



โปรแกรมจำลองเครื่องควบคุมตรรกที่โปรแกรมได้
PROGRAMABLE LOGIC CONTROLLER SIMULATION



โดย

นายเผ่าพันธ์ ฐนุถนัด

นายเลิศพร ไตรรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ 032782
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจำลองเครื่องคอมพิวเตอร์ที่โปรแกรมได้

นาย เผ่าพันธ์	ภณรัตน์	32.1193	4j
นาย เลิศพร	ไตรรัตน์	32.1270	4j

ผศ. พิพัฒน์ เลาสงคราม
อาจารย์ที่ปรึกษา

(PROGRAMABLE LOGIC CONTROLLER SIMULATION)



MR. PIPAT LAOHASONGKRAM

ADVISOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชา
เทคโนโลยีการวัดคุมทางอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง...

โปรแกรมจำลองเครื่องควบคุมตรรกที่โปรแกรมได้
(PROGRAMABLE LOGIC CONTROLLER SIMULATION)

ผู้จัดทำ...

นาย เผ่าพันธ์ ขนุกนัถ	32.1193	4j
นาย เลิศพร ไตรรัตน์	32.1270	4j

ผศ. พิพัฒน์ เลาสงคราม

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ปัจจุบันนี้เครื่องควบคุมแบบ SEQUENCE CONTROLER ได้เจริญก้าวหน้าขึ้นมามากเมื่อเทียบกับอดีตที่ผ่านมา พร้อมทั้งมีความสามารถในการทำงานที่ซับซ้อนได้ และยังสามารถทำการโปรแกรมได้ จึงทำให้เป็นที่แพร่หลายมากในวงการอุตสาหกรรม สำหรับประเทศไทยเราเพิ่งจะได้เริ่มมีการนำเอา PROGRAMABLE LOGIC CONTROLLER (PLC) มาใช้ ซึ่งอาจยังไม่เป็นที่แพร่หลาย ปรวิญญานพจน์ฉบับนี้ จึงขอเสนอการออกแบบพัฒนาโปรแกรมจำลองของเครื่องควบคุม PLC เพื่อเป็นประโยชน์ในการช่วยเผยแพร่ PLC ในอุตสาหกรรมภายในประเทศ

ABSTRACT

At present the sequence controller has been used so much in many industrails. Because of and ability to do a complex work and it can be also programmable. so it is flexsible to do a different work. In Thailand, Programmable Logic Controller(PLC) has just been introduced.

Many people may not know and familar with it so will purposed the SIMULATOR PLC to be used on pc. To make it easy to be familar with PLC. So it could make PLC to be knowby every one.

สารบัญ

บทนำ

บทที่ 1 การควบคุมซีเคັນซ์	1
1.1 ความหมาย	1
1.2 ระบบควบคุมซีเคັນซ์	1
1.3 ข้อเปรียบเทียบระหว่าง PC - วงจรรีเลย์	2
บทที่ 2 ทฤษฎีพื้นฐานและหลักการทํางาน	4
2.1 โครงสร้างของ PLC	4
2.2 หน่วยประมวลผลกลาง	5
2.3 หน่วยอินพุต-เอาต์พุต	7
2.4 หน่วยประมวลผลอินพุต-เอาต์พุต	8
2.5 หน่วยป้อนโปรแกรม	9
บทที่ 3 ภาษาคําสั่งของ PLC	10
3.1 ภาษาแลดเดอร์ไดอะแกรม	10
3.2 ความสัมพันธ์ระหว่างแลดเดอร์ไดอะแกรมกับฟังก์ชันต่างๆ	10
3.3 ภาษาคําสั่งบูลีน	15
บทที่ 4 ชุดคำสั่งใช้งาน	16
- 4.1 จำนวนอินพุต เอาต์พุต ตัวตั้งเวลา ตัวนับ	16
4.2 องค์ประกอบของโปรแกรมผู้ใช้	17
4.3 การจำแนกคิรร์บอร์ด	17
4.4 คำสั่งเบื้องต้น	19
4.5 ฟังก์ชันการตั้งเวลา TIMER : TIM	28
4.6 ฟังก์ชันการนับ COUNTER : CNT	29

ไฟล์ชาร์ตแสดงการทำงานของโปรแกรม

โปรแกรม

บทสรุป

กิจกรรมประกาศ

เอกสารอ้างอิง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

การควบคุมซีเควินซ์

1.1 ความหมาย

การควบคุมซีเควินซ์ เป็นการควบคุมลำดับการทำงานตามเงื่อนไข หรือโปรแกรมที่กำหนดไว้แล้ว

สัญญาณในการควบคุมส่วนใหญ่เป็นสัญญาณเปิด-ปิด (ON/OFF) ให้อุปกรณ์ เช่น มอเตอร์ โซลินอยด์ หลอด ให้ทำงานหรือหยุดทำงานเท่านั้น ในการทำงานของอุปกรณ์เหล่านี้ นอกจากจะพิจารณาแล้ว ยังพิจารณาถึงช่วงเวลาการทำงานด้วย

การใช้งานของมันมีมากมายตั้งแต่เครื่องใช้ไฟฟ้าเช่น เครื่องซักผ้า ตู้เย็น อุปกรณ์ใกล้ตัว เช่น ลิฟท์ บรรดาเคลื่อน ปีม้วน จนกระทั่งเครื่องจักรกลในโรงงานอุตสาหกรรม เช่น เครื่องกลึงอัตโนมัติ สายพานลำเลียง หม้อน้ำ และเครื่องกำเนิดไฟฟ้า เป็นต้น

ปัจจุบัน การควบคุมแบบซีเควินซ์ ได้รับความนิยอย่างแพร่หลายในโรงงานอุตสาหกรรม โดยมีจุดมุ่งหมายในการทำให้เครื่องจักรทำงานเองโดยอัตโนมัติ และเป็น การลดกำลังคน ประหยัดวัตถุดิบและการตรวจคุณภาพสินค้าที่ผลิต

1.2 ระบบควบคุมแบบซีเควินซ์ (SEQUENCE CONTROL)

ระบบควบคุมแบบซีเควินซ์ (SEQUENCE CONTROL) แต่เดิมประกอบด้วยอุปกรณ์ไฟฟ้าเชิงกล (ELECTROMECHANICAL DEVICE) ซึ่งได้แก่รีเลย์ ตัวตั้งเวลา ตัวนับ เป็นต้น ลักษณะของสัญญาณมีอยู่ด้วยกันสองสภาวะ คือ เปิด-ปิด (ON/OFF) ระบบควบคุมแบบซีเควินซ์ที่ประกอบขึ้นจากอุปกรณ์ดังกล่าวข้างต้น มีข้อเสียหลายประการ เช่น มีขนาดใหญ่ ใช้กำลังงานสูง และเมื่อมีการเปลี่ยนแปลงการทำงานก็ต้องสร้างวงจรควบคุมใหม่ ซึ่งไม่สามารถประยุกต์ใช้กับวงจรที่มีความซับซ้อนเหล่านี้ได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยญาติให้มาเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเทคโนโลยีทางด้านไมโครโพรเซสเซอร์ (MICROPROCESSOR) ในปัจจุบันได้เจริญก้าวหน้าไปมาก จึงได้มีการวิจัยด้านการควบคุมแบบต่อเนื่อง เป็นต้นกำเนิดเครื่องควบคุมชนิดโปรแกรมได้ (PROGRAMMABLE SEQUENCE CONTROLLER) หรือเรียกสั้น ๆ ว่า PC PC จะมีส่วนที่เป็นอินพุตที่สามารถต่อใช้งานเข้ากับตัวตรวจจับต่างๆ (SENSOR) และ ส่วนของเอาต์พุตจะต่อไปควบคุมการทำงานของอุปกรณ์ หรือเครื่องจักรกลโดยสามารถสร้างวงจรหรือเงื่อนไขในการทำงานของเครื่องจักรเหล่านี้ โดยการป้อนเป็นโปรแกรมคำสั่งที่เรียกว่า LADDER DIRGRAM เข้าไปโปรแกรมนี้จะทำหน้าที่เหมือนกับวงจรรีเลย์

1.3 ข้อเปรียบเทียบของ PC - วงจรรีเลย์

1. สั้นเปลืองเนื้อที่น้อยกว่าเพราะมีขนาดเล็ก
2. สามารถใช้ควบคุมระบบขนาดใดก็ได้ ถ้าเลือกขนาดให้เหมาะสม
3. การเปลี่ยนลำดับขั้นตอนการทำงานหรือเงื่อนไข สามารถทำได้ง่ายโดยการโปรแกรม
4. ตัวตั้งเวลา ตัวนับ จะเป็นซอฟต์แวร์ทำให้กำหนดค่าต่าง ๆ ได้ง่ายและทำให้ราคาถูกลง
5. รีเลย์ภายใน เป็นซอฟต์แวร์ทำให้ขนาดการเดินสายและค่าใช้จ่ายลดลง
6. การติดตั้งทำได้สะดวก
7. การขยายระบบ ทำได้ง่าย
8. ราคาถูกกว่าแบบรีเลย์
9. ความน่าเชื่อถือสูง
10. สามารถตรวจสอบความผิดพลาดด้วยตัวเอง จึงสามารถแก้ไขเมื่อมีปัญหาง่าย
11. ลดการเดินสายยาว ๆ และลดค่าใช้จ่ายในการเดินสาย เพราะควบคุมด้วย

รีโมต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. การบำรุงรักษาทำได้ง่าย
13. เวลาในการทำงานเร็วกว่า
14. มีฟังก์ชันทางคณิตศาสตร์ บวก ลบ คูณ หาร และอื่น ๆ
15. ต่อเข้ากับคอมพิวเตอร์เพื่อให้ทำโปรแกรมได้หลายแบบ เช่น แลคเตอร์ไดอะแกรม หรือจะเป็นคำสั่งตรรกะก็ได้
16. ใช้ได้ทุกสภาพแวดล้อมการทำงาน



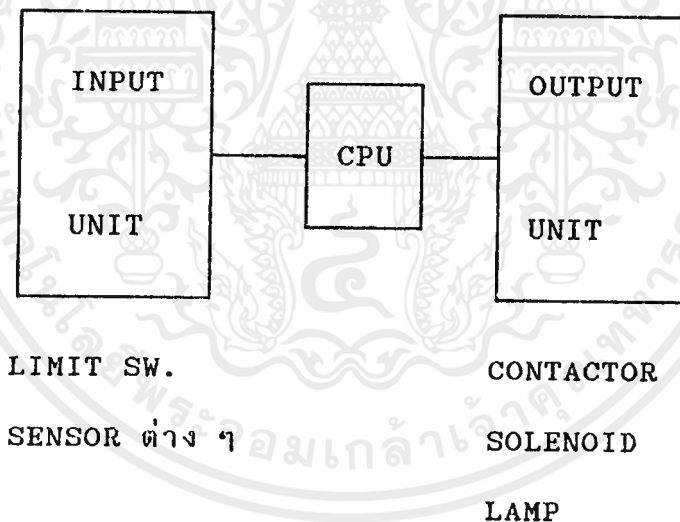
บทที่ 2

ทฤษฎีพื้นฐานและหลักการทำงาน

2.1 โครงสร้างของ PLC

PLC ประกอบด้วยส่วนสำคัญ 2 ส่วนด้วยกัน

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



รูปที่ 2.1 แสดงโครงสร้างของ PLC

นอกจากส่วนประกอบหลัก 2 ส่วนข้างต้นแล้ว PLC ยังประกอบด้วย หน่วยป้อนโปรแกรม (Programming Console) ทำหน้าที่ติดต่อระหว่าง PLC กับผู้ใช้ โดยรับ Program ที่ผู้ใช้เขียนขึ้นเก็บไว้ในหน่วยความจำปกติแล้วหน่วยป้อนโปรแกรมจะเชื่อมต่อกับ PLC เมื่อผู้ใช้ต้องการป้อน ตรวจสอบหรือแก้ไขโปรแกรมเท่านั้นและ PLC ก็สามารถทำงานได้ โดยไม่ต้องพึ่งหน่วยป้อนโปรแกรม ดังนั้น หน่วยป้อนโปรแกรมจึงไม่จัดเป็นส่วนประกอบของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ถ้ากรณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

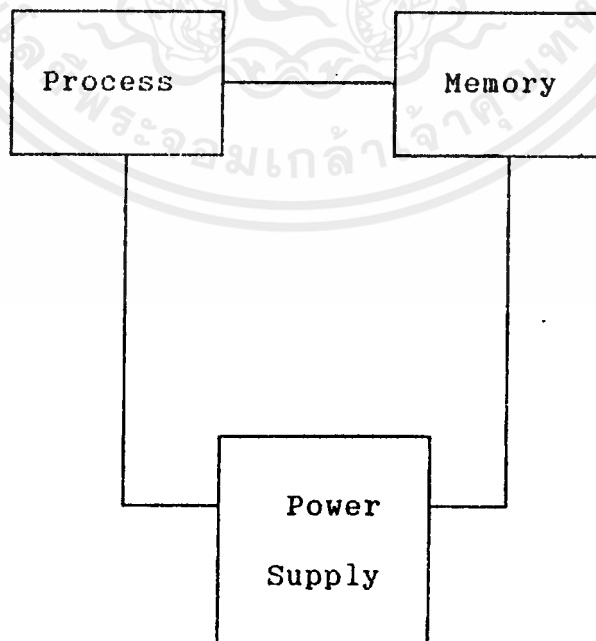
2.2 หน่วยประมวลผลกลาง (CPU UNIT)

เป็นหน่วยที่ทำหน้าที่ประมวลผล โดยจะตรวจรับข้อมูล Input จากอุปกรณ์ตรวจวัด (Sensor Device) แล้วทำการประมวลผลตามโปรแกรม เงื่อนไขที่ได้กำหนดไว้ในหน่วยความจำ จากนั้นก็ส่งสัญญาณ Output ออกไปเพื่อทำการควบคุม Process ต่าง ๆ

หน่วยประมวลผลกลางนี้ประกอบด้วย

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

รูปที่ 2.2 แสดงโครงสร้างของหน่วยประมวลผลกลาง โดยหน่วยประมวลผลทำหน้าที่ควบคุมการทำงานทั้งหมดของเครื่อง นำข้อมูลมาจากหน่วยความจำ มาปฏิบัติเพื่อควบคุมอุปกรณ์ภายนอกผ่านหน่วย Input- Output ในขณะที่หน่วยจ่ายพลังงานทำหน้าที่จ่ายแรงดันไฟฟ้าให้กับหน่วยประมวลผลและหน่วยความจำ



รูปที่ 2.2 โครงสร้างของหน่วยประมวลผลกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 หน่วยประมวลผล (Processor)

ทำหน้าที่ตรวจสอบสภาวะต่างๆในหน่วยความจำ และประมวลผลตามโปรแกรมต่างๆที่สร้างขึ้น (User Program) ควบคุมการติดต่อรับส่งระหว่าง CPU กับ Input-Output Unit และ Input-Output Unit กับอุปกรณ์ภายนอก ติดต่อกับผู้ใช้และอุปกรณ์ร่วม ตรวจสอบสภาพการทำงานของ PLC โดยโปรแกรมบริหารระบบ (Operating System) เป็นผู้ควบคุมอีกทีหนึ่ง

2.2.2 หน่วยความจำ (Memory)

เป็นองค์ประกอบที่สำคัญของระบบ เพราะใช้เป็นที่เก็บโปรแกรม และข้อมูลขนาดของหน่วยความจำ จะเป็นสิ่งกำหนดความสามารถของระบบ ปกติ มักจะมีขนาดวัดกันเป็น Step ของค่าสิ่ง ในโปรแกรมของระบบที่มีขนาดของหน่วยความจำมาก จะทำให้ผู้ใช้สามารถเขียนโปรแกรมที่มีความซับซ้อนได้มากขึ้น ซึ่งจะแบ่งหน่วยความจำออกเป็น 2 ส่วน คือ

1. หน่วยความจำระบบ (System Memory) เก็บโปรแกรมบริหารระบบ และข้อมูลของระบบ
2. หน่วยความจำผู้ใช้ (User Memory) เก็บโปรแกรมผู้ใช้ ข้อมูลของ Input-Output Unit และอุปกรณ์ภายใน หน่วยความจำที่นำมาใช้ใน PLC มี 2 ชนิดเช่นเดียวกับที่ใช้ Computer ทั่วไป คือ

1. Volatile

2. Nonvolatile

ข้อมูลในหน่วยความจำแบบ Volatile จะสูญหายหมดถ้าไม่มีกระแสไฟฟ้า (สามารถใช้ Battery Backup ได้) ส่วนข้อมูลภายในหน่วยความจำแบบ Nonvolatile จะยังคงอยู่ถึงแม้ไม่มีกระแสไฟฟ้า โปรแกรมบริหารระบบ (Supplyvisery Program or Operating System) และข้อมูลของระบบจะเก็บอยู่ในหน่วยความจำแบบ Nonvolatile เช่น PROM EPROM ในขณะที่ User Program อาจเก็บอยู่ในหน่วยความจำแบบใดก็ได้ถ้าเป็นโปรแกรมที่ไม่ต้องการเปลี่ยนแปลงแก้ไขอีกก็จะเก็บในลักษณะโปรแกรมบริหารระบบ แต่ถ้ามีความจำเป็นต้องการตรวจสอบแก้ไขโปรแกรมนั้นอยู่ ก็จะเก็บโปรแกรมไว้ในประ

เมท RAM ควบคุมอาจมีหน่วยจ่ายพลังงานสำรองใช้ความระมัดระวังไฟฟ้าดับ ส่วนข้อมูลที่เป็นข้อมูลของ Input-Output Unit และอุปกรณ์ภายใน (Data Memory) จะถูกเก็บอยู่ในหน่วยความจำแบบ RAM เท่านั้น เพราะเป็นข้อมูลที่มีการประมวลผลเปลี่ยนแปลงตลอดเวลา โดยอาจจะมีหน่วยจ่ายพลังงานสำรองหรือไม่ก็ได้

2.3 INPUT-OUTPUT UNIT

เป็นตัวกลางเชื่อมการติดต่อระหว่าง CPU กับอุปกรณ์ภายนอกโดยมีหน่วย Input ทำหน้าที่รับค่าสถานะจากอุปกรณ์ตรวจวัด และส่งผลที่คำนวณได้ไปหน่วย Output

2.3.1 INPUT UNIT

ทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอกที่เป็นสวิทซ์และตัวตรวจจับต่างๆ แล้วแปลงชนิดของสัญญาณขาเข้าดังกล่าวไม่ว่าจะเป็น AC หรือ DC ให้เป็นสัญญาณที่เหมาะสมเพื่อส่งเข้าไปให้แก่หน่วยประมวลผลกลาง ดังนั้นในการเลือกใช้ประเภทของ Input นั้นผู้ใช้จะต้องเลือกผู้ใช้ให้เหมาะสมและถูกต้องตามประเภทการใช้งานด้วย ไม่เช่นนั้นอาจจะเกิดความเสียหายขึ้นได้ ปกติ Input ที่ดี ควรจะมีหน้าที่ดังนี้

1. เปลี่ยนแปลงระดับของสัญญาณเข้า ให้เป็นระดับสัญญาณที่เหมาะสมกับ PLC
2. การส่งสัญญาณระหว่างหน่วยประมวลผล กับ Input จะติดกันด้วยลำแสงโดยอาศัยอุปกรณ์ที่เรียกว่า Photo Transister ทั้งนี้เพื่อแยกสัญญาณ (Isolate) ออกจากกันเป็นการป้องกันไม่ให้หน่วยประมวลผลได้รับความเสียหาย เมื่อหน่วย Input เกิดลัดวงจร

3. ไม่มีการลื่นของหน้าสัมผัส (Contract Chattering)

2.3.2 OUTPUT UNIT

ทำหน้าที่รับค่าสถานะที่ได้จากการประมวลของ CPU เพื่อนำค่าสถานะเหล่านี้ไปควบคุมอุปกรณ์ภายนอกเช่น Relay Solinoid Lamp แสดงสถานะเป็นต้น นอกจากนี้ยังทำหน้าที่แยกสัญญาณของ CPU ออกจากอุปกรณ์ Output โดยปกติแล้ว Output Unit มีความสามารถในการขับโหลดด้วยกระแสไฟ 1-2 Amp ในกรณีที่โหลดต้องการกระแสมากกว่านี้ผู้ใช้จะต้องนำไปต่อเข้ากับอุปกรณ์ขับหรือขยายกำลังอีกทีหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output ของ PLC เป็นแบบที่ใช้กับไฟกระแสตรงซึ่งมีอยู่ 2 ชนิดด้วยกันคือ Relay และ Transister

2.4 INOUT-OUTPUT PROCESSING

การรับข้อมูลของหน่วย Input-output มาประมวลผลตามโปรแกรมของผู้ใช้และส่งข้อมูลสุดท้ายที่ได้จากการประมวลผลไปที่ Output Unit เรียกว่า "SCANNING" 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 (โดยทั่วไปการหน่วงเวลาใช้เวลาประมาณ 3 ms. เพื่อป้องกันการเกิดสัญญาณ Bounce ที่เกิดจากการกดสวิตซ์และสัญญาณรบกวนอื่นๆที่จะเข้าไปใน PLC) Output จะขับอุปกรณ์โดยตรงเมื่อได้ปฏิบัติตามโปรแกรม และจะ latch ค่าไว้ใน Input-Output Unit (ในส่วนของหน่วยความจำ) เพื่อรักษาสถานะของ Output เอาไว้จนกว่าจะทำคำสั่งใหม่เสร็จ

2.4.2 MASS INPUT-OUTPUT COPYING

PLC ที่มีขนาดใหญ่ I/O เป็น 100 ตัว ถ้าทำแบบวิธีแรก จะทำงานได้ช้ามาก เพราะต้องใช้เวลาถึง 3 ms. ต่อคำสั่ง สำหรับการ Scan วิธีนี้จะ Scan ทีละโปรแกรมเลยคือมีค่าเท่ากับ 1 Scan Time นั้นเอง ซึ่งจะใช้เวลาประมาณ 1-100 ms. ขึ้นอยู่กับขนาดความยาวของโปรแกรมของผู้ใช้ และชนิดของจำนวนของ I/O Unit

การ Scan จะเริ่มด้วยการรับค่าสภาวะ ของอุปกรณ์ภายนอกจาก I/O Unit มาเก็บไว้ในหน่วยความจำ (RAM) แล้วนำโปรแกรมควบคุมที่ผู้ใช้เขียนขึ้นมาทีละคำสั่ง ซึ่งขึ้นอยู่กับค่าของ Input โดยเริ่มจากคำสั่งแรกจนคำสั่งสุดท้าย ซึ่งถือว่าเป็นสิ้นสุดโปรแกรมในหน่วย

ความจำ ถ้าการทำงานของโปรแกรมทำให้สถานะของ Output เปลี่ยนแปลง ผลดังกล่าวจะถูกไว้ที่ส่วน Output ของ I/O RAM และเมื่อจบโปรแกรมแล้ว I/O Unit จะส่งผลการเปลี่ยนแปลง Output จาก I/O RAM ไปที่ I/O Unit และสถานะดังกล่าวจะถูก Latch ไว้ก่อน จนกว่าจะทำการ Scan รอบใหม่เสร็จ ซึ่งค่า Input ที่เปลี่ยนแปลงขณะที่ปฏิบัติตามโปรแกรมนั้นอยู่ จะถูกเก็บไว้ใน RAM ก่อน ซึ่งอาจทำให้เกิดความผิดพลาดได้แต่ก็น้อยมาก เพราะช่วงเวลาการปฏิบัติ (Executed) น้อยมาก

ช่วงเวลา Scan Time ของ PLC จะเป็นตัวบอกถึงความสามารถของ PLC ในการตรวจสอบ และตอบสนองต่อการเปลี่ยนแปลงของสถานะของอุปกรณ์ภายนอก และเครื่องจักรหรือขบวนการผลิตว่ามากน้อยเพียงใด เช่น PLC ที่ช่วงเวลา Scan 10 ms. ย่อมไม่สามารถรับสถานะที่แท้จริงของอุปกรณ์ที่มีการเปลี่ยนแปลงทุก 4 ms. ถ้าใช้ PLC ดังกล่าวจะทำให้การควบคุมผิดพลาดหมด

PLC ที่ถูกจำลองขึ้นนี้จะใช้การ Scan แบบ Mass Input-Output Copying

2.5 หน่วยป้อนโปรแกรม (PROGRAMMING UNIT)

ทำหน้าที่ป้อนโปรแกรมลงในหน่วยความจำ ซึ่งโปรแกรมบริหารระบบจะนำไปประมวลผล เพื่อสร้างโปรแกรมควบคุมอุปกรณ์ภายนอกอีกที การแสดงค่าสถานะของ I/O ตัวตั้งเวลา ตัวนับเวลาต่างๆ ก็สามารถดำเนินการได้ด้วยหน่วยป้อนโปรแกรมนี้นี้

และ

2.6 หน่วยจ่ายพลังงาน (POWER SUPPLY)

ทำหน้าที่จ่ายกระแสไฟฟ้าให้ส่วนต่างๆของ PLC คือ PROCESSOR UNIT , MEMORY UNIT I/O UNIT โดยรักษาระดับแรงดันไฟฟ้าไว้ให้คงที่

**** ความน่าเชื่อถือของ PLC ขึ้นอยู่กับการทำงานของหน่วยจ่ายพลังงานและ I/O PROTECTION ****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ภาษาคำสั่งของ PLC

ภาษาที่ใช้ในการเขียนโปรแกรมคำสั่งของ PLC มีอยู่หลายภาษา แต่ภาษาที่นิยมใช้มีอยู่ 2 ภาษา คือ

1. ภาษาแลดเดอร์ไดอะแกรม (Ladder Diagram Language)
2. ภาษาคำสั่งบูลีน (Boolean Language)

3.1 ภาษาแลดเดอร์ไดอะแกรม

ภาษาแลดเดอร์ไดอะแกรม เป็นภาษาสัญลักษณ์ที่ใช้แทนรีเลย์ไดอะแกรม (Relay Diagram) ซึ่งเป็นไดอะแกรมวงจรลอจิกที่ใช้งานจริง ดังนั้นการแปลงรีเลย์ไดอะแกรมเป็นแลดเดอร์ไดอะแกรมจึงแปลงได้โดยตรง และอำนวยความสะดวกแก่ผู้ใช้เป็นอย่างมาก แต่ถ้าเป็นภาษาอื่นๆ การแปลงรีเลย์ไดอะแกรมให้เป็นภาษาที่ PLC เข้าใจจะต้องมีขั้นตอนเพิ่มขึ้นอีก 1 ขั้นตอน คือแปลงให้รีเลย์ไดอะแกรมเป็นแลดเดอร์ไดอะแกรม แล้วจึงแปลงเป็นภาษา PLC อีกครั้งหนึ่ง PLC ที่ใช้ภาษาแลดเดอร์จะต้องมีหน่วยป้อนโปรแกรมที่มีจอแสดงผลร่วมอยู่ด้วย เพื่อใช้แสดงรูปสัญลักษณ์ของแลดเดอร์ไดอะแกรม หน่วยป้อนโปรแกรมดังกล่าวนี้จึงมีขนาดใหญ่และราคาค่อนข้างสูง เพราะโครงสร้างภายในที่ซับซ้อน

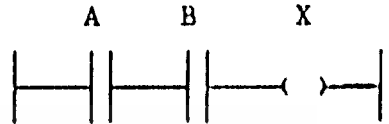
3.2 ความสัมพันธ์ระหว่างแลดเดอร์ไดอะแกรมและฟังก์ชันต่างๆ

PLC จะถูกนำมาใช้งาน แทนระบบที่ใช้รีเลย์ในการควบคุม ซึ่งลักษณะของวงจรควบคุมเป็นแบบฟังก์ชันลอจิกพื้นฐาน ดังนั้นการใช้แลดเดอร์ไดอะแกรมแทนวงจรควบคุมนี้ก็เป็น การแทนฟังก์ชันดังกล่าวนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

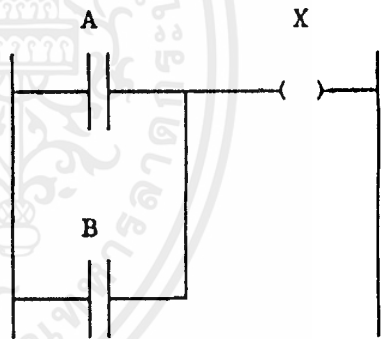
ฟังก์ชันลอจิก AND

แลตเตอร์ไคอะแกรม



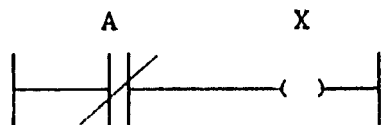
ฟังก์ชันลอจิก OR

แลตเตอร์ไคอะแกรม



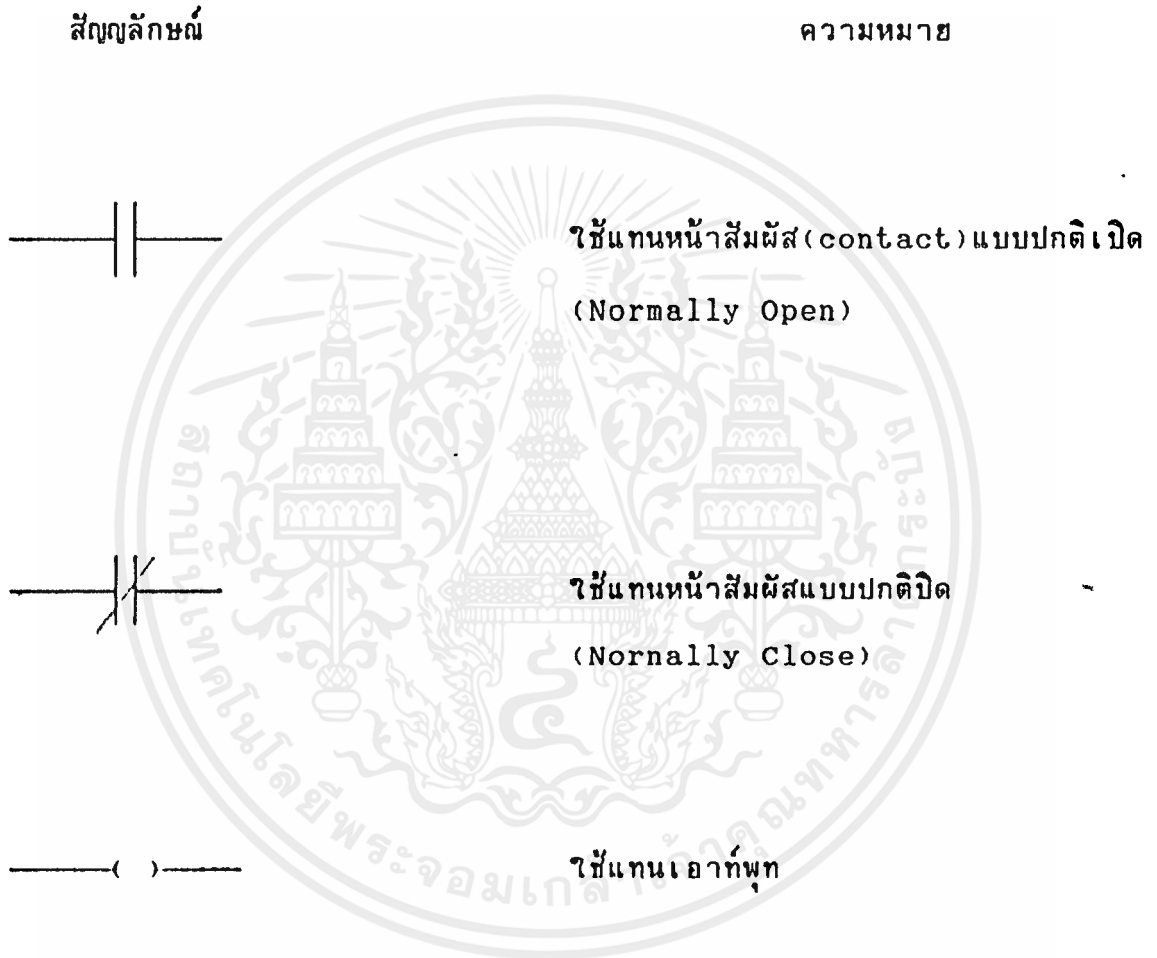
ฟังก์ชันลอจิก NOT

แลตเตอร์ไคอะแกรม



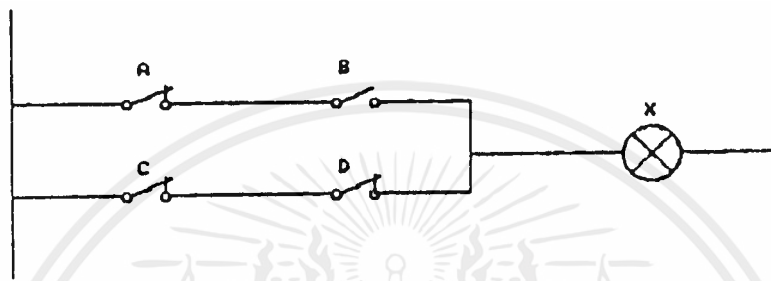
เอกสารนี้เป็นรูปที่ 3.1 แสดงความสัมพันธ์ระหว่างฟังก์ชันลอจิกและแลตเตอร์ไคอะแกรม การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสัมพันธ์ระหว่างฟังก์ชันลอจิกและแลตเตอร์ไดอะแกรมข้างต้น จะพบว่า สัญลักษณ์พื้นฐานที่ใช้ในแลตเตอร์ไดอะแกรม



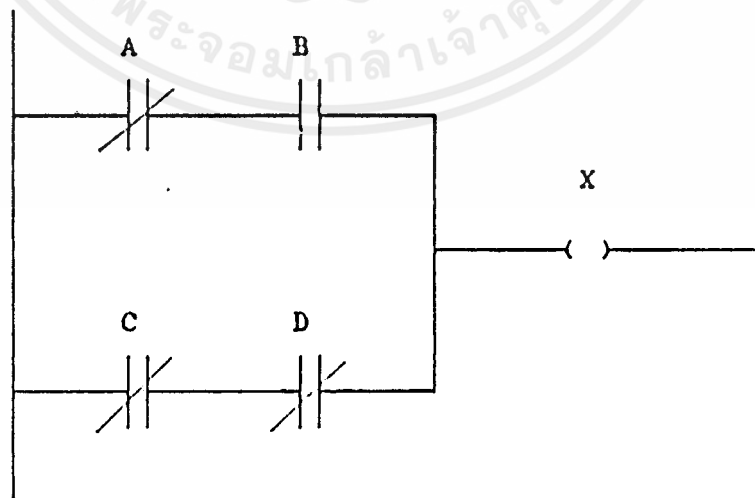
รูปที่ 3.2 แสดงสัญลักษณ์พื้นฐานที่ใช้ในแลตเตอร์ไดอะแกรม

ต่อไปนี้เป็นตัวอย่างที่ใช้แลตเตอร์ไคอะแกรมแทนวงจรรีเลย์



รูปที่ 3.3 วงจรรีเลย์

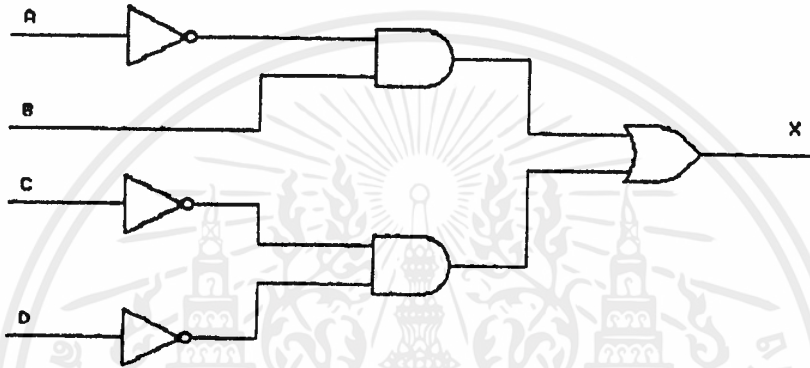
วงจรรีเลย์ในรูปที่ 3.3 สามารถแทนด้วยแลตเตอร์ไคอะแกรมได้ดังนี้



รูปที่ 3.4 แลตเตอร์ไคอะแกรมของรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และวงจรรีเลย์ในรูปที่ 3.3 สามารถแทนได้ด้วยวงจรถลอจิกดังรูปต่อไปนี้

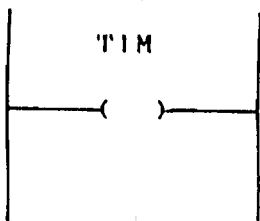


รูปที่ 3.5 วงจรถลอจิกของรูปที่ 3.3

นอกจากสัญลักษณ์พื้นฐานของแลตเตอร์ไดอะแกรมแล้ว ยังมีสัญลักษณ์อื่นๆ ที่ใช้แทนฟังก์ชันบางอย่างที่มีใช้ในวงจรควบคุมรีเลย์ อันได้แก่ ตัวตั้งเวลา ตัวนับ เป็นต้น

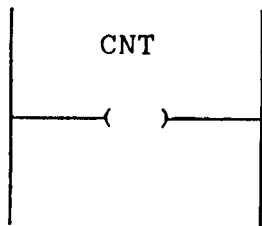
สัญลักษณ์

ความหมาย



ใช้แทนตัวตั้งเวลาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ใช้แทนตัวนับ

รูปที่ 3.6 แสดงสัญลักษณ์ของตัวตั้งเวลา ตัวนับเวลา

3.3 ภาษาคำสั่งบูลีน

ภาษาคำสั่งบูลีนเป็นภาษาที่มีคำสั่งพื้นฐานทางลอจิก คือ แอนด์ ออร์ และน็อท และคำสั่งอื่นๆ อีกที่สอดคล้องกับวีเลย์ไดอะแกรม ลักษณะของภาษาคำสั่งบูลีนจัดว่าเป็นภาษาที่เข้าใจง่ายพอๆ กับแลดเดอร์ไดอะแกรม และเหมาะกับ PLC ที่มีขนาดเล็ก สำหรับปรวิญานิพนธ์ฉบับนี้ ซึ่งจัดการจำลอง PLC ลงบนเครื่องคอมพิวเตอร์ PC โดยการ key ข้อมูลด้วยภาษาคำสั่งบูลีน และจะแสดงผลในรูปแลดเดอร์ไดอะแกรม ภาษาคำสั่งบูลีนที่ใช้ในปรวิญานิพนธ์ฉบับนี้ จะประกอบด้วย คำสั่ง ดังต่อไปนี้ คือ คำสั่ง LOAD(LD) AND OR IS RESET(RST) SET END X Y TIMER(TIM) COUNTER(CNT) 0-9

บทที่ 4

ชุดคำสั่งใช้งาน

4.1 จำนวนอินพุท เอาท์พุท ตัวตั้งเวลา และตัวนับ

4.1.1 อินพุท (Input)

มีจำนวนอินพุททั้งสิ้น 32 อินพุท เรียงลำดับตั้งแต่ X0 จนถึง X31 โดยสามารถป้อนอินพุทโดยผ่านทาง keyboard

4.1.2 เอาท์พุท (Output)

มีจำนวนเอาท์พุททั้งสิ้น 32 เอาท์พุท เรียงลำดับตั้งแต่ Y0 ถึง Y31 โดยสามารถแสดงเอาท์พุทออกทางจอภาพ

*** เอาท์พุทที่ยังไม่ได้นำมาใช้งาน เราสามารถนำเอาท์พุทเหล่านั้นมาใช้เป็น รีเลย์ภายใน เพื่อช่วยในการเขียนโปรแกรมได้

4.1.3 ตัวตั้งเวลา (Timer, High Speed Timer)

มีตัวตั้งเวลาให้ใช้งานทั้งสิ้น 14 ตัว สำหรับฐานเวลา 0.1 วินาที (TIM: โดยเรียงลำดับตั้งแต่ TIM01 ถึง TIM09 ซึ่ง TIM แต่ละตัว จะมีค่าสูงสุดของเวลา คือ 999.9) มี 10 ตัว และสำหรับฐานเวลา 0.01 วินาที (HST: โดยเรียงลำดับตั้งแต่ HST10 ถึง HST13 ซึ่ง HST แต่ละตัวจะมีค่าสูงสุดของเวลา คือ 99.99) อีก 4 ตัว

4.1.4 ตัวนับ (Counter)

มีตัวนับอยู่ 10 ตัว เรียงลำดับตั้งแต่ CNT14 ถึง CNT23

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของบริษัทฯ เพื่อการศึกษาเท่านั้น มิให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 องค์ประกอบของโปรแกรมผู้ใช้

โปรแกรมผู้ใช้ 1 โปรแกรมประกอบด้วย "ชุดคำสั่ง" ตั้งแต่ 1 คำสั่งขึ้นไป คำสั่ง 1 คำสั่งประกอบขึ้นมาจากหลาย "คำสั่ง"

จะเห็นได้ว่า ส่วนที่ย่อยที่สุดสำหรับการเขียนโปรแกรมใน PLC คือคำสั่งคำสั่ง และทุกครั้งที่เรากดคีย์ ENTER เราจะได้คำสั่ง 1 คำสั่งเก็บอยู่ในหน่วยความจำ และคำสั่งหลายๆ คำสั่งที่ได้ปฏิบัติการกันแล้ว จะให้ผลลัพธ์บางอย่างเกิดขึ้น คือกลุ่มคำสั่งหรือโปรแกรมนั้นเอง

4.3 การจำแนกคีย์บอร์ด

สามารถจำแนกได้ 3 ชนิด คือ

1. คีย์สำหรับการใช้ PLC (Operating Keys)
2. คีย์คำสั่งต่างๆของ PLC (Instruction Keys)
3. คีย์โอเปอเรนด์ (Operand Keys)

4.3.1 คีย์สำหรับการใช้เครื่อง PLC

คีย์

การใช้งาน

CLEAR	[CLR]	ล้างหน้าจอ
INSERT	[INS]	แทรกคำสั่งใหม่
DELETE	[DEL]	ลบคำสั่ง
INCLEMENT	[INC]	เลื่อนไปยัง step ถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณี **DECLLEMENT** [DEC] แปลงเนื้อหาและต้อง **เลื่อนไปยัง step** ก่อนหน้าให้นำไปใช้

ENTER	[Enter]	นำคำสั่งต่างๆ เก็บในหน่วยความจำ แสดงสถานะการทำงานที่จอภาพ
F1		เรียก MENU File
F2		เรียก MENU Run

4.3.2 คีย์คำสั่งต่างๆของ PLC

	คีย์	การใช้งาน
LOAD	[LD]	นำค่าสถานะไปเก็บในความจำ
AND	[AND]	กระทำลอจิก AND
OR	[OR]	กระทำลอจิก OR
NOT	[NOT]	กระทำลอจิก NOT
IS	[IS]	ส่งค่าสถานะไปยังเอาต์พุต
SET	[SET]	set ค่าสถานะ output Y__ เป็น on
RESET	[RST]	clear ค่าใน counter และสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 คีย์โอเปอร์แวนด์

เป็นคีย์ที่ใช้กำหนด INPUT, OUTPUT, TIMER, COUNTER ดังรายละเอียดข้างล่างนี้

โอเปอร์แวนด์	คีย์	หมายเลขโอเปอร์แวนด์
INPUT	[X]	0-31
OUTPUT	[Y]	0-31
TIMER	[TIM]	0-9
HIGH SPEED TIMER	[HST]	10-13
COUNTER	[CNT]	0-9

4.4 คำสั่งเบื้องต้น

1. คำสั่ง LD

คำสั่งนี้มักเป็นคำสั่งแรก ของขบวนชุดคำสั่งแต่ละชุด และจะติดตามด้วยโอเปอร์แวนด์ตัวต่อไป เช่น LD X 5

2. คำสั่ง AND

คำสั่งนี้เป็นการต่ออนุกรมกันระหว่างผลลัพธ์ทางลอจิกของขบวนชุดคำสั่ง หรือโอเปอร์แวนด์ก่อนหน้า กับโอเปอร์แวนด์ของคำสั่งนี้

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คำสั่ง OR

คำสั่งนี้แสดงถึงการต่อขนานกันระหว่างผลลัพธ์ทางลอจิกของขบวนการชุดคำสั่ง หรือโอเพอร์แรนด์ก่อนหน้านี กับโอเพอร์แรนด์ของคำสั่งนี้

4. คำสั่ง IS

คำสั่งนี้ใช้กระตุ้นให้โอเพอร์แรนด์ของคำสั่ง กระทำฟังก์ชันตามชนิดของโอเพอร์แรนด์ของคำสั่งนี้ ต่อเมื่อผลลัพธ์ทางลอจิกของขบวนการชุดคำสั่งก่อนหน้านั้นเป็นจริง

จากคำสั่งเบื้องต้นต่างๆที่กล่าวมาแล้วนี้ การทำงานของคำสั่งจะอยู่ในลักษณะ MAKE FUNCTION แต่ในการใช้งานแล้ว จะมีการใช้งานในลักษณะ BRAKE FUNCTION อยู่ด้วย ดังนั้น จึงมีคำสั่งที่เหมาะสมสำหรับงานในลักษณะนี้ คำสั่งเหล่านี้ได้แก่ LD NOT AND NOT OR NOT ซึ่ง หมายถึง LD Inverse AND Inverse และ OR Inverse ตามลำดับ

ตัวอย่างการให้ค่าสิ่งเบ็ดเตล็ดต่างๆ

ตัวอย่างที่ 1



โปรแกรมคำสั่ง : LD X 01
AND X 02
IS Y 01
END

ความหมาย : เอาท์พุท 01 จะทำงาน (Active) เมื่ออินพุท 01 และอินพุท 02 ทำงานทั้งคู่

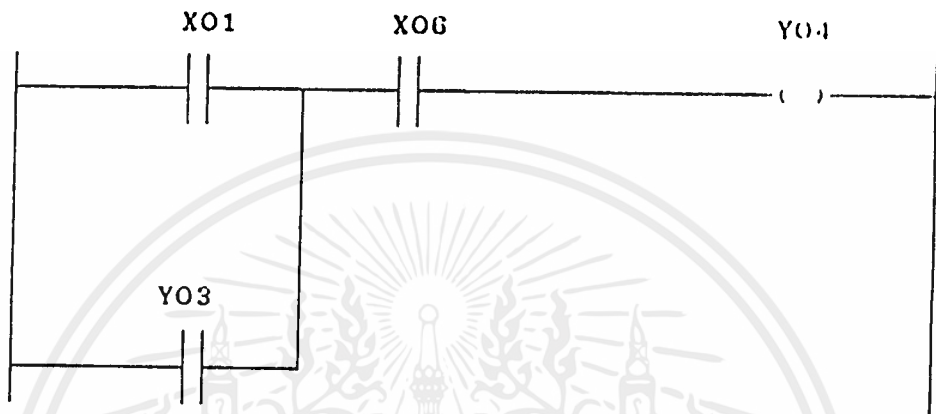
ตัวอย่างที่ 2



โปรแกรมคำสั่ง : LD X O1
AND NOT X O2
IS Y O1
END

ความหมาย : เอาท์พุท 01 จะทำงาน ก็ต่อเมื่ออินพุท 01 ทำงานและอินพุท 02 ไม่ทำงาน

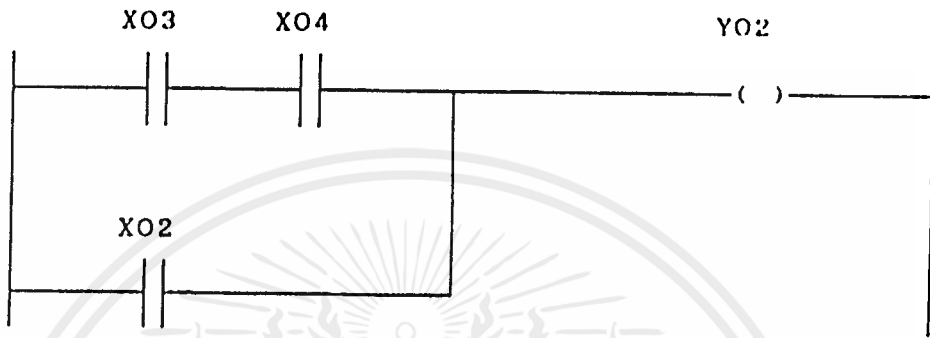
ตัวอย่างที่ ๒



```
โปรแกรมคำสั่ง : LD X 01
                  OR Y 03
                  AND NOT X 06
                  IS Y 04
                  END
```

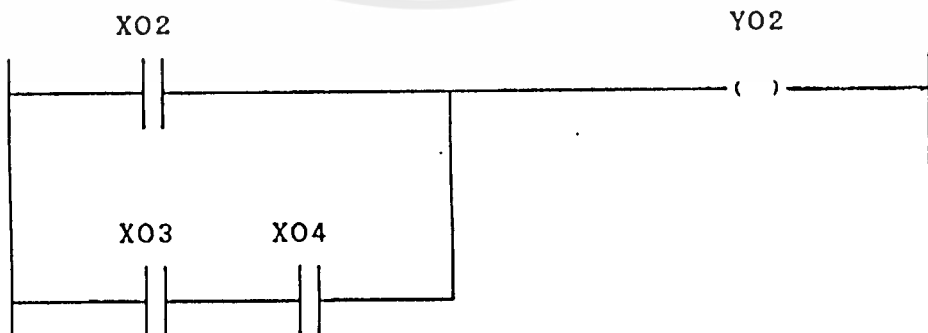
ความหมาย : เอาท์พุท 04 จะทำงานก็ต่อเมื่ออินพุท 01 หรือ เอาท์พุท 03 ทำงาน และอินพุท 06 ต้องไม่ทำงาน

ตัวอย่างที่ 4



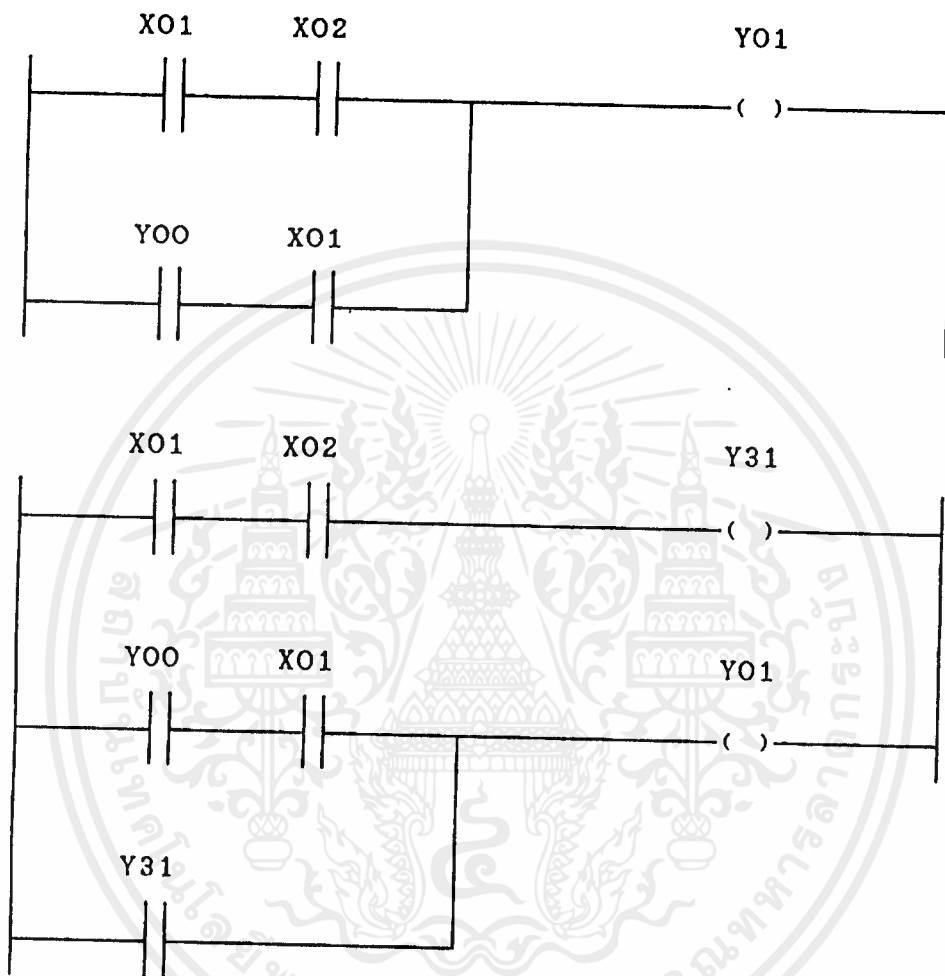
โปรแกรมคำสั่ง : LD X 03
AND X 04
OR X 02
IS Y 02
END

วาดแลตเตอร์ไดอะแกรมใหม่ได้ดังนี้



ความหมาย : เอาท์พุท 02 จะทำงานก็ต่อเมื่ออินพุท 02 หรือทั้งอินพุท 03 และอินพุท 04
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ทำงาน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

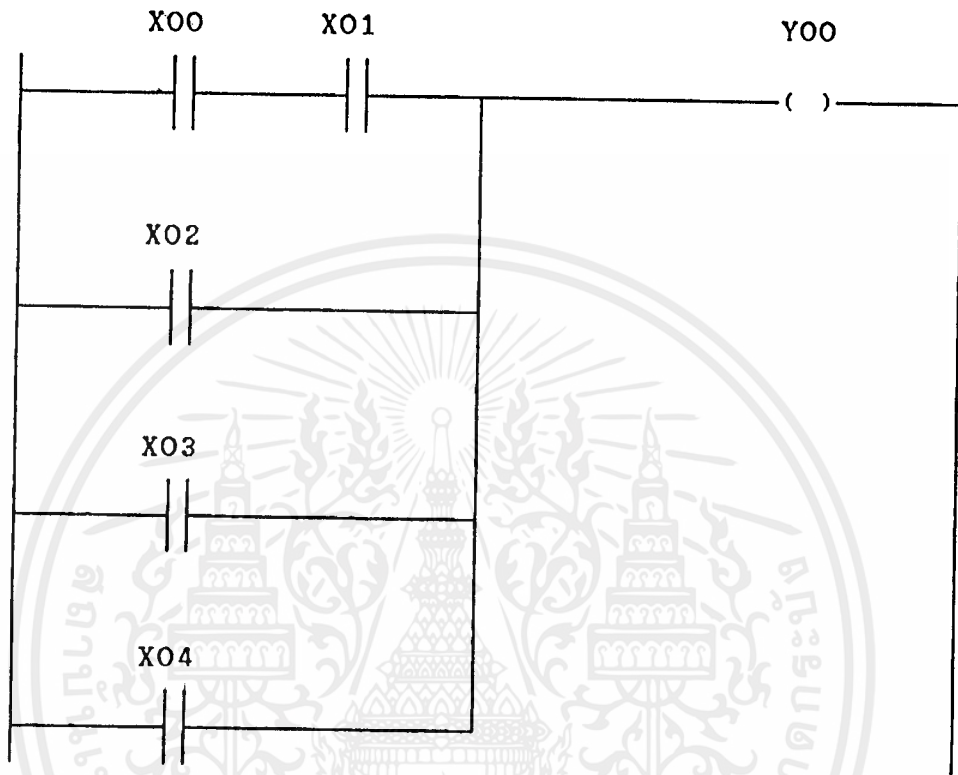
ตัวอย่างที่ 6



โปรแกรมคำสั่ง : LD X 01
 AND NOT X 02
 IS Y 31
 LD Y 00
 AND X 01
 OR Y 31
 IS Y 01
 END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวทางที่ 7



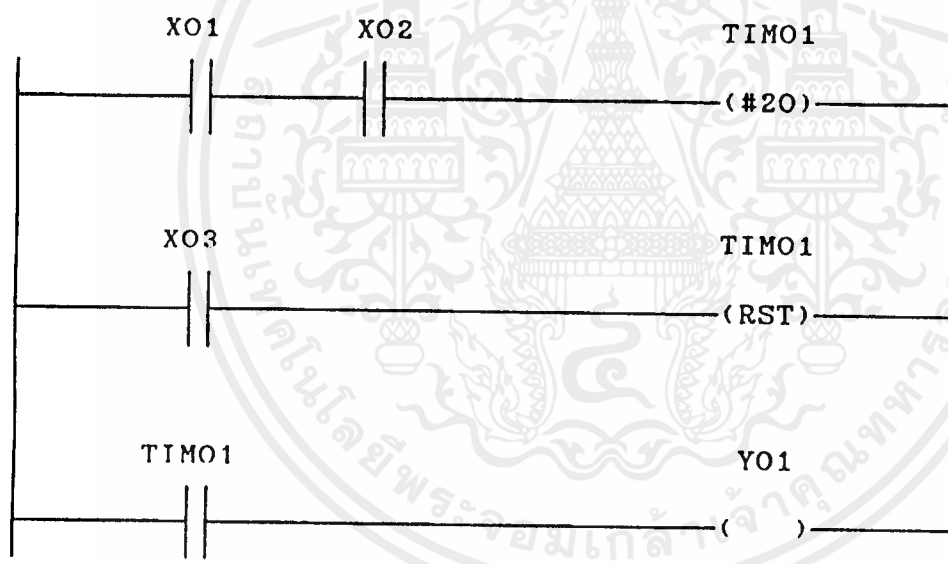
โปรแกรมคำสั่ง : LD X 00
AND X 01
OR X 02
OR X 03
OR X 04
IS Y 00
END

4.5 ฟังก์ชันการตั้งเวลา

TIMER : TIM

ในการควบคุมกระบวนการต่างๆ มักเกี่ยวข้องกับเรื่องของเวลาเสมอ PLC ที่จำลองขั้นนี้มี TIMER ซึ่งมีฐานเวลา 0.1 สามารถตั้งเวลาได้ตั้งแต่ 0.1 จนถึง 999.9 วินาที TIMER นี้มีทั้งหมด 10 ตัว เริ่มจาก TIM 00 จนถึง TIM 09

ตัวอย่างที่ 8



โปรแกรมคำสั่ง :	LD	X	01
	AND	X	02
	IS	TIM	01
		#	20
	LD	X	03
	RST	TIM	01
	LD	TIM	01
	IS	Y	00
	END		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมาย : ตัวตั้งเวลา O1 จะเริ่มทำงานต่อเมื่ออินพุต O1 และอินพุต O2 ทำงานทั้งคู่ และเมื่อครบเวลาที่ตั้งไว้ 2 วินาที เอาท์พุท O1 ก็จะทำงาน

HIGH SPEED TIMER : HST

เป็น TIMER ที่มีความเร็วสูงกว่าแบบ TIM โดยที่ HST จะมีฐานเวลา 0.01 นั่นคือมันสามารถตั้งเวลาได้ตั้งแต่ 0.01 จนถึง 99.99 วินาที แต่ถ้าต้องการให้ช่วงเวลาที่ตั้งมากกว่านี้ ให้ใช้ HST ร่วมกับ CNT หรือ TIM HST นี้มีทั้งหมด 4 ตัว เริ่มจาก HST10 จนถึง HST13 การเรียกใช้ HST มีลักษณะเดียวกับ TIM

4.6 ฟังก์ชันการนับ COUNTER : CNT

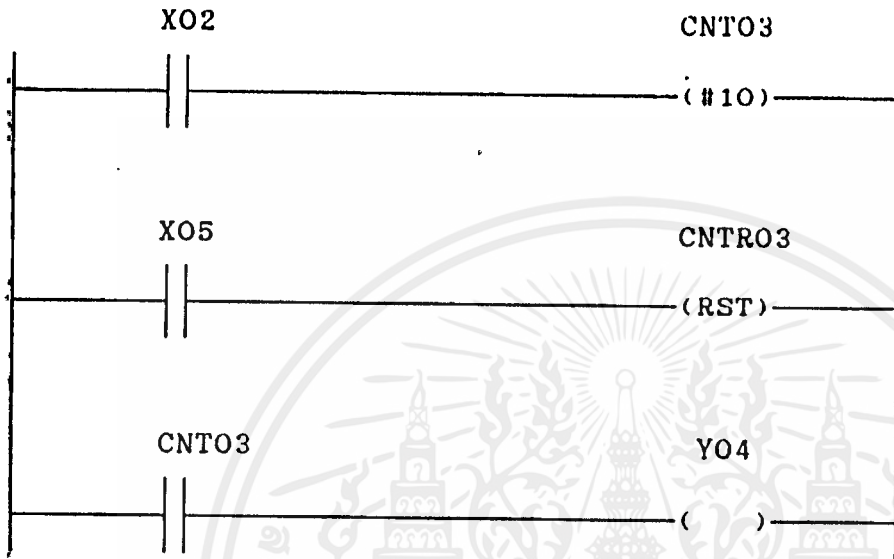
CNT จะทำการนับจำนวนพัลส์ที่ต่ออยู่กับขา CNT (นับเฉพาะขาขึ้น \uparrow)

PLC ที่จำลองขึ้นนี้ประกอบด้วยตัวนับ 0 ตัว เริ่มจาก CNT0 จนถึง CNT23 แต่ละตัวสามารถนับได้ตั้งแต่ 1 ถึง 9999 และในกรณีที่ไม่มี การเรียกใช้ TIM หรือ HST ที่หมายเลขใด ก็จะสามารถเรียก CNT หมายเลขนั้นมาใช้งานแทนได้

ตัวนับมี 2 อินพุต คือ

1. ถิ่นพุทนับ (Count : CNT) : ใช้นับจำนวนพัลส์
2. รีเซ็ตตัวนับ (Count Reset) : ใช้รีเซ็ตตัวนับ

ตัวอย่างที่ ๑



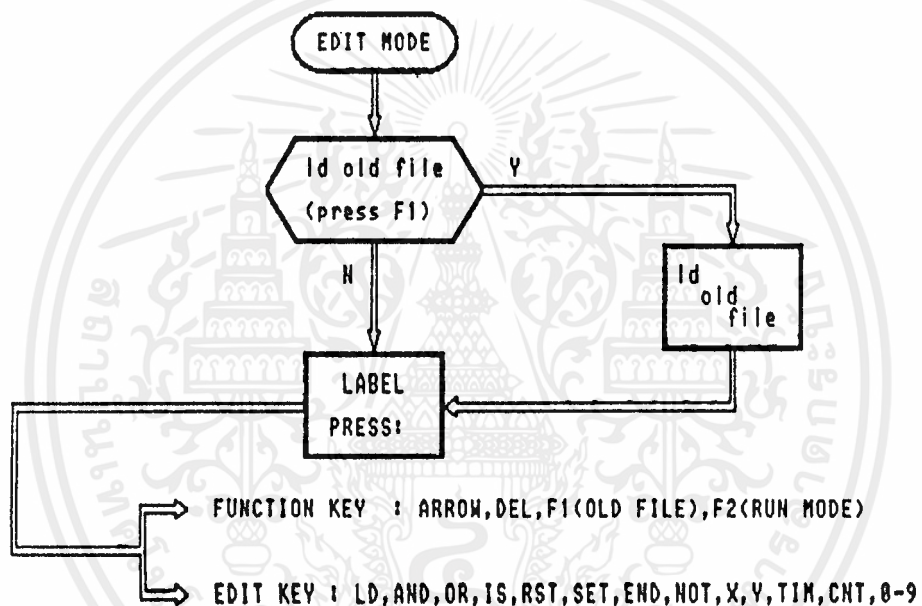
โปรแกรมคำสั่ง : LD X 02
IS CNT 03
10
LD X 05
RST CNT 03
LD CNT 03
IS Y 04
END

ความหมาย : เอาท์พุท 04 จะทำงานเมื่ออินพุท 02 ที่ป้อนเข้าตัวนับมีถึง 10 ครั้ง แต่เมื่อใดก็ตามที่อินพุท 05 ทำงาน ตัวนับจะถูกรีเซ็ตทันทีและเอาท์พุทก็จะถูกรีเซ็ตด้วย

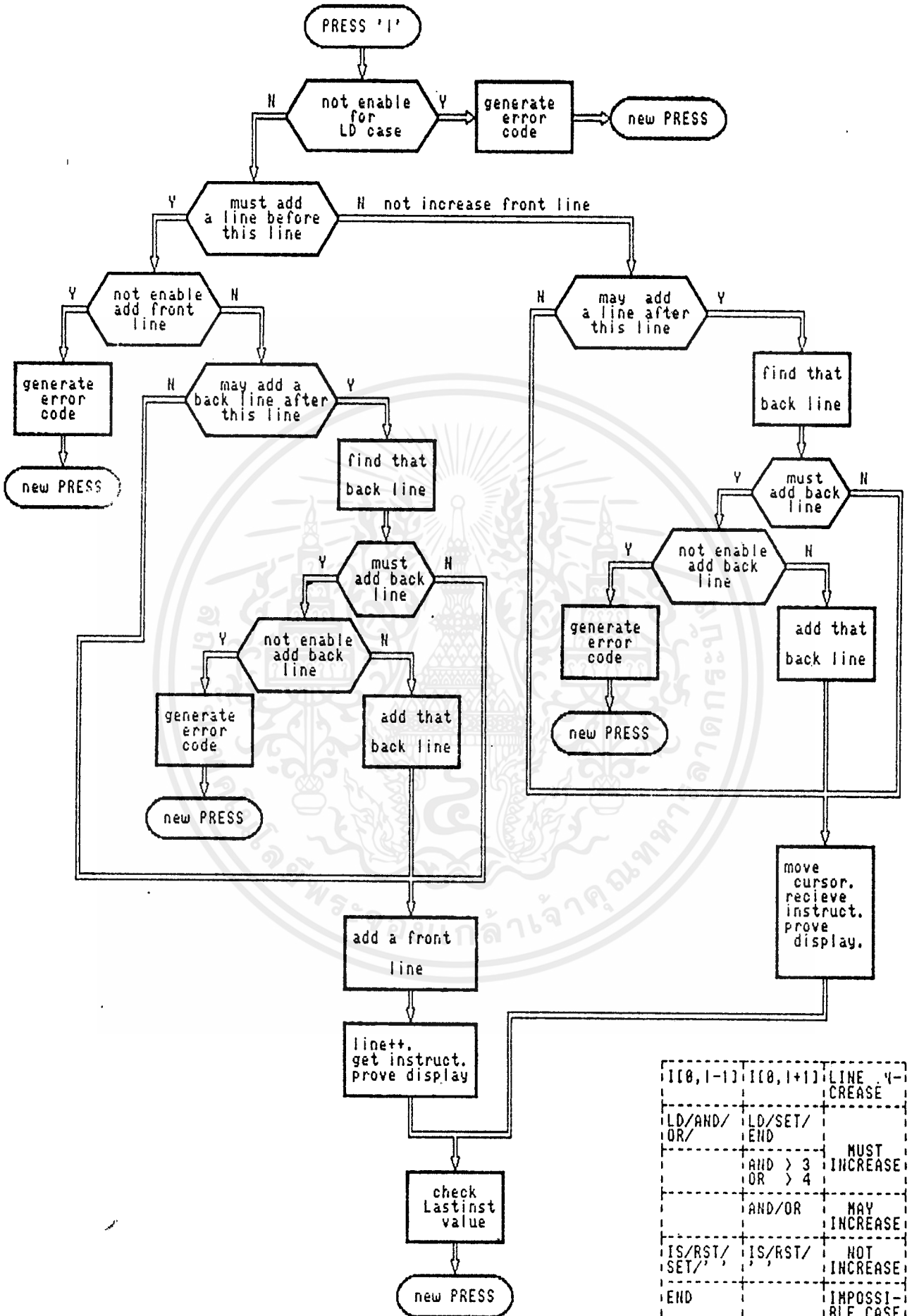
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

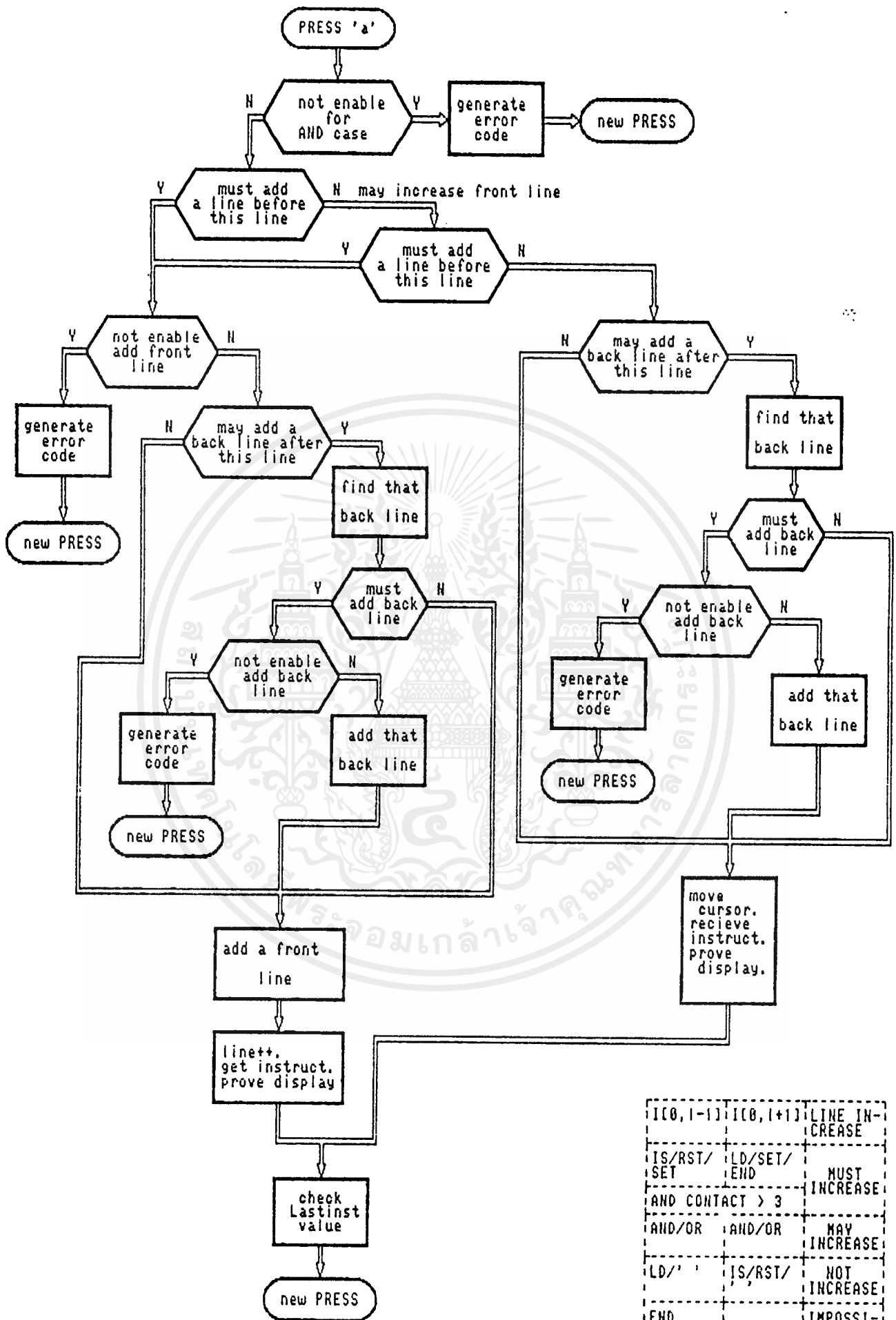


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



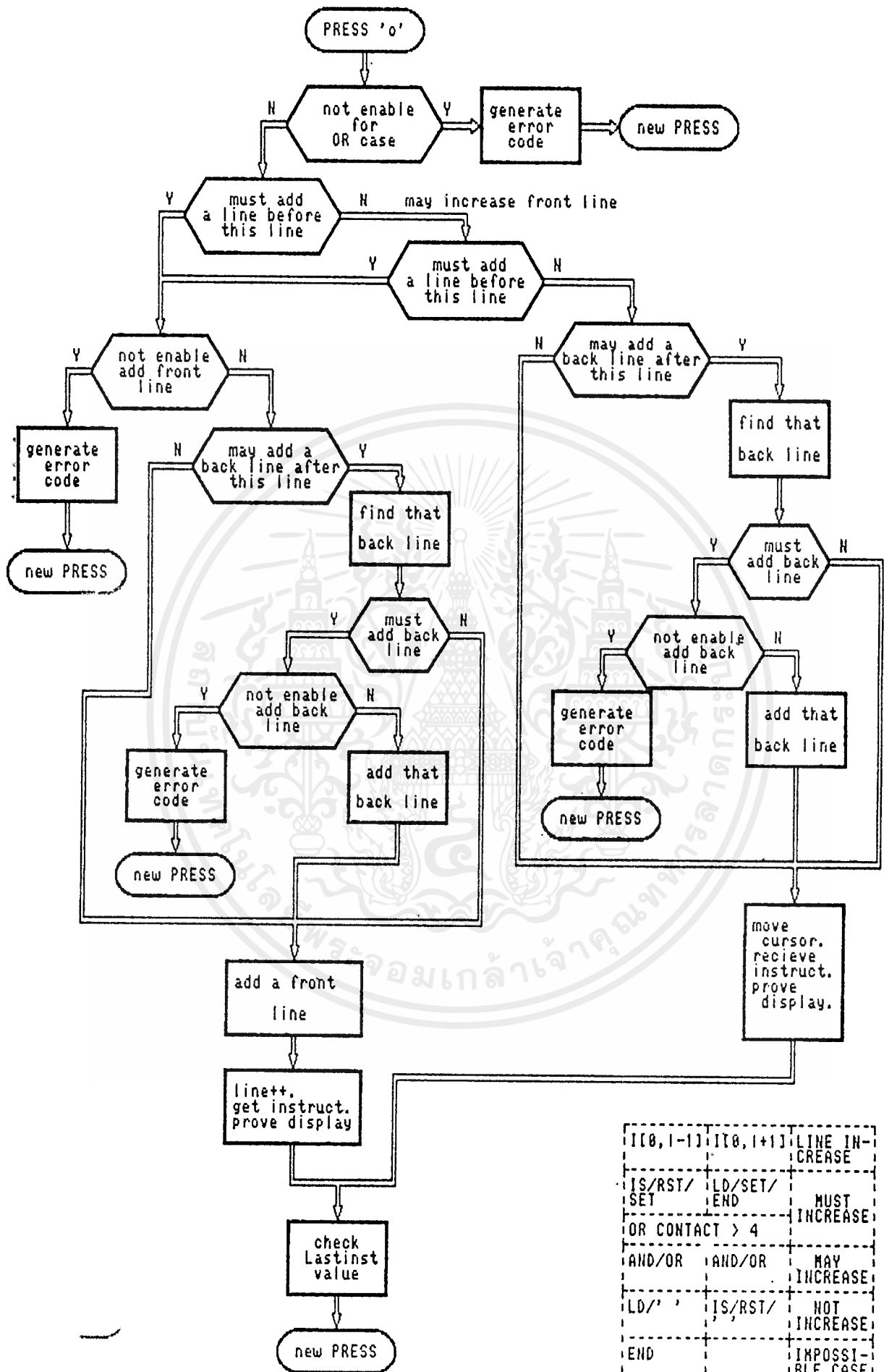
[I[0,1-1]	[I[0,1+1]	LINE 4-
		CREATE
LD/AND/OR/	LD/SET/END	
	AND > 3	MUST INCREASE
	OR > 4	
	AND/OR	MAY INCREASE
IS/RST/SET/	IS/RST/	NOT INCREASE
END		IMPOSSIBLE CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต. ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



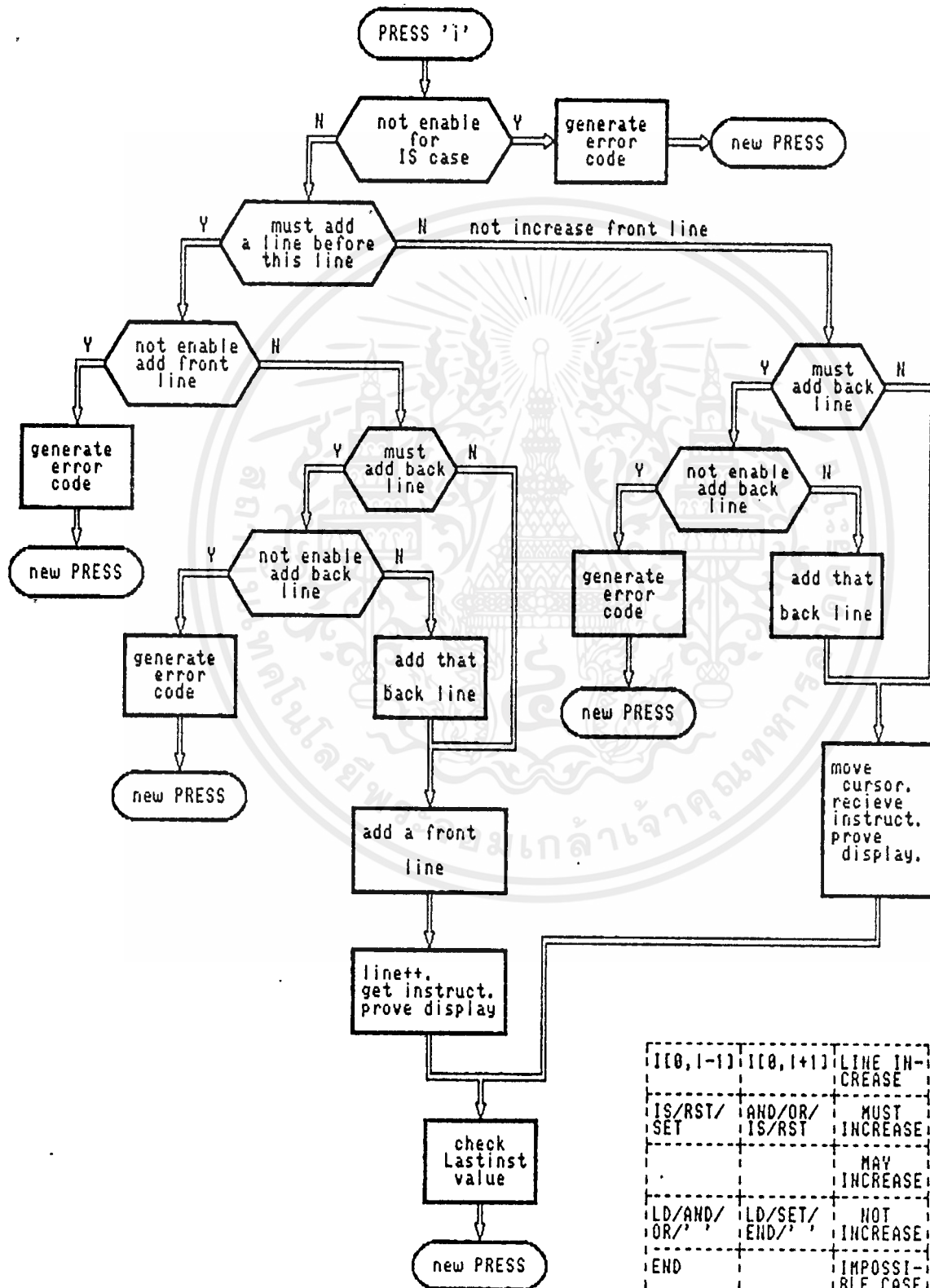
[0, 1-1]	[0, 1+1]	LINE INCREASE
IS/RST/SET	LD/SET/END	MUST INCREASE
AND CONTACT > 3	AND/OR	MAY INCREASE
LD/'	IS/RST/	NOT INCREASE
END		IMPOSSIBLE CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์อื่นใด
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



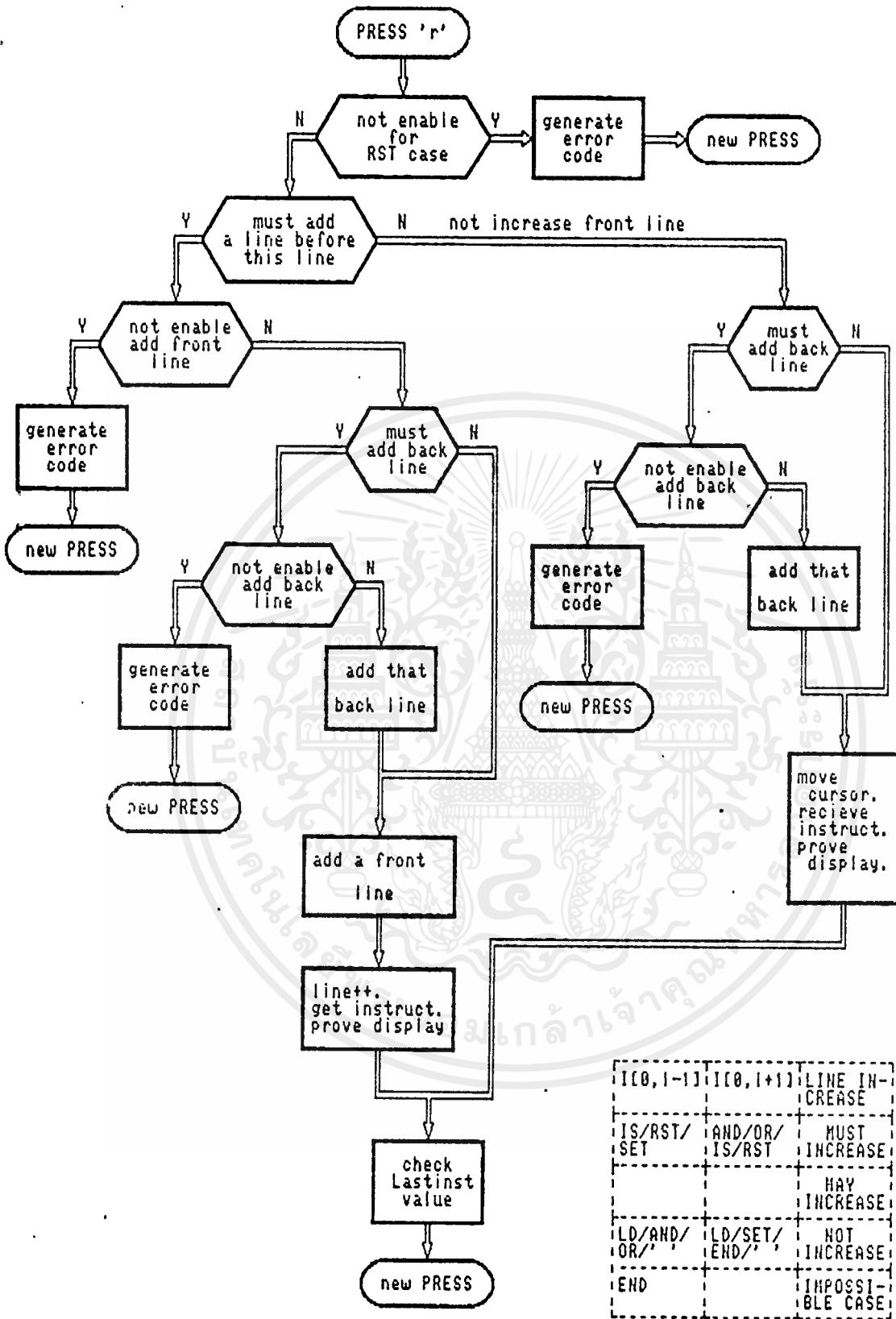
[0, 1-1]	[0, 1+1]	LINE INCREASE
IS/RST/SET	LD/SET/END	MUST INCREASE
OR CONTACT > 4		
AND/OR	AND/OR	MAY INCREASE
LD/'	IS/RST/	NOT INCREASE
END		IMPOSSIBLE CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

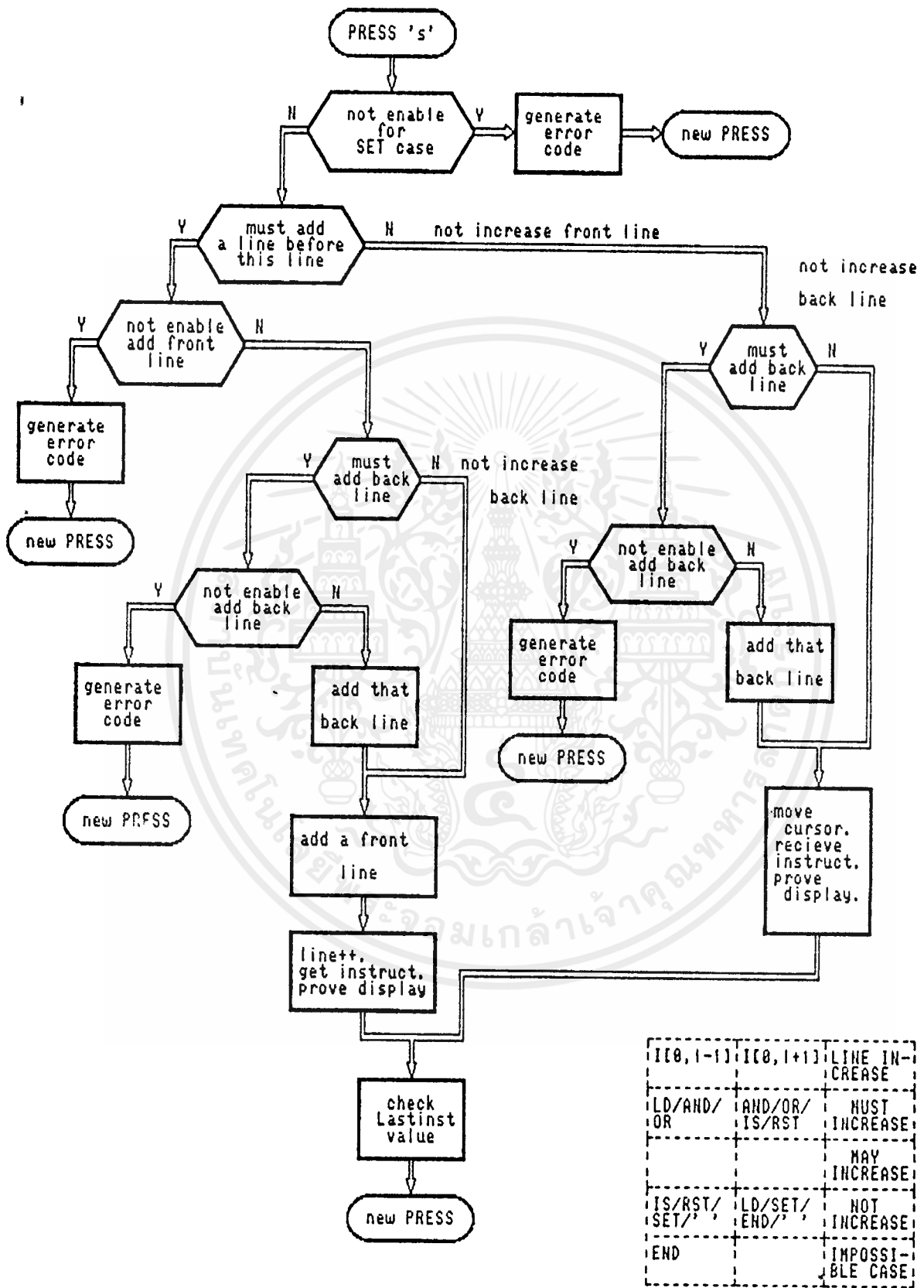


[0, 1-1]	[0, 1+1]	LINE INCREASE
IS/RST/SET	AND/OR/IS/RST	MUST INCREASE
		MAY INCREASE
LD/AND/OR/	LD/SET/END/	NOT INCREASE
END		IMPOSSIBLE CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

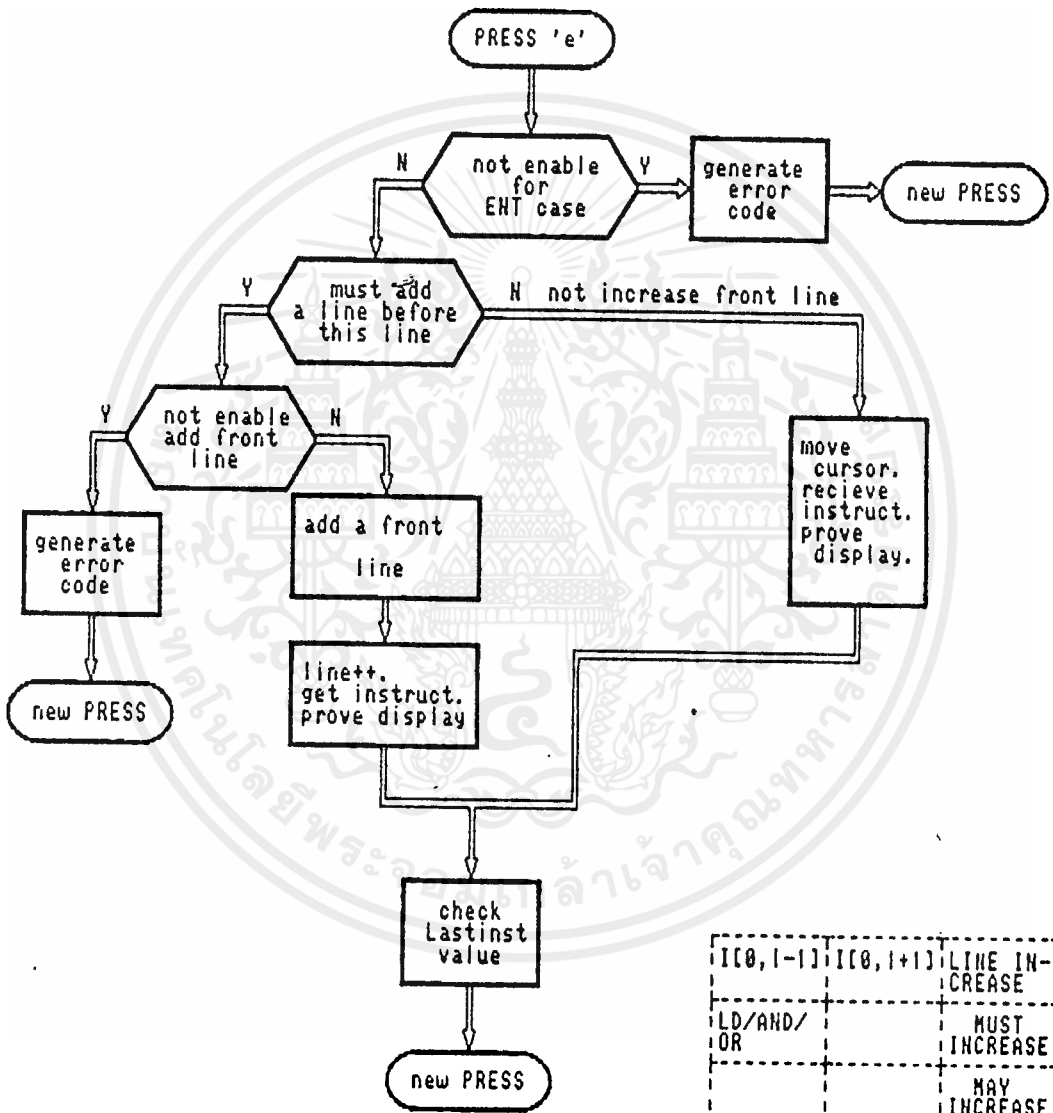


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



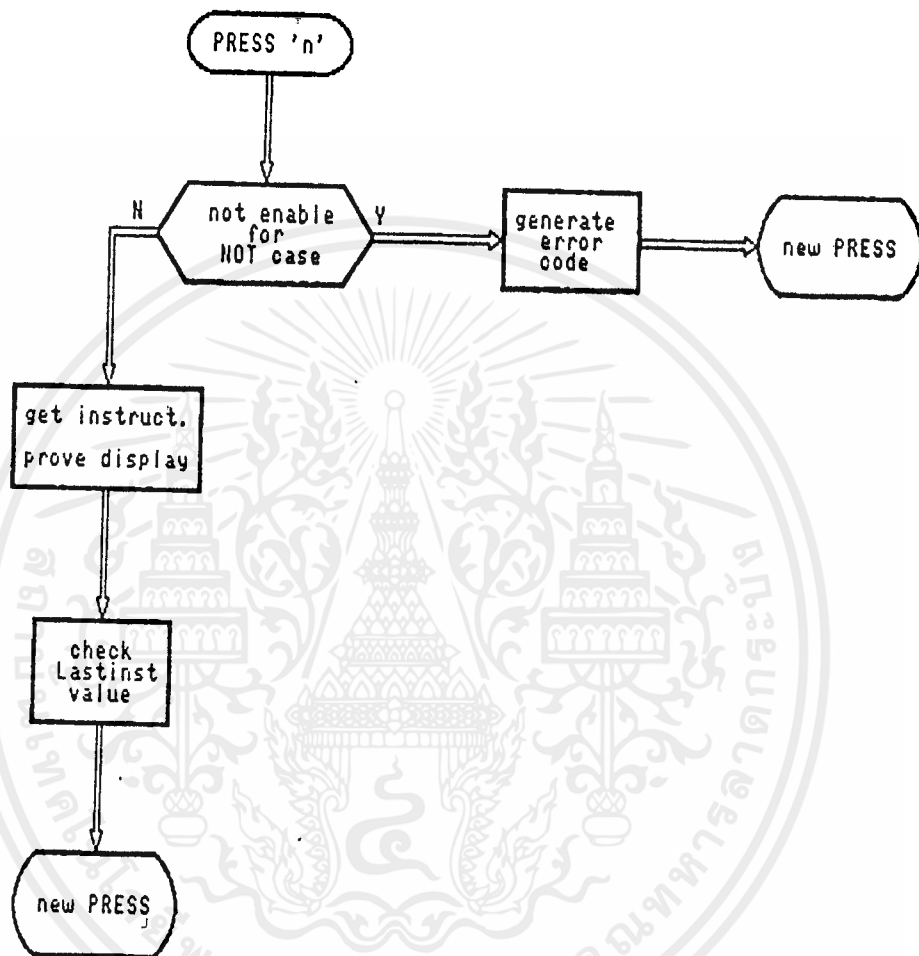
[0, 1-1]	[0, 1+1]	LINE INCREASE
LD/AND/OR	AND/OR/IS/RST	MUST INCREASE
		MAY INCREASE
IS/RST/SET/	LD/SET/END/	NOT INCREASE
END		IMPOSSIBLE CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



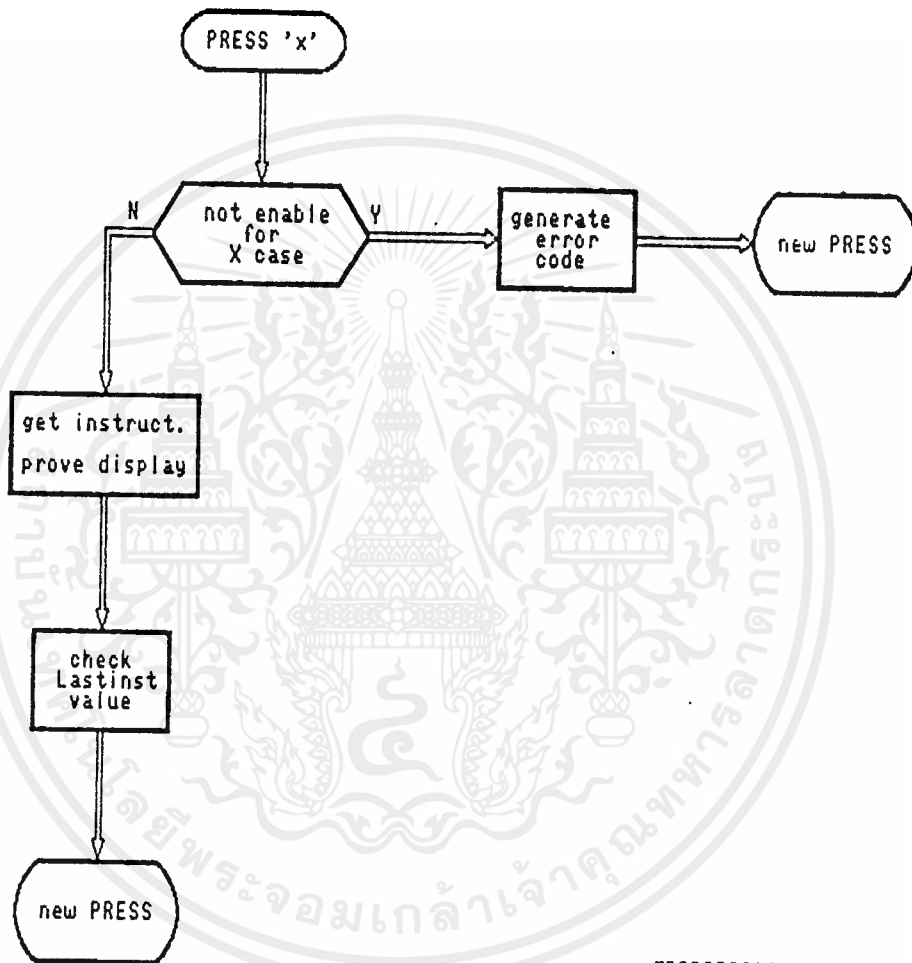
[0, 1-1]	[0, 1+1]	LINE INCREASE
LD/AND/OR		MUST INCREASE
		MAY INCREASE
IS/RST/SET/		NOT INCREASE
END	*ALL*	IMPOSSIBLE CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



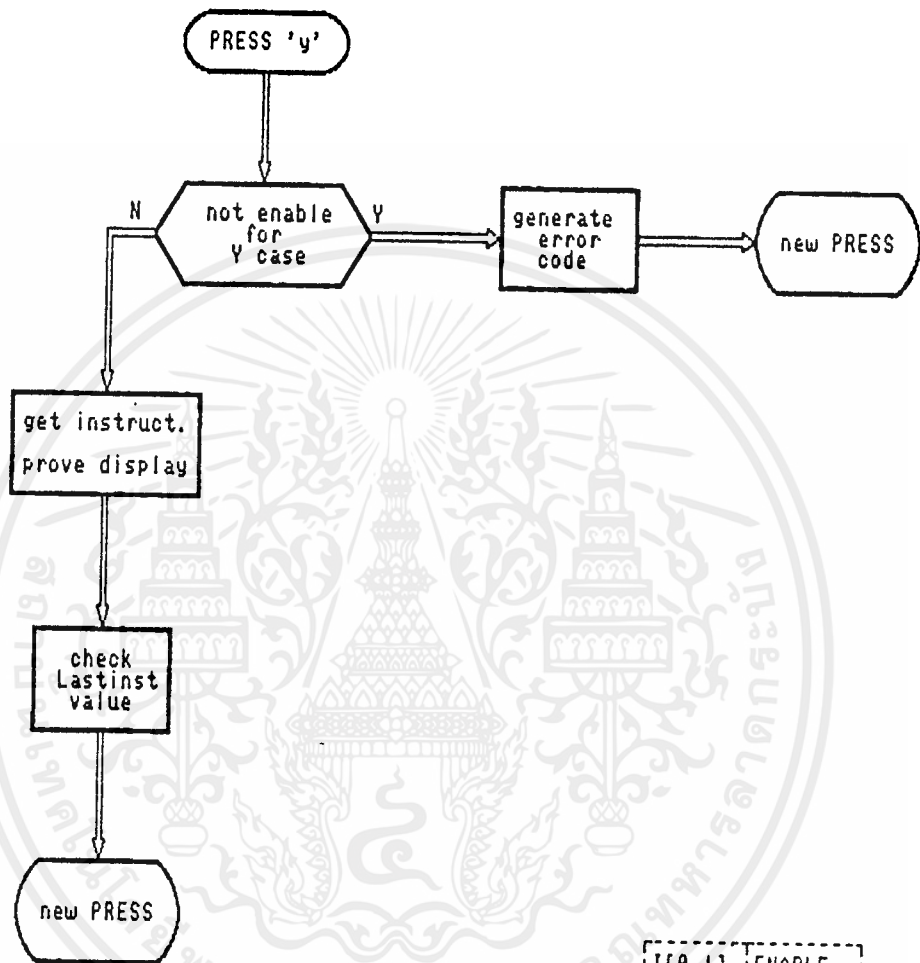
I[0,1]	ENABLE
	GET NOT
LD/AND/OR/IS	ENABLE
RST/SET/END/'	IMPOSSI-CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



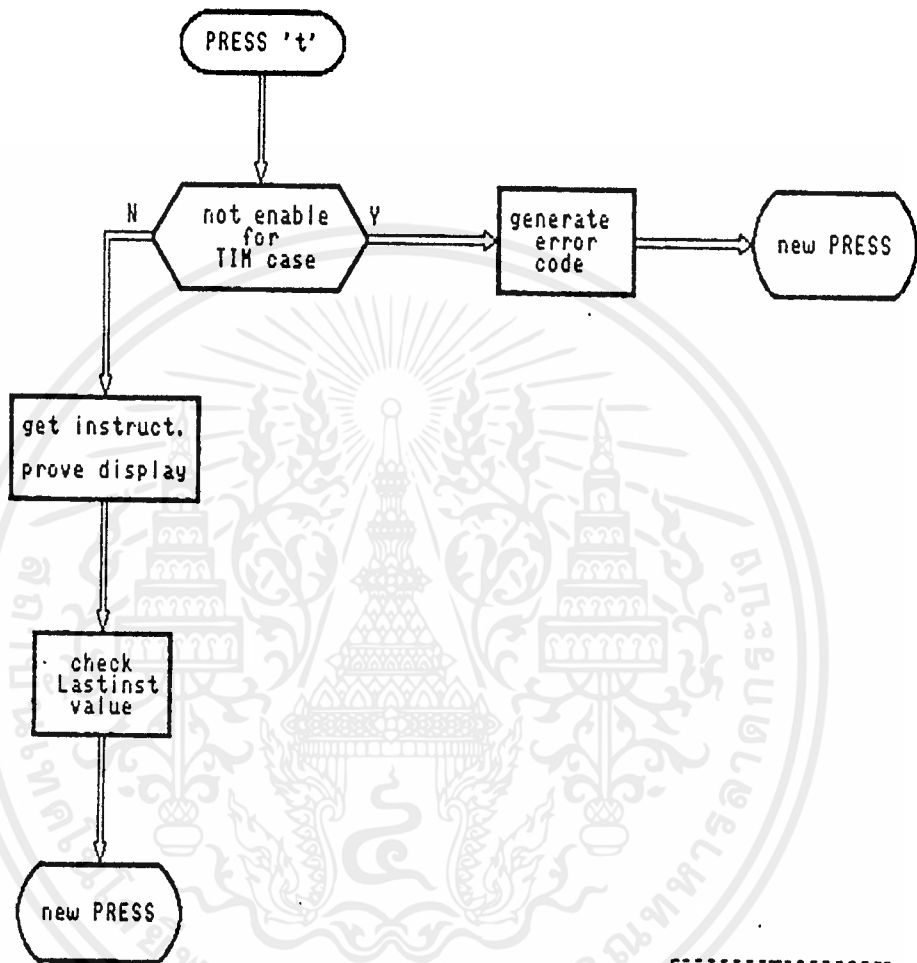
I[0,1]	ENABLE
	GET HOT
LD/AND/	ENABLE
OR	
IS/RST/	IMPOSSI-
SET/END	CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



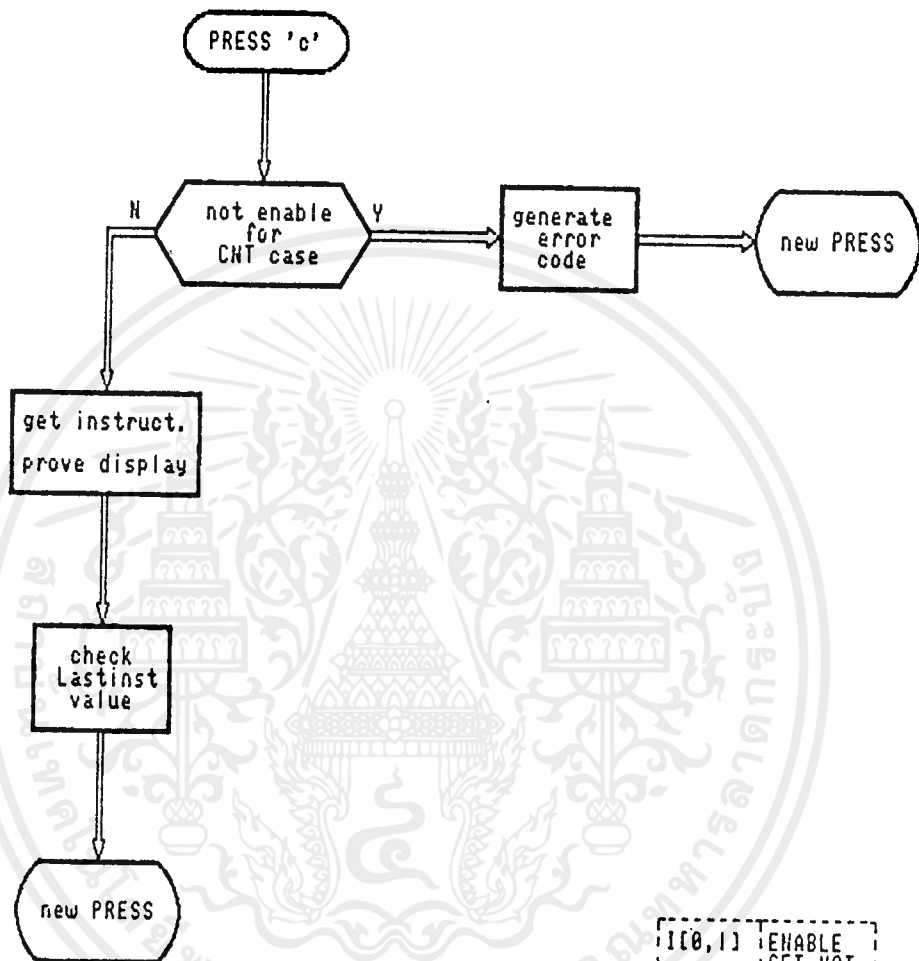
I[0,1]	ENABLE
	GET NOT
LD/AND/	ENABLE
OR/IS	
RST/SET	IMPOSSI-
END/'	CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



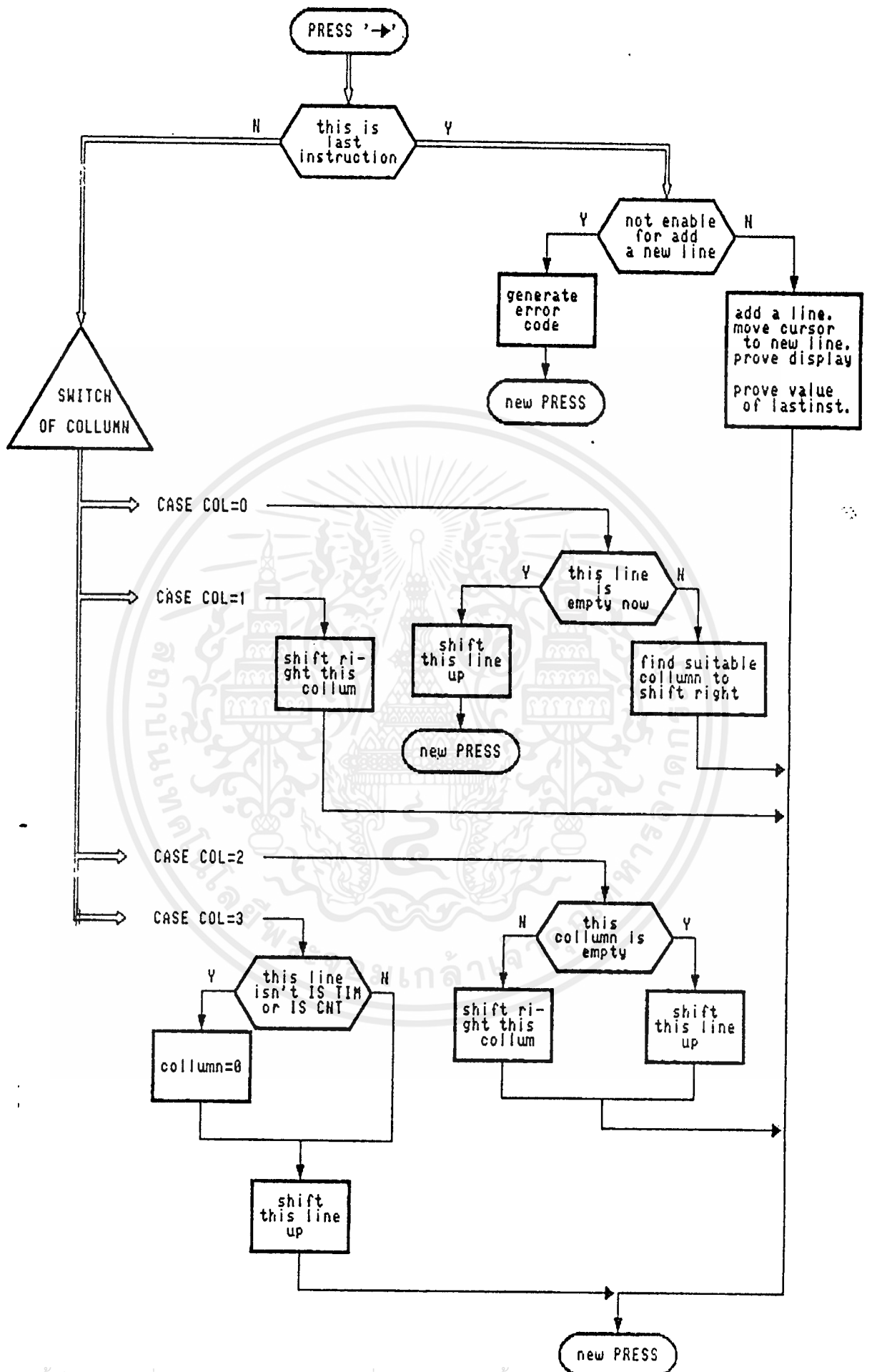
[0, 1]	ENABLE
	GET NOT
LD/AND/	ENABLE
OR/IS/	
RST	
SET/END/	IMPOSSI-
/	CASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

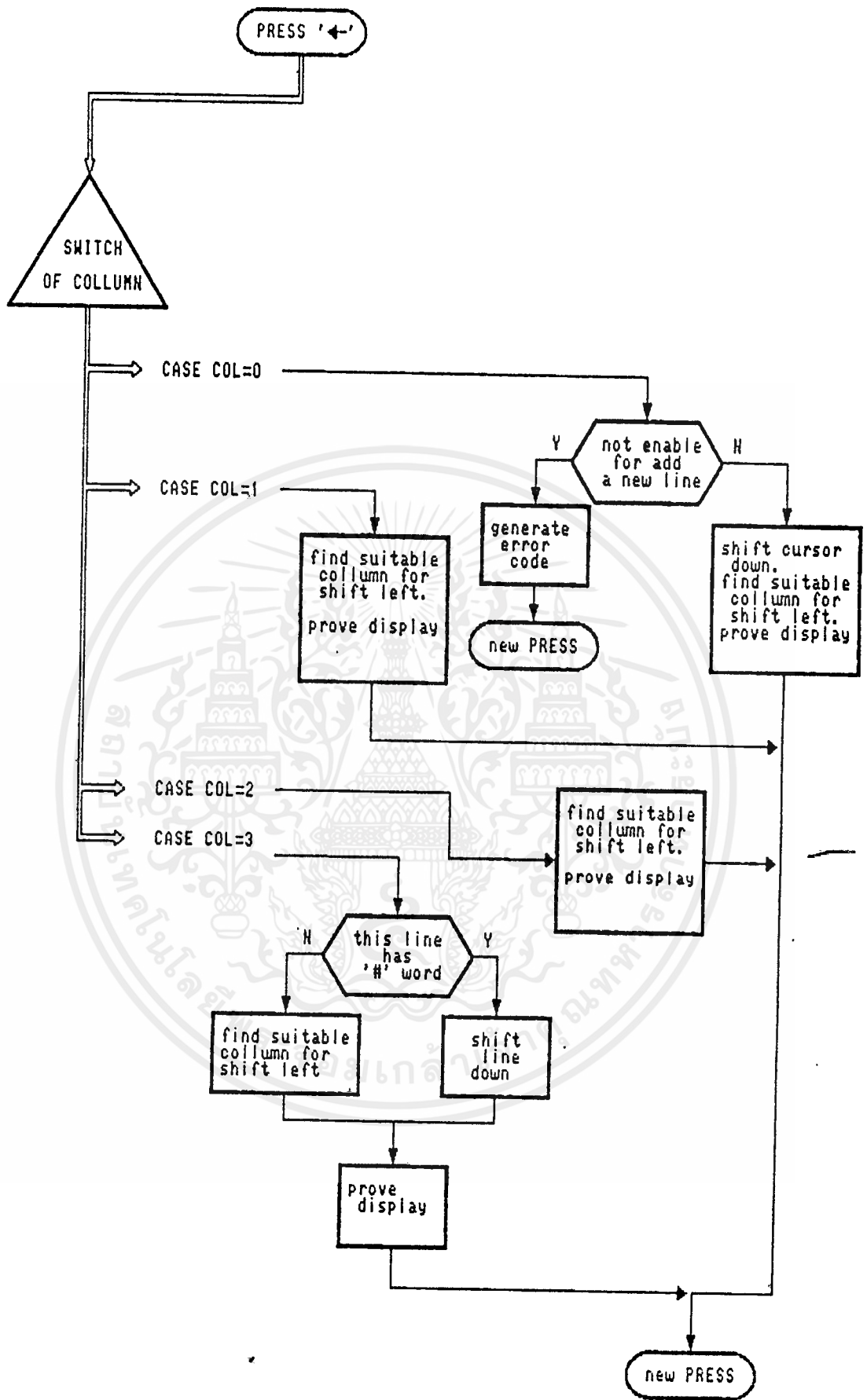


I[0,1]	ENABLE
	GET NOT
LD/AND/	ENABLE
OR/IS/	
RST	
SET/END/	IMPOSSI-
/	CASE

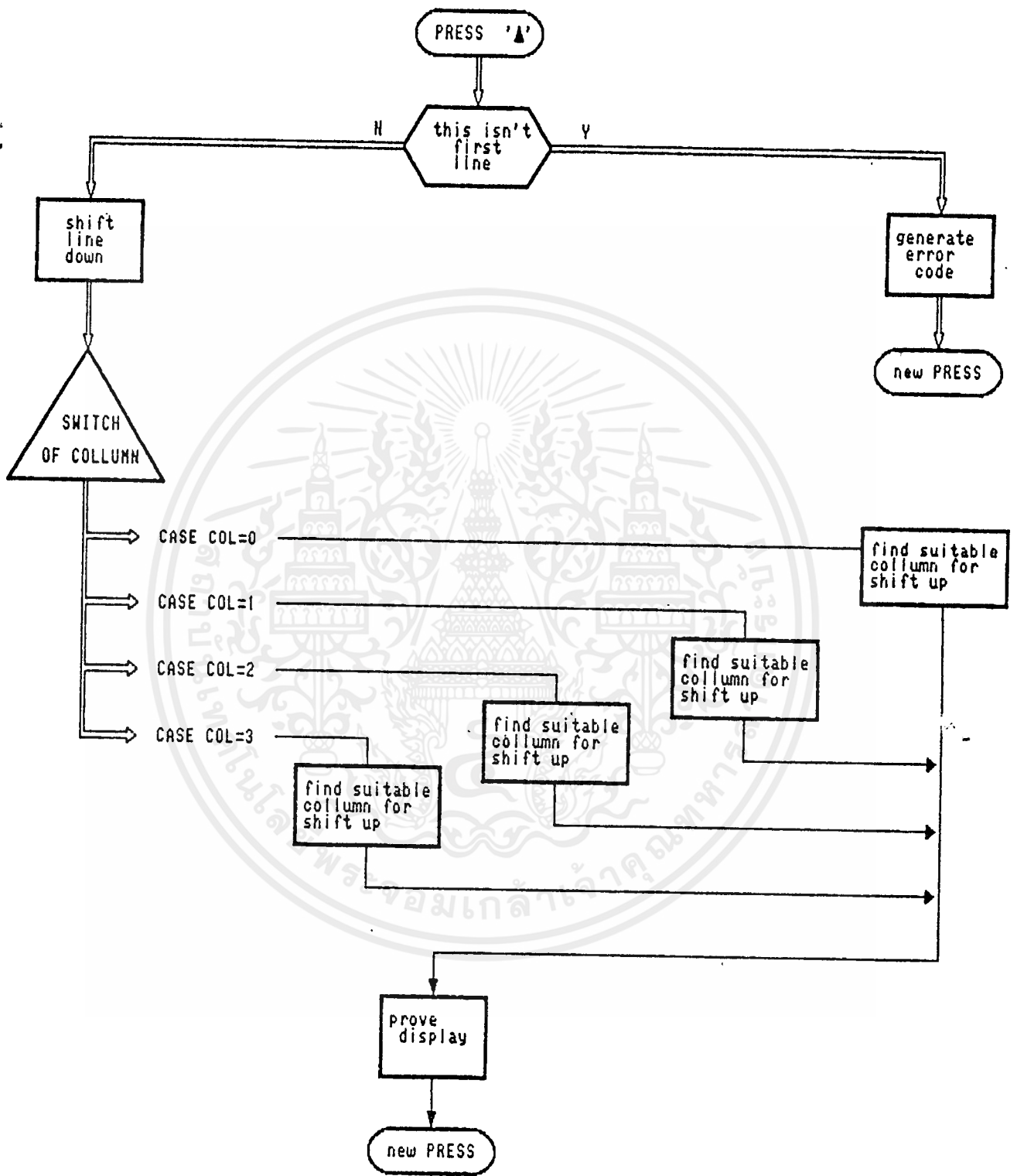
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



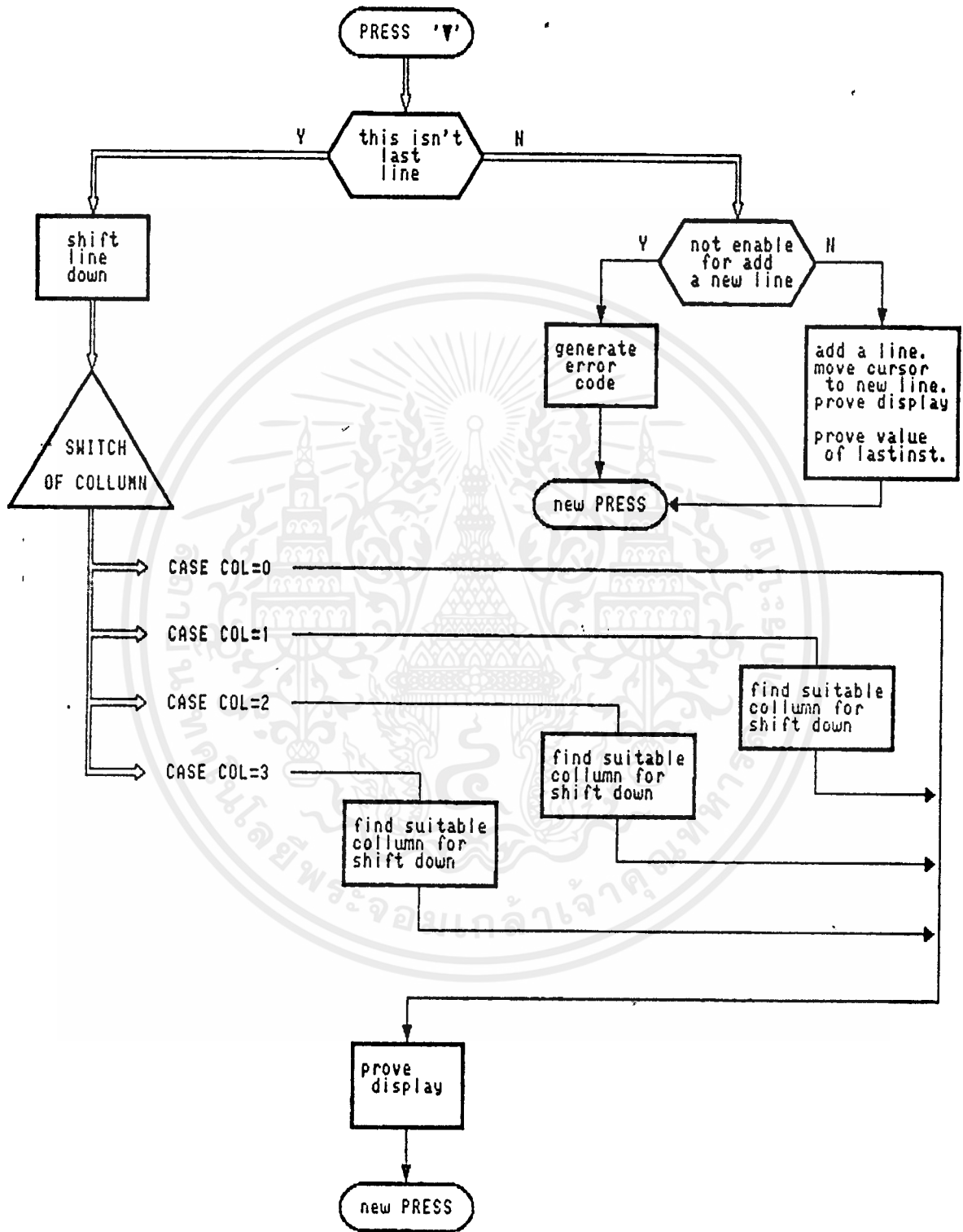
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



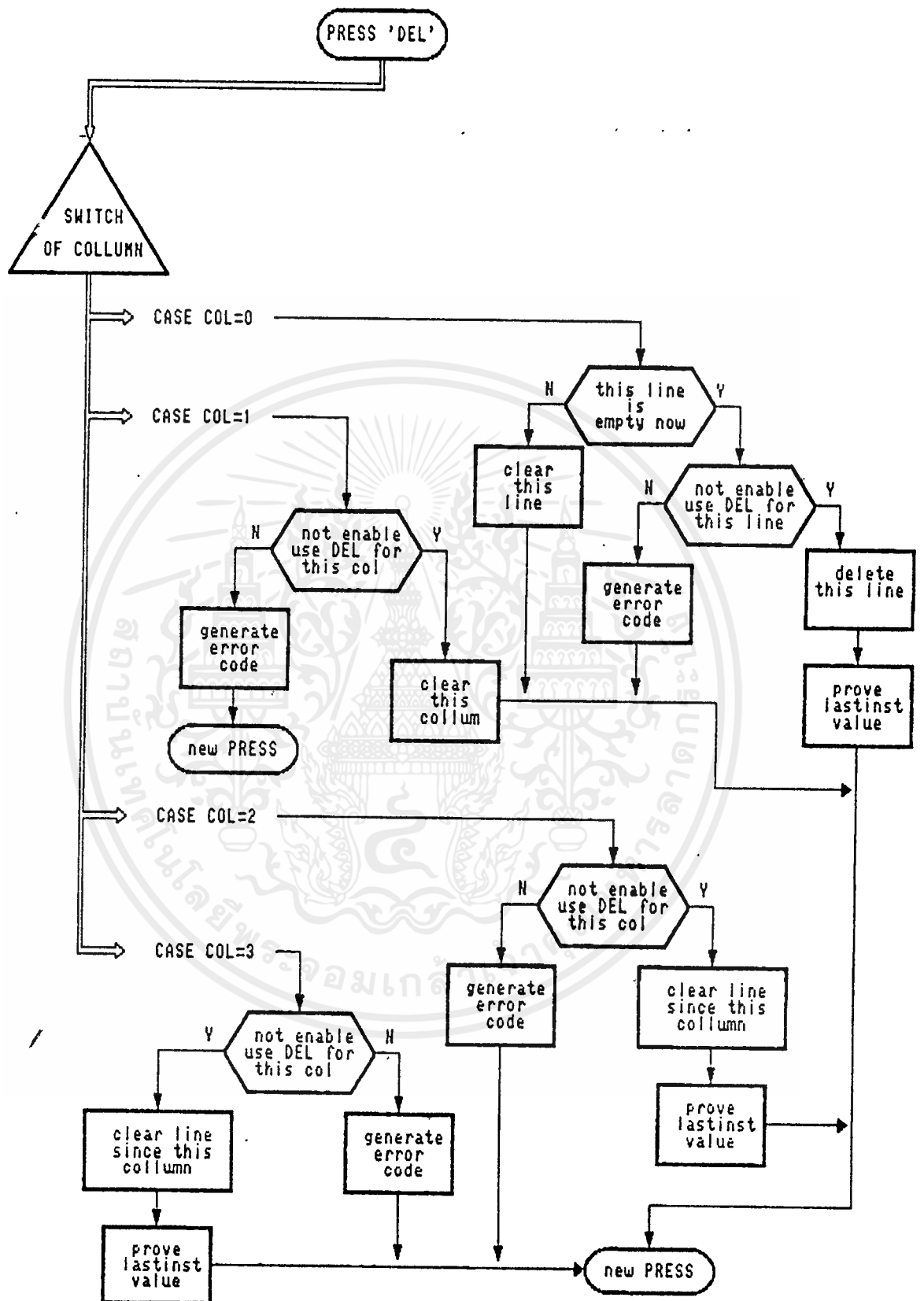
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



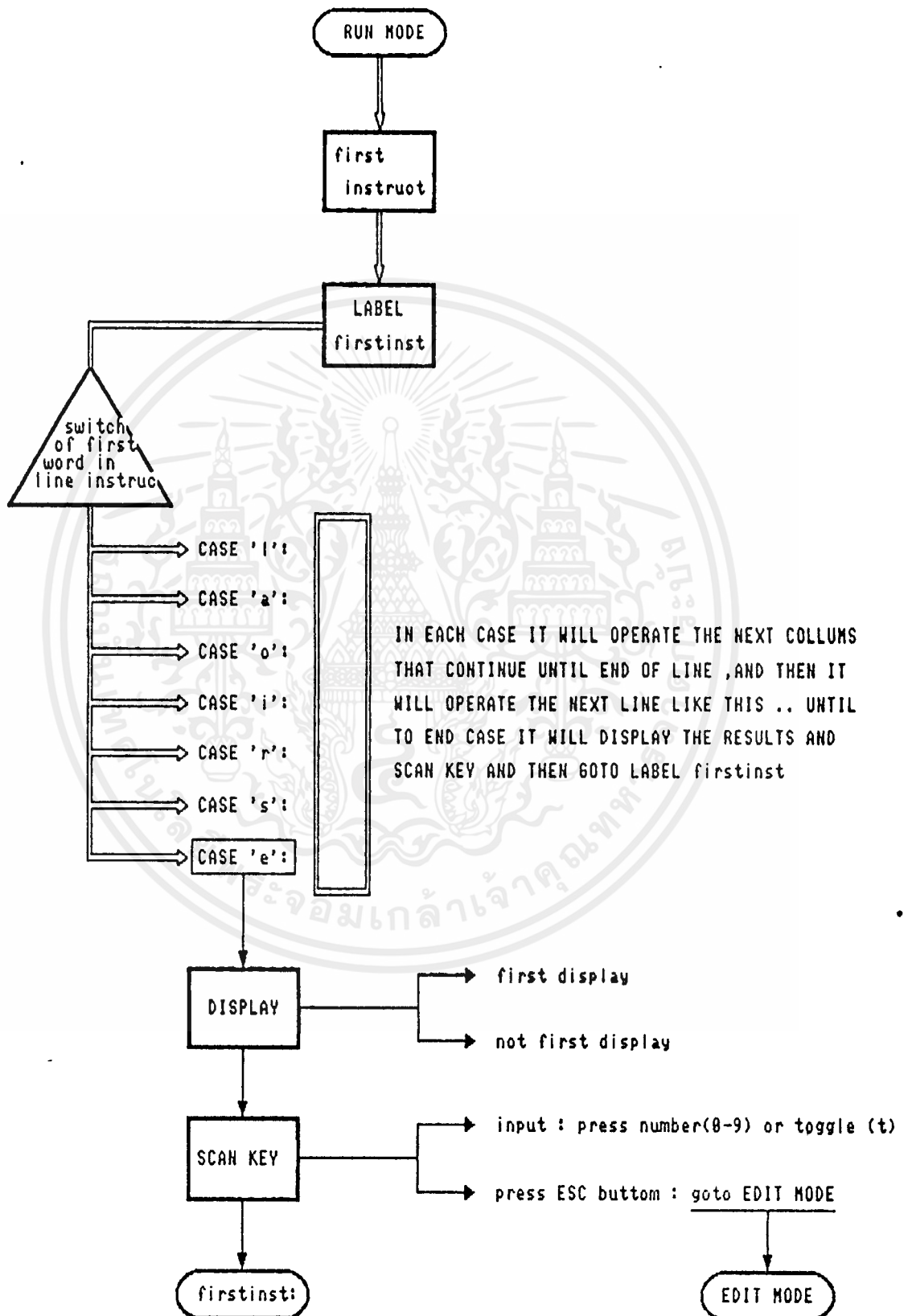
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



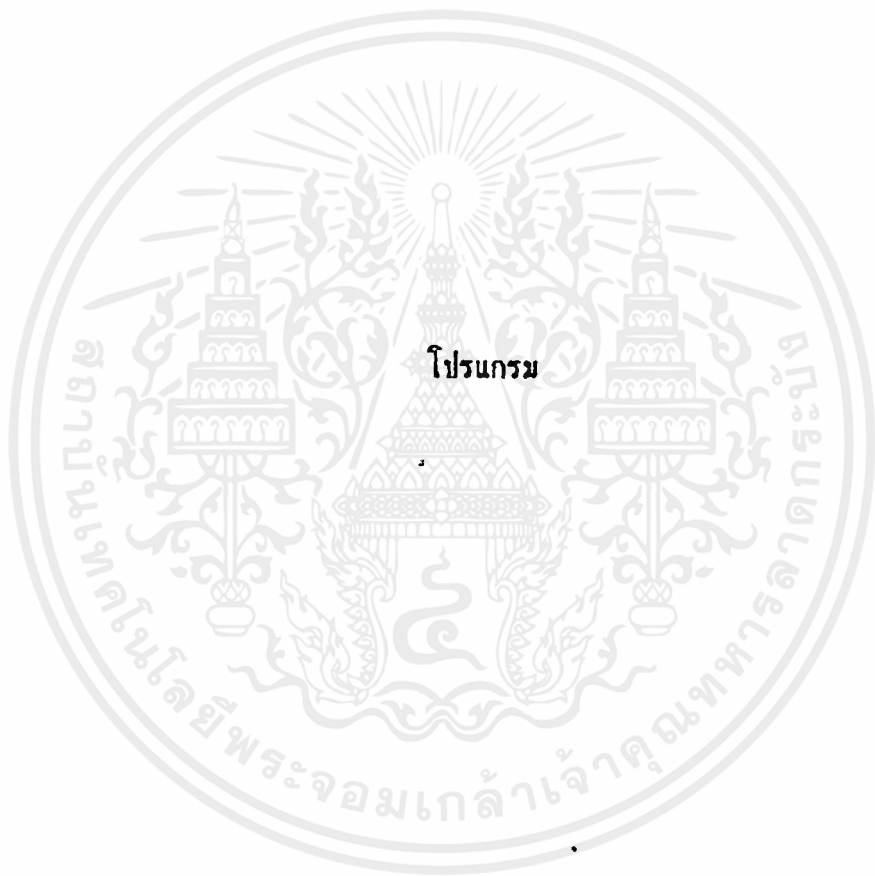
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* PROGRAM INPUT */
#include<stdio.h>
#include<graphics.h>

FILE *fp;
char ch;
char *ll;
char ptr1[931];    /****** MUST CHANGE TO BLACK *****/
char ptr2[1121];
char ptr3[931];
char ptr4[1121];
int a,b,c,l,mnl,I,old_c;
int lastinst,SumAcont,SumOcont,pos;
int MAX,num,n;
int error_code;

char ch;
char cc[3];
struct instruct{
char c0;
int c1;
char c2;
int c3;
}inst[1500];

/*****/
numinst(int,int);
xor(int,int);
ShowCol();
ShowLine(int);
ShowNum(int,int);
lg_wr(int,int);
GC(int);    /****** GENERATE ERROR CODE *****/
AddLine(int);
CMNL(int);
CID();    /****** Collage Instruction and Display *****/
CLV();    /****** Check Lastinst Vaule *****/
wr(int,int,int);
bk(int,int);
insert(int);
indent(int,int,int);
int Acont(int);
int Ocont(int);
IMNL(int,int);
DEL(int);
/*****/

```

```

numinst(int C,int L)
{
return(C+1+L*4);
}

```

```

ShowCol(
)
{
setcolor(BLACK);
outtextxy(476,450,"--");    /* Alt 219 */
setcolor(WHITE);
cc[0]=c+48;
cc[2]='\0';

```

```

outtextxy(476,450,cc);
return;
}

ShowLine(int li)
{
setcolor(BLACK);
outtextxy(576,450,"-----");
setcolor(WHITE);
if(li>999)
{ ll[0]=49; li=li-1000; }
else ll[0]=' ';
if(li>99)
{ ll[1]=li/100+48; li=li%100; }
else { if(ll[0]!=' ') ll[1]=' '; else ll[1]=48; }
if(li>9)
{ ll[2]=li/10+48; li=li%10; }
else { if(ll[1]!=' ') ll[2]=' '; else ll[2]=48; }
ll[3]=li+48; ll[4]='\0';
outtextxy(576,450,ll);
return;
}

ShowNum(int num,int MNL)
{
if(num>999)
{ ll[0]=num/1000+48; num=num%1000; }
else ll[0]=' ';
if(num>99)
{ ll[1]=num/100+48; num=num%100; }
else { if(ll[0]!=' ') ll[1]=' '; else ll[1]='0'; }
if(num>9)
{ ll[2]=num/10+48; num=num%10; }
else { if(ll[1]!=' ') ll[2]=' '; else ll[2]=48; }
ll[3]=num+48; ll[4]='\0';
outtextxy(579,70+MNL*20,ll);
return;
}

xor(C;MNL)
{
if(C==3)
putimage(411+C*50,68+MNL*20,ptr2,1);
else
putimage(411+C*50,68+MNL*20,ptr1,1);
return;
}

GC(int gc)          /***** GENERATE ERROR CODE *****/
{
error_code=gc;
printf("\7");
return;
}

AddLine(int L)
{
insert(L); lastinst+=4;
return;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CMNL(OLD_C)
{
  ShowLine(1);
  if(mnl!=18)
  { xor(OLD_C,mnl); mnl++; }
  else
  {
    for(a=18;a>-1;a--)
    {
      for(b=0;b<4;b++)
      , indent(b,l+a-18,a);
    }
  }
  return;
}

CID() /***** Collage Instruction and Display *****/
{
  switch(c)
  {
    case 0:
      inst[1].c0=ch;
      indent(c,l,mnl);
      c=1;
      switch(ch)
      {
        case 'e':
          c=0;
          break;
        case 'r':
          c=2;
          break;
        case 's':
          inst[1].c2='y'; wr(2,l,mnl); c=3;
      }
      break;
    case 1: /***** possible only 'n' case *****/
      inst[1].c1=1; indent(1,l,mnl); c=2;
      break;
    case 2: /***** MAY_BE *****/
      inst[1].c2=ch; indent(2,l,mnl); c=3;
      if(((inst[1].c2=='t')||(inst[1].c2=='c'))&&(inst[1].c0=='i'))
      {
        inst[l+1].c2='#'; wr(2,l+1,mnl+1);

        /* increase emty line before this */
      }
      break;
    case 3:
      indent(3,l,mnl);
  }
  xor(c,mnl);
  ShowCol();
  return;
}

CLV() /***** Check Lastinst Vaule *****/
{
  if(((inst[1].c2=='t')||(inst[1].c2=='c'))&&(inst[1].c0=='i'))การคำ
  เอกสารนี้เป็นเอกสารที่สว่นไว้สำหรับใช้ในงานพิเศษในเพื่อการศึกษาเท่านั้น
  ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

    { if(lastinst<numinst(c,l+1)) lastinst=numinst(c,l+1); }
    else
    { if(lastinst<numinst(c,l)) lastinst=numinst(c,l); }
    return
}

```

```

wr(int C,int L,int MNL)

```

```

{
  if((MNL<19)&&(L>-1)&&(L<1500))
  {
    switch(C)
    {
      case 0:
        switch(inst[L].c0)
        {
          case ' ':
            bk(C,MNL);
            break;
          case 'l':
            outtextxy(431,70+MNL*20,"LD");
            break;
          case 'o':
            outtextxy(431,70+MNL*20,"OR");
            break;
          case 'i':
            outtextxy(431,70+MNL*20,"IS");
            break;
          case 'a':
            outtextxy(426,70+MNL*20,"AND");
            break;
          case 'r':
            outtextxy(426,70+MNL*20,"RST");
            break;
          case 's':
            outtextxy(426,70+MNL*20,"SET");
            break;
          case 'e':
            outtextxy(426,70+MNL*20,"END");
        }
        break;
      case 1:
        if(inst[L].c1==1) outtextxy(476,70+MNL*20,"NOT");
        else bk(C,MNL);
        break;
      case 2:
        switch(inst[L].c2)
        {
          case ' ':
            bk(C,MNL);
            break;
          case 'x':
            outtextxy(531,70+MNL*20,"X");
            break;
          case 'y':
            outtextxy(531,70+MNL*20,"Y");
            break;
          case '#':
            outtextxy(531,70+MNL*20,"#");
            break;
          case ' ':
            bk(C,MNL);
            break;

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

```

        outtextxy(526,70+MNL*20,"TIM");
        break;
        case 'c':
            outtextxy(526,70+MNL*20,"CNT");
    }
    break;
    case 3:
        bk(C,MNL);
        if(inst[L].c3!=10000)
            ShowNum(inst[L].c3,MNL);
    }
}
return;
}

bk(int C,int MNL)
{
    if((MNL<19)&&(MNL>-1))
    {
        if(C==3)
            putimage(411+C*50,68+MNL*20,ptr4,0/** MAY_BE black/white **/);
        else
            putimage(411+C*50,68+MNL*20,ptr3,0/** MAY_BE **/);
    }
    return;
}

insert(L)
{
    for(a=lastinst/4;a>=L;a--)
    {
        inst[a+1].c0=inst[a].c0;
        inst[a+1].c1=inst[a].c1;
        inst[a+1].c2=inst[a].c2;
        inst[a+1].c3=inst[a].c3;
    }
    inst[L].c0=inst[L].c2=' ';
    inst[L].c1=0;
    inst[L].c3=10000;
    return;
}

indent(C,L,MNL)
{
    bk(C,MNL); wr(C,L,MNL);
    return;
}

lg_wr(int L,int MNL)
{
    for(a=0;(MNL+a<19)&&(L+a<=(lastinst/4));a++)
    {
        for(b=0;b<4;b++)
            indent(b,L+a,MNL+a);
    }
    return;
}

int Acont(int L)

```

{ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int count;
for(count=0,a=L;(a>0)&&((inst[a].c0=='a')||(inst[a].c0=='o'));a--)
{ if(inst[a].c0=='a') count++; }
return count;
}

```

```

int Ocont(int L)
{
int count;
for(count=0,a=L;(a>0)&&((inst[a].c0=='a')||(inst[a].c0=='o'));a--)
{ if(inst[a].c0=='o') count++; }
return count;
}

```

```

IMNL(int increase,int OLD_C)
{
int a,C;
if(increase) C=18;
else C=0;
if(mnl==C)
{
if(increase)
{
for(a=18;a>-1;a--)
{ for(b=0;b<4;b++) indent(b,l+a-18,a); }
}
else
{
for(a=0;a<19;a++)
{ for(b=0;b<4;b++) indent(b,l+a,a); }
}
}
else
{
xor(OLD_C,mnl);
if(increase) mnl++;
else mnl--;
}
xor(c,mnl);
return;
}

```

```

DEL(int L)
{
for(a=L;a<lastinst/4;a++)
{
inst[a].c0=inst[a+1].c0;
inst[a].c1=inst[a+1].c1;
inst[a].c2=inst[a+1].c2;
inst[a].c3=inst[a+1].c3;
}
inst[a].c0=inst[a].c2=' ';
inst[a].c1=0;
inst[a].c3=10000;
if(lastinst<=numinst(3,L))
lastinst=numinst(c,L); /** LAST LINE CASE ***/
else lastinst-=4;
for(a=0;mnl+a<19;a++)
{ for(b=0;b<4;b++) indent(b,L+a,mnl+a); }
xor(c,mnl);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return;
}

/*****/

main()
{
int i=DETECT,j;
initgraph(&i,&j,"a:\\bgi");

for(c=0;c<1500;c++)
{
inst[c].c0=inst[c].c2=' ';
inst[c].c1=0;
inst[c].c3=10000;
}

line(0,0,639,0); /* EXTERNAL BLOCK */
line(639,1,639,479);
line(638,479,0,479);
line(0,478,0,1);

line(20,60,350,60); /******/
line(20,160,350,160);
line(20,290,350,290);
outtextxy(30,75,"ENABLE KEY:");
outtextxy(30,95,"'l'<LD> 'a'<AND> 'o'<OR> 'i'<IS>");
outtextxy(30,105,"'r'<RST> 's'<SET> 'e'<END> 'n'<NOT>");
outtextxy(30,115,"'x'<X> 'y'<Y> 't'<TIM> 'c'<CNT>");
outtextxy(30,125,"num:0-9 ARROW KEY DEL KEY");
outtextxy(30,135,"F1:OLD FILE F2:SAVE F3:RUN");

outtextxy(30,175,"ABOUT THIS PROGRAM:");
outtextxy(30,195,"MAXIMUM LINE IS 1499.");
outtextxy(30,205,"WHEN TWO LINES INSTRUCTION CAN'T CLOSE UP,");
outtextxy(30,215,"THIS WILL PRODUCE SUME BUFFER LINE.");
outtextxy(30,225,"WHEN ERRORS HAVE HAPPENED,IT WILL BELL");
outtextxy(30,235,"AND NOT OPERATE THAT INSTRUCTION.");
outtextxy(30,245,"MAIN ERRORS MAY BE NOT SUITABLE LINE,");
outtextxy(30,255,"COLLUMN. ANOTHER ERRORS MAY BE CAN'T");
outtextxy(30,265,"INSERT BUFFER LINE.");

line(410,47,620,47); /* INSTRUCTION BLOCK */
line(620,48,620,467);
line(619,467,410,467);
line(410,466,410,48);

for(c=67;c<467;c+=20) /* HORIZONTAL GRID FOR INSTRUCTION */
line(411,c,619,c);
line(411,447,619,447);
line(411,448,619,448);

for(c=0;c<150;c+=50) /* VERTICAL GRID FOR INSTRUCTION */
line(460+c,48,460+c,446);
line(510,447,510,466);

floodfill(411,48,WHITE);
floodfill(461,48,WHITE);
floodfill(511,48,WHITE);
floodfill(561,48,WHITE);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

settextstyle(1,0,0);
setusercharsize(2,5,3,7);
outtextxy(426,450,"COL");
outtextxy(523,450,"LINE");

line(1,30,638,30);          /* MONITER BLOCK */
floodfill(1,1,WHITE);
setcolor(BLACK);
setusercharsize(2,3,2,7);
outtextxy(260,10,"PLC SIMULATION");
line(0,29,639,29);
line(0,28,639,28);
line(0,27,639,27);

setusercharsize(2,5,3,7);
outtextxy(428,46,"C0");
line(459,47,459,65);
line(460,47,460,65);
line(461,47,461,65);
outtextxy(478,46,"C1");
line(509,47,509,65);
line(510,47,510,65);
line(511,47,511,65);
outtextxy(528,46,"C2");
line(559,47,559,65);
line(560,47,560,65);
line(561,47,561,65);
outtextxy(580,46,"C3");
line(410,66,620,66);
line(410,65,620,65);
line(410,64,620,64);
setcolor(WHITE);

getimage(5,5,53,23,ptr1);
getimage(5,5,63,23,ptr2);
getimage(411,68,459,86,ptr3);
getimage(561,68,619,86,ptr4);

/***** text colour should be white *****/
/*****
if((fp=fopen("a:\\porn.plc","r"))!=NULL)
{
for(lastinst=0,num=fgetc(fp);isdigit(num);num=fgetc(fp))
  lastinst=num-48+10*lastinst;
for(a=0;a<lastinst/4;a++)
{
  inst[a].c0=num;
  inst[a].c1=fgetc(fp)-48;
  inst[a].c2=fgetc(fp);
  for(inst[a].c3=0,num=fgetc(fp);isdigit(num);num=fgetc(fp))
    inst[a].c3=num-48+10*inst[a].c3;
}
inst[a].c0=num;
if(numinst(1,a)<=lastinst) inst[a].c1=fgetc(fp)-48;
if(numinst(2,a)<=lastinst) inst[a].c2=fgetc(fp);
if(numinst(3,a)=lastinst)
{
for(inst[a].c3=0,num=fgetc(fp);isdigit(num);)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
inst[a].c3=num-48+10*inst[a].c3;
if(!feof(fp)) num=fgetc(fp);
else num='a';
}
}
fclose(fp);
/**
lastinst=1;
***/
lg_wr(0,0);
}
else lastinst=1;

c=1; mnl=0;
xor(c,mnl);
ShowCol();
ShowLine(1);

/*****
PRESS:
for(ch=getch();ch!=0x1b;ch=getch())
{
if(ch==0) /* case: Press Fuction key */
{
ch=getch();
switch(ch)
{
case 75: /* PRESS LEFT ARROW */
old_c=c;
switch(c)
{
case 0:
if(l==0) { GC(14); goto PRESS; }
l--;
if(inst[l].c2==' ')
{
if((inst[l].c0=='r')||(inst[l].c1!=0)) { c=2; ShowCol(); }
else { if(inst[l].c0!=' ') { c=1; ShowCol(); } }
}
else { c=3; ShowCol(); }
IMNL(0,old_c); /***** Increase MNL with careful *****/
ShowLine(1);
break;
case 1:
xor(old_c,mnl); c=0; xor(c,mnl); ShowCol();
break;
case 2:
xor(old_c,mnl);
if(inst[l].c0=='r') c=0;
else c=1;
xor(c,mnl);
ShowCol();
break;
case 3:
if(inst[l].c0=='s')
{ xor(old_c,mnl); c=0; xor(c,mnl); ShowCol(); }
else
{
if(inst[l].c2=='#') { l--; IMNL(0,old_c); ShowLine(1); }

```

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ห้ามเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        else { xor(old_c,mnl); c--; xor(c,mnl); ShowCol(); }
    }
    break;

case 72:                /* PRESS UP ARROW */
if(l==0) { GC(3); goto PRESS; }
old_c=c;
l--;
switch(c)
{
    case 0:
        if(inst[l].c2=='#') { c=3; ShowCol(); }
        break;
    case 1:
        if(inst[l].c2=='#') { c=3; ShowCol(); }
        else
        {
            if((inst[l].c0=='s')||(inst[l].c0=='r')||(inst[l].c0==' '))
            { c=0; ShowCol(); }
        }
        break;
    case 2:
        if((inst[l].c2=='#')||(inst[l].c0=='s')) { c=3; ShowCol(); }
        else
        {
            if(inst[l].c0==' ') { c=0; ShowCol(); }
            else
            {
                if((inst[l].c0!='r')&&(inst[l].c1==0)) { c=1; ShowCol(); }
            }
        }
        break;
    case 3:
        if(inst[l].c2==' ')
        {
            if(inst[l].c0==' ') c=0;
            else
            {
                if((inst[l].c0=='r')||(inst[l].c1!=0)) c=2;
                else c=1;
            }
        }
        ShowCol();
    }
}
IMNL(0,old_c);
ShowLine(l);
break;

case 77:                /* PRESS RIGHT ARROW */
old_c=c;
if(lastinst==numinst(c,l))
{
    if((l==1499)||(inst[l].c0=='e')) { GC(3); goto PRESS; }
    else
    {
        if(c==1)
        {
            xor(old_c,mnl);
            c++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        xor(c,mnl);
        ShowCol();
    }
    else
    {
        l++; c=0;
        IMNL(1,old_c);
        ShowCol();
        ShowLine(1);
    }
    CLV();          /***** Check Lastinst Vaule *****/
}
}
else
{
    switch(c)
    {
        case 0:
            if(inst[l].c0==' ')
            {
                l++; IMNL(1,old_c); ShowLine(1);
                goto PRESS;
            }
            xor(old_c,mnl);
            if(inst[l].c0=='s') c=3;
            else
            {
                if(inst[l].c0=='r') c=2;
                else c=1;
            }
            xor(c,mnl);
            ShowCol();
            break;
        case 1:
            if(inst[l].c1==0) inst[l].c1=2;
            xor(old_c,mnl);
            c=2;
            xor(c,mnl);
            ShowCol();
            break;
        case 2:
            if(inst[l].c2==' '){ c=0; l++; IMNL(1,old_c); ShowLine(1); }
            else { xor(old_c,mnl); c=3; xor(c,mnl); }
            ShowCol();
            break;
        case 3:
            if((inst[l].c0!='i')||((inst[l].c2!='t')&&(inst[l].c2!='c'))
            c=0;
            l++;
            IMNL(1,old_c);
            ShowLine(1);
        }
    }
}
break;

```

```

case 80:          /* PRESS DOWN ARROW */
    old_c=c;

```

```

    if(lastinst>numinst(3,1)) /***** THIS IS LAST LINE *****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

l++;
switch(c)
{
case 0:
  if(inst[l].c2=='#') { c=3; ShowCol(); }
  break;
case 1:
  if(inst[l].c2=='#') { c=3; ShowCol(); }
  else
  {
    I=inst[l].c0;
    if((I=='s')||(I=='r')||(I==' ')||(I=='e'))
      { c=0; ShowCol(); }
  }
  break;
case 2:
  I=inst[l].c0;
  if((inst[l].c2=='#')||(I=='s')) { c=3; ShowCol(); }
  else
  {
    if((I==' ')||(I=='e')) { c=0; ShowCol(); }
    else
    { if((I!='r')&&(inst[l].c1==0)) { c=1; ShowCol(); } }
  }
  break;
case 3:
  if(inst[l].c2==' ')
  {
    I=inst[l].c0;
    if((I==' ')||(I=='e')) c=0;
    else
    {
      if((I=='r')||(inst[l].c1!=0)) c=2;
      else c=1;
    }
  }
  ShowCol();
}
}
else /***** l = LAST LINE *****/
{
  if((l==1499)||(inst[l].c0=='e')) { GC(3); goto PRESS; }
  c=0; l++;
  ShowCol();
  ShowLine(l);
  CLV();
}
IMNL(1,old_c);
ShowLine(l);
break;

```

```

case 83: /***** PRESS DEL *****/

```

```

switch(c)
{
case 0:
  if(inst[l].c0!=' ')
  {
    inst[l].c0=inst[l].c2==' ';
    inst[l].c1=0;
  }

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

inst[l].c3=10000;
for(b=0;b<4;b++) wr(b,l,mnl);
if(l!=1499)
{
  if(inst[l+1].c2=='#')
  {
    inst[l+1].c2=' '; inst[l+1].c3=10000;
    wr(2,l+1,mnl+1); wr(3,l+1,mnl+1);
    if(lastinst<=numinst(3,l+1)) lastinst=numinst(c,l);
  }
}
xor(c,mnl);
if(lastinst<=numinst(3,l)) lastinst=numinst(c,l);
goto PRESS;
}
if(lastinst==numinst(c,l)) { GC(16); goto PRESS; }
if(l==1499) { DEL(1); goto PRESS; }
if(l==0)
{
  I=inst[l+1].c0;
  if((I=='s')||(I=='l')||(I==' ')) DEL(1);
  else GC(17);
  goto PRESS;
}
I=inst[l-1].c0;
if((I==' ')||(inst[l+1].c0==' ')) { DEL(1); goto PRESS; }
if((I=='i')||(I=='r')||(I=='s'))
{
  I=inst[l+1].c0;
  if((I!='l')&&(I!='s')&&(I!='e')) GC(18);
  else DEL(1);
  goto PRESS;
}
I=inst[l+1].c0;
if((I=='l')||(I=='s')||(I=='e')) { GC(18);goto PRESS; }
if((I=='i')||(I=='r')||(I==' ')) { DEL(1); goto PRESS; }
SumAcont=Acont(l-1)+Acont(l+1);
SumOcont=Ocont(l-1)+Ocont(l+1);
for(pos=1;((I=='a')||(I=='o'))&&(SumAcont<4)&&(SumOcont<5);)
{
  pos++;
  I=inst[l+pos].c0;
  SumAcont=Acont(l-1)+Acont(l+pos);
  SumOcont=Ocont(l-1)+Ocont(l+pos);
}
if(pos==1) { GC(18); goto PRESS; }
if((I=='i')||(I=='r')||(I==' ')) { DEL(1); goto PRESS; }
DEL(1);
AddLine(l+pos-1);
lg_wr(l+pos-1,mnl+pos-1);
break;

case 1:
  if(inst[l].c1!=1) { GC(19); goto PRESS; }
  inst[l].c1=2;
  wr(1,l,mnl);
  xcr(c,mnl);
  break;

case 2:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(inst[l].c2==' ') { GC(19); goto PRESS; }
inst[l].c2=' '; inst[l].c3=10000;
wr(2,l,mnl); wr(3,l,mnl); xor(2,mnl);
if(inst[l+1].c2=='#')
{
inst[l+1].c2=' '; inst[l+1].c3=10000;
wr(2,l+1,mnl+1); wr(3,l+1,mnl+1);
}
break;

case 3:
if(inst[l].c3==10000) { GC(19); goto PRESS; }
inst[l].c3=10000;
wr(3,l,mnl);
xor(3,mnl);
if(inst[l+1].c2=='#') { inst[l+1].c3=10000; wr(3,l+1,mnl+1); }
}
break;

case 60: /* press F2 */
for(a=0;a<lastinst/4;a++)
if(inst[a].c3==10000) { GC(20); goto PRESS; }
/* translate data to file */
/* IN F1 MENU (SAVE) */
if((fp=fopen("a:\\porn.plc","w"))==NULL)
printf("Can't open a this file\n");
num=lastinst;
if(num/1000)
{
fputc(num/1000+48,fp);
num=num%1000;
fputc(num/100+48,fp);
num=num%100;
fputc(num/10+48,fp);
fputc(num%10+48,fp);
}
else
{
if(num/100)
{
fputc(num/100+48,fp);
num=num%100;
fputc(num/10+48,fp);
fputc(num%10+48,fp);
}
else
{
if(num/10)
{
fputc(num/10+48,fp);
fputc(num%10+48,fp);
}
else
fputc(num+48,fp);
}
}
for(a=0;a<=lastinst/4-1;a++)
{
fputc(inst[a].c0,fp);
fputc(inst[a].c1+48,fp);

```

```

fputc(inst[a].c2,fp);
num=inst[a].c3;
if(num==10000)
{
fputc('1',fp); fputc('0',fp); fputc('0',fp);
fputc('0',fp); fputc('0',fp);
}
else
{
if(num/1000)
{
fputc(num/1000+48,fp);
num=num%1000;
fputc(num/100+48,fp);
num=num%100;
fputc(num/10+48,fp);
fputc(num%10+48,fp);
}
else
{
if(num/100)
{
fputc(num/100+48,fp);
num=num%100;
fputc(num/10+48,fp);
fputc(num%10+48,fp);
}
else
{
if(num/10)
{
fputc(num/10+48,fp);
fputc(num%10+48,fp);
}
else
fputc(num+48,fp);
}
}
}
}
fputc(inst[a].c0,fp);
fclose(fp);
break;

```

```

default:
GC(21);
}
}

```

```

else /* case: Press Charecter key */
{

```

```

switch(ch)
{

```

```

case '1':

```

```

if(c!=0) { GC(1); goto PRESS; }

```

```

if(inst[l].c0!=' ') { GC(2); goto PRESS; }

```

```

if(l==1499) { GC(3); goto PRESS; }

```

```

old_c=c;

```

```

if((l!=0)&&((inst[l-1].c0=='1')||((inst[l-1].c0=='a')

```

```

||(inst[l+1].c0=='o'))
{
  if((lastinst>5992);!(inst[1498].c0!='e')
  &&(inst[1498].c0!=' '))
  { GC(4); goto PRESS; }
  I=inst[l+1].c0;
  if((I!='i')&&(I!='r')&&(I!=' '))
  {
    pos=1; I=inst[l+pos].c0;
    for(;;((I=='a')||(I=='o'))&&(Acont(l+pos)<4)
    &&(Ocont(l+pos)<5));
    { pos++; I=inst[l+pos].c0; }
    if((I!='i')&&(I!='r')&&(I!=' '))
    {
      if((lastinst>5988);!(inst[1497].c0!='e')
      &&(inst[1497].c0!=' '))
      { GC(4); goto PRESS; }
      AddLine(l+pos);
    }
  }
  AddLine(l);
  l++;
  CMNL(old_c);
  /***** Change MNL with care full about monitor display *****/
  CID();
  /***** Collage Instruction and Display *****/
  lg_wr(l+1,mnl+1);
  /***** D(1) *****/
}
else
{
  I=inst[l+1].c0;
  if((I!='i')&&(I!='r')&&(I!=' '))
  {
    pos=1; I=inst[l+pos].c0;
    for(;;((I=='a')||(I=='o'))&&(Acont(l+pos)<4)
    &&(Ocont(l+pos)<5));
    { pos++; I=inst[l+pos].c0; }
    if((I!='i')&&(I!='r')&&(I!=' '))
    {
      if((lastinst>5992);!(inst[1498].c0!='e')
      &&(inst[1498].c0!=' '))
      { GC(4); goto PRESS; }
      AddLine(l+pos);
      xor(c,mnl);
      CID();
      lg_wr(l+pos,mnl+pos);
    }
  }
  xor(c,mnl);
  CID();
}
else
{
  xor(c,mnl);
  CID();
}
/***** D(0) *****/

```

```

}
CLV();
/***** Check Lastinst If line increase *****/
break;

case 'a':
if(c!=0) { GC(1); goto PRESS; }
if(inst[l].c0!=' ') { GC(2); goto PRESS; }
if(l==1499) { GC(3); goto PRESS; }
if(l==0) { GC(3); goto PRESS; }
old_c=c;
I=inst[l-1].c0;
if((I=='i')||(I=='r')||(I=='s')||((I==' ')
&&(inst[l-1].c2=='#')))
{
if((lastinst>5992)||((inst[1498].c0!='e')
&&(inst[1498].c0!=' ')))
{ GC(4); goto PRESS; }
I=inst[l+1].c0;
if((I!='i')&&(I!='r')&&(I!=' '))
{
for(pos=1;((I=='a')||(I=='o'))&&(Acont(l+pos)<3);)
{ pos++; I=inst[l+pos].c0; }
if((I!='i')&&(I!='r')&&(I!=' '))
{
if((lastinst>5988)||((inst[1497].c0!='e')
&&(inst[1497].c0!=' ')))
{ GC(4); goto PRESS; }
AddLine(l+pos);
}
}
AddLine(l);
l++;
CMNL(old_c);
CID();
lg_wr(l+1,mnl+1);
/***** D(1)*****/
}
else
{
if(Acont(l-1)==3)
{
if((lastinst>5992)||((inst[1498].c0!='e')
&&(inst[1498].c0!=' ')))
{ GC(4); goto PRESS; }
I=inst[l+1].c0;
if((I!='i')&&(I!='r')&&(I!=' '))
{
for(pos=1;((I=='a')||(I=='o'))&&(Acont(l+pos)<3);)
{ pos++; I=inst[l+pos].c0; }
if((I!='i')&&(I!='r')&&(I!=' '))
{
if((lastinst>5988)||((inst[1497].c0!='e')
&&(inst[1497].c0!=' ')))
{ GC(4); goto PRESS; }
AddLine(l+pos);
}
}
}
}
AddLine(l);
l++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CMNL(old_c);
CID();
lg_wr(l+1,mnl+1);
/***** D(1) *****/
}
else
{
I=inst[l+1].c0;
if((I!='i')&&(I!='r')&&(I!=' '))
{
I=inst[l+1].c0;
SumAcont=Acont(l-1)+Acont(l+1);
SumOcont=Ocont(l-1)+Ocont(l+1);
for(pos=1;((I=='a')||(I=='o'))&&(SumAcont<3)&&(SumOcont<5);)
{
pos++; I=inst[l+pos].c0;
SumAcont=Acont(l-1)+Acont(l+pos);
SumOcont=Ocont(l-1)+Ocont(l+pos);
}
if((I!='i')&&(I!='r')&&(I!=' '))
{
if((lastinst>5992)!!((inst[1498].c0!='e')
&&(inst[1498].c0!=' ')))
{ GC(4); goto PRESS; }
AddLine(l+pos);
xor(c,mnl);
CID();
lg_wr(l+pos,mnl+pos);
}
else
{
xor(c,mnl);
CID();
}
}
else
{
xor(c,mnl);
CID();
}
/***** D(0) *****/
}
}
CLV();
break;

```

```

case 'o':
if(c!=0) { GC(1); goto PRESS; }
if(inst[l].c0!=' ') { GC(2); goto PRESS; }
if(l==1499) { GC(3); goto PRESS; }
if(l==0) { GC(3); goto PRESS; }
old_c=c;
I=inst[l-1].c0;
if((I=='i')||(I=='r')||(I=='s')||(I==' '))
&&(inst[l-1].c2=='#'))
{
if((lastinst>5992)!!((inst[1498].c0!='e')
&&(inst[1498].c0!=' ')))
{ GC(4); goto PRESS; }

```

```

I=inst[l+1].c0;
if((I!='i')&&(I!='r')&&(I!=' '))
{
for(pos=1;((I=='a')||(I=='o'))&&(Ocont(l+pos)<4);)
{ pos++; I=inst[l+pos].c0; }
if((I!='i')&&(I!='r')&&(I!=' '))
{
if((lastinst>5988)||((inst[1497].c0!='e')
&&(inst[1497].c0!=' ')))
{ GC(4); goto PRESS; }
AddLine(l+pos);
}
}
AddLine(l);
l++;
CMNL(old_c);
CID();
lg_wr(l+1,mnl+1);
/***** D(1) *****/
}
else
{
if(Ocont(l-1)==4)
{
if((lastinst>5992)||((inst[1498].c0!='e')
&&(inst[1498].c0!=' ')))
{ GC(4); goto PRESS; }
I=inst[l+1].c0;
if((I!='i')&&(I!='r')&&(I!=' '))
{
for(pos=1;((I=='a')||(I=='o'))&&(Ocont(l+pos)<4);)
{ pos++; I=inst[l+pos].c0;}
if((I!='i')&&(I!='r')&&(I!=' '))
{
if((lastinst>5988)||((inst[1497].c0!='e')
&&(inst[1497].c0!=' ')))
{ GC(4); goto PRESS; }
AddLine(l+pos);
}
}
}
AddLine(l);
l++;
CMNL(old_c);
CID();
lg_wr(l+1,mnl+1);
/***** D(1) *****/
}
else
{
I=inst[l+1].c0;
if((I!='i')&&(I!='r')&&(I!=' '))
{
I=inst[l+1].c0;
SumAcont=Acont(l-1)+Acont(l+1);
SumOcont=Ocont(l-1)+Ocont(l+1);
for(pos=1;((I=='a')||(I=='o'))&&(SumAcont<4)&&(SumOcont<4);)
{
pos++; I=inst[l+pos].c0;
SumAcont=Acont(l-1)+Acont(l+pos);
SumOcont=Ocont(l-1)+Ocont(l+pos);
}
}
}
}
}

```

```

}
if((I!='i')&&(I!='r')&&(I!=' '))
{
  if((lastinst>5992)||((inst[1498].c0!='e')
  &&(inst[1498].c0!=' ')))
  { GC(4); goto PRESS; }
  AddLine(l+pos);
  xor(c,mnl);
  CID();
  lg_wr(l+pos,mnl+pos);
}
else
{
  xor(c,mnl);
  CID();
}
}
else
{
  xor(c,mnl);
  CID();
}
}
/***** D(0) *****/
}
}
CLV();
break;

case 'i':
if(c!=0) { GC(1); goto PRESS; }
if(inst[l].c0!=' ') { GC(2); goto PRESS; }
if(l==1499) { GC(3); goto PRESS; }
if(l==0) { GC(3); goto PRESS; }
old_c=c;
I=inst[l-1].c0;
if((I=='i')||(I=='r')||(I=='s')||(inst[l-1].c2=='#'))
{
  if((lastinst>5992)||((inst[1498].c0!='e')
  &&(inst[1498].c0!=' ')))
  { GC(4); goto PRESS; }
  I=inst[l+1].c0;
  if((I=='a')||(I=='o')||(I=='i')||(I=='r'))
  {
    if((lastinst>5988)||((inst[1497].c0!='e')
    &&(inst[1497].c0!=' ')))
    { GC(4); goto PRESS; }
    AddLine(l+1);
  }
  AddLine(l);
  l++;
  CMNL(old_c);
  CID();
  lg_wr(l+1,mnl+1);
  /***** D(1) *****/
}
else
{
  I=inst[l+1].c0;
  if((I=='a')||(I=='o')||(I=='i')||(I=='r'))

```

```

{
  if((lastinst>5992)!!((inst[1498].c0!='e')
  &&(inst[1498].c0!=' ')))
  { GC(4); goto PRESS; }
  AddLine(l+1);
  xor(c,mnl);
  CID();
  lg_wr(l+1,mnl+1);
}
else
{
  xor(c,mnl);
  CID();
}
}
/***** D(0) *****/
}
CLV();
break;

```

```

case 'r':
if(c!=0) { GC(1); goto PRESS; }
if(inst[l].c0!=' ') { GC(2); goto PRESS; }
if(l==1499) { GC(3); goto PRESS; }
if(l==0) { GC(3); goto PRESS; }
old_c=c;
I=inst[l-1].c0;
if((I=='i')!!(I=='r')!!(I=='s'))
{
  if((lastinst>5992)!!((inst[1498].c0!='e')
  &&(inst[1498].c0!=' ')))
  { GC(4); goto PRESS; }
  I=inst[l+1].c0;
  if((I=='a')!!(I=='o')!!(I=='i')!!(I=='r'))
  {
    if((lastinst>5988)!!((inst[1497].c0!='e')
    &&(inst[1497].c0!=' ')))
    { GC(4); goto PRESS; }
    AddLine(l+1);
  }
  AddLine(l);
  l++;
  CMNL(old_c);
  CID();
  lg_wr(l+1,mnl+1);
/***** D(1) *****/
}
else
{
  I=inst[l+1].c0;
  if((I=='a')!!(I=='o')!!(I=='i')!!(I=='r'))
  {
    if((lastinst>5992)!!((inst[1498].c0!='e')
    &&(inst[1498].c0!=' ')))
    { GC(4); goto PRESS; }
    AddLine(l+1);
    xor(c,mnl);
    CID();
    lg_wr(l+1,mnl+1);
  }
}
}

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        xor(c,mnl);
        CID();
    }
    /***** D(0) *****/
}
CLV();
break;

case 's':
if(c!=0) { GC(1); goto PRESS; }
if(inst[l].c0!=' ') { GC(2); goto PRESS; }
if(l==1499) { GC(3); goto PRESS; }
old_c=c;
if((l!=0)&&((inst[l-1].c0=='l')||(inst[l-1].c0=='a')
||(inst[l-1].c0=='o'))
{
    if((lastinst>5992)||((inst[1498].c0!='e')
&&(inst[1498].c0!=' ')))
    { GC(4); goto PRESS; }
    I=inst[l+1].c0;
    if((I=='a')||(I=='o')||(I=='i ')||(I=='r'))
    {
        if((lastinst>5988)||((inst[1497].c0!='e')
&&(inst[1497].c0!=' ')))
        { GC(4); goto PRESS; }
        AddLine(l+1);
    }
    AddLine(l);
    l++;
    CMNL(old_c);
    CID();
    lg_wr(l+1,mnl+1);
    /***** D(1) *****/
}
else
{
    I=inst[l+1].c0;
    if((I=='a')||(I=='o')||(I=='i ')||(I=='r'))
    {
        if((lastinst>5992)||((inst[1498].c0!='e')
&&(inst[1498].c0!=' ')))
        { GC(4); goto PRESS; }
        AddLine(l+1);
        xor(c,mnl);
        CID();
        lg_wr(l+1,mnl+1);
    }
    else
    {
        xor(c,mnl);
        CID();
    }
    /***** D(0) *****/
}
CLV();
break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(c!=0) { GC(1); goto PRESS; }
if(inst[l].c0!=' ') { GC(2); goto PRESS; }
if(l==0) { GC(3); goto PRESS; }
old_c=c;
if(lastinst!=numinst(0,l))
{
  if(l==1499) { GC(4); goto PRESS; }
  I=inst[l+1].c0;
  for(a=1499;a>l;a--)
  { if(inst[a].c0!=' ') { GC(5); goto PRESS; } }
  I=inst[l-1].c0;
  if((I=='l')||(I=='a')||(I=='o'))
  {
    AddLine(l);
    l++;
    CMNL(old_c);
    CID();
    lg_wr(l+1,mnl+1);
    /***** D(1) *****/
  }
  else
  {
    xor(c,mnl);
    CID();
    /***** D(0) *****/
  }
  lastinst=numinst(c,l);
}
else
{
  I=inst[l-1].c0;
  if((I=='l')||(I=='a')||(I=='o'))
  {
    if(l==1499) { GC(4); goto PRESS; }
    AddLine(l);
    l++;
    CMNL(old_c);
    CID();
    lg_wr(l+1,mnl+1);
    /***** D(1) *****/
  }
  else
  {
    xor(c,mnl);
    CID();
    /***** D(0) *****/
  }
}
break;

```

```

cas  n':
if(c!=1) { GC(1); goto PRESS; }
if(inst[l].c1==1) { GC(6); goto PRESS; }
xor(c,mnl);
CID();
/***** D(0) *****/
CLV();
break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 'x':
  if(c!=2) { GC(1); goto PRESS; }
  if(inst[1].c2!=' ') { GC(2); goto PRESS; }
  I=inst[1].c0;
  if((I!='l')&&(I!='a')&&(I!='o')) { GC(7); goto PRESS; }
  xor(c,mnl);
  CID();
  /**** D(0) ****/
  CLV();
  break;

```

```

case 'y':
  if(c!=2) { GC(1); goto PRESS; }
  if(inst[1].c2!=' ') { GC(2); goto PRESS; }
  if(inst[1].c0==' ') { GC(2); goto PRESS; }
  if(inst[1].c0=='r') { GC(8); goto PRESS; }
  xor(c,mnl);
  CID();
  /**** D(0) ****/
  CLV();
  break;

```

```

case 't':
  if(c!=2) { GC(1); goto PRESS; }
  if(inst[1].c2!=' ') { GC(2); goto PRESS; }
  if(inst[1].c0=='i')
  {
    if(inst[l+1].c0!=' ')
    {
      if((lastinst>5992)||((inst[1498].c0!='e')
      &&(inst[1498].c0!=' ')))
      { GC(4); goto PRESS; }
      I=inst[l+1].c0;
      if((I=='a')||(I=='o')||(I=='i')||(I=='r'))
      {
        if((lastinst>5988)||((inst[1497].c0!='e')
        &&(inst[1497].c0!=' ')))
        { GC(4); goto PRESS; }
        AddLine(l+1);
      }
      AddLine(l+1);
      lg_wr(l+1,mnl+1);
    }
  }
  else
  {
    if((I=='a')||(I=='o')||(I=='i')||(I=='r'))
    {
      if((lastinst>5992)||((inst[1498].c0!='e')
      &&(inst[1498].c0!=' ')))
      { GC(4); goto PRESS; }
      AddLine(l+2);
      lg_wr(l+2,mnl+2);
    }
  }
}
xor(c,mnl);
CID();

```

```
/****** D(0) *****/
```

```
CLV();
```

```
break;
```

```
case 'c':
```

```
if(c!=2) { GC(1); goto PRESS; }
```

```
if(inst[1].c2!=' ') { GC(2); goto PRESS; }
```

```
if(inst[1].c0=='i')
```

```
{
```

```
if(inst[l+1].c0!=' ')
```

```
{
```

```
if((lastinst>5992)||((inst[1498].c0!='e')
```

```
&&(inst[1498].c0!=' ')))
```

```
{ GC(4); goto PRESS; }
```

```
I=inst[l+1].c0;
```

```
if((I=='a')||(I=='o')||(I=='i')||(I=='r'))
```

```
{
```

```
if((lastinst>5988)||((inst[1497].c0!='e')
```

```
&&(inst[1497].c0!=' ')))
```

```
{ GC(4); goto PRESS; }
```

```
AddLine(l+1);
```

```
}
```

```
AddLine(l+1);
```

```
lg_wr(l+1,mnl+1);
```

```
}
```

```
else
```

```
{
```

```
if((I=='a')||(I=='o')||(I=='i')||(I=='r'))
```

```
{
```

```
if((lastinst>5992)||((inst[1498].c0!='e')
```

```
&&(inst[1498].c0!=' ')))
```

```
{ GC(4); goto PRESS; }
```

```
AddLine(l+2);
```

```
lg_wr(l+2,mnl+2);
```

```
}
```

```
}
```

```
xor(c,mnl);
```

```
CID();
```

```
/****** D(0) *****/
```

```
CLV();
```

```
break;
```

```
default: /****** PRESS NUM *****/
```

```
if(!(isdigit(ch))) { GC(10); goto PRESS; }
```

```
if(c!=3) { GC(1); goto PRESS; }
```

```
switch(inst[1].c2)
```

```
{
```

```
case 'x':
```

```
MAX=31;
```

```
break;
```

```
case 'y':
```

```
MAX=31;
```

```
break;
```

```
case 't':
```

```
MAX=13;
```

```
break;
```

```
ca 'c':
```

```
MAX=9;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
/* PROGRAM OUTPUT */
```

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#include<sys\timeb.h>
#include<dos.h>

struct timeb buf;

FILE *fp;
int a,b,c,num,lastinst;
char *ll;
char ch;
int Result,And,Or,NOT;
int n,l,x[32],y[32],old;

struct instruction{
char c0;
int c1;
char c2;
int c3;
}inst[1500];

struct tm{
int t[14]; /* t : timer contact */
int num[14]; /* num : contain #num */
unsigned long tim[14]; /* tim : accumulate counting time */
int state[14]; /* state : timer run or rest */
unsigned long start[14]; /* start : update start time for time
counting */
int not[14]; /* not : timer has NOT or hasn't */
int rst[14]; /* rst : reset timer or not */
}t;

struct counter{
int c[10]; /* c : counter contact */
int num[10]; /* num: contain #num */
int count[10]; /* count : amount of pluse */
int oldResult[10]; /* oldResult : for checking upper triggering */
int not[10]; /* not : counter has NOT or hasn't */
int rst[10]; /* rst : reset counter or not */
}ctr;

/* variable for receive input */
int kb,nn,xnum;
char scannum[4];

/* correcting changed output for display */
char cha[200]; /* this array contain changed contact or
output */
int nch; /* this nch point element of cha array */

/* display variable */
int nfd; /* Not First Display */
int d; /* distance for assign line position display */
```

```
main()
```

```
{ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
/***** DECLARATION PATH *****/
```

```

    _AX = 0x03;
    geninterrupt(0x10);
    for(n=0;n<=31;n++)
        x[n]=y[n]=0;

    for(n=0;n<=13;n++) /*** TIMER VARIABLE DECLARATION ***/
        t.t[n]=t.num[n]=t.tim[n]=t.state[n]=t.start[n]=t.not[n]=t.rst[n]=0;

    for(n=0;n<=9;n++)
    {
        ctr.c[n]=ctr.num[n]=ctr.count[n]=0;
        ctr.oldResult[n]=ctr.not[n]=ctr.rst[n]=0;
    }

    for(n=0;n<1500;n++)
    { inst[n].c0=inst[n].c2=' '; inst[n].c1=0; inst[n].c3=10000; }

    /***** PROGRAM *****/
    if((fp=fopen("a:\\porn.plc","r"))==NULL)
        printf("Can't open a this file\n");
    for(lastinst=0,num=fgetc(fp);isdigit(num);num=fgetc(fp))
        lastinst=num-48+10*lastinst;
    for(a=0;a<lastinst/4;a++)
    {
        inst[a].c0=num;
        inst[a].c1=fgetc(fp)-48;
        if(inst[a].c1==2) inst[a].c1=0;
        inst[a].c2=fgetc(fp);
        for(inst[a].c3=0,num=fgetc(fp);isdigit(num);num=fgetc(fp))
            inst[a].c3=num-48+10*inst[a].c3;
    }
    inst[a].c0=num;
    if(numinst(1,a)<=lastinst)
    {
        inst[a].c1=fgetc(fp)-48;
        if(inst[a].c1==2) inst[a].c1=0;
    }
    if(numinst(2,a)<=lastinst) inst[a].c2=fgetc(fp);
    if(numinst(3,a)==lastinst)
    {
        for(inst[a].c3=0,num=fgetc(fp);isdigit(num);)
        {
            inst[a].c3=num-48+10*inst[a].c3;
            if(!feof(fp)) num=fgetc(fp);
            else num='a';
        }
    }
    for(a=lastinst/4+1;a<1500;a++)
    {
        inst[a].c0=' ';
        inst[a].c1=0;
        inst[a].c2=' ';
        inst[a].c3=10000;
    }
    fclose(fp);
    nfd=0; /* display contact or output isn't first time */
    nn=0; /* scan input display variable */
    xnum=32; /* scan input display variable */
    nch=d=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                /***** END DECLARATION PATH *****/
l=-1;
firstinst:      /*** MAIN LABLE ***/
  l++;
  n=inst[l].c3;
  switch(inst[l].c0){ /* now *fp points at next of l/a/o//i/r/s/e */
case 'l':
  NOT=inst[l].c1;
  switch(inst[l].c2)
  {
    case 'x':
      Result=x[n];
      break;
    case 'y':
      Result=y[n];
      break;
    case 't':
      if((t.state[n])&&(t.t[n]==t.not[n]))
        /* COND:TIMER RUNNING && NOT TRIG ON YET */
        {
          ftine(&buf);
          t.tim[n]=((buf.time)*100)+((buf.millitm)/10)-(t.start[n]);
          if((n<10)&&(t.tim[n]>=(t.num[n]*10)))
            /* CONDITION OF TIMER JUST TRIGING */
            {
              t.t[n]=!(t.not[n]); /* TIM TRIG ON */
              t.tim[n]=t.num[n]*10; /* t.tim ACCUMULATE NOT OVER t.num*10 */
              cha[nch]='T'; nch++; itos(n);
            }
          if((n>9)&&(t.tim[n]>=t.num[n]))
            /* CONDITION OF HIGH SPEED TIMER JUST TRIGING */
            {
              t.t[n]=!(t.not[n]); /* HST TRIG ON */
              t.tim[n]=t.num[n]; /* t.tim ACCUMULATE NOT OVER t.num */
              cha[nch]='T'; nch++; itos(n);
            }
          }
      Result=t.t[n];
      break;
    case 'c':
      Result=ctr.c[n];
  }
  Result=(Result&&(!NOT))|!(NOT&&(!Result)); /* Result XOR NOT */
  goto firstinst;
case 'a':
  NOT=inst[l].c1;
  switch(inst[l].c2)
  {
    case 'x':
      And=x[n];
      break;
    case 'y':
      And=y[n];
      break;
    case 't':
      if((t.state[n])&&(t.t[n]==t.not[n]))
        /* COND:TIMER RUNNING && NOT TRIG ON YET */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ftime(&buf);
    t.tim[n]=((buf.time)*100)+((buf.millitm)/10)-(t.start[n]);
if((n<10)&&(t.tim[n]>=(t.num[n]*10)))
/* CONDITION OF TIMER JUST TRIGING */
{
    t.t[n]=!(t.not[n]);    /* TIM TRIG ON */
    t.tim[n]=t.num[n]*10; /* t.tim ACCUMULATE NOT OVER t.num*10 */
    cha[nch]='T'; nch++; itos(n);
}
    if((n>9)&&(t.tim[n]>=t.num[n]))
        /* CONDITION OF HIGH SPEED TIMER JUST TRIGING */
    {
        t.t[n]=!(t.not[n]); /* HST TRIG ON */
        t.tim[n]=t.num[n]; /* t.tim ACCUMULATE NOT OVER t.num */
        cha[nch]='T'; nch++; itos(n);
    }
}
And=t.t[n];
break;
case 'c':
    And=ctr.c[n];
}
And=((And&&(NOT))||(NOT&&(And)));
Result=Result&&And;
goto firstinst;
'
case 'o':
NOT=inst[1].c1;
switch(inst[1].c2)
{
    case 'x':
        Or=x[n];
        break;
    case 'y':
        Or=y[n];
        break;
    case 't':
        if((t.state[n])&&(t.t[n]==t.not[n]))
            /* COND:TIMER RUNNING && NOT TRIG ON YET */
        {
            ftime(&buf);
            t.tim[n]=((buf.time)*100)+((buf.millitm)/10)-(t.start[n]);
if((n<10)&&(t.tim[n]>=(t.num[n]*10)))
/* CONDITION OF TIMER JUST TRIGING */
{
    t.t[n]=!(t.not[n]);    /* TIM TRIG ON */
    t.tim[n]=t.num[n]*10; /* t.tim ACCUMULATE NOT OVER t.num*10 */
    cha[nch]='T'; nch++; itos(n);
}
        if((n>9)&&(t.tim[n]>=t.num[n]))
            /* CONDITION OF HIGH SPEED TIMER JUST TRIGING */
        {
            t.t[n]=!(t.not[n]); /* HST TRIG ON */
            t.tim[n]=t.num[n]; /* t.tim ACCUMULATE NOT OVER t.num */
            cha[nch]='T'; nch++; itos(n);
        }
}
Or=t.t[n];
break;
case 'c':

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Or=ctr.c[n];
}
Or=((Or&&(!NOT))||(!NOT&&(!Or)));
Result=Result||Or;
goto firstinst;

case 'i':
NOT=inst[1].c1;
switch(inst[1].c2)
{
case 'y':
if(y[n]!=((NOT&&(!Result))||(!NOT&&Result)))
{ y[n]=!(y[n]); cha[nch]='y'; nch++; itos(n); }
break;
case 't':
t.num[n]=inst[l+1].c3; /* t.num = #NUM */
if(!t.rst[n])
{
if(!nfd)
t.t[n]=t.not[n]=NOT;
if(t.state[n]) /* TIMER RUNNING */
{
if(Result) /* CONDITION OF RUNNING */
{
if(t.t[n]==(t.not[n])) /* TIMER NOT TRIG ON YET */
{
ftime(&buf);
t.tim[n]=((buf.time)*100)+((buf.millitm)/10)-t.start[n];
if((n<10)&&(t.tim[n]>=(t.num[n]*10)))
/* CONDITION OF TIM JUST TRIG ON */
{
t.t[n]=!(t.not[n]); /* TIM TRIG ON */
t.tim[n]=t.num[n]*10; /* t.tim ACCUMULATE NOT OVER t.num*10 */
cha[nch]='T'; nch++; itos(n);
}
if((n>9)&&(t.tim[n]>=(t.num[n])))
/* CONDITION OF HST JUST TRIG ON */
{
t.t[n]=!(t.not[n]); /* HST TRIG ON */
t.tim[n]=t.num[n]; /* t.tim ACCUMULATE NOT OVER t.num */
cha[nch]='T'; nch++; itos(n);
}
cha[nch]='t'; nch++; itos(n);
}
} /* END IF TIMER NOT TRIG YET */
}
else /* CONDITION OF BREAK TIMER */
/* TIMER JUST REST */
{
ftime(&buf);
t.tim[n]=((buf.time)*100)+((buf.millitm)/10)-t.start[n];
if(t.t[n]==t.not[n]) /* TIMER NOT TRIG ON YET */
{
if((n<10)&&(t.tim[n]>=(t.num[n]*10)))
/* CONDITION OF TIM JUST TRIG ON */
{
t.t[n]=!(t.not[n]); /* TIM TRIG ON */
t.tim[n]=t.num[n]*10; /* t.tim ACCUMULATE NOT OVER t.num*10 */
cha[nch]='T'; nch++; itos(n);
}
if((n>9)&&(t.tim[n]>=(t.num[n])))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        /* CONDITION OF HST JUST TRIG ON */
        {
            t.t[n]=!(t.not[n]); /* HST TRIG ON */
            t.tim[n]=t.num[n]; /* t.tim ACCUMULATE NOT OVER t.num */
        }
        cha[nch]='t'; nch++; itos(n);
    }
    t.state[n]=0; /* SHOW TIMER RESTING */
}
/* END t.state==1 */
else /* CONDITION OF t.state==0 */
    if(Result) /* CONDITION OF RUNNING TIMER AGAIN */
        /* TIMER JUST RUN */
        {
            ftime(&buf);
            t.start[n]=((buf.time)*100)+((buf.millitm)/10)-t.tim[n];
            t.state[n]=1; /* SHOW TIMER RUNNING */
        }
    }
    l++;
    break;
case 'c':
    old=ctr.count[n];
    if((!(ctr.oldResult[n])&&(Result)&&!(ctr.rst[n])))
/* condition of upper trig counting, and counter not reseted yet */
        ctr.count[n]++;
        ctr.oldResult[n]=Result;
        ctr.num[n]=inst[l+1].c3;
        if(ctr.count[n]>=ctr.num[n]) /* now fp points next inst
            UNCERTAIN USE GETNUM */
            { ctr.count[n]=ctr.num[n]; ctr.c[n]=!NOT; }
        else
            ctr.c[n]=NOT;
            ctr.not[n]=NOT;
            if(ctr.count[n]!=old)
                { cha[nch]='c'; nch++; itos(n); }
            l++;
    }
    goto firstinst;

case 'r': /* case 'r': hasn't NOT */
    n=inst[l].c3;
    if(inst[l].c2=='t') /* timer case */
    {
        if(Result)
        {
            if(!(t.rst[n]))
            {
                /* cond: just rst after trig timer */
                t.rst[n]=1;
                t.t[n]=t.not[n];
                t.tim[n]=t.state[n]=t.start[n]=0;
                cha[nch]='T'; nch++; itos(n);
                cha[nch]='t'; nch++; itos(n);
            }
        }
    }
    else
        t.rst[n]=0;
}
else /* counter case */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรเผยแพร่
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  if(Result)
  {
    ctr.c[n]=ctr.not[n];
    ctr.rst[n]=1;
    if(ctr.count[n]!=0)
    { ctr.count[n]=0; cha[nch]='c'; nch++; itos(n); }
  }
  else
    ctr.rst[n]=0;
} /* end else */
goto firstinst;

case 's': /* case y hasn't NOT, and hasn't y */
  if(y[n]!=1)
  { y[n]=1; cha[nch]='y'; nch++; cha[nch]=itos(n); }
  goto firstinst;

case 'e': /* ASSIGN FILE POINTER POINTS TO THE BEGINING POINT */
  l=-1;
  cha[nch]='e';

  /* STRAT OF DISPLAY MODULE */
  /* Not First Display MODULE */
  if(nfd)
  {
    for(nch=0; cha[nch]!='e';)
    {
      switch(cha[nch])
      {
        case 'y':
          n=getnum1();
          d=0;
          display1(1, 'Y', n);
          break;
        case 't':
          n=getnum1();
          d=-1;
          display1(3, 't', n);
          break;
        case 'T':
          n=getnum1();
          d- ;
          display1(2, 'T', n);
          break;
        case 'c':
          n=getnum1();
          d=-2;
          display1(4, 'C', n);
          d=-4;
          display1(5, 'c', n);
        }
      }
      nch=0;
    }
    /* END OF SWITCH */
    /* END OF FOR */
    /* clear nch */
    /* AND BIG IF */
  }
  else /* CONDITION OF FIRST DISPLAY */
  {
    clrscr();
    display(0, 'X', 31);
    display(1, 'Y', 31);
    d=1;
    display(2, 'T', 13);
    /****** NO HST CASE *****/
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

d=-1;
display(3,'t',13);
d=-2;
display(4,'C',9);
d=-4;
display(5,'c',9);
gotoxy(1,24);
printf("the number of input x is");
nfd=1; /* set first not display */
} /* END OF DIPLAY MODE */

kb=kbhit();
if(kb)
{ /* START OF SCAN INPUT MODE */
  ch=getch(); /* NOT SURE ABOUT THE SPEED */
  if(ch==0x1b) /* FOR EXIT THIS PROGRAM PRESS ESC BUTTOM */
    exit(0);

  if(isdigit(ch))
  {
    if(nn==2) /* START NN=2 */
    {
      if(scannum[1]!='0')
      {
        scannum[0]=scannum[1];
        scannum[1]=ch;
        nn=1;
      }
      else
      {
        scannum[0]=ch;
        nn=0;
      }
    }
    /* END NN=2 */
  }
  else
  {
    if(nn) /* START NN=1 */
    {
      if(scannum[0]!='0')
        scannum[1]=ch;
      else
      {
        scannum[0]=ch;
        nn=0;
      }
    }
    /* END NN=1 */
  }
  else scannum[0]=ch; /* NN=0 */
}
nn++;
scannum[nn]='\0';
xnum = atoi(scannum);
/* not sure that "01" may not conver be 1 */
if(nn)
{
  gotoxy(26,24);
  printf(" ");
  gotoxy(26,24);
  printf("%d",xnum); /* DANGER ABOUT THE EAGE OF XNUM DISPLAY */
}
}

```

๑) สารนี้เป็นเอกสารที่ส/* END OF NUMBER SCANNING AND SHOW THIS NUMBER */

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if((ch=='t')&&(xnum<32))          /* toggle contact x */
    {
        x[xnum]=!x[xnum];
        d=0;
        display1(0,'X',xnum);
    }
} /* END OF SCAN INPUT MODE OR END OF KBHIT() */
goto firstinst;
} /* END OF BIG CASE */
} /****** END OF MAIN *****/

itos(int ni)
{
    if(ni>9)
    {
        cha[nch]=(ni/10)+48;
        nch++;
    }
    cha[nch]=(ni%10)+48;
    nch++;
}

getnum1() /* when use getnum1() nch points new contact */
          /* ,or output for display */
{
    int num;
    nch++; /* across y/t/c */
    num=cha[nch++]-48;
    if(isdigit(cha[nch]))
        num=num*10+cha[nch++]-48;
    return(num); /* now cha[nch] contain next contact or output */
}

display(int a,char b,int limit)
{
    int i,j,k,l;
    i=0;
    for(j=1;j<=4;j++)
    {
        if(i>limit)
            break;
        l=0;
        if(i>=7)
            i++;
        for(;;i++)
        {
            if(i>limit)
                break;
            k=i%8;
            if((k==0)&&(l==1))
            {
                i--;
                break;
            }
            l=1;
            gotoxy(10*k+1,j+5*a+d);
            switch(b)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    case 'X':
        printf("X[%d]=%d", i, x[i]);
        break;
    case 'Y':
        printf("Y[%d]=%d", i, y[i]);
        break;
    case 'T':
        printf("T[%d]=%d", i, t.t[i]);
        break;
    case 't':
        if(i<10)
            if(i<5)
                printf("#[%d]=%d", i, t.tim[i]/10);
            else
                printf("#[%d]=%d", i, t.num[i]-(t.tim[i]/10));
            else
                if(i<12)
                    printf("#[%d]=%d", i, t.tim[i]);
                else
                    printf("#[%d]=%d", i, t.num[i]-t.tim[i]);
                break;
    case 'C':
        printf("C[%d]=%d", i, ctr.c[i]);
        break;
    case 'c':
        if(i<5)
            printf("#[%d]=%d", i, ctr.count[i]);
        else
            printf("#[%d]=%d", i, ctr.num[i]-ctr.count[i]);
    }
}
}
}

```

```

display1(int a, char b, int i)
{
    int j, k;
    k=i%8;
    j=i/8;
    gotoxy(10*k+1, j+(5*a)+d+1);
    switch(b)
    {
        case 'X':
            printf("X[%d]=%d", i, x[i]);
            break;
        case 'Y':
            printf("Y[%d]=%d", i, y[i]);
            break;
        case 't':
            printf("#[%d]= ", i);
            gotoxy(10*k+1, j+(5*a)+d+1);
            if(i<10)
                if(i<5)
                    printf("#[%d]=%d", i, t.tim[i]/10);
                else
                    printf("#[%d]=%d", i, t.num[i]-(t.tim[i]/10));
                else
                    if(i<12)
                        printf("#[%d]=%d", i, t.tim[i]);
                    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    printf("#[%d]=%d",i,t.num[i]-t.tim[i]);
    break;
case 'T':
    printf("T[%d]=%d",i,t.t[i]);
    break;
case 'C':
    printf("C[%d]=%d",i,ctr.c[i]);
    break;
case 'c':
    printf("#[%d] ",i);
    gotoxy(10*k+1,j+(5*a)+d+1);
    if(i<5)
        printf("#[%d]=%d",i,ctr.count[i]);
    else
        printf("#[%d]=%d",i,ctr.num[i]-ctr.count[i]);
}
}
numinst(int C,int L)
{
    return(C+1+L*4);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุป

PLC ที่ได้จำลองงานบนโปรแกรมนี้ สามารถใช้ศึกษาการทำงานของ PLC ได้ดีโดยนำไปใช้ร่วมกับคอมพิวเตอร์ (PC) เป็นการสนับสนุนให้ผู้ที่มีสนใจมีโอกาสได้ศึกษา PLC ในราคาถูกลง

อย่างไรก็ตาม ฝั่งที่ผู้ใช้เป็นคำสั่งนั้นฐานเท่านั้น
ปรกติงานแบบนี้ คงจะเป็นประโยชน์กับผู้สนใจบ้าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์ ต้องขอขอบคุณ คุณปราโมทย์ จุฑษณาศิษย์ ผู้เอื้อเฟื้อ คำปรึกษาและ
อนุญาตให้ใช้เครื่องคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. C.T.Jones, L.A.Bryan, "PROGRAMMABLE CONTROLLER CONCEPTS AND APPLICATION", AN/ASTEC Publication, Atlanta, 1983.
2. Lan G Warnock, "PROGRAMMABLE CONTROLLER OPERATION AND APPLICATION", Prentice Hall, 1988.
3. สุนทรณ กุลพาณิชย์, "พีแอลซีและเทคโนโลยีการใช้งานเบื้องต้น", 2533.
4. กฤษดา วิศวธีรานนท์, "PC ตัวควบคุมซีเคิร์ฟซ์ หลักการทำงานและการประยุกต์", 2534.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้