REMOTE CONTROL SCOREBOARD DISPLAY

นายบรรจบ วัฒนสุนทร
นายพยัญ บุญญสุวรรณ

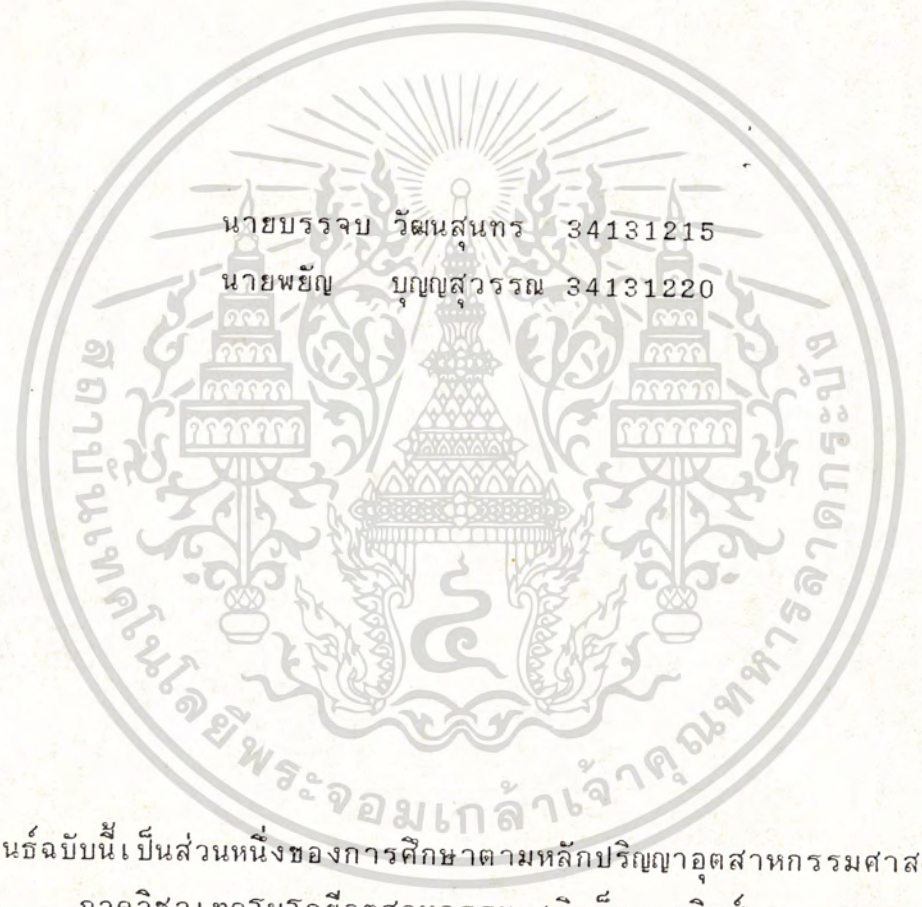
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักปริญญาอุตสาหกรรมศาสตร์บัณฑิต
ภาควิชาเทคโนโลยีอุตสาหกรรม (อิเล็กทรอนิกส์)
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
033222

REMOTE CONTROL SCOREBOARD DISPLAY

นายบรรจบ วัฒนสุนทร 34131215
นายพยัญ บุญญสุวรรณ 34131220



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักปริญญาอุตสาหกรรมศาสตร์บัณฑิต
ภาควิชาเทคโนโลยีอุตสาหกรรม (อิเล็กทรอนิกส์)
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REMOTE CONTROL SCOREBOARD DISPLAY

Mr. BANCHOB WATTANASUNTHORN 34131215
Mr. PAYAN BOONYASUVAN 34131220

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE BACHELOR'S DEGREE
DEPARTMENT OF INDUSTRIAL TECHNOLOGY
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ REMOTE CONTROL SCOREBOARD DISPAY

ชื่อนักศึกษา นาย บรรจบ วัฒนุนทร
นาย พยัญ ุ บุญญสุวรรณ

อาจารย์ที่ปรึกษา อาจารย์ ประดิษฐ์ วัชรพิบูลย์

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2536

ภาคเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของ การศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....อาจารย์ที่ปรึกษา
(.....)

.....กรรมการ
(.....)

.....กรรมการ
(.....)

.....กรรมการ
(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Report Remote Control Scoreboard Display

By Mr. Banchob Wattanasunthorn
Mr. Payan Boonyasuvan

Department Industrial Technology

Project Report Advisor Mr. Pradit Vachcapibool

Accepted by the Faculty of Engineering, King mongkut's Institute of Technology, Ladgrabang in partial Fulfillment of the requirements for the bachelor's degree

Project Report Committee

.....Member
(.....)

.....Member
(.....)

.....Member
(.....)

.....Member
(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบาญ

	เรื่อง	หน้า
	บทคัดย่อ	ก
	กิตติกรรมประกาศ	ข
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีเกี่ยวกับไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์	2
บทที่ 3	บล็อกไดอะแกรมภาพรวมของระบบ	52
	- ชุด REMOTE	
	- ชุด DECODER	
	- ชุด CONTROL	
	- ชุด DISPLAY	
	- วงจร	
	- วิธีการใช้เครื่อง	
	- บทสรุป	
	- โปรแกรมการใช้งาน	
ภาคผนวก		
	- หนังสืออ้างอิง	
	- ข้อมูล IC SUPPORT ที่ใช้งาน	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REMOTE CONTROL SCOREBOARD DISPLAY

บทคัดย่อ

REMOTE CONTROL SCOREBOARD DISPLAY เป็นอุปกรณ์ชุดควบคุมซึ่งมีหน้าที่ควบคุมการนำแสดงภาพข้อความตัวเลขในระยะไกล สามารถเลือกข้อความหรือตัวเลขได้จากชุด REMOTE โดยนำเอาระบบ DTMF มาใช้ในการส่งข้อมูลจากชุด REMOTE ให้กับชุดควบคุมการนำแสดงผลซึ่งใช้สายส่งสัญญาณเพียง 2 เส้น รูปแบบของสัญญาณที่รับมาจากชุด REMOTE จะถูกประมวลผลแล้วนำไปแสดงที่ DISPLAY ด้วยไมโครโปรเซสเซอร์ (Z 80)

หลักการทำงานของเครื่อง REMOTE CONTROL SCOREBOARD DISPLAY

- ชุด REMOTE ใช้คีย์บอร์ดของโทรศัพท์ในการป้อนข้อมูลไปควบคุมสั่งงานแก่ไมโครโปรเซสเซอร์ในการนำข้อมูลออกแสดงผล
- ใช้รหัสที่ชุด REMOTE ในการเลือกข้อความนำมาแสดงผล
- นำเอาระบบ DTMF มาใช้ในการส่งข้อมูล
- ใช้ภาษาแอสแซมบลี้เป็นโปรแกรมควบคุมและโปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REMOTE CONTROL SCOREBOARD DISPLAY

ABSTRACT

REMOTE CONTROL SCOREBOARD DISPLAY IS THE EQUIPMENT CONTROL SYSTEM. USED REMOTE CONTROL NUMERAL OR PICTURE OF MESSAGE ON THE SCREEN DISPLAY. CAN SELECTION THE PICTURE OF MESSAGE OR NUMERAL FROM REMOTE BY MEANS OF DTMF SYSTEM FOR TRANSMISSION FROM REMOTE TO CONTROL SYSTEM. THAT USED TWO TRANSMISSION LINE. THE SIGNAL PATTERN RECEIVE FROM REMOTE THAT WILL PROCESSING. AND SHOW ON SCREEN DISPLAY BY MICROPROCESSER (Z 80)

THE FUNCTIONS OF REMOTE CONTROL SCOREBOARD DISPLAY

- THE REMOTE SYSTEM USED KEYBOARD OF DTMF FOR DATA TO CONTROL MICROPROCESSER FOR OUT PUT
- USE PASSWORD FROM REMOTE SYSTEM FOR SELECTION THE PICTURE OF MESSAGE ON THE SCREEN DISPLAY
- USE DTMF SYSTEM FOR DATA TRANSMISSIO
- USE ASSEMBLY LANGUAGE FOR PROGRAME

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเรื่อง Remote Control Scoreboard Display เป็นโครงงานที่นำเสนอต่อ ภาควิชาเทคนิคอุตสาหกรรม สาขา อีเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรคณะวิศวกรรมศาสตร์บัณฑิต โครงงานนี้สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากคณะผู้จัดทำได้รับความช่วยเหลือและข้อแนะนำจากอาจารย์ ประดิษฐ์ วัชรพิบูลย์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน คณะผู้จัดทำขอขอบคุณบริษัท ETT ที่แนะนำการอัด program ใน EPROM ให้ อีกทั้ง คุณฉัตรชัย กลิ่นแย้ม ที่ให้คำแนะนำการต่อวงจร Hardware และคณะผู้จัดทำขอขอบคุณ คุณพตล บัวสาย ที่ช่วยออกแบบกล่อง Display, น้องสาวทุกคนที่ช่วยพิมพ์รายงาน ให้โครงงานนี้สำเร็จตามวัตถุประสงค์ที่ต้องการ

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ในปัจจุบันการพัฒนาทางด้านสื่อสารข้อมูลเป็นไปอย่างรวดเร็วมากต่างๆ แล้วนี้ทำให้เกิดระบบสื่อสารข้อมูลที่ดีขึ้นตามลำดับ ระบบ DTMF. ก็เป็นระบบหนึ่งที่สามารถส่งข้อมูลได้ในระยะไกล

REMOTE CONTROL SCOREBOARD DISPLAY นำเอาระบบ DTMF. มาใช้ในการควบคุมเพื่อให้คล่องตัว ในการสั่งงานในระยะไกล เช่น ในสนามกีฬา ป้ายโฆษณา สถานีปั๊มน้ำ ห้องที่มีสารเคมี ระบบควบคุมในโรงงานอุตสาหกรรม

<u>คุณสมบัติที่สำคัญของเครื่อง</u>

1. สามารถป้อนข้อมูลให้กับ CPU ในระยะไกลได้โดยผ่านทางคีย์บอร์ดโทรศัพท์
2. ง่ายต่อการติดตั้งเพราะใช้สายส่งสัญญาณเพียง 2 เส้น
3. ประหยัดค่าใช้จ่าย

<u>จุดมุ่งหมายการทำงาน</u>

1. เพื่อนำ keyboard โทรศัพท์ (DTMF.) ")ใช้งานด้าน control
2. เพื่อศึกษาความเป็นไปได้ในการส่งข้อมูลในระยะไกลโดยรวมสัญญาณและไฟไปด้วยกันซึ่งใช้สายส่งสัญญาณ 2 เส้น
3. เพื่อนำความรู้เรื่องโปรแกรมมาประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเกี่ยวกับไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์

ในปัจจุบันนี้ CPU ขนาด 8 บิต ที่เป็นที่นิยมใช้กันอยู่ เป็นเบอร์ Z80 ที่บริษัทไซลอก (Zilog) เป็นผู้พัฒนาขึ้นมา

ในสมัยแรกไมโครคอมพิวเตอร์เริ่มต้นจาก CPU เบอร์ 4004 ขนาด 4 บิตของบริษัทอินเทล (Intel) หลังจากนั้นก็เกิดการเปลี่ยนแปลงขึ้นโดยบริษัทอินเทล ได้ผลิต CPU เบอร์ 8080A ขนาด 8 บิตออกมา

ต่อมาบริษัทไซลอกได้ผลิต CPU เบอร์ Z80 ขนาด 8 บิตออกมาครอบคลุมถึงคำสั่งทั้งหมดที่ 8080A ไม่มีเพิ่มเติมเข้ามา ส่วนทางด้านฮาร์ดแวร์ของ Z80 ก็มีข้อดีอยู่หลายข้อเช่น มีวงจรรีเฟรชไดนามิค RAM อยู่ในตัว ทำให้การสร้างระบบใหญ่ ๆ ที่ใช้หน่วยความจำมาก ๆ สามารถทำได้โดยง่าย ความเร็วในการปฏิบัติงานของคำสั่งก็เร็วกว่าและความสามารถของการอินเตอร์รัพท์มีมากกว่าด้วยเหตุผลเหล่านี้จึงทำให้ Z80 เป็น CPU มาตรฐานในกระบวน 8 บิตด้วยกันไปโดยปริยาย

อินเตอร์เฟสเข้าเป็นระบบ

ระบบไมโครคอมพิวเตอร์ไม่ใช่ว่า CPU อย่างเดียว แต่จะประกอบไปด้วย RAM ROM และอุปกรณ์อินพุท-เอาต์พุทต่าง ๆ ซึ่งเป็นไอซีประเภท LAI ทำหน้าที่อินเตอร์เฟสกับ CPU นอกจากนี้ยังขึ้นอยู่กับการนำไปใช้งาน ซึ่งอาจจะต้องการส่วนตั้งเวลา [TIMER] หรือส่วนของการควบคุมการอินเตอร์รัพท์และส่วนอินเตอร์เฟสอื่น ๆ อีกก็ได้

ไอซี LAI สนับสนุนเหล่านี้ แต่ละบริษัทต่างก็ผลิตออกมาให้มีความสามารถต่าง ๆ กัน โดยปกติแล้ว แต่ละบริษัทจะผลิตออกมาให้สอดคล้องกับ CPU เป็นตัว ๆ ไป ทำให้เกิดเป็นตระกูลของ CPU เบอร์นั้น ๆ ขึ้นมา

เป็นเรื่องธรรมดาที่ไอซี LSI สนับสนุนเบอร์นั้น ๆ จะใช้ได้ผลเต็มที่ก็ต่อเมื่อนำไปใช้ร่วมกับตระกูลของมันเอง แต่ทว่าการนำไปใช้เชื่อมต่อกับต่างตระกูลนั้นก็ไม่มีปัญหาอะไร โดยเฉพาะ CPU เบอร์ Z80 นี้ สามารถเชื่อมต่อเข้ากับไอซี LAI สนับสนุนขนาด 8 บิตของบริษัทอินเทล ได้โดยไม่มีปัญหา

เนื่องจาก CPU เบอร์ 80805A ของบริษัทอินเทลนี้มีไอซีLSIสนับสนุนอยู่มากมายจึงไม่มีการนำเอาไอซีLSIสนับสนุนของตระกูลอื่นมาใช้กับ 80805A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครโปรเซสเซอร์เทคโนโลยี

เทคโนโลยีของแอลเอสไอ (LARGE SCALE INTEGRATION) ทำให้เราสามารถบรรจุวงจรอิเลคทรอนิกส์ลงไปบนแผ่นเดียวได้ เรียกแผ่นวงจรนั้นว่า ชิพ (CHIP) เทคโนโลยี LSI นี้ถูกนำมาในการผลิตไมโครโปรเซสเซอร์ที่จำหน่ายอยู่ในท้องตลาด โดยทั่วไปแบ่งออกตามตระกูลหรือเทคโนโลยีที่นำมาได้ดังต่อไปนี้

1. พวกไบโพลา พวกนี้ได้แก่ลอจิกประกอบด้วยทรานซิสเตอร์ ซึ่งทำงานได้ด้วยกระแสทั้งสองประเภทคือลบและบวกจึงเรียกว่าไบโพล่า (BIPOLAR) ตระกูลนี้แบ่งพวกย่อย ๆ ตามลักษณะวงจรคือ

1.1 ECL (EMITTER COUPLING LOGIC)

1.2 IIL (INTEGRATED INJECTION LOGIC)

1.3 TTL (TRANSISTOR TRANSISTOR LOGIC)

1.4 STTL (SCHOTTKY TRANSISTOR TRANSISTOR LOGIC)

ทั้งสี่พวกนี้มีความแตกต่างกันเรื่องความเร็ว ภายในตระกูลไบโพล่าเองความเร็วก็แตกต่างกัน แต่โดยรวมแล้วตระกูลนี้มีความเร็วมากกว่าตระกูลอื่น

2. พวก MOS (METAL OXIDE SEMICONDUCTOR) พวกนี้ลอจิกเป็นทรานซิสเตอร์เช่นกัน แต่เป็นทรานซิสเตอร์ชนิด MOSFET แบ่งออกเป็นพวกตามลักษณะวงจรและสารที่ทำคือ

2.1 PMOS (P-CHANNEL METAL OXIDE SEMICONDUCTOR)

2.2 NMOS (N-CHANNEL METAL OXIDE SEMICONDUTCTOR)

2.3 CMOS (COMOLEMENTARY METAL OXIDE SEMICONDUCTOR)

พวก MOS มีคุณสมบัติเด่นมากเรื่องกินกำลังไฟต่ำ

ในการเลือกไมโครโปรเซสเซอร์ที่เหมาะสมสำหรับงานต่าง ๆ จะต้องพิจารณาข้อมูลให้ละเอียดจึงจะได้ระบบที่เหมาะกับงานมากที่สุด ข้อนี้ขึ้นอยู่กับจำกัดในการเลือกใช้เอง นอกจากเทคโนโลยีต่าง ๆ แล้วโครงสร้างภายในไมโครโปรเซสเซอร์ของแต่ละเบอร์ก็แตกต่างกันไป โครงสร้างภายในหมายถึงขนาดของข้อมูล จำนวณคำสั่ง การตอบรับสัญญาณจากภายนอก ทั้งนี้จะต้องศึกษารายละเอียดของแต่ละเบอร์ต่าง ๆ กัน

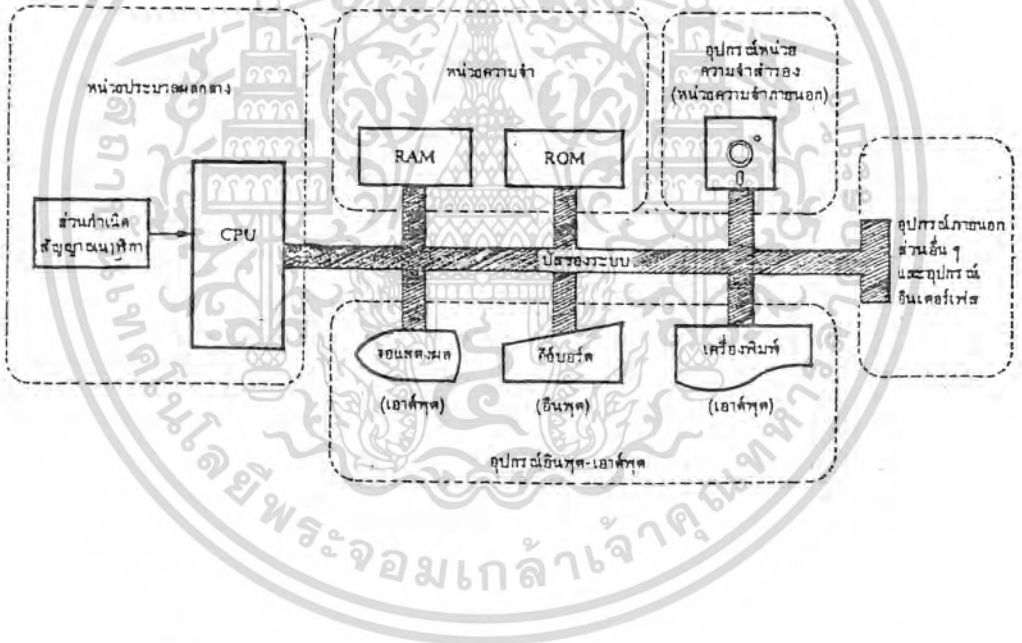
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบของไมโครคอมพิวเตอร์

องค์ประกอบพื้นฐานของโครงสร้างของระบบไมโครคอมพิวเตอร์ แสดงได้ดังรูปที่ 1.1 แบ่งออกเป็นส่วนต่าง ๆ ได้ดังนี้

CPU (หน่วยประมวลผลกลาง)

CPU (CENTRAL PROCEESING UNIT) คือ ส่วนกลางที่สำคัญทำหน้าที่ควบคุมการทำงานต่าง ๆ ของระบบตามโปรแกรมที่เขียนไว้ CPU ยังมีผลในการบ่งบอกถึงชนิดของไมโครคอมพิวเตอร์

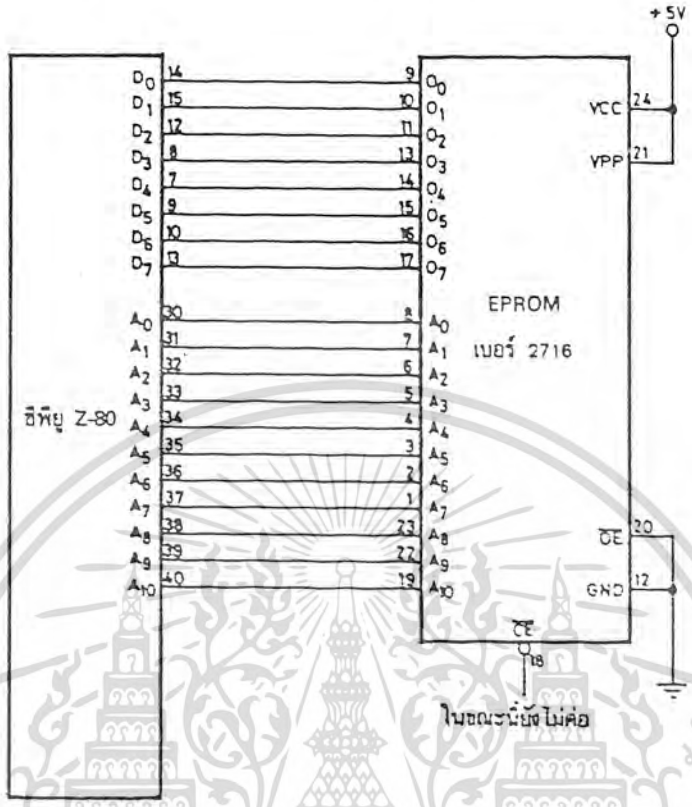


รูปที่ 1.1 แผนผังแสดงพื้นฐานของระบบไมโครคอมพิวเตอร์

หน่วยความจำคือส่วนที่ใช้สำหรับเก็บข้อมูลและโปรแกรมหน่วยความจำนับเป็นส่วนสำคัญที่จะขาดไม่ได้ในระบบของคอมพิวเตอร์ หน่วยความจำแบ่งออกได้เป็น 2 ชนิด คือ ROM และ RAM

1. ROM หรือ READ ONLY MEMORY คือ หน่วยความจำที่ใช้สำหรับเก็บโปรแกรมที่ควบคุมการทำงานของระบบ ถึงแม้จะปิดไฟเลี้ยวแล้วก็ตาม ข้อมูลที่อยู่ในหน่วยความจำ ROM นี้ ก็จะไม่หายไปด้วย ดังนั้นเมื่อเปิดไฟให้กับระบบใหม่ ระบบก็จะเริ่มทำงานตามโปรแกรมที่อยู่ใน ROM ทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



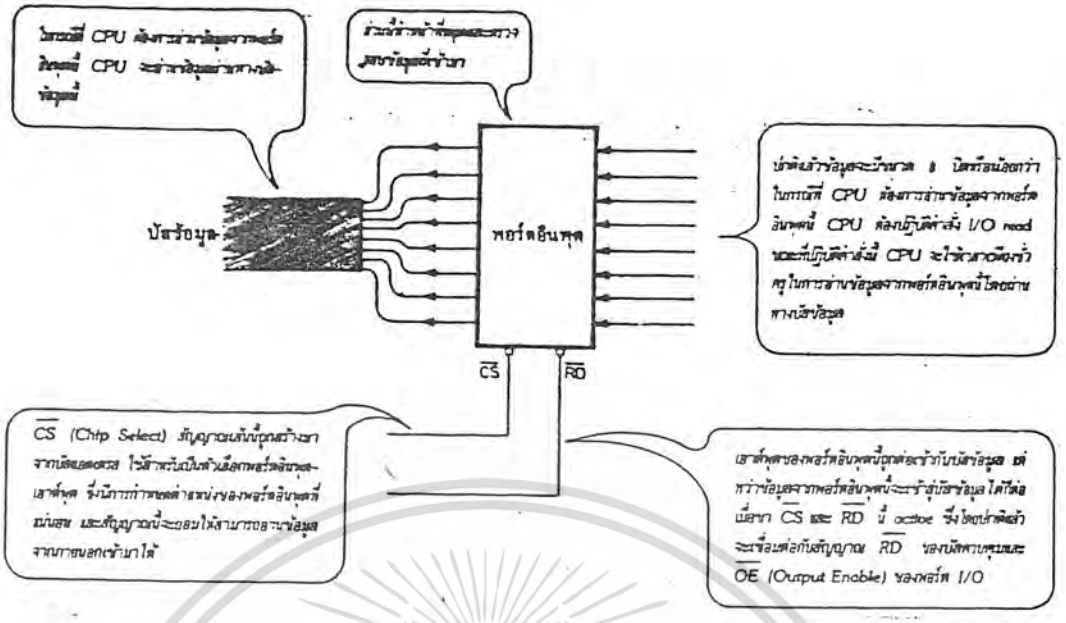
รูปที่ 1.2 แสดงการเชื่อมสาย CPU ต่อกับหน่วยความจำ

2. RAM หรือ RANDOM ACCESS MEMORY คือ หน่วยความจำในระบบคอมพิวเตอร์ ที่ใช้สำหรับเก็บข้อมูลหรือตัวแปรแต่ละชนิดแบบชั่วคราวแต่มีข้อเสียคือ เมื่อตัดไปออกแล้วจะทำให้ข้อมูลหรือสิ่งที่อยู่ในหน่วยความจำนี้หายไปจากการที่ตัดไฟออกและข้อมูลในหน่วยความจำนี้หายไปทำให้เกิดเป็นอาการที่เรียกว่า VOLATILE MEMORY แต่ในกรณีของ ROM เรียกว่า NONVOLATILE MEMORYในระยะหลังนี้ได้มีการสร้างแบตเตอรี่ชนิดพิเศษขึ้นมาใช้สำหรับจ่ายไฟสำรองให้กับ RAM ในกรณีที่ไฟเลี้ยงถูกตัดออกจากระบบ เพื่อให้ข้อมูลหรือสิ่งที่อยู่ใน RAM นี้ไม่หายไปและได้มีการผลิต RAM ชนิด NON-VOLATILE ออกมาจำหน่ายด้วย

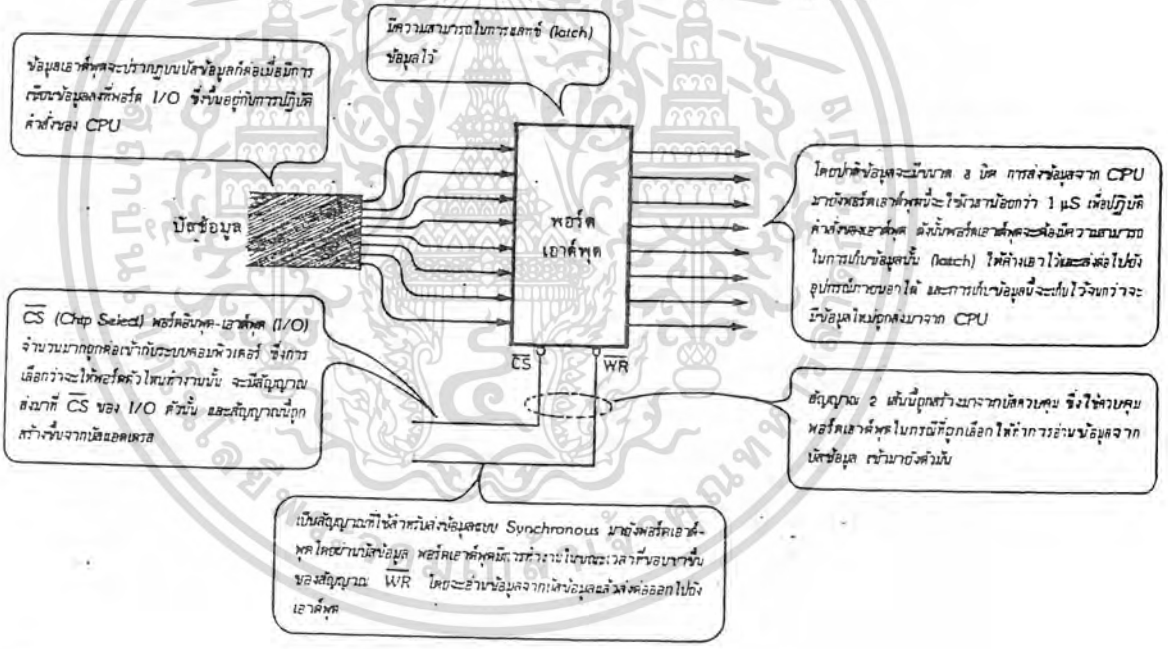
อุปกรณ์อินพุท - เอาต์พุท

ระบบของคอมพิวเตอร์มีการทำงานตามโปรแกรมที่ได้ตั้งไว้ และจำเป็นต้องมีขบวนการที่ดี หรือวิธีที่ดีในการส่งผลลัพธ์ออกไปแสดงผลภายนอกโดยผ่านการพอร์ตเอาต์พุทหรือในการปฏิบัติการบางอย่างมีความจำเป็นต้องอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 แสดงการอินเตอร์เฟสของพอร์ทอินพุท



รูปที่ 1.4 แสดงการอินเตอร์เฟสของพอร์ทเอาต์พุท

วงจรกำเนิดสัญญาณนาฬิกา (CLOCK OSCILLATOR)

วงจรกำเนิดสัญญาณนาฬิกาเป็นพื้นฐานของระบบที่ทำให้ระบบทำงานได้ ถ้าวิเคราะห์การทำงานของแต่ละคำสั่งของโปรแกรมแล้วจะเห็นว่า มีการปฏิบัติมากมายและเป็นอย่างมีระเบียบเรียบร้อย ซึ่งพื้นฐานของการปฏิบัติงานที่เป็นระเบียบเรียบร้อยนี้เป็นผลจากสัญญาณนาฬิกานั่นเองในระบบคอมพิวเตอร์แต่ละระบบจะมีสัญญาณนาฬิกาที่มีสัมพันธ์สอดคล้องกับตัวระบบของมันเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้จะเป็นเครื่องคิดเลขก็ตาม จะมีส่วนกำเนิดสัญญาณนาฬิกาอยู่ด้วย ถ้าหากว่านำเครื่องคิดเลขนี้ไปวางใกล้ ๆ เครื่องรับวิทยุ FM แล้วเกิดมีเสียงรบกวนขึ้นมานั้นเป็นการแสดงว่าตัวไอซีที่ใช้สำหรับควบคุมสัญญาณนาฬิกาของ เครื่องคิดเลขนั้นไม่ดีพอ

ความถี่ของสัญญาณนาฬิกานี้เป็นตัวบอกถึงความเร็วของระบบในการปฏิบัติงานในระยะแรกที่ Z80 ผลิตออกมาจำหน่ายใช้ได้กับความถี่ของสัญญาณนาฬิกาสูงสุดเพียง 2.5 MHz และต่อมาในระยะหลังนี้รุ่นที่ใช้ได้กับความถี่ 6 MHz จนปัจจุบันนี้ที่ผลิตขายอยู่ใช้ได้กับความถี่สูงสุด 8 MHz

บัสหรือสายสัญญาณ

อุปกรณ์แต่ละส่วนที่ได้กล่าวมานั้น ไม่สามารถอยู่ได้โดยลำพัง จำเป็นต้องมีสายสำหรับส่งสัญญาณของข้อมูลหรือ เพื่อควบคุมระหว่างอุปกรณ์แต่ละส่วนนั้น ซึ่งสายสัญญาณเหล่านี้เรียกว่าบัส (BUS) การเปลี่ยนแปลงของสัญญาณในบัสเป็นไปตามเวลาที่ถูกต้องแน่นอน และการเปลี่ยนแปลงของสัญญาณนี้ก็คือ การรับ-ส่งข้อมูลต่าง ๆ นั่นเอง

ในระบบคอมพิวเตอร์ของ Z80 มีบัสอยู่ 3 ประเภทด้วยกันคือ

1. บัสข้อมูล (DATA BUS) เป็นขนาด 8 บิต ใช้สำหรับรับ-ส่งข้อมูลระหว่าง CPU หน่วยความจำและพอร์ตอินพุท-เอาต์พุทต่าง ๆ

2. บัสแอดเดรส (ADDRSEE BUS) เป็นขนาด 16 บิต ใช้สำหรับกำหนดหรือเลือกตำแหน่งของหน่วยความจำและพอร์ตอินพุท-เอาต์พุท

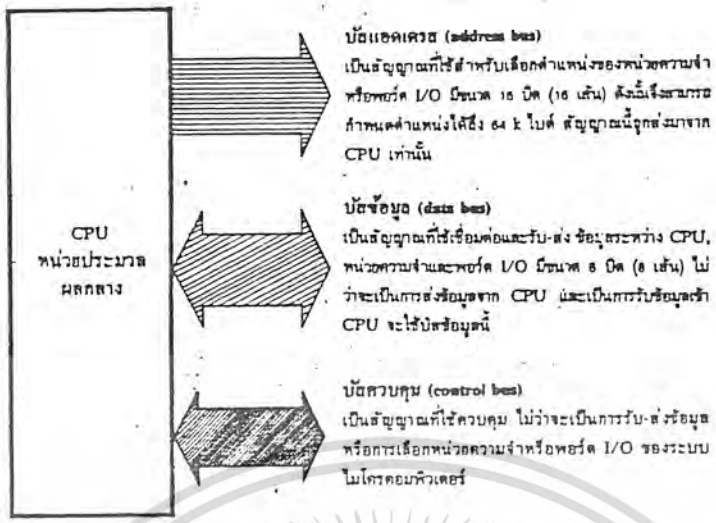
3. บัสควบคุม (CONTROL BUS) เป็นกลุ่มของเส้นสัญญาณควบคุมที่ CPU ส่งออกและรับเข้า เพื่อแสดงสภาวะต่าง ๆ เช่น การอ่าน หรือเขียนข้อมูลจากหน่วยความจำหรือพอร์ตอินพุท-เอาต์พุท

การปฏิบัติคำสั่งของ CPU

ส่วนประกอบพื้นฐานของไมโครคอมพิวเตอร์ ดังแสดงในรูปที่1.1 ซึ่งโครงสร้างที่ดีของไมโครคอมพิวเตอร์พิจารณาตัดสินได้จากส่วนประกอบนี้ จากนี้ไปจะได้กล่าวถึงการทำงานของระบบนี้โดยขั้นแรกจะมาพิจารณาถึงสิ่งที่จำเป็นในขณะที่ระบบคอมพิวเตอร์เริ่มทำงาน

ปัจจุบันนี้คอมพิวเตอร์หลักที่ใช้กันอยู่คือคอมพิวเตอร์ชนิด "นอยแมน"ซึ่งมีโปรแกรมสำเร็จอยู่สามารถคำนวณได้โดยอัตโนมัติ และสามารถที่จะกำหนดการคำนวณได้และนอกจากนี้ยังสามารถเขียนโปรแกรมเปลี่ยนแปลง ควบคุมได้และความเร็วของการปฏิบัติข้อมูลสามารถแสดงได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.5 แสดงบัสแต่ละชนิดของ CPU

ลำดับขั้นตอนการทำงานของ CPU ในการปฏิบัติคำสั่งมีดังนี้

(1) อ่านคำสั่งจากโปรแกรมที่ต้องการจะปฏิบัติ

(2) ทำการวิเคราะห์แปลคำสั่งที่อ่านมาได้

(3) ปฏิบัติงานตามคำสั่งนั้น ซึ่งอาจจะเป็น การเคลื่อนย้ายข้อมูล, การคำนวณหรือการอ่านข้อมูลจากอุปกรณ์ภายนอก เป็นต้น

(4) หลังจากจบการปฏิบัติงานตามคำสั่งนั้นแล้ว ก็เตรียมที่จะอ่านคำสั่งถัดไปปกติการอ่านคำสั่งจะอ่านเรียงตามลำดับลง แต่ในกรณีที่พบคำสั่งสำหรับเปลี่ยนลำดับการทำงานหรือคำสั่งกระโดดระบบจะทำงานตามคำสั่งนั้นโดยเปลี่ยนค่าของแอดเดรสจะบอกคำสั่งให้ไปชี้ที่คำสั่งใหม่ตามที่ต้องการ ซึ่งการเปลี่ยนค่าของแอดเดรสนี้ ทำได้โดยการเปลี่ยนค่าของโปรแกรมเคาน์เตอร์ (PC)

(5) อ่านคำสั่งต่อมาและทำการปฏิบัติการซ้ำตั้งแต่ข้อ (2) ลงมา

การทำงานทั้งหมดนี้จะสัมพันธ์กับสัญญาณนาฬิกา ซึ่งจะมีปัญหาเกี่ยวกับความเร็วของการทำงานของอุปกรณ์แต่ละส่วนอาจจะไม่ทันกัน ดังนั้นเพื่อแก้ปัญหานี้ได้มีการคิดค้นวงจรต่าง ๆ ขึ้น

อย่างไรก็ตาม การทำงานพื้นฐานแล้วจะมีลำดับการทำงานดังได้กล่าวไปแล้ว สำหรับปัญหาต่าง ๆ ทางเทคโนโลยีจะได้กล่าวเป็นตอน ๆ ไปพร้อมกับยกตัวอย่างประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องสมุดคณะวิศวกรรมศาสตร์

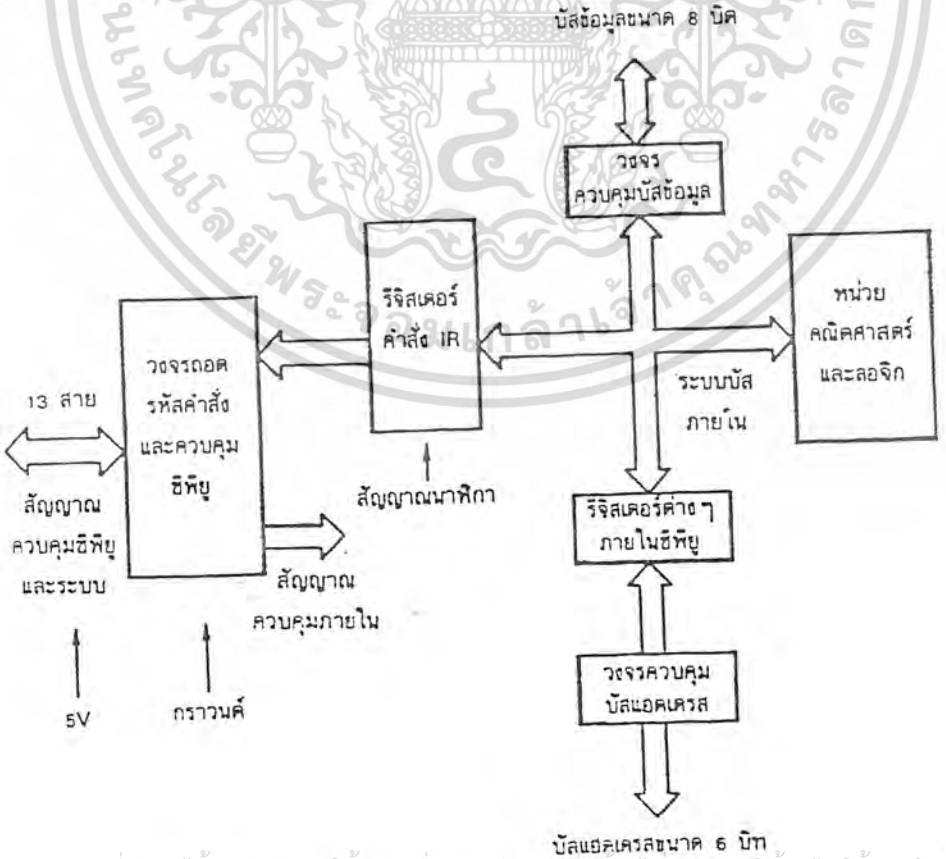
Z80 กับซอฟต์แวร์

ในตอนนี้จะได้อธิบายรายละเอียดต่าง ๆ ที่เกี่ยวเนื่องกับซอฟต์แวร์ของ Z80 ซึ่งมีความสำคัญต่อการเขียนคำสั่งให้ Z80 ทำงาน โดยจะเน้นทางด้านลอจิกของขบวนการรับส่งข้อมูล(DATA)และการประมวลผล (PROCESS) ข้อมูลซึ่งต้องเข้าใจความหมายในคำสั่งของ Z80 เสียก่อนที่จะเขียนโปรแกรมให้ Z80ทำงานตามความต้องการ

เนื้อหาที่จะกล่าวถึงในตอนนี้ได้แก่ชนิดของคำสั่ง (COMMAND) ต่างๆ การทำงานของคำสั่งนั้น ๆ การเก็บข้อมูลไว้ในหน่วยความจำ ตำแหน่งแอดเดรสของคำสั่งและการคาดคะเนความเร็วในการทำงานของคำสั่งต่าง ๆ

โครงสร้างของซีพียู

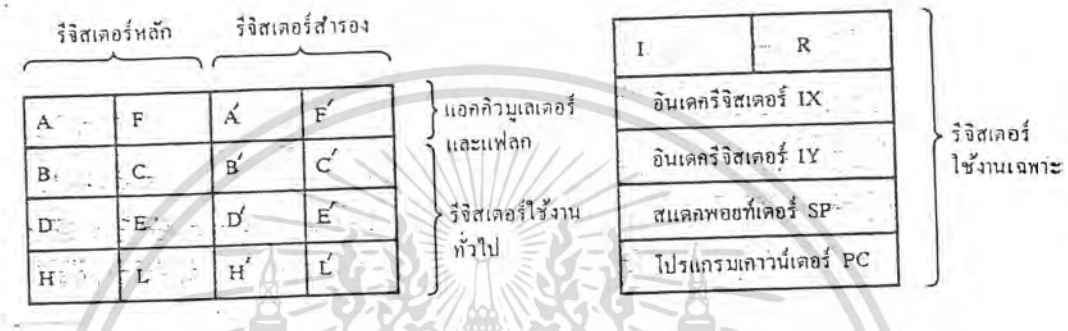
โครงสร้างของซีพียู Z80 มีโครงสร้างที่พัฒนามาจาก 8080 ดังนั้นในแง่โครงสร้างพื้นฐานจะเหมือนกับซีพียูของ 8080 แต่เนื่องจาก Z80 มีการพัฒนามากขึ้นทางซอฟแวร์ จึงทำให้มีรายละเอียดแตกต่างเพิ่มเติมอีกหลายประการด้วยกัน บล็อกไดอะแกรมรูปที่ 1.6 เป็นไดอะแกรมแสดงให้เห็นโครงสร้างของ Z80 โดยโครงสร้างของซีพียูจะบรรจุลงในแอลเอสไอขนาด 40 ขา



รูป 1.6 บล็อกไดอะแกรมซีพียู Z80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
033223

โครงสร้างภายในของ Z80 ซีพียูประกอบด้วยรีจิสเตอร์ภายในที่สามารถเขียนและอ่านได้ถึง 208 บิท โดยแยกเป็นกลุ่มของรีจิสเตอร์ขนาด 8 บิท 18 รีจิสเตอร์และรีจิสเตอร์ขนาด 16 บิทอีก 4 รีจิสเตอร์โดยมีชุดรีจิสเตอร์แสดงไว้ดังรูป 1.7



รูปที่ 1.7 แสดงรีจิสเตอร์ต่าง ๆ ที่มีอยู่ใน Z80

รีจิสเตอร์หลักที่ใช้งานทั่วไป

รีจิสเตอร์ในกลุ่มแรกก็คือ A,F,B,C,D,E,H,L เป็นรีจิสเตอร์ขนาด 8 บิท ที่ใช้งานโดยทั่วไปโดยรีจิสเตอร์เหล่านี้สามารถประกอบรวมกันเป็นคู่รีจิสเตอร์ได้คือ AF,BC,DE,HL โดยคู่รีจิสเตอร์เหล่านี้จะได้รับการใช้งานในลักษณะของรีจจิสเตอร์ขนาด 16 บิท การกระทำภายในซีพียูอาจจะอาศัยเพียงรีจิสเตอร์เดียวหรือกระทำเป็นคู่รีจิสเตอร์ได้โดยที่ A คือแอดคิวมูเลเตอร์ แฟลกของ Z80 จะมีด้วยกันทั้งหมด 6 ตัว จึงใช้เพียง 6 บิท แต่ Z80 อาศัยการเพิ่มบิทขึ้นอีก 2 บิท และกลายเป็นรีจิสเตอร์ F รีจิสเตอร์ F นี้สามารถได้รับการเซทรีเซทการกระทำตามคำสั่งทางคณิตศาสตร์หรือลอจิกได้และเราสามารถใช้ F เหมือนรีจิสเตอร์ซึ่งเมื่อรวมกันกับ A แล้วจะกลายเป็นรีจิสเตอร์ ขนาด 16 บิทได้

กลุ่มรีจิสเตอร์สำรอง

เป็นรีจิสเตอร์ที่สามารถเก็บข้อมูลได้โดยเป็นตัวเก็บข้อมูลที่มาจากรีจิสเตอร์หลัก รีจิสเตอร์ชุดนี้จึงมีด้วยกัน 8 ตัว คือ A',F',B',C',E',H',L' รีจิสเตอร์เหล่านี้เป็นรีจิสเตอร์ที่ใช้กันในการเก็บข้อมูลชั่วคราว ในการที่ต้องการใช้รีจิสเตอร์หลักใช้งานอย่างอื่นก่อน ดังนั้นรีจิสเตอร์กลุ่มนี้จึงไม่สามารถกระทำทางคณิตศาสตร์หรือลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มรีจิสเตอร์ที่ใช้งานเฉพาะอย่าง

โปรแกรมเคาว์เตอร์ (PC-PROGRAM COUNTER) โปรแกรมเคาว์เตอร์ เป็นรีจิสเตอร์ขนาด 16 บิท ที่เป็นตัวกำหนดตำแหน่งของโปรแกรมในขณะสภาวะการกระทำการเฟทช์ โดยขณะทำการเฟทช์ค่าที่อยู่ในโปรแกรมเคาว์เตอร์จะไปปรากฏอยู่ที่แอดเดรสบัสเมื่อชี้ไปยังตำแหน่งในหน่วยความจำให้ CPU อ่านคำสั่งมาตีความหมาย ค่าที่อยู่ในโปรแกรมเคาว์เตอร์จะเพิ่มค่าขึ้นได้อย่างอัตโนมัติหลังการกระทำการเฟทช์ แต่ถ้าหากซีพียูกระทำคำสั่งใช้ข้ามไปยังตำแหน่งอื่น (JUMMP) ค่าแอดเดรสที่จะกระโดดข้ามนั้นจะโหลดเข้ามายังโปรแกรมเคาว์เตอร์ได้อย่างอัตโนมัติ

สแตคพอยท์เตอร์ (SP-STACK POINTER) เป็นรีจิสเตอร์ที่มีขนาด 16 บิท ที่ใช้สำหรับชี้ไปยังแอดเดรสชั้นบนสุดของสแตคที่อยู่ใน RAM โดยส่วนของสแตคมีลักษณะโครงสร้างเป็นหน่วยความจำเป็นแบบเก็บทีหลังเรียกออกได้ก่อน ข้อมูลในสแตคอาจได้รับการพุทหรือพอยออกมาจากข้อมูลรีจิสเตอร์ภายในซีพียู ลักษณะของสแตคในที่นี้ยังเป็นส่วนช่วยในการกระทำอินเตอร์รัพท์และการเรียกโปรแกรมย่อย กล่าวคือในการอินเตอร์รัพท์ค่าของโปรแกรมเคาว์เตอร์ จะได้รับการเก็บรักษาไว้ในชั้นของสแตค ครั้นเมื่อโปรแกรมกลับจากอินเตอร์รัพท์ไปกระทำยังโปรแกรมหลักก็นำค่าจากสแตคกลับเข้ามายังโปรแกรมเคาว์เตอร์ใหม่ ในทำนองเดียวกันการกระโดดไปยังโปรแกรมย่อย ก็เช่นเดียวกันนั้นการกระทำในรูปของอินเตอร์รัพท์หรือโปรแกรมย่อยสามารถซ้อนกันได้ไม่มีที่สิ้นสุด

อินเด็กซ์รีจิสเตอร์ (IX,IY-INDEX REGISTER) ซีพียู Z80 มีอินเด็กรีจิสเตอร์ขนาด 16 บิท 2 ตัว แต่ละตัวใช้ประโยชน์หลักในการทำหน้าที่เป็นตัวเก็บแอดเดรสฐาน (BASE ADDRESS) เพื่อทำหน้าที่ยังแอดเดรสแบบอินเดดแอดเดรสซึ่ง (INDEX ADDRESSING) ในหมวดของอินเดรสให้กับค่าสั่งข้อมูลที่ติดมากับค่าสั่งที่เรียกว่าดิสเพลซเมนท์ (DISPLACEMENT) ซึ่งจะเก็บในรูปตัวเลข 2'S คอมพลีเมนท์

อินเตอร์รัพท์แพจแอดเดรสรีจิสเตอร์ (I-INTERRUPT PAPE ADDRESS REGISTER) การอินเตอร์รัพท์ของ Z80 มีหลายขนาด และหมวดหนึ่งที่ทำให้การอินเตอร์รัพท์ของ Z80 มีประสิทธิภาพสูง กล่าวคือเมื่อเกิดการอินเตอร์รัพท์ ในหมวดนี้ขึ้นมันสามารถอ้างแอดเดรสโดยทางอ้อมไปกระทำโปรแกรมในที่ใดก็ได้หน่วยความจำ โดยอาศัยค่าในรีจิสเตอร์ I รวมกับค่าที่ส่งมาจากอุปกรณ์เพอร์เฟอรัลอีก 8 บิท ชี้ไปยังค่าในรีเจสเตอร์ ในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อนำค่านั้นมาโหลดเข้าในโปรแกรมเคาว์เตอร์เพื่อกระทำต่อไป ด้วยวิธีการนี้เราจึงสามารถกระโดดเข้าไปทำที่ส่วนใดก้ได้ในหน่วยความจำ

รีจิสเตอร์รีเฟรชหน่วยความจำ(R-MEMORY RERLESH REGISTER) การต่อซีพียูกับหน่วยความจำนั้นโดยปกติจะต่อกับหน่วยความจำชนิดสแตคติดได้โดยง่ายแต่อย่างไรก็ดีชนิดไดนามิคที่ต้องการรีเฟรชมีราคาถูกกว่ามีความหนาแน่นสูงกว่า Z80 ให้ข้อดีกว่าประเภทหนึ่งคือมันสามารถให้การรีเฟรชหน่วยความจำได้อย่างอัตโนมัติ โดยค่าในรีเจสเตอร์จะเพิ่มค่าขึ้นอีก 1 ทุกครั้งที่มีการกระทำการเฟทช์คำสั่งและข้อมูลในรีจิสเตอร์ R นี้จะส่งไปยังแอดเดรสยังในส่วนบิทที่มีนัยสำคัญต่ำกว่าจังหวะของการสิ่งนี้จะเป็นจังหวะเดียวกันกับซีพียูส่งสัญญาณรีเฟรชออกมาผู้โปรแกรมสามารถกำหนดค่าให้กับรีเจสเตอร์ R นี้ได้แต่ค่าในรีจิสเตอร์นี้จะเรียกใช้โดยผู้โปรแกรมทางคำสั่งโดยตรงไม่ได้

แอคคิวมูเลเตอร์ (ACCUMULATOR) และแฟลก (FLAG) ซีพียูจะมีรีจิสเตอร์ที่ใช้เป็นหลักในการเป็นตัวโอเปอร์แรนต์ สำหรับกระทำทางคณิตศาสตร์และลอจิก โดยรีจิสเตอร์หลักนี้จะมีเพียง 8 บิท เรียกว่าแอคคิวมูเลเตอร์ (ACCUMULATOR)การกระทำในส่วนของหน่วยคณิตศาสตร์และลอจิกย่อมเกิดเงื่อนไขได้หลายอย่างที่จะต้องแสดงสถานะภาพของเงื่อนไขเหล่านั้นเช่นเงื่อนไขผลลัพท์เป็นศูนย์ผลลัพท์เป็นบวกหรือลบมีตัวทดหรือชอยืมในการกระทำทางคณิตศาสตร์แสดงเงื่อนไขพาร์ตี้คู่หรือคี่ ฯลฯ สิ่งเหล่านี้จะให้ผลลัพท์แสดงสถานะได้ด้วยแฟลก (FLAG) แฟลกเป็นรีจิสเตอร์ขนาด 8 บิท ซึ่งสามารถรวมกับแอคคิวมูเลเตอร์เป็นรีจิสเตอร์ขนาด 16 บิทได้ ผู้โปรแกรมยังสามารถใช้คำสั่งในการเคลื่อนย้ายข้อมูลจากคิวมูเลเตอร์ (A) และแฟลก (F) ไปเก็บไว้ใน A และ F ได้เพื่อให้การใช้งานของ A และ F ประสิทธิภาพดียิ่งขึ้น

หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU-ARITHMETIC AND LOGIC UNIT) การประมวลผลที่สำคัญของซีพียูของคอมพิวเตอร์ยังขึ้นอยู่กับหน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU) ส่วน ALU นี้จะนำข้อมูลซึ่งอาจจะนำมาจากนอกซีพียูหรือภายในซีพียูก็ได้มาประมวลผล การประมวลผลในส่วน ALU ที่สำคัญจะประกอบด้วย

การบวก (ADD)

การลบ (SUBSTRACT)

ลอจิก (AND)

ลอจิก (OR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลอจิก (EX-OR)

เปรียบเทียบ (COMPARE)

การเลื่อนไปทางซ้ายหรือขวา (SHIFT)

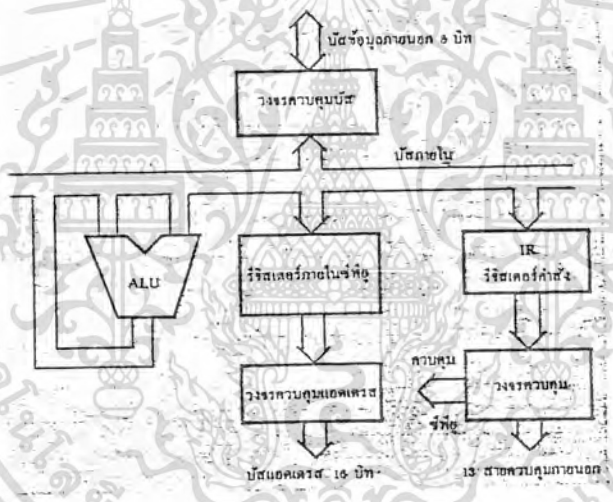
การเพิ่มค่า (INCREMENT)

การลดค่า (DECREMENT)

การเซทบิท (SET BIT)

การรีเซทบิท (RESET BIT)

การทดสอบบิท (TEST BIT)



รูปที่ 1.8 แสดงการทำงาน ALU ภายใน

รีจิสเตอร์คำสั่งและส่วนควบคุม (INTRUCTION REGISTER AND CONTROL) ในการกระทำการเฟทช์ซีพียู จะอ่านคำสั่งจากหน่วยความจำที่เป็นส่วนของโปรแกรมโดยรอคำสั่งนั้นมาเก็บได้ใน IR เพื่อทำการถอดรหัสคำสั่งและส่งสัญญาณควบคุมการทำงานภายในซีพียูหรือ ควบคุมการทำงานของระบบสัญญาณควบคุมเหล่านี้ออกมาในจังหวะต่าง ๆ กันเพื่อควบคุมระบบในการทำงานต่อไป

Z80 กับภาษาแอสเซมบลี้

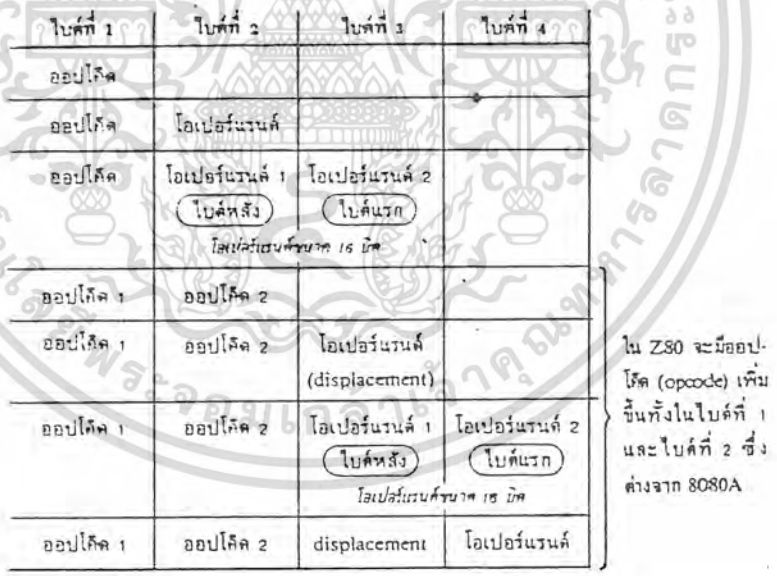
โดยปกติแล้ว ภาวะคำสั่งของเครื่องไมโครคอมพิวเตอร์จะเป็นรหัส (CODE) หรือภาษาเครื่องที่มีความยาวตั้งแต่ 1 ไบต์ขึ้นไป และถูกเก็บบันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน่วยความจำ ซึ่งถ้าจะแสดงคำสั่งเหล่านั้น ก็จะอยู่ในรูปของเลขฐาน 2 หรือฐาน16

สำหรับผู้ที่มีความเคยชินก็คงจะจดจำรหัสของคำสั่งดังกล่าวได้ แต่ในการเขียนโปรแกรมที่ดีนั้น วิธีการจดจำรหัสคำสั่งไม่ใช่เป็นวิธีที่ถูกต้อง วิธีการที่มีประสิทธิภาพมากกว่าคือการเทียบรหัสคำสั่งหรือภาวะเครื่องกับภาษาแอสเซมบลี้

ภาษาแอสเซมบลี้ที่กล่าวถึงนี้ เป็นของบริษัทไซล็อกผู้ผลิต Z80 ซึ่งมีรูปแบบที่เข้าใจง่ายดังแสดงไว้ในรูปที่ 1.10 เพียงแต่ทำความเข้าใจกับรูปแบบพื้นฐานต่าง ๆ ก็จะทำให้สามารถเขียนคำสั่งเหล่านั้นด้วยภาษาแอสเซมบลี้กล่าวนี้โดยง่าย



รูปที่ 1.9 แสดงรหัสภาษาเครื่องของ Z80 ซึ่งมีความยาวจาก 1 ไบต์ถึง 4 ไบต์สังเกตที่โอเปอร์แรนด์ขนาด 16 บิต จะขึ้นต้นด้วยไบต์หลังก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก)รหัสคำสั่งอย่างเดียว	ข)โอเปอร์แรนด์ 1 ตัว	ค)โอเปอร์แรนด์ 2 ตัว
DAA	CP (HL)	LD A,B
EI	AND C	CALL Z,0100H
RLD	RL (IX+2)	IN A,(OBH)
	JR 1CH	

ออปโค๊ด	โอเปอร์แรนด์	โอเปอร์แรนด์
LD	A,	B

รูปที่ 1.10 การเขียนภาษาแอสเซมบลี้ตามวิธีของ บ.ไซลอกผู้ผลิต Z80

แอสเซมบลี้ในรูปของรหัสนีโมนิค

รหัสนีโมนิค (MNEMONIC CODE) ของ Z80 แบ่งออกได้เป็น 3 รูปแบบ ดังแสดงไว้ในรูปที่ 1.10

1. รูปแบบที่แสดงเฉพาะส่วนคำสั่ง (รูปที่ 1.10)

2. รูปแบบที่แสดงเฉพาะคำสั่งและตัวโอเปอร์แรนด์ 1 ตัว โดยปกติของการใช้งานทั่วๆไปแล้ว ตัวโอเปอร์แรนด์จะเป็นส่วนที่เกี่ยวเนื่องกับการกระทำของคำสั่งนั้น ๆ

3. รูปแบบที่แสดงส่วนคำสั่งและตัวโอเปอร์แรนด์ 2 ตัว ตัวอย่างเช่น การคำนวณระหว่างข้อมูลที่เป็นตัวแปร 2 ตัว โดยที่โอเปอร์แรนด์ตัวแรกเป็นตัวแสดงการทำงานของคำสั่งและโอเปอร์แรนด์ตัวที่สองจะเป็นข้อมูลของการคำนวณ

แอสเซมบลี้ (ASSEMBLY)

คำสั่งต่าง ๆ ที่จะทำให้ระบบคอมพิวเตอร์ทำงานจะถูกเก็บไว้ในหน่วยความจำซึ่งคำสั่งต่าง ๆ นี้จะมีความยาวตั้งแต่ 1 ไบต์ไปจนถึงหลายสิบไบต์ และแทนได้ด้วยรหัสของเลขฐาน 2 หรือฐาน 16 ในระยะเริ่มแรกของการพัฒนาระบบคอมพิวเตอร์นั้น มีการเขียนโปรแกรมโดยใช้ภาษาเครื่อง (MACHIMNE LANGAUGE) ดังกล่าวโดยตรงซึ่งปัจจุบันก็ยังคงมีการเขียนโปรแกรมด้วยรหัสเลขฐาน 16 ลงในหน่วยความจำโดยตรงเพื่อทำการทดสอบเมื่อต้องการดีบัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(DEBUG) หรือแก้ไขโปรแกรมแต่ในทางปฏิบัติแล้ว เป็นเรื่องที่ยุ่งยากมากที่จะเขียนโปรแกรมด้วยภาษาเครื่องโดยตรง ดังนั้นจึงมีการคิดหาคำที่รัดกุมมาแทนภาษาเครื่องแต่ละตัว

คำที่ว่านี้เราเรียกว่า รหัสนีโมนิค (MNEMONIC CODE) นี่เองเป็นจุดที่ทำให้เราสามารถนำภาแอสเซมบลี้ที่ดัดแปลงจากรหัสนีโมนิคมาเขียนโปรแกรมได้ง่ายขึ้น และการแปลงโปรแกรมที่เขียนภาษาแอสเซมบลี้ให้เป็นภาษาเครื่องนี้เป็นหน้าที่ของแอสเซมเบลอร์ (ASSEMBLER)

ภาษาแอสเซมบลี้นี้ สามารถเขียนคำสั่งทุก ๆ ตัวได้ ดังนั้นไม่ว่าจะเป็นปัญหาทางด้านฮาร์ดแวร์หรือปัญหาอื่น ๆ ก็ตาม ถ้าหากในระบบมีคำสั่งที่จะจัดการกับปัญหานั้นอยู่แล้ว สามารถที่จะเขียนโปรแกรมเพื่อให้ CPU ทำงานได้เต็มความสามารถถึง 100 %

ฟอร์แมทของแอสเซมเบลอร์

ในการเขียนซอร์สโปรแกรม (SOURCE PROGRAM) นั้น จะต้องเขียนด้วยรูปแบบที่กำหนดไว้เป็นมาตรฐานคงที่ ตัวอย่างที่จะเขียนข้างล่างนี้ เป็นตัวอย่างที่มักจะพบบ่อย ๆ คือ

```
ลาเบล   ออฟโค๊ด   โอเปอร์แรนด์    คอมเมนด์
Loop :   Ld BC   ,MAX            ; GET MAX
```

ลาเบล LOOP ใช้สำหรับบอกแอดเดรสที่คำสั่งของโปรแกรมในบรรทัดนี้ตั้งอยู่

โอเปอร์แรนด์ จะมีเพียง 1 ตัวหรือมากกว่าก็ได้ โดยมีเครื่องหมาย, หรือช่องว่างคั่นเสมอ

คอมเมนต์ จะเริ่มต้นด้วยเครื่องหมาย ;

ลาเบล (LABEL)

ลาเบลเป็นสัญลักษณ์ที่มีขนาดไม่เกิน 16 บิต ใช้กำหนดแอดเดรสหรือข้อมูล ซึ่งมักจะเขียนด้วยคำหรือกลุ่มตัวอักษรที่เข้าใจง่าย ซึ่งตัวอักษร 6 ตัวแรกจะเป็นส่วนที่ถูกใช้งาน (คือมีความหมาย) และส่วนที่เกิน 6 ตัว จะไม่มีความหมายต่อโปรแกรม นอกจากนี้อักษรตัวแรกจะต้องเป็นอักขระภาษาอังกฤษ

การเขียนลาเบลนี้ จะเริ่มต้นที่ตำแหน่งไหนในบรรทัด (LINE) ก็ได้ แต่จะต้องจบลงด้วยเครื่องหมาย : และถ้าเขียนลาเบลโดยเริ่มต้นจากแถว (COLUMN) แรกของบรรทัดนั้นในกรณีไม่จำเป็นต้องมีเครื่องหมาย :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลาเบลจะเป็นค่าที่บอกตำแหน่งที่อยู่ของบรรทัดนั้นในโปรแกรม ซึ่งค่านี้จะถูกเก็บไว้ในตารางของลาเบล (LABEL TABLE) ดังนั้น ถ้าลาเบลถูกเขียนด้วยตัวโอเปอร์แรนด์ก็สามารถที่จะหาค่านี้จากตารางของลาเบลได้

การเขียนรูปของสมการ

รูปแบบของสมการจะประกอบขึ้นด้วยตัวค่าคงที่, ตัวแปร, ฟังค์ชัน ที่ต่อเชื่อมด้วยเครื่องหมายการคำนวณ รูปแบบของสมการนี้ จะมีขอบเขตจำกัดตามความสามารถของแอสเซมเบลอร์ซึ่งใน Z80 เองก็มีข้อจำกัดสำหรับรูปแบบของสมการในทางคณิตศาสตร์และทางลอจิกสมการที่ว่านี้จะคำนวณจากซ้ายไปขวาตามลำดับ โดยไม่สามารถกำหนดการคำนวณก่อนหรือหลัง ด้วยการใส่วงเล็บ เครื่องหมายของการคำนวณมีดังนี้

เครื่องหมาย	หน้าที่	เครื่องหมาย	หน้าที่
+	บวก	/	หาร
-	ลบ	&	การคูณทางลอจิก
*	คูณ	!	การบวกทางลอจิก

การคำนวณทั้งหมดจะอยู่ในรูปของข้อมูล 16 บิต อย่างไรก็ตาม ถ้าค่าของข้อมูลที่ใช้มีเพียง 8 บิต และผลลัพธ์มีเพียง 8 บิตเช่นกัน ในกรณีนี้สามารถใช้สมการที่มีค่าของข้อมูล 16 บิตได้ โดยที่ 8 บิตสุดท้ายจะถูกตัดทิ้งไปเอง

คำสั่งเทียม (PSEUDO ASSEMBLER)

คำสั่งเทียมเป็นคำสั่งเฉพาะทางแอสเซมเบลอร์ ซึ่งแตกต่างจากคำสั่งของ Z80 คำสั่งเทียมนี้ถูกกำหนดขึ้นเพื่อให้การเขียนโปรแกรมสะดวกและรวดเร็วและเข้าใจง่ายคำสั่งเทียมนี้ไม่สามารถที่จะแปลงเป็นภาษาเครื่องได้แต่ตัวโอเปอร์แรนด์เท่านั้นที่จะแปลงเป็นภาษาเครื่องได้ยกตัวอย่างเช่น คำสั่ง DEFB เป็นต้น ในที่นี้จะขอยกตัวอย่างคำสั่งเทียมที่ใช้กันมาก ๆ ได้แก่

ORG NN หมายถึง การกำหนดแอดเดรสด้วยค่าของ NN และจากคำสั่งนี้ทำให้แอดเดรสของคำสั่งอื่น ๆ ในหน่วยความจำถูกกำหนดขึ้น

EQU NN หมายถึง การกำหนดค่าของลาเบลของบรรทัดนี้ด้วย NN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และที่ใดก็ตามภายในโปรแกรมที่มีการใช้ลาเบลนี้เป็นตัวโอเปอร์แรนด์ ก็กำหนดให้ค่าแอดเดรสตรงนั้นเท่ากับ NN

END หมายถึง การสิ้นสุดการทำงานของซอร์สโปรแกรม จะต้องมีคำสั่ง END ที่ท้ายโปรแกรมเสมอ

DEFB N (บางครั้งก็เขียนย่อเป็น DB N) หมายถึง การกำหนดข้อมูลที่แอดเดรสของตำแหน่งบรรทัดนี้ ด้วยค่าของ N

DEFT "S" หมายถึง การกำหนดข้อมูลที่แอดเดรสหรือตำแหน่งของบรรทัดนี้ด้วยรหัส ASCII ของตัวอักษร S

DEFW NN (บางครั้งก็เขียนย่อเป็น DW) หมายถึง การกำหนดข้อมูลที่แอดเดรสของบรรทัดนี้และบรรทัดถัดไปด้วยข้อมูลความยาว 2 ไบต์ คือ NN โดยแทนข้อมูลที่แอดเดรสของบรรทัดนี้ด้วย N ตัวแรก และบรรทัดถัดไปด้วย N ตัวหลัง

DEFS NN (บางครั้งย่อเป็น DA) หมายถึง การจองเนื้อที่ในหน่วยความจำโดยเริ่มจากแอดเดรสของบรรทัดนี้จนถึงตำแหน่งที่มีจำนวนเนื้อที่เท่ากับ NN ไบต์

DEFM "S" (บางครั้งย่อเป็น DM) หมายถึง การกำหนดรหัส ASCII ให้กับกลุ่มตัวอักษร (คือค่า S) ที่อยู่ถัดจากตำแหน่งแอดเดรสของบรรทัดนี้ จำนวนตัวอักษรนี้จะมีได้จาก 1 ตัว ถึง 63 ตัวเท่านั้น

คำสั่งของ Z80

คำสั่งต่าง ๆ ที่ใช้กัน Z80 แบ่งเป็น 2 ส่วนหนึ่งเป็นคำสั่งของ 8080A ของอินเทลทั้งหมดและอีกส่วนหนึ่งเป็นคำสั่งเฉพาะใหม่ ๆ ที่ได้รับการพัฒนาขึ้นจากคำสั่งเดิมของ 8080A ดังนั้นคำสั่งของ 8080A จึงใช่กับ Z80ได้ทั้งหมดนั้นคือโปรแกรมที่เขียนขึ้นกับ 8080A สามารถนำมาใช้กับ Z80 ได้โดยไม่ต้องมีการแก้ไขโปรแกรม

ตัวอย่างเช่น CP/M ซึ่งเป็น OS (OPERATING SYSTEM) หรือระบบการจัดการระบบหนึ่งที่พัฒนาขึ้นมาใช้กับ 8080A นั้น ก็สามารถนำมาใช้กับเครื่องไมโครคอมพิวเตอร์ที่ใช้ Z80 เป็นตัว CPU และในปัจจุบันได้มีการพัฒนาโปรแกรมใช้งานสำเร็จรูปภายใต้ OS ของ CP/M เพื่อใช้กับคำสั่งของ Z80โดยเฉพาะเพื่อการเพิ่มประสิทธิภาพของการทำงาน แต่โปรแกรมที่กล่าวถึงนี้ ไม่สามารถนำมาใช้กับ 8080A ได้

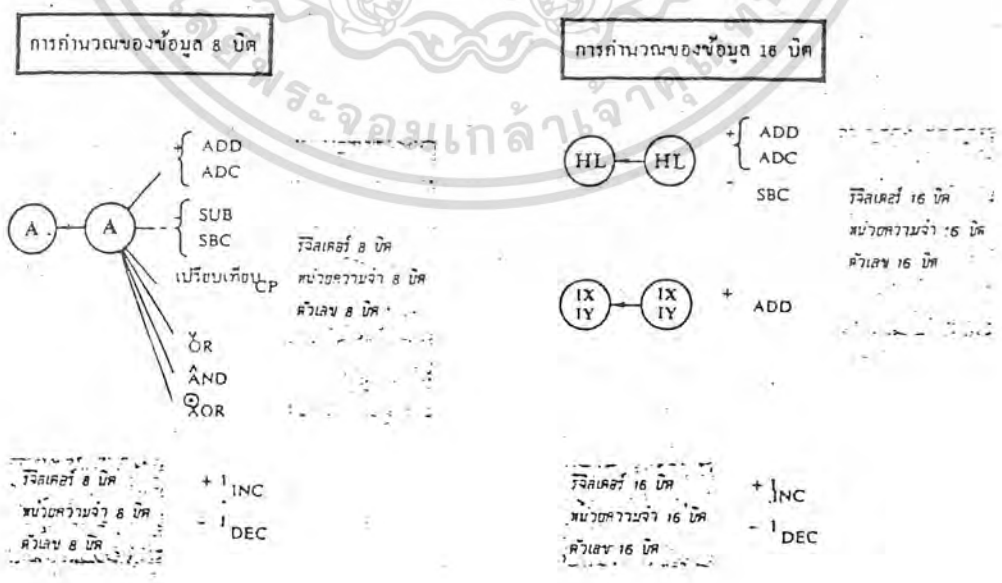
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้การแบ่งประเภทคำสั่งของ Z80 จะมีหลายวิธีก็ตาม ผู้เขียนจะแบ่งเป็นประเภทใหญ่ ๆ ตามการทำงานของคำสั่งนั้น ๆ ออกเป็น 7 ประเภทคือ

1. คำสั่งในการคำนวณ
2. คำสั่งในการเคลื่อนย้ายหรือส่งข้อมูล
3. คำสั่งในการเคลื่อนย้ายเป็นวงรอบและการเลื่อนข้อมูล
4. คำสั่งในการกระทำกับบิต
5. คำสั่งสำหรับควบคุมลำดับการทำงานของโปรแกรม
6. คำสั่งสำหรับควบคุมการทำงานของ CPU
7. คำสั่งเกี่ยวกับอินพุทและเอาต์พุท

คำสั่งในการคำนวณ

แบ่งออกเป็น 2 ประเภทได้แก่ คำสั่งคำนวณทางคณิตศาสตร์และทางลอจิกคำสั่งทางคณิตศาสตร์ได้แก่คำสั่งในการบวก,ลบ คือ ADD, ADC, SBC, SUB, ส่วนคำสั่งทางลอจิกได้แก่คำสั่ง OR, AND, XOR และคำสั่งในการเปรียบเทียบค่า (CP)นอกจากนี้ยังมีคำสั่งในการลดค่า [DEC] และเพิ่มค่า [INC]ดังแสดงไว้ในรูปที่ 1.11



รูปที่ 1.11 คำสั่งในการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งในการเคลื่อนย้ายหรือส่งข้อมูล

ได้แก่คำสั่ง LD ต่าง ๆเป็นคำสั่งในการเคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์กับรีจิสเตอร์หรือรีจิสเตอร์กับหน่วยความจำในรูปที่1.12 แสดงรูปแบบต่างๆ ในการเคลื่อนย้ายข้อมูลและการทำงานของคำสั่งแต่ละรูปแบบ ซึ่งมีทั้งการเคลื่อนย้ายข้อมูลขนาด 8 บิต และ 16 บิต โดยการเคลื่อนย้ายข้อมูลนี้จะมีฝ่ายส่งอยู่ทางขวามือ (หลังเครื่องหมาย,) และฝ่ายรับอยู่ติดกับรหัสคำสั่ง LD

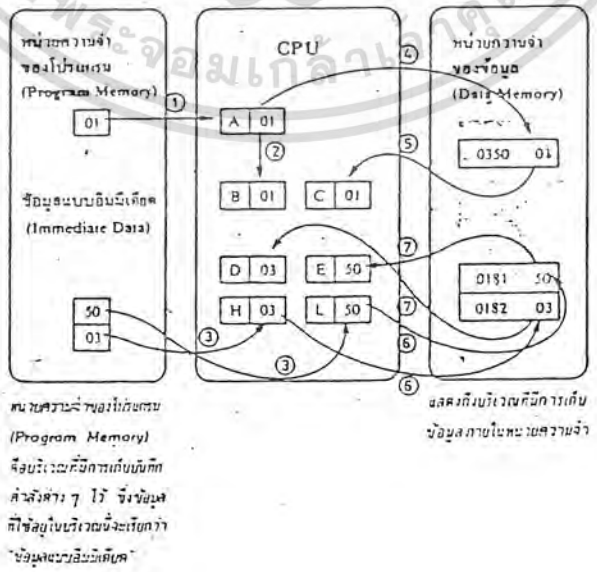
ในการเคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์กับรีจิสเตอร์ เช่นคำสั่งที่ 2 เป็นการก๊อบปี้ข้อมูลจากฝ่ายส่งไปยังฝ่ายรับ โดยที่ข้อมูลในรีจิสเตอร์ฝ่ายส่งจะไม่มีการเปลี่ยนแปลง

การเคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์กับหน่วยความจำเช่นคำสั่งที่ 4 5 6 7 สังเกตว่าจะมีเครื่องหมายวงเล็บอยู่ ในกรณีนี้การแสดงข้อมูลของหน่วยความจำจะะมีหลายวิธีจะได้กล่าวต่อไปในหัวข้อการอ้างแอดเดรส

นอกจากนั้นก็มีการเคลื่อนย้ายข้อมูลแบบค่าตัวแปรเข้าไปในรีจิสเตอร์โดยตรง เช่นคำสั่งที่ 1 และ 3

คำสั่งในการเคลื่อนย้ายข้อมูลเป็นวงรอบและการเลื่อนข้อมูล

คำสั่งในการเคลื่อนย้ายข้อมูลเป็นวงรอบ (ROTATE) และการเลื่อนข้อมูล (SHIFT) จะกระทำกับข้อมูลที่อยู่ในหน่วยความจำ ซึ่งถูกกำหนดโดยรีจิสเตอร์ IX,IY หรือ HL และยังกระทำรีจิสเตอร์ 8 บิตอื่น ๆ กล่าวคือจะทำได้ทั้งในแอดคิวมูเลเตอร์และในรีจิสเตอร์ B,C,D,E,H,L ทั้งหมด

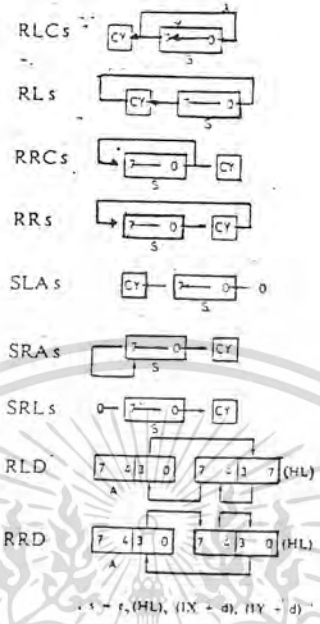


รูปที่ 1.12 การทำงานของคำสั่งในการเคลื่อนย้ายข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลฝ่ายรับ-ฝ่ายส่ง	ภาษาเครื่อง	การทำงาน
1) LD AO1H	3E01	A - 01
2) LD B,A	47	B - A หลังจากที่มีการเคลื่อนย้ายข้อมูลแล้วข้อมูลใน A ไม่เปลี่ยนแปลง
3) LD HL035H	2150 03	H - 03, - 50
4) LD[0350H],A	3250 03	นำข้อมูลจากรีจิสเตอร์ A ไปยังหน่วยความจำของแอดเดรสที่กำหนดโอเปอร์แรนด์
5) LD C,[HL]	4E	นำข้อมูลภายในหน่วยความจำของแอดเดรสที่กำหนดโดยรีจิสเตอร์ HL ไปใส่ไว้ในรีจิสเตอร์ C
6) LD[0181H],HL	2281 01	นำข้อมูลทั้งหมดภายในคู่รีจิสเตอร์ HL ไปใส่ไว้ในหน่วยความจำของแอดเดรสที่กำหนดโดยตัวโอเปอร์แรนด์นั่นคือ แอดเดรส 0181H - L แอดเดรส 0182H - H
7) LD DE [0181H]	ED 5B81 01	นำข้อมูลมั้งหมดในหน่วยความจำของแอดเดรสที่กำหนดโดยตัวโอเปอร์แรนด์ไปใส่ในคู่รีจิสเตอร์ DE นั่นคือ E - แอดเดรส 0181H D - แอดเดรส 0182H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.13 คำสั่งในการเคลื่อนย้ายข้อมูลเป็นวงรอบและการเลื่อนข้อมูล

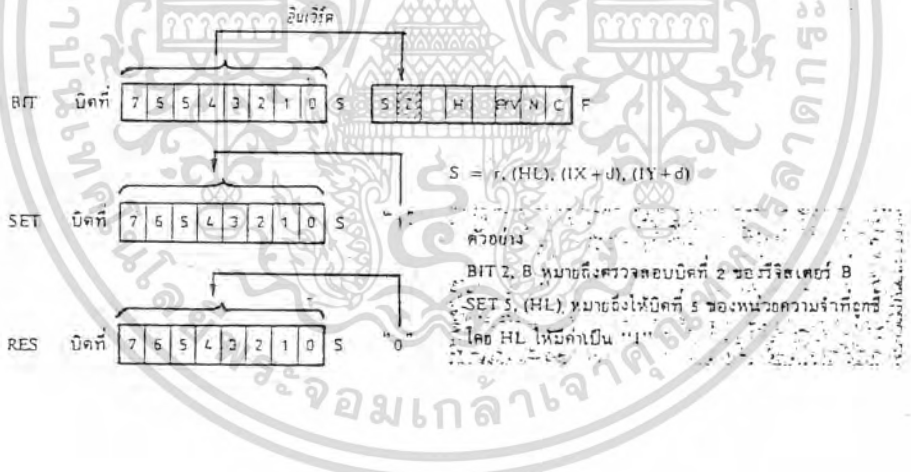
รูปที่ 1.13 แสดงคำสั่งต่าง ๆ ในคำสั่งประเภทนี้ ซึ่งมีรหัสคำสั่งเป็น RLC, RL, RRC, RR, SLA, SRL, RLD และ RRD สังเกตว่าการเคลื่อนย้ายหรือเลื่อนข้อมูลมีทั้งการเลื่อนทางซ้ายขวา และเลื่อนข้อมูลแบบ 8 บิต (บิต 0 ถึง บิต 7) หรือ 9 บิต (รวมแฟลก CARRY ด้วย) ซึ่งเป็นการเลื่อนข้อมูลครั้งละ 1 บิต ไปพร้อม ๆ กัน นอกจากนั้นก็เป็นคำสั่งสำหรับการเลื่อนครั้งละ 4 บิต หรือเคลื่อนย้ายข้อมูลครั้งละ 4 บิต พร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งในการกระทำกับบิต

แบ่งได้ออกเป็น 3 แบบ ได้แก่ การตรวจสอบ (BIT) การเซต (SET) และการรีเซต (RES) ซึ่งจะกระทำเพียงบิตเดียวกับรีจิสเตอร์ 8 บิต หรือหน่วยความจำที่ถูกกำหนดโดยรีจิสเตอร์ IX IY หรือ HL

รูปที่ 1.14 แสดงการทำงานของคำสั่งในกลุ่มนี้ สังเกตว่าคำสั่ง BIT จะมีผลต่อแฟลก ZERO โดยนำบิตที่ต้องการมาอินเวิร์ตลอจิก แล้วใส่เข้าไปในแฟลก ZERO ส่วนคำสั่ง SET,RES เป็นการใส่ลอจิก "1" หรือ "0" เข้าไปยังบิตที่ต้องการในรีจิสเตอร์หรือหน่วยความจำ

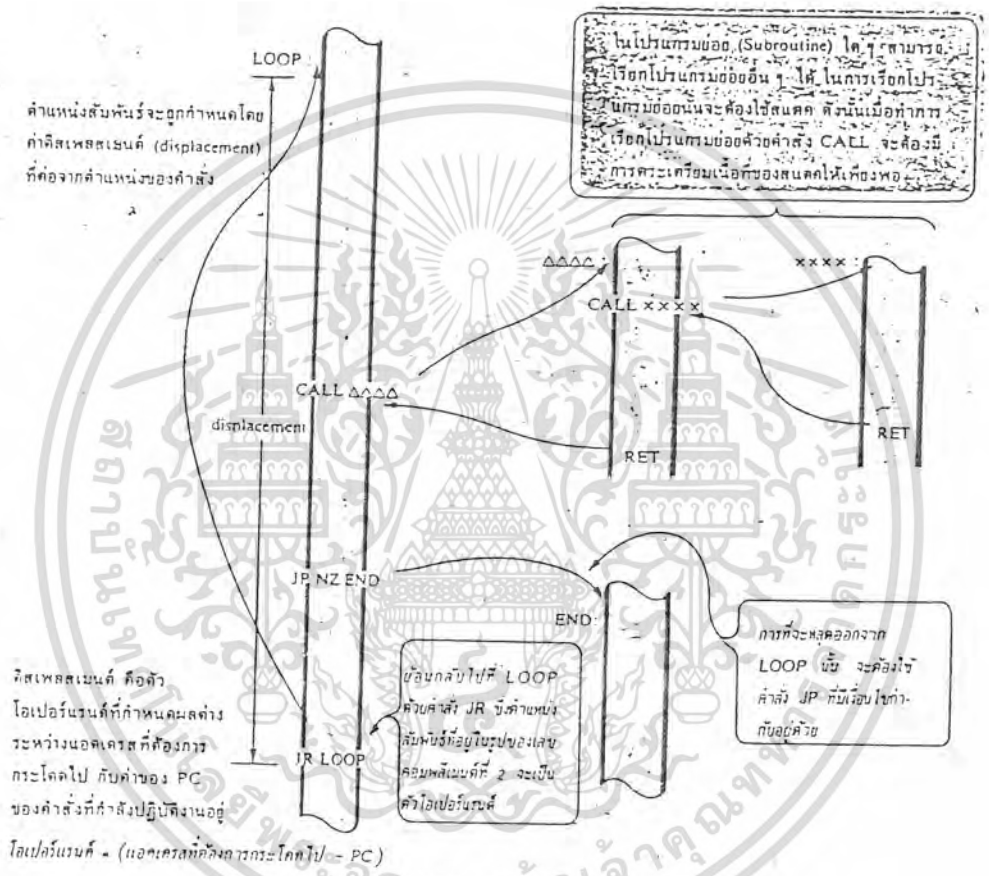


รูปที่ 1.14 การทำงานของคำสั่งในการกระทำกับบิต

คำสั่งสำหรับควบคุมลำดับการทำงานของโปรแกรม

โดยปกติแล้ว คำสั่งของโปรแกรมจะถูกเรียกออกมาปฏิบัติงานทีละคำสั่งตามลำดับที่ถูกเก็บบันทึกไว้ในหน่วยความจำ ซึ่งลำดับการทำงานนี้สามารถเปลี่ยนแปลงได้ เช่นสามารถใช้คำสั่งที่มักใช้บ่อย ๆ ให้มาอยู่ในซับรูทีน [SUBROUTINE] เดียวกันแล้วเรียกออกมาใช้พร้อมกันได้ (โดยใช้คำสั่ง CALL) เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

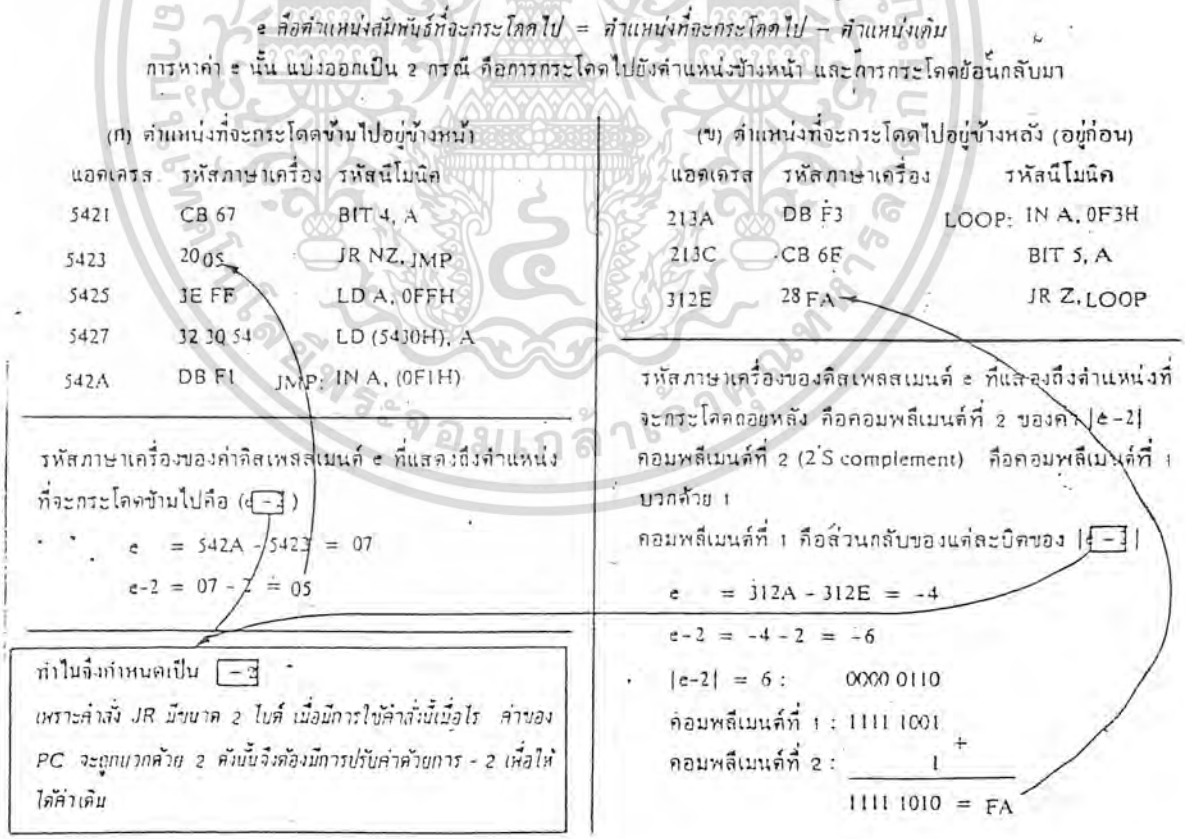


รูปที่ 1.15 การทำงานของคำสั่งสำหรับควบคุมลำดับการทำงาน

นอกจากนี้ ยังมีคำสั่งกระโดดตามเงื่อนไข (CONDITION JUMP) หรือคำสั่งเรียกโปรแกรมย่อยตามเงื่อนไข (CONDITIONAL CALL) คือคำสั่ง JP หรือ CALL ที่มีข้อแม้กำหนดอยู่ด้วย เช่นใช้ในกรณีที่ต้องการเปลี่ยนแปลงขั้นตอนการทำงานของโปรแกรมเมื่อเงื่อนไขเป็นตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแตกต่างของคำสั่ง JP และ CALLนั้นก็คือ JP เป็นคำสั่งที่ใช้สั่งให้ข้ามการทำงานไปยังตำแหน่งของโปรแกรมที่กำหนดโดยที่ไม่มีการย้อนกลับมาที่เดิมอีก ส่วน CALL นั้นก่อนที่จะทำการเรียกโปรแกรมส่วนที่ต้องการออกมา จะมีการเก็บรักษาแอดเดรสของคำสั่ง RET เพื่อให้ย้อนกลับไปที่ตำแหน่งโปรแกรมที่กำหนดไว้ในสแตค โดยการนำค่าแอดเดรสในสแตคนี้ป้อนให้โปรแกรมเคาน์เตอร์ (PC) และและยังอาจจะมีการกำหนดเงื่อนไข (CONDITION) คู่กับคำสั่ง RET ก็ได้ในคำสั่งZ80 มีคำสั่ง JR (RELATIVE JUMP) ซึ่งสามารถกระโดดไปด้วยการกำหนดระยะห่างจากแอดเดรสเดิม โดยการกำหนดค่า RELATIVE ADDRESS นับเป็นข้อดีของ Z80เพราะสามารถย้ายโปรแกรมที่เขียนขึ้นไป RUM ที่แอดเดรสใด ๆ ก็ได้โดยไม่ต้องแก้ไขโปรแกรม



รูปที่ 1.16 แสดงการใช้คำสั่ง JR เพื่อการกระโดดโดยมีการอ้างค่าเปรียบกับแอดเดรสเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งสำหรับควบคุมการทำงานของ CPU

ควบคุมการขัดจังหวะหรืออินเตอร์รัพต์ (INTERRUPT) ทางการทำงาน โดยมีการกำหนดโหมดสำหรับส่วนของงานที่จะแทรกเข้ามาขัดจังหวะ และยังควบคุมให้การทำงานของ CPU ในขณะนั้นหยุดชั่วขณะหนึ่ง (HALT)

คำสั่งเกี่ยวกับอินพุทและเอาต์พุท

เป็นคำสั่งสำหรับการส่ง (OUTPUT) และรับ (INPUT) ข้อมูลกับอุปกรณ์ภายนอกโดยที่พอร์ตอินพุท-เอาต์พุทจะเชื่อมต่อกับส่วนอื่น ๆ ที่เกี่ยวกับการส่งและรับข้อมูล ที่พอร์ตอินพุท-เอาต์พุท สามารถกำหนดแอดเดรสได้โดยใช้ 8 บิตแรกหรือ 8 บิตสุดท้ายของบัสแอดเดรสซึ่งจะได้แอดเดรส OOH-FFH

การเลือกตำแหน่งแอดเดรสที่ว่านี้มี 2 วิธีคือ วิธีแรกกำหนดที่โอเปอร์แรนด์ตัวที่ 2 ของคำสั่งอินพุท-เอาต์พุท ส่วนอีกวิธีกำหนดลงไปในรีจิสเตอร์ C ดังตัวอย่าง

```
IN A, N
OUT R,[C]
```

ในที่นี้ N จะเป็นแอดเดรส I/O ที่มีความยาว 1 ไบต์ และ R จะแสดงถึงรีจิสเตอร์ A,B,C,D,E,H,L

แฟลกมีไว้ใช้สำหรับการตัดสินหรือพิจารณาผลการทำงาน

ในการเขียนโปรแกรมนั้นบางครั้งจำเป็นต้องมีส่วนตัดสินใจ เช่น การเปรียบเทียบค่ามากค่าน้อยของข้อมูล, การเปรียบเทียบความเท่ากันของข้อมูล, การตรวจสอบความเป็น "0" หรือ "1" ของข้อมูลทีละบิต, การตรวจสอบผลการคำนวณว่ามีส่วนที่เหลือเศษ ซึ่งจะแสดงให้อยู่ภายใน 8 บิตได้หรือไม่ เป็นต้นการทำงานในส่วนที่กล่าวมานี้จะใช้รีจิสเตอร์แฟลก (FLAG REGISTER) เป็นตัวปฏิบัติ แฟลกนี้เป็นรีจิสเตอร์ ขนาด 8 บิตซึ่งถูกใช้งานคู่กับรีจิสเตอร์ A ในกรณีนี้จะมีรูปแบบของคำสั่งประกอบด้วยรหัสนีโมนิคคู่กับ AF เช่น

```
PUSH AF
POP AF
```

ตารางที่ 1 แสดงรายละเอียดของการกระทำของคำสั่งต่าง ๆ ที่มีผลต่อแฟลก ผลลัพธ์ที่ได้จากการใช้คำสั่งในตารางที่ 1 จะถูกเก็บไว้ที่รีจิสเตอร์แฟลก ซึ่งสามารถตัดสินผลได้จากการพิจารณาบิตต่าง ๆ ของผลลัพธ์นั้น และโดยปกติแล้วมักจะใช้คำสั่ง JP หรือ JR หรือ RET ที่มีเงื่อนไขกำกับอยู่มาช่วยในการตัดสินผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง	D_7	D_6	D_5	D_4	D_3	D_2	ความหมาย	
	S	Z	H	P/V	N	C		
ADD A, s; ADC A, s	•	•	•	V	0	•	การบวกข้อมูล 8 บิต, การบวกที่มีตัวทด (carry)	
SUB s; SBC A, s; CP s	•	•	•	V	1	•	การลบข้อมูล 8 บิต, การลบที่มีตัวทดยืม, การเปรียบเทียบค่า	
NEG	•	•	•	V	1	•	การหาคอมพลีเมนต์ที่ 2 ของแอคคิวมูเลเตอร์	
AND s	•	•	1	P	0	0	การคำนวณทางลอจิก	
OR s; XOR s	•	•	0	P	0	0		
INC m	•	•	•	V	0	/	การเพิ่มค่าขึ้น 1 ให้ข้อมูล 8 บิต แต่ถ้าเป็นข้อมูล 16 บิต แฟลกทุกตัวจะไม่เปลี่ยนแปลง	
DEC m	•	•	•	V	1	/	การลดค่าลง 1 แก่ข้อมูล 8 บิต แต่ถ้าเป็นข้อมูล 16 บิต แฟลกทุกตัวไม่เปลี่ยนแปลง	
ADD HL, ss; ADD IX, pp; ADD IY, rr	/	/	×	/	0	•	การบวกข้อมูล 16 บิต	
ADC HL, ss	•	•	×	V	0	•	การบวกข้อมูล 16 บิต ที่มีตัวทด	
SBC HL, ss	•	•	×	V	1	•	การลบข้อมูล 16 บิต ที่มีตัวทดยืม	
RLA; RLCA; RRA; RRCA	/	/	0	/	0	•	การโรเตดแอคคิวมูเลเตอร์ (rotate accumulator)	
RLm; RLCm; RRm; RRCm SLA m; SRAm; SRL m	•	•	0	P	0	•	การโรเตด, ชิฟต์ (shift)	
RLD; RRD	•	•	0	P	0	/	การชิฟต์ข้อมูล 4 บิต ระหว่างแอคคิวมูเลเตอร์กับ (HL)	
DAA	•	•	•	P	/	•	การแก้ค่าเลขฐาน 10 ในแอคคิวมูเลเตอร์	
CPL	/	/	1	/	1	/	การกลับค่า (complement) ในแอคคิวมูเลเตอร์	
SCF	/	/	0	/	0	1	การเซตแฟลกตัวทด (carry)	
CCF	/	/	×	/	0	•	การกลับค่าแฟลกตัวทด	
IN r, (C)	•	•	•	P	0	/	การอินพุตโดยผ่านรีจิสเตอร์ C	
INI; IND; OUTI; OUTD	×	•	×	×	1	/	การอินพุต-เอาต์พุต ข้อมูลเป็นบล็อก	Z = 1 เมื่อ B-1 = 0 นอกนั้น Z = 0
INIR; INDR; OTIR; OTDR	×	1	×	×	1	/		
LDI; LDD	/	/	0	•	0	/	การเคลื่อนย้าย ข้อมูลเป็นบล็อก	P/V = 1 เมื่อ BC-1 = 0 นอกนั้น P/V = 0
LDIR; LDDR	/	/	0	0	0	/		
CPI; CPIR; CPD; CPDR	•	•	•	•	1	/	การค้นหา ข้อมูลเป็นบล็อก	Z = 1 เมื่อ A = (HL) นอกนั้น Z = 0 P/V = 1 เมื่อ BC-1 ≠ 0 นอกนั้น P/V = 0
LD A, I; LD A, R	•	•	0	IFF	0	/	ข้อมูลของ IFF ถูกถอบนี้ไว้ที่ P/V	
BIT b, m	×	•	1	×	0	/	ข้อมูลบิตที่ b ของ S ถูกถอบนี้ไว้ที่ Z	

S : แฟลกเครื่องหมาย (sign flag) ถ้าผลการคำนวณเป็นลบ (คือบิตที่ 7 เป็น "1") จะได้ S = 1

Z : แฟลกแสดงค่าศูนย์ (zero flag) ถ้าผลการคำนวณได้ค่าเป็นศูนย์ จะได้ Z = 1

• H : แฟลกตัวทดครึ่งหน่วย (half carry) ถ้ามีการใช้ตัวทดหรือตัวยืมจาก 4 บิตแรกของการคำนวณแบบ BCD จะได้ H = 1

P/V : แฟลกแสดงพาริตี้และโอเวอร์โฟลว์ (parity/overflow flag) ถ้าพาริตี้เป็นเลขคี่จะได้ P/V = 1 ; ถ้าพาริตี้เป็นเลขคู่จะได้ P/V = 0; ถ้ามีโอเวอร์โฟลว์จะได้ P/V = 1

• N : แฟลกแสดงการบวกและลบ (subtract flag) ถ้าเป็นการลบจะได้ N = 1

C : แฟลกตัวทด (carry flag) ถูกกำหนดขึ้นเพื่อแสดงการปัดหน่วยของตัวเลขในแอคคิวมูเลเตอร์จากบิตสูงสุดท้าย ถ้ามีตัวทดในการบวกหรือมีตัวยืมในการลบ จะได้ C = 1 ; มีผลจากการโรเตดหรือชิฟต์

• มีการเปลี่ยนแปลง
/ ไม่มีการเปลี่ยนแปลง
0 ถูกรีเซตให้เป็น "0"
1 ถูกเซ็ตให้เป็น "1"
× ไม่นนอน
V ใช้เป็นแฟลกโอเวอร์โฟลว์
P ใช้เป็นแฟลกพาริตี้
IFF แสดงสภาวะของการอินเตอร์รัพท์

• หมายถึง แฟลกที่ผู้ใช้โปรแกรมไม่จำเป็นต้องทราบสิ่งนี้

ตารางที่ 1 การเปลี่ยนแปลงของบิตต่าง ๆ ของแฟลกไปตามชนิดของคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบค่ามากและค่าน้อยระหว่างข้อมูล

ในการตรวจสอบความสัมพันธ์ของค่ามากและค่าน้อยระหว่างข้อมูลใดๆ จะใช้คำสั่ง CP (มาจาก COMPARE) โดยที่คำสั่งนี้จะนำเอาข้อมูลที่กำหนดด้วยตัวโอเปอร์แรนด์ มาทำการเปรียบเทียบกับข้อมูลในรีจิสเตอร์ A แล้วนำผลการเปรียบเทียบไปแสดงที่แฟลกในการลบ (คือใช้คำสั่ง SUB หรือ SBC) นั้นข้อมูลในรีจิสเตอร์ A จะถูกแทนด้วยผลลัพธ์ของการลบ แต่ในการใช้คำสั่ง CP ข้อมูลภายในรีจิสเตอร์แฟลกเท่านั้นที่เปลี่ยนแปลง โดยที่ข้อมูลในรีจิสเตอร์ A จะไม่เปลี่ยนแปลงเพราะฉะนั้นจึงสามารถทำการตรวจสอบหรือเปรียบเทียบค่ามากค่าน้อยของข้อมูลเมื่อเทียบกับข้อมูลในรีจิสเตอร์ A โดยไม่ทำให้ข้อมูลในรีจิสเตอร์ A สูญหายไปในตาราง 2 ได้แสดงให้เห็นการเปลี่ยนแปลงของแฟลก เมื่อมีการเปรียบเทียบของข้อมูล 2 ตัว คือระหว่างโอเปอร์แรนด์กับรีจิสเตอร์ A

ในการตรวจสอบความมากกว่าหรือน้อยกว่า จะใช้แฟลกตัวทด (C), แฟลก แสดงค่าศูนย์ (Z) และแฟลกเครื่องหมาย (S) แต่ถ้าต้องการตรวจสอบความเท่ากัน ก็เพียงแต่ตรวจสอบแฟลกแสดงค่าศูนย์ (Z) เท่านั้นก็พอ

ผลของการเปรียบเทียบ		สถานะภาพของแฟลก		
		Z	C	S*
	A = S	1	0	0
ไม่คิดเครื่องหมาย	A > S	0	0	
	A < S	0	1	
คิดเครื่องหมาย	A > S	0		0
	A < S	0		1

ตาราง 2 แสดงการเปลี่ยนแปลงของแฟลก เนื่องจากผลของการเปรียบเทียบด้วยคำสั่ง CPs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* แฟลกเครื่องหมายหรือแฟลก S นี้ จะมีการใช้ก็ต่อเมื่อมีการใช้ตัวเลขที่มีเครื่องหมายด้วย

เครื่องหมายนี้จะแสดงไว้ที่บิต 7 ในแอคคิวมูเลเตอร์ ซึ่งจะถูกก๊อปปี้ไปที่แฟลก S

เมื่อเครื่องหมายเป็นลบ (M) จะได้ S = 1

เมื่อเครื่องหมายเป็นบวก (P) จะได้ S = O

```
0000- AF          SUMUP XOR A    การบวกข้อมูลจำนวนหลายไบต์
0001- 86     LOOP ADD  A, (HL)   (ขึ้นอยู่กับรีจิสเตอร์ B) เข้าด้วย
0002- 23          INC  HL        กัน ตามแอดเดรสที่กำหนดโดยรีจิส
0003- 05          DEC  B         เตอร์ HL
0004- 20 FB       JR   NZ, LOOP
0006- C9          RET
                  *
                  *
0007- 21 1B 00  SUM LD HL,DATA1 การบวกข้อมูลจำนวน 5 ไบต์
000A- 06 05         LD B, 5     ที่อยู่ต่อจาก DATA 1 และ
                                ผลลัพธ์จะอยู่ในรีจิสเตอร์ A

000C- CD 00 00  CALL SUMUP
000F- FE 32     CP   50         การเปรียบเทียบค่าระหว่าง A
0011- DO        RET  NC         กับ 50 ซึ่งถ้า A > 50 ก็จะ
                                RETURN
                  *
0012- 21 20 00 SUM2 LD HL,DATA2 โปรแกรมย่อยที่ทำการบวกโดย
0015- 06 03     LD   B, 3       เพียงแต่เปลี่ยนค่าตัวแปรในรีจิส
0017- CD 00 00 CALL SUMUP       เตอร์ HL จาก DATA2 ถึงแม้
```

รูปที่ 1.17 ตัวอย่างโปรแกรมภาษาแอสเซมบลี้และคำอธิบายแสดงการทำงานของคำสั่ง(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
001A- C9        RET                 ข้อมูลที่จะบวกกันแตกต่างกัน วิธี
                                    นี้ช่วยให้เขียนโปรแกรมได้สะดวก
                                    ขึ้น กล่าวคืออาศัยวิธีการของ
                                    REGISTER    INDIRECT
                                    ADDRESSING

                *
001B- 00        DATA1 .DA   #0
001C- 06              .DA   #6      ข้อมูล 5 ไบต์ที่ถูกนำไปใช้บวกกัน
001D- 0C              .DA   #12           โดย SUM
001E- 23              .DA   #35
001F- 26              .DA   #38
                *
0020- 09        DATA2 .DA   #9
0021- 21              .DA   #33     ข้อมูล 3 ไบต์ ที่ถูกนำไปใช้
0022- 23              .DA   #35           บวกกันโดย SUM2
                *
                      .EN

SYMBOL  TABLE

001B- DATA1           ตารางแสดงแอดเดรสของลาเบลที่ผู้เขียนกำหนด
0020- DATA2           ขึ้นเองเพื่อความสะดวกในการอ้างถึงกันและ

0001- LOOP            แสดงความหมายที่เข้าใจได้ง่าย
0007- SUM
0012- SUM2
0000- SUMUP
```

รูปที่ 1.17 ตัวอย่างโปรแกรมภาษาแอสเซมบลี้และคำอธิบายแสดงการทำงานของคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ้างแอดเดรสถึงตำแหน่งที่มีการเก็บข้อมูล

หน่วยความจำของ Z80 มีเนื้อที่สำหรับอ้างแอดเดรสได้เป็นจำนวน 64K ไบต์วิธีการอ้างแอดเดรสภายในหน่วยความจำนี้มีหลายวิธีตามความจำเป็นของการที่จะจัดการกับข้อมูลที่มีความยาวจาก 8 บิต ถึง 16 บิต ดังต่อไปนี้

(1) IMMEDIATE ADDRESSING (การอ้างโดยตรงด้วยตัวเลข)

เป็นวิธีการอ้างข้อมูลโดยตรงในคำสั่ง คือ กำหนดให้โอเปอร์แรนด์ตัวที่สองที่อยู่ถัดจากคำสั่งให้เป็นข้อมูล โดยเป็นตัวเลขที่มีความยาว 1 ไบต์หรือ 2 ไบต์ ซึ่งผลสุดท้ายจะได้รหัสคำสั่งที่มีความยาว 2 หรือ 3 ไบต์ขึ้นไป

ข้อมูลที่เป็นตัวเลข 2 ไบต์ และอยู่ในรูปของรหัสภาษาเครื่องจะถูกบันทึกไว้ในหน่วยความจำตามลำดับต่อจากออปโค๊ด

ตัวอย่าง

```
LD A, n
LD A, 50H ; 3E 50
LD BC,5000H;01 00 50
```

หมายถึง การใส่ค่าตัวเลขเข้าไปในรีจิสเตอร์โดยตรง คำสั่งแรกให้ A มีค่าเป็น 50H และคำสั่งสุดท้ายให้ BC มีค่าเป็น 5000H สังเกตว่าที่รหัสภาษาเครื่องไบต์หลัง (00) จะขึ้นก่อนไบต์แรก (50)

(2) EXTENDED ADDRESSING (การอ้างโดยขยายความยาวของคำสั่ง)

ข้อมูลที่อยู่ในลักษณะนี้ก็คือ ข้อมูลภายในหน่วยความจำของแอดเดรสที่กำหนดโดยตัวเปอร์แรนด์ ถ้าเป็นข้อมูลความยาว 1 ไบต์ละก็ข้อมูลที่ต้องการคือข้อมูลในแอดเดรสถูกกำหนดโดยตัวโอเปอร์แรนด์โดยตรง เช่น คำสั่ง

```
LD A, (NN)
LD A, (4000H) ; 3A 00 40
```

ออปโค๊ด โอเปอร์แรนด์

A - ข้อมูลของแอดเดรส 400H ในหน่วยความจำ

ในภาษาแอสเซมบลี้จะแสดงวิธีการอ้างแอดเดรสแบบนี้ในรูปของ (NN) โดยที่ NN จะแทนแต่ละไบต์ในรูปของตัวเลขที่อยู่ภายในวงเล็บ และการใช้วงเล็บ หมายถึง ข้อมูลภายในหน่วยความจำตรงตำแหน่งแอดเดรสที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีของข้อมูลความยาว 2 ไบต์ สามารถอ้างแอดเดรสได้ด้วยวิธีนี้ โดยที่ข้อมูลที่ต้องการจะอยู่ในหน่วยความจำ 2 แอดเดรส ดังตัวอย่าง

LD HL, (5000H) ; 2A 00 50

ออปโค้ด โอเปอร์แรนด์

L - ข้อมูลของแอดเดรส 5000H ในหน่วยความจำ

H - ข้อมูลของแอดเดรส 5001H ในหน่วยความจำ

(3) REGISTER INDIRECT ADDRESSING (การอ้างทางอ้อมโดยใช้รีจิสเตอร์ช่วย) โดยการใช้รีจิสเตอร์คู่ จะสามารถอ้างแอดเดรสขนาด 16 บิตได้ซึ่งในกรณีนี้ข้อมูลในรีจิสเตอร์คู่จะเป็นตัวชี้ถึงแอดเดรสของหน่วยความจำที่มีข้อมูลที่ต้องการ ดังตัวอย่าง

LD A, (HL) : 7E

ในกรณีนี้ข้อมูลจะมีความยาว 1 ไบต์เท่านั้น สำหรับการอ้างแอดเดรสที่ต้องการข้อมูลความยาวตั้งแต่ 2 ไบต์ขึ้นไปนั้น จะกระทำได้โดยการเพิ่มค่าหรือบวกตัวรีจิสเตอร์คู่เข้าไปทีละ 1 แล้วใช้คำสั่งในการอ้างแอดเดรสแบบเดิมเป็นเช่นนี้ไปเรื่อย ๆ

การอ้างแอดเดรสวิธีนี้จะให้ผลดีในการคำนวณที่ต้องการใช้ตัวแปร ซึ่งสามารถอ้างแอดเดรสตัวแปรได้ โดยที่ก่อนจะทำการประมวลผล ก็ทำการเซตหรือกำหนดแอดเดรสตัวแปรไว้ในรีจิสเตอร์คู่เพราะฉะนั้นจึงสามารถใช้โปรแกรมเดิมทำการประมวลผลในงานเดียวกับที่มีการเปลี่ยนตัวแปรหลาย ๆ ของการทำงาน (ดูรูปที่ 1.17)

(4) RELATIVE ADDESSING (การอ้างโดยอาศัยความสัมพันธ์)

ในการอ้างแอดเดรสของหน่วยความจำ Z80 นั้น โดยปกติจะแสดงเนื้อที่ของแอดเดรสจำนวน 64K ไบต์ ในรูปของ ABSOLUTE ADDRESS ทั้งหมด ซึ่งจำเป็นต้องใช้มูลขนาด 16 บิตเป็นตัวกำหนด และในการใช้คำสั่ง JUMP นั้นตำแหน่งของส่วนที่ต้องการกระโดยไปนั้น มักจะมีแอดเดรสอยู่ใกล้ ๆ หรือไม่ไกลจากแอดเดรสของคำสั่งนั้น

ในกรณีนี้ เรามีวิธีการอ้างแอดเดรสของส่วนที่ต้องการกระโดดไปโดยอาศัยความสัมพันธ์ (RELATIVE) กับแอดเดรสของคำสั่ง JUMP นั้น ซึ่ง RELATIVE ADDRESS ที่ว่านี้จะใช้เนื้อที่เพียง 1 ไบต์ ทำให้ประหยัดเนื้อที่ของหน่วยความจำ สำหรับโปรแกรมลงไปได้ 1 ไบต์ เพราะถ้าเขียนด้วย ABSOLUTE ADDRESS จะต้องใช้ 2 ไบต์ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1000 : C3 90 10 JP 1009H

แต่ถ้าเป็นแบบ RELATIVE ADDRESS จะเขียนได้ดังนี้

1000 : 18 07 JP 1009H

โดยที่ 07 ในที่แสดงถึงระยะห่างของแอดเดรสที่จะกระโดดไปเป็นระยะทาง 9 แอดเดรสจากแอดเดรสเดิม (1000)

ด้วยการใช้ RELATIVE ADDRESS นี้ เราสามารถทำให้โปรแกรมขนาดเล็ก ๆ เป็นโปรแกรมในลักษณะ RELOCATABLE ได้ นั่นคือ ถ้าไม่มีการอ้างถึง ABSOLUTE ADDRESS จะสามารถกำหนดแอดเดรสของโปรแกรมไว้ที่ตำแหน่งใด ๆ ในหน่วยความจำก็ได้ เราเรียกโปรแกรมแบบนี้ว่า "RELOCATABLE PROGRAM" วิธีการนี้ใช้ได้เฉพาะ Z80 เท่านั้นจะใช้กับอินเทล 8085A หรือ 8080Aไม่ได้

(5) INDEXED ADDRESSING (การอ้างแอดเดรสโดยใช้วิธีรีจิสเตอร์อินเด็กซ์) การใช้โหมดของอินเด็กซ์หรือรีจิสเตอร์ IX และ IY ทำให้เราสามารถอ้างแอดเดรสใด ๆ ในเนื้อที่ของหน่วยความจำขนาด 64K ไบต์ของ Z80 ได้ด้วยรีจิสเตอร์ IX และ IY เช่น

LD A, (IX+05H) : DD 7E 05

ออปโค๊ด โอเปอร์แรนด์

ถ้าหากว่า IX มีค่าเป็น 1000H แอดเดรสที่อ้างถึงคือ 1000H+05H = 1005H นั่นหมายความว่า A จะมีค่าเท่ากับข้อมูลที่อยู่ในแอดเดรส 1005H ของหน่วยความจำ

เมื่อเขียนเป็นโปรแกรมย่อย

โดยปกติแล้วซับรูทีน (SUBROTINE) หรือโปรแกรมย่อยจะเป็นส่วนของโปรแกรมที่สร้างขึ้นเพื่อการทำงานเฉพาะอย่างใดอย่างหนึ่งโดยแต่ละซับรูทีนจะรวมอยู่ในเมนโปรแกรม (MAIN PROGRAM) หรือโปรแกรมหลัก และถูกเรียกมาใช้เมื่อใดก็ได้

เมื่อมีการเรียก (CALL) ซับรูทีนมาใช้จากส่วนใดของเมนโปรแกรมก็ตาม จะต้องมีการให้ค่าข้อมูลบางตัวที่จำเป็นต่อการทำงานของซับรูทีนนั้น ๆ จากเมนโปรแกรมด้วยและเมื่อซับรูทีนทำงานเสร็จแล้ว ผลลัพธ์ที่ได้ก็จะย้อนกลับมาที่เมนโปรแกรมอีกครั้งหนึ่งการส่งและรับค่าข้อมูลที่ว่านี้ กระทำในรีจิสเตอร์ของ CPU แต่สำหรับภาษาระดับสูงจะกระทำโดยอาศัยสแตค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

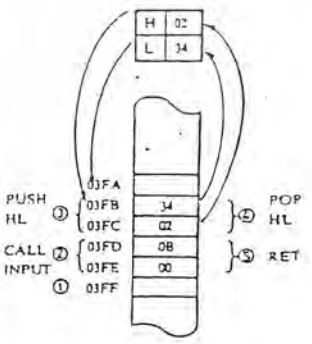
การทำงานของซับรูทีน

ในการเรียกซับรูทีนจะใช้คำสั่ง CALL ซึ่งเมื่อมีการใช้คำสั่ง CALL นั้นจะมีการเพิ่มค่าที่โปรแกรมเคาน์เตอร์ (PC) ที่ชี้ตำแหน่งแอดเดรสของคำสั่ง CALL โดยเพิ่มเข้าไปอีก 3 ซึ่งเป็นการอ้างถึงแอดเดรสของคำสั่งที่จะทำงานต่อไปหลังจากการทำงานของคำสั่ง CALL สิ้นสุดแอดเดรสที่กำหนดขึ้นใหม่โดยการเพิ่มค่า 3 เข้าไปนี้จะถูกเก็บไว้ที่สแตคก่อน หลังจากนั้นการทำงานของโปรแกรมก็จะกระโดดไปที่จุดเริ่มต้นของซับรูทีนที่กำหนดโดยตัวโอเปอร์แรนด์ของคำสั่ง CALL นั้น

การเรียกใช้ซับรูทีนนี้บางครั้งอาจจะเกิดการทำลายข้อมูลในรีจิสเตอร์ของส่วนเมนโปรแกรม ดังนั้นจึงจำเป็นต้องมีการเก็บรักษาข้อมูลดังกล่าวก่อนโดยการใช้คำสั่ง PUSH เพื่อเก็บรักษาข้อมูลนั้นไว้ในสแตคเสียก่อน และเมื่อการทำงานของซับรูทีนสิ้นสุดลงข้อมูลต่าง ๆ ในรีจิสเตอร์ที่ถูกเก็บรักษาไว้ชั่วคราวในสแตคก็จะถูกนำกลับมาใช้งานอีกโดยการใช้คำสั่ง POP และการทำการทำงานทั้งหมดจะกลับไปที่เมนโปรแกรมด้วยคำสั่ง RET ซึ่งการทำงานจะเริ่มต้นอีกครั้งที่ตำแหน่งที่กำหนดไว้ที่โปรแกรมเคาน์เตอร์ที่เพิ่มค่าเข้าไปอีก 3 ซึ่งอยู่ในสแตค

กล่าวโดยสรุปแล้ว ฝ่ายที่ถูกเรียกซับรูทีนจะเป็นตัวกำหนดแอดเดรสของตำแหน่งที่จะย้อนกลับมาโดยเก็บไว้ที่สแตค และแอดเดรสนี้จะถูกดึงกลับมาที่โปรแกรมเคาน์เตอร์เมื่อใช้คำสั่งRET ดังนั้นที่ซับรูทีนจะเพียงแต่มีคำสั่ง RET ในขั้นตอนสุดท้ายของการทำงานก็จะย้อนกลับไปที่เมนโปรแกรมได้

```
แอดเดรส  กลางเครื่อง   คำสั่ง              การทำงาน
0000     31 FF 03       LD SP, 03FFH  ;  กำหนดแอดเดรสเริ่มต้นของสแตค①
0003     21 34 02       LD HL, 0234H  ;  H = 02, L = 34
0006     3E 00    LOOP: LD A, 00H     ;  A = 00
0008     CD 0E 00       CALL INPUT    ;  แอดเดรสของคำสั่งถัดไป (คือ 000BH) ถูกเก็บไว้ในสแตค② ก่อนที่จะกระโดดไปที่ INPUT (ขณะนี้ SP มีค่า 03FDH)
000B     77             LD (HL), A    ;  (HL) = A หรือ (0234H) = 00H
000C     18 FB          JR LOOP       ;  กระโดดไปที่ LOOP
000E     E5       INPUT: PUSH HL       ;  เก็บข้อมูลของ HL ไว้ในสแตค③(ขณะนี้ SP มีค่า 03FBH)
000F     DB FF          INA, (0FFH)   ;  อ่านข้อมูลจากพอร์ตอินพุตแอดเดรสที่ FF แล้วเก็บไว้ที่รีจิสเตอร์ A
0011     E1             POP HL        ;  นำข้อมูลของ HL ที่อยู่ในสแตคกลับมาใช้งานอีกครั้ง④ (ขณะนี้ SP มีค่า 03FDH)
0012     C9             RET           ;  นำแอดเดรสของคำสั่งที่จะทำงานในช่วงต่อไป ออกจากสแตคไปคืนให้กับ PC พร้อมกลับย้อนกลับไปที่เมนโปรแกรม⑤ (ขณะนี้ SP มีค่า 03FFH)
```



รูปที่ 1.18 การทำงานของสแตคตามคำสั่งต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ R_{50}-R_{57} ให้ติดตั้งบนแผ่นวงจรพิมพ์ด้านเดียวกับ LED ส่วนแผ่นกรองแสงก็ติดตั้ง โดยครอบปิดลงบน LED ได้เลย

การปรับแต่ง

ส่วนที่ใช้ในการปรับแต่งจะมีอยู่ 2 ส่วนคือ ซอฟตต์แวร์และฮาร์ดแวร์ โดยจะใช้ซอฟต์แวร์ในการปรับแต่งก็ต่อเมื่อตัวอักษรที่ปรากฏออกมามีลักษณะสั่นไม่นิ่งหรือเห็น LED ติดทีละแถว ก็ให้ทำการเปลี่ยนช่วงการหน่วงเวลาที่ทำให้ LED ติดค้างชั่วขณะเสียใหม่จนไม่เกิดอาการดังกล่าว

ส่วนการปรับแต่งโดยใช้ฮาร์ดแวร์จะกระทำก็ต่อเมื่อมีการใช้สัญญาณอินเตอร์รัพต์ และตัวอักษรที่ปรากฏออกมามีลักษณะพริ้ว ก็ให้ทำการปรับ VR_1 ตัวอักษรที่ปรากฏออกมาไม่เกิดอาการพริ้วก็เป็นอันเรียบร้อย

การใส่จั๊มเปอร์

ในการใช้งาน ET-LDB นี้ต้องมีการใส่จั๊มเปอร์ในตำแหน่งต่างๆ ดังต่อไปนี้

- ☐ ไม่ใช้สัญญาณอินเตอร์รัพต์
- ■ ใช้สัญญาณอินเตอร์รัพต์
- ☐ ไม่ใช้วอตช์ด็อกช์
- ■ ใช้วอตช์ด็อกช์รีเซตด้วยสถานะ "0" สำหรับ Z80
- ■ ใช้วอตช์ด็อกช์รีเซตด้วยสถานะ "1" สำหรับ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำย่อและสัญลักษณ์ที่ใช้ในภาษาแอสเซมบลี้ของ Z80

คำย่อ

(1) ชื่อของรีจิสเตอร์ขนาด 8 บิต : A, B, C, D, E, H, L, I, R

(2) ชื่อของรีจิสเตอร์ขนาด 16 บิต: IX, IY, SP, AF, BC, DE, HL (รวมคู่ของรีจิสเตอร์ 8 บิตด้วย)

(3) ชื่อของคู่รีจิสเตอร์สำรอง : AF, BC, DE, HL

(4) สถานะของแฟลก : C (CARRY), NC (NON-CARRY), Z (ZERO), NZ (NON-ZERO), M (MINUS), P (PLUS), PE (PARITY EVEN), PO (PARITY ODD)

สัญลักษณ์ของโอเปอร์แรนด์

สัญลักษณ์ที่ใช้กับโอเปอร์แรนด์ ได้ถูกกำหนดไว้ดังต่อไปนี้

(1)R : ใช้กำหนดแทนรีจิสเตอร์ A, B, C, D, E, H, L ตัวใดตัวหนึ่ง

(2)(HL) : ชี้ถึงหน่วยความจำที่ถูกกำหนดโดยข้อมูลภายในคู่รีจิสเตอร์ HL

(3)N : คือข้อมูลขนาด 1 ไบต์ โดย $0 \leq N \leq 225$

(4)NN : คือข้อมูลขนาด 2 ไบต์ โดย $0 \leq NN \leq 65535$

(5)D : คือข้อมูลขนาด 1 ไบต์ โดย $-128 \leq D \leq 127$

(6)(NN) : ชี้ถึงหน่วยความจำที่ถูกกำหนดด้วยค่าของ NN

(7)(N) : ชี้ถึงพอร์ตอินพุต-เอาต์พุต ที่ถูกกำหนดโดยค่าของ N

(8)B : มีค่าจาก 0 ~ 7

(9)E : คือข้อมูลขนาด 1 ไบต์ โดย $-126 \leq E \leq 129$

(10)CC : แสดงถึงสถานะภาพของแฟลกที่เกี่ยวเนื่องกับคำสั่ง RET, CALL หรือ JP โดยมีเงื่อนไขควบคู่กับคำสั่งนั้นด้วย

(11)QQ : หมายถึงคู่รีจิสเตอร์ BC, DE, HL หรือ AF คู่ใดคู่หนึ่ง

SS : หมายถึงคู่รีจิสเตอร์ BC, DE, HL หรือ SP คู่ใดคู่หนึ่ง

PP : หมายถึงคู่รีจิสเตอร์ BC, DE, IX หรือ SP คู่ใดคู่หนึ่ง

RR : หมายถึงคู่รีจิสเตอร์ BC, DE, IY หรือ SP คู่ใดคู่หนึ่ง

DD : หมายถึงคู่รีจิสเตอร์ BC, DE, HL หรือ SP คู่ใดคู่หนึ่ง

(12)S : หมายถึง R, N, (HL), (IX+D), (IY+D) ตัวใดตัวหนึ่ง

(13)M : หมายถึง R, (HL), (IX+D) หรือ (IY+D) ตัวใดตัวหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาร์ดแวร์ของไมโครโปรเซสเซอร์

ไมโครคอมพิวเตอร์แผ่นพิมพ์เดียว (SINGLE BOARD MICROCOMPUTER)

การที่จะสร้างระบบไมโครคอมพิวเตอร์ โดยใช้ Z80 เป็น CPU จำเป็นต้องมีส่วนประกอบอย่างน้อยที่สุดดังแสดงในรูปที่ 1.19 ซึ่งได้แก่ CPU, ROM, พอร์ตอินพุท - เอาต์พุท แบบขนาน (PIO) และไอซี TTL แต่ก็สามารถนำไปใช้กับงานสำหรับจัดการกับข้อมูลที่ได้รับการแปลงรูปและส่งผลออกไปได้

แต่อย่าลืมว่า ภายใน Z80 ยังมีรีจิสเตอร์อยู่จำนวนมาก ซึ่งสามารถดัดแปลงไปใช้เป็นบัฟเฟอร์สำหรับการเก็บข้อมูล โดยมีให้ใช้ถึง 14 ไบต์

สำหรับระบบไมโครคอมพิวเตอร์ขนาดเล็กที่มีการเพิ่ม RAM เข้าไปด้วยแสดงไว้ในรูปที่ 1.20 การเพิ่ม RAM เข้าไปในระบบนี้เป็นการเพิ่มความสามารถ ทำให้เก็บข้อมูลชั่วคราวไว้ในหน่วยความจำได้เป็นจำนวนมากขึ้น นอกจากนั้นยังสามารถใช้บางส่วนของ RAM เป็นสแตคในการเขียนโปรแกรมเพื่อให้สามารถเรียกโปรแกรมย่อยหรือใช้คำสั่ง PUSH และ POP ได้อีกด้วย

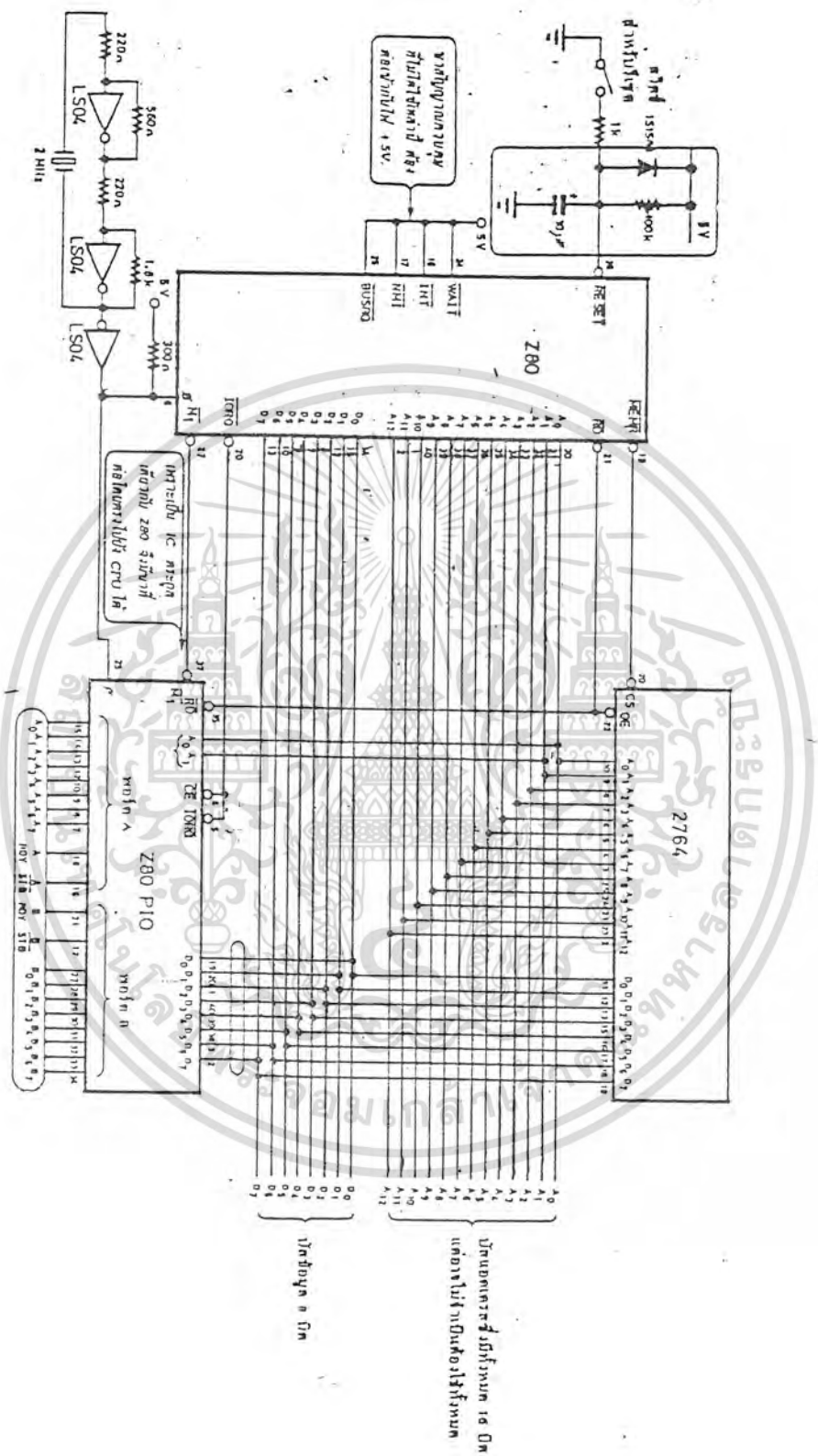
ส่วนประกอบสำคัญในรูปที่ 1.20 แบ่งออกเป็นส่วนๆได้ดังนี้คือตัว CPU, ส่วนกำเนิดสัญญาณนาฬิกาจาก TTL, ROM, RAM และพอร์ตอินพุท-ทั้งแบบอนุกรม SERIAL และแบบขนาน PARALLEL รวมทั้งเกตต่าง ๆ

ในปัจจุบันได้มีการพัฒนาหน่วยความจำ ROM และ RAM ให้มีความจุต่อตัวได้สูงขึ้น ในกรณีของรูปที่ 1.20 สามารถสร้างขึ้นเป็นไมโครคอมพิวเตอร์แผ่นพิมพ์เดียวได้ โดยใช้ ROM ขนาด 8 กิโลไบต์อย่างละ 1 ตัวเท่านั้น และจะเห็นได้ว่าระบบเล็ก ๆ เช่นนี้อุปกรณ์ที่เชื่อมต่อกับอุปกรณ์รอบข้างของ CPU จึงยังคงสามารถทำให้อุปกรณ์รอบข้างทำงานได้ ซึ่งเป็นการประหยัดวงจรบัฟเฟอร์

นอกจากนั้นในส่วนของการถอดรหัสเลือกตำแหน่งของหน่วยความจำและของพอร์ต I/O ไม่จำเป็นต้องใช้บัสแอดเดรสทั้ง 16 บิต(สำหรับหน่วยความจำ)และ 8 บิต สำหรับพอร์ต I/O จากรูปที่ 1.20 จะเห็นว่ามีหน่วยความจำอยู่ 2 ตัว คือ ROM กับ RAM และพอร์ต I/O มี 2 ตัวเช่นกันคือ 8255 กับ 8251 การเลือกตำแหน่งของ ROM กับ RAM ถูกแยกกันด้วย A_{15} เพียงเส้นเดียวส่วนการเลือกตำแหน่งของพอร์ต I/Oใช้ A_0 สำหรับเลือก 8255 และ A_7เป็น "0" เมื่อทำการเลือก

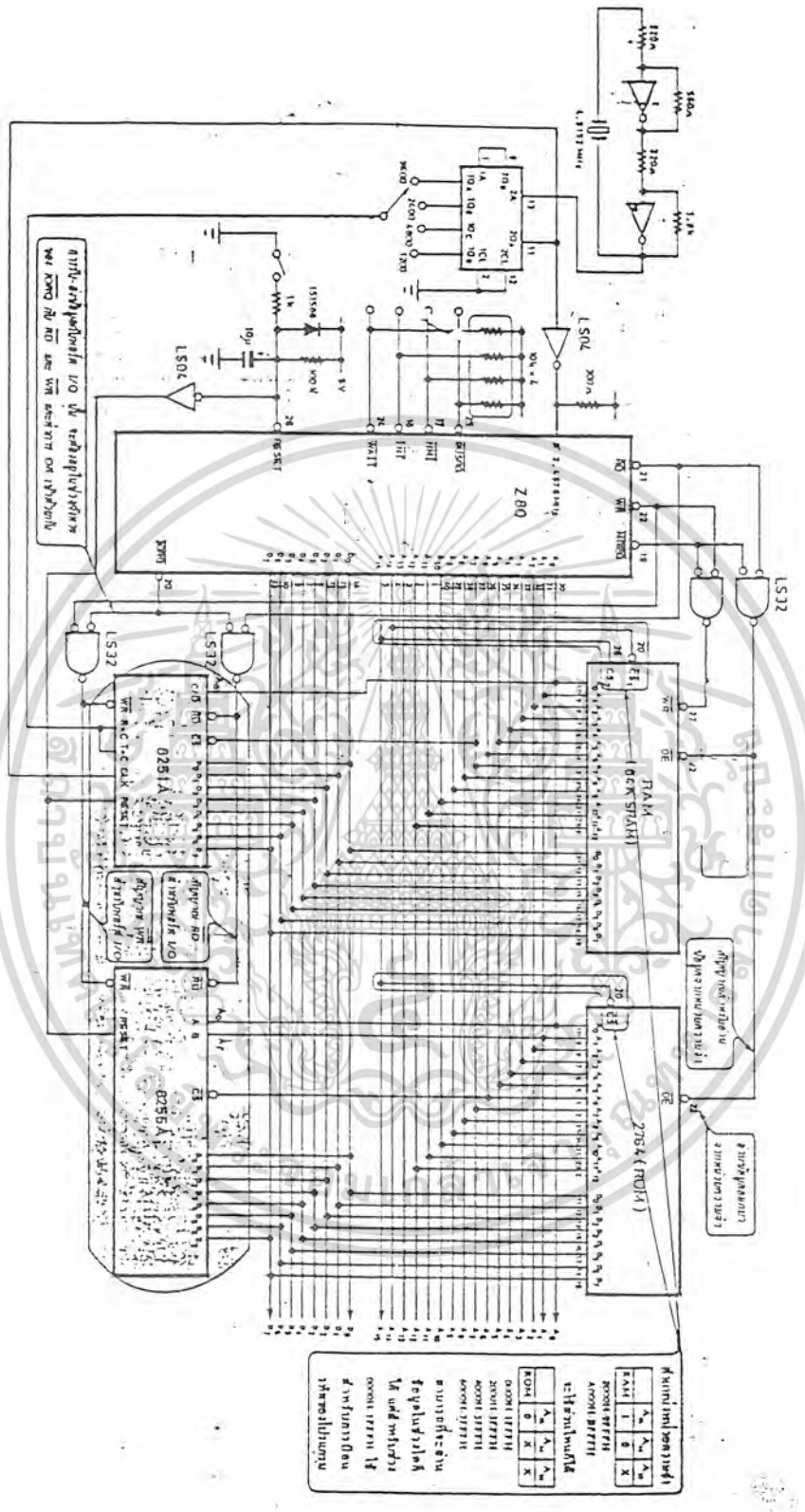
รูปที่ 1.21 แสดงตำแหน่งของพอร์ต I/O เมื่อใช้ A_2-A_7 เพียงเส้นใดเส้นหนึ่งต่อเข้ากับขา CS ของพอร์ต I/O โดยตรง และไม่จำเป็นต้องใช้ส่วนถอดรหัสแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.19 ส่วนประกอบอย่างน้อยที่สุดของระบบไมโครคอมพิวเตอร์ที่ใช้ Z80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.20 ระบบไมโครคอมพิวเตอร์ของ Z80 ที่มีขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาแอดเดรสที่ต่อกับ	จะมีขาใดขาหนึ่งที่ต่อเข้ากับ CS ของพอร์ต I/O						ต่อเข้ากับขาเลือกรีจิสเตอร์ในพอร์ต I/O		แอดเดรสของพอร์ตต่างๆ
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
A_2	1	1	1	1	1	0	0	0	F8H
							0	1	F9H
							1	0	FAH
							1	1	FBH
A_3	1	1	1	1	0	1	0	0	F4H
							0	1	F5H
							1	0	F6H
							1	1	F7H
A_4	1	1	1	0	1	1	0	0	ECH
							0	1	EDH
							1	0	EEH
							1	1	EFH
A_5	1	1	0	1	1	1	0	0	DCH
							0	1	DDH
							1	0	DEH
							1	1	DFH
A_6	1	0	1	1	1	1	0	0	BCH
							0	1	BDH
							1	0	BEH
							1	1	BFH
A_7	0	1	1	1	1	1	0	0	7CH
							0	1	7DH
							1	0	7EH
							1	1	7FH

รูปที่ 1.21 การเลือกตำแหน่งของพอร์ตอินพุทเอาต์พุท โดยไม่ต้องใช้วงจรถอดรหัสสัญญาณแอดเดรส

การเลือกตำแหน่ง เช่น ของ ROM มีตำแหน่งอยู่ที่ 0000H- FFFH หรือ 2000H-3FFFH หรือ 6000H-7FFH ก็ได้ และสำหรับ 8255 มีตำแหน่งอยู่ที่ BCH-BFH หรือ 80H - 83H ก็ได้ จะสังเกตว่าที่แอดเดรสน้อยกว่า 800H เป็นตำแหน่งของ ROM และที่แอดเดรสมากกว่า 800H เป็นตำแหน่งของ RAM ในส่วนของพอร์ต I/O จะสังเกตได้ว่าที่แอดเดรสน้อยกว่า 80H เป็นการเลือก 8251 และที่แอดเดรสมากกว่า 80H เป็นการเลือก 8255 ลักษณะเช่นนี้ไม่ทำให้เกิดปัญหาใด ๆ เมื่อเขียนโปรแกรม แต่อาจจะทำให้เกิดการสับสนได้ เพราะสามารถอ้างแอดเดรสได้หลายตำแหน่ง

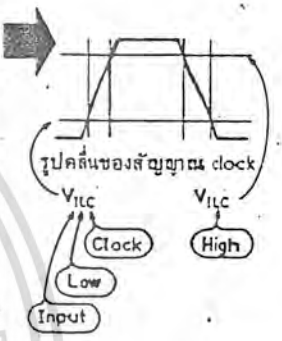
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณลักษณะทางไฟฟ้าของ Z80

คุณลักษณาทางไฟตรง(DC.CHARACTERISTICS) ของขาต่าง ๆของ Z80ได้แสดงไว้ในรูปที่ 1.22 คุณลักษณะทางไฟตรงก็คือของระดับลอจิก "H" และ "L" ค่าของกระแสเข้าและค่าของกำลังเอาต์พุท ส่วนเรื่องของช่วงเวลาและการรับ-ส่งข้อมูลนั้นเป็นคุณสมบัติทางไฟสลับ (AC.CHARACTERISTICS)

(T_A = 0°C ~ +70°C, V_{CC} = +5V ±5%)

สัญลักษณ์	ความหมาย	ต่ำสุด	สูงสุด	หน่วย	ข้อกำหนดในการวัด
V_{ILC}	แรงดันอินพุต "L" ของ clock	−0.3	0.45	V	
V_{IHC}	แรงดันอินพุต "H" ของ clock	V_{CC}−0.6	V_{CC}+0.3	V	
V_{IL}	แรงดันอินพุต "L"	−0.3	0.8	V	
V_{IH}	แรงดันอินพุต "H"	2.0	V_{CC}	V	
V_{OL}	แรงดันเอาต์พุต "L"		0.4	V	I_{OL} = 1.8mA
V_{OH}	แรงดันเอาต์พุต "H"	2.4		V	I_{OH} = −250μA
I_{CC}	กระแสไฟที่ใช้		150	mA	
I_{LI}	กระแสรั่วของขาอินพุต		10	μA	V_{IN} = 0 ~ V_{CC}
I_{LOH}	กระแสรั่วของขาเอาต์พุต 3 สเตต ("H")		10	μA	V_{OUT} = 2.4V ~ V_{CC}
I_{LOL}	กระแสรั่วของขาเอาต์พุต 3 สเตต ("L")		−10	μA	V_{OUT} = 0.4V
I_{LD}	กระแสรั่วของบัสข้อมูลเมื่อทำการอินพุต		±10	μA	0 ≤ V_{IN} ≤ V_{CC}



หมายเหตุ

- *เพราะส่วนรับสัญญาณ clock นั้น ไม่ได้ทับมาตรฐานของ TTL ดังนั้นเมื่อนำสัญญาณ clock จากวงจรกำเนิดสัญญาณมาพิจารณาที่ใช้ TTL จะต้องทำการพูลอัพ (pull up) เข้ากับ V_{cc} ด้วยตัวต้านทานค่าประมาณ 300 Ω*
- *กระแสขับ (drive current) ทางเอาต์พุตของขาต่าง ๆ มีความสามารถเท่ากับ LS-TTL เพียง 4 ตัวเท่านั้น ดังนั้นสำหรับระบบใหญ่ ๆ จะต้องมีการเพิ่มบัฟเฟอร์ให้กับบัสแต่ละเส้น*

รูปที่ 1.22 คุณลักษณะทางไฟตรง (DC.CHARACTERISTIC) ที่ขาสัญญาณต่าง ๆ ของ Z80

รูปที่ 1.23 ลักษณะการรับแรงดันอินพุทและการให้แรงดันเอาต์พุทที่ขาต่าง ๆ ของ Z80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก Z80 มีโครงสร้างเป็นชนิด CMOS ซึ่งคุณลักษณะทางขาอินพุทจะมีค่าอิมพีแดนซ์สูงมากเมื่อต่อเข้ากับเอาต์พุทของไอซีประเภท TTL จึงไม่มีปัญหาในเรื่องของกระแสอินพุท-เอาต์พุท แต่จะมีปัญหาที่ความเร็วจะลดต่ำลงถ้าหากค่าความจุไฟฟ้าที่ขาอินพุทของ Z80 มีค่าเพิ่มขึ้น

แต่ในกรณีของขาเอาต์พุทของ Z80 ซึ่งต่อเข้ากับขาอินพุทของ TTL เนื่องจากคุณลักษณะของกระแสที่รับเข้าที่ระดับลอจิก "L" มีค่าน้อย ดังนั้นตัว Z80 จึงต้องควบคุมให้ TTL ทำงานได้เพียงตัวเดียวในขณะเวลาใดเวลาหนึ่ง หมายความว่า Z80 จะติดต่อกับอุปกรณ์รอบข้างเพียงตัวเดียวในขณะเวลาหนึ่ง ๆ

ตารางที่ 4 แสดงคุณลักษณะทางอินพุท - เอาต์พุทของไอซี TTL ประเภทต่าง ๆ ค่าที่น่าสนใจก็คือค่าของกระแส I_{oL} (กระแสเอาต์พุทเมื่อเป็นลอจิก "L") ค่ากระแส I_{oh} (กระแสเอาต์พุทเมื่อเป็นลอจิก "H" ตามลำดับ และต้องตรวจสอบค่าแรงดันเอาต์พุทนี้ว่าสมดุลกับส่วนของอินพุตที่มาต่ออยู่หรือไม่ ดังแสดงคุณลักษณะของแรงดันอินพุท-เอาต์พุทของ Z80 ไว้ในรูปที่ 1.20

คุณลักษณะทางไฟฟ้าในรูปที่ 1.19 ช่วยให้ตรวจสอบความสามารถของสัญญาณเอาต์พุทของ Z80 ว่ามีความเพียงพอที่จะทำให้อุปกรณ์อื่น ๆ ในระบบทำงานได้หรือไม่ ถ้าสัญญาณที่ส่งไปตามบัสไม่มีความสามารถพอก็จะต้องเพิ่มส่วนบัฟเฟอร์ เพื่อเป็นการเพิ่มประสิทธิภาพในการทำงานให้อุปกรณ์รอบข้างทำงานได้

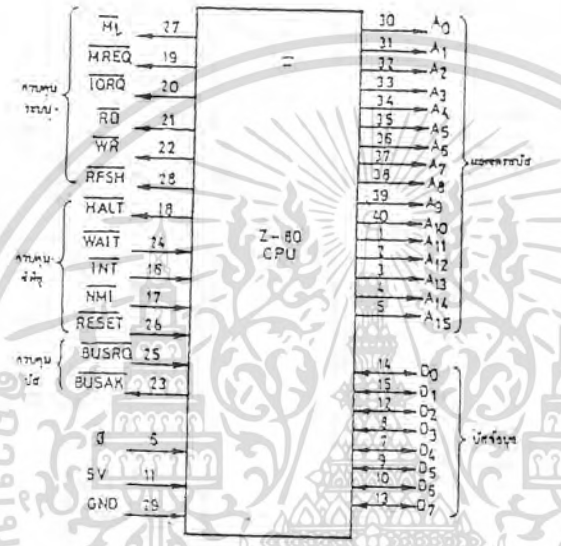
รายละเอียด / ชนิด	ค่าโหลดอินพุทของขาอินพุท (kΩ)	กระแสอินพุทสูงสุดที่ "H" $I_{IH}(\mu A)$	กระแสอินพุทสูงสุดที่ "L" $I_{IL}(mA)$	เวลาหน่วงของเกต (ns)	กำลังงานสูญเสียของแต่ละเกต (mW)	ความถี่ที่ใช้ได้ (MHz)	กระแสเอาต์พุทสูงสุดที่ "H" $I_{OH}(\mu A)$	กระแสเอาต์พุทสูงสุดที่ "L" $I_{OL}(mA)$
54/74	4	40	−1.6	10	10	DC−35	−400	16
54H/74H	2.8	50	−2	6	22	DC−50	−500	20
54L	40	10	−0.18	33	1	DC−3	−100	2
	8	20	−0.8					
54LS/74LS	18	20	−0.4	9.5	2	DC−45	−400	4/8
54S/74S	2.8	50	−2				−1000	20
54ALS/74ALS	40	20	−0.2	4	1	DC−100	−400	4/8
54AS/74AS	2.7	200	−2	1.5	22	DC−200	−2000	20
ตระกูล 74F ของบริษัท Fairchild	-	20	−0.6	3.0	4	DC−100	−1000	20

ตารางที่ 4 คุณลักษณะของอินพุทและเอาทในมาตรฐานของไอซี TTL แบบต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดขาของ Z80

Z80 ซีพียูเป็นไอซีไมโครโปรเซสเซอร์ที่มีขาเพียง 40 ขา โดยหลักการแล้ว Z80 เป็นซีพียูได้โดยสมบูรณ์ กล่าวคือ Z-80 ไม่ต้องประกอบกับอุปกรณ์อื่นที่แยกการทำงานเพื่อรวมเป็นซีพียู ส่วนของสัญญาณจะประกอบด้วยบัสแอดเดรส บัสข้อมูลและสัญญาณควบคุมการจัดขาแสดงในรูปที่ 1.24



รูปที่ 1.24 ลักษณะของขาไอซี Z80 ซีพียู

รายละเอียดของขาต่าง ๆ แสดงได้ดังต่อไปนี้

A0-A15 บัสแอดเดรสสัญญาณที่ออกมาจากไอซีเหล่านี้จะให้แอดดีฟชณะ HIGH โดยขาเหล่านี้เป็นเอาต์พุทแบบไตรสเตรท บัสแอดเดรสมีด้วยกันทั้งหมด 16 สาย เพื่อให้ซีพียูติดต่อกับหน่วยความจำได้ถึง 2=64K ไบต์ นอกจากนี้ส่วนของแอดเดรสยังเป็นตัวกำหนดเบอร์พอร์ตของอุปกรณ์อินพุทเอาต์พุทโดยขณะที่ซีพียูกระทำคำสั่งเกี่ยวกับอินพุทหรือเอาต์พุท ค่าของแอดเดรสบัสในบิทล่าง(A0-A7) จะแสดงค่าเบอร์พอร์ต ดังนั้นเราจึงมีอุปกรณ์อินพุทหรือเอาต์พุทได้ทั้งหมด 2=256 พอร์ตและในขณะช่วงเวลารีเฟทช์ โดยเมื่อสัญญาณรีเฟทช์ปรากฏขึ้นที่รีเฟรช์ (RFSH) ค่าในแอดเดรสบัส A0-A7 จะแสดงค่าแอดเดรสของหน่วยความจำที่จะได้รับการกระทำการรีเฟทช์

D0-D7 บัสข้อมูล (DATA BUS) เป็นลักษณะบัสแบบสองทิศทาง Z80 ซีพียูมีบัสข้อมูล 8 เส้นบัสข้อมูลเป็นเส้นทางผ่านของข้อมูลระหว่างซีพียูกับหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำ ซีพียูกับอุปกรณ์อินพุทเอาต์พุทกับหน่วยความจำ

M1 (MACHINE CYCLE ONE) มีลักษณะเป็นแอคตีฟที่ลอจิก "0" เป็นส่วนที่จะบอกให้ทราบว่าขณะนี้ซีพียูกำลังอยู่ในสภาวะเฟทช์ ในขณะที่ซีพียูเฟทช์คำสั่งที่มีออปโค้ดสองไบต์ส่วนของ M1 จะสร้างชั้นขณะเฟทช์ในแต่ละไบต์ลักษณะของคำสั่งที่มีออปโค้ดสองไบต์จะขึ้นต้นด้วย CBH,DDH,EDH,FDH นอกจากนี้ M1 ยังสร้างสัญญาณร่วมกับ IORQ เพื่อบอกสถานะการตอบรับการอินเตอร์รัพท์

MREQ (MEMORY REQUEST) เป็นลักษณะไตรสเตทในลอจิกแอคตีฟที่ "0" เป็นสัญญาณที่บอกให้ทราบซีพียูต้องการเขียนหรืออ่านหน่วยความจำตามแอดเดรสที่ปรากฏอยู่ในแอดเดรสบัส

IORQ (INPUT OUTPUT REQUEST) เป็นเอาต์พุทลักษณะไตรสเตทให้ลอจิกแอคตีฟที่ "0" เป็นสายสัญญาณที่บอกให้ทราบว่า ซีพียูต้องการติดต่อกับอุปกรณ์อินพุท-เอาต์พุทโดยแอดเดรสบัส 8 บิท ล่างจะให้แสดงค่าเบอร์พอร์ตส่วนบัสข้อมูลจะมีการส่งถ่ายระหว่างซีพียูกับ I/O นอกจากนี้ ถ้าเกิดขึ้นพร้อมสัญญาณ M1 เป็นตัวบอกถึงสถานะที่ซีพียูกำลังตอบสนองผลการอินเตอร์รัพท์ โดยขณะนี้ส่วนของบัสข้อมูลจะมีการส่งผ่านเข้ามาด้วยค่าของอินเตอร์รัพท์เวคเตอร์

RD (MEMORY READ) เป็นเอาต์พุทที่ไตรสเตรทและแอคตีฟขณะลอจิก "0" เป็นตัวบอกว่าขณะนี้ซีพียูต้องการผ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์ WR (MEMORY WRITE)เป็นขาเอาต์พุทแบบไตรสเตทและแอคตีฟขณะลอจิก "0"เป็นสัญญาณบอกว่าซีพียูต้องการเขียนข้อมูลโดยจะเขียนข้อมูลในตำแหน่งที่ ADDRESS BUS กำหนดขึ้นอาจจะเป็นหน่วยความจำหรืออุปกรณ์ I/O ก็ได้

RESH(REFFRESH)เป็นขาเอาต์พุทแอคตีฟขณะลอจิก "0" RESH เป็นสัญญาณบอกให้ทราบว่าสัญญาณในแอดเดรสบัสในส่วน A0-A6 เป็นแอดเดรสที่ใช้ในการรีเฟรช์หน่วยความจำแบบไดนามิกส่วนบิทจะเป็น "0" ส่วน A15-A8 จะแสดงค่าของรีจิสเตอร์ I

HALT(HALT STATE)เป็นเอาต์พุทที่แอคตีฟด้วยลอจิก "0" สัญญาณจะแสดงเมื่อซีพียูได้กระทำคำสั่ง HALT และจะหยุดรอจนกว่าจะมีการอินเตอร์รัพท์หรือรีเซทขณะที่อยู่ในช่วง HALTซีพียูจะเหมือนจะทำคำสั่งNOP(NO OPERTION) เพื่อให้เกิดไซเคิลในการทำงาน เพื่อสัญญาณไปกระทำการรีเฟรช์หน่วยความจำชนิด ไดนามิกส์

WAIT (WAIT) เป็นขาอินพุทจะแอคตีฟด้วยลอจิก "0" WAIT เป็นตัวกำหนดแสดงเพื่อบอก ซีพียูให้ซีพียูรอในกรณีที่อุปกรณ์อินพุท-เอาต์พุท หรือหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำไม่สามารถรับหรือส่งข้อมูลได้ทัน WAIT จะเป็นตัวทำให้ซีพียู ซิงค์ได้พอดีกับอุปกรณ์อินพุท-เอาต์พุทที่ทำงานด้วยความเร็วช้า

INT (INTERRUPT REQUEST) เป็นขาอินพุทแอคตีฟด้วยลอจิก"0" INT เป็นสัญญาณที่สร้างขึ้นมาจากอุปกรณ์อินพุท-เอาต์พุทเพื่อต้องการที่จะอินเตอร์รัพท์ สามารถควบคุมได้โดยซอฟแวร์ด้วยการเซทค่าอินเตอร์รัพท์ฟลิฟลอบ IFF การตอบอินเตอร์รัพท์มีแยกแยะให้ 3 แบบซึ่งจะอธิบายในรายละเอียดต่อไป

NMI (NONMASKABLE INTERRUPT) เป็นขาอินพุทที่จะบอกซีพียูในขณะของพัลซ์ขาลงการอินเตอร์รัพท์ด้วยวิธีนี้ซีพียูจะให้คำสั่งสูงกว่า INT กล่าวคือบัสมันจะตอบสนองและกระทำทันทีด้วยการเริ่มเอ็กซีคิ้ว คำสั่งในตำแหน่ง 0066 โดยอัตโนมัติ การกระโดดไปกระทำในกรณีนี้ซีพียูจะเก็บค่าโปรแกรมเคาว์เตอร์เดิมไว้ในสแตคเพื่อจะได้กลับไปทำงานเดิมเมื่อเสร็จสิ้นการอินเตอร์รัพท์ได้

RESET (RESET) เป็นขาอินพุทที่แอคตีฟด้วยลอจิก "0" การรีเซทในกรณีนี้จะมีผลดังนี้

1. ค่าของ PC มีค่าเป็น"0"
2. IFF จะได้รับการ DISABLE
3. รีจิสเตอร์ I จะมีค่า 00H
4. รีจิสเตอร์ R จะมีค่า 00H
5. จะมีการเซทอินเตอร์รัพท์โดดหมดมาอยู่ที่โหมดที่ 0

ระหว่างการรีเซทสายแอดเดรสบัสและข้อมูลจะได้รับการกระทำให้มีค่าอิมพีแดนซ์สูงเพื่อแยกออกจากซีพียู ส่วนสายสัญญาณควบคุมจะได้รับการทำให้เป็นสัญญาณที่ไม่แอคตีฟรีเฟรทช์จะไม่เกิดขึ้น

BUSRQ (BUS REQUEST) เป็นขาอินพุทที่แอคตีฟด้วยลอจิก "0" BUSRQ เป็นสัญญาณที่ส่งบอกกับซีพียูเพื่อต้องการให้ซีพียูควบคุมบัส กล่าวคือต้องการให้ซีพียูทำให้บัสแอดเดรส และข้อมูลอยู่ในสถานะอินพีแดนซ์สูงคือต้องการแยกซีพียูออกจากบัสนั่นเอง

BUSAK (BUS ACKNOWLEDGE) เป็นขาเอาต์พุทแอคตีฟขณะลอจิก "0" BUSAK เป็นสัญญาณตอบจากซีพียูได้แยกตัวเองออกจากแอดเดรสบัสและบัสข้อมูลเรียบร้อยแล้ว

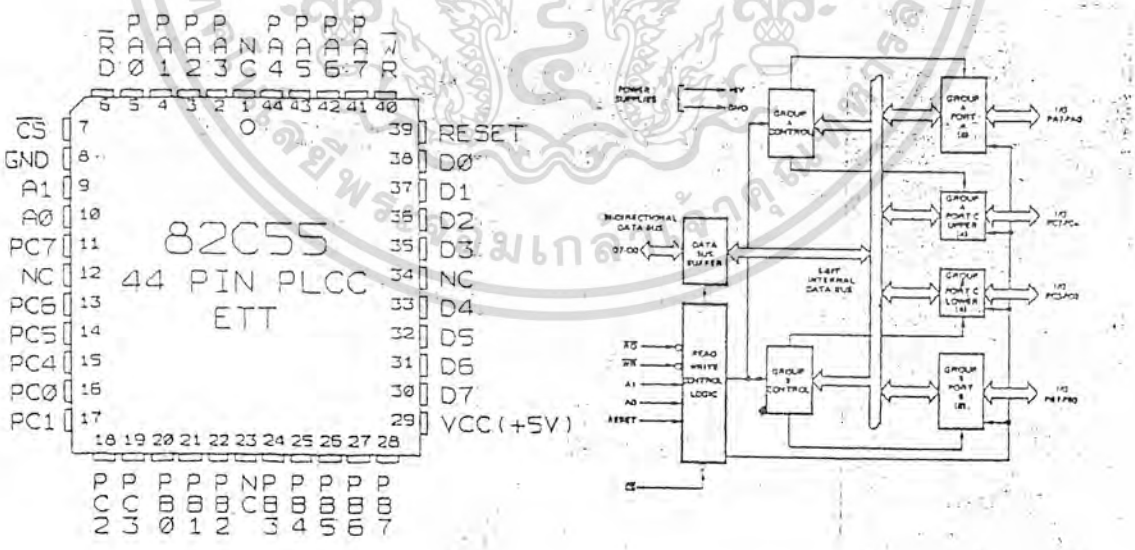
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82C55 I/O INTERFACE

เป็นส่วนที่เชื่อมต่ออุปกรณ์อินพุท. เอาต์พุทเข้ากับซีพียู นิยมใช้ไอซี 28C55 PPI (PROGRAMMABLEPERIPHERAL INTERFACE)ซึ่งเป็นไอซีที่บริษัทอินเทลผลิตขึ้นมาสำหรับสนับสนุนการใช้งานของซีพียู Z80 โดยเฉพาะ และซีพียูตัวอื่นมีลักษณะของสัญญาณการติดต่อควบคุมต่าง ๆ ใกล้เคียงกันได้ โดยมีการดัดแปลงแก้ไขนิดหน่อยเท่านั้นเอง

มีลักษณะพิเศษคือ เป็นชิพอินพุท เอาต์พุท ซึ่งสามารถโปรแกรมกำหนดลักษณะการทำงานโดย SOFTWARE 82C55 เป็นไอซีขนาด 44 ขา โดยจะมีลักษณะของพอร์ทแบบขนานถึง 3 พอร์ตละ 8 บิต ให้ใช้งานรวมเป็น 24 ขาด้วยกัน การติดต่อกับซีพียูทำให้ได้โดยต่อกับ DATA BUS 8 ขา ต่อกับสัญญาณควบคุมอีก 6 ขาและอีก 2 ขาเป็นแหล่งจ่ายไฟกับกราวน์

ซีพียูสามารถที่จะเลือก FUNCTION การทำงานได้โดยส่งข้อมูล 8 บิท ซึ่งเรียกว่าผ่าน DATA BUS มาให้กับ 82C55 ซึ่งจะกล่าวโดยละเอียดต่อไปนี้



รูปที่ 1.25 แสดงตำแหน่งขาและโครงสร้างภายในของ 82C55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งขาต่างและโครงสร้างภายในแสดงให้เห็นได้ดังรูปที่ 1.25 ซึ่งอธิบายหลักการทำงานได้คือ DATA BUS BUFFER มีโครงสร้างการทำงานเพื่อจะต่อกับ DATA BUS ของระบบซึ่งเราจะต่อโดบตรงกับ DATA BUS ของซีพียูดังนั้น BUFFER ตัวนี้จะเป็นตัวกักหรือส่งข้อมูลแบบ 2 ทิศทาง จึงใช้วงจรลอจิกสามสถานะ (TRI STATE) ข้อมูลที่ผ่าน DATA BUS BUFFER นี้จะกระทำได้โดยคำสั่ง IN หรือคำสั่ง OUT ซึ่งเป็นคำสั่งมาจากซีพียูและรวมถึงการเลือก MODE (CONTROL WORD และ STATUS)

READ UNIT CONTROL LOGIC ส่วนนี้จะเป็นส่วนควบคุมและจัดการเกี่ยวกับการเคลื่อนย้ายข้อมูลของ INTERNAL DATA BUS และ EXTERNAL DATA BUS ควบคุมทั้งสัญญาณที่เป็นข้อมูลและสัญญาณควบคุมเองด้วย ส่วนนี้จะรับสัญญาณ ADDRESS และสัญญาณควบคุมซีพียูอีกทีหนึ่งซึ่งตัวมันเองจะทำการถอดรหัสเพื่อควบคุมการทำงานของ GRUOP A CONTROL และ GROUP B CONTROL

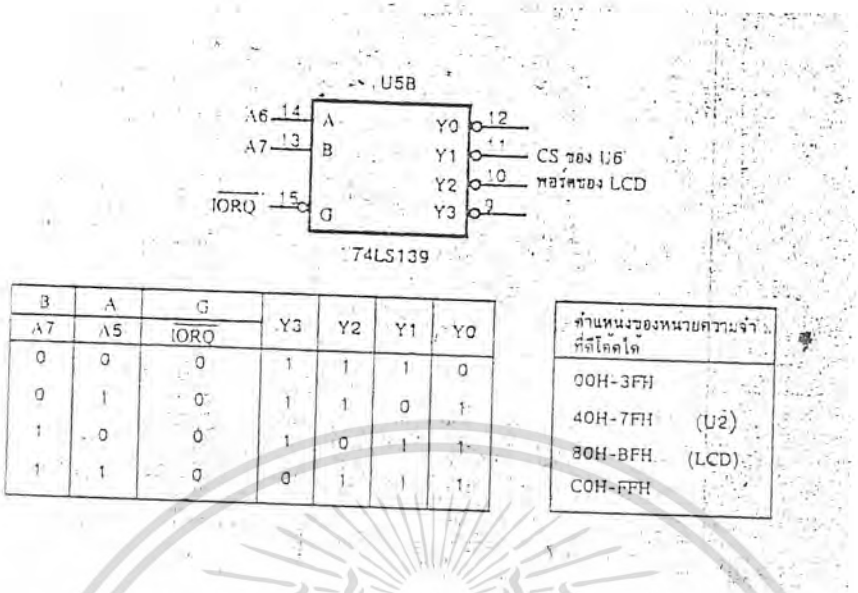
CS CHIP SELECT INPUT ที่ขานี้ถ้าได้รับลอจิก [LOGIC] "0" หมายความว่าซีพียู สามารถติดต่อ (อ่านหรือเขียน) กับ 82C55 ได้

RD READ INPUT ถ้าขานี้ได้รับลอจิก "0" แล้ว 82C55 จะทำการส่งข้อมูลจากพอร์ตหรือข้อมูลที่แสดงสถานะการทำงานไปยังซีพียูโดยผ่าน DATA BUS หรือกล่าวได้ว่าซีพียูจะทำการอ่านข้อมูลจาก 82C55 นั่นเอง

WR WRITE INPUT เมื่อ 82C55 รับสัญญาณเข้ามาที่ขานี้เป็นลอจิก "0" ซีพียูจะสามารถทำการส่ง (เขียน) ข้อมูลซึ่งอาจจเป็น DATA ไปที่พอร์ตหรืออาจเป็นข้อมูลที่เป็นคำสั่งไปยัง 82C55

AO-A1 (ADDESS INPUT) สัญญาณที่ป้อนเข้าที่ขานี้จะทำงานพร้อมกับ RD และ WR ทำให้สามารถเลือกพอร์ตได้ 3 พอร์ทโดยเหลืออีกหนึ่งพอร์ตจะเป็นพอร์ตที่รับคำสั่งงควบคุมที่เรียกว่า "CONTROLPORT REGISTER" โดยทั่วไปขา AO,A2 นี้จะต่ออยู่กับบิทล่างสุดของ ADDRESS BUS ซึ่งก็คือ AO,A1 ADDRESS BUS
นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 1.26 แสดงลักษณะการเลือกพอร์ต

RESET เมื่อใดที่ขาอินพุทนี้เป็น "0"ในทางลอจิก CONTROL WORD REGISTER จะถูกเคลียร์ทำให้ทุกพอร์ตถูกเซทให้เป็นอินพุท โหมด [INPUT MODE]

PA0-PA7 ขาทั้ง 8 ขานี้ถูกใช้เป็นอินพุท เอาต์พุทพอร์ตสำหรับพอร์ต A ซึ่งสามารถที่จะจะติดต่อกับอุปกรณ์ภายนอกได้

PB0-PC7 เหมือนกับ PA0-PA7 และ PB0-PB7 นอกจากนี้ยังสามารถแบ่งเป็นอินพุทพอร์ตเอาต์พุทพอร์ตขนาด 4 บิท ได้สองจุดซึ่งสามารถใช้งานได้อย่างอิสระ กล่าวคือแบ่งออกเป็น PORT C UPPER และ PORT C LOWER และยังสามารถใช้เป็นสายควบคุมของ PA0-PA7 และ PB0-PB7 ในการทำงานโหมดอื่นด้วย

GROUP A และ GROUP B CONTROL

การควบคุมการทำงานของ 82C55 จะถูกสั่งงานโดยการสั่งคำสั่งมาจากซีพียู โดยที่เราสามารถควบคุมได้สองกลุ่มคือ

GROUP A จะควบคุมได้คือ PORT A และ PORT C UPPER (PC7-PC4)

GROUPBจะควบคุม PORT B และ PORT C LOWER PC3-PC0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82C55 ประกอบด้วยพอร์ต พอร์ตซึ่งจะทำหน้าที่เป็นอินพุท เอาต์พุท

PORT A มี 8 บิททำหน้าที่เป็นเอาต์พุท แลทช์ OUTPUT LATCH และอินพุท แลทช์ INPUT LATCH

PORT B มี 8 บิทเป็นอินพุท หรือเอาต์พุท

PORT C อาจจะใช้งานทีละ 8 บิทหรือแบ่งเป็น 4 บิท UPPER และ 4 บิท LOWER นอกจากนี้ยังสามารถเอาไปควบคุม PORT A และ PORT Bได้อีกด้วยซึ่งจะเป็นการใช้งานในโหมด 1,2

นอกจากนี้ยังมีขาสัญญาณควบคุมอีกเส้นหรึ่งที่ต้องพิจารณาคือสัญญาณรีเซท ซึ่งสัญญาณนี้จะใช้รีเซทเครื่อง ในวงจรเครื่องนี้ได้สัญญาณมาจาก POWER SUPPLY ซึ่ง 82C55 จะ RESET ที่ลอจิก "1" แต่ Z-80 จะ RESET ที่โลจิก "0" ซึ่งจะเห็นว่าการทำงานไม่เหมือนกัน เพื่อที่จะให้วงจรสามารถรีเซทได้โดยใช้สัญญาณรีเซทจาก POWER SUPPLY อันเดียวกันคือ รีเซทที่โลจิก "0" ดังนั้นสัญญาณรีเซทจะป้อนให้กับ 82C55 ต้องผ่าน NOT GATE ก่อนเพื่อให้ 82C55 ได้รับสัญญาณรีเซทเป็นโลจิก "1" ในวงจรนี้ใช้ ไอซีเบอร์ 74LSO4 เข้าช่วย

จากนี้ได้กล่าวมาแล้วขา ขา AO,A1 นั้นต่อโดยตรงกับแอดเดรสบัสของ Z80 และเป็นตัวกำหนดให้ Z80 ติดต่อกับ REGISTER ภายในพอร์ตของ 82C55 เมื่อพิจารณาร่วมกับซึ่งได้มาจากการถอดรหัส (DECODER)โดยในวงจรเครื่องนี้ใช้ไอซีเบอร์ 74LS139 เป็นตัวถอดหรัส (DECODER) แอดเดรสที่นำมากำหนดเงื่อนไขการเลือกคือ A2-A3การเลือกเป็นตามรูปที่1.26

การกำหนดทำได้โดยการส่ง CONTROL WORD ขนาด 8 บิท มาให้ CONTROL QORD REGISTER จากสาเหตุนี้เองที่ทำให้เราสามารถโปรแกรมให้ 82C55 ทำงานในโหมดต่าง ๆ ได้ 3 โหมดที่แตกต่างกันโดยการกำหนด BIT DEFINITION ของ CONTROL WORD การโปรแกรมการทำงานของ 82C55 ทำได้ 3 โหมด คือ

โหมด 0 BASIC REEGISTER INPUT/OUTPUT

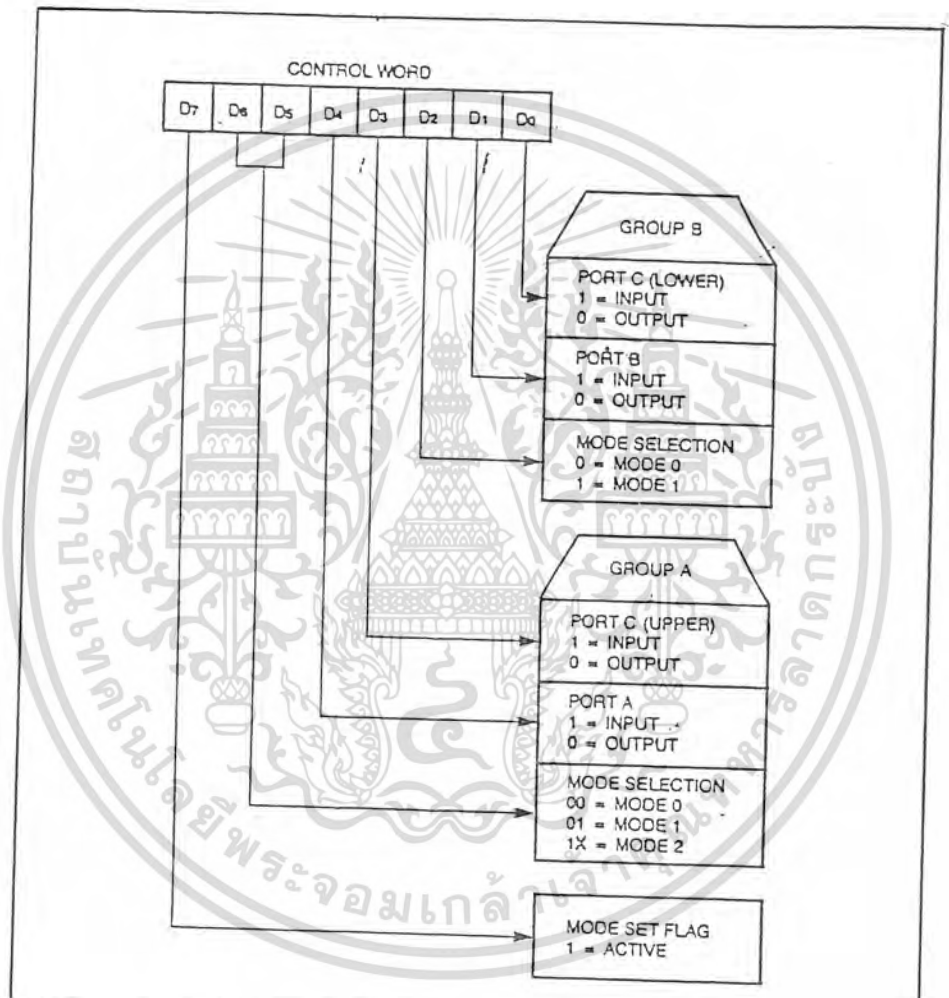
โหมด 1 STROBE INPUT/OUTPUT

โหมด 2 BIDIRECTIONAL I/O PORT

ในการทำงานของ I/O INTERFACE นี้ ต้องการให้ทำงานในโหมด 0 โดยกำหนดให้ PORT A,B เป็นเอาต์พุทพอร์ต,PORT CUPPER (PC7-PC4) เป็นอินพุทพอร์ตและให้ PORT C LOWER (PC3-PC0) เป็นอินพุทพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการให้แต่ละบิทของ CONTROL WORD เป็นไปตามเงื่อนไขของ IT DEFINETION ตามการกำหนดค่าแต่ละบิทจะได้ค่าของ CONTROL WORD เป็นตัวเลขออกมา



รูปที่ 7.5 : แสดงรายละเอียดแต่ละบิทของรีจิสเตอร์ควบคุมของ 8255.

รูป 1.27 แสดง BIT DEFINITION ของ CONTROL WORD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 1.27 การกำหนดให้ 82C55 ทำงานตามความต้องการในวงจรนี้แต่ละบิทจะมี

D7 กำหนดโหมดการทำงาน ถ้าเป็นลอจิก "1" คือ ACTIVE กำหนดว่าเป็นคำสั่งควบคุม (CONTROL WORD)

D6,D5 เป็นบิทกำหนดการเลือกโหมด ถ้าเป็น 00 จะทำให้ 82C55 ทำงานในโหมด 0

D4 เป็นลอจิก "0" กำหนดใน PORT A เป็นเอาต์พุทพอร์ต

D3 เป็นลอจิก "1" กำหนด PORT C UPPER PC7-PC3] เป็นอินพุทพอร์ต

D2 เป็นลอจิก "0" เลือกโหมดการทำงานเป็นโหมด 0

D1 เป็นลอจิก "0" กำหนดให้ PORT B เป็นเอาต์พุทพอร์ต

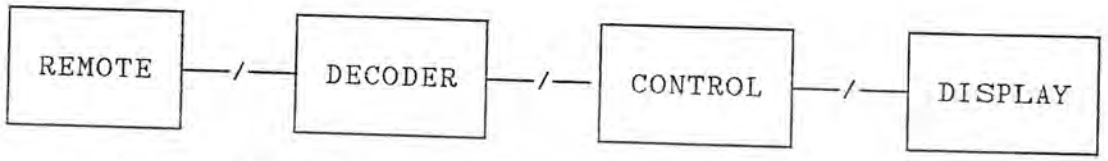
D0 เป็นลอจิก "1" กำหนดให้ PORT C LOWER (PC3-PC0) เป็นอินพุทพอร์ท

จากการกำหนดแต่ละบิทตามนี้จะได้ค่า CONTROL WORD เป็น 91H การเซ็ท 82C55 ทำได้โปรแกรดดังนี้

```
LD A,91H
OUT (43H),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
บล็อกไดอะแกรมภาพรวมของระบบ



ชุด REMOTE ประกอบด้วย

1. KEYBOARD DTMF
2. POWERSUPPLY +24 V

KEYBOARD DTMF

ใช้ KEYBOARD ระบบโทรศัพท์ เป็นระบบการส่งสัญญาณอีกแบบหนึ่งใช้ความถี่คู่ (DUAL TOME MULTIFREQUENCY TYPE) (DTMF)มีวิธีการส่งหมายเลขไปควบคุม โดยการส่งสัญญาณความถี่ 2 ความถี่มอดดูเลตกันไป ซึ่งจะเป็นตัวแทนของหมายเลขที่กด สำหรับแต่ละหมายเลขที่มีอยู่ 10 ตัว ความถี่ที่ส่งออกไปจะอยู่ในย่านความถี่ของเสียงพูด (0-4 กิโลเฮิรตซ์)

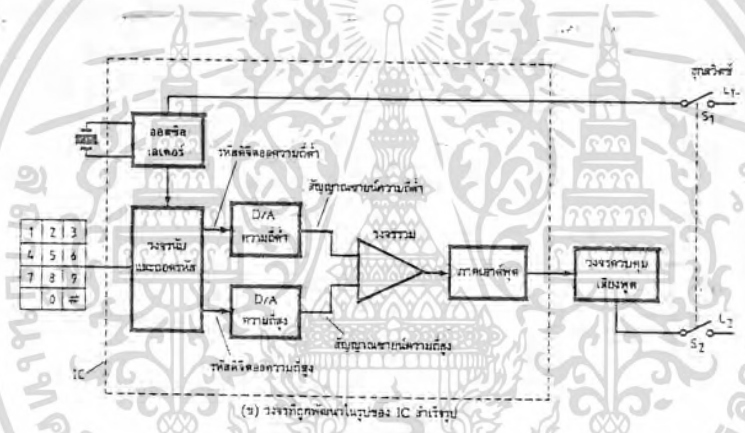
ตารางที่ 1 แสดงความถี่ที่มอดดูเลตกันเมื่อกดหมายเลข

ความถี่ (HZ	รหัสหรือหมายเลข		
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#
	1209	1336	1477 ความถี่ (H_z)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งค่าความถี่ที่ต่ำกว่าจะเป็นความถี่ที่แสดงในแนวนอน และอีกค่าหนึ่งก็จะเป็นความถี่ในแนวตั้ง เช่น เมื่อมีการกดหมายเลข 5 ก็จะมีความถี่ 770 เฮิรตซ์ และ 1336 เฮิรตซ์มอดดูเลตกันออกมา

การส่งรหัสหมายเลขโดยการใช้ไอซีสำเร็จรูป

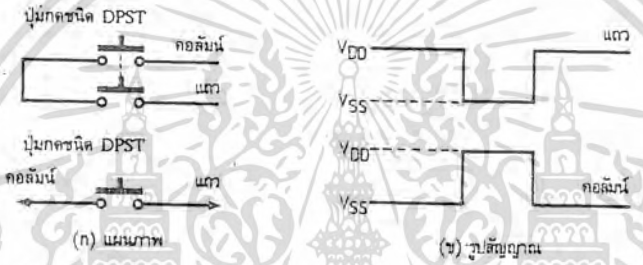


บล็อกไดอะแกรมของการส่งสัญญาณแบบ DTMF

เป็นบล็อกไดอะแกรมของไอซีที่นำมาใช้สร้างสัญญาณในระบบ DTMF ซึ่งวงจรนับและถอดรหัส (COUNTER AND DECODER) ซึ่งวงจรถอดรหัสก็จะแยกแยะว่าการกดหมายเลขแต่ละครั้งจะตรงกับตำแหน่งใดบ้างในแนวแถวและแนวคอลัมน์ เมื่อทำการถอดรหัสจากการกดได้แล้วก็นำค่าในแนวแถวและแนวคอลัมน์ไปหาจากค่าความถี่หลัก สัญญาณที่ออกจากวงจรนับและถอดรหัสได้สัญญาณดิจิตอล 2 สัญญาณที่มีความถี่แตกต่างกัน จากนั้นก็นำทั้ง 2 สัญญาณไปผ่านวงจรแปลงสัญญาณจากดิจิตอลไปเป็นอะนาลอก (D/A CONVERTER) และนำมารวมกันโดยการนำไปผ่านวงจรรวมและขยายสัญญาณ (SUMMING AMP) แล้วจึงถูกส่งผ่านไปยังสายสัญญาณที่ต่อกับชุดควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีอาจจะถูกออกแบบมาให้ใช้ร่วมกับแป้นปุ่มหมายเลข (KEY PAD) ชนิด DPST (DUAL-POLE SINGLE THROW) ซึ่งจะมีหน้าสัมผัส 2 หน้าหรืออาจจะเป็นชนิด SPST (SINGLE-POLE SINGLE THROW) ก็ได้ในรูปที่2 เป็นแผนภาพและรูปของสัญญาณเมื่อมีการกดปุ่มหมายเลขใด ๆ จะสังเกตว่าในการดีโค้ดของแนวแถวจะแอกทีฟที่ลอจิก "0" แต่ในแนวคอลัมน์นี้จะแอกทีฟที่ลอจิก 1



รูปแสดงชนิดของปุ่มกดและรูปสัญญาณ

ข้อควรคำนึงในการเชื่อมต่อระบบ DTMF กับสายส่งสัญญาณ

- ระดับแรงดันและกระแสลูปจะต้องรักษาให้คงที่ตลอดระยะทางของสายส่งสัญญาณ
- ความถี่ที่ถูกผลิตขึ้นจะต้องไม่มีความผิดเพี้ยนทั้งคาบและขนาดของสัญญาณ
- วงจรออสซิลเลเตอร์จะต้องมีอิมพีแดนซ์ที่สมดุลย์ (MATCHING) กับสายส่งสัญญาณ จากข้อสรุปข้างบน ถ้าจะแยกรายละเอียดของแต่ละหัวข้อก็จะมีพารามิเตอร์ที่ต้องคำนึงถึงดังต่อไปนี้

ความผิดเพี้ยนของสัญญาณ

สำหรับข้อกำหนดของความผิดเพี้ยนของสัญญาณมีดังต่อไปนี้

- สัญญาณอื่นที่แทรกสอดเข้ามาในสายส่งสัญญาณรวมกันแล้วจะต้องน้อยกว่าระดับของสัญญาณที่ถูกส่งออกไปจริงอย่างน้อย 20 DB
- สำหรับสัญญาณที่แทรกสอดเข้ามาจะต้องมีข้อกำหนดดังต่อไปนี้

1. ต้องมีค่าไม่มากกว่า -33 DBM ในช่วง 300 ถึง 3400 $H_{\overline{z}}$
2. ที่ความถี่สูงกว่า 3400 $H_{\overline{z}}$ สัญญาณแทรกสอดจะต้องลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12 DB ต่อออกเตฟไปจนถึงความถี่ 50 กิโลเฮิรตซ์

3. ต้องมีระดับสัญญาณไม่มากกว่า -80 DB ที่ความถี่ที่สูงกว่า 50 กิโลเฮิรตซ์ โดยที่ความผิดเพี้ยนถ้าถูกกำหนดในรูปของเดซิเบลคือ

$$\text{ความผิดเพี้ยน} = 20 \text{ LOG } \frac{V_1+V_2 \cdots V_n}{V_L+V_H}$$

โดยที่ V_1 ถึง V_n คือ ระดับแรงดันของสัญญาณที่แทรกสอดเข้ามา

V_L คือ ระดับแรงดันของสัญญาณความถี่ต่ำกว่า

V_H คือ ระดับแรงดันของสัญญาณความถี่ต่ำกว่า

<u>ไดนามิกอิมพีแดนซ์</u>

วงจรกำเนิดความถี่สำหรับระบบ DTMF จะต้องมีอิมพีแดนซ์อย่างน้อย 900 โอห์ม ขณะที่ทำการผลิตความถี่ออกมาและต้องมีอิมพีแดนซ์ให้ต่ำที่สุดขณะที่ไม่ทำการผลิตสัญญาณ

ความสูญเสียที่เกิดจากการสะท้อนกลับของสัญญาณ (RETURN LOSS)

ก็เป็นพารามิเตอร์อีกตัวที่จะต้องควบคุม โดยกำหนดค่าความสูญเสียในการสะท้อนกลับของสัญญาณหรือ RL ด้วยสมการ

$$RL = 20 \text{ LOG } \frac{Z_L + Z_G}{L_L - Z_G}$$

โดยที่ Z_L คือ อิมพีแดนซ์ของสายส่งสัญญาณ

Z_G คือ อิมพีแดนซ์ของเครื่องโทรศัพท์

ค่ามาตรฐานสำหรับ RL จะต้องมากกว่า 14 DB ในช่วงความถี่ระหว่าง 300 ถึง 3400 H_z และมากกว่า 10 DB ในช่วงความถี่ 50 ถึง 300 H_z และ 3400 H_z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีสำหรับการส่งสัญญาณแบบ DTMF

จากข้อมูลทั้งหมดสามารถสรุปถึงข้อดีของระบบ DTMF ได้คือ

- ลดระยะเวลาในการส่งหมายเลข
- สามารถใช้วงจรที่อุปกรณ์โซลิคสเตตได้ ซึ่งจะทำให้เกิดความประหยัดและสะดวก
- ลดอุปกรณ์จำพวกหน่วยความจำที่ใช้ภายในชุมสาย
- สามารถนำไปเชื่อมต่อกับอุปกรณ์ภายในวงจรควบคุมได้อย่างมีประสิทธิภาพ

ชุดแปลงสัญญาณทางโทรศัพท์แบบความถี่ (DECODER TONE)

ใช้ไอซีเบอร์ MT 8870 ในการแปลงความถี่ของการกดปุ่มแต่ละหมายเลข เพื่อแปลงให้เป็นเลขฐานสอง จำนวน 4 หลัก มีค่าตามหมายเลขที่กดซึ่งเป็นประโยชน์ในการบอกให้หน่วยประมวลผลกลาง (CPU) เพื่อนำไปประมวลผล

ในภาคนี้จะใช้อุปกรณ์ IC MT 8870 เป็นตัวรับและถอดรหัสความถี่ DTMF โดยภายใน IC นี้จะประกอบด้วยส่วนต่าง ๆ

- Filter Section
- Decode Section
- Stecring Circuit
- Amplifier
- Oscillator

คุณสมบัติของ MT8870

เป็นตัวรับและถอดรหัสความถี่ (DTMF receiver)

กินไฟน้อย ใช้ไฟระดับเดียวกับ TTL

สามารถตั้งอัตราขยายภายในตัวไอซีได้

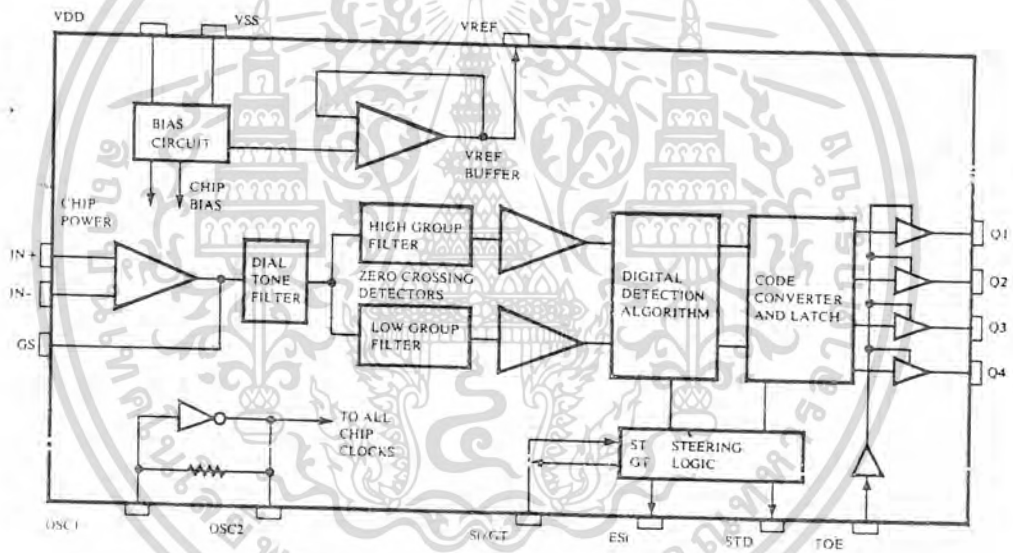
สามารถปรับการ์ดไทม์ (Guaard time) ได้

เป็นไอซีคุณภาพสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังก์ชันทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO^2 - CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิตช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูง และต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอล เพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิต และเช็คช่วงเวลาที่สัญญาณเข้ามา ส่วนภาคอินพุตเป็นออปแอมป์ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอกเอาต์พุตเป็นวงจรแลตช์ 3 สถานะ



รูปที่ แสดงโครงสร้างภายในของ MT8870

ฟังก์ชันการทำงานภายใน MT8870

ภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

ภาคกรองความถี่ (filter section)

ภาคถอดรหัส (decoder section)

ภาคตรวจสอบสสัญญาณ (steering circuit)

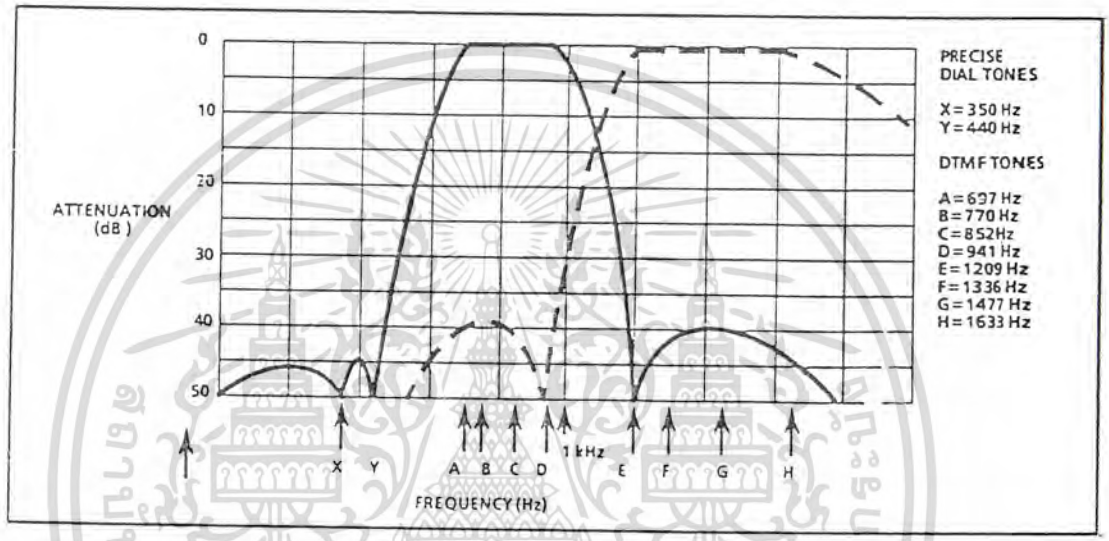
ภาคขยายสัญญาณความแตกต่าง (differential input)

ภาคกำเนิดความถี่ (oscillator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคกรองสัญญาณความถี่

ในส่วนนี้จะจะแยกสัญญาณ DTMF ที่เข้าออกเป็น 2 กลุ่มความถี่ คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิด สวิตช์คาปาซิเตอร์ (six-order switehed capacitor band pass filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วง คือช่วงความถี่สูงและช่วงความถี่ต่ำ



รูปแสดงความถี่ที่ได้จากภาคกรองความถี่

ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิตอล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นที่เข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา ESt (early steering) ก็จะแอคทีฟสำหรับค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ นั้น แสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F_{LOW}	F_{HIGH}	NO	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1339	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1339	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1339	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1209	0	H	1	0	1	0
941	1339	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

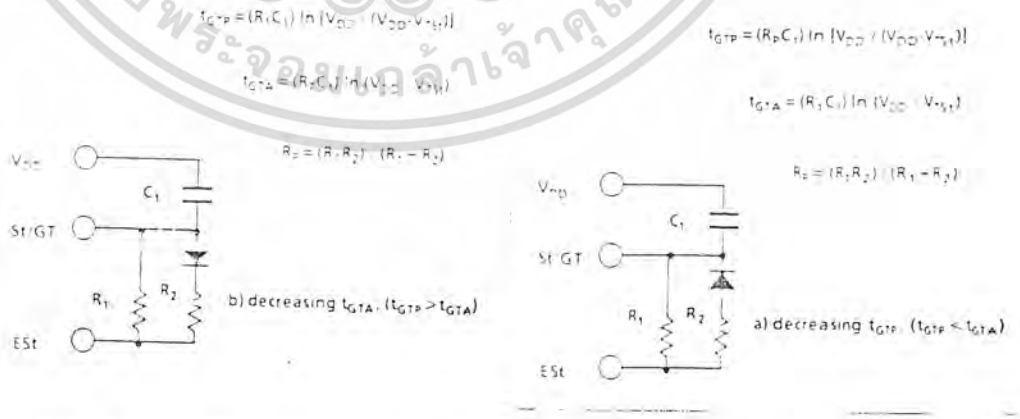
รูปแสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออการกดมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา ESt จะเป็น "High" นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามาจากรูป เมื่อขา ESt เป็น "High" ทำให้ V_c สูงขึ้น ตัวเก็บประจุ C จะคายประจุทำให้แรงดัน V_c สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัส จึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงานขอให้ดูจากแผนภูมิเวลาหรือไทมิ่งไดอะแกรม (timing diagram) ในรูปที่ 9 จะเข้าใจได้ง่ายกว่า

สำหรับคำว่าการ์ดไทม์ (gard time) นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่เราตั้งไว้ จึงจะได้รับการยอมรับว่าสัญญาณความถี่นั้นถูกต้อง หรือพูดได้ว่าเวลาที่เราตั้งไว้โดย RC ก็คือการ์ดไทม์นั้นเอง เมื่อสัญญาณความถี่เข้ามานานเท่าหรือมากกว่าเวลาที่ตั้งไว้จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสเป็นตัวเลขได้ ถ้าสัญญาณความถี่เข้ามาสั้นกว่า ก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลาและคำนวณเวลาดูได้จากรูป



รูปแสดงวงจรตรวจสอบอย่างง่ายและแสดงการกำหนดเวลาการ์ดไทม์ (GARD TIME) พร้อมวิธีการคำนวณ

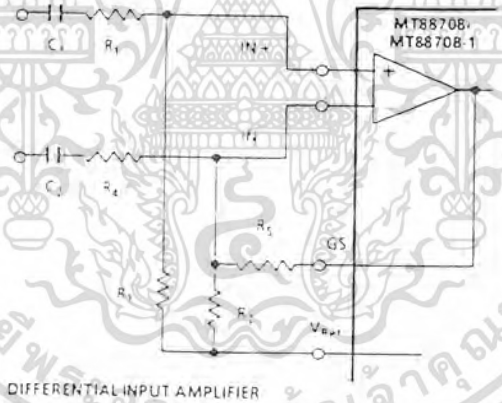
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุตของ MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเข้ากับอินพุตและอิมพีแดนซ์ได้ ดังนี้

$$\text{อัตราขยาย } (A_v\text{diff}) = \frac{R_5}{R_1}$$

$$\text{อินพุตอิมพีแดนซ์ } (Z_{in}\text{diff}) = 2\sqrt{R_1^2 + \left(\frac{1}{\omega C}\right)^2}$$



DIFFERENTIAL INPUT AMPLIFIER

$C_1 = C_2 = 10$ nF

$R_1 = R_3 = R_4 = 100$ kΩ All resistors are ± 1% tolerance

$R_2 = 60$ kΩ, $R_5 = 37.5$ kΩ All capacitors are ± 5% tolerance

$$R_5 = \frac{R_1 R_2}{R_1 - R_2}$$

$$\text{VOLTAGE GAIN } (A_v\text{ diff}) = \frac{R_5}{R_1}$$

INPUT IMPEDANCE

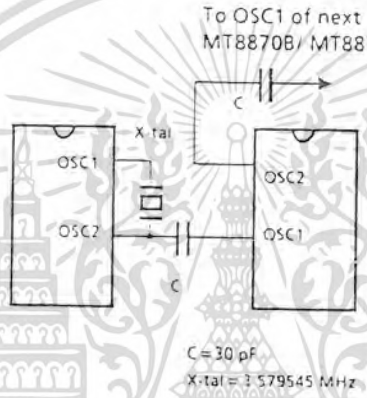
$$(Z_{INDIFF}) = 2\sqrt{R_1^2 - \left(\frac{1}{\omega C}\right)^2}$$

รูปแสดงการต่อวงจรภาคอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

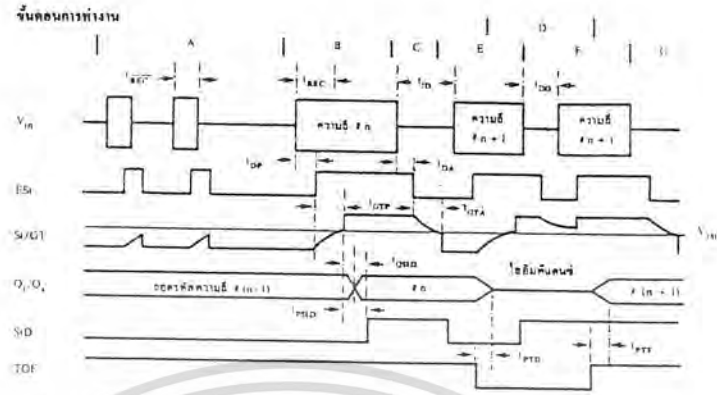
ภาคกำเนิดความถี่

ในภาคนี้ภายในไอซีจะมีวงจรเวลาอยู่ภายในเพียงแต่ต่อแร่คริสตอล ขนาด 3.58 MHz ก็สามารถใช้งานได้ทันทีการต่อวงจรกำเนิดความถี่แสดงในรูป



รูปแสดงการต่อวงจรกำเนิดความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



อธิบายขั้นตอนการทำงาน

A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาต์พุตไม่เปลี่ยน

B - ความถี่ # n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแลตช์ไว้ที่เอาต์พุต

C - จบความถี่ # n ช่วงห่างถูกต้อง เอาต์พุตยังคงแลตช์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D - เอาต์พุตเปลี่ยนเป็นไฮอิมพีแดนซ์

E - ความถี่ # n + 1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแลตช์ไว้

F - ความถี่ # n + 1 หายไป ช่วงห่างไม่ถูกต้อง เอาต์พุตยังคงแลตช์อยู่

C - จบความถี่ # n + 1 ช่วงห่างถูกต้อง เอาต์พุตยังแลตช์อยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

V_{in} - สัญญาณความถี่ DTMF ที่เข้ามา

ESt - Early Sterring output ใช้แสดงความถี่ที่ถูกต้อง

St/GT - Steering input/Guard Time output สำหรับต่อกับ RC ภายนอก

Q_1-Q_4 - เอาต์พุต BCD ขนาด 4 บิต

StD - Delayed Steering output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE - Tone Output Enable (input) ใช้ควบคุม Q_1-Q_4 ให้เป็นไฮอิมพีแดนซ์

$t_{\overline{REC}}$ - คาบเวลานานสุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง

t_{REC} - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

t_{ID} - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

t_{DO} - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{GTP} - การ์ดไทม์ของการปรากฏความถี่ DTMF

t_{GTA} - การ์ดไทม์ของการหายไปของความถี่ DTMF

รูปแสดง TIMEMING DIAGRAM การทำงานของ MT8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดควบคุมหรือ CONTROLER BOARD

SPECIFICATION

CPU	:	Z80B CMOS
MEMORY	:	ROM 2764,27256
	:	RAM 6264
PORT	:	82C55 (PLCC 44 PIN)
CLOCK RATE	:	4 MHz
POWER SSUPPLY	:	+ 5 VDC
CONNECTOR	:	40 PIN EXPANSION HEADER-STRIP
	:	34 PIN EXPANSION HEADER-STRIP
LED	:	POWER RED LED
	:	HALT GREEN LED

หลักการทำงาน

เป็นวงจรควบคุมการทำงานทั้งหมดของระบบทุกๆ ส่วนมีไมโครโปรเซสเซอร์ เป็นตัวควบคุม และกำหนดตามโปรแกรม โดยใช้ Z80B เป็น CPU ต่อร่วมกับ ROM เบอร์ 2764 เพื่อที่จะทำเป็นโปรแกรมมอนิเตอร์ และใช้ RAM เบอร์ 6264 เพื่อเป็นหน่วยความจำในการเขียนหรืออ่าน และเป็น สแต็ก

ในการอ้างแอดเดรสหน่วยความจำ ROM และ RAM จะใช้ไอซี 74LS139เป็นตัว DECODER หน่วยความจำโดยเราใช้แอดเดรส A14 และเป็น 4 ช่วงแอดเดรส ช่วงละ 16 K ไบต์ แต่ ROM เราต้องการ 32 K ไบต์ จึงต้องใช้ไดโอด D_3, D_4 และ R_{10} ต่อร่วมเป็น AND GATE ช่วงของ ROM คือ 0000H-7FFFH ส่วน RAM ใช้สัญญาณจากขา Y_2 ของ 74LS139 ได้โดยตรงทำให้ได้ช่วงแอดเดรสเป็น 8000H-BFFFH

ส่วนพอร์ตใช้ 82C55 เป็นพอร์ตอินพุต เอาต์พุตโดยใช้ 74LS139 เป็นตัว DECODER ตำแหน่งของพอร์ต ใช้แอดเดรส A6 และ A7 มาทำการ DECODE พร้อมกับสัญญาณ IORQ โดยจะ DECODE แบ่งออกเป็น 4 ช่วง แออดเดรสช่วงละ 64 ตำแหน่งแบ่งเป็นส่วนของ 82C55 ตั้งแต่ตำแหน่ง 40H-7FH ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT A = 40H

PORT B = 41H

PORT C = 42H

CONTROL PORT = 43H

PORT A เป็นเอาต์พุตเพื่อที่จะเลือกตำแหน่งของ DISPLAY โดยจะต่อร่วมกับ 74145 ซึ่งเป็น BCD to DECIMAL

PORT B เป็นเอาต์พุตเพื่อที่จะใช้เป็น DATA ของ DISPLAY โดยต่อเข้ากับ BUFFER

การส่งรหัสหมายเลขโดยใช้การส่งสัญญาณแบบ DTMF ซึ่งเป็นการส่งสัญญาณภานในสาย 2 เส้น ซึ่งสามารถที่จะส่งสัญญาณภายในสายได้ระยะไกลโดยใช้หลักการการนำความถี่ที่มีความแตกต่างกัน 2 ความถี่ ซึ่งเกิดจากการ DECODE ปุ่มหมายเลขให้เป็นสัญญาณความถี่แล้วนำมามอดูเลตกัน ซึ่งการ DECODE ความถี่และการมอดูเลตนี้จะใช้ร่วมกับแป้นปุ่มหมายเลข [KEY PAD] ชนิด DPST หรือชนิด SPST ไอซี UPC 757C นี้จะประกอบด้วยวงจรถอดรหัสลอจิกเกตวงจรหารความถี่หรือ CTRDIV K เมื่อมีการกดหมายเลขทำให้สัญญาณโรว์และคอลัมน์เกิดขึ้น และขาสัญญาณ SINGLE TONE ENABLE จะต้องถูกทำให้แอกทีฟสัญญาณดรว์ และคอลัมน์ และขา 18,19,20,21 เป็นโรว์ เมื่อมีการกด KEY สัญญาณ DTMF ที่ออกเป็น 2 ความถี่ ซึ่งสัญญาณจากสายจะส่งเข้า MT 8870 เป็น IC DECODE สัญญาณ DTMF ให้เป็นตัวเลข BCD ขนาด 4 บิต เพื่อที่จะเข้าไปเป็นอิตพุตของ82C55เพื่อให้CPU ประมวลผลซึ่งสายที่ใช้ส่งสัญญาณนี้จะต้องมีไฟ DC ไปควบคู่กับสัญญาณเพื่อเป็นไฟไปเลี้ยง BOARD ต่าง ๆ ด้วย POWER SUPPLY ที่ใช้เท่ากับ +24VDC โดยใช้ AC24V ผ่านวงจร RECTIFIER และ FILTER ทำให้ได้ 24VDC จากนั้นเข้าวงจร REGULATOR เพื่อให้ได้ 24VDC ไปเลี้ยง CONTROLLER BOARD,DTME KEYBOARD,DTME DECONDER และ DISPLAYBOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุด DISPLAY

ในปัจจุบันนี้ป้ายโฆษณาที่มีลักษณะเป็นตัวอักษรเลื่อนกำลังเป็นที่นิยม ดังเห็นได้จากตามห้างสรรพสินค้าทั่วไป มักจะมีป้ายโฆษณาในลักษณะนี้ติดตั้งอยู่มากมาย สามารถดึงดูดความสนใจได้มากกว่าป้ายโฆษณาทั่วๆ ไป เนื่องจากมองเห็นได้เด่นชัด มีรูปแบบในการแสดงผลที่น่าสนใจ จึงมีผู้นิยมใช้กันมากขึ้นเรื่อยๆ ในการที่จะมาศึกษาถึงการทำป้ายโฆษณาในลักษณะนี้ก็ไม่ใช่เรื่องยากนัก โดยเริ่มจากป้ายเล็กๆ ก่อนแต่ก็สามารถนำไปใช้งานได้ และเมื่อเข้าใจหลักการต่างๆ แล้วก็สามารถทำป้ายที่มีขนาดใหญ่กว่านี้และมีลูกเล่นมากกว่านี้ได้

ในการทำไฟวิ่งมีหลักการอยู่การสแกน ซึ่งการสแกนนี้สามารถทำได้ 2 วิธีคือ สแกนทางคอลัมน์กับสแกนทางโรว์

การสแกนทางคอลัมน์จะทำการส่งข้อมูลออกไปทางโรว์ โดยส่งข้อมูลตัวที่ 1 ออกไป แล้วให้คอลัมน์ที่ 1 แอกตีฟ จากนั้นก็ทำการส่งข้อมูลตัวที่ 2 ออกไป แล้วให้คอลัมน์ที่ 2 แอกตีฟ ทำเช่นนี้ไปจนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกคอลัมน์ก็จะเป็นการสแกนครบ 1 รอบดังนั้นถ้าจำนวนหลักที่จะแสดงผลออกมาเป็นตัวอักษรมีจำนวนหลายหลัก วิธีนี้จะไม่เหมาะสมนักที่จะนำมาใช้งานเพราะว่าเมื่อให้ LED ที่คอลัมน์สุดท้ายจะติดจะต้องใช้เวลานาน

ส่วนการสแกนทางโรว์จะทำการส่งข้อมูลออกไปทางคอลัมน์ โดยจะส่งข้อมูลออกไปจนครบหมดทุกหลักก่อนแล้วให้โรว์ที่ 1 แอกตีฟ จากนั้นก็ทำการส่งข้อมูลชุดถัดไปออกไปจนครบหมดทุกหลัก แล้วให้โรว์ที่ 2 แอกตีฟทำเช่นนี้จนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกโรว์ ก็จะเป็นการสแกนครบ 1 รอบ วิธีนี้มีข้อดีคือสามารถแสดงผลเป็นตัวอักษรพร้อมกันได้หลายหลักและถ้าจัดเวลาให้เหมาะสมแล้ว เวลาทำการสแกนจะไม่ทำให้เกิดอาการพริ้ว แต่มีข้อเสียคือการเขียนโปรแกรมควบคุมให้ตัวอักษรเลื่อนทำได้ยากกว่าแบบแรก ในโครงงานนี้จะใช้วิธีการสแกนทางโรว์เพื่อควบคุมการเลื่อนของตัวอักษร

การทำงานทางฮาร์ดแวร์

วงจรของแผงอักษรไฟวิ่งแสดงดังรูปที่ 1

เริ่มที่ IC_{19} 74LS145 จะทำหน้าที่เลือกตำแหน่งหลักในการส่งข้อมูลออกไปโดยใช้ 4 บิต ล่างของพอร์ต B คือ PB0,PB1,PB2,PB3 เป็นตัวควบคุมเอ๊าต์พุตของ IC_{19} จะต่อเข้ากับขา CLK ของ IC_{10}-IC_{14} เบอร์ 74LS374 IC_{10}-IC_{14} จะทำหน้าที่แลตช์ข้อมูลค้างเอาไว้ โดยที่ข้อมูลจะถูกส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากพอร์ต A ในการส่งข้อมูลนั้นจะต้องทำการเลือกหลักที่จะส่งข้อมูลนั้นก่อน โดยให้ขา CLK ของ IC_{10}-IC_{14} เป็น "0" แล้วจึงทำการส่งข้อมูลออกไป จากนั้นจะต้องให้ขา CLK ของ IC_{10}-IC_{14} เป็น "1" ข้อมูลจะถูกแลตช์ค้างไว้ (ข้อมูลจะถูกแลตช์ไว้ที่ขอบขาขึ้น)

เอาต์พุตของ IC_{10}-IC_{14} จะต่อกับอินพุตของ IC_4-IC_9 ถ้าอินพุตของ IC_4-IC9 เป็น "1" จะได้เอาต์พุตเป็น "0" ใช้ในการขับ LED ด้วย เมื่อข้อมูลถูกส่งออกมาจากพอร์ต A ทั้ง 8 บิต แล้ว ทรานซิสเตอร์ Q_1-Q_8 ที่ต่อไว้ขับ LED ติดตามข้อมูลที่ส่งออกมา

ส่วน IC_1 เบอร์ 74HCT138 จะทำหน้าที่สแกนทางโรว์ โดยใช้ 4 บิตบนของพอร์ตเป็นตัวควบคุมการสแกน โดยแบ่งเป็น PB4,PB5,PB6 ทำหน้าที่สแกนส่วน PB7 จะทำหน้าที่เปิด/ปิดการสแกนถ้า PB7 เป็น "1" ก็จะเป็นการปิดการสแกนนั่นคือ เอาต์พุตของ IC_1 จะเป็น "1" หมด ก็จะทำให้ Q_1-Q_8 อยู่ในสภาวะคัตออฟ เอาต์พุตของ IC_1 จะติดกับ IC_2 เบอร์ 74LS07 ซึ่งจะทำหน้าที่เป็นบัฟเฟอร์และขับทรานซิสเตอร์ โดยมีเอาต์พุตเป็นแบบคอลเล็กเตอร์จึงต้องต่อต้านทาน R_{50} และ $R_{50/1}$ พูลอัพไว้

IC_{17} และ IC_{18} เบอร์ 74HCT123 จะต่อเป็นวงจรโมโนสเตเบิล ถ้าอินพุตของ IC_{17} และIC_{18} ถูกทริกภายในช่วงเวลาที่ถูกกำหนดโดยค่า RC เอาต์พุตของไอซีทั้งสองก็จะเป็น "0" ถ้าไม่เช่นนั้นเอาต์พุตเป็น "1" IC 18 จะทำหน้าที่ป้องกัน LED ติดค้างหรือนานเกินไป เพราะจะทำให้หลอดขาด เนื่องจากกระแสไหลมากเกิน ทั้งนี้เพราะว่าตัวต้านทานที่ต่อจำกัดกระแสให้ LED มีขนาดเพียง 10 โอห์มเท่านั้น หากใช้ค่าตัวต้านทานสูงกว่านี้ LED ก็จะไม่สว่างเท่าที่ควร

อินพุตของ IC_{18} จะต่ออยู่กับเขา A ของ IC_1 ส่วนเอาต์พุต Q ก็จะต่อยู่กับขา IC_1 ทำให้เกิดการสแกน ซึ่งช่วยให้ LED ติดสว่างไม่นานเกินไป

IC_{17}จะต่ออยู่กับ $IC_{15/2}$ เบอร์ 556 ทำหน้าที่เป็นวงจรวอตช์ด็อก ถ้าซีพียูเกิดอาการแฮงก์หรือติดอยู่ในลูปใดลูปหนึ่งของโปรแกรม แล้วส่งสัญญาณทริกออกมาทาง PC7 ไม่ทัน ตามเวลาที่กำหนดโดยค่า RC ที่ต่ออยู่ในวงจร (ประมาณ 3 วินาที) วงจรในส่วนนี้ก็จะส่งสัญญาณมารีเซตซีพียูให้เริ่มต้นทำงานใหม่ สัญญาณทริกที่ส่งออกมาทาง PC7 จะต้องส่งออกมาในลักษณะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัลส์ ส่วน $IC_{15/1}$ ต่อเป็นวงจรอะสเตเบิลเพื่อกำเนิดสัญญาณอินเตอร์รัพต์ที่สามารถปรับช่วงเวลาของการอินเตอร์รัพต์ได้ โดยการปรับค่า VR_1 ช่วงเวลาที่สามารถปรับได้อยู่ระหว่าง 1-1.75 มิลลิวินาที

การสแกน

ก่อนอื่นต้องทำการปิดการสแกนก่อน ซึ่งก็คือ การทำให้ทรานซิสเตอร์ทั้ง 8 ตัว อยู่ในสภาวะคัตออฟทำได้โดยการเซตให้ PB7 เป็น "1" IC_1 จะถูกดิสเอเบิลเพื่อไม่ให้ข้อมูลที่ไม่ต้องการหลุดออกไป ในการสแกนนี้จะทำการเลือกหลักที่จะส่งข้อมูลออกไปด้วย เพราะพอร์ต B ถูกแบ่งออกเป็น 2 ส่วนคือ 4 บิตบนทำหน้าที่สแกนทางด้านโรว์ และควบคุมว่าจะเปิดหรือปิดการสแกน

โดยปกติจะให้หลักแรกทำงานก่อนพร้อมทั้งปิดการสแกนด้วย หลังจากนั้นก็นำข้อมูลที่อยู่ในดิสเพลย์บัฟเฟอร์ของแถวที่ 1 หลักที่ 1 ออกไปที่พอร์ต A ข้อมูลที่ถูกส่งออกไปจะยังไม่ถูกแลตช์ไว้ จนกว่าจะให้ขา CLK ของ IC_{10} เป็น "1" ข้อมูลจึงจะถูกแลตช์ข้อมูลไว้แล้ว ก็ให้หลักที่ 2 ทำงาน แล้วจึงนำข้อมูลในดิสเพลย์บัฟเฟอร์ของแถวที่ 1 หลักที่ 2 ส่งออกไปที่พอร์ต A

ทำการส่งข้อมูลในลักษณะนี้จนครบ 5 หลัก จากนั้นจึงเลือกให้แถวที่ 1 ทำงาน พร้อมกับเปิดการสแกนด้วย ขั้นตอนถัดมาก็ทำการหน่วงเวลาเพื่อให้ LED ติดชั่วขณะหนึ่งตามข้อมูลที่ได้ส่งออกไปทั้ง 5 หลัก ในการเปิดการสแกนก็เพียงแต่ให้ PB7 เป็น "0" จากนั้นจะต้องทำการเคลียร์ข้อมูลเก่าเสียก่อน โดยการส่งข้อมูลเป็น 00H ออกไปทั้ง 5 หลักตามวิธีที่กล่าวมาแล้ว ทั้งนี้เพื่อป้องกันความผิดพลาดที่อาจจะเกิดขึ้นของข้อมูลที่ไม่ต้องการเกิด จากนั้นจึงทำการส่งข้อมูลชุดใหม่ที่อยู่ในดิสเพลย์บัฟเฟอร์ที่ต้องการให้ติดในแถวที่ 2 ออกไปทั้ง 5 หลัก และให้แถวที่ 2 ทำงาน

ตามวิธีที่กล่าวมาแล้วทำจนครบทั้ง 8 แถวก็จะมองเป็น LED ติดเป็นตัวอักษร แต่จะเห็นเพียงชั่วขณะเท่านั้น จึงจำเป็นต้องส่งข้อมูลชุดเดิมทั้งชุดออกไปอีก จึงจะมองเห็น LED ติดเป็นตัวอักษร

ลักษณะการสแกนและเลื่อนข้อมูลแสดงดังในรูปที่ 2

การจัดหน่วยความจำ

ในการใช้งาน ET-LDB ต้องมีการเขียนโปรแกรมควบคุมและกำหนดข้อมูลที่ต้องการแสดง ดังนั้นจึงต้องมีการจัดหน่วยความจำที่เรียกว่า ดิสเพลย์บัฟเฟอร์มีตำแหน่งตั้งแต่ 3000H-302FH แบ่งออกเป็น 6 ส่วน ดังรูปที่ 3 นั่นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ตำแหน่ง 3000H-302FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 1 ที่ตำแหน่ง 0008H-300FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 2

ที่ตำแหน่ง 3010H-3017H เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 3

ที่ตำแหน่ง 3018H-301FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 4

ที่ตำแหน่ง 3020H-3027FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 5

ที่ตำแหน่ง 3028H-3022FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 6

ในแต่ละหลักจะมีข้อมูลอยู่ 8 ไบต์ ซึ่งจะเป็นข้อมูลของตัวอักษร 1 ตัวลักษณะของข้อมูลของตัวอักษร 1 ตัว ดังแสดงในรูที่ 4 ซึ่งเป็นตัวอักษรตัว A

ข้อมูลที่อยู่ดิสเพลย์บัฟเฟอร์ของหลักที่ 1 ถึงหลักที่ 5 เป็นข้อมูลที่นำออกมาแสดงผลทาง LED ส่วนข้อมูลที่อยู่ในดิสเพลย์บัฟเฟอร์ของหลักที่ 6 จะทำหน้าที่เก็บข้อมูลของตัวอักษรถัดไปที่จะนำมาแสดงผล

ในตอนแรกข้อมูลของหลักที่ 1 ถึง 5 จะมีค่าทั้งหมดเป็น 00H ส่วนข้อมูลของตัวอักษรตัวแรกที่จะนำออกมาแสดงผลจะถูก เก็บไว้ในหลักที่ 6 เมื่อผ่านขั้นตอนการสแกนและเลื่อนข้อมูลจนข้อมูลถูกเลื่อนไปครบ 8 บิต(ครบ 1 หลัก) ตัวอักษรตัวแรกก็จะมาอยู่ที่หลักที่ 5 เราก็จะทำการส่งข้อมูลของตัวอักษรตัวที่ 2 มาที่หลักที่ 6 และเมื่อผ่านขั้นตอนการสแกนและเลื่อนข้อมูลจนข้อมูลถูกเลื่อนไปอีก 8 บิต ตัวอักษรตัวแรกก็จะมาอยู่ที่หลักที่ 4 ตัวอักษรตัวที่ 2 ก็จะย้ายมาอยู่ที่หลักที่ 5 จากนั้นก็ทำการส่งข้อมูลของตัวอักษรตัวที่ 3 มาที่หลักที่ 6

ดังนั้นดิสเพลย์บัฟเฟอร์ในหลักที่ 6 จะทำหน้าที่รับข้อมูลของตัวอักษรตัวใหม่ที่จะนำมาแสดงผลเก็บไว้ และจะใช้เป็นส่วนหนึ่งในกระบวนการเลื่อนข้อมูลอีกด้วย

การเลื่อนข้อมูล

ในการที่จะทำให้เห็นเป็นอักษรวิ่งนั้นในที่นี้เราใช้วิธีการเลื่อนข้อมูล โดยที่ข้อมูลในแต่ละหลักจะมีขนาด 8 บิต เมื่อทำการเลื่อนข้อมูลไปทางซ้าย 1 ครั้ง ข้อมูลที่อยู่ในแต่ละหลักจะถูกเลื่อนไปทางซ้าย 1 บิต และข้อมูลในบิตที่ 7 จะถูกเลื่อนไปยังหลักถัดไป นั่นคือข้อมูลที่อยู่บิต 7 ของหลักถัดไป นั่นคือข้อมูลที่อยู่ที่บิต 7 ของหลักที่ 2,3,4,5 และ 6 จะถูกเลื่อนมาอยู่ที่บิต 0 ของหลักที่ 1.2.3,4 และ 5

เมื่อเราทำการสแกนข้อมูลที่ถูกเลื่อนข้อมูลไปแล้ว ก็จะมองเห็นตัวอักษรถูกเลื่อนมาทางซ้าย 1 ตำแหน่ง และเมื่อทำการเลื่อนข้อมูลและทำการสแกนอีก ก็จะมองเห็นตัวอักษรถูกเลื่อนไปอีก 1 ตำแหน่ง ดังนั้นเมื่อทำเช่นนี้ไปเรื่อยๆ ก็จะสามารถมองเห็นเป็นตัวอักษรวิ่งไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเลื่อนข้อมูลสามารถอธิบายได้ดังนี้

สมมติว่ามีข้อมูลที่จะนำออกมาแสดงผลทาง LED 1 หลัก ก็จะต้องมีข้อมูลตัวถัดไปที่จะนำออกมาแสดงผลอยู่ในหลักที่ 2 เพื่อที่จะทำการเลื่อนข้อมูลในหลักที่ 2 มายังหลักที่ 1 ซึ่งจะทำให้เห็นเป็นตัวอักษรเลื่อนไปได้ สมมติว่าข้อมูลทั้ง 2 หลัก มีอยู่หลักละ 1 ไบต์ แล้วต้องการที่จะทำการเลื่อนข้อมูลจากหลักที่ 2 มายังหลักที่ 1 สามารถอธิบายได้ดังนี้

ขั้นแรกจะต้องทำการหมุนข้อมูลของทั้ง 2 หลัก โดยใช้คำสั่ง RLCA ก่อน (คำสั่งใน Z80) จากนั้นก็ให้ข้อมูลในหลักที่ 1 บิต 0 มีค่าเป็น "1" และให้ข้อมูลในหลักที่ 2 บิต 1 ถึงบิต 7 มีค่าเป็น "0" จากนั้นก็นำข้อมูลของทั้ง 2 หลัก มาบวกกันก็จะได้ข้อมูลตัวใหม่ของหลักที่ 1 ส่วนข้อมูลในหลักที่ 2 จะไม่นำมาแสดงผล เมื่อทำการสแกนข้อมูลของหลักที่ 1 ก็จะมองเห็นว่าข้อมูลทั้งหมดของหลักที่ 1 ถูกเลื่อนไปทางซ้าย 1 ตำแหน่ง และถ้าต้องการเลื่อนข้อมูลไปทางซ้ายอีก 1 ตำแหน่ง ก็สามารถทำได้โดยวิธีที่กล่าวมาแล้ว โดยข้อมูลที่จะนำมาใช้คือ ข้อมูลในหลักที่ 1 ที่ผ่านการเลื่อนข้อมูลแล้วในครั้งแรก ส่วยข้อมูลในหลักที่ 2 ก็คือ ข้อมูลก่อนที่จะถูกทำให้บิต 1 ถึงบิต 7 มีค่าเป็น "0" และเมื่อทำการเลื่อนข้อมูลจนครบ 8 ไบต์ในหนึ่งหลัก ดังนั้นเวลาทำการเลื่อนข้อมูลก็จะต้องทำทั้ง 8 ไบต์ด้วย

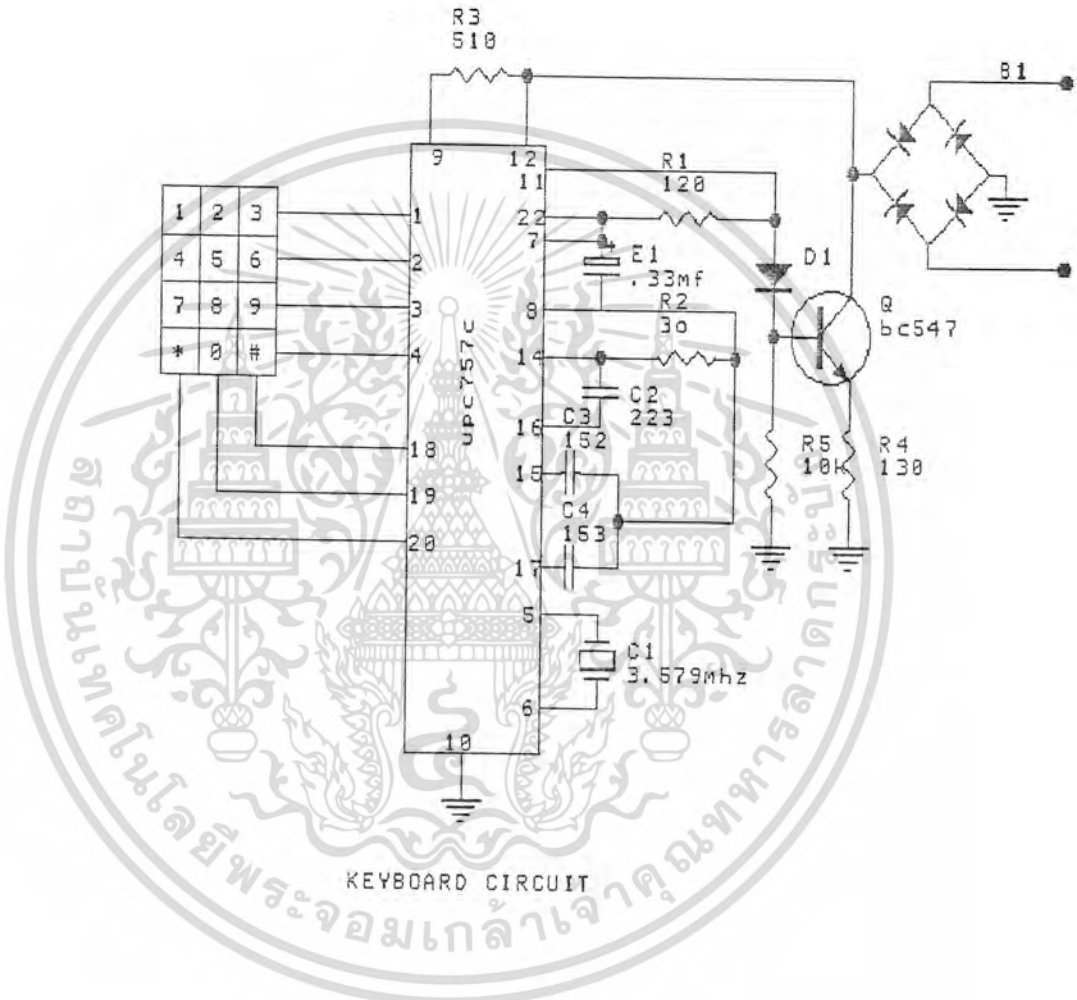
ถ้าต้องการนำข้อมูลออกมาแสดงผลทั้งหมด 5 หลัก ก็จะต้องมีข้อมูลทั้งหมด 6 หลัก โดยหลักที่ 6 จะทำหน้าที่ดังที่กล่าวมาแล้วในตอนแรก ในรูปที่ 5 จะแสดงถึงวิธีการเลื่อนข้อมูลที่ได้กล่าวมาแล้ผู้ใช้สามารถใช้วิธีอื่นๆ ในการเลื่อนข้อมูลได้ โดยใช้คำสั่งที่มีอยู่ในซีพียูเบอร์นั้นๆ หรือจะใช้เทคนิคอื่นๆ ก็ได้

การสร้าง

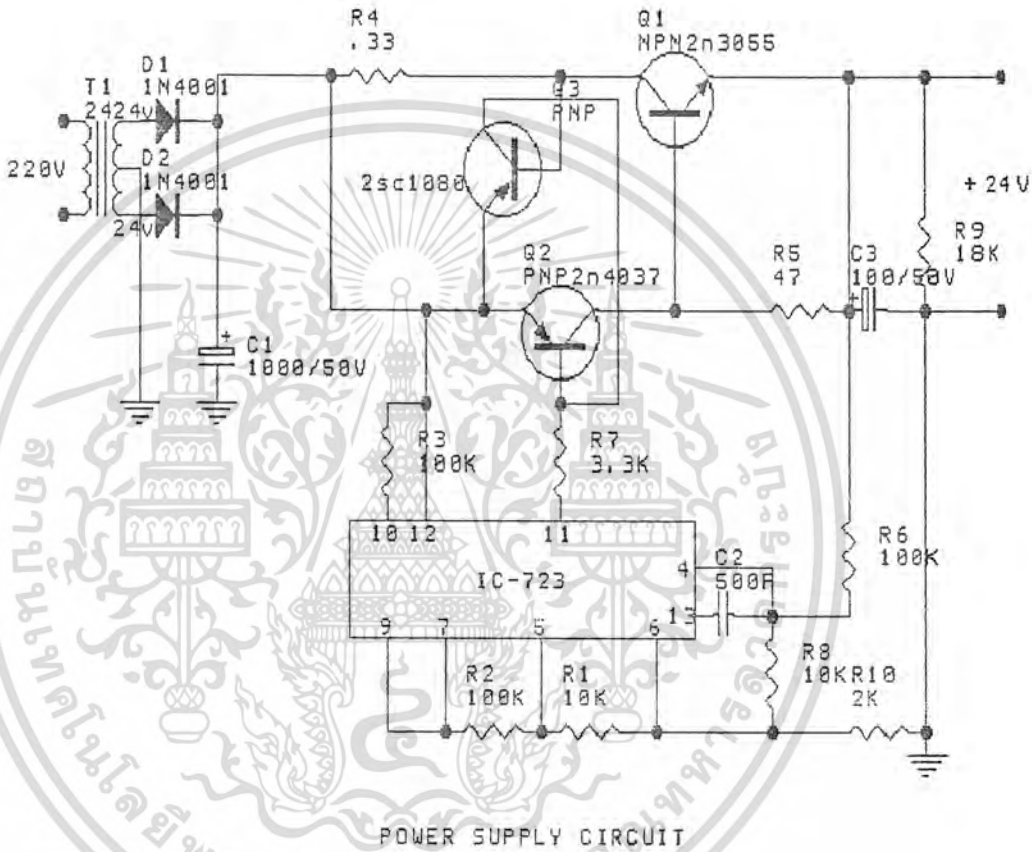
ก็ต้องเริ่มที่แผ่นวงจรพิมพ์ ซึ่งต้องเป็นชนิดสองหน้าเพลตทรูโฮลด์ รูปที่ 6 เป็นลายทองแดงของแผ่นวงจรพิมพ์ทั้ง 2 ด้าน ซึ่งมีขนาด 50% ของขนาดจริง รูปที่ 7 เป็นการวางอุปกรณ์ .

การติดตั้งLED จะต้องค่อนข้างพิถีพิถันอย่างมากคือ ต้องทำให้ตรงกันมีความสูงเท่ากัน ดังนั้นจึงติด LED ให้ชิด กับแผ่นวงจรพิมพ์เลย ดังในรูปที่ 8 จะเห็นได้ว่า LED ทั้ง 320 ดวง จะติดตั้งบนแผ่นวงจรพิมพ์ครบระดับกับอุปกรณ์อื่นๆ เมื่อประกอบเรียบร้อยจะมีหน้าตาดังรูปที่ 9 และ 10 โดยในรูปที่ 9 แสดงแผ่นกรองแสงที่ได้ปิดแผงอักษรไฟวิ่งด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

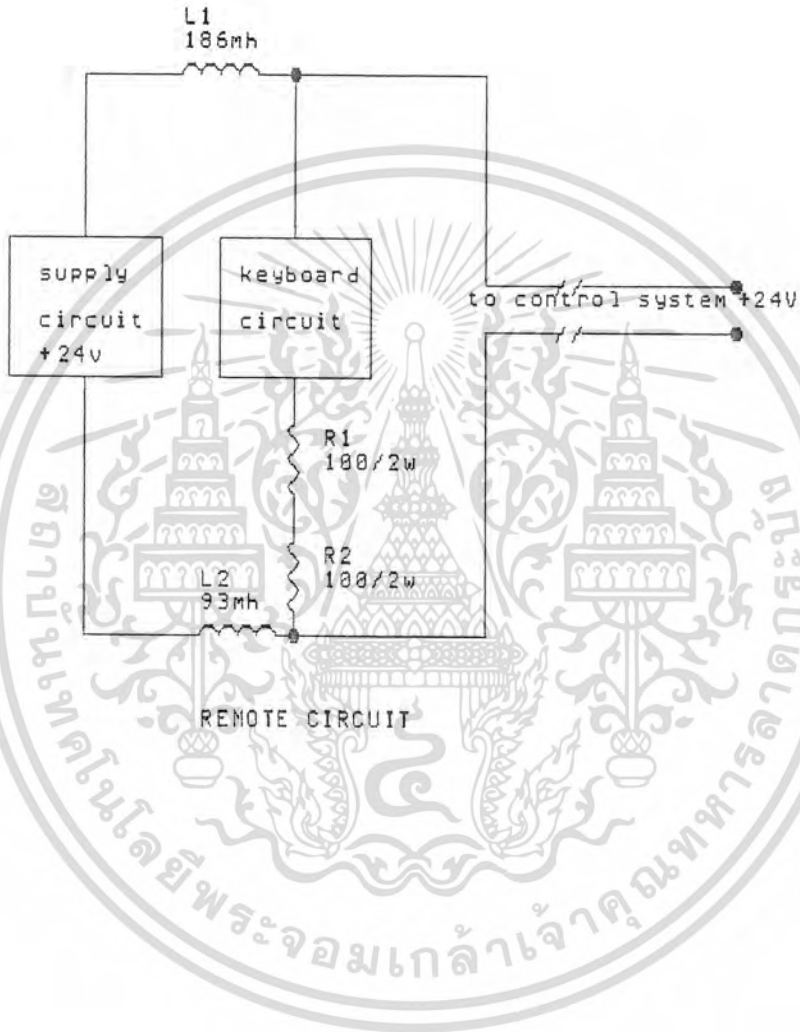


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

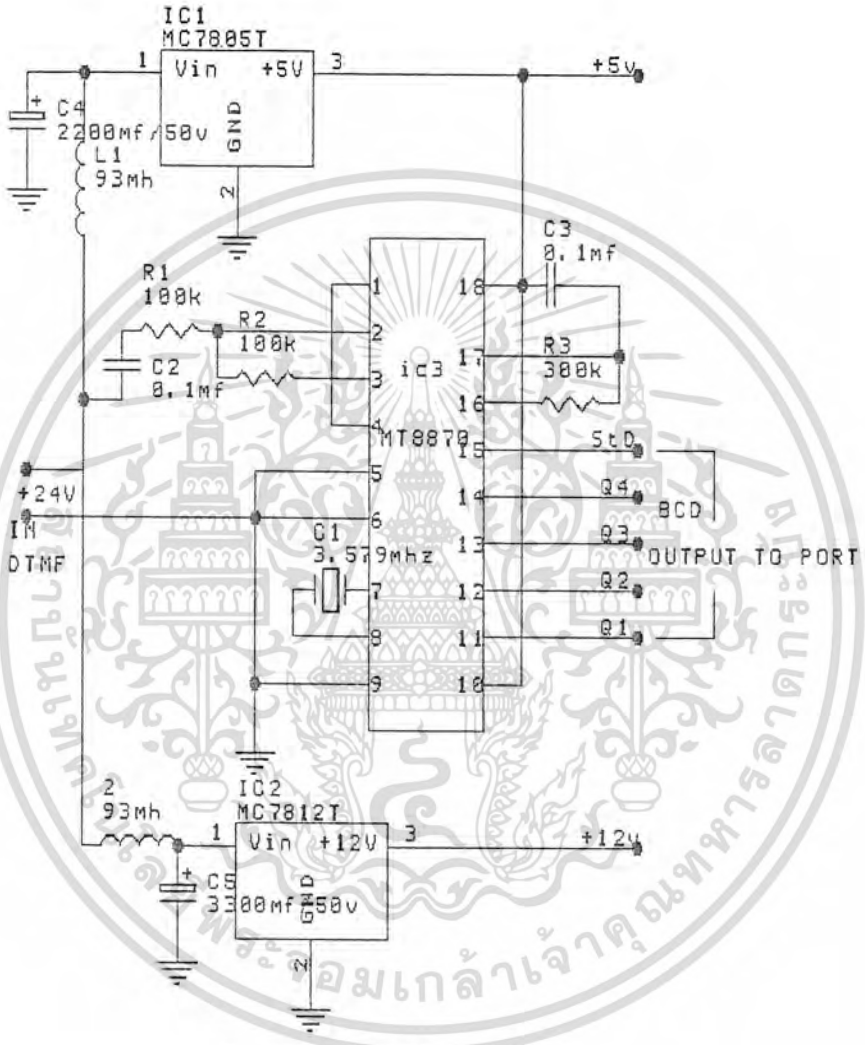


POWER SUPPLY CIRCUIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

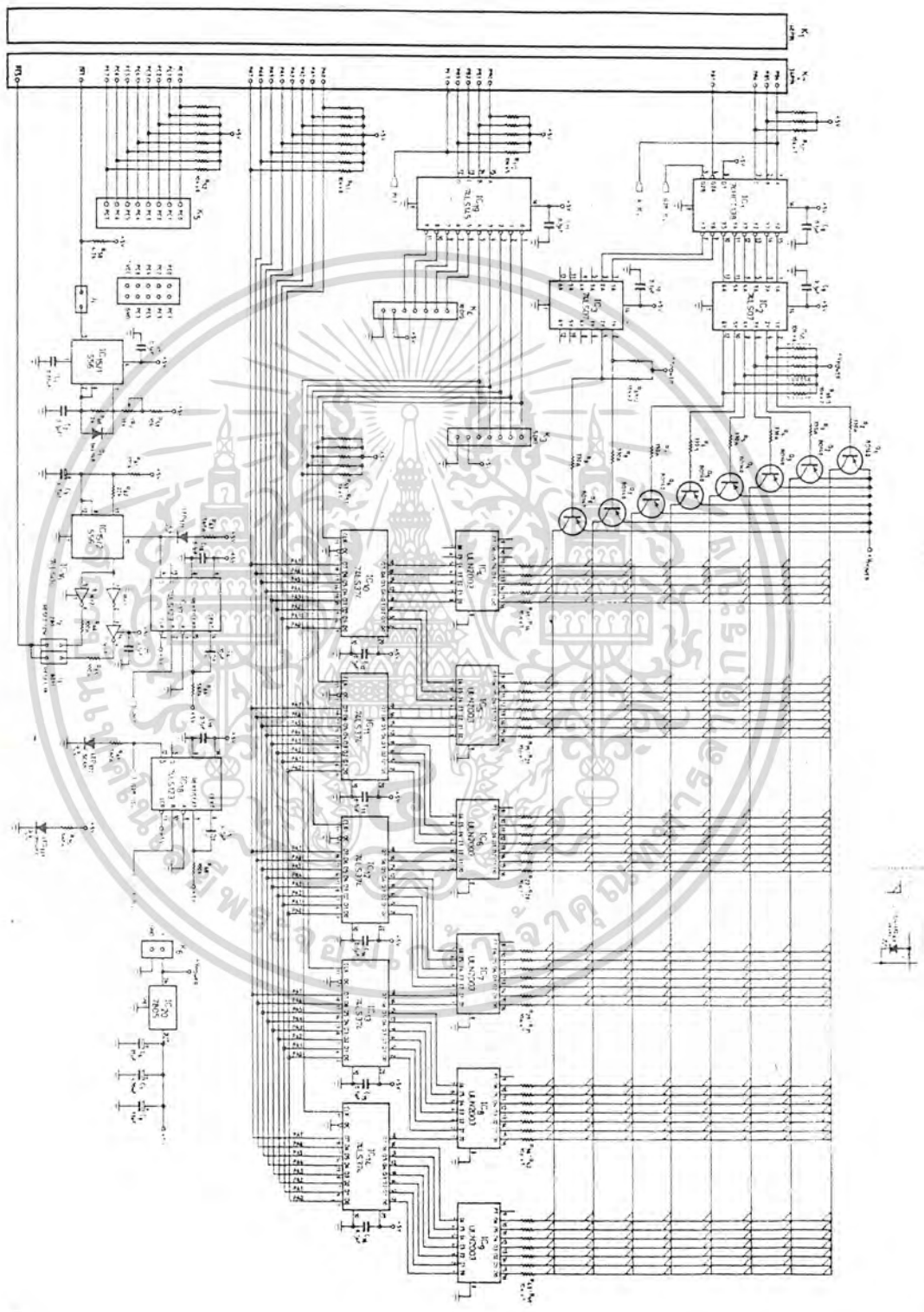


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

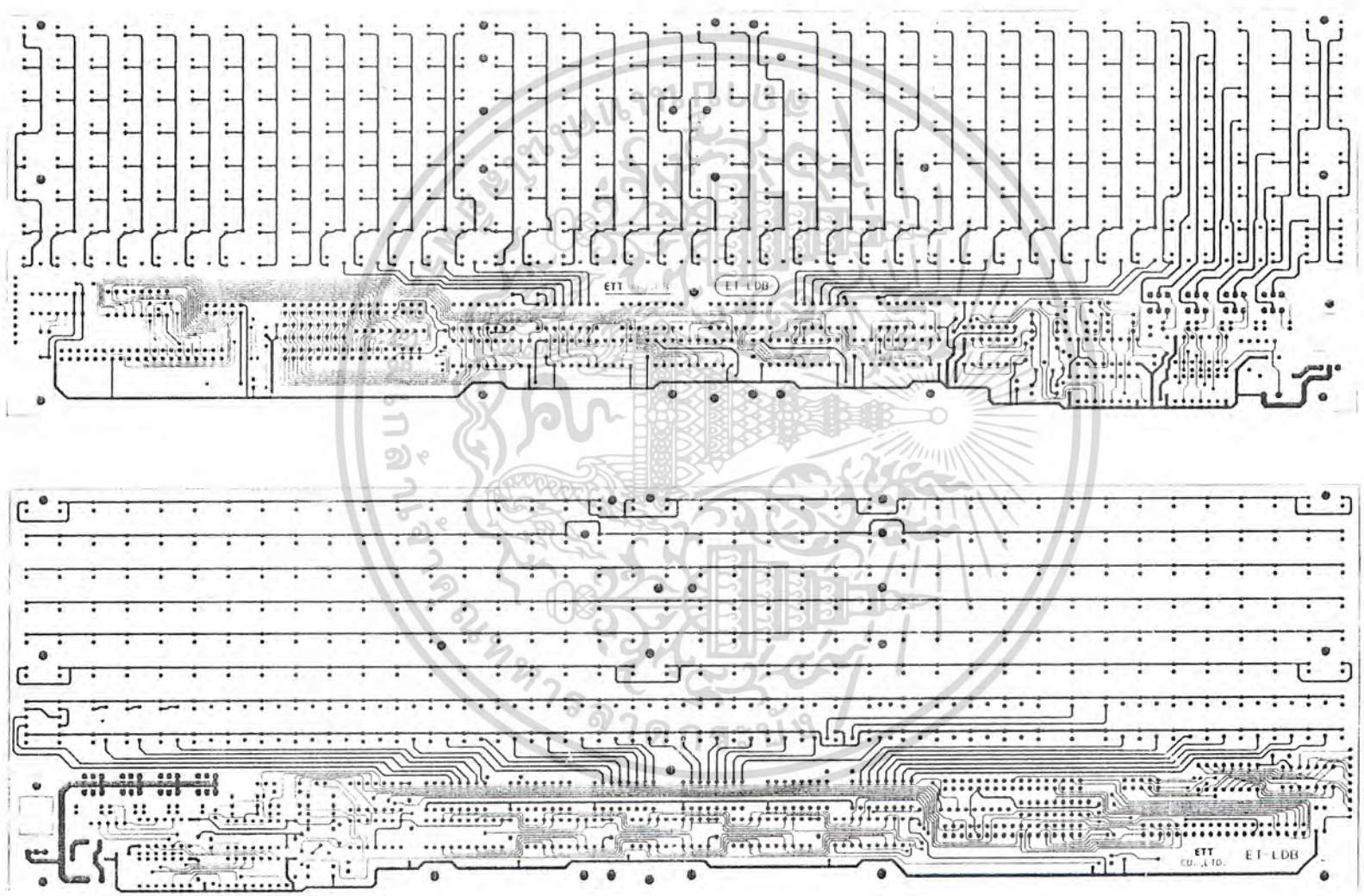


DECODER CIRCUIT

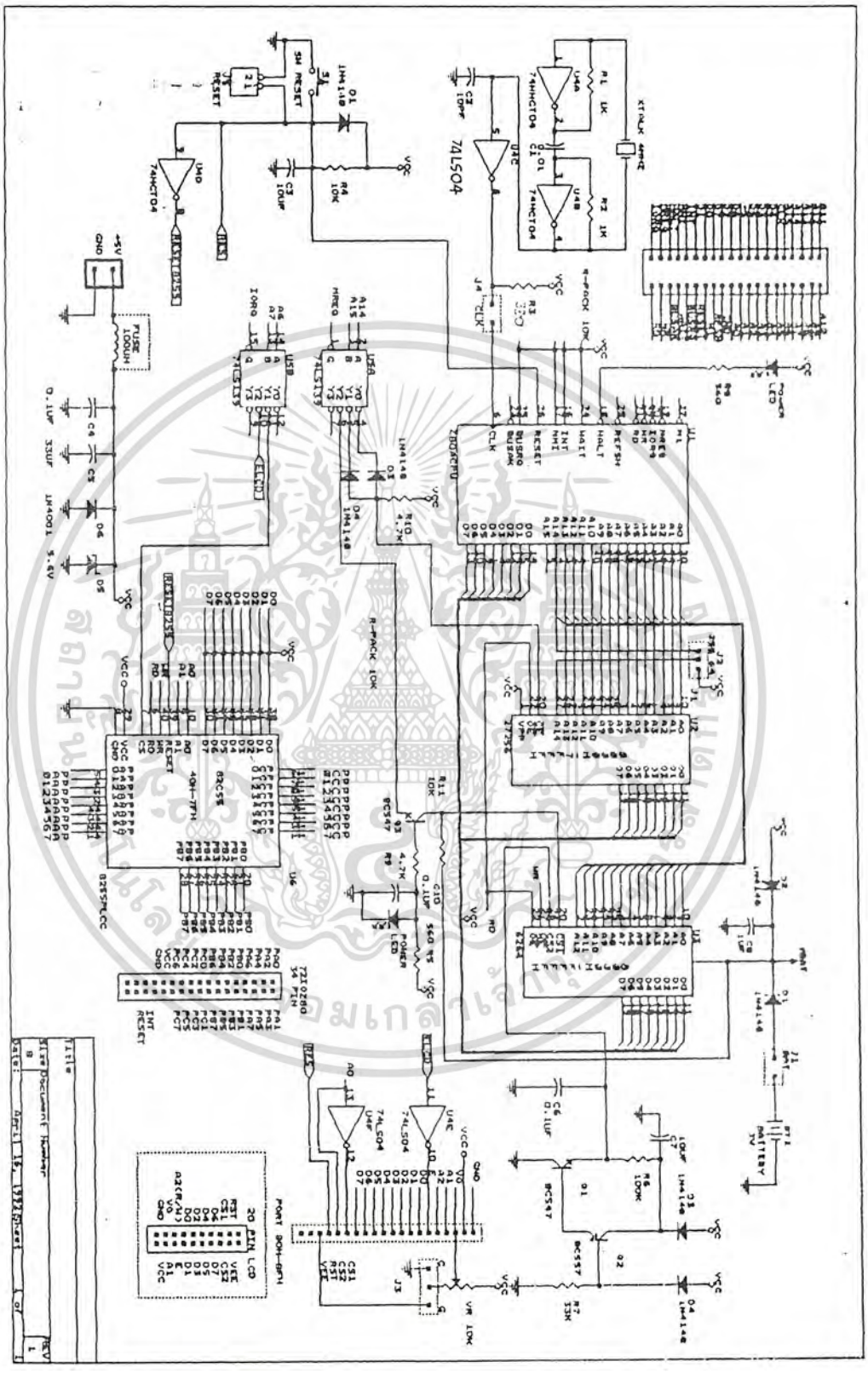
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



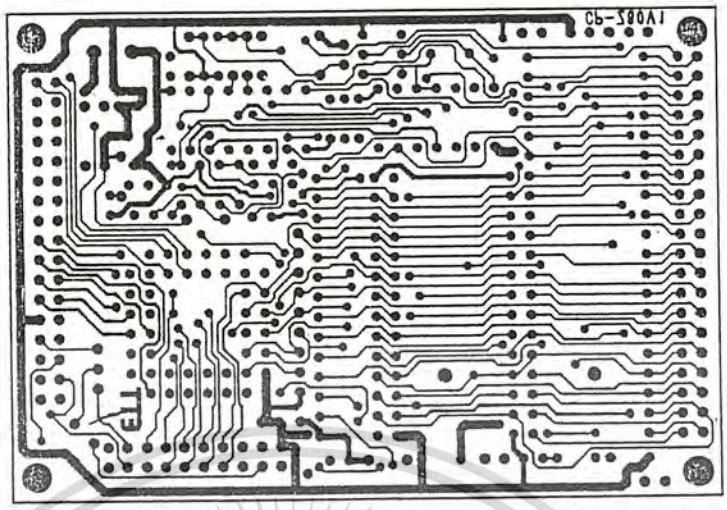
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

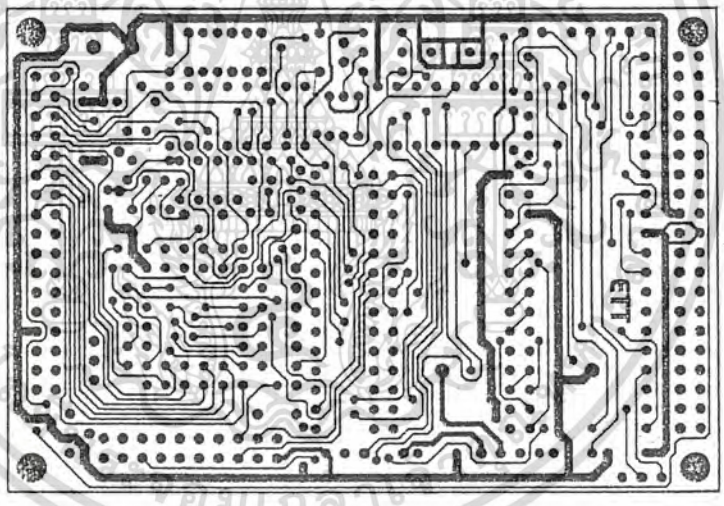


ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้เครื่อง

เครื่อง Remote Control Scoreboard diplayเป็นเครื่องที่เก็บข้อมูลไว้ในหน่วยความจำเมื่อจะใช้เครื่องต้องกดหมายเลขที่ Key Board ของ DTMF กด 0 จะขึ้นข้อความ Pottom Selection เพื่อให้เลือก Mode การทำงานว่าจะเลือก Mode ไหนในการนำแสดงผล ซึ่ง Mode การทำงานมีอยู่ 2 Mode

- Mode 1 กดหมายเลข 1 แสดงค่าตัวเลขการแสดงค่าตัวเลขสามารถแสดงค่าตัวเลขได้ 5 ตำแหน่ง ซึ่งสามารถเลือกได้ดังนี้

กด # 1 = แสดงตำแหน่งที่ 1

กด # 2 = แสดงตำแหน่งที่ 2

กด # 3 = แสดงตำแหน่งที่ 3

กด # 4 = แสดงตำแหน่งที่ 4

กด # 5 = แสดงตำแหน่งที่ 5

กด * ลบ หมายเลขตำแหน่งที่ต้องการ เช่น ต้องการลบตำแหน่งที่ 1 กด # 1 * ออกจาก mode ตัวเลขกด # ตามด้วย 0

- mode 2 กดหมายเลข 2,3,4,5,6 แสดงข้อความ ถ้าต้องการออกจาก mode 2 ให้กด 0

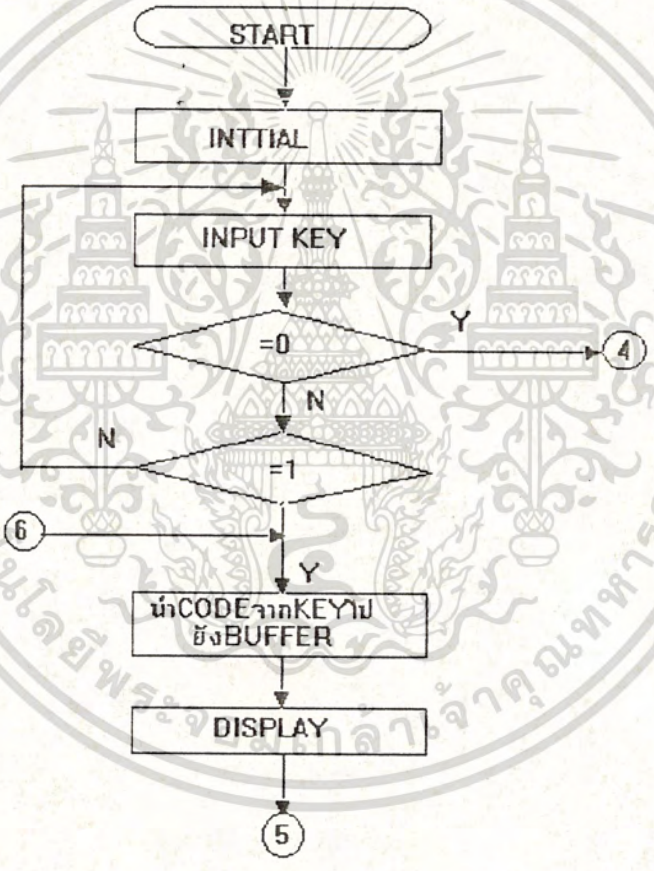
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

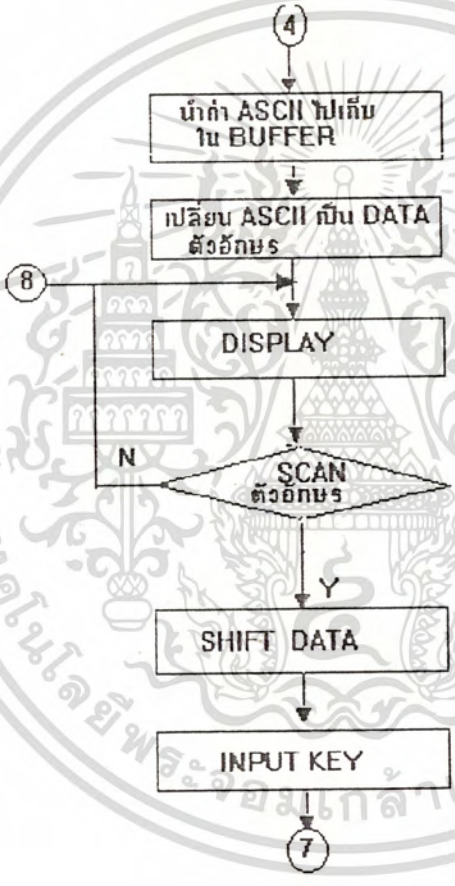
ในการออกแบบและสร้าง PROGRAMMABLE SCOREBOARD DISPLAY นี้ ผู้ทำโครงงาน ได้มีความรู้และความเข้าใจในการพัฒนาและใช้งานไมโครโปรเซสเซอร์ ให้มีขีดความสามารถสนองตอบความต้องการในการใช้งาน ดังนั้น การศึกษา ค้นคว้า เรียนรู้และการออกแบบอุปกรณ์ให้ทำงานได้อย่างมีประสิทธิภาพตามที่เราต้องการนั้น จำเป็นต้องมีพื้นความรู้ทาง HARDWRE และ SOFTWARE ประกอบกันการศึกษาและออกแบบป้ายแสดงอักษรวิ่งแบบโปรแกรมได้นี้ เป็นส่วนในการเริ่มให้รู้จักการประยุกต์ใช้งาน MICROPROCESSOR รวมทั้งการออกแบบโปรแกรมการใช้งานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

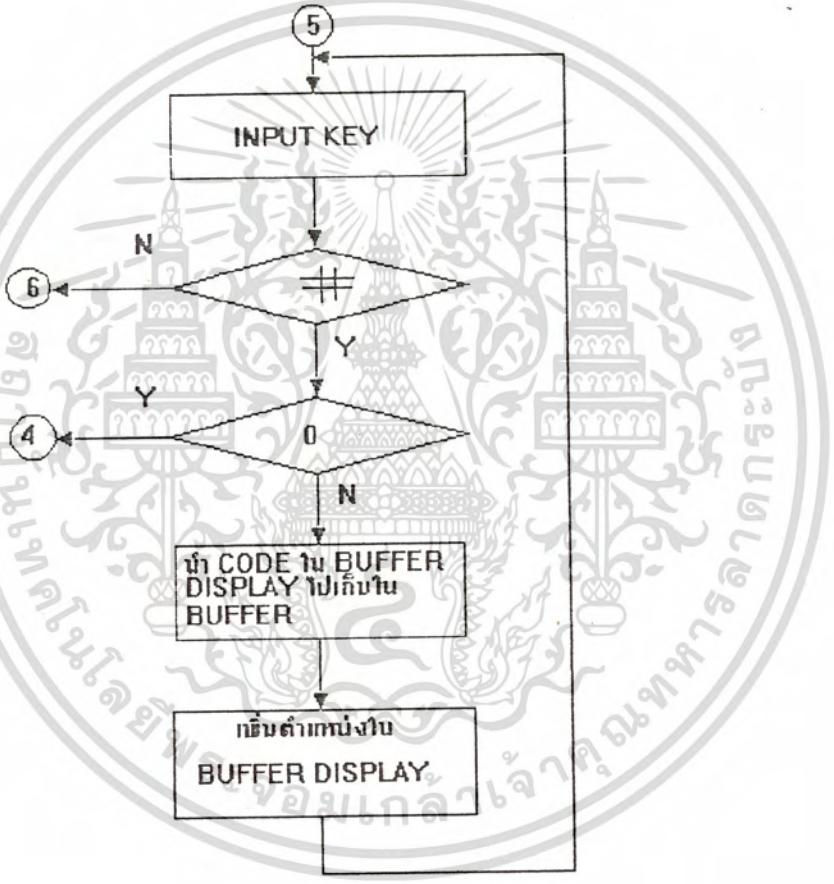
FLOWCHART



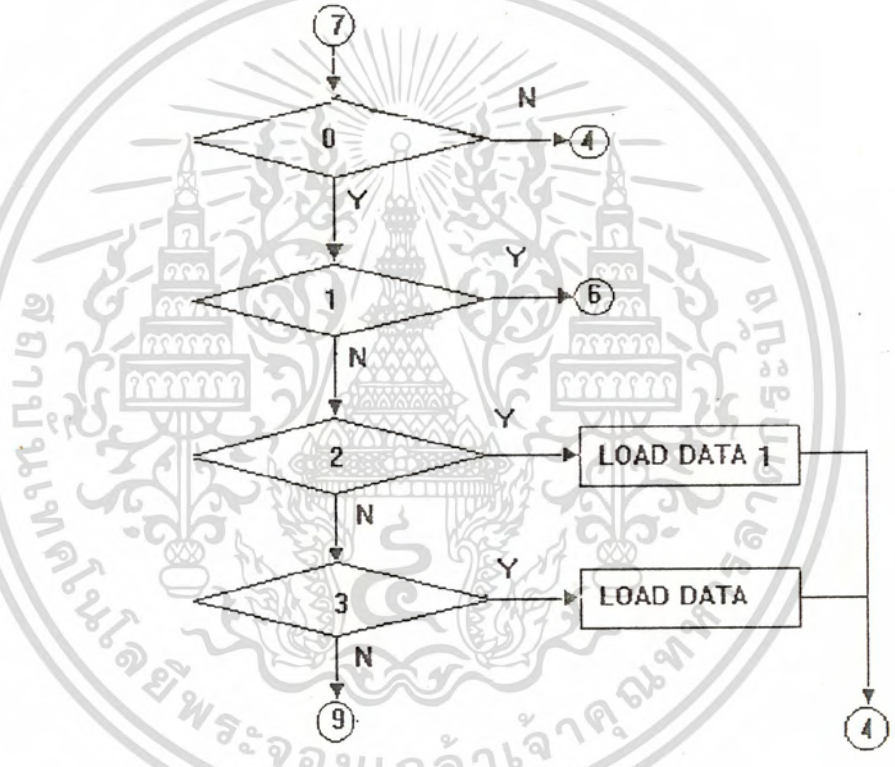
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



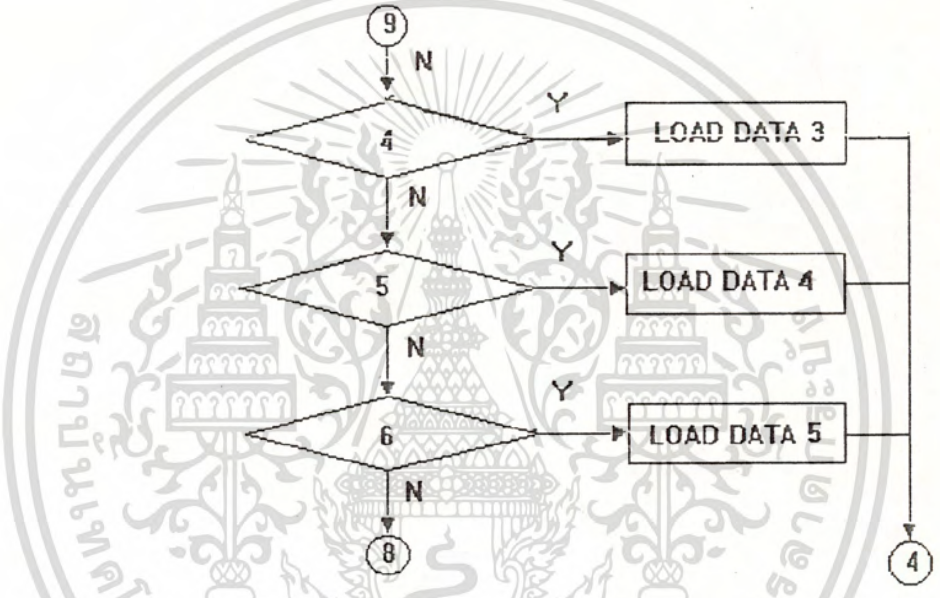
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ORG 8000H
COUNT5   EQU 9100H  ; นับจำนวนหลัก(5หลัก)
COUNT6   EQU 9101H  ; นับจำนวนหลักที่ใช้ในการ
                    ; เลื่อนข้อมูล(6หลัก)
COUNT8   EQU 9102H  ; นับจำนวนแถว(8แถว)
LOOP     EQU 9103H  ; ตำแหน่งในการเก็บ
                    ; จำนวนลูปของการ scan
LOOPCH   EQU 9104H  ; ตรวจสอบว่าตัวอักษรถูกเลื่อนไป
                    ; ครบ 8 บิตหรือยัง
ENDCH    EQU 9105H  ; ตรวจสอบว่าพบรหัสสิ้นสุดไฟล์หรือยัง
C_ENDCH EQU 9106H   ; ตรวจสอบว่าเมื่อพบรหัสสิ้นสุดไฟล์แล้ว
                    ; ตัวอักษรถูกเลื่อนไปครบ 5 หลักหรือยัง
BUFFER   EQU 9000H  ; ตำแหน่งเริ่มต้นของ display buffer
                    ; 5หลัก(9000H-9027H)
DISPY    EQU 9028H  ; ตำแหน่งเริ่มต้นของ display buffer
                    ; หลักที่ 6(9028H-902FH)
BUF1     EQU 9035H
NUMLP    EQU 05H    ; จำนวนลูปของการ scan
PORTA    EQU 40H    ; port A ของ 8255 บน ET-BOARD
PORTB    EQU 41H    ; port B ของ 8255 บน ET-BOARD
PORTC    EQU 42H    ; port C ของ 8255 บน ET-BOARD
CTPORT   EQU 43H    ; port control ของ 8255 บน ET-BOARD

         XOR A
START:   LD HL,0
ST1:     DEC HL
         LD A,L
         OR H
         JR NZ,ST1
         LD SP,9F00H
         XOR A
         CALL DELAY

         LD  A,89H
         OUT (CTPORT),A
         OUT (PORTB),A      ; OFF SCAN
         CALL DELAY
;-----------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ND:     CALL INITIAL
        ;-----------
        CALL CLEAR
        ;-----------
        LD B,00H
KKK:    IN A,(PORTC)
        AND 1FH
        CP 11H
        JR Z,MAIN
        CP 1AH
        JR Z,K1
        JP KKK
K1:     JP XX
        ;-----------
MAIN:   PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM:    PUSH IX
        ;-----------
        CALL OPEN
        ;-----------
        CALL CLOSE
        ;-----------
        POP IX
        INC IX
        CALL DEC8

        JR NZ,MMM
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN
        CP 1CH
        JP Z,FUNTT
        JP MAIN
;------------------------;
;       KEY2             ;
;------------------------;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FUNT2:  PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM2:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMM2
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN2
        CP 1CH
        JP Z,FUNTT
        JP FUNT2

;-----------------------;
;       KEY3            ;
;-----------------------------------------;
FUNT3:  LD IY,BUF1
        CALL MOV3
        PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM3:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL DEC8
        JR NZ,MMM3
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN3
        CP 1CH
        JP Z,FUNTT
        JP FUNT3
;-----------------------;
;       KEY4            ;
;-----------------------;
FUNT4:  LD IY,BUF1
        CALL MOV4
        PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM4:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMM4
        vLD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN4
        CP 1CH
        JP Z,FUNTT
        JP FUNT4
;-----------------------;
;       KEY5            ;
;-----------------------;
FUNT5:  LD IY,BUF1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FUNT5:  LD IY,BUF1
        CALL MOV5
        PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM5:   PUSH IX
        ;---------
        CALL OPEN

        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMM5
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN5
        CP 1CH
        JP Z,FUNTT
        JP FUNT5
;-----------------------;
FUNTT:  PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMMT:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMMT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD A,08H
                LD (COUNT8),A
                POP BC

                IN A,(PORTC)
                AND 1FH
          CP 12H
          JP Z,FUNT2
          CP 13H
          JP Z,FUNT3
          CP 14H
          JP Z,FUNT4
          CP 15H
          JP Z,FUNT5
          CP 11H
          JP Z,MAIN
          CP 1AH
          JP Z,FUNT10
          JP FUNTT
FUNT10:   JP XX
          ;-----------
          CALL DELAY
          ;-----------
          CALL INITIAL
          ;-----------
          CALL CLEAR
          ;-----------
          LD IY,DTBF1
          CALL DECODE
          CALL NEW

          ;-----------
MA11:     LD D,00H
          LD E,00H
          LD IX,BUFFER
M11:      PUSH IX

          ;-----------
          CALL OPEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------
CALL CLOSE
;-----------
POP IX
INC IX
CALL DEC8
JR NZ,M11
LD A,08H
LD (COUNT8),A

LD A,(LOOP)
DEC A
LD (LOOP),A
JR NZ,MA11      ; ตรวจสอบจำนวนรอบของการ scan
LD A,NUMLP
LD (LOOP),A
;-----------

LD A,(LOOPCH)
DEC A
LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
JR NZ,RS11      ; ไปครบ 8 บิตหรือยัง
LD A,08H
LD (LOOPCH),A

LD A,(ENDCH)
CP 01H
JR NZ,CO11      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
LD A,(C_ENDCH)
DEC A

LD (C_ENDCH),A
CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
JR NZ,TA11      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
LD A,05H
LD (C_ENDCH),A
XOR A
LD (ENDCH),A
LD IY,DTBF1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          JR CO11

TA11:     LD HL,TABLE
          JR CO21
CO11:     CALL DECODE
CO21:     CALL NEW

RS11:     ;-----------
          CALL ROTATE
          ;-----------
          CALL SHIFT
          ;-----------
          IN A,(PORTC)
          AND 1FH
          CP 11H
          JP Z,LL
          CP 12H
          JP Z,LLL
          CP 13H
          JP Z,LLLL
          CP 14H
          JP Z,LLLL1
          CP 15H
          JP Z,LLLL2

          CP 16H
          JP Z,LLLL3
          JP MA11
LL:       JP ND
LLL:      JP XXX1
LLLL:     JP XXX2
LLLL1:    JP XXX3
LLLL2:    JP XXX4
LLLL3:    JP XXX5
          ;-----------
          CALL DELAY
          ;-----------
          CALL INITIAL
          ;-----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                IN A,(PORTC)
                AND 1FH
            CP 12H
            JP Z,FUNT2
            CP 13H
            JP Z,FUNT3
            CP 14H
            JP Z,FUNT4
            CP 15H
            JP Z,FUNT5
            CP 11H
            JP Z,MAIN
            CP 1AH
            JP Z,FUNT10
            JP FUNTT
FUNT10:     JP XX
            ;-----------
xx:          CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF1
            CALL DECODE
            CALL NEW

            ;-----------
MA11:       LD D,00H
            LD E,00H
            LD IX,BUFFER
M11:        PUSH IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL CLEAR
        ;-----------
        LD IY,DTBF
        CALL DECODE
        CALL NEW
        ;-----------
MA22:   LD D,00H
        LD E,00H
        LD IX,BUFFER
M22:    PUSH IX
        ;-----------
        CALL OPEN
        ;-----------
        CALL CLOSE
        ;-----------
        POP IX
        INC IX
        CALL DEC8

        JR NZ,M22
        LD A,08H
        LD (COUNT8),A
        LD A,(LOOP)
        DEC A
        LD (LOOP),A
        JR NZ,MA22      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS22      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO22      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CP 16H
            JP Z,LLLL3
            JP MA11
LL:         JP ND
LLL:        JP XXX1
LLLL:       JP XXX2
LLLL1:      JP XXX3
LLLL2:      JP XXX4
LLLL3:      JP XXX5
            ;-----------
xxx1:       CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF
            CALL DECODE
            CALL NEW
            ;-----------
MA22:       LD D,00H
            LD E,00H
            LD IX,BUFFER
M22:        PUSH IX
            ;-----------
            CALL OPEN
            ;-----------
            CALL CLOSE
            ;-----------
            POP IX
            INC IX
            CALL DEC8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DEC A
        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA22      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
        LD A,05H
        LD (C_ENDCH),A
        XOR A
        LD (ENDCH),A
        LD IY,DTBF
        JR CO2
TA22:   LD HL,TABLE
        JR CO22

CO22:   CALL DECODE
CO2:    CALL NEW

RS22:   ;-----------
        CALL ROTATE
        ;-----------
        CALL SHIFT
        ;-----------
        IN A,(PORTC)
        AND 1FH
        CP 1AH
        JP Z,LAY1
        JP MA22
LAY1:   JP XX
        ;-----------
XXX2:   CALL DELAY
        ;-----------
        CALL INITIAL
        ;-----------
        CALL CLEAR
        ;-----------
        LD IY,DTBF2
        CALL DECODE
        CALL NEW
        ;-----------
MA33:   LD D,00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD E,00H
        LD IX,BUFFER
M33:    PUSH IX
        ;-----------
        CALL OPEN
        ;-----------

        CALL CLOSE
        ;-----------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,M33
        LD A,08H
        LD (COUNT8),A
        LD A,(LOOP)
        DEC A
        LD (LOOP),A
        JR NZ,MA33      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS33      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO33      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
        DEC A
        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA33      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
        LD A,05H
        LD (C_ENDCH),A
        XOR A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD (ENDCH),A
        LD IY,DTBF2
        JR CO33
TA33:   LD HL,TABLE
        JR CO3
CO33:   CALL DECODE
CO3:    CALL NEW
RS33:   ;----------
        CALL ROTATE
        ;----------
        CALL SHIFT
        ;----------
        IN A,(PORTC)
        AND 1FH
        CP 1AH
        JP Z,LAY1
        JP MA33
        ;----------
XXX3:   CALL DELAY
        ;----------
        CALL INITIAL
        ;----------
        CALL CLEAR
        ;----------
        LD IY,DTBF3
        CALL DECODE
        CALL NEW
        ;-----------
MA44:   LD D,00H
        LD E,00H
        LD IX,BUFFER
M44:    PUSH IX

        ;----------
        CALL OPEN
        ;-----------
        CALL CLOSE
        ;----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP IX
        INC IX
        CALL DEC8
        JR NZ,M44
        LD A,08H
        LD (COUNT8),A
        LD A,(LOOP)
        DEC A
        LD (LOOP),A
        JR NZ,MA44      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS44      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO44      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
        DEC A
        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA44      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง

        LD A,05H
        LD (C_ENDCH),A
        XOR A
        LD (ENDCH),A
        LD IY,DTBF3
        JR CO44
TA44:   LD HL,TABLE
        JR CO4
CO44:   CALL DECODE
CO4:    CALL NEW
RS44:   ;-----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL ROTATE
            ;-----------
            CALL SHIFT
            ;-----------
            IN A,(PORTC)
            AND 1FH
            CP 1AH
            JP Z,LAY1
            JP MA44
            ;-----------
XXX4:       CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF4
            CALL DECODE
            CALL NEW
            ;-----------
MA55:       LD D,00H

            LD E,00H
            LD IX,BUFFER
M55:        PUSH IX
            ;-----------
            CALL OPEN
            ;-----------
            CALL CLOSE
            ;-----------
            POP IX
            INC IX
            CALL DEC8
            JR NZ,M55
            LD A,08H
            LD (COUNT8),A
            LD A,(LOOP)
            DEC A
            LD (LOOP),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR NZ,MA55      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS55      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO55      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
        DEC A

        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA55      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
        LD A,05H
        LD (C_ENDCH),A
        XOR A
        LD (ENDCH),A
        LD IY,DTBF4
        JR CO55
TA55:   LD HL,TABLE
        JR CO5
CO55:   CALL DECODE
CO5:    CALL NEW
RS55:   ;-----------
        CALL ROTATE
        ;-----------
        CALL SHIFT
        ;-----------
        IN A,(PORTC)
        AND 1FH
        CP 1AH
        JP Z,LAY1
        JP MA55
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ;-----------
            CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF5
            CALL DECODE

            CALL NEW
            ;-----------
MA66:       LD D,00H
            LD E,00H
            LD IX,BUFFER
M66:        PUSH IX
            ;-----------
            CALL OPEN
            ;-----------
            CALL CLOSE
            ;-----------
            POP IX
            INC IX
            CALL DEC8
            JR NZ,M66
            LD A,08H
            LD (COUNT8),A
            LD A,(LOOP)
            DEC A
            LD (LOOP),A
            JR NZ,MA66      ; ตรวจสอบจำนวนรอบของการ scan
            LD A,NUMLP
            LD (LOOP),A
            ;-----------
            LD A,(LOOPCH)
            DEC A
            LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
            JR NZ,RS66      ; ไปครบ 8 บิตหรือยัง
            LD A,08H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD (LOOPCH),A
            LD A,(ENDCH)
            CP 01H

            JR NZ,CO66      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
            LD A,(C_ENDCH)
            DEC A
            LD (C_ENDCH),A
            CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
            JR NZ,TA66      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
            LD A,05H
            LD (C_ENDCH),A
            XOR A
            LD (ENDCH),A
            LD IY,DTBF4
            JR CO66
TA66:       LD HL,TABLE
            JR CO6
CO66:       CALL DECODE
CO6:        CALL NEW
RS66:       ;-----------
            CALL ROTATE
            ;-----------
            CALL SHIFT
            ;-----------
            IN A,(PORTC)
            AND 1FH
            CP 1AH
            JP Z,LAY1
            JP MA66
        ;---------------;
        ;     MOV3      ;
        ;---------------;
MOV3:       LD B,08H
            CALL MV1
            RET

        ;---------------;
MV1:        PUSH IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD IX,BUFFER
MV:     LD A,(IX+8)
        LD (IY),A
        INC IX
        INC IY
        DJNZ MV
        POP IX
        RET
    ;---------------;
    ;    MOV4       ;
    ;---------------;
MOV4:   LD B,10H
        CALL MV1
        RET
    ;---------------;
    ;    MOV5       ;
    ;---------------;
MOV5:   LD B,18H
        CALL MV1
        RET
    ;---------------;
    ;    KEYIN      ;
    ;---------------;
KEYIN:  PUSH DE
        IN A,(PORTC)
        AND 1FH
        LD D,A
        INC B
        CP 11H
        JR Z,KEY1

        CP 12H
        JR Z,KEY2
        CP 13H
        JR Z,KEY3
        CP 14H
        JR Z,KEY4
        CP 15H
        JR Z,KEY5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CP 16H
        JR Z,KEY6
        CP 17H
        JR Z,KEY7
        CP 18H
        JR Z,KEY8
        CP 19H
        JR Z,KEY9
        CP 1AH
        JR Z,KEY0
        CP 1BH
        JR Z,KEY01
        POP DE
        DEC B
        RET
KEY01:  LD HL,TABLE
        CALL KE1
        CALL K22
        POP DE
        RET
KEY0:   LD HL,T0
        CALL KE1
        CALL K22
        POP DE

        RET
KEY1:   LD HL,T1
        CALL KE1
        CALL K22
        POP DE
        RET
KEY2:   LD HL,T2
        CALL KE1
        CALL K22
        POP DE
        RET
KEY3:   LD HL,T3
        CALL KE1
        CALL K22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP DE
        RET
KEY4:   LD HL,T4
        CALL KE1
        CALL K22
        POP DE
        RET
KEY5:   LD HL,T5
        CALL KE1
        CALL K22
        POP DE
        RET
KEY6:   LD HL,T6
        CALL KE1
        CALL K22
        POP DE
        RET
KEY7:   LD HL,T7
        CALL KE1
        CALL K22
        POP DE
        RET
KEY8:   LD HL,T8
        CALL KE1
        CALL K22
        POP DE
        RET
KEY9:   LD HL,T9
        CALL KE1
        CALL K22
        POP DE
        RET
KEY10:  LD HL,T0
        CALL KE1
        CALL K22
        POP DE
        RET
KEY11:  LD HL,T1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CALL KE1
          CALL K22
          POP DE
          RET
KEY12:    LD HL,T2
          CALL KE1
          CALL K22
          POP DE
          RET
KEY13:    LD HL,T3
          CALL KE1
          CALL K22
          POP DE
          RET
KEY14:    LD HL,T4
          CALL KE1
          CALL K22
          POP DE
          RET
KEY15:    LD HL,T5
          CALL KE1
          CALL K22
          POP DE
          RET
KEY16:    LD HL,T6
          CALL KE1
          CALL K22
          POP DE
          RET
KEY17:    LD HL,T7
          CALL KE1
          CALL K22
          POP DE
          RET
KEY18:    LD HL,T8
          CALL KE1
          CALL K22
          POP DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
KEY19:  LD HL,T9
        CALL KE1
        CALL K22
        POP DE
        RET

K22:    LD B,8H
K9:     LD A,(HL)
        LD (IX),A
        INC HL
        INC IX
        DJNZ K9
        RET
KE1:    PUSH BC
        LD B,08H
KE2:    LD A,(IX)
        DEC IX
        LD (IX),A
        DJNZ KE2
        POP BC
        RET
  ;---------------;
  ;   INKEY2      ;
  ;---------------;
KEYIN2: PUSH DE
        IN A,(PORTC)
        AND 1FH
        LD D,A
        INC B
        CP 1AH
        JR Z,KEY20
        CP 11H
        JR Z,KEY21
        CP 12H
        JR Z,KEY22
        CP 13H
        JR Z,KEY23
        CP 14H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR Z,KEY24
        CP 15H
        JR Z,KEY25
        CP 16H
        JR Z,KEY26
        CP 17H
        JR Z,KEY27
        CP 18H
        JR Z,KEY28
        CP 19H
        JR Z,KEY29
        CP 1BH
        JR Z,KEY02
        POP DE
        DEC B
        RET
KEY20:  LD HL,T0
        CALL K22
        POP DE
        RET
KEY21:  LD HL,T1
        CALL K22
        POP DE
        RET
KEY22:  LD HL,T2
        CALL K22
        POP DE
        RET
KEY23:  LD HL,T3
        CALL K22
        POP DE
        RET

KEY24:  LD HL,T4
        CALL K22
        POP DE
        RET
KEY25:  LD HL,T5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CALL K22
          POP DE
          RET
KEY26:    LD HL,T6
          CALL K22
          POP DE
          RET
KEY27:    LD HL,T7
          CALL K22
          POP DE
          RET
KEY28:    LD HL,T8
          CALL K22
          POP DE
          RET
KEY29:    LD HL,T9
          CALL K22
          POP DE
          RET
KEY02:    LD HL,TABLE
          CALL K22
          POP DE
          RET
    ;---------------;
    ;    INKEY3     ;
    ;---------------;
KEYIN3:   PUSH DE

          IN A,(PORTC)
          AND 1FH
          LD D,A
          INC B
          CP 1AH
          JR Z,KEY30
          CP 11H
          JR Z,KEY31
          CP 12H
          JR Z,KEY32
          CP 13H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR Z,KEY33
        CP 14H
        JR Z,KEY34
        CP 15H
        JR Z,KEY35
        CP 16H
        JR Z,KEY36
        CP 17H
        JR Z,KEY37
        CP 18H
        JR Z,KEY38
        CP 19H
        JR Z,KEY39
        CP 1BH
        JR Z,KEY03
        POP DE
        DEC B
        RET
KEY30:  LD HL,T0
        CALL KE11
        CALL K22

        POP DE
        RET
KEY31:  LD HL,T1
        CALL KE11
        CALL K22
        POP DE
        RET
KEY32:  LD HL,T2
        CALL KE11
        CALL K22
        POP DE
        RET
KEY33:  LD HL,T3
        CALL KE11
        CALL K22
        POP DE
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KEY34:  LD HL,T4
        CALL KE11
        CALL K22
        POP DE
        RET
KEY35:  LD HL,T5
        CALL KE11
        CALL K22
        POP DE
        RET
KEY36:  LD HL,T6
        CALL KE11
        CALL K22
        POP DE
        RET

KEY37:  LD HL,T7
        CALL KE11
        CALL K22
        POP DE
        RET
KEY38:  LD HL,T8
        CALL KE11
        CALL K22
        POP DE
        RET
KEY39:  LD HL,T9
        CALL KE11
        CALL K22
        POP DE
        RET
KEY03:  LD HL,TABLE
        CALL KE11
        CALL K22
        RET
KE11:   PUSH BC
        LD B,08H
        LD IY,BUF1
KE12:   LD A,(IY)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD (IX),A
        INC IY
        INC IX
        DJNZ KE12
        POP BC
        RET
    ;---------------;
    ;    INKEY4     ;
    ;---------------;

KEYIN4: PUSH DE
        IN A,(PORTC)
        AND 1FH
        LD D,A
        CP 1AH
        JR Z,KEY40
        CP 11H
        JR Z,KEY41
        CP 12H
        JR Z,KEY42
        CP 13H
        JR Z,KEY43
        CP 14H
        JR Z,KEY44
        CP 15H
        JR Z,KEY45
        CP 16H
        JR Z,KEY46
        CP 17H
        JR Z,KEY47
        CP 18H
        JR Z,KEY48
        CP 19H
        JR Z,KEY49
        CP 1BH
        JR Z,KEY04
        POP DE
        RET
KEY40:  LD HL,TO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CALL KE14
          CALL K22
          POP DE

          RET
KEY41:    LD HL,T1
          CALL KE14
          CALL K22
          POP DE
          RET
KEY42:    LD HL,T2
          CALL KE14
          CALL K22
          POP DE
          RET
KEY43:    LD HL,T3
          CALL KE14
          CALL K22
          POP DE
          RET
KEY44:     LD HL,T4
          CALL KE14
          CALL K22
          POP DE
          RET
KEY45:    LD HL,T5
          CALL KE14
          CALL K22
          POP DE
          RET
KEY46:    LD HL,T6
          CALL KE14
          CALL K22
          POP DE
          RET
KEY47:    LD HL,T7

          CALL KE14
          CALL K22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          POP DE
          RET
KEY48:    LD HL,T8
          CALL KE14
          CALL K22
          POP DE
          RET
KEY49:    LD HL,T9
          CALL KE14
          CALL K22
          POP DE
          RET
KEY04:    LD HL,TABLE
          CALL KE14
          CALL K22
          POP DE
          RET
KE14:     LD B,10H
          LD IY,BUF1
KE4:      LD A,(IY)
          LD (IX),A
          INC IY
          INC IX
          DJNZ KE4
          RET
     ;---------------;
     ;    INKEY5     ;
     ;---------------;
KEYIN5:   PUSH DE
          IN A,(PORTC)

          INC B.
          CP 1AH
          JR Z,KEY50
          CP 11H
          JR Z,KEY51
          CP 12H
          JR Z,KEY52
          CP 13H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR Z,KEY53
        CP 14H
        JR Z,KEY54
        CP 15H
        JR Z,KEY55
        CP 16H
        JR Z,KEY56
        CP 17H
        JR Z,KEY57
        CP 18H
        JR Z,KEY58
        CP 19H
        JR Z,KEY59
        CP 1BH
        JR Z,KEY05
        POP DE
        DEC B
        RET
KEY05:  LD HL,TABLE
        CALL KE15
        CALL K22
        POP DE
        RET
KEY50:  LD HL,T0

        CALL KE15
        CALL K22
        POP DE
        RET
KEY51:  LD HL,T1
        CALL KE15
        CALL K22
        POP DE
        RET
KEY52:  LD HL,T2
        CALL KE15
        CALL K22
        POP DE
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KEY53:  LD HL,T3
        CALL KE15
        CALL K22
        POP DE
        RET
KEY54:  LD HL,T4
        CALL KE15
        CALL K22
        POP DE
        RET
KEY55:  LD HL,T5
        CALL KE15
        CALL K22
        POP DE
        RET
KEY56:  LD HL,T6
        CALL KE15
        CALL K22
        POP DE
        RET
KEY57:  LD HL,T7
        CALL KE15
        CALL K22
        POP DE
        RET
KEY58:  LD HL,T8
        CALL KE15
        CALL K22
        POP DE
        RET
KEY59:  LD HL,T9
        CALL KE15
        CALL K22
        POP DE
        RET
KE15:   PUSH BC
        LD B,18H
        LD IY,BUF1
KE5:    LD A,(IY)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD (IX),A
            INC IY
            INC IX
            DJNZ KE5
            POP BC
            RET
        ;---------------;
        ;    DALAY      ;
        ;---------------;
DELAY:      PUSH DE
            LD E,0F9H
DDD:        DEC E

            JR NZ,DDD
            POP DE
            RET
;-------------------------; หาตำแหน่งเริ่มต้นของฟอนต์
; DECODE  ASCII -> TABLE ; ตัวอักษรโดยเปลี่ยน data
;-------------------------; ของตัวอักษรจากรหัส ASCII
DECODE:     LD HL,TABLE     ; 1 ไบต์ ไปเป็นฟอนตของตัวอักษร
            LD BC,0008H     ; ขนาด 8 ไบต์ เพื่อที่จะนำ data
            LD A,(IY+0)     ; ขนาด 8 ไบต์นี้มาแสดงผล
            CP 1AH
            JR NZ,D_CODE

            LD A,(ENDCH)
            SET 0,A
            LD (ENDCH),A
            JR RETURN
D_CODE:     ADD HL,BC
            DEC A
            JR NZ,D_CODE
            DEC H
            INC IY
RETURN:     RET
;-------------------------; ทำการส่ง data ที่อยู่ในตำแหน่ง
;           ON_LED        ; display buffer ทั้ง 5 หลัก
;-------------------------; ออกไปยัง LED
OPEN:       PUSH DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
A1:     LD A,D
        OR 0F0H
        OUT (PORTB),A
        LD A,(IX+0)
        OUT (PORTA),A
        LD A,0FFH

        OUT (PORTB),A
        INC D
        CALL IX8
        CALL DEC5
        JR NZ,A1
        LD A,05H
        LD (COUNT5),A
        POP DE
        LD A,D
        AND 0F0H
        OUT (PORTB),A
        ADD A,10H
        LD D,A
        CALL DELAY
        RET
;-------------------------; ทำการ clear data
;       OFF LED           ; ที่ส่งไปยัง LED
;-------------------------;
CLOSE:  PUSH DE
A2:     LD A,E
        OR 0F0H
        OUT (PORTB),A
        LD A,00H
        OUT (PORTA),A
        LD A,0FFH
        OUT (PORTB),A
        INC E
        CALL DEC5
        JR NZ,A2
        LD A,05H
        LD (COUNT5),A
        POP DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD A,E
            AND 0F0H
            OUT (PORTB),A
            ADD A,10H
            LD E,A
            CALL DELAY
            RET
;-------------------------; ทำการหมุนข้อมูลที่อยู่ในตำแหน่ง
;         ROTATE DATA     ; display buffer ทั้ง 6 หลัก
;-------------------------;
ROTATE:     LD IX,BUFFER
R1:         PUSH IX
R2:         LD A,(IX+0)
            RLCA
            LD (IX+0),A
            CALL IX8
            CALL DEC6
            JR NZ,R2
            LD A,06H
            LD (COUNT6),A
            POP IX
            INC IX
            CALL DEC8
            JR NZ,R1
            LD A,08H
            LD (COUNT8),A
            RET
;-------------------------; ทำการเลื่อนข้อมูลที่อยู่ในตำแหน่ง
;         SHIFT DATA      ; display buffer ทั้ง 6 หลัก
;-------------------------; ไปทางซ้าย 1 บิต
SHIFT:      LD IX,BUFFER
S1:         PUSH IX

S2:         RES 0,(IX+0)
            LD A,(IX+8)
            AND 01H
            ADD A,(IX+0)
            LD (IX+0),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL IX8
        CALL DEC5
        JR NZ,S2
        LD A,05H
        LD (COUNT5),A
        POP IX
        INC IX
        CALL DEC8
        JR NZ,S1
        LD A,08H
        LD (COUNT8),A
        RET
;-------------------------;
;    INITIAL VALUE        ; กำหนดค่าเริ่มต้นต่างๆ
;-------------------------;
INITIAL: XOR A
        LD (ENDCH),A
        LD A,05H
        LD (COUNT5),A
        LD (C_ENDCH),A
        LD A,06H
        LD (COUNT6),A
        LD A,NUMLP
        LD (LOOP),A
        LD A,08H
        LD (COUNT8),A
        LD (LOOPCH),A

        RET
;-------------------------; ทำการ clear data ที่อยู่ในตำแหน่ง
;       CLEAR DATA        ; display buffer ทั้ง 5 หลัก
;-------------------------; ให้มีค่าเป็น "0"
CLEAR:  LD B,28H
        LD IX,BUFFER
        LD A,00H
CCC:    LD (IX+0),A
        INC IX
        DJNZ CCC
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------; นำ data ของตัวอักษรตัวใหม่ ที่จะ
;     NEW CHARACTER       ; นำออกมาแสดงผลไปไว้ในตำแหน่ง
;-------------------------; display buffer หลักที่ 6
NEW:    PUSH HL
        LD IX,DISPY
        LD B,08H
X1:     LD A,(HL)
        LD (IX+0),A
        INC HL
        INC IX
        DJNZ X1
        POP HL
        RET
;-------------------------;
;       DALAY             ;
;-------------------------;
DELAY:  PUSH DE
        LD E,0FEH
DDD:    DEC E
        LD D,0FFH
D1:     DEC D

        JR NZ,D1
        JR NZ,DDD
        POP DE
        RET
IX8:    LD BC,0008H       ; เพิ่มค่า IX ขึ้น 8
        ADD IX,BC
        RET
DEC5:   LD A,(COUNT5)     ; ลดค่า count5 ลง 1
        DEC A
        LD (COUNT5),A
        RET
DEC6:   LD A,(COUNT6)     ; ลดค่า count6 ลง 1
        DEC A
        LD (COUNT6),A
        RET
DEC8:   LD A,(COUNT8)     ; ลดค่า count8 ลง 1
        DEC A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD (COUNT8),A
        RET
;-----------------------------------------------;
;      ฟอนต์ตัวอักษร โดยเริ่มจากรหัส ASCII 20H-7FH    ;
;-----------------------------------------------;
TABLE:  DB 00H,00H,00H,00H
        DB 00H,00H,00H,00H    ;

        DB 10H,10H,10H,10H
        DB 00H,00H,10H,00H    ; !

        DB 28H,28H,28H,00H
        DB 00H,00H,00H,00H    ; "

        DB 28H,28H,7CH,28H

        DB 7CH,28H,28H,00H    ; #

        DB 10H,3CH,50H,38H
        DB 14H,78H,10H,00H    ; $

        DB 60H,64H,08H,10H
        DB 20H,4CH,0CH,00H    ; %

        DB 30H,48H,50H,20H
        DB 54H,48H,34H,00H    ; &

        DB 30H,10H,20H,00H
        DB 00H,00H,00H,00H    ; '

        DB 08H,10H,20H,20H
        DB 20H,10H,08H,00H    ; (

        DB 20H,10H,08H,08H
        DB 08H,10H,20H,00H    ; )

        DB 00H,10H,54H,38H
        DB 54H,10H,00H,00H    ; *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 00H,10H,10H,7CH
        DB 10H,10H,00H,00H      ; +

        DB 00H,00H,00H,00H
        DB 30H,10H,20H,00H      ; ,

        DB 00H,00H,00H,7CH
        DB 00H,00H,00H,00H      ; -


        DB 00H,00H,00H,00H
        DB 00H,30H,30H,00H      ; .

        DB 00H,04H,08H,10H
        DB 20H,40H,00H,00H      ; /

T0:     DB 38H,44H,4CH,54H
        DB 64H,44H,38H,00H      ; 0

T1:     DB 10H,30H,10H,10H
        DB 10H,10H,38H,00H      ; 1

T2:     DB 38H,44H,04H,08H
        DB 10H,20H,7CH,00H      ; 2

T3:     DB 7CH,08H,10H,08H
        DB 04H,44H,38H,00H      ; 3

T4:     DB 08H,18H,28H,48H
        DB 7CH,08H,08H,00H      ; 4

T5:     DB 7CH,40H,78H,04H
        DB 04H,44H,38H,00H      ; 5

T6:     DB 18H,20H,40H,78H
        DB 44H,44H,38H,00H      ; 6

T7:     DB 7CH,04H,08H,10H
        DB 20H,20H,20H,00H      ; 7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
T8:     DB 38H,44H,44H,38H
        DB 44H,44H,38H,00H      ; 8

T9:     DB 38H,44H,44H,3CH
        DB 04H,08H,30H,00H      ; 9

        DB 00H,30H,30H,00H
        DB 30H,30H,00H,00H      ; :

        DB 00H,30H,30H,00H
        DB 30H,10H,20H,00H      ; ;

        DB 08H,10H,20H,40H
        DB 20H,10H,08H,00H      ; <

        DB 00H,00H,7CH,00H
        DB 7CH,00H,00H,00H      ; =

        DB 20H,10H,08H,04H
        DB 08H,10H,20H,00H      ; >

        DB 38H,44H,04H,08H
        DB 10H,00H,10H,00H      ; ?

        DB 38H,44H,04H,34H
        DB 54H,54H,38H,00H      ; @

        DB 38H,44H,44H,44H
        DB 7CH,44H,44H,00H      ; A

        DB 78H,44H,44H,78H
        DB 44H,44H,78H,00H      ; B

        DB 38H,44H,40H,40H
        DB 40H,44H,38H,00H      ; C

        DB 70H,48H,44H,44H
        DB 44H,48H,70H,00H      ; D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 7CH,40H,40H,78H
DB 40H,40H,7CH,00H      ; E

DB 7CH,40H,40H,78H
DB 40H,40H,40H,00H      ; F

DB 38H,44H,40H,5CH
DB 44H,44H,3CH,00H      ; G

DB 44H,44H,44H,7CH
DB 44H,44H,44H,00H      ; H

DB 38H,10H,10H,10H
DB 10H,10H,38H,00H      ; I

DB 1CH,08H,08H,08H
DB 08H,48H,30H,00H      ; J

DB 44H,48H,50H,60H
DB 50H,48H,44H,00H      ; K

DB 40H,40H,40H,40H
DB 40H,40H,7CH,00H      ; L

DB 44H,6CH,54H,54H
DB 44H,44H,44H,00H      ; M

DB 44H,44H,64H,54H
DB 4CH,44H,44H,00H      ; N

DB 38H,44H,44H,44H
DB 44H,44H,38H,00H      ; O

DB 78H,44H,44H,78H
DB 40H,40H,40H,00H      ; P

DB 38H,44H,44H,44H
DB 54H,48H,34H,00H      ; Q
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 78H,44H,44H,78H
DB 50H,48H,44H,00H      ; R

DB 3CH,40H,40H,38H
DB 04H,04H,78H,00H      ; S

DB 7CH,10H,10H,10H
DB 10H,10H,10H,00H      ; T

DB 44H,44H,44H,44H
DB 44H,44H,38H,00H      ; U

DB 44H,44H,44H,44H
DB 44H,28H,10H,00H      ; V

DB 44H,44H,44H,54H
DB 54H,54H,28H,00H      ; W

DB 44H,44H,28H,10H
DB 28H,44H,44H,00H      ; X

DB 44H,44H,44H,28H
DB 10H,10H,10H,00H      ; Y

DB 7CH,04H,08H,10H
DB 20H,40H,7CH,00H      ; Z

DB 38H,20H,20H,20H
DB 20H,20H,38H,00H      ; [

DB 40H,20H,10H,08H
DB 04H,02H,01H,00H      ; \

DB 38H,08H,08H,08H
DB 08H,08H,38H,00H      ; ]

DB 10H,28H,44H,00H
DB 00H,00H,00H,00H      ; ^
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 00H,00H,00H,00H
DB 00H,00H,7CH,00H      ; _

DB 20H,10H,08H,00H
DB 00H,00H,00H,00H      ;'

DB 00H,00H,38H,08H
DB 38H,48H,34H,00H      ; a

DB 40H,40H,58H,64H
DB 44H,44H,78H,00H      ; b

DB 00H,00H,3CH,40H
DB 40H,40H,3CH,00H      ; c

DB 04H,04H,34H,4CH
DB 44H,44H,3CH,00H      ; d

DB 00H,00H,38H,44H
DB 78H,40H,38H,00H      ; e

DB 18H,24H,20H,70H
DB 20H,20H,20H,00H      ; f

DB 00H,00H,38H,44H
DB 44H,3CH,04H,38H      ; g

DB 40H,40H,58H,64H
DB 44H,44H,44H,00H      ; h

DB 10H,00H,30H,10H
DB 10H,10H,38H,00H      ; i

DB 00H,08H,00H,08H
DB 08H,28H,10H,00H      ; j

DB 40H,40H,48H,50H
DB 60H,50H,48H,00H      ; k
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 30H,10H,10H,10H
DB 10H,10H,18H,00H    ; l

DB 00H,00H,68H,54H
DB 54H,54H,44H,00H    ; m

DB 00H,00H,58H,24H
DB 24H,24H,24H,00H    ; n

DB 00H,00H,38H,44H
DB 44H,44H,38H,00H    ; o

DB 00H,00H,70H,48H
DB 48H,70H,40H,40H    ; p

DB 00H,00H,38H,48H
DB 48H,38H,08H,08H    ; q

DB 00H,00H,58H,64H
DB 40H,40H,40H,00H    ; r

DB 00H,00H,38H,40H
DB 38H,04H,78H,00H    ; s

DB 20H,20H,78H,20H
DB 20H,24H,18H,00H    ; t

DB 00H,00H,48H,48H
DB 48H,48H,34H,00H    ; u

DB 00H,00H,44H,44H
DB 28H,28H,10H,00H    ; v

DB 00H,00H,44H,44H
DB 54H,54H,28H,00H    ; w

DB 00H,00H,44H,28H
DB 10H,28H,44H,00H    ; x
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 00H,00H,48H,48H
        DB 48H,38H,08H,70H      ; y

        DB 00H,00H,7CH,08H
        DB 10H,20H,7CH,00H      ; z

        DB 08H,10H,10H,20H
        DB 10H,10H,08H,00H      ; {

        DB 18H,18H,18H,18H
        DB 18H,18H,18H,00H      ; |

        DB 20H,10H,10H,08H
        DB 10H,10H,20H,00H      ; }

        DB 00H,00H,00H,32H
        DB 4CH,00H,00H,00H      ; ~

        DB 0FFH,0FFH,0FFH,0FFH
        DB 0FFH,0FFH,0FFH,0FFH


;-------------------------------------------------------;
;       DATA ที่จะนำออกมาแสดงผล                         ;
;-------------------------------------------------------;
        ORG 9200H
DTBF:   DB 44H,44H,45H,45H
        DB 20H
        DB 4DH,72H,2EH,50H
        DB 52H,41H,44H,49H
        DB 54H,20H,57H,41H
        DB 43H,48H,43H,41H
        DB 52H,11H,50H,49H
        DB 42H,4FH,4FH,4CH
        DB 20H,41H,72H,2EH
        DB 20H,56H ,49H,43H
        DB 48H,41H,49H,20H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 53H,55H, 52H,41H
        DB 50H,41H, 54H,20H
        DB 4DH,72H, 2EH,55H
        DB 41H,54H, 48H,41H
        DB 49H,20H, 53H,52H
        DB 49H,54H,45H,52H
        DB 41H,56H, 49H,52H
        DB 4FH,44H,20H,20H
        DB 4BH,4DH,49H,54H
        DB 4CH,1AH
DTBF1:  DB 50H,4FH,54H,54H
        DB 4FH,4DH,20H,53H
        DB 45H,4CH,45H,43H
        DB 54H,49H,4FH,4EH
        DB 20H
        DB 31H,3DH,4EH,55H
        DB 4DH,45H,52H,41H
        DB 4CH,20H,44H,49H
        DB 53H,50H,4CH,41H
        DB 59H,20H,32H,2CH
        DB 33H,2CH,34H,2CH
        DB 3DH,44H,41H,54H
        DB 41H,20H,44H,49H
        DB 53H,50H,4CH,41H
        DB 59H,1AH
DTBF2:  DB 52H,45H,4DH,4FH
        DB 54H,45H,20H,43H
        DB 4FH,4EH,54H,52H
        DB 4FH,4CH,20H,53H
        DB 43H,4FH,52H,45H
        DB 42H,4FH,41H,52H
        DB 44H,20H,44H,49H

        DB 53H,50H,4CH,41H
        DB 59H,20H,50H,52H
        DB 4FH,4AH ,45H,43H
        DB 54H,20H,4DH,72H
        DB 2DH,42H, 41H,4EH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 43H,48H, 4FH,43H
        DB 20H,57H, 41H,54H
        DB 54H,41H, 4EH,41H
        DB 53H,55H, 4EH,54H
        DB 48H,4FH,52H,4EH
        DB 20H,4DH, 72H,2DH
        DB 50H,41H,59H,41H
        DB 4EH,20H,42H,4FH
         DB 4EH,59H,41H,53H
        DB 55H,56H, 41H,4EH
        DB 20H,33H,4CH,1AH
DTBF3:  DB 57H,45H,4CH,43H
        DB 4FH,4DH,45H,20H
        DB 54H,4FH,20H,46H
        DB 41H,43H,55H,4CH
        DB 54H,59H,20H,45H
        DB 4EH,47H,49H,4EH
        DB 45H,45H,52H,20H
        DB 4FH,46H,20H,4BH
        DB 4DH,49H,54H,4CH
        DB 1AH
DTBF4:  DB 49H,20H ,57H,49H
        DB 53H,48H,20H,59H
        DB 4FH,55H, 20H,47H
        DB 4FH,4FH, 44H,20H
        DB 4CH,55H, 43H,4BH
        DB 59H,1AH

DTBF5:  DB 2AH,2AH
        DB 54H,48H,41H,4EH
        DB 4BH,20H,59H,4FH
        DB 55H,2AH,2AH,1AH
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. THE Z80 MICROPROCESSOR HARDWAR SOFTWARE PROGRAMMING AND INTERFACING BARRY B.BREY
2. ไมโครโปรเซสเซอร์พื้นฐาน อาจารย์อรรถสิทธิ์ หล่าสกุล
3. ไมโครโปรเซสเซอร์และการออกแบบเบื้องต้น จิติ หนูแก้ว
4. Z80 MICROCOMPUTER SYSTEM ห้องปฏิบัติการโทรคมนาคม กองการฝึกอบรม การสื่อสารแห่งประเทศไทย
5. ET DISPLAY 8*8 FOR ET BOARD VERSION 3.0 ทีม ETT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISO²-CMOS MT8870B/MT8870B-1
Integrated DTMF Receiver

9161-002-051-NA ISSUE 2 December 1987

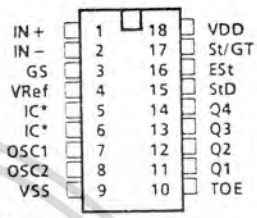
Features

- Complete DTMF Receiver
- Low Power Consumption
- Internal Gain Setting Amplifier
- Adjustable Guard Time
- Central Office Quality

Applications

- Receiver System for British Telecom (BT) or CEPT Spec (MT8870B-1)
- Paging Systems
- Repeater Systems/Mobile Radio
- Credit Card Systems
- Remote Control
- Personal Computers

Pin Connections



* Connect to VSS

Ordering Information -40°C to +85°C

MT8870BE/MT8870BE-1	Plastic DIP
MT8870BC/MT8870BC-1	Cerdip

Description

The MT8870B/MT8870B-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions, fabricated in Mitel's double poly ISO²-CMOS technology. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

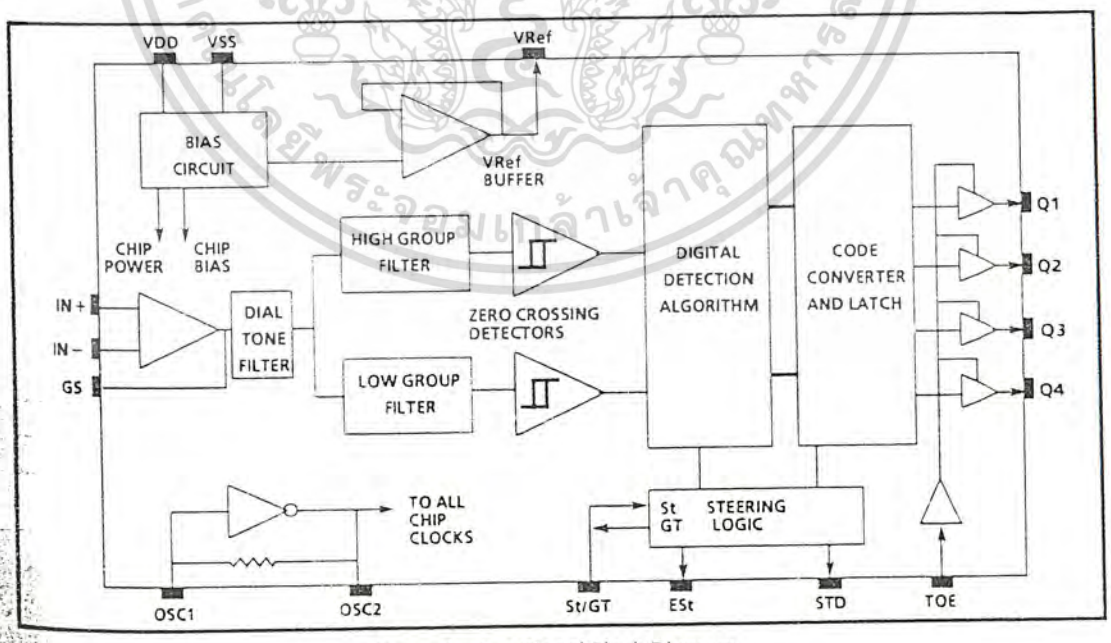


Figure 1 - Functional Block Diagram

Absolute Maximum Ratings*

	Parameter	Symbol	Min	Max	Units
1	Power supply voltage V_{DD}-V_{SS}			6	V
2	Voltage on any pin		$V_{SS}-0.3$	$V_{DD}+0.3$	V
3	Current at any pin (other than supply)			10	mA
4	Operating temperature	T_A	−40	+85	°C
5	Storage temperature		−65	+150	°C
6	Package power dissipation			1000	mW

*Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.
Derate above 75 °C at 16 mW / °C. All leads soldered to board.

Recommended Operating Conditions - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ†	Max	Units	Test Conditions
1		Positive Supply Voltages	V_{DD}		5		V	$V_{SS}=0V$
2		Oscillator Clock Frequency	fc		3.579545		MHz	
3		Oscillator Frequency Tolerance	Δfc		±0.1		%	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

DC Electrical Characteristics - $V_{DD}=5.0V\pm5\%$, $V_{SS}=0V$. Voltages are with respect to ground (V_{SS}) unless otherwise stated.

		Characteristics	Sym	Min	Typ†	Max	Units	Test Conditions
1	SUPPLY	Operating supply voltage	V_{DD}	4.75	5.0	5.25	V	
2	SUPPLY	Operating supply current	I_{DD}		3.0	9.0	mA	
3	SUPPLY	Power consumption	P_D		15	45	mW	f = 3.58 MHz; V_{DD} = 5V
4	INPUTS	High level input	V_{IH}	3.5			V	
5	INPUTS	Low level input voltage	V_{IL}			1.5	V	
6	INPUTS	Input leakage current	I_{IH}/I_{IL}		0.1		μA	$V_{IN}=V_{SS}$ or V_{DD}
7	INPUTS	Pull-up (source) current	I_{SO}		7.5	15	μA	TOE (pin 10) = 0V
8	INPUTS	Input impedance (IN+, IN−)	R_{IN}		10		MΩ	@ 1 kHz
9	INPUTS	Steering threshold voltage	V_{TSt}	2.2		2.5	V	
10	OUTPUTS	Low level output voltage	V_{OL}			$V_{SS}+0.03$	V	No load
11	OUTPUTS	High level output voltage	V_{OH}	V_{DD}-0.03			V	No load
12	OUTPUTS	Output low (sink) current	I_{OL}	1	2.5		mA	$V_{OUT}=0.4$ V
13	OUTPUTS	Output high (source) current	I_{OH}	0.4	0.8		mA	$V_{OUT}=4.6$ V
14	OUTPUTS	V_{Ref} output voltage	V_{Ref}	2.4		2.7	V	No load
15	OUTPUTS	V_{Ref} output resistance	R_{OR}		10		kΩ	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

Operating Characteristics[1] - Voltages are with respect to ground (V_{SS}) unless otherwise stated

Gain Setting Amplifier

	Characteristics	Sym	Min	Typ[2]	Max	Units	Test Conditions
1	Input leakage current	I_{IN}		100		nA	$V_{SS} \leq V_{IN} \leq V_{DD}$
2	Input resistance	R_{IN}		10		MΩ	
3	Input offset voltage	V_{OS}		25		mV	
4	Power supply rejection	PSRR		60		dB	1 kHz
5	Common mode rejection	CMRR		60		dB	$-3.0V \leq V_{IN} \leq 3.0V$
6	DC open loop voltage gain	A_{VOL}		65		dB	
7	Open loop unity gain bandwidth	f_C		1.5		MHz	
8	Output voltage swing	V_O		4.5		V_{pp}	$R_L \geq 100K\Omega$ to V_{SS}
9	Maximum capacitive load (GS)	C_L		100		pF	
10	Maximum resistive load (GS)	R_L		50		KΩ	
11	Common mode range	V_{CM}		3.0		V_{pp}	No Load

[1] $V_{DD} = 5$ V, $V_{SS} = 0$ V, $T_A = 25°C$

[2] Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

MT8870B AC Electrical Characteristics[1] - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ	Max	Units	Notes
1	SIGNAL CONDITIONS	Valid input signal levels (each tone of composite signal)		-29			dBm	1,2,3,5,6,9
				27.5			mV_{RMS}	1,2,3,5,6,9
						+1	dBm	1,2,3,5,6,9
						869	mV_{RMS}	1,2,3,5,6,9
2		Positive twist accept			10		dB	2,3,6,9
3		Negative twist accept			10		dB	2,3,6,9
4		Freq. deviation accept		±1.5% ± 2Hz			Nom.	2,3,5,9
5		Freq. deviation reject		±3.5%			Nom.	2,3,5,9
6		Third tone tolerance			-16		dB	2,3,4,5,9
7		Noise tolerance			-12		dB	2,3,4,5,7,9,10
8		Dial tone tolerance			+22		dB	2,3,4,5,8,9,11

[1] $V_{DD} = 5$ V, $V_{SS} = 0$, $T_A = 25°C$ and $f_C = 3.579545$ MHz using test circuit shown in Figure 2

NOTES

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all DTMF tones
3. Tone duration = 40 ms, tone pause = 40 ms
4. Signal condition consists of nominal DTMF frequencies
5. Both tones in composite signal have an equal amplitude.
6. Tone pair is deviated by ± 1.5% ± 2Hz.
7. Bandwidth limited (3KHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %
9. For an error rate of better than 1 in 10,000
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. For guard time calculation purposes

ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

MT8870B-1 AC Electrical Characteristics* - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ	Max	Units	Notes
1	SIGNAL CONDITIONS	Valid input signal levels (each tone of composite signal)		-31			dBm	1,2,3,5,6,9
				21.8			mV_{RMS}	1,2,3,5,6,9
						+1	dBm	1,2,3,5,6,9
						869	mV_{RMS}	1,2,3,5,6,9
2		Input Signal Level Reject		-37			dBm	1,2,3,5,6,9
				10.9			mV_{RMS}	1,2,3,5,6,9
3		Positive twist accept				6	dB	2,3,6,9
4		Negative twist accept				6	dB	2,3,6,9
5		Freq. deviation accept		±1.5% ± 2Hz				2,3,5,9
6		Freq. deviation reject		±3.5%				2,3,5,9
7		Third tone tolerance		-18.5			dB	2,3,4,5,9,13
8		Noise tolerance			-12		dB	2,3,4,5,7,9,10
9		Dial tone tolerance			+22		dB	2,3,4,5,8,9,11

* V_{DD} = 5 V, V_{SS} = 0 , T_A = 25° C and f_c = 3.579545 MHz using test circuit shown in Figure 2

NOTES

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all DTMF tones
3. Tone duration = 40 ms, tone pause = 40 ms.
4. Signal condition consists of nominal DTMF frequencies
5. Both tones in composite signal have an equal amplitude.
6. Tone pair is deviated by ±1.5% ± 2Hz.
7. Bandwidth limited (3 kHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %.
9. For an error rate of better than 1 in 10,000.
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. For guard time calculation purposes.
13. Referenced to Fig. 10 Input DTMF Tone Level at -25 dBm (-28 dBm at GS Pin) Interference Frequency Range between 480-3400 Hz.

AC Electrical Characteristics - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ‡	Max	Units	Conditions
1	TIMING	Tone present detect time	t_{DP}	5	11	14	ms	Note 12
2		Tone absent detect time	t_{DA}	0.5	4	8.5	ms	Note 12
3		Tone duration accept	t_{REC}			40	ms	User adjustable
4		Tone duration reject	$t_{\overline{REC}}$	20			ms	User adjustable
5		Interdigit pause accept	t_{ID}			40	ms	User adjustable
6		Interdigit pause reject	t_{DO}	20			ms	User adjustable
7	OUTPUTS	Propagation delay (St to Q)	t_{PQ}		8	11	μs	TOE = V_{DD}
8		Propagation delay (St to StD)	t_{PStD}		12		μs	TOE = V_{DD}
9		Output data setup (Q to StD)	t_{QStD}		3.4		μs	TOE = V_{DD}
10		Propagation delay (TOE to Q ENABLE)	t_{PTE}		50		ns	R_L = 10kΩ C_L = 50 pF
11		Propagation delay (TOE to Q DISABLE)	t_{PTD}		300		ns	R_L = 10kΩ C_L = 50 pF
12	CLOCK	Crystal /clock frequency	f_C	3.5759	3.5795	3.5831	MHz	
13		Clock input rise time	t_{LHCL}			110	ns	Ext. clock
14		Clock input fall time	t_{HLCL}			110	ns	Ext. clock
15		Clock input duty cycle	DC_{CL}	40	50	60	%	Ext. clock
16		Capacitive load (OSC2)	C_{LO}			30	pF	

† V_{DD} = 5.0V, V_{SS} = 0V, T_A = 25°C and f_c = 3.579545 MHz, using test circuit shown in Figure 2.

‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

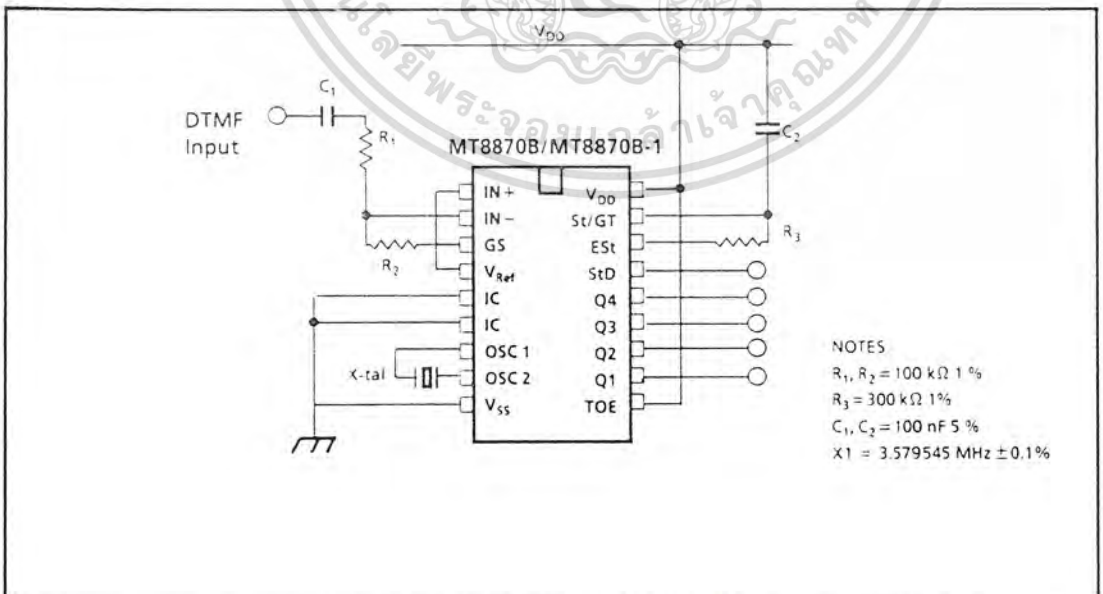
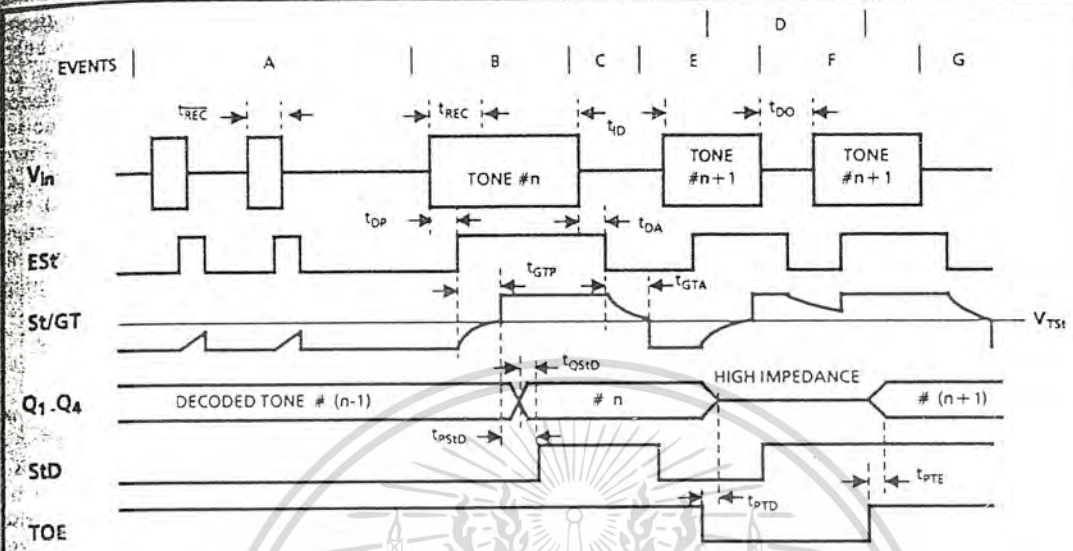


Figure 2 - Single-Ended Input Configuration

Pin Description

Pin #	Name	Description
1	IN+	Non-Inverting Op-Amp (Input).
2	IN−	Inverting Op-Amp (Input).
3	GS	Gain Select. Gives access to output of front end differential amplifier for connection of feedback resistor.
4	V_{Ref}	Reference Voltage (Output). Nominally $V_{DD}/2$ is used to bias inputs at mid-rail (see Fig.2).
5	IC	Internal Connection. Must be tied to V_{SS}.
6	IC	Internal Connection. Must be tied to V_{SS}.
7	OSC1	Clock (Input).
8	OSC2	Clock (Output). A 3.579545 MHz crystal connected between pins OSC1 and OSC2 completes the internal oscillator circuit.
9	V_{SS}	Negative Power Supply (Input).
10	TOE	Three State Output Enable (Input). Logic high enables the outputs Q1-Q4. This pin is pulled up internally.
11-14	Q1-Q4	Three State Data (Output). When enabled by TOE, provide the code corresponding to the last valid tone-pair received (see Table 1). When TOE is logic low, the data outputs are high impedance.
15	StD	Delayed Steering (Output). Presents a logic high when a received tone-pair has been registered and the output latch updated; returns to logic low when the voltage on St/GT falls below V_{TSt}.
16	ESt	Early Steering (Output). Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause ESt to return to a logic low.
17	St/GT	Steering Input/Guard time (Output) Bidirectional. A voltage greater than V_{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{TSt} frees the device to accept a new tone pair. The GT output acts to reset the external steering time-constant; its state is a function of ESt and the voltage on St.
18	V_{DD}	Positive power supply (Input).

สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



EXPLANATION OF EVENTS

A) TONE BURSTS DETECTED, TONE DURATION INVALID, OUTPUTS NOT UPDATED.
B) TONE #n DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS.
C) END OF TONE #n DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.
D) OUTPUTS SWITCHED TO HIGH IMPEDANCE STATE.
E) TONE #n+1 DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS (CURRENTLY HIGH IMPEDANCE).
F) ACCEPTABLE DROPOUT OF TONE #n+1, TONE ABSENT DURATION INVALID, OUTPUTS REMAIN LATCHED.
G) END OF TONE #n+1 DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.

EXPLANATION OF SYMBOLS

V_{in} DTMF COMPOSITE INPUT SIGNAL.
ESt EARLY STEERING OUTPUT. INDICATES DETECTION OF VALID TONE FREQUENCIES.
St/GT STEERING INPUT/GUARD TIME OUTPUT. DRIVES EXTERNAL RC TIMING CIRCUIT.
Q_1-Q_4 4-BIT DECODED TONE OUTPUT.
StD DELAYED STEERING OUTPUT. INDICATES THAT VALID FREQUENCIES HAVE BEEN PRESENT/ABSENT FOR THE REQUIRED GUARD TIME THUS CONSTITUTING A VALID SIGNAL.
TOE TONE OUTPUT ENABLE (INPUT). A LOW LEVEL SHIFTS Q_1-Q_4 TO ITS HIGH IMPEDANCE STATE.

$\overline{t_{REC}}$ MAXIMUM DTMF SIGNAL DURATION NOT DETECTED AS VALID.
t_{REC} MINIMUM DTMF SIGNAL DURATION REQUIRED FOR VALID RECOGNITION.
t_{ID} MINIMUM TIME BETWEEN VALID DTMF SIGNALS.
t_{DO} MAXIMUM ALLOWABLE DROPOUT DURING VALID DTMF SIGNAL.
t_{DP} TIME TO DETECT THE PRESENCE OF VALID DTMF SIGNALS.
t_{DA} TIME TO DETECT THE ABSENCE OF VALID DTMF SIGNALS.
t_{GTP} GUARD TIME, TONE PRESENT
t_{GTA} GUARD TIME, TONE ABSENT

Figure 3- Timing Diagram

Functional Description

The MT8870B/MT8870B-1 monolithic DTMF receiver offers small size, low power consumption and high performance. Its architecture consists of a bandsplit filter section, which separates the high and low group tones, followed by a digital counting section which verifies the frequency and duration of the received tones before passing the corresponding code to the output bus.

Filter Section

Separation of the low group and high group tones is achieved by applying the DTMF signal to the inputs of two sixth-order switched capacitor bandpass filters, the bandwidths of which correspond to the low and high group frequencies. The filter section also incorporates notches at 350 and 440 Hz for exceptional dial tone rejection (see Figure 4). Each filter output is followed by a single order switched capacitor filter section which smooths the signals prior to limiting. Limiting is performed by high-gain comparators which are provided with hysteresis to prevent detection of unwanted low-level signals. The outputs of the comparators provide full rail logic swings at the frequencies of the incoming DTMF signals.

Decoder Section

Following the filter section is a decoder employing digital counting techniques to determine the frequencies of the incoming tones and to verify that they correspond to standard DTMF frequencies. A complex averaging algorithm protects against tone simulation by extraneous signals such as voice while providing tolerance to small frequency deviations and variations. This averaging algorithm has been developed to ensure an optimum combination of immunity to talk-off and tolerance to the presence of interfering frequencies (third tones) and noise. When the detector recognizes the presence of two valid tones (this is referred to as the "signal condition" in some industry specifications) the "Early Steering" (ESt) output will go to an active state. Any subsequent loss of signal condition will cause ESt to assume an inactive state (see "Steering Circuit").

Steering Circuit

Before registration of a decoded tone pair, the receiver checks for a valid signal duration (referred to as character recognition condition). This check is performed by an external RC time constant driven by ESt. A logic high on ESt causes v_c (see Figure 5) to rise as the capacitor discharges. Provided signal condition is maintained (ESt remains high) for the validation period (t_{GTP}), v_c reaches the threshold (V_{TSt}) of the steering logic to register the tone pair, latching its corresponding 4-bit code (see Table 1) into the output latch. At this point the GT output is activated and drives v_c to V_{DD}. GT continues to drive high as long as ESt remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (StD) goes high, signalling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the three state control input (TOE) to a logic high. The steering circuit works in reverse to validate the interdigit

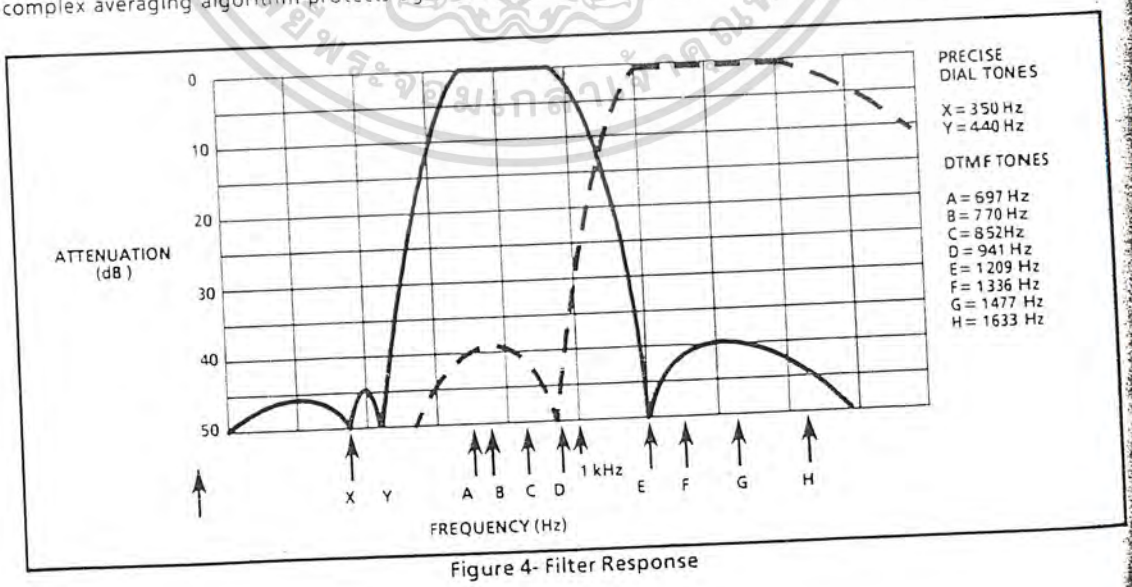


Figure 4- Filter Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้าน

F_{LOW}	F_{HIGH}	NO.	TOE	Q_4	Q_3	Q_2	Q_1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

L = LOGIC LOW, H = LOGIC HIGH, Z = HIGH IMPEDANCE

Table 1. Functional Decode Table

pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (dropout) too short to be considered a valid pause. This facility, together with the capability of selecting the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

Guard Time Adjustment

In many situations not requiring selection of tone duration and interdigital pause, the simple steering circuit shown in Figure 5 is applicable. Component values are chosen according to the formula:

$$t_{REC} = t_{DP} + t_{GTP}$$
$$t_{ID} = t_{DA} + t_{GTA}$$

The value of t_{DP} is a device parameter (see Figure 3) and t_{REC} is the minimum signal duration to be recognized by the receiver. A value for C of 0.1 μF is recommended for most applications, leaving R to be selected by the designer.

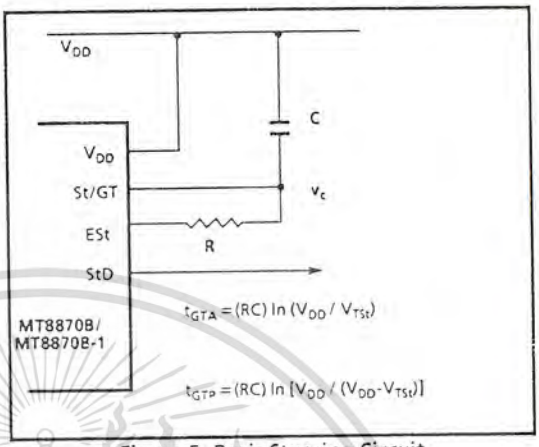


Figure 5- Basic Steering Circuit

Different steering arrangements may be used to select independently the guard times for tone present (t_{GTP}) and tone absent (t_{GTA}). This may be necessary to meet system specifications which place both accept and reject limits on both tone duration and interdigital pause. Guard time adjustment also allows the designer to tailor system parameters such as talk off and noise immunity. Increasing t_{REC} improves talk-off performance since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. Alternatively, a relatively short t_{REC} with a long t_{DO} would be appropriate for extremely noisy environments where fast acquisition time and immunity to tone dropouts are required. Design information for guard time adjustment is shown in Figure 6.

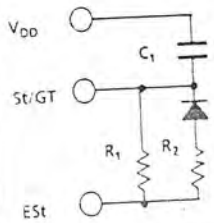
Differential Input Configuration

The input arrangement of the MT8870B/MT8870B-1 provides a differential-input operational amplifier as well as a bias source(V_{Ref}) which is used to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the op-amp output (GS) for adjustment of gain. In a single-ended configuration, the input pins are connected as shown in Figure 2 with the op-amp connected for unity gain and V_{Ref} biasing the input at $\frac{1}{2}V_{DD}$. Figure 7 shows the differential configuration, which permits the adjustment of gain with the feedback resistor R_5.

$t_{GTP} = (R_P C_1) \ln [V_{DD} / (V_{DD} - V_{TSt})]$

$t_{GTA} = (R_1 C_1) \ln (V_{DD} / V_{TSt})$

$R_P = (R_1 R_2) / (R_1 + R_2)$

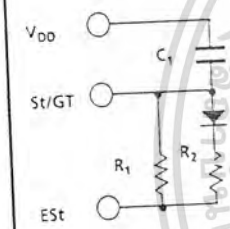


a) decreasing t_{GTP}. ($t_{GTP} < t_{GTA}$)

$t_{GTP} = (R_1 C_1) \ln [V_{DD} / (V_{DD} - V_{TSt})]$

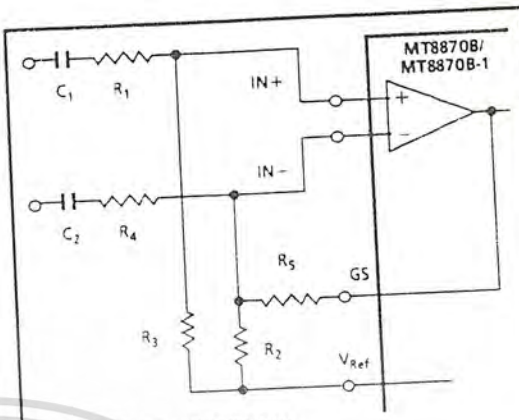
$t_{GTA} = (R_P C_1) \ln (V_{DD} / V_{TSt})$

$R_P = (R_1 R_2) / (R_1 + R_2)$



b) decreasing t_{GTA}. ($t_{GTP} > t_{GTA}$)

Figure 6- Guard Time Adjustment



DIFFERENTIAL INPUT AMPLIFIER

$C_1 = C_2 = 10$ nF

$R_1 = R_4 = R_5 = 100$ kΩ

$R_2 = 60$ kΩ, $R_3 = 37.5$ kΩ

$R_3 = \frac{R_2 R_5}{R_2 + R_5}$

VOLTAGE GAIN (A_V diff) = $\frac{R_5}{R_1}$

INPUT IMPEDANCE

$(Z_{INDIFF}) = 2\sqrt{R_1^2 + \left(\frac{1}{\omega C}\right)^2}$

All resistors are ±1 % tolerance.
All capacitors are ± 5 % tolerance.

Figure 7- Differential Input Configuration

Crystal Oscillator

The internal clock circuit is completed with the addition of an external 3.579545 MHz crystal and is normally connected as shown in Figure 2 (Single Ended Input Configuration). However, it is possible to configure several MT8870B/MT8870B-1 devices employing only a single oscillator crystal. The oscillator output of the first device in the chain is coupled through a 30 pF capacitor to the oscillator input (OSC1) of the next device. Subsequent devices are connected in a similar fashion. Refer to Figure 8 for details. The problems associated with unbalanced loading are not a concern with the arrangement shown, ie; precision balancing capacitors are not required.

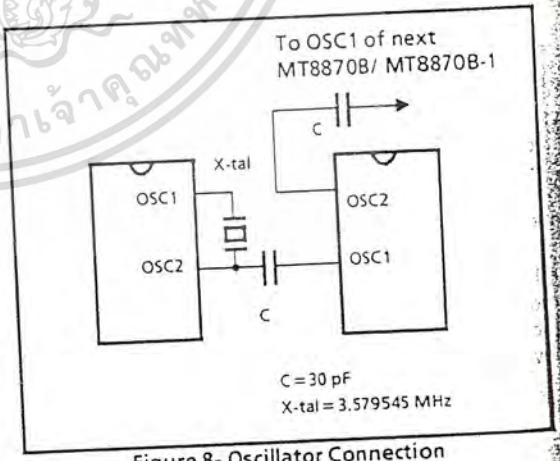


Figure 8- Oscillator Connection

ISO²-CMOS MT8870C/MT8870C-1
Integrated DTMF Receiver

9161-002-094-NA ISSUE 1 November 1990

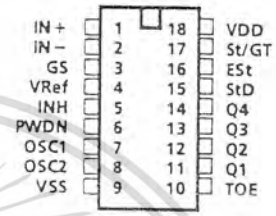
Features

- Complete DTMF Receiver
- Low Power Consumption
- Internal Gain Setting Amplifier
- Adjustable Guard Time
- Central Office Quality
- Power-down Mode
- Inhibit Mode

Applications

- Receiver System for British Telecom (BT) or CEPT Spec (MT8870C-1)
- Paging Systems
- Repeater Systems/Mobile Radio
- Credit Card Systems
- Remote Control
- Personal Computers
- Telephone Answering Machine

Pin Connections



Ordering Information

MT8870CE/MT8870CE-1	Plastic DIP
MT8870CC/MT8870CC-1	Cerdip
MT8870CS	SOIC
−40 °C to +85 °C	

Description

The MT8870C/MT8870C-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions, fabricated in Mitel's double poly ISO²-CMOS technology. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone-pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

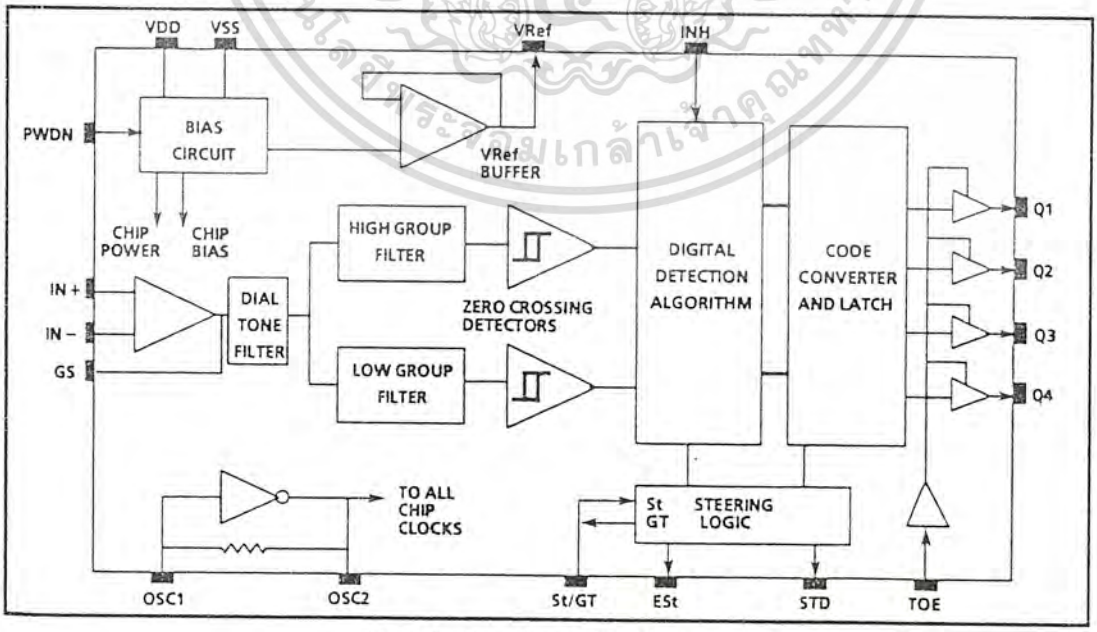


Figure 1 - Functional Block Diagram

APPLICATION

RECEIVER SYSTEM FOR BRITISH TELECOM SPEC POR 1151

The circuit shown in Fig. 10 illustrates the use of MT8870B-1 device in a typical receiver system. BT Spec defines the input signals less than -34 dBm as the non-operate level. This condition can be attained by choosing a suitable values of R_1 and R_2 to provide 3 dB attenuation, such that -34 dBm input signal will correspond to -37 dBm at the gain setting pin GS of MT8870B-1. As shown in the diagram, the component values of R_3 and C_2 are the guard time requirements when the total component tolerance is 6%. For better performance, it is recommended to use the non-symmetric guard time circuit in Fig. 9.

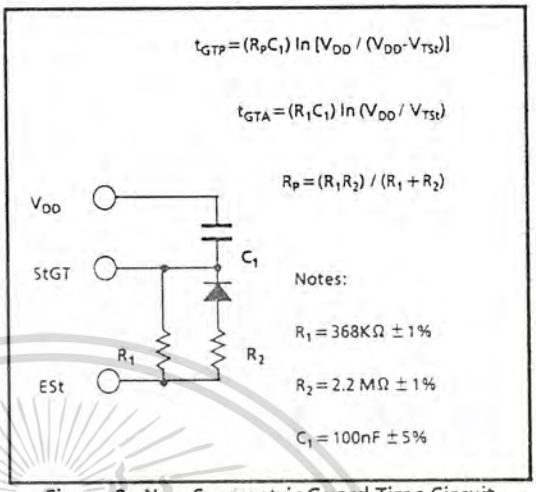


Figure 9 - Non-Symmetric Guard Time Circuit

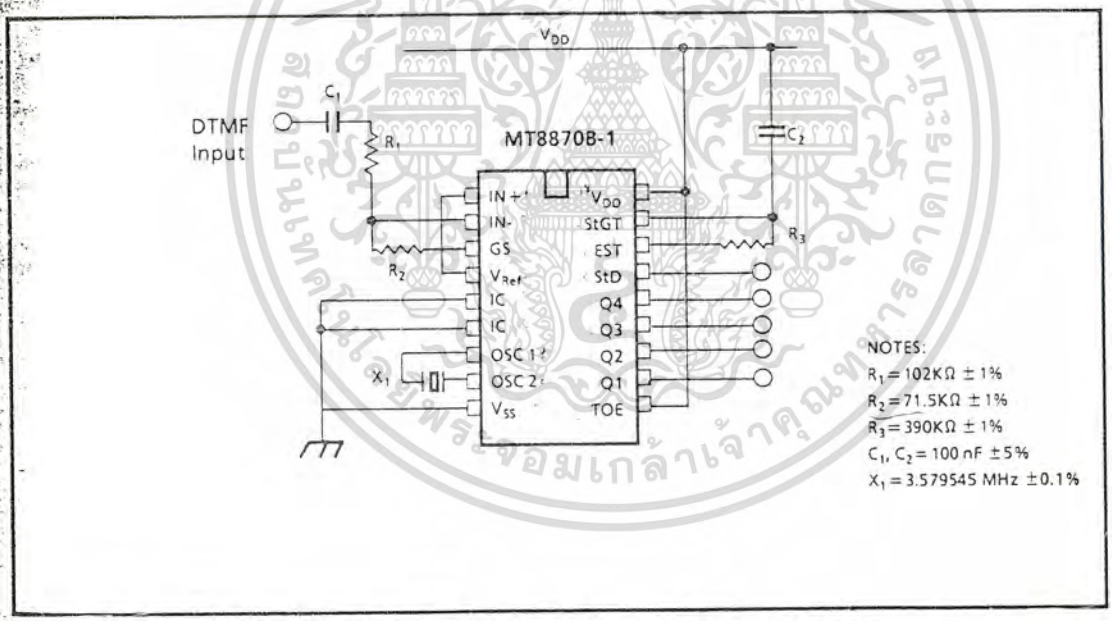


Figure 10 - Single-Ended Input Configuration for BT or CEPT Spec

UMC

UM82C55A

CMOS Programmable Peripheral Interface

Features

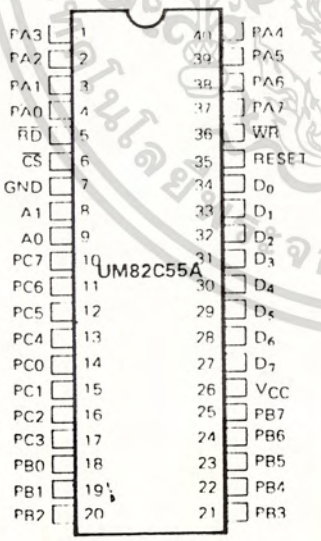
- Pin compatible with NMOS 8255A
- 24 programmable I/O pins
- Fully TTL compatible
- Bus hold circuitry on all I/O ports eliminates pull up resistors
- High speed, no "wait state" operation with 8MHz 80C86
- Direct bit set/reset capability
- Enhanced control word read capability
- Single 5V power supply
- 2.5mA drive capability on all I/O port outputs
- Low standby power – I_{CCSB} = 10μA

General Description

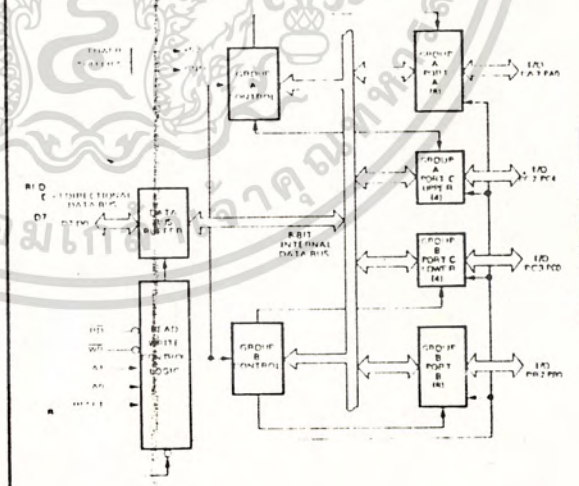
The UM82C55A is a high performance CMOS version of the industry standard 8255A and is manufactured using a selfaligned silicon gate CMOS process. It is a general purpose programmable I/O device which may be used with many different microprocessors. There are 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The high performance of the UM82C55A make it compatible with microprocessors such as the 8086, 8048, 8051.

Static CMOS circuit design insures low operating power. TTL compatibility of V_{IH} = 2.0 volts over the industrial temperature range and bus hold circuitry eliminate the need for pull up resistors.

Pin Configuration



Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Supply Voltage	+8.0 VOLTS
Operating Voltage Range	+4V to +7V
Input Voltage Applied	GND -2.0V to 6.5V
I/O Pin Voltage Applied	GND-0.5V to VCC+0.5V
Storage Temperature Range	-65°C to +150°C
Operating Temperature Range	0°C to +70°C
Maximum Power Dissipation	1 Watt

*Comments

Stresses above those listed in the "ABSOLUTE MAXIMUM RATINGS" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. Electrical Characteristics

($VCC = 5.0V+/-5\%$, $T_A = 0°C$ to $+70°C$)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IH}	Logical One Input Voltage	2.0		V	
V_{IL}	Logical Zero Input Voltage		0.8	V	
V_{OH}	Logical One Output Voltage	3.0 VCC-0.4		V V	$I_{OH} = -2.5mA$ $I_{OH} = -100\ \mu A$
V_{OL}	Logical Zero Output Voltage		0.4	V	$I_{OL} = +2.5\ mA$
I_{IL}	Input Leakage Current	-1.0	1.0	μA	$0V \leq V_{IN} \leq V_{CC}$
I_O	I/O Pin Leakage Current	-10.0	10.0	μA	$0V \leq V_O \leq V_{CC}$
I_{BHH}	Bus Hold High Leakage Current	-50	-300	μA	$V_O = 3.0V$ Ports A, B, C
I_{BHL}	Bus Hold Low Leakage Current	+50	+300	μA	$V_O = 1.0V$ Port A only
I_{DAR}	Darlington Drive Current	-2.0		mA	Ports A, B, C Test Condition 3
I_{CC}	Power Supply Current		10	μA	$V_{CC} = 5.5V$ $V_{IN} = V_{CC}$ or GND Outputs Open

Capacitance

($T_A = 25°C$; $V_{CC} = GND = 0V$; $V_{IN} = +5V$ or GND)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
C_{IN}*	Input Capacitance		5	pF	FREQ = 1 MHZ Unmeasured Pins Returned to GND
$C_{I/O}$*	I/O Pin Capacitance		20	pF	

*Guaranteed and sampled, but not 100% tested

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Characteristics

(V_{CC} = +5V ±5%, GND = 0V; T_A = 0°C to +70°C)

READ

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AR}	Address Stable Before READ	0		ns	
t_{RA}	Address Stable After READ	0		ns	
t_{RR}	READ Pulse Width	150		ns	
t_{RD}	Data Valid From READ		100	ns	1
t_{DF}	Data Float After READ	10	75	ns	2
t_{RV}	Time Between READs and/or WRITEs	300		ns	

WRITE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Stable Before WRITE	0		ns	
t_{WA}	Address Stable After WRITE	20		ns	Ports A & B
		60		ns	Port C
t_{WW}	WRITE Pulse Width	100		ns	
t_{DW}	Data Valid to WRITE High	100		ns	
t_{WD}	Data Valid After WRITE High	30		ns	Ports A & B
		60		ns	Port C

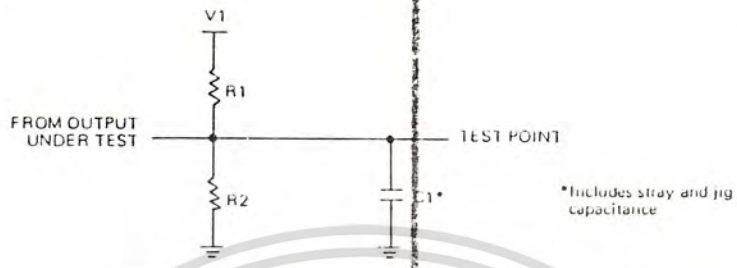
OTHER TIMINGS

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{WB}	WR = 1 to Output		350	ns	1
t_{IR}	Peripheral Data Before RD	0		ns	
t_{HR}	Peripheral Data After RD	0		ns	
t_{AK}	ACK Pulse Width	100		ns	
t_{ST}	STB Pulse Width	100		ns	
t_{PS}	Per. Data Before STB High	20		ns	
t_{PH}	Per. Data After STB High	50		ns	
t_{AD}	ACK = 0 to Output		175	ns	1
t_{KD}	ACK = 1 to Output Float	20	250	ns	2
t_{WOB}	WR = 1 to OBF = 0		150	ns	1
t_{AOB}	ACK = 0 to OBF = 1		150	ns	1
t_{SIB}	STB = 0 to IBF = 1		150	ns	1
t_{RIB}	RD = 1 to IBF = 0		150	ns	1
t_{RIT}	RD = 0 to INTR = 0		200	ns	1
t_{SIT}	STB = 1 to INTR = 1		150	ns	1
t_{AIT}	ACK = 1 to INTR = 1		150	ns	1
t_{WIT}	WR = 0 to INTR = 0		200	ns	1
t_{RES}	Reset Pulse Width	500		ns	see note 1

Note: Period of initial Reset pulse after power on must be at least 50usec. Subsequent Reset pulses may be 500 ns minimum.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. Test Circuits

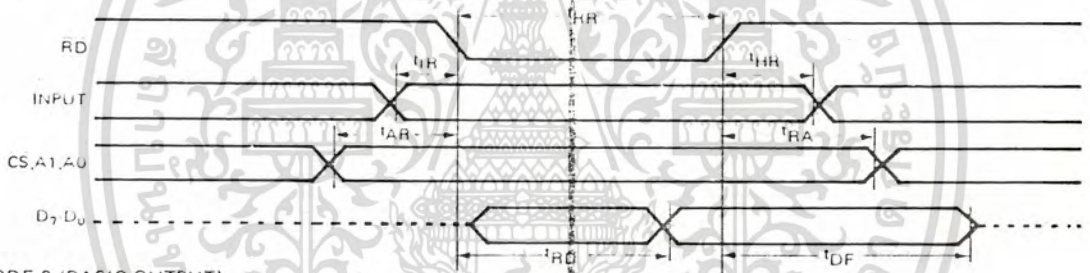


Test Condition	V1	R1	R2	C1
1	1.7V	523Ω	Open	150 pf
2	5.0V	2kΩ	1.7kΩ	50 pf
3	1.5V	750Ω	Open	Open

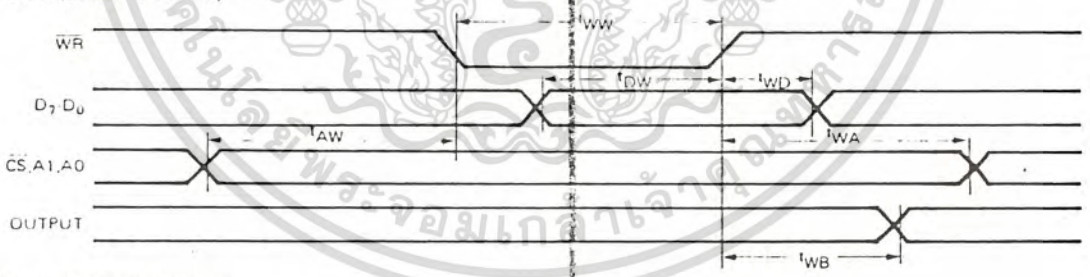
TEST CONDITION DEFINITION TABLE

Waveforms

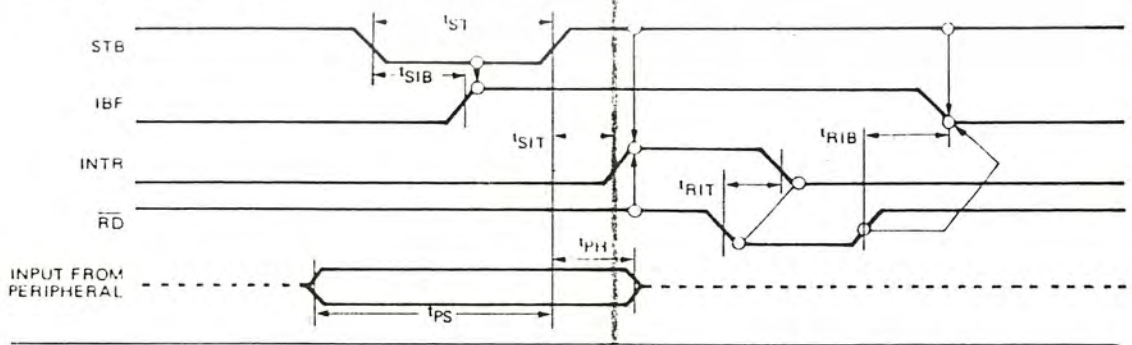
MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)

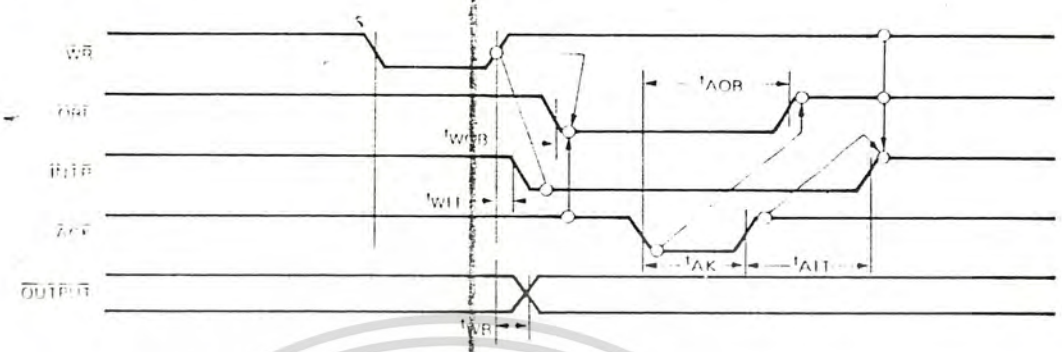


MODE 1 (STROBED INPUT)

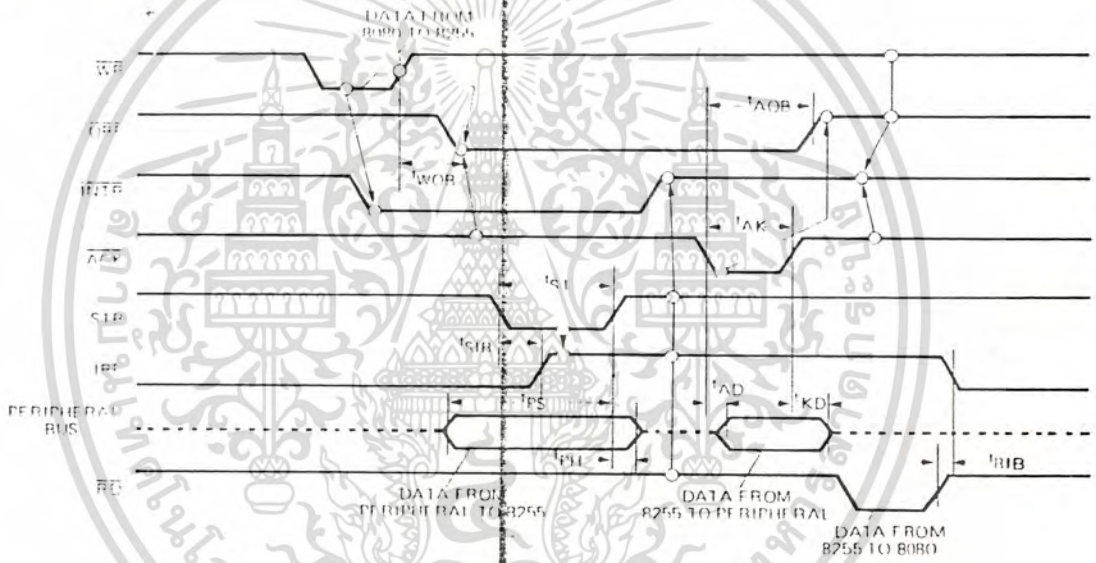


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 1 (Strobe Output)

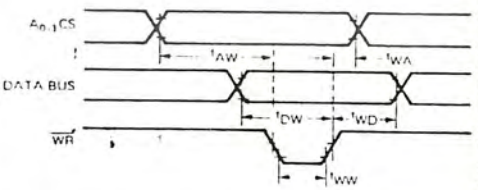


MODE 2 (BIDIRECTIONAL)

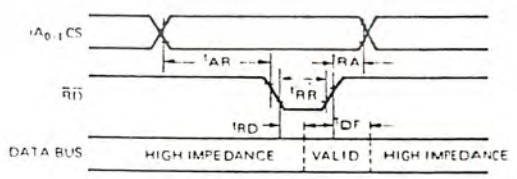


Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible. (INTR = IBF · $\overline{MASK}$ · $\overline{STB}$ · $\overline{RD}$ + $\overline{OBF}$ · $\overline{MASK}$ · $\overline{ACK}$ · $\overline{WR}$)

WRITE TIMING



READ TIMING



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the UM82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

(CS)

Chip Select. A "low" on this input pin enables the communication between the UM82C55A and the CPU.

(RD)

Read. A "low" on this input pin enables the UM82C55A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the UM82C55A.

(WR)

Write. A "low" on this input pin enables the CPU to write data or control words into the UM82C55A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

UM82C55A BASIC OPERATION

A	A_2	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (Read)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
x	x	x	x	1	Data Bus → 3 State
x	x	1	1	0	Data Bus → 3 State

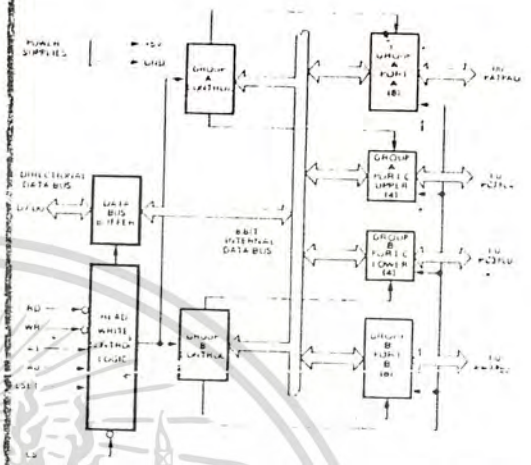


Figure 1. UM82C55A Block Diagram Data Bus Buffer and Read/Write Control Logic Functions

(Reset)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode. "Bus hold" devices internal to the UM82C55A will hold the I/O port inputs to a logic "1" state with a maximum hold current of 300 μA.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the UM82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the UM82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)
Control Group B – Port B and Port C lower (C3-C0)

The control word register can be both written and read as shown in the "Basic Operation" table. Figure 4 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B and C

The UM82C55A contains three 8 bit ports (A, B, and C). All can be configured to a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the UM82C55A.

Port A One 8 bit data output latch/buffer and one 8-bit data input latch. Both "pull up" and "pull-down" bus-hold devices are present on Port A.

General Purpose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port B One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input)x This port can be divided into two 4-bit ports under the mode control. Each 4 bit port contains a 4 bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

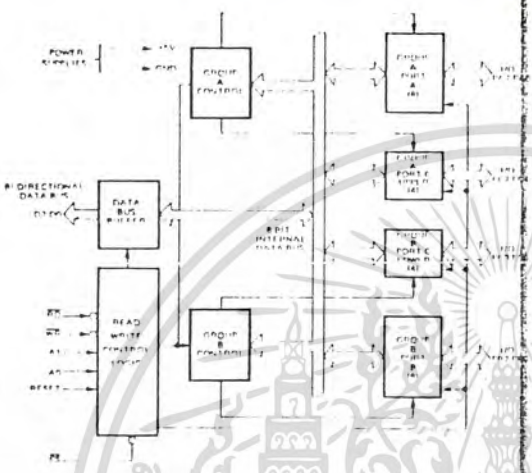


Figure 2. UM82C55A Block Diagram Showing Group A and Group B Control Functions

Operational Description

Mode Selection

There are three basic modes of operation that can be selected by the system software.

Mode 0 – Basic Input/Output
Mode 1 – Strobed Input/Output
Mode 2 – Bi-Directional Bus

When the reset input goes "high", all ports will be set to the input mode with all 24 port lines held at a logic "one" level by internal bus hold devices. After the reset is removed, the UM82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown resistors in all-CMOS designs. During the execution of the system program, any of the other modes may be selected using a single output instruction. This allows a single UM82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance, Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt driven basis.

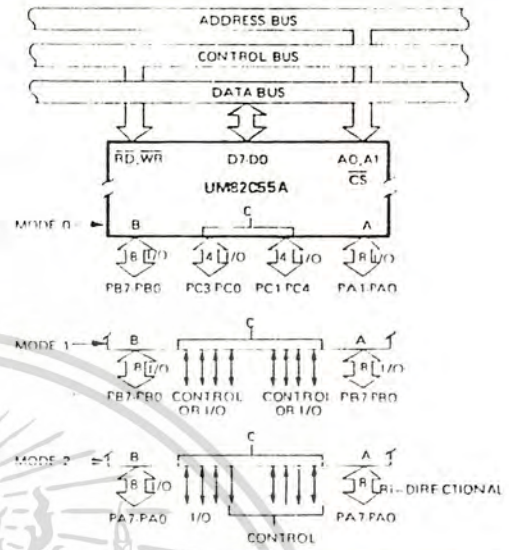


Figure 3. Basic Mode Definitions and Bus Interface

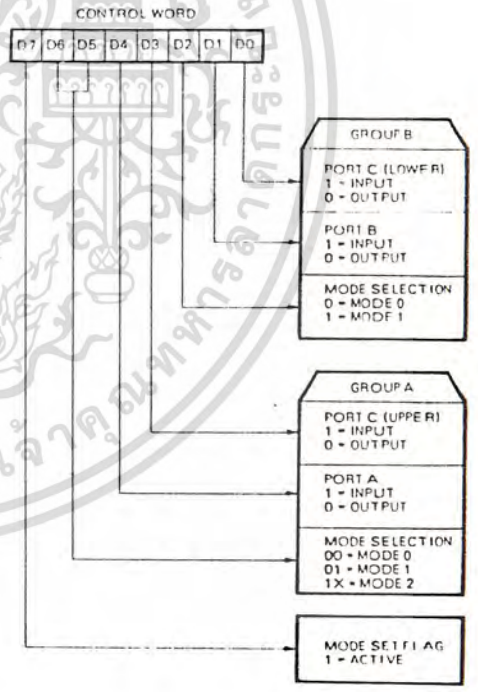


Figure 4. Mode Definition Format

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the UM82C55A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

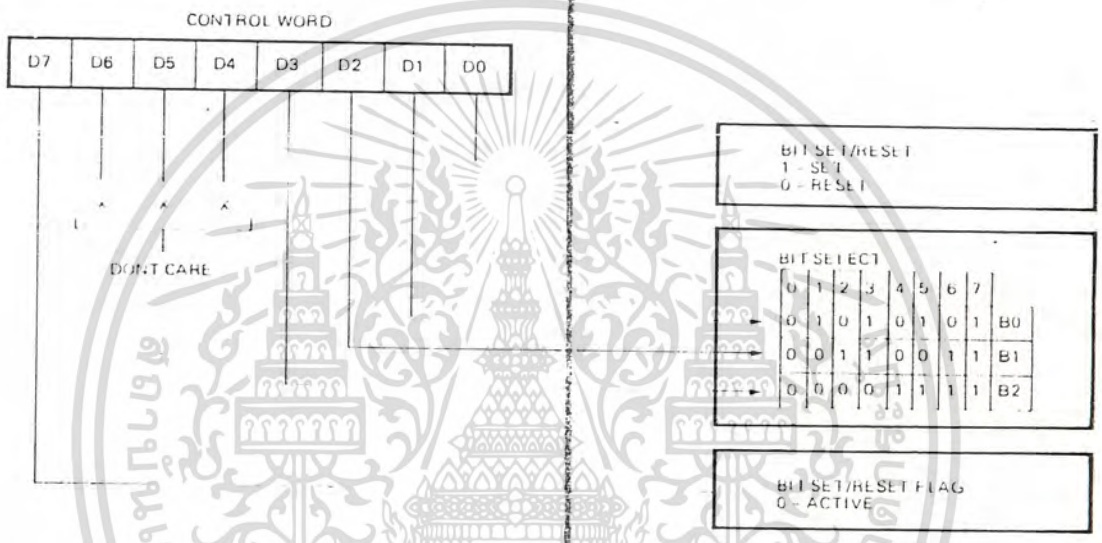


Figure 5. Bit Set/Reset Format

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUT put instruction. This feature reduces software requirements in control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the UM82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the programmer to enable or disable a CPU interrupt by a specific I/O device without affecting any other device in the interrupt structure.

INTE Flip-flop Definition

(BIT SET) - INTE is SET - Interrupt enable.
(BIT - RESET) - INTE is RESET - Interrupt disable.

Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specific port.

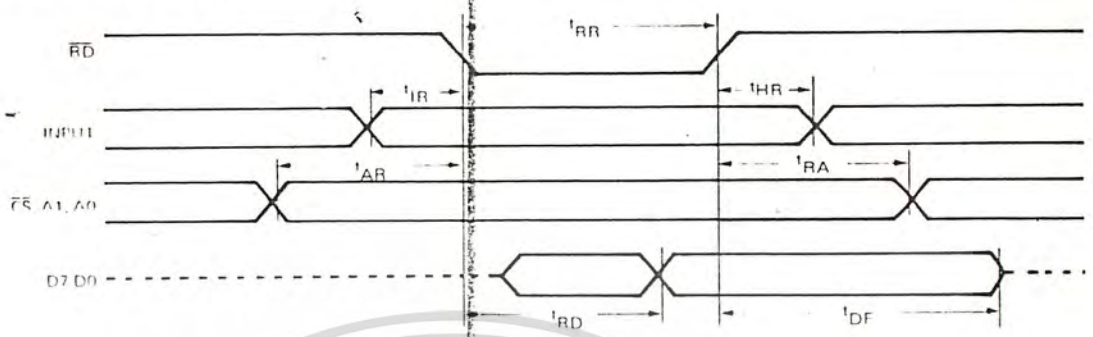
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports
- Any port can be input or output
- Outputs are latched
- Inputs are not latched
- 16 different Input/Output configurations possible.

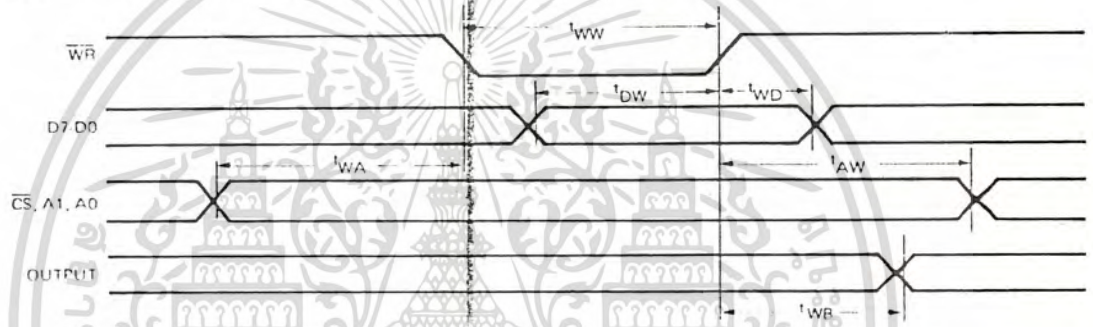
General Purpose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)



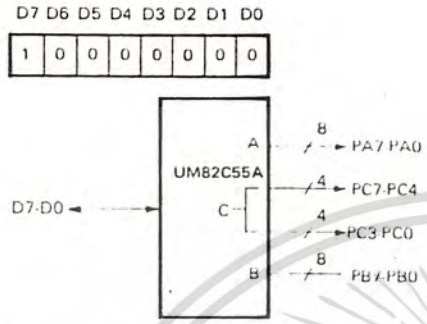
Mode 0 Port Definition

A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	Output	Output	0	Output	Output
0	0	0	1	Output	Output	1	Output	Input
0	0	1	0	Output	Output	2	Input	Output
0	0	1	1	Output	Output	3	Input	Input
0	1	0	0	Output	Input	4	Output	Output
0	1	0	1	Output	Input	5	Output	Input
0	1	1	0	Output	Input	6	Input	Output
0	1	1	1	Output	Input	7	Input	Input
1	0	0	0	Input	Output	8	Output	Output
1	0	0	1	Input	Output	9	Output	Input
1	0	1	0	Input	Output	10	Input	Output
1	0	1	1	Input	Output	11	Input	Input
1	1	0	0	Input	Input	12	Output	Output
1	1	0	1	Input	Input	13	Output	Input
1	1	1	0	Input	Input	14	Input	Output
1	1	1	1	Input	Input	15	Input	Input

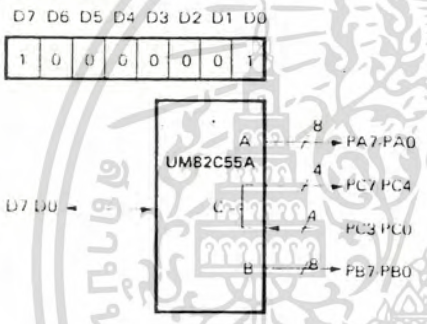
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode 0 Configurations

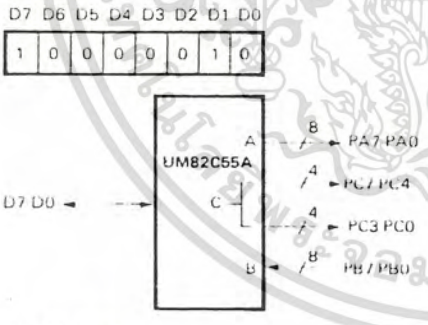
CONTROL WORD #0



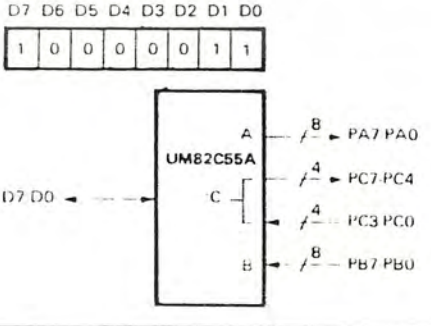
CONTROL WORD #1



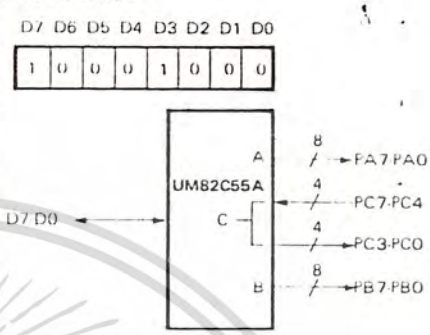
CONTROL WORD #2



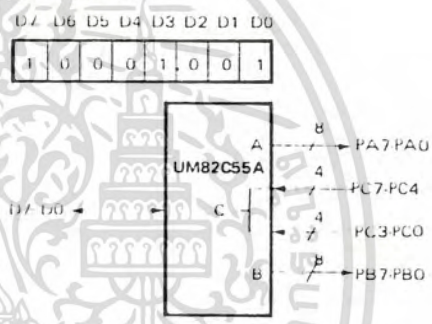
CONTROL WORD #3



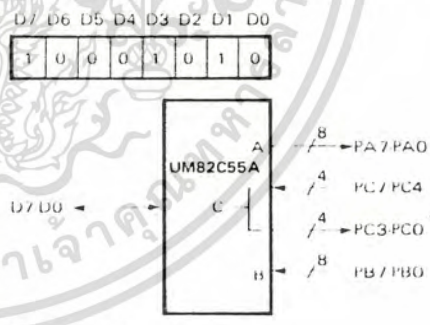
CONTROL WORD #4



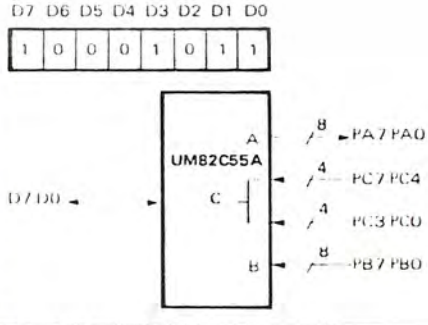
CONTROL WORD #5



CONTROL WORD #6



CONTROL WORD #7

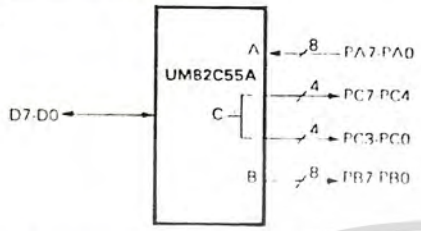


General Purpose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

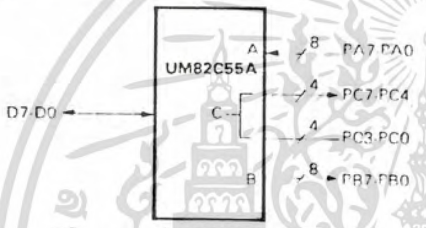
CONTROL WORD #8

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	0



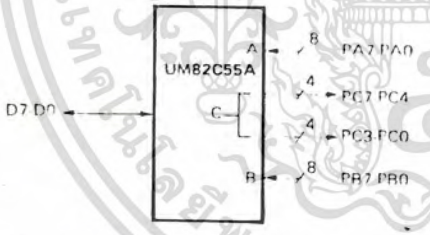
CONTROL WORD #9

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	1



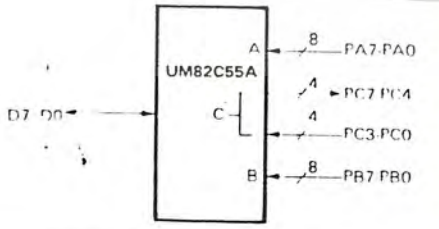
CONTROL WORD #10

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



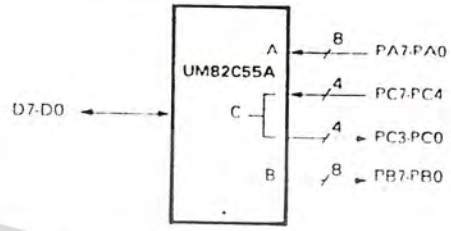
CONTROL WORD #11

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1



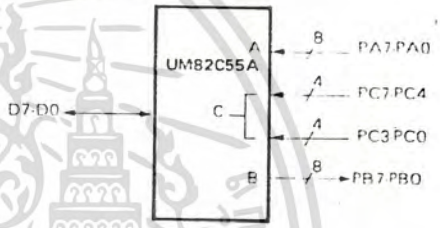
CONTROL WORD #12

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0



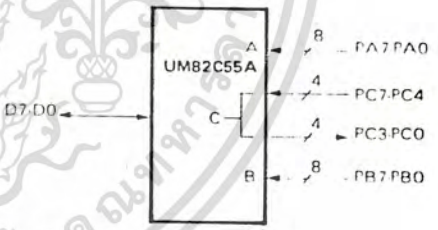
CONTROL WORD #13

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	1



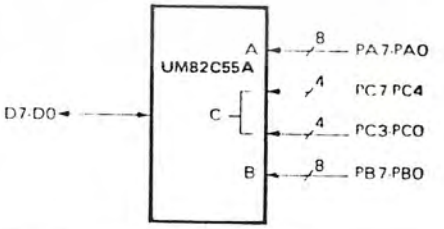
CONTROL WORD #14

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	0



CONTROL WORD #15

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operating Modes

Mode 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions

- Two Groups (Group A and Group B)
- Each group contains one 8-bit port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit port.

Input Control Signal Definition

$\overline{STB}$ (Strobe Input)
A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch, in essence, an acknowledgement. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the condition: $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

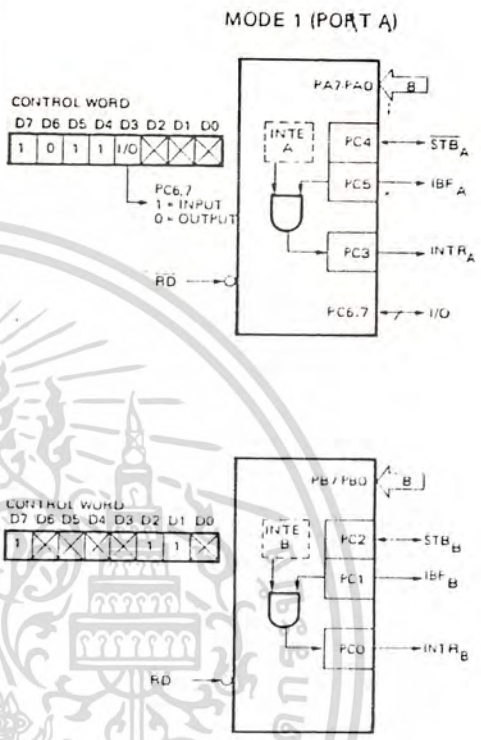


Figure 6. MODE 1 Input

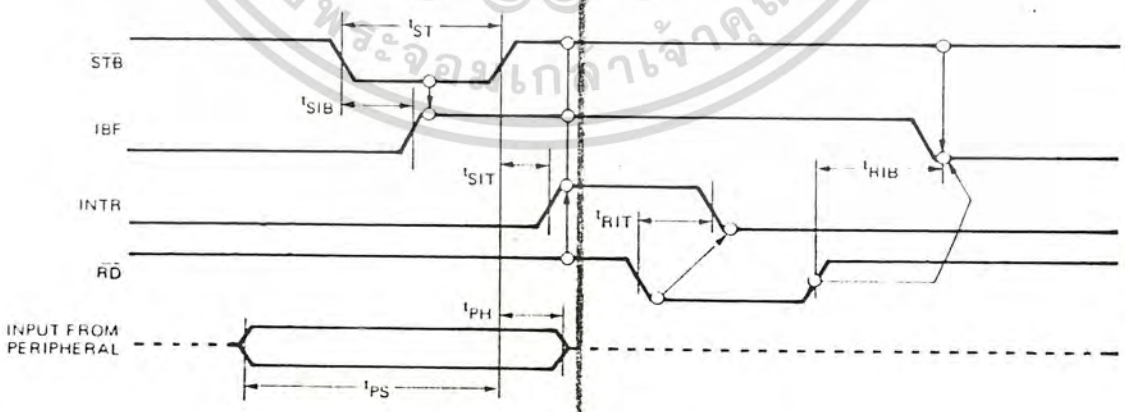


Figure 7. MODE 1 (Strobed Input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the UM82C55A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the $\overline{CPU}$.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the $\overline{CPU}$ when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by Bit Set/Reset of PC_6.

INTE B

Controlled by Bit Set/Reset of PC_2.

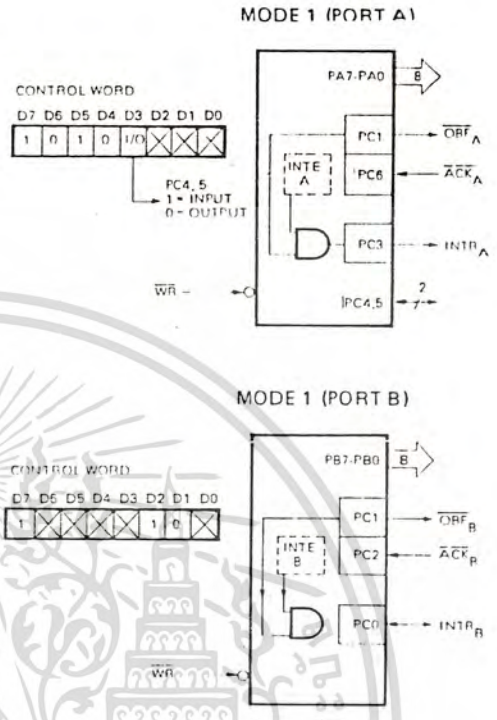


Figure 8. MODE 1 Output

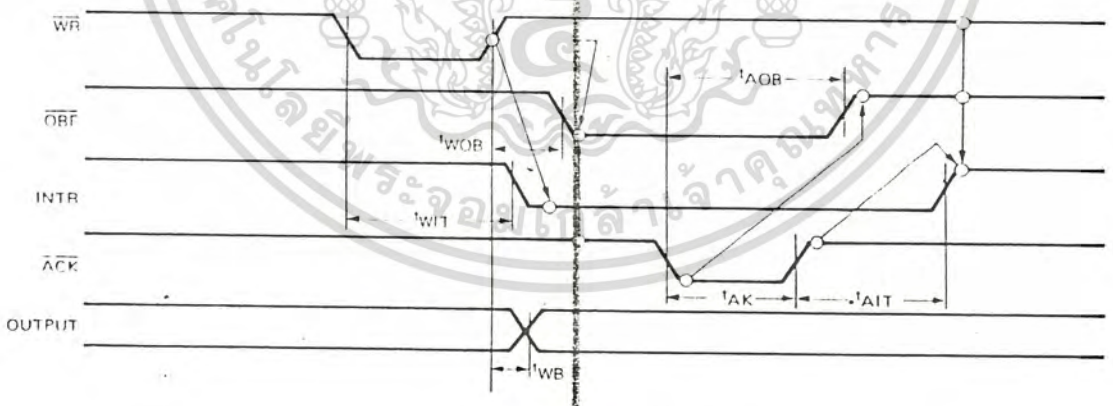


Figure 9. MODE 1 (Strobed Output)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Combinations of MODE 1, Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

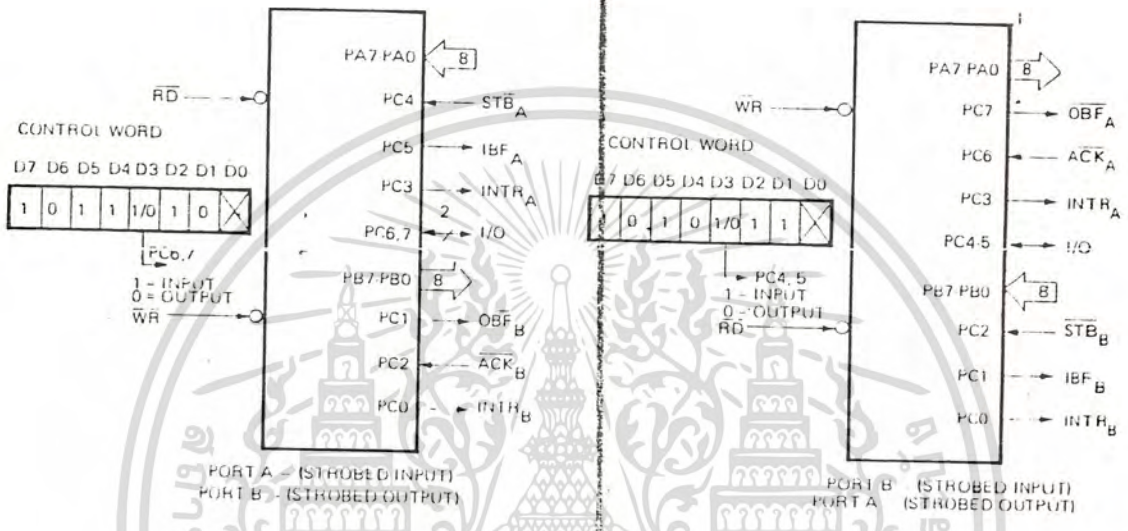


Figure 10. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O)

The functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline similar to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Function Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with $\overline{OBF}$). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

General Purpose Peripheral IC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

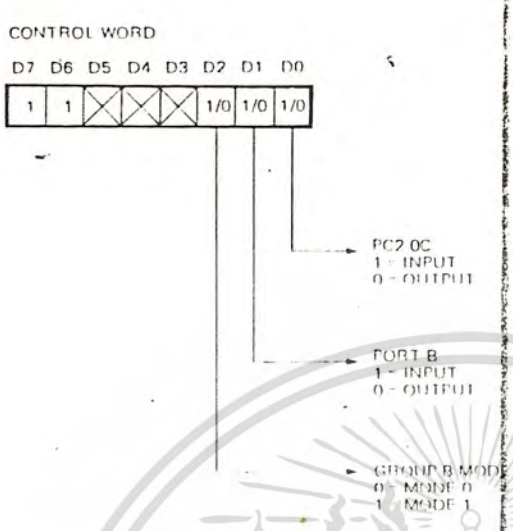


Figure 11. MODE Control Word

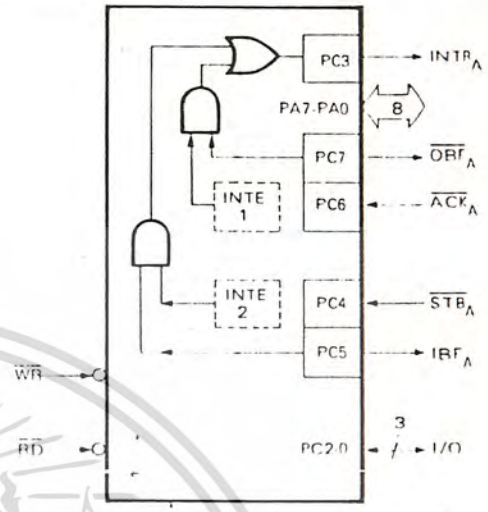


Figure 12. MODE 2

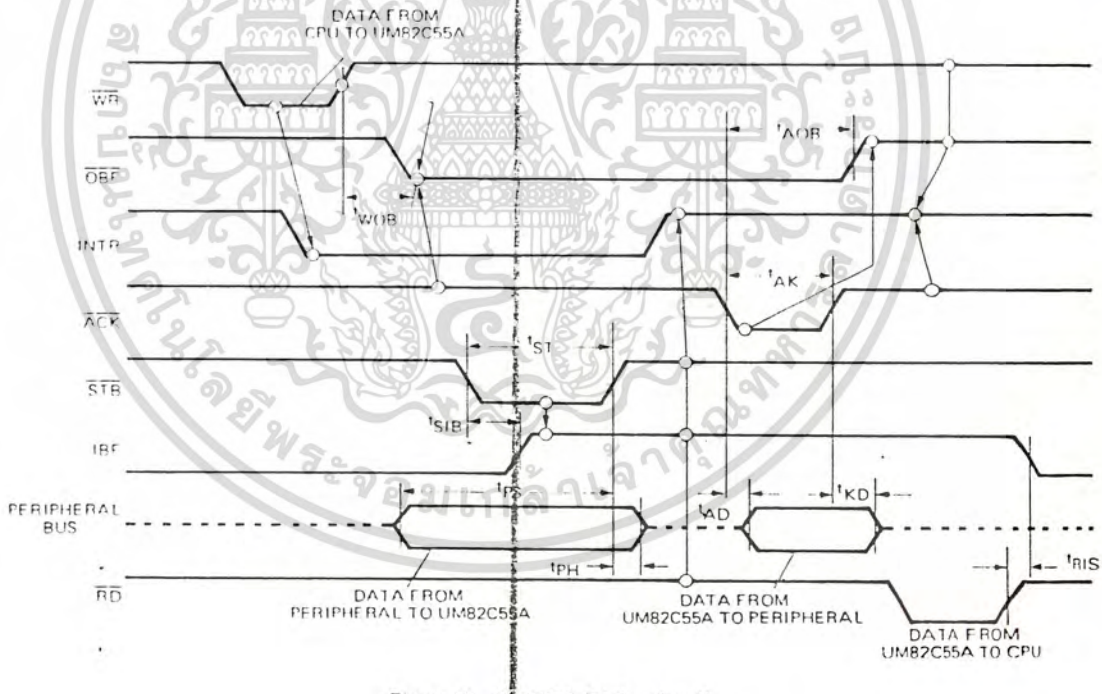
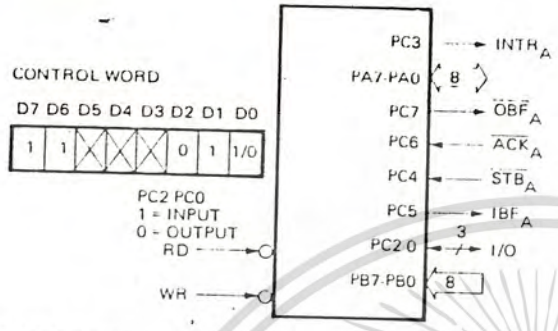


Figure 13. MODE 2 (Bidirectional)

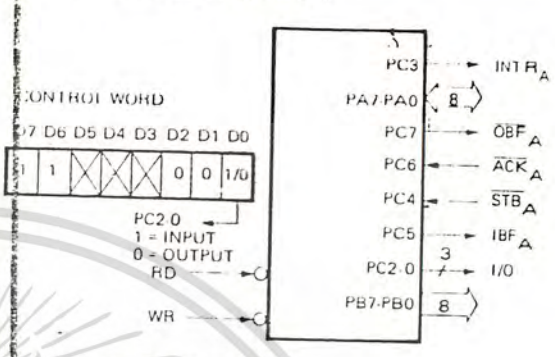
Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before RD is permissible. (INTR = IBF · MASK · $\overline{STB}$ · RD + $\overline{OBF}$ · MASK · $\overline{ACK}$ · $\overline{WR}$)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

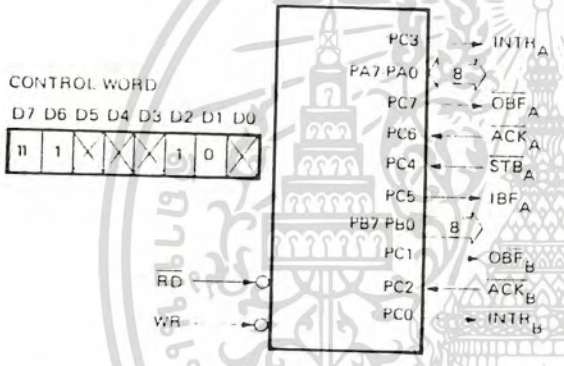
MODE 2 AND MODE 0 (INPUT)



MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)

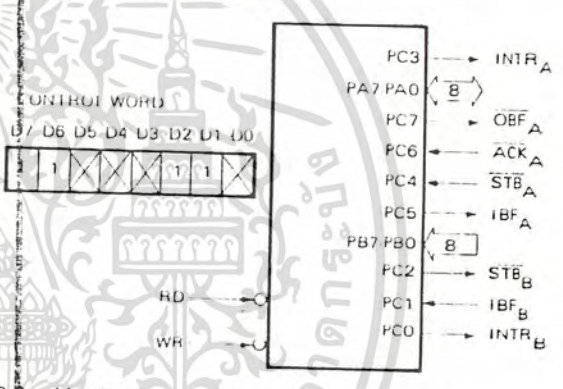


Figure 14. MODE 2 Combinations

Mode Definition Summary

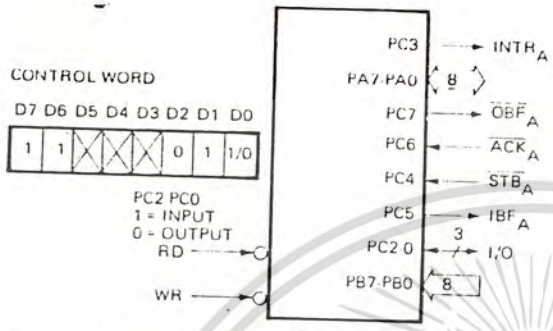
	MODE 0		MODE 1		MODE 2
	IN	OUT	IN	OUT	Group A Only
PA_0	IN	OUT	IN	OUT	
PA_1	IN	OUT	IN	OUT	
PA_2	IN	OUT	IN	OUT	
PA_3	IN	OUT	IN	OUT	
PA_4	IN	OUT	IN	OUT	
PA_5	IN	OUT	IN	OUT	
PA_6	IN	OUT	IN	OUT	
PA_7	IN	OUT	IN	OUT	
PB_0	IN	OUT	IN	OUT	
PB_1	IN	OUT	IN	OUT	
PB_2	IN	OUT	IN	OUT	
PB_3	IN	OUT	IN	OUT	
PB_4	IN	OUT	IN	OUT	
PB_5	IN	OUT	IN	OUT	
PB_6	IN	OUT	IN	OUT	
PB_7	IN	OUT	IN	OUT	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O
PC_3	IN	OUT	$INTR_A$	$INTR_A$	INTRA
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$
PC_5	IN	OUT	IBF_A	I/O	IBF_A
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$

MODE 0 OR MODE 1 ONLY

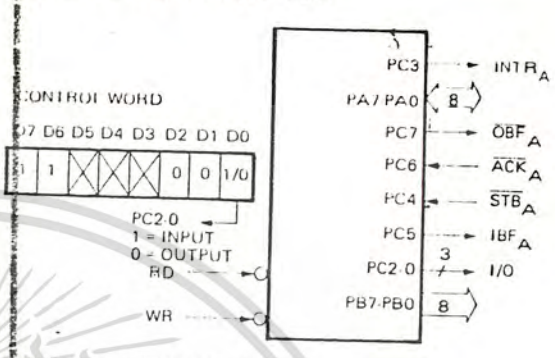
General Purpose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

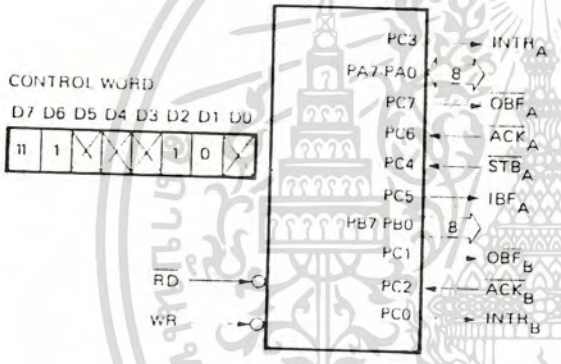
MODE 2 AND MODE 0 (INPUT)



MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)

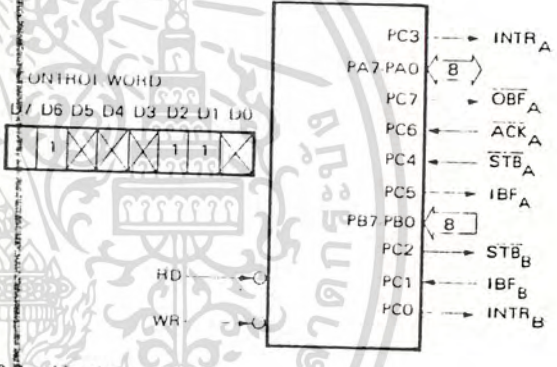


Figure 14. MODE 2 Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2 Group A Only
	IN	OUT	IN	OUT	
PA_0	IN	OUT	IN	OUT	
PA_1	IN	OUT	IN	OUT	
PA_2	IN	OUT	IN	OUT	
PA_3	IN	OUT	IN	OUT	
PA_4	IN	OUT	IN	OUT	
PA_5	IN	OUT	IN	OUT	
PA_6	IN	OUT	IN	OUT	
PA_7	IN	OUT	IN	OUT	
PB_0	IN	OUT	IN	OUT	
PB_1	IN	OUT	IN	OUT	
PB_2	IN	OUT	IN	OUT	
PB_3	IN	OUT	IN	OUT	
PB_4	IN	OUT	IN	OUT	
PB_5	IN	OUT	IN	OUT	
PB_6	IN	OUT	IN	OUT	
PB_7	IN	OUT	IN	OUT	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O
PC_3	IN	OUT	$INTR_A$	$INTR_A$	INTRA
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$
PC_5	IN	OUT	IBF_A	I/O	IBF_A
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$

MODE 0 OR MODE 1 ONLY

General Purpose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Mode Combination Considerations:

There are several combinations of modes possible. For any combination, some or all of Port Clines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port Clines, except the $\overline{ACK}$ and $\overline{STB}$ lines, will be placed on the data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 17.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and $\overline{OBF}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{ACK}$ and $\overline{STB}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 17.

INPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
		GROUP A			GROUP B		

OUTPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
		GROUP A			GROUP B		

Figure 15. MODE 1 Status Word Format

D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$			
		GROUP A			GROUP B		

(DEFINED BY MODE 0 OR MODE 1 SELECTION)

Figure 16. MODE 2 Status Word Format

Interrupt Enable Flag*	Position	Alternate Port C Pin Signal (Mode)
INTE B	PC2	$\overline{ACK}_B$ (Output Mode 1) or $\overline{STB}_B$ (Input Mode 1)
INTE A2	PC4	$\overline{STB}_A$ (Input Mode 1 or Mode 2)
INTE A1	PC6	$\overline{ACK}_A$ (Output Mode 1 or Mode 2)

Figure 17. Interrupt Enable Flags in Modes 1 and 2

Current Drive Capability:

Any output on Port A, B or C can sink or source 2.5mA. This feature allows the UM82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the UM82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device.. Reading the contents of Port C allows that programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applications of the UM82C55A

The UM82C55A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the UM82C55A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the UM82C55A to exactly "fit" the application. Figures 10 through 24 present a few examples of typical applications of the UM82C55A.

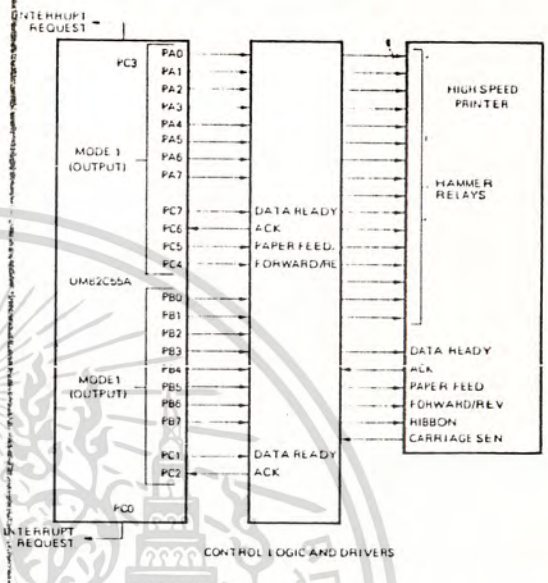


Figure 18. Printer Interface

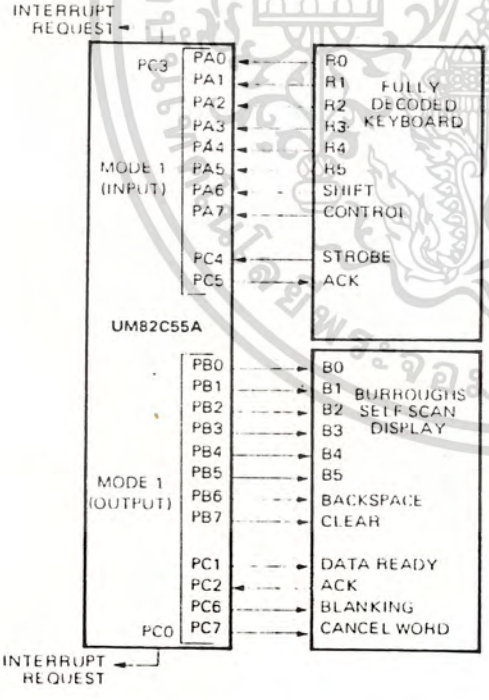


Figure 19. Keyboard and Display Interface

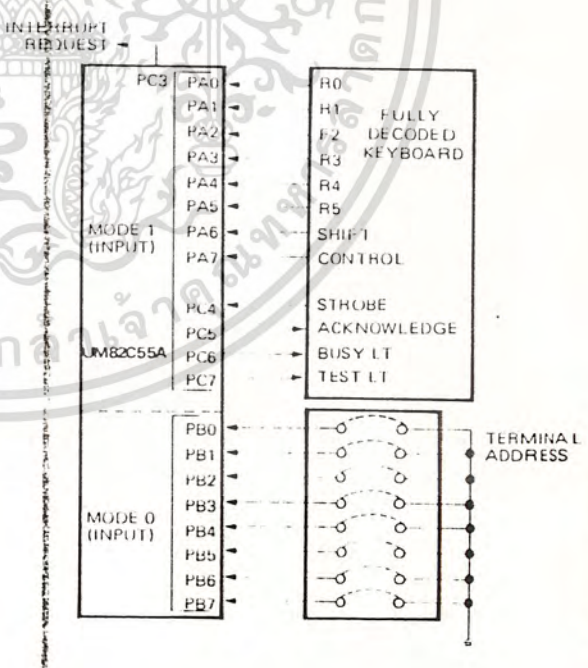


Figure 20. Keyboard and Terminal Address Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

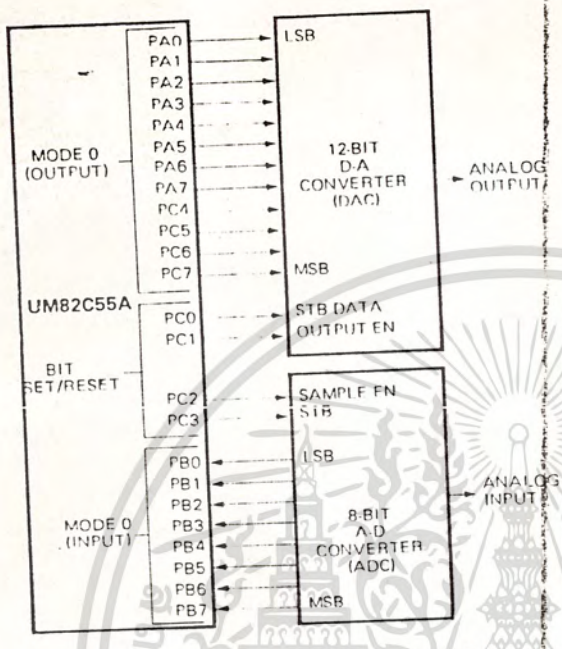


Figure 21. Digital to Analog, Analog to Digital

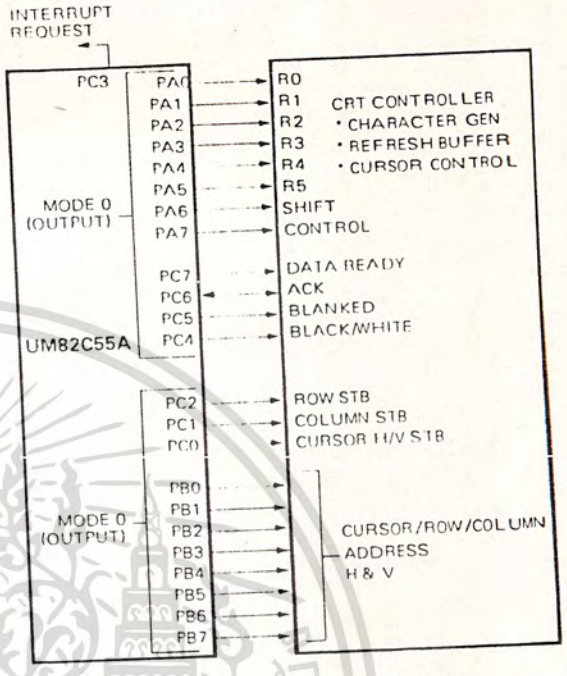


Figure 22. Basic CRT Controller Interface

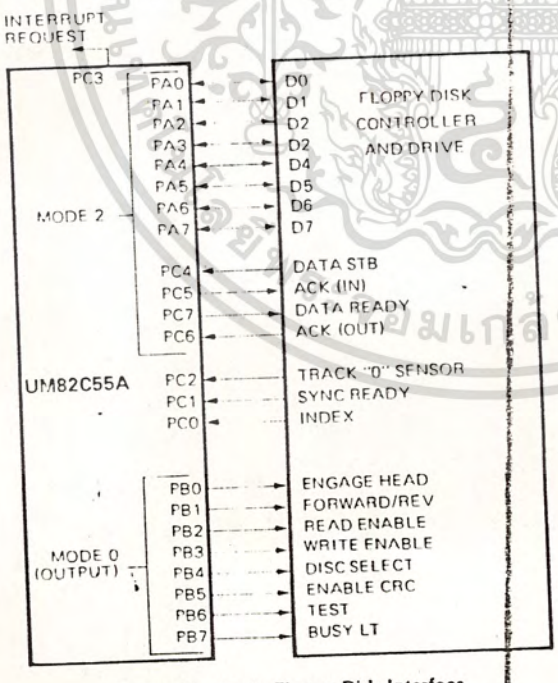


Figure 23. Basic Floppy Disk Interface

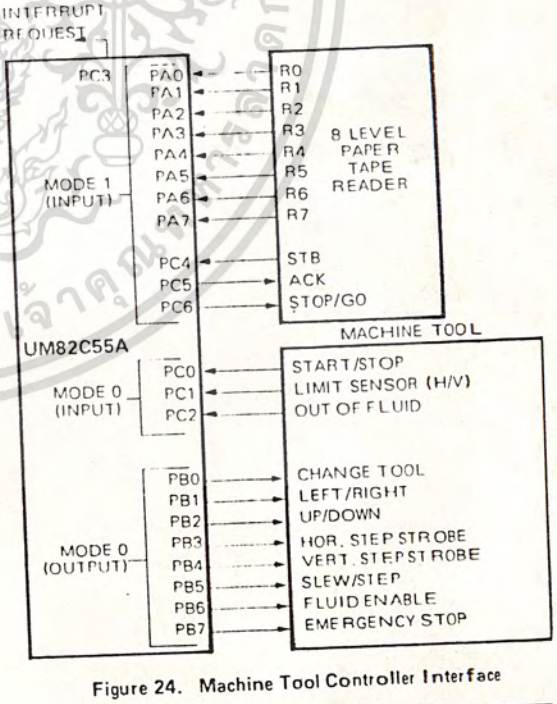
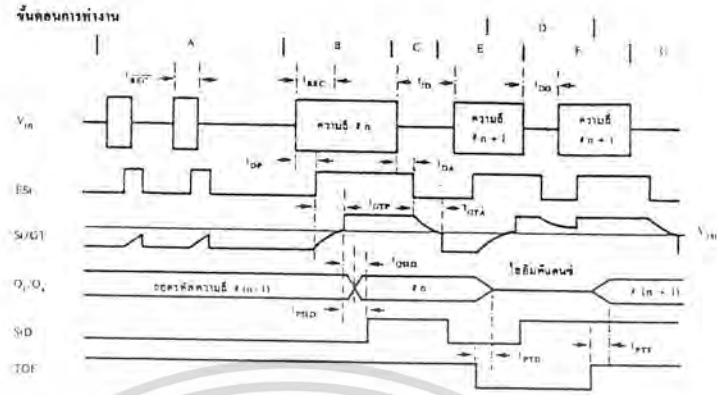


Figure 24. Machine Tool Controller Interface

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



อธิบายขั้นตอนการทำงาน

A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาต์พุตไม่เปลี่ยน

B - ความถี่ # n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแลตช์ไว้ที่เอาต์พุต

C - จบความถี่ # n ช่วงห่างถูกต้อง เอาต์พุตยังคงแลตช์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D - เอาต์พุตเปลี่ยนเป็นไฮอิมพีแดนซ์

E - ความถี่ # n + 1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแลตช์ไว้

F - ความถี่ # n + 1 หายไป ช่วงห่างไม่ถูกต้อง เอาต์พุตยังคงแลตช์อยู่

C - จบความถี่ # n + 1 ช่วงห่างถูกต้อง เอาต์พุตยังแลตช์อยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

V_{in} - สัญญาณความถี่ DTMF ที่เข้ามา

ESt - Early Sterring output ใช้แสดงความถี่ที่ถูกต้อง

St/GT - Steering input/Guard Time output สำหรับต่อกับ RC ภายนอก

Q_1-Q_4 - เอาต์พุต BCD ขนาด 4 บิต

StD - Delayed Steering output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE - Tone Output Enable (input) ใช้ควบคุม Q_1-Q_4 ให้เป็นไฮอิมพีแดนซ์

$t_{\overline{REC}}$ - คาบเวลานานสุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง

t_{REC} - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

t_{ID} - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

t_{DO} - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{GTP} - การ์ดไทม์ของการปรากฏความถี่ DTMF

t_{GTA} - การ์ดไทม์ของการหายไปของความถี่ DTMF

รูปแสดง TIMEMING DIAGRAM การทำงานของ MT8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดควบคุมหรือ CONTROLER BOARD

SPECIFICATION

CPU	:	Z80B CMOS
MEMORY	:	ROM 2764,27256
	:	RAM 6264
PORT	:	82C55 (PLCC 44 PIN)
CLOCK RATE	:	4 MHz
POWER SSUPPLY	:	+ 5 VDC
CONNECTOR	:	40 PIN EXPANSION HEADER-STRIP
	:	34 PIN EXPANSION HEADER-STRIP
LED	:	POWER RED LED
	:	HALT GREEN LED

หลักการทำงาน

เป็นวงจรควบคุมการทำงานทั้งหมดของระบบทุกๆ ส่วนมีไมโครโปรเซสเซอร์ เป็นตัวควบคุม และกำหนดตามโปรแกรม โดยใช้ Z80B เป็น CPU ต่อร่วมกับ ROM เบอร์ 2764 เพื่อที่จะทำเป็นโปรแกรมมอนิเตอร์ และใช้ RAM เบอร์ 6264 เพื่อเป็นหน่วยความจำในการเขียนหรืออ่าน และเป็น สแต็ก

ในการอ้างแอดเดรสหน่วยความจำ ROM และ RAM จะใช้ไอซี 74LS139เป็นตัว DECODER หน่วยความจำโดยเราใช้แอดเดรส A14 และเป็น 4 ช่วงแอดเดรส ช่วงละ 16 K ไบต์ แต่ ROM เราต้องการ 32 K ไบต์ จึงต้องใช้ไดโอด D_3, D_4 และ R_{10} ต่อร่วมเป็น AND GATE ช่วงของ ROM คือ 0000H-7FFFH ส่วน RAM ใช้สัญญาณจากขา Y_2 ของ 74LS139 ได้โดยตรงทำให้ได้ช่วงแอดเดรสเป็น 8000H-BFFFH

ส่วนพอร์ตใช้ 82C55 เป็นพอร์ตอินพุต เอาต์พุตโดยใช้ 74LS139 เป็นตัว DECODER ตำแหน่งของพอร์ต ใช้แอดเดรส A6 และ A7 มาทำการ DECODE พร้อมกับสัญญาณ IORQ โดยจะ DECODE แบ่งออกเป็น 4 ช่วง แออดเดรสช่วงละ 64 ตำแหน่งแบ่งเป็นส่วนของ 82C55 ตั้งแต่ตำแหน่ง 40H-7FH ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT A = 40H

PORT B = 41H

PORT C = 42H

CONTROL PORT = 43H

PORT A เป็นเอาต์พุตเพื่อที่จะเลือกตำแหน่งของ DISPLAY โดยจะต่อร่วมกับ 74145 ซึ่งเป็น BCD to DECIMAL

PORT B เป็นเอาต์พุตเพื่อที่จะใช้เป็น DATA ของ DISPLAY โดยต่อเข้ากับ BUFFER

การส่งรหัสหมายเลขโดยใช้การส่งสัญญาณแบบ DTMF ซึ่งเป็นการส่งสัญญาณภานในสาย 2 เส้น ซึ่งสามารถที่จะส่งสัญญาณภายในสายได้ระยะไกลโดยใช้หลักการการนำความถี่ที่มีความแตกต่างกัน 2 ความถี่ ซึ่งเกิดจากการ DECODE ปุ่มหมายเลขให้เป็นสัญญาณความถี่แล้วนำมามอดูเลตกัน ซึ่งการ DECODE ความถี่และการมอดูเลตนี้จะใช้ร่วมกับแป้นปุ่มหมายเลข [KEY PAD] ชนิด DPST หรือชนิด SPST ไอซี UPC 757C นี้จะประกอบด้วยวงจรถอดรหัสลอจิกเกตวงจรหารความถี่หรือ CTRDIV K เมื่อมีการกดหมายเลขทำให้สัญญาณโรว์และคอลัมน์เกิดขึ้น และขาสัญญาณ SINGLE TONE ENABLE จะต้องถูกทำให้แอกทีฟสัญญาณดรว์ และคอลัมน์ และขา 18,19,20,21 เป็นโรว์ เมื่อมีการกด KEY สัญญาณ DTMF ที่ออกเป็น 2 ความถี่ ซึ่งสัญญาณจากสายจะส่งเข้า MT 8870 เป็น IC DECODE สัญญาณ DTMF ให้เป็นตัวเลข BCD ขนาด 4 บิต เพื่อที่จะเข้าไปเป็นอิตพุตของ82C55เพื่อให้CPU ประมวลผลซึ่งสายที่ใช้ส่งสัญญาณนี้จะต้องมีไฟ DC ไปควบคู่กับสัญญาณเพื่อเป็นไฟไปเลี้ยง BOARD ต่าง ๆ ด้วย POWER SUPPLY ที่ใช้เท่ากับ +24VDC โดยใช้ AC24V ผ่านวงจร RECTIFIER และ FILTER ทำให้ได้ 24VDC จากนั้นเข้าวงจร REGULATOR เพื่อให้ได้ 24VDC ไปเลี้ยง CONTROLLER BOARD,DTME KEYBOARD,DTME DECONDER และ DISPLAYBOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุด DISPLAY

ในปัจจุบันนี้ป้ายโฆษณาที่มีลักษณะเป็นตัวอักษรเลื่อนกำลังเป็นที่นิยม ดังเห็นได้จากตามห้างสรรพสินค้าทั่วไป มักจะมีป้ายโฆษณาในลักษณะนี้ติดตั้งอยู่มากมาย สามารถดึงดูดความสนใจได้มากกว่าป้ายโฆษณาทั่วๆ ไป เนื่องจากมองเห็นได้เด่นชัด มีรูปแบบในการแสดงผลที่น่าสนใจ จึงมีผู้นิยมใช้กันมากขึ้นเรื่อยๆ ในการที่จะมาศึกษาถึงการทำป้ายโฆษณาในลักษณะนี้ก็ไม่ใช่เรื่องยากนัก โดยเริ่มจากป้ายเล็กๆ ก่อนแต่ก็สามารถนำไปใช้งานได้ และเมื่อเข้าใจหลักการต่างๆ แล้วก็สามารถทำป้ายที่มีขนาดใหญ่กว่านี้และมีลูกเล่นมากกว่านี้ได้

ในการทำไฟวิ่งมีหลักการอยู่การสแกน ซึ่งการสแกนนี้สามารถทำได้ 2 วิธีคือ สแกนทางคอลัมน์กับสแกนทางโรว์

การสแกนทางคอลัมน์จะทำการส่งข้อมูลออกไปทางโรว์ โดยส่งข้อมูลตัวที่ 1 ออกไป แล้วให้คอลัมน์ที่ 1 แอกตีฟ จากนั้นก็ทำการส่งข้อมูลตัวที่ 2 ออกไป แล้วให้คอลัมน์ที่ 2 แอกตีฟ ทำเช่นนี้ไปจนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกคอลัมน์ก็จะเป็นการสแกนครบ 1 รอบดังนั้นถ้าจำนวนหลักที่จะแสดงผลออกมาเป็นตัวอักษรมีจำนวนหลายหลัก วิธีนี้จะไม่เหมาะสมนักที่จะนำมาใช้งานเพราะว่าเมื่อให้ LED ที่คอลัมน์สุดท้ายจะติดจะต้องใช้เวลานาน

ส่วนการสแกนทางโรว์จะทำการส่งข้อมูลออกไปทางคอลัมน์ โดยจะส่งข้อมูลออกไปจนครบหมดทุกหลักก่อนแล้วให้โรว์ที่ 1 แอกตีฟ จากนั้นก็ทำการส่งข้อมูลชุดถัดไปออกไปจนครบหมดทุกหลัก แล้วให้โรว์ที่ 2 แอกตีฟทำเช่นนี้จนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกโรว์ ก็จะเป็นการสแกนครบ 1 รอบ วิธีนี้มีข้อดีคือสามารถแสดงผลเป็นตัวอักษรพร้อมกันได้หลายหลักและถ้าจัดเวลาให้เหมาะสมแล้ว เวลาทำการสแกนจะไม่ทำให้เกิดอาการพริ้ว แต่มีข้อเสียคือการเขียนโปรแกรมควบคุมให้ตัวอักษรเลื่อนทำได้ยากกว่าแบบแรก ในโครงงานนี้จะใช้วิธีการสแกนทางโรว์เพื่อควบคุมการเลื่อนของตัวอักษร

การทำงานทางฮาร์ดแวร์

วงจรของแผงอักษรไฟวิ่งแสดงดังรูปที่ 1

เริ่มที่ IC_{19} 74LS145 จะทำหน้าที่เลือกตำแหน่งหลักในการส่งข้อมูลออกไปโดยใช้ 4 บิต ล่างของพอร์ต B คือ PB0,PB1,PB2,PB3 เป็นตัวควบคุมเอ๊าต์พุตของ IC_{19} จะต่อเข้ากับขา CLK ของ IC_{10}-IC_{14} เบอร์ 74LS374 IC_{10}-IC_{14} จะทำหน้าที่แลตช์ข้อมูลค้างเอาไว้ โดยที่ข้อมูลจะถูกส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากพอร์ต A ในการส่งข้อมูลนั้นจะต้องทำการเลือกหลักที่จะส่งข้อมูลนั้นก่อน โดยให้ขา CLK ของ IC_{10}-IC_{14} เป็น "0" แล้วจึงทำการส่งข้อมูลออกไป จากนั้นจะต้องให้ขา CLK ของ IC_{10}-IC_{14} เป็น "1" ข้อมูลจะถูกแลตช์ค้างไว้ (ข้อมูลจะถูกแลตช์ไว้ที่ขอบขาขึ้น)

เอาต์พุตของ IC_{10}-IC_{14} จะต่อกับอินพุตของ IC_4-IC_9 ถ้าอินพุตของ IC_4-IC9 เป็น "1" จะได้เอาต์พุตเป็น "0" ใช้ในการขับ LED ด้วย เมื่อข้อมูลถูกส่งออกมาจากพอร์ต A ทั้ง 8 บิต แล้ว ทรานซิสเตอร์ Q_1-Q_8 ที่ต่อไว้ขับ LED ติดตามข้อมูลที่ส่งออกมา

ส่วน IC_1 เบอร์ 74HCT138 จะทำหน้าที่สแกนทางโรว์ โดยใช้ 4 บิตบนของพอร์ตเป็นตัวควบคุมการสแกน โดยแบ่งเป็น PB4,PB5,PB6 ทำหน้าที่สแกนส่วน PB7 จะทำหน้าที่เปิด/ปิดการสแกนถ้า PB7 เป็น "1" ก็จะเป็นการปิดการสแกนนั่นคือ เอาต์พุตของ IC_1 จะเป็น "1" หมด ก็จะทำให้ Q_1-Q_8 อยู่ในสภาวะคัตออฟ เอาต์พุตของ IC_1 จะติดกับ IC_2 เบอร์ 74LS07 ซึ่งจะทำหน้าที่เป็นบัฟเฟอร์และขับทรานซิสเตอร์ โดยมีเอาต์พุตเป็นแบบคอลเล็กเตอร์จึงต้องต่อต้านทาน R_{50} และ $R_{50/1}$ พูลอัพไว้

IC_{17} และ IC_{18} เบอร์ 74HCT123 จะต่อเป็นวงจรโมโนสเตเบิล ถ้าอินพุตของ IC_{17} และIC_{18} ถูกทริกภายในช่วงเวลาที่ถูกกำหนดโดยค่า RC เอาต์พุตของไอซีทั้งสองก็จะเป็น "0" ถ้าไม่เช่นนั้นเอาต์พุตเป็น "1" IC 18 จะทำหน้าที่ป้องกัน LED ติดค้างหรือนานเกินไป เพราะจะทำให้หลอดขาด เนื่องจากกระแสไหลมากเกิน ทั้งนี้เพราะว่าตัวต้านทานที่ต่อจำกัดกระแสให้ LED มีขนาดเพียง 10 โอห์มเท่านั้น หากใช้ค่าตัวต้านทานสูงกว่านี้ LED ก็จะไม่สว่างเท่าที่ควร

อินพุตของ IC_{18} จะต่ออยู่กับเขา A ของ IC_1 ส่วนเอาต์พุต Q ก็จะต่อยู่กับขา IC_1 ทำให้เกิดการสแกน ซึ่งช่วยให้ LED ติดสว่างไม่นานเกินไป

IC_{17}จะต่ออยู่กับ $IC_{15/2}$ เบอร์ 556 ทำหน้าที่เป็นวงจรวอตช์ด็อก ถ้าซีพียูเกิดอาการแฮงก์หรือติดอยู่ในลูปใดลูปหนึ่งของโปรแกรม แล้วส่งสัญญาณทริกออกมาทาง PC7 ไม่ทัน ตามเวลาที่กำหนดโดยค่า RC ที่ต่ออยู่ในวงจร (ประมาณ 3 วินาที) วงจรในส่วนนี้ก็จะส่งสัญญาณมารีเซตซีพียูให้เริ่มต้นทำงานใหม่ สัญญาณทริกที่ส่งออกมาทาง PC7 จะต้องส่งออกมาในลักษณะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัลส์ ส่วน $IC_{15/1}$ ต่อเป็นวงจรอะสเตเบิลเพื่อกำเนิดสัญญาณอินเตอร์รัพต์ที่สามารถปรับช่วงเวลาของการอินเตอร์รัพต์ได้ โดยการปรับค่า VR_1 ช่วงเวลาที่สามารถปรับได้อยู่ระหว่าง 1-1.75 มิลลิวินาที

การสแกน

ก่อนอื่นต้องทำการปิดการสแกนก่อน ซึ่งก็คือ การทำให้ทรานซิสเตอร์ทั้ง 8 ตัว อยู่ในสภาวะคัตออฟทำได้โดยการเซตให้ PB7 เป็น "1" IC_1 จะถูกดิสเอเบิลเพื่อไม่ให้ข้อมูลที่ไม่ต้องการหลุดออกไป ในการสแกนนี้จะทำการเลือกหลักที่จะส่งข้อมูลออกไปด้วย เพราะพอร์ต B ถูกแบ่งออกเป็น 2 ส่วนคือ 4 บิตบนทำหน้าที่สแกนทางด้านโรว์ และควบคุมว่าจะเปิดหรือปิดการสแกน

โดยปกติจะให้หลักแรกทำงานก่อนพร้อมทั้งปิดการสแกนด้วย หลังจากนั้นก็นำข้อมูลที่อยู่ในดิสเพลย์บัฟเฟอร์ของแถวที่ 1 หลักที่ 1 ออกไปที่พอร์ต A ข้อมูลที่ถูกส่งออกไปจะยังไม่ถูกแลตช์ไว้ จนกว่าจะให้ขา CLK ของ IC_{10} เป็น "1" ข้อมูลจึงจะถูกแลตช์ข้อมูลไว้แล้ว ก็ให้หลักที่ 2 ทำงาน แล้วจึงนำข้อมูลในดิสเพลย์บัฟเฟอร์ของแถวที่ 1 หลักที่ 2 ส่งออกไปที่พอร์ต A

ทำการส่งข้อมูลในลักษณะนี้จนครบ 5 หลัก จากนั้นจึงเลือกให้แถวที่ 1 ทำงาน พร้อมกับเปิดการสแกนด้วย ขั้นตอนถัดมาก็ทำการหน่วงเวลาเพื่อให้ LED ติดชั่วขณะหนึ่งตามข้อมูลที่ได้ส่งออกไปทั้ง 5 หลัก ในการเปิดการสแกนก็เพียงแต่ให้ PB7 เป็น "0" จากนั้นจะต้องทำการเคลียร์ข้อมูลเก่าเสียก่อน โดยการส่งข้อมูลเป็น 00H ออกไปทั้ง 5 หลักตามวิธีที่กล่าวมาแล้ว ทั้งนี้เพื่อป้องกันความผิดพลาดที่อาจจะเกิดขึ้นของข้อมูลที่ไม่ต้องการเกิด จากนั้นจึงทำการส่งข้อมูลชุดใหม่ที่อยู่ในดิสเพลย์บัฟเฟอร์ที่ต้องการให้ติดในแถวที่ 2 ออกไปทั้ง 5 หลัก และให้แถวที่ 2 ทำงาน

ตามวิธีที่กล่าวมาแล้วทำจนครบทั้ง 8 แถวก็จะมองเป็น LED ติดเป็นตัวอักษร แต่จะเห็นเพียงชั่วขณะเท่านั้น จึงจำเป็นต้องส่งข้อมูลชุดเดิมทั้งชุดออกไปอีก จึงจะมองเห็น LED ติดเป็นตัวอักษร

ลักษณะการสแกนและเลื่อนข้อมูลแสดงดังในรูปที่ 2

การจัดหน่วยความจำ

ในการใช้งาน ET-LDB ต้องมีการเขียนโปรแกรมควบคุมและกำหนดข้อมูลที่ต้องการแสดง ดังนั้นจึงต้องมีการจัดหน่วยความจำที่เรียกว่า ดิสเพลย์บัฟเฟอร์มีตำแหน่งตั้งแต่ 3000H-302FH แบ่งออกเป็น 6 ส่วน ดังรูปที่ 3 นั่นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ตำแหน่ง 3000H-302FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 1 ที่ตำแหน่ง 0008H-300FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 2

ที่ตำแหน่ง 3010H-3017H เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 3

ที่ตำแหน่ง 3018H-301FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 4

ที่ตำแหน่ง 3020H-3027FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 5

ที่ตำแหน่ง 3028H-3022FH เป็นดิสเพลย์บัฟเฟอร์ของหลักที่ 6

ในแต่ละหลักจะมีข้อมูลอยู่ 8 ไบต์ ซึ่งจะเป็นข้อมูลของตัวอักษร 1 ตัวลักษณะของข้อมูลของตัวอักษร 1 ตัว ดังแสดงในรูที่ 4 ซึ่งเป็นตัวอักษรตัว A

ข้อมูลที่อยู่ดิสเพลย์บัฟเฟอร์ของหลักที่ 1 ถึงหลักที่ 5 เป็นข้อมูลที่นำออกมาแสดงผลทาง LED ส่วนข้อมูลที่อยู่ในดิสเพลย์บัฟเฟอร์ของหลักที่ 6 จะทำหน้าที่เก็บข้อมูลของตัวอักษรถัดไปที่จะนำมาแสดงผล

ในตอนแรกข้อมูลของหลักที่ 1 ถึง 5 จะมีค่าทั้งหมดเป็น 00H ส่วนข้อมูลของตัวอักษรตัวแรกที่จะนำออกมาแสดงผลจะถูก เก็บไว้ในหลักที่ 6 เมื่อผ่านขั้นตอนการสแกนและเลื่อนข้อมูลจนข้อมูลถูกเลื่อนไปครบ 8 บิต(ครบ 1 หลัก) ตัวอักษรตัวแรกก็จะมาอยู่ที่หลักที่ 5 เราก็จะทำการส่งข้อมูลของตัวอักษรตัวที่ 2 มาที่หลักที่ 6 และเมื่อผ่านขั้นตอนการสแกนและเลื่อนข้อมูลจนข้อมูลถูกเลื่อนไปอีก 8 บิต ตัวอักษรตัวแรกก็จะมาอยู่ที่หลักที่ 4 ตัวอักษรตัวที่ 2 ก็จะย้ายมาอยู่ที่หลักที่ 5 จากนั้นก็ทำการส่งข้อมูลของตัวอักษรตัวที่ 3 มาที่หลักที่ 6

ดังนั้นดิสเพลย์บัฟเฟอร์ในหลักที่ 6 จะทำหน้าที่รับข้อมูลของตัวอักษรตัวใหม่ที่จะนำมาแสดงผลเก็บไว้ และจะใช้เป็นส่วนหนึ่งในกระบวนการเลื่อนข้อมูลอีกด้วย

การเลื่อนข้อมูล

ในการที่จะทำให้เห็นเป็นอักษรวิ่งนั้นในที่นี้เราใช้วิธีการเลื่อนข้อมูล โดยที่ข้อมูลในแต่ละหลักจะมีขนาด 8 บิต เมื่อทำการเลื่อนข้อมูลไปทางซ้าย 1 ครั้ง ข้อมูลที่อยู่ในแต่ละหลักจะถูกเลื่อนไปทางซ้าย 1 บิต และข้อมูลในบิตที่ 7 จะถูกเลื่อนไปยังหลักถัดไป นั่นคือข้อมูลที่อยู่บิต 7 ของหลักถัดไป นั่นคือข้อมูลที่อยู่ที่บิต 7 ของหลักที่ 2,3,4,5 และ 6 จะถูกเลื่อนมาอยู่ที่บิต 0 ของหลักที่ 1.2.3,4 และ 5

เมื่อเราทำการสแกนข้อมูลที่ถูกเลื่อนข้อมูลไปแล้ว ก็จะมองเห็นตัวอักษรถูกเลื่อนมาทางซ้าย 1 ตำแหน่ง และเมื่อทำการเลื่อนข้อมูลและทำการสแกนอีก ก็จะมองเห็นตัวอักษรถูกเลื่อนไปอีก 1 ตำแหน่ง ดังนั้นเมื่อทำเช่นนี้ไปเรื่อยๆ ก็จะสามารถมองเห็นเป็นตัวอักษรวิ่งไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเลื่อนข้อมูลสามารถอธิบายได้ดังนี้

สมมติว่ามีข้อมูลที่จะนำออกมาแสดงผลทาง LED 1 หลัก ก็จะต้องมีข้อมูลตัวถัดไปที่จะนำออกมาแสดงผลอยู่ในหลักที่ 2 เพื่อที่จะทำการเลื่อนข้อมูลในหลักที่ 2 มายังหลักที่ 1 ซึ่งจะทำให้เห็นเป็นตัวอักษรเลื่อนไปได้ สมมติว่าข้อมูลทั้ง 2 หลัก มีอยู่หลักละ 1 ไบต์ แล้วต้องการที่จะทำการเลื่อนข้อมูลจากหลักที่ 2 มายังหลักที่ 1 สามารถอธิบายได้ดังนี้

ขั้นแรกจะต้องทำการหมุนข้อมูลของทั้ง 2 หลัก โดยใช้คำสั่ง RLCA ก่อน (คำสั่งใน Z80) จากนั้นก็ให้ข้อมูลในหลักที่ 1 บิต 0 มีค่าเป็น "1" และให้ข้อมูลในหลักที่ 2 บิต 1 ถึงบิต 7 มีค่าเป็น "0" จากนั้นก็นำข้อมูลของทั้ง 2 หลัก มาบวกกันก็จะได้ข้อมูลตัวใหม่ของหลักที่ 1 ส่วนข้อมูลในหลักที่ 2 จะไม่นำมาแสดงผล เมื่อทำการสแกนข้อมูลของหลักที่ 1 ก็จะมองเห็นว่าข้อมูลทั้งหมดของหลักที่ 1 ถูกเลื่อนไปทางซ้าย 1 ตำแหน่ง และถ้าต้องการเลื่อนข้อมูลไปทางซ้ายอีก 1 ตำแหน่ง ก็สามารถทำได้โดยวิธีที่กล่าวมาแล้ว โดยข้อมูลที่จะนำมาใช้คือ ข้อมูลในหลักที่ 1 ที่ผ่านการเลื่อนข้อมูลแล้วในครั้งแรก ส่วยข้อมูลในหลักที่ 2 ก็คือ ข้อมูลก่อนที่จะถูกทำให้บิต 1 ถึงบิต 7 มีค่าเป็น "0" และเมื่อทำการเลื่อนข้อมูลจนครบ 8 ไบต์ในหนึ่งหลัก ดังนั้นเวลาทำการเลื่อนข้อมูลก็จะต้องทำทั้ง 8 ไบต์ด้วย

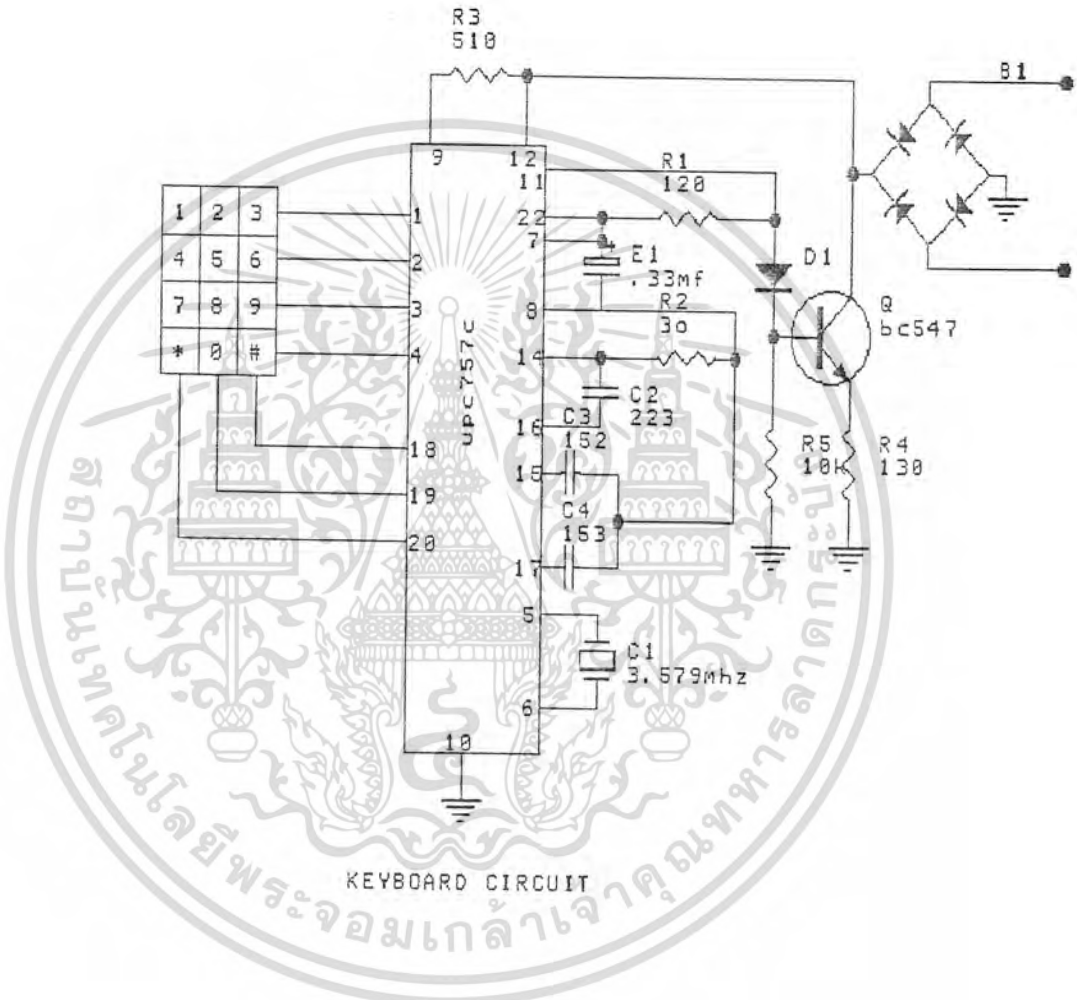
ถ้าต้องการนำข้อมูลออกมาแสดงผลทั้งหมด 5 หลัก ก็จะต้องมีข้อมูลทั้งหมด 6 หลัก โดยหลักที่ 6 จะทำหน้าที่ดังที่กล่าวมาแล้วในตอนแรก ในรูปที่ 5 จะแสดงถึงวิธีการเลื่อนข้อมูลที่ได้กล่าวมาแล้ผู้ใช้สามารถใช้วิธีอื่นๆ ในการเลื่อนข้อมูลได้ โดยใช้คำสั่งที่มีอยู่ในซีพียูเบอร์นั้นๆ หรือจะใช้เทคนิคอื่นๆ ก็ได้

การสร้าง

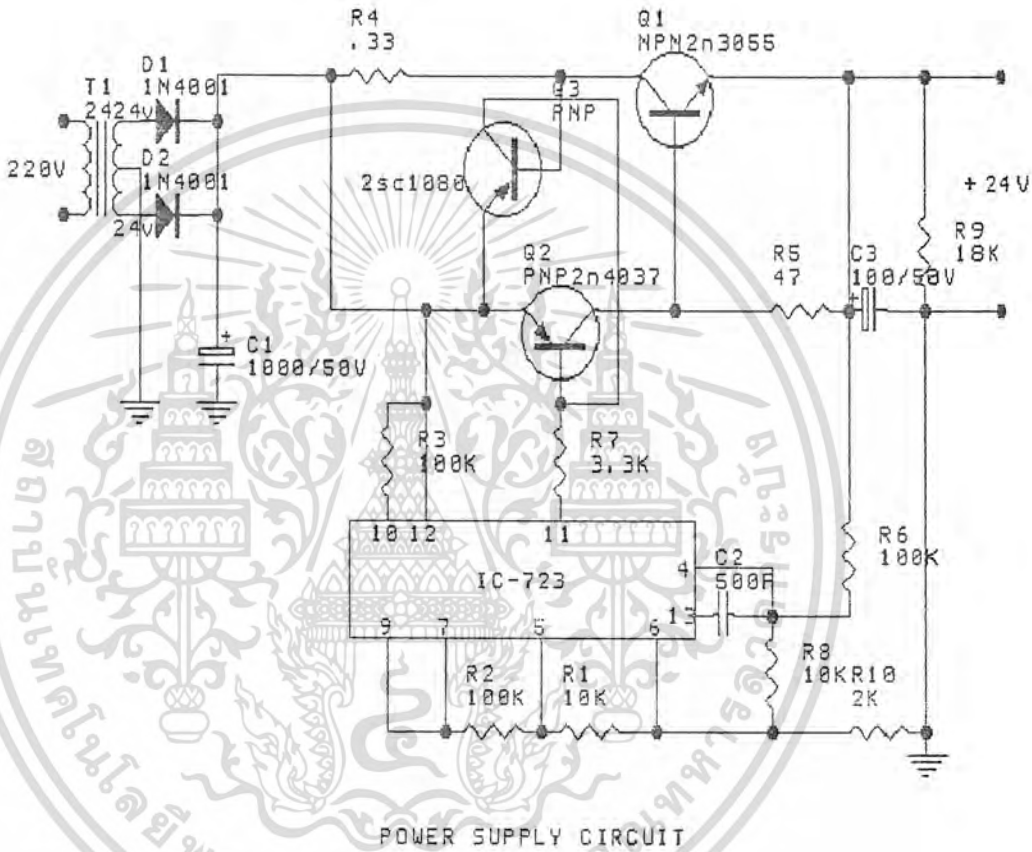
ก็ต้องเริ่มที่แผ่นวงจรพิมพ์ ซึ่งต้องเป็นชนิดสองหน้าเพลตทรูโฮลด์ รูปที่ 6 เป็นลายทองแดงของแผ่นวงจรพิมพ์ทั้ง 2 ด้าน ซึ่งมีขนาด 50% ของขนาดจริง รูปที่ 7 เป็นการวางอุปกรณ์ .

การติดตั้งLED จะต้องค่อนข้างพิถีพิถันอย่างมากคือ ต้องทำให้ตรงกันมีความสูงเท่ากัน ดังนั้นจึงติด LED ให้ชิด กับแผ่นวงจรพิมพ์เลย ดังในรูปที่ 8 จะเห็นได้ว่า LED ทั้ง 320 ดวง จะติดตั้งบนแผ่นวงจรพิมพ์ครบระดับกับอุปกรณ์อื่นๆ เมื่อประกอบเรียบร้อยจะมีหน้าตาดังรูปที่ 9 และ 10 โดยในรูปที่ 9 แสดงแผ่นกรองแสงที่ได้ปิดแผงอักษรไฟวิ่งด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

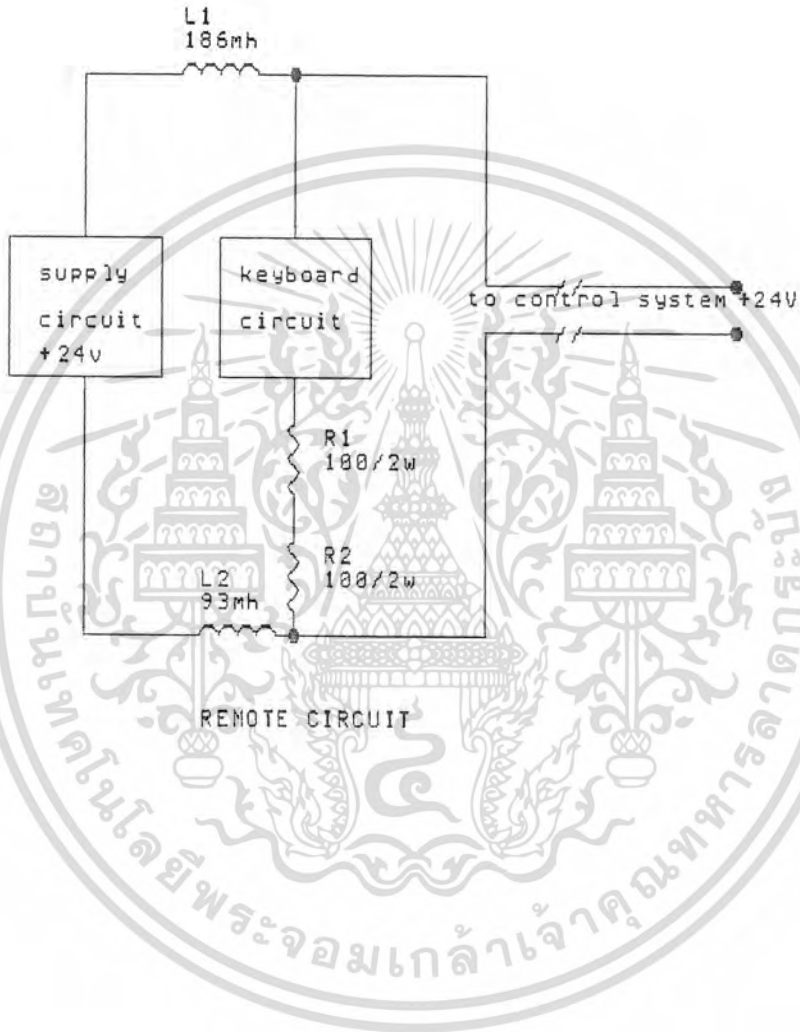


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

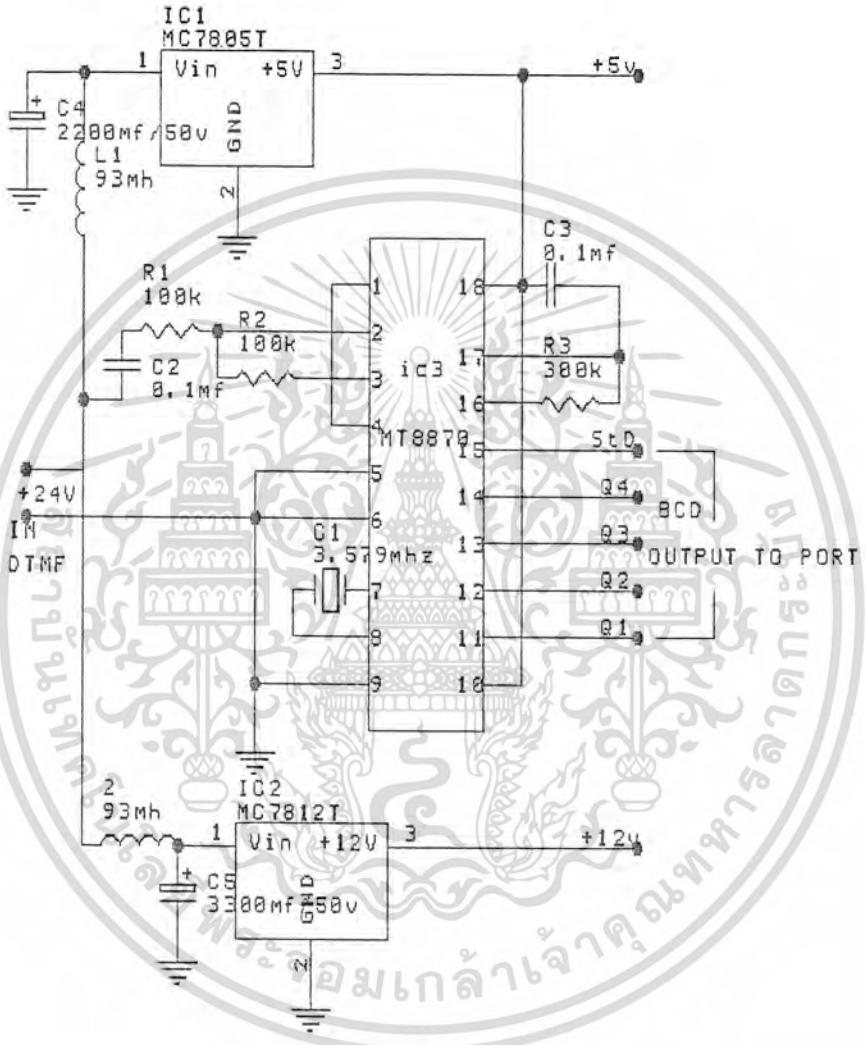


POWER SUPPLY CIRCUIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

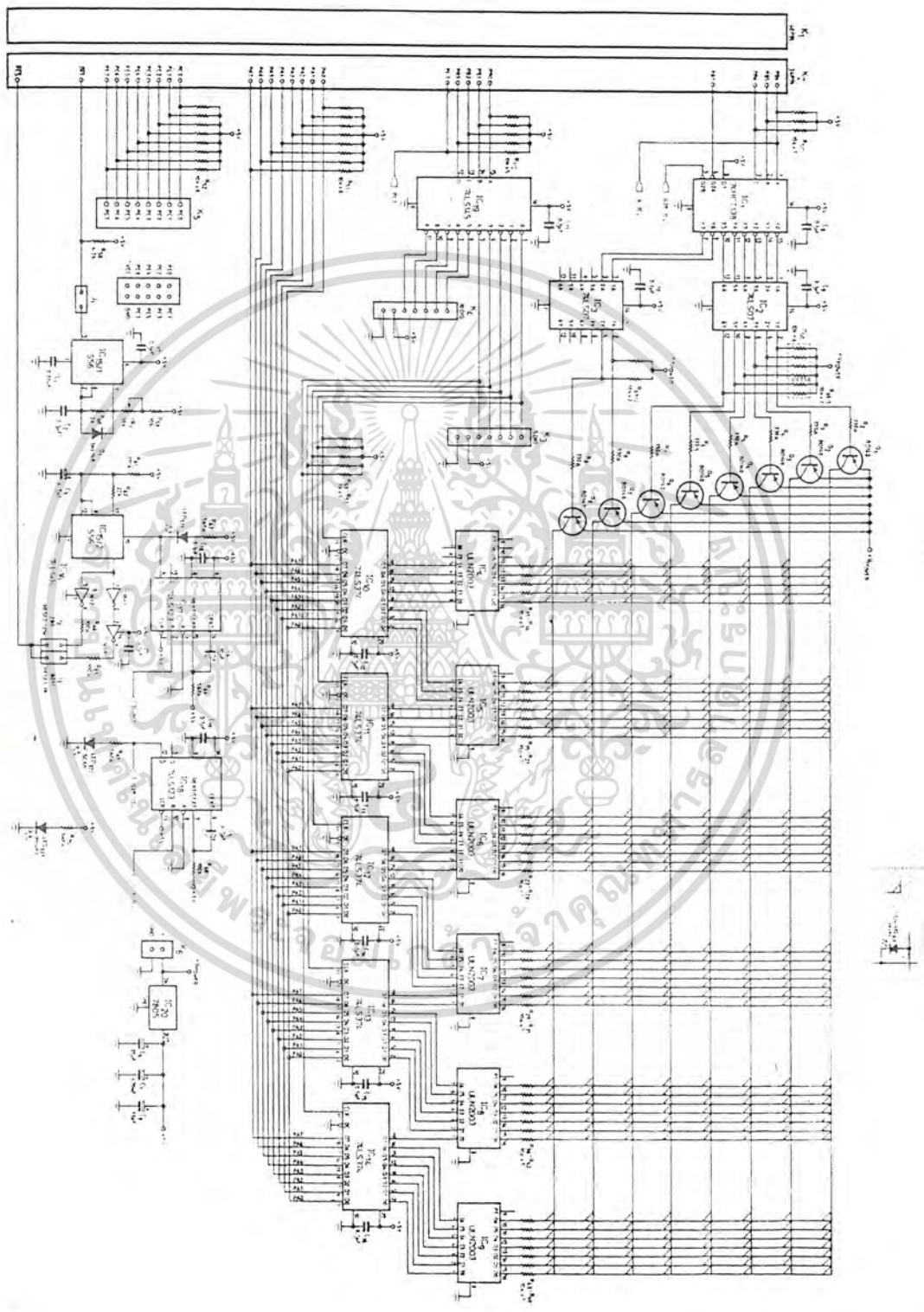


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

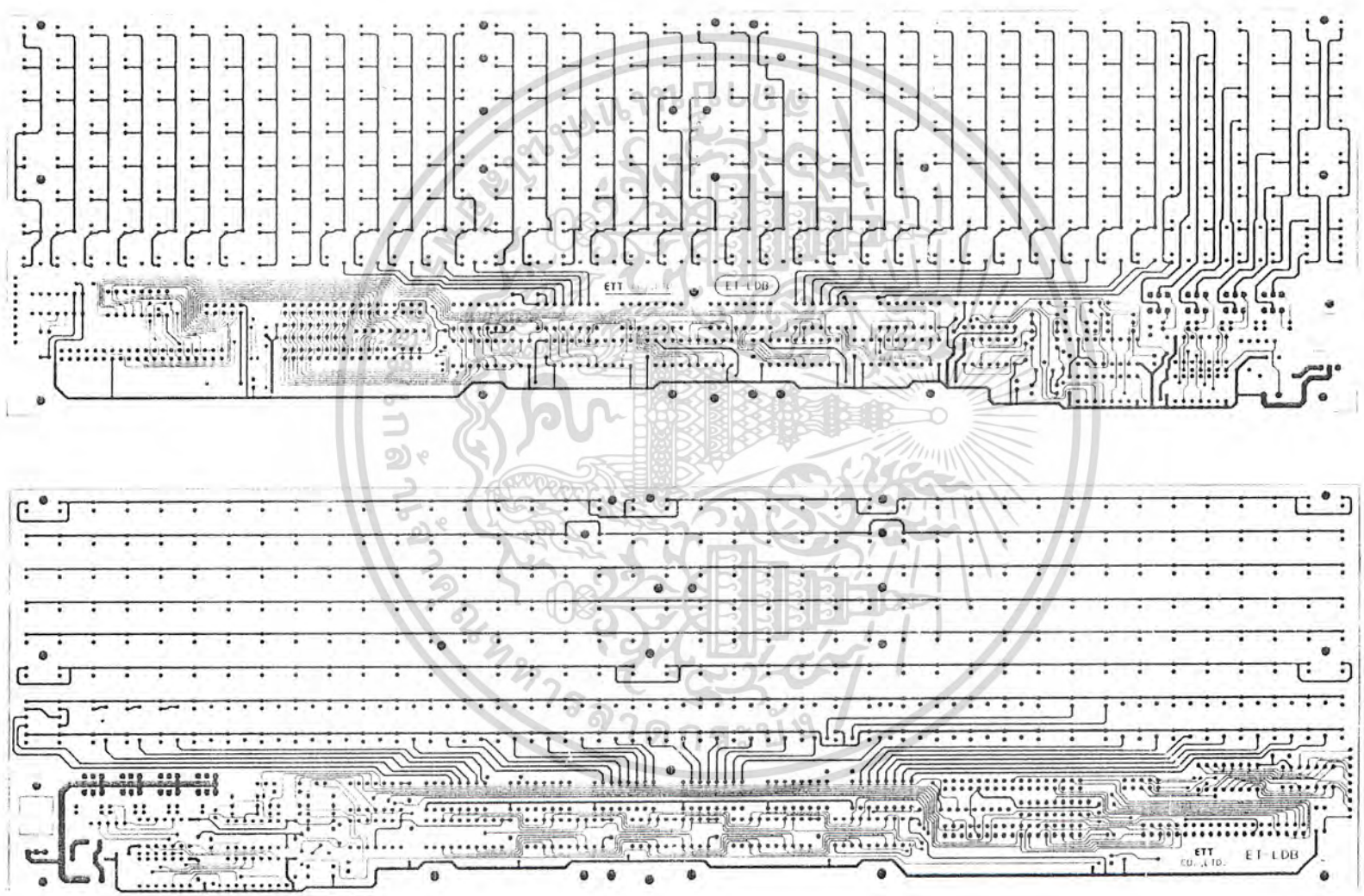


DECODER CIRCUIT

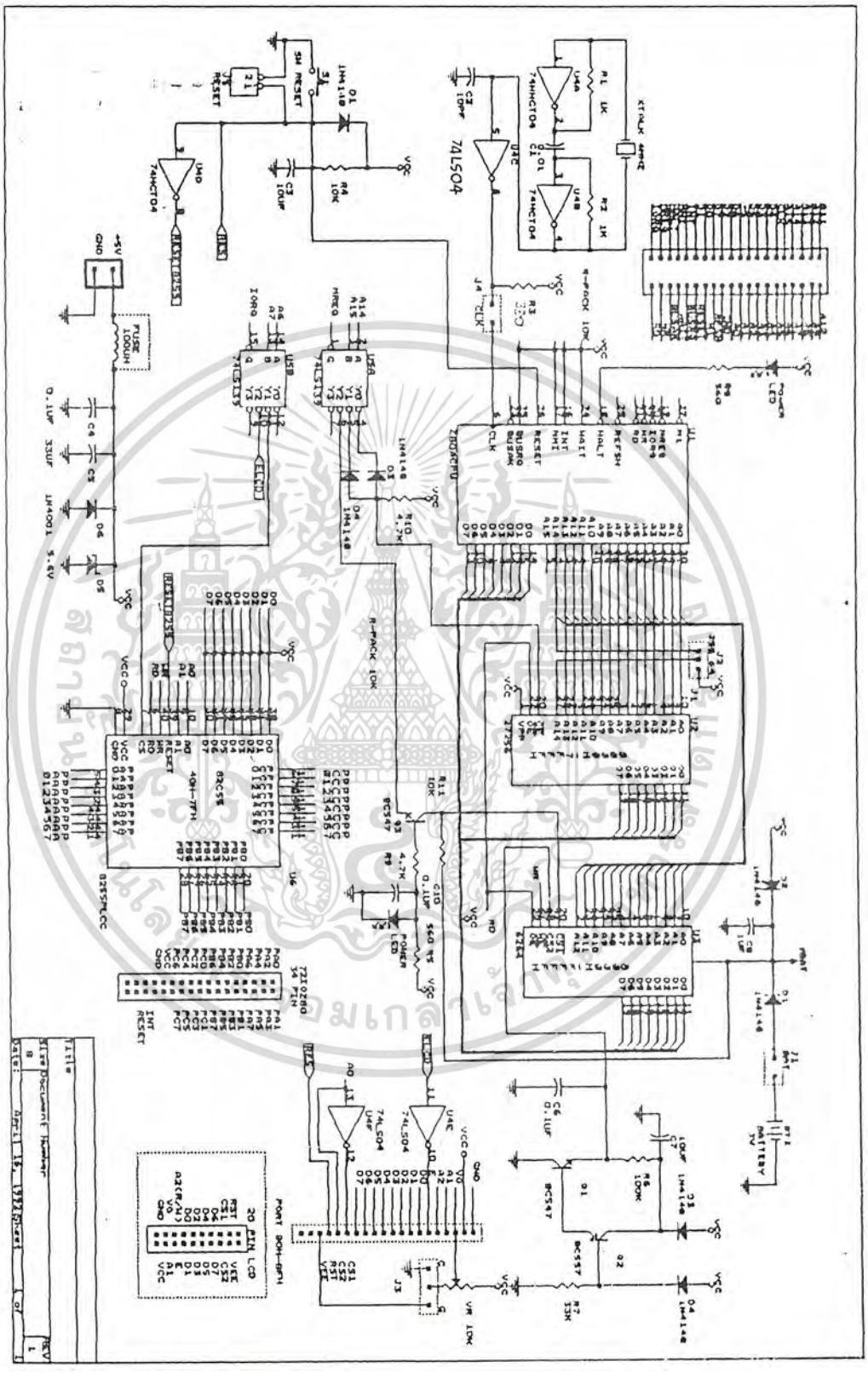
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



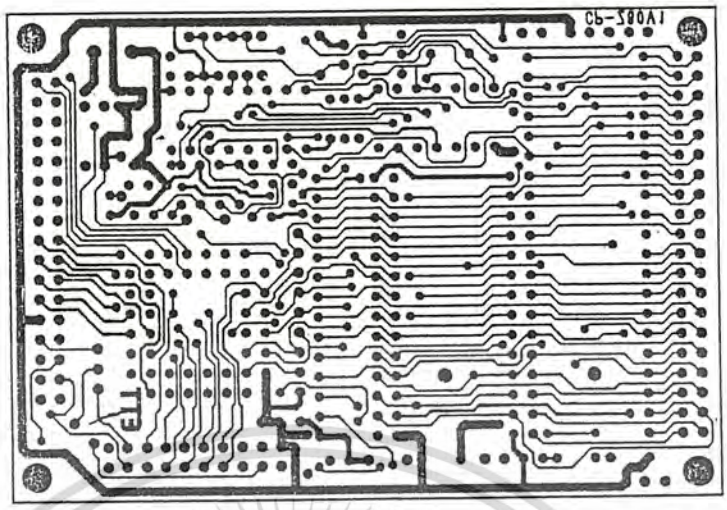
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

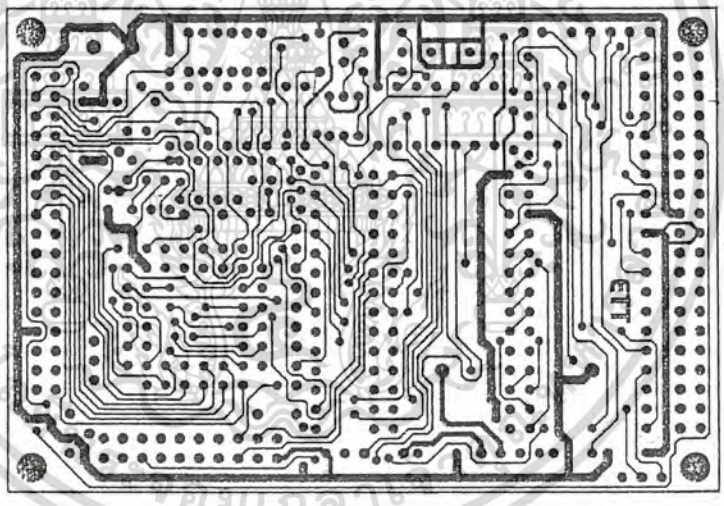


ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้เครื่อง

เครื่อง Remote Control Scoreboard diplayเป็นเครื่องที่เก็บข้อมูลไว้ในหน่วยความจำเมื่อจะใช้เครื่องต้องกดหมายเลขที่ Key Board ของ DTMF กด 0 จะขึ้นข้อความ Pottom Selection เพื่อให้เลือก Mode การทำงานว่าจะเลือก Mode ไหนในการนำแสดงผล ซึ่ง Mode การทำงานมีอยู่ 2 Mode

- Mode 1 กดหมายเลข 1 แสดงค่าตัวเลขการแสดงค่าตัวเลขสามารถแสดงค่าตัวเลขได้ 5 ตำแหน่ง ซึ่งสามารถเลือกได้ดังนี้

กด # 1 = แสดงตำแหน่งที่ 1

กด # 2 = แสดงตำแหน่งที่ 2

กด # 3 = แสดงตำแหน่งที่ 3

กด # 4 = แสดงตำแหน่งที่ 4

กด # 5 = แสดงตำแหน่งที่ 5

กด * ลบ หมายเลขตำแหน่งที่ต้องการ เช่น ต้องการลบตำแหน่งที่ 1 กด # 1 * ออกจาก mode ตัวเลขกด # ตามด้วย 0

- mode 2 กดหมายเลข 2,3,4,5,6 แสดงข้อความ ถ้าต้องการออกจาก mode 2 ให้กด 0

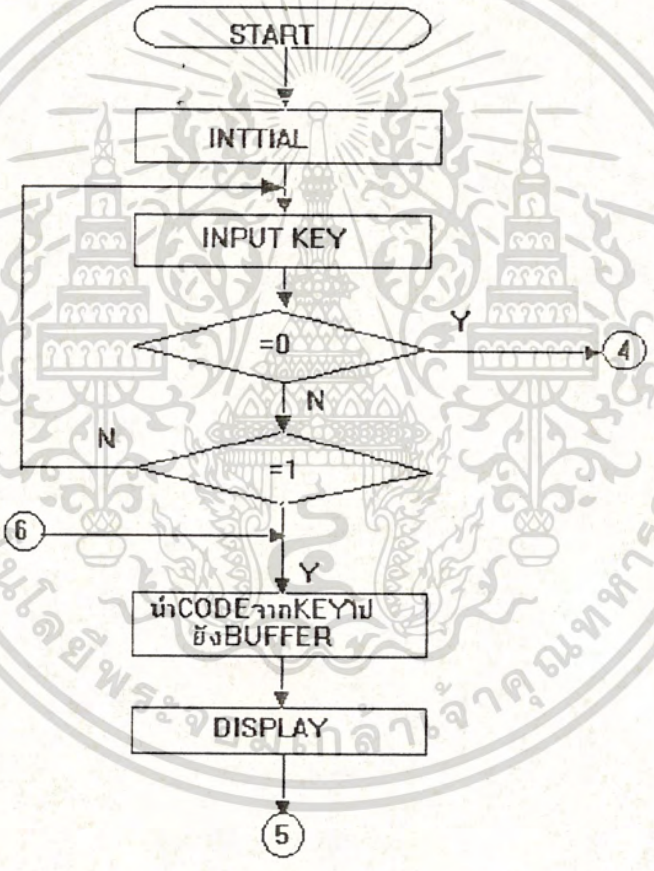
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

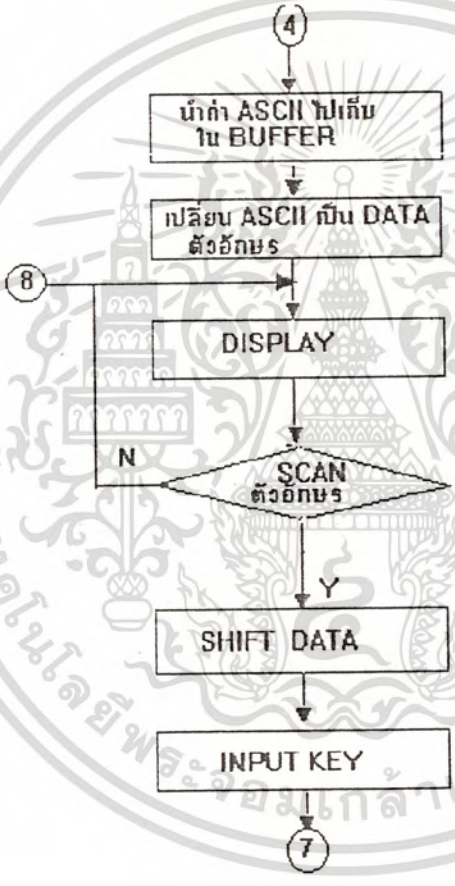
ในการออกแบบและสร้าง PROGRAMMABLE SCOREBOARD DISPLAY นี้ ผู้ทำโครงงาน ได้มีความรู้และความเข้าใจในการพัฒนาและใช้งานไมโครโปรเซสเซอร์ ให้มีขีดความสามารถสนองตอบความต้องการในการใช้งาน ดังนั้น การศึกษา ค้นคว้า เรียนรู้และการออกแบบอุปกรณ์ให้ทำงานได้อย่างมีประสิทธิภาพตามที่เราต้องการนั้น จำเป็นต้องมีพื้นความรู้ทาง HARDWRE และ SOFTWARE ประกอบกันการศึกษาและออกแบบป้ายแสดงอักษรวิ่งแบบโปรแกรมได้นี้ เป็นส่วนในการเริ่มให้รู้จักการประยุกต์ใช้งาน MICROPROCESSOR รวมทั้งการออกแบบโปรแกรมการใช้งานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

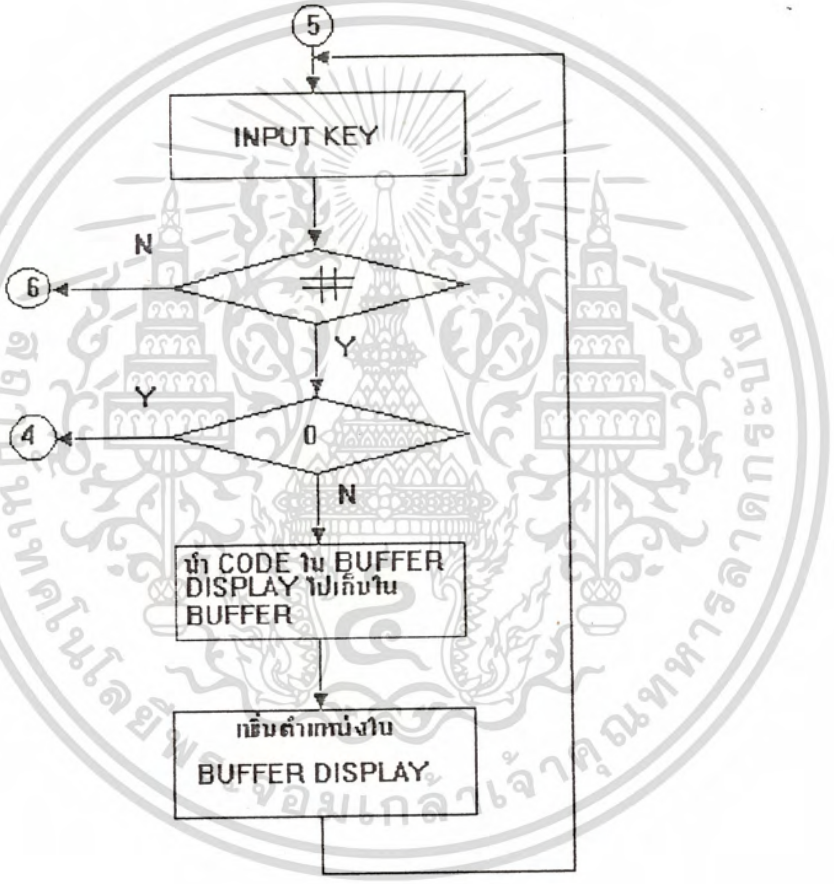
FLOWCHART



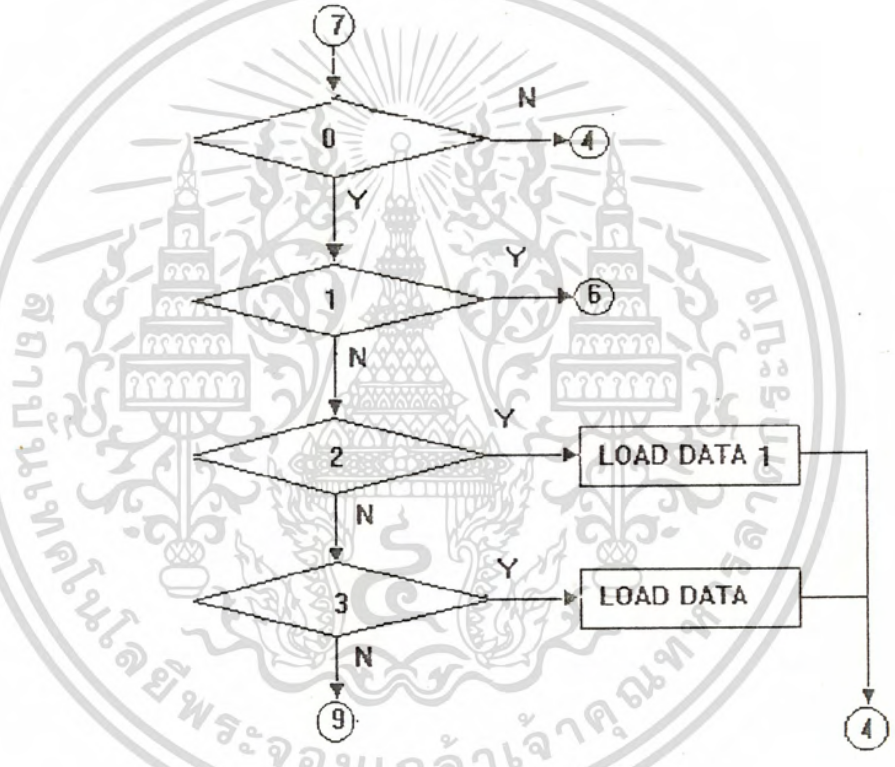
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



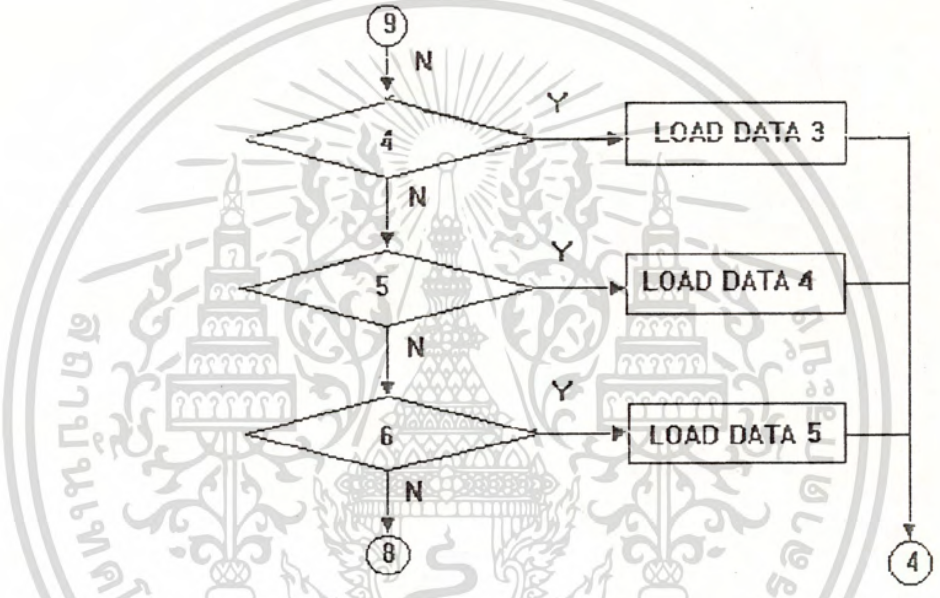
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ORG 8000H
COUNT5   EQU 9100H  ; นับจำนวนหลัก(5หลัก)
COUNT6   EQU 9101H  ; นับจำนวนหลักที่ใช้ในการ
                    ; เลื่อนข้อมูล(6หลัก)
COUNT8   EQU 9102H  ; นับจำนวนแถว(8แถว)
LOOP     EQU 9103H  ; ตำแหน่งในการเก็บ
                    ; จำนวนลูปของการ scan
LOOPCH   EQU 9104H  ; ตรวจสอบว่าตัวอักษรถูกเลื่อนไป
                    ; ครบ 8 บิตหรือยัง
ENDCH    EQU 9105H  ; ตรวจสอบว่าพบรหัสสิ้นสุดไฟล์หรือยัง
C_ENDCH EQU 9106H   ; ตรวจสอบว่าเมื่อพบรหัสสิ้นสุดไฟล์แล้ว
                    ; ตัวอักษรถูกเลื่อนไปครบ 5 หลักหรือยัง
BUFFER   EQU 9000H  ; ตำแหน่งเริ่มต้นของ display buffer
                    ; 5หลัก(9000H-9027H)
DISPY    EQU 9028H  ; ตำแหน่งเริ่มต้นของ display buffer
                    ; หลักที่ 6(9028H-902FH)
BUF1     EQU 9035H
NUMLP    EQU 05H    ; จำนวนลูปของการ scan
PORTA    EQU 40H    ; port A ของ 8255 บน ET-BOARD
PORTB    EQU 41H    ; port B ของ 8255 บน ET-BOARD
PORTC    EQU 42H    ; port C ของ 8255 บน ET-BOARD
CTPORT   EQU 43H    ; port control ของ 8255 บน ET-BOARD

         XOR A
START:   LD HL,0
ST1:     DEC HL
         LD A,L
         OR H
         JR NZ,ST1
         LD SP,9F00H
         XOR A
         CALL DELAY

         LD  A,89H
         OUT (CTPORT),A
         OUT (PORTB),A      ; OFF SCAN
         CALL DELAY
;-----------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ND:     CALL INITIAL
        ;-----------
        CALL CLEAR
        ;-----------
        LD B,00H
KKK:    IN A,(PORTC)
        AND 1FH
        CP 11H
        JR Z,MAIN
        CP 1AH
        JR Z,K1
        JP KKK
K1:     JP XX
        ;-----------
MAIN:   PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM:    PUSH IX
        ;-----------
        CALL OPEN
        ;-----------
        CALL CLOSE
        ;-----------
        POP IX
        INC IX
        CALL DEC8

        JR NZ,MMM
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN
        CP 1CH
        JP Z,FUNTT
        JP MAIN
;------------------------;
;       KEY2             ;
;------------------------;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FUNT2:  PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM2:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMM2
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN2
        CP 1CH
        JP Z,FUNTT
        JP FUNT2

;-----------------------;
;       KEY3            ;
;-----------------------------------------;
FUNT3:  LD IY,BUF1
        CALL MOV3
        PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM3:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL DEC8
        JR NZ,MMM3
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN3
        CP 1CH
        JP Z,FUNTT
        JP FUNT3
;-----------------------;
;       KEY4            ;
;-----------------------;
FUNT4:  LD IY,BUF1
        CALL MOV4
        PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM4:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMM4
        vLD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN4
        CP 1CH
        JP Z,FUNTT
        JP FUNT4
;-----------------------;
;       KEY5            ;
;-----------------------;
FUNT5:  LD IY,BUF1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FUNT5:  LD IY,BUF1
        CALL MOV5
        PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMM5:   PUSH IX
        ;---------
        CALL OPEN

        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMM5
        LD A,08H
        LD (COUNT8),A
        POP BC
        CALL KEYIN5
        CP 1CH
        JP Z,FUNTT
        JP FUNT5
;-----------------------;
FUNTT:  PUSH BC
        LD D,00H
        LD E,00H
        LD IX,BUFFER
MMMT:   PUSH IX
        ;---------
        CALL OPEN
        ;---------
        CALL CLOSE
        ;---------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,MMMT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD A,08H
                LD (COUNT8),A
                POP BC

                IN A,(PORTC)
                AND 1FH
          CP 12H
          JP Z,FUNT2
          CP 13H
          JP Z,FUNT3
          CP 14H
          JP Z,FUNT4
          CP 15H
          JP Z,FUNT5
          CP 11H
          JP Z,MAIN
          CP 1AH
          JP Z,FUNT10
          JP FUNTT
FUNT10:   JP XX
          ;-----------
          CALL DELAY
          ;-----------
          CALL INITIAL
          ;-----------
          CALL CLEAR
          ;-----------
          LD IY,DTBF1
          CALL DECODE
          CALL NEW

          ;-----------
MA11:     LD D,00H
          LD E,00H
          LD IX,BUFFER
M11:      PUSH IX

          ;-----------
          CALL OPEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------
CALL CLOSE
;-----------
POP IX
INC IX
CALL DEC8
JR NZ,M11
LD A,08H
LD (COUNT8),A

LD A,(LOOP)
DEC A
LD (LOOP),A
JR NZ,MA11      ; ตรวจสอบจำนวนรอบของการ scan
LD A,NUMLP
LD (LOOP),A
;-----------

LD A,(LOOPCH)
DEC A
LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
JR NZ,RS11      ; ไปครบ 8 บิตหรือยัง
LD A,08H
LD (LOOPCH),A

LD A,(ENDCH)
CP 01H
JR NZ,CO11      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
LD A,(C_ENDCH)
DEC A

LD (C_ENDCH),A
CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
JR NZ,TA11      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
LD A,05H
LD (C_ENDCH),A
XOR A
LD (ENDCH),A
LD IY,DTBF1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          JR CO11

TA11:     LD HL,TABLE
          JR CO21
CO11:     CALL DECODE
CO21:     CALL NEW

RS11:     ;-----------
          CALL ROTATE
          ;-----------
          CALL SHIFT
          ;-----------
          IN A,(PORTC)
          AND 1FH
          CP 11H
          JP Z,LL
          CP 12H
          JP Z,LLL
          CP 13H
          JP Z,LLLL
          CP 14H
          JP Z,LLLL1
          CP 15H
          JP Z,LLLL2

          CP 16H
          JP Z,LLLL3
          JP MA11
LL:       JP ND
LLL:      JP XXX1
LLLL:     JP XXX2
LLLL1:    JP XXX3
LLLL2:    JP XXX4
LLLL3:    JP XXX5
          ;-----------
          CALL DELAY
          ;-----------
          CALL INITIAL
          ;-----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                IN A,(PORTC)
                AND 1FH
            CP 12H
            JP Z,FUNT2
            CP 13H
            JP Z,FUNT3
            CP 14H
            JP Z,FUNT4
            CP 15H
            JP Z,FUNT5
            CP 11H
            JP Z,MAIN
            CP 1AH
            JP Z,FUNT10
            JP FUNTT
FUNT10:     JP XX
            ;-----------
xx:          CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF1
            CALL DECODE
            CALL NEW

            ;-----------
MA11:       LD D,00H
            LD E,00H
            LD IX,BUFFER
M11:        PUSH IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL CLEAR
        ;-----------
        LD IY,DTBF
        CALL DECODE
        CALL NEW
        ;-----------
MA22:   LD D,00H
        LD E,00H
        LD IX,BUFFER
M22:    PUSH IX
        ;-----------
        CALL OPEN
        ;-----------
        CALL CLOSE
        ;-----------
        POP IX
        INC IX
        CALL DEC8

        JR NZ,M22
        LD A,08H
        LD (COUNT8),A
        LD A,(LOOP)
        DEC A
        LD (LOOP),A
        JR NZ,MA22      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS22      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO22      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CP 16H
            JP Z,LLLL3
            JP MA11
LL:         JP ND
LLL:        JP XXX1
LLLL:       JP XXX2
LLLL1:      JP XXX3
LLLL2:      JP XXX4
LLLL3:      JP XXX5
            ;-----------
xxx1:       CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF
            CALL DECODE
            CALL NEW
            ;-----------
MA22:       LD D,00H
            LD E,00H
            LD IX,BUFFER
M22:        PUSH IX
            ;-----------
            CALL OPEN
            ;-----------
            CALL CLOSE
            ;-----------
            POP IX
            INC IX
            CALL DEC8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DEC A
        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA22      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
        LD A,05H
        LD (C_ENDCH),A
        XOR A
        LD (ENDCH),A
        LD IY,DTBF
        JR CO2
TA22:   LD HL,TABLE
        JR CO22

CO22:   CALL DECODE
CO2:    CALL NEW

RS22:   ;-----------
        CALL ROTATE
        ;-----------
        CALL SHIFT
        ;-----------
        IN A,(PORTC)
        AND 1FH
        CP 1AH
        JP Z,LAY1
        JP MA22
LAY1:   JP XX
        ;-----------
XXX2:   CALL DELAY
        ;-----------
        CALL INITIAL
        ;-----------
        CALL CLEAR
        ;-----------
        LD IY,DTBF2
        CALL DECODE
        CALL NEW
        ;-----------
MA33:   LD D,00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD E,00H
        LD IX,BUFFER
M33:    PUSH IX
        ;-----------
        CALL OPEN
        ;-----------

        CALL CLOSE
        ;-----------
        POP IX
        INC IX
        CALL DEC8
        JR NZ,M33
        LD A,08H
        LD (COUNT8),A
        LD A,(LOOP)
        DEC A
        LD (LOOP),A
        JR NZ,MA33      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS33      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO33      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
        DEC A
        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA33      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
        LD A,05H
        LD (C_ENDCH),A
        XOR A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD (ENDCH),A
        LD IY,DTBF2
        JR CO33
TA33:   LD HL,TABLE
        JR CO3
CO33:   CALL DECODE
CO3:    CALL NEW
RS33:   ;----------
        CALL ROTATE
        ;----------
        CALL SHIFT
        ;----------
        IN A,(PORTC)
        AND 1FH
        CP 1AH
        JP Z,LAY1
        JP MA33
        ;----------
XXX3:   CALL DELAY
        ;----------
        CALL INITIAL
        ;----------
        CALL CLEAR
        ;----------
        LD IY,DTBF3
        CALL DECODE
        CALL NEW
        ;-----------
MA44:   LD D,00H
        LD E,00H
        LD IX,BUFFER
M44:    PUSH IX

        ;----------
        CALL OPEN
        ;-----------
        CALL CLOSE
        ;----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP IX
        INC IX
        CALL DEC8
        JR NZ,M44
        LD A,08H
        LD (COUNT8),A
        LD A,(LOOP)
        DEC A
        LD (LOOP),A
        JR NZ,MA44      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS44      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO44      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
        DEC A
        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA44      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง

        LD A,05H
        LD (C_ENDCH),A
        XOR A
        LD (ENDCH),A
        LD IY,DTBF3
        JR CO44
TA44:   LD HL,TABLE
        JR CO4
CO44:   CALL DECODE
CO4:    CALL NEW
RS44:   ;-----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL ROTATE
            ;-----------
            CALL SHIFT
            ;-----------
            IN A,(PORTC)
            AND 1FH
            CP 1AH
            JP Z,LAY1
            JP MA44
            ;-----------
XXX4:       CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF4
            CALL DECODE
            CALL NEW
            ;-----------
MA55:       LD D,00H

            LD E,00H
            LD IX,BUFFER
M55:        PUSH IX
            ;-----------
            CALL OPEN
            ;-----------
            CALL CLOSE
            ;-----------
            POP IX
            INC IX
            CALL DEC8
            JR NZ,M55
            LD A,08H
            LD (COUNT8),A
            LD A,(LOOP)
            DEC A
            LD (LOOP),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR NZ,MA55      ; ตรวจสอบจำนวนรอบของการ scan
        LD A,NUMLP
        LD (LOOP),A
        ;-----------
        LD A,(LOOPCH)
        DEC A
        LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
        JR NZ,RS55      ; ไปครบ 8 บิตหรือยัง
        LD A,08H
        LD (LOOPCH),A
        LD A,(ENDCH)
        CP 01H
        JR NZ,CO55      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
        LD A,(C_ENDCH)
        DEC A

        LD (C_ENDCH),A
        CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
        JR NZ,TA55      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
        LD A,05H
        LD (C_ENDCH),A
        XOR A
        LD (ENDCH),A
        LD IY,DTBF4
        JR CO55
TA55:   LD HL,TABLE
        JR CO5
CO55:   CALL DECODE
CO5:    CALL NEW
RS55:   ;-----------
        CALL ROTATE
        ;-----------
        CALL SHIFT
        ;-----------
        IN A,(PORTC)
        AND 1FH
        CP 1AH
        JP Z,LAY1
        JP MA55
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ;-----------
            CALL DELAY
            ;-----------
            CALL INITIAL
            ;-----------
            CALL CLEAR
            ;-----------
            LD IY,DTBF5
            CALL DECODE

            CALL NEW
            ;-----------
MA66:       LD D,00H
            LD E,00H
            LD IX,BUFFER
M66:        PUSH IX
            ;-----------
            CALL OPEN
            ;-----------
            CALL CLOSE
            ;-----------
            POP IX
            INC IX
            CALL DEC8
            JR NZ,M66
            LD A,08H
            LD (COUNT8),A
            LD A,(LOOP)
            DEC A
            LD (LOOP),A
            JR NZ,MA66      ; ตรวจสอบจำนวนรอบของการ scan
            LD A,NUMLP
            LD (LOOP),A
            ;-----------
            LD A,(LOOPCH)
            DEC A
            LD (LOOPCH),A   ; ตรวจสอบว่าตัวอักษรถูกเลื่อน
            JR NZ,RS66      ; ไปครบ 8 บิตหรือยัง
            LD A,08H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD (LOOPCH),A
            LD A,(ENDCH)
            CP 01H

            JR NZ,CO66      ; ตรวจสอบรหัสสิ้นสุดไฟล์(1A)
            LD A,(C_ENDCH)
            DEC A
            LD (C_ENDCH),A
            CP 00H          ; ตรวจสอบตัวอักษรตัวสุดท้าย
            JR NZ,TA66      ; ว่าถูกเลื่อนไปครบ 5 หลักหรือยัง
            LD A,05H
            LD (C_ENDCH),A
            XOR A
            LD (ENDCH),A
            LD IY,DTBF4
            JR CO66
TA66:       LD HL,TABLE
            JR CO6
CO66:       CALL DECODE
CO6:        CALL NEW
RS66:       ;-----------
            CALL ROTATE
            ;-----------
            CALL SHIFT
            ;-----------
            IN A,(PORTC)
            AND 1FH
            CP 1AH
            JP Z,LAY1
            JP MA66
        ;---------------;
        ;     MOV3      ;
        ;---------------;
MOV3:       LD B,08H
            CALL MV1
            RET

        ;---------------;
MV1:        PUSH IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD IX,BUFFER
MV:     LD A,(IX+8)
        LD (IY),A
        INC IX
        INC IY
        DJNZ MV
        POP IX
        RET
    ;---------------;
    ;    MOV4       ;
    ;---------------;
MOV4:   LD B,10H
        CALL MV1
        RET
    ;---------------;
    ;    MOV5       ;
    ;---------------;
MOV5:   LD B,18H
        CALL MV1
        RET
    ;---------------;
    ;    KEYIN      ;
    ;---------------;
KEYIN:  PUSH DE
        IN A,(PORTC)
        AND 1FH
        LD D,A
        INC B
        CP 11H
        JR Z,KEY1

        CP 12H
        JR Z,KEY2
        CP 13H
        JR Z,KEY3
        CP 14H
        JR Z,KEY4
        CP 15H
        JR Z,KEY5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CP 16H
        JR Z,KEY6
        CP 17H
        JR Z,KEY7
        CP 18H
        JR Z,KEY8
        CP 19H
        JR Z,KEY9
        CP 1AH
        JR Z,KEY0
        CP 1BH
        JR Z,KEY01
        POP DE
        DEC B
        RET
KEY01:  LD HL,TABLE
        CALL KE1
        CALL K22
        POP DE
        RET
KEY0:   LD HL,T0
        CALL KE1
        CALL K22
        POP DE

        RET
KEY1:   LD HL,T1
        CALL KE1
        CALL K22
        POP DE
        RET
KEY2:   LD HL,T2
        CALL KE1
        CALL K22
        POP DE
        RET
KEY3:   LD HL,T3
        CALL KE1
        CALL K22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP DE
        RET
KEY4:   LD HL,T4
        CALL KE1
        CALL K22
        POP DE
        RET
KEY5:   LD HL,T5
        CALL KE1
        CALL K22
        POP DE
        RET
KEY6:   LD HL,T6
        CALL KE1
        CALL K22
        POP DE
        RET
KEY7:   LD HL,T7
        CALL KE1
        CALL K22
        POP DE
        RET
KEY8:   LD HL,T8
        CALL KE1
        CALL K22
        POP DE
        RET
KEY9:   LD HL,T9
        CALL KE1
        CALL K22
        POP DE
        RET
KEY10:  LD HL,T0
        CALL KE1
        CALL K22
        POP DE
        RET
KEY11:  LD HL,T1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CALL KE1
          CALL K22
          POP DE
          RET
KEY12:    LD HL,T2
          CALL KE1
          CALL K22
          POP DE
          RET
KEY13:    LD HL,T3
          CALL KE1
          CALL K22
          POP DE
          RET
KEY14:    LD HL,T4
          CALL KE1
          CALL K22
          POP DE
          RET
KEY15:    LD HL,T5
          CALL KE1
          CALL K22
          POP DE
          RET
KEY16:    LD HL,T6
          CALL KE1
          CALL K22
          POP DE
          RET
KEY17:    LD HL,T7
          CALL KE1
          CALL K22
          POP DE
          RET
KEY18:    LD HL,T8
          CALL KE1
          CALL K22
          POP DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
KEY19:  LD HL,T9
        CALL KE1
        CALL K22
        POP DE
        RET

K22:    LD B,8H
K9:     LD A,(HL)
        LD (IX),A
        INC HL
        INC IX
        DJNZ K9
        RET
KE1:    PUSH BC
        LD B,08H
KE2:    LD A,(IX)
        DEC IX
        LD (IX),A
        DJNZ KE2
        POP BC
        RET
  ;---------------;
  ;   INKEY2      ;
  ;---------------;
KEYIN2: PUSH DE
        IN A,(PORTC)
        AND 1FH
        LD D,A
        INC B
        CP 1AH
        JR Z,KEY20
        CP 11H
        JR Z,KEY21
        CP 12H
        JR Z,KEY22
        CP 13H
        JR Z,KEY23
        CP 14H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR Z,KEY24
        CP 15H
        JR Z,KEY25
        CP 16H
        JR Z,KEY26
        CP 17H
        JR Z,KEY27
        CP 18H
        JR Z,KEY28
        CP 19H
        JR Z,KEY29
        CP 1BH
        JR Z,KEY02
        POP DE
        DEC B
        RET
KEY20:  LD HL,T0
        CALL K22
        POP DE
        RET
KEY21:  LD HL,T1
        CALL K22
        POP DE
        RET
KEY22:  LD HL,T2
        CALL K22
        POP DE
        RET
KEY23:  LD HL,T3
        CALL K22
        POP DE
        RET

KEY24:  LD HL,T4
        CALL K22
        POP DE
        RET
KEY25:  LD HL,T5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CALL K22
          POP DE
          RET
KEY26:    LD HL,T6
          CALL K22
          POP DE
          RET
KEY27:    LD HL,T7
          CALL K22
          POP DE
          RET
KEY28:    LD HL,T8
          CALL K22
          POP DE
          RET
KEY29:    LD HL,T9
          CALL K22
          POP DE
          RET
KEY02:    LD HL,TABLE
          CALL K22
          POP DE
          RET
    ;---------------;
    ;    INKEY3     ;
    ;---------------;
KEYIN3:   PUSH DE

          IN A,(PORTC)
          AND 1FH
          LD D,A
          INC B
          CP 1AH
          JR Z,KEY30
          CP 11H
          JR Z,KEY31
          CP 12H
          JR Z,KEY32
          CP 13H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR Z,KEY33
        CP 14H
        JR Z,KEY34
        CP 15H
        JR Z,KEY35
        CP 16H
        JR Z,KEY36
        CP 17H
        JR Z,KEY37
        CP 18H
        JR Z,KEY38
        CP 19H
        JR Z,KEY39
        CP 1BH
        JR Z,KEY03
        POP DE
        DEC B
        RET
KEY30:  LD HL,T0
        CALL KE11
        CALL K22

        POP DE
        RET
KEY31:  LD HL,T1
        CALL KE11
        CALL K22
        POP DE
        RET
KEY32:  LD HL,T2
        CALL KE11
        CALL K22
        POP DE
        RET
KEY33:  LD HL,T3
        CALL KE11
        CALL K22
        POP DE
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KEY34:  LD HL,T4
        CALL KE11
        CALL K22
        POP DE
        RET
KEY35:  LD HL,T5
        CALL KE11
        CALL K22
        POP DE
        RET
KEY36:  LD HL,T6
        CALL KE11
        CALL K22
        POP DE
        RET

KEY37:  LD HL,T7
        CALL KE11
        CALL K22
        POP DE
        RET
KEY38:  LD HL,T8
        CALL KE11
        CALL K22
        POP DE
        RET
KEY39:  LD HL,T9
        CALL KE11
        CALL K22
        POP DE
        RET
KEY03:  LD HL,TABLE
        CALL KE11
        CALL K22
        RET
KE11:   PUSH BC
        LD B,08H
        LD IY,BUF1
KE12:   LD A,(IY)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD (IX),A
        INC IY
        INC IX
        DJNZ KE12
        POP BC
        RET
    ;---------------;
    ;    INKEY4     ;
    ;---------------;

KEYIN4: PUSH DE
        IN A,(PORTC)
        AND 1FH
        LD D,A
        CP 1AH
        JR Z,KEY40
        CP 11H
        JR Z,KEY41
        CP 12H
        JR Z,KEY42
        CP 13H
        JR Z,KEY43
        CP 14H
        JR Z,KEY44
        CP 15H
        JR Z,KEY45
        CP 16H
        JR Z,KEY46
        CP 17H
        JR Z,KEY47
        CP 18H
        JR Z,KEY48
        CP 19H
        JR Z,KEY49
        CP 1BH
        JR Z,KEY04
        POP DE
        RET
KEY40:  LD HL,TO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CALL KE14
          CALL K22
          POP DE

          RET
KEY41:    LD HL,T1
          CALL KE14
          CALL K22
          POP DE
          RET
KEY42:    LD HL,T2
          CALL KE14
          CALL K22
          POP DE
          RET
KEY43:    LD HL,T3
          CALL KE14
          CALL K22
          POP DE
          RET
KEY44:     LD HL,T4
          CALL KE14
          CALL K22
          POP DE
          RET
KEY45:    LD HL,T5
          CALL KE14
          CALL K22
          POP DE
          RET
KEY46:    LD HL,T6
          CALL KE14
          CALL K22
          POP DE
          RET
KEY47:    LD HL,T7

          CALL KE14
          CALL K22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          POP DE
          RET
KEY48:    LD HL,T8
          CALL KE14
          CALL K22
          POP DE
          RET
KEY49:    LD HL,T9
          CALL KE14
          CALL K22
          POP DE
          RET
KEY04:    LD HL,TABLE
          CALL KE14
          CALL K22
          POP DE
          RET
KE14:     LD B,10H
          LD IY,BUF1
KE4:      LD A,(IY)
          LD (IX),A
          INC IY
          INC IX
          DJNZ KE4
          RET
     ;---------------;
     ;    INKEY5     ;
     ;---------------;
KEYIN5:   PUSH DE
          IN A,(PORTC)

          INC B.
          CP 1AH
          JR Z,KEY50
          CP 11H
          JR Z,KEY51
          CP 12H
          JR Z,KEY52
          CP 13H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR Z,KEY53
        CP 14H
        JR Z,KEY54
        CP 15H
        JR Z,KEY55
        CP 16H
        JR Z,KEY56
        CP 17H
        JR Z,KEY57
        CP 18H
        JR Z,KEY58
        CP 19H
        JR Z,KEY59
        CP 1BH
        JR Z,KEY05
        POP DE
        DEC B
        RET
KEY05:  LD HL,TABLE
        CALL KE15
        CALL K22
        POP DE
        RET
KEY50:  LD HL,T0

        CALL KE15
        CALL K22
        POP DE
        RET
KEY51:  LD HL,T1
        CALL KE15
        CALL K22
        POP DE
        RET
KEY52:  LD HL,T2
        CALL KE15
        CALL K22
        POP DE
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KEY53:  LD HL,T3
        CALL KE15
        CALL K22
        POP DE
        RET
KEY54:  LD HL,T4
        CALL KE15
        CALL K22
        POP DE
        RET
KEY55:  LD HL,T5
        CALL KE15
        CALL K22
        POP DE
        RET
KEY56:  LD HL,T6
        CALL KE15
        CALL K22
        POP DE
        RET
KEY57:  LD HL,T7
        CALL KE15
        CALL K22
        POP DE
        RET
KEY58:  LD HL,T8
        CALL KE15
        CALL K22
        POP DE
        RET
KEY59:  LD HL,T9
        CALL KE15
        CALL K22
        POP DE
        RET
KE15:   PUSH BC
        LD B,18H
        LD IY,BUF1
KE5:    LD A,(IY)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD (IX),A
            INC IY
            INC IX
            DJNZ KE5
            POP BC
            RET
        ;---------------;
        ;    DALAY      ;
        ;---------------;
DELAY:      PUSH DE
            LD E,0F9H
DDD:        DEC E

            JR NZ,DDD
            POP DE
            RET
;-------------------------; หาตำแหน่งเริ่มต้นของฟอนต์
; DECODE  ASCII -> TABLE ; ตัวอักษรโดยเปลี่ยน data
;-------------------------; ของตัวอักษรจากรหัส ASCII
DECODE:     LD HL,TABLE     ; 1 ไบต์ ไปเป็นฟอนตของตัวอักษร
            LD BC,0008H     ; ขนาด 8 ไบต์ เพื่อที่จะนำ data
            LD A,(IY+0)     ; ขนาด 8 ไบต์นี้มาแสดงผล
            CP 1AH
            JR NZ,D_CODE

            LD A,(ENDCH)
            SET 0,A
            LD (ENDCH),A
            JR RETURN
D_CODE:     ADD HL,BC
            DEC A
            JR NZ,D_CODE
            DEC H
            INC IY
RETURN:     RET
;-------------------------; ทำการส่ง data ที่อยู่ในตำแหน่ง
;           ON_LED        ; display buffer ทั้ง 5 หลัก
;-------------------------; ออกไปยัง LED
OPEN:       PUSH DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
A1:     LD A,D
        OR 0F0H
        OUT (PORTB),A
        LD A,(IX+0)
        OUT (PORTA),A
        LD A,0FFH

        OUT (PORTB),A
        INC D
        CALL IX8
        CALL DEC5
        JR NZ,A1
        LD A,05H
        LD (COUNT5),A
        POP DE
        LD A,D
        AND 0F0H
        OUT (PORTB),A
        ADD A,10H
        LD D,A
        CALL DELAY
        RET
;-------------------------; ทำการ clear data
;       OFF LED           ; ที่ส่งไปยัง LED
;-------------------------;
CLOSE:  PUSH DE
A2:     LD A,E
        OR 0F0H
        OUT (PORTB),A
        LD A,00H
        OUT (PORTA),A
        LD A,0FFH
        OUT (PORTB),A
        INC E
        CALL DEC5
        JR NZ,A2
        LD A,05H
        LD (COUNT5),A
        POP DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD A,E
            AND 0F0H
            OUT (PORTB),A
            ADD A,10H
            LD E,A
            CALL DELAY
            RET
;-------------------------; ทำการหมุนข้อมูลที่อยู่ในตำแหน่ง
;         ROTATE DATA     ; display buffer ทั้ง 6 หลัก
;-------------------------;
ROTATE:     LD IX,BUFFER
R1:         PUSH IX
R2:         LD A,(IX+0)
            RLCA
            LD (IX+0),A
            CALL IX8
            CALL DEC6
            JR NZ,R2
            LD A,06H
            LD (COUNT6),A
            POP IX
            INC IX
            CALL DEC8
            JR NZ,R1
            LD A,08H
            LD (COUNT8),A
            RET
;-------------------------; ทำการเลื่อนข้อมูลที่อยู่ในตำแหน่ง
;         SHIFT DATA      ; display buffer ทั้ง 6 หลัก
;-------------------------; ไปทางซ้าย 1 บิต
SHIFT:      LD IX,BUFFER
S1:         PUSH IX

S2:         RES 0,(IX+0)
            LD A,(IX+8)
            AND 01H
            ADD A,(IX+0)
            LD (IX+0),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL IX8
        CALL DEC5
        JR NZ,S2
        LD A,05H
        LD (COUNT5),A
        POP IX
        INC IX
        CALL DEC8
        JR NZ,S1
        LD A,08H
        LD (COUNT8),A
        RET
;-------------------------;
;    INITIAL VALUE        ; กำหนดค่าเริ่มต้นต่างๆ
;-------------------------;
INITIAL: XOR A
        LD (ENDCH),A
        LD A,05H
        LD (COUNT5),A
        LD (C_ENDCH),A
        LD A,06H
        LD (COUNT6),A
        LD A,NUMLP
        LD (LOOP),A
        LD A,08H
        LD (COUNT8),A
        LD (LOOPCH),A

        RET
;-------------------------; ทำการ clear data ที่อยู่ในตำแหน่ง
;       CLEAR DATA        ; display buffer ทั้ง 5 หลัก
;-------------------------; ให้มีค่าเป็น "0"
CLEAR:  LD B,28H
        LD IX,BUFFER
        LD A,00H
CCC:    LD (IX+0),A
        INC IX
        DJNZ CCC
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------; นำ data ของตัวอักษรตัวใหม่ ที่จะ
;     NEW CHARACTER       ; นำออกมาแสดงผลไปไว้ในตำแหน่ง
;-------------------------; display buffer หลักที่ 6
NEW:    PUSH HL
        LD IX,DISPY
        LD B,08H
X1:     LD A,(HL)
        LD (IX+0),A
        INC HL
        INC IX
        DJNZ X1
        POP HL
        RET
;-------------------------;
;       DALAY             ;
;-------------------------;
DELAY:  PUSH DE
        LD E,0FEH
DDD:    DEC E
        LD D,0FFH
D1:     DEC D

        JR NZ,D1
        JR NZ,DDD
        POP DE
        RET
IX8:    LD BC,0008H       ; เพิ่มค่า IX ขึ้น 8
        ADD IX,BC
        RET
DEC5:   LD A,(COUNT5)     ; ลดค่า count5 ลง 1
        DEC A
        LD (COUNT5),A
        RET
DEC6:   LD A,(COUNT6)     ; ลดค่า count6 ลง 1
        DEC A
        LD (COUNT6),A
        RET
DEC8:   LD A,(COUNT8)     ; ลดค่า count8 ลง 1
        DEC A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD (COUNT8),A
        RET
;-----------------------------------------------;
;      ฟอนต์ตัวอักษร โดยเริ่มจากรหัส ASCII 20H-7FH    ;
;-----------------------------------------------;
TABLE:  DB 00H,00H,00H,00H
        DB 00H,00H,00H,00H    ;

        DB 10H,10H,10H,10H
        DB 00H,00H,10H,00H    ; !

        DB 28H,28H,28H,00H
        DB 00H,00H,00H,00H    ; "

        DB 28H,28H,7CH,28H

        DB 7CH,28H,28H,00H    ; #

        DB 10H,3CH,50H,38H
        DB 14H,78H,10H,00H    ; $

        DB 60H,64H,08H,10H
        DB 20H,4CH,0CH,00H    ; %

        DB 30H,48H,50H,20H
        DB 54H,48H,34H,00H    ; &

        DB 30H,10H,20H,00H
        DB 00H,00H,00H,00H    ; '

        DB 08H,10H,20H,20H
        DB 20H,10H,08H,00H    ; (

        DB 20H,10H,08H,08H
        DB 08H,10H,20H,00H    ; )

        DB 00H,10H,54H,38H
        DB 54H,10H,00H,00H    ; *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 00H,10H,10H,7CH
        DB 10H,10H,00H,00H      ; +

        DB 00H,00H,00H,00H
        DB 30H,10H,20H,00H      ; ,

        DB 00H,00H,00H,7CH
        DB 00H,00H,00H,00H      ; -


        DB 00H,00H,00H,00H
        DB 00H,30H,30H,00H      ; .

        DB 00H,04H,08H,10H
        DB 20H,40H,00H,00H      ; /

T0:     DB 38H,44H,4CH,54H
        DB 64H,44H,38H,00H      ; 0

T1:     DB 10H,30H,10H,10H
        DB 10H,10H,38H,00H      ; 1

T2:     DB 38H,44H,04H,08H
        DB 10H,20H,7CH,00H      ; 2

T3:     DB 7CH,08H,10H,08H
        DB 04H,44H,38H,00H      ; 3

T4:     DB 08H,18H,28H,48H
        DB 7CH,08H,08H,00H      ; 4

T5:     DB 7CH,40H,78H,04H
        DB 04H,44H,38H,00H      ; 5

T6:     DB 18H,20H,40H,78H
        DB 44H,44H,38H,00H      ; 6

T7:     DB 7CH,04H,08H,10H
        DB 20H,20H,20H,00H      ; 7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
T8:     DB 38H,44H,44H,38H
        DB 44H,44H,38H,00H      ; 8

T9:     DB 38H,44H,44H,3CH
        DB 04H,08H,30H,00H      ; 9

        DB 00H,30H,30H,00H
        DB 30H,30H,00H,00H      ; :

        DB 00H,30H,30H,00H
        DB 30H,10H,20H,00H      ; ;

        DB 08H,10H,20H,40H
        DB 20H,10H,08H,00H      ; <

        DB 00H,00H,7CH,00H
        DB 7CH,00H,00H,00H      ; =

        DB 20H,10H,08H,04H
        DB 08H,10H,20H,00H      ; >

        DB 38H,44H,04H,08H
        DB 10H,00H,10H,00H      ; ?

        DB 38H,44H,04H,34H
        DB 54H,54H,38H,00H      ; @

        DB 38H,44H,44H,44H
        DB 7CH,44H,44H,00H      ; A

        DB 78H,44H,44H,78H
        DB 44H,44H,78H,00H      ; B

        DB 38H,44H,40H,40H
        DB 40H,44H,38H,00H      ; C

        DB 70H,48H,44H,44H
        DB 44H,48H,70H,00H      ; D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 7CH,40H,40H,78H
DB 40H,40H,7CH,00H     ; E

DB 7CH,40H,40H,78H
DB 40H,40H,40H,00H     ; F

DB 38H,44H,40H,5CH
DB 44H,44H,3CH,00H     ; G

DB 44H,44H,44H,7CH
DB 44H,44H,44H,00H     ; H

DB 38H,10H,10H,10H
DB 10H,10H,38H,00H     ; I

DB 1CH,08H,08H,08H
DB 08H,48H,30H,00H     ; J

DB 44H,48H,50H,60H
DB 50H,48H,44H,00H     ; K

DB 40H,40H,40H,40H
DB 40H,40H,7CH,00H     ; L

DB 44H,6CH,54H,54H
DB 44H,44H,44H,00H     ; M

DB 44H,44H,64H,54H
DB 4CH,44H,44H,00H     ; N

DB 38H,44H,44H,44H
DB 44H,44H,38H,00H     ; O

DB 78H,44H,44H,78H
DB 40H,40H,40H,00H     ; P

DB 38H,44H,44H,44H
DB 54H,48H,34H,00H     ; Q
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 78H,44H,44H,78H
DB 50H,48H,44H,00H      ; R

DB 3CH,40H,40H,38H
DB 04H,04H,78H,00H      ; S

DB 7CH,10H,10H,10H
DB 10H,10H,10H,00H      ; T

DB 44H,44H,44H,44H
DB 44H,44H,38H,00H      ; U

DB 44H,44H,44H,44H
DB 44H,28H,10H,00H      ; V

DB 44H,44H,44H,54H
DB 54H,54H,28H,00H      ; W

DB 44H,44H,28H,10H
DB 28H,44H,44H,00H      ; X

DB 44H,44H,44H,28H
DB 10H,10H,10H,00H      ; Y

DB 7CH,04H,08H,10H
DB 20H,40H,7CH,00H      ; Z

DB 38H,20H,20H,20H
DB 20H,20H,38H,00H      ; [

DB 40H,20H,10H,08H
DB 04H,02H,01H,00H      ; \

DB 38H,08H,08H,08H
DB 08H,08H,38H,00H      ; ]

DB 10H,28H,44H,00H
DB 00H,00H,00H,00H      ; ^
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 00H,00H,00H,00H
DB 00H,00H,7CH,00H      ; _

DB 20H,10H,08H,00H
DB 00H,00H,00H,00H      ;'

DB 00H,00H,38H,08H
DB 38H,48H,34H,00H      ; a

DB 40H,40H,58H,64H
DB 44H,44H,78H,00H      ; b

DB 00H,00H,3CH,40H
DB 40H,40H,3CH,00H      ; c

DB 04H,04H,34H,4CH
DB 44H,44H,3CH,00H      ; d

DB 00H,00H,38H,44H
DB 78H,40H,38H,00H      ; e

DB 18H,24H,20H,70H
DB 20H,20H,20H,00H      ; f

DB 00H,00H,38H,44H
DB 44H,3CH,04H,38H      ; g

DB 40H,40H,58H,64H
DB 44H,44H,44H,00H      ; h

DB 10H,00H,30H,10H
DB 10H,10H,38H,00H      ; i

DB 00H,08H,00H,08H
DB 08H,28H,10H,00H      ; j

DB 40H,40H,48H,50H
DB 60H,50H,48H,00H      ; k
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB 30H,10H,10H,10H
DB 10H,10H,18H,00H    ; l

DB 00H,00H,68H,54H
DB 54H,54H,44H,00H    ; m

DB 00H,00H,58H,24H
DB 24H,24H,24H,00H    ; n

DB 00H,00H,38H,44H
DB 44H,44H,38H,00H    ; o

DB 00H,00H,70H,48H
DB 48H,70H,40H,40H    ; p

DB 00H,00H,38H,48H
DB 48H,38H,08H,08H    ; q

DB 00H,00H,58H,64H
DB 40H,40H,40H,00H    ; r

DB 00H,00H,38H,40H
DB 38H,04H,78H,00H    ; s

DB 20H,20H,78H,20H
DB 20H,24H,18H,00H    ; t

DB 00H,00H,48H,48H
DB 48H,48H,34H,00H    ; u

DB 00H,00H,44H,44H
DB 28H,28H,10H,00H    ; v

DB 00H,00H,44H,44H
DB 54H,54H,28H,00H    ; w

DB 00H,00H,44H,28H
DB 10H,28H,44H,00H    ; x
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 00H,00H,48H,48H
        DB 48H,38H,08H,70H      ; y

        DB 00H,00H,7CH,08H
        DB 10H,20H,7CH,00H      ; z

        DB 08H,10H,10H,20H
        DB 10H,10H,08H,00H      ; {

        DB 18H,18H,18H,18H
        DB 18H,18H,18H,00H      ; |

        DB 20H,10H,10H,08H
        DB 10H,10H,20H,00H      ; }

        DB 00H,00H,00H,32H
        DB 4CH,00H,00H,00H      ; ~

        DB 0FFH,0FFH,0FFH,0FFH
        DB 0FFH,0FFH,0FFH,0FFH


;-------------------------------------------------------;
;       DATA ที่จะนำออกมาแสดงผล                         ;
;-------------------------------------------------------;
        ORG 9200H
DTBF:   DB 44H,44H,45H,45H
        DB 20H
        DB 4DH,72H,2EH,50H
        DB 52H,41H,44H,49H
        DB 54H,20H,57H,41H
        DB 43H,48H,43H,41H
        DB 52H,11H,50H,49H
        DB 42H,4FH,4FH,4CH
        DB 20H,41H,72H,2EH
        DB 20H,56H ,49H,43H
        DB 48H,41H,49H,20H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 53H,55H, 52H,41H
        DB 50H,41H, 54H,20H
        DB 4DH,72H, 2EH,55H
        DB 41H,54H, 48H,41H
        DB 49H,20H, 53H,52H
        DB 49H,54H,45H,52H
        DB 41H,56H, 49H,52H
        DB 4FH,44H,20H,20H
        DB 4BH,4DH,49H,54H
        DB 4CH,1AH
DTBF1:  DB 50H,4FH,54H,54H
        DB 4FH,4DH,20H,53H
        DB 45H,4CH,45H,43H
        DB 54H,49H,4FH,4EH
        DB 20H
        DB 31H,3DH,4EH,55H
        DB 4DH,45H,52H,41H
        DB 4CH,20H,44H,49H
        DB 53H,50H,4CH,41H
        DB 59H,20H,32H,2CH
        DB 33H,2CH,34H,2CH
        DB 3DH,44H,41H,54H
        DB 41H,20H,44H,49H
        DB 53H,50H,4CH,41H
        DB 59H,1AH
DTBF2:  DB 52H,45H,4DH,4FH
        DB 54H,45H,20H,43H
        DB 4FH,4EH,54H,52H
        DB 4FH,4CH,20H,53H
        DB 43H,4FH,52H,45H
        DB 42H,4FH,41H,52H
        DB 44H,20H,44H,49H

        DB 53H,50H,4CH,41H
        DB 59H,20H,50H,52H
        DB 4FH,4AH ,45H,43H
        DB 54H,20H,4DH,72H
        DB 2DH,42H, 41H,4EH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB 43H,48H, 4FH,43H
        DB 20H,57H, 41H,54H
        DB 54H,41H, 4EH,41H
        DB 53H,55H, 4EH,54H
        DB 48H,4FH,52H,4EH
        DB 20H,4DH, 72H,2DH
        DB 50H,41H,59H,41H
        DB 4EH,20H,42H,4FH
         DB 4EH,59H,41H,53H
        DB 55H,56H, 41H,4EH
        DB 20H,33H,4CH,1AH
DTBF3:  DB 57H,45H,4CH,43H
        DB 4FH,4DH,45H,20H
        DB 54H,4FH,20H,46H
        DB 41H,43H,55H,4CH
        DB 54H,59H,20H,45H
        DB 4EH,47H,49H,4EH
        DB 45H,45H,52H,20H
        DB 4FH,46H,20H,4BH
        DB 4DH,49H,54H,4CH
        DB 1AH
DTBF4:  DB 49H,20H ,57H,49H
        DB 53H,48H,20H,59H
        DB 4FH,55H, 20H,47H
        DB 4FH,4FH, 44H,20H
        DB 4CH,55H, 43H,4BH
        DB 59H,1AH

DTBF5:  DB 2AH,2AH
        DB 54H,48H,41H,4EH
        DB 4BH,20H,59H,4FH
        DB 55H,2AH,2AH,1AH
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. THE Z80 MICROPROCESSOR HARDWAR SOFTWARE PROGRAMMING AND INTERFACING BARRY B.BREY
2. ไมโครโปรเซสเซอร์พื้นฐาน อาจารย์อรรถสิทธิ์ หล่าสกุล
3. ไมโครโปรเซสเซอร์และการออกแบบเบื้องต้น จิติ หนูแก้ว
4. Z80 MICROCOMPUTER SYSTEM ห้องปฏิบัติการโทรคมนาคม กองการฝึกอบรม การสื่อสารแห่งประเทศไทย
5. ET DISPLAY 8*8 FOR ET BOARD VERSION 3.0 ทีม ETT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISO²-CMOS MT8870B/MT8870B-1
Integrated DTMF Receiver

9161-002-051-NA ISSUE 2 December 1987

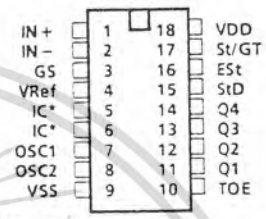
Features

- Complete DTMF Receiver
- Low Power Consumption
- Internal Gain Setting Amplifier
- Adjustable Guard Time
- Central Office Quality

Applications

- Receiver System for British Telecom (BT) or CEPT Spec (MT8870B-1)
- Paging Systems
- Repeater Systems/Mobile Radio
- Credit Card Systems
- Remote Control
- Personal Computers

Pin Connections



* Connect to VSS

Ordering Information -40°C to +85°C

MT8870BE/MT8870BE-1 Plastic DIP
MT8870BC/MT8870BC-1 Cerdip

Description

The MT8870B/MT8870B-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions, fabricated in Mitel's double poly ISO²-CMOS technology. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

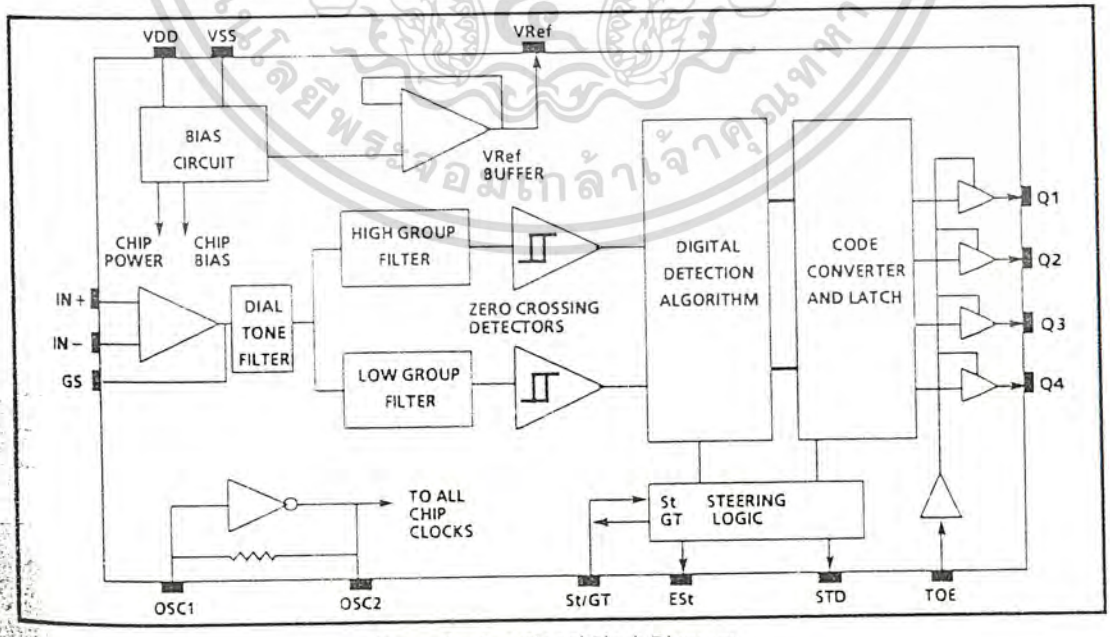


Figure 1 - Functional Block Diagram

Absolute Maximum Ratings*

	Parameter	Symbol	Min	Max	Units
1	Power supply voltage V_{DD}-V_{SS}			6	V
2	Voltage on any pin		$V_{SS}-0.3$	$V_{DD}+0.3$	V
3	Current at any pin (other than supply)			10	mA
4	Operating temperature	T_A	−40	+85	°C
5	Storage temperature		−65	+150	°C
6	Package power dissipation			1000	mW

*Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.
Derate above 75 °C at 16 mW / °C. All leads soldered to board.

Recommended Operating Conditions - Voltages are with respect to ground (V_{SS}) unless otherwise stated

	Characteristics	Sym	Min	Typ†	Max	Units	Test Conditions
1	Positive Supply Voltages	V_{DD}		5		V	$V_{SS}=0V$
2	Oscillator Clock Frequency	fc		3.579545		MHz	
3	Oscillator Frequency Tolerance	Δfc		±0.1		%	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

DC Electrical Characteristics - $V_{DD}=5.0V\pm5\%$, $V_{SS}=0V$. Voltages are with respect to ground (V_{SS}) unless otherwise stated.

		Characteristics	Sym	Min	Typ†	Max	Units	Test Conditions
1	SUPPLY	Operating supply voltage	V_{DD}	4.75	5.0	5.25	V	
2		Operating supply current	I_{DD}		3.0	9.0	mA	
3		Power consumption	P_O		15	45	mW	f = 3.58 MHz; V_{DD} = 5V
4	INPUTS	High level input	V_{IH}	3.5			V	
5		Low level input voltage	V_{IL}			1.5	V	
6		Input leakage current	I_{IH}/I_{IL}		0.1		μA	$V_{IN}=V_{SS}$ or V_{DD}
7		Pull-up (source) current	I_{SO}		7.5	15	μA	TOE (pin 10) = 0V
8		Input impedance (IN+, IN−)	R_{IN}		10		MΩ	@ 1 kHz
9		Steering threshold voltage	V_{TSt}	2.2		2.5	V	
10	OUTPUTS	Low level output voltage	V_{OL}			$V_{SS}+0.03$	V	No load
11		High level output voltage	V_{OH}	V_{DD}-0.03			V	No load
12		Output low (sink) current	I_{OL}	1	2.5		mA	$V_{OUT}=0.4$ V
13		Output high (source) current	I_{OH}	0.4	0.8		mA	$V_{OUT}=4.6$ V
14		V_{Ref} output voltage	V_{Ref}	2.4		2.7	V	No load
15		V_{Ref} output resistance	R_{OR}		10		kΩ	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

Operating Characteristics[1] - Voltages are with respect to ground (V_{SS}) unless otherwise stated

Gain Setting Amplifier

	Characteristics	Sym	Min	Typ[2]	Max	Units	Test Conditions
1	Input leakage current	I_{IN}		100		nA	$V_{SS} \leq V_{IN} \leq V_{DD}$
2	Input resistance	R_{IN}		10		MΩ	
3	Input offset voltage	V_{OS}		25		mV	
4	Power supply rejection	PSRR		60		dB	1 kHz
5	Common mode rejection	CMRR		60		dB	$-3.0V \leq V_{IN} \leq 3.0V$
6	DC open loop voltage gain	A_{VOL}		65		dB	
7	Open loop unity gain bandwidth	f_C		1.5		MHz	
8	Output voltage swing	V_O		4.5		V_{pp}	$R_L \geq 100K\Omega$ to V_{SS}
9	Maximum capacitive load (GS)	C_L		100		pF	
10	Maximum resistive load (GS)	R_L		50		KΩ	
11	Common mode range	V_{CM}		3.0		V_{pp}	No Load

[1] $V_{DD} = 5$ V, $V_{SS} = 0$ V, $T_A = 25°C$

[2] Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

MT8870B AC Electrical Characteristics[1] - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ	Max	Units	Notes
1	SIGNAL CONDITIONS	Valid input signal levels (each tone of composite signal)		-29			dBm	1,2,3,5,6,9
				27.5			mV_{RMS}	1,2,3,5,6,9
						+1	dBm	1,2,3,5,6,9
						869	mV_{RMS}	1,2,3,5,6,9
2		Positive twist accept			10		dB	2,3,6,9
3		Negative twist accept			10		dB	2,3,6,9
4		Freq. deviation accept		± 1.5% ± 2Hz			Nom.	2,3,5,9
5		Freq. deviation reject		± 3.5%			Nom.	2,3,5,9
6		Third tone tolerance			-16		dB	2,3,4,5,9
7		Noise tolerance			-12		dB	2,3,4,5,7,9,10
8		Dial tone tolerance			+22		dB	2,3,4,5,8,9,11

[1] $V_{DD} = 5$ V, $V_{SS} = 0$, $T_A = 25°C$ and $f_C = 3.579545$ MHz using test circuit shown in Figure 2

NOTES

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all DTMF tones
3. Tone duration = 40 ms, tone pause = 40 ms
4. Signal condition consists of nominal DTMF frequencies
5. Both tones in composite signal have an equal amplitude.
6. Tone pair is deviated by ± 1.5% ± 2Hz.
7. Bandwidth limited (3KHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2%
9. For an error rate of better than 1 in 10,000
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. For guard time calculation purposes.

MT8870B-1 AC Electrical Characteristics* - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ	Max	Units	Notes
1	SIGNAL CONDITIONS	Valid input signal levels (each tone of composite signal)		-31			dBm	1,2,3,5,6,9
				21.8			mV_{RMS}	1,2,3,5,6,9
						+1	dBm	1,2,3,5,6,9
						869	mV_{RMS}	1,2,3,5,6,9
2		Input Signal Level Reject		-37			dBm	1,2,3,5,6,9
				10.9			mV_{RMS}	1,2,3,5,6,9
3		Positive twist accept				6	dB	2,3,6,9
4		Negative twist accept				6	dB	2,3,6,9
5		Freq. deviation accept		±1.5% ± 2Hz				2,3,5,9
6		Freq. deviation reject		±3.5%				2,3,5,9
7		Third tone tolerance		-18.5			dB	2,3,4,5,9,13
8		Noise tolerance			-12		dB	2,3,4,5,7,9,10
9		Dial tone tolerance			+22		dB	2,3,4,5,8,9,11

* V_{DD} = 5 V, V_{SS} = 0, T_A = 25° C and f_C = 3.579545 MHz using test circuit shown in Figure 2

NOTES

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all DTMF tones
3. Tone duration = 40 ms, tone pause = 40 ms.
4. Signal condition consists of nominal DTMF frequencies
5. Both tones in composite signal have an equal amplitude.
6. Tone pair is deviated by ±1.5% ± 2Hz.
7. Bandwidth limited (3 kHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %
9. For an error rate of better than 1 in 10,000.
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. For guard time calculation purposes.
13. Referenced to Fig. 10 Input DTMF Tone Level at -25 dBm (-28 dBm at GS Pin) Interference Frequency Range between 480-3400 Hz.

AC Electrical Characteristics - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ‡	Max	Units	Conditions
1	TIMING	Tone present detect time	t_{DP}	5	11	14	ms	Note 12
2		Tone absent detect time	t_{DA}	0.5	4	8.5	ms	Note 12
3		Tone duration accept	t_{REC}			40	ms	User adjustable
4		Tone duration reject	$t_{\overline{REC}}$	20			ms	User adjustable
5		Interdigit pause accept	t_{ID}			40	ms	User adjustable
6		Interdigit pause reject	t_{DO}	20			ms	User adjustable
7	OUTPUTS	Propagation delay (St to Q)	t_{PQ}		8	11	µs	TOE = V_{DD}
8		Propagation delay (St to StD)	t_{PStD}		12		µs	TOE = V_{DD}
9		Output data setup (Q to StD)	t_{QStD}		3.4		µs	TOE = V_{DD}
10		Propagation delay (TOE to Q ENABLE)	t_{PTE}		50		ns	R_L = 10kΩ C_L = 50 pF
11		Propagation delay (TOE to Q DISABLE)	t_{PTD}		300		ns	R_L = 10kΩ C_L = 50 pF
12	CLOCK	Crystal /clock frequency	f_C	3.5759	3.5795	3.5831	MHz	
13		Clock input rise time	t_{LHCL}			110	ns	Ext. clock
14		Clock input fall time	t_{HLCL}			110	ns	Ext. clock
15		Clock input duty cycle	DC_{CL}	40	50	60	%	Ext. clock
16		Capacitive load (OSC2)	C_{LO}			30	pF	

† V_{DD} = 5.0V, V_{SS} = 0V, T_A = 25°C and f_c = 3.579545 MHz, using test circuit shown in Figure 2.

‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

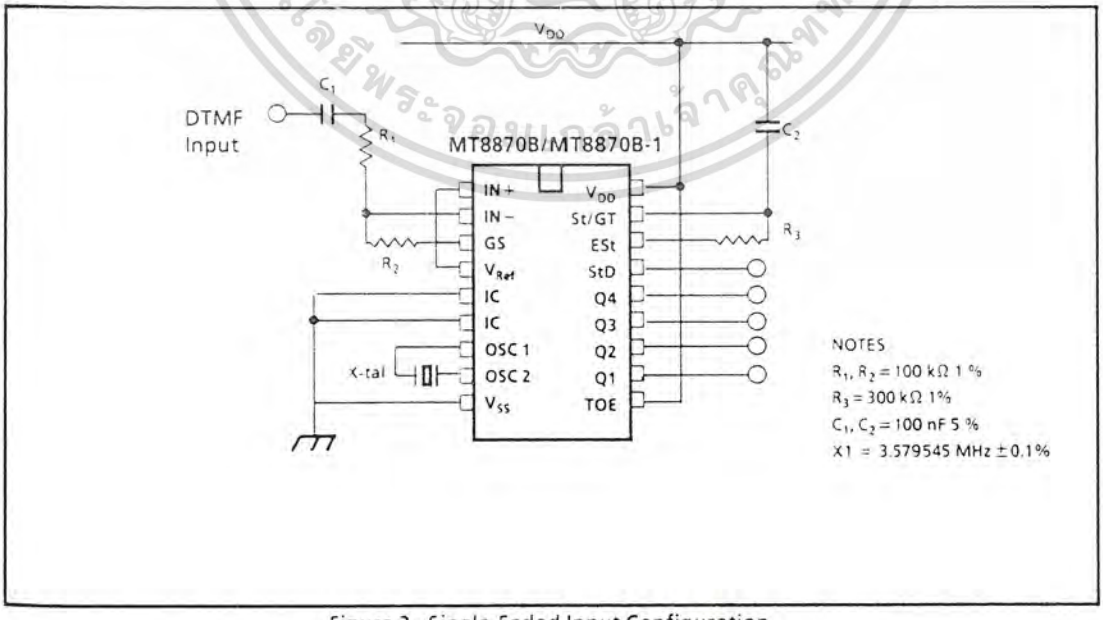
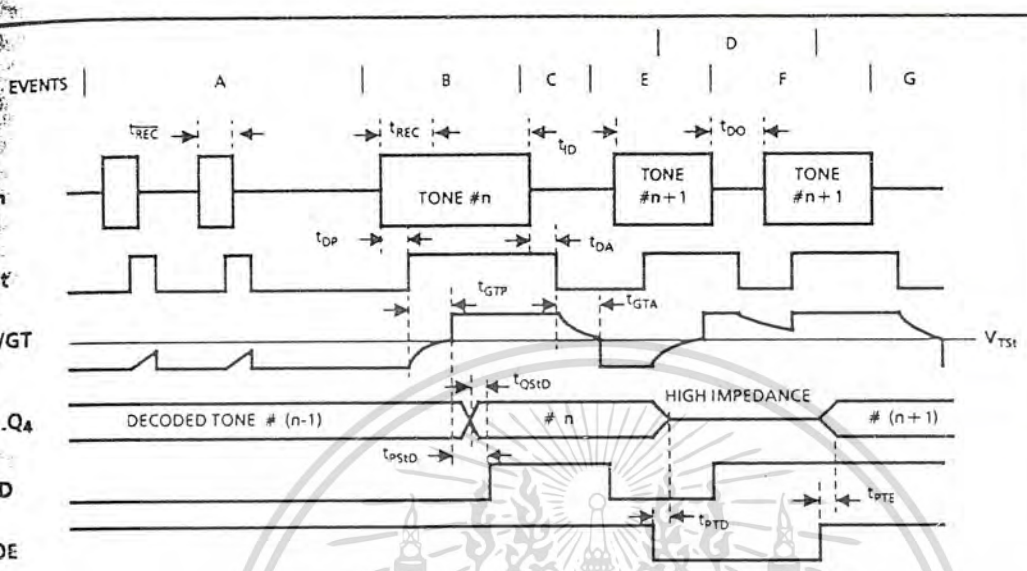


Figure 2 - Single-Ended Input Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Pin Description

Pin #	Name	Description
1	IN+	Non-Inverting Op-Amp (Input).
2	IN−	Inverting Op-Amp (Input).
3	GS	Gain Select. Gives access to output of front end differential amplifier for connection of feedback resistor.
4	V_{Ref}	Reference Voltage (Output). Nominally $V_{DD}/2$ is used to bias inputs at mid-rail (see Fig.2).
5	IC	Internal Connection. Must be tied to V_{SS}.
6	IC	Internal Connection. Must be tied to V_{SS}.
7	OSC1	Clock (Input).
8	OSC2	Clock (Output). A 3.579545 MHz crystal connected between pins OSC1 and OSC2 completes the internal oscillator circuit.
9	V_{SS}	Negative Power Supply (Input).
10	TOE	Three State Output Enable (Input). Logic high enables the outputs Q1-Q4. This pin is pulled up internally.
11-14	Q1-Q4	Three State Data (Output). When enabled by TOE, provide the code corresponding to the last valid tone-pair received (see Table 1). When TOE is logic low, the data outputs are high impedance.
15	StD	Delayed Steering (Output). Presents a logic high when a received tone-pair has been registered and the output latch updated; returns to logic low when the voltage on St/GT falls below V_{TSt}.
16	ESt	Early Steering (Output). Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause ESt to return to a logic low.
17	St/GT	Steering Input/Guard time (Output) Bidirectional. A voltage greater than V_{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{TSt} frees the device to accept a new tone pair. The GT output acts to reset the external steering time-constant; its state is a function of ESt and the voltage on St.
18	V_{DD}	Positive power supply (Input).



EXPLANATION OF EVENTS

A) TONE BURSTS DETECTED, TONE DURATION INVALID, OUTPUTS NOT UPDATED.
B) TONE #n DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS.
C) END OF TONE #n DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.
D) OUTPUTS SWITCHED TO HIGH IMPEDANCE STATE.
E) TONE #n+1 DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS (CURRENTLY HIGH IMPEDANCE).
F) ACCEPTABLE DROPOUT OF TONE #n+1, TONE ABSENT DURATION INVALID, OUTPUTS REMAIN LATCHED.
G) END OF TONE #n+1 DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.

EXPLANATION OF SYMBOLS

V_{in} DTMF COMPOSITE INPUT SIGNAL.
ESt EARLY STEERING OUTPUT. INDICATES DETECTION OF VALID TONE FREQUENCIES.
St/GT STEERING INPUT/GUARD TIME OUTPUT. DRIVES EXTERNAL RC TIMING CIRCUIT.
Q_1-Q_4 4-BIT DECODED TONE OUTPUT.
StD DELAYED STEERING OUTPUT. INDICATES THAT VALID FREQUENCIES HAVE BEEN PRESENT/ABSENT FOR THE REQUIRED GUARD TIME THUS CONSTITUTING A VALID SIGNAL.
TOE TONE OUTPUT ENABLE (INPUT). A LOW LEVEL SHIFTS Q_1-Q_4 TO ITS HIGH IMPEDANCE STATE.

$t_{\overline{REC}}$ MAXIMUM DTMF SIGNAL DURATION NOT DETECTED AS VALID.
t_{REC} MINIMUM DTMF SIGNAL DURATION REQUIRED FOR VALID RECOGNITION.
t_{ID} MINIMUM TIME BETWEEN VALID DTMF SIGNALS.
t_{DO} MAXIMUM ALLOWABLE DROPOUT DURING VALID DTMF SIGNAL.
t_{DP} TIME TO DETECT THE PRESENCE OF VALID DTMF SIGNALS.
t_{DA} TIME TO DETECT THE ABSENCE OF VALID DTMF SIGNALS.
t_{GTP} GUARD TIME, TONE PRESENT
t_{GTA} GUARD TIME, TONE ABSENT

Figure 3- Timing Diagram

Functional Description

The MT8870B/MT8870B-1 monolithic DTMF receiver offers small size, low power consumption and high performance. Its architecture consists of a bandsplit filter section, which separates the high and low group tones, followed by a digital counting section which verifies the frequency and duration of the received tones before passing the corresponding code to the output bus.

Filter Section

Separation of the low group and high group tones is achieved by applying the DTMF signal to the inputs of two sixth-order switched capacitor bandpass filters, the bandwidths of which correspond to the low and high group frequencies. The filter section also incorporates notches at 350 and 440 Hz for exceptional dial tone rejection (see Figure 4). Each filter output is followed by a single order switched capacitor filter section which smooths the signals prior to limiting. Limiting is performed by high-gain comparators which are provided with hysteresis to prevent detection of unwanted low-level signals. The outputs of the comparators provide full rail logic swings at the frequencies of the incoming DTMF signals.

Decoder Section

Following the filter section is a decoder employing digital counting techniques to determine the frequencies of the incoming tones and to verify that they correspond to standard DTMF frequencies. A complex averaging algorithm protects against tone simulation by extraneous signals such as voice while providing tolerance to small frequency deviations and variations. This averaging algorithm has been developed to ensure an optimum combination of immunity to talk-off and tolerance to the presence of interfering frequencies (third tones) and noise. When the detector recognizes the presence of two valid tones (this is referred to as the "signal condition" in some industry specifications) the "Early Steering" (ESt) output will go to an active state. Any subsequent loss of signal condition will cause ESt to assume an inactive state (see "Steering Circuit").

Steering Circuit

Before registration of a decoded tone pair, the receiver checks for a valid signal duration (referred to as character recognition condition). This check is performed by an external RC time constant driven by ESt. A logic high on ESt causes v_c (see Figure 5) to rise as the capacitor discharges. Provided signal condition is maintained (ESt remains high) for the validation period (t_{GTP}), v_c reaches the threshold (V_{TSt}) of the steering logic to register the tone pair, latching its corresponding 4-bit code (see Table 1) into the output latch. At this point the GT output is activated and drives v_c to V_{DD}. GT continues to drive high as long as ESt remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (StD) goes high, signalling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the three state control input (TOE) to a logic high. The steering circuit works in reverse to validate the interdigit

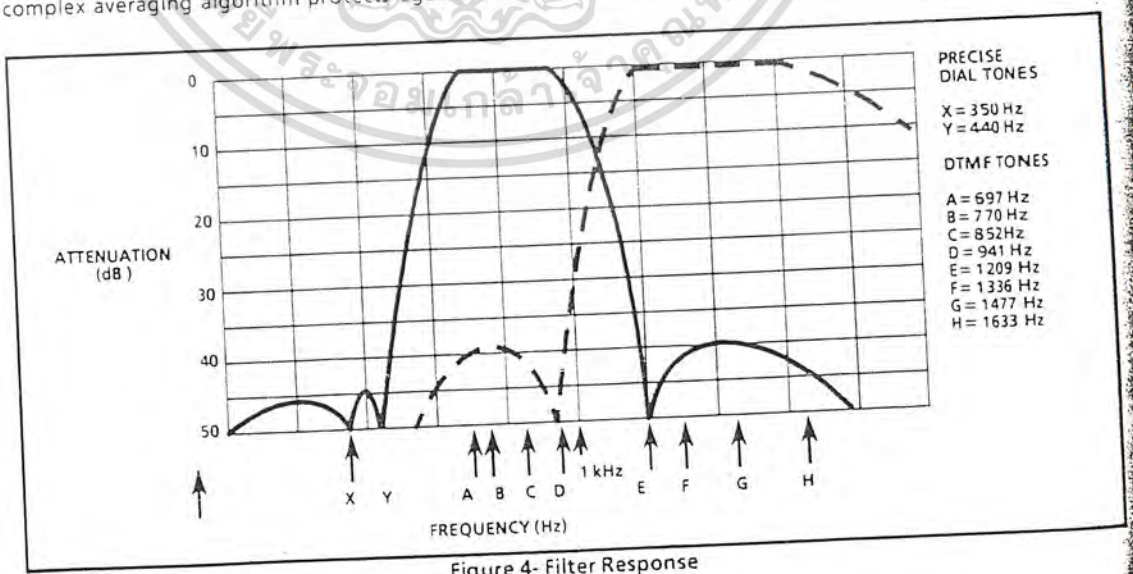


Figure 4- Filter Response

F_{LOW}	F_{HIGH}	NO.	TOE	Q_4	Q_3	Q_2	Q_1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

L = LOGIC LOW, H = LOGIC HIGH, Z = HIGH IMPEDANCE

Table 1. Functional Decode Table

pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (dropout) too short to be considered a valid pause. This facility, together with the capability of selecting the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

Guard Time Adjustment

In many situations not requiring selection of tone duration and interdigital pause, the simple steering circuit shown in Figure 5 is applicable. Component values are chosen according to the formula:

$$t_{REC} = t_{DP} + t_{GTP}$$
$$t_{ID} = t_{DA} + t_{GTA}$$

The value of t_{DP} is a device parameter (see Figure 3) and t_{REC} is the minimum signal duration to be recognized by the receiver. A value for C of 0.1 μF is recommended for most applications, leaving R to be selected by the designer.

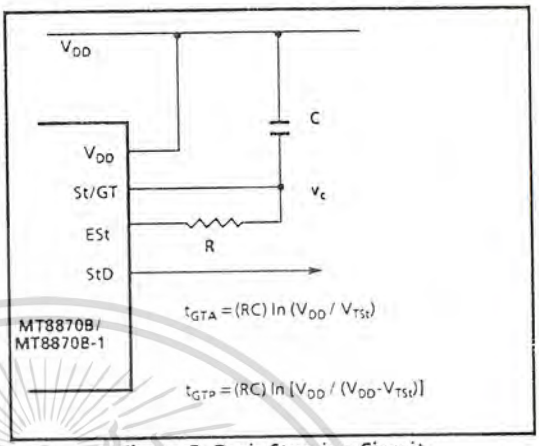


Figure 5- Basic Steering Circuit

Different steering arrangements may be used to select independently the guard times for tone present (t_{GTP}) and tone absent (t_{GTA}). This may be necessary to meet system specifications which place both accept and reject limits on both tone duration and interdigital pause. Guard time adjustment also allows the designer to tailor system parameters such as talk off and noise immunity. Increasing t_{REC} improves talk-off performance since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. Alternatively, a relatively short t_{REC} with a long t_{DO} would be appropriate for extremely noisy environments where fast acquisition time and immunity to tone dropouts are required. Design information for guard time adjustment is shown in Figure 6.

Differential Input Configuration

The input arrangement of the MT8870B/MT8870B-1 provides a differential-input operational amplifier as well as a bias source(V_{Ref}) which is used to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the op-amp output (GS) for adjustment of gain. In a single-ended configuration, the input pins are connected as shown in Figure 2 with the op-amp connected for unity gain and V_{Ref} biasing the input at $\frac{1}{2}V_{DD}$. Figure 7 shows the differential configuration, which permits the adjustment of gain with the feedback resistor R_5.

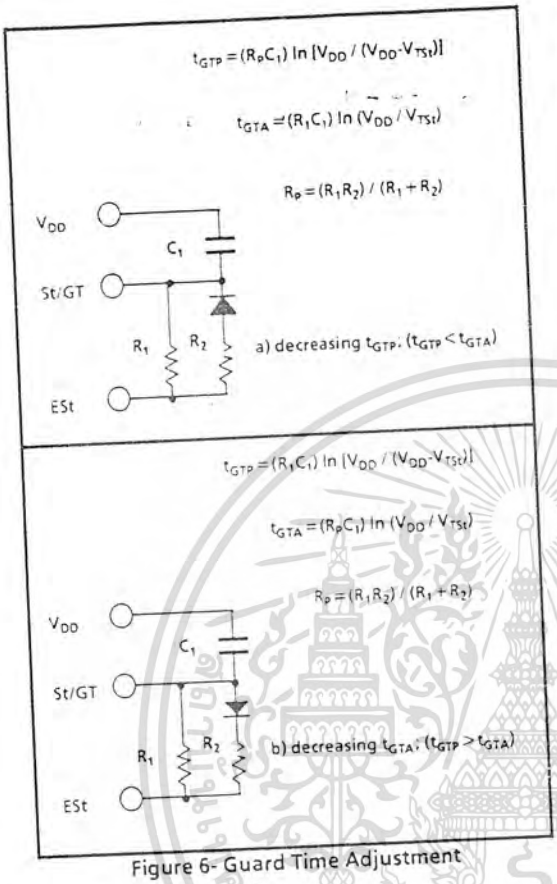


Figure 6- Guard Time Adjustment

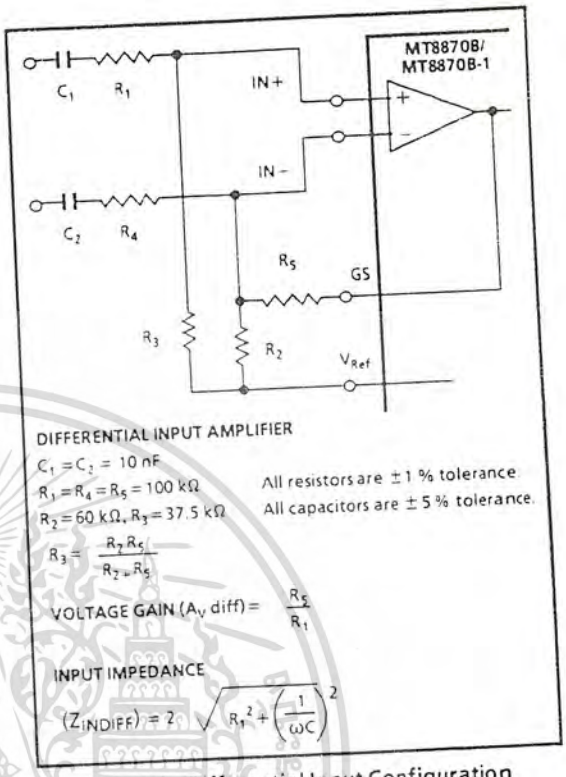


Figure 7- Differential Input Configuration

Crystal Oscillator

The internal clock circuit is completed with the addition of an external 3.579545 MHz crystal and is normally connected as shown in Figure 2 (Single Ended Input Configuration). However, it is possible to configure several MT8870B/MT8870B-1 devices employing only a single oscillator crystal. The oscillator output of the first device in the chain is coupled through a 30 pF capacitor to the oscillator input (OSC1) of the next device. Subsequent devices are connected in a similar fashion. Refer to Figure 8 for details. The problems associated with unbalanced loading are not a concern with the arrangement shown, ie; precision balancing capacitors are not required.

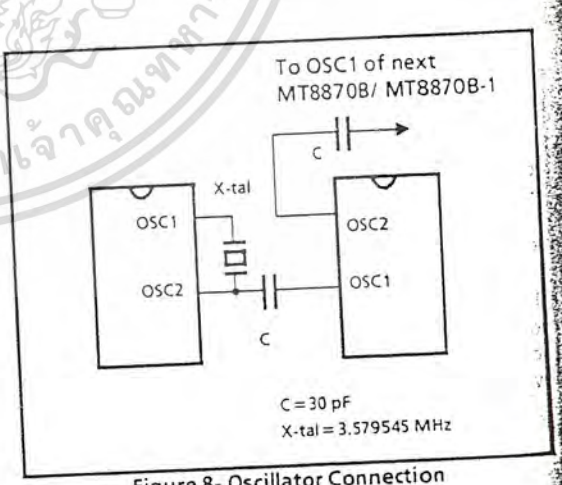


Figure 8- Oscillator Connection

ISO²-CMOS MT8870C/MT8870C-1
Integrated DTMF Receiver

9161-002-094-NA ISSUE 1 November 1990

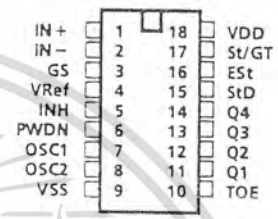
Features

- Complete DTMF Receiver
- Low Power Consumption
- Internal Gain Setting Amplifier
- Adjustable Guard Time
- Central Office Quality
- Power-down Mode
- Inhibit Mode

Applications

- Receiver System for British Telecom (BT) or CEPT Spec (MT8870C-1)
- Paging Systems
- Repeater Systems/Mobile Radio
- Credit Card Systems
- Remote Control
- Personal Computers
- Telephone Answering Machine

Pin Connections



Ordering Information

MT8870CE/MT8870CE-1	Plastic DIP
MT8870CC/MT8870CC-1	Cerdip
MT8870CS	SOIC
	−40 °C to +85 °C

Description

The MT8870C/MT8870C-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions, fabricated in Mitel's double poly ISO²-CMOS technology. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone-pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

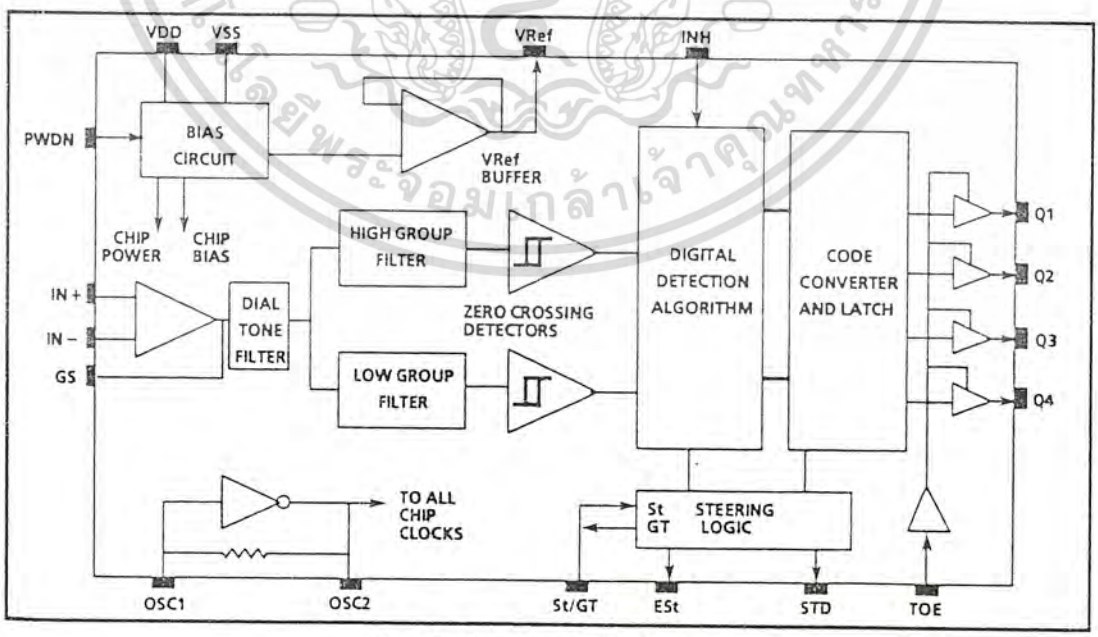


Figure 1 - Functional Block Diagram

APPLICATION

RECEIVER SYSTEM FOR BRITISH TELECOM SPEC POR 1151

The circuit shown in Fig. 10 illustrates the use of MT8870B-1 device in a typical receiver system. BT Spec defines the input signals less than -34 dBm as the non-operate level. This condition can be attained by choosing a suitable values of R_1 and R_2 to provide 3 dB attenuation, such that -34 dBm input signal will correspond to -37 dBm at the gain setting pin GS of MT8870B-1. As shown in the diagram, the component values of R_3 and C_2 are the guard time requirements when the total component tolerance is 6%. For better performance, it is recommended to use the non-symmetric guard time circuit in Fig. 9.

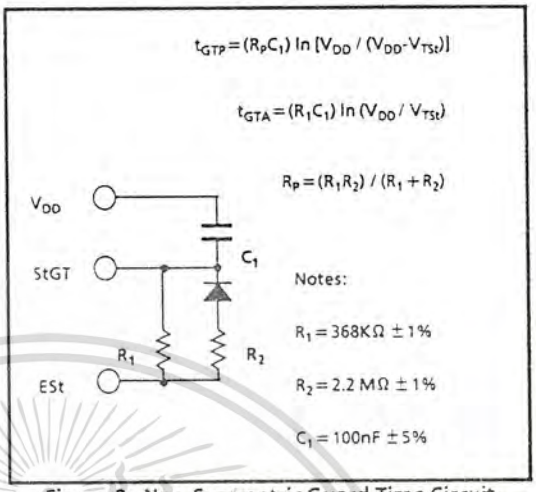


Figure 9 - Non-Symmetric Guard Time Circuit

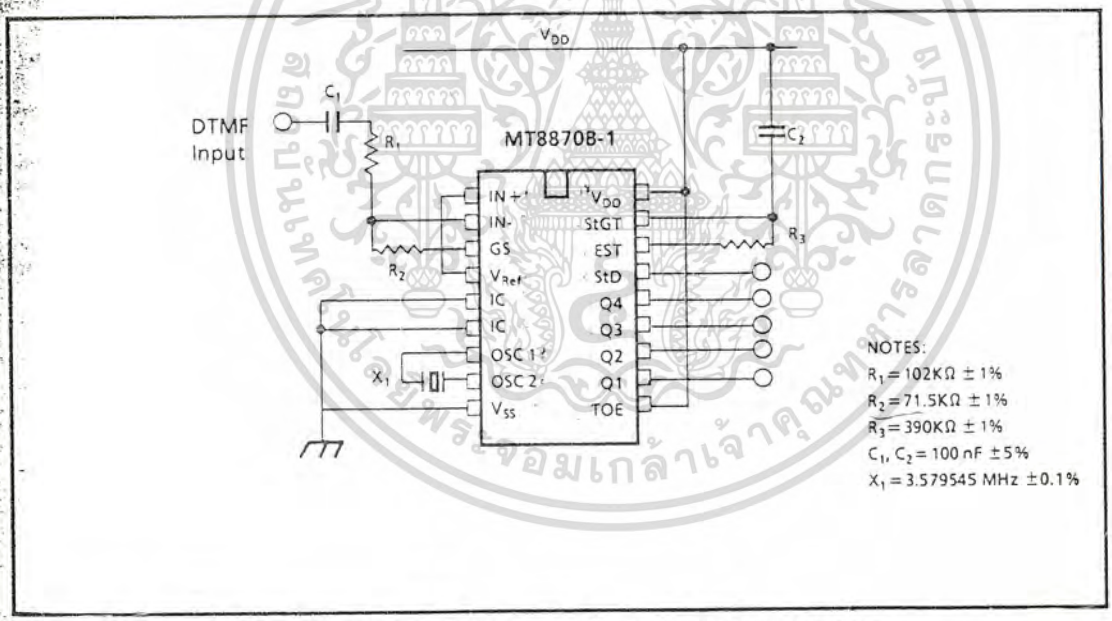


Figure 10 - Single-Ended Input Configuration for BT or CEPT Spec

UMC

UM82C55A

CMOS Programmable Peripheral Interface

Features

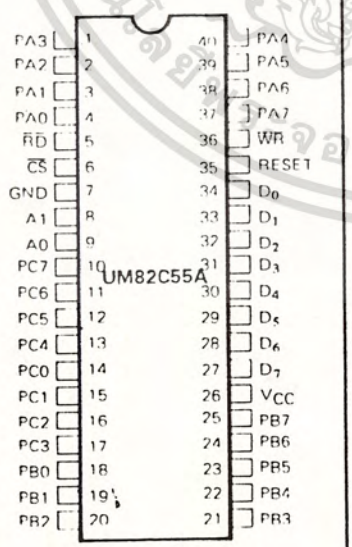
- Pin compatible with NMOS 8255A
- 24 programmable I/O pins
- Fully TTL compatible
- Bus hold circuitry on all I/O ports eliminates pull up resistors
- High speed, no "wait state" operation with 8MHz 80C86
- Direct bit set/reset capability
- Enhanced control word read capability
- Single 5V power supply
- 2.5mA drive capability on all I/O port outputs
- Low standby power – I_{CCSB} = 10μA

General Description

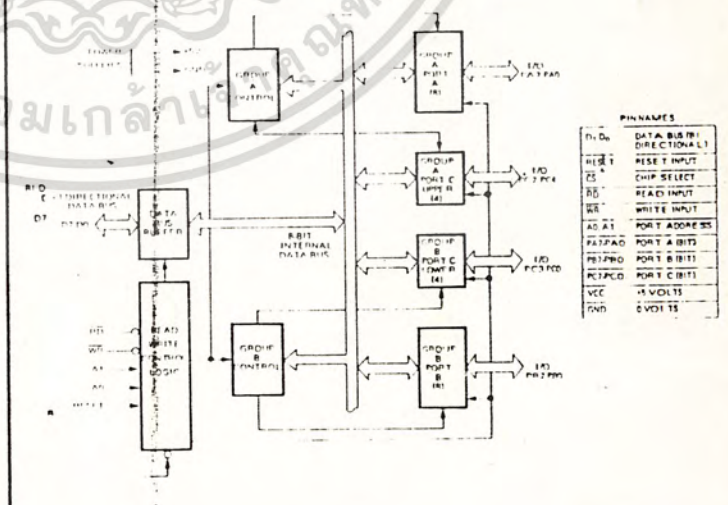
The UM82C55A is a high performance CMOS version of the industry standard 8255A and is manufactured using a selfaligned silicon gate CMOS process. It is a general purpose programmable I/O device which may be used with many different microprocessors. There are 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The high performance of the UM82C55A make it compatible with microprocessors such as the 8086, 8048, 8051.

Static CMOS circuit design insures low operating power. TTL compatibility of V_{IH}=2.0 volts over the industrial temperature range and bus hold circuitry eliminate the need for pull up resistors.

Pin Configuration



Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Absolute Maximum Ratings*

Supply Voltage	+8.0 VOLTS
Operating Voltage Range	+4V to +7V
Input Voltage Applied	GND -2.0V to 6.5V
I/O Pin Voltage Applied	GND-0.5V to VCC+0.5V
Storage Temperature Range	-65°C to +150°C
Operating Temperature Range	0°C to +70°C
Maximum Power Dissipation	1 Watt

*Comments

Stresses above those listed in the "ABSOLUTE MAXIMUM RATINGS" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. Electrical Characteristics

($VCC = 5.0V +/- 5\%$, $T_A = 0°C$ to $+70°C$)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IH}	Logical One Input Voltage	2.0		V	
V_{IL}	Logical Zero Input Voltage		0.8	V	
V_{OH}	Logical One Output Voltage	3.0 VCC -0.4		V V	I_{OH} = -2.5mA I_{OH} = -100 μA
V_{OL}	Logical Zero Output Voltage		0.4	V	I_{OL} = +2.5 mA
I_{IL}	Input Leakage Current	-1.0	1.0	μA	$0V \leq V_{IN} \leq V_{CC}$
I_O	I/O Pin Leakage Current	-10.0	10.0	μA	$0V \leq V_O \leq V_{CC}$
I_{BHH}	Bus Hold High Leakage Current	-50	-300	μA	V_O = 3.0V Ports A, B, C
I_{BHL}	Bus Hold Low Leakage Current	+50	+300	μA	V_O = 1.0V Port A only
I_{DAR}	Darlington Drive Current	-2.0		mA	Ports A, B, C Test Condition 3
I_{CC}	Power Supply Current		10	μA	V_{CC} = 5.5V $V_{IN} = V_{CC}$ or GND Outputs Open

Capacitance

($T_A = 25°C$; $V_{CC} = GND = 0V$; $V_{IN} = +5V$ or GND)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
C_{IN}*	Input Capacitance		5	pF	FREQ = 1 MHZ Unmeasured Pins Returned to GND
$C_{I/O}$*	I/O Pin Capacitance		20	pF	

*Guaranteed and sampled, but not 100% tested

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Characteristics

(V_{CC} = +5V ±5%, GND = 0V; T_A = 0°C to +70°C)

READ

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AR}	Address Stable Before READ	0		ns	
t_{RA}	Address Stable After READ	0		ns	
t_{RR}	READ Pulse Width	150		ns	
t_{RD}	Data Valid From READ		100	ns	1
t_{DF}	Data Float After READ	10	75	ns	2
t_{RV}	Time Between READs and/or WRITEs	300		ns	

WRITE

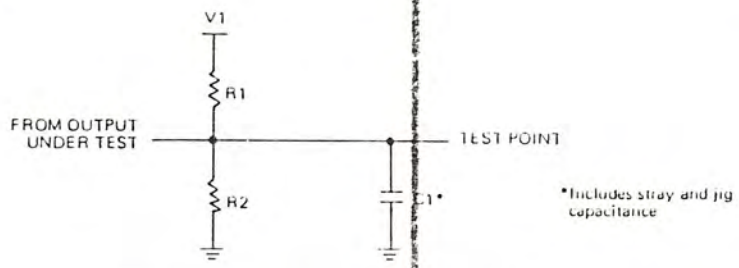
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Stable Before WRITE	0		ns	
t_{WA}	Address Stable After WRITE	20		ns	Ports A & B
		60		ns	Port C
t_{WW}	WRITE Pulse Width	100		ns	
t_{DW}	Data Valid to WRITE High	100		ns	
t_{WD}	Data Valid After WRITE High	30		ns	Ports A & B
		60		ns	Port C

OTHER TIMINGS

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{WB}	WR = 1 to Output		350	ns	1
t_{IR}	Peripheral Data Before RD	0		ns	
t_{HR}	Peripheral Data After RD	0		ns	
t_{AK}	ACK Pulse Width	100		ns	
t_{ST}	STB Pulse Width	100		ns	
t_{PS}	Per. Data Before STB High	20		ns	
t_{PH}	Per. Data After STB High	50		ns	
t_{AD}	ACK = 0 to Output		175	ns	1
t_{KD}	ACK = 1 to Output Float	20	250	ns	2
t_{WOB}	WR = 1 to OBF = 0		150	ns	1
t_{AOB}	ACK = 0 to OBF = 1		150	ns	1
t_{SIB}	STB = 0 to IBF = 1		150	ns	1
t_{RIB}	RD = 1 to IBF = 0		150	ns	1
t_{RIT}	RD = 0 to INTR = 0		200	ns	1
t_{SIT}	STB = 1 to INTR = 1		150	ns	1
t_{AIT}	ACK = 1 to INTR = 1		150	ns	1
t_{WIT}	WR = 0 to INTR = 0		200	ns	1
t_{RES}	Reset Pulse Width	500		ns	see note 1

Note: Period of initial Reset pulse after power on must be at least 50usec. Subsequent Reset pulses may be 500 ns minimum.

A.C. Test Circuits

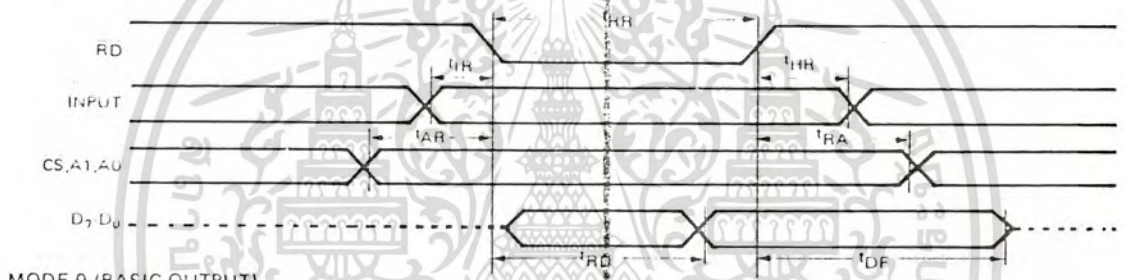


Test Condition	V1	R1	R2	C1
1	1.7V	523Ω	Open	150 pf
2	5.0V	2kΩ	1.7kΩ	50 pf
3	1.5V	750Ω	Open	Open

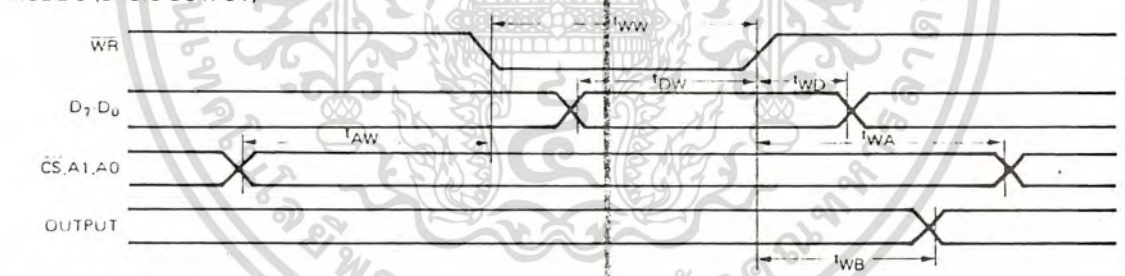
TEST CONDITION DEFINITION TABLE

Waveforms

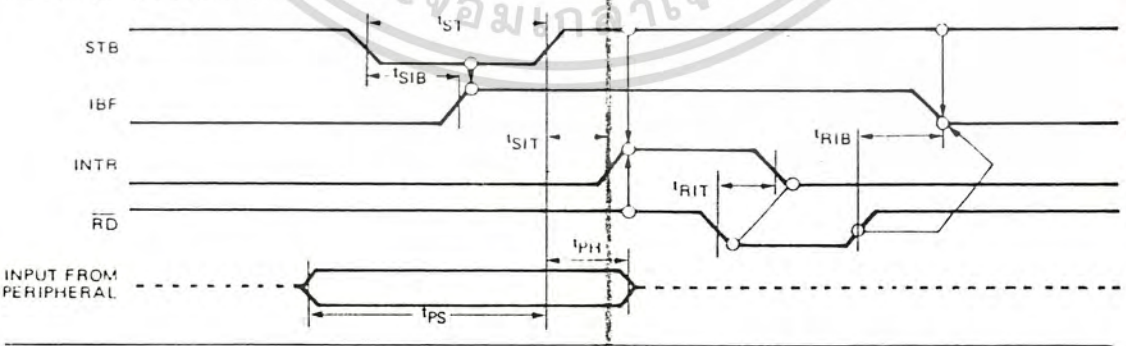
MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)



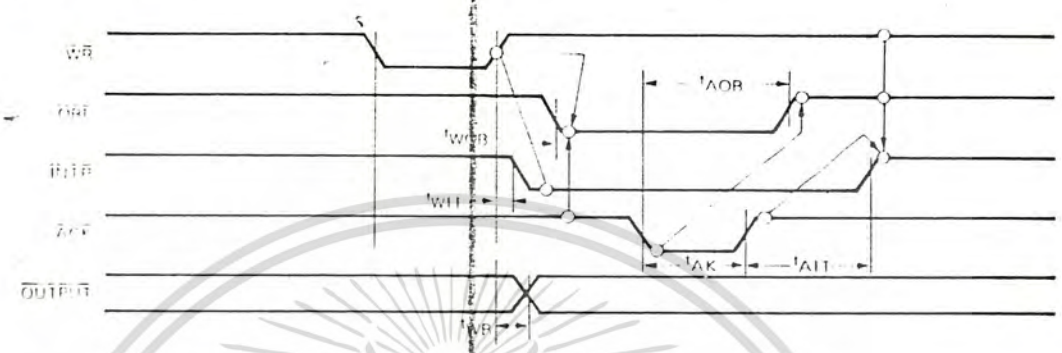
MODE 1 (STROBED INPUT)



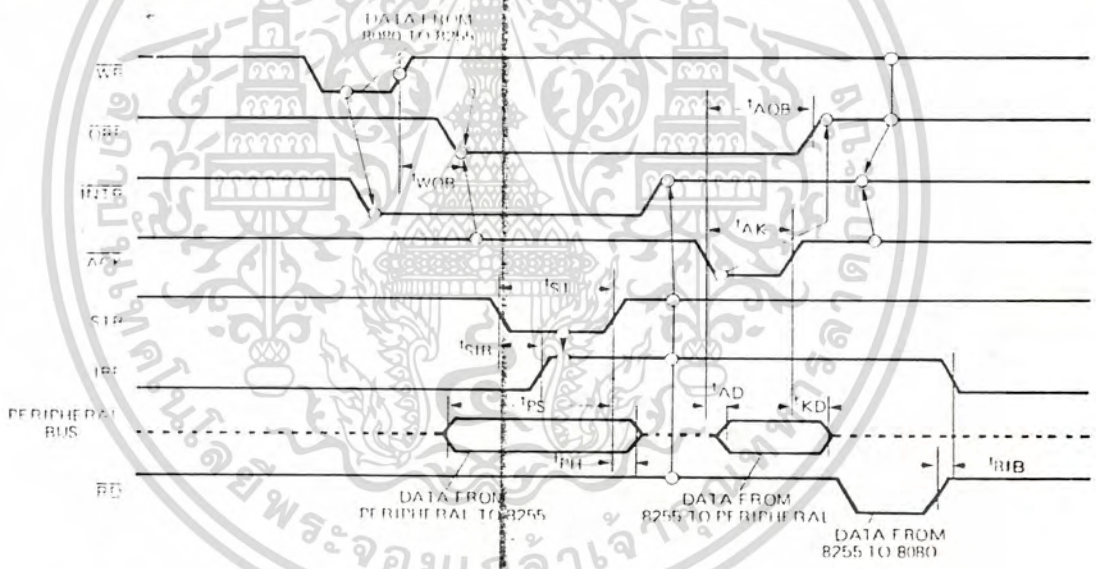
General Purpose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

MODE 1 (Strobe Output)

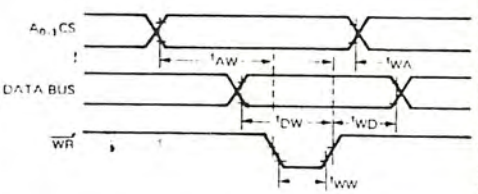


MODE 2 (BIDIRECTIONAL)

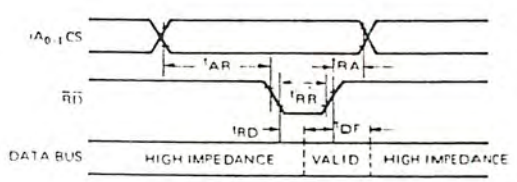


Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible. ($INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + \overline{OBF} \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR}$)

WRITE TIMING



READ TIMING



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Pin Description

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the UM82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

(CS)

Chip Select. A "low" on this input pin enables the communication between the UM82C55A and the CPU.

(RD)

Read. A "low" on this input pin enables the UM82C55A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the UM82C55A.

(WR)

Write. A "low" on this input pin enables the CPU to write data or control words into the UM82C55A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

UM82C55A BASIC OPERATION

A	A_2	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (Read)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
x	x	x	x	1	Data Bus → 3 State
x	x	1	1	0	Data Bus → 3 State

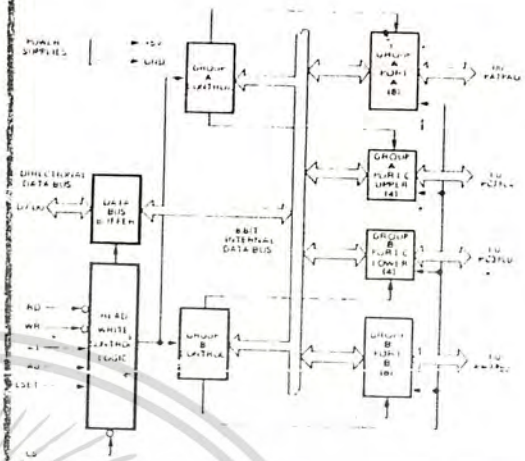


Figure 1. UM82C55A Block Diagram Data Bus Buffer and Read/Write Control Logic Functions

(Reset)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode. "Bus Hold" devices internal to the UM82C55A will hold the I/O port inputs to a logic "1" state with a maximum hold current of 300 μA.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the UM82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the UM82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)
Control Group B – Port B and Port C lower (C3-C0)

The control word register can be both written and read as shown in the "Basic Operation" table. Figure 4 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B and C

The UM82C55A contains three 8 bit ports (A, B, and C). All can be configured to a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the UM82C55A.

Port A One 8 bit data output latch/buffer and one 8-bit data input latch. Both "pull up" and "pull-down" bus hold devices are present on Port A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Port B One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input)x This port can be divided into two 4-bit ports under the mode control. Each 4 bit port contains a 4 bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

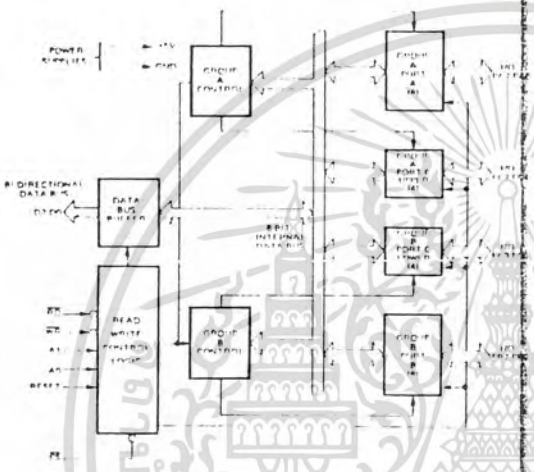


Figure 2. UM82C55A Block Diagram Showing Group A and Group B Control Functions

Operational Description

Mode Selection

There are three basic modes of operation that can be selected by the system software.

Mode 0 – Basic Input/Output
Mode 1 – Strobed Input/Output
Mode 2 – Bi-Directional Bus

When the reset input goes "high", all ports will be set to the input mode with all 24 port lines held at a logic "one" level by internal bus hold devices. After the reset is removed, the UM82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown resistors in all-CMOS designs. During the execution of the system program, any of the other modes may be selected using a single output instruction. This allows a single UM82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance, Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt driven basis.

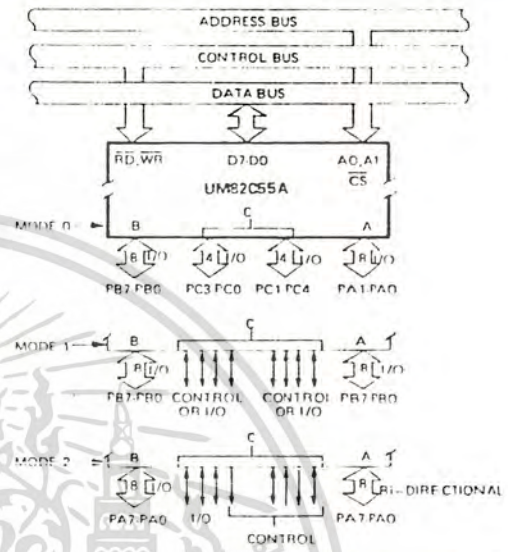


Figure 3. Basic Mode Definitions and Bus Interface

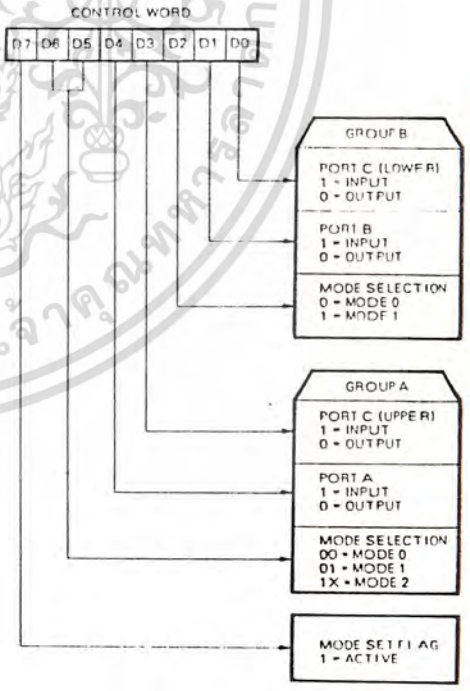


Figure 4. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the UM82C55A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

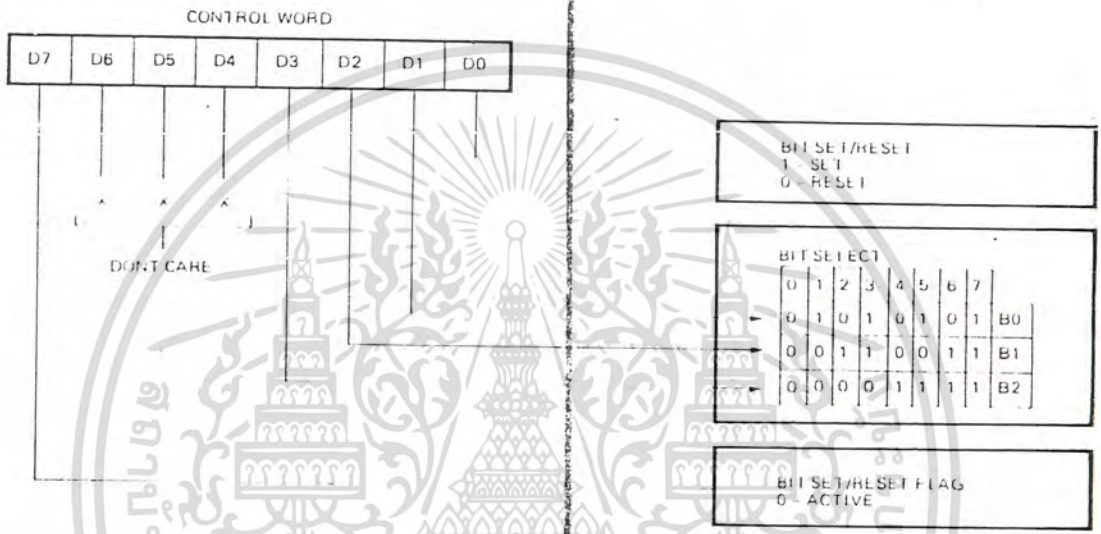


Figure 5. Bit Set/Reset Format

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUT put instruction. This feature reduces software requirements in control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the UM82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the programmer to enable or disable a CPU interrupt by a specific I/O device without affecting any other device in the interrupt structure.

INTE Flip-flop Definition

(BIT SET) – INTE is SET – Interrupt enable.
(BIT – RESET) – INTE is RESET – Interrupt disable.

Note: All Mask flip flops are automatically reset during mode selection and device Reset.

Operating Modes

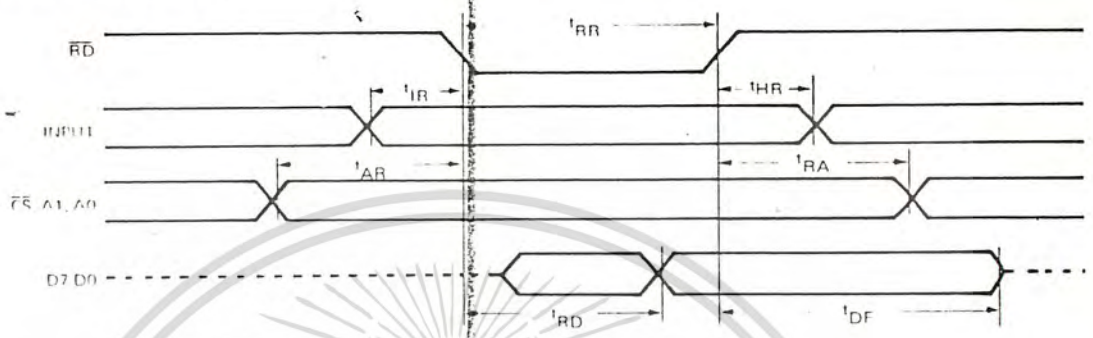
Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specific port.

Mode 0 Basic Functional Definitions:

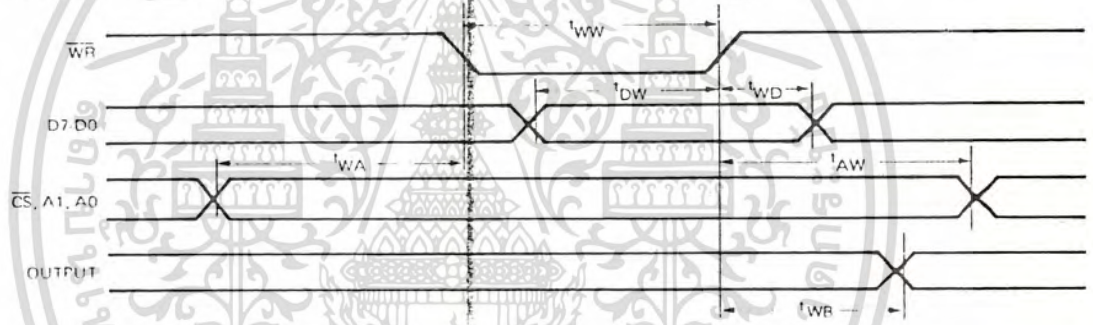
- Two 8 bit ports and two 4-bit ports
- Any port can be input or output
- Outputs are latched
- Inputs are not latched
- 16 different Input/Output configurations possible.

Central Purpose

MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)



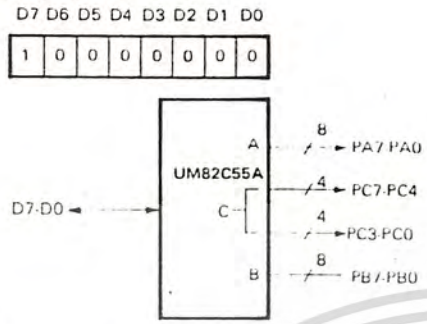
Mode 0 Port Definition

A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	Output	Output	0	Output	Output
0	0	0	1	Output	Output	1	Output	Input
0	0	1	0	Output	Output	2	Input	Output
0	0	1	1	Output	Output	3	Input	Input
0	1	0	0	Output	Input	4	Output	Output
0	1	0	1	Output	Input	5	Output	Input
0	1	1	0	Output	Input	6	Input	Output
0	1	1	1	Output	Input	7	Input	Input
1	0	0	0	Input	Output	8	Output	Output
1	0	0	1	Input	Output	9	Output	Input
1	0	1	0	Input	Output	10	Input	Output
1	0	1	1	Input	Output	11	Input	Input
1	1	0	0	Input	Input	12	Output	Output
1	1	0	1	Input	Input	13	Output	Input
1	1	1	0	Input	Input	14	Input	Output
1	1	1	1	Input	Input	15	Input	Input

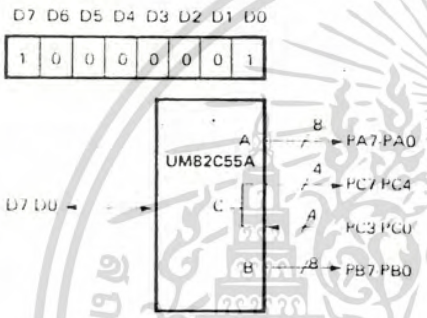
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Mode 0 Configurations

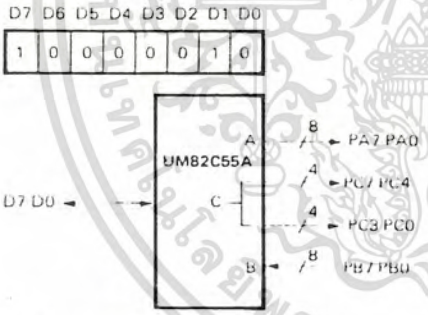
CONTROL WORD #0



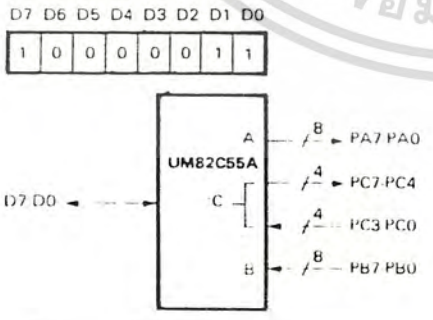
CONTROL WORD #1



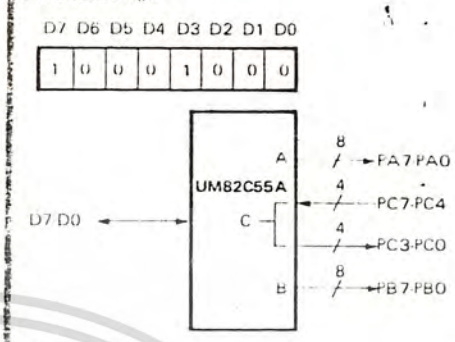
CONTROL WORD #2



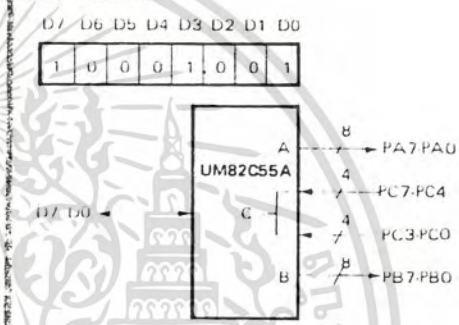
CONTROL WORD #3



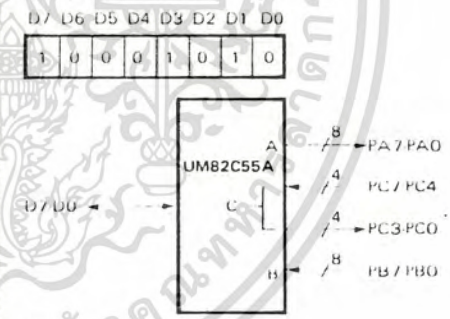
CONTROL WORD #4



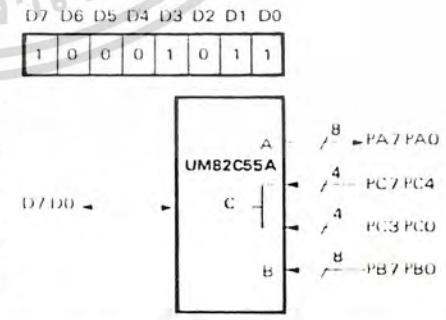
CONTROL WORD #5



CONTROL WORD #6



CONTROL WORD #7

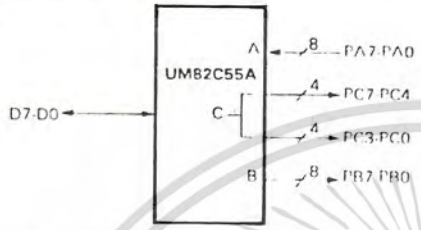


General Purpose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

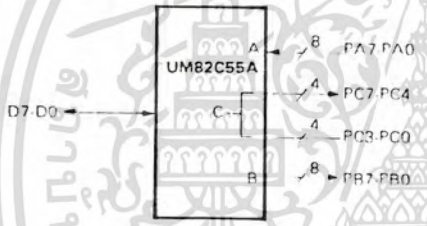
CONTROL WORD #8

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	0



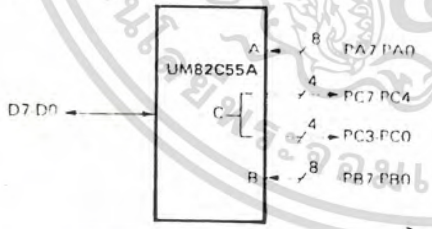
CONTROL WORD #9

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	1



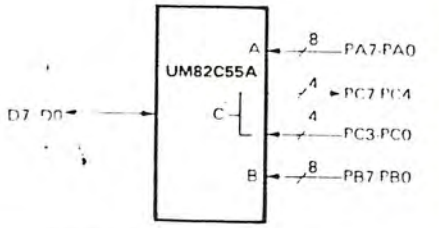
CONTROL WORD #10

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



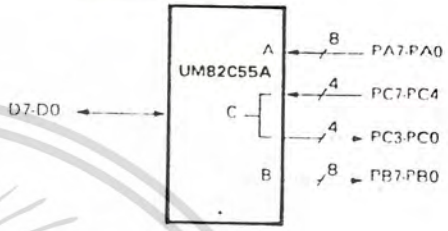
CONTROL WORD #11

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1



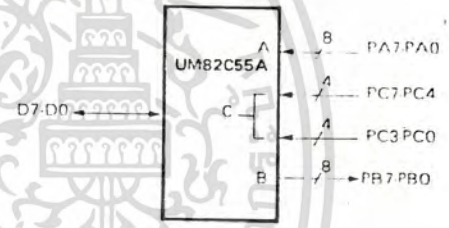
CONTROL WORD #12

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0



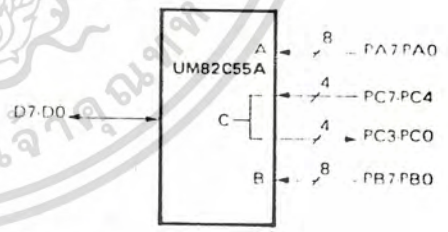
CONTROL WORD #13

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	1



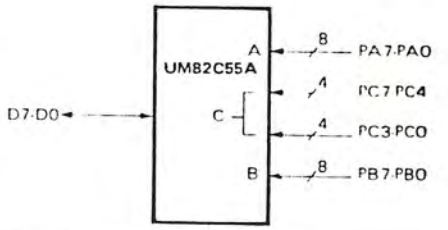
CONTROL WORD #14

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	0



CONTROL WORD #15

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Operating Modes

Mode 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions

- Two Groups (Group A and Group B)
- Each group contains one 8-bit port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit port.

Input Control Signal Definition

$\overline{STB}$ (Strobe Input)
A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch, in essence, an acknowledgement. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the condition: $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

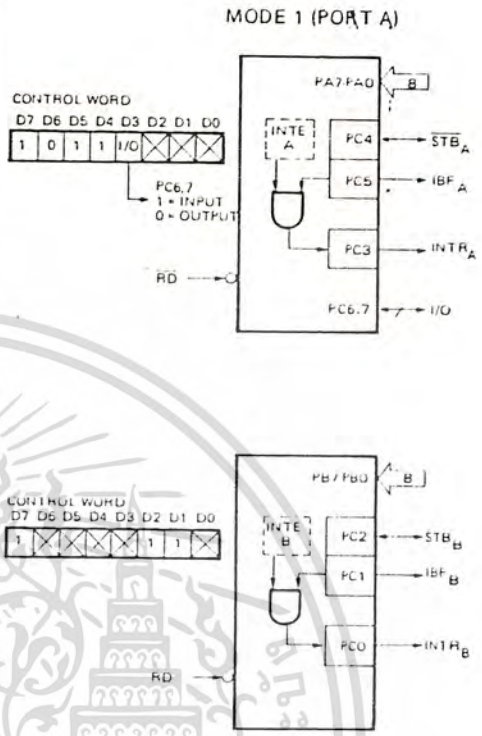


Figure 6. MODE 1 Input

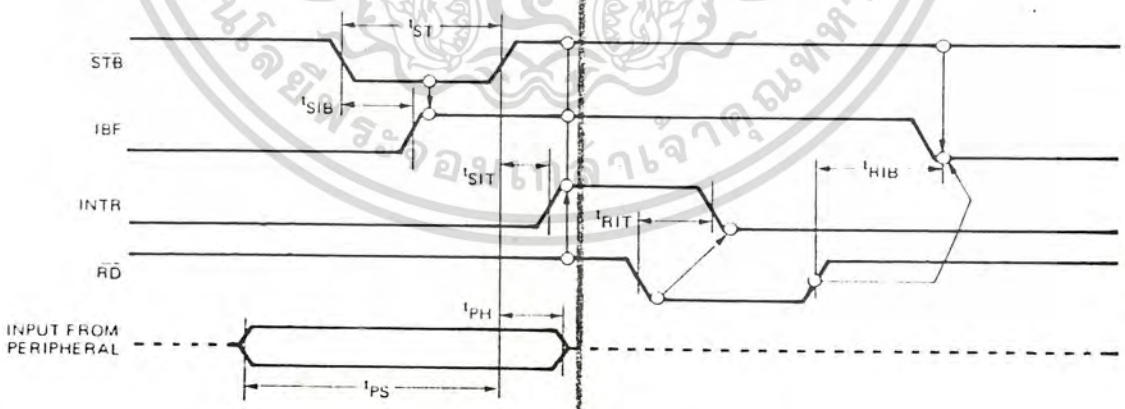


Figure 7. MODE 1 (Strobed Input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the UM82C55A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the $\overline{CPU}$.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the $\overline{CPU}$ when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by Bit Set/Reset of PC_6.

INTE B

Controlled by Bit Set/Reset of PC_2.

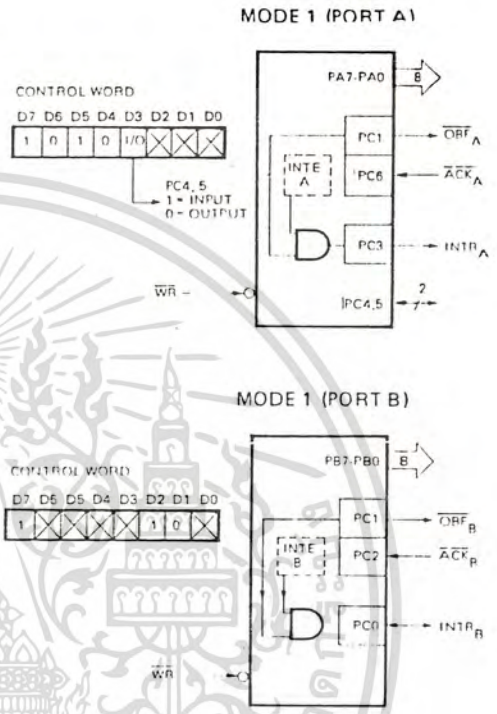


Figure 8. MODE 1 Output

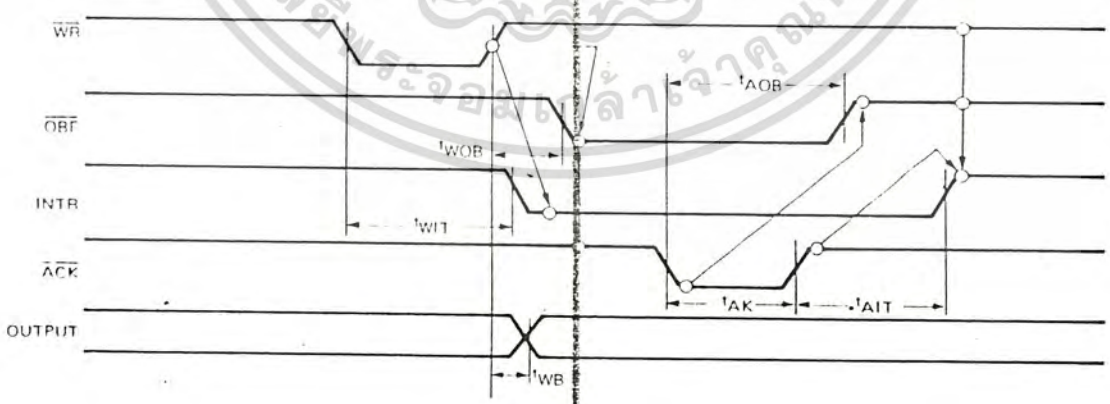


Figure 9. MODE 1 (Strobed Output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Combinations of MODE 1. Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

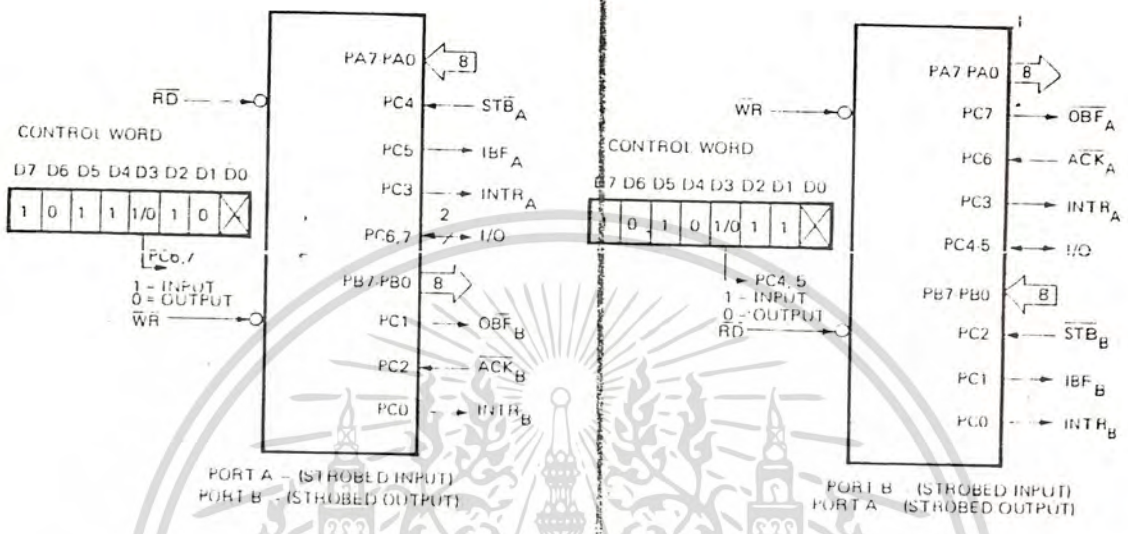


Figure 10. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O)

The functional configuration provides a means for communicating with a peripheral device or structure on a single 8 bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline similar to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Function Definitions:

- Used in Group A only.
- One 8 bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "Low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with $\overline{OBF}$). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

General Purpose Peripheral IC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

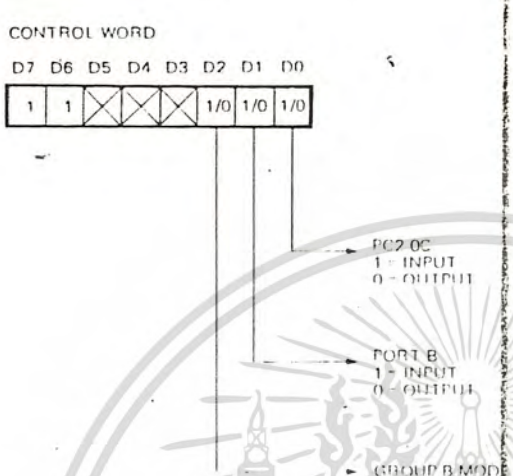


Figure 11. MODE Control Word

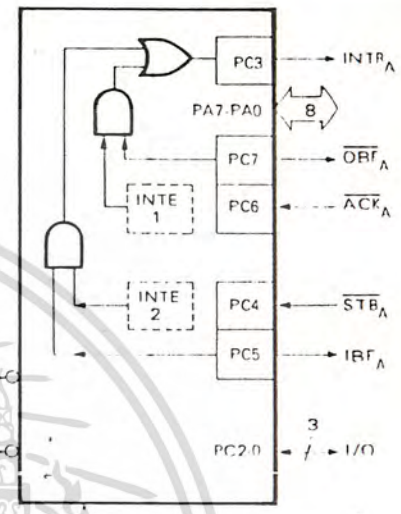


Figure 12. MODE 2

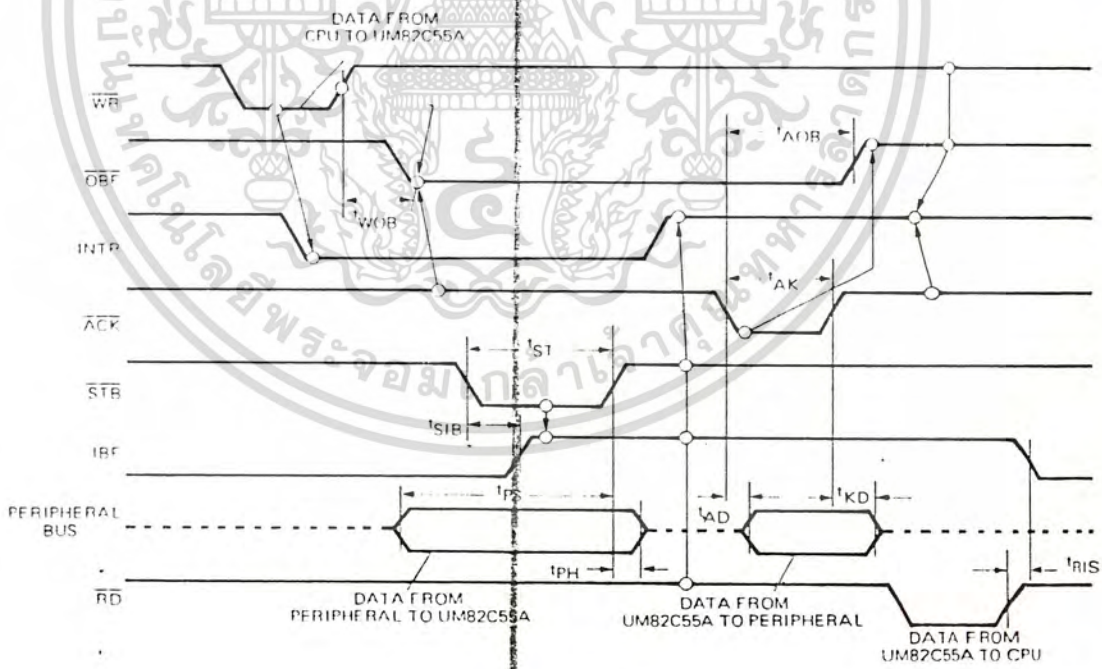
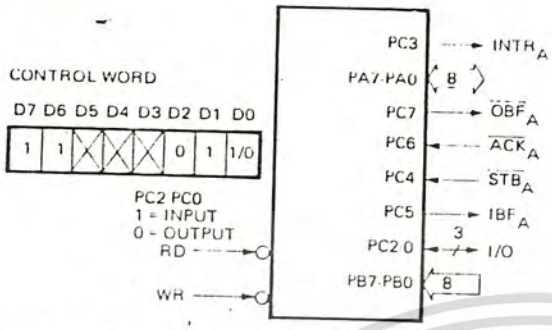


Figure 13. MODE 2 (Bidirectional)

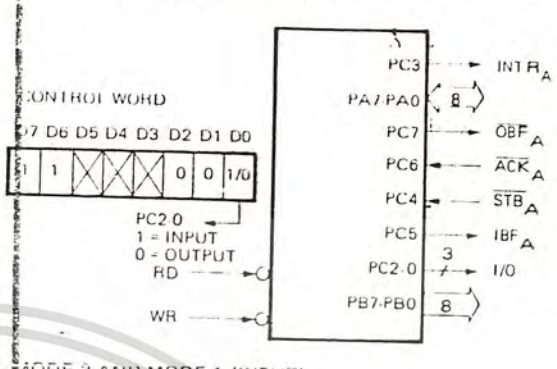
Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before RD is permissible. (INTR = IBF · MASK · $\overline{STB}$ · $\overline{RD}$ + $\overline{OBF}$ · MASK · $\overline{ACK}$ · $\overline{WR}$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

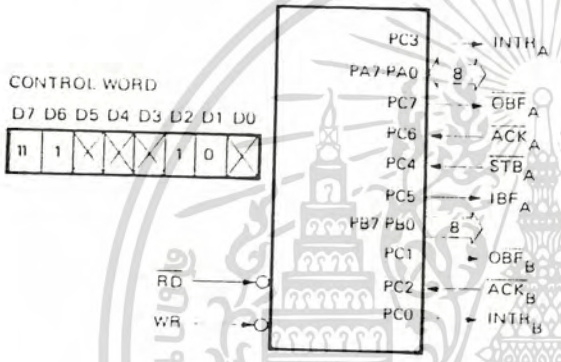
MODE 2 AND MODE 0 (INPUT)



MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)

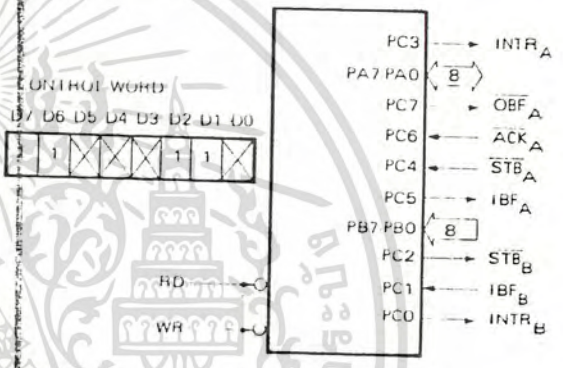


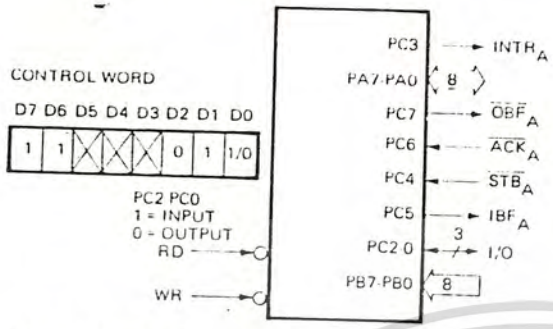
Figure 14. MODE 2 Combinations

Mode Definition Summary

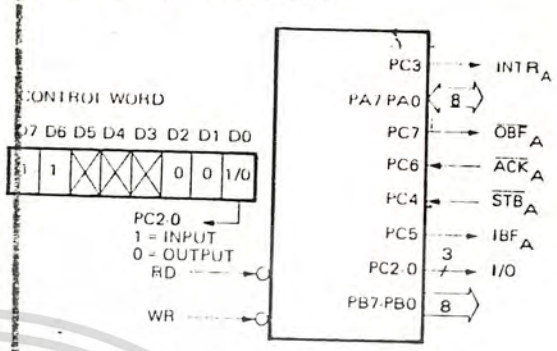
	MODE 0		MODE 1		MODE 2 Group A Only
	IN	OUT	IN	OUT	
PA_0	IN	OUT	IN	OUT	
PA_1	IN	OUT	IN	OUT	
PA_2	IN	OUT	IN	OUT	
PA_3	IN	OUT	IN	OUT	
PA_4	IN	OUT	IN	OUT	
PA_5	IN	OUT	IN	OUT	
PA_6	IN	OUT	IN	OUT	
PA_7	IN	OUT	IN	OUT	
PB_0	IN	OUT	IN	OUT	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	
PB_2	IN	OUT	IN	OUT	
PB_3	IN	OUT	IN	OUT	
PB_4	IN	OUT	IN	OUT	
PB_5	IN	OUT	IN	OUT	
PB_6	IN	OUT	IN	OUT	
PB_7	IN	OUT	IN	OUT	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O
PC_3	IN	OUT	$INTR_A$	$INTR_A$	INTRA
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$
PC_5	IN	OUT	IBF_A	I/O	IBF_A
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

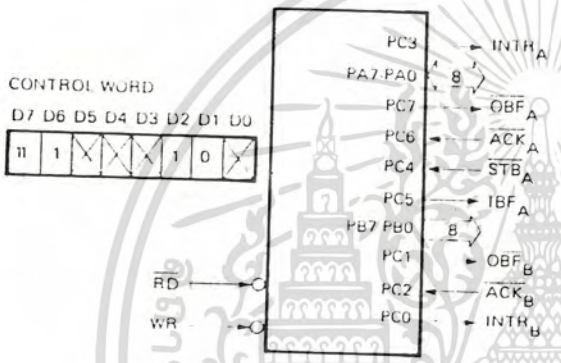
MODE 2 AND MODE 0 (INPUT)



MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)

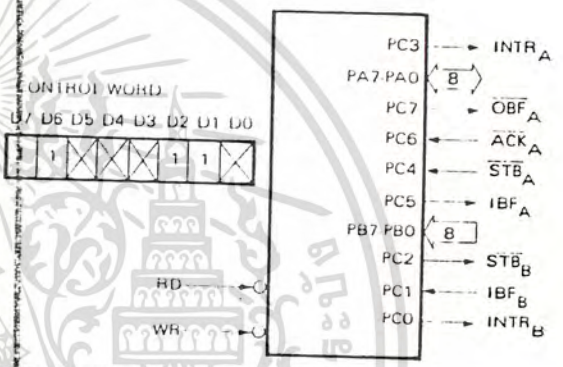


Figure 14. MODE 2 Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2 Group A Only
	IN	OUT	IN	OUT	
PA_0	IN	OUT	IN	OUT	
PA_1	IN	OUT	IN	OUT	
PA_2	IN	OUT	IN	OUT	
PA_3	IN	OUT	IN	OUT	
PA_4	IN	OUT	IN	OUT	
PA_5	IN	OUT	IN	OUT	
PA_6	IN	OUT	IN	OUT	
PA_7	IN	OUT	IN	OUT	
PB_0	IN	OUT	IN	OUT	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	
PB_2	IN	OUT	IN	OUT	
PB_3	IN	OUT	IN	OUT	
PB_4	IN	OUT	IN	OUT	
PB_5	IN	OUT	IN	OUT	
PB_6	IN	OUT	IN	OUT	
PB_7	IN	OUT	IN	OUT	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O
PC_3	IN	OUT	$INTR_A$	$INTR_A$	INTRA
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$
PC_5	IN	OUT	IBF_A	I/O	IBFA
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Special Mode Combination Considerations:

There are several combinations of modes possible. For any combination, some or all of Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the $\overline{ACK}$ and $\overline{STB}$ lines, will be placed on the data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 17.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and $\overline{OBF}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{ACK}$ and $\overline{STB}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 17.

INPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
			GROUP A			GROUP B	

OUTPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
			GROUP A			GROUP B	

Figure 15. MODE 1 Status Word Format

D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$			
		GROUP A				GROUP B	

(DEFINED BY MODE 0 OR MODE 1 SELECTION)

Figure 16. MODE 2 Status Word Format

Interrupt Enable Flag*	Position	Alternate Port C Pin Signal (Mode)
INTE_B	PC2	$\overline{ACK}_B$ (Output Mode 1) or $\overline{STB}_B$ (Input Mode 1)
INTE_A2	PC4	$\overline{STB}_A$ (Input Mode 1 or Mode 2)
INTE_A1	PC6	$\overline{ACK}_A$ (Output Mode 1 or Mode 2)

Figure 17. Interrupt Enable Flags in Modes 1 and 2

Current Drive Capability:

Any output on Port A, B or C can sink or source 2.5mA. This feature allows the UM82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the UM82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device.. Reading the contents of Port C allows that programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Applications of the UM82C55A

The UM82C55A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the UM82C55A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the UM82C55A to exactly "fit" the application. Figures 10 through 24 present a few examples of typical applications of the UM82C55A.

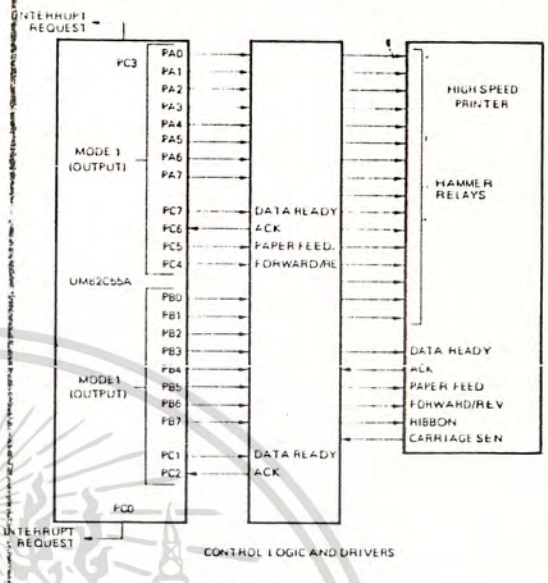


Figure 18. Printer Interface

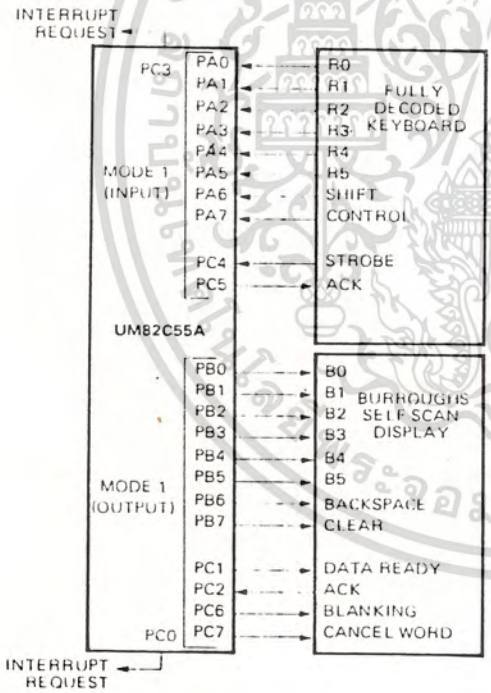


Figure 19. Keyboard and Display Interface

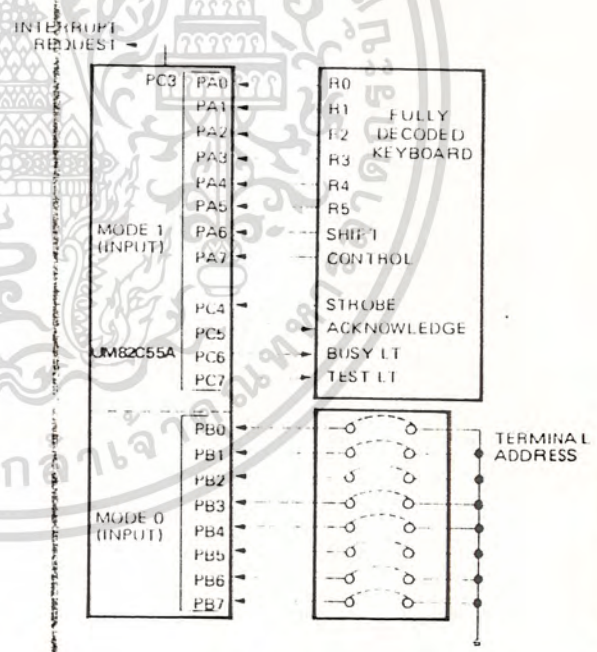


Figure 20. Keyboard and Terminal Address Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

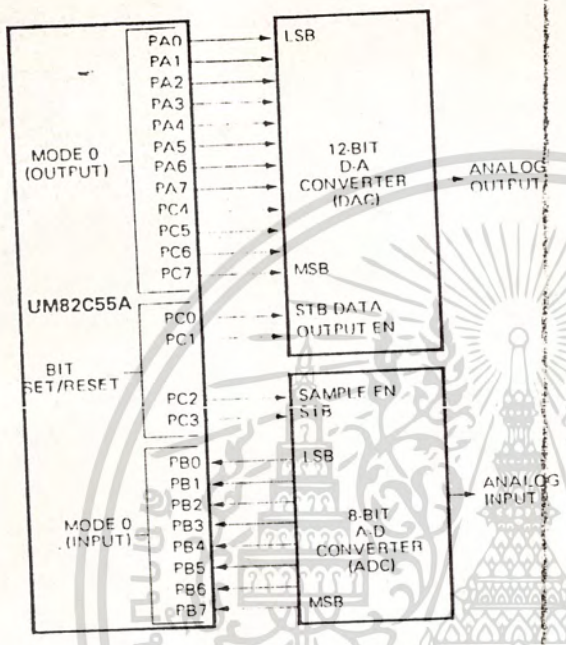


Figure 21. Digital to Analog, Analog to Digital

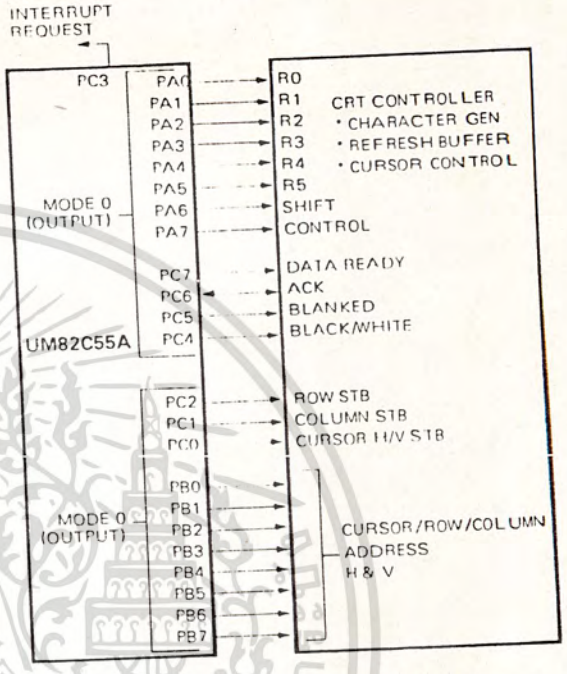


Figure 22. Basic CRT Controller Interface

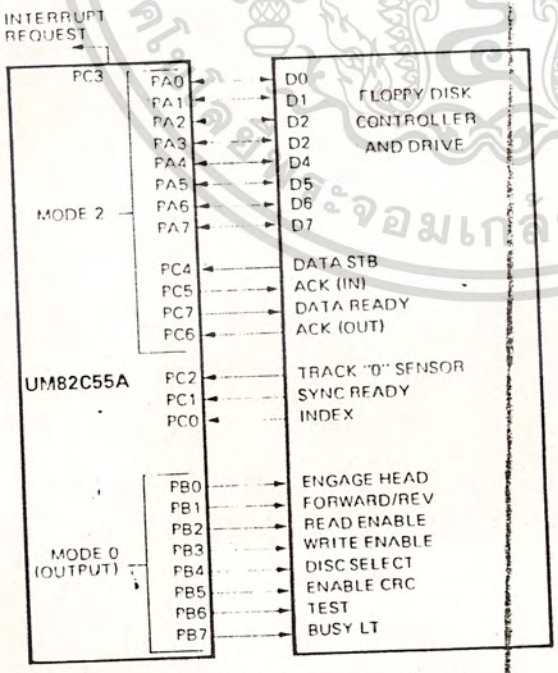


Figure 23. Basic Floppy Disk Interface

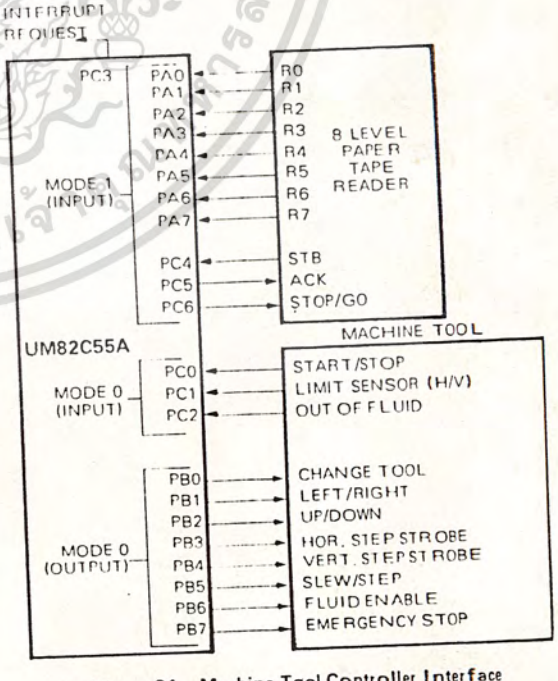


Figure 24. Machine Tool Controller Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า