

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องวัดผลได้ทางอุตสาหกรรม



2 ก.ย.
5/529
25๖4

เลขหมู่.....
เลขทะเบียน.....
วันเดือนปี.....

นาย ชนพัฒน์ ยงพัฒนวงศ์
นาย บุญส่ง จรรย์วรพรรณ

61254906x

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการรื้อศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

YIELDOMETER



**A Special Project Submitted in Partial Fulfillment of the
Requirement for the Degree of Bachelor of Science
Department of Applied Physics
Faculty of Science
King Mongkut's Institute of technology ladkrabang**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ 1991 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ

เครื่องวัดผลได้ทางอุตสาหกรรม

โดย

นาย ธนวัฒน์ ยงพินิจวงศ์

นาย ญุ่ล่ง จรรย์วรรณ

ภาควิชา

ฟิสิกส์ประยุกต์

อาจารย์ที่ปรึกษา

ดร. เสน่ห์ เอกะวิภาต

อ. จิติ หนูแก้ว

อ. อนันต์ จารูนาวัฒน์

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตร์บัณฑิต

.....
(ดร. เสน่ห์ เอกะวิภาต) หัวหน้าภาควิชาฟิสิกส์ประยุกต์

คณะกรรมการโครงการพิเศษ

..... ประธานกรรมการ

(อ. สวรรณ คล้าราญ)

..... กรรมการ

(ดร. เสน่ห์ เอกะวิภาต)

..... กรรมการ

(อ. จิติ หนูแก้ว)

..... กรรมการ

(อ. อนันต์ จารูนาวัฒน์)

..... กรรมการ

(อ. เกร็ววัลย์ คีตะจิตต์)

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานพิเศษ
นักศึกษา

อาจารย์ที่ปรึกษา

ภาควิชา
ปีการศึกษา

เครื่องวัดผลได้ทางอุตสาหกรรม

นาย ธนวัฒน์ ยงนิพนธ์วงศ์

นาย ญุ่ส่ง จรรย์วรรณ

ดร. เสน่ห์ เอกะวิภาต

อ. จิติ หนูแก้ว

อ. อนชิต จารุrawnาวินทร์

ฟิลิกส์ประยุกต์

2534

บทคัดย่อ

เครื่องวัดผลได้ทางอุตสาหกรรม เป็นชุดอุปกรณ์ที่ควบคุมโดยระบบไมโครโปรเซสเซอร์ ใช้หาค่าผลได้ในขบวนการผลิตทั้งหมด เครื่องมือชุดนี้ประกอบด้วยชุดแสดงผลหลักและชุดแสดงผลย่อย ชุดแสดงผลหลักทำหน้าที่รับข้อมูลจากชุดแสดงผลย่อยพร้อมกับประมวลผลและแสดงค่าผลได้กับค่าสำคัญต่าง ๆ ในขบวนการผลิตทางจอภาพขนาดใหญ่ โดยมีแผงวงจรเดี่ยว Z80 CP-A เป็นตัวควบคุม และมีการคำนวณค่าผลได้ 2 วิธี

สำหรับชุดแสดงผลย่อย ควบคุมด้วยแผงวงจรเดี่ยว Z80180 โดยรับข้อมูลจากผู้ตรวจสอบตามจุดต่างๆ และนำข้อมูลทั้งหมดไปประมวลผลเพื่อหาค่าผลได้ ณ เวลานั้น แล้วแสดงค่าผลได้และค่าสำคัญต่าง ๆ ในขบวนการผลิตทางจอภาพ รวมทั้งจัดส่งข้อมูลไปยังชุดแสดงผลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title **Yieldometer**
NAME **Mr. Thanaphat Yongphiphetwong**
 Mr. Bunsong Charunworapan
Special Project Advisor **Dr. Senay Akavipet**
 Mr. Jiti Nukeaw
 Mr. Anuchit Jaruwnewet
Department **Applied Physics**
Academic Year **1991**

Abstract

Yieldometer is an equipment controlled by microprocessor system used for displaying yield of products in industrial process. There are 2 parts in this equipment, main-station and sub-stations. The main-station has 3 functions : receive all data from sub-station, calculate yields and display the results on screen 7-segment. This one is controlled by single-chip Z80 CP-A. The sub-station, which is controlled by single-chip Z80180, receives data from inspectors at various places for calculates and sends to the main-station.

กิติกรรมประกาศ

โครงการพิเศษนี้สามารถเสร็จสมบูรณ์ได้ด้วยความช่วยเหลือจากบุคคลต่าง ๆ
ดังนี้

ผู้ปกครอง ผู้คอยสนับสนุน และให้ความอุปการะ จนสำเร็จการศึกษา

ดร. เสน่ห์ เอกะวิภาต ผู้ให้คำปรึกษาด้านรายละเอียดต่าง ๆ ของโครงการ
และช่วยเหลือทุก ๆ ด้านตลอดการดำเนินงาน

อ. จิตี หนูแก้ว ผู้ให้คำปรึกษาและแนะนำเทคนิคต่าง ๆ ตลอดการ
ดำเนินงาน รวมถึงจัดหาอุปกรณ์ที่จำเป็นในโครงการพิเศษ

อ. อนุชิต จารุวนาวังษ์ ผู้ให้คำปรึกษา

รุ่นพี่ คุณ วิชิต, คุณ ชมธรา, คุณ เลมภพ, และ คุณ ไพบูลย์

เพื่อน ๆ ทุกคน ที่ให้กำลังใจและความช่วยเหลือ โดยเฉพาะ นาย ณรงค์,
นาย เพ็ญภูมิ, น.ส. วิบูลรัตน์, น.ส. ศรีโลภา และน้อง
ตง

ฝ่ายธุรการภาค ที่ให้ใช้อุปกรณ์ต่าง ๆ อย่างเต็มใจ

ขอขอบคุณบุคคลเหล่านี้อย่างจริงใจ

นาย ธนวัฒน์ ยงนิพันธ์วงศ์

นาย บุญส่ง จรรย์วรรณ

สารบัญ

หน้า

บทคัดย่อโครงงานพิเศษภาษาไทย.....	ก
บทคัดย่อโครงงานพิเศษภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญตาราง.....	ง
สารบัญรูป.....	จ
บทที่ 1 บทนำ.....	1
บทที่ 2 ระบบอาร์ดแวร์ของเครื่องวัดผลได้ทางอุตสาหกรรม.....	5
- ระบบไมโครโปรเซสเซอร์.....	5
2.1) ชุดแสดงผลหลัก (MAIN-STATION).....	15
2.1.1) ระบบควบคุมด้วยไมโครโปรเซสเซอร์ Z80 CP-A.....	15
2.1.2) ส่วนขยาย I/O PORT (7210Z80).....	18
2.1.3) ส่วนแสดงผล (7 SEGMENT).....	20
2.2) ชุดแสดงผลย่อย (SUB-STATION).....	20
2.2.1) ระบบควบคุมด้วยไมโครโปรเซสเซอร์ Z80180.....	20
2.3) วงจรรับส่งข้อมูล (CURRENT LOOP 0-20 mA).....	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3	ระบบโปรแกรมควบคุม.....	26
3.1)	ระบบโปรแกรมควบคุมของชุดแสดงผลหลัก.....	26
3.2)	ระบบโปรแกรมควบคุมของชุดแสดงผลย่อย.....	29
บทที่ 4	การวิจัยและการดำเนินงาน.....	33
4.1)	ส่วนที่ 1 การศึกษาและเพิ่มเติมอุปกรณ์ในชุดแสดงผลย่อย.....	33
4.2)	ส่วนที่ 2 การศึกษาวงจรรับส่งข้อมูล.....	38
4.3)	ส่วนที่ 3 การศึกษาอุปกรณ์ต่าง ๆ ในชุดแสดงผลหลัก.....	40
4.4)	การใช้เครื่องวัดผลได้ทางอุตสาหกรรม.....	42
บทที่ 5	ผลการวิจัยและวิจารณ์.....	47
5.1)	ชุดแสดงผลหลัก.....	47
5.2)	ชุดแสดงผลย่อย.....	48
5.3)	วงจรการรับส่งข้อมูล.....	49
บทที่ 6	สรุปผลการศึกษาวิจัยและข้อเสนอแนะ.....	53

ภาคผนวก ก. FLOW CHART

ภาคผนวก ข. โปรแกรมแอลแชนบลิ้

เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่ผู้เขียน
ประวัติผู้เขียน ให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1	หน้าที่ของพอร์ต C ในโหมด 1.....	13
ตารางที่ 2.2	หน้าที่ของพอร์ต C ในโหมด 2.....	14



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1	แสดงโครงสร้างอย่างง่ายของเครื่องวัดผลได้ทางอุตสาหกรรม....	6
รูปที่ 2.2	แสดงระบบการทำงานของ Z80.....	7
รูปที่ 2.3	แสดงแผนผังวงจรภายในและการจัดขาของไอซี 8255.....	8
รูปที่ 2.4	การเชื่อมต่อ 8255 เข้ากับระบบของ Z80.....	10
รูปที่ 2.5	แสดงการรายละเอียดแต่ละบิตของคำสั่งควบคุมของ 8255.....	11
รูปที่ 2.6	วงจรการต่อ 8255 ในโหมด 1.....	13
รูปที่ 2.7	โครงสร้างของพอร์ท A ที่ทำงานแบบสองทิศทาง.....	14
รูปที่ 2.8	แสดงโครงสร้างของชุดแสดงผลหลัก.....	16
รูปที่ 2.9	แสดงการเชื่อมต่อระหว่าง I/O PORT ของ Z80 CP-A กับชุดแสดงผล.....	17
รูปที่ 2.10	แสดงการใช้งานของ 8255 ของ 7210Z80.....	19
รูปที่ 2.11	แสดงโครงสร้างภายในของชุดควบคุมแสดงผลย่อย.....	21
รูปที่ 2.12	แสดงการเชื่อมต่อระหว่าง 8255 กับ LCD.....	22
รูปที่ 2.13	แสดงการเชื่อมต่อระหว่าง 8255 กับ ปุ่มอินพุทและ LED แบบ PULL DOWN.....	23
รูปที่ 2.14	(บน) แสดงการแปลงสัญญาณ TTL ไปเป็น 0-20 mA (ล่าง) แสดงการแปลงสัญญาณกลับจาก 0-20 mA ไปเป็น TTL...	25
รูปที่ 3.1	แสดงลักษณะโครงสร้างการทำงานของชุดแสดงผลหลัก.....	27
รูปที่ 3.2	แสดงลักษณะโครงสร้างการทำงานของชุดแสดงผลย่อย.....	30
รูปที่ 4.1	แสดงรูปบนหน้าจอ LCD ทั้ง 2 จอ.....	34
รูปที่ 4.2	แสดงรูปบนหน้าจอ LCD ตามการควบคุมของปุ่มต่างๆ.....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3	แสดงการต่ออุปกรณ์ภายในของชุดแสดงผลย่อย.....	37
รูปที่ 4.4	แสดงวงจรการรับส่งข้อมูล.....	38
รูปที่ 4.5	แสดงการต่ออุปกรณ์ภายในชุดแสดงผลหลัก.....	40
รูปที่ 4.6	แสดงการการวางปุ่มต่าง ๆ บนชุดแสดงผลย่อย.....	46
รูปที่ 5.1	แสดงชุดแสดงผลหลักที่พร้อมใช้ในทางอุตสาหกรรม.....	47
รูปที่ 5.2	แสดงการแสดงผลของหน้าจอ LCD.....	48
รูปที่ 5.3	แสดงชุดแสดงผลย่อยที่พร้อมใช้ในงานอุตสาหกรรม.....	49
รูปที่ 5.4	แสดงการเปรียบเทียบสัญญาณอินพุท TTL (รูปบน) กับ สัญญาณที่แปลงเป็น CURRENT LOOP (0-20 mA) (รูปล่าง).....	50
รูปที่ 5.5	แสดงการเปรียบเทียบสัญญาณอินพุท TTL (รูปบน) กับ สัญญาณ เอาต์พุท TTL ที่แปลงจาก CURRENT LOOP (0-20 mA) จากรูปล่างของรูปที่ 5.3 (รูปล่าง).....	50
รูปที่ 5.6	แสดงอุปกรณ์ครบชุดของเครื่องวัดผลได้ทางอุตสาหกรรม.....	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ที่มาของปัญหาที่ทำการวิจัย

ในขบวนการผลิตสินค้าใด ๆ ก็ตามในอุตสาหกรรมต่าง ๆ นั้น สิ่งที่สำคัญหรือสำคัญเป็นอย่างมากในการที่จะทำผลการผลิตได้ผลดีหรือไม่ขึ้นอยู่กับค่าผลได้ (YIELD) ที่จะเป็นตัวบอกว่าสินค้าที่ผลิตจากวัตถุดิบประเภทต่าง ๆ และขบวนการประเภทต่าง ๆ นั้น มีประสิทธิภาพแค่ไหน ค่าผลได้ (YIELD) นี้จึงเป็นสิ่งจำเป็นต่อโรงงานอุตสาหกรรมต่าง ๆ เป็นอย่างมาก

ปัญหาของโรงงานอุตสาหกรรมในปัจจุบันนี้ เกิดขึ้นเนื่องจากการผลิตสินค้าต่าง ๆ จะต้องแข่งขันกันทั้งทางด้านปริมาณและคุณภาพ ดังนั้นค่าผลได้ (YIELD) จึงมีความจำเป็นที่จะต้องทราบให้รวดเร็วเท่าที่จะทำได้ เพื่อให้โรงงานได้ทราบสถานการณ์ของการผลิตว่าได้ผลดี มากน้อยแค่ไหนเพื่อจะได้แก้ไขปัญหาต่าง ๆ ซึ่งอาจจะเนื่องมาจากวัตถุหรือขบวนการผลิตของโรงงานต่าง ๆ จึงมีการหาค่าผลได้ (YIELD) ขึ้น แต่ขบวนการหาค่าผลได้นี้ก็นำไปได้อย่างลำบาก โดยใช้กำลังคนในการเก็บข้อมูลต่าง ๆ จากนั้นก็มาทำการวิเคราะห์ซึ่งกว่าจะรู้ผลของค่าผลได้ (YIELD) นี้ว่าดีหรือไม่ดี การผลิตสินค้าต่าง ๆ ก็ผ่านพ้นไปแล้วถ้าค่าผลได้ (YIELD) ต่ำ โรงงานก็จะเสียผลประโยชน์ไปอย่างมหาศาลเนื่องจากไม่สามารถจะแก้ไขขบวนการผลิตได้ การที่ปล่อยในการผลิตผลิตสินค้าผ่านไป 8 ชม. แล้วจึงทราบว่าผลได้ (YIELD) ต่ำควรจะเปลี่ยนแปลงขบวนการผลิตนั้น ค่าเสียหายที่เกิดขึ้นกับบริษัทบางครั้งก็มีค่าเป็น 10 ล้านบาท โรงงานอุตสาหกรรมต่าง ๆ จึงพยายามที่จะหาวิธีการที่จะหาค่าผลได้ (YIELD) นี้ให้

ได้เร็วที่สุดเท่าที่จะทำได้ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือวัดผลได้ทางอุตสาหกรรม (YIELDOMETER) เป็นเครื่องมือทางอิเล็กทรอนิกส์ ที่ใช้ระบบทางไมโครโปรเซสเซอร์เข้าทำงาน โดยมีโปรแกรมที่จะเป็นตัวรวบรวมข้อมูลต่าง ๆ เช่น จำนวนผลผลิต , จำนวนผลผลิตที่เสียไป , สาเหตุของการเสียของผลผลิต , ฯลฯ ของแต่ละหน่วยที่อยู่ในขบวนการผลิต แล้วแสดงให้ผู้ควบคุมทราบผลในแต่ละหน่วยได้ในทันทีขณะที่การผลิตดำเนินอยู่ นอกจากนี้ยังเชื่อมต่อกับระหว่างหน่วยต่าง ๆ ที่เกี่ยวข้องและรวบรวมข้อมูลต่าง ๆ ของแต่ละหน่วยมาทำการวิเคราะห์ เพื่อหาผลได้ (YIELD) ของขบวนการผลิตแล้วแสดงให้ผู้ควบคุมหรือเกี่ยวข้องทราบในลักษณะที่เป็น REAL TIME หรือ ปัจจุบันทันที เพื่อให้ผู้ควบคุมได้ทราบผลได้ (YIELD) ของขบวนการผลิตหรือหน่วยย่อยได้ในทันทีทันใดเพื่อแก้ไข ในกรณีที่ผลได้ (YIELD) ตกลงมาต่ำกว่าที่ทางบริษัทตั้งเป้าไว้ นอกจากนี้เครื่องนี้ยังสามารถเก็บข้อมูลต่าง ๆ ไว้เพื่อทำการวิเคราะห์ค่าผลได้ (YIELD) เป็นรายวัน , สัปดาห์ , หรือ รายเดือน ก็ย่อมได้ และสามารถจะบอกจุดบกพร่องของการผลิตว่าจุดใดหรือหน่วยใดทำให้ผลผลิตเสียมากน้อยเพียงไร และด้วยสาเหตุใด

จะเห็นได้ว่า การศึกษาและสร้างเครื่องมือวัดผลได้ทางอุตสาหกรรม (YIELDOMETER) จะเป็นประโยชน์อย่างสูงในอุตสาหกรรมต่าง ๆ จะเป็นการประหยัดเงินตราที่จะต้องสูญเสียไปในความผิดพลาดของการผลิต ซึ่งเป็นสิ่งที่ไม่จำเป็นและเป็นประโยชน์อย่างสูงต่อประเทศไทย ซึ่งมีโรงงานอุตสาหกรรมที่มีความจำเป็นต้องใช้เครื่องมืออยู่อย่างมากมาย งานวิจัยนี้จึงมีใช้เพื่อการศึกษาเท่านั้น หากแต่ว่ามีประโยชน์ต่อประเทศชาติเป็นอย่างมากด้วย

วัตถุประสงค์

- เพื่อศึกษาและสร้างเครื่องมือวัดผลได้ทางอุตสาหกรรม (YIELDOMETER)
- เผยแพร่และนำเครื่องมือวัดผลได้ทางอุตสาหกรรมไปใช้งานกับโรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา แด-2 ของอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ที่คาดว่าจะได้รับ

- สร้างเครื่องมือวัดผลได้ทางอุตสาหกรรม
- ทำให้ประหยัดค่าใช้จ่ายอย่างมากมาใช้ในการทำอุตสาหกรรมต่าง ๆ
- เป็นการพัฒนาเครื่องมือทางวิทยาศาสตร์เพื่อช่วยเหลืองานด้านอุตสาหกรรม
- เป็นการประยุกต์ใช้ระบบไมโครโปรเซสเซอร์ เพื่อใช้ในเครื่องมือทาง

อุตสาหกรรม

วิธีดำเนินงานโดยย่อ

จากข้างต้นทำให้ทราบว่าการทำงานทั้งหมดของระบบจะขึ้นอยู่กับระบบไมโครโปรเซสเซอร์ และการรับส่งข้อมูลระหว่างไมโครโปรเซสเซอร์ด้วยกันเอง ดังนั้นการดำเนินงานโดยย่อจึงแบ่งขั้นตอนดังนี้

1. ศึกษาไมโครโปรเซสเซอร์ Z80180 ร่วมกับอุปกรณ์ภายนอก เช่น ลวิตซ์ , LCD (LIQUID CRYSTAL DISPLAY) ฯลฯ พร้อมกับการพัฒนาโปรแกรมให้เหมาะสมกับข้อมูลต่าง ๆ ที่ต้องการแสดงและประมวลผล ซึ่งอุปกรณ์ทั้งหมดในชุดนี้จะเรียกว่า ชุดแสดงผลย่อย

2. ศึกษาการรับส่งข้อมูลระยะไกล โดยใช้ไอซีประเภท OPTO ISOLATOR

3. ศึกษาไมโครโปรเซสเซอร์ Z80 CP-A พร้อมกับการพัฒนาโปรแกรมให้สอดคล้องกับงานที่ใช้ ซึ่งอุปกรณ์ทั้งหมดในชุดนี้จะเรียกว่า ชุดแสดงผลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำอุปกรณ์ทั้งหมดมาประกอบเข้าด้วยกัน โดยให้ชุดแสดงผลย่อยสามารถส่ง
ข้อมูลต่าง ๆ ที่เกี่ยวข้องกับค่าผลได้ ไปยังชุดแสดงผลหลักเพื่อนำไปประมวลผลต่อไป
ซึ่งในชุดควบคุมทั้งสองนี้จะมีภาคแสดงผลแสดงให้ผู้ใช้งานสามารถดูได้ตลอดเวลา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ระบบอาร์คแวร์ของเครื่องวัดผลได้ทางอุตสาหกรรม

ภายในระบบอาร์คแวร์ของเครื่องนี้ประกอบด้วยส่วนต่างๆ 3 ส่วน คือ ชุดแสดงผลหลัก (MAIN-STATION) ชุดแสดงผลย่อย (SUB-STATION) และ วงจรรับส่งข้อมูล (CURRENT-LOOP) ซึ่งทั้งสามส่วนนี้สามารถแสดงให้เห็นโดยแผนภาพอย่างง่ายดังรูปที่ 2.1

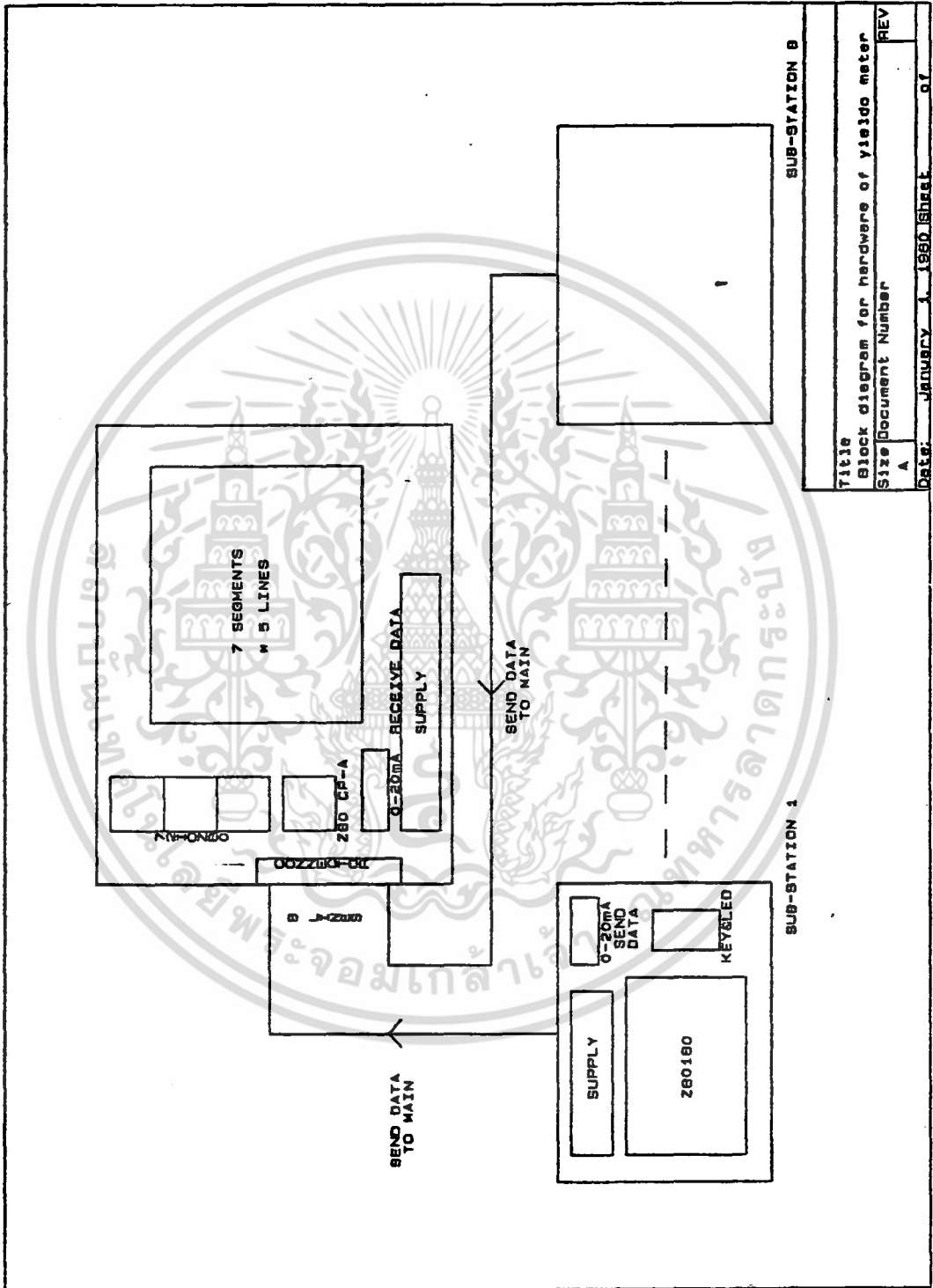
จากรูปจะเห็นว่า ระบบไมโครโปรเซสเซอร์เป็นหัวใจในการทำงานของเครื่อง โดยในชุดแสดงผลหลักจะใช้แผ่นวงจรเดี่ยว Z80 CP-A และชุดแสดงผลย่อยจะใช้แผ่นวงจรเดี่ยว Z80180 ดังนั้นจะขอกล่าวถึงระบบไมโครโปรเซสเซอร์เบื้องต้นพอสังเขป

ระบบไมโครโปรเซสเซอร์

ไมโครโปรเซสเซอร์

Z-80 ซีพียู นับว่าเป็นส่วนสำคัญที่สุดในการทำงานของระบบซึ่งทำหน้าที่คำนวณทางคณิตศาสตร์ ปฏิบัติการทางลอจิก สร้างสัญญาณควบคุมจากขาสัญญาณต่างๆ ไมโครโปรเซสเซอร์เพียงตัวเดียวจึงไม่สามารถทำงานเป็นระบบคอมพิวเตอร์ได้ ต้องอาศัยอุปกรณ์ส่วนสำคัญอีก 2 ส่วน คือ หน่วยความจำ (Memory) และหน่วยรับส่งข้อมูลเข้าออก (I/O Devices) โดยซีพียูจะทำงานตามคำสั่งในหน่วยความจำซึ่งถูกเรียงไว้เป็นลำดับ ซึ่งเรียกว่า โปรแกรม คำสั่งจะมีลักษณะเป็นเลขฐานสอง เมื่อซีพียูรับ (Fetch) คำสั่งจากหน่วยความจำ ซีพียูจะแปลความหมายของคำสั่งและทำงาน (Execute) ตามคำสั่งนั้น จะทำให้เกิดการควบคุมระบบภายนอกจากบัสควบคุม (Control Bus)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในวงการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

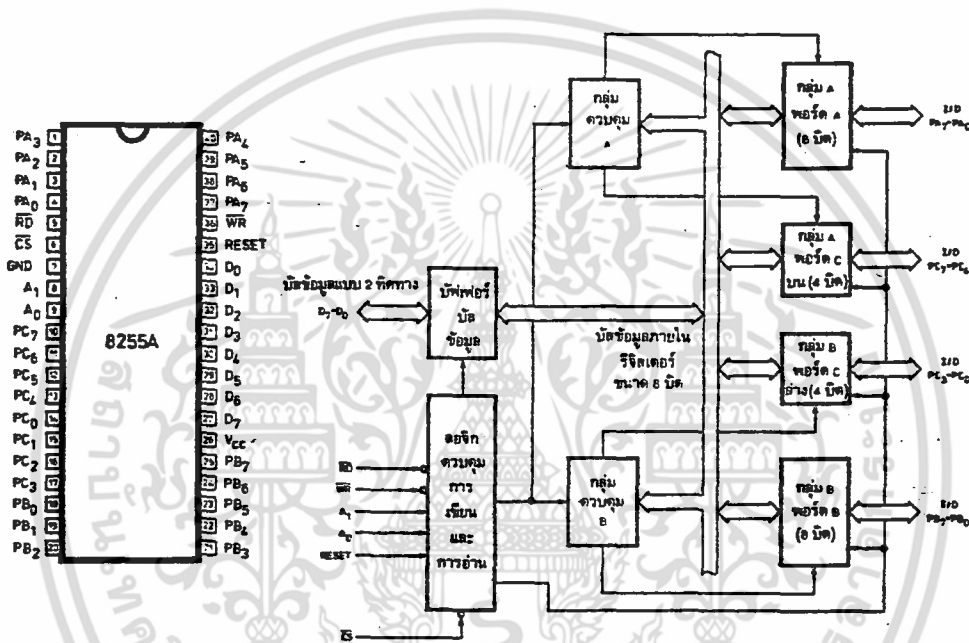


เอกสารนี้เป็นเอกสารที่วางไว้สำหรับใช้เฉพาะในโครงการวิจัยเท่านั้น การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การคัดลอกหรือการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

รูปที่ 2.1 แสดงโครงสร้างอย่างง่ายของเครื่องวัดผลได้ทางอุตสาหกรรม

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตต่างๆว่า พอร์ต A พอร์ต B และ พอร์ต C โดยพอร์ต C แยกออกเป็น 2 ส่วน คือ พอร์ต C ล่าง หรือ ตั้งแต่ PC0-PC3 มีจำนวน 4 บิต และ พอร์ต C บน หรือ ตั้งแต่ PC4-PC7 ที่มีเศษคือ พอร์ตทุกพอร์ตเป็นได้ทั้งพอร์ตอินพุตและเอาต์พุต



รูปที่ 2.3 แสดงแผนผังวงจรภายในและการจัดการของไอซี 8255

จากรูปที่ 2.3 แสดงการทำงานของ 8255 ซึ่งสามารถแบ่งการทำงานออกได้เป็นกลุ่ม 3 กลุ่ม คือ

- ก) กลุ่มของพอร์ต พอร์ต A (PA0-PA7), พอร์ต B (PB0-PB7) และ พอร์ต C ล่าง กับ C บน (PC0-PC3 และ PC4-PC7) ซึ่งเป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกอื่นๆ ทำหน้าที่ส่งผ่านข้อมูลระหว่างอุปกรณ์ภายนอกกับ 8255 และแต่ละพอร์ตมีหลายสัญญาณเชื่อมเข้ากับบัสข้อมูลภายใน 8255

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยศูนย์วิจัยและพัฒนาเทคโนโลยีสารสนเทศเพื่อการพัฒนาที่ยั่งยืน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข) กลุ่มควบคุม แบ่งเป็นกลุ่มควบคุม A และ กลุ่มควบคุม B เป็นตัวกำหนดลักษณะการทำงานของทั้ง 3 พอร์ต

ค) กลุ่มบัฟเฟอร์และสัญญาณควบคุม Data Buffer เป็นบัฟเฟอร์ให้กับบัลข้อมูลของ CPU และ Read/Write Control Logic ซึ่งเป็นลatches ที่ควบคุมให้ข้อมูลเข้าหรือออกจากรีจิสเตอร์ภายในตัวที่ถูกต้องและในเวลาที่เหมาะสม

รายละเอียดการจัดขาของ 8255

8255 มีลักษณะเป็นตัวถัง 40 ขาแบบดิพ แต่ละขามีหน้าที่แตกต่างกันออกไป ซึ่งมีรายละเอียดดังต่อไปนี้

DO-D7 (DATA BUS) เป็นสายข้อมูลอินพุต/เอาต์พุต แบบ 2 ทิศทาง จะเป็นทางผ่านของข้อมูลระหว่างพอร์ตต่างๆของ 8255 กับบัลข้อมูลของ Z-80

CS (Chip Select Input) เมื่อขานี้มีสถานะลอจิกเป็น 0 ซีพียูสามารถที่จะอ่านหรือเขียนข้อมูลกับ 8255 ได้

RD (Read Input) เมื่อขานี้และขา CS มีสถานะลอจิกเป็น 0 ข้อมูลจาก 8255 จะเข้าสู่บัลข้อมูล

WR (Write Input) เมื่อขานี้และขา CS มีสถานะลอจิกเป็น 0 ข้อมูลจากบัลข้อมูล จะเข้าสู่ 8255

AO-A1 (ADDRESS INPUT) จะเป็นตัวกำหนดการเลือกใช้รีจิสเตอร์ภายในของ 8255

RESET เมื่อขานี้มีสถานะเป็น 1 8255 จะอยู่ในสภาวะรีเซ็ต

PA0-PA7 , PBO-PB7 ขาสัญญาณเหล่านี้จะถูกใช้เพื่อเป็นพอร์ต I/O ขนาด 8 บิต ใช้ต่อเข้ากับอุปกรณ์ภายนอกอื่นๆ

PC0-PC7 ใช้เป็นพอร์ต I/O ขนาด 8 บิต และสามารถแบ่งออกเป็น 2 กลุ่ม โดยแต่ละกลุ่มมีขนาด 4 บิต กลุ่มแรกใช้ควบคุม PBO-PB7 และกลุ่มที่สอง

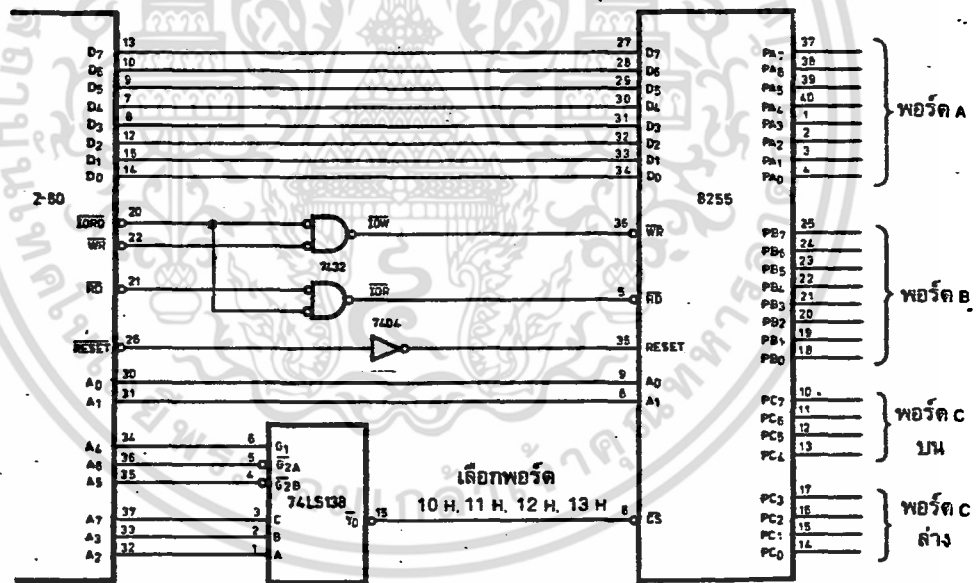
ใช้ควบคุม PA0-PA7 สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อ 8255 เข้ากับระบบ Z-80

ในการต่อระบบ 8255 เข้ากับระบบ Z-80 นั้น ต้องนำสัญญาณ A0-A7 จาก Z-80 มาถอดรหัส เพื่อสร้างสัญญาณเลือกพอร์ท และเพราะว่า 8255 มีขาแอดเดรสอินพุตอยู่แล้ว 2 ขา (A0-A1) จะนำต่อตรงเข้ากับ A0, A1 จากแอดเดรสบัล หมายความว่า 8255 หนึ่งตัวจะใช้ค่าพอร์ทแอดเดรสถึง 4 ค่า (2^2) ส่วนสัญญาณอีก 6 เส้น (A2-A7) จะนำไปถอดรหัสเพื่อสร้างสัญญาณเลือกชิพ (CS: CHIP SELECT) ให้แก่ 8255

สมมติว่าต้องการให้ Z-80 มองเห็น 8255 เป็นพอร์ทหมายเลข 10H, 11H, 12H และ 13H การเชื่อมต่อ 8255 เข้ากับระบบของ Z-80 จะเป็นดังรูป 2.4



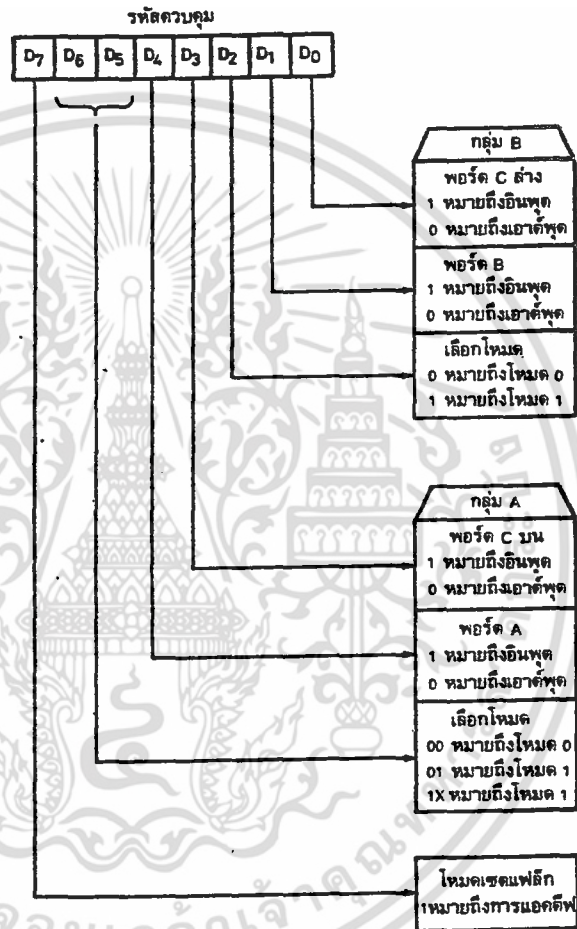
รูปที่ 2.4 การเชื่อมต่อ 8255 เข้ากับระบบของ Z-80

ลักษณะการทำงานของ 8255 แบ่งออกเป็น 3 โหมด แต่ละโหมดมีความแตกต่างกันตามความเหมาะสมในการเลือกใช้งาน ในการเซต 8255 ให้อยู่ใน

โหมดใดนั้น จะสั่งได้จากคำสั่งควบคุม แบบที่นิยมใช้จะเป็นโหมด 0 ใช้ประโยชน์ด้านการค้าไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในโหมด 0

การกำหนดโหมดการทำงาน จะต้องส่งข้อมูลเข้าไปในโปรแกรมในพอร์ทควบคุมของ 8255 แต่ละบิตที่ส่งไปจะมีความหมายในตัวเอง ลักษณะความหมายของแต่ละบิตในรหัสควบคุมแสดงดังรูปที่ 2.5 และมีรายละเอียดดังนี้



รูปที่ 2.5 แสดงรายละเอียดแต่ละบิตของคำสั่งควบคุม ของ 8255

ตัวอย่างคำสั่งของ Z-80 ในการเซต 8255 เป็นดังนี้

LD A, 80H ; เซตคำสั่งควบคุม

OUT (13H), A ; ส่งคำสั่งควบคุมให้ 8255 ที่พอร์ทควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Z-80 ทำงานตามคำสั่งข้างต้นแล้ว 8255 จะถูกเซตให้ทุกพอร์ท เป็นเอาต์พุตและอยู่ในโหมด 0 ซึ่งสามารถส่งข้อมูลไปยังพอร์ทต่างๆได้ด้วยคำสั่ง OUT ของ Z-80 เช่น

LD A,99H ; กำหนดข้อมูล

OUT (10H),A ; ส่งค่า 99H ไปที่พอร์ท A (10H)

LD A,8AH ; กำหนดข้อมูล

OUT (11H),A ; ส่งค่า 8AH ไปที่พอร์ท B (11H)

การทำงานในโหมด 1

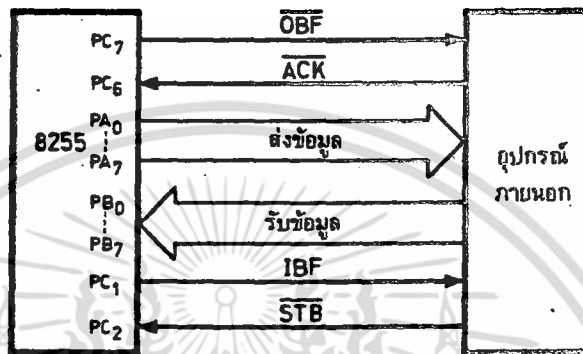
การทำงานของ 8255 ในโหมด 1 เป็นโหมดที่ทำให้อินพุต/เอาต์พุตมีการตรวจสอบสัญญาณ (Handshaking) โดยใช้อินพุต/เอาต์พุตของพอร์ท A และ B เป็นหลัก และใช้พอร์ท C บน เป็นตัวตรวจสอบสัญญาณ (Handshake) ของพอร์ท A ส่วนพอร์ท C ล่างเป็นตัวตรวจสอบสัญญาณของพอร์ท B การใช้พอร์ทอินพุต/เอาต์พุตโดยมีการตรวจสอบสัญญาณ เพื่อให้มีการซิงโครไนซ์ระหว่างอุปกรณ์ภายนอกที่ทำงานได้เข้า เช่น เครื่องพิมพ์ ลักษณะของการรับส่งข้อมูลแบบนี้จะใช้ PA0-PA7 เป็นเอาต์พุตและ PB0-PB7 เป็นอินพุต โดยมีพอร์ท C เป็นตัวตรวจสอบสัญญาณ ดังรูปที่ 2.6

เมื่อโปรแกรม 8255 ให้เป็นโหมด 1 แล้ว ตัว 8255 จะให้พอร์ท C เป็นสัญญาณควบคุม โดยแต่ละบิตของพอร์ท C เป็นไปตามที่กำหนดไว้ดังตารางที่ 2.1

การทำงานในโหมด 2

ในการกำหนดโหมดนี้ 8255 จะใช้พอร์ท A เป็นพอร์ทแบบสองทิศทางคือสามารถเป็นได้ทั้งพอร์ทอินพุตและพอร์ทเอาต์พุต โดยโครงสร้างของพอร์ท A ทั้งอินพุต/เอาต์พุตตัวตรวจสอบสัญญาณทั้งคู่ โดยใช้พอร์ท C โดยมีสัญญาณแต่ละขา ดังตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 วงจรการต่อ 8255 ในโหมด 1

ขา	กรณีอินพุต	กรณีเอาต์พุต
PC0	INTR _B	INTR _B
PC1	IBF _B	\overline{OBF}_B
PC2	\overline{STB}_B	\overline{ACK}_B
PC3	INTR _A	INTR _A
PC4	\overline{STB}_A	I/O
PC5	IBF _A	I/O
PC6	I/O	\overline{ACK}_A
PC7	I/O	\overline{OBF}_A

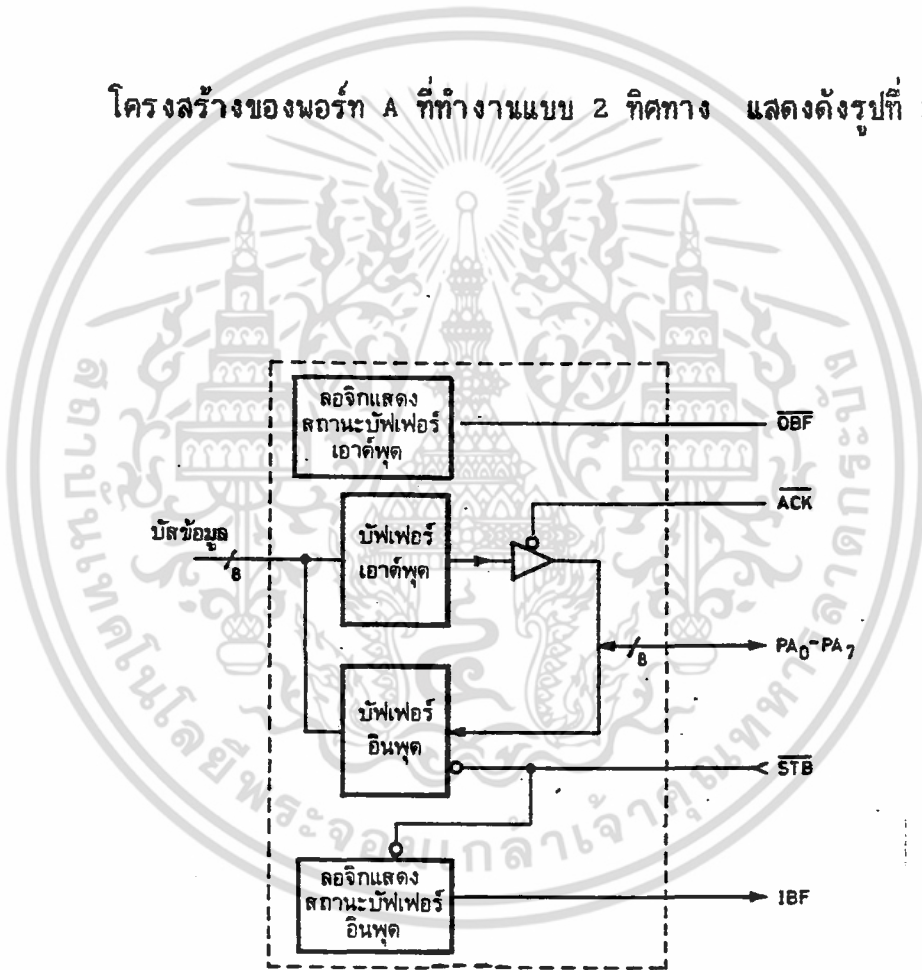
ตารางที่ 2.1 หน้าทีของพอร์ต C ในโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต C	ความหมาย
PC ₀	IO
PC ₁	IO
PC ₂	IO
PC ₃	INTR _A
PC ₄	$\overline{\text{STB}}_A$
PC ₅	IBF _A
PC ₆	$\overline{\text{ACK}}_A$
PC ₇	$\overline{\text{OBF}}_A$

ตารางที่ 2.2 หน้าทีของพอร์ต C ในโหมด 2

โครงสร้างของพอร์ต A ที่ทำงานแบบ 2 ทิศทาง แสดงดังรูปที่ 2.7



รูปที่ 2.7 โครงสร้างของพอร์ต A ที่ทำงานแบบสองทิศทาง

เมื่อโปรแกรมพอร์ต A เป็นโหมด 2 แล้ว พอร์ต B จะต้องโปรแกรม

เอกสารนี้เป็นโหมด 0 หรือโหมด 1 ก็ได้ ซึ่งก็ทำงานแบบแยกอิสระให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากระบบไมโครโปรเซสเซอร์นี้ เป็นส่วนสำคัญในการใช้งานเกี่ยวกับเครื่อง YEILDO METER ซึ่งส่วนต่างๆของเครื่องนี้จะได้กล่าวอย่างละเอียดดังนี้



2.1) ชุดแสดงผลหลัก (MAIN-STATION)

โครงสร้างของส่วนชุดแสดงผลหลักจะมีลักษณะดังรูปที่ 2.8

จากแผนภาพมีอุปกรณ์ประกอบกันอยู่หลายส่วน คือ ระบบควบคุมด้วยไมโครโปรเซสเซอร์ Z80 CP-A ส่วนขยาย I/O PORT(7210Z80) ส่วนแสดงผล(7 SEGMENT) ส่วนรับข้อมูล ส่วนป้อนแรงดันไฟฟ้า แต่ส่วนจะมีหน้าที่แตกต่างกันออกไป ซึ่งจะกล่าวถึงรายละเอียดและความสำคัญของส่วนต่างๆดังนี้

2.1.1) ระบบควบคุมด้วยไมโครโปรเซสเซอร์ Z80 CP-A

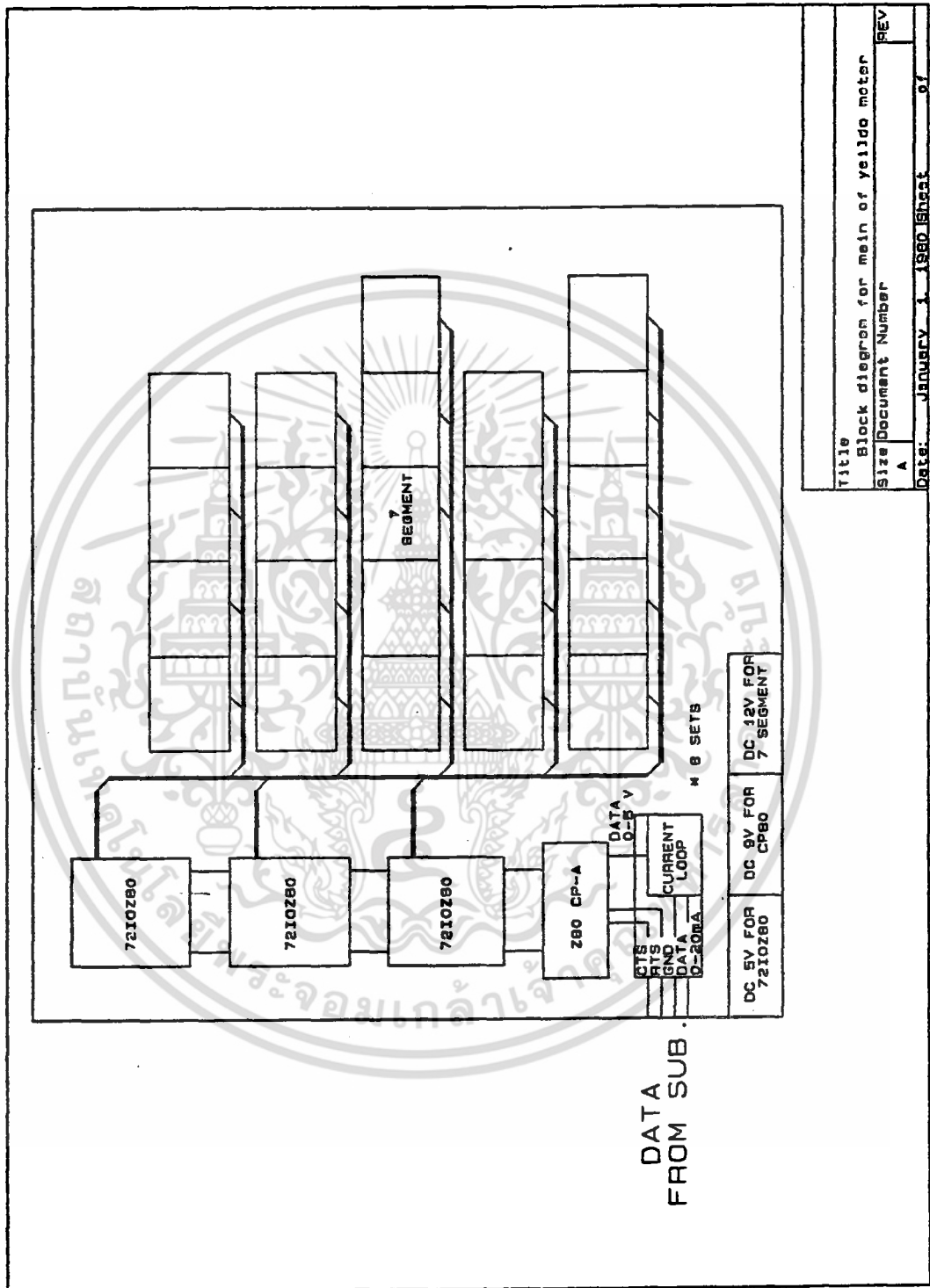
แผ่นวงจรเดี่ยว Z80 CP-A ที่ใช้เป็นตัวควบคุมของระบบ โดยมี Z80A เป็นชิพโดยที่ชิพนี้จะเชื่อมต่อเข้ากับ 7210Z80 (รายละเอียดอยู่ในหัวข้อ 2.1.2) จำนวน 3 ชุด ส่วนทาง I/O PORT ของแผ่นวงจรเดี่ยวนี้นำไปต่อเข้ากับข้อมูล ซึ่งได้รับมาจากชุดแสดงผลย่อย คือ

- PORT A จะเป็น INPUT PORT รับข้อมูลจากขา RTS (RESET TO SEND) ของ Z80180 ซึ่งเป็นชิพของชุดแสดงผลย่อยทั้ง 8 บิต คือ PA0-PA7 (จะใช้ 1 บิตต่อชุดแสดงผลย่อย 1 ชุด)

- PORT B จะเป็น OUTPUT PORT ส่งข้อมูลให้กับขา CTS (CLEAR TO SEND) ของ Z80180 ซึ่งเป็นชิพของชุดแสดงผลย่อยทั้ง 8 บิต คือ PBO-PB7 (จะใช้ 1 บิตต่อชุดแสดงผลย่อย 1 ชุด)

- PORT C จะเป็น INPUT PORT รับข้อมูลจากขา TXAO (TRANSMIT DATA 0) ของ Z80180 ซึ่งผ่านชุดรับส่งข้อมูลทั้ง 8 บิต คือ PC0-PC7 (จะใช้ 1 บิตต่อชุดแสดงผลย่อย 1 ชุด)

เอกสารนี้เป็นเอกสารที่วางไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงโครงสร้างของชุดแสดงผลหลัก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่อผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ I/O PORT ของ Z80 CP-A จะใช้เมื่อต้องการเพิ่มชุดแสดงผลย่อย จาก 8 ชุด เป็น 16 ชุด เพราะการควบคุม I/O PORT ของ 8 ชุดแรก ได้ตั้งไว้ที่ พอร์ตของ 7210Z80

ที่มิพียูจะมีการต่อเข้ากับปุ่ม CLEAR ที่ขา NON-MARKBLE INTERRUPT/NMI เพื่อใช้ในการลบล้างข้อมูลทั้งหมดในการเริ่มงานใหม่ และมีปุ่มรีเซตเพื่อใช้เมื่อต้องการให้ระบบเริ่มต้นใหม่ เมื่อเครื่องเกิดการ HANG โดยไม่ต้องการให้ข้อมูลเก่าสูญหาย

2.1.2) ส่วนขยาย I/O PORT (7210Z80)

เนื่องจากการใช้งานในส่วนอินพุตและเอาต์พุตของชุดแสดงผลหลักต้องใช้เป็นจำนวนมาก ดังนั้น I/O PORT ภายในของ Z80 CP-A จึงไม่เพียงพอในการใช้สำหรับส่วนแสดงผลของชุดแสดงผลหลัก จึงต้องมีชุดขยาย I/O PORT (7210Z80) เพิ่มขึ้นอีก 3 ชุด (เพิ่ม 8255 รวม 9 ตัว) โดยมีมิพียูของ Z 80 CP-A เป็นตัวควบคุมทุกชุด

การใช้งานของ 8255 ส่วนมากจะใช้กับการแสดงผลของ 7 SEGMENT มีเพียง 2 ตัว ตัวแรกใช้กับ ปุ่ม (TOGGLE SWITCH), สวิตช์ (SWITCH ON/OFF) และ DIP SWITCH ดังรูปที่ 2.10

ประโยชน์ของปุ่มซึ่งประกอบไปด้วย 3 ปุ่ม คือ

- สีเขียว ใช้ในการเพิ่มเปอร์เซ็นต์เป้าหมาย (TARGET YEILD)

- สีแดง ใช้ในการลดเปอร์เซ็นต์เป้าหมาย

- สีเหลือง ใช้ในการลบล้างข้อมูลเดิม เพื่อให้เครื่องเริ่มทำงานใหม่

ส่วนสวิตช์หนึ่งตัวซึ่งเป็นชนิดเปิด/ปิด จะใช้เป็นการเลือกแบบการคำนวณ

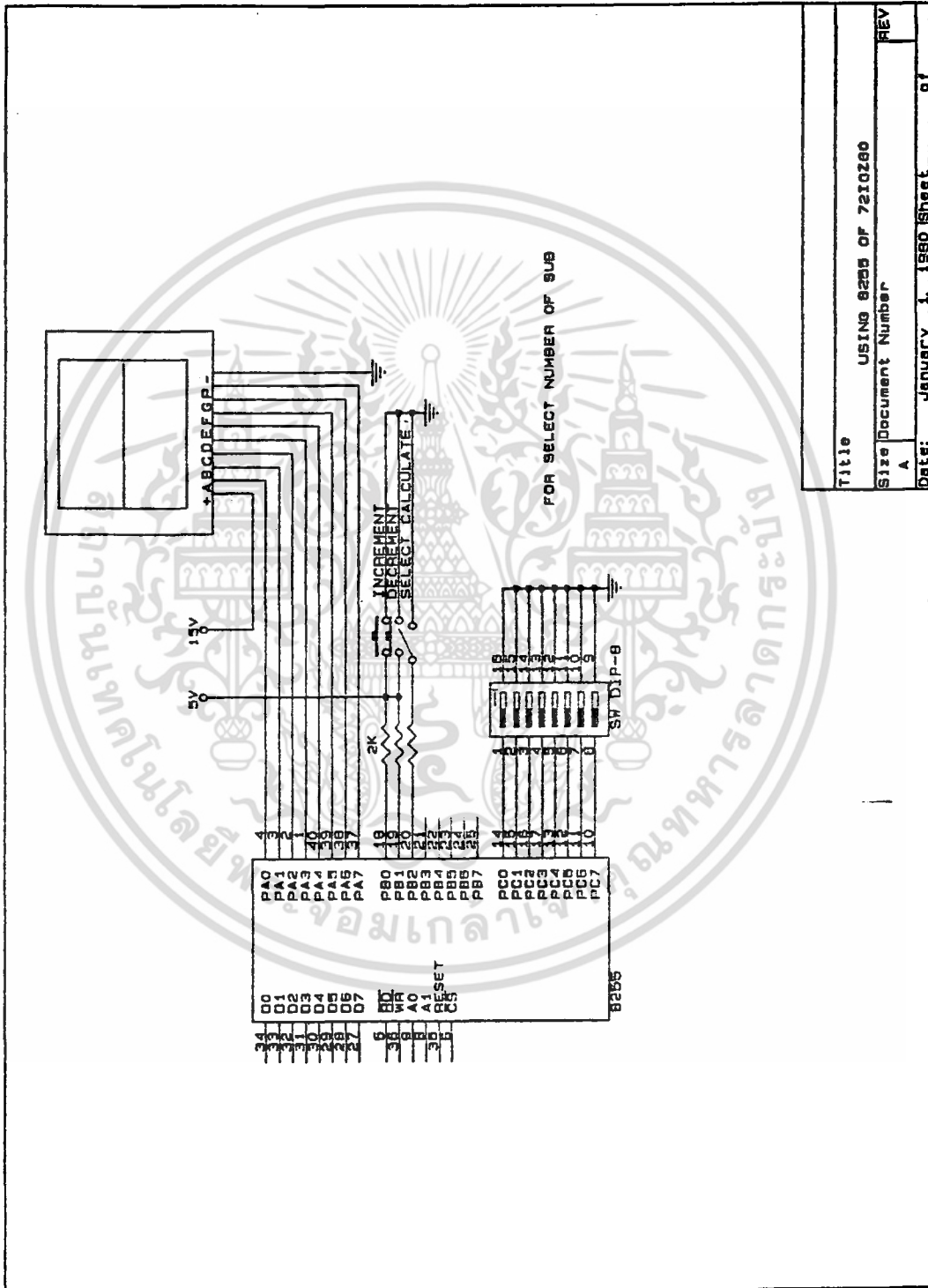
ซึ่งสามารถเลือกได้ 2 แบบ (รายละเอียดการคำนวณจะอยู่ในบทที่ 4)

DIP SWITCH มีหน้าที่ในการเลือกใช้จำนวนชุดแสดงผลย่อย ซึ่งสามารถเลือกใช้ได้สูงสุดถึง 8 ชุดพร้อมกัน ถ้าต้องการใช้ไม่ครบตามจำนวน 8 ชุด

เอกสารนี้เปลี่ยนแปลง DIP SWITCH ON ให้เป็น OFF (รายละเอียดอยู่ในบทที่ 4) ข้อประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนอีกตัวหนึ่งใช้ทำหน้าที่เหมือนกับ I/O PORT ของ Z80 CP-A ในหัวข้อ

2.1.1 ซึ่งใช้ในการควบคุมหลายข้อมูลของชุดแปลงผลย่อย 8 ชุดแรก ลักษณะ การต่อใช้งานจะเป็นดังรูปที่ 2.9



Title	USING 8255 OF 7210Z80
Size/Document Number	A
REV	
Date:	January 1, 1980
of	01

รูปที่ 2.10 แสดงการใช้งานของ 8255 ของ 7210Z80 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การรังสรรค์ขึ้นเพื่อใช้เท่านั้น เมื่อผู้ผู้ใดเห็นใบโฆษณาหรือโฆษณาการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3) ส่วนแสดงผล (7 SEGMENT)

ส่วนนี้จะใช้งานร่วมกับ 7210Z80 โดยมีไฟเลี้ยง 15 Volt ดังรูปที่ 2.10 นอกจากนี้ 3 ส่วนนี้แล้วยังมีส่วนของการรับข้อมูลซึ่งจะกล่าวในหัวข้อ 2.3

2.2) ชุดแสดงผลย่อย (SUB-STATION)

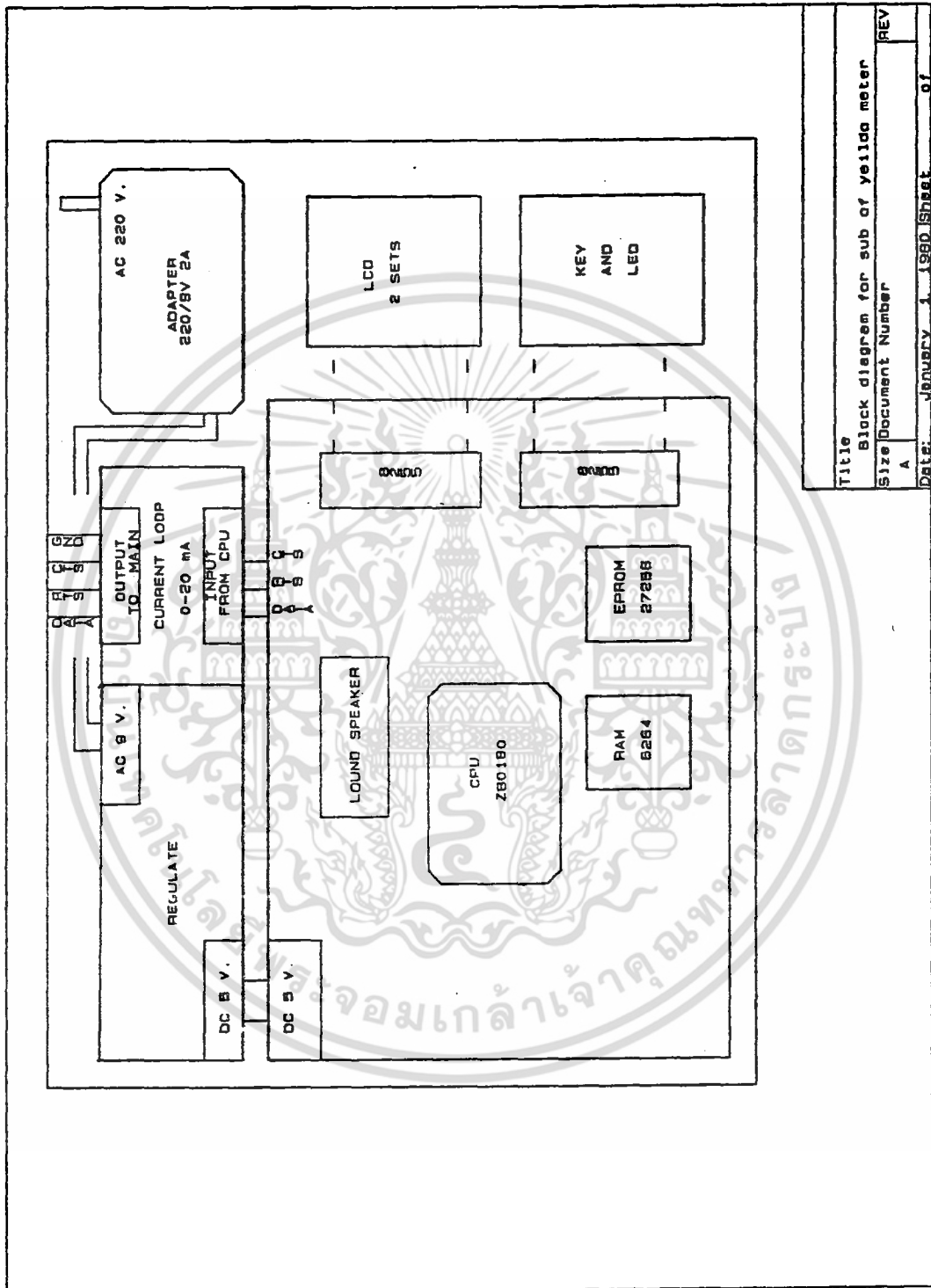
โครงสร้างภายในของชุดแสดงผลย่อยแสดงดังรูปที่ 2.11

จากแผนภาพมีอุปกรณ์ประกอบกันอยู่หลายส่วนและแต่ละส่วนจะมีหน้าที่แตกต่างกันออกไป ซึ่งจะกล่าวถึงรายละเอียดและความสำคัญของส่วนต่างๆดังนี้

2.2.1) ระบบควบคุมด้วยไมโครโปรเซสเซอร์ Z80180

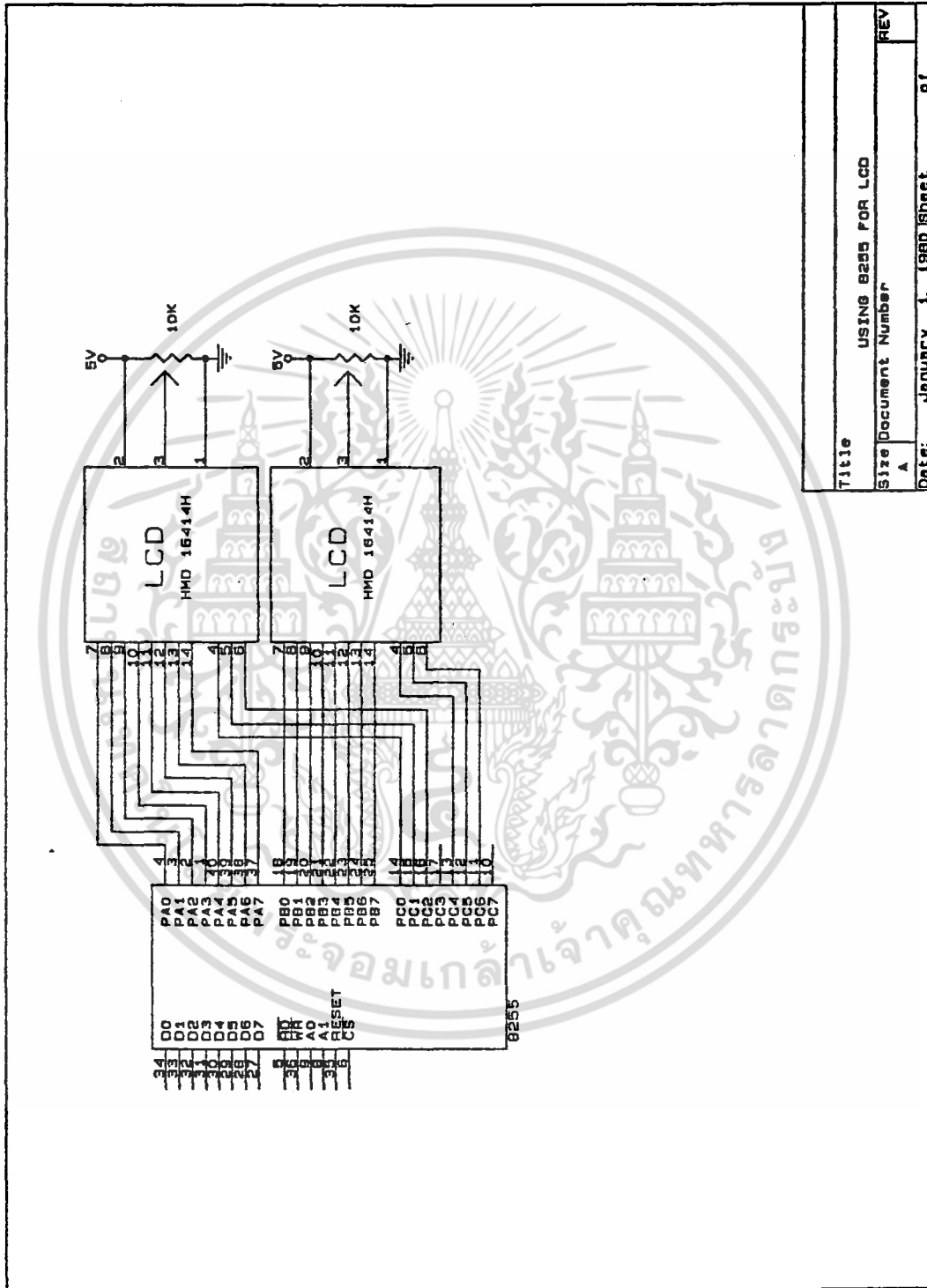
แผ่นวงจรเดี่ยว Z80180 เป็นตัวควบคุมของระบบ แผ่นวงจรเดี่ยวนี้นประกอบไปด้วย ซีพียู Z80180 EPROM27256 RAM6264 8255 ฯลฯ ในระบบฮาร์ดแวร์ของชุดแสดงผลย่อยได้นำส่วนต่างๆของแผ่นวงจรเดี่ยว Z80180 ไปใช้งานดังนี้

- 8255 ภายในแผ่นวงจรเดี่ยวนี้นประกอบด้วย 8255 ถึง 2 ตัว ตัวแรกใช้ในการควบคุมส่วนแสดงผลซึ่งใช้ LCD (LIQUID CRYSTAL DISPLAY) 2 ชุด ส่วนตัวที่สองใช้กับปุ่มที่เป็นอินพุตกับ LED ที่ใช้แสดงสถานะของปุ่มนั้นๆ โดยการใช้การต่อวงจรแบบ PULL DOWN ดังรูปที่ 2.12 และ 2.13



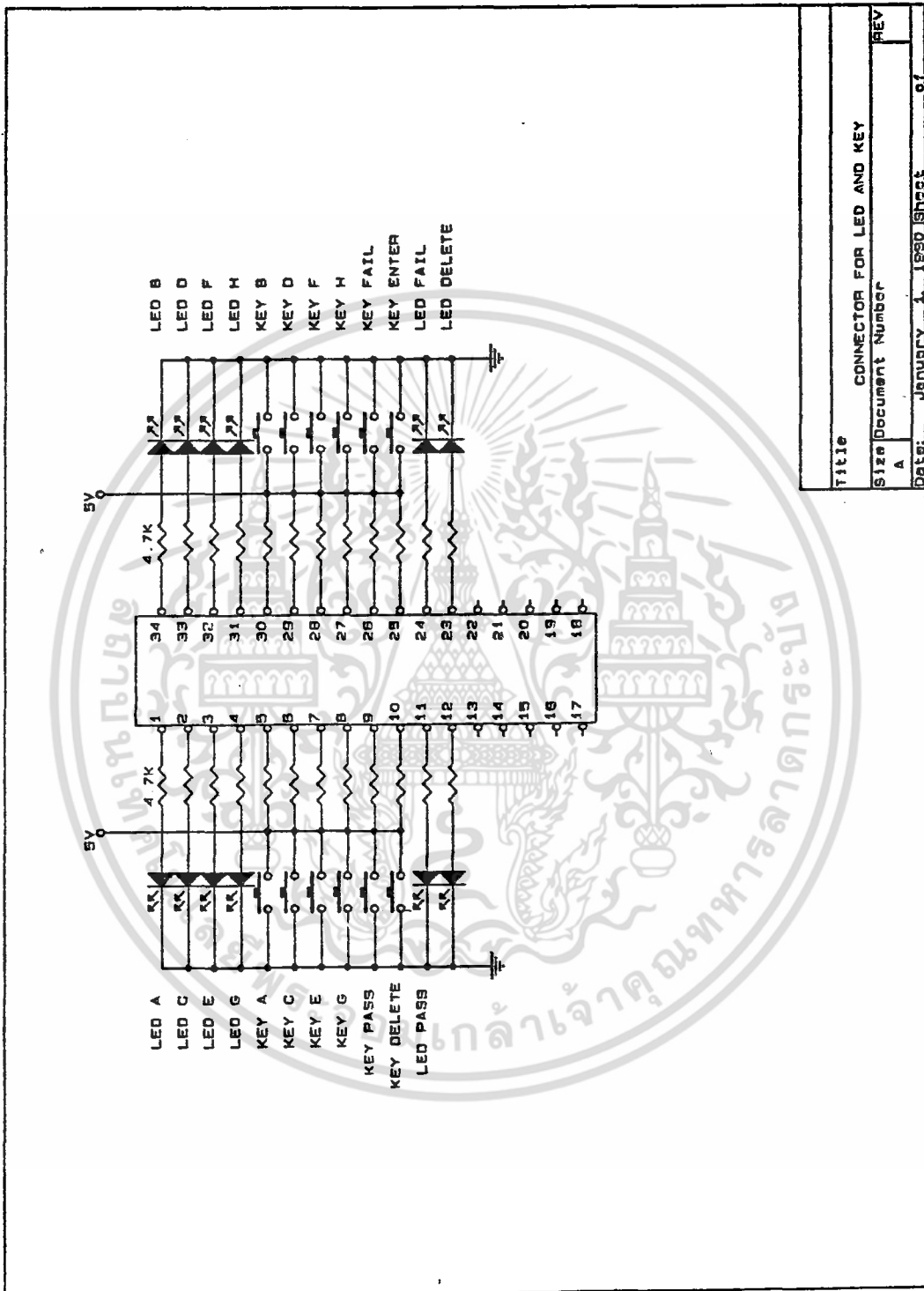
รูปที่ 2.11 แสดงโครงสร้างภายในของชุดแสดงผลย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title	USING 8255 FOR LCD
Size	Document Number
REV	A
Date:	January 1, 1990 (Sat)
	07

รูปที่ 2.12 แสดงการเชื่อมต่อระหว่าง 8255 กับ LCD
 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title	CONNECTOR FOR LED AND KEY
Size	Document Number
A	REV
Date:	January 1, 1990 (Srac)
	01

รูปที่ 2.13 แสดงการเชื่อมต่อระหว่าง 8255 กับ ปุ่มอินพุตและLED แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

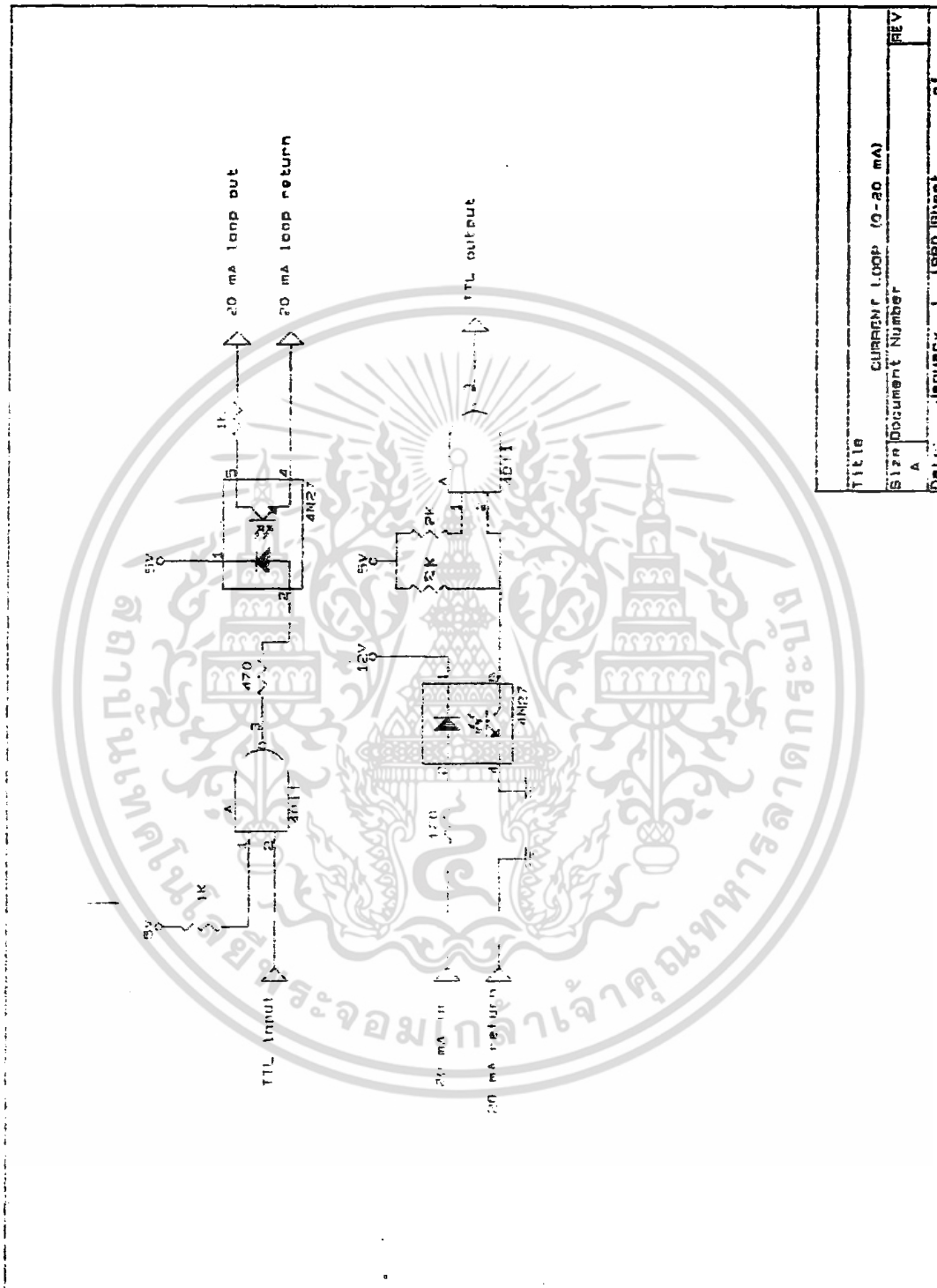
- ซีพียู Z80180 จะใช้ขา 3 ขาของซีพียูในการส่งข้อมูล โดยที่ขา CTS, RTS ต่อตรงเข้ากับ 8255 ของแผ่นวงจรเดี่ยว Z80 CP-A ของชุดแสดงผลหลัก ส่วนอีกขาหนึ่งคือ TXAO ซีพียูจะส่งข้อมูลมาทางขานี้ (โดยการควบคุมทางโปรแกรม) ผ่านชุดส่งข้อมูลไปยัง 8255 ของแผ่นวงจรเดี่ยวของชุดแสดงผลหลัก และอีกขาหนึ่งจะใช้ในการส่งสัญญาณให้กับลำโพงคือ TOUT (TIMER OUT) เทียบกับ GND ดังรูปที่ 2.12

ภายในแผ่นวงจรเดี่ยว Z80180 นั้นจะมีการต่อปุ่ม CLEAR และ RESET เพื่อทำหน้าที่เหมือนกับปุ่มทั้งสองของแผ่นวงจรเดี่ยว Z80 CP-A

ส่วนที่เหลือคือส่วนของการส่งข้อมูลจะกล่าวในหัวข้อ 2.3

2.3) วงจรรับส่งข้อมูล (CURRENT LOOP 0-20 mA)

วงจรที่ใช้รับส่งข้อมูลชุดนี้จะใช้ในระยยะไกล โดยอาศัยหลักการเปลี่ยนแรงดันไฟฟ้าไปเป็นกระแส เพราะถ้าส่งแรงดันไฟฟ้าไปตามสายไฟในระยยะไกลมาก (มากกว่า 50 ฟุต) จะมีการสูญเสียของสัญญาณเนื่องจากสัญญาณรบกวน ความต้านทานของสายไฟ ฯลฯ จึงต้องอาศัยหลักการของ CURRENT LOOP 0-20 mA ดังรูปที่ 2.14



TITLE	CURRENT LOOP (0-20 mA)
SYM	Document Number
REV	A
DATE	JANUARY 1, 1980 BUREAU

รูปที่ 2.14 (ขบ) แสดงการแปลงสัญญาณ TTL ไปเป็น 0-20 mA

(ล่าง) แสดงการแปลงสัญญาณกลับจาก 0-20 mA ไปเป็น TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น เมื่ออนุญาตเห็นไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระบบโปรแกรมควบคุม

เครื่องวัดผลได้ทางอุตสาหกรรม มีระบบโปรแกรมควบคุมหลักสำคัญในการทำงานของเครื่อง ซึ่งระบบนี้จะช่วยให้ระบบฮาร์ดแวร์สามารถแสดงผลให้ชุดแสดงผลย่อยรับข้อมูลจากอินพุทภายนอก (ปุ่มการใช้งานต่างๆ) และช่วยในการรับส่งข้อมูล บทนี้จะกล่าวถึงระบบโปรแกรมควบคุม ในลักษณะของแผนภาพอย่างง่าย (BLOCK DIAGRAM) ส่วนรายละเอียดของ FLOW CHART และ โปรแกรมทั้งหมดจะอยู่ในส่วนของภาคผนวก

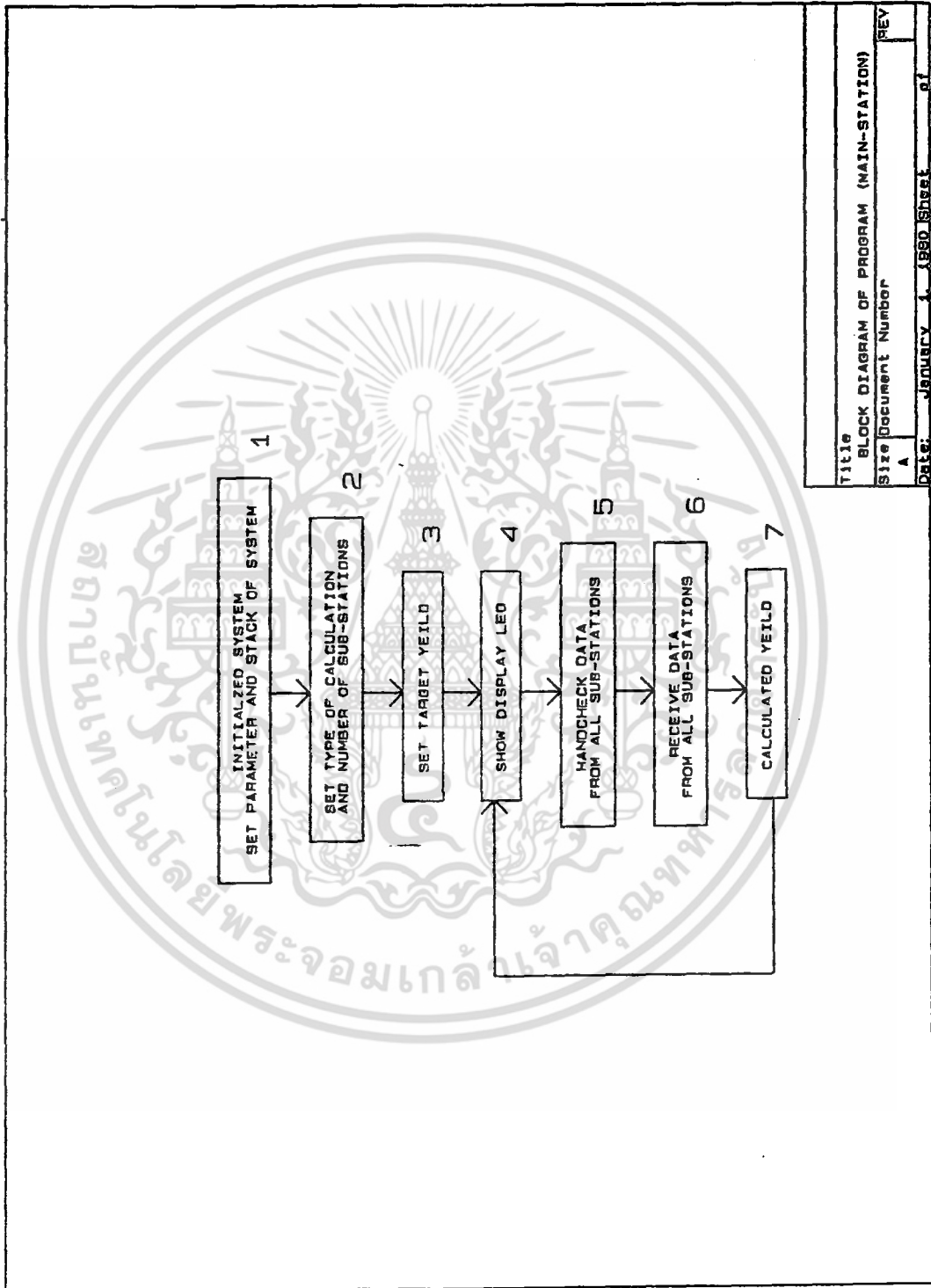
แผนภาพอย่างง่ายของระบบโปรแกรมควบคุมแบ่งเป็น 2 ส่วน คือ

3.1) ระบบโปรแกรมควบคุมของชุดแสดงผลหลัก

แผนภาพอย่างง่ายแสดงการทำงานของระบบโปรแกรมควบคุมของชุดแสดงผลหลักเป็นดังรูปที่ 3.1

ภายในแผนภาพจะบอกลำดับการทำงานของโปรแกรมที่ใช้ควบคุมและแสดงผล ซึ่งในแต่ละขั้นตอนการทำงานมีดังนี้

- 1) เป็นการตั้งค่าพารามิเตอร์และจองหน่วยความจำเพื่อใช้ในการเก็บข้อมูลต่างๆที่ได้รับเข้ามาเพื่อประมวลผล



รูปที่ 3.1 แสดงลักษณะโครงสร้างการทำงานของชุดแสดงผลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การตั้งแบบของการคำนวณและจำนวนของชุดแสดงผลย่อย ซึ่งการตั้งค่านี้จะใช้ระบบฮาร์ดแวร์เป็นตัวช่วยคือ ในส่วนของแบบการคำนวณจะใช้สวิทช์เปิดปิดเพื่อเป็นสัญญาณ 0 หรือ 1 ให้ระบบโปรแกรมควบคุมนำไปประมวลผลว่าเป็นการคำนวณแบบไหน (วิธีการคำนวณอยู่ในบทที่ 4) ในส่วนจำนวนของชุดควบคุมและแสดงผลย่อยก็ใช้การเปิดปิดของ DIP SWITCH เป็นอินพุตป้อนเข้าไปทาง 8255 เหมือนการเลือกการคำนวณ และนำข้อมูลมาประมวลผลในระบบโปรแกรมควบคุม

3) การตั้งค่าของผลได้ทางอุตสาหกรรม เป็นการแสดงค่าที่ตั้งไว้คือ 85.0 ออกทาง 7 SEGMENT ซึ่งระบบโปรแกรมควบคุมจะสามารถเปลี่ยนแปลงค่านี้ได้จากการเปลี่ยนแปลงอินพุตภายนอก คือ TOGGLE SWITCH ในการให้ลดหรือเพิ่มค่าได้ตลอดเวลาของการทำงาน

4) การนำเอาค่าต่างๆที่อยู่ใน BUFFER มาแสดงทาง 7 SEGMENT

5) การตรวจว่าชุดแสดงผลหลัก และ ชุดแสดงผลย่อยอยู่ในสถานะที่พร้อมจะทำการรับ-ส่งข้อมูลซึ่งกันและกัน โดยจะอาศัยสัญญาณจากขา RTS0 และ CTS0 ของ ชุดแสดงผลย่อย ช่วยในการตรวจสอบ

6) การรับข้อมูลจากชุดแสดงผลย่อย เมื่อชุดแสดงผลหลัก และ ชุดแสดงผลย่อย อยู่ในสถานะพร้อม (หลังจากผ่านการตรวจสอบ ในข้อ 5 เรียบร้อยแล้ว)

ขณะทำงาน ในข้อ 5 และข้อ 6 จะกระทำกันชุดแสดงผลย่อยทีละตัว กล่าวคือ เมื่อทำการตรวจสอบสถานะพร้อม (ข้อ 5) ของ ชุดแสดงผลย่อย ตัวที่หนึ่งเรียบร้อยแล้ว ชุดแสดงผลหลักก็จะทำการรับข้อมูลจากชุดแสดงผลย่อย (ข้อ 6) ตัวที่หนึ่งจนเสร็จแล้วจึงจะทำการตรวจสอบ และรับข้อมูลจากชุดแสดงผลย่อยตัวต่อไปจนครบชุดแสดงผลย่อยทุกตัว จึงทำการตรวจสอบสถานะโดยการวนรอบใหม่

7) นำค่าที่อยู่ใน BUFFER ที่ส่งมาจากค่า BUFFER ในชุดแสดงผลย่อย มาทำการคำนวณค่าผลได้ทางอุตสาหกรรม

หลังจากการทำงานของโปรแกรมถึงข้อ 7 จะมีการวน LOOP การทำ

เอกสารในงานในข้อ 4 สอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2) ระบบโปรแกรมควบคุมของชุดแสดงผลย่อย

แผนภาพอย่างง่ายแสดงการทำงานของระบบโปรแกรมควบคุมของชุดแสดงผลย่อยเป็นดังรูปที่ 3.2

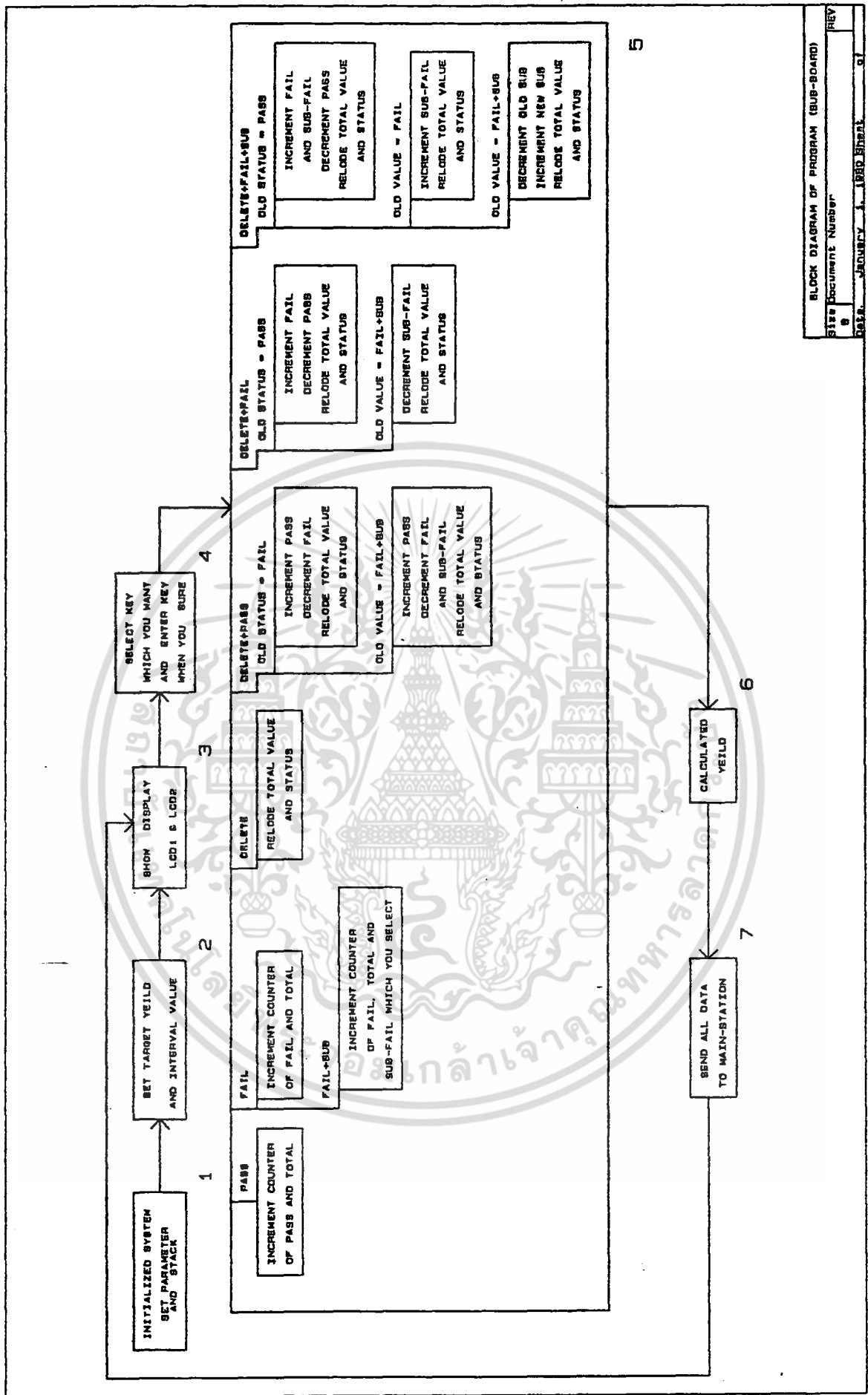
ภายในแผนภาพจะบอกลำดับการทำงานของโปรแกรมที่ใช้ควบคุมและแสดงผลซึ่งในแต่ละขั้นตอนการทำงานมีดังนี้

- 1) เป็นการตั้งค่าพารามิเตอร์และจองหน่วยความจำเพื่อใช้ในการเก็บข้อมูลต่างๆที่ได้รับเข้ามาเพื่อประมวลผล
- 2) การตั้งค่าเป้าหมายของผลได้ทางอุตสาหกรรม (TARGET YEILD) และช่วงของจำนวนที่กำหนด (INTERVAL VALUE) จะทำให้โปรแกรมรู้ขอบเขตของการทำงาน ในการใส่ค่าเหล่านี้จะใส่หลังจอ LCD สั่งว่าให้ใส่ข้อมูลอะไรเข้าไปก่อน ซึ่งการใส่ข้อมูลเข้าไปในระบบโปรแกรมควบคุมจะสั่งการให้งานในปุ่มต่างๆดังนี้
 - ใส่ค่าเป้าหมายของผลได้ทางอุตสาหกรรม โดยใช้ปุ่ม PASS ในการเพิ่มค่า ปุ่ม FAIL ในการลดค่า และ ปุ่ม ENTER สำหรับใช้ในการตกลงว่าเลือกค่าเป้าหมายที่ปรากฏบนจอ LCD
 - ใส่ค่าช่วงของจำนวนที่กำหนด จะใช้ปุ่มในลักษณะเดียวกัน

ซึ่งการใส่ค่าเป้าหมายของผลได้ทางอุตสาหกรรมจะเสร็จสิ้นก่อน จึงจะมีการใส่ค่าช่วงของจำนวนที่กำหนด ซึ่งข้อมูลทั้งสองอยู่ที่จอเดียวกันแต่แสดงทางจอภาพต่างเวลา

- 3) หลังจากการตั้งค่าในข้อ 2 แล้ว จะมีโปรแกรมจัดค่าต่างๆที่ได้กำหนดไว้ในโปรแกรมแสดงออกมาทางจอ LCD ซึ่งรายละเอียดการจัดหน้าจอ LCD ทั้ง 2 จออยู่ในบทที่ 4 ซึ่งแสดงว่าเครื่องพร้อมที่จะรับข้อมูลการตรวจผลผลิตทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงลักษณะโครงสร้างการทำงานของชุดแสดงผลย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลระบบใช้เอกสารนี้ในการดำเนินการ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เมื่อกดปุ่มต่างๆ ระบบโปรแกรมควบคุมจะได้รับข้อมูลทุกครั้งที่เกิด และจะนำข้อมูลต่างๆใน BUFFER เหล่านั้นไปประมวลผล

5) ในการประมวลผลของข้อมูลในแต่ละลักษณะที่ได้รับจากปุ่มต่างๆ มี ลักษณะโครงสร้างแตกต่างกันดังนี้

- PASS จะทำการเพิ่มค่าที่เก็บไว้ใน BUFFER ที่จองไว้สำหรับ ปุ่มนี้ทีละหนึ่งหลังจากการกดหนึ่งครั้ง และจะทำการเพิ่มค่าใน BUFFER ของ TOTAL ที่จองไว้ทีละหนึ่งเช่นเดียวกัน

- FAIL จะทำการเพิ่มค่าที่เก็บไว้ใน BUFFER ที่จองไว้สำหรับ ปุ่มนี้ทีละหนึ่งหลังจากการกดหนึ่งครั้ง และจะทำการเพิ่มค่าใน BUFFER ของ TOTAL ที่จองไว้ทีละหนึ่งเช่นเดียวกันในกรณีที่ไม่มีสาเหตุของผลิตภัณฑ์ที่เสียกำหนดไว้ แต่ถ้ามี กรณีที่มีสาเหตุของผลิตภัณฑ์ที่เสียกำหนดไว้ให้กด ก็จะเพิ่มค่าที่เก็บไว้ใน BUFFER ที่ จองไว้สำหรับสาเหตุการเสียของผลิตภัณฑ์ในแต่ละกรณี (มีให้เลือกได้ 8 กรณี) ทีละหนึ่งค่า

- DELETE จะทำการแสดงสถานะของการเลือกเมื่อครั้งก่อนหน้านี้ หนึ่งค่าต่อการกดปุ่มหนึ่งครั้ง โดยจะทำการเก็บค่าใน BUFFER ของ TOTAL และ สถานะปัจจุบันไว้ใน BUFFER สำรอง แต่ถ้ามีการกดปุ่ม PASS หรือ FAIL และตามด้วยสาเหตุของการเสีย เครื่องก็จะทำการเพิ่มค่าใน BUFFER ของปุ่ม นั้นๆ และทำการลดค่าใน BUFFER ของสถานะที่ถูกเปลี่ยนแปลงไป โดยไม่มีการเพิ่มค่า ใน BUFFER ของ TOTAL หลังจากนั้นจะคืนค่าที่เก็บไว้ใน BUFFER สำรอง ให้กับ BUFFER ของ TOTAL และ BUFFER สถานะ นี้คือลักษณะการทำงานของ DELETE ทั้งหมดในเครื่อง

6) เป็นส่วนที่นำข้อมูลทั้งหมดภายใน BUFFER มาประมวลผลในสูตรการคำนวณของการหาผลได้ทางอุตสาหกรรม

7) ส่งข้อมูลใน BUFFER ของ PASS, FAIL, และ TOTAL ไปที่ชุด แสดงผลหลักเพื่อนำไปประมวลผลและแสดงผลรวมกับชุดแสดงผลย่อยอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนี้จะมีการวน LOOP เพื่อไปแสดงผลข้อมูลต่างๆที่มีอยู่ใน BUFFER
ที่ข้อ 3 และจะทำงานในลักษณะนี้เรื่อยไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การวิจัยและการดำเนินการ

การวิจัยและการดำเนินการแบ่งแต่ละส่วนเป็นขั้นตอนตามลำดับดังต่อไปนี้

4.1) ส่วนที่ 1 การศึกษาและเพิ่มเติมอุปกรณ์ในชุดแสดงผลย่อย

ขั้นตอนของการดำเนินการของส่วนชุดแสดงผลย่อย เป็นการศึกษาส่วนของฮาร์ดแวร์พร้อมกับโปรแกรมควบคุม ดังขั้นตอนต่อไปนี้

1) ศึกษาแผ่นวงจรเดี่ยว Z80180 ซึ่งมีชิพ Z80180 ในขั้นแรกจะใช้ DEBUGGER ในการทดลองเพื่อพัฒนาโปรแกรมและการใช้โปรแกรมในการควบคุมระบบฮาร์ดแวร์ เช่น การควบคุม I/O PORT การ INTERRUPT ฯลฯ

2) นำ LCD มาประกอบร่วมกับแผ่นวงจรเดี่ยว Z80180 ใช้โปรแกรมควบคุมโดยให้แสดงอักษรต่างๆที่ต้องการ โดยให้หน้าจอของ LCD ทั้ง 2 จอเป็นดังรูปที่ 4.1 และเปลี่ยนการพัฒนาจาก DEBUGGER มาใช้ EPROMULATE แทน

3) นำเอาปุ่มมาประกอบร่วมกับแผ่นวงจรเดี่ยว Z80180 ใช้โปรแกรมควบคุมปุ่มโดยให้เพิ่มหรือลดทีละ 1 ค่า แล้วทดสอบกับ LCD โดยให้ค่าที่กดปุ่มแสดงบน LCD ในตำแหน่งที่ต้องการ ดังรูปที่ 4.2

การนำปุ่มต่างๆมาประกอบให้เหมาะสมกับการใช้งานมีดังต่อไปนี้

3.1) ปุ่ม PASS ใช้ในการเพิ่มจำนวนของหรือผลิตภัณฑ์ที่ผ่านการตรวจสอบว่าถูกต้องทีละหนึ่ง

3.2) ปุ่ม FAIL ใช้ในการลดจำนวนของหรือผลิตภัณฑ์ที่ผ่านการตรวจสอบว่าไม่ถูกต้องทีละหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3) ปุ่ม ENTER ใช้ในการตกลงว่าข้อมูลที่กดก่อนหน้าถูกต้อง

3.4) ปุ่ม DELETE ใช้ในการแก้ไขข้อมูลเมื่อเกิดความผิดพลาดและต้อง

การแก้ไข

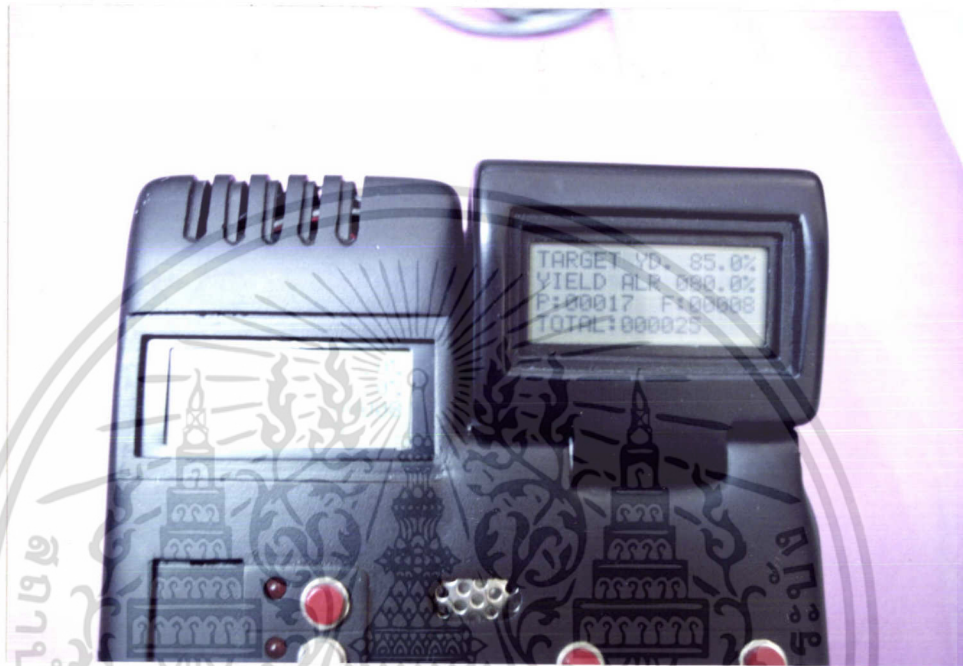
3.5) ปุ่มลาเหตุ 8 ปุ่ม ใช้หลังจากกดปุ่ม FAIL แล้วเลือกสาเหตุของความผิดพลาดของผลิตภัณฑ์ 8 แบบ

ซึ่งการใช้งานของปุ่มเหล่านี้จะกล่าวอย่างละเอียด ในเรื่องของการทำงานของวัดผลได้ทางอุตสาหกรรม



รูปที่ 4.1 แสดงรูปแบบหน้าจอของ LCD ทั้ง 2 จอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงรูปแบบหน้าจอของ LCD ตามการควบคุมของปุ่มต่างๆ

4) นำเอา LED มาประกอบเพื่อแสดงให้เห็นว่ากำลังกดปุ่มใดอยู่ในขณะนี้-
หลังจากที่ทำตามขั้นตอนทั้ง 4 จะได้ส่วนของชุดแสดงผลย่อยและจัดทำ

ชุดป้อนแรงดันไฟฟ้ากระแสตรง 5 โวลต์ 2 แอมป์

จากรูปด้านบนแสดงว่าต้องทำโปรแกรมแสดงผลข้อมูลต่างๆบนจอ LCD การรับ
ปุ่มอินพุท และส่วนสำคัญอีกส่วนคือ การคำนวณค่าผลได้ทางอุตสาหกรรม คือค่า
YIELD COMMUTATIVE และ YIELD ALERT

จากค่า YIELD COMMUTATIVE และ YIELD ALERT จะมีการคำนวณ
แตกต่างกันออกไป เพื่อสามารถนำไปประมวลผลการวัดผลได้ทางอุตสาหกรรมได้

เอกสารนี้ 21 แบบสาร การคำนวณเพื่อให้ได้ค่าทั้งลองนี้จึงมีวิธีต่างกัน หลักการคำนวณทั้ง 21 แบบมีดังนี้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) YIELD COMMUTATIVE คือ การคิดคำนวณผลิตรถยนต์ที่เพิ่มขึ้นที่ผ่านการตรวจสอบ โดยใช้สูตรคำนวณเทียบเปอร์เซ็นต์พื้นฐานดังนี้

$$\text{YIELD COMMUTATIVE (\%)} = \frac{\text{จำนวนผลิตรถยนต์ที่ผ่านการตรวจสอบ} * 100}{\text{จำนวนผลรวมทั้งหมดของผลิตรถยนต์ที่เข้าตรวจสอบ}}$$

2) YIELD ALERT คือ การคิดคำนวณผลิตรถยนต์เพียงช่วงของจำนวนที่กำหนดไว้ โดยก่อนที่เครื่องเริ่มทำการประมวลผลข้อมูลอินพุตเข้ามา จะให้มีการตั้งค่าช่วงของจำนวน (INTERVAL) ซึ่งสามารถตั้งได้ตั้งแต่ 10, 20, 30, ..., 250 ซึ่งจะนำค่านี้มาประมวลผลตามสูตร

$$\text{YIELD ALERT (\%)} = \frac{\text{จำนวนผลิตรถยนต์ที่ผ่านการตรวจสอบในช่วงของจำนวน} * 100}{\text{จำนวนผลรวมทั้งหมดของผลิตรถยนต์ที่เข้าตรวจสอบในช่วงของจำนวน}}$$

การคำนวณในแบบนี้จะไม่เกิดขึ้นในตอนเริ่มตรวจสอบผลิตรถยนต์ จนกว่าจำนวนผลิตรถยนต์ที่ผ่านการตรวจสอบจะมีค่าเท่ากับช่วงของจำนวนที่กำหนด หลังจากนั้นการคำนวณจะเกิดขึ้นทุกค่าที่ใส่ค่าข้อมูลเข้าไป โดยการคำนวณจะตัดค่าแรกสุดออกจากการคำนวณ และนำค่าใหม่ล่าสุดที่ป้อนให้กับเครื่องมาร่วมในการคำนวณแทน ตัวอย่างเช่น

ตั้งค่าช่วงของจำนวนที่กำหนด = 10

ค่าที่ป้อนให้กับเครื่อง ค่าที่ 1-9 เป็น PASS

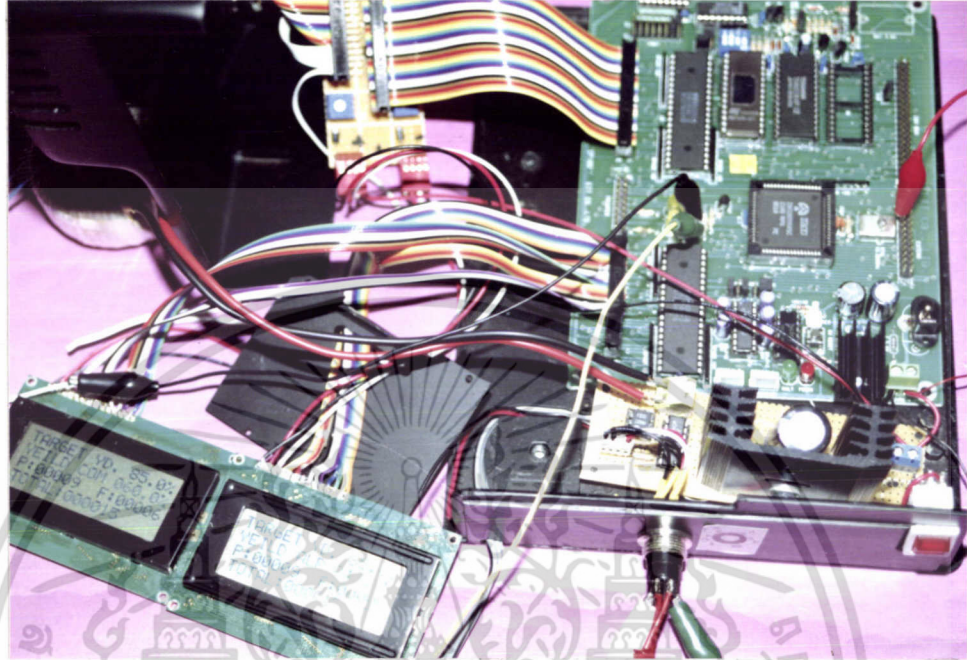
ค่าที่ 10 เป็น FAIL

หลังจากที่ป้อนค่าที่ 10 แล้ว เครื่องจะเริ่มคำนวณเป็น

$$(9 * 100) / 10 = 90\%$$

ถ้ามีการป้อนค่าที่ 11 เป็น FAIL เครื่องจะคำนวณเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาดูงานเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อ (8*100) / 10 ไปถึง 80% เอกสารทุกครั้งที่มีการนำไปใช้

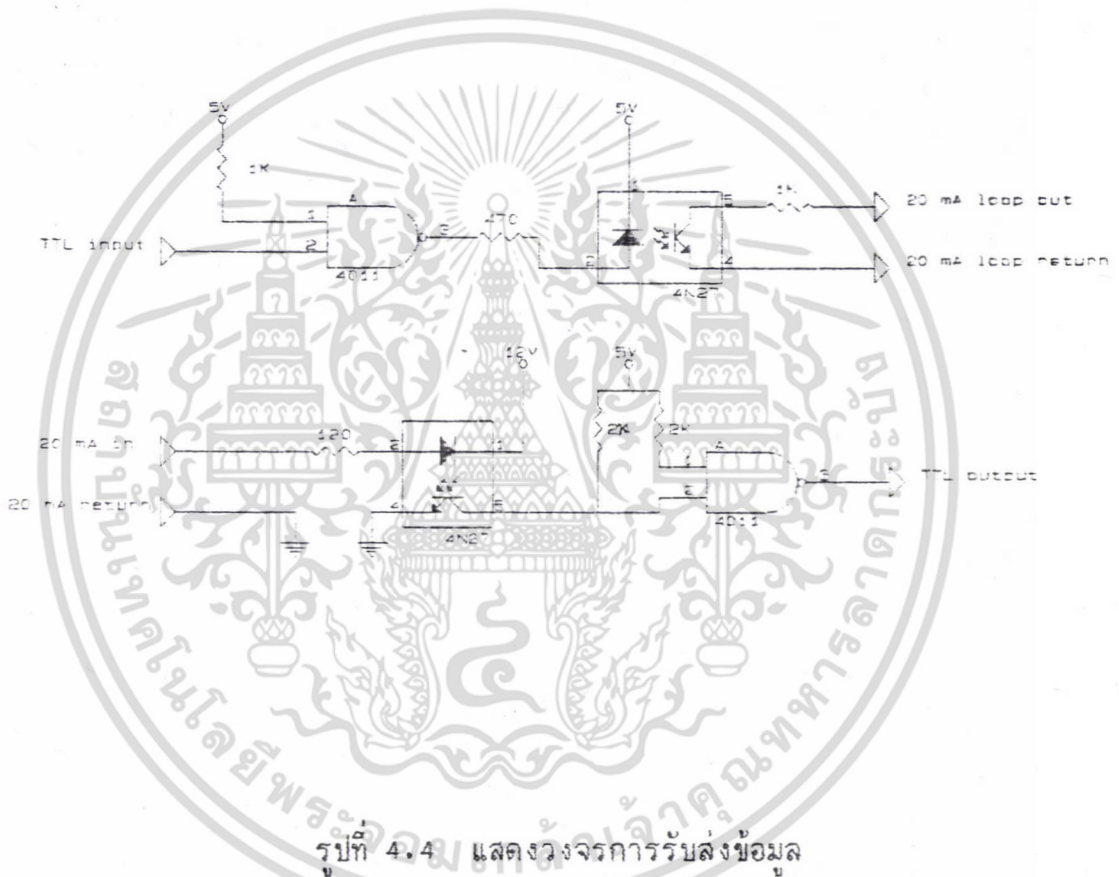


รูปที่ 4.3 แสดงการต่ออุปกรณ์ภายในของชุดแสดงผลย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2) ส่วนที่ 2 การศึกษาวงจรรับส่งข้อมูล

การศึกษาในส่วนนี้ได้ใช้ไอซี OPTO ISOLATOR เบอร์ 4N27 มาเป็นตัวแปลงสัญญาณ TTL เป็น 0-20 mA ดังรูปที่ 4.4



รูปที่ 4.4 แสดงวงจรการรับส่งข้อมูล

ขั้นตอนในการทดลองการส่งสัญญาณเป็นดังนี้

1) นำฟังก์ชันเจนเนอเรเตอร์มาป้อนสัญญาณคลื่นรูปสี่เหลี่ยม (SQUARE WAVE) แทนสัญญาณอินพุท TTL

2) นำ OSCILLOSCOPE ช่องแรกมาจับสัญญาณที่ขาอินพุทจะได้คลื่นรูปสี่เหลี่ยมระดับ 5 โวลต์ นำช่องที่สองมาจับที่เอาต์พุทของ NAND GATE ตัวแรกจะได้

สัญญาณขนาดเท่ากันแต่กลับเฟส

3) นำ OSCILLOSCOPE ช่องสองมาจับที่ขา 5 ของ 4N27 จะได้

ขนาดของ TIM/DIV เท่ากับช่องแรก ส่วน VOLT/DIV จะลดลงอยู่ในระดับมิลลิโวลต์ และสัญญาณยังกลับเฟลอยู่เช่นเดิม

4) นำ OSCILLOSCOPE ช่องสองมาจับขาเอาท์พุทของ NAND GATE ตัวที่สอง จะได้สัญญาณเหมือนกับช่องแรก

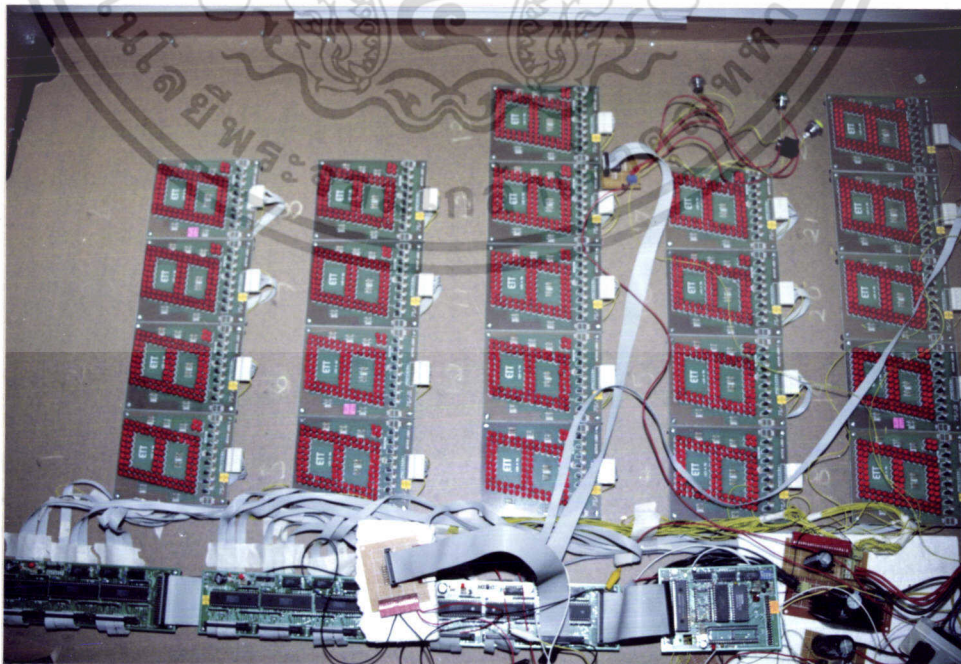
จากขั้นตอนดังกล่าวแสดงว่าได้ข้อมูลส่งมาแบบ CURRENT LOOP (0-20 mA) ในข้อ 3 และแปลงกลับมาเป็น TTL เพื่อป้อนให้กลับชุดแสดงผลหลักในข้อ 4 ปัญหาที่เกิดขึ้นส่วนใหญ่เกิดจากค่าความต้านทานที่กำกับอยู่ แก้ปัญหาได้โดยการลดค่าความต้านทานลง ส่วนปัญหาอื่นก็คือ การที่ป้อนความถี่จากฟังก์ชันเจเนเรเตอร์มากเกินไป ความสามารถของ OPTO ISOLATOR ทำให้ไม่มีสัญญาณในข้อ 4 หรือสัญญาณที่ได้มีสัญญาณคลื่นเท่ากันแต่ภายในหนึ่งลูกคลื่นนั้น จะมีสัญญาณ 0 หรือ 1 มีความกว้างของสัญญาณมากกว่าสัญญาณอินพุท ซึ่งปัญหานี้จะส่งผลถึงปัญหา BAUD RATE ของชุดแสดงผลหลักและย่อย เพราะในส่วนของโปรแกรมจะมีการตรวจสอบขอบขาลงของสัญญาณ ถ้ามีการไม่ตรงกันเกิดขึ้นเพียงเล็กน้อยจะแก้ไขโดยใช้การหน่วงเวลาทางโปรแกรมควบคุมได้ แต่ถ้ามีมากเกินไปจะทำให้การส่งสัญญาณผิดพลาดได้ การแก้ไขควรจะลดความถี่ของฟังก์ชันเจเนเรเตอร์ โดยให้ทราบความถี่ที่แน่นอนก่อน แล้วจึงเปลี่ยน OPTO ISOLATOR ที่มีความไวตามต้องการ

จากนั้นนำขาสัญญาณอินพุท TTL มาต่อเข้ากับขา TXA0 ของซีพียู Z80180 ของแผ่นวงจรเดี่ยว Z80180 แล้วส่งข้อมูลจากแผ่นวงจรเดี่ยวโดยใช้ลักษณะวน LOOP ใช้ OSCILLOSCOPE จับสัญญาณ ในตอนแรกให้ลองส่งข้อมูล 55H ถ้าได้แล้วให้ทดลองส่งข้อมูลอื่นๆ

4.3) ส่วนที่ 3 การศึกษาอุปกรณ์ต่างๆในชุดทดลองหลัก

ภายในชุดทดลองผลหลักประกอบไปด้วยอุปกรณ์สำเร็จรูปหลายชนิดซึ่งจะต้องนำอุปกรณ์เหล่านั้นมาประกอบให้ทำงานร่วมกัน ขั้นตอนการทำงานในส่วนนี้ประกอบไปด้วย

- 1) ศึกษาแผ่นวงจรเดี่ยว Z80 CP-A เช่น I/O PORT
- 2) ศึกษาแผ่นวงจร 7210Z80 ในการเชื่อมต่อ CONTROL PORT ไม่ให้ซ้ำกัน เพราะมีถึง 3 แผ่น การเชื่อมต่อกับ Z80 CP-A การเชื่อมต่อกับ 7 SEGMENT
- 3) ศึกษาแผ่นวงจร 7 SEGMENT ในการเชื่อมต่อกับ 7210Z80
- 4) นำอุปกรณ์ทั้ง 3 ข้อแรกมาประกอบได้ดังรูปที่ 4.5 พร้อมทั้งทำชุดป้อนแรงดันไฟฟ้ากระแสตรงให้ตามความต้องการของอุปกรณ์
- 5) เขียนโปรแกรมควบคุมการทำงานของระบบ โดยใช้ ET BOARD VERSION 3.5 ในการพัฒนาโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงแก้ไขและเผยแพร่ในวงจำกัดของเอกสารฉบับนี้ที่มีการนำไปใช้

รูปที่ 4.5 แสดงการต่ออุปกรณ์ภายในของชุดทดลองผลหลัก

ภายในภาคแสดงผลของชุดแสดงผลหลัก ประกอบด้วย

1) แถวแรก แสดงเปอร์เซ็นต์เป้าหมาย (TARGET YIELD) โดยสามารถใช้ปุ่มเปลี่ยนแปลงค่าได้ตลอด

2) แถวสอง แสดงเปอร์เซ็นต์ที่ผ่านการคำนวณจากค่า PASS และ FAIL ได้แก่ค่า YIELD COMMUTATIVE

3) แถวสาม แสดงจำนวนผลรวมทั้งหมดของผลิตภัณฑ์ที่เข้าตรวจสอบ

4) แถวลี่ แสดงจำนวนผลิตภัณฑ์ที่ไม่ผ่านการตรวจสอบ (FAIL)

5) แถวห้า แสดงจำนวนผลิตภัณฑ์ที่ผ่านการตรวจสอบ (PASS) มี 2 กรณี ถ้าเป็นการคำนวณแบบแรก ค่าที่แสดงคือจำนวนผลิตภัณฑ์ที่ผ่านการตรวจสอบของชุดแสดงผลย่อยทุกตัวรวมกัน อีกแบบหนึ่งจะแสดงค่าจำนวนผลิตภัณฑ์ของชุดแสดงผลย่อยชุดสุดท้าย

ในลวดแสดงผลแถวที่สองมีวิธีการคำนวณ YIELD COMMUTATIVE 2 แบบ เพื่อให้เหมาะสมกับการใช้งานทางอุตสาหกรรมประเภทต่างๆ ซึ่งจะมีลิทซ์แบบเปิดปิดเป็นตัวเลือกในการคำนวณ ซึ่งวิธีการในการคำนวณมีดังนี้

แบบที่ 1 เป็นการคำนวณแบบนำผลของชุดแสดงผลย่อยทั้งหมดมารวมกันแล้วเทียบกับ 100 ดังสมการ

$$YIELD COMMUTATIVE (\%) =$$

$$\frac{\text{จำนวนของผลิตภัณฑ์ที่ผ่านการตรวจสอบทั้งหมด} * 100}{\text{จำนวนของชุดแสดงผลย่อย} * \text{ผลรวมของผลิตภัณฑ์ที่เข้ารับการตรวจสอบทั้งหมด}}$$

แบบที่ 2 เป็นการคำนวณผลิตภัณฑ์ที่ต้องผ่านการตรวจสอบหลายขั้นตอน ซึ่งทุกขั้นตอนนั้นผลิตภัณฑ์จะต้องสมบูรณ์ เพราะฉะนั้นจึงต้องมีการกำหนดชุดแสดงผลย่อย

เอกสารนี้ตัวลุดท้ายโดยใช้ DIP SWITCH เพื่อการใช้งานที่ต้องการให้ชุดแสดงผลย่อยเพียงค่า 6 การชุดไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก 8 ชุดซึ่งเป็นจำนวนชุดที่มากที่สุด ก็ให้สวิตช์ของช่องที่ 7 และ 8 ให้ปิดไป หลังจากนั้นโปรแกรมก็จะใช้ข้อมูลในการคำนวณดังนี้

สมมติ ให้ A = ผลรวมของผลิตภัณฑ์ที่ไม่ผ่านการตรวจสอบทั้งหมด

B = ผลรวมของจำนวนผลิตภัณฑ์ทั้งหมดที่เข้ารับการตรวจ

สอบที่ชุดแสดงผลย่อยที่ 1

$$\text{YIELD COMMUTATIVE (\%)} = [(B-A)*100]/B$$

ข้อแตกต่างของการคำนวณทั้งสองแบบคือ แบบแรกจะเป็นการคำนวณของผลิตภัณฑ์ทุกชิ้นที่เข้ารับการตรวจสอบ โดยผลิตภัณฑ์แต่ละชิ้นจะผ่านการตรวจสอบเพียงครั้งเดียว แต่ในแบบหลังจะเป็นการตรวจสอบผลิตภัณฑ์หลายครั้งในผลิตภัณฑ์ตัวเดียว เมื่อสามส่วนนี้เข้าสู่ข้อมูลต่อเข้ากับส่วนขยาย I/O PORT ของ Z80 CP-A และต่อขา RTS และ CTS จาก Z80180 เข้าสู่ส่วนขยาย I/O PORT ของ Z80 CP-A พร้อมกับทำ GND ร่วม และทำการพัฒนาโปรแกรมจนสามารถรับข้อมูลได้ถูกต้อง และนำโปรแกรมทั้งสองส่วนไปอัดลง EPROM เพื่อนำไปใช้แทน EPROMULATOR และ DEBUGGER ของ ET BOARD VERSION 3.5

4.4) การใช้งานเครื่องวัดผลได้ทางอุตสาหกรรม

การใช้งานจะแบ่งเป็นขั้นตอนตามลำดับดังนี้

1) เลือกแบบการคำนวณโดยใช้สวิตช์เปิดปิด และใช้ DIP SWITCH ในการเลือกจำนวนชุดแสดงผลย่อยให้ทำงาน ถ้าไม่ใช่ชุดไหนก็ให้เปลี่ยน DIP SWITCH เป็นปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เมื่อเปิดเครื่องชุดแสดงผลหลัก แดวก็แสดง TARGET YIELD จะ

ขึ้นจำนวน 85.0 ซึ่งมีหน่วยเป็นเปอร์เซ็นต์ ถ้าต้องการเพิ่มหรือลดค่า TARGET YIELD จะมีปุ่มในการเพิ่มและลดค่า

3) เปิดเครื่องชุดแสดงผลย่อย จอ LCD ทั้งสองจะแสดงผลดังนี้

จอที่ 1

จอที่ 2

< YIELDOMETER >

YIELDOMETER

HOW MANY TARGET

BY

YIELD DO YOU

KMIT'L

WANT ? 85.0 %

APPLIED PHYSICS

จอที่ 1 จะมีการกำหนดค่า TARGET YIELD มาให้ ถ้าต้องการเพิ่มให้กดปุ่ม PASS ถ้าต้องการให้ลดให้กดปุ่ม FAIL เมื่อได้ค่าตามที่ต้องการแล้วให้กดปุ่ม ENTER หน้าจอที่ 1 จะเปลี่ยนไปเป็น

INTERVAL VALUE

ส่วนจอที่ 2 ยังไม่มีการเปลี่ยนแปลง ใสค่าช่วงของจำนวนที่กำหนด (INTERVAL VALUE) แล้วกดปุ่ม ENTER จอทั้ง 2 จะเปลี่ยนหน้าจอดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TARGET YD. 85.0%

TARGET YD. 85.0%

YIELD ALR 00.0%

YIELD COM 00.0%

P:00000 F:00000

P:00000 F:00000

TOTAL:000000 _

TOTAL:000000

เมื่อนำจอทั้งสองแสดงข้อมูลดังกล่าวแสดงว่าพร้อมในการทำงาน โดยตัว
ย่อดังกล่าวมีความหมายดังนี้

TARGET YD. = TARGET YIELD (เป้าหมายของผลได้ทาง
อุตสาหกรรม)

YIELD ALR = YIELD ALERT (การวัดผลได้แบบช่วง
จำนวนที่กำหนด)

YIELD COM = YIELD COMMUTATIVE (การวัดผลได้แบบ
เทียบ 100)

P = PASS (ผลิตภัณฑ์ที่ผ่านการตรวจสอบ)

F = FAIL (ผลิตภัณฑ์ที่ไม่ผ่านการตรวจสอบ)

TOTAL = ผลรวมของผลิตภัณฑ์ที่เข้ารับการตรวจสอบ

จอที่ 1 แถวที่ 4 ตำแหน่งสุดท้าย (ที่เป็นเครื่องหมาย _ แทนอยู่)

จะมีตัว P, F เพื่อแสดงให้รู้ว่าตอนนี้เกิดปุม PASS หรือ FAIL

จอที่ 2 จะมีปุมเปลี่ยนหน้าจอเพื่อให้ทราบสาเหตุแต่ละแบบมีจำนวนผลิตภัณฑ์
มากน้อยเท่าใด ซึ่งหน้าจอแสดงดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A:00000 B:00000

C:00000 D:00000

E:00000 F:00000

G:00000 H:00000

โดยที่ A, B, C, D, E, F, G และ H เป็นสาเหตุของการที่ผลิตภัณฑ์
ไม่ผ่านการตรวจสอบ

4) หน้าที่ของปุ่มต่างๆหลังจากที่ทุกอย่างพร้อมจะทำงาน

PASS ใช้ในการเพิ่มผลิตภัณฑ์ที่ผ่านการตรวจสอบทีละหนึ่ง

FAIL ใช้ในการลดผลิตภัณฑ์ที่ไม่ผ่านการตรวจสอบทีละหนึ่ง

ENTER ใช้ในการตกลงว่าจะส่งข้อมูลต่างๆที่กดก่อนหน้านี้

DELETE ใช้ในการแก้ไขข้อมูลที่ผิดพลาดก่อนหน้านี้ วิธีใช้ปุ่มนี้มีดังนี้

- กดปุ่มเท่ากับจำนวนครั้งที่ต้องถอยหลังไปแก้ไข

- กดข้อมูลที่ต้องการเปลี่ยนแปลง เช่น เดิมเป็น F ก็กด

ปุ่ม PASS แทน ถ้าต้องการแก้ที่สาเหตุ เช่น เดิมเป็น

F สาเหตุ A ต้องการแก้เป็น F สาเหตุ B ให้กดปุ่ม F

ก่อนแล้วจึงกดปุ่มสาเหตุ B ตาม

- กดปุ่ม ENTER ก็จะกลับไปทำงานปกติ

CAUSE ใช้หลังจากกดปุ่ม FAIL โดยที่จะมีปุ่มนี้ 8 ปุ่ม สาเหตุ

ที่กำหนดในปุ่มเหล่านี้จะมาจากผู้ควบคุมการผลิต ถ้าผลิตภัณฑ์ที่เสียเป็นเพราะสาเหตุใดก็

ให้กดปุ่มนั้น 1 ครั้งต่อ 1 ชิ้นงาน

CLEAR ใช้เมื่อต้องการลบล้างข้อมูลทั้งหมด เพราะว่าการ

ใช้เครื่อง เมื่อมีการปิดเครื่องจะไม่มีกรลบล้างข้อมูลเพราะจะมี แหล่งจ่ายไฟสำรอง

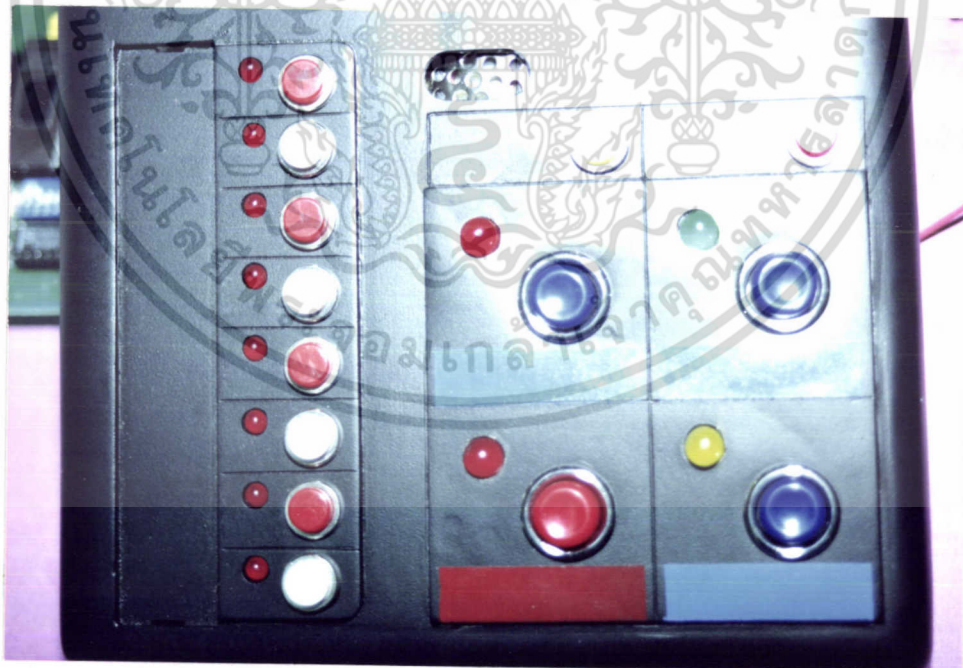
เป็นไฟเลี้ยงตลอดเวลา เนื่องจากการทำงานในโรงงานอุตสาหกรรมอาจจะไม่เสถียรภายใน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในวันเดียว ดังนั้นข้อมูลดังกล่าวจะให้หายไปไม่ได้ เพราะฉะนั้นเมื่อมีงานใหม่เข้าถึง จะลบล้างข้อมูลโดยกดปุ่ม CLEAR ข้อมูลทั้งหมดจะหายไปแล้วจึงเริ่มการทำงานทั้งหมด ใหม่โดยค่าทุกค่าจะเริ่มที่ศูนย์ และตั้งค่าต่าง ๆ ใหม่ (ลักษณะการใช้งานจะเหมือนกัน ทั้งชุดแสดงผลหลักและย่อย)

RESET ใช้เมื่อเครื่องเกิดการ HANG และต้องการเริ่มการทำงานต่อโดยข้อมูลที่แสดงต่าง ๆ ยังไม่ถูกลบล้าง (ลักษณะการใช้งานจะเหมือนกัน ทั้งชุดแสดงผลหลักและย่อย)

CHANGE ใช้เปลี่ยนหน้าจอที่ 2 ให้แสดงจำนวนผลิตภัณฑ์ที่ไม่ผ่านการตรวจสอบแต่ละลาเหตุ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกรูปที่ 4.6 แสดงการวางปุ่มต่างๆบนชุดแสดงผลย่อย ทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการวิจัยและวิจารณ์

ผลการวิจัยแบ่งออกเป็น 3 ส่วน ดังนี้

5.1) ชุดแสดงผลหลัก สามารถรับและประมวลผลข้อมูลที่ส่งมาจากชุดแสดงผลย่อยได้ โดยอาศัยโปรแกรมรับข้อมูล



รูปที่ 5.1 แสดงชุดแสดงผลหลักที่พร้อมใช้ในทางอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2) ชุดแสดงผลย่อย สามารถรับข้อมูลได้ตามที่ป้อนเข้าไป ทางปุ่มต่างๆและแสดงผลทางจอ LCD ซึ่งข้อมูลที่ได้ออกมาป้อนเข้าไปเป็นจำนวนตัวเลขหลักร้อย แต่ความสามารถของโปรแกรมที่พัฒนาเอาไว้สามารถใช้ได้ถึงหลักหมื่น ผลการวิจัยและอุปกรณ์ที่ประกอบเรียบร้อยเป็นดังรูป



รูปที่ 5.2 แสดงการแสดงผลของหน้าจอ LCD

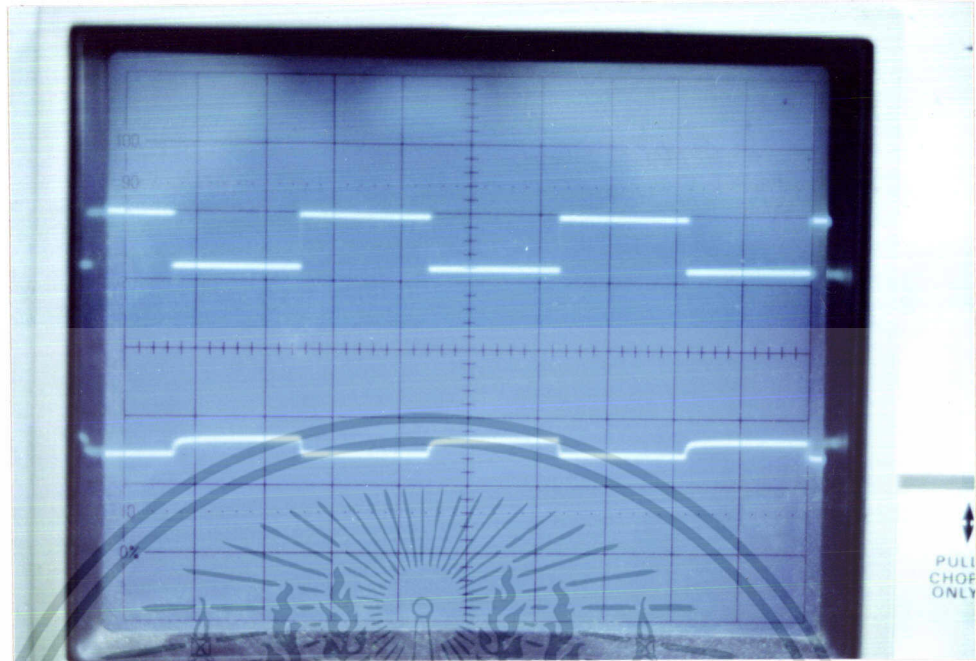
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



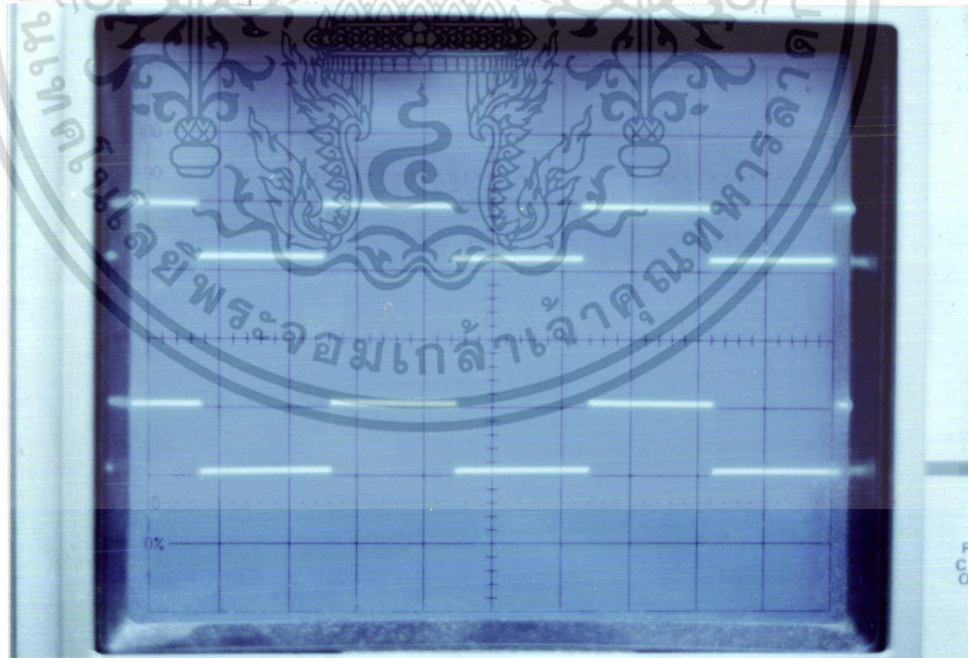
รูปที่ 5.3 แสดงชุดแสดงผลย่อยที่พร้อมใช้ในงานอุตสาหกรรม

5.3) วงจรการส่งข้อมูล ผลของการวิจัยได้สัญญาณขนาดตามต้องการของเครื่อง คือหลังจากแปลงสัญญาณ CURRENT LOOP (0-20mA) กลับมาเป็นสัญญาณ TTL จะมีขนาดของสัญญาณใกล้เคียงกับสัญญาณอินพุท TTL ในขนาดของความถี่ที่ BAUD RATE 2400 เพราะฉะนั้นจึงสามารถส่งข้อมูลระหว่างแผ่นวงจรเดี่ยวโดยใช้ BAUD RATE 2400 ในการส่งข้อมูล รูปของการทดลองเพื่อจะได้เห็นสัญญาณชัดเจนจะใช้สัญญาณจากฟังก์ชันเจนเนอเรชันแทนสัญญาณอินพุท TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงการเปรียบเทียบสัญญาณอินพุท TTL (รูปบน) กับสัญญาณที่แปลงเป็น CURRENT LOOP (0-20mA) (รูปล่าง)



รูปที่ 5.5 แสดงการเปรียบเทียบสัญญาณอินพุท TTL (รูปบน) กับสัญญาณเอาต์พุท TTL ที่แปลงจาก CURRENT LOOP (0-20mA) จากรูปล่างของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากงานวิจัยเรื่องเครื่องวัดผลได้ทางอุตสาหกรรม ส่วนที่คิดว่าได้ผลมาก
คือส่วนของโปรแกรมการควบคุมของชุดแสดงผลย่อย เพราะมีรายละเอียดต่างๆ เกี่ยว
กับการวัดผลได้ทางอุตสาหกรรมครบถ้วน การออกแบบกล่องเหมาะกับการใช้งานใน
อุตสาหกรรมและการจัดตำแหน่งปุ่มให้เหมาะกับการใช้งานและผู้ใช้งาน

ส่วนที่ควรปรับปรุงแก้ไขก็คือ

- การส่งข้อมูลควรรีวิธี pulling เนื่องจากการส่งข้อมูลใน
ลักษณะนี้จะไม่ทำให้ข้อมูลมีการสูญหาย ในกรณีที่มีชุดแสดงผลย่อยหลายชุดต้องการส่งข้อมูล
เกือบจะพร้อมกัน

- เพิ่มภาคแสดงผลภายในชุดแสดงผลหลักในรายละเอียดในส่วนของ
ลาเหตุที่ผลิตภัณฑ์ไม่ผ่านการตรวจสอบ ในการทำงานวิจัยยังขาดส่วนนี้ไปเพราะไม่มี
อุปกรณ์ที่เหมาะสมในการแสดงส่วนนี้ออกมา จากหลักการทำเครื่องการทำงานวิจัย
ครั้งนี้ใช้ 8255 ไปถึง 9 ตัว (27 PORTS) ในส่วนของชุดแสดงผลหลักแทบทุกพอร์ต
จะถูกใช้ในการแสดงผล 7 SEGMENT ดังนั้นถ้ามีอุปกรณ์แสดงผลขนาดใหญ่และใช้สายใน
การควบคุมน้อยๆเหมือนใน LCD ก็จะเหมาะสมกว่ามาก ในส่วนของแผ่นวงจรเดี่ยว
(Z80 CP-A) ที่ใช้คุมการทำงานของชุดแสดงผลหลัก ตัวซีพียูไม่เหมาะสมในการใช้
งานเนื่องจากซีพียูที่ใช้ในนั้นมีความถี่ต่ำเกินไปคือ ประมาณ 2 MHz จึงต้องเปลี่ยนจาก
ซีพียู Z80 ไปเป็น Z80A ซึ่งมีความถี่ประมาณ 3.5 MHz ถึงจะมีแรงขับสัญญาณของซีพียู
จากแผ่นวงจรเดี่ยวไปควบคุม 7210Z80 ได้ถึง 3 ชุด

ในการวิจัยนี้ได้ทำชุดรับส่งข้อมูลสำรองไว้ 8 ชุด ซึ่งสามารถรองรับ
ชุดแสดงผลย่อยได้ 8 ตัว การที่จะเพิ่มจำนวนชุดนั้นทำได้โดยไม่ยากขึ้นอยู่กับ I/O
PORT ที่เหลือภายในชุดแสดงผลหลัก ถ้าต้องการปรับปรุงให้มีมากชุดกว่านี้อาจทำได้
โดยเพิ่มแผ่นวงจร 7210Z80 ซึ่งเป็นแผ่นวงจรที่ใช้เพิ่มเติม I/O PORT และเปลี่ยน
ซีพียูที่มีความถี่มากพอกับการเพิ่มแผ่นวงจร 7210Z80 และต้องเพิ่มจำนวน DIP SWITCH

เท่ากับจำนวนชุดแสดงผลย่อยที่เพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงอุปกรณ์ตรวจเช็คของเครื่องวัดผลได้ทางอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการศึกษาวิจัยและข้อเสนอแนะ

สรุปผลการศึกษาวิจัย

ในการศึกษาและวิจัยในโครงการนี้ ได้สร้างเครื่องวัดผลได้ทางอุตสาหกรรม ซึ่งประกอบด้วย ชุดแสดงผลหลัก ชุดแสดงผลย่อย ซึ่งมีวงจรรับและส่งข้อมูลอยู่ในชุดควบคุมทั้งสอง เพื่อใช้ในโรงงานอุตสาหกรรมได้หลายประเภท ซึ่งคุณสมบัติของเครื่องที่สร้างขึ้นมีดังนี้

- 1.1) สามารถส่งข้อมูลผลได้ทางอุตสาหกรรมในระยะไกล เพื่อให้ฝ่ายบริหารทราบผลผลิตตลอดเวลา เพื่อแก้ไขขบวนการผลิตได้ทันทั่วทั้ง
 - 1.2) สามารถคำนวณค่าผลได้ทางอุตสาหกรรมได้ 2 แบบในชุดแสดงผลย่อย และสามารถคำนวณค่าผลได้ทางอุตสาหกรรมรวมได้ 2 แบบ
 - 1.3) ประหยัดเวลาในการทำงานเพราะใช้ระบบไมโครโปรเซสเซอร์ควบคุมและเก็บข้อมูล
 - 1.4) สามารถเพิ่มชุดแสดงผลย่อยได้มากเท่าที่ทางโรงงานต้องการ ทั้งนี้ขึ้นอยู่กับความสามารถของชิพและจำนวนของ I/O PORT ของชุดแสดงผลหลัก
- จากการวิจัยได้สร้างเครื่องวัดผลได้ทางอุตสาหกรรมขึ้นมา ประกอบด้วยชุดแสดงผลหลัก 1 ชุด และ ชุดแสดงผลย่อย 2 ชุด เพื่อใช้ในการศึกษาและวิจัยเท่านั้น ดังนั้นในทางอุตสาหกรรมจริง เครื่องนี้สามารถจะพัฒนาและขยายขอบเขตการทำงาน ซึ่งการขยายนั้นจะต้องเพิ่มส่วนของฮาร์ดแวร์และโปรแกรมควบคู่กันไป เช่น
- การเพิ่มจำนวนของชุดแสดงผลย่อย ต้องเพิ่มจำนวนอินพุตและเอาต์พุตพอร์ตซึ่งได้แสดงไว้ในบทที่ 2 และเพิ่ม DIP SWITCH เท่ากับจำนวนของชุดแสดงผล ในการวิจัยได้จัด DIP SWITCH สำรองไว้สำหรับชุดรับส่งข้อมูล 8 ชุดเท่านั้น และจะต้องแก้ไขโปรแกรมในส่วนของการรับข้อมูลและการคำนวณให้สอดคล้องกับจำนวนชุดแสดงผล

- การเพิ่มแบบในการคำนวณ ต้องเปลี่ยนจากสวิตช์เปิด/ปิดไป เป็น DIP SWITCH เพื่อใช้ในการเลือกการคำนวณแบบต่างๆได้มากขึ้น และจะต้องมีการพัฒนาของโปรแกรมด้วยเช่นกัน

ข้อเสนอแนะ

งานวิจัยนี้ยังสามารถพัฒนาให้มีความสมบูรณ์มากกว่านี้ได้หลายวิธี เช่น

2.1) นำส่วนของชุดแสดงผลหลักไปเชื่อมต่อกับระบบ IBM PC เพื่อไปประมวลผลต่างๆเพิ่มเติมและทำรายงานต่างๆได้โดยใช้ SOFTWARE สำเร็จรูป LOTUS

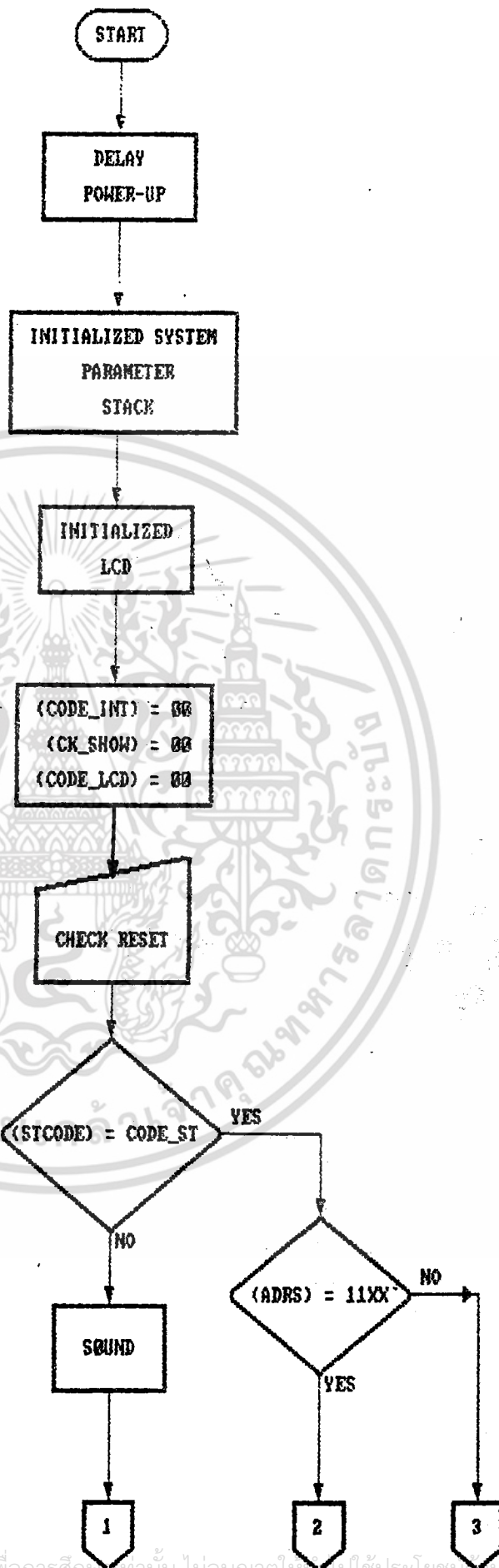
2.2) ถ้ามีการเพิ่มจำนวนชุดแสดงผลย่อยจำนวนมาก ควรแก้ไขระบบอาร์คแวร์และซอฟต์แวร์ในชุดแสดงผลย่อยให้มีการแสดงค่า P และ F บนจอภาพให้เร็วกว่านี้ เพราะถ้าตัว P และ F ยังไม่ปรากฏบนจอ LCD ข้อมูลจะไม่สามารถส่งไปยังชุดแสดงผลหลัก ดังนั้นจึงความลด LOOP การทำงานนี้ลง เช่น เลิกใช้ปุ่ม ENTER ในกรณี PASS และ FAIL ส่วนเวลาที่ใช้ในการส่งข้อมูลจำนวนชุดมากขึ้นคิดว่า จะไม่มีปัญหา เพราะเวลาที่ใช้ในการส่งข้อมูลของชุดแสดงผลย่อยแต่ละตัวใช้เวลา 0.03 วินาที ได้มาจาก จำนวนบิต*จำนวนชุดแสดงผลย่อย*(1/BAUD RATE)

2.3) การส่งข้อมูลอาจใช้วิธี pulling แทน เพื่อจะได้ลดปัญหาเรื่องเวลาในการส่งข้อมูล ในกรณีที่มีจำนวนของชุดแสดงผลย่อยมาก ๆ (เหมาะกับการอุตสาหกรรมขนาดเล็กไปจนถึงอุตสาหกรรมขนาดกลาง)

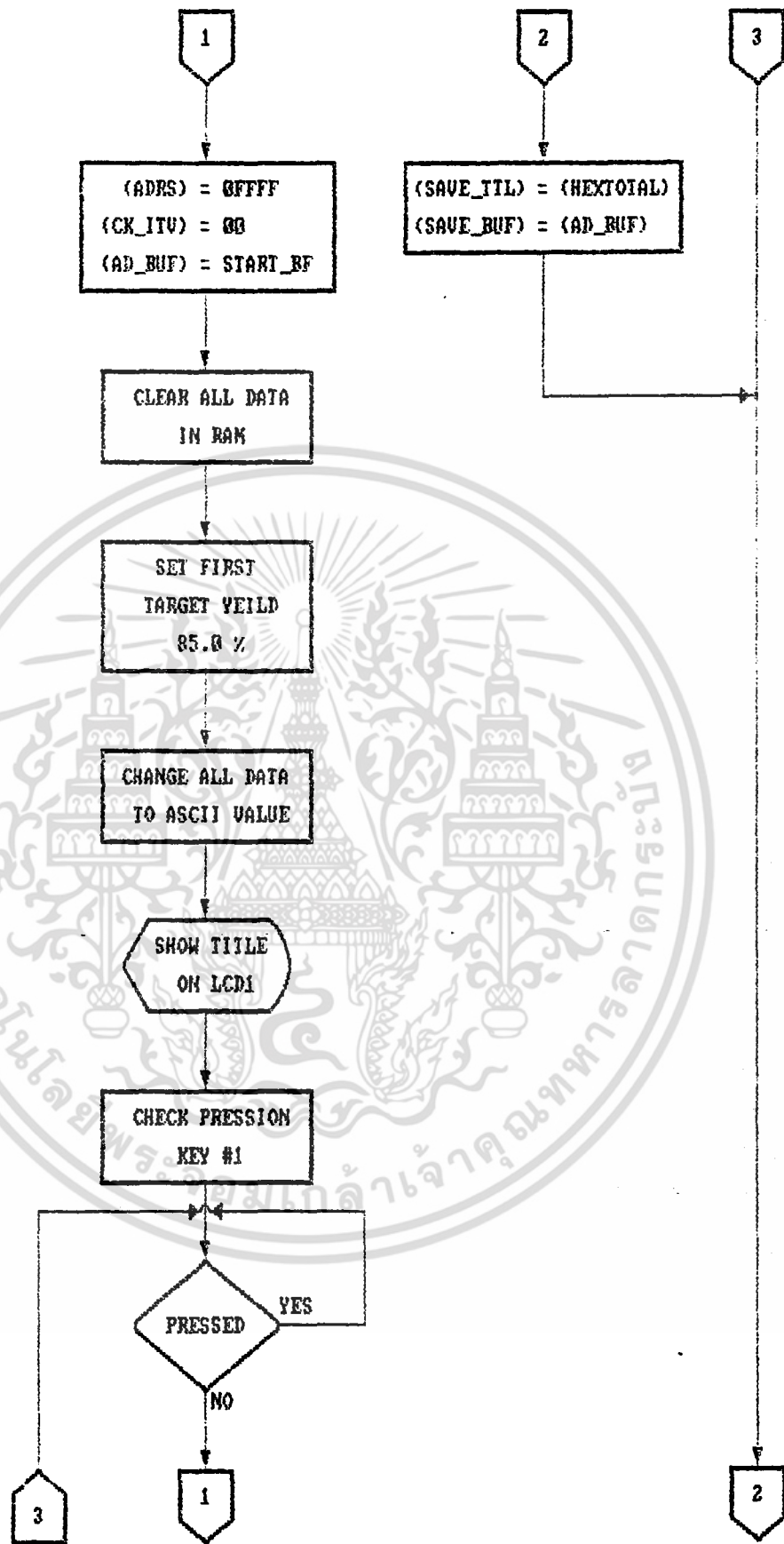


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

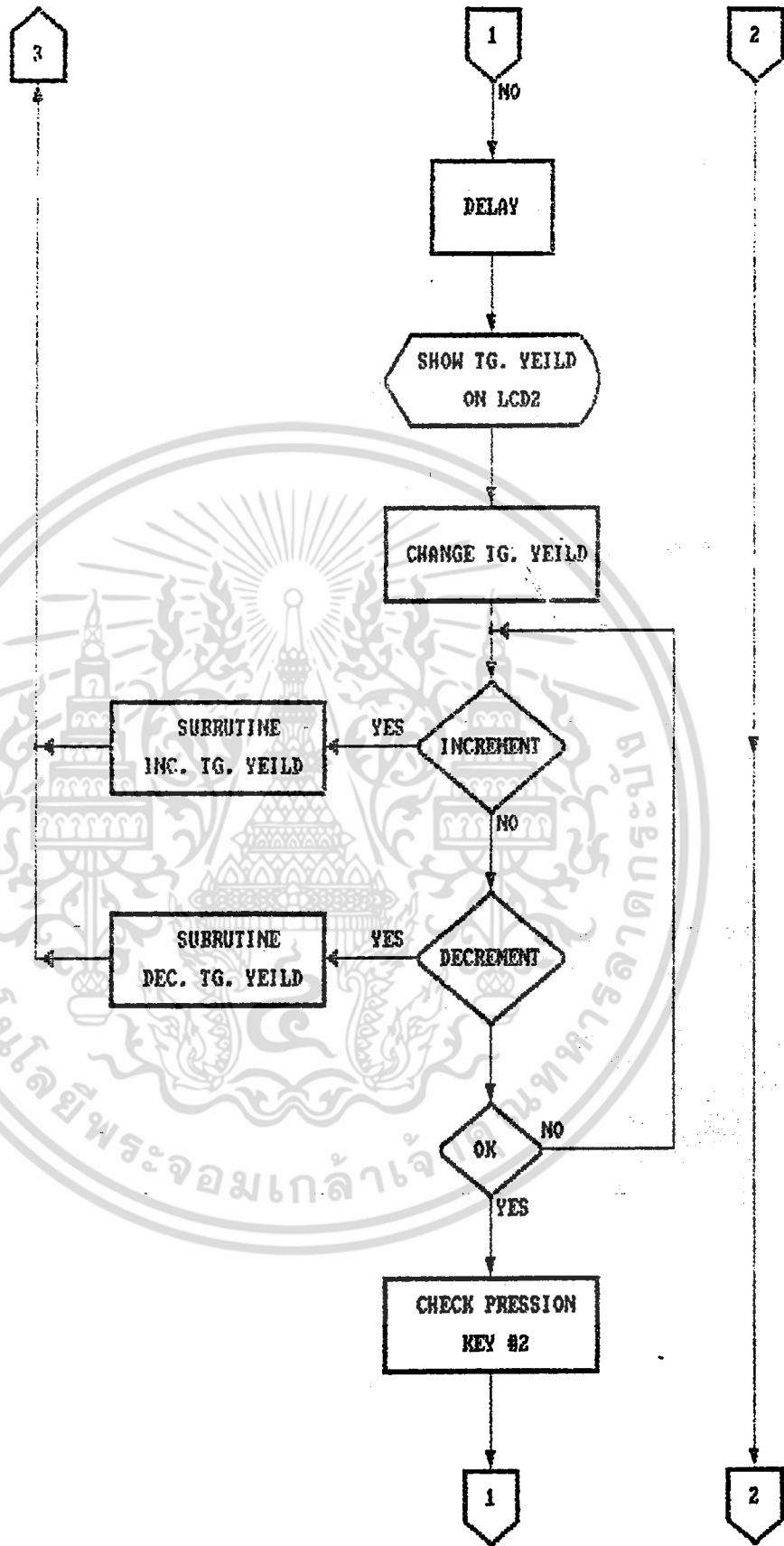
FLOW CHART
OF MAIN PROGRAM
SUB-BOARD



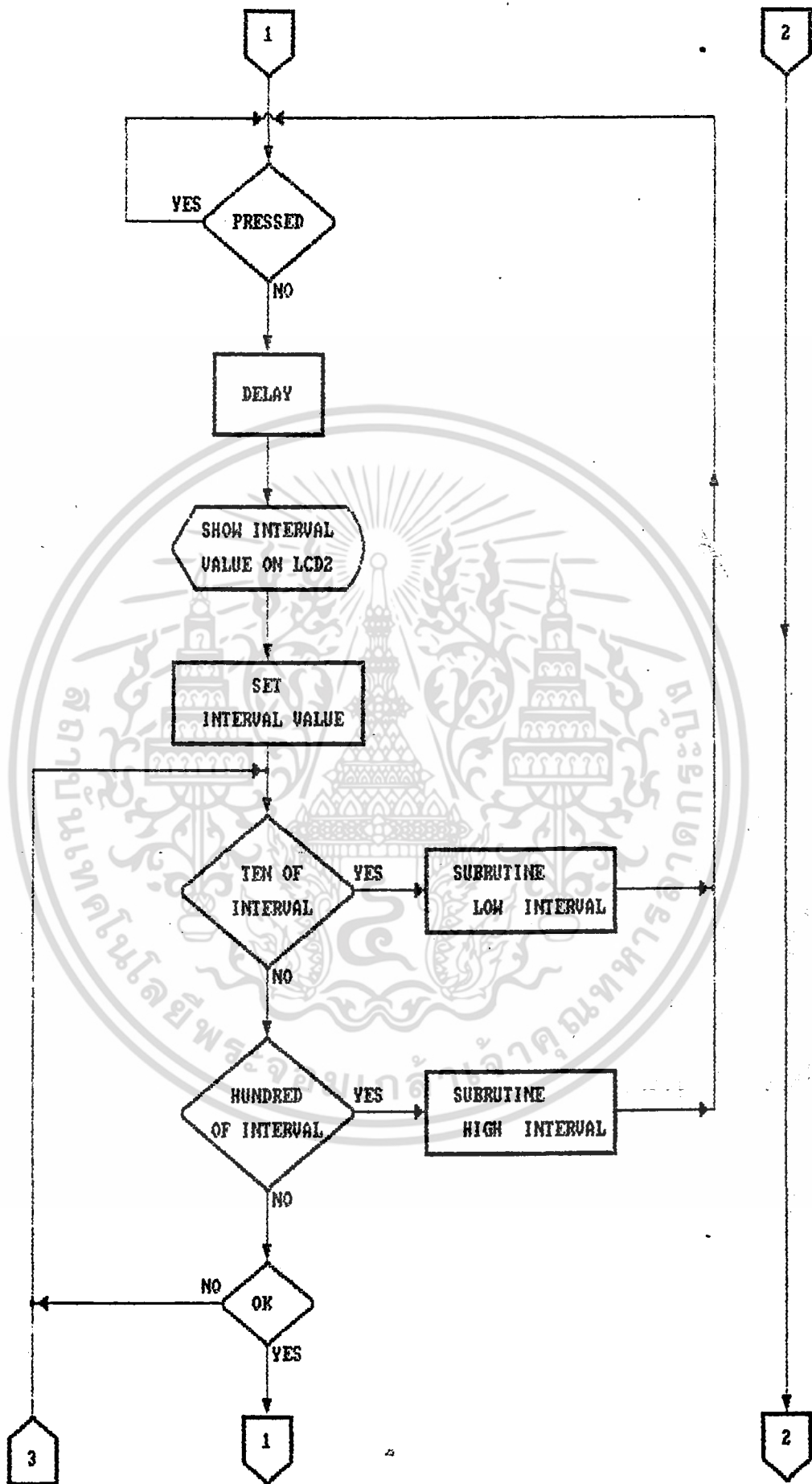
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ใช้ประโยชน์การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



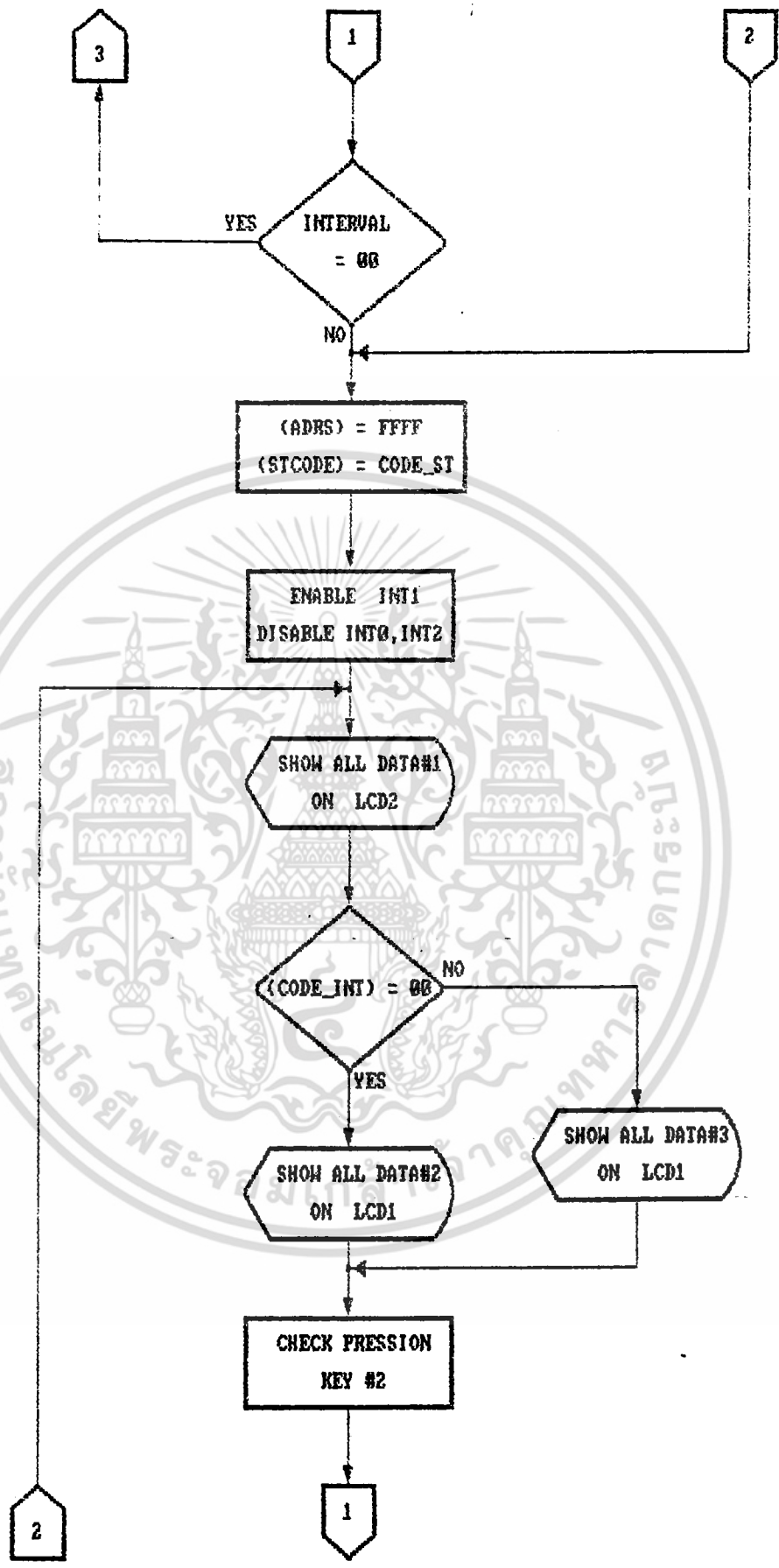
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



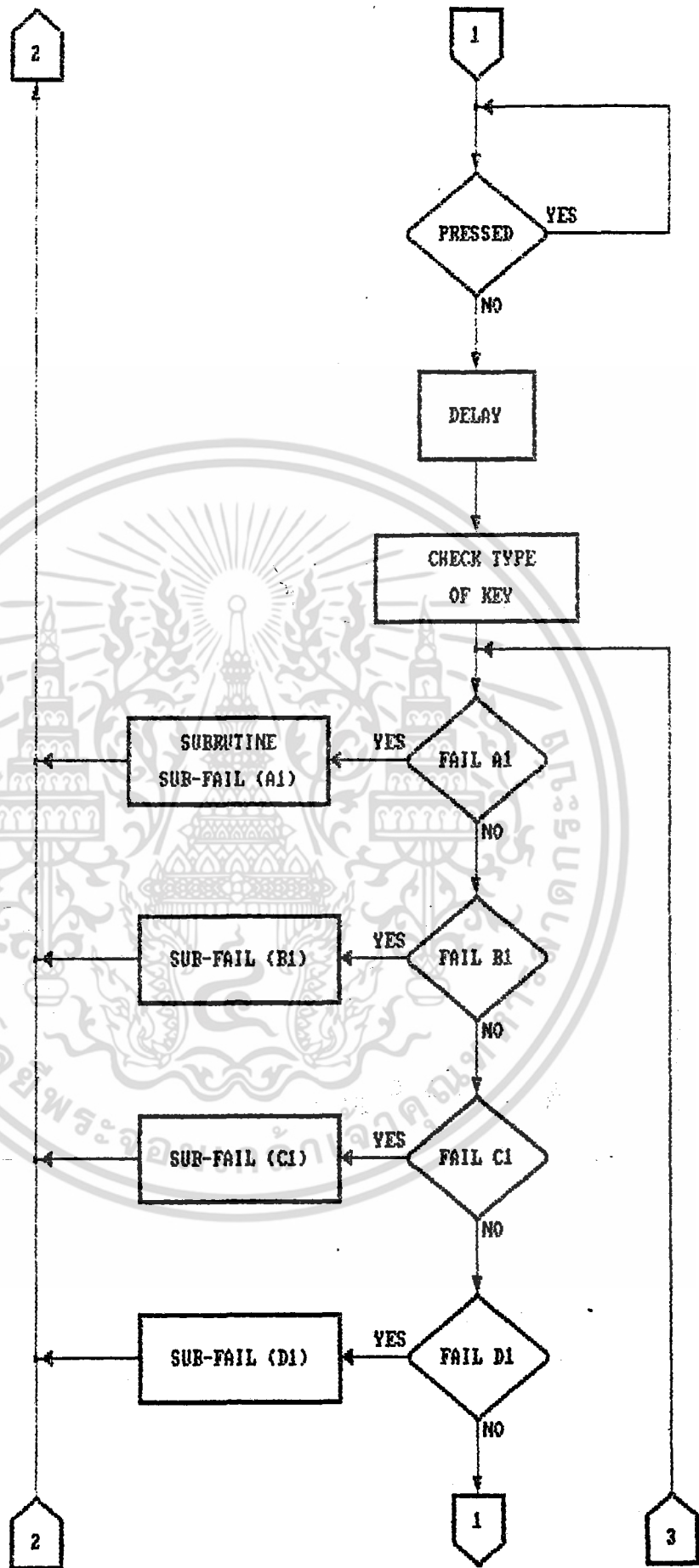
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



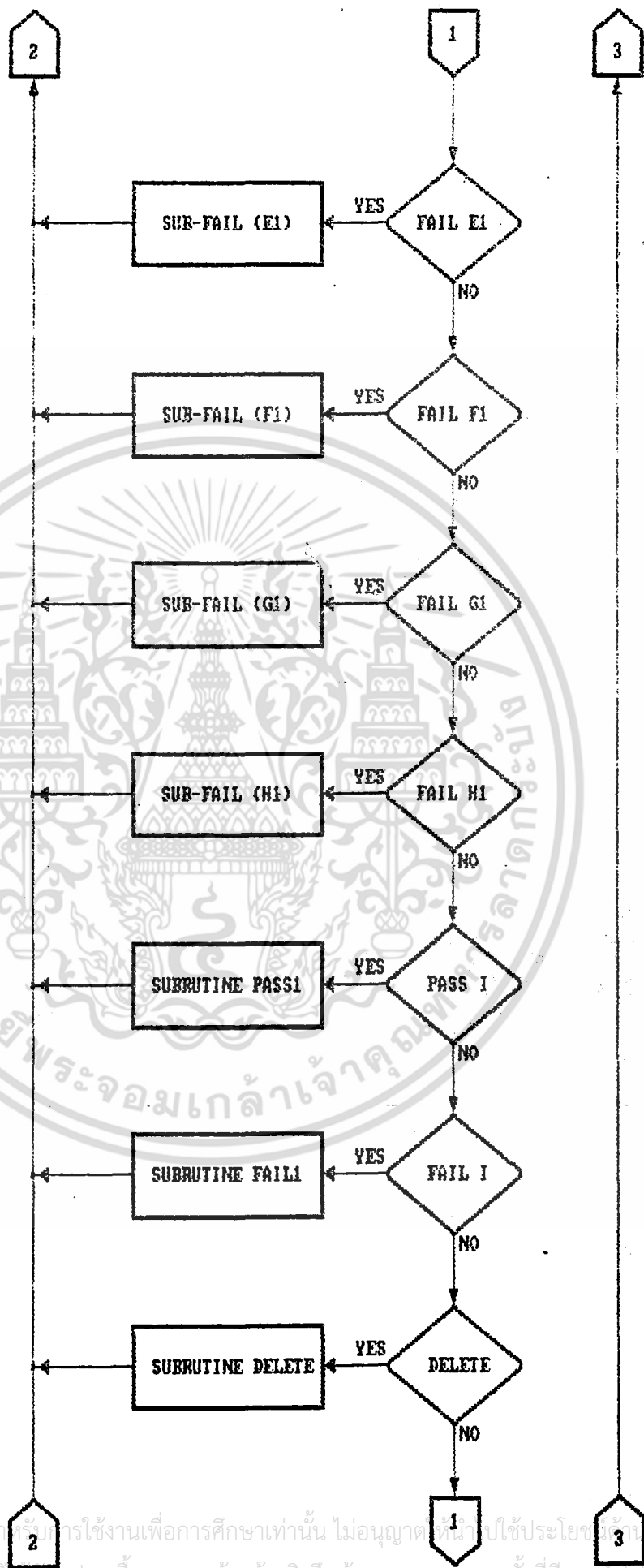
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



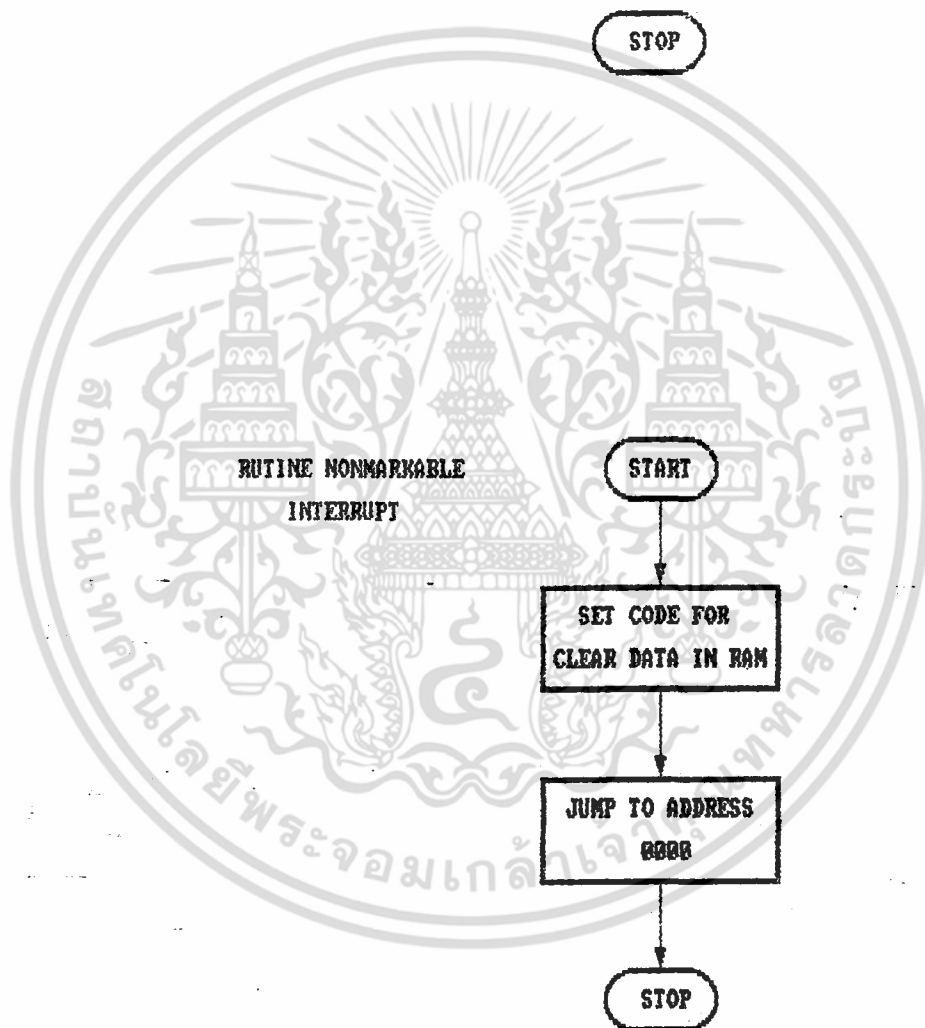
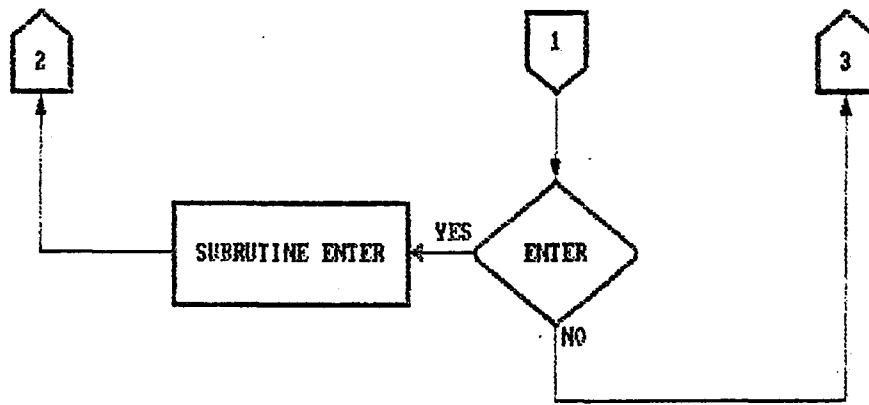
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

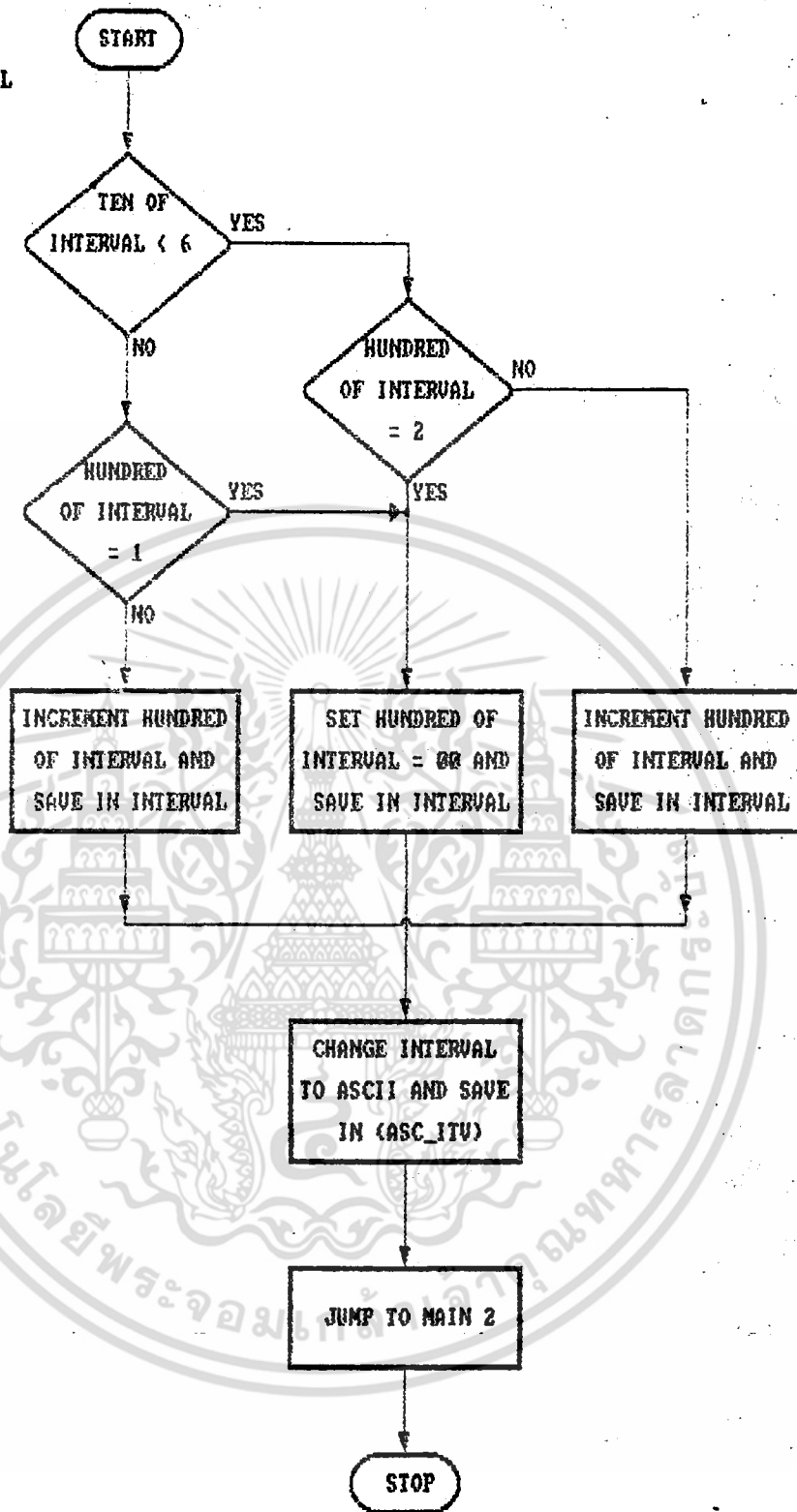


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



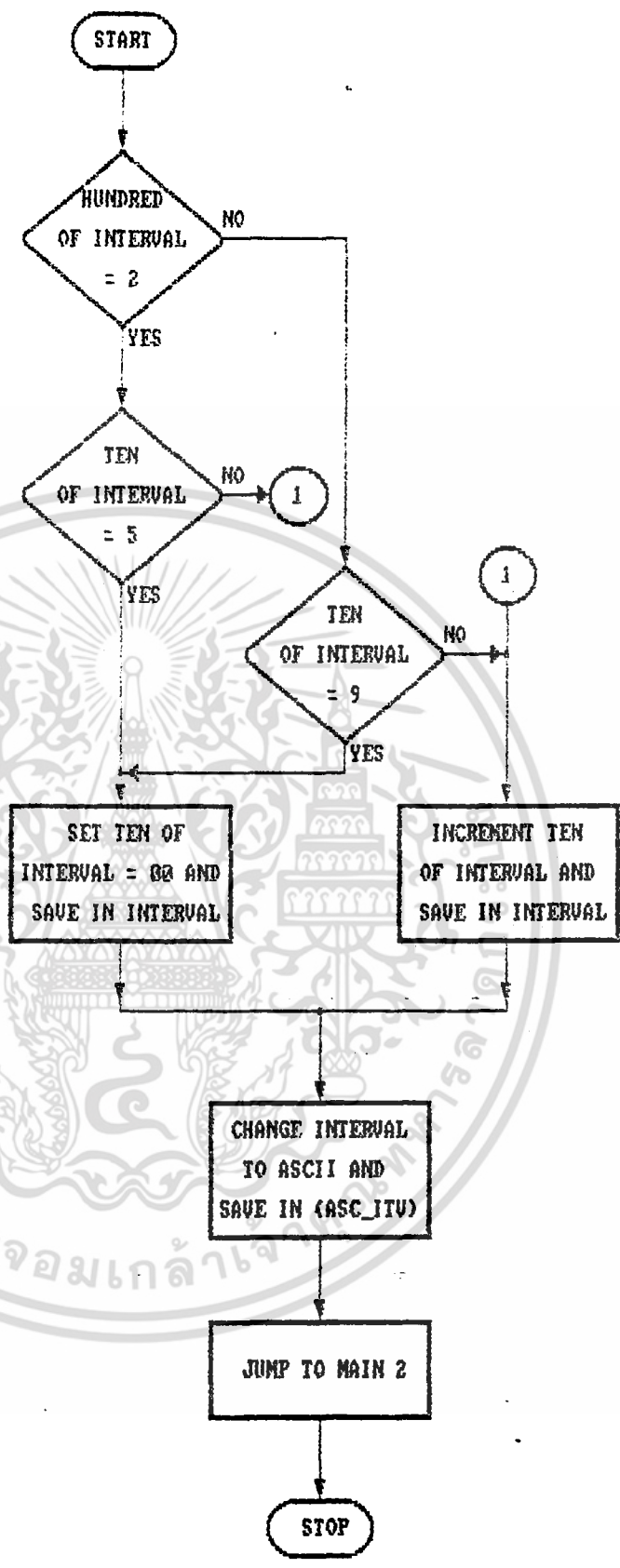
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
HUNDRED OF INTERVAL



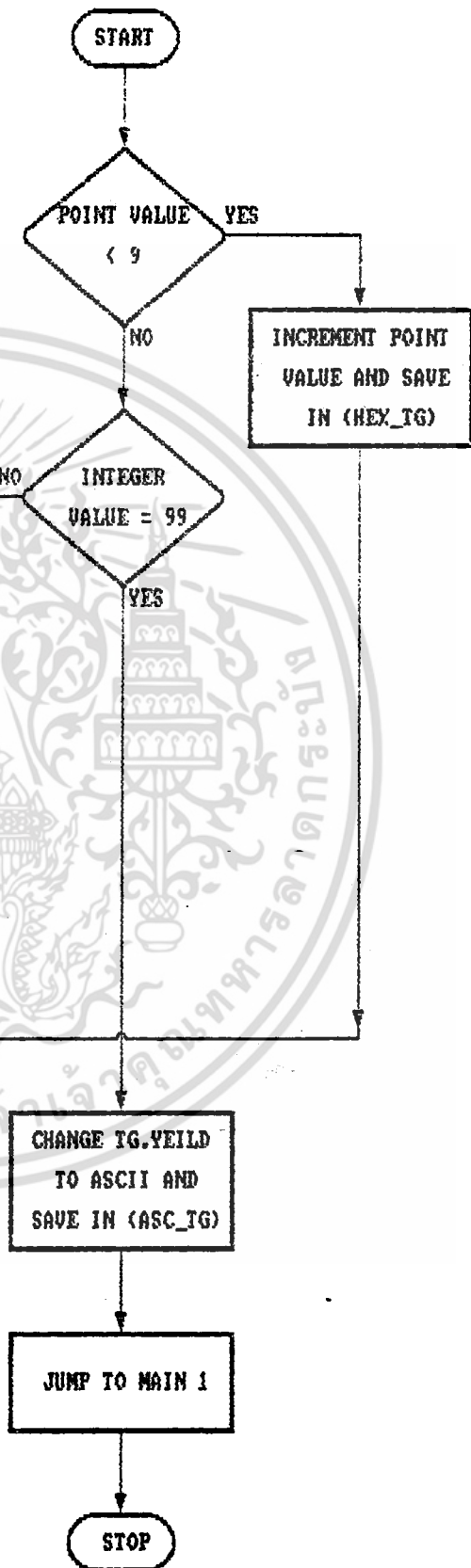
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
TEN OF INTERVAL



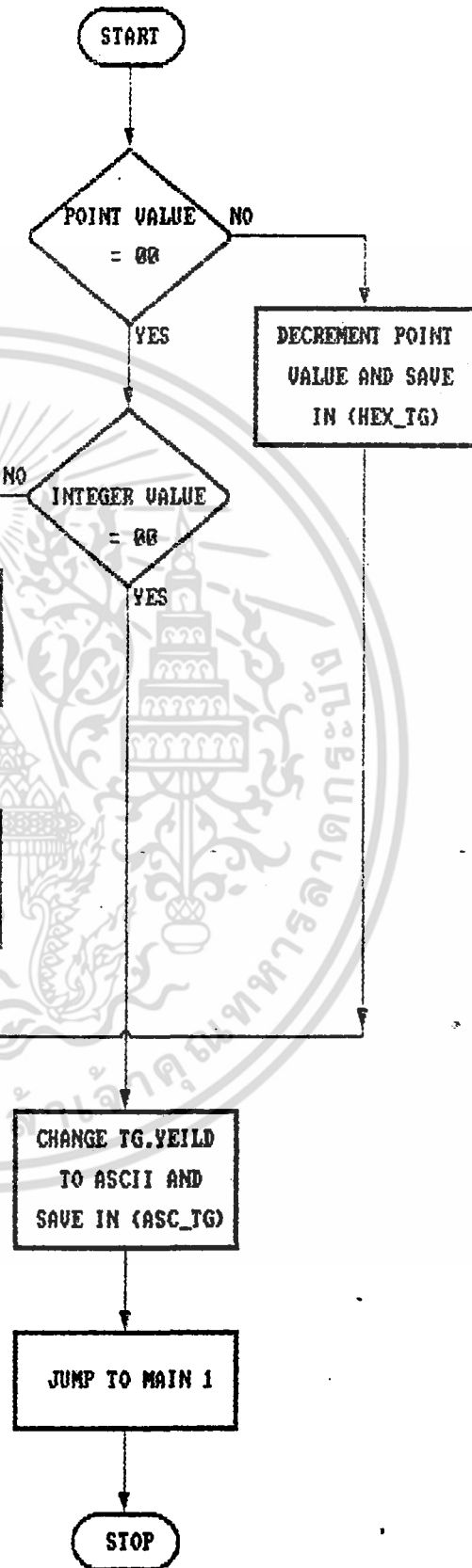
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE INCREMENT
TARGET YEILD



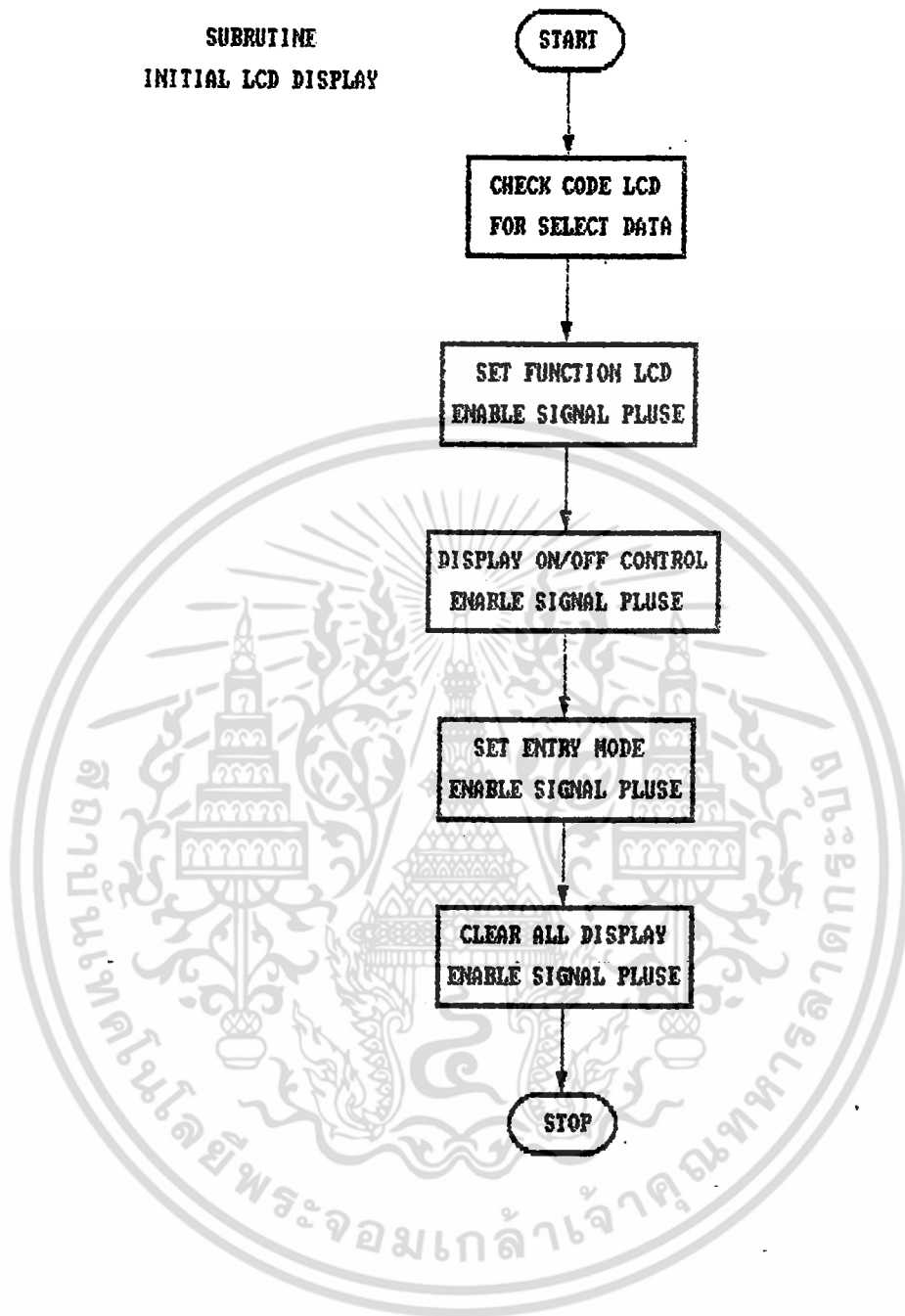
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE DECREMENT
TARGET YEILD



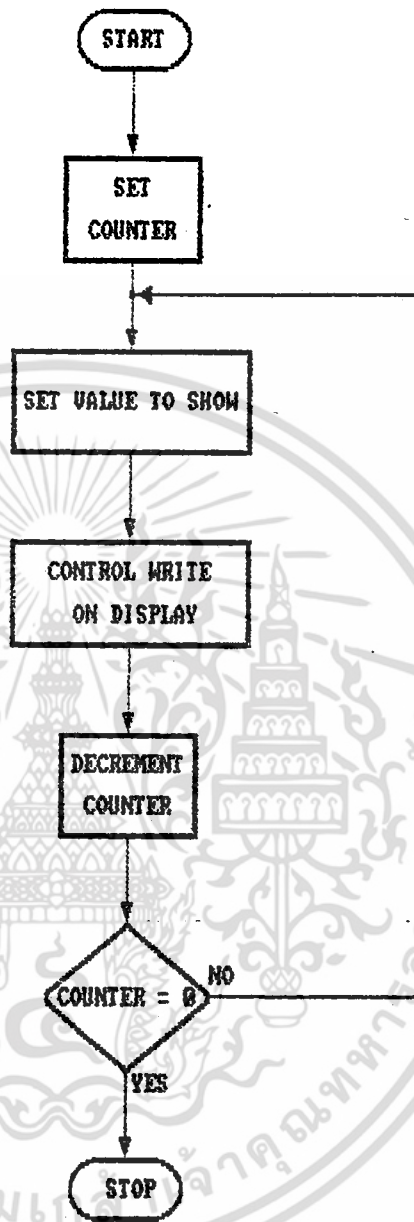
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
INITIAL LCD DISPLAY



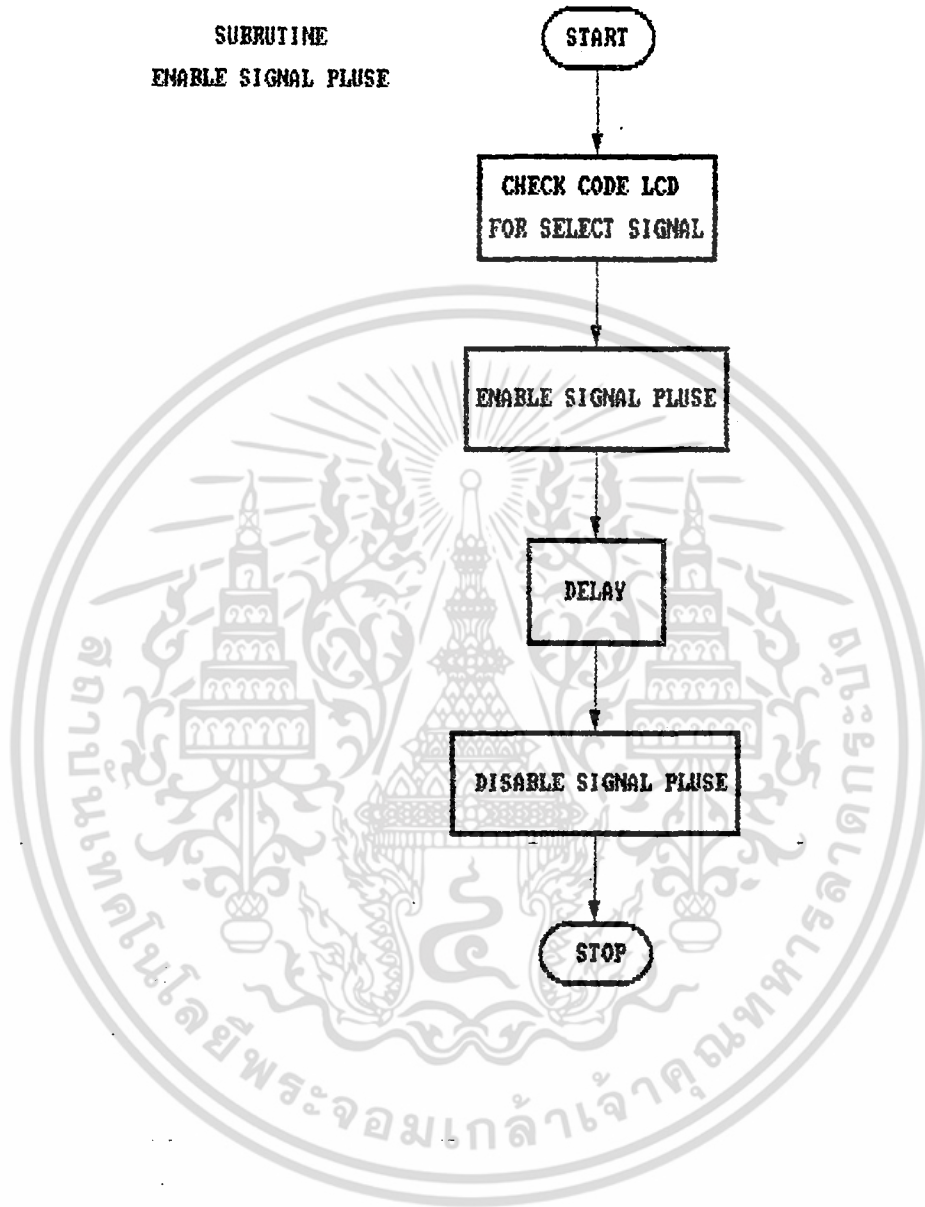
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
WRITE LINE CHARACTER



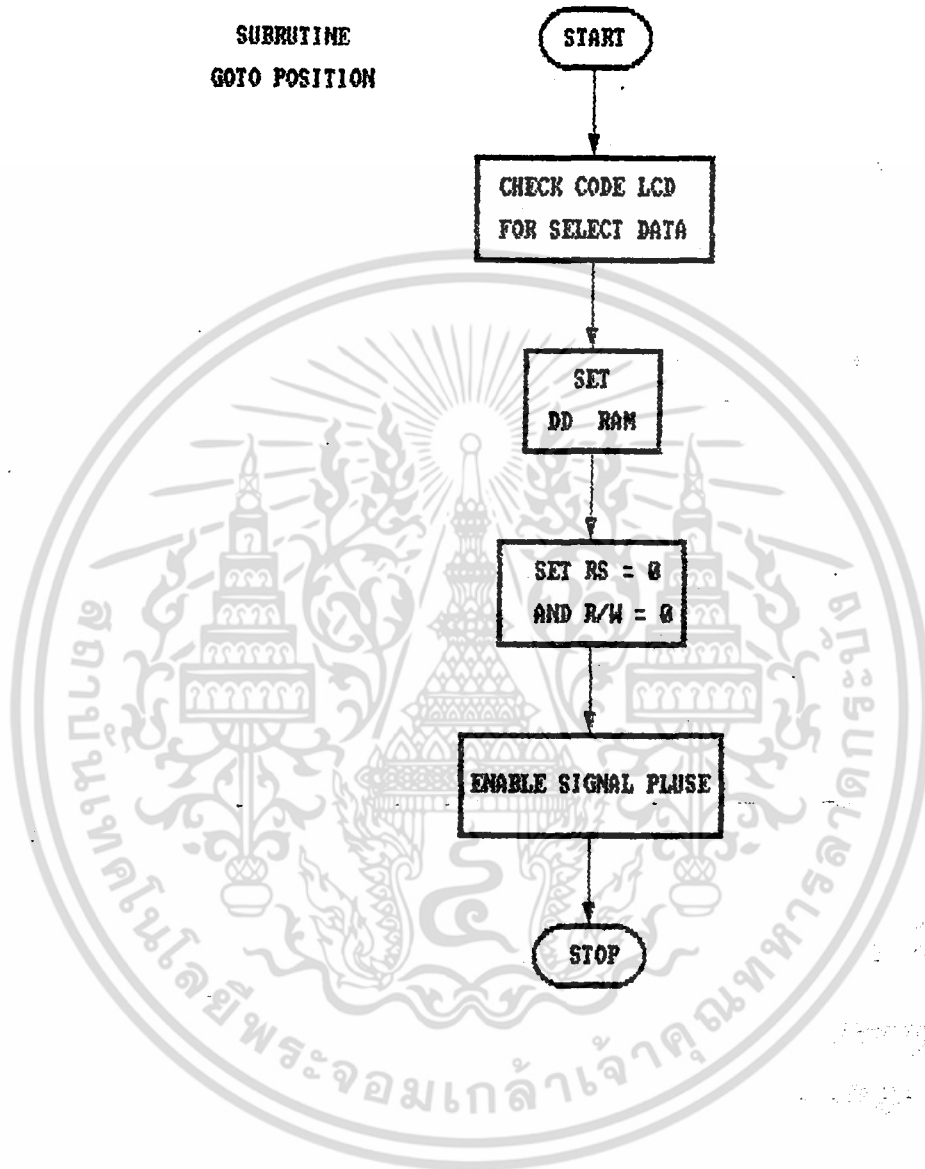
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
ENABLE SIGNAL PLUSE



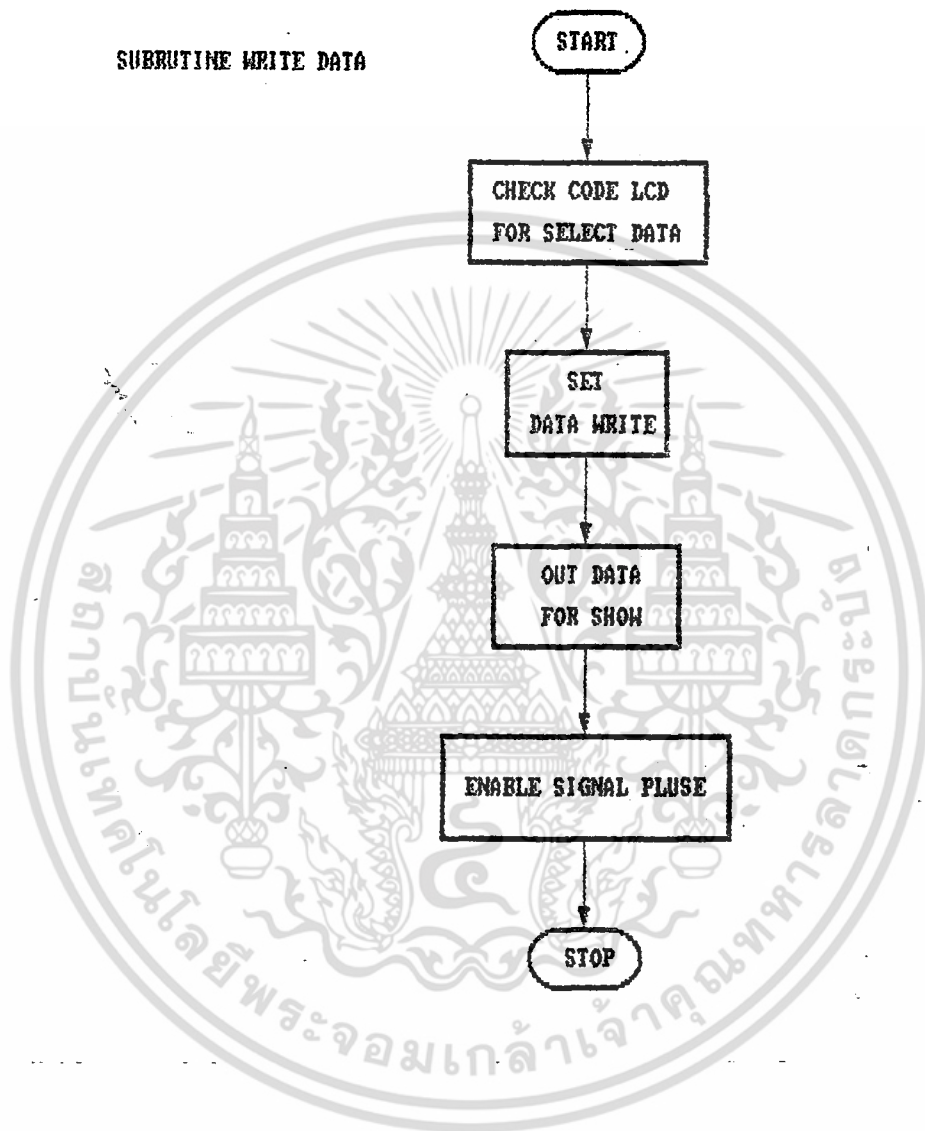
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
GOTO POSITION



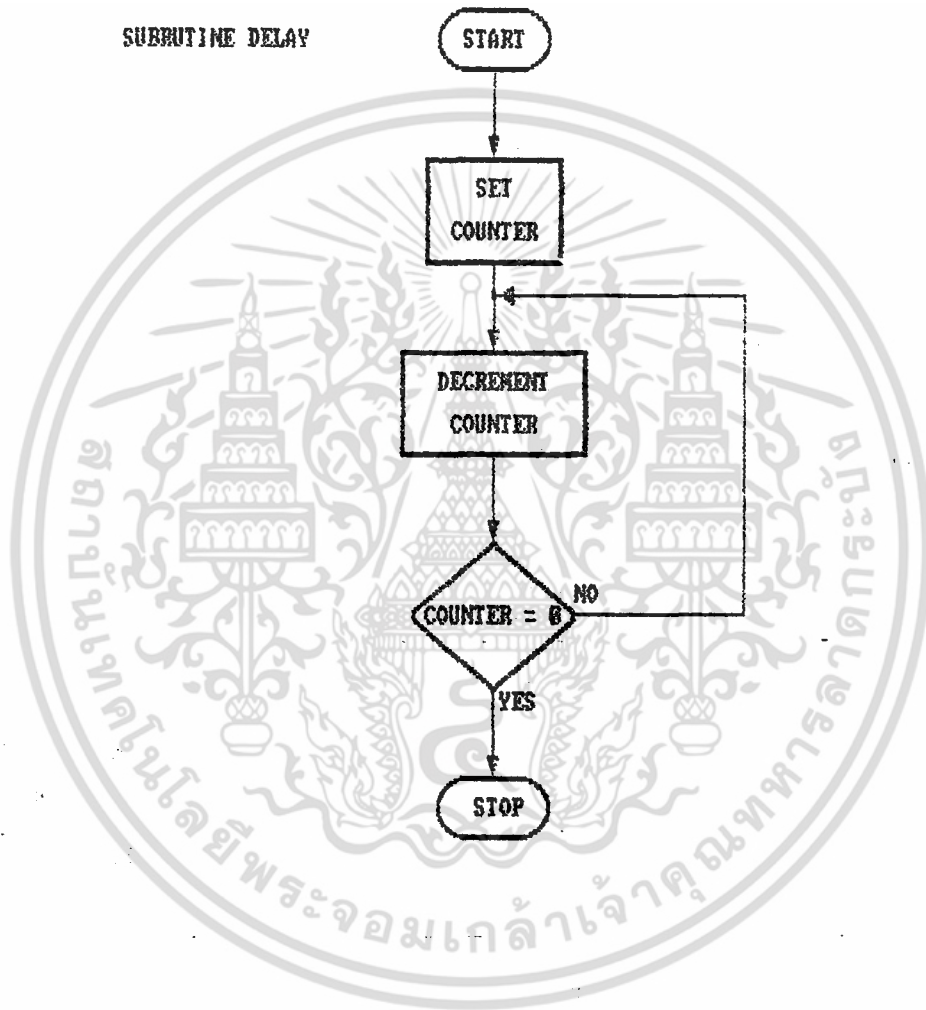
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE WRITE DATA



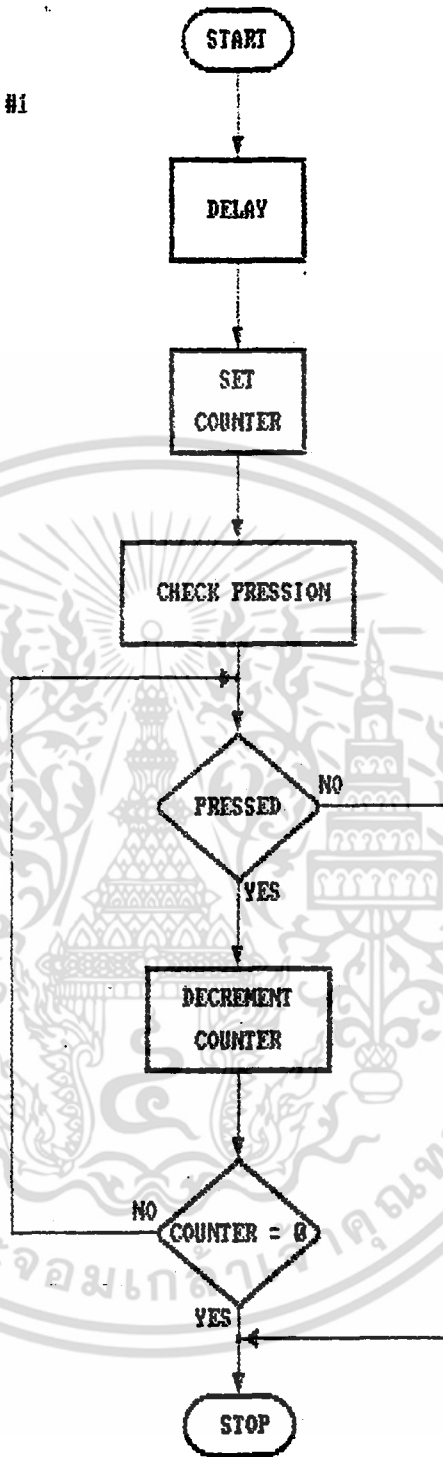
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE DELAY



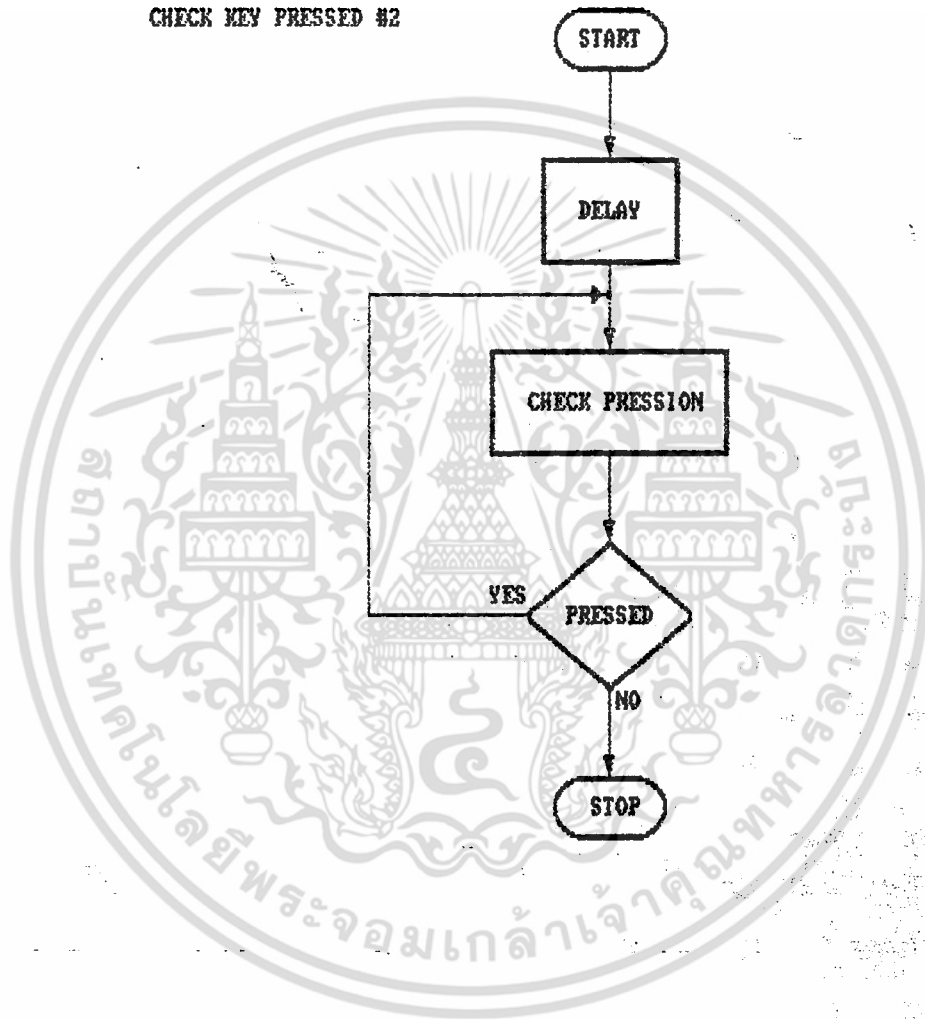
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
CHECK KEY PRESSED #1



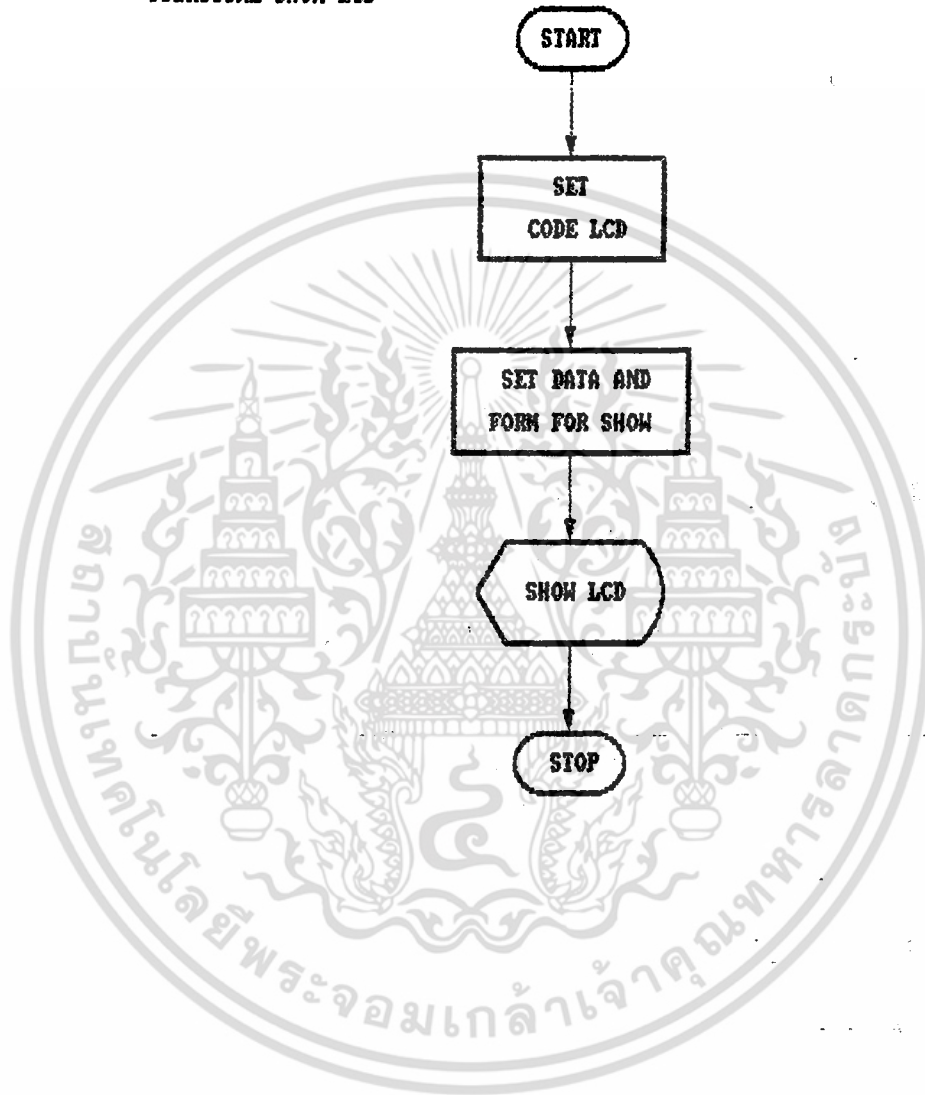
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SUBROUTINE
CHECK KEY PRESSED #2**



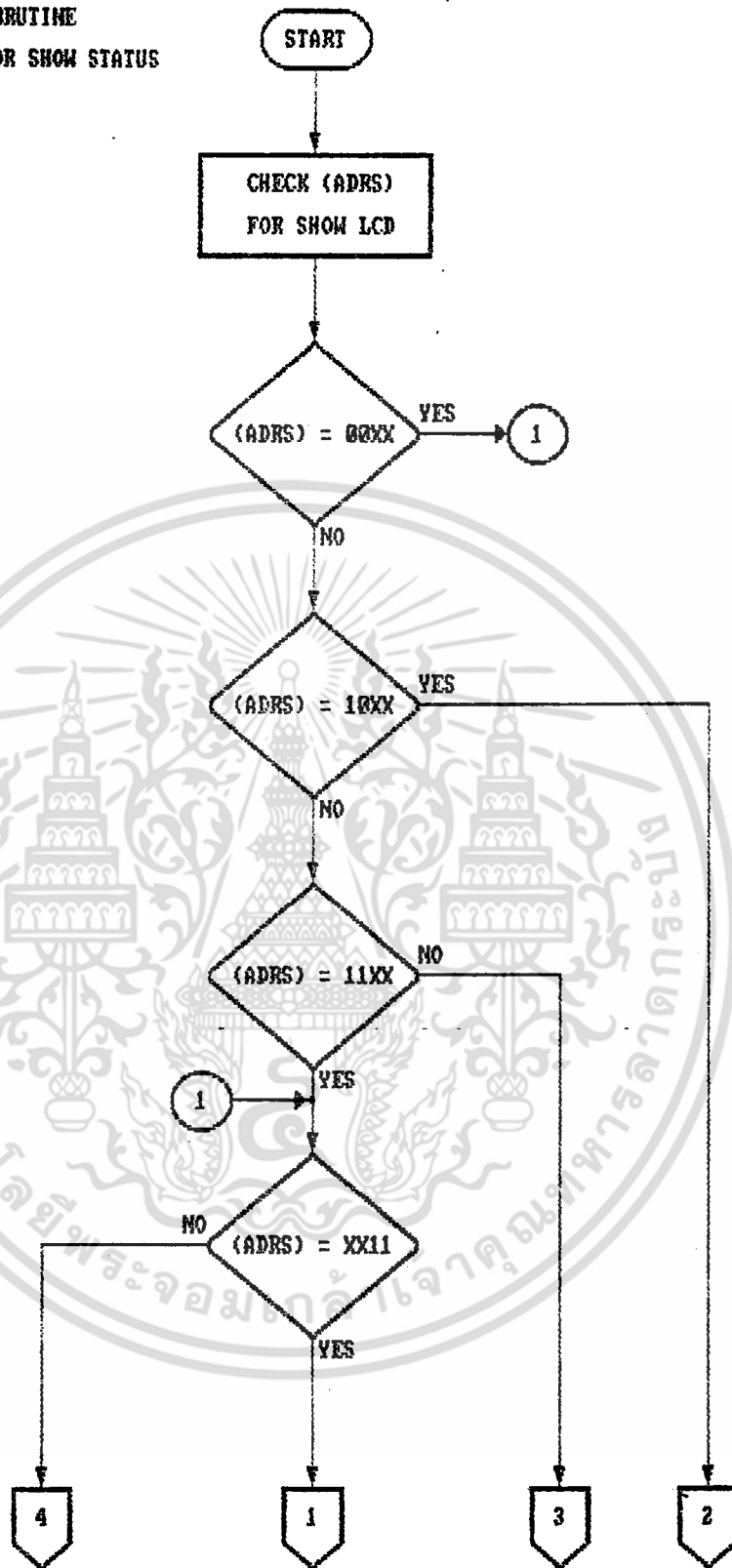
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE SHOW LCD

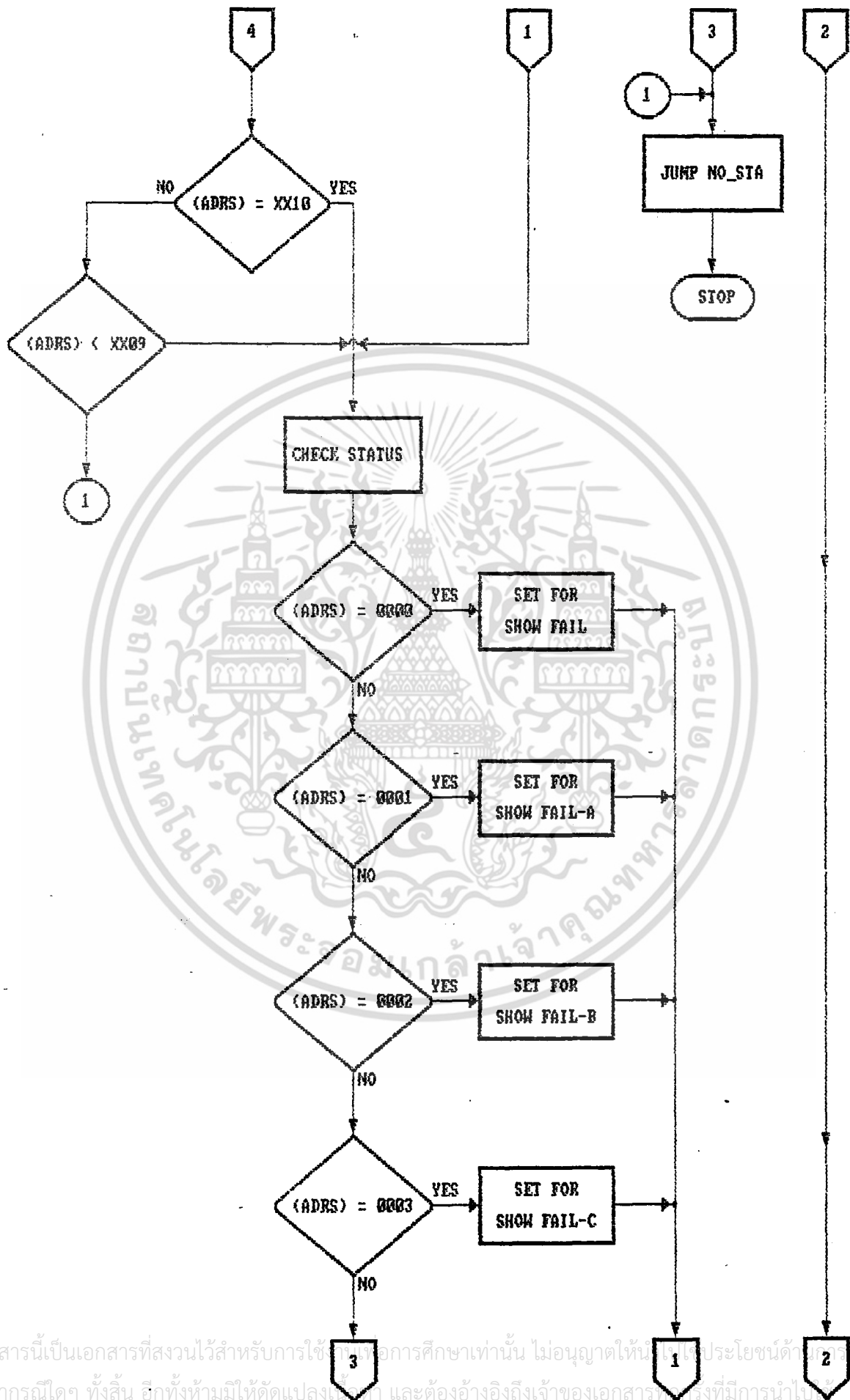


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

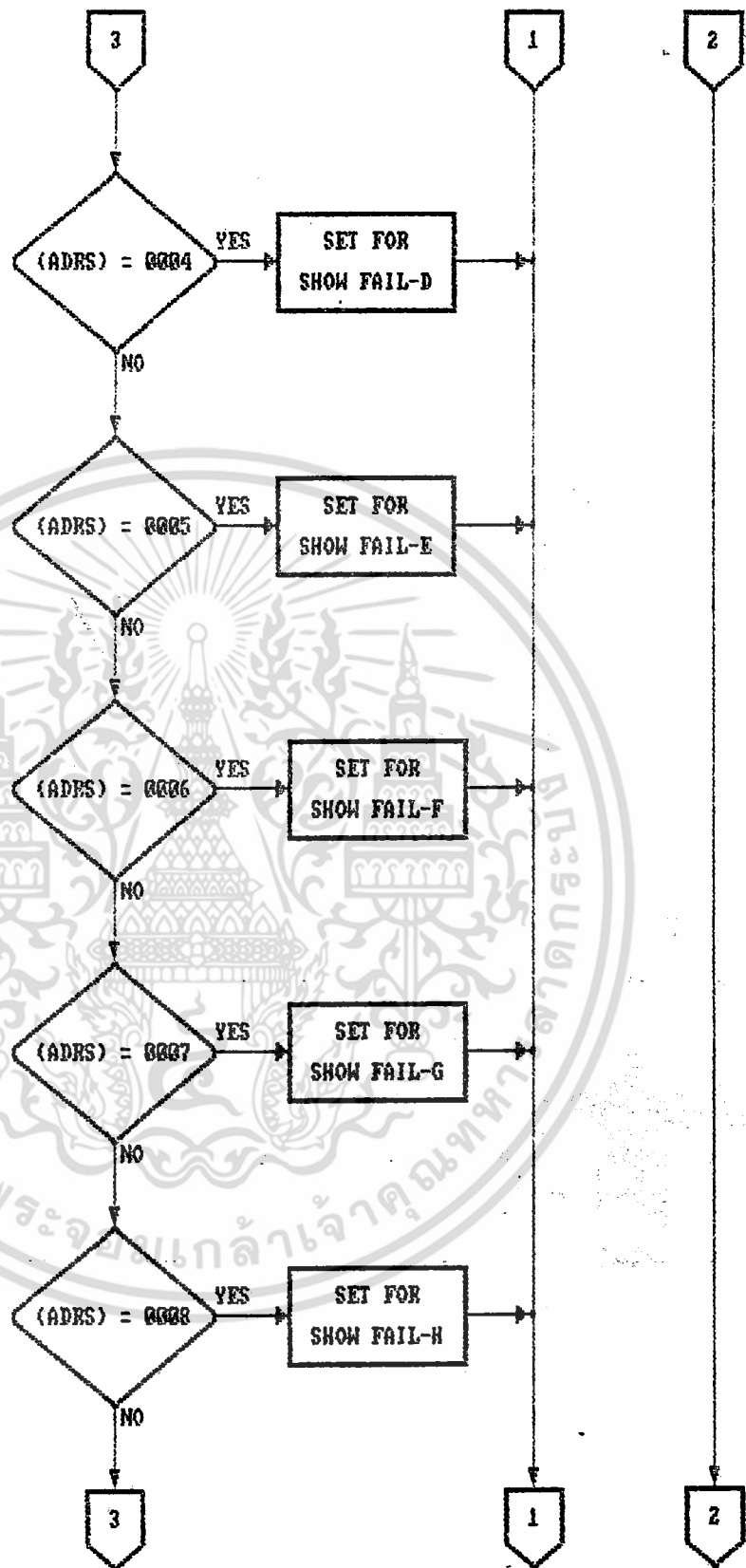
SUBROUTINE
CHECK AND/OR SHOW STATUS



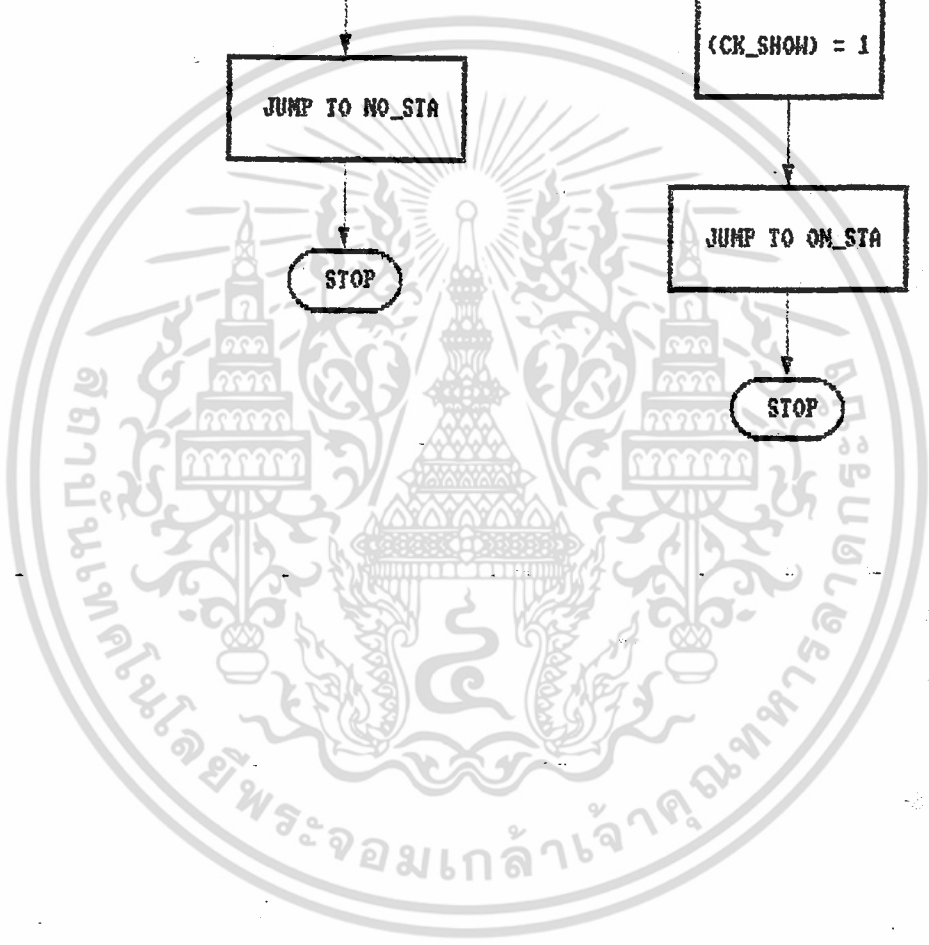
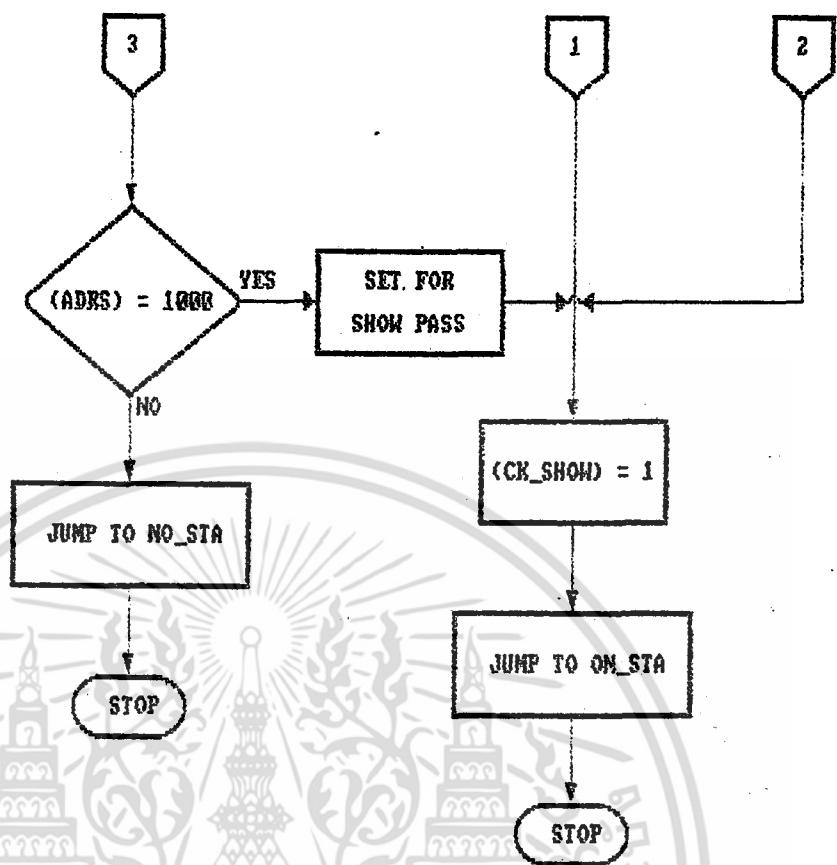
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในของมหาวิทยาลัยเท่านั้น ไม่อนุญาตให้เผยแพร่ไปภายนอกมหาวิทยาลัย
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

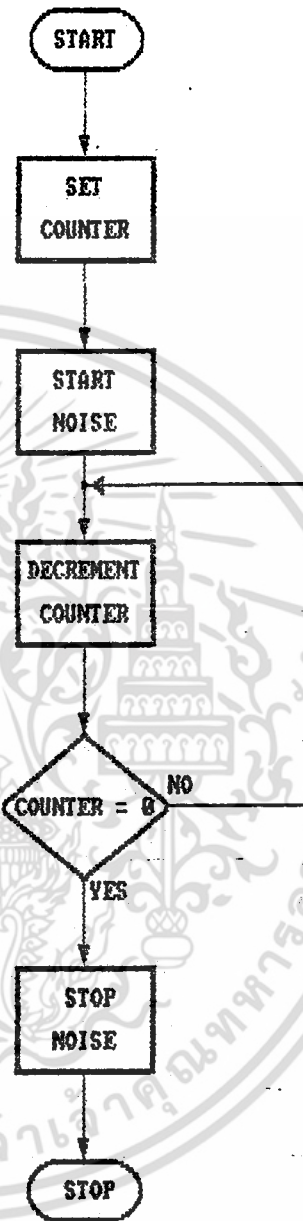


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



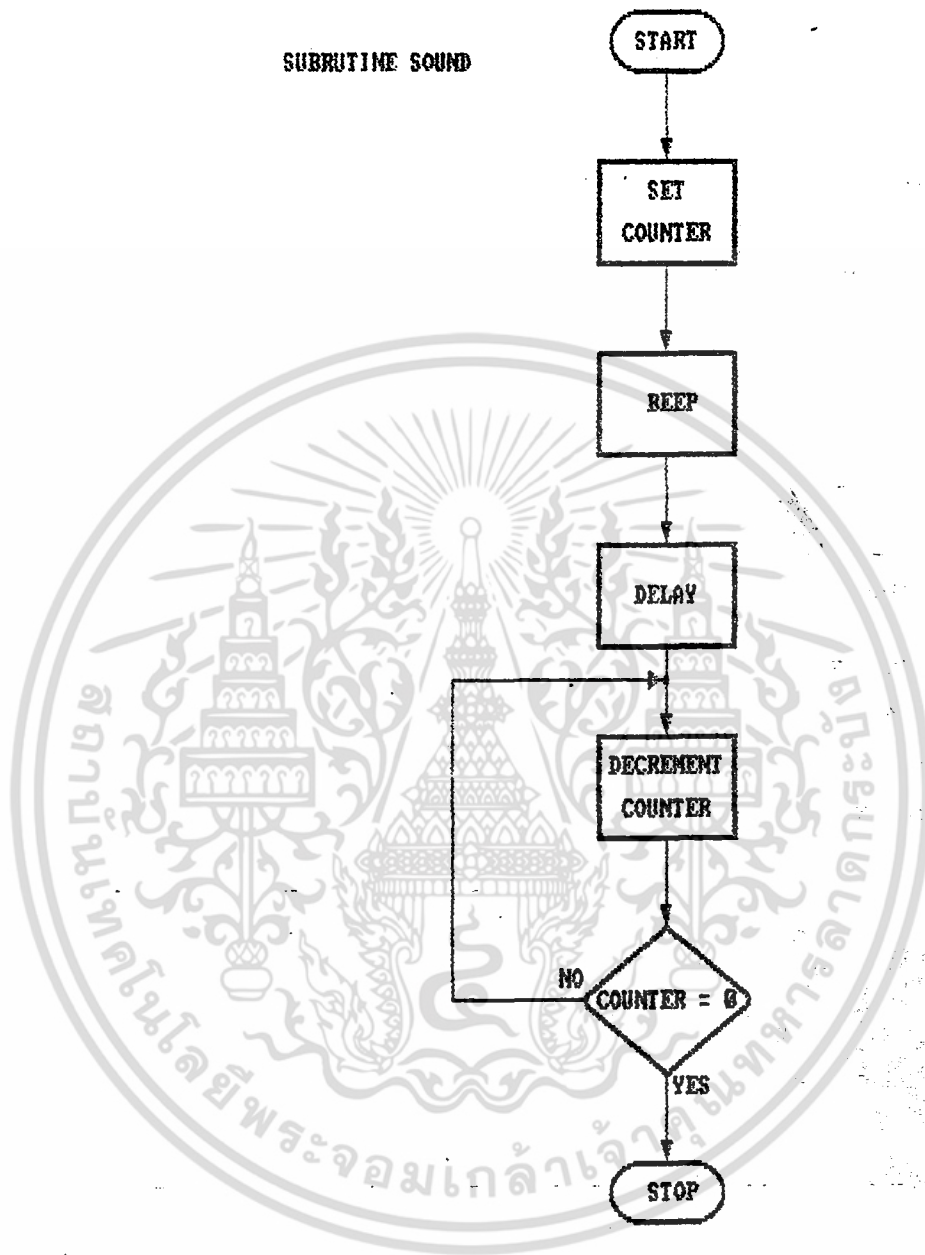
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE BEEP



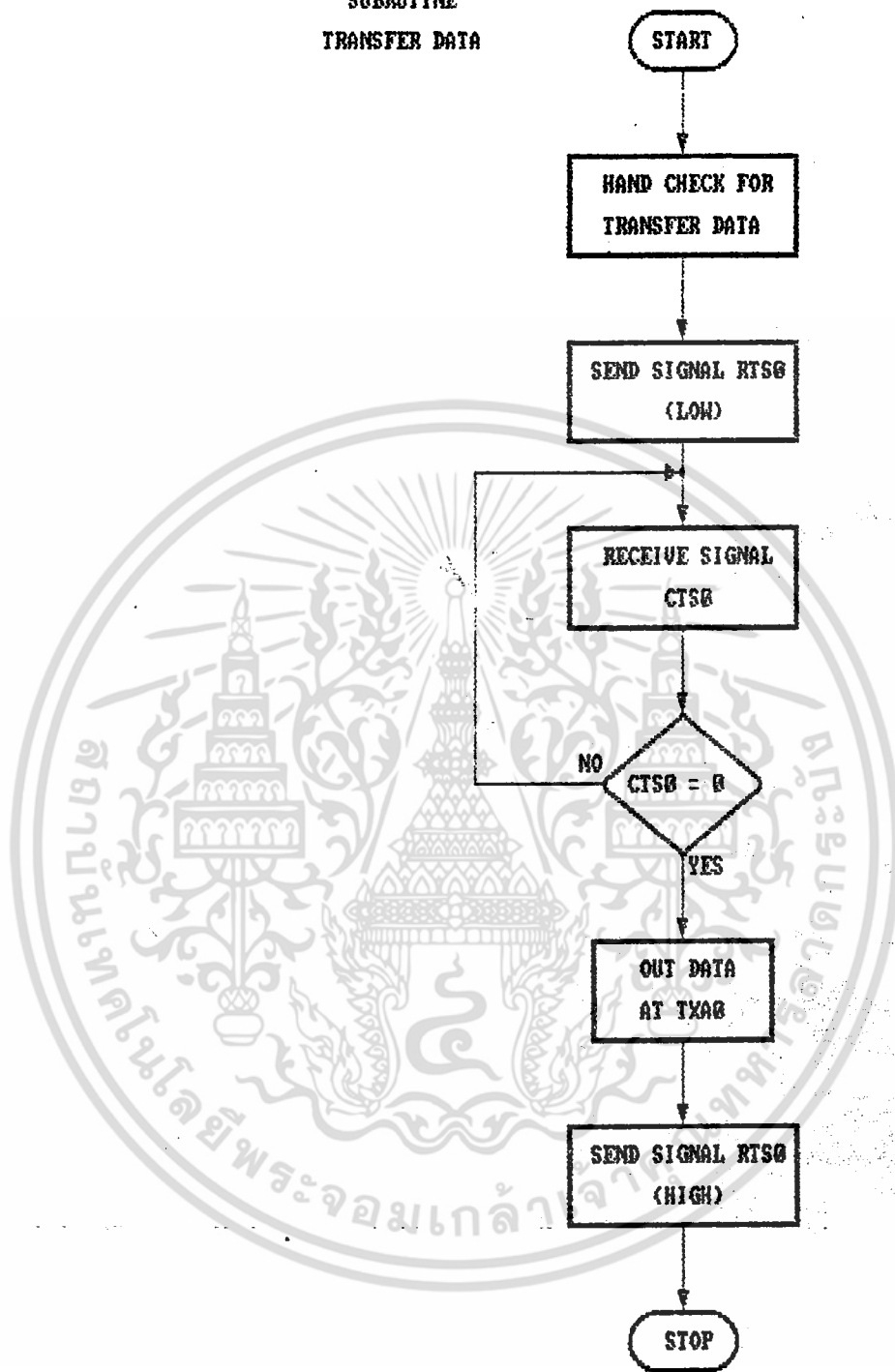
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE SOUND



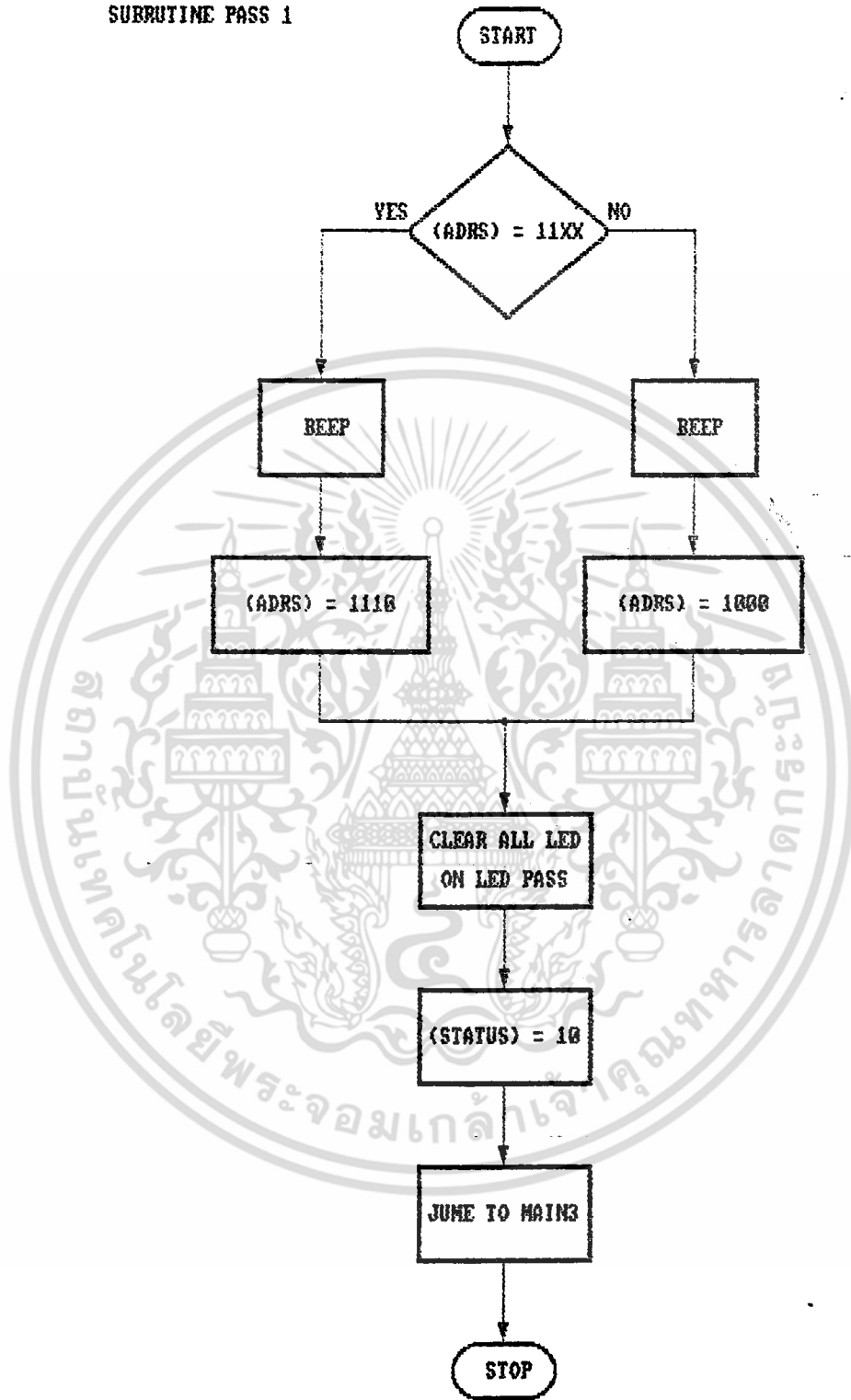
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
TRANSFER DATA



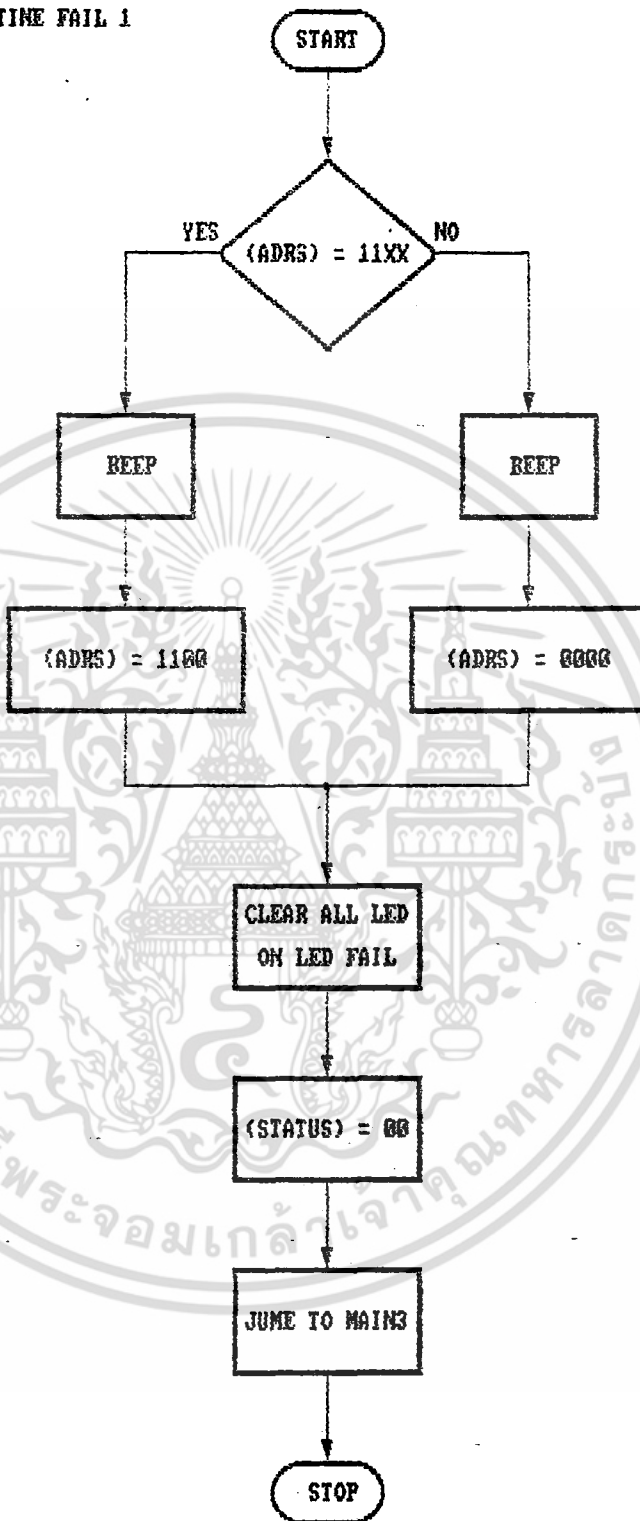
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE PASS 1



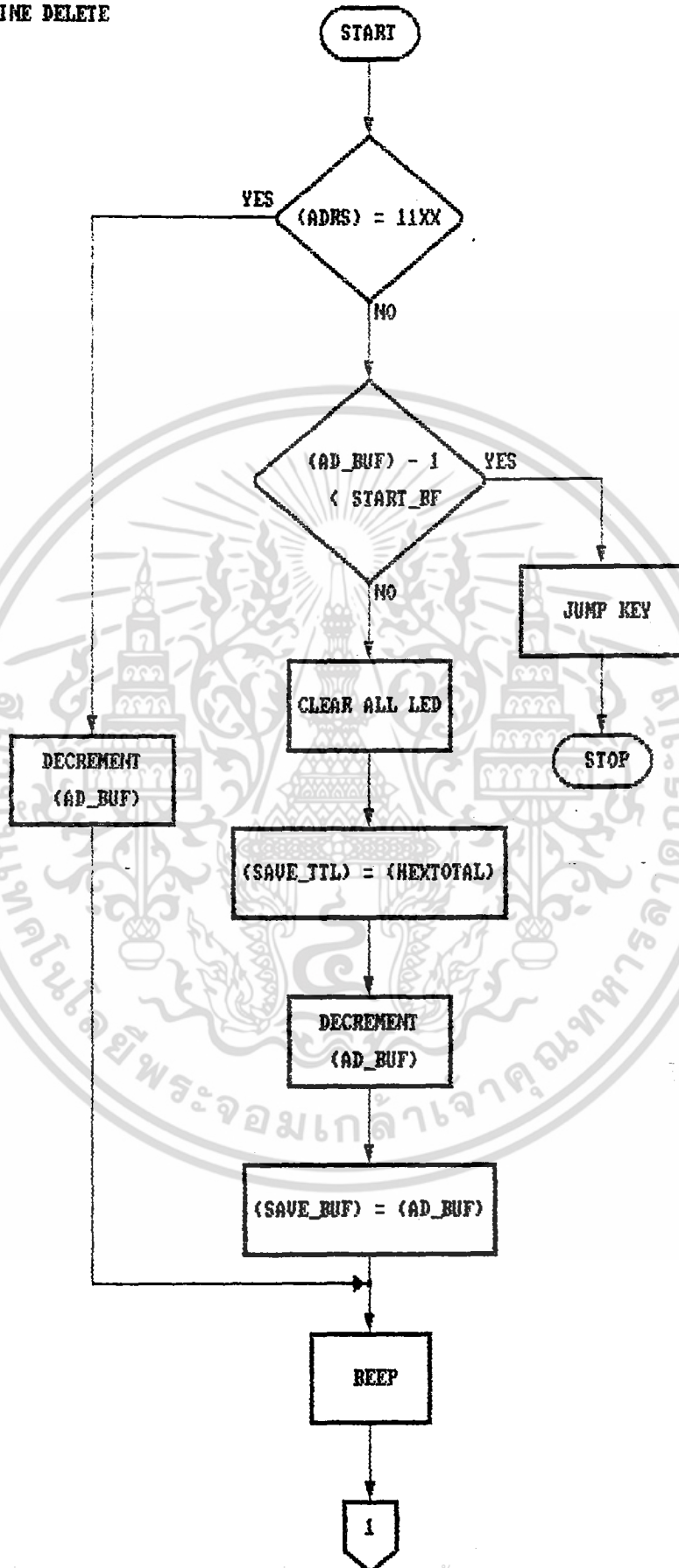
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE FAIL 1

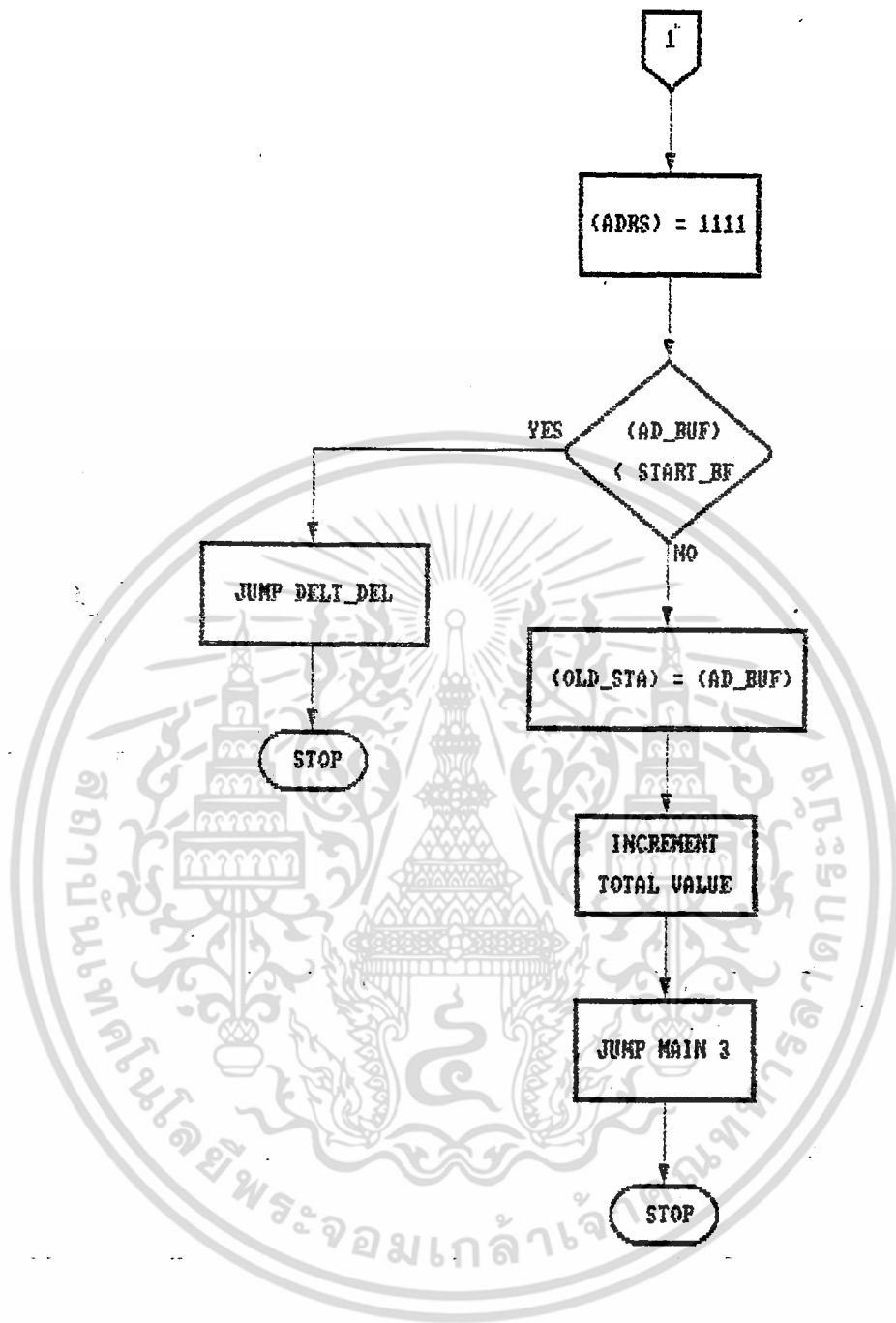


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE DELETE

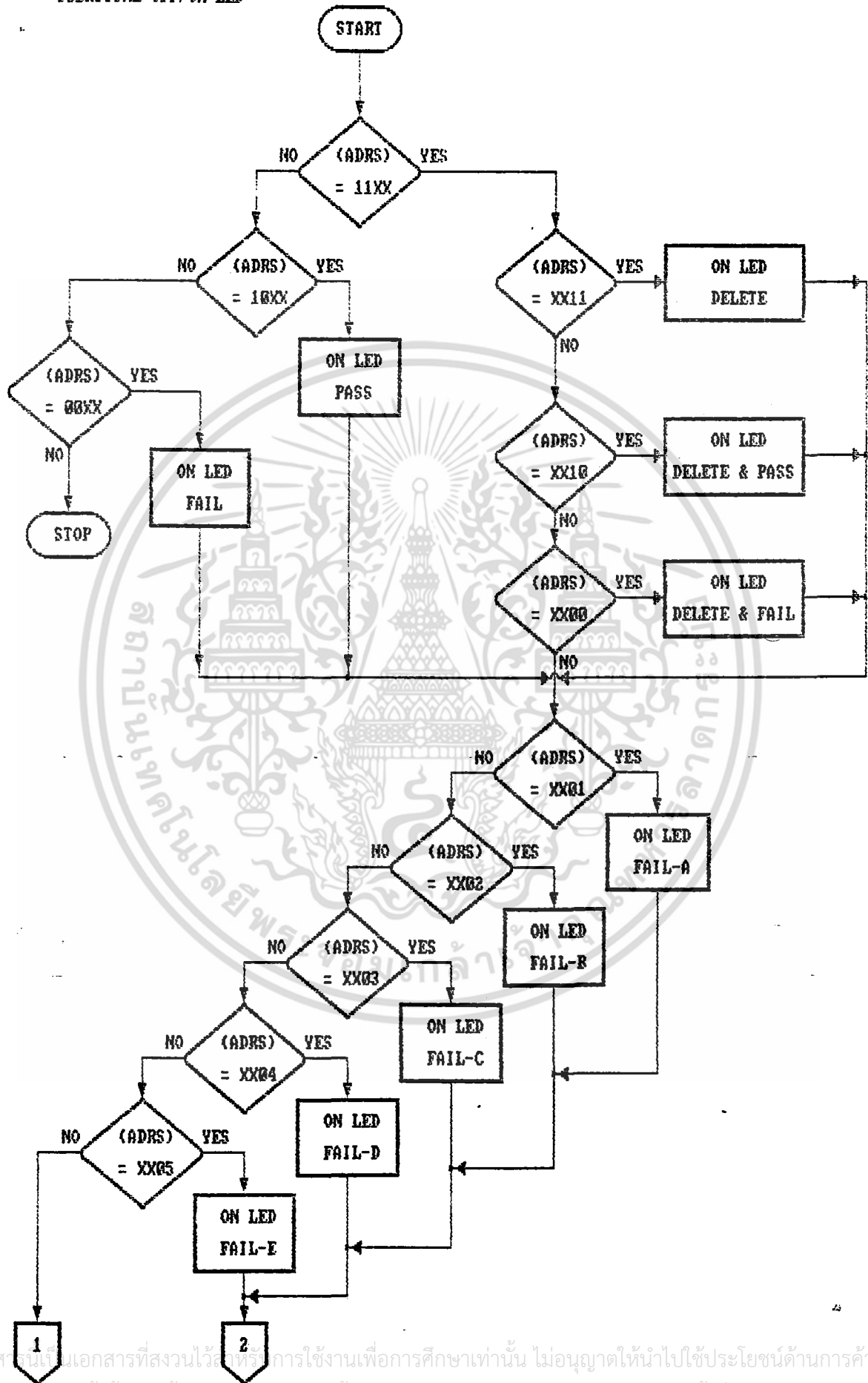


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

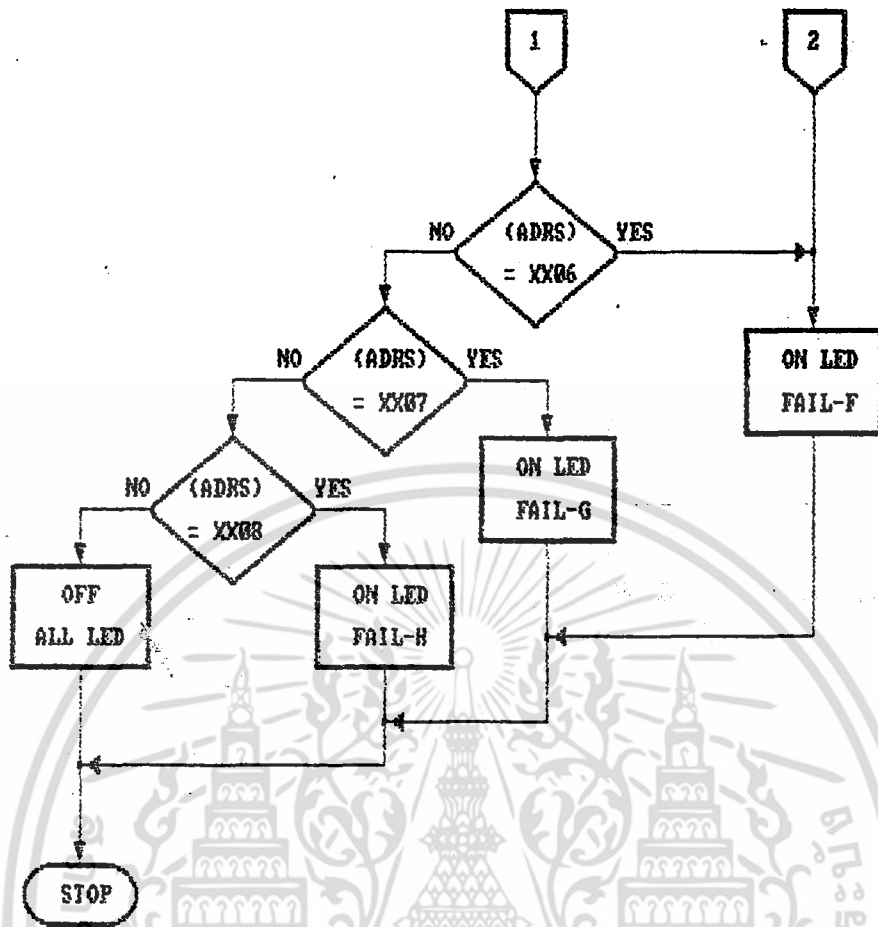


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE OFF/ON LED

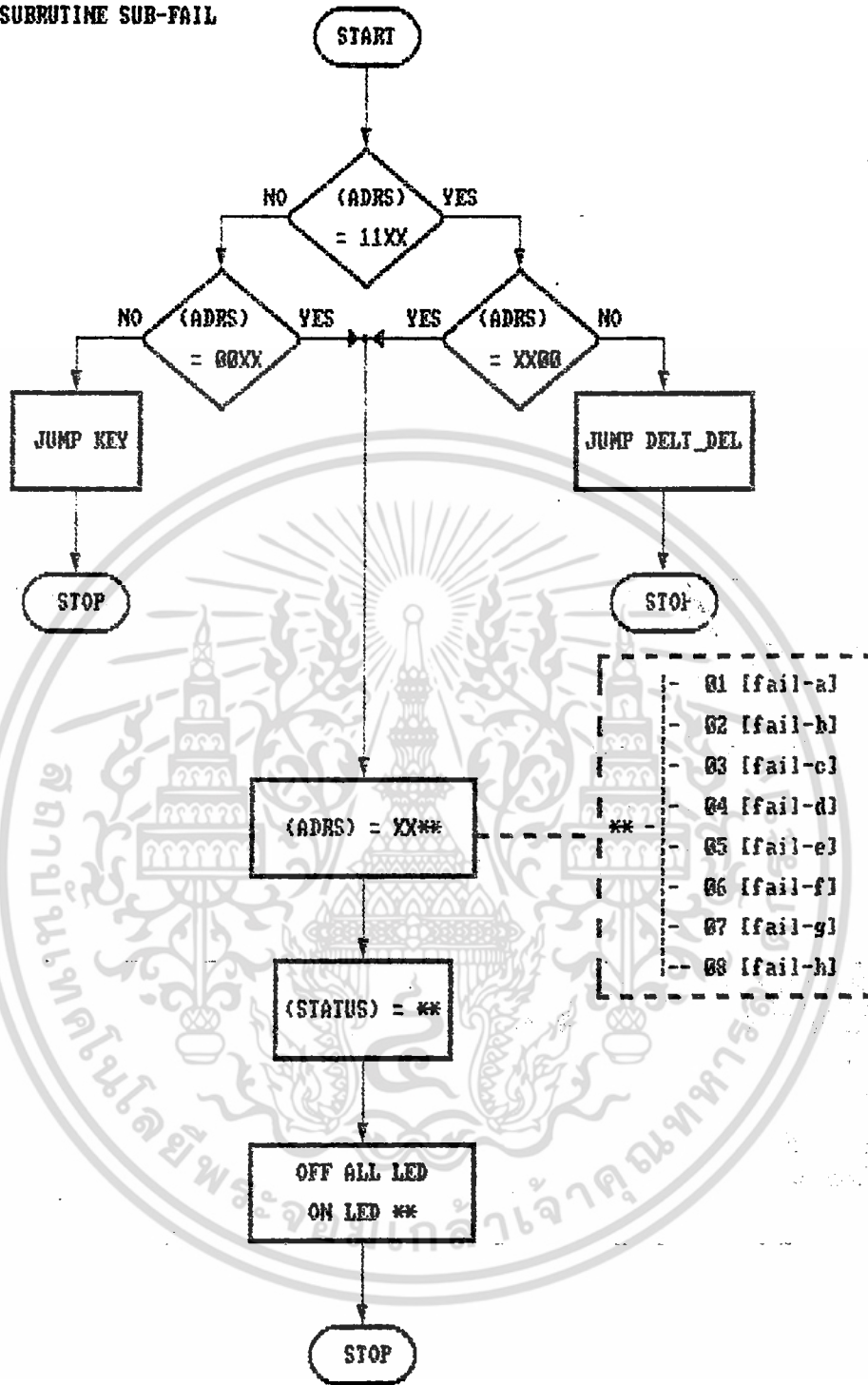


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



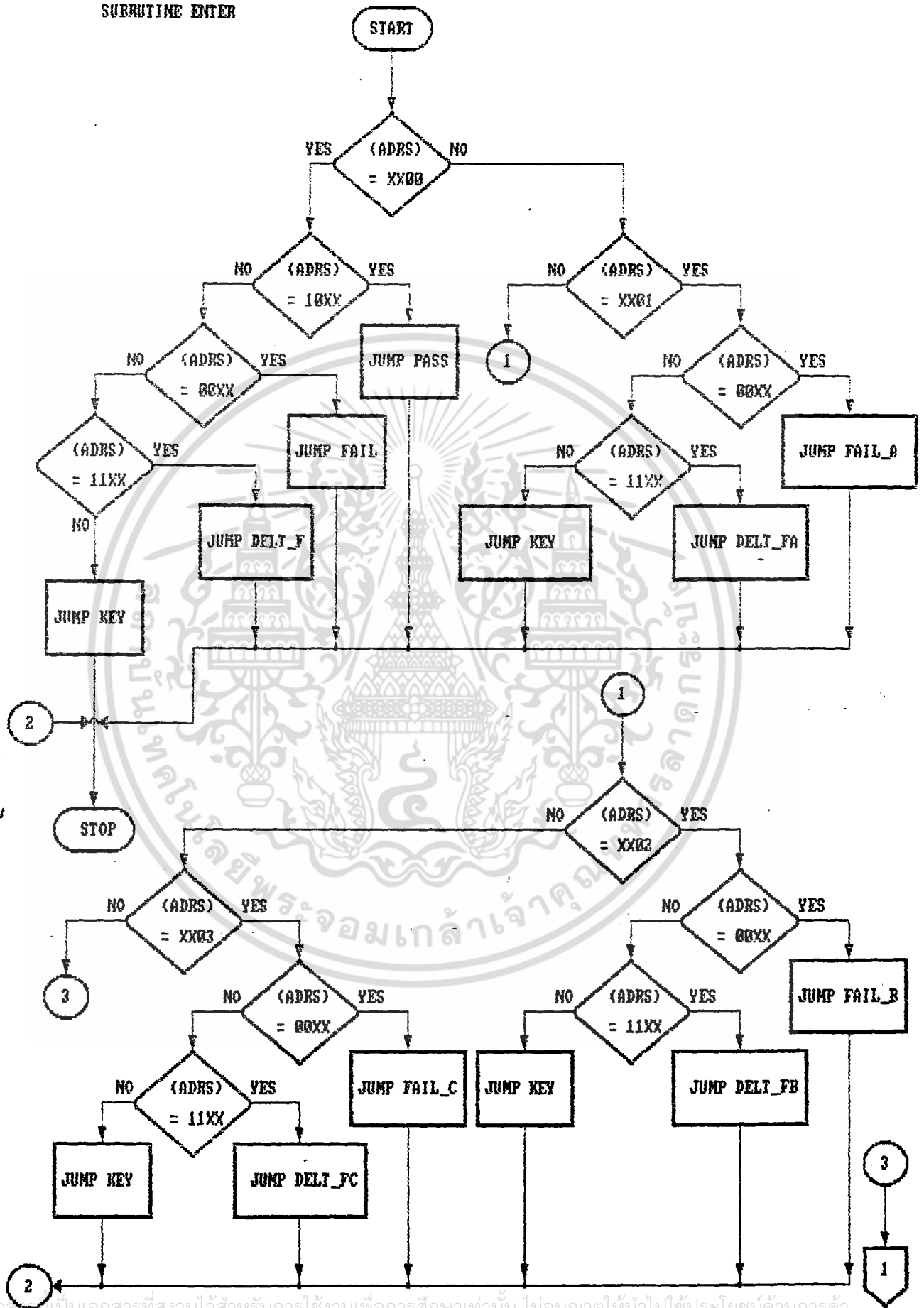
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE SUB-FAIL

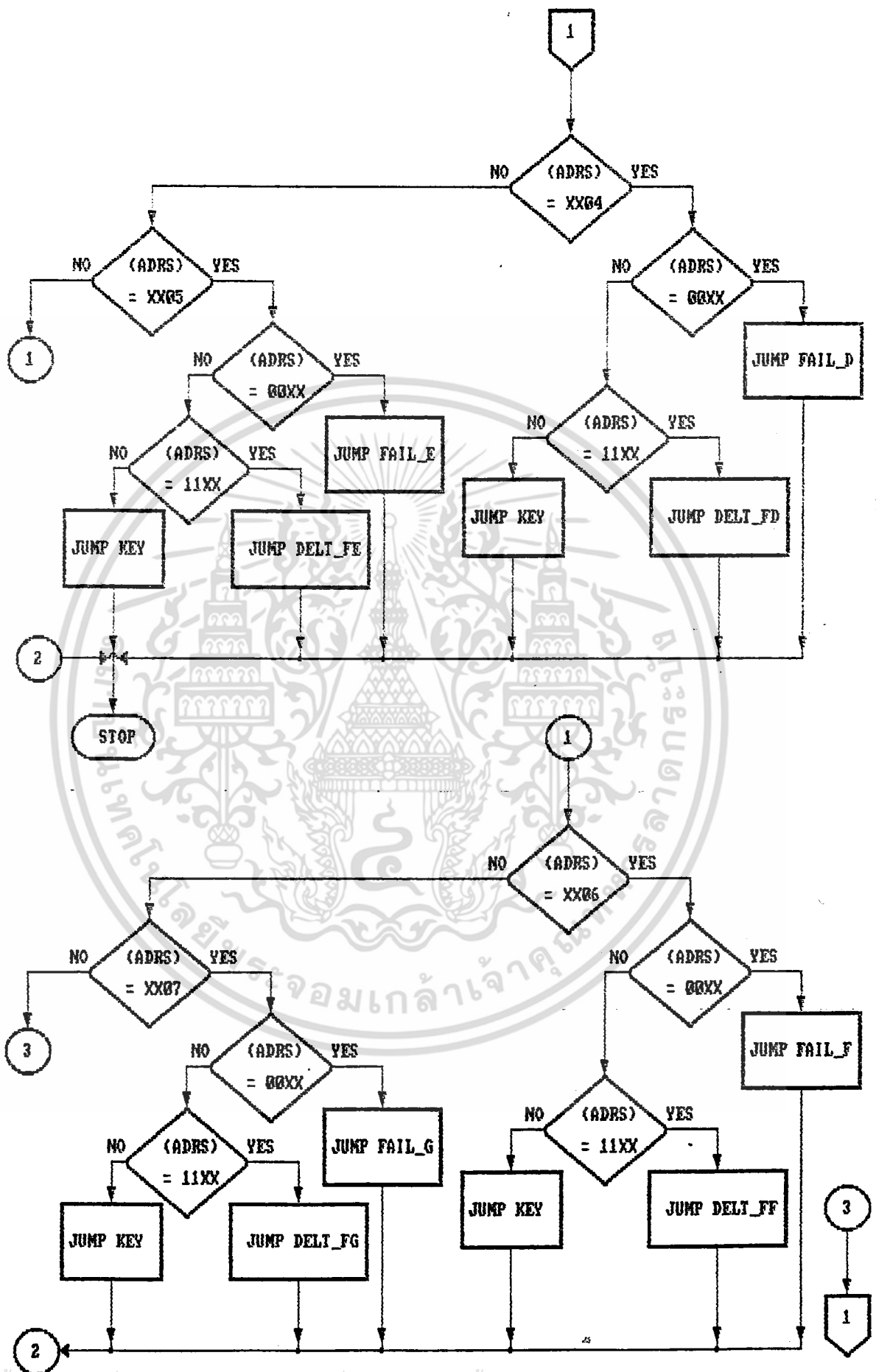


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

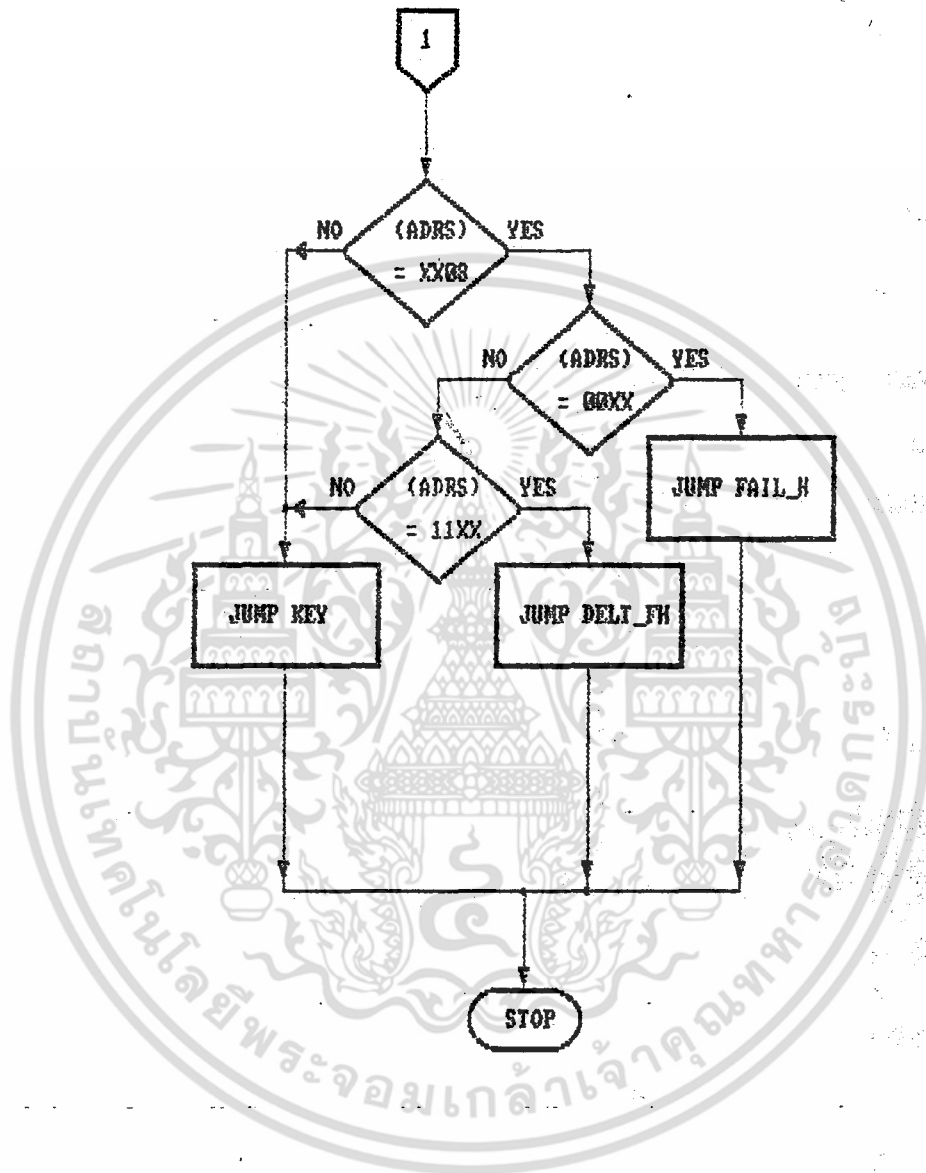
SUBROUTINE ENTER



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

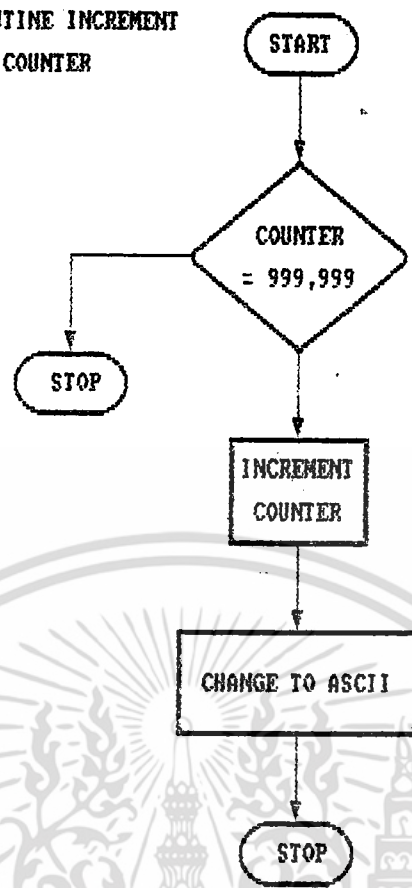


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น. อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

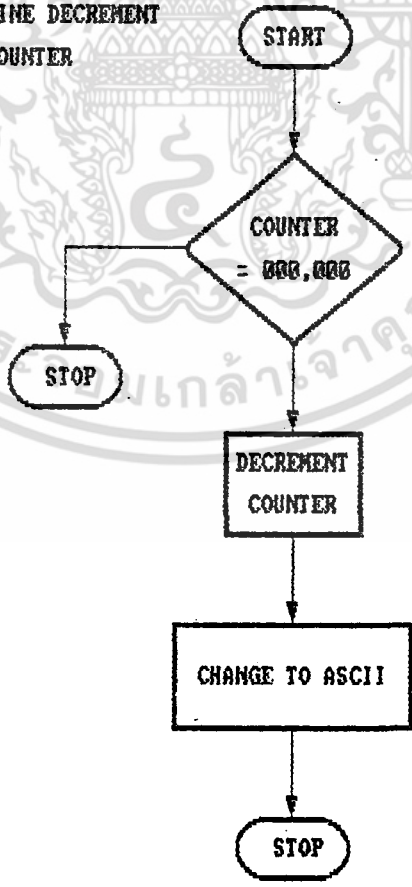


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE INCREMENT
COUNTER

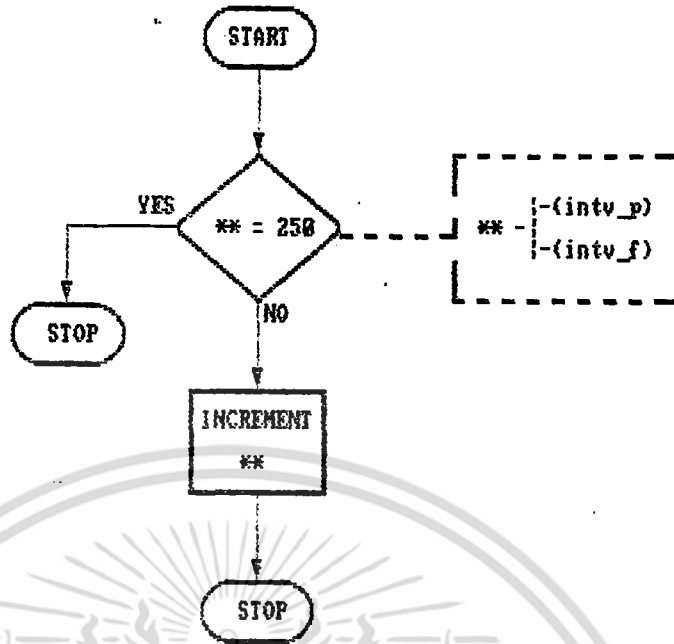


SUBROUTINE DECREMENT
COUNTER

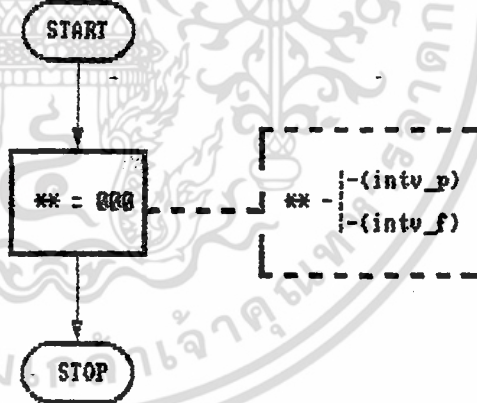


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SUBROUTINE INCREMENT
COUNTER OF INTERVAL**

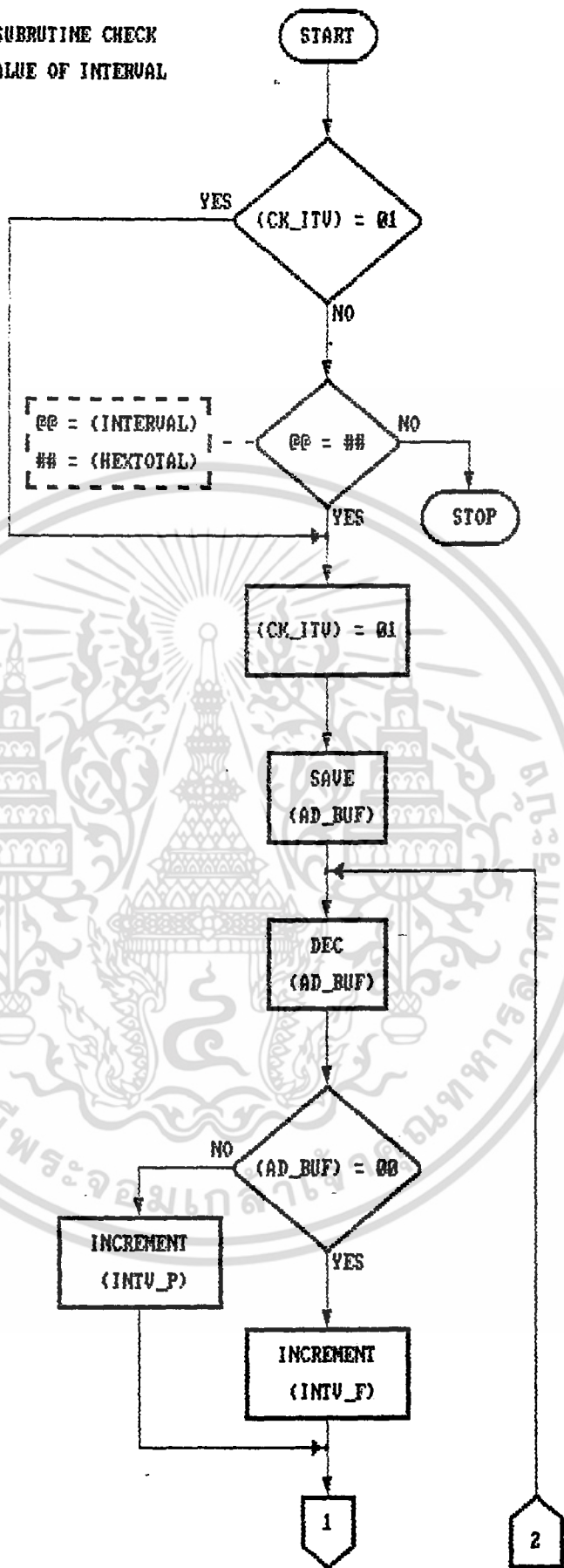


**SUBROUTINE DECREMENT
COUNTER OF INTERVAL**

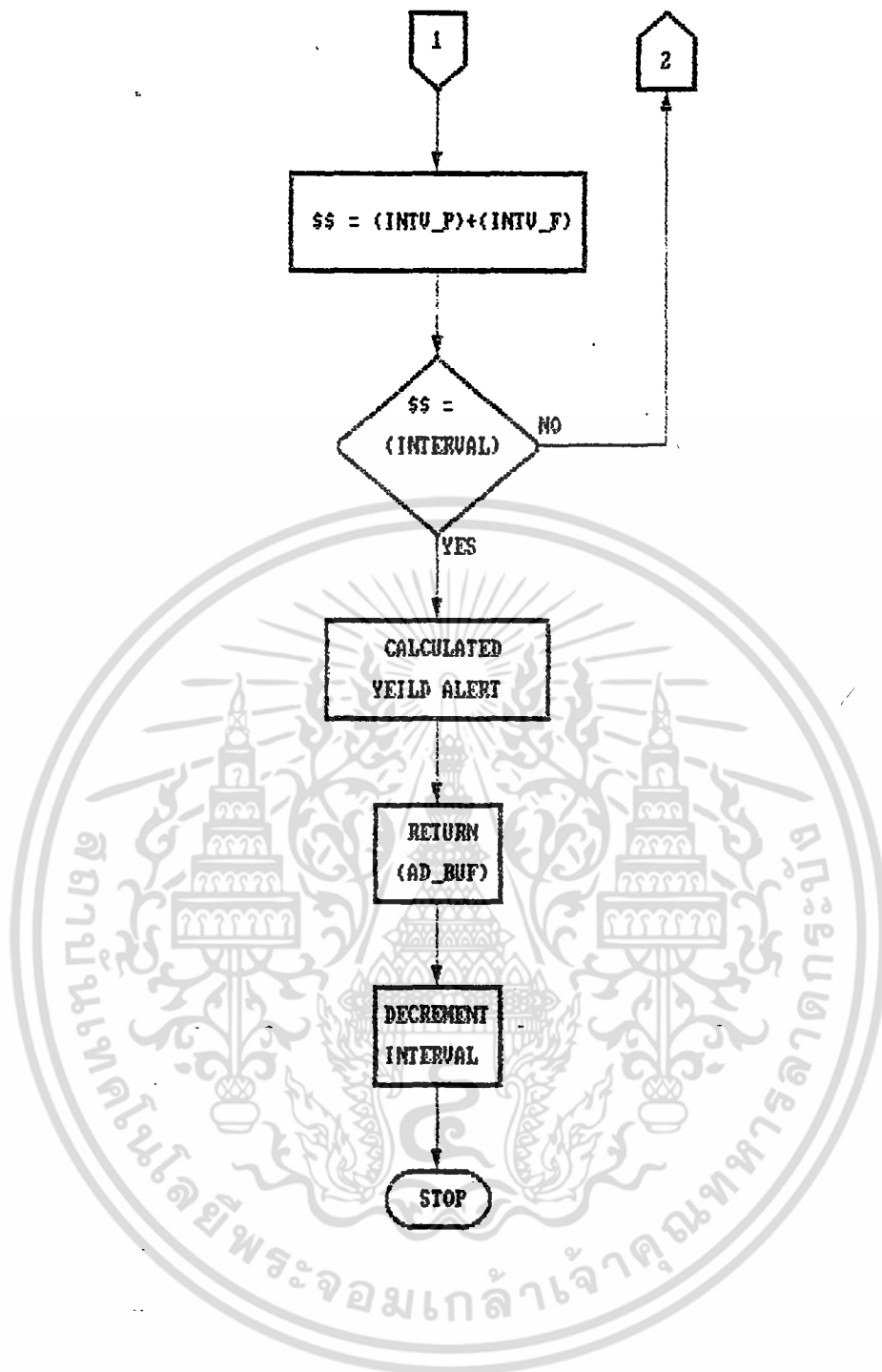


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE CHECK
VALUE OF INTERVAL

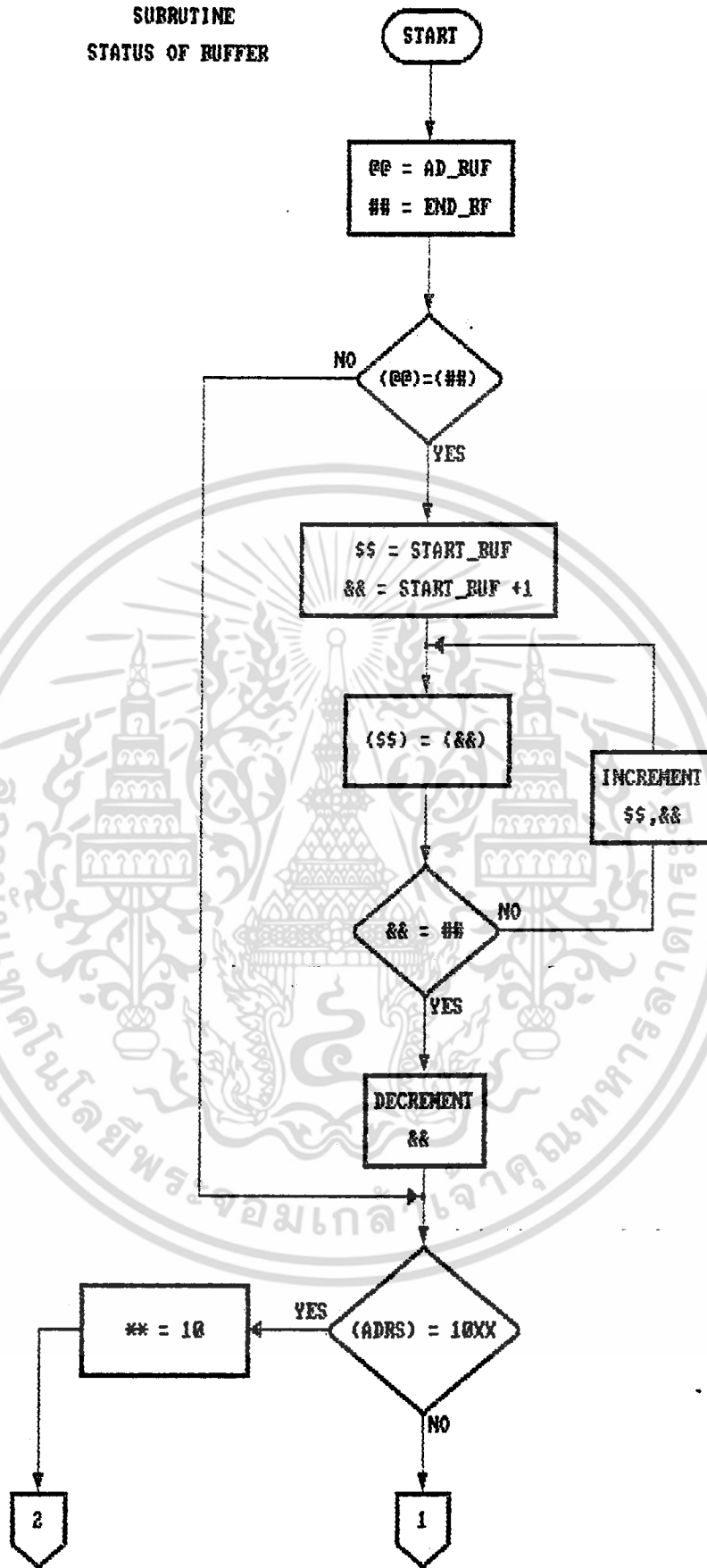


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

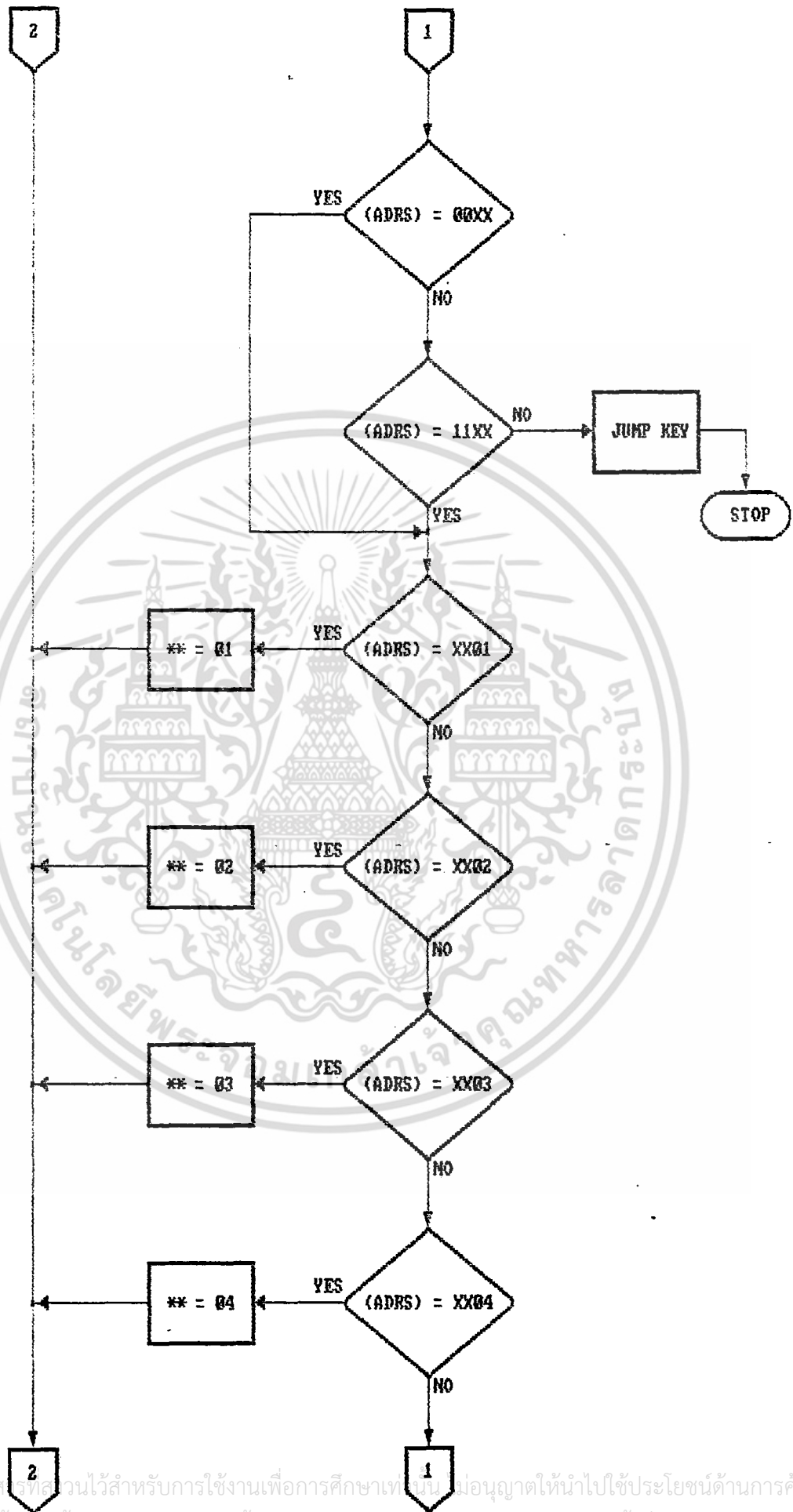


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

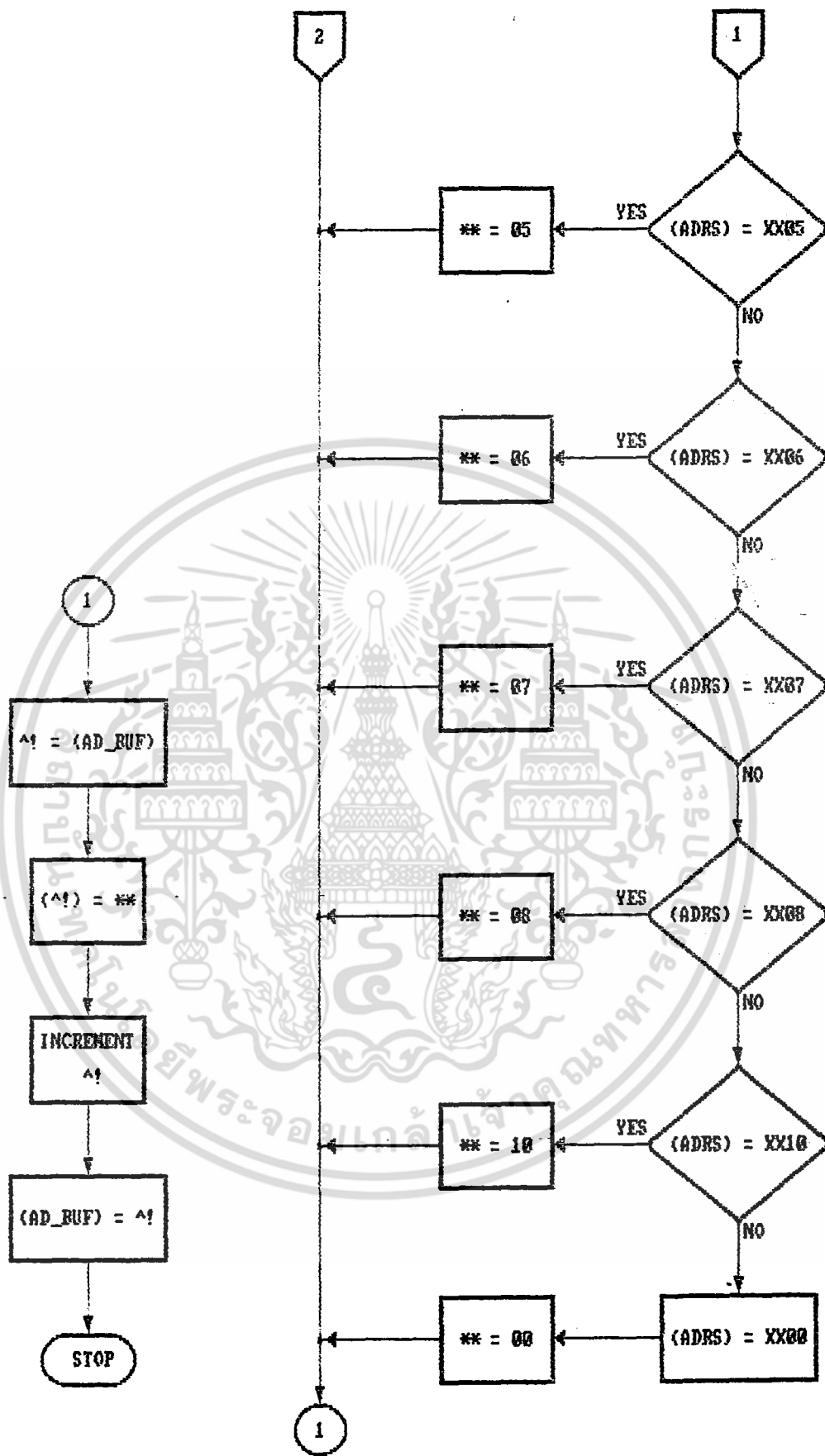
SUBROUTINE
STATUS OF BUFFER



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

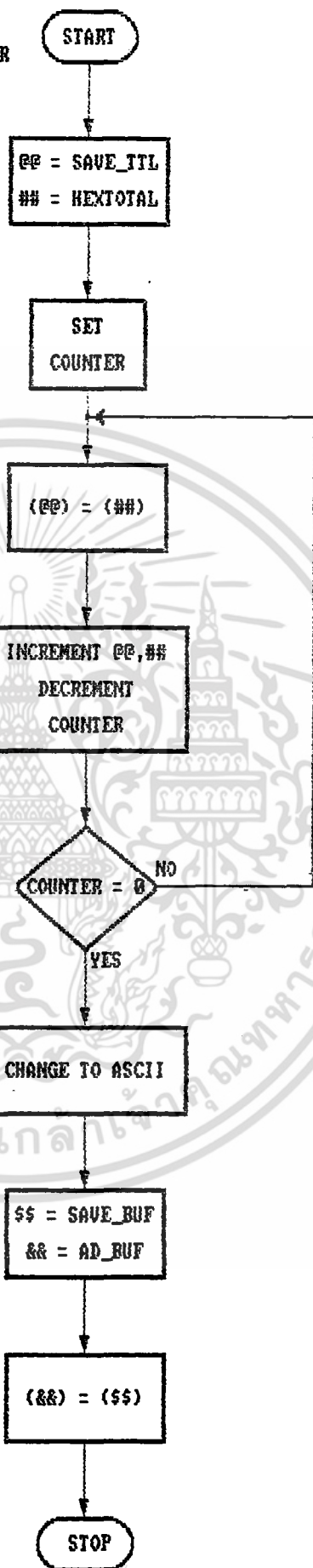


เอกสารนี้เป็นเอกสารที่สละนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



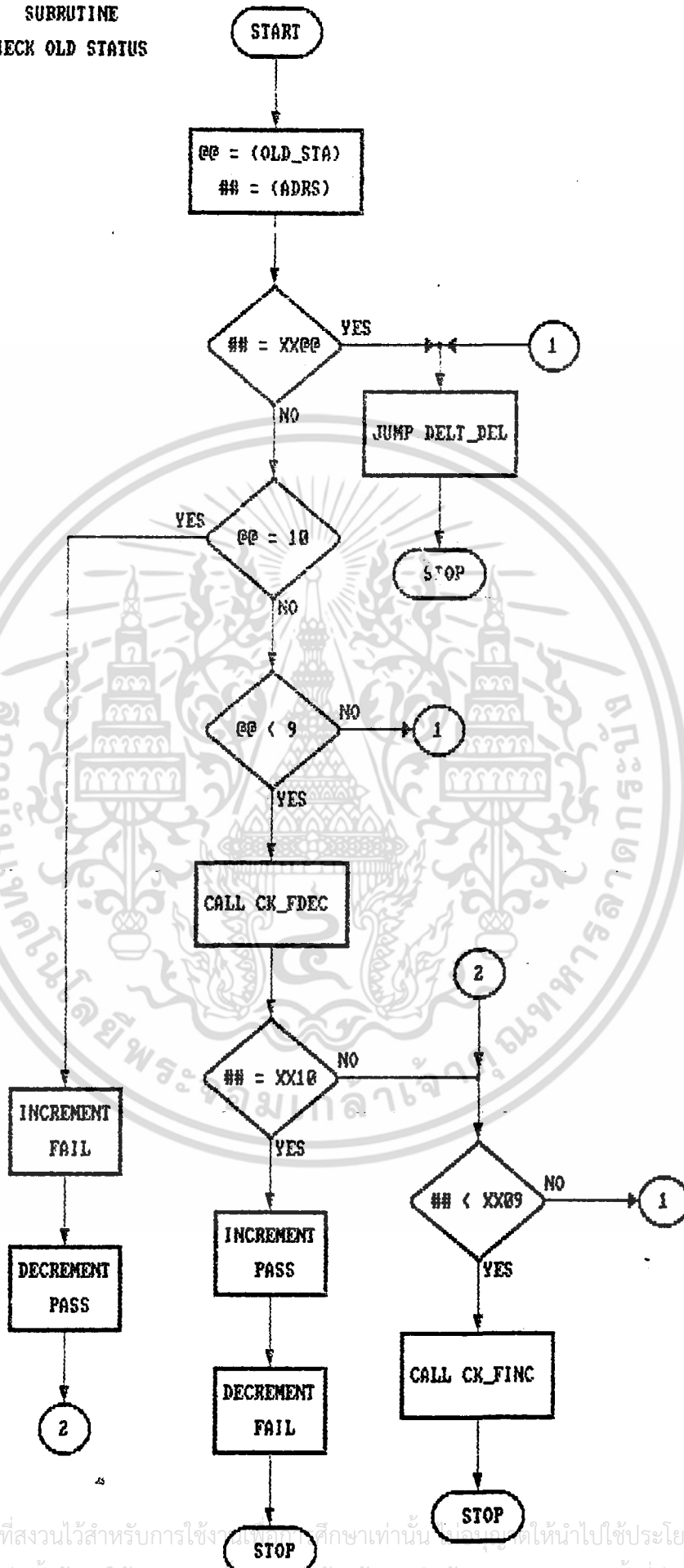
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
OLD VALUE OF BUFFER



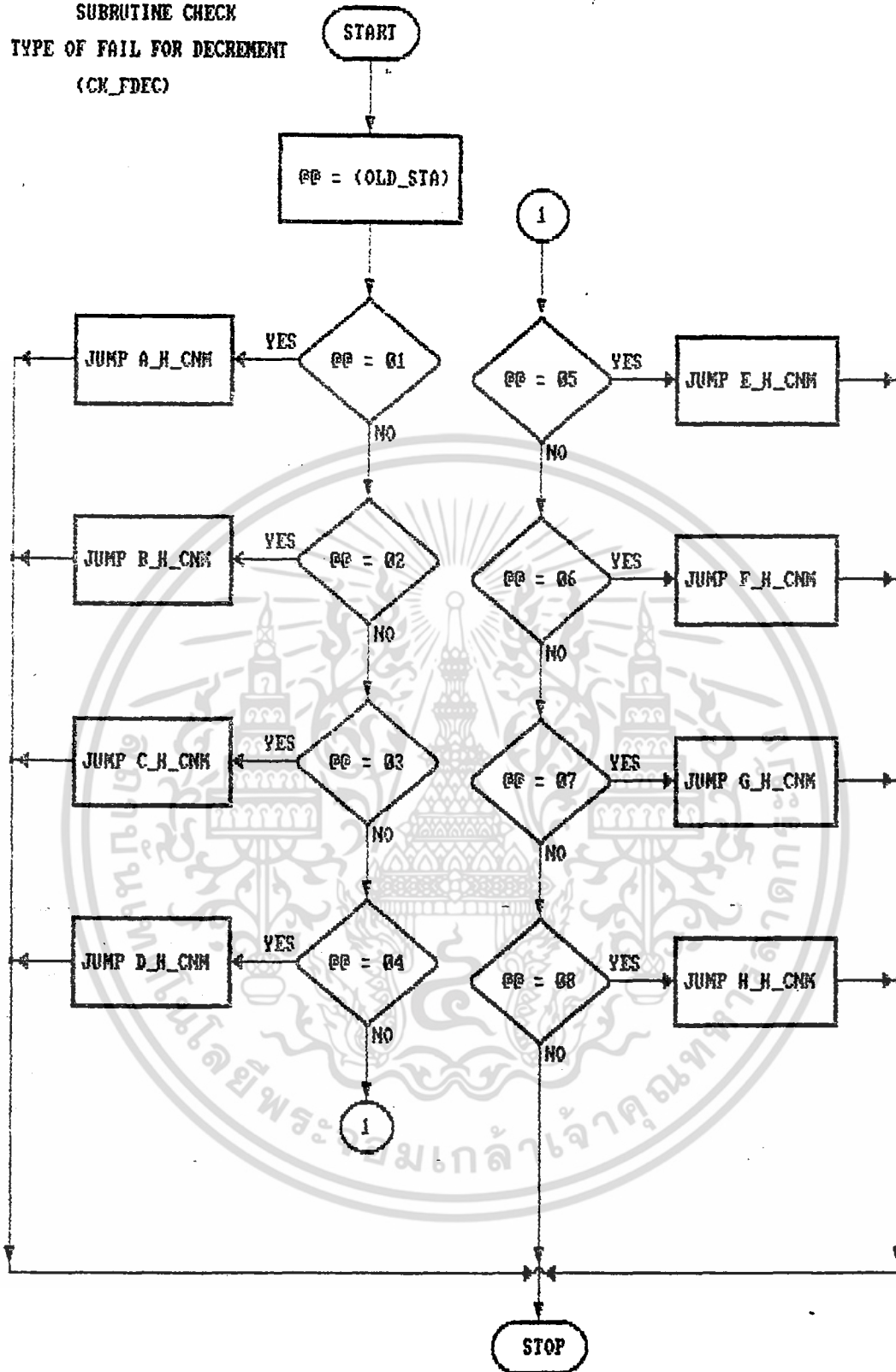
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
CHECK OLD STATUS



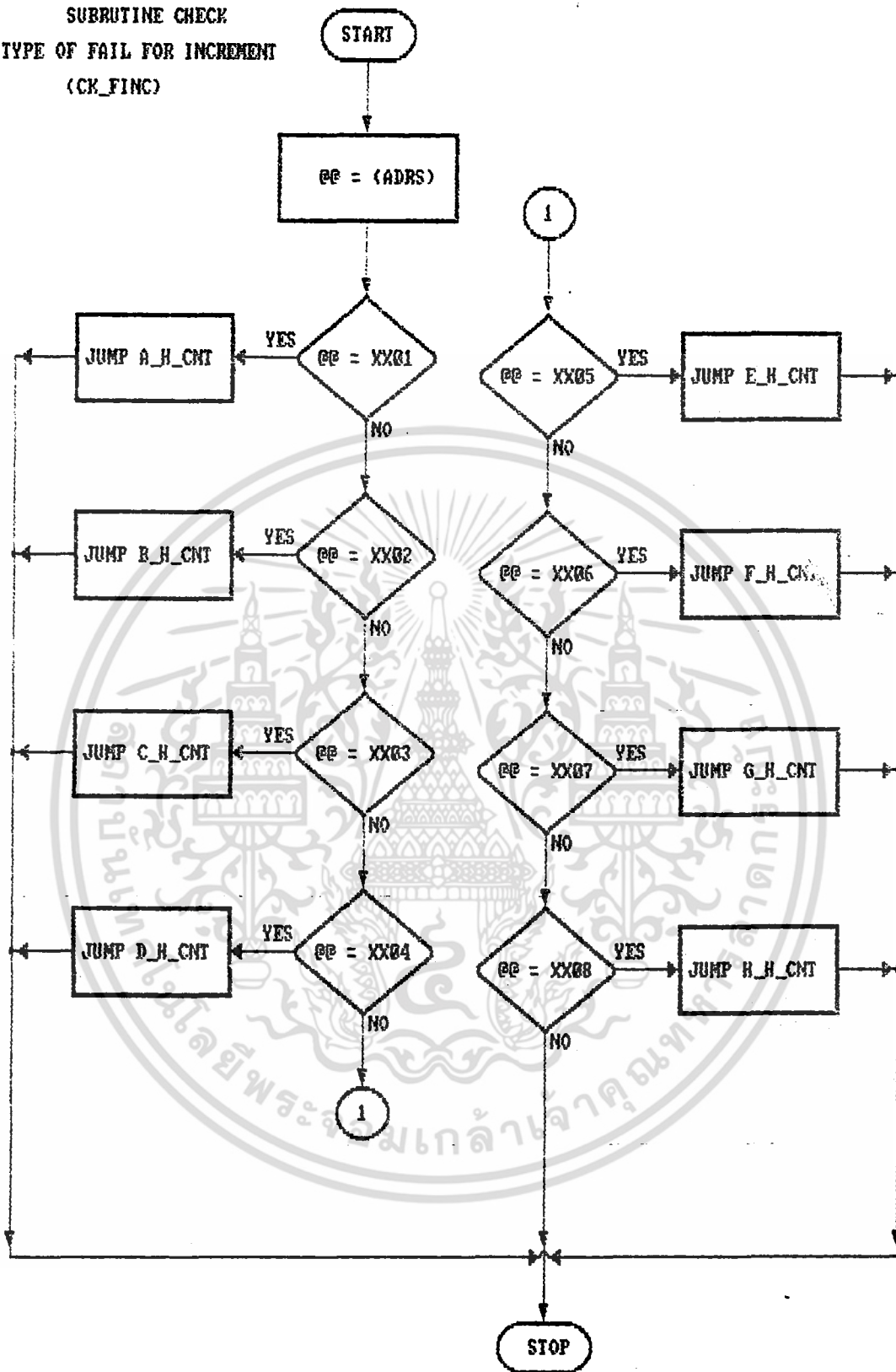
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE CHECK
TYPE OF FAIL FOR DECREMENT
(CK_FDEC)



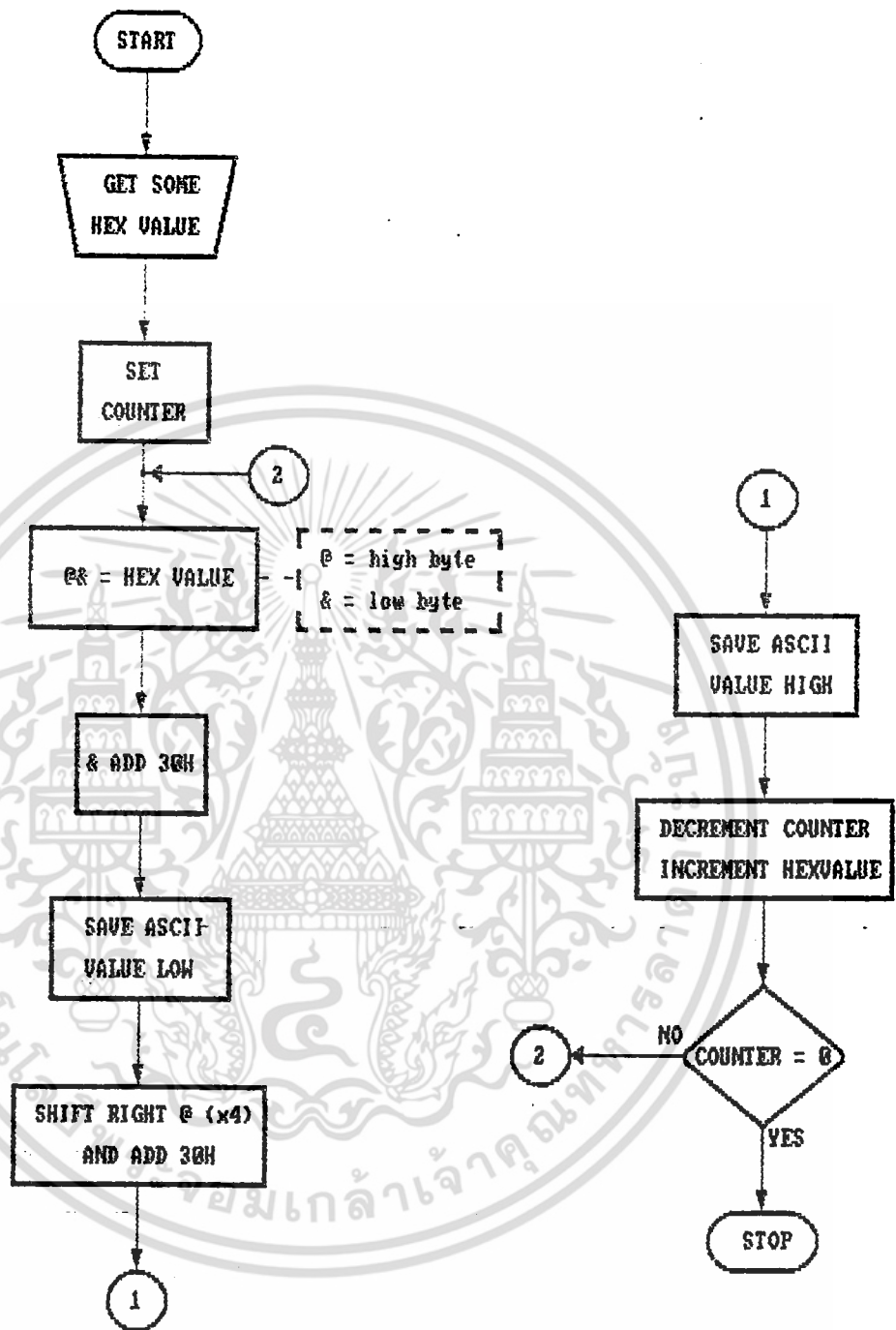
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE CHECK
TYPE OF FAIL FOR INCREMENT
(CK_FINC)



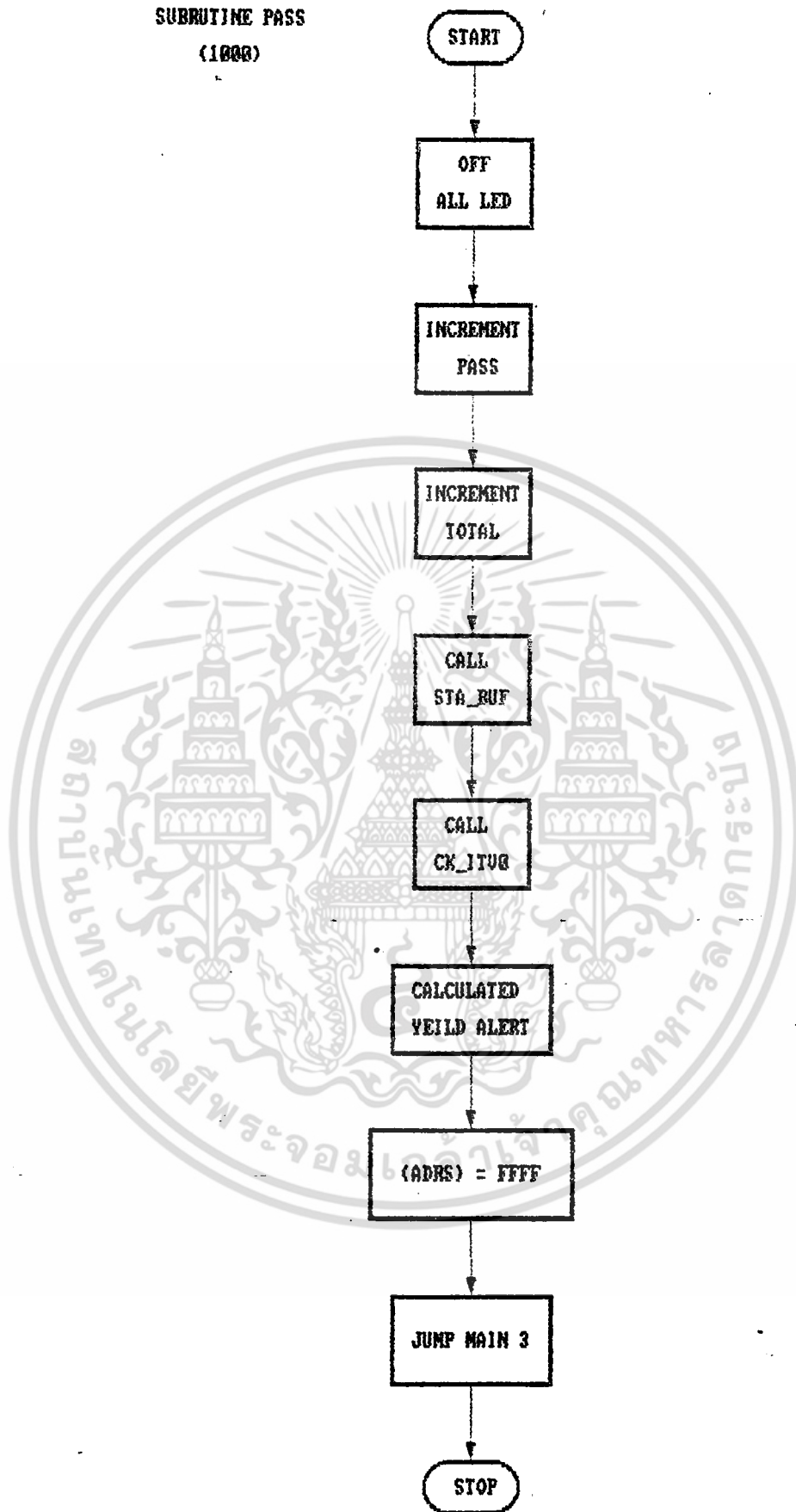
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE CHANGE
TO ASCII



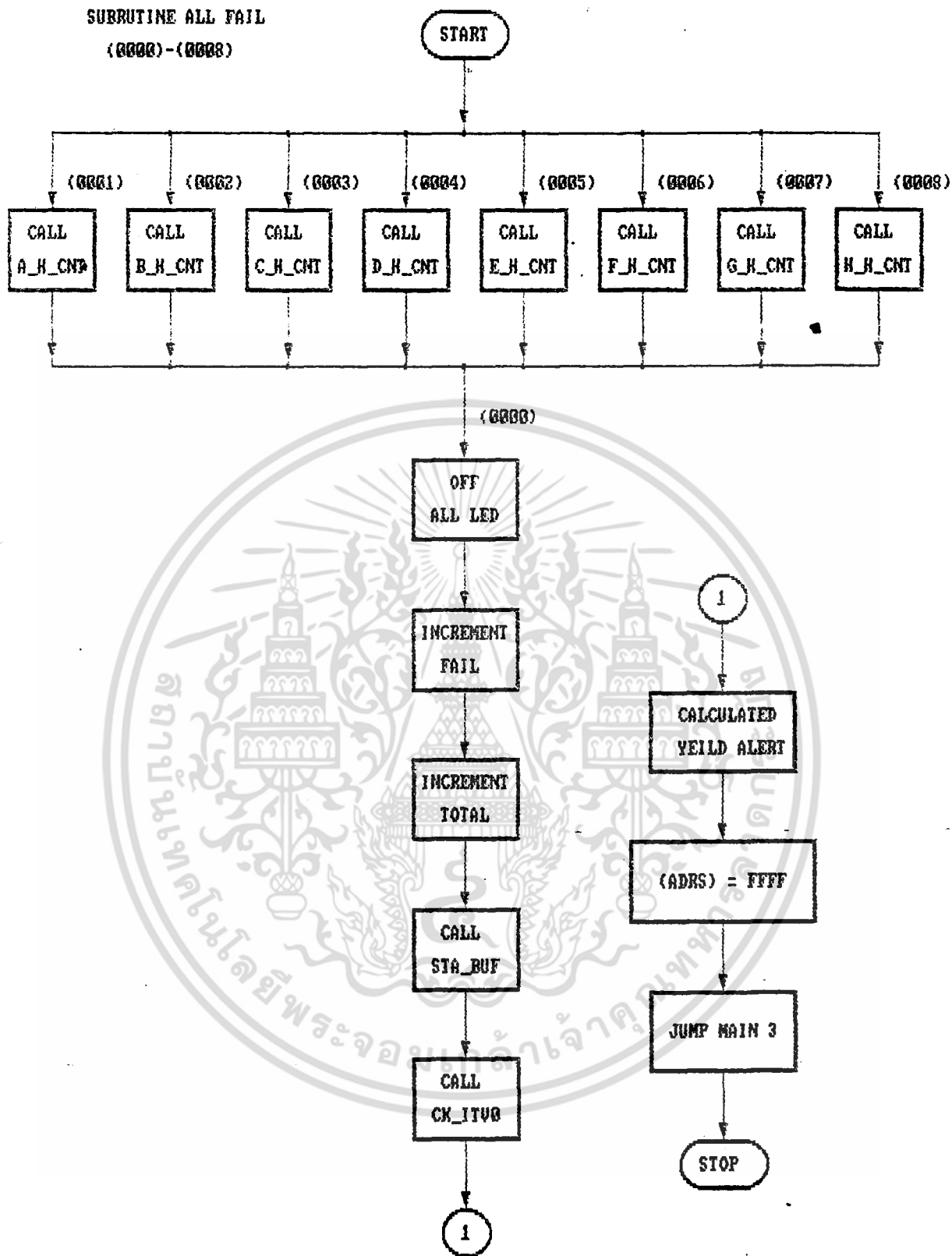
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE PASS
(1008)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE ALL FAIL
(0000)-(0008)

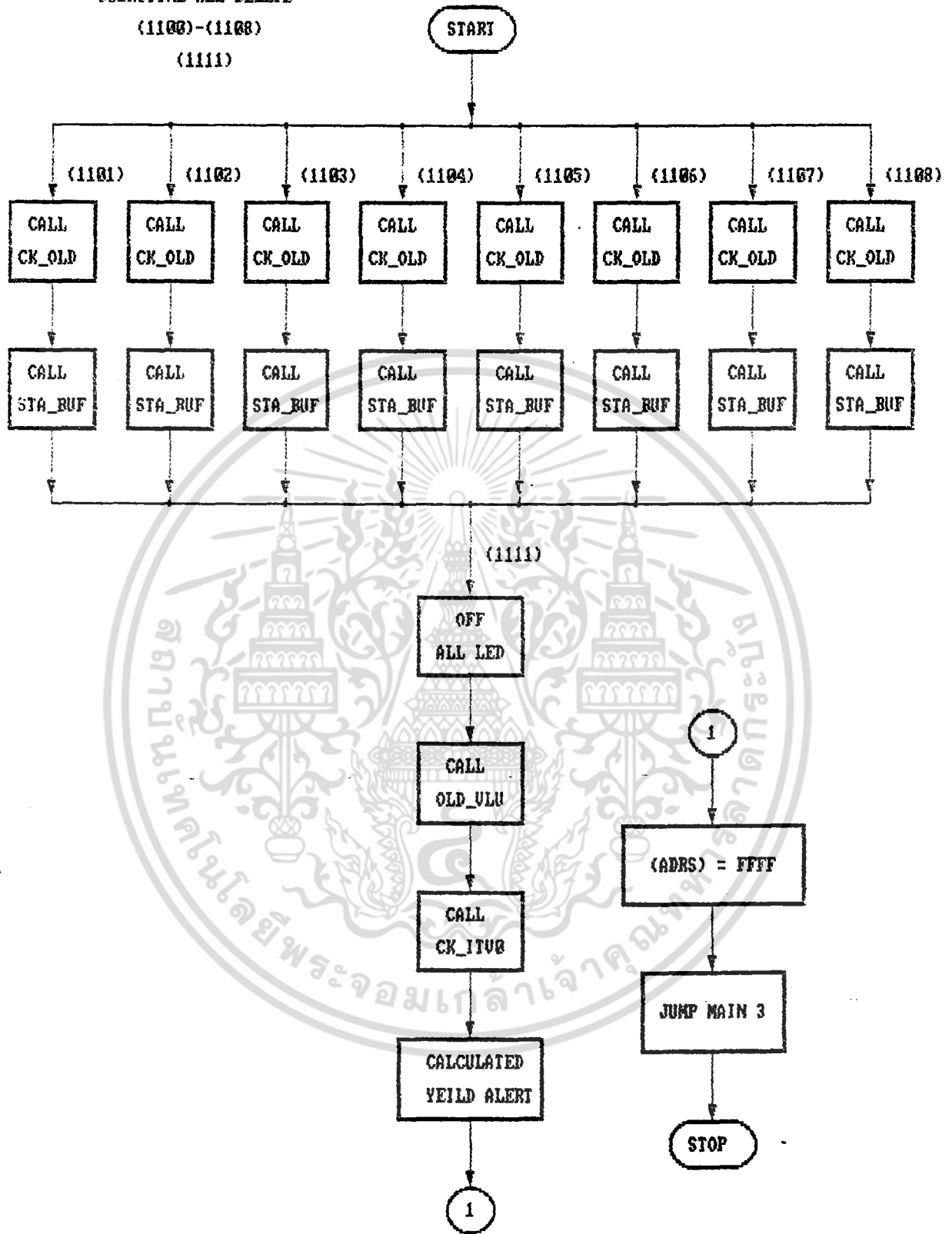


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE ALL DELETE

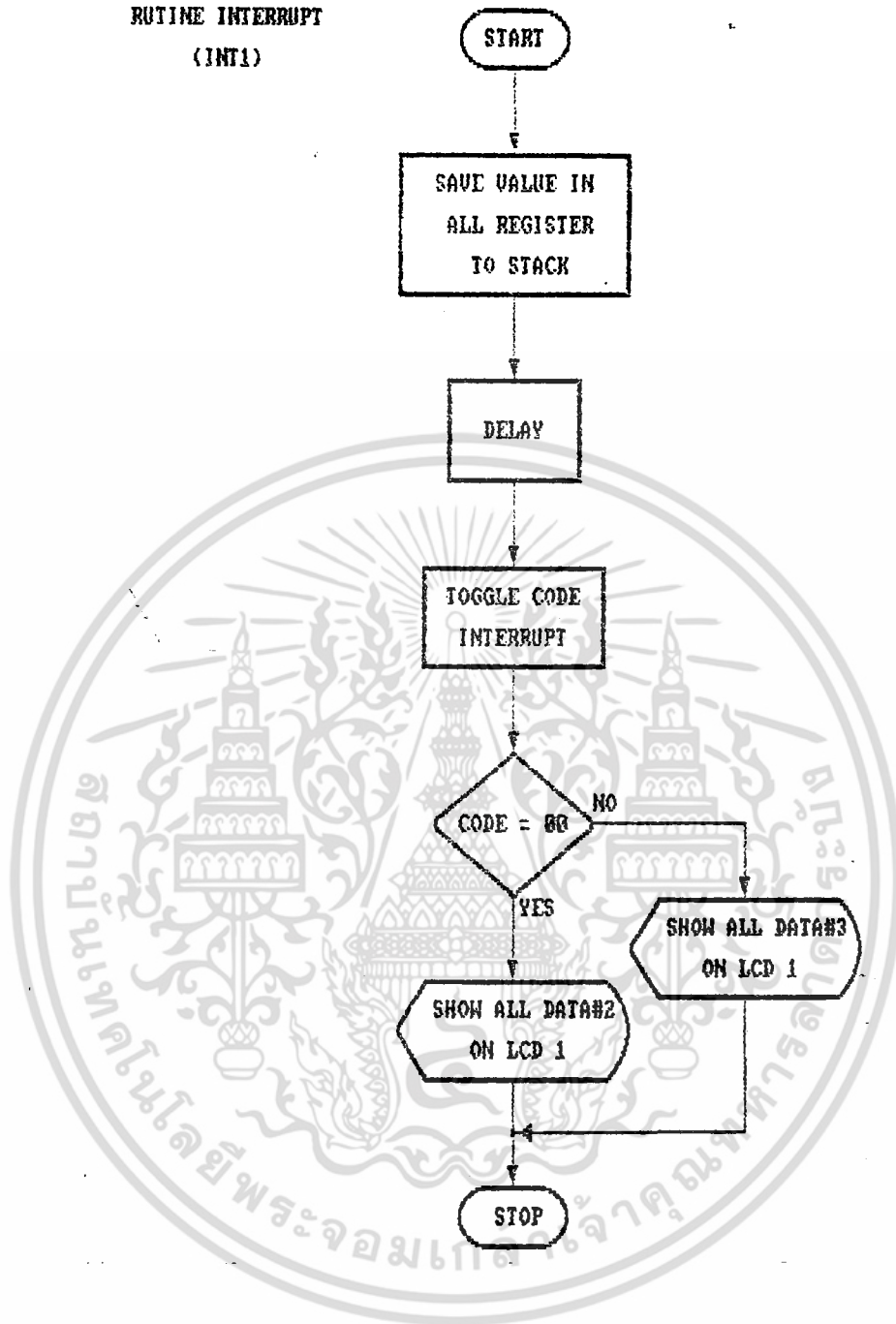
(1100)-(1108)

(1111)



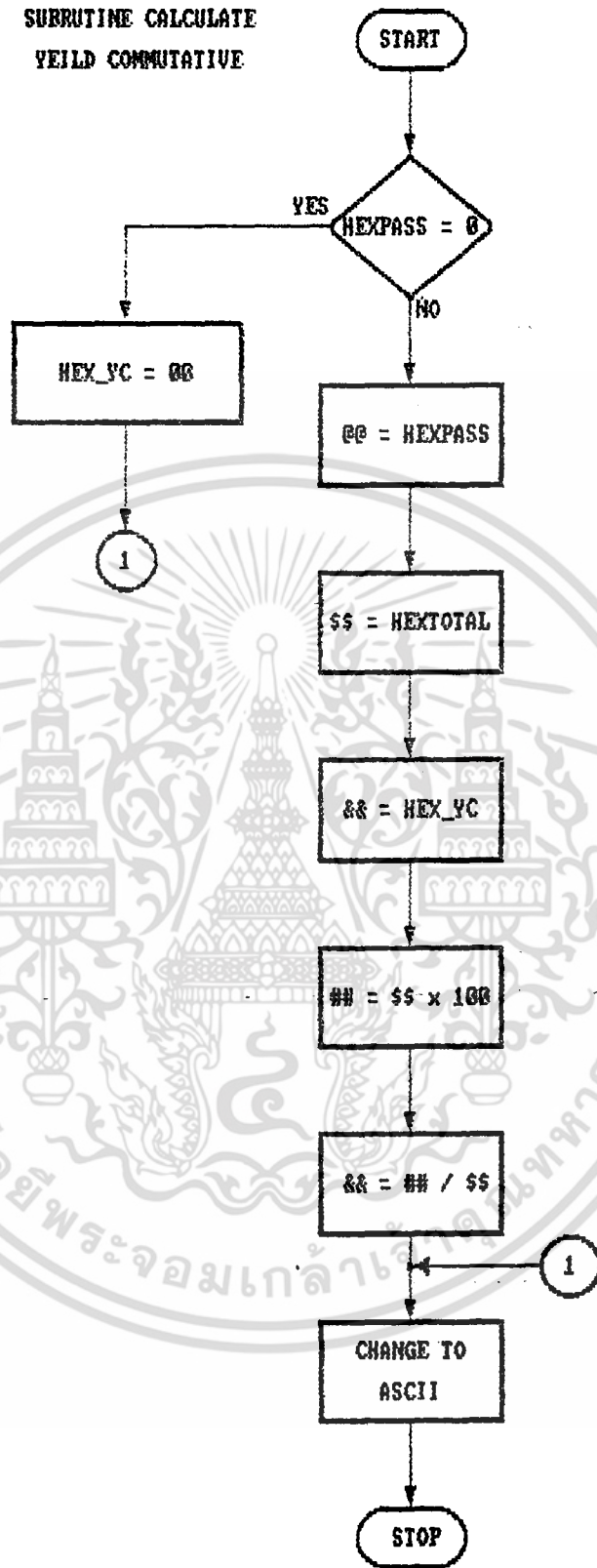
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RUTINE INTERRUPT
(INT1)



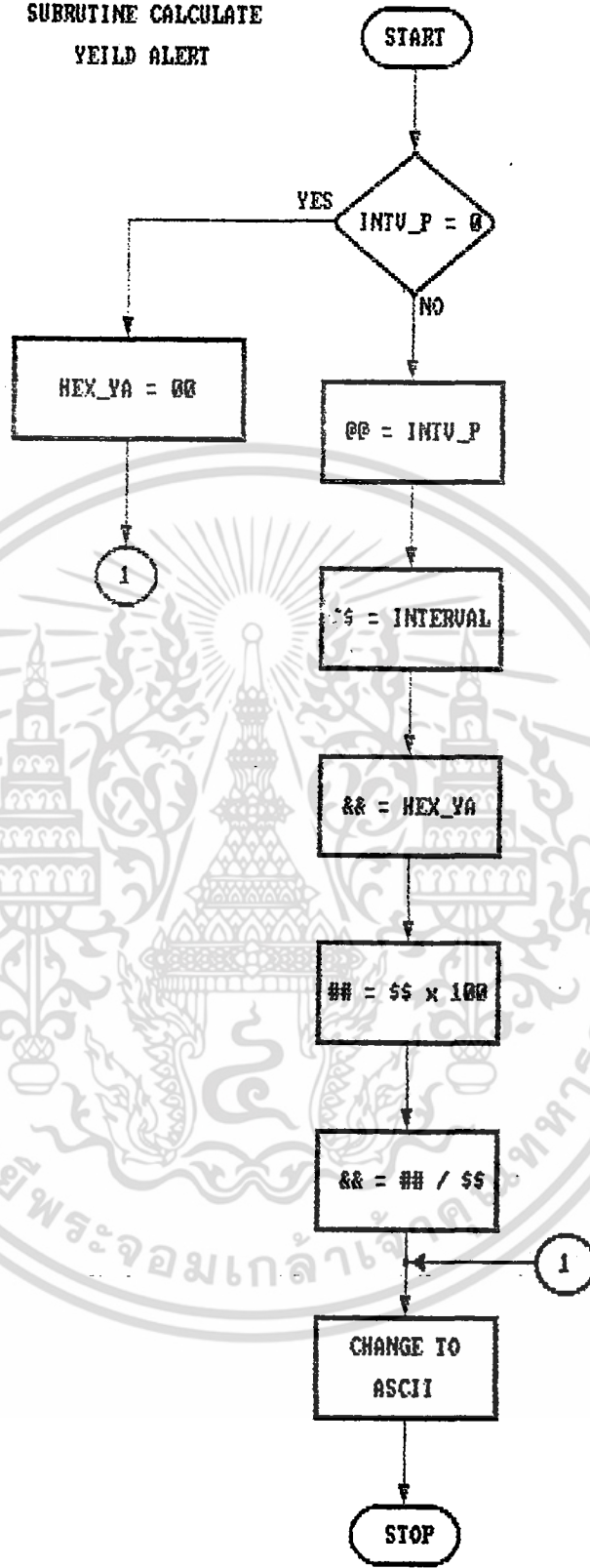
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE CALCULATE
YEILD COMPUTATIVE



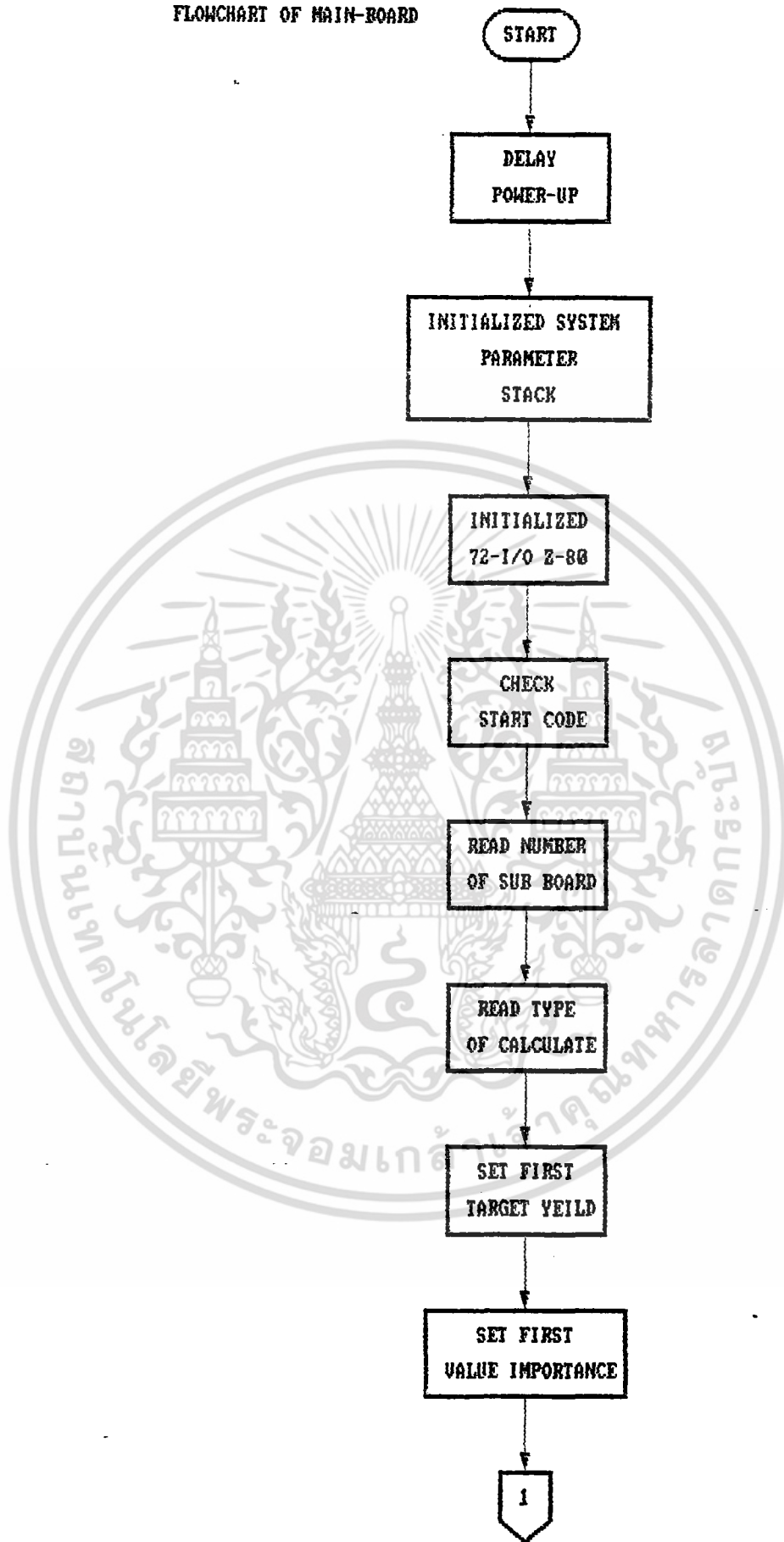
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE CALCULATE
YEILD ALERT

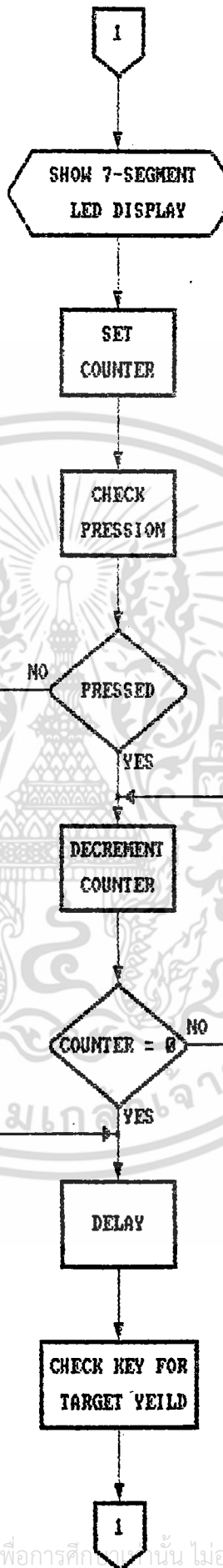


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

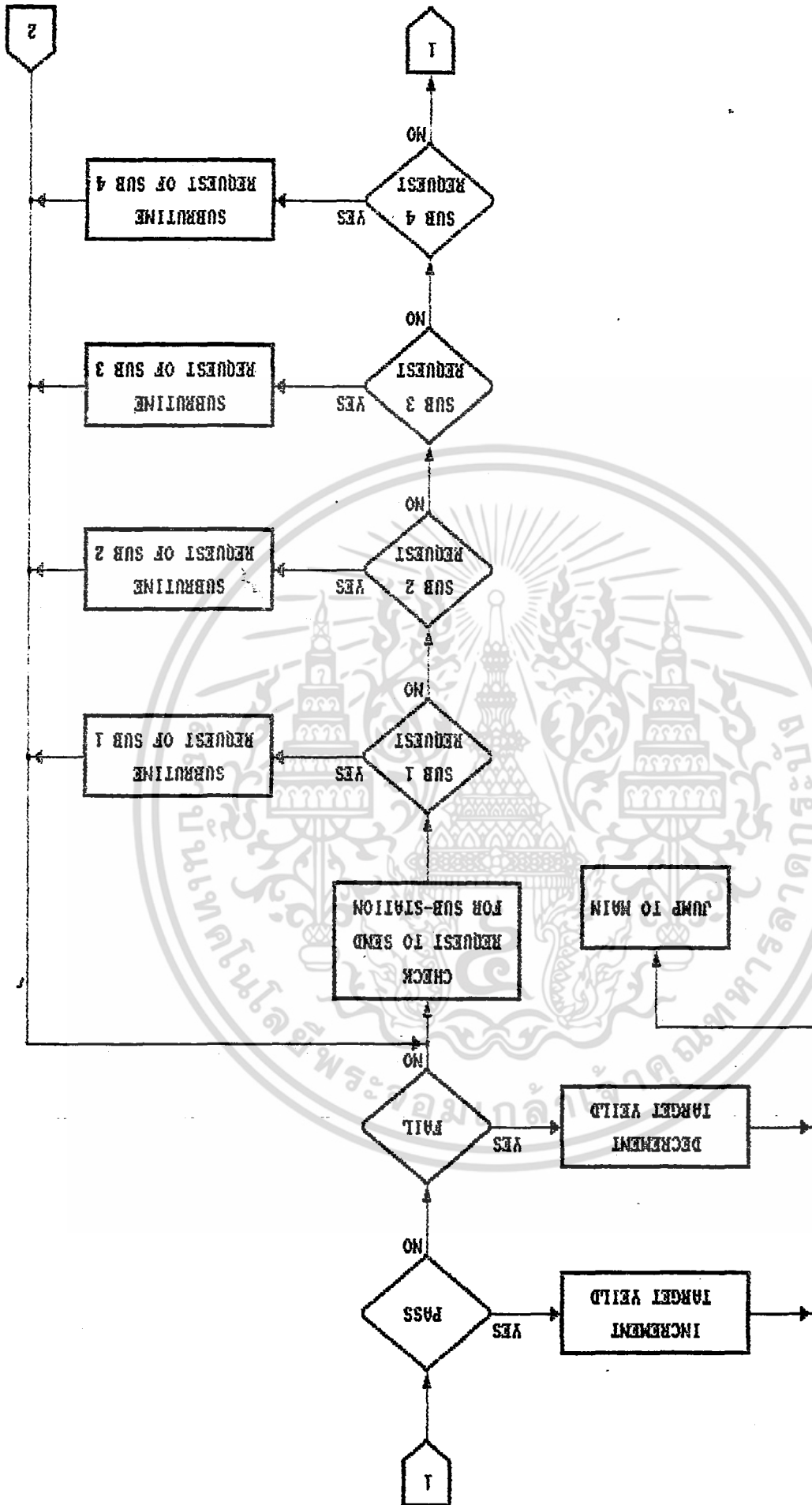
FLOWCHART OF MAIN-BOARD



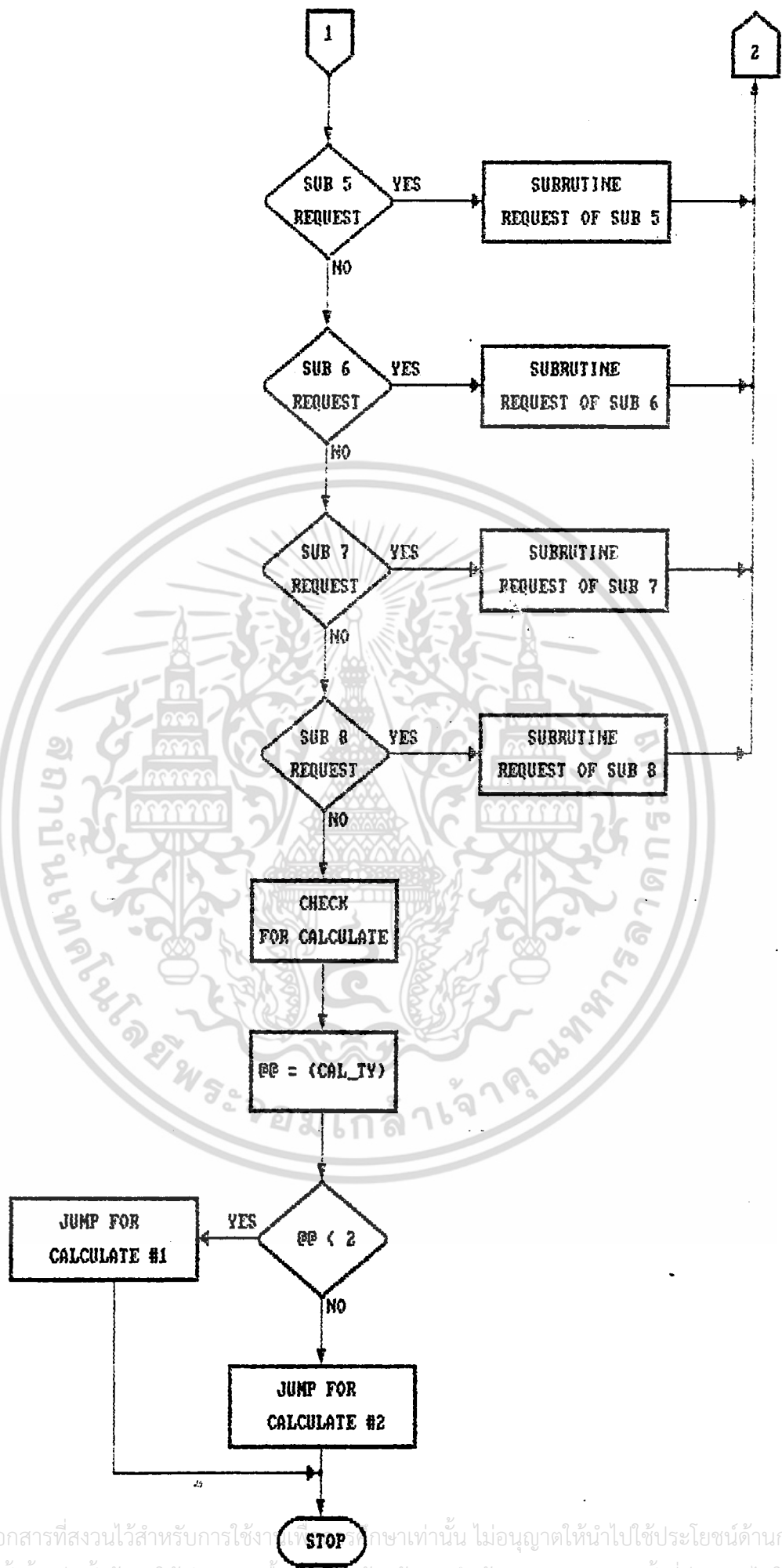
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

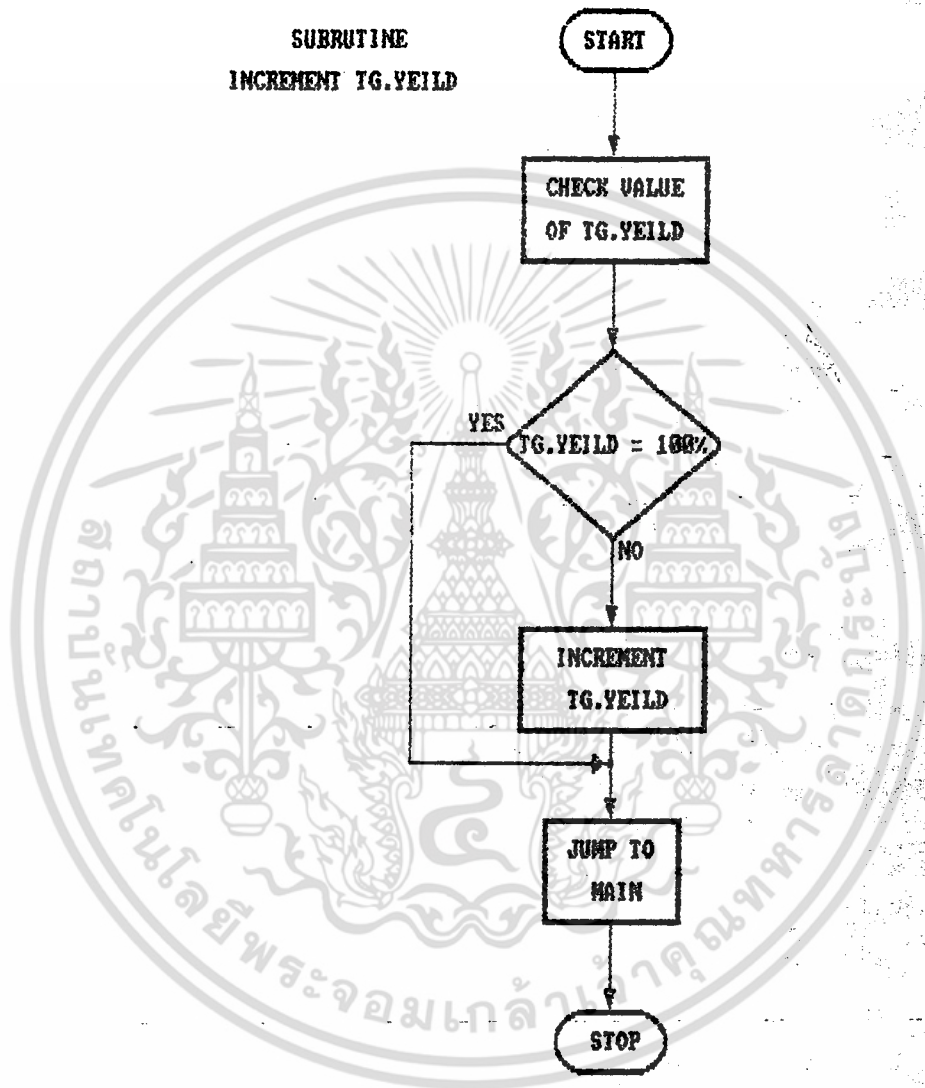


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



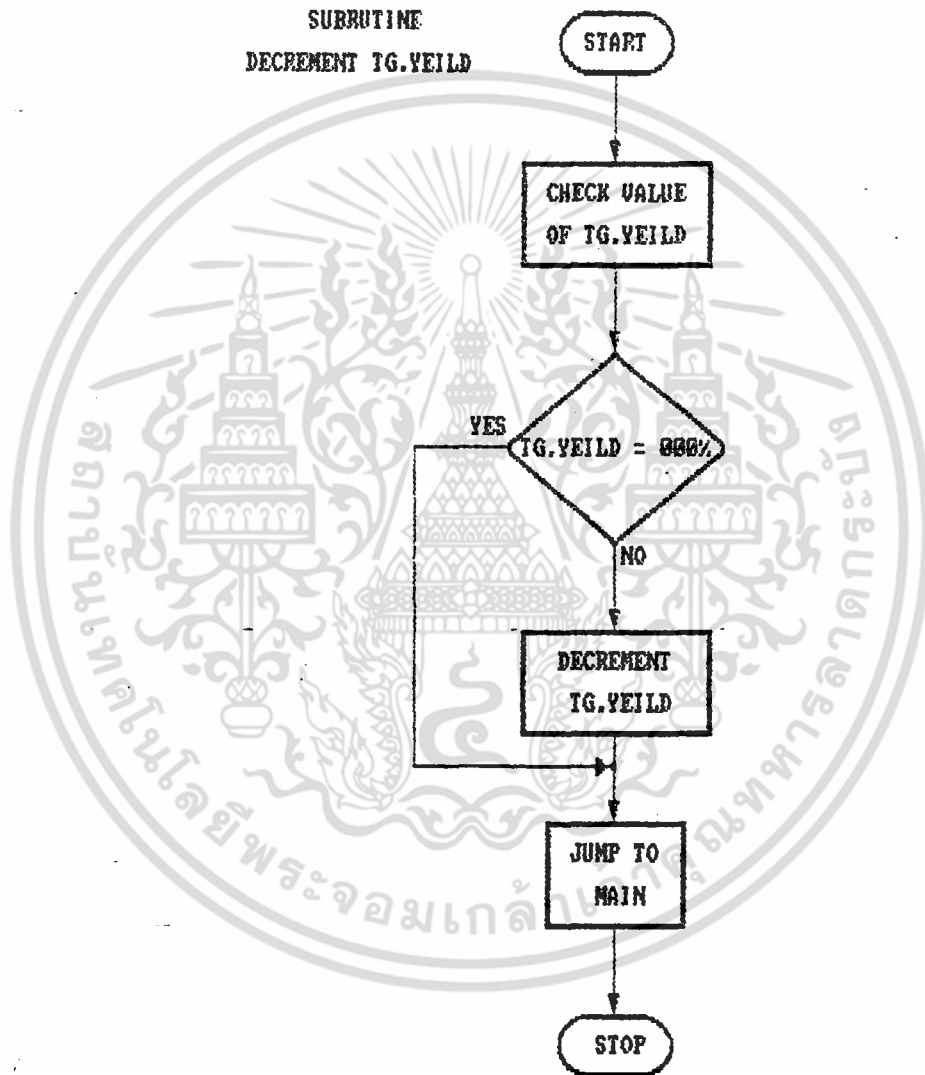
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
INCREMENT TG.YEILD



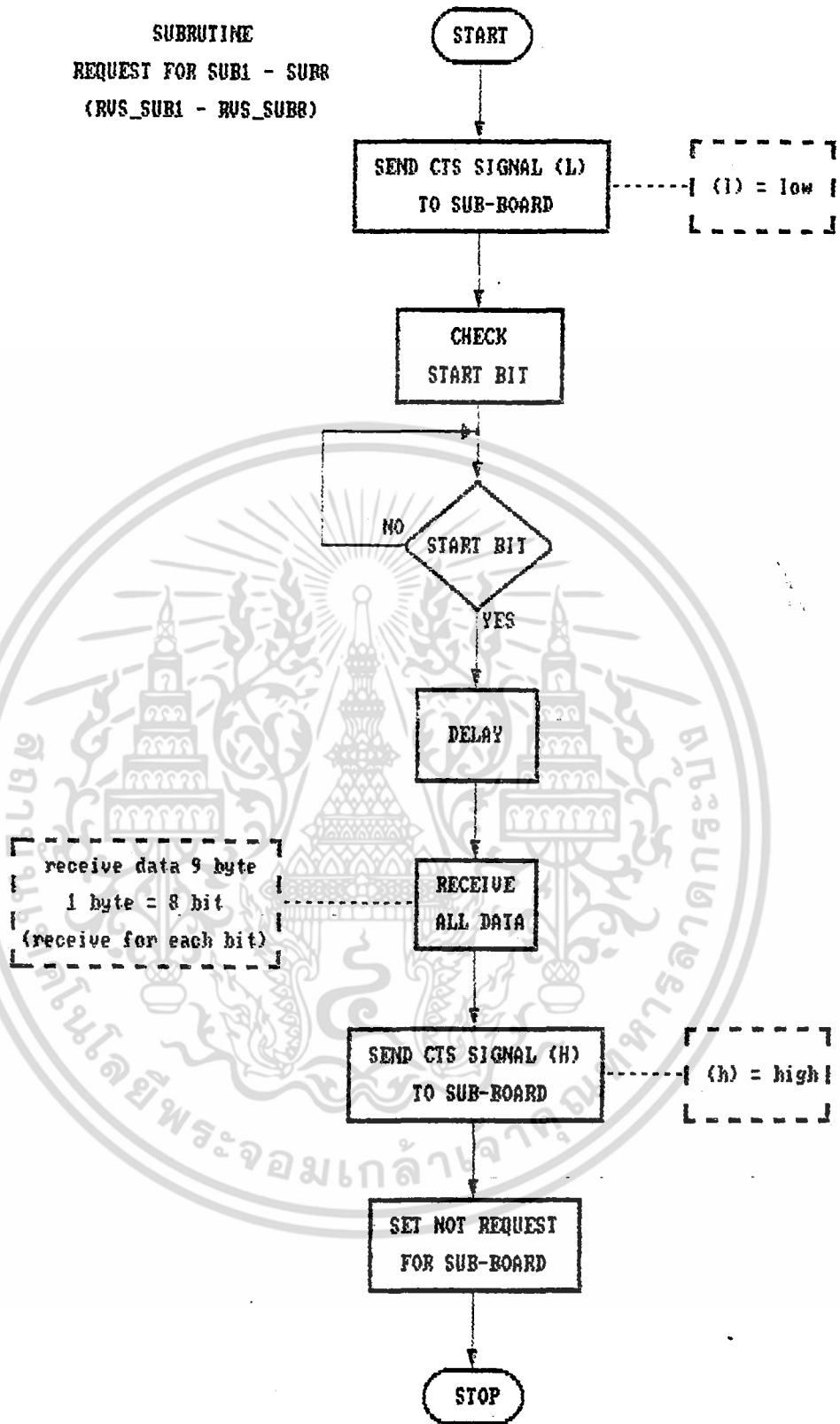
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
DECREMENT TG.YEILD



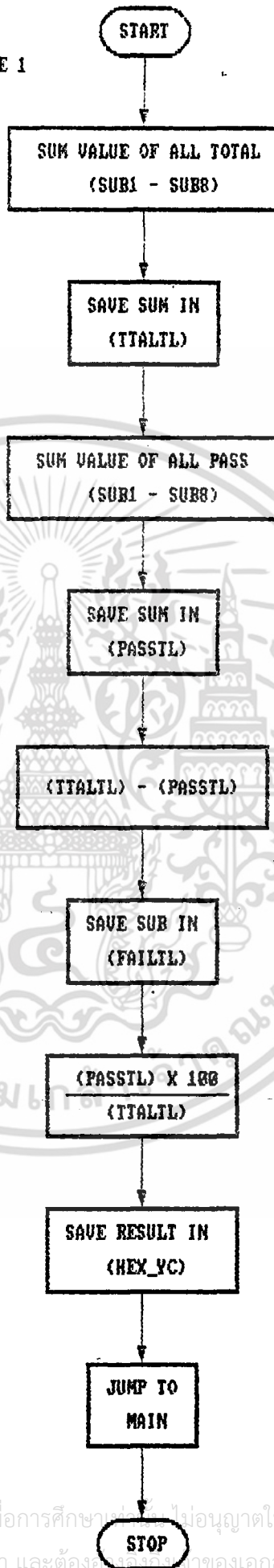
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
REQUEST FOR SUB1 - SUBR
(RVS_SUB1 - RVS_SUBR)



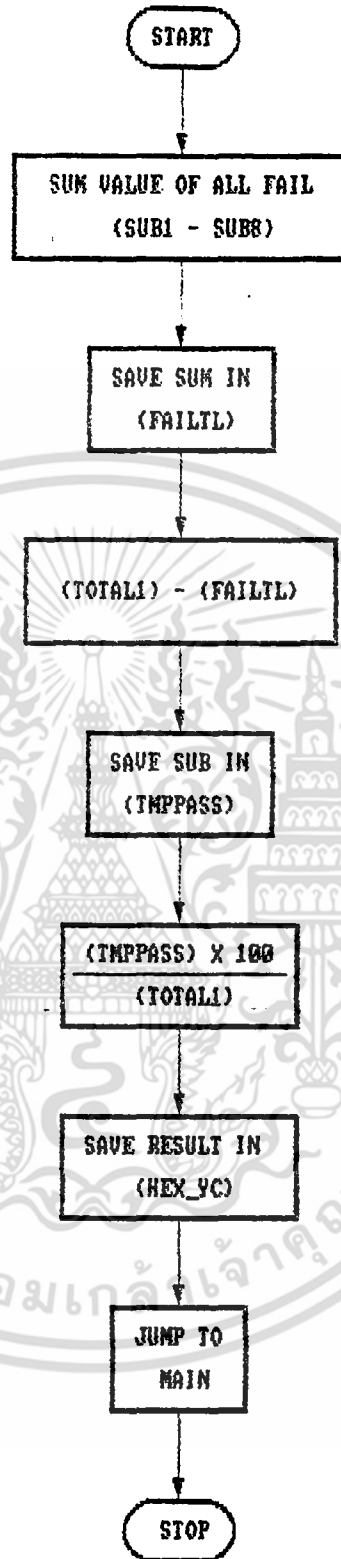
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
CALCULATE TYPE 1



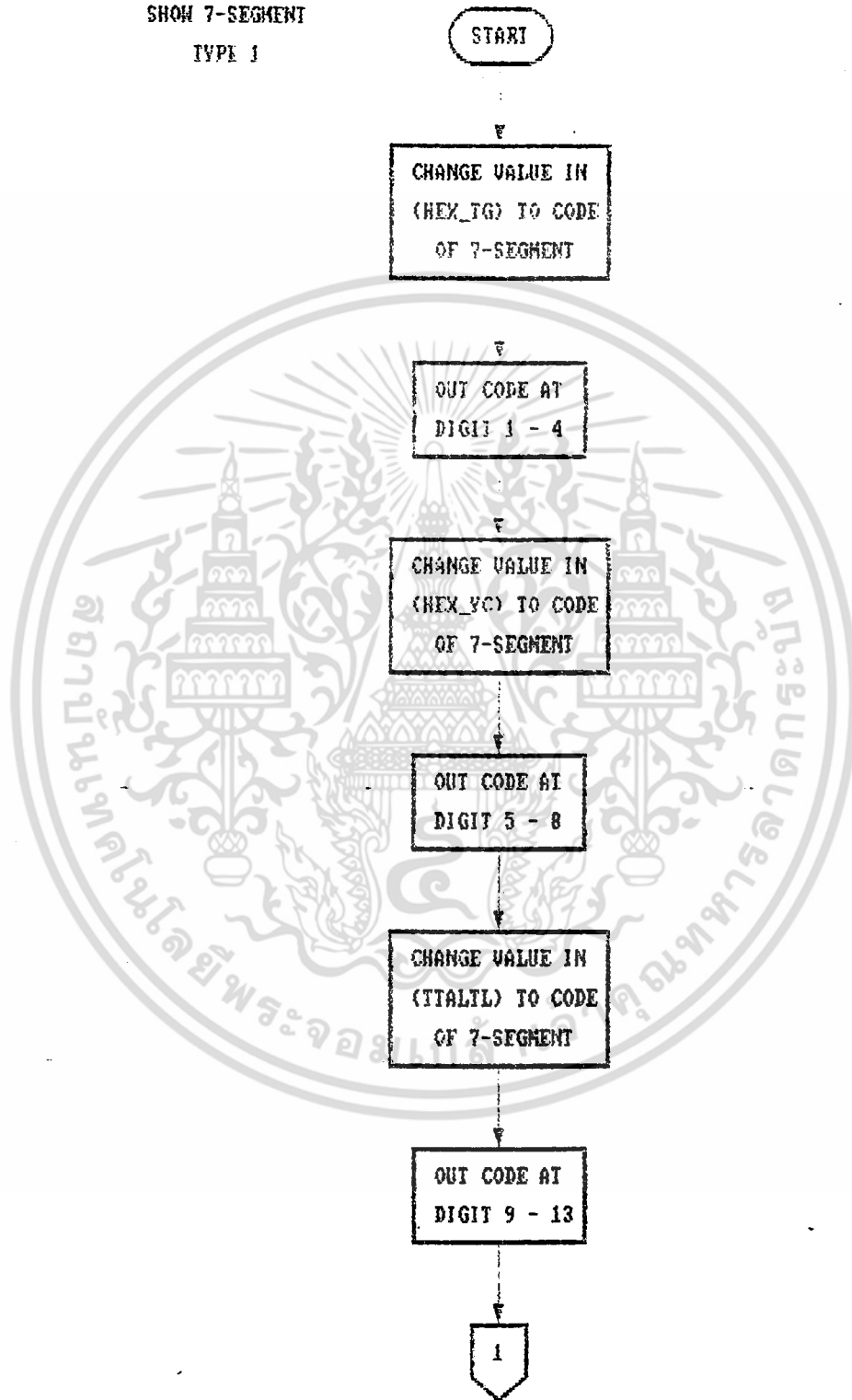
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องขอสงวนสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
CALCULATE TYPE 2

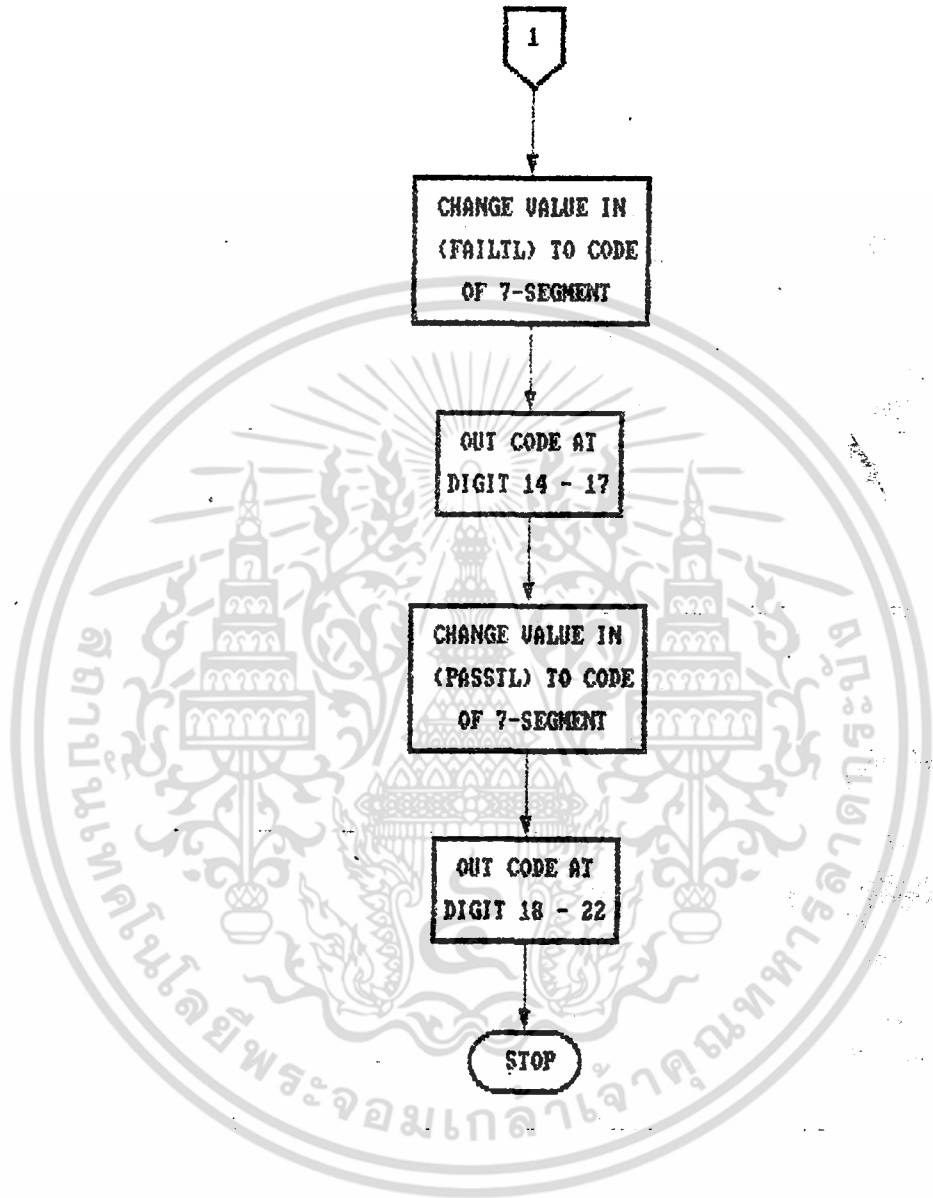


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
SHOW 7-SEGMENT
TYPE 1

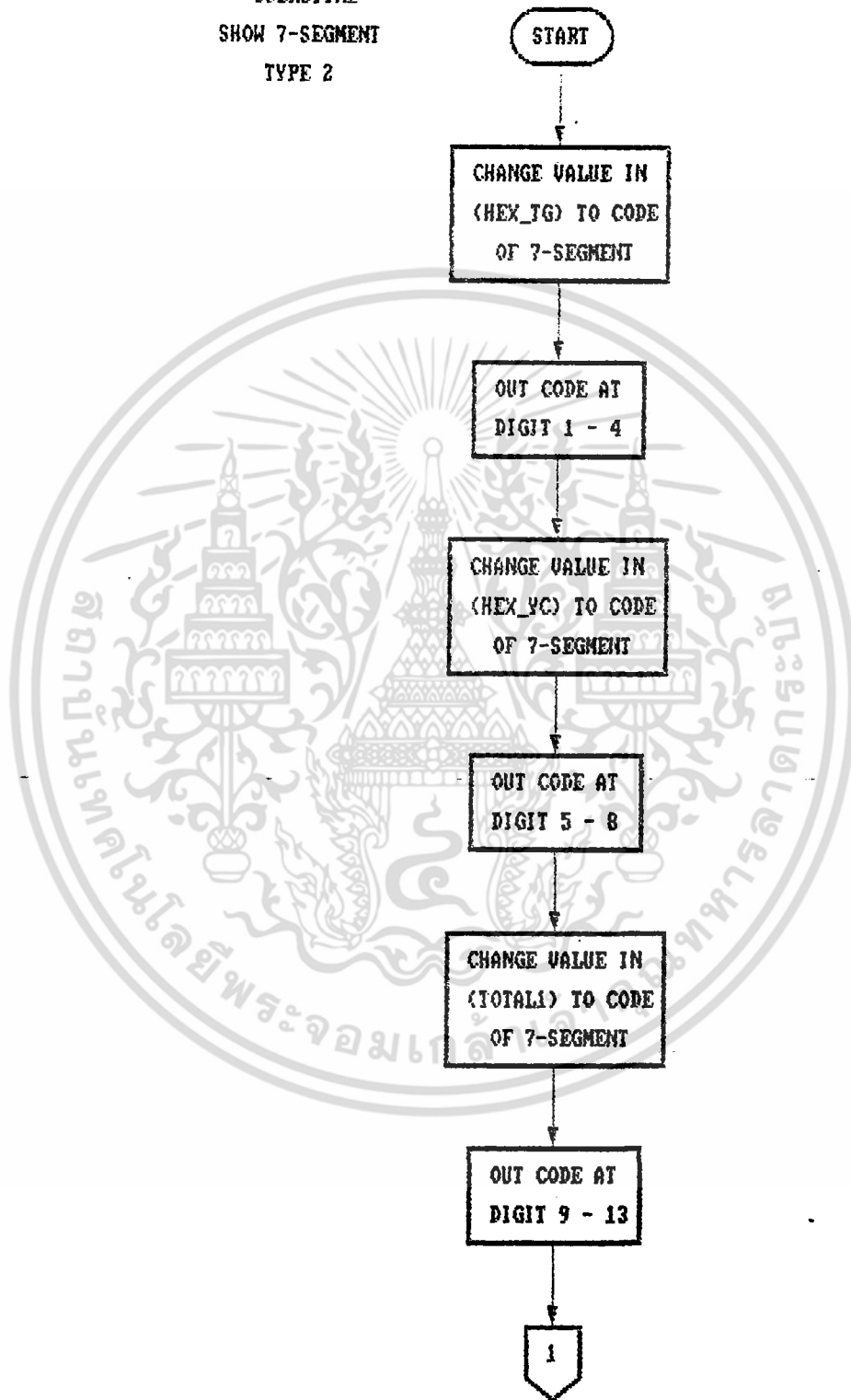


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

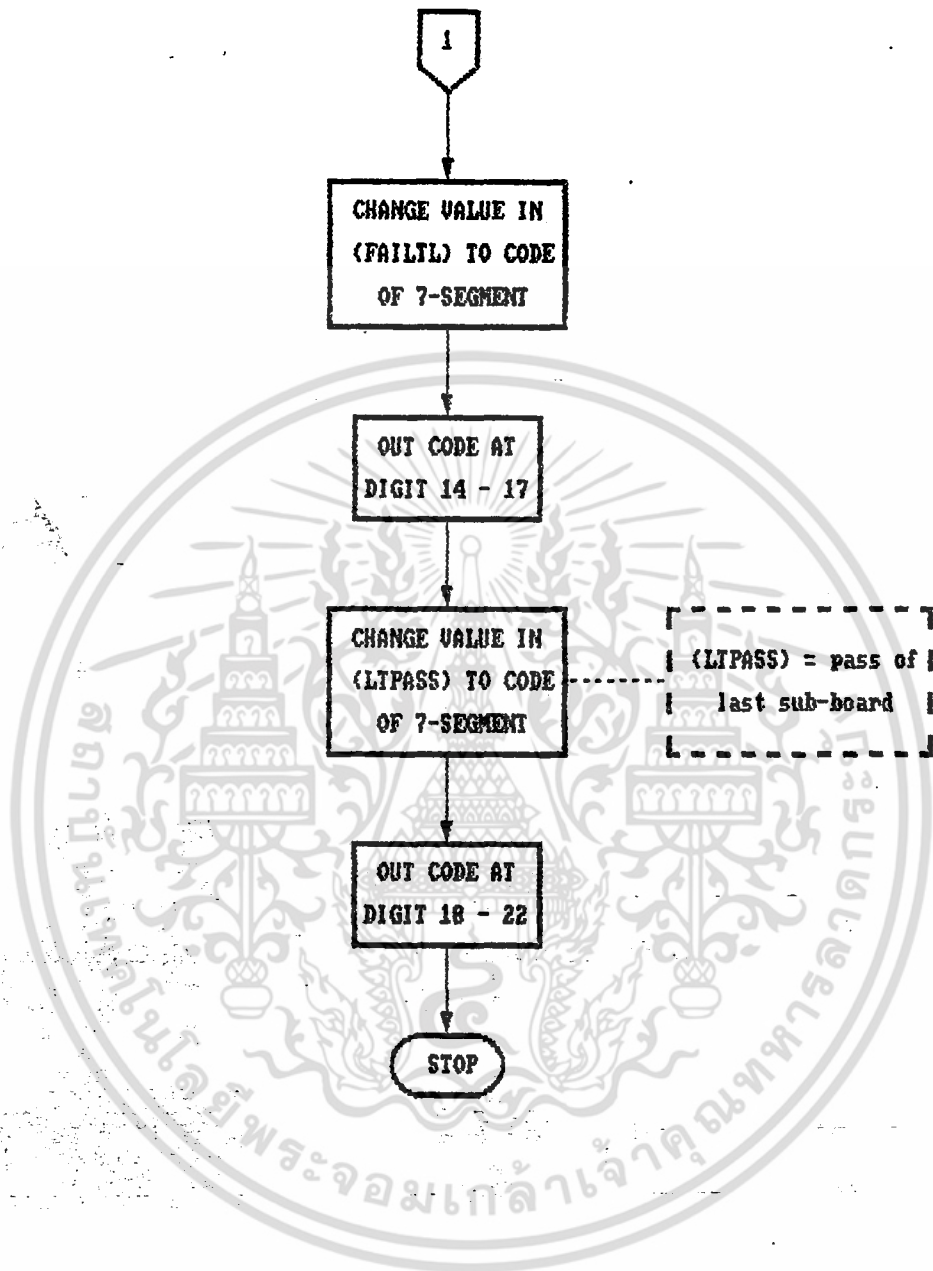


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUBROUTINE
SHOW 7-SEGMENT
TYPE 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่ใช้



ภาคผนวก ข.
โปรแกรมแอสซมบลี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีควรนำไปใช้

```

; *****
; *      MONITOR PROGRAM      *
; *  FOR SUB-BOARD OF YEILDO METER  *
; *  BY Z80180 SINGLE CHIP CONTROL  *
; *  CROSS - ASSEMBLER : A180Z.EXE  *
; *                                  *
; *      VERSION 1.0          *
; *  DATE WRITE : JANUARY 1992  *
; *  WRITEN BY KMIT'L SOFTWARE  *
; *  SCIENCE PHYSICS - SOLIDY (07) *
; *    : THANAPAT YONGPIPATWONG  *
; *    : BUNSONG CHARUNWORAPAN  *
; *****

```

.ORG 0000H

```

; *****
; *      REGISTER I/O Z80180    *
; *****

```

```

.EQU CNTLA0,00H ;ASCI CONTROL REG.A (0)
.EQU CNTLA1,01H ;ASCI CONTROL REG.A (1)
.EQU CNTLB0,02H ;ASCI CONTROL REG.B (0)
.EQU CNTLB1,03H ;ASCI CONTROL REG.B (1)
.EQU STAT0,04H  ;ASCI STATUS REG. (0)
.EQU STAT1,05H  ;ASCI STATUS REG. (1)
.EQU TDR0,06H   ;ASCI TRANSMIT DATA REG. (0)
.EQU TDR1,07H   ;ASCI TRANSMIT DATA REG. (1)
.EQU RDR0,08H   ;ASCI RECEIVE DATA REG. (0)
.EQU RDR1,09H   ;ASCI RECEIVE DATA REG. (1)
.EQU CNTR,0AH   ;CSI/O CONTROL REG.
.EQU TRDR,0BH   ;CSI/O TRANSMIT/RECEIVE DATA REG.
.EQU TMDROL,0CH ;TIME DATA REG. (0L)
.EQU TMDROH,0DH ;TIME DATA REG. (0H)
.EQU RLDROL,0EH ;RELODE REG. (0L)
.EQU RLDROH,0FH ;RELODE REG. (0H)
.EQU TCR,10H    ;TIMER CONTROL REG.
.EQU TMDR1L,14H ;TIMER DATA REG. (1L)
.EQU TMDR1H,15H ;TIMER DATA REG. (1H)
.EQU RLDR1L,16H ;RELODE REG. (1L)
.EQU RLDR1H,17H ;RELODE REG. (1H)
.EQU FRC,18H    ;FREE RUNNING COUNTER
.EQU SAROL,20H  ;DMA SOURCE ADD. REG. (0L)
.EQU SAROH,21H  ;DMA SOURCE ADD. REG. (0H)
.EQU SAROB,22H  ;DMA SOURCE ADD. REG. (0B)
.EQU DAROL,23H  ;DMA DESTINATION ADD. REG. (0L)
.EQU DAROH,24H  ;DMA DESTINATION ADD. REG. (0H)
.EQU DAROB,25H  ;DMA DESTINATION ADD. REG. (0B)
.EQU BCROL,26H  ;DMA BYTE COUNT REG. (0L)
.EQU BCROH,27H  ;DMA BYTE COUNT REG. (0H)
.EQU MAR1L,28H  ;MEMORY ADD. REG. (1L)
.EQU MAR1H,29H  ;MEMORY ADD. REG. (1H)
.EQU MAR1B,2AH  ;MEMORY ADD. REG. (1B)
.EQU IAR1L,2BH  ;DMA I/O ADD. REG. (1L)
.EQU IAR1H,2CH  ;DMA I/O ADD. REG. (1H)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.EQU BCR1L,2EH          ;DMA BYTE COUNT REG. (1L)
.EQU BCR1H,2FH          ;DMA BYTE COUNT REG. (1H)
.EQU DSTAT,30H          ;DMA STATUS REG.
.EQU DMODE,31H          ;DMA MODE REG.
.EQU DCNTL,32H          ;DMA/WAIT CONTROL REG.
.EQU IL,33H             ;IL REG. INTERRUPT VECTOR LOW REG.
.EQU ITC,34H            ;INT/TRAP CONTROL REG.
.EQU RCR,36H            ;REFRESH CONTROL REG.
.EQU CBR,38H            ;MMU COMMON BASE REG.
.EQU BBR,39H            ;MMU BANK BASE REG.
.EQU CBAR,3AH           ;MMU COMMON/BANK AREA REG.
.EQU OMCR,3EH           ;OPERATION MODE CONTROL REG.
.EQU ICR,3FH            ;I/O CONTROL REG.

; *****
; *          VARIABLE NUMBER          *
; *****

DATA_B: .CEQU 0C080H      ;PA 8255 (1)
DATA_D: .CEQU 0C081H      ;PB 8255 (1)
PSIGN: .CEQU 0C082H      ;PC 8255 (1)
PCONT: .CEQU 0C083H      ;P-CONTROL 8255 (1)
PLEDS: .CEQU 0C090H      ;PA 8255 (2)
PSWIT: .CEQU 0C091H      ;PB 8255 (2)
SWLED: .CEQU 0C092H      ;PC 8255 (2)
SWCON: .CEQU 0C093H      ;P-CONTROL 8255 (2)
CODE_ST: .CEQU 0A3H       ;CODE START OR RESET
INTERRUP: .CEQU 1A00H     ;INTERRUPT ROUTINE
START_BF: .CEQU 80AEH     ;START BUFFER OF STATUS
END_BF: .CEQU 81ACH       ;END BUFFER OF STATUS
SYSSTK: .CEQU 9F00H       ;START SYSTEM STACK

; *****
; *          POWER-UP          *
; *****

POWER: XOR A              ;POWER-UP DELAY
POWER1: DEC A
        NOP
        JP NZ,POWER1
        JP RESET

; *****
; *          ROUTINE OF NMI          *
; *          FOR CLEAR ALL DATA IN RAM          *
; *****

.ORG 0066H
LD HL,STCODE
LD (HL),00H
JP 0000H

; *****
; *          RESET SYSTEM          *
; *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RESET:   .ORG 100H
         LD   HL,4000H           ;DELAY
RESETO:  DEC  HL
         LD   A,H
         OR   L
         JP  NZ,RESETO

```

```

; *****
; *           SET INITIALIZE           *
; *****

```

```

LD   A,5FH
OUTO (OMCR),A
LD   A,80H
OUTO (CBAR),A
LD   A,38H
OUTO (ITC),A
LD   A,00H
OUTO (TMDR1H),A
OUTO (RLDR1H),A
LD   A,4BH
OUTO (TMDR1L),A
OUTO (RLDR1L),A
LD   HL,INTERRUP
LD   (DET_INT),HL
LD   BC,DET_INT
LD   A,B
LD   I,A
LD   A,C
OUTO (IL),A
LD   A,35H
OUTO (CNTLAO),A
LD   A,04H
OUTO (CNTLBO),A

```

```

; *****
; *           START SYSTEM             *
; *****

```

```

START:  LD   SP,SYSSTK           ;SYSTEM STACK
         LD   A,80H             ;SET CONTROL PORT 8255 (1)
         LD   BC,PCONT
         OUT  (C),A
         LD   A,83H             ;SET CONTROL PORT 8255 (2)
         LD   BC,SWCON
         OUT  (C),A

```

```

; *****
; *           SET SYSTEM OF LCD       *
; *           AND CLEAR LCD           *
; *****

```

```

LD   A,0AAH           ;CODE OF LCD-BOARD
LD   (CODE_LCD),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL INITLCD ;SET LCD-BOARD
LD A,0BBH ;CODE OF LCD-DISPLAY
LD (CODE_LCD),A
CALL INITLCD ;SET LCD-DISPLAY

; *****
; * SET IMPORTANCE CODE VALUE *
; *****

LD A,00H
LD (CODE_INT),A ;FOR INTERRUPT
LD (CK_SHOW),A ;FOR SHOW LCD
LD (CODE_LCD),A ;FOR CHECK LCD

; *****
; * CHECK START SYSTEM *
; *****

LD A,(STCODE)
CP CODE_ST
JP Z,SETIO

; *****
; * CLEAR ALL DATA IN RAM *
; *****

CALL SOUND

; *****
; * BACKUP RAM OR RESET *
; *****

CLR_SET: LD HL,OFFFH
LD (ADRS),HL
LD A,00H
LD (CK_ITV),A ;FOR CHECK INTERVAL
LD HL,START_BF
LD (AD_BUF),HL ;INSTORE ADDRESS IN BUFFER

; *****
; * SET ALL VALUE (BCD) *
; *****

LD HL,INTERVAL ;FIRST VALUE BCD
LD B,52 ;COUNTER
SET0: LD (HL),00H ;FOR ANY VALUE
INC HL
DEC B
JP NZ,SET0
LD (HL),10H ;FOR YEILD ALERT
LD HL,CLSTK ;FOR CALCULATE
LD B,30 ;COUNTER
SET1: LD (HL),00H
INC HL
DEC B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JP    NZ,SET1

; *****
; *      SET ALL VALUE (ASCII)      *
; *****

LD    IX,ASC_ITV      ;FIRST VALUE ASCII
LD    B,63            ;COUNTER
SET2: LD    (IX+0),30H ;FOR ANY VALUE
      INC  IX
      DEC  B
      JP  NZ,SET2
      LD  (IX+0),31H   ;FOR YEILD ALERT
      LD  (IX+1),30H
      LD  (IX+2),30H
      LD  (IX+3),30H

; *****
; *      SET FIRST TARGET YEILD      *
; *      5.0 %                        *
; *****

LD    HL,HEX_TG
LD    (HL),00H
INC  HL
LD    (HL),85H
LD    IX,ASC_TG
LD    (IX+0),38H
LD    (IX+1),35H
LD    (IX+2),30H
JP    MAIN0

; *****
; *      SET TARGET YEILD            *
; *****

MAIN0: CALL SHOLCDOD
MAIN1: CALL KEYPRS1
      CALL DELAY
      CALL SHOLCD1B
LD    IX,ASC_TG
LD    IY,HEX_TG
SKEY1: XOR  A          ;CLEAR REG.A
      LD  BC,SWLED
      IN  A,(C)        ;CHECK KEY FOR TG.YEILD
      AND OFH
      BIT 0,A
      JP  Z,TG_INC    ;INCREMENT TG.YEILD
      BIT 1,A
      JP  Z,TG_DEC    ;DECREMENT TG.YEILD
      BIT 3,A
      JP  NZ,SKEY1    ;NOT SURE

; *****
; *      SET INTERVAL VALUE          *
; *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAIN2:   CALL KEYPRS
         CALL DELAY
         CALL SHOLCD2B
         LD   IX,ASC_ITV
         LD   IY,INTERVAL
SKEY2:   XOR   A                               ;CLEAR REG.A
         LD   BC,SWLED
         IN   A,(C)                             ;CHECK KEY FOR INTERVAL
         AND  0FH
         BIT  0,A
         JP   Z,TEN                             ;SET TEN OF,INTERVAL
         BIT  1,A
         JP   Z,HNRED                           ;SET HUNDRED OF INTERVAL
         BIT  3,A
         JP   NZ,SKEY2
         LD   A,(IY+1)                          ;CHECK VALUE INTERVAL
         CP   00H
         JP   NZ,SETI1                          ;HUNDRED # 00
         LD   A,(IY+0)
         CP   00H
         JP   Z,SKEY2                          ;TEN = 00
         JP   SETI1

; *****
; *   CHECK FOR CONTINUATION   *
; *****

SETI0:   LD   HL,(ADRS)
         XOR  A
         LD   A,H
         CP   11H
         JP   NZ,SETI1                          ;NO DELETION
         LD   HL,SAVE_TTL
         LD   DE,HEXTOTAL
         LD   BC,03H
         LDIR
         LD   IX,HEXTOTAL
         LD   HL,ASC_TOTAL
         LD   A,06H
         LD   (NUMBER),A
         LD   D,03H
         LD   B,05H
         CALL CH_AC16
         LD   IX,SAVE_BUF
         LD   C,(IX+0)
         LD   B,(IX+1)
         INC  BC
         LD   (AD_BUF),BC

SETI1:   LD   HL,OFFFH
         LD   (ADRS),HL                          ;FOR CHECK SWITCH

SET_CODE: LD  A,CODE_ST
          LD  (STCODE),A                        ;SET START CODE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ,ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *   SET TYPE OF INTERRUPT   *
; *****

                LD  A,3AH
                OUTO (ITC),A                ;ENABLE INT1 AND
                                           ;DISABLE INTO,INT2

; *****
; *   MAIN PROGRAM           *
; *****

MAIN3:          EI                        ;ENABLE INTERRUPT
                CALL SHOLCD3B
                LD   HL,CODE_INT          ;CHECK CODE INTERRUPT
                LD   A,(HL)
                CP   \ 00H
                JP   Z,SHO3                ;(CODE_INT) = 00
                CALL SHOLCD1D
                JP   KEY                    ;(CODE_INT) = 01
SHO3:           CALL SHOLCD2D
KEY:            CALL KEYPRS
                CALL DELAY
                JP   SKEY3

; ***** SUBROUTINE CHECK KEY *****

SKEY3:          LD   BC,PSWIT
                XOR  A
                IN  A,(C)
                BIT  0,A
                JP  Z,FAIL_A1              ;CHOOSE A
                BIT  1,A
                JP  Z,FAIL_B1              ;CHOOSE B
                BIT  2,A
                JP  Z,FAIL_C1              ;CHOOSE C
                BIT  3,A
                JP  Z,FAIL_D1              ;CHOOSE D
                BIT  4,A
                JP  Z,FAIL_E1              ;CHOOSE E
                BIT  5,A
                JP  Z,FAIL_F1              ;CHOOSE F
                BIT  6,A
                JP  Z,FAIL_G1              ;CHOOSE G
                BIT  7,A
                JP  Z,FAIL_H1              ;CHOOSE H
                LD   BC,SWLED
                XOR  A
                IN  A,(C)
                BIT  0,A
                JP  Z,PASS1                ;CHOOSE PASS
                BIT  1,A
                JP  Z,FAIL1                ;CHOOSE FAIL
                BIT  2,A
                JP  Z,DELETE              ;CHOOSE DELETE
                BIT  3,A
                JP  Z,ENTER                ;CHOOSE ENTER

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JP    SKEY3                                ;NO CHOOSE

; *****
; *   SUBROUTINE TEN OF INTERVAL   *
; *****
;
; REG = AIXIY

TEN:           LD    A,(IY+1)                            ;CHECK HUNDRED OF INTERVAL
               CP    02H
               JP    NZ,TEN1
               LD    A,(IY+0)                            ;CHECK TEN OF INTERVAL
               CP    50H
               JP    NZ,TEN1
               LD    A,00H                               ;DECREMENT TEN INTERVAL
               JP    TEN3
TEN1:          LD    A,(IY+0)                            ;CHECK TEN OF INTERVAL
               CP    90H
               JP    NZ,TEN2
               LD    A,00H                               ;DECREMENT TEN OF INTERVAL
               JP    TEN3
TEN2:          ADD   A,10H                               ;INCREMENT TEN OF INTERVAL
TEN3:          LD    (IY+0),A                            ;INSTORE TEN VALUE
               AND   0F0H
               SRL   A
               SRL   A
               SRL   A
               SRL   A
               ADD   A,30H                               ;CHANGE TO ASCII
               LD    (IX+1),A                            ;INSTORE ASCII OF TEN
               JP    MAIN2

; *****
; *   SUBROUTINE
; *   HUNDRED OF INTERVAL         *
; *****
;
; REG = AIXIY

HNRED:         XOR   A                                  ;CLEAR CARRY FLAG
               LD    A,(IY+0)                            ;CHECK TEN OF INTERVAL
               CP    60H
               JP    C,HNRED1
               LD    A,(IY+1)                            ;CHECK HUNDRED OF INTERVAL
               CP    01H
               JP    NZ,HNRED3
               JP    HNRED2
HNRED1:        LD    A,(IY+1)                            ;CHECK HUNDRED OF INTERVAL
               CP    02H
               JP    NZ,HNRED3
HNRED2:        LD    A,00H                               ;DECREMENT HUNDRED OF INTERVAL
               JP    HNRED4
HNRED3:        LD    A,(IY+1)
               ADD   A,01H                               ;INCREMENT HUNDRED OF INTERVAL
HNRED4:        LD    (IY+1),A                            ;INSTORE HUNDRED VALUE
               ADD   A,30H                               ;CHANGE TO ASCII

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (IX+0),A ;INSTORE ASCII OF HUNDRED
JP MAIN2

; *****
; * SUBROUTINE INCREMENT *
; * TARGET YEILD *
; *****
;
; REG = AIXIY

TG_INC: LD A,(IY+0) ;CHECK POINT OF TG.YEILD
AND OF0H
CP 90H
JP Z,TG_IN1 ;POINT = 90 (x.90)
ADD A,10H ;INCREMENT POINT OF TG.YEILD
LD (IY+0),A ;INSTORE POINT VALUE
SRL A
SRL A
SRL A
SRL A
ADD A,30H ;CHANGE TO ASCII
LD (IX+2),A ;INSTORE ASCII POINT
JP MAIN1

TG_IN1: LD A,(IY+1) ;CHECK VALUE OF TG.YEILD
CP 99H
JP Z,MAIN1 ;TG.YEILD = 99 (99.x)
LD A,00H ;DECREMENT POINT OF TG.YEILD
LD (IY+0),A ;INSTORE POINT VALUE
ADD A,30H ;CHANGE TO ASCII
LD (IX+2),A ;INSTORE ASCII POINT
LD A,(IY+1)
ADD A,01H ;INCREMENT TG.YEILD
DAA
LD (IY+1),A ;INSTORE TG.YEILD
AND OF0H
ADD A,30H ;CHANGE TO ASCII
LD (IX+1),A ;INSTORE ASCII TG.YEILD (UNIT)
LD A,(IY+1)
AND OF0H
SRL A
SRL A
SRL A
SRL A
ADD A,30H ;CHANGE TO ASCII
LD (IX+0),A ;INSTORE ASCII TG.YEILD (TEN)
JP MAIN1

```

```

; *****
; * SUBROUTINE DECREMENT *
; * TARGET YEILD *
; *****
;
; REG = AIXIY

```

```

TG_DEC: LD A,(IY+0) ;CHECK POINT OF TG.YEILD
AND OF0H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP      00H
JP      Z,TG_DE1      ;POINT = 00 (x.00)
SUB     10H          ;DECREMENT POINT OF TG.YEILD
LD      (IY+0),A     ;INSTORE POINT VALUE
SRL     A
SRL     A
SRL     A
SRL     A
ADD     A,30H        ;CHANGE TO ASCII
LD      (IX+2),A     ;INSTORE ASCII POINT
TG_DE1: JP      MAIN1
LD      A,(IY+1)     ;CHECK VALUE OF TG.YEILD
CP      00H
JP      Z,MAIN1      ;TG.YEILD = 00 (00.x)
LD      A,90H        ;INCREMENT POINT OF TG.YEILD
LD      (IY+0),A     ;INSTORE POINT VALUE
LD      A,09H
ADD     A,30H        ;CHANGE TO ASCII
LD      (IX+2),A     ;INSTORE ASCII POINT
LD      A,(IY+1)
SUB     01H          ;DECREMENT TG.YEILD
DAA
LD      (IY+1),A     ;INSTORE TG.YEILD
AND     0FH
ADD     A,30H        ;CHANGE TO ASCII
LD      (IX+1),A     ;INSTORE ASCII TG.YEILD (UNIT)
LD      A,(IY+1)
AND     0FOH
SRL     A
SRL     A
SRL     A
SRL     A
ADD     A,30H        ;CHANGE TO ASCII
LD      (IX+0),A     ;INSTORE ASCII TG.YEILD (TEN)
JP      MAIN1

```

```

; *****
; *          SUBROUTINE          *
; *   INITIAL LCD DISPLAY      *
; *****
;
; PA0-PA7 : PIN D0-D7 (DATA READ/WRITE LCD OF BOARD)
; PC2     : PIN E     (ENABLE SIGNAL PULSE OF BOARD)
; PC1     : PIN R/W   (READ/WRITE OF BOARD)
; PC0     : PIN RS    (REGITER SELECTION OF BOARD)
; PB0-PB7 : PIN D0-D7 (DATA READ/WRITE LCD OF DISPLAY)
; PC6     : PIN E     (ENABLE SIGNAL PULSE OF DISPLAY)
; PC5     : PIN R/W   (READ/WRITE OF DISPLAY)
; PC4     : PIN RS    (REGITER SELECTION OF DISPLAY)
; REG = ABC

```

```

INITLCD: LD  A,00H
          LD  BC,PSIGN
          OUT (C),A
          LD  A,(CODE_LCD)      ;CHECK CODE LCD
          CP  0AAH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JP  NZ,LCD_2I                ;(CODE_LCD) # 0AA
                LD  BC,DATA_B
                JP  INITL
LCD_2I:         LD  BC,DATA_D
INITL:         LD  A,38H                    ;FUNCTION SET 38H
                                                ;DL=1 8 BIT,N=1 1/16 DUTY,F=0 5x7

                OUT (C),A
                CALL EPLUSE                  ;ENABLE SIGNAL PLUSE
                CALL DELAY                   ;DELAY >4.31 MS
                LD  A,0FH                    ;DISPLAY ON/OFF CONTROL
                                                ;D=1 OFF,C=1 CURSOR ON,B=1 BLINK

                OUT (C),A
                CALL EPLUSE
                LD  A,06H                    ;ENTRY MODE SET
                                                ;I/D=1 INCREMENT,S=0 RIGHT

                OUT (C),A
                CALL EPLUSE
                LD  A,01H                    ;CLEAR ALL DISPLAY
                OUT (C),A
                CALL EPLUSE
                RET

; *****
; *          SUBROUTINE          *
; *    WRITE LINE 16 CHAR      *
; *****
;
; REG = BCDHL
WRLM:         LD  B,16                      ;16 CHAR
WRL:          LD  D,(HL)
                PUSH BC
                CALL WRBYTE
                POP BC
                INC HL
                DEC B
                JP  NZ,WRL
                RET

; *****
; *          SUBROUTINE          *
; *    ENABLE PLUSE SUB        *
; *****
;
; REG = ABCD

EPLUSE:       PUSH BC
                LD  BC,PSIGN
                IN  A,(C)
                LD  D,A
                LD  A,(CODE_LCD)
                CP  0AAH
                JP  NZ,LCD_2E
                LD  A,D
                SET 2,A                      ;ENABLE BIT 2 = 1
                OUT (C),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EP1:      LD   B,00H
          DEC  B
          JP   NZ,EP1
          RES  2,A
          LD   BC,PSIGN
          OUT  (C),A           ;ENABLE BIT 2 = 0
          POP  BC
          RET

LCD_2E:   LD   A,D
          SET  6,A           ;ENABLE BIT 6 = 1
          OUT  (C),A
          LD   B,00H

EP2:      DEC  B
          JP   NZ,EP2
          RES  6,A
          LD   BC,PSIGN
          OUT  (C),A           ;ENABLE BIT 6 = 0
          POP  BC
          RET

; *****
; *   SUBROUTINE GOTO POSITION   *
; *****
;
; REG = ABC

GOTO:     PUSH BC
          PUSH AF
          LD   A,(CODE_LCD)
          CP   0AAH
          JP   NZ,LCD_2G
          LD   BC,DATA_B
          JP   GOTO1

LCD_2G:   LD   BC,DATA_D

GOTO1:    POP  AF
          SET  7,A           ;SET DD RAM 1
          OUT  (C),A
          XOR  A
          LD   BC,PSIGN
          OUT  (C),A           ;SET RS=0,R/W=0
          CALL EPLUSE
          POP  BC
          RET

; *****
; *   SUBROUTINE WRITE DATA SUB   *
; *****
;
; REG = ABCD

WRBYTE:   LD   A,(CODE_LCD)
          CP   0AAH
          JP   NZ,LCD_2W
          LD   A,01H           ;DATA WRITE
          LD   BC,PSIGN
          OUT  (C),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                LD  BC,DATA_B
                JP  WRB
LCD_2W:        LD  A,10H
                LD  BC,PSIGN
                OUT (C),A
                LD  BC,DATA_D
WRB:           LD  A,D                ;DATA BYTE
                OUT (C),A
                CALL EPLUSE
                RET

```

```

; *****
; *          SUBROUTINE DELAY          *
; *****
;
; REG = B

```

```

DELAY:        PUSH BC
                LD  B,00H
DE1:          NOP
                NOP
                DEC B
                JP  NZ,DE1
                POP BC
                RET

```

```

; *****
; *          SUBROUTINE                *
; *          CHECK KEY PRESSED 1      *
; *****
;
; REG = ABCDE

```

```

KEYPRS1:      CALL DELAY10
                LD  E,OFFH                ;DELAY IN SUBROUTINE
PRS1:         LD  BC,SWLED
                LD  D,04H                ;COUNT BIT
                XOR  A                    ;CLEAR REG.A
                IN  A,(C)
SH_BIT1:      BIT  0,A                    ;CHECK PRESSION
                JP  Z,PRESS1              ;PRESSED KEY
                RR  A
                DEC D
                JP  NZ,SH_BIT1
                JP  NO_PRS1                ;NOT PRESSED KEY
PRESS1:       DEC  E
                JP  NZ,PRS1
NO_PRS1:      RET

```

```

; *****
; *          SUBROUTINE                *
; *          CHECK KEY PRESSED 2      *
; *****
;
; REG = ABCDE

```

```

KEYPRS:       CALL DELAY10

```

```

                LD    BC,PSWIT
                LD    D,08H                ;COUNT BIT
                LD    E,02H                ;COUNT PORT
CH_PORT:      XOR    A                    ;CLEAR REG.A
                IN    A,(C)
SH_BIT:      BIT    0,A                    ;CHECK PRESSION
                JP    Z,KEYPRS            ;PRESSED KEY
                RR    A
                DEC  D
                JP    NZ,SH_BIT
                DEC  E
                JP    Z,NO_PRS            ;NOT PRESSED KEY
                LD    BC,SWLED
                LD    D,04H                ;COUNT BIT
                JP    CH_PORT
NO_PRS:      RET

```

```

; *****
; *      SUBROUTINE DELAY 10 ms      *
; *****
;
;
; REG = ABC

```

```

DELAY10:     PUSH AF
                PUSH BC
                LD    BC,05FOOH
DE10:      DEC  BC
                LD    A,B
                OR   C
                JP   NZ,DE10
                POP  BC
                POP  AF
                RET

```

```

; *****
; *      SUBROUTINE                  *
; *      SHOW LCD 1B (BOARD)        *
; *****
;
;
; REG = ABHL

```

```

SHOLCD1B:   LD    A,0AAH
                LD    (CODE_LCD),A        ;SET CODE LCD FOR BOARD
                LD    HL,LCD1B
                CALL ALL_LCD
                LD    HL,ASC_TG
                LD    A,59H
                CALL GOTO
                LD    B,2                    ;2 CHAR
                CALL WRL
                LD    HL,ASC_DOT
                LD    A,5CH
                CALL GOTO
                LD    B,1                    ;1 CHAR
                CALL WRL
                LD    HL,PERCENT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,5EH
CALL GOTO
LD B,2 ;2 CHAR
CALL WRL
RET

; *****
; * SUBROUTINE *
; * SHOW LCD 2B (BOARD) *
; *****
;
; REG = ABHL

SHOLCD2B: LD A,0AAH
LD (CODE_LCD),A ;SET CODE LCD FOR BOARD
LD HL,LCD2B
CALL ALL_LCD
LD HL,ASC_ITV
LD A,17H
CALL GOTO
LD B,3 ;3 CHAR
CALL WRL
LD HL,EMPTY
LD A,1AH
CALL GOTO
LD B,6 ;6 SPACE
CALL WRL
RET

; *****
; * SUBROUTINE *
; * SHOW LCD 3B (BOARD) *
; *****
;
; REG = ABHL

SHOLCD3B: LD A,0AAH
LD (CODE_LCD),A ;SET CODE LCD FOR BOARD
LD HL,LCD3B
CALL ALL_LCD
LD HL,ASC_TG
LD A,0BH
CALL GOTO
LD B,2 ;2 CHAR
CALL WRL
LD HL,ASC_DOT
LD A,0EH
CALL GOTO
LD B,1 ;1 CHAR
CALL WRL
LD HL,ASC_YA
LD A,4AH
CALL GOTO
LD B,3 ;3 CHAR
CALL WRL
LD HL,ASC_DA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,4EH
CALL GOTO
LD B,1 ;1 CHAR
CALL WRL
LD HL,ASC_PASS
LD A,12H
CALL GOTO
LD B,5 ;5 CHAR
CALL WRL
LD HL,ASC_FAIL
LD A,1BH
CALL GOTO
LD B,5 ;5 CHAR
CALL WRL
LD HL,ASC_TOTAL
LD A,56H
CALL GOTO
LD B,6 ;6 CHAR
CALL WRL
JP SHOW_ST
NO_STA: LD HL,EMPTY
ON_STA: LD A,5EH
CALL GOTO
LD B,2 ;2 CHAR
CALL WRL
LD A,00H
LD (CK_SHOW),A
RET
; *****
; * SUBROUTINE *
; * CHECK AND/OR SHOW STATUS *
; *****
;
; REG = ABCDEHLIX
SHOW_ST: LD BC,(ADRS) ;CHECK FOR SHOW
LD A,B ;LOAD ADRS-HIGH(B) TO REG.A
LD E,A
CP 00H
JP Z,SHT1 ;ADRS-HIGH(B) = 00H
CP 10H
JP NZ,SHT2 ;ADRS-HIGH(B) # 10H
LD D,E
JP SHT3 ;ADRS-HIGH(B) = 10H
SHT2: CP 11H
JP NZ,NO_STA ;ADRS-HIGH(B) # 11H
LD IX,AD_BUF
LD L,(IX+0)
LD H,(IX+1)
LD A,(HL) ;LOAD STATUS IN BUFFER TO REG.A
LD E,A
SHT1: LD A,(STATUS) ;LOAD VALUE IN (STATUS) TO REG.A
LD D,A
LD A,C ;LOAD ADRS-LOW(C) TO REG.A
CP 11H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JP Z,CK_BUF ;ADRS-LOW(C) = 11H
CP 10H
JP Z,CK_STA ;ADRS-LOW(C) = 10H
XOR A ;CLEAR CARRY FLAG
LD A,C
CP 09H
JP C,CK_STA ;ADRS-LOW(C) < 09H
JP NO_STA
CK_BUF: LD D,E ;LOAD STATUS IN BUFFER TO REG.D
CK_STA: LD A,D ;LOAD VALUE OF STATUS TO REG.A
CP 00H
CALL Z,SHOW_F ;STATUS = 00H
CP 01H
CALL Z,SHOW_FA ;STATUS = 01H
CP 02H
CALL Z,SHOW_FB ;STATUS = 02H
CP 03H
CALL Z,SHOW_FC ;STATUS = 03H
CP 04H
CALL Z,SHOW_FD ;STATUS = 04H
CP 05H
CALL Z,SHOW_FE ;STATUS = 05H
CP 06H
CALL Z,SHOW_FF ;STATUS = 06H
CP 07H
CALL Z,SHOW_FF ;STATUS = 06H
CP 08H
CALL Z,SHOW_FF ;STATUS = 07H
CP 08H
CALL Z,SHOW_FF ;STATUS = 07H
CP 08H
CALL Z,SHOW_FF ;STATUS = 08H
SHT3: LD A,D
CP 10H
CALL Z,SHOW_P ;STATUS = 10H
LD A,(CK_SHOW)
CP 01H
JP Z,ON_STA ;(CK_SHOW) = 01H
JP NO_STA

; *****
; * SUBROUTINE SHOW_xx *
; *****
;
;
; REG = AHL

SHOW_F: LD HL,F1
JP CKK
SHOW_FA: LD HL,FA1
JP CKK
SHOW_FB: LD HL,FB1
JP CKK
SHOW_FC: LD HL,FC1
JP CKK
SHOW_FD: LD HL,FD1
JP CKK
SHOW_FE: LD HL,FE1
JP CKK
SHOW_FF: LD HL,FF1
JP CKK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SHOW_FG: LD HL,FG1
          JP CKK
SHOW_FH: LD HL,FH1
          JP CKK
SHOW_P:  LD HL,P1
CKK:     PUSH AF
          LD A,01H ;SAVE 01H IN (CK_SHOW)
          LD (CK_SHOW),A ;FOR CHECK
          POP AF
          RET

```

```

; *****
; * SUBROUTINE *
; * SHOW LCD OD (DISPLAY) *
; *****
;
; REG = AHL

```

```

SHOLCDOD: LD A,0BH
           LD (CODE_LCD),A ;SET CODE LCD FOR DISPLAY
           LD HL,LCDOD
           CALL ALL_LCD
           RET

```

```

; *****
; * SUBROUTINE *
; * SHOW LCD 1D (DISPLAY) *
; *****
;
; REG = ABHL

```

```

SHOLCD1D: LD A,0BH
           LD (CODE_LCD),A ;SET CODE LCD FOR DISPLAY
           LD HL,LCD1D
           CALL ALL_LCD
           LD HL,ASC_A
           LD A,02H
           CALL GOTO
           LD B,5 ;5 CHAR
           CALL WRL
           LD HL,ASC_B
           LD A,0BH
           CALL GOTO
           LD B,5 ;5 CHAR
           CALL WRL
           LD HL,ASC_C
           LD A,42H
           CALL GOTO
           LD B,5 ;5 CHAR
           CALL WRL
           LD HL,ASC_D
           LD A,4BH
           CALL GOTO
           LD B,5 ;5 CHAR
           CALL WRL
           LD HL,ASC_E

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,12H
CALL GOTO
LD B,5 ;5 CHAR
CALL WRL
LD HL,ASC_F
LD A,1BH
CALL GOTO
LD B,5 ;5 CHAR
CALL WRL
LD HL,ASC_G
LD A,52H
CALL GOTO
LD B,5 ;5 CHAR
CALL WRL
LD HL,ASC_H
LD A,5BH
CALL GOTO
LD B,5 ;5 CHAR
CALL WRL
RET

; *****
; * SUBROUTINE *
; * SHOW LCD 2D (DISPLAY) *
; *****
;
; REG = ABHL

SHOLCD2D: LD A,0BBH
LD (CODE_LCD),A ;SET CODE LCD FOR DISPLAY
LD HL,LCD2D
CALL ALL_LCD
LD HL,ASC_TG
LD A,0BH
CALL GOTO
LD B,2 ;2 CHAR
CALL WRL
LD HL,ASC_DOT
LD A,0EH
CALL GOTO
LD B,1 ;1 CHAR
CALL WRL
LD HL,ASC_YC
LD A,4AH
CALL GOTO
LD B,3 ;3 CHAR
CALL WRL
LD HL,ASC_DC
LD A,4EH
CALL GOTO
LD B,1 ;1 CHAR
CALL WRL
LD HL,ASC_PASS
LD A,12H
CALL GOTO
LD B,5 ;5 CHAR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL WRL
LD HL,ASC_FAIL
LD A,1BH
CALL GOTO
LD B,5 ;5 CHAR
CALL WRL
LD HL,ASC_TOTAL
LD A,56H
CALL GOTO
LD B,6 ;6 CHAR
CALL WRL
LD HL,EMPTY
LD A,5EH
CALL GOTO
LD B,2 ;2 CHAR
CALL WRL
RET

```

```

; *****
; * SUBROUTINE ALL_LCD *
; *****
;
;
; REG = A

```

```

ALL_LCD: LD A,00H
CALL GOTO
CALL WRLM
LD A,40H
CALL GOTO
CALL WRLM
LD A,10H
CALL GOTO
CALL WRLM
LD A,50H
CALL GOTO
CALL WRLM
RET

```

```

; *****
; * SUBROUTINE BEEP *
; *****
;
;
; REG = AHL

```

```

BEEP: PUSH AF
LD HL,2500H
INO A,(TCR)
SET 1,A
SET 2,A
OUTO (TCR),A
BEEP1: DEC HL
LD A,L
OR H
JP NZ,BEEP1
INO A,(TCR)
RES 1,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        OUTO (TCR),A
        POP AF
        RET

; *****
; *          SUBROUTINE SOUND          *
; *****
;
; REG = A

SOUND:  LD  A,05H
SOUND1: CALL BEEP
        LD  B,0FOH
DLY_SND: DEC  B
        JP  NZ,DLY_SND
        DEC A
        JP  NZ,SOUND1
        RET

; *****
; *          SUBROUTINE TRANSFER DATA  *
; *****
;
; REG = A

TFRDATA: INO  A,(CNTLAO)
        RES 4,A
        OUTO (CNTLAO),A
TFR1:   INO  A,(CNTLBO)
        BIT 5,A
        JP  NZ,TFR1
        CALL DELAY
        INO  A,(CNTLAO)
        SET 4,A
        OUTO (CNTLAO),A
        CALL PRINT
        RET

; *****
; *          PRINT TO CONSOLE          *
; *****
;
; REG = ABHL

PRINT:  LD  HL,HEXPASS
        LD  B,09H
PRINTO: LD  A,(HL)
PRINT1: CALL CONOUT
        INC HL
        NOP
        NOP
        NOP
        DJNZ PRINTO
        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; * SEND 1 CHARACTER TO CONSOLE *
; *****
;
; IN = A
; REG = E

CONOUT: INO E,(STATO)
        BIT 1,E
        JP Z,CONOUT
        OUTO (TDRO),A
        RET

; *****
; * SUBROUTINE PASS 1 *
; *****
;
; REG = ABCHL

PASS1: LD BC,(ADRS)
        LD A,B ;CHECK ADRS-HIGH(B)
        CP 11H
        JP Z,DLT_YP ;ADRS-HIGH(B) = 11H
        CALL BEEP
        LD HL,1000H
        LD (ADRS),HL ;SAVE CODE PASS
        JP DLT_YP1

DLT_YP: CALL BEEP
        LD HL,1110H
        LD (ADRS),HL ;SAVE CODE DELETE-PASS

DLT_YP1: CALL CLRLED
        LD A,10H
        LD (STATUS),A ;SAVE STATUS OF PASS
        JP MAIN3

; *****
; * SUBROUTINE FAIL 1 *
; *****
;
; REG = ABCHL

FAIL1: LD BC,(ADRS)
        LD A,B ;CHECK ADRS-HIGH(B)
        CP 11H
        JP Z,DLT_YF ;ADRS-HIGH(B) = 11H
        CALL BEEP
        LD HL,0000H
        LD (ADRS),HL ;SAVE CODE FAIL
        JP DLT_YF1

DLT_YF: CALL BEEP
        LD HL,1100H
        LD (ADRS),HL ;SAVE CODE DELETE-FAIL

DLT_YF1: CALL CLRLED
        LD A,00H
        LD (STATUS),A ;SAVE STATUS OF FAIL
        JP MAIN3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *      SUBROUTINE DELETE      *
; *****
;
; REG = ABCDEHLIX

DELETE:  LD  BC,(ADRS)
         LD  A,B                ;CHECK ADRS-HIGH(B)
         CP  11H
         JP  Z,DLT_YD          ;ADRS-HIGH(B) = 11H
         CALL BEEP
         LD  BC,(AD_BUF)       ;LOAD FUTURE ADDRESS OF BUFFER
         DEC BC                ;PRESENT ADDRESS OF BUFFER
         PUSH BC               ;SAVE PRESENT ADDRESS OF BUFFER
         PUSH BC
         POP  HL
         LD  DE,START_BF      ;CHECK ADDRESS OF BUFFER
         SBC HL,DE
         JP  NC,DLT0
         POP  BC
         JP  KEY

DLT0:   LD  HL,1111H
         LD  (ADRS),HL        ;SAVE CODE DELETE
         CALL CLRLED          ;START FIRST DELETE
         LD  HL,HEXTOTAL
         LD  IX,SAVE_TTL
         LD  B,03H            ;COUNTER

DLT1:   LD  A,(HL)
         LD  (IX+0),A         ;SAVE VALUE OF HEXTOTAL
         INC HL                ;IN SAVE_TTL
         INC IX
         DEC B
         JP  NZ,DLT1          ;COUNTER(B) = 00
         POP  BC
         LD  (AD_BUF),BC      ;LOAD PRESENT ADDRESS OF BUFFER
         LD  IX,(AD_BUF)      ;TO (AD_BUF) & (SAVE_BUF)
         LD  (SAVE_BUF),IX
         JP  DLT_YD1

DLT_YD: CALL BEEP
         LD  HL,1111H
         LD  (ADRS),HL        ;SAVE CODE DELETE
         LD  BC,(AD_BUF)
         DEC BC                ;BUFFER DOWN 1 (STATUS - 1)

DLT_YD1: PUSH BC             ;CHECK ADDRESS OF BUFFER
         POP  HL              ;SAVE BC TO HL
         XOR  A               ;CLEAR CARRY FLAG
         LD  DE,START_BF
         SBC HL,DE
         JP  NC,DLT_YD2
         JP  DELT_DEL

DLT_YD2: LD  (AD_BUF),BC
         LD  A,(BC)
         LD  (OLD_STA),A
         CALL T_COUNTD        ;TOTAL - 1
         JP  MAIN3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *      SUBROUTINE OFF LED      *
; *****
;
; REG = ABC

CLRLED:  LD  BC, (ADRS)
         LD  A,B                ;CHECK ADRS-HIGH(B)
         CP  11H
         JP  Z,D_LED            ;ADRS-HIGH(B) = 11H
         CP  10H
         JP  Z,P_LED            ;ADRS-HIGH(B) = 10H
         CP  00H
         JP  Z,F_LED            ;ADRS-HIGH(B) = 00H
         JP  FINAL

D_LED:   LD  A,C                ;CHECK ADRS-LOW(C)
         CP  11H
         JP  Z,D_LED1           ;ADRS-LOW(C) = 11H
         CP  10H
         JP  Z,P_LED1           ;ADRS-LOW(C) = 10H
         CP  00H
         JP  Z,F_LED1           ;ADRS-LOW(C) = 00H
         JP  SF_LED

P_LED:   LD  BC,SWLED
         LD  A,1FH
         OUT (C),A              ;ON LED PASS
         JP  SF_LED

F_LED:   LD  BC,SWLED
         LD  A,2FH
         OUT (C),A              ;ON LED FAIL
         JP  SF_LED

D_LED1:  LD  BC,SWLED
         LD  A,8FH
         OUT (C),A              ;ON LED DELETE
         JP  SF_LED

P_LED1:  LD  BC,SWLED
         LD  A,9FH
         OUT (C),A              ;ON LED DELETE & PASS
         JP  SF_LED

F_LED1:  LD  BC,SWLED
         LD  A,0AFH
         OUT (C),A              ;ON LED DELETE & FAIL

SF_LED:  LD  BC, (ADRS)
         LD  A,C                ;CHECK ADRS-LOW(C)
         CP  01H
         JP  Z,FA_LED           ;ADRS-LOW(C) = 01H
         CP  02H
         JP  Z,FB_LED           ;ADRS-LOW(C) = 02H
         CP  03H
         JP  Z,FC_LED           ;ADRS-LOW(C) = 03H
         CP  04H
         JP  Z,FD_LED           ;ADRS-LOW(C) = 04H
         CP  05H
         JP  Z,FE_LED           ;ADRS-LOW(C) = 05H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 00H
JP Z,FAIL_F ;ADRS-HIGH(B) = 00H
CP 11H
JP Z,DELT_FF ;ADRS-HIGH(B) = 11H
JP KEY

CK_CB7: CP 07H ;CHECK ADRS-LOW(C)
JP NZ,CK_CB8 ;ADRS-LOW(C) # 07H
LD A,B ;CHECK ADRS-HIGH(B)
CP 00H
JP Z,FAIL_G ;ADRS-HIGH(B) = 00H
CP 11H
JP Z,DELT_FG ;ADRS-HIGH(B) = 11H
JP KEY

CK_CB8: CP 08H ;CHECK ADRS-LOW(C)
JP NZ,CK_CBP ;ADRS-LOW(C) # 08H
LD A,B ;CHECK ADRS-HIGH(B)
CP 00H
JP Z,FAIL_H ;ADRS-HIGH(B) = 00H
CP 11H
JP Z,DELT_FH ;ADRS-HIGH(B) = 11H
JP KEY

CK_CBP: CP 10H ;CHECK ADRS-LOW(C)
JP NZ,CK_CBD ;ADRS-LOW(C) # 10H
LD A,B ;CHECK ADRS-HIGH(B)
CP 11H
JP Z,DELT_P ;ADRS-HIGH(B) = 11H
JP KEY

CK_CBD: CP 11H ;CHECK ADRS-LOW(C)
JP NZ,KEY ;ADRS-LOW(C) # 11H
LD A,B ;CHECK ADRS-HIGH(B)
CP 11H
JP Z,DELT_DEL ;ADRS-HIGH(B) = 11H
JP KEY

```

```

; *****
; * SUBROUTINE *
; * INCREMENT COUNTER *
; *****
;
; REG = ABCDHLIX

```

```

T_COUNT: LD IX,HEXTOTAL
LD HL,ASC_TOTAL
LD A,06H ;CODE OF TOTAL COUNTER
LD (NUMBER),A ;SAVE CODE
LD D,03H
LD B,05H
JP CK_COUNT

P_COUNT: LD IX,HEXPASS
LD HL,ASC_PASS
JP COUNT

F_COUNT: LD IX,HEXFAIL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 06H
JP Z,FF_LED ;ADRS-LOW(C) = 06H
CP 07H
JP Z,FG_LED ;ADRS-LOW(C) = 07H
CP 08H
JP Z,FH_LED ;ADRS-LOW(C) = 08H
LD BC,PLEDS
LD A,00H
OUT (C),A ;OFF ALL SUB-LED
JP FINAL

FA_LED: LD BC,PLEDS
LD A,01H
OUT (C),A ;ON LED A
JP FINAL

FB_LED: LD BC,PLEDS
LD A,02H
OUT (C),A ;ON LED B
JP FINAL

FC_LED: LD BC,PLEDS
LD A,04H
OUT (C),A ;ON LED C
JP FINAL

FD_LED: LD BC,PLEDS
LD A,08H
OUT (C),A ;ON LED D
JP FINAL

FE_LED: LD BC,PLEDS
LD A,10H
OUT (C),A ;ON LED E
JP FINAL

FF_LED: LD BC,PLEDS
LD A,20H
OUT (C),A ;ON LED F
JP FINAL

FG_LED: LD BC,PLEDS
LD A,40H
OUT (C),A ;ON LED G
JP FINAL

FH_LED: LD BC,PLEDS
LD A,80H
OUT (C),A ;ON LED H

FINAL: RET

; *****
; * SUBROUTINE SUB-FAIL *
; *****
;
; REG = ABCDE

FAIL_A1: LD A,01H
PUSH AF
JP FAIL_S

FAIL_B1: LD A,02H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                PUSH AF
                JP   FAIL_S

FAIL_C1: LD    A,03H
          PUSH AF
          JP   FAIL_S

FAIL_D1: LD    A,04H
          PUSH AF
          JP   FAIL_S

FAIL_E1: LD    A,05H
          PUSH AF
          JP   FAIL_S

FAIL_F1: LD    A,06H
          PUSH AF
          JP   FAIL_S

FAIL_G1: LD    A,07H
          PUSH AF
          JP   FAIL_S

FAIL_H1: LD    A,08H
          PUSH AF

FAIL_S:   LD    BC,(ADRS)
          LD    A,B                ;CHECK ADRS-HIGH(B)
          CP    11H                ;ADRS-HIGH(B) = 11H
          JP    Z,DLT_YS
          CP    00H
          JP    Z,SAVE_S
          POP  AF
          JP    KEY

DLT_YS:  LD    A,C                ;CHECK ADRS-LOW(C)
          CP    00H
          JP    NZ,DELT_DEL       ;ADRS-LOW(C) # 00H

SAVE_S:  XOR  A                ;CLEAR REG.A
          POP  DE                ;GET CODE CONDITION
          ADD  A,D
          LD   C,A
          LD   (ADRS),BC         ;SAVE CODE CONDITION IN (ADRS)
          LD   (STATUS),A       ;SAVE CODE SUB-FAIL IN (STATUS)
          CALL CLRLED
          JP   MAIN3

```

```

; *****
; *      SUBROUTINE ENTER      *
; *****
;
; REG = ABC

```

```

ENTER:   LD    BC,(ADRS)
          LD    A,C                ;CHECK ADRS-LOW(C)
          CP    00H
          JP    NZ,CK_CB1         ;ADRS-LOW(C) # 00H
          LD    A,B                ;CHECK ADRS-HIGH(B)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP      10H
JP      Z,PASS                ;ADRS-HIGH(B) = 10H
CP      00H
JP      Z,FAIL                ;ADRS-HIGH(B) = 00H
CP      11H
JP      Z,DELT_F             ;ADRS-HIGH(B) = 11H
JP      KEY

CK_CB1: CP      01H                ;CHECK ADRS-LOW(C)
JP      NZ,CK_CB2            ;ADRS-LOW(C) # 01H
LD      A,B                  ;CHECK ADRS-HIGH(B)
CP      00H
JP      Z,FAIL_A             ;ADRS-HIGH(B) = 00H
CP      11H
JP      Z,DELT_FA           ;ADRS-HIGH(B) = 11H
JP      KEY

CK_CB2: CP      02H                ;CHECK ADRS-LOW(C)
JP      NZ,CK_CB3            ;ADRS-LOW(C) # 02H
LD      A,B                  ;CHECK ADRS-HIGH(B)
CP      00H
JP      Z,FAIL_B             ;ADRS-HIGH(B) = 00H
CP      11H
JP      Z,DELT_FB           ;ADRS-HIGH(B) = 11H
JP      KEY

CK_CB3: CP      03H                ;CHECK ADRS-LOW(C)
JP      NZ,CK_CB4            ;ADRS-LOW(C) # 03H
LD      A,B                  ;CHECK ADRS-HIGH(B)
CP      00H
JP      Z,FAIL_C             ;ADRS-HIGH(B) = 00H
CP      11H
JP      Z,DELT_FC           ;ADRS-HIGH(B) = 11H
JP      KEY

CK_CB4: CP      04H                ;CHECK ADRS-LOW(C)
JP      NZ,CK_CB5            ;ADRS-LOW(C) # 04H
LD      A,B                  ;CHECK ADRS-HIGH(B)
CP      00H
JP      Z,FAIL_D             ;ADRS-HIGH(B) = 00H
CP      11H
JP      Z,DELT_FD           ;ADRS-HIGH(B) = 11H
JP      KEY

CK_CB5: CP      05H                ;CHECK ADRS-LOW(C)
JP      NZ,CK_CB6            ;ADRS-LOW(C) # 05H
LD      A,B                  ;CHECK ADRS-HIGH(B)
CP      00H
JP      Z,FAIL_E             ;ADRS-HIGH(B) = 00H
CP      11H
JP      Z,DELT_FE           ;ADRS-HIGH(B) = 11H
JP      KEY

CK_CB6: CP      06H                ;CHECK ADRS-LOW(C)
JP      NZ,CK_CB7            ;ADRS-LOW(C) # 06H
LD      A,B                  ;CHECK ADRS-HIGH(B)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,ASC_FAIL
JP COUNT
A_H_CNT: LD IX,HEX_A
LD HL,ASC_A
JP COUNT
B_H_CNT: LD IX,HEX_B
LD HL,ASC_B
JP COUNT
C_H_CNT: LD IX,HEX_C
LD HL,ASC_C
JP COUNT
D_H_CNT: LD IX,HEX_D
LD HL,ASC_D
JP COUNT
E_H_CNT: LD IX,HEX_E
LD HL,ASC_E
JP COUNT
F_H_CNT: LD IX,HEX_F
LD HL,ASC_F
JP COUNT
G_H_CNT: LD IX,HEX_G
LD HL,ASC_G
JP COUNT
H_H_CNT: LD IX,HEX_H
LD HL,ASC_H

COUNT: LD A,05H ;CODE OF ANY COUNTER
LD (NUMBER),A ;SAVE CODE
LD D,03H
LD B,04H
CK_COUNT: XOR A ;CLEAR CARRY FLAG
LD A,(NUMBER) ;CHECK CODE
CP 06H
JP C,COUNT_S ;CODE = 05H
LD A,(IX+2) ;CHECK COUNTER OF TOTAL VALUE
CP 99H
JP NZ,COUNT1 ;NO MAXIMUM
JP COUNT0
COUNT_S: LD A,(IX+2) ;CHECK COUNTER OF ANY VALUE
CP 09H
JP NZ,COUNT1 ;NO MAXIMUM
COUNT0: LD A,(IX+1) ;CHECK COUNTER OF ALL VALUE
CP 99H
JP NZ,COUNT1 ;NO MAXIMUM
LD A,(IX+0)
CP 99H
JP Z,INC_END ;MAXIMUM
COUNT1: LD A,(IX+0) ;START COUNTER
LD C,A
AND 0FH
CP 09H
JP Z,COUNT2
LD A,C ;BIT0 < 9
INC A
LD (IX+0),A
JP INC_ACI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COUNT2: LD A,C ;BIT0 = 9
          AND OF0H
          CP 90H
          JP Z,COUNT3
          ADD A,10H ;BIT1 < 9
          LD (IX+0),A
          JP INC_ACI
COUNT3: LD (IX+0),00H
          LD A,(IX+1) ;BIT1 = 9
          LD C,A
          AND OFH
          CP 09H
          JP Z,COUNT4 ;BIT2 < 9
          LD A,C
          INC A
          LD (IX+1),A
          JP INC_ACI
COUNT4: LD A,C ;BIT2 = 9
          AND OF0H
          CP 90H
          JP Z,COUNT5 ;BIT3 < 9
          ADD A,10H
          LD (IX+1),A
          JP INC_ACI
COUNT5: LD (IX+1),00H ;BIT3 = 9
          LD A,(IX+2)
          LD C,A
          AND OFH
          CP 09H
          JP Z,COUNT6 ;BIT4 < 9
          LD A,C
          INC A
          LD (IX+2),A
          JP INC_ACI
COUNT6: LD A,(NUMBER) ;CHECK CODE COUNTER
          CP 06H
          JP C,COUNT9 ;CODE = 05H
          LD A,C ;BIT4 = 9
          AND OF0H
          CP 90H
          JP Z,COUNT7
          ADD A,10H
          JP COUNT8
COUNT7: LD A,99H
COUNT8: LD (IX+2),A
          JP INC_ACI
COUNT9: LD A,09H ;BIT4 = 9
          LD (IX+2),A
INC_ACI: CALL CH_ASCII ;CHANGE TO ASCII COUNTER
INC_END: RET

```

```

; *****
; * SUBROUTINE *
; * DECREMENT COUNTER *
; *****
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; REG = ABCDHLIX

```
T_COUNTD: LD IX,HEXTOTAL
           LD HL,ASC_TOTAL
           LD A,06H ;CODE OF TOTAL COUNTER
           LD (NUMBER),A ;SAVE CODE
           LD D,03H
           LD B,05H
           JP C_COUNTD
P_COUNTD: LD IX,HEXPASS
           LD HL,ASC_PASS
           JP COUNTD
F_COUNTD: LD IX,HEXFAIL
           LD HL,ASC_FAIL
           JP COUNTD
A_H_CNM: LD IX,HEX_A
          LD HL,ASC_A
          JP COUNTD
B_H_CNM: LD IX,HEX_B
          LD HL,ASC_B
          JP COUNTD
C_H_CNM: LD IX,HEX_C
          LD HL,ASC_C
          JP COUNTD
D_H_CNM: LD IX,HEX_D
          LD HL,ASC_D
          JP COUNTD
E_H_CNM: LD IX,HEX_E
          LD HL,ASC_E
          JP COUNTD
F_H_CNM: LD IX,HEX_F
          LD HL,ASC_F
          JP COUNTD
G_H_CNM: LD IX,HEX_G
          LD HL,ASC_G
          JP COUNTD
H_H_CNM: LD IX,HEX_H
          LD HL,ASC_H

COUNTD: LD A,05H ;CODE OF ANY COUNTER
          LD (NUMBER),A ;SAVE CODE
          LD D,03H
          LD B,04H
C_COUNTD: LD A,(IX+2) ;CHECK COUNTER OF ALL VALUE
          CP 00H
          JP NZ,COUNTD_S ;NO MINIMUM
          LD A,(IX+1)
          CP 00H
          JP NZ,COUNTD_S ;NO MINIMUM
          LD A,(IX+0)
          CP 00H
          JP Z,DEC_END ;MINIMUM
COUNTD_S: LD A,(IX+0) ;START COUNTER
           LD C,A
           AND 0FH
           CP 00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JP    Z,COUNTD1
        LD    A,C                ;BIT0 > 0
        DEC  A
        LD    (IX+0),A
        JP    DEC_ACI
COUNTD1: LD    A,C                ;BIT0 = 0
        AND  OFOH
        CP    00H
        JP    Z,COUNTD2
        SUB  10H
        ADD  A,09H              ;BIT1 > 0
        LD    (IX+0),A
        JP    DEC_ACI
COUNTD2: LD    (IX+0),99H
        LD    A,(IX+1)         ;BIT1 = 0
        LD    C,A
        AND  OFH
        CP    00H
        JP    Z,COUNTD3         ;BIT2 > 0
        LD    A,C
        DEC  A
        LD    (IX+1),A
        JP    DEC_ACI
COUNTD3: LD    A,C                ;BIT2 = 0
        AND  OFOH
        CP    00H
        JP    Z,COUNTD4         ;BIT3 > 0
        SUB  10H
        ADD  A,09H
        LD    (IX+1),A
        JP    DEC_ACI
COUNTD4: LD    (IX+1),99H       ;BIT3 = 0
        LD    A,(IX+2)
        LD    C,A
        AND  OFH
        CP    00H
        JP    Z,COUNTD5         ;BIT4 > 0
        LD    A,C
        DEC  A
        LD    (IX+2),A
        JP    DEC_ACI
COUNTD5: LD    A,(NUMBER)       ;CHECK CODE COUNTER
        CP    06H
        JP    C,COUNTD6         ;CODE = 05H
        LD    A,C                ;BIT4 = 0
        AND  OFOH
        CP    00H
        JP    Z,COUNTD6
        SUB  10H
        ADD  A,09H
        JP    COUNTD7
COUNTD6: LD    A,00H           ;BIT4 = 0
COUNTD7: LD    (IX+2),A
DEC_ACI:  CALL  CH_ASCII        ;CHANGE TO ASCII
DEC_END:  RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *      SUBROUTINE INCREMENT      *
; *      COUNTER OF INTERVAL      *
; *****
;
; REG = ACIX

CON_4P:  LD  IX,INTV_P
         JP  COUNTP
CON_4F:  LD  IX,INTV_F
COUNTP: LD  A,(IX+1)                ;CHECK COUNTER OF INTERVAL
         CP  02H
         JP  NZ,COUN_S              ;NO MAXIMUM
         LD  A,(IX+0)
         CP  50H
         JP  NZ,COUN_S              ;NO MAXIMUM
         JP  Z,INCP_END             ;MAXIMUM
COUN_S:  LD  A,(IX+0)                ;START COUNTER
         LD  C,A
         AND 0FH
         CP  09H
         JP  Z,COUN1
         LD  A,C                    ;BIT0 < 9
         INC A
         LD  (IX+0),A
         JP  INCP_END
COUN1:   LD  A,C                    ;BIT0 = 9
         AND 0FOH
         LD  C,A
         CP  90H
         JP  Z,COUN2
         LD  A,C
         ADD A,10H                  ;BIT1 < 9
         LD  (IX+0),A
         JP  INCP_END
COUN2:   LD  (IX+0),00H
         LD  A,(IX+1)              ;BIT1 = 9
         INC A
         LD  (IX+1),A
INCP_END: RET

; *****
; *      SUBROUTINE DECREMENT      *
; *      COUNTER OF INTERVAL      *
; *****
;
; REG = AIX

DCON_4P: LD  IX,INTV_P
         JP  COUNTDP
DCON_4F: LD  IX,INTV_F
COUNTDP: LD  (IX+0),00H           ;CLEAR INTERVAL COUNTER
         LD  (IX+1),00H
         RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *      SUBROUTINE CHECK      *
; *      VALUE OF INTERVAL    *
; *****
;
; REG = ABCDHLIXIY

CK_ITV0: LD  A,(CK_ITV)          ;LOAD CODE CHECK INTERVAL
         CP  01H
         JP  Z,CK_ITV1          ;CODE INTERVAL = 01H
         LD  IX,HEXTOTAL        ;CHECK START INTERVAL
         LD  IY,INTERVAL
         LD  A,(IX+0)
         LD  B,(IY+0)
         CP  B
         JP  C,ITV_END          ;REG.A < REG.B
         LD  A,(IX+1)
         LD  B,(IY+1)
         CP  B
         JP  C,ITV_END          ;REG.A < REG.B
CK_ITV1: LD  A,01H
         LD  (CK_ITV),A        ;SAVE CODE INTERVAL
         LD  BC,(AD_BUF)
         PUSH BC                ;SAVE FUTURE (AD_BUF)
CK_ITV2: DEC  BC
         PUSH BC                ;SAVE PRESENT (AD_BUF)
         LD  A,(BC)
         BIT  4,A
         JP  NZ,ITV_P          ;(BC) = 10H
ITV_F:   CALL CON_4F           ;(BC) = 00H
         JP  ITV
ITV_P:   CALL CON_4P           ;(BC) = 10H
ITV:     LD  IX,INTV_F
         LD  IY,INTV_P
         XOR  A                 ;CLEAR REG.A
         LD  A,(IX+0)
         LD  D,(IY+0)
         ADD  A,D
         DAA                    ;CHANGE TO BCD
         LD  L,A
         LD  A,(IX+1)
         LD  D,(IY+1)
         ADC  A,D
         DAA                    ;CHANGE TO BCD
         LD  H,A
         LD  BC,(INTERVAL)
         LD  A,L
         CP  C
         JP  NZ,CK_ITV3
         LD  A,H
         CP  B
         JP  Z,CK_ITV4
CK_ITV3: POP  BC                ;RETURN BC IN STACK
         JP  CK_ITV2
CK_ITV4: POP  BC                ;CLEAR STACK
         CALL CL_YDAR
         POP  HL                ;RETURN FUTURE BC TO HL
         LD  (AD_BUF),HL       ;LOAD (AD_BUF)
         CALL DCON_4F          ;CLEAR INTERVAL COUNTER

```

```

CALL DCON_4P
ITV_END: RET

; *****
; *          SUBROUTINE          *
; *          STATUS OF BUFFER    *
; *****
;
; REG = ABCDEHLIXIY

STA_BUF: LD IX,AD_BUF           ;CHECK NUMBER AT END-BUFFER
          LD HL,END_BF
          INC HL
          LD A,(IX+1)
          CP H
          JP NZ,NDELO           ;AD_BUF <= END_B
          LD A,(IX+0)
          CP L
          JP NZ,NDELO           ;AD_BUF <= END_B

; ***** SHIFT DATA UP 1 BYTE *****
;
;          LD DE,START_BF       ;SHIFT STATUS IN STATUS
;          PUSH DE              ;BUFFER UP 1 BYTE
;          POP BC               ;SAVE REG.DE IN REG.BC
;          INC BC
DELE: LD A,(BC)
      LD (DE),A
      LD A,H
      CP B
      JP NZ,NEND
      LD A,L
      CP C
      JP Z,NDEL1
NEND: INC BC
      INC DE
      JP DELE
NDEL1: DEC BC
       LD (IX+1),B
       LD (IX+0),C
NDELO: LD HL,(ADRS)           ;FIND CODE STATUS FOR
       LD A,H                 ;SAVE IN STATUS BUFFER
       CP 10H
       JP NZ,NDEL2
       LD A,10H               ;CODE STATUS OF PASS
       JP GET_STA
NDEL2: CP 00H
       JP Z,NDEL_FL
       CP 11H
       JP NZ,KEY
NDEL_FL: LD A,L
         CP 00H
         JP NZ,NDELA
         LD A,00H             ;CODE STATUS OF FAIL
         JP GET_STA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NDELA:  CP  01H
        JP  NZ,NDELB
        LD  A,01H           ;CODE STATUS OF SUB-A
        JP  GET_STA
NDELB:  CP  02H
        JP  NZ,NDELC
        LD  A,02H           ;CODE STATUS OF SUB-B
        JP  GET_STA
NDELC:  CP  03H
        JP  NZ,NDELD
        LD  A,03H           ;CODE STATUS OF SUB-C
        JP  GET_STA
NDELD:  CP  04H
        JP  NZ,NDELE
        LD  A,04H           ;CODE STATUS OF SUB-D
        JP  GET_STA
NDELE:  CP  05H
        JP  NZ,NDELF
        LD  A,05H           ;CODE STATUS OF SUB-E
        JP  GET_STA
NDELF:  CP  06H
        JP  NZ,NDELG
        LD  A,06H           ;CODE STATUS OF SUB-F
        JP  GET_STA
NDELG:  CP  07H
        JP  NZ,NDELH
        LD  A,07H           ;CODE STATUS OF SUB-G
        JP  GET_STA
NDELH:  CP  08H
        JP  NZ,NDEL_PS
        LD  A,08H           ;CODE STATUS OF SUB-H
        JP  GET_STA
NDEL_PS: CP  10H
        JP  NZ,KEY
        LD  A,10H           ;CODE STATUS OF PASS
GET_STA: LD  C,(IX+0)
        LD  B,(IX+1)
        LD  (BC),A
        INC BC              ;FUTURE AD_BUF
        LD  (IX+0),C
        LD  (IX+1),B
        RET

```

```

; *****
; *          SUBROUTINE          *
; *    OLD VALUE TO BUFFER      *
; *****
;
; REG = ABCHLIX

```

```

; ***** OLD TOTAL VALUE *****

```

```

OLD_VLU: LD  HL,SAVE_TTL
        LD  DE,HEXTOTAL
        LD  BC,03H
        LDIR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD IX,HEXTOTAL
LD HL,ASC_TOTAL
LD A,06H
LD (NUMBER),A
LD D,03H
LD B,05H
CALL CH_AC16 ;CHANGE TO ASCII

; ***** OLD BUFFER VALUE *****

LD IX,SAVE_BUF
LD C,(IX+0)
LD B,(IX+1)
INC BC
LD (AD_BUF),BC
RET

; *****
; * SUBROUTINE *
; * CHECK OLD STATUS *
; *****
;
; REG = ABCDE

CK_OLD: LD A,(OLD_STA) ;OLD STATUS VALUE
LD E,A ;OLD STATUS VALUE
LD BC,(ADRS)
LD A,C ;NEW STATUS VALUE
LD D,A ;NEW STATUS VALUE
SUB E
JP Z,DELT_DEL
LD A,E
CP 10H
JP Z,OLD_P ;OLD VALUE IS PASS
XOR A ;CLEAR CARRY FLAG
LD A,E
SUB 09H
JP NC,DELT_DEL
PUSH DE ;SAVE OLD&NEW STATUS IN STACK
CALL CK_FDEC ;CHECK TYPE OF FAIL FOR DELETE
POP DE ;GET OLD(E) & NEW(D) STATUS
LD A,D
CP 10H
JP NZ,CK_OLD1
CALL P_COUNT
CALL F_COUNTD
RET

OLD_P: PUSH DE
CALL F_COUNT
CALL P_COUNTD
POP DE

CK_OLD1: LD A,D
CP 09H
JP NC,DELT_DEL
CALL CK_FINC
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *      SUBROUTINE CHECK      *
; *      TYPE OF FAIL FOR DELETE *
; *****
;
; REG = A

```

```

CK_FDEC: LD  A,(OLD_STA)
          CP  01H
          JP  NZ,OLD02
          JP  A_H_CNM
OLD02:   CP  02H
          JP  NZ,OLD03
          JP  B_H_CNM
OLD03:   CP  03H
          JP  NZ,OLD04
          JP  C_H_CNM
OLD04:   CP  04H
          JP  NZ,OLD05
          JP  D_H_CNM
OLD05:   CP  05H
          JP  NZ,OLD06
          JP  E_H_CNM
OLD06:   CP  06H
          JP  NZ,OLD07
          JP  F_H_CNM
OLD07:   CP  07H
          JP  NZ,OLD08
          JP  G_H_CNM
OLD08:   CP  08H
          JP  NZ,OLD09
          JP  H_H_CNM
OLD09:   RET

```

```

; *****
; *      SUBROUTINE CHECK      *
; *      TYPE OF FAIL FOR INCREMENT *
; *****
;
; REG = ADE

```

```

CK_FINC: CP  01H
          JP  NZ,NEW02
          JP  A_H_CNT
NEW02:   CP  02H
          JP  NZ,NEW03
          JP  B_H_CNT
NEW03:   CP  03H
          JP  NZ,NEW04
          JP  C_H_CNT
NEW04:   CP  04H
          JP  NZ,NEW05
          JP  D_H_CNT
NEW05:   CP  05H
          JP  NZ,NEW06
          JP  E_H_CNT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NEW06:  CP   06H
        JP   NZ,NEW07
        JP   F_H_CNT
NEW07:  CP   07H
        JP   NZ,NEW08
        JP   G_H_CNT
NEW08:  CP   08H
        JP   NZ,NEW09
        JP   H_H_CNT
NEW09:  RET

```

```

; *****
; *      SUBROUTINE CALCURATE      *
; *      YEILD COMMUTATIVE        *
; *****
;
; REG = ABCDEHLIXIY

```

```

CL_YDCM: LD   IX,HEXPASS
        LD   A,(IX+2)
        CP   00H
        JP   NZ,CAL1
        LD   A,(IX+1)
        CP   00H
        JP   NZ,CAL1
        LD   A,(IX+0)
        CP   00H
        JP   NZ,CAL1
        LD   BC,0000H
        LD   (HEX_YC),BC ;SHOW YEILD COM. = 000.0%
        LD   A,06H ;FOR ASCII EVEN
        LD   (NUMBER),A
        LD   IX,HEX_YC
        LD   HL,ASC_YC
        LD   B,03H
        LD   D,02H
        JP   CH_ASCII ;CHANGE TO ASCII
CAL1:   LD   IX,CLDEND ;DIVIDEND (PASS)
        LD   IY,CLDSOR ;DIVISOR (TOTAL)
        LD   (IX+3),00H ;FOR DOT
        LD   (IX+4),00H ;DIVIDEND X 100%
        LD   (IY+0),00H
        LD   (IY+1),00H
        LD   HL,HEXTOTAL
        LD   B,03H
CAL11:  LD   A,(HL)
        LD   (IY+4),A
        INC  HL
        DEC  IY
        DEC  B
        JP   NZ,CAL11
        LD   HL,HEXPASS
        LD   B,03H
CAL12:  LD   A,(HL)
        LD   (IX+2),A
        INC  HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEC IX
DEC B
JP NZ,CAL12
LD IX,CLDEND ;START CALCULATE
CALL CLE40
CALL CLDH
LD IX,CLDSOR
CALL CLE40
CALL CLDH
CALL CLDV
CALL CLHD
CALL CLX40
CL_ENDC: LD IY,HEX_YC
LD IX,CLBUF
LD A,(IX+9)
AND 0FH
CP 06H
JP C,CL_ENDC1 ;CLEAR CARRY FLAG
XOR A
LD A,(IX+9)
ADD A,10H
DAA
LD (IX+9),A
LD A,(IX+8)
ADC A,00H
DAA
LD (IX+8),A
LD A,(IX+7)
ADC A,00H
DAA
LD (IX+7),A
CL_ENDC1: LD C,02H
CL_ENDC2: LD A,(IX+9)
AND 0FOH
SRL A
SRL A
SRL A
SRL A
LD B,A
LD A,(IX+8)
AND 0FH
SLA A
SLA A
SLA A
SLA A
ADD A,B
LD (IY+0),A
DEC IX
INC IY
DEC C
JP NZ,CL_ENDC2 ;FOR ASCII EVEN
LD A,06H
LD (NUMBER),A
LD IX,HEX_YC
LD HL,ASC_YC
LD D,02H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD B,03H
JP CH_ASCII ;CHANGE TO ASCII

; *****
; * SUBROUTINE CALCURATE *
; * YEILD ALERT *
; *****
;
; REG = ABCDEHLIXIY

CL_YDAR: LD IX,INTV_P
LD A,(IX+1)
CP 00H
JP NZ,CAL2
LD A,(IX+0)
CP 00H
JP NZ,CAL2
LD BC,1000H
LD (HEX_YA),BC ;SHOW YEILD COM. = 100.0%
LD A,06H ;FOR ASCII EVEN
LD (NUMBER),A
LD IX,HEX_YA
LD HL,ASC_YA
LD B,03H
LD D,02H
JP CH_ASCII ;CHANGE TO ASCII
CAL2: LD IX,CLDEND ;DIVIDEND *PASS (INTV)é
LD IY,CLDSOR ;DIVISOR (INTERVAL VALUE)
LD (IX+0),00H
LD (IX+3),00H ;FOR DOT
LD (IX+4),00H ;DIVIDEND X 100%
LD (IY+0),00H
LD (IY+1),00H
LD (IY+2),00H
LD HL,INTERVAL
LD B,02H
CAL21: LD A,(HL)
LD (IY+4),A
INC HL
DEC IY
DEC B
JP NZ,CAL21
LD HL,INTV_P
LD B,02H
CAL22: LD A,(HL)
LD (IX+2),A
INC HL
DEC IX
DEC B
JP NZ,CAL22
LD IX,CLDEND ;START CALCULATE
CALL CLE40
CALL CLDH
LD IX,CLDSOR
CALL CLE40
CALL CLDH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL CLDV
CALL CLHD
CALL CLX40
CL_ENDA: LD IY,HEX_YA
LD IX,CLBUF
LD A,(IX+9)
AND OFH
CP 06H
JP C,CL_ENDA1
XOR A ;CLEAR CARRY FLAG
LD A,(IX+9)
ADD A,10H
DAA
LD (IX+9),A
LD A,(IX+8)
ADC A,00H
DAA
LD (IX+8),A
LD A,(IX+7)
ADC A,00H
DAA
LD (IX+7),A
CL_ENDA1: LD C,02H
CL_ENDA2: LD A,(IX+9)
AND OFOH
SRL A
SRL A
SRL A
SRL A
LD B,A
LD A,(IX+8)
AND OFH
SLA A
SLA A
SLA A
SLA A
ADD A,B
LD (IY+0),A
DEC IX
INC IY
DEC C
JP NZ,CL_ENDA2
LD A,06H ;FOR ASCII EVEN
LD (NUMBER),A
LD IX,HEX_YA
LD HL,ASC_YA
LD D,02H
LD B,03H
JP CH_ASCII ;CHANGE TO ASCII

;***** CLE SUB *****
;REG = BCDEHLIX

CLE: LD HL,CLSTK+14
LD DE,CLSTK+19
LD BC,15

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LDDR
RET

;***** CLE40 SUB *****
;IN = (IX+0)
;REG = BCDEHL

CLE40:   CALL CLE
        PUSH IX
        POP HL
        LD DE,CLSTK
        LD BC,5
        LDIR
        RET

;***** CLX SUB *****
;REG = BCDEHL

CLX:     LD HL,CLSTK+5
        LD DE,CLSTK
        LD BC,15
        LDIR
        LD HL,CLSTK+19
        XOR A
        LD B,5
CLX1:    LD (HL),A
        DEC HL
        DEC B
        JP NZ,CLX1
        RET

;***** CLX40 SUB *****
;OUT = (IX+0)
;REG = ABCDEHL

CLX40:   LD HL,CLSTK
        PUSH IX
        POP DE
        LD BC,5
        LDIR
        CALL CLX
        RET

;***** CLDV SUB *****
;REG = ABCDEHL

CLDV:    CALL CLCLB
        LD C,40
CLDV1:   CALL CLDVX
        CALL CLDVZ
        CALL CLDVS
        JP NC,CLDV2
        CALL CLDVA
CLDV2:   CCF
        CALL CLDVZ4
        DEC C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JP    NZ,CLDV1
        CALL CLEND
        RET
CLDVX:  XOR   A                ;ROTATE CLSTK4
        LD   HL,CLSTK+9
        LD   B,5
CLDVX1:  RL   (HL)
        DEC  HL
        DEC  B
        JP   NZ,CLDVX1
        RET  NC
        LD   HL,CLSTK+9
        SET  O,(HL)
        RET
CLDVZ:  LD   HL,CLBUF+4        ;ROTATE CLBUF
        LD   B,5
        JP   CLDVZ41
CLDVZ4:  LD   HL,CLBUF+9        ;ROTATE CLBUF4
        LD   B,5
CLDVZ41:  RL   (HL)
        DEC  HL
        DEC  B
        JP   NZ,CLDVZ41
        RET
CLDVS:  LD   HL,CLSTK+4        ;CLBUF=CLBUF-CLSTK
        LD   DE,CLBUF+4
        LD   B,5
CLDVS1:  LD   A,(DE)
        SBC  A,(HL)
        LD   (DE),A
        DEC  HL
        DEC  DE
        DEC  B
        JP   NZ,CLDVS1
        RET
CLDVA:  XOR   A                ;CLBUF=CLBUF+CLSTK
        LD   HL,CLSTK+4
        LD   DE,CLBUF+4
        LD   B,5
CLDVA1:  LD   A,(DE)
        ADC  A,(HL)
        LD   (DE),A
        DEC  HL
        DEC  DE
        DEC  B
        JP   NZ,CLDVA1
        RET

```

```

;***** CLHD SUB *****
;REG = ABCDEHL

```

```

CLHD:   CALL CLCLB
        LD   C,40
CLHD1:  LD   HL,CLSTK+4
        LD   B,5
CLHD2:  RL   (HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEC HL
DEC B
JP NZ,CLHD2
LD HL,CLBUF+9
LD B,10
CLHD3: LD A,(HL)
ADC A,A
DAA
LD (HL),A
DEC HL
DEC B
JP NZ,CLHD3
DEC C
JP NZ,CLHD1
CALL CLBTX
CALL CLFAG
RET

```

```

;***** CLDH SUB *****
;REG = ABCDEHL

```

```

CLDH: LD C,40
CLDH1: LD B,5
XOR A
LD HL,CLSTK
CLDH2: LD A,(HL)
RRA
PUSH AF
BIT 7,A
JP Z,CLDH3
SUB 30H
CLDH3: BIT 3,A
JP Z,CLDH4
SUB 3
CLDH4: LD (HL),A
INC HL
POP AF
DEC B
JP NZ,CLDH2
LD HL,CLBUF+5
LD B,5
CLDH5: RR (HL)
INC HL
DEC B
JP NZ,CLDH5
DEC C
JP NZ,CLDH1
CALL CLBTX
CALL CLFAG
RET

```

```

;***** CLSW SUB *****
;REG = BCDEHL

```

```

CLSW: LD HL,CLSTK+5
LD DE,CLBUF+5
LD BC,5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LDIR
CALL CLBTX
RET

;***** CLEND SUB ***** (LOCAL USE)
;REG = ABCDEHL

CLEND:   CALL CLX
         CALL CLBTX
         CALL CLFAG
         RET

;***** CLBTX SUB ***** (LOCAL USE)
;LOAD (CLBUF+4) TO (CLSTK)
;REG = BCDEHL

CLBTX:   LD   HL,CLBUF+5
         LD   DE,CLSTK
         LD   BC,5
         LDIR
         RET

;***** CLCLR & CLCLB SUB ***** (LOCAL USE)
;CLEAR (CLSTK) & (CLBUF)
;REG = ABCDEHL

CLCLB:   LD   HL,CLBUF
         LD   B,10
         JP   CLCLR1
CLCLR:   LD   HL,CLSTK
         LD   B,5
CLCLR1:  XOR  A
CLCLR2:  LD   (HL),A
         INC  HL
         DEC  B
         JP  NZ,CLCLR2
         RET

;***** CLFAG SUB ***** (LOCAL USE)
;REG = ABCHL
;SET CARRY & ZERO FLAG

CLFAG:   LD   C,00H
         LD   HL,CLBUF
         LD   B,5
CLFAG1:  LD   A,(HL)
         CP   00H
         JP  Z,CLFAG2
         SET 0,C
         JP  CLFAG3
CLFAG2:  INC  HL
         DEC  B
         JP  NZ,CLFAG1
CLFAG3:  LD   HL,CLSTK
         LD   B,5
CLFAG4:  LD   A,(HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP    00H
JP    NZ,CLFAG5
INC   HL
DEC   B
JP    NZ,CLFAG4
SET   7,C
CLFAG5:  PUSH BC
        POP  AF
        RET

```

```

; *****
; *   SUBROUTINE CHANGE TO ASCII *
; *****
;
; REG = ABDEHLIX
; IN  = B : (NUMBER OF ASCII) - 1
;      D : NUMBER OF BYTE FOR HEX VALUE

CH_ASCII: LD   A,(NUMBER)
           CP   05H                ;FOR CHANGE TO ASCII (ASCII_ODD)
           JP   NZ,CH_ACI6
CH_ACI5:  INC  HL
           DEC  B                  ;B = (NUMBER OF ASCII) - 1
           JP   NZ,CH_ACI5
ASC1:    LD   A,(IX+0)            ;IX = HEX VALUE
           LD   E,A
           AND  0FH
           ADD  A,30H
           LD   (HL),A           ;HL = ASCII VALUE
           DEC  D
           JP   Z,OKC
           INC  D
           LD   A,E
           AND  0FOH
           RR   A
           RR   A
           RR   A
           RR   A
           ADD  A,30H
           DEC  HL
           LD   (HL),A
           DEC  D
           JP   Z,OKC
           INC  IX
           DEC  HL
           JP   ASC1

;FOR CHANGE TO ASCII (ASCII_EVEN)
CH_ACI6:  INC  HL
           DEC  B                  ;B = (NUMBER OF ASCII) - 1
           JP   NZ,CH_ACI6
ASC2:    LD   A,(IX+0)            ;IX = HEX VALUE
           LD   E,A
           AND  0FH
           ADD  A,30H
           LD   (HL),A           ;HL = ASCII VALUE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,E
AND OFOH
RR A
RR A
RR A
RR A
RR A
ADD A,30H
DEC HL
LD (HL),A
DEC D
JP Z,OKC
INC IX
DEC HL
JP ASC2
OKC: RET

```

```

; *****
; * SUBROUTINE PASS (1000) *
; *****
;
; REG = ABCHL

```

```

PASS: LD BC,SWLED
LD A,OFH ;OFF LED PASS
OUT (C),A
CALL P_COUNT
CALL T_COUNT
CALL STA_BUF
CALL CK_ITVO
CALL CL_YDCM
CALL TFRDATA ;TRANSFER DATA
LD HL,OFFFH
LD (ADRS),HL
JP MAIN3

```

```

; *****
; * SUBROUTINE FAIL (0000) *
; *****
;
; REG = ABCHL

```

```

FAIL: LD BC,SWLED
LD A,OFH ;OFF LED FAIL
OUT (C),A
LD BC,PLED5
LD A,OOH ;OFF LED ALL SUB-FAIL
OUT (C),A
CALL F_COUNT
CALL T_COUNT
CALL STA_BUF
CALL CK_ITVO
CALL CL_YDCM
CALL TFRDATA ;TRANSFER DATA
LD HL,OFFFH
LD (ADRS),HL
JP MAIN3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ, ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; *****  
; * SUBROUTINE FAIL-A (0001) *  
; *****
```

```
FAIL_A: CALL A_H_CNT  
JP FAIL
```

```
; *****  
; * SUBROUTINE FAIL-B (0002) *  
; *****
```

```
FAIL_B: CALL B_H_CNT  
JP FAIL
```

```
; *****  
; * SUBROUTINE FAIL-C (0003) *  
; *****
```

```
FAIL_C: CALL C_H_CNT  
JP FAIL
```

```
; *****  
; * SUBROUTINE FAIL-D (0004) *  
; *****
```

```
FAIL_D: CALL D_H_CNT  
JP FAIL
```

```
; *****  
; * SUBROUTINE FAIL-E (0005) *  
; *****
```

```
FAIL_E: CALL E_H_CNT  
JP FAIL
```

```
; *****  
; * SUBROUTINE FAIL-F (0006) *  
; *****
```

```
FAIL_F: CALL F_H_CNT  
JP FAIL
```

```
; *****  
; * SUBROUTINE FAIL-G (0007) *  
; *****
```

```
FAIL_G: CALL G_H_CNT  
JP FAIL
```

```
; *****  
; * SUBROUTINE FAIL-H (0008) *  
; *****
```

```
FAIL_H: CALL H_H_CNT  
JP FAIL
```

```

; *****
; *   SUBROUTINE DELETE (1111) *
; *****
;
; REG = ABCHL

DELT_DEL: LD   BC,SWLED
          LD   A,OFH           ;OFF LED DELETE
          OUT  (C),A
          LD   BC,PLED5
          LD   A,OOH           ;OFF LED ALL SUB-FAIL
          OUT  (C),A
          CALL OLD_VLU
          CALL CK_ITVO
          CALL CL_YDCM
          CALL TFRDATA        ;TRANSFER DATA
          LD   HL,OFFFFH
          LD   (ADRS),HL
          JP   MAIN3

; *****
; *   SUBROUTINE
; *   DELETE-PASS (1110) *
; *****

DELT_P:  CALL CK_OLD
          CALL STA_BUF
          JP   DELT_DEL

; *****
; *   SUBROUTINE
; *   DELETE-FAIL (1100) *
; *****

DELT_F:  JP   DELT_P

; *****
; *   SUBROUTINE
; *   DELETE-FAIL_A (1101) *
; *****

DELT_FA: JP   DELT_P

; *****
; *   SUBROUTINE
; *   DELETE-FAIL_B (1102) *
; *****

DELT_FB: JP   DELT_P

; *****
; *   SUBROUTINE
; *   DELETE-FAIL_C (1103) *
; *****

```

DELT_FC: JP DELT_P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *          SUBROUTINE          *
; *    DELETE-FAIL_D (1104)      *
; *          *****

```

DELT_FD: JP DELT_P

```

; *****
; *          SUBROUTINE          *
; *    DELETE-FAIL_E (1105)      *
; *          *****

```

DELT_FE: JP DELT_P

```

; *****
; *          SUBROUTINE          *
; *    DELETE-FAIL_F (1106)      *
; *          *****

```

DELT_FF: JP DELT_P

```

; *****
; *          SUBROUTINE          *
; *    DELETE-FAIL_G (1107)      *
; *          *****

```

DELT_FG: JP DELT_P

```

; *****
; *          SUBROUTINE          *
; *    DELETE-FAIL_H (1108)      *
; *          *****

```

DELT_FH: JP DELT_P

```

; *****
; *    SERVICE INTERRUPT ROUTINE *
; *          *****
;
;
; REG = ABCDEHL

```

```

.ORG INTERRUPT
DI
PUSH AF
PUSH BC
PUSH DE
PUSH HL
LD HL,0FFFFH
LD A,(CODE_INT)
XOR 01H
LD (CODE_INT),A
PUSH AF
LD A,0BBH
LD (CODE_LCD),A
POP AF
CP 00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JP    Z,IN_SHO3
                CALL  SHOLCD1D
                JP    IN_DEL
IN_SHO3:      CALL  SHOLCD2D
IN_DEL:      CALL  DELAY
                DEC   HL
                LD    A,L
                CP    H
                JP    NZ,IN_DEL
                POP   HL
                POP   DE
                POP   BC
                POP   AF
                EI
                RETI

```

```

; *****
; *      TABLE DATA FOR LCD      *
; *****

```

```

LCD1B:      .DB    "< YEILDO METER >"
            .DB    " HOW MANY TARGET"
            .DB    " YEILD DO YOU  "
            .DB    " WANT ?      "
PERCENT:    .DB    "% "

```

```

LCD2B:      .DB    "*****"
            .DB    " INTERVAL VALUE "
            .DB    " "
            .DB    "*****"
EMPTY:      .DB    " "

```

```

LCD3B:      .DB    "TARGET YD.    . %"
            .DB    "YEILD ALR    . %"
            .DB    "P:          F:    "
            .DB    "TOTAL:         "

```

```

F1:         .DB    "F "
FA1:        .DB    "FA"
FB1:        .DB    "FB"
FC1:        .DB    "FC"
FD1:        .DB    "FD"
FE1:        .DB    "FE"
FF1:        .DB    "FF"
FG1:        .DB    "FG"
FH1:        .DB    "FH"
P1:         .DB    "P "

```

```

LCD0D:      .DB    " YEILDO METER  "
            .DB    "          BY          "
            .DB    "          KMIT'L      "
            .DB    " APPLY PHYSIC    "

```

```

LCD1D:      .DB    "A:          B:          "
            .DB    "C:          D:          "
            .DB    "E:          F:          "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        .DB "G:      H:      "
LCD2D:  .DB "TARGET YD.  ."
        .DB "YEILD COM  ."
        .DB "P:      F:      "
        .DB "TOTAL:      "

; *****
; *          RAM AREA          *
; *****

.ORG 8000H
DET_INT: .RS 2          ;DESTINATION OF INTERRUPT
STCODE:  .RS 1
CODE_INT: .RS 1        ;CODE INTERRUPT
ADRS:    .RS 2         ;ADDRESS
SAVE_TTL: .RS 3        ;SAVE TOTAL
SAVE_BUF: .RS 2        ;SAVE BUFFER
STATUS:   .RS 1        ;STATUS
CK_SHOW:  .RS 1        ;CHECK SHOW LCD
AD_BUF:   .RS 2        ;ADDRESS BUFFER
OLD_STA:  .RS 1        ;OLD STATUS
CODE_LCD: .RS 1        ;CODE LCD
NUMBER:   .RS 1        ;CHECK ASCII
CK_ITV:   .RS 1        ;CHECK INTERVAL

INTERVAL: .RS 2        ;HEX INTERVAL
INTV_P:   .RS 2        ;HEX PASS INTERVAL
INTV_F:   .RS 2        ;HEX FAIL INTERVAL
HEXPASS:  .RS 3        ;HEX PASS
HEXFAIL:  .RS 3        ;HEX FAIL
HEXTOTAL: .RS 3        ;HEX TOTAL
HEX_A:    .RS 3        ;HEX FAIL-A
HEX_B:    .RS 3        ;HEX FAIL-B
HEX_C:    .RS 3        ;HEX FAIL-C
HEX_D:    .RS 3        ;HEX FAIL-D
HEX_E:    .RS 3        ;HEX FAIL-E
HEX_F:    .RS 3        ;HEX FAIL-F
HEX_G:    .RS 3        ;HEX FAIL-G
HEX_H:    .RS 3        ;HEX FAIL-H
CLDEND:   .RS 5        ;FOR CALCULATE (DIVIDEND)
CLDSOR:   .RS 5        ;FOR CALCULATE (DIVISOR)
HEX_YC:   .RS 2        ;HEX YEILD COMMUTATIVE
HEX_YA:   .RS 2        ;HEX YEILD ALERT
HEX_TG:   .RS 2        ;HEX TARGET YEILD

ASC_ITV:  .RS 3        ;ASCII INTERVAL
ASC_PASS: .RS 5        ;ASCII PASS
ASC_FAIL: .RS 5        ;ASCII FAIL
ASC_A:    .RS 5        ;ASCII FAIL-A
ASC_B:    .RS 5        ;ASCII FAIL-B
ASC_C:    .RS 5        ;ASCII FAIL-C
ASC_D:    .RS 5        ;ASCII FAIL-D
ASC_E:    .RS 5        ;ASCII FAIL-E
ASC_F:    .RS 5        ;ASCII FAIL-F
ASC_G:    .RS 5        ;ASCII FAIL-G
ASC_H:    .RS 5        ;ASCII FAIL-H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ASC_TOTAL: .RS 6 ;ASCII TOTAL
ASC_YC: .RS 3 ;ASCII YEILD COMMUTATIVE
ASC_DC: .RS 1 ;ASCII DOT YEILD COMMUTATIVE
ASC_YA: .RS 3 ;ASCII YEILD ALERT
ASC_DA: .RS 1 ;ASCII DOT YEILD ALERT
ASC_TG: .RS 2 ;ASCII TARGET YEILD
ASC_DOT: .RS 1 ;ASCII DOT TARGET YEILD

CLSTK: .RS 20 ;STACK CALCULATE
CLBUF: .RS 10 ;BUFFER CALCULATE
BUFFER_ST: .RS OFFH ;BUFFER OF STATUS

ENDRAM:

```

.END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *      MONITOR PROGRAM      *
; *  FOR MAIN-BOARD OF YEILDO METER  *
; *    BY Z80 CP-A CONTROL PACK  *
; *  CROSS - ASSEMBLER :  A180Z.EXE  *
; *                               *
; *      VERSION 1.0          *
; *    DATE WRITE : JANUARY 1992  *
; *    WRITEN BY KMIT'L SOFTWARE  *
; *    SCIENCE PHYSICS - SOLIDY (07) *
; *      : THANAPAT YONGPIPATWONG  *
; *      : BUNSONG CHARUNWORAPAN  *
; *****

```

.ORG 0000H

```

; *****
; *      VARIABLE SET        *
; *****

```

```

.EQU STCODE,3AH      ;CODE FOR START
.EQU SEG1,34H        ;PA 8255 (I/O 1-1)
.EQU SEG2,35H        ;PB 8255 (I/O 1-1)
.EQU SEG3,36H        ;PC 8255 (I/O 1-1)
.EQU CONT1_1,37H     ;P-CONTROL (I/O 1-1)
.EQU SEG4,38H        ;PA 8255 (I/O 1-2)
.EQU SEG5,39H        ;PB 8255 (I/O 1-2)
.EQU SEG6,3AH        ;PC 8255 (I/O 1-2)
.EQU CONT1_2,3BH     ;P-CONTROL (I/O 1-2)
.EQU SEG7,3CH        ;PA 8255 (I/O 1-3)
.EQU SEG8,3DH        ;PB 8255 (I/O 1-3)
.EQU SEG9,3EH        ;PC 8255 (I/O 1-3)
.EQU CONT1_3,3FH     ;P-CONTROL (I/O 1-3)
.EQU SEG10,54H       ;PA 8255 (I/O 2-1)
.EQU SEG11,55H       ;PB 8255 (I/O 2-1)
.EQU SEG12,56H       ;PC 8255 (I/O 2-1)
.EQU CONT2_1,57H     ;P-CONTROL (I/O 2-1)
.EQU SEG13,58H       ;PA 8255 (I/O 2-2)
.EQU SEG14,59H       ;PB 8255 (I/O 2-2)
.EQU SEG15,5AH       ;PC 8255 (I/O 2-2)
.EQU CONT2_2,5BH     ;P-CONTROL (I/O 2-2)
.EQU SEG16,5CH       ;PA 8255 (I/O 2-3)
.EQU SEG17,5DH       ;PB 8255 (I/O 2-3)
.EQU SEG18,5EH       ;PC 8255 (I/O 2-3)
.EQU CONT2_3,5FH     ;P-CONTROL (I/O 2-3)
.EQU SEG19,74H       ;PA 8255 (I/O 3-1)
.EQU SEG20,75H       ;PB 8255 (I/O 3-1)
.EQU SEG21,76H       ;PC 8255 (I/O 3-1)
.EQU CONT3_1,77H     ;P-CONTROL (I/O 3-1)
.EQU SEG22,78H       ;PA 8255 (I/O 3-2)
.EQU KEY,79H         ;PB 8255 (I/O 3-2)
.EQU KEYCK,7AH       ;PC 8255 (I/O 3-3)
.EQU CONT3_2,7BH     ;P-CONTROL (I/O 3-2)
.EQU RF_RTS,7CH      ;PA 8255 (CPA)
.EQU ST_CTS,7DH      ;PB 8255 (CPA)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.EQU RS_DATA,7EH           ;PC 8255 (CPA)
.EQU CPA_CONT,7FH         ;P-CONTROL (CPA)
.EQU SYSSTK,2F00H        ;START SYSTEM STACK

; *****
; *           POWER-UP           *
; *****

POWER:   XOR   A                ;POWER-UP DELAY
POWER1:  DEC   A
        NOP
        JP   NZ,POWER1
        JP   RESET

; *****
; *           ROUTINE OF NMI      *
; *   FOR CLEAR ALL DATA IN RAM *
; *****

.ORG 0066H
LD   HL,ST_CODE
LD   (HL),00H
JP   0000H

; *****
; *           RESET SYSTEM       *
; *****

RESET:   .ORG 100H
        LD   HL,4000H           ;DELAY
RESETO:  DEC   HL
        LD   A,H
        OR   L
        JP   NZ,RESETO

; *****
; *           START SYSTEM       *
; *****

START:   LD   SP,SYSSTK        ;SYSTEM STACK
        LD   A,80H            ;SET P-CONTROL 8255
        LD   C,CONT1_1        ;PA-PC IS OUTPUT PORT
        OUT  (C),A            ;I/O 1-1
        LD   C,CONT1_2
        OUT  (C),A            ;I/O 1-2
        LD   C,CONT1_3
        OUT  (C),A            ;I/O 1-3
        LD   C,CONT2_1
        OUT  (C),A            ;I/O 2-1
        LD   C,CONT2_2
        OUT  (C),A            ;I/O 2-2
        LD   C,CONT2_3
        OUT  (C),A            ;I/O 2-3
        LD   C,CONT3_1
        OUT  (C),A            ;I/O 3-1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,8BH ;SET P-CONTROL 8255
LD C,CONT3_2 ;PA OUTPUT; PB,PC INPUT
OUT (C),A ;I/O 3-2
LD A,99H ;SET P-CONTROL 8255-CPA
LD C,CPA_CONT ;PA,PC INPUT; PB OUTPUT
OUT (C),A
LD A,OFFH
LD C,ST_CTS
OUT (C),A

```

```

; *****
; * CHECK START CODE *
; *****

```

```

CK_CODE: LD A,(ST_CODE)
CP STCODE
JP Z,SET_CODE

```

```

; *****
; * READ NUMBER OF SUB-BOARD *
; *****

```

```

LD C,KEYCK
IN A,(C)
LD (CDSUB),A

```

```

; *****
; * READ TYPE OF CALCULATE *
; *****

```

```

LD C,KEY
IN A,(C)
LD HL,CAL_TY
BIT 3,A
JP NZ,CAL_TY1
LD (HL),01H ;FOR TYPE 1
JP SET_TG

```

```

CAL_TY1: LD (HL),02H ;FOR TYPE 2

```

```

; *****
; * SET FIRST TARGET YEILD *
; *****

```

```

SET_TG: LD HL,HEX_TG
LD (HL),08H
INC HL
LD (HL),50H

```

```

; *****
; * SET FIRST VALUE *
; *****

```

```

LD HL,BUADLY
LD (HL),5FH
INC HL
LD BC,256

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SET1:    LD    (HL),00H
         INC  HL
         DEC  BC
         LD   A,B
         OR   C
         JP   NZ,SET1
         LD   (HL),01H
         INC  HL
         LD   (HL),00H
         INC  HL
         LD   (HL),00H

SET_CODE: LD  A,STCODE
          LD  (ST_CODE),A

; *****
; *      MAIN PROGRAM      *
; *****

; ***** SET TARGET YEILD *****

MAIN:    JP   SHOW_SEG           ;SHOW 7-SEGMENT DISPLAY
MAIN1:   LD   IX,HEX_TG
SKEY1:   XOR  A                   ;CLEAR REG.A
         LD   C,KEY
         IN   A,(C)               ;CHECK KEY FOR TG.YEILD
         BIT  0,A
         JP   Z,TG_INC            ;IF INCREMENT TG.YEILD
         BIT  1,A
         JP   Z,TG_DEC            ;IF DECREMENT TG.YEILD

; ***** CHECK FOR RECEIVE *****

LD   C,RF_RTS
IN   A,(C)
LD   (BUF_RVS),A
RVS:   LD   A,(BUF_RVS)
       BIT  0,A                   ;OF SUB-BOARD 1
       JP   Z,RVS_SUB1
       BIT  1,A                   ;OF SUB-BOARD 2
       JP   Z,RVS_SUB2
       BIT  2,A                   ;OF SUB-BOARD 3
       JP   Z,RVS_SUB3
       BIT  3,A                   ;OF SUB-BOARD 4
       JP   Z,RVS_SUB4
       BIT  4,A                   ;OF SUB-BOARD 5
       JP   Z,RVS_SUB5
       BIT  5,A                   ;OF SUB-BOARD 6
       JP   Z,RVS_SUB6
       BIT  6,A                   ;OF SUB-BOARD 7
       JP   Z,RVS_SUB7
       BIT  7,A                   ;OF SUB-BOARD 8
       JP   Z,RVS_SUB8

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** CHECK FOR CALCULATE *****

MAIN2:   JP    VLCAL
        XOR   A                               ;CLEAR CARRY FLAG
        LD   A,(CAL_TY)
        SUB  02H
        JP   NC,CAL2
        JP   CAL1

; *****
; *      SUBROUTINE INCREMENT      *
; *      TARGET YEILD             *
; *****
;
; REG = AIXIY

TG_INC:  LD   A,(IX+0)                       ;CHECK VALUE OF TG.YEILD
        AND  0FOH
        CP   10H
        JP   Z,MAIN

        XOR  A
        LD   A,(IX+1)
        ADD  A,01H
        DAA
        LD   (IX+1),A
        LD   A,(IX+0)
        ADC  A,00H
        DAA
        LD   (IX+0),A
        JP   MAIN

; *****
; *      SUBROUTINE DECREMENT     *
; *      TARGET YEILD           *
; *****
;
; REG = AIXIY

TG_DEC:  LD   A,(IX+0)                       ;CHECK VALUE OF TG.YEILD
        CP   00H
        JP   NZ,TG_DE1
        LD   A,(IX+1)
        CP   00H
        JP   Z,MAIN

TG_DE1:  XOR  A
        LD   A,(IX+1)
        SUB  01H
        DAA
        LD   (IX+1),A
        LD   A,(IX+0)
        SBC  A,00H
        DAA
        LD   (IX+0),A
        JP   MAIN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *          SUBROUTINE          *
; *    CHECK KEY PRESSED 1    *
; *****
;
; REG = ABCDE

KEYPRS1:  JP   DELAY10
PRSO:     LD   E,OFFH           ;DELAY IN SUBROUTINE
PRS1:     LD   C,KEY
          LD   D,02H           ;COUNT BIT ,
          XOR  A               ;CLEAR REG.A
          IN  A,(C)
SH_BIT1:  BIT  0,A             ;CHECK PRESSION
          JP  Z,PRESS1         ;IF PRESSED SWITCH
          RR  A
          DEC  D
          JP  NZ,SH_BIT1
          JP  NO_PRS1          ;IF NO PRESSED SWITCH
PRESS1:   DEC  E
          JP  NZ,PRS1
NO_PRS1:  JP   DELAY

; *****
; *          SUBROUTINE DELAY    *
; *****
;
; REG = BC
DELAY:    PUSH BC
          LD   B,00H
DE1:      NOP
          NOP
          DJNZ DE1
          POP  BC
          JP  MAIN1

; *****
; *          SUBROUTINE DELAY 10 ms *
; *****
;
; REG = ABC

DELAY10:  PUSH AF
          PUSH BC
          LD   BC,05F00H
DE10:     DEC  BC
          LD   A,B
          OR   C
          JP  NZ,DE10
          POP  BC
          POP  AF
          JP  PRSO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; * SUBROUTINE FOR RECEIVE DATA *
; *****

```

```

RVS_SUB1: LD IX,RVPASS1
          XOR A
          LD (IX+0),A
          LD (IX+1),A
          LD (IX+2),A
          LD (IX+3),A
          LD (IX+4),A
          LD (IX+5),A
          LD (IX+6),A
          LD (IX+7),A
          LD (IX+8),A
          LD HL,RVPASS1
          LD E,09H
          LD C,ST_CTS
          LD A,0FEH
          OUT (C),A
          PUSH BC
RVS_B11: IN A,(RS_DATA)
          BIT 0,A
          JP NZ,RVS_B11

          CALL PCLDLY

          LD D,00H
          LD B,08H
RVS_B12: XOR A
          IN A,(RS_DATA)
          RRA
          RR D
          CALL PCLDLY
          DJNZ RVS_B12
          LD (HL),D
          INC HL
          DEC E
          JP NZ,RVS_B11
          LD A,(BUF_RVS)
          SET 0,A
          LD (BUF_RVS),A
          POP BC
          LD A,OFFH
          OUT (C),A
          JP RVS

RVS_SUB2: LD IX,RVPASS2
          XOR A
          LD (IX+0),A
          LD (IX+1),A
          LD (IX+2),A
          LD (IX+3),A
          LD (IX+4),A
          LD (IX+5),A
          LD (IX+6),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (IX+7),A
LD (IX+8),A
LD HL,RVPASS2
LD E,09H
LD C,ST_CTS
LD A,OFDH
OUT (C),A
PUSH BC
RVSB21: XOR A
IN A,(RS_DATA)
BIT 1,A
JP NZ,RVSB21

```

```
CALL PCLDLY
```

```

LD D,00H
LD B,08H
RVSB22: IN A,(RS_DATA)
RRA
RRA
RR D
CALL PCLDLY
DJNZ RVSB22
LD (HL),D
INC HL
DEC E
JP NZ,RVSB21
LD A,(BUF_RVS)
SET 1,A
LD (BUF_RVS),A
POP BC
LD A,OFFH
OUT (C),A
JP RVS

```

```

RVS_SUB3: LD IX,RVPASS3
XOR A
LD (IX+0),A
LD (IX+1),A
LD (IX+2),A
LD (IX+3),A
LD (IX+4),A
LD (IX+5),A
LD (IX+6),A
LD (IX+7),A
LD (IX+8),A
LD HL,RVPASS3
LD E,09H
LD C,ST_CTS
LD A,OFBH
OUT (C),A
PUSH BC
RVSB31: XOR A
IN A,(RS_DATA)
BIT 2,A
JP NZ,RVSB31

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL PCLDLY

LD D,00H
LD B,08H
RVSB32: IN A,(RS_DATA)
RRA
RRA
RRA
RR D
CALL PCLDLY
DJNZ RVSB32
LD (HL),D
INC HL
DEC E
JP NZ,RVSB31
LD A,(BUF_RVS)
SET 2,A
LD (BUF_RVS),A
POP BC
LD A,OFFH
OUT (C),A
JP RVS

RVB_SUB4: LD IX,RVPASS4
XOR A
LD (IX+0),A
LD (IX+1),A
LD (IX+2),A
LD (IX+3),A
LD (IX+4),A
LD (IX+5),A
LD (IX+6),A
LD (IX+7),A
LD (IX+8),A
LD HL,RVPASS4
LD E,09H
LD C,ST_CTS
LD A,OF7H
OUT (C),A
PUSH BC
RVSB41: XOR A
IN A,(RS_DATA)
BIT 3,A
JP NZ,RVSB41

```

```
CALL PCLDLY
```

```

LD D,00H
LD B,08H
RVSB42: IN A,(RS_DATA)
RRA
RRA
RRA
RRA
RR D

```

```
CALL PCLDLY
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ถ้าทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DJNZ RVSB42
LD (HL),D
INC HL
DEC E
JP NZ,RVSB41
LD A,(BUF_RVS)
SET 3,A
LD (BUF_RVS),A
POP BC
LD A,OFFH
OUT (C),A
JP RVS

```

```

RVS_SUB5: LD IX,RVPASS5
XOR A
LD (IX+0),A
LD (IX+1),A
LD (IX+2),A
LD (IX+3),A
LD (IX+4),A
LD (IX+5),A
LD (IX+6),A
LD (IX+7),A
LD (IX+8),A
LD HL,RVPASS5
LD E,09H
LD C,ST_CTS
LD A,OEFH
OUT (C),A
PUSH BC

```

```

RVSB51: XOR A
IN A,(RS_DATA)
BIT 4,A
JP NZ,RVSB51
CALL PCLDLY

```

```

LD D,00H
LD B,08H
RVSB52: IN A,(RS_DATA)
RLA
RLA
RLA
RLA
RR D
CALL PCLDLY
DJNZ RVSB52
LD (HL),D
INC HL
DEC E
JP NZ,RVSB51
LD A,(BUF_RVS)
SET 4,A
LD (BUF_RVS),A
POP BC
LD A,OFFH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUT (C),A
JP RVS

RVS_SUB6: LD IX,RVPASS6
XOR A
LD (IX+0),A
LD (IX+1),A
LD (IX+2),A
LD (IX+3),A
LD (IX+4),A
LD (IX+5),A
LD (IX+6),A
LD (IX+7),A
LD (IX+8),A
LD HL,RVPASS6
LD E,09H
LD C,ST_CTS
LD A,ODFH
OUT (C),A
PUSH BC
RVS61: XOR A
IN A,(RS_DATA)
BIT 5,A
JP NZ,RVS61
CALL PCLDLY
LD D,00H
LD B,08H
RVS62: IN A,(RS_DATA)
RLA
RLA
RLA
RR D
CALL PCLDLY
DJNZ RVS62
LD (HL),D
INC HL
DEC E
JP NZ,RVS61
LD A,(BUF_RVS)
SET 5,A
LD (BUF_RVS),A
POP BC
LD A,OFFH
OUT (C),A
JP RVS

RVS_SUB7: LD IX,RVPASS7
XOR A
LD (IX+0),A
LD (IX+1),A
LD (IX+2),A
LD (IX+3),A
LD (IX+4),A
LD (IX+5),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (IX+6),A
LD (IX+7),A
LD (IX+8),A
LD HL,RVPASS7
LD E,09H
LD C,ST_CTS
LD A,0BFH
OUT (C),A
PUSH BC
RVS71: XOR A
IN A,(RS_DATA)
BIT 6,A
JP NZ,RVS71

```

```

CALL PCLDLY
LD D,00H
LD B,08H
RVS72: IN A,(RS_DATA)
RLA
RLA
RR D
CALL PCLDLY
DJNZ RVS72
LD (HL),D
INC HL
DEC E
JP NZ,RVS71
LD A,(BUF_RVS)
SET 6,A
LD (BUF_RVS),A
POP BC
LD A,OFFH
OUT (C),A
JP RVS

```

```

RVS_SUB8: LD IX,RVPASS8
XOR A
LD (IX+0),A
LD (IX+1),A
LD (IX+2),A
LD (IX+3),A
LD (IX+4),A
LD (IX+5),A
LD (IX+6),A
LD (IX+7),A
LD (IX+8),A
LD HL,RVPASS8
LD E,09H
LD C,ST_CTS
LD A,7FH
OUT (C),A
PUSH BC

```

```

RVS81: XOR A
IN A,(RS_DATA)
BIT 7,A

```

```

JP NZ,RVSB81

CALL PCLDLY

PUSH BC
LD D,00H
LD B,08H
RVSB82: IN A,(RS_DATA)
RLA
RR D
CALL PCLDLY
DJNZ RVSB82
LD (HL),D
INC HL
DEC E
JP NZ,RVSB81
LD A,(BUF_RVS)
SET 7,A
LD (BUF_RVS),A
POP BC
LD A,OFFH
OUT (C),A
JP RVS

; ***** ONE BIT SPEED DELAY *****

PCLDLY: LD A,(BUADLY)
PCLDLY1: DEC A
JR NZ,PCLDLY1
RET

; *****
; * SUBROUTINE VALUE-CALCULATE *
; *****
;
; REG = IX,IY

VLCAL: LD IX,RVPASS1
LD IY,PASS1
CALL EXVAL
LD IX,RVPASS2
LD IY,PASS2
CALL EXVAL
LD IX,RVPASS3
LD IY,PASS3
CALL EXVAL
LD IX,RVPASS4
LD IY,PASS4
CALL EXVAL
LD IX,RVPASS5
LD IY,PASS5
CALL EXVAL
LD IX,RVPASS6
LD IY,PASS6
CALL EXVAL
LD IX,RVPASS7

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD IY,PASS7
CALL EXVAL
LD IX,RVPASS8
LD IY,PASS8
CALL EXVAL

LD IX,RVFAIL1
LD IY,FAIL1
CALL EXVAL
LD IX,RVFAIL2
LD IY,FAIL2
CALL EXVAL
LD IX,RVFAIL3
LD IY,FAIL3
CALL EXVAL
LD IX,RVFAIL4
LD IY,FAIL4
CALL EXVAL
LD IX,RVFAIL5
LD IY,FAIL5
CALL EXVAL
LD IX,RVFAIL6
LD IY,FAIL6
CALL EXVAL
LD IX,RVFAIL7
LD IY,FAIL7
CALL EXVAL
LD IX,RVFAIL8
LD IY,FAIL8
CALL EXVAL

LD IX,RVTOTAL1
LD IY,TOTAL1
CALL EXVAL
LD IX,RVTOTAL2
LD IY,TOTAL2
CALL EXVAL
LD IX,RVTOTAL3
LD IY,TOTAL3
CALL EXVAL
LD IX,RVTOTAL4
LD IY,TOTAL4
CALL EXVAL
LD IX,RVTOTAL5
LD IY,TOTAL5
CALL EXVAL
LD IX,RVTOTAL6
LD IY,TOTAL6
CALL EXVAL
LD IX,RVTOTAL7
LD IY,TOTAL7
CALL EXVAL
LD IX,RVTOTAL8
LD IY,TOTAL8
CALL EXVAL
JP MAIN2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *****
; *   SUBROUTINE EXCHANGE VALUE   *
; *****
; IN  = IX:SOURCE, IY:TARGET
; REG = AB

```

```

EXVAL:  LD  B,03H
EXVAL1: LD  A,(IX+0)
        LD  (IY+4),A
        INC IX
        DEC IY
        DEC B
        JP  NZ,EXVAL1
        RET

```

```

; *****
; *   SUBROUTINE CALCULATE        *
; *****
;
; REG = ABCDEHLIX

```

```

CAL1:   LD  HL,TOTAL1
        CALL DVDEND
        CALL CLE40
        CALL CLDH
        LD  HL,TOTAL2
        CALL OPRATE
        CALL CLE40
        CALL CLDH
        CALL CLAD
        LD  HL,TOTAL3
        CALL OPRATE
        CALL CLE40
        CALL CLDH
        CALL CLAD
        LD  HL,TOTAL4
        CALL OPRATE
        CALL CLE40
        CALL CLDH
        CALL CLAD
        LD  HL,TOTAL5
        CALL OPRATE
        CALL CLE40
        CALL CLDH
        CALL CLAD
        LD  HL,TOTAL6
        CALL OPRATE
        CALL CLE40
        CALL CLDH
        CALL CLAD
        LD  HL,TOTAL7
        CALL OPRATE
        CALL CLE40
        CALL CLDH
        CALL CLAD
        LD  HL,TOTAL8

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
CALL CLHD
LD DE, TTALTL
CALL SAVE_VAL
CALL CLX40

;SAVE TOTAL-TOTAL IN TTALTL

LD HL, PASS1
CALL DVDEND
CALL CLE40
CALL CLDH
LD HL, PASS2

CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD

LD HL, PASS3
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD

LD HL, PASS4
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD

LD HL, PASS5
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD

LD HL, PASS6
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD

LD HL, PASS7
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD

LD HL, PASS8
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD

LD DE, PASSTL
CALL SAVE_VAL
CALL CLX40

;SAVE PASS-TOTAL IN PASSTL

LD HL, PASSTL
CALL OPRATE
CALL CLE40

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL CLDH
LD HL,TTALTL
CALL DVDEND
CALL CLE40
CALL CLDH
CALL CLSB ;TTALTL - PASSTL = FAILTL
CALL CLHD
LD DE,FAILTL
CALL SAVE_VAL ;SAVE FAIL-TOTAL IN FAILTL
CALL CLX40 ;CLEAR STACK CALCULATE

LD IX,PASSTL
LD A,(IX+2)
CP 00H
JP NZ,CAL11
LD A,(IX+3)
CP 00H
JP NZ,CAL11
LD A,(IX+4)
CP 00H
JP NZ,CAL11
LD BC,0000H
LD (HEX_YC),BC ;SHOW YEILD COM. = 000.0%
JP MAIN
CAL11: LD HL,PASSTL
CALL DVDEND
CALL CLE40
CALL CLDH
LD HL,PERCENT
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLMP ;PASSTL X PERCENT =DIVIDEND

LD HL,TTALTL ;TTALTL = DIVISOR
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLDV
CALL CLHD
CALL CLX40
JP EXC_SAVE ;SAVE RESULT FOR SHOW
;SAVE TO HEX_YC

```

; ***** CALCULATE FOR TYPE 2 *****

```

CAL2: LD HL,FAIL1
CALL DVDEND
CALL CLE40
CALL CLDH
LD HL,FAIL2
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
LD HL,FAIL3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
LD HL, FAIL4
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
LD HL, FAIL5
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
LD HL, FAIL6
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
LD HL, FAIL7
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
LD HL, FAIL8
CALL OPRATE
CALL CLE40
CALL CLDH
CALL CLAD
LD DE, FAILTL
CALL SAVE_VAL ;SAVE FAIL-TOTAL IN FAILTL
CALL CLX40

LD HL, FAILTL
CALL OPRATE
CALL CLE40
CALL CLDH
LD HL, TOTAL1
CALL DVDEND
CALL CLE40
CALL CLDH
CALL CLSB ;TOTAL1 - FAILTL = TMPPASS
CALL CLHD
LD DE, TMPPASS
CALL SAVE_VAL
CALL CLX40

LD IX, TMPPASS
LD A, (IX+2)
CP 00H
JP NZ, CAL21
LD A, (IX+3)
CP 00H
JP NZ, CAL21
LD A, (IX+4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อี-ทัชทั้งหมดให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP      OOH
JP      NZ,CAL21
LD      BC,0000H
LD      (HEX_YC),BC      ;SHOW YEILD COM. = 000.0%
JP      MAIN
CAL21:  LD      HL,TMPPASS
        CALL   DVDEND
        CALL   CLE40
        CALL   CLDH
        LD      HL,PERCENT
        CALL   OPRATE
        CALL   CLE40
        CALL   CLDH
        CALL   CLMP      ;TMPPASS X PERCENT = DIVIDEND

        LD      HL,TOTAL1      ;TOTAL1 = DIVISOR
        CALL   OPRATE
        CALL   CLE40
        CALL   CLDH
        CALL   CLDV
        CALL   CLHD
        CALL   CLX40
        JP      EXC_SAVE

; *****
; *      SUBROUTINE EXCHANGE      *
; *      FOR CALCURATE          *
; *****
;
; REG = BCDEIX
DVDEND:  PUSH   DE
        PUSH   BC
        LD     DE,DEND
        LD     BC,05H
        LDIR
        LD     IX,DEND
        POP   BC
        POP   DE
        RET

OPRATE:  PUSH   DE
        PUSH   BC
        LD     DE,RATE
        LD     BC,05H
        LDIR
        LD     IX,RATE
        POP   BC
        POP   DE
        RET

```

```

; *****
; *      SUBROUTINE SAVE VALUE    *
; *****
; IN = DE
; REG = BCDE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SAVE_VAL: LD HL,CLSTK
          LD BC,05H
          LDIR
          RET

```

```

; *****
; * SUBROUTINE EXCHANGE DATA *
; *****
;
; REG = ABCIXIY

```

```

EXC_SAVE: LD IY,HEX_YC
          LD IX,CLBUF
          LD A,(IX+9)
          AND 0FH
          CP 06H
          JP C,CL_ENDC1
          XOR A ;CLEAR CARRY FLAG
          LD A,(IX+9)
          ADD A,10H
          DAA
          LD (IX+9),A
          LD A,(IX+8)
          ADC A,00H
          DAA
          LD (IX+8),A
          LD A,(IX+7)
          ADC A,00H
          DAA
          LD (IX+7),A
CL_ENDC1: LD C,02H
CL_ENDC2: LD A,(IX+9)
          AND 0FOH
          SRL A
          SRL A
          SRL A
          SRL A
          LD B,A
          LD A,(IX+8)
          AND 0FH
          SLA A
          SLA A
          SLA A
          SLA A
          ADD A,B
          LD (IY+1),A
          DEC IX
          DEC IY
          DEC C
          JP NZ,CL_ENDC2
          JP MAIN

```

```

;***** CLE SUB *****
;REG = BCDEHLIX
CLE: LD HL,CLSTK+14

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD DE,CLSTK+19
LD BC,15
LDDR
RET
```

```
;***** CLE40 SUB *****
;IN = (IX+0)
;REG = BCDEHL
CLE40: CALL CLE
      PUSH IX
      POP HL
      LD DE,CLSTK
      LD BC,5
      LDIR
      RET
```

```
;***** CLX SUB *****
;REG = BCDEHL
CLX: LD HL,CLSTK+5
     LD DE,CLSTK
     LD BC,15
     LDIR
     LD HL,CLSTK+19
     XOR A
     LD B,5
CLX1: LD (HL),A
     DEC HL
     DEC B
     JP NZ,CLX1
     RET
```

```
;***** CLX40 SUB *****
;OUT = (IX+0)
;REG = ABCDEHL
CLX40: LD HL,CLSTK
      PUSH IX
      POP DE
      LD BC,5
      LDIR
      CALL CLX
      RET
```

```
;***** CLMP SUB *****
;REG = ABCDEHL
CLMP: CALL CLCLB
     LD B,40
CLMP1: PUSH BC
      CALL CLMPS
      CALL CLMPR
      JP NC,CLMP2
      CALL CLMPA
CLMP2: POP BC
     DEC B
     JP NZ,CLMP1
     CALL CLEND
     RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLMPS:   XOR  A           ;SHIFT RESULT (CLBUF)
         LD  HL,CLBUF+9
         LD  B,10
CLMPS1:  RL   (HL)
         DEC HL
         DEC B
         JP  NZ,CLMPS1
         RET

CLMPR:   XOR  A           ;ROTATE MULTIPLICAND (CLSTK+4)
         LD  HL,CLSTK+9
         LD  B,5
CLMPR1:  RL   (HL)
         DEC HL
         DEC B
         JP  NZ,CLMPR1
         RET  NC
         LD  HL,CLSTK+9
         SET 0,(HL)
         RET

CLMPA:   XOR  A           ;ADD CLSTK TO CLBUF
         LD  HL,CLSTK+4
         LD  DE,CLBUF+9
         LD  B,5
CLMPA1:  LD  A,(DE)
         ADC A,(HL)
         LD  (DE),A
         DEC HL
         DEC DE
         DEC B
         JP  NZ,CLMPA1
         RET  NC
         LD  BC,500H
CLMPA2:  LD  A,(DE)
         ADC A,C
         LD  (DE),A
         DEC DE
         DEC B
         JP  NZ,CLMPA2
         RET

```

```

;***** CLDV SUB *****
;REG = ABCDEHL
CLDV:   CALL CLCLB
         LD  C,40
CLDV1:  CALL CLDVX
         CALL CLDVZ
         CALL CLDVS
         JP  NC,CLDV2
         CALL CLDVA
CLDV2:  CCF
         CALL CLDVZ4
         DEC C
         JP  NZ,CLDV1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL CLEND
RET
CLDVX: XOR A ;ROTATE CLSTK4
LD HL,CLSTK+9
LD B,5
CLDVX1: RL (HL)
DEC HL
DEC B
JP NZ,CLDVX1
RET NC
LD HL,CLSTK+9
SET 0,(HL)
RET
CLDVZ: LD HL,CLBUF+4 ;ROTATE CLBUF (3)
LD B,5
JP CLDVZ41
CLDVZ4: LD HL,CLBUF+9 ;ROTATE CLBUF4 (7)
LD B,5
CLDVZ41: RL (HL)
DEC HL
DEC B
JP NZ,CLDVZ41
RET
CLDVS: LD HL,CLSTK+4 ;CLBUF=CLBUF-CLSTK (3)
LD DE,CLBUF+4
LD B,5
CLDVS1: LD A,(DE)
SBC A,(HL)
LD (DE),A
DEC HL
DEC DE
DEC B
JP NZ,CLDVS1
RET
CLDVA: XOR A ;CLBUF=CLBUF+CLSTK
LD HL,CLSTK+4
LD DE,CLBUF+4
LD B,5
CLDVA1: LD A,(DE)
ADC A,(HL)
LD (DE),A
DEC HL
DEC DE
DEC B
JP NZ,CLDVA1
RET

```

```

;***** CLAD SUB *****

```

```

;REG = ABCDEHL

```

```

CLAD: PUSH IX
CALL CLCLB
LD HL,CLSTK+9
LD DE,CLSTK+4
LD IX,CLBUF+9
LD B,5

```

เอกสารนี้เป็น CLAD1: ที่สงวนลิขสิทธิ์ไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADC  A,(HL)
LD   (IX+0),A
DEC  HL
DEC  DE
DEC  IX
DEC  B
JP   NZ,CLAD1
LD   A,00H
ADC  A,A
LD   (IX+0),A
POP  IX
CALL CLEND
RET

```

***** CLSB SUB *****

```

;REG = ABCDEHL
CLSB:  PUSH  IX
        CALL CLCLB
        LD   HL,CLSTK+9
        LD   DE,CLSTK+4
        LD   IX,CLBUF+9
        LD   B,5
CLSB1: LD   A,(DE)
        SBC  A,(HL)
        LD   (IX+0),A
        DEC  HL
        DEC  DE
        DEC  IX
        DEC  B
        JP   NZ,CLSB1
        LD   A,0
        ADC  A,A
        LD   (IX+0),A
        POP  IX
        CALL CLEND
        RET

```

***** CLHD SUB *****

```

;REG = ABCDEHL
CLHD:  CALL CLCLB
        LD   C,40
CLHD1: LD   HL,CLSTK+4
        LD   B,5
CLHD2: RL   (HL)
        DEC  HL
        DEC  B
        JP   NZ,CLHD2
        LD   HL,CLBUF+9
        LD   B,10
CLHD3: LD   A,(HL)
        ADC  A,A
        DAA
        LD   (HL),A
        DEC  HL
        DEC  B
        JP   NZ,CLHD3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DEC C
JP NZ,CLHD1
CALL CLBTX
CALL CLFAG
RET
```

```
;***** CLDH SUB *****
```

```
;REG = ABCDEHL
```

```
CLDH: LD C,40
CLDH1: LD B,5
      XOR A
      LD HL,CLSTK
CLDH2: LD A,(HL)
      RRA
      PUSH AF
      BIT 7,A
      JP Z,CLDH3
      SUB 30H
CLDH3: BIT 3,A
      JP Z,CLDH4
      SUB 3
CLDH4: LD (HL),A
      INC HL
      POP AF
      DEC B
      JP NZ,CLDH2
      LD HL,CLBUF+5
      LD B,5
CLDH5: RR (HL)
      INC HL
      DEC B
      JP NZ,CLDH5
      DEC C
      JP NZ,CLDH1
      CALL CLBTX
      CALL CLFAG
      RET
```

```
;***** CLSW SUB *****
```

```
;REG = BCDEHL
```

```
CLSW: LD HL,CLSTK+5
      LD DE,CLBUF+5
      LD BC,5
      LDIR
      CALL CLBTX
      RET
```

```
;***** CLEND SUB ***** (LOCAL USE)
```

```
;REG = ABCDEHL
```

```
CLEND: CALL CLX
      CALL CLBTX
      CALL CLFAG
      RET
```

```
;***** CLBTX SUB ***** (LOCAL USE)
```

```
;LOAD (CLBUF+4) TO (CLSTK)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;REG = BCDEHL
CLBTX: LD HL,CLBUF+5
        LD DE,CLSTK
        LD BC,5
        LDIR
        RET

```

```

;***** CLCLR & CLCLB SUB ***** (LOCAL USE)

```

```

;CLEAR (CLSTK) & (CLBUF)

```

```

;REG = ABCDEHL

```

```

CLCLB: LD HL,CLBUF

```

```

        LD B,10

```

```

        JP CLCLR1

```

```

CLCLR: LD HL,CLSTK

```

```

        LD B,5

```

```

CLCLR1: XOR A

```

```

CLCLR2: LD (HL),A

```

```

        INC HL

```

```

        DEC B

```

```

        JP NZ,CLCLR2

```

```

        RET

```

```

;***** CLFAG SUB ***** (LOCAL USE)

```

```

;REG = ABCHL

```

```

;SET CARRY & ZERO FLAG

```

```

CLFAG: LD C,00H

```

```

        LD HL,CLBUF

```

```

        LD B,5

```

```

CLFAG1: LD A,(HL)

```

```

        CP 00H

```

```

        JP Z,CLFAG2

```

```

        SET 0,C

```

```

        JP CLFAG3

```

```

CLFAG2: INC HL

```

```

        DEC B

```

```

        JP NZ,CLFAG1

```

```

CLFAG3: LD HL,CLSTK

```

```

        LD B,5

```

```

CLFAG4: LD A,(HL)

```

```

        CP 00H

```

```

        JP NZ,CLFAG5

```

```

        INC HL

```

```

        DEC B

```

```

        JP NZ,CLFAG4

```

```

        SET 7,C

```

```

CLFAG5: PUSH BC

```

```

        POP AF

```

```

        RET

```

```

; *****

```

```

; * SUBROUTINE SHOW SEGMENT *

```

```

; *****

```

```

; REG = ACIX

```

```

SHOW_SEG: LD IX,HEX_TG ;SHOW TARGET YEILD

```

```

CALL SHOW_HG
LD C,SEG1
OUT (C),A
CALL SHOW_LW
LD C,SEG2
OUT (C),A
INC IX
CALL SHOW_HG
SET 7,A
LD C,SEG3
OUT (C),A
CALL SHOW_LW
LD C,SEG4
OUT (C),A

LD IX,HEX_YC ;SHOW COMMUTATIVE YEILD
CALL SHOW_HG
LD C,SEG5
OUT (C),A
CALL SHOW_LW
LD C,SEG6
OUT (C),A
INC IX
CALL SHOW_HG
SET 7,A
LD C,SEG7
OUT (C),A
CALL SHOW_LW
LD C,SEG8
OUT (C),A

LD A,(CAL_TY) ;CHECK FOR SHOW
CP 02H
JP Z,SHOW2
SHOW1: LD IX,TTALTL+2 ;SHOW TOTAL
CALL SHOW_LW
LD C,SEG9
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG10
OUT (C),A
CALL SHOW_LW
LD C,SEG11
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG12
OUT (C),A
CALL SHOW_LW
LD C,SEG13
OUT (C),A

LD IX,FAILTL+3 ;SHOW FAIL
CALL SHOW_HG
LD C,SEG14

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUT (C),A
CALL SHOW_LW
LD C,SEG15
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG16
OUT (C),A
CALL SHOW_LW
LD C,SEG17
OUT (C),A
```

```
LD IX,PASSTL+2 ;SHOW PASS
```

```
CALL SHOW_LW
LD C,SEG18
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG19
OUT (C),A
CALL SHOW_LW
LD C,SEG20
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG21
OUT (C),A
CALL SHOW_LW
LD C,SEG22
OUT (C),A
JP KEYPRS1
```

```
SHOW2: LD IX,TOTAL1+2 ;SHOW TOTAL
```

```
CALL SHOW_LW
LD C,SEG9
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG10
OUT (C),A
CALL SHOW_LW
LD C,SEG11
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG12
OUT (C),A
CALL SHOW_LW
LD C,SEG13
OUT (C),A
```

```
LD IX,FAILTL+3 ;SHOW FAIL
```

```
CALL SHOW_HG
LD C,SEG14
OUT (C),A
CALL SHOW_LW
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีวทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD C,SEG15
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG16
OUT (C),A
CALL SHOW_LW
LD C,SEG17
OUT (C),A
```

;SHOW PASS

```
LD A,(CDSUB)
CK_CDS1: CP 01H
JP Z,CDS1
CK_CDS2: CP 03H
JP Z,CDS2
CK_CDS3: CP 07H
JP Z,CDS3
CK_CDS4: CP 0FH
JP Z,CDS4
CK_CDS5: CP 1FH
JP Z,CDS5
CK_CDS6: CP 3FH
JP Z,CDS6
CK_CDS7: CP 7FH
JP Z,CDS7
CK_CDS8: CP 0FFH
JP Z,CDS8
CDS1: LD IX,PASS1+2
JP SHOW21
CDS2: LD IX,PASS2+2
JP SHOW21
CDS3: LD IX,PASS3+2
JP SHOW21
CDS4: LD IX,PASS4+2
JP SHOW21
CDS5: LD IX,PASS5+2
JP SHOW21
CDS6: LD IX,PASS6+2
JP SHOW21
CDS7: LD IX,PASS7+2
JP SHOW21
CDS8: LD IX,PASS8+2
SHOW21:
```

```
CALL SHOW_LW
LD C,SEG18
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG19
OUT (C),A
CALL SHOW_LW
LD C,SEG20
OUT (C),A
INC IX
CALL SHOW_HG
LD C,SEG21
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีเมลทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUT (C),A
CALL SHOW_LW
LD C,SEG22
OUT (C),A
JP KEYPRS1

```

```

; *****
; *          SUBROUTINE FOR SHOW          *
; *****

```

```

SHOW_HG: LD HL,TAB_SEG
LD A,(IX+0)
AND OFOH
SRL A
SRL A
SRL A
SRL A
ADD A,L
LD L,A
LD A,(HL)
RET

```

```

SHOW_LW: LD HL,TAB_SEG
LD A,(IX+0)
AND OFH
ADD A,L
LD L,A
LD A,(HL)
RET

```

```

; *****
; *          TABLE OF SEGMENT CODE          *
; *****

```

```

TAB_SEG: .DB 3FH ;0
         .DB 06H ;1
         .DB 5BH ;2
         .DB 4FH ;3
         .DB 66H ;4
         .DB 6DH ;5
         .DB 7DH ;6
         .DB 07H ;7
         .DB 7FH ;8
         .DB 6FH ;9

```

```

; *****
; *          RAM AREA          *
; *****

```

```

.ORG 2000H
ST_CODE: .RS 1
BUF_RVS: .RS 1
CAL_TY: .RS 1
CDSUB: .RS 1
HEX_TG: .RS 2
BUADLY: .RS 1

```

HEX_YC:	.RS	2
RVPASS1:	.RS	3
RVFAIL1:	.RS	3
RVTOTAL1:	.RS	3
RVPASS2:	.RS	3
RVFAIL2:	.RS	3
RVTOTAL2:	.RS	3
RVPASS3:	.RS	3
RVFAIL3:	.RS	3
RVTOTAL3:	.RS	3
RVPASS4:	.RS	3
RVFAIL4:	.RS	3
RVTOTAL4:	.RS	3
RVPASS5:	.RS	3
RVFAIL5:	.RS	3
RVTOTAL5:	.RS	3
RVPASS6:	.RS	3
RVFAIL6:	.RS	3
RVTOTAL6:	.RS	3
RVPASS7:	.RS	3
RVFAIL7:	.RS	3
RVTOTAL7:	.RS	3
RVPASS8:	.RS	3
RVFAIL8:	.RS	3
RVTOTAL8:	.RS	3
PASS1:	.RS	5
FAIL1:	.RS	5
TOTAL1:	.RS	5
PASS2:	.RS	5
FAIL2:	.RS	5
TOTAL2:	.RS	5
PASS3:	.RS	5
FAIL3:	.RS	5
TOTAL3:	.RS	5
PASS4:	.RS	5
FAIL4:	.RS	5
TOTAL4:	.RS	5
PASS5:	.RS	5
FAIL5:	.RS	5
TOTAL5:	.RS	5
PASS6:	.RS	5
FAIL6:	.RS	5
TOTAL6:	.RS	5
PASS7:	.RS	5
FAIL7:	.RS	5
TOTAL7:	.RS	5
PASS8:	.RS	5
FAIL8:	.RS	5
TOTAL8:	.RS	5
FAILTL:	.RS	5
PASSTL:	.RS	5
TTALTL:	.RS	5
TMPPASS:	.RS	5
DEND:	.RS	5
RATE:	.RS	5

;DIVIDEND
;OPERATER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLSTK: .RS 20 ;STACK CALCULATE
CLBUF: .RS 10 ;BUFFER CALCULATE
PERCENT: .RS 5 ;FOR CALCULATE

ENDRAM:

.END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ยืน ภู่วรรณ, ทฤษฎีและการประยุกต์ไมโครโปรเซสเซอร์ Z-80, พิมพ์ครั้งที่ 1, บริษัท ซีเอ็ดยเคชั่น จำกัด, 2532
2. ทีมงาน อีทีที, CP - 180 CONTROL PACK, บริษัท อีทีที จำกัด, 2534
3. ทีมงาน อีทีที, Z80180 MPU, บริษัท อีทีที จำกัด
4. ทีมงาน อีทีที, Z80 ET - BOARD SINGLE BOARD MICROCOMPUTER, พิมพ์ครั้งที่ 1, บริษัท อีทีที จำกัด, 2532
5. ทีมงาน อีทีที, DOT MATRIX LCD MODULE, บริษัท อีทีที จำกัด
6. ทีมงาน อีทีที, 7210 Z80, บริษัท อีทีที จำกัด
7. William, B.Jr. in The Z-80 Microprocessor Handbook, 1st ed., Howard W. Sams & Co., Indiana, 1978
8. Nichols, J.C. in Data Communications Microprocessor, 1st ed., pp.117-123, Mc Graw-Hill , 1982
9. Zilog, MICROPROCESSOR DATA BOOK TECHNICAL MANUAL, Zilog Co., Ltd., 1988

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

นาย **บุญส่ง จรรย์วรพรต**

เกิดเมื่อวันที่ 16 ตุลาคม พุทธศักราช 2512 ที่กรุงเทพมหานคร จบการศึกษามัธยมศึกษาต้นจาก โรงเรียน มัธยมวัดธาตุทอง กรุงเทพฯ และจบการศึกษาระดับมัธยมศึกษาตอนปลายจาก โรงเรียน รัตนธิเบศร จังหวัด นนทบุรี แล้วเข้าศึกษาต่อในระดับปริญญาตรี ในภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง กรุงเทพฯ ในปีการศึกษา 2531 และจบการศึกษาในปี 2534

นาย **ธนวัฒน์ ยงนิรันดร์วงศ์**

เกิดเมื่อวันที่ 17 เมษายน พุทธศักราช 2513 ที่กรุงเทพมหานคร จบการศึกษาประถมต้นถึงมัธยมปลายจาก โรงเรียน เซนต์คาเบรียล กรุงเทพฯ แล้วเข้าศึกษาต่อในระดับปริญญาตรี ในภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง กรุงเทพฯ ในปีการศึกษา 2531 และจบการศึกษาในปี 2534