



เครื่อง โปรแกรมควบคุมได้  
PROGRAMMABLE LOGIC CONTROLLER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้เผยแพร่โดยไม่ขออนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## เครื่องโปรแกรมควบคุมฯ

นาย ศีรวิทย์	จาบาสวา	33.161105	2I
นาย ปิยพล	ฉิมทับ	33.161111	2I
นาย มานิตษ์	โกลดศิริ	33.161119	2I

ผศ. ศิพจน์ เลาสงคราม

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## PROGRAMMABLE LOGIC CONTROLLER



Mr. NATTAVUT

CHAMPASA

Mr. PIYAPON

CHINTUP

Mr. MANIT

LOHITSIRI

Mr. PIPAT

LAOHASONGKRAM

ADVISOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
<b>บทที่ 1</b> บทนำ	1
<b>บทที่ 2</b> ทฤษฎีพื้นฐานและหลักการ PLC	2
2.1    โครงสร้างของ PLC	2
2.2    หน่วยประมวลผลกลาง	3
2.3    INPUT-OUTPUT UNIT	5
2.4    INPUT-OUTPUT PROCESSING	7
2.5    หน่วยอื่นโปรแกรม	11
2.6    การใช้งานจอภาพ LCD	11
2.7    หน่วยจ่ายพลังงาน	14
<b>บทที่ 3</b> การออกแบบสร้าง PLC	15
3.1    หน่วยประมวลผลกลาง	15
3.2    INPUT-OUTPUT UNIT	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	3.3	หน่วยป้อนโปรแกรม	48
	3.4	หน่วยจ่ายพลังงาน	52
<b>บทที่</b>	<b>4</b>	<b>การออกแบบโปรแกรมบริหารระบบ PLC</b>	<b>55</b>
	4.1	โปรแกรมบริหารระบบทำหน้าที่อะไร	55
	4.2	การจัดหน่วยความจำของ PLC	57
	4.3	การจัดการเกี่ยวกับการป้อนโปรแกรมของผู้ใช้	58
<b>บทที่</b>	<b>5</b>	<b>ภาษาคำสั่งของ PLC</b>	<b>62</b>
	5.1	ภาษาแลดเดอร์โคธแกรม	62
	5.2	ความสัมพันธ์ระหว่างแลดเดอร์โคธแกรมและฟังก์ชันต่างๆ	62
	5.3	ภาษาคำสั่งบูลีน	68
<b>บทที่</b>	<b>6</b>	<b>ชุดคำสั่งใช้งาน</b>	<b>69</b>
	6.1	จำนวน INPUT-OUTPUT, COUNTER, TIMER	69
	6.2	การจำแนกคีย์บอร์ด	71
	6.3	คำสั่งเบื้องต้น	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4	ฟังก์ชันการตั้งเวลา	81
6.5	ฟังก์ชันการนับ	83
6.6	REGISTER	85
<b>บทที่ 7</b>	<b>การใช้ PLC และตัวอย่างการใช้งาน</b>	<b>87</b>
7.1	การใช้ PLC	87
7.2	การใช้โหมดเอ็ดิเตอร์	87
<b>บทสรุป</b>		<b>98</b>
<b>เอกสารอ้างอิง</b>		<b>116</b>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSTRACT

Nowadays, "Sequence Controller"; one type of control device; is very advanced if we compare it with its counterparts of the past. The characteristics of present-day sequence controller stress the importance of using this device easily, and enhance the efficiency of its performance at the same time, i.e. it can do more complicated jobs and can be programmed. This thesis present a design of control device used in the sequential control system by means of programming. The above-mentioned sequence controller is called "PLC" (Programmable logic controller) Which is much used in developed countries; but in Thailand, PLC is just making its debut. This designed device is developed from MCS 51 number 8032 an 8-bit microprocessor, which functions both as a central processing unit and a programming unit. The advantage of the designed controller lies in its ability to interface with a large number of I/O devices, change the control sequence easily, process time swiftly.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทคัดย่อ

ปัจจุบันนี้เครื่องควบคุมแบบซีควเอนซ์ (Sequence Controller) ได้เจริญก้าวหน้าขึ้นมามากเมื่อเทียบกับในอดีตที่ผ่านมา แต่มีความสามารถในการทำงานสูงขึ้น คือทำงานที่ซับซ้อนได้ และสามารถทำการโปรแกรมได้ ปัญญาในขั้นนี้ขอเสนอการออกแบบพัฒนาเครื่องควบคุมแบบซีควเอนซ์ ที่สามารถควบคุมกระบวนการแบบลำดับอย่างต่อเนื่อง โดยวิธีการโปรแกรม เครื่องควบคุมนี้เรียกว่า "พีแอลซี" (PLC: Programmable Logic Controller) ซึ่งเป็นที่แพร่หลายมากในวงการอุตสาหกรรมต่างประเทศ แต่ในประเทศไทยเพิ่งจะเริ่มมีการใช้ PLC เครื่องควบคุมที่ออกแบบนี้ใช้ Microprocessor MCS 51 เบอร์ 8032 ขนาด 8 bit เป็นหน่วยประมวลผลกลาง (CPU) และหน่วยป้อนโปรแกรมในเครื่องเดียวกัน ข้อดีของเครื่องควบคุมดังกล่าวคือ สามารถจะใช้ติดต่อกับอุปกรณ์ภายนอกได้เป็นจำนวนมาก เปลี่ยนแปลงลักษณะและแก้ไขการควบคุมได้โดยง่าย นอกจากนี้การประมวลผลยังกระทำได้อย่างรวดเร็ว

บทที่ 1

บทนำ

กระบวนการ (Process) ต่างๆ ในอุตสาหกรรมส่วนใหญ่ จะมีระบบการทำงานแบบเป็นลำดับขั้น (Sequence Control) ซึ่งแต่เดิมจะใช้แต่อุปกรณ์ไฟฟ้าเชิงกล (Electromechanical Device) อันได้แก่ Relay, Timer, Counter และอื่นๆ ลักษณะการทำงานของระบบนี้มีเพียง 2 สถานะคือ เปิด (ON) และปิด (OFF) ซึ่งมีขนาดใหญ่และเปลี่ยนแปลงการทำงานได้ยาก และไม่สามารถประยุกต์ใช้กับระบบควบคุมที่ซับซ้อนได้

ในปัจจุบันจึงได้นำ Microprocessor มาใช้เป็นหัวใจในการควบคุมแทนอุปกรณ์แบบเก่า ทำให้มีความสามารถสูงขึ้น ใช้งานได้สะดวก และกว้างขวางมากขึ้น เครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Controller) ซึ่งเป็นที่รู้จักกันดีในชื่อย่อว่า "PLC" จัดเป็นเครื่องควบคุมแบบใหม่ชนิดหนึ่งที่มีใช้กันอยู่ทั่วไปในกระบวนการต่างๆ ทางอุตสาหกรรมในประเทศที่พัฒนาแล้ว สามารถใช้แทนอุปกรณ์ไฟฟ้าเชิงกลได้เป็นจำนวนมาก

ภายใน PLC ประกอบด้วย Microcontroller MCS 51 เบอร์ 8032 หน่วย ความจำแบบ ROM และ RAM อุปกรณ์ติดต่อแบบขนาน (8255 PIO) อุปกรณ์นับและตัวตั้งเวลา (Timer & Counter) คีย์บอร์ด (Membrane Switch) จอภาพแสดงผลแบบ LCD และอุปกรณ์อิเล็กทรอนิกส์อื่นๆ โดยสามารถสร้างเงื่อนไขควบคุมการทำงานต่างๆ ด้วย Software ที่สร้างเป็นชุดคำสั่งต่างๆ ของตัวเอง โดยเฉพาะ ง่ายแก่การป้อนโปรแกรม แก้ไขโปรแกรม ตรวจสอบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

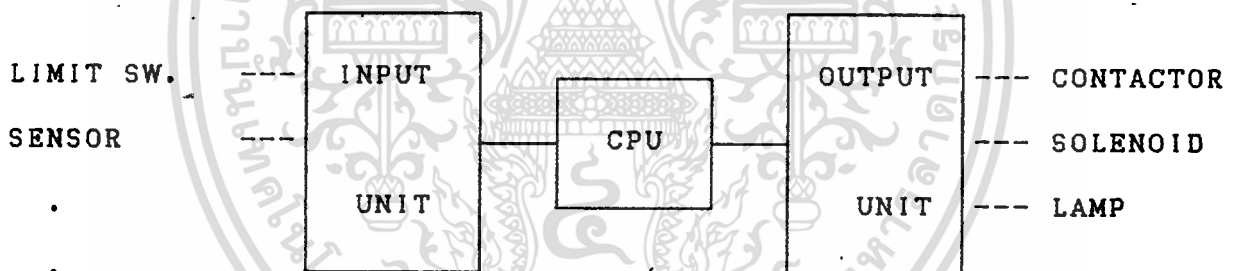
## บทที่ 2

### ทฤษฎีพื้นฐานและหลักการของ PLC

#### 2.1 โครงสร้างของ PLC

PLC ประกอบด้วยส่วนสำคัญ 2 ส่วนคือ

1. หน่วยประมวลผลกลาง (CPU Unit)
2. หน่วยอินพุทเอาต์พุท (Input/Output Unit)



รูปที่ 2.1 โครงสร้างของ PLC

นอกจากส่วนประกอบหลัก 2 ส่วนข้างต้นแล้ว PLC ยังประกอบด้วย หน่วยป้อนโปรแกรม (Programming Unit) ทำหน้าที่ติดต่อระหว่าง PLC กับผู้ใช้ รับ Program ที่ผู้ใช้เขียนขึ้นเก็บไว้ในหน่วยความจำ ปกติแล้วหน่วยป้อนโปรแกรมจะเชื่อมต่อกับ PLC เมื่อผู้ใช้ต้องการป้อน ตรวจสอบ หรือแก้ไขโปรแกรมเท่านั้น และ PLC เองก็สามารถทำงานได้โดยไม่ต้องพึ่งหน่วยป้อนโปรแกรม ดังนั้นตามปกติแล้วหน่วยป้อนโปรแกรมจึงไม่จัดเป็นส่วนประกอบของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

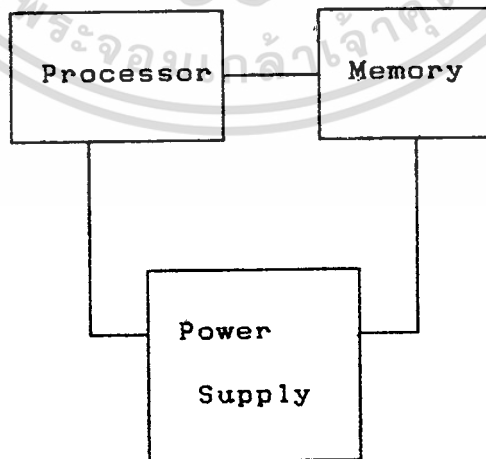
## 2.2 หน่วยประมวลผลกลาง (CPU UNIT)

เป็นหน่วยที่ทำหน้าที่ประมวลผล โดยจะตรวจรับข้อมูล Input จากอุปกรณ์ตรวจวัด (Sensor Device) แล้วทำการประมวลผลตามโปรแกรมเงื่อนไขที่ได้กำหนดไว้ในหน่วยความจำ จากนั้นก็ส่งสัญญาณ Output ออกไปเพื่อทำการควบคุม Process ต่างๆ

หน่วยประมวลผลกลางนี้ประกอบด้วย

1. หน่วยประมวลผล (Processor)
2. หน่วยความจำ (Memory)
3. แหล่งจ่ายพลังงาน (Power Supply)

รูปที่ 2.2 แสดงโครงสร้างของหน่วยประมวลผลกลาง โดยหน่วยประมวลผลทำหน้าที่ควบคุมการทำงานทั้งหมดของเครื่องนำข้อมูลมาจากหน่วยความจำ มาปฏิบัติเพื่อควบคุมอุปกรณ์ภายนอกผ่านหน่วย Input-Output ในขณะที่หน่วยจ่ายพลังงานทำหน้าที่จ่ายแรงดันไฟฟ้าให้กับหน่วยประมวลผลและหน่วยความจำ



รูปที่ 2.2 โครงสร้างของหน่วยประมวลผลกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1 หน่วยประมวลผล (PROCESSOR)

ทำหน้าที่ตรวจสอบสถานะต่างๆ ในหน่วยความจำและประมวลผลตามโปรแกรมต่างๆที่สร้างขึ้น (User Program) ควบคุมการติดต่อรับส่งข้อมูลระหว่าง CPU กับหน่วย Input-Output และหน่วย Input-Output กับอุปกรณ์ภายนอก ติดต่อกับผู้ใช้และอุปกรณ์ร่วม ตรวจสอบสภาพการทำงานของ PLC โดยโปรแกรมบริหารระบบ (Operating System) เป็นผู้ควบคุมอีกทีหนึ่ง

### 2.2.2 หน่วยความจำ (MEMORY)

เป็นองค์ประกอบที่สำคัญของระบบ เพราะใช้เป็นที่เก็บโปรแกรม และข้อมูลขนาดของหน่วยความจำ จะเป็นสิ่งที่กำหนดความสามารถของระบบ ปกติมักจะมีขนาดวัดกันเป็น step ของคำสั่ง ในการโปรแกรมระบบที่มี ขนาดของหน่วยความจำมาก จะทำให้ผู้ใช้สามารถเขียน โปรแกรมที่มีความซับซ้อนได้มากขึ้น เราจะแบ่งหน่วยความจำออกเป็น 2 ส่วนคือ

1. หน่วยความจำระบบ (SYSTEM MEMORY) เก็บโปรแกรมบริหารระบบและข้อมูลของระบบ
2. หน่วยความจำผู้ใช้ (USER MEMORY) เก็บโปรแกรมผู้ใช้ ข้อมูลของหน่วย Input-Output และอุปกรณ์ภายใน หน่วยความจำที่นำมาใช้กับ PLC มี 2 ชนิดเช่นเดียวกับที่ใช้ Computer ทั่วไป คือ

1. VOLATILE

2. NONVOLATILE

ข้อมูลภายในหน่วยความจำแบบ Volatile จะสูญหายหมดถ้าไม่มีกระแสไฟฟ้า (สามารถใส่ Battery Backup ได้) ส่วนข้อมูลภายในหน่วยความจำแบบ Nonvolatile จะยังคงอยู่ถึงแม้ว่าจะไม่มีกระแสไฟฟ้า โปรแกรมบริหารระบบ (Supervisory Program หรือ Operating System) และข้อมูลของระบบจะเก็บอยู่ในหน่วยความจำแบบ Nonvolatile เช่น PROM EPROM EEPROM ในขณะที่โปรแกรมผู้ใช้ (User program) อาจเก็บอยู่ในหน่วยความจำแบบใดก็ได้ ถ้าเป็นโปรแกรมที่ไม่ต้องการเปลี่ยนแปลงแก้ไขอีกก็จะเก็บในลักษณะเดียวกับโปรแกรมบริหารระบบ แต่ถ้ายังมีความจำเป็นต้องมีการตรวจสอบแก้ไขโปรแกรมนั้น อยู่ก็จะเก็บไว้ในหน่วยความจำประเภท RAM โดยอาจมีหน่วยจ่ายพลังงานสำรองชั่วคราวขณะไฟฟ้ามดับ ส่วนข้อมูลที่เป็นข้อมูลของหน่วย Input-Output และอุปกรณ์ภายใน (Data Memory) จะถูกเก็บอยู่ในหน่วยความจำแบบ RAM เท่านั้น เพราะเป็นข้อมูลที่มีการประมวลผลเปลี่ยนแปลงตลอดเวลา โดยอาจจะมีหน่วยจ่ายพลังงานสำรองหรือไม่ก็ได้

### 2.3 INPUT-OUTPUT UNIT

เป็นตัวกลางเชื่อม การติดต่อระหว่างหน่วยประมวลผลกลาง กับอุปกรณ์ภายนอก โดยที่หน่วย Input ทำหน้าที่ รับค่าสภาวะจากอุปกรณ์ตรวจวัด และส่งผลที่ประมวลได้ไปที่หน่วย Output

### 2.3.1 INPUT UNIT

หน่วยInputทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอก ที่เป็น SW. และตัวตรวจจับต่างๆ แล้วแปลงชนิดของสัญญาณเข้าดังกล่าวไม่ว่าจะเป็น AC, DC ให้เป็นสัญญาณที่เหมาะสม เพื่อส่งเข้าไปให้แก่หน่วยประมวลผลกลาง ดังนั้นในการเลือกใช้ประเภทของ Input นั้น ผู้ใช้จะต้องเลือกใช้ให้เหมาะสม และถูกต้องตามประเภทการใช้งานด้วย ไม่งั้นนั้นอาจจะเกิดความเสียหายขึ้นได้ ปกติInputที่ดีควรจะต้องมีหน้าที่ดังต่อไปนี้

1. เปลี่ยนแปลงระดับของสัญญาณเข้า ให้เป็นระดับสัญญาณ ที่เหมาะสมกับระบบของ PLC
2. การส่งสัญญาณระหว่างหน่วยInputกับหน่วยประมวลผลจะติดต่อกันด้วยลำแสง โดยอาศัยอุปกรณ์ประเภท Photo Transister ทั้งนี้เพื่อแยกสัญญาณ Isolate ทางไฟฟ้าออกจากกัน เป็นการป้องกันไม่ให้หน่วยประมวลผลได้รับความเสียหายเมื่อหน่วยInput เกิดลัดวงจร
3. ไม่มีการลั่นของหน้าสัมผัส (Contract Chattering)

### 2.3.2 OUTPUT UNIT

หน่วยOutputทำหน้าที่รับคำสั่งภาวะที่ได้จากการประมวลของหน่วยประมวลผล เพื่อนี้คำสั่งภาวะเหล่านี้ไปควบคุมอุปกรณ์ภายนอก เช่น Relay Solenoid Lamp แสดงสถานะเป็นต้น นอกจากนั้นยังทำหน้าที่ Isolate สัญญาณของหน่วยประมวลผลออกจากอุปกรณ์Output โดยปกติแล้วหน่วยOutputมีความสามารถในการขับLoadด้วยกระแส 1-2 Amp. ในกรณีนี้Loadต้องการกระแสมากกว่านี้ ผู้ใช้จะต้องนำไปต่อเข้ากับอุปกรณ์ขับหรือขยายอีกทีหนึ่ง

Output ของPLCเป็นแบบใช้กับไฟกระแสตรง ซึ่งมีอยู่ 2 ชนิดด้วยกันคือ

Relay และ Transister

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 INPUT OUTPUT PROCESSING

การรับข้อมูลของหน่วย Input-Output มาประมวลผลตามโปรแกรมของผู้ใช้ และส่งข้อมูลสุดท้ายที่ได้จากการประมวลผลไปที่หน่วย Output เรียกว่าการ SCAN ซึ่งมีอยู่ 2 วิธีคือ

### 1. CONTINUOUS UPDATING

### 2. MASS INPUT-OUTPUT COPYING

#### 2.4.1 CONTINUOUS UPDATING

การ SCAN วิธีนี้จะ Scan ทีละคำสั่ง ขณะที่ CPU กำลัง Scan Input อยู่ในโปรแกรมคำสั่งจะมีการหน่วงเวลาเกิดขึ้น (Delay) เพื่อให้แน่ใจว่า Input ที่ได้มานั้นถูกต้อง (โดยทั่วไปการหน่วงจะใช้เวลาประมาณ 3 ms. เพื่อป้องกันการเกิดสัญญาณ Bounce ที่เกิดจากการกดสวิตช์และสัญญาณรบกวนอื่นๆที่จะเข้าไปใน PLC) Output จะขับอุปกรณ์โดยตรง เมื่อได้ปฏิบัติตามโปรแกรม และจะ Latch ค่าไว้ในหน่วย Input-Output (ส่วนของหน่วยความจำ) เพื่อรักษาสถานะของ Output เอาไว้จนกว่าจะทำคำสั่งใหม่เสร็จ

#### 2.4.2 MASS INPUT-OUTPUT COPYING

PLC ที่มีขนาดใหญ่มี Input-Output เป็น 100 ตัว ถ้าทำแบบวิธี Continuous Updating จะทำงานได้ช้ามากเพราะต้องใช้เวลารั้งถึง 3ms. ต่อคำสั่ง สำหรับการ Scan วิธี Mass I/O Copying นี้จะ Scan ทีละโปรแกรมเลยคือจากต้นโปรแกรมจนถึงจบโปรแกรมเลย คือการ Scan หนึ่งรอบนั่นเอง (Scan Time) ซึ่งจะใช้เวลาประมาณ 1-100 ms. ขึ้นอยู่กับขนาดความยาวของโปรแกรมของผู้ใช้ และชนิดของจำนวนของหน่วย Input-Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การScan จะเริ่มด้วยการรับค่าสถานะ ของอุปกรณ์ภายนอกจากหน่วย Input-Output มาเก็บไว้ในหน่วยความจำ(RAM) แล้วนำโปรแกรมควบคุมที่ผู้ใช้เขียนขึ้นมาทีละคำสั่ง ซึ่งขึ้นอยู่กับค่าของInputที่รับเข้ามา โดยเริ่มจากคำสั่งแรกจนถึงคำสั่งสุดท้าย ซึ่งถือว่าสิ้นสุดโปรแกรมในหน่วยความจำ ถ้าการทำงานของโปรแกรมทำให้สถานะของ Outputเปลี่ยนแปลง ผลดังกล่าวจะถูกเก็บไว้ที่ส่วนOutputของ I/O RAM และเมื่อจบโปรแกรมแล้วหน่วยI/O ส่งการเปลี่ยนแปลงของOutput จากI/O RAM ไปที่หน่วย Output สถานะของOutputจะถูกLatchไว้ก่อน จนกว่าจะทำการScanรอบใหม่เสร็จ การScanแบบนี้ขณะที่กำลังปฏิบัติตามโปรแกรมอยู่ จะไม่ยอมรับInputจนกว่าจะปฏิบัติเสร็จ ซึ่งค่าที่Inputเปลี่ยนแปลง ขณะที่กำลังปฏิบัติตามโปรแกรมอยู่นั้น จะถูกเก็บไว้ในRamก่อน ซึ่งอาจทำให้เกิดความผิดพลาดได้แต่ก็น้อยมาก เพราะช่วงเวลาการปฏิบัติ(Executed) ตามโปรแกรมน้อยมาก

(a)  
Start

Fetch, decode and execute first instruction	Scan necessary contacts	Next instruction	Scan or actuate devices	Next instruction	Scan I/O or actuate O/P	etc
Typical time delay = 5 $\mu$ s	3 ms delay	5 $\mu$ s	3 ms delay	5 $\mu$ s	3 ms delay	

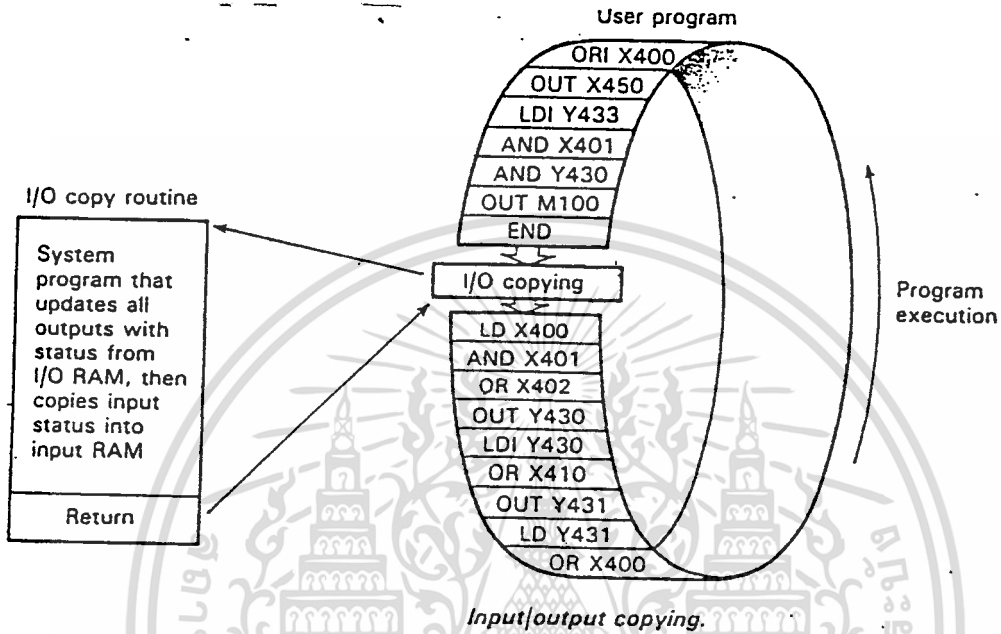
(b)

Start	I/O copy	Program execution	End I/O copy	Prog
	Copy all inputs into RAM	Fetch, decode and execute all instructions in sequence	Copy all outputs from O/P RAM; to output unit, and inputs into RAM	
		Time depends on length of total program e.g. 1K program = 5 ms	Fixed length delay e.g. 5 ms	

*Input scanning and reaction times: (a) continuous updating; (b) mass I/O copying.*

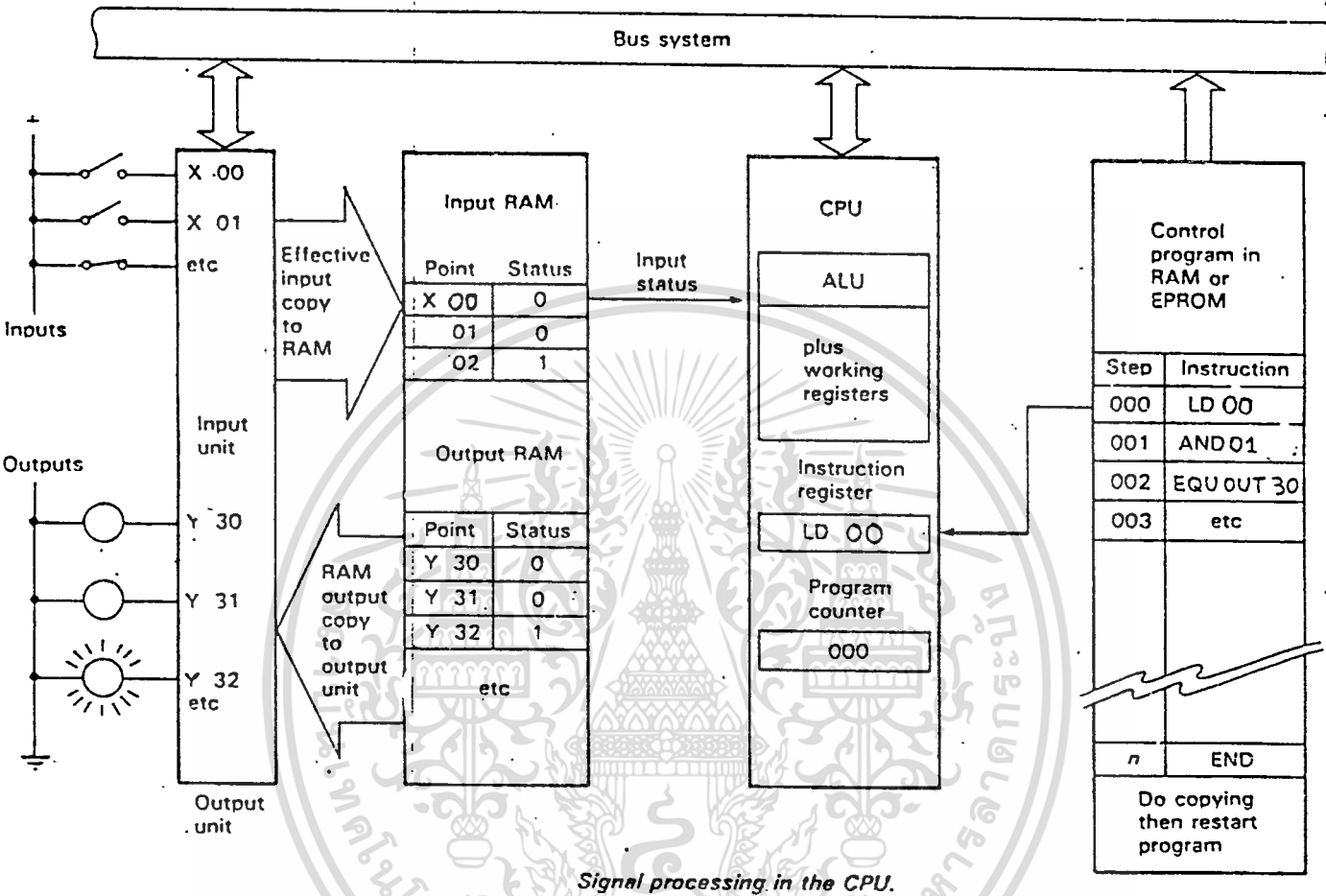
รูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4

ช่วงเวลา Scan Time ของ PLC จะเป็นตัวบอกถึงความสามารถของ PLC ในการตรวจสอบ และตอบสนองต่อการเปลี่ยนแปลงสถานะ ของอุปกรณ์ภายนอกและควบคุม เครื่องจักรหรือกระบวนการผลิตว่ามีมากน้อยเพียงไร เช่น PLC ที่ช่วงเวลา Scan 10ms. ย่อมไม่สามารถรับสถานะที่แท้จริงของอุปกรณ์ที่มีการเปลี่ยนแปลงทุก 4 ms. ถ้าใช้ PLC ดังกล่าวจะทำให้การควบคุมผิดพลาดหมด



Signal processing in the CPU.

รูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 หน่วยป้อนโปรแกรม (PROGRAMMING UNIT)

หน่วยป้อนโปรแกรม ทำหน้าที่ป้อนโปรแกรมลงในหน่วยความจำ ซึ่งโปรแกรมบริหารระบบจะนำไปประมวลผลเพื่อควบคุมอุปกรณ์ภายนอกอีกทีหนึ่ง การตรวจสอบแก้ไขโปรแกรมผู้ใช้ การแสดงค่าสถานะของ I/O ตัวตั้งเวลา ตัวนับเวลาต่างๆ ก็สามารถดำเนินการได้ด้วยหน่วยป้อนโปรแกรมนี้

## 2.6 การใช้งานจอภาพแสดงผลแบบ DOT MATRIX LCD MODULE

จอภาพแสดงผลแบบ Dot Matrix LCD Module เป็นส่วนแสดงผลที่มีประสิทธิภาพสูง สามารถแสดงตัวอักษร ตัวเลขต่างๆ ได้ครบถ้วน อีกทั้งสามารถตัวอักษรชี้ใช้เองได้อีกด้วย โดยเขียนข้อมูลเก็บไว้ใน Character Generator RAM ของ LCD Module

ภายใน Dot Matrix LCD Module จะมีหน่วยควบคุมการทำงาน (Controller) อยู่ในตัวเอง นั่นคือ HD44780 Dot Matrix LCD Controller & Driver ดังนั้นเพียงแต่เราเขียนคำสั่งควบคุมต่างๆ ตามลำดับขั้นตอนที่ถูกต้อง ก็สามารถควบคุมและใช้งาน LCD นี้ได้ตามต้องการ

### 2.6.1 INITIALIZATION

ก่อนเริ่มใช้งาน LCD Module ต้องมีการ Initialization ให้แก่ HD44780 (Controller) เสียก่อนซึ่งมีอยู่ 2 วิธีคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Initialization by internal reset circuit

ตามปกติแล้วทุกครั้งที่เราเริ่มจ่ายไฟเลี้ยงให้กับ Module HD44780 จะทำการ Reset (Initialization) โดยอัตโนมัติ ด้วยวงจรรภายในตัวเองซึ่งมีลำดับขั้นตอนตามคำสั่งข้างล่างนี้ และ Busy Flag จะมีค่า = 1 จนกระทั่งสิ้นสุด Initialization ซึ่งใช้เวลาประมาณ 10ms. หลังจาก Vcc มีค่า = 4.5 volts

- Display Clear

- Function Set

DL = 1 : รับส่งข้อมูลขนาด 8 บิต

N = 0 : แสดงผล 1 บรรทัด

F = 0 : 5\*7 dot character font

-Display ON/OFF

Control

D = 0 : display OFF

C = 0 : cursor OFF

B = 0 : blink OFF

-Entry mode set

I/D = 1 : +1 (increment)

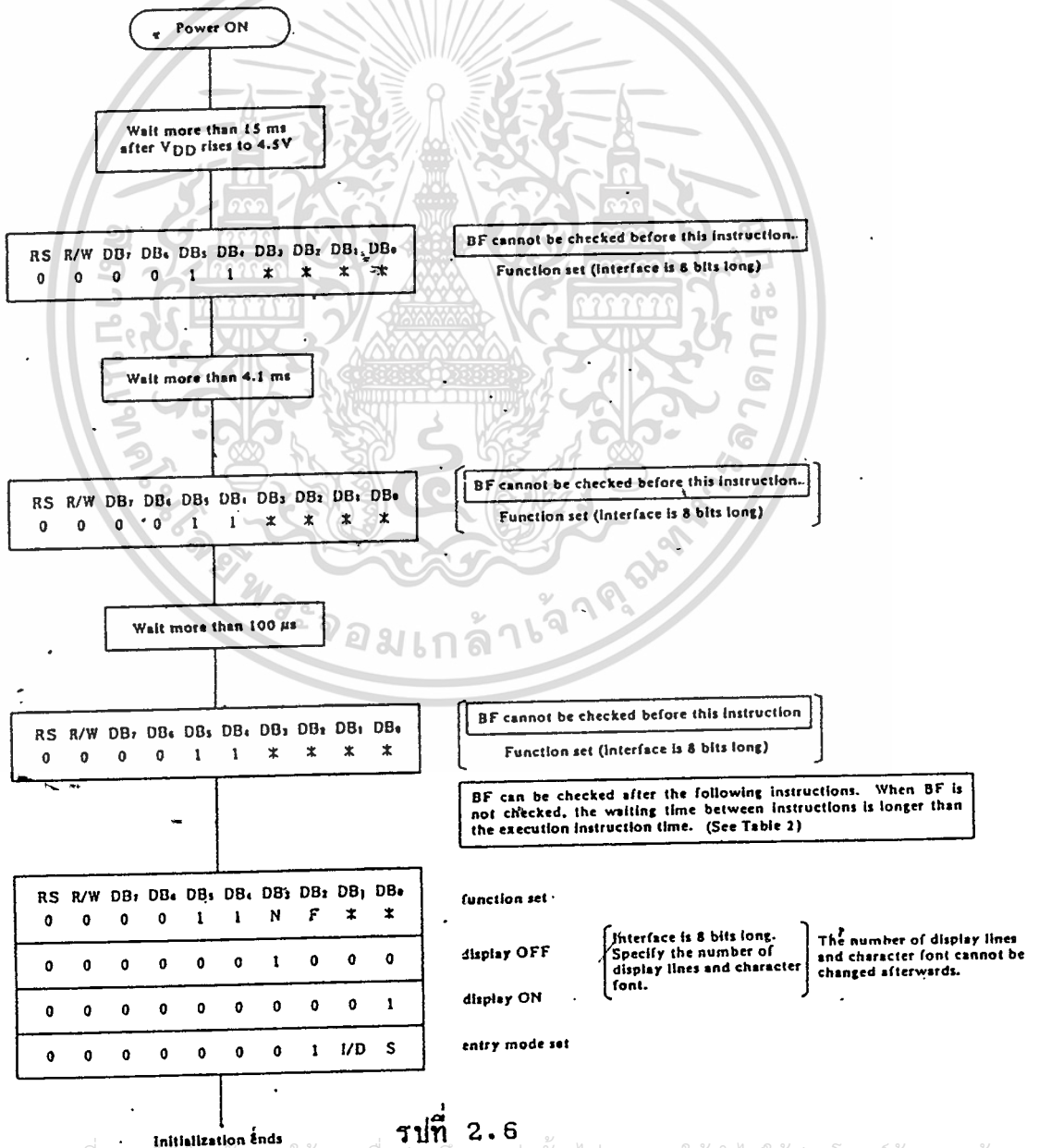
s = 0 : ไม่มีการเลื่อนจอแสดงผล

-Write DD RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. Initialization ด้วย คำสั่ง

ในกรณีที่ Internal reset circuit ไม่สามารถทำงานได้ตามปกติ (ซึ่งมักจะ  
มีผลมาจากไฟเลี้ยง) เป็นผลให้ Initialization ไม่ได้ด้วยตัวเอง จึงมีความจำเป็น  
ต้องดำเนินการ ด้วยคำสั่งดัง FLOW CHART ดังในรูป 2.6



รูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 หน่วยจ่ายพลังงาน (POWER SUPPLY)

หน่วยจ่ายพลังงานทำหน้าที่จ่ายกระแสไฟฟ้าให้ส่วนต่างๆของPLC คือ หน่วยประมวลผล หน่วยความจำ หน่วยInput-Output โดยรักษาระดับแรงดันไฟฟ้าให้คงที่ "ความน่าเชื่อถือของ PLC ขึ้นกับการทำงานของหน่วยจ่ายพลังงาน"



## บทที่ 3

### การออกแบบสร้าง PLC

#### 3.1 หน่วยประมวลผลกลาง

หน่วยประมวลผล นับเป็นหัวใจของหน่วยประมวลผลกลาง ซึ่งจะใช้ไมโครคอนโทรลเลอร์เบอร์ 8032 (U1) ต่อร่วมกับอุปกรณ์ต่อจำหน่วยความจำ EPROM และ ROM

##### 3.1.1 โครงสร้างสถาปัตยกรรม MCS-51

ตระกูล MCS-51 จะมีทั้งแบบมี ROM ในตัว หรือไม่มี ROM หรือมี EPROM บนชิป เดียวกันและจะมีตำแหน่งขาที่เหมือนกัน รูปที่ 3.1 แสดงถึงตารางรายละเอียดของเบอร์ต่างๆ ในตระกูล MCS-51 ที่มีจำหน่ายในท้องตลาด

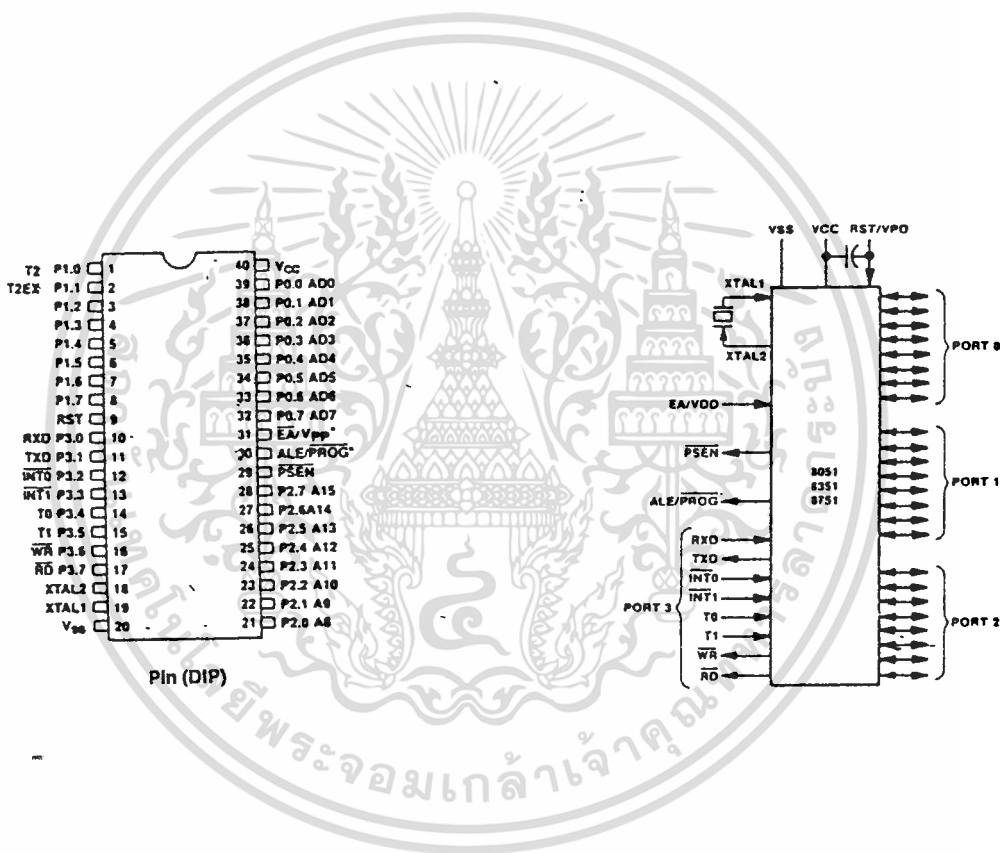
เบอร์	หน่วยความจำภายใน		ตัวตั้งเวลา /	อินเตอร์รัพต์
	โปรแกรม	ข้อมูล	ตัวนับจำนวน	
8052 AH	8K x 8 ROM	256 x 8 RAM	3 x 16 BIT	6
8051 AH	4K x 8 ROM	128 x 8 RAM	2 x 16 BIT	5
8051	4K x 8 ROM	128 x 8 RAM	2 x 16 BIT	5
8032 AH	ไม่มี ROM	256 x 8 RAM	3 x 16 BIT	6
8031 AH	ไม่มี ROM	128 x 8 RAM	2 x 16 BIT	5
8031	ไม่มี ROM	128 x 8 RAM	2 x 16 BIT	5
8751 H	4K x 8 EPROM	128 x 8 RAM	2 x 16 BIT	5
8751 H-12	4K x 8 EPROM	128 x 8 RAM	2 x 16 BIT	5

รูปที่ 3.1 ตารางรายละเอียดของตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 การจัดขาลักษณะภายนอกของ MCS-51

รูปที่ 3.2 แสดงการจัดขาตามลักษณะภายนอกของชิป MCS-51 ซึ่งมีรายละเอียดดังนี้คือ



(A) ลักษณะขาภายนอกของ MCS-51

(B) สัญญลักษณ์ทางตรรกของ MCS-51

รูปที่ 3.2

-ขา V<sub>ss</sub> (ขา 20) เป็นขาสำหรับต่อลงดิน

-ขา V<sub>cc</sub> (ขา 40) เป็นขาที่ต่อแรงดันไฟกระแสตรงขนาด 5 V. และใช้

#### สำหรับการโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา PORT 0 (PO.0-PO.7/ADO-AD7) (ขา 32-39) เป็นพอร์ตไอโอ 8 บิต แบบ Open Drain Bidirectional สามารถที่จะรับโหนดทีทีแอลได้ 8 ตัว การเขียนค่า "1" ไปที่พอร์ตนี้ จะเป็นการปล่อยลอย (float) ขาของพอร์ตนี้ ทำให้ทำงานเป็นอินพุต มีสถานะอิมพีแดนซ์สูง ในการให้พอร์ตนี้บริการแบบ ไอโอ พอร์ต 0 จะทำงานเป็นมัลติเพลกซ์ ด้วยสัญญาณแอดเดรสไบต์ต่ำกับข้อมูล สำหรับการใช้งานด้านหน่วยความจำภายนอกในการใช้งานแบบนี้จะใช้ลักษณะภายในเป็นตัวพูลอัพ พอร์ต 0 ยังใช้งานเป็นตัวส่งข้อมูลออกจากพอร์ตนี้ เมื่อใช้บริการตรวจสอบโปรแกรม ROM ภายใน และการโปรแกรมตัว EPROM ภายใน ถ้าใช้งานในลักษณะนี้การพูลอัพจากภายนอกจะต้องด้วยค่า 10 กิโลโอมห์

- ขา PORT 1 (P1.0 - P1.7) (ขา 1-8) เป็นพอร์ตไอโอ 8 บิตแบบ open Drain Bidirectional พร้อมด้วยการพูลอัพภายในถ้าเป็นพอร์ตเอาต์พุต บัฟเฟอร์สามารถรับโหนดทีทีแอลตระกูลแอลเอสไอได้ 4 ตัว พอร์ต 1 เมื่อถูกเขียนค่า "1" ด้วยโปรแกรม มันจะมีสถานะสูงด้วยการพูลอัพภายใน การให้สถานะเช่นนี้ จะเป็นการ Inicial ใช้งานพอร์ตนี้ให้เป็นอินพุต ขณะที่พอร์ต 1 เป็นอินพุต การให้สัญญาณลงต่ำ จะเป็นการจ่ายกระแสออกเนื่องจากการพูลอัพภายใน

- ขา PORT 2 (P2.0 - P2.7) (ขา 21-28) เป็นพอร์ตไอโอ 8 บิตแบบ Open Drain Bidirectional ด้วยการพูลอัพภายใน พอร์ต 2 ที่ทำหน้าที่เป็นบัฟเฟอร์เอาต์พุตสามารถจ่ายโหนดทีทีแอลตระกูลแอลเอสไอได้ 4 ตัว พอร์ตจะถูกใช้งานเป็นตัวส่งแอดเดรสไบต์สูงด้วยเมื่อใช้งานร่วมกับหน่วยความจำภายนอกเพื่อให้แอดเดรสได้ถึง 16 บิต ด้วยการใช้งานแบบนี้ มันจะมีพูลอัพภายใน ที่ช่วยให้การส่งค่า "1" ได้ระดับที่แน่นอน นอกจากการใช้งานสำหรับแอดเดรสอันดับสูงยังใช้เป็นขาควบคุม ในการใช้งานตรวจสอบและเขียนโปรแกรมเบอร์ 8751 และตรวจสอบโปรแกรมภายใน 8051

- ขา PORT 3 (P3.0-P3.7) (ขา 10-17) เป็นพอร์ตไอโอ 8 บิตแบบ พูลอัพภายใน นอกจากทำเป็นพอร์ตไอโอ ที่สามารถรับโหนดทีทีแอลพวกตระกูลแอลเอสไอได้ 4 ตัวแล้วยังใช้งานเป็นพิเศษสำหรับตระกูล MCS-51 ตามรายการข้างล่างนี้ด้วย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ขาพอร์ต</u>	<u>ขา</u>	<u>การทำงานตามฟังก์ชันพิเศษ</u>
P3.0	10	RxD พอร์ตอนุกรมอินพุต
P3.1	11	TxD พอร์ตอนุกรมเอาต์พุต
P3.2	12	INT0 อินเตอร์รัพภายนอกตัวที่ 1
P3.3	13	INT1 อินเตอร์รัพภายนอกตัวที่ 2
P3.4	14	TO สัญญาณกระตุ้นเข้าที่ตัวตั้งเวลาและตัวนับ 0
P3.5	15	T1 สัญญาณกระตุ้นเข้าที่ตัวตั้งเวลาและตัวนับ 1
P3.6	16	WR สัญญาณควบคุมการเขียน
P3.7	17	RD สัญญาณควบคุมการเขียน

การที่จะให้ทำงานตามฟังก์ชันข้างบน จะต้องเริ่มโปรแกรมด้วยการส่งค่า "1" ไปแลตซ์ไว้ก่อนที่ให้ทำงานตามฟังก์ชันข้างบน

-ขา RST (ขา 9) ต้องคงสถานะค่าสูงเป็นเวลายาวนาน้อย สองวัฏจักรระหว่างที่ออสซิลเลเตอร์ทำงานขณะที่ต้องการรีเซ็ตทั้งระบบงาน โดยจะต่อรีจิสเตอร์พูลดาวน์ (8.2 กิโลโอมห์) จากขา RES ไปลงดินและเพื่อให้ตัวชิปรีเซ็ตได้โดยอัตโนมัติ ขณะเปิดไฟจะใช้คาปาซิเตอร์ (10 ไมโครฟารัด) ต่อคร่อมระหว่างขา RST กับ ขา Vcc

-ขา ALE / PROG (ขา 30) เป็นขาแอดเดรสแลตซ์อินพุตเข้าด้วยการส่งพัลส์ออกไปใช้สำหรับแลตซ์ค่าแอดเดรสไบต์ต่ำจากพอร์ต 0 ในระหว่างการเข้าถึงข้อมูลจากหน่วยความจำภายใน ALE จะถูกส่งสัญญาณนาฬิกาออกมา ในอัตราความเร็วคงที่ที่ 1/8 ของความถี่ออสซิลเลเตอร์ตลอดเวลา แม้ว่าอาจจะไม่มีการเข้าถึงข้อมูลจากภายใน ดังนั้นจึงสามารถที่จะใช้สัญญาณจากขานี้เป็นตัวตั้งเวลาภายนอกหรือเป็นความถี่สัญญาณนาฬิกา แต่อย่างไรก็ตาม ความถี่สัญญาณนี้จะลดความถี่ช้าลงไปเท่าหนึ่งคองระหว่างการทำงานแบบการเข้าถึงของหน่วยความจำข้อมูลภายนอก ขานี้ยังใช้เป็นสัญญาณพัลส์เข้า สำหรับการควบคุมการโปรแกรม EPROM ภายในชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ขา PSEN (ขา 29) Program Storage Enable เป็นสวิตช์อ่านข้อมูลจากโปรแกรมหน่วยความจำภายนอก เมื่อชิปทำงานด้วยโปรแกรมภายนอก ขา PSEN จะสร้างสวิตช์ต่ำสองครั้งภายในแต่ละวัฏจักรแมซิน สัญญาณจะมีสถานะสูง หรือพัลส์ต่ำทั้งสองลูกจะหายไป เมื่อทำงานในช่วงการอ่านหรือเขียนข้อมูลจากหน่วยความจำข้อมูลภายนอก และไม่มีพัลส์ส่งออก ถ้าชิปทำงานด้วยโปรแกรมหน่วยความจำภายใน

-ขา EA/Vpp (ขา 31) มีสถานะสูง ตัวชิปในชิปจะทำงานตามโปรแกรมที่อยู่ในหน่วยความจำภายใน (โดยที่โปรแกรมจะต้องไม่ยาวกว่า 4 กิโลไบต์ สำหรับเบอร์ 8051 AH และ 8 กิโลไบต์ สำหรับเบอร์ 8052 AH) การทำให้ EA มีสถานะต่ำจะเป็นการควบคุมให้ชิปทำงานตามโปรแกรมหน่วยความจำภายนอก ซึ่งขยายโปรแกรมได้ยาวถึง 64 กิโลไบต์ ในตัว 8031 AH และ 8032 AH ขา EA จะต้องต่อลงดินเช่นกัน แม้ว่าจะไม่มี ROM อยู่ภายในก็ตาม ในตัว 8751 H จะใช้ขานี้จ่ายแรงดัน 21 V. ขณะทำการเขียนโปรแกรมเข้า EPROM ของชิป 8751 H ตัวนี้

-ขา XTAL1 ขา 19 ใช้เป็นตัวอินพุตเข้าสู่ตัวออสซิลเลเตอร์ขยายแบบ Invert

-ขา XTAL2 ขา 18 ใช้เป็นตัวเอาต์พุตเข้าสู่ตัวออสซิลเลเตอร์ขยายแบบ Invert

### 3.1.3 หน่วยประมวลผลกลาง (CPU)

CPU เป็นมันสมองของระบบไมโครคอมพิวเตอร์ การอ่านโปรแกรม และทำงานตามคำสั่งโปรแกรมจะกระทำที่ส่วนนี้ โดยการใช้ส่วนคณิตศาสตร์ และตรรกศาสตร์ทำงานร่วมกับเรจิสเตอร์ A, B, PSW(Program Status Word), SP (Stack pointer) ตัวนับโปรแกรม (PC: Program Counter) ขนาด 16 บิต และตัวชี้ตำแหน่งข้อมูล (DPTR: DataPointer) ส่วนคณิตศาสตร์และตรรกศาสตร์(ALU:Arithmetic Logic Unit) ALU นี้ทำงานในฟังก์ชันทางคณิตศาสตร์และตรรกศาสตร์ด้วยตัวแปรต่างๆขนาด 8 บิต ที่มีลักษณะการทำงานทางคณิตศาสตร์เป็น บวก ลบ คูณ หาร รวมทั้งทางตรรกศาสตร์ เช่น AND OR XOR รวมทั้งการเลื่อนและวนรอบบิต การเคลียร์ค่าและกลับค่า (Complement) เป็นต้น AUL ยังสามารถที่จะตัดสินใจในการให้กระโดดไปทำคำสั่งของโปรแกรมในส่วนอื่นๆ ตามเงื่อนไขที่ตั้งขึ้นและยังแบบเรจิสเตอร์ชั่วคราวไว้สำหรับเป็นทางผ่านชั่วคราว ของข้อมูลในการถ่ายเทภายในระบบคำสั่งอื่นที่มีการ AUL ยังมีความสามารถที่จะเพิ่มค่าในเรจิสเตอร์ในลักษณะการบวกด้วยหนึ่ง (Increment) หรือ คำนวณเลขที่อยู่ของข้อมูลที่จะนำไปเก็บ หรือการลดค่าลงครึ่งละหนึ่ง

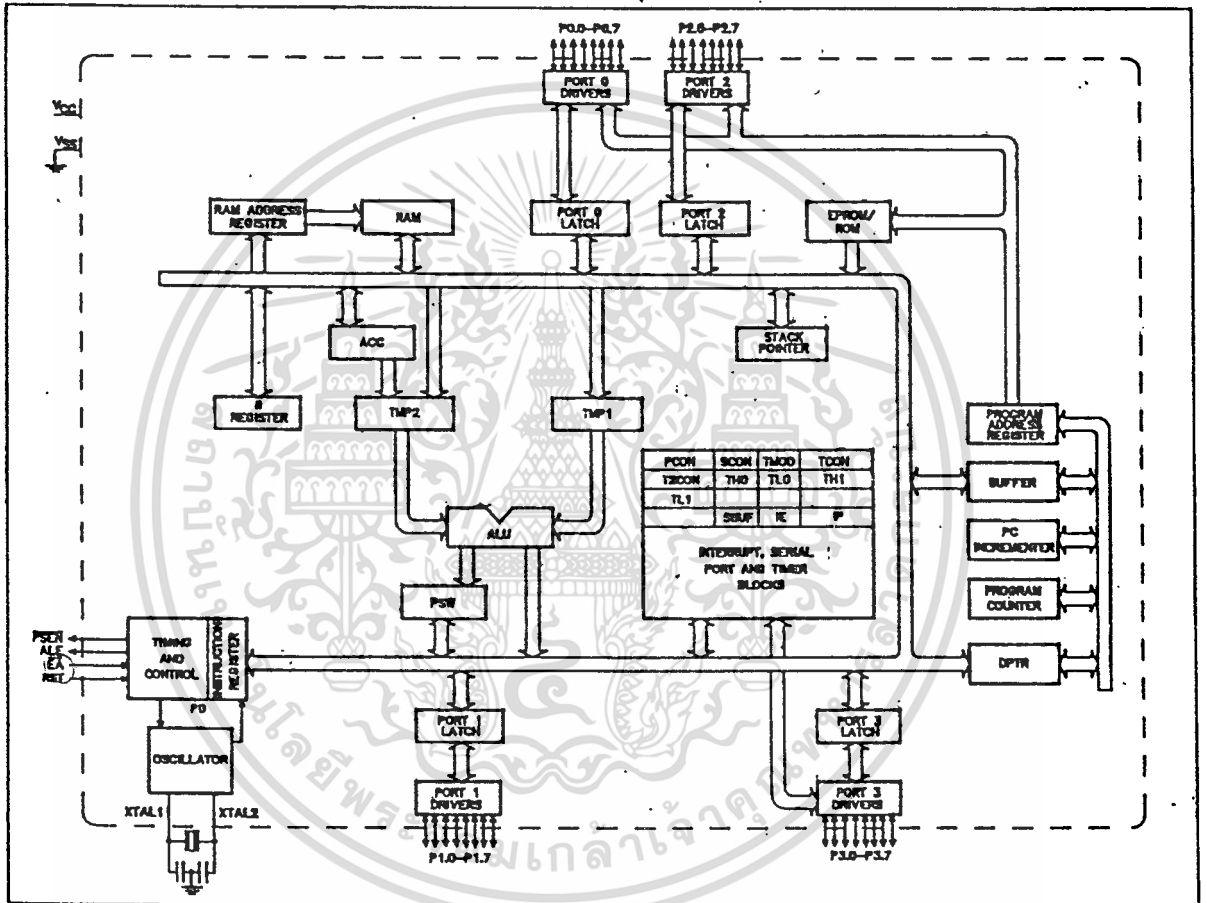
ในลักษณะการลบด้วยค่าหนึ่ง (Decrement) โดยอัตโนมัติ หรือใช้ในการเปรียบเทียบค่าของตัวแปรทั้งสองสิ่งสำคัญในการทำงานทางสถาปัตยกรรมของ MCS-51 คือ ความสามารถในการทำงาน สำหรับข้อมูลขนาด 8 บิต และ 1 บิต การใช้งานในระดับบิต ในการเซต เคลียร์ หรือกลับค่า การเคลื่อนย้าย การทดสอบ และใช้ในการคำนวณทางตรรกขนาด 1 บิต ความสามารถเช่นนี้เหมาะสำหรับใช้ในงานควบคุมของสัญญาณเข้าและออกที่มีการคิด และออกแบบทางตรรกด้วยพีชคณิต Boolean ซึ่งโดยปกติทำได้ลำบากสำหรับ ไมโครโพรเซสเซอร์ต่างๆ ไป งานในลักษณะเช่นนี้จึงได้ชื่ออีกอย่างหนึ่งว่า ตัวประมวลผลบูลีน (Boolean Processor)

### 3.1.4 แอควิวมิวเลเตอร์ (Accumulator : ACC)

MCS-51 ก็เช่นเดียวกับ MCS-48 ที่มีขนาด 8 บิตเป็นแอควิวมิวเลเตอร์หลักคำสั่งส่วนใหญ่จะอ้างถึงตัวเรจิสเตอร์นี้ โดยถือค่าภายในเป็นค่าตัวตั้ง และรับค่าผลลัพธ์ที่ได้จากคำสั่งทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร เข้ามาเก็บไว้ ตัว ACC ยังสามารถใช้เป็นตัวแห่งกระทำหรือถูกกระทำในการทำงานทางตรรก และใช้เป็นตัวกลางในการถ่ายเทข้อมูลในการติดต่อกับอุปกรณ์ภายนอกไอโอ และหน่วยความจำภายนอก รวมถึงการตรวจสอบตารางข้อมูล

### 3.1.5 เรจิสเตอร์ B

เป็นเรจิสเตอร์พิเศษ ที่ใช้งานสำหรับคำสั่งของการคูณ และหารโดยใช้เป็นที่เก็บตัวคูณหรือตัวหาร และเป็นที่เก็บผลลัพธ์ตัวที่สองหลังการคูณ และเศษหลังการหาร



รูปที่ 3.3 สถาปัตยกรรม MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.6 ตัวจับเวลา/ตัวนับ (Timer/Counter)

MCS-51 มี 16 บิต ตัวจับเวลา/ตัวนับ 2 ตัว คือ Time/Counter 0 และ Time/Counter 1 ส่วน 8032/8052 มีเพิ่มอีก 1 ชุด คือ Time/Counter 2 ขณะที่แต่ละตัวจับเวลา/ตัวนับ (Time/counter) สามารถที่จะกำหนดให้ทำงานได้เป็นตัวจับเวลาหรือตัวนับ

### 3.1.7 ตัวจับเวลา/ตัวนับ 0 และตัวจับเวลา/ตัวนับ 1

แต่ละตัวจะถูกกำหนดให้ทำงานเป็นตัวจับเวลาหรือเป็นตัวนับ ได้ด้วยการเซตหรือเคลียร์บิตที่ตัวควบในเรจิสเตอร์ TMOD ในกลุ่ม SFR

ในการเลือกทำงานระหว่างตัวนับกับตัวจับเวลา จะเลือกได้ 4 โหมด คือ โหมด 0 , 1 และ 2 เลือกได้ทั้งสองตัวของ Timer/Counter ส่วนโหมด 3 จะทำงานแตกต่างออกไป

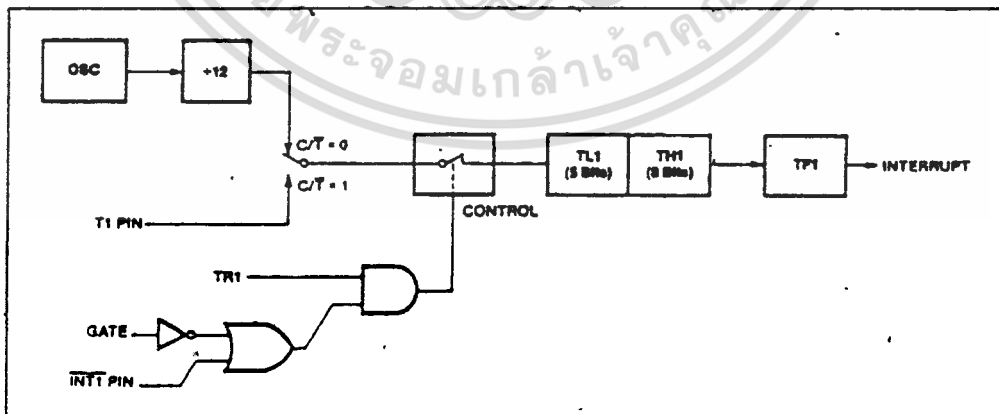
#### โหมด 0

การใช้ตัวจับเวลา/ตัวนับ 0 หรือ 1ให้อยู่ในโหมด 0 จะทำงานคล้ายกับของ MCS-48 โดยตัวจับเวลาของ MCS-48 มีขนาด 8 บิต มีตัว Prescaler เป็นตัวหาร 12 รูปที่ 3.4 แสดงการทำงานในโหมด 0 ของตัวจับเวลา/ตัวนับ 1

ในโหมดนี้ เรจิสเตอร์ตัวจับเวลาถูกกำหนดให้มี 13 บิต ด้วยการนับขึ้น เมื่อเป็น "1" หมดทุกบิต จะกลับไปที่ "0" ทุกบิตใหม่ เมื่อกลับเป็น "0" ทุกบิต จะเป็นการเกิด Overflow ไปทดให้แฟลกอินเตอร์รัพต์ TF1 ปรับเป็น "1" การควบคุมให้เริ่มนับตัวอินพุตจะควบคุมด้วยการอีนาเบิ้ล  $TR1 = 1, GATE = 0$  และ  $INT1 = 1$  การปรับ  $GATE = 1$  เป็นการตั้งตัวนับให้ถูกควบคุมด้วยสัญญาณจากภายนอกเข้า  $INT1$   $TR1$  จะเป็นบิตควบคุมในเรจิสเตอร์ TMOD ของ FSR

เรจิสเตอร์ตัวนับจะมี 13 บิต ประกอบด้วย TH1 8 บิต และ TL1 อีก 5 บิต อันดับต่ำส่วนอีก 3 บิตที่เหลือในอันดับสูงของ TL1 จะไม่ใช้ การปรับแฟลก  $TR1$  ให้ทำงานจะไม่เคลียร์ค่าในเรจิสเตอร์

การทำงานในโหมด 0 ในตัวจับเวลา/ตัวนับ 0 จะทำงานเหมือนกับตัวจับเวลา/ตัวนับ 1 โดยใช้  $TRO$  และ  $INT0$  รวมกันควบคุมแทนสัญญาณต่างๆ ในรูปที่ 3.4 มีความแตกต่างในการควบคุม คือ บิตของ  $GATE$  ทั้งสอง ตัวหนึ่งจะแทนตัวจับเวลา/ตัวนับ 1 (TMOD.7) และอีกตัวจะแทน ตัวจับเวลา/ตัวนับ 0 (TMOD.3)



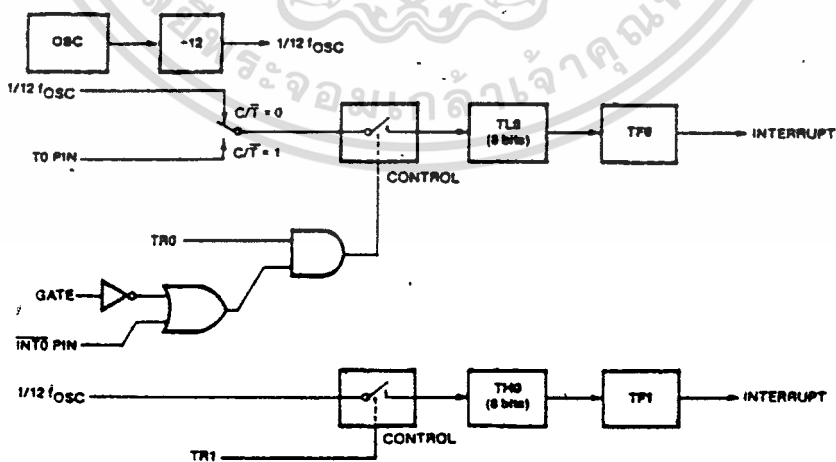
รูป 3.4 แสดงการทำงานในโหมด 0 ของตัวจับเวลา/ตัวนับ 1 ขนาด 13 บิต เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### โหมด 3

ใช้ตัวจับเวลา/ตัวนับ 1 ในโหมด 3 มีการทำงานเป็นตัวนับ มีผลเช่นเดียวกับการตั้ง  $TR1 = 0$  และใช้ตัวจับเวลา/ตัวนับ 0 ในโหมด 3 จัดการให้ TLO และ THO เป็นตัวนับสองตัวแรกที่แยกออกจากกัน วงจรตรรกควบคุมสำหรับโหมด 3 ที่ใช้ตัวจับเวลา/ตัวนับ 0 แสดงในรูปที่ 3.6 TLO ใช้ตัวจับเวลา/ตัวนับ 0 เป็นบิตควบคุมของ C/T, GATE, TRO, INTO และ TFO ตัว THO ถูกบล็อกให้ทำงานในฟังก์ชันรูปที่ 3.6 ใช้ตัวจับเวลา/ตัวนับ 1 ในโหมด 3 เป็นกลุ่มตัวนับขนาด 8 บิต สองตัว

โหมด 3 จะใช้งานในความต้องการตัวจับเวลา/ตัวนับ ขนาด 8 บิต เพิ่มขึ้นด้วยการใช้ตัวจับเวลา 0 ในโหมด 3 ดังนั้น 8051 สามารถที่จะทำงานใช้ตัวจับเวลา /ตัวนับได้เป็น 3 ชุดขณะที่ 8052 ก็ใช้งานได้ 4 ชุด เมื่อใช้ตัวจับเวลา 0 อยู่ในโหมด 3 ตัวจับเวลา 1 สามารถที่จะเปิด-ปิดให้เข้าสู่หรือออกจากการทำงานของโหมด 3 หรือสามารถที่จะยังคงใช้ตัว GENERATOR ที่สร้างอัตราบิตของการส่งข้อมูลอนุกรม หรือการใช้งานใดๆ ที่ไม่ต้องการการอินเทอร์รัปต์



รูปที่ 3.6 ใช้ตัวจับเวลา/ตัวนับ 1 ในโหมด 3 เป็นกลุ่มตัวนับขนาด 8 บิต สองตัว เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



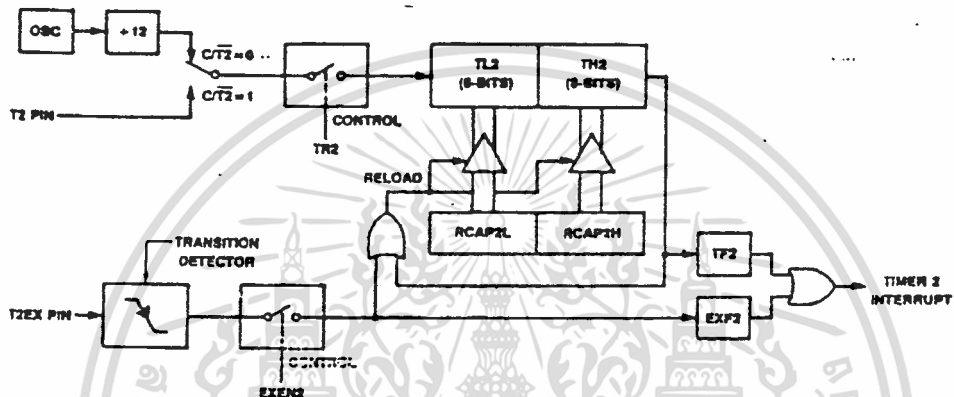
### 3.1.8 ตัวจับเวลา/ตัวนับ2

ตัวจับเวลา/ตัวนับ2 เป็นตัวจับเวลาและตัวนับขนาด 16 บิต แบบบรรจุเข้าโดยอัตโนมัติและใช้เรจิสเตอร์ควบคุมของ SFR เป็น T2CON

เมื่อตัวจับเวลา/ตัวนับ2 ทำงานเป็นตัวจับเวลา เรจิสเตอร์ตัวจับเวลา/ตัวนับ2 จะเพิ่มค่าทุกครั้งในแต่ละวัฏจักรแมซิน ขณะที่ใช้เป็นตัวนับ ตัวจับเวลา/ตัวนับ2 จะเพิ่มค่าขึ้นอยู่กับการเปลี่ยนสถานะจาก "1" -> "0" ที่เข้ามาที่ T2 (p1.0) สัญญาณอินพุตถูก Sampling ที่ S3P2 ของทุกวัฏจักรแมซินในการใช้ฟังก์ชันให้แก็ททิฟ เพราะฉะนั้นเมื่อ Sample แสดงสถานะสูง ในช่วงวัฏจักรแมซินหนึ่งและระดับต่ำในอีกวัฏจักรหนึ่ง การนับจะเพิ่มขึ้นหนึ่ง ค่าตัวใหม่จะปรากฏที่เรจิสเตอร์ตัวจับเวลา/ตัวนับ2 ช่วง S3P1 ของแต่ละวัฏจักร เป็นการนับหนึ่งที่สัญญาณการเปลี่ยนแปลง ถูกกระตุ้น ดังนั้น อัตราการนับสูงสุดจะเป็น 1/24 ของความถี่ออสซิลเลเตอร์ไม่มีการกำหนดความกว้างของพัลส์ของสัญญาณที่เข้ามาจากภายนอกแต่ต้องแน่ใจว่า ระดับหนึ่งที่ถูก Sample อย่างน้อยจะต้องคลุมหนึ่งวัฏจักรแมซินก่อนที่มันจะเปลี่ยนระดับใหม่

โหมดการทำงานของตัวจับเวลา/ตัวนับ2 มีสามโหมดด้วยกันคือ แบบธรรมดา CAPTURE แบบบรรจุใหม่อัตโนมัติ และเป็นตัวสร้างอัตราบิต ซึ่งสามารถจะเลือกการทำงานโดยเซตค่าต่างๆ ใน T2CON

วิธีตั้งให้ทำงานในโหมด CAPTURE จะมีโอกาสให้เลือกการทำงาน 2 อย่างด้วยการเลือกตั้งค่าที่บิต EXEN ของ T2CON โดยตั้ง EXEN2 = 0 ก็จะเป็นการเลือกตัวจับเวลา/ตัวนับ2 เป็นตัวจับเวลาหรือตัวนับขนาด 16 บิต ซึ่งเมื่อเกิด Overflow ก็จะไปเซตค่าใน TF2 ซึ่งเป็น Overflow บิต ของตัวจับเวลา2 ที่สามารถสร้างให้มีการใช้อินเตอร์รัพท์ที่เกิดจากตัวจับเวลาตัวนี้ได้



รูปที่ 3.7 เป็นการทำงานในโหมดของตัวจับเวลา/ตัวนับ 2

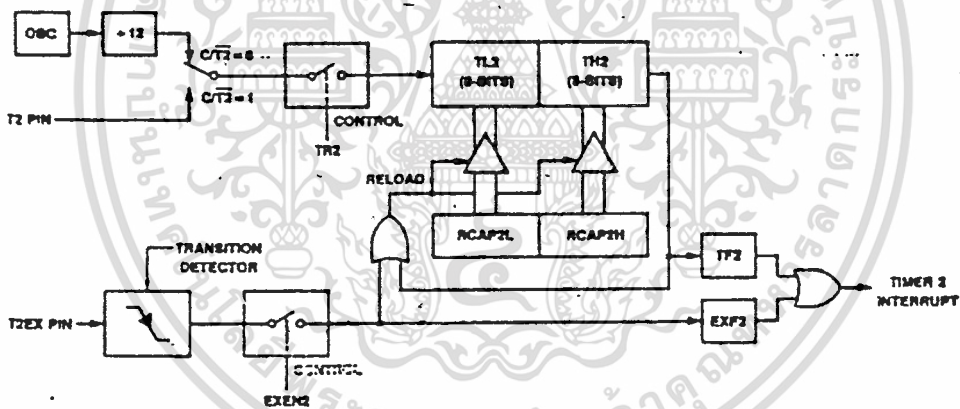
แต่ถ้า EXEN2 = 1 ตัวจับเวลา/ตัวนับยังคงทำงานเช่นเดิมเพียงแต่เพิ่มการทำงานการตรวจสอบสัญญาณจากภายนอกที่เข้ามาที่ขา T2 EX ในสถานะที่เกิดการเปลี่ยนแปลงจาก 1 เป็น 0 จะเป็นเหตุให้มีการย้ายค่าข้อมูลใน TL2 และ TH2 ในขณะนั้นเข้าสู่เรจิสเตอร์ RCAP2L และ RCAP2H ตามลำดับ โดยที่เรจิสเตอร์ทั้งสองจัดอยู่ในกลุ่ม SFR ที่มีแต่เฉพาะในไมโครเบอร์ 8032/8052 เท่านั้นในกรณีเปลี่ยนแปลงสถานะที่ n T2EX เช่นนี้ จะทำให้บิต T2EX ในเรจิสเตอร์ T2CON เขตค่าเป็นหนึ่งและ EXF2 ก็จะเป็นเช่นเดียวกับบิต TF2 คือสามารถนำไปใช้เป็นตัวกระตุ้นให้เกิดการอินเตอร์รัพท์ จากเหตุการณ์ที่เกิดขึ้นเช่นนี้ได้ ซึ่งลักษณะการทำงานในโหมด CAPTURE มีแสดงในรูป 3.7

วิธีตั้งให้ทำงานในโหมดบรรจุใหม่อัตโนมัติก็มีให้เลือกทำงานได้ 2 อย่างเช่นกัน ด้วยการปรับค่าในบิต EXEN ของ T2CON โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ถ้าปรับ  $EXEN = 0$  เมื่ออาร์กตามเวลาที่ตัวจับเวลา 2 เพิ่มค่าถึง 0 ไม่เพียงแต่จะเซตบิต TF2 เป็น 1 เท่านั้น แต่ตัวจับเวลา 2 จะถูกรบรจุค่าที่ตั้งไว้ใน RCAP2L และ PACP2H ด้วยซอฟต์แวร์แต่แรกกลับเข้าไปใหม่

2. ถ้า  $EXEM2 = 1$  ก็ทำงานเช่นเดียวกับในวิธีที่ 1 และถ้าเกิดการเปลี่ยนแปลงสถานะ จาก 1 เป็น 0 ที่ T2EX จากภายนอก ก็จะเป็นเหตุให้มีการบรจุค่าจาก RCAP2L และ RCAP2H กลับเข้าไปใหม่โดยอัตโนมัติ พร้อมทั้งเซตค่าบิตใน EXF2 ของ เรจิสเตอร์ T2CON ด้วยแผนภูมิการทำงานในโหมดบรจุอัตโนมัติอยู่ในรูป 3.8



รูป 3.8 แผนภูมิการทำงานในโหมดบรจุอัตโนมัติ

### 3.1.9 Timer/Counter Control และเรจิสเตอร์ Status

การกำหนดโหมดการทำงานและความคุมฟังก์ชันต่างๆ ของตัวจับเวลา/ตัวนับจะควบคุมได้ที่ SFR : Special Function Register TMOD, TCON และ T2CON (สำหรับ 8032/8052) ด้วยซอฟต์แวร์ โดยที่เมื่อมีคำสั่งเปลี่ยนค่าบิตต่างๆ ใน TMOD, TCON หรือ T2CON ค่าที่ถูกเปลี่ยนก็จะถูกแลตซ์เข้าไปที่ SFR และเกิดมีผลตามคำสั่งควบคุมในช่วง S1P1 ของวัฏจักรตัวแรกของคำสั่งต่อมา

### 3.1.10 การต่อเชื่อมแบบอนุกรม

พอร์ตอนุกรมเป็นแบบ Full Duplex สามารถที่จะส่งและรับพร้อมกันได้โดยทำหน้าที่เป็นบัฟเฟอร์การรับ หมายถึง พอร์ตสามารถที่จะรับไบต์ที่สอง ก่อนที่ตัวแรกจะถูกรับไปจากเรจิสเตอร์ตัวรับ อย่างไรก็ตาม ไบต์ตัวแรกจะต้องถูกอ่านไปก่อนที่ช่วงเวลาการรับไบต์ตัวที่สองจะสิ้นสุด มิฉะนั้นไบต์ตัวแรก จะถูกซ้อนและสูญหายไปได้ ในพอร์ตอนุกรมเรจิสเตอร์ตัวรับและส่ง จะเข้าถึงติดต่อกันด้วยเรจิสเตอร์ SBUF ใน SFR แม้ว่าทางโครงสร้างเรจิสเตอร์ทั้งสองจะแยกกันอยู่ก็ตาม

### 3.1.11 การอินเตอร์รัพต์

โดยทั่วไปความสามารถในควบคุมการอินเตอร์รัพต์ เป็นการทำงานชนิดหนึ่งของซีพียู ที่จะต้องศึกษาถึงความสามารถและเทคนิคการทำงาน การอินเตอร์รัพต์ของซิงเกิ้ลชิปมีความสัมพันธ์กันอย่างใกล้ชิดระหว่างอุปกรณ์ต่อพ่วงกับระบบ การทำงานของอุปกรณ์ต่อพ่วงเหล่านี้ มีระบบฮาร์ดแวร์ ที่ช่วยให้การส่งสัญญาณ Real Time กับมอนิเตอร์ได้อย่างต่อเนื่อง โดยปราศจากการรบกวนต่อสัญญาณการทำงานของซีพียู ตัวอย่างเช่น ขณะที่มีการรับสัญญาณแอนุกรมจากซีอาร์ทีตัวหนึ่ง ก็จะมีการส่งสัญญาณไปยังอุปกรณ์ตัวอื่น และตัวจับเวลา/ตัวนับ ก็จะนับพัลส์การเปลี่ยนแปลงที่เข้ามาอย่างรวดเร็วไปพร้อมกันนำตัดด้วย ในขณะที่ตัวจับเวลา/ตัวนับอีกตัวก็กำลังวัดความกว้างของพัลส์ที่เข้ามา

ซีพียูตัวนี้ จะรู้ได้อย่างไรว่าเมื่อไรถึงจะมีการรับและส่งสัญญาณแอนุกรมซีอาร์ทีหรือให้ตัวจับเวลา/ตัวนับ มีการนับจำนวนและวัดความกว้างของพัลส์ว่าจะสิ้นสุดลงเมื่อไร

ตัวโปรแกรม MCS-51 สามารถที่จะเลือกการโปรแกรมได้ 3 วิธีด้วยกัน คือ นิยามการโปรแกรมตัวเรจิสเตอร์ TCON และ SCON ที่ประกอบด้วยสถานะบิตที่ถูกเซตทางเซตทางฮาร์ดแวร์เมื่อตัวจับเวลาตัวหนึ่งเกิด Overflow หรือเมื่อการรับส่งข้อมูลที่พอร์ตอนุกรมสิ้นสุดลง

เทคนิคการโปรแกรมวิธีแรกก็โดยการอ่านสถานะของเรจิสเตอร์ควบคุมเข้าไปยังแอกคิวมิวเลเตอร์ แล้วทดสอบสถานะบิตตามลักษณะการทำงานนั้นๆ แล้วทำการกระโดดไปยังโปรแกรมย่อยตามผลที่เกิดขึ้นชนิดนั้น ๆ ลักษณะการทดสอบร่วมกันครั้งละหลายลักษณะงานเช่นนี้เปรียบเสมือนตัวโปรแกรมใช้ระบบไมโครโพรเซสเซอร์หลายตัวควบคุมชิปอุปกรณ์ต่อพ่วงต่างๆ ซึ่งผู้โปรแกรมจะต้องทำความเข้าใจอย่างลึกซึ้งถึงระบบ และจังหวะที่จะเกิดในแต่ละงาน และ การทดสอบแต่ละครั้ง จะใช้คำสั่งไม่น้อยกว่า 3 คำสั่ง

วิธีที่สอง MCS-51 สามารถที่จะทำงานด้วยการกระโดด ไปตามสถานะของการควบคุม หรือสถานะบิตของเรจิสเตอร์ควบคุมงานหรือการรับสัญญาณที่เข้ามาตามขาอินพุต แต่ละบิตด้วยการใช้คำสั่งเพียงคำสั่งเดียว ใช้เวลาประมาณภายใน 8 ไมโครวินาที

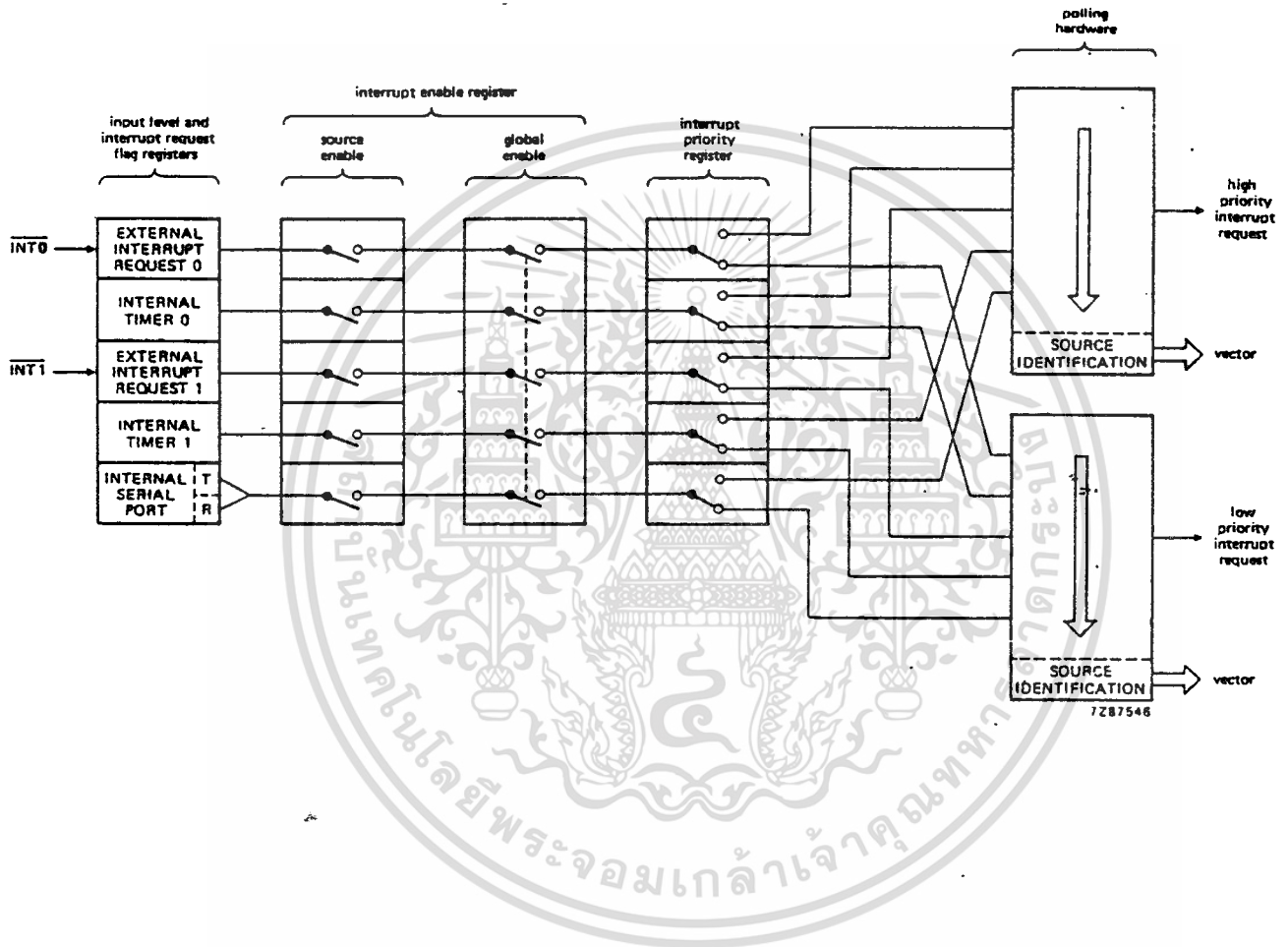
เอ็กส์เพอริเอนซ์เอ็กสาร์ทสกรีนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้วิธีที่สามจะเป็นวิธีที่ดีที่สุดในการใช้งานลักษณะนี้ด้วยการอินเทอร์รัพต์ทางอาร์แวร์

แผนภูมิทางอาร์แวร์ของการอินเทอร์รัพต์ชนิดต่างๆ ตามรูป 3.9 จะมีรายละเอียดดังนี้

แบบการอินเทอร์รัพต์จากภายนอก INTO และ INT1 สามารถจะใช้สัญญาณทั้งแบบขอบขาลงหรือระดับแรงดันต่ำขึ้นอยู่กับที่กำหนดบิต ITO และ IT1 ของเรจิสเตอร์ TCON แฟล็กที่เกี่ยวข้องกับการอินเทอร์รัพต์ภายนอกคือ IEO และ IE1 ในเรจิสเตอร์ TCON เมื่อเกิดสัญญาณอินเทอร์รัพต์จากภายนอก แฟล็กดังกล่าวก็จะถูกทำให้เป็น 1 ด้วยอาร์แวร์เป็นการติลเอเบิล การอินเทอร์รัพต์ซ้อนกันและเป็นการบอกว่า เป็นการอินเทอร์รัพต์ที่เกิดจากสัญญาณอินเทอร์รัพต์แบบขอบขาลง ถ้าเป็นแบบการอินเทอร์รัพต์ด้วยสัญญาณระดับต่ำก็จะเป็นการร้องแฟล็กให้เป็น 1 ตลอดที่สัญญาณเข้าที่ขาอินเทอร์รัพต์ยังคงเป็น 0

การอินเทอร์รัพต์ตัวจับเวลา 0 และตัวจับเวลาเวลา 1 จะทำงานได้ด้วยค่าของแฟล็ก TFO และ TF1 ที่เซตเป็น 1 เมื่อค่าในเรจิสเตอร์ของตัวจับเวลา/ตัวนับถูกเพิ่มจากหนึ่งหมกทุกบิตกลายเป็นศูนย์หมกทุกบิต (การตั้งอินเทอร์รัพต์ของตัวจับเวลา 0 จะตั้งให้ทำงานอินเทอร์รัพต์ไม่ได้ ถ้าให้ทำงานในโหมด 3) หลังจากการเกิดอินเทอร์รัพต์ตัวจับเวลาแฟล็กดังกล่าวจะถูกเคลียร์ให้เป็น 0 เมื่ออาร์แวร์บนชิปเข้าทำงานในโปรแกรมบริ-การของการอินเทอร์รัพต์ตัวจับเวลาแล้ว



รูปที่ 3.9 แผนภูมิทางอาร์แวร์ของการอินเตอร์รัทต์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.12 ลำดับความสำคัญการอินเตอร์รัพต์

แต่ละแหล่งอินเตอร์รัพต์สามารถที่จะโปรแกรมให้มีระดับไพโอริตี้สูงหรือต่ำได้ด้วยการเซตหรือเคลียร์ค่าบิตต่างๆ ใน IP ของ SFR เป็นรายละเอียดของ IP โดยที่ตัวแฟลกอินเตอร์รัพต์ทุกตัว สามารถเซตหรือเคลียร์ ได้ด้วยซอฟต์แวร์ซึ่งจะมีผลเช่นเดียวกับผลที่เกิดขึ้นจาก ฮาร์ดแวร์

แหล่งที่มาของการอินเตอร์รัพต์	ลำดับความสำคัญภายใน
การอินเตอร์รัพต์ 0 จากภายนอก (IE0)	1
การเกิด Overflow ของตัวจับเวลา/ตัวนับ 0 (TFO)	2
การอินเตอร์รัพต์ 1 จากภายนอก (IE1)	3
การเกิด Overflow ของตัวจับเวลา/ตัวนับ 1 (TF1)	4
พอร์ตอนุกรม (RI+TI)	5
การเกิด Overflow ของตัวจับเวลา/ตัวนับ 2	
การเกิด Overflow ของตัวจับเวลา 2/ขอบขาลงบน T2EX (ใช้ใน 8032/8052 เท่านั้น) (TF2+EXF2)	6

การทำงานจะเริ่มจาก Vector แอดแตรสที่ให้มาจนกระทั่งถึงคำสั่ง RETI ซึ่งอยู่ในโปรแกรมย่อย การบริการอินเตอร์รัพต์ต่าง ๆ ดังกล่าวข้างบน คำสั่ง RETI แอ็คทิวระดับความสำคัญของการอินเตอร์รัพต์ของฟิลิป-ฟลอป ที่ถูกเซตเมื่อได้รับการร้องขอการอินเตอร์รัพต์ในตอนแรก ดังนั้นมันจะ POP เอายอดของสแต็กสองไบต์มาบรรจุใหม่ใน PC เพื่อจะกลับเข้าทำงานโปรแกรมเดิมก่อนที่จะถูกอินเตอร์รัพต์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.13 การอินเทอร์รัทจากภายนอก

แหล่งกำเนิดภายนอกสามารถที่จะถูกโปรแกรมเลือกระดับการแอ็กทีฟหรือช่วงการเปลี่ยนแปลงด้วยการเซตหรือเคลียร์บิตที่ IT1 หรือ ITO ในเรจิสเตอร์ TCON ถ้า  $ITx = 0$  การอินเทอร์รัทภายนอก  $x$  จะถูกกระตุ้นด้วยการกระตุ้นระดับต่ำที่ขา  $INTx$  แต่ถ้า  $ITx = 1$  การอินเทอร์รัทภายนอก  $x$  เป็นการกระตุ้นใช้สัญญาณขาลง ในโหมดนี้ ถ้าตัวอย่างสัญญาณของขา  $INTx$  แสดงถึงระดับสูงในวัฏจักรลูกหนึ่งและต่ำในวัฏจักรอีกลูกหนึ่ง แฟล็กการร้องขออินเทอร์รัท  $IEx$  ในเรจิสเตอร์ TCON จะถูกเซตตั้งนั้น แฟล็กบิตของ  $IEx$  จะเป็นการแสดงถึงการร้องขออินเทอร์รัท

เพราะสัญญาณที่ขาการอินเทอร์รัทจะถูกลุ่มตัวอย่างหนึ่งครั้งในแต่ละวัฏจักรแมกซีนสัญญาณที่เข้าที่ต้องรักษาระดับสูงหรือต่ำอย่างน้อยภายในช่วง 12 คาบของความถี่ออสซิลเลเตอร์เพื่อให้มั่นใจในการลุ่มตัวอย่างที่รับเข้าไปได้ค่าแน่นอน ถ้าการอินเทอร์รัทภายนอกถูกแอ็กทีฟเปลี่ยนแปลง แหล่งสัญญาณภายนอกจะต้องรักษาค่าการร้องขอสถานะสูง เป็นเวลาอย่างน้อยหนึ่งลูก และรักษาค่าสถานะต่ำอีกอย่างน้อยเป็นเวลาหนึ่งวัฏจักรเพื่อให้แน่ใจว่าการเปลี่ยนแปลงค่าจะสามารถทำให้แฟล็กการร้องขออินเทอร์รัทของ  $IEx$  จะถูกเซตค่าใน  $IEx$  จะถูกเคลียร์โดยอัตโนมัติด้วยซีพียูเมื่อโปรแกรมการบริการอินเทอร์รัทถูกเรียกมาใช้

ถ้าการอินเทอร์รัทภายนอกอยู่ในระดับการแอ็กทีฟ แหล่งภายนอกจะต้องเก็บการแอ็กทีฟการร้องขอไว้ จนกว่าสัญญาณการร้องขออินเทอร์รัทจะถูกสร้างขึ้นมาเรียบร้อยแล้ว แล้วมันจะต้องกลับมารับแอ็กทีฟการร้องขอใหม่ ก่อนที่การทำงานบริการอินเทอร์รัทเดิมจะสิ้นสุดลง หรือการอินเทอร์รัทอีกลูกหนึ่งจะถูกสร้างขึ้นมาใหม่

### 3.1.14 การจัดการหน่วยความจำ

MCS-51 แบ่งตามพื้นฐานหน่วยความจำของการกำหนดเลขที่อยู่แอดเดรส ได้เป็น 3 ส่วนที่ประกอบด้วยเนื้อที่

64 กิโลไบต์ หน่วยความจำโปรแกรม

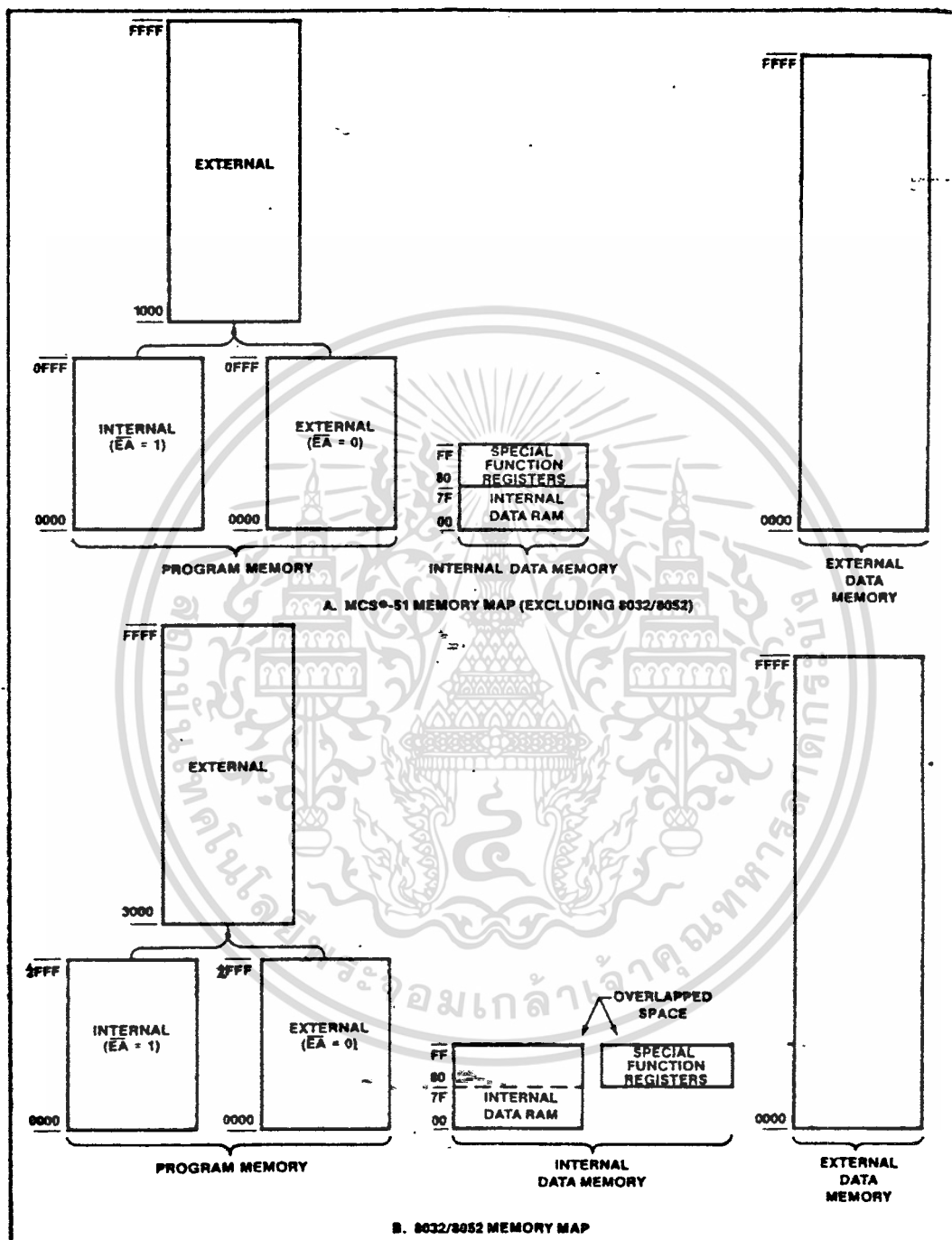
64 กิโลไบต์ หน่วยความจำข้อมูลภายนอก

256 ไบต์ เป็นหน่วยความจำข้อมูลภายใน ส่วน 8032/8052 มีขนาด 384 ไบต์

### 3.1.15 เนื้อที่หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมจะประกอบด้วย ส่วนภายในและภายนอกชิป ถ้าขา EA มีสถานะสูง MCS-51 จะบริการโปรแกรมภายใน ถ้าโปรแกรมมีความยาวไม่เกิน 0FFF (4K) หรือ 1 FFF (8K) สำหรับตัว 8052 ตำแหน่งตั้งแต่ 1000 H ถึง 0FFFFH (หรือ 2000 H 0FFFFH สำหรับ 8052) จะเป็นการเฟรชข้อมูลโปรแกรมภายนอก ถ้าขา EA มีสถานะต่ำ MCS-51 จะเฟรชข้อมูลโปรแกรมภายนอกทั้งหมด ในทุกกรณีตัวนับโปรแกรมขนาด 16 บิต จะเป็นตัวกำหนดเลขที่อยู่โปรแกรม

ตำแหน่ง 00 ถึง 32 H (หรือ 00 ถึง 2BH สำหรับเบอร์ 8032/8052) ในหน่วยความจำโปรแกรมจะสำรองไว้ สำหรับใช้บริการการอินเตอร์รัพต์



รูปที่ 3.10 แสดงแผนภูมิพื้นที่ของหน่วยความจำใน MCS-51

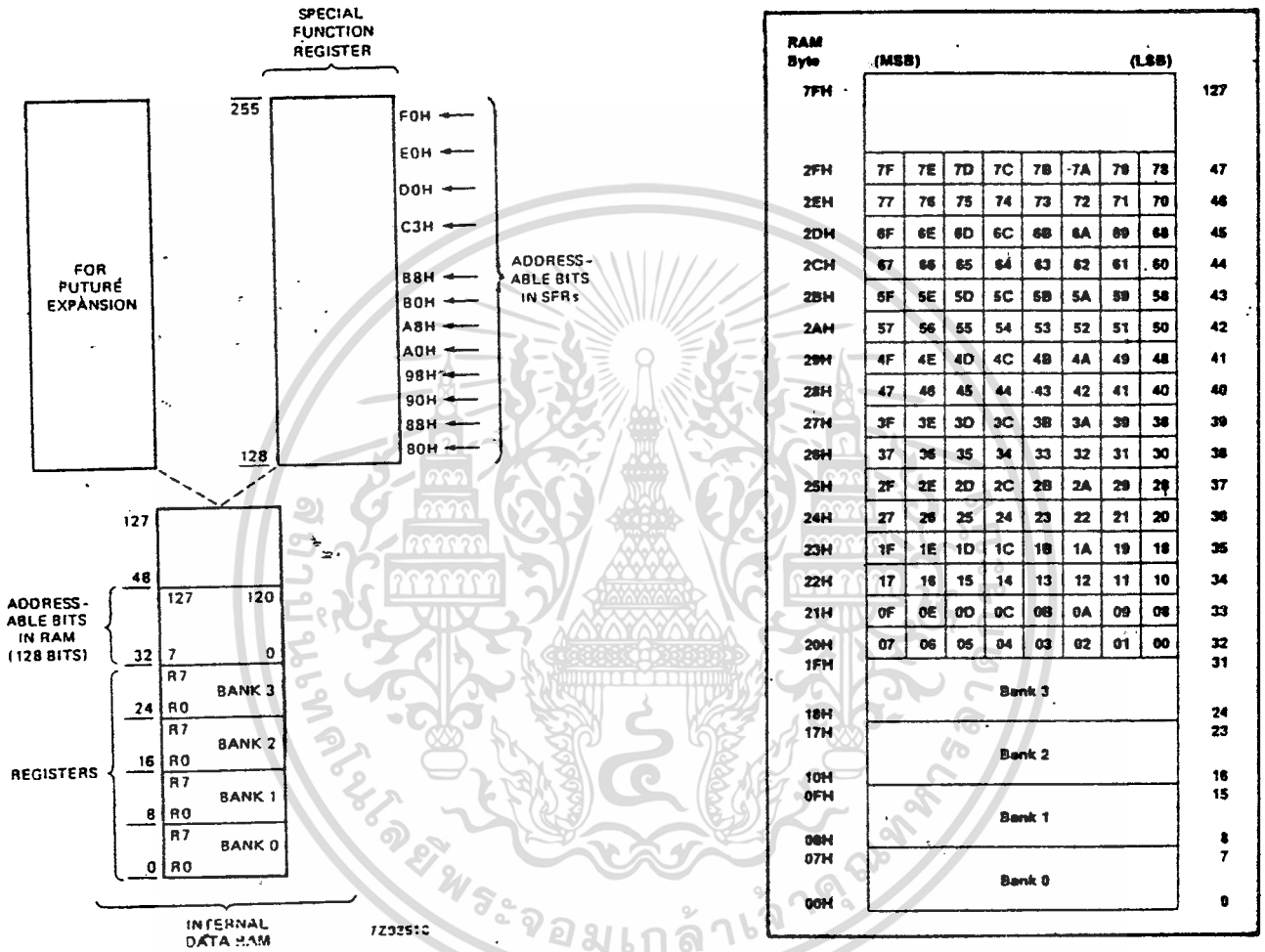
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.16 เนื้อที่หน่วยความจำข้อมูล

หน่วยความจำข้อมูลจะประกอบด้วยความจำข้อมูลภายในและภายนอก หน่วยความจำข้อมูลภายนอก จะเข้าถึงได้ด้วยการใช้คำสั่ง MOVX

หน่วยความจำข้อมูลภายในจะแบ่งเป็นลักษณะงานดังนี้คือ จำนวน 128 ไบต์ ของบริเวณตำแหน่งล่างในเนื้อที่แรมภายใน และอีก 128 ไบต์ เป็นของบริเวณตำแหน่งบนของแรมภายใน ส่วนบนนี้จะมีเฉพาะในเบอร์ 8032/8052 เท่านั้น และส่วนของ 128 ไบต์ อีกบริเวณหนึ่ง ใช้เป็นเรจิสเตอร์ฟังก์ชันพิเศษ ขณะที่ใช้ส่วนบนของแรมภายใน และบริเวณของ SFR ทั้งสองส่วนนี้จะถูกปันส่วนให้ใช้ค่าแอดเดรสภายใน แต่จะเข้าถึงในแต่ละบริเวณของทั้งสองบริเวณนี้ได้ ด้วยการให้โหมกการกำหนดเลขที่อยู่ที่แตกต่างกัน ซึ่งโหมกเหล่านี้จะอธิบายในหัวข้อต่อไป

รูปที่ 3.11 แสดงถึงแผนที่ของหน่วยความจำข้อมูล โดยแบ่งเป็น 4 แบนด์ ในแต่ละแบนด์มีเตจิสเตอร์ 8 ตัว มีตำแหน่งตั้งแต่ 0 ถึง 31 ในบริเวณล่างของแรม แบนด์เหล่านี้ จะถูกเรียกใช้ให้อินาเบิลได้คราวละหนึ่งแบนด์เท่านั้น ด้วยการกำหนดเริ่มแรกภายในสองบิตของเรจิสเตอร์ PSWว่าจะเลือกใช้ในแบนด์ใดภายใน 4 แบนด์ และบริเวณตำแหน่งตั้งแต่ 20H ถึง 2FH จำนวน 16 ตำแหน่งๆ ละ 1 ไบต์ สามารถที่จะกำหนดเลขที่อยู่ของแต่ละบิตได้ ดังแสดงในรูปที่ 3.11B เป็นบิตแรมแอดเดรส เนื้อที่เรจิสเตอร์ SFR สามารถที่จะกำหนดตำแหน่งได้เช่นกัน ดังรูป 3.12



- A) แสดงถึงแผนที่ของหน่วยความจำข้อมูล      B) แสดงถึงแผนที่การกำหนดตำแหน่งบิต

รูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Direct Byte Address	Bit Addresses								Hardware Register Symbol
	(MSB)				(LSB)				
240	F7	F6	F5	F4	F3	F2	F1	F0	B
224	E7	E6	E5	E4	E3	E2	E1	E0	ACC
208	CY	AC	FO	RS1	RS0	OV		P	PSW
184				PS	PT1	PX1	PT0	PX0	IP
176	B7	B6	B5	B4	B3	B2	B1	B0	P3
168	EA			ES	ET1	EX1	ET0	EX0	IE
160	AF			AC	AB	AA	A9	A8	P2
152	A7	A6	A5	A4	A3	A2	A1	A0	P1
144	SMO	SM1	SM2	REN	TB8	RB8	T1	RI	SCON
136	9F	9E	9D	9C	9B	9A	99	98	P0
128	87	86	85	84	83	82	81	80	P0

รูปที่ 3.12 ตำแหน่งของเรจิสเตอร์ SFR และบิตแอดเดรสของ SFR

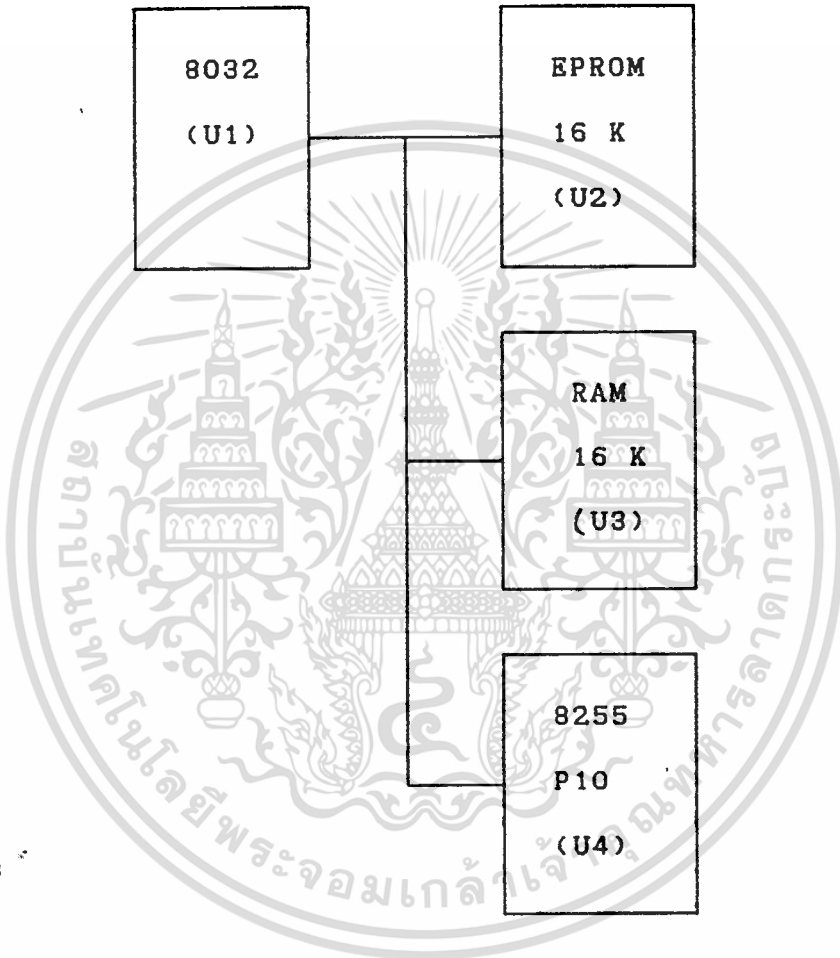
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 หน่วยอินพุต-เอาต์พุต (INPUT-OUTPUT UNIT)

ในการควบคุมกระบวนการต่างๆ โดยใช้พีแอลซีนั้น จำเป็นต้องมีการติดต่อกันระหว่างอุปกรณ์ภายนอกกับพีแอลซี ผ่านหน่วยอินพุต-เอาต์พุต สำหรับหน่วยอินพุต-เอาต์พุตที่ออกแบบเป็นหน่วยอินพุตภายนอก 16 อินพุต โดยต่อร่วมกับออปโตคัปเปอเรอร์ (OPTO COUPLER) และหน่วยเอาต์พุตภายนอก 16 เอาต์พุต โดยต่อร่วมกับทรานซิสเตอร์แบบชาคอลเล็คเตอร์เปิด (OPEN COLLECTOR TRANSISTOR) 8 เอาต์พุต ต่อร่วมกับรีเลย์ (RELAY) 8 เอาต์พุต

ทั้งออปโตคัปเปอเรอร์ของอินพุต ทรานซิสเตอร์แบบชาคอลเล็คเตอร์เปิดและรีเลย์ของเอาต์พุต ต่างก็ติดต่อกับหน่วยประมวลผลกลางผ่านทางไอซีเบอร์ 8255 (U5) ซึ่งเป็นพอร์ตแบบขนาน (PARALLEL PORT) รับส่งข้อมูลครั้งละ 8 บิต

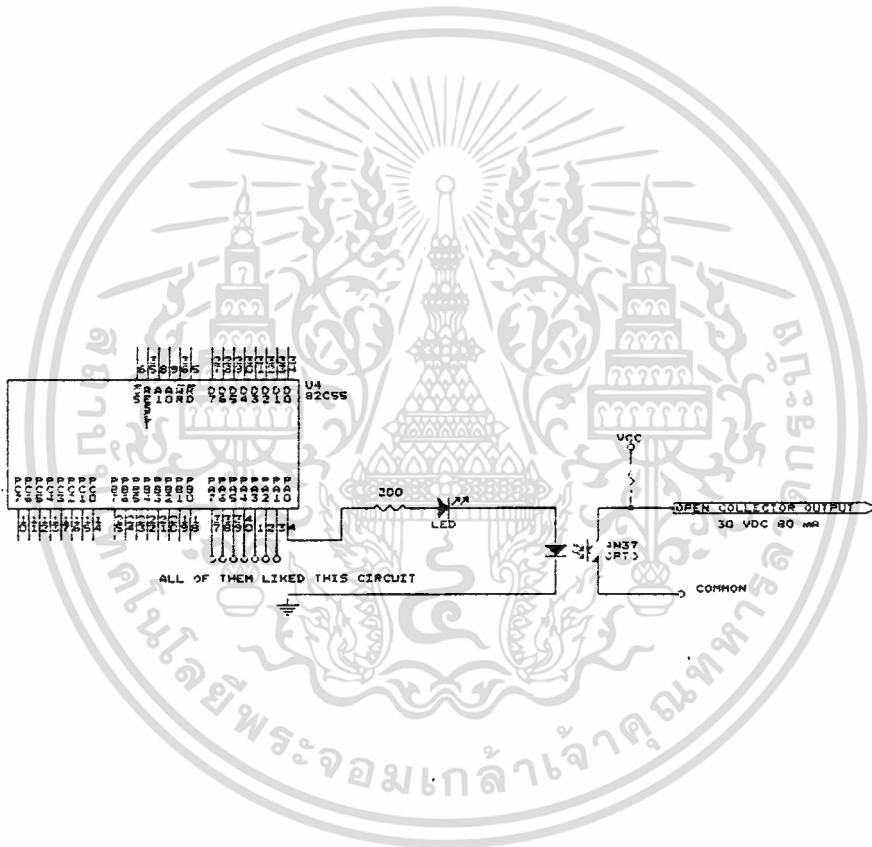
ไอซีเบอร์ 8255 สามารถเป็นได้ทั้งอินพุตพอร์ต โดยภายในแบ่งเป็น 3 พอร์ต คือ พอร์ต 0 พอร์ต 1 และพอร์ต 2 ตามลำดับ การกำหนดให้พอร์ตใดเป็นอินพุตหรือเอาต์พุต สามารถทำได้โดยใช้คำสั่งควบคุมส่งมายังพอร์ตควบคุมของไอซีเบอร์ 8255



รูปที่ 3.13 โครงสร้างทางฮาร์ดแวร์ของพีแอลซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

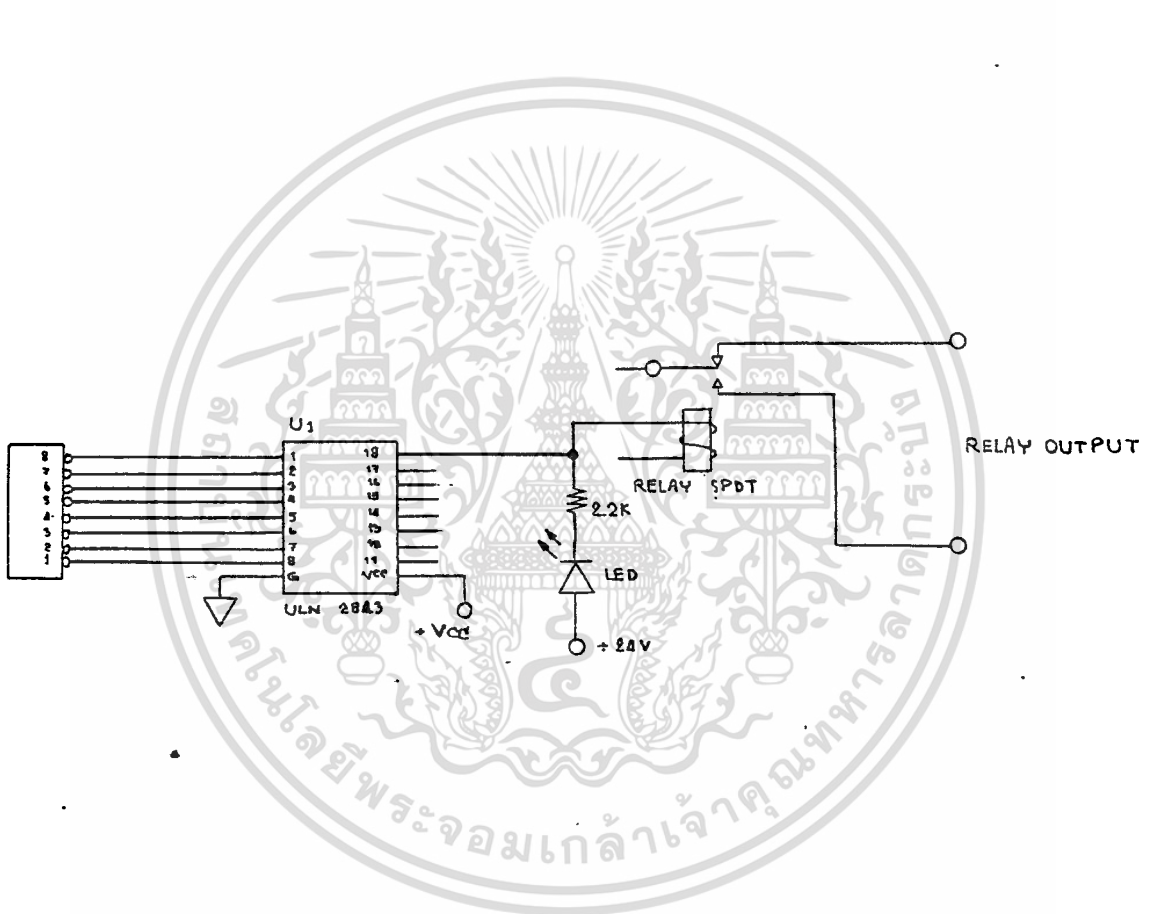
สำหรับ 8255 (U4) ที่ใช้สำหรับหน่วยอินพุต-เอาต์พุต เรากำหนดพอร์ท A เป็นเอาต์พุตพอร์ท ติดต่อกับออปโตทรานซิสเตอร์แบบชาคอลเล็กเตอร์เปิดเบอร์ 4N26 จำนวน 8 ชุด (รูปที่ 3.14) เป็นเอาต์พุตแลกรานซิสเตอร์แบบชาคอลเล็กเตอร์เปิด 8 เอาต์พุต



รูปที่ 3.14 วงจรเอาต์พุตทรานซิสเตอร์แบบชาคอลเล็กเตอร์เปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับพอร์ท B กำหนดเป็นเอาต์พุตพอร์ท เช่นเดียวกันโดย จะติดต่อกับไอซี ULN 2803 ซึ่งทำหน้าที่ขับกระแสให้กับกระแส 8 ตัวเป็นเอาต์พุตแบบ รีเลย์ 8 เอาต์พุต (รูปที่ 3.15)



รูปที่ 3.15 แสดงวงจรเอาต์พุตแบบรีเลย์

สำหรับพอร์ท กำหนด 4 บิตล่าง (PC0-PC3) เป็นอินพุตพอร์ท (รูปที่ 3.16) โดยต่อร่วมกับไอซีเบอร์ 74150 ซึ่งเป็นไอซีมัลติเพล็กซ์ และ PC7 เป็น สายอินพุตรับสัญญาณ เข้าสู่หน่วยประมวลผลกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 3.3 หน่วยป้อนโปรแกรม

เป็นหน่วยติดต่อสื่อสารระหว่างผู้ใช้กับเครื่องควบคุม เพื่อการป้อนชุดคำสั่งเป็นโปรแกรมควบคุม แก๊สโปรแกรม ตลอดจนตรวจสอบสภาพการทำงานของเครื่อง

หน่วยป้อนโปรแกรมนี้จัดแบ่งออกเป็น 4 ส่วนคือ

1. สวิตช์เลือกโหมดการทำงาน
2. คีย์บอร์ดสำหรับเขียน ตรวจสอบ และแก๊สโปรแกรม
3. จอภาพแสดงผลแบบ LCD
4. ลำโพง

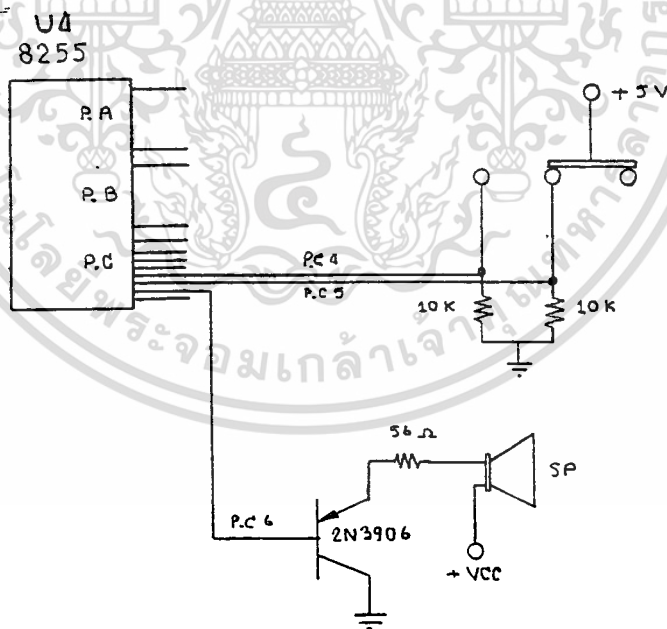
#### 3.3.1 สวิตช์เลือกโหมดการทำงาน

เป็นสวิตช์ที่ทำหน้าที่เลือกโหมดการทำงานของพีแอลซี ซึ่งมีอยู่ 2 โหมดโปรแกรม (PROGRAM MODE) โหมดทำงานจริง (RUN MODE)

สวิตช์เลื่อน 2 จัหวะต่อเข้ากับ 8255 (U4) โดยต่อเข้าดังรูปที่ 3.17 จากวงจรในรูปสามารถสรุป "ข้อมูล" ที่เกิดขึ้นเมื่อเลือกโหมดต่างๆกันได้คือ

MODE	PC4	PC5
PROGRAM	1	1
RUN	0	1

แสดงค่าสภาวะประจำโหมดต่างๆ



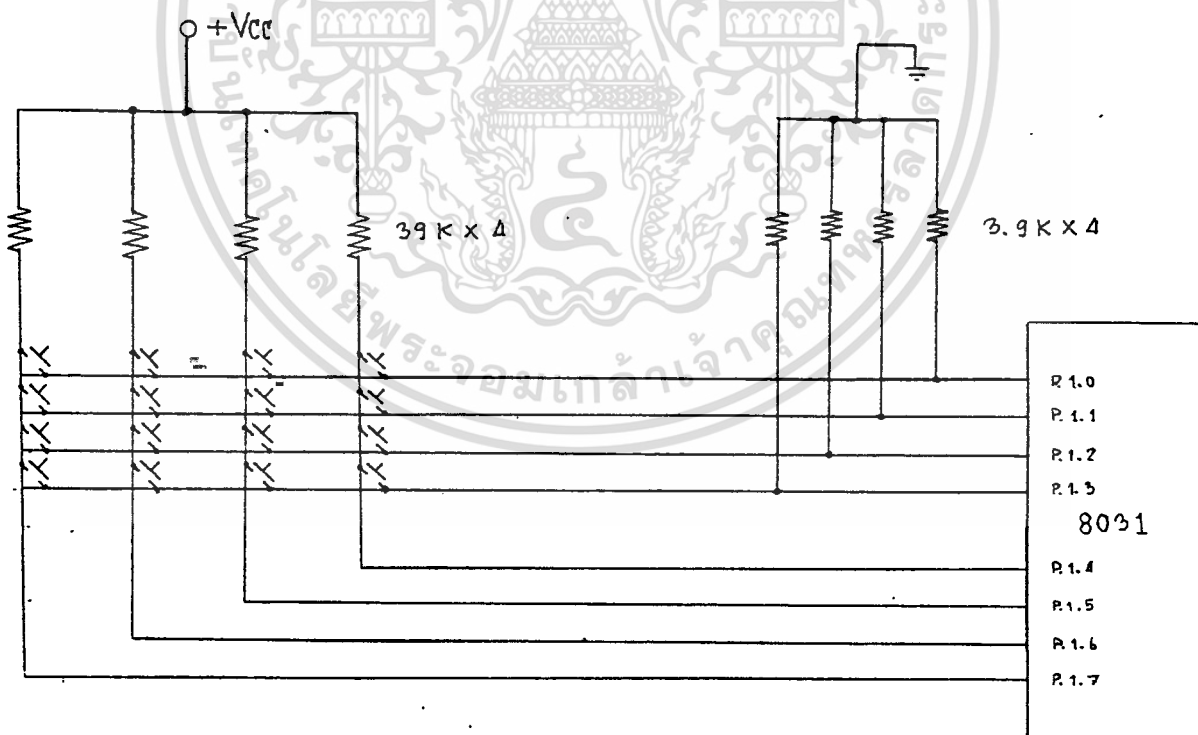
รูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 คีย์บอร์ด

คีย์บอร์ดที่ใช้เป็นเมมบริคีย์บอร์ดที่เรียกว่า เมมเบรนสวิตช์ (MEMBRANE SWITCH) ขนาด 4 X 4 (16 คีย์) ดังนั้นการเชื่อมต่อคีย์บอร์ดเข้ากับหน่วยประมวลผลกลางจึงอาศัยหลักการสแกนคีย์ โดยต่อพอร์ท 1 ของ 8032 โดยมี P1.4-P1.7 เป็นข้อมูล สแกนเอาท์ และ P1.0-P1.3 เป็นข้อมูล สแกนอิน ตามรูปที่ 3.18

จะเห็นว่าจำนวนคีย์บอร์ดมีอยู่เพียง 16 คีย์ ดังนั้นคำสั่งพิเศษบางคำสั่งจะถูกเรียกใช้งานในรูปแบบของฟังก์ชันนัมเบอร์ (FUNCTION NUMBER) คือกดคีย์ FUN แล้วตามด้วยตัวเลขต่างๆ






รูปที่ 3.18 แสดงการต่อคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 INS	2 EQU	1 NOT	SFT
7 RES	6 REG	5 IN	4 LD
RUN 7 SET	0 TIM	9 CNT	8 OUT
FUN DEL	DEC AND	INC OR	CLR ENT

รูปที่ 3.19 ภาพแสดงตำแหน่งของคีย์ต่างๆ (Keyboard Map)

FUNTION	INSTRUCTION
1	T 
2	T 
3	T 

**ตารางแสดงค่าฟังก์ชันพิเศษต่างๆ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.3 จอภาพแสดงผลแบบ LCD

เป็นส่วนแสดงผลเพื่อติดต่อสื่อสารกับผู้ใช้เป็น Dot Matrix LCD ขนาด 2 บรรทัด บรรทัดละ 16 ตัวอักษร (ดังมีรายละเอียดการนำมาใช้งานที่จะกล่าวถึงต่อไป) โดยสามารถแสดงผลได้เป็นตัวอักษร ง่ายแก่การอ่านและการเข้าใจ ใน Program - Mode ทำหน้าที่แสดงลำดับคำสั่งและคำสั่งที่ผู้ใช้ตรวจสอบแก้ไขคำสั่งต่างๆได้ โดยรับผลการแก้ไขจากคีย์บอร์ด แล้วมาแสดงผลที่จอภาพ ใน RUN Mode ก็จะทำหน้าที่แสดงสถานะของ Input, Output, Timer, Counter ตลอดจน Internal Relay เพื่อตรวจสอบการทำงานแต่ไม่สามารถ set value ได้

### 3.3.4 ลำโพง

เป็นอุปกรณ์ช่วยในการโปรแกรม ถ้าหากมีการผิดพลาดในการพีซีเครื่องจะมีเสียงทอบสนองออกมา เพื่อให้ผู้ใช้รับรู้และแก้ไขได้

ในการเชื่อมต่อเข้ากับหน่วยประมวลผลกลางสามารถทำได้โดยรับสัญญาณจาก PC 6 ของ 8255 (U5) ดังรูปที่ 3.17

### 3.4 หน่วยจ่ายพลังงาน

หน่วยจ่ายพลังงาน ที่ใช้แบ่งออกเป็น 2 ส่วน คือ

1. แหล่งจ่ายแรงดัน 24 โวลต์
2. แหล่งจ่ายแรงดัน 5 โวลต์

### 3.4.1 หม้อแปลง

หม้อแปลงที่ใช้ใน ส่วนของขดลวดทุติยภูมิแยกเป็น 2 ขด แปลงแรงดันจาก ขดลวดปฐมภูมิซึ่งมีขนาด 220 โวลต์ ให้มีระดับแรงดันเป็น 10 โวลต์ 3 แอมป์ และ 28 โวลต์ 2 แอมป์ ตามลำดับ

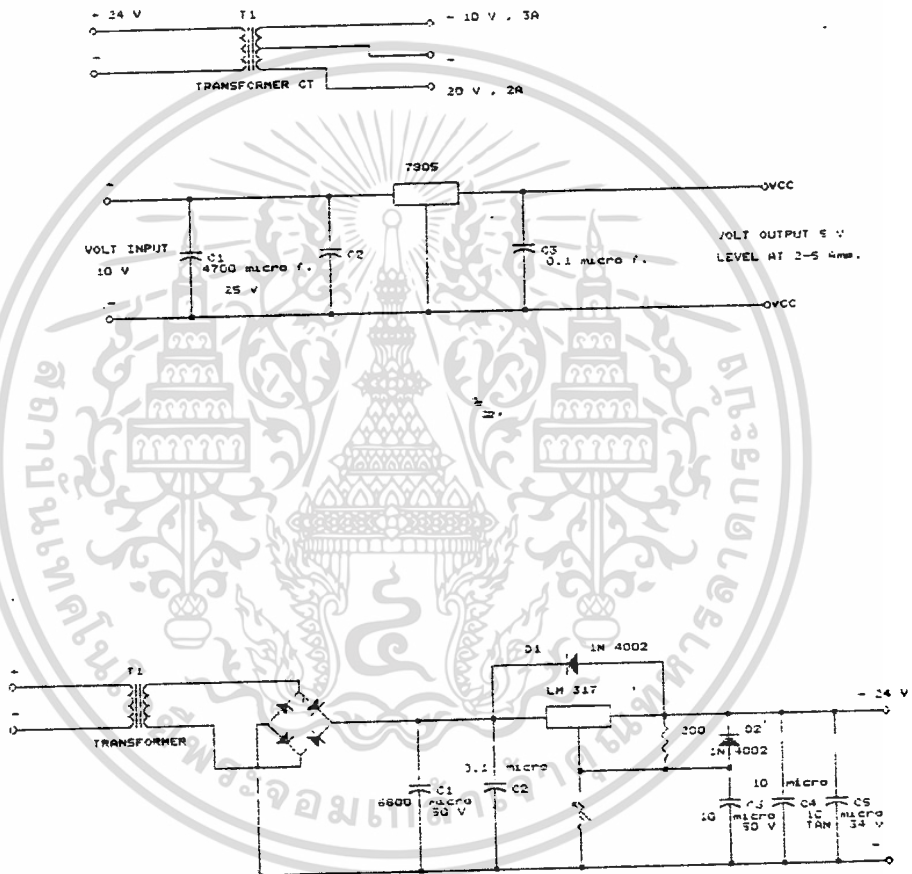
### 3.4.2 วงจรเรกกูเลเตอร์ (Regulator)

สำหรับวงจรจ่ายแรงดันขนาด 5 โวลต์ ใช้ไอซีเรกกูเลเตอร์เบอร์ 7805 ซึ่งมีความสามารถในการจ่ายกระแสได้ 1.5 แอมป์ แต่เนื่องจากเรามีความต้องการความสามารถในการจ่ายกระแสเอาท์พุทที่สูงกว่านั้น เราจึงขอแบบวงจรเพิ่มทรานซิสเตอร์เบอร์ MJ 2955 ต่อขนานเข้าไปกับไอซีเรกกูเลเตอร์เบอร์ 7805 เพื่อเป็นตัวขับกระแสให้มีขนาดสูง

ส่วนวงจรจ่ายแรงดันขนาด 24 โวลต์ ใช้ไอซีเรกกูเลเตอร์เบอร์ LM317 ต่อวงจรดังรูปที่ 3.20

### 3.4.3 วงจรป้องกันกระแสกระชาก

ส่วนนี้ได้รับการออกแบบเพิ่มเติม เข้าไปป้องกันกระแสกระชาก ก่อนที่จะเข้าสู่หม้อแปลง ซึ่งผลของกระแสกระชากนี้อาจทำให้เกิดการรบกวนการทำงานของหน่วยประมวลผลกลางได้ ทำให้การควบคุมผิดพลาด ลักษณะวงจรได้แสดงไว้แล้วในรูปที่ 3.20



รูปที่ 3.20 วงจรหน่วยจ่ายพลังงานขนาด 5 โวลต์ และ 24 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การออกแบบโปรแกรมบริหารระบบ PLC

พีแอลซีที่ออกแบบเป็นเครื่องมือที่สามารถเปลี่ยนแปลงแก้ไข เงื่อนไขในการควบคุมได้ โดยผู้ใช้สามารถเขียนโปรแกรมผู้ใช้ได้ ดังรายละเอียดที่ได้กล่าวมาแล้วนอกจากนี้ พีแอลซีมีโปรแกรมผู้ใช้อยู่แล้วจะต้องสามารถ ควบคุมการทำงานกับอุปกรณ์ภายนอกให้ได้ ตามโปรแกรมที่เขียนมา อีกทั้งในระหว่างที่กำลังควบคุมการทำงานของกระบวนการอยู่นั้น ผู้ใช้สามารถมอดิไฟเออร์ค่าสถานะต่างๆ ได้

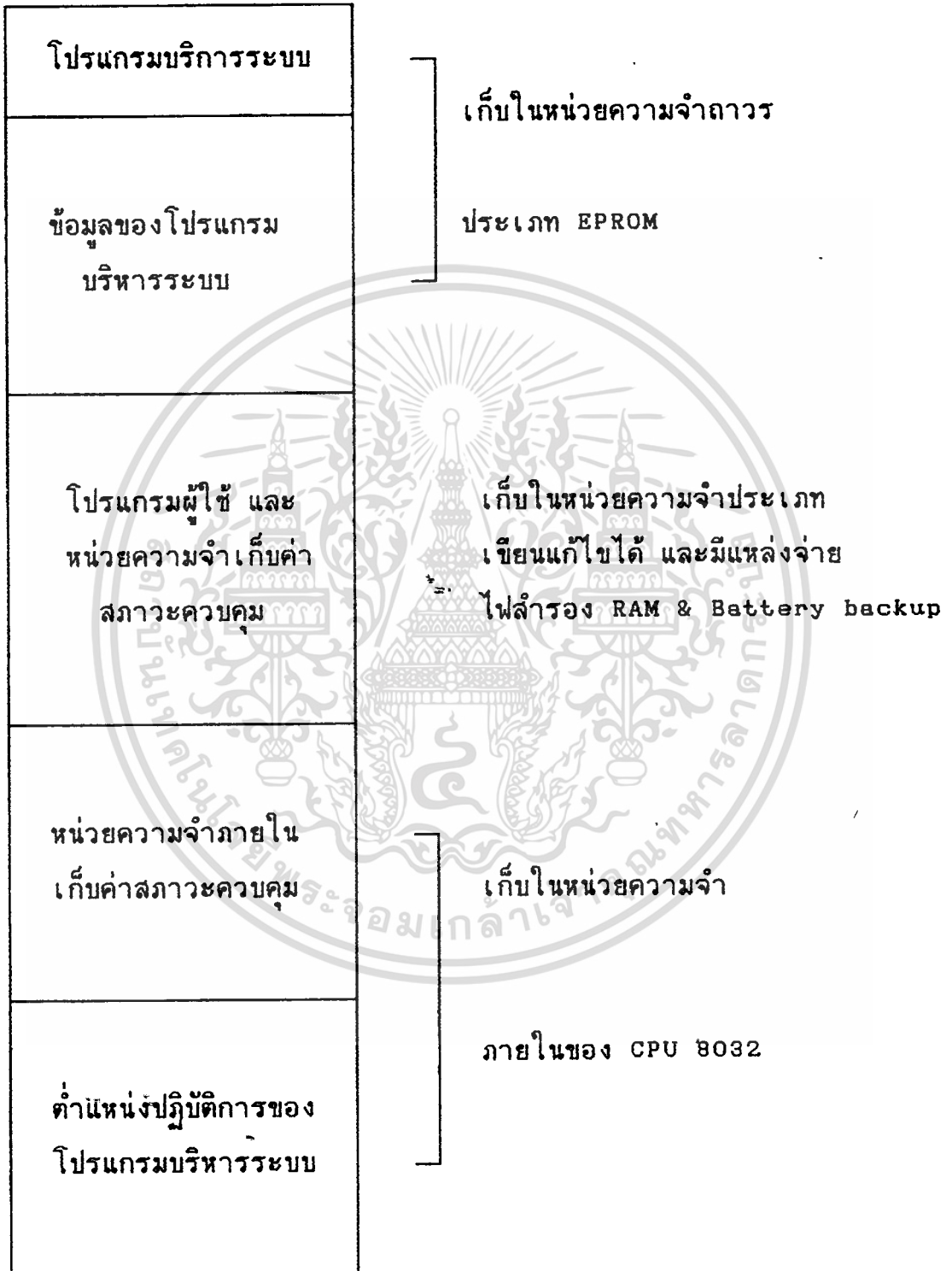
ทั้งหมดที่กล่าวมาแล้วข้างต้นนั้น พีแอลซี ทำได้อย่างไร คำถามนี้เองนำไปสู่ แนวความคิดในการพัฒนา "โปรแกรมบริหารระบบพีแอลซี" ดังกล่าวดังต่อไปนี้

### 4.1 โปรแกรมบริหารระบบทำหน้าที่อะไรบ้าง

เราสามารถแบ่งแยกการทำงานของโปรแกรมบริหารระบบ ออกเป็นส่วนๆ เพื่อสร้างแนวความคิดในการพัฒนาต่อไป ได้ดังนี้

1. จักการเกี่ยวกับการป้อนโปรแกรมของผู้ใช้
2. ตรวจสอบโปรแกรมผู้ใช้
3. จัดการให้โปรแกรมผู้ใช้สามารถทำงานได้ตามฟังก์ชันที่ออกแบบไว้

ก่อนที่จะกล่าวถึงรายละเอียดในการออกแบบโปรแกรม ขอทำความเข้าใจเกี่ยวกับการจัดการหน่วยความจำของพีแอลซี ดังต่อไปนี้



รูปที่ 4.1 การจัดหน่วยความจำของพีแอลซี (PLC MEMORY MAP)  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2 การจัดหน่วยความจำของพีแอลซี

จัดแบ่งหน่วยความจำออกเป็น 5 ส่วน คือ

1. โปรแกรมบริหารระบบ
2. ข้อมูลของโปรแกรมบริหารระบบ
3. โปรแกรมผู้ใช้
4. หน่วยความจำภายในเก็บค่าสถานะควบคุม
5. ตำแหน่งปฏิบัติการของโปรแกรมบริหารระบบ

##### 4.2.1 โปรแกรมบริหารระบบ

หน่วยความจำส่วนนี้อยู่ที่ตำแหน่งแรก ของหน่วยประมวลผลกลาง เป็นหัวใจหลักของการทำงานของเครื่อง เป็นส่วนที่เราพัฒนาขึ้นมา ทำหน้าที่บริหาร ควบคุมการทำงาน โปรแกรมผู้ใช้ อีกทีหนึ่ง

##### 4.2.2 ข้อมูลของโปรแกรมบริหารระบบ

เป็นข้อมูลที่โปรแกรมบริหารระบบ อ่านไปเพื่อปฏิบัติการต่างๆ ส่วนใหญ่แล้ว เป็นข้อมูลสำหรับการแสดงผลออกที่จอภาพแบบ แอลซีดี

##### 4.2.3 โปรแกรมผู้ใช้

หน่วยความจำส่วนนี้จัดสรร ได้ให้เก็บโปรแกรมผู้ใช้ที่เขียนขึ้น การจกเก็บ การปฏิบัติการตามโปรแกรมเป็นไปตาม การควบคุมของโปรแกรมบริหารระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.4 หน่วยความจำภายในเก็บค่าสถานะควบคุม

เก็บค่าสถานะของ โอเพอร์แรนด์ ต่างๆ ได้แก่

- สถานะ เปิด-ปิด ของอินพุท เอาท์พุท ตัวตั้งเวลา ตัวนับ
- ค่าตั้งเวลา (Timer Set Value) ค่าตั้งตัวนับ (Counter Set) ค่าปัจจุบันของตัวตั้งเวลา (Running Time), ค่าปัจจุบันของตัวนับ (Running Count)

#### 4.2.5 ตำแหน่งปฏิบัติการของโปรแกรมบริหารระบบ

เป็นกลุ่มหน่วยความจำที่เป็นที่ปฏิบัติของโปรแกรม เป็นผลแตกของโปรแกรมเป็น บัฟเฟอร์ต่างๆ ทำหน้าที่ตั้งเงื่อนไขต่างๆแก่โปรแกรม โปรแกรมบริหารระบบจะมีการ อ่านและเขียนกับหน่วยความจำส่วนนี้ตลอดเวลา

#### 4.3 การจัดการเกี่ยวกับการป้อนโปรแกรมของผู้ใช้

ในโหมดโปรแกรมการทำงานหลักของโปรแกรมบริหารระบบ คือ การจัดการ เกี่ยวกับการจัดเก็บโปรแกรมผู้ใช้ ตลอดจนการแก้ไขตรวจสอบโปรแกรมที่ผู้ใช้เขียนขึ้นแล้ว

แนวคิดในการออกแบบโปรแกรมบริหารระบบในส่วนที่ทำหน้าที่นี้ คือ

1. การตรวจสอบคีย์ที่กด (SCAN KEY AND EXECUTE)
2. วิธีการตรวจสอบกลุ่มคีย์ที่กดและจัดเก็บ โค้ดคำสั่ง ลงในหน่วยความจำ ส่วนที่เป็นโปรแกรมผู้ใช้

จะเห็นได้ว่าเราจะต้องสร้างโค๊ดมาตรฐานขึ้นมา 2 ชุด คือ

1. โค้ดสแกนคีย์ (SCANKEY CODE) รูปที่ 4.2
2. โค้ดคำสั่ง (PLC OPERATION CODE) รูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 (03H)	2 (02H)	1 (01H)	(00H)
INS (43H)	EQU (42H)	NOT (41H)	SFT (40H)
7 (13H)	6 (12H)	5 (11H)	4 (10H)
RES (53H)	REG (52H)	IN (51H)	LD (50H)
RUN (23H)	0 (22H)	9 (21H)	8 (20H)
SET (63H)	TIM (62H)	CNT (61H)	OUT (60H)
FUN (33H)	DEC (32H)	INC (31H)	CLR (30H)
DEL (73H)	AND (72H)	OR (71H)	ENT (70H)

รูปที่ 4.2 แสดงโคดสแกนคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCTION	CODE
LD IN	3 0 0 0 H
LD OUT	3 0 0 6 H
LD REG	3 0 0 C H
LD TIM	3 0 1 2 H
LD CNT	3 0 1 6 H
LD NOT IN	3 0 0 3 H
LD NOT OUT	3 0 0 9 H
LD NOT REG	3 0 0 F H
LD NOT TIM	3 0 1 5 H
LD NOT CNT	3 0 1 8 H
AND IN	3 0 1 E H
AND OUT	3 0 2 4 H
AND REG	3 0 2 A H
AND TIM	3 0 3 0 H
AND CNT	3 0 3 6 H
AND NOT IN	3 0 2 1 H
AND NOT OUT	3 0 2 7 H
AND NOT REG	3 0 2 D H
AND NOT TIM	3 0 3 3 H
AND NOT CNT	3 0 3 9 H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCTION	CODE
OR IN	3 0 3 C H
OR OUT	3 0 4 2 H
OR REG	3 0 4 8 H
OR TIM	3 0 4 E H
OR CNT	3 0 5 4 H
OR NOT IN	3 0 3 F H
OR NOT OUT	3 0 4 5 H
OR NOT REG	3 0 4 B H
OR NOT TIM	3 0 5 1 H
OR NOT CNT	3 0 5 7 H
SET CNT	3 0 6 0 H
SET TIM	3 0 7 2 H
RES CNT	3 0 6 3 H
RES REG	3 0 6 9 H
EQU OUT	3 0 5 D H
EQU REG	3 0 5 A H
EQU TIM	3 0 6 F H
EQU CNT	3 0 7 5 H

รูปที่ 4.3 แสดงโค้ดคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาษาคำสั่งของ PLC

ภาษาที่ใช้ในการเขียนโปรแกรมคำสั่งของ PLC มีอยู่หลายภาษา แต่ภาษาที่นิยมใช้มีอยู่ 2 ภาษาคือ

1. ภาษานอกแลดเดอร์ไดอะแกรม (Ladder Diagram Language)
2. ภาษาคำสั่งบูลีน (Boolean language)

### 5.1 ภาษาแลดเดอร์ไดอะแกรม

ภาษาแลดเดอร์ไดอะแกรม เป็นภาษาลัญญาที่ใช้แทนรีเลย์ไดอะแกรม (Relay Diagram) ซึ่งเป็นไดอะแกรมวงจรลอจิกที่ใช้งานจริง ดังนั้นการแปลงรีเลย์ไดอะแกรมเป็นแลดเดอร์ไดอะแกรมจึงทำได้โดยตรง และอำนวยความสะดวกแก่ผู้ใช้อย่างมากแต่ถ้าเป็นภาษาอื่นๆ การแปลงรีเลย์ไดอะแกรมให้เป็นภาษาที่พีแอลซีเข้าใจมีขั้นตอนเพิ่มขึ้นอีก 1 ขั้นตอน คือ จากรีเลย์ไดอะแกรมเป็นแลดเดอร์ไดอะแกรม จึงแปลงเป็นภาษาพีแอลซีใช้อีกครั้งหนึ่ง พีแอลซีที่ใช้ภาษาแลดเดอร์จะต้องมีหน่วยโปรแกรมที่มีจอภาพแสดงผลร่วมอยู่ด้วย เพื่อใช้แสดงรูปสัญลักษณ์ของแลดเดอร์ไดอะแกรม หน่วยโปรแกรมดังกล่าวนี้จึงมีขนาดใหญ่ และราคาค่อนข้างสูง ทั้งนี้เพราะโครงสร้างภายในค่อนข้างจะซับซ้อน

### 5.2 ความสัมพันธ์ระหว่างแลดเดอร์ไดอะแกรมและฟังก์ชันต่างๆ

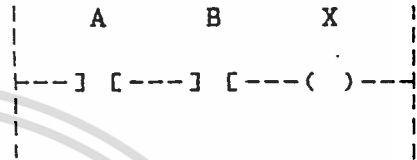
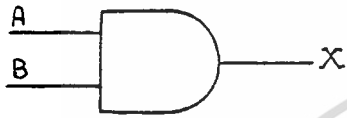
พีแอลซีจะถูกนำมาใช้แทนรีเลย์ในระบบควบคุมแบบ ON-OFF ซึ่งลักษณะของวงจรควบคุมเป็นแบบฟังก์ชันลอจิกพื้นฐาน ดังนั้นการใช้แลดเดอร์ไดอะแกรมแทนวงจรควบคุมนี้ ก็เป็นการแทนฟังก์ชันดังกล่าวนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

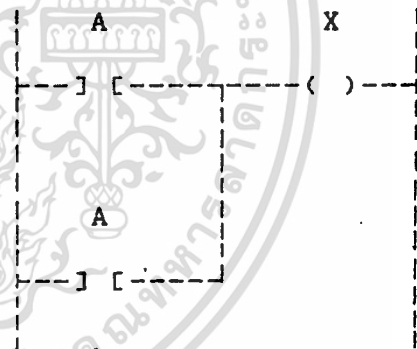
ฟังก์ชันลอจิก AND

แลตเตอร์ไคอะแกรม



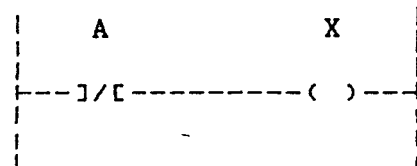
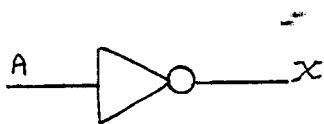
ฟังก์ชันลอจิก OR

แลตเตอร์ไคอะแกรม



ฟังก์ชันลอจิก NOT

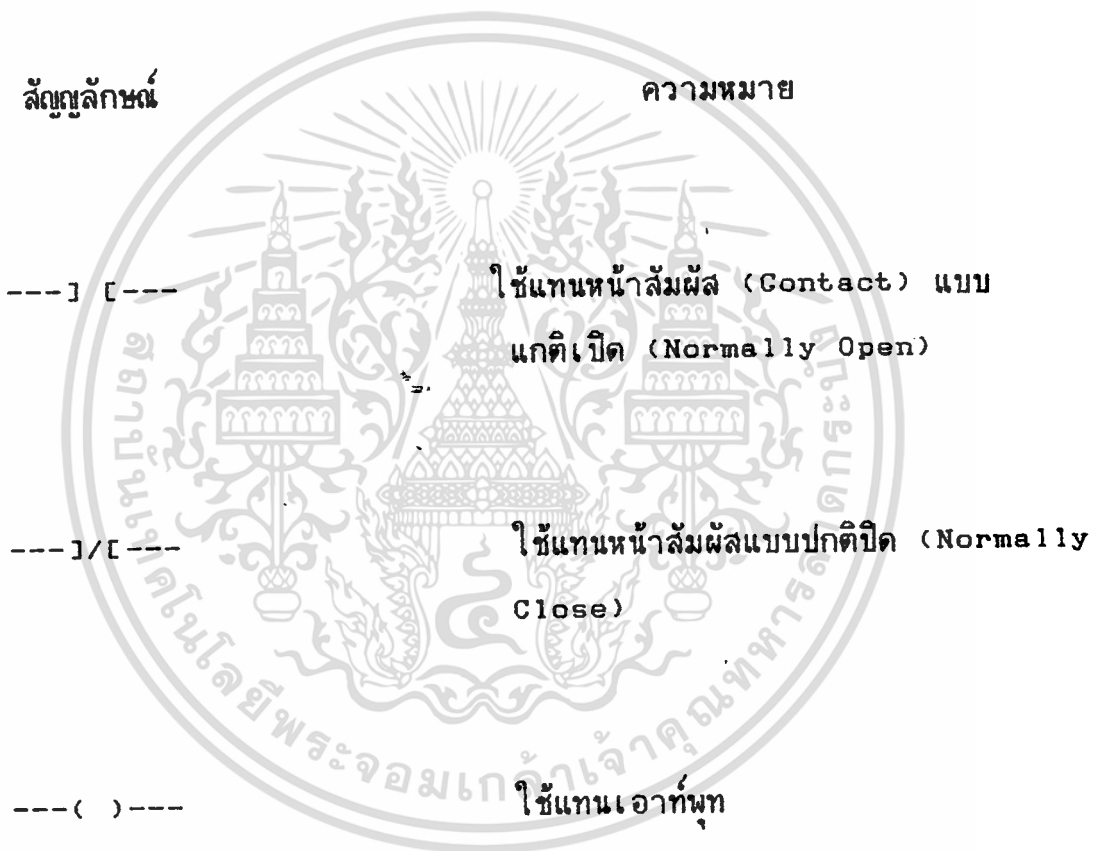
แลตเตอร์ไคอะแกรม



รูปที่ 5.1 ความสัมพันธ์ระหว่างฟังก์ชันลอจิกและแลตเตอร์ไคอะแกรม

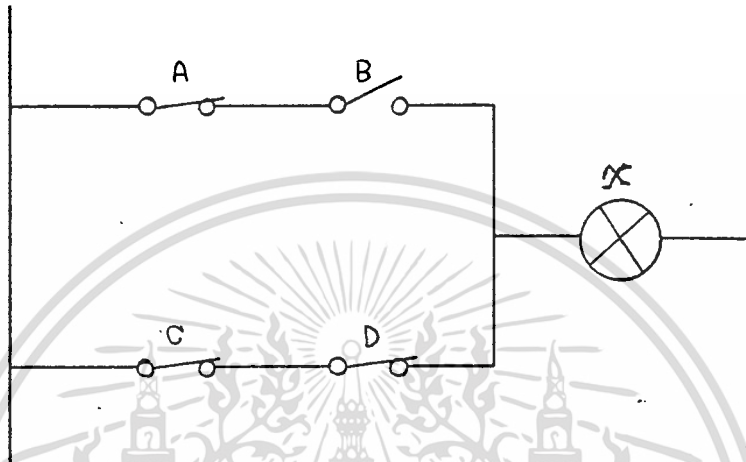
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสัมพันธ์ระหว่างแลตเตอร์ไดอะแกรมกับฟังก์ชันลอจิกข้างต้น จะพบว่า สัญญลักษณ์พื้นฐานที่ใช้ในแลตเตอร์ไดอะแกรมมีดังนี้คือ



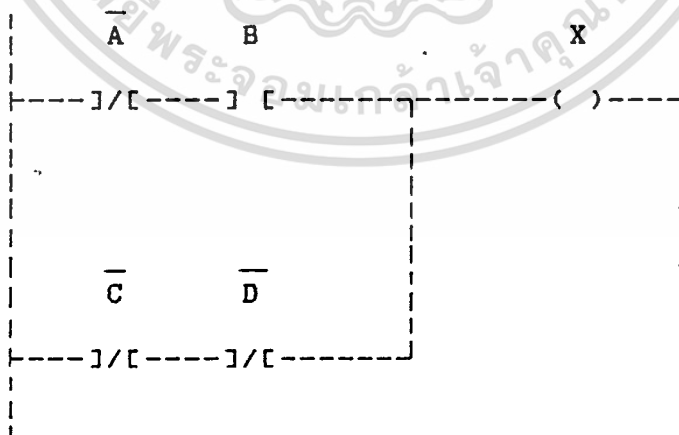
รูปที่ 5.2 สัญญลักษณ์พื้นฐานที่ใช้ในแลตเตอร์ไดอะแกรม

ต่อไปนี้ เป็นตัวอย่างที่ใช้แลตเตอร์ไดอะแกรมแทนวงจรีเลย์



รูปที่ 5.3 วงจรีเลย์

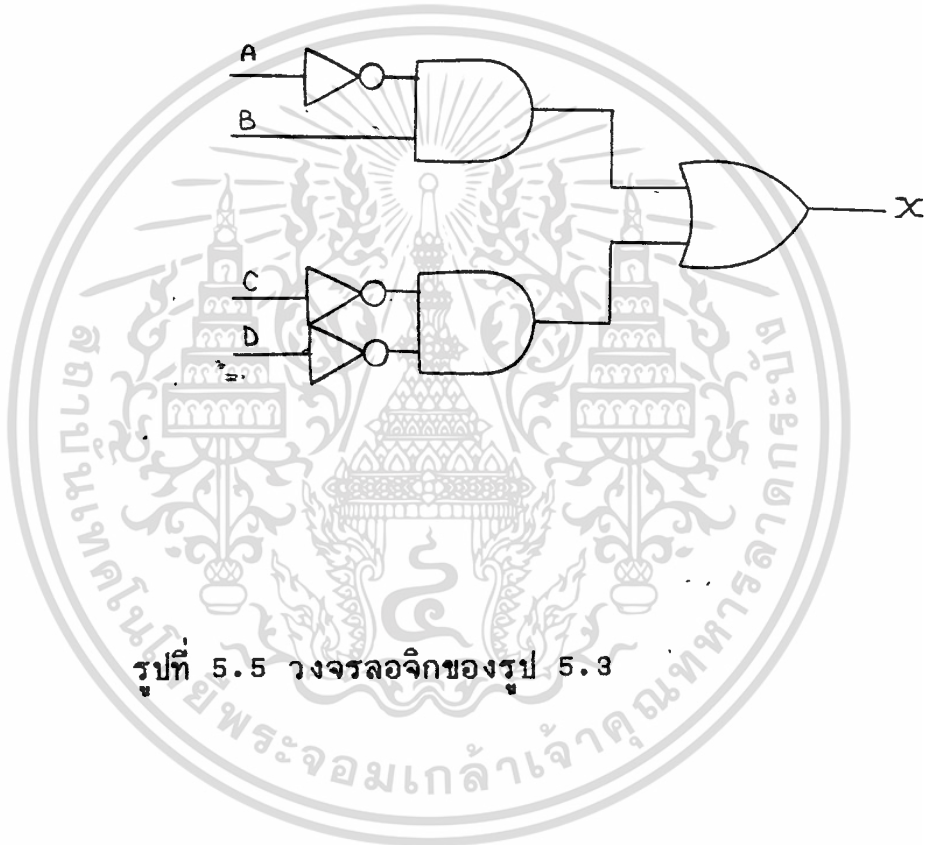
ซึ่งสามารถแทนด้วยแลตเตอร์ไดอะแกรมได้ดังนี้



รูปที่ 5.4 แลตเตอร์ไดอะแกรมของรูป 5.3

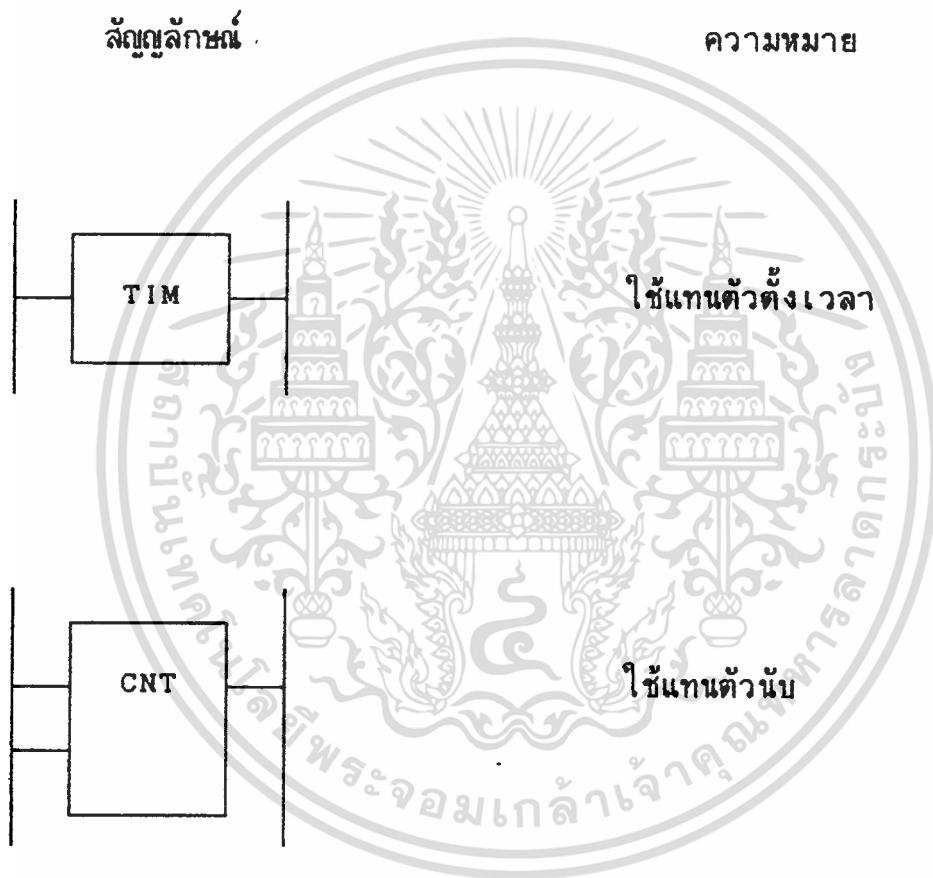
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และวงจรรีเลย์ในรูป 5.3 สามารถแทนด้วยวงจรถลอจิกได้ดังรูป 5.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากสัญลักษณ์พื้นฐานของแลคเคอร์ไคอะแกรมแล้ว ยังมีสัญลักษณ์อื่นๆ ที่ใช้แทนฟังก์ชันบางอย่างที่มีใช้ในวงจรควบคุมรีเลย์ ได้แก่ ตัวตั้งเวลา ตัวนับ เป็นต้น



รูปที่ 5.6 สัญลักษณ์ของตัวตั้งเวลา และตัวนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 ภาษาคำสั่งบูลีน

ภาษาคำสั่งบูลีนเป็นภาษาที่มีคำสั่งพื้นฐานทางลอจิก คือ แอนด์ ออร์ และน็อก และคำสั่งอื่น ๆ อีกที่สอดคล้องกับรีเลย์ไดอะแกรม ลักษณะของภาษาคำสั่งบูลีนจัดว่าเป็นภาษาที่เข้าใจได้ง่ายพอๆ กับแลดเดอร์ไดอะแกรม และเหมาะกับพีแอลซีที่มีขนาดเล็กดังนั้นปรวิญญาพินธ์ฉบับนี้จะใช้ภาษาคำสั่งบูลีนกับพีแอลซีที่ได้ออกแบบขึ้น ภาษาคำสั่งบูลีนโดยทั่วๆ ไปจะประกอบด้วยคำสั่งต่อไปนี้ คือ คำสั่ง LOAD (LD) AND OR NOT (INVERDE) OUT (=) TIMER (TIM) COUNTER (CNT) SET RESET (RST) JUMP (JMP) INTERNAL RELAY (REGISTER : REG) และอื่นๆ

## บทที่ 6

### ชุดคำสั่งใช้งาน

#### 6.1 จำนวนอินพุต เอาท์พุท ตัวตั้งเวลา และตัวนับ

##### 6.1.1 อินพุต (Input)

มีจำนวน อินพุท ทั้งสิ้น 16 อินพุท เป็นอินพุทที่ใช้งานได้จริงภายนอกพร้อมทั้งสามารถเซต-รีเซต ได้จากหน่วยป้อนโปรแกรมจำนวน 16 อินพุท

##### 6.1.2 เอาท์พุท (Output)

มีจำนวน เอาท์พุท ทั้งสิ้น 16 เอาท์พุท ซึ่งสามารถใช้เป็นเอาท์พุทจริงๆ ภายนอกได้ 16 เอาท์พุท แยกเป็นเอาท์พุทแบบคอลเลคเตอร์เปิด จำนวน 8 เอาท์พุท

เอาท์พุททุกตัวที่เราไม่ได้ใช้ในการขับโหลดภายนอก เราย่อมสามารถใช้เอาท์พุทเหล่านั้นเป็น รีเลย์ภายใน เพื่อช่วยในการเขียนโปรแกรมได้

### 6.1.3 ตัวตั้งเวลา (Timer)

มีตัวตั้งเวลาให้ใช้งานทั้งสิ้น 16 ช่อง ในการทำงานภายใต้การตั้งเวลาในโหมดต่างๆ ทั้ง 3 โหมด หมายความว่า ถ้ามีการเรียกใช้ตัวตั้งเวลาช่องใดช่องหนึ่งมาใช้แล้ว ไม่ว่าจะถูกเรียกใช้ในโหมดใด เราก็จะไม่สามารถเรียกใช้ตัวตั้งเวลาช่องนั้นได้อีกฐานเวลาสำหรับตัวตั้งเวลาของพีแอลซีเรื่องนี้ คือ 0.1 วินาทีและค่าสูงสุดของแต่ละช่องที่จะตั้งเวลาได้คือ 999.9 วินาที

### 6.1.4 ตัวนับ (Counter)

มีตัวนับให้ใช้งานทั้งสิ้น 16 ช่อง ตัวนับและตัวตั้งเวลาเป็นอิสระต่อกัน สามารถใช้เลขที่ช่องซ้ำกันได้ เช่น สามารถเรียกใช้ TIM01 ได้ พร้อมกับ CNT01

## 6.2 องค์ประกอบของโปรแกรมผู้ใช้

โปรแกรมผู้ใช้ 1 โปรแกรม ประกอบขึ้นด้วย "ชุดคำสั่ง" ตั้งแต่ 1 ชุดขึ้นไป ชุดคำสั่ง 1 ชุดคำสั่ง ประกอบขึ้นด้วย "คำสั่ง" ตั้งแต่ 1 คำสั่งขึ้นไป คำสั่ง 1 คำสั่ง ประกอบขึ้นมาจาก หลาย "คีย์คำสั่ง"

จะเห็นได้ว่า ส่วนที่ย่อยที่สุดสำหรับการเขียนโปรแกรมให้เรื่องพีแอลซี คือคีย์คำสั่ง และทุกครั้งที่เรากด คีย์ ENTER เราจะได้ คำสั่ง 1 คำสั่ง เก็บอยู่ใน หน่วยความจำ และ คำสั่งหลายๆ คำสั่งที่ปฏิบัติการกันแล้วจะให้ผลลัพธ์บางอย่างเกิดขึ้น กลุ่มคำสั่ง นั้นเอง

### 6.3 การจำแนกคีย์บอร์ด

สามารถจำแนกได้ 3 ชนิด คือ

1. คีย์สำหรับการใช้เครื่องพีแอลซี (Operating Keys)
2. คีย์คำสั่งต่างๆ ของเครื่องพีแอลซี (Instruction Keys)
3. คีย์โอเปอเรนด์ (Operand Keys)

#### 6.3.1 คีย์สำหรับการใช้เครื่องพีแอลซี

คีย์		การใช้งาน
SHIFT	[SFT]	เลือกใช้ชุดคำสั่งบน(Upper Case)
CLEAR	[CLR]	ลบโปรแกรมภายในหน่วยความจำ
INSERT	[INS]	แทรกคำสั่งใหม่
DELETE	[DEL]	ลบคำสั่ง
INCREMENT	[INC]	เลื่อนไปยัง Step ข้างหน้า
DECREMENT	[DEC]	เลื่อนไปยัง Step ข้างหลัง
ENTER	[ENT]	นำคำสั่งต่างๆ เก็บในหน่วยความจำ แสดงสภาวะการทำงานที่จอภาพ
FUNCTION	[FUN]	เลือกใช้คำสั่งพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3.2 คีย์คำสั่งต่างๆ ของเครื่องพีแอลซี

คีย์	การใช้งาน
LD (LD)	นำค่าสถานะไปเก็บในหน่วยความจำ
AND (AND)	กระทำลอจิก และ (AND)
OR (OR)	กระทำลอจิก หรือ (OR)
NOT (NOT)	กระทำลอจิก ไม่ (NOT)
OUT (OUT)	ส่งค่าสถานะไปยังเอาต์พุต
SET (SET)	ตั้งค่าให้กับ COUNTER และ TIMER
RES (RES)	ลบค่าที่ตั้งใน COUNTER และ สถานะของ COUNTER

Select function [FUN] [0]                      ตัวตั้งเวลา (มี 3 แบบ)

DELAY ON [FUN] [1]

DELAY OFF [FUN] [2]

ONE SHORT [FUN] [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3.3 คีย์โอเปอร์แรนด์

คือ คีย์ที่ใช้กำหนดอินพุท-เอาต์พุท ตัวตั้งเวลา ตัวนับ ตั้งรายละเอียดข้างล่างนี้

โอเปอร์แรนด์	คีย์	หมายเลขโอเปอร์แรนด์
INPUT	[IN]	0000 - 0015
OUTPUT	[OUT]	0600 - 0615
TIMER	[FUN] [1] (หรือ [FUN] [2]) (หรือ [FUN] [3])	0000 - 0015
COUNTTER	[CNT]	0000 - 0015
REGITER	[REG]	0000 - 0093

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.4 คำสั่งเบื้องต้น

### 1. คำสั่ง LD

คำสั่งนี้มักจะใช้เป็นคำสั่งแรกของ ขบวนการคำสั่งแต่ละชุด และจะติดตามด้วย โอเปอเรนด์ตัว แรกที่ใช้ในการประมวลผลของขบวนการคำสั่งนั้น เช่น LD IN 5

### 2. คำสั่ง AND

คำสั่งนี้แสดงถึงการต่ออนุกรมกันระหว่างผลลัพธ์ทางลอจิกของขบวนการคำสั่งหรือ โอเปอเรนด์กรอนหน้านี้ กับโอเปอเรนด์ของคำสั่งนี้

### 3. คำสั่ง OR

คำสั่งนี้แสดงถึงการต่อขนานกันระหว่างผลลัพธ์ทางลอจิก ของขบวนการคำสั่งหรือ โอเปอเรนด์ก่อนหน้านี้ กับ โอเปอเรนด์ของคำสั่งนี้

### 4. คำสั่ง EQU

คำสั่งนี้ใช้กระตุ้นให้โอเปอเรนด์ของคำสั่งกระทำฟังก์ชันตามชนิดของโอเปอเรนด์ ของคำสั่งนี้ ต่อเมื่อผลลัพธ์ทางลอจิกของขบวนการคำสั่งก่อนหน้านั้นเป็นจริง

จากคำสั่งเบื้องต้นต่างๆ ที่กล่าวมาข้างต้นนี้ การทำงานของคำสั่งจะอยู่ในลักษณะเมคฟังก์ชัน (Make Function) แต่ในการใช้งานแล้วจะมีการทำงานในลักษณะเบรคฟังก์ชัน (Brake Function) อยู่ด้วย ดังนั้นจึงมีคำสั่งที่เหมาะสมสำหรับงานในลักษณะนี้ คำสั่ง เหล่านี้ได้แก่ LD NOT AND NOT OR NOT ซึ่งจะหมายถึง LD Inverse AND Inverse และ OR Inverse ตามลำดับ

### ตัวอย่างการใช้คำสั่งเบื้องต้นต่างๆ

#### ตัวอย่างที่ 1



```
โปรแกรมคำสั่ง : LD IN 0001
                  AND IN 0002
                  EQU OUT 0001
                  END
```

ความหมาย : เอาท์พุท 0001 จะทำงาน (Active) เมื่ออินพุท 0001 และอินพุท 0002 ทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2

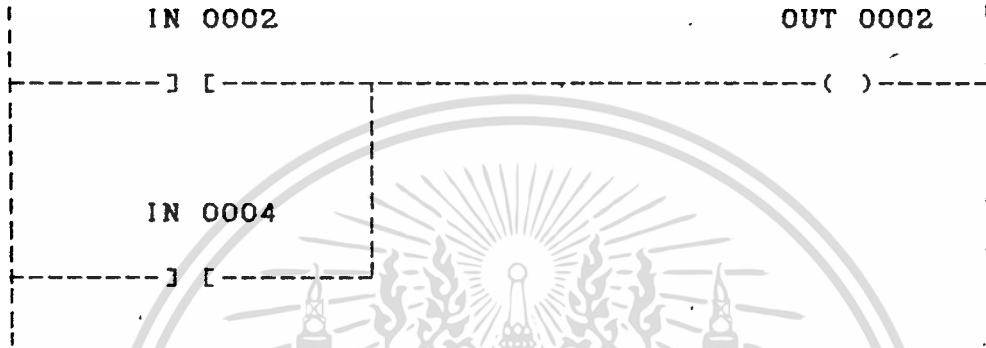


โปรแกรมคำสั่ง :

```
LD IN 0001
AND NOT IN 0003
EQU OUT 0001
END
```

ความหมาย : เอาท์พุท 0001 จะทำงาน ก็ต่อเมื่ออินพุท 0001 ทำงานและอินพุท 0003 ไม่ทำงาน

ตัวอย่างที่ 3



โปรแกรมคำสั่ง :

```
LD IN 0002
OR IN 0004
EQU OUT 0002

END
```

ความหมาย : เอาท์พุท 0002 จะทำงาน ก็ต่อเมื่ออินพุท 0002 หรือ อินพุท 0004  
ตัวใดตัวหนึ่งทำงาน

ตัวอย่างที่ 4



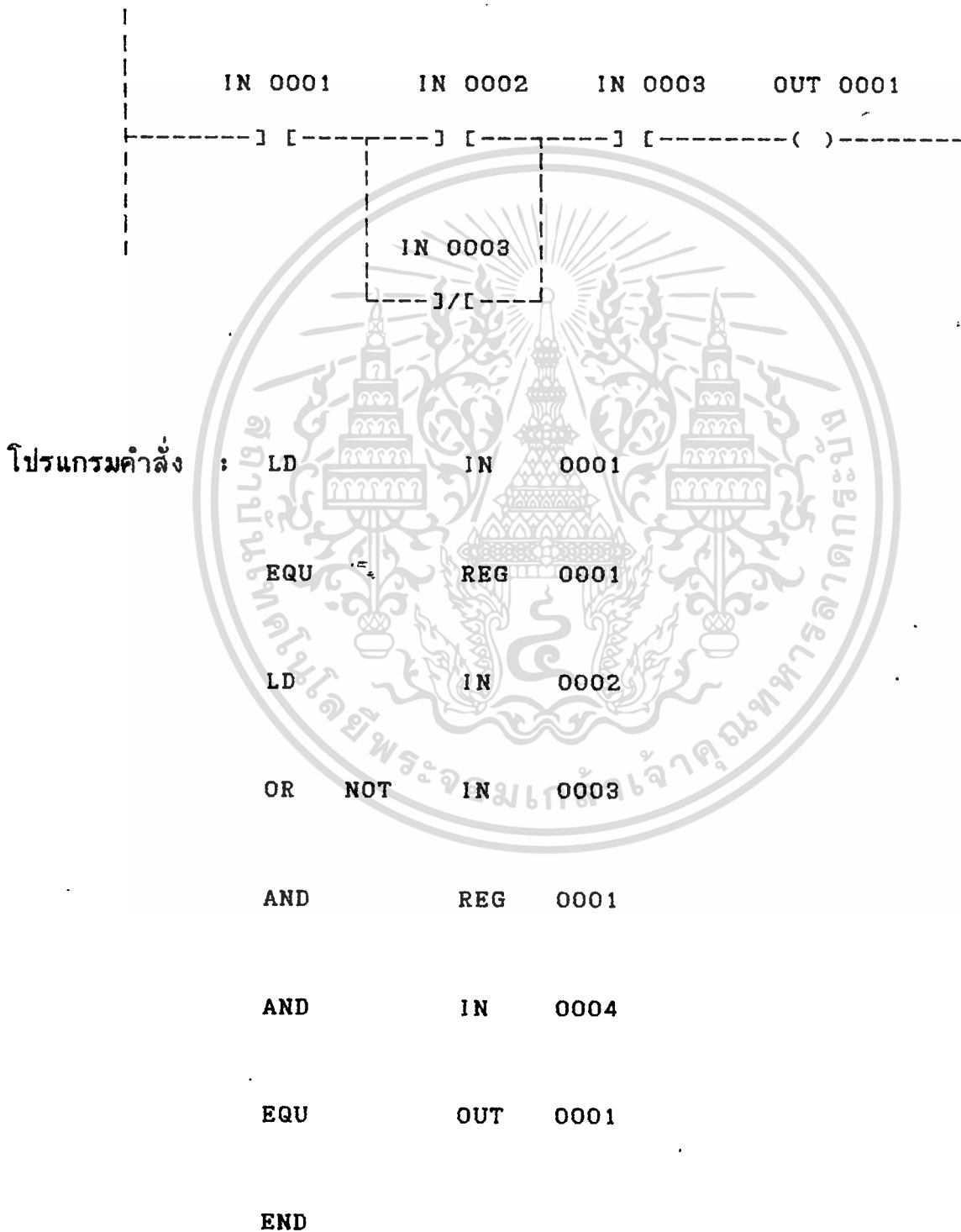
โปรแกรมคำสั่ง :

```
LD      IN      0001
OR      OUT     0003
AND     NOT     IN      0006
EQU     OUT     0004

END
```

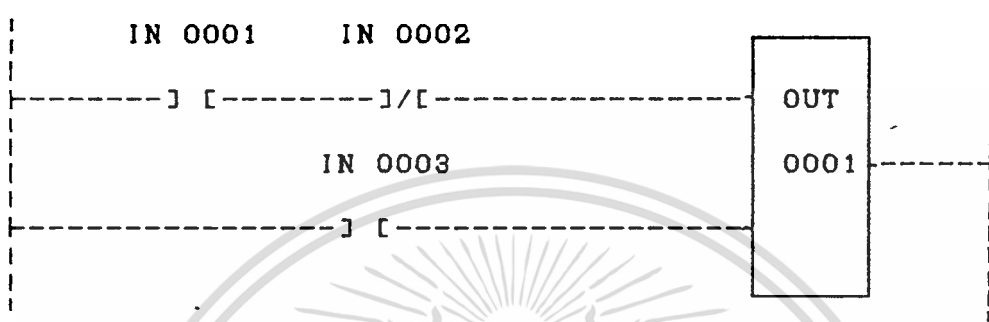
ความหมาย : เอาท์พุท 0004 จะทำงานก็ต่อเมื่ออินพุท 0001 หรือ เอาท์พุท 0003 ทำงาน และอินพุท 0006 ต้องไม่ทำงาน

ตัวอย่างที่ 5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 6



โปรแกรมคำสั่ง :

```

LD      IN      0001
AND     NOT     IN      0002
SET     OUT     0001

LD      IN      0003
RST     OUT     0001

END

```

ความหมาย : เอาท์พุท 0001 จะถูกเซทให้ทำงาน ก็ต่อเมื่ออินพุท 0001 ทำงานและอินพุท 0002 ไม่ทำงานและ ภายหลังจากที่เอาท์พุท 0001 เซตแล้วนี้แม้ว่าอินพุท 0001 และ 0002 เปลี่ยนแปลงไปก็ไม่มีผลต่อเอาท์พุท 0001 นอกจากอินพุท 0003 จึงจะเป็นการรีเซทเอาท์พุท 0001 ไม่ให้ทำงาน

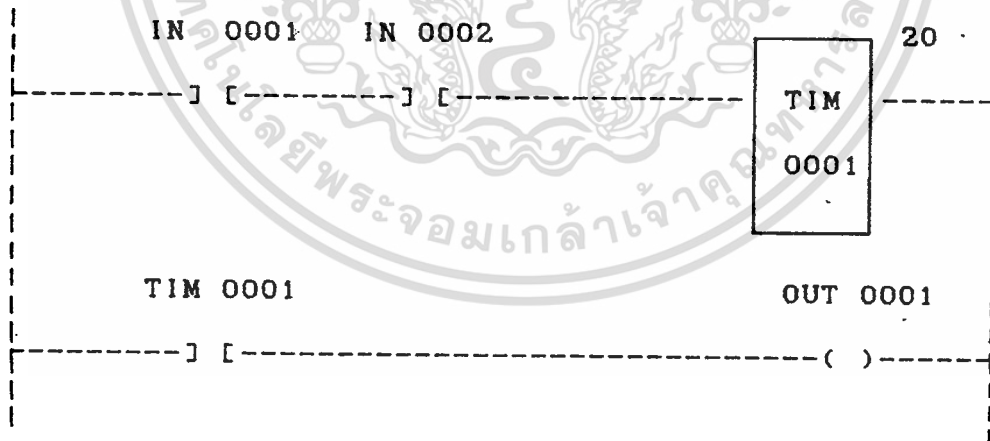
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.5 ฟังก์ชันการตั้งเวลา

ในการควบคุมกระบวนการต่างๆ จะมีเรื่องเกี่ยวกับเวลาที่เกี่ยวข้องอยู่ด้วยสำหรับพีแอลซีนี้มีตัวตั้งเวลาอยู่ 16 ตัว เริ่มจาก เลขที่ 00 01 02 .....จนถึง 15 แต่ละตัวสามารถตั้งเวลาได้ตั้งแต่ 0.1 ถึง 999.9 วินาที แต่ถ้าต้องการให้ช่วงเวลาที่ตั้งมากกว่านี้ ก็สามารถทำได้โดยการต่ออนุกรมตัวตั้งเวลาเข้ากับตัวนับ

#### ตัวอย่างที่ 7

เป็นการใช้ตัวตั้งเวลาทำการหน่วงเวลาในการที่จะให้เอาท์พุททำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมคำสั่ง : LD IN 0001

AND IN 0002

EQU TIM 0001 ; ตัวตั้งเวลาเริ่มทำงาน

SET TIM 0020 ; กำหนดช่วงเวลา 2 วินาที

LD TIM 0001

EQU OUT 0001

END

ความหมาย : ตัวตั้งเวลา 0001 จะเริ่มทำงานเมื่ออินพุต 0001 และอินพุต 0002 ทำงานทั้งคู่ และเมื่อครบเวลาที่ตั้งไว้ 2 วินาที เอาท์พุท 0001 ก็ จะทำงาน

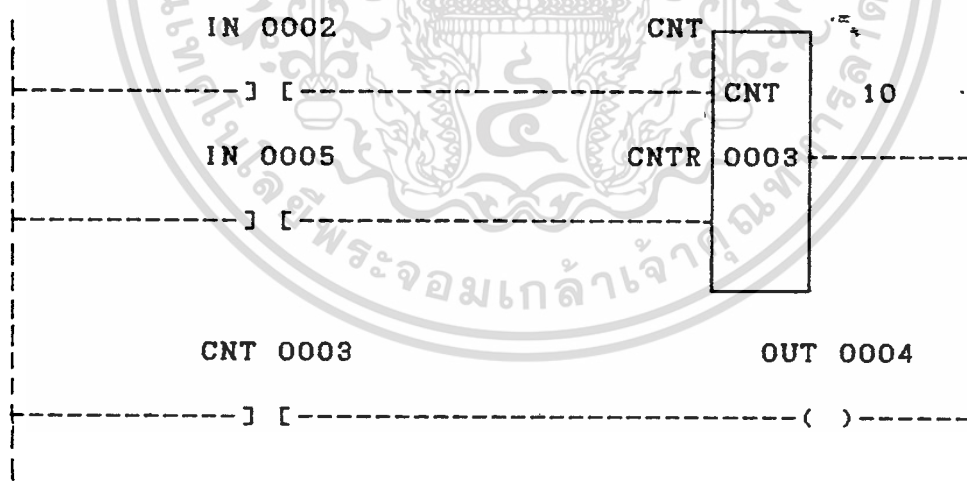
### 6.6 ฟังก์ชันการนับ

พีแอลซีที่ออกแบบประกอบด้วยตัวนับ 16 ตัว เริ่มจากเลขที่ 00 01 02 .... จนถึง 15 เรียงไปตามลำดับ แต่ละตัวสามารถนับได้ตั้งแต่ 1 ถึง 9999

อินพุทของตัวนับมี 2 อินพุท คือ

1. อินพุทนับ (Count : CNT) ; ใช้นับจำนวนพัลส์
2. รีเซ็ตตัวนับ (Count Reset : CNTR) ; ใช้รีเซ็ตตัวนับ

ตัวอย่างที่ 8



โปรแกรมคำสั่ง : LD IN 0002

EQU CNT 0003

SET CNT 0010

LD IN 0005

RES CNT 0003

LD CNT 0003

EQU OUT 0004

END

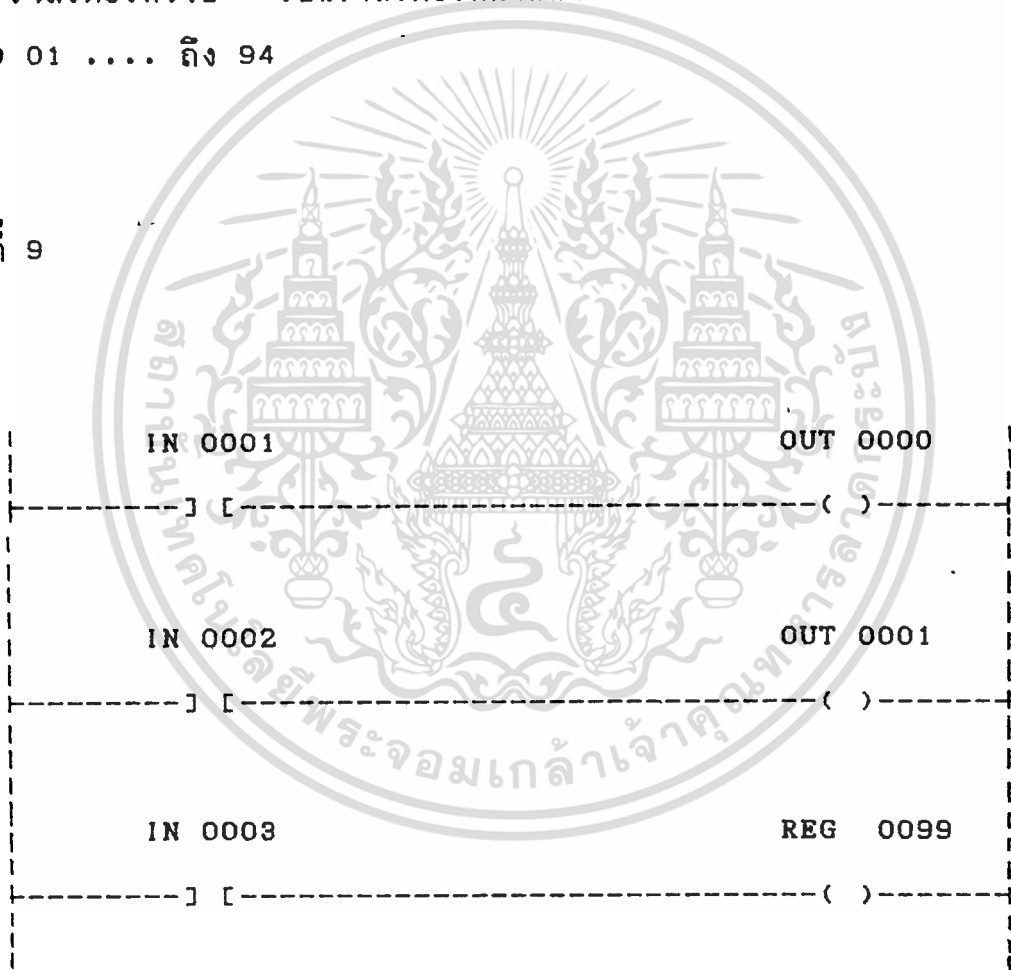
ความหมาย : เอาพท 0004 จะทำงานเมื่ออินพุท 0002 ที่ป้อนเข้าตัวนับมีถึง 10 ครั้ง แต่เมื่อใดก็ตามที่อินพุท 0005 ทำงาน ตัวนับจะถูกรีเซ็ตทันที และ เอาทพุท ก็จะถูกรีเซ็ตด้วย

### 6.7 รีจิสเตอร์

พีแอลซีที่ออกแบบมีรีจิสเตอร์ 96 ตัว ใช้สำหรับเก็บข้อมูลชั่วคราว เริ่มจากเลขที่ 00 01 02 .... จนถึง 95 เป็น Register ทั่วไป

รีจิสเตอร์ทั่วไป เป็นรีจิสเตอร์ที่ทำหน้าที่เก็บค่าสภาวะชั่วคราว โดยเริ่มจากเลขที่ 00 01 .... ถึง 94

ตัวอย่างที่ 9



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
โปรแกรมคำสั่ง : LD IN 0001  
  
EQU OUT 0000  
  
LD IN 0002  
  
EQU OUT 0001  
  
LD IN 0003  
  
EQU REG 0099  
  
END
```

ความหมาย : เอาท์พุท 0000 และ 0001 ทำงาน เมื่ออินพุท 0001 และ 0002  
ทำงานแต่ถ้าอินพุท 0003 ทำงาน เอาท์พุท 0000 และ 0001 จะ  
ไม่ทำงาน แม้ว่าอินพุท 0001 และ 0002 ยังคงทำงานอยู่ก็ตาม

## การใช้ PLC และตัวอย่างการใช้งาน

### 7.1 การใช้พีแอลซี

การใช้พีแอลซี แบ่งออกเป็น 2 โหมด (Mode) คือ

1. โหมดผู้ใช้ (User Mode) โหมดผู้ใช้เป็นโหมดของการป้อนคำสั่งต่างๆ ที่เป็นโปรแกรมควบคุม เช่น คำสั่ง LD AND OR LDI ANI ORI T SET JUMP เป็นต้น และโอเพอร์เรนด์ต่างๆ คือ IN 01 IN 02 ..... OUT 01 OUT 02 ..... TIM CNT CNTR เป็นต้น รายละเอียดเกี่ยวกับโหมดผู้ใช้นี้ได้อธิบายไว้แล้วในบทที่ 6
2. โหมดเอดิเตอร์ (Editor Mode) โหมดเอดิเตอร์เป็นโหมดของวิธีการป้อนโปรแกรมและ แก้ไขโปรแกรม เช่น การแทรกคำสั่งเข้าไปในโปรแกรม การลบคำสั่งบางคำสั่งออกจากโปรแกรม การตรวจหาคำสั่งในโปรแกรม และอื่นๆ

### 7.2 การใช้โหมดเอดิเตอร์

การใช้โหมดเอดิเตอร์สามาถจะแยกอธิบายออกเป็นส่วนๆ ได้ดังนี้

#### 1. เริ่มต้นเปิดพีแอลซี

เมื่อเริ่มเปิดเครื่องพีแอลซี ต้องตามด้วยการกดคีย์ RES ก่อน หลังจากกดคีย์แล้ว ส่วนแสดงผลของหน่วยป้อนโปรแกรมจะแสงคำว่า PLC READY จากนั้นจะต้องตามด้วยการกดคีย์ CLR พีแอลซีก็จะแสดง STEP 000 เพื่อรับคำสั่งของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การป้อนคำสั่งให้พีแอลซี

ในการป้อนคำสั่งให้พีแอลซี แต่ละคำสั่งต้องลงท้ายด้วยการกodic RET เสมอคำสั่งดังกล่าวจึงจะถูกนำไปเก็บไว้ในหน่วยความจำ และ STEP ถัดไปก็จะถูกแสดงออกมายังส่วนแสดงผลเพื่อให้ป้อนคำสั่งต่อไป เพื่อให้เห็นจริงจึงขอยกตัวอย่างแลดเดอร์โคยะแกรมในรูปแบบ

7.1 เพื่อให้ขั้นตอนการป้อนคำสั่งของตัวอย่างนี้และตัวอย่างต่อไป



รูปที่ 7.1 แลดเดอร์โคยะแกรมตัวอย่าง

STEP NO.	INSTRUCTION	KEYBOARD OPERATION
0000		
0000	LD	LD
0000	LD IN	IN
0000	LD IN 0001	0001
0000		RET
0001	AND	AND
0001	AND IN	IN
0001	AND IN 0002	0002
0001		RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STEP NO.	INSTRUCTION	KEYBOARD
	DISPAY	OPERATION
0002	EQU	EQU
0002	EQU    OUT	OUT
0002	EQU    OUT    0001	0001
0002		RET
0003	LD	LD
0003	LD    OUT	OUT
0003	LD    OUT    0001	0001
0003		RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STEP NO.	INSTRUCTION	KEYBOARD OPERATION
0004	OR NOT	OR NOT
0004	OR NOT IN	IN
0004	OR NOT IN 0003	0003
0004		RET
0005	EQU	EQU
0005	EQU OUT	OUT
0005	EQU OUT 0000	0002
0005		RET
0006	END	END
0006		RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



STEP NO.	INSTRUCTION	KEYBOARD OPERATION
.....	FUN 0 0007	RES
0007	PLC READY	PRO
0000	LD IN 0001	INC
0001	AND IN 0002	INC
0002	EQU OUT 0001	INC
0003	LD OUT 0001	INC
0004	OR NOT IN 0003	INC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

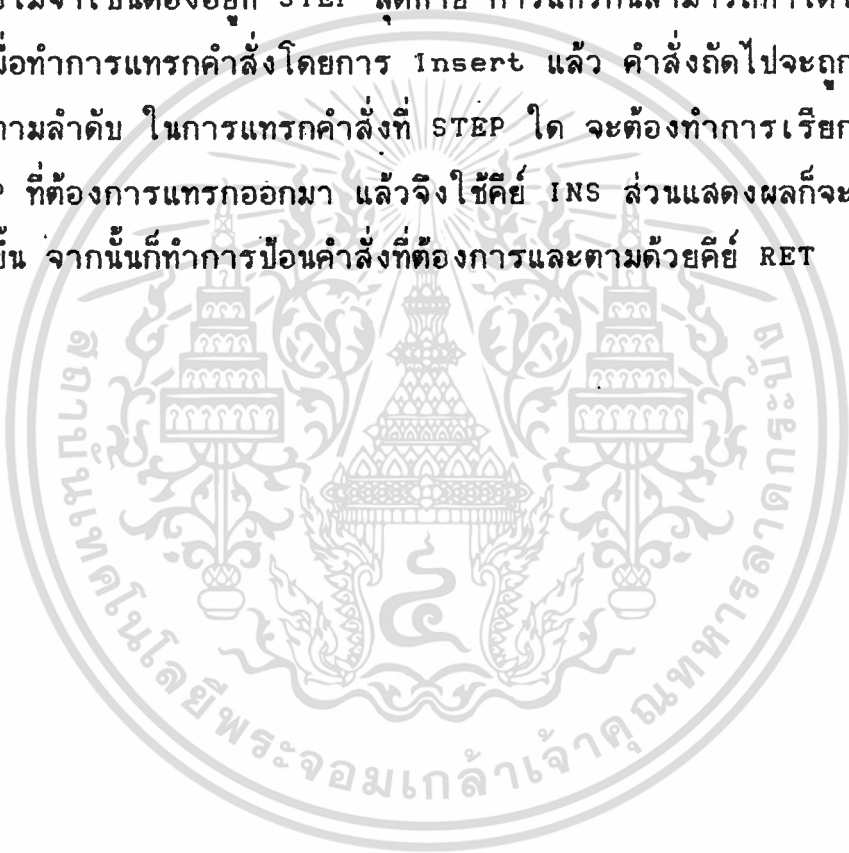
#### 4. การลบคำสั่งที่ STEP ใดๆ ในโปรแกรมควบคุม

เมื่อต้องการลบคำสั่งที่ STEP ใดๆ ในโปรแกรมควบคุม ก็สามารถทำได้โดยการเรียกคำสั่งที่ STEP นั้นมาดู แล้วก็กดคีย์ DEL คำสั่งก็จะถูกลบออกจาก STEP นั้น คำสั่งต่างๆ ที่อยู่ใน STEP ถัดมาจะเลื่อนเข้ามาแทนที่ตามลำดับ

STEP NO.	INSTRUCTION	KEYBOARD OPERATION
0001	DISPAY AND IN	0002 DEL
0001	EQU OUT	0001 INC
0002	LD OUT	0001 INC
0003	OR NOT IN	0003

## 5. การสอดแทรกคำสั่งเข้าไปในโปรแกรมควบคุม

ที่กล่าวมาแล้วมีการลบคำสั่งออกจากโปรแกรมควบคุม และมีการลบคำสั่งทั้งโปรแกรม แต่ถ้าหากต้องการแทรกคำสั่งเข้าไปในโปรแกรม โดยแทรกเข้าไปยังส่วนต่างๆ ของโปรแกรมซึ่งไม่จำเป็นต้องอยู่ที่ STEP สุดท้าย การแทรกนี้สามารถทำได้โดยการใช้คำสั่ง Insert เมื่อทำการแทรกคำสั่งโดยการ Insert แล้ว คำสั่งถัดไปจะถูกเลื่อนออกไปทีละ STEP ตามลำดับ ในการแทรกคำสั่งที่ STEP ใด จะต้องทำการเรียก STEP ที่อยู่ก่อนหน้า STEP ที่ต้องการแทรกออกมา แล้วจึงใช้คีย์ INS ส่วนแสดงผลก็จะแสดง STEP ที่ต้องการแทรกขึ้น จากนั้นก็ทำการป้อนคำสั่งที่ต้องการและตามด้วยคีย์ RET



STEP NO.	INSTRUCTION	KEYBOARD OPERATION
0001	AND IN 0002	INS
0002		AND
0002	AND IN	IN
0002	AND IN 0003	0003
0002	AND IN 0003	RET
0003	EQU OUT 0001	INC
0004	LD OUT 0001	INC
0005	OR NOT IN 0003	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6. การลบบ้างขณะโปรแกรม

การลบบ้างขณะโปรแกรมในที่นี้หมายถึง กรณีขณะทำการป้อน้างขณะใดๆ โดยยังไม่ได้กดคีย์ RET หากต้องการเปลี่ยน้างขณะใหม่ ก็สามารถแก้ไขได้โดยการลบบ้างขณะนั้นออกไป วิธีการดังกล่าวทำได้โดยการกดคีย์ DEL และตามด้วยคีย์ PRO ฟีแอลซีก็แสดง STEP เดิมขึ้นมาอีกเพื่อรับ้างขณะใหม่



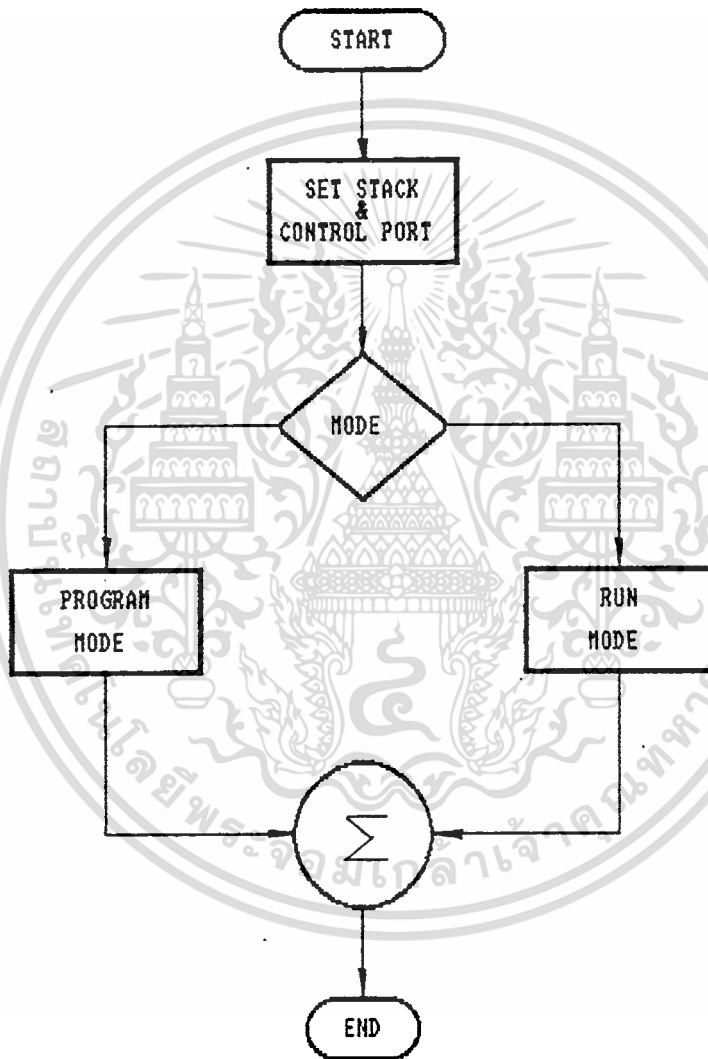
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทสรุป

เครื่องพีแอลซีที่ได้ออกแบบขึ้นมา นี้ สามารถใช้ในงานควบคุมได้ดีเช่นเดียวกับพีแอลซีอื่นๆ ที่ได้สร้างขึ้นมาขายอยู่ในท้องตลาดปัจจุบันนี้ แต่มีราคาถูกกว่าหลายเท่าตัวคือราคาประมาณไม่เกิน 4,000.-บาท (สี่พันบาท) ถ้ามีการพัฒนาสร้างพีแอลซีนี้ออกมาเพื่อสนับสนุนให้โรงงานอุตสาหกรรมทั้งขนาดเล็กและขนาดกลางไปใช้งานแล้วจะเป็นการประหยัดเงินไม่ให้สูญเสียออกนอกประเทศได้ทางหนึ่ง และยังทำให้ความน่าเชื่อถือ (Reliability) ของระบบควบคุมเครื่องจักรหรือกระบวนการที่ใช้พีแอลซีในการควบคุมดีขึ้นด้วย เมื่อเทียบกับระบบเดิมที่ใช้วงจรควบคุมแบบรีเลย์

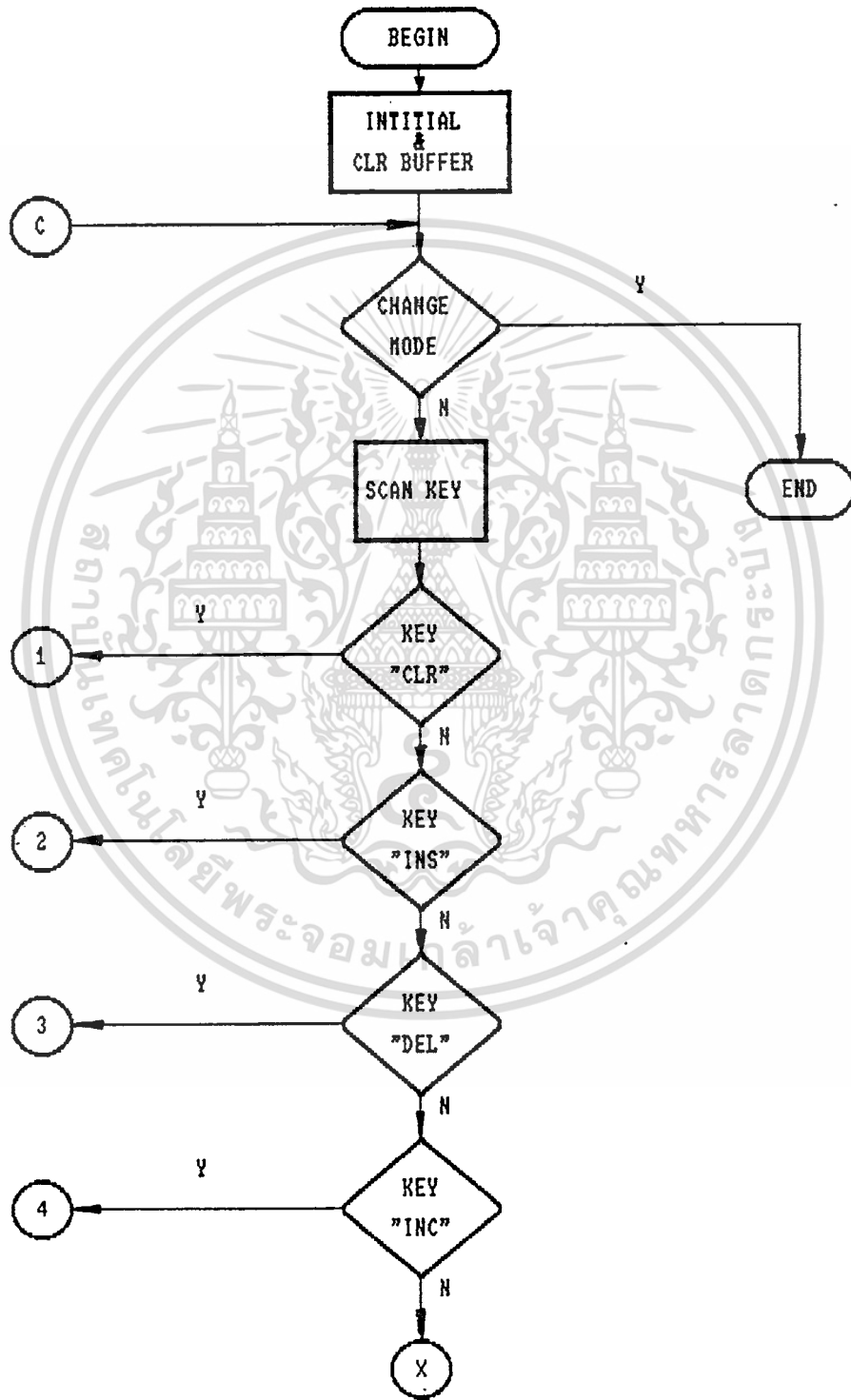
อย่างไรก็ดี ฟังก์ชันพิเศษเพื่อการทำงานที่ย่งยากซับซ้อนมาก ๆ ซึ่งจะพบได้ในกระบวนการขนาดใหญ่ ๆ นั้น ในพีแอลซีที่ได้ออกแบบขึ้นมา นี้ยังไม่ได้มีการพัฒนาไว้ทั้งนี้ เพราะจะทำให้พีแอลซีมีขนาดใหญ่ขึ้น และมีราคาสูง เนื่องจากในกรณีที่ต้องการให้มีฟังก์ชันการทำงานที่ย่งยากซับซ้อน เช่น การบวก การลบ การคูณ การหาร หรือการควบคุมแบบต่อเนื่องที่มีฟังก์ชันการควบคุมพีไอดี (PID : Proportional Integral Derivative Controller) เป็นต้นนั้น จำเป็นจะต้องใช้ไมโครโปรเซสเซอร์ขนาด 8 บิต จำนวนหลาย ๆ ตัว หรือใช้ไมโครโปรเซสเซอร์ขนาดมากกว่า 8 บิต เพื่อให้มีการทำงานที่รวดเร็ว และทันต่อการเปลี่ยนแปลงของกระบวนการ โดยทั่วไปแล้วพีแอลซีที่มีฟังก์ชันการทำงานเหล่านี้รวมอยู่ด้วยนั้นจะเรียกว่าพีซี (PC : Programmable Controller) ซึ่งน่าจะมีการพัฒนาและออกแบบสร้างขึ้นมาเองต่อไป

ปริญญาโทฉบับนี้คงจะเป็นประโยชน์ต่อผู้ที่สนใจ และต้องการจะทำการศึกษาวิจัยเกี่ยวกับพีแอลซีหรือพีซีเพิ่มขึ้น และคาดว่าพีแอลซีที่ได้ออกแบบพัฒนาสร้างมานี้จะเป็นประโยชน์ต่อโรงงานอุตสาหกรรมของประเทศต่อไปในอนาคต



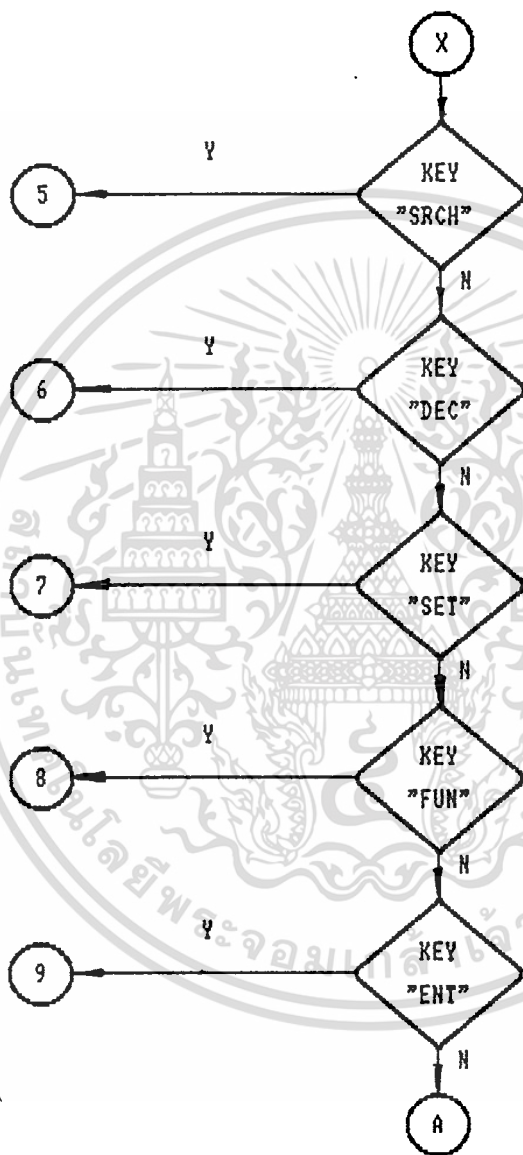
### — มสสภรทำงานการเลือกโหมดของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



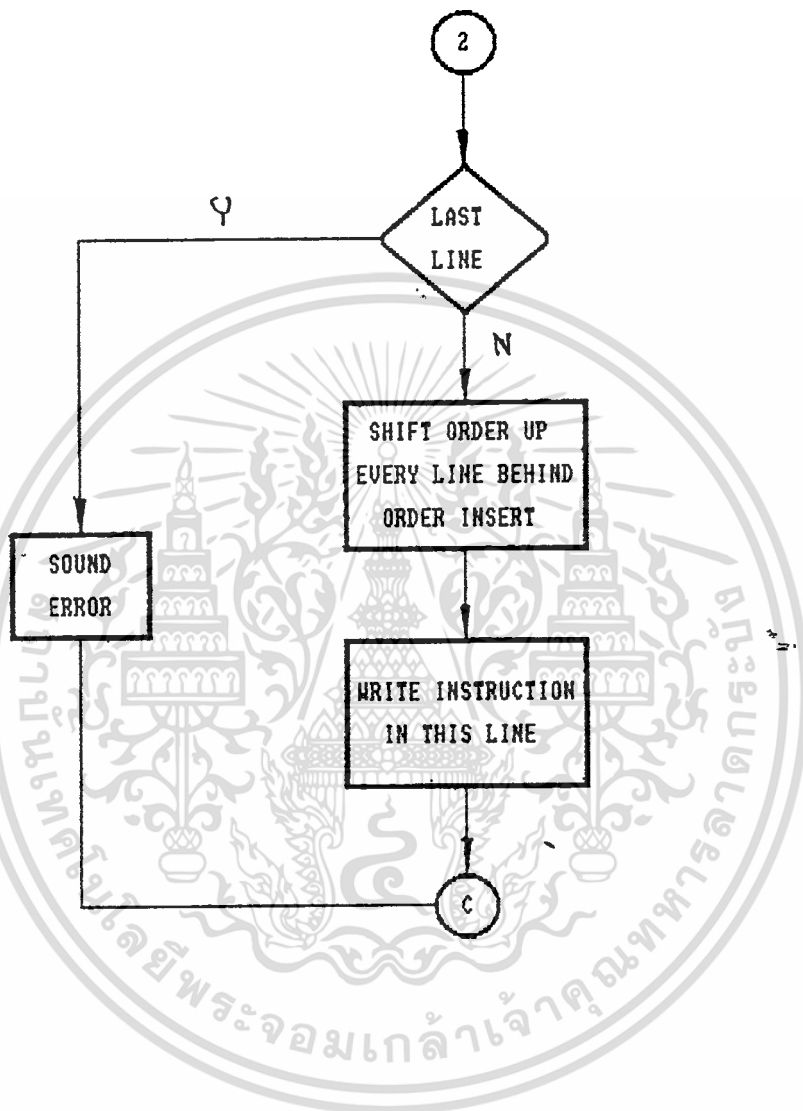
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้ทำงานเพื่อการศึกษาเท่านั้น โดยเอกสารนี้จะไม่ไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นแสดงการทำงานขยวระยช PC เชน โทมดไปรกรม



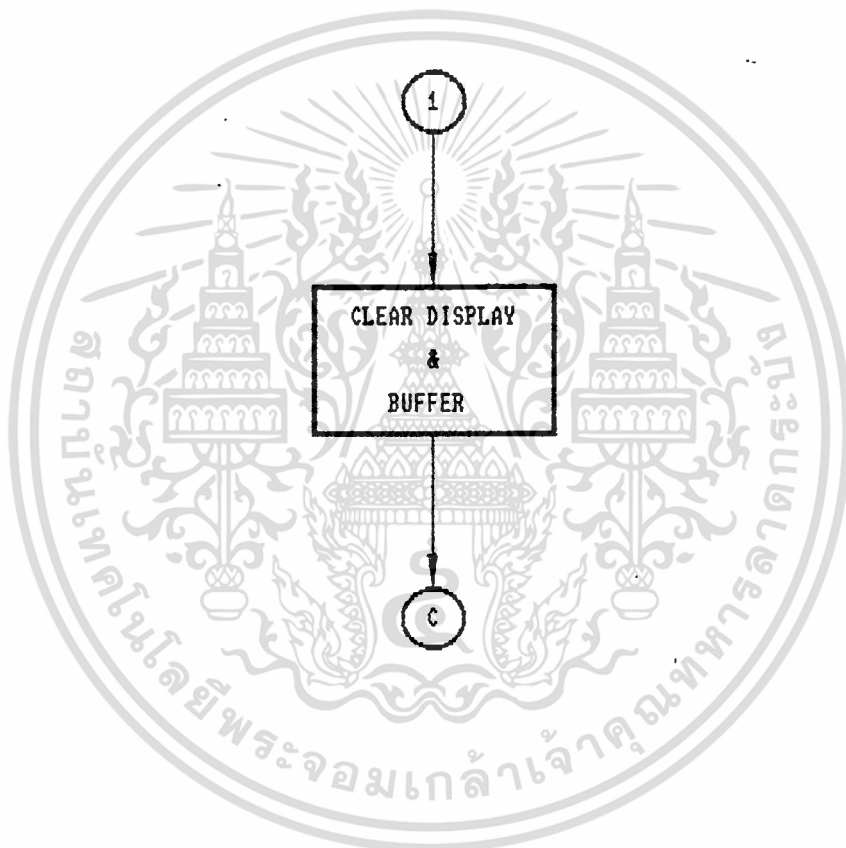
— ผลงานการทำมาของรยช P LC โย โภคต โพรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



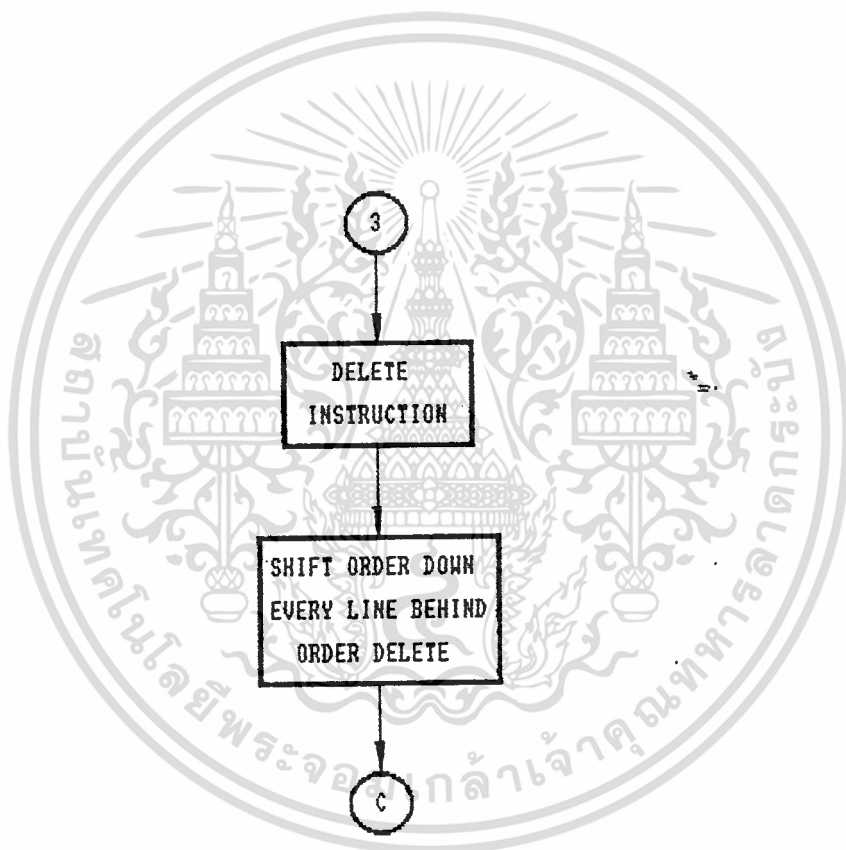
— แสดงการทำงานของ การ INSERT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



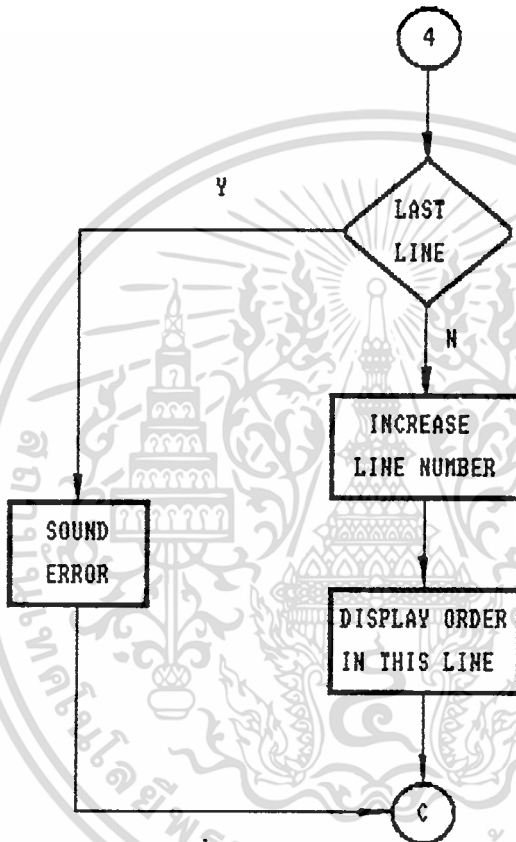
— แสดงการทำงาน การ CLEAR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



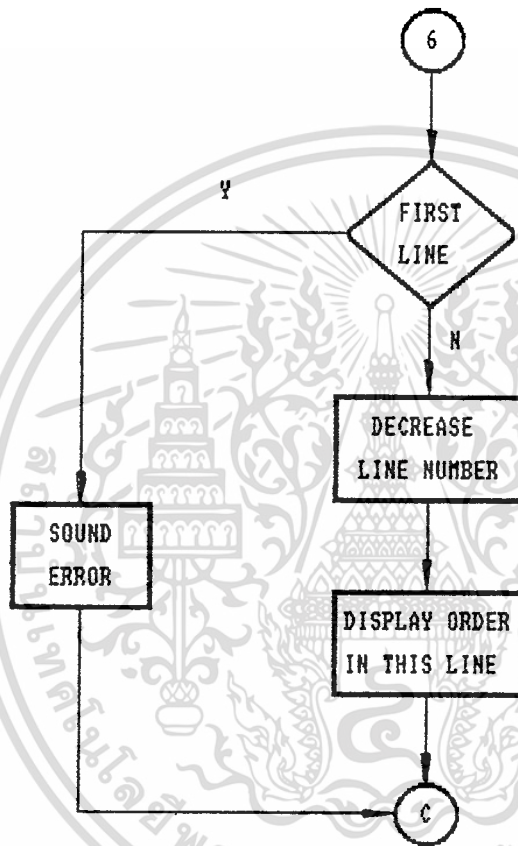
### คำสั่งการทำงาน การ DELETE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



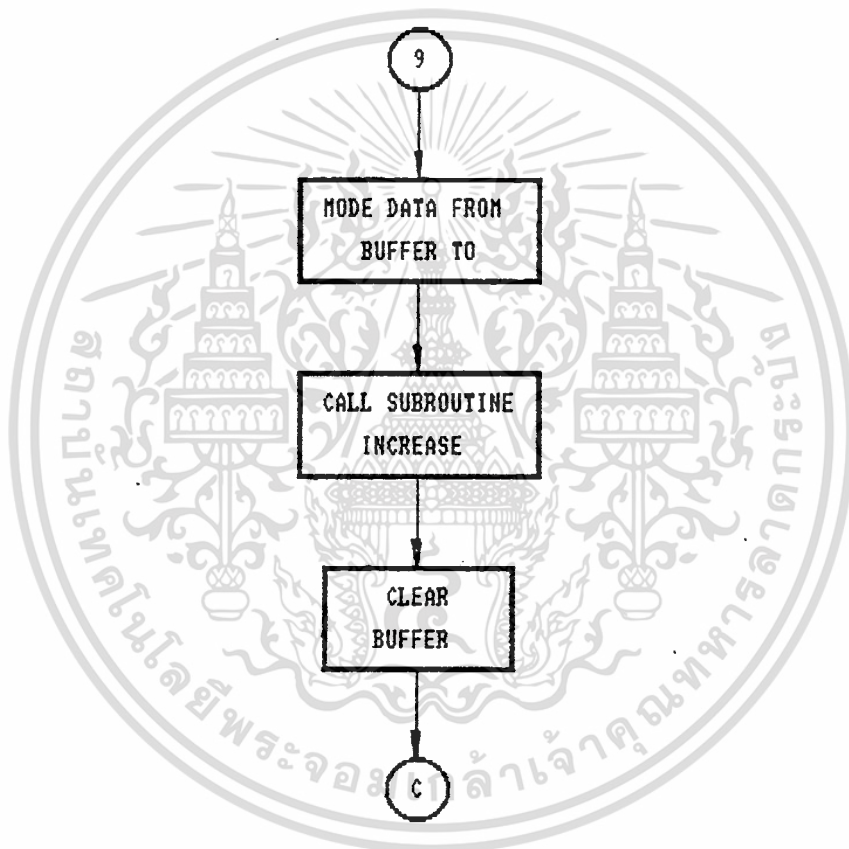
แผนผังการทำงานของการ INCREASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



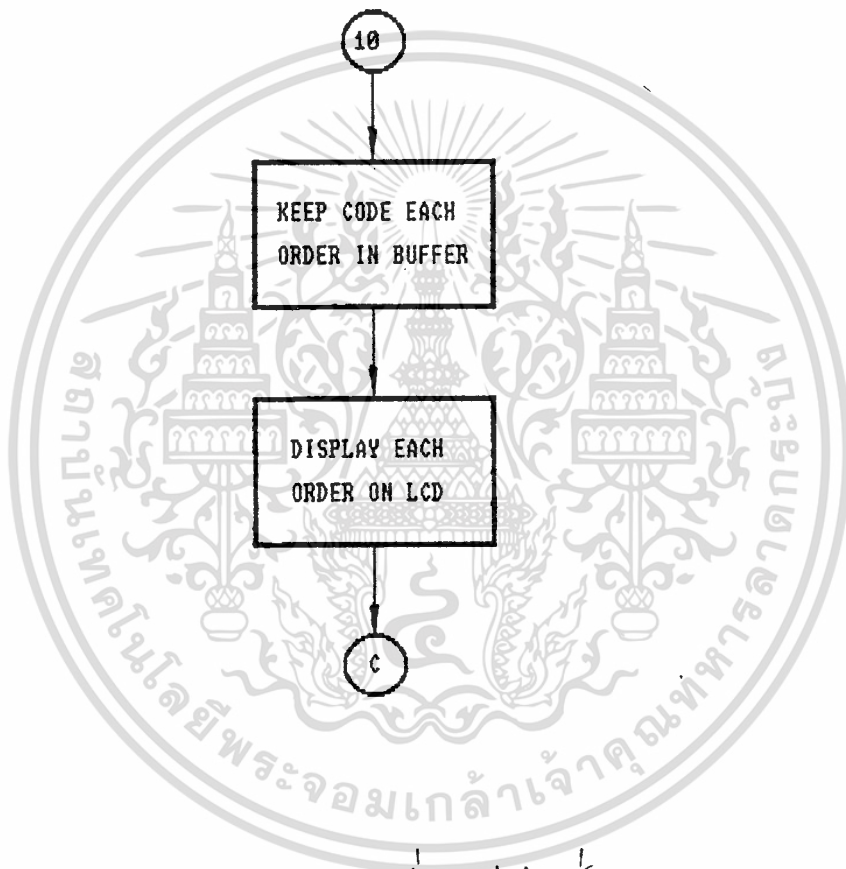
แสดงการทำงานขงฉร DECREASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



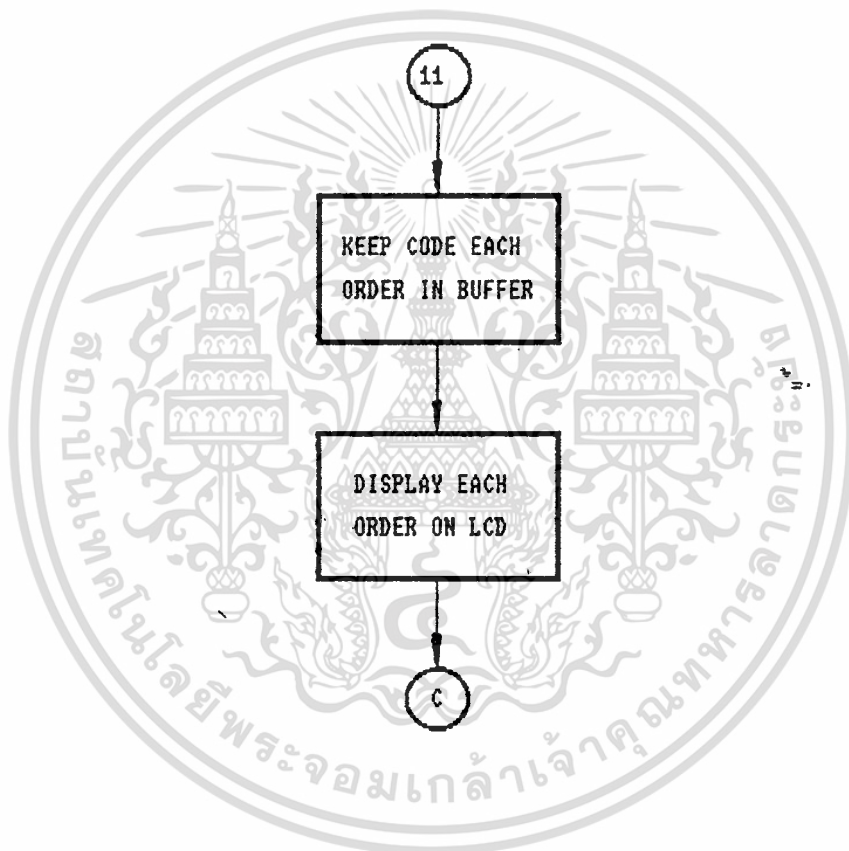
----- มีสวิตช์การทำงาน ของการ ENTER -----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



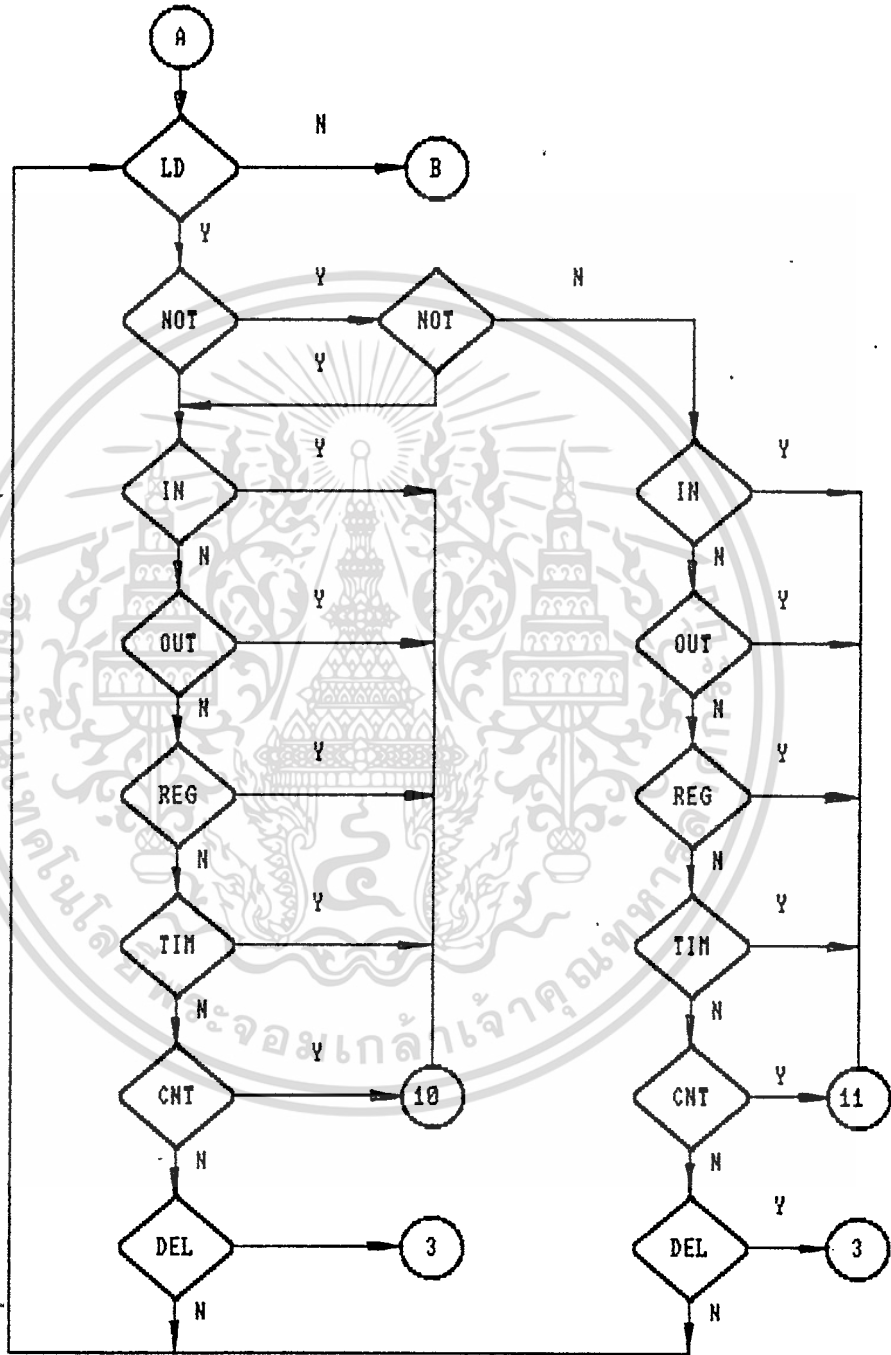
ผลของการทำมาชมมออกดดยด้าสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



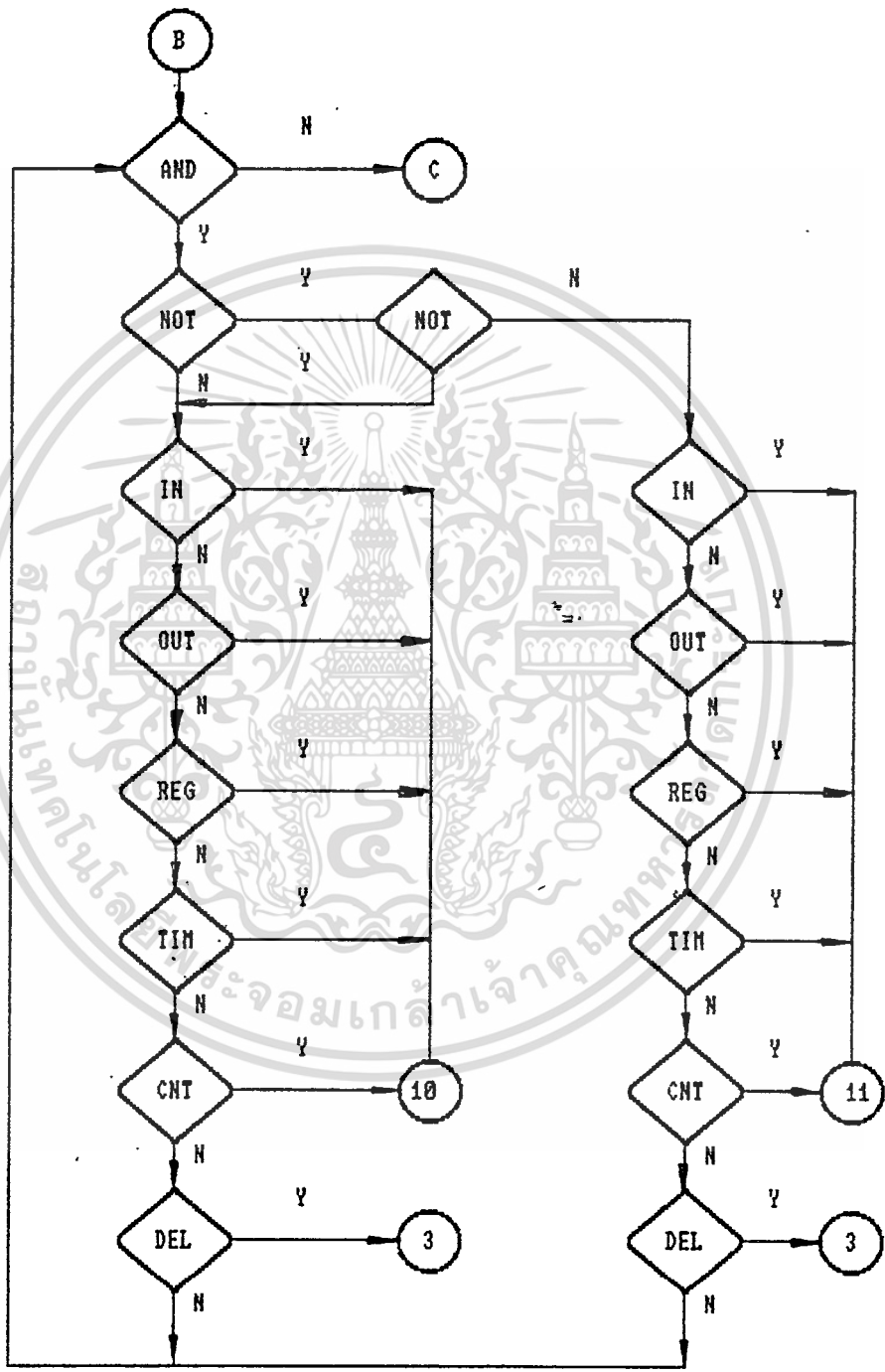
—ผลงานการทำงาน เมื่อทดลองคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

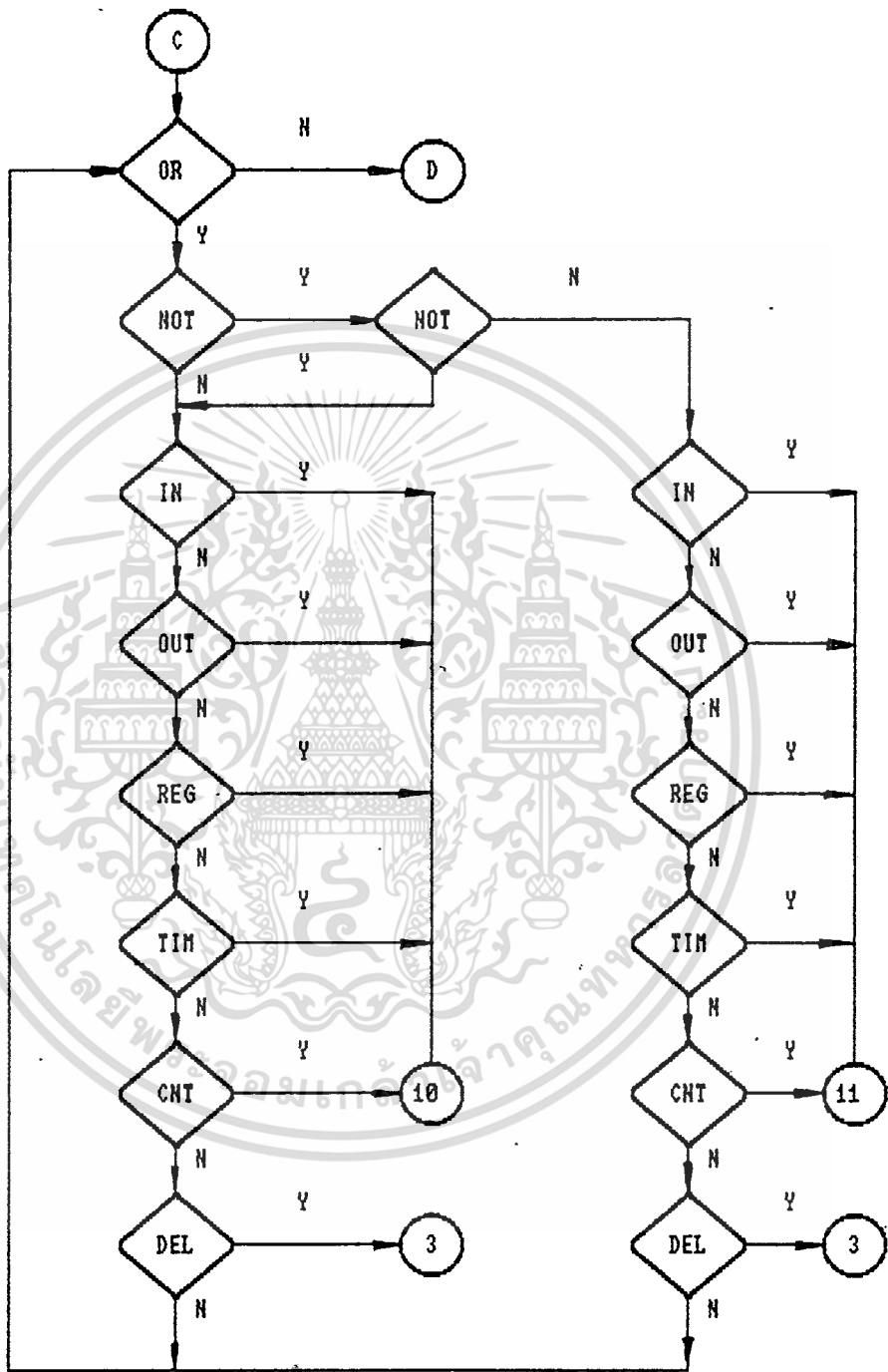


— มสสงครามทำงานของกร LOAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

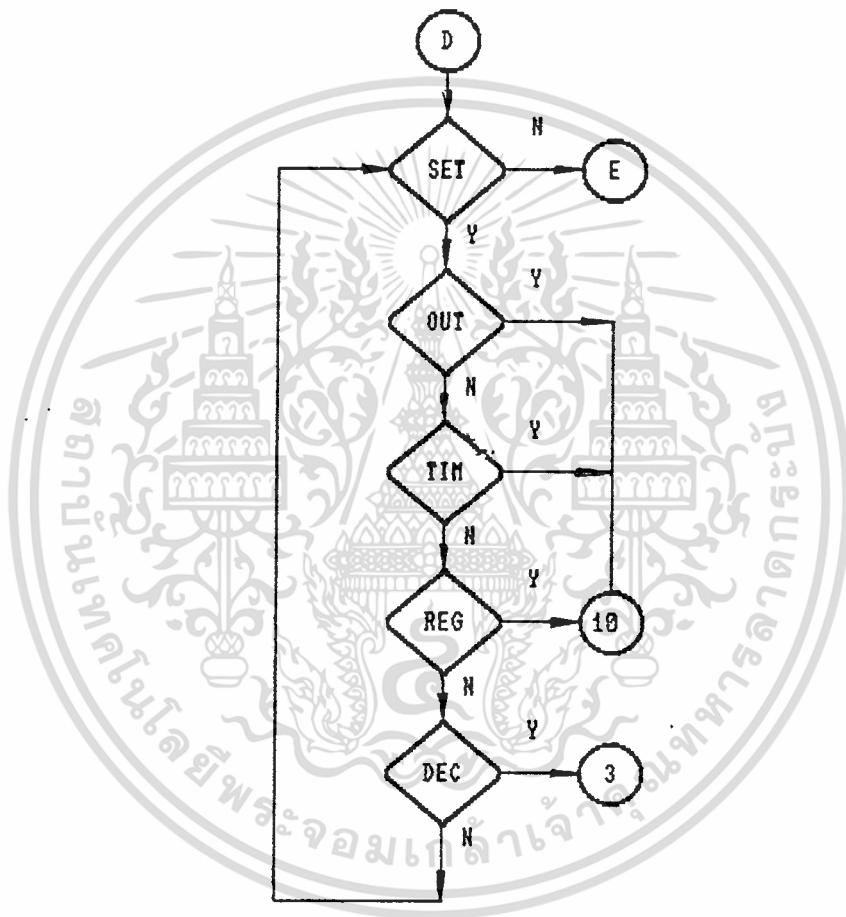


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรทางงานของภาค AND มอนูญาติให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



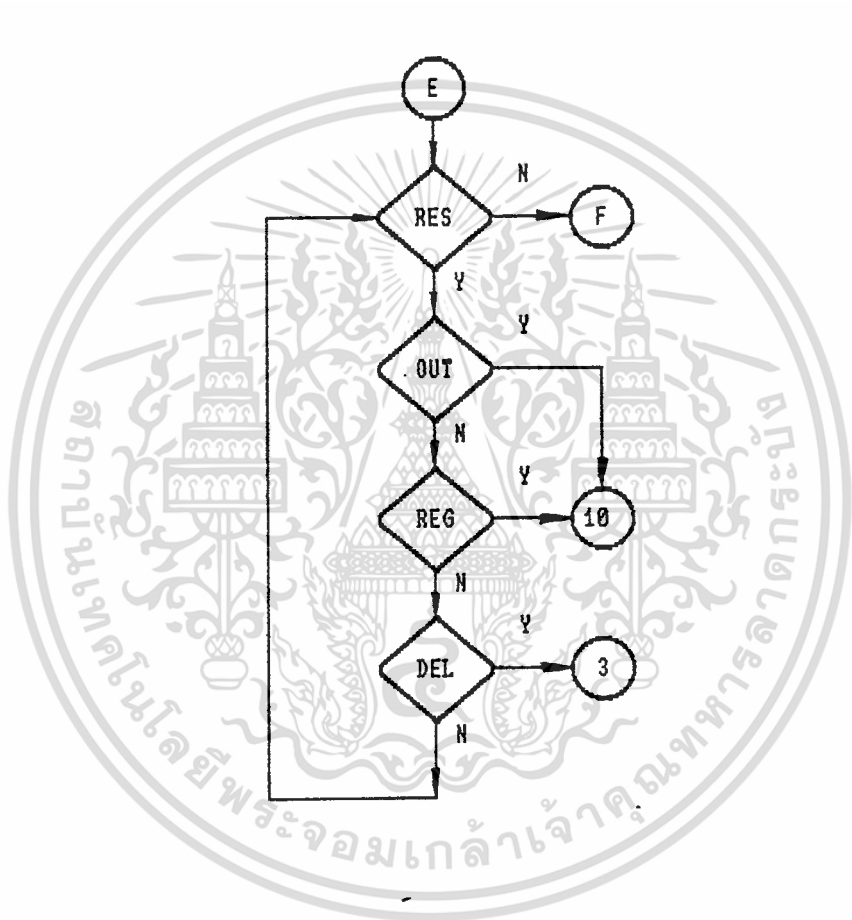
โมเดลการทำงานของเกท OR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



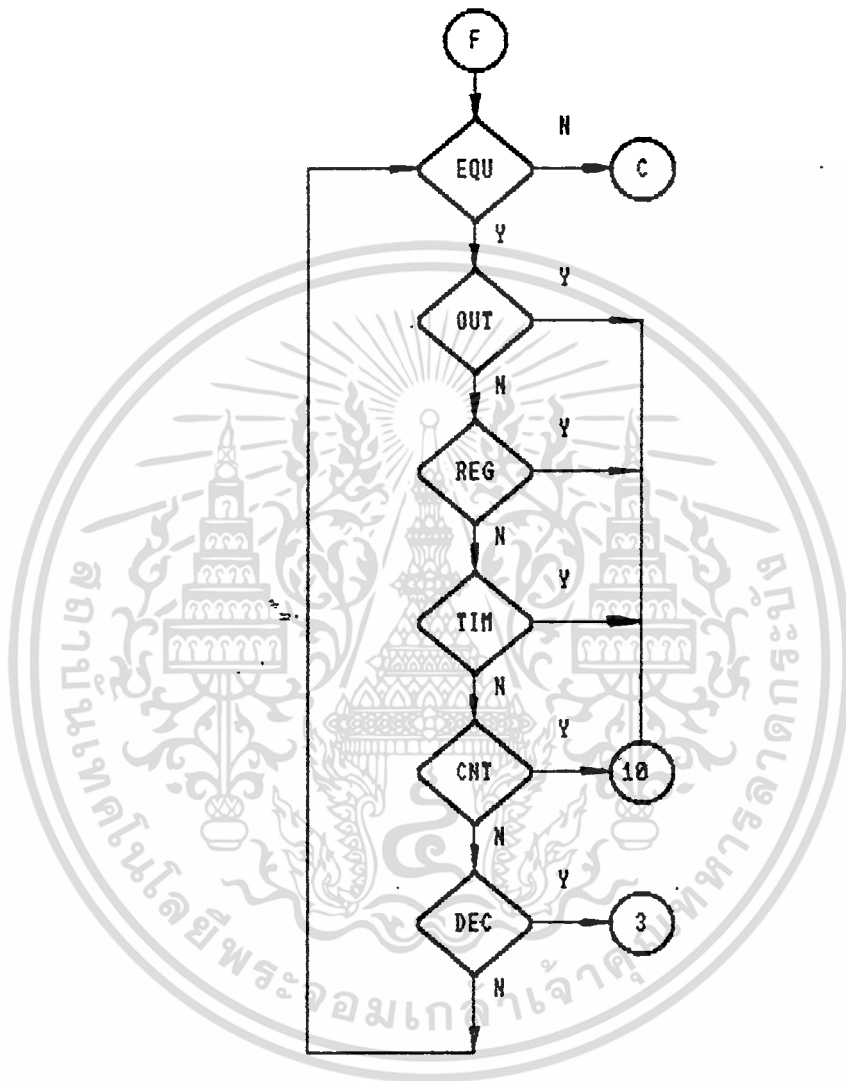
— นวัตกรรมทางวิศวกรรม SET —

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงการทำงานของวงจร RESET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงการทำงานของคำสั่ง EQU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. C.T.Jones, L.A. Bryan, "PROGRAMMABLE CONTROLLERS CONCEPTS AND APPLICATION" ,An Ipc/ASTEC Publication, Atlanta, 1983.
2. Ian G. Warnock, "PROGRAMMABLE CONTROLLERS OPERATION AND APPLICATION", Prentice Hall, 1988.
3. "OPTO ELECTRONICS DEVICE DATA", MOTOROLA Inc., 1983.
4. "LOGIC DATABOOK", National Semiconductor Corp., 1981.
5. "MCS-51 MICROPROCESSOR DATABOOK AND MICROCONTROLLER", INTEL Inc.
6. ศิวัฒน์ เลาสงคราม, "คู่มือ MCS-51 MICROCONTROLLER".
7. สุพรรณ กุลหาญชัย, "พีแอลซีและเทคโนโลยีการงานเบื้องต้น", 2533.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้