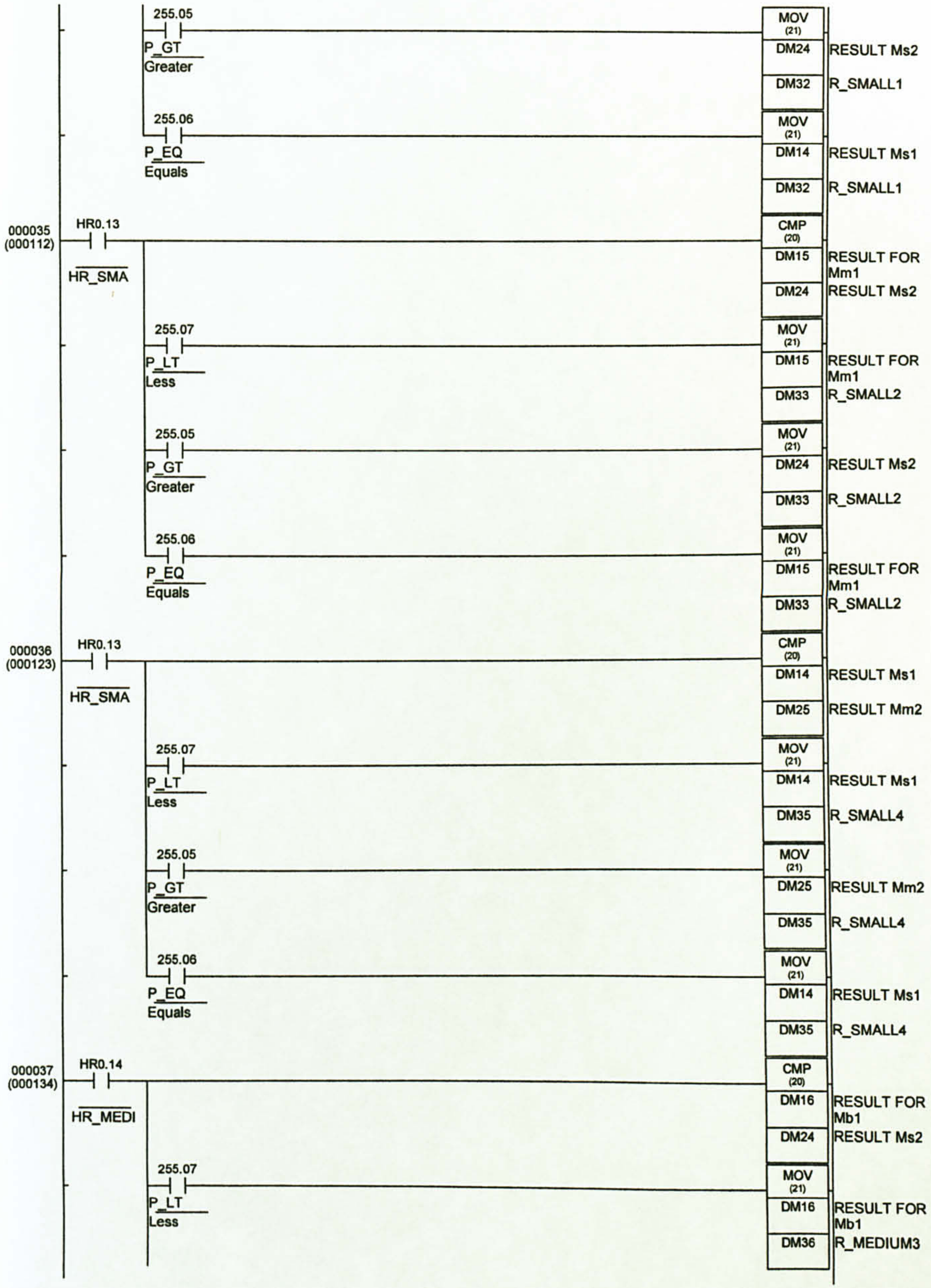

255.07
P_LT
Less
HR0.02
CMP
CMP WITH 150
a025
000008
(000021)
HR0.01
Set input
ADD
(30)
#10
#0
DM14
RESULT Ms1
ADD
(30)
#0
#0
DM15
RESULT FOR
Mm1
ADD
(30)
#0
#0
DM16
RESULT FOR
Mb1
000009
(000025)
HR0.02
CMP
CMP
(20)
DM1
INPUT1
DM4
SET VALUE 150
255.07
P_LT
Less
HR0.03
GO TO
GO TO RANGE
100
a061
255.07
P_LT
Less
HR0.04
INPUT 1
INPUT 1 SET TO
150
a033
000010
(000033)
HR0.04
INPUT 1
ADD
(30)
#10
#0
DM16
RESULT FOR
Mb1
ADD
(30)
#0
#0
DM14
RESULT Ms1
ADD
(30)
#0
#0
DM15
RESULT FOR
Mm1

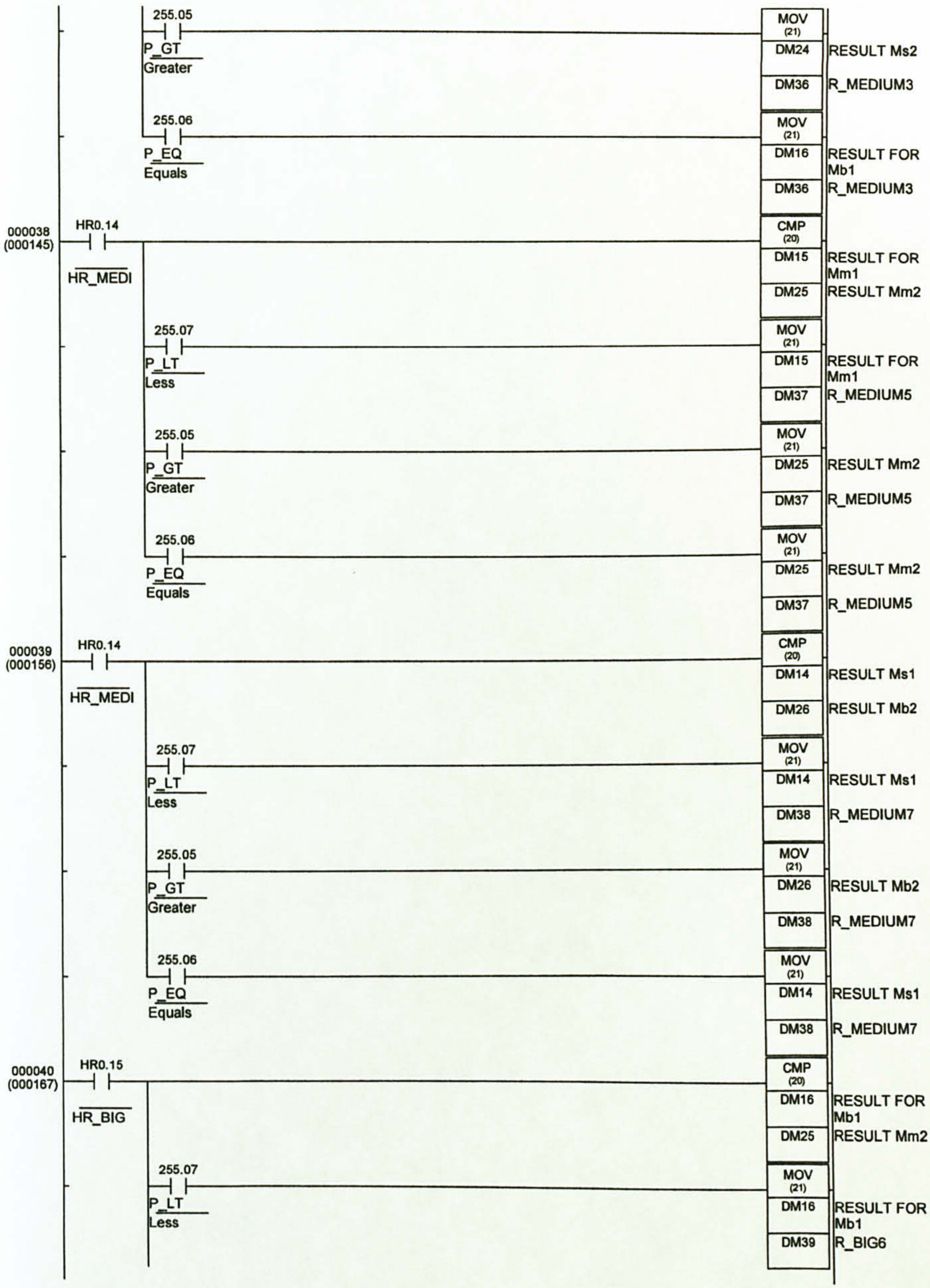

000013
(000037)
HR0.00
HR0
CMP
(20)
DM2
INPUT2
DM3
SET VALUE 50
255.07
P_LT
Less
HR0.05
Set input2 to 50
a045 a224
Set input2
255.07
P_LT
Less
HR0.06
cmp with 150
a049 a232 a240
cmp with
000016
(000045)
HR0.05
Set input2
ADD
(30)
#10
#0
DM24
RESULT Ms2
ADD
(30)
#0
#0
DM25
RESULT Mm2
ADD
(30)
#0
#0
DM26
RESULT Mb2
000017
(000049)
HR0.06
cmp with
CMP
(20)
DM2
INPUT2
DM4
SET VALUE 150
255.07
P_LT
Less
HR0.07
Go to Range 100
a081
Go to
255.07
P_LT
Less
HR0.08
Input 2 set to 150
a057
Input 2
000018
(000057)
HR0.08
Input 2
ADD
(30)
#10
#0
DM26
RESULT Mb2
ADD
(30)
#0
#0
DM24
RESULT Ms2


ADD (30)
#0
#0
DM25
RESULT Mm2
000022 (000061)
HR0.03
GO TO
CMP (20)
DM1
INPUT1
#100
255.07
P_LT Less
HR0.09
LESS THAN
a069
LESS
255.07
P_LT Less
HR0.10
NOT_LESS THAN
a075
NOT_LES
000023 (000069)
HR0.09
LESS
SUB (31)
#100
DM1
INPUT1
DM12
RESULT FOR SUB
DIV (33)
DM12
RESULT FOR SUB
#50
DM13
RESULT FOR DIV
MUL (32)
DM13
RESULT FOR DIV
#10
DM14
RESULT Ms1
SUB (31)
#10
DM14
RESULT Ms1
DM15
RESULT FOR Mm1
SUB (31)
#0
#0
DM16
RESULT FOR Mb1
000024 (000075)
HR0.10
NOT_LES
SUB (31)
#0
#0
DM14
RESULT Ms1

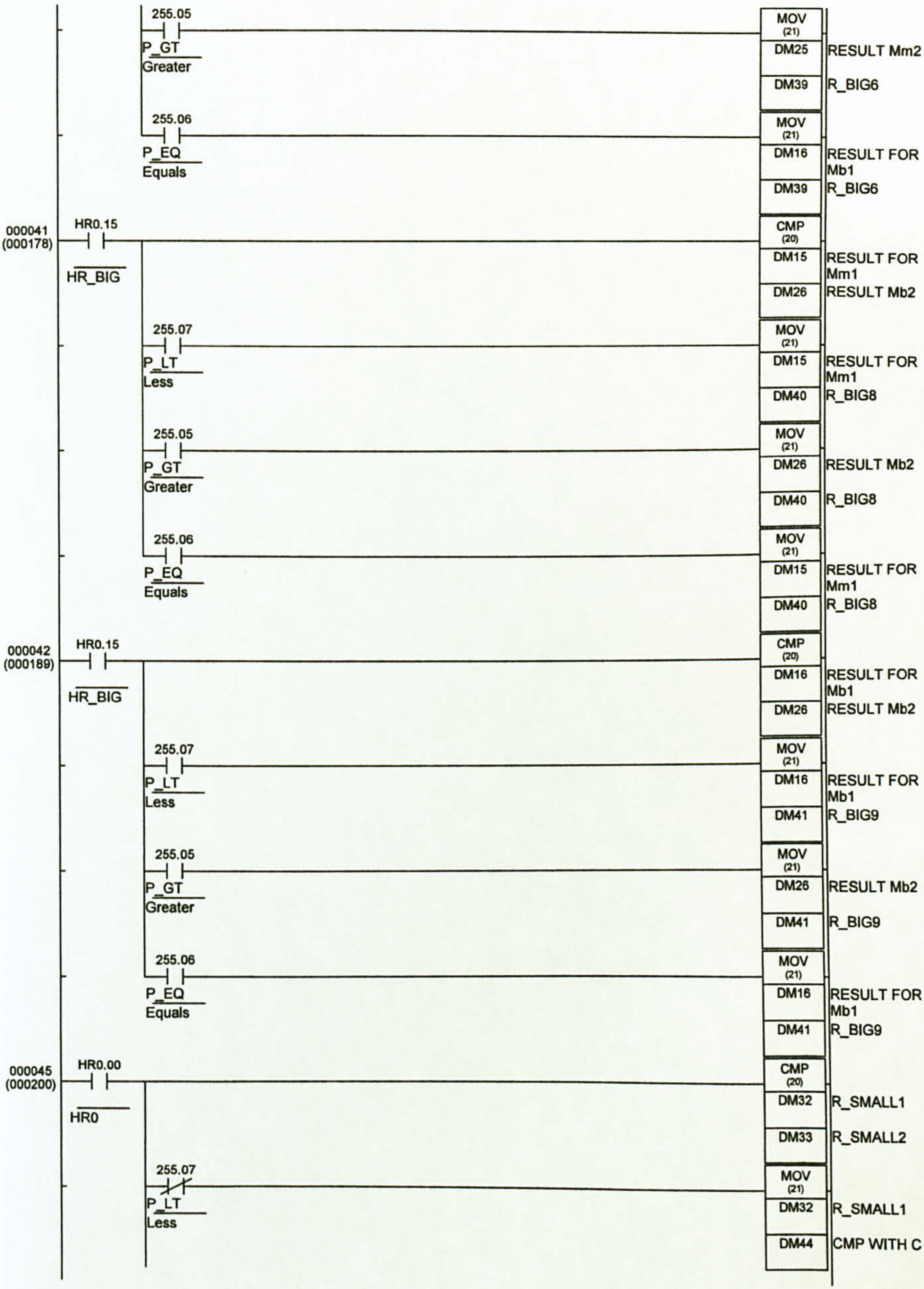

SUB (31)
#150
DM1
INPUT1
DM18
rESULT SUB2
DIV (33)
DM18
rESULT SUB2
#50
DM19
RESULT DIV2
MUL (32)
DM19
RESULT DIV2
#10
DM15
RESULT FOR Mm1
SUB (31)
#10
DM15
RESULT FOR Mm1
DM16
RESULT FOR Mb1
000027 (000081)
HR0.07
Go to
CMP (20)
DM2
INPUT2
#100
255.07
P_LT
Less
HR0.11
LESS
LESS THAN (INPUT2)
a089
255.07
P_LT
Less
HR0.12
NOT_LES
NOT_LESS THAN (INPUT2)
a095
000030 (000089)
HR0.11
LESS
SUB (31)
#100
DM2
INPUT2
DM22
RESULT FOR SUB1
DIV (33)
DM22
RESULT FOR SUB1
#50
DM23
RESULT FOR DIV1
MUL (32)
DM23
RESULT FOR DIV1
#10
DM24
RESULT Ms2

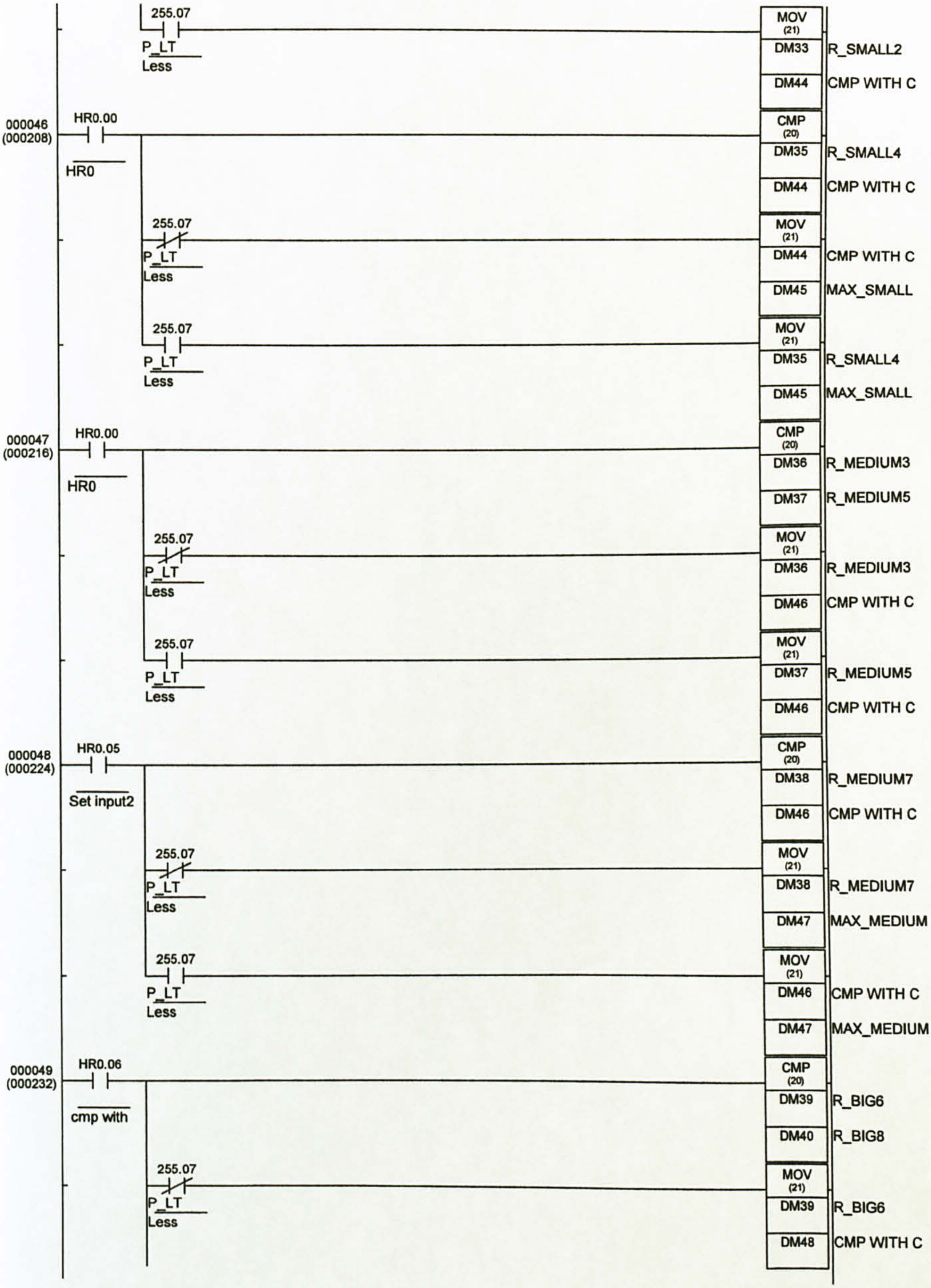

SUB
(31)
#10
DM24
RESULT Ms2
DM25
RESULT Mm2
SUB
(31)
#0
#0
DM26
RESULT Mb2
000031
(000095)
HR0.12
NOT_LES
SUB
(31)
#0
#0
DM24
RESULT Ms2
SUB
(31)
#150
DM2
INPUT2
DM28
RESULT SUB2
DIV
(33)
DM28
RESULT SUB2
#50
DM29
RESULT DIV2
MUL
(32)
DM29
RESULT DIV2
#10
DM25
RESULT Mm2
SUB
(31)
#10
DM25
RESULT Mm2
DM26
RESULT Mb2
000034
(000101)
HR0.13
HR_SMA
CMP
(20)
DM14
RESULT Ms1
DM24
RESULT Ms2
255.07
P_LT
Less
MOV
(21)
DM14
RESULT Ms1
DM32
R_SMALL1


255.05
P_GT
Greater
MOV
(21)
DM24
RESULT Ms2
DM32
R_SMALL1
255.06
P_EQ
Equals
MOV
(21)
DM14
RESULT Ms1
DM32
R_SMALL1
000035
(000112)
HR0.13
HR_SMA
CMP
(20)
DM15
RESULT FOR Mm1
DM24
RESULT Ms2
255.07
P_LT
Less
MOV
(21)
DM15
RESULT FOR Mm1
DM33
R_SMALL2
255.05
P_GT
Greater
MOV
(21)
DM24
RESULT Ms2
DM33
R_SMALL2
255.06
P_EQ
Equals
MOV
(21)
DM15
RESULT FOR Mm1
DM33
R_SMALL2
000036
(000123)
HR0.13
HR_SMA
CMP
(20)
DM14
RESULT Ms1
DM25
RESULT Mm2
255.07
P_LT
Less
MOV
(21)
DM14
RESULT Ms1
DM35
R_SMALL4
255.05
P_GT
Greater
MOV
(21)
DM25
RESULT Mm2
DM35
R_SMALL4
255.06
P_EQ
Equals
MOV
(21)
DM14
RESULT Ms1
DM35
R_SMALL4
000037
(000134)
HR0.14
HR_MEDI
CMP
(20)
DM16
RESULT FOR Mb1
DM24
RESULT Ms2
255.07
P_LT
Less
MOV
(21)
DM16
RESULT FOR Mb1
DM36
R_MEDIUM3

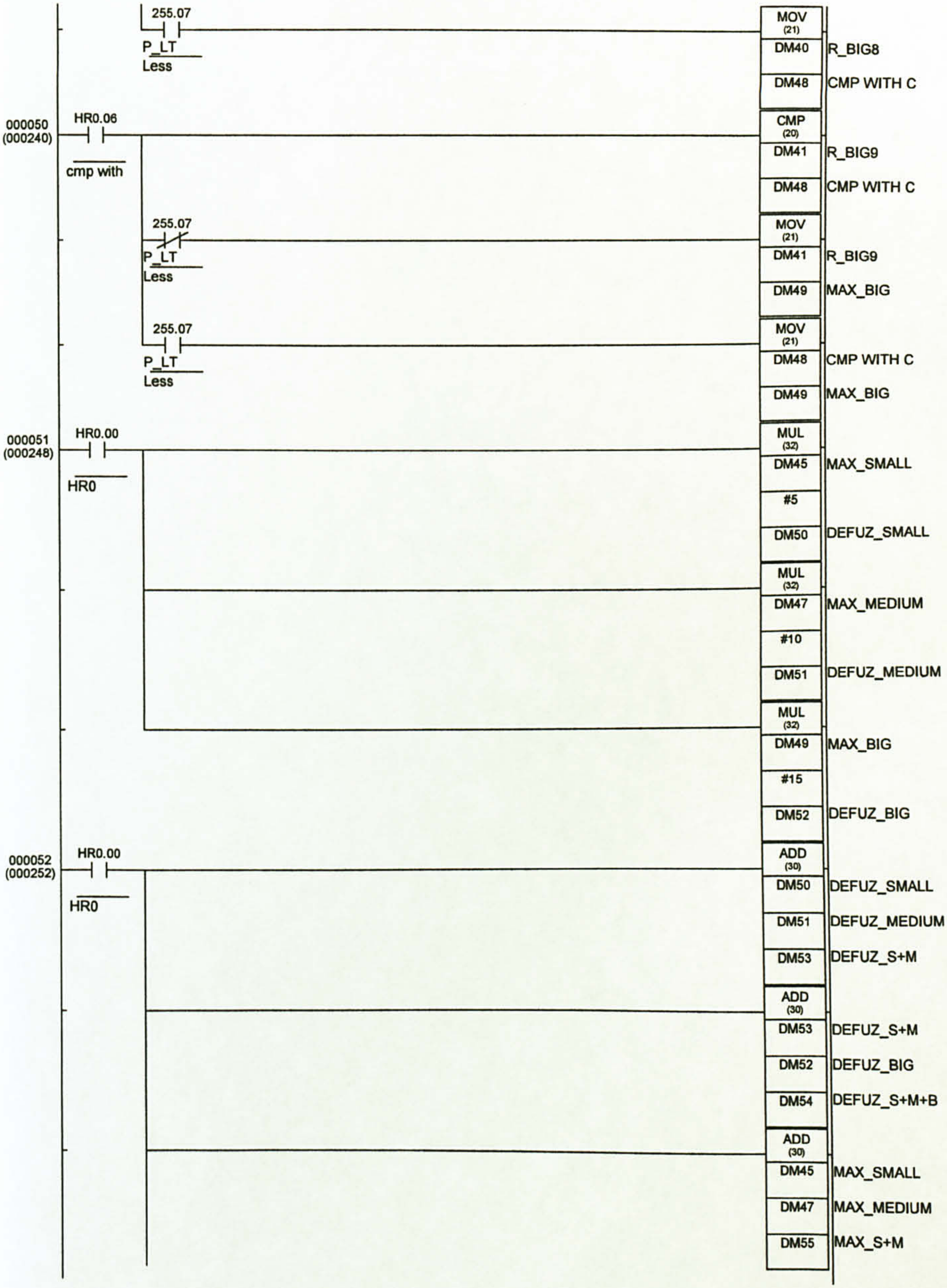


255.05
P_GT
Greater
MOV
(21)
DM24
RESULT Ms2
DM36
R_MEDIUM3
255.06
P_EQ
Equals
MOV
(21)
DM16
RESULT FOR Mb1
DM36
R_MEDIUM3
000038
(000145)
HR0.14
HR_MEDI
CMP
(20)
DM15
RESULT FOR Mm1
DM25
RESULT Mm2
255.07
P_LT
Less
MOV
(21)
DM15
RESULT FOR Mm1
DM37
R_MEDIUM5
255.05
P_GT
Greater
MOV
(21)
DM25
RESULT Mm2
DM37
R_MEDIUM5
255.06
P_EQ
Equals
MOV
(21)
DM25
RESULT Mm2
DM37
R_MEDIUM5
000039
(000156)
HR0.14
HR_MEDI
CMP
(20)
DM14
RESULT Ms1
DM26
RESULT Mb2
255.07
P_LT
Less
MOV
(21)
DM14
RESULT Ms1
DM38
R_MEDIUM7
255.05
P_GT
Greater
MOV
(21)
DM26
RESULT Mb2
DM38
R_MEDIUM7
255.06
P_EQ
Equals
MOV
(21)
DM14
RESULT Ms1
DM38
R_MEDIUM7
000040
(000167)
HR0.15
HR_BIG
CMP
(20)
DM16
RESULT FOR Mb1
DM25
RESULT Mm2
255.07
P_LT
Less
MOV
(21)
DM16
RESULT FOR Mb1
DM39
R_BIG6

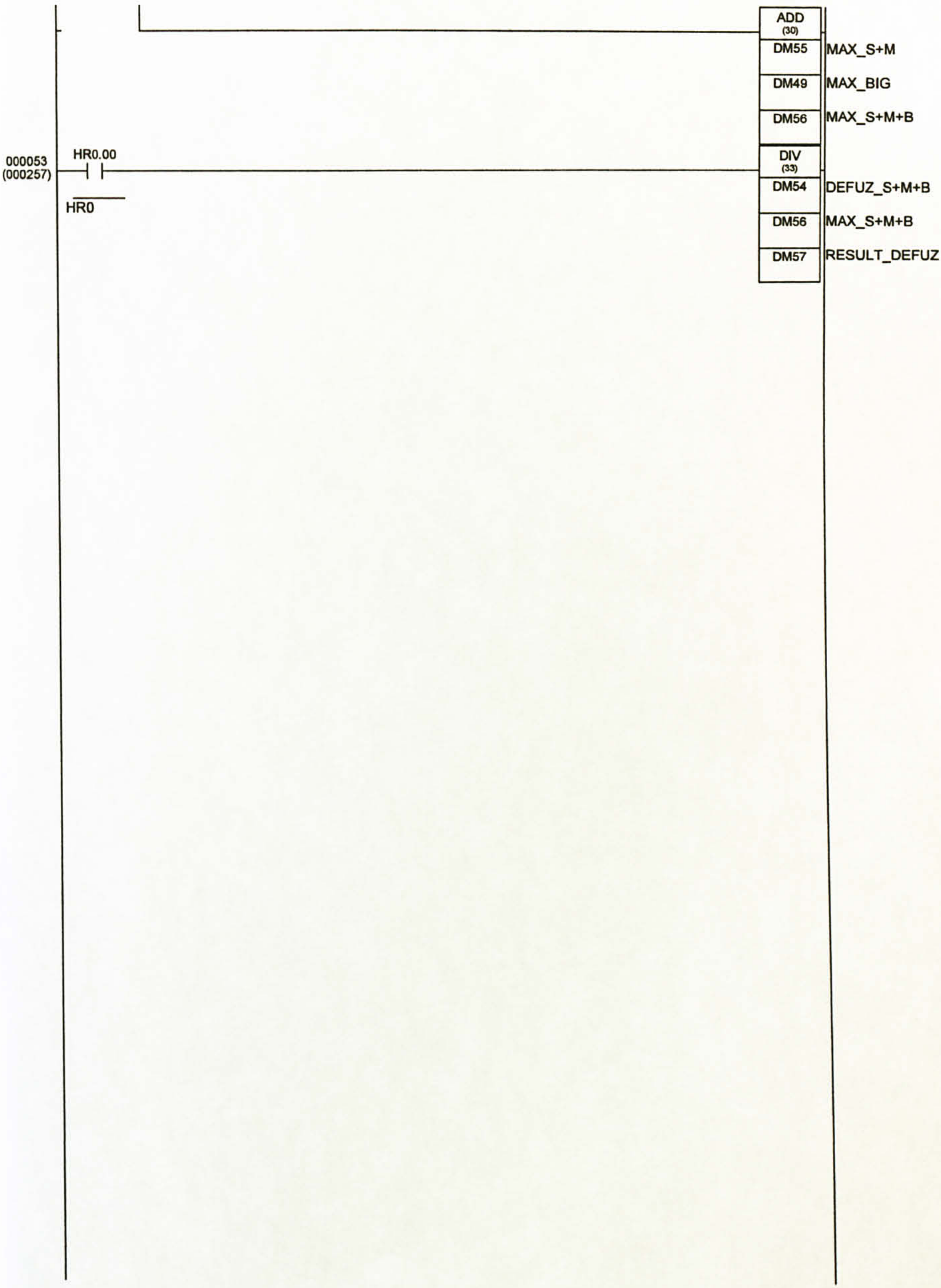

255.05
P_GT
Greater
MOV
(21)
DM25
RESULT Mm2
DM39
R_BIG6
255.06
P_EQ
Equals
MOV
(21)
DM16
RESULT FOR Mb1
DM39
R_BIG6
000041
(000178)
HR0.15
HR_BIG
CMP
(20)
DM15
RESULT FOR Mm1
DM26
RESULT Mb2
255.07
P_LT
Less
MOV
(21)
DM15
RESULT FOR Mm1
DM40
R_BIG8
255.05
P_GT
Greater
MOV
(21)
DM26
RESULT Mb2
DM40
R_BIG8
255.06
P_EQ
Equals
MOV
(21)
DM15
RESULT FOR Mm1
DM40
R_BIG8
000042
(000189)
HR0.15
HR_BIG
CMP
(20)
DM16
RESULT FOR Mb1
DM26
RESULT Mb2
255.07
P_LT
Less
MOV
(21)
DM16
RESULT FOR Mb1
DM41
R_BIG9
255.05
P_GT
Greater
MOV
(21)
DM26
RESULT Mb2
DM41
R_BIG9
255.06
P_EQ
Equals
MOV
(21)
DM16
RESULT FOR Mb1
DM41
R_BIG9
000045
(000200)
HR0.00
HR0
CMP
(20)
DM32
R_SMALL1
DM33
R_SMALL2
255.07
P_LT
Less
MOV
(21)
DM32
R_SMALL1
DM44
CMP WITH C



255.07
P_LT
Less
MOV
(21)
DM33
R_SMALL2
DM44
CMP WITH C
000046
(000208)
HR0.00
HR0
CMP
(20)
DM35
R_SMALL4
DM44
CMP WITH C
255.07
P_LT
Less
MOV
(21)
DM44
CMP WITH C
DM45
MAX_SMALL
255.07
P_LT
Less
MOV
(21)
DM35
R_SMALL4
DM45
MAX_SMALL
000047
(000216)
HR0.00
HR0
CMP
(20)
DM36
R_MEDIUM3
DM37
R_MEDIUM5
255.07
P_LT
Less
MOV
(21)
DM36
R_MEDIUM3
DM46
CMP WITH C
255.07
P_LT
Less
MOV
(21)
DM37
R_MEDIUM5
DM46
CMP WITH C
000048
(000224)
HR0.05
Set input2
CMP
(20)
DM38
R_MEDIUM7
DM46
CMP WITH C
255.07
P_LT
Less
MOV
(21)
DM38
R_MEDIUM7
DM47
MAX_MEDIUM
255.07
P_LT
Less
MOV
(21)
DM46
CMP WITH C
DM47
MAX_MEDIUM
000049
(000232)
HR0.06
cmp with
CMP
(20)
DM39
R_BIG6
DM40
R_BIG8
255.07
P_LT
Less
MOV
(21)
DM39
R_BIG6
DM48
CMP WITH C



255.07
P_LT
Less
MOV
(21)
DM40
R_BIG8
DM48
CMP WITH C
000050
(000240)
HR0.06
cmp with
CMP
(20)
DM41
R_BIG9
DM48
CMP WITH C
255.07
P_LT
Less
MOV
(21)
DM41
R_BIG9
DM49
MAX_BIG
255.07
P_LT
Less
MOV
(21)
DM48
CMP WITH C
DM49
MAX_BIG
000051
(000248)
HR0.00
HR0
MUL
(32)
DM45
MAX_SMALL
#5
DM50
DEFUZ_SMALL
MUL
(32)
DM47
MAX_MEDIUM
#10
DM51
DEFUZ_MEDIUM
MUL
(32)
DM49
MAX_BIG
#15
DM52
DEFUZ_BIG
000052
(000252)
HR0.00
HR0
ADD
(30)
DM50
DEFUZ_SMALL
DM51
DEFUZ_MEDIUM
DM53
DEFUZ_S+M
ADD
(30)
DM53
DEFUZ_S+M
DM52
DEFUZ_BIG
DM54
DEFUZ_S+M+B
ADD
(30)
DM45
MAX_SMALL
DM47
MAX_MEDIUM
DM55
MAX_S+M


ADD
(30)
DM55
MAX_S+M
DM49
MAX_BIG
DM56
MAX_S+M+B
000053
(000257)
HR0.00
HR0
DIV
(33)
DM54
DEFUZ_S+M+B
DM56
MAX_S+M+B
DM57
RESULT_DEFUZ

APPENDIX D
PLC DATA SHEET

2-4 Analog Setting Board

2-4-1 Model

Name	Model	Specifications
Analog Setting Board	CQM1H-AVB41	Four analog setting screws

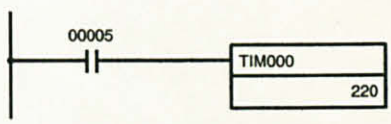
2-4-2 Function

Each of the values set using the four variable resistors located on the front of the Analog Settings Board is stored as a 4-digit BCD between 0000 and 0200 in the analog settings words (IR 220 to IR 223).

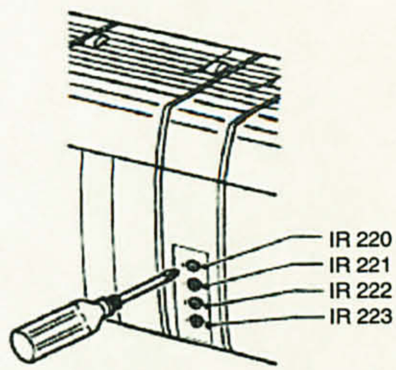
By using the Analog Setting Board, an operator can, for example, set the value of a timer instruction using an analog setting (IR 220 to IR 223), and thereby slightly speed up or slow down the speed or timing of a conveyor belt simply by adjusting a control with a screwdriver, removing the need for a Programming Device.

Using the Analog Timer

The following example shows the 4-digit BCD setting (0000 to 0200) stored in IR 220 to IR 223 being used as a timer setting.



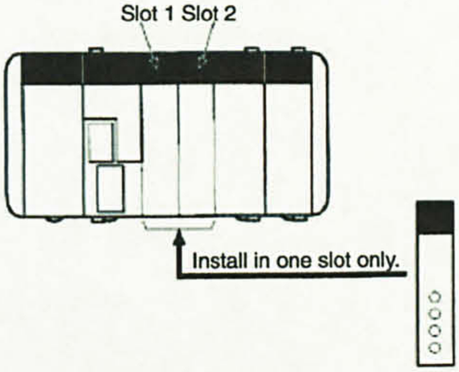
The setting of TIM000 is set externally in IR 220. (Timer is executed using the setting of analog control 0.)



Phillips screwdriver

2-4-3 Applicable Inner Board Slots

The Analog Setting Board can be installed in either slot 1 (left slot) or slot 2 (right slot) of the CQM1H-CPU51/61 CPU Unit. Both slots, however, cannot be used at the same time.

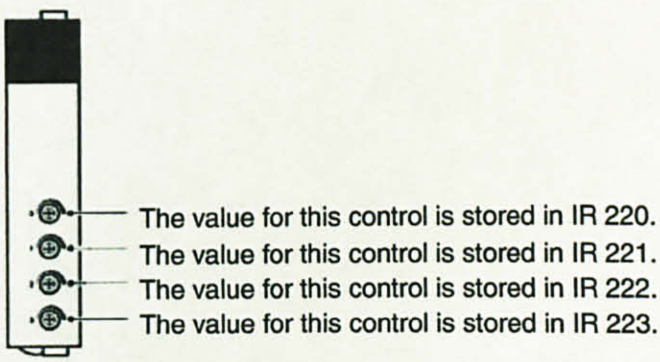


2-4-4 Names and Functions

The four analog controls of the Analog Setting Board are located on the front panel. The front panel does not have any indicators.

The value of the setting increases as the control is rotated clockwise. Use a small Phillips screwdriver for this purpose.

Specifying IR 220 to IR 223 as the set value of a TIM instruction enables the Board to be used as an analog timer. When the timer is started, the analog settings are stored as the timer set value.



⚠ Caution While the power is turned ON, the contents of IR 220 to IR 223 are constantly refreshed with the values of the corresponding controls. Be sure that these words are not written to from the program or a Programming Device.

2-4-5 Specifications

Relevant Bits

The values of the Analog Setting Board analog controls are stored in the following addresses of the Inner Board area regardless of the slot in which the Board is mounted.

Word	Bits	Name	Function
IR 220	00 to 15	Analog control 1	With each cycle, the values of analog controls 0 to 3 are stored as 4-digit BCD values between 0000 and 0200.
IR 221	00 to 15	Analog control 2	
IR 222	00 to 15	Analog control 3	
IR 223	00 to 15	Analog control 4	

Related PC Setup Settings None

2-5 Analog I/O Board

2-5-1 Model

Name	Model	Specifications
Analog I/O Board	CQM1H-MAB42	4 analog inputs (–10 to +10 V; 0 to 5 V; 0 to 20 mA; separate signal range for each point) 2 analog outputs (–10 to +10 V; 0 to 20 mA; separate signal range for each point)

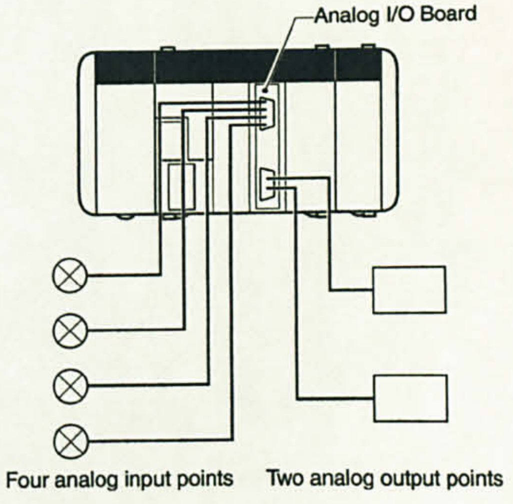
2-5-2 Function

The Analog I/O Board is an Inner Board featuring four analog inputs and two analog outputs.

The signal ranges that can be used for each of the four analog input points are –10 to +10 V, 0 to 5 V, and 0 to 20 mA. A separate range is set for each point. The settings in DM 6611 determine the signal ranges.

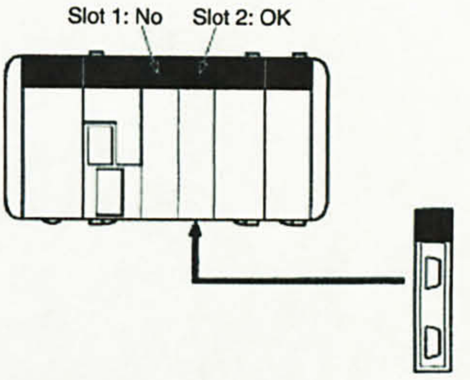
The signal ranges that can be used for each of the two analog output points are –10 to +10 V and 0 to 20 mA. A separate signal range can be selected for each point. The settings in DM 6611 determine the signal range.

2-5-3 System Configuration



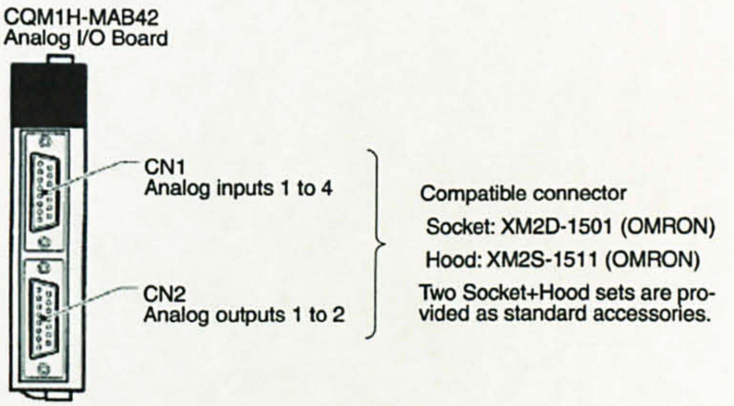
2-5-4 Applicable Inner Board Slot

The Analog I/O Board can only be mounted in slot 2 (right slot) of the CQM1H-CPU51/61 CPU Unit.

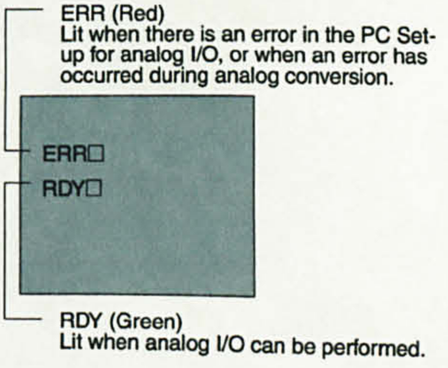


2-5-5 Names and Functions

The Analog I/O Board has a CN1 connector for the four analog inputs and a CN2 connector for 2 analog outputs.



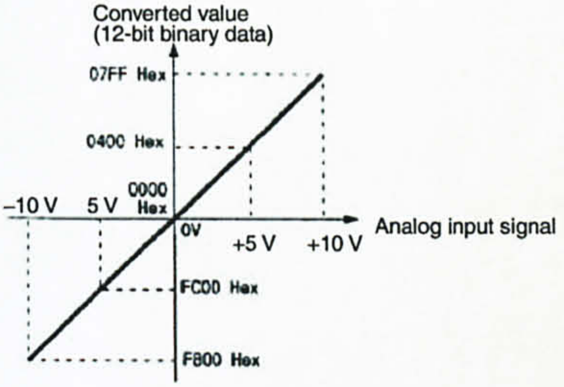
LED Indicators



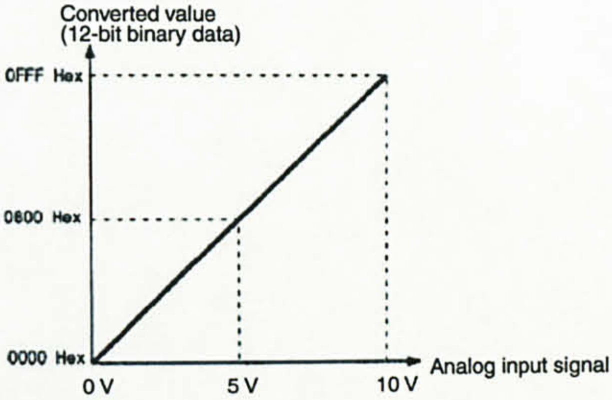
2-5-6 Specifications

Analog Inputs: Input Data and Converted Values

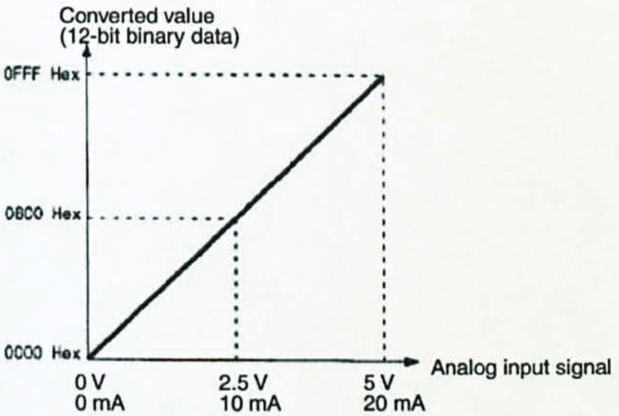
–10 to +10 V



0 to +10 V

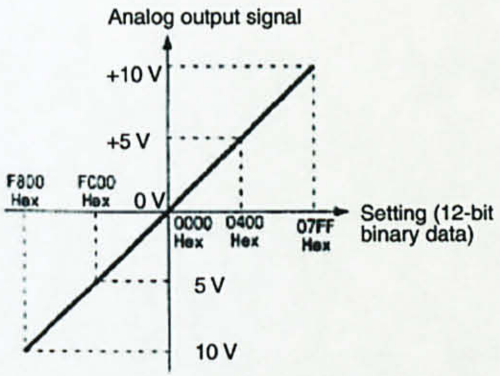


0 to 5 V or 0 to 20 mA

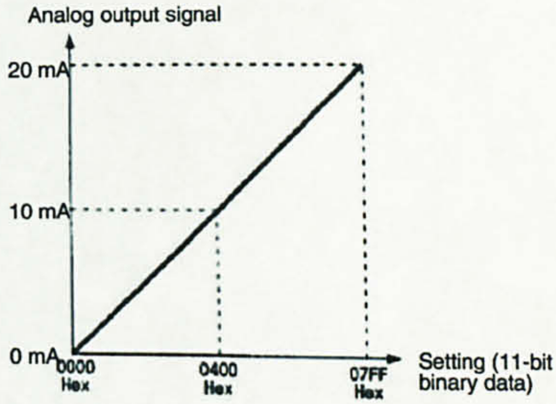


Analog Outputs: Settings and Output Data

–10 to +10 V



0 to 20 mA



Applications Examples

The Board uses no special instructions. MOV(21) is used to read analog input values and set analog output values.

Relevant Bits

Bits Used by Inner Board in Slot 2

Word	Bits	Name	Function
IR 232	00 to 15	Analog input 1 converted value	The converted value from each input from the Analog I/O Board is stored as a 4-digit Hex each cycle. –10 to +10 V: F800 to 07FFF Hex 0 to 10 V: 0000 to 0FFF Hex 0 to 5 V/0 to 20 mA: 0000 to 0FFF Hex
IR 233	00 to 15	Analog input 2 converted value	
IR 234	00 to 15	Analog input 3 converted value	
IR 235	00 to 15	Analog input 4 converted value	
IR 236	00 to 15	Analog output 1 setting	The setting of each output from the Analog I/O Board is stored as a 4-digit Hex. (Read each cycle.) –10 to +10 V: F800 to 07FF Hex 0 to 20 mA: 0000 to 07FF Hex
IR 237	00 to 15	Analog output 2 setting	

SR Area Flags

Word	Bit	Function
SR 254	15	Inner Board Error Flag

AR Area Flags

Word	Bits	Function	
AR 04	08 to 15	Error codes for Inner Board in slot 2	00 Hex: Normal 01 or 02 Hex: Hardware error 03 Hex: PC Setup error 04 Hex: A/D or D/A conversion error

Relevant PC Setup Settings

Word	Bits	Function	
DM 6611	00 to 07	00, 01: Analog input 1 input signal range 02, 03: Analog input 2 input signal range 04, 05: Analog input 3 input signal range 06, 07: Analog input 4 input signal range	00: –10 to +10 V 01: 0 to 10 V 10: 0 to 5 V/0 to 20 mA 11: Not used. (0 to 20 mA are distinguished by the connected terminal.)
	08	Analog input 1 usage selection	Specifies use or non-use of A/D conversion for each port. 0: Use input (conversion) 1: Do not use input (no conversion)
	09	Analog input 2 usage selection	
	10	Analog input 3 usage selection	
	11	Analog input 4 usage selection	
	12 to 15	Not used. (Fixed at 0.)	

Note The level of the analog output signal is determined by the connected terminal, and there is no PC Setup setting. These settings are reflected in status at power ON.

2-5-7 Application Procedure

Determine analog input ranges and number of inputs. Determine analog output ranges and number of outputs.	
↓	
Wire analog inputs and analog outputs.	0 to 5 V and 0 to 20 mA analog inputs are selected by the terminals connected. 0 to 5 V and 0 to 20 mA analog outputs are selected by the terminals connected.
↓	
PC Setup (DM 6611)	Set signal ranges for analog inputs. Set whether or not to use analog inputs.
↓	
Ladder Program	Analog inputs: Read converted values. Analog outputs: Write settings.

2-6 Serial Communications Board

This section provides an introduction to the Serial Communications Board. Detailed information can be found in the *Serial Communications Board Operation Manual* (W365).

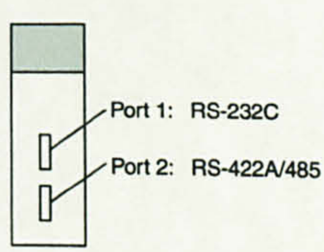
2-6-1 Model Number

Name	Model	Specifications
Serial Communications Board	CQM1H-SCB41	One RS-232 port One RS-422A/485 port

2-6-2 Serial Communications Boards

The Serial Communications Board is an Inner Board for the CQM1H-series PCs. One Board can be installed in Inner Board slot 1 of a CQM1H-series CPU Unit. The Board cannot be installed in slot 2.

The Board provides two serial communications ports for connecting host computers, Programmable Terminals (PTs), general-purpose external devices, and Programming Devices (excluding Programming Consoles). This makes it possible to easily increase the number of serial communications ports for a CQM1H-series PC.



3-1 Memory Area Structure

The following memory areas can be used with the CQM1H.

Data area		Size	Words	Bits	Function
IR area (note 1)	Input area	256 bits	IR 000 to IR 015	IR 00000 to IR 01515	Input bits can be allocated to Input Units or I/O Units. The 16 bits in IR 000 are always allocated to the CPU Unit's built-in inputs.
	Output area	256 bits	IR 100 to IR 115	IR 10000 to IR 11515	Output bits can be allocated to Output Units or I/O Units.
	Work areas	2,528 bits min. (note 2)	IR 016 to IR 089	IR 01600 to IR 08915	Work bits do not have any specific function, and they can be freely used within the program.
			IR 116 to IR 189	IR 11600 to IR 18915	
			IR 216 to IR 219	IR 21600 to IR 21915	
			IR 224 to IR 229	IR 22400 to IR 22915	
Controller Link status areas		96 bits	IR 090 to IR 095	IR 09000 to IR 09615	Used to indicate the Controller Link Data Link status information. (Can be used as work bits when a Controller Link Unit is not mounted.)
		96 bits	IR 190 to IR 195	IR 19000 to IR 19615	Used to indicate the Controller Link error and network participation information. (Can be used as work bits when a Controller Link Unit is not mounted.)
MACRO operand area (note 1)	Input area	64 bits	IR 096 to IR 099	IR 09600 to IR 09915	Used when the MACRO instruction, MCRO(99), is used. (Can be used as work bits when the MACRO instruction is not used.)
	Output area	64 bits	IR 196 to IR 199	IR 19600 to IR 19915	
Inner Board slot 1 area		256 bits	IR 200 to IR 215	IR 20000 to IR 21515	These bits are allocated to the Inner Board mounted in slot 1 of the CQM1H-CPU51/61. (Can be used as work bits when the CQM1H-CPU11/CPU21 is being used or slot 1 is empty.) CQM1H-CTB41 High-speed Counter Board: IR 200 to IR 213 (14 words): Used by the Board IR 214 and IR 215 (2 words): Not used. CQM1H-SCB41 Serial Communications Board: IR 200 to IR 207 (8 words): Used by the Board IR 208 to IR 215 (8 words): Not used.
Analog settings area (note 1)		64 bits	IR 220 to IR 223	IR 22000 to IR 22315	Used to store the analog settings when the CQM1H-AVB41 Analog Setting Board is mounted. (Can be used as work bits when an Analog Setting Board is not mounted.)
High-speed Counter 0 PV (note 1)		32 bits	IR 230 to IR 231	IR 23000 to IR 23115	Used to store the present values of the built-in high-speed counter (high-speed counter 0). (Can be used as work bits when high-speed counter 0 is not being used.)
Inner Board slot 2 area		192 bits	IR 232 to IR 243	IR 23200 to IR 24315	These bits are allocated to the Inner Board mounted in slot 2 of the CQM1H-CPU51/61. (Can be used as work bits when the CQM1H-CPU11/21 is being used or slot 2 is empty.) CQM1H-CTB41 High-speed Counter Board: IR 232 to IR 243 (12 words): Used by the Board CQM1H-PLB21 Pulse I/O Board: IR 232 to IR 239 (8 words): Used by the Board IR 240 to IR 243 (4 words): Not used. CQM1H-ABB21 Absolute Encoder Interface Board: IR 232 to IR 239 (8 words): Used by the Board IR 240 to IR 243 (4 words): Not used. CQM1H-MAB42 Analog I/O Board: IR 232 to IR 239 (8 words): Used by the Board IR 240 to IR 243 (4 words): Not used.

Data area		Size	Words	Bits	Function
SR area		184 bits	SR 244 to SR 255	SR 24400 to SR 25507	These bits serve specific functions such as flags and control bits.
HR area		1,600 bits	HR 00 to HR 99	HR 0000 to HR 9915	These bits store data and retain their ON/OFF status when power is turned OFF.
AR area		448 bits	AR 00 to AR 27	AR 0000 to AR 2715	These bits serve specific functions such as flags and control bits.
TR area		8 bits	---	TR 0 to TR 7	These bits are used to temporarily store ON/OFF status at program branches.
LR area (note 1)		1,024 bits	LR 00 to LR 63	LR 0000 to LR 6315	Used for 1:1 Data Link through the RS-232 port or through a Controller Link Unit.
Timer/Counter area (note 3)		512 bits	TIM/CNT 000 to TIM/CNT 511 (timer/counter numbers)		The same numbers are used for both timers and counters. When TIMH(15) is being used, timer numbers 000 to 015 can be interrupt-refreshed to ensure proper timing during long cycles.
DM area	Read/write	3,072 words	DM 0000 to DM 3071	---	DM area data can be accessed in word units only. Word values are retained when power is turned OFF.
		3,072 words	DM 3072 to DM 6143	---	Available in CQM1H-CPU51/61 CPU Units only.
	Read-only (note 4)	425 words	DM 6144 to DM 6568	---	Cannot be overwritten from program (only a Programming Device). DM 6400 to DM 6409 (10 words): Controller Link DM parameter area DM 6450 to DM 6499 (50 words): Routing table area DM 6550 to DM 6559 (10 words): Serial Communications Board settings
	Error log area (note 4)	31 words	DM 6569 to DM 6599	---	Used to store the time of occurrence and error code of errors that occur.
	PC Setup (note 4)	56 words	DM 6600 to DM 6655	---	Used to store various parameters that control PC operation.
EM area		6,144 words	EM 0000 to EM 6143	---	EM area data can be accessed in word units only. Word values are retained when power is turned OFF. Available in the CQM1H-CPU61 CPU Unit only.

Note

1. IR and LR bits that are not used for their allocated functions can be used as work bits.
2. A minimum 2,528 bits are available as work bits. Other bits can be used as work bits when they are not used for their allocated functions, so the total number of available work bits depends on the configuration of the PC.
3. When accessing a PV, TIM/CNT numbers are used as word data; when accessing Completion Flags, they are used as bit data.
4. Data in DM 6144 to DM 6655 cannot be overwritten from the program.

Name	Range
Error log area	DM 6569 to DM 6599
PC Setup (see note 2)	DM 6600 to DM 6655

Note
1. The read-only area ranges from DM 6144 to DM 6568.
2. The read-only area, PC Setup, program, and expansion instruction assignments can be transferred to and from the Memory Cassette as a single block of data. See *3-11 Using Memory Cassettes* for details.

Read/Write DM Area

The read/write area has no particular functions assigned to it and can be used freely. It can be read and written from the program or Programming Devices. The size of the read/write area depends upon the model of CPU Unit, as shown in the following table.

CPU Unit	Range	Access from instructions		Access from Programming Devices	
		Read	Write	Read	Write
CQM1H-CPU11	DM 0000 to DM 3071	YES	YES	YES	YES
CQM1H-CPU21					
CQM1H-CPU51	DM 0000 to DM 6143				
CQM1H-CPU61					

Read-only Area (DM 6144 to DM 6568)

DM addresses from DM 6144 to DM 6568 make up the read-only area. Data in the read-only area can be read from instructions (not overwritten) and it can be read and overwritten from Programming Devices. Use the read-only area to store data that you don't want to be changed from the program.

To prevent data from being overwritten by Programming Devices, turn ON pin 1 on the DIP switch on the front of the CPU Unit.

When a Controller Link Unit or Serial Communications Board is being used, part of the read-only area is used for the Controller Link parameters/routing table or Serial Communications Board settings, as shown in the following table.

Name	Range	Access from instructions		Access from Programming Devices	
		Read	Write	Read	Write
Controller Link DM parameters area	DM 6400 to DM 6409	YES	No	YES	YES (See note.)
Routing table area	DM 6450 to DM 6499				
Serial Communications Board settings	DM 6550 to DM 6559				

Note Data cannot be overwritten from Programming Devices when pin 1 on the DIP switch on the front of the CPU Unit is ON.

Error Log Area (DM 6569 to DM 6599)

The CPU Unit automatically records the error code and date/time of up to 10 errors (fatal and non-fatal) in the error log area.

Access from Instructions		Access from Programming Devices	
Read	Write	Read	Write
YES	No	YES	No

PC Setup (DM 6600 to DM 6655)

The PC Setup contains all of the PC Setup settings except for the Serial COmmunications Board settings (stored in DM 6550 to DM 6559). Make the PC Setup settings from a Programming Device.

Access from Instructions		Access from Programming Devices	
Read	Write	Read	Write
YES	No	YES	YES

3-10 EM Area

The EM area can be used in CQM1H-CPU61 CPU Units only. EM data is accessed in word units. Since only one bank of EM is available, bank specification is not necessary.

EM area addresses range from EM 0000 to EM 6143. The area has no particular functions assigned to it and can be used freely. It can be read and written from the program or Programming Devices.

3-11 Using Memory Cassettes

This section provides general information on Memory Cassette specifications and explains how to read, write, and compare information in a Memory Cassette. Refer to the *CQM1H Operation Manual* for details on installing the Memory Cassette, write-protecting flash-memory or EEPROM Memory Cassettes, replacing EPROM chips, and changing the EPROM version switch settings.

An optional Memory Cassette can be used to record the program, read-only DM (DM 6144 to DM 6568), PC Setup (DM 6600 to DM 6655), and expansion instruction assignments. Recording this data on a Memory Cassette prevents the program and vital settings from being changed accidentally. In addition, the settings and the program required for different control processes can be easily changed by simply replacing the Memory Cassette.

The program can be written to the CPU Unit's internal RAM to operate the CQM1H without a Memory Cassette, but the CQM1H can operate even if the CPU Unit's battery fails when a Memory Cassette is used and it's contents are transferred at startup.

Clock Function

The CQM1H PCs can be equipped with a clock by installing a Memory Cassette with a clock. There is an "R" at the end of the model number of Memory Cassettes with a built-in clock. See *3-6-4 Using the Clock* for more details.

Compatibility Between Different CPU Units

Data written to a Memory Cassette by a CQM1H CPU Unit cannot be read by a CQM1 CPU Unit, but data written by a CQM1 CPU Unit can be read by a CQM1H CPU Unit.

Data written to a Memory Cassette by a CQM1H-CPU61 can be read by CQM1H-CPU51, CQM1H-CPU21, and CQM1H-CPU11 CPU Units, but the program will not operate properly if EM area addresses have been used.

3-11-1 Memory Cassettes and Contents

Available Memory Cassettes

The following Memory Cassettes are available.

Memory	Model	Specifications
EEPROM (see note 2)	CQM1-ME04K	4 Kwords without clock
	CQM1-ME04R	4 Kwords with clock
	CQM1-ME08K	8 Kwords without clock
	CQM1-ME08R	8 Kwords with clock

Memory	Model	Specifications
Flash (see notes 1 and 2)	CQM1H-ME16K	16 Kwords without clock
	CQM1H-ME16R	16 Kwords with clock
EPROM (see note 2)	CQM1-MP08K	8 Kwords, 16 Kwords, or 32 Kwords without clock
	CQM1-MP08R	8 Kwords, 16 Kwords, or 32 Kwords with clock

Note

1. Data can be read and written for a EEPROM Memory Cassette with a Programming Device.
2. Data can be read from a EPROM Memory Cassette with a Programming Device, but must be written with a PROM Writer. An EPROM chip with 8 Kwords, 16 Kwords, or 32 Kwords can be installed in the Memory Cassette.
3. The CQM1H-ME16K and CQM1H-ME16R cannot be used in CQM1 PCs.

The following EPROM chips (sold separately) are required for EPROM Memory Cassettes.

Model	ROM version	Capacity	Access speed
ROM-ID-B	27128 or equivalent	8 Kwords	150 ns
ROM-JD-B	27256 or equivalent	16 Kwords	150 ns
ROM-KD-B	27512 or equivalent	32 Kwords	150 ns

Refer to the *CQM1H Operation Manual* for details on replacing EPROM chips and changing the Memory Cassette's EPROM version switch settings.

Contents

The data stored in a Memory Cassette is mainly the CPU Unit's read-only DM, PC Setup, and program, as shown in the following table. All of this data is handled as a single unit; the 4 areas cannot be read, written, or compared individually.

Information		Contents
DM area	Read-only area	Read-only DM cannot be written from the program. The range is DM 6144 to DM 6568. These words can be used freely.
	PC Setup	The PC Setup sets the operating parameters of the CQM1H and it stored in DM 6600 to DM 6655.
	Expansion instruction assignments	These settings indicate which expansion instructions have been assigned function codes.
User program		The entire user program

3-11-2 Memory Cassette Capacity and Program Size

The following table shows the largest program that can be stored in each size Memory Cassette.

Memory Cassette size	Max. program size
4 Kwords	3.2 Kwords
8 Kwords	7.2 Kwords
16 Kwords	15.2 Kwords

A non-fatal error will occur and the transfer will not be executed if an attempt is made to store a program that is too large for the Memory Cassette or read a program that is too large for the CPU Unit. Two examples are shown below.

1,2,3...

1. When a 4-Kword EEPROM Memory Cassette is installed in a CPU Unit with a 7.2-Kword UM (user program) area, programs up to 3.2 Kwords long can be written to the Memory Cassette. A non-fatal error will occur if an at-