การประยุกต์เครื่องควบคุมระบบโครงข่ายกับระบบลิฟท์หลายชุด

NETWORK CONTROL SYSTEM APPLICATION FOR MULTI-LIFT

โดย

นาย.อดิศักดิ์ เซ่งเข็ม 36. 012128

นาย.บัญชา นิยมญาติ 36. 013292

นาย.เสนอ พุดเกลี้ยง 36. 013316

วัน เดือน ปี 1 ธ.ค. 2540

เลขทะเบียน 037174

เลขเรียกหนังสือ T 38967 0129 ก.

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037174

ภาควิชา : เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชา : วิศวกรรมการวัดคุมทางอุตสาหกรรม

คณะ : วิศวกรรมศาสตร์

สถาบัน : เทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง : การประยุกต์เครื่องควบคุมระบบโครงข่ายกับระบบลิฟท์หลายชุด

NETWORK CONTROL SYSTEM APPLICATION FOR MULTI - LIFT

จัดทำโดย : 1. นาย อดิศักดิ์ เซ่งเซ็ม 36. 012128

2. นาย บัญชา นิยมญาติ 36. 013292

3. นาย เสนอ พุดเกลี้ยง 36. 013316

..... อาจารย์ที่ปรึกษา

(อาจารย์ สุพรรณ กุลพาณิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ : การประยุกต์เครื่องควบคุมระบบโครงข่าย กับระบบการทำงานของลิฟท์หลายชุด

นักศึกษา : นาย อดิศักดิ์ เซ่งเข็ม 36 012128
นาย บัญชา นิยมญาติ 36 013292
นาย เสนอ พุดเกลี้ยง 36 013316
อาจารย์ที่ปรึกษา : อาจารย์ สุพรรณ กุลพาณิชย์

ระดับการศึกษา : วิศวกรรมศาสตรบัณฑิตทางเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ปีการศึกษา : พ.ศ. 2538

บทคัดย่อ

โครงานนี้เป็นการนำเอา PLC (PROGRAMMABLE CONTROLLER) มาประยุกต์ใช้งาน โดยเป็นการใช้ในระบบโครงข่าย PLC โดยมี C200H เป็นตัวควบคุม ซึ่งได้นำมาประยุกต์กับการควบคุมการทำงานของ ELEVATOR เพื่อเป็นการจัดลำดับความสำคัญการทำงาน และความพร้อมของระบบควบคุม โดยในที่นี้เราจะใช้ ELEVATOR จำนวนสองชุด โดยแต่ละชุดจะมี CPU รุ่น C200H ควบคุมการทำงานอยู่ และการใช้ระบบโครงข่ายนี้ PLC สองชุดสามารถที่จะรับรู้การทำงานของกันและกันได้ หรือ เรียกว่า PC LINK และ สามารถที่จะจัดลำดับความสำคัญของการทำงานของระบบได้และการทำงานของโครงข่ายเราสารมารถที่จะดูได้ทางจอภาพของคอมพิวเตอร์ (COMPUTER) โดยการติดต่อสื่อสาร (INTERFACE) ระหว่าง PLC กับ COMPUTER

ในการประยุกต์การใช้งานของระบบโครงข่ายนี้สามารถที่จะนำเอาหลักการของระบบโครงข่าย PLC ไปใช้ในการควบคุมระบบการทำงานในโรงงานอุตสาหกรรมได้เป็นอย่างดี

ABSTRACT

The Project present about the application of Programmable controller(PC) . This PC used for control two elevators by checked condition of each elevator and link between two Programmable (PC LINK) . Each elevator used C200H as controller. This condition of two elevators we can monitoring on the Computer by interfacing between the Pcs and Computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1. บทนำ	1
บทที่ 2. โครงสร้างและการทำงานของ PC	6
2.1 โครงสร้างโดยทั่วไปของ PC	6
2.2 ตัวอย่างการทำงานของ PC ตามโปรแกรม	10
บทที่ 3. ระบบการสื่อสารข้อมูลระหว่าง PC	17
3.1 ระบบการสื่อสารของ PC	17
3.2 ระดับการดำเนินการ	17
3.3 การออกแบบระบบ	18
3.4 การส่งผ่าน PC	20
3.5 ข้อจำกัดของระบบ	22
3.6 การแลกเปลี่ยนข้อมูล	23
3.7 พื้นที่ของข้อมูล	24
3.8 การติดตั้งและการต่อ	24
3.9 ตัวอย่างของโปรแกรม	26.
บทที่ 4. หน่วย ASCII	29
4.1 หน่วย ASCII	29
4.2 รูปลักษณ์ของระบบ	30
4.3 การแบ่งข้อมูล	31
4.4 การโปรแกรมและการติดต่อ	32
บทที่ 5. รูปลักษณ์ของโครงงาน	35
กิตติกรรมประกาศ	
บรรณานุกรม	
ภาคผนวก	
ภาคผนวก 1 โปรแกรมการติดต่อระหวาง PC กับ PC และสั่งระบบทำงาน	
ภาคผนวก 2 โปรแกรมแสดงภาพการทำงานของระบบทาง COMPUTER	
ภาคผนวก 3 ข้อมูลทางเทคนิค	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในการควบคุมการผลิตของโรงงานอุตสาหกรรมเกือบทุกประเภทในปัจจุบัน จะมีการควบคุมซีเคว้นซ์ (SEQUENCE CONTROL) อยู่เสมอ เช่น การควบคุมการสตาร์ทมอเตอร์ การควบคุมสายพานลำเลียง ระบบกำจัดน้ำเสีย การควบคุมแบบนี้มักจะใช้ รีเลย์แม่เหล็กไฟฟ้า ไทเมอร์ คอนแทคเตอร์ ลิมิตสวิตซ์ปุ่มกด เป็นองค์ประกอบที่สำคัญในการนำมาประกอบเป็นวงจรในตู้ควบคุม และเป็นวิธีการแบบเดียวที่เราคุ้นเคยกันมาแต่ในอดีต

ปัจจุบัน วิทยาการทางด้านอิเล็กทรอนิกส์ และไมโครคอมพิวเตอร์เจริญขึ้นมาก ทำให้เกิดอุปกรณ์ควบคุมชนิดใหม่ที่เรียกว่า **PROGRAMMABLE CONTROLLER** หรือเรียกย่อๆว่า "PC" แนวความคิดของอุปกรณ์ควบคุมแบบใหม่นี้แตกต่างจากวงจรควบคุมแบบใช้รีเลย์แม่เหล็กไฟฟ้าอย่างสิ้นเชิง เป็นการปฏิวัติเทคนิคทางด้านการควบคุมอัตโนมัติจากฮาร์ดแวร์มาเป็นซอฟต์แวร์ ต่อไปการออกแบบวงจรควบคุมซีเคว้นซ์จะเร็วขึ้นอีก 10 เท่า การประกอบวงจรจะง่ายมากเพียงแต่โปรแกรมเข้าไปใน PC นี้เท่านั้น PC จะจดจำวงจรและทำงานตามวงจรได้โดยไม่ผิดพลาด

การใช้ PC ในโรงงานอุตสาหกรรมของประเทศไทยยังอยู่ในขั้นเริ่มต้น แต่ได้รับความนิยมเพิ่มขึ้นเรื่อยๆ และคงจะแทนวงจรรีเลย์แม่เหล็กไฟฟ้าในที่สุด แม้ PC จะสะดวก และใช้งานได้สารพัด แต่ก็ยังต้องใช้งานร่วมกับวงจรรีเลย์พื้นฐานอยู่ดี ดังนั้นการประยุกต์การใช้งานต่างๆ ของ PC ได้พัฒนาเรื่อยๆ ดังนั้นการติดต่อสื่อสารข้อมูลในการควบคุมจึงจำเป็นอย่างมาก ซึ่งการติดต่อสื่อสารข้อมูลนี้จะทำให้เกิดความสะดวกในการควบคุม เพราะจะทำให้เราสามารถรู้ถึงสภาวะการทำงานของแต่ละส่วนของระบบได้ ดังนั้นจุดประสงค์หลักของการทำโครงงานชิ้นนี้ คือ เพื่อพัฒนาการใช้งานระบบการติดต่อสื่อสาร ระหว่าง PC กับ PC โดยนำมาประยุกต์การใช้งานควบคุมการทำงานของระบบควบคุมลิฟท์

การใช้ระบบการติดต่อ (LINK SYSTEM)นี้ จะมีการเซ็ทให้ PC ตัวใดตัวหนึ่งให้เป็นตัวเจ้านาย (MASTER) และตัวที่เหลือเป็นบริวาร (SLAVE) ซึ่งการทำงานจะมีการประมวลผลที่ตัว MASTER ซึ่งเราสามารถที่จะนำข้อมูลจากตัว MASTER นี้ไปติดต่อกับ COMPUTER ได้ ทำให้เราสามารถที่จะรับรู้การทำงานของระบบจากจอภาพ (MONITER) ของ COMPUTER ได้ และ เราสามารถที่จะพัฒนาระบบการควบคุมให้ดีขึ้นได้ในอนาคต

ซึ่งการติดต่อสื่อสารระหว่าง PC กับ PC นี้เราสามารถที่จะควบคุมการทำงานจากจุดหนึ่งเพื่อที่จะไปสั่งให้จุดหนึ่งทำงานตามที่เราต้องการได้ ซึ่งจะสะดวกมากในการควบคุมในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์ของโครงงาน

1. เพื่อเป็นการพัฒนาระบบการควบคุมการทำงานของ PC ให้มีความสามารถยิ่งขึ้น
2. สามารถที่จะนำ PC ไปประยุกต์การใช้งาน ในการควบคุมการทำงานของลิฟท์ในงานจริง ได้อย่างมีประสิทธิภาพ
3. สามารถติดต่อสื่อสารข้อมูลระหว่าง PC กับ PC และ PC กับ COMPUTER เพื่อประโยชน์ในการควบคุมการทำงานได้

ขอบเขตของโครงงาน

1. เขียนโปรแกรมควบคุมลิฟท์ 2 ตัว
2. เขียนโปรแกรมการติดต่อสื่อสารระหว่าง PC กับ PC
3. เขียนโปรแกรมการติดต่อสื่อสารระหว่าง PC กับ COMPUTER
4. เขียนโปรแกรมแสดงผลทาง COMPUTER
5. สร้างระบบจำลองการทำงานของลิฟท์
6. จากนั้นนำระบบต่างๆมาประกอบกัน
7. ป้อนโปรแกรมการทำงานของ PC จาก COMPUTER

ขั้นตอนและการดำเนินงาน

1. สร้างอุปกรณ์จำลองของลิฟท์ โดยลิฟท์นั้นสามารถที่จะทำงานได้เหมือนระบบจริง
2. นำอุปกรณ์จำลองลิฟท์ มาเขียนโปรแกรมการควบคุมการทำงาน โดยในการเขียนโปรแกรมนั้น จะทำการเขียนโปรแกรมระบบการติดต่อสื่อสารไว้ในโปรแกรมเดียวกัน
3. เขียนโปรแกรมส่งค่าสภาวะการทำงาน ไปเก็บไว้ที่ ASCII UNIT
4. เขียนโปรแกรมภาษา BASIC ในการส่งค่าจาก ASCII UNIT ไปยัง COMPUTER
5. เขียนโปรแกรมภาษา C ที่ COMPUTER เพื่อที่จะรับค่าสภาวะการทำงานที่ ASCII UNIT ส่งมา แล้วนำค่าการทำงานไปแสดงผลที่จอภาพของ COMPUTER
6. เขียนโปรแกรมเป็น LADDER DIAGRAM ของ PC บน COMPUTER เพื่อที่จะสามารถ DOWN LOAD ค่าลงใน PC ได้ในโอกาสต่อไป
7. ประกอบระบบการทำงานเข้าด้วยกัน

หมายเหตุ PC ในที่นี้ คือ PROGRAMMABLE CONTROLLER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ที่จะได้รับจากโครงงานนี้

1. ทำให้เข้าใจระบบการติดต่อสื่อสารระหว่าง PC และสามารถที่จะนำทักษะไปพัฒนาในการใช้งานจริงได้
2. สามารถที่จะเขียนโปรแกรมการทำงานที่ยุ่งยากในการนำเอา PC ไปควบคุมได้
3. ทำให้เข้าใจการทำงานของระบบลิฟท์ และสามารถนำไปพัฒนาระบบที่ใหญ่ขึ้นได้
4. เขียนโปรแกรมการติดต่อสื่อสารระบบ PC กับ COMPUTER ได้
5. เขียนโปรแกรมการแสดงผลทาง COMPUTER ได้
6. นำโครงงานนี้ไปพัฒนาในการใช้งานจริงได้ในอนาคต
7. ทำให้เข้าใจระบบการติดต่อสื่อสารระหว่าง CPU ได้

การควบคุมซีเคว้นซ์

การควบคุมซีเคว้นซ์คืออะไร

เป็นการควบคุมลำดับการทำงานของอุปกรณ์ในระบบให้ทำงานตามเงื่อนไข หรือโปรแกรมที่ได้กำหนดไว้แล้ว สัญญาณในการควบคุมอุปกรณ์ส่วนใหญ่จะเป็นสัญญาณเปิด-ปิด (ON - OFF) ให้อุปกรณ์ เช่น มอเตอร์ โซลินอยด์ หลอด ทำงาน หรือหยุดทำงานเท่านั้น ในการควบคุมการทำงานของอุปกรณ์นั้น นอกจากจะพิจารณาเงื่อนไขในการทำงาน แล้วยังพิจารณาถึงในช่วงการทำงานด้วย

การใช้งาน

การควบคุมซีเคว้นซ์มีการใช้งานมากมาย ตั้งแต่เครื่องใช้ไฟฟ้าในบ้าน เช่น เครื่องซักผ้า ตู้เย็น อุปกรณ์ใกล้ตัวเช่น ลิฟท์ บันไดเลื่อน ปั๊มน้ำจนกระทั่งถึงเครื่องจักรในโรงงานอุตสาหกรรม เช่น เครื่องกลึง สายพานลำเลียง หม้อน้ำ และเครื่องกำเหนิดไฟฟ้า เป็นต้น

ใช้งานมากที่สุดในโรงงานอุตสาหกรรม

ปัจจุบันการควบคุมซีเคว้นซ์จะใช้ควบคุมเครื่องจักร อุปกรณ์ในสายการผลิตในโรงงานอุตสาหกรรมอย่างมาก โดยมีจุดมุ่งหมายในการทำให้เครื่องจักรทำงานได้เองอัตโนมัติ และเป็นการลดกำลังคน ประหยัดวัตถุดิบ และการตรวจสอบคุณภาพสินค้าที่ผลิต

ข้อดีของ PC

1. ราคาถูก

PC จะมีราคาถูกกว่าวงจรรีเลย์ ในกรณีที่ใช้รีเลย์มากกว่า 10 ตัวในตู้ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ออกแบบได้ง่าย

ออกแบบตู้ควบคุมทำได้ง่าย ออกแบบวงจรควบคุมทำได้ง่าย การติดตั้ง และการทดสอบระบบก็สามารถทำได้ง่าย

3. ทำได้เร็วขึ้น

การประกอบตู้ควบคุมทำได้เร็วขึ้นมากเนื่องจากใช้อุปกรณ์น้อยลง การเดินสายน้อยลง วงจรซีเคว้นซ์ใช้การโปรแกรมสามารถเปลี่ยนแปลงแก้ไขได้รวดเร็วโดยไม่ต้องเดินส่ายใหม่ ตู้ควบคุมกับเครื่องจักรสามารถแยกกันได้

4. มีขนาดเล็ก และเป็นมาตราฐาน

มีขนาดเล็กลงมากเมื่อเทียบกับตู้ควบคุมรีเลย์ ขนาดของเครื่องเป็นมาตราฐาน ทำให้การทำตู้ควบคุมสะดวกขึ้น โปรแกรมที่ทำขึ้นสามารถโหลดลงใน PC ได้รวดเร็ว และง่าย สามารถทำให้สามารถผลิตตู้ควบคุมที่เหมือนกันได้เร็วมาก

5. ความเชื่อถือสูง

ไม่มีปัญหาเรื่องรีเลย์ ไทเมอร์เสีย ทำโปรแกรมครั้งเดียวสามารถที่จะใช้ได้ตลอดกาล

6. บำรุงรักษา

ชิ้นส่วนน้อยสามารถที่จะบำรุงรักษาได้ง่าย และใน PC สามารถที่จะตรวจเช็คตัวเองทำให้การตรวจซ่อมทำได้ง่ายมาก

7. น้ำหนักเบา

เป็นเครื่องควบคุมที่ประกอบขึ้นด้วยระบบอิเล็คทรอนิกส์ จึงมีน้ำหนักน้อยกระทัดลัด

8. การแก้ไขโปรแกรม

สะดวกในการพัฒนาระบบ เพราะเพียงการแก้ไขโปรแกรม ในขณะที่ HARDWARE คงเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบรีเลย์ กับ PC

อุปกรณ์ควบคุม / หัวข้อ	รีเลย์	PC
ฟังก์ชัน	ใช้ในการควบคุมซับซ้อนได้ ถ้าใช้รีเลย์จำนวนมาก	การควบคุมซับซ้อนเพียงใดก็สามารถโปรแกรมได้
การเปลี่ยนแปลงการควบคุม	เปลี่ยนแปลงโดยการเดินสายใหม่	เปลี่ยนแปลงโดยการเปลี่ยนโปรแกรม
ความเชื่อถือ	ปรกติเชื่อถือได้ แต่มีปัญหาเรื่องจุดต่อสายหลวม และอายุการใช้งานของรีเลย์	องค์ประกอบหลักคือ สารกึ่งตัวนำ จึงไม่มีปัญหาเรื่องจุดจุดต่อสายหลวม มีความเชื่อถือได้สูง
ใช้งานได้เอนกประสงค์	ใช้ได้กับงานที่มีการออกแบบมาเฉพาะเท่านั้น	ใช้งานได้เอนกประสงค์ โดยการโปรแกรม
การขยายระบบ	ทำได้ยาก ต้องเพิ่มอุปกรณ์หรือเปลี่ยนแปลงแก้ไข	ขยายได้เรื่อยๆจนเต็มขีดความสามารถ
การบำรุงรักษา	ต้องตรวจเช็คบ่อยๆ และต้องเปลี่ยนอุปกรณ์ที่มีอายุจำกัด	ซ่อมโดยการเปลี่ยนส่วนประกอบ
การเข้าใจเทคนิค	มีการใช้กันอย่างแพร่หลาย คนส่วนมากเข้าใจเทคนิคการใช้	ยังไม่แพร่หลาย คนส่วนใหญ่ยังไม่เข้าใจเทคนิคการใช้
ขนาด	ใหญ่	เล็ก และไม่ใหญ่ตามวงจรการควบคุม
เวลาที่ใช้ในการออกแบบ	ต้องเขียนแบบจำนวนมาก และต้องใช้เวลาในการประกอบ และการทดสอบ	ออกแบบได้ง่ายแม้เป็นการควบคุมที่ซับซ้อน และการประกอบวงจรควบคุมทำได้ง่ายและเร็ว
จุดคุ้มทุน	รีเลย์น้อยกว่า 10 ตัว	รีเลย์ 10 ตัวขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

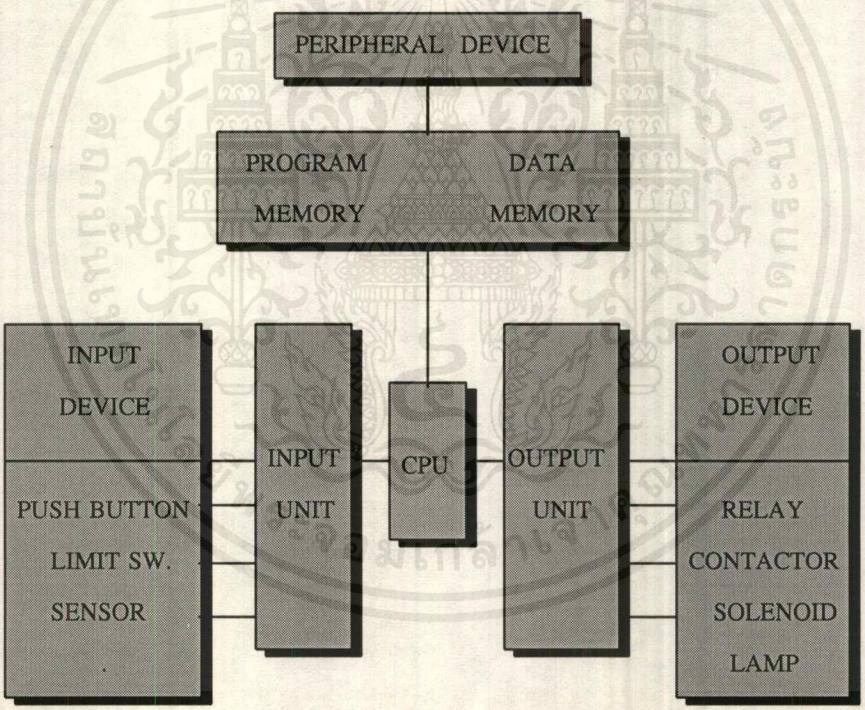
บทที่ 2

โครงสร้างและการทำงานของ PC

2.1 โครงสร้างโดยทั่วไปของ PC

ส่วนประกอบที่สำคัญของ PC แบ่งออกเป็น 4 ส่วน

1. หน่วยประมวลผล (CPU UNIT)
2. หน่วยความจำ (MEMORY UNIT)
3. หน่วยอินพุท - เอาท์พุท (INPUT - OUTPUT UNIT)
4. อุปกรณ์ต่อร่วม (PERIPHERAL DEVICES)



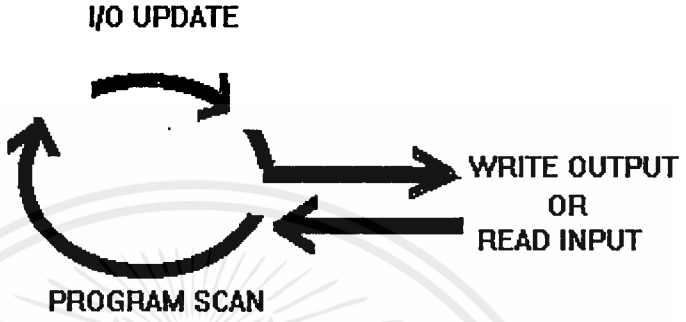
รูปที่ 2.1 โครงสร้างของ PC

2.1.1 หน่วยประมวลผล (CPU UNIT)

หน่วยประมวลผลทำหน้าที่ควบคุมการทำงานของระบบทั้งหมด โดยรับข้อมูลอินพุทเข้ามาทำการประมวลผลแล้วส่งผลที่ได้ออกไป จากนั้นก็วนกลับไปรับข้อมูลอินพุทเข้ามาอีก แล้วจะทำซ้ำๆ ในลักษณะเช่นนี้ไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ CPU จะอยู่ภายใต้การควบคุมของโปรแกรมที่ผู้ใช้ป้อนเข้าไป โดยที่การทำงานในแต่ละรอบนี้เรียกว่า การสแกน (SCANING) สำหรับเวลาของการสแกนขึ้นอยู่กับขนาดของหน่วยความจำ และความเร็วของหน่วยประมวลผล ช่วงเวลาของการสแกนจะทำให้ทราบถึงความสามารถในการตอบสนองต่อการเปลี่ยนแปลงของอินพุท - เอาท์พุท ว่ามีความเร็วเพียงใด



รูปที่ 2.2 การสแกนที่มีผลตอบสนองต่ออินพุท - เอาท์พุท

2.1.2 หน่วยความจำ (MEMORY UNIT)

เป็นองค์ประกอบสำคัญของระบบ เพราะใช้เป็นที่เก็บโปรแกรมและข้อมูล และขนาดของหน่วยความจำจะเป็นสิ่งที่กำหนดความสามารถของระบบ ปกติจะมีขนาดวัดกันเป็นสเต็ปของคำสั่งในการโปรแกรมระบบที่ขนาดของหน่วยความจำมาก จะทำให้ผู้ใช้สามารถเขียนโปรแกรมที่มีความซับซ้อนได้มากขึ้น

หน่วยความจำของ PC แบ่งเป็น

1. RAM (RANDOM ACCESS MEMORY)
2. ERPOM (ERASABLE PROGRAMMABLE READ ONLY MEMORY)

หน่วยความจำ RAM

เป็นหน่วยความจำที่ใช้เก็บโปรแกรมควบคุมที่ป้อนโดยผู้ใช้ให้กับ PC ทั้งนี้เพราะโปรแกรมควบคุมนี้อาจต้องมีการเปลี่ยนแปลงแก้ไข ดังนั้นจำเป็นต้องใช้หน่วยความจำที่สามารถลบข้อมูลเดิม และนำโปรแกรมใหม่เข้าไปเก็บไว้ได้ ในการใช้งานจริงๆแล้ว ต้องมีแหล่งจ่ายไฟสำรองต่อไว้เพื่อป้องกัน ไม่ให้ข้อมูลสูญหายเมื่อไฟฟ้าดับ

หน่วยความจำ EPROM

เป็นหน่วยความจำสำหรับเก็บโปรแกรมที่มีการพัฒนาจนใช้งานได้ดีให้เป็นการถาวร และในการอัดโปรแกรมจะทำได้โดยการถ่ายโปรแกรม จะทำได้โดยการถ่ายข้อมูลจากหน่วยความจำ RAM ลงสู่หน่วยความจำ EPROM โดยอาศัยเครื่องอัดโปรแกรมชนิดพิเศษ (EPROM WRITER) ต่อร่วมกับชุด PC ซึ่งจะทำให้ได้โปรแกรมถาวรดังกล่าวในหน่วยความจำ EPROM นั้น และพร้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะนำมาติดตั้ง (INSTALL) ลงใน PC เพื่อให้ PC ทำงานตามโปรแกรมที่บรรจุ หน่วยความจำประเภทนี้โปรแกรมจะไม่สูญหายเมื่อไฟฟ้าดับ แต่ถ้ามีความจำเป็นที่จะลบโปรแกรมภายในก็สามารถทำได้โดยใช้เครื่องล้างโปรแกรม

2.1.3 หน่วยอินพุท - เอาท์พุท (INPUT - OUTPUT UNIT)

ในที่นี้จะกล่าวเฉพาะหน่วยอินพุท - เอาท์พุท แบบดิจิตอลเท่านั้น

หน่วยอินพุท (INPUT UNIT)

หน่วยอินพุททำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอก ที่เป็นสวิทช์ และตัวตรวจจับต่างๆ แล้วแปลงชนิดของสัญญาณอินพุทดังกล่าว ไม่ว่าจะเป็น AC , DC ให้เป็นสัญญารที่เหมาะสม เพื่อส่งเข้าไปให้แก่หน่วยประมวลผลกลาง ดังนั้นในการเลือกใช้ประเภทของอินพุทนั้น ผู้ใช้จะต้องเลือกให้เหมาะสม และถูกต้องตามประเภทของการใช้งานด้วย ไม่เช่นนั้นอาจจะเกิดความเสียหายได้

ปกติอินพุทที่ดีต้องมีคุณสมบัติดังนี้

1. เปลี่ยนแปลงระดับสัญญาณอินพุทให้เป็นสัญญาณที่เหมาะสมกับ PC

2. การส่งสัญญาณระหว่างหน่วยอินพุทกับหน่วยประมวลผล จะติดต่อกันด้วยลำแสงโดยอาศัยอุปกรณ์ประเภทโฟโต้ทรานซิสเตอร์ ทั้งนี้เพื่อแยกสัญญาณ (ISOLATE)ทางไฟฟ้าให้ออกจากกัน เป็นการป้องปันไม่ให้หน่วยประมวลผล ได้รับความเสียหายเมื่ออินพุทเกิดการลัดวงจร

3.ไม่มีการสั่นของหน้าสัมผัส (CONTACT CHATTEING)

หน่วยเอาท์พุท (OUTPUT)

หน่วยเอาท์พุททำหน้าที่รับค่าสภาวะที่ได้จากการประมวลผลของหน่วยประมวลผล เพื่อนำค่าสภาวะเหล่านั้นไปควบคุมอุปกรณ์ภายนอก พวก รีเลย์ โซลีนอยด์ หลอดไฟแสดงสภาวะ เป็นต้น นอกจากนั้นยังทำหน้าที่แยกสัญญาณ (ISOLATE) ของหน่วยประมวลผลออกจากอุปกรณ์เอาท์พุท โดยปกติแล้วหน่วยประมวลผลนี้มีความสามารถในการขับโหลดด้วยกระแสไฟไฟ้า 1 ถึง 2 แอมแปร์ ในกรณีที่โหลดต้องการกระแสมากไปกว่านี้ ผู้ใช้จะต้องนำไปต่อเข้ากับอุปกรณ์ขับ หรือขยายอีกทีหนึ่ง เช่น รีเลย์ โซลิสเตทรีเลย์ และ คอนแทคเตอร์ เป็นต้น

เอาท์พุทของ PC จะมีอยู่ด้วยกันหลายแบบ ผู้ใช้ต้องพิจารณาเลือกใช้ให้ถูกต้องคือ

เอาท์พุทที่ใช้กับไฟฟ้ากระแสสลับ เอาท์พุทประเภทนี้จะมีอยู่ด้วยกัน 2 ชนิด คือ รีเลย์และไทรแอค สำหรับไทรแอคนั้นเป็นสารกึ่งตัวนำจะถูกนำมาใช้กับโหลดที่มีการเปิด-ปิด อยู่บ่อยๆ เพื่อลดการอาร์คเมื่อมีการตัดต่อวงจรไฟฟ้า

เอาท์พุทใช้กับไฟฟ้ากระแสตรง เอาท์พุทชนิดนี้มีอยู่ 2 ชนิดด้วยกัน คือ ชนิดรีเลย์ และทรานซิสเตอร์ สำหรับเอาท์พุทชนิดทรานซิสเตอร์ ใช้งานที่มีการเปิด- ปิดอยู่บ่อยๆเช่นกัน ตลอดจนการนำไปใช้ขับโหลดที่เป็นวงจรอิเล็คทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำจำกัดความที่เป็นคุณลักษณะของหน่วยอินพุท-เอาท์พุท

1. แรงดันอินพุท (INPUT VOLTAGE RATING) หมายถึง ระดับสัญญาณอินพุทที่หน่วยอินพุทใช้แยกความแตกต่างระหว่างสภาวะ " ON " และสภาวะ " OFF " ของอุปกรณ์ภายนอก ที่สภาวะ ON สัญญาณอินพุทสามารถเปลี่ยนแปลงได้ประมาณ 10% - 50% ของ INPUT VOLTAGE ที่กำหนดไว้

2. กระแสอินพุท (INPUT CURRENT RATING) หมายถึง จำนวนกระแสไฟฟ้าที่หน่วยอินพุทต้องการจากอุปกณ์อินพุท เมื่อสภาวะ ON และ OFF มีค่าประมาณ 7-10 mA

3. ระดับเทรสโฮล (INPUT - THRESHOLD VOLTAGE) หมายถึง ช่วงการเปลี่ยนแปลงระดับสัญญาณอินพุทที่หน่วยอินพุทสามารถแยกความแตกต่างระหว่างสภาวะ ON กับ OFF

4. ช่วงเวลาอินพุท (INPUT DELAY) หมายถึง ช่วงเวลาที่อินพุทรับสภาวะของอุปกณ์อินพุท เช่น หน่วยอินพุท AC หรือ DC ช่วงเวลาอินพุทประมาณ 9-25 mS

5. แรงดันเอาท์พุท (OUTPUT VOLTAGE RATING) หมายถึง ระดับสัญญาณไฟฟ้าของหน่วยเอาท์พุทเมื่อมีสภาวะ ON และ OFF

6. กระแสเอาท์พุท (OUTPUT CURRENT RATING) หมายถึง จำนวนกระแสไฟฟ้าที่สูงสุดที่หน่วยเอาท์พุทสามารถจ่ายให้กับอุปกรณ์ภายนอก

7. กำลังเอาท์พุท (OUTPUT POWER RATING) หมายถึง ขนาดกำลังไฟฟ้าสูงสุดที่หน่วยเอาท์พุทจ่ายให้กับอุปกรณ์ภายนอก

8. อัตราการใช้กระแส (CURRENT REQUIREMENT) หมายถึง จำนวนกระแสไฟฟ้าที่หน่วยอินพุท และเอาท์พุทต้องการจากหน่วยจ่ายกำลังงาน

9. อัตราการทนกระแสสูงสุด (MAXIMUM SURGE CURRENT) หมายถึง จำนวนกระแสไฟฟ้าสูงสุด ที่มีค่ามากกว่ากระแสเอาท์พุท และหน่วยเอาท์พุทสามารถจ่ายให้กับอุปกรณ์ภายนอก ได้โดยไม่เกิดความเสียหายชั่วระยะเวลาหนึ่ง เช่น 20 A / 1 S หมายถึง หน่วยเอาท์พุทสามารถจ่ายกระแสเอาท์พุทได้ 20 A เป็นเวลา 1 วินาที ก่อนหน่วยเอาท์พุทจะเกิดความเสียหาย

2.1.4 อุปกรณ์ต่อร่วม (PERIPHERAL DEVICE)

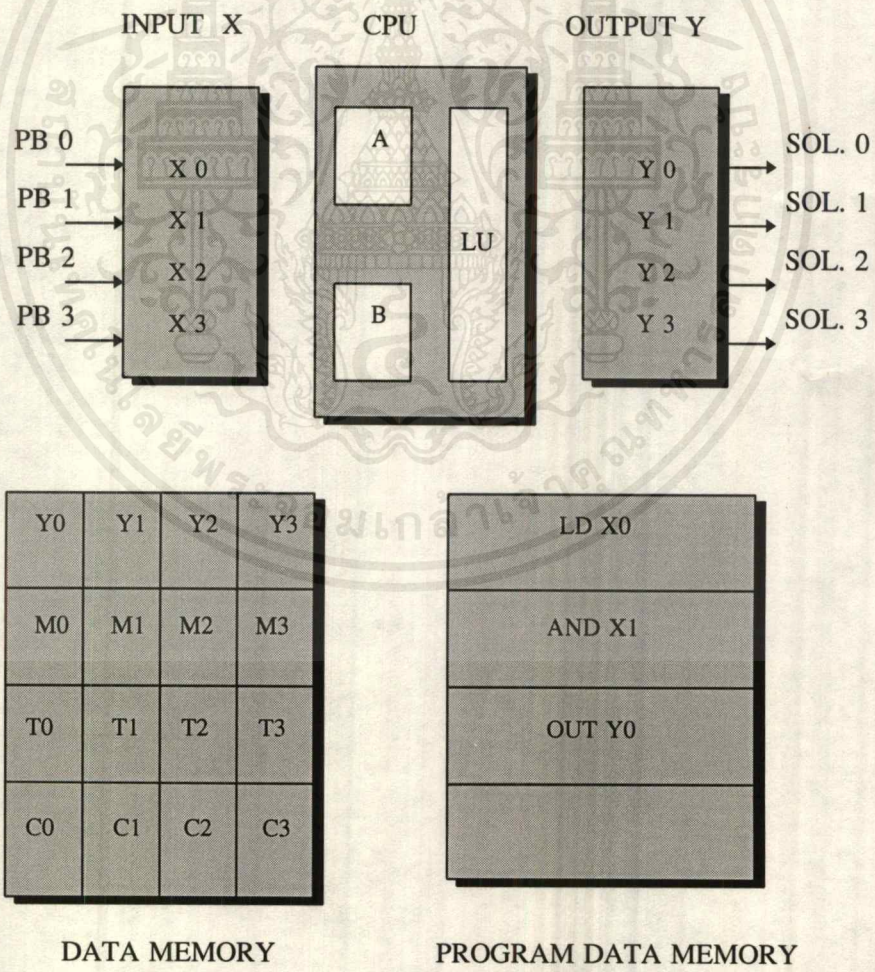
อุปกรณ์ที่ใช้ต่อร่วมกับ PC มีอยู่ด้วยกันหลายชนิดหลายประเภท ทั้งนี้ก็เพื่ออำนวยความสะดวกในการพัฒนาโปรแกรม ทำให้เขียนโปรแกรมสำเร็จรูปในเวลาอันสั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทและหน้าที่ของอุปกรณ์ต่อร่วมดังตารางนี้

อุปกรณ์ต่อร่วม	หน้าที่การใช้งานเกี่ยวกับโปรแกรม				
	ป้อน	แก้ไข	โหลดใหม่	พิมพ์	สภาวะ
1. PROGRAMMING CONSOLE	*	*			*
2. EPROM WRITER			*		
3. PRINTER				*	
4. GRAPHIC PROGRAMMING	*	*			*
5. CRT MONITOR	*	*			*
6. AUDIO CASSETTE			*		
7. LADDER SOFEWARE	*	*	*	*	*

2.2 ตัวอย่างการทำงานของ PC.ตามโปรแกรม



รูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

A: A REGISTER

B: B REGISTER

LU: LOGIC UNIT

X: INPUT MULTIPLEXER

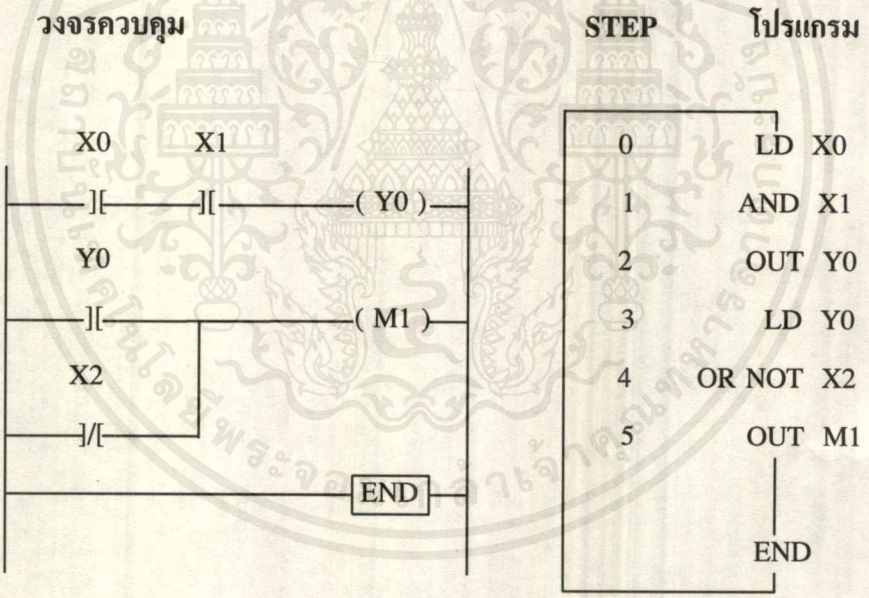
Y: OUTPUT DEMULTIPLEXER

Y0 - Y1: OUTPUT STATUS DATA

M0 - M1: AUXILIARY RELAY STATUS DATA

C0 - C1: COUNTER STATUS DATA

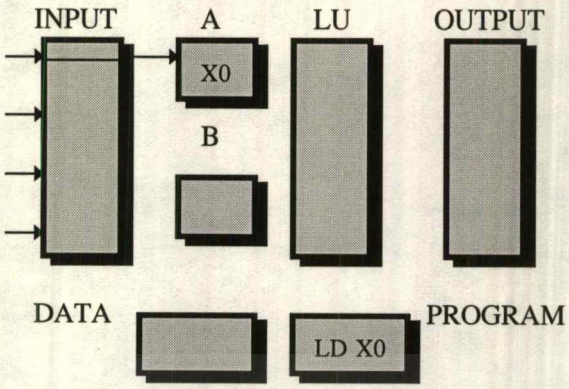
T0 - T1: TIMER STATUS DATA



ตัวอย่างวงจรควบคุมและรายละเอียดโปรแกรมของวงจรตัวอย่างแต่ละ STEP เราสามารถแสดงการเคลื่อนไหวของข้อมูลใน REGISTER และ MEMORY ภายในดังนี้

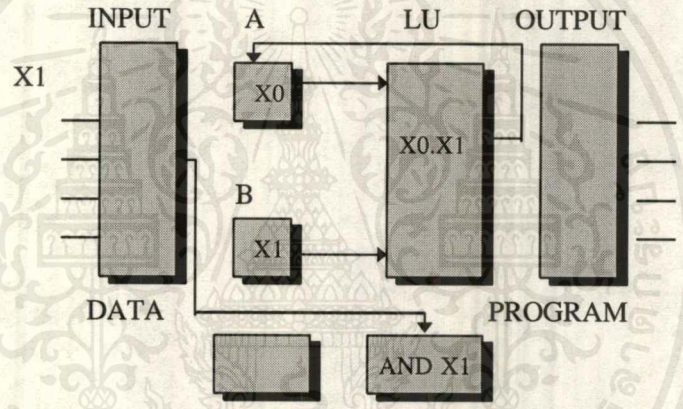
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STEP 0



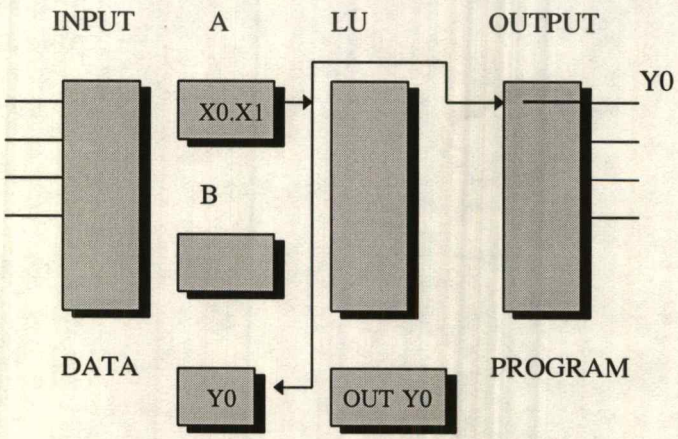
ค่าสภาวะของ X0 จะถูกนำมาเก็บไว้ใน REGISTER A.

STEP 1



ค่าสภาวะ X1 ถูกนำเข้าไปไว้ที่ REGISTER B. เสร็จแล้ว LOGIC UNIT (LU) จะทำหน้าที่นำค่าสภาวะภายใน REGISTER ทั้งสองมาทำเงื่อนไข AND กัน แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ที่ REGISTER A. เช่นเดิม

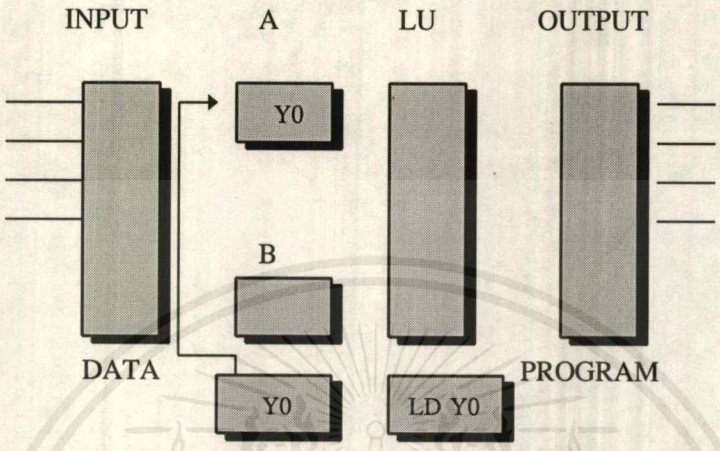
STEP 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

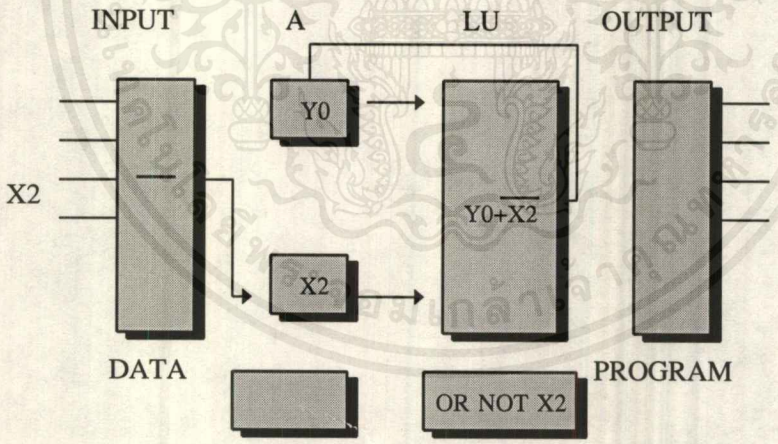
นำค่าสภาวะภายในของ REGISTER A. ที่มีผลมาจาก STEP ที่ผ่านมา ออกไปยังเอาท์พุทตำแหน่ง Y0 ในขณะเดียวกันก็นำมาเก็บไว้ที่ DATA MEMORY ด้วย

STEP 3



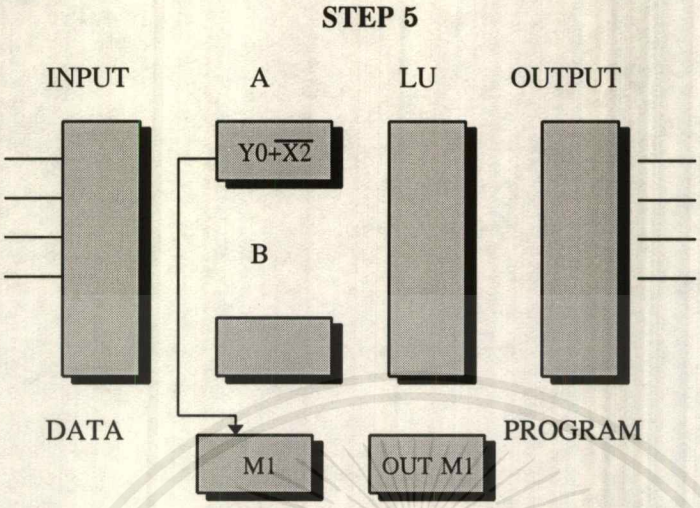
นำค่าสภาวะของ Y0 ที่อยู่ภายใน DATA MEMORY ไปไว้ที่ REGISTER A.

STEP 4



ค่าสภาวะของ X2 ถูกนำไปใช้ที่ REGISTER B แต่เนื่องจากโปรแกรมที่เขียนให้กับ PC จะเป็นการ OR NOT ดังนั้น LOGIC UNIT จึงทำการเปลี่ยนค่าสภาวะ X2 ให้เป็นค่าตรงกันข้าม $\overline{X2}$ เสียก่อนแล้วจึงทำเงื่อนไข OR จากนั้นนำค่าสภาวะผลลัพธ์ที่ได้ไปเก็บไว้ใน REGISTER A เช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



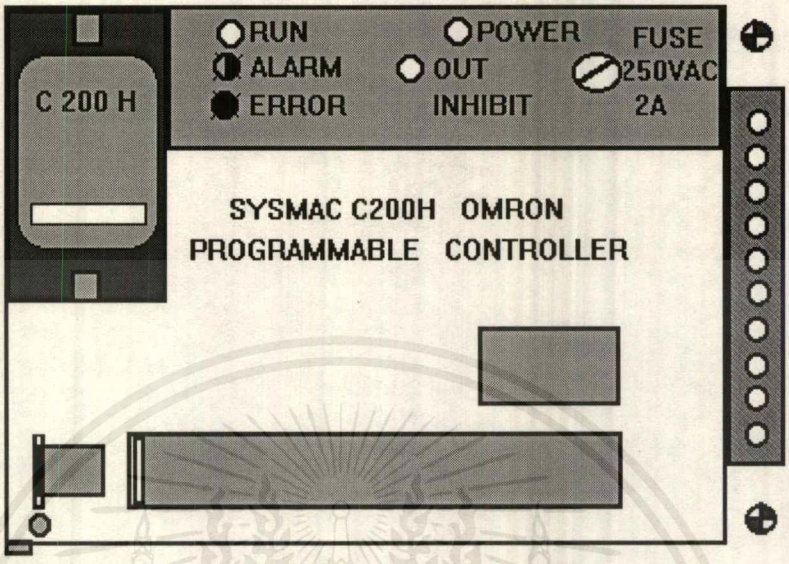
รูปที่ 2.4 การเคลื่อนไหวของข้อมูลเมื่อทำงานตามคำสั่ง

นำค่าสภาวะภายใน REGISTER A ที่ได้จาก STEP ที่ผ่านมาไปเก็บไว้ที่ AUXILIARY RELAY หรือ รีเลย์ช่วย แต ในที่นี้จะหมายถึงรีเลย์ภายใน ซึ่งเป็นส่วนหนึ่งของ MEMORY

นี่เป็น ตัวอย่าง การทำงานของโปรแกรมที่มีการทำงานไม่เท่าใดนัก ในทำนองเดียวกัน โปรแกรมที่มีการทำงานที่ยุ่งยากไปกว่านี้ จะมีลำดับขั้นที่ไม่แตกต่างไปจากตัวอย่างเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปลักษณ์ของ PC รุ่น C 200 H



จากรูป สัญลักษณ์ที่แสดงบอกสถานะการทำงานมีดังนี้

ตัวแสดง	หน้าที่
POWER	จะติดเมื่อมีการจ่ายไฟให้แก่ CPU
RUN	จะติดเมื่อ CPU กำลังดำเนินการปกติ
ALARM / ERROR	ALARM : จะกระพริบเมื่อตรวจพบ NON - FATAL ERROR แต่ PC จะทำงานปกติ ERROR : จะติดเมื่อตรวจ FATAL ERROR และ ไม่สามารถที่จะเข้า MODE RUN ได้
OUT INHIBIT	จะติดเมื่อเอาท์พุทบิตหยุดทำงาน SR 25215 จะทำงาน และ เอาท์พุททุกๆตัวจะ OFF ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นที่หน่วยความจำ (MEMORY AREAS)

พื้นที่หน่วยความจำจะแบ่งเป็นพื้นที่ต่างๆ ดังตารางข้างล่าง โดยจะมีจำนวนเวิร์ด (WORD) และหน้าที่การทำงานในลักษณะต่างๆกัน

พื้นที่	สัญลักษณ์	ขนาด	หน้าที่
รีเลย์ภานใน (INTERNAL RELAY	IR	WORDS: 000 ถึง 235 BITS : 0000 ถึง 23515	ใช้ในการควบคุมจำนวนจุดอินพุท - เอาท์พุท , บิตอื่นๆ , ไทร์เมอร์ , และ เคาท์เตอร์ และ ใช้ในการพักข้อมูลชั่วคราว
รีเลย์ชนิดพิเศษ (SPECIAL RELAY)	SR	WORDS: 236 ถึง 255 BITS : 23600 ถึง 25507	บรรจุเกี่ยวกับเวลา , แฟลก , บิตควบคุม และสถานะข้อมูล
รีเลย์ช่วย (AUXILIARL RELAY)	AR	WORDS: AR00 ถึง AR27 BITS : AR00 ถึง AR2715	บรรจุแฟลก และบิตสำหรับฟังก์พิเศษ และบอกสถานะของระบบเมื่อระบบเกิดการใช้ไม่ได้บางจุด
หน่วยความจำข้อมูล (DATA MEMORY)	DM	อ่าน/เขียนDM0000ถึงDM0999 อ่านอย่างเดียว DM 1000 - DM 1999	ใช้สำหรับการเก็บขอมูลภายในและดำเนินการจัดการ
โฮลดิ้งรีเลย์ (HOLDING RELAY)	HR	WORDS: HR00 ถึง HR 99 BITS : HR0000 ถึง HR9915	ใช้สำหรับการเก็บข้อมลและรักษาข้อมูลภายในไม่ให้หายเมื่อระบบหยุดทำงาน
ตัวตั้งเวาลา / ตัวนับวเลา (TIMER / COUNTER)	TC	TC 000 - TC 511	ใช้ในการกำหนดตัวตั้งเวลาและตัวนับเวลาและเป็นตัวจัดการแฟลกให้สมบูรณ์ PVและ SV เมื่อ WORDS นี้ถูกใช้ระบบจะแสดงการนับโดยตั้งเวลาและตัวนับเวลา
รีเลย์ที่ใช้ในการ (LINK RELAY)	LR	WORDS: LR00 ถึง LR 63 BITS : LR0000 ถึง LR6315	สามารถใช้ประโยชน์สำหรับการกำหนดให้บิตทำงาน
รีเลย์ชั่วคราว (TEMPORARY RELAY)	TR	TR 00 - TR 07 (สำหรับบิตเท่านั้น)	ใช้ในการเก็บข้อมูลไว้ชั่วคราวและสามารถที่จะกู้คืนมาใช้ได้ บิตนี้สามารถใช้ในการโหลด และออกจากระบบได้
หน่วยความจำโปรแกรม (PROGRAM MEMORY)	UM	UM : จะใช้งานได้อย่างไม่จำกัด	จะบรรจุโปรแกรมให้ปฏิบัติการโดย CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระบบการสื่อสารข้อมูลระหว่าง PC

3.1 ระบบการสื่อสารของ PC (PC LINK SYSTEM)

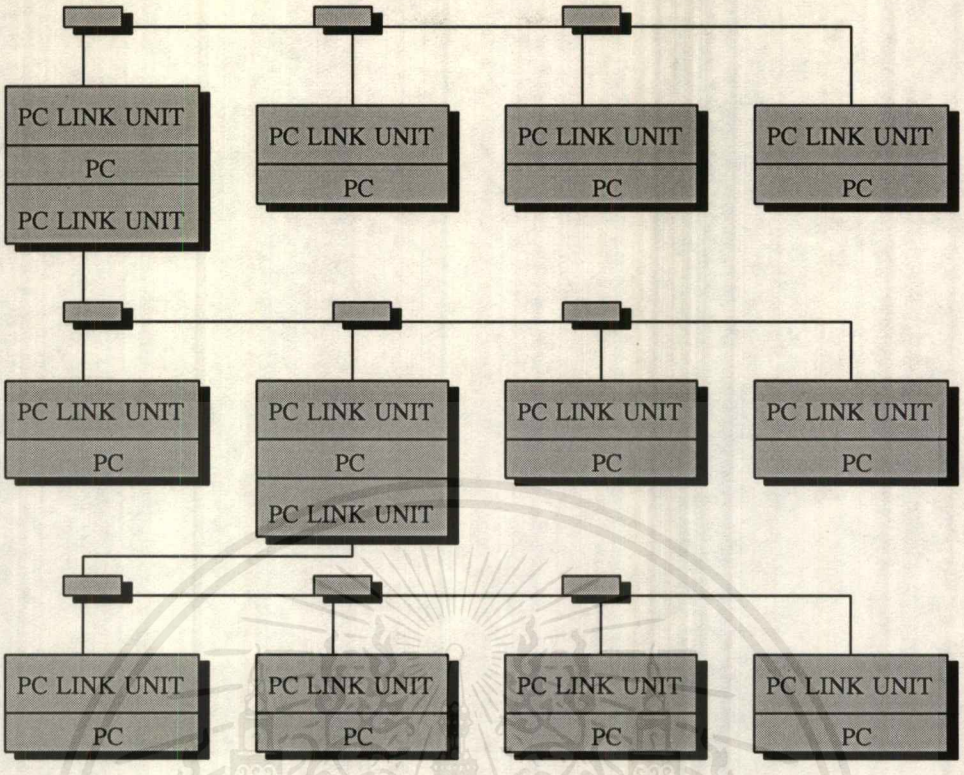
PC LINK SYSTEMS คือ การติดต่อกันของ C- SERIES PC จำนวน 2 ตัว หรือมากกว่าผ่านทาง PC LINK UNITS โดยการส่งผ่านข้อมูลจะส่งผ่านทาง LR AREA ของ PC แต่ละตัว PC LINK SUBSYSTEM สามารถที่จะติดตั้งภายใน PC ใน PC LINK SYSTEM แม้ว่าระดับการดำเนินการจะแตกต่างกัน PC แต่ละตัวใน PC LINK SYSTEM จะมีการแลกเปลี่ยนข้อมูลกันอย่างอัตโนมัติใน SYSTEM นั้นๆ

3.2 ระดับของการดำเนินการ (OPERATING LEVELS POLLING)

PC LINK UNITS ทุกตัวใน SYSTEM จะต้องมีการกำหนด UNIT NUMBERS ซึ่งจะทำการกำหนดผ่านทาง LR AREA ของแต่ละตัว อย่างไรก็ตามการติดต่อกันของ PC ใน PC LINK SYSTEM จะต้องทำการ SET PC LINK UNIT ตัวใดตัวหนึ่งให้เป็น POLLING UNIT (UNIT#0) และ PC LINK UNIT ตัวอื่นๆที่เหลือต้อง SET ให้เป็น POLLED UNIT (มากกว่า UNIT # 0) POLLING UNIT ของแต่ละ PC LINK SYSTEM ไม่สามารถที่ควบคุม SYSTEM อื่นได้

PC ตัวเดียวสามารถที่จะติดต้ัง PC LINK UNIT ได้สูงสุด 2 ตัว และถ้าสอง PC LINK UNIT ถูกติดตั้งใน PC ตัวใดตัวหนึ่ง ในระบบ ระบบนั้นจะถือว่าเป็นระบบ MULTILEVEL และจะต้อง SET เป็น SYSTEM แต่ละ SYSTEM จะมี POLLING UNIT ของตัวมันเอง ในกรณีที่มีหลาย SYSTEM PC ตัวที่มี PC LINK UNIT ติดตั้งอยู่ 2 ตัว จะมีจำนวนน้อยกว่า SYSTEM อยู่หนึ่ง ซึ่งสอง PC LINK UNIT ใน PC เดียวกันนั้น จะต้อง SET LEVEL ให้แตกต่างกัน คือ LEVEL #0 และ LEVEL # 1 PC LINK SYSTEM มี 3 SUBSYSTEM ดังแสดงข้างล่าง PC LINK UNIT ใน SUBSYSTEM ตัวใดตัวหนึ่งต้องกำหนดให้เป็น POLLING UNIT และ ตัวอื่นๆ จะต้อง SET ให้เป็น POLLED UNIT และ LINK ADAPTER จะใช้เป็นตัวเชื่อมต่อ PC LINK UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงการต่อเชื่อม PC

3.3 การออกแบบระบบ (SYSTEM DESIGN)

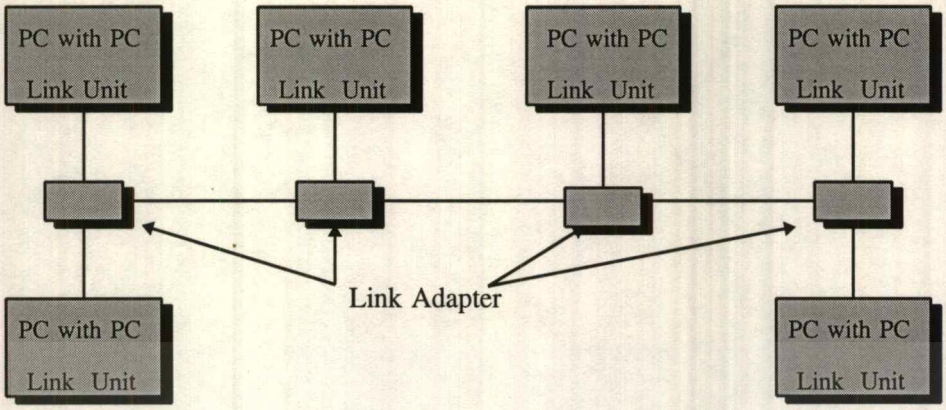
3.3.1 รูปแบบของระบบ (SYSTEM CONFIGURATION)

PC LINK UNIT จะติดตั้งบน PC RACKS และต่อให้ PC LINK UNIT ตัวอื่นๆ และ PC LINK UNIT แต่ละตัวจะบรรจุ BUFFER ซึ่งข้อมูลจะทำการส่ง และรับกันระหว่าง PC LINK UNIT ที่ติดต่อกัน ซึ่ง C500-LK009-V1 สามารถที่จะใช้กับ PC รุ่น C500, C1000H, C2000H ส่วน C200H-LK401 สามารถใช้กับ C200 เท่านั้น

3.3.2 ระบบการสื่อสารหลายตัว (MULTILINK SYSTEM)

PC LINK UNIT แต่ละตัวจะต่อได้หนึ่งครั้ง โดยเราไม่สามารถที่จะต่อระบบที่มีมากกว่า 2 ตัวได้โดยตรง ดังนั้นถ้าหากมีระบบที่มี PC LINK UNIT หลายตัวเราสามารถที่จะติดต่อกันได้ โดยจะผ่านทาง LINK ADAPTER ซึ่งลักษณะการต่อแสดงดังรูปข้างล่าง ซึ่งจะเห็นว่าสามารถที่รวม PC 6ตัว ไว้ใน PC LINK SYSTEM ได้ ระบบเหล่านี้จะมีการเตรียมการป้องกันระบบ FAIL ในสายที่สมบูรณ์ เช่น ถ้าการสื่อสารสาขาใดสาขาหนึ่งขาดจากการติดต่อข้อมูล ระบบที่เหลือจะยังคงมีการติดต่อสื่อสารผ่าน PC LINK UNIT ไปยัง POLLING UNIT ได้สมบูรณ์เหมือนเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงระบบ LINK หลายตัว

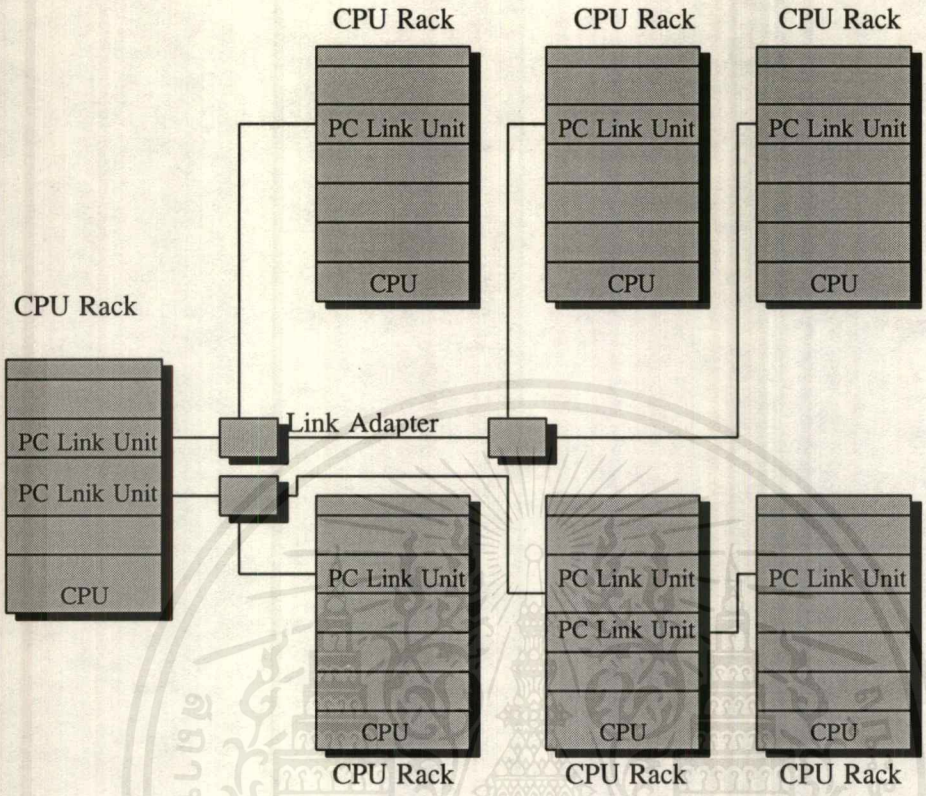
3.3.3 ระบบการติดต่อหลายระดับ (MULTI LEVEL SYSTEM)

PC LINK UNIT 2 ตัวสามารถที่จะติดตั้งบน PC หนึ่งตัวได้ ซึ่งถ้ามีการติดตั้งสอง PC LINK UNIT บน PCหนึ่งตัว หรือถ้าระบบใดมี PC LINK UNIT ติดตั้งบน PC ตัวใดตัวหนึ่ง มีสองตัว เราสามารถบอกได้เลยว่าระบบนั้นเป็นระบบ MULTILEVEL SYSTEM

ในแต่ละกลุ่มของ PC จะติดต่อโดย PC LINK UNIT และ จะมีการแบ่งส่วนของ LR AREA ให้เป็นส่วนของ PC LINK SYSTEM เดียวกัน ส่วน PC ตัวที่มีการติดตั้ง 2 PC LINK UNIT นั้นจะเป็นส่วนหนึ่งของทั้งสอง PC LINK SYSTEM ซึ่งโดยแต่ละSYSTEM จะมีการขยายจาก PC และ PC LINK UNIT แต่ละ SYSTEM จะมี POLLING UNIT ของตัวมันเอง และจะต้องกำหนดให้ OPERATING LEVEL ของสอง SYSTEM ใน PC ตัวเดียวกัน ให้แตกต่างกัน การดำเนินการทาง LEVEL นี้ไม่ได้หมายความว่าจะเป็นการบังคับ หรือมีผลทางใดทางหนึ่งของ SYSTEM ยกเว้นเป็นการตัดสินใจของ LR WORDS ของ SYSTEM

ตัวอย่าง ดังรูปที่ 3.3 จะเป็นระบบ MULTI LEVEL PC LINK ซึ่งมี 3 SYSTEM แม้ว่า SYSTEM 1 และ SYSTEM 3 จะเหมือนกันแต่ไม่ได้มีความสัมพันธ์อะไรกันเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
037174



รูปที่ 3.3 แสดงระบบ MULTI-LEVEL

3.4 การส่งผ่านPC (TRANSFER PC)

PC ตัวที่มีการติดตั้ง PC LINK UNIT 2 ตัว เราเรียกว่า TRANSFER PC เพราะว่าในการส่งข้อมูลระหว่างสอง SYSTEM จะขึ้นอยู่กับมัน ซึ่งการเขียนโปรแกรม และการส่งข้อมูลระหว่าง SYSTEM จะขึ้นอยู่กับการจัด WORDS และในการดำเนินการส่งก็จะใช้โปรแกรมนั่นเอง

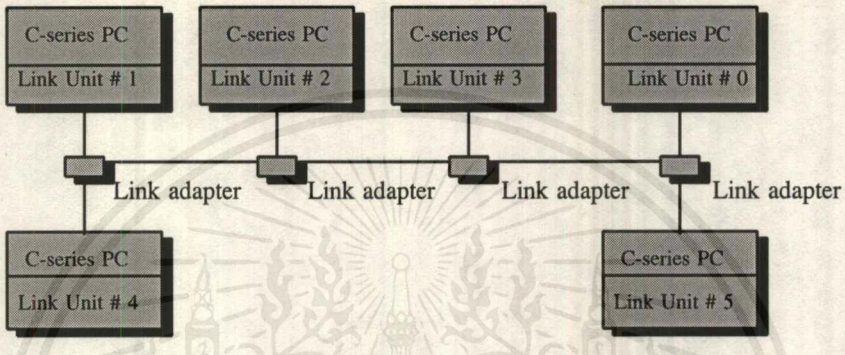
3.4.1 การต่อสายติดต่อ (TRANSMISSION DISTANCE)

ความยาวทั้งหมดของสายเคเบิล (RS - 485, RS - 422) จะต้องไม่เกิน 500 เมตร แต่ละสาขาของสายจาก LINK ADAPTER ถึง PC LINK UNIT จะต้องไม่เกิน 10 เมตร ส่วนระยะการส่งที่มากกว่านี้สามารถทำได้โดยการใช้ลักษณะการ LINK ระหว่าง PC LINK UNIT

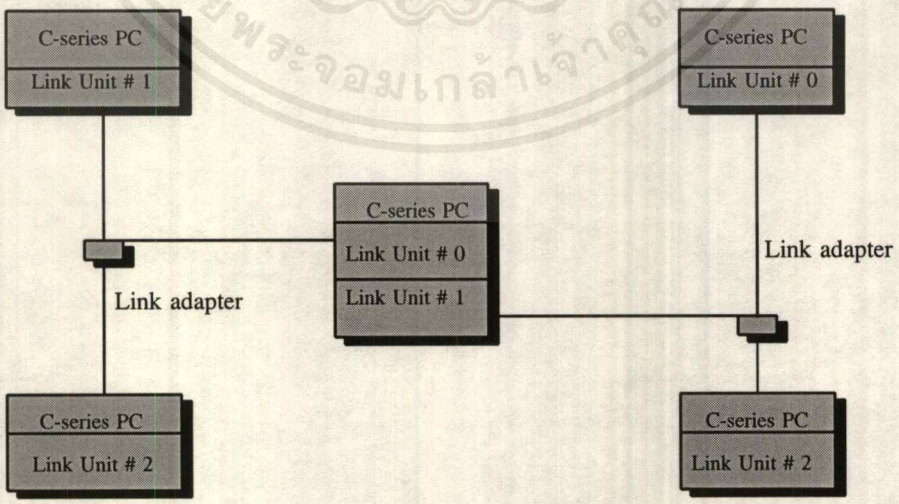
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่แสดงข้างล่างเป็นส่วนหนึ่งของการต่อใน PC LINK SYSTEM

ระบบอันดับหนึ่ง (SINGLE - LEVEL SYSTEM)

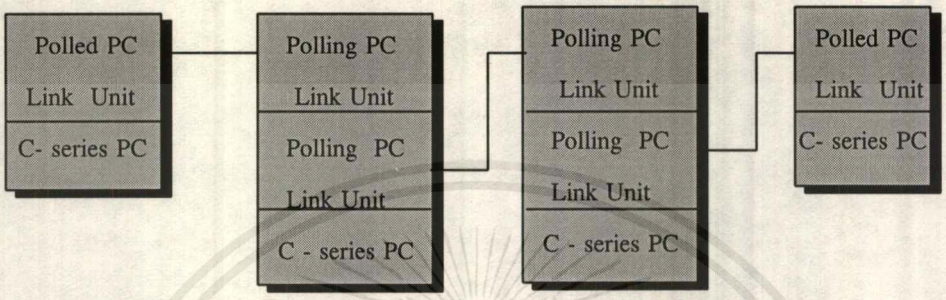


ระบบอันดับสอง (TWO - LEVEL SYSTEM)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบอันดับสาม (THREE - LEVEL SYSTEM)



รูปที่ 3.4 แสดงลักษณะของ PC LINK SYSTEM

3.5 ข้อจำกัดของระบบ (SYSTEM LIMITATION)

จำนวนสูงสุดของ PC ที่จะใช้ใน PC LINK SYSEM จะจำกัดโดยจำนวนของ LR AREA ที่จะใช้ประโยชน์โดยการกำหนดของ LEVEL , SPACE ของ PC ที่ใช้และ MODE ของการ SET บน PC LINK UNIT

ในการกำหนด NUMBER ของ PC LINK UNIT จะต้องน้อยกว่าจำนวนของ PC ใน SYSTEM อยู่หนึ่ง การที่จะกำหนด NUMBER มากกว่าระบบจะไม่ยอมรับ จำนวนสูงสุดของหน่วยที่ใช้ประโยชน์ในระบบของ PC LINK SYSTEM แสดงดังตารางข้างล่าง

	POLLED UNITS			
	LK 401 on C200H	LK009-V1 on C500	LK009-V1 on C1000H, C200H	MAX. total
MULTI - LEVEL	16	8	16	16
SINGLE- LEVEL	32	8	32	32

รูปที่ 3.5 POLLING UNIT : C200H - LK401 PC LINK UNIT ON C200H PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนด NUMBER ของแต่ละ PC LINK UNIT จะเป็นตัวกำหนดส่วนของ LR AREAของ PC ซึ่งแต่ละ NUMBER จะเป็นตัวบ่งบอกให้ทราบว่า PC LINK UNIT ตัวใดเป็น POLLING UNIT และเป็น POLLED UNIT และจะเป็นการ SET LR AREA โดยอัตโนมัติด้วย

PC LINK UNIT สามารถที่จะส่งข้อมูลได้ 2-32 WORD (32-512 บิท) ระหว่าง PC การแบ่ง LR AREA ขึ้นอยู่กับจำนวนของ PC LINK UNIT และจำนวนของ SUBSYSTEM ในระบบด้วย และ LR AREA ส่วนที่ไม่ใช้จะเป็น WORK AREA ซึ่งทั้งนี้ขึ้นอยู่กับการ SET SWITCH

ตารางการ SET ระบบสำหรับ PC LINK UNIT 2 ชุด สำหรับ C200H

จำนวนของ PC LINK UNIT / LR bits / Unit	2.
LR Word	
00 - 31	UNIT # 0
32 - 63	UNIT # 1

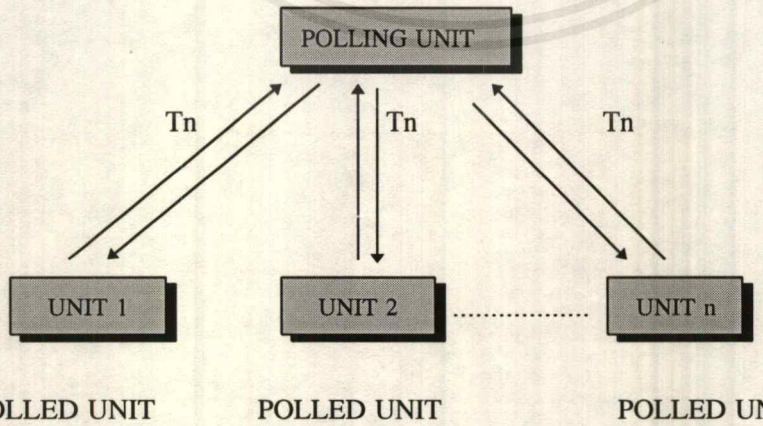
3.6 การแลกเปลี่ยนข้อมูล (DATA EXCHANGE)

ระบบควบคุมใน PC LINK UNIT จะเป็นการควบคุมในลักษณะจุดศูนย์กลางไปยังสาขาย่อย โดย POLLING UNIT กับ POLLED UNIT โดย POLLED UNIT จะรองรับพื้นที่ของ LR AREA ที่เป็น WRITE AREA

POLLING UNIT จะส่ง LR DATA ให้ PC ตัวอื่น ๆ และจะรับ LR DATA จาก POLLED UNIT โดยใช้เวลาดังสมการ

TOTAL POLLING TIME = Tn * จำนวนของ PC LINK UNIT +10 mS

Tn คือ เวลาการส่งของแต่ละ PC



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกำหนดได้ดังตาราง

จำนวนของ LR BIT	32	64	128	256	512
เวลาในการส่ง mS	2.2	2.5	2.8	3.8	5.8

จะเห็นได้ว่า PC LINK 2 ตัวจะใช้เวลาในการส่ง5.8 mS

TOTAL POLLINK TIME = 5.8 * 2+10 = 21.6 mS

3.7 พื้นที่ของข้อมูล (DATA AREA)

PC LINK UNIT จะใช้ประโยชน์ของ DATA AREA ในการติดต่อข้อมูลของ PC และใช้ในการดำเนินงาน ซึ่งแสดงได้ดังตารางข้างล่าง

	C 200 H
LR LAREA	LR 0000 - LR 6315 (1,024 bits)
ERROR FLAGS	24708 - 24715 24808 - 24815 24908 - 24915 25008 - 25015
PC RUN FLAGS	24700 - 24707 24800 - 24807 24900 - 24907 25000 - 25007

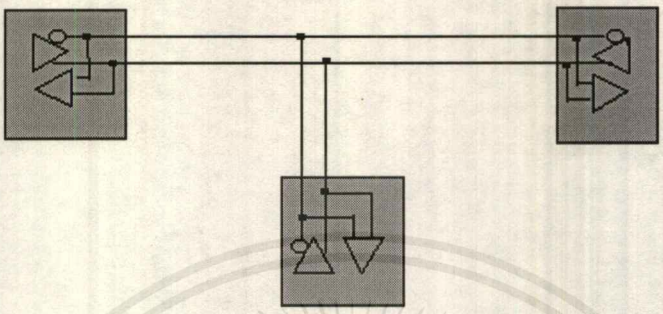
3.8 การติดตั้งและการต่อ (MOUTING AND CONNECTION)

PC LINK UNIT สามารถติดตั้งบน SLOT ของ CPU RACK หรือ ที่ส่วนขยาย I/O RACE และการต่อจะใช้ PORT RS - 485 เป็นตัวเชื่อม PC LINK UNIT กับ LINK ADAPTER โดยใช้ ELECTRICAL CABLE ดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC LINK UNIT LINK ADAPTER PC LINK UNIT

LK 009-V1 AL 001 LK 009-V1



รูปที่ 3.6 แสดงการต่อ PC LINK UNIT

PC Link Unit connector or AL 001 Link Adapter

Pin No	Signal name
1	
2	
3	
4	
5	Transfer data B (DB)
6	
7	Frame ground (FG)
8	
9	Transfer data A(DA)
Hood	Frame ground (FG)

AL 001 Link Adapter

Pin No	Signal name
1	
2	
3	
4	
5	Transfer data B (DB)
6	
7	Frame ground (FG)
8	
9	Transfer data A(DA)
Hood	Frame ground (FG)

รูปที่ 3.7 แสดงการต่อ CONNECTOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

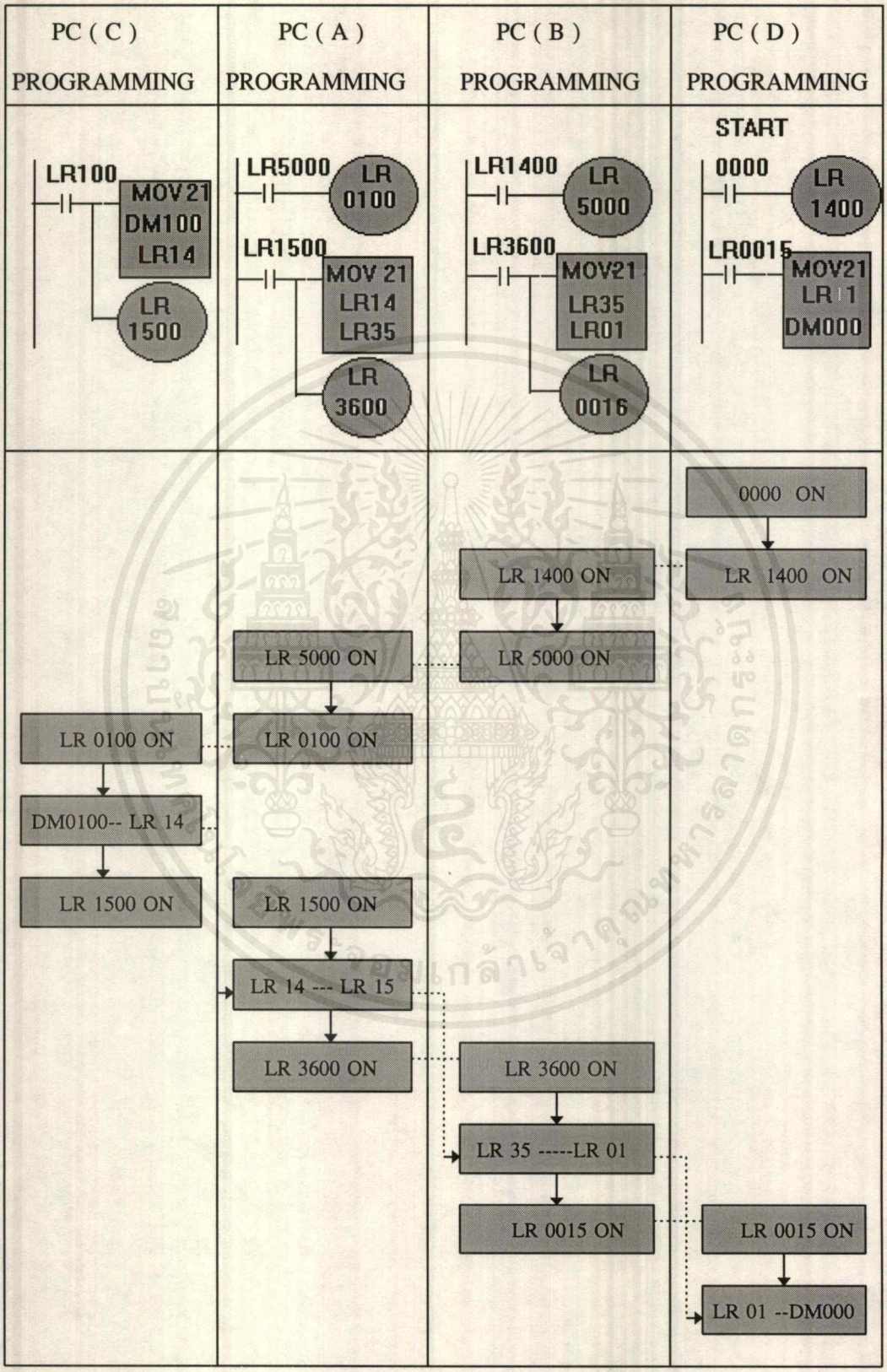
3.9 ตัวอย่างโปรแกรม (PROGRAMMING EXAMPLE)

PROGRAMMING EXAMPLE 1

จากโปรแกรมข้างล่าง เป็นการส่งข้อมูลจาก PC (C) ไปยัง PC (D) โดยเมื่อมีการ START ON INPUT 0000 ที่ PC(D) จะทำให้ LR 1400 ON จากนั้นจะส่งผลไปยัง PC(B) โดยผ่านทาง LR 1400 จะทำให้ LR 5000 ของ PC(B) ON เมื่อ LR 5000 ON ส่งผลไปยัง PC (A) โดยจะทำให้ LR 0100 ON เมื่อ LR 0100 ON จะส่งผลไปยัง PC (C) โดยจะทำให้มีการMOV ค่าไปเก็บไว้ใน DM 0100 ไปสู่ LR 14 และในขณะเดียวกัน จะทำให้ LR 1500 ของ PC (C) ON ด้วย ในขณะนั้น LR 1500 จะสั่งให้ PC (A) MOV ค่าจาก LR 14 ไปเก็บไว้ที่ LR 15 และในขณะเดียวกันนั้น LR 3600 ของ PC (A) ON ด้วย และจะส่งผลไปยัง PC (B) โดยจะ MOV ค่าจาก LR35 ไปเก็บไว้ที่ LR 01 และในขณะนั้น LR 0015 ON ด้วย ะส่งผลไปยัง PC (D) โดยจะ MOV ค่าจาก LR 01 ไปเก็บไว้ที่ DM 0000

จะเห็นได้ว่ามีการ START ครั้งเดียวจะส่งผลให้ PC ทั้ง 4 ตัว มีการทำงานอย่างต่อเนื่องโดยติดต่อกันทาง LR AREA จะเห็นได้ว่า จากโปรแกรมเป็นการนำค่าจาก DM 0000 ของ PC (C) มาเก็บไว้ที่ DM 0000 ของ PC (D)

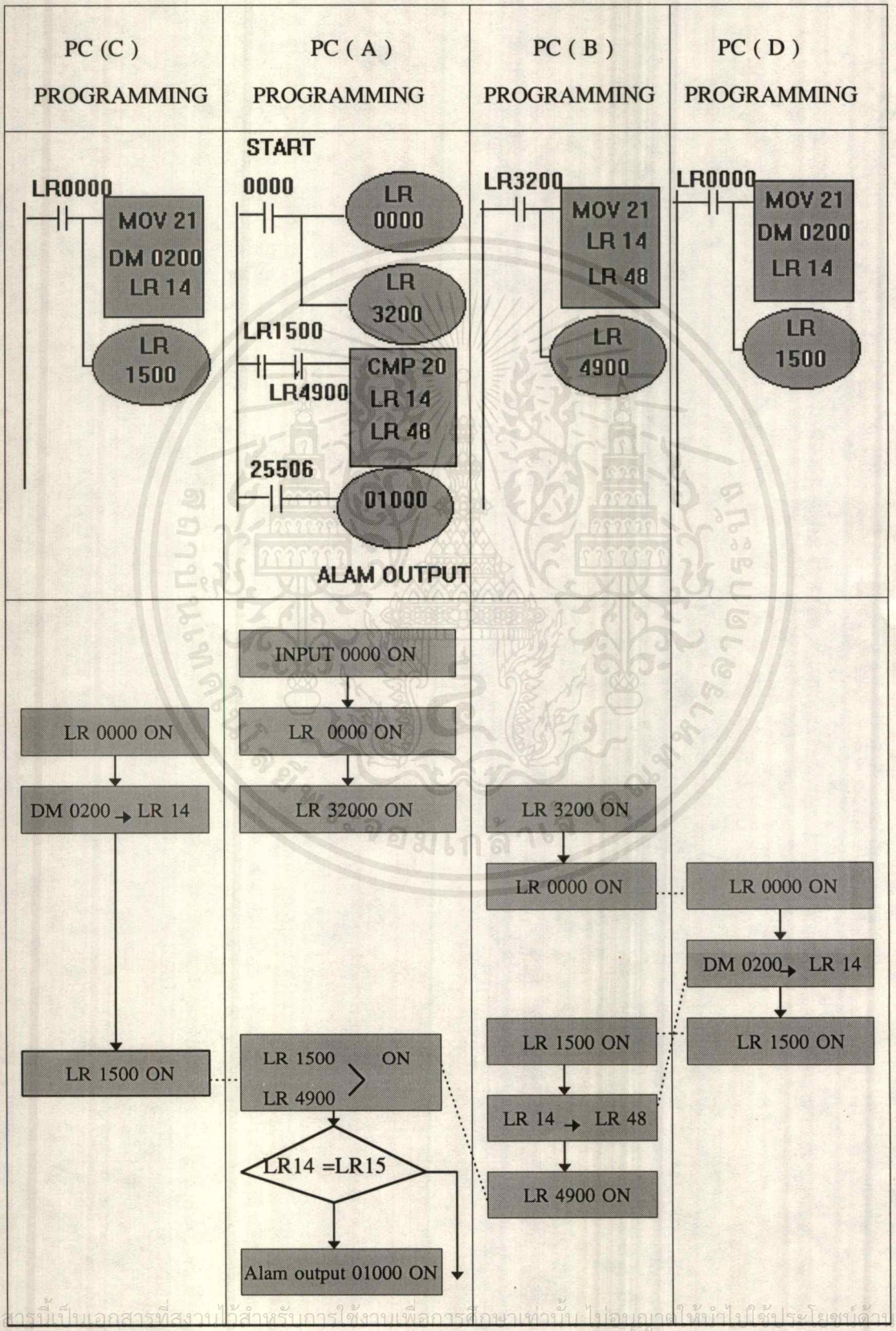
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMING EXAMPLE 2.

เป็นการ START การทำงานที่ PC (A) แล้วโปรแกรมจะทำการเปรียบเทียบค่าใน DM 0200 ใน PC (C) กับ PC (D) แล้วทำการแสดงผลที่ PC (A)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

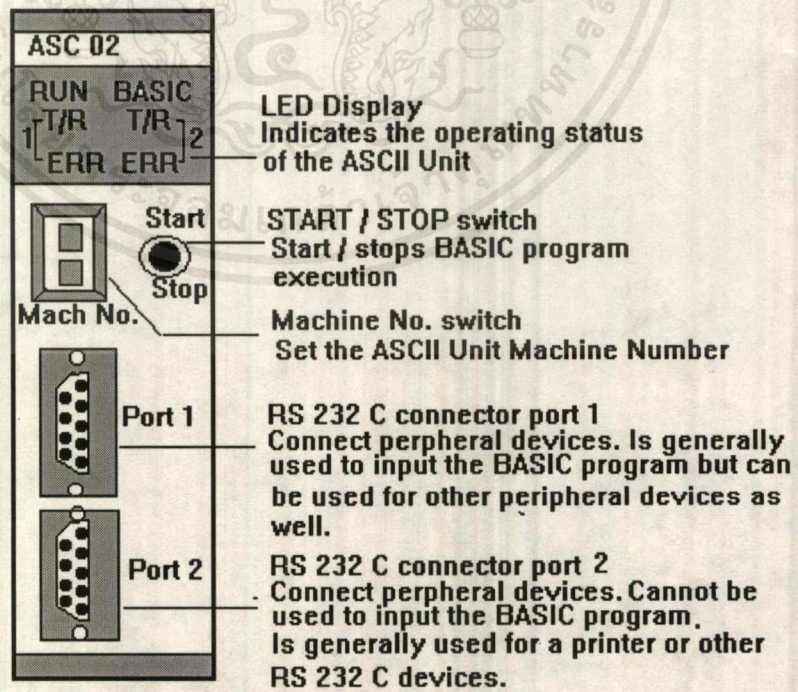
หน่วย แอสกี้ (ASCII UNIT)

4.1 หน่วย แอสกี้ (ASCII UNIT)

คือ อุปกรณ์ที่นำไปใช้ในการ INTERFACE ระหว่าง PC กับ COMPUTER และให้มีการรับส่งข้อมูลระหว่าง PC กับ COMPUTER โดยในการติดต่อ จะทำการเขียนโปรแกรมภาษา BASIC ซึ่งสามารถที่จะแสดงผลการทำงานของPROCESS ที่ PC CONTROL อยู่ทาง MONITER ของCOMPUTER ได้ ซึ่งจะแบ่งเบาภาระในการบำรุงรักษา PROCESS ซึ่งในโปรแกรมที่ยุ่งยาก ASCII UNIT จะช่วย PC ในการตัดสินใจสั่งงานเพื่อควบคุม PROCESS เพราะ ASCII UNIT ใช้โปรแกรมภาษา BASIC แทนที่จะเป็น LADDER และจะเหมาะสมกว่าสำหรับการประมวลผลข้อมูลซึ่ง ASCII UNIT มีความสะดวกในการควบคุมระบบได้มากกว่า

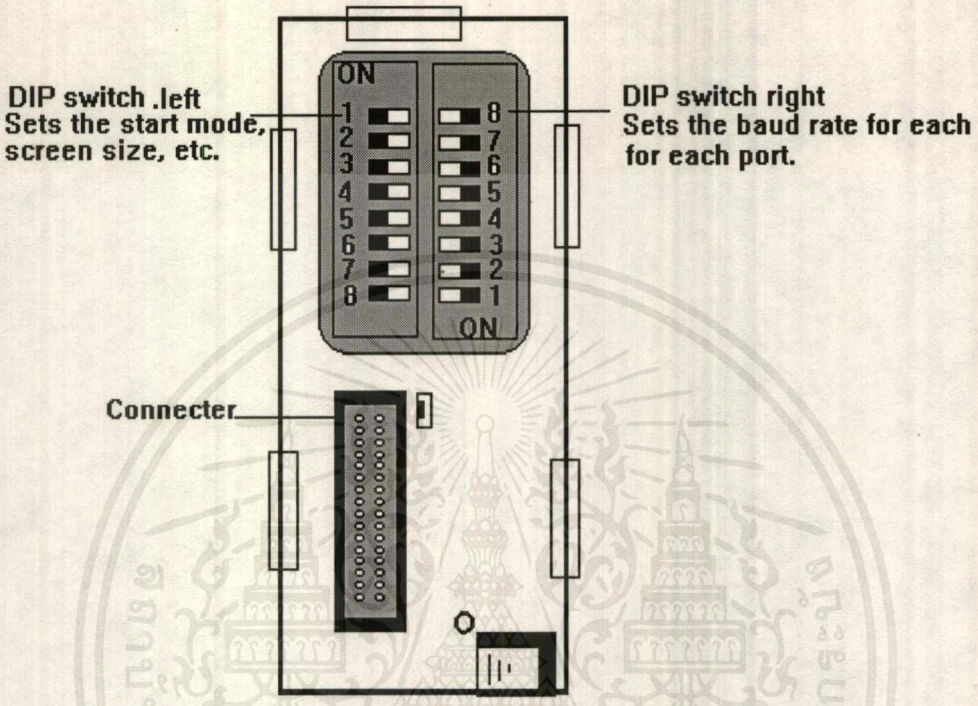
ลักษณะโครงสร้างทั่วไปของ ASCII UNIT

ด้านหน้า (FRONT PANEL)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านหลัง (BACK PANEL)

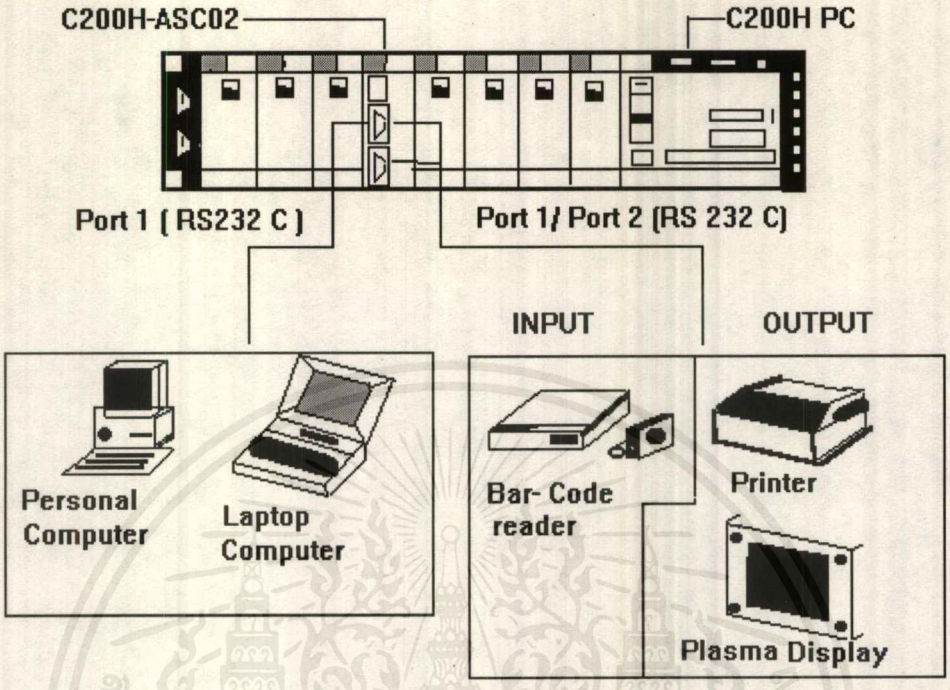


รูปที่ 4.1 แสดงลักษณะของ ASCII UNIT

4.2 รูปลักษณ์ของระบบ (SYSTEM CONFIGURATION)

ASCII UNIT สามารถที่จะเสียบเข้ากับ CPU RACK ก่อนที่จะติดตั้ง ASCII UNIT จะต้อง SET DIP SWITCH ตามเงื่อนไขที่ต้องการเสียก่อน และตอนที่นำไปติดตั้งต้องแน่ใจว่าแหล่งจ่ายอยู่ในสภาวะ OFF และในการติดต่อกับ COMPUTER สามารถที่จะติดต่อทาง PORT 1 ส่วน PORT 2 เราสามารถที่จะติดต่อเข้ากับอุปกรณ์ภายนอกอื่นๆได้ เช่น PRINTER, BAR- CODE READER, PLASMA DISPLAY ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงการต่อเชื่อม ASCII UNIT

4.3 การแบ่งข้อมูล (DATA SECTION)

4.3.1 บิต และ ไบต์ (BITS AND BYTES)

หน่วยความจำของ PC แบ่งออกเป็นหลายส่วน ซึ่งแต่ละส่วนจะมีชื่อและจุดมุ่งหมายของตัวมันเอง ซึ่ง ASCII UNIT สามารถที่จะเข้าไปในหน่วยความจำโดยการกำหนด BASIC READ และ BASIC WRITE อย่างไรก็ตามจะมีพื้นที่พิเศษของ PC คือ IR DATA AREA เป็นตัวกำหนด ASCII UNIT แต่ละส่วน MACHINE No SWITCH บนส่วนของ ASCII UNIT จะเลือกใช้ตั้งแต่ตำแหน่ง 1-9 หน่วยความจำของ PC ที่รวมเป็น UNIT จะเรียกว่า WORD รายละเอียดที่ถูกใช้จะเก็บไว้ใน WORD หรือ MULTIPLE WORD UNITS ซึ่งแต่ละ WORD จะมี ADDRESS อยู่ที่หน่วยความจำของ COMPUTER และสามารถที่จะเข้าถึงโดยระบบ ADDRESS ของมัน ซึ่งแต่ละ WORD จะประกอบด้วย 16 BIT จะเป็นส่วนเล็กๆ และจะเข้าถึงโดย COMPUTER BIT จะมีค่า 0 หรือ 1 เสมอ BIT สามารถที่จะ SET หรือ CREAR โดยโปรแกรมเมอร์ที่จะติดต่อกำหนดค่าตัวแปรหรือวางเงื่อนไขที่ CPU

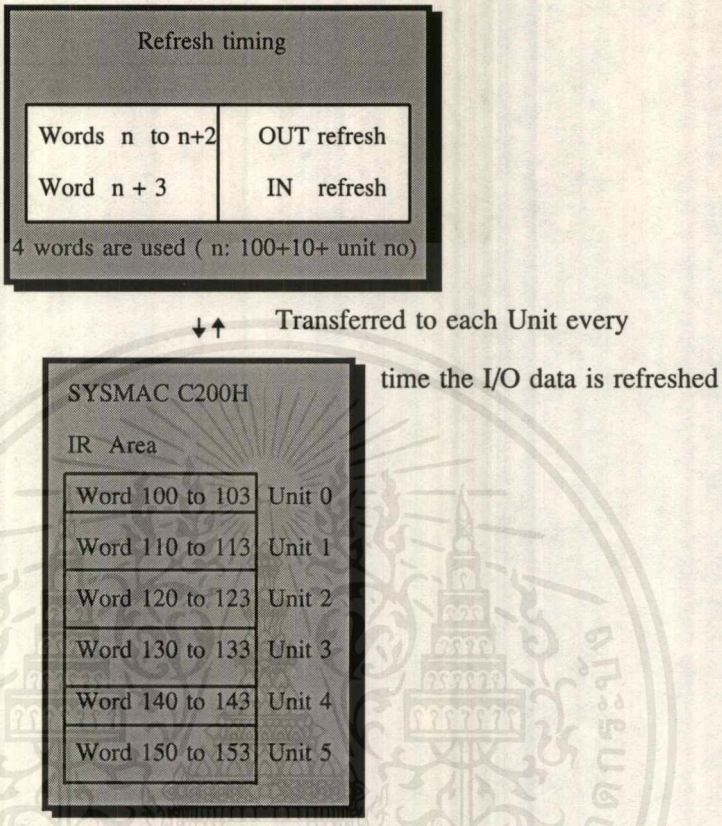
4.3.2 การแบ่งข้อมูล (DATA SECTION)

แต่ละ ASCII UNIT จะแบ่งหน่วยความจำออกเป็น 4 WORD ซึ่งเราเรียกว่า DATA SECTION สำหรับการสื่อสารของ PC WORD จะถูกแบ่งจาก ADDRESS 100 - 199 ของพื้นที่ PC IR MEMORY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงรายละเอียดที่ตั้ง BREAKDOWN และจุดมุ่งหมายของแต่ละ BIT ของส่วน DATA

ASCII UNIT



4.4การโปรแกรมและการติดต่อ

(PROGRAMMING AND COMMUNICATION)

4.4.1 โปรแกรม (PROGRAM)

ในการใช้ ASCII UNIT ร่วมกับ PC โดยโปรแกรมที่เขียนบน ASCII UNIT จะใช้ภาษา BASIC การแลกเปลี่ยนข้อมูลที่จะรวมกันใน PC PROGRAM ยกเว้นเมื่อ PC จะทำการอ่านและเขียน โดยกำหนดพื้นที่ที่กำหนดไว้ใน SPECIFIC MEMORY AREA โดยในการส่งข้อมูลจะต้องทำการกำหนดหมายเลขของคำที่จะทำการส่ง(NUMBER OF WORDS) และ BASE ADDRESS และ SPECIFIC MEMORY AREA เหล่านี้สามารถใช้ได้จากโครงสร้างของ PC MOVE

การสื่อสารระหว่าง ASCII UNIT กับ PC มี 2 วิธี

วิธีที่ 1

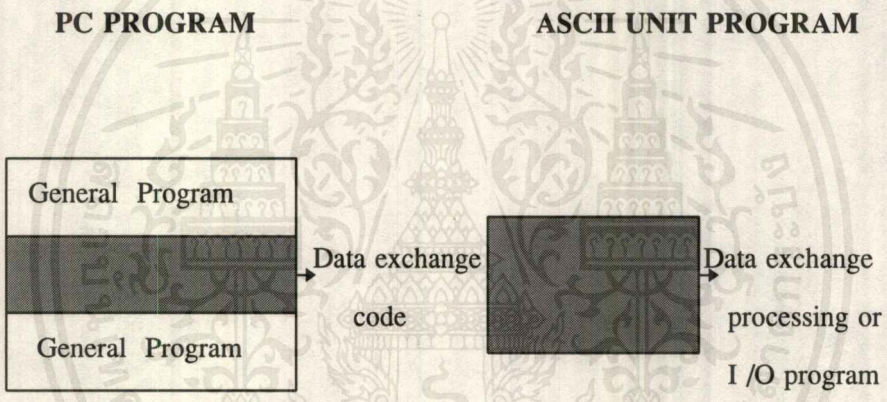
PC จะเป็นตัวควบคุมเวลาในการส่งผ่านข้อมูลระหว่างอุปกรณ์ทั้งสอง ASCII UNIT สามารถที่จะเข้าถึงข้อมูลใน PC DATA MEMORY ได้โดยใช้คำสั่ง PC READ , PC WRITE , PC GET ,PC PUT หลังจากนั้น PC จะส่งผลการตอบสนองจากการ RAED และ WRITE ไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

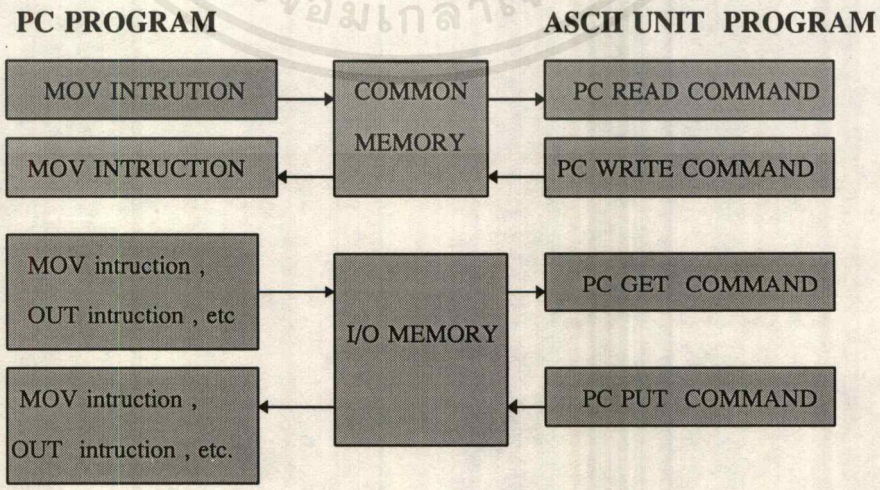
ASCII UNIT การแลกเปลี่ยนข้อมูลของ PC จะดำเนินการปฏิบัติโดยการระบุเจาะจง และเมื่อ PC ทำการอ่าน FLAG จะ SET และ ASCII UNIT จะดำเนินการส่งข้อมูลต่อไป

วิธีที่ 2

เมื่อไม่มีข้อมูลส่งผ่านพิเศษ (SPECIAL PC DATA XECHANGE CODE) จะมีความสะดวกในการสื่อสารข้อมูลระหว่างอุปกรณ์ทั้งสอง ถ้าหากพื้นที่ของ MEMORY ถูกออกแบบเป็นพิเศษสำหรับการอ่าน หรือการเขียน ASCII UNIT ก็สามารถติดต่อได้โดยตรงกับพื้นที่ ที่ทำการ MEMORY ไว้ ดังรูปที่ 4.4 อธิบายถึงความสัมพันธ์ของ PROGRAM ของ PC PROGRAM กับ ASCII UNIT PROGRAM และรูปที่ 4.5 แสดงถึงการแลกเปลี่ยนข้อมูลระหว่าง PC กับ ASCII UNIT



รูปที่ 4.4



รูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การส่งผ่านโปรแกรม (PROGRAM TRANSFER)

การเตรียมการ (PREPARATION): สำหรับ PC ที่ติดต่อกับ ASCII UNIT จะต้อง SET COMPUTER COMMUNICATION SOLFWARE ดังนี้

BAUD RATE : เหมือนกับ ASCII UNIT

DATA LENGTH : 8 BITS , PARITY : NONE

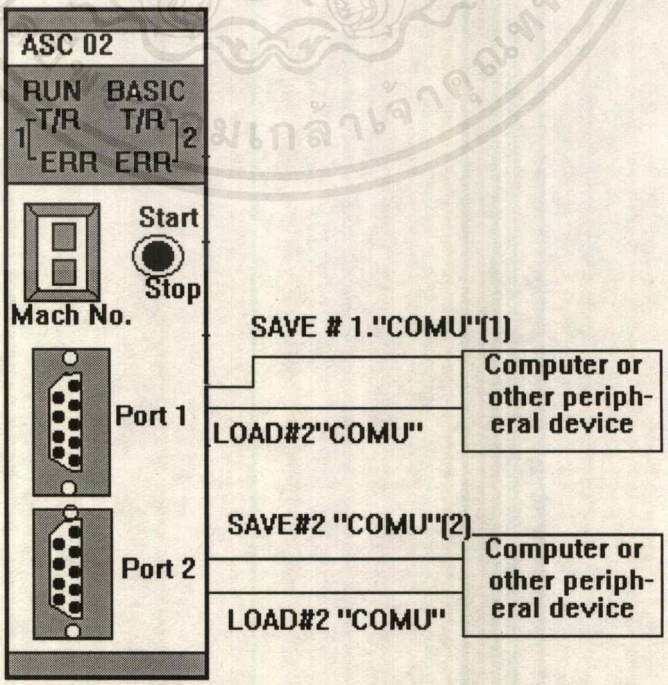
No. STOP BITS : 2.

ด้วยวิธี : FULL DUPLEX , NO ECHO , no XON/ XOFF BUFFER BURY CONTROL , NO AUTO LINE FEED

โดยการ : ตั้ง DIP SWITCH ของ ASCII UNIT ตามที่ต้องการใช้

การส่ง (TRANSFER) : ASCII UNIT BASIC PROGRAM จะเขียนบน PERSONAL COMPUTER ซึ่งจะต่อกับ ASCII UNIT ทาง PORT 1. ผ่านทาง RS-232C ในการ INTERFACE โปรแกรมจะถูกส่งจาก PERSONAL COMPUTER ไปยัง ASCII UNIT หรืออุปกรณ์เก็บข้อมูลอื่นที่มี PORT การติดต่อสื่อสารเดียวกัน โดยใช้คำสั่ง LOAD ในภาษาเบสิก และโปรแกรมจะถูกส่งจาก ASCII UNIT EEPROM ไปยัง ASCII UNIT RAM โดยคำสั่ง LOAD และโปรแกรมจะถูกส่งจาก ASCII UNIT RAM ไปยัง EEPROM หรือไปยัง PERSONAL COMPUTER หรือแหล่งข้อมูลอื่นโดยใช้คำสั่ง BASIC SAVE

ในการส่งผ่านข้อมูล (DATA TRANSFER) จะเกิด OVER FLOW ถ้าหากความจุ BUFFER ของ BAUD RATE ระหว่างCOMPUTER กับ ASCII UNIT ไม่ตรงกัน ถ้าหากเกิดปัญหาดังกล่าว จะต้องทำการ SET BAUD ที่กำลังโชว์ หรือ SPECIFIC XON ด้วยคำสั่ง OPEN

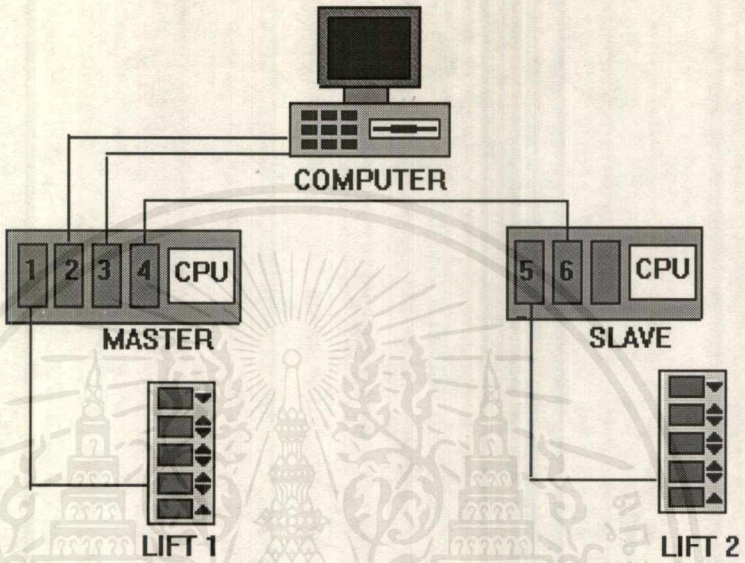


รูปที่4.6 DIRECTION OF DATA TRANSFER

เอกสารนี้เป็นเอกสารที่สงวนไว้ [illegible] ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5.

รูปลักษณ์ของโครงงาน



หมายเหตุ

หมายเลข 1, 5 คือ หน่วนอินพุท-เอาท์พุท (INPUT-OUTPUT UNIT)

หมายเลข 2 คือ HOST LINK UNIT

หมายเลข 3 คือ ASCII UNIT

หมายเลข 4, 6 คือ PC LINK UNIT

ระบบการทำงาน

แนวทางในการคิดโปรแกรมการทำงานของระบบลิฟท์

ในตอนแรกนั้นเราให้ลิฟท์อยู่ชั้นที่ 1 รอการเรียกทำงาน เมื่อมีการกดปุ่มเรียกการทำงาน ลิฟท์จะเก็บสภาวะการเรียกไว้ก่อน หลังจากนั้นก็จะนำไปเปรียบเทียบตำแหน่งของลิฟท์ กับจุดที่เรียก และเปรียบเทียบตำแหน่งของลิฟท์ทั้งสองตัว เมื่อลิฟท์มีการตัดสินใจว่าจะสั่งให้ตัวใดทำงานนั้น โปรแกรมจะทำการตัดสภาวะของอีกตัวออกเพื่อที่จะให้ตัวที่เหลือรอทำงานในสภาวะต่อไป จากนั้นลิฟท์มีการเปรียบเทียบว่าปุ่มกดนั้นอยู่ชั้นไหน อยู่ในชั้นที่สูงกว่าหรือต่ำกว่า ถ้าปุ่มกดเรียกลิฟท์นั้นอยู่ในชั้นที่สูงกว่า การทำงานก็ต้องสั่งมอเตอร์ควบคุมลิฟท์ หมุนไป ในทางที่ทำให้ลิฟท์สูงขึ้นไปเรื่อยๆ จนกว่าตำแหน่งของตัวลิฟท์นั้นอยู่ในชั้นเดียวกับจุดที่มีการเรียก จะทำให้มอเตอร์หยุดหมุน แล้วให้ประตูลิฟท์จะทำการเปิดออกภายในระยะเวลาหนึ่ง แล้วทำการปิดประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และทำการทำงานตามที่คนในลิฟท์นั้นเรียก และเมื่อมีการเรียกลิฟท์หลายๆจุดนั้น ลิฟท์ทั้งสองตัวจะมีการแบ่งภาระการทำงานกัน โดยมีการเปรียบเทียบความเหมาะสม และความพร้อมของลิฟท์แต่ละตัว เช่นในกรณีที่มีการเรียกขึ้นในชั้นที่ 2 ในขณะนั้นลิฟท์ตัวที่ 1กำลังทำงานในสภาวะขึ้นอยู่ชั้นที่ 3 และลิฟท์ตัวที่ 2 กำลังอยู่ชั้นที่ 1 ในสภาวะนี้การทำงานของโปรแกรมจะสั่งให้ลิฟท์ตัวที่ 1 รับสภาวะการทำงานนั้นไป และ ในกรณีที่มีการเรียกลงก็เช่นเดียวกัน โปรแกรมจะมีการเปรียบเทียบก่อน ก่อนที่จะสั่งให้ลิฟท์ตัวใดทำงาน และการทำงานของโปรแกรมนั้นก็ต้องมีการเปรียบเทียบกับรีเล์ภายในของ PC เพื่อที่จะสั่งให้โปรแกรมทำงานตามเงื่อนไขของการทำงานจริงของลิฟท์ และในการติดต่อสื่อสารระหว่าง PC กับ PC นั้นจะต้องทำการแบ่งพื้นที่การทำงานของแต่ละตัว และทำการเขียนโปรแกรมควบคุม และใช้ในการเปรียบเทียบการทำงานของลิฟท์แต่ละตัว ดังนั้นจุดสำคัญของโปรแกรมการควบคุมการทำงานของระบบลิฟท์นั้นต้องมีการเปรียบเทียบความพร้อมในของการทำงานของแต่ละตัว และการเปรียบเทียบความถูกต้องระหว่างชั้นที่เรียกกับชั้นที่ลิฟท์อยู่

สำหรับโปรแกรมการแสดงผลการทำงานของลิฟท์แต่ละตัวทาง COMPUTER นั้น จะแสดงบอกชั้นที่ลิฟท์แต่ละตัวกำลังอยู่ในขณะนั้น และแสดงการเรียกลิฟท์ว่ามีการเรียกในชั้นใดบ้างและลักษณะใด และมีโปรแกรมแยกแสดงของแต่ละตัว ซึ่งจะแสดงถึงว่าลิฟท์ตัวนั้นอยู่ชั้นใดและมีการทำงานขึ้นหรือลง และภายในลิฟท์นั้นผู้ใช้มีการเรียกชั้นใดบ้าง ภาพจะแสดงเปลี่ยนไปตามลักษณะความเป็นจริงของการทำงานในสภาวะนั้น ซึ่งจะทำให้เราสะดวกในการที่จะต้องการทราบการทำงานของแต่ละตัว

ลักษณะของแบบจำลองลิฟท์ที่ใช้ควบคุม

1. เป็นลิฟท์จำนวน 5 ชั้น 2 ตัว
2. มี PUSH BUTTON SWITCH สำหรับกดเรียกลิฟท์ต้องการขึ้น-ลง ตามชั้นต่างๆตามต้องการ
3. มี PUSH BUTTON SWITCH สำหรับผู้โดยสารกดในกรณีที่จะไปชั้นที่เท่าใด
4. มี LIMIT SWITCH ติดอยู่ที่ตำแหน่งชั้นต่างๆ เพื่อที่จะให้ทราบว่าลิฟท์อยู่ที่ชั้นที่เทาใด
5. การขึ้น-ลง ของลิฟทีจะใช้มอเตอร์ไฟฟ้ากระแสตรง และอาศัยการกลับทางหมุน
6. การเปิดปิดประตูของลิฟท์ จะใช้มอเตอร์ไฟฟ้ากระแสตรง และอาศัยการกลับทางหมุน
7. มี PUSH BUTTON SWITCH สำหรับสั่งให้ประตู ปิด-เปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนด INPUT - OUTPUT ของ PC ที่ใช้ในการควบคุมระบบลิฟท์

ลิฟท์ตัวที่ 1 (CPU # 1)

INPUT CH 00

LIMIT SWITCH ชั้นที่ 1	BIT 01
LIMIT SWITCH ชั้นที่ 2	BIT 02
LIMIT SWITCH ชั้นที่ 3	BIT 03
LIMIT SWITCH ชั้นที่ 4	BIT 04
LIMIT SWITCH ชั้นที่ 5	BIT 05
LIMIT DOOR CLOSE	BIT 06
LIMIT DOOR OPEN	BIT 07
PUSH BUTTON เรียกขึ้นชั้นที่ 1	BIT 08
PUSH BUTTON เรียกขึ้นชั้นที่ 2	BIT 09
PUSH BUTTON เรียกขึ้นชั้นที่ 3	BIT 10
PUSH BUTTON เรียกขึ้นชั้นที่ 4	BIT 11
PUSH BUTTON เรียกลงชั้นที่ 2	BIT 12
PUSH BUTTON เรียกลงชั้นที่ 3	BIT 13
PUSH BUTTON เรียกลงชั้นที่ 4	BIT 14
PUSH BUTTON เรียกลงชั้นที่ 5	BIT 15

INPUT CH 01

PUSH BUTTON เลือกชั้นที่ 1	BIT 01
PUSH BUTTON เลือกชั้นที่ 2	BIT 02
PUSH BUTTON เลือกชั้นที่ 3	BIT 03
PUSH BUTTON เลือกชั้นที่ 4	BIT 04
PUSH BUTTON เลือกชั้นที่ 5	BIT 05
PUSH BUTTON DOOR CLOSE	BIT 06
PUSH BUTTON DOOR OPEN	BIT 07

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT CH 05

มอเตอร์ขับลิฟท์ขึ้น	BIT 00
มอเตอร์ขับลิฟท์ลง	BIT 01
มอเตอร์เปิดประตูลิฟท์	BIT 02
มอเตอร์ปิดประตูลิฟท์	BIT 03
LED แสดงการเรียกขึ้นชั้นที่ 1	BIT 04
LED แสดงการเรียกขึ้นชั้นที่ 2	BIT 05
LED แสดงการเรียกขึ้นชั้นที่ 3	BIT 06
LED แสดงการเรียกขึ้นชั้นที่ 4	BIT 07
LED แสดงการเรียกลงชั้นที่ 2	BIT 08
LED แสดงการเรียกลงชั้นที่ 3	BIT 09
LED แสดงการเรียกลงชั้นที่ 4	BIT 10
LED แสดงการเรียกลงชั้นที่ 5	BIT 11

OUTPUT CH 06

LED แสดงการเลือกชั้นที่ 1	BIT 00
LED แสดงการเลือกชั้นที่ 2	BIT 02
LED แสดงการเลือกชั้นที่ 3	BIT 03
LED แสดงการเลือกชั้นที่ 4	BIT 04
LED แสดงการเลือกชั้นที่ 5	BIT 05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิฟท์ตัวที่ 2 (CPU # 2)

INPUT CH 00

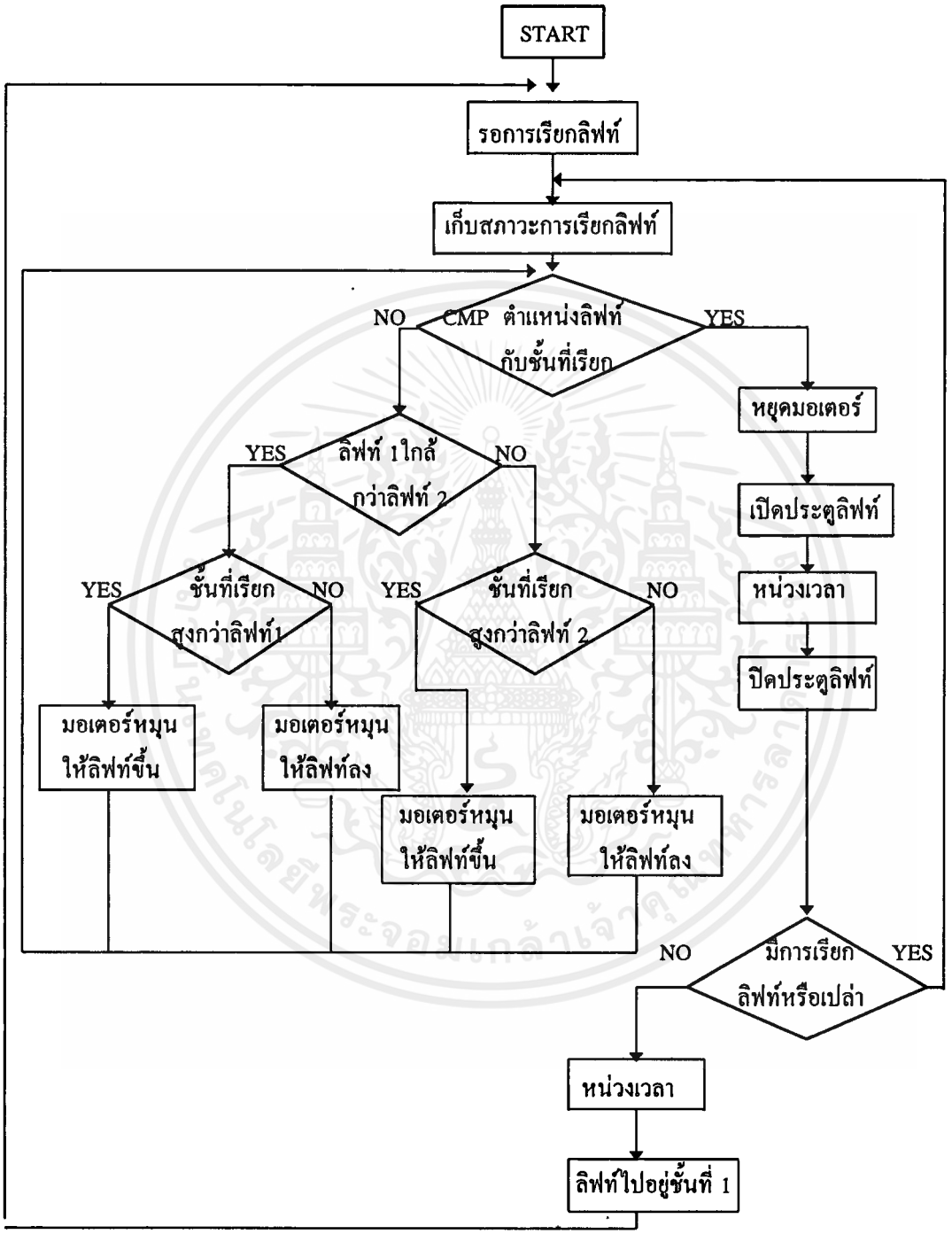
LIMIT SWITCH ชั้นที่ 1 BIT 01
LIMIT SWITCH ชั้นที่ 2 BIT 02
LIMIT SWITCH ชั้นที่ 3 BIT 03
LIMIT SWITCH ชั้นที่ 4 BIT 04
LIMIT SWITCH ชั้นที่ 5 BIT 05
LIMIT SWITCH DOOR CLOSE BIT 06
LIMIT SWITCH DOOR OPEN BIT 07
PUSH BUTTON เลือกชั้นที่ 1 BIT 08
PUSH BUTTON เลือกชั้นที่ 2 BIT 09
PUSH BUTTON เลือกชั้นที่ 3 BIT 10
PUSH BUTTON เลือกชั้นที 4 BIT 11
PUSH BUTTON เลือกชั้นที่ 5 BIT 12
PUSH BUTTON DOOR CLOSE BIT 13
PUSH BUTTON DOOR OPEN BIT 14

OUTPUT CH 05

มอเตอร์ขับลิฟท์ ขึ้น BIT 00
มอเตอร์ขับลิฟท์ ลง BIT 01
มอเตอร์ปิดประตูลิฟท์ BIT 02
มอเตอร์เปิดประตูลิฟท์ BIT 03
LED แสดงการเลือกชั้นที่ 1 BIT 07
LED แสดงการเลือกชั้นที่ 2 BIT 08
LED แสดงการเลือกชั้นที่ 3 BIT 09
LED แสดงการเลือกชั้นที่ 4 BIT 10
LED แสดงการเลือกชั้นที่ 5 BIT 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการทำงานของลิฟท์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลระหว่าง PC กับ PC

เราจะทำการแบ่งพื้นที่ของ PC แต่ละตัวโดยการเซ็ต DIP SWITCH ที่ PC LINK UNIT แล้วทำการเขียนโปรแกรม โดยใช้พื้นที่นั้นในการส่งผ่านข้อมูล ซึ่งในการส่งผ่านข้อมูลระหว่างกันจะใช้ PC LINK UNIT (LK- 401) ในการทำงาน ดังนั้น PCทั้งสองตัวจะสามารถที่จะรับรู้การทำงานของกันและกันได้ ซึ่งโปรแกรมการทำงานของ CPU แต่ละตัวจะทำการเก็บไว้ในแต่ละตัว โดยมีการรับรู้กันทางพื้นที่การติดต่อ (LR AREA) ซึ่งจะทำให้มีความสะดวกในการควบคุมระบบมากขึ้น

ดังนั้นการทำงานของโครงงานนั้น PC แต่ละตัวจะทำการแบ่งหน้าที่กันทำงานตลอดเวลาที่มีการสั่งให้ทำงาน โดย PC สองตัวจะไม่มีการทำงานที่ซ้ำกัน มันจะทำงานกันอย่างอิสระภายใต้การเปรียบเทียบถึงความพร้อมของแต่ละตัว ซึ่งในการเขียนโปรแกรมการติดต่อนี้เราต้องมีความรู้ในส่วนของพื้นที่ในการติดต่อของ PC แต่ละตัว เพราะพื้นที่นี้จะมีการแบ่งเป็นส่วนการเขียน และส่วนของการอ่าน ซึ่งการทำงานในลักษณะนี้เราสามารถที่นำไปพัฒนา ในการสั่งงานของระบบโดยผ่านอีกระบบได้ จะทำให้มีความสะดวกในการควบคุม

การสื่อสารระหว่าง PC กับ COMPUTER

การสื่อสารระหว่าง PC กับ COMPUTER นั้นเราใช้ ASCII UNIT ซึ่งจะพูดว่าเป็น COMPUTER ตัวหนึ่งก็ได้ ซึ่ง ASCII UNIT จะทำงานร่วมกับ PC โดยทำการเสียบลงใน RACK ของ PC รุ่น C 200 H ซึ่งการติดค่อกันนั้นเราจะแสดงภาพสภาวะการทำงานของระบบแต่ละตัว ดังนั้นในส่วนนี้จะมีหน้าที่การทำงานอยู่สองส่วน คือ ตัวส่งค่า และตัวรับค่า

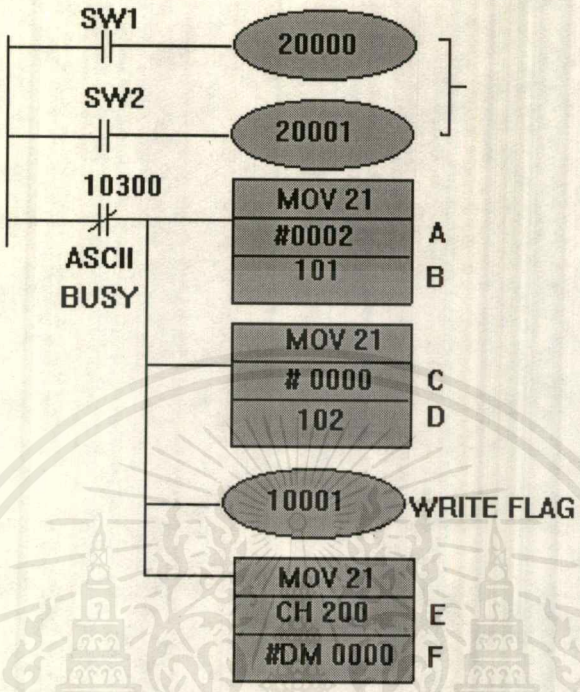
ตัวส่งค่า (ASCII UNIT)

การส่งค่าผ่าน ASCII UNIT นั้นเราจะทำการเขียนโปรแกรมสองส่วนด้วยกัน คือ

1. เขียนโปรแกรมที่ PC คือ จะทำการเขียนเป็น LADDER DIAGRAM ซึ่งลักษณะของโปรแกรมจะทำการส่งค่าสภาวะการทำงานของระบบ โดยผ่านทาง ASCII UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของโปรแกรม



เมื่อ

IR จะต้องตรงกับ CH ใน

A : NUMBER OF WORDS ที่จะส่ง

B : WORD n+1

C : TRANSFER BASE WORD (DM 0000)

D : WORD n + 2

(Where n = 100 + 10 * unit No)

E : CHANNEL ที่มีการเก็บ

F : DM ที่นำมาเก็บ

จาก LADDER DIAGRAM ในตัวอย่าง เป็นการส่งค่าสภาวะ INPUT 2 ตัว คือ SW1,SW2 ไปเก็บไว้ที่ DM 0000 โดยผ่านทาง BIT 00 และ 01 ของ CH 200 ซึ่งค่าของ DM 0000 จะมีการเปลี่ยนแปลงไปเรื่อยๆ ตามสภาวะการทำงานของ SW. เมื่อทำการเขียน LADDER DIAGRAM เสร็จแล้ว ก็ทำการเขียนโปรแกรมที่ ASCII UNIT

2. การเขียนโปรแกรมที่ ASCII UNIT จะใช้ภาษา BASIC เพราะที่ ASCII UNIT ได้บรรจุภาษา BASIC ไว้ และการเขียนโปรแกรมภาษา BASIC จะต้องมีความสัมพันธ์กับ RADDER-DIAGRAM และจาก LADDER DIAGRAMข้างบนสามารถที่จะเขียนภาษา BASIC ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
10  OPEN # 1, " COMU : ( 47 ) "
20  FOR  I = 1   TO  5000
30  NEXT  I
40  CLC #  1
50  PC READ " 2 I 4 " ; A,B
60  PRINT # 1, A, B
70  GOTO 50
```

จากนั้นจึงทำการเก็บค่าไว้ในหน่วยความจำของ ASCII UNIT ซึ่งจะทำให้ค่าที่ ASCII UNITมีการเปลี่ยนแปลงไปตามสภาวะของ INPUT

ตัวรับค่า และแสดงผล (COMPUTER)

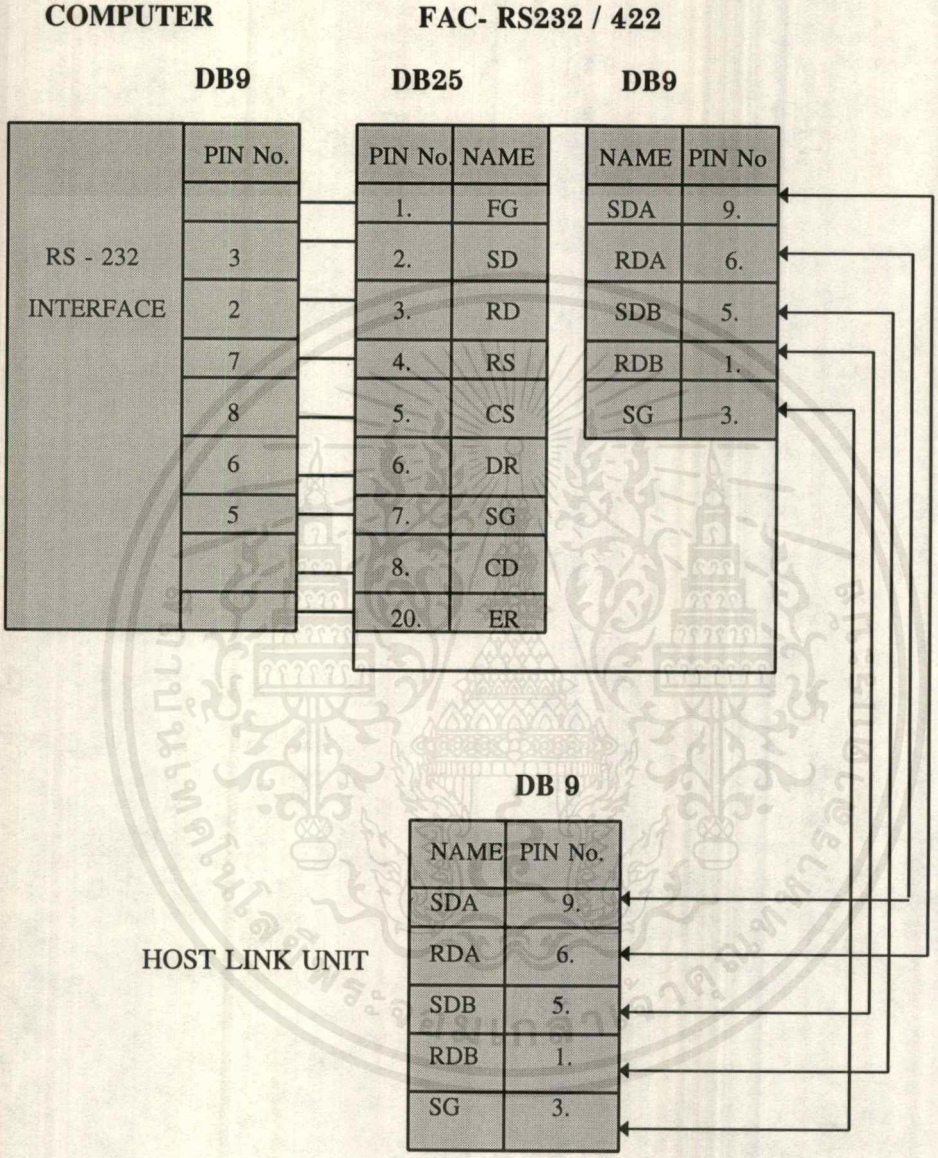
ASCII UNIT จะทำการส่งค่าสภาวะการทำงานมายัง COMPUTER เมื่อ COMPUTER ทำการรับสภาวะการทำงานของ INPUT แล้วจะทำการทำงานตามรูปแบบที่ทำการเขียนโปรแกรมแสดงผลไว้ ซึ่งโปรแกรมแสดงผลในที่นี้ใช้ ภาษา C ในการเขียน ซึ่งจะทำให้เราสามารถที่จะรู้การทำงานของระบบแต่ละตัวได้

การ LOAD ค่าโดยใช้โปรแกรม

การ LOAD ค่าโดยการใช้โปรแกรมนี้เราใช้ HOST LINK UNIT (LK- 202) ร่วมกับโปรแกรมสำเร็จรูปของ OMRON (LSS)โดยเราสามารถที่จะดึงโปรแกรมจาก PC ได้และในลักษณะเดียวกันเราสามารถที่จะป้อนโปรแกรมลงสู่ PC โดยผ่านทาง COMPUTER ได้ จึงสะดวกในกรณีที่เราทำการเปลี่ยน CPU ตัวใหม่โดยเราไม่ต้องป้อนโปรแกรมใหม่ และเราสามารถที่จะดึงเอา LADDER DIAGRAM มาเพื่อทำการ PRINT ได้ โดยเราสามารถที่ทำการเก็บโปรแกรมของ PC ไว้ใน COMPUTER

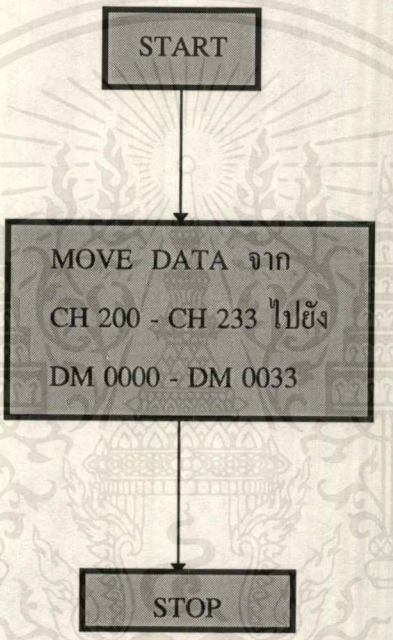
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการต่อสายระหว่าง HOST LINK UNIT กับ COMPUTER



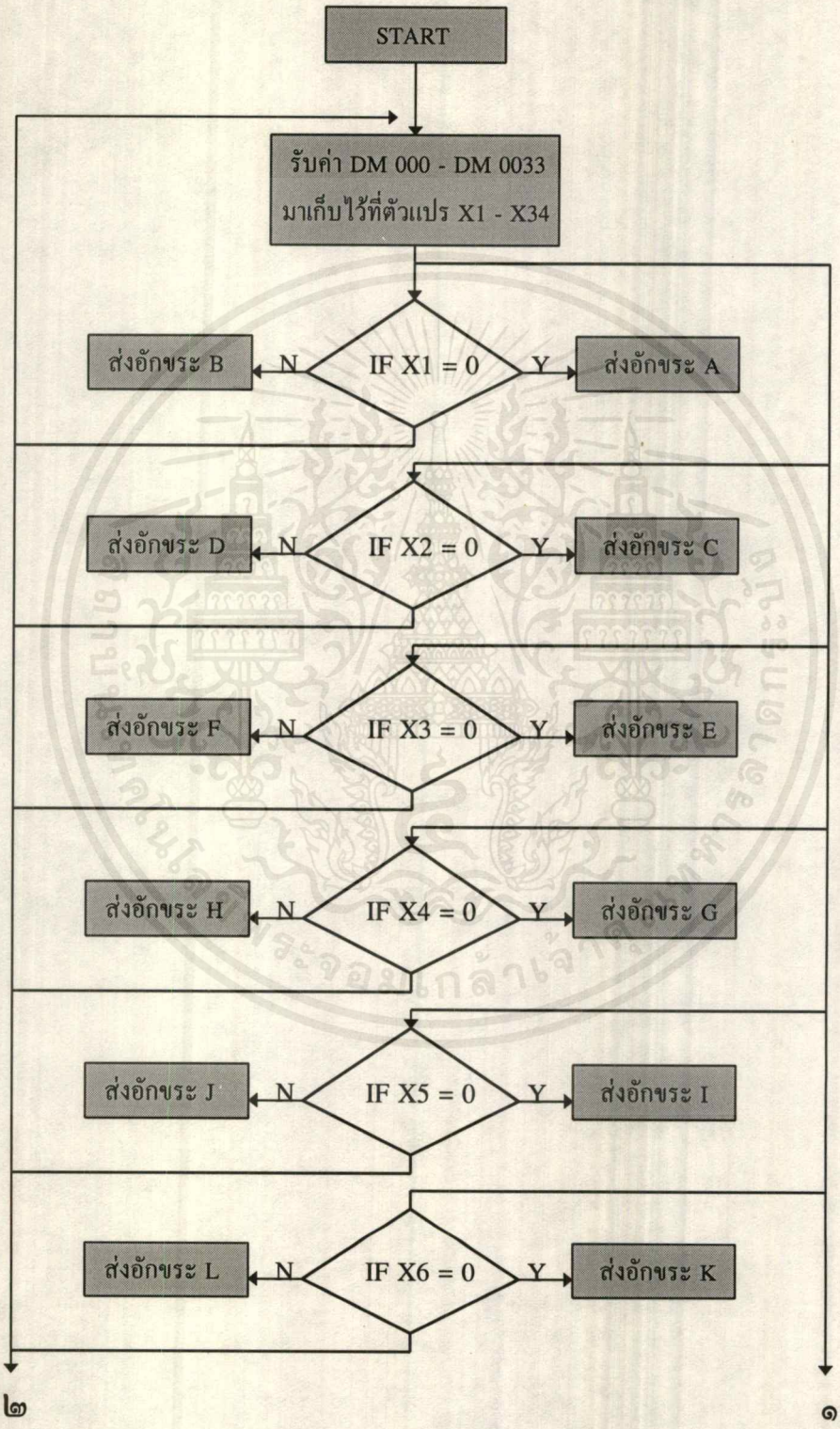
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการทำงานของการส่งค่าจาก PLC สู่ ASCII UNIT

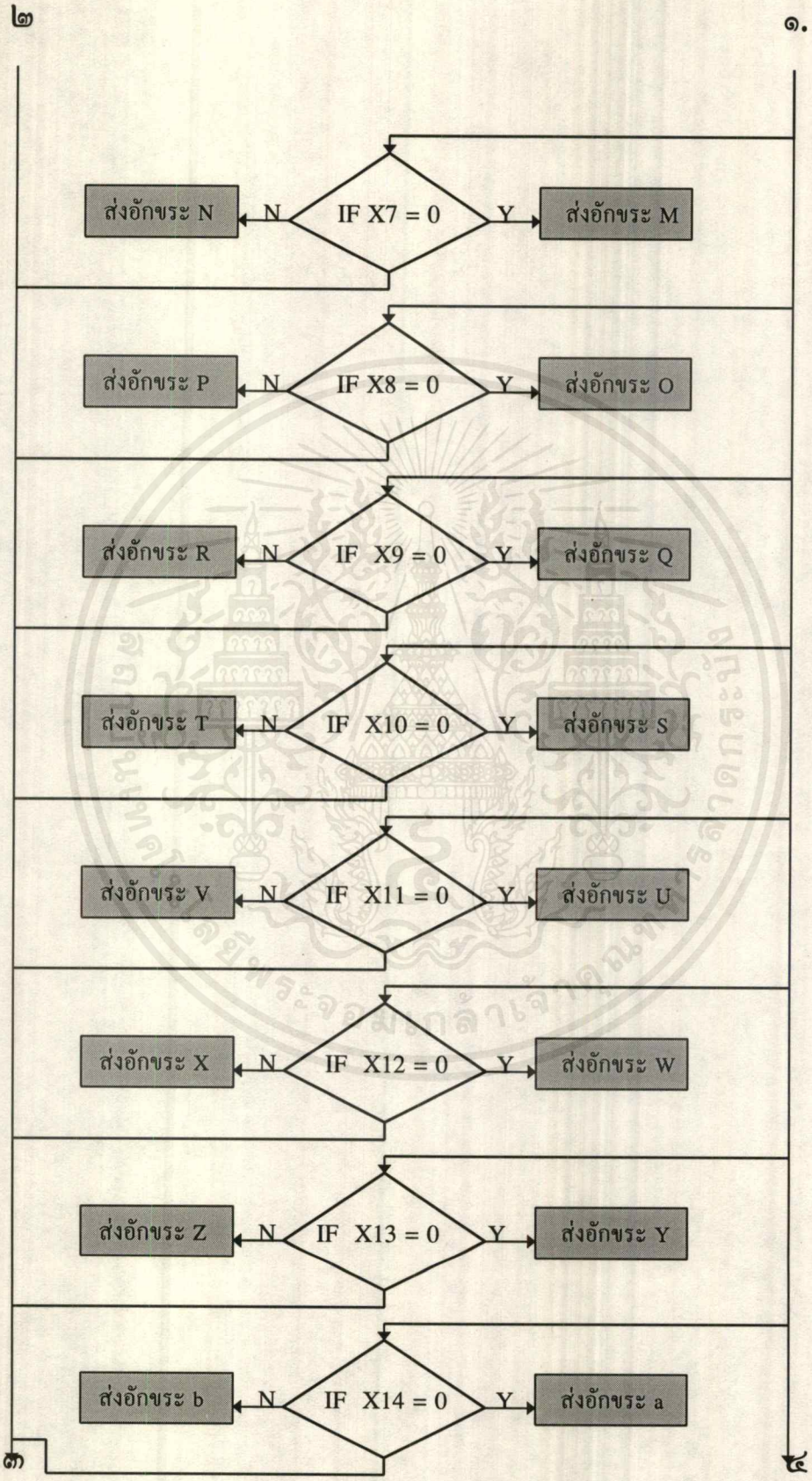


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

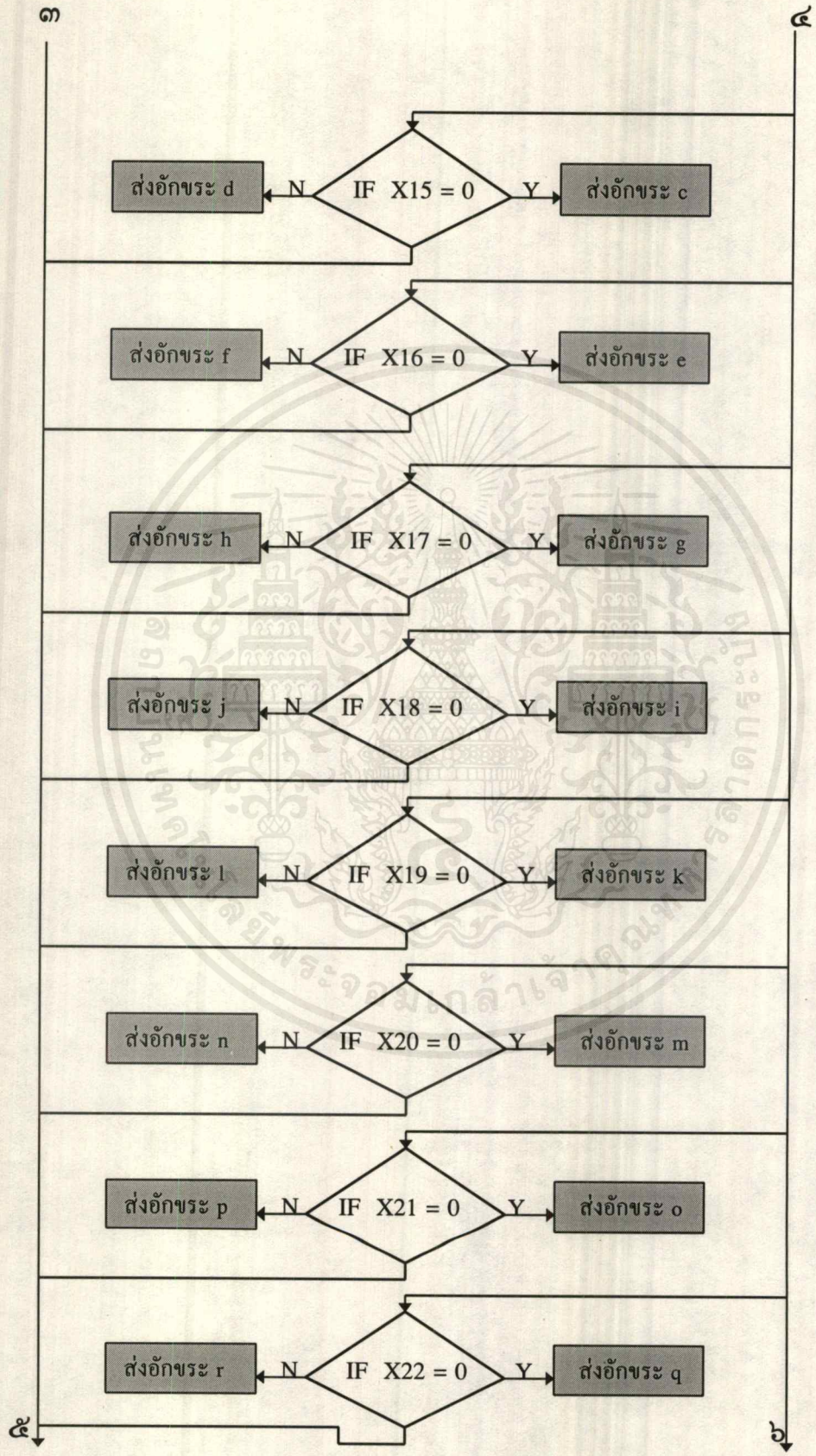
กระบวนการทำงานของการส่งค่าจาก ASCII UNIT สู่ COMPUTER



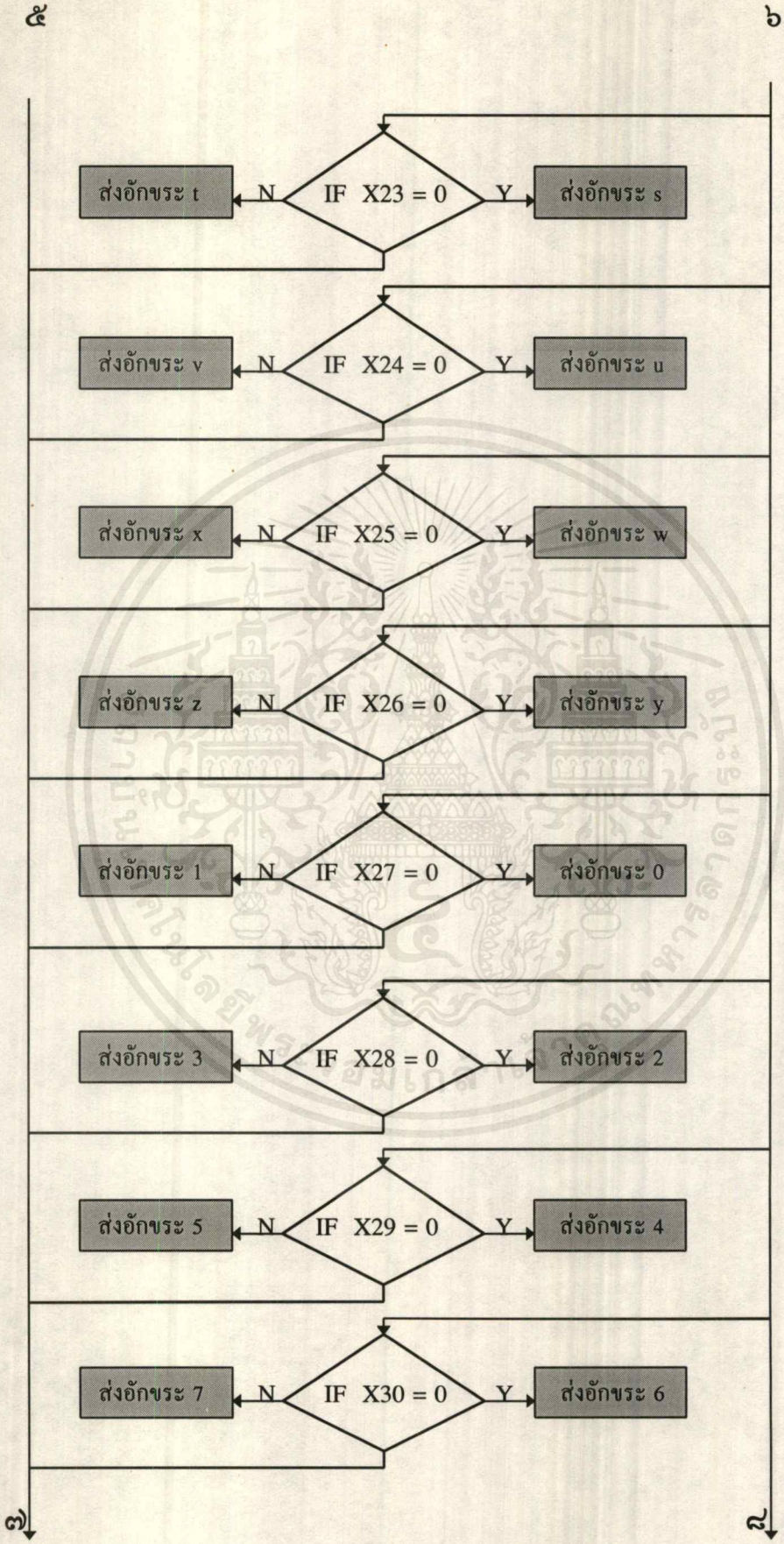
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



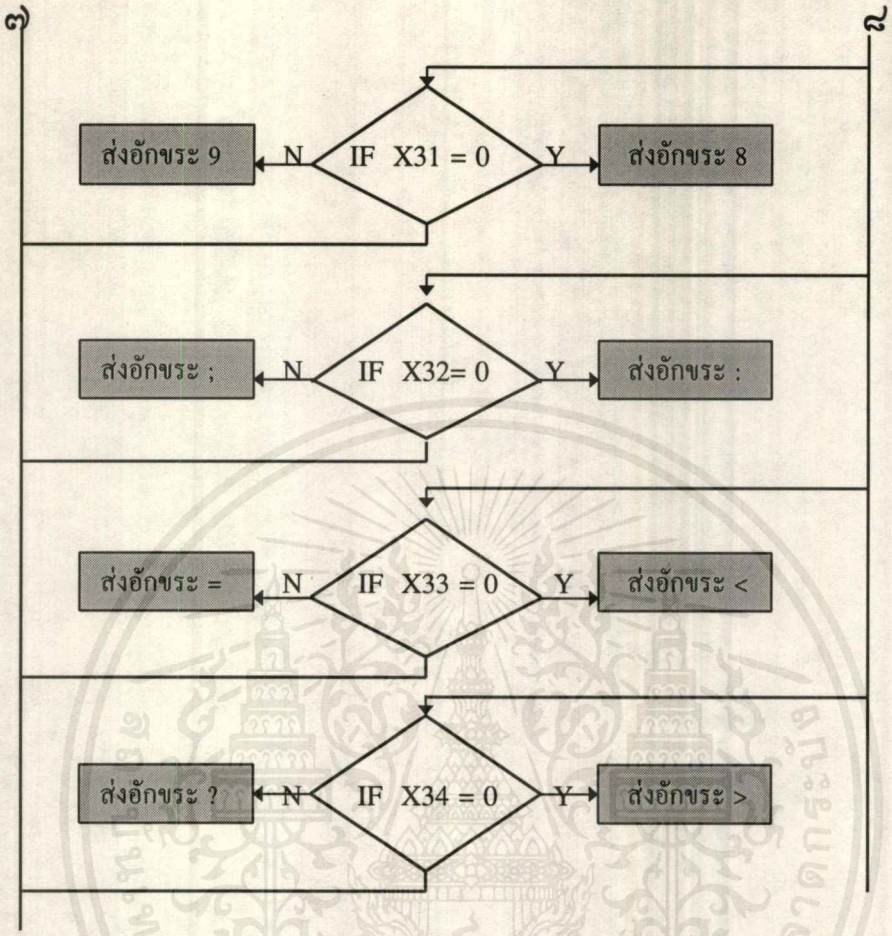
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาษา BASIC ที่เขียนลงใน ASCII UNIT เพื่อที่ใช้ในการส่งข้อมูลสู่ COMPUTER

```
10 OPEN #1,"COMU:(47)"
20 FOR I = 1 TO 5000
30 NEXT I
40 CLS #1
50 PC READ"@D,0,34,34H4";A,B,C,D,E,F,G,H,I,J,
   K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,A1,B1,C1,D1,E1,F1,G1,H1
60      IF A = 0    THEN PRINT #1, "A"
70      IF A = 1    THEN PRINT #1, "B"
80      IF B = 0    THEN PRINT #1, "C"
90      IF B = 1    THEN PRINT #1, "D"
100     IF C = 0    THEN PRINT #1, "E"
110     IF C = 1    THEN PRINT #1, "F"
120     IF D = 0    THEN PRINT #1, "G"
130     IF D = 1    THEN PRINT #1, "H"
140     IF E = 0    THEN PRINT #1, "I"
150     IF E = 1    THEN PRINT #1, "J"
160     IF F = 0    THEN PRINT #1, "K"
170     IF F = 1    THEN PRINT #1, "L"
180     IF G = 0    THEN PRINT #1, "M"
190     IF G = 1    THEN PRINT #1, "N"
200     IF H = 0    THEN PRINT #1, "O"
210     IF H = 1    THEN PRINT #1, "P"
220     IF I = 0    THEN PRINT #1, "Q"
230     IF I = 1    THEN PRINT #1, "R"
240     IF J = 0    THEN PRINT #1, "S"
250     IF J = 1    THEN PRINT #1, "T"
260     IF K = 0    THEN PRINT #1, "U"
270     IF K = 1    THEN PRINT #1, "V"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

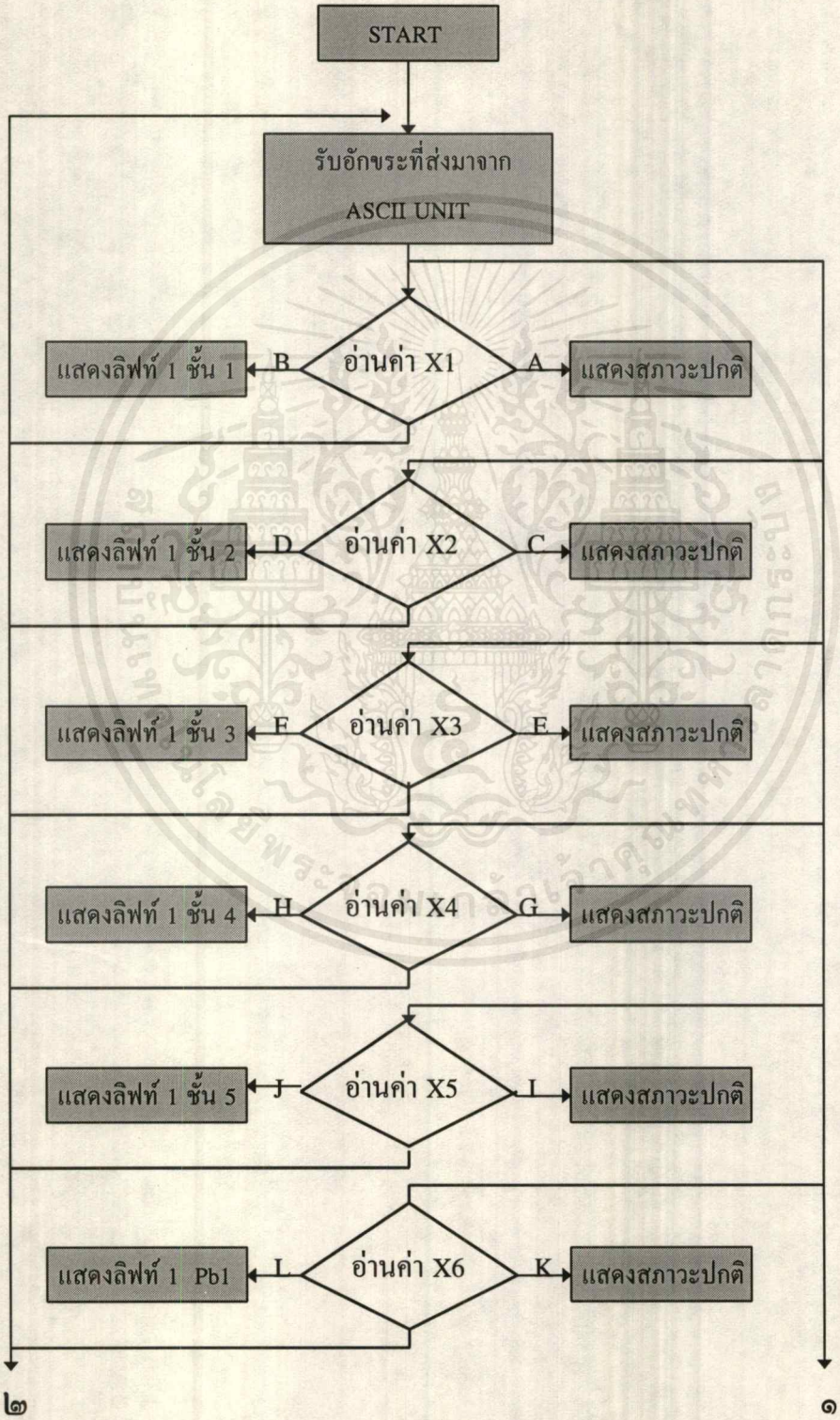
```
280     IF L = 0     THEN PRINT #1, "W"
290     IF L = 1     THEN PRINT #1, "X"
300     IF M = 0     THEN PRINT #1, "Y"
310     IF M = 1     THEN PRINT #1, "Z"
320     IF N = 0     THEN PRINT #1, "a"
330     IF N = 1     THEN PRINT #1, "b"
340     IF O = 0     THEN PRINT #1, "c"
350     IF O = 1     THEN PRINT #1, "d"
360     IF P = 0     THEN PRINT #1, "e"
370     IF P = 1     THEN PRINT #1, "f"
380     IF Q = 0     THEN PRINT #1, "g"
390     IF Q = 1     THEN PRINT #1, "h"
400     IF R = 0     THEN PRINT #1, "i"
410     IF R = 1     THEN PRINT #1, "j"
420     IF S = 0     THEN PRINT #1, "k"
430     IF S = 1     THEN PRINT #1, "l"
440     IF T = 0     THEN PRINT #1, "m"
450     IF T = 1     THEN PRINT #1, "n"
460     IF U = 0     THEN PRINT #1, "o"
470     IF U = 1     THEN PRINT #1, "p"
480     IF V = 0     THEN PRINT #1, "q"
490     IF V = 1     THEN PRINT #1, "r"
500     IF W = 0     THEN PRINT #1, "s"
510     IF W = 1     THEN PRINT #1, "t"
520     IF X = 0     THEN PRINT #1, "u"
530     IF X = 1     THEN PRINT #1, "v"
540     IF Y = 0     THEN PRINT #1, "w"
550     IF Y = 1     THEN PRINT #1, "x"
560     IF Z = 0     THEN PRINT #1, "y"
570     IF Z = 1     THEN PRINT #1, "z"
580     IF A1 = 0    THEN PRINT #1, "0"
590     IF A1 = 1    THEN PRINT #1, "1"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

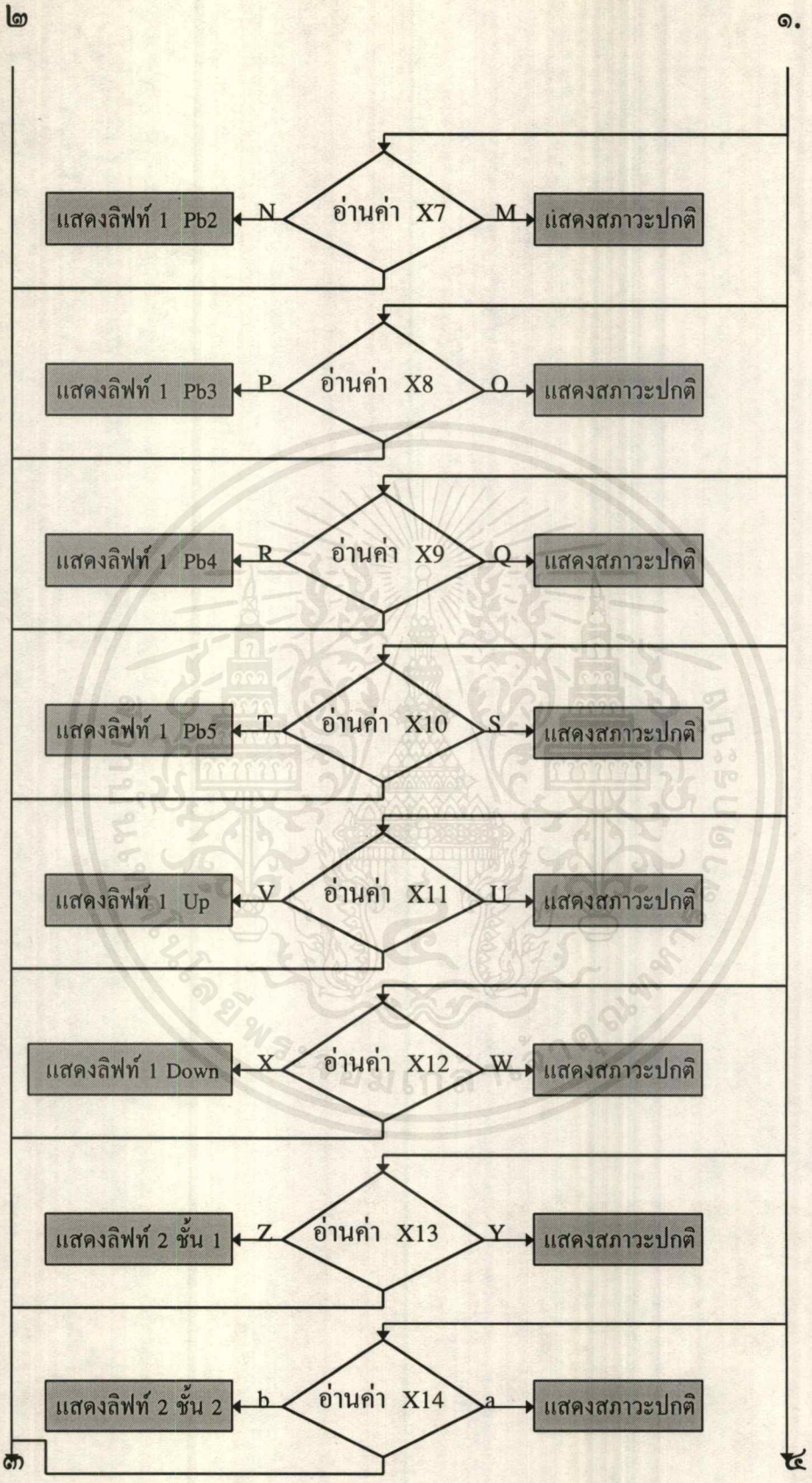
```
600     IF B1 = 0     THEN PRINT #1, "2"
610     IF B1 = 1     THEN PRINT #1, "3"
620     IF C1 = 0     THEN PRINT #1, "4"
630     IF C1 = 1     THEN PRINT #1, "5"
640     IF D1 = 0     THEN PRINT #1, "6"
650     IF D1 = 1     THEN PRINT #1, "7"
660     IF E1 = 0     THEN PRINT #1, "8"
670     IF E1 = 1     THEN PRINT #1, "9"
680     IF F1 = 0     THEN PRINT #1, ":"
690     IF F1 = 1     THEN PRINT #1, ";"
700     IF G1 = 0     THEN PRINT #1, "<"
710     IF G1 = 1     THEN PRINT #1, "="
720     IF H1 = 0     THEN PRINT #1, ">"
730     IF H1 = 1     THEN PRINT #1, "?"
740     GOTO 50
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

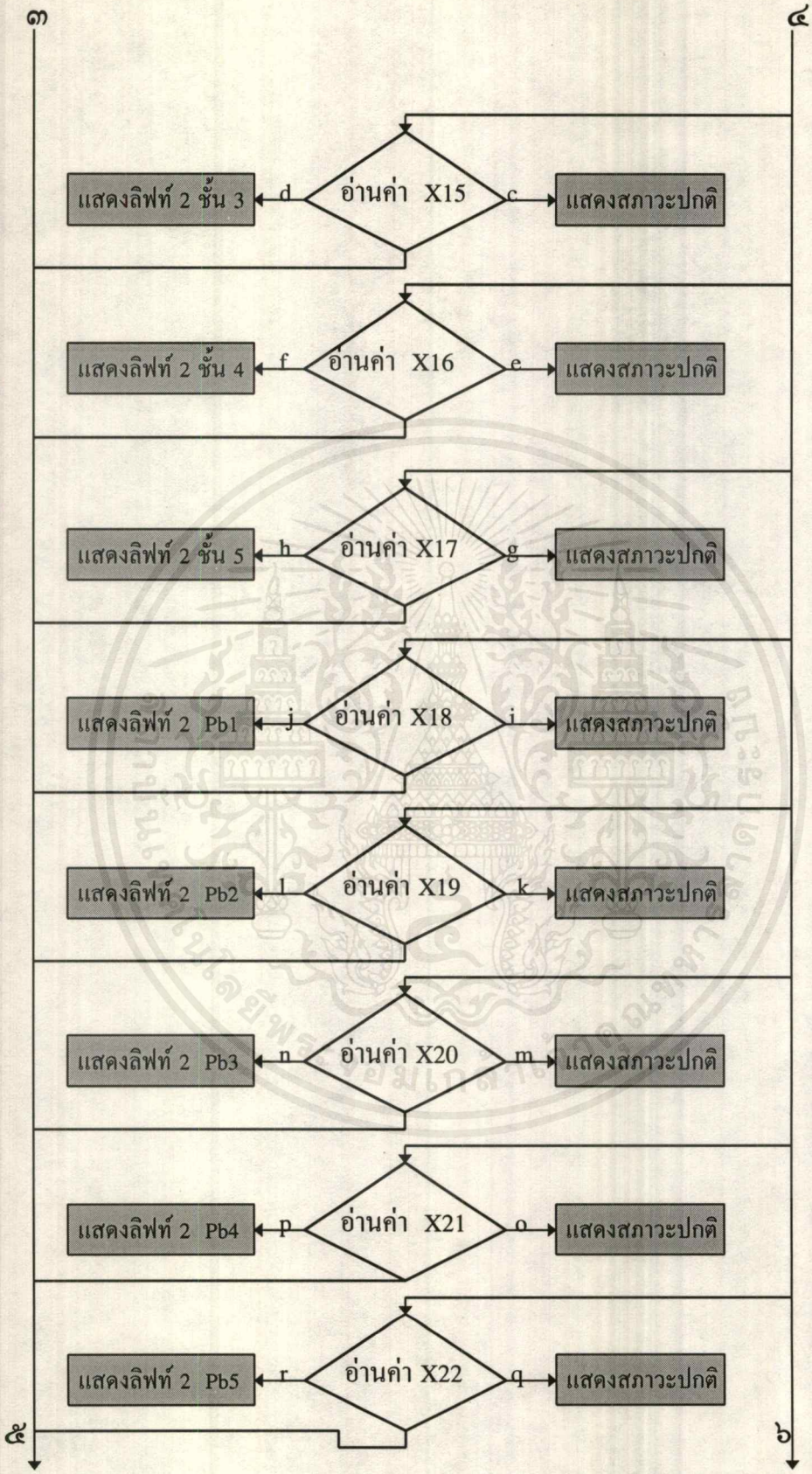
กระบวนการทำงานของการรับค่าจาก ASCII UNIT และแสดงผลการทำงานของกระบวนการที่ COMPUTER



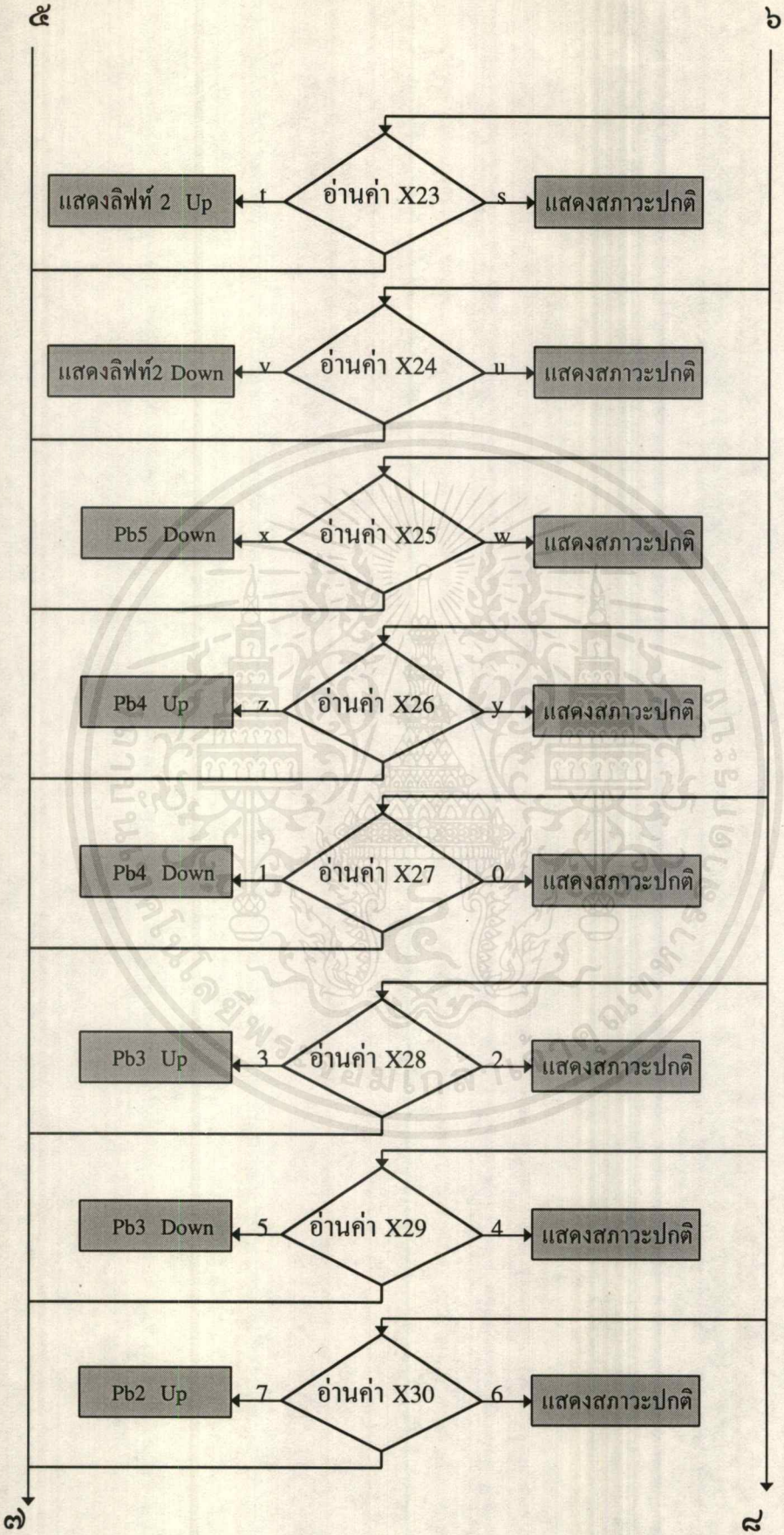
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



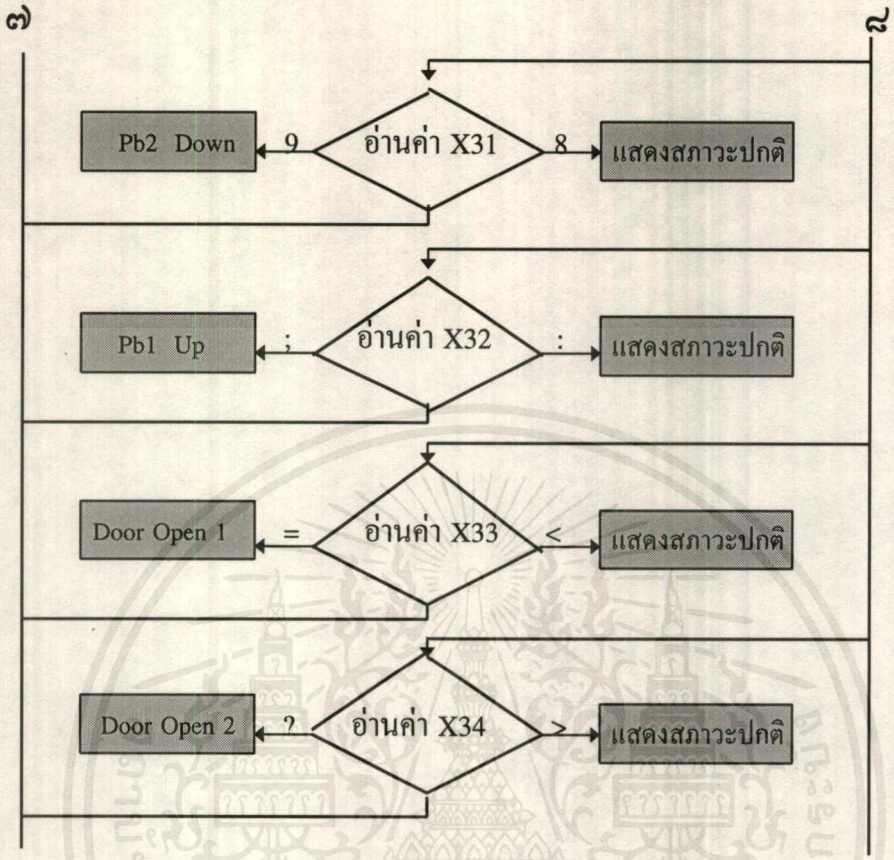
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



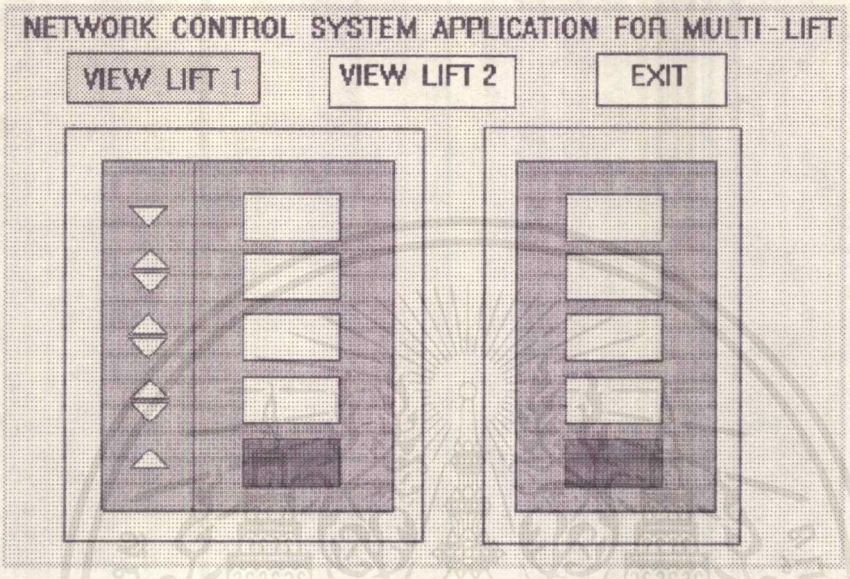
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



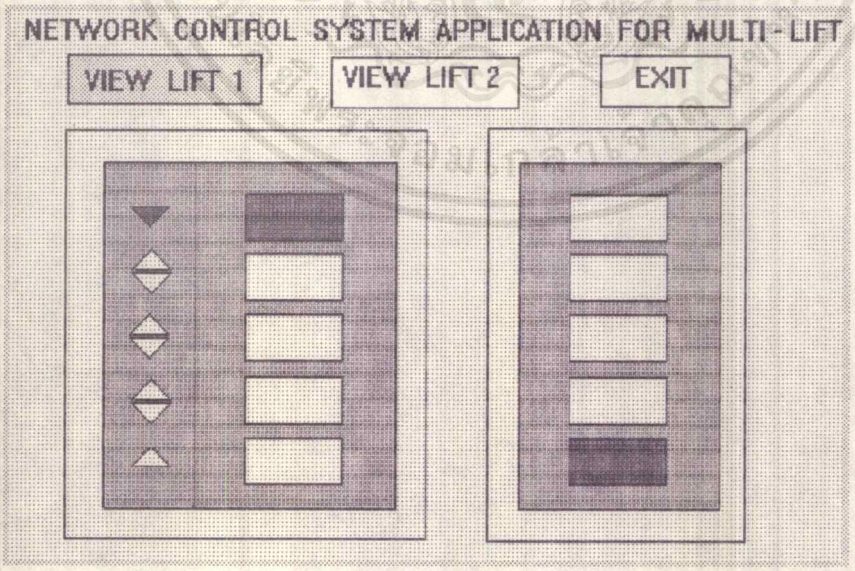
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างผลที่แสดงทางจอภาพของ COMPUTER

1. ภาพแสดงสภาวะเริ่มต้นเมื่อเข้าสู่โปรแกรม และในขณะนั้นลิฟท์ทั้งสองตัวอยู่ชั้นที่ 1

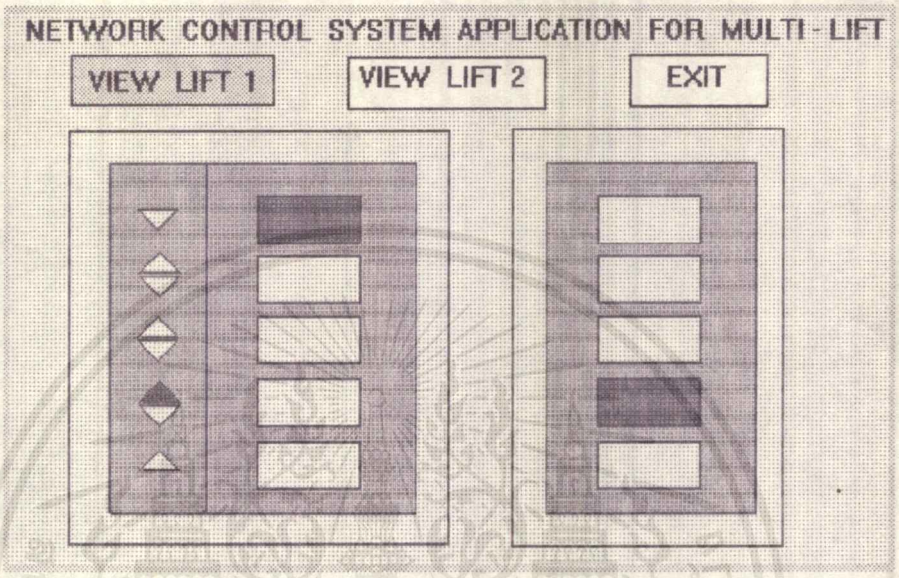


2. เมื่อมีการเรียกลงชั้นที่ 5 ลิฟท์จะทำการเปรียบเทียบความพร้อม และมีการสั่งให้ตัวที่ 1 ขึ้นไปรับ

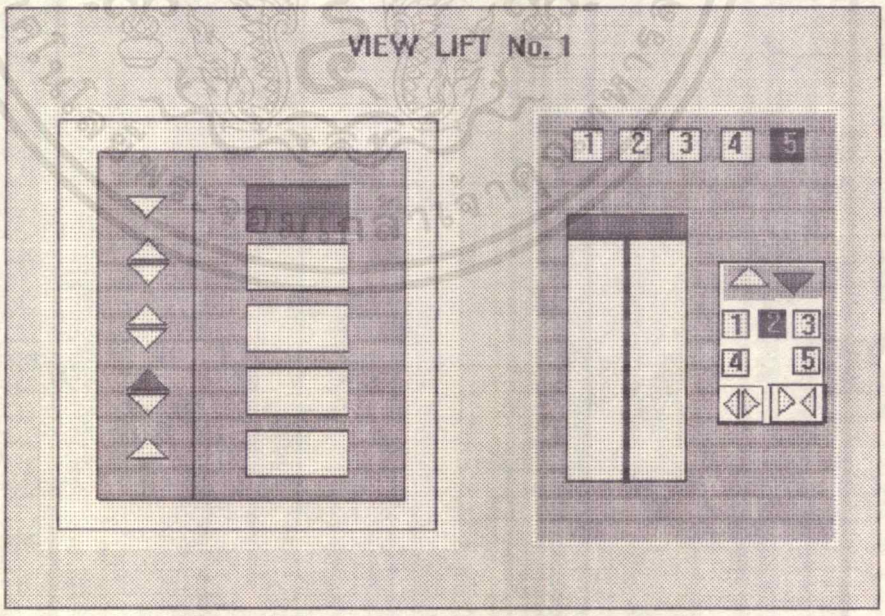


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ในสภาวะเดียวกันมีการเรียกขึ้นชั้นที่ 2 จะทำให้ลิฟท์ตัวที่ 2 ซึ่งมีความพร้อมในการทำงาน ขึ้นไปรับ

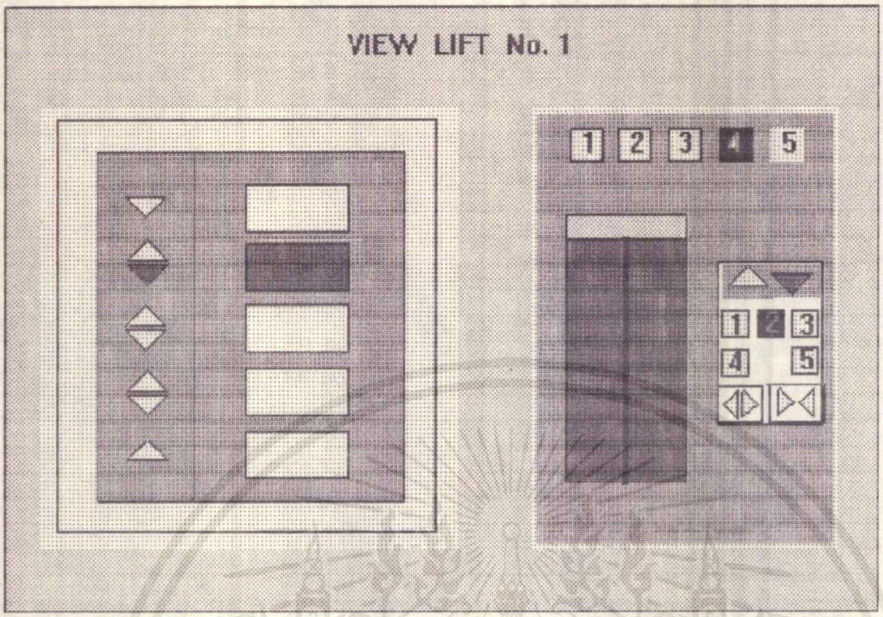


4. แสดงการทำงานของลิฟท์ตัวที่ 1 และสภาวะภายใน เมื่อมีความต้องการลงมาชั้นที่ 2 และ ลิฟท์ในขณะนั้นอยู่ชั้นที่ 5

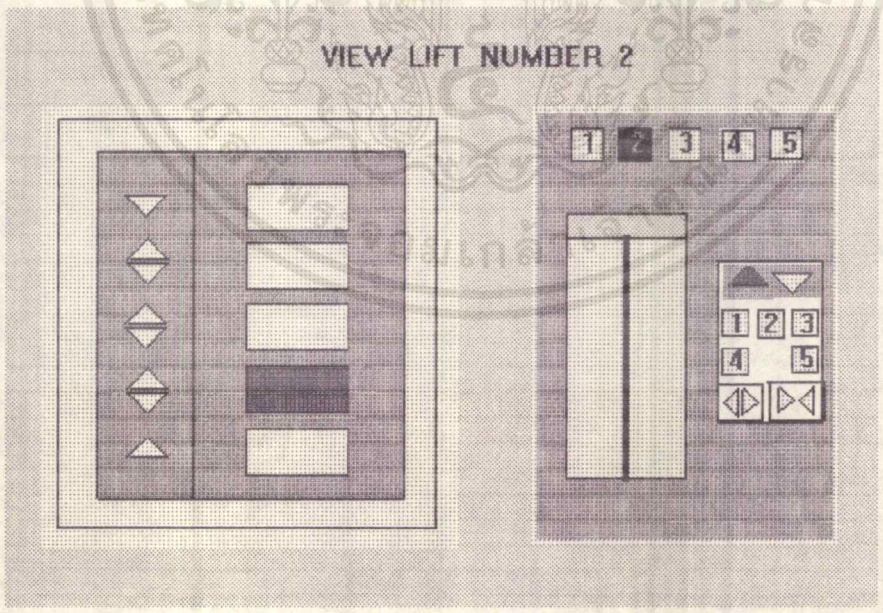


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. แสดงสภาวะเมื่อลิฟท์ลงมาถึงชั้นที่4 และในชั้นนี้มีคนเรียกลงด้วยจึงทำให้ลิฟท์หยุด และมีการเปิดประตูรับ



6. แสดงการทำงานของลิฟท์ตัวที่ 2 ขณะที่ลิฟท์อยู่ชั้นที่ 2 และต้องการขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ข้าพเจ้าและเพื่อนร่วมกลุ่มที่ทำโครงงานนี้ ขอขอบพระคุณ อาจารย์ สุพรรณ กุลพาณิชย์ ที่มีความกรุณารับเป็นที่ปรึกษาในการทำโครงงานนี้ ซึ่งท่านคอยให้คำแนะนำ ชี้แนะ พร้อมทั้งให้ข้อมูลที่มีประโยชน์ที่ทำให้โครงงานนี้สำเร็จ ลุล่วงลงได้ และขอขอบพระคุณอาจารย์ท่านอื่นที่ประสาทวิชาความรู้ ที่เป็นแนวทางในการดำเนินชีวิตต่อไป และอีกท่านที่ลืมไม่ได้ นาย สมชาย เหรียญเงิน ที่เปิดบริษัทให้ข้าพเจ้าทำการพิมพ์เอกสาร

ข้าพเจ้าหวังว่าโครงงานนี้คงมีประโยชน์มากสำหรับผู้ที่มีความสนใจ ในการค้นคว้าทางด้านนี้ ซึ่งความดีที่ได้จากโครงงานนี้ขอมอบให้ พ่อ แม่ และอาจารย์ ทุกท่าน ร่วมทั้งผู้ให้กำลังใจทุกคน

ผู้จัดทำ

พ.ย. 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ผศ. สุพรรณ กุลพาณิชย์ : PROGRAMMABLE CONTROLLER เทคนิคและการใช้งานเบื้องต้น : ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง : 2533

รศ. กฤษดา วิศวธีรานนท์ : PC ตัวควบคุมซีเควันซ์ หลักการทำงานและการประยุกต์ ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย : ตุลาคม 2534

OMRON : PC LINK SYSTEM MANUAL , OMRON TATEISI ELECTRONICS , REVISED, JUNE 1990

OMRON : C 200 H - ASC 04 ASCII UNIT , OMRON TATEISI ELECTRONICS, REVISED, JULY 1990

OMRON : C 200 H OPERATION MANUAL , OMRON TATEISI ELECTRONICS, REVISED, JULY 1990

ธันวา ศรีประโมง : การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม : มหาวิทยาลัยเทคโนโลยีมหานคร : 2537

จิรศักดิ์ เหลืองอุไร : การสื่อสารอนุกรมบน PC : บริษัท ซีเอ็ดยูเคชั่น จำกัด : 2538

ประทีป บัญญัตินพรัติ : การเขียนโปรแกรมภาษา BASIC : คณะวิศวกรรมศาสตร์ สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง : 2529

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก 1

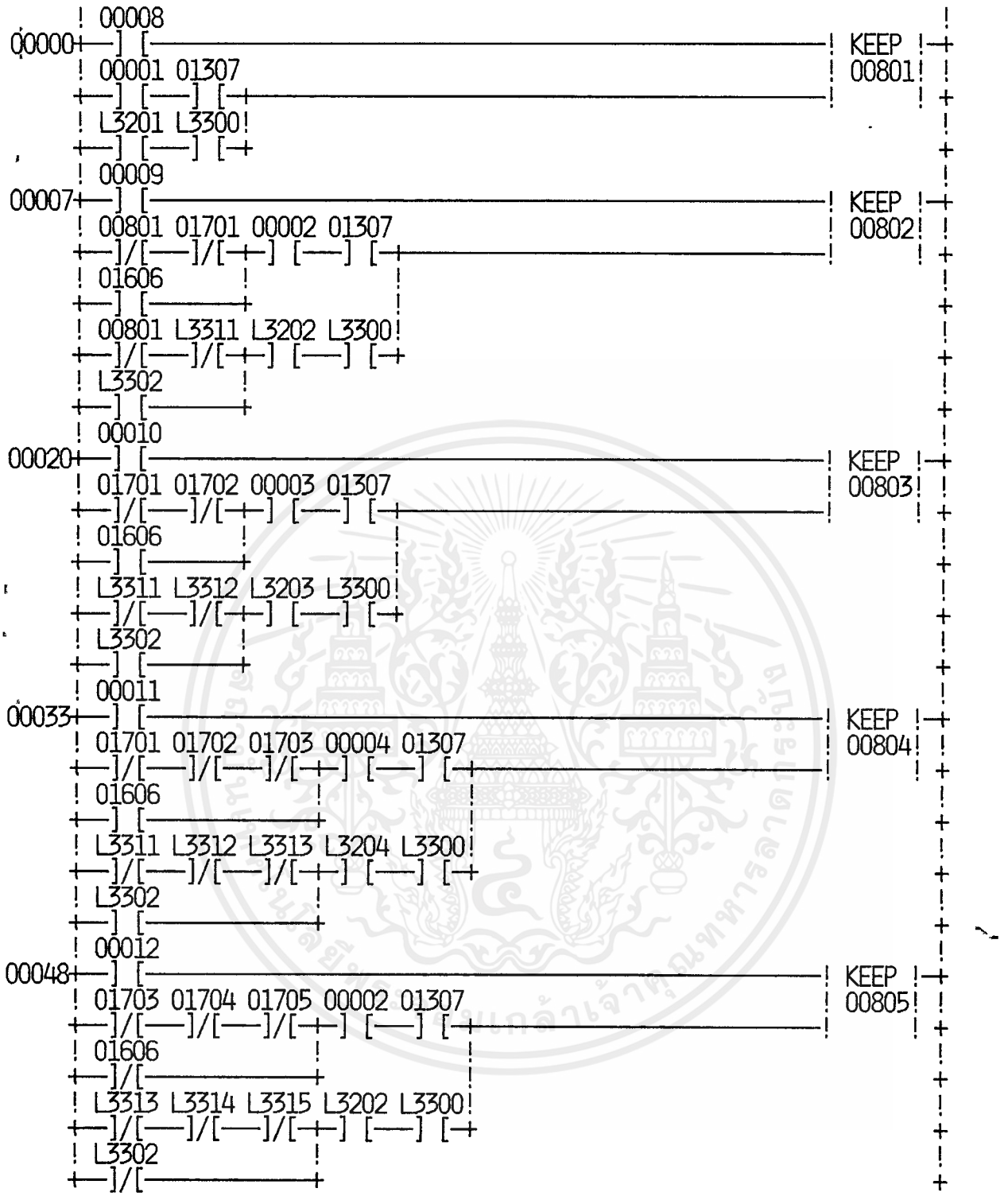
โปรแกรมการติดต่อระหว่าง PC กับ PC และ สั่งให้ระบบทำงาน

(LADDER DIAGRAM)

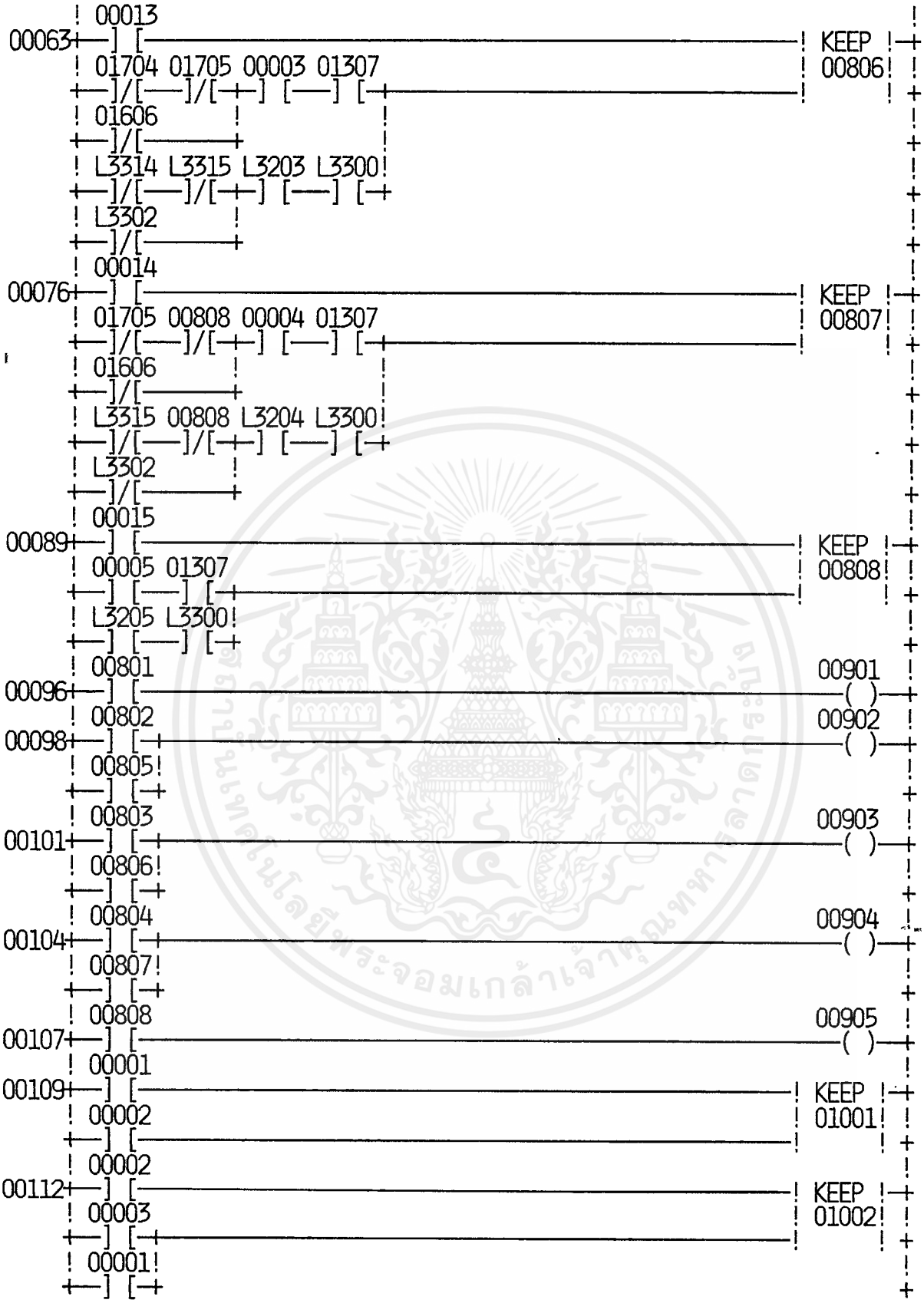
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU UNIT # O (MASTER)

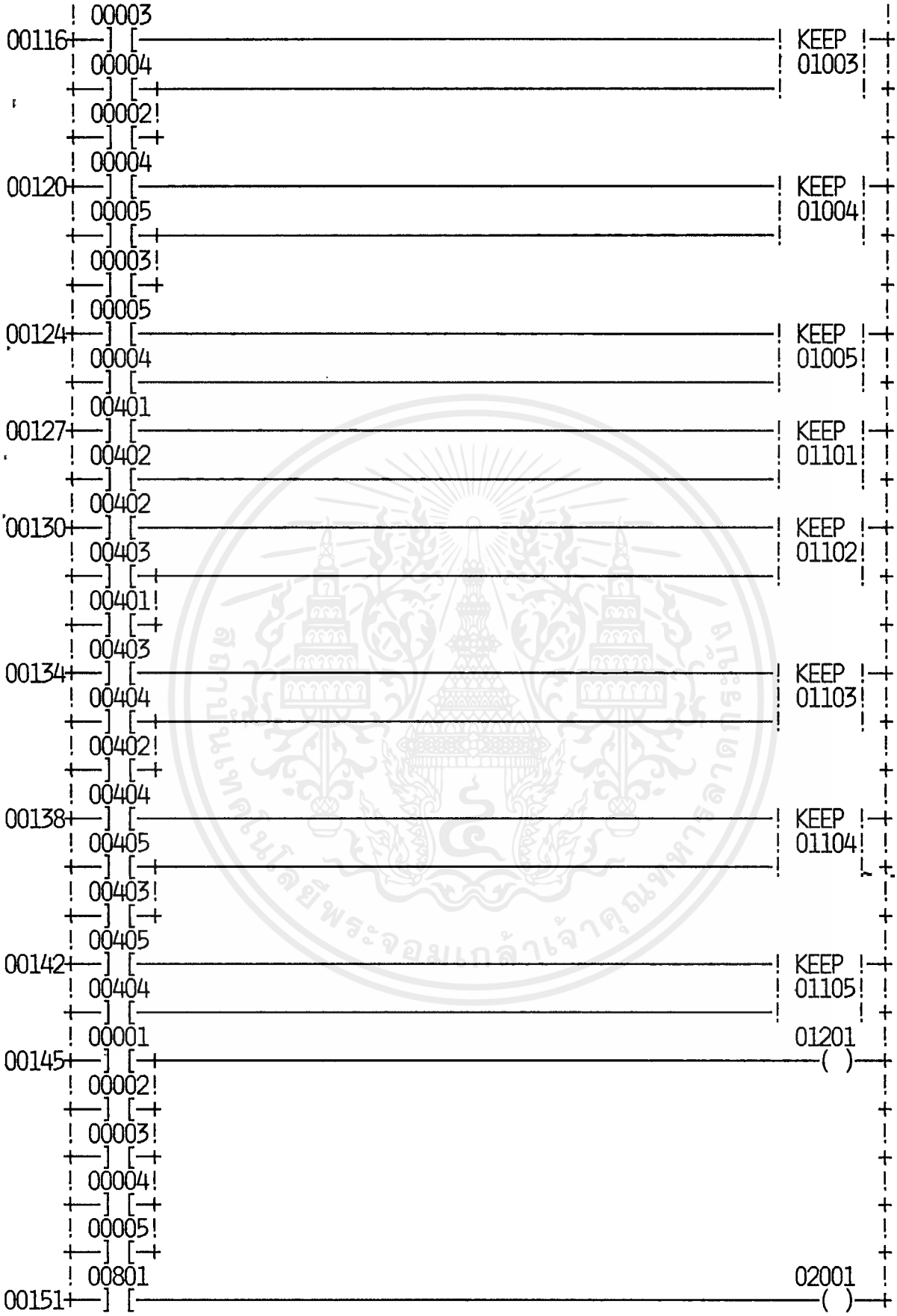
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



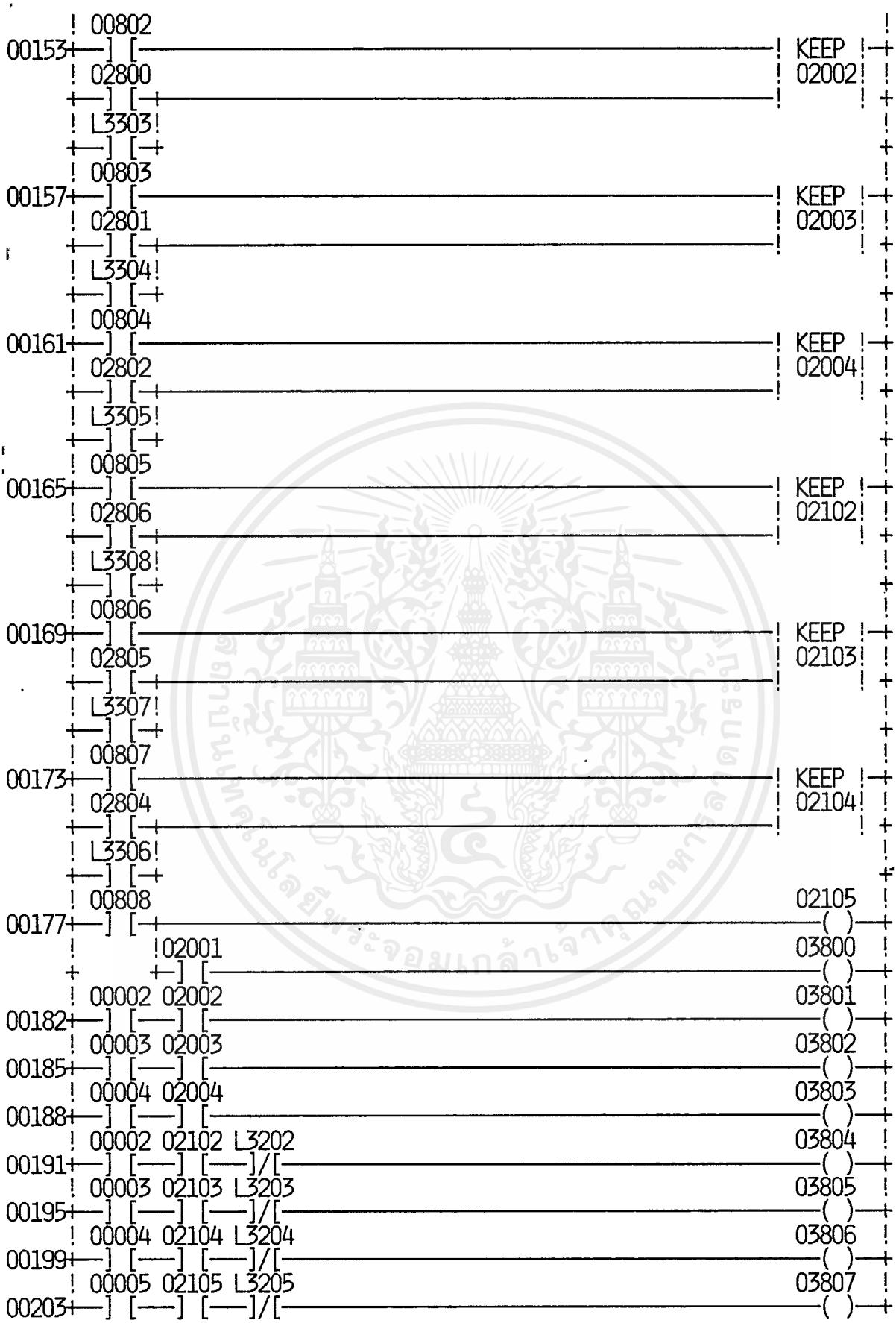
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



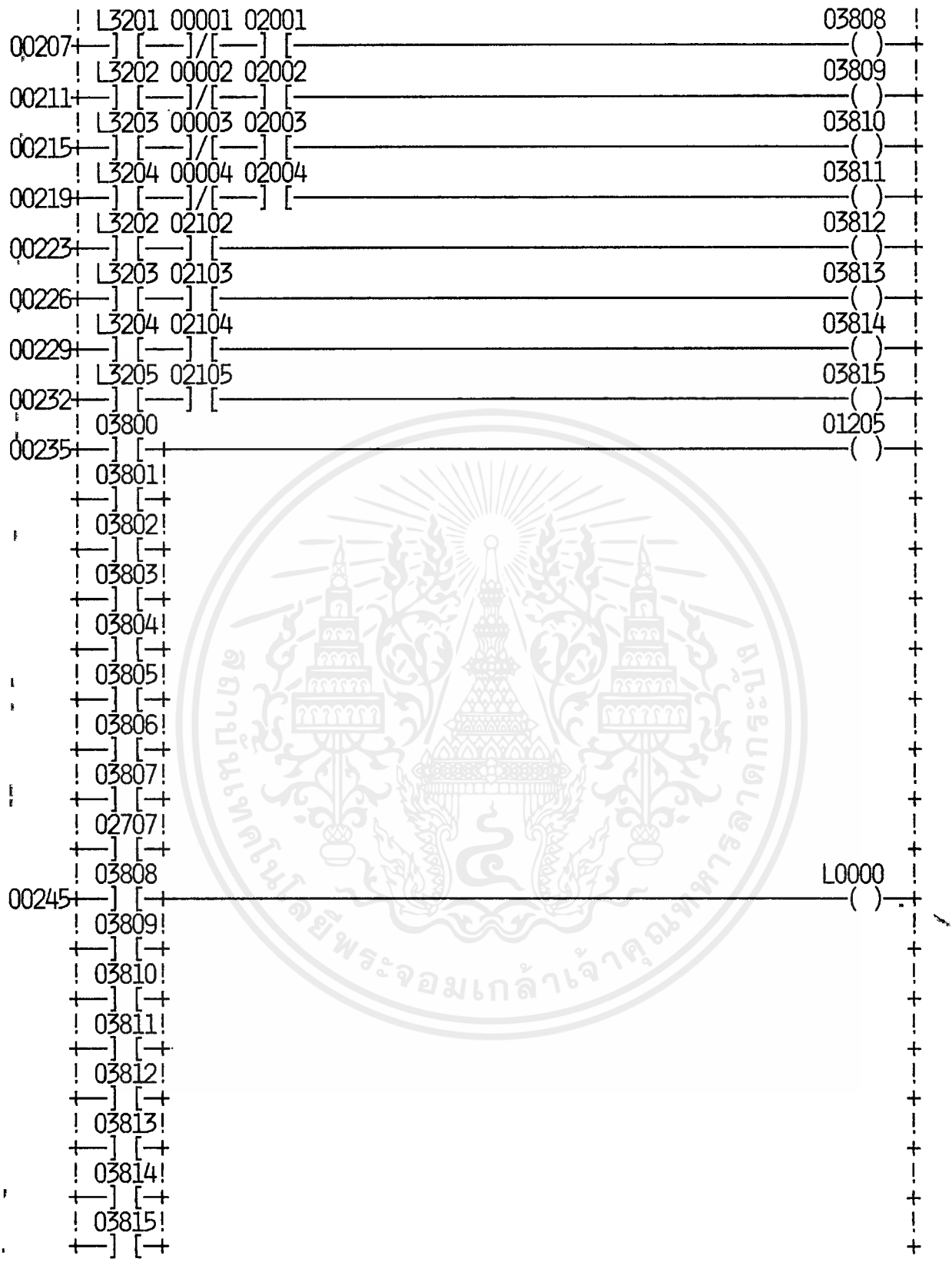
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



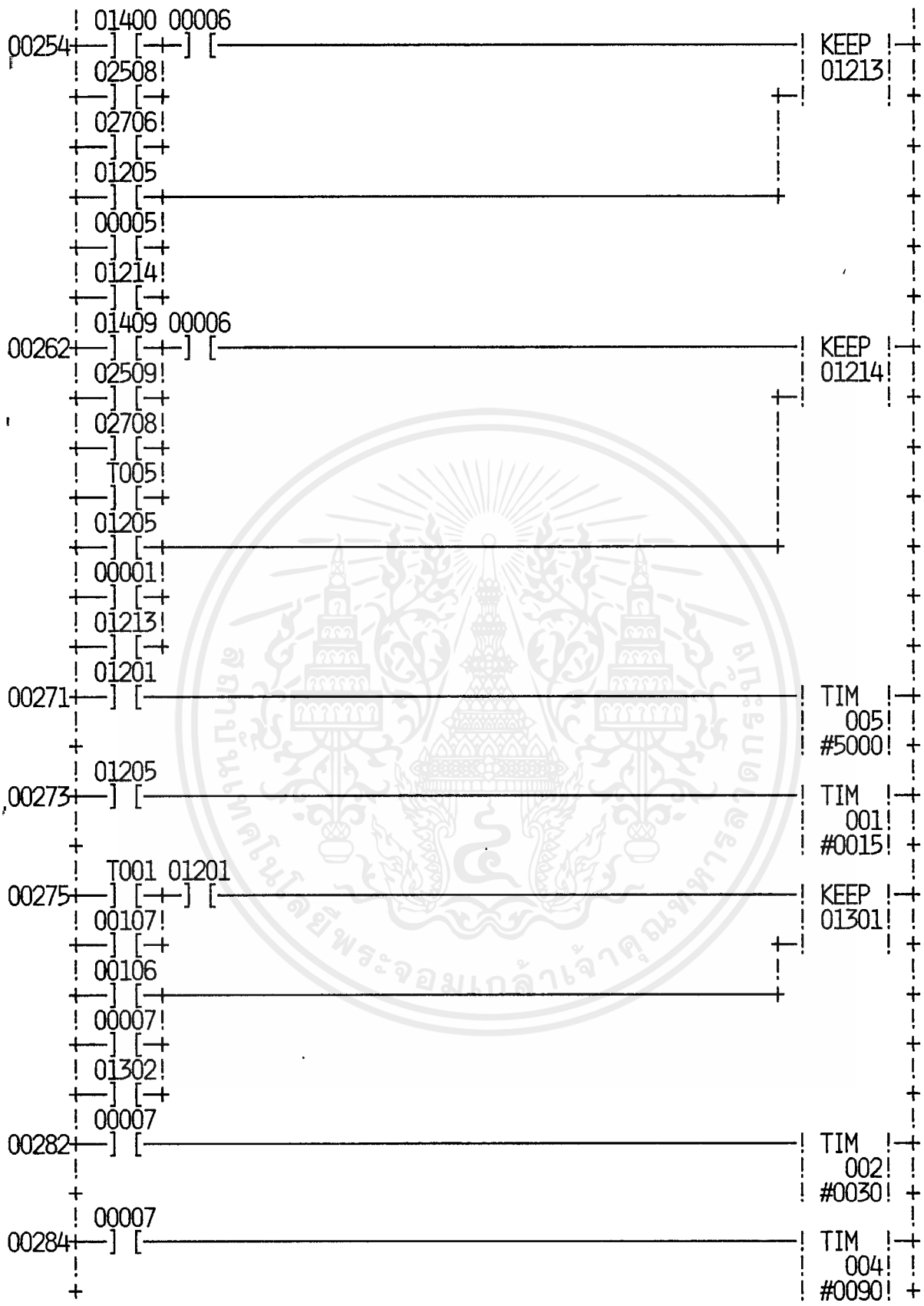
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



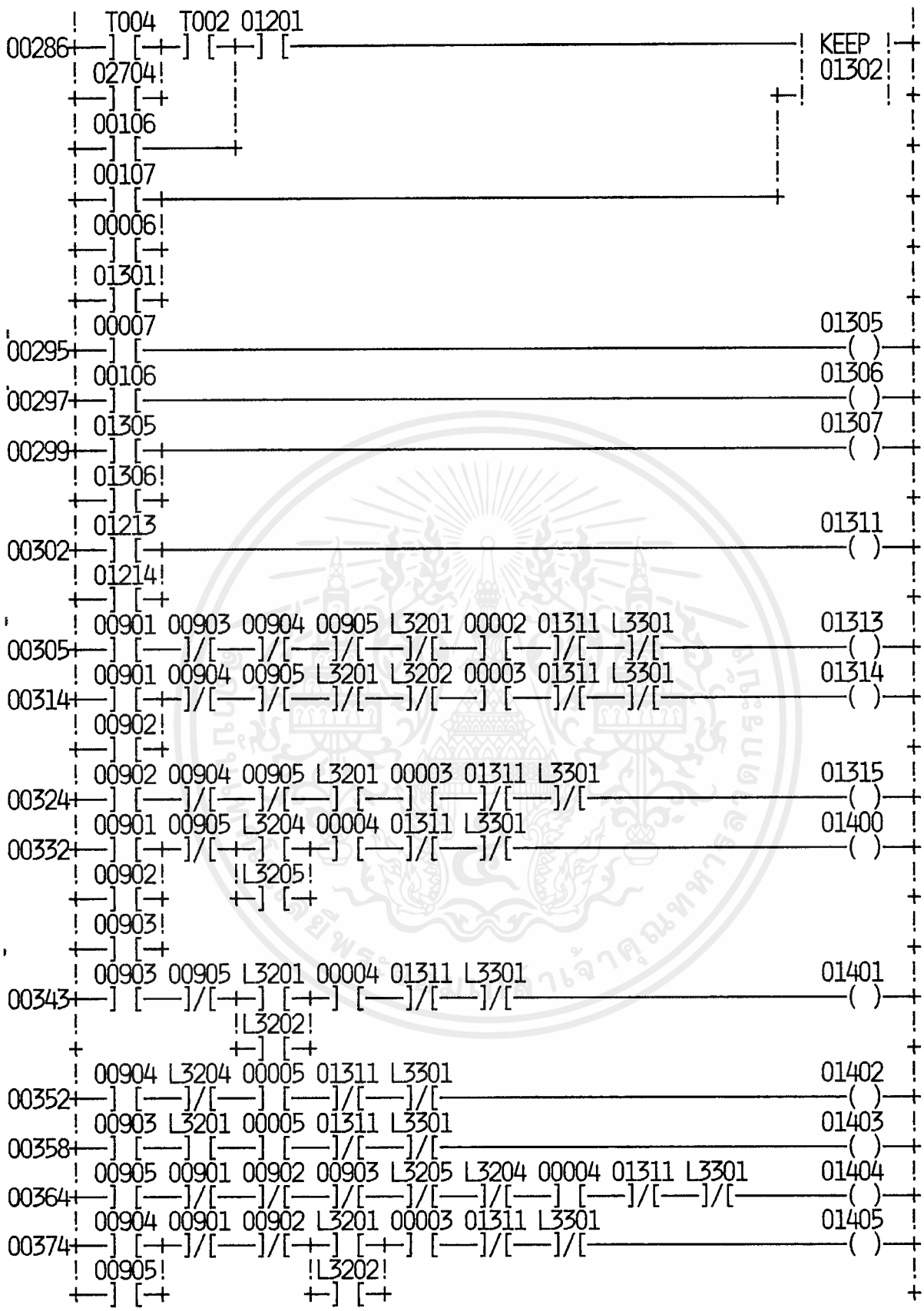
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



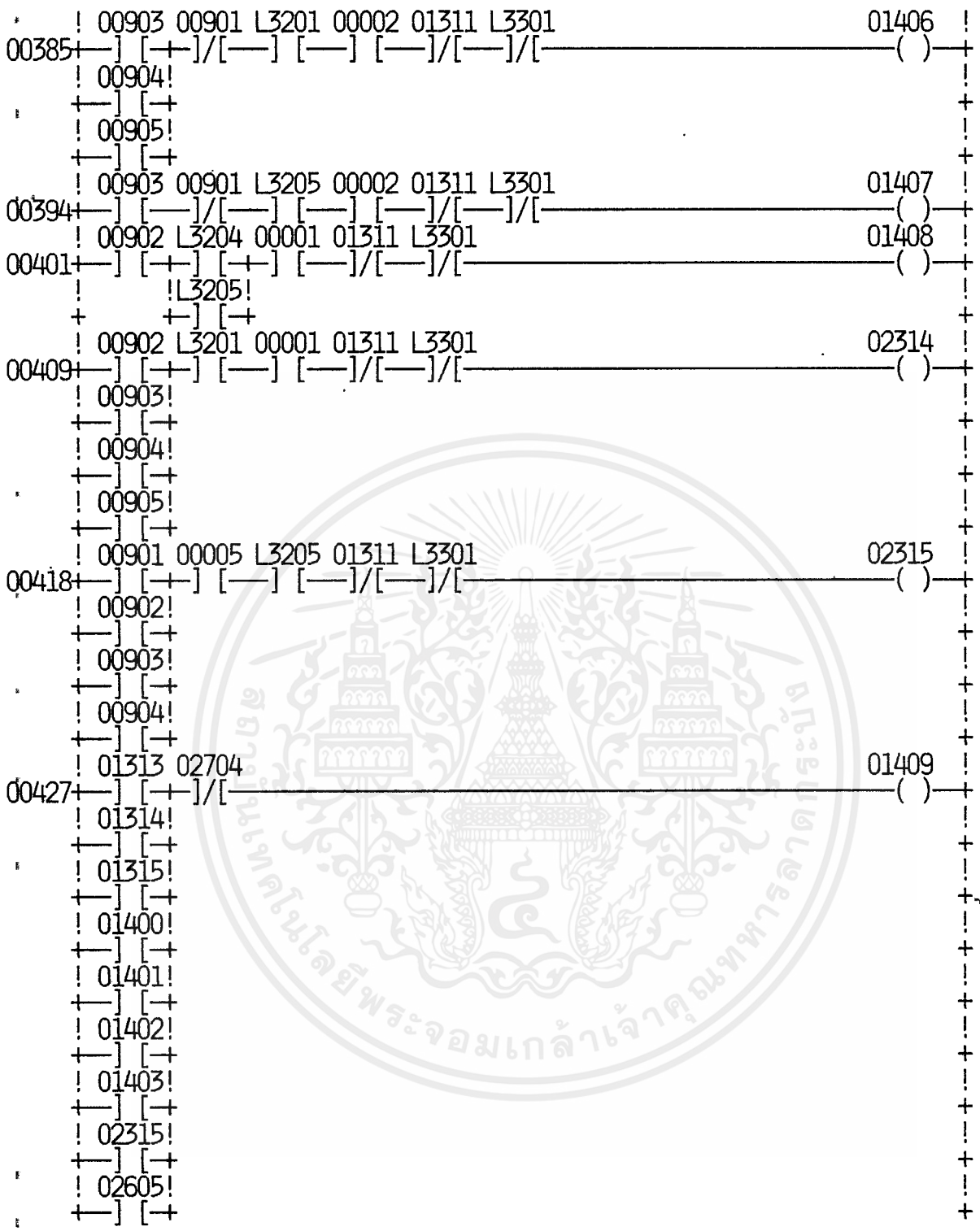
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



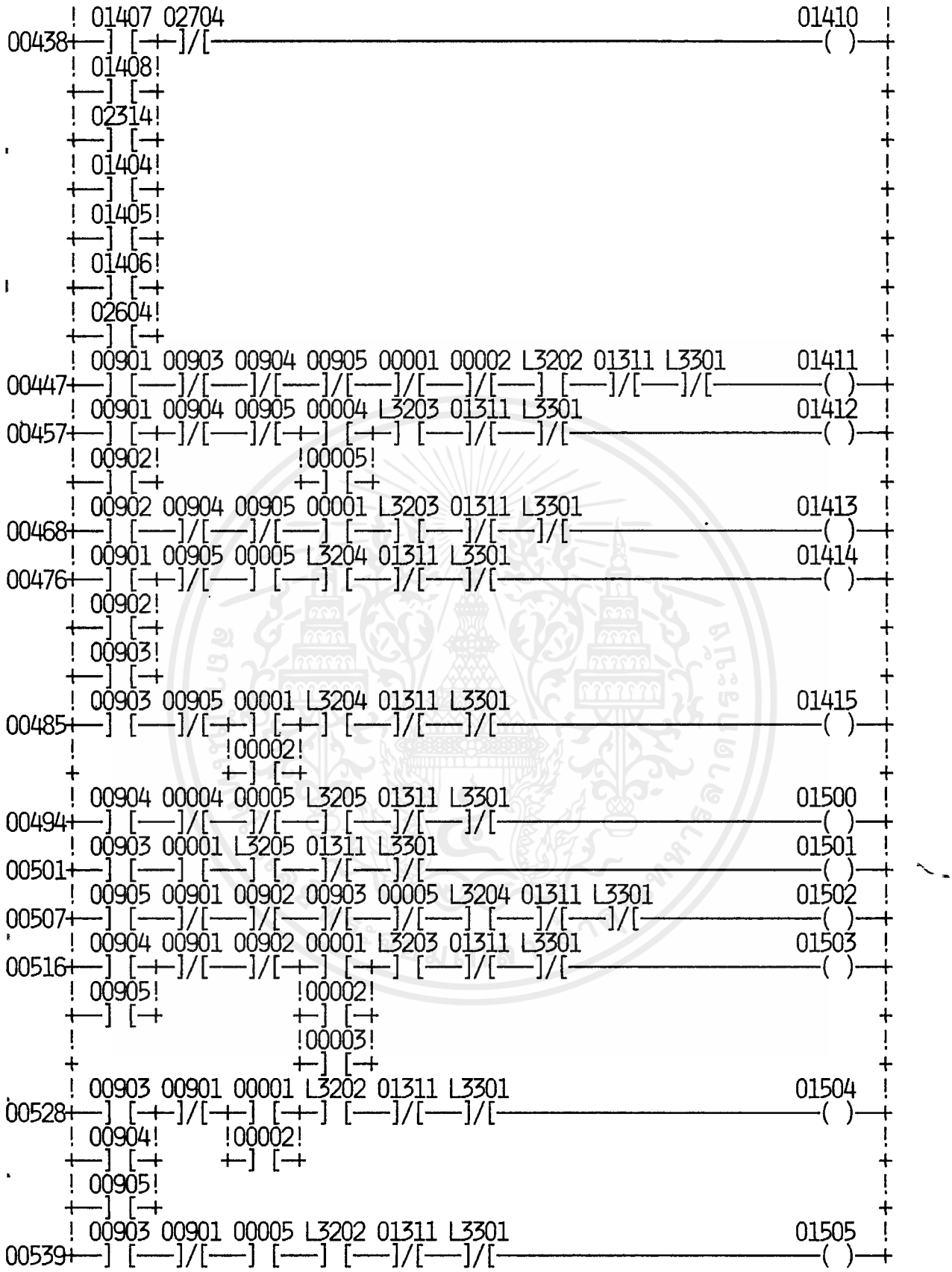
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



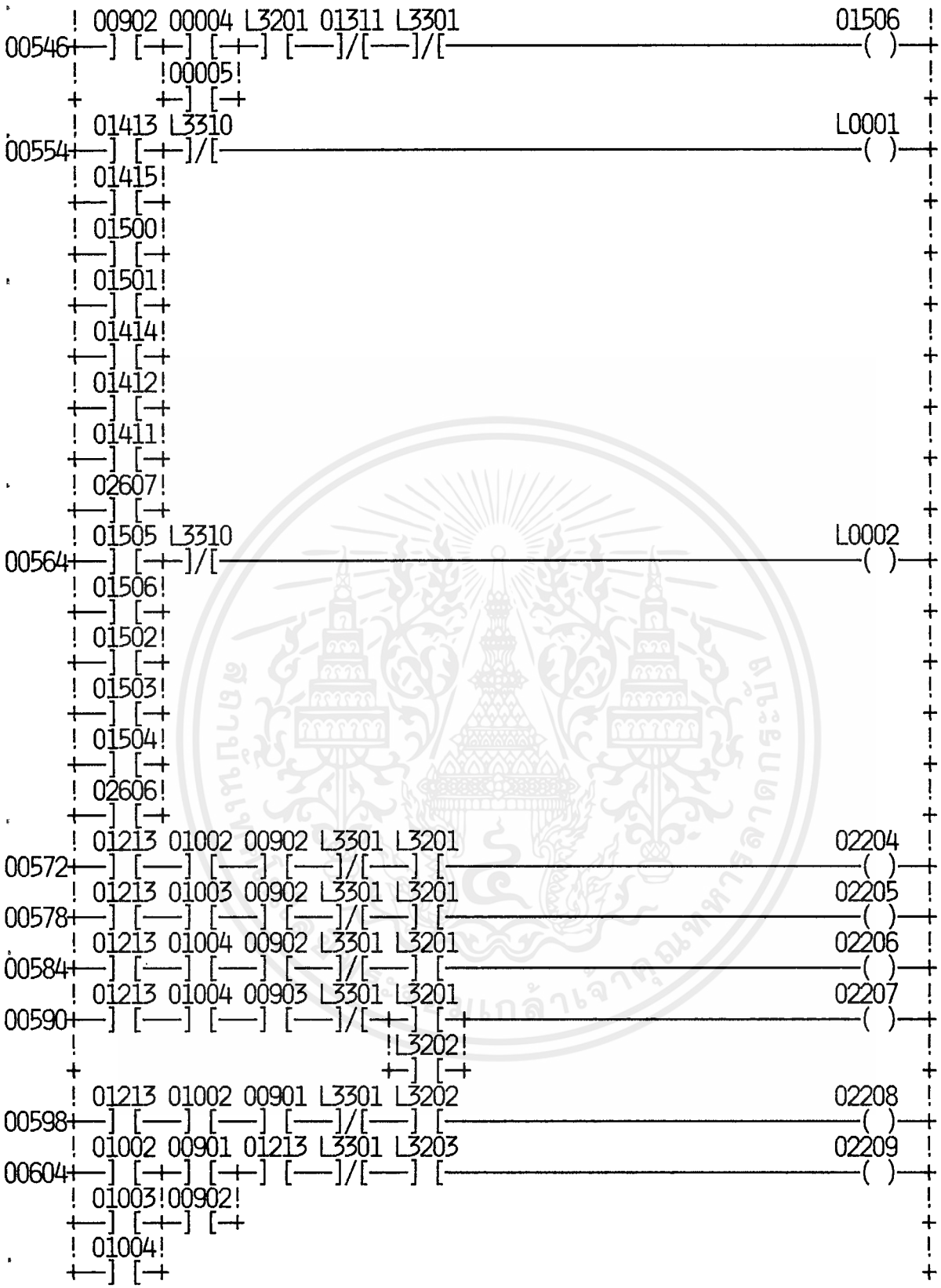
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



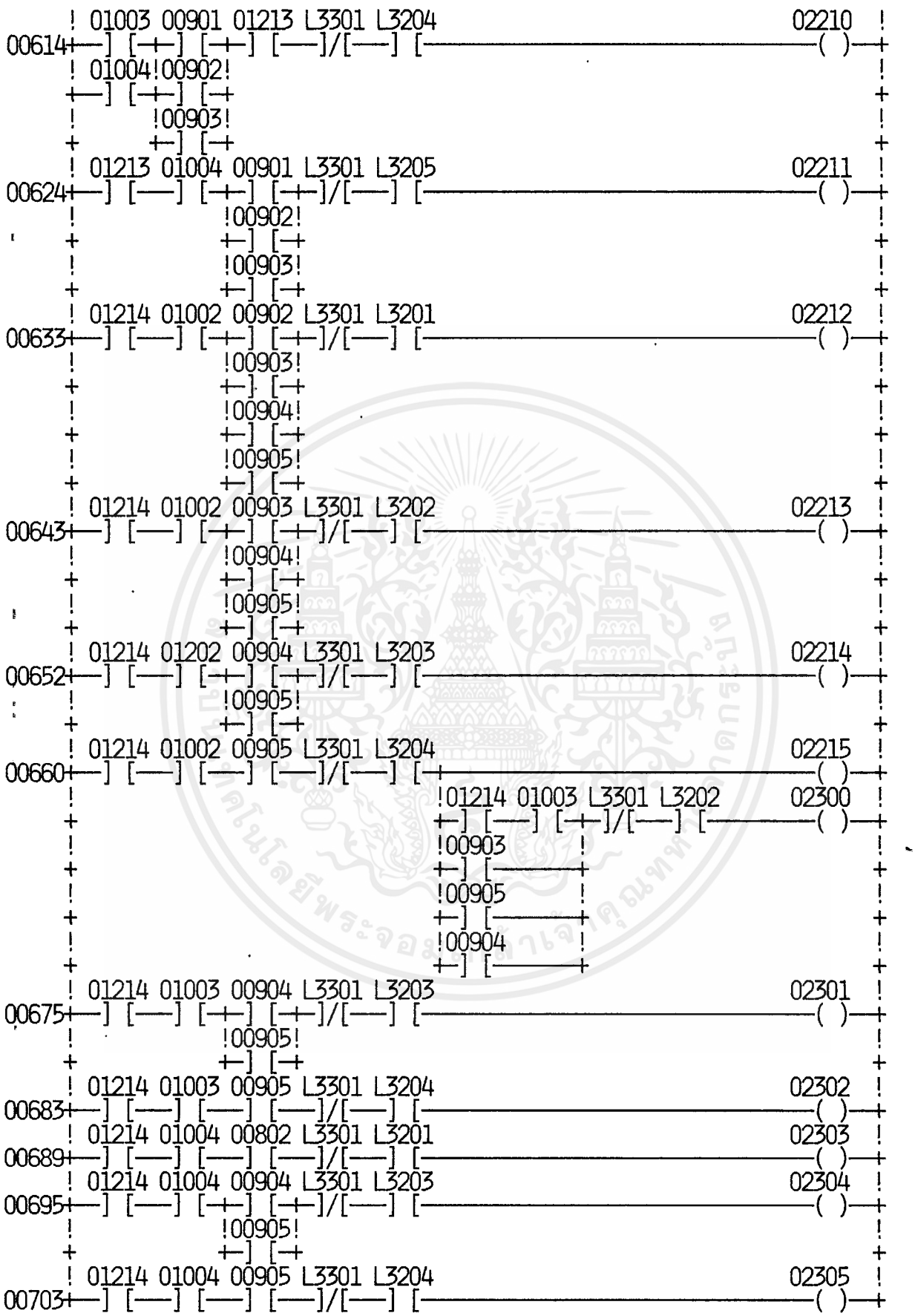
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



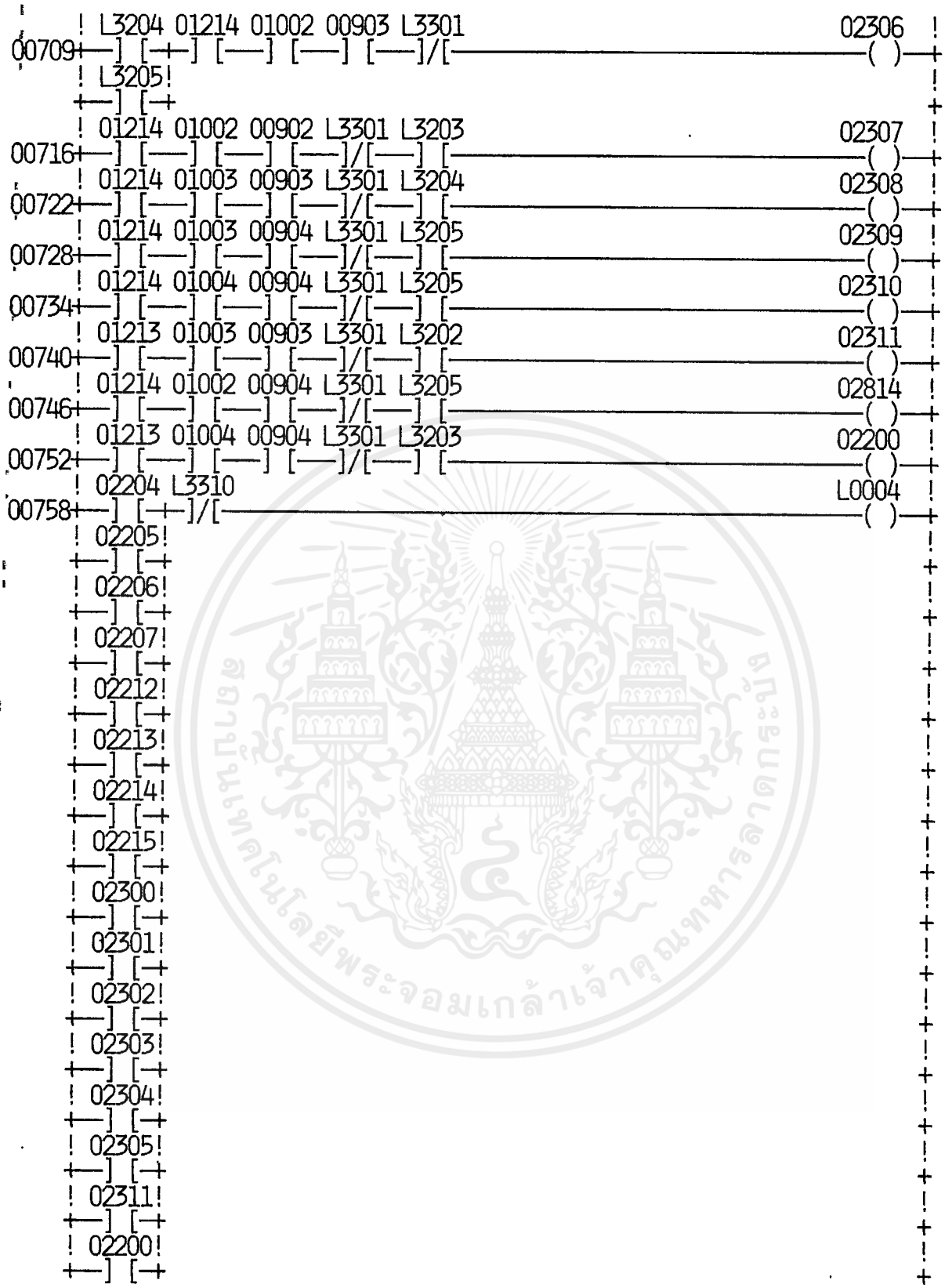
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



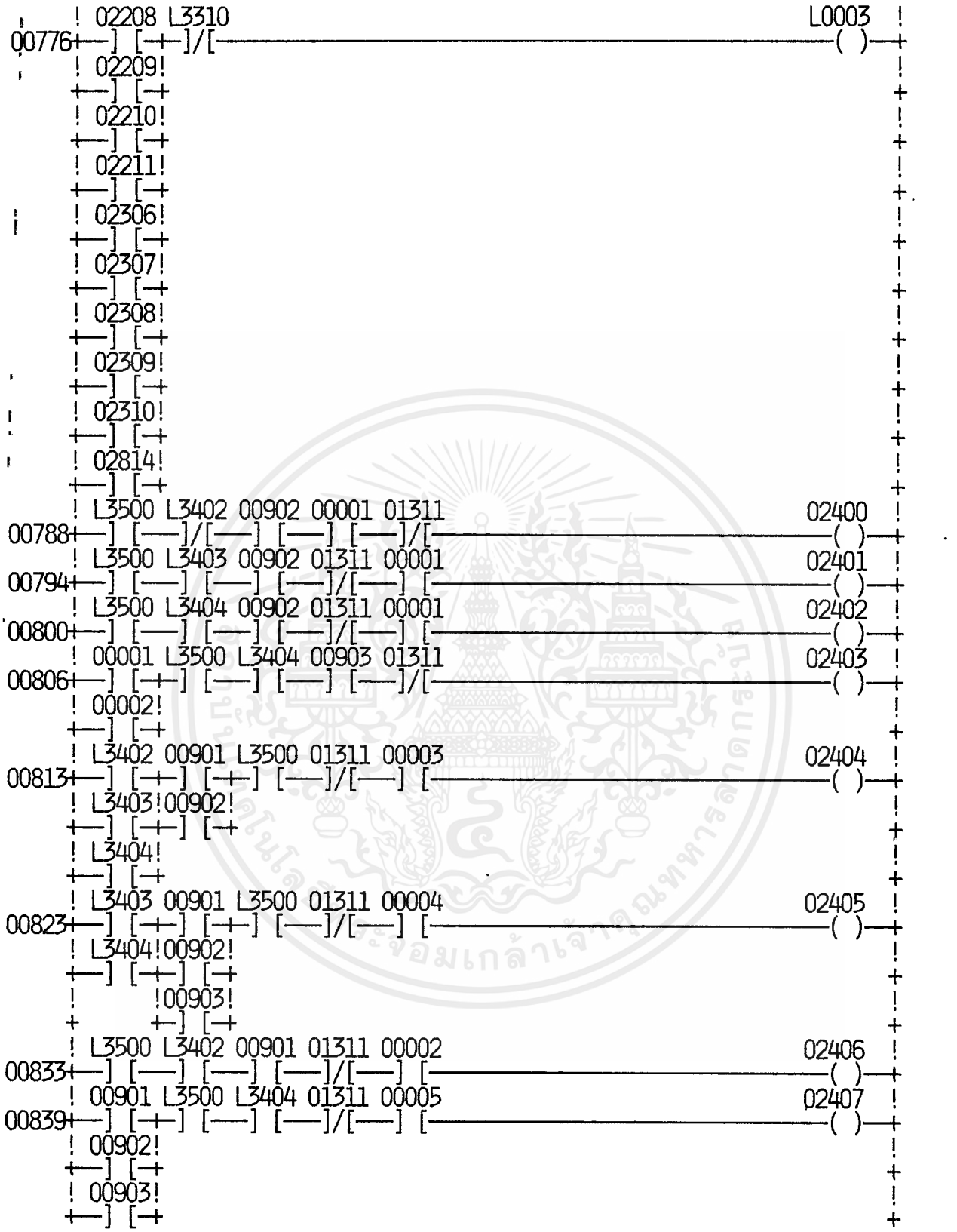
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



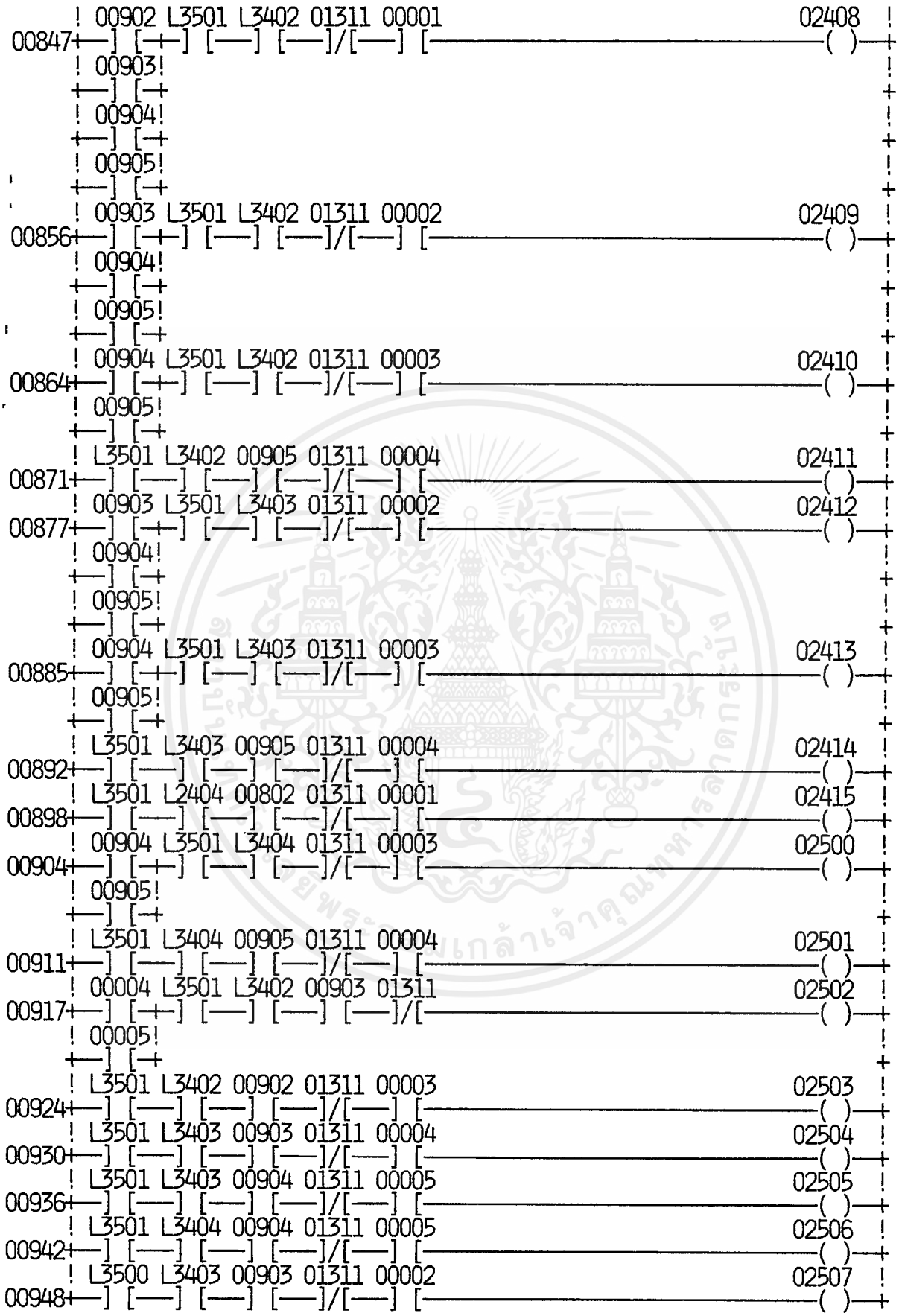
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



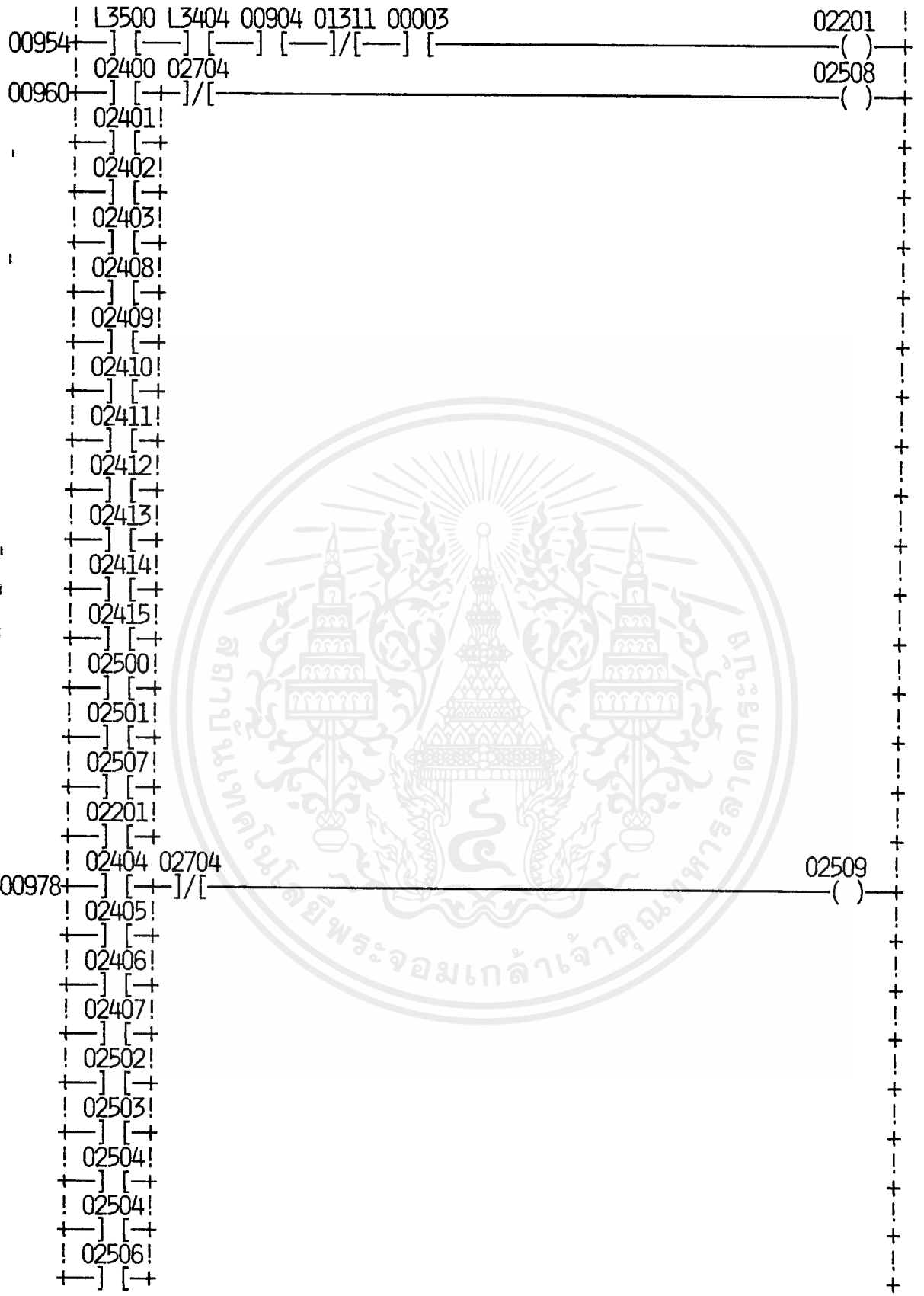
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



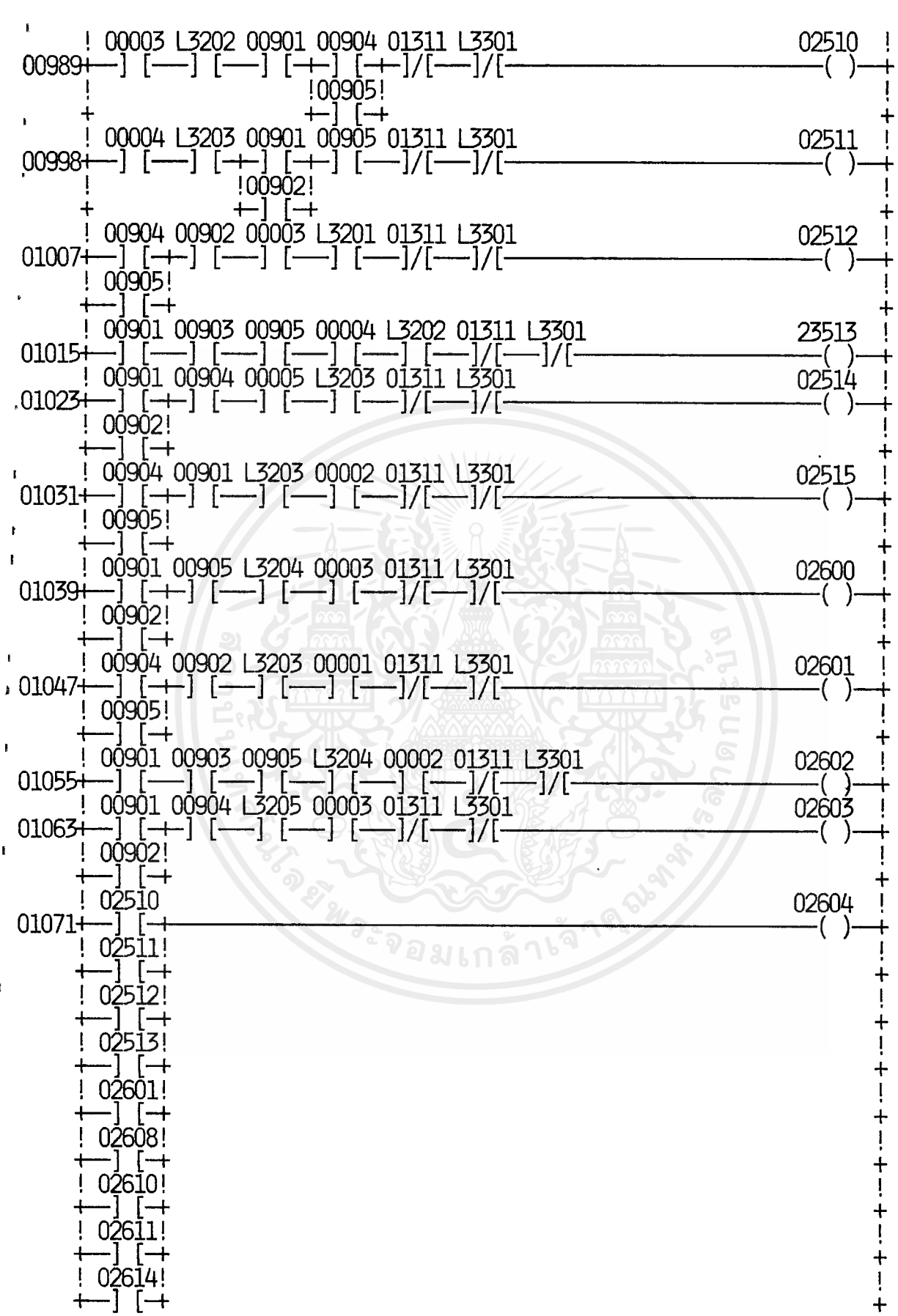
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



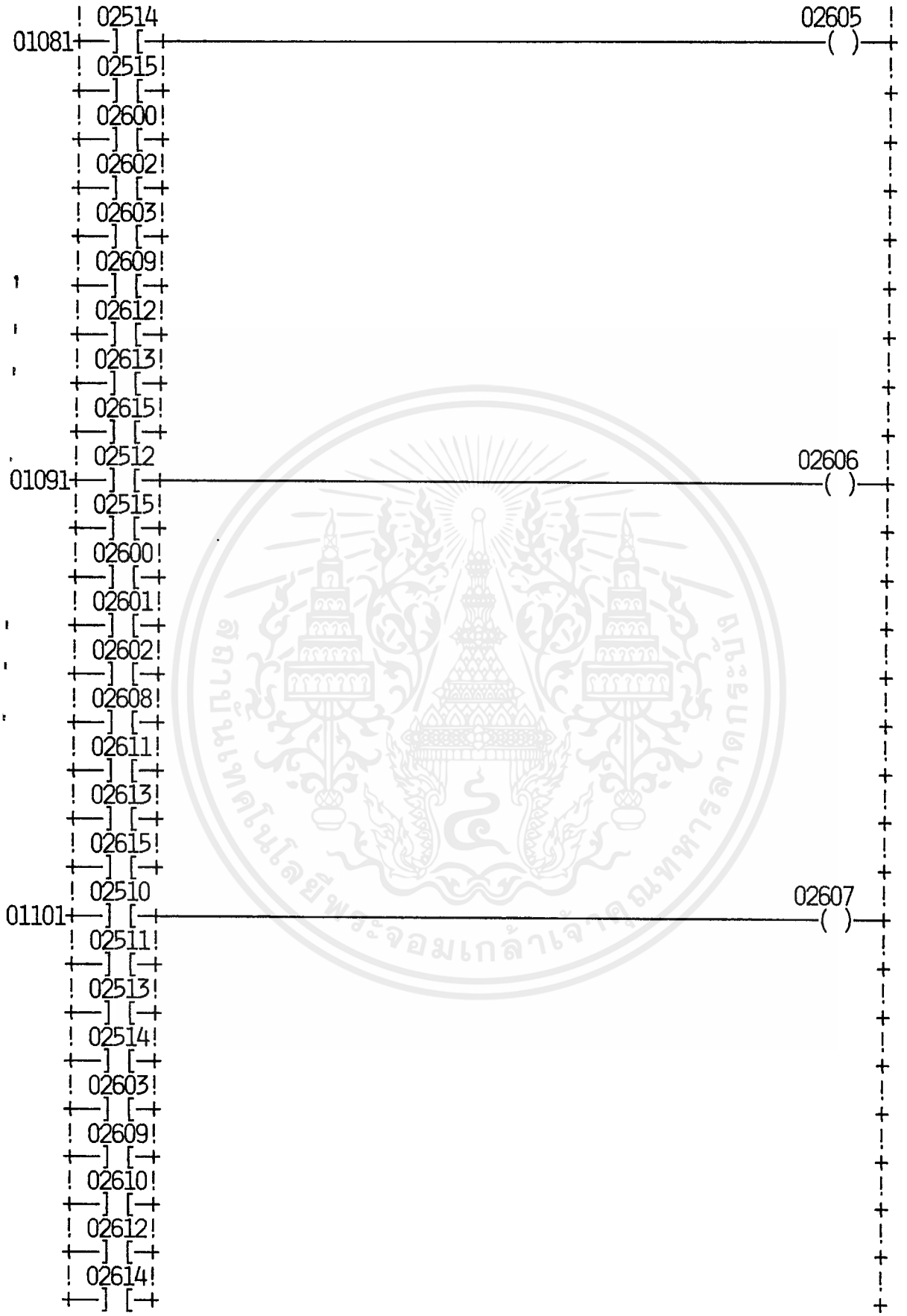
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



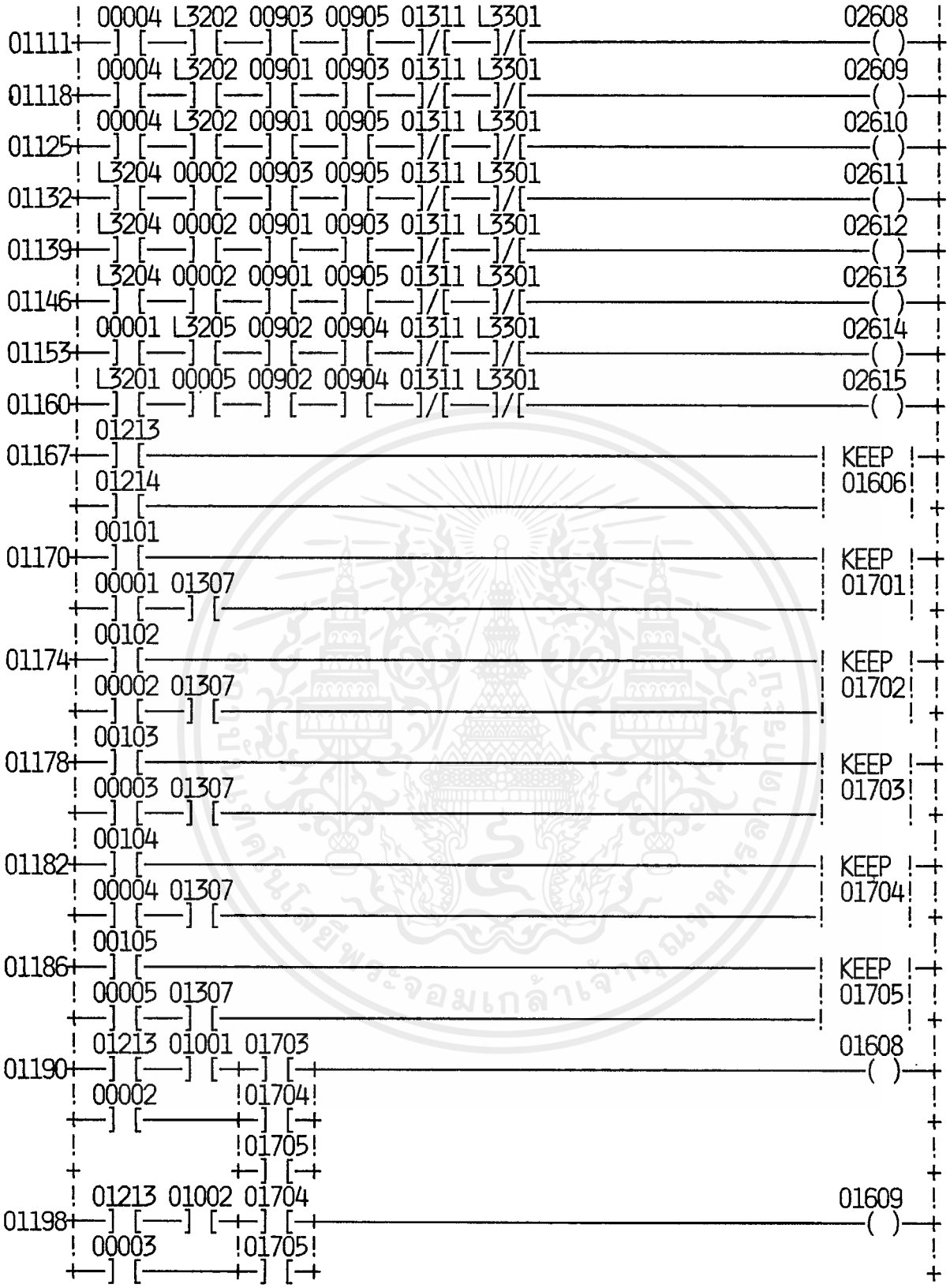
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



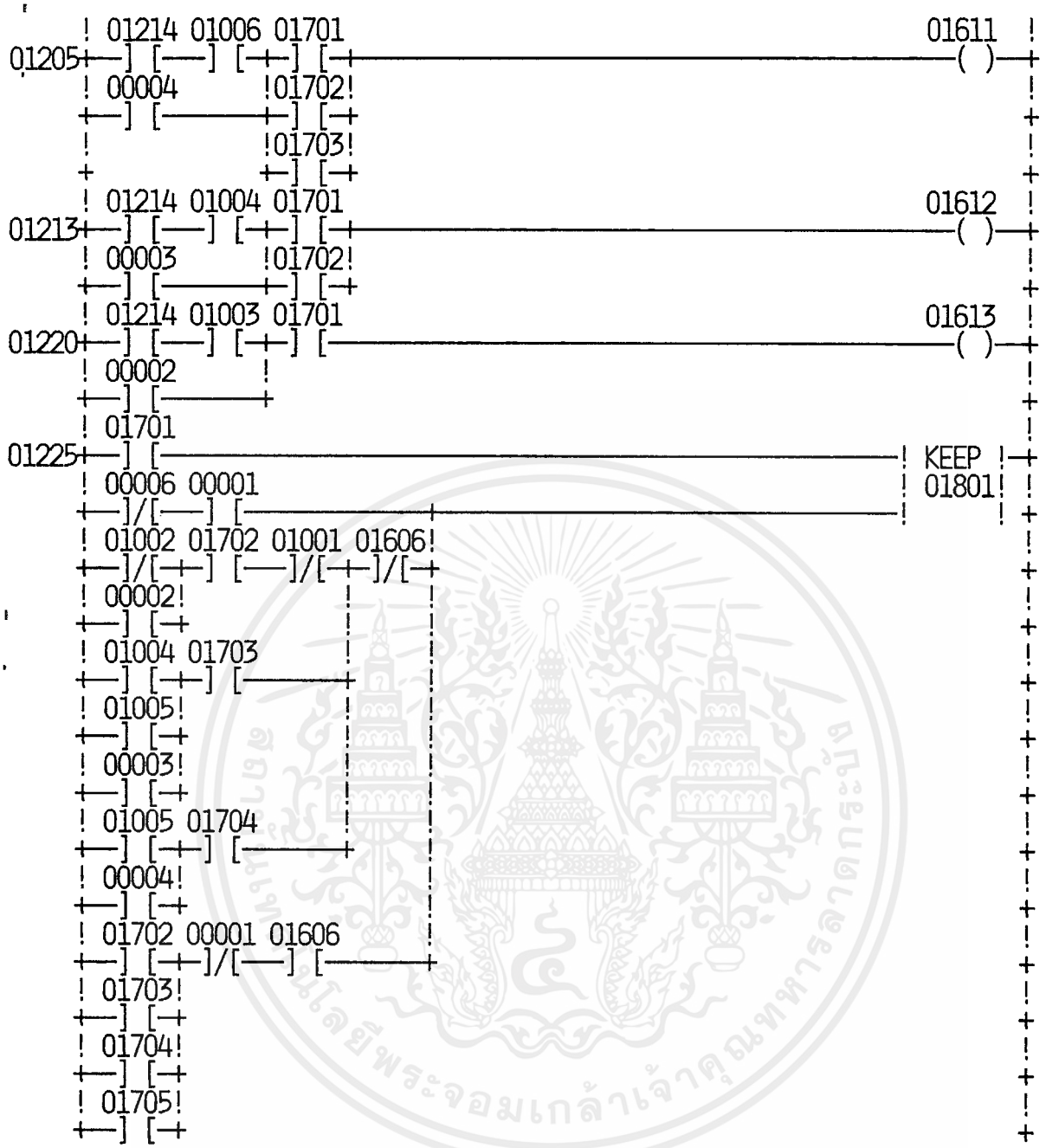
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



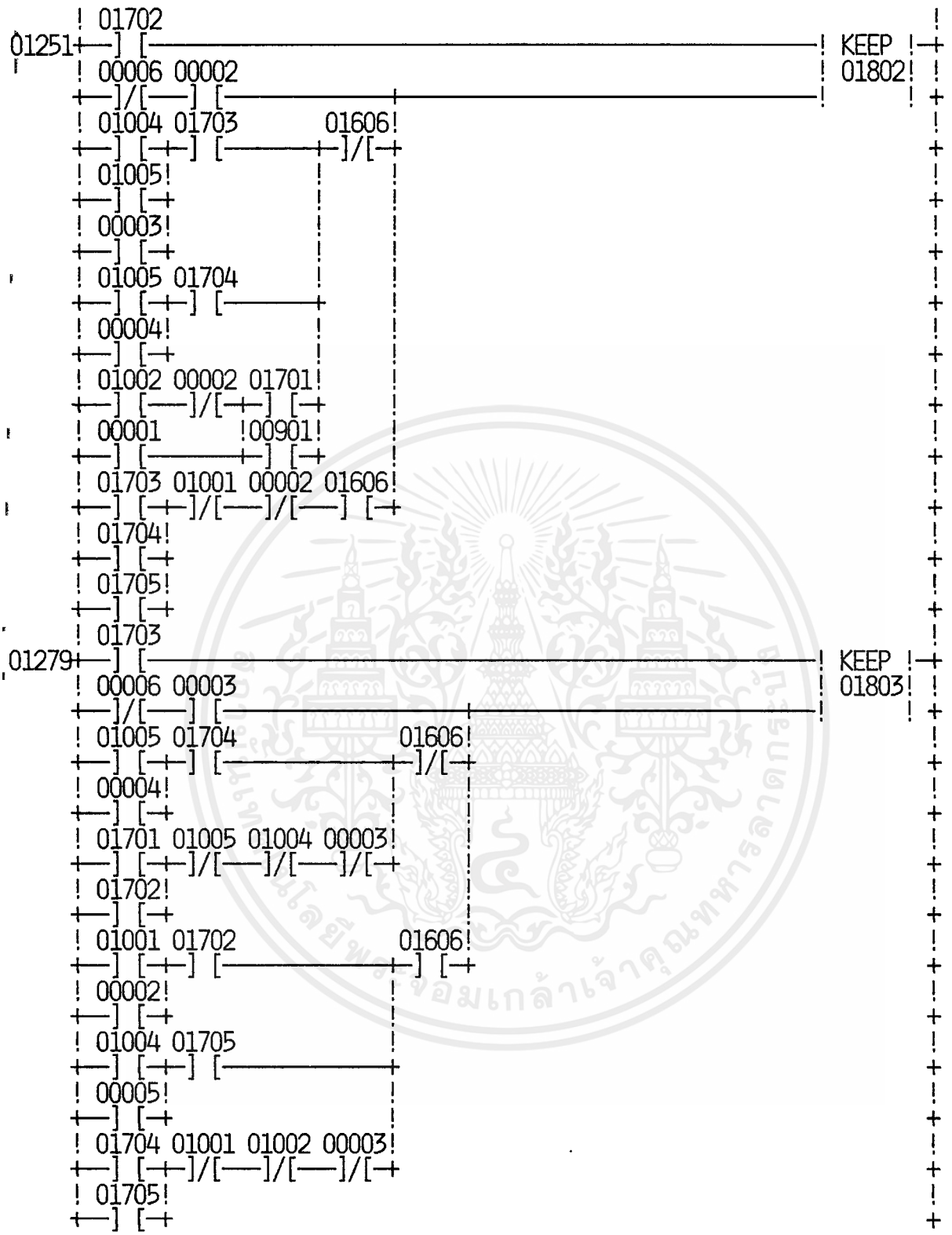
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



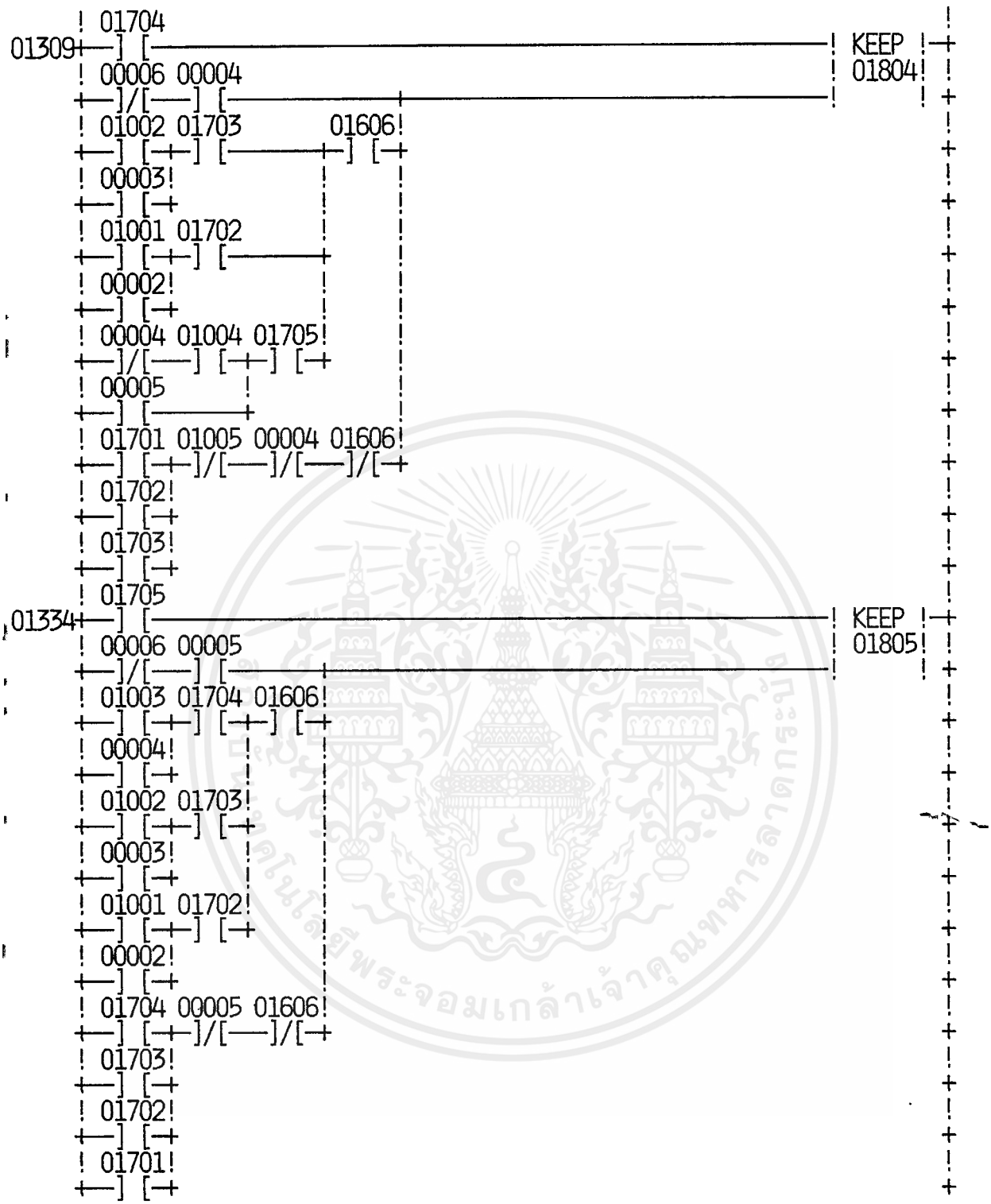
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



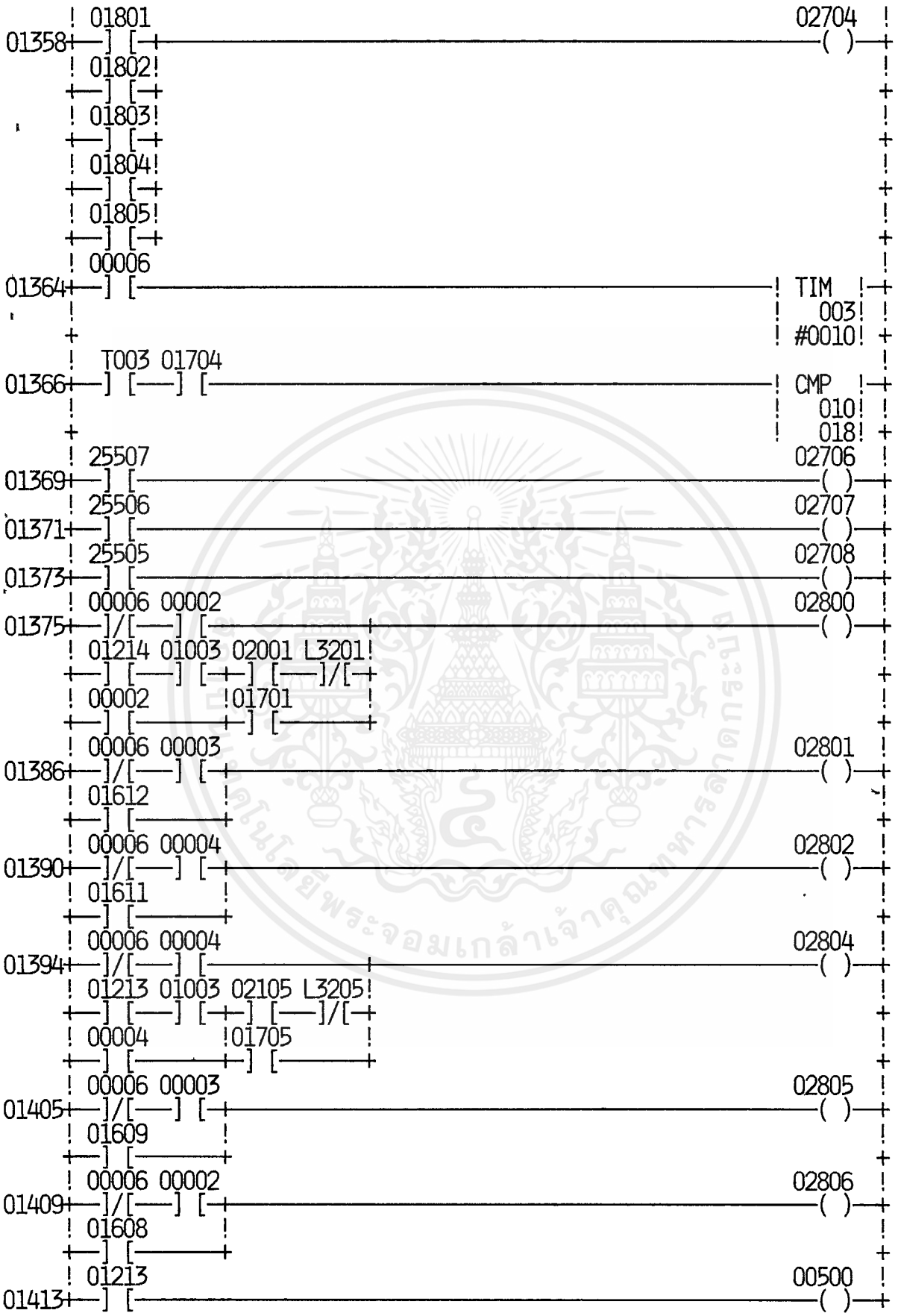
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



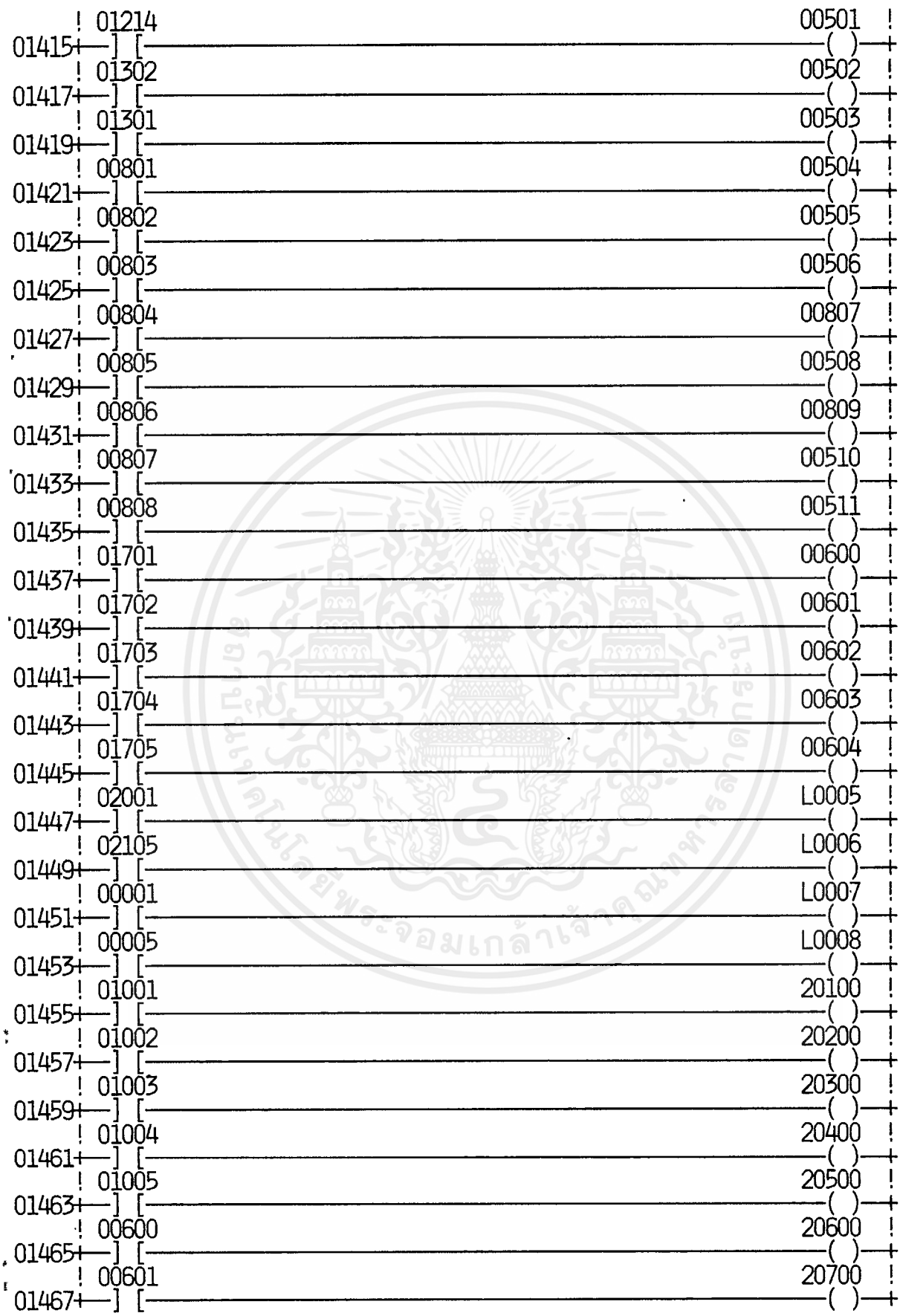
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



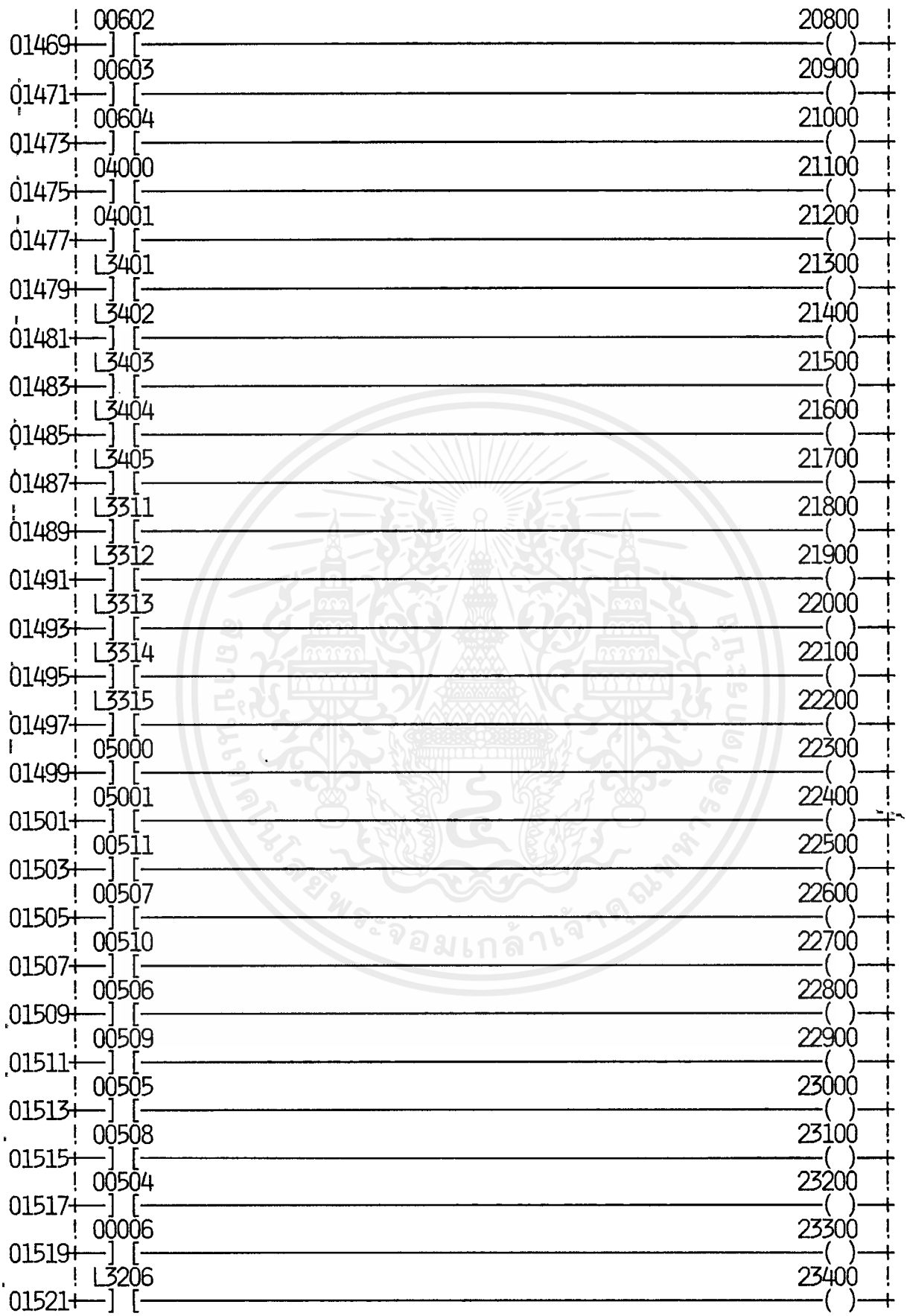
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



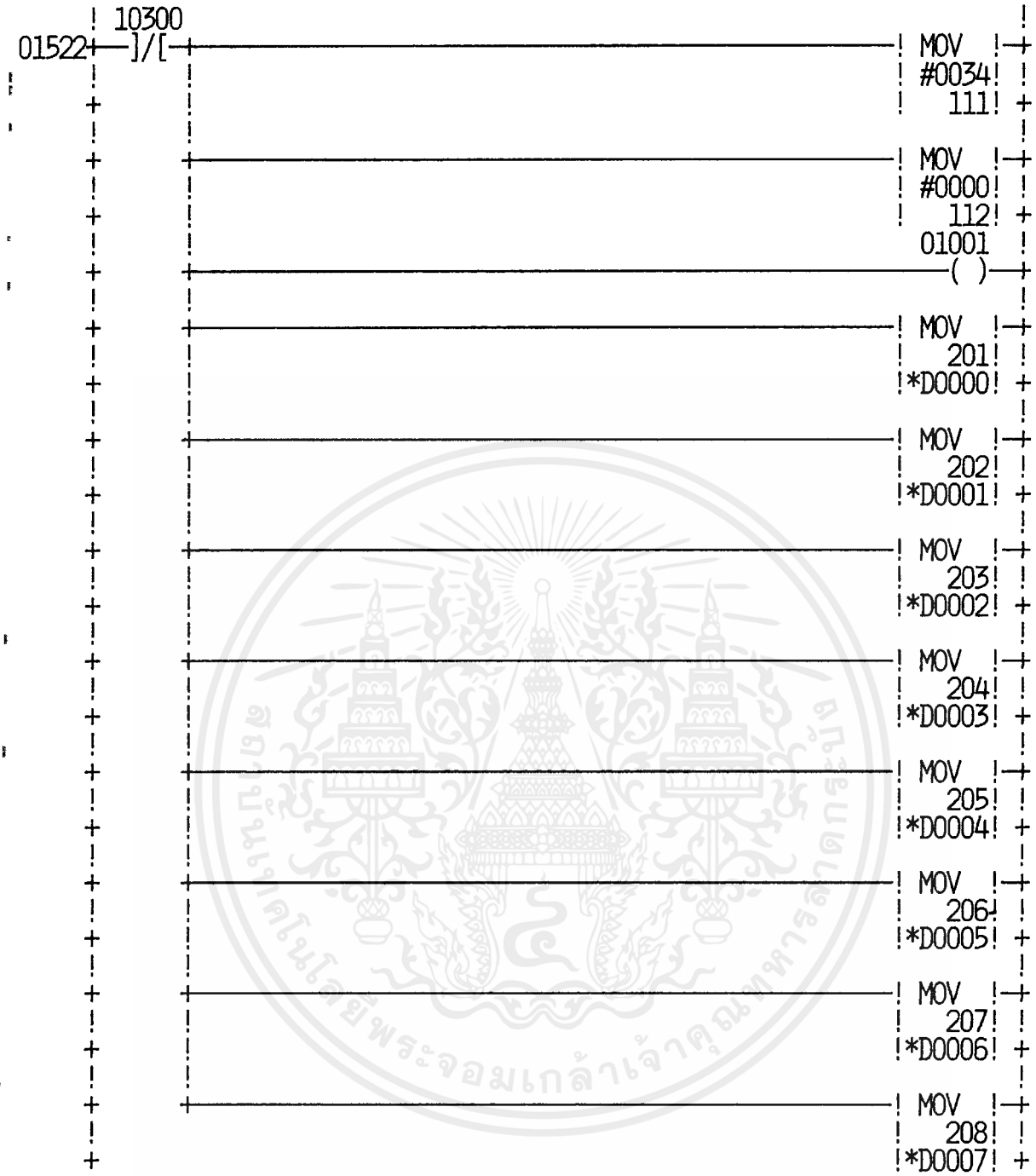
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



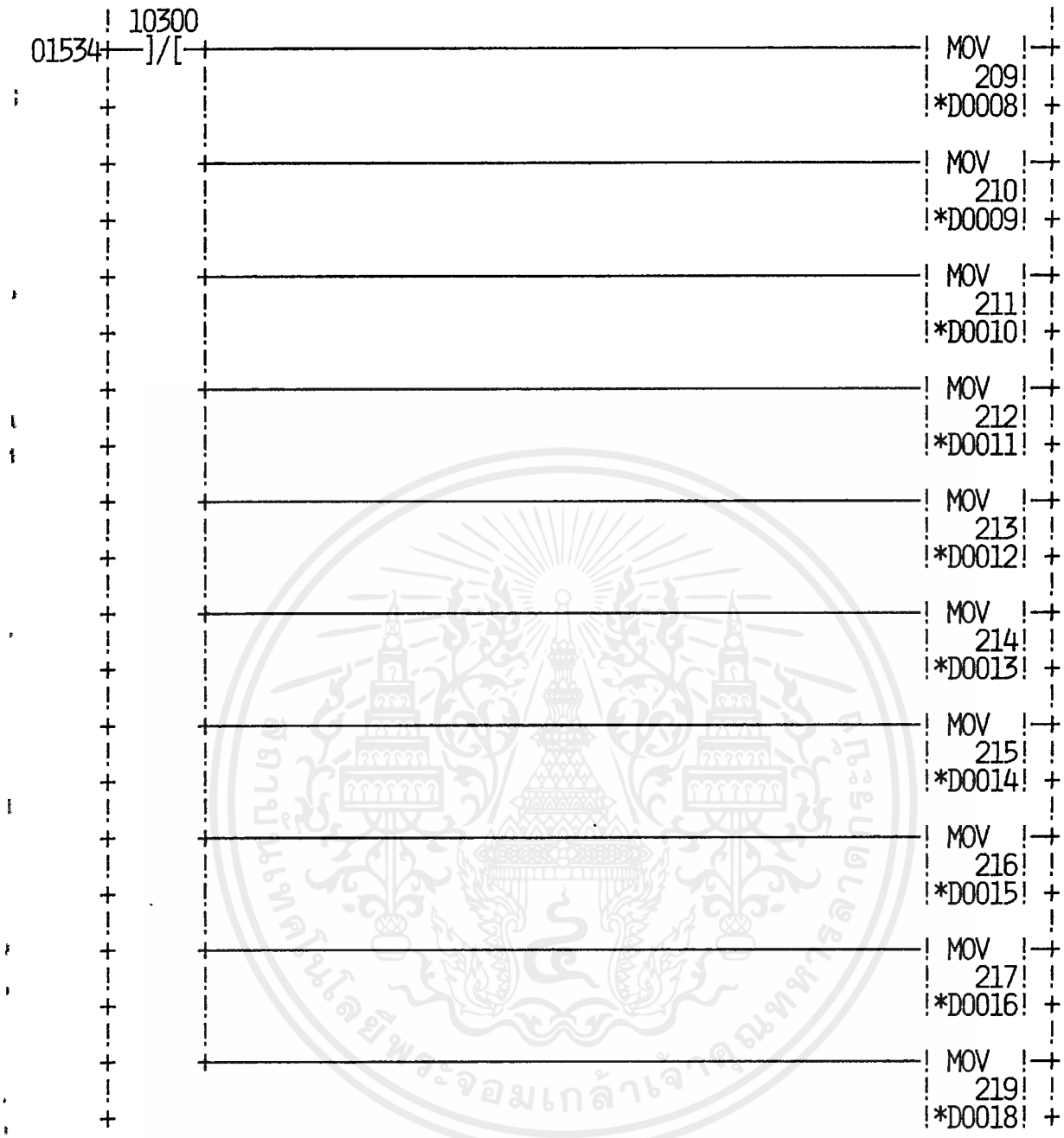
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



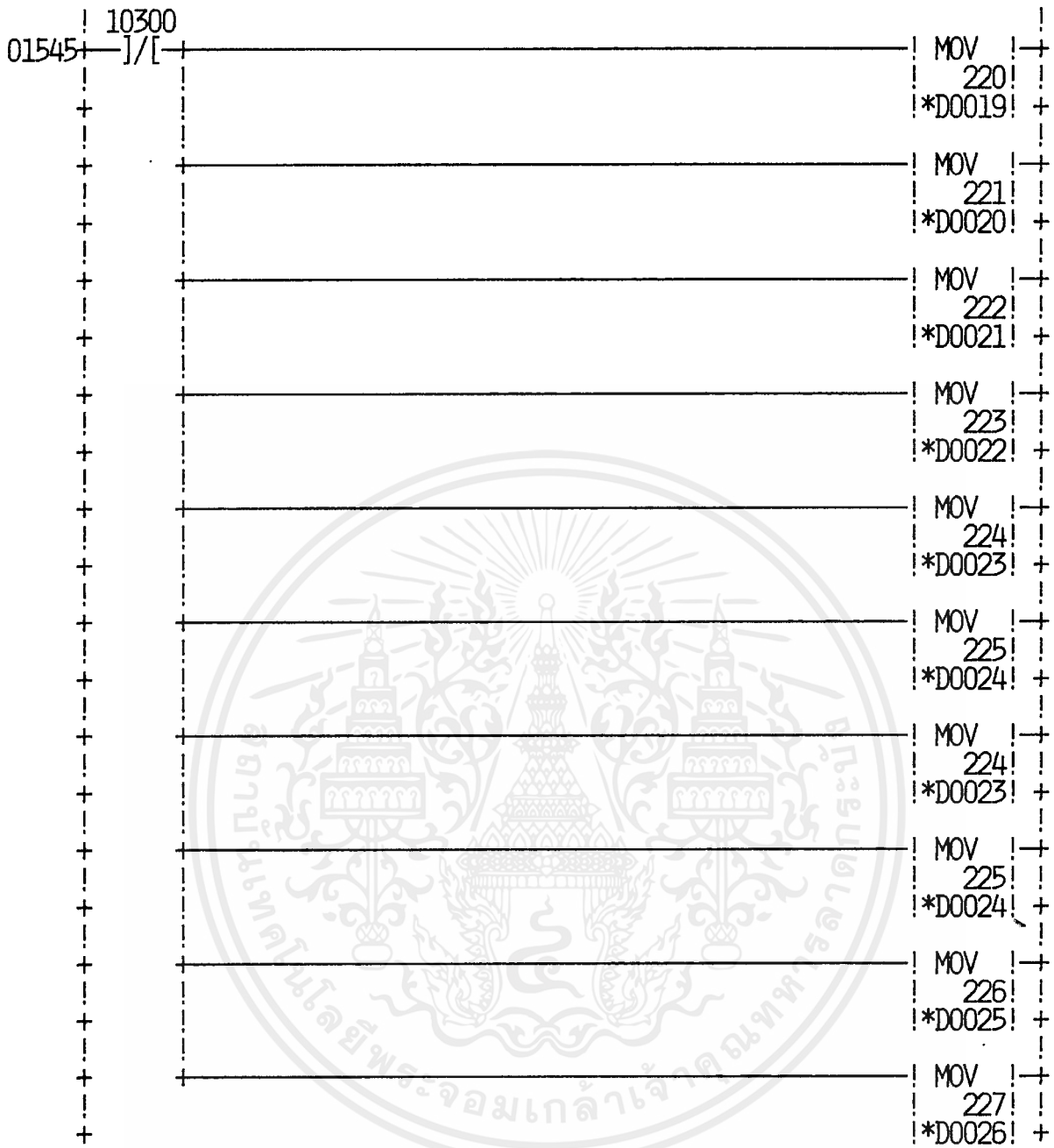
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



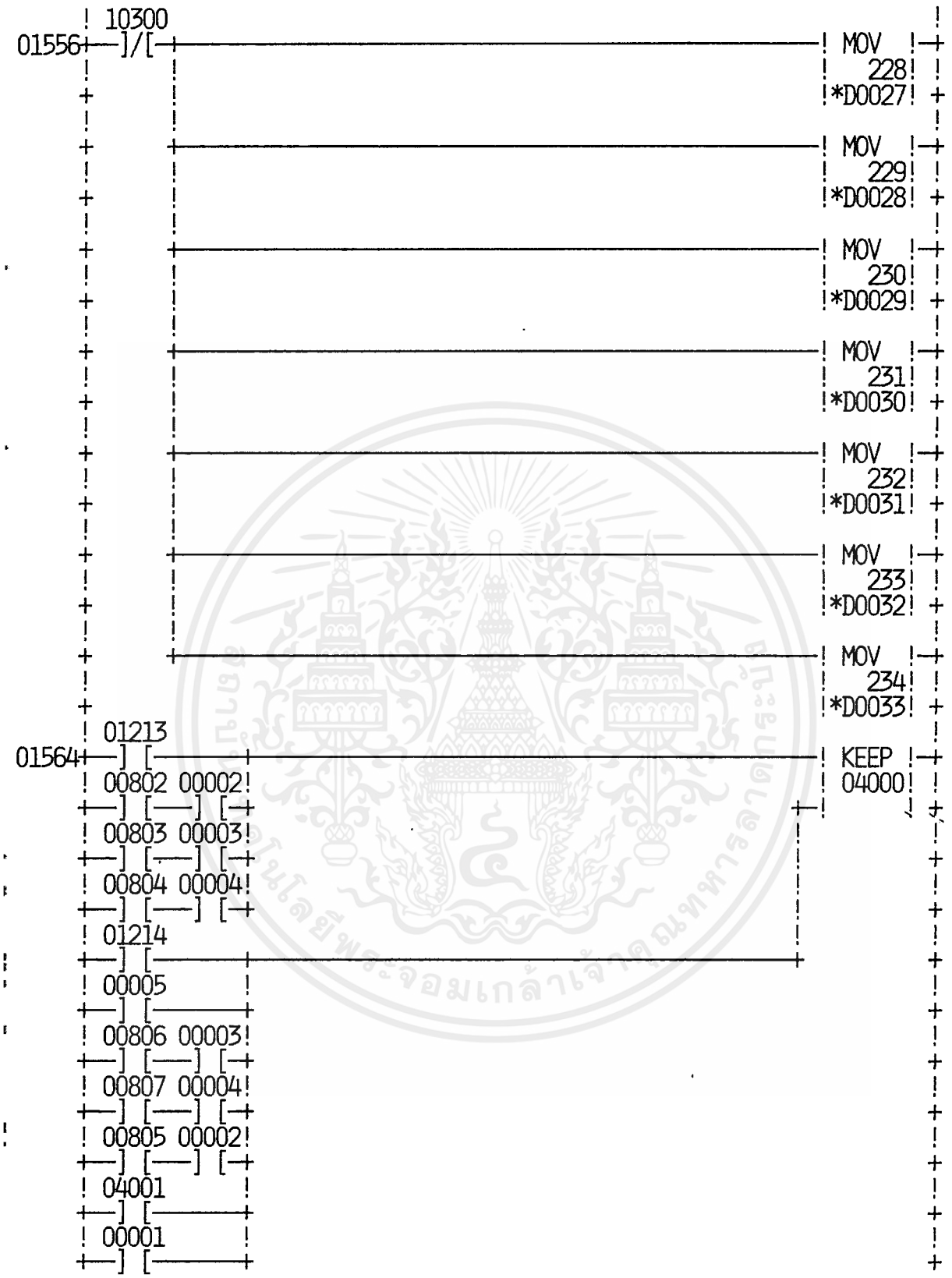
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



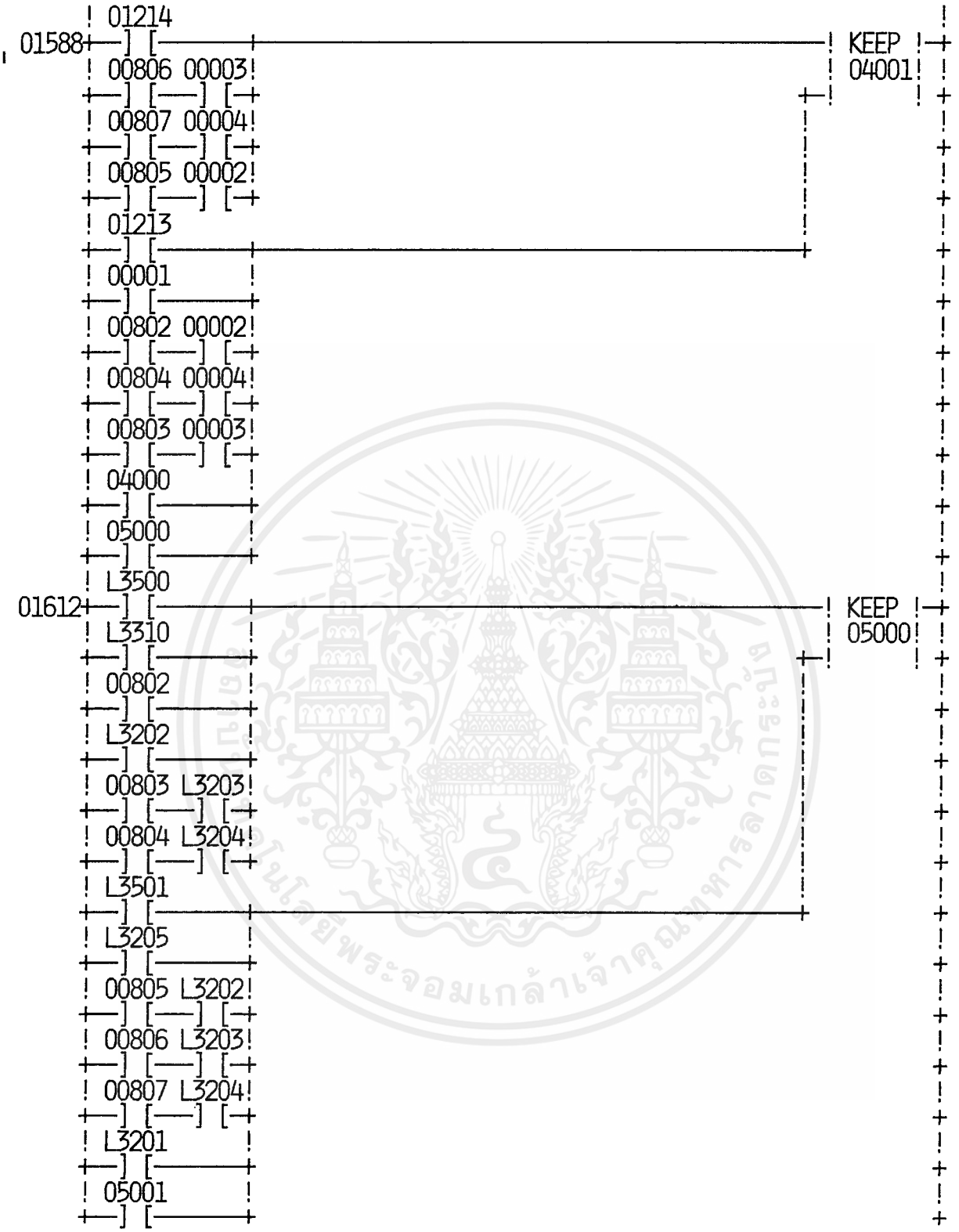
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



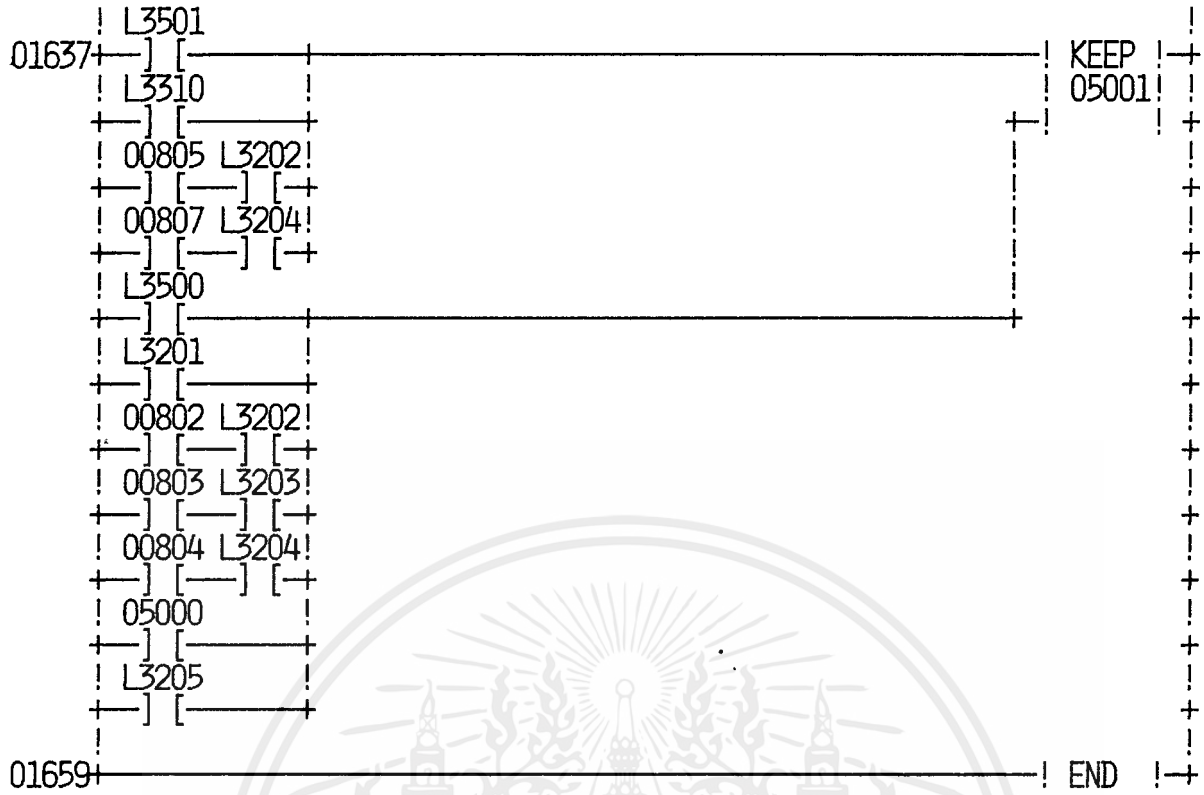
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

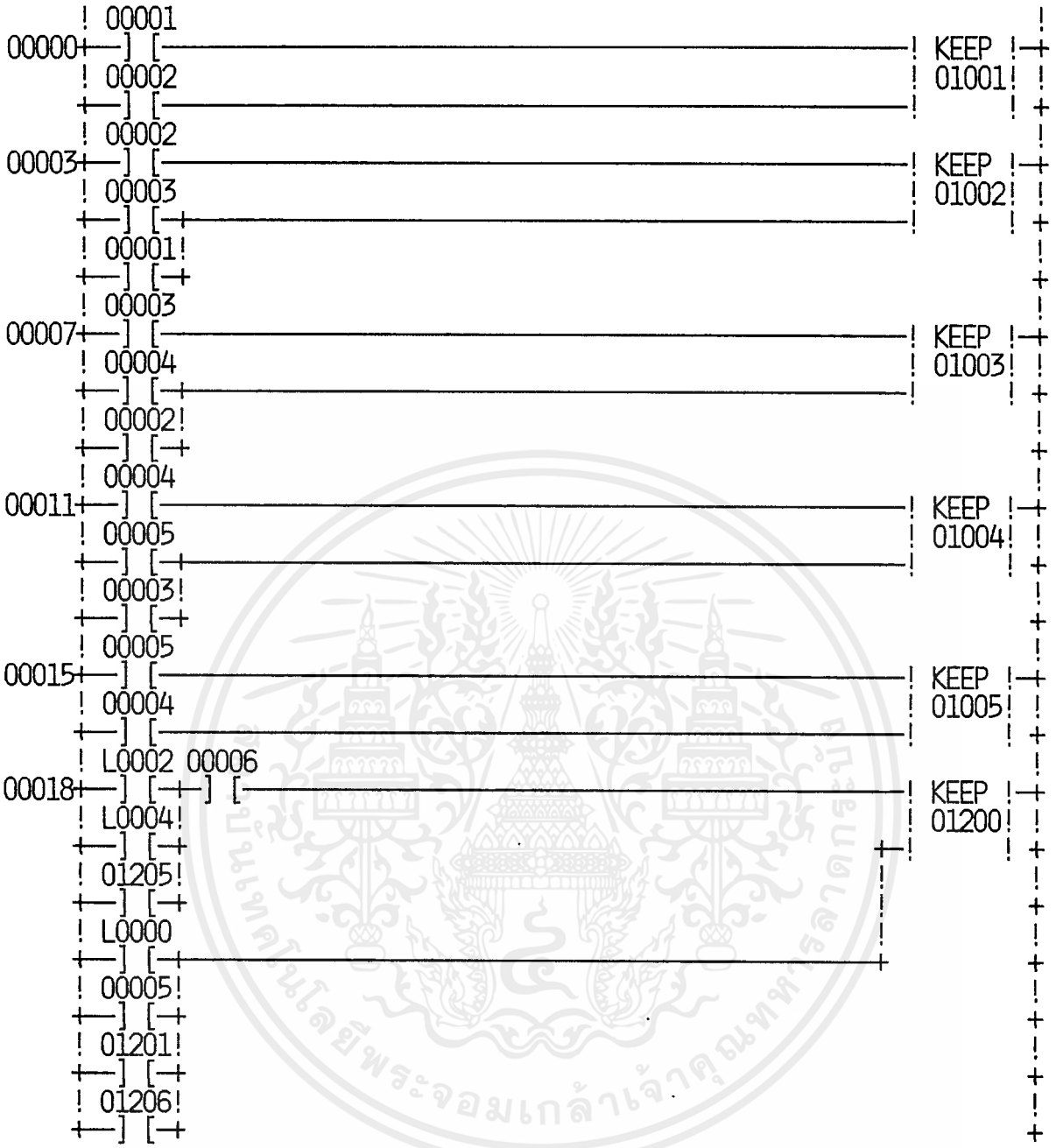


END

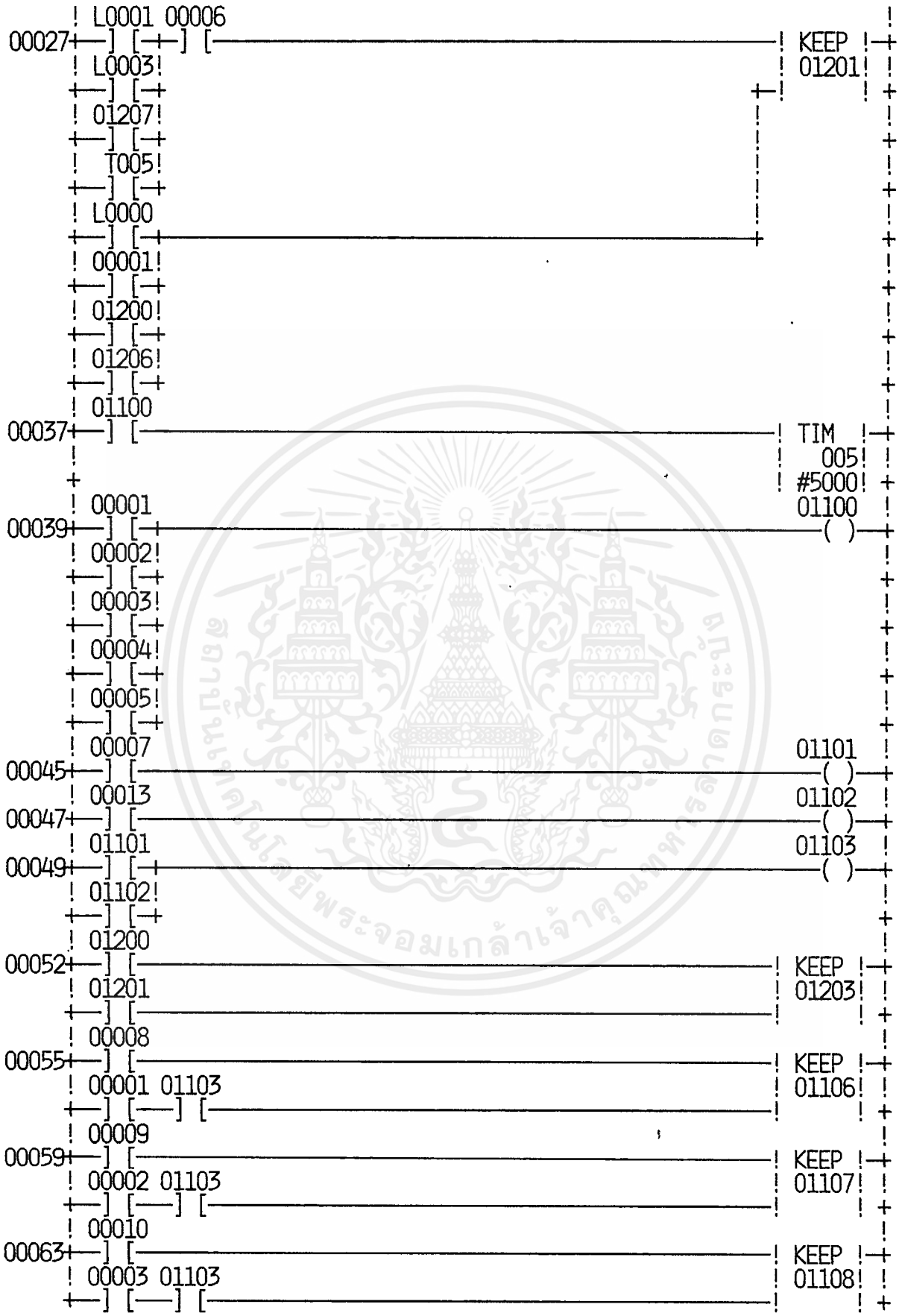
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU UNIT # 1 (SLAVE)

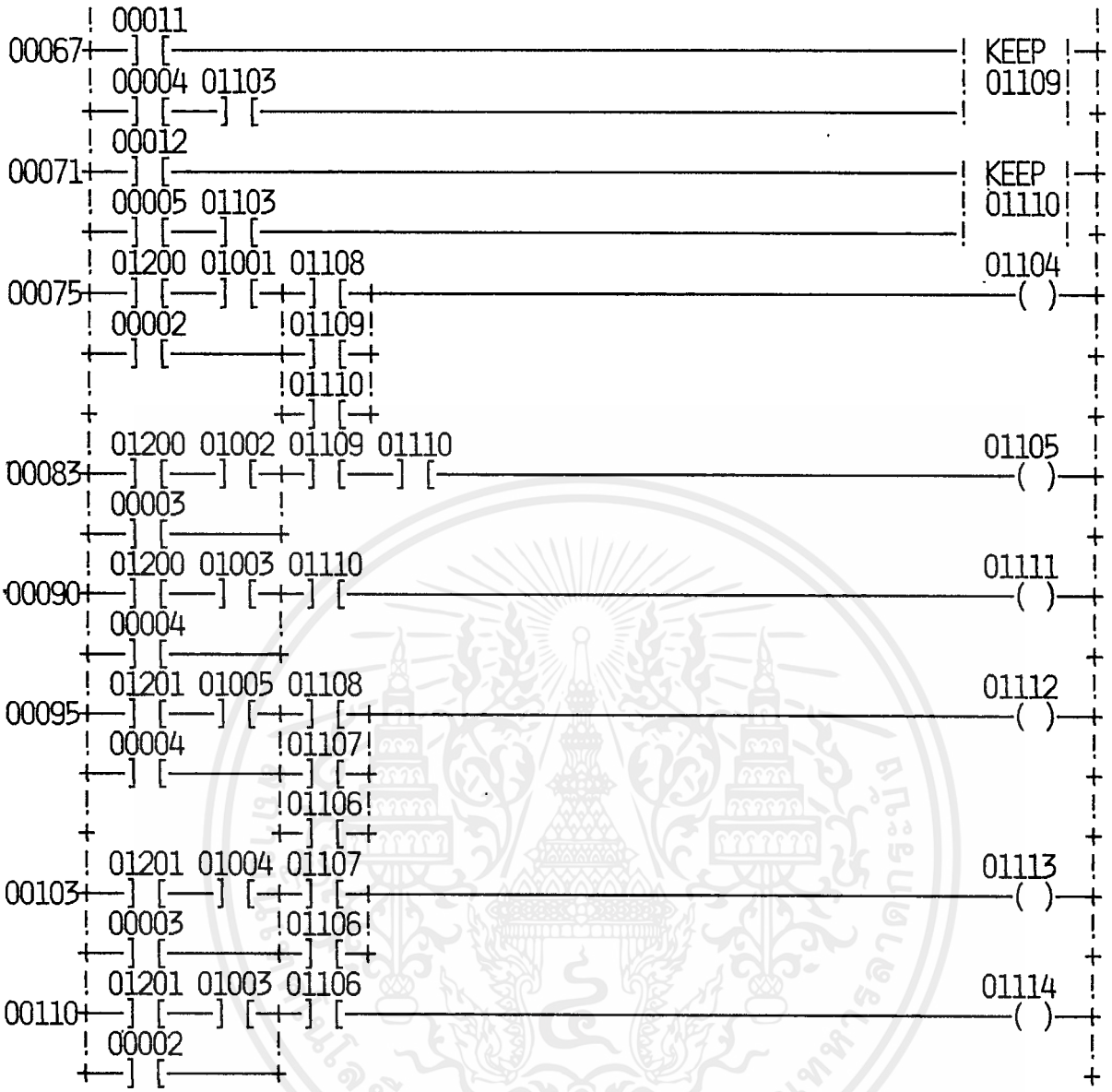
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



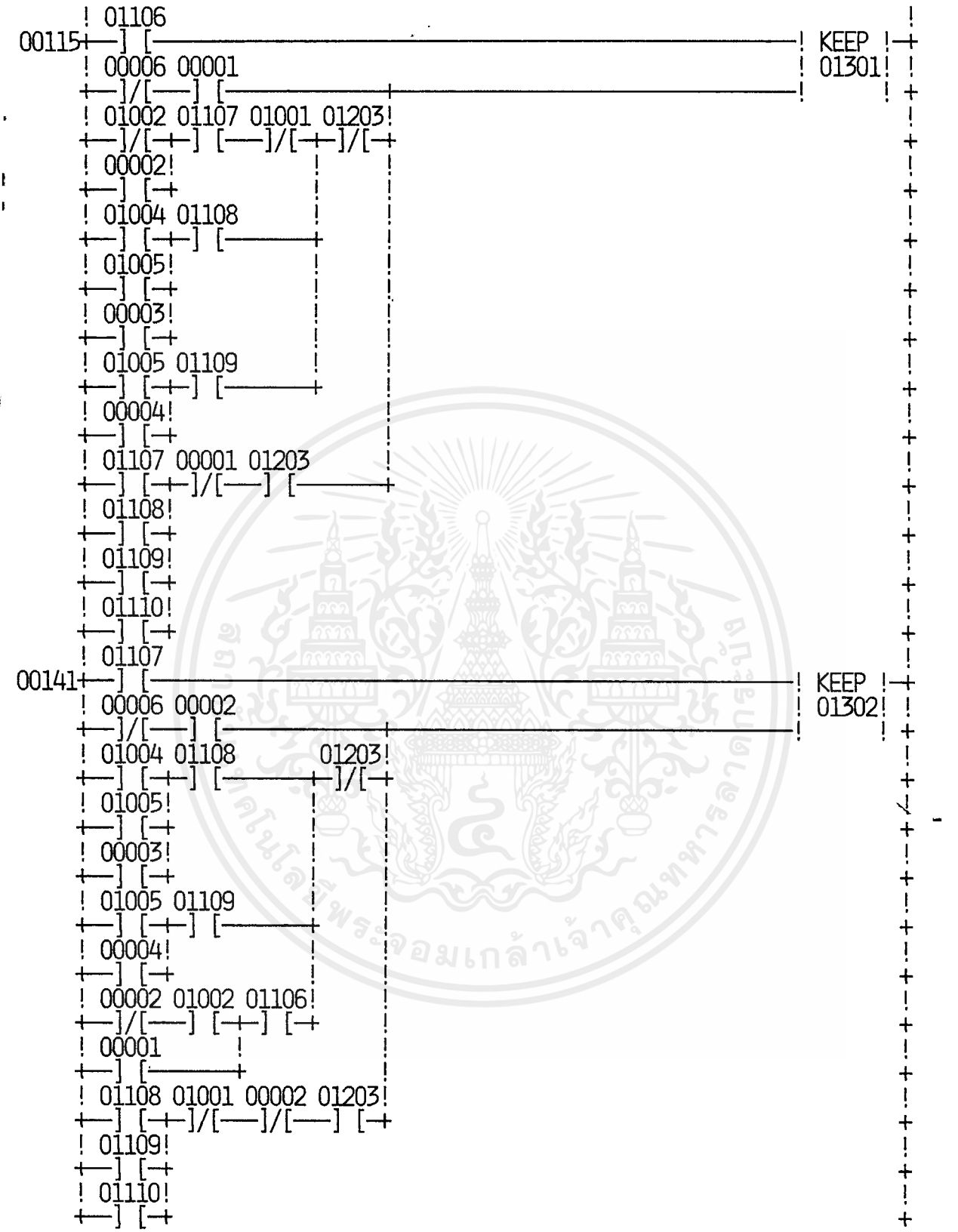
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



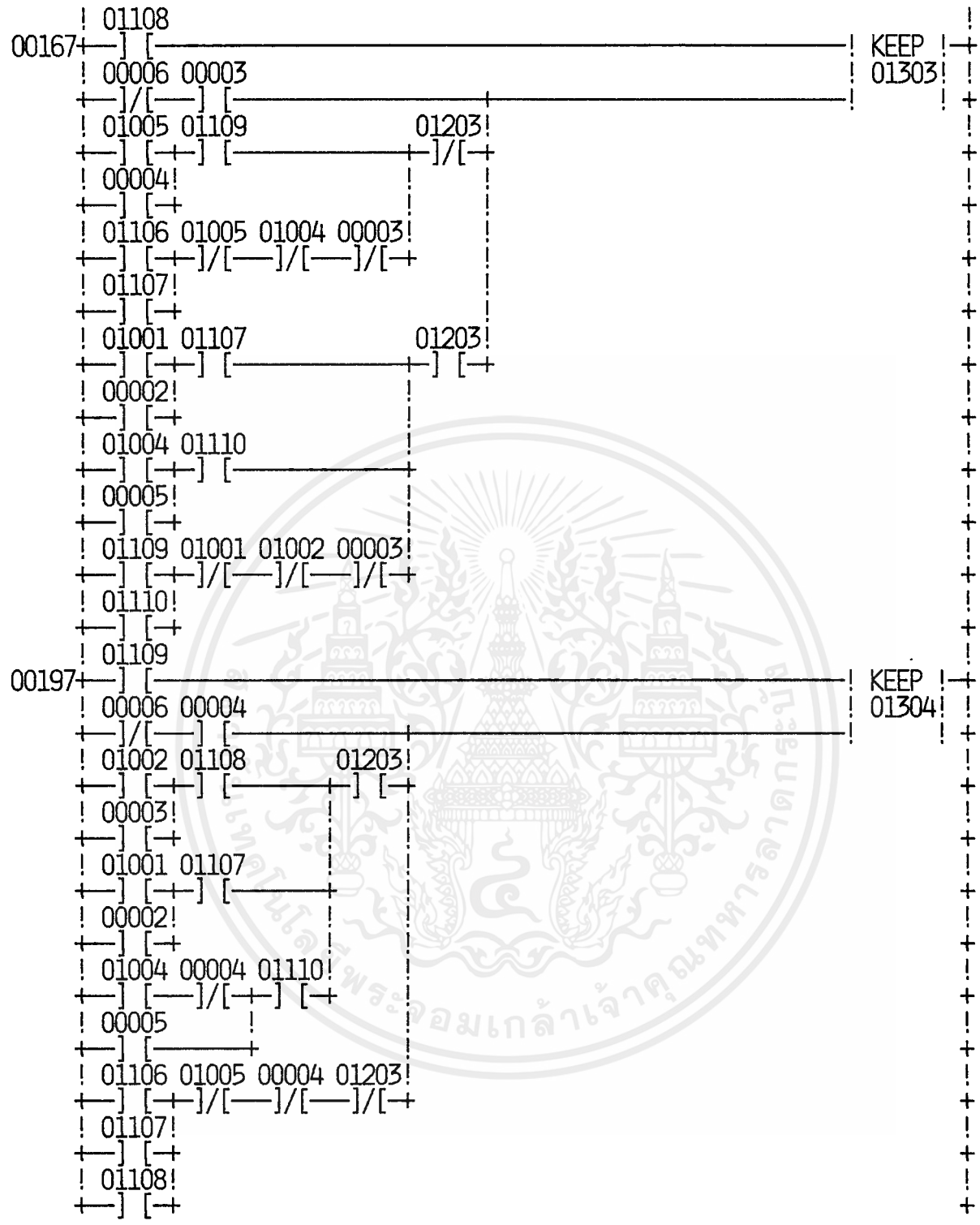
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



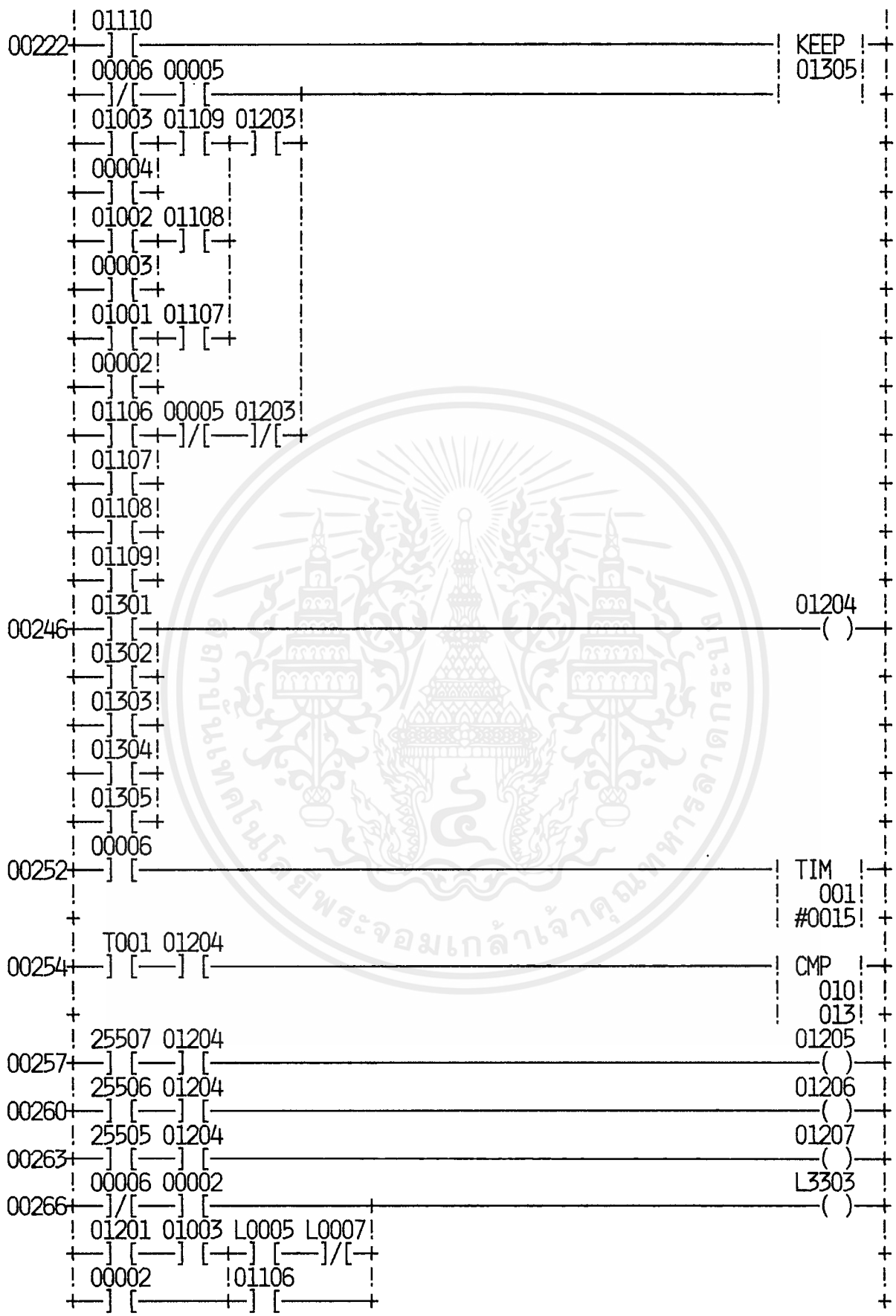
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



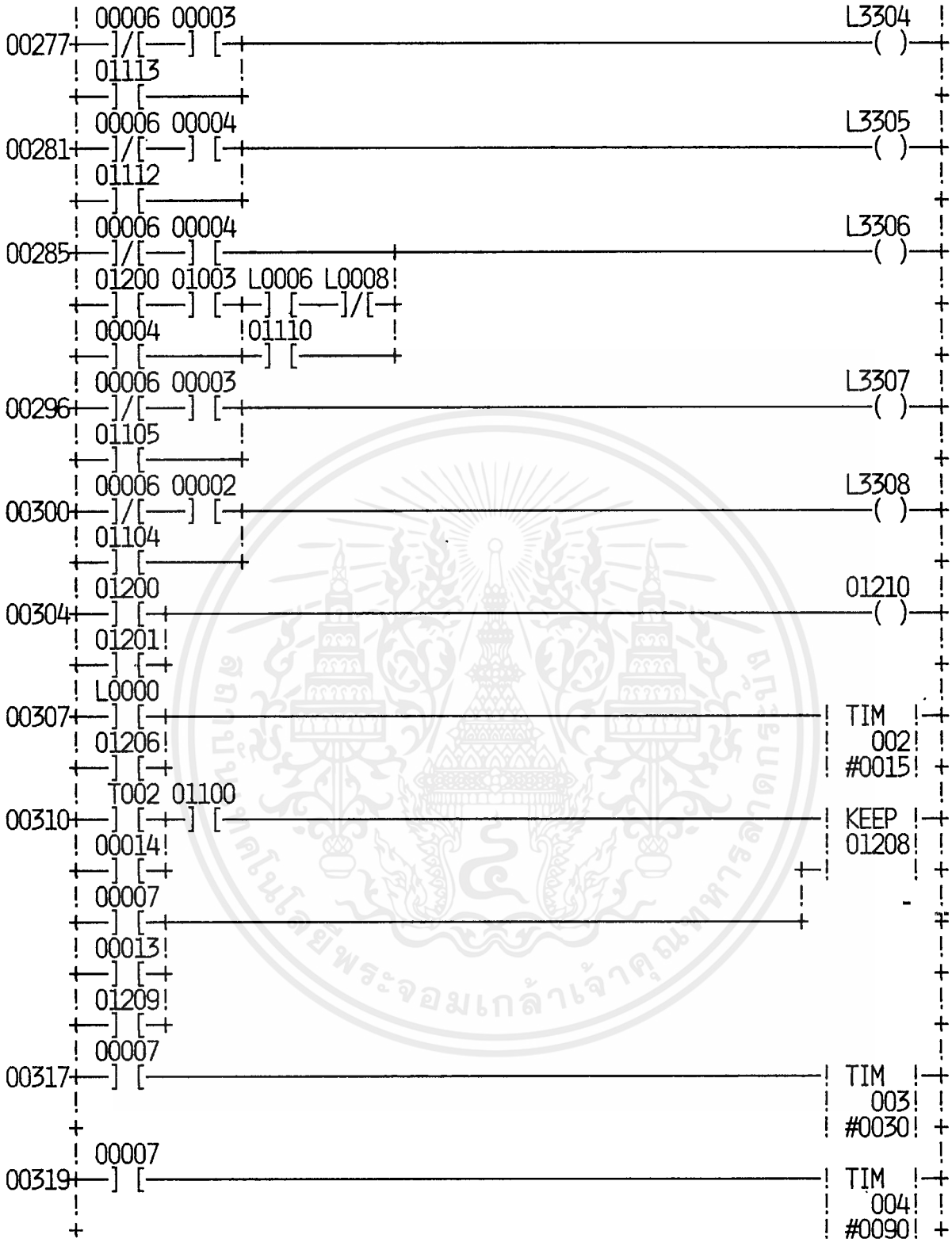
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



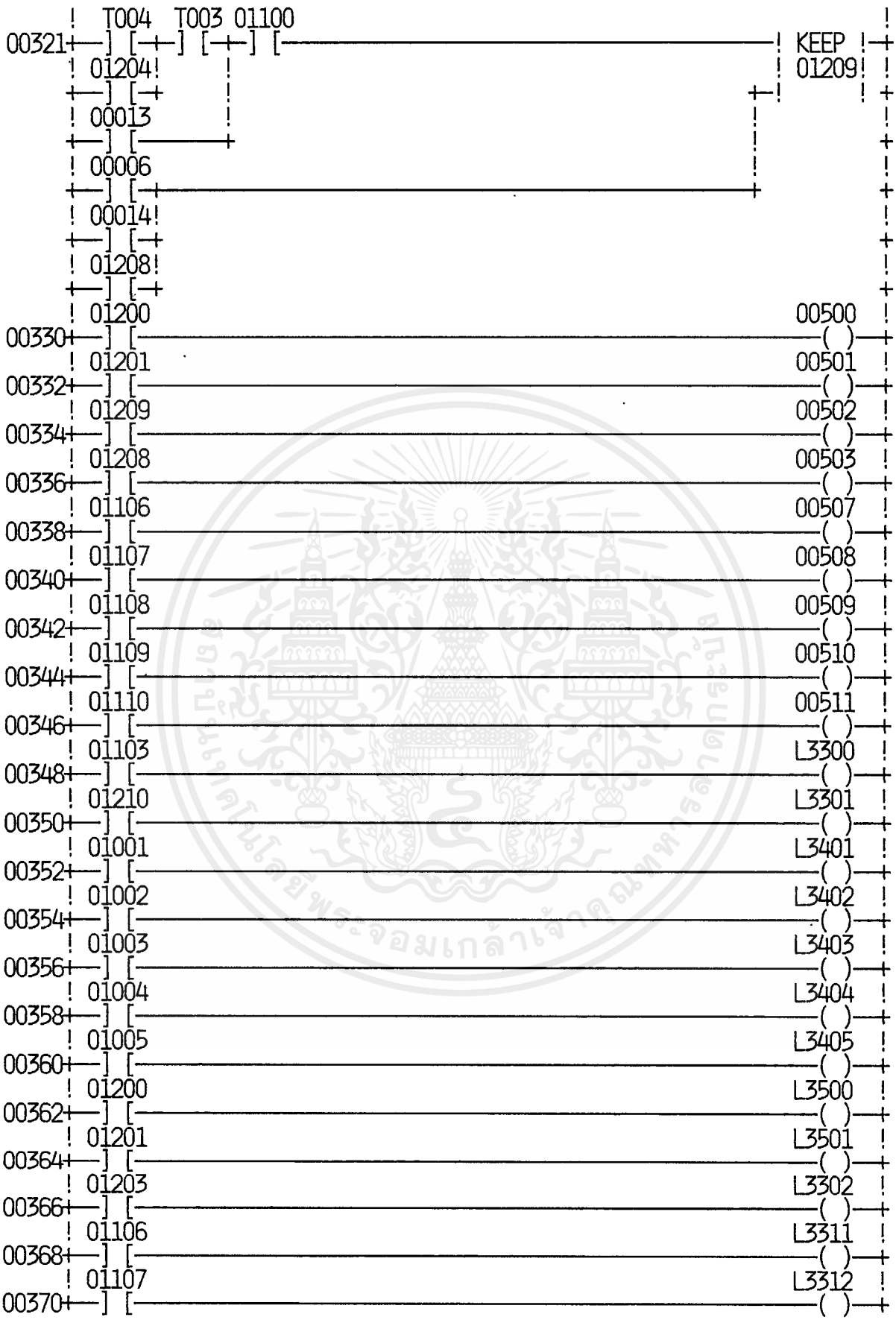
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



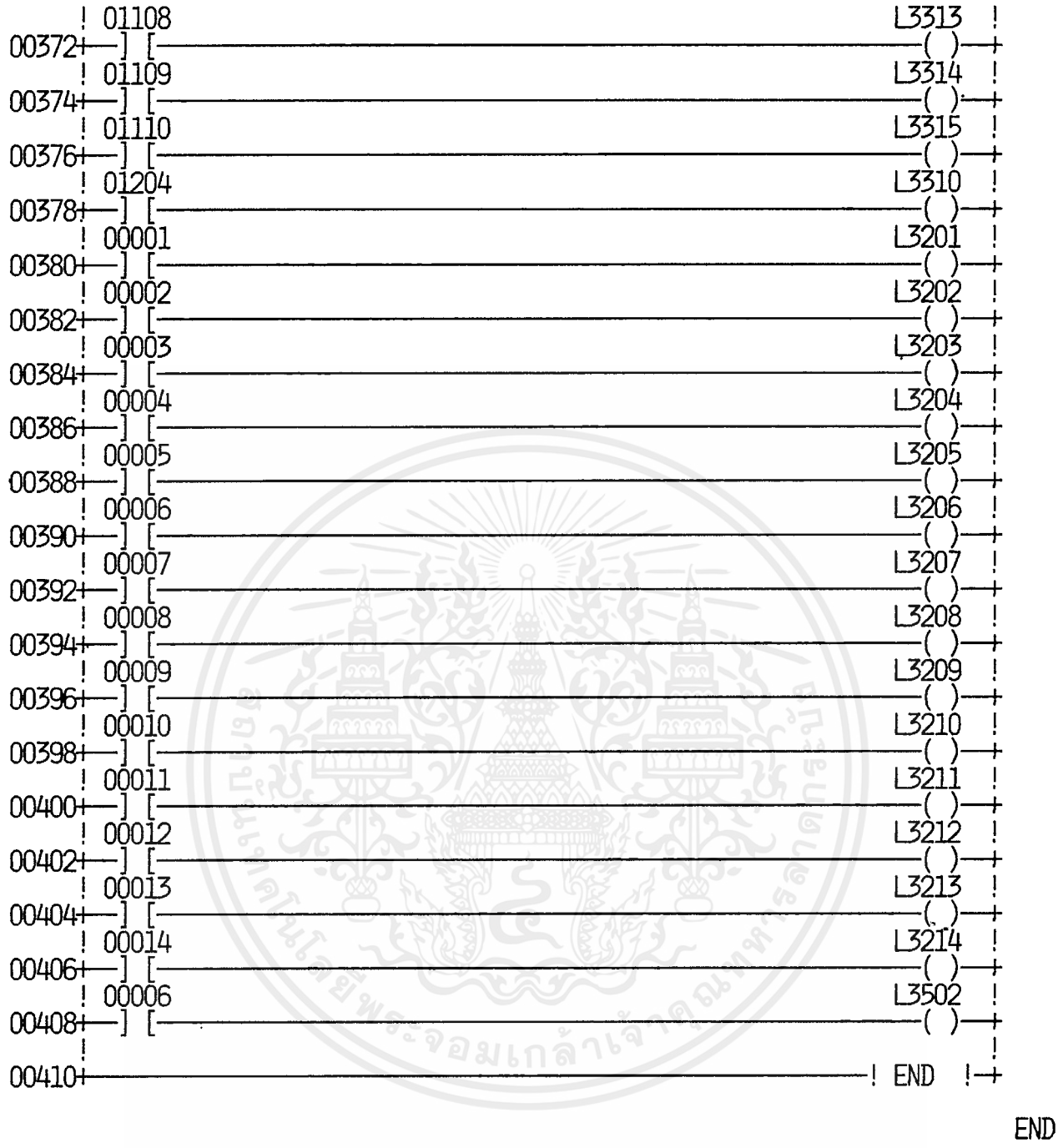
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก 2

โปรแกรมแสดงภาพการทำงานของระบบทาง COMPUTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// THIS PROGRAM USE BORLANDC C++
//   VERSION 3.0  FOR DOS
// COPYRIGHT(C) 1990,1991 BY
// BORLAND INTERNATIONAL,INC.
//          BY
// Mr. ADISAK   CHENGKHEM
// Mr. SANOR    PUDKLIANG
// Mr. BANCHAR  NIYOMYART
//          KMITL

#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
#include <graphics.h>
#include <alloc.h>
#include <conio.h>
#include <bios.h>
#define COM2 1
#define DATA_READY 0x100
#define TRUE 1
#define FALSE 0
#define SETTINGS (_COM_9600|_COM_CHR8|_COM_STOP2|_COM_NOPARITY)


#define LEFT    0x4B00
#define RIGHT   0X4D00
#define ESCAPE  0X111B
#define ENTER   0X1C0D

int Read_Key(int key);
void key(int x,int y,char ch[],int color);
void select(int x,int y,char ch[],int color);
void level_5(int x,int y,int color);
void level_4_DOWN(int x,int y,int color);
void level_4_UP(int x,int y,int color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void level_3_DOWN(int x,int y,int color);
void level_3_UP(int x,int y,int color);
void level_2_DOWN(int x,int y,int color);
void level_2_UP(int x,int y,int color);
void level_1(int x,int y,int color);

void level_5_1(int x,int y,int color);
void level_4_DOWN_1(int x,int y,int color);
void level_4_UP_1(int x,int y,int color);
void level_3_DOWN_1(int x,int y,int color);
void level_3_UP_1(int x,int y,int color);
void level_2_DOWN_1(int x,int y,int color);
void level_2_UP_1(int x,int y,int color);
void level_1_1(int x,int y,int color);

void level_5_2(int x,int y,int color);
void level_4_DOWN_2(int x,int y,int color);
void level_4_UP_2(int x,int y,int color);
void level_3_DOWN_2(int x,int y,int color);
void level_3_UP_2(int x,int y,int color);
void level_2_DOWN_2(int x,int y,int color);
void level_2_UP_2(int x,int y,int color);
void level_1_2(int x,int y,int color);

void UP(int x,int y,int color);
void DOWN(int x,int y,int color);
void OPEN(int x,int y);
void CLOSE(int x,int y);
void OPEN_1(int x,int y);

void UP_2(int x,int y,int color);
void DOWN_2(int x,int y,int color);
void OPEN_2(int x,int y);
void CLOSE_2(int x,int y);
void OPEN_1_2(int x,int y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void comreceive(void);
void comreceive_1(void);
void comreceive_2(void);

void FIVE_1(int color);
void FOUR_1(int color);
void THREE_1(int color);
void TWO_1(int color);
void ONE_1(int color);

void FIVE_2(int color);
void FOUR_2(int color);
void THREE_2(int color);
void TWO_2(int color);
void ONE_2(int color);

void FIVE_1_1(int color);
void FOUR_1_1(int color);
void THREE_1_1(int color);
void TWO_1_1(int color);
void ONE_1_1(int color);

void FIVE_1_2(int color);
void FOUR_1_2(int color);
void THREE_1_2(int color);
void TWO_1_2(int color);
void ONE_1_2(int color);

void POS1_2(int color);
void POS2_2(int color);
void POS3_2(int color);
void POS4_2(int color);
void POS5_2(int color);

void POS1(int color);
void POS2(int color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void POS3(int color);
void POS4(int color);
void POS5(int color);

void SELECT1_1(int color);
void SELECT2_1(int color);
void SELECT3_1(int color);
void SELECT4_1(int color);
void SELECT5_1(int color);
void SELECT1_2(int color);
void SELECT2_2(int color);
void SELECT3_2(int color);
void SELECT4_2(int color);
void SELECT5_2(int color);

void DOPEN_1(int color);
void DOPEN_2(int color);
void DCLOSE_1(int color);
void DCLOSE_2(int color);
void DCLOSE_3(int color);
void DCLOSE_4(int color);
char  *MEN_[] = {"     VIEW LIFT 1 ","      VIEW LIFT 2 ",
                 "          EXIT "};
int xm,ym;
int pos=0;
void MENU(void);
void VIEW_LIFT1(void);
void VIEW_LIFT2(void);
void VIEW_LIFT3(void);
void initial_menu(void);

main()
{
  int ErrorCode,GraphDriver,GraphMode;
 GraphDriver = DETECT;
 initgraph(&GraphDriver,&GraphMode,"");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 ErrorCode = graphresult();
 if(ErrorCode != grOk)
 {
  printf("Graphics System Error: %s\n",grapherrormsg(ErrorCode));
    exit(1);
 }
     MENU();
   closegraph();
   return 0;
}
/*************************************************************/
     void MENU(void)
 { int i,KEY;
   initial_menu();
   do
 {
   KEY=Read_Key(0);
   switch(KEY) {


case LEFT       :      select(20+pos*215,70,MEN_[pos],BLACK);
                              pos=(pos==0) ? 2 : --pos;
                              select(20+pos*215,70,MEN_[pos],2);
                              break;


case RIGHT      :      select(20+pos*215,70,MEN_[pos],BLACK);
                              pos=(pos==2) ? 0 : ++pos;
                              select(20+pos*215,70,MEN_[pos],2);
                              break;
case ENTER      :      if(pos==0) {VIEW_LIFT1();
                              initial_menu(); } else;
                     if(pos==1) {VIEW_LIFT2();
                              initial_menu(); } else;
                              break;


 default   :      break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        }
    } while(!((pos==2)&&(KEY==ENTER)));
   }

   void select(int x,int y,char ch[],int color)
   {
   setfillstyle(1,color);
   bar(x-9,y-15,x+180,y+5);
   setcolor(WHITE);
   settextstyle(1,0,1);
   outtextxy(x-14,y-14,ch);
   }
   void initial_menu(void)
   { int i; pos=0;
   xm=getmaxx(); ym=getmaxy();
   setfillstyle(1,RED);
   bar(0,0,xm,ym);
   setfillstyle(1,BLUE);
   bar(5,5,xm-5,ym-5);
   setfillstyle(1,RED);
   bar(5,45,xm-5,40);
   setfillstyle(1,RED);
   bar(5,88,xm-5,83);
   setcolor(YELLOW);
   settextstyle(1,0,1);
outtextxy(58,10,"NETWORK CONTROL SYSTEM APPLICATION FOR MULTI-LIFT");
   setfillstyle(1,2);
   bar(10,55,190,75);
   for(i=1;i<=2;i++) {
   setfillstyle(1,BLACK);
   bar(10+i*218,55,190+i*218,75);  }
   for(i=0;i<=2;i++)
   {
   setcolor(WHITE);
   outtextxy((20+i*190)-14,70-14,MEN_[i]);
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 setfillstyle(1,RED);
 bar(45,152,184,395);   // blackground red color
 bar(445,152,584,395);  //
 setfillstyle(1,2);
 bar(59,166,101,381);   // blackground green color
 bar(106,166,170,381);  //
 bar(459,166,501,381);  //
 bar(506,166,570,381);  //
 setfillstyle(1,7);
 bar(111,180,165,214);  bar(511,180,565,214);
 bar(111,219,165,253);  bar(511,219,565,253);
 bar(111,258,165,292);  bar(511,258,565,292);
 bar(111,297,165,331);  bar(511,297,565,331);
 bar(111,336,165,370);  bar(511,336,565,370);

 bar(62,188,80,206); //  bar(462,188,480,206);
 bar(62,227,80,245); //  bar(462,227,480,245);
 bar(62,269,80,286); //  bar(462,269,480,286);
 bar(62,307,80,325); //  bar(462,307,480,325);
 bar(62,346,80,364); //  bar(462,346,480,364);

 for(i=0;i<=0;i++)
 {
 level_5(71+(i*200),194,8);
 level_4_UP(71+(i*200),235,8);
 level_4_DOWN(71+(i*200),237,8);
 level_3_UP(71+(200*i),277,8);
 level_3_DOWN(71+(200*i),279,8);
 level_2_UP(71+(i*200),315,8);
 level_2_DOWN(71+(i*200),317,8);
 level_1(71+(i*200),358,8);
 }
 comreceive();
 }
 void FIVE_1(int color)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
setfillstyle(1,color);
bar(111,180,165,214);
}
void FOUR_1(int color)
{
setfillstyle(1,color);
bar(111,219,165,253);
}
void THREE_1(int color)
{
setfillstyle(1,color);
bar(111,258,165,292);
}
void TWO_1(int color)
{
setfillstyle(1,color);
bar(111,297,165,331);
}
void ONE_1(int color)
{
setfillstyle(1,color);
bar(111,336,165,370);
}
void FIVE_2(int color)
{
setfillstyle(1,color);
bar(511,180,565,214);
}
void FOUR_2(int color)
{
setfillstyle(1,color);
bar(511,219,565,253);
}
void THREE_2(int color)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setfillstyle(1,color);
bar(511,258,565,292);
}
void TWO_2(int color)
{
setfillstyle(1,color);
bar(511,297,565,331);
}
void ONE_2(int color)
{
setfillstyle(1,color);
bar(511,336,565,370);
}

void comreceive(void)
{
unsigned out;
unsigned in,status;
_bios_serialcom(_COM_INIT,COM2,SETTINGS);
for(;;)
{
status = _bios_serialcom(_COM_STATUS,COM2,0);
if( status & DATA_READY)
if((out = _bios_serialcom(_COM_RECEIVE,COM2,0)&0x7F) != 0)
switch(out)
{
 case  65   :    ONE_1(7);
                    break;
 case  66   :    ONE_1(14);
                    break;
 case  67   :    TWO_1(7);
                    break;
 case  68   :    TWO_1(14);
                    break;
 case  69   :    THREE_1(7);
                    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 70  :  THREE_1(14);
            break;
case 71  :  FOUR_1(7);
            break;
case 72  :  FOUR_1(14);
            break;
case 73  :  FIVE_1(7);
            break;
case 74  :  FIVE_1(14);
            break;
case 89  :  ONE_2(7);
            break;
case 90  :  ONE_2(14);
            break;
case 97  :  TWO_2(7);
            break;
case 98  :  TWO_2(14);
            break;
case 99  :  THREE_2(7);
            break;
case 100 :  THREE_2(14);
            break;
case 101 :  FOUR_2(7);
            break;
case 102 :  FOUR_2(14);
            break;
case 103 :  FIVE_2(7);
            break;
case 104 :  FIVE_2(14);
            break;
case 119 :  level_5(71,194,8);
            break;
case 120 :  level_5(71,194,14);
            break;
case 121 :  level_4_UP(71,235,8);
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  case  122   :    level_4_UP(71,235,14);
                    break;
  case  48    :    level_4_DOWN(71,237,8);
                    break;
  case  49    :    level_4_DOWN(71,237,14);
                    break;
  case  50    :    level_3_UP(71,277,8);
                    break;
  case  51    :    level_3_UP(71,277,14);
                    break;
  case  52    :    level_3_DOWN(71,279,8);
                    break;
  case  53    :    level_3_DOWN(71,279,14);
                    break;
  case  54    :    level_2_UP(71,315,8);
                    break;
  case  55    :    level_2_UP(71,315,14);
                    break;
  case  56    :    level_2_DOWN(71,317,8);
                    break;
  case  57    :    level_2_DOWN(71,317,14);
                    break;
  case  58    :    level_1(71,358,8);
                    break;
  case  59    :    level_1(71,358,14);
                    break;
   default    :   break;
 }

 if(kbhit())
{
if((in = getch()) == '\x0D')
break;
  }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

 void level_5(int x,int y,int color)
{
 struct PTS{
 int x,y;  };
 struct PTS poly[5];
 setfillstyle(SOLID_FILL,color);
 setcolor(7);
 poly[0].x=x+9;
 poly[0].y=y;
 poly[1].x=x-9;
 poly[1].y=y;
 poly[2].x=x;
 poly[2].y=y+9;
 fillpoly(3,(int far *)poly);
 }
 void level_4_UP(int x,int y,int color)
 {
 struct PTS{
 int x,y;  };
 struct PTS poly[5];
 setfillstyle(SOLID_FILL,color);
 setcolor(7);
 poly[0].x=x-9;
 poly[0].y=y;
 poly[1].x=x+9;
 poly[1].y=y;
 poly[2].x=x;
 poly[2].y=y-9;
 fillpoly(3,(int far *)poly);
  }
 void level_4_DOWN(int x,int y,int color)
 {
  struct PTS{
  int x,y;  };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  struct PTS poly[5];
  setfillstyle(SOLID_FILL,color);
  setcolor(7);
  poly[0].x=x+9;
  poly[0].y=y;
  poly[1].x=x-9;
  poly[1].y=y;
  poly[2].x=x;
  poly[2].y=y+9;
  fillpoly(3,(int far *)poly);
  }
   void level_3_UP(int x,int y,int color)
 {
 struct PTS{
 int x,y;  };
 struct PTS poly[5];
 setfillstyle(SOLID_FILL,color);
 setcolor(7);
 poly[0].x=x-9;
 poly[0].y=y;
 poly[1].x=x+9;
 poly[1].y=y;
 poly[2].x=x;
 poly[2].y=y-9;
 fillpoly(3,(int far *)poly);
 }
 void level_3_DOWN(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(SOLID_FILL,color);
poly[0].x=x+9;
poly[0].y=y;
poly[1].x=x-9;
poly[1].y=y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 poly[2].x=x;
 poly[2].y=y+9;
 fillpoly(3,(int far *)poly);
 }
 void level_2_UP(int x,int y,int color)
 {
 struct PTS{
 int x,y;  };
 struct PTS poly[5];
 setfillstyle(SOLID_FILL,color);
 poly[0].x=x-9;
 poly[0].y=y;
 poly[1].x=x+9;
 poly[1].y=y;
 poly[2].x=x;
 poly[2].y=y-9;
 fillpoly(3,(int far *)poly);
 }
 void level_2_DOWN(int x,int y,int color)
 {
 struct PTS{
 int x,y;  };
 struct PTS poly[5];
 setfillstyle(SOLID_FILL,color);
 poly[0].x=x+9;
 poly[0].y=y;
 poly[1].x=x-9;
 poly[1].y=y;
 poly[2].x=x;
 poly[2].y=y+9;
 fillpoly(3,(int far *)poly);
 }
 void level_1(int x,int y,int color)
 {
 struct PTS{
 int x,y;  };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 struct PTS poly[5];
 setfillstyle(SOLID_FILL,color);
 poly[0].x=x-9;
 poly[0].y=y;
 poly[1].x=x+9;
 poly[1].y=y;
 poly[2].x=x;
 poly[2].y=y-9;
 fillpoly(3,(int far *)poly);
  }

 int Read_Key(int key)
 { return bioskey(key); }

/*****************************************************************/

 void VIEW_LIFT1(void)
 {
  int i;
 xm = getmaxx(); ym = getmaxy();
  setfillstyle(1,8);
  bar(0,0,xm,ym);
  setfillstyle(1,1);
  bar(40,38,xm-40,ym-38);
  settextstyle(1,0,2);
  setcolor(WHITE);
  outtextxy(250,7,"VIEW LIFT NUMBER 1");
  setfillstyle(1,RED);  /* background  lift no 1 */
  bar(70,73,300,407);
  bar(350,73,570,407);
  setfillstyle(1,2);
  bar(90,93,200,387);
  bar(205,93,280,387);
  setfillstyle(1,BLACK);
  bar(350,123,570,133);
  bar(365,153,480,387);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setfillstyle(1,7);
bar(370,158,475,382);
setfillstyle(1,BLACK);
bar(370,178,475,183);
bar(419,178,424,382);
setfillstyle(1,7);
bar(485,200,565,345);
setfillstyle(1,8);
bar(495,210,555,235);     /* LIFT UP & DOWN  */
bar(485,245,565,345);
setfillstyle(1,BLACK);
bar(492,255,510,280);     /*  number 1 */
bar(492,285,510,310);     /*  number 4 */
bar(492,315,517,333);     /*  door open */
bar(516,255,534,280);     /*  number 2 */
bar(534,315,558,333);     /*  door close */
bar(540,255,558,280);     /*  number 3  */
bar(540,285,558,310);     /*  number 5  */
setfillstyle(1,BLACK);
bar(400,83,520,113);
setfillstyle(1,RED);
bar(420,83,425,113);
bar(445,83,450,113);
bar(470,83,475,113);
bar(495,83,500,113);
setcolor(7);
settextstyle(1,0,1);
outtextxy(407,85,"1");   /* top room */
outtextxy(430,85,"2");
outtextxy(456,85,"3");
outtextxy(482,85,"4");
outtextxy(507,85,"5");
outtextxy(497,255,"1");
outtextxy(521,255,"2");
outtextxy(545,255,"3");
outtextxy(497,285,"4");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 outtextxy(545,285,"5");
   UP(510,228,7);
   DOWN(540,219,7);
   CLOSE(546,324);
   OPEN(516,324);
   OPEN_1(492,324);
 setfillstyle(1,7);
 bar(102,105,188,155);   /* fill level 5 */
 bar(102,160,188,210);   /*            4 */
 bar(102,215,188,265);   /*            3 */
 bar(102,270,188,315);   /*            2 */
 bar(102,320,188,370);   /*            1 */
/****END OF LEVEL****/
 bar(245,115,265,145);
 bar(245,170,265,200);
 bar(245,225,265,255);
 bar(245,280,265,310);
 bar(245,330,265,360);
/****END OF BOTTOM LEVEL****/
 level_1_1(255,348,8);
 level_2_DOWN_1(255,297,8);
 level_2_UP_1(255,293,8);
 level_3_DOWN_1(255,242,8);
 level_3_UP_1(255,237,8);
 level_4_DOWN_1(255,188,8);
 level_4_UP_1(255,182,8);
 level_5_1(255,127,8);
 comreceive_1();
 }
  void comreceive_1(void)
 {
 unsigned out;
 unsigned in,status;
 _bios_serialcom(_COM_INIT,COM2,SETTINGS);
 for(;;)
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
status = _bios_serialcom(_COM_STATUS,COM2,0);
if( status & DATA_READY)
if((out = _bios_serialcom(_COM_RECEIVE,COM2,0)&0x7F) != 0)
switch(out)
{
 case 65  :    ONE_1_1(7);
                POS1(7);
                break;
 case 66  :    ONE_1_1(14);
                POS1(14);
                break;
 case 67  :    TWO_1_1(7);
                POS2(7);
                break;
 case 68  :    TWO_1_1(14);
                POS2(14);
                break;
 case 69  :    THREE_1_1(7);
                POS3(7);
                break;
 case 70  :    THREE_1_1(14);
                POS3(14);
                break;
 case 71  :    FOUR_1_1(7);
                POS4(7);
                break;
 case 72  :    FOUR_1_1(14);
                POS4(14);
                break;
 case 73  :    FIVE_1_1(7);
                POS5(7);
                break;
 case 74  :    FIVE_1_1(14);
                POS5(14);
                break;
 case 75  :    SELECT1_1(7);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
case 76  :    SELECT1_1(14);
                    break;
case 77  :    SELECT2_1(7);
                    break;
case 78  :    SELECT2_1(14);
                    break;
case 79  :    SELECT3_1(7);
                    break;
case 80  :    SELECT3_1(14);
                    break;
case 81  :    SELECT4_1(7);
                    break;
case 82  :    SELECT4_1(14);
                    break;
case 83  :    SELECT5_1(7);
                    break;
case 84  :    SELECT5_1(14);
                    break;
case 85  :    UP(510,228,7);
                    break;
case 86  :    UP(510,228,14);
                    break;
case 87  :    DOWN(540,219,7);
                    break;
case 88  :    DOWN(540,219,14);
                    break;
case 119 :    level_5_1(255,127,8);
                    break;
case 120 :    level_5_1(255,127,14);
                    break;
case 121 :    level_4_UP_1(255,182,8);
                    break;
case 122 :    level_4_UP_1(255,182,14);
                    break;
case 48  :    level_4_DOWN_1(255,188,8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
 case 49  :     level_4_DOWN_1(255,188,14);
                    break;
 case 50  :     level_3_UP_1(255,237,8);
                    break;
 case 51  :     level_3_UP_1(255,237,14);
                    break;
 case 52  :     level_3_DOWN_1(255,242,8);
                    break;
 case 53  :     level_3_DOWN_1(255,242,14);
                    break;
 case 54  :     level_2_UP_1(255,293,8);
                    break;
 case 55  :     level_2_UP_1(255,293,14);
                    break;
 case 56  :     level_2_DOWN_1(255,297,8);
                    break;
 case 57  :     level_2_DOWN_1(255,297,14);
                    break;
 case 58  :     level_1_1(255,348,8);
                    break;
 case 59  :     level_1_1(255,348,14);
                    break;
 case 60  :     DOPEN_1(14);
                    break;
 case 61  :     DCLOSE_1(7);
                    DCLOSE_2(16);
 default  :    break;
 }
 if(kbhit())
  {
 if((in = getch()) == '\x0D')
 break;
  }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  getch();
 }

 void FIVE_1_1(int color)
 {
setfillstyle(1,color);
bar(102,105,188,155);
 }
 void FOUR_1_1(int color)
 {
setfillstyle(1,color);
bar(102,160,188,210);
  }
 void THREE_1_1(int color)
 {
setfillstyle(1,color);
bar(102,215,188,265);
  }
 void TWO_1_1(int color)
 {
setfillstyle(1,color);
bar(102,270,188,315);
  }
 void ONE_1_1(int color)
 {
setfillstyle(1,color);
bar(102,320,188,370);
  }
 void SELECT1_1(int color)
 {
setcolor(color);
settextstyle(1,0,1);
outtextxy(497,255,"1");
  }
 void SELECT2_1(int color)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 {
 setcolor(color);
 settextstyle(1,0,1);
 outtextxy(521,255,"2");
  }
 void SELECT3_1(int color)
 {
 setcolor(color);
 settextstyle(1,0,1);
 outtextxy(545,255,"3");
  }
 void SELECT4_1(int color)
 {
 setcolor(color);
 settextstyle(1,0,1);
 outtextxy(497,285,"4");
  }
 void SELECT5_1(int color)
  {
 setcolor(color);
 settextstyle(1,0,1);
 outtextxy(545,285,"5");
  }
 void DOPEN_1(int color)
  {
 setfillstyle(1,color);
 bar(370,183,475,382);
   }
 void DCLOSE_1(int color)
  {
 setfillstyle(1,BLACK);
 bar(419,178,424,382);
 setfillstyle(1,color);
 bar(370,183,475,382);
   }
void DCLOSE_2(int color)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
setfillstyle(1,color);
bar(419,178,424,382);
   }

  void UP(int x,int y,int color)
{
  struct PTS{
  int x,y;  };
  struct PTS poly[5];
  setfillstyle(SOLID_FILL,color);
  setcolor(1);
  poly[0].x=x-10;
  poly[0].y=y;
  poly[1].x=x+10;
  poly[1].y=y;
  poly[2].x=x;
  poly[2].y=y-10;
  fillpoly(3,(int far *)poly);

}
  void DOWN(int x,int y,int color)
{
  struct PTS{
  int x,y;  };
  struct PTS poly[5];
  setfillstyle(SOLID_FILL,color);
  poly[0].x=x+10;
  poly[0].y=y;
  poly[1].x=x-10;
  poly[1].y=y;
  poly[2].x=x;
  poly[2].y=y+10;
  fillpoly(3,(int far *)poly);
}
  void CLOSE(int x,int y)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
 struct PTS{
 int x,y;  };
 struct PTS poly[3];
 setfillstyle(SOLID_FILL,4);
 poly[0].x=x;
 poly[0].y=y;
 poly[1].x=x-10;
 poly[1].y=y-7;
 poly[2].x=x-10;
 poly[2].y=y+7;
 fillpoly(3,(int far *)poly);
 poly[0].x=x;
 poly[0].y=y;
 poly[1].x=x+10;
 poly[1].y=y-7;
 poly[2].x=x+10;
 poly[2].y=y+7;
 fillpoly(3,(int far *)poly);
}
void OPEN(int x,int y)
{
struct PTS{
int x,y;  };
struct PTS poly[3];
setfillstyle(SOLID_FILL,4);
poly[0].x=x;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y-7;
poly[2].x=x-10;
poly[2].y=y+7;
fillpoly(3,(int far *)poly);
}
void OPEN_1(int x,int y)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
struct PTS{
int x,y;  };
struct PTS poly[3];
setfillstyle(SOLID_FILL,4);
poly[0].x=x;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y-7;
poly[2].x=x+10;
poly[2].y=y+7;
fillpoly(3,(int far *)poly);
}
void level_1_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(SOLID_FILL,color);
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
void level_2_DOWN_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(SOLID_FILL,color);
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
void level_2_UP_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(SOLID_FILL,color);
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
void level_3_DOWN_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void level_3_UP_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
void level_4_DOWN_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
void level_4_UP_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
void level_5_1(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
void POS1(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(407,85,"1");
}
void POS2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(430,85,"2");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void POS3(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(456,85,"3");
}
void POS4(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(482,85,"4");
}
void POS5(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(507,85,"5");
}
/*****************************************************************/
void VIEW_LIFT2(void)
{
int i;
xm = getmaxx(); ym = getmaxy();
setfillstyle(1,8);
bar(0,0,xm,ym);
setfillstyle(1,1);
bar(40,38,xm-40,ym-38);
settextstyle(1,0,2);
setcolor(WHITE);
outtextxy(250,7,"VIEW LIFT NUMBER 2");
setfillstyle(1,RED);  /* background  lift no 1 */
bar(70,73,300,407);
bar(350,73,570,407);
setfillstyle(1,2);
bar(90,93,200,387);
bar(205,93,280,387);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setfillstyle(1,BLACK);
bar(350,123,570,133);
bar(365,153,480,387);
setfillstyle(1,7);
bar(370,158,475,382);
setfillstyle(1,BLACK);
bar(370,178,475,183);
bar(419,178,424,382);
setfillstyle(1,7);
bar(485,200,565,345);
setfillstyle(1,8);
bar(495,210,555,235);     /* LIFT UP & DOWN  */
bar(485,245,565,345);
setfillstyle(1,BLACK);
bar(492,255,510,280);     /*  number 1 */
bar(492,285,510,310);     /*  number 4 */
bar(492,315,517,333);     /*  door open */
bar(516,255,534,280);     /*  number 2 */
bar(534,315,558,333);     /*  door close */
bar(540,255,558,280);     /*  number 3  */
bar(540,285,558,310);     /*  number 5  */
setfillstyle(1,BLACK);
bar(400,83,520,113);
setfillstyle(1,RED);
bar(420,83,425,113);
bar(445,83,450,113);
bar(470,83,475,113);
bar(495,83,500,113);
setcolor(7);
settextstyle(1,0,1);
outtextxy(407,85,"1");   /* top room */
outtextxy(430,85,"2");
outtextxy(456,85,"3");
outtextxy(482,85,"4");
outtextxy(507,85,"5");
outtextxy(497,255,"1");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 outtextxy(521,255,"2");
 outtextxy(545,255,"3");
 outtextxy(497,285,"4");
 outtextxy(545,285,"5");
   UP_2(510,228,7);
   DOWN_2(540,219,7);
   CLOSE_2(546,324);
   OPEN_2(516,324);
   OPEN_1_2(492,324);
 setfillstyle(1,7);
 bar(102,105,188,155);
 bar(102,160,188,210);
 bar(102,215,188,265);
 bar(102,270,188,315);
 bar(102,320,188,370);
/****END OF LEVEL****/
 bar(245,115,265,145);
 bar(245,170,265,200);
 bar(245,225,265,255);
 bar(245,280,265,310);
 bar(245,330,265,360);
/****END OF BOTTOM LEVEL****/
 level_1_2(255,348,8);
 level_2_DOWN_2(255,297,8);
 level_2_UP_2(255,293,8);
 level_3_DOWN_2(255,242,8);
 level_3_UP_2(255,237,8);
 level_4_DOWN_2(255,188,8);
 level_4_UP_2(255,182,8);
 level_5_2(255,127,8);
 comreceive_2();
 }
 void comreceive_2(void)
 {
unsigned out;
unsigned in,status;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_bios_serialcom(_COM_INIT,COM2,SETTINGS);
for(;;)
{
status = _bios_serialcom(_COM_STATUS,COM2,0);
if( status & DATA_READY)
if((out = _bios_serialcom(_COM_RECEIVE,COM2,0)&0x7F) != 0)
switch(out)
{
 case 89  :    ONE_1_2(7);
                 POS1_2(7);
                 break;
 case 90  :    ONE_1_2(14);
                 POS1_2(14);
                 break;
 case 97  :    TWO_1_2(7);
                 POS2_2(7);
                 break;
 case 98  :    TWO_1_2(14);
                 POS2_2(14);
                 break;
 case 99  :    THREE_1_2(7);
                 POS3_2(7);
                 break;
 case 100 :    THREE_1_2(14);
                 POS3_2(14);
                 break;
 case 101 :    FOUR_1_2(7);
                 POS4_2(7);
                 break;
 case 102 :    FOUR_1_2(14);
                 POS4_2(14);
                 break;
 case 103 :    FIVE_1_2(7);
                 POS5_2(7);
                 break;
 case 104 :    FIVE_1_2(14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                POS5_2(14);
                break;
case 105 :      SELECT1_2(7);
                break;
case 106 :      SELECT1_2(14);
                break;
case 107 :      SELECT2_2(7);
                break;
case 108 :      SELECT2_2(14);
                break;
case 109 :      SELECT3_2(7);
                break;
case 110 :      SELECT3_2(14);
                break;
case 111 :      SELECT4_2(7);
                break;
case 112 :      SELECT4_2(14);
                break;
case 113 :      SELECT5_2(7);
                break;
case 114 :      SELECT5_2(14);
                break;
case 115 :      UP_2(510,228,7);
                break;
case 116 :      UP_2(510,228,14);
                break;
case 117 :      DOWN_2(540,219,7);
                break;
case 118 :      DOWN_2(540,219,14);
                break;
case 119 :      level_5_2(255,127,8);
                break;
case 120 :      level_5_2(255,127,14);
                break;
case 121 :      level_4_UP_2(255,182,8);
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 122 :    level_4_UP_2(255,182,14);
                break;
case 48  :    level_4_DOWN_2(255,188,8);
                break;
case 49  :    level_4_DOWN_2(255,188,14);
                break;
case 50  :    level_3_UP_2(255,237,8);
                break;
case 51  :    level_3_UP_2(255,237,14);
                break;
case 52  :    level_3_DOWN_2(255,242,8);
                break;
case 53  :    level_3_DOWN_2(255,242,14);
                break;
case 54  :    level_2_UP_2(255,293,8);
                break;
case 55  :    level_2_UP_2(255,293,14);
                break;
case 56  :    level_2_DOWN_2(255,297,8);
                break;
case 57  :    level_2_DOWN_2(255,297,14);
                break;
case 58  :    level_1_2(255,348,8);
                break;
case 59  :    level_1_2(255,348,14);
                break;
case 62  :    DOPEN_2(14);
                break;
case 63  :     DCLOSE_3(7);
               DCLOSE_4(16);
              break;
default  :   break;
}
if(kbhit())
{
if((in = getch()) == '\x0D')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  break;
   }
  }


getch();
  }

void FIVE_1_2(int color)
{
setfillstyle(1,color);
bar(102,105,188,155);
}
void FOUR_1_2(int color)
{
setfillstyle(1,color);
bar(102,160,188,210);
 }
void THREE_1_2(int color)
{
setfillstyle(1,color);
bar(102,215,188,265);
 }
void TWO_1_2(int color)
{
setfillstyle(1,color);
bar(102,270,188,315);
}
void ONE_1_2(int color)
{
setfillstyle(1,color);
bar(102,320,188,370);
 }
void DOPEN_2(int color)
{
setfillstyle(1,color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bar(370,183,475,382);
 }
void DCLOSE_3(int color)
{
setfillstyle(1,BLACK);
bar(419,178,424,382);
setfillstyle(1,color);
bar(370,183,475,382);
 }
void DCLOSE_4(int color)
{
setfillstyle(1,color);
bar(419,178,424,382);
 }
  void UP_2(int x,int y,int color)
{
  struct PTS{
  int x,y;  };
  struct PTS poly[5];
  setfillstyle(SOLID_FILL,color);
  setcolor(1);
  poly[0].x=x-10;
  poly[0].y=y;
  poly[1].x=x+10;
  poly[1].y=y;
  poly[2].x=x;
  poly[2].y=y-10;
  fillpoly(3,(int far *)poly);

}
  void DOWN_2(int x,int y,int color)
{
  struct PTS{
  int x,y;  };
  struct PTS poly[5];
  setfillstyle(SOLID_FILL,color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 poly[0].x=x+10;
 poly[0].y=y;
 poly[1].x=x-10;
 poly[1].y=y;
 poly[2].x=x;
 poly[2].y=y+10;
 fillpoly(3,(int far *)poly);
}
 void CLOSE_2(int x,int y)
{
 struct PTS{
 int x,y;  };
 struct PTS poly[3];
 setfillstyle(SOLID_FILL,4);
 poly[0].x=x;
 poly[0].y=y;
 poly[1].x=x-10;
 poly[1].y=y-7;
 poly[2].x=x-10;
 poly[2].y=y+7;
 fillpoly(3,(int far *)poly);
 poly[0].x=x;
 poly[0].y=y;
 poly[1].x=x+10;
 poly[1].y=y-7;
 poly[2].x=x+10;
 poly[2].y=y+7;
 fillpoly(3,(int far *)poly);
}
void OPEN_2(int x,int y)
{
struct PTS{
int x,y;  };
struct PTS poly[3];
setfillstyle(SOLID_FILL,4);
poly[0].x=x;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y-7;
poly[2].x=x-10;
poly[2].y=y+7;
fillpoly(3,(int far *)poly);
}
void OPEN_1_2(int x,int y)
{
struct PTS{
int x,y;  };
struct PTS poly[3];
setfillstyle(SOLID_FILL,4);
poly[0].x=x;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y-7;
poly[2].x=x+10;
poly[2].y=y+7;
fillpoly(3,(int far *)poly);
}
void level_1_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(SOLID_FILL,color);
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void level_2_DOWN_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(SOLID_FILL,color);
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
void level_2_UP_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(SOLID_FILL,color);
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
void level_3_DOWN_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
void level_3_UP_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
void level_4_DOWN_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
void level_4_UP_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x-10;
poly[0].y=y;
poly[1].x=x+10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y-10;
fillpoly(3,(int far *)poly);
}
void level_5_2(int x,int y,int color)
{
struct PTS{
int x,y;  };
struct PTS poly[5];
setfillstyle(1,color);
setcolor(1);
poly[0].x=x+10;
poly[0].y=y;
poly[1].x=x-10;
poly[1].y=y;
poly[2].x=x;
poly[2].y=y+10;
fillpoly(3,(int far *)poly);
}
void POS1_2(int color)
{
setcolor(color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
settextstyle(1,0,1);
outtextxy(407,85,"1");
}
void POS2_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(430,85,"2");
}
void POS3_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(456,85,"3");
}
void POS4_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(482,85,"4");
 }
void POS5_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(507,85,"5");
 }
void SELECT1_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(497,255,"1");
 }
void SELECT2_2(int color)
{
setcolor(color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
settextstyle(1,0,1);
outtextxy(521,255,"2");
 }
void SELECT3_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(545,255,"3");
 }
void SELECT4_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(497,285,"4");
}
void SELECT5_2(int color)
{
setcolor(color);
settextstyle(1,0,1);
outtextxy(545,285,"5");
}
```

/**********END OF PROGRAM**********/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

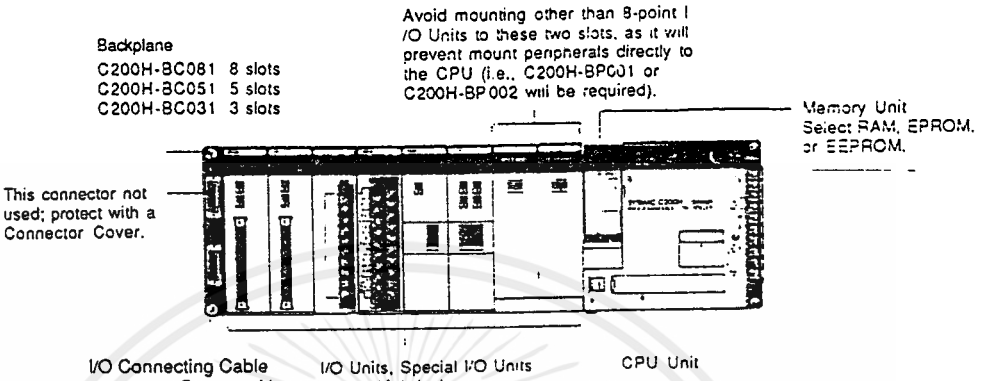
ภาคผนวก 3

ข้อมูลทางเทคนิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU Rack

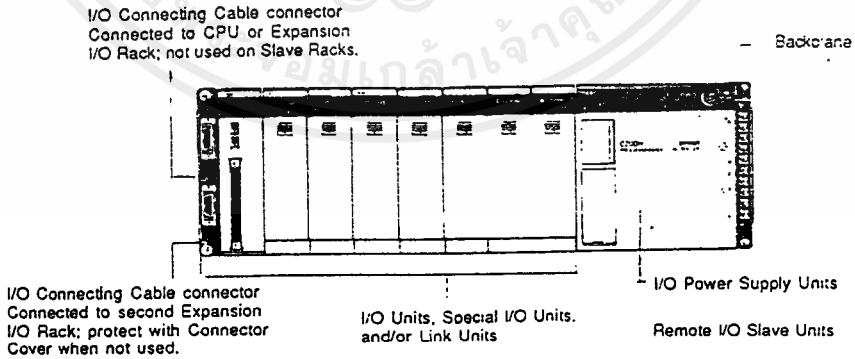
The CPU Rack consists of a Backplane, CPU, and Memory Unit, plus any I/O Units, Special I/O Units, or Link Units required by the system.



I/O Connecting Cable
(10m max. for one cable:
12m max. total for all cables)

Expansion I/O Racks and Slave Racks

Up to two Expansion I/O Racks and five Slave Racks can be connected. Expansion I/O Racks consists of a Backplane and I/O Power Supply Unit, plus any I/O Units, Special I/O Units, and/or Link Units required by the system. Slave Racks are the same, but they require no I/O Power Supply Unit and have a Remote I/O Slave Unit that connects the Rack to a Remote I/O Master Unit on a CPU or Expansion I/O Rack. Host Link Units, PC Link Units, and Remote I/O Master Units cannot be mounted to Slave Racks.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C200H Racks

Name		Specifications		Model number	Standards
Backplane (same for all Racks)		8 slots		C200H–BC081	U, C
		5 slots		C200H–BC051	U, C
		3 slots		C200H–BC031	—
CPU Rack	CPU	100 to 120/200 to 240 VAC		C200H–CPU01–E	U, C
		24 VDC		C200H–CPU03–E	—
	Memory Unit	CMOS–RAM Unit; built–in backup battery	UM: 3K words; DM: 1K channels	C200H–MR431	U, C
			UM: 7K words; DM: 1K channels	C200H–MR831	U, C
		CMOS–RAM Unit; capacitor back-up	UM: 3K words; DM: 1K channels	C200H–MR432	U, C
			UM: 7K words; DM: 1K channels	C200H–MR832	U, C
		EPROM Unit (EPROM ordered separately)	UM: 7K words; DM: 1K channels	C200H–MP831	U, C
		EEPROM Unit	UM: 3K words; DM: 1K channels	C200H–ME431	U, C
		EEPROM Unit	UM: 7K words; DM: 1K channels	C200H–ME831	—
Expansion I/O Racks	EPROM	27128; 200 ns; write voltage: 21 V		ROM–IB	—
		27128; 150 ns; write voltage: 21 V		ROM–ID	—
	I/O Power Supply Unit	100 to 120/200 to 240 VAC		C200H–PS221	U, C
		24 VDC		C200H–PS211	—
	I/O Connecting Cable (max. total length: 12 m)	30 cm		C200H–CN311	U, C
		70 cm		C200H–CN711	U, C
		2 m		C200H–CN221	U, C
		5 m		C200H–CN521	U, C
		10 m		C200H–CN131	U, C

- U: UL, C: CSA, N: NK

See Omron sales representatives concerning operating conditions under which UL, CSA, and NK standards were met (Aug. 1988).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C200H I/O Units

Name		Specifications		Model number	Standards
Input Units	AC Input Unit	8 pts	100 VAC	C200H–IA121	U, C
		16 pts	100 VAC	C200H–IA122	U, C
		8 pts	200 VAC	C200H–IA221	U, C
		16 pts	200 VAC	C200H–IA222	—
	DC Input Unit	8 pts	No–voltage contact; NPN	C200H–ID001	U, C
		8 pts	No–voltage contact; PNP	C200H–ID002	U, C
		8 pts	12 to 24 VDC	C200H–ID211	U, C
		16 pts	24 VDC	C200H–ID212	U, C
	AC/DC Input Unit	8 pts	12 to 24 VAC/DC	C200H–IM211	U, C
		16 pts	24 VAC/DC	C200H–IM212	U, C
Output Units	Relay Output Unit	8 pts	2 A, 250 VAC/24 VDC (for resistive load)	C200H–OC221	U, C
		12 pts	2 A, 250 VAC/24 VDC (for resistive load)	C200H–OC222	U, C
		5 pts	2 A, 250 VAC/24 VDC (for resistive load) Independant commons	C200H–OC223	U, C
		8 pts	2 A, 250 VAC/24 VDC (for resistive load) Independant commons	C200H–OC224	U, C
	Triac Output Unit	8 pts	1 A, 200 VAC	C200H–OA221	—
		12 pts	0.3 A, 200 VAC	C200H–OA222	U, C
	Transistor Output Unit	8 pts	1 A, 12 to 48 VDC	C200H–OD411	U, C
		12 pts	0.3 A, 24 VDC	C200H–OD211	U, C
		8 pts	2.1 A, 24 VDC	C200H–OD213	—
		8 pts	0.8 A, 24 VDC; source type (positive common); w/load short protection	C200H–OC214	—

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C200H Special and Intelligent I/O Units

All of the following are classified as Special I/O Units except for the ASCII Unit, which is an Intelligent I/O Unit.

Name		Specifications		Model number	Standards
Multipoint I/O Units	DC Input Unit	32 pts	5 VDC (TTL inputs); w/quick response function	C200H–ID501	—
		32 pts	24 VDC; w/quick response function	C200H–ID215	
	Transistor Output Unit	32 pts	0.1 A, 24 VDC (useable as 128–point dynamic output unit)	C200H–OD215	
		32 pts	35 mA, 5 VDC (TTL outputs) (useable as 128–point dynamic output unit)	C200H–OD501	
	DC Input/Transistor Output Unit	16 input pts	24–VDC inputs; w/quick response function 0.1 A, 24–VDC outputs (useable as 128–point dynamic input unit)	C200H–MD215	
		16 output pts	5 VDC (TTL inputs); w/quick response function 35 mA, 5 VDC Output (TTL outputs) (useable as 128–point dynamic input unit)	C200H–MD501	
Analog I/O Units	Analog Input Unit	4 to 20 mA, 1 to 5/0 to 10 V (switchable); 4 input		C200H–AD001	
	Analog Output Unit	4 to 20 mA, 1 to 5/0 to 10 V (switchable); 2 input		C200H–DA001	
	Temperature Sensor Unit	Thermocouple (K(CA) or J(IC))(switchable)		C200H–TS001	
		Platinum resistance thermometer (Pt)(switchable)		C200H–TS101	
Positioning Units	Position Control Unit	1 axis	Pulse output; speeds: 1 to 99,990 pps	C200H–NC111	
		1 axis	Directly connectable to servomotor driver; compatible with line driver; speeds: 1 to 250,000 pps	C200H–NC112	
		2 axis	1 to 250000. pps. 53 pts per axis	C200H–NC211	
	High–speed Counter Unit	1 axis	Pulse input; counting speed: 50 kcps; 5 VDC/12 VDC/24 VDC	C200H–CT001–V1	
		1 axis	Pulse input; counting speed: 75 kcps; RS–422 line driver	C200H–CT002	
ASCII Unit		EEPROM		C200H–ASC02	

- U: UL, C: CSA, N: NK

See Omron sales representatives concerning operating conditions under which UL, CSA, and NK standards were met (Aug. 1988).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C200H Link Units

Name	Specifications			Model number	Standards
Host Link Unit	Rack–mounting	C200H only	(APF/PCF)	C200H–LK101–P	
			RS–422	C200H–LK202	
			RS–232C	C200H–LK201	
	CPU–mounting	C2000H C1000H	(PCF)	3G2A6–LK101–EV1	
		C500	(APF/PCF)	3G2A6–LK101–PEV1	
		C200H	RS–232C	3G2A6–LK201–EV1	
		C120	RS–422	3G2A6–LK202–EV1	
PC Link Unit	Multilevel		RS–485	C200H–LK401	—
Remote I/O Master Unit	Up to two per PC; connectable to up to 5 Slaves per PC total		(APF/PCF)	C200H–RM001–PV	
			Wired	C200H–RM201	
Remote I/O Slave Unit	100 to 120/200 to 240 VAC (switchable)			C200H–RT001–P	
	24 VDC		(APF/PCF)	C200H–RT002–P	
	100 to 120/200 to 240 VAC (switchable)		Wired	C200H–RT201	
	24 VDC			C200H–RT202	
Analog Timer Unit	4 timers	0.1 to 1 s/1 to 10 s/10 to 60 s/1 min to 10 min (switchable)		C200H–TM001	
Variable Resister Connector (Related Product)	Connector w/lead wire (2 m) for 1 external resistor			C4K–CN223	—

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Optional Products

Name	Specifications			Model number	Standards
I/O Unit Cover	Cover for 10–pin terminal block			C200H–COV11	
Terminal Block Cover	Short protection for 10–pin terminal block			C200H–C0V02	
	Short protection for 19–pin terminal block			C200H–COV03	
Connector Cover	Protective cover for unused I/O Connecting Cable connectors			C500–COV02	
Space Unit	Used for vacant slots			C200H–SP001	
Battery Set	For C200H RAM Memory Unit only			C200H–BAT09	
Relay	24 VDC			G6B–1174P–FD–US DC24	
Backplane Insulation Plate	For 8–slot Backplane			C200H–ATT81	
	For 5–slot Backplane			C200H–ATT51	
	For 3–slot Backplane			C200H–ATT31	
I/O Bracket	For 5–slot Backplane			C200H–ATT53	
	For 8–slot Backplane			C200H–ATT83	
	For 3–slot Backplane			C200H–ATT33	
Memory Unit Lock Fitting	To secure Memory Unit to CPU			C200H–ATT02	
Programming Console Adapter	Used to mount 16– or 32–point I/O Units to rightmost two slots when mounting peripherals directly to CPU	Height	29 mm	C200H–BP001	
			49 mm	C200H–BP002	

- U: UL, C: CSA, N: NK

See Omron sales representatives concerning operating conditions under which UL, CSA, and NK standards were met (Aug. 1988).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Optional Products Continued

Name	Specifications	Model number	Standards
External Connector	Solder terminal; 40p and a Connector Cover	C500–CE401	—
	Solderless terminal; 40p and a Connector Cover (Crimp–type)	C500–CE402	—
	Pressure welded terminal; 40p	C500–CE403	—

DIN Products

Name	Specifications	Model number	Standards
DIN Track Mounting Bracket	1 set (1 included)	C200H–DIN01	—
DIN Track	Length: 50 cm; height: 7.3 cm	PFP–50N	
	Length: 1 m; height: 7.3 cm	PFP–100N	
	Length: 1 m; height: 16 mm	PFP–100N2	
End Plate	—	PFP–M	
Spacer	—	PFP–S	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Sets

Basic Instructions

The following tables describe all ladder diagram programming instructions and list the data areas that can be applied to each.

Differentiated instructions (indicated with @) are written by pressing NOT on the Programming Console following the function code.

The DM area can be indirectly addressed (*DM) by inputting the word that contains the word number of the desired data.

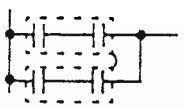
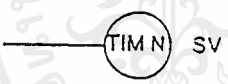
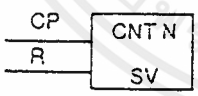
Name Mnemonic	Symbol	Function	Operands	Page
Load LD	B	Used to start instruction line with status of designated bit.	B: IR SR HR AR LR TC TR	67
Load NOT LD NOT	B	Used to start instruction line with inverse of designated bit.	B: IR SR HR AR LR TC	67
AND AND	B	Logically ANDs status of designated bit with execution condition.	B: IR SR HR AR LR TC	67
AND NOT AND NOT	B	Logically ANDs inverse of designated bit with execution condition.	B: IR SR HR AR LR TC	67
OR OR	B	Logically ORs status of designated bit with execution condition.	B: IR SR HR AR LR TC	67
OR NOT OR NOT	B	Logically ORs inverse of designated bit with execution condition.	B: IR SR HR AR LR TC	67

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 22515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Block AND AND LD		Logically ANDs results of preceding blocks.	None	68
Block OR OR LD		Logically ORs results of preceding blocks.	None	68
Output OUT		Turns ON B for ON execution condition; turns OFF B for OFF execution condition.	B: IR SR HR AR LR TR	68
Output NOT OUT NOT		Turns OFF B for ON execution condition; turns ON B for OFF execution condition.	B: IR SR HR AR LR	68
Timer TIM		ON-delay (decrementing) timer operation. Set value: 999.9 s; accuracy: +0/–0.1 s. Same TC bit cannot be assigned to more than one timer/counter. The TC bit is input as a constant.	N: SV: TC IR HR AR LR DM #	77
Counter CNT		A decrementing counter. SV: 0 to 9999; CP: count pulse; R: reset input. The TC bit is input as a constant.	N: SV: TC IR HR AR LR DM #	82

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Instructions

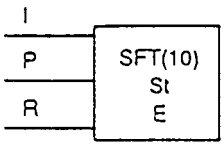
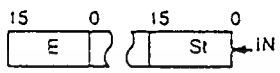
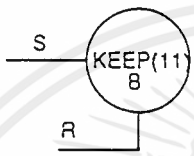
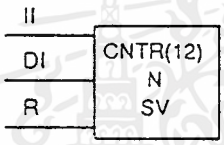
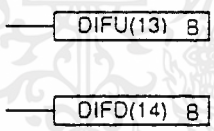
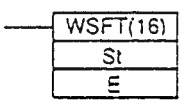
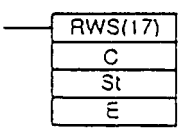
Name Mnemonic	Symbol	Function	Operands	Page
End END(01)	END(01)	Required at the end of the program.	None	76
Interlock IL(02) Interlock Clear ILC(03)	IL(02) ILC(03)	If interlock condition is OFF, all outputs are turned OFF and all timer PVs reset between this IL(02) and the next ILC(03). Other instructions are treated as NOP; counter PVs are maintained.	None	73
Jump JMP(04) Jump End JME(05)	JMP(04) N JME(05) N	Cause all instructions between JMP(04) and the corresponding JME(05) to be ignored. Corresponding JME is one of same number; 01 through 99 only usable once per program (direct jumps); 00 may be used as many times as necessary, but instructions between JMP 00 and JME 00 treated as NOP, increasing scan time over other jumps.	N: 00—99	75
Failure Alarm and Reset FAL(06)	FAL(06) N	When N is 01 to 99, an error that will not stop the CPU is indicated by outputting N (the FAL number) to the FAL output area. If N is 00, any data in the FAL output area is cleared and any other FAL number recorded in memory replaces it. The same FAL numbers are used for both FAL(06) and FALS(07).	N: 00—99	162
Severe Failure Alarm FALS(07)	FALS(07) N	An error is indicated by outputting N to the FAL output area and the CPU is stopped. The same FAL numbers are used for both FAL(06) and FALS(07).	N: 01—99	162
Step Define STEP(08)	STEP(08) B	When used with a control bit (B), defines the start of a new step and resets the previous step. When used without N, defines the end of step execution.	B: IR HR AR LR	155
Step Start SNXT(09)	SNXT(09) B	Used with a control bit (B) to indicate the end of the step, reset the step, and start the next step.	B: IR HR AR LR	155

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Shift Register SFT(10)		Creates a bit shift register from the starting word (St) through the ending word (E). I: input bit; P: shift pulse; R: reset input. St must be less than or equal to E and St and E must be in the same data area. 	St/E: IR HR AR LR	87
Latching Relay KEEP(11)		Defines a bit (B) as a latch controlled by set (S) and reset (R) inputs.	B: IR HR AR LR	71
Reversible Counter CNTR (12)		Increases or decreases PV by one whenever the increment input (II) or decrement input (DI) signals, respectively, go from OFF to ON. SV: 0 to 9999; R: reset input. Must not access the same TC bit as another timer/counter. The TC bit is input as a constant.	N: TC SV: IR SR HR AR LR DM #	35
Differentiate Up DIFU(13) Differentiate Down DIFD(14)		DIFU turns ON the designated bit (B) for one scan on the rising edge of the input signal; DIFD turns ON the bit for one scan on the trailing edge.	B: IR HR AR LR	69
High-speed Timer TIMH(15)		A high-speed, ON-delay (decrementing) timer. SV: 0.01 to 99.99 s; accuracy: +0/–0.1 s. Must not be assigned the same TC bit as another timer or counter. The TC bit is input as a constant.	N: TC SV: IR SR HR AR LR HR #	81
Word Shift WSFT(16)		Left shifts data between starting (St) and ending (E) words in word units, writing zeros into starting word. St must be less than or equal to E and St and E must be in the same data area.	St/E: IR HR AR LR DM	94
Reversible Word Shift RWS(17)		Creates and controls a reversible non-synchronous word shift register between St and E. Only shifts words when the next word in the register is zero. The shift direction is determined by C. St and E must be in the same data area. (CPU11 only)	C: IR SR HR AR LR TC DM # St/E: IR SR HR AR LR TC DM	95

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

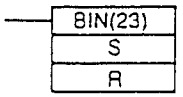
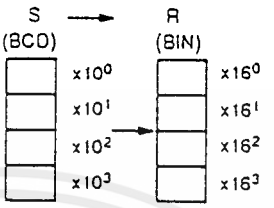
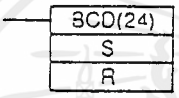
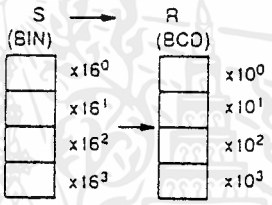
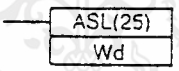
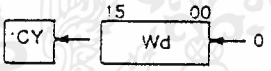
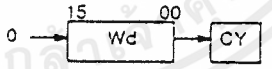
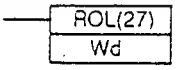
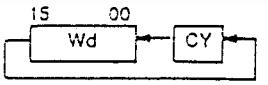
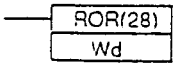
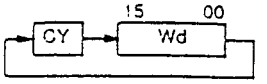
Name Mnemonic	Symbol	Function	Operands	Page
Scan Time SCAN(18)	SCAN(18) Mi — —	Sets the minimum scan time, Mi, in tenths of Milliseconds. The possible setting range is from 0 to 999.0 ms. If the actual scan time is less than the time set with SCAN(18), the CPU will wait until the designated time has elapsed before starting the next scan. (CPU11 only)	Mi: IR, SR, HR, AR, LR, TC, DM, # —: Not used.	163
Multiword Compare MCMP(19)	MCMP(19) TB1 TB2 R	MCMP(19) compares the content of TB1 to TB2, TB1+1 to TB2+1, TB1+2 to TB2+2, ..., and TB1+15 to TB2+15. If the first pair is equal, the first bit in R is turned OFF, etc., i.e., if the content of TB1 equals the content of TB2, bit 00 is turned OFF, if the content of TB1+1 equals the content of TB2+1, bit 01 is turned OFF, etc. The rest of the bits in R will be turned ON. (CPU11 only)	TB1: IR, SR, HR, AR, LR, TC, DM TB2: IR, SR, HR, AR, LR, TC, DM R: IR, HR, AR, LR, TC, DM	105
Compare CMP(20)	CMP(20) Co1 Co2	Compares two sets of four-digit hexadecimal data (Cp1 and Cp2) and outputs result to GR, EQ, and LE.	Cp1/Cp2: IR, SR, HR, AR, LR, TC, DM, #	106
Move MOV(21)	MOV(21) S D	Transfers source data (S) (word or four-digit constant) to destination word (D).	S: IR, SR, HR, AR, LR, TC, DM, # D: IR, HR, AR, LR, DM	96
Move NOT MVN(22)	MVN(22) S D	Inverts source data (S) (word or four-digit constant) and then transfers it to destination word (D).	S: IR, SR, HR, AR, LR, TC, DM, # D: IR, HR, AR, LR, DM	97

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

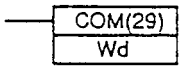
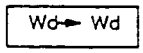
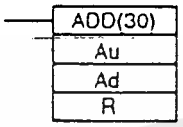
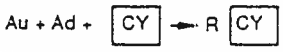
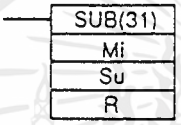
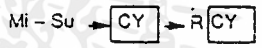
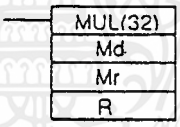
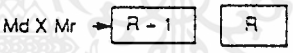
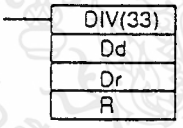
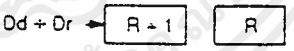
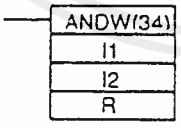
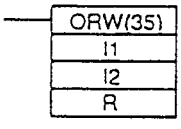
Name Mnemonic	Symbol	Function	Operands	Page
BCD to Binary BIN(23)		Converts four-digit, BCD data in source word (S) into 16-bit binary data, and outputs converted data to result word (R). 	S: IR SR HR AR LR TC DM R: IR HR AR LR DM	112
Binary to BCD BCD(24)		Converts binary data in source word (S) into BCD, and outputs converted data to result word (R). 	S: IR SR HR AR LR DM R: IR HR AR LR DM	113
Arithmetic Shift Left ASL(25)		Shifts each bit in single word (Wd) of data one bit to left, with CY. 	Wd: IR HR AR LR DM	91
Arithmetic Shift Right ASR(26)		Shifts each bit in single word (Wd) of data one bit to right, with CY. 	Wd: IR HR AR LR DM	91
Rotate Left ROL(27)		Rotates bits in single word (Wd) of data one bit to left, with carry (CY). 	Wd: IR HR AR LR DM	92
Rotate Right ROR(28)		Rotates bits in single word (Wd) of data one bit to right, with carry (CY). 	Wd: IR HR AR LR DM	92

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Complement COM(29)		Inverts bit status of one word (Wd) of data. 	Wd: IR HR AR LR DM	144
BCD Add ADD(30)		Adds two four-digit BCD values (Au and Ad) and content of CY, and outputs result to specified result word (R). 	Au/Ad: R: IR IR SR HR HR AR AR LR LR DM TC DM #	126
BCD Subtract SUB(31)		Subtracts both four-digit BCD subtrahend (Su) and content of CY from four-digit BCD minuend (Mi) and outputs result to specified result word (R). 	Mi/Su: R: IR IR SR HR HR AR AR LR LR DM TC DM #	128
BCD Multiply MUL(32)		Multiplies four-digit BCD multiplicand (Md) and four-digit BCD multiplier (Mr) and outputs result to specified result words (R and R + 1). R and R + 1 must be in the same data area. 	Md/Mr: R: IR IR SR HR HR AR AR LR LR DM TC DM #	131
BCD Divide DIV(33)		Divides four-digit BCD dividend (Dd) by four-digit BCD divisor (Dr) and outputs result to specified result words. R receives quotient; R + 1 receives remainder. R and R + 1 must be in the same data area. 	Dd/Dr: R: IR IR SR HR HR AR AR LR LR DM TC DM #	133
Logical AND ANDW(34)		Logically ANDs two 16-bit input words (I1 and I2) and sets corresponding bit in result word (R) if corresponding bits in input words are both are ON.	I1/I2: R: IR IR SR HR HR AR AR LR LR DM TC DM #	144
Logical OR ORW(35)		Logically ORs two 16-bit input words (I1 and I2) and sets corresponding bit in result word (R) if one or both of corresponding bits in input data are ON.	I1/I2: R: IR IR SR HR HR AR AR LR LR DM TC DM #	145

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

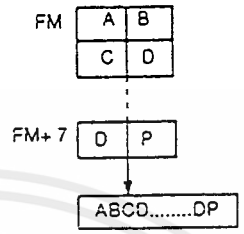
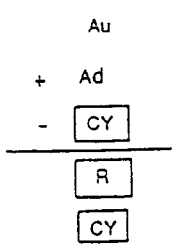
Name Mnemonic	Symbol	Function	Operands	Page
Exclusive OR XORW(36)	XORW(36) I1 I2 R	Exclusively ORs two 16-bit input words (I1 and I2) and sets bit in result (R) word when corresponding bits in input words differ in status.	I1/I2: IR SR HR AR LR TC DM # R: IR HR AR LR DM	146
Exclusive NOR XNRW(37)	XNRW(37) I1 I2 R	Exclusively NORs two 16-bit input words (I1 and I2) and sets bit in result word (R) when corresponding bits in input words are same in status.	I1/I2: IR SR HR AR LR TC DM # R: IR HR AR LR DM	146
Increment INC(38)	INC(38) Wd	Increments four-digit BCD word (Wd) by one, without affecting carry (CY).	Wd: IR HR AR LR DM	125
Decrement DEC(39)	DEC(39) Wd	Decrements four-digit BCD word by 1, without affecting carry (CY).	Wd: IR HR AR LR DM	125
Set Carry STC(40)	STC(40)	Sets carry flag (i.e., turns CY ON).	None	125
Clear Carry CLC(41)	CLC(41)	CLC clears carry flag (i.e. turns CY OFF).	None	126

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Message MSG(46)	MSG(46) FM	Displays on the Programming Console or GPC 8 words of ASCII code starting from FM. All 8 words must be in the same data area. 	FM: IR HR AR LR TC DM #	163
Long Message LMSG(47)	LMSG(47) S D —	Outputs a 32-character message to either a Programming Console or through the RS-232C interface. The output message must be in ASCII beginning in S. The destination of the message is designated in D. (CPU11 only)	S: IR HR AR LR TC DM D: #000 Not #001 used. #002 —:	165
TERMINAL Mode Conversion TRM(48)	TRM(48) — — —	The Programming Console can be switched to TERMINAL mode by pressing the CHG key on the Programming Console. The Programming Console will enter the CONSOLE mode when the CHG key is pressed again. Instructions MSG(46), LSMG(47), and the keyboard mapping function, are executed in the CONSOLE mode. (CPU11 only)	—: Not used	165
Set System SYS(49)	SYS(49) P — —	Either controls certain operating parameters or executes the same system commands that are possible from the AR area. The contents of the leftmost 8 bits of P determine which function SYS(49) will have. (CPU11 only)	P: # —: Not used.	166
Binary Add ADB(50)	ADB(50) Au Ad R	Adds four-digit augend (Au), four-digit addend (Ad), and content of CY and outputs result to specified result word (R). 	Au/Ad: IR SR HR AR LR TC DM # R: IR HR AR LR DM	139

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

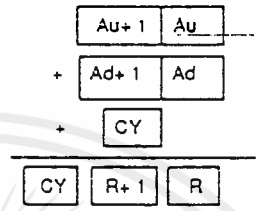
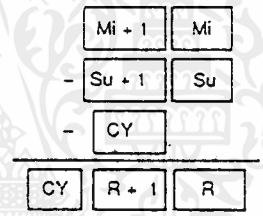
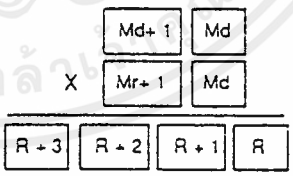
Name Mnemonic	Symbol	Function	Operands	Page
Binary Subtract S88(51)	S88(51) Mi Su R	Subtracts four-digit hexadecimal subtrahend (Su) and content of carry from four-digit hexadecimal minuend (Mi) and outputs result to specified result word (R). Mi − Su − CY R CY	Mi/Su: R: IR IR SR HR HR AR AR LR LR DM TC DM #	141
Binary Multiply MLB(52)	MLB(52) Md Mr R	Multiplies four-digit hexadecimal multiplicand (Md) by four-digit multiplier (Mr) and outputs eight-digit hexadecimal result to specified result words (R and R + 1). R and R + 1 must be in the same data area. Md X Mr Quotient R Remainder R+ 1	Md/Mr: R: IR IR SR HR HR AR AR LR LR DM TC DM #	143
Binary Divide DV8(53)	DV8(53) Dd Dr R	Divides four-digit hexadecimal dividend (Dd) by four-digit divisor (Dr) and outputs result to designated result words (R and R + 1). R and R + 1 must be in the same data area. Dd ÷ Dr Quotient R Remainder R+ 1	Dd/Dr: R: IR IR SR HR HR AR AR LR LR TC DM #	143

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Double BCD Add ADDL(54)	ADDL(54) Au Ad R	Adds two eight-digit values (2 words each) and content of CY, and outputs result to specified result words. All words for any one operand must be in the same data area. 	Au/Ad: R: IR IR SR HR HR AR AR LR LR DM TC DM	127
Double BCD Subtract SUBL(55)	SUBL(55) Mi Su R	Subtracts both eight-digit BCD subtrahend and content of CY from eight-digit BCD minuend and outputs result to specified result words. All words for any one operand must be in the same data area. 	Mi/Su: R: IR IR SR HR HR AR AR LR LR DM TC DM	130
Double BCD Multiply MULL(56)	MULL(56) Md Mr R	Multiplies eight-digit BCD multiplicand and eight-digit BCD multiplier and outputs result to specified result words. All words for any one operand must be in the same data area. 	Md/Mr: R: IR HR SR HR HR AR AR LR LR DM TC DM	132

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	*
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Double BCD Divide DIVL(57)	DIVL(57) Dd Dr R	Divides eight-digit BCD dividend by eight-digit BCD divisor and outputs result to specified result words. All words for any one operand must be in the same data area. Dd + 1 / Dd ÷ Dr + 1 / Dr Quotient: R + 1 / R Remainder: R + 3 / R + 2	Dd/Dr: IR, SR, HR, AR, LR, TC, DM R: IR, HR, AR, LR, DM	134
Double BCD to Double Binary BINL(58)	BINL(58) S R	Converts BCD value in two source words (S: starting word) into binary and outputs converted data to two result words (R: starting word). All words for any one operand must be in the same data area. S, S + 1 → R, R + 1	S: IR, SR, HR, AR, LR, TC, DM R: IR, HR, AR, LR, DM	112
Double Binary to Double BCD BCDL(59)	BCDL(59) S R	Converts binary value in two source words (S: starting word) into eight digits of BCD data, and outputs converted data to two result words (R: starting result word). Both words for any one operand must be in the same data area. S, S + 1 → R, R + 1	S: IR, SR, HR, AR, LR, DM R: IR, SR, HR, AR, LR, DM	114
Long Compare LCMP(60)	LCMP(60) Cp1 Cp2	LCMP(60) joins the 4-digit hexadecimal content of Cp1+1 with that of Cp1, and that of Cp2+1 with that of Cp2 to create two 8-digit hexadecimal numbers, Cp+1,Cp1 and Cp2+1,Cp2. The two 8-digit numbers are then compared and the result is output to the GR, EQ, and LE flags in the SR area. (CPU11 only)	Cp1: IR, SR, HR, AR, LR, TC, DM Cp2: IR, SR, HR, AR, LR, TC, DM	108

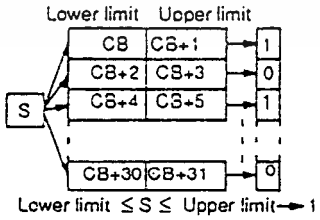
Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 22515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Write: DM 0000 to DM 0999 Read only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Move from Column MVFC(63)	MVFC(64) S D Bi	MVFC(63) copies bit column Bi from the 16 word set (S to S+15) to the 16 bits of word D (00 to 15). Bit Bi of word S is copied to bit 00 of word D, bit Bi of word S+1 is copied to bit 01 of D, and so on. (CPU11 only)	S: IR SR HR AR LR DM D: IR HR AR LR DM Bi: IR HR AR TC DM #	97
Move to Column MVTC(64)	MVTC(64) S D Bi	MVTC(64) copies the 16 bits of word S (00 to 15) to bit column Bi of the 16 word set (D to D+15). Bit 00 of word S is copied to bit Bi of word D, bit 01 of word S is copied to bit Bi of word D+1, and so on. (CPU11 only)	S: IR SR HR AR LR DM D: IR HR AR LR TC DM Bi: IR HR AR TC DM #	98
Hours to Seconds HTS(65)	HTS(65) S R —	Converts a time notation in hours/minutes/seconds (S and S+1) to an equivalent in seconds only (R and R+1). S and S+1 must be BCD and in the same data area. R and R+1 must also be in the same data area. (CPU11 only)	S: IR SR HR AR LR TC DM R: IR SR HR AR LR TC DM —: Not used.	114
Seconds to Hours STH(66)	STH(66) S R —	Converts a time notation in seconds (S and S+1) to an equivalent in hours/minutes/seconds (R and R+1). S and S+1 must be BCD between 0 and 35,999,999, and in the same data area. R and R+1 must also be in the same data area. (CPU11 only)	S: IR SR HR AR LR TC DM R: IR SR HR AR LR TC DM —: Not used.	115
Bit Counter BCNT(67)	BCNT(67) N SB R	Counts number of ON bits in one or more words (SB: source beginning word) and outputs result to specified word (R). N: number of words to be counted . All words to be counted must be in the same data area.	N: IR SR HR AR LR TC DM R: IR HR AR LR TC DM SB: IR SR HR AR LR TC DM	167
Block Compare BCMP(68)	BCMP(68) S CB R	Compares 1-word binary value (S) with 16 ranges in comparison table (CB: starting word of comparison block). If value falls within any ranges, corresponding bits of result word (R) will set. The comparison block data must all be in the same data area. Lower limit ≤ S ≤ Upper limit → 1	S: IR SR HR AR LR TC DM # CB: IR SR HR AR LR TC DM R: IR SR HR AR LR TC DM	109



Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 22515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Trigonometric and Linear Approximation TRIG(69)	TRIG(69) C In R	The operation of TRIG(69) depends on the control word C. If C is #0000 or #0001, TRIG(69) computes sin(x) or cos(x). If C is an address, TRIG(69) computes f(x) of the function entered in advance at word C. The function is a series of line segments (which can approximate a curve) determined by the operator. The result is output to word R. (x is the content of In) (CPU11 only)	C: IR, HR, AR, LR, TC, DM, # In: IR, SR, HR, AR, LR, TC, DM R: IR, HR, AR, LR, TC, DM	167
Block Transfer XFER(70)	XFER(70) N S D	Moves content of several consecutive source words (S: starting source word) to consecutive destination words (D: starting destination word). All source words must be in the same data area, as must all destination words. Transfers can be within one area or between two data areas, but the source and destination words must not overlap. S, S + 1, S+N−1 → D, D + 1, D+N−1 (No. of Words)	N: IR, SR, HR, AR, LR, TC, DM, # S: IR, HR, AR, LR, TC, DM D: IR, SR, HR, AR, LR, TC, DM, #	100
Block Set BSET(71)	BSET(71) S St E	Copies content of one word or constant (S) to several consecutive words (starting word (St) through ending word (E)). St and E must be in the same data area. S → St, E	St/E: IR, HR, AR, LR, TC, DM S: IR, SR, HR, AR, LR, TC, DM, #	99
Square Root ROOT(72)	ROOT(72) Sq R	Computes square root of eight-digit BCD value (Sq and Sq + 1) and outputs truncated four-digit integer result to specified result word (R). Sq and Sq + 1 must be in the same data area. √(Sq+1 | Sq) → R	Sq: IR, SR, HR, AR, LR, TC, DM R: IR, HR, AR, LR, DM	137

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

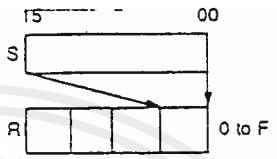
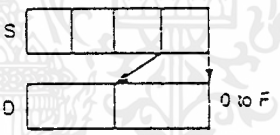
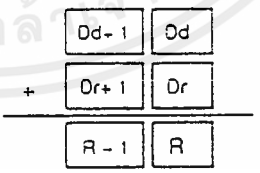
Name Mnemonic	Symbol	Function	Operands	Page
Data Exchange XCHG(73)	XCHG(73) E1 E2	Exchanges contents of two different words (E1 and E2). E1 ⟷ E2	E1/E2: IR HR AR LR TC DM	101
One Digit Shift Left SLD(74)	SLD(74) St E	Left shifts data between starting (St) and ending (E) words by one digit (four bits). St and E must be in the same data area. St → 0 St − 1 → E →	St/E: IR HR AR LR DM	93
One Digit Shift Right SRD(75)	SRD(75) E St	Right shifts data between starting (St) and ending (E) words by one digit (four bits). St and E must be in the same data area. E → E − 1 → 0 → St	St/E: IR HR AR LR DM	94
4 to 16 Decoder MLPX(76)	MLPX(76) S Di R	Converts up to four hexadecimal digits in source word (S) into decimal values from 0 to 15 and turns ON, in result word(s) (R), bit(s) whose position corresponds to converted value. Digits to be converted designated by Di (rightmost digit: indicates the first digit; next digit to left: gives the number of digits minus 1). S → 0 to F R → 15 00	S: IR SR HR AR LR TC DM Di: IR HR AR LR TC DM # R: IR HR AR LR DM	116

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

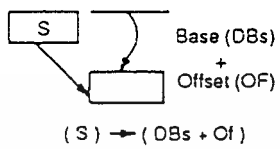
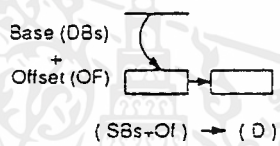
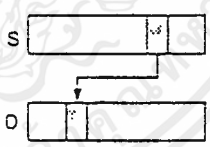
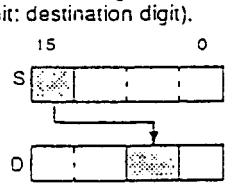
Name Mnemonic	Symbol	Function	Operands	Page
16 to 4 Encoder DMPX(77)	DMPX(77) S R Di	Determines position of highest ON bit in source word(s) (starting word: S) and turns ON corresponding bit(s) in result word (R). Digits to receive converted value are designated by Di (rightmost digit: indicates the first digit; next digit to left: gives number of words to be converted minus 1). 	S: IR SR HR AR LR TC DM R: IR HR AR LR DM Di: IR HR AR LR TC DM #	118
7-segment Decoder SDEC(78)	SDEC(78) S Di D	Converts hexadecimal values from source word (S) to data for seven-segment display. Results placed in consecutive half words starting at the first destination word (D). Di designates digit and destination details (rightmost digit: gives the first digit to be converted; next digit to the left: number of digits to be converted minus 1; next digit:: 1 = transfer first digit to left half of first destination word, 0 = transfer to right half). 	S: IR SR HR AR LR TC DM Di: IR HR AR LR TC DM # D: IR HR AR LR DM	120
Floating Point Divide FDIV(79)	FDIV(79) Od Or R	Divides one floating point value by another and outputs floating point result. Rightmost seven digits of each set of two words (eight digits) are used for mantissa, and leftmost digit used for the exponent and its sign. 	Od/Or: IR SR HR AR LR TC DM R: IR HR AR LR DM	135

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Write: DM 0000 to DM 0999 Read only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

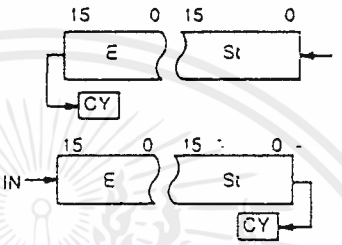
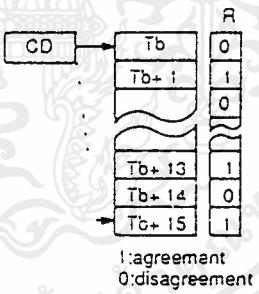
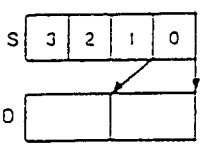
Name Mnemonic	Symbol	Function	Operands	Page
Single Word Distribute DIST(80)	DIST(80) S DBs Of	Moves one word of source data (S) to destination word whose address is given by destination base word (DBs) plus offset (Of). 	S: IR SR HR AR LR TC DM # DBs: IR HR AR LR TC DM Of: IR HR AR LR TC DM #	102
Data Collect COLL(81)	COLL(81) SBs Of D	Extracts data from source word and writes it to destination word (D). Source word is determined by adding offset (Of) to source base word (SBs). The offset cannot be input as a constant with the C120 or C500. 	SBs: IR SR HR AR LR TC DM Of: IR HR AR LR TC DM # D: IR HR AR LR TC DM	102
Move Bit MOVB(82)	MOVB(82) S Bi D	Transfers designated bit of source word or constant (S) to designated bit of destination word (D). Rightmost two digits of bit designator (Bi) designate the source bit; leftmost two, the destination bit. 	S: IR SR HR AR LR DM # Bi: IR HR AR LR TC DM # D: IR HR AR LR DM	103
Move Digit MOVD(83)	MOVD(83) S Di D	Moves hexadecimal content of specified four-bit source digit(s) (S: source word) to specified destination digit(s) (D: destination word) for up to four digits. Source and destination digits specified in Digit Designator (Di) digits (rightmost digit: source digit; next digit to left: number of digits to be moved; next digit: destination digit). 	S: IR SR HR AR LR TC DM # Di: IR HR AR LR TC DM # D: IR SR HR AR LR TC DM	104

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9515	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Reversible Shift Register SFTR(84)	SFTR(84) C St E	Shifts data in specified word or series of words to either left or right. Starting (St) and ending words (E) must be specified. Control word (C) contains shift direction, reset input, and data input. St and E must be in the same data area and St must be less than or equal to E. 	St/E/C: IR HR AR TC LR DM	89
Table Compare TCMP(85)	TCMP(85) CD TB R	Compares four-digit hexadecimal value (CD) with values in table consisting of 16 words (TB: First word of comparison table). If value equals any values, corresponding bits of result word (R) are set. The entire table must be in the same data area. 	CD: IR SR HR AR LR TC DM # TB/R: IR HR AR LR TC DM	111
ASCII Code Convert ASC(86)	ASC(86) S Di D	Converts hexadecimal values from source word (S) to eight-bit ASCII code starting at leftmost or rightmost half of starting destination word (D). Rightmost digit of Di designates first source digit; the next digit to the left, the number of digits; the next digit, the rightmost (1) or leftmost (0) half of the first destination word; and the leftmost digit even (1) or odd (0) parity. 	S: IR SR HR AR LR TC DM Di: IR HR LR TC DM # D: IR HR LR DM	123

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 23515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Interrupt Control FUN(89)	FUN(89) CC N D	Controls interrupts. CC: control code (defines the process); N: Interrupt Unit number (004: scheduled interrupt); D: control data.	CC: 000 to 002 N: 000 to 00 D: IR HR AR LR TC DM #	150..
SYSMAC NET Link/SYSMAC LINK Write SEND(90)	SEND(90) S D C	Sends data to device linked through SYSMAC NET Link/SYSMAC LINK Unit. S: starting source word to be sent from PC; D: starting destination word on node to receive transmission; C: first of three control words. For SYSMAC NET Link Systems, C gives the number of words to be transferred; bits 00 to 07 of C+1 give the network number, bit 14 of C+1 specifies either operating level 0 or 1, bits 00 to 07 of C+2 give the destination node number, and bits 08 to 15 of C+2 give the destination port number. All other bits are turned OFF. For SYSMAC LINK Systems, C gives the number of words to be transferred; bits 00 to 07 of C+1 give the response time limit, bit 13 of C+1 is turned OFF to request response, bit 14 of C+1 specifies either operating level 0 or 1, bit 15 of C+1 is turned ON to designate a SYSMAC LINK System transfer, and bits 00 to 07 of C+2 give the destination node number. All other bits are turned OFF. (CPU11 only) Source: S, S+1, S+n-1 Node N: D, D+1, D+n-1	S: IR SR HR AR LR TC DM D/C: IR HR AR LR TC DM	171
Subroutine Define SBS(91)	SBS(91) N	Calls subroutine N (moves program operation to it).	N: 00 to 99	148
Subroutine Entry SBN(92)	SBN(92) N	Marks start of subroutine N.	N: 00 to 99	147

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 22515	22600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Wr: DM 0000 to DM 0999 Rd only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name Mnemonic	Symbol	Function	Operands	Page
Subroutine Return RET(93)	RET(93)	Marks the end of a subroutine and returns control to main program.	None	147
Watchdog Timer Refresh WDT(94)	WDT(94) T	Sets the maximum and minimum limits for the watchdog timer (normally 0 to 130 ms). New limits: Maximum time = 130 + (100 x T) Minimum time = 130 + (100 x (T−1))	T: 0 to 63	170
I/O Refresh IORF(97)	IORF(97) St E	Refreshes all I/O words between the start (St) and end (E) words. Only I/O words may be designated. Normally these words are refreshed only once per scan, but refreshing words before use in an instruction can increase execution speed. St must be less than or equal to E.	St/E: IR	170
SYSMAC NET Link/SYSMAC LINK Read RECV(98)	RECV(98) S D C	Receives data from device linked through SYSMAC NET Link/SYSMAC LINK Unit. S: starting source word on node from which to receive; D: starting destination word in PC to receive transmission; C: first of three control words. For SYSMAC NET Link Systems, C gives the number of words to be transferred; bits 00 to 07 of C+1 give the network number, bit 14 of C+1 specifies either operating level 0 or 1, bits 00 to 07 of C+2 give the source node number, and bits 08 to 15 of C+2 give the source port number. All other bits are turned OFF. For SYSMAC LINK Systems, C gives the number of words to be transferred; bits 00 to 07 of C+1 give the response time limit, bit 14 of C+1 specifies either operating level 0 or 1, bit 15 of C+1 is turned ON to designate a SYSMAC LINK System transfer, and bits 00 to 07 of C+2 give the source node number. All other bits are turned OFF. (CPU 11 only) Source → Destination S → D S + 1 → D + 1 S + n − 1 → D + n − 1	S: D/C: IR IR SR HR HR AR AR LR LR TC TC DM DM	173

Data Areas

These footnote tables show the actual ranges of all data areas. Bit numbers are provided (except for DM and TC areas); remove the rightmost two digits for word numbers.

IR	SR	HR	TR	AR	LR	TC	DM	#
00000 to 22515	23600 to 25507	HR 0000 to 9915	TR 0 to 7	AR 0000 to 2715	LR 0000 to 6315	TC 000 to 511	Read/Write: DM 0000 to DM 0999 Read only: DM 1000 to DM 1999	0000 to 9999 or 0000 to FFFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OMRON

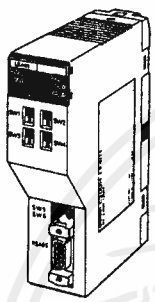
Model

C200H-LK401

PC Link Unit

INSTRUCTION SHEET

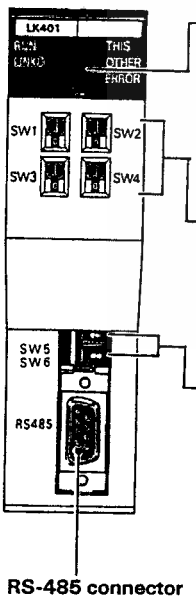
Thank you for purchasing an OMRON product. Read this thoroughly and familiarize yourself with the functions and characteristics of the product before using it.
Keep this instruction sheet for future reference.



OMRON Corporation

©OMRON Corporation 1990 All Rights Reserved.

■Nomenclature



Indicators

RUN	Running
LINK 0	Unit No.0
THIS	Your Unit
OTHER	Other Unit
ERROR	Error

Switches 1 to 4

SW1	$\times 10^1$	PC Link Unit No.
SW2	$\times 10^0$	
SW3	Special I/O Unit No.	
SW4	LR transfer bits	

Switches 5 and 6

SW No.	Left	Right
SW5	Four-wire mode (for optical fiber cable)	Two-wire mode (for wire cable)
SW6	Termination resistance ON	Termination resistance OFF

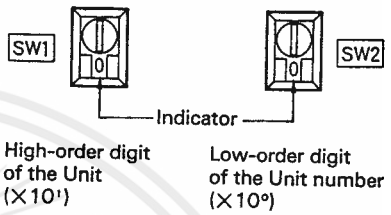
RS-485 connector

■Indicators

Indicators	Function
RUN	Lights while the Unit is operating.
LINK 0	Lights when the Unit No.0 (Master) is linked normally. Goes out if a link error occurs.
THIS	Lights when the Unit is linked normally. Goes out if a link error occurs.
OTHER	Lights when the other Units are linked normally. Goes out if a link error occurs.
ERROR	Lights when the Unit No. is set to the same No. that another Unit is set to. Blinks when PC link to Unit #0, THIS, or OTHER is broken.

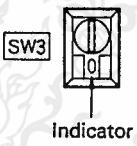
■Front Panel Switches

●Setting the PC Link Unit Number



- Use switch 1 and switch 2 to set the Unit number for the PC Link Unit.
- The PC Link Unit transmits data using the link relay area. By setting the Unit number, you can allocate part of the link relay area to your Link Unit.
- In the multilevel mode (rear panel switch SW7-3 = OFF), set a Unit number from 0 to 15. In the single-level mode (rear panel switch SW7-3 = ON), set a Unit number from 0 to 31. Do not allocate to your Unit a number that has already been allocated to some other Unit.

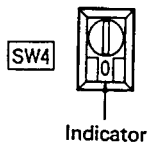
●Setting the Unit number of a Special I/O Unit



- Use switch 3 to set the Unit number for a Special I/O Unit.
- Set a Unit number from 0 to 9. Because Special I/O Units and PC Link Units share the link relay area, be careful to avoid duplicating the allocated Unit number.
- The PC link operation does not allocate the internal auxiliary relay area to each Link Unit. This is a user-defined area.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● **Setting the number of LR transfer points**

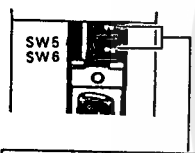


The number of LR transfer bits and maximum number of Units in the same operating level.

SW4	Single-level mode		Multi-level mode	
	LR points	Maximum Units	LR points	Maximum Units
0	32	32	32	16
1	64	16	64	8
2	128	8	128	4
3	256	4	256	2
4	512	2	Not used	
5 to 9	Not used			

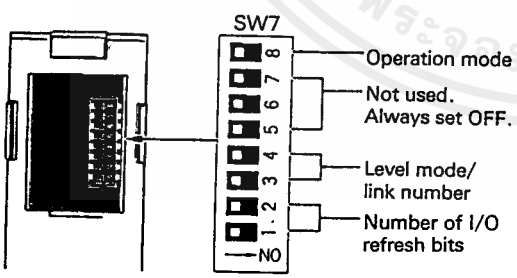
● Setting the number of LR transfer bits determines the bits of the link relay area through which data is transferred.
● The total number of LR transfer bits is 1024 in the single level mode, and 512 per level in the multilevel mode.
● Set the number of LR transfer bits via Unit number 0 (the Master). You cannot set this value via another Units.

■Setting transfer line and termination resistance



SW No.	Function	Left	Right
SW5	Sets transfer line	4-wire mode (for optical fiber cable)	2-wire mode (for wire cable)
SW 6	Sets termination resistance	Termination resistance ON	Termination resistance OFF

■Rear Panel DIP Switch



● **Setting the number of I/O refresh bits**

1	2	I/O refresh bits
0	0	512
0	1	256
1	0	128
1	1	64

● Set the number of LR bits for every I/O refresh cycle.
● Select either from 64, 128, 256, or 512.
● Set a value greater than the number of LR transfer bits per unit. Otherwise, the setting is invalid and the number of LR transfer bits will be used as the number of I/O refresh bits.
● Setting a small number of I/O refresh bits will result in a shorter scan time for the PC but a longer transmission delay.

● **Relationship between I/O refresh bits and scan time**

1	2	I/O refresh bits	Scan time
0	0	512	8.9 ms
0	1	256	5.7 ms
1	0	128	3.6 ms
1	1	64	2.8 ms

● **Relationship between the I/O refresh bits, LR transfer bits, and transmission delay**

LR transfer bits \ I/O refresh bits		64	128	256	512
Single-level mode	32	16 scans	8 scans	4 scans	—
	64				
	128	—			2 scans
	256	—			
	512	—			
Multi-level mode	32	8 scans	4 scans	2 scans	—
	64				
	128	—			1 scan
	256	—			

● The above table shows the scan count required to refresh 64 channels in the single level mode, or 32 channels in the multilevel mode.

● **Setting level mode/link number**

3	4	Level mode/link number
1	—	Single level mode
0	0	Multilevel mode, Link #0
0	1	Multilevel mode, Link #1

● If you configure your PC Link System with only one PC Link Unit mounted to any C200H PC, set the Unit to the single-level mode. In this mode, no more than one PC Link Unit can be mounted to a PC.
● If you configure your PC Link System with two PC Link Units mounted to any C200H PC, set one Unit to the multilevel mode as Link #0, and the other Unit to the multilevel mode as Link #1. Even if you assign the same link number to both Units, the ERROR indicator will not light.

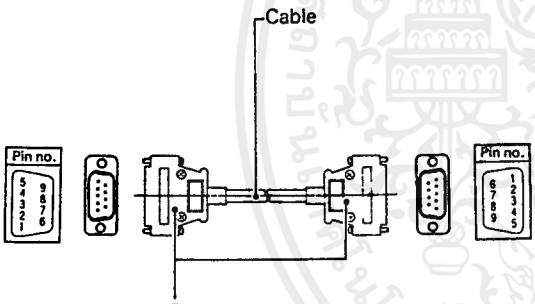
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■Troubleshooting

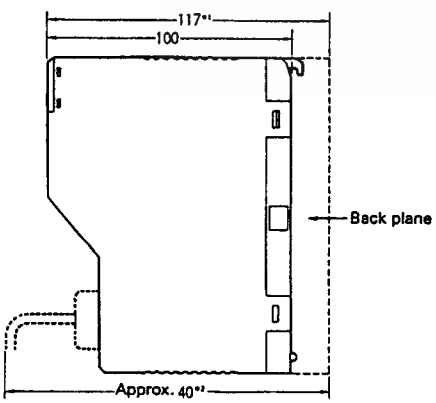
Error / States		Error in CPU of Unit #0 (PC power OFF)	Error in Special I/O Unit #0	Cable failure in Unit #0 (disconnected or broken)	Cable failure in Unit #n (diLconnected or broken)	Wrong data sent from Unit #n	Error in CPU of Unit #n (PC power OFF)	Error in Special I/O Unit #n
Unit #0	On start-up	PC link not established.			—			
	During normal cycle	All indicators OFF (PC link stopped)	RUN OFF (PC link stopped)	PC link error flag ON in all linking Units except #0; OTHER OFF; ERROR flashing. (All nonpolling Unts abnormal)	PC link error flag ON in Unit #n; OTHER OFF; ERROR flashing. (Unit #n are broken from PC link)			
Other than Unit #0 and #n	On start-up	PC link not established.			—			
	During normal cycle	PC link error flag ON in Unit #0; LINK 0 and OTHER OFF; ERROR flashing. (PC link stopped)			PC link error flag ON in Unit #n; OTHER OFF; ERROR flashing.	PC link error flag ON in this Unit; THIS OFF; ERROR flashing.	PC link error flag ON in Unit #n; OTHER OFF; ERROR flashing.	
Unit #n	On start-up	PC link not established.			—			
	During normal cycle	Error flag ON in Unit #0; LINK 0 and OTHER OFF; ERROR flashing. (PC link stopped)			PC link error flag ON in Unit #0; LINK 0 OFF; ERROR flashing.	—	All indicators OFF. (PC link stopped)	RUN OFF. (PC link stopped)

■Notes

- The following are included with the PC Link Unit.
  Confirm that you have them before using the Unit.
  Connector : XM2A-0901 (Omron)
  Connector cover : XM2S-0911 (Omron)
- Cable Connections



- Ensure that there is adequate depth on the control panel to allow for mounting the unit.



1. Mounting height of the PC Link Unit including the Backplane
2. Height including the connector

OMRON

OMRON CORPORATION
FA Division H.Q.
13F Gotenyama Mori Bldg. 4-7-35,
Kitashinagawa, Shinagawa-ku, Tokyo 140 Japan
Phone: 03-5488-3183 Fax: 03-5488-3271
Telex: 522-2484 OMRONO J

OMRON ELECTRONICS INC.
1 East Commerce Drive, Schaumburg, IL 60173
Phone: (312)843-7900 Fax: (312)843-8568
Telex: 910-291-0426 OMRONELEC SHBU
OMRON ELECTRONICS B.V.
Planetenweg 77, 2132HM Hoofddorp The Netherlands
Phone: 31-2503-62100 Fax: 31-2503-36313 Telex: 44-71343
OMRON ELECTRONICS ASIA LTD.
Unit 1510-12, Silvercord Tower 1
30, Canton Road, Tsimshatsui, Kowloon
Phone: (3)7233827 Fax: (3)7231475 Telex: 41092 OMRON HX
OMRON SINGAPORE (PTE.) LTD.
5 Little Rode #08-01, Cemtex Industrial Bldg, 1953
Phone: 2830006 Fax: 2850682 Telex: RS23403
OMRON TAIWAN ELECTRONICS INC.
6th Fl., China Trust Bldg.
No. 122 Tun Hua North Rode, Taipei, ROC
Phone: 2-715-3331 Fax: 2-712-6712

NOTE: Specification subject to change without notice.
Printed in Japan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● Setting the Operation Mode

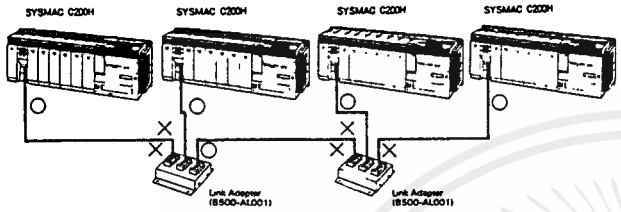
SW7-8	Operation Mode
0	Single-Unit Mode
1	Double-Unit Mode

● The single-unit/double-unit mode setting is available only for the Slave Unit.
● The operation mode for the Master Unit is always single-unit mode, regardless of the DIP switch setting.
● The single-unit mode allows the PC Link to write only to the standard LR channels.
● The double-unit mode allows the PC Link to write to the standard LR channels of the first Unit number, and to the next Unit number. Therefore, two times the standard LR area is available in the double-unit mode.

■System Configuration

There are two different types of cables that can be used to connect PC Link Units.

(1) Wire Cable (through Link Adapter B500-AL001)



The field ground is connected only at one end to prevent current from flowing to the shielded wire.
○: Shielded wire connected to FG.
×: Shielded wire not connected to FG.

Connector on PC Link Unit or Link Adapter (B500-AL001)

Pin no.	Signal name
1	
2	
4	
5	Send/receive data B (DA)
6	
7	Frame ground (FG)
8	
9	Send/receive data A (DA)
Hood	Field ground (FG)

Shielded twisted pair cable

Connector on Link Adapter (B500-AL001)

Pin no.	Signal name
1	
2	
4	
5	Send/receive data B (DB)
6	
7	Frame ground (FG)
8	
9	Send/receive data A (DA)
Hood	Field ground (FG)

● Connect the shielded twisted pair cable to the FG terminal only on the Link Adapter and, if there are Link Adapters on both ends, connect it to only one of them.
● Twist the wires connected to the DA and DB terminals.
● Refer to the PC Link Unit manual for wiring methods.

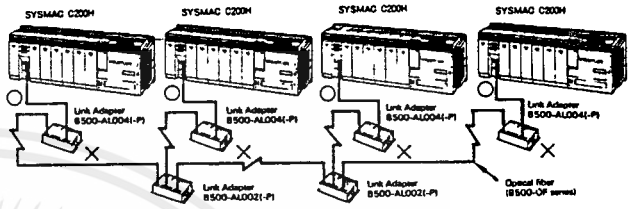
Note:
1. Set SW5 on the front panel to select wire cable.
2. Keep the total cable length to within 500m.

Connection Point for Shielded Cable

C200H-LK401	Connector hood or pin # 7
Link Adapter (B500-AL001)	Connector pin # 7

When a connector without a metal hood is used in place of the enclosed connector, connect the shielded cable to conncetor pin # 7.

(2) Optical Fiber Cable (through Link Adapter B500-AL004(-P) or B500-AL002(-P))



The field ground is connected only at one end to prevent current from flowing to the shielded wire.
○: Shielded wire connected to FG.
×: Shielded wire not connected to FG.

Connector on PC Link Unit

Pin no.	Signal name
1	Receive data B (RDB)
2	
4	
5	Send data B (SDB)
6	Receive data A (RDA)
7	Frame ground (FG)
8	
9	Send data A (SDA)
Hood	Frame ground (FG)

Shielded twisted pair cable

Connector on Link Adapter (B500-AL004(-P))

Pin no.	Signal name
1	Receive data B (RDB)
2	
4	
5	Send data B (SDB)
6	Receive data A (RDA)
7	Frame ground (FG)
8	
9	Send data A (SDA)
Hood	Frame ground (FG)

● Connect the shielded twisted pair cable to the FG terminal (connector hood or pin #7) only on the Link Adapter.
● Twist the wires connected to the SDA and SDB terminals and those connected to the RDA and RDB terminals.
● Refer to the PC Link Unit manual for wiring methods.

Note:
1. Set SW5 on the front panel to select wire cable.
2. Keep the cable length as short as possible (preferably, to within 10 m).
3. Link Adapter B500-AL001 cannot be used in a system using optical fiber cables.
4. Refer to the PC Link Unit manual for details on Link Adapters and optical fiber cables.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้