สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

ปริญญานิพนธ์ ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

8088 SINGLE BOARD MICROCOMPUTER

T030117

นักศึกษา 1.นาย ชัยสิทธิ์ พลัดพริ้ง

2.นาย ดุสิต พันธุ์พฤกษ์

3.นาย วิเชษฐ ยิ้มละมัย

4.นาย สมบัติ หทัยรัตนานนท์

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ผู้ควบคุมปริญญานิพนธ์

1.อาจารย์ปิยะ จิตธรรมมาภิรมย์

2.อาจารย์อำพล ทองระอา

3.อาจารย์ไพบูลย์ พวงวงศ์ตระกูล

คณะกรรมการสอบปริญญานิพนธ์	ลายมือชื่อ
1.อาจารย์กิติพงค์ มะโน	
2.อาจารย์วรวิทย์ สมหา	
3.อาจารย์ปิยะ จิตธรรมมาภิรมย์	
4.อาจารย์อำพล ทองระอา	
5.อาจารย์ไพบูลย์ พวงวงศ์ตระกูล	

วันเดือนปีที่สอบ วันที่ 12 ธันวาคม 2540 เวลา 19.30 น. ถึง 20.30 น.

สถานที่สอบ ห้อง ค 310 คณะครุศาสตร์อุตสาหกรรม

ภาควิชารับรองแล้ว

.....

(ผู้ช่วยศาสตราจารย์ ธีรพล เทพหัสดิน ณ อยุธยา)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 25 เดือน ก.ค. พ.ศ. 41

เลขหมู่.....
เลขทะเบียน 30117
วัน, เดือน, ปี 8 ส.ค. 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช

ปริญญานิพนธ์

ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

8088 SINGLE BOARD MICROCOMPUTER

นาย ชัยสิทธิ์ พลัดพริ้ง

นาย ดุสิต พันธุ์พฤกษ์

นาย วิเชษฐ ยิ้มละมัย

นาย สมบัติ หทัยรัตนานนท์

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบันทิต

สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

8088 SINGLE BOARD MICROCOMPUTER

ผู้จัดทำ

1. นาย ชัยสิทธิ์ พลัดพริ้ง
2. นาย ดุสิต พันธุ์พฤกษ์
3. นาย วิเชษฐ ยิ้มละมัย
4. นาย สมบัติ หทัยรัตนานนท์

อาจารย์ที่ปรึกษา

ลงนาม.....

(อาจารย์ปิยะ จิตธรรมมาภิรมย์)

ลงนาม.....

(อาจารย์อำพล ทองระอา)

ลงนาม.....

(อาจารย์ไพบูลย์ พวงวงศ์ตระกูล)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....

(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

8088 SINGLE BOARD MICROCOMPUTER

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของไมโครโปรเซสเซอร์ 8088
2. เพื่อศึกษาการอินเตอร์เฟสของอุปกรณ์อิเล็กทรอนิกส์ร่วมกับไมโครโปรเซสเซอร์ 8088
3. เพื่อออกแบบแผ่นพิมพ์เดี่ยวของไมโครโปรเซสเซอร์ 8088
4. เพื่อจัดสร้างแผ่นพิมพ์เดี่ยวและเขียนโปรแกรมควบคุมระบบ
5. เพื่อเป็นแนวทางในการศึกษาและพัฒนาความรู้ด้านไมโครโปรเซสเซอร์ 8088

ประโยชน์ที่คาดว่าจะได้รับ

1. ความรู้และเข้าใจในโครงสร้างและการทำงานของไมโครโปรเซสเซอร์ 8088
2. ความรู้และความเข้าใจในการอินเตอร์เฟสอุปกรณ์อิเล็กทรอนิกส์ ร่วมกับไมโครโปรเซสเซอร์ 8088
3. ออกแบบแผ่นพิมพ์เดี่ยวของไมโครโปรเซสเซอร์ 8088 ได้
4. สร้างแผ่นพิมพ์เดี่ยวและเขียนโปรแกรมควบคุมระบบได้
5. นำไมโครโปรเซสเซอร์มาประยุกต์ใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

นาย ชัยสิทธิ์ พลัดพริ้ง
นาย ดุสิต พันธุ์พฤกษ์
นาย วิเชษฐ ยิ้มละมัย
นาย สมบัติ หทัยรัตนานนท์

อาจารย์ที่ปรึกษา
อาจารย์ปิยะ จิตธรรมมาภิรมย์
อาจารย์อำพล ทองระอา
อาจารย์ไพบูลย์ พวงวงศ์ตระกูล

ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอระบบไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 โดยออกแบบระบบให้สามารถทำงานได้ที่ความถี่ 5 MHz มีหน่วยความจำ 64 KB หน่วยรับข้อมูลเป็นสวิตช์เมทตริก์ 6 x 4 และหน่วยแสดงผลแบบจอแสดงผลผลึกเหลวนอกจากนี้ได้ออกแบบโปรแกรมควบคุมไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 ให้มีความพร้อมสำหรับการรับและแสดงผลข้อมูลในการประมวลผลที่ซับซ้อนได้ จากผลการทดลองใช้งานระบบไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 ดังกล่าวปรากฏให้ผลการทำงานสอดคล้องกับหลักการนำเสนอไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8088 SINGLE BOARD MICROCOMPUTER

MR.CHAISIT PARDPRING

MR.DUSIT PUNPLUG

MR.WICHET YIMLAMAI

MR.SOMBUT HATHAIRATTANANON

ADVISORS

MR.PIYA JITTHAMMAPIROM

MR.AMPHON THONGRA-AR

MR.PAIBOON PONGWONGTRAGULL

1997

ABSTRACT

This thesis presents 8088 Single Board Microcomputer. It is designed to work at 5MHz , and has 64 KB memory. It has 6 x 4 matrix switchs and show on the liquid cystal display. In addition , it is modified to receive and show the complex processing by 8088 Microcomputer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงลงได้ด้วยดีซึ่งได้รับความช่วยเหลือและการให้คำแนะนำปรึกษาจากท่านอาจารย์ปิยะ จิตธรรมมาภิรมย์และอาจารย์วรวิทย์ สมหา อีกทั้งอาจารย์ที่ปรึกษาปริญญานิพนธ์และคณาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ตลอดจนสมาชิกในกลุ่มที่ร่วมมือกันทำงานอีกทั้งเพื่อนนักศึกษาและนักศึกษารุ่นน้องทุกท่านที่ช่วยให้กำลังใจและอำนวยความสะดวกในการทำปริญญานิพนธ์ฉบับนี้จนสำเร็จลุล่วงไปด้วยดี

อนึ่ง คณะผู้จัดทำปริญญานิพนธ์ฉบับนี้ ต้องขอบพระคุณบุพการีที่เคารพยิ่งที่ได้ให้การอบรม เลี้ยงดู สนับสนุนส่งเสริมทางด้านการศึกษามาโดยตลอด อีกทั้งคณะครู - อาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ ให้การอบรม รวมทั้งขอขอบคุณ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การสนับสนุนด้านงบประมาณ ในการจัดทำปริญญานิพนธ์ ฉบับนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	3
1.3 เนื้อหาโดยสังเขป	3
บทที่ 2 ทฤษฎีและหลักการ	5
2.1 กล่าวนำ	5
2.2 8088 ไมโครโปรเซสเซอร์ 16 บิต	5
2.2.1 สถาปัตยกรรม 8088 ไมโครโปรเซสเซอร์ 16 บิต	5
2.2.2 การเชื่อมต่อในโหมด MINIMUM SYSTEM MODE	7
2.2.3 การเชื่อมต่อในโหมด MAXIMUM SYSTEM MODE	12
2.2.4 รีจิสเตอร์ภายใน 8088 ไมโครโปรเซสเซอร์ 16 บิต	18
2.3 บัสแอดเดรสและบัสข้อมูล	25
2.3.1 AD0 - AD7 (ADDRESS / DATA BUS ; ขา 16 - 9)	25
2.3.2 A8 - A15 (ADDRESS BUS ; ขา 8 - 2)	25
2.3.3 A16 / S3 (ADDRESS / STATUS ; ขา 38)	25
2.3.4 A17 / S4 (ADDRESS / STATUS ; ขา 37)	26
2.3.5 A18 / S5 (ADDRESS / STATUS ; ขา 36)	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.3.6 A19 / S6 (ADDRESS / STATUS ; ขา 35)	26
2.4 บัสควบคุมและแสดงสถานะ	26
2.4.1 $\overline{RD}$ (READ ; ขา 32)	27
2.4.2 READY (ขา 22)	27
2.4.3 $\overline{TEST}$ (ขา 23)	27
2.4.4 INTR (INTERTERRUPT REQUEST ; ขา 18)	27
2.4.5 NMI (NON - MASKABLE INTERRUPT ; ขา 17)	27
2.4.6 RESET (ขา 21)	27
2.4.7 $\overline{S0}$; $\overline{DEN}$ (STATUS ; DATA ENABLE ; ขา 26)	28
2.4.8 $\overline{S1}$; DT / $\overline{R}$ (STATUS ; DATA TRANSMIT / RECEIVE ; ขา 27)	29
2.4.9 $\overline{S2}$; IO / $\overline{M}$ (STATUS ; IO or MEMORY ACCESS ; ขา 28)	29
2.4.10 QS0 ; ALE (QUEUE STATUS ; ADDRESS LATCH ENABLE)	29
2.4.11 QS1 ; $\overline{INTA}$ (QUEUE STATUS ; INTERRUPT ACKNOWLEDGE ; ขา 24)	30
2.4.12 $\overline{RQ}$ / $\overline{GT0}$; HOLD (REQUEST / GRANT0 ; HOLD ; ขา 31)	30
2.4.13 $\overline{RQ}$ / $\overline{GT1}$; HOLDA (REQUEST / GRANT1 ; HOLD ACKNOWLEDGE ; ขา 30)	30
2.4.14 LOCK ; $\overline{WR}$ (LOCK ; WRITA SIGNAL ; ขา 29)	31
2.5 บัสจ่ายกำลังและฐานเวลา	31
2.5.1 CLK (CLOCK ; ขา 19)	31
2.5.2 VCC (ขา 40)	31
2.5.3 GND (GROUND ; ขา 1 และ 20)	31
2.6 การแสดงแอดเดรสของ 8088	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.7 บัสไซเคิลของระบบ	32
2.7.1 บัสไซเคิลที่สร้างขึ้นโดย 8088	33
บทที่ 3 การออกแบบและการสร้าง	**38**
3.1 กล่าวนำ	38
3.2 การออกแบบและการสร้าง	39
3.3 การออกแบบด้านฮาร์ดแวร์	39
3.3.1 ชุด กำเนิดสัญญาณนาฬิกา	39
3.3.2 ชุด รีเซ็ต	40
3.3.3 ชุด แลตช์ข้อมูล	40
3.3.4 ชุด หน่วยความจำ	41
3.3.5 ชุด พอร์ตอินพุต / เอาต์พุต	42
3.3.6 ชุด จอแสดงผลผลึกเหลว	44
3.3.7 ชุด คีย์บอร์ด 6 x 4 และวงจร D/A CONVERTER	44
3.3.8 ชุด A/D CONVERTER	46
3.4 การออกแบบด้านซอฟท์แวร์	50
3.4.1 ผังการทำงานโปรแกรมสแกนคีย์	51
บทที่ 4 ผลการทดลองและทดสอบ	**57**
4.1 กล่าวนำ	57
4.2 การทดลองชุดอุปกรณ์คีย์บอร์ด	57
4.2.1 ลำดับขั้นการทดลอง	57
4.2.2 ผลการทดลอง	62
4.3 การทดลองการทำงานของอินพุต / เอาต์พุต	63
4.3.1 ลำดับขั้นการทดลอง	63
4.3.2 ผลการทดลอง	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
4.4 การทดลองการทำงานของชุด จอแสดงผลผลึกเหลว	66
4.4.1 ลำดับขั้นการทดลอง	66
4.4.2 ผลการทดลอง	68
4.5 การทดลองการทำงานร่วมกันของคีย์บอร์ดกับจอแสดงผลผลึกเหลว	70
4.5.1 ลำดับขั้นการทดลอง	70
4.5.2 ผลการทดลอง	77
4.6 การทดลองการควบคุมการแสดงผลของจอแสดงผลผลึกเหลว	79
4.6.1 ลำดับขั้นการทดลอง	79
4.6.2 ผลการทดลอง	86
บทที่ 5 สรุปอภิปรายและข้อเสนอแนะ	**88**
5.1 บทสรุป	88
5.2 ปัญหาและแนวทางแก้ไข	88
5.3 ข้อเสนอแนะและแนวทางในการพัฒนา	90
ภาคผนวก ก การจัดแบ่งหน่วยความจำ	92
ภาคผนวก ข ผังการทำงานและโปรแกรมการทำงาน	94
ภาคผนวก ค คำสั่งที่ใช้ในการสั่งงาน	142
ภาคผนวก ง รายละเอียดข้อมูลและคุณสมบัติของอุปกรณ์	179
ภาคผนวก จ รายละเอียดอุปกรณ์	198
ภาคผนวก ฉ วงจรรวมของไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 และแผ่นวงจรพิมพ์	202
ภาคผนวก ช รูปต้นแบบไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088	211
บรรณานุกรม	213
ประวัติผู้แต่ง	214

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การควบคุมแบบ COMMON SIGNAL	8
ตารางที่ 2.2 การควบคุมแบบ MINIMUM MODE SIGNAL	10
ตารางที่ 2.3 การควบคุมแบบ MAXIMUM MODE SIGNAL	10
ตารางที่ 2.4 สถานะของสัญญาณขา S4 S3	12
ตารางที่ 2.5 การกำหนดค่า บัสสถานะ	16
ตารางที่ 2.6 ระดับค่าของแรงดันอินพุต / เอาต์พุต	17
ตารางที่ 2.7 การใช้ เซกเมนต์รีจิสเตอร์และออฟแซ็ต	21
ตารางที่ 2.8 การใช้งานรีจิสเตอร์เซกเมนต์	26
ตารางที่ 2.9 กลุ่มของสัญญาณที่ทำงานขึ้นอยู่กับสัญญาณ MN / $\overline{MX}$	28
ตารางที่ 2.10 สถานะการทำงานของสัญญาณ S2 S1 S0	29
ตารางที่ 2.11 สถานะ QS0 QS1 ในการเก็บชุดคำสั่ง	30
ตารางที่ 4.1 ค่าตำแหน่งของแป้นพิมพ์	61
ตารางที่ 5.1 ปัญหาและแนวทางการแก้ไขปัญหาในการทำปริญญานิพนธ์	89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปภาพ	หน้า
รูปที่ 1.1 โครงสร้างบล็อกไดอะแกรมของไมโครโปรเซสเซอร์ 8080	2
รูปที่ 1.2 โครงสร้างบล็อกไดอะแกรมของไมโครโปรเซสเซอร์ 8088	2
รูปที่ 2.1 รูปร่างและชื่อขา 8088	6
รูปที่ 2.2 รีจิสเตอร์ที่ใช้ใน 8088 ไมโครคอมพิวเตอร์ 16 บิต	7
รูปที่ 2.3 โครงสร้างบัสภายในของ 8088 ไมโครโปรเซสเซอร์ 16 บิต	8
รูปที่ 2.4 การต่อใช้งานใน MINIMUM MODE 8088 ไมโครโปรเซสเซอร์ 16 บิต	11
รูปที่ 2.5 การต่อใช้งานใน MAXIMUM MODE 8088 ไมโครโปรเซสเซอร์ 16 บิต	14
รูปที่ 2.6 (a) บล็อกไดอะแกรมของ 8288 (b) ตำแหน่งขาของ 8289	15
รูปที่ 2.7 (a) บล็อกไดอะแกรมของ 8289 (b) ตำแหน่งขาของ 8289	16
รูปที่ 2.8 โครงสร้างของรีจิสเตอร์ของ 8088 ไมโครโปรเซสเซอร์ 16 บิต กับการคำนวณแอดเดรส	19
รูปที่ 2.9 เซกเมนต์รีจิสเตอร์ทั้งสี่จะแบ่งกลุ่มของหน่วยความจำโดยแต่ละเซกเมนต์จะชี้ตำแหน่งจุดเริ่มต้นของเซกเมนต์นั้น	22
รูปที่ 2.10 เซกเมนต์รีจิสเตอร์ แต่ละตัวประกอบด้วยแอดเดรส 16 บิต ทางด้านบนสี่บิตล่างที่เหลือจะถือว่าเป็นศูนย์	23
รูปที่ 2.11 การคำนวณให้ได้ 20 บิตเพื่อใช้ในการอ้างแอดเดรส	24
รูปที่ 2.12 แฟลีกของ 8086 ที่ใช้ในการทดสอบการทำงานของซีพียู	24
รูปที่ 2.13 บัสไซเคิลของการอ่านข้อมูลจากหน่วยความจำ	33
รูปที่ 2.14 บัสไซเคิลของการเขียนข้อมูลลงบนหน่วยความจำ	34
รูปที่ 2.15 บัสไซเคิลของการอ่านข้อมูลจากพอร์ตอินพุต / เอาต์พุต	36
รูปที่ 2.16 บัสไซเคิลของการเขียนข้อมูลลงบนพอร์ตอินพุต / เอาต์พุท	37
รูปที่ 3.1 ส่วนประกอบของ 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยว	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปภาพ	หน้า
รูปที่ 3.2 วงจร กำเนิดสัญญาณนาฬิกา	39
รูปที่ 3.3 วงจร รีเซ็ต	40
รูปที่ 3.4 วงจร แลตช์ข้อมูล	41
รูปที่ 3.5 การจัดหน่วยความจำ	42
รูปที่ 3.6 วงจร ดีโค้ดหน่วยความจำ	43
รูปที่ 3.7 วงจร พอร์ตอินพุต / เอาต์พุต	43
รูปที่ 3.8 วงจร จอแสดงผลผลึกเหลว	45
รูปที่ 3.9 วงจร D/A CONVERTER	45
รูปที่ 3.10 วงจร คีย์บอร์ดขนาด 6 x 4	46
รูปที่ 3.11 วงจร A/D CONVERTER	46
รูปที่ 3.12 ชุดควบคุมของ 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยว	47
รูปที่ 3.13 ชุดอุปกรณ์ต่อพวงในการทดลองชุดควบคุมของ 8088	47
รูปที่ 3.14 ส่วนจอแสดงผลผลึกเหลว	48
รูปที่ 3.15 ชุดคีย์บอร์ดขนาด 6x4	48
รูปที่ 3.16 ชุดวงจร D/A CONVERTER	49
รูปที่ 3.17 ชุดวงจรพอร์ต อินพุต / เอาต์พุต	49
รูปที่ 3.18 ผังการทำงานโปรแกรมหลัก	50
รูปที่ 3.19 ผังการทำงานโปรแกรมสแกนคีย์ (PART I)	51
รูปที่ 3.20 ผังการทำงานโปรแกรมสแกนคีย์ (PART II)	52
รูที่ 3.21 ผังการทำงานโปรแกรมสแกนคีย์ (PART III)	53
รูปที่ 3.22 ผังการทำงานโปรแกรมสแกนคีย์ (PART IV)	54
รูปที่ 3.23 ผังการทำงานโปรแกรมสแกนคีย์ (PART V)	55
รูปที่ 3.24 ผังการทำงานโปรแกรมสแกนคีย์ (PART VI)	56
รูปที่ 4.1 จอภาพเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.1	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปภาพ	หน้า
รูปที่ 4.2 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.2	65
รูปที่ 4.3 ต่อวงจรแสดงผลการทดลอง	65
รูปที่ 4.4 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.3	69
รูปที่ 4.5 การต่อวงจรภาคแสดงผลการทดลอง	69
รูปที่ 4.6 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.4	78
รูปที่ 4.7 การต่อวงจรทดลองและผลการทดลองโปรแกรมการทดลองที่ 4.4	78
รูปที่ 4.8 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.5	87
รูปที่ 4.9 การต่อวงจรทดลองและผลการทดลองโปรแกรมการทดลองที่ 4.5	87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

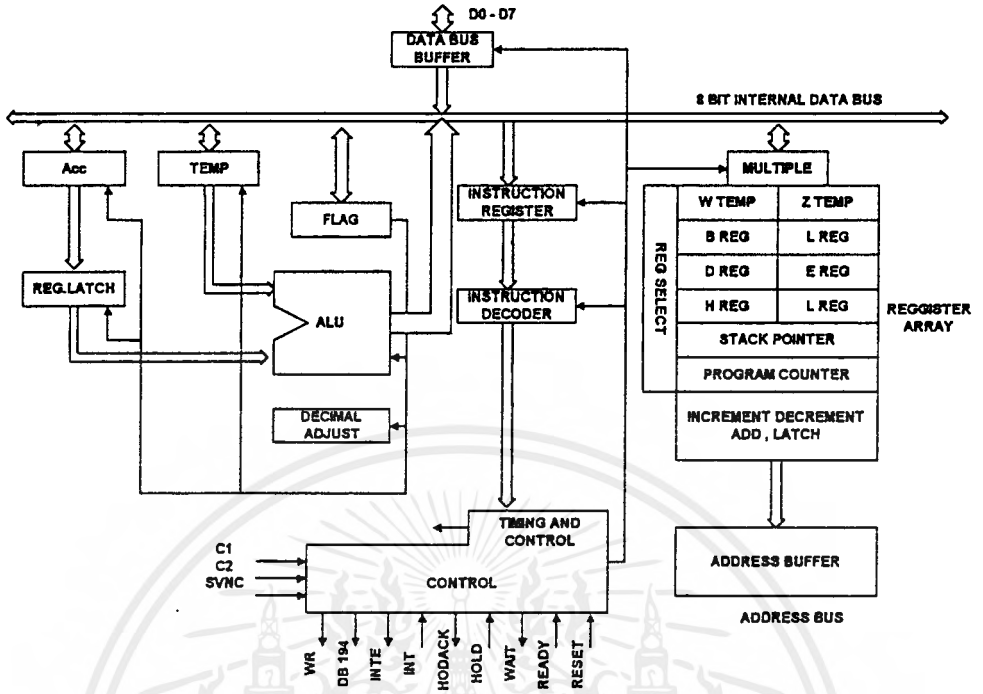
บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

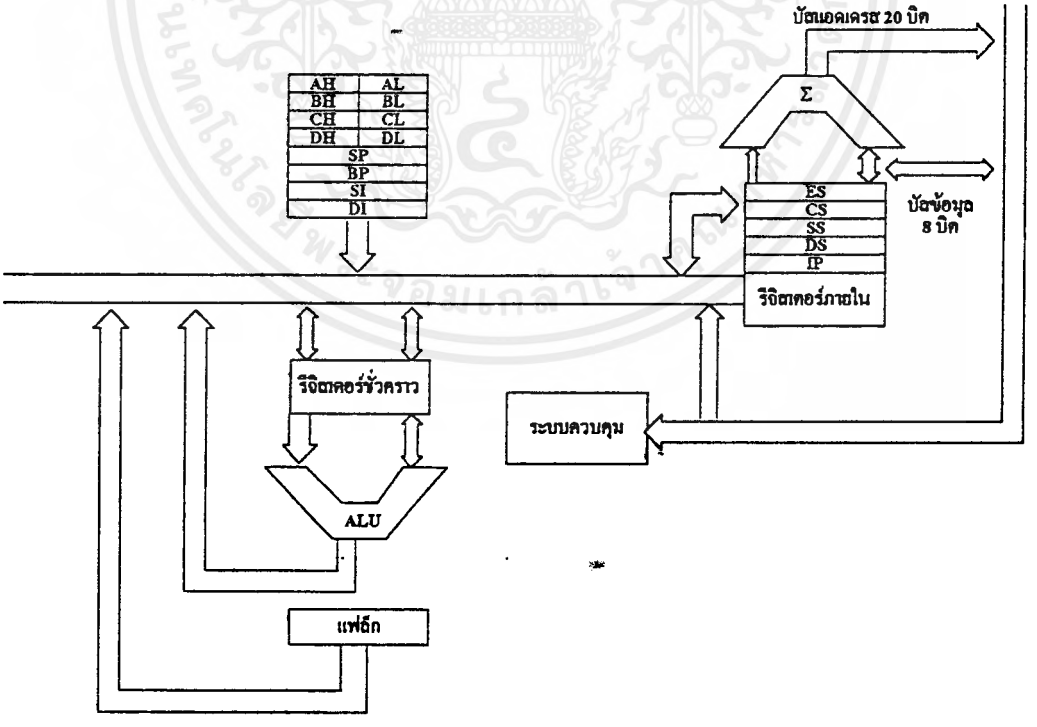
ปัจจุบันได้มีการนำคอมพิวเตอร์เข้ามาใช้ในชีวิตประจำวันมากขึ้นโดยจะสังเกตเห็นได้จากการนำเอาไมโครโปรเซสเซอร์มาใช้ในด้านต่างๆ เช่น ด้านอุตสาหกรรม , ด้านการผลิต , ด้านการโทรคมนาคมสื่อสาร , ด้านเกษตรกรรมและสิ่งอำนวยความสะดวกในด้านต่างๆ โดยไมโครโปรเซสเซอร์ที่นิยมใช้กันในระยะแรกๆจะเป็นไมโครโปรเซสเซอร์ที่ผลิตจากบริษัทอินเทล มีขนาด 8 บิต เบอร์ 8080 แต่ในปัจจุบันได้มีการพัฒนาไมโครโปรเซสเซอร์ให้มีขนาด 16 , 32 , และ64 บิต ในด้านการศึกษาของประเทศไทยส่วนใหญ่นั้น ไมโครโปรเซสเซอร์ที่ใช้ในการศึกษายังคงมีขนาด 8 บิต อยู่ซึ่งเป็นไมโครโปรเซสเซอร์เบอร์ 8080 ดังนั้นเพื่อเป็นการพัฒนาการใช้งานของไมโครโปรเซสเซอร์ที่ใช้ในการศึกษาของประเทศไทยคณะผู้จัดทำจึงได้นำไมโครโปรเซสเซอร์ขนาด 16 บิต มาใช้แทนของเดิมทั้งนี้เพื่อเป็นการพัฒนาขีดความสามารถในการนำไมโครโปรเซสเซอร์ขนาด 16 บิต มาประยุกต์ใช้งาน

จากรูปที่ 1.1 จะแสดงให้ทราบเกี่ยวกับโครงสร้างภายในของไมโครโปรเซสเซอร์เบอร์ 8080 ซึ่งขีดความสามารถในการทำงานของ 8080 นั้นสามารถทำงานได้อย่างจำกัด โดยสามารถอ้างหน่วยความจำได้เพียง 64 กิโลไบต์ มีแอดเดรสบัส 16 บิต บัสข้อมูลขนาด 8 บิต และมีรีจิสเตอร์หลักที่ใช้งานทั่วไปมีขนาด 8 บิต และรีจิสเตอร์ที่ใช้งานเฉพาะอย่างขนาด 16 บิต จากลักษณะดังกล่าวเมื่อนำ 8080 มาใช้งานจะทำให้ไม่เพียงพอกับการทำงานและความเร็วในการส่งข้อมูลได้ไม่สูงนักและสำหรับรูปที่ 1.2 จะแสดงให้เห็นถึงลักษณะของโครงสร้างภายในของไมโครโปรเซอร์เบอร์ 8088 โดยประกอบด้วยแอดเดรสบัส 20 บิต ทำให้สามารถอ้างตำแหน่งหน่วยความจำได้ถึง 1,045,576 ไบต์ หรือ 1 เมกะไบต์ โดยจะมากเป็น 16 เท่าของไมโครโปรเซสเซอร์ที่มีขนาด 8 บิต ส่วนของขนาดบัสข้อมูลยังคงมีขนาด 8 บิต เท่ากับไมโครโปรเซสเซอร์ 8080 ทั้งนี้เพราะว่าจะสามารถใช้ได้กับตัว 8080 ซึ่งมีอยู่จำนวนมาก ทำให้ใช้อุปกรณ์สนับสนุนได้คล่องตัว ด้วยเหตุนี้เองทางผู้จัดทำจึงใช้ไมโครโปรเซสเซอร์เบอร์ 8088 เป็นตัวประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่1.1 โครงสร้างบล็อกไดอะแกรมของไมโครโปรเซสเซอร์ 8080



รูปที่ 1.2 โครงสร้างบล็อกไดอะแกรมของไมโครโปรเซสเซอร์ 8088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์ของการทำปริญญานิพนธ์เพื่อศึกษาการทำงาน , การอินเตอร์เฟสของอุปกรณ์ต่อพ่วงกับไมโครโปรเซสเซอร์ 8088 อีกทั้งยังเพื่อทำการออกแบบและจัดสร้างแผ่นพิมพ์เดี่ยวของไมโครโปรเซสเซอร์ 8088 รวมทั้งการเขียนโปรแกรมควบคุมระบบเพื่อให้ 8088 ทำงานได้ตามแนวทางที่กำหนดให้ ทั้งนี้เพื่อเป็นแนวทางในการศึกษาและพัฒนาความรู้ด้านไมโครโปรเซสเซอร์ 8088

1.2ขีดความสามารถของโครงงาน

ขอบเขตของการทำปริญญานิพนธ์ นำแผ่นพิมพ์เดี่ยว 8088 มาใช้ศึกษาคำสั่งการควบคุมโดยใช้ 8088 ควบคุมการทำงานกับอุปกรณ์ต่อร่วมเช่น จอแสดงผลผลึกเหลวให้แสดงผลตามคำสั่งที่ให้ 8088 ทำงาน คีย์เมตริกซ์ 6 × 4 ซึ่งประโยชน์ที่คาดว่าจะได้รับจากการทำก็คือความรู้และความเข้าใจในโครสร้างการทำงานและการต่ออินเตอร์เฟสของไมโครโปรเซสเซอร์ 8088 อีกทั้งสามารถที่จะออกแบบและสร้างแผ่นพิมพ์เดี่ยวไมโครโปรเซสเซอร์ 8088 ได้ สามารถที่นำไมโครโปรเซสเซอร์มาประยุกต์ใช้งานได้ตลอดจนเขียนโปรแกรมควบคุมระบบได้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆเพื่อความสะดวกและเข้าใจต่อการศึกษา ในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 2 ทฤษฎีและหลักการประกอบด้วยเนื้อหาทางสถาปัตยกรรมของไมโครโปรเซสเซอร์ 16 บิต ทฤษฎีการอินเตอร์เฟส หลักการทำงานของไมโครคอมพิวเตอร์เบอร์ 8088

บทที่ 3 การออกแบบการสร้าง ชุด 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวซึ่งจะประกอบด้วยวงจร กำเนิดสัญญาณนาฬิกา , วงจรแลตช์ข้อมูล , วงจรชุด D/A CONVERTER ชุดควบคุมการแสดงผลบนจอแสดงผลผลึกเหลวอีกทั้งลักษณะการทำงานของซอฟแวร์ ที่ใช้ในการควบคุมการทำงานของชุด 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยว

บทที่ 4 การทดลองและผลการทดลองในการทำงานของชุด 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวกับชุดทดลองอาทิเช่น ชุดแป้นพิมพ์ ชุดจอแสดงผลผลึกเหลวและชุดทดลองการทำงานของพอร์ตอินพุต /เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 สรุปปัญหาแนวทางการแก้ไขปัญหาที่เกิดขึ้นตลอดจนทั้งข้อบกพร่องที่พบในการทำงานของชุด 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวพร้อมทั้งแนวคิดพัฒนาชุด 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเพื่อให้มีประสิทธิภาพและการใช้งานที่สูงกว่าเดิม

ในภาคผนวกแสดงรายละเอียดของโปรแกรมควบคุมระบบและรายการอุปกรณ์ต่างๆ ที่ใช้จัดทำโครงงานดังนี้

ภาคผนวก ก การจัดแบ่งหน่วยความจำ

ภาคผนวก ข ผังการทำงานและโปรแกรมการทำงาน

ภาคผนวก ค คำสั่งที่ใช้ในการสั่งงาน

ภาคผนวก ง รายละเอียดข้อมูลและคุณสมบัติของอุปกรณ์

ภาคผนวก จ รายละเอียดอุปกรณ์

ภาคผนวก ฉ วงจรรวมของไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 และแผ่นวงจรพิมพ์

ภาคผนวก ช รูปต้นแบบไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1กล่าวนำ

ในการที่จะทำการออกแบบและสร้างชุด 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยว ได้นั้นผู้จักทำมีความจำเป็นอย่างมากในการที่จะต้องทำการศึกษาเกี่ยวกับโครงสร้างสถาปัตยกรรมไมโครโปรเซสเซอร์ขนาด 16 บิต,ทฤษฎีการอินเตอร์เฟส,หลักการทำงานของ 8088 ให้เกิดความเข้าใจการทำงานของ 8088 ให้มากขึ้นอีกทั้งยังเป็นแนวทางในการพัฒนาโครงงาน 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวและเนื้อหาที่ได้กล่าวไว้ข้างต้นนี้จะได้แสดงรายละเอียดตามหัวข้อต่อไปนี้

2.2 8088ไมโครโปรเซสเซอร์ 16 บิต

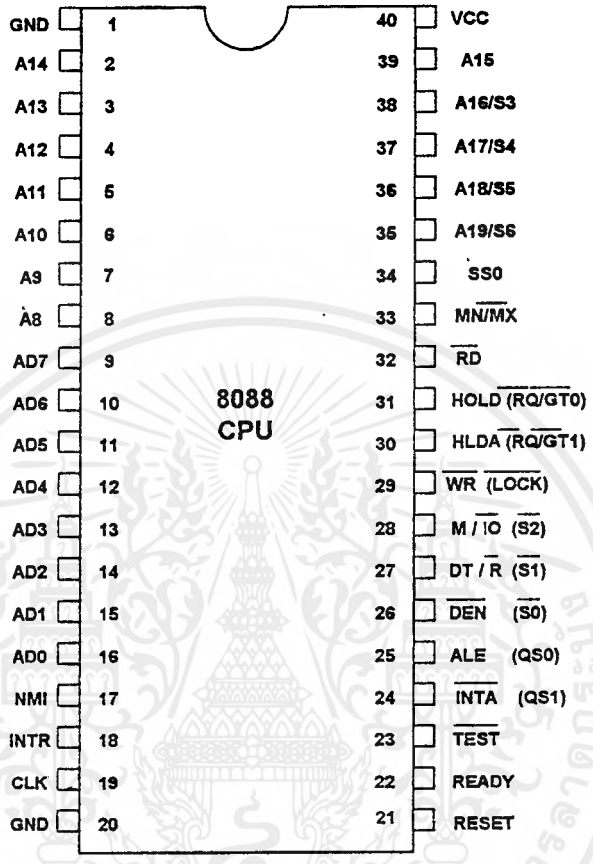
2.2.1 สถาปัตยกรรม 8088 ไมโครโปรเซสเซอร์ 16 บิต

8088 ไมโครโปรเซสเซอร์ถูกนำมาใช้ในครั้งแรกปี พ.ศ.2521 เป็นแบบ 16 บิต ผลิตโดยบริษัท Intel 8088 ถูกผลิตโดยเทคโนโลยี HMOS และวงจรบนตัวอุปกรณ์ใช้ทรานซิสเตอร์ประมาณ 29,000 ตัว 8088 มีขาทั้งหมด 40 ขา ซึ่งมีการทำงานเป็นแบบหลายหน้าที่เช่นขา Address A0-A7 กับขา D0-D7 จะใช้ขาเดียวกัน 8088 เป็นไมโครโปรเซสเซอร์แบบ 16 บิต ทั้งภายในและภายนอกมันสามารถอ้างหน่วยความจำได้ทั้งหมด 1 เมกะไบต์ มีแอดเดรสที่ใช้กับพอร์ตอินพุตเอาต์พุตได้ถึง 64 กิโลไบต์ 8088 มีการทำงานอยู่ 2 โหมด คือ Minimum System mode และ Maximun System mode การทำงาน Minimum System mode ทำการเลือกโดยให้ขา MN / $\overline{MX}$ เป็นลอจิก “1” ในโหมดนี้เป็นโหมดที่ใช้ไมโครโปรเซสเซอร์เพียงตัวเดียวส่วน Maximum System mode ทำงานที่ MN / $\overline{MX}$ เป็นลอจิก “0” เป็นระบบที่รองรับตัวไมโครโปรเซสเซอร์ได้หลายตัวขาและหน้าที่การทำงานของขาต่างๆแสดงไว้ดังรูปที่ 2.1

รีจิสเตอร์ของ 8088 ประกอบด้วยรีจิสเตอร์สำหรับจุดประสงค์ทั่วไป รีจิสเตอร์ชี้คำสั่งแฟล็ก , รีจิสเตอร์อินเด็กซ์และเซกเมนต์รีจิสเตอร์ดังรูปที่ 2.2 รีจิสเตอร์เหล่านี้ถูกใช้ สำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ้างแอดเดรสแสดงสถานะของ8088และสำหรับเฟตช์คำสั่งเพื่อมาตีความและกระทำตามกระบวนการที่ถูกกำหนดให้โดยการเขียนคำสั่งให้ปฏิบัติ



รูปที่ 2.1 รูปร่างและชื่อขา 8088

ในบรรดารีจิสเตอร์ที่มีอยู่ทั้งหมดของ 8088 เป็นรีจิสเตอร์ขนาด 16 บิตรีจิสเตอร์ AX จะประกอบด้วย AH และ AL โดย AX ทำหน้าที่ในการใช้คูณหารหรือเป็นรีจิสเตอร์เกี่ยวกับอินพุตเอาต์พุตที่เป็นเวิร์ด AL ทำหน้าที่ใช้ในการ คูณ หาร หรือทำงานเกี่ยวกับอินพุตเอาต์พุต แปลงข้อมูล จัดการคำนวณแบบตัวเลขฐานสิบแบบ 8 บิต ส่วน AH ใช้เป็นรีจิสเตอร์ในการคูณและหารได้ BX ใช้ในการแปลงข้อมูล CX ใช้ในคำสั่งจัดการเกี่ยวกับสตริงและการลูป CL ใช้เป็นตัวแปรสำหรับการเลื่อนบิตหรือหมุนบิต DX ใช้ในคำสั่งคูณ หาร เป็นเวิร์ดหรือใช้ ในรูปแบบอ้างอินพุตเอาต์พุตแบบอ้อม SPใช้กับคำสั่งสแตก SI และ DI ใช้กับคำสั่งที่เกี่ยวกับสตริงในทางปฏิบัติแล้ว 8088 มีแอดเดรส 20 เส้น ซึ่งทำให้อ้างแอดเดรสได้ถึง 1,045,576ไบต์ หรือ 1 เมกะไบต์หรือมากเป็น 16 เท่า ของไมโครโปรเซสเซอร์ขนาด 8 บิต บัสข้อมูลมีขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8 บิต ทำให้ 8088 สามารถเรียกข้อมูลได้ทีละ 1 เวิร์ดหรือ 2 ไบต์ได้และยังเตรียมพร้อมสำหรับทำงานแบบใช้หลาย ซีพียู และทำงานร่วมกับอุปกรณ์อื่นๆที่จะทำงานร่วมกับ 8088 ได้ด้วย

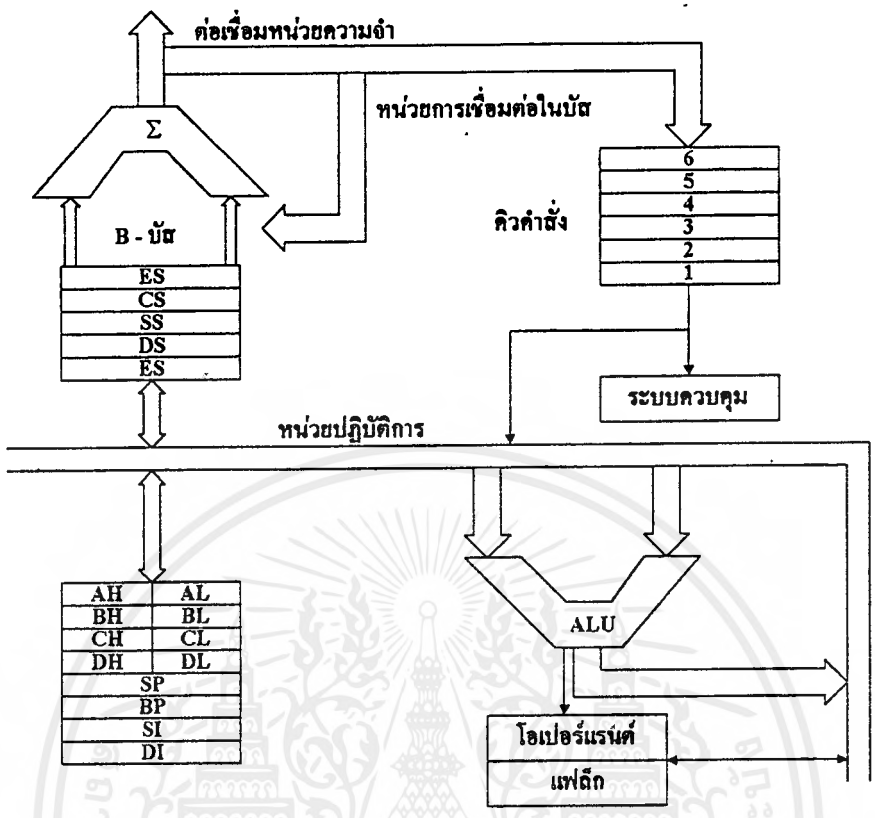
โครงสร้างสถาปัตยกรรมของ 8088 แสดงเป็นไดอะแกรมได้ดังรูปที่ 2.3 เพื่อที่จะทำให้จะได้สามารถเข้าใจในลักษณะของรีจิสเตอร์ได้ดียิ่งขึ้นและนำหลักการไปใช้ในการเขียนโปรแกรมควบคุมใช้ 8088 ไมโครโปรเซสเซอร์ 16 บิต

รูปที่ 2.2 รีจิสเตอร์ที่ใช้ใน 8088 ไมโครโปรเซสเซอร์ 16 บิต

2.2.2 การเชื่อมต่อในโหมด Minimum System Mode

เมื่อทำการเลือกในโหมดนี้ 8088 จะทำการจัดสัญญาณควบคุมที่ต้องการใช้ให้แก่อุปกรณ์ที่ทำการเชื่อมต่อกับหน่วยความจำและอินพุตเอาต์พุต รูปที่ 2.3 บล็อกไดอะแกรมของการต่อในโหมด Minimum-System Mode สัญญาณจะถูกแบ่งออกเป็นกลุ่มได้แก่ แอดเดรสบัส / บัสข้อมูล , สัญญาณสถานะ , สัญญาณควบคุม , สัญญาณอินเตอร์รัพท์ และ สัญญาณอินเตอร์เฟส DMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โครงสร้างบัสภายในของ 8088 ไมโครโปรเซสเซอร์ 16 บิต

ตารางที่ 2.1 การควบคุมแบบ Common Signal

ชื่อขา	หน้าที่	สภาวะของขา
AD0-AD7	ขาสัญญาณ บัสข้อมูล และแอดเดรสบัส	2 ทิศทาง , 3สภาวะ
A8-A15	ขาสัญญาณ แอดเดรสบัส	เอาต์พุต , 3สภาวะ
A19/S6-A16/S3	ขา แอดเดรสบัส / สถานะ	เอาต์พุต , 3สภาวะ
MN / $\overline{MX}$	ถ้าเป็น “0” ใช้โหมด Max ถ้าเป็น “1” ใช้โหมด Min	อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1(ต่อ) การควบคุมแบบ Common Signal

ชื่อขา	หน้าที่	สภาวะของขา
RD	สัญญาณการอ่าน	เอาต์พุต , 3สภาวะ
TEST	หยุดรอสำหรับควบคุมการทดสอบ	อินพุต
READY	สัญญาณที่ต้องการ Wait	อินพุต
RESET	สัญญาณ รีเซ็ต	อินพุต
NMI	สัญญาณ Non-Maskable อินเตอร์รัพท์	อินพุต
CLK	นาฬิกาของระบบ	อินพุต
VCC	+5โวลต์	อินพุต
GND	กราวด์	อินพุต

1.แอดเดรสบัสและบัสข้อมูลที่ขานี้เส้นเดียวใช้ 2 ฟังค์ชัน โดยแอดเดรสบัสที่ใช้ในตำแหน่งของหน่วยความจำและอินพุตเอาต์พุตพอร์ต แอดเดรสบัส มี 20 เส้น ประกอบด้วย A0-A19 สามารถอ้างหน่วยความจำได้ 1 เมกะไบต์ และสามารถอ้างตำแหน่งอินพุต /เอาต์พุตพอร์ตได้ 64 กิโลไบต์ บัสข้อมูล มีทั้งหมด 8 เส้นคือ D0-D7 ทั้งสองของบัสนี้จะใช้หลักการของมัลติเพลกซ์ คือจะทำงานคนละคาบเวลากันเมื่อไหร่ที่บัสข้อมูลจะทำงาน คือจะทำงานเมื่อต้องการอ่าน / เขียนข้อมูลกับหน่วยความจำอินพุต /เอาต์พุตพอร์ตและ Interrupt-type code จากตัว ควบคุมการอินเตอร์รัพต์

2.สัญญาณสถานะ สัญญาณขา A16-A19 มีการทำงานเป็นแบบมัลติเพลกซ์ด้วยแต่จะทำกับสัญญาณสถานะ S3-S6 สถานะของบิตเหล่านี้จะให้เอาต์พุตบนบัสในเวลาเดียวกันกับที่ข้อมูลทำการโอนหรือย้ายข้ามไปยังบัสอื่นๆบิต S4 และบิต S3 จะใช้ร่วมกันในรูปที่ 2.1 บิตไบนารีโค้ดซึ่งจะเป็นตัวบอกชื่อของเซกเมนต์รีจิสเตอร์ ที่อยู่ใน 8088 ค่าภายในเซกเมนต์รีจิสเตอร์ เหล่านี้จะถูกนำไปใช้ในการสร้างฟิสิคัลแอดเดรสโดยเซกเมนต์รีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

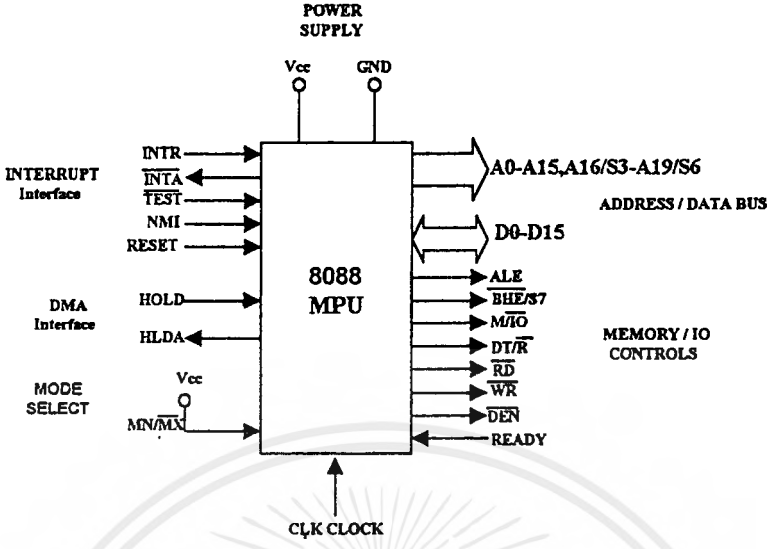
ตารางที่ 2.2 การควบคุมแบบ Minimum Mode Signal

Minimum Mode Signals (MN / $\overline{MX}$ = VCC)		
ชื่อขา	หน้าที่	สภาวะของขา
HOLD	ต้องการขอใช้บัสข้อมูลและแอดเดรสบัสของอุปกรณ์จาก 8088 ที่ควบคุมบัสอยู่	อินพุต
HLDA	ตอบรับการขอ Hold	เอาต์พุต
WR	สัญญาณการเขียน	เอาต์พุต , 3 สภาวะ
M/IO	สัญญาณเลือกหน่วยความจำหรือพอร์ต	เอาต์พุต , 3 สภาวะ
DT/R	ส่ง/รับ ข้อมูล	เอาต์พุต , 3 สภาวะ
DEN	สัญญาณการอีนาเบิ้ลข้อมูล	เอาต์พุต , 3 สภาวะ
ALE	สัญญาณควบคุมการ แลตช์แอดเดรส	เอาต์พุต
INTA	สัญญาณตอบรับการอินเตอร์รัพต์	เอาต์พุต

ตารางที่ 2.3 การควบคุมแบบ Maximum Mode Signal

•Maximum Mode Signals (MN / $\overline{MX}$ = GND)		
ชื่อขา	หน้าที่	สภาวะของขา
RQ/GT1,0	สัญญาณควบคุมการลำดับของบัส	2 ทิศทาง
LOCK	สัญญาณควบคุมการใช้บัส	เอาต์พุต , 3 สภาวะ
S2-S0	แสดงสถานะของซีพียู 8088	เอาต์พุต , 3 สภาวะ
QS1,QS0	สัญญาณแสดงคิวคำสั่ง	เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การต่อใช้งานใน Minimum Mode 8088 ไมโครโปรเซสเซอร์ 16 บิต

ภายในมีทั้งหมด 4 ตัวดังแสดงในตารางที่ 2.4 และ S5 แสดงสถานะสัมประสิทธิ์ภายในอื่นๆ ของ 8088 มักเป็นระดับลอจิก "0" เสมอ

3.สัญญาณควบคุม

- ALE (Address Latch Enable) จะให้ลอจิก 1 กับอุปกรณ์ภายนอกเพื่อทำการวางตำแหน่งลงบนบัสหลังจากนั้นแอดเดรสบัสจะถูกแลตช์ด้วยอุปกรณ์ภายนอกที่คมลบของพัลส์

- M / $\overline{IO}$ ถ้าเป็นลอจิก "1" จะติดต่อกับหน่วยความจำแต่ถ้าเป็นลอจิก "0" จะติดต่อกับอุปกรณ์อินพุต

- DT / $\overline{R}$ เป็นการกำหนดทิศทางถ้าเป็นลอจิก "1" จะทำการส่งข้อมูลให้กับหน่วยความจำและอุปกรณ์อินพุต/เอาต์พุตแต่ถ้าเป็นลอจิก "0" จะทำการรับ

- $\overline{DEN}$ จะให้ลอจิก "0" แก่อุปกรณ์ภายนอกเพื่อที่จะทำการวางข้อมูลลงบนบัส

- $\overline{RD}$ / $\overline{WR}$ ให้ลอจิก "0" เพื่อการอ่าน / เขียนกับอุปกรณ์ภายนอกและหน่วยความจำ จะถูกใช้ในการแทรกสภาวะในการคอยของบัสไซเคิล โดยจะทำการเพิ่มคาบเวลาในการคอย สัญญาณนี้ได้จากอุปกรณ์กำเนิดสัญญาณนาฬิกาภายนอก 8088 จะรับเข้ามาเพื่อทำการติดต่อกับหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุต เพื่อทำให้การถ่ายโอนข้อมูลถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 สถานะของสัญญาณขา S4 S3

S4	S3	เซกเมนต์รีจิสเตอร์
0	0	Extra
0	1	Stack
1	0	Code/none
1	1	Data

4. สัญญาณอินเตอร์รัพต์

- INTR เป็นขาอินพุตเมื่ออุปกรณ์ภายนอกต้องการขอใช้บริการ ขานี้จะเป็นลอจิก "1" หลังจากนั้นจะทำการส่งลอจิก "0" ออกไปที่ขา $\overline{\text{INTA}}$

- TEST เป็นอินพุตที่เกี่ยวข้องกับการเชื่อมต่อในการอินเตอร์รัพต์ภายนอกและจะทำการตรวจเช็คในกรณีที่ใช้โปรเซสเซอร์รวมโดยจะคอยบอกเวลาการทำงานว่าในขณะที่ไมโครโปรเซสเซอร์ตัวแรกทำงานเสร็จแล้วโปรเซสเซอร์อีกตัวจะทำงานในเวลาต่อมาในการกระทำของคำสั่ง WAIT 8088 จะทำการเช็คที่ขานี้ถ้าเป็นลอจิก "1" จะเกิดการคอยถ้าเป็นลอจิก "0" จะทำคำสั่งถัดไป

- NMI เป็นอินพุตทำงานที่คมบวก

- RESET เป็นอินพุตได้จาก ฮาร์ดแวร์

5.สัญญาณอินเตอร์รัพต์ DMA

- DMA (Direct Memory Access) ในระบบ Minimum Mode จะประกอบไปด้วยสัญญาณ HOLD และ HLDA เมื่อไหร่ที่อุปกรณ์ภายนอกต้องการควบคุมระบบบัสจะส่งลอจิก "1" ไปให้เขา HOLD 8088 จะเข้าสู่สภาวะของการ HOLD ขาสัญญาณ A0-A15 , A16 / S3 , A19 / S6, $\overline{\text{BHE}}$, M / $\overline{\text{IO}}$, DT / $\overline{\text{R}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{DEN}}$ และ INTR ทั้งหมดจะอยู่ในสภาวะความต้านทานสูงหลังจากนั้นจะทำการตอบรับโดยการส่งลอจิก "1" ที่ขา HLDA ออกไป

2.2.3 การเชื่อมต่อในโหมด Maximum System Mode

เมื่อ 8088 ถูกเลือกในโหมดนี้ 8088 จะทำการผลิตสัญญาณสนับสนุนการทำงานแบบมัลติโปรเซสเซอร์ / โคโปรเซสเซอร์ คือจะมีตัวไมโครโปรเซสเซอร์มากกว่าหนึ่งตัวในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และโปรเซสเซอร์แต่ละตัวจะทำการปฏิบัติงานตามโปรแกรมของเราที่เขียนโดยทั่วไปสภาวะแวดล้อมของระบบแบบนี้มีบางระบบที่เป็นโปรเซสเซอร์ธรรมดาทั้งหมดถูกเรียกว่า " Grobal Resource " และมีบางระบบถูกกำหนดให้ใช้โปรเซสเซอร์ชนิดพิเศษเรียกว่า Local หรือ Private Resources

คำว่า โคโปรเซสเซอร์ หมายความว่ามีโปรเซสเซอร์ตัวที่สองในระบบอย่างไรก็ตาม โปรเซสเซอร์ตัวที่สองนี้จะไม่เข้าถึงบัสที่เวลาเดียวกันมันจะผ่านการควบคุมของระบบบัสไปยังส่วนอื่นๆ

1. คอนโทรลเลอร์บัส 8288 และสัญญาณควบคุม

จากรูปที่ 2.5 บล็อกไดอะแกรมของ Maximum Mode เราจะเห็นว่าสัญญาณทั้งหมดที่ต้องการควบคุมหน่วยความจำอุปกรณ์อินพุต / เอาต์พุตและการเชื่อมต่ออินเตอร์เฟสไม่ได้ส่งโดยตรงสัญญาณ $\overline{WR}$, M / $\overline{IO}$, DT / R, $\overline{DEN}$, ALE และ $\overline{INTA}$ ไม่ได้ผลิตโดย 8088 แต่จะใช้กับสัญญาณสถานะ $\overline{S0} - \overline{S2}$ เป็นตัวระบุสัญญาณแต่ละเส้นของบัสไซเคิล รหัสของบัส 3 บิต จะเป็นตัวบอกชนิดของ บัสไซเคิล สัญญาณที่ออกจากขาเอาต์พุต $\overline{S0} - \overline{S2}$

จะถูกส่งไปให้กับอุปกรณ์ควบคุมบัส (8288) ซึ่งจะทำการถอดรหัสเพื่อให้ได้สัญญาณควบคุม บล็อกไดอะแกรมและขาสัญญาณของ 8288 ดังรูปที่ 2.6 (a) และ(b) ตามลำดับและตารางที่ 2.5 ความสัมพันธ์ระหว่างบัสสถานะโค้ดและสัญญาณควบคุม

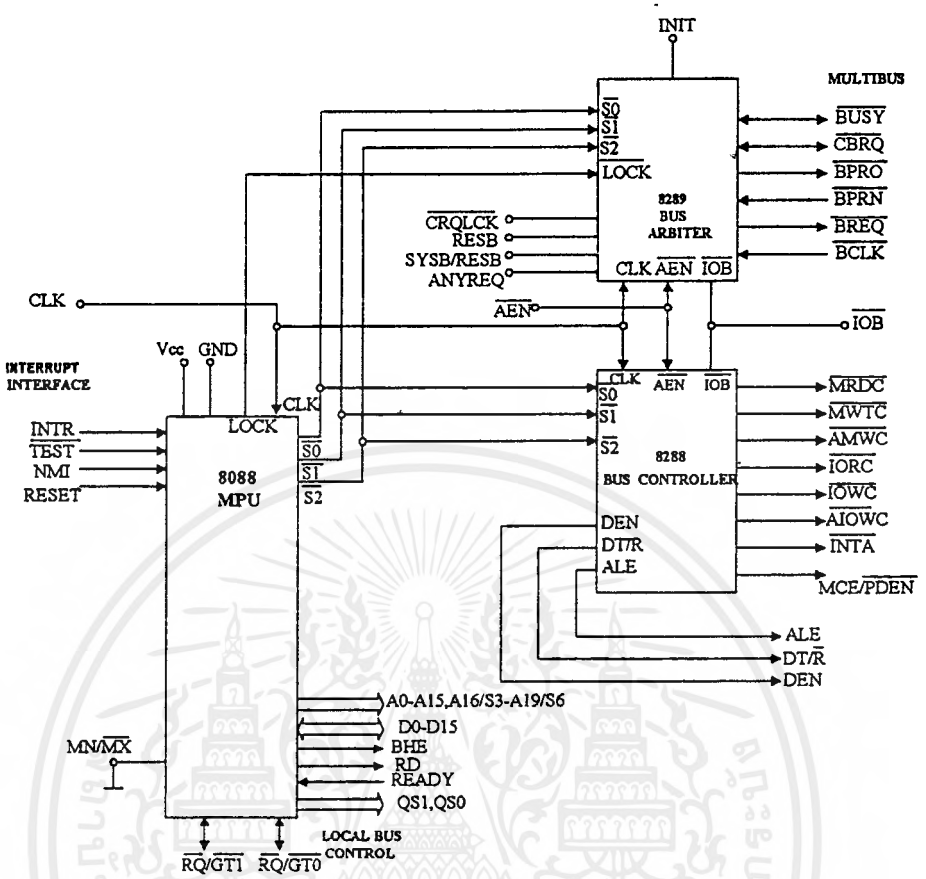
2. 8289 บัสและสัญญาณควบคุม

ในรูปที่ 2.7 จะเห็นว่า 8289 Bus Arbiter ถูกเพิ่มเข้าไปใน Maximum Mode มันเป็นอุปกรณ์ที่ยอมให้ใช้โปรเซสเซอร์ได้หลายตัวบนระบบบัสมันเป็นการสนับสนุนโปรโตคอลในระบบ พื้นฐาน 8088 ดังแสดงในรูปที่ 2.7 (a) และ(b) เป็นบล็อกไดอะแกรมและขาของ 8289

3. สัญญาณสถานะ คิว (Queue)

สองสัญญาณใหม่ที่ผลิตโดย 8088 ใน Maximum Mode เป็นแถวสถานะเอาต์พุต QS0 และ QS1 จะมีรูปแบบเป็น 2 บิต คือ QS0-QS1 รหัสนี้จะเป็นตัวบอกวงจรภายนอกถึงชนิดของข้อมูลที่เคลื่อนจากแถวลำดับระหว่างสัญญาณนาฬิกาต่างๆดังในรูปที่ 2.9 แสดงถึงความแตกต่างทั้งสี่ของแถวสถานะสังเกตุว่า QS0-QS1 = 00 จะบอกว่าไม่มีการทำงานระหว่างที่สัญญาณนาฬิกาสุดท้ายจะไม่มีการดึงข้อมูลจากแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การต่อใช้งาน Maximum Mode 8088 ไมโครโปรเซสเซอร์ 16 บิต

4. สัญญาณการขอใช้บัสข้อมูลและการอนุญาตให้ใช้บัสข้อมูล

ใน Maximum Mode ขาของ HOLD HLDA อินเตอร์เฟสใน Minimum Mode จะถูกเปลี่ยนเป็น Request / Grant ใน Maximum Mode $\overline{RQ}$ / GT0 และ $\overline{RQ}$ / $\overline{GT1}$ ตามลำดับมันจะเป็นการให้ลำดับก่อนหลังของสัญญาณบัส

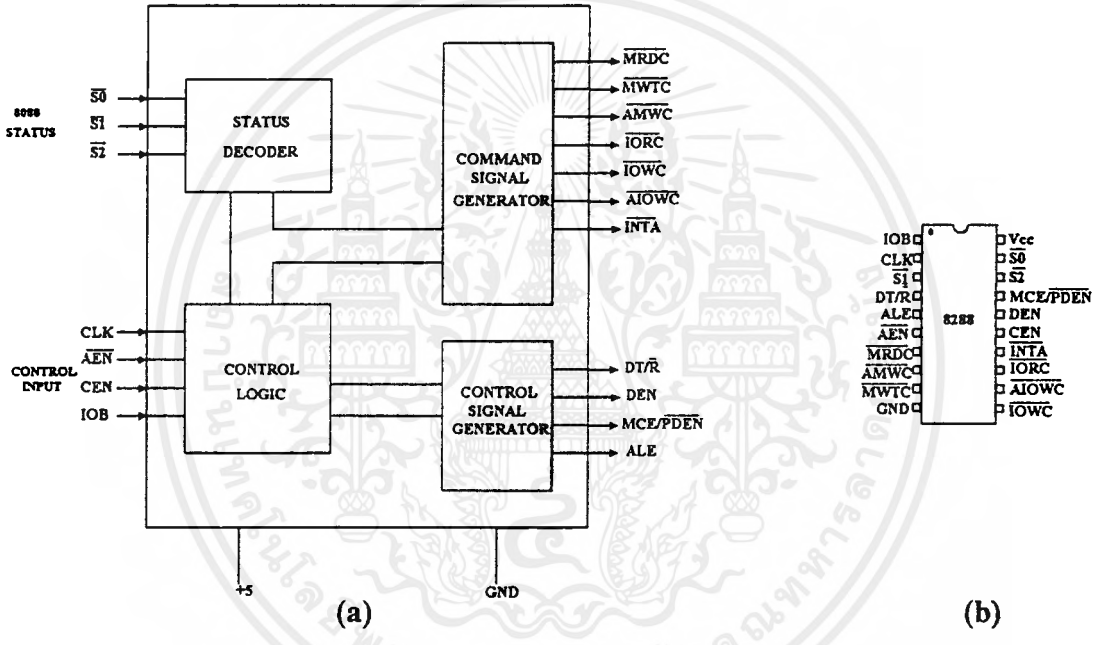
5. คุณสมบัติทางไฟฟ้า

แหล่งจ่ายไฟ +5โวลต์มีค่าเปลี่ยนแปลงได้ 10 % คือ 4.5-5.5โวลต์ที่อุณหภูมิห้อง 25 องศาเซลเซียส ดึงกระแสได้สูงสุด 340 มิลลิแอมป์ ดังตารางที่ 2.6

ในรูปที่ 2.3 สถาปัตยกรรมภายในของ 8088 ซึ่งจะถูกแบ่งออกเป็น 2 ส่วน ได้แก่ หน่วยการอินเตอร์เฟสและหน่วยปฏิบัติการ โดยทั่วไปหน่วยการอินเตอร์เฟสจะตอบสนองการทำงานของบัสทั้งหมดเช่น คำสั่งเฟต คำสั่งในการอ่านเขียนข้อมูลกับหน่วยความจำและอุปกรณ์อินพุต / เอาต์พุตส่วนหน่วยปฏิบัติการจะตอบสนองสำหรับคำสั่งการปฏิบัติงาน ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งสองส่วนนี้ทำงานสัมพันธ์กัน การเชื่อมต่อบัสออกไปภายนอกโดยให้บัสข้อมูล 8 บิต เป็นแบบสองทิศทางแอดเดรสบัสมีขนาด 20 บิต หน่วยการอินเตอร์เฟสจะตอบสนองต่อการกระทำของบัสภายนอกทั้งหมดหน่วยการอินเตอร์เฟส ประกอบไปด้วยฟังก์ชั่นส่วนของการเฟตแถวของลำดับคำสั่ง การเฟตโอเปอร์แรนด์และเก็บโอเปอร์แรนด์และบัสควบคุมส่วนของหน่วยการอินเตอร์เฟสภายในจะประกอบด้วยเซกเมนต์รีจิสเตอร์ , การติดต่อภายในรีจิสเตอร์ , พอยน์เตอร์ , ออฟเจ็กโค้ดและบัสควบคุมลอจิก องค์ประกอบเหล่านี้แสดงในรูปที่ 2.2



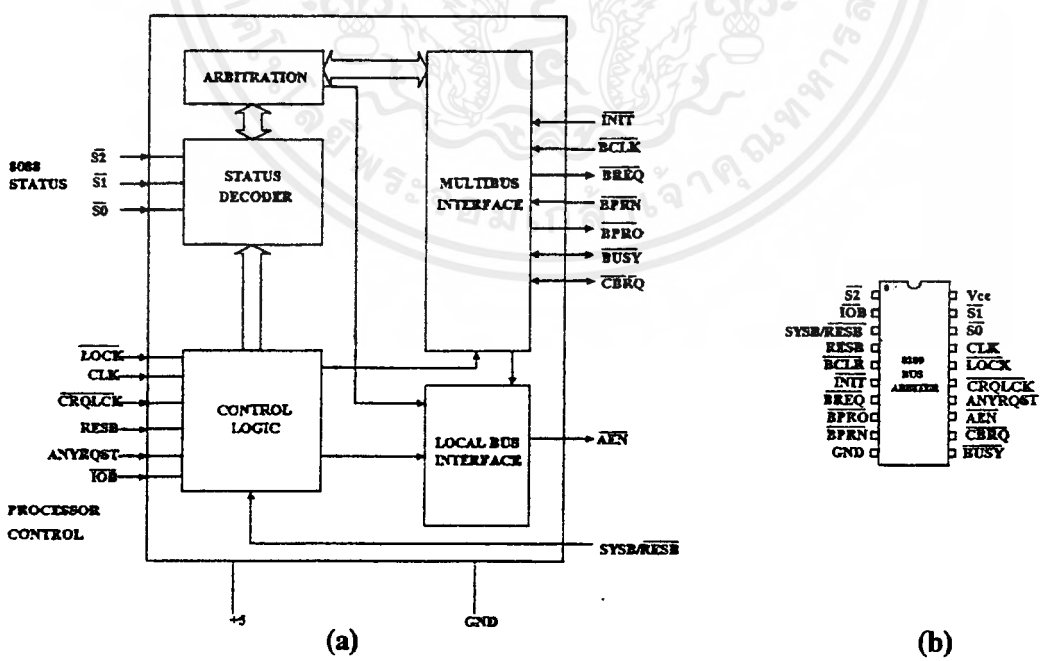
รูปที่ 2.6 (a) บล็อกไดอะแกรมของ8288, (b) ตำแหน่งขาของ 8288

หน่วยการอินเตอร์เฟสต้องเข้าถึงภายในเพื่อที่จะรู้ถึงของลำดับของกระแสคำสั่งโดยแถวของลำดับจะเฟตได้ถึง 6ไบต์ของรหัสคำสั่งอย่างไรก็ตามลำดับแถวของคำสั่งของหน่วยการอินเตอร์เฟสจะไม่มีวันเต็มที่เป็นเช่นนี้ก็เพราะว่ามันมีห้องมากกว่า 2ไบต์ในเวลาเดียวกัน หน่วยปฏิบัติการจะทำการอ่านเขียนโอเปอร์แรนด์จากหน่วยความจำ หน่วยการอินเตอร์เฟสจะทำงานแบบอิสระโดยจะมองไปที่ส่วนหัวของโปรแกรมก่อนที่จะเฟตคำสั่งถัดไป การเฟตของแถวนี้ มีลักษณะแบบ First in /First out (FIFO) คือเข้าก่อนออกก่อนด้วยข้อมูลแบบ 8 บิต ในการติดต่อกับหน่วยความจำหนึ่งครั้งหน่วยการอินเตอร์เฟส จะทำการเฟต 2 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 การกำหนดค่าบัสสถานะ

อินพุต			ซีพียูไซเคิล	การทำงาน 8288
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$		
0	0	0	Interrupt Acknowledge	$\overline{INTA}$
0	0	1	Read I/O Port	$\overline{IORC}$
0	1	0	Write I/O Port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	NONE
1	0	0	Instruction Fetch	$\overline{MRDC}$
1	0	1	Read Memory	$\overline{MRDC}$
1	1	0	Write Memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	NONE



รูปที่ 2.7 (a) บล็อกไดอะแกรมของ8289, (b) ตำแหน่งขาของ 8289

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่ 2.6 ระดับค่าของแรงดัน อินพุต / เอาต์พุต

สัญลักษณ์	ความหมาย	ค่าต่ำสุด	ค่าสูงสุด	เงื่อนไขทดสอบ
V_{IL}	Input low voltage	-0.5โวลต์	+0.8โวลต์	-
V_{IH}	Input high voltage	+2.0โวลต์	V_{CC} +0.5โวลต์	-
V_{OL}	Output low voltage	-	+0.45โวลต์	I_{OL}= 2.5 มิลลิแอมป์
V_{OH}	Output high voltage	+2.4โวลต์	-	I_{OH}= -400 มิลลิแอมป์

คำสั่งหลังจากที่ข้อมูลถูกเฟตที่ตำแหน่งสุดท้ายของส่วนคิวอินพุตจะทำการเลื่อนไปโดยอัตโนมัติ ในแบบ FIFO ไปยังส่วนที่ว่างใกล้กับเอาต์พุต หน่วยปฏิบัติการจะกระทำกับแถวห้องสุดท้าย โดยจะทำการอ่านมา 1ไบต์คำสั่งจากคิวเอาต์พุต

ถ้าแถวเต็มและหน่วยปฏิบัติการไม่ต้องการที่จะกระทำกับโอเปอร์แรนด์ในหน่วยความจำ หน่วยการอินเตอร์เฟสจะไม่กระทำกับบัสไซเคิลใดๆในเวลาพัก (ว่าง) นี้จะไม่มีบัสทำงานซึ่งมันอาจเกิดขึ้นระหว่างวงรอบของบัสอีกอย่างความจำหรืออินพุต/เอาต์พุตในครั้งแรกของหน่วยการอินเตอร์เฟสจะมีความพร้อมในการเฟตคำสั่งก่อนที่จะทำการเริ่มต้นวงรอบของการเขียนอ่านโอเปอร์แรนด์ ใหม่

หน่วยการอินเตอร์เฟสจะมีส่วนของตัวบวกให้เพื่อที่จะผลิตฟิสิคัลแอดเดรส 20 บิต ที่เอาต์พุตบนแอดเดรสบัสที่ตำแหน่งของฟิสิคัลแอดเดรสนี้ได้มาโดยการบวก 16 บิต เซกเมนต์แอดเดรสและ 16 บิต ออฟเซ็ตแอดเดรส สำหรับตัวอย่างคือ ตำแหน่งของ คำสั่งถัดไปได้มาโดยการรวมกันของ CS และ IP รีจิสเตอร์ ในส่วนหน่วยการอินเตอร์เฟสนี้จะตอบสนองกับการผลิตสัญญาณควบคุมบัสเช่น ในการอ่านเขียนหน่วยความจำและอินพุต/เอาต์พุตสัญญาณเหล่านี้จะถูกนำไปใช้ในการควบคุมหน่วยความจำและในบางส่วนของระบบอินพุตเอาต์พุต หน่วยปฏิบัติการจะต้องตอบสนองการถอดรหัสและปฏิบัติตามคำสั่งทั้งหมดจากรูปที่ 2.2 จะประกอบด้วย ส่วนหน่วยการคำนวณทางคณิตศาสตร์ , แฟล็กควบคุม , รีจิสเตอร์ทั่วไป 8 ตัว , รีจิสเตอร์ชั่วคราว

หน่วยปฏิบัติการจะทำการตัดแบ่งท่อนของคำสั่งจากส่วนหัวของแถวคำสั่งในหน่วยการอินเตอร์เฟสจะทำการถอดรหัส ถ้าจำเป็นจะให้สร้างตำแหน่งของโอเปอร์แรนด์และจะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 0 1 1 7

ป้อนผ่านไปในหน่วยการอินเตอร์เฟสถ้ามีโอเปอร์แรนด์จะกระทำในวงรอบของการเขียนอ่านบัส ระหว่างที่มีการปฏิบัติคำสั่งหน่วยปฏิบัติการจะทำการทดสอบแฟล็กของสถานะและแฟล็กควบคุมเพื่อที่จะทำการปรับเปลี่ยนตามผลของการปฏิบัติคำสั่ง ถ้าแถววางหน่วยปฏิบัติการจะคอยไบต์ของคำสั่งถัดไป เพื่อจะทำการเฟตและเลื่อนไปยังส่วนบนของแถว เมื่อหน่วยปฏิบัติการทำการปฏิบัติคำสั่ง Branch หรือ Jump มันจะย้ายการควบคุมไปยังตำแหน่งที่ตั้งตรงกับที่ระบุในคำสั่ง อย่างไรก็ตามเมื่อการกระทำนี้เกิดในหน่วยการอินเตอร์เฟสจะทำให้การรีเซ็ตแถวเกิดอย่างอัตโนมัติและหลังจากนั้นจะเริ่มเฟตคำสั่งจากตำแหน่งใหม่นี้เพื่อเพิ่มเติมให้กับแถว

2.2.4 รีจิสเตอร์ภายใน 8088 ไมโครโปรเซสเซอร์ 16 บิต

8088 มีกลุ่มของรีจิสเตอร์ภายในที่เข้าถึงได้ 4 กลุ่มได้แก่ รีจิสเตอร์ชี้คำสั่ง , รีจิสเตอร์ข้อมูล 4 ตัว , กลุ่มตัวชี้และอินเด็กรีจิสเตอร์และเซกเมนต์รีจิสเตอร์ 4 ตัว,รีจิสเตอร์ทั้งหมดมี 13 ตัว แต่ละตัวเป็นแบบ 16 บิตนอกจากนี้ยังมีรีจิสเตอร์ที่เพิ่มเข้าไปอีก 1 ตัวที่เป็นแบบ 16 บิต เรียกว่า รีจิสเตอร์สถานะซึ่งมีอยู่ 9 บิต ทำหน้าที่แสดงสถานะและการควบคุม

รีจิสเตอร์ใน 8088 จะประกอบด้วยรีจิสเตอร์ต้นทางหรือซอร์สและรีจิสเตอร์ปลายทางหรือเดสทิเนชัน, BX , SP , BP , SI และ DI ซึ่งสามารถที่จะนำมาเป็นตัวอ้างแอดเดรสได้ ดังรูปที่ 2.8

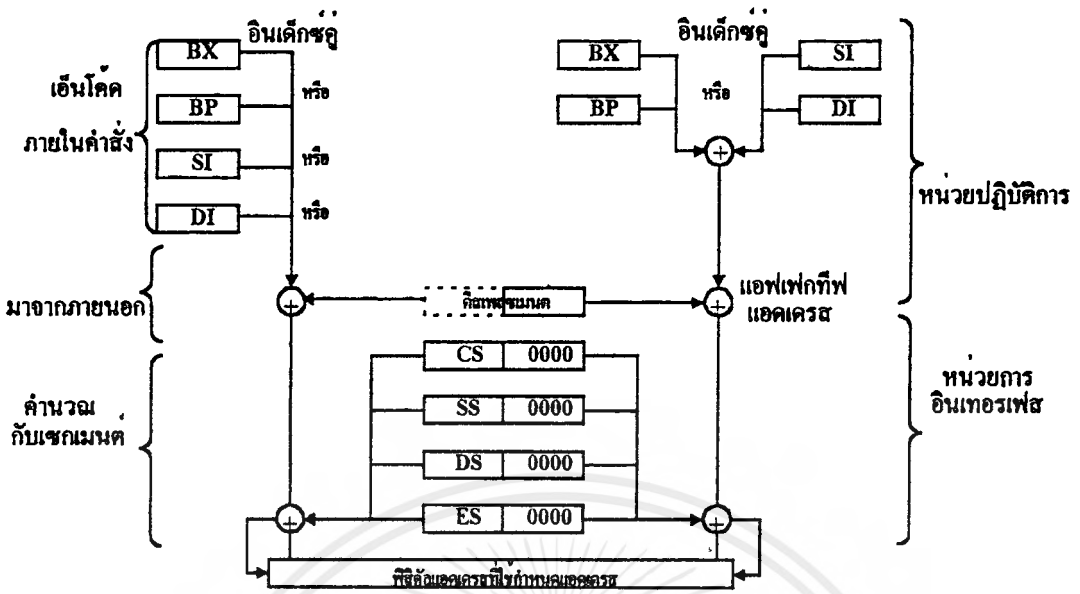
ผู้เขียนโปรแกรมส่วนใหญ่มักใช้รีจิสเตอร์ที่แน่นอนสำหรับเก็บค่าเฉพาะของโปรแกรมซึ่งอาจเซตไว้เป็นค่าตัวนับให้เป็นตัวชี้สแตก เป็นต้น 8088 ก็เตรียมรีจิสเตอร์สำหรับ จุดประสงค์พิเศษต่างๆไว้เรียบร้อยแล้วทำให้ชุดคำสั่งสั้นและมีประสิทธิภาพมากขึ้น

รีจิสเตอร์ขนาด 16 บิต 4 ตัว ซึ่งได้แก่ AX ,BX ,CX และ DX ยังถูกแบ่งออกเป็นครึ่งบน (AH,BH,CH,DH) และครึ่งล่าง (AL,BL,CL,DL) รีจิสเตอร์เหล่านี้อาจทำงานในลักษณะเป็นเวิร์ดหรือเป็นไบต์ก็ได้ ในกลุ่มนี้ถือว่าเป็นรีจิสเตอร์ข้อมูล

รีจิสเตอร์ในกลุ่มพอยน์เตอร์และอินเด็กซ์มักจะนำไปใช้ในการอ้างแอดเดรสและยังใช้ในการกระทำทางคณิตศาสตร์และลอจิกได้

รีจิสเตอร์ BP และ SP มักใช้ในการชี้สแตกเพื่อทำการเก็บตำแหน่งแอดเดรสกลับจากโปรแกรมย่อย โดยใช้หลักการเดียวกับซีพียู 8 บิต ส่วน BP นั้นจะใช้สำหรับเป็นตัวบวกเพื่อชี้ค่าสแตกอีกทีหนึ่งเพื่อจะเป็นวิธีที่จะอำนวยความสะดวกในการส่งค่าพารามิเตอร์เข้าสู่โปรแกรมโดยผ่านทางสแตก ซึ่งช่วยในการส่งค่าของตัวแปรระหว่างโปรแกรมได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 โครงสร้างของรีจิสเตอร์ของ 8088 ไมโครโปรเซสเซอร์ 16 บิต กับการคำนวณแอดเดรส

รีจิสเตอร์ SI และ DI ทำการติดต่อกับหน่วยความจำได้ซึ่งจะใช้งานได้ดีในกลุ่มคำสั่งพวกสตริงและการทำงานเป็นบล็อกหรือการเชื่อมโยงข้อมูลในโครงสร้างแบบอาเรย์อีกด้วย

ในการเฟตคำสั่งของ 8088 จะใช้ IP ขนาด 16 บิต เป็นตัวชี้ตำแหน่งแอดเดรสของคำสั่งถัดไปที่จะถูกทำงานซึ่งความจริงแล้วค่าใน IP ไม่ได้กำหนด ตำแหน่งโดยตรงแต่จะมีการคำนวณมาก่อนหน้านี้แต่จะทำการปฏิบัติในการชี้ตำแหน่งจริง หรือ ฟิสิคัลแอดเดรส นั้นจะใช้ IP และ โค้ดเซกเมนต์รีจิสเตอร์ร่วมกันที่เป็นเช่นนี้เพราะว่า 8088 มีรีจิสเตอร์และหน่วยความจำแบบ 16 บิต แต่ต้องการตำแหน่งของแอดเดรส 20 บิต ค่าออฟเซ็ตที่อยู่ในโค้ดเซกเมนต์จะต้องถูกนำมาร่วมกับ IP เพื่อจะใช้ในการอ้างตำแหน่ง ฟิสิคัลแอดเดรส ทุกครั้งเมื่อมีคำสั่งแบบเวิร์ด จะมีการเฟตจากหน่วยความจำ ที่หน่วยการอินเตอร์เฟสจะทำให้การเปลี่ยนแปลงค่าใน IP เพื่อให้มันชี้ไปยังคำสั่งของเวิร์ดถัดไปในหน่วยความจำโดยรีจิสเตอร์ทั่วไปมีด้วยกัน 4 ตัว ดังรูปที่ 2.2 ซึ่งอยู่ภายในหน่วยปฏิบัติการของ 8088 ระหว่างที่โปรแกรมทำการปฏิบัติงานอยู่ รีจิสเตอร์เหล่านี้จะถูกใช้ในการเก็บข้อมูลชั่วคราวประโยชน์ใช้ในการเก็บข้อมูลไว้ภายในแทนที่จะเก็บในหน่วยความจำ โดยจะทำงานได้เร็วกว่าและสามารถทำงานได้ทั้งแบบ 16 บิต (เวิร์ด)และ 8 บิต (ไบต์) ถ้ามีการอ้างแบบเวิร์ด ที่ตัวอ้างรีจิสเตอร์จะมีตัว อักษร X อยู่ข้างท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น AX,BX,CX,DX เป็นต้นในการอ้างแบบไบต์จะแบ่งออกเป็น 2 ชนิด คือ AH,AL โดยมีไบต์ละ 8 บิต ดาต้ารีจิสเตอร์แต่ละตัวสามารถที่จะใช้ในการทำงานแบบคำนวณและลอจิกได้เช่น การบวก (OR) , การแอนท์ (AND)

รีจิสเตอร์ใช้เป็นตัวชี้มี 2 ตัวได้แก่ SP และ BP ส่วนรีจิสเตอร์ที่ใช้ชี้อินเด็กซ์ก็มี 2 ตัว ได้แก่ SI , DI รีจิสเตอร์เหล่านี้ถูกใช้ในการเก็บตำแหน่งออฟเซ็ตของหน่วยความจำที่สัมพันธ์กับเซกเมนต์รีจิสเตอร์ค่าที่อยู่ในสามารถกระทำได้ทางซอฟแวร์รีจิสเตอร์ SP และ BP ใช้ในการชี้สแต็ก เพื่อที่จะทำการเก็บค่าตำแหน่งแอดเดรสก่อนที่จะกระโดดไปทำงานยังโปรแกรมย่อยโดยใช้หลักการเหมือนกับซีพียู 8 บิต BP รีจิสเตอร์ให้กับ SS รีจิสเตอร์เพื่อชี้ค่าสแต็กอีกครั้งหนึ่ง

อินเด็กซ์รีจิสเตอร์ใช้ในการเก็บค่าออฟเซ็ตแอดเดรสสำหรับโอเปอร์แรนด์เริ่มต้นและ DI รีจิสเตอร์ใช้ในการเก็บค่าออฟเซ็ตปลายทางในตำแหน่งโอเปอร์แรนด์สุดท้าย ตัวอย่างเช่นกลุ่มคำสั่งสตริงและการทำงานเป็นบล็อกจะต้องการออฟเซ็ตในส่วนของซอร์ตโอเปอร์แรนด์และเดสทิเนชั่นโอเปอร์แรนด์จะใช้รีจิสเตอร์เหล่านี้ เซกเมนต์รีจิสเตอร์ที่ตำแหน่งฟิสิคัลแอดเดรสของ 8088 มีขนาด 20 บิต แต่รีจิสเตอร์ และหน่วยความจำจะมีขนาด 16 บิต และต้องการอ้างให้ได้ 1 เมกะไบต์อย่างไรก็ตาม 8088 จะมองหน่วยความจำที่มีขนาดความจำเป็นเซกเมนต์ในหนึ่งเซกเมนต์จะชี้ได้ถึง 64 กิโลไบต์โดยแต่ละเซกเมนต์จะชี้ที่ตำแหน่งค่าเริ่มต้นของเซกเมนต์นั้นๆทั้งเซ็กเมนท์ของหน่วยความจำ สามารถทำงานที่เวลาเดียวกันโดยจะทำการเลือกใช้ โดยเซกเมนต์รีจิสเตอร์ซึ่งได้แก่ CS , DS , SS , ES รีจิสเตอร์ ตัวอย่างค่าในรีจิสเตอร์ CS จะเป็นตัวบอกจุดเริ่มต้นของ 64 กิโลไบต์ ในส่วนโค้ดเซกเมนต์ของหน่วยความจำตำแหน่งเริ่มต้นคือ ตำแหน่งล่างสุดโค้ดเซกเมนต์ของหน่วยความจำจะบรรจุคำสั่งของโปรแกรม DS จะเก็บค่าดาต้าเซกเมนต์ขณะนั้นๆ SS จะเก็บค่าสแต็กเซกเมนต์ขณะนั้นและ ES จะกำหนดเซกเมนต์ของข้อมูลรวมจุดสำคัญอีกจุดหนึ่งก็คือเซกเมนต์จะแสดงตำแหน่งเหมือนกับเป็นตารางโดยจะทำการเลื่อนไปทางซ้าย 4 บิต เพื่อที่จะกำหนดหรืออ้างตำแหน่งให้ครบ 20 เส้น โดยจุดเริ่มต้นของตารางจะต้องมี 4 บิต หลังสุด เป็น 0000H ในการติดต่อกับข้อมูล 1 ไบต์ หรือ 1 เวิร์ด 8088 จะเตรียมค่าออฟเซ็ตเพื่อใช้อ้างตั้งแต่จุดเริ่มต้นของตำแหน่งแต่ละเซกเมนต์ในตำแหน่งฟิสิคัลแอดเดรสจะได้มาจากการบวกค่าเซกเมนต์รีจิสเตอร์กับค่าออฟเซ็ต 16 บิต เช่น เซกเมนต์มีค่า E89FH และเราให้ออฟเซ็ตมีค่า 0003H จะทำให้ได้ฟิสิคัลแอดเดรสชี้ไปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

E89F0

\+

0003

E89F0 ตำแหน่งที่ชี้ไป

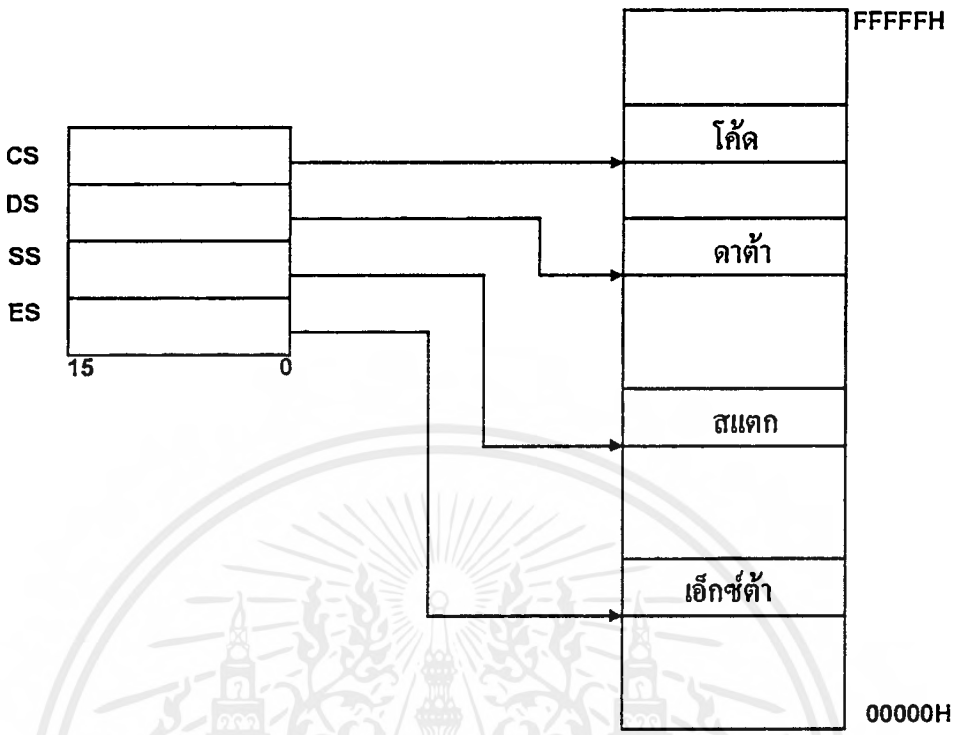
รีจิสเตอร์ 4 ตัวสุดท้ายเป็นเซกเมนต์รีจิสเตอร์ซึ่งได้แก่ CS ,DS ,SS ,ES ซึ่งนำมาใช้ในหลักการแบบบล็อกแอดเดสของ 8088 หน่วยความจำได้รับการจัดสรรเป็นสามส่วนใหญ่ๆคือ โค้ด , ข้อมูล และสแต็กสำหรับการเก็บค่าแอดเดรสกับคืนจากโปรแกรมย่อยเพื่อที่จะป้องกันการสับสนจึงกำหนดค่าที่แยกกันได้

8088 จะมองลักษณะหน่วยความจำโดยการแบ่งหน่วยความจำเป็นกลุ่มๆในรูปแบบของเซกเมนต์จะชี้ได้ถึง 64 กิโลไบต์ เซกเมนต์ทั้งสี่ตัวจะแสดงแอดเดรสเริ่มต้นของหน่วยความจำที่จะติดต่อ ด้วยดังรูปที่ 2.9 CS จะบรรจุค่าแสดงแอดเดรสเริ่มต้นของโปรแกรม DS จะเก็บค่าดาต้าเซกเมนต์ขณะนั้น SS ก็จะเก็บค่าสแตกเซกเมนต์ขณะนั้นและ ES จะกำหนดเซกเมนต์ของข้อมูลรวม

ตารางที่ 2.7 การใช้เซกเมนต์รีจิสเตอร์และออฟเซ็ต

ชนิดของการอ้างหน่วยความจำ	ค่าเซกเมนต์โดยปกติ	ค่าเซกเมนต์ที่เลือกได้	ค่าออฟเซ็ต
การเฟตคำสั่ง	CS	-	IP
การทำแสต็ก	SS	-	SP
กำหนดตัวแปร	DS	CS,ES,SS	ค่าแอดเดรสที่กำหนด
สตริงต้นทาง	DS	CS,ES,SS	SI
สตริงปลายทาง	ES	-	DI
เบสรีจิสเตอร์	SS	CS,ES,SS	ค่าแอดเดรสที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



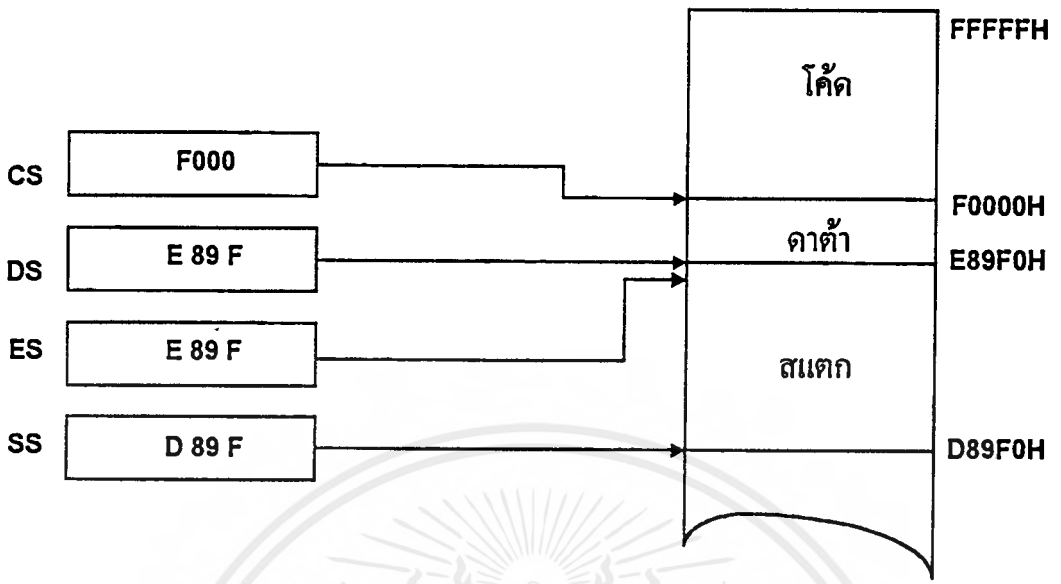
รูปที่ 2.9 เซกเมนต์รีจิสเตอร์ทั้งสี่จะแบ่งกลุ่มของหน่วยความจำโดยแต่ละเซกเมนต์จะชี้ตำแหน่งจุดเริ่มต้นของเซกเมนต์นั้น

จุดสำคัญอีกจุดหนึ่งก็คือ เซกเมนต์จะแสดงตำแหน่งเหมือนกับเป็นตารางโดยจะเลื่อนไปทางซ้าย 4 บิตเพื่อที่จะกำหนดหรืออ้างแอดเดรสให้ครบ 20 เส้น โดยจุดเริ่มต้นของตารางจะต้องมี 4 บิตหลังสุดเป็น 0 เช่น 00000H , 00010H , 00020H เป็นต้น จากรูปที่ 2.10 ค่าของ SS จะชี้ตำแหน่งของเซกเมนต์ค่า D89F0H สังเกตว่าส่วนพื้นที่ของหน่วยความจำอาจจะสลับหรือใช้บริเวณเดียวกันของหน่วยความจำได้เพื่อที่จะทำการติดต่อกับข้อมูลหนึ่งไบต์หรือหนึ่งเวิร์ดนั้นได้เตรียมค่าออฟเซ็ตเพื่อใช้อ้างตำแหน่งใดๆจะได้มาจากการบวกค่าเซกเมนต์กับค่าออฟเซ็ต

การจะใช้เซกเมนต์และค่าออฟเซ็ตตัวใดนั้นจะขึ้นอยู่กับชนิดของคำสั่งด้วยและตัวอย่างการทำงานของเซกเมนต์และออฟเซ็ตที่ใช้รีจิสเตอร์ต่างๆได้ดังรูปที่ 2.11

แฟล็กรีจิสเตอร์มีขนาด 16 บิต ภายในหน่วยปฏิบัติการและจะมีผลเพียง 9 บิต โดย 6 บิต จะเป็นแฟล็กสถานะและที่เหลือเป็นแฟล็กควบคุม 3 บิต

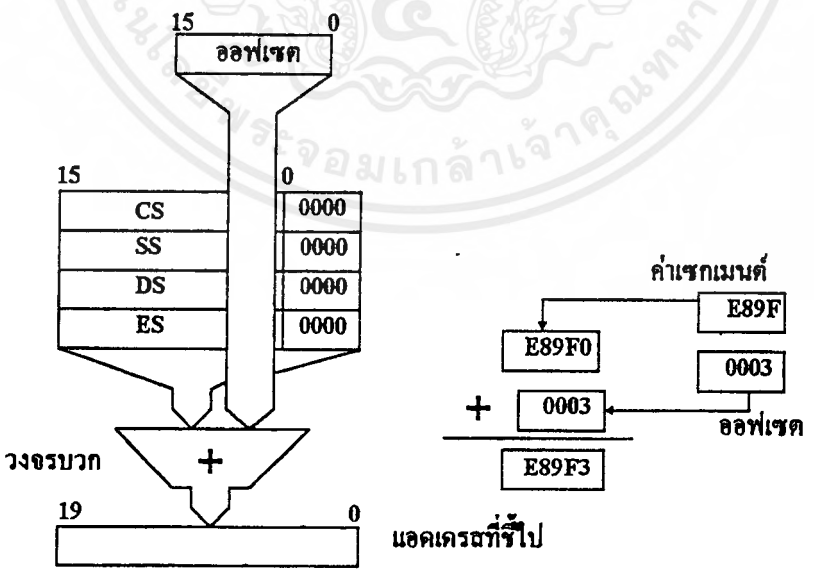
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 เซกเมนต์รีจิสเตอร์แต่ละตัวจะประกอบด้วยแอดเดรส 16 บิต ทางด้านบนส่วนสี่บิตล่างที่เหลือจะถือว่าเป็นศูนย์

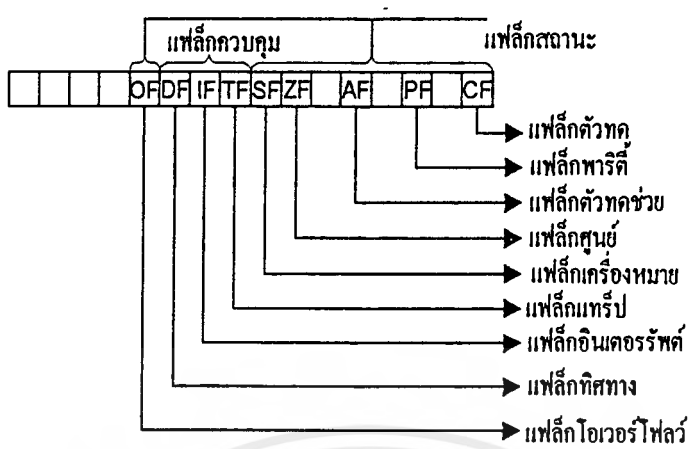
1. แฟล็กสถานะ

- CF แฟล็กตัวทดบอกการทดหรือการยืดของบิตสูงสุดของผลลัพธ์
- PF แฟล็กพาริตี้ใช้แสดงจำนวนคู่หรือคี่ ของตัวเลขที่เป็น "1" ถ้าคู่จะถูก เซ็ต



รูปที่ 2.11 การคำนวณให้ได้ 20 บิต เพื่อใช้ในการอ้างแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แฟล็กของ 8086 ที่ใช้ในการทดสอบการทำงานของซีพียู

- AF แฟล็กทดช่วยถูกเซ็ตเมื่อมีตัวทดจาก 4 บิต ล่างไป 4 บิตบนหรือกลับกัน

- ZF แฟล็กศูนย์ถูกเซ็ตเมื่อมีผลการคำนวณและลอจิกเป็นศูนย์

- SF แฟล็กเครื่องหมายถูกเซ็ตเมื่อผลลัพธ์เป็นเลขลบ

- DF แฟล็กโอเวอร์โฟลวถูกเซ็ตเมื่อมีความผิดพลาดในการดำเนินหรือเกินช่วงค่าที่กำหนด

2. แฟล็กควบคุม

- TF ถ้าถูกเซ็ตจะมีการทำงานในแบบลำดับทีละขั้นซึ่งจะมีประโยชน์มากตอนแก้ไขโปรแกรม

- IF ถ้าถูกเซ็ตจะยอมรับการอินเตอร์รัพต์จากภายนอก (ขา INTR)

- DF จะกำหนดทิศทางของกลุ่มคำสตริงถ้าถูกเซ็ตจะมีทิศทางการทำงานจากตำแหน่งมากไปหาน้อยแต่ถ้าเป็นศูนย์จะทำจากน้อยไปหามาก

2.3 บัสแอดเดรสและบัสข้อมูล

2.3.1 AD0-AD7 (Address / Data Bus ; ขา 16-9)

เส้นสัญญาณทั้ง 8 เส้นนี้ จะถูกใช้สำหรับส่งแอดเดรสหรือข้อมูลให้ กับหน่วยความจำหรือวงจรอินพุต / เอาต์พุต ที่ทำงานร่วมกับ 8088 ซึ่งเส้นสัญญาณทั้ง 8 เส้นนี้จะใช้หลักการของการมัลติเพลกซ์ กล่าวคือในช่วงเวลาของคล็อกลูกแรกในบัสไซเคิล (บัสไซเคิลคือช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการส่ง ผ่าน ข้อมูลหนึ่งๆของ 8088) 8088 จะใช้ขา AD0-AD7 นี้เป็นบัสแอดเดรสโดยจะส่งแอดเดรส A0 - A7 ส่วนในช่วงเวลา (ในบัสไซเคิลเดียวกัน) ที่ถัดจากคล็อกลูกแรกของบัสไซเคิล 8088 ก็จะใช้ขา AD0 - AD7 นี้เป็นบัสข้อมูลเพื่อทำการส่งผ่านข้อมูลต่างๆระหว่างหน่วยความจำหรืออุปกรณ์อินพุต / เอาต์พุตกับ 8088

สัญญาณที่ขา AD0-AD7 นี้มีค่าเป็นความต้านทานสูงในขณะที่ 8088 ตอบรับการขออินเตอร์รัพต์หรือในขณะที่ 8088 ตอบรับการขอใช้บัสจากภายนอก

2.3.2 A8-A15 (Address Bus ; ขา 8-2)

เส้นสัญญาณทั้ง 8 เส้นนี้จะถูกใช้สำหรับส่งแอดเดรสให้กับหน่วยความจำหรือวงจรอินพุต/เอาต์พุตซึ่ง 8088 จะส่งแอดเดรส (A8-A15) ของหน่วยความจำหรืออินพุต/เอาต์พุต ที่ต้องการติดต่อด้วยออกมาในช่วงของคล็อกลูกที่หนึ่งในแต่ละบัสไซเคิลและจะรักษาสถานะของแอดเดรสนี้ไว้จนสิ้นสุดบัสไซเคิลขึ้น

2.3.3 A16/S3 (Address/Status ; ขา 38)

ในช่วงเวลาคล็อกลูกที่หนึ่งในแต่ละบัสไซเคิลของ 8088 จะส่งแอดเดรส A16 ออกมายังขานี้แต่ถ้าคำสั่งที่ 8088 กำลังทำงานอยู่นั้นเป็นคำสั่งที่เกี่ยวกับอินพุต/เอาต์พุต (คือถ้า 8088 ต้องการติดต่อกับพอร์ตอินพุต/เอาต์พุต) เช่นคำสั่ง IN และ OUT 8088 ก็จะปรับสภาวะขานี้ในช่วงเวลาของคล็อกลูกที่หนึ่งให้เป็นลอจิก "0" ส่วนในช่วงเวลาที่ถัดจากคล็อกลูกแรกของบัสไซเคิลจนสิ้นสุดช่วงเวลาของไซเคิลนั้น 8088 จะใช้สัญญาณในส่วนนี้ร่วมกับ A17/S4 เพื่อแสดงรีจิสเตอร์เซกเมนต์ใดที่ถูกใช้งานอยู่ในไซเคิลนั้น

2.3.4 A17/S4 (Address/Status ; ขา 37)

ในช่วงเวลาของคล็อกลูกที่หนึ่งในแต่ละบัสไซเคิล 8088 จะส่งแอดเดรส A17 ออกมาที่ขานี้แต่ถ้าคำสั่งที่ 8088 กำลังทำงานอยู่นั้นเป็นคำสั่งที่เกี่ยวกับอินพุตเอาต์พุต 8088 ก็จะปรับให้สถานะของขานี้เป็นลอจิก "0" ส่วนในช่วงเวลาที่ถัดจากคล็อกลูกแรกของบัสไซเคิลจนสิ้นสุดช่วงเวลาของบัสไซเคิลนั้น 8088 จะใช้เส้นสัญญาณนี้ร่วมกับ A17/S4 เพื่อแสดงว่ารีจิสเตอร์เซกเมนต์ ใดถูกใช้งานอยู่ในไซเคิลนั้น

สำหรับสัญญาณที่แสดงเซกเมนต์ที่ 8088 ใช้งานอยู่นี้จะช่วยให้เราสามารถขยายจำนวนหน่วยความจำสำหรับแต่ละรีจิสเตอร์เซกเมนต์ให้เป็น 1 เมกะไบต์ ได้ซึ่งจะเป็นผลให้จำนวนของหน่วยความจำในระบบเพิ่มขึ้นเป็น 4 เมกะไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 การใช้งานรีจิสเตอร์เซกเมนต์

A17/S4	A16/S3	รีจิสเตอร์เซกเมนต์
0	0	Extra Segment
0	1	Stack Segment
1	0	Code Segment or no Segment
1	1	Data Segment

2.3.5 A18/S5 (Address Status ; ขา 36)

ในช่วงเวลาของคล็อกลูกที่หนึ่งในแต่ละบัสไซเคิล 8088 จะส่งแอดเดรส A18 ออกมาที่ขานี้ แต่ถ้าคำสั่งที่ 8088 กำลังทำงานอยู่นั้น เป็นคำสั่งที่เกี่ยวกับอินพุต/เอาต์พุตจะปรับให้เป็นสถานะของขานี้เป็นลอจิก"0"ส่วนในช่วงเวลาที่ถัดจากคล็อกลูกแรกของบัสไซเคิลจนสิ้นสุดช่วงเวลาของบัสไซเคิลนั้น 8088 จะใช้เส้นสัญญาณนี้แสดงสถานะของแฟล็ก IF

2.3.6 A19/S6 (Address Data Bus ; ขา 35)

ในช่วงเวลาของคล็อกลูกที่หนึ่งในแต่ละบัสไซเคิลนั้น 8088 จะส่งแอดเดรส A19 ออกมาที่ขานี้แต่ถ้าคำสั่งที่ 8088 กำลังทำงานอยู่นั้นเป็นคำสั่งที่เกี่ยวกับอินพุต/เอาต์พุตก็จะปรับให้สถานะของขานี้เป็นลอจิก "0" ส่วนในช่วงเวลาที่ถัดมาจากคล็อกลูกแรกของบัสไซเคิลจนสิ้นสุดช่วงเวลาของบัสไซเคิลนั้นถ้า 8088 เป็นผู้ที่ควบคุมบัสต่างๆอยู่ก็จะให้เอาต์พุตที่ขานี้เป็นลอจิก "0" แต่ถ้าเป็นช่วงที่ 8088 ทำการตอบรับการขอใช้บัส 8088 ก็จะปรับให้เอาต์พุตที่ขานี้อยู่ในสภาวะความต้านทานสูง

2.4 บัสควบคุมและแสดงสถานะ

สำหรับหน้าที่ของขาต่างๆที่อยู่ในบัสควบคุมและแสดงสถานะนั้น จะแบ่งออกเป็น 2 กลุ่ม คือ กลุ่มที่มีหน้าที่การทำงานไม่ขึ้นกับสถานะของลอจิกที่ขา MN / $\overline{MX}$ และกลุ่มที่มีหน้าที่การทำงานขึ้นอยู่กับสถานะของลอจิกที่ขา MN / $\overline{MX}$

ในส่วนแรกนี้จะกล่าวถึงกลุ่มที่มีการทำงานที่ไม่ขึ้นกับสถานะลอจิกที่ขา MN / $\overline{MX}$ ของ 8088 ซึ่งประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 $\overline{RD}$ (Read ; ขา 32)

สัญญาณที่ขานี้จะเป็นลอจิก "0" ในช่วงเวลาที่ 8088 อ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุตโดยที่สถานะของขา $\overline{S2}$ - IO / $\overline{M}$ (ขา 28) จะแสดงว่า 8088 ต้องการอ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุต

2.4.2 READY (ขา 22)

เป็นอินพุตที่หน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุตส่งให้กับ 8088 เพื่อให้ 8088 ทราบว่าหน่วยความจำหรืออุปกรณ์นั้นพร้อมที่จะส่งหรือรับข้อมูลจาก 8088 แล้วหรือไม่ ในกรณีที่อินพุตที่ขานี้ได้รับลอจิก "0" ซึ่งทำให้ 8088 รอจนกว่าหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุต พร้อมที่จะส่งหรือรับข้อมูลจาก 8088 (คือ 8088 จะรอจนกว่าสถานะที่ขา READY กลับไปเป็น "1")

2.4.3 $\overline{TEST}$ (ขา 23)

ขานี้เป็นอินพุตที่ 8088 จะใช้เฉพาะในกรณีที่อยู่ในระหว่างการทำงานในคำสั่ง WAIT เท่านั้น โดยเมื่อ 8088 ทำงานในคำสั่ง WAIT 8088 จะหยุดรอจนกว่าขา $\overline{TEST}$ นี้จะได้รับลอจิกเป็น "0"

2.4.4 INTR (Interrupt Request ; ขา 18)

ขานี้เป็นอินพุตที่วงจรหรืออุปกรณ์ภายนอกใช้เพื่ออินเตอร์รัพต์ 8088 ซึ่ง 8088 จะทำการตรวจสอบระดับสัญญาณที่ขานี้ในช่วงเวลาของคล็อกลูกสุดท้ายในการทำงานในแต่ละคำสั่งในกรณีที่ 8088 ตรวจพบว่าระดับลอจิกที่ขา INTR นี้เป็น "1"

2.4.5 NMI (Non - Maskable Interrupt ; ขา 17)

ขานี้เป็นมีสถานะเป็นอินพุตของ 8088 ที่อุปกรณ์ภายนอกใช้เพื่อขออินเตอร์รัพต์แบบ Non-Maskable (NMI) จาก 8088 ซึ่งการอินเตอร์รัพต์แบบ MNI นี้ 8088 จะทำการตอบรับในทุกกรณี

2.4.6 RESET (ขา 21)

เป็นขาอินพุตที่ใช้สำหรับรีเซ็ต 8088 โดยทั่วไปจะต่อจากเอาต์พุต 8284 ตัวกำเนิดสัญญาณนาฬิกาหลังจากที่ 8088 ถูกการรีเซ็ตมีขั้นตอนดังนี้ 1.8088 จะเคลียร์การีจิสเตอร์แฟล็กที่มีอยู่ให้เป็น 0000H ซึ่งจะเป็นผลให้ 8088 ทำการ ดิสเอเบิลการขออินเตอร์รัพต์แบบ Maskable และเข้าสู่การทำงานในโหมดปฏิบัติการครั้งละ ลำดับขั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.8088 จะเคลียร์ค่ารีจิสเตอร์แฟล็กที่มีอยู่ให้เป็น 0000H ซึ่งจะเป็นผลให้ 8088 ทำการดิสเอเบิลการขออินเตอร์รัพต์แบบ Maskable และเข้าสู่การทำงานในโหมดปฏิบัติการครั้งละลำดับขั้น

2.8088 จะเคลียร์ค่าของรีจิสเตอร์ DS , SS , ES และ PC ให้เป็น 0000H

3.8088 จะเซ็ตค่าของรีจิสเตอร์ CS ให้เป็น 0FFFFH

จากการทำงานตามขั้นตอนข้างต้นจะทำให้ 8088 กระโดดไปทำงานในตำแหน่งแอดเดรสที่ 0FFFFH

สำหรับในส่วนต่อไปจะกล่าวถึง กลุ่มที่หน้าที่การทำงานขึ้นอยู่กับสถานะของลอจิกที่ขา MN / $\overline{MX}$ ซึ่งจะแสดงดังตารางที่ 2.9

2.4.7 $\overline{S0}$, $\overline{DEN}$ (Status ; data Enable ; ขา 26)

โหมด Mamimum ขานี้จะถูกใช้เป็นขา S0 ในการแสดงสถานะการทำงานของ 8088 ซึ่งโหมด Minimum ขานี้จะถูกใช้งานเป็น $\overline{DEN}$ ซึ่ง 8088 จำทำให้ขา $\overline{DEN}$ แอคทีฟเมื่อ 8088

ตารางที่ 2.9 กลุ่มของสัญญาณที่ทำงานขึ้นกับสัญญาณ MN / $\overline{MX}$

ตำแหน่ง	หน้าที่การทำงาน	
	โหมด Maximun	โหมด Minimum
26	S0	DEN
27	S1	DT/R
28	S2	IO/M
31	RQ/GT0	HOLD
30	RQ/GT1	HLDA
25	QS0	ALE
24	QS1	INTA
29	LOCK	WR
34	-	SSO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.8 S1 , DT / $\overline{R}$ (Status ; Data Transmit / Receive ; ขา 27)

โหมด Maximum ขานี้จะถูกใช้เป็น S1ซึ่งจะถูกใช้ร่วมกับ S2 และ S0 ในการแสดงสถานะการทำงานของ 8088

โหมด Minimum ขานี้จะถูกใช้งานเป็น DT / $\overline{R}$ ซึ่งจะใช้เพื่อบอกอุปกรณ์ภายนอกให้ทราบว่า 8088 ต้องการส่งข้อมูลหรือรับข้อมูลบนบัสข้อมูล

2.4.9 S2 , IO / $\overline{M}$ (Status ; IO or Memory Access ; ขา 28)

โหมด Maximum ขานี้ใช้เป็น S2 ซึ่งใช้ร่วมกับ S0 และ S1ในการแสดงสถานะการทำงาน ของ 8088 ดังตารางที่ 2.10 ข้างล่างนี้

โหมด Minimum ขานี้จะถูกใช้งานเป็น IO / $\overline{M}$ โดยขานี้จะให้เอาต์พุตลอจิกเป็น "1" ในระหว่างการติดต่อกับอุปกรณ์ I/O และจะให้เอาต์พุตเป็นลอจิก "0" ในระหว่างการติดต่อกับหน่วยความจำ

ตารางที่ 2.10 สถานะการทำงานของสัญญาณ S2 S1 S0

S2	S1	S0	สถานะการทำงาน
0	0	0	อินเตอร์รัพต์การขอใช้บัส
0	0	1	อินพุต/เอาต์พุตการอ่านข้อมูล
0	1	0	อินพุต/เอาต์พุตการเขียนข้อมูล
0	1	1	Halt
1	0	0	การชี้แฟล็กรีจิสเตอร์
1	0	1	หน่วยความจำการอ่าน
1	1	0	หน่วยความจำการเขียน
1	1	1	-

2.4.10 QS0 ; ALE (Queue Status ; Address Latch Enable)

โหมด Maximum ขานี้จะถูกใช้เป็น QS0 ซึ่งจะใช้งานร่วมกับ QS1 เพื่อแสดงสถานะของชุดเก็บคำสั่งของ 8088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด Minimum ขานี้ถูกใช้งานเป็น ALE เพื่อแสดงให้วงจรภายนอกทราบว่าข้อมูลที่อยู่บนบัสแอดเดรสขณะนี้เป็นค่าแอดเดรสที่ 8088 ต้องการติดต่อด้วย

2.4.11 QS1 ; $\overline{\text{INTA}}$ (Queue Statua ; Interrupt Acknowledge ; ขา 24)

โหมด Maximum ขานี้จะถูกใช้เป็น QS1 ซึ่งจะใช้ร่วมกับขา QS0 เพื่อแสดงสถานะชุดเก็บคำสั่งของ 8088 ดังตารางที่ 2.11 ข้างล่างนี้

โหมด Minimum ขานี้จะถูกใช้งานเป็น $\overline{\text{INTA}}$ โดยที่ 8088 จะทำให้เอาต์พุตที่ลอจิกนี้เป็น "0" ในระหว่างกระบวนการตอบสนองการอินเตอร์รัพต์ของอุปกรณ์ภายนอก

2.4.12 $\overline{\text{RQ}}$ / $\overline{\text{GT0}}$; HOLD (Request / Grant 0 ; Hold ; ขา 31)

โหมด Maximum ขานี้จะถูกใช้เป็น $\overline{\text{RQ}}$ / $\overline{\text{GT0}}$ สำหรับขานี้ของ 8088 จะถูกใช้เพื่ออุปกรณ์ภายนอกหรืออุปกรณ์ควบคุมบัส สามารถที่จะควบคุมใช้งานบัสต่างๆ ในระบบแทนซีพียู 8088 ได้

โหมด Minimum ขานี้จะถูกใช้งานเป็น HOLD โดยถ้ามีสัญญาณลอจิก "1" เข้ามาที่ขานี้ 8088 ก็จะเข้าสู่สภาวะ HOLD และจะส่งเอาต์พุต "1" ออกมาที่ขา HLDA เพื่อให้อุปกรณ์ภายนอกทราบว่า 8088 เข้าสู่สภาวะ HOLD แล้ว

ตารางที่ 2.11 สถานะ QS0 QS1 ในการเก็บชุดคำสั่ง

QS0	QS1	สถานะของส่วนเก็บชุดคำสั่ง
0	0	-
0	1	ไบต์แรกข้อคำสั่งถูกเอ็กซิคิวต์
1	0	ส่วนเก็บชุดคำสั่งว่าง
1	1	ไบต์ต่อไปของคำสั่งที่ถูกอ่านจากส่วนเก็บชุดคำสั่ง

2.4.13 $\overline{\text{RQ}}$ / $\overline{\text{GT1}}$; HLDA ; (Request / Grant 1 ; Holdacknowledge ; ขา 30)

โหมด Maximum ขานี้จะมีลักษณะการทำงานเหมือนกับขา $\overline{\text{RQ}}$ / $\overline{\text{GT0}}$ แต่มีข้อแตกต่างกันตรงที่ว่าจะมีระดับสูงกว่า $\overline{\text{RQ}}$ / $\overline{\text{GT1}}$

โหมด Minimum ขานี้จะถูกใช้งานเป็น HLDA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.14 LOCK ; $\overline{WR}$ (Lock ; Writa Signal ; ขา 29)

โหมด Maximum ขานี้จะถูกใช้งานเป็น LOCK โดยเมื่อ 8088 พบ LOCK 8088 ก็จะทำให้เอาต์พุตที่ลอจิกนี้ เป็น "0" และจะคงสภาวะนี้ไว้จนกระทั่ง 8088 เสร็จจากการปฏิบัติคำสั่ง LOCK เพื่อแสดงว่าตอนนี้ 8088 ไม่พร้อมที่จะให้อุปกรณ์ภายนอกเข้ามาควบคุมบัส

โหมด Minimum ขานี้จะถูกใช้เป็นขา $\overline{WR}$ ซึ่ง 8088 จะทำให้สัญญาณนี้แอดทีฟในช่วงของคล็อก T2 ,T3 และ TW

2.5 บัสจ่ายกำลังและฐานเวลา

บัสในส่วนนี้จะใช้สำหรับจ่ายไฟให้กับ 8088 และป้อนสัญญาณนาฬิกาให้กับ 8088 เพื่อควบคุมการทำงานของ 8088

2.5.1 CLK (Clock; ขา 19)

สัญญาณคล็อกที่ป้อนให้กับขา CLK นี้ จะถูกใช้สำหรับควบคุมจังหวะการทำงานของส่วนต่างๆ ภายใน 8088 สัมพันธ์ กัน

2.5.2 Vcc (ขา 40)

เป็นขาที่ใช้สำหรับจ่ายไฟให้กับ 8088 โดยแรงดันที่ป้อนให้กับขา Vcc ของ 8088 มีค่าเท่ากับ +5โวลต์

2.5.3 GND (Ground ; ขา 1 และ 20)

สำหรับ 8088 จะมีขาGND 2 ขา คือ 1 และ 20

2.6 การแสดงแอดเดรสของ 8088

สำหรับไมโครโปรเซสเซอร์ 8088 นี้จะมีเส้นแสดงของแอดเดรสจำนวน 20 เส้นคือ A0-A19 ดังนั้นจึงทำให้ 8088 สามารถที่จะอ้างหน่วยความจำได้ 1024 กิโลไบต์ ในขณะที่ไมโครโปรเซสเซอร์ขนาด 8 บิตทั่วไปเช่น Z80 นั้นสามารถใช้หน่วยความจำที่มีจำกัดได้เพียง 64 กิโลไบต์ เท่านั้นดังนั้นเพื่อความสะดวกในการเขียนโปรแกรม 8088 โดยการแบ่งแอดเดรสทั้ง 20 บิต ออกเป็นเซกเมนต์และออฟเซ็ต ซึ่งสำหรับในกรณีที่แอดเดรสเป็นเซกเมนต์นั้นจะเป็นแอดเดรสขนาด 16 บิต ทั้งคู่ส่วนค่าแอดเดรสที่แท้จริงนั้นจะได้จากการนำเอาค่าแอดเดรสที่เป็น เซกเมนต์และออฟเซ็ต มารวมกันโดยมีวิธีการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XXXX XXXX XXXX XXXX 0000 : แอดเดรสเซสเมนต์
+
0000 YYYY YYYY YYYY YYYY : แอดเดรสออฟเซ็ต

XXXX ZZZZ ZZZZ ZZZZ ZZZZ : แอดเดรสที่แท้จริง

ในที่นี้ X , Y และ Z เป็นเลขฐานสอง

2.7 บัสไซเคิลของระบบ

บัสไซเคิลคือ ขบวนการของสัญญาณที่เกิดขึ้นในช่วงของการส่งผ่านข้อมูลกันระหว่างหน่วยความจำ , อุปกรณ์อินพุต/เอาต์พุตและไมโครโปรเซสเซอร์ 8088 ดังนั้นในการออกแบบวงจรอินเตอร์เฟสเราต้องทราบไซเคิลของการส่งข้อมูลในลักษณะต่างๆก่อน

สำหรับบัสไซเคิลของระบบนั้น เราสามารถที่จะแบ่งออกตามลักษณะของอุปกรณ์ที่สร้างบัสไซเคิลได้ 2 กลุ่ม

1.บัสไซเคิลที่สร้างขึ้นโดย 8088 สำหรับบัสไซเคิลในกลุ่มนี้จะเป็นบัสไซเคิลที่สร้างขึ้นในการส่งผ่านข้อมูลระหว่าง 8088 กับอินพุตเอาต์พุตหรือระหว่าง 8088 กับหน่วยความจำ โดยในบัสไซเคิลเหล่านั้น 8088 จะส่งแอดเดรสของหน่วยความจำหรือพอร์ตที่จะต้องการติดต่อด้วยออกมาบนแอดเดรสบัส,ส่งหรือรับข้อมูลจากบัสข้อมูลและส่งสัญญาณควบคุมที่จำเป็นออกมาบนบัสควบคุมซึ่งบัสไซเคิลของกลุ่มเหล่านี้จะแบ่งออกเป็น 5 ชนิด คือบัสไซเคิลในการอ่านข้อมูลจากหน่วยความจำ,บัสไซเคิลในการเขียนข้อมูลลงในหน่วยความจำ,บัสไซเคิลในการรับข้อมูลจากพอร์ต,บัสไซเคิลการเขียนข้อมูลลงพอร์ตและบัสไซเคิลการตอบรับการอินเตอร์รัพต์

2.บัสไซเคิลที่สร้างขึ้นโดย DMA คอนโทรลเลอร์ สำหรับบัสไซเคิลนี้จะเกิดขึ้นจากกระบวนการ DMA ใน 8088 จะทำให้แอดเดรสบัส,บัสข้อมูล,บัสควบคุมบางเส้นมีสถานะความต้านทานสูงซึ่งจะทำให้ 8237-5 DMA สามารถควบคุมบัสต่างๆได้จากนั้นก็จะสร้างบัสไซเคิลเพื่อให้อุปกรณ์อินพุต/เอาต์พุตสามารถรับส่งข้อมูลได้โดยตรง

ในที่นี้จะกล่าวถึงบัสของข้อมูลที่สร้างขึ้นโดย 8088 เท่านั้นเพื่อใช้ประกอบในการทำความเข้าใจในวิชาโครงงาน

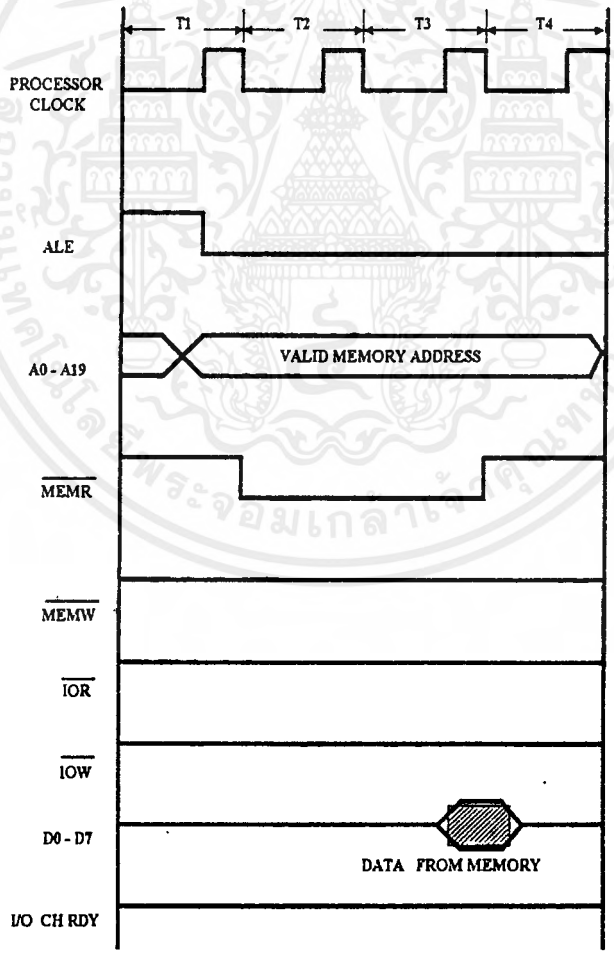
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1บัสไซเคิลที่สร้างขึ้นโดย 8088

การทำงานในแต่ละบัสไซเคิลของ 8088 นั้นจะใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อกที่ป้อนให้กับ 8088 จำนวน 4 ลูก สำหรับ 8088 จะมีความถี่ประมาณ 4.77 MHz หรือมีคาบเวลาของคล็อก 1 ลูก ประมาณ 210 nsec ดังนั้นใน 1 บัสไซเคิล จะใช้เวลาประมาณ 4 × 210 ได้เท่ากับ 840 nsec

1.บัสไซเคิลในการอ่านข้อมูลจากหน่วยความจำ

8088 จะทำการสร้างบัสไซเคิลนี้ขึ้นในช่วงที่ 8088 ทำการเฟตช์คำสั่งหรืออ่านข้อมูลจากหน่วยความจำสำหรับขบวนการของสัญญาณที่เกิดขึ้นในบัสไซเคิลนี้สามารถจะแสดงได้ดังรูป



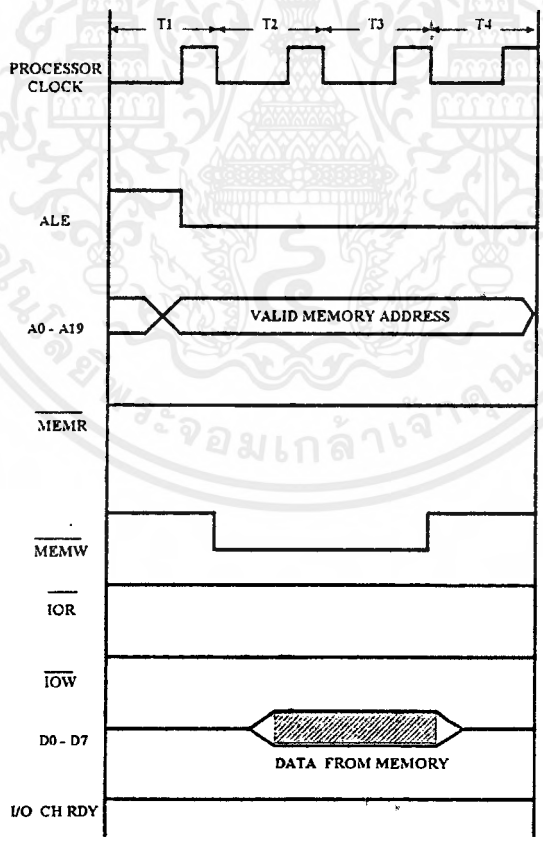
รูปที่ 2.13 บัสไซเคิลของการอ่านข้อมูลจากหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเริ่มต้นในช่วงคล็อกที่หนึ่ง T1 ซึ่งเป็นช่วงเวลาที่สัญญาณ ALE แอดทีฟ "1" สัญญาณ ALE นี้จะถูกใช้เพื่อแสดงให้อุปกรณ์ที่ทำงานร่วมกับ 8088 ทราบว่าข้อมูลที่อยู่บนบัสข้อมูลแอดเดรสในช่วงขอบขาลงของสัญญาณ ALE เป็นแอดเดรสของหน่วยความจำที่ 8088 ต้องการจะติดต่อด้วยเป็นการทำให้หน่วยความจำมีค่าแอดเดรสตรงกับค่าแอดเดรสที่มีอยู่บนบัสแอดเดรสนั้นส่งของมูลออกมาบนบัสข้อมูล เมื่อหน่วยความจำส่งข้อมูลออกมาบนบัสข้อมูลแล้ว 8088 จะทำการอ่านข้อมูลนั้นในช่วงเริ่มต้นของ T4 จากนั้นสัญญาณ $\overline{\text{MEMR}}$ จะถูกปรับให้กลับเป็นลอจิก "1" และจะสิ้นสุดการทำงานในบัสไซเคิลนี้เมื่อสิ้นสุดคล็อก T4

2. บัสไซเคิลในการเขียนข้อมูลลงบนหน่วยความจำ

8088 จะทำการสร้างบัสไซเคิลนี้ขึ้น ในช่วงที่ 8088 ทำการเขียนข้อมูลลงในหน่วยความจำ สำหรับขบวนการของสัญญาณที่เกิดขึ้นในบัสไซเคิลนี้สามารถดูได้ดังรูปที่ 2.14



รูปที่ 2.14 บัสไซเคิลของการเขียนข้อมูลลงบนหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปบัสไซเคิลนี้จะเริ่มต้นในช่วงของคล็อก T1 ซึ่งเป็นช่วงเวลาที่ ALE แอคทีฟ "1" สัญญาณ ALE นี้จะถูกใช้เพื่อแสดงให้อุปกรณ์ที่ทำงานร่วมกับ 8088 ทราบว่าข้อมูลที่อยู่บนแอดเดรสในช่วงขอบขาลงของสัญญาณ ALE นั้นเป็นแอดเดรสของหน่วยความจำที่ 8088 ต้องการติดต่อด้วยหลังจากนั้นช่วงคล็อก T2 สัญญาณ $\overline{\text{MEMR}}$ จะแอคทีฟ "0" ซึ่งเป็นการแสดงให้อุปกรณ์ที่ทำงานร่วมกับ 8088 ทราบว่าบัสไซเคิลนี้เป็นบัสไซเคิลในการเขียนข้อมูลลงบนหน่วยความจำจากนั้น 8088 จะทำการส่งข้อมูลที่ต้องการที่จะเขียนลงในหน่วยความจำที่กำหนดนั้นออกมาบนบัสข้อมูลในช่วงของคล็อก T4 สัญญาณ $\overline{\text{MEMR}}$ จะถูกปรับให้กลับเป็น "0" และจะสิ้นสุดการทำงานในบัสไซเคิลเมื่อสิ้นสุดช่วงเวลาคล็อก T4

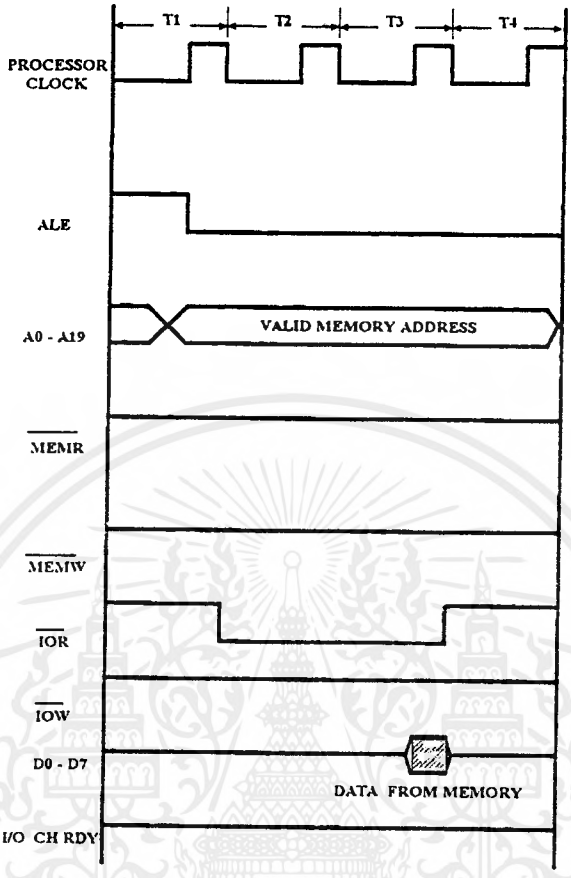
3. บัสไซเคิลในการอ่านข้อมูลจากพอร์ตอินพุต / เอาต์พุต

ในขณะที่ 8088 ปฏิบัติงานชุดคำสั่ง IN เช่น IN ac , DATA ซึ่งเป็นคำสั่งชุดที่ทำให้ 8088 ทำการอ่านข้อมูลจากพอร์ตที่กำหนดในส่วนของโอเปอแรนด์นั้น 8088 จะสร้างบัสไซเคิลในการอ่านข้อมูลจากพอร์ต เพื่อให้พอร์ตที่ถูกกำหนดนั้นส่งข้อมูลออกมาบนบัสข้อมูล สำหรับขบวนการของสัญญาณที่เกิดขึ้นในบัสนี้สามารถแสดงได้ดังรูปที่ 2.15

จากบัสไซเคิลนี้จะเริ่มต้นในช่วงของคล็อก T1 ซึ่งเป็นช่วงเวลาที่สัญญาณ ALE แอคทีฟ "1" สัญญาณ ALE นี้จะถูกใช้เพื่อแสดงให้อุปกรณ์ที่ทำงานร่วมกับ 8088 ทราบว่าข้อมูลที่อยู่บนบัสแอดเดรสในช่วงขอบขาลงของสัญญาณ ALE นั้นเป็นแอดเดรสของพอร์ตที่ 8088 ต้องการติดต่อด้วย

สำหรับใน 8088 สามารถอ้างแอดเดรสของพอร์ตได้เพียง 64 กิโลไบต์ เท่านั้นในขณะที่สามารถอ้างแอดเดรสของหน่วยความจำได้ถึง 1 เมกะไบต์ ดังนั้นในกรณีของบัสไซเคิลที่เกี่ยวกับการอ้างแอดเดรสพอร์ต 8088 จะใช้เส้นแอดเดรสเพียง 16 เส้น คือ A0-A15 เท่านั้น หลังจากนั้นในช่วงของคล็อก T2 สัญญาณ $\overline{\text{IOR}}$ จะแอคทีฟ "0" ซึ่งเป็นการแสดงให้กับอุปกรณ์ที่ทำงานร่วมกับ 8088 ทราบว่าบัสไซเคิลนี้เป็นบัสไซเคิลในการอ่านข้อมูลจากพอร์ตและเป็นการทำให้พอร์ตที่มีตำแหน่งตรงกับค่าตำแหน่งที่อยู่บนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูล เมื่อพอร์ตที่ถูกอ้างถึงบนบัสข้อมูลทำการส่งข้อมูลออกมาบนบัสข้อมูลแล้ว 8088 จะอ่านข้อมูลนั้นเป็นช่วงเริ่มต้นของคล็อก T4 จากนั้นสัญญาณ $\overline{\text{IOR}}$ จะถูกปรับให้เป็น "1" และจะสิ้นสุดการทำงานในบัสไซเคิลเมื่อสิ้นสุดช่วงเวลาของคล็อก T4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 บัสไซเคิลของการอ่านข้อมูลจากพอร์ตอินพุต / เอาต์พุต

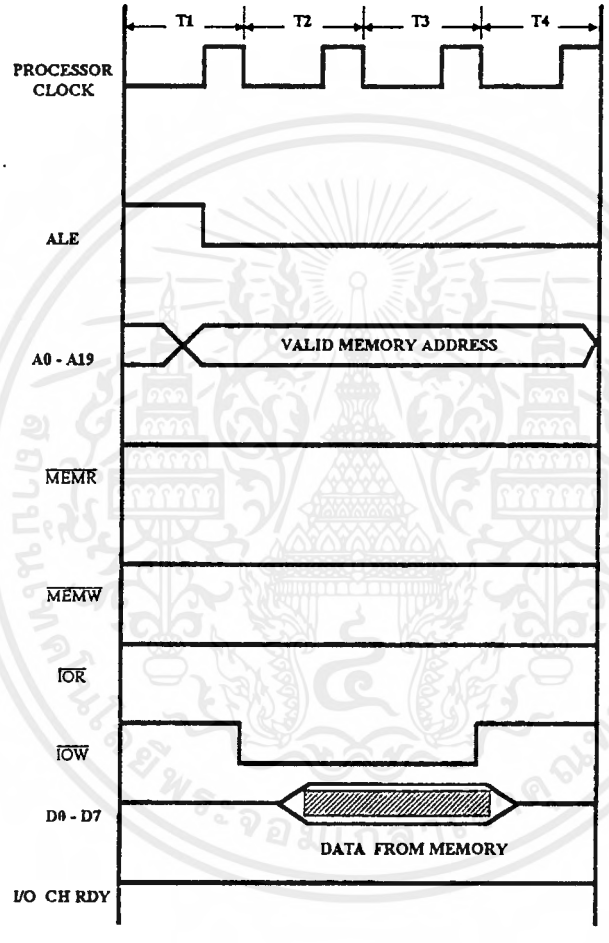
4. บัสไซเคิลในการเขียนข้อมูลลงบนพอร์ตอินพุต / เอาต์พุต

ในขณะที่ 8088 ปฏิบัติงานชุดคำสั่ง OUT เช่น OUT DATA ,ac ซึ่งเป็นชุดคำสั่งที่ทำให้ 8088 ทำการอ่านข้อมูลจากพอร์ตที่กำหนดในส่วนของโอเปอแรนด์นั้น 8088 จะสร้างบัสไซเคิลในการเขียนข้อมูลลงบนพอร์ต เพื่อให้พอร์ตที่ถูกกำหนดนั้นรับข้อมูลบนบัสข้อมูล สำหรับขบวนการของสัญญาณที่เกิดขึ้นในบัสนี้สามารถแสดงได้ดังรูปที่ 2.16

บัสไซเคิลนี้จะเริ่มต้นในช่วงคล็อก T1 ซึ่งเป็นช่วงที่สัญญาณ ALE แอคทีฟ "1" สัญญาณ ALE นี้จะถูกใช้แสดงให้อุปกรณ์ที่ทำงานร่วมกับ 8088 ทราบว่าข้อมูลที่อยู่บนบัสแอดเดรสช่วงขอบขาลงของสัญญาณ ALE นั้นเป็นแอดเดรสของพอร์ตที่ 8088 ต้องการติดต่อด้วย สำหรับเส้นแอดเดรสที่ใช้ในบัสไซเคิลนี้จะมีจำนวน 16 เส้นเท่ากับในกรณีของบัสไซเคิลในการอ่านข้อมูลจากพอร์ตคือ A0-A15 หลังจากนั้นช่วงของคล็อก T2 สัญญาณ $\overline{IOW}$ จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอคทีฟ “0” ซึ่งเป็นการแสดงให้อุปกรณ์ที่ทำงานร่วมกับ 8088 ทราบว่าบัสไซเคิลนี้เป็นบัสไซเคิลในการเขียนข้อมูลลงในพอร์ต จากนั้น 8088 จะทำการส่งข้อมูลที่ต้องการให้กับพอร์ตที่กำหนดนั้นออกมาบนบัสข้อมูลในช่วงคล็อกที่ T4 สัญญาณ $\overline{IOW}$ จะถูกปรับให้กลับเป็น “1” และจะสิ้นสุดการทำงานในบัสไซเคิลเมื่อสิ้นสุดช่วงเวลาของคล็อก T4



รูปที่ 2.16 บัสไซเคิลของการเขียนข้อมูลลงบนพอร์ตอินพุต / เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

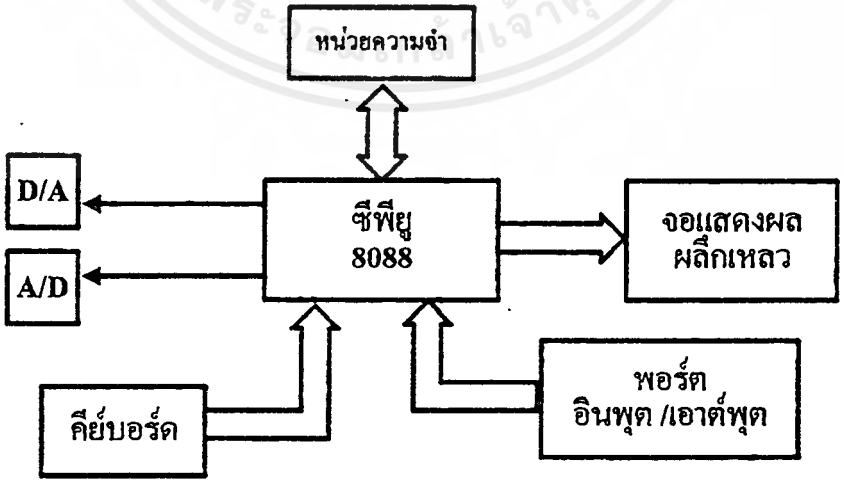
บทที่ 3

การออกแบบและการสร้าง

3.1 กล่าวนำ

ชุดไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 จะประกอบด้วยหลายส่วน แสดงดังรูปที่ 3.1 กันซึ่งในการออกแบบจะต้องเริ่มทำการออกแบบที่ละส่วน โดยส่วนประกอบหลักในชุด ของ แผ่นพิมพ์เดี่ยว นั้นจะประกอบด้วย

1. ชุด กำเนิดสัญญาณนาฬิกา
2. ชุด รีเซ็ต
3. ชุด แลตช์ข้อมูล
4. ชุด หน่วยความจำ
5. ชุด พอร์ตอินพุต / เอาต์พุต
6. ชุด จอแสดงผลผลึกเหลว
7. ชุด คีย์บอร์ดขนาด 6 × 4 และ วงจร D/A Converter
8. ชุด A/D Converter



รูปที่ 3.1 ส่วนประกอบของไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบและการสร้าง

ในการออกแบบและสร้างชุด ไมโครโปรเซสเซอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 สามารถแบ่งได้ออกเป็น 2 ส่วน ดังนี้

1. ด้านฮาร์ดแวร์
2. ด้านซอฟต์แวร์

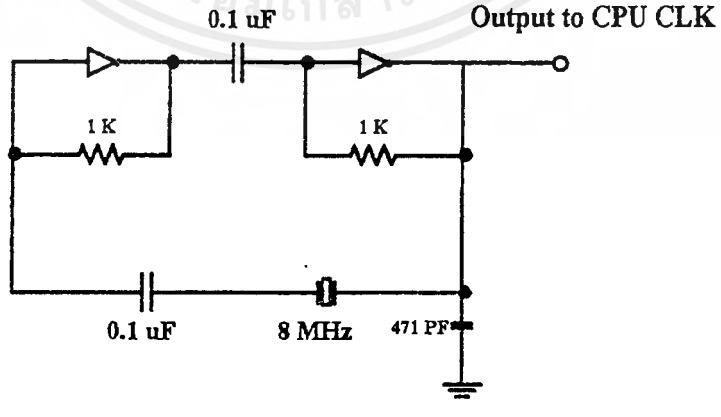
รายละเอียดในการสร้างวงจรของแต่ละส่วนนั้นจะกล่าวในส่วนของเนื้อหาในลำดับต่อไปนี้

3.3 การออกแบบด้านฮาร์ดแวร์

3.3.1 ชุด กำเนิดสัญญาณนาฬิกา

- ในการทำงานของซีพียูนั้นจะต้องอาศัยสัญญาณนาฬิกาซึ่งมีความถี่ที่แน่นอน โดยซีพียู 8088 นี้จะใช้สัญญาณนาฬิกาขนาด 8 MHz

- ซึ่งวงจรที่ใช้นั้นเป็นวงจรกำเนิดสัญญาณนาฬิกาแบบ NOT Gate โดยสัญญาณเอาต์พุตที่เกิดขึ้นที่ คริสตอล จะถูกส่งไป NOT Gate แรกสัญญาณจะถูกอินเวิร์ดและจะถูกอินเวิร์ดอีกครั้งให้กลับเป็นเหมือนเดิม โดย NOT Gate ตัวที่ 2 เอาต์พุตที่ได้ส่งไปยังขา 19 ของซีพียูดังรูปที่ 3.2

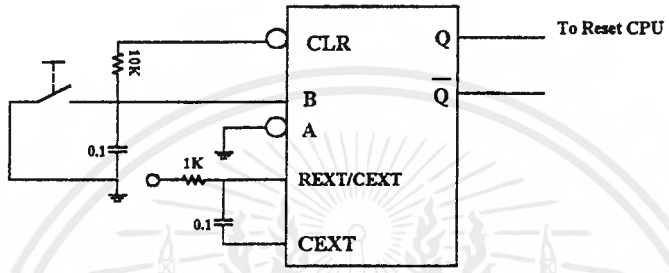


รูปที่ 3. 2 วงจรกำเนิดสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ชุด รีเซ็ต

- เนื่องจากซีพียูนั้นจะต้องใช้สัญญาณรีเซ็ตในการเคลียร์ ข้อมูลเพื่อตั้งต้นการทำงานใหม่ และให้ซีพียูพร้อมที่จะทำงานสัญญาณรีเซ็ตที่เกิดขึ้นจากการชาร์จ และดิสชาร์จของตัวเก็บประจุ 0.1 ไมโครฟารัสโดยแรงดันจะเป็นบวก (+) ต่อผ่าน D Flip - Flop เข้ามาที่ ขา 21 ของซีพียูดังรูปที่ 3.2



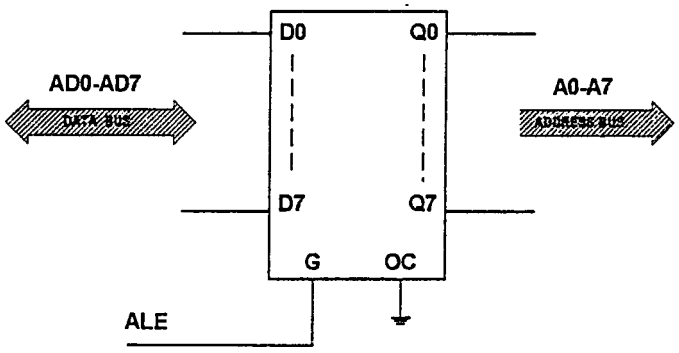
รูปที่ 3.3 วงจร รีเซ็ต

3.3.3 ชุดแลตช์ข้อมูล

เนื่องจากตัวซีพียูมีโครงสร้างภายในประกอบไปด้วย แอดเดรสบัสกับบัสข้อมูล ซึ่งทำการต่อแบบมัลติเพลกซ์กันอยู่ เพราะฉะนั้นเราจึงต้องมีการสร้าง วงจร ดีมัลติเพลกซ์ ของข้อมูลออกจาก แอดเดรสบัส โดยพียู จะมีขาสัญญาณ ALE เป็นตัวบอกว่า ช่วงเวลาใดที่ มีการส่งแอดเดรสออกมา ซึ่งที่ขา ALE จะแสดงสถานะลอจิก “1” ออกมา

จากคุณสมบัตินี้จึงใช้ IC 74LS373 (Latch) มาทำการดีมัลติเพลกซ์สัญญาณแอดเดรสออกจาก สัญญาณของข้อมูล แล้วใช้สัญญาณ ALE เป็นสัญญาณควบคุม โดยคุณสมบัติของ 74LS373 จะมีขา G ที่ใช้ควบคุมการทำงานถ้าได้รับลอจิก “1” จาก ALE ของซีพียู โดยจะทำการส่งค่าแอดเดรสออกทางเอาต์พุตและถ้าที่ขา G ได้รับลอจิก “0” สภาวะเอาต์พุตจะคงสภาวะค่าแอดเดรสไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรแลตช์ข้อมูล

3.3.4 ชุด หน่วยความจำ

ในการทำงานของซิงเกิ้ลบอร์ดนั้นจะต้องมีการติดต่อกับหน่วยความจำเพื่อนำข้อมูลมาประมวลผลซึ่งหน่วยความจำนี้จำเป็นอย่างมาก โดยซีพียู 8088 สามารถอ้างหรือติดต่อกับหน่วยความจำที่มีขนาด 1 เมกะไบต์

แต่ในการออกแบบนั้นเราจะต้องทำการ แบ่งหน่วยความจำออกเป็นส่วนๆ ซึ่งจะอยู่ใน เซกเมนต์เดียวกันโดยใช้หลักการของ Decoder โดยกำหนดให้

0000H-1FFFH เป็น RAM ขนาด 8 กิโลไบต์

2000H-3FFFH เป็น RAM ขนาด 8 กิโลไบต์

4000H-5FFFH เป็น RAM ขนาด 8 กิโลไบต์

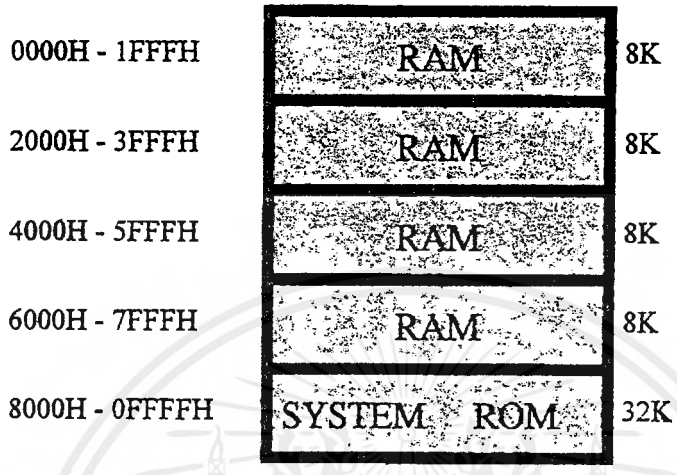
6000H-7FFFH เป็น RAM ขนาด 8 กิโลไบต์

8000H-FFFFH เป็น SYSTEM ROM

ในการDecode ได้นำไอซี 74LS156 เป็นตัวดีโค้ดเลือกตำแหน่งของแอดเดรสโดยใช้ไอซี 74LS156 เนื่องจากทำการออกแบบให้อยู่ในเซกเมนต์เดียวกันจึงทำให้ไม่สนใจในส่วนของแอดเดรสบัส A16-A19 แต่ใช้ A13-A15 มาทำการดีโค้ด โดยให้ A15 ต่อเข้ากับขา 1C, $\overline{2C}$ โดย A15 จะแยกระหว่างหน่วยความจำระบบ (SYSTEM ROM)กับหน่วยความจำ (RAM) อื่นๆ ส่วนของหน่วยความจำระบบจะมีขนาด 32 กิโลไบต์ โดยนำสัญญาณเอาต์พุตจาก 1Y0-1Y3 มาเป็นสัญญาณควบคุมและที่เอาต์พุต 2Y0-2Y3 จะเป็นส่วนของหน่วยความจำของผู้ใช้ซึ่งจะถูกแบ่งให้กลุ่มละ 8 กิโลไบต์ โดยสัญญาณ A13 และ A140 จะนำมาควบคุมการเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของหน่วยความจำและขา $\overline{1G}$, $\overline{2G}$ จะถูกขา IO / $\overline{M}$ ซึ่งจะมีส่วนทำให้หน่วยความจำถูกใช้งานเมื่อซีพียูส่งสัญญาณมา



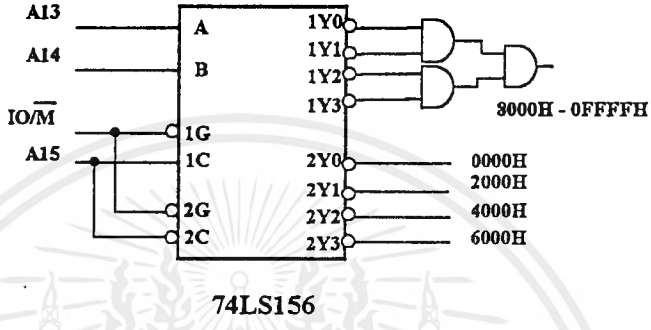
รูปที่ 3.5 การจัดหน่วยความจำ

3.3.5 ชุดพอร์ตอินพุต / เอาต์พุต

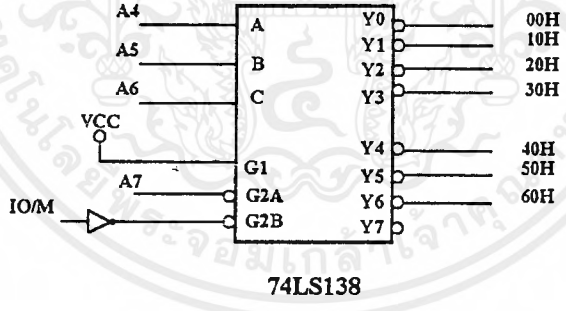
โดยในการออกแบบให้พอร์ตอินพุตและเอาต์พุตเราจะแยกออกจากกันโดยพอร์ตอินพุต / เอาต์พุต ได้ทำการดีโค้ด อยู่ที่ตำแหน่ง 10 H และ 20 H มีจำนวน 2พอร์ต ซึ่งพอร์ตเอาต์พุตต่อผ่านไอซี 74LS377 ใช้สัญญาณ $\overline{WR}$ เป็นตัวสั่งงาน และพอร์ตอินพุตที่ต้องต่อผ่านไอซี 74LS541 ใช้สัญญาณ $\overline{RD}$ สั่งงาน ในการดีโค้ดพอร์ต ให้ทำงานตามต้องการโดยใช้ 74LS138 โดยใช้สัญญาณ A4-A7 มาทำการดีโค้ดโดยนำขา G_1 ต่อกับ Vcc และ $\overline{G_{2A}}$ ต่อกับ A7 เพื่อต้องการที่จะทำงานเมื่อ A7 เป็นลอจิก "0" ซึ่งขา $\overline{G_{2B}}$ ต่อกับสัญญาณอินเวิร์ตของสัญญาณ IO / $\overline{M}$ จะเป็นลอจิก "0" เมื่อมีการติดต่อกับอุปกรณ์อินพุตเอาต์พุต สามารถเลือกตำแหน่งพอร์ตได้ 8 พอร์ตโดยใช้สัญญาณ A4-A6 ต่อที่ขา A B C พอร์ตใช้งานจะถูกแยกส่วนของ อินพุตและเอาต์พุตออกจากกันโดยพอร์ตนี้จะกำหนดให้อยู่ในตำแหน่ง 10H และ 20H มีจำนวน 2พอร์ต โดยพอร์ตเอาต์พุตจะต่อผ่านไอซีเบอร์ 74LS377 และใช้สัญญาณ $\overline{WR}$ เป็นสัญญาณควบคุมที่ขา $\overline{E}$ ของไอซีเบอร์ 74LS377 ต่ออยู่กับสัญญาณชีเล็กซ์พอร์ตเมื่อมีการติดต่อพอร์ตจะมีลอจิก "0" ส่งมาทำให้ไอซีเบอร์ 74LS377 พร้อมที่จะทำงาน ส่วนของพอร์ตอินพุต จะถูกต่อผ่านไอซีเบอร์ 74LS541 ใช้สัญญาณ $\overline{RD}$ เป็นสัญญาณควบคุมโดยที่ขา $\overline{G_1}$

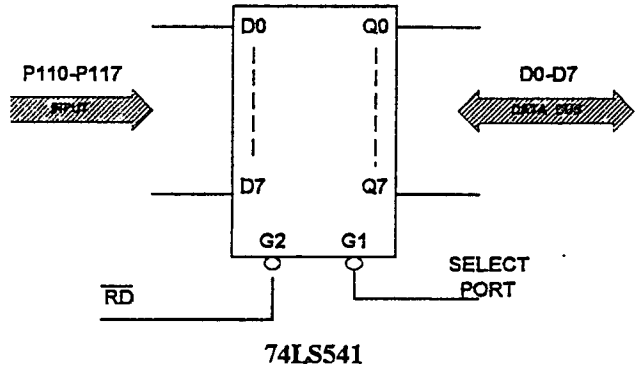
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ 74LS541 ต่ออยู่กับสัญญาณชีเล็กช์พอร์ตเมื่อมีการติดต่อกับพอร์ตจะมีการส่งลอจิก "0" มาทำให้ 74LS541 พร้อมที่จะทำงาน และที่ $\overline{G_2}$ ต่ออยู่กับสัญญาณ $\overline{RD}$ ของซีพียู เมื่อซีพียูต้องการอ่านข้อมูลเข้าก็จะส่งลอจิก "0" ที่ขา$\overline{RD}$ ทำให้ 74LS541 ส่งข้อมูลผ่านพอร์ตลงสู่บัสข้อมูล



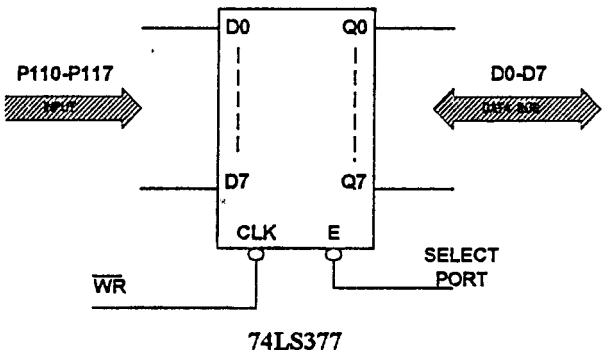
รูปที่ 3.6 วงจร ดีโค้ดหน่วยความจำ





รูปที่ 3.7 วงจรพอร์ตอินพุต / เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 (ต่อ) วงจรพอร์ตอินพุต / เอาต์พุต

3.3.6 ชุด จอแสดงผลผลึกเหลว

ในการทำงานของ จอแสดงผลผลึกเหลวจะใช้ตัว คอนโทรลเลอร์เบอร์ HD44780 ซึ่งเป็นไอซี LSI ตัวหนึ่งซึ่งใช้ควบคุมจอแสดงผลผลึกเหลว โดยจะแสดงผลในรูปตัวอักษรหรือสัญลักษณ์ต่างๆ ตัวมันเองสามารถต่อใช้งานแบบ 4 บิต หรือ 8 บิต ก็ได้ ซึ่งถ้าเราต่อแบบ 4 บิต จะต่อใช้งานที่ DB7-DB4 เท่านั้นโดยข้อมูลครั้งแรกที่ส่งนั้น HD44780 จะถือเป็นข้อมูล 4 บิต บนและข้อมูลที่ส่งต่อมานั้นเป็นข้อมูล 4 บิตล่าง

จอแสดงผลผลึกเหลว ขนาด 16 ตัว 1 บรรทัดรุ่น ตัวอักษร และออกแบบให้ต่อกับบัสโดยตรงไม่ต้องต่อผ่านพอร์ต ซึ่งใช้สัญญาณ A0 , A1 เป็นตัวเลือกแล้วต่อกับบัสข้อมูล ให้พอร์ตของจอแสดงผลผลึกเหลวอยู่ที่ตำแหน่ง 60 H โดยขาอีนาเบิ้ลมีการใช้ร่วมกับสัญญาณ DEN แล้วผ่าน NOT Gate เพื่อให้เป็นสัญญาณอีนาเบิ้ล

3.3.7 ชุด คีย์บอร์ดขนาด 6 × 4 และ วงจร D/A Converter

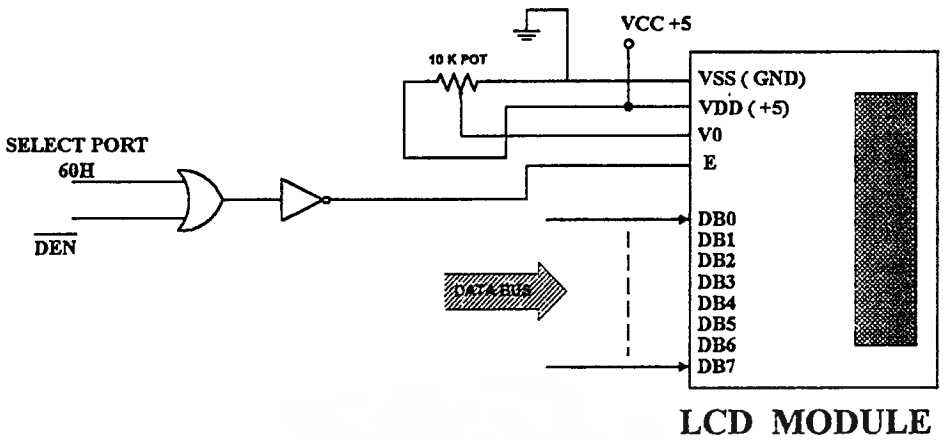
โดยการใช้พอร์ต 8255 ซึ่งเป็นพอร์ตอินพุต / เอาต์พุตขนาด 8 บิต มีจำนวน 3พอร์ต และใช้วงจร ดีโค้ด โดย 74LS138 ให้อยู่ตำแหน่ง 50 H ในการแบ่งพอร์ตการทำงาน มีลักษณะดังนี้

พอร์ต A ของ 8255 ไม่ได้ใช้

พอร์ต B ของ 8255 ต่อกับวงจร D/A Converter

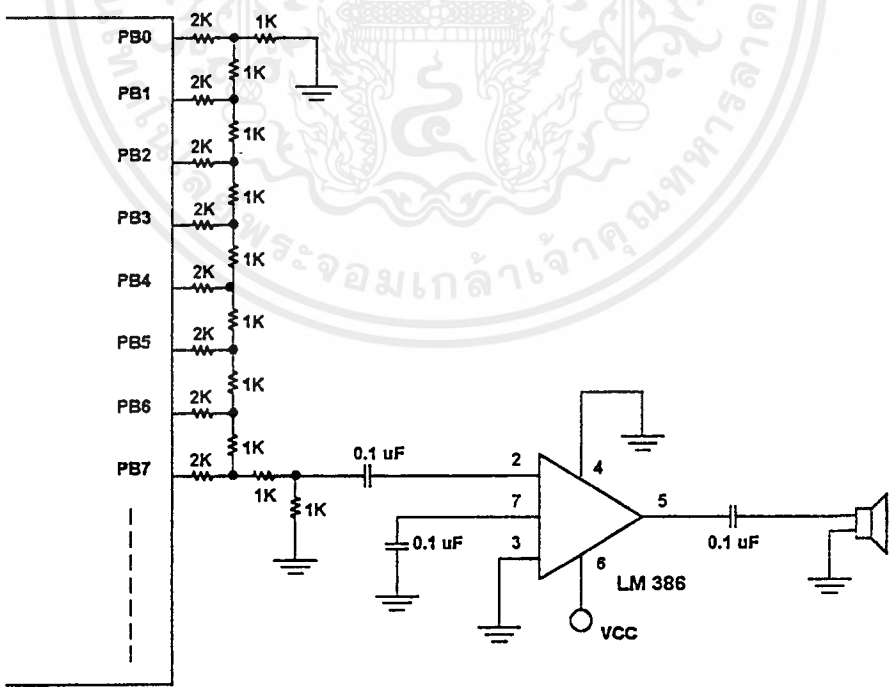
พอร์ต C ของ 8255 ต่อกับวงจร คีย์บอร์ดขนาด 6 × 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.8 วงจร จอแสดงผลผลึกเหลว

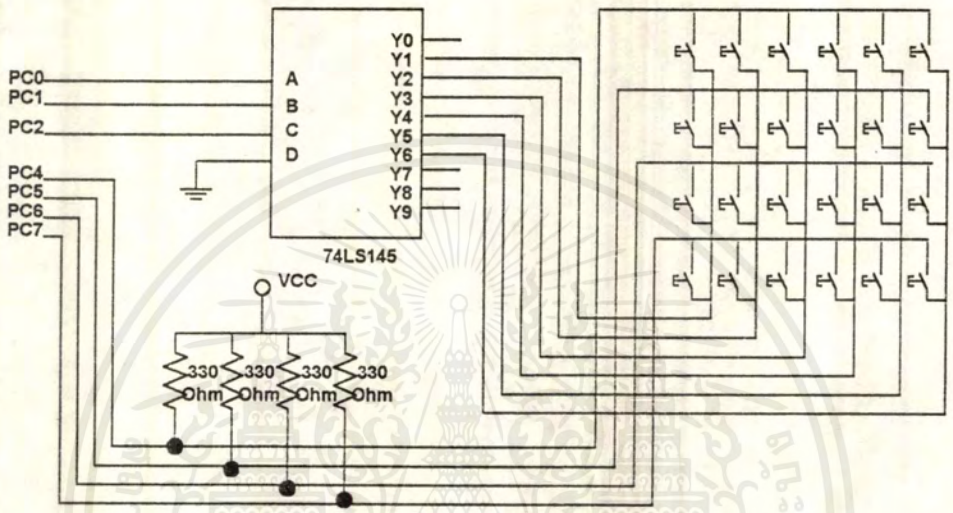
วงจร D/A Converter ใช้ลักษณะของการนำเอาค่าความต้านทานมาต่อผสมกัน และเอาต์พุตที่ได้ไปทำการต่อกับวงจรขยาย ซึ่งลักษณะนี้เรียกว่า R Ladder และกำหนดให้พอร์ต B เป็น เอาต์พุต ดังรูป3.9



รูปที่ 3.9 วงจร D/A CONVERTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

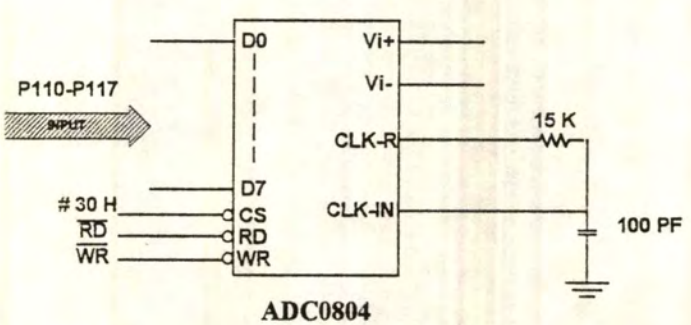
วงจรคีย์บอร์ด ต่อที่พอร์ต C ของ 8255 โดยให้พอร์ต C ล่างเป็นเอาต์พุตและต่อผ่านไอซี 74LS145 ซึ่งในพอร์ต C บนนั้นจะเป็นอินพุตคอยรับค่าที่ได้จากการ สแกนคีย์ จากการเขียนโปรแกรมสั่งงาน ดังรูป



รูปที่ 3.10 วงจรคีย์บอร์ดขนาด 6 × 4

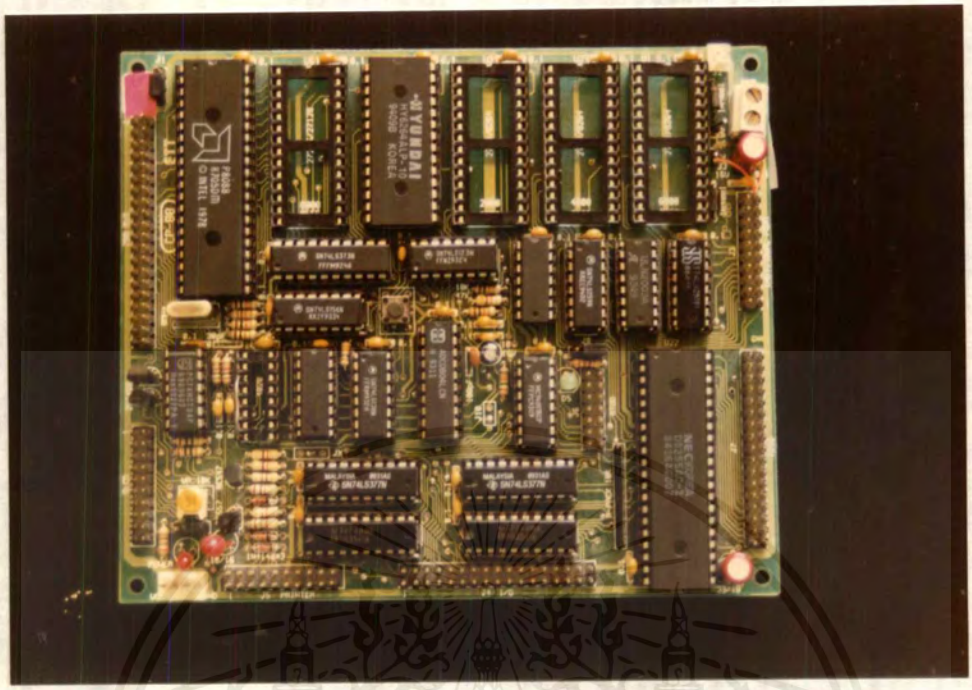
3.3.8 ชุด A/D Converter

-ใช้ ADC0804 เป็นตัวแปลงสัญญาณอะนาลอกเป็นดิจิตอล โดยให้อยู่ที่ตำแหน่งพอร์ต 30 H

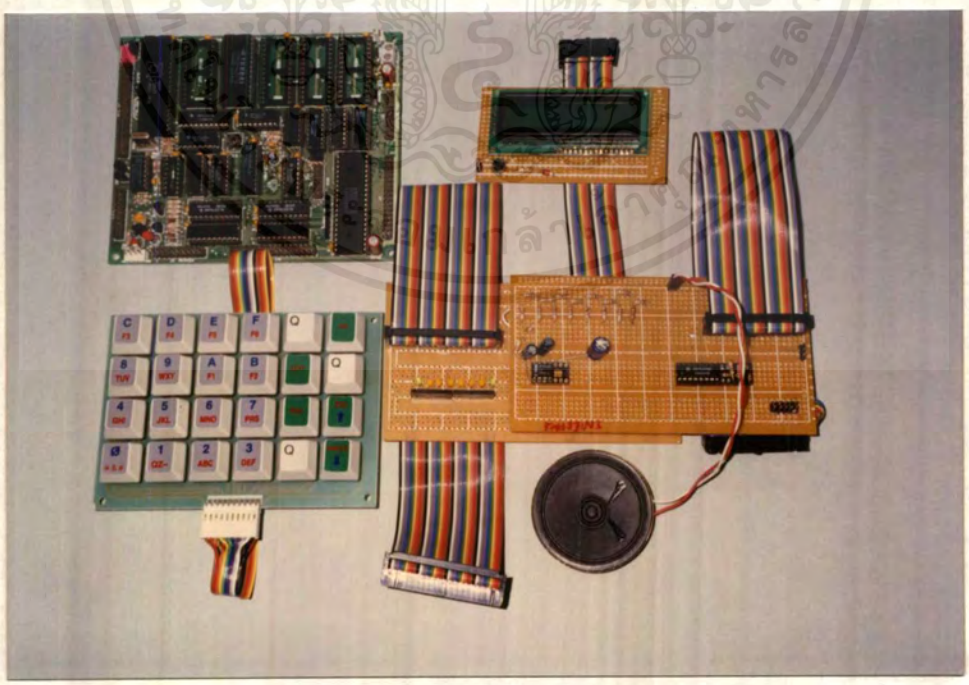


รูปที่ 3.11 วงจร A/D CONVERTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

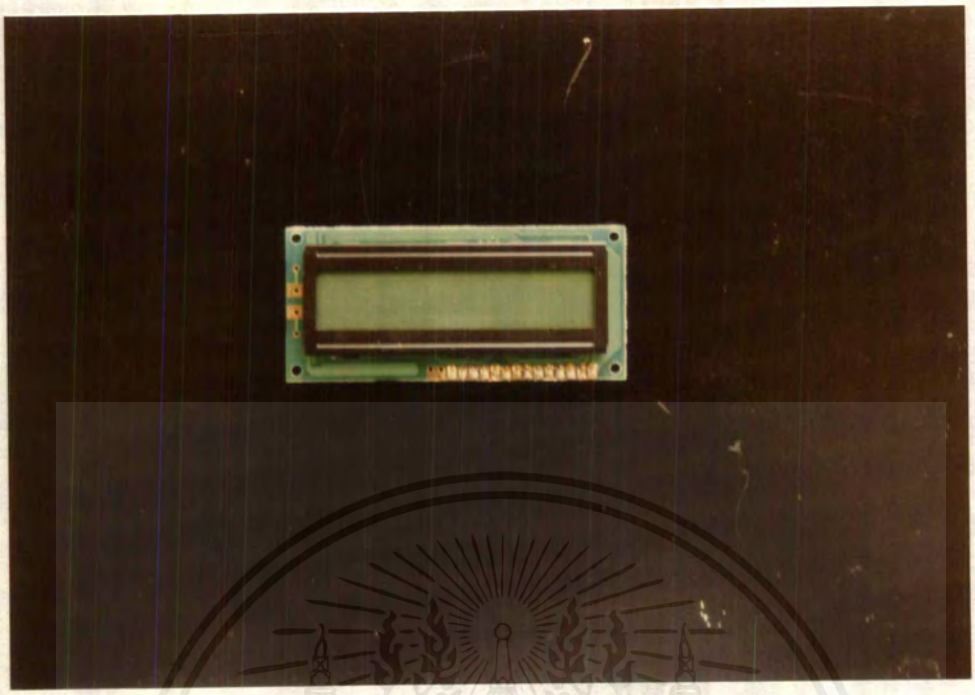


รูปที่ 3.12 ชุดควบคุมของ 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยว

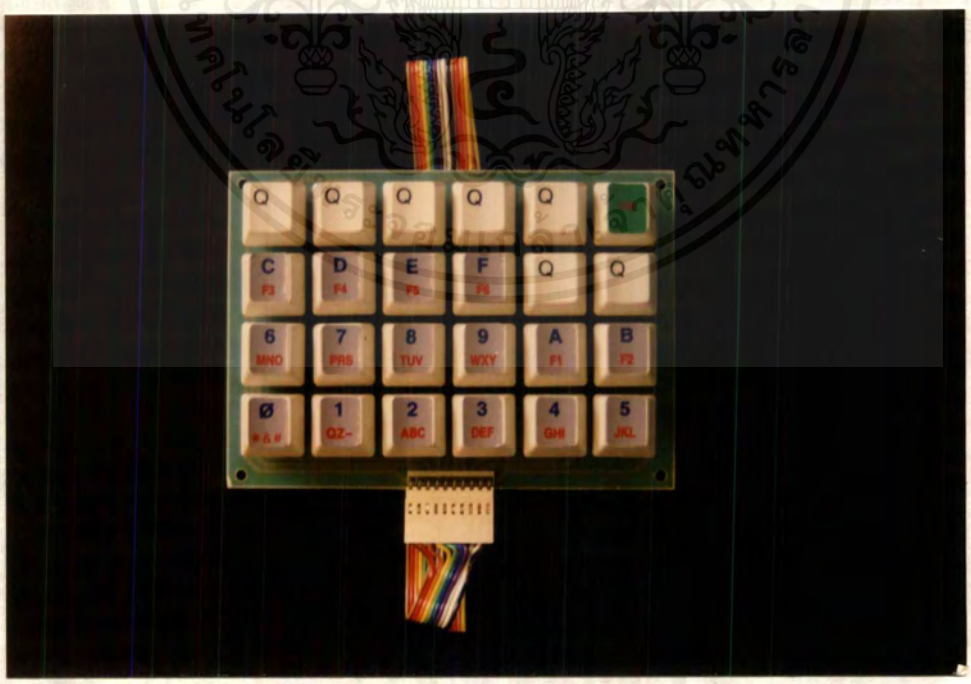


รูปที่ 3.13 ชุดอุปกรณ์ต่อพ่วงในการทดลองชุดควบคุมของ 8088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

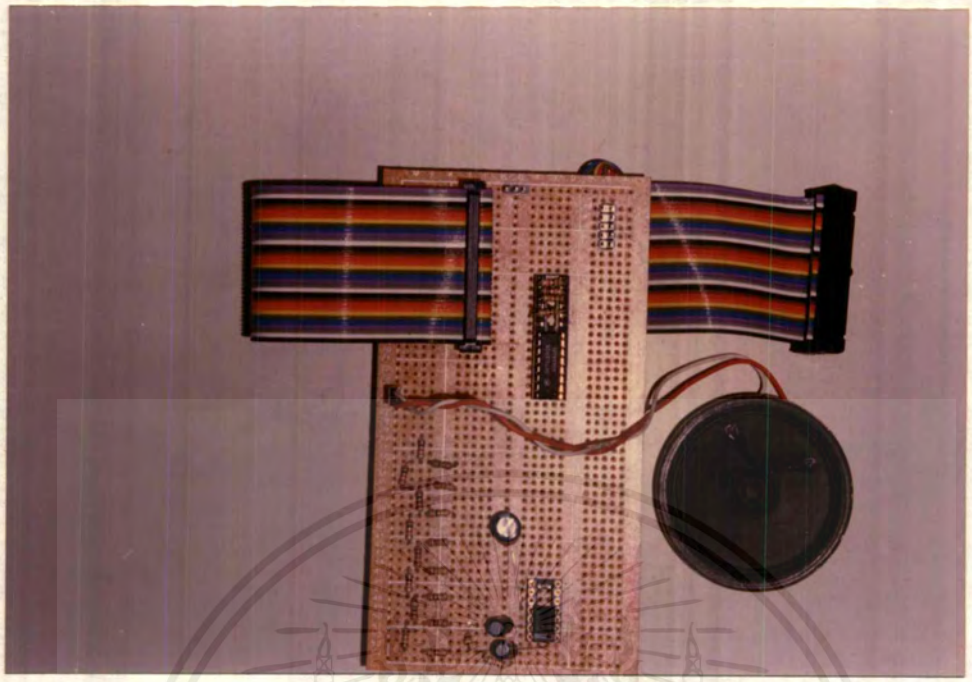


รูปที่ 3.14 ส่วนจอแสดงผลผลึกเหลว

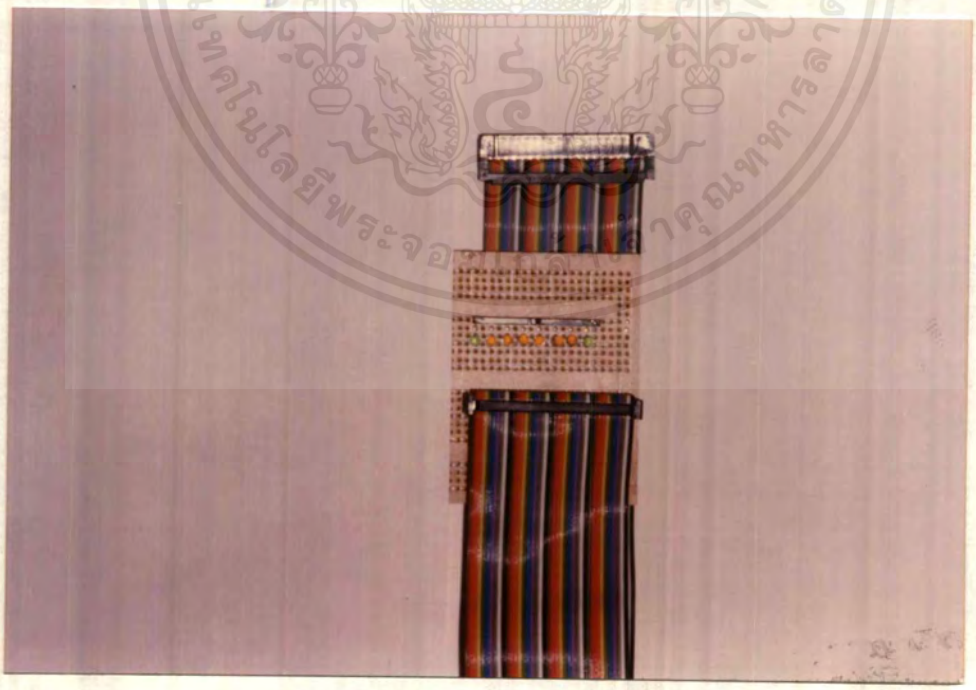


รูปที่ 3.15 ชุดคีย์บอร์ดขนาด 6 × 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ชุดวงจร D/A CONVERTER

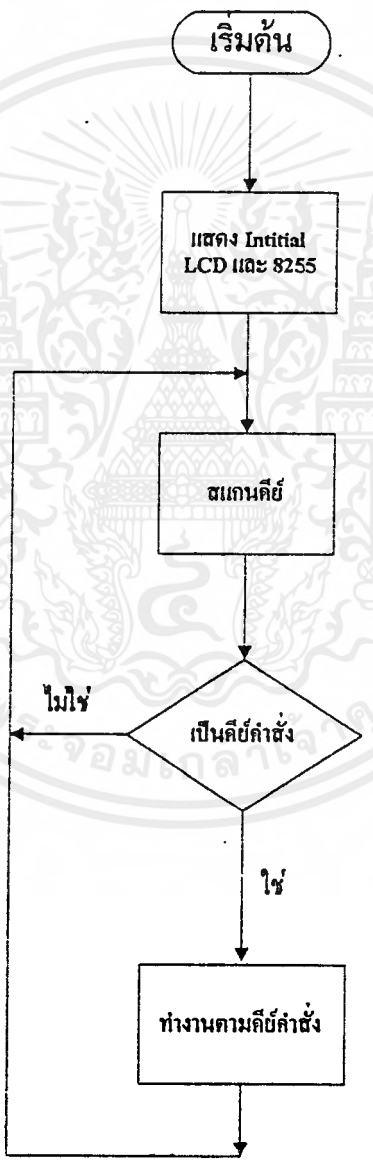


รูปที่ 3.17 ชุดวงจรพอร์ต อินพุต / เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบด้านซอฟต์แวร์

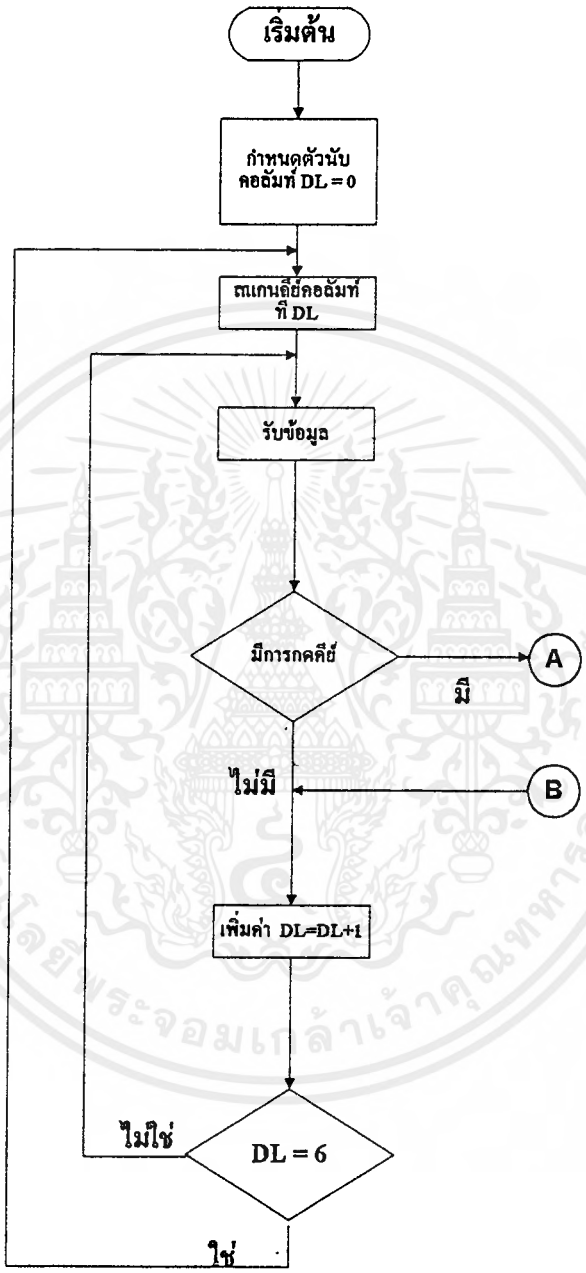
ในด้านซอฟต์แวร์นั้นจะเป็นตัวควบคุมการทำงานของอุปกรณ์ทางด้านฮาร์ดแวร์ซึ่งจะต้องออกแบบให้สอดคล้องกับการทำงานของอุปกรณ์นั้นๆ โดยหลักของการออกแบบนั้นจะต้องมีการเขียนผังการทำงานจึงแบ่งการทำงานออกเป็นส่วนๆเพื่อกำหนดการทำงานให้มีลำดับและขั้นตอนดังนี้



รูปที่ 3.18 ผังการทำงานโปรแกรมหลัก

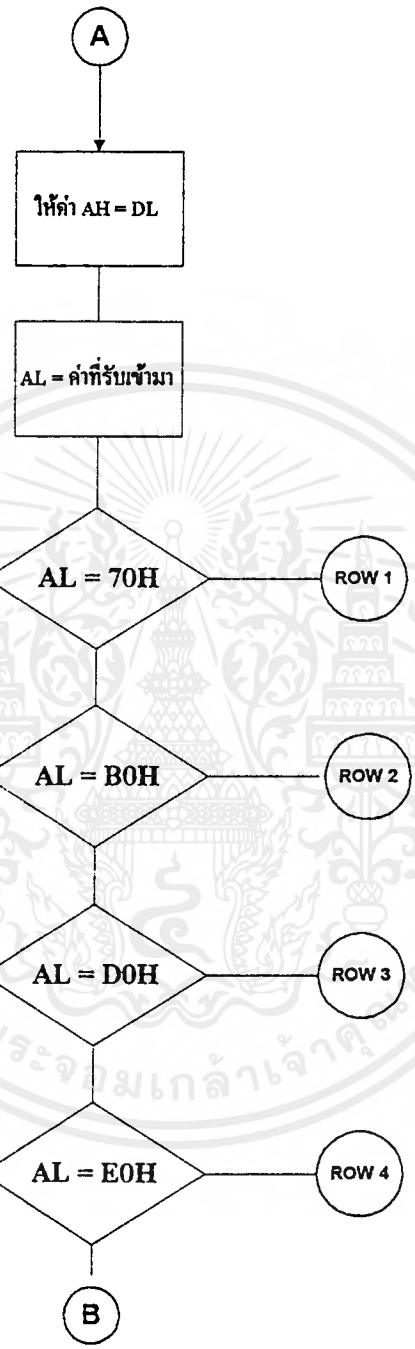
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 ผังการทำงานโปรแกรมสแกนคีย์



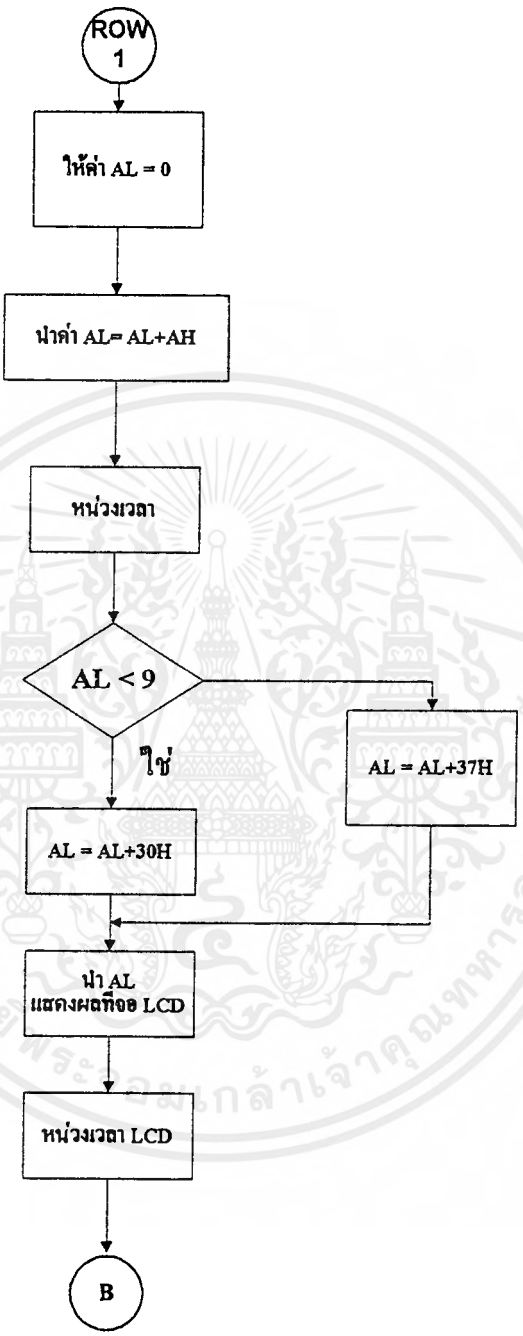
รูปที่ 3.19 ผังการทำงานโปรแกรมสแกนคีย์ (PART I)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



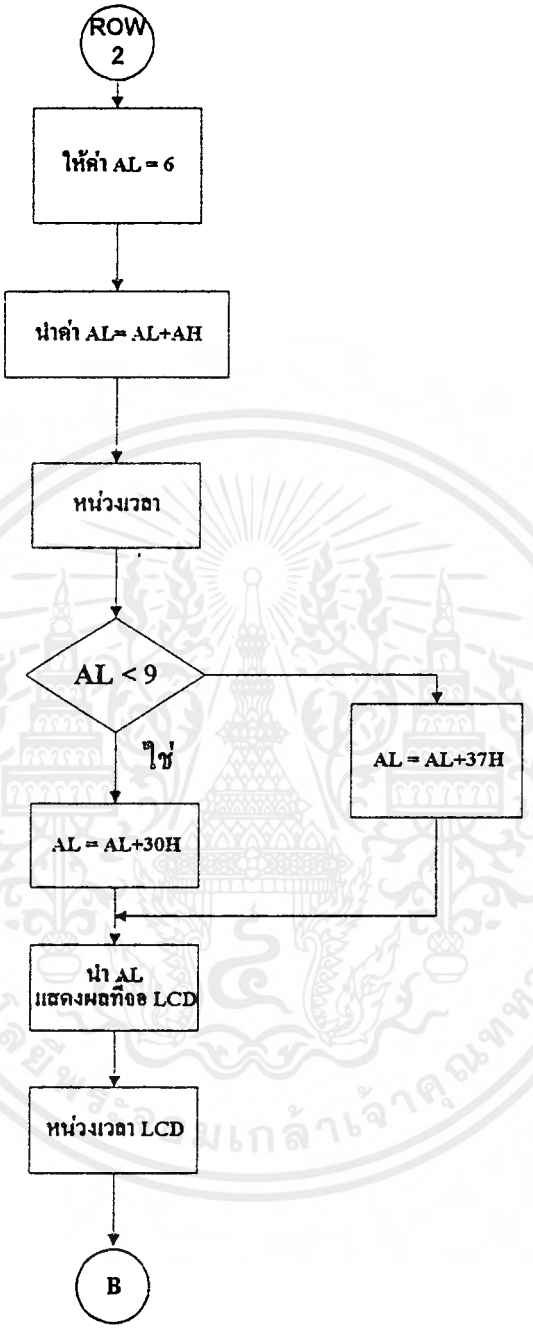
รูปที่ 3.20 ผังการทำงานโปรแกรมสแกนคีย์ (PART II)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



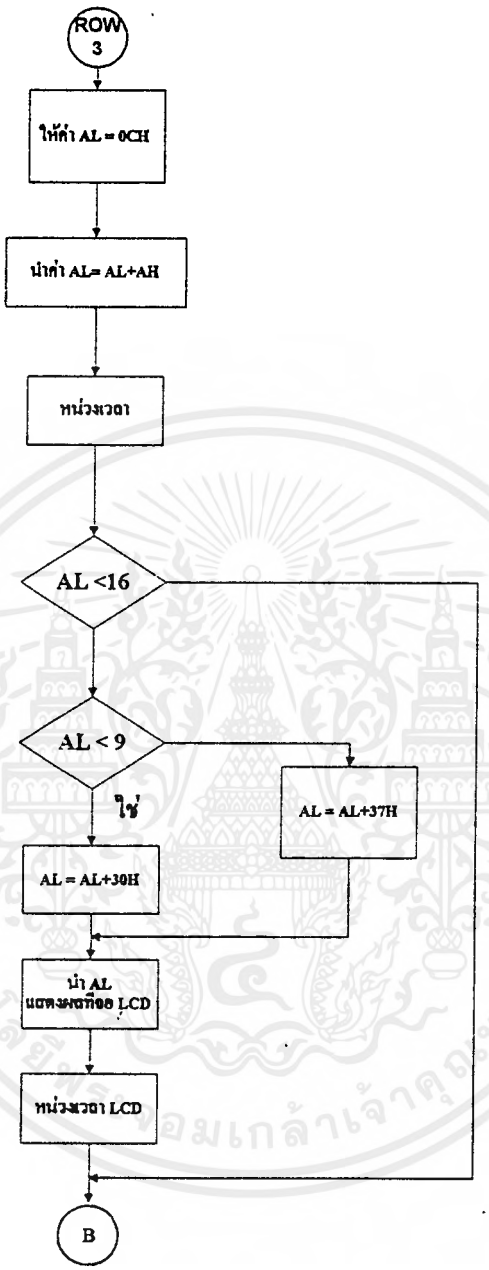
รูปที่ 3.21 ผังการทำงานโปรแกรมสแกนคีย์ (PART III)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



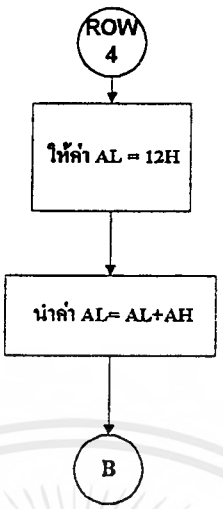
รูปที่ 3.22 ผังการทำงานโปรแกรมสแกนคีย์ (PART IV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 ผังการทำงานโปรแกรมสแกนคีย์ (PART V)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 ผังการทำงานโปรแกรมสแกนคีย์ (PART VI)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและทดสอบ

4.1 กล่าวนำ

การทำงานของไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 สามารถทดสอบได้โดยการใช้โปรแกรมทำการทดสอบโดยโปรแกรมที่จะนำมาทดสอบจะถูกเขียนขึ้นด้วยภาษาแอสแซมบลี้ในการทดสอบจะทำการทดสอบชุดวงจรที่ต่อพ่วงกับ 8088 อาทิเช่น ชุดคีย์บอร์ด ชุดจอแสดงผลผลึกเหลวและชุดทดสอบการทำงานของพอร์ตอินพุต / เอาต์พุต

4.2 การทดลองชุดอุปกรณ์คีย์บอร์ด

4.2.1 ลำดับขั้นการทดลอง

1. ให้พลังงานไฟฟ้าจากแหล่งจ่ายกับชุดควบคุม 8088 ไมโครคอมพิวเตอร์
2. นำจุดต่อจากวงจรคีย์บอร์ดต่อกับพอร์ตของ 8255 ที่สร้างขึ้น (J7)
3. เสียบสายจุดต่อจาก EPROM EMULATORลงบนตำแหน่ง (U5) ซึ่งเป็นส่วนของ ROM ของชุด 8088 ไมโครคอมพิวเตอร์
4. เขียนโปรแกรมที่ 4.1 ใช้ทำการทดสอบชุดของคีย์บอร์ดด้วยภาษาแอสแซมบลี้ ซึ่งทำการแปลงคำแล้วส่งออกทางพอร์ตเครื่องพิมพ์ ของเครื่อง คอมพิวเตอร์
5. ทำการกดสวิตช์รีเซ็ต ของชุด 8088 แล้วสังเกตผลที่ได้

โปรแกรมการทดลองที่ 4.1

```
.MODEL  SMALL
.CODE
        P_A     EQU   50H
        P_B     EQU   51H
        P_C     EQU   52H
        P_CON   EQU   53H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WR_COMM   EQU   60H
RD_BF     EQU   61H
WR_DATA   EQU   62H
RD_DATA   EQU   63H
P_10      EQU   10H
START     PROC  NEAR
          MOV   AX,CS
          MOV   DS,AX
          MOV   ES,AX
          MOV   AX,1FFFH
          MOV   SP,AX
INITIAL:
          MOV   AL,88H
          OUT   P_CON,AL
          MOV   AL,00H
          OUT   10H,AL
          MOV   AL,38H
          OUT   WR_COMM,AL
          MOV   AL,0FH
          OUT   WR_COMM,AL
          MOV   AL,6H
          OUT   WR_COMM,AL
          MOV   AL,1H
          OUT   WR_COMM,AL
XXX:      MOV   DL,0H
XXX1:     CALL  SCAN
          MOV   AL,DH
          CMP   DH,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JZ          SKIPP
            OR          AL,DL
            CALL        TTT
SKIPP:
            INC         DL
            CMP         DL,6
            JNZ         XXX1
            JMP         XXX
START  ENDP
TTT   PROC  NEAR
            MOV         AH,AL
            AND         AH,0FH   ;/
            AND         AL,0F0H  ;/
            CMP         AL,70H
            JZ          S_ROW0
            CMP         AL,0B0H
            JZ          S_ROW1
            CMP         AL,0D0H
            JZ          S_ROW2
            CMP         AL,0E0H
            JZ          S_ROW3
S_RQW0:     MOV         AL,0
            ADD         AL,AH
            OUT         10H,AL
            CALL        DELAY
            JMP         SO
S_ROW1:     MOV         AL,6
            ADD         AL,AH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                OUT         10H,AL
                CALL        DELAY
                JMP         SO
S_ROW2:         MOV         AL,12
                ADD         AL,AH
                OUT         10H,AL
                CALL        DELAY
                JMP         SO
S_ROW3:         MOV         AL,18
                ADD         AL,AH
                OUT         10H,AL
                CALL        DELAY
SO:             RET
                TTT         ENDP
        SCAN    PROC    NEAR
                PUSH        AX
                MOV         AL,DL    ;COLUM
                OUT         P_C,AL
                IN          AL,P_C
                AND         AL,0F0H
                CMP         AL,0F0H
                JNZ         ONKEY
                MOV         AL,0
ONKEY:
                MOV         DH,AL
ENDSCAN:
                POP         AX
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SCAN        ENDP
DELAY   PROC   NEAR
        PUSH        CX
        MOV         CX,1FFH
        LOOP        $
        POP         CX
        RET
DELAY   ENDP
        ORG         7FF0H
        JMP         START
        END         START
```

การทำงานของโปรแกรมที่ 4.1 จะทำการสแกนคีย์ในแนวของคอลัมท์ที่ 0 ถึงคอลัมท์ที่ 5 เมื่อมีการกดแป้นพิมพ์ ณ ตำแหน่งใดๆ จะทำให้ได้ค่าค่าหนึ่งเก็บไว้ที่ รีจิสเตอร์ AL และนำออกแสดงผลที่พอร์ต 10 H โดยต่อชุดจอแสดงผลผลึกเหลว ทำให้ทราบถึงตำแหน่งของแป้นพิมพ์ที่ทำการกด โดยที่ค่าของรีจิสเตอร์ AL ซึ่งแสดงไว้ในตารางที่ 4.1

ตารางที่ 4.1 ค่าตำแหน่งของแป้นพิมพ์

ค่า Reg.AL	ตำแหน่งแป้นพิมพ์	ค่า Reg.AL	ตำแหน่งแป้นพิมพ์
70	0	B0	4
71	1	B1	5
72	2	B2	6
73	3	B3	7
74	-	B4	-
75	-	B5	-

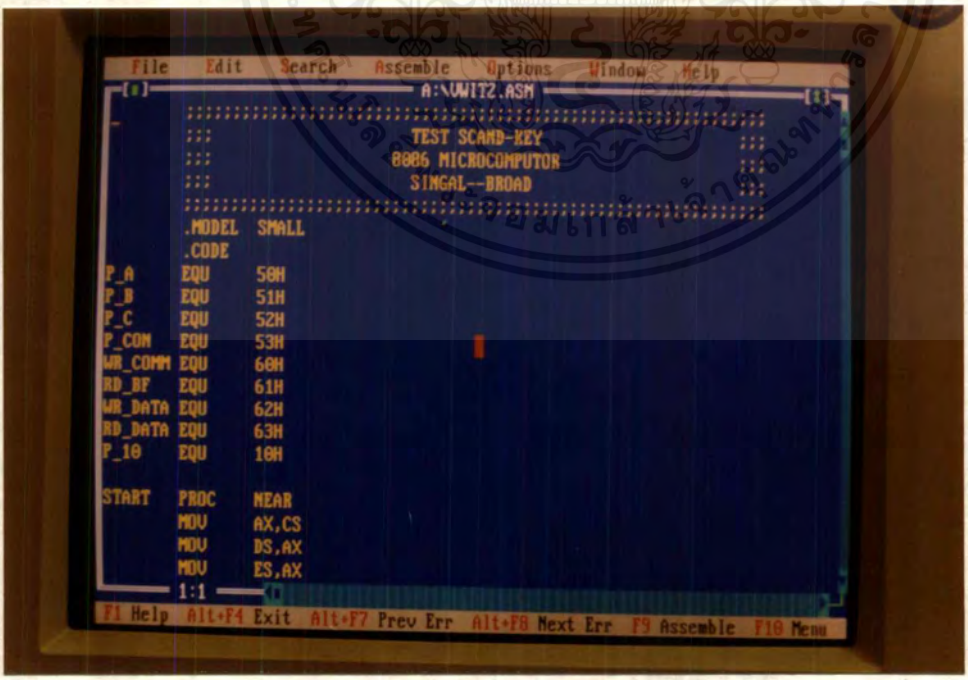
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1(ต่อ) ค่าตำแหน่งของแป้นพิมพ์

ค่า Reg.AL	ตำแหน่งแป้นพิมพ์	ค่า Reg.AL	ตำแหน่งแป้นพิมพ์
D0	8	E0	C
D1	9	E1	D
D2	A	E2	E
D3	B	E3	F
D4	-	E4	-
D5	-	E5	-

2 ผลการทดลอง

เมื่อทำการกดสวิตช์ใดๆ ที่แป้นพิมพ์จะทำให้สามารถทราบตำแหน่งของแป้นพิมพ์ได้ ตำแหน่งของแป้นพิมพ์จะแสดงผลออกทางพอร์ต 10 H



รูปที่ 4.1 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.3 การทดลองการทำงานของอินพุต / เอาต์พุต

.3.1 ลำดับขั้นการทดลอง

1. นำจุดต่อของชุดวงจร แอลอีดี ต่อเข้ากับชุด 8088 ตรงตำแหน่ง (J4)

2. ให้พลังงานไฟฟ้าจากแหล่งจ่ายกับชุดควบคุม 8088 ไมโครคอมพิวเตอร์

3. ทำการเขียนโปรแกรมที่ 4.2 แล้วทำการแปลงคำสั่งโปรแกรมที่ 4.2 แล้วทำการส่งอมูลที่เขียนออกจากเครื่องคอมพิวเตอร์มาสู่ EPROM EMULATOR ลงบนตำแหน่ง U5

4. นำจุดต่อสายจาก EPROM EMULATOR ต่อเข้ากับชุดไมโครคอมพิวเตอร์แบบแผ่นมพ์เดียวเบอร์ 8088

5. ทำการกดสวิตช์รีเซ็ต ของชุด 8088

ปรแกรมการทดลองที่ 4.2

```
MODEL  SMALL
CODE
TART:   MOV     AX,1FFFH
        MOV     SP,AX
2:      MOV     BX,8
        MOV     AL,1
1:      OUT     10H,AL
        ROL     AL,1
        CALL    DELAY
        DEC     BX
        JNZ     S1
        MOV     AL,0FFH
        OUT     10H,AL
        CALL    DELAY
        CALL    DELAY
```

กสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค
ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

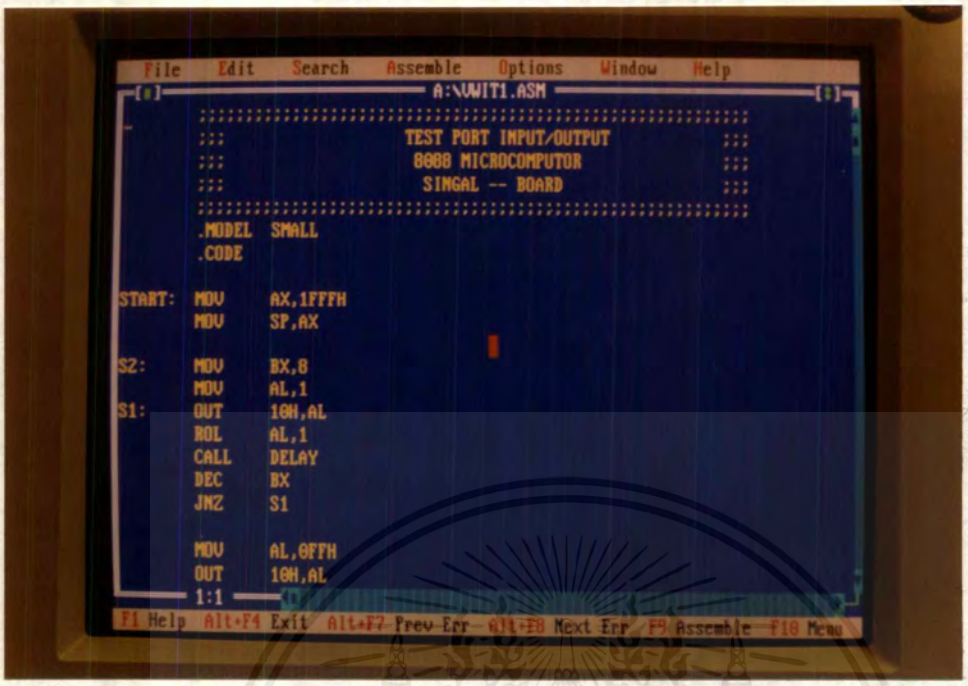
```
        CALL    DELAY
        MOV     AL,0
        OUT     10H,AL
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        JMP     S2
DELAY:  PUSH    CX
        MOV     CX,8000H
D1:     LOOP    D1
        POP     CX
        RET
        ORG     7FF0H
        JMP     START
        END
```

ลักษณะการทำงานของโปรแกรมที่4.2โปรแกรมจะสั่งให้มีการแสดงผลที่พอร์ต 10 H ดยจะแสดงผลให้ แอลอีดี สว่างทีละดวง เมื่อสว่างครบแล้วก็จะสว่างพร้อมกันหมดทุกดวง ละดับพร้อมกันทุกดวงและจะเริ่มสว่างทีละดวงใหม่อีกครั้ง

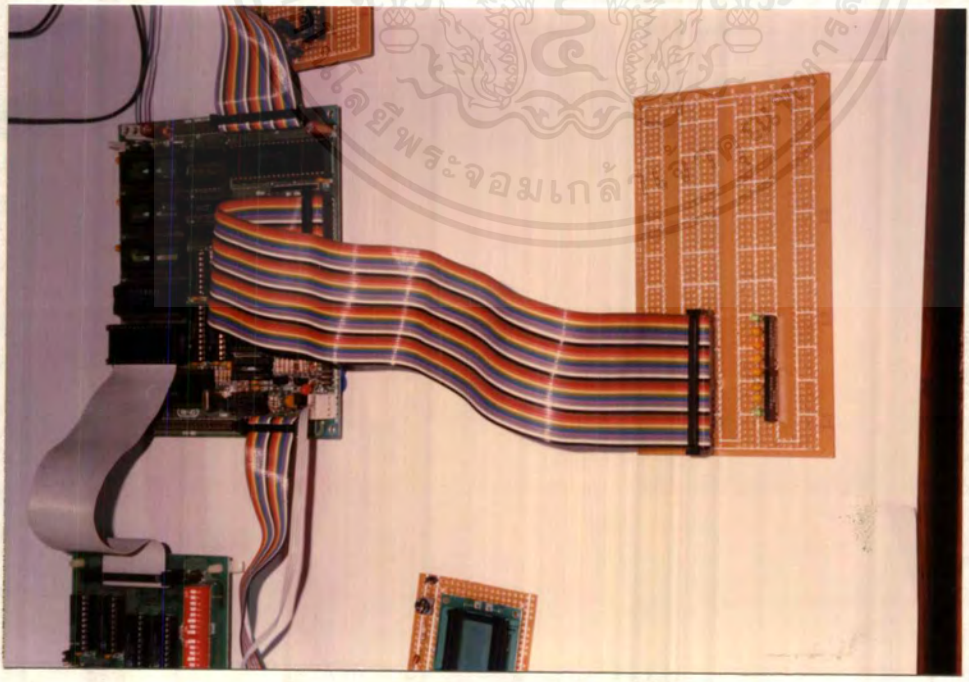
3.2 ผลการทดลอง

เมื่อกดสวิตช์รีเซ็ต แล้วจะทำให้หลอด แอลอีดี แสดงผลตามที่กำหนดในโปรแกรม งแอลอีดี จะสว่างที่ละดวงแล้ววิ่งจากซ้ายไปขวาเมื่อครบรอบแล้วจะดับและจะแสดงผลใหม่ นอยู่เช่นนี้เรื่อยไป

อกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค
ม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.2



รูปที่ 4.3 ต่อวงจรแสดงผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองการทำงานของชุดจอแสดงผลผลึกเหลว

จอแสดงผลผลึกเหลวที่ใช้นั้นเป็นแบบชนิดตัวอักษร มี 14 ขา

4.4.1 ลำดับขั้นการทดลอง

1. นำจุดต่อจาก EPROM EMULATOR ต่อลงบนตำแหน่ง U5 ในชุด 8088
2. ให้พลังงานไฟฟ้าจากแหล่งจ่ายกับชุดควบคุม 8088 ไมโครคอมพิวเตอร์
3. นำจุดต่อจากชุดจอแสดงผลผลึกเหลว ต่อลงบนตำแหน่ง (J11) ของชุด 8088
4. เขียนโปรแกรม 4.3 แล้วทำการแปลงคำสั่งส่งออกจากคอมพิวเตอร์ทางพอร์ตเครื่องพิมพ์
5. กดสวิตช์รีเซ็ต ของชุด 8088 พร้อมสังเกตการทำงานของชุดจอแสดงผลผลึกเหลว

โปรแกรมการทดลองที่ 4.3

```
MODEL  SMALL
CODE
        WR_INST         EQU   60H
        WR_DATA         EQU   62H  ;
        RD_BUSY         EQU   61H  ;
        RD_DATA         EQU   63H
        P_10            EQU   10H
TART:
                MOV     AX,1FFFH
                MOV     SP,AX
                ;-----------------------------------------
LL:             CALL    INI_LCD
                MOV     AL,41H
                MOV     BL,8
T3:             CALL    WR_ASCII
                INC     AL
                DEC     BL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JNZ         ST3
        JMP         LLL
        ;-----------------------------------------
INI_LCD  PROC  NEAR
        MOV         AL,38H      ;INITIAL LCD
        CALL        LCD_COM
        MOV         AL,0FH
        CALL        LCD_COM
        MOV         AL,6
        CALL        LCD_COM
        MOV         AL,1
        CALL        LCD_COM
        RET
        INI_LCD     ENDP
        DELAY       PROC  NEAR
        PUSH        CX
        MOV         CX,0FFFFH
2:      LOOP        D2
        POP         CX
        RET
        DELAY       ENDP
LCD_COM  PROC  NEAR
        CALL        BUSY
        OUT         WR_INST,AL
        RET
LCD_COM             ENDP
WR_ASCII  PROC  NEAR
        CALL        BUSY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค
ม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     AL,'A'
        OUT     WR_DATA,AL
        RET
R_ASCII ENDP
USY     PROC    NEAR
        PUSH    AX
OT_BUSY: IN     AL,RD_BUSY
        TEST    AL,80H
        JNZ     NOT_BUSY
        CALL    DELAY
        POP     AX
        RET
        BUSY    ENDP
        ORG     7FF0H
        JMP     START
        END     START
```

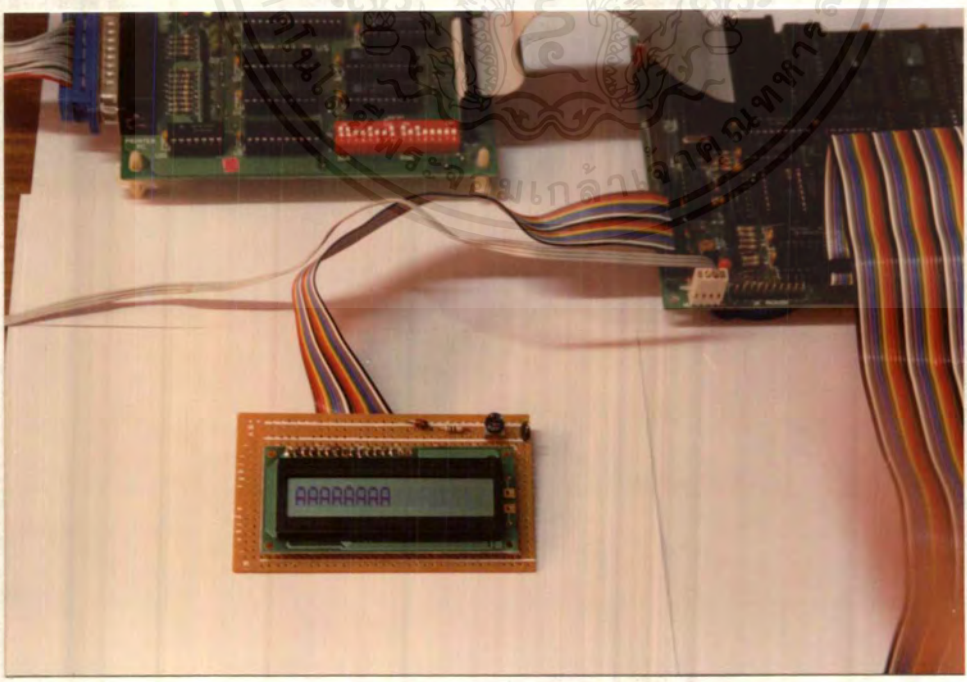
การทำงานของโปรแกรมการทดลองที่ 4.3 จะส่งให้มีการแสดงตัวอักษร A ที่จอแสดงผลึกเหลว โดยจะแสดงที่ละหลักจนครบ 8 หลัก และเมื่อครบ 8 หลักก็จะเริ่มแสดงตัวษร A ซ้ำอีกครั้ง

.2 ผลการทดลอง

เมื่อกดสวิตช์รีเซ็ต จะปรากฏว่ามีตัวอักษร A สว่างที่ตำแหน่งแรกแล้วค่อยๆเลื่อนหน่งไปจนถึงหลักที่แปด ซึ่งครบรอบตามโปรแกรมกำหนดและจะมาเริ่มต้นที่ตำแหน่งอีกครั้ง แสดงว่าชุดไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดียวเบอร์ 8088 สามารถต่อใช้งานส่วนจอแสดงผลผลึกเหลวได้

กสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค
ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.3



รูปที่ 4.5 การต่อวงจรภาคแสดงผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองการทำงานร่วมกันของคีย์บอร์ดกับจอแสดงผลผลึกเหลว

4.5.1 ลำดับขั้นการทดลอง

1. นำจุดต่อจาก EPROM EMULATOR ต่อลงบนตำแหน่ง U5 ในชุด 8088
2. นำจุดต่อจากชุด คีย์บอร์ด ต่อลงบนตำแหน่ง (J7) ของชุด 8088
3. นำจุดต่อจากชุดจอแสดงผลผลึกเหลว ต่อลงบนตำแหน่ง (J11) ของชุด 8088
4. ทำการเขียนโปรแกรมที่ 4.4 แล้วแปลงคำสั่งส่งข้อมูลออกทางพอร์ตเครื่องพิมพ์
5. กดสวิตช์รีเซ็ต ชุด 8088 พร้อมทั้งสังเกตการทำงาน

โปรแกรมการทดลองที่ 4.4

```
.MODEL  SMALL
.CODE
        P_A         EQU   50H
        P_B         EQU   51H
        P_C         EQU   52H
        P_CON       EQU   53H
        WR_INST     EQU   60H
        WR_DATA     EQU   62H  ;
        RD_BUSY     EQU   61H  ;
        RD_DATA     EQU   63H
        P_10        EQU   10H
        ;-------------------------------------
    START   PROC   NEAR
        MOV         AX,1FFFH
        MOV         SP,AX
        CALL        INITIAL
        CALL        ST_KEY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        START   ENDP
        ;-------------------------------------
INITIAL PROC    NEAR
        MOV     AL,88H      ;INITIAL 8255 KEY
        OUT     P_CON,AL
        MOV     AL,38H      ;INITIAL LCD
        CALL    LCD_COM
        MOV     AL,0FH
        CALL    LCD_COM
        MOV     AL,6
        CALL    LCD_COM
        MOV     AL,1
        CALL    LCD_COM
        RET
        INITIAL ENDP
        ;-------------------------------------
ST_KEY  PROC    NEAR
        MOV     CH,00
NEW_COL: MOV    DL,0H
SH_COL: CALL    SCAN
        MOV     AL,DH       ;COLL
        CMP     DH,0
        JZ      SKIPP
        OR      AL,DL
        CALL    NUM_K
SKIPP:  INC     DL
        ;-------------------------------------
        CMP     CH,08H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JNZ         QQ
            MOV         CH,0
            MOV         AL,80H
            CALL        LCD_COM
            MOV         AL,06H
            CALL        LCD_COM
            ;-------------------------------------------
QQ:         CMP         DL,6
            JNZ         SH_COL
            JMP         NEW_COL
            RET
      ST_KEY    ENDP
            ;-------------------------------------------
      SCAN      PROC   NEAR
            PUSH        AX
            MOV         AL,DL     ;COLUM
            OUT         P_C,AL
            IN          AL,P_C
            AND         AL,0F0H
            CMP         AL,0F0H
            JNZ         ONKEY
            MOV         AL,0
ONKEY:      MOV         DH,AL
ENDSCAN:    POP         AX
            RET
      SCAN      ENDP
            ;-------------------------------------------
      NUM_K     PROC   NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     AH,AL
            AND     AH,0FH    ; AS AH=72 NEW AH =02
            AND     AL,0F0H   ; AS AL=72 NEW AH =70
            CMP     AL,70H
            JZ      S_ROW0
            CMP     AL,0B0H
            JZ      S_ROW1
            CMP     AL,0D0H
            JZ      S_ROW2
            CMP     AL,0E0H
            JZ      S_ROW3
S_ROW0:     INC     CH
            MOV     AL,0      ; ROW0 + 00  KEY 0,1,2,3,FU,DEC
            ADD     AL,AH     ; AH = 0 TO 5
            OUT     10H,AL
            CALL    DELAY
            CALL    DIS_FNK
            ADD     AL,30H
            OUT     WR_DATA,AL
            CALL    DELAYLCD
            JMP     SO
S_ROW1:     INC     CH
            MOV     AL,06H    ; ROW1 + 06   KEY 4,5,6,7,RUN,INC
            ADD     AL,AH     ; AH = 0 TO 5
            OUT     10H,AL
            CALL    DELAY
            CALL    DIS_FNK
            ADD     AL,2EH    ; OUT LCD 4,5,6,7 NO KEY RUN INC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        OUT     WR_DATA,AL
        CALL    DELAYLCD
        JMP     SO
S_ROW2: INC     CH
        MOV     AL,0CH      ; ROW2 + 12   KEY 8,9,A,B,DATA,STEP
        ADD     AL,AH       ; AH = 0 TO 5
        OUT     10H,AL
        CALL    DELAY
        CALL    DIS_FNK
        CMP     AH,02H
        JNZ     NK_A
        JMP     K_AB
NK_A:   CMP     AH,03H
        JNZ     NK_B
K_AB:   ADD     AL,07H
NK_B:   ADD     AL,2CH      ; OUT LCD 8,9,A,B NO KEY DATA STEP
OUT_AB: OUT     WR_DATA,AL
        CALL    DELAYLCD
        JMP     SO
S_ROW3: INC     CH
        MOV     AL,12H      ;ROW3 + 18  KEY C,D,E,F,REG,MON
        ADD     AL,AH
        OUT     10H,AL
        CALL    DELAY
        CALL    DIS_FNK
        ADD     AL,31H      ; OUT LCD C,D,E,F NO KEY REG,MON
        OUT     WR_DATA,AL
        CALL    DELAYLCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SO:         RET
            NUM_K   ENDP
            ;-----------------------------------

        DIS_FNK   PROC  NEAR
            CMP         AH,04H    ;NO DISPLAY KEY FUNC
            JZ          SO
            CMP         AH,05H
            JZ          SO
            RET
        DIS_FNK   ENDP
            ;-----------------------------------
        DELAY   PROC  NEAR
            PUSH        CX
            MOV         CX,1FFH
            LOOP        $
            POP         CX
            RET
            DELAY       ENDP
            ;-----------------------------------
        DISPLAY   PROC  NEAR;
LLL:        CALL        INI_LCD   ;PRO LCD
            MOV         AL,41H
            MOV         BL,8
ST3:        CALL        WR_ASCII
            INC         AL
            DEC         BL
            JNZ         ST3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;       JMP         LLL
        RET

        DISPLAY     ENDP
        ;-------------------------------------
DELAYLCD  PROC  NEAR
        PUSH        CX
        MOV         CX,0FFFFH
DD2:    LOOP        DD2
        POP         CX
        RET
DELAYLCD  ENDP
        ;-------------------------------------
LCD_COM   PROC  NEAR
        CALL        BUSY
        OUT         WR_INST,AL
        RET
LCD_COM   ENDP
        ;-------------------------------------
WR_ASCII  PROC  NEAR
        CALL        BUSY
        MOV         AL,'A'
        OUT         WR_DATA,AL
        RET
WR_ASCII  ENDP
        ;-------------------------------------
BUSY    PROC  NEAR
        PUSH        AX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NOT_BUSY:  IN      AL,RD_BUSY
           TEST    AL,80H
           JNZ     NOT_BUSY
           CALL    DELAY
           POP     AX
           RET
           BUSY    ENDP
           ;-----------------------------------------
           ORG     7FF0H
           JMP     START
           END     START
```

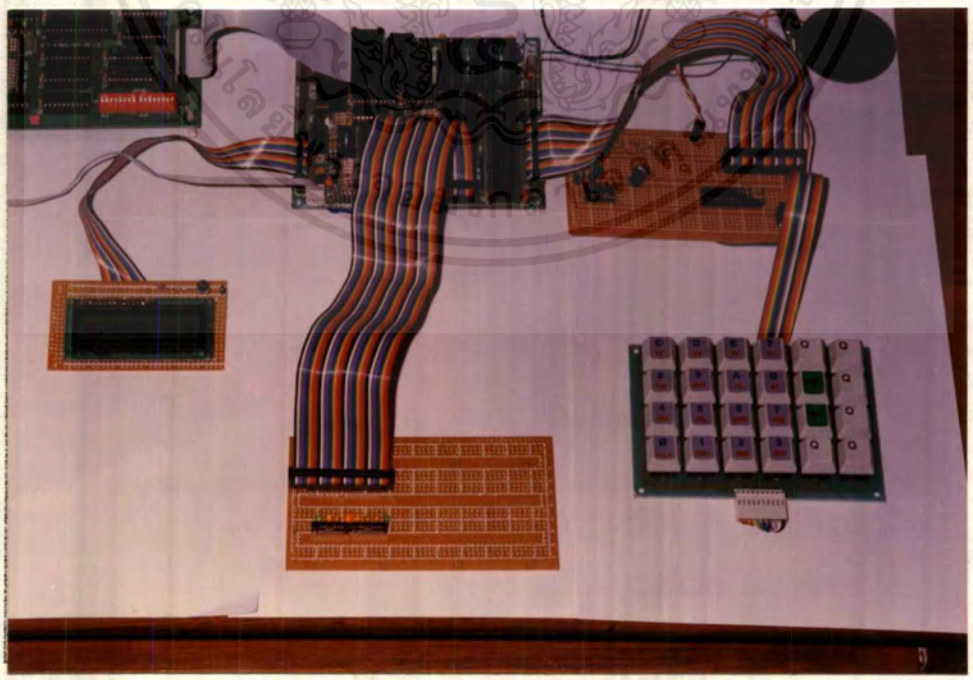
การทำงานของโปรแกรมการทดลองที่ 4.4 จะเป็นการรวมโปรแกรมที่ 4.1 กับโปรแกรมที่ 4.3 ให้สามารถทำงานรวมกันได้ โดยที่จะนำค่าของรีจิสเตอร์ AL มาเทียบกับรหัสแอสกี แล้วส่งออกแสดงผลที่จอผลึกเหลวแอลซีดี โดยจะตรงกับตำแหน่งของแป้นพิมพ์

4.5.2 ผลการทดลอง

เมื่อกดสวิตช์รีเซ็ต แล้วทำการกดสวิตช์บนแป้นพิมพ์ ผลที่จดแสดงผลผลึกเหลวปรากฏตัวอักษรตรงกับตำแหน่งของแป้นพิมพ์ทุกตำแหน่งแสดงว่าสามารถทำงานระหว่างชุดจอแสดงผลผลึกเหลว กับชุดคีย์บอร์ดได้ โดยโปรแกรมที่ 4.4 จะทำให้จอแสดงผลผลึกเหลวสามารถแสดงครบ 16 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.4



รูปที่ 4.7 การต่อวงจรทดลองและผลการทดลองโปรแกรมการทดลองที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดสอบการควบคุมการแสดงผลของจอแสดงผลผลึกเหลว

4.6.1 ลำดับขั้นการทดลอง

1. นำจุดต่อจาก EPROM EMULATOR ต่อลงบนตำแหน่ง U5 ในชุด 8088
2. นำจุดต่อจากชุด คีย์บอร์ด ต่อลงบนตำแหน่ง (J7) ของชุด 8088
3. นำจุดต่อจากชุดจอแสดงผลผลึกเหลวลงบนตำแหน่ง (J11) ของชุด 8088
4. ทำการเขียนโปรแกรมที่ 4.5 แล้วแปลงคำสั่งส่งข้อมูลออกทางพอร์ตเครื่องพิมพ์มายัง EPROM EMULATOR
5. กดสวิตช์รีเซ็ต ชุด 8088 พร้อมทั้งสังเกตการทำงาน

โปรแกรมการทดลองที่ 4.5

```
.MODEL  SMALL
.CODE
        P_A         EQU   50H
        P_B         EQU   51H
        P_C         EQU   52H
        P_CON       EQU   53H
        WR_INST     EQU   60H
        WR_DATA     EQU   62H  ;
        RD_BUSY     EQU   61H  ;
        RD_DATA     EQU   63H
        P_10        EQU   10H
        ;-----------------------------------------
START   PROC  NEAR
        MOV         AX,1FFFH
        MOV         SP,AX
        CALL        INITIAL
        CALL        ST_KEY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        START       ENDP
        ;-----------------------------------------
INITIAL  PROC  NEAR
        MOV         AL,88H      ;INITIAL 8255 KEY
        OUT         P_CON,AL
        MOV         AL,38H      ;INITIAL LCD
        CALL        LCD_COM
        MOV         AL,0FH
        CALL        LCD_COM
        MOV         AL,6
        CALL        LCD_COM
        MOV         AL,1
        CALL        LCD_COM
        RET
        INITIAL  ENDP
        ;-----------------------------------------
ST_KEY   PROC  NEAR
        MOV         CH,00
NEW_COL: MOV        DL,0H
SH_COL:  CALL       SCAN
        MOV         AL,DH       ;COLL
        CMP         DH,0
        JZ          SKIPP
        OR          AL,DL
        CALL        NUM_K
SKIPP:  INC         DL
        ;-----------------------------------------
        CMP         CH,08H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JNZ         QQ
                MOV         CH,0
                MOV         AL,1
                CALL        LCD_COM
        ;       MOV         AL,06H
        ;       CALL        LCD_COM
                ;-----------------------------------------
QQ:             CMP         DL,6
                JNZ         SH_COL
                JMP         NEW_COL
                RET
        ST_KEY  ENDP
                ;-----------------------------------------
        SCAN    PROC  NEAR
                PUSH        AX
                MOV         AL,DL    ;COLUM
                OUT         P_C,AL
                IN          AL,P_C
                AND         AL,0F0H
                CMP         AL,0F0H
                JNZ         ONKEY
                MOV         AL,0
ONKEY:          MOV         DH,AL
ENDSCAN:        POP         AX
                RET
                SCAN        ENDP
                ;-----------------------------------------
        NUM_K   PROC  NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     AH,AL
        AND     AH,0FH    ; AS AH=72 NEW AH =02
        AND     AL,0F0H   ; AS AL=72 NEW AH =70
        CMP     AL,70H
        JZ      S_ROW0
        CMP     AL,0B0H
        JZ      S_ROW1
        CMP     AL,0D0H
        JZ      S_ROW2
        CMP     AL,0E0H
        JZ      S_ROW3
S_ROW0: INC     CH
        MOV     AL,0      ; ROW0 + 00  KEY 0,1,2,3,FU,DEC
        ADD     AL,AH     ; AH = 0 TO 5
        OUT     10H,AL
        CALL    DELAY
        CALL    DIS_FNK
        ADD     AL,30H
        OUT     WR_DATA,AL
        CALL    DELAYLCD
        JMP     SO
S_ROW1: INC     CH
        MOV     AL,06H    ; ROW1 + 06   KEY 4,5,6,7,RUN,INC
        ADD     AL,AH     ; AH = 0 TO 5
        OUT     10H,AL
        CALL    DELAY
        CALL    DIS_FNK
        ADD     AL,2EH    ; OUT LCD 4,5,6,7 NO KEY RUN INC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        OUT     WR_DATA,AL
        CALL    DELAYLCD
        JMP     SO
S_ROW2: INC     CH
        MOV     AL,0CH      ; ROW2 + 12  KEY 8,9,A,B,DATA,STEP
        ADD     AL,AH       ; AH = 0 TO 5
        OUT     10H,AL
        CALL    DELAY
        CALL    DIS_FNK
        CMP     AH,02H
        JNZ     NK_A
        JMP     K_AB
NK_A:   CMP     AH,03H
        JNZ     NK_B
K_AB:   ADD     AL,07H
NK_B:   ADD     AL,2CH      ; OUT LCD 8,9,A,B NO KEY DATA STEP
OUT_AB: OUT     WR_DATA,AL
        CALL    DELAYLCD
        JMP     SO
S_ROW3: INC     CH
        MOV     AL,12H      ;ROW3 + 18  KEY C,D,E,F,REG,MON
        ADD     AL,AH
        OUT     10H,AL
        CALL    DELAY
        CALL    DIS_FNK
        ADD     AL,31H      ; OUT LCD C,D,E,F NO KEY REG,MON
        OUT     WR_DATA,AL
        CALL    DELAYLCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SO:             RET
        NUM_K   ENDP
                ;-----------------------------------------


        DIS_FNK PROC  NEAR
                CMP             AH,04H      ;NO DISPLAY KEY FUNC
                JZ              SO
                CMP             AH,05H
                JZ              SO
                RET
        DIS_FNK ENDP
                ;-----------------------------------------
        DELAY   PROC  NEAR
                PUSH            CX
                MOV             CX,1FFH
                LOOP            $
                POP             CX
                RET
                DELAY           ENDP
                ;-----------------------------------------
        DISPLAY PROC  NEAR
;LLL:           CALL            INI_LCD     ;PRO LCD
                MOV             AL,41H
                MOV             BL,8
ST3:            CALL            WR_ASCII
                INC AL
                DEC BL
                JNZ ST3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า . ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;       JMP LLL
        RET

        DISPLAY     ENDP
        ;-----------------------------------------
DELAYLCD  PROC  NEAR
        PUSH        CX
        MOV         CX,0FFFFH
DD2:    LOOP        DD2
        POP         CX
        RET
        DELAYLCD  ENDP
        ;-----------------------------------------
LCD_COM   PROC  NEAR
        CALL        BUSY
        OUT         WR_INST,AL
        RET
LCD_COM   ENDP
        ;-----------------------------------------
WR_ASCII  PROC  NEAR
        CALL        BUSY
        MOV         AL,'A'
        OUT         WR_DATA,AL
        RET
        WR_ASCII    ENDP
        ;-----------------------------------------
BUSY    PROC  NEAR
        PUSH        AX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

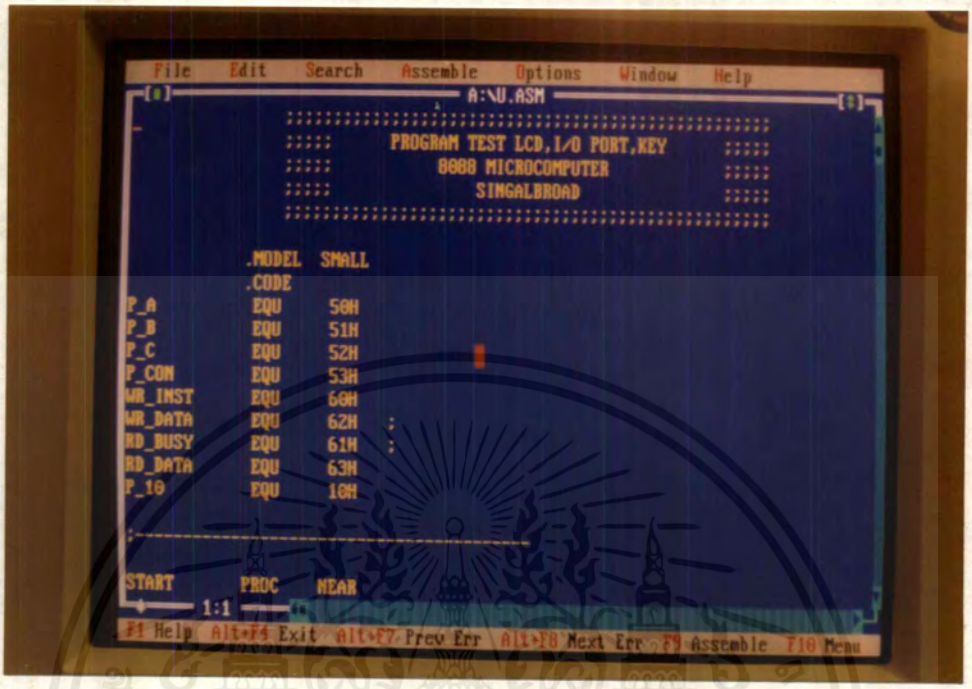
```
NOT_BUSY:  IN      AL,RD_BUSY
           TEST    AL,80H
           JNZ     NOT_BUSY
           CALL    DELAY
           POP     AX
           RET
           BUSY    ENDP
           ;-----------------------------------------
           ORG     7FF0H
           JMP     START
           END     START
```

การทำงานของโปรแกรมการทดลองที่ 4.5 โดยการนำเอาโปรแกรมการทดลองที่ 4.4 มาใช้งานแต่ควบคุมการแสดงผล โดยจะให้มีการแสดงผลในตำแหน่งที่ 3 จนถึงตำแหน่งที่ 6 และถ้ามีการกดปุ่ม EDIT จะเลื่อนการแสดงผลมาที่ตำแหน่งที่ 11 ถึง 12 แทนและจะกลับไปแสดงที่ตำแหน่งที่ 3 ถึงตำแหน่งที่ 6 ใหม่อีกครั้ง

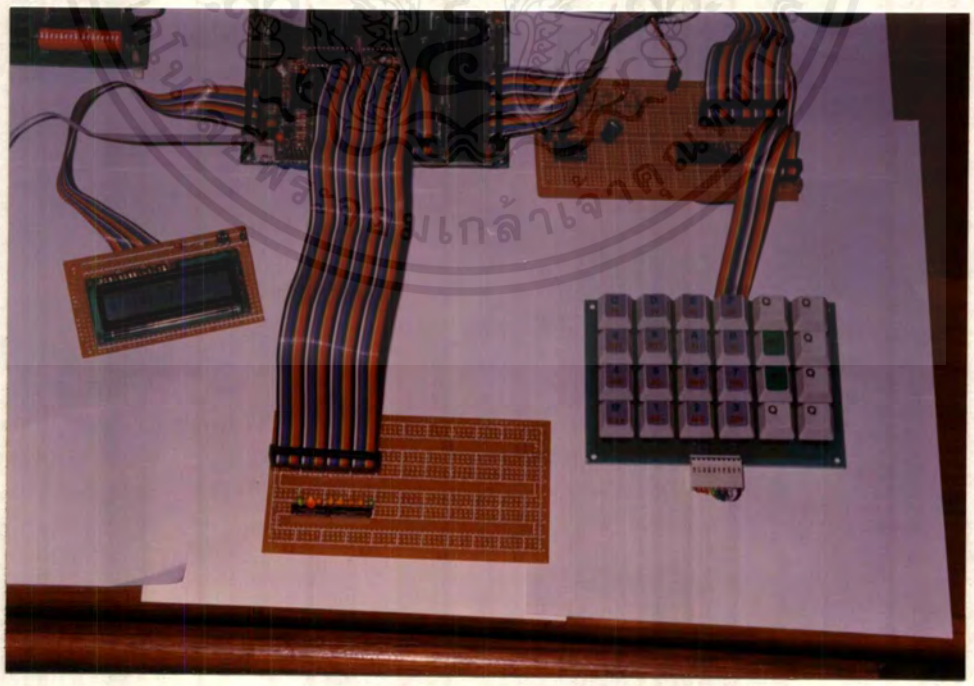
4.6.2 ผลการทดลอง

การทดลองของโปรแกรมที่ 4.5 จะทำการควบคุมการแสดงผลของจอแสดงผลผลึกเหลว ให้แสดงที่ หลักที่ 4 จนถึงหลักที่ 7 และหลักที่ 13ถึงหลักที่ 16 ผลที่จากการทดลองจากโปรแกรม ที่จอแสดงผลผลึกเหลวสามารถแสดงผลได้อย่างถูกต้องตามที่โปรแกรมที่ 4.5 กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 จอภาพการเตรียมแปลงคำสั่งโปรแกรมการทดลองที่ 4.5



รูปที่ 4.7 การต่อวงจรทดลองและผลการทดลองโปรแกรมการทดลองที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปอภิปรายและข้อเสนอแนะ

5.1 บทสรุป

ไมโครคอมพิวเตอร์แผ่นพิมพ์เดี่ยวเบอร์ 8088 ในปริญญานิพนธ์ฉบับนี้ สร้างขึ้นเพื่อนำไปใช้ในการศึกษาเรียนรู้การทำงานต่างๆของไมโครคอมพิวเตอร์เบอร์ 8088 ซึ่งเป็นไมโครคอมพิวเตอร์ที่มีประสิทธิภาพสูงตัวหนึ่ง ในการสร้างคณะผู้จัดทำได้กำหนดขอบเขตการสร้างและพัฒนาเพื่อเป็นแนวทางในการออกแบบไว้ดังนี้

- ใช้ไมโครโปรเซสเซอร์เบอร์ 8088 เป็นซีพียู

- การพัฒนาโปรแกรมสามารถป้อนโปรแกรมโดยการใส่ออฟโค้ดทางคีย์บอร์ดของเครื่องและสามารถจะพัฒนาโปรแกรมผ่านทางเครื่องคอมพิวเตอร์ โดยเขียนเป็นภาษาแอสแซมบลี้บนเครื่องคอมพิวเตอร์แล้วส่งผ่านทางพอร์ตเครื่องพิมพ์ กับเครื่องอีมูเลเตอร์

- ใช้อีพรอมขนาด 32 กิโลไบต์ เก็บโปรแกรมมอนิเตอร์ของไมโครคอมพิวเตอร์แผ่นพิมพ์เดี่ยวเบอร์ 8088

- มอนิเตอร์โปรแกรมการแสดงผลของไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 จะแสดงผลบนจอผลึกเหลวแบบ 16 ตัวอักษร 1 แถว โดยอาศัยโปรแกรมควบคุมเอาต์พุทเป็นตัวเลือกการแสดงผลและแสดงผลทางอุปกรณ์เอาต์พุต

- มีพอร์ตซึ่งมีหน้าที่เป็นทั้งอินพุตและเอาต์พุตโดยพอร์ตจะอยู่ที่ตำแหน่ง 10H และ 20H มีขนาด 8 บิท

- 8255 อินพุต / เอาต์พุตใช้เป็นขาสัญญาณควบคุมอินพุต / เอาต์พุตใช้ในการต่อร่วมกับอุปกรณ์อื่นได้ อาทิเช่น แป้นพิมพ์ เป็นต้น

5.2 ปัญหาและแนวทางแก้ไข

ในระหว่างการทำโครงงานพิเศษในครั้งนี้ คณะผู้จัดทำได้พบกับปัญหาและอุปสรรคต่างๆที่ทำให้สูญเสียเวลาในการแก้ไขข้อบกพร่องผิดพลาดเหล่านี้บางปัญหาอาจมองดูแล้วเล็กน้อย แต่ถ้าไม่จัดการแก้ไขก็จะทำให้เสียเวลามากในการแก้ไขภายหลังซึ่งปัญหาต่างๆและแนวทางแก้ไข ได้แสดงให้เห็นดังตารางที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 ปัญหาและแนวทางการแก้ไขปัญหาในการทำปริญญานิพนธ์

ปัญหา	แนวทางแก้ไข
1.โปรแกรมมอนิเตอร์ ในการเขียนนั้นตอนแรกในการศึกษาต้องเขียนด้วยภาษาเบสิก ซึ่งทำให้เกิดความลำบากในการเขียนมากจึงเปลี่ยนมาใช้ภาษาแอสแซมบลี้ ในการเขียนโปรแกรมมอนิเตอร์แทน	1.ทำการยกเลิกการติดต่อใช้อินเทอร์รัพต์แบบ NMI ไม่ใช้ระบบ REAL TIME CLOCK โดยถอด ไอซีเบอร์ 74HC4020 ออกต่อสัญญาณอินเทอร์รัพต์ที่ขา 3 ของ U12
2.ในการเขียนโปรแกรมทดลองพอร์ตให้แสดงผลออกที่จอแสดงผลผลึกเหลว ปรากฏว่าไม่แสดงผลตามที่กำหนด	2.ทำการตรวจสอบโปรแกรมที่เขียนทดลองการทำงานปรากฏว่าทำการติดต่อผิดตำแหน่งที่จอผลึกเหลวแอลซีดีต่ออยู่
3.การทดลองวงจร D/A เพื่อให้ลำโพงส่งเสียงสัญญาณออกมา ผลที่ปรากฏไม่มีเสียงสัญญาณออกทางลำโพง	3.ตรวจสอบดูที่สายที่ทำการบัดกรีว่าตรงตำแหน่งที่วงจรออกแบบหรือเปล่า ปรากฏว่า ไม่ตรงตามที่ได้ออกแบบไว้ทำการแก้ไขให้ตรงกับจุดที่ออกแบบ
4.ถึงแม้ว่าได้ทำการดัดแปลงแก้ไขวิธีการเขียนโปรแกรมมอนิเตอร์ที่ใช้งานได้แล้วแต่เนื่องด้วยการวางแผนไม่เป็นขั้นตอน และประสบการณ์ จึงทำให้โปรแกรมมอนิเตอร์มีความล่าช้า	4.ทำการออกแบบและวางแผนใหม่ให้เป็นไปตามลำดับขั้นที่ถูกต้องของการเขียนโปรแกรมมอนิเตอร์
5.ในการออกแบบทางด้านฮาร์ดแวร์ โดยใช้ CPU เบอร์ 8086 แต่เมื่อออกแบบแล้วไม่สามารถที่จะติดต่อกับหน่วยความจำได้	5.ทำการยกเลิกการออกแบบวงจรที่ใช้ CPU เบอร์ 8086 แล้วทำการใช้แผงวงจรที่มี CPU เบอร์ 8088 แทน
6.ในการวาดวงจรรวมของชุด ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 ก่อนที่จะสามารถวาดนั้น เราจะต้องทราบรายละเอียด	6.ในการศึกษาการทำงานของโปรแกรมสำเร็จรูป Protel for Windows Schmatic และ PCB Design ควรที่จะศึกษามาก่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1(ต่อ) ปัญหาและแนวทางการแก้ไขปัญหาในการทำปริญญานิพนธ์

ปัญหา	แนวทางแก้ไข
ของการต่อวงจรโดยรวมแล้วจึงนำมาวาดวงจรรวมบนโปรแกรมสำเร็จรูป Protel for Windows Schmatic เนื่องจากเป็นการศึกษาการใช้งานในครั้งแรกจึงทำให้เกิดความล่าช้าในการวาดวงจรซึ่งในการวางจุดต่อสายกับขาอุปกรณ์เกิดการวางซ้อนกันและรายละเอียดของอุปกรณ์แต่ละชนิดที่จะนำไปแปลงไฟล์ข้อมูลเป็นไฟล์ที่มีนามสกุล *.NET เกิดความผิดพลาดในการนำตัวอุปกรณ์มาวางบนแผ่นวงจรพิมพ์ในโปรแกรมสำเร็จรูป Protel for Windows PCB Design	จะทำโครงงาน ส่วนของปัญหาในการวางซ้อนกันจะต้องทำการตรวจโดยขยายรูปวงจรให้มีขนาดใหญ่แล้วตรวจลายเส้นวงจรทีละเส้น ส่วนในการนำไฟล์ข้อมูลมาแปลงเป็น *. NET ต้องเข้าไปตรวจสอบที่เมนู PART ที่ช่อง Footprint แล้วเติมลักษณะของตัวอุปกรณ์เช่น ไอซี เบอร์ 74LS32 ให้เติมลักษณะตัวอุปกรณ์ในช่อง Footprint เป็น DIP 14 แล้วนำไปแปลงเป็นไฟล์ *.NET จะปรากฏ ตัวอุปกรณ์บนแผ่นวงจรพิมพ์ตามที่เราได้กำหนดรูปแบบไว้ข้างต้น
7.เนื่องด้วยประสบปัญหาในการติดต่อระหว่างCPU เบอร์ 8086 กับ หน่วยความจำทำให้ต้องมีการเปลี่ยนแปลง CPU เป็นเบอร์ 8088 ทำให้ต้องทำการวาดและออกแบบวงจรแผ่นพิมพ์เดี่ยวใหม่ทั้งหมดทำให้เกิดความล่าช้าและไม่สมบูรณ์ของแผ่นวงจรพิมพ์	7.ต้องใช้การออกแบบที่ชำนาญและรวดเร็วในการออกแบบแผ่นวงจรพิมพ์

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

คณะผู้จัดทำได้พยายามจัดทำและสร้างไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 ซึ่งให้มีประสิทธิภาพและสมบูรณ์ที่สุด เท่าที่จะทำได้โดยตลอดภาคการศึกษาได้จัดได้ทำการพัฒนามาอย่างต่อเนื่อง ถึงแม้ว่าจะต้องแก้ไขในรายละเอียดทางด้านฮาร์ดแวร์ก็ตามแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากระยะที่จำกัดตลอดจนขาดประสบการณ์ในการเขียนโปรแกรมมอนิเตอร์พอสมควร ซึ่งจากการเขียนโปรแกรมที่ผ่านมาจึงมีข้อคิดที่จะเสนอ คิดว่าผู้ที่จะเขียนโปรแกรมมอนิเตอร์ควรที่จะทำการศึกษาโปรแกรมมอนิเตอร์ของบริษัทอื่นๆไว้ด้วยรวมทั้งมีการวางแผนเป็นลำดับขั้นที่แน่นอนซึ่งจะส่งผลให้ได้โปรแกรมที่สมบูรณ์และใช้เวลาในการเขียนไม่มาก

ระยะเวลาที่ผ่านมาในการจัดทำชุดไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 มีข้อจำกัดและข้อบกพร่องที่ควรจะได้รับการพัฒนาและแก้ไขให้มีประสิทธิภาพต่อไปดังนี้

1.ซีพียูที่ใช้ในการประมวลผลการทำงานควรจะพัฒนาให้มีความสามารถที่สูงขึ้น อาทิ เช่น ไมโครโปรเซสเซอร์เบอร์ 8086 , 80186 , 80286 ตลอดจนเพนเทียมเป็นต้น

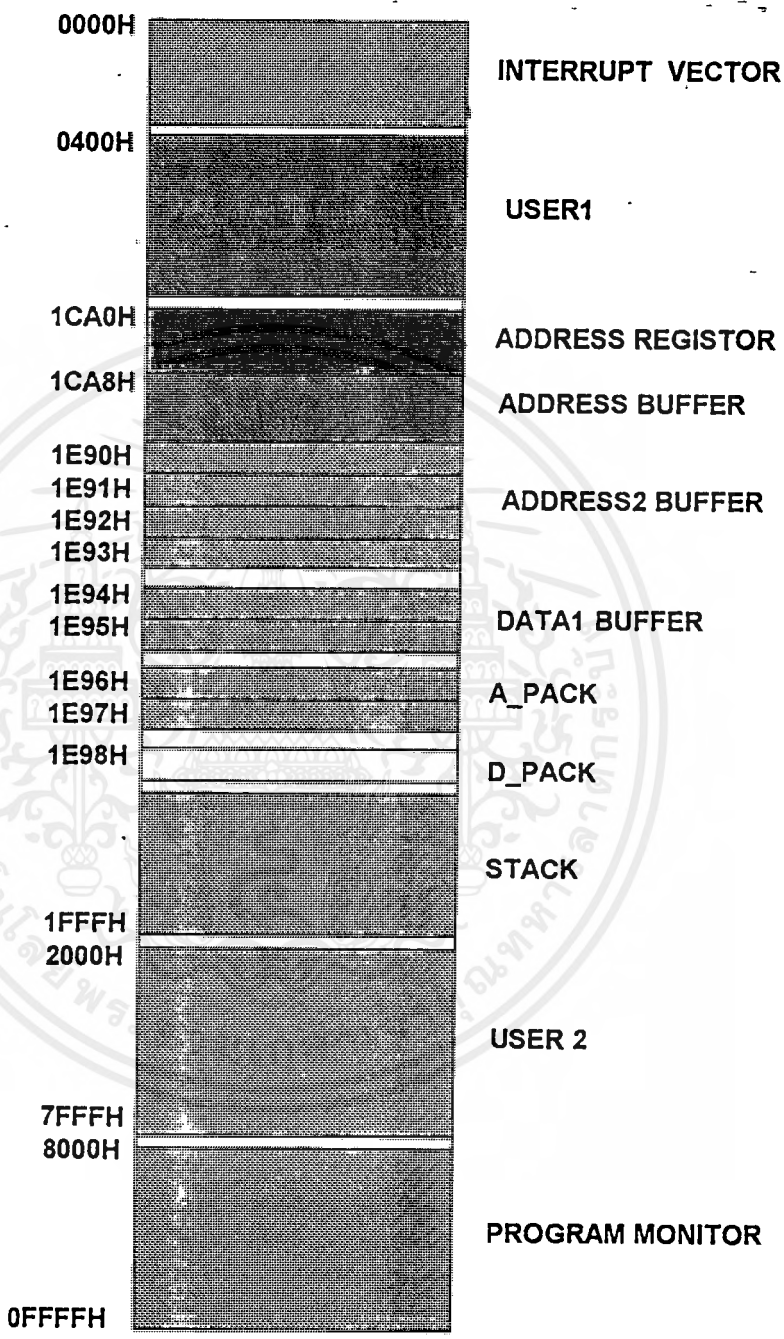
2.ในการพัฒนาการออกแบบทางด้านฮาร์ดแวร์ ควรมีการขยายพอร์ตในการติดต่อให้เพิ่มมากขึ้นกว่าที่มีอยู่ตลอดจนสามารถใช้งานได้

3.โปรแกรมมอนิเตอร์ที่เขียนขึ้นยังมีข้อบกพร่องอยู่ควรได้มีการพัฒนาให้มีความสมบูรณ์และมีประสิทธิภาพมากยิ่งขึ้น เช่น มีฟังค์ชั่นการใช้งานมากขึ้นและเขียนโปรแกรมส่ง-รับข้อมูลทางพอร์ตสื่อสารอนุกรม RS - 232 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การจัดหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

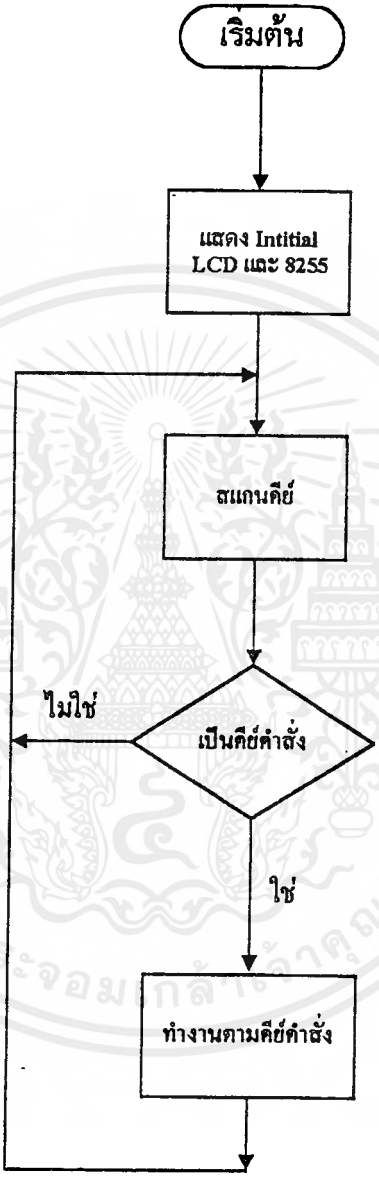


รูปที่ ก.1 ผังการจัดหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

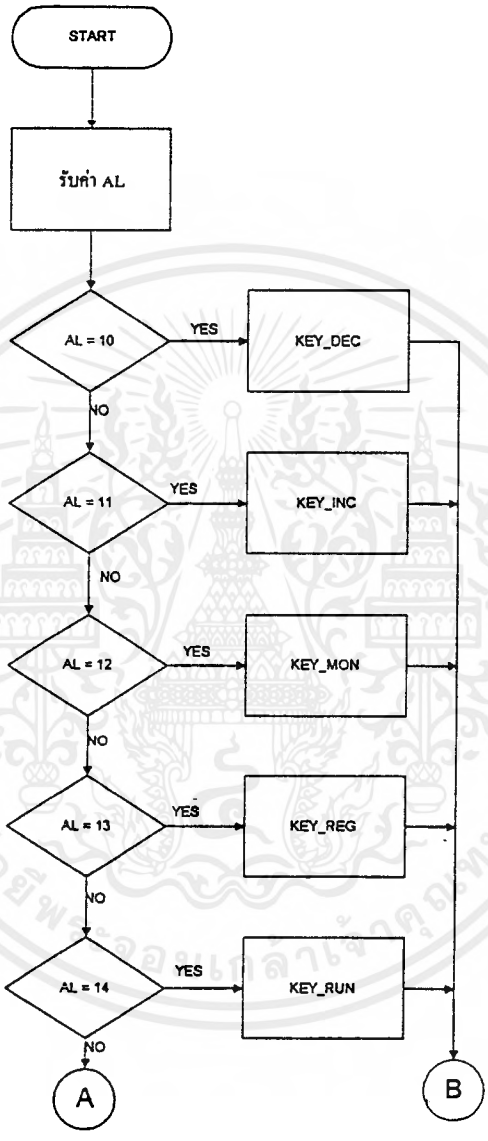
ภาคผนวก ข
ผังการทำงานและโปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



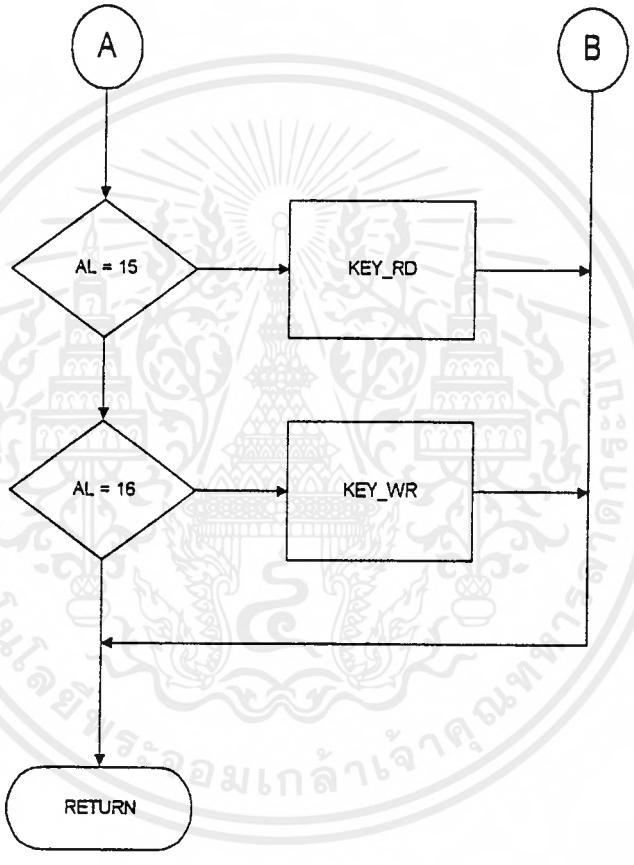
รูปที่ ข.1 ผังการทำงานโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



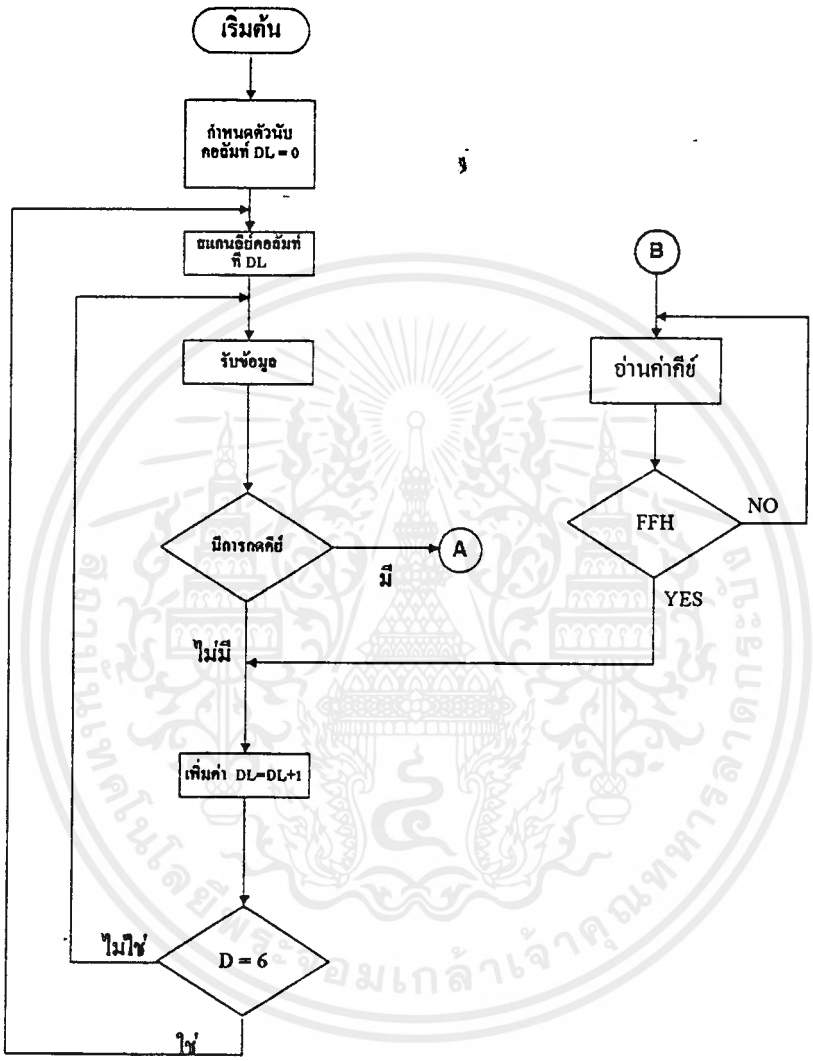
รูปที่ ข.2 ผังการทำ FUNCTION KEY_FUNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



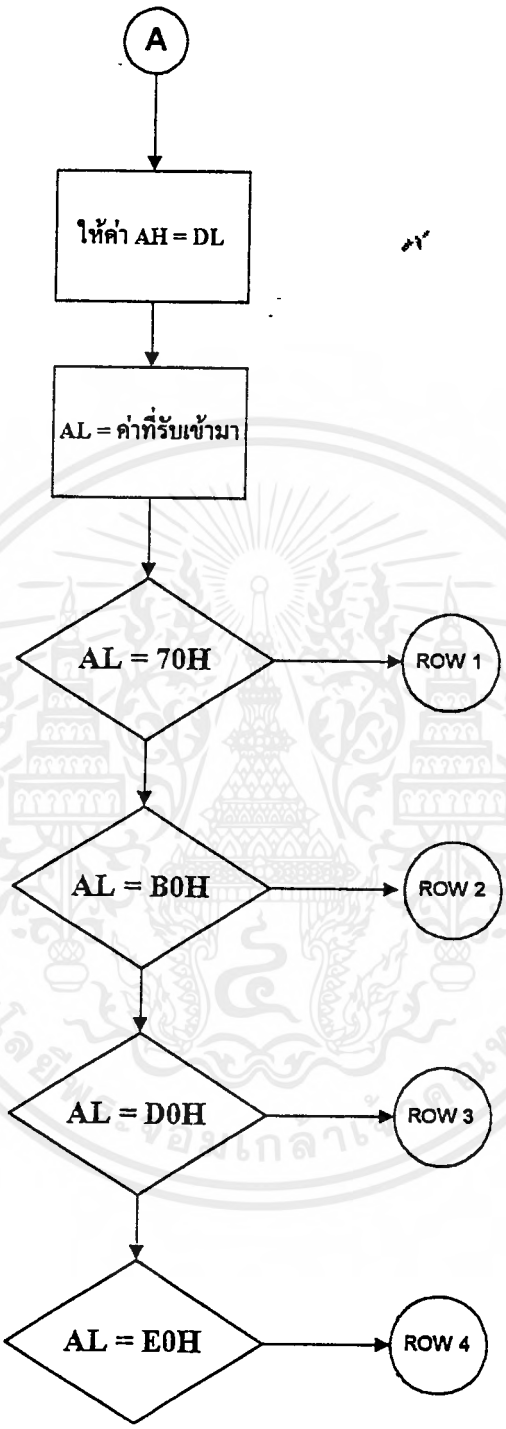
รูปที่ ข.2(ต่อ) ผังการทำ FUNCTION KEY_FUNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



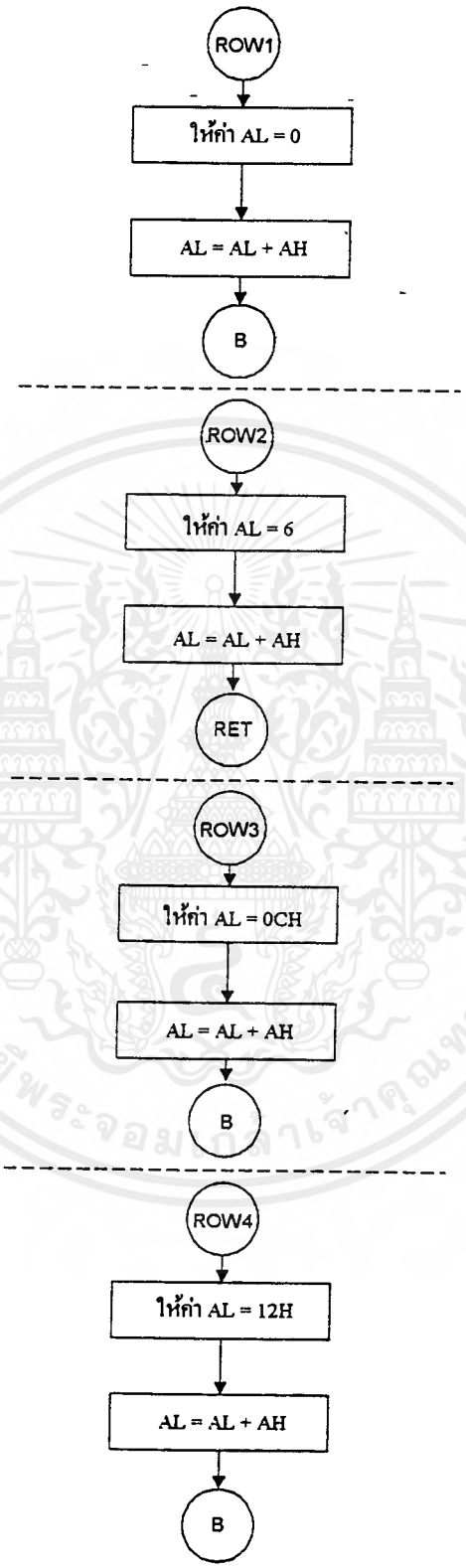
รูปที่ ข.3 ผังการทำงาน FUNCTION GETCH_

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



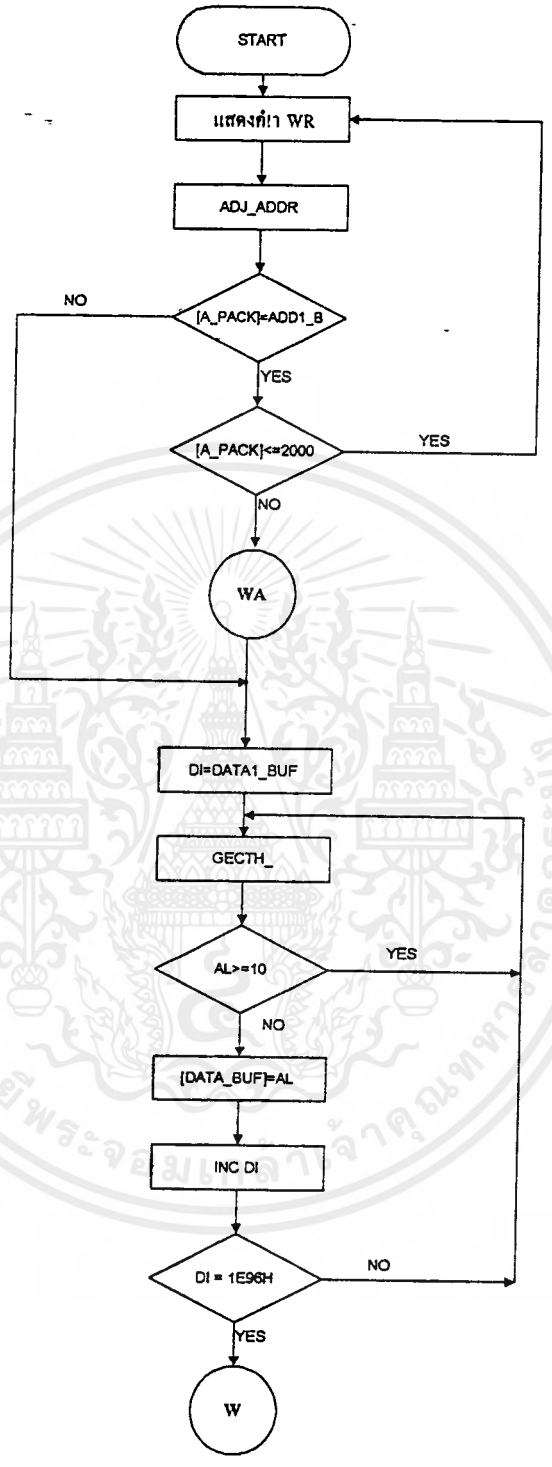
รูปที่ ข.3 (ต่อ) ผังการทำงาน FUNCTION GETCH_

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



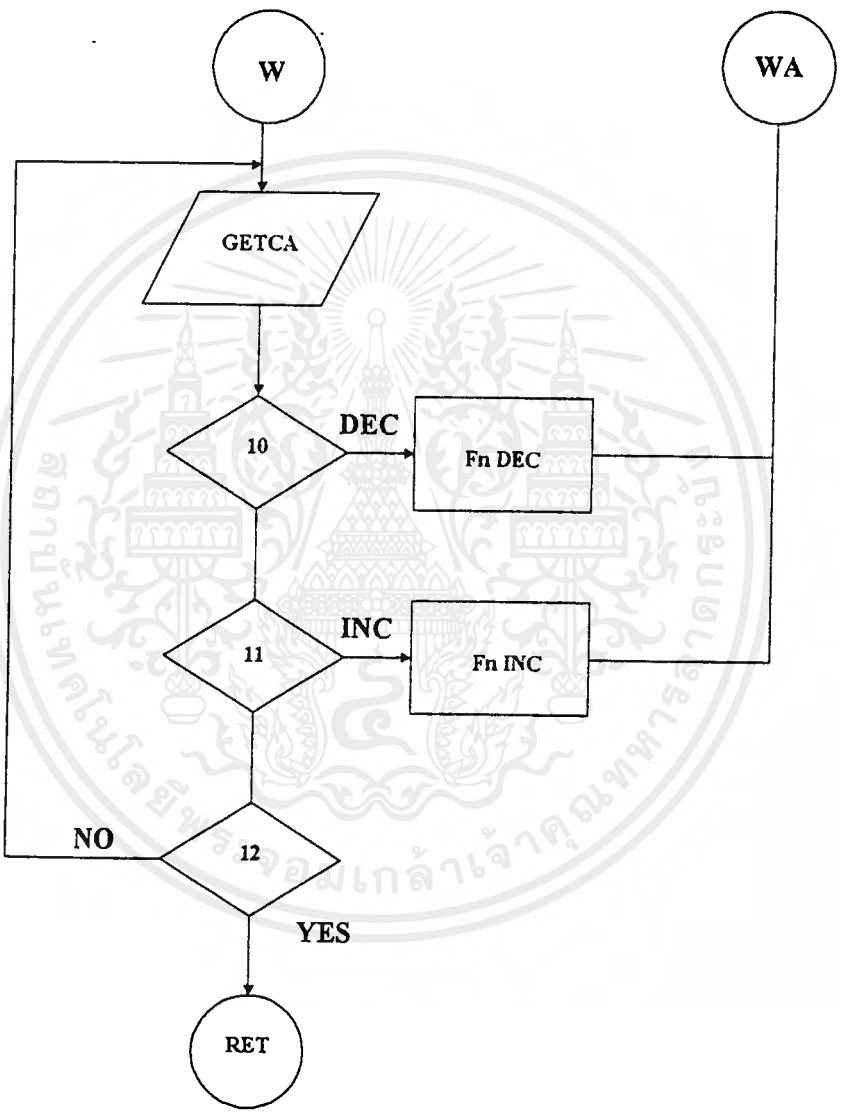
รูปที่ ข.3 (ต่อ) ผังการทำงาน FUNCTION GETCH_

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



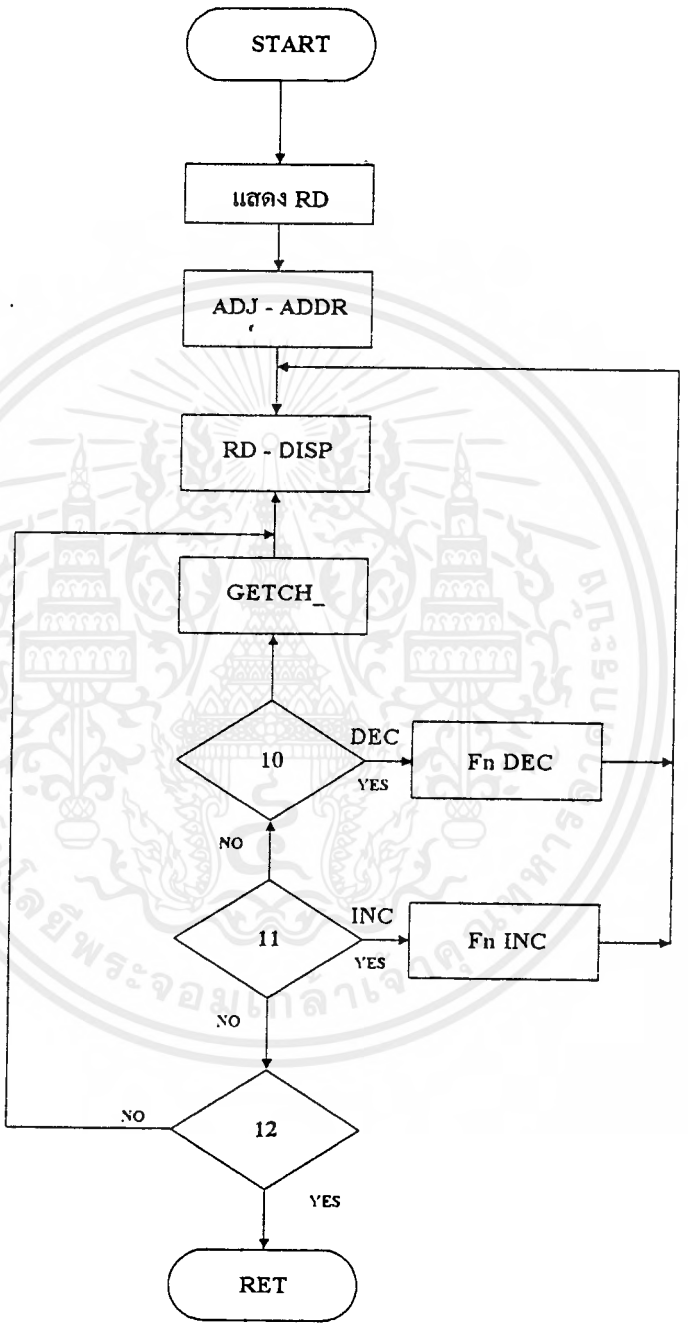
รูปที่ ข.4 ผังการทำงาน FUNCTION K_WR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



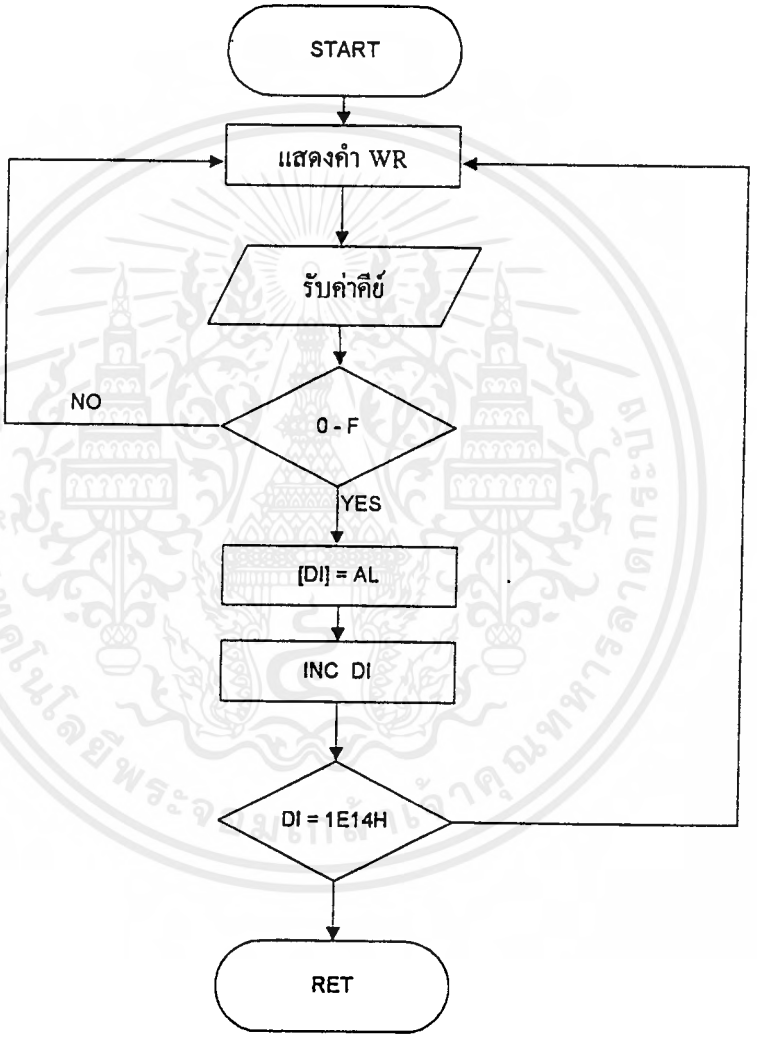
รูปที่ ข.4 (ต่อ) ผังการทำงาน FUNCTION K_WR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



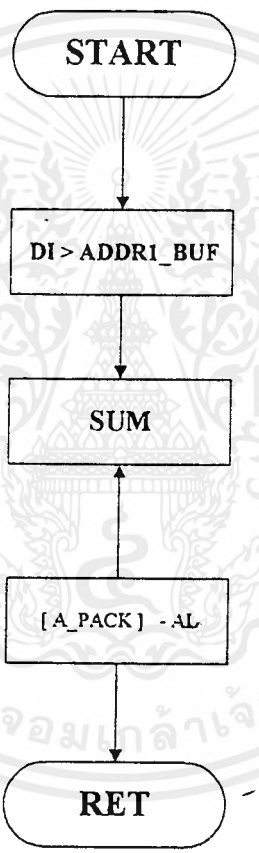
รูปที่ ข.5 ผังการทำงาน FUNCTION READ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



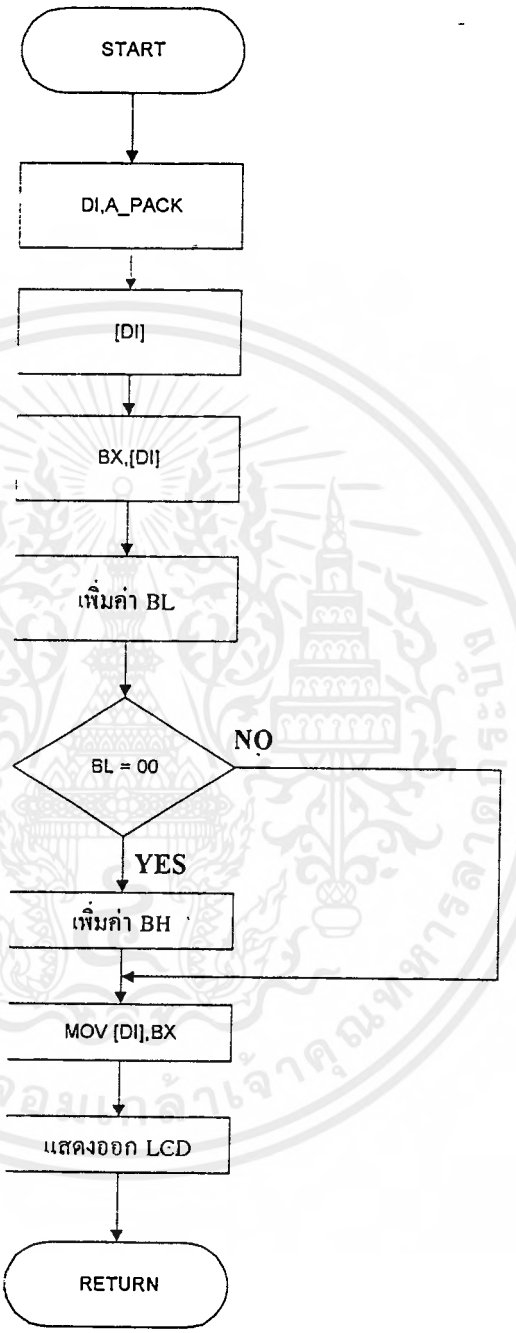
รูปที่ ข.5 (ต่อ) ผังการทำงาน FUNCTION READ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



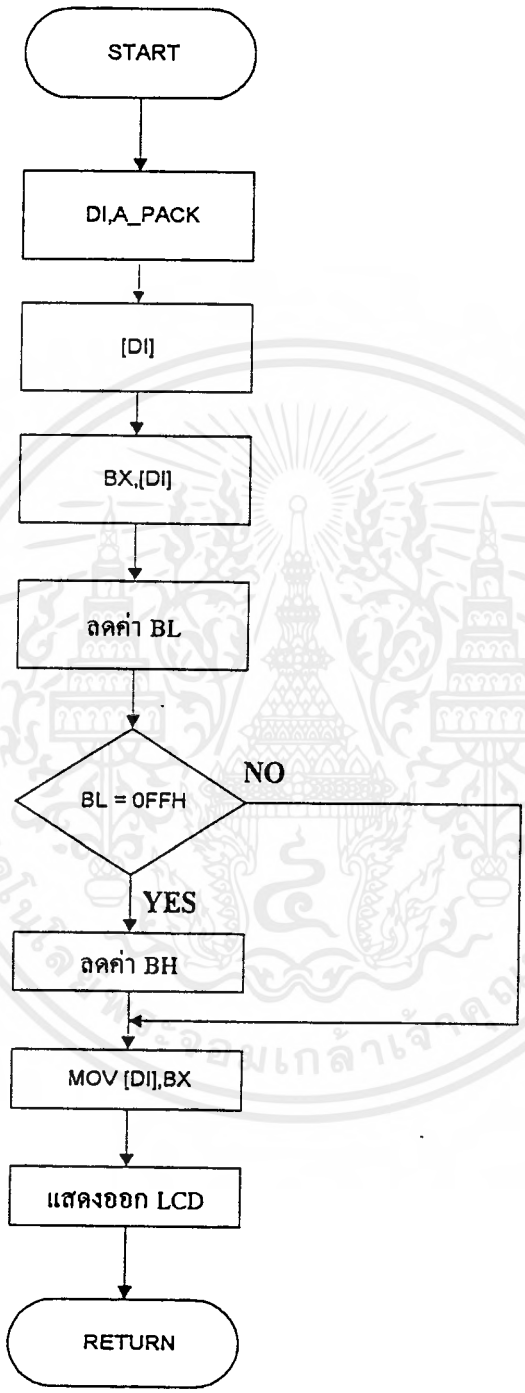
รูปที่ ข.5 (ต่อ) ผังการทำงาน FUNCTION READ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



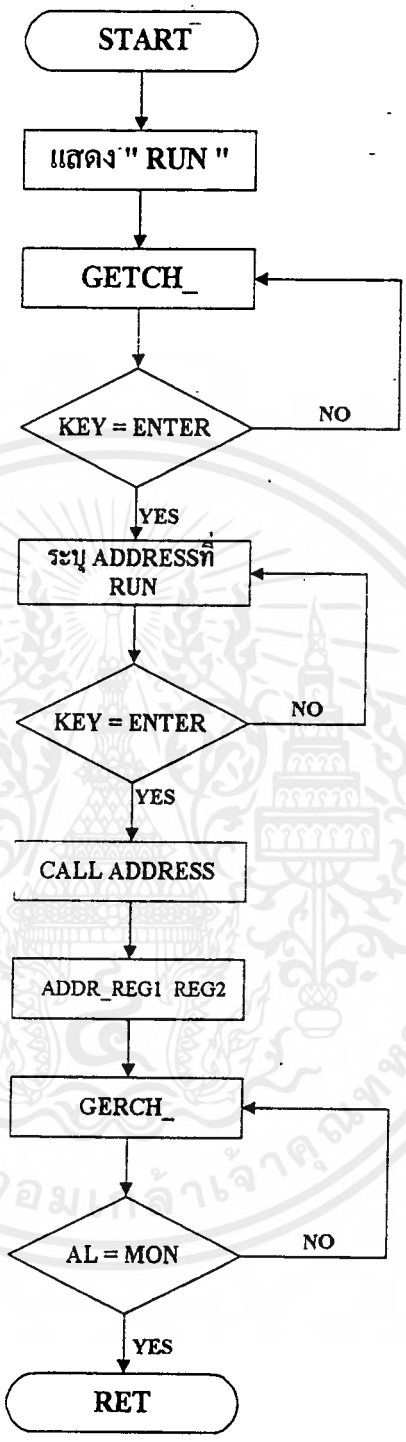
รูปที่ ข.6 ผังการทำงาน FUNCTION K_INC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



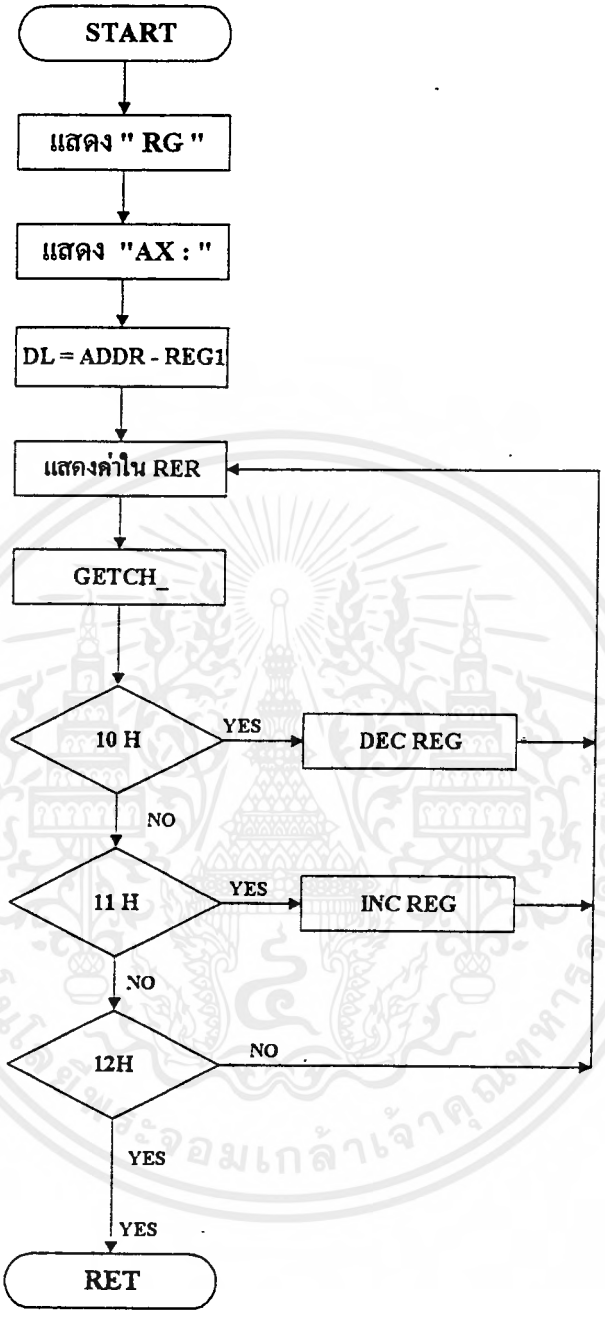
รูปที่ ข.7 ผังการทำงาน FUNCTION K_DEC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.8 ผังการทำงาน FUNCTION RUN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.9 ผังการทำงาน FUNCTION K_REG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมมอนิเตอร์

```
                        .MODEL SMALL
                        .CODE
= 0050      P_A         EQU   50H
= 0051      P_B         EQU   51H
= 0052      P_C         EQU   52H
= 0053      P_CON       EQU   53H
= 0060      WR_INST     EQU   60H
= 0062      WR_DATA     EQU   62H ;
= 0061      RD_BUSY     EQU   61H ;
= 0063      RD_DATA     EQU   63H
= 0010      P_10        EQU   10H
= 1E90      ADDR1_BUF   EQU   1E90H
= 1E92      ADDR2_BUF   EQU   1E92H
= 1E96      A_PACK      EQU   1E96H
= 1E94      DATA1_BUF   EQU   1E94H
= 1E98      D_PACK      EQU   1E98H
= 0400      ADR_USER    EQU   400H
= 1CA0      ADDR_REG1   EQU   1CA0H
= 1CA8      ADDR_REG2   EQU   1CA8H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

0000                    START    PROC   NEAR
0000  B8 1FFF                    MOV    AX,1FFFH
0003  8B E0                      MOV    SP,AX
0005  E8 02FB R                  CALL   INITIAL
0008  E8 031E R         RUB:     CALL   GETCH_
000B  3C 09                      CMP    AL,9
000D  7E F9                      JLE    RUB
000F  E8 0012 R                  CALL   KEY_FUNC
0012                             START  ENDP

                        ;****************************************
                        ;       KEY FUNCTION
                        ;****************************************

0012                    KEY_FUNC PROC   NEAR
0012  3C 13                      CMP    AL,13H
0014  74 15                      JZ     KEY_REG
0016  3C 14                      CMP    AL,14H
0018  74 17                      JZ     KEY_RUN
001A  3C 10                      CMP    AL,10H
001C  74 19                      JZ     KEY_DEC
001E  3C 11                      CMP    AL,11H
0020  74 1B                      JZ     KEY_INC
0022  3C 15                      CMP    AL,15H
0024  74 23                      JZ     KEY_RD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0026 3C 16                      CMP   AL,16H
0028 74 19                      JZ    KEY_WR
002A C3                         RET
002B E8 004D R          KEY_REG:  CALL  K_REG
002E EB 1C 90                   JMP   SO1
0031 E8 011D R          KEY_RUN:  CALL  K_RUN
0034 EB 16 90                   JMP   SO1
0037 E8 00FD R          KEY_DEC:  CALL  K_DEC
003A EB 10 90                   JMP   SO1
003D E8 0168 R          KEY_INC:  CALL  K_INC
0040 EB 0A 90                   JMP   SO1
0043 E8 019A R          KEY_WR:   CALL  K_WR
0046 EB 04 90                   JMP   SO1
0049 E8 0252 R          KEY_RD:   CALL  K_RD
004C C3                 SO1:      RET
004D                    KEY_FUNC  ENDP

                        ;*********************************
                        ; FUNCTION KEY DISPLAY REGISTER
                        ;*********************************

004D                    K_REG     PROC  NEAR
004D E8 00DE R                    CALL  T_D_REG1
0050 B0 41                        MOV   AL,41H    ; A
0052 E8 03D0 R                    CALL  WR_ASCII
0055 B0 58                        MOV   AL,58H    ; X
0057 E8 03D0 R                    CALL  WR_ASCII
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
005A  B0 85                       MOV   AL,85H
005C  E8 03CA R                   CALL  LCD_COM
005F  B0 3A                       MOV   AL,3AH    ; :
0061  E8 03D0 R                   CALL  WR_ASCII
0064  BF 1CA0          REG_N:    MOV   DI,ADDR_REG1 ; ADDR 1CA0H
0067  B6 41                       MOV   DH,41H
0069  B0 82                       MOV   AL,82H
006B  E8 03CA R                   CALL  LCD_COM
006E  8A C6                       MOV   AL,DH
0070  E8 03D0 R                   CALL  WR_ASCII
0073  E8 00B2 R        RUB_KREG:  CALL  DISP_REG
                                        ; LCD JMP---> C0H
0076  E8 031E R                   CALL  GETCH_
0079  3C 10                       CMP   AL,10H
007B  74 1F                       JZ    K_REG_DEC
007D  3C 11                       CMP   AL,11H
007F  74 05                       JZ    K_REG_INC
0081  3C 12                       CMP   AL,12H
0083  75 EE                       JNZ   RUB_KREG
0085  C3                          RET
0086  47                K_REG_INC: INC   DI
0087  47                          INC   DI
0088  81 FF 1CA8                  CMP   DI,ADDR_REG2 ;1CA8H
008C  7D D6                       JNL   REG_N
008E  FE C6                       INC   DH
0090  B0 82                       MOV   AL,82H
0092  E8 03CA R                   CALL  LCD_COM
0095  8A C6                       MOV   AL,DH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0097  E8 03D0 R                    CALL   WR_ASCII
009A  EB D7                        JMP    RUB_KREG
009C                         K_REG_DEC:
009C  4F                           DEC    DI
009D  4F                           DEC    DI
009E  81 FF 1CA0                   CMP    DI,ADDR_REG1 ;1CA0H
00A2  7C C0                        JL     REG_N        ;>
00A4  FE CE                        DEC    DH
00A6  B0 82                        MOV    AL,82H
00A8  E8 03CA R                    CALL   LCD_COM
00AB  8A C6                        MOV    AL,DH
00AD  E8 03D0 R                    CALL   WR_ASCII
00B0  EB C1                        JMP    RUB_KREG
00B2                         K_REG         ENDP

                             ;*********************************
                             ;     DISPLAY VALUE REGISTER
                             ;*********************************

00B2                         DISP_REG   PROC   NEAR    ; RUN VALUE DI
00B2  B0 C0                        MOV    AL,0C0H
00B4  E8 03CA R                    CALL   LCD_COM
00B7  8B 1D                        MOV    BX,[DI]
00B9  53                           PUSH   BX
00BA  8A C7                        MOV    AL,BH
00BC  E8 00C6 R                    CALL   LCD_REG
00BF  5B                           POP    BX
00C0  8A C3                        MOV    AL,BL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
00C2  E8 00C6 R                    CALL   LCD_REG
00C5  C3                           RET
00C6                       DISP_REG    ENDP

00C6                       LCD_REG   PROC   NEAR     ;WANT AL FOR FN
                                     YAG_DATA
00C6  51                           PUSH   CX
00C7  E8 02DE R                    CALL   YAG_DATA   ;---> AL ---> A
                                     X: AH=0_ , AL=0_
00CA  8B C8                        MOV    CX,AX
00CC  8A C1                        MOV    AL,CL
00CE  E8 03E2 R                    CALL   HEX_ASCII
00D1  E8 03D0 R                    CALL   WR_ASCII
00D4  8A C5                        MOV    AL,CH
00D6  E8 03E2 R                    CALL   HEX_ASCII
00D9  E8 03D0 R                    CALL   WR_ASCII
00DC  59                           POP    CX
00DD  C3                           RET
00DE                       LCD_REG     ENDP

                           ;*********************************
                           ; TRIAM  REGISTER >     REG
                           ;*********************************
00DE                       T_D_REG1   PROC   NEAR
00DE  B0 C6                          MOV    AL,0C6H
00E0  E8 03CA R                      CALL   LCD_COM
00E3  B0 52                          MOV    AL,52H    ; R
00E5  E8 03D0 R                      CALL   WR_ASCII
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
00E8  B0 47                     MOV   AL,47H    ; G
00EA  E8 03D0 R                 CALL  WR_ASCII
00ED  B0 80                     MOV   AL,80H
00EF  E8 03CA R                 CALL  LCD_COM
00F2  B0 3E                     MOV   AL,3EH    ; >
00F4  E8 03D0 R                 CALL  WR_ASCII
00F7  B0 82                     MOV   AL,82H
00F9  E8 03CA R                 CALL  LCD_COM
00FC  C3                        RET
00FD                T_D_REG1    ENDP

                    ;++++++++++++++++++++++++++++++++++++

                    ;**********************************
                    ;       FUNCTION   DEC
                    ;**********************************

00FD                K_DEC       PROC  NEAR
00FD  BF 1E96                   MOV   DI,A_PACK
0100  8B 1D                     MOV   BX,[DI]
0102  FE CB                     DEC   BL
0104  80 FB FF                  CMP   BL,0FFH
0107  75 02                     JNZ   GEB_DEC     ;VALUE IN
                                                  [ BL ] ADDR LOW
0109  FE CF                     DEC   BH
010B  89 1D             GEB_DEC:MOV   [DI],BX
010D  B0 82                     MOV   AL,82H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
010F  E8 03CA R                    CALL   LCD_COM
0112  8A C7                       MOV    AL,BH
0114  E8 0188 R                    CALL   DISP_N_A
0117  8A C3                       MOV    AL,BL
0119  E8 0188 R                    CALL   DISP_N_A
011C  C3                           RET
011D                    K_DEC      ENDP

                        ;**********************************
                        ;       FUNCTION   RUN
                        ;**********************************

011D                    K_RUN      PROC   NEAR
011D  B0 C5                        MOV    AL,0C5H
011F  E8 03CA R                    CALL   LCD_COM
0122  B0 52                        MOV    AL,52H    ; R
0124  E8 03D0 R                    CALL   WR_ASCII
0127  B0 55                        MOV    AL,55H    ; U
0129  E8 03D0 R                    CALL   WR_ASCII
012C  B0 4E                        MOV    AL,4EH    ; N
012E  E8 03D0 R                    CALL   WR_ASCII
                        ;-------------------------
0131  B0 AA                        MOV    AL,0AAH
0133  E6 10                        OUT    10H,AL
0135  E8 031E R                UUU:   CALL   GETCH_
0138  3C 17                        CMP    AL,17H
013A  75 F9                        JNZ    UUU
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
013C  B0 82                    MOV   AL,82H
013E  E8 03CA R                CALL  LCD_COM
0141  E8 028C R                CALL  ADJ_ADDR
0144  BF 1E90                  MOV   DI,ADDR1_BUF ;1E90H
0147  8B 75 06                 MOV   SI,[DI+6]
014A  8B DE                    MOV   BX,SI   ;BX = ADDR WILL RUN
014C  FF D3                    CALL  BX
014E  BF 1CA0          AFTER_R: MOV   DI,ADDR_REG1
0151  89 05                    MOV   [DI+0],AX
0153  89 5D 02                 MOV   [DI+2],BX
0156  89 4D 04                 MOV   [DI+4],CX
0159  89 55 06                 MOV   [DI+6],DX
015C  B0 BB                    MOV   AL,0BBH
015E  E6 10                    OUT   10H,AL
0160  E8 031E R        UUU1:   CALL  GETCH_
0163  3C 12                    CMP   AL,12H
0165  75 F9                    JNZ   UUU1
0167  C3                      RET
0168                   K_RUN   ENDP

                       ;*************************************
                       ;       FUNCTION INCREMENT
                       ;*************************************

0168                   K_INC   PROC  NEAR
0168  BF 1E96                  MOV   DI,A_PACK   ;1E96H
016B  8B 1D                    MOV   BX,[DI]
016D  FE C3                    INC   BL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
016F  80 FB 00                    CMP   BL,00H
0172  75 02                       JNZ   GEB_INC     ;VALUE IN
                                                    [ BL ] ADDR LOW
0174  FE C7                       INC   BH
0176  89 1D             GEB_INC:  MOV   [DI],BX
0178  B0 82                       MOV   AL,82H
017A  E8 03CA R                   CALL  LCD_COM
017D  8A C7                       MOV   AL,BH
017F  E8 0188 R                   CALL  DISP_N_A
0182  8A C3                       MOV   AL,BL
0184  E8 0188 R                   CALL  DISP_N_A
0187  C3                         RET
0188                    K_INC     ENDP

                        ;************************************
                        ; DISP ADDRESS AT PRESS INC OR DEC
                        ;************************************

0188                    DISP_N_A  PROC  NEAR
0188  E8 02DE R                   CALL  YAG_DATA
018B  E8 03E2 R                   CALL  HEX_ASCII ; AL 4 BIT
                                                    LOW = 0_H
018E  E8 03D0 R                   CALL  WR_ASCII
0191  8A C4                       MOV   AL,AH   ;   AH 4 BIT
                                                    HIGH = 0_H
0193  E8 03E2 R                   CALL  HEX_ASCII
0196  E8 03D0 R                   CALL  WR_ASCII
0199  C3                         RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
019A                    DISP_N_A    ENDP

                        ;*******************************
                        ;       FUNCTION   WRITE
                        ;*******************************

019A                    K_WR    PROC   NEAR
019A                    W_VVV:
019A  B0 01                         MOV    AL,1
019C  E8 03CA R                     CALL   LCD_COM
019F  B0 3E                         MOV    AL,3EH
01A1  E8 03D0 R                     CALL   WR_ASCII
01A4  B0 C6                         MOV    AL,0C6H      ;PRINT " WR"
01A6  E8 03CA R                     CALL   LCD_COM
01A9  B0 57                         MOV    AL,57H
01AB  E8 03D0 R                     CALL   WR_ASCII
01AE  B0 52                         MOV    AL,52H
01B0  E8 03D0 R                     CALL   WR_ASCII
01B3  B0 82                         MOV    AL,82H
01B5  E8 03CA R                     CALL   LCD_COM
01B8  E8 028C R                     CALL   ADJ_ADDR
01BB  BF 1E90                       MOV    DI,ADDR1_BUF  ;1E90H
01BE  8B 75 06                      MOV    SI,[DI+6]
01C1  81 FE 1E90                    CMP    SI,ADDR1_BUF
                                    ;1E90H  COM PARE VALUE OVER
01C5  7D 03                         JNL    W_VVV1
01C7  EB 07 90                      JMP    CONTI_1
                                    ;++++
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                CMP    SI,400H
                                                ;++++
                                JL   W_VVV
01CA 81 FE 1FFF         W_VVV1: CMP    SI,1FFFH
01CE 7E CA                      JLE    W_VVV
                        ;---------------------------------------
01D0                    CONTI_1:
01D0 E8 02EB R                  CALL   TRIAM_D
01D3 BF 1E94                    MOV    DI,DATA1_BUF ;1E94H
01D6 E8 031E R          K_D2:   CALL   GETCH_
01D9 3C 10                      CMP    AL,10H
01DB 7D F9                      JNL    K_D2     ;>=
01DD 88 05                      MOV    [DI],AL
01DF E8 03E2 R                  CALL   HEX_ASCII
01E2 E8 03D0 R                  CALL   WR_ASCII
01E5 47                         INC    DI
01E6 81 FF 1E96                 CMP    DI,1E96H
01EA 75 EA                      JNZ    K_D2
01EC E8 02C7 R                  CALL   PACK_DATA
01EF BF 1E90                    MOV    DI,ADDR1_BUF  ;1E90H
01F2 8A 5D 08                   MOV    BL,[DI+8]
01F5 8B 75 06                   MOV    SI,[DI+6]
01F8 88 1C                      MOV    [SI],BL       ;AX = [ 8765H ]
                        ;---------------------------------------
01FA E8 031E R          VOL1:   CALL   GETCH_
01FD 3C 12                      CMP    AL,12H
01FF 74 14                      JZ   WR_MON
0201 3C 17                      CMP    AL,17H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0203  74 06                    JZ    WR_CON_INC
0205  3C 10                    CMP   AL,10H
0207  74 07                    JZ    WR_CON_DEC
0209  EB EF                    JMP   VOL1
020B  E8 0168 R          WR_CON_INC: CALL  K_INC
020E  EB C0                    JMP   CONTI_1
0210  E8 00FD R          WR_CON_DEC: CALL  K_DEC
0213  EB BB                    JMP   CONTI_1
0215  B0 01               WR_MON:   MOV   AL,1
0217  E8 03CA R                 CALL  LCD_COM
021A  C3                       RET
021B                     K_WR      ENDP

                         ;*******************************
                         ;       INPUT ADDR IN BUFFER
                         ;*******************************


021B                     INPUT_AD  PROC  NEAR
021B  B0 00                    MOV   AL,0
021D  E6 10                    OUT   10H,AL
021F  BF 1E90            WR_ADR:   MOV   DI,ADDR1_BUF  ;1E90H
0222  B0 00                    MOV   AL,0
                                   ;CLEAR ADDR 1E90 - 1EFFH = 00H
0224  88 05              IN_0:     MOV   [DI],AL
0226  47                       INC   DI
0227  81 FF 1EFF               CMP   DI,1EFFH
022B  75 F7                    JNZ   IN_0
022D                     WR_ADR1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
022D  BF 1E90                    MOV   DI,ADDR1_BUF   ;1E90H
0230  B0 82                      MOV   AL,82H
0232  E8 03CA R                  CALL  LCD_COM
0235  E8 031E R          WR_ADR2: CALL  GETCH_
0238  3C 17                      CMP   AL,17H      ;VALUE " ENTER "
023A  75 01                      JNZ   CONTI       ;<= 9 DI =1004?
023C  C3                        RET
023D  3C 10               CONTI: CMP   AL,10H
023F  7D F4                      JNL   WR_ADR2 ; >=
0241  88 05                      MOV   [DI],AL
0243  E8 03E2 R                  CALL  HEX_ASCII
0246  E8 03D0 R                  CALL  WR_ASCII
0249  47                        INC   DI
024A  81 FF 1E94                 CMP   DI,DATA1_BUF  ;1E94H
024E  75 E5                      JNZ   WR_ADR2
0250  EB DB                      JMP   WR_ADR1
0252                      INPUT_AD   ENDP

                          ;*********************************
                          ;       FUNCTION  READ
                          ;*********************************


0252                      K_RD     PROC  NEAR
0252  B0 C6                        MOV   AL,0C6H   ; PRINT " RD "
0254  E8 03CA R                    CALL  LCD_COM
0257  B0 52                      MOV   AL,52H
0259  E8 03D0 R                  CALL  WR_ASCII
025C  B0 44                      MOV   AL,44H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
025E  E8 03D0 R                    CALL   WR_ASCII
0261  B0 82                       MOV    AL,82H
0263  E8 03CA R                    CALL   LCD_COM
0266  E8 028C R                    CALL   ADJ_ADDR
0269  E8 0293 R                    CALL   RD_DISP
026C  E8 031E R            RD_LX:  CALL   GETCH_
026F  3C 10                        CMP    AL,10H   ;KEY DEC
0271  74 11                        JZ     K_RD_DEC
0273  3C 11                        CMP    AL,11H   ;KEY INC
0275  74 05                        JZ     K_RD_INC
0277  3C 12                        CMP    AL,12H   ;KEY MON
0279  75 F1                        JNZ    RD_LX
027B  C3                           RET
027C  E8 0168 R           K_RD_INC: CALL   K_INC
027F  E8 0293 R                    CALL   RD_DISP
0282  EB E8                        JMP    RD_LX
0284  E8 00FD R           K_RD_DEC: CALL   K_DEC
0287  E8 0293 R                    CALL   RD_DISP
028A  EB E0                        JMP    RD_LX
028C                       K_RD    ENDP
                           ;*********************************
028C                       ADJ_ADDR PROC  NEAR
028C  E8 021B R                    CALL   INPUT_AD
028F  E8 02B2 R                    CALL   PACK_ADDR
0292  C3                           RET
0293                       ADJ_ADDR ENDP

                           ;*********************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0293                    RD_DISP   PROC   NEAR
0293  E8 02EB R                    CALL   TRIAM_D
0296  BF 1E90                     MOV    DI,ADDR1_BUF ;1E90H
0299  8B 75 06                    MOV    SI,[DI+6]
029C  8A 1C                      MOV    BL,[SI]
029E  8A C3                      MOV    AL,BL
02A0  E8 02DE R                  CALL   YAG_DATA
02A3  E8 03E2 R                  CALL   HEX_ASCII
                                                ;AL 4 BIT LOW  = 0_ H
02A6  E8 03D0 R                  CALL   WR_ASCII
02A9  8A C4                      MOV    AL,AH
                                                ;AH 4 BIT HIGH = 0_ H
02AB  E8 03E2 R                  CALL   HEX_ASCII
02AE  E8 03D0 R                  CALL   WR_ASCII
02B1  C3                         RET
02B2                             RD_DISP   ENDP

                        ;*********************************
                        ;       PACK   ADDRESS
                        ;*********************************

02B2                    PACK_ADDR  PROC   NEAR
02B2  50                         PUSH   AX
02B3  BF 1E90                    MOV    DI,ADDR1_BUF ;1E90H
02B6  E8 02D1 R                  CALL   SUM_
02B9  88 45 07                   MOV    [DI+7],AL
02BC  BF 1E92                    MOV    DI,ADDR2_BUF ;1E92H
02BF  E8 02D1 R                  CALL   SUM_
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
02C2  88 45 04                    MOV   [DI+4],AL
02C5  58                          POP   AX
02C6  C3                          RET
02C7                        PACK_ADDR  ENDP

                            ;*******************************
                            ;       PACK   DATA
                            ;*******************************

02C7                        PACK_DATA  PROC   NEAR
02C7  BF 1E94                     MOV   DI,DATA1_BUF ;1E94H
02CA  E8 02D1 R                   CALL  SUM_
02CD  88 45 04                    MOV   [DI+4],AL
02D0  C3                          RET
02D1                        PACK_DATA  ENDP

                            ;*******************************
                            ;       FN  SUM ADDR OR DATA
                            ;*******************************

02D1                        SUM_     PROC   NEAR
02D1  53                          PUSH  BX
02D2  8B 1D                       MOV   BX,[DI]  ;BH = 02 BL=01
02D4  8A C3                       MOV   AL,BL
02D6  B1 04                       MOV   CL,4
02D8  D2 C8                       ROR   AL,CL
02DA  0A C7                       OR    AL,BH
02DC  5B                          POP   BX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
02DD C3                         RET
02DE                SUM_        ENDP

                    ;*********************************
                    ;       YAG   DATA
                    ;*********************************

02DE                YAG_DATA    PROC   NEAR
02DE 51                         PUSH   CX    ; ------> AL
02DF B4 00                      MOV    AH,0
02E1 B1 04                      MOV    CL,4
02E3 D3 C8                      ROR    AX,CL
02E5 B1 04                      MOV    CL,4
02E7 D2 CC                      ROR    AH,CL
                                ;DATA AT  AX -----> RET
02E9 59                         POP    CX
02EA C3                         RET
02EB                YAG_DATA    ENDP

                    ;*********************************
                    ;       TRIAM DATA :_ __
                    ;*********************************

02EB                TRIAM_D     PROC   NEAR
02EB B0 C0                      MOV    AL,0C0H
02ED E8 03CA R                  CALL   LCD_COM
02F0 B0 3A                      MOV    AL,3AH
02F2 E8 03D0 R                  CALL   WR_ASCII
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
02F5  B0 C2                         MOV   AL,0C2H
02F7  E8 03CA R                     CALL  LCD_COM
02FA  C3                            RET
02FB                    TRIAM_D     ENDP

                        ;*********************************
                        ;   INITIAL HARDWARE
                        ;*********************************

02FB                    INITIAL   PROC  NEAR
02FB  B0 88                         MOV   AL,88H      ;INITIAL 8255 KEY
02FD  E6 53                         OUT   P_CON,AL
02FF  B0 38                         MOV   AL,38H      ;INITIAL LCD
0301  E8 03CA R                     CALL  LCD_COM
0304  B0 0F                         MOV   AL,0FH
0306  E8 03CA R                     CALL  LCD_COM
0309  B0 06                         MOV   AL,6
030B  E8 03CA R                     CALL  LCD_COM
030E  B0 01                         MOV   AL,1
0310  E8 03CA R                     CALL  LCD_COM
0313  B0 3E                         MOV   AL,3EH
0315  E8 03D0 R                     CALL  WR_ASCII
0318  B0 FE                         MOV   AL,0FEH
031A  E8 03D0 R                     CALL  WR_ASCII
031D  C3                            RET
031E                    INITIAL     ENDP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;************************************
                        ;       SCAN KEY
                        ;************************************

031E                    GETCH_   PROC  NEAR
031E B2 00              NEW_COL:  MOV   DL,0H
0320 8A C2              SHIFT_COL: MOV  AL,DL        ;COLUM
0322 E6 52                        OUT   P_C,AL
0324 E4 52                        IN    AL,P_C
0326 24 F0                        AND   AL,0F0H
0328 3C F0                        CMP   AL,0F0H
032A 75 02                        JNZ   ONKEY
032C B0 00                        MOV   AL,0
032E 3C 00              ONKEY:    CMP   AL,0
0330 74 10                        JZ    SKIPP
0332 0A C2                        OR    AL,DL
0334 E8 0353 R                    CALL  NUM_KEY
                                  ;*** AL = OUTPUT VALUE KEY
0337 50                           PUSH  AX
0338 E4 52              OPEN_K:   IN    AL,P_C  ;*** CHECK OPEN KEY
033A 24 F0                        AND   AL,0F0H
033C 3C F0                        CMP   AL,0F0H
033E 75 F8                        JNZ   OPEN_K
0340 58                           POP   AX
0341 C3                           RET
0342 FE C2              SKIPP:    INC   DL
0344 75 05                        JNZ   QQ
0346 B0 82                        MOV   AL,82H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0348  E8 03CA R                    CALL   LCD_COM
034B  80 FA 06          QQ:        CMP    DL,6
034E  75 D0                        JNZ    SHIFT_COL
0350  EB CC                        JMP    NEW_COL
0352  C3                           RET
0353                    GETCH_     ENDP
                        ;*****************************************
0353                    NUM_KEY    PROC   NEAR
0353  8A E0                        MOV    AH,AL
0355  80 E4 0F                     AND    AH,0FH
                                             ; AS AH=72 NEW AH =02
0358  24 F0                        AND    AL,0F0H
                                             ; AS AL=72 NEW AH =70
035A  3C 70                        CMP    AL,70H
035C  74 0C                        JZ     ROW1
035E  3C B0                        CMP    AL,0B0H
0360  74 17                        JZ     ROW2
0362  3C D0                        CMP    AL,0D0H
0364  74 22                        JZ     ROW3
0366  3C E0                        CMP    AL,0E0H
0368  74 2D                        JZ     ROW4
036A  B0 00             ROW1:      MOV    AL,0
                                   ; ROW0 + 00  KEY 0,1,2,3,FU,DEC
036C  02 C4                        ADD    AL,AH      ; AH = 0 TO 5
036E  E6 10                        OUT    10H,AL
0370  E8 03A4 R                    CALL   DELAY
0373  E8 03FA R                    CALL   SOUND
0376  EB 2B 90                     JMP    SO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0379 B0 06          ROW2:    MOV   AL,06H
                             ; ROW1 + 06 KEY 4,5,6,7,RUN,INC
037B 02 C4                   ADD   AL,AH     ; AH = 0 TO 5
037D E6 10                   OUT   10H,AL
037F E8 03A4 R               CALL  DELAY
0382 E8 03FA R               CALL  SOUND
0385 EB 1C 90                JMP   SO
0388 B0 0C          ROW3:    MOV   AL,0CH
                             ; ROW2 + 12 KEY 8,9,A,B,DATA,STEP
038A 02 C4                   ADD   AL,AH     ; AH = 0 TO 5
038C E6 10                   OUT   10H,AL
038E E8 03A4 R               CALL  DELAY
0391 E8 03FA R               CALL  SOUND
0394 EB 0D 90                JMP   SO
0397 B0 12          ROW4:    MOV   AL,12H
                             ;ROW3 + 18  KEY C,D,E,F,REG,MON
0399 02 C4                   ADD   AL,AH
039B E6 10                   OUT   10H,AL
039D E8 03A4 R               CALL  DELAY
03A0 E8 03FA R               CALL  SOUND
03A3 C3             SO:      RET
03A4                NUM_KEY  ENDP
                    ;****************************************
03A4                DELAY    PROC  NEAR
03A4 51                      PUSH  CX
03A5 B9 01FF                 MOV   CX,1FFH
03A8 E2 FE                   LOOP  $
03AA 59                      POP   CX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
03AB  C3                        RET
03AC                    DELAY   ENDP

                        ;*********************************
                        ;       FUNC PRINTF
                        ;*********************************

03AC                    PRINTF  PROC  NEAR
03AC  50                        PUSH  AX
03AD  53                        PUSH  BX
03AE  8A D8                     MOV   BL,AL
03B0  8A C4                     MOV   AL,AH
03B2  E8 03CA R                 CALL  LCD_COM
03B5  8A C3                     MOV   AL,BL
03B7  E8 03E2 R                 CALL  HEX_ASCII
03BA  E8 03D0 R                 CALL  WR_ASCII
03BD  E8 03C2 R                 CALL  DELAYLCD
03C0  5B                        POP   BX
03C1  58                        POP   AX
03C2                    PRINTF  ENDP

                        ;*********************************
                        ;       DELAY LCD
                        ;*********************************

03C2                    DELAYLCD  PROC  NEAR
03C2  51                        PUSH  CX
03C3  B9 FFFF                   MOV   CX,0FFFFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
03C6  E2 FE          DD2:        LOOP   DD2
03C8  59                         POP    CX
03C9  C3                         RET
03CA                 DELAYLCD    ENDP

                     ;**********************************
                     ;       WRITE COMMAND LCD
                     ;**********************************

03CA                 LCD_COM     PROC   NEAR
03CA  E8 03D6 R                  CALL   BUSY
03CD  E6 60                      OUT    WR_INST,AL
03CF  C3                         RET
03D0                 LCD_COM     ENDP

                     ;**********************************
                     ;       WRITE DATA LCD
                     ;**********************************

03D0                 WR_ASCII    PROC   NEAR
03D0  E8 03D6 R                  CALL   BUSY
03D3  E6 62                      OUT    WR_DATA,AL
03D5  C3                         RET
03D6                 WR_ASCII    ENDP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;************************************
                        ;       CHECK BUSY LCD
                        ;************************************

03D6                    BUSY    PROC   NEAR
03D6  50                        PUSH   AX
03D7  E4 61             NOT_BUSY: IN   AL,RD_BUSY
03D9  A8 80                     TEST   AL,80H
03DB  75 FA                     JNZ    NOT_BUSY
03DD  E8 03A4 R                 CALL   DELAY
03E0  58                        POP    AX
03E1  C3                        RET
03E2                    BUSY    ENDP

                        ;************************************
                        ;       HEX TO ASCII
                        ;************************************

03E2                    HEX_ASCII  PROC   NEAR
03E2  3C 09                     CMP    AL,09H
03E4  7E 05                     JLE    N_09
03E6  04 37                     ADD    AL,37H
03E8  EB 03 90                  JMP    G_RET
03EB  04 30             N_09:   ADD    AL,30H
03ED  C3                G_RET:  RET
03EE                    HEX_ASCII  ENDP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;***********************************
                        ;       ASCII TO HEX
                        ;***********************************

03EE                    ASCII_HEX  PROC   NEAR
03EE  3C 39                        CMP    AL,39H
03F0  7E 05                        JLE    N30_39          ;<=
03F2  2C 37                        SUB    AL,37H
03F4  EB 03 90                     JMP    G_RET_AH
03F7  2C 30             N30_39:    SUB    AL,30H
03F9  C3                G_RET_AH:  RET
03FA                    ASCII_HEX  ENDP

                        ;*********************************
                        ;     FUNCTION SOUND
                        ;*********************************

03FA                    SOUND      PROC   NEAR
03FA  51                           PUSH   CX
03FB  50                           PUSH   AX
03FC  B9 0030                      MOV    CX,030H ;25
03FF  B0 A0             LLS:       MOV    AL,0A0H ;50
0401  E6 51                        OUT    P_B,AL
0403  E8 041F R                    CALL   DELAYS1
0406  B0 20                        MOV    AL,20H
0408  E6 51                        OUT    P_B,AL
040A  E8 041F R                    CALL   DELAYS1
040D  49                           DEC    CX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
040E 75 EF                      JNZ   LLS
0410 B0 00                      MOV   AL,0
0412 E6 51                      OUT   P_B,AL
0414 58                         POP AX
0415 59                         POP CX
0416 C3                         RET
0417                SOUND       ENDP

                    ;*******************************
                    ;   FUNCTION DELAY SOUND
                    ;*******************************

0417                DELAYS  PROC  NEAR
0417 51                         PUSH  CX
0418 B9 01FF                    MOV   CX,1FFH
041B E2 FE                      LOOP  $
041D 59                         POP   CX
041E C3                         RET
041F                DELAYS      ENDP
041F                DELAYS1     PROC  NEAR
041F 51                         PUSH  CX
0420 B9 00FF                    MOV   CX,0FFH
0423 E2 FE                      LOOP  $
0425 59                         POP   CX
0426 C3                         RET
0427                DELAYS1     ENDP

                    ;*******************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
7FF0                    ORG   7FF0H
7FF0 E9 0000 R          JMP   START
                        END   START
```

Segments and Groups:

Name	Length	Align	Combine	Class
DGROUP		GROUP		
_DATA	0000	WORD	PUBLIC	'DATA'
_TEXT	7FF3	WORD	PUBLIC	'CODE'

Symbols:

Name	Type	Value	Attr
ADDR1_BUF	NUMBER	1E90	
ADDR2_BUF	NUMBER	1E92	
ADDR_REG1	NUMBER	1CA0	
ADDR_REG2	NUMBER	1CA8	
ADJ_ADDR	N PROC	028C	_TEXT Length = 0007
ADR_USER	NUMBER	0400	
AFTER_R	L NEAR	014E	_TEXT
ASCII_HEX	N PROC	03EE	_TEXT Length = 000C
A_PACK	NUMBER	1E96	
BUSY	N PROC	03D6	_TEXT Length = 000C
CONTI	L NEAR	023D	_TEXT
CONTI_1	L NEAR	01D0	_TEXT
DATA1_BUF	NUMBER	1E94	
DD2	L NEAR	03C6	_TEXT
DELAY	N PROC	03A4	_TEXT Length = 0008
DELAYLCD	N PROC	03C2	_TEXT Length = 0008
DELAYS	N PROC	0417	_TEXT Length = 0008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DELAYS1	N PROC	041F	_TEXT Length = 0008
DISP_N_A	N PROC	0188	_TEXT Length = 0012
DISP_REG	N PROC	00B2	_TEXT Length = 0014
D_PACK	NUMBER	1E98	
GEB_DEC	L NEAR	010B	_TEXT
GEB_INC	L NEAR	0176	_TEXT
GETCH_	N PROC	031E	_TEXT Length = 0035
G_RET	L NEAR	03ED	_TEXT
G_RET_AH	L NEAR	03F9	_TEXT
HEX_ASCII	N PROC	03E2	_TEXT Length = 000C
INITIAL	N PROC	02FB	_TEXT Length = 0023
INPUT_AD	N PROC	021B	_TEXT Length = 0037
IN_0	L NEAR	0224	_TEXT
KEY_DEC	L NEAR	0037	_TEXT
KEY_FUNC	N PROC	0012	_TEXT Length = 003B
KEY_INC	L NEAR	003D	_TEXT
KEY_RD	L NEAR	0049	_TEXT
KEY_REG	L NEAR	002B	_TEXT
KEY_RUN	L NEAR	0031	_TEXT
KEY_WR	L NEAR	0043	_TEXT
K_D2	L NEAR	01D6	_TEXT
K_DEC	N PROC	00FD	_TEXT Length = 0020
K_INC	N PROC	0168	_TEXT Length = 0020
K_RD	N PROC	0252	_TEXT Length = 003A
K_RD_DEC	L NEAR	0284	_TEXT
K_RD_INC	L NEAR	027C	_TEXT
K_REG	N PROC	004D	_TEXT Length = 0065
K_REG_DEC	L NEAR	009C	_TEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K_REG_INC	L NEAR	0086	_TEXT
K_RUN	N PROC	011D	_TEXT Length = 004B
K_WR	N PROC	019A	_TEXT Length = 0081
LCD_COM	N PROC	03CA	_TEXT Length = 0006
LCD_REG	N PROC	00C6	_TEXT Length = 0018
LLS	L NEAR	03FF	_TEXT
N30_39	L NEAR	03F7	_TEXT
NEW_COL	L NEAR	031E	_TEXT
NOT_BUSY	L NEAR	03D7	_TEXT
NUM_KEY	N PROC	0353	_TEXT Length = 0051
N_09	L NEAR	03EB	_TEXT
ONKEY	L NEAR	032E	_TEXT
OPEN_K	L NEAR	0338	_TEXT
PACK_ADDR	N PROC	02B2	_TEXT Length = 0015
PACK_DATA	N PROC	02C7	_TEXT Length = 000A
PRINTF	N PROC	03AC	_TEXT Length = 0016
P_10	NUMBER	0010	
P_A	NUMBER	0050	
P_B	NUMBER	0051	
P_C	NUMBER	0052	
P_CON	NUMBER	0053	
QQ	L NEAR	034B	_TEXT
RD_BUSY	NUMBER	0061	
RD_DATA	NUMBER	0063	
RD_DISP	N PROC	0293	_TEXT Length = 001F
RD_LX	L NEAR	026C	_TEXT
REG_N	L NEAR	0064	_TEXT
ROW1	L NEAR	036A	_TEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROW2	L NEAR	0379	_TEXT
ROW3	L NEAR	0388	_TEXT
ROW4	L NEAR	0397	_TEXT
RUB	L NEAR	0008	_TEXT
RUB_KREG	L NEAR	0073	_TEXT
SHIFT_COL	L NEAR	0320	_TEXT
SKIPP	L NEAR	0342	_TEXT
SO	L NEAR	03A3	_TEXT
SO1	L NEAR	004C	_TEXT
SOUND	N PROC	03FA	_TEXT Length = 001D
START	N PROC	0000	_TEXT Length = 0012
SUM_	N PROC	02D1	_TEXT Length = 000D
TRIAM_D	N PROC	02EB	_TEXT Length = 0010
T_D_REG1	N PROC	00DE	_TEXT Length = 001F
UUU	L NEAR	0135	_TEXT
UUU1	L NEAR	0160	_TEXT
VOL1	L NEAR	01FA	_TEXT
WR_ADR	L NEAR	021F	_TEXT
WR_ADR1	L NEAR	022D	_TEXT
WR_ADR2	L NEAR	0235	_TEXT
WR_ASCII	N PROC	03D0	_TEXT Length = 0006
WR_CON_DEC	L NEAR	0210	_TEXT
WR_CON_INC	L NEAR	020B	_TEXT
WR_DATA	NUMBER	0062	
WR_INST	NUMBER	0060	
WR_MON	L NEAR	0215	_TEXT
W_VVV	L NEAR	019A	_TEXT
W_VVV1	L NEAR	01CA	_TEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
YAG_DATA . . . . . .  N PROC      02DE  _TEXT Length = 000D
@CODE . . . . . .     TEXT _TEXT
@CODESIZE . . . . . .  TEXT 0
@CPU . . . . . .      TEXT 0101h
@DATASIZE . . . . . .  TEXT 0
@FILENAME . . . . . .  TEXT LK
@VERSION . . . . . .  TEXT 510

  780 Source  Lines
  780 Total   Lines
  118 Symbols
47590 + 422980 Bytes symbol space free
    0 Warning Errors
    0 Severe  Errors
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
คำสั่งที่ใช้ในการสั่งงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่ใช้ในการสั่งงาน

ตารางที่ ค.1 ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
13 C3	ADC	AX,BX
13 C1	ADC	AX,CX
13 C2	ADC	AX,DX
13 D8	ADC	BX,AX
13 D9	ADC	BX,CX
13 DA	ADC	BX,DX
13 C8	ADC	CX,AX
13 CB	ADC	CX,BX
13 CA	ADC	CX,DX
13 D0	ADC	DX,AX
13 D3	ADC	X,BX
13 D1	ADC	DX,CX
12 E7	ADC	AH,BH
12 E5	ADC	AH,CH
12 E6	ADC	AH,DH
12 E3	ADC	AH,BL
12 E1	ADC	AH,CL
12 E2	ADC	AH,DL
12 C3	ADC	AL,BL
12 C1	ADC	AL,CL
12 C2	ADC	AL,DL
12 C7	ADC	AL,BH

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
12 C5	ADC	AL,CH
12 C6	ADC	AL,DH
12 FC	ADC	BH,AH
12 FD	ADC	BH,CH
12 FE	ADC	BH,DH
12 F8	ADC	BH,AL
12 F9	ADC	BH,CL
12 FA	ADC	BH,DL
12 EC	ADC	CH,AH
12 EF	ADC	CH,BH
12 EE	ADC	CH,DH
12 E8	ADC	CH,AL
12 EB	ADC	CH,BL
12 EA	ADC	CH,DL
12 C8	ADC	CL,AL
12 CB	ADC	CL,BL
12 CA	ADC	CL,DL
12 CC	ADC	CL,AH
12 CF	ADC	CL,BH
12 CE	ADC	CL,DH
12 F4	ADC	DH,AH
12 F7	ADC	DH,BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
12 F5	ADC	DH,CH
12 F0	ADC	DH,AL
12 F3	ADC	DH,BL
12 F1	ADC	DH,CL
12 D4	ADC	DL,AH
12 D7	ADC	DL,BH
12 D5	ADC	DL,CH
12 D0	ADC	DL,AL
12 D3	ADC	DL,BL
12 D1	ADC	DL,CL
13 C7	ADC	AX,DI
13 05	ADC	AX,[DI]
13 45 01	ADC	AX,[DI+1]
13 C6	ADC	AX,SI
13 04	ADC	AX,[SI]
13 44 01	ADC	AX,[SI+1]
13 C5	ADC	AX,BP
13 DF	ADC	BX,DI
13 1D	ADC	BX,[DI]
13 5D 01	ADC	BX,[DI+1]
13 DE	ADC	BX,SI
13 1C	ADC	BX,[SI]
13 5C 01	ADC	BX,[SI+1]
13 DD	ADC	BX,BP

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
13 0D	ADC	CX,[DI]
13 4D 01	ADC	CX,[DI+1]
13 CE	ADC	CX,SI
13 0C	ADC	CX,[SI]
13 4C 01	ADC	CX,[SI+1]
13 CD	ADC	CX,BP
13 D7	ADC	DX,DI
13 15	ADC	DX,[DI]
13 55 01	ADC	DX,[DI+1]
13 D6	ADC	DX,SI
13 14	ADC	DX,[SI]
13 54 01	ADC	DX,[SI+1]
13 D5	ADC	DX,BP
11 44 01	ADC	[SI+1],AX
11 5C 01	ADC	[SI+1],BX
11 4C 01	ADC	[SI+1],CX
11 54 01	ADC	[SI+1],DX
11 04	ADC	[SI],AX
11 1C	ADC	[SI],BX
11 0C	ADC	[SI],CX
11 14	ADC	[SI],DX
11 46 01	ADC	[BP+1],AX
11 5E 01	ADC	[BP+1],BX
11 4E 01	ADC	[BP+1],CX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
13 E8	ADC	BP,AX
13 EB	ADC	BP,BX
13 E9	ADC	BP,CX
13 EA	ADC	BP,DX
11 45 01	ADC	DI+1],AX
11 5D 01	ADC	[DI+1],BX
11 4D 01	ADC	[DI+1],CX
11 55 01	ADC	[DI+1],DX
11 05	ADC	[DI],AX
11 1D	ADC	[DI],BX
11 0D	ADC	[DI],CX
11 15	ADC	[DI],DX
15 0B 10	ADC	AX,100BH
81 D3 0A 10	ADC	BX,100AH
81 D1 0A 10	ADC	CX,100AH
81 D2 0A 10	ADC	DX,100AH
80 D4 0B	ADC	AH,0BH
14 0B	ADC	AL,0BH
80 D7 0A	ADC	BH,0AH
80 D5 0A	ADC	CH,0AH
80 D1 0A	ADC	CL,0AH
80 D6 0A	ADC	DH,0AH
80 D2 0A	ADC	DL,0AH
03 C3	ADD	AX,BX

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
03 C2	ADD	AX,DX
03 D8	ADD	BX,AX
03 D9	ADD	BX,CX
03 DA	ADD	BX,DX
03 C8	ADD	CX,AX
03 CB	ADD	CX,BX
03 CA	ADD	CX,DX
03 D0	ADD	DX,AX
03 D3	ADD	DX,BX
03 D1	ADD	DX,CX
02 E7	ADD	AH,BH
02 E5	ADD	AH,CH
02 E6	ADD	AH,DH
02 E3	ADD	AH,BL
02 E1	ADD	AH,CL
02 E2	ADD	AH,DL
02 C3	ADD	AL,BL
02 C1	ADD	AL,CL
02 C2	ADD	AL,DL
02 C7	ADD	AL,BH
02 C5	ADD	AL,CH
02 C6	ADD	AL,DH
02 FC	ADD	BH,AH
02 FD	ADD	BH,CH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
02 F8	ADD	BH,AL
02 F9	ADD	BH,CL
02 FA	ADD	BH,DL
02 EC	ADD	CH,AH
02 EF	ADD	CH,BH
02 EE	ADD	CH,DH
02 E8	ADD	CH,AL
02 EB	ADD	CH,BL
02 EA	ADD	CH,DL
02 C8	ADD	CL,AL
02 CB	ADD	CL,BL
02 CA	ADD	CL,DL
02 CC	ADD	CL,AH
02 CF	ADD	CL,BH
02 CE	ADD	CL,DH
02 F4	ADD	DH,AH
02 F7	ADD	DH,BH
02 F5	ADD	DH,CH
02 F0	ADD	DH,AL
02 F3	ADD	DH,BL
02 F1	ADD	DH,CL
02 D4	ADD	DL,AH
02 D7	ADD	DL,BH
02 D5	ADD	DL,CH

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
02 D3	ADD	DL,BL
02 D1	ADD	DL,CL
03 C7	ADD	AX,DI
03 05	ADD	AX,[DI]
03 45 01	ADD	AX,[DI+1]
03 C6	ADD	AX,SI
03 04	ADD	AX,[SI]
03 44 01	ADD	AX,[SI+1]
03 C5	ADD	AX,BP
03 DF	ADD	BX,DI
03 1D	ADD	BX,[DI]
03 5D 01	ADD	BX,[DI+1]
03 DE	ADD	BX,SI
03 1C	ADD	BX,[SI]
03 5C 01	ADD	BX,[SI+1]
03 DD	ADD	BX,BP
03 CF	ADD	CX,DI
03 0D	ADD	CX,[DI]
03 4D 01	ADD	CX,[DI+1]
03 CE	ADD	CX,SI
03 0C	ADD	CX,[SI]
03 4C 01	ADD	CX,[SI+1]
03 CD	ADD	CX,BP
03 D7	ADD	DX,DI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
03 55 01	ADD	DX,[DI+1]
03 D6	ADD	DX,SI
03 14	ADD	DX,[SI]
03 54 01	ADD	DX,[SI+1]
03 D5	ADD	DX,BP
01 44 01	ADD	[SI+1],AX
01 5C 01	ADD	[SI+1],BX
01 4C 01	ADD	[SI+1],CX
01 54 01	ADD	[SI+1],DX
01 04	ADD	[SI],AX
01 1C	ADD	[SI],BX
01 0C	ADD	[SI],CX
01 14	ADD	[SI],DX
01 46 01	ADD	[BP+1],AX
01 5E 01	ADD	[BP+1],BX
01 4E 01	ADD	[BP+1],CX
01 56 01	ADD	[BP+1],DX
03 E8	ADD	BP,AX
03 EB	ADD	BP,BX
03 E9	ADD	BP,CX
03 EA	ADD	BP,DX
01 45 01	ADD	[DI+1],AX
01 5D 01	ADD	[DI+1],BX
01 4D 01	ADD	[DI+1],CX

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
01 05	ADD	[DI],AX
01 1D	ADD	[DI],BX
01 0D	ADD	[DI],CX
01 15	ADD	[DI],DX
05 0B 10	ADD	AX,100BH
81 C3 0A 10	ADD	BX,100AH
81 C1 0A 10	ADD	CX,100AH
81 C2 0A 10	ADD	DX,100AH
80 C4 0B	ADD	AH,0BH
04 0B	ADD	AL,0BH
80 C7 0A	ADD	BH,0AH
80 C5 0A	ADD	CH,0AH
80 C1 0A	ADD	CL,0AH
80 C6 0A	ADD	DH,0AH
80 C2 0A	ADD	DL,0AH
23 C3	AND	AX,BX
23 C1	AND	AX,CX
23 C2	AND	AX,DX
23 D8	AND	BX,AX
23 D9	AND	BX,CX
23 DA	AND	BX,DX
23 C8	AND	CX,AX
23 CB	AND	CX,BX
23 CA	AND	CX,DX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
23 D3	AND	DX,BX
23 D1	AND	DX,CX
22 E7	AND	AH,BH
22 E5	AND	AH,CH
22 E6	AND	AH,DH
22 E3	AND	AH,BL
22 E1	AND	AH,CL
22 E2	AND	AH,DL
22 C3	AND	AL,BL
22 C1	AND	AL,CL
22 C2	AND	AL,DL
22 C7	AND	AL,BH
22 C5	AND	AL,CH
22 C6	AND	AL,DH
22 FC	AND	BH,AH
22 FD	AND	BH,CH
22 FE	AND	BH,DH
22 F8	AND	BH,AL
22 F9	AND	BH,CL
22 FA	AND	BH,DL
22 EC	AND	CH,AH
22 EF	AND	CH,BH
22 EE	AND	CH,DH
22 E8	AND	CH,AL

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
22 EA	AND	CH,DL
22 C8	AND	CL,AL
22 CB	AND	CL,BL
22 CA	AND	CL,DL
22 CC	AND	CL,AH
22 CF	AND	CL,BH
22 CE	AND	CL,DH
22 F4	AND	DH,AH
22 F7	AND	DH,BH
22 F5	AND	DH,CH
22 F0	AND	DH,AL
22 F3	AND	DH,BL
22 F1	AND	DH,CL
22 D4	AND	DL,AH
22 D7	AND	DL,BH
22 D5	AND	DL,CH
22 D0	AND	DL,AL
22 D3	AND	DL,BL
22 D1	AND	DL,CL
23 C7	AND	AX,DI
23 05	AND	AX,[DI]
23 45 01	AND	AX,[DI+1]
23 C6	AND	AX,SI
23 04	AND	AX,[SI]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
23 C5	AND AX,BP
23 DF	AND BX,DI
23 1D	AND BX,[DI]
23 5D 01	AND BX,[DI+1]
23 DE	AND BX,SI
23 1C	AND BX,[SI]
23 5C 01	AND BX,[SI+1]
23 DD	AND BX,BP
23 CF	AND CX,DI
23 0D	AND CX,[DI]
23 4D 01	AND CX,[DI+1]
23 CE	AND CX,SI
23 0C	AND CX,[SI]
23 4C 01	AND CX,[SI+1]
23 CD	AND CX,BP
23 D7	AND DX,DI
23 15	AND DX,[DI]
23 55 01	AND DX,[DI+1]
23 D6	AND DX,SI
23 14	AND DX,[SI]
23 54 01	AND DX,[SI+1]
23 D5	AND DX,BP
21 44 01	AND [SI+1],AX
21 5C 01	AND [SI+1],BX

ค่าออฟโค้ด	ลักษณะของคำสั่ง
21 54 01	AND [SI+1],DX
21 04	AND [SI],AX
21 1C	AND [SI],BX
21 0C	AND [SI],CX
21 14	AND [SI],DX
21 46 01	AND [BP+1],AX
21 5E 01	AND [BP+1],BX
21 4E 01	AND [BP+1],CX
21 56 01	AND [BP+1],DX
23 E8	AND BP,AX
23 EB	AND BP,BX
23 E9	AND BP,CX
23 EA	AND BP,DX
21 45 01	AND [DI+1],AX
21 5D 01	AND [DI+1],BX
21 4D 01	AND [DI+1],CX
21 55 01	AND [DI+1],DX
21 05	AND [DI],AX
21 1D	AND [DI],BX
21 0D	AND [DI],CX
21 15	AND [DI],DX
25 0B 10	AND AX,100BH
81 E3 0A 10	AND BX,100AH
81 E1 0A 10	AND CX,100AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
80 E4 0B	AND	AH,0BH
80 24 0B	AND	AL,0BH
80 E7 0A	AND	BH,0AH
80 E5 0A	AND	CH,0AH
80 E1 0A	AND	CL,0AH
80 E6 0A	AND	DH,0AH
80 E2 0A	AND	DL,0AH
3BC3	CMP	AX,BX
3BC1	CMP	AX,CX
3BC2	CMP	AX,DX
3BD8	CMP	BX,AX
3BD9	CMP	BX,CX
3BDA	CMP	BX,DX
3BC8	CMP	CX,AX
3BCB	CMP	CX,BX
3BCA	CMP	CX,DX
3BD0	CMP	DX,AX
3BD3	CMP	DX,BX
3BD1	CMP	DX,CX
3AE7	CMP	AH,BH
3AE5	CMP	AH,CH
3AE6	CMP	AH,DH
3AE3	CMP	AH,BL
3AE1	CMP	AH,CL

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
3AC3	CMP	AL,BL
3AC1	CMP	AL,CL
3AC2	CMP	AL,DL
3AC7	CMP	AL,BH
3AC5	CMP	AL,CH
3AC6	CMP	AL,DH
3AFC	CMP	BH,AH
3AFD	CMP	BH,CH
3AFE	CMP	BH,DH
3AF8	CMP	BH,AL
3AF9	CMP	BH,CL
3AFA	CMP	BH,DL
3AEC	CMP	CH,AH
3AEF	CMP	CH,BH
3AEE	CMP	CH,DH
3AE8	CMP	CH,AL
3AEB	CMP	CH,BL
3AEA	CMP	CH,DL
3AC8	CMP	CL,AL
3ACB	CMP	CL,BL
3ACA	CMP	CL,DL
3ACC	CMP	CL,AH
3ACF	CMP	CL,BH
3ACE	CMP	CL,DH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
3AF7	CMP	DH,BH
3AF5	CMP	DH,CH
3AF0	CMP	DH,AL
3AF3	CMP	DH,BL
3AF1	CMP	DH,CL
3AD4	CMP	DL,AH
3AD7	CMP	DL,BH
3AD5	CMP	DL,CH
3AD0	CMP	DL,AL
3AD3	CMP	DL,BL
3AD1	CMP	DL,CL
3BC7	CMP	AX,DI
3B05	CMP	AX,[DI]
3B4501	CMP	AX,[DI+1]
3BC6	CMP	AX,SI
3B04	CMP	AX,[SI]
3B4401	CMP	AX,[SI+1]
3BC5	CMP	AX,BP
3BDF	CMP	BX,DI
3B1D	CMP	BX,[DI]
3B5D01	CMP	BX,[DI+1]
3BDE	CMP	BX,SI
3B1C	CMP	BX,[SI]
3B5C01	CMP	BX,[SI+1]

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
3BCF	CMP	CX,DI
3B0D	CMP	CX,[DI]
3B4D01	CMP	CX,[DI+1]
3BCE	CMP	CX,SI
3B0C	CMP	CX,[SI]
3B4C01	CMP	CX,[SI+1]
3BCD	CMP	CX,BP
3BD7	CMP	DX,DI
3B15	CMP	DX,[DI]
3B5501	CMP	DX,[DI+1]
3BD6	CMP	DX,SI
3B14	CMP	DX,[SI]
3B5401	CMP	DX,[SI+1]
3BD5	CMP	DX,BP
394401	CMP	[SI+1],AX
395C01	CMP	[SI+1],BX
394C01	CMP	[SI+1],CX
395401	CMP	[SI+1],DX
3904	CMP	[SI],AX
391C	CMP	[SI],BX
390C	CMP	[SI],CX
3914	CMP	[SI],DX
394601	CMP	[BP+1],AX
395E01	CMP	[BP+1],BX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
395601	CMP	[BP+1],DX
3BE8	CMP	BP,AX
3BEB	CMP	BP,BX
3BE9	CMP	BP,CX
3BEA	CMP	BP,DX
394501	CMP	[DI+1],AX
395D01	CMP	[DI+1],BX
394D01	CMP	[DI+1],CX
395501	CMP	[DI+1],DX
3905	CMP	[DI],AX
391D	CMP	[DI],BX
390D	CMP	[DI],CX
3915	CMP	[DI],DX
3D0B10	CMP	AX,100BH
81FB0A10	CMP	BX,100AH
81F90A10	CMP	CX,100AH
81FA0A10	CMP	DX,100AH
80FC0B	CMP	AH,0BH
3C0B	CMP	AL,0BH
80FF0A	CMP	BH,0AH
80FD0A	CMP	CH,0AH
80F90A	CMP	CL,0AH
80FE0A	CMP	DH,0AH
80FA0A	CMP	DL,0AH

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
4B	DEC	BX
49	DEC	CX
4A	DEC	DX
FE CC	DEC	AH
FE CF	DEC	BH
FE CD	DEC	CH
FE CE	DEC	DH
FE C8	DEC	AL
FE CB	DEC	BL
FE C9	DEC	CL
FE CA	DEC	DL
4E	DEC	SI
4D	DEC	BP
4F	DEC	DI
4C	DEC	SP
F7 F0	DIV	AX
F7 F3	DIV	BX
F7 F1	DIV	CX
F7 F2	DIV	DX
F6 F4	DIV	AH
F6 F7	DIV	BH
F6 F5	DIV	CH
F6 F6	DIV	DH
F6 F0	DIV	AL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
F6 F1	DIV	CL
F6 F2	DIV	DL
F7 F6	DIV	SI
F7 F5	DIV	BP
F7 F7	DIV	DI
F7 F4	DIV	SP
F7 F8	IDIV	AX
F7 FB	IDIV	BX
F7 F9	IDIV	CX
F7 FA	IDIV	DX
F6 FC	IDIV	AH
F6 FF	IDIV	BH
F6 FD	IDIV	CH
F6 FE	IDIV	DH
F6 F8	IDIV	AL
F6 FB	IDIV	BL
F6 F9	IDIV	CL
F6 FA	IDIV	DL
F7 FE	IDIV	SI
F7 FD	IDIV	BP
F7 FF	IDIV	DI
F7 FC	IDIV	SP
F7 E8	IMUL	AX
F7 EB	IMUL	BX

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
F7 EA	IMUL	DX
F6 EC	IMUL	AH
F6 EF	IMUL	BH
F6 ED	IMUL	CH
F6 EE	IMUL	DH
F6 E8	IMUL	AL
F6 EB	IMUL	BL
F6 E9	IMUL	CL
F6 EA	IMUL	DL
F7 EE	IMUL	SI
F7 ED	IMUL	BP
F7 EF	IMUL	DI
F7 EC	IMUL	SP
40	INC	AX
43	INC	BX
41	INC	CX
42	INC	DX
FE C4	INC	AH
FE C7	INC	BH
FE C5	INC	CH
FE C6	INC	DH
FE C0	INC	AL
FE C3	INC	BL
FE C1	INC	CL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
46	INC	SI
45	INC	BP
47	INC	DI
44	INC	SP
F7 E0	MUL	AX
F7 E3	MUL	BX
F7 E1	MUL	CX
F7 E2	MUL	DX
F6 E4	MUL	AH
F6 E7	MUL	BH
F6 E5	MUL	CH
F6 E6	MUL	DH
F6 E0	MUL	AL
F6 E3	MUL	BL
F6 E1	MUL	CL
F6 E2	MUL	DL
F7 E6	MUL	SI
F7 E5	MUL	BP
F7 E7	MUL	DI
F7 E4	MUL	SP
98	CBW	
F8	CLC	
FC	CLD	
FA	CLI	

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
F5	CMC	
99	CWD	
27	DAA	
2F	DAS	
F4	HLT	
9F	LAHF	
F0	LOCK	
90	NOP	
9C	PUSHF	
9D	POPF	
9E	SAHF	
F9	STC	
FD	STD	
FB	STI	
9B	WAIT	
E5 52	IN	AX,52H
E4 51	IN	AL,51H
ED	IN	AX,DX
E7 10	OUT	10H,AX
E6 10	OUT	10H,AL
E7 20	OUT	20H,A
E6 20	OUT	20H,AL
EF	OUT	DX,AX
E2 F0	LOOP	TT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
F7 D0	NOT	AX
F7 D3	NOT	BX
F7 D1	NOT	CX
F7 D2	NOT	DX
F6 D4	NOT	AH
F6 D7	NOT	BH
F6 D5	NOT	CH
F6 D6	NOT	DH
F6 D0	NOT	AL
F6 D3	NOT	BL
F6 D1	NOT	CL
F6 D2	NOT	DL
F7 D6	NOT	SI
F7 D7	NOT	DI
F7 D4	NOT	SP
F7 D5	NOT	BP
93	XCHG	AX,BX
91	XCHG	AX,CX
92	XCHG	AX,DX
95	XCHG	AX,BP
97	XCHG	AX,DI
96	XCHG	AX,SI
94	XCHG	AX,SP
86 E7	XCHG	AH,BH

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
86 E6	XCHG	AH,DH
86 E3	XCHG	AH,BL
86 E1	XCHG	AH,CL
86 E2	XCHG	AH,DL
86 C7	XCHG	AL,BH
86 C5	XCHG	AL,CH
86 C6	XCHG	AL,DH
86 C3	XCHG	AL,BL
86 C1	XCHG	AL,CL
86 C2	XCHG	AL,DL
87 D9	XCHG	BX,CX
87 DA	XCHG	BX,DX
86 FC	XCHG	BH,AH
86 FD	XCHG	BH,CH
86 FE	XCHG	BH,DH
86 F8	XCHG	BH,AL
86 F9	XCHG	BH,CL
86 FA	XCHG	BH,DL
86 DC	XCHG	BL,AH
86 DD	XCHG	BL,CH
86 DE	XCHG	BL,DH
86 D8	XCHG	BL,AL
86 D9	XCHG	BL,CL
86 DA	XCHG	BL,DL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
87 CB	XCHG	CX,BX
87 CA	XCHG	CX,DX
86 EC	XCHG	CH,AH
86 EF	XCHG	CH,BH
86 EE	XCHG	CH,DH
86 E8	XCHG	CH,AL
86 EB	XCHG	CH,BL
86 EA	XCHG	CH,DL
86 CC	XCHG	CL,AH
86 CF	XCHG	CL,BH
86 CE	XCHG	CL,DH
86 C8	XCHG	CL,AL
86 CB	XCHG	CL,BL
86 CA	XCHG	CL,DL
92	XCHG	DX,AX
87 D3	XCHG	DX,BX
87 D1	XCHG	DX,CX
86 F4	XCHG	DH,AH
86 F7	XCHG	DH,BH
86 F5	XCHG	DH,CH
86 F0	XCHG	DH,AL
86 F3	XCHG	DH,BL
86 F1	XCHG	DH,CL
86 F6	XCHG	DH,DH

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
86 D5	XCHG	DL,CH
86 D0	XCHG	DL,AL
86 D3	XCHG	DL,BL
86 D1	XCHG	DL,CL
96	XCHG	SI,AX
87 F3	XCHG	SI,BX
87 F1	XCHG	SI,CX
87 F2	XCHG	SI,DX
87 04	XCHG	[SI],AX
87 1C	XCHG	[SI],BX
87 0C	XCHG	[SI],CX
87 14	XCHG	[SI],DX
87 44 01	XCHG	[SI+1],AX
87 5C 01	XCHG	[SI+1],BX
87 4C 01	XCHG	[SI+1],CX
87 54 01	XCHG	[SI+1],DX
86 24	XCHG	[SI],AH
86 3C	XCHG	[SI],BH
86 2C	XCHG	[SI],CH
86 34	XCHG	[SI],DH
86 64 01	XCHG	[SI+1],AH
86 7C 01	XCHG	[SI+1],BH
86 6C 01	XCHG	SI+1],CH
86 74 01	XCHG	[SI+1],DH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
86 1C	XCHG [SI],BL
86 0C	XCHG [SI],CL
86 14	XCHG [SI],DL
86 44 01	XCHG [SI+1],AL
86 5C 01	XCHG [SI+1],BL
86 4C 01	XCHG [SI+1],CL
86 54 01	XCHG [SI+1],DL
97	XCHG DI,AX
87 FB	XCHG DI,BX
87 F9	XCHG DI,CX
87 FA	XCHG DI,DX
87 05	XCHG [DI],AX
87 1D	XCHG [DI],BX
87 0D	XCHG [DI],CX
87 15	XCHG [DI],DX
87 45 01	XCHG [DI+1],AX
87 5D 01	XCHG [DI+1],BX
87 4D 01	XCHG [DI+1],CX
87 55 01	XCHG [DI+1],DX
86 25	XCHG [DI],AH
86 3D	XCHG [DI],BH
86 2D	XCHG [DI],CH
86 35	XCHG [DI],DH
86 65 01	XCHG [DI+1],AH

ค่าออฟโค้ด	ลักษณะของคำสั่ง
86 6D 01	XCHG [DI+1],CH
86 75 01	XCHG [DI+1],DH
86 05	XCHG [DI],AL
86 1D	XCHG [DI],BL
86 0D	XCHG [DI],CL
86 15	XCHG [DI],DL
86 45 01	XCHG [DI+1],AL
86 5D 01	XCHG [DI+1],BL
86 4D 01	XCHG [DI+1],CL
86 55 01	XCHG [DI+1],DL
94	XCHG SP,AX
87 E3	XCHG SP,BX
87 E1	XCHG SP,CX
87 E2	XCHG SP,DX
95	XCHG BP,AX
87 EB	XCHG BP,BX
87 E9	XCHG BP,CX
87 EA	XCHG BP,DX
D9 D0	ESC 10,AX
D9 D3	ESC 10,BX
D9 D1	ESC 10,CX
D9 D2	ESC 10,DX
DE D4	ESC 50,AH
DE D7	ESC 50,BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
DE D6	ESC	50,DH
DE D0	ESC	50,AL
DE D3	ESC	50,BL
DE D1	ESC	50,CL
DE D2	ESC	50,DL
D9 D6	ESC	10,SI
D9 14	ESC	10,[SI]
D9 54 01	ESC	10,[SI+1]
D9 D7	ESC	10,DI
D9 15	ESC	10,[DI]
D9 55 01	ESC	10,[DI+1]
D9 D4	ESC	10,SP
D9D5	ESC	10,BP
D9 56 01	ESC	10,[BP+1]
74 EE	JZ	TT
E8 00 00	CALL	V
48	DEC	AX
7F E5	JG	TT
7F E3	JNLE	TT
7D E1	JGE	TT
7D DF	JNL	TT
7C DD	JL	TT
7C DB	JNGE	TT
70 D9	JO	TT

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
71 D5	JNO	TT
79 D3	JNS	TT
77 D1	JA	TT
77 CF	JNBE	TT
73 CD	JAE	TT
73 CB	JNB	TT
72 C9	JB	TT
72 C7	JNAE	TT
76 C5	JBE	TT
76 C3	JNA	TT
72 C1	JC	TT
74 BF	JE	TT
74 BD	JZ	TT
7A BB	JP	TT
7A B9	JPE	TT
73 B7	JNC	TT
75 B5	JNE	TT
75 B3	JNZ	TT
7B B1	JNP	TT
7B AF	JPO	TT
E3 AD	JCXZ	TT
C3	RET	
EB BF	JMP	START
BF 00 40	MOV	DI,4000H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
BD 00 20	MOV	BP,2000H
BC 00 30	MOV	SP,3000H
8B C3	MOV	AX,BX
8B C1	MOV	AX,CX
8B C2	MOV	AX,DX
8B D8	MOV	BX,AX
8B D9	MOV	BX,CX
8B DA	MOV	BX,DX
8B C8	MOV	CX,AX
8B CB	MOV	CX,BX
8B CA	MOV	CX,DX
8B D0	MOV	DX,AX
8B D3	MOV	DX,BX
8B D1	MOV	DX,CX
8A E7	MOV	AH,BH
8A E5	MOV	AH,CH
8A E6	MOV	AH,DH
8A E3	MOV	AH,BL
8A E1	MOV	AH,CL
8A E2	MOV	AH,DL
8A C3	MOV	AL,BL
8A C1	MOV	AL,CL
8A C2	MOV	AL,DL
8A C7	MOV	AL,BH

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
8A C6	MOV	AL,DH
8A FC	MOV	BH,AH
8A FD	MOV	BH,CH
8A FE	MOV	BH,DH
8A F8	MOV	BH,AL
8A F9	MOV	BH,CL
8A FA	MOV	BH,DL
8A EC	MOV	CH,AH
8A EF	MOV	CH,BH
8A EE	MOV	CH,DH
8A E8	MOV	CH,AL
8A EB	MOV	CH,BL
8A EA	MOV	CH,DL
8A C8	MOV	CL,AL
8A CB	MOV	CL,BL
8A CA	MOV	CL,DL
8A CC	MOV	CL,AH
8A CF	MOV	CL,BH
8A CE	MOV	CL,DH
8A F4	MOV	DH,AH
8A F7	MOV	DH,BH
8A F5	MOV	DH,CH
8A F0	MOV	DH,AL
8A F3	MOV	DH,BL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
8A D4	MOV DL,AH
8A D7	MOV DL,BH
8A D5	MOV DL,CH
8A D0	MOV DL,AL
8A D3	MOV DL,BL
8A D1	MOV DL,CL
8B C7	MOV AX,DI
8B 05	MOV AX,[DI]
8B 45 01	MOV AX,[DI+1]
8B C6	MOV AX,SI
8B 04	MOV AX,[SI]
8B 44 01	MOV AX,[SI+1]
8B C5	MOV AX,BP
8B DF	MOV BX,DI
8B 1D	MOV BX,[DI]
8B 5D 01	MOV BX,[DI+1]
8B DE	MOV BX,SI
8B 1C	MOV BX,[SI]
8B 5C 01	MOV BX,[SI+1]
8B DD	MOV BX,BP
8B CF	MOV CX,DI
8B 0D	MOV CX,[DI]
8B 4D 01	MOV CX,[DI+1]
8B CE	MOV CX,SI

ค่าออฟโค้ด	ลักษณะของคำสั่ง
8B 4C 01	MOV CX,[SI+1]
8B CD	MOV CX,BP
8B D7	MOV DX,DI
8B 15	MOV DX,[DI]
8B 55 01	MOV DX,[DI+1]
8B D6	MOV DX,SI
8B 14	MOV DX,[SI]
8B 54 01	MOV DX,[SI+1]
8B D5	MOV DX,BP
89 44 01	MOV [SI+1],AX
89 5C 01	MOV [SI+1],BX
89 4C 01	MOV [SI+1],CX
89 54 01	MOV [SI+1],DX
89 04	MOV [SI],AX
89 1C	MOV [SI],BX
89 0C	MOV [SI],CX
89 14	MOV [SI],DX
89 46 01	MOV [BP+1],AX
89 5E 01	MOV [BP+1],BX
89 4E 01	MOV [BP+1],CX
89 56 01	MOV [BP+1],DX
8B E8	MOV BP,AX
8B EB	MOV BP,BX
8B E9	MOV BP,CX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
9 45 01	MOV [DI+1],AX
89 5D 01	MOV [DI+1],BX
89 4D 01	MOV [DI+1],CX
89 55 01	MOV [DI+1],DX
89 05	MOV [DI],AX
89 1D	MOV [DI],BX
89 0D	MOV [DI],CX
89 15	MOV [DI],DX
B8 0B 10	MOV AX,100BH
BB 0A 10	MOV BX,100AH
B9 0A 10	MOV CX,100AH
BA 0A 10	MOV DX,100AH
B4 0B	MOV AH,0BH
B0 0B	MOV HL,0BH
B7 0A	MOV BH,0AH
B5 0A	MOV CH,0AH
B1 0A	MOV CL,0AH
B6 0A	MOV DH,0AH
B2 0A	MOV DL,0AH
0B C3	OR AX,BX
0B C1	OR AX,CX
0B C2	OR AX,DX
0B D8	OR BX,AX
0B D9	OR BX,CX

ค่าออฟโค้ด	ลักษณะของคำสั่ง
0B C8	OR CX,AX
0B CB	OR CX,BX
0B CA	OR CX,DX
0B D0	OR DX,AX
0B D3	OR DX,BX
0B D1	OR DX,CX
0A E7	OR AH,BH
0A E5	OR AH,CH
0A E6	OR AH,DH
0A E3	OR AH,BL
0A E1	OR AH,CL
0A E2	OR AH,DL
0A C3	OR AL,BL
0A C1	OR AL,CL
0A C2	OR AL,DL
0A C7	OR AL,BH
0A C5	OR AL,CH
0A C6	OR AL,DH
0A FC	OR BH,AH
0A FD	OR BH,CH
0A FE	OR BH,DH
0A F8	OR BH,AL
0A F9	OR BH,CL
0A FA	OR BH,DL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
0A EF	OR	CH,BH
0A EE	OR	CH,DH
0A E8	OR	CH,AL
0A EB	OR	CH,BL
0A EA	OR	CH,DL
0A C8	OR	CL,AL
0A CB	OR	CL,BL
0A CA	OR	CL,DL
0A CC	OR	CL,AH
0A CF	OR	CL,BH
0A CE	OR	CL,DH
0A F4	OR	DH,AH
0A F7	OR	DH,BH
0A F5	OR	DH,CH
0A F0	OR	DH,AL
0A F3	OR	DH,BL
0A F1	OR	DH,CL
0A D4	OR	DL,AH
0A D7	OR	DL,BH
0A D5	OR	DL,CH
0A D0	OR	DL,AL
0A D3	OR	DL,BL
0A D1	OR	DL,CL
0B C7	OR	AX,DI

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
0B 45 01	OR	AX,[DI+1]
0B C6	OR	AX,SI
0B 04	OR	AX,[SI]
0B 44 01	OR	AX,[SI+1]
0B C5	OR	AX,BP
0B DF	OR	BX,DI
0B 1D	OR	BX,[DI]
0B 5D 01	OR	BX,[DI+1]
0B DE	OR	BX,SI
0B 1C	OR	BX,[SI]
0B 5C 01	OR	BX,[SI+1]
0B DD	OR	BX,BP
0B CF	OR	CX,DI
0B 0D	OR	CX,[DI]
0B 4D 01	OR	CX,[DI+1]
0B CE	OR	CX,SI
0B 0C	OR	CX,[SI]
0B 4C 01	OR	CX,[SI+1]
0B CD	OR	CX,BP
0B D7	OR	DX,DI
0B 15	OR	DX,[DI]
0B 55 01	OR	DX,[DI+1]
0B D6	OR	DX,SI
0B 14	OR	DX,[SI]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
0B D5	OR DX,BP
09 44 01	OR [SI+1],AX
09 5C 01	OR [SI+1],BX
09 4C 01	OR [SI+1],CX
09 54 01	OR [SI+1],DX
09 04	OR [SI],AX
09 1C	OR [SI],BX
09 0C	OR [SI],CX
09 14	OR [SI],DX
09 46 01	OR [BP+1],AX
09 5E 01	OR [BP+1],BX
09 4E 01	OR [BP+1],CX
09 56 01	OR [BP+1],DX
0B E8	OR BP,AX
0B EB	OR BP,BX
0B E9	OR BP,CX
0B EA	OR BP,DX
09 45 01	OR [DI+1],AX
09 5D 01	OR [DI+1],BX
09 4D 01	OR [DI+1],CX
09 55 01	OR [DI+1],DX
09 05	OR [DI],AX
09 1D	OR [DI],BX
09 0D	OR [DI],CX

ค่าออฟโค้ด	ลักษณะของคำสั่ง
0D 0B 10	OR AX,100BH
81 CB 0A 10	OR BX,100AH
81 C9 0A 10	OR CX,100AH
81 CA 0A 10	OR DX,100AH
80 CC 0B	OR AH,0BH
0C 0B	OR AL,0BH
80 CF 0A	OR BH,0AH
80 CD 0A	OR CH,0AH
80 C9 0A	OR CL,0AH
80 CE 0A	OR DH,0AH
80 CA 0A	OR DL,0AH
50	PUSH AX
53	PUSH BX
51	PUSH CX
57	PUSH DI
56	PUSH SI
55	PUSH BP
D1 D0	RCL AX,1
D1 D3	RCL BX,1
D1 D1	RCL CX,1
D1 D2	RCL DX,1
D0 D4	RCL AH,1
D0 D7	RCL BH,1
D0 D5	RCL CH,1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D0 D0	RCL	AL,1
D0 D3	RCL	BL,1
D0 D1	RCL	CL,1
D0 D2	RCL	DL,1
D3 D0	RCL	AX,CL
D3 D3	RCL	BX,CL
D3 D1	RCL	CX,CL
D3 D2	RCL	DX,CL
D2 D4	RCL	AH,CL
D2 D7	RCL	BH,CL
D2 D5	RCL	CH,CL
D2 D6	RCL	DH,CL
D2 D0	RCL	AL,CL
D2 D3	RCL	BL,CL
D2 D1	RCL	CL,CL
D2 D2	RCL	DL,CL
D1 D6	RCL	SI,1
D1 D7	RCL	DI,1
D1 D5	RCL	BP,1
D3 D6	RCL	SI,CL
D3 D7	RCL	DI,CL
D3 D5	RCL	BP,CL
D1 D8	RCR	AX,1
D1 DB	RCR	BX,1

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D1 DA	RCR	DX,1
D0 DC	RCR	AH,1
D0 DF	RCR	BH,1
D0 DD	RCR	CH,1
D0 DE	RCR	DH,1
D0 D8	RCR	AL,1
D0 DB	RCR	BL,1
D0 D9	RCR	CL,1
D0 DA	RCR	DL,1
D3 D8	RCR	AX,CL
D3 DB	RCR	BX,CL
D3 D9	RCR	CX,CL
D3 DA	RCR	DX,CL
D2 DC	RCR	AH,CL
D2 DF	RCR	BH,CL
D2 DD	RCR	CH,CL
D2 DE	RCR	DH,CL
D2 D8	RCR	AL,CL
D2 DB	RCR	BL,CL
D2 D9	RCR	CL,CL
D2 DA	RCR	DL,CL
D1 DE	RCR	SI,1
D1 DF	RCR	DI,1
D1 DD	RCR	BP,1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D3 DF	RCR	DI,CL
D1 C0	ROL	AX,1
D1 C3	ROL	BX,1
D1 C1	ROL	CX,1
D1 C2	ROL	DX,1
D0 C4	ROL	AH,1
D0 C7	ROL	BH,1
D0 C5	ROL	CH,1
D0 C6	ROL	DH,1
D0 C0	ROL	AL,1
D0 C3	ROL	BL,1
D0 C1	ROL	CL,1
D0 C2	ROL	DL,1
D3 C0	ROL	AX,CL
D3 C3	ROL	BX,CL
D3 C1	ROL	CX,CL
D3 C2	ROL	DX,CL
D2 C4	ROL	AH,CL
D2 C7	ROL	BH,CL
D2 C5	ROL	CH,CL
D2 C6	ROL	DH,CL
D2 C0	ROL	AL,CL
D2 C3	ROL	BL,CL
D2 C1	ROL	CL,CL

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D1 C6	ROL	SI,1
D1 C7	ROL	DI,1
D1 C5	ROL	BP,1
D3 C6	ROL	SI,CL
D3 C7	ROL	DI,CL
D3 C5	ROL	BP,CL
D1 C8	ROR	AX,1
D1 CB	ROR	BX,1
D1 C9	ROR	CX,1
D1 CA	ROR	DX,1
D0 CC	ROR	AH,1
D0 CF	ROR	BH,1
D0 CD	ROR	CH,1
D0 CE	ROR	DH,1
D0 C8	ROR	AL,1
D0 C9	ROR	CL,1
D0 CA	ROR	DL,1
D3 C8	ROR	AX,CL
D3 CB	ROR	BX,CL
D3 C9	ROR	CX,CL
D3 CA	ROR	DX,CL
D2 CC	ROR	AH,CL
D2 CF	ROR	BH,CL
D2 CD	ROR	CH,CL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D2 C8	ROR	AL,CL
D2 CB	ROR	BL,CL
D2 C9	ROR	CL,CL
D2 CA	ROR	DL,CL
D1 CE	ROR	SI,1
D1 CF	ROR	DI,1
D1 CD	ROR	BP,1
D3 CE	ROR	SI,CL
D3 CF	ROR	DI,CL
D3 CD	ROR	BP,CL
D1 E0	SAL	AX,1
D1 E3	SAL	BX,1
D1 E1	SAL	CX,1
D1 E2	SAL	DX,1
D0 E4	SAL	AH,1
D0 E7	SAL	BH,1
D0 E5	SAL	CH,1
D0 E6	SAL	DH,1
D0 E0	SAL	AL,1
D0 E3	SAL	BL,1
D0 E1	SAL	CL,1
D0 E2	SAL	DL,1
D3 E0	SAL	AX,CL
D3 E3	SAL	BX,CL

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D3 E2	SAL	DX,CL
D2 E4	SAL	AH,CL
D2 E7	SAL	BH,CL
D2 E5	SAL	CH,CL
D2 E6	SAL	DH,CL
D2 E0	SAL	AL,CL
D2 E3	SAL	BL,CL
D2 E1	SAL	CL,CL
D2 E2	SAL	DL,CL
D1 E6	SAL	SI,1
D1 E7	SAL	DI,1
D1 E5	SAL	BP,1
D3 E6	SAL	SI,CL
D3 E7	SAL	DI,CL
D3 E5	SAL	BP,CL
D1 E0	SHL	AX,1
D1 E3	SHL	BX,1
D1 E1	SHL	CX,1
D1 E2	SHL	DX,1
D0 E4	SHL	AH,1
D0 E7	SHL	BH,1
D0 E5	SHL	CH,1
D0 E6	SHL	DH,1
D0 E0	SHL	AL,1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D0 E1	SHL	CL,1
D0 E2	SHL	DL,1
D3 E0	SHL	AX,CL
D3 E3	SHL	BX,CL
D3 E1	SHL	CX,CL
D3 E2	SHL	DX,CL
D2 E4	SHL	AH,CL
D2 E7	SHL	BH,CL
D2 E5	SHL	CH,CL
D2 E6	SHL	DH,CL
D2 E0	SHL	AL,CL
D2 E3	SHL	BL,CL
D2 E1	SHL	CL,CL
D2 E2	SHL	DL,CL
D1 E6	SHL	SI,1
D1 E7	SHL	DI,1
D1 E5	SHL	BP,1
D3 E6	SHL	SI,CL
D3 E7	SHL	DI,CL
D3 E5	SHL	BP,CL
D1 F8	SAR	AX,1
D1 FB	SAR	BX,1
D1 F9	SAR	CX,1
D1 FA	SAR	DX,1

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D0 FF	SAR	BH,1
D0 FD	SAR	CH,1
D0 FE	SAR	DH,1
D0 F8	SAR	AL,1
D0 FB	SAR	BL,1
D0 F9	SAR	CL,1
D0 FA	SAR	DL,1
D3 F8	SAR	AX,CL
D3 FB	SAR	BX,CL
D3 F9	SAR	CX,CL
D3 FA	SAR	DX,CL
D2 FC	SAR	AH,CL
D2 FF	SAR	BH,CL
D2 FD	SAR	CH,CL
D2 FE	SAR	DH,CL
D2 F8	SAR	AL,CL
D2 FB	SAR	BL,CL
D2 F9	SAR	CL,CL
D2 FA	SAR	DL,CL
D1 FE	SAR	SI,1
D1 FF	SAR	DI,1
D1 FD	SAR	BP,1
D3 FE	SAR	SI,CL
D3 FF	SAR	DI,CL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D1 E8	SHR	AX,1
D1 EB	SHR	BX,1
D1 E9	SHR	CX,1
D1 EA	SHR	DX,1
D0 EC	SHR	AH,1
D0 EF	SHR	BH,1
D0 ED	SHR	CH,1
D0 EE	SHR	DH,1
D0 E8	SHR	AL,1
D0 EB	SHR	BL,1
D0 E9	SHR	CL,1
D0 EA	SHR	DL,1
D3 E8	SHR	AX,CL
D3 EB	SHR	BX,CL
D3 E9	SHR	CX,CL
D3 EA	SHR	DX,CL
D2 EC	SHR	AH,CL
D2 EF	SHR	BH,CL
D2 ED	SHR	CH,CL
D2 EE	SHR	DH,CL
D2 E8	SHR	AL,CL
D2 EB	SHR	BL,CL
D2 E9	SHR	CL,CL
D2 EA	SHR	DL,CL

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
D1 EF	SHR	DI,1
D1 ED	SHR	BP,1
D3 EE	SHR	SI,CL
D3 EF	SHR	DI,CL
D3 ED	SHR	BP,CL
58	POP	AX
5B	POP	BX
59	POP	CX
5F	POP	DI
5E	POP	SI
5D	POP	BP
2B C3	SUB	AX,BX
2B C1	SUB	AX,CX
2B C2	SUB	AX,DX
2B D8	SUB	BX,AX
2B D9	SUB	BX,CX
2B DA	SUB	BX,DX
2B C8	SUB	CX,AX
2B CB	SUB	CX,BX
2B CA	SUB	CX,DX
2B D0	SUB	DX,AX
2B D3	SUB	DX,BX
2B D1	SUB	DX,CX
2A E7	SUB	AH,BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
2A E6	SUB	AH,DH
2A E3	SUB	AH,BL
2A E1	SUB	AH,CL
2A E2	SUB	AH,DL
2A C3	SUB	AL,BL
2A C1	SUB	AL,CL
2A C2	SUB	AL,DL
2A C7	SUB	AL,BH
2A C5	SUB	AL,CH
2A C6	SUB	AL,DH
2A FC	SUB	BH,AH
2A FD	SUB	BH,CH
2A FE	SUB	BH,DH
2A F8	SUB	BH,AL
2A F9	SUB	BH,CL
2A FA	SUB	BH,DL
2A EC	SUB	CH,AH
2A EF	SUB	CH,BH
2A EE	SUB	CH,DH
2A E8	SUB	CH,AL
2A EB	SUB	CH,BL
2A EA	SUB	CH,DL
2A C8	SUB	CL,AL
2A CB	SUB	CL,BL

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
2A CC	SUB	CL,AH
2A CF	SUB	CL,BH
2A CE	SUB	CL,DH
2A F4	SUB	DH,AH
2A F7	SUB	DH,BH
2A F5	SUB	DH,CH
2A F0	SUB	DH,AL
2A F3	SUB	DH,BL
2A F1	SUB	DH,CL
2A D4	SUB	DL,AH
2A D7	SUB	DL,BH
2A D5	SUB	DL,CH
2A D0	SUB	DL,AL
2A D3	SUB	DL,BL
2A D1	SUB	DL,CL
2B C7	SUB	AX,DI
2B 05	SUB	AX,[DI]
2B 45 01	SUB	AX,[DI+1]
2B C6	SUB	AX,SI
2B 04	SUB	AX,[SI]
2B 44 01	SUB	AX,[SI+1]
2B C5	SUB	AX,BP
2B DF	SUB	BX,DI
2B 1D	SUB	BX,[DI]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
2B DE	SUB BX,SI
2B 1C	SUB BX,[SI]
2B 5C 01	SUB BX,[SI+1]
2B DD	SUB BX,BP
2B CF	SUB CX,DI
2B 0D	SUB CX,[DI]
2B 4D 01	SUB CX,[DI+1]
2B CE	SUB CX,SI
2B 0C	SUB CX,[SI]
2B 4C 01	SUB CX,[SI+1]
2B CD	SUB CX,BP
2B D7	SUB DX,DI
2B 15	SUB DX,[DI]
2B 55 01	SUB DX,[DI+1]
2B D6	SUB DX,SI
2B 14	SUB DX,[SI]
2B 54 01	SUB DX,[SI+1]
2B D5	SUB DX,BP
29 44 01	SUB [SI+1],AX
29 5C 01	SUB [SI+1],BX
29 4C 01	SUB [SI+1],CX
29 54 01	SUB [SI+1],DX
29 04	SUB [SI],AX
29 1C	SUB [SI],BX

ค่าออฟโค้ด	ลักษณะของคำสั่ง
29 14	SUB [SI],DX
29 46 01	SUB [BP+1],AX
29 5E 01	SUB [BP+1],BX
29 4E 01	SUB [BP+1],CX
29 56 01	SUB [BP+1],DX
2B E8	SUB BP,AX
2B EB	SUB BP,BX
2B E9	SUB BP,CX
2B EA	SUB BP,DX
29 45 01	SUB [DI+1],AX
29 5D 01	SUB [DI+1],BX
29 4D 01	SUB [DI+1],CX
29 55 01	SUB [DI+1],DX
29 05	SUB [DI],AX
29 1D	SUB [DI],BX
29 0D	SUB [DI],CX
29 15	SUB [DI],DX
2D 0B 10	SUB AX,100BH
81 EB 0A 10	SUB BX,100AH
81 E9 0A 10	SUB CX,100AH
81 EA 0A 10	SUB DX,100AH
80 EC 0B	SUB AH,0BH
2C 0B	SUB AL,0BH
80 EF 0A	SUB BH,0AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
80 E9 0A	SUB	CL,0AH
80 EE 0A	SUB	DH,0AH
80 EA 0A	SUB	DL,0AH
1B C3	SBB	AX,BX
1B C1	SBB	AX,CX
1B C2	SBB	AX,DX
1B D8	SBB	BX,AX
1B D9	SBB	BX,CX
1B DA	SBB	BX,DX
1B C8	SBB	CX,AX
1B CB	SBB	CX,BX
1B CA	SBB	CX,DX
1B D0	SBB	DX,AX
1B D3	SBB	DX,BX
1B D1	SBB	DX,CX
1A E7	SBB	AH,BH
1A E5	SBB	AH,CH
1A E6	SBB	AH,DH
1A E3	SBB	AH,BL
1A E1	SBB	AH,CL
1A E2	SBB	AH,DL
1A C3	SBB	AL,BL
1A C1	SBB	AL,CL
1A C2	SBB	AL,DL

ค่าออฟโค้ด	ลักษณะของคำสั่ง	
1A C5	SBB	AL,CH
1A C6	SBB	AL,DH
1A FC	SBB	BH,AH
1A FD	SBB	BH,CH
1A FE	SBB	BH,DH
1A F8	SBB	BH,AL
1A F9	SBB	BH,CL
1A FA	SBB	BH,DL
1A EC	SBB	CH,AH
1A EF	SBB	CH,BH
1A EE	SBB	CH,DH
1A E8	SBB	CH,AL
1A EB	SBB	CH,BL
1A EA	SBB	CH,DL
1A C8	SBB	CL,AL
1A CB	SBB	CL,BL
1A CA	SBB	CL,DL
1A CC	SBB	CL,AH
1A CF	SBB	CL,BH
1A CE	SBB	CL,DH
1A F4	SBB	DH,AH
1A F7	SBB	DH,BH
1A F5	SBB	DH,CH
1A F0	SBB	DH,AL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
1A F1	SBB DH,CL
1A D4	SBB DL,AH
1A D7	SBB DL,BH
1A D5	SBB DL,CH
1A D0	SBB DL,AL
1A D3	SBB DL,BL
1A D1	SBB DL,CL
1B C7	SBB AX,DI
1B 05	SBB AX,[DI]
1B 45 01	SBB AX,[DI+1]
1B C6	SBB AX,SI
1B 04	SBB AX,[SI]
1B 44 01	SBB AX,[SI+1]
1B C5	SBB AX,BP
1B DF	SBB BX,DI
1B 1D	SBB BX,[DI]
1B 5D 01	SBB BX,[DI+1]
1B DE	SBB BX,SI
1B 1C	SBB BX,[SI]
1B 5C 01	SBB BX,[SI+1]
1B DD	SBB BX,BP
1B CF	SBB CX,DI
1B 0D	SBB CX,[DI]
1B 4D 01	SBB CX,[DI+1]

ค่าออฟโค้ด	ลักษณะของคำสั่ง
1B 0C	SBB CX,[SI]
1B 4C 01	SBB CX,[SI+1]
1B CD	SBB CX,BP
1B D7	SBB DX,DI
1B 15	SBB DX,[DI]
1B 55 01	SBB DX,[DI+1]
1B D6	SBB DX,SI
1B 14	SBB DX,[SI]
1B 54 01	SBB DX,[SI+1]
1B D5	SBB DX,BP
19 44 01	SBB [SI+1],AX
19 5C 01	SBB [SI+1],BX
19 4C 01	SBB [SI+1],CX
19 54 01	SBB [SI+1],DX
19 04	SBB [SI],AX
19 1C	SBB [SI],BX
19 0C	SBB [SI],CX
19 14	SBB [SI],DX
19 46 01	SBB [BP+1],AX
19 5E 01	SBB [BP+1],BX
19 4E 01	SBB [BP+1],CX
19 56 01	SBB [BP+1],DX
1B E8	SBB BP,AX
1B EB	SBB BP,BX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
1B EA	SBB BP,DX
19 45 01	SBB [DI+1],AX
19 5D 01	SBB [DI+1],BX
19 4D 01	SBB [DI+1],CX
19 55 01	SBB [DI+1],DX
19 05	SBB [DI],AX
19 1D	SBB [DI],BX
19 0D	SBB [DI],CX
19 15	SBB [DI],DX
1D 0B 10	SBB AX,100BH
81 DB 0A 10	SBB BX,100AH
81 D9 0A 10	SBB CX,100AH
81 DA 0A 10	SBB DX,100AH
80 DC 0B	SBB AH,0BH
1C 0B	SBB AL,0BH
80 DF 0A	SBB BH,0AH
80 DD 0A	SBB CH,0AH
80 D9 0A	SBB CL,0AH
80 DE 0A	SBB DH,0AH
80 DA 0A	SBB DL,0AH
85 C3	TEST AX,BX
85 C1	TEST AX,CX
85 C2	TEST AX,DX
85 D8	TEST BX,AX

ค่าออฟโค้ด	ลักษณะของคำสั่ง
85 DA	TEST BX,DX
85 C8	TEST CX,AX
85 CB	TEST CX,BX
85 CA	TEST CX,DX
85 D0	TEST DX,AX
85 D3	TEST DX,BX
85 D1	TEST DX,CX
84 E7	TEST AH,BH
84 E5	TEST AH,CH
84 E6	TEST AH,DH
84 E3	TEST AH,BL
84 E1	TEST AH,CL
84 E2	TEST AH,DL
84 C3	TEST AL,BL
84 C1	TEST AL,CL
84 C2	TEST AL,DL
84 C7	TEST AL,BH
84 C5	TEST AL,CH
84 C6	TEST AL,DH
84 FC	TEST BH,AH
84 FD	TEST BH,CH
84 FE	TEST BH,DH
84 F8	TEST BH,AL
84 F9	TEST BH,CL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
84 EC	TEST CH,AH
84 EF	TEST CH,BH
84 EE	TEST CH,DH
84 E8	TEST CH,AL
84 EB	TEST CH,BL
84 EA	TEST CH,DL
84 C8	TEST CL,AL
84 CB	TEST CL,BL
84 CA	TEST CL,DL
84 CC	TEST CL,AH
84 CF	TEST CL,BH
84 CE	TEST CL,DH
84 F4	TEST DH,AH
84 F7	TEST DH,BH
84 F5	TEST DH,CH
84 F0	TEST DH,AL
84 F3	TEST DH,BL
84 F1	TEST DH,CL
84 D4	TEST DL,AH
84 D7	TEST DL,BH
84 D5	TEST DL,CH
84 D0	TEST DL,AL
84 D3	TEST DL,BL
84 D1	TEST DL,CL

ค่าออฟโค้ด	ลักษณะของคำสั่ง
85 05	TEST AX,[DI]
85 45 01	TEST AX,[DI+1]
85 C6	TEST AX,SI
85 04	TEST AX,[SI]
85 44 01	TEST AX,[SI+1]
85 C5	TEST AX,BP
85 DF	TEST BX,DI
85 1D	TEST BX,[DI]
85 5D 01	TEST BX,[DI+1]
85 DE	TEST BX,SI
85 1C	TEST BX,[SI]
85 5C 01	TESTBX,[SI+1]
85 DD	TEST BX,BP
85 CF	TEST CX,DI
85 0D	TEST CX,[DI]
85 4D 01	TEST CX,[DI+1]
85 CE	TEST CX,SI
85 0C	TEST CX,[SI]
85 4C 01	TEST CX,[SI+1]
85 CD	TEST CX,BP
85 D7	TEST DX,DI
85 15	TEST DX,[DI]
85 55 01	TEST DX,[DI+1]
85 D6	TEST DX,SI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 (ต่อ) ลักษณะของคำสั่งที่ใช้งาน

ค่าออฟโค้ด	ลักษณะของคำสั่ง
85 54 01	TEST DX,[SI+1]
85 D5	TEST DX,BP
F7 C3 0A 10	TEST BX,100AH
F7 C1 0A 10	TEST CX,100AH
F7 C2 0A 10	TEST DX,100AH
F6 C4 0B	TEST AH,0BH
A8 0B	TEST AL,0BH
F6 C7 0A	TEST BH,0AH
F6 C5 0A	TEST CH,0AH
F6 C1 0A	TEST CL,0AH
F6 C6 0A	TEST DH,0AH
F6 C2 0A	TEST DL,0AH

ค่าออฟโค้ด	ลักษณะของคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของคำสั่งในการใช้งาน

```
3E              SEG   DS
2E              SEG   CS
26              SEG   ES
36              SEG   SS

E8 C4 FF        CALL  BEGIN
E8 C1 FF        CALL  NEAR PTR BEGIN
9A 00 01 00 00  CALL  FAR PTR BEGIN
9A FF FF 00 F0  CALL  FAR PTR $FFFFF
9A 34 12 78 56  CALL  FAR PTR 1234H,5678H      ;CALL IP,CS ***
EA 34 12 78 56  JMP   FAR PTR 1234H,5678H      ;CALL IP,CS ***
                                  ;CALL_WORD_PTR_mod_00_disp0_r/m
OPCODE      COMMAND

F1 0        CALL  [BX+SI]
FF 11       CALL  [BX+DI]
FF 12       CALL  [BP+SI]
FF 13       CALL  [BP+DI]
FF 14       CALL  [SI]
FF 15       CALL  [DI]
FF 16 57 05 CALL  [DRWRD]
FF 17       CALL  [BX]
FF D0       CALL  AX
FF D1       CALL  CX
FF D2       CALL  DX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FF D3          CALL  BX
FF D4          CALL  SP
FF D5          CALL  BP
FF D6          CALL  SI
FF D7          CALL  DI
C3             RET
C2 02 00       RET   2
C2 FF FF       RET   65535
C3             RETN
C2 02 00       RETN  2
C2 FF FF       RETN  65535
CB             RETF
CA 02 00       RETF  2
CA FF FF       RETF  65535

        ;*******************************************************
        ;
        ; NOTE!  ALL MEMORY OPERANDS AND THEIR DISPLACEMENTS
        ;         MUST BE PLACED WITHIN BRACKETS []. ALL
        ; INSTRUCTIONS WHOSE SIZE (BYTE, WORD OR DWORD) CAN
        ; NOT BE DETERMINED BY REGISTER SIZE OR INSTRUCTION
        ; TYPE, MUST BE EXPLICITLY STATED.
        ;
OPCODE          COMMAND

8B C7          MOV   AX,DI
8B CE          MOV   CX,SI
8B D5          MOV   DX,BP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8B DC       MOV  BX,SP
8B E3       MOV  SP,BX
8B EA       MOV  BP,DX
8B F1       MOV  SI,CX
8B F8       MOV  DI,AX
8E DF       MOV  DS,DI
8E DE       MOV  DS,SI
8E C5       MOV  ES,BP
8E D4       MOV  SS,SP
8E DB       MOV  DS,BX
8E DA       MOV  DS,DX
8E C1       MOV  ES,CX
8E D0       MOV  SS,AX          ;MOV   mod 11 r16,segreg
8C D8       MOV  AX,DS
8C C9       MOV  CX,CS
8C C2       MOV  DX,ES
8C D3       MOV  BX,SS
8C DC       MOV  SP,DS
8C CD       MOV  BP,CS
8C C6       MOV  SI,ES
8C D7       MOV  DI,SS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

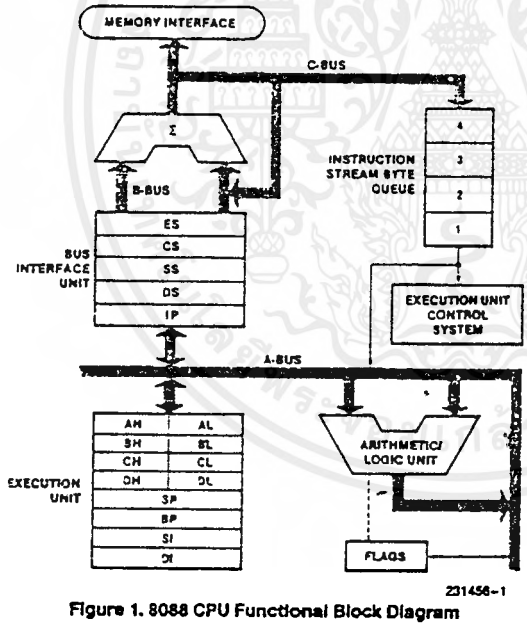
ภาคผนวก ง
รายละเอียดข้อมูลและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8088
8-BIT HMOS MICROPROCESSOR
8088/8088-2

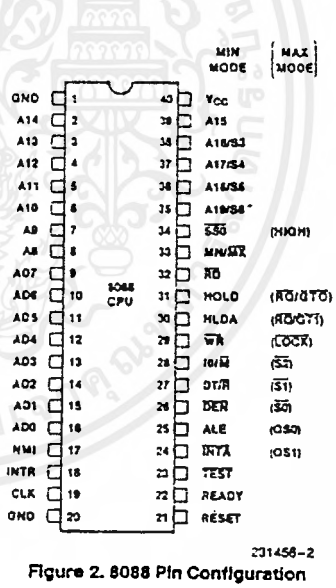
- 8-Bit Data Bus Interface
- 16-Bit Internal Architecture
- Direct Addressing Capability to 1 Mbyte of Memory
- Direct Software Compatibility with 8086 CPU
- 14-Word by 16-Bit Register Set with Symmetrical Operations
- 24 Operand Addressing Modes
- Byte, Word, and Block Operations
- 8-Bit and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal, Including Multiply and Divide
- Two Clock Rates:
  - —5 MHz for 8088
  - —8 MHz for 8088-2
- Available in EXPRESS
  - —Standard Temperature Range
  - —Extended Temperature Range

The Intel® 8088 is a high performance microprocessor implemented in N-channel, depletion load, silicon gate technology (HMOS), and packaged in a 40-pin CERDIP package. The processor has attributes of both 8- and 16-bit microprocessors. It is directly compatible with 8086 software and 8080/8085 hardware and peripherals.



231456-1

Figure 1. 8088 CPU Functional Block Diagram



231456-2

Figure 2. 8088 Pin Configuration

October 1987
Order Number: 231456-003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Pin Description

The following pin function descriptions are for 8088 systems in either minimum or maximum mode. The "local bus" in these descriptions is the direct multiplexed bus interface connection to the 8088 (without regard to additional bus buffers).

Symbol	Pin No.	Type	Name and Function
AD7–AD0	9–16	I/O	**ADDRESS DATA BUS:** These lines constitute the time multiplexed memory/IO address (T1) and data (T2, T3, Tw, T4) bus. These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".
A15–A8	2–8, 39	O	**ADDRESS BUS:** These lines provide address bits 8 through 15 for the entire bus cycle (T1–T4). These lines do not have to be latched by ALE to remain valid. A15–A8 are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".
A19/S6, A18/S5, A17/S4, A16/S3	35–38	O	**ADDRESS/STATUS:** During T1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is available on these lines during T2, T3, Tw, and T4. S6 is always low. The status of the interrupt enable flag bit (S5) is updated at the beginning of each clock cycle. S4 and S3 are encoded as shown. This information indicates which segment register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge".

S4	S3	Characteristics
0 (LOW)	0	Alternate Data
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data
S6 is 0 (LOW)		

$\overline{RD}$	32	O	**READ:** Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the IO/$\overline{M}$ pin or S2. This signal is used to read devices which reside on the 8088 local bus. $\overline{RD}$ is active LOW during T2, T3 and Tw of any read cycle, and is guaranteed to remain HIGH in T2 until the 8088 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge".
READY	22	I	**READY:** is the acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 8284 clock generator to form READY. This signal is active HIGH. The 8088 READY input is not synchronized. Correct operation is not guaranteed if the set up and hold times are not met.
INTR	18	I	**INTERRUPT REQUEST:** is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
$\overline{TEST}$	23	I	**$\overline{TEST}$:** input is examined by the "wait for test" instruction. If the $\overline{TEST}$ input is LOW, execution continues, otherwise the processor waits in an "idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
NMI	17	I	**NON-MASKABLE INTERRUPT:** is an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
RESET	21	I	**RESET:** causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.
CLK.	19	I	**CLOCK:** provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
V_{CC}	40		V_{CC}: is the +5V ±10% power supply pin.
GND	1, 20		**GND:** are the ground pins.
MN/$\overline{MX}$	33	I	**MINIMUM/MAXIMUM:** indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

The following pin function descriptions are for the 8088 minimum mode (i.e., MN/$\overline{MX}$ = V_{CC}). Only the pin functions which are unique to minimum mode are described; all other pin functions are as described above.

Symbol	Pin No.	Type	Name and Function
IO/$\overline{M}$	28	O	**STATUS LINE:** is an inverted maximum mode $\overline{S2}$. It is used to distinguish a memory access from an I/O access. IO/$\overline{M}$ becomes valid in the T4 preceding a bus cycle and remains valid until the final T4 of the cycle (I/O = HIGH, M = LOW). IO/$\overline{M}$ floats to 3-state OFF in local bus "hold acknowledge".
$\overline{WR}$	29	O	**WRITE:** strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/$\overline{M}$ signal. WR is active for T2, T3, and Tw of any write cycle. It is active LOW, and floats to 3-state OFF in local bus "hold acknowledge".
$\overline{INTA}$	24	O	**INTA:** is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T2, T3, and Tw of each interrupt acknowledge cycle.
ALE	25	O	**ADDRESS LATCH ENABLE:** is provided by the processor to latch the address into an address latch. It is a HIGH pulse active during clock low of T1 of any bus cycle. Note that ALE is never floated.
DT/$\overline{R}$	27	O	**DATA TRANSMIT/RECEIVE:** is needed in a minimum system that desires to use a data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically, DT/$\overline{R}$ is equivalent to $\overline{S1}$ in the maximum mode, and its timing is the same as for IO/$\overline{M}$ (T = HIGH, R = LOW). This signal floats to 3-state OFF in local "hold acknowledge".
$\overline{DEN}$	26	O	**DATA ENABLE:** is provided as an output enable for the data bus transceiver in a minimum system which uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access, and for $\overline{INTA}$ cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T2 until the middle of T4, while for a write cycle, it is active from the beginning of T2 until the middle of T4. $\overline{DEN}$ floats to 3-state OFF during local bus "hold acknowledge".

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
HOLD, HLDA	31, 30	I, O	**HOLD:** indicates that another master is requesting a local bus "hold". To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement, in the middle of a T4 or T1 clock cycle. Simultaneous with the issuance of HLDA the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. Hold is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set up time.
$\overline{SSO}$	34	O	**STATUS LINE:** is logically equivalent to $\overline{SO}$ in the maximum mode. The combination of $\overline{SSO}$, IO/$\overline{M}$ and DT/$\overline{R}$ allows the system to completely decode the current bus cycle status.

IO/$\overline{M}$	DT/$\overline{R}$	$\overline{SSO}$	Characteristics
1(HIGH)	0	0	Interrupt Acknowledge
1	0	1	Read I/O Port
1	1	0	Write I/O Port
1	1	1	Halt
0(LOW)	0	0	Code Access
0	0	1	Read Memory
0	1	0	Write Memory
0	1	1	Passive

The following pin function descriptions are for the 8088/8288 system in maximum mode (i.e., MN/$\overline{MX}$ = GND). Only the pin functions which are unique to maximum mode are described; all other pin functions are as described above.

Symbol	Pin No.	Type	Name and Function
$\overline{S2}$, $\overline{S1}$, $\overline{S0}$	26–28	O	**STATUS:** is active during clock high of T4, T1, and T2, and is returned to the passive state (1,1,1) during T3 or during Tw when READY is HIGH. This status is used by the 8288 bus controller to generate all memory and I/O access control signals. Any change by $\overline{S2}$, $\overline{S1}$, or $\overline{S0}$ during T4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T3 and Tw is used to indicate the end of a bus cycle. These signals float to 3-state OFF during "hold acknowledge". During the first clock cycle after RESET becomes active, these signals are active HIGH. After this first clock, they float to 3-state OFF.

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Characteristics
0(LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1(HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
$\overline{RQ}/\overline{GT0}$, $\overline{RQ}/\overline{GT1}$	30, 31	I/O	**REQUEST/GRANT:** pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT0}$ having higher priority than $\overline{RQ}/\overline{GT1}$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor, so may be left unconnected. The request/grant sequence is as follows (See Figure 8): 1. A pulse of one CLK wide from another local bus master indicates a local bus request ("hold") to the 8088 (pulse 1). 2. During a T4 or T1 clock cycle, a pulse one clock wide from the 8088 to the requesting master (pulse 2), indicates that the 8088 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge". The same rules as for HOLD/HOLDA apply as for when the bus is released. 3. A pulse one CLK wide from the requesting master indicates to the 8088 (pulse 3) that the "hold" request is about to end and that the 8088 can reclaim the local bus at the next CLK. The CPU then enters T4. Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active LOW. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T4 of the cycle when all the following conditions are met: 1. Request occurs on or before T2. 2. Current cycle is not the low bit of a word. 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is idle when the request is made the two possible events will follow: 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
$\overline{LOCK}$	29	O	$\overline{LOCK}$: indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active (LOW). The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to 3-state off in "hold acknowledge".
QS1, QS0	24, 25	O	**QUEUE STATUS:** provide status to allow external tracking of the internal 8088 instruction queue. The queue status is valid during the CLK cycle after which the queue operation is performed.

QS1	QS0	Characteristics
0(LOW)	0	No Operation
0	1	First Byte of Opcode from Queue
1(HIGH)	0	Empty the Queue
1	1	Subsequent Byte from Queue

—	34	O	Pin 34 is always high in the maximum mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bus Operation

The 8088 address/data bus is broken into three parts—the lower eight address/data bits (AD0–AD7), the middle eight address bits (A8–A15), and the upper four address bits (A16–A19). The address/data bits and the highest four address bits are time multiplexed. This technique provides the most efficient use of pins on the processor, permitting the use of a standard 40 lead package. The middle eight address bits are not multiplexed, i.e. they remain valid throughout each bus cycle. In addition, the bus can be demultiplexed at the processor with a single address latch if a standard, non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3, and T4 (See Figure 8). The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for chang-

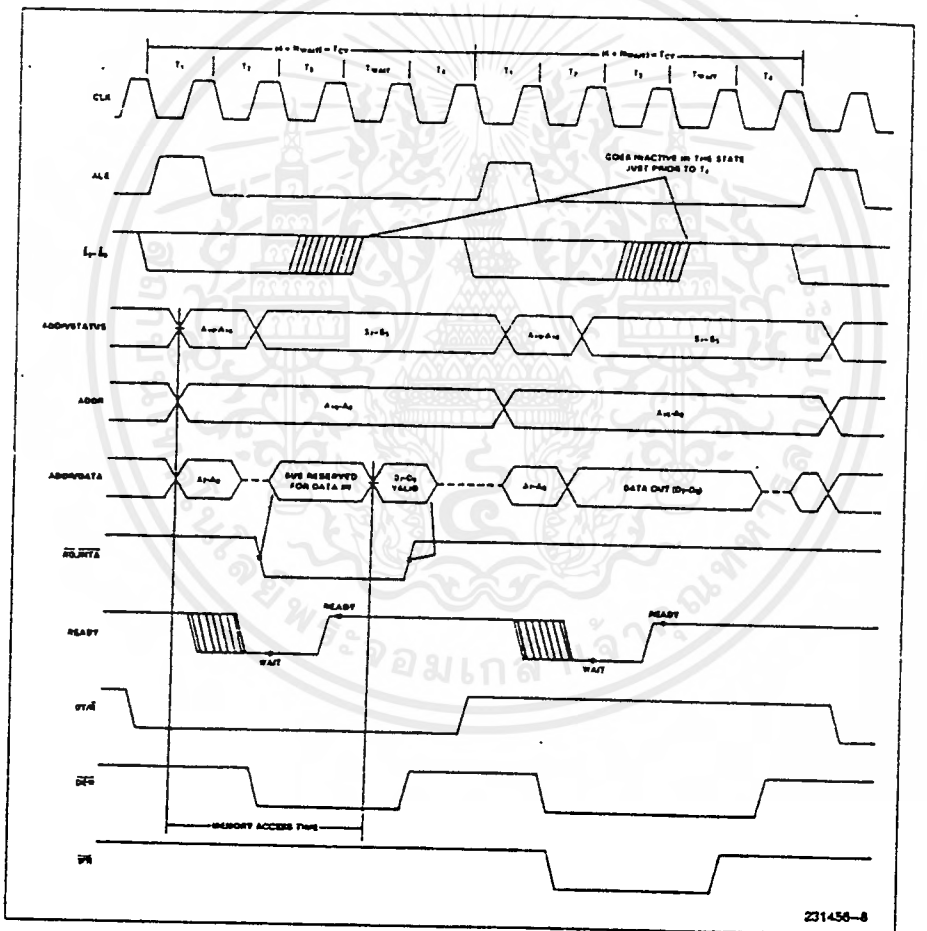


Figure 8. Basic System Timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

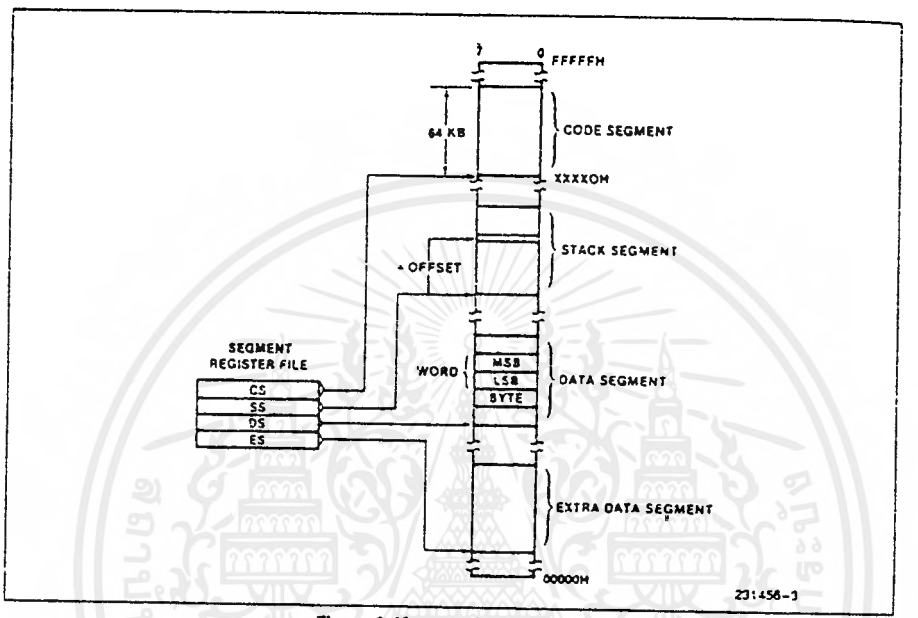


Figure 3. Memory Organization

FUNCTIONAL DESCRIPTION

Memory Organization

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra data, and stack segments of up to 64K bytes each, with each segment falling on 16-byte boundaries (See Figure 3).

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the rules of the following table. All information in one segment type share the same logical attributes (e.g. code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster, and more structured.

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location. The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Memory Reference Used	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when: relative to stack, destination of string operation, or explicity overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment override.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Certain locations in memory are reserved for specific CPU operations (See Figure 4). Locations from addresses FFFF0H through FFFFFH are reserved for operations including a jump to the initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations 00000H through 003FFH are reserved for interrupt operations. Four-byte pointers consisting of a 16-bit segment address and a 16-bit offset address direct program flow to one of the 256 possible interrupt service routines. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

Minimum and Maximum Modes

The requirements for supporting minimum and maximum 8088 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 8088 is equipped with a strap pin (MN/$\overline{MX}$) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/$\overline{MX}$ pin is strapped to GND, the 8088 defines pins 24 through 31 and 34 in maximum mode. When the MN/$\overline{MX}$ pin is strapped to V_{CC}, the 8088 generates bus control signals itself on pins 24 through 31 and 34.

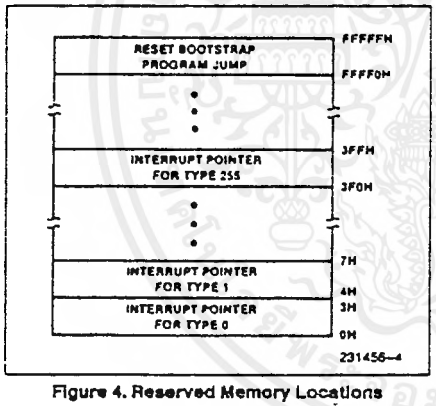


Figure 4. Reserved Memory Locations

The minimum mode 8088 can be used with either a multiplexed or demultiplexed bus. The multiplexed bus configuration is compatible with the MCS-85™ multiplexed bus peripherals. This configuration (See Figure 5) provides the user with a minimum chip count system. This architecture provides the 8088 processing power in a highly integrated form.

The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). A third latch can be used for buffering if the address bus loading requires it. A transceiver can also be used if data bus buffering is required (See Figure 6). The 8088 provides $\overline{DEN}$ and DT/$\overline{R}$ to control the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the 8288 bus controller (See Figure 7). The 8288 decodes status lines $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$, and provides the system with all bus control signals. Moving the bus control to the 8288 provides better source and sink current capability to the control lines, and frees the 8088 pins for extended large system features. Hardware lock, queue status, and two request/grant interfaces are provided by the 8088 in maximum mode. These features allow co-processors in local bus and remote bus configurations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "wait" states (Tw) are inserted between T3 and T4. Each inserted "wait" state is of the same duration as a CLK cycle. Periods can occur between 8088 driven bus cycles. These are referred to as "idle" states (Ti), or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

During T1 of any bus cycle, the ALE (address latch enable) signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ are used by the bus controller, in maximum mode, to identify the type of bus transaction according to the following table:

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Characteristics
0(LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1(HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (No Bus Cycle)

Status bits S3 through S6 are multiplexed with high order address bits and are therefore valid during T2 through T4. S3 and S4 indicate which segment register was used for this bus cycle in forming the address according to the following table:

S_4	S_3	Characteristics
0(LOW)	0	Alternate Data (Extra Segment)
0	1	Stack
1(HIGH)	0	Code or None
1	1	Data

S5 is a reflection of the PSW interrupt enable bit. S6 is always equal to 0.

I/O Addressing

In the 8088, I/O operations can address up to a maximum of 64K I/O registers. The I/O address appears in the same format as the memory address on bus lines A15–A0. The address lines A19–A16 are zero in I/O operations. The variable I/O instructions, which use register DX as a pointer, have full address capability, while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

Designers familiar with the 8085 or upgrading an 8085 design should note that the 8085 addresses I/O with an 8-bit address on both halves of the 16-bit address bus. The 8088 uses a full 16-bit address on its lower 16 address lines.

EXTERNAL INTERFACE

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8088 RESET is required to be HIGH for greater than four clock cycles. The 8088 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 clock cycles. After this interval the 8088 operates normally, beginning with the instruction in absolute locations FFFF0H (See Figure 4). The RESET input is internally synchronized to the processor clock. At initialization, the HIGH to LOW transition of RESET must occur no sooner than 50 μs after power up, to allow complete initialization of the 8088.

NMI asserted prior to the 2nd clock after the end of RESET will not be honored. If NMI is asserted after that point and during the internal reset sequence, the processor may execute one instruction before responding to the interrupt. A hold request active immediately after RESET will be honored before the first instruction fetch.

All 3-state outputs float to 3-state OFF during RESET. Status is active in the idle state for the first clock after RESET becomes active and then floats to 3-state OFF. ALE and HLDA are driven low.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the instruction set description in the iAPX 88 book or the iAPX 86,88 User's Manual. Hardware interrupts can be classified as nonmaskable or maskable.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interrupts result in a transfer of control to a new program location. A 256 element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (See Figure 4), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number, during the interrupt acknowledge sequence, which is used to vector through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt (NMI) pin which has higher priority than the maskable interrupt request (INTR) pin. A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW to HIGH transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than two clock cycles, but is not required to be synchronized to the clock. Any higher going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves (2 bytes in the case of word moves) of a block type instruction. Worst case response to NMI would be for multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 8088 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable (IF) flag bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block type instruction. During interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single step), although the FLAGS register which is automatically pushed onto the stack reflects the state of the processor prior to the interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (See Figure 9), the processor executes two successive (back to back) interrupt acknowledge cycles. The 8088 emits the LOCK signal (maximum mode only) from T2 of the first bus cycle until T2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interrupt system (e.g., 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The interrupt return instruction includes a flags pop which returns the status of the original interrupt enable bit when it restores the flags.

HALT

When a software HALT instruction is executed, the processor indicates that it is entering the HALT state in one of two ways, depending upon which mode is strapped. In minimum mode, the processor issues ALE, delayed by one clock cycle, to allow the system to latch the halt status. Halt status is available on IO/$\overline{M}$, DT/$\overline{R}$, and $\overline{SSO}$. In maximum mode, the processor issues appropriate HALT status on $\overline{S2}$, $\overline{S1}$, and $\overline{S0}$, and the 8288 bus controller issues one ALE. The 8088 will not leave the HALT state when a local bus hold is entered while in HALT. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An interrupt request or RESET will force the 8088 out of the HALT state.

Read/Modify/Write (Semaphore) Operations via LOCK

The LOCK status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This allows the processor to perform read/modify/write operations on memory (via the "exchange register with memory" instruction), without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{LOCK}$ signal is activated (LOW) in the clock cycle following decoding of the $\overline{LOCK}$ prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the $\overline{LOCK}$ prefix. While $\overline{LOCK}$ is active, a request on a $\overline{RQ}$/$\overline{GT}$ pin will be recorded, and then honored at the end of the $\overline{LOCK}$.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

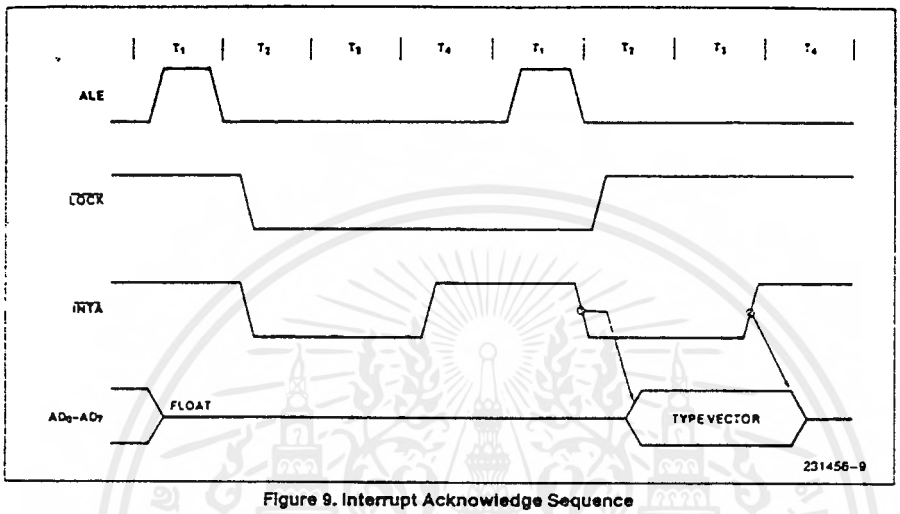


Figure 9. Interrupt Acknowledge Sequence

External Synchronization via $\overline{TEST}$

As an alternative to interrupts, the 8088 provides a single software-testable input pin ($\overline{TEST}$). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the $\overline{TEST}$ input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the 8088 3-states all output drivers. If interrupts are enabled, the 8088 will recognize interrupts and process them. The WAIT instruction is then refetched, and reexecuted.

Basic System Timing

In minimum mode, the MN/$\overline{MX}$ pin is strapped to V_{CC} and the processor emits bus control signals compatible with the 8085 bus structure. In maximum mode, the MN/$\overline{MX}$ pin is strapped to GND and the processor emits coded status information which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals.

System Timing—Minimum System

(See Figure 8)

The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal. The trailing (low going) edge of this signal is used to latch the address information, which is valid on the address/data bus (AD0–AD7) at this time, into the 8282/8283 latch. Address lines A8 through A15 do not need to be latched because they remain valid throughout the bus cycle. From T1 to T4 the IO/$\overline{M}$ signal indicates a memory or I/O operation. At T2 the address is removed from the address/data bus and the bus goes to a high impedance state. The read control signal is also asserted at T2. The read ($\overline{RD}$) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver is required to buffer the 8088 local bus, signals DT/$\overline{R}$ and $\overline{DEN}$ are provided by the 8088.

A write cycle also begins with the assertion of ALE and the emission of the address. The IO/$\overline{M}$ signal is again asserted to indicate a memory or I/O write operation. In T2, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T4. During T2, T3, and Tw, the processor asserts the write control signal. The write ($\overline{WR}$) signal becomes active at the beginning of T2, as opposed to the read, which is delayed somewhat into T2 to provide time for the bus to float.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge (INTA) signal is asserted in place of the read (RD) signal and the address bus is floated. (See Figure 9) In the second of two successive INTA cycles, a byte of information is read from the data bus, as supplied by the interrupt system logic (i.e. 8259A priority interrupt controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into the interrupt vector lookup table, as described earlier.

Bus Timing—Medium Complexity Systems

(See Figure 10)

For medium complexity systems, the MN/MX pin is connected to GND and the 8288 bus controller is added to the system, as well as a latch for latching the system address, and a transceiver to allow for bus loading greater than the 8088 is capable of handling. Signals ALE, DEN, and DT/R are generated by the 8288 instead of the processor in this configuration, although their timing remains relatively the same. The 8088 status outputs (S2, S1, and S0) provide type of cycle information and become 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The transceiver receives the usual T and OE inputs from the 8288's DT/R and DEN outputs.

The pointer into the interrupt vector table, which is passed during the second INTA cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8289A priority interrupt controller is positioned on the local bus, a TTL gate is required to disable the transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll".

The 8088 Compared to the 8086

The 8088 CPU is an 8-bit processor designed around the 8086 internal structure. Most internal functions of the 8088 are identical to the equivalent 8086 functions. The 8088 handles the external bus the same way the 8086 does with the distinction of handling only 8 bits at a time. Sixteen-bit operands are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time. The internal register structure is identical and all instructions have the same end result. The differences between the 8088 and 8086 are outlined below. The engineer who is unfamiliar with the 8086 is referred to the iAPX 86, 88 User's Manual, Chapters 2 and 4, for function description and instruction set information. Internally, there are three differences between the 8088 and the 8086. All changes are related to the 8-bit bus interface.

- The queue length is 4 bytes in the 8088, whereas the 8086 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 8088 BIU will fetch a new instruction to load into the queue each time there is a 1 byte hole (space available) in the queue. The 8086 waits until a 2-byte space is available.
- The internal execution time of the instruction set is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU is also limited by the speed of instruction fetches. This latter problem only occurs when a series of simple operations occur. When the more sophisticated instructions of the 8088 are being used, the queue has time to fill and the execution proceeds as fast as the execution unit will allow.

The 8088 and 8086 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally as well on an 8088 and an 8086.

The hardware interface of the 8088 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes:

- A8–A15—These pins are only address outputs on the 8088. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- BHE has no meaning on the 8088 and has been eliminated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $\overline{SSO}$ provides the $\overline{SO}$ status information in the minimum mode. This output occurs on pin 34 in minimum mode only. DT/$\overline{R}$, IO/$\overline{M}$, and $\overline{SSO}$ provide the complete bus status in minimum mode.
- IO/$\overline{M}$ has been inverted to be compatible with the MCS-85 bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with ALE.

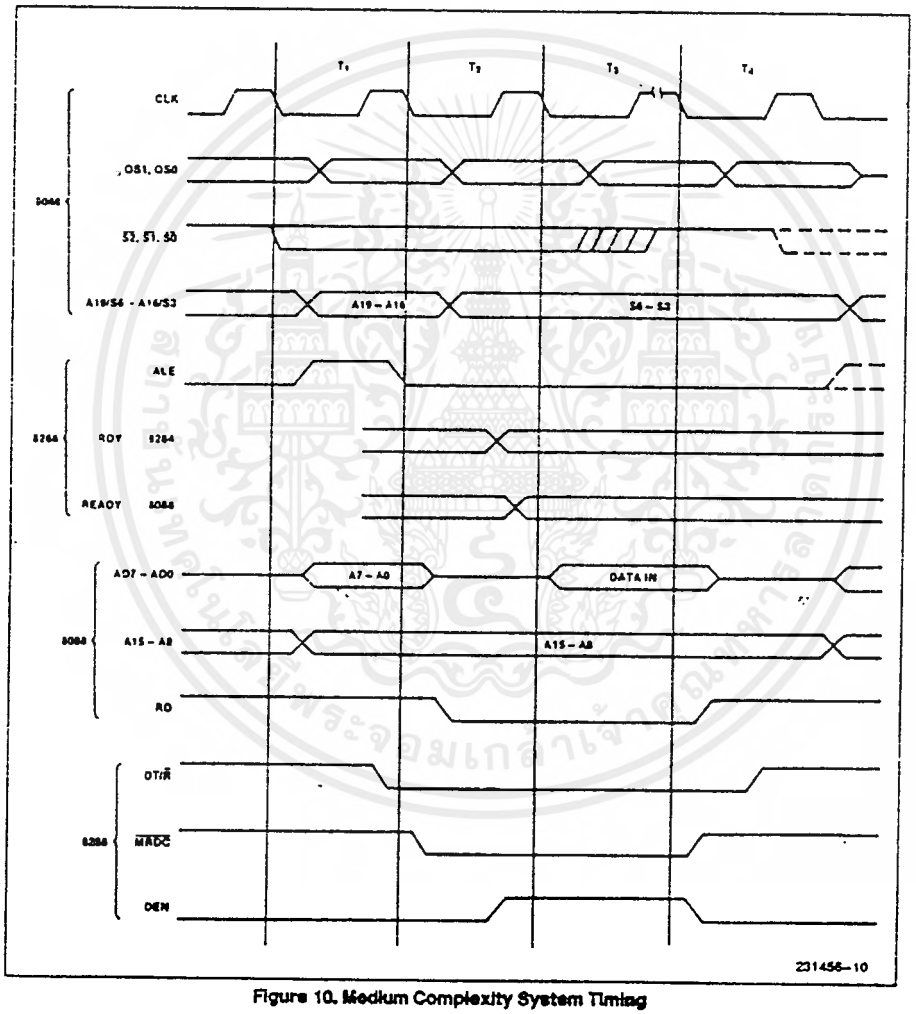


Figure 10. Medium Complexity System Timing

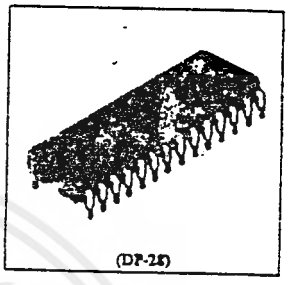
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HM6264P-10, HM6264P-12, HM6264P-15

8192-word x 8-bit High Speed Static CMOS RAM

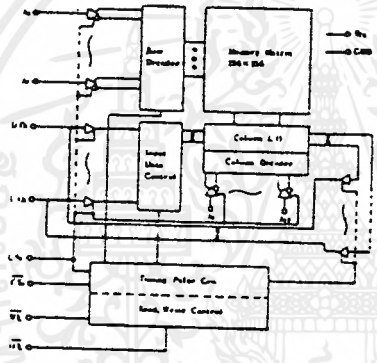
■ FEATURES

- Fast access Time — 100ns/120ns/150ns (max;)
- Low Power Standby — Standby: 0.1mW (typ.)
- Low Power Operation — Operating: 200mW (typ.)
- Single +5V Supply
- Completely Static Memory..... No clock or Timing Strobe Required
- Equal Access and Cycle Time
- Common Data Input and Output, Three State Output
- Directly TTL Compatible: All Input and Output
- Standard 28pin Package Configuration
- Pin Out Compatible with 64K EPROM HN482764

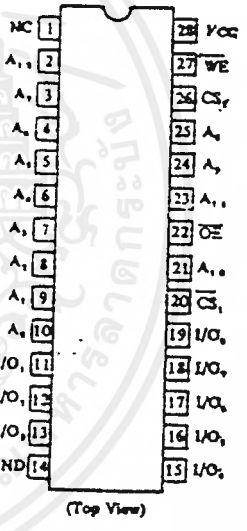


(DP-28)

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Rating	Unit
Terminal Voltage *	V_T	−0.5 ** to +7.0	V
Power Dissipation	P_T	1.0	W
Operating Temperature	T_{opr}	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C
Storage Temperature (Under Bias)	T_{bias}	−10 to +85	°C

* With respect to GND. ** Pulse width 50ns: −3.0V

■ TRUTH TABLE

$\overline{WE}$	$\overline{CS_1}$	CS_2	$\overline{OE}$	Mode	I/O Pin	V_{CC} Current	Note
X	H	X	X	Not Selected (Power Down)	High Z	I_{SB}, I_{SB1}	
X	X	L	X		High Z	I_{SB}, I_{SB2}	
H	L	H	H	Output Disabled	High Z	I_{CC}, I_{CC2}	
H	L	H	L	Read	Dout	I_{CC}, I_{CC1}	
L	L	H	H	Write	Din	I_{CC}, I_{CC1}	Write Cycle (1)
L	L	H	L		Din	I_{CC}, I_{CC1}	Write Cycle (2)

x : Don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel®

27256
256K (32K x 8) PRODUCTION AND UV ERASABLE PROMS

- **New Quick-Pulse Programming™ Algorithm for Plastic P27256**
  - **— 4 Second Programming**
  - **— int$_e$ligent Programming™ Algorithm Compatible**
- **Fast Access Time**
  - **— 170 ns D27256-1**
  - **— 200 ns P27256-2**
- **int$_e$ligent Identifier™ Mode**
- **Plastic Production P27256 is Compatible with Auto-Insertion Equipment**
- **Moisture Resistant**
- **Industry Standard Pinout ... JEDEC Approved ... 28 Lead Cerdip and Plastic Package**

(See Packaging Spec, Order # 231369)

The Intel 27256 is a 5V only, 262,144-bit Ultraviolet Erasable (Cerdip)/plastic production (P27256) electrically programmable read-only memory (EPROM). Organized as 32K words by 8 bits, individual bytes can be accessed in less than 170 ns (27256-1). This is compatible with high performance microprocessors, such as the Intel iAPX 186, allowing full speed operation without the addition of performance-degrading WAIT states. The 27256 is also directly compatible with Intel's 8051 family of microcontrollers.

The Plastic P27256 is ideal for high volume production environments where code flexibility is crucial. Plastic packaging is also well-suited to auto-insertion equipment in cost-effective automated assembly lines. Intel's new Quick-Pulse Programming Algorithm enables the P27256 to be programmed within four seconds (plus programmer overhead). Programming equipment which takes advantage of this innovation will electronically identify the EPROM with the help of the int$_e$ligent Identifier and rapidly program it using a superior programming method. The int$_e$ligent Programming Algorithm may be utilized in the absence of such equipment.

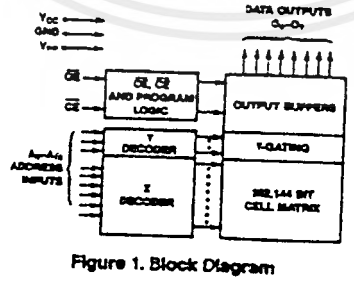
The 27256 enables implementation of new, advanced systems with firmware-intensive architectures. The combination of the 27256's high-density, cost-effective EPROM storage, and new advanced microprocessors having megabit addressing capability provides designers with opportunities to engineer user-friendly, high reliability, high-performance systems.

The 27256's large storage capability of 32 K-bytes enables it to function as a high-density software carrier. Entire operating systems, diagnostics, high-level language programs and specialized application software can reside in a 27256 EPROM directly on a system's memory bus. This permits immediate microprocessor access and execution of software and eliminates the need for time-consuming disk accesses and downloads.

Two-line control and JEDEC-approved, 28-pin packaging are standard features of all Intel high-density EPROMs. This assures easy microprocessor interfacing and minimum design efforts when upgrading, adding, or choosing between nonvolatile memory alternatives.

The 27256 is manufactured using Intel's advanced HMOS*II-E technology.

*HMOS is a patented process of Intel Corporation.



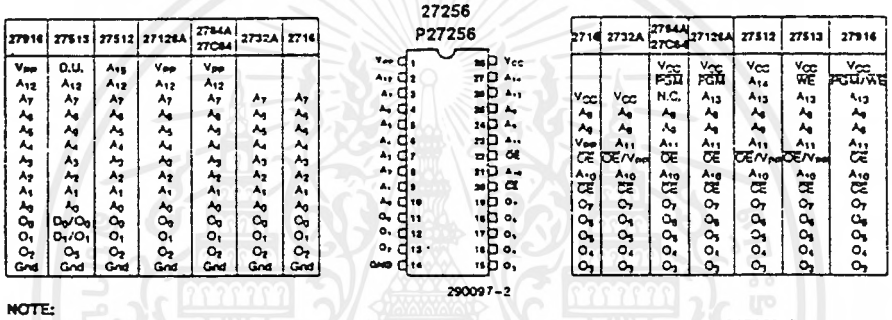
290087-1

Figure 1. Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Names

A_0–A_{14}	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
D.U.	Don't Use
$\overline{WE}$	Write Enable



NOTE:
Intel "Universal Site" Compatible EPROM pin configurations are shown in the blocks adjacent to the P27256 pins.

Figure 2. Cerdip/Plastic DIP Pin Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

74156, LS156
Decoders/Demultiplexers

Dual 2-Line To 4-Line Decoder/Demultiplexer (Open Collector)
Product Specification

Logic Products

FEATURES
- Common Address inputs
- True or complement data demultiplexing
- Dual 1-of-4 or 1-of-8 decoding
- Function generator applications
- Outputs can be tied together

DESCRIPTION
The '156 is a Dual 1-of-4 Decoder/Demultiplexer with common Address inputs and gated Enable inputs. Each decoder section, when enabled, will accept the binary weighted Address inputs (A_0, A_1) and provide four mutually exclusive active-LOW outputs ($\overline{0} - \overline{3}$). When the enable requirements of each decoder are not met, all outputs of that decoder are HIGH.

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74156	20ns	25mA
74LS156	31ns	6.1mA

ORDERING CODE

PACKAGES	COMMERCIAL RANGE V_{CC} = 5V ±5%; T_A = 0°C to +70°C
Plastic DIP	N74156N, N74LS156N
Plastic SO	N74LS156D

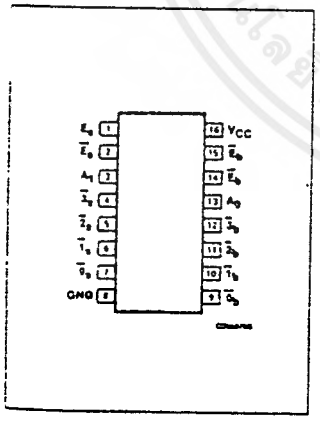
NOTE:
For information regarding devices processed to Military Specifications, see the Signetics Military Products Data Manual.

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

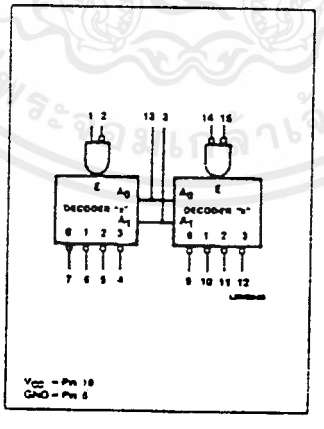
PINS	DESCRIPTION	74	74LS
All	Inputs	1ul	1LSul
All	Outputs	10ul	10LSul

NOTE:
Where a 74 unit load (ul) is understood to be 40µA I_{IH} and −1.6mA I_{IL}, and a 74LS unit load (LSul) is 20µA I_{IH} and −0.4mA I_{IL}.

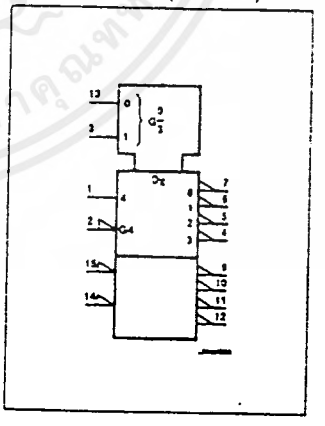
PIN CONFIGURATION



LOGIC SYMBOL

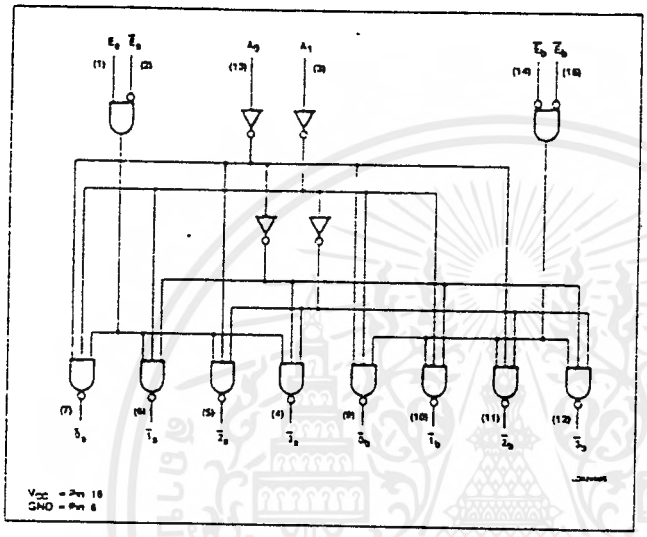


LOGIC SYMBOL (IEEE/IEC)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOGIC DIAGRAM



Both decoder sections have a 2-input enable gate. For decoder "a" the enable gate requires one active-HIGH input and one active-LOW input ($E_a \cdot \overline{E}_a$). Decoder "a" can accept either true or complemented data in demultiplexing applications, by using the $\overline{E}_a$ or E_a inputs respectively. The decoder "b" enable gate requires two active-LOW inputs ($\overline{E}_b \cdot \overline{E}_b$). The device can be used as a 1-of-8 decoder/demultiplexer by tying E_a to $\overline{E}_b$ and relabelling the common connection address as (A_2); forming the common enable by connecting the remaining $\overline{E}_b$ and $\overline{E}_a$.

The '156 can be used to generate all four minterms of two variables. The four minterms are useful to replace multiple gate functions in some applications. A further advantage of the '156 is being able to AND the minterm functions by tying outputs together. Any number of terms can be wired-AND as shown in the formula below:

$$f = (E + A_0 + A_1) \cdot (E + \overline{A}_0 + A_1) \cdot (E + A_0 + \overline{A}_1) \cdot (E + \overline{A}_0 + \overline{A}_1)$$

where $E = \overline{E}_a + E_a$; $E = E_b + E_b$.

FUNCTION TABLE

ADDRESS		ENABLE "a"		OUTPUT "a"				ENABLE "b"		OUTPUT "b"			
A_0	A_1	E_a	$\overline{E}_a$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{E}_b$	$\overline{E}_b$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
X	X	L	X	H	H	H	H	H	X	H	H	H	H
X	X	X	H	H	H	H	H	X	H	H	H	H	H
L	L	H	L	L	H	H	H	L	L	L	H	H	H
H	L	H	L	H	L	H	H	L	L	H	L	H	H
L	H	H	L	H	H	L	H	L	L	H	H	L	H
H	H	H	L	H	H	H	L	L	L	H	H	H	L

H = HIGH voltage level
L = LOW voltage level
X = Don't care

ABSOLUTE MAXIMUM RATINGS (Over operating free-air temperature range unless otherwise noted.)

PARAMETER		74	74LS	UNIT
V_{CC}	Supply voltage	7.0	7.0	V
V_{IN}	Input voltage	−0.5 to +5.5	−0.5 to +7.0	V
I_{IN}	Input current	−30 to +5	−30 to +1	mA
V_{OUT}	Voltage applied to output in HIGH output state	−0.5 to +V_{CC}	−0.5 to +V_{CC}	V
T_A	Operating free-air temperature range	0 to 70		°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

รายละเอียดอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ที่ใช้ในการสร้างชุด 8088 ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยว

รีซิสเตอร์

1. 10 โอห์ม	2 ตัว
2. 100 โอห์ม	1 ตัว
3. 270 โอห์ม	2 ตัว
4. 330 โอห์ม	4 ตัว
5. 1 กิโลโอห์ม	12 ตัว
6. 2 กิโลโอห์ม	8 ตัว
7. 1.8 กิโลโอห์ม	1 ตัว
8. 4.7 กิโลโอห์ม	4 ตัว
9. 10 กิโลโอห์ม	7 ตัว
10. 15 กิโลโอห์ม	1 ตัว
11. 47 กิโลโอห์ม	1 ตัว
12. รีจิสเตอร์แพค	1 ตัว

ตัวเก็บประจุ

1. 0.1 uF	30 ตัว
2. 30 pF	1 ตัว
3. 100 pF	1 ตัว
4. 10 uF 16 V	2 ตัว
5. 33 uF 16 V	2 ตัว
6. 470 pF	1 ตัว
7. 4.7 uF 16 V	1 ตัว
8. 220 uF	1 ตัว
9. 33 uF	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานซิสเตอร์

1. BC 547	1 ตัว
2. BC 557	1 ตัว
3. 2SC458	2 ตัว

ไอซี

1. 8088 CPU	1 ตัว
2. 74HCT04	1 ตัว
3. 74LS32	1 ตัว
4. 74LS123	1 ตัว
5. 74LS138	1 ตัว
6. 74LS156	1 ตัว
7. 74LS175	1 ตัว
8. 74LS251	1 ตัว
9. 74LS259	1 ตัว
10. 74LS541	2 ตัว
11. 74LS377	2 ตัว
12. 74LS373	1 ตัว
13. ADC 0804	1 ตัว
14. CD 4051	1 ตัว
15. ULN 2003A	1 ตัว
16. 74HC4020	1 ตัว
17. 8255	1 ตัว
18. 74LS145	1 ตัว
19. LM386	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอลอีดี

1. แอลอีดีสีเขียวตัวเล็ก 1 ตัว
2. แอลอีดีสีแดง 17 ตัว

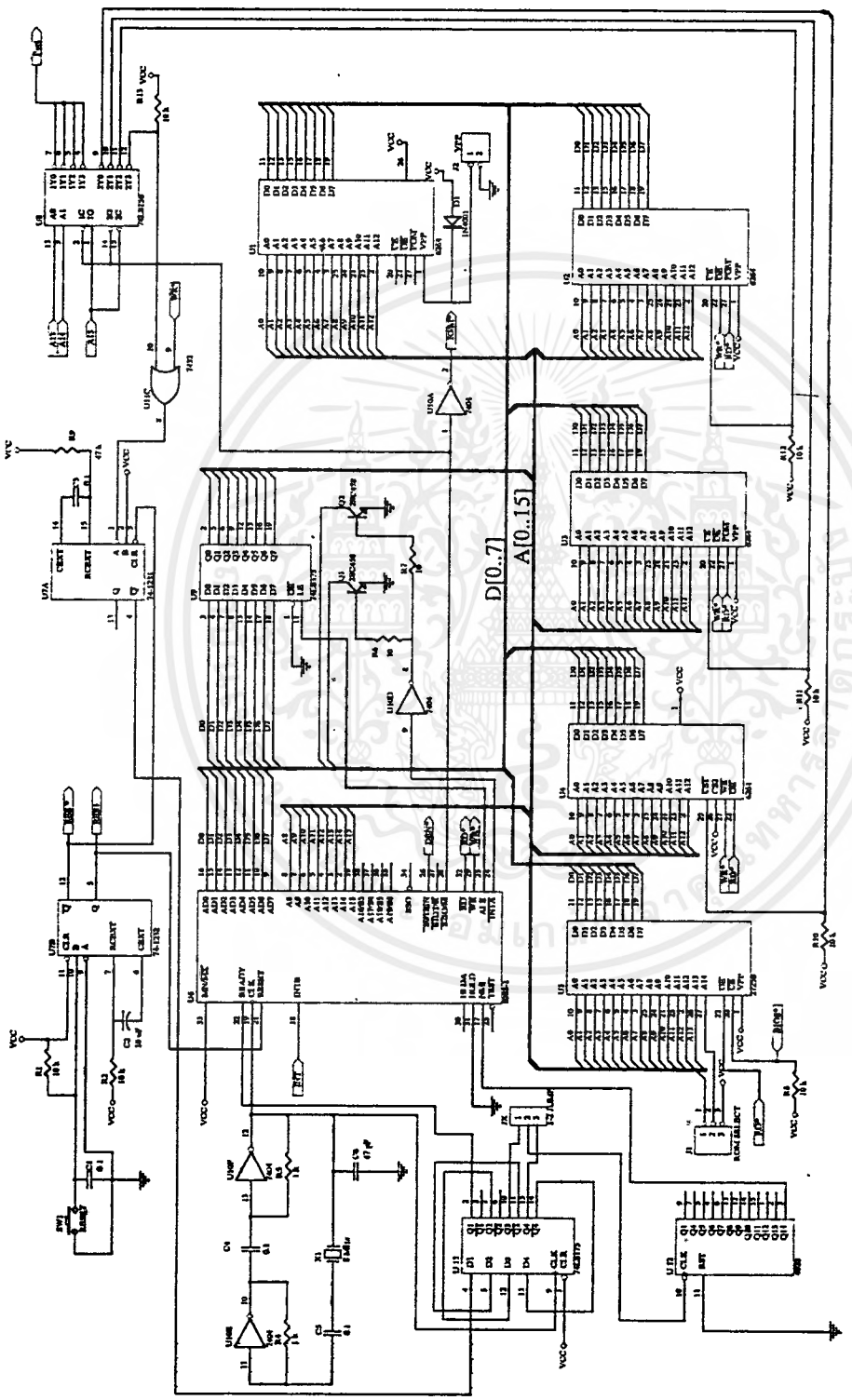
เบ็ดเตล็ด

1. MINI SW 1 ตัว
2. XTAL 8 MHz 1 ตัว
3. JUMPER 3 ตัว
4. คอนเน็กเตอร์ แบบ 10 พิน 1 ตัว
   แบบ 14 พิน 1 ตัว
   แบบ 20 พิน 3 ตัว
   แบบ 34 พิน 2 ตัว
   แบบ 40 พิน 1 ตัว
5. ลำโพง 0.25 W 1 ตัว
6. คียบอร์ดขนาด 6 x 4 1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

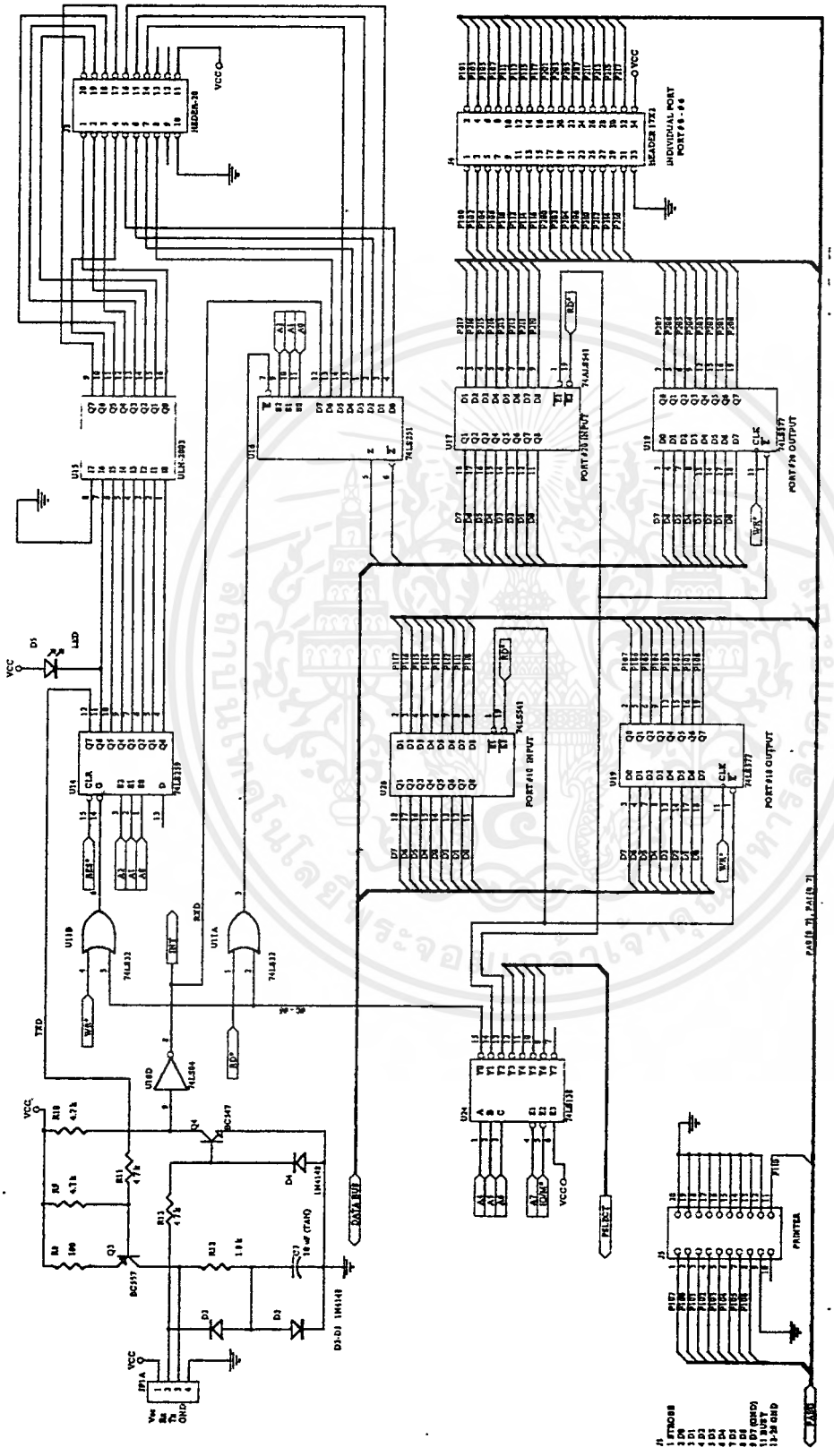
ภาคผนวก ฉ
วงจรรวมไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088 และแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



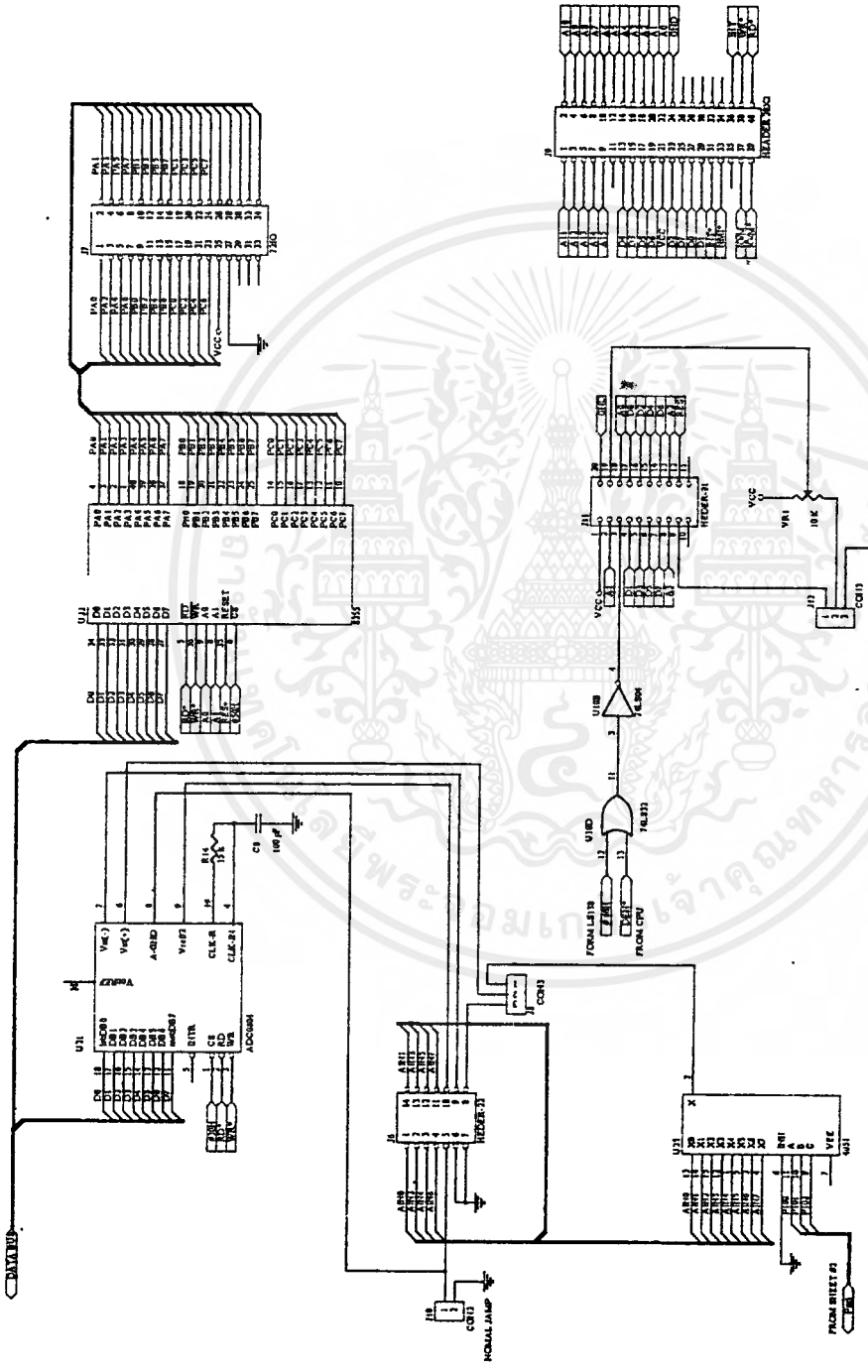
รูปที่ ก.1 วงจรไมโครคอมพิวเตอร์แบบแผงพิมพ์เดียวเบอร์ 8088 สามารถ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



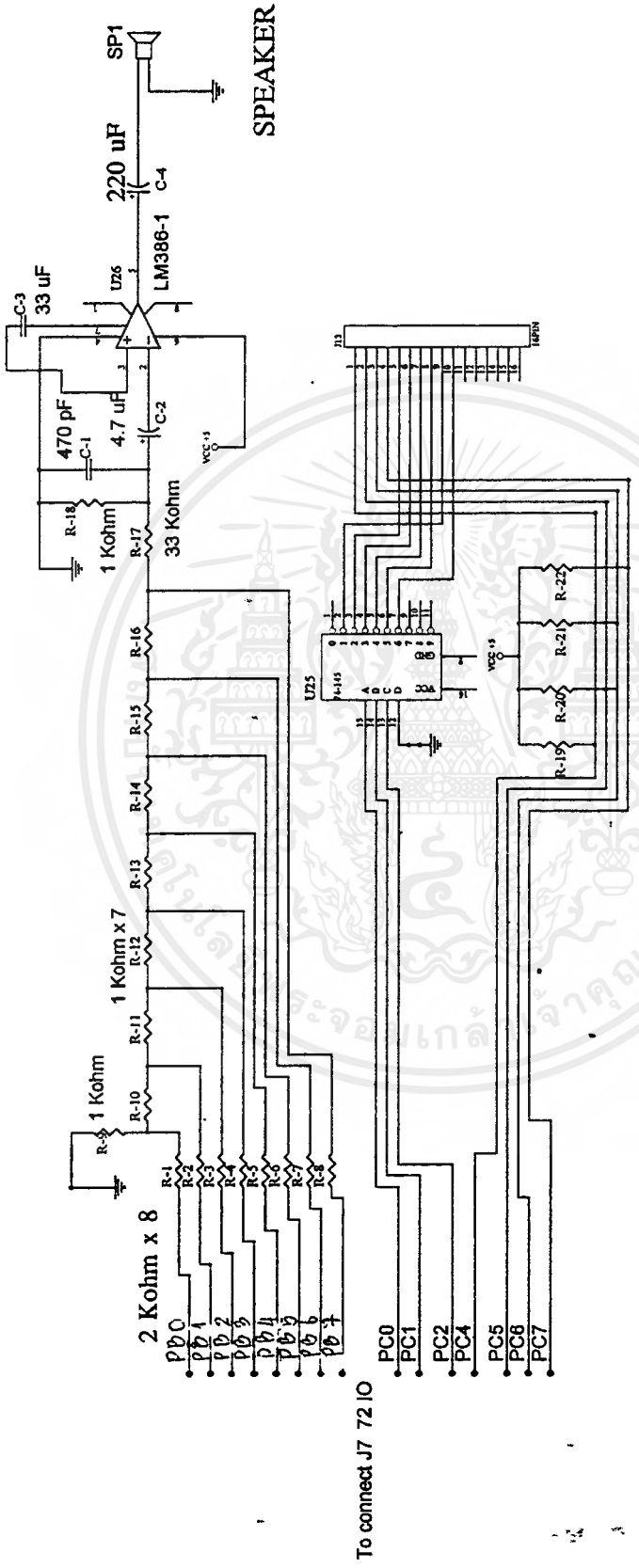
รูปที่ ก.2 วงจรไมโครคอมพิวเตอร์แบบแผงพิมพ์ตัวเบอร์ 8088 ส่วนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



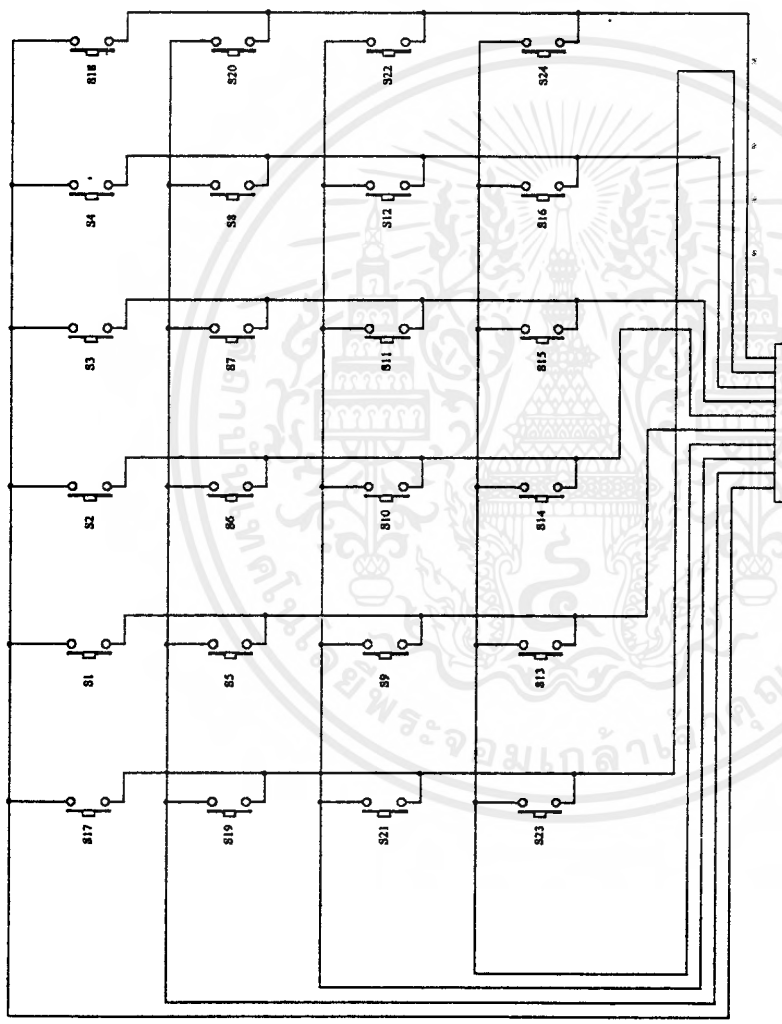
รูปที่ ฉ.3 วงจรไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดียวเบอร์ 8088 ส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



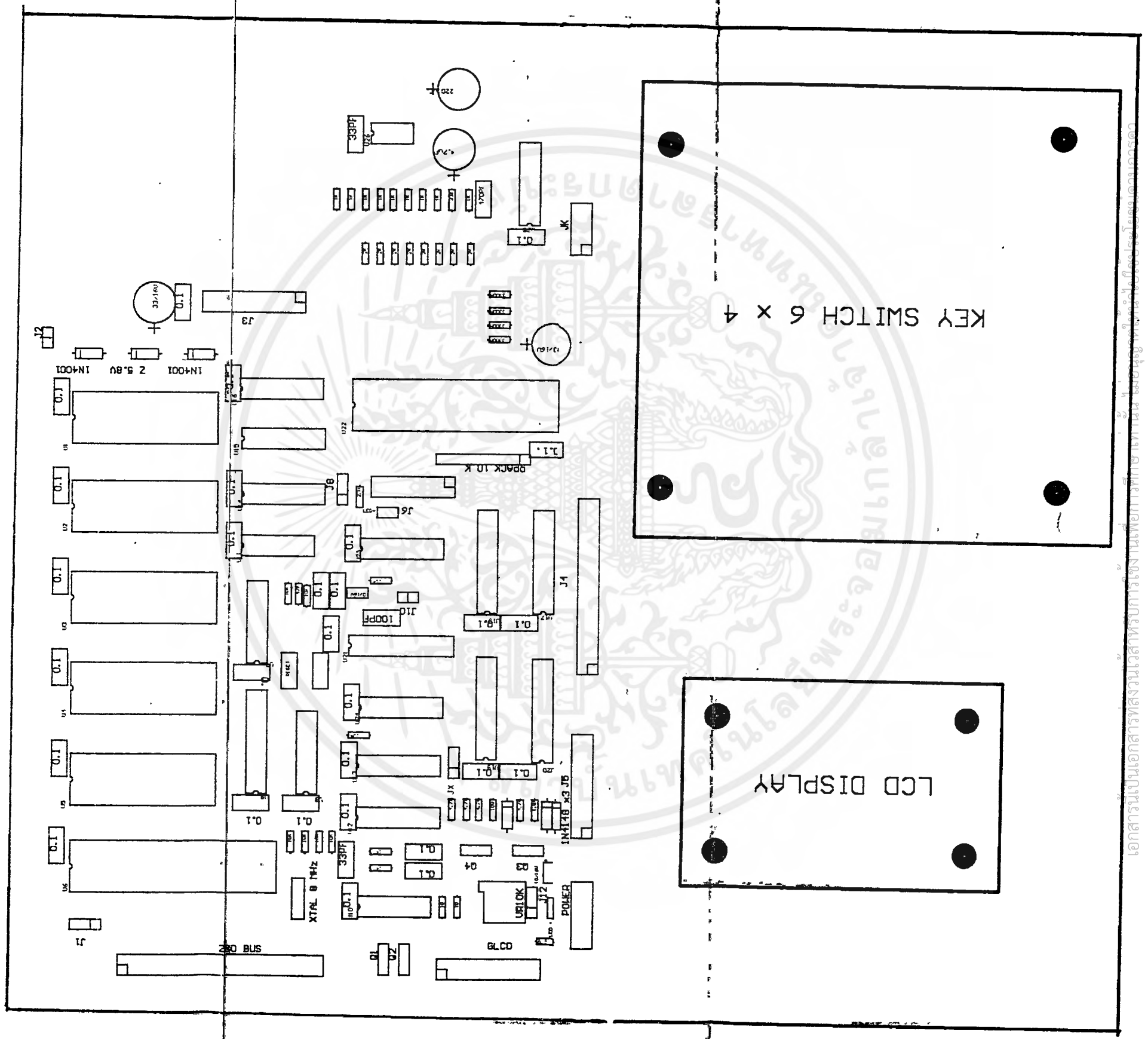
รูปที่ ฉ.4 วงจรไมโครคอมพิวเตอร์แบบแผงพิมพ์เดี่ยวเบอร์ 8088 ส่วนที่ 4

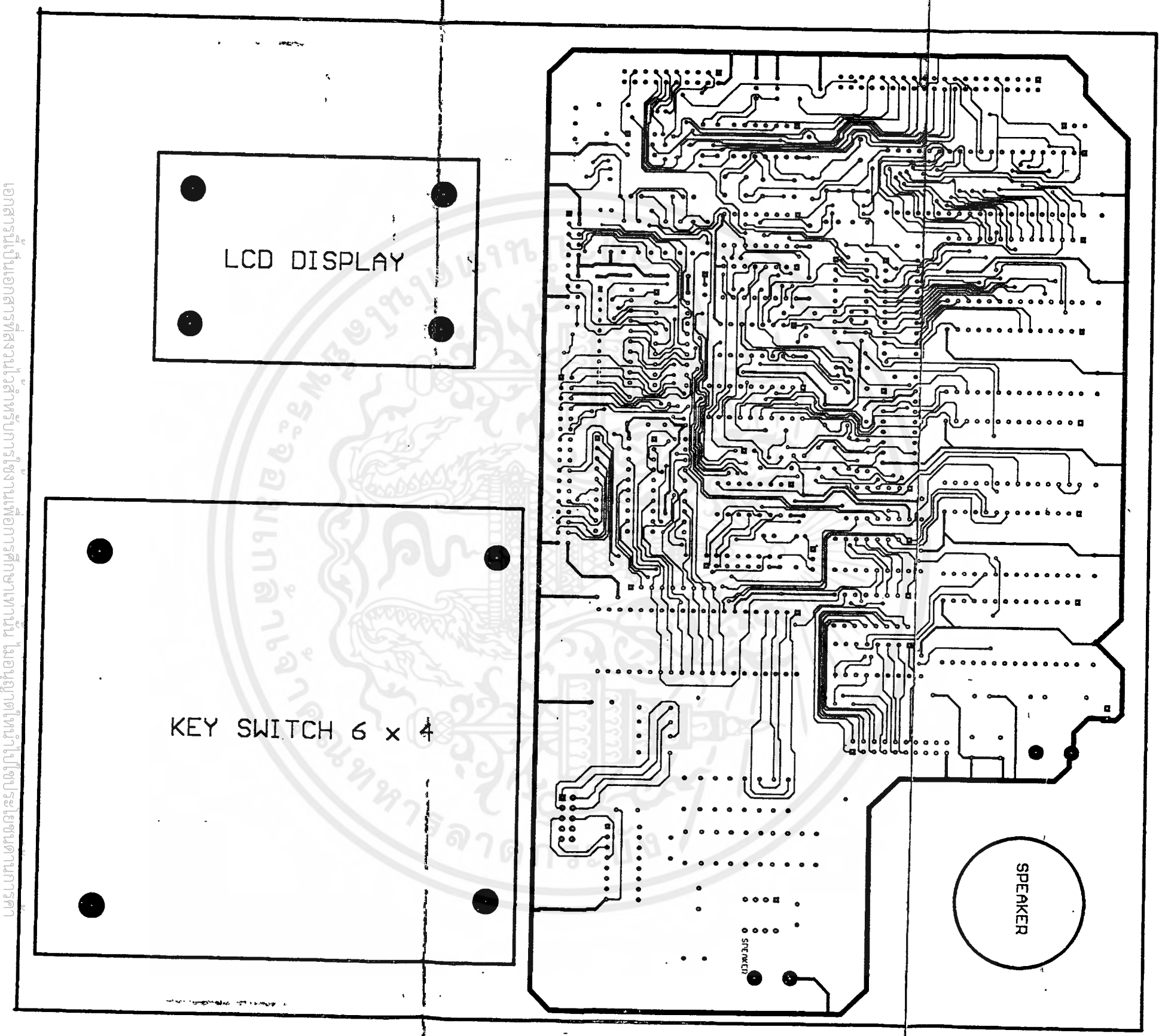
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



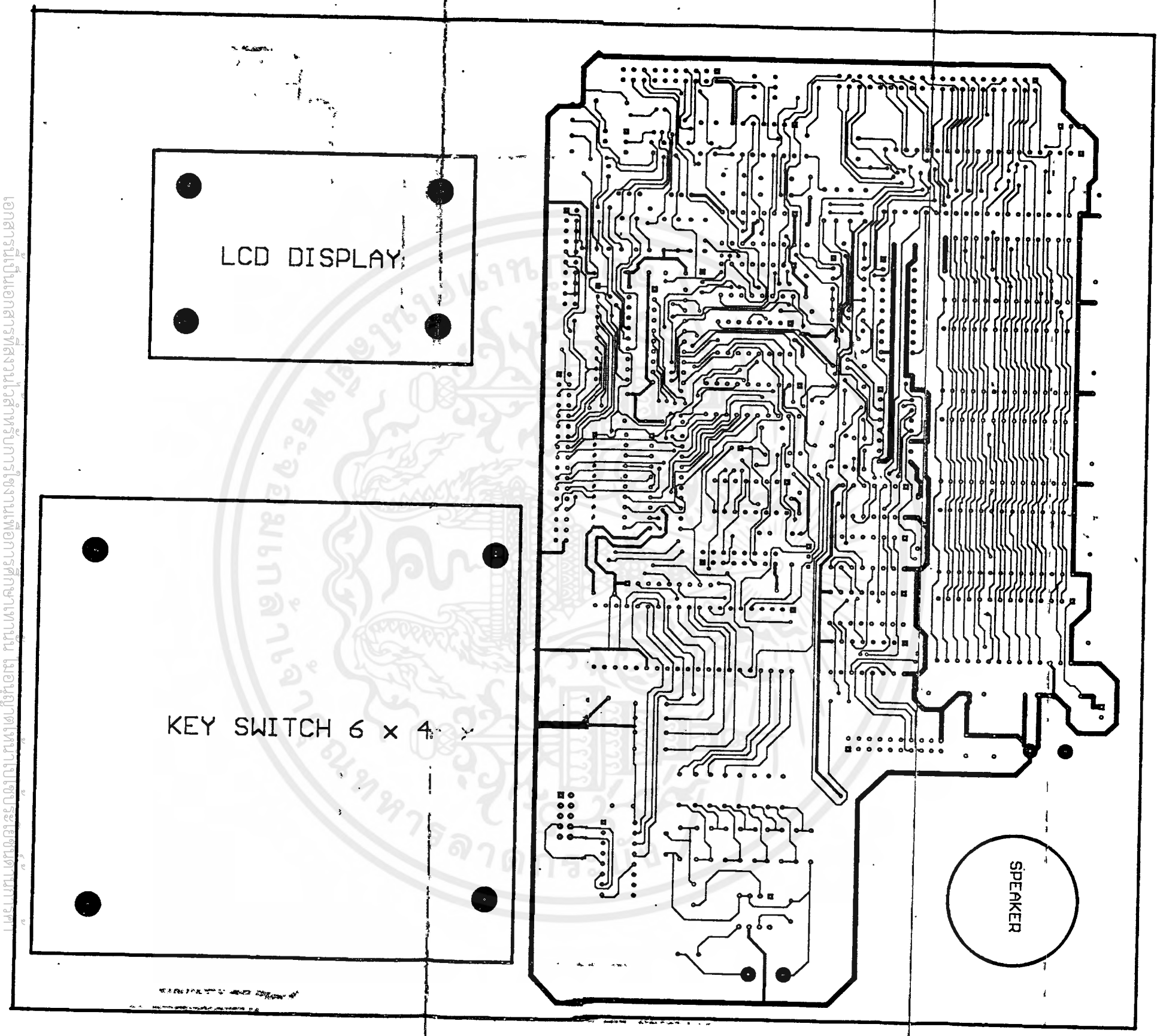
รูปที่ ผ.5 วงจรไมโครคอมพิวเตอร์แบบแผงวงจรพิมพ์เดี่ยวเบอร์ 8088 ส่วนที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


KEY SWITCH 6 x 4
LCD DISPLAY
220
33PF
100K
0.1
JK
J3
1N4001
Z 5.8V
1N4001
1.0
J22
3.1.
DPACK 10 K
J8
J6
J4
J10
100PF
J11
1N4148 x3
J5
XTAL 8 MHz
33PF
Q4
Q3
VR10K
J12
POWER
J7
280 BUS
Q1
Q2
GLCD



รูปที่ ฉ.7 วงจรพิมพ์ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088
ส่วนที่ 2 ลายวงจรพิมพ์ด้านบน



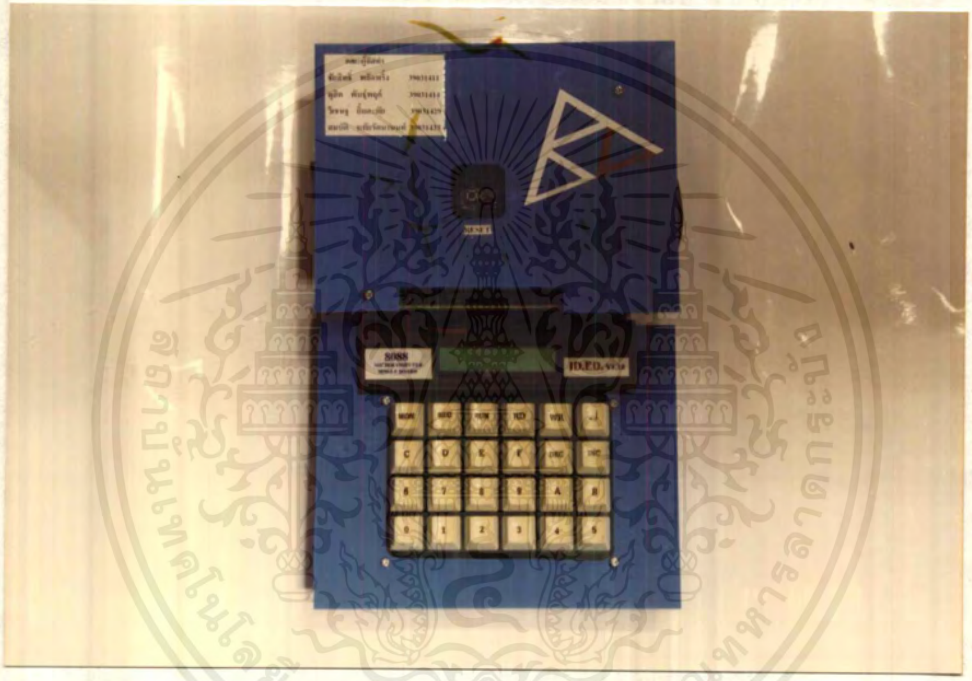
รูปที่ ฉ.8 วงจรพิมพ์ไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

ส่วนที่ 2 ลายวงจรพิมพ์ด้านล่าง

ภาคผนวก ช

รูปเครื่องต้นแบบไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า. ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ช.1 เครื่องต้นแบบไมโครคอมพิวเตอร์แบบแผ่นพิมพ์เดี่ยวเบอร์ 8088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ธานินทร์ ถาวรศาสนวงศ์ , การอินเทอร์เฟส , พิมพ์ครั้งที่ 1 , ห้างหุ้นส่วนจำกัด สำนักพิมพ์ฟิสิกส์เซ็นเตอร์ , มีนาคม 2536 .

ยืน ภู่วรวรรณ และ สุรศักดิ์ สงวนพงษ์ , โปรแกรมคอมพิวเตอร์ภาษาแอสเซมบลี้ 8086/8088 , พิมพ์ครั้งที่ 1 , บริษัท ซีเอ็ดยูเคชั่น จำกัด , ห้างหุ้นส่วนจำกัด เอช-เอน การพิมพ์ , 2535

BARRY B. BREY ; THE INTEL MICROPROCESSORS , PRENTICE HALL INTERNATIONAL , PRENTICE HALL ENGLEWOOD CLIFFS , 1995 .

SINGH AVTAR and WALTER A. TRIEBEL ; THE 8086/88 AND 80286 MICROPROCESSORS HARDWARE - SOFTWARE AND INTERFACING , 4 , . ENGLEWOOD CLIFFS , N.J ∴ PRENTICE HALL , 1995 .

SINGH AVTAR and WALTER TRIEBEL , THE 8088 MICROPROCESSOR : PROGRAMMING , INTERFACINH , SOFTWARE , HARE WARE and APPLICATIONS . ENGLEWOOD CLIFFS , N.J ∴ PRENTICE HALL , 1989 .

VITRO P.NELSON H. TROY NAGLE BILL D. CARROLL and J.DAVID IRWIN , DIGITAL CIRCUIT ANALYSIS and DESIGN . . ENGLEWOOD CLIFFS , N.J ∴ PRENTICE HALL , 1995 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นาย ชัยสิทธิ์ พลัดพลิ้ง
วันเดือนปีเกิด	10 สิงหาคม 2519
สถานที่เกิด	จังหวัด กรุงเทพฯ
ภูมิลำเนาเดิม	จังหวัด กรุงเทพฯ
ที่อยู่ปัจจุบัน	81 หมู่ 8 ซอยโชคดี ถนนบางแวก
	แขวงบางไผ่ เขตภาษีเจริญ กรุงเทพฯ 10160
โทรศัพท์	4103712

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนตรีมิตรวิทยา
มัธยมศึกษาตอนต้น	โรงเรียนวัดนายโรง
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตนนทบุรี
ปริญญาตรี	อิเล็กทรอนิกส์และคอมพิวเตอร์
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ทำให้ได้ ทำให้ดี ทำให้เร็ว ทำให้ถูก ทำให้เสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นาย ดุสิต พันธ์พฤกษ์
วันเดือนปีเกิด	11 ตุลาคม 2518
สถานที่เกิด	จังหวัด ตาก
ภูมิลำเนาเดิม	จังหวัด ยะลา
ที่อยู่ปัจจุบัน	87 /1 ถนนวิฑูรอุทิศ 10 อ.เมือง จ.ยะลา 95000
โทรศัพท์	073 - 211636

ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนนิบงชนูปถัม
มัธยมศึกษาตอนต้น	โรงเรียนผดุงประชา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคยะลา
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคยะลา

ปริญญาตรี	อิเล็กทรอนิกส์และคอมพิวเตอร์
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	นอนน้อยอิ่มมาก นอนมากอิ่มน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นาย วิเชษฐ ยิ้มละมัย
วันเดือนปีเกิด	10 ธันวาคม 2519
สถานที่เกิด	จังหวัด พระนครศรีอยุธยา
ภูมิลำเนาเดิม	จังหวัด พระนครศรีอยุธยา
ที่อยู่ปัจจุบัน	266/1 หมู่ 4 ตำบล ผักไห่ อำเภอ ผักไห่ จังหวัด พระนครศรีอยุธยา
โทรศัพท์	035 - 239076

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนวัดโพธิ์ผักไห่วิทยาการ
มัธยมศึกษาตอนต้น	โรงเรียนวัดโพธิ์ผักไห่ (เวชพันธุ์อนุสรณ์)
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคพระนครศรีอยุธยา
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคพระนครศรีอยุธยา
ปริญญาตรี	อิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ไม่มีความสำเร็จใดที่ได้มาด้วยความสบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นาย สมบัติ หทัยรัตนานนท์
วันเดือนปีเกิด	24 มกราคม 2519
สถานที่เกิด	จังหวัด สมุทรสาคร
ภูมิลำเนาเดิม	จังหวัด สมุทรสาคร
ที่อยู่ปัจจุบัน	18 หมู่ 1 ตำบลหนองสองห้อง อำเภอบ้านแพ้ว จังหวัดสมุทรสาคร 70210
โทรศัพท์	034 - 481412

ประวัติการศึกษา

ประถมศึกษา	ตรงจิตวิทยา
มัธยมศึกษาตอนต้น	ประสาทรัฐประชากิจ (จังหวัดราชบุรี)
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคราชบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตเทคนิคกรุงเทพฯ
ปริญญาตรี	อิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	**จงทำตนให้เป็นวัตถุทรงกลม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้