



OPEN/SHORT TESTER



ปริญญาโทนั้นเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

009594

ปฏิญานิพนธ์ เรื่อง OPEN/SHORT TESTER.

โดย.

นาย ถวัลย์ กระจ่อมแก้ว

นาย พีรเดช ชนภักดี

นาย ชศขจร สนั่นนาม

ภาควิชา เทคนิคอุตสาหกรรม

อาจารย์ที่ปรึกษา อ. อุกัย ศรีธีระวิโรจน์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
อนุมัติให้นับปฏิญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
อุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบ ปฏิญานิพนธ์

(_____) อาจารย์ที่ปรึกษา

(_____) กรรมการ

(_____) กรรมการ

(_____) กรรมการ

(_____) กรรมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ในการสร้างเครื่อง OPEN SHORT TESTER ออกแบบโดยอาศัย MICROPROCESSOR ควบคุมการทำงานทั้งระบบ สำหรับปริภูมิตัวฉบับนี้ จะกล่าวถึงรายละเอียดเกี่ยวกับทางด้านตัวเครื่อง (HARD WARE) , การทำงานของระบบ, ตลอดจน PROGRAMME ที่ใช้ควบคุม (SOFT WARE)

คุณสมบัติ ของเครื่อง OPEN/SHORT TESTER เป็นเครื่องที่สามารถใช้ตรวจสอบหรือทดสอบ IC. ภายหลังจากการผลิต ซึ่งผ่าน ขบวนการผลิตต่างๆ มาแล้ว โดยจะตรวจสอบในลักษณะของการ SHORT, หรือ ขาด หรือ มีข้อผิดพลาด ที่เกิดขึ้นกับโครงสร้างภายในของ IC. เป็นการทดสอบว่า ดีพอที่จะนำไปทดสอบผลทางด้าน CHARACTERISTIC หรือไม่ หรือถ้าทดสอบพบว่า มีการ SHORT หรือ ขาด หรือ คุณภาพไม่ได้ตามต้องการ ของโครงสร้างภายใน OPEN SHORT TESTER สามารถตรวจสอบและตัด IC เหล่านี้ออกได้ ก่อนจะนำไปผ่าน CHARAC TERISTIC TESTER จะทำให้ประหยัดเวลาและค่าใช้จ่ายอย่างมากๆ ในอุตสาหกรรมการผลิต IC.

ABSTRACT .

THE OPEN SHORT TESTER BUIDING IS DESIGNED AND CONTROLLED BY MICROPROCESSER. FOR THIS PROJECT REPORT WILL BE EXPLAINING DETAIL ABOUT HARD WARE SYSTEM OPERATION ,AND SOFT WARE

THE CHARACTERISTIC FOR OPEN/SHORT TESTER. IT CAN TEST OR CHECK IC. AFTER MANUFACTURING AND PROCESSING PASSED. BY TESTING THE CHARACTERISTIC FOR OPEN, SHORT AND ERROR INSIDE CONSTRUCTION OF IC. AND TO BE TESTING THAT. IT GOOD FOR GO TO CHARACTERISTIC TESTING OR NOT. IF TESTING FOUND THAT ABOUT OPEN OR SHORT OR LOW QUALITY OF INSIDE IC CONSTRUCTION. OPEN/SHORT TESTER CAN CHECK AND REJECT OUT BE FORE IT WILL GO TO PASS THE CHARACTERISTIC TESTER . WHICH IT CAN SAVE TIMES AND COSTS FOR IC. MANUFACTURING INDUSTRIAL.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

เครื่อง OPEN/SHORT TESTER เป็นเครื่อง TEST IC ชนิดหนึ่ง ซึ่งใช้ TEST ในลักษณะของการ ขาด หรือ ว่า SHORT , ผิดพลาด ที่เกิดขึ้นภายในตัว IC เช่น WIRE BOND ขาด หรือว่า SHORT กับ WIRE BOND ของขาอื่นหรือเกิดการลัดวงจรระหว่างขา IC ซึ่งอยู่ภายใน PACKAGE IC ไม่สามารถมองเห็นได้ นอกจากนำ มา X-RAY ดู แต่ในการผลิตจะมีการผลิตเป็นจำนวนมากเป็นล้านๆ ตัวซึ่งจะนำมา X-RAY นั้นจะเสียเวลามากๆ จึงต้องใช้เครื่อง OPEN SHORT TESTER มาช่วย หลักการของเครื่องคือ นำเอาระดับ VOLTAGE อ้างอิงที่ ขา IC ตัวที่ได้มาตรฐาน มา เก็บในรูปของ DATA ใน EPROM โดยจะเก็บทุกขาของ IC เบอร์นั้นๆ ถ้าต้องการ TEST IC เบอร์อะไรก็เอา DATA ของ IC เบอร์นั้นที่เก็บใน EPROM ออกมาเปรียบเทียบกับระดับ VOLTAGE ที่ได้จากการวัด IC ตัวที่กำลัง TEST ซึ่งเบอร์เดียวกัน โดยเปรียบเทียบ ขาดต่อขา แล้ว นำมาตัดสินว่าเท่ากัน หรือ อยู่ในย่านที่เรากำหนด ได้หรือไม่ ถ้าไม่ได้ก็แสดงออกมาเป็นลักษณะของ LED REJECTED DISPLAY ถ้า ได้ก็จะ DISPLAY ไปที่ LED ACCEPTED DISPLAY แต่การทำงานทั้งหมดที่กล่าว มานั้น MICROPROCESSOR เป็นตัวสั่งงาน และควบคุม วิทยานิพนธ์นี้ยังสามารถนำไป ต่อกับ ชุด HANDLER เป็นชุด LOAD IC ในการ TEST อัตโนมัติ ซึ่งทำให้ TEST ได้รวดเร็วและได้ปริมาณมากๆ ใช้ในอุตสาหกรรมการผลิต IC

ถวัลย์ กระจ่อมแก้ว

33 - 131208

พีรเดช ชุนภักดิ์

33 - 131213

ยศขจร สนั่นนาม

33 - 131217

สารบัญ

		หน้า
บทที่ 1	ทฤษฎี open/short Tester	1-13
บทที่ 2	ทฤษฎี Current Force และ circuit ของเครื่อง	14-24
บทที่ 3	ทฤษฎี EPROM Program Mapping และ Flowchart	25-37
บทที่ 4	Soft Ware ของเครื่อง	38-50
บทที่ 5	CPA - Control Pack A	51-59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPEN/SHORT TEST

คุณสมบัติ ใช้ความต่อเนื่องของอุปกรณ์อิเล็กทรอนิกส์ ได้แก่ IC, TRANSISTOR, DIODE DIP RESISTOR โดยสามารถวัดได้ถึง 40ขา สูงสุด สำหรับ IC นั้นสามารถใช้วัดได้ทุกตะกั่ว และทุก PIN นั้นจะ Short (ได้แก่ op-Amp ความถี่สูงๆ)

ประโยชน์ ให้ Check อุปกรณ์ทางอิเล็กทรอนิกส์ ที่มีจำนวนมากได้แก่โรงงานผลิต IC บางแห่งไม่มีเครื่อง TESTER จึงจำเป็นต้องลงไป TEST ที่อื่น เราสามารถใช้เครื่อง Continuty tester ตรวจสอบในขั้นต้นคัดแยกส่วนที่มีข้อผิดพลาด ที่เกิดจากขั้นตอนการผลิตทำให้มีบาง PIN ของอุปกรณ์ ไม่ได้ถูกต่อออกมา ทำให้ประหยัดค่าขนส่งในส่วนนี้ได้

ทฤษฎี จากหลักการของ CORVER TRACER เราจะใช้วิธี SCAN VOLTAGE จาก บวกสูงที่สุดและลดลงเรื่อยๆ จนถึงลบมากที่สุด ซึ่งขณะที่ไม่มีการวัดหรือไม่มีอุปกรณ์ใดๆ นั่นคือไม่มีกระแสไหล (กระแสจะอยู่ในแนวตั้ง) ถ้าเราต่อ Resistor ที่ Input ของ Curve Tracer เราจะเห็นเส้นเอียงๆ ที่หน้าจอโดยเส้นจะยังเป็นเส้นตรงอยู่ แต่มุมของมันจะขึ้นอยู่กับค่า Resistor ถ้า R มีค่ามาก เส้นจะทำมุมน้อย (กับแกนนอน) และถ้าเราเพิ่มค่า R เส้นก็จะตั้งขึ้นถ้าเรา Short input ของ Curve Tracer จะทำให้เกิดเส้นตรงในแนวตั้ง นั่นคือกระแสไฟสูงสุด Voltage ตกลงต่ำที่สุดนั่นเอง

ดังนั้นเมื่อเราเอา Diode ไปใส่ที่ Input ของ Curve tracer เราจะได้ เส้นที่เป็นเส้นตรงที่หักมุม คล้ายมุมฉาก ขึ้นอยู่กับค่าของ break down Voltage ของ diode ดังแสดงในรูปที่ 1 โดยทั่วไป ถ้าเป็น Silicon diode

จุดที่เป็นมุมฉากจะมีค่าประมาณ 0.66 V นั่นคือในช่วงที่เป็น Forward bias ถ้า Voltage มีค่ามากกว่า break down Voltage (0.66 V) จะมีกระแสไหลจำนวนมาก และถ้าลดลงจนต่ำกว่าค่า break down Voltage ก็จะมีกระแสไหลน้อยมาก

ดังนั้นถ้าเราต้องการจะวัด ค่า break down Voltage ของ diode ตัวนี้สามารถอ่านค่าได้โดยตรงกับจอของ Curve Tracer แต่ สำหรับเครื่องวัดความต่อเนื่องที่จะกล่าวถึงต่อไปนี้จะใช้หลักการที่ตรงกันข้ามกับ Curve Tracer นั่นคือเราจะป้อน Constant current มีค่าประมาณ - 5 mA โดย จะป้อน 2 ครั้งคือ +5mA แล้ววัดค่า Voltage แล้วจึงป้อน -5 mA เราจะสามารถ วัด Voltage ที่ตกคล่อมเท่ากับค่า Break down Voltage ของ Function นั้น และถ้าอุปกรณ์ที่จะวัดเป็น Resistor เราก็จะทราบค่าความต้านทานจากกฎของ OHM นั่นคือ $R = E / I$ โดย E เป็นค่าที่วัดได้และ I คือค่า Constant Current นั้นเอง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROLLER เป็นส่วนที่ควบคุมการทำงานของ วงจรโดยในที่นี้จะใช้ Z80 เป็น CPU
ในการควบคุม

KEYBOARD & DISPLAY ใช้สำหรับ Load ข้อมูลที่ต้องการ set ค่า ที่ต้องการจะวัด
หรือจะเลือก load จาก EPROM ที่อยู่บน Controller ก็ได้ นอกจากนี้จะช่วยในการ Set up
หรือตรวจซ่อมเมื่อเครื่องมีปัญหาด้วย

DECODER decode ค่าจาก Controller ซึ่งเป็นฐานสองให้เป็นฐานสิบ สำหรับควบคุม
Chanel (PIN ของ DEVICE) ในที่นี้สามารถต่อได้ 40 Chanel

DRIVER เป็นส่วนที่ใช้ขยายกำลังจาก Decoder ให้สามารถ drive relay ได้

CHANEL RELAY ใช้ต่อ PIN ที่ต้องการจะวัด เข้ากับ FORCE และ PIN อื่นๆ จะถูกต่อ
ลง GROUND

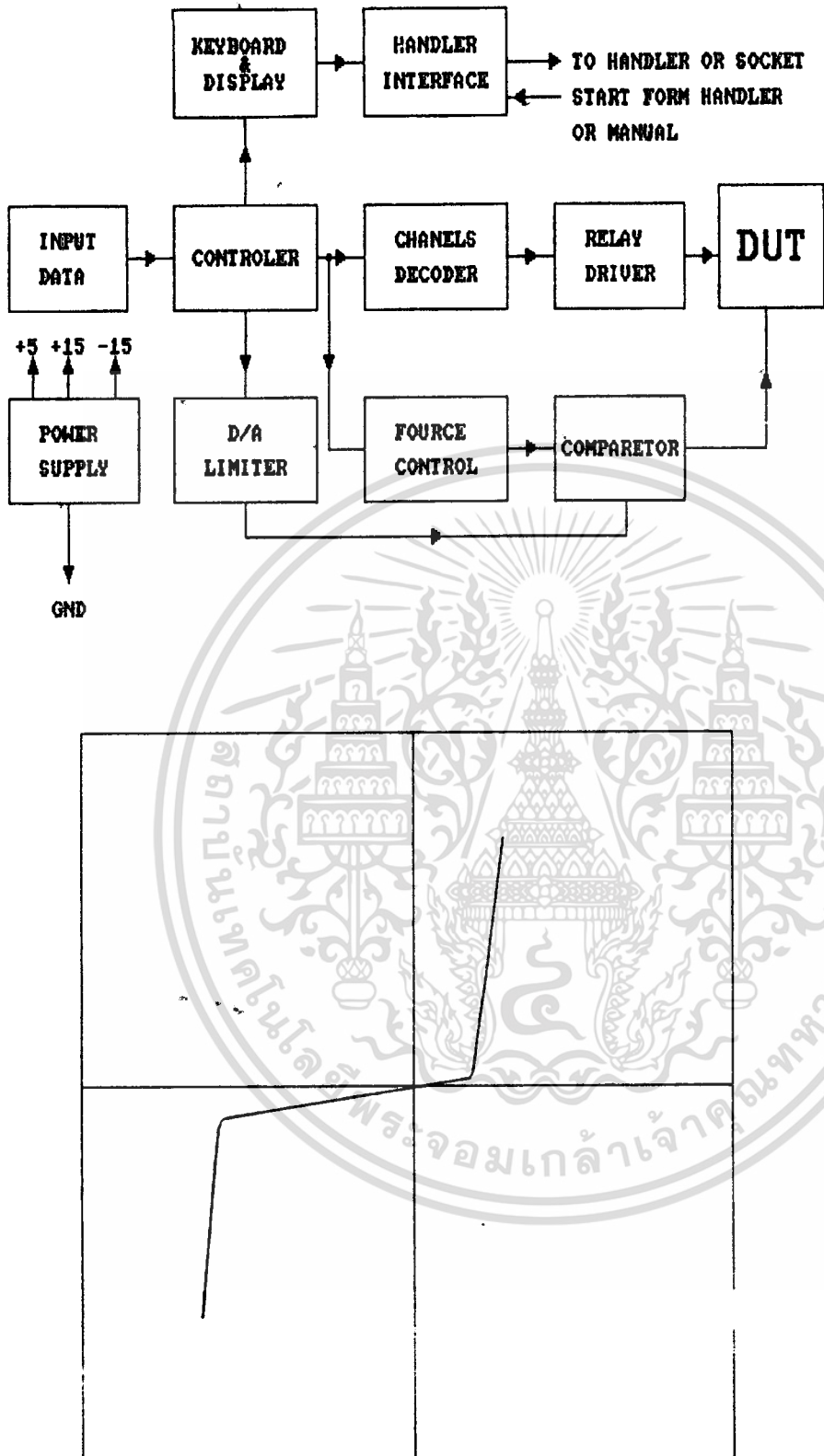
DUT Device under test เป็นอุปกรณ์ที่ต้องการจะ test

CURRENT FORCE สร้าง Constant Current มีค่าประมาณ +5 mA และ -5mA

D/A LIMITER รับ Limit ที่กำหนดจาก Controller มาแปลงเป็น Analog Signal

COMMPARE เปรียบเทียบค่าจาก D/A กับค่าที่วัดได้ แล้วส่งไปให้ Controller ตัด
สินว่า Pass หรือ reject.

HANDLER INTERFACE นำค่าที่ Controller ตัดสินว่า Pass หรือ Reject มา
แสดง (BIN) และสามารถต่อเข้ากับ Handler เพื่อความรวดเร็วในการ Test



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

O/S REJECT

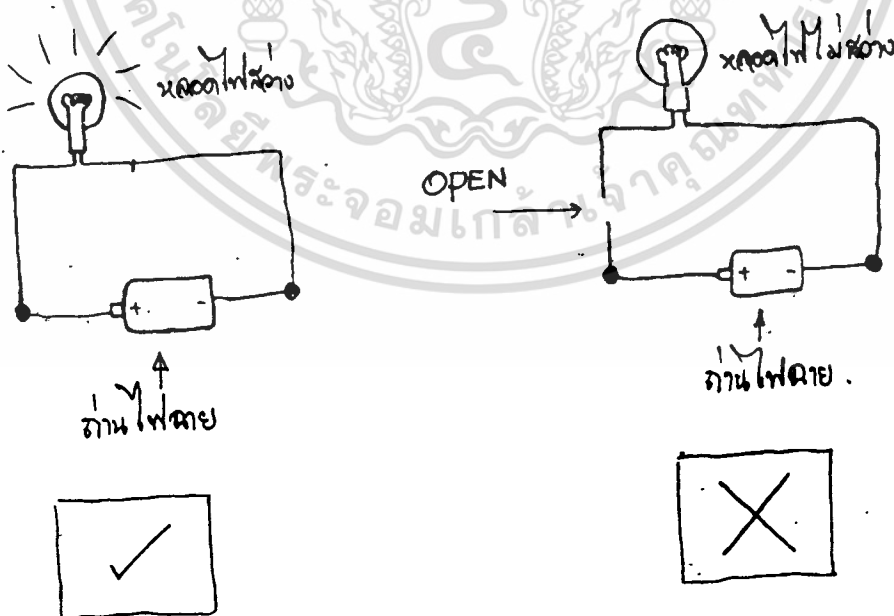
ย่อมาจาก OPEN หรือ SHORT (O/S)

ความหมายของ O :-

O เป็นอักษรย่อของคำว่า OPEN แปลว่า เปิด
 ในที่นี้หมายความว่า ขาดจากกัน หรือ ไม่เชื่อม
ต่อกัน การ BOND คือ การเชื่อมโยงลวดจากตัว
 DIE ไปหา LEAD เพื่อเป็นทางเดินของไฟฟ้าให้ตัว
 I.C สามารถทำงานได้เมื่อการ BOND มีปัญหา
 OPEN คือลวดขาด ณ. จุดใดจุดหนึ่งจะทำให้ตัว I.C
 นั้นไม่สามารถทำงานได้

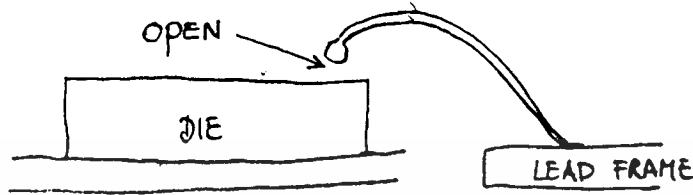
ตัวอย่างง่าย ๆ เปรียบเหมือน วงจรหลอดไฟ ถ้ามีจุด
 ใดจุดหนึ่งเกิด OPEN คือ ขาดจากกัน หลอดไฟก็จะไม่
 ติด (REJECT)

ภาพประกอบ วงจรหลอดไฟ

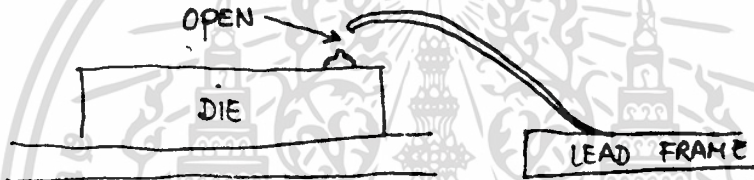


ปัญหา OPEN ที่เกิดจากการ BOND

1. BALL LIFT PAD ลูกบอลไม่ติดที่ PAD หรือ ยกที่ PAD



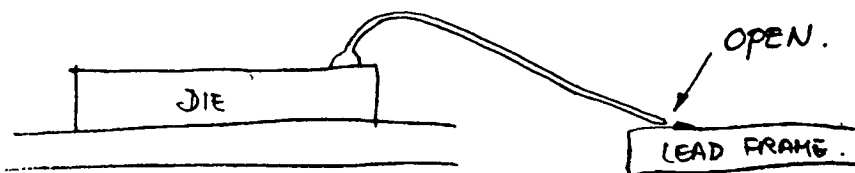
2. BALL NECK BROKEN ลวดขาดที่คอรอยบอนด์



3. STITCH LIFTING บอนด์ไม่ติด หรือ บอนด์ยกที่ LEAD

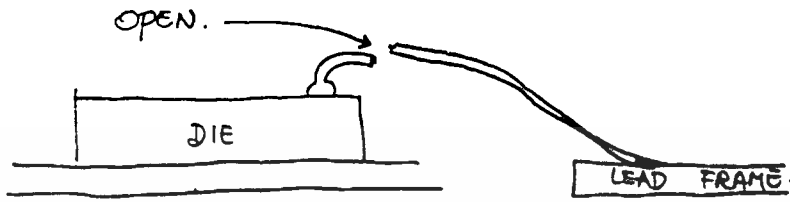


4. STITCH NECK BROKEN รอยบอนด์ขาดที่คอ LEAD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. BROKEN WIRE ลวดขาดจากการบอนด์ หรือ การจับต้อง หรือ มีสิ่งใดไปกระทบเส้นลวดจนขาด หรือ ที่เรียกกันว่า "MISHANDLING"



ปัญหาต่าง ๆ เหล่านี้ เมื่อตรวจพบขณะบอนด์ ต้อง หยุดเครื่องทำการแก้ไข หรือ แจ้งให้ผู้เกี่ยวข้อง ทราบทันที

"ไม่ควรปล่อยงานที่มีปัญหาให้ผ่านไป"

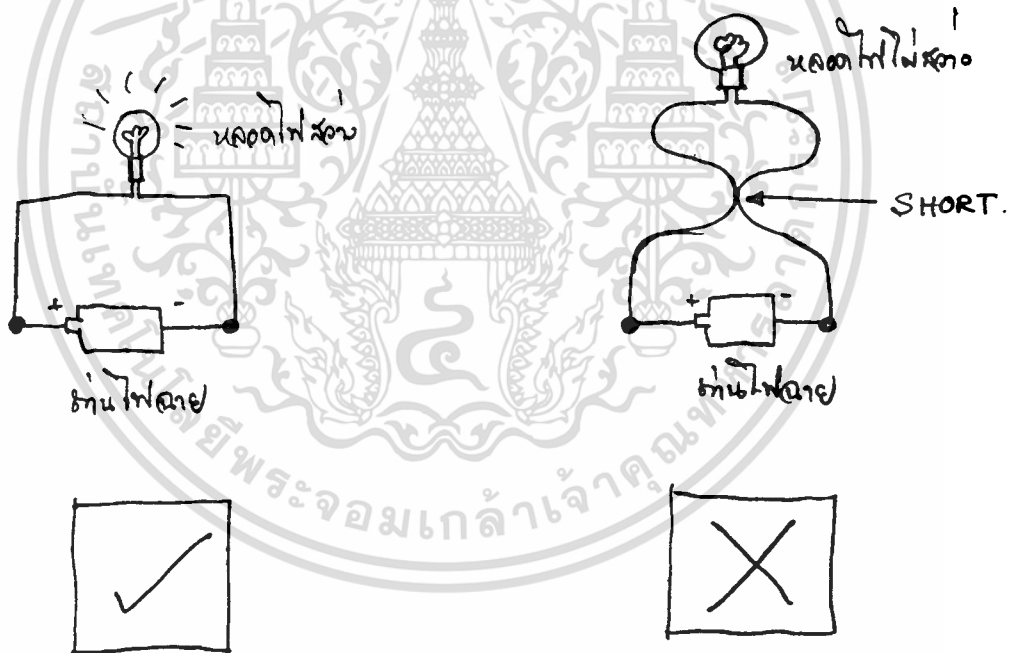
OPEN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของ S :-

S เป็นอักษรย่อของ คำว่า SHORT แปลว่า เชื่อมใน
 ที่นี้ หมายความว่า เข้าไปชิดกัน หรือ เชื่อมติดกัน
 โดยไม่เป็นไปตาม BONDING DIAGRAM กำหนด
 ตัวอย่างใกล้ตัว จะได้ยินเสมอตามบ้าน คือ ไฟฟ้าดับ
 เนื่องจากไฟ SHORT (สายไฟสองเส้นมาแตะกัน)

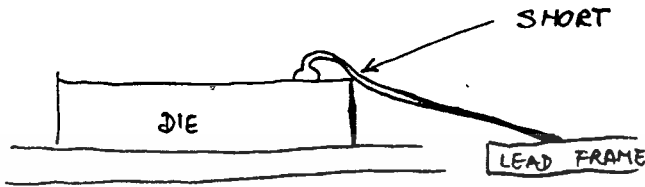
ภาพประกอบ



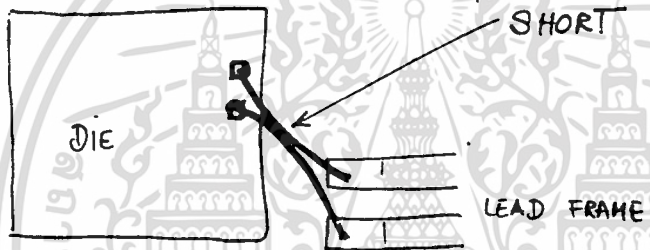
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหา SHORT ที่เกิดจากการ BOND

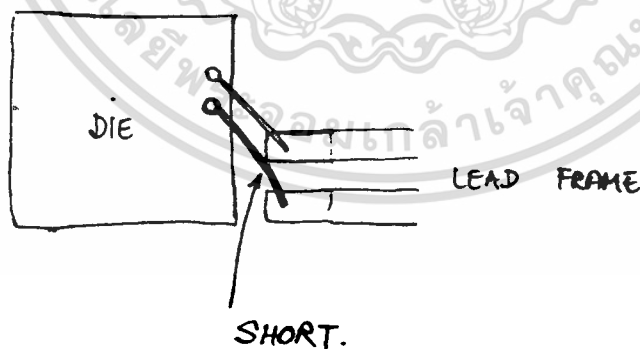
1. WIRE TO DIE EDGE SHORT ลวดแตะขอบ DIE



2. WIRE SHORT WIRE ลวดแตะลวด



3. WIRE SHORT LEAD ลวดแตะ LEAD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4. WIRE SHORT ลวดแตะกันโดยการจับต้อง หรือ มีสิ่งใด
ไปกระทบเส้นลวด จนลวดไปแตะกัน หรือ
ที่เรียกกันว่า "MISHANDLING"

ปัญหาต่าง ๆ เหล่านี้ เมื่อตรวจพบขณะบอนด์ ต้อง
หยุดเครื่องทำการแก้ไข หรือ แจ้งให้ผู้เกี่ยวข้อง
ทราบทันที

"ไม่ควรปล่อยงานที่มีปัญหาให้ผ่านไป"



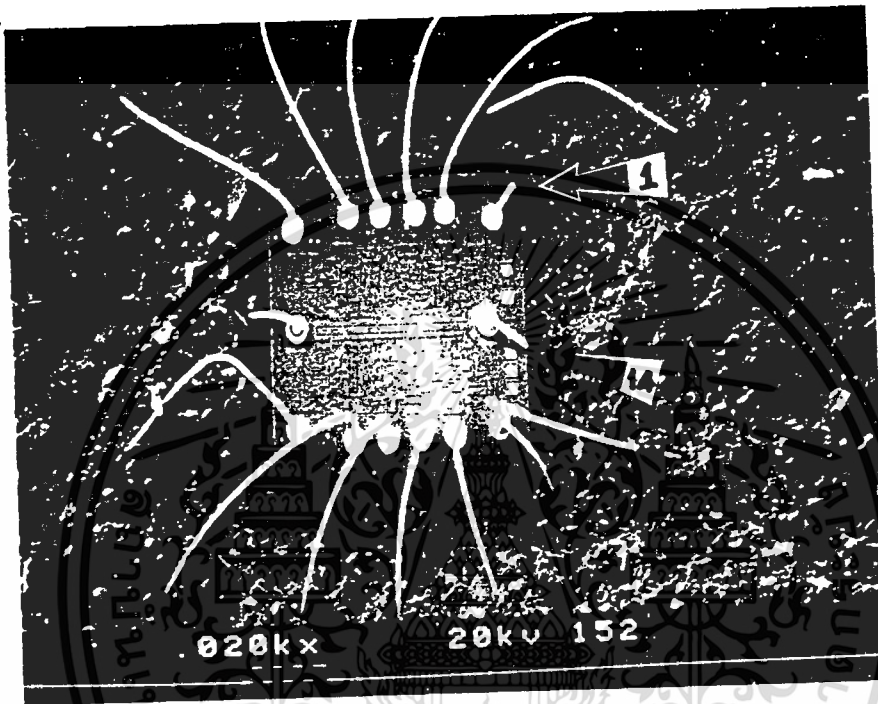


Figure 1

Bond wires at pins 1,14 were broken.

PREPARED BY : Chuntanee S. การ DATE : May 31, 1990 APPROVED BY : SUTHEE A.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

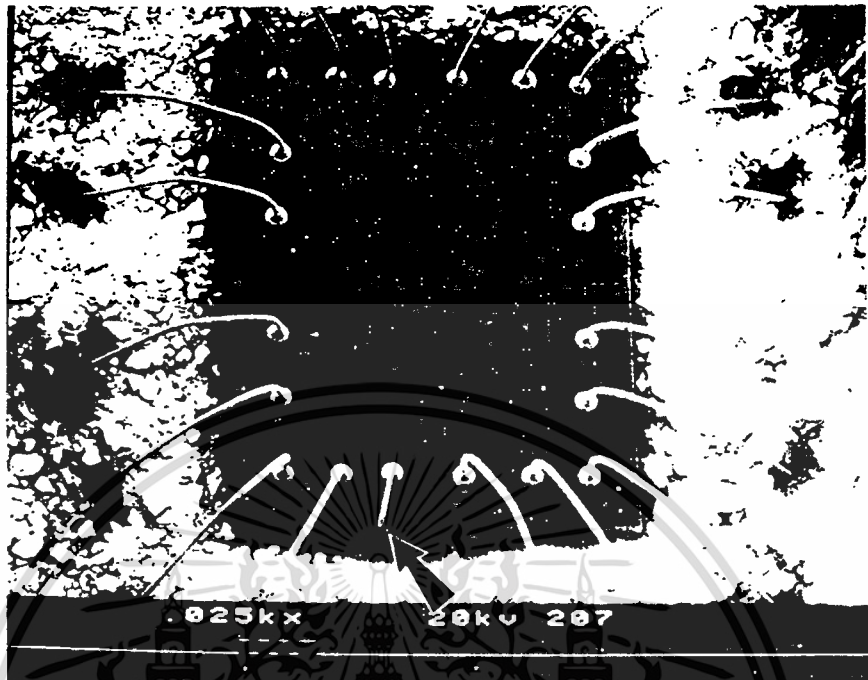


Figure 1
The arrow points to wire break at pin 10.

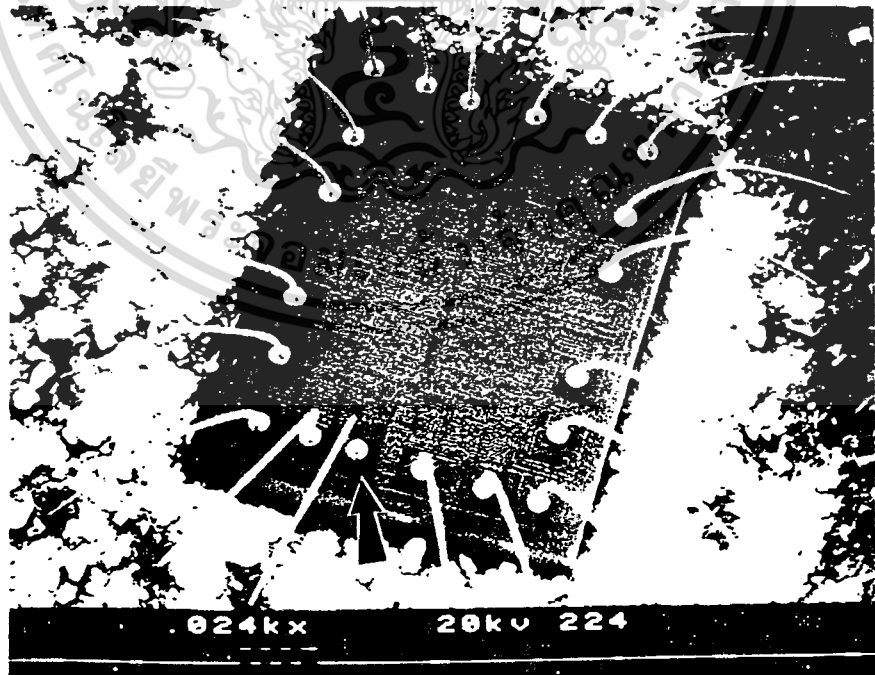


Figure 2
The arrow points to bond neck broken at pin 10.

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

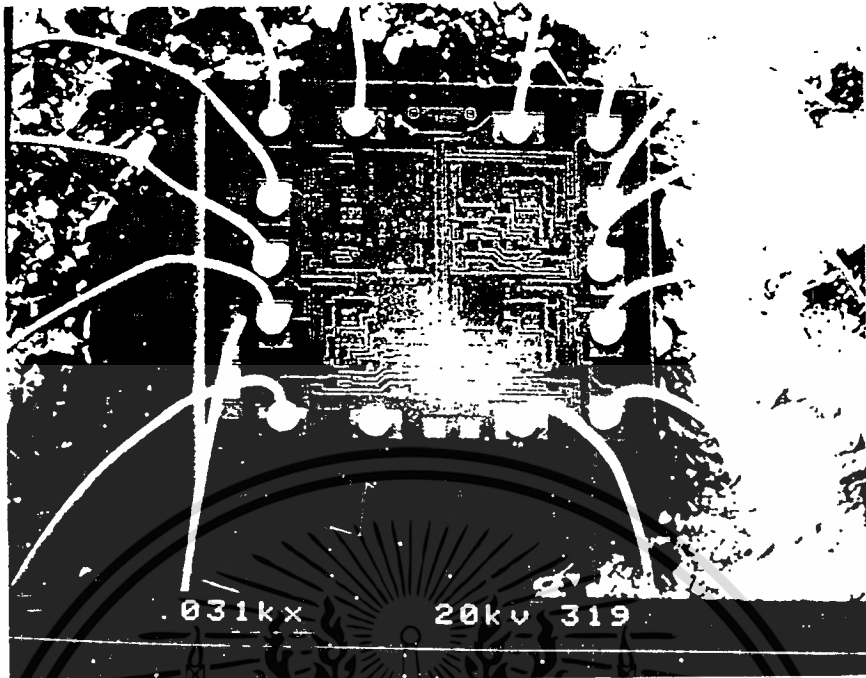


FIGURE 1

The arrow points to bond neck broken at pin 7.

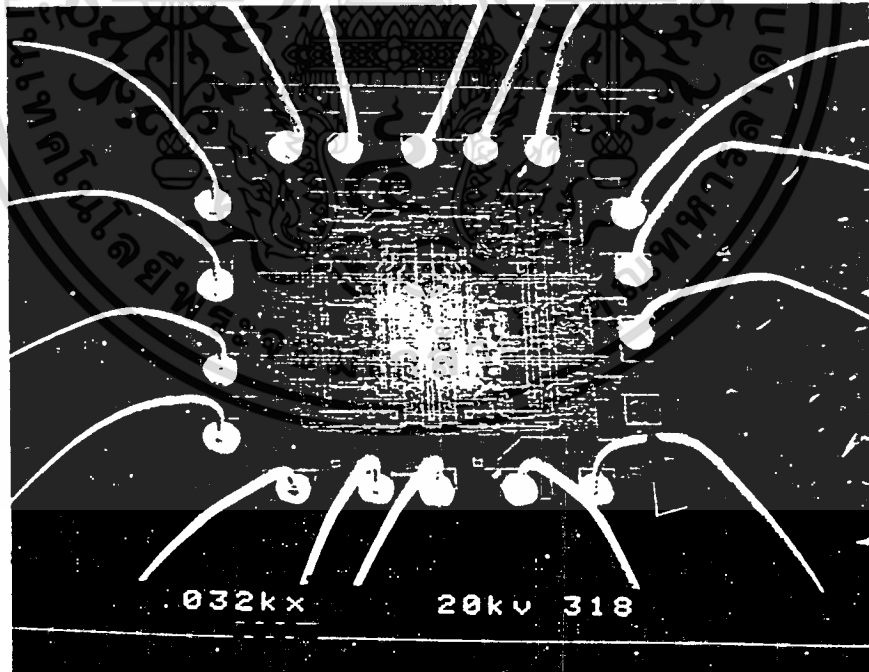


FIGURE 2

The arrow points to bond lift pad at pin 11.

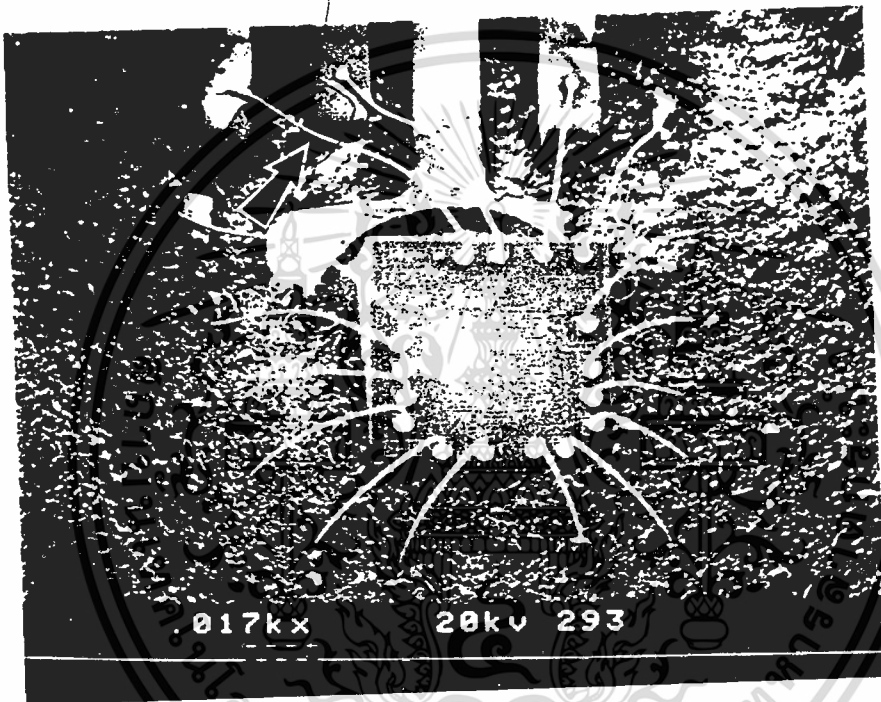
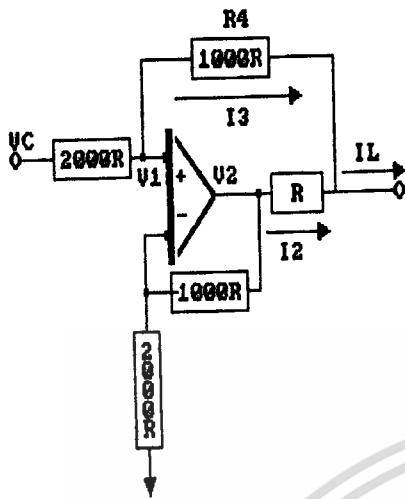


Figure 3

The arrow points to wire pin 2 short to lead pin1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CURRENT FORCE

$$I_3 = \frac{U_C - U_L}{R_3 + R_4}$$

$$= \frac{U_C - U_L}{3000R} \quad \text{-----(1)}$$

$$U_1 = U_L + I_3 R_4$$

$$= U_L + \frac{U_C - U_L}{3} \frac{1000R}{1000R}$$

$$= \frac{2U_L + U_C}{3} \quad \text{-----(2)}$$

GAIN OF THIS OP-AMP ARE $1 + (R_2/R_1)$

THEN $1 + (1000/2000) = \frac{3}{2}$

FORM $U_2 = AU_1 = \frac{3U_1}{2}$

$$= \frac{3}{2} \left(\frac{U_L + U_C}{3} \right)$$

$$= U_L + \frac{U_C}{2} \quad \text{-----(3)}$$

VOLTAGE DROP ON R = $U_2 - U_L$

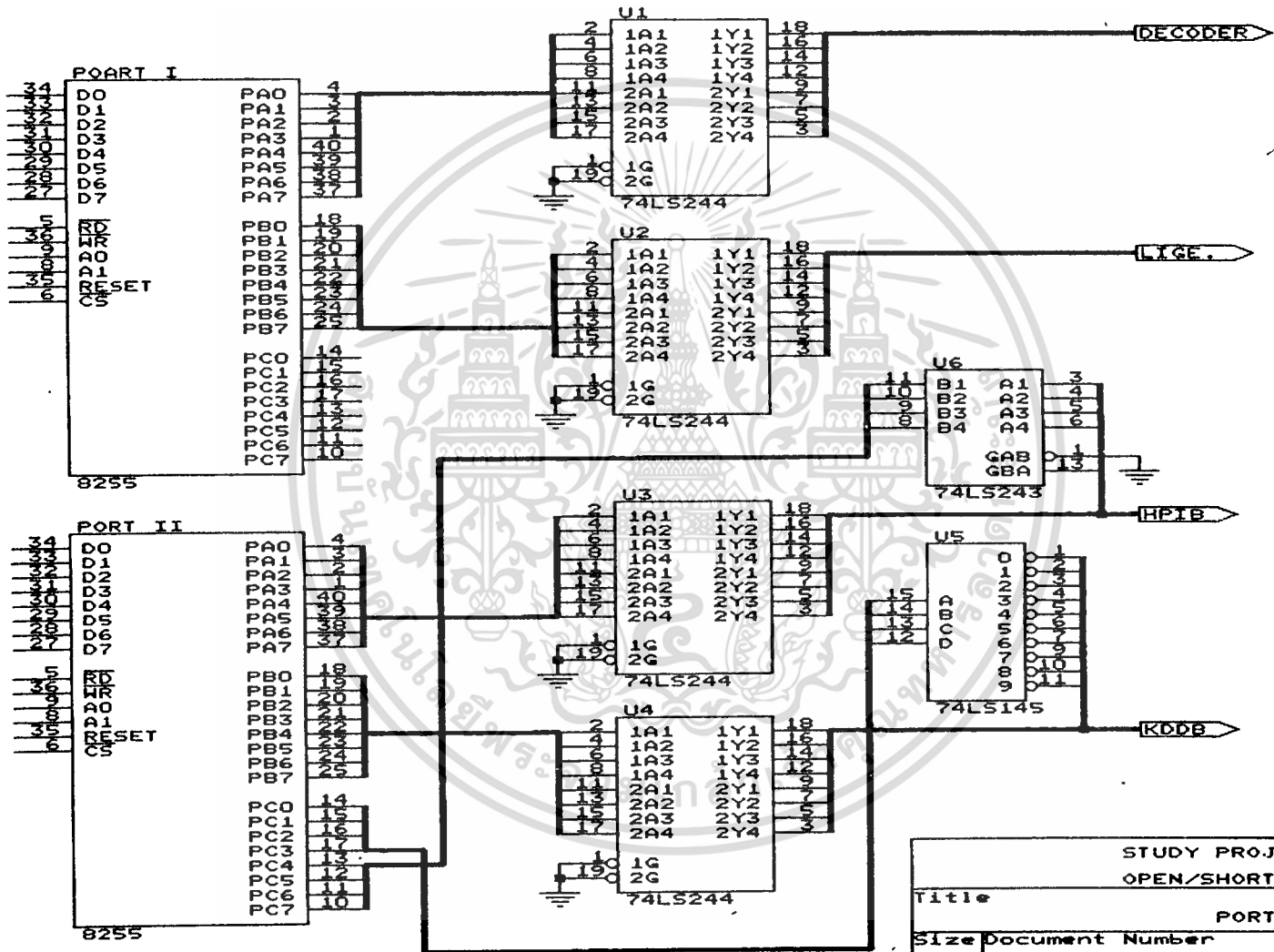
$$= \frac{(U_L + U_C) - U_L}{2}$$

$$= \frac{U_C}{2} \quad \text{-----(4)}$$

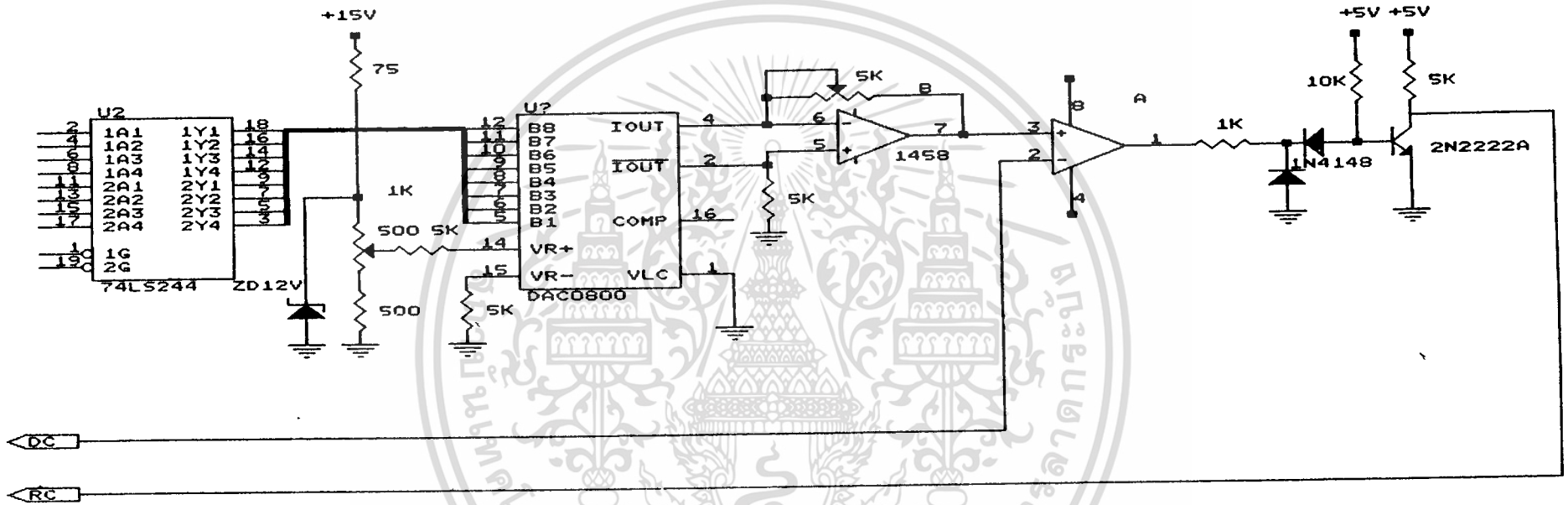
THEN $I_s = \frac{U_C}{2R} ; I_L = I_s + I_3$

IF $|I_3| \ll |I_s|$ THEN $I_L \approx I_s$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไปอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

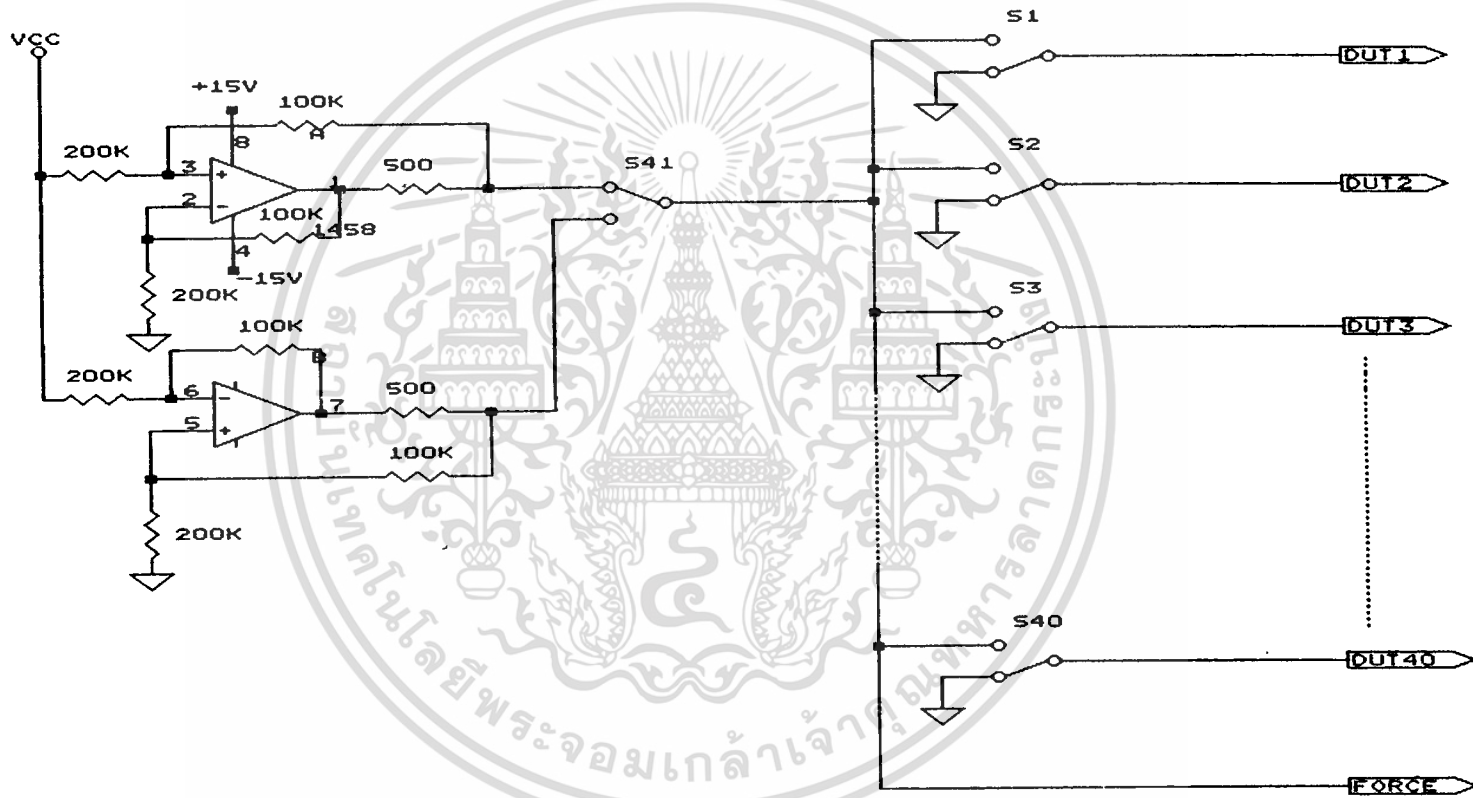


STUDY PROJECT		
OPEN/SHORT TESTER		
Title	PORT	
Size	Document Number	REV
A		
Date:	March 21, 1992	Sheet 1 of 5

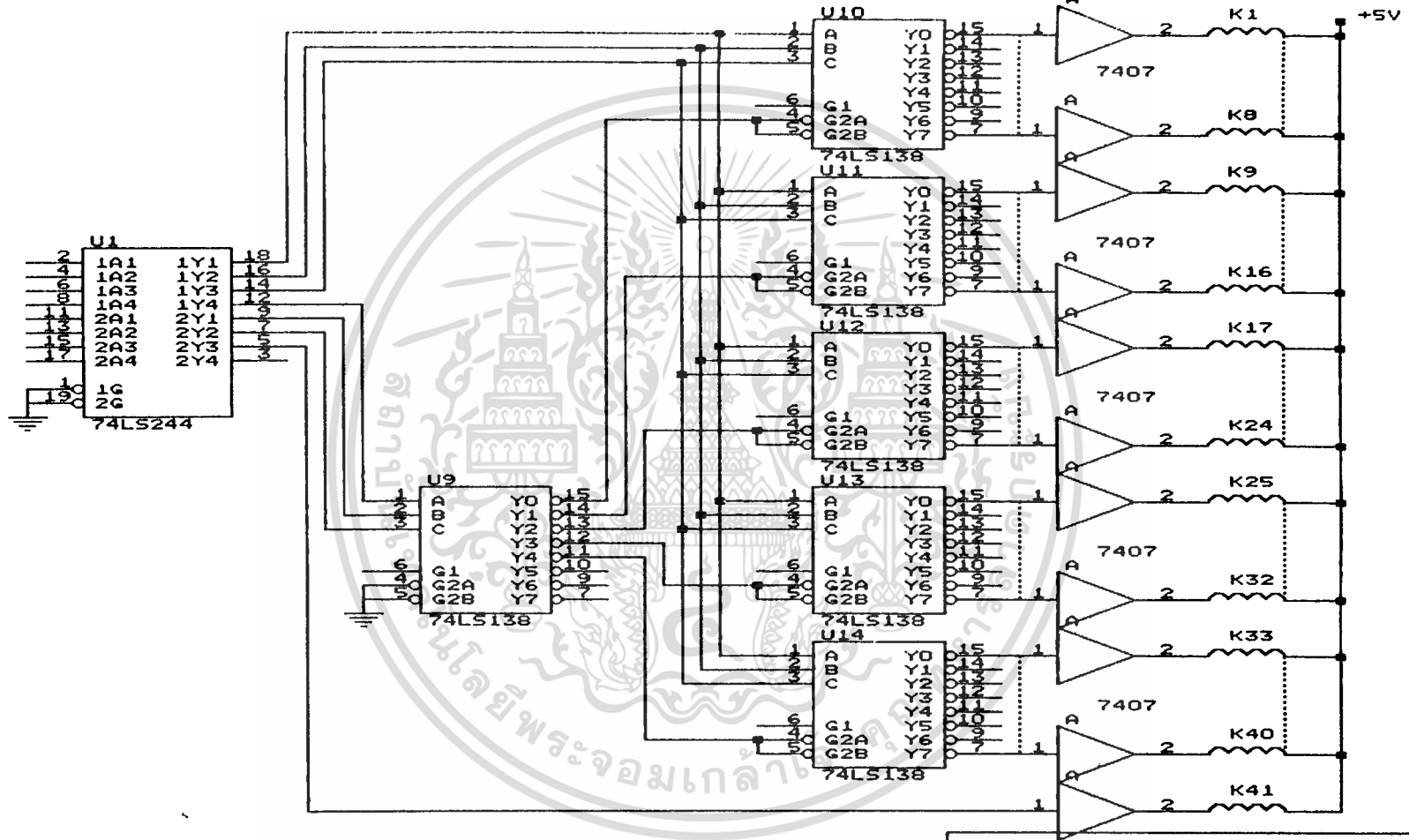


16.

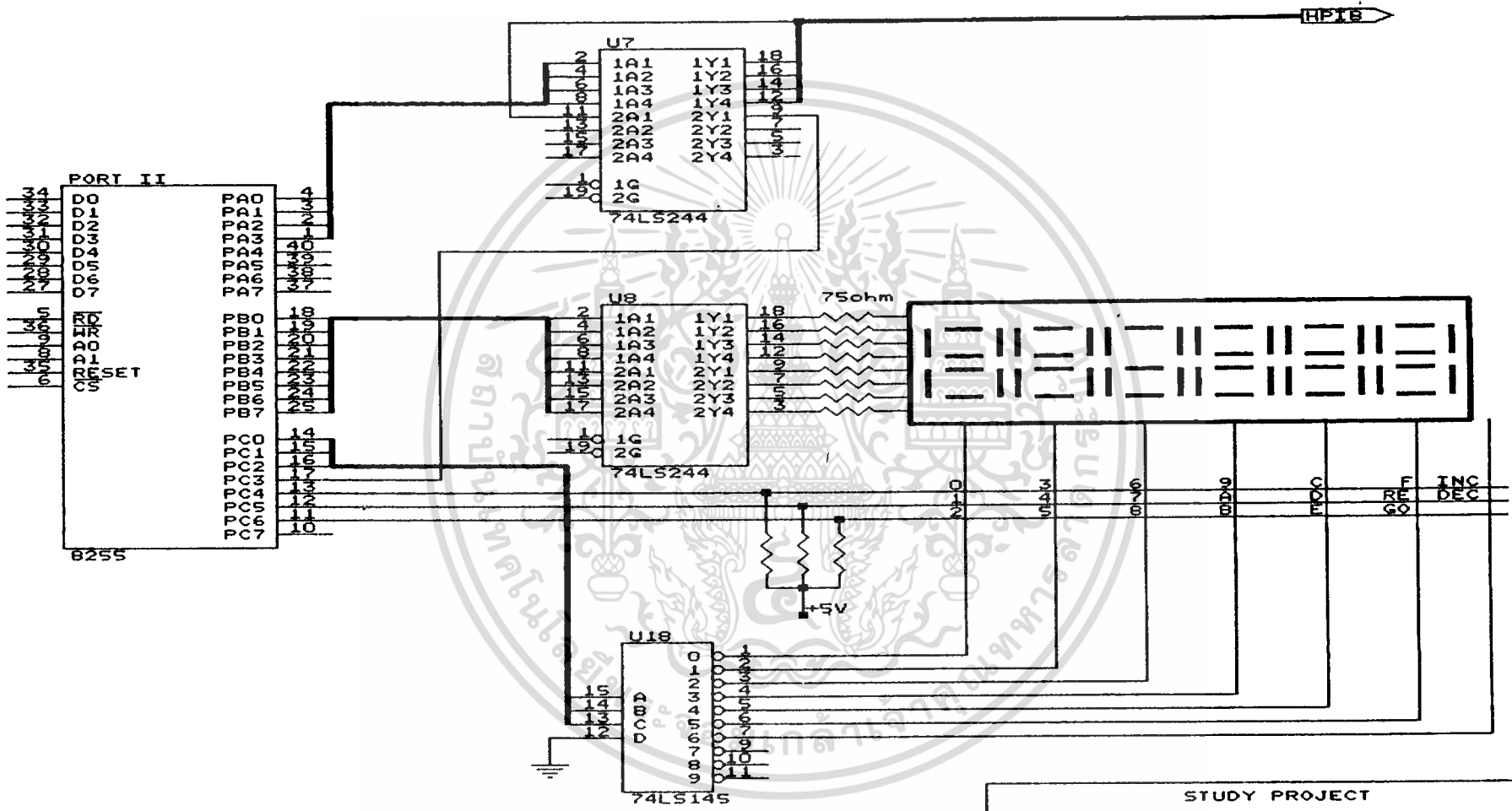
STUDY PROJECT	
OPEN/SHORT TESTER	
Title	
LIMITE GENERATOR	
Size Document Number	
A	REV A
Date:	March 21, 1992 Sheet 2 of 5



STUDY PROJECT		
OPEN/SHORT TESTER		
Title		
FORCE SELECT		
Size	Document Number	REV
A		A
Date:	March 21, 1992	Sheet 3 of 5



STUDY PROJECT		
OPEN/SHORT TESTER		
DECODER		
Size	Document Number	REV
A		A
Date:	March 21, 1992	Sheet 4 of 5

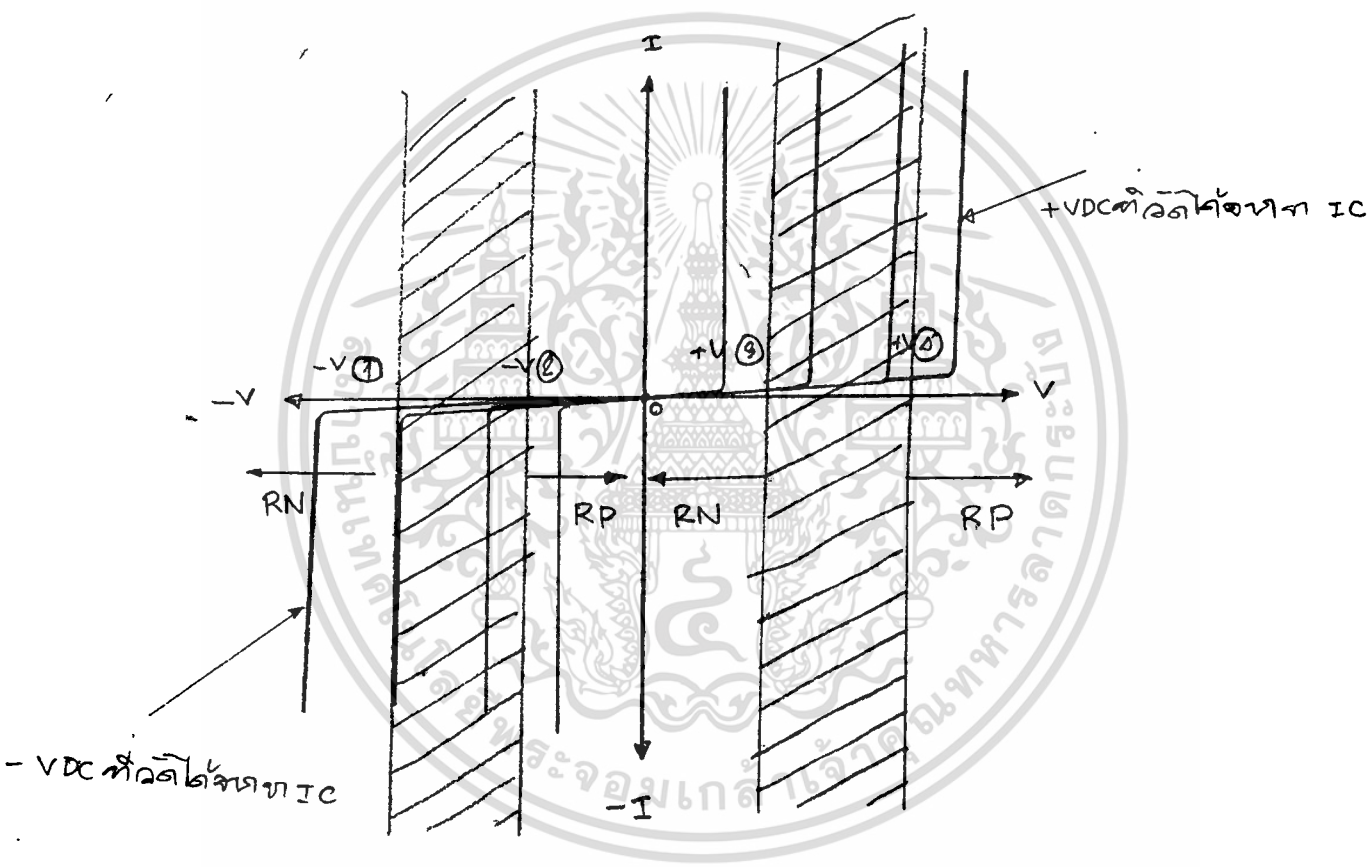


STUDY PROJECT		
OPEN/SHORT TESTER		
Title		
KEYBOARD/DISPLAY DECODE		
Size	Document Number	REV
A		A
Date:	March 21, 1992	Sheet 5 of 5

PORT I การทำงานของ PORT ซึ่งประกอบด้วย 8255 2 ตัว โดยตัวบนทำหน้าที่เป็น PORT I ในการทำงาน ใช้ PORT A (PA) กับ (PB) ส่วน (PC) จะไม่ใช้งานมาดู PA0-PA1 จะใช้ในการส่งรหัส ข้อมูลขนาด 8BIT จากหน่วยCONTROL กลางผ่าน 74LS244 ซึ่งเป็น BUFFER. ขนาด 8 BIT เพื่อส่งไปยังส่วนของ DECODER (ดูวงจรภาค DECODER ประกอบ) โดย 3 BIT บนคือ PA0-PA2 จะใช้เป็นรหัสส่งไปให้ U10-U14 ซึ่งเป็น 74LS138 ถอดรหัสออกเป็น 8 LINE ของแต่ละตัวและรหัสที่ถูกถอดออกเป็น 8 LINE จาก U10-U14 นี้จะถูกส่งไปยังชุด DRIVER ซึ่งเป็น IC # 7407 เพื่อขับให้ COIL ของ LEAD RELAY แต่ละตัวให้ต่อ CONTACT เพื่อให้ TEST SIGNAL ผ่านเข้าไปยังขาของ IC ที่ TEST (DUT) ตามที่ถอดรหัส โดยจะถอดรหัสได้ 40 ขา จาก YO ของ U10 จนถึง 17 ของ U14 เพราะฉะนั้น จะต้องใช้ LEAD RELAY ถึง 40 ตัว (เพื่อให้สามารถ TEST ขนาด 40ขาได้) ส่วนอีก 3 BIT ต่อมาคือ PA3-PA5 นั้นจะเป็นรหัสให้กับ U9 ซึ่งเป็น 74LS138 เพื่อ DECODE สัญญาณไปควบคุมการถอดรหัสของ U10-U14 ให้ทำงานทีละตัวในลักษณะของการ SCAN อีกทีหนึ่ง โดยจะ SCAN จาก U10 ไปยัง U14 ส่วน U10 ถึง U14 จะ SCAN OUT PUT ได้ถึง 40 CHANNEL เพื่อขับให้ LEAD RELAY ทำงานต่อ CONTACT ต่อขาของ EC ที่นำมา TEST จาก ขา 1 ถึง ขา 40 ตามลำดับเพื่อที่จะจ่าย TEST SIGNAL เข้าที่ละขาของ IC ที่นำมา TEST พร้อมกับวัด DC VOLTAGE ที่ขานั้นออกมาด้วย ซึ่งจะกล่าวในส่วนต่อไป ส่วนขาที่ไม่ได้ถูกเลือกที่จะ TEST ต่อสายกราวด์หมดโดย CONTACT NC ของ LEAD RELAY เช่นถ้า ถอดรหัสให้ TEST ขา 1 CONTACT ของ LEAD RELAY ตัวที่ 1 ก็จะทำงาน ส่วน ขา 2 - ขา 40 ถูกต่อส่งกราวด์หมด โดย NC ของ LEAD RELAY ให้ดูวงจร FORCE SELECT ประกอบส่วนสัญญาณจาก PA6 จะนำไปผ่าน DRIVER เหมือนกัน เพื่อนำไป ขับ LEAD RELAY ในการ SELECT TEST SIGNAL ว่าจะให้เป็นบวกหรือลบ ให้ดู วงจร FORCE SELECT ประกอบด้วย โดย FORCE CURRENT หรือ CURRENT TEST SIGNAL ใช้ประมาณ +5mA และ -5mA โดย OP-AMP IC 1458 จะจ่าย CURRENT +5mA ส่วน OP- AMP ตัวล่างจ่าย -5mA และ O/P ของทั้ง 2 จะไปเข้า CONTACT ของ LEAD RELAY ตัวที่ 41 เพื่อใช้ในการเลือก ว่าจะต้องการ TEST SIGNAL แบบบวกหรือลบ

ส่วนของ PBO -PB7 เป็นส่วนที่สำคัญเพราะเป็นส่วนของ DATA ที่จะนำมาเปรียบเทียบกับ DC VOLTAGE ที่วัดได้จากขาของ IC ที่นำมา TEST หลังจากป้อน TEST SIGNAL เข้าไปแล้ว เพราะฉะนั้น DATA พวกนี้เป็น DATA ที่แสดงถึงลักษณะ VOLTAGE ค่าของ VOLTAGE ที่ขา IC แต่ละขา ของ IC แต่ละเบอร์ ซึ่งได้จากการทดสอบของบริษัทผู้ผลิตนำมาเก็บไว้ใน EPROM หรือ ROM ไว้แล้ว ในรูปของ DATA ขนาด 8 BIT เพราะฉะนั้น DATA พวกนี้ต้องถูกแปลงให้เป็นสัญญาณ ANALOG เสียก่อน เพื่อให้ได้ระดับ VOLTAGE ออกมาเสียก่อน เพราะฉะนั้นจาก PBO - PB7 จะถูกส่งผ่าน BUFFER 8 BIT 74LS244 U2. OUT POT จาก BUFFER U2 ถูกส่งไปยังชุด DAC 0800 (ให้ดูวงจร ชุด LIMITE GENERATOR ประกอบด้วย) ซึ่งเป็นตัวแปลงสัญญาณ DIGITAL ขนาด 8 BIT ให้เป็นสัญญาณ ANALOG โดยจะให้ O/P PORT สองแบบคือเป็นบวกกับเป็นลบ I OUT, I OUT และ OUT POT ทั้งสองจะถูกส่งไปยังขาบวกลบของ OP-AMP เพื่อแปลงให้เป็นลักษณะของ VOLTAGE อีกที (I-V CONVERTER) การที่เราใช้ VOLTAGE เป็นค่าบวกและลบนั้น เพื่อว่า เวลาเรา TEST แบบจ่าย SIGNAL TEST แบบลบเวลาเราเปรียบเทียบเราจะต้องเอา DATA ที่เป็นลบมาแปลงเป็น ANALOG ซึ่งจะได้ค่าเป็นลบ นำมาเปรียบเทียบกันได้ส่วนค่าบวกก็ทำนองเดียวกัน หลังจาก DATA ถูกแปลงเป็น ANALOG และ ผาสย O-V CONVERTER แล้วจะได้ OUT PUT ออกที่ขา 7 ส่งมายังขา 3 ของ OP-AMP ตัวที่ 2 เพื่อนำมาเปรียบเทียบกับ DC VOLTAGE ที่วัดออกมาได้หลังจากการป้อน TEST SIGNAL เข้าไปที่ขา IC ที่ TEST ที่ขา 2 ของ OP - AMP ผลการเปรียบเทียบออกที่ ขา 1 ส่งไปยัง BASE ของ 2N 2222A โดยผ่าน CATHODE ของ DIODE IN4148 TRANSISTOR จะทำงานตลอดเวลาคือจะให้ OUT PUT เป็น 0 ตลอดเวลาจะให้ OUT PUT ที่ขา 1 เป็นลบ VDC+5VOLT ที่จ่ายผ่าน R 10K จะถูก BY PASS ผ่าน DIODE IN4148 ลงกราวด์ไปที่ให้ BASE ของ 2N 2222A ไม่มี BIAS ทำให้เกิดการ CUT OFF ขึ้น O/P จะเป็น "1" ออกมา ส่งไปยังหน่วย ควบคุม เพื่อให้ตัดสินใจว่าจะผ่านการทดสอบหรือไม่ผ่านการทดสอบ

ลักษณะของการป้อน TEST SIGNAL เข้าไปที่ขา IC ที่ TEST จะป้อน 2 แบบคือ แบบเป็นบวกกับเป็นลบแล้ววัด DC VOLTAGE ออกมา ซึ่งจะได้ค่าออกมาเป็นลบซึ่งจะเป็นระดับไปลบเท่าไรนั้นเราไม่ทราบซึ่งขึ้นอยู่กับว่า IC ที่เรานำมา TEST นั้นมีความผิดพลาดหรือ ERROR มากน้อยเพียงไร หรือบางตัวอาจจะ SHORT ซึ่งจะไม่มีความผิดพลาดหรือ ERROR ออกมาจะมีกระแสไหลเต็มที่หรือบางตัวอาจจะขาดซึ่งจะมี VOLTAGE ออกมาเท่ากับแกน- V แต่การที่เราจะ CHECK ให้ได้ค่าใดค่าหนึ่งเป็นค่าที่มาตราฐานนั้นไม่ได้ซึ่งย่อมจะมีย่านของการเปลี่ยนแปลงอยู่บ้าง ขึ้นอยู่กับบริษัทผู้ผลิตและการ DOPE สารถึงตัวนำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเราจึงทำการ CHECK โดยการส่ง DATA ที่เป็นลบมา 2 ระดับคือที่ -V1 กับ -V2 มาทำการเปรียบเทียบ -V1, -V2 เป็น -DC VOLTAGE ที่ผ่านการแปลงจาก DAC0800 มาแล้วและผ่าน I-V CONVERTOR มาเข้าที่ขา 3 ของ OP-AMP เป็นขา INVERTING โดยครั้งแรกจะส่ง V1 มาก่อนถ้าเปรียบเทียบกันแล้ว OUTPUT ที่ขา ของ OP-AMP เป็นบวกแสดงว่า ค่า-DC VOLTAGE ที่วัดได้จากการ TEST มีค่าลบมากกว่าซึ่งเกินไปจากย่านที่เราจะ CHECK ดูจากรูป มีที่ศทางไปทาง-V เกินจากส่วนที่แลเงาเราจะถือว่าเป็นการ REJECT NEGATIVE (RN) คือมีค่าเป็นลบเกินกว่าที่จะยอมรับได้แต่ถ้า OUTPUT ของ OP-AMP เป็นลบเราจะถือว่ายังอยู่ในย่านอยู่ แต่ยังไม่นำมาตัดสินใจว่าจะ REJECT หรือไม่ ต่อไปเราจะจ่าย -V2 มา CHECK อีกที่หนึ่งซึ่งเป็นลบอีกระดับหนึ่งซึ่งน้อยกว่า -V1 มาเปรียบเทียบกับ -DC VOLTAGE ที่วัดได้จากการ TEST อีกครั้ง ถ้า OUTPUT ที่ได้ที่ขา 1 เป็นลบเราถือว่า -DC VOLTAGE ที่วัดได้มีค่าเป็นลบน้อยผิดปกติเกินจากย่านที่เราต้องการถือเป็นการ REJECT POSITIVE (RP) ส่วนทางด้าน การ CHECK ทางด้านบวกก็มีหลักการคล้ายกันแต่แทนที่จะเป็นไฟลบกลับเป็นไฟบวกแทนโดยจ่าย +V3, +V4 มา CHECK ทีละครั้งเหมือนกัน การ REJECT ก็เหมือนกัน ซึ่ง OUTPUT ของ OP-AMP ที่ผ่านการเปรียบเทียบแล้วถ้าเป็นบวก OUTPUT ที่ขา COLLECTER ของ TR. 2N2222A จะเป็น 0 และถ้าเป็นลบ OUTPUT จะเป็น 1 ส่งไปยังหน่วยควบคุมตัดสินใจว่า IC ที่ TEST นั้นดีหรือเสีย ส่งผลออกมาแสดงผลจาก DISPLAY ต่อไป แต่ถ้ามากกว่าเกินจากส่วนที่แลเงาทั้งทางด้านบวกและลบแล้วถือว่า IC ที่ TEST นั้นมีข้อบกพร่องใช้ไม่ได้

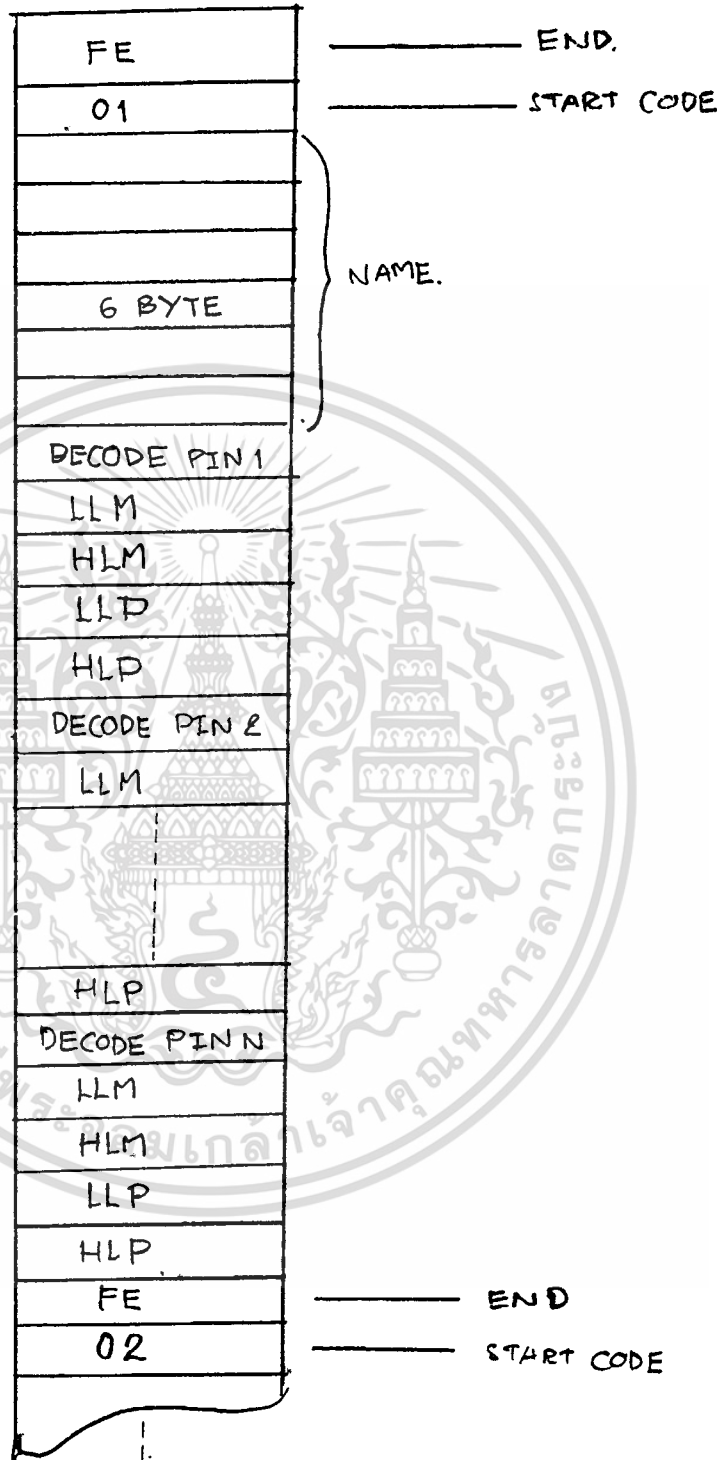
PORT II 8255 ตัวที่ 2 ส่วนของ PA0 - PA1 ทำหน้าที่เป็น OUTPUT PORT โดยผ่าน 74LS244 U3 ส่งไปควบคุม ชุด (HPIB) HANDLE PROBER INTERFACE ซึ่งเป็นชุดการ LOAD งานเข้าเครื่อง TEST แบบอัตโนมัติ แทนการ LOAD ด้วยมือสามารถทำงานได้เร็วกว่าการ LOAD ด้วยมือ ถ้าต้องการที่จะต่อใช้งานในที่นี้จะไม่ขอล่าถึงเพราะไม่ได้ต่อใช้งาน ส่วน PBO-PB7 เป็นส่วนของ OUTPUT ที่จะส่งบ DATA ออกไป DISPLAY ที่ 7-SEGMENT โดยผ่าน U8 74LS244 ส่วน PC0-PC2 เป็นส่วนของการ SCANKEYBOARD ในแนว COLUMN และ DISPLAY เป็นทั้ง I/O PORT ให้ดูรูป วงจร KEY BOARD DISPLAY DECODE โดยผ่าน U5 74LS145 BCD TO DECIMAL DECODER. ส่วน PC4-PC6 ทำการ SCAN KEY BOARD ในแนว ROW. เพื่อคอย CHECK การกด KEY BOARD

EPROM PROGRAM MAPPING

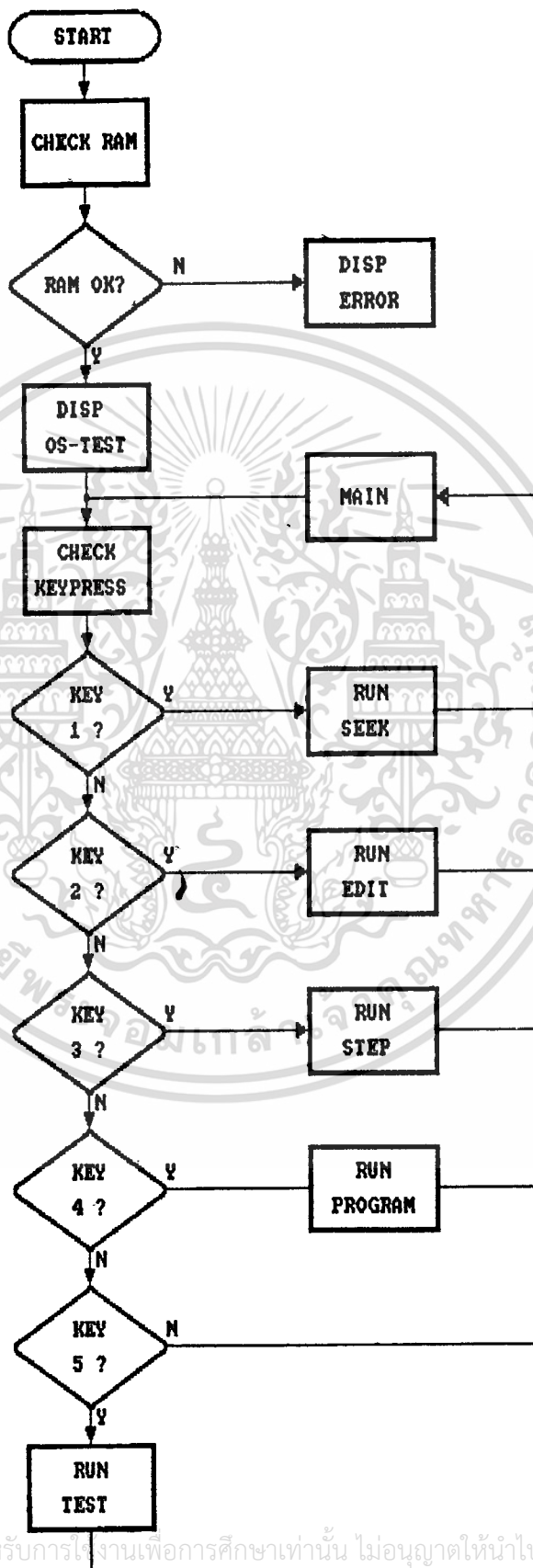
ลักษณะโปรแกรมแบ่งได้เป็น 8 ส่วนดังนี้

1. เป็นรหัสที่ใช้แทนเบอร์ของ IC ตามที่เรากำหนดขึ้นเพื่อเป็นตัวกำหนดส่วนหรือตำแหน่งที่อยู่ของ IC เบอร์นั้น
2. PRODUCT NAME 6BYTE. เป็นส่วนที่เก็บโปรแกรมใช้แสดงว่ากำลัง TEST PRODUCT เบอร์อะไรอยู่ จะใช้เนื้อที่ 6 BYTE.
3. DECODER 1 BYTE เป็นส่วนที่เก็บโปรแกรมใช้สำหรับการถอดรหัสขา IC ที่ จะ TEST ใช้เนื้อที่ 1 BYTE
4. LOW LIMITE MINUS 1BYTE. (LLM) เป็นส่วนที่เก็บโปรแกรมที่ใช้สำหรับการ TEST ทางด้านไฟลบระดับต่ำใช้เนื้อที่ 1BYTE.
5. HIGH LIMITE MINUS 1 BYTE เป็นส่วนที่ใช้เก็บโปรแกรมสำหรับการ TEST ทางด้านไฟลบระดับสูงใช้เนื้อที่ 1BYTE
6. LOW LIMITE PLUS 1BYTE. เป็นที่เก็บโปรแกรมที่ใช้สำหรับการ TEST ทางด้านไฟบวกระดับต่ำใช้เนื้อที่ 1BYTE.
7. HIGH LIMITE PLUS 1BYTE. เป็นส่วนที่เก็บโปรแกรมที่ใช้สำหรับการ TEST ทางด้านไฟบวกระดับสูง ใช้เนื้อที่ 1BYTE.
8. END (FE_H) 1BYTE เป็นส่วนที่เก็บโปรแกรมที่บอกให้เครื่องทราบว่า จบ หรือสิ้นสุดของการ TEST IC ตัวนั้นแล้ว ใช้เนื้อที่ 1BYTE.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



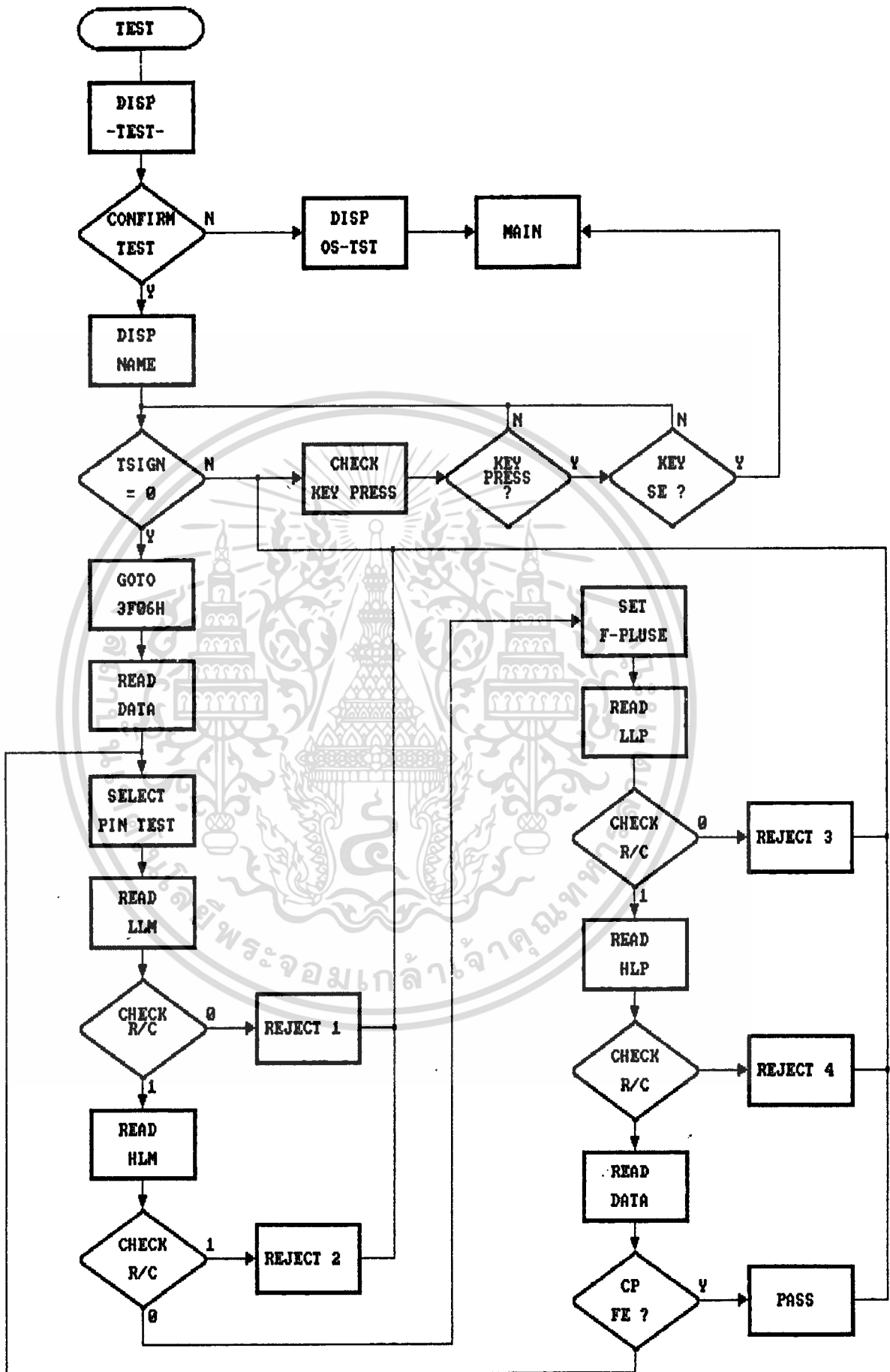
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ flow chart. เมื่อ start เครื่องชุด control จะเริ่มทำการ check ram ถ้าเกิดการ ERROR ของ ram จะ display " ERROR" ถ้า ram ปกติก็จะแสดง display " os-tst " แล้วทำการ check การกด key โดยจะ check key no. 1 ถึง 5 ว่ามีการกด key เหล่านี้หรือไม่ โดยทำการ check ดังนี้ อันดับแรก check การกด key 1 ถ้ากด key "1" จะวิ่งไปทำการ run program mode "1" ซึ่งจะได้กล่าวต่อไปเมื่อทำเสร็จจะวนกลับไป check key ใหม่ ถ้าไม่ได้กด key "1" จะวิ่งไป check การกด key "2" ถ้ากด key "2" จะวิ่งไป run mode "2" แล้ววนกลับมา check key ใหม่ ถ้าไม่ได้กด key "2" จะวิ่งไป check การกด key "3" ถ้ามีการกด key "3" ก็จะวิ่งไป run program mode "3" แล้ววนกลับมา check key ใหม่ ถ้าไม่มีการกด key "3" ก็จะวิ่งไป check การกด key "4" ถ้ามีการกด key "4" ก็จะวิ่งไป run mode "4" แล้วกลับไป check key ใหม่ ถ้าไม่มีการกด key "4" ก็จะไป check key "5" ถ้ามีการกด key "5" ก็จะไป run mode "5" แล้ววนกลับมา check key ใหม่ ถ้าไม่มีการกด key "5" ก็จะวนกลับมา check key "1-5" ใหม่วนแบบนี้ไปเรื่อยๆ

ความหมายของ mode "1-5"

- mode "1" = seek mode
- mode "2" = edit mode
- mode "3" = step mode
- mode "4" = program mode
- mode "5" = test mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

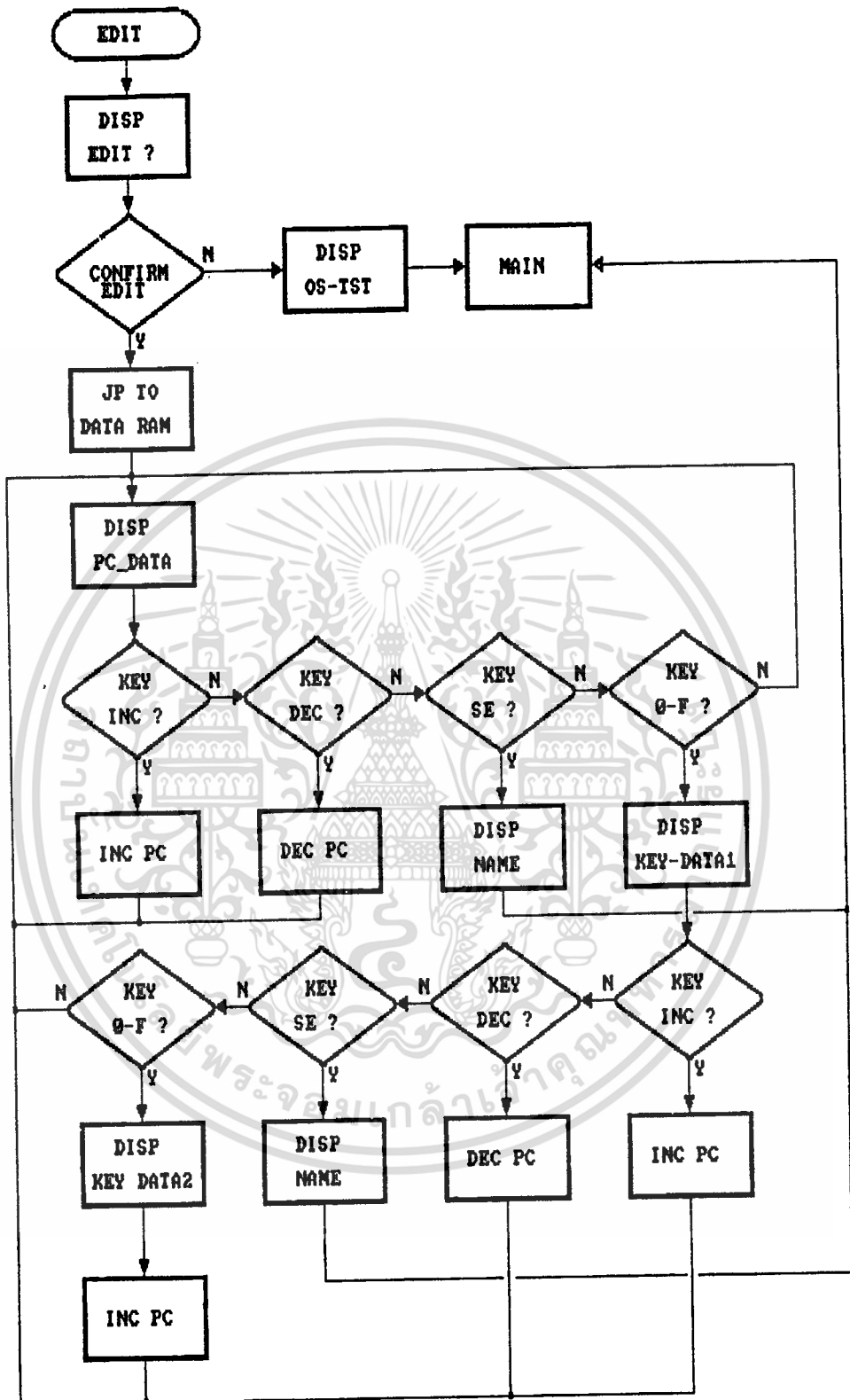
modetest .

เริ่มโปรแกรมเครื่อง display "test" นอกให้ทราบว่าอยู่ใน mode test จากนั้นรอกการกด key ถ้ากด key "dec" เครื่องจะ display "06-test" เป็นการบอกให้ทราบว่า จะออกไปสู่ main program แต่ถ้ากด key "inc" จะเป็นการเข้า mode test และเครื่องจะ display ชื่อของเบอร์ ic ที่จะ test ออกมา จากนั้นรอกการกด key เพื่อ start เป็น key start ถ้าไม่มีการกด key start เครื่องจะ check การกด key ถ้าไม่มีการกด key เครื่องจะวนไปรอ check การกด start ถ้ามีการกด key เครื่องจะ check ว่าเป็น key "se" หรือไม่ถ้าไม่เครื่องก็จะวนไปรอสัญญาณการ start อีกแต่ถ้าเป็นการกด key "se" เครื่องจะออกไปสู่ main program และถ้ามีการกด start ขึ้นมาเครื่องจะไปทำงานที่ 3F06 H อ่านข้อมูล แล้วทำการกำหนด pin ในการ tset จากนั้นมาอ่าน data ทางด้าน low limit minute (llm) ส่งมาเปรียบเทียบกับสัญญาณที่วัดได้จากการป้อนกระแสไฟที่ขา ic ด้วยภาค current force ถ้าเปรียบเทียบแล้วอยู่ใน rang ที่ยอมรับไม่ได้ก็จะทำ program reject 1. เป็นการ reject ทางด้าน -rn แต่ถ้าอยู่ใน rang ที่ยอมรับได้ก็จะไปอ่าน high limit minute (hlm) แล้วทำการเปรียบเทียบ ถ้าไม่ได้ตามที่กำหนดก็จะไป run program reject 2 เป็นการ reject ทางด้าน -rp แต่ถ้า check แล้วอยู่ในย่านที่กำหนดเครื่องจะไป set การใช้ program check ทางด้านบวกโดยเริ่มจาก low limit positive (llp) จากนั้นทำการเปรียบเทียบถ้าไม่อยู่ใน rang ที่กำหนด เครื่องจะไป run program reject 3 reject ทางด้าน +rn แต่ถ้าอยู่ใน rang ที่ยอมรับได้ เครื่องจะอ่านข้อมูลทางด้าน high limit

positive (hlm) แล้วทำการเปรียบเทียบถ้าออกนอก rang ที่กำหนด เครื่องจะไป run programe reject 4 reject +rp แต่ถ้าอยู่ในย่านที่ยอมรับได้จะทำการ check "FE" ถ้าไม่เจอจะวนกลับไปกำหนด pin ต่อไป ถ้าเจอก็จะวนไปรอรับสัญญาณ การ start ใหม่.

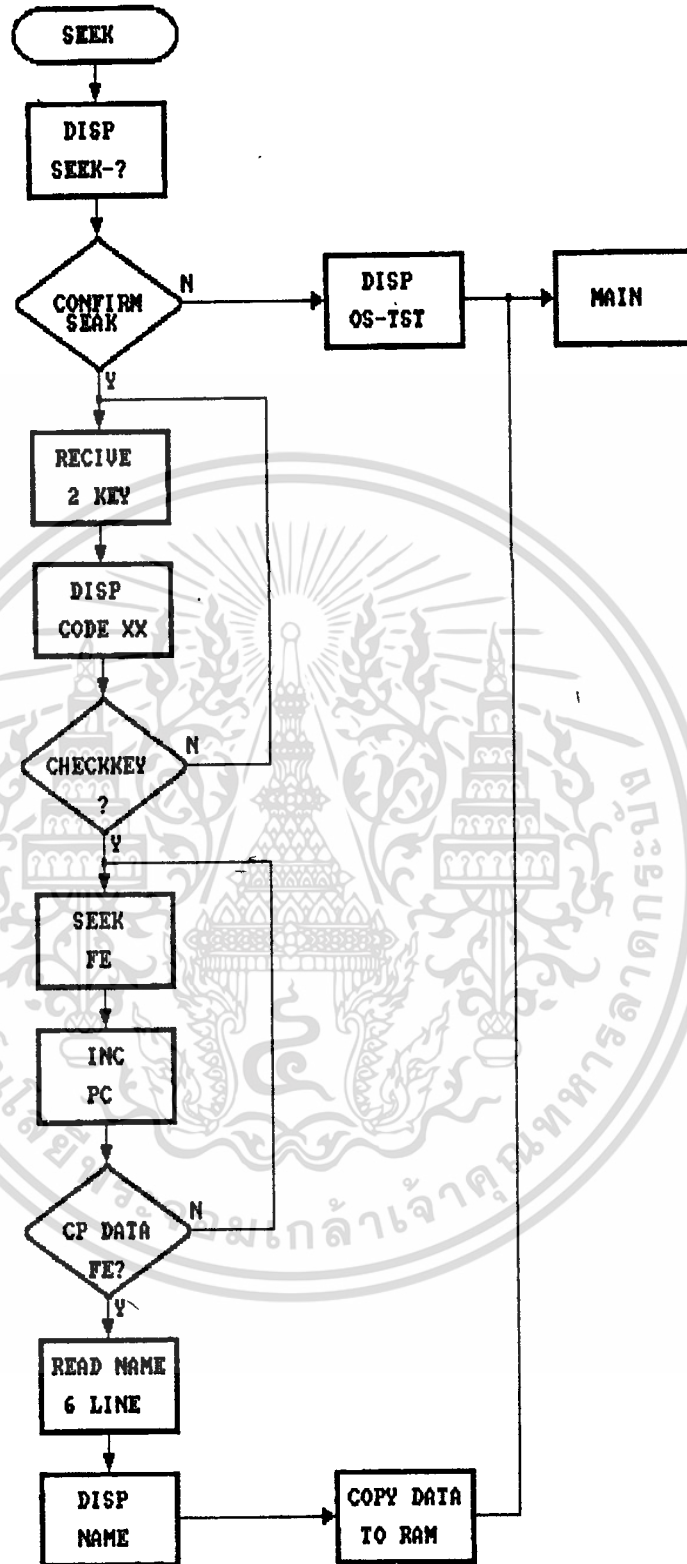


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มการทำงานจะ display "edit?" แล้วจะรอการกด ถ้ากด key "dec" จะออกจาก mode edit เครื่องจะ show `os-tst` แล้วออกสู่ main program แต่ถ้ากด key "inc" เครื่องจะทำการ jump ไปทำงานที่หน่วยความจำ ram จะแสดงค่า address และ data ออกมาให้เห็นจากนั้นรอ การกด key ถ้ากด key "inc", "dec", "se", "0-f" ถ้ากด inc จะเป็นการเพิ่มค่า address ขึ้นครั้งถ้า กด dec จะเป็นการลดค่า address 1 ตำแหน่งแล้วกลับไป display address และ data ค่าใหม่ที่ให้จากการเพิ่มหรือลดค่า address ถ้ากด key "se" เป็นการออกจาก mode ไปยัง main program แต่ถ้ากด key "0-f" ตัวใดตัวหนึ่งเครื่องจะนำข้อมูลนี้ ไปใส่ไว้ใน ช่อง data bit แรกทางขวามือสุดจากนั้นก็ check key แบบเดิมอีกถ้า มีการกด key "0-f" อีกเครื่องจะนำข้อมูลนี้ไปใส่ไว้ที่ data ตัวที่ 2 เป็นข้อมูลตัวที่ สองจากนั้นเพิ่มค่า address ให้อีก 1 ค่าแล้ววนกลับไป display ค่าข้อมูลที่ใส่เข้าไปใหม่วนอยู่แบบนี้เรื่อยๆจนกว่าแก้ไขเสร็จและกด key "se" จึงจะออกจาก mode ไปสู่ main program. (ให้ดูภาพ flow chart edit ประกอบ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

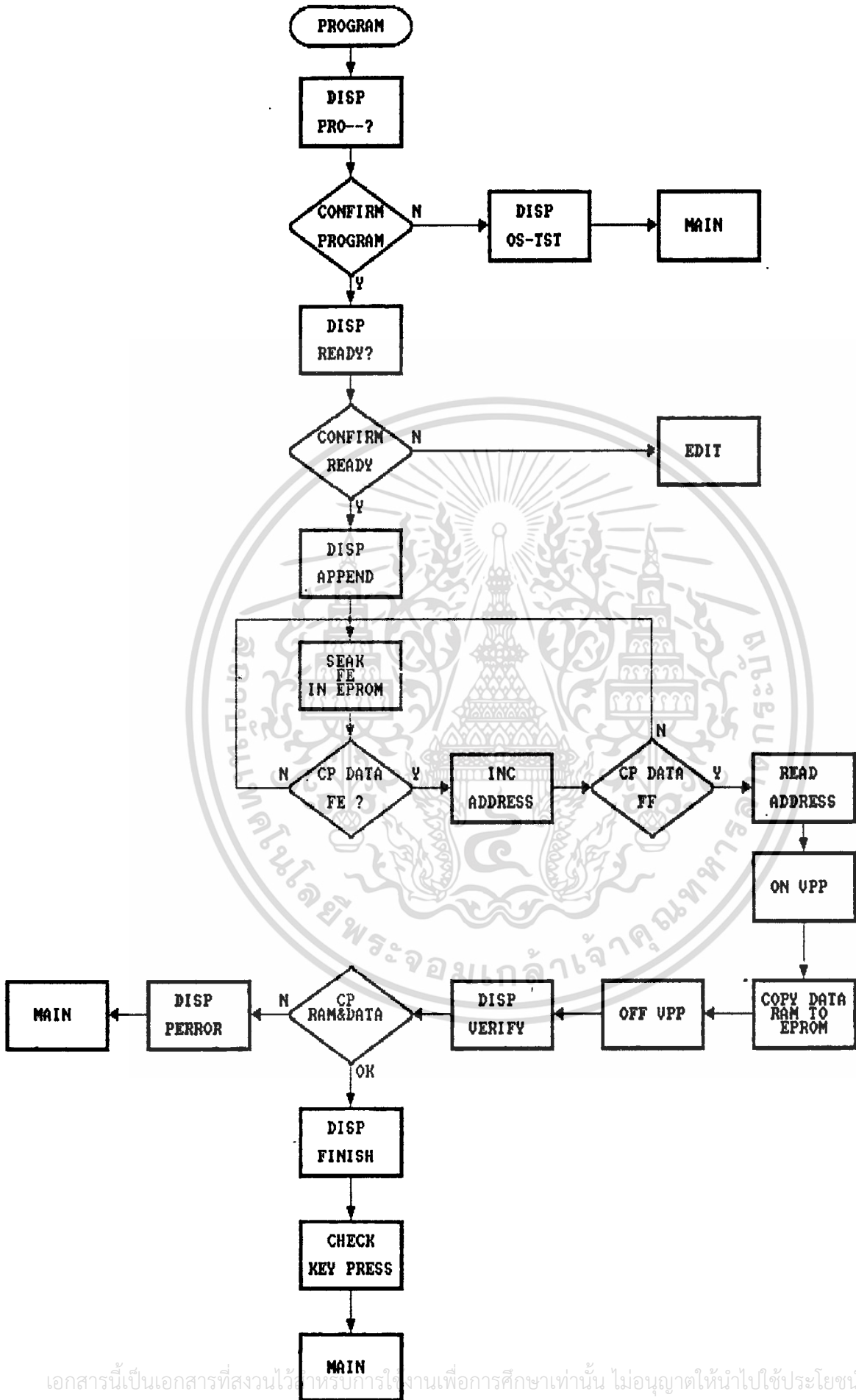
seek mode.

การทำงานของ seek mode ตาม flow chart. เมื่อมีการกด key "1" เครื่องจะมา run program seek mode โดยเริ่มด้วยการ display "see-?" เป็นการถามเพื่อความแน่ใจว่าต้องการเข้า mode seek จริงโดยถ้ากด key "dec" เป็นการออกจาก mode seek เครื่องจะ display "os-tst" แล้วออกจาก mode seek ไปแต่ถ้ากด key "inc" หมายถึงต้องการเข้า mode นี้จริงๆ เครื่องจะ display "code ?" ให้ใส่ code เพียง 2 หลักตั้งแต่ 01, 02... จนถึง code ตัวสุดท้ายที่เราใส่ไว้ code เหล่านี้เป็น code บอกที่อยู่ของ program ที่เราต้องการหาซึ่งเป็น program ที่ใช้ test ic. เบอร์ต่าง ๆ นั้นเองเมื่อเราใส่ตัวเลข 2 หลักแล้วเครื่องจะ display "code xx". เป็นการแสดงให้เห็นข้อมูลที่เรใส่เข้าไปจากนั้นถ้าเรากด key "inc" ในกรณีเราใส่ code ผิดหรือไม่ต้องการ code เบอร์นี้เครื่องจะวนกลับไปรอรับการกด key ใหม่โดยจะ display "code ?" เพื่อให้เราใส่ code ใหม่ตามที่เรต้องการแต่ถ้ากด key "dec" หมายถึงเรต้องการ code หมายเลขนั้นๆ แล้วเครื่องจะทำการค้นหา data ที่เป็น FE ซึ่งเป็น data ตัวสุดท้ายของการสิ้นสุดของการ test ของ code ทุกหมายเลขและจะอยู่เป็น address แรกก่อนถึง code เบอร์แรกคือ "01" ดังนั้นเมื่อเครื่องเจอ data FE แล้วเครื่องจะ inc ค่า address อีกหนึ่งค่าซึ่งเป็นตำแหน่งที่อยู่ของ code จากนั้นนำมา compare กับ code ที่เรต้องการถ้าไม่ใช่จะวนกลับไปหา data FE ใหม่ซึ่งจะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FE ของตัวถัดไปเป็นตัวสุดท้ายของ programe ของ code นั้นๆ(ให้ดูภาพ eeprom programe mapping ประกอบ) แล้ว inc address อีก 1 ค่าซึ่งจะเป็น address ที่อยู่ของ code ตัวถัดไปจากนั้นเอา code มา compare แล้วจะวนจนกว่า จะเจอถ้าเจอก็จะอ่าน ชื่อเบอร์ ic ออกมาแล้ว display เบอร์ ic ออกมาที่ display จากนั้นจะ copy ชื่อ และ data ต่างๆของ ic เบอร์นั้นไปเก็บไว้ที่ ram เมื่อเสร็จแล้วจะออกจาก mode ไป เป็นอันจบการทำงานของ mode seek.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

programme mode. 37.

เริ่มต้นด้วยการ display "pro-?" แล้วรอการกด key อีกที ถ้ากด key "dec" จะเป็นการออกจาก mode programme เครื่องจะ display "os-test" แล้วออกไปสู่ main programme แต่ถ้ากด key "inc" จะทำการ display "ready" เป็นการถามอีกครั้งถ้าพร้อมให้กด key ถ้ากด key "dec" แสดงว่ายังไม่พร้อมต้องการแก้ไข programme เครื่องจะออกไปยัง mode edit ให้เพื่อทำการแก้ไข programme ถ้ากด key "inc" เครื่องจะ display "append" แล้วเครื่องจะค้นหา data FE ซึ่งจะเป็น data ตัวสุดท้ายของ programme ถ้าหากว่าไม่เจอก็จะวนหาต่อไปและเปรียบเทียบค่าไปเรื่อยๆ ถ้าเจอจะทำการ inc address 1 ค่าแล้วจากนั้นทำการ compare ค่า FF ใน address นั้นซึ่งเป็นค่าที่ไม่ได้ programme ไว้ ถ้าไม่ใช่ค่า FF ก็จะวนกลับไปหาค่า FE ใหม่วนอยู่แบบนี้ ถ้าเจอ FF ก็จะทำกรอ่านค่า address ตัวนั้นออกมาแล้วสั่งให้ vpp ของ eprom ต่อกับไฟ เพื่อที่จะ programme จากนั้น copy ข้อมูลจาก ram มาเก็บใน eprom โดยเริ่มจาก address ที่เครื่องทำการอ่านออกมาเป็นต้นไป เมื่อเสร็จจะทำการ off vpp แล้ว display "verify" เป็นการ เปรียบเทียบข้อมูลที่ copy มาว่าใน ram กับใน eprom มีข้อมูลตรงกันหรือไม่ ถ้าไม่ตรงก็จะ display "error" แล้วออกสู่ main programme ถ้าข้อมูลเหมือนกันจะ display "finish" แล้ว check การกด key ถ้ามีการกด key จะออกไปยัง main programme ถ้าไม่มีการกด key จะ check key รอดต่อไปเรื่อยๆ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HOF "INT8"
CPU "Z80.TBL"
ORG 0000H
```

```

;--- VARIABLE SET ---
PPRN_DA: EQU 20H
PSEG7: EQU 21H
PDIGIT: EQU 22H
PCTRLKD: EQU 23H
CTRLWKD: EQU 88H
;
PDECODE: EQU 40H
PDAC: EQU 41H
PSIGTST: EQU 42H
PCTRLCD: EQU 43H
CTRLWCD: EQU 88H
PRC: EQU 42H
PBIN: EQU 0F0H
;
REP1: EQU 60H
REP2: EQU 16H
HIGFRE: EQU 10H
LOWFRE: EQU 14H
SLO_DLY: EQU 80H
MID_DLY: EQU 30H
FAS_DLY: EQU 15H
NOPRESS: EQU 0FFH
K_DE: EQU 06H
K_GO: EQU 07H
K_YES: EQU 01H
K_SE: EQU 06H

OUTDATA: EQU 0FFH
REPEAT: EQU 0DAH
;
ST_N_LD: EQU 2000H
ST_LD: EQU 2100H
ST_LD2: EQU 2200H
STARTPRN: EQU 2300H
;*****
DATADATE: EQU 26B0H
AD_NUMTK: EQU 26C0H
AD_BCD_D: EQU 26D0H
DISPBUF: EQU 26F0H
;
GAIN: EQU 26FBH
KEYIN: EQU 26FCH
MEMX: EQU 26FDH
REPDLY: EQU 26FEH
SYSFAG: EQU 26FFH
SYSSTK: EQU 27FFH
;
MONF: EQU 0C00H
RUN1: EQU 0C20H
MEM1: EQU 0C40H
STEP1: EQU 0C60H
REG: EQU 0C80H
NAMEP: EQU 3F00H
FTEST: EQU 3F06H
MASPROM EQU 4000H

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น หากต้องการนำเอกสารนี้ไปใช้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;--- POWER-UP DELAY -----
RESET:   XOR    A
RESO:   DEC    A
        NOP
        JR     NZ,RESO
        JP     START

;--- SYSTEM RESTART -----
        ORG   0008H
;RESTART 08H
;--- SYSTEM RESET -----
        ORG   0040H
START:  LD     HL,4400H
WAIT:   DEC    HL
        LD    A,H
        OR    L
        JR    NZ,WAIT
        LD    SP,SYSSTK ; SYSTEM STACK
START1: LD     E,30H
STDELAY: CALL  DELAY ; DELAY
        DEC   E
        JR   NZ,STDELAY

;--- INITIAL PORT -----
LD     A,CTRLWKD ; CONTROL PORT FOR KEY & DISP
OUT    (PCTRLKD),A ; A,B,C DN = OUT , C UP = IN
LD     A,CTRLWCD ; CONTROL PORT FOR DECODE & DAC
OUT    (PCTRLCD),A ; A,B,C DN = OUT , C UP = IN
IM     1

;--- SET PARAMETER OF SYSTEM FLAG -----
LD     A,1
LD     (SYSEAG),A ; SYSTEM FLAG = 1 (26FF)

;--- START POIWGNUM -----
LD     A,0FFH
LD     (KEYIN),A
JP     TITL

;----- SET OUTPUT PORT TO ZERO -----
XOR    A
OUT    (PPRN_DA),A
OUT    (PSEG7),A
OUT    (PDIGIT),A
OUT    (PDECODE),A
OUT    (PDAC),A
OUT    (PBIN),A
OUT    (PSIGTST),A

;--- DISPLAY "OS-TEST" -----
TITL:  LD     HL,TBLE_TITL
        LD     DE,DISPBUF
        LD     BC,6
        LDIR

SHOW:  CALL  SCAN ;--- WAIT KEYPRESS ---
        CP   K_SE ; NO ----> WAIT 'SE' KEY PRESS
        JP   NZ,SHOW ; YES --> RECEIVE IC CODE

```

เอกสารนี้เป็นเอกสารที่...
 ไม่ว่ากรณีใดๆ ทั้งสิ้น...
 2

```

STARTPG: LD HL, TBLE_NWT
          LD DE, DISPBUF
          LD BC, 6
          LDIR

```

```

;--- AND BLINK --> NO ? -----
          LD D, 3
BLINK9:  LD C, 20H ;10
LOOPYS:  CALL SCAN1CY
          DEC C
          JR NZ, LOOPYS ;--- BLINK DISPLAY ----
          XOR A
          OUT (PDIGIT), A
          LD E, 2
DELAY1:  CALL DELAY
          DEC E
          JR NZ, DELAY1
          DEC D
          JR NZ, BLINK9

```

```

;--- WAIT FOR KEY CODE -----
WAITJ:   LD B, 2
          LD IX, DISPBUF

```

```

;--- SCAND & SCANK & CONVERT KEYCODE TO SEGCODE -----
;OUTPUT : SEGCODE
;REGISTER: A, BC, DE, HL
NOKJ1:   PUSH BC
NOKJ2:   CALL SCAN ;
          LD A, (KEYIN)
          CP OFFH
          JR Z, NOKJ2

          LD A, (IX+1)
          LD (IX+0), A
          LD A, (IX+2)
          LD (IX+1), A
          LD A, (IX+3)
          LD (IX+2), A
          LD A, (IX+4)
          LD (IX+3), A
          LD A, (IX+5)
          LD (IX+4), A
          LD A, (KEYIN)
          CALL KEY_DISP
          LD (IX+5), A
          POP BC
          DJNZ NOKJ1 ;NOK 1
          LD C, 0
LOD:     CALL SCAN1CY
          DEC C
          JP NZ, LOD
          CALL PACK

```

```

;--- FINE START OF BLOCK OF IC -----
          LD IX, TBLE_IC
NEXTBOK: LD IY, DISPBUF
HASTART: LD A, (IX+0)
          CP OFEH
          JR Z, GERSTART
          INC IX
          JR HASTART
GERSTART: INC IX

```

LD A,(AD_NUMTK+2)

41.

CP (IX+0)
JR NZ,NEXTBOK
LD B,6
INC IX
SIBER: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ SIBER
LP1: CALL SCAN
CP K_DE
JP Z,STARTPG
CP K_GO
JP NZ,LP1

;--- DISPLAY "TEST" ---
SIGCK: LD A,00H
OUT (PDECODE),A
LD HL,TBLE_TEST
LD DE,DISPBUF
LD BC,6

LDIR
DOO: CALL SCANICY
IN A,(PSIGTST)
BIT 5,A
JR Z,DOO

CHIPAGIN: PUSH IX
HLDTSTP: PUSH IX
;---- START SEQUENTIAL TEST

SIGCHK: CALL SCANICY
IN A,(PSIGTST)
BIT 5,A
JR Z,SIGCHK

;----- SWITCH FOR ON MINUS -CURRENT FORCE

PINS: LD A,(IX+0) ;SWITCH ONLY PIN TO FORCE CURRENT
OR 00H ;BIT 7&6 = 0 FORCE -
OUT (PDECODE),A
CALL DELAY
IN A,(PSIGTST)
BIT 5,A
JR Z,PINS

;----- LLM -----
INC IX ;IX --> DATA LLM
LD A,(IX+0)
OUT (PDAC),A

;--- DELAY ---

LD E,1
DELAY2: CALL DELAY
DEC E
JR NZ,DELAY2
IN A,(PRC) ;ใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
BIT 6,A
JR NZ,REJECT1 ;ไม่ว่ากรณีใดๆ ทั้งสิ้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น

```

;----- HLM -----
INC IX ;IX --> DATA HLM
LD A,(IX+0)
OUT (PDAC),A
;-- DELAY -----
LD E,1
DELAY3: CALL DELAY
DEC E
JR NZ,DELAY3
IN A,(PRC)
BIT 6,A
JR Z,REJECT2
;----- SWITCH FOR ON PLUS +CURRENT FORCE -----
POP AF
OR 40H
OUT (PDECODE),A
;----- LLP -----
INC IX ;IX --> DATA LLP
LD A,(IX+0)
OUT (PDAC),A
;-- DELAY -----
LD E,1
DELAY4: CALL DELAY
DEC E
JR NZ,DELAY4
IN A,(PRC)
BIT 6,A
JR NZ,REJECT3
;----- HLP -----
INC IX ;IX --> DATA HLP
LD A,(IX+0)
OUT (PDAC),A
;-- DELAY -----
LD E,1
DELAY5: CALL DELAY
DEC E
JR NZ,DELAY5
IN A,(PRC)
BIT 6,A
JR Z,REJECT4
;--- CHECK END BLOCK ? -----LOOK "FE"-----
INC IX
LD A,(IX+0)
CP OFEH
JP NZ,HLDTSTP
JP TPASS
;--- DISPLAY "REJECT1" -----
REJECT1: POP AF
LD HL,TBLE_JEC1
LD DE,DISPBUF
LD BC,6
LDIR
LD A,9FH
OUT (PBIN),A
JP SIGCHK

```

```

;--- DISPLAY "REJECT2" -----
REJECT2: POP AF
          LD HL, TBLE_JEC2
          LD DE, DISPBUF
          LD BC, 6
          LDIR
          LD A, 0AFH
          OUT (PBIN), A
          JP SIGCHK

```

```

;--- DISPLAY "REJECT3" -----
REJECT3: LD HL, TBLE_JEC3
          LD DE, DISPBUF
          LD BC, 6
          LDIR
          LD A, 0B7H
          OUT (PBIN), A
          JP SIGCHK

```

```

;--- DISPLAY "REJECT4" -----
REJECT4: LD HL, TBLE_JEC4
          LD DE, DISPBUF
          LD BC, 6
          LDIR
          LD A, 0BBH
          OUT (PBIN), A
          JP SIGCHK

```

```

;--- DISPLAY "PASS" -----
TPASS:  LD HL, TBLE_PAS
          LD DE, DISPBUF
          LD BC, 6
          LDIR
          LD A, 7FH
          OUT (PBIN), A
          POP IX
          POP IX
          JP DOO

```

```

;=====
;*****
;--- CONVERT KEYCODE 1 BYTE TO DISPBUF -----
; INPUT & OUTPUT ! A -----
KEY_DISP: LD HL, SEGTAB
          ADD A, L
          LD L, A
          LD A, 0 ;*****
          ADC A, H ;*****
          LD H, A ;*****
          LD A, (HL)
          RET

```

```

;----- SCAN SUBROUTINE -----
SCAN:  LD B, 8 ; LOOP COUNT
        LD C, B ; RELEASE KEY COUNT
        LD E, 0 ; TABLE COUNT
        LD HL, DISPBUF ; DISPLAY BUFFER

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SCAN1:  CALL SCANS          ; SCAN DISPLAY
        LD  (MEMX),HL
        IN  A,(PDIGIT)    ; IN KEY
        AND 70H
        CP  70H
        JR  NZ,SCAN3      ; PRESS
        DEC  C
        JR  NZ,SCAN2
        LD  HL,SYSFAG
        RES 0,(HL)        ; IS RELEASE -> RESET PRESS FLAG
        LD  HL,REPDLY
        LD  (HL),REP1     ; DEFAULT REP1
SCAN2:  LD  HL,(MEMX)
        INC  E
        DJNZ SCAN1
        JR  SCAN

;KEY PRESS
SCAN3:  OR   E              ;OR WITH TABLE COUNT
        PUSH BC
        LD  HL,KEYTBL+22
        LD  B,16H
SCAN31: CP  (HL)           ; LOOKUP TABLE
        JR  Z,SCAN32
        DEC HL
        DJNZ SCAN31
SCAN32: LD  D,B
        POP  BC
        LD  HL,SYSFAG

        BIT 0,(HL)        ; PRESS FLAG
        JR  Z,SCAN5       ; NEW PRESS
        LD  A,7           ; CHECK AUTO REPEAT KEY
        CP  E
        JR  Z,SCAN4       ; INC DEC -> DOUBLE REPEAT
        LD  A,D
        CALL SCANC
        JR  C,SCAN2       ; NOT AUTO KEY
SCAN4:  LD  A,(KEYIN)     ; CHECK LAST KEYIN
        CALL SCANC
        JR  C,SCAN2       ; LAST KEYIN NOT AUTO

; AUTO REPEAT
        LD  HL,REPDLY    ; COUNT DOWN DELAY
        DEC (HL)
        JR  NZ,SCAN2
        LD  (HL),REP2
        RET

```

;--- NEW PRESS -----

```

SCAN5:  LD  A,D
        LD  (KEYIN),A
        SETT 0,(HL)      ; SET PRESS FLAG
        BIT 3,(HL)       ; BEEP FLAG
        JR  NZ,SCAN7
        LD  HL,400H      ; BEEP GENERATE
        LD  C,A
        LD  A,(GAIN)
        LD  B,A
        LD  A,C

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานส่งเสริมการค้าในต่างประเทศ
ไม่ว่ากรณีใดๆ ห้ามมิให้คัดลอกหรือใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD    C,HIGFRE
CP    10H
JR    NC,SCAN6
LD    C,LOWFRE
SCAN6:  NOP
        NOP
        NOP

```

;--- INSTANT KEY CHECK -----

```

SCAN7:  LD    A,(KEYIN)
        LD    HL,SYSFAG
        BIT   4,(HL)          ; INSTANT KEY FLAG
        RET   NZ              ; EXIT
        CP    10H
        JP    Z,MONF
        CP    12H
        JP    Z,RUN1
        CP    14H
        JP    Z,MEM1
        CP    15H
        JP    Z,STEP1
        CP    16H
        JP    Z,REG
        RET

```

;--- AUTO REPEAT KEY CHECK SUB. -----

```

SCANC:  CP    11H
        RET   Z
        CP    13H
        RET   Z
        CP    15H
        RET   Z
        SCF
        RET

```

;--- SCANS DISPLAY SUB -----
; MUST USE

FIRST: HL --> DISPLAY BUFFER

```

SCANS:  LD    A,E
        OUT  (PDIGIT),A
        LD  A,(HL)
        OUT  (PSEG7),A
        XOR  A
SCAN1:  DEC  A
        JR   NZ,SCAN1
        OUT  (PSEG7),A
        INC  HL
        RET

```

;--- SCAN 1 CYCLE NO CHK KEY -----
; REGISTER A,B,E,HL

```

SCAN1CY: LD    B,6          ; LOOP COUNT
          LD    E,0          ; START DIGIT TO DISPLAY
          LD    HL,DISPBUF   ; DISPLAY BUFFER

```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทางที่มีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOOP1CY: CALL SCANS          ; SCAN DISPLAY
          INC E
          DJNZ LOOP1CY
          RET

```

```

;--- SCAND SUB -----
; INPUT      B (DELAY)
; REGISTER   A,B,DE,HL
;
SCAND:    LD  DE,800H
          LD  HL,DISPBUF
SCAND1:   CALL SCANS
          INC E
          DEC D
          JR  NZ,SCAND1
          DJNZ SCAND
          RET

```

```

;--- SCANK SUBROUTINE -----
; INPUT      DISPBUF
; OUTPUT     (KEYIN)
;
SCANK:    LD  DE,800H
          LD  HL,DISPBUF
          LD  A,0FFH
          LD  (KEYIN),A

```

```

SCANK1:   CALL SCANS
          IN  A,(PDIGIT)
          AND 70H
          CP  70H
          JR  Z,SCANK2
          OR  E
          PUSH BC
          PUSH DE
          PUSH HL
          LD  HL,KEYTBL+22
          LD  B,16H

```

```

SCANK11:  CP  (HL)
          JR  Z,SCANK12
          DEC HL
          DJNZ SCANK11

```

```

SCANK12:  LD  A,B
          POP HL
          POP DE
          POP BC
          LD  (KEYIN),A

```

```

SCANK2:   INC E
          DEC D
          JR  NZ,SCANK1
          RET

```

```

;--- BLANK DISPLAY BUFFER -----
; REGISTER   A,B,HL

```

```

BNK_DISP: LD  HL,DISPBUF
          LD  B,6
          XOR A

```

งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUTO:   LD   (HL),A
        INC  HL
        DJNZ PUTO
        LD   HL,DISPBUF
        RET

```

```

;--- DELAY 1 mSEC. SUBROUTINE -----
;REGISTER B,C

```

```

DELAY:  LD   B,50H
LOOPD2: LD   C,0DDH
LOOPD1: DEC  C
        JR   NZ,LOOPD1
        DJNZ LOOPD2
        RET

```

```

;--- CONVERT HEX,BCD TO DISPLAY [ 3 BYTE --> 6 BYTE ] -----
; INPUT! ADDRESS 26E0H - 26E2H ----- ( HEXDISP1 )-----
; INPUT! ADDRESS 26E8H - 26EAH ----- ( HEXDISP2 )-----
; OUTPUT! ADDRESS --> DISPLAY BUFFER -----
; REGISTER A,B,C,DE,HL -----

```

```

HEXDISP1: LD   HL,AD_BCD_D+1
HEXDISP2: LD   DE,DISPBUF

```

```

        CALL UNPKS1
        DEC  HL
        CALL UNPKS1
        INC  HL
        INC  HL

```

```

UNPKS1:  LD   B,2

```

```

UNPKS2:  XOR  A
        RLBCD
        PUSH HL
        CALL UNPKT1
        POP  HL
        LD   (DE),A
        INC  DE
        DJNZ UNPKS2
        RET

```

```

UNPKT1:  LD   HL,SEGTAB
        ADD  A,L
        LD   L,A
        LD   A,0
        ADC  A,H
        LD   H,A
        LD   A,(HL)
        RET

```

```

;--- CONVERT HEX,BCD TO ASC [ 3 BYTE --> 6 BYTE ] -----
; OUTPUT! ADDRESS --> ASC -----
; REGISTER A,B,C,DE,HL -----

```

```

HEXDISP3: CALL UNPKS3
        DEC  HL
        CALL UNPKS3
        INC  HL
        INC  HL

```

```

UNPKS3:  LD   B,2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UNPKS4:  XOR  A
          RLBCD
          PUSH HL
          CALL UNPKT2
          POP  HL
          LD   (DE),A
          INC  DE
          DJNZ UNPKS4
          RET

```

```

UNPKT2:  LD   HL,ASCTAB
          ADD  A,L
          LD   L,A
          LD   A,0
          ADC  A,H
          LD   H,A
          LD   A,(HL)
          RET

```

```

;--- PACK SUBROUTINE -----
; INPUT : DISPLAY BUFFER
; OUTPUT:
; REGISTER : A,BC,DE,HL
;--- CONVERT DISPLAY BUFF [ 6 BYTE --> 3 BYTE HEX ] -----
; INPUT | ADDRESS --> DISPLAY BUFFER -----
; OUTPUT| AD_NUMTK --> 26C0 - 26C2 -----
; REGISTER A,B,C,DE,HL -----

```

```

PACK:    LD   DE,DISPBUF
          LD   HL,AD_NUMTK+1
          CALL PACKS
          DEC  HL
          CALL PACKS
          INC  HL
          INC  HL
PACKS:   LD   C,2
PACKS1:  LD   A,(DE)
          PUSH HL
          CALL PACKT
          POP  HL
          RLBCD
          INC  DE
          DEC  C
          JR   NZ,PACKS1
          RET
PACKT:   LD   HL,SEGTAB+15
          LD   B,0FH
PACKT1:  CP   (HL)
          JR   Z,PACKT2
          DEC  HL
          DJNZ PACKT1
PACKT2:  LD   A,B
          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;--- CONVERT BCD TO DISPLAY -----
; REGISTER : A,B,C,D,HL,IX,IY
; INPUT   : (HL+00H - HL+05H) KEEP DATA THAT WANT TO CHANGE
; OUT     : DISPLAY BUFFER

```

```

;AD_BCD_D EQU 26E0H
BCD_DISP: LD D,6
          LD IY,DISPBUF-1
          LD HL,AD_BCD_D
LOOP:     LD IX,TBLEB_D
          LD B,0
          LD A,(HL)
          CP 10
          JR NC,RESULT
          LD C,A
          ADD IX,BC
          LD B,(IX+0)
RESULT:   INC HL
          INC IY
          LD (IY+0),B
          DEC D
          JR NZ,LOOP
          RET

```

```

;--- DELAY 1 uS -----
D10:     LD A,15H
D10_1:   DEC A
          JR NZ,D10_1
          RET

```

```

;--- DELAY .5 uS -----
D05:     LD A,10H
D05_1:   DEC A
          JR NZ,D05_1
          RET

```

```

;--- TABLE CONTENT 0-9 (7 SEG CODE) -----
;--- KEY TABLE IN ROM -----
TBLEB_D: DFB 3FH,06H,5BH,4FH,66H

```

```

;-----
KEYTBL:  DFB 60H,61H,62H,63H,50H,51H ;0,1,2,3,4,5
          DFB 52H,53H,30H,31H,32H,33H ;6,7,8,9,A,B
          DFB 35H,55H,65H,36H,64H,67H ;C,D,E,F,10,11
          DFB 54H,57H,34H,36H,66H ;12,13,14,15,16

```

```

;-----
ASCTAB:  DFB 30H,31H,32H,33H,34H,35H ;0,1,2,3,4,5
          DFB 36H,37H,38H,39H,41H,42H ;6,7,8,9,A,B
          DFB 43H,44H,45H,46H,20 ;C,D,E,F,BLANK

```

```

;-----
SEGTAB:  DFB 3FH,06H,5BH,4FH,66H,6DH ;0,1,2,3,4,5
          DFB 7DH,07H,7FH,6FH,77H,7CH ;6,7,8,9,A,B
          DFB 39H,5EH,79H,71H ;C,D,E,F=71

```

```

;-----
HXHXTAB: DFB 00H,01H,02H,03H,04H,05H ;0,1,2,3,4,5
          DFB 06H,07H,08H,09H,5CH,5CH ;6,7,8,9,A,B
          DFB 5CH,5CH,5CH,5CH ;C,D,E,F ทุกครั้งที่มีการนำไปใช้

```

```

;
TBLE_TITL: DFB 3FH,0EDH,40H,78H,6DH,78H ; SHOW OS-TST
TBLE_HERE: DFB 40H,76H,79H,50H,79H,00H ; SHOW HERE
TBLE_BANK: DFB 00H,00H,00H,00H,00H,00H ; SHOW BLANK
TBLE_NWT: DFB 00H,54H,3FH,08H,53H,00H ; SHOW NO ?
TBLE_YES: DFB 00H,6EH,79H,6DH,80H,53H ; SHOW YES ?
TBLE_TEST: DFB 00H,78H,79H,6DH,78H,00H ; SHOW TEST
TBLE_PAS: DFB 00H,73H,77H,6DH,6DH,00H ; SHOW PASS
TBLE_JEC1: DFB 50H,79H,0EH,79H,39H,06H ; SHOW REJEC1
TBLE_JEC2: DFB 50H,79H,0EH,79H,39H,5BH ; SHOW REJEC2
TBLE_JEC3: DFB 50H,79H,0EH,79H,39H,4FH ; SHOW REJEC3
TBLE_JEC4: DFB 50H,79H,0EH,79H,39H,66H ; SHOW REJEC4

```

```

;
TBLE_IC: DFB 0FEH,01H,07H,66H,38H,6DH ; ST, CODBER, 7, 4, L, S
DFB 6FH,06H,00H,90H,0F9H,70H ; 91, PIN1, 3, 5, 7
DFB 90H ; 9
DFB 0FEH,02H,07H,66H,38H,6DH ; ST, CODBER, 7, 4, L, S
DFB 6FH,5BH,01H,04H,06H,08H ; 92, PIN2, 4, 6, 8
DFB 10H ; 10
DFB 0FEH,03H,07H,66H,38H,6DH ; ST, CODBER, 7, 4, L, S
DFB 6FH,4FH,02H,03H,05H,07H ; 93, PIN3, 3, 5, 7
DFB 09H ; 9
DFB 0FEH,04H,07H,66H,38H,6DH ; ST, CODBER, 7, 4, L, S
DFB 6FH,66H,03H,04H,06H,08H ; 94, PIN4, 4, 6, 8
DFB 10H ; 10
DFB 0FEH

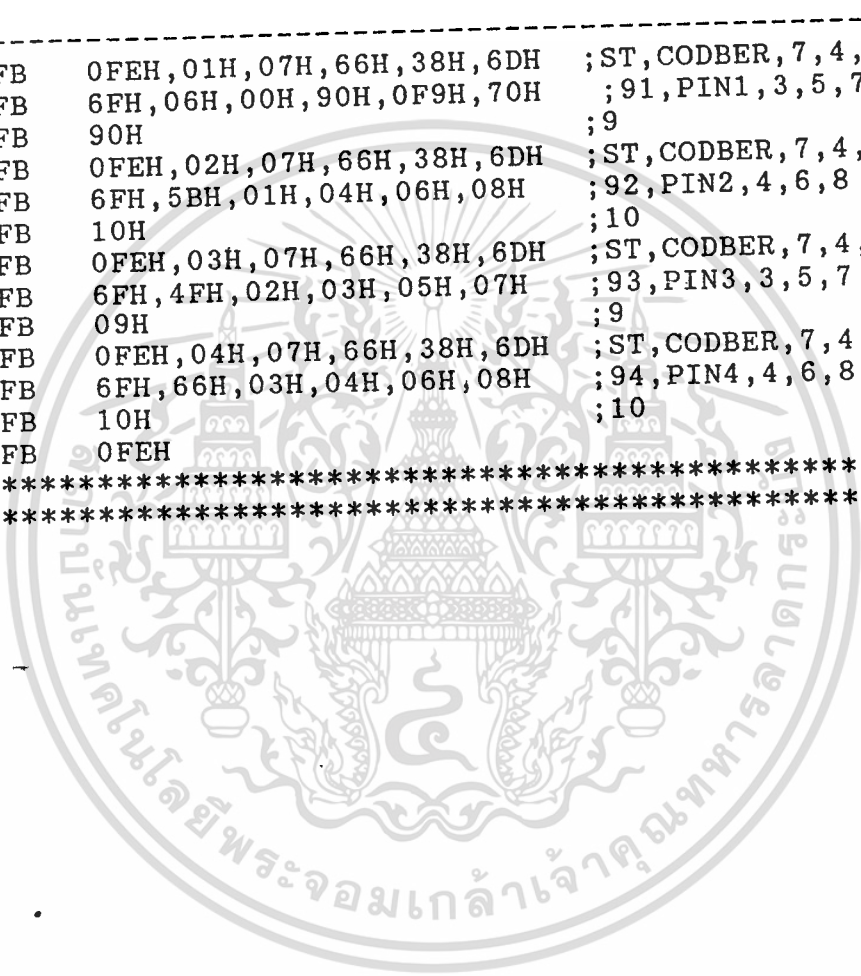
```

```

;*****
;*****

```

END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปของระบบควบคุมด้วยไมโครโปรเซสเซอร์กับ Z80 CP-A

คืออะไร

คือบอร์ดไมโครโปรเซสเซอร์ขนาดเล็ก ที่ประกอบด้วยระบบพื้นฐานในการทำงาน เช่น CPU หน่วยความจำ รวมทั้งการจัด Decode ไว้อย่างเรียบร้อย และยังมี I/O Port สำหรับการต่อกับอุปกรณ์ Input Output ภายนอกให้อีก

ทำอะไรได้บ้าง

Z80 CP-A คือโครงสร้างทาง Hardware อย่างเดียว ซึ่งจะต้องมีการพัฒนาโปรแกรม เพื่อให้มันทำงานตามที่ต้องการ โดยจะขึ้นอยู่กับลักษณะของงานนั้นๆที่นำไปใช้ โปรแกรมนี้เองที่จะเป็นตัวกำหนดความสามารถของ Z80 CP-A ซึ่งทำให้มีความจำกัดน้อยมาก เพราะมันจะทำงานให้คุณตามแต่จินตนาการอันกว้างไกลของคุณ

เหมาะกับใคร

สำหรับผู้ที่จะทำโครงการต่างๆทางด้านอิเล็กทรอนิกส์ คอมพิวเตอร์ และระบบควบคุมอุตสาหกรรม ซึ่งจะช่วยลดขั้นตอนในการออกแบบ การทำเครื่องต้นแบบ รวมทั้งการผลิต และเนื่องจากเป็นบอร์ดที่ออกแบบมาโดยเฉพาะ จึงทำให้มีเสถียรภาพในการทำงานสูง ในขณะที่เครื่องต้นแบบทั่วไปมักจะมีปัญหาเรื่องสัญญาณรบกวน เนื่องจากจุดต่อต่างๆ

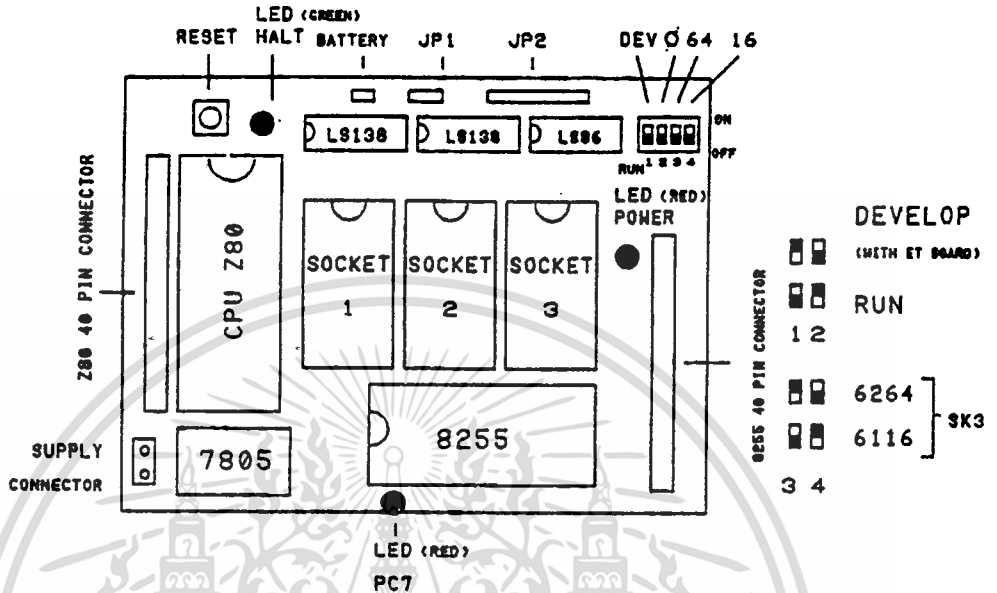
ยากหรือไม่

ถ้าผู้ที่มีความเข้าใจทางด้านไมโครโปรเซสเซอร์อยู่แล้ว ก็สามารถทำได้ไม่ยากเลย แต่กรณีผู้ที่เริ่มต้นใหม่ก็ใช้เวลาในตอนต้นมากหน่อย อย่างไรก็ตาม ทีมงานอีกทีที่พร้อมที่จะสนับสนุนการให้คำปรึกษาและวิชาการแก่กลุ่มลูกค้าเสมอ จึงทำให้มีความมั่นใจได้ในโครงการที่จะพัฒนาขึ้น

คุ้มค่าแค่ไหน

จริงอยู่ งานบางอย่างก็สามารถใช้วงจรอิเล็กทรอนิกส์ทำงานได้ โดยไม่จำเป็นต้องใช้ไมโครโปรเซสเซอร์ แต่อย่างไรก็ตามเมื่อเทียบกับความเร็วในการพัฒนาและราคาที่เหมาะสมแล้ว วิธีนี้เป็นการลงทุนที่คุ้มค่าและให้ผลเป็นที่น่าพอใจ ที่สำคัญการทัวโครงการด้านไมโครโปรเซสเซอร์จะมีการปรับปรุงแก้ไขเพิ่มเติมได้ง่าย โดยการกระทำทางโปรแกรม ซึ่งจะเป็นเรื่องยากถ้าเป็นวงจรตายตัว จึงเหมาะสมเป็นอย่างยิ่งที่คุณจะฝากโครงการชิ้นสำคัญของคุณไว้กับ Z80 CP-A

ภาพแสดงลักษณะ Z80 CP-A



- Socket 1 ใช้กับเบอร์ 2732, 2764, 6264
- Socket 2 ใช้กับเบอร์ 6264 โดยจะสามารถ Backup ข้อมูลได้ ถ้าต่อ Battery
- Socket 3 ใช้กับเบอร์ 6116, 6264 โดยเลือกได้จาก Dip Switch ตัวที่ 3, 4
- JP 1 สำหรับเลือกสถานะของ Socket 1 โดยถ้าต่อกับ WR (62) จะใช้กับ 6264 และ 2732 แต่ถ้าต่อกับ Vcc (27) จะใช้กับ 2764
- JP 2 สำหรับเลือกเบอร์ Port ของ 8255
- Dip 1 สำหรับเลือกโหมดในการใช้งาน Dev. คือโหมดในขณะทำการพัฒนา Run คือโหมดสำหรับการใช้งาน
- Dip 2 สำหรับตัดหรือต่อความถี่ Clock ปกติจะใช้ร่วมกับตัวที่ 1 คือถ้าขณะทำการพัฒนาจะ off ไว้
- Dip 3 เลือก Socket 3 ให้ใช้กับ 6264
- Dip 4 เลือก Socket 4 ให้ใช้กับ 6116
(ตัวที่ 3, 4 นี้จะ On ตัวใดตัวหนึ่งเท่านั้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

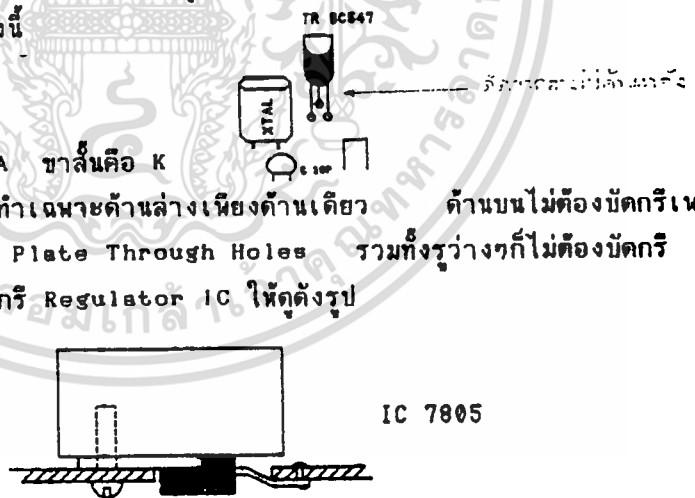
ในโหมด Dev นี้ ผู้ใช้จะเห็น Address ในตำแหน่ง 8000H A000H C000H ตามลำดับ และถ้านำ Ram 6264 มาเสียบที่ Socket 1 ก็จะสามารถทำโปรแกรมได้เลย โดยใช้การทำงานต่างๆทั้งหมดของอิทีบอร์ด การนำ Ram 6264 มาเสียบ จะต้องเปลี่ยน JP1 ให้อยู่ในตำแหน่ง 62 ด้วย ซึ่งจะต้องตัดลสายปรีนที่ Default ไว้ที่ตำแหน่ง 27 ก่อน แล้วจึงต่อสายใหม่

การนำไปใช้งาน

1. ทำการปรับ Dip switch ตัวที่ 1 ให้อยู่ในตำแหน่ง Run และตัวที่ 2 ให้อยู่ในตำแหน่ง On
2. ใส่ตัว CPU Z80 ลงใน Socket ให้เรียบร้อย
3. นำโปรแกรมที่พัฒนาเรียบร้อยแล้วลงในตัว Eprom โดยใช้เครื่อง ET Burner หรือจะใช้เครื่องอื่นๆก็ได้ โดยโปรแกรมนี้จะต้องแปลงให้ทำงานที่ Address 0000H จากนั้นก็นำมาใส่ลงบนบอร์ดที่ Socket 1
4. ถ้ามีการแก้ไขที่ JP1 ในขณะพัฒนา ก็ต้องเปลี่ยนกลับเป็นตำแหน่ง 27 เหมือนเดิม

คำแนะนำในการประกอบ

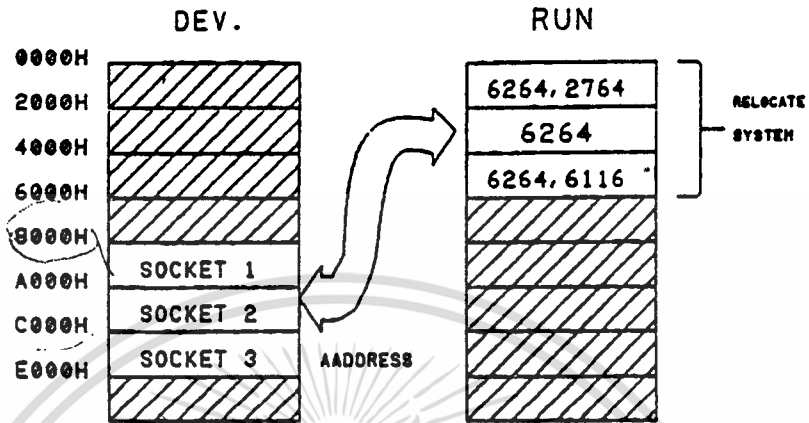
1. ควรใส่อุปกรณ์ตามลำดับคือ ตัวต้านทาน, ไดโอด, ตัวเก็บประจุ, Socket
2. ตัวเก็บประจุและไดโอดต้องใส่ให้ถูกขั้ว รวมทั้งทรานซิสเตอร์ต้องใส่ขาให้ถูก โดยให้ดูจากรูปดังนี้
3. LED ขาวคือ A ขาสีคือ K
4. การบัดกรีให้กระทำเฉพาะด้านล่างเพียงด้านเดียว ด้านบนไม่ต้องบัดกรีเพราะแผ่นปรีนเป็นแบบ Plate Through Holes รวมทั้งรูต่างๆก็ไม่ต้องบัดกรี
5. การติดตั้งและบัดกรี Regulator IC ให้ดูดังรูป



6. เมื่อเรียบร้อยแล้วจึงใส่ IC ลงใน Socket โดยต้องระวังไม่ให้ผิดเบอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดการจัดหน่วยความจำและ Port



การจัดหน่วยความจำของ CP-A นี้จะอ้าง Address เริ่มต้นที่ 8000H เพื่อให้ทำโปรแกรมได้สะดวกจากเครื่องอิทบอร์ด และเมื่อจะนำโปรแกรมไปใช้งาน CP-A มีระบบ Relocate เพื่อให้หน่วยความจำไปเริ่มต้นที่ 0000H ได้ (ในโหมด Run) เพื่อให้เป็น Address ที่ทำงานเมื่อเริ่มเปิดเครื่องหรือกด Reset

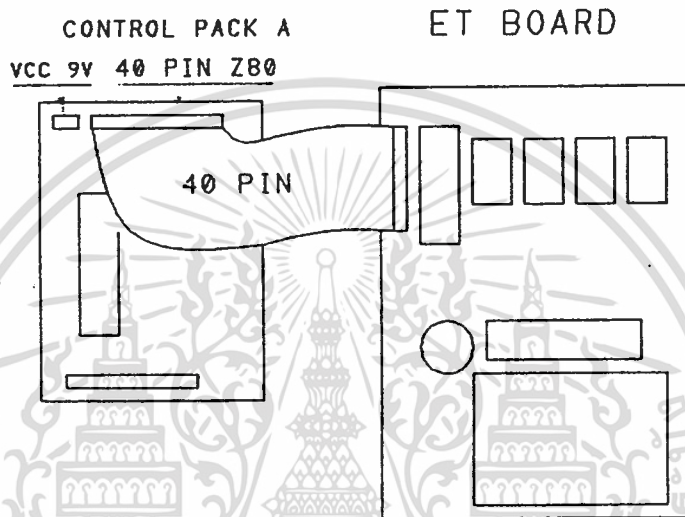
ระบบ Port จะทำการ Decode ไว้ทั้งหมด 8 ช่วง และสามารถเลือกเบอร์ได้ 7 ตำแหน่ง โดยการ Jump สายบนแผ่นปรีน ซึ่งมีเบอร์ Port ดังนี้

ตำแหน่ง	เบอร์ Port
1	00 - 1F (จริง ๆ ใช้เพียง 00-03)
2	20 - 3F
3	60 - 7F
4	80 - 9F
5	A0 - BF
6	C0 - DF
7	E0 - FF

ส่วนเบอร์ Port 40 - 5F นั้น ไว้สำหรับการขยายได้อีกทางด้าน 8255 Connector

การใช้งานร่วมกับอิทิบอร์ด

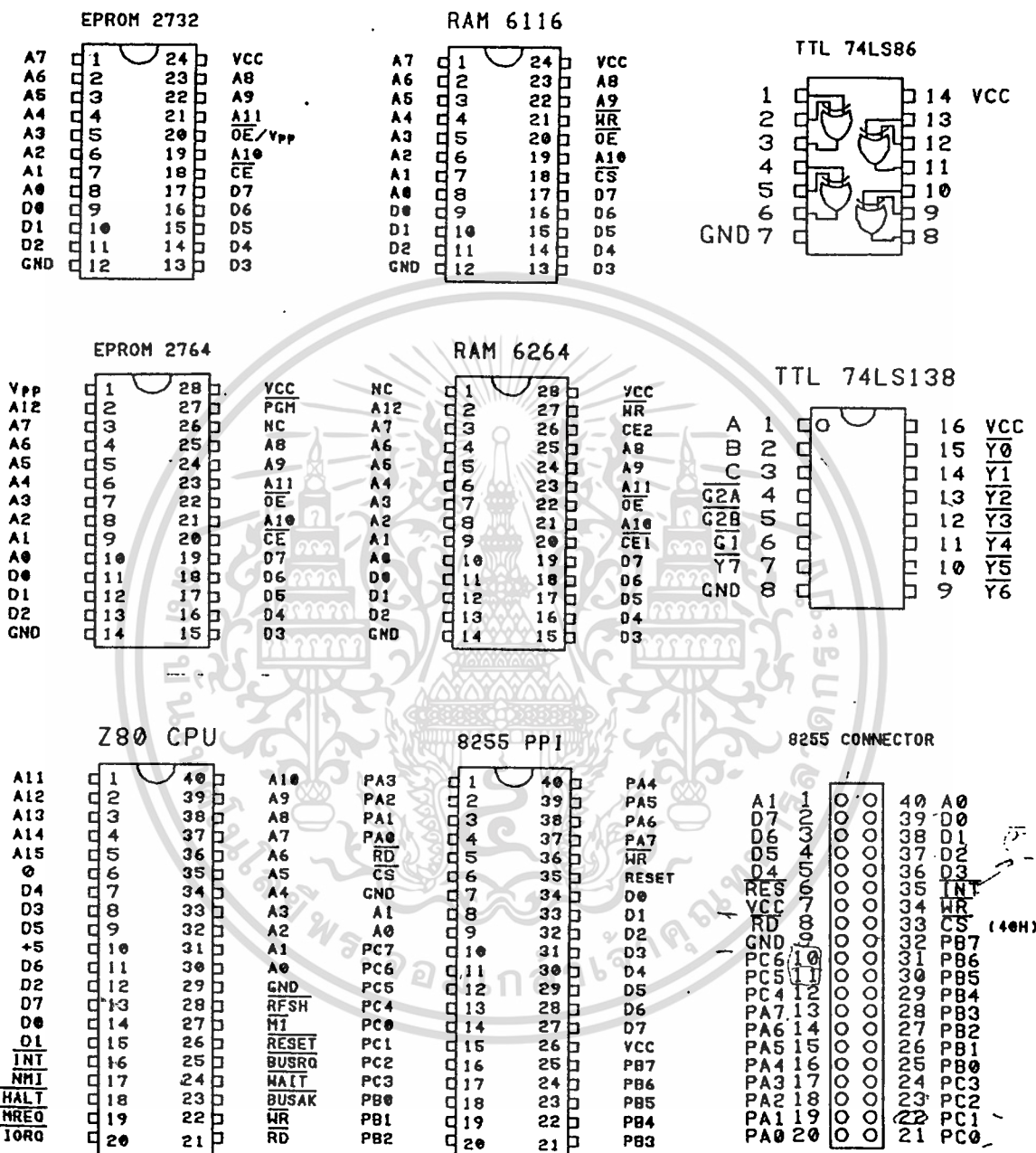
Z80 CP-A มีโหมดในการทำงาน 2 ลักษณะคือ การพัฒนาและการนำไปใช้งาน ในขั้นของการพัฒนานั้นจะทำได้โดยเสียบเข้ากับอิทิบอร์ดได้เลย หรือจะทำโดยใช้เครื่องอื่นาก็ได้ แล้วจึงนำโปรแกรมมาทำงานโดยผ่าน Eprom หรือ Ram-pack ซึ่งผู้ใช้จะต้องเขียนโปรแกรมโดยอ้าง Address ตามลักษณะที่กำหนดไว้ ในการเสียบกับอิทิบอร์ดนั้นจะแสดงได้ดังภาพต่อไปนี้



ซึ่งพอจะสรุปขั้นตอนได้ดังนี้

1. ทำการปรับ Dip switch ตัวที่ 1 ให้อยู่ในตำแหน่ง Dev. และตัวที่ 2 ให้อยู่ในตำแหน่ง Off ซึ่งเป็นการตัดความถี่ Clock ให้เหลือเพียง Clock จากอิทิบอร์ด
2. ผู้ใช้จะต้องเอา CPU Z80 ตัวใดตัวหนึ่งออกจากบอร์ด คือบอร์ด CP-A หรือบอร์ดอิทิบอร์ดก็ได้ เพื่อที่จะได้ไม่ซ้อนกัน
3. แหล่งจ่ายไฟ 5V ของทั้งสองบอร์ดจะต่อกันหมด และเนื่องจากใช้กระแสมากขึ้น ผู้ใช้ควรจะใช้ Adapter ที่มีขนาดใหญ่พอสมควร (ของที่มาพร้อมกับเครื่องอิทิบอร์ดจะไม่พอ) และก็เสียบกับบอร์ดใดบอร์ดหนึ่งได้เลย ทั้งนี้ถ้ามีการต่อ Port ออกไปใช้งาน ก็ต้องดูด้วยว่ากระแสจ่ายได้เพียงพอหรือไม่ ถ้าไม่พอก็ควรแยกแหล่งจ่ายไฟออกเป็นอีกชุดหนึ่ง

รายละเอียดขา IC และ Connector



pcv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specification

CPU	Z-80A
Memory	6264/2764/2732 Socket (8 Kbyte at 8000H) 6264 Ram (8 Kbyte at A000H) & Backup Circuit 6264/6116 Socket (8 Kbyte at C000H)
Port	8255 I/O Port (8*3 bit)
LED	1 Power Red LED 1 PC7 Port Red LED 1 Halt Green LED
Diode	2 1N60
Transistor	1 BC547
TTL IC	3 74LS86 74LS138*2
Regulator IC	1 7805
Connector	1 40-Pin Expansion Header-Strip (Z80 Pin) 1 40-Pin Peripheral Header-Strip (8255 Port) 1 2-Pin For DC 9V Supply
Switch	4 Dip-Switch (Run/Develop, Clock-On, 6116, 6264 Select)
Clock Rate	3.579545 MHz
Power Supply	Consumption 5V DC Main Input 9-12V DC
PCB Size	8.5 * 12 cm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์

ชุด IC

1 ตัว	Z-80A CPU	1 ตัว	8255 I/O port
1 ตัว	6264 Ram	1 ตัว	74LS86
2 ตัว	74LS138		

ชุด Socket

2 ตัว	40-Pins Socket	3 ตัว	28-Pins Socket
3 ตัว	16-Pins Socket		

ชุด Resistor

8 ตัว	10 K	2 ตัว	1 K
1 ตัว	4.7 K	1 ตัว	220
2 ตัว	330	1 ตัว	680

ชุด Condensor

1 ตัว	220 uf ELE	1 ตัว	33 uf ELE
1 ตัว	22 uf ELE	1 ตัว	4.7 uf TAN
1 ตัว	1 uf TAN	6 ตัว	0.1 uf TAN
1 ตัว	0.01 uf ไมลาร์	1 ตัว	10 pf CER

ชุดเบ็ดเตล็ด

3 ตัว	LED (แดง 2 ตัว , เขียว 1 ตัว)		
2 ตัว	1N60	1 ตัว	1N4148
1 ตัว	7805	1 ตัว	Head Sink พร้อมน็อต
2 ตัว	Connector 40 Pin	1 ชุด	ขั้วต่อ Supply
1 ตัว	Dip Switch (4*)	1 ตัว	Reset Switch
1 ตัว	X'TAL 3.579545 MHz	1 ตัว	BC 547

อื่นๆ

1 แผ่น	CP-A PCB
1 เล่ม	ชีทรายละเอียดการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้