เครื่องอำนวยความสะดวกในการใช้งานโทรศัพท์อัตโนมัติแบบเอนกประสงค์

MULTIPURPOSE AUTOMATIC TELEPHONE FACILITIES

โดย

นายจักรพรรดิ ไชยสิทธิ์

นายเดี่ยว ชัยมณี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เครื่องอำนวยความสะดวกในการใช้งานโทรศัพท์อัตโนมัติแบบเอนกประสงค์

MULTIPURPOSE AUTOMATIC TELEPHONE FACILITIES

โดย

นายจักรพรรดิ ไชยสิทธิ์ 42010052

นายเดี่ยว ชัยมณี 42010121

อาจารย์ที่ปรึกษา

รศ.ดร.กอบชัย เดชหาญ

อ.สุรพล บุญจันทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เครื่องอำนวยความสะดวกในการใช้โทรศัพท์อัตโนมัติแบบเอนกประสงค์

MULTIPURPOSE AUTOMATIC TELEPHONE FACILITIES

โดย 1. นายจักรพรรดิ ไชยสิทธิ์ 42010052

2. นายเดี่ยว ชัยมณี 42010121

อาจารย์ที่ปรึกษา ....

1. (รศ.ดร.กอบชัย เดชหาญ)

....

2. (อ.สุรพล บุญจันทร์)

บทคัดย่อ

ในปัจจุบันโทรศัพท์นับเป็นอุปกรณ์สื่อสารที่มีประโยชน์ และจำเป็นต่อมนุษย์อย่างมากในการติดต่อสื่อสารกัน ดังนั้นจึงได้มีการพัฒนาเทคโนโลยีต่าง ๆ เพื่ออำนวยความสะดวกแก่มนุษย์ในการใช้โทรศัพท์ ซึ่งในโครงงานจะรวมเอาเครื่องตอบรับโทรศัพท์อัตโนมัติ เครื่องสั่งงานทางโทรศัพท์ เครื่องบันทึกหมายเลขการโทรออก คิดคำนวณค่าบริการโทรศัพท์ และนำมาแสดงผลที่จอแอลซีดี โดยใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวควบคุมการทำงานของวงจรต่าง ๆ ที่จะทำงานสัมพันธ์กับโปรแกรมที่ออกแบบไว้

ABSTRACT

Now, The telephone is useful and important device communication for human communication. The technology of facilities telephone is developed to increase the conveniences. This project concerns about the facilities telephone which include the automatic answering machine, controlling appliance, telephone number recording, the calculation of telephone service charges and showing on LCD display by using microcontroller MCS-51 to control all circuit work relatively by using the designed program.

สารบัญ

สารบัญตาราง

หน้า

สารบัญรูป

หน้า

บทที่ 1
บทนำ

ในปัจจุบันโทรศัพท์นับเป็นอุปกรณ์สื่อสารที่มีประโยชน์ และมีความสำคัญในชีวิตประจำวันเพิ่มมากขึ้นทุกขณะ ไม่ว่าจะใช้งานโทรศัพท์ในบ้านพักอาศัย หรือการใช้งานในสำนักงาน, โรงงาน หรือธุรกิจทั่วๆ ไป เพราะสามารถช่วยในการติดต่อสื่อสารระหว่างกันได้สะดวกและรวดเร็วมากขึ้น และไม่ต้องสูญเสียเวลาไปกับการเดินทาง และชีวิตความเป็นอยู่ของผู้คนในสังคมปัจจุบันต้องมีความรีบเร่งเพื่อหารายได้ในการดำรงชีวิต จนดูเหมือนกับว่าเวลาส่วนใหญ่จะอยู่ภายนอกบ้านมากกว่า ทำให้บางครั้งพลาดจากการติดต่อที่สำคัญจากคนที่ติดต่อทางโทรศัพท์เข้ามาภายในบ้านในขณะที่ไม่มีใครอยู่บ้าน ดังนั้นจึงเป็นเหตุผลสำคัญที่จัดทำโครงงานนี้ขึ้นมา เพื่อที่จะทำหน้าที่ในการตอบรับโทรศัพท์ในขณะที่ไม่มีใครอยู่บ้าน พร้อมกันนั้นยังสามารถที่จะฝากข้อความของผู้ที่โทรเข้ามา ทำให้ไม่พลาดการติดต่อที่สำคัญ นอกจากนี้ยังมีความสามารถในด้านอื่นๆ อีกคือ สามารถที่จะคิดเงินค่าโทรศัพท์พร้อมบันทึกเลขหมายที่โทรออกเก็บไว้ด้วย เพื่อใช้ในการเปรียบเทียบกับใบแจ้งนี้ค่าโทรศัพท์ และในบางครั้งเมื่อเราออกไปนอกบ้านแล้วอาจจะลืมปิดเครื่องใช้ไฟฟ้าภายในบ้าน เช่น โทรทัศน์, วิทยุ, แอร์, หลอดไฟ, หรืออุปกรณ์ไฟฟ้าอื่นๆ เป็นต้น เพราะว่ามันจะเป็นการสิ้นเปลืองพลังงานไฟฟ้าโดยไม่จำเป็น โครงงานนี้จึงได้มีส่วนของการควบคุมการปิด-เปิดเครื่องใช้ไฟฟ้าผ่านคู่สายโทรศัพท์

โครงงานนี้สามารถใช้งานร่วมกับเครื่องรับโทรศัพท์โดยทั่วไป โดยโครงสร้างของโครงงานนี้จะแบ่งออกเป็น 2 ส่วนใหญ่ๆ ดังต่อไปนี้คือ

1. ส่วนฮาร์ดแวร์ จะประกอบไปด้วย
    - 1.1 ส่วนของวงจรตรวจจับสัญญาณโทรศัพท์
    - 1.2 ส่วนของวงจรตอบรับ และบันทึกข้อความ
    - 1.3 ส่วนของวงจรควบคุมปิด-เปิดเครื่องใช้ไฟฟ้า
    - 1.4 ส่วนของระบบไมโครคอนโทนเลอร์
    - 1.5 ส่วนของวงจรแสดงผลแบบเอลซีดีพร้อมทั้งส่วนคีบอร์ด
2. ส่วนซอฟต์แวร์ จะประกอบไปด้วย
    - 2.1 โปรแกรมตรวจสอบสัญญาณทางโทรศัพท์
    - 2.2 โปรแกรมควบคุมการตอบรับและบันทึกข้อความ
    - 2.3 โปรแกรมควบคุมการปิด-เปิดเครื่องใช้ไฟฟ้า
    - 2.4 โปรแกรมเก็บข้อมูลเลขหมายโทรศัพท์
    - 2.5 โปรแกรมแสดงผลแอลซีดีและควบคุมคีย์บอร์ด
    - 2.6 โปรแกรมคิดเงินค่าโทรศัพท์

ขั้นตอนในการดำเนินงาน

ภาคการศึกษา 1/2545 ทำในส่วนของฮาดแวร์ โดยส่วนที่ได้ทำไปแล้วมีดังนี้

- ส่วนของวงจรตรวจจับสัญญาณโทรศัพท์
- ส่วนของวงจรบันทึกข้อความ

ภาคการศึกษา 2/2545 ทำในส่วนของฮาดแวร์ที่เหลือทั้งหมด และทำการเขียนโปรแกรมควบคุมการทำงานทั้งหมด

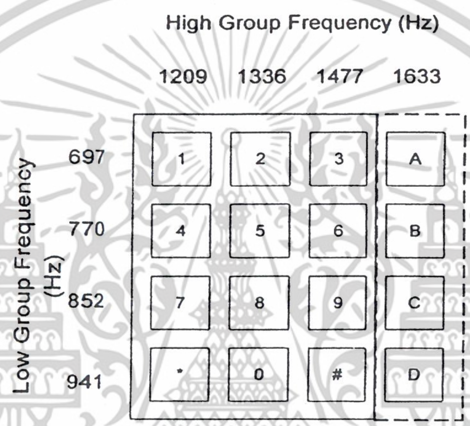
บทที่2

ทฤษฎีและหลักการ

2.1 หลักการเบื้องต้นของโทรศัพท์

เครื่องโทรศัพท์แบ่งออกได้ 2 ชนิด คือ แบบหมุน (Rotary Dial) และแบบกดปุ่ม (Touch Tone) ในที่นี้ขอกล่าวถึงแบบกดปุ่มเท่านั้น

โทรศัพท์แบบกดปุ่มจะสร้างสัญญาณ DTMF (Dual Tone Mutiple Frequency) ในการส่งเลขหมาย โดยการกดแต่ละเลขหมายบนหน้าปัดโทรศัพท์ จะมีการส่งความถี่ออกไป 2 ความถี่พร้อมกัน โดยทั่วไปหน้าปัดจะมี 12 ปุ่ม แบ่งอกได้ 4 แถว (Rows) 3 หลัก (Columns) ความถี่ใช้ในแต่ละแถวและหลักจะมีความถี่ต่างหัน ความถี่ของทั้ง 4 แถว เรียกว่ากลุ่มความถี่ต่ำ (Low Group Frequency)และความถี่ของทั้ง 3 หลัก เรียกว่ากลุ่มความถี่สูง บางแบบอาจมี 4 หลัก ดังรูป 2.1



รูปที่ 2.1 แสดงหน้าปัดของเครื่องโทรศัพท์แบบกดปุ่มและความถี่ที่ใช้

ระบบการต่อของเครื่องโทรศัพท์กับชุมสายโทรศัพท์

สายส่ง 2 เส้น มีชื่อเรียกว่า ทิป (Tip) และ ริง (Ring) เครื่องโทรศัพท์ไปชุมสายปลายทางมีค่าอิมพีแดนซ์ของสายประมาณ 500-1000 โอห์ม ชุมสายปลายทางมีการติดตั้งแหล่งจ่ายไฟร่วมไฟตรงขนาด 48 โวลต์ให้แต่ละลูปของผู้ใช้โทรศัพท์ โดยริงจะต่อกับสัญญาณไฟ -48 โวลต์ เป็น -10 โวลต์ เมื่อผู้ใช้ยกหูโทรศัพท์ และกระแสไฟฟ้าไฟตรงขนาด 20 mA ไหลอยู่ในลูป ทำให้ชุมสายโทรศัพท์รู้ว่าเป็นการเริ่มต้นการเรียกก็จะส่งสัญญาณไปยังผู้เรียกซึ่งสัญญาณต่างๆ มีดังนี้

- สัญญาณระหว่างผู้เช่าไปยังชุมสาย

1.สัญญาณที่ส่งจากผู้เช่าไปยังชุมสาย

1.1 Off-Hook คือ สภาพผู้เช่ายกหูโทรศัพท์สายจะมีสภาพ Closed Loop(Low Impedance)

1.2 On-Hook คือ สภาพผู้เช่าวางหูโทรศัพท์สายจะมีสภาพ Open Loop(High Impedance)

1.3 Dialing คือ สภาพผู้เช่าเลขหมายเข้าเครื่องเป็น Rotary Dial สัญญาณจะเป็น Pulsing ค่า Impedance จะสูง ต่ำสลับกันไปตามเลขหมายที่หมุน ถ้าเป็นแบบกดปุ่ม Touch-Tone สัญญาณ ออกจะเป็นความถี่ DTMF ส่งออกไปชุมสาย

2.สัญญาณที่ส่งมาจากชุมสายมายังผู้เช่า

2.1 สัญญาณให้หมุน (Dial Tone) คือสัญญาณที่บอกสภาพการว่างของชุมสายพร้อมที่จะรับเลขหมาย สัญญาณ Dial Tone .จะเป็นสัญญาณต่อเนื่องมีความถี่ 425 Hz มอดูเลทกับ 50 Hz ผู้เช่าจะได้ยินเสียงสัญญาณเมื่อทำการยกหูโทรศัพท์

2.2 สัญญาณไม่ว่าง (Busy Tone) คือ สัญญาณบอกให้ทราบว่าสายไม่ว่าง สัญญาณนี้ถ้าได้ยินตอนยกหูแสดงว่าอุปกรณ์ในการตัดต่อชุมสายไม่ว่างและถ้าได้ยินสัญญาณนี้หลังจากหมุนหมายเลขไปแล้วแดสงว่าผู้เช่าฝ่ายถูกเรียกสายไม่ว่างลักษณะสัญญาณที่ส่งไปจะเป็นสัญญาณที่ขาดตอนเป็นช่วงๆ ช่วงเวลาส่งประมาณ 0.5 วินาที เงียบประมาณ 0.5 วินาที ความถี่ของสัญญาณ 425 Hz เป็นรูปไซน์

2.3 สัญญาณเรียกกลับ (Ring Back Tone) คือ สัญญาณที่ผู้เรียกได้ยินหลังจากหมุนเลขหมายครบแล้วเพื่อบอกให้ทราบว่าชุมสายโทรศัพท์ทำการต่อกระทำได้เสร็จเป็นสัญญาณ 425 Hz เป็นสัญญาณรูปไซน์โดยส่ง 1 วินาที หยุด 4 วินาที

2.4 สัญญาณกริ่งเรียก (Ringing Tone) คือ สัญญาณที่ผู้ถูกเรียก ได้ยินโดยชุมสายโทรศัพท์ ดำเนินการส่งสัญญาณ 25 Hz ช่วงเวลาส่ง 2 วินาที หยุด 4 วินาที สัญญาณนี้จะทำการส่งก็ต่อเมื่อ ชุมสายได้ทำการต่อระหว่างผู้เรียกกับผู้ถูกเรียกได้สำเร็จ

- สัญญาณติดต่อระหว่างชุมสายกับชุมสาย (Inter Exchange Signaling)

1. สัญญาณพื้นฐานมีอยู่ 5 ประเภท คือ

1.1สัญญาณจับวงจร (seizure) เป็นสัญญาณเรียกชุมสายปลายทางทราบว่าคู่สายขณะนี้ถูกใช้งานอยู่ ชุมสายปลายทางจะทำการจัดเตรียมอุปกรณ์ที่รับเลขหมายของผู้เรียกที่จะส่งมา

1.2 สัญญาณบอกเลขหมาย (Address Information) เป็นสัญญาณบอกเลขหมายหรือประเภทของผู้เช่า

1.3 สัญญาณตอบรับ (Answer Signal) สัญญาณนี้ถูกส่งเมื่อผู้เช่าฝ่ายถูกเรียกยกหูรับหน้าที่หลักของสัญญาณนี้คือ

- เริ่มต้นคิดค่าบริการ
- ส่งสัญญาณคิดค่าบริการ
- ตัดวงจรจับเวลาการใช้อุปกรณ์
- สัญญาณยกเลิกการต่อตรง (Clear-Forward) จะถูกส่งเมื่อฝ่ายเรียกวางหูผลของสัญญาณจะทำให้วงจรทางปลายทางทำการยกเลิกการติดวงจรต่าง ๆ
- สัญญาณยกเลิกการต่อกลับ (Clear-Back) จะถูกส่งเมื่อผู้เช่าฝ่ายถูกเรียกวางหู ผลของสัญญาณนี้จะทำให้ชุมสายต้นทางเริ่มต้นจับเวลาเมื่อเวลาผ่านไป 90-120 วินาที ชุมสายต้นทางจะยกเลิกการติดต่อพร้อมกับส่งสัญญาณ (Clear-Forward) ออกไปเพื่อให้ ชุมสายปลายทางยกเลิกเช่นกัน

2.2 ข้อมูลทั่วไปเกี่ยวกับโทรศัพท์ที่ใช้ในประเทศไทยในปัจจุบัน

ระบบโทรศัพท์ที่ใช้ในประเทศไทย ปัจจุบันสามารถแบ่งได้เป็น 2 ระบบใหญ่ ๆ คือ ระบบโทรศัพท์พื้นฐาน และระบบโทรศัพท์เคลื่อนที่ ซึ่งในแต่ละระบบมีรายละเอียดดังนี้

1.ระบบโทรศัพท์พื้นฐานมีทั้งหมด 2 ระบบต่อพื้นที่ทั่วประเทศ ผู้ให้บริการทั้งหมด 3 รายได้แก่

1.1 Telecom Asia (TA) ให้บริการเฉพาะในเขตกรุงเทพฯ

1.2 TT&T ให้บริการเฉพาะในเขตภูมิภาค

1.3 องค์การโทรศัพท์แห่งประเทศไทย มีทั้งส่วนที่ให้บริการในเขตกรุงเทพฯ และเขตภูมิภาค

2. ระบบโทรศัพท์เคลื่อนที่ มีทั้งหมด 9 ระบบต่อพื้นที่ทั่วไทย และผู้ให้บริการทั้งสิ้น 6 รายโดยสามารถ จำแนกตามระบบได้ดังนี้

2.1 ระบบ NMT470 โดยองค์การโทรศัพท์แห่งประเทศไทย

2.2 ระบบ 800 Band A โดยการสื่อสารแห่งประเทศไทย

2.3 ระบบเซลลูล่าร์ 900 โดย บมจ. แอดวานซ์ อินโฟร์ เซอร์วิส

2.4 ระบบเวิล์ดโฟน 800 บมจ. โทเทิล แอ็คเซส คอมมูนิเคชั่น

2.5 ระบบดิจิตอล GSM โดย บมจ. แอดวานซ์ อินโฟร์ เซอร์วิส

2.6 ระบบเวิล์ดโฟน 1800 ดิจิตอล โดย บมจ. โทเทิล แอ็คเซส คอมมูนิเคชั่น

2.7 ระบบ PCN 1800 * โดย บริษัท WCS ในเครือข่ายของ IEC

2.8 ระบบ PCN 1800 * โดย บริษัท DPC ในเครือข่ายของสามารถ

2.9 ระบบ Band A CDMA* โดยการสื่อสารแห่งประเทศไทย

ทั้ง 3 ระบบสุดท้าย ที่กำกับด้วยเครื่องหมาย (*) เป็นระบบใหม่ และยังมีโครงการของระทรวงคมนาคมที่คิดจะเปิดให้มีการบริการเพิ่มขึ้นอีก 2 ระบบ คือ ระบบในคลื่นความถี่ 1500 MHz และ ระบบ 1900 MHz อีก 2 ระบบ ในอนาคตอันใกล้นี้

- การบริการต่างๆ ขององค์การโทรศัพท์แห่งประเทศไทย

1. บริการสื่อสารร่วมระบบดิจิตอล คือ บริการสื่อสารที่สามรถรับส่งได้ทั้งสัญญาณข้องมูลและ ภาพเป็นระบบดิจิตอล ด้วยคู่สาย ISDN เพียง 1 คู่สายได้อย่างมีประสิทธิภาพ

2. บริการประชุมทางไกลผ่านจอภาพ เป็นบริการติดต่อสื่อสารทั้งเสียงและภาพเคลื่อนไหวเหมือนจริงเหมาะกับกิจการขนาดใหญ่ ที่มีสำนักงานและสาขาที่ต้องมีการจัดการประชุมอยู่เป็นประจำ

3. บริการพิเศษ SPC ประกอบด้วยบริการพิเศษให้เลือกขอใช้ 8 ประเภท คือ

- บริการแจ้งค่าพูดโทรศัพท์ทางไกลทันที
- บริการเปลี่ยนเรียกหมายเลข
- บริการเรียกซ้ำอัตโนมัติ
- บริการหมายเลขย่อ
- บริการประชุมทางโทรศัพท์
- บริการจำกัดการเรียกออก
- บริการหมายเลขด่วน
- บริการรับสายเรียกซ้อน

4. บริการโทรศัพท์ติดตามตัว เช่น โฟนลิ้งค์ (Phone link), เพจโฟน (Page-Phone) หรือ เวิลด์เพจ (World page) เป็นบริการโทรศัพท์ติดตามตัวแบบรับได้ทางเดียว ที่อาศัยคลื่นวิทยุเป็นสื่อ

5. บริการโทรศัพท์เคลื่อนที่เฉพาะกลุ่มธุรกิจ (Trucked Mobile Radio) เป็นบริการโทรศัพท์เคลื่อนที่เฉพาะกลุ่มธุรกิจทั่วประเทศ เพื่อการสื่อสารระหว่างผู้ใช้ในกลุ่มเดียวกัน

6. บริการส่งข้อมูลเพื่อธุรกิจผ่านดาวเทียม (ISBN) เป็นการสื่อสารข้อมูลผ่านดาวเทียม และโทรศัพท์ผ่านดาวเทียม โดยผ่านสถานีกลาง (HUB)

7. บริการโทรศัพท์สาธารณะแบบใช้บัตร (Card phone) จะเป็นการให้บริการโทรศัพท์สาธารณะแบบใช้บัตร

8. บริการโทรศัพท์เคลื่อนที่ (Cellular Mobile Telephone) เป็นบริการโทรศัพท์เคลื่อนที่ ระบบ NMT 900, GSM

9. บริการสื่อสารข้อมูลโดยใช้ระบบ Data kit Virtual Circuit Switch (DATANET) เป็นการให้บริการข้อมูลบนสายโทรศัพท์

10. บริการ 108 (PIN PHONE) เป็นการให้บริการโทรศัพท์โดยใช้รหัส

- โทรศัพท์ทางไกลระหว่างประเทศ

บริการโทรศัพท์ทางไกลระหว่างประเทศไทยกับเทศเพื่อนบ้าน ขององค์การโทรศัพท์แห่งประเทศไทยขณะนี้เปิดให้บริการอยู่ 2 ประเทศ คือ มาเลเซีย และ ลาว ท่านสามารถโทรศัพท์ไปประเทศทั้งสองได้ด้วยวิธีง่ายๆ ดังนี้

1. ต่อทางไกลอัตโนมัติด้วยตัวท่านเอง

กด 007 + รหัสประเทศ + รหัสเมือง + เลขหมายปลายทางรหัสประเทศมาเลเซีย 60 รหัสประเทศลาว 856

ตัวอย่างเช่น หากท่านต้องการ โทรศัพท์ไปยังเมืองต่างๆ โปรดกดรหัสดังนี้

- มาเลเซีย / เมืองกัวลาลัมเปอร์ กด 007 + 60 + 3 xxxxxxx

- ลาว / เมืองเวียงจันทร์ กด 007 + 856 + 21 + xxxxxxx

2. ต่อผ่านพนักงานต่อ

กด 101 เพื่อติดต่อโอเปอร์เรเตอร์ พร้อมบอกเลขหมายปลายทางให้ โอเปอร์เรเตอร์ต่อให้โทรศัพท์ทางไกลในประเทศ

- ปัจจุบันโทรศัพท์ทางไกลในประเทศแบ่งออกเป็น 2 ประเภท คือ

1. โทรศัพท์ทางไกลอัตโนมัติ

เป็นระบบโทรศัพท์ทางไกลที่สามารถติดต่อได้ด้วยตนเอง เพียงกดรหัสทางไกลแล้วตามด้วยหมายเลขโทรศัพท์ปลายทางที่ต้องการติดต่อ ดังนั้นเมื่อต้องการโทรศัพท์ทางไกลอัตโนมัติถึงผู้ใดควรตรวจดูรหัสทางไกลของชุมสายปลายทางที่จะติดต่อว่าอยู่ในเขตรหัสเดียวกันหรือต่างหัน เพราะถ้าเป็นเขตรหัสเดียวกันต่อพียงหมายเลขปลายทางเท่านั้นไม่ต้องต่อรหัสทางไกล

2. โทรศัพท์ทางไกลแบบพนักงานต่อ

“ หมุน 101 “ เพื่อขอใช้บริการโทรศัพท์ทางไกล โดยบอกร เยละเอียดแก่พนักงานต่อช้าๆ และชัดเจนดังนี้

- รหัสเขตโทรศัพท์ทาไกลของหมายเลขปลายทาง

- หมายเลขโทรศัพท์ปลายทาง
- หมายเลขโทรศัพท์ของตนเอง
- ชื่อของตนเอง

2.3 การคิดค่าบริการของโทรศัพท์ทางไกลในประเทศ

1. ค่าบริการให้พนักงานต่อ คิดเพิ่มขึ้นในอัตราครั้งละ 20 บาท สำหรับภาคกลางวัน และ 30 บาท สำหรับภาคค่ำและภาคดึก

2. ค่าบริการโทรศัพท์ คิดตามระยะทางและระยะเวลาที่ใช้ ดังนี้

-ภาคกลางวัน 07.00 น. – ก่อน 18.00 น.

ค่าบริการพนักงานต่อ 20 บาท และค่าโทรศัพท์ทางไกล

-ค่าค่ำ 18.00 น. – ก่อน 22.00 น.

ค่าบริการพนักงานต่อ 30 บาท และค่าโทรศัพท์ทางไกล

-ภาคดึก 22.22 น. – ก่อน 07.00 น.

ค่าบริการให้พนักงานต่อ 30 บาท และค่าโทรศัพท์ทางไกล

หมายเหตุ : บริการโทรศัพท์ทางไกลโดยพนักงานต่อประเภทเก็บเงินปลายทางคิดค่า บริการเก็บเงินปลายทางอีกครั้งละ 20 บาท

ตารางที่ 2.1 เปรียบเทียบอัตราลดภาคกลางวัน ภาคค่ำและภาคดึก (บาท/นาที)

ภาคกลางวัน 07.00 น. – ก่อน 18.00 น.	ภาคค่ำ 18.00 น. – ก่อน 22.00 น.	ภาคดึก 22.00 น. – ก่อน 07.00 น.
3.00	1.50	1.00
6.00	3.00	2.00
9.00	4.50	3.00
12.00	6.00	4.00
15.00	7.50	5.00
18.00	9.00	6.00

อัตราค่าบริการโทรศัพท์ทางไกลต่างจังหวัด

สำหรับอัตรานี้ถูกกำหนดไว้เป็นมาตรฐานโดยองค์การโทรศัพท์แห่งประเทศไทย โดยแสดงอยู่ในอัตราของราคาต่อเวลา 1 นาที โดยการคิดเศษของนาทีปัดขึ้นเป็น 1 นาที ตารางที่ แสดงอัตราค่าบริการโทรศัพท์ทางไกลต่างจังหวัดเมื่อคิดจุดโทรศัพท์ออกจากกรุงเทพฯ โดยแยกเป็นเขตภูมิภาค เขต รหัสทางไกล แล้วแยกย่อยออกไปเป็นเขตแต่ละจังหวัด แล้วแสดงค่าบริการในแต่ละจังหวัด

ตารางที่ 2.2 แสดงอัตราค่าบริการโทรศัพท์ทางไกลต่างจังหวัด

รหัสทางไกล 3 ตัวแรก	เลขนำหน้า ตัวแรก	จังหวัด	อัตราค่าบริการ (บาท/นาที)
032	2,3	ราชบุรี	6
	4	เพชรบุรี	6
	5,6	ประจวบคีรีขันธ์	12
034	2,3	นครปฐม	3
	4	สมุทรสาคร	3
	5,6	กาญจนบุรี	9
	7	สมุทรสงคราม	6
035	2,3	อยุธยา	6
	5	สุพรรณบุรี	6
	6	อ่างทอง	6
036	2,3	สระบุรี	6
	4	ลพบุรี	9
	5	สิงห์บุรี	9
037	2	ปราจีนบุรี,สระแก้ว	6
	3	นครนายก	6
038	2,3,4	ชลบุรี	6
	5	ฉะเชิงเทรา	6
	6	ระยอง	9
039	3,4	จันทบุรี	12
	5	ตราด	12
042	2,3	อุดรธานี,หนองบัวลำภู	15
	4	หนองคาย	18
	5	นครพนม	18
	6	มุกดาหาร	18
	7	สกลนคร	18
	8	เลย	15
043	2,3,4	ขอนแก่น	15
	5	ร้อยเอ็ด	15
	7	มหาสารคาม	15
	8	กาฬสินธุ์	15

รหัสทางไกล 3 ตัวแรก	เลขนำหน้า ตัวแรก	จังหวัด	อัตราค่าบริการ (บาท/นาที)
044	2,3,4	นครราชสีมา	12
	5	สุรินทร์	12
	6	บุรีรัมย์	12
	8	ชัยภูมิ	12
045	2,3,4,5	อุบลราชธานี,อำนาจเจริญ	15
	6	ศรีษะเกษ	15
	7	ยโสธร	15
053	2,3	เชียงใหม่	18
	5	ลำพูน	18
	6	แม่ฮ่องสอน	18
	7	เชียงราย	18
054	2	ลำปาง	18
	4	พะเยา	18
	5,6	แพร่	15
	7	น่าน	18
055	2,3	พิษณุโลก	12
	4	อุตรดิตถ์	15
	5	ตาก	15
	6	สุโขทัย	15
	7	กำแพงเพชร	12
056	2,3	นครสวรรค์	12
	4	ชัยนาท	9
	5	อุทัยธานี	9
	6	พิจิตร	9
	7	เพชรบูรณ์	12
073	2	ยะลา	18
	3,4	ปัตตานี	18
	5,6	นราธิวาส	18
074	2,3,4,5	สงขลา	18
	6	พัทลุง	18
	7	สตูล	18

รหัสทางไกล 3 ตัวแรก	เลขนำหน้าตัวแรก	จังหวัด	อัตราค่าบริการ (บาท/นาที)
076	2,3	ภูเก็ต	18
	4	พังงา	18
077	2,3,4	สุราษฎร์ธานี	18
	5	ชุมพร	15
	8	ระนอง	15
075	2	ตรัง	18
	3,4,5	นครศรีธรรมราช	18
	6	กระบี่	18

หมายเหตุ : -ค่าบริการที่แสดงในตารางข้างบนนี้คิดเมื่อต้นทางอยู่ในเขตพื้นที่นครหลวง

-อัตราค่าบริการที่แสดงในตารางเป็นอัตราภาคกลางวัน

อัตราค่าบริการภายในท้องถิ่นหรือภายในจังหวัด

การคิดอัตราค่าบริการในลักษณะจะเป็นแบบราคาเดียว คืออัตราค่าบริการไม่ได้ขึ้นกับเวลาในการใช้งาน และสำหรับประเทศไทยนั้นได้กำหนดให้อัตรานี้ คือ 3 บาท ต่อการใช้โทรศัพท์ภายในจังหวัด 1 ครั้ง และมีหน่วยงานพิเศษบางหน่วยงานที่ผู้โทรเข้าไม่ต้องเสียค่าบริการ ซึ่งหมายเลขโทรศัพท์ของหน่วยงานเหล่านี้จะมีหมายเลขโทรศัพท์ที่ขึ้นต้นด้วย 1 เสมอ แต่ไม่ได้หมายความว่า หมายเลขโทรศัพท์ที่ขึ้นต้นด้วย 1 จะเป็นหมายเลขที่โทรเข้าโดยไม่เสียค่าบริการเสมอไป หน่วยงานพิเศษเหล่านี้นั้นได้แก่

ตารางที่ 2.3 แสดงหมายเลขโทรศัพท์ที่ขึ้นต้นด้วย 1 แต่ไม่ต้องเสียค่าบริการใดๆ

หมายเลขโทรศัพท์	ชื่อหน่วยงานหรือสถานที่บริการ
1133	บริการสอบถามเบอร์โทรศัพท์ในเขตนครหลวง
183	บริการสอบถามเบอร์โทรศัพท์ในเขตภูมิภาคและเลขหมายของโทรศัพท์เคลื่อนที่
191	แจ้งเหตุด่วนเหตุร้าย
193	ตำรวจทางหลวง
197	ศูนย์ควบคุมการจราจร
199	ศูนย์ดับเพลิงศรีอยุธยา
195	กองปราบปราม
1699	ตำรวจท่องเที่ยว
17xxxxxx	แจ้งเหตุขัดข้องในการใช้โทรศัพท์ (กด 17 ตามด้วยหมายเลข 3 ตัวแรก ของหมายเลขโทรศัพท์ที่ขัดข้อง)
1644	สถานีวิทยุ สวพ. 91
108	PINPHONE

และยังมีหมายเลขพิเศษ ซึ่งขึ้นต้นด้วย 1 เช่นกัน และที่เป็นหมายเลข 7 ตัวตามปกติ (เฉพาะในเขตกรุงเทพฯ คิดค่าบริการครั้งละ 3 บาท) สำหรับส่วนภูมิภาคจะมีบางหมายเลขที่ไม่สามารถเรียกได้เช่นเดียวกับในกรุงเทพฯ และอัตราค่าบริการในกรณีที่หน่วยงานนั้น ๆ ตั้งอยู่ในจังหวัดของผู้เรียกนั้นจะคิดค่าบริการครั้งละ 3 บาท เช่นเดียวกับในกรุงเทพฯ ส่วนกรณีที่หน่วยงานนั้นๆ มิได้ตั้งอยู่ในจังหวัดของผู้เรียกก็ยังสามารถเรียกได้ และจะเรียกไปยังหน่วยงานนั้นๆ ที่ตั้งอยู่ในเขตรหัสทางไกลเดียวกันการคิดค่าบริการจะคิดในอัตราเดียวกับการโทรทางไกลต่างจังหวัดในเขตเดียวกัน ตามที่กำหนดไว้ในอัตราปกติ และสำหรับหมายเลขที่เรียกได้ทั้งในส่วนกรุงเทพฯ และส่วนภูมิภาคได้แก่

ตารางที่ 2.4 แสดงหมายเลขโทรศัพท์ที่ขึ้นต้นด้วย 1 แต่ต้องเสียค่าบริการ

หมายเลขโทรศัพท์	บริการ
151,152	PHONE LINK
1141,1142,1144	PACK LINK
1500,1501	EASY CALL
161,162	HUTCHISON PAGE PHONE
1188	POSTEL
142	WORLDPAGE
1567	ศูนย์บริการข่างสารมหาดไทย
1665	สำนักงานอาสากาชาดสภากาชาด
1125	แจ้งท่อประปาแตก
181	สอบถามเวลา
182	ศูนย์บริการข่าวอากาศ
184	ศูนย์บริการรับร้องเรียนขนส่งมวลชนกรุงเทพฯ
1506	ศูนย์ข้อมูลสำนักงานประกันสังคม
1507	ศูนย์บริการให้คำปรึกษาทางโทรศัพท์กรมประชาสงเคราะห์

2.4 การคิดค่าบริการของระบบโทรศัพท์เคลื่อนที่

สำหรับการคิดค่าบริการในระบบนี้นั้น จะแบ่งรูปแบบของการคิดค่าบริการจากลักษณะของการเรียกจากระบบใดไประบบใดได้ดังนี้คือ

1. การเรียกจากโทรศัพท์เคลื่อนที่ไปยังโทรศัพท์เคลื่อนที่

สำหรับการเรียกในรูปแบบนี้จะคิดค่าบริการ โดยคิดจากจุดที่ผู้เรียกทำการเรียกไปยังจุดลงทะเบียนของเครื่องโทรศัพท์ที่ถูกเรียก โดยไม่เกี่ยวกับพื้นที่ที่ผู้ถูกเรียกอยู่เลย

2. การเรียกจากโทรศัพท์เคลื่อนที่ไปยังโทรศัพท์พื้นฐาน

สำหรับการเรียกในรูปแบบนี้จะคิดค่าบริการ โดยคิดจากจุดที่ผู้เรียกทำการเรียกไปยังพื้นที่หรือจังหวัดของโทรศัพท์พื้นฐานที่ถูกเรียก

3. การเรียกจากโทรศัพท์พื้นฐานไปยงโทรศัพท์เคลื่อนที่

สำหรับการเรียกในรูปแบบนี้จะคิดค่ายริการ โดยคิดจากพื้นที่จังหวัดที่ตั้งของโทรศัพท์พื้นฐานเครื่องนั้นไปยังจุดลงทะเบียนของโทรศัพท์เคลื่อนที่ โดยไม่เกี่ยวกับพื้นที่ที่ผู้ถูกเรียกอยู่ในขณะเวลานั้นเลย

สำหรับกรณีทั้ง 3 นี้ กรณีที่เราสนใจก็คือกรณีที่ 3 เนื่องจากโครงงานนี้เป็นอุปกรณ์ที่ถูกใช้กับโทรศัพท์พื้นฐาน ดังนั้นหากเราได้ทราบข้อมูลของโทรศัพท์เคลื่อนที่ทั้งหมดที่มีอยู่ในขณะนี้ ว่าการแบ่งเขตการให้บริการของโทรศัพท์เคลื่อนที่แต่ละระบบนั้น ในแต่ละเขตประกอบด้วยหมายเลขใดบ้างที่ให้บริการจดทะเบียนในเขตนั้น ๆ เราก็จะสามารถคำนวณค่าบริการออกมาได้ในทุกระบบ

ในกรณีนี้เรามีข้อมูลของระบบเซลูล่าร์ 900 และระบบดิจิตอล จีเอสเอ็ม ที่ให้บริการโดย บมจ. แอดวานซ์ อินโฟร์ เซอร์วิส เพียงสองระบบ การคำนวณจึงสามารถที่จะคำนวณได้เฉพาะหมายเลขโทรศัพท์ที่ถูกให้บริการ โดยสองระบบนี้เท่านั้น โดยจากการแบ่งเขตให้บริการของโทรศัพท์เคลื่อนที่โดยทั่วไปทุกระบบจะมีอัตราค่าบริการอยู่เพียง 3 ระดับเท่านั้น โดยแบ่งได้ดังนี้คือ

1. การเรียกระหว่างเขตรหัสโทรศัพท์เคลื่อนที่เดียวกัน คิดอัตรา 3 บาทต่อนาที
2. การเรียกระหว่างเขตรหัสโทรศัพท์เคลื่อนที่ติดกัน คิดอัตรา 8 บาทต่อนาที
3. การเรียกระหว่างเขตรหัสโทรศัพท์เคลื่อนที่ คิดอัตรา 12 บาทต่อนาที

ตารางที่ 2.5 แสดงอัตราค่าบริการ โทรศัพท์เคลื่อนที่ (เมื่อคิดจากจุดเรียกคือกรุงเทพฯ)

เรียกไปยังโทรศัพท์เคลื่อนที่ที่จดทะเบียนในเขตการให้บริการ	เขตให้บริการ	อัตราค่าบริการ (บาท/นาที)
1	เขตภาคกลาง	3
2	เขตภาคอีสารตอนบน	12
3	เขตภาคอีสานตอนกลาง	8
4	เขตภาคเหนือตอนบน	12
5	เขตภาคเหนือตอนล่าง	8
6	เขตภาคใต้ตอนล่าง	12
7	เขตภาคใต้ตอนบน	8

ตารางที่ 2.6 การแบ่งเขตการให้บริการโทรศัพท์เคลื่อนที่ระบบเซลลูล่าร์ 900 และดิจิตอล จีเอสเอ็ม

เขตที่	รหัสทางไกล	จังหวัด	หมายโทรศัพท์ที่ให้บริการจดทะเบียน	
			ระบบเซลลูล่าร์ 900	ระบบดิจิตอล จีเอสเอ็ม
1	02	กรุงเทพฯ (BKK)	01-900-939-xxxx	01-811-815-xxxx
		นนทบุรี (BKK)	01-984-986-xxxx	01-821-823-xxxx
		สมุทรปราการ (BKK)	01-988-989-xxxx	01-824-829-xxxx
		ปทุมธานี (BKK)	01-251-255-xxxx	01-831-835-xxxx
	032	ประจวบคีรีขันธ์ (PKN)		01-841-845-xxxx
		เพชรบุรี (PBI)		01-856-857-xxxx
		ราชบุรี (RBR)		
	034	กาญจนบุรี (KPI)		
		นครปฐม (NPT)		
		สมุทรสาคร (SKM)		
		สุพรรณบุรี (SPB)		
		อยุธยา (AYA)		
	035	อ่างทอง (ATG)		
		ลพบุรี (LRI)		
		สิงห์บุรี (SBR)		
	036	สระบุรี (SRI)		
		ปราจีนบุรี (PRI)		
		นครนายก (NYK)		
	037	สระแก้ว (SKW)		
		ชลบุรี (CBI)		
		ฉะเชิงเทรา (CCO)		
	038	ระยอง (RYG)		
		จันทบุรี (CTI)		
		ตราด (TRT)		
	039			
2	042	อุดรธานี (UDN)	01-955-xxxx	01-871-872-xxxx
		หนองคาย (NKI)	01-966-967-xxxx	
		สกลนคร (SNK)	01-976-977-xxxx	
		นครพนม (NPM)	01-999-xxxx	
		มุกดาหาร (MDH)		

เขตที่	รหัสทางไกล	จังหวัด	หมายโทรศัพท์ที่ให้บริการจดทะเบียน	
			ระบบเซลลูล่าร์ 900	ระบบดิจิตอล จีเอสเอ็ม
	043	เลย (LEI)		
		หนองบัวลำพู (NBP)		
		ขอนแก่น (KKN)		
		มหาสารคาม (MKM)		
		ร้อยเอ็ด (RET)		
		กาฬสินธุ์ (KSN)		
3	044	นครราชสีมา (MNA)	01-954-XXXX	01-876-877-XXXX
		บุรีรัมย์ (BRM)	01-964-969-XXXX	
		ชัยภูมิ (CPM)	01-974-975-XXXX	
		สุรินทร์ (SRN)		
	045	อุบลราชธานี (UBN)		
		ศรีสะเกษ (SSK)		
		ยโสธร (YST)		
		อำนาจเจริญ (ACR)		
4	053	เชียงราย (CRI)	01-950-952-XXXX	01-881-882-XXXX
		เชียงใหม่ (CMI)	01-960-961-XXXX	
		ลำพูน (LPN)	01-980-XXXX	
		แม่ฮ่องสอน (MSN)	01-992-933-XXXX	
	054	ลำปาง (LPG)		
		พะเยา (PYO)		
		แพร่ (PRE)		
		น่าน (NAN)		
5	055	พิษณุโลก (PLK)	01-953-XXXX	01-851-852-XXXX
		สุโขทัย (STI)	01-962-XXXX	01-886-XXXX
		กำแพงเพชร (KPT)	01-971-973-XXXX	01-887-XXXX
		ตาก (TAK)		
		อุตรดิตถ์ (UTI)		
	056	นครสวรรค์ (NSN)		
		พิจิตร (PCT)		
		เพชรบูรณ์ (PBN)		
		อุทัยธานี (UTI)		
		ชัยนาท (CNT)		

เขตที่	รหัสทางไกล	จังหวัด	หมายโทรศัพท์ที่ให้บริการจดทะเบียน	
			ระบบเซลลูล่าร์ 900	ระบบดิจิตอล จีเอสเอ็ม
6	073	ยะลา (YLA)	01-957-959-XXXX	01-896-897-XXXX
		ปัตตานี (PTN)	01-963-969-XXXX	
		นราธิวาส (NWT)	01-990-XXXX	
		สงขลา (SKA)		
		พัทลุง (PLG)		
		สตูล (STN)		
	075	นครศรีธรรมราช (NRT)	01-956-959-XXXX	01-891-892-XXXX
		ตรัง (TRG)	01-968-970-XXXX	
		กระบี่ (KBI)	01-978-979-XXXX	
		ภูเก็ต (PKT)		
	045	พังงา (PNA)		
		สุราษฎร์ธานี (SNI)		
		ชุมพร (CPN)		
		ระนอง (RNG)		

2.5 การทำงานของ IC เบอร์ MT8870

MT8870 เป็นไอซีถอดรหัสความถี่ (Integrated DTMF Receiver) หรือเป็นตัวแปลงสัญญาณ DTMF เป็นสัญญาณดิจิตอล

- คุณสมบัติของ MT8870

1. เป็นตัวรับและถอดรหัสความถี่ (DTMF Receiver)
2. กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
3. สามารถตั้งอัตราขยายภายในตัวไอซีได้
4. สามารถปรับการ์ดไทม์ (Guard time) ได้
5. เป็นไอซีคุณภาพสูง

- การนำ MT8870 ไปใช้งาน

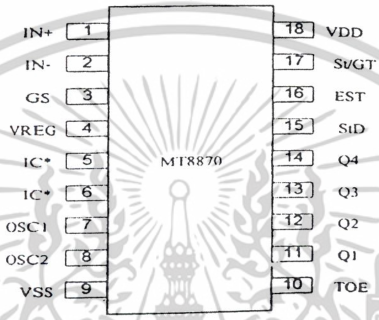
1.นำไปใช้งานด้านรีโมตคอนโทรล
2.เครื่องป้องกันโทรศัพท์ทางไกล
3.ใช้งานเกี่ยวกับเครดิตการ์ด
4.ใช้งานร่วมกับคอมพิวเตอร์
5.ใช้ในเครื่องชุมสายขนาดย่อยหรือ PABX
6.ใช้กับงานทางด้านโทรศัพท์ทั่วไป
7.เครื่องกันขโมย

8.การควบคุมอุปกรณ์ทางโทรศัพท์

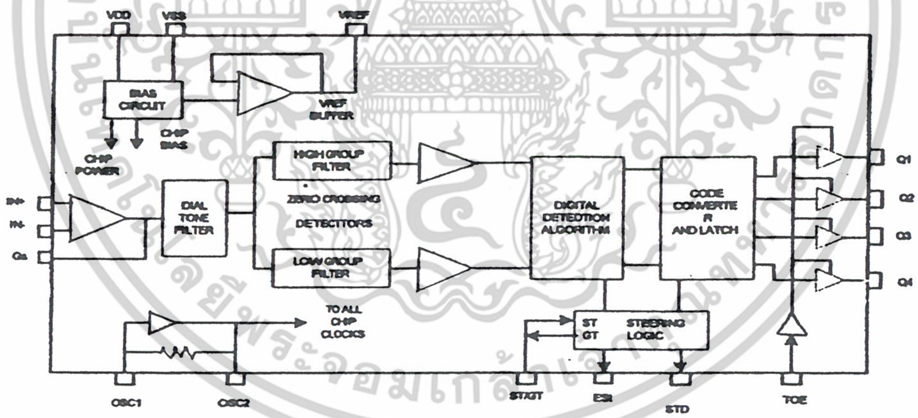
9.ใช้ทำเครื่องสอบถามทางโทรศัพท์

- โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรรองความถี่และวงจรถอดรหัสฟังก์ชั่นทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO-CMOS ในส่วนของคาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่ใช้เทคนิคของสวิตช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสทั้ง 16 ความถี่ใช้ออกเป็นเลขฐานสองขนาด 4 บิต และเช็คช่วงเวลาที่สัญญาณเข้ามา ส่วนภาคอินพุตเป็นออปแอมป์ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอกเอาต์พุตเป็นวงจรแลตช์ 3 สถานะ รูปที่ 2.2 แสดงขา ของ MT8870 และรูปที่ 2.3 แสดงโครงสร้างภายในของ MT8870



รูปที่ 2.2 แสดงรายละเอียดของ MT8870



รูปที่ 2.3 แสดงโครงสร้างภายในของ MT8870

- ฟังก์ชั่นการทำงานภายใน MT8870

ภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

1.ภาคกรองสัญญาณความถี่ (Filter Section)

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือ ช่วงความถี่สูงและช่วงความถี่ต่ำโดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิด สวิตช์คาปาซิเตอร์ (Six-Order Switched Capacitor Band Pass Filter) ซึ่งความถี่ที่แยกได้ 2 ช่วง คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ

2.ภาคถอดรหัส (Decoder Section)

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคการนับแบบดิจิตอลและมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐานDTMF หรือไม่ เพื่อป้องกันความถี่เข้ามาผสมเมื่อตรวจว่าความถี่นั้นถูกต้อง สัญญาณที่ขา EST (Early Steering) ก็จะแอคทีฟ สำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆ นั้นแสดงในตารางที่ 2.7

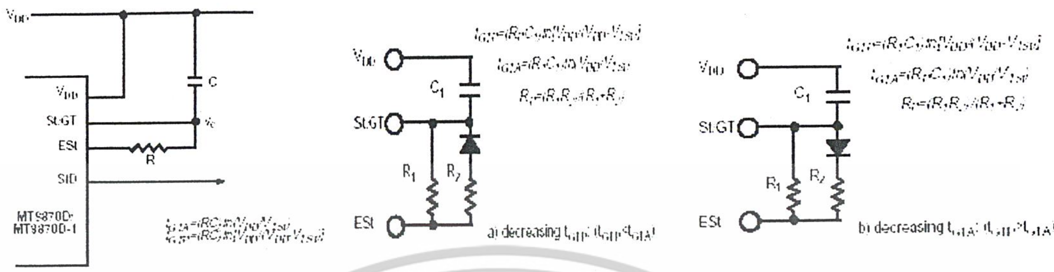
ตารางที่ 2.7 แสดงค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ

F_{LOW}	F_{HIGH}	NO	TOE	Q4	Q3	Q2	Q1
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	14477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1447	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

3.ภาคตรวจสอบสัญญาณ (Steering Circuit)

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการดกปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าไร สามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา EST จะเป็น “High” นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 2.4 เมื่อขา EST เป็น “High” ทำให้ VC สูงขึ้นตั้งเก็บประจุ C จะคายประจุทำให้แรงดัน VC สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสออกเป็นตัวเลขขนาด 4 บิต สำหรับคำว่าการ์ดไทม์ (Guard time) นั้นหมายถึงช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับ

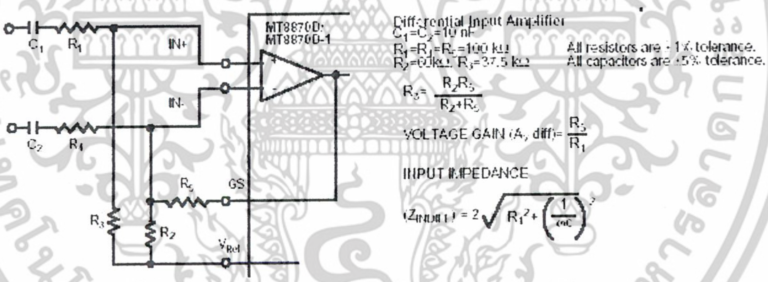
หรือมากกว่าช่วงเวลาที่เราตั้งไว้ จึงจะได้รับการยอมรับว่าสัญญาณความถี่นั้นถูกต้องหรือพูดได้ว่าเวลาที่เราตั้งไว้โดย RC ก็คือการ์ดไทม์นั้นเองเมื่อสัญญาณความถี่เข้ามานานเท่าหรือมากกว่าที่ตั้งไว้จึงสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่าก็จะมีการถอดรหัสเป็นตัวเลขได้ ถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลาคำนวณดูได้จากรูปที่ 2.4



รูปที่ 2.4 แสดงการกำหนดการ์ดไทม์ (Guard Time) พร้อมวิธีคำนวณ

4.ภาคขยายสัญญาณความแตกต่าง (Differential Input)

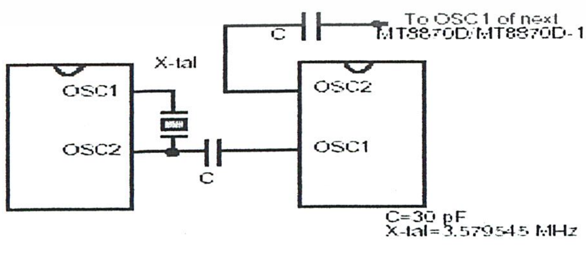
วงจรส่วนอินพุตของ MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเพิ่มเข้ามารูปที่ 2.5 แสดงการต่อวงจรภายนอกเข้ากับอินพุตซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอิมพีแดนซ์ได้ดังนี้



รูปที่ 2.5 แสดงการต่อวงจรภาคอินพุต

5.ภาคกำเนิดความถี่ (Oscillator)

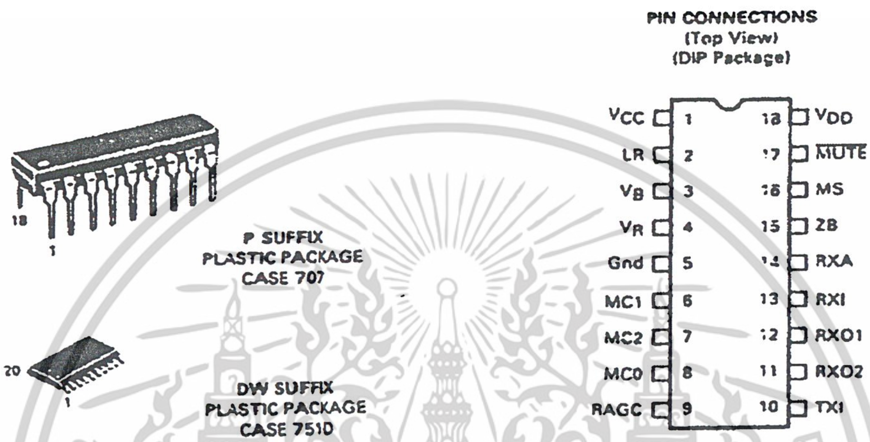
ในภาคนี้ภายในไอซีจะมีเวลาอยู่ภายใน เพียงแต่ต่อแร่คริสตอลขนาด 3.58 MH_z ก็จะสามารถใช้ได้ทันที การต่อวงจรกำเนิดความถี่แสดงในรูปที่ 2.6



รูปที่ 2.6 แสดงการต่อวงจรผลิตความถี่

2.6 วงจรควบคุมเสียงพูด MC34114

ไอซีเบอร์ MC34114 ซึ่งเป็นอุปกรณ์ที่รวมเอาทุกส่วนที่เกี่ยวข้องกับการทำงานในวงจรหรือเน็ตเวิร์คการสนทนาทั้งด้านรับและส่งในเวลาเดียวกัน เปรียบเสมือนกับว่าทำงานแทนวงจรไฮบริดจ์ภายในตัว MC34114 ได้รับรวบรวมเอาส่วนที่ทำหน้าที่เรียกได้ว่าเป็นวงจรแมกเนติกไฮบริดจ์ (magnetic hybrid) ซึ่งประกอบไปด้วยส่วนทำงานเป็นภาคขยายด้านส่ง , ภาคขยายด้านรับ , ควบคุมสัญญาณไซค์โทน (side tone) , สามารถปรับอัตราการขยายได้จากภายนอก , ภาคขยายสัญญาณไมโครโฟนสัญญาณรบกวนต่ำ และป้องกันการรบกวนเนื่องจากความถี่วิทยุ (RFI)



รูปที่ 2.7 รูปร่างและลักษณะการจัดขาของไอซี

2.6.1 คุณสมบัติของ MC34114

- สามารถทำงานได้ที่แรงดันต่ำสุด 1.2 โวลต์
- ควบคุมอัตราการขยายด้านส่ง , ด้านรับ และสัญญาณไซค์โทนได้จากภายนอก
- ภาคขยายความแตกต่างสัญญาณไมโครโฟน ไม่มีปัญหาจากการรบกวนเนื่องจากความถี่วิทยุ
- นอกจากการส่งและรับรวมทั้งสัญญาณไซค์โทนของสัญญาณเสียงพูดแล้ว สัญญาณดีทีเอ็มเอฟก็สามารถมอดูเลตออกสู่คู่สายได้
- มีแรงดันเรกูเลเตอร์เอาท์พุต 1.7 โวลต์ สำหรับใช้กับไมโครโฟนคอนเดนเซอร์
- มีแรงดันเรกูเลเตอร์เอาท์พุต 3.3 โวลต์ สำหรับวงจรกำเนิดสัญญาณดีทีเอ็มเอฟ และวงจรหน่วยความจำ
- สามารถนำไปประยุกต์ใช้งานร่วมกับอุปกรณ์อื่น หรือในระบบโทรศัพท์มาตรฐานทั่วไปได้

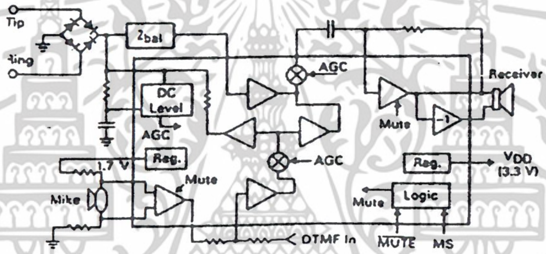
2.6.2 หน้าที่ของ MC34114 และการทำงานเบื้องต้น

หน้าที่หลักของไอซีตัวนี้ดังที่กล่าวมาแล้วข้างต้นว่าทำงานเป็นวงจรไฮบริดจ์ สังเกตได้จากในวงจรโทรศัพท์แบบเก่า จะอาศัยหม้อแปลงไฮบริดจ์ทำหน้าที่รับสัญญาณและป้อนกลับสัญญาณทำให้สามารถทำการสนทนาสวนทางกันได้ คือการรับและการส่งเกิดขึ้นพร้อมกัน ในปัจจุบันนี้ได้ใช้ไอซีชนิดนี้ในการรวมเอาส่วนประกอบการทำงานในส่วนอื่นเข้ามาไว้รวมกันภายในตัวถังไอซีตัวเดียว เพื่อง่ายแก่

การออกแบบใช้งานและพัฒนาวงจรให้ทันสมัยขึ้น จะเห็นการรวมเอาส่วนต่าง ๆ ไว้ด้วยกันดังบล็อกไดอะแกรมภายในตัวไอซีในรูปที่ 2.8

ตารางที่ 2.8 ข้อมูลทางเทคนิคของ MC34114

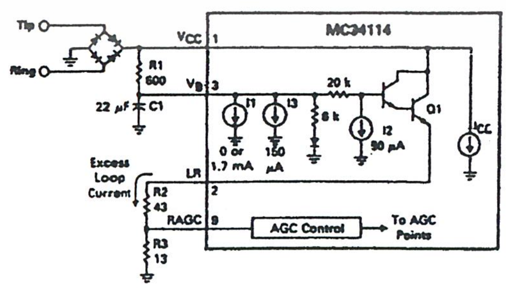
ข้อมูล	ค่า
แรงดันไฟเลี้ยง V_{cc}	+1.2 ถึง 10.5 โวลต์
กระแสไหลในวงจรตัวไอซี	4.0 ถึง 120 มิลลิแอมป์
ค่าความต้านทานด้านรับ	50 โอห์ม
ค่าความต้านทานระหว่างขา V_{cc} กับ VB (Rx)	100 ถึง 1800 โอห์ม
อุณหภูมิใช้งาน	-60 ถึง 150 องศาเซลเซียส
อัตราการขยายของไมโครโฟน	28 ถึง 32 เดซิเบล
ความต้านทานทางอินพุตของไมโครโฟน	14 ถึง 27 กิโลโอห์ม
ความต้านทานทางเอาท์พุตของไมโคร โฟน	270 โอห์ม
แรงดันเอาท์พุตวิงสูงสุด	2 โวลต์พีคทูพีค



รูปที่ 2.8 บล็อกไดอะแกรมภายในตัวไอซี MC34114

2.6.3 การอินเตอร์เฟสกับคู่สาย

สำหรับการอินเตอร์เฟสกับแรงดันไฟตรงในคู่สายโทรศัพท์นั้นจะมีวงจรอินเตอร์เฟสภายในโดยมีการต่อใช้งานที่ขา 1 (V_{cc}) , ขา 2 (LR) และขา 3 (VB) เพื่อทำการเซตระดับแรงดันไฟตรงให้เหมาะสมกับระบบทำงาน ดังวงจรสมมูลย์ในรูปที่ 2.9 แสดงถึงการไหลของกระแสผ่านทรานซิสเตอร์ภายใน (Q1) ผ่านตัวต้านทาน R2 , R3 ซึ่งทำหน้าที่แบ่งแรงดันให้กับวงจรควบคุมอัตราขยายอัตโนมัติ (AGC) ภายในไอซี



รูปที่ 2.9 วงจรสมมูลย์ของการอินเตอร์เฟสกับคู่สายโทรศัพท์

กระแส Icc ในวงจร คือ ค่ากระแสที่ใช้ไบแอสให้กับวงจรการทำงานภายในของ MC34114 ซึ่งปกติจะมีค่ากระแสประมาณ 10 มิลลิแอมป์ และการที่จะทำให้กระแสเกิดการไหลขึ้นในวงจรจะมีสภาวะการทำงานอยู่ 3 สภาวะ คือ เมื่อมีการสนทนาเกิดขึ้นพร้อมกับอยู่ในโหมดของการส่งสัญญาณแบบพัลส์ และอยู่ในโหมดของการส่งสัญญาณแบบโทน

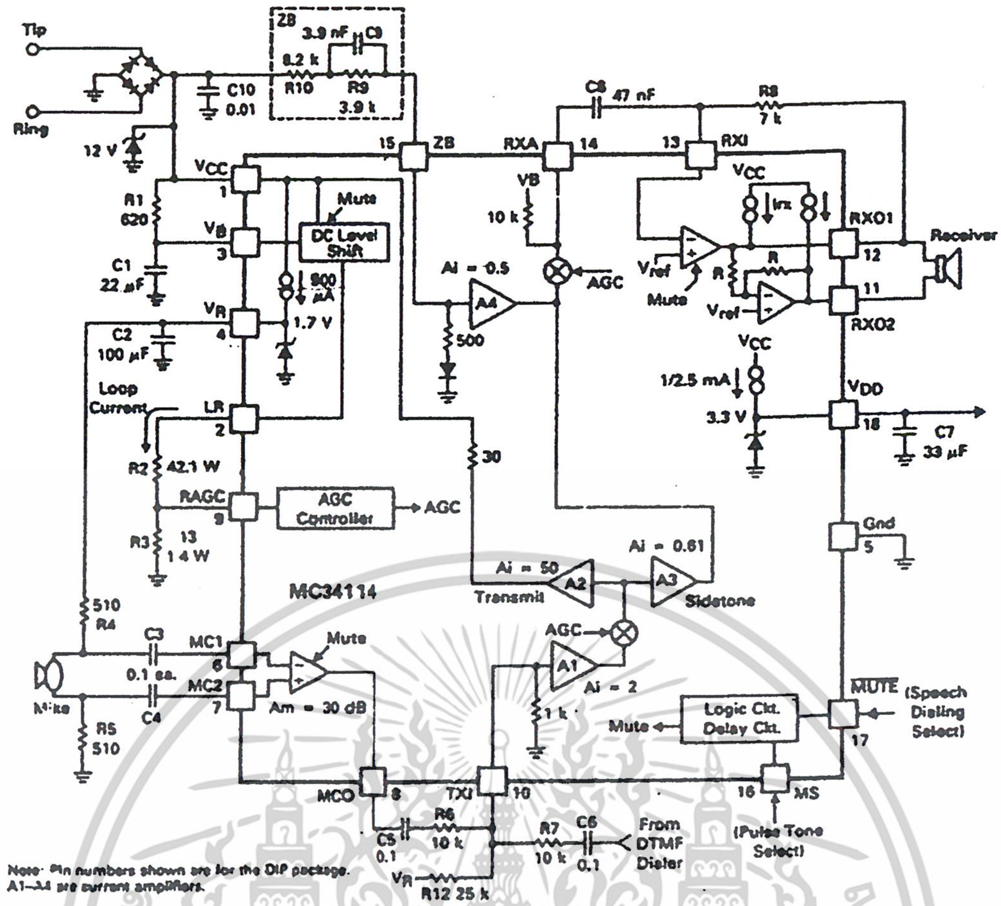
ในโหมดการสนทนาและการส่งสัญญาณแบบพัลส์ กระแสจากแหล่งจ่าย II จะหยุดนำกระแส เป็นผลทำให้แรงดันตกคร่อมระหว่างขาเบสและอิมิตเตอร์ Q1 เพิ่มขึ้น มีค่าประมาณ 1.4 โวลต์ พร้อมกันนั้นจะมีแรงดันตกคร่อมตัวต้านทานที่ขาเบสของ Q1 ค่า 20 กิโลโอห์ม ประมาณ 1 โวลต์ ซึ่งแรงดันที่ตกคร่อม R1 นั้นสามารถเปลี่ยนแปลงได้ตั้งแต่ค่า 0.15 โวลต์ ถึง 1 โวลต์ เมื่อมีกระแสหรือสัญญาณจากคู่สายโทรศัพท์เข้ามาจะทำให้เกิดการไหลของกระแสขึ้นภายในลูป Icc และกระแสไหลผ่าน Q1 , R2 และ R3

ในโหมดการสนทนาระบบหมุนหมายเลข กระแสจากแหล่งจ่าย II จะนำกระแส ซึ่งจะมีกระแสไหลผ่าน R1 ประมาณ 1.7 มิลลิแอมป์ และมีแรงดันตกคร่อม R1 เพิ่มขึ้นเป็น 1 โวลต์ (R1 = 600 โอห์ม) และจะมีแรงดัน V_{cc} สำหรับสัญญาณดีทีเอ็มเอฟ และกระแส I_{cc} จะมีค่าประมาณ 1.3 มิลลิแอมป์ในโหมดนี้ ซึ่งค่าของ R1 นั้นสามารถเปลี่ยนแปลงค่าความต้านทานได้ในช่วง 100-1800 โอห์ม และแรงดันที่ตกคร่อม R3 ก็คือแรงดันที่ควบคุมการทำงานของวงจรควบคุมอัตราขยายอัตโนมัติอยู่ในช่วง 0.4 โวลต์ เพื่อควบคุมอัตราการขยายกระแสใน 2 จุดภายในตัวไอซี

2.6.4 ส่วนจ่ายแรงดันคงที่ (Voltage Regulators)

ไอซี MC34114 มีตัวจ่ายแรงดันคงที่ 2 ตัว เพื่อจ่ายแรงดันให้แก่ทั้งวงจรภายในและวงจรภายนอก ตัวจ่ายแรงดันคงที่ VR จ่ายแรงดัน 1.7 โวลต์ ค่ากระแสสูงสุด 500 ไมโครแอมป์ (ตามรูปที่ 2.10) ซึ่งผลที่ได้นี้จะนำไปใช้ไบแอสขา 10 (TXI) และไบแอสไมโครโฟนชนิดคอนเดนเซอร์ โดยปกติ VR มีค่าน้อยกว่า V_{cc} ประมาณ 0.3 โวลต์ เมื่อ V_{cc} มีค่าน้อยกว่า 2.0 โวลต์

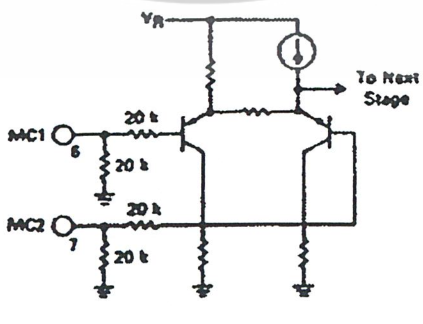
ตัวจ่ายแรงดันคงที่ V_{DD} จ่ายแรงดัน 3.3 โวลต์ที่กระแสสูงสุด 1.0 มิลลิแอมป์ในโหมดการสนทนา และมีค่ากระแส 2.5 มิลลิแอมป์ในการส่งสัญญาณแบบพัลส์หรือโทน (ตามรูปที่ 2.10) ปกติจะใช้ Vdd ในการจ่ายพลังงานให้กับวงจรแป้นกดที่อยู่ภายนอกรวมทั้งวงจรอื่นที่ต่ออยู่ด้วยกัน ปกติ V_{DD} จะมีค่าน้อยกว่า V_{cc} ประมาณ 0.5 โวลต์



รูปที่ 2.10 แสดงการต่อของอุปกรณ์ภายนอกกับ MC34114

2.6.5 วงจรขยายสัญญาณจากไมโครโฟน (Microphone Amp;ifier)

วงจรขยายสัญญาณจากไมโครโฟน (ขา 6 , 7 , 8) มีสัญญาณเข้าแบบผลต่าง (differential) สัญญาณออกแบบซิงเกิลเอนด์และอัตราขยายในคงที่ +30 เดซิเบล (31.1 V/V) เอาท์พุตตรงเฟสกันกับ MC2 และกลับเฟสกันกับ MC1 อินพุต (ตามรูปที่ 2.11) มีความต้านทาน 20 กิโลโอห์ม และแมตชิ่งเป็นอย่างดีเพื่อ CMRR (Common Mode Rejection Ratio) ที่สูง ประมาณ 26 เดซิเบล เพื่อที่จะมีการขจัดสัญญาณจากการเหนี่ยวนำจากสายนำสัญญาณที่ไม่ต้องการ ไมโครโฟนควรจะมีการไบแอสจากความต้านทานที่มีค่าเท่ากันดังในรูปที่ 2.11



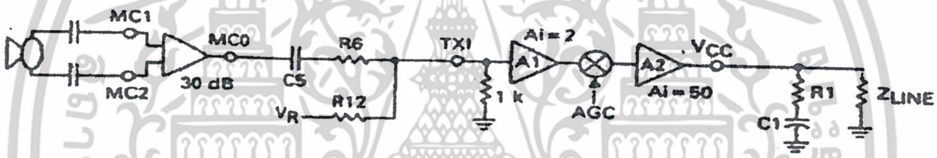
รูปที่ 2.11 วงจรภายในส่วนอินพุตของไมโครโฟน

เอาท์พุต (MCO) มีแรงดันไบแอสตรง อยู่ประมาณ 1.1 โวลต์ (เมื่อ Vcc มีค่ามากกว่า 3.0 โวลต์) มีอัตราการสวิงประมาณ 2.0 โวลต์พีคทูพีค (ที่ 500 มิลลิโวลต์ เมื่อ V_{cc} มีค่า 1.2 โวลต์) เอาท์พุตอิมพีแดนซ์มีค่าประมาณ 270 โอห์ม และมีกระแสสูงสุดประมาณ 160 ไมโครแอมป์ ที่ 5% ของ THD (Total Harmonic Distortion)

เมื่อ MC34114 อยู่ในระหว่างการส่งสัญญาณหมุนหมายเลข วงจรขยายไมโครโฟนจะถูกลดกำลังการส่ง (MUTE) ลงไปประมาณ 70 เดซิเบล (300 เฮิรตซ์ - 4,000 กิโลเฮิรตซ์) ซึ่งเพียงพอในการหยุดการทำงานของไมโครโฟน ระดับแรงดันไฟตรงที่ MCO มีค่าประมาณ 80 มิลลิโวลต์เมื่อถูกลดกำลังส่งลง

2.6.6 วงจรในการส่งสัญญาณ (Transmit Path)

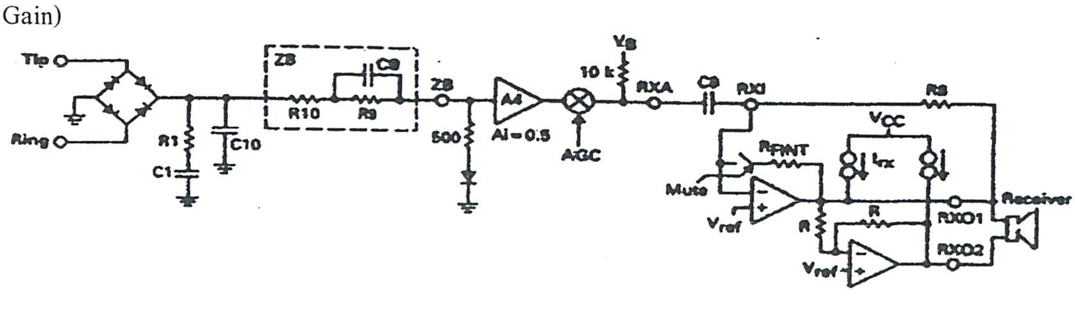
วงจรที่ใช้ในการส่งสัญญาณออกไปมีอุปกรณ์ดังรูป 2.12 แรงดันเอาท์พุตที่ MCO ถูกเปลี่ยนไปเป็นกระแสเข้าที่ TXI โดย C5 , R6 และความต้านทานภายในของ TXI คือ 1 กิโลโอห์ม A1 และ A2 คืออุปกรณ์ขยายกระแสที่มีอัตราการขยายรวมกันเป็น 100 วงจรควบคุมอัตราขยายอัตโนมัติที่เข้ามามีค่าเป็น 1 เมื่อมีกระแสลูปน้อย และลดลงเป็น 0.5 เมื่อกระแสลูปมีค่าเพิ่มขึ้น ดังนั้นจะทำให้อัตราการขยายจาก TXI ไปจนถึง V_{cc} มีค่าตั้งแต่ 100 ถึง 50 เป็นผลทำให้กระแสที่ V_{cc} กระทำต่อ R1 และอิมพีแดนซ์ของสายส่งประมาณ 600 โอห์ม ก่อให้เกิดแรงดันที่ V_{cc} และเช่นเดียวกันที่ขั้วทิปกับริง



รูปที่ 2.12 ส่วนของวงจรด้านส่ง

2.6.7 วงจรในการรับสัญญาณ (Receive Path)

วงจรที่ใช้รับสัญญาณเข้ามามีอุปกรณ์ดังในรูปที่ 2.22 R1 ซึ่งปกติมีค่า 600 โอห์ม จะเป็นตัวกำหนดจุดสิ้นสุดของสายส่งของสัญญาณที่ส่งมาจากขั้วทิปและริง สัญญาณที่ได้รับจะสร้างกระแสไฟสลับผ่าน ZB เน็ตเวิร์ค และความต้านทาน 500 โอห์ม ที่ขา ZB A4 จะลดกระแสลงครึ่งหนึ่งแล้วส่งต่อไปให้วงจรควบคุมอัตราขยายอัตโนมัติ จากนั้นจึงผ่าน C8 ไปยัง RXI (จุดรวมอัตราการขยาย ซึ่งถ้า C8 มีค่ามาก RXA จะเปรียบเป็นกราวด์เสมือน และไม่มีกระแสไฟสลับไหลผ่านความต้านทานภายใน 10 กิโลโอห์ม) แรงดันที่ RXO1 ถูกกำหนดกระแสจาก C8 และความต้านทานป้อนกลับ R8 ออปแอมป์ตัวที่ 2 (ที่ขา RXO2) มีการกำหนดไว้แล้วว่าให้มีการขยายแบบกลับขั้ว และมีอัตราการขยายเป็น 1 (Inverting Unity Gain)



รูปที่ 2.13 ส่วนของวงจรด้านรับ

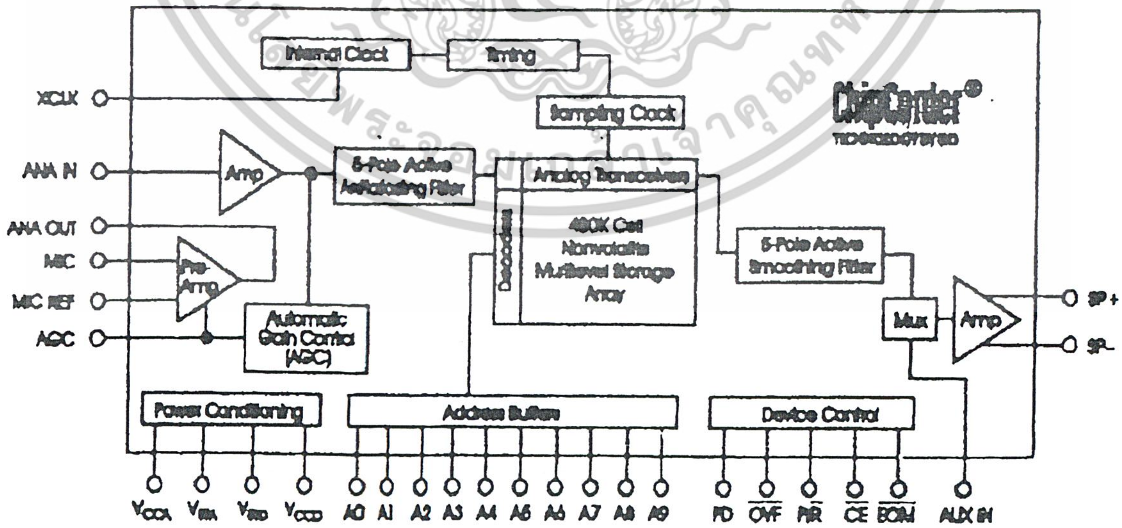
2.7 วงจรส่วนบันทึกและเล่นกลับ ISD 2590

ไอซี ISD 2590 รูปที่ 2.23 เป็นไอซีที่สามารถบันทึกเสียงลงไปในตัวไอซีได้โดยไม่ต้องต่อกับหน่วยความจำภายนอก เป็นอุปกรณ์อยู่ในตระกูลซีมอส สามารถบันทึกเสียงและเล่นกลับได้นาน 90 วินาที

2.7.1 คุณสมบัติโดยทั่วไปของ ISD 2590

1. ใช้งานง่ายในการบันทึกและเล่นกลับบนตัวไอซี
2. สัญญาณเสียงที่ถูกบันทึกเมื่อถูกเล่นกลับจะมีคุณภาพเสียงที่ดี
3. สามารถควบคุมการทำงานโดยใช้สวิตช์โดยทั่วไป หรือใช้ไมโครคอนโทรลเลอร์ควบคุมก็ได้
4. ไอซีหนึ่งตัวสามารถบันทึกเสียงได้นาน 90 วินาที
5. สามารถต่อแคสเคดได้โดยตรงเพื่อให้การบันทึกและเล่นกลับมีเวลานานขึ้น
6. ไม่ต้องใช้แบบเตอรี่ในการแบ็คอัพในขณะที่ไม่ได้ใช้งาน โดยไม่ทำให้ข้อมูลสูญหาย
7. มีวงรอบการบันทึกมากกว่า 100,000 ครั้ง
8. สามารถเก็บสัญญาณเสียงที่บันทึกไว้ได้นาน 100 ปี
9. มีสัญญาณนาฬิกาภายในตัวไอซี
10. ใช้แรงดันไฟตรง +5 โวลต์
11. สามารถกำหนดแอดเดรสในการแบ่งช่วงเวลาของการบันทึกและเล่นกลับได้

จากคุณสมบัติต่าง ๆ ที่รวบรวมอยู่ในไอซีเพียงตัวเดียว ทำให้ง่ายแก่การใช้งานตั้งแต่วงจรขยายสัญญาณจากไมโครโฟนจนถึงหน่วยจัดเก็บข้อมูลที่บันทึกและขับออกลำโพง ก็ถูกรวบรวมไว้ในไอซีเพียงตัวเดียว ในโหมดการบันทึกจะจัดเก็บข้อมูลต่าง ๆ ไว้ในหน่วยความจำที่เป็นเซลล์แบบไม่ต้องการแรงดันสำรองเพื่อรักษาข้อมูลไม่ให้สูญหาย



รูปที่ 2.14 บล็อกไดอะแกรมภายในไอซี ISD2590

2.7.2 หน้าที่ของขาอุปกรณ์

- ขา V_{CCA} , ขา V_{CCD} (Voltage Inputs)

เป็นขารับแรงดันที่จะต้องแยกกันต่างหากระหว่างขารับแรงดันของวงจรอนาล็อกและวงจรดิจิตอลที่ประกอบอยู่ภายในตัวไอซี ขารับแรงดันต้องการแรงดันไฟตรง +5 โวลต์ และต้องเป็นแรงดันไฟตรงที่มีสัญญาณรบกวนต่ำมาก

- ขา V_{SSA} , ขา V_{SSD} (Ground Inputs)

เป็นขากราวด์ของสัญญาณอนาล็อกและสัญญาณดิจิตอลตามลำดับ โดยขากราวด์ทั้งสองนี้จะถูกต่อ และปิดไว้ภายในตัวถังของไอซี การใช้งานของขากราวด์ทั้งสองจะเลือกต่อกับกราวด์ของแหล่งจ่ายไฟในส่วนที่มี ค่าอิมพีแดนซ์ต่ำ เพื่อไม่ต้องการให้เกิดค่าแรงดันที่แตกต่างกันระหว่างกราวด์ทั้งสอง

- ขา PD (Power Down Input)

ในขณะที่ไม่มีการบันทึกหรือเล่นกลับ ที่ขานี้จะมีสภาวะเป็น "1" ก็จะเป็นการรักษาระดับการสิ้นเปลือง พลังงานในระดับต่ำมาก ๆ แต่เมื่อขา OVF มีสภาวะเป็น "0" ที่แสดงถึงการเล่นกลับสิ้นสุดลงปรากฏขึ้น ขาPD ปกติจะเป็น "1" อยู่ในขณะนั้นจะถูกรีเซตและจะเริ่มกระบวนการบันทึกหรือเล่นกลับใหม่อีกครั้งหนึ่ง

- ขา CE (Chip Enable Input)

ขา CE จะต้องได้รับสัญญาณ ในสภาวะเป็น "0" เพื่อเข้าสู่สภาวะของการบันทึกหรือเล่นกลับ ที่ขาแอด เดรสอินพุตและขา P/R อินพุตจะถูกแลตช์จากพัลส์ขอบขาลงของพัลส์ที่ขา CE

- ขา P/R (Playback / Record Input)

เมื่อขานี้มีสภาวะเป็น "1" จะเป็นวงรอบของการเล่นกลับ และถ้ามีสภาวะเป็น "0" จะเป็นวงรอบของ การบันทึก ถ้าหากได้รับพัลส์ที่ขอบขาลงของขา CE จะเป็นการแลตช์อินพุตที่ขานี้

- ขา EOM (End-of-Message / Run Output)

ขานี้จะใช้กำหนดหรือระบุการสิ้นสุดของการเก็บสัญญาณเสียงที่ทำการบันทึก โดยขานี้จะให้เอาท์พุต ออกมาเป็น "0" เมื่อข้อมูลที่ถูกบันทึกอยู่ถูกเล่นกลับออกมาหมดแล้ว

- ขา OVF (Overflow Output)

สัญญาณพัลส์ "0" จะปรากฏออกทางขานี้ เพื่อเป็นการแสดงว่าสิ้นสุดการเล่นกลับหรือหน่วยความจำ ภายในตัวไอซีได้ถูกอ่านออกมาหมดแล้ว และจะแสดงเป็นสภาวะหยุดการเล่นกลับ พัลส์เอาท์พุตจากขานี้จะ จ่ายให้กับขา CE จนกว่าขา OVF นี้จะได้รับพัลส์ เพื่อทำการรีเซตและเริ่มวงรอบการเล่นกลับใหม่อีกครั้ง พัลส์ที่ ขา OVF นี้ สามารถใช้เริ่มต้นการทำงานของ ISD2590 ในตัวถัดไปได้เมื่อถูกต่อแคสเคดกันหลายตัว

- ขา MIC (Microphone Input)

ขานี้จะรับสัญญาณอินพุตที่ผ่านเข้ามายังไมโครโฟน แล้วส่งผ่านสัญญาณเข้าสู่วงจรปรีแอมป์ที่ประกอบอยู่ภายในตัวไอซี ซึ่งจะประกอบด้วยวงจรควบคุมอัตราการขยายอัตโนมัติ โดยวงจรนี้จะทำหน้าที่ควบ คุมอัตราการขยายของวงจรปรีแอมป์ให้มีการขยายอยู่ในช่วง -15 ถึง 24 เดซิเบล ไมโครโฟนจากภายนอกจะถูก คัปปลิ้งผ่านตัวเก็บประจุในลักษณะอนุกรมกับขานี้ ค่าความจุของตัวเก็บประจุจะถูก

กำหนดโดยคำนึงถึงความ ต้านทานภายในของไอซีซึ่งมีค่า 10 กิโลโอห์ม เพื่อให้เกิดการคัตออฟที่ความถี่ต่ำ

- ขา MIC REF (Microphone Reference Input)

ขานี้จะต่อกับกราวด์อนาล็อก โดยต่ออนุกรมกับตัวเก็บประจุเพื่อทำหน้าที่กำจัดสัญญาณรบกวนทาง อินพุตของขา MIC และเพื่อให้เกิดการชดเชยทางด้านสัญญาณรบกวนให้ดีกว่า 10 เดซิเบล

- ขา AGC (Automatic Gain Control Input)

ขานี้เป็นอินพุตเพื่อควบคุมอัตราการขยายของปรีแอมป์ไมโครโฟน เพื่อให้เกิดความเหมาะสมกับระดับ สัญญาณที่มีช่วงกว้างมากของสัญญาณทางด้านอินพุตจากไมโครโฟน และเพื่อให้ระดับสัญญาณที่ทำการอิน พุตมีความผิดเพี้ยนน้อยที่สุด ขานี้จะต่อร่วมกับอุปกรณ์ RC เพื่อกำหนดค่าเวลาคงที่โดยค่าความต้านทาน ภายใน 5 กิโลโอห์ม และจะต่อกับตัวเก็บประจุภายนอกอีกตัวหนึ่งเพื่อผ่านลงกราวด์อนาล็อกโดยค่าที่เหมาะสม จะกำหนดไว้ที่ R = 470 กิโลโอห์ม , C = 4.7 ไมโครฟารัด

- ขา ANA OUT (Analog Output)

ขานี้จัดเตรียมเพื่อต่อเข้ากับปรีแอมป์เอาท์พุต โดยอัตราการขยายแรงดันของปรีแอมป์จะถูกกำหนด โดยระดับแรงดันที่ขา AGC

- ขา ANA IN (Analog Input)

สัญญาณจะถูกส่งผ่านเข้ามาที่ขานี้เมื่อใช้ในการบันทึก สำหรับการต่อระหว่างขา ANA IN กับขา ANA OUT ควรจะมีตัวเก็บประจุต่อขั้นอยู่ด้วย ซึ่งค่าของตัวเก็บประจุจะต้องเหมาะสมกับค่าความต้านทานภายใน ของขา ANA IN โดยการเลือกให้ได้ค่าความถี่คัตออฟอยู่ในช่วงความถี่เสียงผ่าน ถ้าสัญญาณที่มาจากแหล่งอื่น ที่ไม่ใช่จากไมโครโฟนสามารถที่จะป้อนเข้ามาโดยผ่านตัวเก็บประจุ และจากนั้นจึงผ่านมาที่ขา ANA IN ได้โดย ตรง

- ขา XCLK (External Clock Input)

ขานี้เป็นขารับสัญญาณนาฬิกาจากภายนอก เพื่อกำหนดค่าความถี่สัญญาณนาฬิกาในการชักค่าสัญญาณ แต่โดยปกติสัญญาณนาฬิกาในการชักค่าสัญญาณจะถูกกำหนดไว้ภายในตัวไอซีซึ่งจะไม่ขึ้นกับอุณหภูมิภายนอกหรือย่านแรงดันไฟตรงที่ไม่คงที่ การใช้ปกติแล้วจะต่อขานี้กับกราวด์

- ขา SP+ , ขา SP- (Speaker Outputs)

เป็นขาเอาท์พุตที่ต่อกับลำโพง ซึ่งในไอซีจะมีสัญญาณความแตกต่างเพื่อใช้ขับออกลำโพง (differential speaker driver) ซึ่งสามารถขับลำโพงที่เอาท์พุตได้ 50 มิลลิวัตต์ เพื่อลำโพงมีอิมพีแดนซ์ 16 โอห์ม โดยที่ขานี้ไม่สามารถนำลำโพงมาต่อขนานกันหลายตัว ซึ่งอาจจะทำให้เกิดความเสียหายกับตัวไอซีได้

- ขา AUX IN (Auxiliary Input)

ขานี้จะเป็นขารับสัญญาณอินพุตจากภายนอก ซึ่งเป็นการมัลติเพล็กซ์สัญญาณผ่านออกไปทางเอาท์พุตของวงจรขยายภายในและขับสู่ลำโพง โดยขั้ตอนการทำงานนี้จะเกิดขึ้นเมื่อขา CE มีสภาวะเป็น "1" วงรอบของการเล่นกลับก็จะสิ้นสุดลง หรือเมื่อสัญญาณที่บันทึกไว้ถูกเล่นกลับจนหมด ถ้ามีการต่อ

แคสเคดหลายตัว ทีขานี้จะถูกต่เข้ากับสัญญาณเล่นกลับที่ออกมาจาขาเอาท์พุตลำโพงของตัวก่อนหน้าที่หรือจากตัวแรก

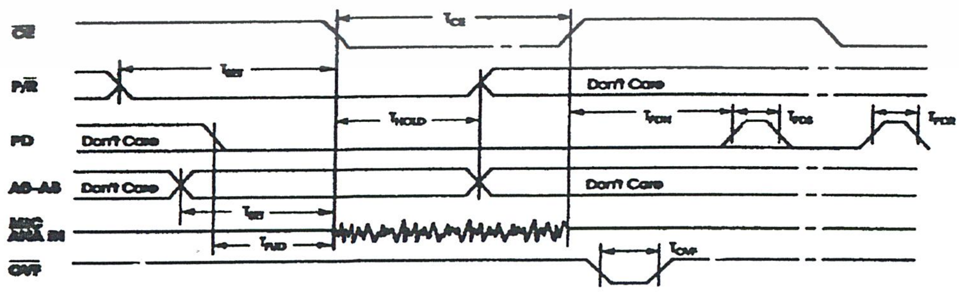
- ขา AX/MX (Address / Mode Inputs)

ขานี้จะมีสองหน้าที่ซึ่งขึ้นอยู่กับของ 2MSB (Most Singnificant Bits) ของขาแอดเดรส A8 และ A9 ถ้าขาหนึ่งขาใดหรือทั้งสองขานี้มีสภาวะเป็น "0" ที่อินพุตทุกตัวจะทำหน้าที่เป็นอินพุตแอดเดรสและถูกใช้เป็นแอดเดรสเริ่มต้นสำหรับการบันทึกและเล่นกลับ โดยในขณะนี้จะเป็นอินพุตเท่านั้น ถ้าขา A8 และ A9 มีสภาวะเป็น "1" ขาแอดเดรสทั้งหมดจะทำหน้าที่เป็นโหมดบิตตามโหมดการทำงานต่าง ๆ ดังแสดงในตารางที่ 2.9 โดยมีโหมดการทำงานทั้งหมด 6 โหมด

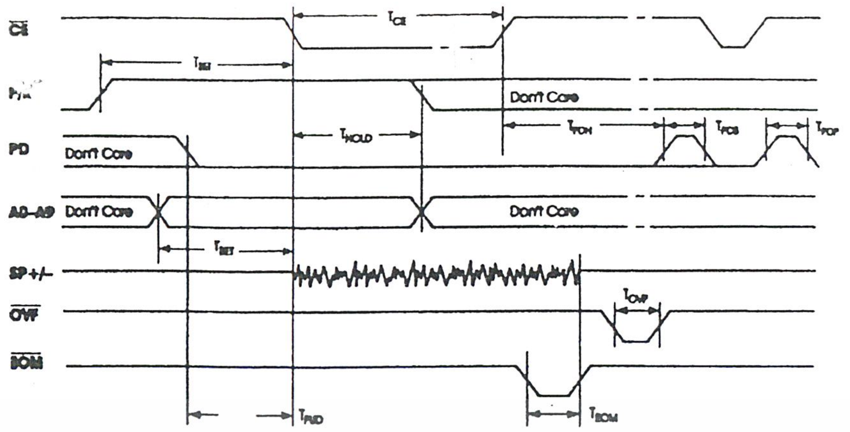
ตางรางที่ 2.9 แสดงโหมดการทำงานของ ISD2590

โหมดควบคุม	หน้าที่	การใช้	ต่อใช้งานร่วมกับโหมด
M0	Message cueing	ข้อความเคลื่อนที่ไปข้างหน้าอย่างรวดเร็ว	M4, M5, M6
M1	Delete EOM markers	ตำแหน่ง EOM marker ที่จุดปลายของข้อความที่แล้ว	M3, M4, M5, M6
M2	Not applicable	สำรองไว้	N/A
M3	Looping	การเล่นกลับแบบต่อเนื่องจากแอดเดรส 0	M1, M5, M6
M4	Consecutive addressing	บันทึก/เล่นติดต่อกันหลายข้อความ	M0, M1, M5
M5	CE level-activated	ยอมให้หยุดข้อความ	M0, M1, M3, M4
M6	Push-button control	อินเตอร์เฟสกับอุปกรณ์อื่น	M0, M1, M3

2.7.3 ไทม์มิ่งไดอะแกรมของสัญญาณต่าง ๆ



รูปที่ 2.15 แสดงไทม์มิ่งไดอะแกรมของการบันทึก



รูปที่ 2.16 แสดงไทม์มิ่งไดอะแกรมของการเล่นกลับ

2.8 ไมโครคอนโทรลเลอร์ตระกูล MCS - 51

2.8.1 คุณสมบัติของ MCS - 51 คุณสมบัติที่สำคัญ ๆ ของชิปไมโครคอนโทรลเลอร์ตระกูล MCS-51มีดังนี้

- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในชิปจำนวน 4 กิโลไบต์ (เบอร์ 8031, 8032 ไม่มีหน่วยความจำส่วนนี้ ส่วนเบอร์ 8052 มีหน่วยความจำส่วนนี้ 8 กิโลไบต์ และสำหรับเบอร์ 83C51FB จะมีหน่วยความจำส่วนนี้รวมทั้งสิ้น 16 กิโลไบต์)
- มีหน่วยความจำสำหรับโปรแกรมและข้อมูล (RAM) อยู่ภายในชิปจำนวน 128 ไบต์ (ในเบอร์ 8031, 8051) หรือ 256 ไบต์ (ในเบอร์ 8031, 8052)
- สามารถใช้หน่วยความจำสำหรับโปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้ อย่างละ 64 กิโลไบต์ แยกจากกัน
- คำสั่งส่วนใหญ่ใช้เวลาทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกกะเฮิรซ์
- มีพอร์ตที่สามารถรับหรือส่งข้อมูลได้ทั้ง 2 ทิศทาง จำนวน 4 พอร์ต ๆ ละ 8 บิต หรือสามารถใช้งานเป็นพอร์ตขนาด 1 บิตแยกจากกัน ทำให้เสมือนมีพอร์ตขนาด 1 บิตใช้งานรวมทั้งสิ้น 32 พอร์ต
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราเร็วในการับและส่งข้อมูล (Baud Rate) ได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที
- จัดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ได้ 2 ระดับ
- มีรีจิสเตอร์สำหรับใช้งานเป็นไทม์เมอร์หรือเคาน์เตอร์ เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิปหรือนับการเปลี่ยนสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อให้สำหรับนับจำนวนพัลส์วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา (ในเบอร์ 8052 จะมี 3 ตัว)
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วน สามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น
- มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
- สามารถประมวลผลแบบบูลีนเพื่อใช้ในงานควบคุมโดยเฉพาะ

- ใช้โปรแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-48 (upwardly compatible) ได้

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ที่จัดได้ว่าเป็นเบอร์พื้นฐานในตระกูลนี้คือ เบอร์ 8051, 8751 และ 8031 ซึ่งมีจำนวนขาภายนอก 40 ขาเหมือนกัน ใช้เวลาในการปฏิบัติคำสั่งแต่ละคำสั่งเท่ากัน (มีไทม์มิ่งไดอะแกรมเหมือนกัน) ใช้แรงดันไฟเท่ากัน สิ่งที่แตกต่างกันระหว่างสามเบอร์นี้ก็คือ ขนาดของหน่วยความจำสำหรับเก็บลงในชิป (on chip program memory) ซึ่งมีไว้เพื่อตอบสนองความต้องการที่ไม่เหมือนกัน ดังแสดงในตารางที่ 2.10

ตารางที่ 2.10 แสดงความแตกต่างของสมาชิกตระกูล MCS-51

Device	ROMiess Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/ Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (PCA)	Global Serial Channel (GSC)	DMA Chan-nels	A/C Chan-nels	Interrupt Sources/ Vectors	Power Down And Idia Modes
8051	8031	-	4K	128	4	2		✓					6/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					8/6	
80C51BH	80C31BH	87C51	4K	128	4	2							6/5	✓
80C52	80C32	-	8K	256	4	3		✓					8/6	✓
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
83C51FB	80C51FB	87C51FB	16K	256	4	3	✓	✓					14/7	✓
83C152JA	80C152JA	-	8K	256	5	2		✓		✓	2		19/11	✓
-	80C152JB	-	-	256	7	2		✓		✓	2		19/11	✓
83C152JC	80C152JC	-	8K	256	5	2		✓		✓	2		19/11	✓
-	80C152JD	-	-	256	7	2		✓		✓	2		19/11	✓
83C452	80C452	87C452P	8K	256	5	2		✓					9/8	✓

คุณสมบัติของไมโครคอนโทรลเลอร์ในตระกูล MCS-51

: เบอร์ 8751 มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเป็น EPROM ขนาด 4 กิโลไบต์ ซึ่งจะสามารถทำการลบข้อมูลเดิมแล้วเขียนซ้ำลงไปใหม่ได้ ทั้งนี้เพื่อความสะดวกในการแก้ไขหรือปรับปรุงโปรแกรม

: เบอร์ 8051 หลังจากทดสอบโปรแกรมจนไม่พบข้อผิดพลาดแล้ว จะเป็นช่วงของการผลิตจริง ซึ่งต้องคำนึงถึงต้นทุนมาเป็นอันดับแรก ในการผลิตจริงจะใช้ไมโครคอนโทรลเลอร์เบอร์ 8051 ซึ่งมีหน่วยความจำภายในเป็น ROM ขนาด 4 กิโลไบต์แทน เพราะราคาต่ำกว่ามาก แต่มีข้อจำกัดที่ไม่สามารถทำการแก้ไขโปรแกรมที่บรรจุไปแล้วได้ ไม่ว่าจะด้วยวิธีใดก็ตาม

: เบอร์ 8031 เบอร์นี้จะไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป แต่สามารถใช้หน่วยความจำเพื่อเก็บโปรแกรมภายนอกชิปได้มากถึง 64 กิโลไบต์ ซึ่งอาจจะใช้เป็น ROM , PROM หรือ EPROM ตามความต้องการของผู้ใช้

: เบอร์ 8751 และ 8051 จะใช้โปรแกรมจากหน่วยความจำภายนอกได้เอง เมื่อโปรแกรมมีความยาวเกิน 4 กิโลไบต์ หรืออาจบังคับให้โมโครคอนโทรลเลอรืทั้งสองเบอร์ใช้โปรแกรมจากหน่วยความจำภายนอกเพียงอย่างเดียวด้วยการต่อขา 31 (EA) ลงกราวด์ ทำให้มีคุณสมบัติเหมือนกับเบอร์ 8031 ที่ไม่มีหน่วยความจำสำหรับโปรแกรมภายในชิป

2.8.2 โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51

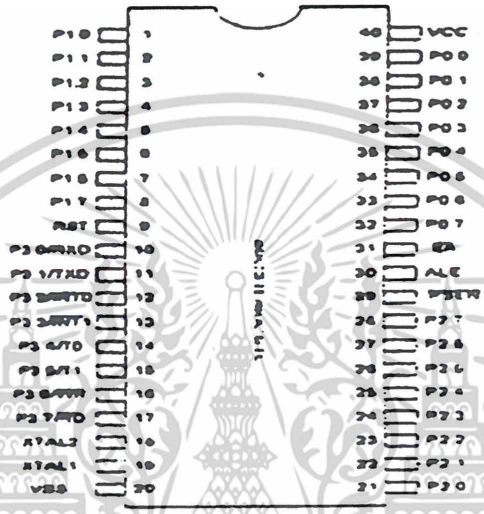
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีสมาชิกในตระกูลหลายเบอร์ด้วยกัน แต่ละเบอร์จะมีคุณสมบัติพิเศษบางอย่างแตกต่างกัน เช่น มีหน่วยความจำภายในสำหรับเก็บโปรแกรมและข้อมูลภายในชิปเพิ่มขึ้น มีวงจรเปลี่ยนค่าสัญญาณอนาล็อกเป็นดิจิตอลในตัว สามารถรับสัญญาณอินเตอร์รัปต์ได้หลายชนิด ทำกระบวนการ DMA (Direct Memory Access) ได้ในตัว มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์เพิ่มขึ้น คุณสมบัติพิเศษที่แตกต่างกันของไมโครคอนโทรลเลอร์แต่ละเบอรืในตระกูลนี้ดังแสดงในตารางที่ผ่านมา

ไมโครคอนโทรลเลอร์เบอร์ที่นับได้ว่าเป็นเบอร์พื้นฐานสำหรับตระกูล MCS-51 นี้ ได้แก่ เบอร์ 8051, 8031,8751, โดยเบอร์ 8051 จัดเป็นสมาชิกตัวแรกในตระกูล ซึ่งมีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป เป็นROM ขนาด 4 กิโลไบต์ และหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายใน MCS-51 (RAM) เองจำนวน 128 ไบต์ มีพอร์ตขนาด 8 บิต 4 พอร์ต มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์ขนาด 16 บิตรวม 2 ตัว รับสัญญาณอินเตอร์รัปต์จากภายนอกได้ 2 ชนิด สามารถรับและส่งข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารข้อมูลแบบอนุกรม มีวงจรออสซิลเลเตอร์เพื่อสร้างสัญญาณนาฬิกาควบคุมการทำงานในตัวเอง ส่วนเบอร์ 8751 จะมีคุณสมบัติเหมือน เบอร์ 8051 ทุกอย่าง ต่างกันเพียงชนิดของหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป คือ เบอร์ 8751 จะเป็น EPROM แทนที่จะเป็น ROM ส่วนเบอร์ 8031 จะเหมือนกับเบอร์ 8051 ต่างกันเพียงในเบอร์ 8031 ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเท่านั้น

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์ใช้แรงดันไฟเพียง 5 โวลต์ในการทำงานส่วนกระแสไฟฟ้าที่ใช้จะแตกต่างกันไปตามชนิดของเทคโนโลยีที่ใช้ในการผลิต เบอร์ของไมโคร

คอนโทรลเลอร์ตระกูลนี้ที่มีตัวอักษร C อยู่ตรงกลางเบอร์ เช่น 80C31, 80C51 จะเป็นเบอร์ของชิปที่ผลิตโดยอาศัยเทคโนโลยี CHMOS ซึ่งใช้พลังงานในการทำงานน้อยกว่าและสามารถควบคุมการใช้พลังงานของตัวชิปได้จากโปรแกรมเพื่อการประหยัดพลังงานในระบบ

MCS-51 เป็นตระกูลของไมโครคอนโทรลเลอร์ที่ถูกพัฒนาขึ้นมาจากตระกูล MCS-48 ดังนั้นจึงมีความสามารถเหนือกว่าหลายอย่าง ข้อดีของ MCS-51 คือสามารถใช้ความถี่ได้ถึง 12 เมกกะเฮิรตซ์ หรือสำหรับบางเบอร์ในตระกูลสามารถใช้ได้ถึง 16 เมกกะเฮิรตซ์ ทำให้ช่วงเวลาในการทำงานแต่ละคำสั่งที่ใช้เวลามากที่สุดจะใช้เวลาเพียง 4 ไมโครวินาทีเท่านั้น



รูปที่ 2.17 การจัดวางขาของ 8051

2.8.3 ตำแหน่งขาของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์ จะมีตำแหน่งขาพื้นฐานที่เหมือนกันดังแสดงใน

รูปที่ 2.17

หน้าที่การใช้งานแต่ละขาของไมโครคอนโทรลเลอร์ในตระกูล MCS-51 มีดังนี้

- ขา VSS (ขา 20) สำหรับต่อลงกราวด์
- ขา VCC (ขา 40) สำหรับต่อแหล่งจ่ายแรงดันกระแสตรงขนาด 5 โวลต์ (DC. 5 volt)
- ขาพอร์ต 0 (ขา 32-39) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 0 ขนาด 8 บิต (PO.0-PO.7) แบบโอเพ่นเดรน ไบไดเร็คชั่นแนล (Open Drain Bidirectional) พอร์ตนี้สามารถใช้งานเป็นอินพุต-เอาท์พุตพอร์ตทั่วไปได้ โดยหากใช้เป็นอินพุตพอร์ตต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อบังคับให้เขาอยู่ในสถานะถูกปล่อยลอย (มีสถานะ high impedance) นอกจากใช้งานเป็นอินพุต-เอาท์พุตพอร์ตแล้ว พอร์ต 0 ยังใช้ในการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายนอกชิปได้ด้วย โดยส่งค่าแอดเดรสบัสไบต์ต่ำ (A0-A7) และมัลติเพล็กซ์กับการรับส่งข้อมูล (D0-D7) จากหน่วยความจำภายในระหว่างการเขียนหรืออ่านข้อมูลโดยมีวงจรพูลอัพภายใน

- ขาพอร์ต 1 (ขา 1-8) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต (P1.0-P1.7) สามารถใช้งานเป็นอินพุตหรือเอาท์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อให้มีสถานะถูกปล่อยลอย โดยมีวงจรพูลอัพภายใน

- ขาพอร์ต 2 (ขา 21-28) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 2 (P2.0-P2.7) ขนาด 8 บิต แบบโอเพ่นเดรน ไบไดเร็คชั่นแนล พอร์ตนี้สามารถใช้งานเป็นอินพุต-เอาท์พุตพอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อบังคับให้เขาอยู่ในสถานะถูกปล่อยลอย นอกจากจะใช้งานเป็นอินพุต-เอาท์พุตพอร์ตทั่วไปแล้ว พอร์ต 2 ยังใช้ในการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายนอกด้วย โดยใช้สำหรับส่งค่าแอดเดรสบัสไบต์สูง (A8-A15) และมีวงจรพูลอัพภายใน

- ขาพอร์ต 3 (ขา 10-17) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 3 (P3.0-P3.7) สามารถใช้งานเป็นอินพุต-เอาท์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้เพื่อให้สถานะถูกปล่อยลอย โดยใช้วงจรพูลอัพภายใน นอกจากนี้ยังใช้งานในหน้าที่พิเศษต่าง ๆ อีกหลายอย่างดังนี้

: ขา P3.0 ใช้รับข้อมูลจากภายนอกแบบอนุกรม

: ขา P3.1 ใช้ส่งข้อมูลออกไปภายนอกแบบอนุกรม

: ขา P3.2 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 0

: ขา P3.3 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 1

: ขา P3.4 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 0

: ขา P3.5 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 1

: ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป

: ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำ สำหรับเก็บข้อมูลภายนอกชิป

การใช้งานพอร์ต 3 ในหน้าที่พิเศษดังกล่าวนี้จะต้องโหลดค่า 1 ไปยังแต่ละบิตที่ต้องการใช้ก่อนทุกครั้ง

- ขา RST (ขา9) ใช้สำหรับการรีเซตวงจรทุกอย่างภายในชิปเพื่อเริ่มต้นการทำงานใหม่ การรีเซตใช้เมื่อเริ่ม จ่ายพลังงานหรือเมื่อโปรแกรมเกิดทำงานผิดพลาด เมื่อต้องการรีเซตชิป MCS-51 ขานี้ต้องมีสถานะ 1 เป็นเวลาอย่างน้อย 2 แมชชีนไซเกิลระหว่างที่ออสซิลเลเตอร์ยังทำงานอยู่ โดยต้องต่อตัวต้านทานค่า 8.2 กิโลโอห์ม เพื่อทำหน้าที่พูลดาวน์ (รักษาค่าแรงดันไฟฟ้าให้มีสถานะเป็นกราวด์) และเพื่อให้ตัวชิปรีเซตเองเมื่อเริ่มจ่ายพลังงานให้ต่อตัวเก็บประจุขนาด 10 ไมโครฟารัดคร่อมระหว่างขา RST กับ VCC

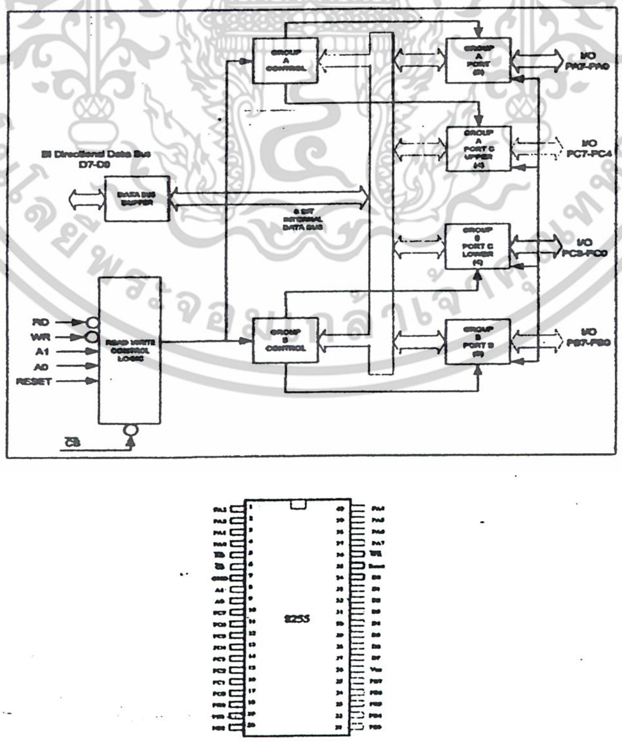
- ขา ALE/PROG (ขา 30) เป็นขาสำหรับใช้ส่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตช์ค่าแอดเดรสไบต์ต่ำ (address latch enable) จากพอร์ต 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอก ปกติเมื่อไม่มีการติดต่อหน่วยความจำภายนอก ขานี้จะส่งสัญญาณพัลส์

ออกมาด้วยความถี่ 1/8 ของความถี่ออสซิลเลเตอร์ที่ใช้ตลอดเวลา ดังนั้นเราสามารถใช้ความถี่ที่ได้จากขานี้ไปใช้งานอย่างอื่นได้ แต่ความถี่ที่ขานี้จะลดลงครึ่งหนึ่งในระหว่างติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลที่อยู่ภายนอกชิป นอกจากนี้ขา ALE ยังใช้สำหรับควบคุมการเขียนโปรแกรมลงไปใน EPROM สำหรับ MCS-51 เบอร์ที่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเป็น EPROM

- ขา EA/V_{PP} (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจากโปรแกรมที่อยู่ภายในหรือภายนอกชิป โดยหากขานี้มีสถานะเป็น 0 หมายถึงให้ใช้โปรแกรมจากหน่วยความจำที่เก็บโปรแกรมภายนอกหากขานี้มีสถานะเป็น 1 หมายถึงบังคับให้ MCS-51 ใช้โปรแกรมจากหน่วยความจำ สำหรับเก็บโปรแกรมภายในชิป และสำหรับ MCS-51 ที่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป สามารถเลือกให้ทำงานได้ทั้งจากโปรแกรมที่เก็บในหน่วยความจำภายในชิปหรือจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิปด้วยการต่อขา EA กับไฟเลี้ยงหรือกราวด์ตามลำดับ ส่วนใน MCS-51 ที่ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปให้ต่อขานี้ลงกราวด์เสมอ
- ขา XTAL 1 (ขา 19) ใช้ต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์
- ขา XTAL 2 (ขา 18) ใช้ต่อคริสตอลภายนอก โดยเป็นเอาท์พุตออกจากวงจรออสซิลเลเตอร์

2.9 การเชื่อมโยง 8255 กับ MCS-51

8255A PROGRAMMABLE PERIPHERAL INTERACE เป็นชิพขนาด 40 ขา มีอยู่ 3 พอร์ต คือ A,B,C. เป็นพอร์ตขนาด 8 บิตที่สามารถโปรแกรมให้เป็นอินพุตหรือเอาต์พุตก็ได้ โดยพอร์ต C ยังแบ่งเป็น 4 บิตล่าง และ 4 บิตบน โดยมีโครงสร้างตามรูปที่ 2.18



รูปที่ 2.18 การตัดวางขาและโครงสร้างของ 8255

ตารางที่ 2.11 สรุปโหมดต่าง ๆ ของ 8255

Mode / Port	MODE1		MODE2		MODE3
	IN	OUT	IN	OUT	GROUP A ONLY
PA0	IN	OUT	IN	OUT	⇔
PA1	IN	OUT	IN	OUT	⇔
PA2	IN	OUT	IN	OUT	⇔
PA3	IN	OUT	IN	OUT	⇔
PA4	IN	OUT	IN	OUT	⇔
PA5	IN	OUT	IN	OUT	⇔
PA6	IN	OUT	IN	OUT	⇔
PA7	IN	OUT	IN	OUT	⇔
PB0	IN	OUT	IN	OUT	----
PB1	IN	OUT	IN	OUT	----
PB2	IN	OUT	IN	OUT	----
PB3	IN	OUT	IN	OUT	----
PB4	IN	OUT	IN	OUT	----
PB5	IN	OUT	IN	OUT	----
PB6	IN	OUT	IN	OUT	----
PB7	IN	OUT	IN	OUT	----
PC0	IN	OUT	$INTR_B$	$INTR_B$	I/O
PC1	IN	OUT	IBF_B	OBF_B	I/O
PC2	IN	OUT	STB_B	ACK_B	I/O
PC3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$
PC4	IN	OUT	STB_A	I/O	STB_A
PC5	IN	OUT	IBF_A	I/O	IBF_A
PC6	IN	OUT	I/O	ACK_A	ACK_A
PC7	IN	OUT	I/O	OBF_A	OBF_A

1. โหมดการทำงาน

การทำงานมีอยู่ด้วยกัน 3 โหมด ดังตารางที่ 2.11

- โหมด 0 มีการทำงานแบบ BASIC I/O ไม่มีการตรวจสอบสัญญาณ (Handshake)
- โหมด 1 โหมดนี้ใช้พอร์ต A,B ในการรับหรือส่งข้อมูล และใช้พอร์ต C ในการตรวจสอบสัญญาณ (Handshake)

- โหมด 2 โหมดนี้ใช้พอร์ต A ในการรับส่งข้อมูล 2 ทิศทาง และพอร์ต B ในการรับหรือส่งข้อมูลและใช้พอร์ต C บิต 0,1,2 ในการรับส่งข้อมูลบิตและบิต 4,5,6 ในการการตรวจสอบสัญญาณ(Handshake)

2. สัญญาณต่างๆ ของ 8255

D0-D7 บัสข้อมูลเชื่อมโยงกับ CPU

A0-A7 ใช้เลือกพอร์ต A B C และพอร์ตควบคุม

RESET เมื่อขานี้ได้สัญญาณกระตุ้นลอจิก"1" จะทำให้ 8255 ถูกรีเซ็ตมีผลทำให้ทุกพอร์ตเป็นอินพุตทันที

PA0-PA7 เป็นพอร์ตขนาน 8 บิต

PB0-PB7 เป็นพอร์ตขนาน 8 บิต

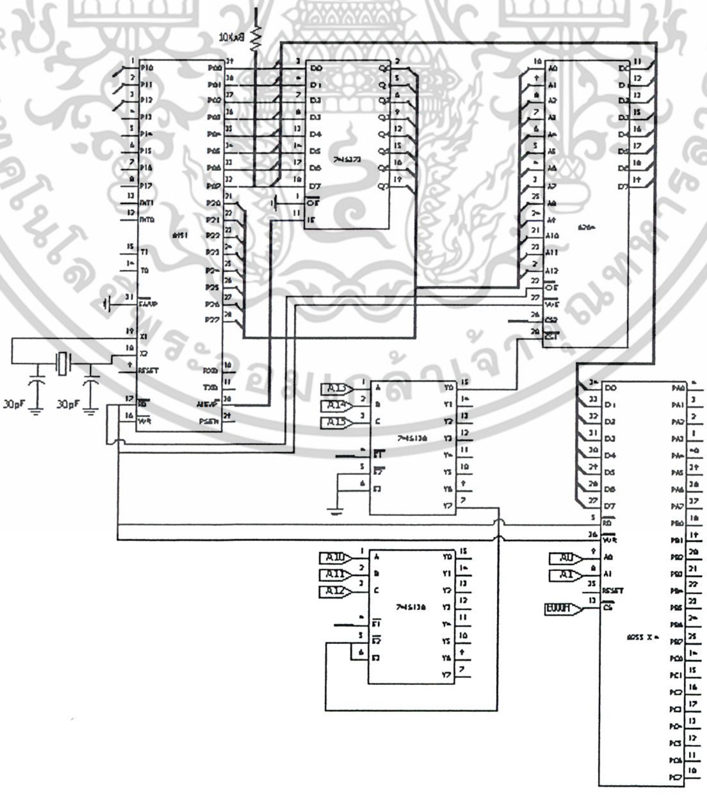
PC0-PC7 เป็นพอร์ตขนาน 8 บิต

RD ในการอ่านข้อมูลที่พอร์ตของ 8255 ต้องทำให้ขานี้เป็นลอจิก "0" พร้อมกับ CS

WR ในการเขียนข้อมูลหรือโปรแกรมลงบนที่พอร์ตของ 8255 ต้อทำให้ขานี้เป็นลอจิก "0" พร้อมกับ CS

CS เป็นขาเลือกชิพ 8255 ได้ ขานี้มักต่อกับ I/O DECODER

3. การเชื่อมต่อ 8255 เข้ากับ CPU ดังแสดงการต่อในรูปที่ 2.19



รูปที่ 2.19 การเชื่อมโยง 8255 เข้ากับ CPU

2.10 การรับส่งข้อมูลแบบอนุกรม

2.3.1 พื้นฐานการรับส่งข้อมูล

การรับส่งข้อมูลในระบบคอมพิวเตอร์โดยทั่วไปจะหมายถึง การรับส่งข้อมูลเป็นจำนวนไบต์ ๆให้กับอุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์ ซึ่งอาจจะแบ่งการรับส่งได้เป็น 2 แบบคือ

1) การรับส่งข้อมูลแบบขนาน (Parallel)

2) การรับส่งข้อมูลแบบอนุกรม (Serial)

การรับส่งข้อมูลแบบขนานจะเป็นการรับส่งข้อมูล จำนวน 1 ไบต์ ออกไปทางพอร์ตในเวลาเดียวกันในระบบ

คอมพิวเตอร์ 1 ไบต์จะมีจำนวน 8 บิต คือ D0-D7 ถ้ามีการส่งข้อมูลแบบขนานจะใช้สาย สัญญาณอย่างน้อย 9 เส้น คือสายข้อมูล 8 เส้น และสายกราวด์ 1 เส้น

การรับส่งข้อใลแบบอนุกรม คือ การรับส่งข้อมูลทีละบิต จนครบ 1 ไบต์ ถ้าต้องการส่งข้อมูล 1 ไบต์ คือ D0-D7 อาจจะส่งบิต D0 ออกไปก่นแล้วตามด้วย D1 ไปเรื่อยๆ จนถึง D7 การส่งข้อมูลทั้งสองแบบมีข้อดีข้อเสียแตกต่างกันคือการส่งข้อมูลแบบขนาน สามารถส่งข้อมูลได้เร็วคือส่งทีเดียวจะได้ข้อมูลครบ 1 ไบต์ แต่ถ้าต้องการส่งเป็นระยะทางไกลๆ จะสิ้นเปลืองสายสัญญาณมาก ถ้าเป็นการส่งแบบอนุกรมเมื่อต้องการส่งข้อเป็นระยะไกลๆ จะช่วยประหยัดสายสัญญาณเนื่องจากจะใช้สายอย่างน้อยเพียง 2 เส้นคือสายสัญญาณกับสายกราวด์ แต่การรับส่งข้อมูลจะใช้เวลานานเป็นการส่งทีละบิต

- การรับส่งข้อมูลแบบซิงโครนัส (Synchronous Input / Output)

การรับส่งข้อมูลแบบนี้ไม่ว่าจะเป็นการส่งแบบอนุกรมหรือการส่งแบบขนาน ข้อมูลแต่ละไบต์ที่ถูกส่งออกไปจะมีช่วงเวลาห่างกันแน่นอน

- การรับส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Inpupt / Output)

การรับส่งข้อมูลแบบนี้ที่ส่งออกไปจะไม่มีเวลาที่แน่นอน ซึ่งจะขึ้นอยู่กับความพร้อมของผู้ส่งและผู้รับ โดยจะสายสัญญาณตรวจสอบความพร้อมของระบบทั้งสองว่าพร้อมที่จะติดต่อกันหรือไม่ โดยสัญญาณที่เพิ่มขึ้นมาจากระบบแบบ ซิงโครนัสเรียกว่า สายสเตตัส (status Line)

2.3.2 รูปแบบของการรับส่งข้อบุลแบบอนุกรม

เมื่อไมโครคอมพิวเตอร์ต้องการจะส่งข้อมูลแบบอนุกรม ตัวไมโครคอมพิวเตอร์จะสงข้อมูลออกไปทางพอร์ตซึ่งเป็นพอร์ตแบบขนานก่อน จากนั้นจะมีอุปกรณ์มาต่อที่พอร์ต เพื่อแปลงข้อมูลแบบขนานให้เป็นแบบอนุกรมอีกทีหนึ่ง(Parallel-to-serial Conversion) ตัวแปลงข้อมูลนี้อาจพิจารณาง่ายๆ ว่าเป็น Shift Register เมื่อข้อมูลที่จะส่งอยู่ใน Shift Register แล้วตัดสัญญาณนาพิกาจะเป็นตัวกระตุ้นให้ส่งข้อมูลบิตต่ำออกไปในเวลาแรก จากนั้นจะส่งบิตต่อไปตามออกมา

สำหรับตัวรับข้อมูลแบบอนุกรมเมื่อตัวรับรับข้อมูลจะเป็นการรับเข้ามาใน Shift Register แล้วส่งข้อมูลให้ไมโครคอมพิวเตอร์แบบขนานอีกทีหนึ่ง (Serial-to-parallel) ระบบคอมพิวเตอร์ในปัจจุบันจะมีตัวแปลง Paraell-to-serial และ Serial-to-paraell อยู่ในชิพไอซี เรียกว่า Universal Asynchronous Receiver Transmitter (UART)การส่งข้อมูลแบบอนุกรมนั้นจะต้องมีการเพิ่มเติมข้อมูลบางอย่างเข้าไปเพื่อให้การรับส่งข้อมูลสามารถ โดยมีการเติมค่าบิตต่างๆลงไป

Stop	P	D_7	D_6	D_5	D_4	D_3	D_2	D_0	Start

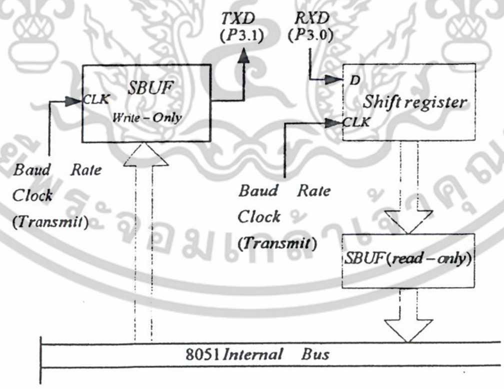
รูปที่ 2.20 บิตต่าง ของข้อมูลที่ส่งออกแบบอนุกรม

ถ้ามีการส่งข้อมูลแบบ 8 บิต จะต้องส่งบิตแรกออกไปก่อน เรียกว่า บิตเริ่มต้น (Start Bit) ถ้ามีการส่งข้อมูลหลายๆ ไบต์ออกมา บิตนี้จะเป็นตัวบอกว่า มีข้อมูลใหม่มาแล้ว โดยทั่วไปบิตเริ่มต้นมักมีระดับลอจิก “0” ต่อจากบิตเริ่มต้นจะเป็นข้อมูลบิต D0-D7 จากนั้นจะตามด้วยบิตตรวจสอบความถูกต้อง (พาริตี้ Bit) ถ้าข้อมูล 8 บิตที่ส่งออกมา จำนวนของบิตที่มีค่าเป็น “1” เป็นจำนวนคู่ บิตนี้จะมีค่าเป็น 0 แต่ถ้าจำนวนของบิตที่มีค่าเป็น “1” เป็นจำนวนคี่ บิตนี้จะมีค่าเป็น “1” จากนั้น ข้อมูลที่จะส่งออกไปจะตามด้วยบิตสิ้นสุด (Stop Bit) เพื่อเป็นการบอกว่าข้อมูลที่ส่งมา 8 บิตนั้นหมดแล้ว ตัวบิต Stop อาจมีจำนวนมากกว่า 1 บิต ก็ได้ เช่น 1 1-2 บิต , 2 บิต

2.3.3 MCS-51 กับการทำรับส่งข้อมูลแบบอนุกรม

พอร์ตอนุกรมของ MCS-51 จะใช้ขา TXD และ RXD ในการรับส่งข้อมูล โดยขาทั้งสองจะอยู่ในพอร์ต 3 คือ P3.1 หรือขา 11 เป็น TXD และ P3.0 หรือขา10 เป็น RXD พอร์ตอนุกรมของ MCS-51 สามารถทำงานแบบ Full-Duplex ได้ คือสามารถส่งและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูลจะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

รีจีสเตอร์ที่สำคัญในการรับส่งข้อมูลคือ SBUF และ SCON ซึ่งเป็นรีจีสเตอร์ที่อยู่ใน Special Function Register โดยรีจีสเตคร์ Serial Port Buffer (SBUF) จะอยู่ในตำแหน่ง 99H ถ้าเขียนข้อมูลไปที่ตำแหน่งนี้ จะเป็นการส่งข้อมูลออกทางพอร์ตอนุกรม และถ้าอ่านข้อมูลจากตำแหน่งนี้จะเป็นการรับข้อมูลจากพอร์ตอนุกรม โดยใน SBUFจะประกอบด้วยบัฟเฟอร์ 2 ตัว สำหรับส่งและรับข้อมูล ดังรูปที่2.21



รูปที่ 2.21 การรับส่งข้อมูลระหว่างรีจีสเตอร์กับบัสภายใน

สำหรับ Serial Port Control Register (SCON) ซึ่งอยู่ที่ตำแหน่ง 98 H จะเป็นรีจีสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ รีจีสเตอร์นี้จะทำหน้าที่ควบคุมแล้บอกสถานะต่างๆ ของการรับส่งข้อมูลแบบอนุกรม

สำหรับความเร็วของการส่งข้อมูล (บอดเรต) สามารถหาได้จากการหารศัญญาณนาฬิกาที่ใช้กับ MCS-51

Serinal Port Control Register

MCS-51 มีโหมดการทำงานของพอร์ตอนุกรมหลายโหมด ซึ่งสามารถโปรแรมโหมดการทำงานได้โดยเขียนข้อมูลไปยังรีจีสเตอร์ SCON ความหมายของแต่ละบิตแสดงดังตารางที่ 2.12

ตารางที่ 2.12 บิตต่างๆ ของรีจีสเตอร์ SCON

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON .7	SMO	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON .6	SMO	9EH	บิตเลือกโหมดการทำงานบิต 1
SCON .5	SMO	9DH	บิตเลือกโหมดการทำงานบิต 2
SCON .4	REN	9CH	บิตแฟลกกำหนดยอมให้มีการรับข้อมูล
SCON .3	TB8	9BH	ค่าของบิต 9 สำหรับการส่งข้อมูลในโหมด 2 และ 3 สามารถเซ็ตและเครียร์ได้โดยซอร์แวร์
SCON .2	TB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา
SCON .1	TI	99H	บิตแฟลกแสดงการอินเทอร์รัพท์ภายหลังการส่งข้อมูลออกไปโดยจะ set เมื่อส่งข้อมูลออกไปหมดแล้วและสามารถ Clear ได้ด้วย Software
SCON .0	RI	98H	แฟลกแสดงการอินเทอร์รัพท์ภายหลังรับข้อมูลเข้าสามารถ Clear ได้ด้วย Software

ตารางที่ 2.13 แสดงโหมดต่างๆ ของการรับส่งแบบอนุกรม

SMo	SMi	MODE	ความหมาย	บอดเรต
0	0	0	Shift Register	เปลี่ยนแปลงไม่ได้ (Oscillator Frequency +12)
0	1	1	8 – bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจากไทเมอร์
1	0	2	9 – bit UART	เปลี่ยนแปลงไม่ได้ (Oscillator Frequny + 12 หรือ + 64)
1	1	3	9 – bit UART	สามารถเปลี่ยนแปลงได้โยกำหนดจากไทเมอร์

ก่อนที่จะใช้พอร์ตอนุกรมจะต้องโปรแกรมให้กับ SCON เสียก่อนเพื่อกำหนดการทำงานและลักษณะต่างๆ

ในการส่งข้อมูลทุกโหมดสามารถทำได้โดยเขียนข้อมูลไปยัง SBUF เมื่อข้อมูลถูกส่งไปแล้วบิต TI จะถูกเซ็ตเป็น " 1 " ในการส่งข้อมูลจะต้องคอยตรวจสอบบิต TI เพราะว่าถ้า TI ยังไม่เป็น "1 " แสดงว่าข้อมูลยังส่งไปไม่หมด ถ้าหากมีการเขียนข้อมูลไบต์ต่อไปยัง SBUF จะทำให้เกิดข้อผิดพลาดขึ้นสำหรับในการรับข้อมูลบิต REN จะต้องเซตให้เป็น "1 " ยกเว้นโหมด 0 เพื่ออนุญาตให้รับข้อมูลได้ เมื่อข้อมูลรับเข้ามาเรียบร้อยแล้วบิต RI จะถูกเซ็ทเป็น "1 "

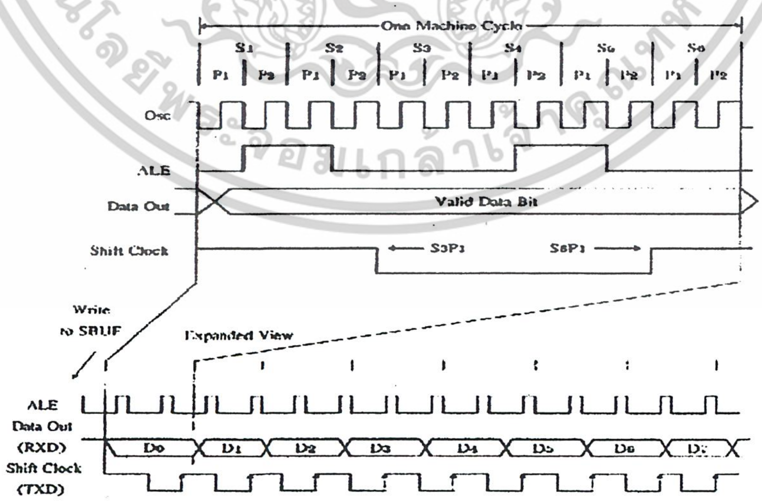
2.11 โหมดการทำงาน (Mode Of Operation)

ใน MCS-51 การสื่อสารทางพอร์ตจะมีอยู่ 4 ประเภท หรือ 4 โหมด ซึ่งจะกำหนดได้ที่บิต SMO และ SMI ใน SCON โดยจะมี 3 โหมด เป็นการสื่อสารแบบ Asynchronous โดยลักษณะของข้อมูลที่ส่งจะมีบิตเริ่มต้น Start Bit)และบิตจบ (Stop Bit) คล้ายกับการสื่อสราแบบ RS – 232 ในระบบคอมพิวเตอร์ อีกโหมดหนึ่งจะเป็นการใช้พอร์ตอนุกรมในลักษณะชิพจีสเตอร์

- Mode 0

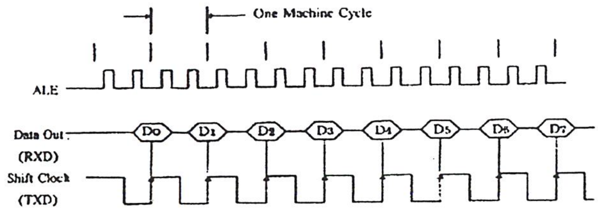
การทำงานในโหมดนี้จะใช้ขา REX ในการรับส่งข้อมูลโดยต่อกับ Shift Register ภายนอก ส่วนขา TXD จะเป็น Out Shift Clock เพื่อกระตุ้นรีจิสเตอร์ภายนอกให้เลื่อนบิตถ้ามีการส่งข้อมูลหรือรับข้อมูล 8 บิต จะเริ่มที่บิตต่ำสุดก่อน โดยมีค่าบอดเรต เท่า ½ ของความถี่ที่ใช้บนชิพ

ในการส่งข้อมูลจะทำโดย เขียนข้อมูลไปที่รีจิสเตอร์ SBUF ข้อมูลจะถูกส่งออกมาทางขา RXD (P3.0) โดยจะสอดคร้องกับสัญญาณออกมาทางขา TXD ซึ่งสัญญาณของขา TXD จะถูกส่งออกมาทุกๆ Machine Cycle โดยจะเป็นลอจิก "0" ใน S3P1 และจะกลับเป็นลอจิก " 1 " ใน S6P1 ซึ่งแสดงได้ดังรูปที่ 2.22



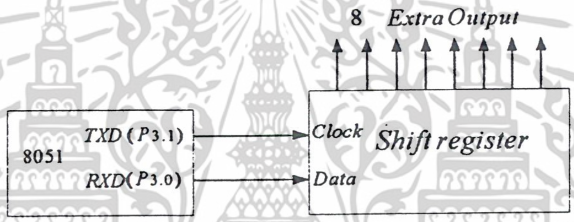
รูปที่ 2.22 แสดงไดอะแกรมเวลาการส่งข้อมูล

สำหรับการรับข้อมูลจะรับได้เมื่อเซตขา Receive Enable Bit (REN) เป็น " 1 " และคียรขา Receive Interrpt Bit (RI) เป็น " 0 " ข้อมูลจะเข้าสู่ MSC-51 เมื่อ clock Shift ถูกส่งออกไปทาง TXD ที่ขอบขาขึ้นเอง Clock Shift บิตต่ำจะถูกส่งเข้ามาก่อนดังรูปที่ 2.23



รูปที่2.23 แสดงไดอะแกรมเวลาการรับข้อมูล

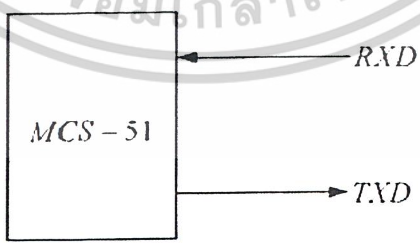
ในการประยุกต์ใช้งานโหมดนี้จะต้องมีไอซีชิพรีจิสเตอร์มาต่อภายนอก เช่น ถ้าหากต้องการส่งข้อมูลออกทางพอร์ตอนุกรม อาจต่อวงจรได้ดังรูป 2.24 โดยใช้ไอซี Serial – to – Parallel Shift Register โดยข้อมูลส่งข้อมูลออกมาทาง RXD และใช้ TXD เป็น Clock



รูปที่ 2.24 การส่งข้อมูลออกโดยใช้ชิพจีสเตอร์ช่วย

- Mode 1

ในโหมดนี้จะเป็นการรับส่งข้อมูลแบบ 10 บิตซึ่งประกอบด้วยบิตเริ่มต้น (เป็น " 0 ") ข้อมูล 8 บิตและบิตจบ (เป็น " 1 ") นอกจากนี้ยังสามารถกำหนดค่าบอดเรตได้โดยค่าบอดเรตนี้จะแปรตามตัวจับเวลาตัวที่ 1 ใน โหมดนี้ จะส่งข้อมูลออกทาง TXD และรับข้อมูลเข้าทาง RXD ถ้าเป็นการรับข้อมูลเข้าตัว Stop Bit จะเข้ามายังบิต RB8 ใน SCON



รูปที่ 2.25 การรับส่งข้อมูลในโหมด 1

ค่าบอดเรตที่ใช้ในการรับว่งข้อมูลจะกำหนดโดยไทเมอร์ 1 หลังจากโปรแกรมไปในไทเมอร์ 1 แล้วสามารถเลือกค่าบอดเรตได้อีกสองค่าคือ ค่าจากไทเมอร์ 1 โอเวอร์โฟว์ หาร 32 กับค่าจาก ไทเมอร์ 1 โอเวอร์โฟว์ หาร 16

การส่งข้อมูลทำได้โดยการเขียนข้อมูล 8 บิตไปที่ SBUF โดยบิตที่ 9 (Stop Bit) ให้เขียนลงใน TB8 ใน SCON จากนั้นข้อมูลจะถูกส่งออกทางขา TXD โดยส่ง Start Bit ออกมาก่อน ตามด้วยข้อมูล 8 บิต และจบด้วย Stop Bit เมื่อข้อมูลถูกส่งออกไปหมดแล้วบิต Interrupt Flag จะเป็น “1” ดังนั้นในการเขียนข้อมูลใหม่ลงไปจะต้องตรวจสอบบิตนี้

ในการรับข้อมูล จะเริ่มจากเริ่มเมื่อมีการเปลี่ยนแปลงจิกจาก 1 เป็น 0 ทางขา RXD หมายความว่าเริ่มต้นรับบิตเริ่มตัน จากนั้นข้อมูลอีก 8 บิต จะถูกเก็บลงใน SBUF และ Stop Bit จะถูกเก็บในบิต RB8 ของรีจีสเตอร์ SCON เมื่อข้อมูลเข้ามาครบแล้วบิต Interrupt Flag(RI) RI จะถูกเซต ดังนั้นในกาอ่านข้อมูลจะอ่านได้เมื่อ RI ถูกเซตแล้ว เมื่ออ่านข้อมูลไปแล้วจะต้องต้องเคลียร์บิตนี้

- Mode 2

การทำงานในโหมดนี้ไม่สามารถกำหนดค่าบอดเรตได้ซึ่งค่าบอดเรต จะมีสองค่าคือ 1/64 และ 1/32 ของสัญญาณนาฬิกาบนชิฟ การรับส่งข้อมูลจะเป็นชุดข้อมูล 9 บิต บิตเริ่มต้น บิตหยุด รวมเป็น 11 บิต โดยข้อมูล 9 บิตจะเป็นจำนวนข้อมูล 8 บิต และบิตที่โปรแกรมได้อีก 1 บิต โดยบิตนี้จะเป็นบิตที่ 9 ซึ่งจะใช้เป็นพาริตี้บิต ในการส่งข้อมูลจะต้องเขียนไปที่ บิต TB8 ในรีจีสเตอร์ SCON สำหรับการรับข้อมูลบิตที่ 9 จะถูกเก็บในบิต RB8

- Mode 3

การทำงานในโหมดนี้จะคล้ายกับโหมด 2 แต่สามารถกำหนดค่าบอดเรตได้ โดยการโปรแกรมไปที่ไทเมอร์ 1 หลังจากการโปรแกรมแล้วยังสามารถเลือกได้อีก 2 ค่าคือ ความถี่การโอเวอร์โฟว์ของไทเมอร์ 1 หารด้วย 16 และหารด้วย 32

ตารางที่2.14 การเลือกอัตราบอดโดยการกำหนดการกำหนดค่าของไทเมอร์ 1 ได้ดังนี้

Baud Rate (บิตต่อวินาที)	ความถี่สัญญาณนาฬิกา	SMOD	Timer 1		
			C/T	โหมด	ค่ารีโหลด
โหมด0:สูงสุด 1 MHz	12 MHz	X	X	X	X
โหมด1:สูงสุด 375 KHz	12 MHz	1	X	X	X
โหมด2:ต่ำสุด 187.5 KHz	12 MHz	0	X	X	X
โหมด1,3 : 62.5 KHz	12 MHz	1	0	2	FFH
19.2 KHz	11.0592 MHz	1	0	2	FDH
9.6 KHz	11.0592 MHz	0	0	2	FDH
4.8 KHz	11.0592 MHz	0	0	2	FAH
2.4 KHz	11.0592 MHz	0	0	2	F4H
1.2 KHz	11.0592 MHz	0	0	2	E8H
137.5 Hz	11.0592 MHz	0	0	2	1DH
110 Hz	6 MHz	0	0	2	72H
110 Hz	12 MHz	0	0	1	FEEBH

2.12 การสร้างฐานเวลาให้แก่ไมโครคอนโทรลเลอร์ MCS-51

ชิป RTC มีหลายชนิด บางชนิดสามารถอินเตอร์รัปต์ไมโครคอนโทรลเลอร์ภายในช่วงเวลาที่กำหนดได้ ส่วนชิป RTC เบอร์ DS1202 ทำได้เพียงแค่ให้ข้อมูลที่เป็นเวลาในขณะใด ๆ แก่ไมโครคอนโทรลเลอร์เท่านั้น ไม่สามารถอินเตอร์รัปต์ซีพียูได้ การเลือกใช้งานชิป RTC ประเภทใด ขึ้นอยู่กับผู้ออกแบบว่าต้องการความสามารถมากน้อยขนาดไหน

ชิป RTC เบอร์ DS1202 มีความเที่ยงตรงในการทำงานสูงมาก สามารถนำมาต่อร่วมกับระบบเพื่อบอกเวลาให้แก่ไมโครคอนโทรลเลอร์ได้สะดวก เพราะใช้จำนวนสายในการติดต่อระหว่างตัวชิปเองกับไมโครอนโทรล เลอร์เพียง 3 เส้นเท่านั้น เนื่องจากชิป RTC เบอร์นี้ใช้การติดต่อรับส่งข้อมูลแบบอนุกรม คุณสมบัติคร่าว ๆ ของชิป RTC เบอร์นี้มีดังนี้

- ทำหน้าที่นับวินาที นาที ชั่วโมง วันที่ของเดือน เดือน ปี รวมทั้งคำนวณปีอธิกสุรทินให้เองโดยอัตโนมัติ

- มีหน่วยความจำขนาด 24 ไบต์สำหรับเก็บข้อมูลทั่ว ๆ ไป ส่วนใหญ่ไว้เก็บข้อมูลที่ต้องการสำรองในขณะที่ไม่มีพลังงานจ่ายให้แก่ระบบ เช่น รหัสผ่านที่เปลี่ยนค่าได้ เวลาที่ต้องการให้เครื่องจักรทำงานทำให้ไม่จำเป็นต้องสำรองหน่วยความจำทั้งระบบ

- ใช้การติดต่อแบบอนุกรม จึงใช้จำนวนสายในการเชื่อมต่อกับระบบเพียง 3 เส้นเท่านั้น

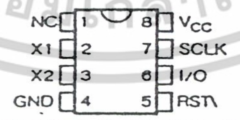
- ใช้แรงดันไฟฟ้าเพียง 2.0 ถึง 5.5 โวลต์ และใช้กระแสเพียง 300 นาโนแอมแปร์ที่ระดับแรงดัน 2.0 โวลต์

- การโอนย้ายข้อมูล สามารถกระทำได้ทั้งในแบบครั้งละ 1 ไบต์ (single byte) หรือครั้งละหลาย ๆ ไบต์ (multiple byte หรือ burst mode) ไม่ว่าจะเป็นการเขียนหรืออ่านข้อมูล

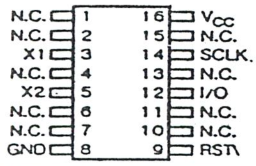
- ตัวชิปมีให้เลือกทั้งแบบ 8 ขา (8 PIN DIP) หรือ 16 ขา (16 PIN SOIC) เพื่อใช้สำหรับแผ่นวงจรชนิดเซอเฟสเมาท์ (surface mount)

- ระดับสัญญาณ TTL compatible (V_{CC} = 5 โวลต์)

- ช่วงอุณหภูมิในการใช้งานกว้างมา ระหว่าง - 40 องศา +88 องศาเซลเซียส



DS1202 8-Pin DIP(300mil)



DS1202S 16-Pin SCLC(300mil)

รูปที่ 2.26 แสดงลักษณะรูปร่างทั้งสองแบบของไอซีเบอร์ DS1202

2.12.1รายละเอียดของชิป RTC เบอร์ DS1202

ชิป RTC เบอร์ DS1202 มีเรียลไทม์คล็อก/คาเลนเดอร์ (Real Time Clock/Calendar) และสตาติกแรม (Static RAM) ขนาด 24 ไบต์ ใช้สายเพียง 3 เส้นในการเชื่อมต่อกับไมโครคอนโทรลเลอร์เพื่อรับส่งข้อมูลเกี่ยวกับเวลา ข้อมูลที่ชิป RTC DS1202 มีให้ ประกอบด้วย วินาที นาที ชั่วโมง วันที่ วัน เดือน ปี

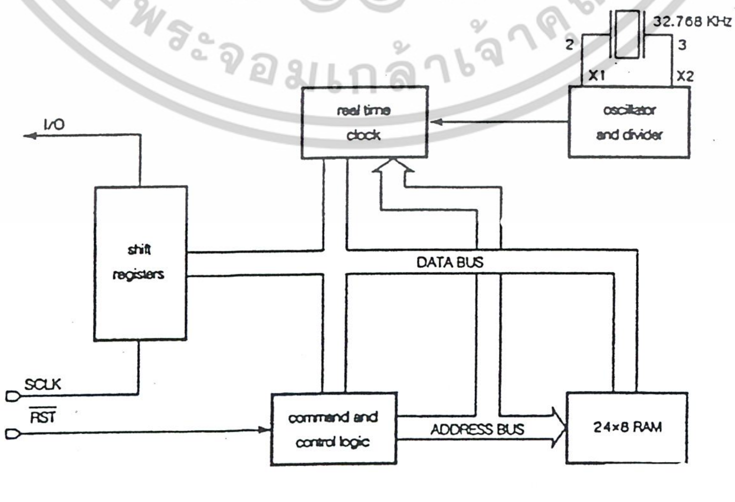
วันที่ในวันสุดท้ายของเดือนจะถูกปรับโดยอัตโนมัติสำหรับเดือนที่มีจำนวนวันน้อยกว่า 31 วัน และมีการคำนวณจำนวนวันของเดือนกุมภาพันธ์ในปีอธิกสุรทินให้เอง ข้อมูลที่ส่งให้แก่ไมโครคอนโทรลเลอร์สามารถเลือกรูปแบบได้ทั้งแบบ 24 ชั่งโมง (0.00 - 23.59 นาฬิกา) หรือแบบ 12 ชั่วโมง (0.00 - 12.00 นาฬิกา โดยมีข้อมูลเพิ่มเพื่อบอกให้ทราบว่าเป็นเวลาในช่วงเวลากลางวันหรือกลางคืน)

การเชื่อมต่อชิป RTC DS12025 เข้ากับระบบ มีความสะดวกมาก เนื่องจากใช้จำนวนสายเพียง 3 เส้นเท่านั้น เพราะใช้การติดต่อแบบอนุกรมชนิดซินโครนัส (Synchronous Serial Communication) ขาที่ต้องใช้ในการรับส่งข้อมูลทั้งสาม คือ

- ขา RST (reset)
- ขา I/O (data line)
- ขา SCLX (serial clock)

เนื่องจากในชิป RTC DS1202 มีนาฬิกาหรือเวลาที่เดินอยู่ตลอดเวลา รวมทั้งนี้มีหน่วยความจำจำนวนหนึ่ง ดังนั้นในการติดต่อชิป RTC DS1202 ผู้ใช้สามารถเลือกได้ว่าต้องการข้อมูลจากนาฬิกาหรือจากหน่วยความจำภายในชิป (CLOCK/RAM) การรับส่งข้อมูลสามารถกระทำได้ทั้งแบบทีละไบต์หรือรับส่งกันครวละหลายไบต์ นอกจากนี้ RTC DS1202 ยังถูกออกแบบให้ใช้พลังงานน้อยมาก และสิ้นเปลืองพลังงานจากแบตเตอรี่น้อยที่สุดเพื่อความสะดวกในการสำรองพลังงาน โดยชิปตัวนี้สามารถเก็บรักษาข้อมูลในหน่วยความจำและเวลาที่เดินอยู่ตลอดเวลาได้ที่กำลังไฟฟ้าน้อยกว่า 1 ไมโครวัตต์

2.12.2โครงสร้างของชิป RTC เบอร์ DS1202



รูปที่ 2.27 โครงสร้างภายในของ RTC DS1202

โครงสร้างภายในของ RTC DS1202 ประกอบไปด้วยส่วนที่สำคัญ ๆ ดังแสดงในรูปที่ 2.27 จะเห็นว่าชิป RTC เบอร์นี้ประกอบไปด้วยส่วนที่สำคัญ ๆ ดังนี้คือ

- ชิปรีจิสเตอร์ (shift register)
- คอนโทรลลอจิก (control logic)
- ออสซิลเลเตอร์ (oscillator)
- เรียลไทม์คล็อก (real time clock) และ
- หน่วยความจำ (RAM)

ในการรับหรือส่งข้อมูลใด ๆ ให้แก่ RTC DS1202 ไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์ที่ต้องการติดต่อด้วย จะต้องส่งข้อมูลที่เป็นคำสั่งควบคุมการติดต่อซึ่งมีขนาด 8 บิตเสียก่อนโดยเริ่มต้นด้วยการให้ขา RTC มีสถานะเป็น 1 (อยู่ในช่วงการติดต่อ) จากนั้นส่งข้อมูลจำนวน 8 บิตเข้าไปไว้ในชิปรีจิสเตอร์ของ RTC ข้อมูลขนาด 8 บิตจะประกอบด้วยคำสั่งในการควบคุมชิป RTC และตำแหน่งของหน่วยความจำที่ต้องการติดต่อ(address/command byte) ในแต่ละครั้ง การับข้อมูลจากไมโครโปรเซสเซอร์แต่ละบิตจะกระทำที่ช่วงขอบขาขึ้นสัญญาณที่ขา SCLK (serial clock)

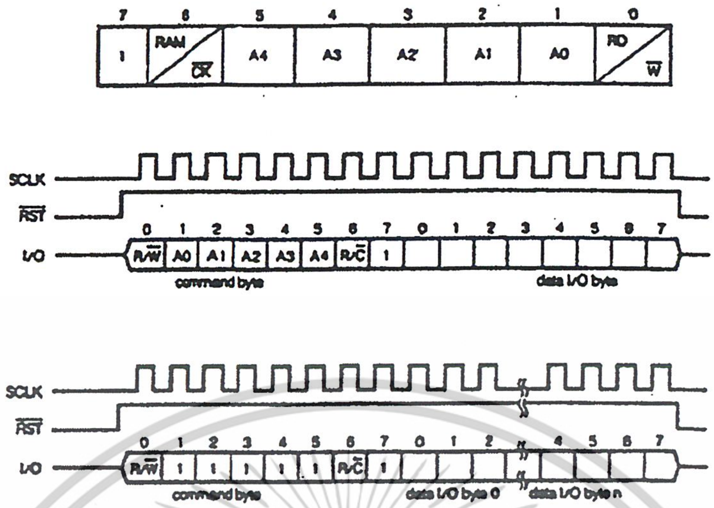
ภายในชิป RTC เบอร์ DS1202 ประกอบด้วยหน่วยความจำขนาด 24 ไบต์และรีจิสเตอร์ที่ทำหน้าที่เก็บเวลาของชิปในขณะปัจจุบันจำนวน 8 ตัว รีจิสเตอร์ทั้ง 8 ตัวนี้ สามารถเข้าถึงได้เสมือนเป็นหน่วยความจำตำแหน่งหนึ่ง ดังนั้นต่อไปชิป RTC เบอร์นี้มีหน่วยความจำรวมทั้งสิ้น 32 ตำแหน่ง โดยประกอบขึ้นจากรีจิสเตอร์ 8 ตำแหน่งและหน่วยความจำ 24 ตำแหน่ง

ข้อมูลขนาด 8 บิตแรก (address/command byte) จะระบุตำแหน่งหน่วยความจำที่ต้องการติดต่อ (ทั้งตำแหน่งของหน่วยความจำทั่วไปและตำแหน่งของรีจิสเตอร์สำหรับเก็บเวลา) และบอกว่าเป็นการเขียนหรืออ่านข้อมูลจากหน่วยความจำตำแหน่งนั้น ๆ รวมทั้งระบุว่าให้การรับส่งข้อมูลเป็นแบบครั้งละ 1 ไบต์ หรือครั้งละหลาย ๆ ไบต์

หลังจากมีสัญญาณนาฬิกาเกิดขึ้น 8 ครั้ง (สัญญาณ SCLK) ในระหว่างการเขียนข้อมูล 8 บิต แรกเข้าไปในชิปรีจิสเตอร์ สัญญาณนาฬิกาต่อไปที่จะเกิดขึ้นจะเป็นการนำข้อมูลออกจากชิป RTC สำหรับการอ่านข้อมูลหรือนำข้อมูลเข้าไปยังชิป RTC สำหรับการเขียนข้อมูล จำนวนของสัญญาณนาฬิกาทั้งหมดที่เกิดขึ้นในการติดต่อครั้งหนึ่ง ๆ จึงเท่ากับ 8+8 ในการส่งข้อมูลครั้งละ 1 ไบต์ (single byte mode) หรือ 8 บวก มากที่สุด 192 (8X24) สำหรับการส่งข้อมูลครั้งละหลาย ๆ ไบต์

2.12.3 โครงสร้างของคอมมานด์ไบต์ (Command Byte)

โครงสร้างคอมมานด์ไบต์มีดังแสดงในรูปที่ 2.28



รูปที่ 2.28 แสดงโครงสร้างของคอมมานด์ไบต์

การรับหรือส่งข้อมูลระหว่างชิป RTC DS1202 ในตอนเริ่มต้น ไมโครคอนโทรลเลอร์จะต้องส่งข้อมูลเพื่อกำหนดการทำงานให้แก่ RTC ก่อน ข้อมูลที่ RTC ได้รับในตอนเริ่มต้นนี้จะมีขนาด 1 ไบต์ ซึ่งมีชื่อว่าคอมมานไบต์ และเนื่องจากข้อมูลในไบต์นี้จะเป็นตัวกำหนดการทำงานของ RTC ดังนั้นแต่ละบิตในไบต์นี้จะมีความหมายแตกต่างกันต่อไปดังนี้

: MSB (บิต 7) ต้องเป็น 1 เสมอ ถ้าเป็น 0 การทำงานต่อจากนี้จะถูกหยุดไว้หมด

: บิต 6 ถ้าเป็น 0 จะระบุว่าต้องการติดต่อกับรีจิสเตอร์สำหรับเก็บเวลา (clock/calender register) ดังนั้นข้อมูลที่รับส่งกันจะเป็นเวลา หากบิตนี้มีค่าเป็น 1 จะระบุว่าต้องการติดต่อกับหน่วยความจำ

: บิต 1 ถึง 5 เป็นตัวระบุตำแหน่งหน่วยความจำ (ทั้งหน่วยความจำที่ใช้เป็นรีจิสเตอร์สำหรับเก็บเวลาและหน่วยความจำทั่วไป) ที่ต้องการเข้าถึง ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูล ซึ่งควบคุมด้วยบิต

- เบิสท์โหมด (Burst Mode) หมายถึงการรับหรือส่งข้อมูลครั้งละหลายไบต์ในการติดต่อแต่ละครั้ง โดยสามารถกำหนดได้ว่าข้อมูลที่ต้องการติดต่อเป็นรีจิสเตอร์สำหรับเก็บเวลาหรือหน่วยความจำที่ใช้เก็บข้อมูล หากเป็นการรับส่งข้อมูลกับรีจิสเตอร์สำหรับเก็บเวลาก็จะรับส่งกันครั้งละ 24 ไบต์ การกำหนดให้รับส่งข้อมูลครั้งละหลายไบต์กำหนดโดยคอมมานด์ไบต์

ในการรับส่งในเบิสท์โหมด จะเริ่มต้นที่บิต 0 ของหน่วยความจำตำแหน่ง 0 ก่อนเสมอ ไม่ว่าจะเป็นรีจิสเตอร์สำหรับเก็บเวลาหรือความจำสำหรับเก็บข้อมูลทั่วไป

เบิสท์โหมดนี้มีไว้เพื่อความสะดวกในการรับส่งข้อมูลครั้งละจำนวนมากๆ ทำให้ไม่ต้องส่งคอมมานด์ไบต์หลายครั้งนั่นเอง

- ไรท์โพรเท็ค คอมมานด์ไบต์ (Write Protect Command Byte) เวลาที่เดินอยู่ภายในรีจิสเตอร์ที่ทำหน้าที่เก็บเวลาและข้อมูลที่อยู่ในหน่วยความจำทั้ง 24 ตำแหน่ง สามารถป้องกันไม่ให้เขียนข้อมูลใด ๆ

ซ้อนลงไปได้ เพื่อป้องกันเหตุการณ์ที่เกิดขึ้นโดยไม่ได้ตั้งใจ ทั้งนี้โดยการควบคุมจากไรท์โพรเท็ค คอมมานด์ไบต์ซึ่งเป็นรีจิสเตอร์ตำแหน่งที่ 7 ของรีจิสเตอร์ที่ทำหน้าที่เก็บเวลา โดยมีไรท์โพรเท็คบิต (write protect bit) ซึ่งเป็นบิตที่ 7 ของ รีจิสเตอร์ตัวนี้เป็นตัวกำหนดการทำงาน หากไรท์โพรเท็คบิตเป็น 0 หมายถึงสามารถเขียนข้อมูลใด ๆ ลงไปยัง รีจิสเตอร์ที่เก็บเลาหรือหน่วยความจำได้ หากบิตนี้เป็น 1 หมายถึงชิป RTC อยู่ในสถานะป้องกันการเขียนข้อมูล ดังนั้นก่อนการเขียนข้อมูลใด ๆ ไปยังรีจิสเตอร์สำหรับเก็บเวลาหรือหน่วยความจำ ไรท์โพรเท็คบิตต้องเป็น 0 เสมอ โดยการให้ RST เป็น 1 (อยู่ในช่วงการติดต่อ) และโหลดไรท์โพรเท็คชั่น คอมมานด์ไบต์ (8EH) ตามด้วยข้อมูลที่มีค่า OOH (เคลียร์ให้ไรท์โพรเท็คบิตเป็น 0) หลังจากนั้น RST ต้องกลับมามีสถานะเป็น 0 ก่อนที่คำสั่งอื่น ๆ จะเริ่มต้นทำงานได้

ส่วนในการบังคับให้ชิป RTC อยู่ในสถานะป้องกันการเขียนข้อมูล ก็ต้องให้ RST เป็น 1 แล้วโหลดคอมมานด์ไบต์ที่มีค่า 8EH ตามด้วยข้อมูล 8OH (ให้ไรท์โพรเท็คบิตเป็น 1)

ในการทำงานแบบเบิสท์โหมด เราไม่สามารถเขียนค่าใด ๆ เข้าไปไรท์โพรเท็คบิตได้

- รีเซตแอนด์คล็อกคอนโทรล (Reset and Clock Control) การรับหรือส่งข้อมูลทั้งหมดจะต้องเริ่มโดยใช้ขา RST มีสถานะเป็น 1 ก่อนเสมอ โดย RST มีหน้าที่หลักอยู่ 2 ประการดังนี้

1. RST ใช้ควบคุมการเขียนหรืออ่านข้อมูลในชิปรีจิสเตอร์

2. RST ใช้เป็นสัญญาณในการหลุดการทำงานใด ๆ กับชิป RTC DS1202 โดยปกติการเขียนข้อมูลเข้าไปใน RTC DS1202 จะเกิดขึ้นในช่วงขอบขาขึ้นของสัญญาณที่ขา SCLK ส่วนการอ่านข้อมูลจาก RTC DS1202 จะเกิดขึ้นในช่วงขอบขาลงของสัญญาณที่ขา SCLK โดยในระหว่างการติดต่อ ขา RST ต้องมีสถานะเป็น 1 ตลอดเวลา หากขา RST มีสถานะเป็น 0 หมายถึงยกเลิกการติดต่อหรือสิ้นสุดการติดต่อ

- ดาต้าอินพุต (Date Input) ในตอนเริ่มต้นติดต่อระหว่างชิป RTC DS1202 กับไมโครคอนโทรลเลอร์ไบต์แรกจะต้องเป็นคอมมานด์ไบต์เสมอ หากในคอมมานด์ไบต์ระบุว่าเป็นการเขียนข้อมูลไปในชิป ข้อมูลจะถูกรับเข้ามาในช่วงขอบขาขึ้น (rising edge) ของ SCLK เท่านั้น โดยเริ่มต้นด้วยบิต 0 ก่อนเสมอ และหากเป็นคำสั่งให้รับส่งครั้งละ 1 ไบต์ เมื่อข้อมูลได้รับเข้ามาครบแล้ว สัญญาณ SCLK ที่รับได้เกินจะถูกละเลยไป หากเป็นคำสั่งให้รับส่งแบบเบิสท์โหมดซึ่งรับส่งครั้งละ 24 ไบต์ก็มีลักษณะเช่นเดียวกันคือ เมื่อรับข้อมูลครบ 24 ไบต์แล้ว สัญญาณ SCLK ที่รับได้เกินจะถูกละเลยเช่นกัน

- ดาต้าเอาท์พุต (Data Output) หลังจากรับคอมมานด์ไบต์แล้ว หากมีการระบุว่าเป็นการอ่านข้อมูลจากชิป RTC DS1202 ข้อมูลจะถูกส่งออกจากชิปสู่ภายในขณะช่วงขอบขาลง (falling edge) ของ SCLK หลังจากมีการรับคอมมานด์ไบต์เรียบร้อยแล้ว นั่นคือบิตแรกที่จะถูกส่งออกจาก RTC จะเกิดขึ้นในช่วงขอบขาลงของสัญญาณนาฬิกาลูกที่ 9 นั่นเอง (ต่อจากคอมมานด์ไบต์)

- คล็อก/คาเลนเดอร์ (Clock Halt Flag) บิต 7 ของรีจิสเตอร์ที่เก็บค่าวินาที จะเป็นตัวบอกให้ชิป RTC DS1202 หยุดการทำงานของวงจรในส่วนออสซิลเลเตอร์เมื่อบิตนี้มีค่าเป็น 1 ซึ่งเป็นผลให้นาฬิกาภายในชิปหยุดการทำงานไปด้วย และจะบังคับให้ชิปอยู่ในสถานะใช้พลังงานต่ำ (low power standby mode) โดยใช้กระแสไม่เกิน 100 นาโนแอมแปร์ และเมื่อบิตนี้เป็น 0 อีกครั้ง วงจรออสซิลเลเตอร์จะเริ่มทำงานต่อทันที

- โหมดเลือก AM-PM/12-24 ชั่วโมง (AM-PM/12-24 mode) บิต 7 ของรีจิสเตอร์ที่เก็บค่าชั่วโมงถูกกำหนดให้เป็น 12/24 ชั่วโมง (hour mode select bit) นั่นคือเป็นตัวเลือกว่าจะให้รีจิสเตอร์นี้เก็บค่าชั่วโมงแบบ 12 ชั่วโมงหรือ 24 ชั่วโมง โดย

: บิตที่ 7 เป็น 1 จะเป็ฯการเลือกให้เก็บค่าแบบ 12 ชั่วโมง โดยมีบิต 5 เป็นตัวบอกว่าเป็นช่วงกลางวันหรือกลางคืน (AM/PM indicator) โดย 1 จะหมายถึงกลางคืน และ 0 หมายถึงกลางวัน

: บิตที่ 7 เป็น 0 จะเป็นการเลือกให้เก็บค่าแบบ 24 ชั่วโมง และบิต 5 จะเป็นบิตที่แสดงหลักสิบตัวที่ 2 ของชั่วโมง (20-23)

- ไรท์โพรเท็ค (Write Protect) บิต 7 ของไรท์โพรเท็ครีจิสเตอร์จะเป็นไรท์โพรเท็คบิต โดย 7 บิตแรกถูกบังคับให้เป็น 0 หมด ทำให้อ่านค่าได้เป็น0 เสมอ

- คล็อก/คาเลนเดอร์เบิสท์โหมด (Clock/Calender Burst mode) คอมมานด์ไบต์มีค่า BEH จะเป็นการระบุว่าให้มีการเขียนข้อมูลในรีจิสเตอร์ที่เก็บเวลาในแบบเบิสท์โหมด หากคอมมานด์ไบต์มีค่า BFH จะระบุว่าให้มีการเขียนข้อมูลในรีจิสเตอร์นี้ในแบบเบิสท์โหมดเช่นกัน ซึ่งในคล็อก/คาเลนเดอร์เบิสท์โหมดจะมีข้อแตกต่างกันก็เพียงจำนวนข้อมูลที่รับหรือส่งเท่านั้น เพราะหน่วยความจำในชิป RTC เบอร์ DS1202 มีขนาด 24 ไบต์ ดังนั้นในแรกเบิสท์โหมด (RAM burst mode) นี้ ข้อมูลจำนวน 24 ไบต์จะรับส่งกันโดยเริ่มต้นด้วยบิต 0 ของหน่วยความจำตำแหน่ง 0 ก่อนเสมอ

2.12.4 การเลือกได้คริสตอล

ชิป RTC DS1202 ใช้คริสตอลความถี่ 32.768 กิโลเฮิรตซ์ เป็นตัวกำหนดคาบเวลาในการทำงาน ซึ่งมักจะใช้ของบริษัทไดวา เบอร์ DT26S หรือของบริษัทไซโก้ เบอร์ DS-VT-200 หรือเบอร์ที่เทียบเท่ากัน

คริสตอลความถี่ 32.762 กิโลเฮิรตซ์นี้ สามารถต่อโดยตรงเข้ากับขา 203 (x1, x2) ของ DS1200 ได้เลย โดยค่าอิมพีแดนซ์ของตัวคริสตอลเองควรเป็นค่าคาพาซิแดนซ์ (CL) ขนาด 6 พิโคฟารัด

ตารางที่ 2.15 แสดงการใช้เบิสท์โหมด

BURST FUNCTION	DATA (BYTESS)	SCLK
CLOCK	8	72
RAM	24	200

ตารางที่ 2.16 ตารางข้อมูลการทำงานของรีจิสเตอร์และ RAM ใน DS 1202

รีจิสเตอร์	ฟังชันท์	COMMAND ADDRESS (HEX)	เขียน = W อ่าน = R	Range DATA (BCD)	รีจิสเตอร์ที่กำหนด							
					7	6	5	4	3	2	1	0
0	วินาที	80 81	W R	00-59	CH	10วินาที			วินาที			
1	นาที	82 83	W R	00-59	0	10นาที			นาที			
2	12ชม. 24ชม.	84 85	W R	01-12	12/24	0 0	AP 10	ชม. ชม.	ชั่วโมง			
3	วัน	86 87	W R	00-23 01-31	0	0	10 วัน		วัน			
4	เดือน	88 89	W R	01-12	0	0	0	10 เดือน	เดือน			
5	วัน	8A 8B	W R	01-07	0	0	0	0	วัน			
6	ปี	8C 8D	W R	00-99	10 ปี				ปี			
7	Write Protect	8E 8F	W R	00-80	W/P	เป็น 0 ทั้งหมด						

2.13การใช้งานแอลซีดีโมดูล (LCD Module)

ปัจจุบันแอลซีดี เป็นที่นิยมกันเป็นอย่างมาก สำหรับการแสดงผลในเครื่องมือเครื่องใช้ต่าง ๆ ทั้งนี้เนื่องจากมีความเหมาะสมหลายด้าน เช่น การใช้กระแสต่ำ สามารถแสดงผลเป็นดังอักษรและตัวเลขหรือแสดงเป็นกราฟฟิก (เฉพาะรุ่น) จะมีปัญหาก็คือในด้านวงจรซึ่งมีระบบการทำงานที่ซับซ้อนและหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิตแอลซีดีจะทำรุ่นที่เป็นแอลซีดีโมดูลออกมาคือเป็นโมดูลที่มีตัวแอลซีดีและวงจรควบคุมมาให้พร้อม (เรียกว่า แอลซีเอ็ม) ซึ่งทำให้ผู้ใช้สามารถต่อเข้าระบบไมโครคอนโทรลเลอร์ได้ง่ายและสะดวกสำหรับการเขียนโปรแกรม รวมทั้งมีจำหน่ายกันอย่างกว้างขวางและมีราคาที่เหมาะสม ทำให้ผู้ใช้ทางด้านไมโครคอนโทรลเลอร์หันมาใช้แสดงผลด้วยแอลซีดีโมดูลกันมากขึ้น

แอลซีดีโมดูลมีอยู่หลายรุ่นและคุณสมบัติแตกต่างกันออกไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลัก คือแบบด็อท เมตริก (Dot Matrix) และกราฟฟิก (Graphic) โดยแบบด็อท เมตริกจะแสดงผลเป็นตัวอักษรขนาด 5 X 8 ด็อทและมีจำนวนอักษรและบรรทัดแตกต่างกันออกไปในแต่ละรุ่น ส่วนแบบกราฟฟิกจะสามารถแสดงผลในแบบบิตแม็ป (Bit map) คือจะสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ แนวทางในการใช้งานของทั้งสองแบบจะมีคุณลักษณะที่ใกล้เคียงกัน การใช้งานโดยทั่วไปมักใช้แบบด็อท เมตริกมากกว่าเนื่องจากราคาถูกกว่าและเพียงพอต่อการใช้งานใหญ่ คุณสมบัติของด็อท เมตริกแอลซีดีโมดูลสามารถสรุปเป็นข้อ ๆ ดังนี้

1. มีให้เลือกหลายรุ่นตามความต้องการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วยด็อท เมตริกขนาด 5 X 8 ด็อท
3. สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะคือ แบบเมมโมรีแม็ป และแบบผ่าน 8255 พอร์ตซึ่งจะใช้ขาสัญญาณทั้งหมด 14 พิน
4. การใช้งานง่ายและสะดวก ระบบไมโครคอนโทรลเลอร์เพียงแค่ส่งข้อมูลให้กับ แอลซีดีโมดูลเท่านั้น ข้อความก็จะปรากฏบนแผงแสดงและจะค้างไว้ตลอดทำให้ไม่ต้องเสียเวลาของระบบ
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น เคลียร์ (Clear) , ดิสเพลย์ (Display) , โฮม (Home) , เคอร์เซอร์ออนอ็อฟ (Cursor On Off) และอื่น ๆ อีก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัวรวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสน้อยและมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์ เท่านั้น

2.13.1 ขาสัญญาณของ แอลซีดีโมดูล

ตารางที่ 2.17 ขาสัญญาณของแอลซีดีโมดูล

ขา	สัญลักษณ์	ระดับ	หน้าที่
1	V_{SS}	-	0 โวลต์ กราวด์
2	V_{CC}	-	+5 โวลต์ พาวเวอร์ซัพพลาย
3	V_{ee}	-	+โวลต์ ลิขวิด คริสตอล ไดรฟ์ (Liquid Crystal Drivr)
4	RS	H/L	เลือกกกรีจิสเตอร์ (register select) H : Data Input L : Instruct Input
5	R/W	H/L	H : Data Read L : Data Write
6	E	H	Enable Signal (L-H)
7	DB0	H/L	Data Bus Bit 0
8	DB1	H/L	Data Bus Bit 1

ขา	สัญลักษณ์	ระดับ	หน้าที่
9	DB2	H/L	Data Bus Bit 2
10	DB3	H/L	Data Bus Bit 3
11	DB4	H/L	Data Bus Bit 4
12	DB5	H/L	Data Bus Bit 5
13	DB6	H/L	Data Bus Bit 6
14	DB7	H/L	Data Bus Bit 7

2.13.2 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์

แอลซีดีโมดูล จะต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะได้กล่าวมาแล้ว ซึ่งทั้งสองแบบนี้จะมีข้อดีและข้อเสียแตกต่างกันออกไป โดยแต่ละแบบจะมีหลักการดังนี้

1) การต่อแบบเมมโมรีแม็ป

1. สามารถต่อเข้ากับชิปเบอร์ทั่ว ๆ ไปได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครคอนโทรลเลอร์มองเห็นแอลซีดีโมดูลในลักษณะของเมมโมรีได้ทันที
2. ผู้ใช้สามารถเขียนและอ่านข้อมูลจากแอลซีดีโมดูลได้ ทำให้มองเสมือนว่าเป็นเมมโมรีบัฟเฟอร์ไปในตัว
3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบแฟล็กความพร้อมในขณะที่แอลซีดีโมดูลกำลังทำงานได้
4. ใช้ได้กับบอร์ดที่มีแอลซีดีบัสมาให้พร้อม
5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการดีโค้ดละเอียดพอสมควร
6. การจัดขาสัญญาณจะต้องเป็นไปตามแบบของชิป

2) การต่อแบบ อินพุท/เอาท์พุท พอร์ต

1. สามารถต่อเข้ากับอินพุต/เอาท์พุตพอร์ตใด ๆ ก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมา ให้ตรงกับข้อกำหนดของแอลซีดีโมดูล
2. ผู้ใช้จะเขียนข้อมูลให้แอลซีดีโมดูลได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนดเมมโมรีส่วนหนึ่งให้เสมือนบัฟเฟอร์ให้กับแอลซีดีโมดูล
3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วยเวลาของระบบเอง เพื่อรอให้แอลซีดีโมดูลกระทำขบวนการต่าง ๆ
4. ใช้ได้กับบอร์ดทั่ว ๆ ไปที่มีพอร์ต
5. ไม่เปลืองส่วนของเมมโมรีในการใช้งาน
6. การจัดขาสัญญาณกระทำได้อย่างอิสระ

2.13.3ชุดคำสั่งควบคุมและการแสดงข้อความ

ขาสัญญาณ VEE มีไว้สำหรับกำหนดความเข้มของตัวอักษร โดยถ้าต่อกับกราวด์ (GND) จะมีความเข้มสูงสุด แต่ถ้าต่อกับ VCC จะมีความเข้มต่ำสุด ปกติแอลซีดีรุ่นธรรมดาอาจจะต่อกับกราวด์ไว้เลยก็ได้ ไม่ต้องใส่ VR ให้สิ้นเปลือง แต่ถ้าเป็นรุ่น STN (มุมมองกว้าง) ให้ใช้ตัวต้านทาน 2 กิโลโอห์มต่อลงกราวด์อีกที เพื่อให้ความเข้มมีความเหมาะสม การเขียนหรืออ่านข้อมูลกับแอลซีดีโมดูล ก็คือการกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของแอลซีดีตามชุดคำสั่งควบคุม และรวมไปถึงการเขียนข้อมูลที่เป็นข้อความเพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางที่ 2.18

1) ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับแอลซีดีโมดูล จะแบ่งเป็น 2 ลักษณะ คือ สัญญาณควบคุม (Instruction) และข้อมูล (Data) โดยจะกำหนดด้วยขาสัญญาณ RS คือ ถ้า RS = 0 จะหมายถึงส่งสัญญาณควบคุม หรืออ่านค่าเฟล็กสภาพทำงานงานของแอลซีดีโมดูล และถ้า RS = 1 จะหมายถึงการเขียนหรือการอ่านข้อมูลกับแอลซีดีโมดูล
2. หลักการในการเขียนข้อมูลให้แอลซีดีนี้คือ เมื่อมีการเขียนข้อมูลไปแล้ว ตัวแอลซีดีโมดูลจะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง {ตามค่าเอ็กซ์คิวท์ ไทม์ (execte time) ในตารางที่ 2.8 ซึ่งระบบไมโครคอนโทรลเลอร์สามารถตรวจสอบได้จากบีซี่แฟล็ก (Busy Flag : BF) และถ้าเรียบร้อยแล้วจึงจะสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบอินพุต/เอาท์พุตพอร์ต คือ ไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครคอนโทรลเลอร์ก็จะต้องใช้วิธีหน่วงเวลาแทน
3. การเขียนข้อมูลให้กับแอลซีดีโมดูลนี้ สามารถทำได้ทั้งแบบ 8 บิต และ 4 บิต โดยกรณี 4 บิต จะใช้สายสัญญาณข้อมูลเพียง 4 เส้น คือ DB4-DB7 (ใช้สำหรับระบบไมโครคอนโทรลเลอร์แบบ 4 บิตหรือเพื่อต้องการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 บิต เพียงแต่ให้เขียน 2 ครั้ง คือ DB4-DB7 ก่อน แล้วตามด้วย DB0-DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่งฟังก์ชันเซ็ต (Function set) ด้วย
4. DDRAM (Display Data Ram) คือ หน่วยความจำภายในตัวแอลซีดีโมดูลที่เป็นบัฟเฟอร์ของข้อมูล โดยถ้าเขียนเป็นรหัส ASCII ใด ๆ ลงไปในหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่แผงแสดงทันที
5. CGRAM (Character Generator Ram) คือ หน่วยความจำภายในตัวแอสซีดีโมดูล สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โดยจะอ้างแอดเดรสได้ทั้งหมด 64 ไบต์ คือ 8 ตัวอักษร คูณ กับ 8 แถว

ตารางที่ 2.18 การกำหนดคุณสมบัติในการใช้งานของแอลซีดีตามชุดคำสั่งควบคุม

Instruction	RS	RW	Data Bit								EXE.
			7	6	5	4	3	2	1	0	Time
Clear Diaplsy	0	0	0	0	0	0	0	0	0	1	1640
Cursor At Home	0	0	0	0	0	0	0	0	1	*	1640
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	40
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	40
Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	40
Function Set	0	0	0	0	1	DL	N	F	*	*	40
Set CGRAM Add.	0	0	0	1	CGRAM ADDRESS						40
Set DDRAM Add.	0	0	1	DDRAM ADDRESS							40
Busy, Add. Read	0	1	BF	ADDRESS							0
CGRAM, DDRAM WR	1	0	WRITE DATA								40
CGRAM, DDRAM RD	1	1	READ DATA								40

2) รายละเอียดของแต่ละคำสั่ง

1. เคลียร์ดิสเพลย์ (Clear Display) ดังแสดงในตารางที่ 2.19

ตารางที่ 2.19 แสดงการเคลียร์ดิสเพลย์

RS	R/W	DB7	DB6	DB5	DB4	DB4	DB3	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับการเคลียร์ดิสเพลย์ โดยจะทำการเขียนตัวอักษรสเปซ (Space) ลงไปใน DDRAM ทั้งหมด และทำการกำหนดค่า DDRAM แอดเดรสให้เป็น 0 พร้อมทั้งเคอร์เซอร์จะกลับไปที่ตำแหน่งซ้ายบนสุดของจอภาพ

2. เคอร์เซอร์ แอท โฮม (Cursor At Home) ดังแสดงในตารางที่ 2.20

ตารางที่ 2.20 แสดงการเคอร์เซอร์ แอท โฮม

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	.

สำหรับกำหนดค่า DDRAM แอดเดรสให้เป็น 0 พร้อมทั้งเคอร์เซอร์จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

3.เอ็นทรี โหมด เซ็ท (Entry Mode Set) ดังแสดงในตารางที่ 2.21

ตารางที่ 2.21 แสดงการเอ็นทรี โหมด เซ็ท

RS	R/w	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

I/D = 0 กำหนดทิศทางของเคอร์เซอร์และ DDRAM ให้เป็นแบบลดลง (Decrement)

I/D = 1 กำหนดทิศทางของเคอร์เซอร์และ DDRAM ให้เป็นแบบเพิ่มขึ้น (Increment)

S = 0 เมื่อเขียนข้อมูลแล้ว ตัวเคอร์เซอร์จะถูกเลื่อนไปตามทิศทางของค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้ว ตัวเคอร์เซอร์จะอยู่กับที่ และตัวอักษรจะถูกดันไปตามทิศทางค่า I/D

การกำหนด I/D และ S นี้ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้วจะต้องไม่ใช้คำสั่งเคลียร์ดิสเพลย์อีก

4.ดิสเพลย์ออน/อ็อฟ (Display On/Off) ดังแสดงในตารางที่ 2.22

ตารางที่ 2.22 แสดงการดิสเพลย์ออน/อ็อฟ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D = 0 กำหนดให้อ็อฟดิสเพลย์ (Off display)

D = 1 กำหนดให้ออนดิสเพลย์ (On display)

C = 0 กำหนดให้อ็อฟเคอร์เซอร์ (Off cursor)

C = 1 กำหนดให้ออนเคอร์เซอร์ (On cursor) โดยเคอร์เซอร์จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่งเคอร์เซอร์

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่งเคอร์เซอร์ (กระพริบเป็นรูปสี่เหลี่ยม)

5.ดิสเพลย์ชิป (Display Shift) ดังแสดงในตารางที่ 2.23

ตารางที่ 2.23 แสดงการดิสเพลย์ชิป

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

S/C = 0 กำหนดให้เลื่อนเคอร์เซอร์ตามทิศทาง R/L ไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความบนแผงแสดงตามทิศทาง R/L ไป 1 คอลัมน์ (เลื่อนทุกบรรทัด)

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

6. ฟังก์ชันเซ็ต (Function Set) ดังแสดงในตารางที่ 2.24

ตารางที่ 2.24 แสดงการฟังก์ชันเซ็ต

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้การติดต่อกับแอลซีดีโมดูลเป็นแบบ 4 บิต

DL = 1 กำหนดให้การติดต่อแอลซีดีโมดูลเป็นแบบ 8 บิต จะสังเกตว่าการกำหนดค่า D/L นี้สามารถ

กระทำได้ที่ DB4-DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 บิตตั้งแต่ครั้งแรกหลังจากจ่ายไฟ

เลี้ยง ก็จะทำให้แอลซีดีมีการรับข้อมูล 4 บิตทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 ดิวตี้ (Duty) และ 1/11 ดิวตี้

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 ดิวตี้

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5*7 ด็อท

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5*10 ด็อท (กรณีที่แอลซีดีเป็นแบบ 5*7 อยู่แล้วก็จะไม่มีผลอะไร)

7. เซ็ต CGRAM แอดเดรส (Set CGRAM Address) ดังแสดงในตารางที่ 2.2 5

ตารางที่ 2.25 แสดงการเซ็ต CGRAM แอดเดรส

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

แอดเดรสของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียนดาต้าที่ต่อจากนี้ จะเป็นไปตามแอดเดรสที่กำหนดทันที

8. เซ็ต DDRAM แอดเดรส (Set DDRAM Address) ดังแสดงในตารางที่ 2.26

ตารางที่ 2.26 แสดงการเซ็ต DDRAM แอดเดรส

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

สำหรับการกำหนดแอดเดรสของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การจ่ายและเขียนดาต้าที่ต่อจากนี้ จะเป็นไปตามแอดเดรสที่กำหนดทันที ตำแหน่งของแอดเดรสในแต่ละรุ่นจะมีความแตกต่างกันบ้างเพราะจำนวนตัวอักษรต่อบรรทัดไม่เท่ากัน ซึ่งแสดงดังตารางต่อไปนี้ (ตารางนี้จะกำหนดให้บิตที่ 7 เท่ากับ 1 เสมอ เพื่อความสะดวกในการเรียกใช้)

ตารางที่ 2.27 แสดงตำแหน่งแอดเดรสของแอลซีดี รุ่น DMC164

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF

9. บีซี่แฟล็ก แอ็นด์ แอดเดรสรีด (Busy Falg And Address Read) ดังแสดงในตารางที่ 2.28

ตารางที่ 2.28 แสดงการบีซี่แฟล็ก แอ็นด์ แอดเดรสรีด

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	ADDRESS						

สำหรับการอ่านค่าบีซี่แฟล็ก ซึ่งบอกถึงความพร้อมของแอลซีดีในการรับข้อมูล ถ้า BF = 0 หมายความว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายความว่ายังไม่พร้อม นอกจากนี้การอ่านค่าแอดเดรสของ CGRAM หรือ DDRAM ด้วย

3) การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM

1. การเขียนข้อมูลให้ DDRAM หรือ CGRAM (Write Data To DDRAM or CGRAM) ดังแสดงในตารางที่ 2.29

ตารางที่ 2.29 แสดงการเขียนข้อมูลให้ DDRAM หรือ CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	DATA							

สำหรับการเขียนข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้วแอดเดรสจะถูกเพิ่มหรือลงโดยอัตโนมัติตามที่กำหนดจากค่า I/D ในคำสั่งเอ็นทรีโหมดเซ็ต และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้มีการกำหนดแอดเดรสที่ใด

2. การอ่านข้อมูลจาก DDRAM หรือ CGRAM (Read Data From DDRAM or CGRAM) ดังแสดงในตารางที่ 2.30

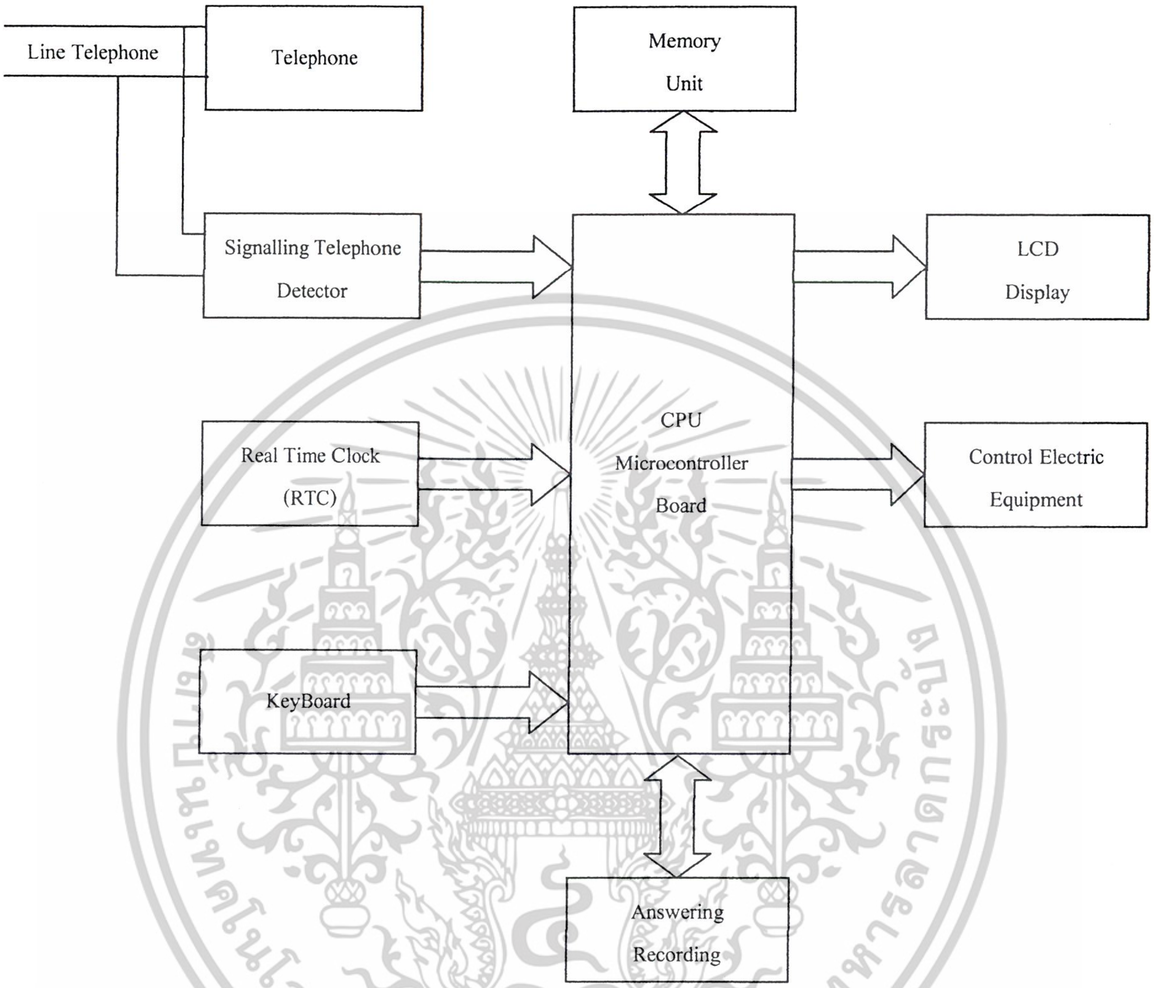
ตารางที่ 2.30 แสดงการอ่านข้อมูลจาก DDRAM หรือ CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	DATA							

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้วแอดเดรสจะถูกเพิ่มหรือลงโดยอัตโนมัติตามที่กำหนดจากค่า I/D ในคำสั่งเอ็นทรีโหมดเซ็ต และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่าก่อนหน้าคำสั่งนี้มีการกำหนดแอดเดรสที่ใด

บทที่ 3
การคำนวณและการสร้าง

3.1 Block Diagram



รูปที่ 3.1 Block Diagram

3.1.1 หลักการทำงานของโครงงาน

จากบล็อกไดอะแกรมเครื่องนี้จะทำการต่อเข้ากับโทรศัพท์ในลักษณะต่อขนาน จึงจะสะดวกสบายไม่ต้องไปแก้ไขส่วนใดในวงจรของเครื่องรับโทรศัพท์เลย หลักการทำงานเบื้องต้นของเครื่องนี้จะแบ่งการทำงานออกเป็น 2 กรณี คือ

1. กรณีที่มีการโทรเข้ามา
2. กรณีของการโทรออกไป

1. กรณีที่มีการโทรเข้ามา

ในสภาวะปกติ เครื่องจะทำการตรวจสอบสัญญาณเรียกอยู่ตลอดเวลา ว่ามีสัญญาณเรียกจากชุมสายหรือไม่ ถ้าเกิดมีสัญญาณเรียกจากชุมสายเกิดขึ้น ไมโครคอนโทรลเลอร์ก็จะสั่งงานให้เข้าสู่

โปรแกรมตอบรับอัตโนมัติ โดยจะตรวจนับจำนวนกระดิ่งที่ดังว่าดังครบเท่ากับจำนวนที่เราตั้งเอาไว้หรือไม่ ถ้าครบก็จะทำการยกหูอัตโนมัติและสั่งการให้วงจรตอบรับทำงาน แจ้งให้คู่สนทนาทราบว่าขณะนี้ไม่มีใครอยู่ และกำลังเข้าสู่ระบบอัตโนมัติ จากนั้นเครื่องก็จะให้เลือกโหมดการทำงานซึ่งจะแบ่งได้เป็น 2 โหมด คือ

1.1 โหมดการฝากข้อความอัตโนมัติ คือ คู่สนทนาสามารถที่จะฝากข้อความลงในหน่วยความจำของไมโครคอนโทรลเลอร์ และสามารถเล่นฟังข้อความที่บันทึกไว้ได้

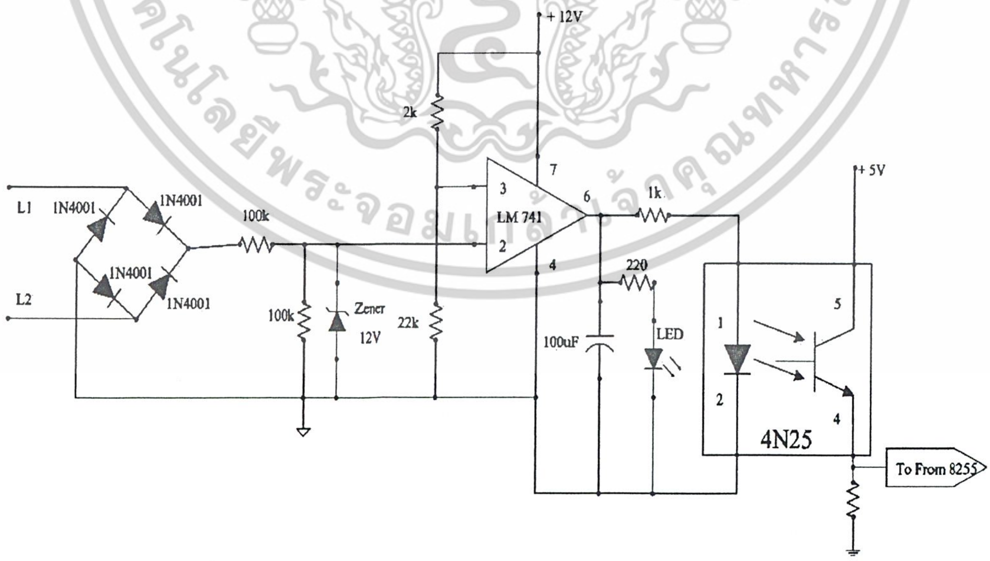
1.2 โหมดการควบคุมเครื่องใช้ไฟฟ้า คือ เมื่อเราอยู่นอกบ้านและจะต้องการสั่งงานให้อุปกรณ์ไฟฟ้าภายในบ้านเปิดหรือปิด เราก็แค่โทรเข้ามาและเลือกโหมดการทำงานนี้ โดยจะใช้ตัวเลขต่าง ๆ บนหน้าปัดโทรศัพท์เป็นตัวสั่งการเท่านั้น

2. กรณีของการโทรออกไป

เมื่อผู้ใช้ทำการยกหูโทรศัพท์เพื่อจะทำการโทรออก ไมโครคอนโทรลเลอร์ก็จะทำการตรวจเช็คสัญญาณหมุนหมายเลขว่าอุปกรณ์ปลายทางพร้อมหรือไม่ จากนั้นก็จะทำการรอรับหมายเลขปลายทางเมื่อชุมสายต่อเลขหมายปลายทางให้แล้ว วงจรก็จะทำการตรวจเช็คว่าที่ปลายทางมีคนมารับโทรศัพท์แล้วหรือไม่ เมื่อมีคนมารับเครื่องก็จะทำการบันทึกหมายเลขที่กดไปยังปลายทาง และค่าเวลาจริง แล้วทำการบันทึกข้อมูลเหล่านี้ไว้ในหน่วยความจำของระบบ เมื่อการสนทนาเสร็จสิ้นลงเครื่องก็จะทำการบันทึกเวลาจริงที่วางสาย และ โปรแกรมก็จะทำการคำนวณคิดค่าโทรศัพท์ เพื่อจะแสดงผลให้กับผู้ใช้โทรศัพท์ให้รู้ทันที

3.2 ส่วนของวงจรตรวจจับสัญญาณโทรศัพท์

3.2.1 วงจรตรวจสอบการยกหู (Hook Switch Detector)



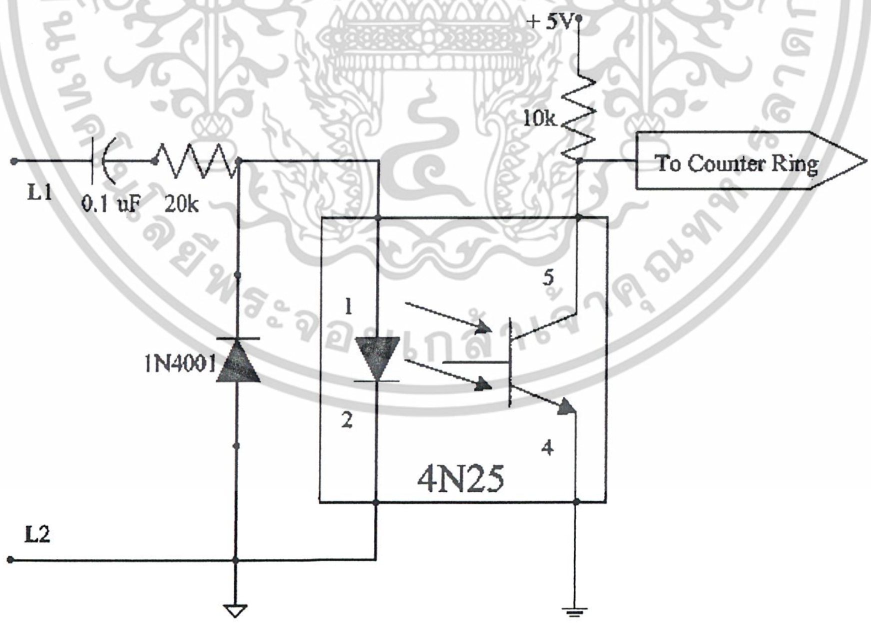
รูปที่ 3.2 วงจรตรวจสอบการยกหู

หลักการทำงานของวงจร คือ ใช้การเปรียบเทียบของแรงดันระหว่างการยกหูและการวางหู ซึ่งในขณะที่วางหูอยู่นั้นจะมีแรงดันตกคร่อมคู่สายโทรศัพท์ประมาณ 48 โวลต์ และในขณะที่ทำการวางหูจะทำให้แรงดันตกลงเหลือประมาณ 6-10 โวลต์ ซึ่งก็แล้วแต่ค่าความต้านทานของเครื่องโทรศัพท์แต่ละเครื่อง เมื่อเป็นเช่นนี้เราจึงได้ใช้ออปแอมป์ ซึ่งตั้งระดับแรงดันอ้างอิงที่ขา 2 ไว้ที่ 11 โวลต์ มาทำงานในส่วนของการเปรียบระดับแรงดัน และใช้ออปโต้ไอโซเลเตอร์มาเป็นตัวเปลี่ยนระดับของแรงดันจาก 12 โวลต์มาเป็น 5 โวลต์ เพื่อที่จะให้สามารถใช้ร่วมกับแรงดันในส่วนของไมโครคอนโทรลเลอร์ได้

พิจารณาในขณะที่ทำการวางหู แรงดันไฟ 48 โวลต์จะผ่านวงจรบริดจ์เร็คติไฟเออร์ซึ่งจะให้ขั้วของแรงดันแน่นอน จากนั้นก็จะตกคร่อมตัวต้านทานและซีเนอร์ไดโอด 12 โวลต์ ดังนั้นแรงดันที่ตกคร่อมขา 3 ของออปแอมป์ จึงมีค่าสูงสุดได้แค่ 12 โวลต์ เนื่องจากถูกซีเนอร์จำกัดแรงดันไว้เมื่อระดับแรงดันที่เข้ามามีค่ามากกว่าระดับแรงดันอ้างอิง ทำให้เอาท์พุตของออปแอมป์มีค่าประมาณ 12 โวลต์ เป็นผลทำให้เกิดการนำกระแสของแอลอีดีในออปโต้ไอโซเลเตอร์ และไปกระตุ้นให้ทรานซิสเตอร์นำกระแสตามไปด้วย เกิดแรงดันตกคร่อมตัวต้านทาน 100 กิโลโอห์มที่เอาท์พุตเป็น 5 โวลต์ ซึ่งแทนด้วยระดับลอจิก “1”

พิจารณาในขณะที่ทำการยกหู แรงดันจะตกลงเหลือประมาณ 6-9 โวลต์ ตกคร่อมที่ขา3 ของออปแอมป์ และเมื่อเปรียบเทียบกับระดับอ้างอิงแล้วมีค่าน้อยกว่า ทำให้เอาท์พุตของออปแอมป์มีค่าประมาณ 0 โวลต์ เป็นผลทำให้แอลอีดีและทรานซิสเตอร์ในออปโต้ไอโซลเตอร์ไม่นำกระแสแรงดันที่เอาท์จึกตกลงเป็น 0 โวลต์ ซึ่งแทนด้วยระดับลอจิก “0”

3.2.2 วงจรตรวจสอบสัญญาณเรียก (Ringing Detector)



รูปที่ 3.3 วงจรตรวจสอบสัญญาณเรียก

เนื่องจากสัญญาณกระดิ่งเป็นกระแสสลับ และมีขนาดค่อนข้างสูง เราจึงนำสัญญาณนี้ไปใช้โดยตรงไม่ได้จึงต้องทำการแปลงสัญญาณให้มีขนาดต่ำลง เพื่อให้วงจรอื่นปลอดภัย

สัญญาณกระดิ่งจะถูกลดทอนลงเมื่อผ่านตัวเก็บประจุ และตัวต้านทานที่ต่อตอนต้นของวงจรจากนั้นจะผ่าไดโอด เพื่อใช้เฉพาะสัญญาณด้านบวกเท่านั้นผ่านได้ และสัญญาณที่ผ่านมาจากไดโอดจะไปเข้าไอซี 4N25 ซึ่งเป็น Opto Copler ทำหน้าที่แยกกราวด์ของสัญญาณกระดิ่งออกจากกราวด์ของระบบ เอาต์พุตที่ได้จาก 4N25 จะเป็นพัลซ์ที่มีความถี่เท่ากับสัญญาณกระดิ่ง เพื่อให้มีความเที่ยงตรงมากขึ้น จึงนำไปผ่านวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ ในที่นี้ใช้ค่าไอซี 74LS123 เพื่อสร้างพัลซ์ขึ้นมาใหม่ โดยความกว้างพัลซ์ สามาหาได้จากสูตรดังนี้

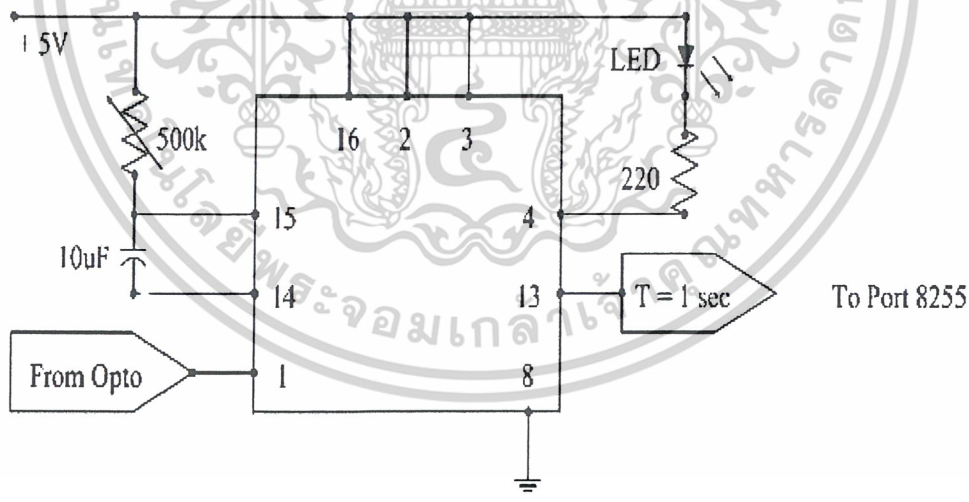
สูตร $\tau = 1.1$ RC โดยจะทำการกำหนดค่าความจุของคาปาซิเตอร์ไว้ก่อนที่ค่า 10 ไมโครฟารัด ดังนั้นเราจึงหาค่าของตัวต้านได้ คือ

$$R = \tau / 1.1C$$

$$= 1/1.1*10 \text{ ไมโครฟาหรัด}$$

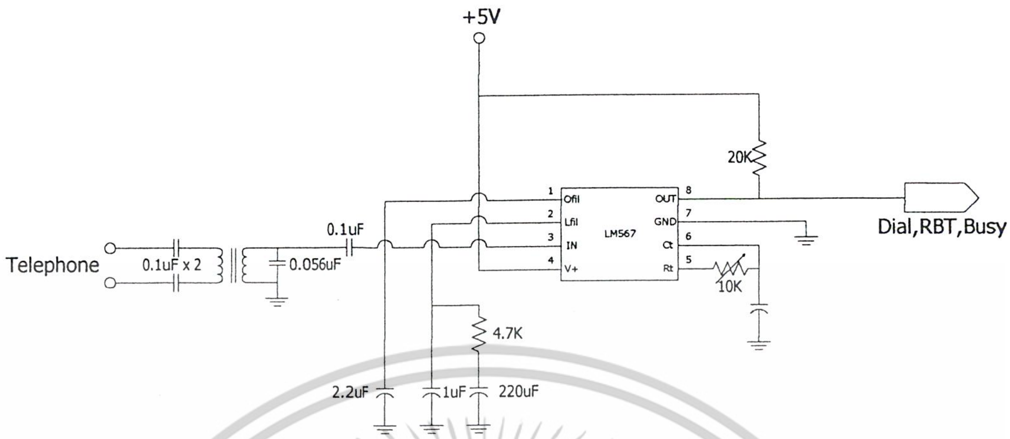
$$= 90.9 \text{ กิโลโอห์ม}$$

เอาท์พุตที่ได้ออกที่ขา 13 ของไอซี 74123 ซึ่งมีค่า = 1 วินาที จะถูกป้อนให้กับวงจรนับซึ่งจะให้เอาท์พุต Q0 – Q9 เรียงลำดับออกมาเมื่อมีสัญญาณนาฬิกามากระตุ้นที่ขา CLK ของไอซี 74123 การกำหนดว่าต้องการให้เสียงกระดิ่งดังกี่ครั้งจึงจะตัดเข้าสู่การทำงานในโหมดของการตอบรับอัตโนมัติ ทำได้โดยเลือกจากดิพสวิตซ์ก็จะสามารถกำหนด ได้แล้ว



รูปที่ 3.4 วงจรตรวนับสัญญาณเรียก

3.2.3 วงจรตรวจสอบสัญญาณหมุนหมายเลข (DIAL), สัญญาณเรียกกลับ (RBT), สัญญาณไม่ว่าง (BUSY)



รูปที่ 3.5 วงจรตรวจจับสัญญาณ Dialling, RBT, BUSY

หลักการทำงาน คือ จะเห็นได้ว่าสัญญาณหมุนหมายเลข, สัญญาณเรียกกลับ, สัญญาณไม่ว่างนั้น จะมีลักษณะเป็นคลื่นรูปไซน์ และความถี่ประมาณ 400 เฮิรตซ์เท่านั้น จะแตกต่างที่ช่วงของเวลาการเกิดของสัญญาณแต่ละสัญญาณเท่านั้น จึงใช้ไอซีเบอร์ LM567 ซึ่งเป็นไอซีโทนดีโค้ดเดอร์ (Tone Decoder) เป็นตัวตรวจจับสัญญาณความถี่ 400 เฮิรตซ์ ถ้าความถี่ที่เข้ามาไม่ตรงกับความถี่ที่กำหนดไว้ เอาท์พุตก็จะมีค่าเป็นลอจิก "1" การกำหนดค่าของความถี่ที่ต้องการดีเทกออกมานั้นสามารถได้จากสูตร

$$f = 1/1.1\ RC$$

ในวงจรนี้ C_8 และ V_{R1} จะเป็นตัวกำหนดความถี่ที่ต้องการออกมา โดยในการคำนวณเราจะกำหนดค่า C ไว้ที่ 0.33 ไมโครฟารัด ความถี่ที่ต้องการ 400 เฮิรตซ์ เราก็จะได้ค่า R ออกมา คือ

$$R = 1/1.1 * f * C$$

$$= 1/1.1 * 400 * 0.33 \text{ ไมโครฟารัด}$$

$$= 6.88 \text{ กิโลโอห์ม}$$

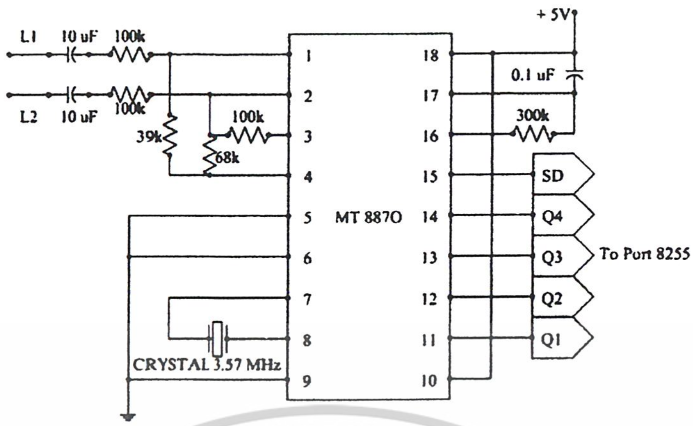
สัญญาณที่เอาท์พุตจะมีลักษณะเป็นพัลส์สแควร์เวฟ เกิดขึ้นตามการสัญญาณที่เข้ามากล่าวคือ

1. ถ้าเป็นสัญญาณหมุนหมายเลข จะมีลักษณะการเกิดของการสัญญาความถี่ 400 เฮิรตซ์ที่ต่อเนื่องและยาว เอาท์พุตที่ออกมาจะมีลักษณะเป็นลอจิก "0" ยาวติดกันไปตลอด

2. ถ้าเป็นสัญญาณไม่ว่าง จะมีลักษณะการเกิดของสัญญาณความถี่ 400 เฮิรตซ์ 0.5 วินาที และเวลาหยุด 0.5 วินาที สลับกันไป เอาท์พุตที่ได้จะมีลักษณะเป็นลอจิก "0" และลอจิก "1" สลับกันตลอด

3. ถ้าเป็นสัญญาณเรียกกลับ จะมีลักษณะการเกิดของสัญญาณความถี่ 400 เฮิรตซ์ 1 วินาที และเวลาหยุด 4 วินาที สลับกันไปเรื่อย ๆ เอาท์พุตที่ได้จะมีลักษณะเป็นลอจิก "0" 1 วินาที และลอจิก "1" 4 วินาที

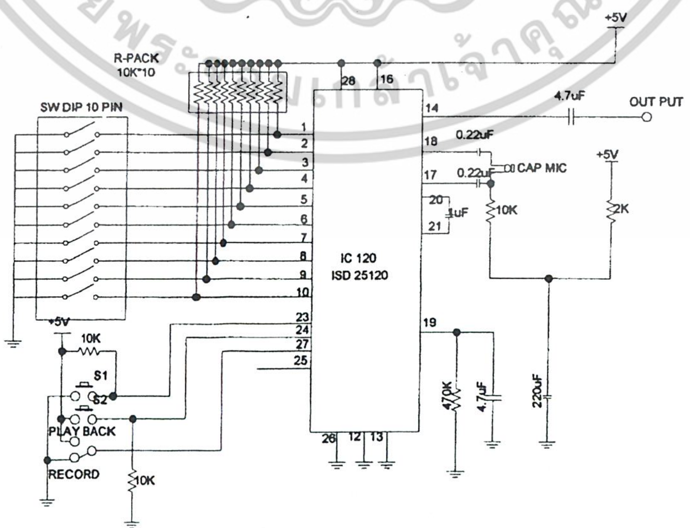
3.2.4 วงจรถอดรหัสหมายเลขดีทีเอ็มเอฟ (DTMF Decoder)



รูปที่ 3.6 วงจรถอดรหัสหมายเลขดีทีเอ็มเอฟ

จะใช้ไอซีสำเร็จรูปเบอร์ MT8870 ซึ่งจะทำหน้าที่ในการถอดรหัสความถี่ดีทีเอ็มเอฟออกมาเป็นรหัสไบนารี่ 4 บิต ภายใน MT8870 ซึ่งจะประกอบไปด้วยภาคกรองความถี่ ภาคถอดรหัสภาคตรวจสอบสัญญาณ ภาคขยายสัญญาณความแตกต่าง และภาคกำเนิดความถี่ โดยจะใช้คริสตอล 3.579 เมกกะเฮิรตซ์เป็นฐานเวลา จากวงจรจะใช้แมชชิ่ง 600 โอห์มมาเป็นตัวคัปปลิ้งสัญญาณจากคู่สายโทรศัพท์เข้ามายังวงจรเฉพาะสัญญาณไฟสลับเท่านั้น และยังทำหน้าที่ในการแยกกราวด์ของวงจรอีกด้วยเพราะในคู่สายโทรศัพท์จะมีสัญญาณรบกวนต่าง ๆ มากมาย ซึ่งอาจจะมีผลต่อการโค้ดสัญญาณ ได้หลังจากที่ที่ทำการกดหมายเลขและไอซีทำการดีโค้ดรหัสแล้ว ก็จะส่งรหัสไบนารี่มาที่เอาท์พุต Q1 – Q4 จากนั้นไอซีก็จะทำการแลตช์ค่าสัญญาณนั้นเอาไว้จนกว่าจะได้รับสัญญาณใหม่เข้ามา แล้วจึงค่อยเปลี่ยนเป็นรหัสตัวใหม่

3.3 วงจรการใช้งานของ ISD2590



รูปที่ 3.7 แสดงวงจรการใช้งานที่ต่อร่วมกับอุปกรณ์ภายนอกของ ISD2590

3.3.1 ลำดับขั้นตอนในการบันทึกและเล่นกลับ แสดงดังตารางที่ 3.1

ตารางที่ 3.1 การใช้งานและควบคุมการทำงานของ ISD2590

ขั้นตอบการควบคุม	หน้าที่	การปฏิบัติ
1	จ่ายไฟเลี้ยงและเลือกโหมดการบันทึกหรือเล่นกลับ	(1) PD = “0” (2) เลือกบันทึก/เล่นกลับ
2	เซตจุดเริ่มต้นของการบันทึกหรือเล่นกลับ	เซตขาแอดเดรส A0 – A9
3A	เริ่มต้นการเล่นกลับ	P/R = “1” ,CE = พัลส์ “0”
3B	เริ่มต้นการบันทึก	P/R = “0” ,CE = “0”
4A	สิ้นสุดการเล่นกลับ	อัตโนมัติ
4B	สิ้นสุดการบันทึก	PD หรือ CE = “1”

3.3.2 การกำหนดตำแหน่งแอดเดรสเพื่อใช้ในการบันทึกและเล่นกลับ

ISD2590 สามารถอ้างแอดเดรสเพื่อกำหนดช่วงในการบันทึกและเล่นกลับได้ โดยการควบคุมที่ขา A9 ถึง A0 โดยค่าของ A9 ถึง A0 จะถูกควบคุมโดยการกำหนดสภาวะให้เป็น “1” หรือ “0” เพื่อเป็นการกำหนดตำแหน่งเริ่มต้นของข้อความที่บันทึกหรือเล่นกลับ ดูจากตารางที่ 3.2 ซึ่งแสดงตำแหน่งของแอดเดรสต่าง ๆ ดังต่อไปนี้

ตารางที่ 3.2 การกำหนดแอดเดรสของ ISD2590

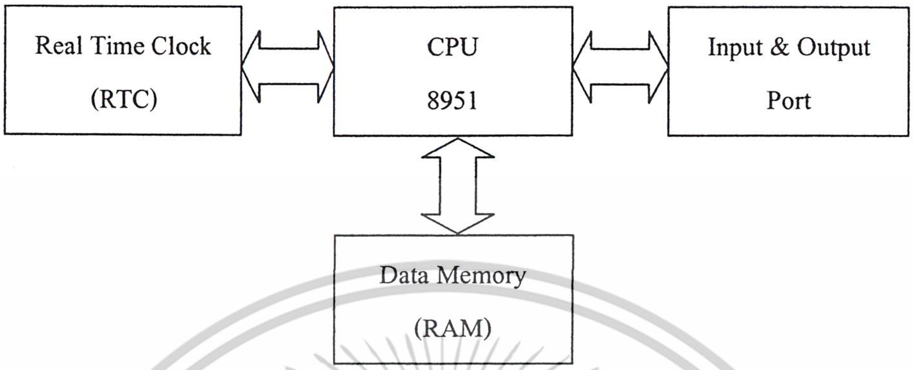
เลขฐานสิบ	เลขฐานสอง										วินาที
	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	1	1	1	1	0	0	0	18
240	0	0	1	1	1	1	0	0	0	0	36
360	0	1	0	1	1	0	1	0	0	0	54
480	0	1	1	1	1	0	0	0	0	0	72

จากตารางที่ 3.2 ถ้ากำหนดให้แอดเดรส A9 ถึง A0 มีสภาวะเป็น “0” ทั้งหมด การบันทึกการเล่นกลับจะเริ่มต้นในวินาทีที่ 0 ในการออกแบบจะใช้ไอซี ISD2590 ทั้งหมด 4 ตัว โดยให้ตัวแรกทำหน้าที่ในการตอบรับ และอีก 3 ตัวที่เหลือทำหน้าที่ในการบันทึกผู้ที่โทรเข้ามาในขณะที่ไม่มีผู้ใดรับสาย โดยทั้ง 3 ตัวที่ใช้ในการบันทึกนี้จะให้แต่ละตัวสามารถบันทึกได้ 5 ข้อความ ดังนั้นจึงทำให้ข้อความหนึ่งข้อความมีเวลาในการบันทึก 18 วินาที โดยกำหนดค่าแอดเดรสดังตารางที่ 3.2

3.4 ส่วนระบบไมโครคอนโทรลเลอร์

ในโครงงานนี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ 8031 ทั้งนี้เพื่อที่สะดวกในการทดลองและใช้งานจริงได้ ซึ่งส่วนประกอบของส่วนไมโครคอนโทรลเลอร์นี้ จะมีส่วนประกอบดังบล็อกไดอะแกรมนี้

3.4.1 บล็อกไดอะแกรม



รูปที่ 3.8 แสดงบล็อกไดอะแกรมระบบไมโครคอนโทรลเลอร์

จากบล็อกไดอะแกรม จะประกอบไปด้วย

- ส่วนดาต้าเมมโมรี ไว้สำหรับเป็นหน่วยความจำที่ใช้ในการเก็บข้อมูลของการใช้โทรศัพท์ที่เราต้องการ
- ส่วน RTC เป็นส่วนของการสร้างฐานเวลาเพื่อใช้ในการคำนวณเวลาในการใช้โทรศัพท์และเป็นส่วนอ้างอิงเวลาของระบบ
- ส่วนที่เชื่อมต่อกัน ระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ส่วนอื่น ๆ ใช้เป็นส่วนอินพุต/เอาท์พุตพอร์ต ซึ่งจะใช้ไอซีเบอร์ 8255 เป็นตัวเชื่อมต่อ ในที่นี้จะใช้เป็นจำนวน 4 ตัว ซึ่งแบ่งแอดเดรสในการใช้งานดังนี้

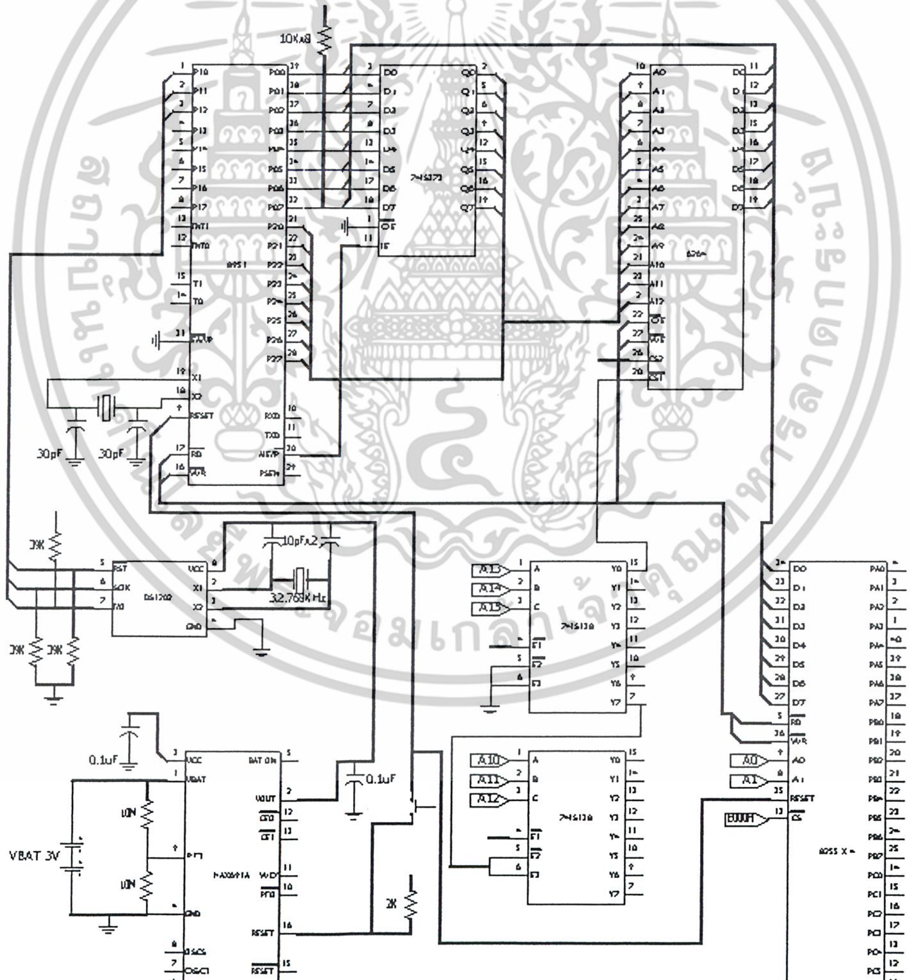
ตารางที่ 3.3 แสดงตำแหน่งเมมโมรีแม็ป

74138 ตัวที่ 1 ตำแหน่งที่ 0000H – FFFFH

ตำแหน่ง ADDRESS	อุปกรณ์ที่ติดต่อ
000H – 1FFFH	RAM # 1
2000H – 3FFFH	-
4000H – 5FFFH	-
6000H – 7FFFH	-
8000H – 9FFFH	-
A000H – BFFFH	-
C000H – DFFFH	-
E000H – FFFFH	INPUT & OUTPUT PORT

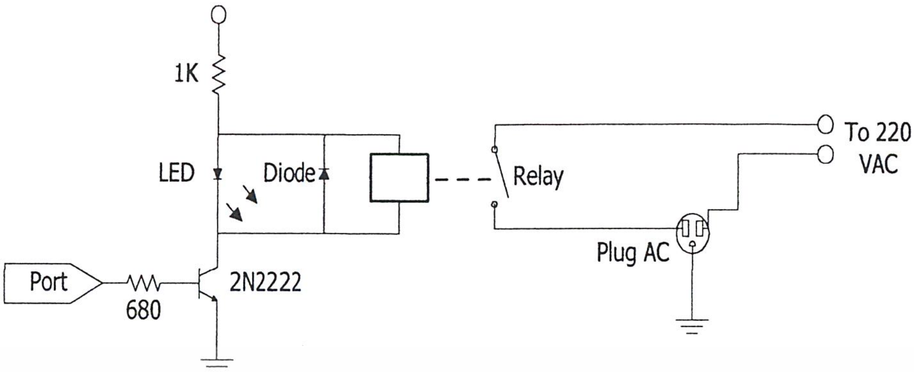
ตารางที่ 3.4 แสดงตำแหน่งของอินพุต – เอาท์พุตพอร์ต
74138 ตัวที่ 2 ตำแหน่งที่ E000H – FFFFH

ตำแหน่ง ADDRESS	อุปกรณ์ที่ติดต่อ
E000H – E3FFH	8255#1...........แอลซีดีโมดูลและคีย์บอร์ด
E400H – E7FFH	8255#2...........ดีเทกสัญญาณโทรศัพท์
E800H – EBFFH	8255#3...........เครื่องใช้ไฟฟ้า
EC00H – EFFFH	8255#3...........คีย์บอร์ดและเครื่องใช้ไฟฟ้า
F000H – F3FFH	-
F400H – F700H	-
F800H – FB00H	-
FC00H - FFFFH	-



รูปที่ 3.9 วงจรไมโครคอนโทรลเลอร์

3.5 วงจรควบคุมปิด – เปิดเครื่องใช้ไฟฟ้า

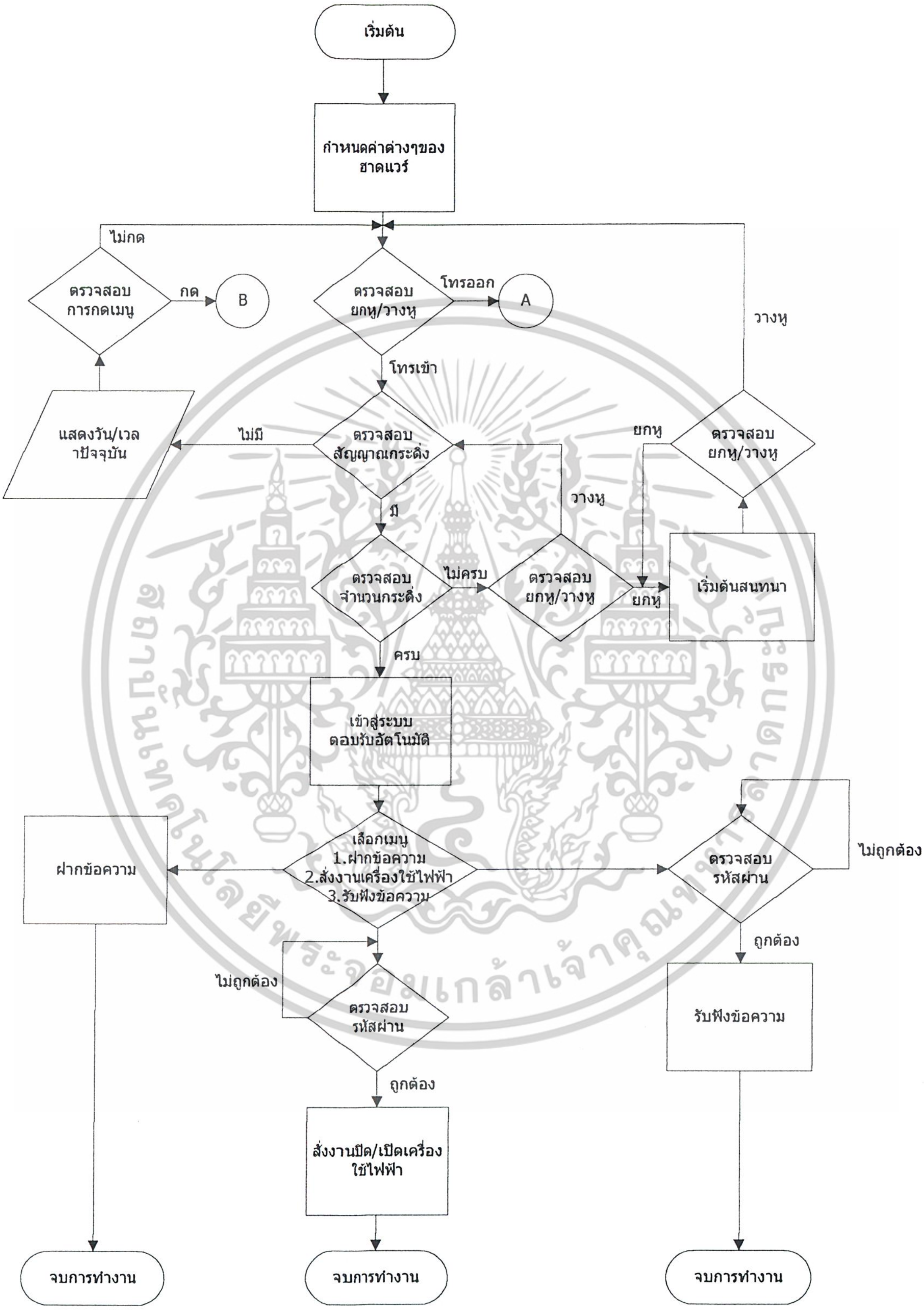


รูปที่ 3.10 แสดงวงจรควบคุมปิด – เปิดเครื่องใช้ไฟฟ้า 1 แชนแนล

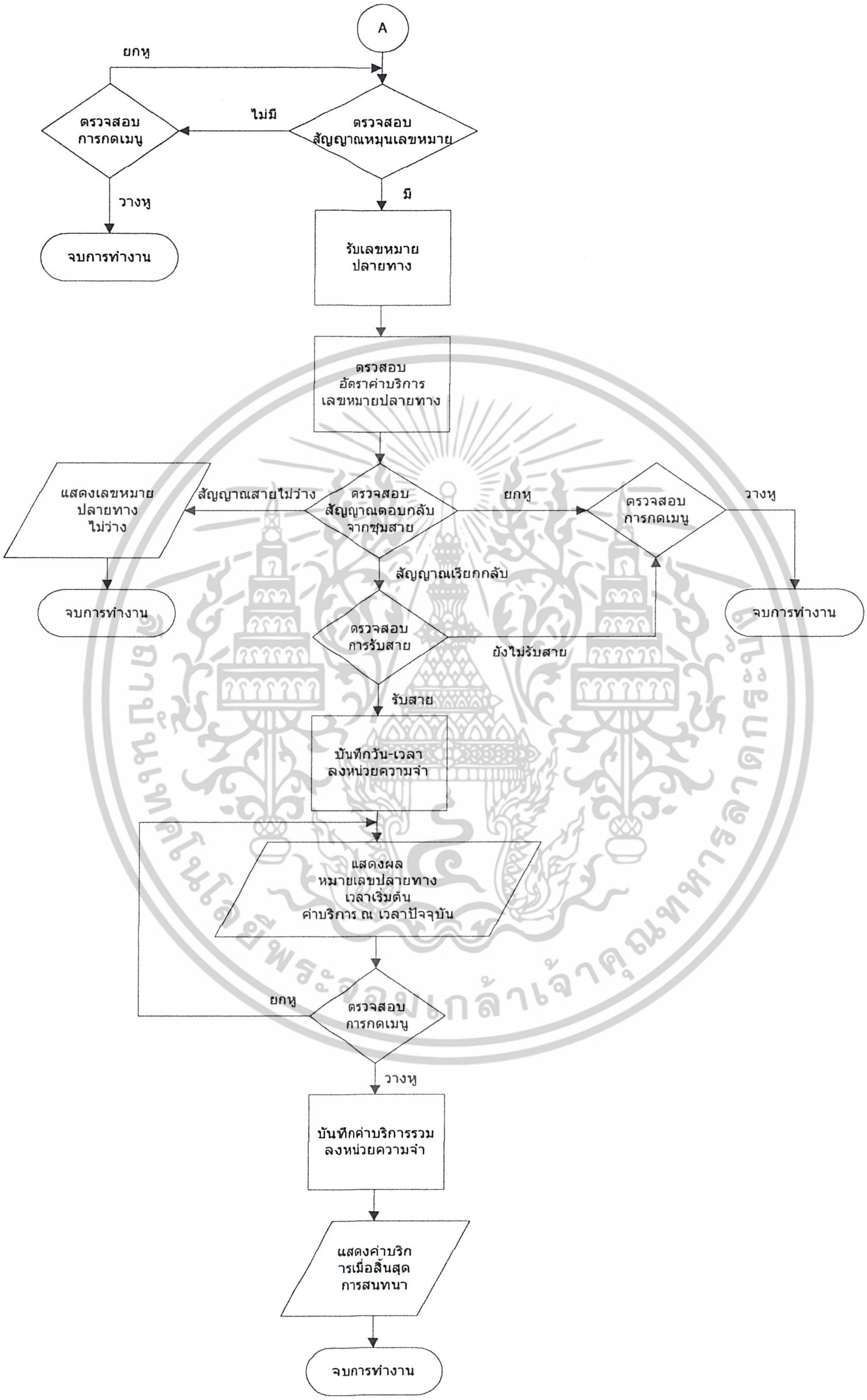
หลักการทำงานของวงจร คือ กำหนดการทำงานของทรานซิสเตอร์ให้ทำหน้าที่เป็นสวิตช์เพื่อควบคุมการปิด – เปิดของรีเลย์ โดยเมื่อป้อนไฟ +5 โวลต์เข้ามาที่ขาเบสของทรานซิสเตอร์ทำให้ทรานซิสเตอร์ทำงานในย่านอิ่มตัว ส่งผลให้หน้าสัมผัสของรีเลย์แตะกัน ซึ่งเมื่อต่อกับเครื่องใช้ไฟฟ้า ทำให้เครื่องใช้ไฟฟ้าทำงานได้ เมื่อป้อนไฟ 0 โวลต์เข้ามาที่ขาเบสของทรานซิสเตอร์ ทำให้ทรานซิสเตอร์ทำงานในย่านไม่นำกระแส ส่งผลให้หน้าสัมผัสของรีเลย์ไม่แตะกัน ทำให้เครื่องใช้ไฟฟ้าที่ต่ออยู่หยุดทำงานลงได้ โดยในการออกแบบจะใช้วงจรดังรูปที่ 3.13 ทั้งหมด 8 วงจร เพื่อควบคุมเครื่องใช้ไฟฟ้าทั้งหมด 8 แชนแนล โดยที่ขาเบสของทรานซิสเตอร์แต่ละตัวจะต่อเข้ากับพอร์ตของ 8255 จำนวน 1 พอร์ต

3.6 Flow Chart

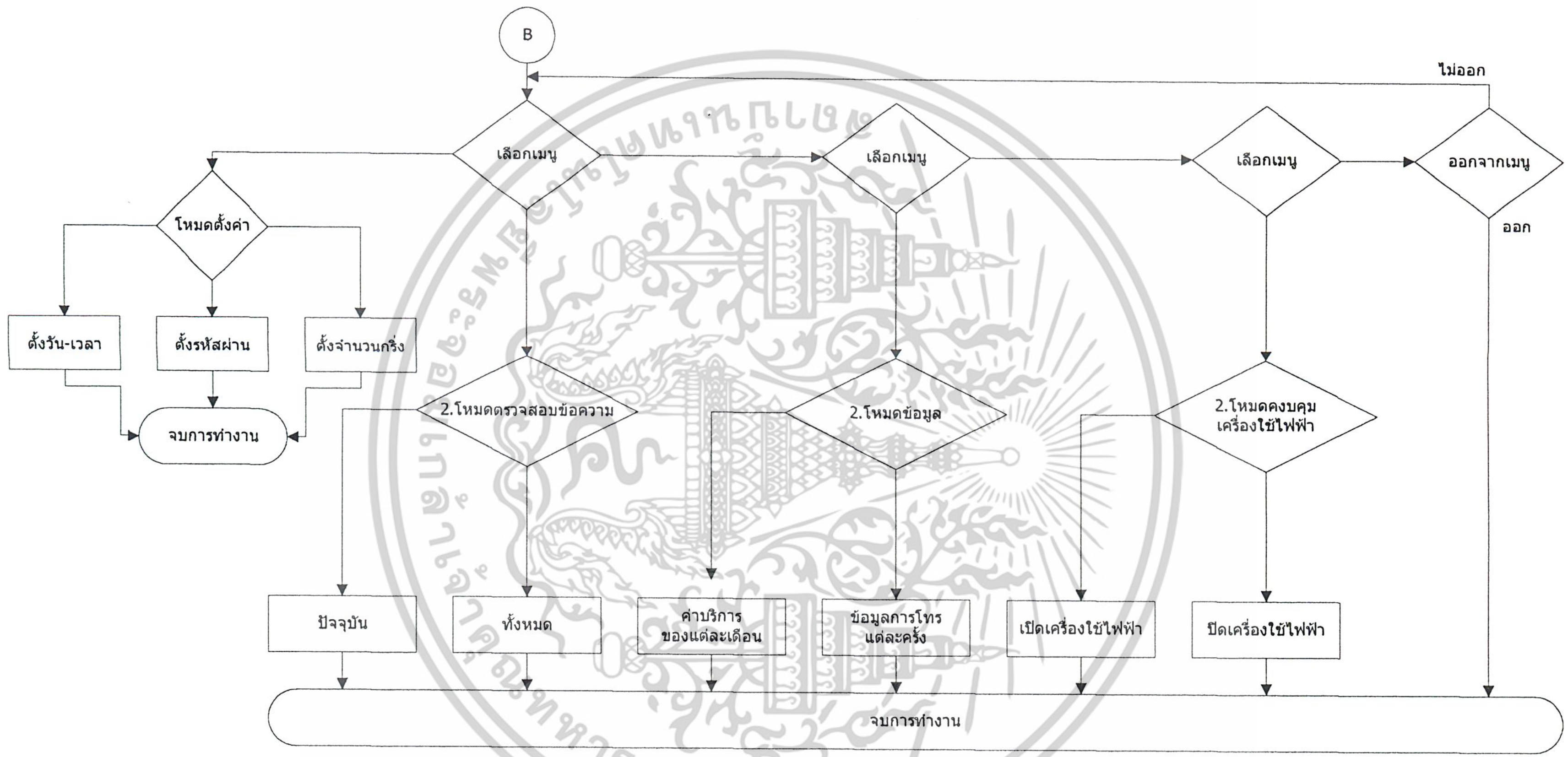
3.6.1 แสดงการทำงานเมื่อมีการโทรเข้า



3.6.2 แสดงการทำงานเมื่อมีการโทรออก



3.6.3 แผนผังการตั้งค่าแสดงการใช้งานในโหมดต่างๆ

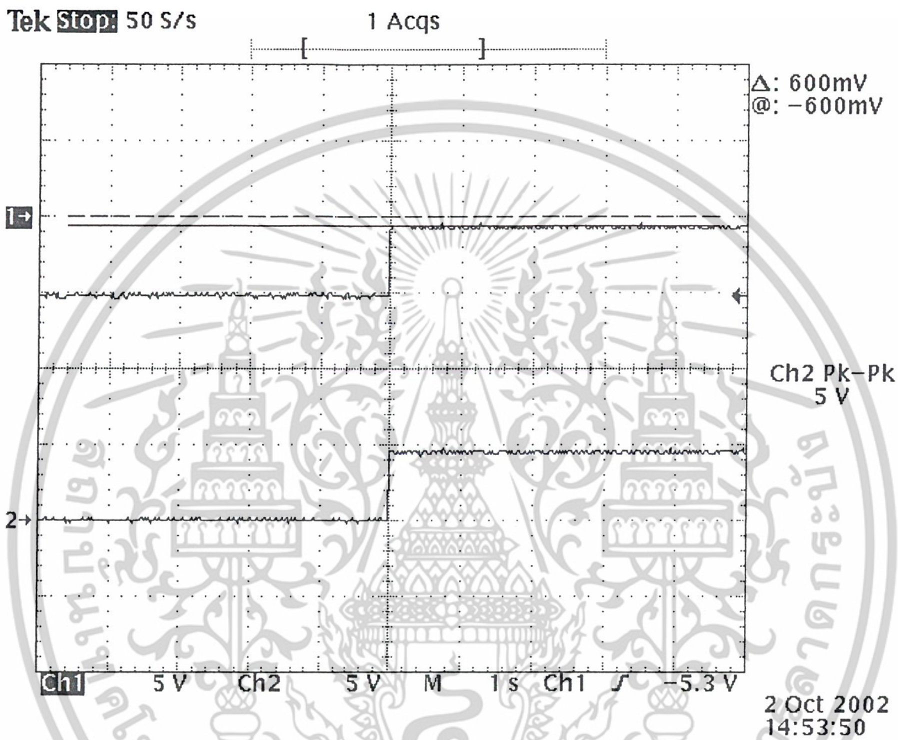


บทที่ 4
การทดลองและผลการทดลอง

4.1 วงจรตรวจสอบสัญญาณโทรศัพท์

4.1.1 วงจรตรวจจับการยกหู จะได้ผลการทดลองนี้

ผลการทดลอง



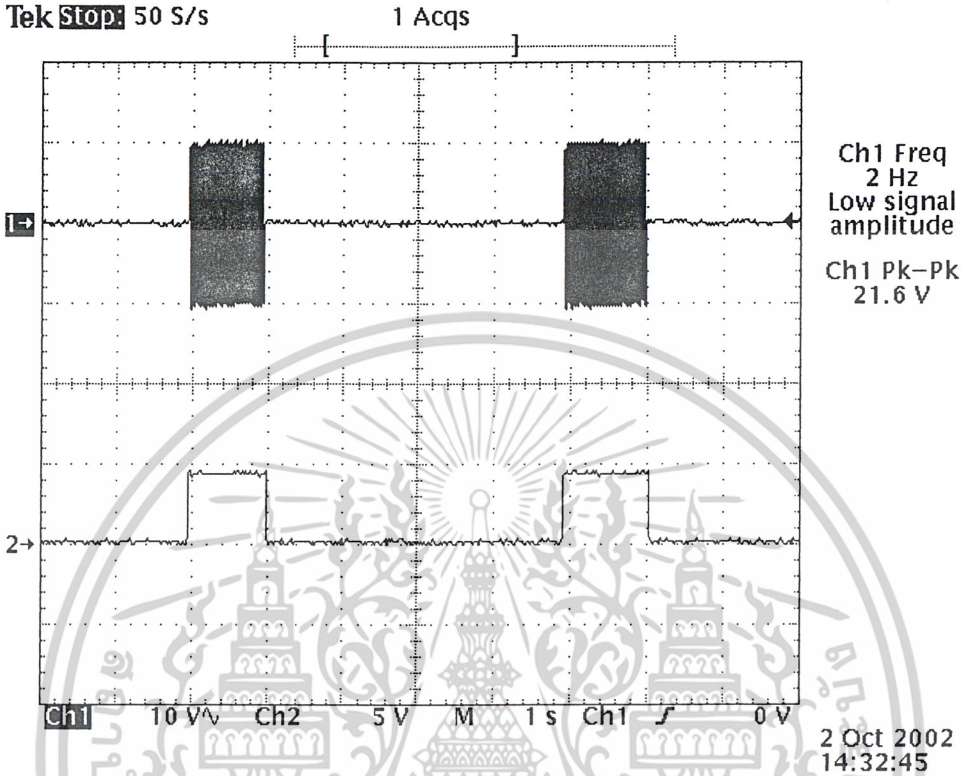
รูปที่ 4.1 แสดงรูปสัญญาณที่ได้จากวงจรตรวจสอบการยกหู

จากรูปที่ 4.1 รูปบนเป็นสัญญาณที่วัดคร่อมคู่สายโทรศัพท์ ซึ่งในขณะที่ทำการวางหูระดับแรงดันไฟที่ตกคร่อมคู่สายจะมีค่าประมาณ 48 โวลต์ และขณะที่ยกหู แรงดันไฟจะตกลงเหลือประมาณ 6-10 โวลต์

รูปสัญญาณด้านล่าง จะเป็นเอาท์พุตที่ได้จากวงจรตรวจจับการวางหู จะเห็นและเปรียบเทียบได้ว่าขณะที่คู่สายวางหูอยู่ แรงดันไฟ 48 โวลต์จะถูกเปลี่ยนให้เป็นแรงดันไฟ 5 โวลต์แทนระดับลอจิก "1" เพื่อให้สามารถใช้งานร่วมกับสัญญาณในระบบไมโครคอนโทรลเลอร์ได้ ส่วนในขณะการยกหู แรงดันก็จะตกลงเป็น 0 โวลต์แทนด้วยระดับลอจิก "0"

4.1.2 วงจรตรวจสอบสัญญาณเรียก จะได้ผลการทดลอง ดังนี้

ผลการทดลอง

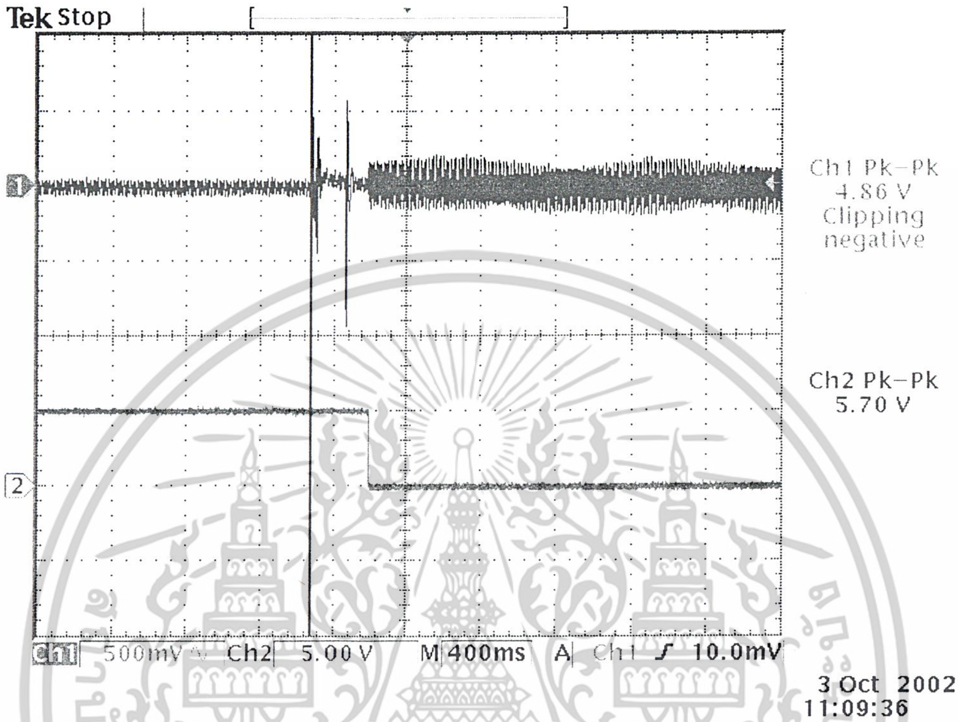


รูปที่ 4.2 แสดงรูปสัญญาณที่ได้จากวงจรตรวจสอบสัญญาณเรียก

จากรูปที่ 4.2 จะเห็นได้ว่าในขณะที่มีสัญญาณกระดิ่งเรียกเข้ามา เป็นจังหวะมีสัญญาณ 1 วินาที และหยุด 4 วินาที เอาท์พุตที่ได้จากวงจรตรวจจับสัญญาณเรียก ก็จะเปลี่ยนแปลงตามไปด้วย

4.1.3 วงจรตรวจสอบสัญญาณหมุนหมายเลข, สัญญาณเรียกกลับ, สัญญาไม่ว่าง จะได้ผลการทดลองดังนี้

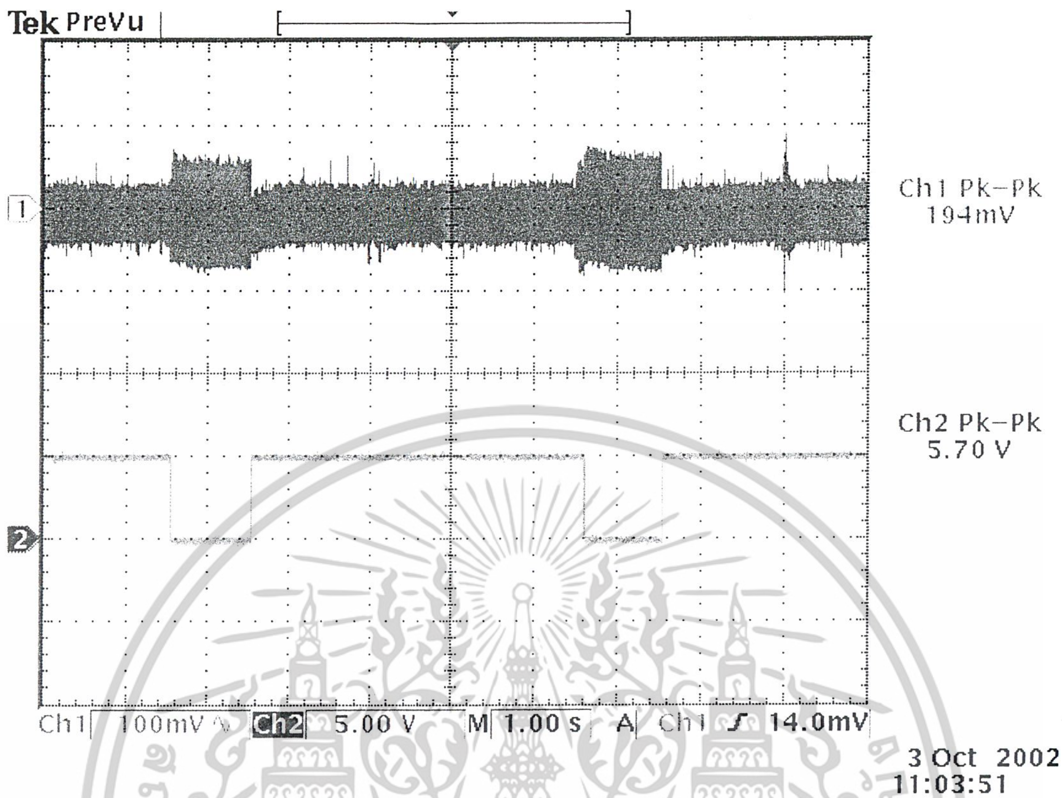
ผลการทดลองวงจรตรวจจับสัญญาณหมุนหมายเลข



รูปที่ 4.3 แสดงสัญญาณที่ได้จากวงจรตรวจสอบสัญญาณหมุนหมายเลข

จากรูปที่ 4.3 เป็นสัญญาณหมุนหมายเลข ซึ่งจะเกิดขึ้นในขณะที่ยกหูโทรศัพท์และทาง ชุมสายก็จะส่งสัญญาณนี้มาให้เพื่อเป็นการบอกว่าอุปกรณ์ปลายทางพร้อมแล้ว วงจรตรวจจับก็จะ ทำการตรวจจับสัญญาณนี้ โดยถ้ามีสัญญาณนี้เกิดขึ้นเอาท์พุตที่ออกมาก็จะมีค่าเป็น 0 โวลต์แทน ด้วยระดับลอจิก “0” และถ้าไม่มีสัญญาณนี้เอาท์พุตก็จะมีค่าเป็น 5 โวลต์ แทนด้วยระดับลอจิก “1” ซึ่งช่วงเวลาในการเกิดนั้นจะยาวไปตลอดจนกว่าจะเริ่มป้อนเลขหมายเข้ามา

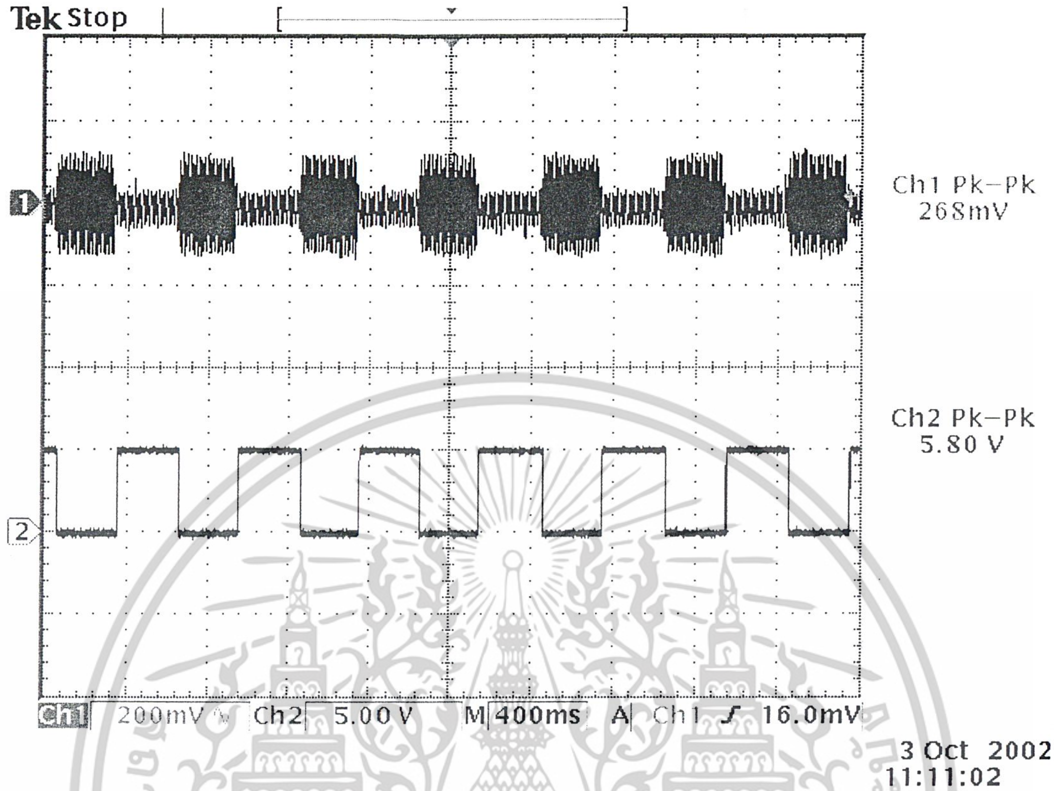
ผลการทดลองวงจรตรวจจับสัญญาณเรียกกลับ



รูปที่ 4.4 แสดงรูปสัญญาณที่ได้จากวงจรตรวจสอบสัญญาณเรียกกลับ

จากรูปที่ 4.4 จะคล้ายกับวงจรตรวจจับสัญญาณหมุนหมายเลข คือ จะมีค่าเป็นลอจิก "0" ก็ต่อเมื่อมีสัญญาณเรียกกลับ 400 H_z เข้ามา แต่จะแตกต่างกันที่ช่วงของระยะเวลาการเกิดของ สัญญาณเท่านั้น คือ สัญญาณเรียกกลับจะเกิด 1 วินาที และหยุด 4 วินาที

ผลการทดลองวงจรตรวจจับสัญญาณไม่ว่าง

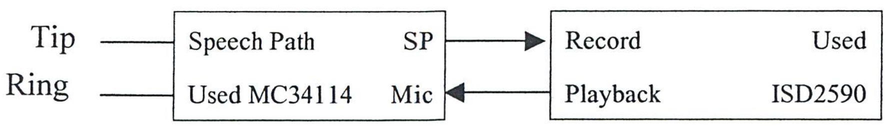


รูปที่ 4.5 แสดงรูปสัญญาณที่ได้จากวงจรตรวจสอบสัญญาณไม่ว่าง

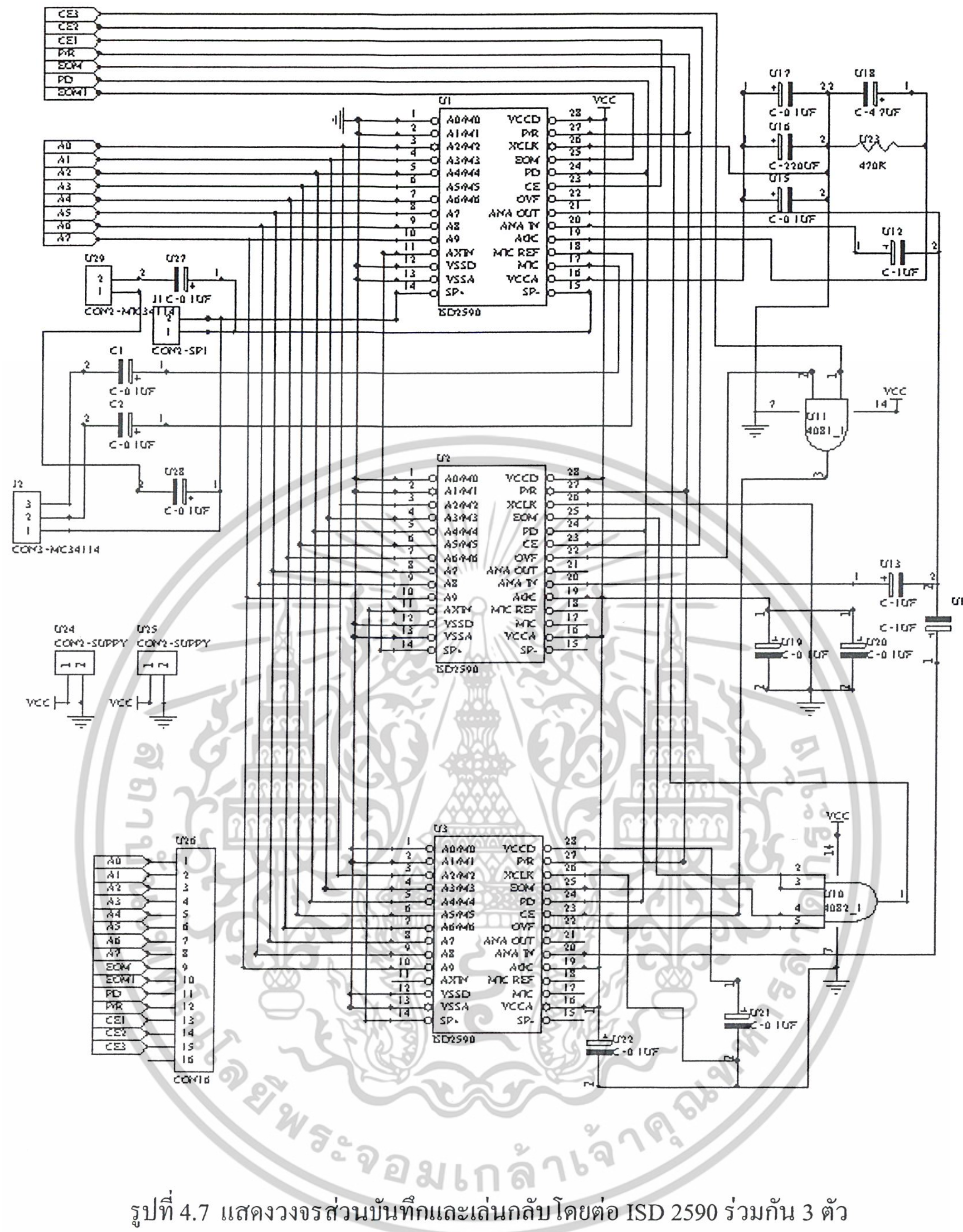
จากรูปที่ 4.5 ก็จะคล้ายกับ 2 วงจรที่แล้ว คือจะให้เอาท์พุตมีค่าเป็น “0” ก็ต่อเมื่อมีความถี่ 400 H_Z เข้ามา ระยะเวลาในการเกิดสัญญาณ 0.5 วินาที และหยุด 0.5 วินาที

4.2 ส่วนวงจรของการบันทึกและเล่นกลับ

วงจรส่วนนี้จะใช้วงจรส่วนควบคุมเสียงพูดโดยใช้ไอซีเบอร์ MC3441114 และวงจรส่วนบันทึกและเล่นกลับซึ่งใช้ไอซี ISD2590 ต่อร่วมกัน โดยมีลักษณะการต่อตามบล็อกไดอะแกรม ดังรูปที่ 4.6



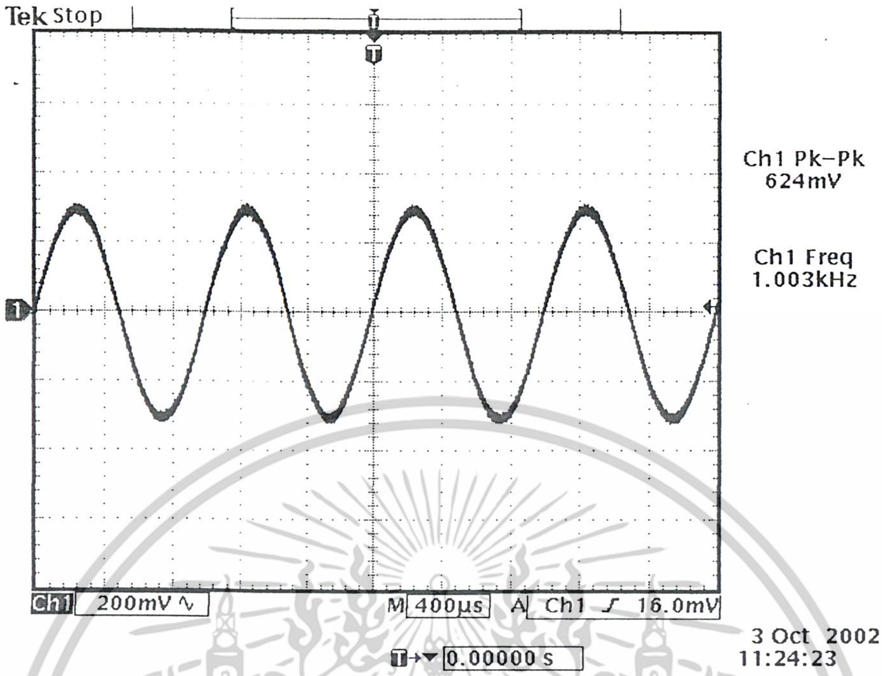
รูปที่ 4.6 บล็อกไดอะแกรมส่วนวงจรตอบรับและบันทึกข้อความ



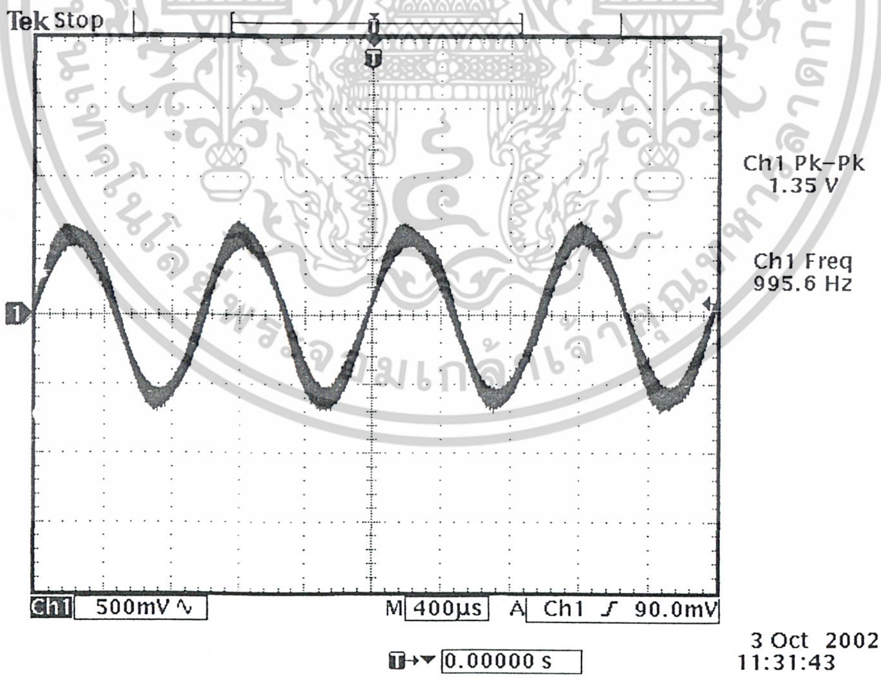
รูปที่ 4.7 แสดงวงจรส่วนบันทึกและเล่นกลับโดยต่อ ISD 2590 ร่วมกัน 3 ตัว

เมื่อต่อวงจรส่วนบันทึกและเล่นกลับดังรูปที่ 4.7 แล้ว ทำการบันทึกสัญญาณไซน์เวฟ ความถี่ 1 กิโลเฮิรตซ์ โดยมีขนาดแอมปลิจูด เท่ากับ 1 โวลต์พีคทูพีค จากนั้นจึงทำการเล่นกลับ โดยวัดสัญญาณที่ขา SP+ เทียบกับกราวด์ ได้สัญญาณออกมาดังรูปที่ 4.8

ผลการทดลอง ของวงจรบันทึกเสียง



รูปที่ 4.8 แสดงรูปสัญญาณที่อินพุตก่อนเข้าวงจรบันทึกเสียง



รูปที่ 4.9 แสดงรูปสัญญาณที่เอาท์พุตที่ได้จากวงจรบันทึกเสียง

4.3 การใช้งานเครื่องอำนวยความสะดวกในการใช้โทรศัพท์

สามารถแบ่งลักษณะการทำงานออกเป็น 3 สภาวะ

1. สภาวะที่ไม่มีการโทรเข้าและโทรออก
2. สภาวะเมื่อมีการโทรเข้า
3. สภาวะเมื่อมีการโทรออก

4.3.1 สภาวะเมื่อไม่มีการโทรเข้าและโทรออก

ในสภาวะนี้ เครื่องจะทำการแสดงวันและเวลาปัจจุบันที่หน้าจอแอลซีดีที่หน้าเครื่อง ซึ่งจะเป็นการแจ้งสภาวะว่าอยู่ในสภาวะใด แสดงได้ดังรูปที่ 4.10

Telecom.
21/03/03 20 : 08 : 38

รูปที่ 4.10 แสดงสภาวะเมื่อไม่มีการโทรเข้าและโทรออก

นอกจากนี้ที่หน้าเครื่องจะมีคีย์บอร์ดติดตั้งไว้ด้วย เพื่อสำหรับป้อนข้อมูลเพื่อกำหนดค่าต่าง ๆ ของเครื่องที่เราต้องการลงไปได้ โดยลักษณะของคีย์บอร์ดจะเป็นดังรูปที่ 4.11

1	2	3	MENU
4	5	6	⇦
7	8	9	⇨
*	0	#	ENTER

รูปที่ 4.11 แสดงลักษณะคีย์บอร์ดในการใช้งาน

ที่ปุ่มเลือกเมนูจะประกอบไปด้วยโหมดการทำงานต่าง ๆ ดังต่อไปนี้

1) โหมดการตั้งค่าให้กับเครื่อง ประกอบไปด้วย

- การตั้งวันและเวลาปัจจุบัน
- การตั้งรหัสผ่าน
- การตั้งจำนวนกระดิ่ง

Set Mode

1. Time-Date

2. Password

3. Ringing

รูปที่ 4.12 แสดงสภาวะในการตั้งค่าต่าง ๆ

2) โหมดการตรวจสอบข้อความที่ถูกฝากไว้ ประกอบไปด้วย

- การตรวจสอบข้อความ ณ วันปัจจุบันที่ทำการตรวจสอบ
- การตรวจสอบข้อความทั้งหมด 15 ข้อความ

Check Mesg.

1. Today

2. All

3. Exit

รูปที่ 4.13 แสดงสภาวะการตรวจสอบข้อความ

3) โหมดการตรวจสอบข้อมูลการใช้โทรศัพท์ ประกอบไปด้วย

- การตรวจสอบค่าบริการโทรศัพท์รวมของแต่ละเดือน

Month.. 03

Price = 0003 Baht

รูปที่ 4.14 แสดงสภาวะค่าบริการรวมของแต่ละเดือน

- การตรวจสอบเลขหมายปลายทาง, วัน-เวลา ในการโทรออกและจำนวนเงินค่าโทรศัพท์และจำนวนเงินค่าโทรศัพท์ (แสดงการโทรภายนอกไปยังเบอร์ 023871216)

Phone 023871216

Date 21/03/03

Time 22 : 18 : 45

Price 0003 Baht

รูปที่ 4.15 แสดงรายละเอียดข้อมูลการโทรแต่ละครั้ง

4) โหมดการควบคุมเครื่องใช้ไฟฟ้า 8 ชาแนล ประกอบไปด้วย
   - เปิดเครื่องใช้ไฟฟ้า
   - ปิดเครื่องใช้ไฟฟ้า

On/Off Electric

Press * on

Press # off

Number :

รูปที่ 4.16 แสดงสภาวะการปิด-เปิดเครื่องใช้ไฟฟ้า

4.3.2 สภาวะเมื่อมีการโทรเข้า

ในสภาวะนี้เมื่อมีการเรียกเข้าเครื่องจะทำการแสดงสภาวะของการโทรเข้า ดังรูปที่ 4.18 ถ้าไม่มีผู้มารับสายภายในระยะเวลาการกำหนดจำนวนกระดิ่งที่ตั้งเอาไว้ เครื่องก็จะทำการตัดเข้าสู่ระบบตอบรับอัตโนมัติทันที ดังรูปที่ 4.17

Incoming Call

รูปที่ 4.17 แสดงสภาวะการเรียกเข้า

Auto Answering

รูปที่ 4.18 แสดงสภาวะการเข้าสู่ระบบอัตโนมัติ

จากนั้นเครื่องก็จะทำการแจ้งให้คู่สายปลายทางเลือกโหมดการทำงาน ประกอบไปด้วย 3 โหมด

1) โหมดการฝากข้อความ โดยเครื่องจะทำการบันทึกข้อความของผู้ฝากเอาไว้ แล้วทำการเก็บเอาไว้ในไอซีบันทึกเสียงซึ่งสามารถฝากข้อความได้สูงสุด 15 ข้อความ ข้อความละ 18 วินาที และจะทำการบันทึกวัน-เวลาขณะที่ทำการบันทึกลงในหน่วยความจำ

2) โหมดการควบคุมเครื่องใช้ไฟฟ้า โดยสามารถสั่งให้เครื่องใช้ไฟฟ้าปิดหรือเปิดได้ตามต้องการโดยควบคุมได้จากคีย์บอร์ดที่แป้นโทรศัพท์

3) โหมดการรับฟังข้อความ สามารถตรวจสอบข้อความที่ถูกฝากเอาไว้ โดยสามารถเลือกฟังแบบทั้งหมดหรือฟัง ณ วันที่ทำการตรวจสอบข้อความก็ได้

4.3.2 สภาวะเมื่อมีการโทรออก

ในสภาวะนี้เมื่อผู้ใช้ทำการยกหูโทรศัพท์เพื่อโทรออก เครื่องจะทำการแสดงสภาวะการโทรออก และถ้าหมายเลขปลายทางไม่ว่างเครื่องก็จะแจ้งว่าสายไม่ว่าง ถ้าทำการติดต่อได้สำเร็จและปลายทางรับสายแล้ว เครื่องก็จะทำการบันทึกวันและเวลาที่ติดต่อได้สำเร็จ บันทึกลงในหน่วยความจำจากนั้นที่หน้าจอแอลซีดีก็จะการแสดงผลซึ่งจะประกอบไปด้วยเลขหมายปลายทาง, เวลาที่เริ่มต้นติดต่อสำเร็จ, เวลาปัจจุบัน และค่าบริการโทรศัพท์ ณ ขณะนั้น แสดงได้ดังรูปที่ 4.19

Phone 023871216
Start 22 : 18 :45
Stop 22 : 20 : 36
Price 0003 Baht

รูปที่ 4.19 แสดงสภาวะเมื่อมีการสนทนา

เมื่อผู้ใช้สิ้นสุดการสนทนาและทำการวางสาย เครื่องก็จะแสดงค่าโทรศัพท์ทั้งหมด และทำการบันทึกลงในหน่วยความจำ

บทที่ 5
บทวิจารณ์และบทสรุป

5.1 ปัญหาในการทำโครงการ มีดังต่อไปนี้

5.1.1 ในส่วนของดีเทกสัญญาณโทรศัพท์ คือ ความถี่ของสัญญาณโทรศัพท์ภายนอกทั่วไป กับความถี่ของสัญญาณโทรศัพท์ภายในสถาบันซึ่งเป็นระบบตู้สาขาแบบ PABX นั้นมีความถี่ไม่เท่ากันทำให้ในการทดลองซึ่งได้ออกแบบไว้สำหรับสัญญาณโทรศัพท์ภายนอกสถาบัน และเมื่อนำไปทดลองกับสัญญาณภายในจะเกิดความผิดพลาดขึ้น อันเนื่องมาจากความถี่ของสัญญาณโทรศัพท์ทั้งสองมีค่าความถี่ไม่เท่ากันนั่นเอง

5.1.2 เนื่องจากปัจจุบันโทรศัพท์ตามบ้านนั้นได้มีอยู่ 2 บริษัท ที่ให้บริการ คือ ทศท.(TOT) และเทเลคอมพ์เอเชีย(TA) ซึ่งก็จะมีความถี่ที่ไม่เท่ากัน แต่แตกต่างกันเล็กน้อย ทำให้อาจดีเทกสัญญาณต่างๆ ของโทรศัพท์ได้ไม่ดีนัก ถ้าหากเราตั้งค่าความถี่ไว้ตรงที่ของบริษัทใดบริษัทหนึ่ง

5.1.3 ในปัจจุบันนี้ได้มีโทรศัพท์แบบ PCT เกิดขึ้น ซึ่งเวลาที่โทรเข้าเครื่อง PCT นี้จะไม่มีสัญญาณเรียกกลับ ตอบกลับมา แต่จะเป็นเสียงคนพูดตอบกลับมาแทนซึ่งเป็นผลทำให้วงจรดีเทกสัญญาณนี้ ตรวจจับสัญญาณไม่ได้

5.1.4 ในส่วนของโปรแกรมนั้นเมื่อทำการแปลงเป็นไฟล์ HEX จะมีค่าประมาณ 22 Kbytes ซึ่งค่อนข้างจะใหญ่ ทำให้ทำการถ่ายข้อมูลลงไมโครคอนโทรลเลอร์ตระกูล 8051 ได้

5.2 วิธีการแก้ไขปัญหา มีดังต่อไปนี้

1. ในส่วนของการดีเทกสัญญาณโทรศัพท์นั้น อาจทำได้โดยการออกแบบสร้างวงจรดีเทกสัญญาณสำหรับสัญญาณโทรศัพท์ภายในสถาบันขึ้นมาอีกชุดหนึ่ง แล้วจึงนำผลที่ได้ของทั้งสองวงจรมารวมกันจากนั้นจึงนำไปใช้งาน

2. ในส่วนของเรื่องความถี่ที่ต่างกันเล็กน้อยนั้น อาจแก้ปัญหาด้วยวิธีการข้างต้นก็ได้ หรืออาจใช้การปรับตัวอุปกรณ์ดีเทกสัญญาณโทรศัพท์ให้ทำงานที่ความถี่ตรงช่วงกลางระหว่างความถี่ทั้งสอง

3. ในส่วนของโปรแกรมที่มีขนาดใหญ่นั้น อาจแก้ไขโดยใช้ ROM มาต่อเข้าไปด้วยเพื่อเพิ่มเนื้อที่ในการบันทึกโปรแกรม หรืออาจไปใช้ไมโครคอนโทรลเลอร์ตระกูล 8951 ก็ได้เนื่องจากมีพื้นที่ในการบันทึกข้อมูล 32 Kbytes

5.3 แนวทางการพัฒนาต่อไป

5.3.1 ถ้าต้องการเพิ่มเวลาในการบันทึกข้อความมากขึ้น ทำได้โดยการเพิ่ม ISD 2590 เข้าไปอีกซึ่ง 1 ตัว จะสามารถบันทึกข้อความได้ 90 วินาที

5.3.2 เพิ่มเนื้อที่ในการเก็บข้อมูลการโทร โดยการเพิ่มหน่วยความจำให้มากขึ้น

5.3.3 เพิ่มโปรแกรมส่วนแสดงผลออกทางเครื่องพิมพ์

ภาคผนวก ก.

โปรแกรมการใช้งาน

```
;***********************INITIALPORT******************************
PORTA1          EQU     0E000H      ;LCD DATA
PORTB1          EQU     0E001H      ;LCD CONTROL
PORTC1          EQU     0E002H      ;KEYBOARD
PORTD1          EQU     0E003H      ;CONTROL WORD8255#1
PORTA2          EQU     0E400H      ;SIGNAL DETECTOR
PORTB2          EQU     0E401H      ;DTMF
PORTC2          EQU     0E402H      ;SIGNAL CONTROL
PORTD2          EQU     0E403H      ;CONTROL WORD8255#2
PORTA3          EQU     0E800H      ;CONTROL ELECTRIC
PORTB3          EQU     0E801H
PORTC3          EQU     0E802H
PORTD3          EQU     0E803H      ;CONTROL WORD8255#3
PORTA4          EQU     0EC00H      ;ANSWER ADDRESS
PORTB4          EQU     0EC01H      ;ANSWER CONTROL
PORTC4          EQU     0EC02H      ;INDICATOR
PORTD4          EQU     0EC03H      ;CONTROL WORD8255#4
MONEY_R         EQU     09H         ;MONEY PRESENT
MONEY_RR        EQU     0BH         ;MONEY TOTAL
NUMBER_AD       EQU     0DH         ;DIGIT&NUMBER
TIME_SP         EQU     06H         ;TIME STOP
DIGIT_B         EQU     1FH         ;DIGIT NUMBER
BUFF_N          EQU     20H         ;BUFFER NUMBER
RATE0           EQU     61H         ;FULL RATE
RATE            EQU     2AH         ;REAL RATE
MONEY_B         EQU     2BH         ;MONEY PRESENT
MONEY_B2        EQU     2CH         ;MONEY&PRESENT
MONEY_T         EQU     2DH         ;MONEY TOTAL
MONEY_T2        EQU     2EH         ;MONEY&TOTAL
R_COUNT         EUQ     0065H       ;RINGING
PASSWD          EQU     0066H       ;PASSWORD
ELECTRIC        EQU     0005H       ;ELEC DATA
REC             EQU     0006H       ;PRESENT RECORD
MONTH_P         EQU     006BH       ;PRESENT MONTH
TOTAL_M         EQU     006CH       ;TOTAL MONEY
CALL_OUT        EQU     006EH       ;PRESENT CALL OUT
DATA            EQU     0070H       ;START DATA OUT GOING
T_DATA          EQU     P1.0
T_CLK           EQU     P1.1
T_RST           EQU     P1.2
THBUF           EQU     30H         ;TIME HEX
DHBUF           EQU     33H         ;DATE HEX
TASCBU          EQU     36H         ;TIME ASCII
DASCBU          EQU     40H         ;DATE ASCII
KEYBUF          EQU     48H         ;SAVE KEY
KEYMEM          EQU     49H         ;MEM KEY
CORON           EQU     4AH         ; :,/
PASSMEM1        EQU     4BH         ;PASSWORD MEM1
PASSMEM2        EQU     4FH         ;PASSWORD MEM2
CONELEC         EQU     53H         ;CONTROL ELEC
BILL1           EQU     54H
BILL2           EQU     55H
P_AT            EQU     60H
;************************INITIAL8255*****************************
        ;-----------------------PORT 8255---------------------------
                ORG     0000H
                MOV     SP,#07H
                MOV     DPTR,#PORTD1
```

```
        MOV     A,#81H              ;A,B=OUT , CH=OUT
        MOVX    @DPTR,A             ;CL=IN
        MOV     R2,#01H
        LCALL   DELAY
        MOV     DPTR,#PORTD2
        MOV     A,#92H              ;A,B=IN , C=OUTPUT
        MOVX    @DPTR,A
        MOV     R2,#01H
        LCALL   DELAY
        MOV     DPTR,#PORTD3
        MOV     A,#89H              ;A,B=OUT , C=IN
        MOVX    @DPTR,A
        MOV     R2,#01H
        LCALL   DELAY
        MOV     DPTR,#PORTD4
        MOV     A,#89H              ;A,B=OUT , C=IN
        MOVX    @DPTR,A
        MOV     R2,#01H
        LCALL   DELAY
        MOV     A,#3FH
        LCALL   SET_P
;----------------------------LCD INITIALIZE-----------------------
        MOV     A,#00111000B        ;FUNCTION SET
        LCALL   LCDWI
        MOV     A,00001110B         ;DISPLAY ON/OFF
        LCALL   LCDWI
        MOV     A,#01H              ;CLEAR
        LACLL   LCDWI
        MOV     R2,#2
        LCALL   DELAY
        MOV     A,#03H
        LCALL   LCDWI
        MOV     R2,#2
        LCALL   DELAY
        MOV     DPTR,#PAGE1         ;SHOW SET DEFUALT
        LCALL   LCDLD
        MOV     R2,#15H
        LCALL   DELAY
;-------- -------------RING COUNT,PASSWORD,ELECTRIC------------
DEFAULT: MOV    DPTR,#R_COUNT
        MOV     A,#02H              ;RING=02H
        MOVX    @DPTR,A
        MOV     R6,#00H             ;PASSWORD=1111
        MOV     DPTR,#PASSWD
LP:     MOV     A,#01H
        MOVX    @DPTR,A
        INC     R6
        INC     DPTR
        CJNE    R6,#04H,LP
        MOV     DPTR,#ELECTRIC
        MOV     A,#00H
        MOVX    @DPTR,A
;----------------------------RECORD ISD-----------------------
DEF_REC: MOV    DPTR,#REC           ;ADD 0006H
        MOV     A,#78H
        MOVX    @DPTR,A
        INC     DPTR                ;ADD 0007H
        MOV     A,#1CH
        MOVX    @DPTR,A
        INC     DPTR                ;ADD 0008H
        MOV     A,#00H
```

```
        MOVX    @DPTR,A
        INC     DPTR                    ;ADD 0009H
        MOV     A,#5EH
        MOVX    @DPTR,A
REC_START: MOV  DPTR,#006EH
        MOV     A,#00H
        MOVX    @DPTR,A
        INC     DPTR
        MOV     A,#70H
        MOVX    @DPTR,A
;---------------------------MONEY-------------------------------
MONEY_TOTAL:MOV 2AH,#00H
        MOV     2BH,#00H
        MOV     2CH,#00H
        MOV     2EH,#00H
        MOV     R2,#02H
        MOV     DPTR,#006CH
M_T:    MOV     A,#00H
        MOVX    @DPTR,A
        INC     DPTR
        DJNZ    R2,M_T
        LCALL   D_READ
        MOV     DPTR,#MONTH_P
        MOV     A,34H
        MOVX    @DPTR,A
        MOV     30H,#00H                ;CLEAR TIME HEX BUF
        MOV     31H,#00H
        MOV     32H,#00H
        MOV     49H,#00H                ;CLEAR KEYMEM
        MOV     4AH,#0C4H               ;LCD ADDRESS OF DATE
        LJMP    MAIN0
;******************************************************************
;                       MAIN PROGRAM
;******************************************************************
MAIN0:  CLR     T_RST                   ;MAIN RESET
        SETB    T_CLK
        LCALL   TDELAY
        MOV     R0,#8EH
        MOV     R1,#00H
        LCALL   RTC_W
        MOV     R0,#80H
        MOV     R1,#30H
        LCALL   RTC_W
        MOV     R0,#8EH
        MOV     R1,#80H
        LCALL   RTC_W
MAIN:   MOV     DPTR,#PAGE2
        LCALL   LCDLD
        MOV     R2,#03H
        LCALL   DELAY
        LCALL   CLEAR
        MOV     A,#01H
        LCALL   LCDWI
        MOV     R5,#00H
        MOV     R7,#00H
MAIN00: MOV     DPTR,#PAGE17
        LCALL   LCDLD
MAIN1:  LCALL   HOOK
        CJNE    R0,#10H,M_1
        LJMP    DIAL_CK
M_1:    LCALL   RING
```

```
          CJNE      R0,#08H,M_2
          MOV       R7,#00H
          LJMP      RING_C
M_2       LCALL     T_READ          ;SHOW TIME
          MOV       CORON,#3AH      ;DISPLAY TIME
          MOV       R0,#0D4H
          MOV       R1,#TASCBU
          LCALL     DIS_LCD
          LCALL     D_READ          ;SHOW DATE
          MOV       CORON,2FH       ;DISPLAY DATE
          MOV       R0,#094H
          MOV       R1,#DASCBU
          LCALL     DIS_LCD
          LCALL     SCAN
          MOV       A,KEYMEM
          CJNE      A,#4DH,EEM      ;CHECK KEY MENU
          LJMP      MENU            ;IF YES GO TO MENU
EEM:      MOV       KEYMEM,#00H     ;CLEAR KEYMEM
          LJMP      MAIN1
CONV:     MOV       DPTR,#PAGE3
          LCALL     LCDLD
          MOV       DPTR,#PORTC2    ;CONV=PC2
          MOV       A,#04H
          MOVX      @DPTR,A
          MOV       R2,#01H
          LCALL     DELAY
          LCALL     HOOK
          CJNE      R0,#00H,CONV    ;00H=END CONV
          LJMP      MAIN
RING_C2:  LCALL     HOOK
          CJNE      R0,#00H,CONV
RING_C21: INC       R7
          MOV       R2,#0B3H
          LCALL     DELAY1
          LCALL     RING
          CJNE      R0,#00H,RING_C
          CJNE      R7,#35H,RING_C21 ;5 SEC
          LJMP      MAIN
RING_C3:  LCALL     HOOK
          CJNE      R0,#01H,RING_C1
          LJMP      CONV
RING_C:   LCALL     RING
          CJNE      R0,#08H,RING_C2
RING_C1:  LCALL     RING
          CJNE      R0,#00H,RING_C3
          INC       R5
          MOV       R2,#0FH
          LCALL     DELAY
          MOV       A,#01H
          LCALL     LCDWI
          MOV       DPTR,#PAGE1
          LCALL     LCDLD
MAIN11:   LCALL     HOOK
          CJNE      R0,#00H,CONV
          MOV       DPTR,#R_COUNT
          MOVX      A,@DPTR
          SUBB      A,R5
          CJNE      A,#00H,RING_C
AUTO_H    MOV       DPTR,#PAGE5
          LCALL     LCDLD
          MOV       DPTR,#PORTC2
```

```
          MOV       A,#01H               ;AUTO_HOOK=ON
          MOVX      @DPTR,A
          MOV       R2,#01H
          LCALL     DELAY
          MOV       R6,#00H              ;ANSWER MSG#1
          MOV       R7,#39H              ;ISD#1
          LCALL     PLAY1
CHOOSE:   LCALL     DTMF_IN
          CJNE      R5,#01H,NUM2         ;CHK NUMBER DTMF
NUM1:     MOV       DPTR,#PAGE6          ;LEAVE MSG
          LCALL     LCDLD
          MOV       R6,#1AH
          MOV       R7,#39H
          LCALL     PLAY1
          LCALL     T_READ
          LCALL     D_READ
          LCALL     RECORD
          LJMP      MAIN
NUM2:     CJNE      R5,#02H,NUM3         ;CONTROL ELECTRIC
          MOV       DPTR,#PAGE7
          LCALL     LCDLD
          MOV       R6,#2AH
          MOV       R7,#39H
          LCALL     PLAY1
          LCALL     PWD
          LJMP      CONTROL_E
NUM3:     CJNE      R5,#03H,CHOOSE       ;CHK MSG
          MOV       DPTR,#PAGE8
          LCALL     LCDLD
          MOV       R6,#2AH
          MOV       R7,#39H
          LCALL     PLAY1
          LCALL     PWD
AGAIN1:   MOV       DPTR,#PAGE22
          LCALL     LCDLD
          MOV       R6,#5CH
          MOV       R7,#39H
          LCALL     PLAY1
          LCALL     DTMF_IN
          CJNE      R5,#01H,ALL
TO_DAY:   MOV       P_AT,R5
          LJMP      P_TODAY
ALL:      CJNE      R5,#02H,AGAN2
          MOV       P_AT,R5
          LJMP      P_ALL
AGAN2:    MOV       R6,#32H
          MOV       R7,#39H
          LCALL     PLAY1
          LJMP      AGAN1
     ;-----------------------DIAL CHECK----------------------------
DIAL_CK:  MOV       DPTR,#PAGE9
          LCALL     LCDLD
          LCALL     DIAL
          CJNE      R0,#02H,HOOK_CK1
          MOV       R6,#00H
          MOV       R1,#BUFF_N           ;BUFF_NUM=20H
RD_DTMF:  LCALL     DTMF_OUT
          CJNE      R5,#00H,INNER01
OUTER0:   LCALL     DIAL
          CJNE      R0,#02H,HOOK_CK2
OUTER1:   LCALL     DTMF_OUT
```

```
            CJNE        R5,#01H,OUTER2
            INC         R6
            LCALL       WR_BUFF
            LCALL       DTMF_OUT
            CJNE        R5,#01H,OUTER3
OUTER11:    INC         R6
            LCALL       WR_BUFF
            CJNE        R6,#04H,OUTER12
            LJMP        OUTER_R
OUTER12:    LCALL       DTMF_OUT
            SJMP        OUTER11
INNER01:    LJMP        INNER1
HOOK_CK1:   LCALL       HOOK
            CJNE        R0,#00H,DIAL_CK
            LJMP        MAIN
HOOK_CK2:   LCALL       HOOK
            CJNE        R0,#00H,OUTER0
            LJMP        MAIN
OUTER2:     CJNE        R5,#00H,OUTER21
            LJMP        FAR
OUTER21:    INC         R6
            LCALL       WR_BUFF
            CJNE        R6,#07H,OUTER22
            MOV         DIGIT_B,R6
            MOV         RATE,#03H
            MOV         MONEY_B,#00H
            LCALL       WAIT
            LJMP        RTD
OUTER22:    LCALL       DTMF_OUT
            SJMP        OUTER21
OUTER3:     CJNE        R5,#03H,OUTER4
            INC         R6
            LCALL       WR_BUFF
            LJMP        OUTER_R
OUTER4:     CJNE        R5,#05H,OUTER5
            INC         R6
            LCALL       WR_BUFF
            LCALL       DTMF_OUT
            CJNE        R5,#00H,OUTER43
OUTER41:    INC         R6
            LCALL       WR_BUFF
            CJNE        R6,#04H,OUTER42
            LJMP        OUTER_R
OUTER42:    LCALL       DTMF_OUT
            SJMP        OUTER41
OUTER43:    INC         R6
            LCALL       WR_BUFF
            LJMP        OUTER_R
OUTER5:     CJNE        R5,#07H,OUTER6
            INC         R6
            LCALL       WR_BUFF
            CJNE        R6,#05H,OUTER51
            LJMP        OUTER_R
OUTER51:    LCALL       DTMF_OUT
            SJMP        OUTER5
OUTER6:     INC         R6
            LCALL       WR_BUFF
            CJNE        R6,#03H,OUTER61
            LJMP        OUTER_R
OUTER61:    LCALL       DTMF_OUT
            SJMP        OUTER6
```

```
OUTER_R:    MOV     DIGIT_B,R6
            LCALL   RATE_M
            LCALL   WAIT
            LJMP    RTD
FAR:        INC     R6
            LCALL   WR_BUFF
            CJNE    R6,#09H,FAR1
            MOV     DIGIT_B,R6
            LCALL   RATE_FAR
            MOV     MONEY_B,#00H
            LCALL   WAIT
            LJMP    RTD
FAR1:       LCALL   DTMF_OUT
            SJMP    FAR
INNER:      LCALL   DTMF_OUT            ;INNER KMITL
INNER1:     INC     R6                  ;R6=COUNT NUMBER
            LCALL   WR_BUFF
            CJNE    R6,#04H,INNER
            MOV     DIGIT_B,R6
            MOV     RATE,#03H           ;2AH=MONEY RATE
            LCALL   WAIT
            LJMP    RTD
RTD:        MOV     RATE0,RATE
            LCALL   REC_T_D
            MOV     DPTR,#CALL_OUT
            MOVX    A,@DPTR
            MOV     R0,A
            INC     DPTR
            MOVX    A,@DPTR
            ADD     A,#NUMBER_AD        ;ADD_DIGIT&NUMBER
            MOV     R1,A
            MOV     DPH,R0
            MOV     DPL,R1
            MOVX    A,@DPTR
            MOV     R5,A
            MOV     R0,#86H             ;ADDRESS START LCD
            MOV     R1,#BUFF_N
            LCALL   DIS_NUM1
            LCALL   T_START
            LCALL   RUN_TIME
            AJMP    TOTAL
TOTAL:      MOV     DPTR,#PAGE32
            LCALL   LCDLD
            MOV     R0,#MONEY_B         ;SHOW_PRICE
            MOV     R1,#BUFF_N
            MOV     R3,#04H
TOTAL1:     MOV     A,@RO
            ANL     A,#0F0H
            SWAP    A
            ORL     A,#30H
            MOV     @R1,A
            MOV     A,@R0
            ANL     A,#0FH
            ORL     A,#30H
            INC     R1
            MOV     @R1,A
            INC     R0
            INC     R1
            DJNZ    R3,TOTAL1
            MOV     R3,#0C7H
            MOV     R1,BUFF_N
```

```
        MOV     R5,#04H
        LCALL   DIS_NUM1
        MOV     R2,#0FH
        LCALL   DELAY
NEXT_REC:  MOV     DPTR,#CALL_OUT     ;NEXT_RECORD
        MOVX    A,@DPTR
        MOV     R0,A
        INC     DPTR
        MOVX    A,@DPTR
        MOV     R1,A
        MOV     R2,#15H
NX:     INC     R1
        CJNE    R1,#00H,NX1
        INC     R0
NX1:    DJNZ    R2,NX
        MOV     DPTR,#CALL_OUT
        MOV     A,R0
        MOVX    @DPTR,A
        INC     DPTR
        MOV     A,R1
        MOVX    @DPTR,A
        LJMP    MAIN0
;-------------------------WRITE BUFF&RAM-----------------------
WR_BUFF:   MOV     A,R5
        MOV     @R1,A
        INC     R1
        RET
WR_RAM:    MOV     R1,#DIGIT_B
        MOV     A,@R1
        MOV     R6,A
        MOVX    @DPTR,A
        INC     DPTR
        MOV     R0,#BUFF_N
WR_RAM1:   MOV     A,@R0
        SWAP    A
        DEC     R6
        CJNE    R6,#00H,WR_RAM11
        ORL     A,#00H
        SJMP    WR_RAM12
WR_RAM11:  INC     R0
        ORL     A,@R0
        MOVX    @DPTR,A
        INC     DPTR
        INC     R0
        DEC     R1
        CJNE    R1,#00H,WR_RAM1
        RET
WR_RAM12:  MOVX    @DPTR,A
        RET
;---------------------SHOW TIME&DATE TO LCD--------------------
REC_T_D:   LCALL   T_READ          ;READ TIME AND DATE
        LCALL   D_READ
        MOV     DPTR,#CALL_OUT
        MOVX    A,@DPTR
        MOV     R0,A
        INC     DPTR
        MOVX    A,@DPTR
        MOV     R1,A
        MOV     DPH,R0
        MOV     DPL,R1
        MOV     R2,#06H
```

```
            MOV     R0,#THBUF
REC_T_D1:   MOV     A,@R0
            MOVX    @DPTR,A
            INC     R0
            INC     DPTR
            DJNZ    R2,REC_T_D1
            MOV     DPTR,#CALL_OUT
            MOVX    A,@DPTR
            MOV     R0,A
            INC     DPTR
            MOVX    A,@DPTR
            ADD     A,#NUMBER_AD        ;ADD DIGIT&NUMBER
            MOV     R1,A
            MOV     DPH,R0
            MOV     DPL,R1
            LCALL   WR_RAM
DISPLAY:    MOV     DPTR,#PAGE3
            LCALL   LCDLD
            MOV     R2,#03H
            LCALL   DELAY
            MOV     DPTR,#CALL_OUT
            MOVX    A,@DPTR
            MOV     R0,A
            INC     DPTR
            MOVX    A,@DPTR
            ADD     A,#NUMBER_AD
            MOV     R1,A
            MOV     DPH,R0
            MOV     DPL,R1
            MOVX    A,@DPTR
            MOV     R6,A
            INC     DPTR
            MOV     R0,#BUFF_N
DISPLAY1:   MOVX    A,@DPTR
            ANL     A,#0F0H             ;FIRST DIGIT
            SWAP    A
            ORL     A,#30H
            MOV     @R0,A
            INC     R0
            DEC     R6
            CJNE    R6,#00H,DISPLAY2
            SJMP    DISPLAY3
DISPLAY2:   MOVX    A,@DPTR
            ANL     A,#0FH
            ORL     A,#30H
            MOV     @R0,A
            INC     R0
            DEC     R6
            INC     DPTR
            CJNE    R6,#00H,DISPLAY1
            SJMP    DISPLAY3
DISPLAY3:   MOV     DPTR,#CALL_OUT
            MOVX    A,@DPTR
            MOV     R0,A
            INC     DPTR
            MOVX    A,@DPTR
            MOV     R1,A
            MOV     DPH,R0
            MOV     DPL,R1
            MOV     R2,#03H
            MOV     R0,#THBUF
```

```
DISPLAY4:   MOVX    A,@DPTR
            MOV     @R0,A
            INC     R0
            INC     DPTR
            DJNZ    R2,DISPLAY4
            MOV     R7,32H
            MOV     R0,#THBUF
            MOV     R1,#TASCBU
            LCALL   H_T_A
DIS_NUM:    MOV     A,#01H
            LCALL   LCDWI
            MOV     DPTR,#PAGE29
            LCALL   LCDLD
            RET
DIS_NUM1:   MOV     A,R0
            LCALL   LCDWI
            MOV     A,@R1
            LCALL   LCDWD
            MOV     R2,#01H
            LCALL   DELAY
            INC     R0
            INC     R1
            DJNZ    R5,DIS_NUM1
            RET
T_START:    MOV     CORON,#3AH
            MOV     R0,#0C6H
            MOV     R1,#TASCBU
            LCALL   DIS_LCD
            RET
;---------------------------------RUN TIME2-----------------------------
RUN_TIME:   MOV     MONEY_B,#00H
            MOV     MONEY_B2,#00H
            MOV     DPTR,#MONTH_B
            MOVX    A,@DPTR
            MOV     R7,A
            MOV     DPTR,#TOTAL_M
            MOVX    A,@DPTR
            MOV     MONEY_T,A
            INC     DPTR
            MOVX    A,@DPTR
            MOV     MONEY_T2,A
            LCALL   D_READ
            MOV     A,#34H              ;MONTH
            SUBB    A,R7
            CJNE    A,#00H,RTT
            AJMP    RT0
RTT:        MOV     MONEY_T,#00H
            MOV     MONEY_T2,#00H
RT0:        MOV     R3,RATE
            CJNE    R3,#00H,RTT1
            SJMP    NEAR
RTT1:       CJNE    R3,#00H,RT01
            SJMP    NEAR
NEAR:       MOV     R2,RATE
            LCALL   MONEY_P
            MOV     R2,RATE
            LCALL   MONEY_TT
            MOV     R0,#MONEY_B         ;SHOW MONEY TO LCD
            MOV     R1,#BUFF_N
            MOV     R3,#02H
NEAR1:      MOV     A,@R0
```

```
        ANL     A,#0F0H
        SWAP    A
        ORL     A,#30H
        MOV     @R1,A
        MOV     A,@R0
        ANL     A,#0FH
        ORL     A,#30H
        INC     R1
        MOV     @R1,A
        INC     R0
        INC     R1
        JNZ     R3,NEAR1
        MOV     R0,#0D6H
        MOV     R1,#BUFF_N
        MOV     R5,#04H
NEAR2:  MOV     A,R0
        LCALL   LCDWI
        MOV     A,@R1
        LCALL   LCDWD
        MOV     R2,#01H
        LCALL   DELAY
        INC     R0
        INC     R1
        DJNZ    R5,NEAR2
NEAR3:  LCALL   T_READ
        MOV     CORON,#3AH
        MOV     R0,#96H
        MOV     R1,#TASCBU
        LCALL   DIS_LCD
        LCALL   HOOK
        CJNE    R0,#00H,NEAR3
        LJMP    REC_MONEY
RT01:   LCALL   T_READ
        MOV     R7,32H
RT00:   MOV     R2,RATE         ;FAR RATE
        LCALL   MONEY_P
        MOV     R2,RATE
        LCALL   MONEY_TT
        MOV     R0,#MONEY_B
        MOV     R1,#BUFF_N
        MOV     R3,#02H
RT1:    MOV     A,@R0           ;R0=THBUF , DHBUF
        ANL     A,#0F0H         ;R1=TASCBU , DASCBU
        SWAP    A
        ORL     A,#30H
        MOV     @R1,A
        MOV     A,@R0
        ANL     A,#0FH
        ORL     A,#30H
        INC     R1
        MOV     @R1,A
        INC     R0
        INC     R1
        DJNZ    R3,RT1
        MOV     R0,#0D6H
        MOV     R1,#BUFF_N
        MOV     R5,#04H
RT2:    MOV     A,R0
        LCALL   LCDWI
        MOV     A,@R1
        LCALL   LCDWD
```

```
            MOV         R2,#01H
            LCALL       DELAY
            INC         R0
            INC         R1
            DJNZ        R5,RT2
RT3:        LCALL       T_READ              ;CHECK EVERY MINUTES
            MOV         A,32H
            SUBB        A,R7
            CJNE        A,#00H,RUN_TIME2
            MOV         R2,#03H
            LCALL       DELAY
            LCALL       T_READ
            MOV         R6,30H
            MOV         A,R6
            CLR         0D7H
            CJNE        A,#07H,RT4
            SJMP        RT00
RT4:        JC          MID_NIGHT           ;UNDER 06.00 O'CLOCK
            MOV         A,R6
            CLR         0D7H
            CJNE        A,#18H,RT5
NIGHT:      MOV         A,RATE0
            MOV         B,#02H
            DIV         AB
            MOV         RATE,A
            LJMP        RT00
RT5:        JC          RT00
            MOV         A,R6
            CLR         0D7H
            CJNE        A,#22H,RT6
MID_NIGHT:  MOV         A,RATE0
            MOV         B,#03H
            DIV         AB
            MOV         RATE,A
            LJMP        RT00
RT6:        JC          NIGHT
            SJMP        MID_NIGHT
            LJMP        RT00
RUN_TIME2:  MOV         CORON,#3AH
            MOV         R0,#96H             ;ADDRESS TIME ON LCD
            MOV         R1,#TASCBU
            LCALL       DIS_LCD
            LCALL       HOOK
            CJNE        R0,#10H,REC_MONEY
            SJMP        RT3
REC_MONEY:  MOV         R2,#01H
            LCALL       DELAY
            MOV         DPTR,#CALL_OUT      ;REC MONEY TO RAM
            MOVX        A,@DPTR
            MOV         R0,A
            INC         DPTR
            MOVX        A,@DPTR
            ADD         A,#MONEY_R
            MOV         R1,A
            MOV         DPH,R0
            MOV         DPL,R1
            MOV         R2,#04H
            MOV         R0,#MONEY_B
REC_MONEY1: MOV         A,@R0
            MOVX        @DPTR,A
            INC         R0
```

```
        INC     DPTR
        DJNZ    R2,REC_MONEY1
        LCALL   D_READ
        MOV     DPTR,#MONEY_P
        MOV     A,34H                   ;MONTH NOW
        MOVX    ,@DPTR,A
        MOV     DPTR,#TOTAL_M
        MOV     A,MONEY_T
        MOVX    @DPTR,A
        INC     DPTR
        MOV     A,MONEY_T2
        MOVX    @DPTR,A
        MOV     DPTR,#CALL_OUT          ;REC TIME STOP TO RAM
        MOVX    A,@DPTR
        MOV     R0,A
        INC     DPTR
        MOVX    A,@DPTR
        ADD     A,#TIME_SP
        MOV     R1,A
        MOV     DPH,R0
        MOV     DPL,R1
        MOV     R2,#03H
        MOV     R0,#THBUF
LP_T:   MOV     A,@R0
        MOVX    @DPTR,A
        INC     R0
        INC     DPTR
        DJNZ    R2,LP_T
        RET
MONEY_P: MOV    R3,MONEY_B2             ;IF BILL1 AND 2= 00
        MOV     R4,MONEY_B
MONEY_1: MOV    A,R3
        ANL     A,#0FH
        CJNE    A,#09H,MONEY2           ;CHECK 09
        MOV     A,R3                    ;IF YES
        ANL     A,#0F0H
        MOV     R3,A
        CJNE    A,#90H,MONYE3           ;CHECK 99
        MOV     R3,#00H                 ;IF YES SET = 00
        MOV     A,R4                    ;CHECK BILL2
        ANL     A,#0FH
        CJNE    A,#09H,MONEY4           ;CHECK 09
        MOV     A,R4                    ;IF YES
        ANL     A,#0F0H
        MOV     R4,A
        CJNE    A,#90H,MONEY5           ;CHECK 99
        MOV     R4,#00H                 ;IF YES SET = 00
        SJMP    MONEY_E
MONEY_2: INC    R3
        SJMP    MONEY_E
MONEY_3: MOV    A,R3
        ADD     A,#10H
        MOV     R3,A
        SJMP    MONEY_E
MONEY_4: INC    R4
        SJMP    MONEY_E
MONEY_5: MOV    A,R4
        ADD     A,#10H
        MOV     R4,A
MONEY_E: DJNZ   R2,MONEY1
        MOV     MONEY_B2,R3
```

```
            MOV         MONEY_B,R4
            RET
MONET_TT:   MOV         R3,MONEY_T2         ;IF BILL1 AND 2 = 0
            MOV         R4,MONEY_T
MONEY11:    MOV         A,R3
            ANL         A,#0FH
            CJNE        A,#09H.,MONEY22     ;CHECK 09
            MOV         A,R3                ;IF YES
            ANL         A,#0F0H
            MOV         R3,A
            CJNE        A,#09H,MONEY33      ;CHECK 99
            MOV         R3,#00H             ;IF YES SET = 00
            MOV         A,R4                ;CHECK BILL2
            ANL         A,#0FH
            MOV         R4,A
            CJNE        A,#90H,MONEY55      ;CHECK 99
            MOV         R4,#00H             ;IF YES SET = 00
            SJMP        MONEY_EE
MONEY22:    INC         R3
            SJMP        MONEY_EE
MONEY33:    MOV         A,R3
            ADD         A,#10H
            MOV         R3,A
            SJMP        MONEY_EE
MONEY44:    INC         R4
            SJMP        MONEY_EE
MONEY55:    MOV         A,R4
            ADD         A,#10H
            MOV         R4,A
MONEY_EE:   DJNZ        R2,MONEY11          ;R2 IS RATE
            MOV         MONEY_T2,R3
            MOV         MONEY_T,R4
            RET
;-----------------------------BILL MODE---------------------------------
BILL_M:     MOV         DPTR,#PAGE33
            LCALL       LCDLD
B1:         LCALL       SCAN
            MOV         A,KEYMEM
            CJNE        A,#31H,B1
            LJMP        T_PRICE
B2:         CJNE        A,#32H,B3
            LJMP        D_PRICE
B3:         CJNE        A,#4DH,B4
            LJMP        MENU
B4:         MOV         KEYMEM,#00H
            LJMP        B1
T_PRICE:    MOV         DPTR,#PAGE35        ;TOTAL PRICE
            LCALL       LCDLD
            LCALL       D_READ
            MOV         R3,34H
            MOV         DPTR,#MONTH_P
            MOVX        A,@DPTR
            SUBB        A,R3
            CJNE        A,#00H,T_P1
            INC         DPTR
            MOV         2FH,R3
            LCALL       T_P_M1
            LCALL       T_P_M2
            LJMP        WAIT_KEY
T_P1:       MOV         MONEY_T,#00H
            MOV         MONEY_T2,#00H
```

```
        MOV     2FH,R3
        LCALL   T_P_M2
        MOV     DPTR,#CALL_OUT
        MOVX    A,@DPTR
        MOV     R0,A
        INC     DPTR
        MOVX    A,@DPTR
        MOV     R1,A
        MOV     DPH,R0
        MOV     DPL,R1
        MOV     A,#0FFH
        MOV     R2,#15H
T_P11:  MOVX    @DPTR,A
        INC     DPTR
        DJNZ    R2,T_P11
        LJMP    WAIT_KEY
WAIT_KEY: LCALL D_READ          ;LET DPTR = ADD.RATE
        MOV     2FH,34H
        MOV     3DH,34H
        MOV     DPTR,#CALL_OUT
        MOVX    A,@DPTR
        MOV     R0,A
        INC     DPTR
        MOVX    A,@DPTR
        ADD     A,#04H
        MOV     R1,A
        MOV     50H,R0
        MOV     51H,R1
;--------------------CHOOSE UP&DOWN--------------------------
T_P5:   MOV     KEYMEM,#00H
        LCALL   SCAN
        MOV     A,KEYMEM
        CJNE    A,#2BH,UP
DOWN:   MOV     R7,2FH
        DEC     R7
        CJNE    R7,#0FH,DOWN1
        MOV     R7,#09H
        SJMP    DOWN2
DOWN1:  CJNE    R7,#00H,DOWN2
        MOV     R7,#12H
DOWN2:  MOV     R0,50H
        MOV     R1,51H
T_P51:  LCALL   DECRETE
        MOV     50H,R0
        MOV     51H,R1
        MOV     DPH,R0
        MOV     DPL,R1
        MOVX    A,@DPTR
        CLR     0D7H
        SUBB    A,R7
        CJNE    A,#00H,T_P52
        SJMP    T_P53
T_P52:  CJNE    A,#01H,T_P511
T_P521: MOV     R0,50H
        MOV     R1,51H
        LJMP    DOWN2
T_P511: MOVX    A,@DPTR
        SUBB    A,R7
        JC      T_P521
        JNC     T_P521
T_P53:  MOV     R0,50H
```

```
        MOV     R1,51H
        MOV     A,R1
        ADD     A,#07H
        MOV     R1,A
        MOV     DPH,R0
        MOV     DPL,R1
        MOV     2FH,R7
        LCALL   T_P_M1
        LCALL   T_P_M2
        LJMP    T_P5
;------------------------UP--------------------------------
UP:     CJNE    A,#2DH,T3
        MOV     R7,2FH
        LCALL   D_READ
        MOV     3DH,34H
        INC     R7
        CJNE    R7,#0AH,UP1
        MOV     R7,#10H
        SJMP    UP11
UP1:    CJNE    R7,#13H,UP11
        MOV     R7,#01H
UP11:   MOV     A,R7
        CLR     0D7H
        CJNE    A,3DH,UP2
        MOV     DPTR,#MONTH_P
        MOVX    A,@DPTR
        SUBB    A,3DA
        CJNE    A,#00H,UP01
        INC     DPTR
        MOV     2FH,34H
        LCALL   T_P_M1
        LCALL   T_P_M2
        LJMP    UP02
T3:     LJMP    T31
UP01:   MOV     MONEY_T,#00H
        MOV     MONEY_T2,#00H
        MOV     2FH,3DH
        LCALL   T_P_M2
UP02:   MOV     2FH,3DH
        MOV     DPTR,#CALL_OUT
        MOVX    A,@DPTR
        MOV     50H,A
        INC     DPTR
        MOVX    A,@DPTR
        ADD     A,#04H
        MOV     51H,A
UP03:   LJMP    T_P5
UP2:    JNC     UP22
UP20:   MOV     R0,50H
        MOV     R1,51H
        LCALL   INCRETE
        MOV     50H,R0
        MOV     51H,R1
        MOV     DPH,R0
        MOV     DPL,R1
        MOVX    A,@DPTR
        SUBB    A,R7
        CJNE    A,#00H,UP30
        LJMP    UP20
UP22:   MOV     R0,50H
        MOV     R1,51H
```

```
            LCALL       INCRETE
            MOV         50H,R0
            MOV         51H,R1
            MOV         DPH,R0
            MOV         DPL,R1
            MOVX        A,@DPTR
            SUBB        A,R7
            CJNE        A,#00H,UP30
            LJMP        UP22
UP30:       MOV         R0,50H
            MOV         R1,51H
            LCALL       DECRETE
            MOV         50H,R0
            MOV         51H,R1
UP31:       MOV         2FH,R7
            MOV         A,R1
            ADD         A,#07H
            MOV         R1,A
            MOV         DPH,R0
            MOV         DPL,R1
            LCALL       T_P_M1
            LCALL       T_P_M2
            LJMP        T_P5
T31:        CJNE        A,#4DH,T_P31
            MOV         KEYMEM,#00H
            LJMP        MENU
T_P31:      LJMP        T_P5
D_PRICE:    MOV         DPTR,#PAGE30
            LCALL       LCDLD
            LCALL       D_READ
            MOV         52H,34H
            MOV         DPTR,#CALL_OUT
            MOVX        A,@DPTR
            MOV         R0,A
            INC         DPTR
            MOVX        A,@DPTR
            MOV         R1,A
            LCALL       DECRETE
            MOV         50H,R0
            MOV         51H,R1
D_10:       MOV         DPH,R0
            MOV         DPL,R1
            MOV         R2,#06H
            MOV         R0,#TH_BUF
D_11:       MOVX        A,@DPTR
            MOV         @R0,A
            INC         R0
            INC         DPTR
            DJNZ        R2,D_11
            MOV         R0,#THBUF
            MOV         R1,#TASCBU
            LCALL       H_T_A
            MOV         R0,#DHBUF
            MOV         R1,#DASCBU
            LCALL       H_T_A
            MOV         CORON,#3AH
            MOV         R0,#96H
            MOV         R1,#TASCBU
            LCALL       DIS_LCD
            MOV         DPH,50H
            MOV         DPL,51H
```

```
        MOV     A,DPL
        ADD     A,#09H
        MOV     DPL,A
        MOV     DPH,50H
        MOV     R2,#02H
        MOV     R0,#BUFF_N
        LCALL   HX_ASC
        MOV     R0,#0D6H
        MOV     R1,#BUFF_N
        MOV     R5,#04H
        LCALL   D_122
        MOV     DPH,50H
        MOV     DPL,51H
        MOV     A,DPL
        ADD     A,#0DH
        MOV     DPL,A
        MOVX    A,@DPTR
        MOV     R6,A
        MOV     R5,A
        INC     DPTR
        MOV     R0,#BUFF_N
        LCALL   D_13
        MOV     R0,#86H
        MOV     R1,#BUFF_N
        LCALL   D_122
D_111:  MOV     KEYMEM,#00H
        LCALL   SCAN
        MOV     A,KEYMEM
        CJNE    A,#2BH,UPP
DOWN:   MOV     R0,50H
        MOV     R1,51H
        LCALL   DECRETE
        MOV     50H,R0
        MOV     51H,R1
        LJMP    D_10
UPP:    CJNE    A,#2DH,D_22
        MOV     R0,50H
        MOV     R1,51H
        LCALL   INCRETE
        MOV     50H,R0
        MOV     51H,R1
        MOV     DPTR,#CALL_OUT
        MOVX    A,@DPTR
        CJNE    A,50H,UPP1
        INC     DPTR
        MOVX    A,@DPTR
        CJNE    A,51H,UPP1
        LJMP    OVD
UPP1:   LJMP    D_10
D_22:   CJNE    A,#4DH,D_21
        MOV     KEYMEM,#00H
        LJMP    MENU
D_21:   LJMP    D_111
EOD:    MOV     DPTR,#PAGE36
        LCALL   LCDLD
        MOV     R2,#0FH
        LCALL   DELAY
        LJMP    MAIN00
OVD:    MOV     DPTR,#PAGE37
        LCALL   LCDLD
        MOV     R2,#0FH
```

```
        LCALL       DELAY
        LJMP        MAIN00
;-----------------HEX TO ASCII FROM RAM TO BUFFER-------------
;R0     =  ADDRESS BUFFER
;R2     =  _BYTE
;DPTR=  START ADDRESS RAM
HX_ASC:     MOVX        A,@DPTR
            ANL         A,#0F0H
            SWAP        A
            ORL         A,#30H
            MOV         @R0,A
            INC         R0
            MOVX        A,@DPTR
            ANL         A,#0FH
            ORL         A,#30H
            MOV         @R0,A
            INC         R0
            INC         DPTR
            DJNZ        R2,HX_ASC
            RET
D_122:      MOV         A,R0
            LCALL       LCDWI
            MOV         A,@R1
            LCALL       LCDWD
            MOV         R2,#01H
            LCALL       DELAY
            INC         R0
            INC         R1
            DJNZ        R5,D_122
            RET
D_13:       MOVX        A,@DPTR
            ANL         A,#0F0H             ;FIRST DIGIT
            SWAP        A
            ANL         A,#30H
            MOV         @R0,A
            INC         R0
            DEC         R6
            CJNE        R6,#00H,D-14
            SJMP        D_15
D-14:       MOVX        A,@DPTR
            ANL         A,#0FH              ;SECOND DIGIT
            ORL         A,#30H
            MOV         @R0,A
            INC         R0
            DEC         R6
            INC         DPTR
            CJNE        R6,#00H,D_13
D_15:       RET
;----------------------INCRETE DPTR--------------------------
INCRETE:    MOV         DPTR,#CALL_OUT
            MOVX        A,@DPTR
            MOV         61H,A
            INC         DPTR
            MOVX        A,@DPTR
            ADD         A,#05H
            MOV         62H,A
            MOV         R2,#15H
I_1:        INC         R1
            MOV         A,R1
            CJNE        A,62H,I_2
            MOV         A,R0
```

```
            CJNE        A,61H,I_2
            LJMP        OVD
I_2:        CJNE        R1,#00H,I_3
            INC         R0
I_3:        DJNZ        R2,I_1
            RET
       ;------------------------DECRETE DPTR--------------------------
DECRETE:    MOV         R2,#15H
D_1:        DEC         R1
            CJNE        R1,#71H,D_12
            CJNE        R0,#00H,D_12
            LJMP        EOD
D_12:       CJNE        R1,#0FFH,D_2
            DEC         R0
D_2:        DJNZ        R2,D_1
            RET
       ;------------------------DISPLAY TOTAL PRICE-------------------
T_P_M1:     MOVX        A,@DPTR             ;DISPLAY MONEY
            MOV         MONEY_T,A
            INC         DPTR
            MOVX        A,@DPTR
            MOV         MONEY_T2,A
            RET
T_P_M2:     MOV         R0,#MONEY_T         ;SHOW MONEY TO LCD
            MOV         R1,#BUFF_N
            MOV         R3,#03H
T_P_M3:     MOV         A,@R0
            ANL         A,#0F0H
            SWAP        A
            ORL         A,#30H
            MOV         @R1,A
            MOV         A,@R0
            ANL         A,#0FH
            ORL         A,#30H
            INC         R1
            MOV         @R1,A
            INC         R0
            INC         R1
            DJNZ        R3,T_P_M3
            MOV         R0,#097H
            MOV         R1,#BUFF_N
            MOV         R5,#04H
            LCALL       T_P4
            MOV         R0,#0C9H
            MOV         R5,#02H
            LCALL       T_P4
            RET
T_P4:       MOV         A,R0
            LCALL       LCDWI
            MOV         A,@R1
            LCALL       LCDWD
            MOV         R2,#01H
            LCALL       DELAY
            INC         R0
            INC         R1
            DJNZ        R5,T_P4
            RET
       ;------------------------WAIT BUSY OR RBT----------------------
WAIT:       MOV         DPTR,#PORTC2
            MOV         A,#00H
            MOVX        @DPTR,A
```

```
WAIT_RB:      LCALL         R_B
              CJNE          R0,#01H,WAIT_RB
RB_CK:        MOV           R7,#00H
LOOP_R_B:     INC           R7
              MOV           R2,#0B3H
              LCALL         DELAY1
              LCALL         R_B
              CJNE          R0,#00H,LOOP_R_B
              CLR           0D7H
              CJNE          R7,#05H,OMPARE
              LJMP          MAIN
COMPARE:      JC            BUSY_CK
              MOV           DPTR,#PAGE11
              LCALL         LCDLD
              MOV           R2,#03H
              LCALL         DELAY
              MOVX          @DPTR,A
              MOV           R2,#01H
              LCALL         DELAY
RBT_CK1:      LCALL         HOOK
              CJNE          R0,#00H,RBT_CK2
              LJMP          MAIN
RBT_CK2:      LCALL         R_B
              CJNE          R0,#01H,RBT_CK1
              MOV           R7,#00H
RBT_CK3:      INC           R7
              MOV           R2,#0B3H
              LCALL         DELAY1
              LCALL         R_B
              CJNE          R0,#00H,RET_CK3
              CLR           0D7H
              CJNE          R7,#05H,RBT_CK4
              LJMP          MAIN
RBT_CK4:      JNC           RBT_CK2
              MOV           DPTR,#PORTC2
              MOV           A,#00H
              MOVX          @DPTR,A
              RET
RBT_CK:       MOV           DPTR,#PORTC2
              MOV           A,#80H
              MOVX          @DPTR,A
              LCALL         HOOK
              CJNE          R0,#00H,RBT_CK
              LJMP          MAIN
PCT:          MOV           DPTR,#PAGE3
              LCALL         LCDLD
              LCALL         HOOK
              CJNE          R0,#00H,PCT
              LJMP          MAIN
      ;------------------------CONTROL ELECTRIC----------------------
CONTROL_E:    MOV           DPTR,#PAGE7
              LCALL         LCDLD
              MOV           R6,#3BH
              MOV           R7,#39H
              LCALL         PLAY1
CONTROL_E1:   LCALL         DTMF_IN
              CJNE          R5,#0BH,CLOSE_E    ;PRESS "#" OPEN
OPEN_E:       LCALL         DTMF_IN
              MOV           DPTR,#ELECTRIC
              MOVX          A,@DPTR
              CJNE          R5,#01H,EO2
```

```
EO1:        SETB    ACC.0
            LJMP    RUN_E
EO2:        CJNE    R5,#02H,EO3
            SETB    ACC.1
            LJMP    RUN_E
EO3:        CJNE    R5,#03H,EO4
            SETB    ACC.2
            LJMP    RUN_E
EO4:        CJNE    R5,#04H,EO5
            SETB    ACC.3
            LJMP    RUN_E
EO5:        CJNE    R5,#05H,EO6
            SETB    ACC.4
            LJMP    RUN_E
EO6:        CJNE    R5,#06H,EO7
            SETB    ACC.5
            LJMP    RUN_E
EO7:        CJNE    R5,#07H,EO8
            SETB    ACC.6
            LJMP    RUN_E
EO8:        CJNE    R5,#08H,CONTROL_E1
            SETB    ACC.7
            LJMP    RUN_E
CLOSE_E:    CJNE    R5,#0CH,AGAIN
            LCALL   DTMF_IN
            MOV     DPTR,#ELECTRIC
            CJNE    R5,#01H,EC2
EC1:        CLR     ACC.0
            LJMP    RUN_E
EC2:        CJNE    R5,#02H,EC3
            CLR     ACC.1
            LJMP    RUN_E
EC3:        CJNE    R5,#03H,EC4
            CLR     ACC.2
            LJMP    RUN_E
EC4:        CJNE    R5,#04H,EC5
            CLR     ACC.3
            LJMP    RUN_E
EC5:        CJNE    R5,#05H,EC6
            CLR     ACC.4
            LJMP    RUN_E
EC6:        CJNE    R5,#06H,EC7
            CJLR    ACC.5
            LJMP    RUN_E
EC7:        CJNE    R5,#07H,EC8
            CLR     ACC.6
            LJMP    RUN_E
EC8:        CJNE    R5,#08H,CLOSE_E
            CLR     ACC.7
            LJMP    RUN_E
RUN_E:      MOV     DPTR,#ELECTRIC
            MOVX    @DPTR,A
            MOV     DPTR,#PORTA3
            MOVX    @DPTR,A
            LJMP    CONTROL_E1
AGAIN:      CJNE    R5,#0AH,AGAIN1
            LJMP    MAIN
AGAIN1:     MOV     R6,#32H
            MOV     R7,#39H
            LCALL   PLAY1
            LJMP    CONTROL_E1
```

```
        ;-----------------------------------RATE MONEY------------------------------
RATE_M:         MOV         R0,#BUFF_N
                INC         R0
                MOV         A,@R0
                CJNE        A,#01H,RATE1
                LJMP        THREE
RATE1:          CJNE        A,#02H,RATE2
                INC         R0
                MOV         A,@R0
                CJNE        A,#03H,THREE
                LJMP        ZERO
RATE2:          CJNE        A,#03H,RATE3
                LJMP        ZERO
RATE3:          CJNE        A,#04H,RATE4
                LJMP        THREE
RATE4:          CJNE        A,#05H,RATE5
                LJMP        THREE
RATE5:          CJNE        A,#06H,RATE6
                LJMP        THREE
RATE6:          CJNE        A,#07H,RATE7
                LJMP        ZERO
RATE7:          CJNE        A,#08H,RATE8
                INC         R0
                MOV         A,@R0
                CJNE        A,#01H,RATE71
                LJMP        THREE
RATE71:         CJNE        A,#02H,RATE72
                LJMP        THREE
RATE72:         CJNE        A,#04H,RATE73
                LJMP        THREE
RATE73:         LJMP        ZERO
RATE8:          CJNE        A,#09H,RATE9
                LJMP        ZERO
RATE9:          CJNE        A,#00H,RATE10
                LJMP        ZERO
RATE10:         LCALL       HOOK
                CJNE        R0,#00H,RATE_M
                LJMP        MAIN
THREE:          MOV         RATE,#03H
                MOV         MONEY_B,#00H
                RET
ZERO:           MOV         RATE,#00H
                MOV         MONEY_B,#00H
                RET
        ;-----------------------------------RATE FAR--------------------------------
RATE_FAR:       MOV         R0,#BUFF_N
                INC         R0
                MOV         A,@R0
                CJNE        A,#03H,F_04
                INC         R0
                MOV         A,@R0
                CJNE        A,#02H,F_034
F_032:          INC         R0
                MOV         A,@R0
                CJNE        A,#05H,S_321
                LJMP        TWELVE
S_321:          CJNE        A,#06H,S_322
                LJMP        TWELVE
S_322:          LJMP        SIX
F_034:          CJNE        A,#04H,F_035
                INC         R0
```

```
            MOV     A,@R0
            CJNE    A,#05H,S_341
            LJMP    NINE
S_341:      CJNE    A,#06H,S_342
            LJMP    NINE
S_342:      CJNE    A,#07H,S-343
            LJMP    SIX
S_343:      LJMP    THREE_F
F_035:      CJNE    A,#05H,F_036
            LJMP    SIX
F_036:      CJNE    A,#06H,F_037
            INC     R0
            MOV     A,@R0
            CJNE    A,#02H,S_361
            LJMP    SIX
S_361:      CJNE    A,#03H,S_362
            LJMP    SIX
S_362:      LJMP    NINE
F_037:      CJNE    A,#07H,F_038
            INC     R0
            MOV     A,@R0
            CJNE    A,#04H,S_371
            LJMP    NINE
S_371:      LJMP    SIX
F_038:      CJNE    A,#08H,F_039
            INC     R0
            MOV     A,@R0
            CJNE    A,#05H,S_381
            LJMP    SIX
S_381:      CJNE    A,#06H,S_382
            LJMP    NINE
S_382:      LJMP    SIX
F_039:      LJMP    TWELVE
F_04:       CJNE    A,#04H,F_05
            INC     R0
            MOV     A,@R0
F_042:      CJNE    A,#02H,F_043
            INC     R0
            MOV     A,@R0
            CJNE    A,#02H,S_421
            LJMP    FIFTEEN
S_421:      CJNE    A,#03H,S_422
            LJMP    FIFTEEN
S_422:      CJNE    A,#08H,S_423
            LJMP    FIFTEEN
S_423:      CJNE    A,#09H,S_424
            LJMP    FIFTEEN
S_424:      LJMP    EIGHTEEN
F_043:      CJNE    A,#03H,F_044
            LJMP    FIFTEEN
F_044:      CJNE    A,#04H,F_045
            LJMP    TWELVE
F_045:      LJMP    FIFTEEN
F_05:       CJNE    A,#05H,F_07
            INC     R0
            MOV     A,@R0
F_053:      CJNE    A,#03H,F_054
            LJMP    EIGHTEEN
F_054:      CJNE    A,#04H,F_055
            INC     R0
            MOV     A,@R0
```

```
          CJNE     A,#05H,S_541
          LJMP     FIFTEEN
S_541:    CJNE     A,#06H,S_542
          LJMP     FIFTEEN
S_542:    LJMP     EIGHTEEN
F_055:    CJNE     A,#05H,F_056
          INC      R0
          MOV      A,@R0
          CJNE     A,#02H,S_551
          LJMP     TWELVE
S_551:    CJNE     A,#03H,S_552
          LJMP     TWELVE
S_552:    CJNE     A,#07H,S_553
          LJMP     TWELVE
S_553:    LJMP     FIFTEEN
F_056:    INC      R
          MOV      A,@R0
          CJNE     A,#04H,S_561
          LJMP     NINE
S_561:    CJNE     A,#05H,S_562
          LJMP     NINE
S_562     LJMP     TWELVE
F_07:     INC      R0
          MOV      A,@R0
          CJNE     A,#07H,F_07S
          INC      R0
          MOV      A,@R0
          CJNE     A,#05H,S_771
          LJMP     FIFTEEN
S_771:    CJNE     A,#06H,F_07S
          LJMP     FIFTEEN
F_07S:    LJMP     EIGHTEEN
THREE_F:  MOV      RATE,#03H
          RET
SIX:      MOV      RATE,#06H
          RET
NINE:     MOV      RATE,#09H
          RET
TWELVE:   MOV      RATE,#0CH
          RET
FIFTEEN:  MOV      RATE,#0FH
          RET
EIGHTEEN: MOV      RATE,#12H
          RET
;****************************SUB PROGRAM***************************
     ;-------------------READ DETECT SIGNAL-----------------------
HOOK:     MOV      DPTR,#PORTA2
          MOVX     A,@DPTR
          ANL      A,#10H          ;HOOK ONLY
          MOV      R0,A
          RET
RING:     MOV      DPTR,#PORTA2
          MOVX     A,@DPTR
          ANL      A,#08H          ;RING ONLY
          MOV      R0,A
          RET
COUNT:    MOV      R2,#01H
          LCALL    DELAY
          MOV      DPTR,#PORTA2
          MOVX     A,@DPTR
          ANL      A,#04H          ;COUNTER ONLY
```

```
        MOV     R0,A
        RET
DIAL:   MOV     DPTR,#PORTA2
        MOVX    A,@DPTR
        ANL     A,#02H          ;DIAL ONLY
        MOV     R0,A
        RET
R_B:    MOV     DPTR,#PORTA2
        MOVX    A,@DPTR
        ANL     A,#01H          ;RBT&BUSY ONLY
        MOV     R0,A
        RET
CLEAR:  MOV     DPTR,#PORTC2
        MOV     A,#02H          ;RING COUNT = RESET
        MOVX    @DPTR,A
        MOV     R2,#01H
        LCALL   DELAY
        MOV     DPTR,#PORTC2
        MOV     A,#00H          ;RING COUNT = NORMAL
        MOVX    @DPTR,A
        MOV     R2,#01H
        LCALL   DELAY
        RET
;--------------------------DTMF CHECK---------------------------
DTMF_OUT:       MOV     DPTR,#PORTB2
        MOVX    A,@DPTR
        ANL     A,#10H
        MOV     R0,A
        CJNE    R0,#10H,HK_DTMF
        MOV     DPTR,#PORTB2
        MOVX    A,@DPTR
        MOV     R2,#01H
        LCALL   DELAY
        ANL     A,#0FH
        CJNE    A,#0AH,DTMF_
        MOV     A,#00H
DTMF_:  MOV     R5,A
DTMF1:  MOV     DPTR,#PORTB2
        MOVX    A,@DPTR
        ANL     A,#10H
        MOV     R0,A
        CJNE    R0,#00H,DTMF1
        RET
HK_DTMF:        LCALL   HOOK
        CJNE    R0,#00H,DTMF_OUT
        LJMP    MAIN
DTMF_IN:        MOV     R7,#00H
DTMF_IN1:       MOV     DPTR,#PORTB2
        MOVX    A,@DPTR
        ANL     A,#10H
        MOV     R0,A
        CJNE    R0,#10H,TIMER
        MOV     DPTR,#PORTB2
        MOVX    A,@DPTR
        MOV     R2,#01H
        LCALL   DELAY
        ANL     A,#0FH
        MOV     R5,A
DTMF11: MOV     DPTR,#PORTB2
        MOVX    A,@DPTR
        ANL     A,#10H
```

```
                MOV         R0,A
                CJNE        R0,#00H,DTMF11
                RET
TIMER:          MOV         R2,#0B3H    ;0.1 SEC
                LCALL       DELAY1
                INC         R7
                CJNE        R7,#34H,DTMF_IN1;5.32 SEC
                LJMP        MAIN
;--------------------SET INITIAL TIME/DATE--------------------
;------------------------SETTING DATE------------------------
TD_INIT:        MOV         DPTR,#PAGE24
                LCALL       LCDLD
                LCALL       T_READ
                MOV         CORON,#3AH          ;DISPLAY TIME
                MOV         R0,#0D4H
                MOV         R1,#TASCBU
                LCALL       DIS_LCD
                LCALL       D_READ
                MOV         CORON,#2FH          ;DISPLAY DATE
                MOV         R0,#94H
                MOV         R1,#DASCBU
                LCALL       DIS_LCD
                MOV         R2,#03H             ;DELAY KEY & DISPLAY
                LCALL       DELAY
                MOV         A,#00001111B
                LCALL       LCDWI
                MOV         R2,#01H
                LCALL       DELAY
DATE:           MOV         A,#94H              ;SET DATE
                LCALL       LCDWI
                MOV         KEYMEM,#00H         ;CLEAR KEYMEM
                LCALL       SCAN
                MOV         A,KEYMEM
                CJNE        A,#7FH,SD0
                AJMP        EXIT_D
SD0:            CJNE        A,#30H,SD1          ;CHECK 0
                SJMP        SD_E
SD1:            CJNE        A,#31H,SD2          ;CHECK 1
                SJMP        SD_E
SD2:            CJNE        A,#32H,SD3          ;CHECK 2
                SJMP        SD_E
SD3:            CJNE        A,#33H,DATE         ;CHECK 3
SD_E:           MOV         DASCBU,A
                LCALL       LCDWD
                MOV         R2,#02H             ;DELAY
                LCALL       DELAY
SD_2:           MOV         KEYMEM,#00H
                MOV         A,#95H
                LCALL       LCDWI
SD_21:          LCALL       SCAN
                MOV         A,KEYMEM
                CJNE        A,#7FH,SD_22
                AJMP        EXIT_D
SD_22:          MOV         R1,#DASCBU
                CJNE        @R1,#33H,SD_24      ;CHECK DASCBU =3 ?
                CJNE        A,#30H,SD_23        ;IF YES
                SJMP        SD_E1
SD_23:          CJNE        A,#31H,SD_21
                SJMP        SD_E1
SD_24:          MOV         A,KEYMEM            ;IF NO
                MOV         R3,#0AH
```

```
            MOV         R4,#30H
SD6_1:      CJNE        A,#04H,SD6_2
            SJMP        SD_E1
SD6_2:      INC         R4
            DJNZ        R3,SD6_1
            SJMP        SD_2
SD_E1:      MOV         DASCBU+1,A
            LCALL       LCDWD
            MOV         R2,#02H
            LCALL       DELAY
      ;--------------------------SET MOUNT---------------------------
MON:        MOV         A,#97H
            LCALL       LCDWI
            MOV         KEYMEM,#00H
            LCALL       SCAN
            MOV         A,KEYMEM
            CJNE        A,#7FH,MON_0
            AJMP        EXIT_D
MON_0:      CJNE        A,#30H,MON1
            SJMP        MON_E
MON_1:      CJNE        A,#31H,MON
MON_E:      MOV         DASCBU+2,A
            LCALL       LCDWD
            MOV         R2,#02H
            LCALL       DELAY
MON2:       MOV         KEYMEM,#00H
            MOV         A,#98H
            LCALL       LCDWI
MON2_0:     LCALL       SCAN
            MOV         A,KEYMEM
            CJNE        A,#7FH,MON2_1
            AJMP        EXIT_D
MON2_1:     MOV         R1,#DASCBU+2
            CJNE        @R1,#31H,MON2_2
            CJNE        A,#30H,MON2_21
            SJMP        MON2_E
MON2_21:    CJNE        A,#31H,MON2_24
            SJMP        MON2_E
MON2_24:    CJNE        A,#32H,MON2_0
            SJMP        MON2_E
MON2_2:     MOV         A,KEYMEM
            MOV         R3,#0AH
            MOV         R4,#30H
MON2_22:    CJNE        A,04H,MON2_23
            SJMP        MON2_E
MON2_23:    INC         R4
            DJNZ        R3,MON2_22
            SJMP        MON2
MON2_E:     MOV         DASCBU+3,A
            LCALL       LCDWD
            MOV         R2,#02H
            LCALL       DELAY
      ;--------------------------SET YEAR----------------------------
YEAR:       MOV         R0,#DASCBU+4
            MOV         R1,#9AH
            MOV         R5,#02H
YEAR_:      MOV         KEYMEM,#00H
            MOV         A,R1
            LCALL       LCDWI
            LCALL       SCAN
            MOV         A,KEYMEM
```

```
                CJNE        A,#7FH,YEAR0
                AJMP        EXIT_D
YEAR0:          MOV         R3,#0AH
                MOV         R4,#30H
YEAR1:          CJNE        A,04H,YEAR2
                SJMP        YEAR_E
YEAR2:          INC         R4
                DJNZ        R3,YEAR1
                SJMP        YEAR_
YEAR_E:         MOV         @R0,A
                LCALL       LCDWD
                INC         R0
                INC         R1
                MOV         R2,#02H
                LCALL       DELAY
                DJNZ        R5,YEAR_
                AJMP        DATE
EXIT_D:         MOV         A,#00001110B
                LCALL       LCDWI
                MOV         R0,#DASCBU
                MOV         R1,DHBUF
                LCALL       A_T_H
                LCALL       D_WRITE
        ;---------------------------SET TIME--------------------------------
T_SET:          MOV         A,#00001111B
                LCALL       LCDWI
                MOV         R2,#01H
                LCALL       DELAY
HOUR:           MOV         A,#0D4H
                LCALL       LCDWI
                MOV         KEYMEM,#00H           ;CLEAR KEYMEM
                LCALL       SCAN                  ;SET HOUR
                MOV         A,KEYMEM
                CJNE        A,#7FH,H0
                AJMP        EXIT_T
H0:             CJNE        A,#30H,H1             ;CHECK 0
                SJMP        H1_E
H1:             CJNE        A,#31H,H2             ;CHECK 1
                SJMP        H1_E
H2:             CJNE        A,#32H,HOUR
H1_E:           MOV         TASCBU,A
                LCALL       LCDWD
                MOV         R2,#02H               ;DELAY KEY
                LCALL       DELAY
HOUR2:          MOV         KEYMEM,#00H           ;CLEAR KEYMEM
                MOV         A,#0D5H
                LCALL       LCDWI
HOU:            LCALL       SCAN
                MOV         A,KEYMEM
                CJNE        A,#7FH,HU
                AJMP        EXIT_T
HU:             MOV         R1,#TASCBU
                CJNE        @R1,#32H,H2_1
                CJNE        A,#30H,HU1
                SJMP        H2_E
HU1:            CJNE        A,#31H,HU2
                SJMP        H2_E
HU2:            CJNE        A,#32H,HU3
                SJMP        H2_E
HU3:            CJNE        A,#33H,HOU
                SJMP        H2_E
```

```
H2_1:       MOV     A,KEYMEM
            MOV     R3,#0AH
            MOV     R4,#30H
H2_2:       CJNE    A,04H,HO1
            SJMP    H2_E
HO1:        INC     R4
            DJNZ    R3,H2_2
            SJMP    HOUR2
H2_E:       MOV     TASCBU+1,A
            LCALL   LCDWD
            MOV     R2,#02H
            LCALL   DELAY
MIN:        MOV     R0,#0D7H
            MOV     R1,#TASCBU+2
            MOV     R5,#02H
M_S:        MOV     A,R0
            LCALL   LCDWI
            MOV     KEYMEM,#00H
            LCALL   SCAN
            MOV     A,KEYMEM
            CJNE    A,#7FH,M_S0
            AJMP    EXIT_T
M_S0:       MOV     R3,#06H
            MOV     R4,#30H
M_S1:       CJNE    A,04H,M_S2
            SJMP    M_SE
M_S2:       INC     R4
            DJNZ    R3,M_S1
            SJMP    M_S
M_SE:       MOV     @R1,A
            LCALL   LCDWD
            MOV     R2,#02H
            LCALL   DELAY
            INC     R0
            INC     R1
M_S3:       MOV     A,R0
            LCALL   LCDWI
            MOV     KEYMEM,#00H
            LCALL   SCAN
            MOV     A,KEYMEM
            CJNE    A,#7FH,,M_S4
            AJMP    EXIT_T
M_S4:       MOV     R3,#0AH
            MOV     R4,#30H
M_S5:       CJNE    A,04H,M_S6
            SJMP    M_SE1
M_S6:       INC     R4
            DJNZ    R3,M_S5
            SJMP    M_S3
M_SE1:      MOV     @R1,A
            LCALL   LCDWD
            MOV     R2,#02H
            LCALL   DELAY
            INC     R0
            INC     R0
            INC     R1
            DJNZ    R5,M_S
            AJMP    HOUR
EXIT_T:     MOV     A,#00001110B
            LCALL   LCDWI
            MOV     R0,#TASCBU
```

```
        MOV        R1,#THBUF
        LCALL      A_T_H
        LCALL      T_WRITE
        LJMP       EE_
;------------------------WRITE TIME----------------------------
T_WRITE: MOV       R0,#8EH
        MOV        R1,#00H
        LCALL      RTC_W
        MOV        R0,#80H
        MOV        R1,THBUF+2
        LCALL      RTC_W
        MOV        R0,#82H
        MOV        R1,THBUF+1
        LCALL      RTC_W
        MOV        R0,#84H
        MOV        R1,THBUF
        LCALL      RTC_W
        MOV        R0,#8EH
        MOV        R1,#80H
        LCALL      RTC_W
        RET
;----------------------------WRITE DATE------------------------
D_WRITE: MOV       R0,#8EH
        MOV        R1,#0H
        LCALL      RTC_W
        MOV        R0,#8CH
        MOV        R1,DHBUF+2
        LCALL      RTC_W
        MOV        R0,#88H
        MOV        R1,DHBUF+2
        LCALL      RTC_W
        MOV        R0,#86H
        MOV        R1,DHBUF
        LCALL      RTC_W
        MOV        R0,#8EH
        MOV        R1,#80H
        LCALL      RTC_W
        RET
;-------------------------READ TIME----------------------------
T_READ:  MOV       R0,#81H
        LCALL      RTC_R
        MOV        THBUF+2,R1
        MOV        R0,#83H
        LCALL      RTC_R
        MOV        THBUF+1,R1
        MOV        R0,#85H
        LCALL      RTC_R
        MOV        THBUF,R1
        MOV        R0,#THBUF
        MOV        R1,#TASCBU
        LCALL      H_T_A
        LCALL      T_DELAY
        RET
;---------------------------READ DATE--------------------------
D_READ:  MOV       R0,#8DH
        LCALL      RTC_R
        MOV        DHBUF+2,R1
        MOV        R0,#89H
        LCALL      RTC_R
        MOV        DHBUF+1,R1
        MOV        R0,#87H
```

```
        LCALL   RTC_R
        MOV     DHBUF,R1
        MOV     R0,#DHBUF
        MOV     R1,DASCBU
        LCALL   H_T_A
        LCALL   TDELAY
        RET
;-------------------------WRITE COMMAND&DATA-----------------
RTC_W:  CLR     T_CLK
        LCALL   T_DELAY
        SETB    T_RST
        LCALL   TDELAY
        MOV     A,R0
        LCALL   T_W_8
        MOV     A,R1
        LCALL   T_W_8
        CLR     T_RST
        LCALL   TDELAY
        RET
;-----------------------SEND DATA 8 BIT TO RTC-----------------
T_W_8:  MOV     R3,#08H
T_W_8_1:        RRC     A
        MOV     T_DATA,C
        SETB    T_CLK
        LCALL   TDELAY
        CLR     T_CLK
        LCALL   T_DELAY
        DJNZ    R3,T_W_8_1
        RET
;-----------------------READ DATA FROM RTC---------------------
RTC_R:  CLR     T_CLK
        LCALL   TDELAY
        SETB    T_RST
        LCALL   TDELAY
        MOV     A,R0
        LCALL   T_W_8
        MOV     R3,#8
        CLR     A
READ:   CLR     T_CLK
        LCALL   T_DELAY
        MOV     C,T_DATA
        RRC     A
        SETB    T_CLK
        LCALL   TDELAY
        DJNZ    R3,READ
        MOV     R1,A
        CLR     T_RST
        LCALL   TDELAY
        RET
;--------------------CONVERSE HEX TO ASCII--------------------
H_T_A:  MOV     R3,#3
CON1:   MOV     A,@R0
        ANL     A,#0F0H
        SWAP    A
        ORL     A,#30H
        MOV     @R1,A
        MOV     A,@R0
        ANL     A,#0FH
        ORL     A,#30H
        INC     R1
        MOV     @R1,A
```

```
MESS3:      CJNE      A,#33H,MESS1
MESS_E:     MOV       A,#01H
            LCALL     LCDWI
            LJMP      EE_
         ;-------------SCAN KEY ON PORTC  (PCH =OUT  CPL = IN)----------
SCAN:       PUSH      00H
            PUSH      01H
            MOV       R0,#01111111B       ;START COLUMN 1
            MOV       R1,#04H
SCAN1:      MOV       DPTR,#PORTC1
            MOV       A,R0
            MOV       @DPTR,A
            LCALL     KDELAY
INKEY:      MOV       DPTR,#PORTC1
            MOVX      A,@DPTR
            ANL       A,#0FH
            CJNE      A,#0FH,KEY
            MOV       A,R0
            RR        A
            MOV       R0,A
            DJNZ      R1,SCAN1
            POP       01H
            POP       00H
            RET
KEY:        MOV       KEYBUF,A            ;SAVE AT 48H
            MOVX      A,@DPTR
            ANL       A,#0FH
            CJNE      A,48H,KEY
            LCALL     CHECK
            POP       01H
            POP       00H
            RET
         ;-------------------------CHECK KEY-------------------------------
CHECK:      MOVX      A,@DPTR
            ANL       A,#0FH
ROW1:       CJNE      A,#07H,ROW2  ;CHK ROW1
            MOV       DPTR,#ASCII
            SJMP      COL1
ROW2:       CJNE      A,#0BH,ROW3  ;CHK ROW2
            MOV       DPTR,#ASCII+4
            SJMP      COL1
ROW3:       CJNE      A,#0DH,ROW4  ;CHK ROW3
            MOV       DPTR,#ASCII+8
            SJMP      COL1
ROW4:       CJNE      A,#0EH,BACK  ;CHK ROW4
            MOV       DPTR,#ASCII+12
            SJMP      COL1
BACK:       RET
         ;-------------------------CHECK COLUMN-----------------------
COL1:       MOV       A,R0
            ANL       A,#0F0H
            CJNE      A,#70H,COL2
            CLR       A
            MOVC      A,@A+DPTR
            MOV       KEYMEM,A
            RET
COL2:       CJNE      A,#0B0H,COL3
            CLR       A
            INC       DPTR
            MOVC      A,@A+DPTR
            MOV       KEYMEM,A
```

```
                RET
COL :           CJNE        A,#0D0H,COL4
                CLR         A
                INC         DPTR
                INC         DPTR
                MOVC        A,@A+DPTR
                MOV         KEYMEM,A
                RET
COL4:           CJNE        A,#0E0H,CHECK
                INC         DPTR
                INC         DPTR
                INC         DPTR
                CLR         A
                MOVC        A,@A+DPTR
                MOV         KEYMEM,A
                RET
        ;------------------------CHECK KEY MENU------------------------
MENU:           MOV         A,KEYMEM
                CJNE        A,#4DH,EE_
M_:             MOV         DPTR,#PAGE19
                LCALL       LCDLD
M1:             LCALL       SCAN
                MOV         A,KEYMEM
                CJNE        A,#31H,K2
                LJMP        SETMODE
K2:             CJNE        A,#32H,SUM
                LJMP        MESS
SUM:            CJNE        A,#2DH,M1
                MOV         DPTR,#PAGE20
                LCALL       LCDLD
M2:             LCALL       SCAN
                MOV         A,KEYMEM
                CJNE        A,#2BH,M2_1
                MOV         KEYMEM,#00H
                MOV         R2,#03H
                LCALL       DELAY
                SJMP        M_
M2_1:           CJNE        A,#33H,M2_2
                LJMP        BILL_M
M2_2:           CJNE        A,#34H,M2_5
                LJMP        ELEC
M2_5:           CJNE        A,#35H,,M2
EE_:            MOV         A,#01H
                LCALL       LCDWI
                MOV         A,#00001110B
                LCALL       LCDWI
                MOV         DPTR,#PAGE17
                LCALL       LCDLD
                LJMP        EEM
        ;-------------------------SETTING MODE-------------------------
SETMODE:        MOV         DPTR,#PAGE21
                LCALL       LCDLD
                MOV         R2,#03H
                LCALL       DELAY
SET0:           MOV         KEYMEM,#00H
                LCALL       SCAN
                MOV         A,KEYMEM
                CJNE        A,#31H,SET1
                LJMP        TD_INIT
SET1:           CJNE        A,#32H,SET2
                LJMP        SETPASS
```

```
SET2:        CJNE       A,#33H,SET3
             LJMP       SETRING
SET3:        CJNE       A,#4DH,SET0
             LJMP       MENU
     ;-----------------------SET PASSWORD-------------------------
SETPASS:     MOV        DPTR,#PAGE25
             LCALL      LCDLD
             MOV        R2,#03H
             LCALL      DELAY
             MOV        A,#00001111B
             LCALL      LCDWI
             MOV        R2,#01H
             LCALL      DELAY
             MOV        A,#0C4H              ;SET ADDRESS CURSOR
             LCALL      LCDWI
             MOV        R1,#0C4H
             MOV        R0,#PASSMEM1
             MOV        R5,#04H
SETPASS1:    MOV        KEYMEM,#00H
             LCALL      SCAN
             MOV        A,KEYMEM
             CJNE       A,#4DHM,SETPASS2     ;NOT MENU
             SJMP       SETPASS1
SETPASS2:    CJNE       A,#2BH,SETPASS3      ;NOT +
             SJMP       SETPASS1
SETPASS3:    CJNE       A,#2DH,SETPASS4      ;NOT -
             SJMP       SETPASS1
SETPASS4:    CJNE       A,#7FH,SETPASS5      ;NOT ENTER
             SJMP       SETPASS1
SETPASS5:    CJNE       A,#00H,SET_P1
             SJMP       SETPASS1
SET_P1:      ANL        A,#0FH
             MOV        @R0,A
             MOV        R2,#02H
             LCALL      DELAY
             INC        R0
             INC        R1
             MOV        A,R1
             LCALL      LCDWI
             DJNZ       R5,SETPASS1
;----------------------COMPARE PASSWORD--------------------------------
COMPARE_P:   MOV        DPTR,#PASSWD
             MOVX       A,@DPTR
             CJNE       A,4BH,COMP_F
             INC        DPTR
             MOVX       A,@DPTR
             CJNE       A,4CH,COMP_F
             INC        DPTR
             MOVX       A,@DPTR
             CJNE       A,4DH,COMP_F
             INC        DPTR
             MOVX       A,@DPTR
             CJNE       A,4EH,COMP_F
             SJMP       COMP_T
COMP_F:      MOV        DPTR,#PAGE27
             LCALL      LCDLD
             MOV        R2,#03H
             LCALL      DELAY
             LJMP       EE_
     ;----------------------ENTER NEW PASSWORD----------------------
COMP_T:      MOV        A,#94H
```

```
            LCALL       LCDWI
            MOV         R1,#94H
            MOV         R0,#PASSMEM1
            MOV         R6,#02H
SETPASS11:  MOV         R5,#04H
SETPASS6:   MOV         KEYMEM,#00H
            LCALL       SCAN
            MOV         A,KEYMEM
            CJNE        A,#4DH,SETPASS7
            SJMP        SETPASS6
SETPASS7:   CJNE        A,#2BH,SETPASS8
            SJMP        SETPASS6
SETPASS8:   CJNE        A,#2DH,SETPASS9
            SJMP        SETPASS6
SETPASS9:   CJNE        A,#7FH,SETPASS10
            SJMP        SETPASS6
SETPASS10:  CJNE        A,#00H,SET_P2
            SJMP        SETPASS6
SET_P2:     ANL         A,#0FH
            MOV         @R0,A
            MOV         R2,#02H
            LCALL       DELAY
            INC         R0
            INC         R1
            MOV         A,R1
            LCALL       LCDWI
            DJNZ        R5,SETPASS6
            MOV         R0,#PASSMEM2
            MOV         A,#0D6H
            LCALL       LCDWI
            MOV         R1,#0D6H
            DJNZ        R6,SETPASS11
;----------------------------COMPARE NEW PASSWORD----------------------
COMPARE_P1: MOV         A,4FH
            CJNE        A,4BH,COMP_F1
            MOV         A,50H
            CJNE        A,4CH,COMP_F1
            MOV         A,51H
            CJNE        A,4DH,COMP_F1
            MOV         A,52H
            CJNE        A,4EH,COMP_F1
            SJMP        COMP_T1
COMP_F1:    MOV         DPTR,#PAGE27
            LCALL       LCDLD
            MOV         R2,#03H
            LCALL       DELAY
            LJMP        EE_
;------------------SEND NEW PASSWORD TO RAM--------------------
COMP_T1:    MOV         DPTR,#PAGE28
            LCALL       LCDLD
            MOV         R2,#03H
            LCALL       DELAY
            MOV         R3,#04H
            MOV         DPTR,#PASSWD
            MOV         R0,#PASSMEM1
SEND_PASS:  MOV         A,@R0
            MOVX        @DPTR,A
            INC         DPTR
            INC         R0
            DJNZ        R3,SEND_PASS
            LJMP        EE_
```

```
        ;-----------------------------------SET RING----------------------------
SETRING:        MOV          DPTR,#PAGE26
                LCALL        LCDLD
                MOV          R2,#03H
                LCALL        DELAY
                MOV          A,#00001111B
                LCALL        LCDWI
                MOV          R2,#01H
                LCALL        DELAY
                MOV          A,#94H
                LCALL        LCDWI
                MOV          DPTR,#R_COUNT
                MOVX         A,@DPTR
                ORL          A,#30H
                LCALL        LCDWD
SETRING1:       MOV          A,#0D4H
                LCALL        LCDWI
                MOV          KEYMEM,#00H
                LCALL        SCAN
                MOV          A,KEYMEM
                CJNE         A,#4DH,SETRING2
                SJMP         SETRING1
SETRING2:       CJNE         A,#2BH,SETRING3
                SJMP         SETRING1
SETRING3:       CJNE         A,#2DH,SETRING4
                SJMP         SETRING1
SETRING4:       CJNE         A,#00H,SETRING5
                SJMP         SETRING1
SETRING5:       CJNE         A,#23H,SETRING6
                SJMP         SETRING1
SETRING6:       CJNE         A,#2AH,SETRING7
                SJMP         SETRING1
SETRING7:       CJNE         A,#7FH,SET_R1
                LJMP         EE_
SET_R1:         ANL          A,#0FH
                MOV          DPTR,#R_COUNT
                MOVX         @DPTR,A
                MOV          A,KEYMEM
                LCALL        LCDWD
                MOV          R2,#02H
                LCALL        DELAY
                SJMP         SETRING1
        ;-------------------------------CONTROL ELECTRIC----------------------
ELEC:           MOV          DPTR,#PAGE31
                LCALL        LCDLD
                MOV          R2,#01H
                LCALL        DELAY
                MOV          A,#0DAH
                LCALL        LCDWI
                MOV          A,#00001111B
                LCALL        LCDWI
ELEC1:          MOV          KEYMEM,#00H
                LCALL        SCAN
                MOV          R3,KEYMEM
                CJNE         R3,#2AH,ELEC_E
                AJMP         ELECOFF
ELEC_E:         CJNE         R3,#7FH,ELECON
                LJMP         EE_
ELECON:         CJNE         R3,#23H,ELEC1
        ;-------------------------ON ELECTRIC---------------------------
                MOV          A,#6FH
```

```
            LCALL       LCDWD
            MOV         A,#6EH
            LCALL       LCDWD
ON:         MOV         KEYMEM,#00H
            LCALL       SCAN
            MOV         R3,KEYMEM
            MOV         DPTR,ELECTRIC
            MOVX        A,@DPTR
            CJNE        R3,#31H,ON2
            SETB        0E0H
            SJMP        ON_E
ON2:        CJNE        R3,#32H,ON3
            SETB        0E1H
            SJMP        ON_E
ON3:        CJNE        R3,#33H,ON4
            SETB        0E2H
            SJMP        ON_E
ON4:        CJNE        R3,#34H,ON5
            SETB        0E3H
            SJMP        ON_E
ON5:        CJNE        R3,#35H,ON6
            SETB        0E4H
            SJMP        ON_E
ON6:        CJNE        R3,#36H,ON7
            SETB        0E5H
            SJMP        ON_E
ON7:        CJNE        R3,#37H,ON8
            SETB        0E6H
            SJMP        ON_E
ON8;        CJNE        R3,#38H,ON_S
            SETB        0E7H
            SJMP        ON_E
ON_S:       AJMP        ON
ON_E:       MOV         DPTR,#ELECTRIC
            MOVX        @DPTR,A
            MOV         DPTR,#PORTA3
            MOVX        @DPTR,A
            MOV         A,R3
            LCALL       LCDWD
            MOV         R2,#01H
            LCALL       DELAY
            LJMP        ELEC
    ;-------------------------OFF ELECTRIC----------------------------
ELECOFF:    MOV         A,#6FH
            LCALL       LCDWD
            MOV         A,#66H
            LCALL       LCDWD
            MOV         A,#66H
            LCALL       LCDWD
OFF:        MOV         KEYMEM,#00H
            LCALL       SCAN
            MOV         R3,KEYMEM
            MOV         DPTR,#ELECTRIC
            MOVX        A,@DPTR
            CJNE        R3,#31H,OFF2
            CLR         0E0H
            SJMP        OFF_E
OFF2:       CJNE        R3,#32H,OFF3
            CLR         0E1H
            SJMP        OFF_E
OFF3:       CJNE        R3,#33H,OFF4
```

```
        CLR     0E2H
        SJMP    OFF_E
OFF4:   CJNE    R3,#34H,OFF5
        CLR     0E3H
        SJMP    OFF_E
OFF5:   CJNE    R3,#35H,OFF6
        CLR     0E4H
        SJMP    OFF_E
OFF6:   CJNE    R3,#36H,OFF7
        CLR     0E5H
        SJMP    OFF_E
OFF7:   CJNE    R3,#37H,OFF8
        CLR     0E6H
        SJMP    OFF_E
OFF8:   CJNE    R3,#38H,OFF_S
        CLR     0E7H
        SJMP    OFF_E
OFF_S:  LJMP    OFF
OFF_E:  MOV     DPTR,#ELECTRIC
        MOVX    @DPTR,A
        MOV     DPTR,PORTA3
        MOVX    @DPTR,A
        MOV     A,R3
        LCALL   LCDWD
        MOV     R2,#01H
        LCALL   DELAY
        LJMP    ELEC
;--------------------------LCDDIS----------------------------------
LCDLD:  MOV     A,#80H
        LCALL   LCDLDS
        MOV     A,#0C0H
        LCALL   LCDLDS
        MOV     A,#90H
        LCALL   LCDLDS
        MOV     A,#0D0H
        LCALL   LCDLDS
        RET
LCDLDS: PUSH    DPH
        PUSH    DPL
        LCALL   LCDWI
        POP     DPH
        POP     DPL
        MOV     R2,#16H
LCDLDS1: CLR    A
        MOVC    A,@A+DPTR
        PUSH    DPH
        PUSH    DPL
        LCALL   LCDWD
        POP     DPL
        POP     DPH
        INC     DPTR
        DJNZ    R2,LCDLDS1
        RET
;-------------------------LCDWI--------------------------------
LCDWI:  MOV     DPTR,#PORTA1
        MOVX    @DPTR,A
        MOV     DPTR,#PORTB1
        MOVX    A,@DPTR
        CLR     ACC.0
        CLR     ACC.1
        CLR     ACC.2
```

```
            MOVX        @DPTR,A
            SETB        ACC.2
            MOVX        @DPTR,A
            SETB        ACC.2
            MOVX        @DPTR,A
            MOV         A,#0
LCDWI1:     DEC         A
            JNZ         LCDWI1
            RET
      ;-------------------------LCDWD-------------------------------
LCDWD:      MOV         DPTR,#PORTA1
            MOVX        @DPTR,A
            MOV         DPTR,#PORTB1
            MOVX        @A,DPTR
            SETB        ACC.0
            CLR         ACC.1
            CLR         ACC.2
            MOVX        @DPTR,A
            SETB        ACC.2
            MOVX        @DPTR,A
            CLR         ACC.2
            MOVX        @DPTR,A
            MOV         A,#0
LCDWD1:     DEC         A
            JNZ         LCDWD1
            RET
      ;-------------------PASSWORD CHECK----------------------------
PWD:        MOV         DPTR,#PAGE12
            LCALL       LCDLD
            MOV         R6,#00H
            MOV         R1,#BUFF_N
PWD1:       LCALL       DTMF_IN
            LCALL       BUFF_PW
            INC         R6
            CJNE        R6,#04H,PWD1
            MOV         R1,#BUFF_N
            MOV         DPTR,#PASSWD
PWD2:       MOVX        A,@DPTR
            SUBB        A,@R1
            CJNE        A,#00H,PWD_ON
            INC         DPTR
            INC         R1
            DEC         R6
            CJNE        R6,#00H,PWD2
            RET
PWD_NO:     MOV         DPTR,#PAGE13
            LCALL       LCDLD
            MOV         R6,#32H
            MOV         R7,#39H
            LCALL       PLAY1
            LJMP        PWD
BUFF_PW:    MOV         A,R5
            MOV         @R1,A
            INC         R1
            RET
      ;-------------------------DELAY TIME--------------------------
            ;-----------------------20MS----------------------------
-
TDELAY:     NOP
            RET
KDELAY:     NOP
```

```
            NOP
            MOV         R4,#00H
D:          DJNZ        R4,$
            RET
            ;---------------0.143SEC - 36.62 SEC--------------------
DELAY:      MOV         R3,#00H
LOOP1:      MOV         R4,#00H
            DJNZ        R4,$
            DJNZ        R3,LOOP1
            DJNZ        R2,DELAY
            RET
DELAY1:     MOV         R3,#00H
            DJNZ        R3,$
            DJNZ        P2,DELAY1
            RET
      ;---------------------PROGRAM ANSWER RECORD--------------------
            ;--------------------PLAY PROGRAM-----------------------
            ;ADD ISD #1 =39H , #2 =35H , #3 =2DH , #4 =1DH
            ;R6 = ADD ,R7 = ISD
            ;-----------------------PLAY ISD # 1--------------------
PLAY1:      MOV         A,#3FH             ;SET INITIAL PORT
            LCALL       SET_P
            MOV         A,R6               ;SET ADDRESS #5
            LCALL       SET_ADD
            MOV         A,#3DH
            LCALL       SET_P
            MOV         A,R7
            LCALL       SET_P
            MOV         A,#3DH
            LCALL       SET_P
            LCALL       CK_EOM1
            RET
            ;-------------------PLAY 15 MESSAGE----------------------
PLAY2:      MOV         A,#3FH             ;SET INNITIAL FOR PLAY BACK
            LCALL       SET_P
            MOV         A,R6               ;SET ADDRESS
            LCALL       SET_ADD
            MOV         A,#3DH
            LCALL       SET_P
            MOV         A,R7
            LCALL       SET_P
            MOV         A,#3DH
            LCALL       SET_P
            LCALL       CK_EOM2
            RET
SET_ADD:    MOV         DPTR,#PORTA4
            MOVX        @DPTR,A
            RET
SET_P:      MOV         DPTR,#PORTB4
            MOVX        @DPTR,A
            MOV         R2,#04H
            LCALL       DELAY
            RET
            ;--------------SUB PLAYBACK FOR CHK MESSAGE--------------
P_TODAY:    LCALL       D_READ
            MOV         A,#01H
            LCALL       LCDWI
            MOV         DPTR,#REC
            MOVX        A,@DPTR
            MOV         R6,A
            INC         DPTR
```

```
          MOVX    A,@DPTR
          MOV     R7,A
          INC     R7
          INC     DPTR
          MOVX    A,@DPTR
          MOV     R1,A
          MOV     B,A
TODAY:    MOV     A,B
          ADD     A,#03H
          MOV     DPL,A
          MOV     DPH,#00H
          MOVX    A,@DPTR
          MOV     R1,A
          MOV     R0,#LHBUF
          MOV     A,@R0
          SUBB    A,R1
          CJNE    A,#00H,END_MESG
          LCALL   PLAY_R
          LJMP    TODAY
P_ALL:    MOV     R5,#00H
          MOV     A,#01H
          LCALL   LCDWI
          MOV     DPTR,#REC
          MOVX    A,@DPTR
          MOV     R6,A
          INC     DPTR
          MOVX    A,@DPTR
          MOV     R7,A
          INC     R7
          INC     DPTR
          MOVX    A,@DPTR
          MOV     R0,A
          INC     DPTR
          MOVX    A,@DPTR
          MOV     R1,A
          MOV     B,A
P_ALL1:   LCALL   PLAY_R
          INC     R5
          CJNE    R5,#0FH,P_ALL1
END_MESG: MOV     DPTR,#PAGE16
          LCALL   LCDLD
          MOV     R6,#78H
          MOV     R7,#39H
          LCALL   PLAY1
          MOV     R2,#0FH
          LCALL   DELAY
          LJMP    MAIN
PLAY_R:   MOV     A,#0CH
          LCALL   LCDWI
          MOV     R2,#02H
          LCALL   DELAY
          MOV     A,#01H
          LCALL   LCDWI
          MOV     R2,#02H
          LCALL   DELAY
          MOV     R2,#03H
          MOV     DPH,#00H
          MOV     DPL,B
          MOV     R0,#TASCBU
CON11:    MOVX    A,@DPTR
          ANL     A,#0F0H
```

```
            SWAP        A
            ORL         A,#30H
            MOV         @R0,A
            INC         R0
            MOVX        A,#DPTR
            ANL         A,#0FH
            ORL         A,#30H
            MOV         @R0,A
            INC         DPTR
            INC         R0
            DJNZ        R2,CON11
            MOV         R2,#03H
            MOV         R0,#DASCBU
CON12:      MOVX        A,@DPTR
            ANL         A,#0F0H
            SWAP        A
            ORL         A,#30H
            MOV         @R0,A
            INC         R0
            MOVX        A,@DPTR
            ANL         A,#0FH
            ORL         A,#30H
            MOV         @R0,A
            INC         DPTR
            INC         R0
            DJNZ        R2,CON12
            MOV         A,P_AT
            CJNE        A,#01H,PG_15
PG_14:      MOV         DPTR,#PAGE14
            LCALL       LCDLD
            SJMP        CO_RON
PG_15:      MOV         DPTR,#PAGE15
            LCALL       LCDLD
            SJMP        CO_RON
CO_RON:     MOV         CORON,#3AH
            MOV         R0,#0D4H
            MOV         R1,#TASCBU
            LCALL       DIS_LCD
            MOV         CORON,#2FH
            MOV         R0,#094H
            MOV         R1,#DASCBU
            LCALL       DIS_LCD
            LCALL       PLAY2
            MOV         A,B
            CJNE        A,#0AH,CHG1
            MOV         A,#5EH
            MOV         R1,A
            MOV         B,A
CON_122:    CJNE        R6,#00H,PLAYER1
            MOV         R6,#78H
            CJNE        R7,#1DH,PLAYER2
            MOV         R7,#2DH
            RET
CHG1:       SUBB        A,#06H
            MOV         R1,A
            MOV         B,A
            LJMP        CON_122
PLAYER1:    MOV         A,R6
            SUBB        A,#1EH
            MOV         R6,A
            RET
```

```
PLAYER2:    CJNE     R7,#2DH,PLAYER3
            MOV      R7,35H
            RET
PLAYER3:    MOV      R7,#1DH
            RET
            ;------------------SUB RECORD FOR PRESENT-----------------
RECORD:     MOV      DPTR,#REC
            MOVX     A,@DPTR
            MOV      R6,A
            INC      DPTR
            MOVX     A,@DPTR
            MOV      R7,A
            INC      DPTR
            MOVX     A,@DPTR
            MOV      R0,A
            INC      DPTR
            MOVX     A,@DPTR
            MOV      R1,A
            CJNE     R1,#5EH,CHG
            MOV      R1,#0AH
            MOV      B,R1
            LJMP     REC0
CHG:        MOV      A,R1
            ADD      A,#06H
            MOV      R1,A
            MOV      B,A
REC0:       MOV      DPH,R0
            MOV      DPL,R1
            MOV      R2,#00H
            MOV      R0,#THBUF
REC00:      MOV      A,@R0
            MOVX     @DPTR,A
            INC      R0
            INC      DPTR
            INC      R2
            CJNE     R2,#06H,REC00
            CJNE     R6,#78H,REC1
            MOV      R6,#00H
            CJNE     R7,#34H,REC01
            MOV      R7,#2CH
            LJMP     REC_R
REC01:      CJNE     R7,#2CH,REC02
            MOV      R7,#1CH
            LJMP     REC_R
REC02:      MOV      R7,#34H
            LJMP     REC_R
REC1:       MOV      A,R6
            ADD      A,#1EH
            MOV      R6,A
REC_R:      MOV      A,#3CH
            LCALL    SET_P
            MOV      A,R6
            LCALL    SET_ADD
            MOV      A,R7
            LCALL    SET_P
            MOV      R2,#76H
            LCALL    DELAY
            MOV      A,#3CH
            LCALL    SET_P
REC_NOW:    MOV      DPTR,#REC    ;0006H
            MOV      A,R6
```

```
        MOVX        @DPTR,A
        INC         DPTR
        MOV         A,R7
        MOVX        @DPTR,A
        INC         DPTR
        MOV         R0,#00H
        MOV         A,R0
        MOVX        @DPTR,A
        INC         DPTR
        MOV         A,B
        MOVX        @DPTR,A
        RET
CK_EOM1: MOV        DPTR,#PORTC4
        MOVX        A,@DPTR
        ANL         A,#03H
        MOV         R0,A
        CJNE        R0,#01H,CK_EOM1    ;02 = EOM2
        MOV         A,#3FH
        LCALL       SET_P
        RET
CK_EOM2: MOV        DPTR,#PORTC4
        MOVX        A,@DPTR
        ANL         A,#03H
        MOV         R0,A
        CJNE        R0,#02H,CK_EOM2
        MOV         A,#3FH
        LCALL       SET_P
        RET
;***********************DATA DISPLAY LCD***************************
PAGE1:  DB          "     Facilities      "
        DB          "    Via telephone    "
        DB          "         2003        "
        DB          "   Telecom.Projct    "
PAGE2:  DB          "- - - - - - - - - - - - "
        DB          "      ..Ssytem..     "
        DB          "         O.K         "
        DB          "- - - - - - - - - - - - "
PAGE3:  DB          "- - - - - - - - - - - - "
        DB          "     Conversation    "
        DB          "- - - - - - - - - - - - "
        DB          "                     "
PAGE4:  DB          "- - - - - - - - - - - - "
        DB          "    Incoming Call    "
        DB          "- - - - - - - - - - - - "
        DB          "                     "
PAGE5:  DB          "- - - - - - - - - - - - "
        DB          "   Auto Answering    "
        DB          "- - - - - - - - - - - - "
        DB          "                     "
PAGE6:  DB          "- - - - - - - - - - - - "
        DB          "      Recording      "
        DB          "       **Mode**      "
        DB          "- - - - - - - - - - - - "
PAGE7:  DB          "- - - - - - - - - - - - "
        DB          "  Coctrol   Electric "
        DB          "       **Mode**      "
        DB          "- - - - - - - - - - - - "
PAGE8:  DB          "- - - - - - - - - - - - "
        DB          "      Play  Back     "
        DB          "       **Mode**      "
        DB          "- - - - - - - - - - - - "
```

```
PAGE9:      DB      "- - - - - - - - - - - - - - "
            DB      "      Outgoing Call        "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
PAGE10:     DB      "- - - - - - - - - - - - - - "
            DB      "          Busy....         "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
PAGE11:     DB      "- - - - - - - - - - - - - - "
            DB      "       Connecting...       "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
PAGE12:     DB      "- - - - - - - - - - - - - - "
            DB      "      Enter  Password      "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
PAGE13:     DB      "- - - - - - - - - - - - - - "
            DB      "          Incorrect        "
            DB      "         Try Again....     "
            DB      "- - - - - - - - - - - - - - "
PAGE14:     DB      "       Message  Today      "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
            DB      "                            "
PAGE15:     DB      "        Message All        "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
            DB      "                            "
PAGE16:     DB      "- - - - - - - - - - - - - - "
            DB      "     End  of Message       "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
PAGE17:     DB      "          Telecom..        "
            DB      "- - - - - - - - - - - - - - "
            DB      "                            "
            DB      "                            "
PAGE18:     DB      "   Check Message           "
            DB      "   1. Today                "
            DB      "   2. All                  "
            DB      "   3. Exit                 "
PAGE19:     DB      "         **Menu**          "
            DB      "- - - - - - - - - - - - - - "
            DB      "   1. Setmode              "
            DB      "   2. Chk Msg.             "
PAGE20:     DB      "         **Menu**          "
            DB      "   3. Bill Mode            "
            DB      "   4. Control Elec         "
            DB      "   5. Exit                 "
PAGE21:     DB      "     **Set Mode**          "
            DB      "   1. Time-Date            "
            DB      "   2. Password             "
            DB      "   3. Ringing              "
PAGE22:     DB      "        Msg.Mode           "
            DB      "- - - - - - - - - - - - - - "

            DB      "   1. Today                "
            DB      "   2. All                  "
PAGE23:     DB      "        Data Mode          "
            DB      "- - - - - - - - - - - - - - "
            DB      "  1. Total Price           "
            DB      "  2. Detail Price          "
```

```
PAGE24:     DB      "        Time-Date         "
            DB      "- - - - - - - - - - - - - - "
            DB      "                          "
            DB      "                          "
PAGE25:     DB      "        **Password**      "
            DB      "Old:                      "
            DB      "New:                      "
            DB      "Retry:                    "
PAGE26:     DB      "       Ring Count         "
            DB      "- - - - - - - - - - - - - - "
            DB      "Old:                      "
            DB      "New:                      "
PAGE27:     DB      "- - - - - - - - - - - - - - "
            DB      "         Password         "
            DB      "           Wrong          "
            DB      "- - - - - - - - - - - - - - "
PAGE28:     DB      "- - - - - - - - - - - - - - "
            DB      "       New  Password      "
            DB      "            O.K           "
            DB      "- - - - - - - - - - - - - - "
PAGE29:     DB      "Phone:                    "
            DB      "Start:                    "
            DB      "Stop:                     "
            DB      "Price                 Baht"
PAGE30:     DB      "Phone:                    "
            DB      "Date:                     "
            DB      "Time:                     "
            DB      "Price                 Baht"
PAGE31:     DB      "       On/Off Elec        "
            DB      "       Press * On         "
            DB      "       Press # Off        "
            DB      "       Number :           "
PAGE32:     DB      "                          "
            DB      "Price                 Baht"
            DB      "                          "
            DB      "                          "
PAGE33:     DB      "          Data Mode       "
            DB      "- - - - - - - - - - - - - - "
            DB      "  1. Total Price          "
            DB      "  2. Detail Price         "
PAGE34:     DB      "          Choose          "
            DB      "- - - - - - - - - - - - - - "
            DB      " 1.MessageToday           "
            DB      " 2.Message All            "
PAGE35:     DB      "- - - - - - - - - - - - - - "
            DB      "Month...                  "
            DB      "Price                 Baht"
            DB      "- - - - - - - - - - - - - - "
PAGE36:     DB      "- - - - - - - - - - - - - - "
            DB      "      End of Data         "
            DB      "- - - - - - - - - - - - - - "
            DB      "                          "
PAGE37:     DB      "- - - - - - - - - - - - - - "
            DB      "      Over of Data        "
            DB      "- - - - - - - - - - - - - - "
            DB      "                          "
ASCII:      DB      31H,32H,33H,4DH
            DB      34H,35H,36H,2BH
            DB      37H,38H,39H,2DH
            DB      23H,30H,2AH,7FH
END
```

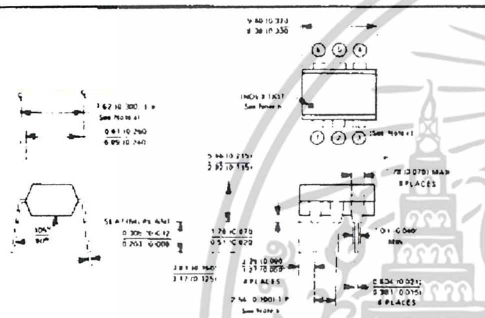
SOOS035 D2493 SEPTEMBER 1978 – REVISED MARCH 1983

COMPATIBLE WITH STANDARD TTL INTEGRATED CIRCUITS

- **Gallium Arsenide Diode Infrared Source Optically Coupled to a Silicon N-P-N Phototransistor**
- **High Direct-Current Transfer Ratio**
- **High-Voltage Electrical Isolation . . . 2.5-kV, 1.5-kV, or 0.5-kV Rating**
- **Plastic Dual-In-Line Package**
- **High-Speed Switching . . . t_r = 2 μs, t_f = 2 μs Typical**

mechanical data

The package consists of a gallium arsenide infrared-emitting diode and an n-p-n silicon phototransistor mounted on a 6-lead frame encapsulated within an electrically nonconductive plastic compound. The case will withstand soldering temperature with no deformation and device performance characteristics remain stable when operated in high-humidity conditions. Unit weight is approximately 0.52 grams.



NOTES:
a. Leads are within 0,127 (0.005) radius of true position (T.P.) with maximum material condition and unit installed.
b. Pin 1 identified by index dot.
c. Terminal connections:
1. Anode } Infrared-emitting diode
2. Cathode
3. No internal connection
4. Emitter } Phototransistor
5. Collector
6. Base

FALLS WITHIN JEDEC MO-001AM DIMENSIONS
ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)

*Peak Input-to-Output Voltage: 4N25	±2.5 kV
4N26, 4N27	±1.5 kV
4N28	±0.5 kV
*Collector-Base Voltage	70 V
*Collector-Emitter Voltage (See Note 1)	30 V
*Emitter-Collector Voltage	7 V
Emitter-Base Voltage	7 V
*Input-Diode Reverse Voltage	3 V
*Input-Diode Continuous Forward Current at (or below) 25°C Free-Air Temperature (See Note 2)	80 mA
*Input-Diode Peak Forward Current (t_w = 300 μs, duty cycle = 2%)	3 A
*Continuous Power Dissipation at (or below) 25°C Free-Air Temperature:	
Infrared-Emitting Diode (See Note 3)	150 mW
Phototransistor (See Note 3)	150 mW
Total, Infrared-Emitting Diode plus Phototransistor (See Note 4)	250 mW
*Storage Temperature Range	−55°C to 150°C
*Lead Temperature 1,6 mm (1/16 inch) from Case for 10 Seconds	260°C

*JEDEC registered data. This data sheet contains all applicable JEDEC-registered data in effect at the time of publication.

NOTES:
1. This value applies when the base-emitter diode is open-circulated.
2. Derate linearly to 100°C free-air temperature at the rate of 1.33 mA/°C.
3. Derate linearly to 100°C free-air temperature at the rate of 2 mW/°C.
4. Derate linearly to 100°C free-air temperature at the rate of 3.33 mW/°C.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

electrical characteristics at 25°C free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	4N25, 4N26			4N27, 4N28			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
*$V_{(BR)CBO}$	Collector-Base Breakdown Voltage	I_C = 100 µA, I_E = 0, I_F = 0	70			70			V
*$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage	I_C = 1 mA, I_B = 0, I_F = 0	30			30			V
*$V_{(BR)ECO}$	Emitter-Collector Breakdown Voltage	I_E = 100 µA, I_B = 0, I_F = 0	7			7			V
*I_R	Input Diode Static Reverse Current	V_R = 3 V			100			100	µA
*$I_{C(on)}$	On-State Collector Current (Phototransistor Operation)	V_{CE} = 10 V, I_B = 0, I_F = 10 mA	2	5		1	3		mA
$I_{C(on)}$	On-State Collector Current (Photodiode Operation)	V_{CB} = 10 V, I_E = 0, I_F = 10 mA		20			20		µA
*$I_{C(off)}$	Off-State Collector Current (Phototransistor Operation)	V_{CE} = 10 V, I_B = 0, I_F = 0		1	50		1	50	nA
*$I_{C(off)}$	Off-State Collector current (Photodiode Operation)	V_{CB} = 10 V, I_E = 0, I_F = 0		0.1	20		0.1	20	nA
*V_F	Input Diode Static Forward Voltage	I_F = 10 mA		1.25	1.5		1.25	1.5	V
*$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	I_C = 2 mA, I_B = 0, I_F = 50 mA		0.25	0.5		0.25	0.5	V
r_{IO}	Input-to-Output Internal resistance	V_{in-out} = ±2.5 kV for 4N25, ±1.5 kV for 4N26, 4N27, ±0.5 kV for 4N28, See Note 5	10^{11}	10^{12}		10^{11}	10^{12}		Ω
C_{io}	Input-to-Output Capacitance	V_{in-out} = 0, f = 1 MHz, See Note 5		1			1		pF

*JEDEC registered data

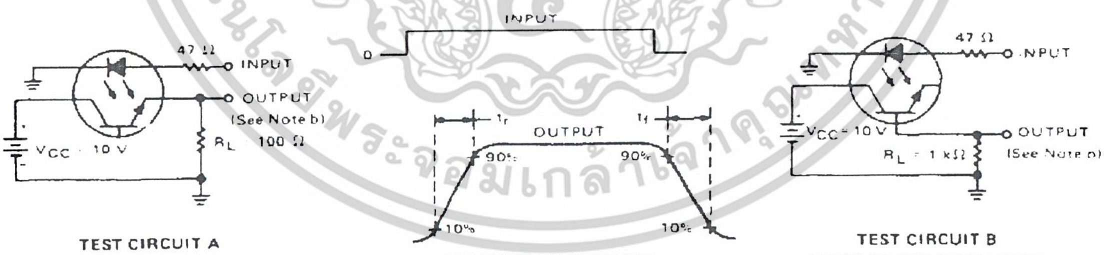
NOTE 5: These parameters are measured between both input diode leads shorted together and all the phototransistor leads shorted together.

switching characteristics at 25°C free-air temperature

PARAMETER			TEST CONDITIONS	TYP	UNIT
t_r	Rise Time	Phototransistor Operation	V_{CC} = 10 V, I_B = 0, $I_{C(on)}$ = 2 mA, R_L = 100 Ω, See Test Circuit A of Figure 1	2	µs
t_f	Fall Time			2	
t_r	Rise Time	Photodiode Operation	V_{CC} = 10 V, I_E = 0, $I_{C(on)}$ = 20 µA, R_L = 1 kΩ, See Test Circuit B of Figure 1	1	µs
t_f	Fall Time			1	

PARAMETER MEASUREMENT INFORMATION

Adjust amplitude of input pulse for:
$I_{C(on)}$ = 2 mA (Test Circuit A) or
$I_{C(on)}$ = 20 µA (Test Circuit B)



TEST CIRCUIT A
PHOTOTRANSISTOR OPERATION

VOLTAGE WAVEFORMS

TEST CIRCUIT B
PHOTODIODE OPERATION

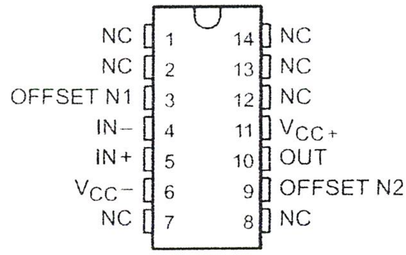
NOTES a. The input waveform is supplied by a generator with the following characteristics: Z_{out} = 50 Ω, t_r ≤ 15 ns, duty cycle ≈ 1%, t_w = 100 µs.

b. The output waveform is monitored on an oscilloscope with the following characteristics: t_r ≤ 12 ns, R_{in} ≥ 1 MΩ, C_{in} ≤ 20 pF.

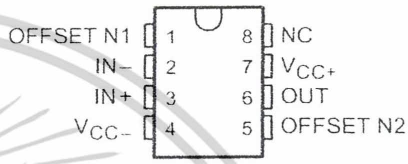
FIGURE 1 – SWITCHING TIMES

- Short-Circuit Protection
- Offset-Voltage Null Capability
- Large Common-Mode and Differential Voltage Ranges
- No Frequency Compensation Required
- Low Power Consumption
- No Latch-Up
- Designed to Be Interchangeable With Fairchild μA741

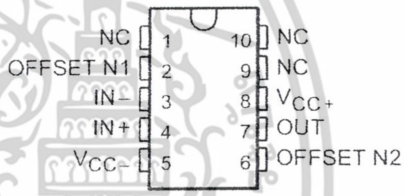
μA741M . . . J PACKAGE
(TOP VIEW)



μA741M . . . JG PACKAGE
μA741C, μA741I . . . D, P, OR PW PACKAGE
(TOP VIEW)



μA741M . . . U PACKAGE
(TOP VIEW)



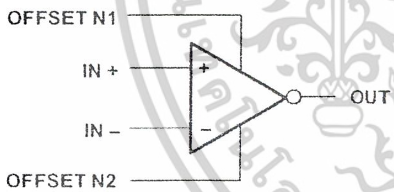
description

The μA741 is a general-purpose operational amplifier featuring offset-voltage null capability.

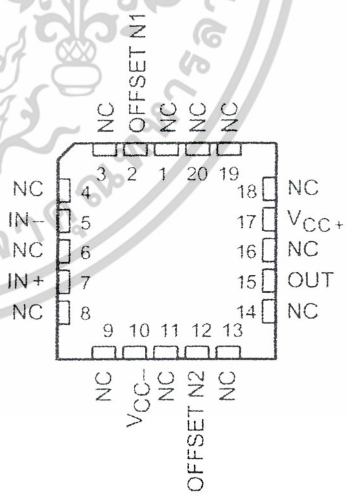
The high common-mode input voltage range and the absence of latch-up make the amplifier ideal for voltage-follower applications. The device is short-circuit protected and the internal frequency compensation ensures stability without external components. A low value potentiometer may be connected between the offset null inputs to null out the offset voltage as shown in Figure 2.

The μA741C is characterized for operation from 0°C to 70°C. The μA741I is characterized for operation from −40°C to 85°C.The μA741M is characterized for operation over the full military temperature range of −55°C to 125°C.

symbol



μA741M . . . FK PACKAGE
(TOP VIEW)



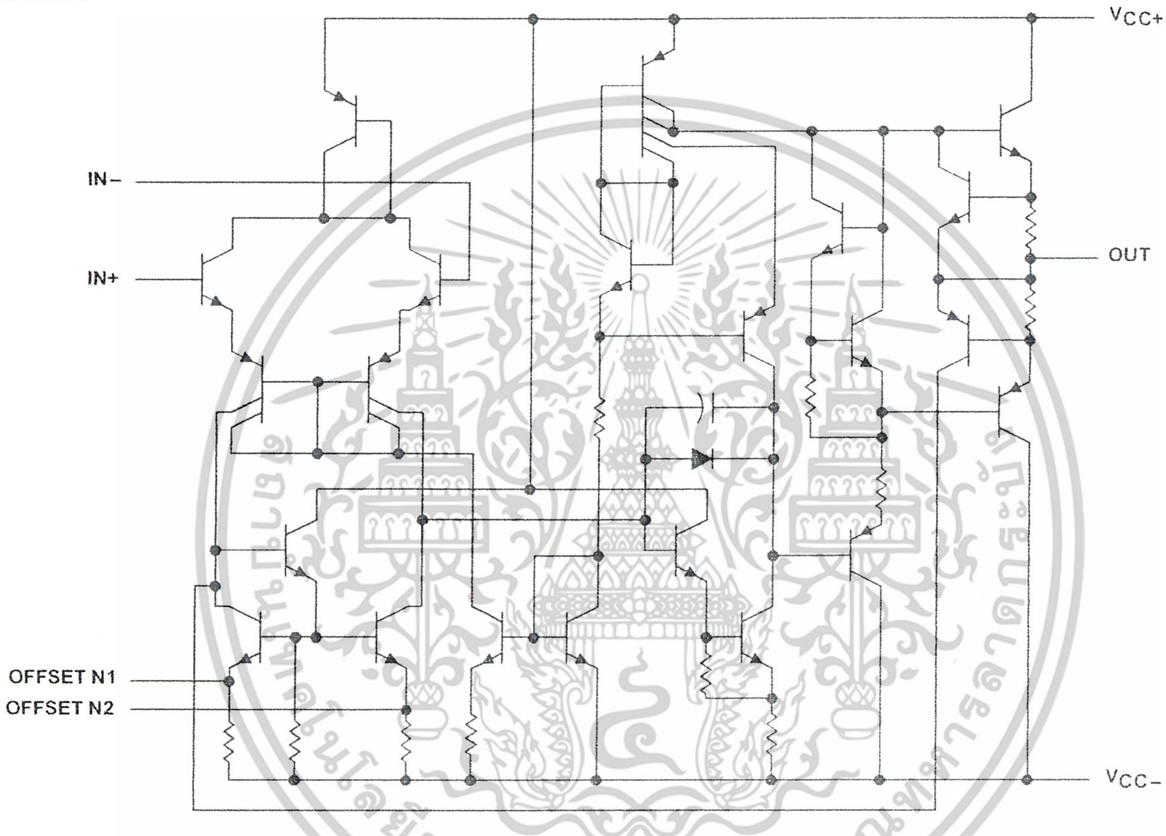
NC – No internal connection

AVAILABLE OPTIONS

T_A	PACKAGED DEVICES							CHIP FORM (Y)
	SMALL OUTLINE (D)	CHIP CARRIER (FK)	CERAMIC DIP (J)	CERAMIC DIP (JG)	PLASTIC DIP (P)	TSSOP (PW)	FLAT PACK (U)	
0°C to 70°C	μA741CD				μA741CP	μA741CPW		μA741Y
–40°C to 85°C	μA741ID				μA741IP			
–55°C to 125°C		μA741MFK	μA741MJ	μA741MJG			μA741MU	

The D package is available taped and reeled. Add the suffix R (e.g., μA741CDR).

schematic

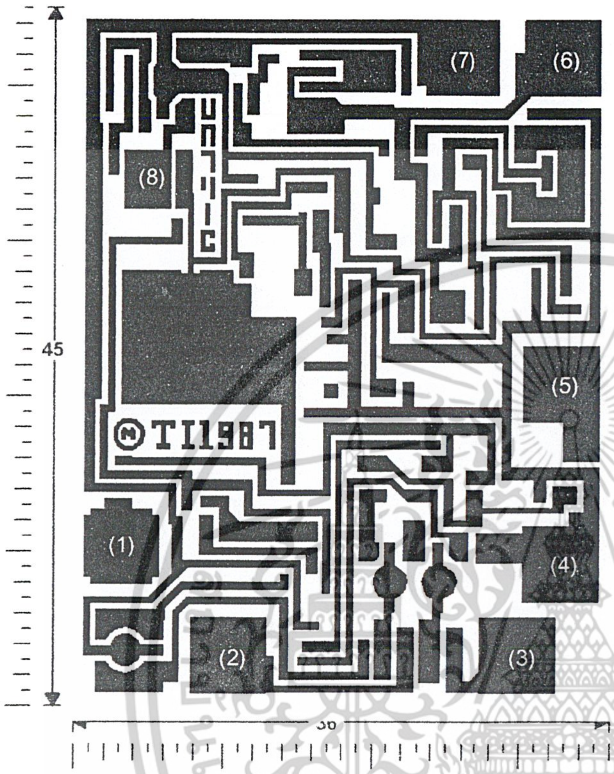


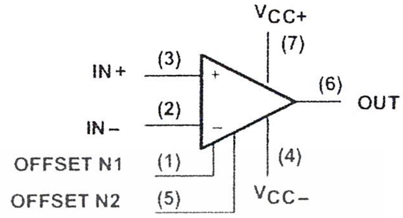
Component Count	
Transistors	22
Resistors	11
Diode	1
Capacitor	1

μA741Y chip information

This chip, when properly assembled, displays characteristics similar to the μA741C. Thermal compression or ultrasonic bonding may be used on the doped-aluminum bonding pads. Chips may be mounted with conductive epoxy or a gold-silicon preform.

BONDING PAD ASSIGNMENTS





CHIP THICKNESS: 15 TYPICAL

BONDING PADS: 4 × 4 MINIMUM

$T_J$max = 150°C.

TOLERANCES ARE ±10%.

ALL DIMENSIONS ARE IN MILS.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

		µA741C	µA741I	µA741M	UNIT
Supply voltage, V_{CC+} (see Note 1)		18	22	22	V
Supply voltage, V_{CC-} (see Note 1)		–18	–22	–22	V
Differential input voltage, V_{ID} (see Note 2)		±15	±30	±30	V
Input voltage, V_I any input (see Notes 1 and 3)		±15	±15	±15	V
Voltage between offset null (either OFFSET N1 or OFFSET N2) and V_{CC-}		±15	±0.5	±0.5	V
Duration of output short circuit (see Note 4)		unlimited	unlimited	unlimited	
Continuous total power dissipation		See Dissipation Rating Table			
Operating free-air temperature range, T_A		0 to 70	–40 to 85	–55 to 125	°C
Storage temperature range		–65 to 150	–65 to 150	–65 to 150	°C
Case temperature for 60 seconds	FK package			260	°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds	J, JG, or U package			300	°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	D, P, or PW package	260	260		°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values, unless otherwise noted, are with respect to the midpoint between V_{CC+} and V_{CC-}.
2. Differential voltages are at IN+ with respect to IN–.
3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 V, whichever is less.
4. The output may be shorted to ground or either power supply. For the µA741M only, the unlimited duration of the short circuit applies at (or below) 125°C case temperature or 75°C free-air temperature.

DISSIPATION RATING TABLE

PACKAGE	$T_A \le 25°C$ POWER RATING	DERATING FACTOR	DERATE ABOVE T_A	$T_A = 70°C$ POWER RATING	$T_A = 85°C$ POWER RATING	$T_A = 125°C$ POWER RATING
D	500 mW	5.8 mW/°C	64°C	464 mW	377 mW	N/A
FK	500 mW	11.0 mW/°C	105°C	500 mW	500 mW	275 mW
J	500 mW	11.0 mW/°C	105°C	500 mW	500 mW	275 mW
JG	500 mW	8.4 mW/°C	90°C	500 mW	500 mW	210 mW
P	500 mW	N/A	N/A	500 mW	500 mW	N/A
PW	525 mW	4.2 mW/°C	25°C	336 mW	N/A	N/A
U	500 mW	5.4 mW/°C	57°C	432 mW	351 mW	135 mW

RETRIGGERABLE MONOSTABLE MULTIVIBRATORS

SDLS043 – DECEMBER 1983 – REVISED MARCH 1988

- **D-C Triggered from Active-High or Active-Low Gated Logic Inputs**
- **Retriggerable for Very Long Output Pulses, Up to 100% Duty Cycle**
- **Overriding Clear Terminates Output Pulse**
- **'122 and 'LS122 Have Internal Timing Resistors**

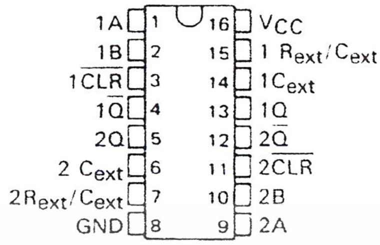
description

These d-c triggered multivibrators feature output pulse-duration control by three methods. The basic pulse time is programmed by selection of external resistance and capacitance values (see typical application data). The '122 and 'LS122 have internal timing resistors that allow the circuits to be used with only an external capacitor, if so desired. Once triggered, the basic pulse duration may be extended by retriggering the gated low-level-active (A) or high-level-active (B) inputs, or be reduced by use of the overriding clear. Figure 1 illustrates pulse control by retriggering and early clear.

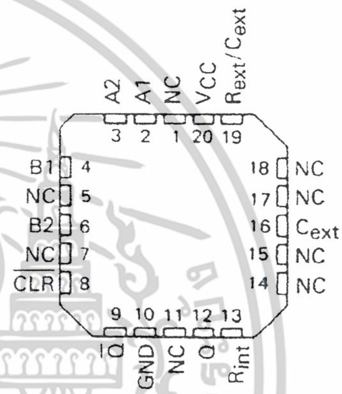
The 'LS122 and 'LS123 are provided enough Schmitt hysteresis to ensure jitter-free triggering from the B input with transition rates as slow as 0.1 millivolt per nanosecond.

The R_{int} in nominall 10 kΩ for '122 and 'LS122.

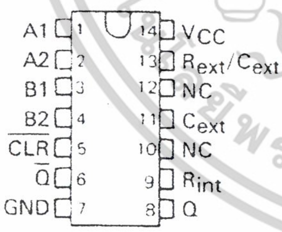
SN54123, SN54130, SN54LS123 . . . J OR W PACKAGE
SN74123, SN74130 . . . N PACKAGE
SN74LS123 . . . D OR N PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



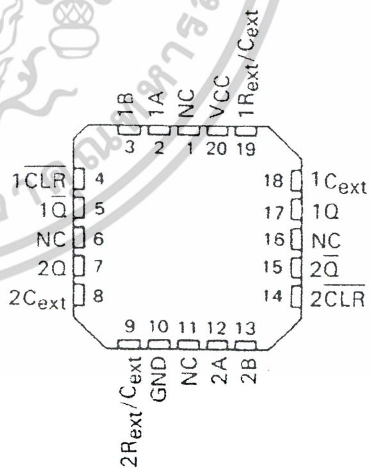
SN54LS122 . . . FK PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



SN54122, SN54LS122 . . . J OR W PACKAGE
SN74122 . . . N PACKAGE
SN74LS122 . . . D OR N PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



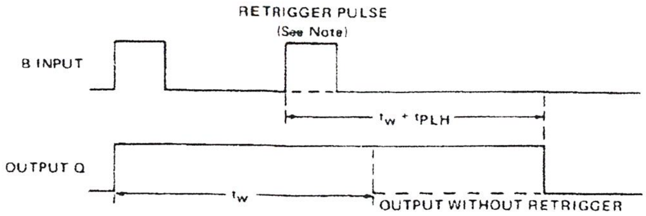
SN54LS123 . . . FK PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



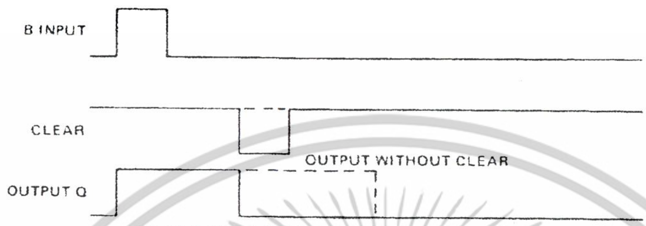
NC - No internal connection

NOTES: 1. An external timing capacitor may be connected between C_{ext} and R_{ext}/C_{ext} (positive).
2. To use the internal timing resistor of '122 or 'LS122, connect R_{int} to V_{CC}.
3. For improved pulse duration accuracy and repeatability, connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{int} open-circuited.
4. To obtain variable pulse durations, connect an external variable resistance between R_{int} or R_{ext}/C_{ext} and V_{CC}.

description (continued)



OUTPUT PULSE CONTROL USING RETRIGGER PULSE



OUTPUT PULSE CONTROL USING CLEAR INPUT

NOTE: Retrigger pulses starting before 0.22 C_{ext} (in picofrads) nanoseconds after the initial trigger pulse will be ignored and the output duration will remain unchanged.

FIGURE 1–TYPICAL INPUT/OUTPUT PULSES

'122, 'LS122
FUNCTION TABLE

INPUTS					OUTPUTS	
CLEAR	A1	A2	B1	B2	Q	$\bar{Q}$
L	X	X	X	X	L	H
X	H	H	X	X	L†	H†
X	X	X	L	X	L†	H†
X	X	X	X	L	L†	H†
H	L	X	↑	H	⊓	⊔
H	L	X	H	↑	⊓	⊔
H	X	L	↑	H	⊓	⊔
H	X	L	H	↑	⊓	⊔
H	H	↓	H	H	⊓	⊔
H	↓	↓	H	H	⊓	⊔
H	↓	H	H	H	⊓	⊔
↑	L	X	H	H	⊓	⊔
↑	X	L	H	H	⊓	⊔

'123, '130, 'LS123
FUNCTION TABLE

INPUTS			OUTPUTS	
CLEAR	A	B	Q	$\bar{Q}$
L	X	X	L	H
X	H	X	L†	H†
X	X	L	L†	H†
H	L	↑	⊓	⊔
H	↓	H	⊓	⊔
↑	L	H	⊓	⊔

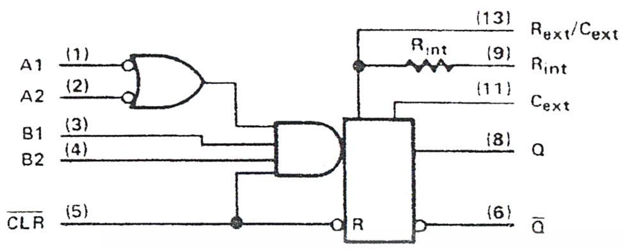
See explanation of function tables on page

† These lines of the functional tables assume that the indicated steady-state conditons at the A and B inputs have been set up long enough to complete any pulse started before the set up.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

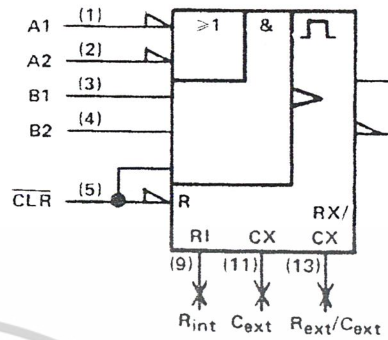
logic diagram (positive logic)

'122, 'LS122



logic symbol†

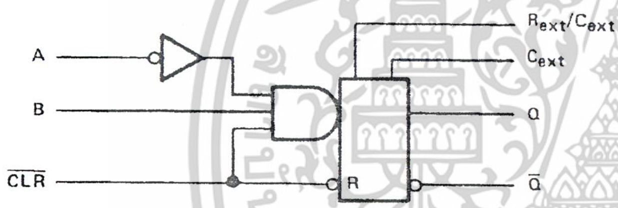
'122, 'LS122



R_{int} is nominally 10 kΩ for '122 and 'LS122

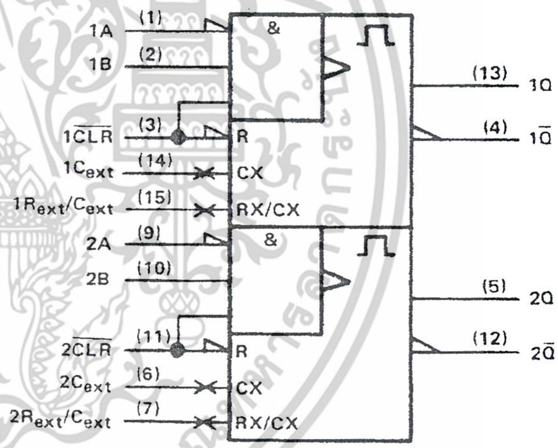
logic diagram (positive logic) (each multivibrator)

'123, '130, 'LS123



logic symbol†

'123, '130, 'LS123



Pin numbers shown are for D, J, N, and W packages.

†These symbols are in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

P89C51RB2/P89C51RC2/ P89C51RD2Hxx

DESCRIPTION

The P89C51RB2/RC2/RD2Hxx device contains a non-volatile 16KB/32KB/64KB Flash program memory that is both parallel programmable and serial In-System and In-Application Programmable. In-System Programming (ISP) allows the user to download new code while the microcontroller sits in the application. In-Application Programming (IAP) means that the microcontroller fetches new program code and reprograms itself while in the system. This allows for remote programming over a modem link. A default serial loader (boot loader) program in ROM allows serial In-System programming of the Flash memory via the UART without the need for a loader in the Flash code. For In-Application Programming, the user program erases and reprograms the Flash memory by use of standard routines contained in ROM.

This device executes one machine cycle in 6 clock cycles, hence providing twice the speed of a conventional 80C51. An OTP configuration bit lets the user select conventional 12 clock timing if desired.

This device is a Single-Chip 8-Bit Microcontroller manufactured in advanced CMOS process and is a derivative of the 80C51 microcontroller family. The instruction set is 100% compatible with the 80C51 instruction set.

The device also has four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits.

The added features of the P89C51RB2/RC2/RD2Hxx makes it a powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control.

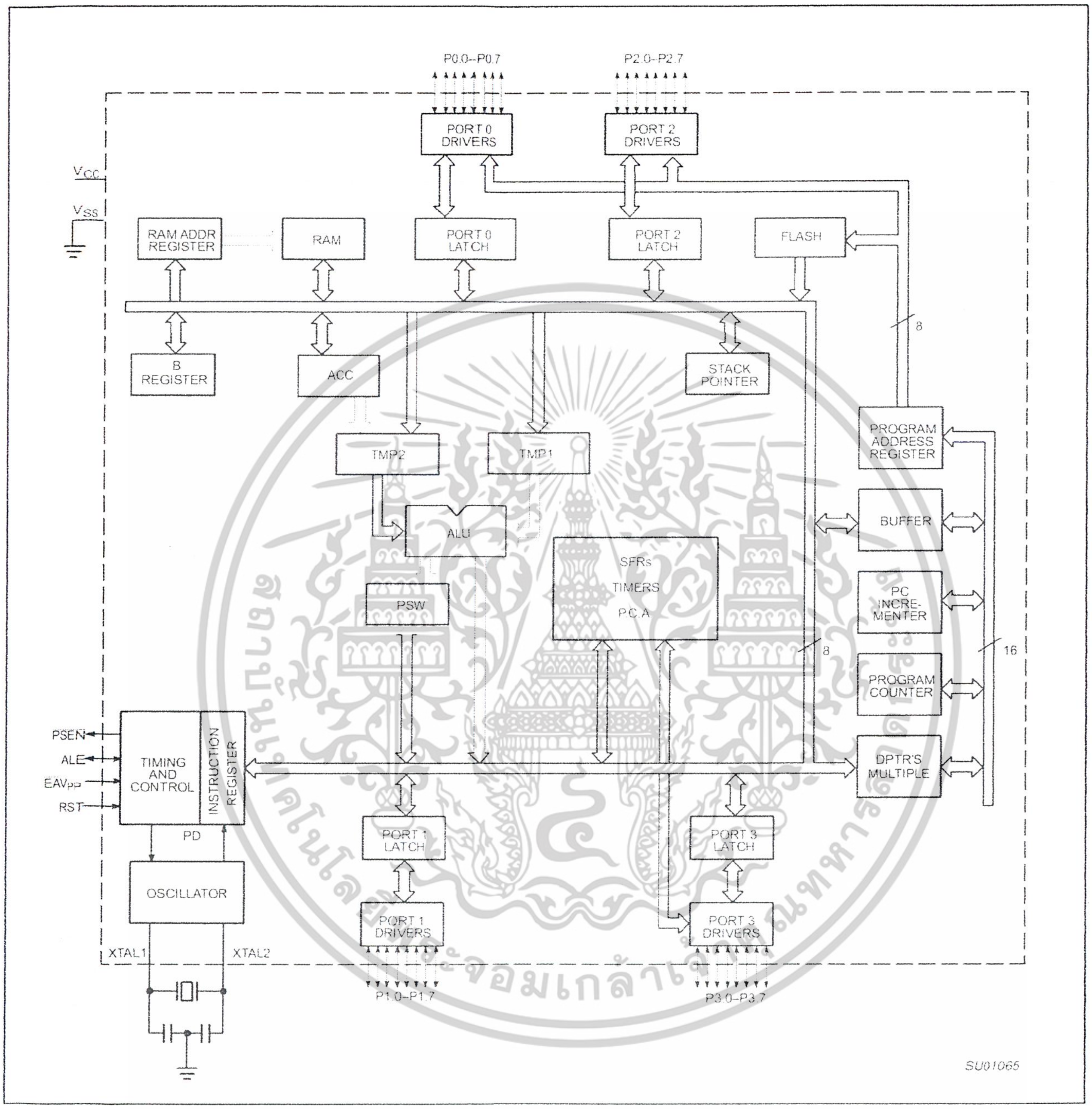
FEATURES

- 80C51 Central Processing Unit
- On-chip Flash Program Memory with In-System Programming (ISP) and In-Application Programming (IAP) capability
- Boot ROM contains low level Flash programming routines for downloading via the UART
- Can be programmed by the end-user application (IAP)
- Parallel programming with 87C51 compatible hardware interface to programmer
- Six clocks per machine cycle operation (standard)
- 12 clocks per machine cycle operation (optional)
- Speed up to 20 MHz with 6 clock cycles per machine cycle (40 MHz equivalent performance); up to 33 MHz with 12 clocks per machine cycle
- Fully static operation
- RAM expandable externally to 64 kbytes
- Four interrupt priority levels
- Seven interrupt sources
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power down mode
- Programmable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Programmable Counter Array (PCA)
  - PWM
  - Capture/compare

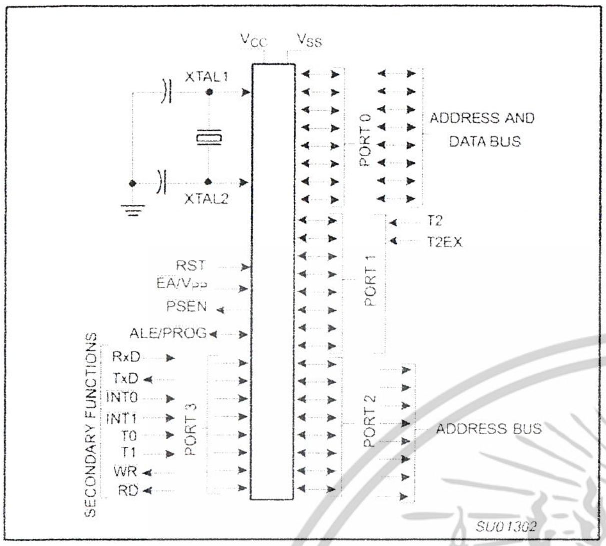
ORDERING INFORMATION

	PART ORDER NUMBER	MEMORY		TEMPERATURE RANGE (°C) AND PACKAGE	VOLTAGE RANGE	FREQUENCY (MHz)		DWG #
		FLASH	RAM			6 CLOCK MODE	12 CLOCK MODE	
1	P89C51RB2HBA	16 kB	512 B	0 to +70, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
2	P89C51RB2HBBD	16 kB	512 B	0 to +70, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
3	P89C51RC2HBP	32 kB	512 B	0 to +70, PDIP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT129-1
4	P89C51RC2HBA	32 kB	512 B	0 to +70, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
5	P89C51RC2HFA	32 kB	512 B	–40 to +85, PLCC	4.75–5.25 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
6	P89C51RC2HBBD	32 kB	512 B	0 to +70, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
7	P89C51RC2HFBD	32 kB	512 B	–40 to +85, LQFP	4.75–5.25 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
8	P89C51RD2HBP	64 kB	1 kB	0 to +70, PDIP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT129-1
9	P89C51RD2HBA	64 kB	1 kB	0 to +70, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
10	P89C51RD2HBBD	64 kB	1 kB	0 to +70, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1

BLOCK DIAGRAM

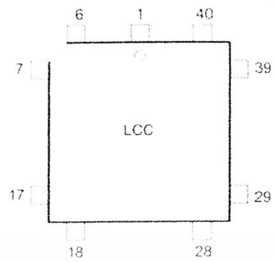


LOGIC SYMBOL



SU01302

Plastic Leaded Chip Carrier



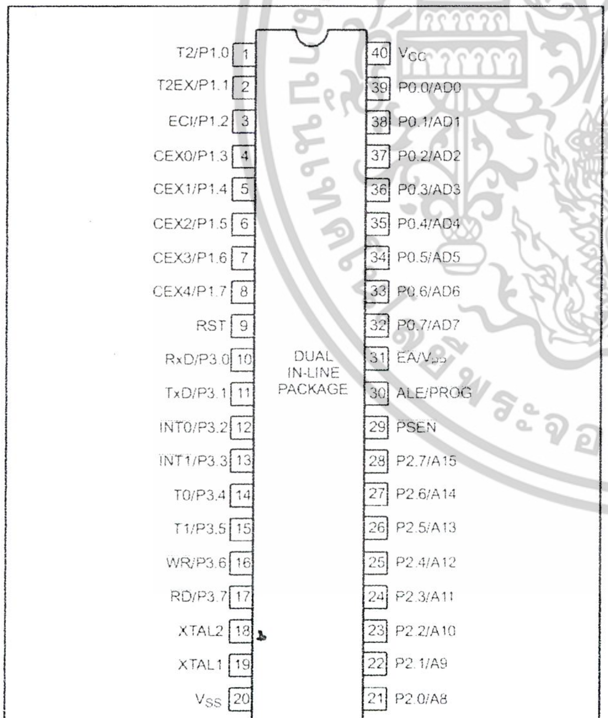
Pin	Function	Pin	Function	Pin	Function
1	NIC*	16	P3.4/T0	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1	32	PSEN
3	P1.1/T2EX	18	P3.6/WR	33	ALE/PROG
4	P1.2/ECI	19	P3.7/RD	34	NIC*
5	P1.3/CEX0	20	XTAL2	35	EA/V_{PP}
6	P1.4/CEX1	21	XTAL1	36	P0.7/AD7
7	P1.5/CEX2	22	V_{SS}	37	P0.6/AD6
8	P1.6/CEX3	23	NIC*	38	P0.5/AD5
9	P1.7/CEX4	24	P2.0/A8	39	P0.4/AD4
10	RST	25	P2.1/A9	40	P0.3/AD3
11	P3.0/RxD	26	P2.2/A10	41	P0.2/AD2
12	NIC*	27	P2.3/A11	42	P0.1/AD1
13	P3.1/TxD	28	P2.4/A12	43	P0.0/AD0
14	P3.2/INT0	29	P2.5/A13	44	V_{CC}
15	P3.3/INT1	30	P2.6/A14		

* NO INTERNAL CONNECTION

SU00023

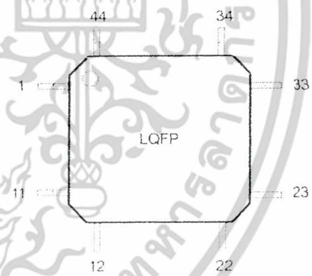
PINNING

Plastic Dual In-Line Package



SU00021

Plastic Quad Flat Pack



Pin	Function	Pin	Function	Pin	Function
1	P1.5/CEX2	16	V_{SS}	31	P0.6/AD6
2	P1.6/CEX3	17	NIC*	32	P0.5/AD5
3	P1.7/CEX4	18	P2.0/A8	33	P0.4/AD4
4	RST	19	P2.1/A9	34	P0.3/AD3
5	P3.0/RxD	20	P2.2/A10	35	P0.2/AD2
6	NIC*	21	P2.3/A11	36	P0.1/AD1
7	P3.1/TxD	22	P2.4/A12	37	P0.0/AD0
8	P3.2/INT0	23	P2.5/A13	38	V_{CC}
9	P3.3/INT1	24	P2.6/A14	39	NIC*
10	P3.4/T0	25	P2.7/A15	40	P1.0/T2
11	P3.5/T1	26	PSEN	41	P1.1/T2EX
12	P3.6/WR	27	ALE/PROG	42	P1.2/ECI
13	P3.7/RD	28	NIC*	43	P1.3/CEX0
14	XTAL2	29	EA/V_{PP}	44	P1.4/CEX1
15	XTAL1	30	P0.7/AD7		

* NO INTERNAL CONNECTION

SU01400

PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	PDIP	PLCC	LQFP		
V_{SS}	20	22	16	I	**Ground:** 0 V reference.
V_{CC}	40	44	38	I	**Power Supply:** This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	**Port 0:** Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pull-ups on all pins. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I_{IL}). Alternate functions for P89C51RB2/RC2/RD2Hxx Port 1 include:
	1	2	40	I/O	**T2 (P1.0):** Timer/Counter 2 external count input/Clockout (see Programmable Clock-Out)
	2	3	41	I	**T2EX (P1.1):** Timer/Counter 2 Reload/Capture/Direction Control
	3	4	42	I	**ECI (P1.2):** External Clock Input to the PCA
	4	5	43	I/O	**CEX0 (P1.3):** Capture/Compare External I/O for PCA module 0
	5	6	44	I/O	**CEX1 (P1.4):** Capture/Compare External I/O for PCA module 1
	6	7	1	I/O	**CEX2 (P1.5):** Capture/Compare External I/O for PCA module 2
	7	8	2	I/O	**CEX3 (P1.6):** Capture/Compare External I/O for PCA module 3
	8	9	3	I/O	**CEX4 (P1.7):** Capture/Compare External I/O for PCA module 4
P2.0–P2.7	21–28	24–31	18–25	I/O	**Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I_{IL}). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. P2.7 must be a "l" to program and erase the device.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I_{IL}). Port 3 also serves the special features of the P89C51RB2/RC2/RD2Hxx, as listed below:
	10	11	5	I	**RxD (P3.0):** Serial input port
	11	13	7	O	**TxD (P3.1):** Serial output port
	12	14	8	I	**$\overline{\text{INT0}}$ (P3.2):** External interrupt
	13	15	9	I	**$\overline{\text{INT1}}$ (P3.3):** External interrupt
	14	16	10	I	**T0 (P3.4):** Timer 0 external input
	15	17	11	I	**T1 (P3.5):** Timer 1 external input
	16	18	12	O	**$\overline{\text{WR}}$ (P3.6):** External data memory write strobe
	17	19	13	O	**$\overline{\text{RD}}$ (P3.7):** External data memory read strobe
RST	9	10	4	I	**Reset:** A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal resistor to V_{SS} permits a power-on reset using only an external capacitor to V_{CC}.
ALE	30	33	27	O	**Address Latch Enable:** Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted twice every machine cycle, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. ALE can be disabled by setting SFR auxiliary.0. With this bit set, ALE will be active only during a MOVX instruction.

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	PDIP	PLCC	LQFP		
$\overline{PSEN}$	29	32	26	O	**Program Store Enable:** The read strobe to external program memory. When executing code from the external program memory, PSEN is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory. $\overline{PSEN}$ is not activated during fetches from internal program memory.
EA/V_{PP}	31	35	29	I	**External Access Enable/Programming Supply Voltage:** EA must be externally held low to enable the device to fetch code from external program memory locations. If EA is held high, the device executes from internal program memory. The value on the EA pin is latched when RST is released and any subsequent changes have no effect. This pin also receives the programming supply voltage (V_{PP}) during Flash programming.
XTAL1	19	21	15	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	**Crystal 2:** Output from the inverting oscillator amplifier.

NOTE:
To avoid "latch-up" effect at power-on, the voltage on any pin (other than V_{PP}) must not be higher than V_{CC} + 0.5 V or less than V_{SS} – 0.5 V.

SPECIAL FUNCTION REGISTERS

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	–	–	–	–	–	–	EXTRAM	AO	xxxxxx00B
AUXR1#	Auxiliary 1	A2H	–	–	ENBOOT	–	GF2	0	–	DPS	xxxxxxx0B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH									xxxxxxxxB
CCAP1H#	Module 1 Capture High	FBH									xxxxxxxxB
CCAP2H#	Module 2 Capture High	FCH									xxxxxxxxB
CCAP3H#	Module 3 Capture High	FDH									xxxxxxxxB
CCAP4H#	Module 4 Capture High	FEH									xxxxxxxxB
CCAP0L#	Module 0 Capture Low	EAH									xxxxxxxxB
CCAP1L#	Module 1 Capture Low	EBH									xxxxxxxxB
CCAP2L#	Module 2 Capture Low	ECH									xxxxxxxxB
CCAP3L#	Module 3 Capture Low	EDH									xxxxxxxxB
CCAP4L#	Module 4 Capture Low	EEH									xxxxxxxxB
CCAPM0#	Module 0 Mode	DAH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	Module 1 Mode	DBH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	Module 2 Mode	DCH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	Module 3 Mode	DDH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	Module 4 Mode	DEH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
			DF	DE	DD	DC	DB	DA	D9	D8	
CCON*#	PCA Counter Control	D8H	CF	CR	–	CCF4	CCF3	CCF2	CCF1	CCF0	00x00000B
CH#	PCA Counter High	F9H									00H
CL#	PCA Counter Low	E9H									00H
CMOD#	PCA Counter Mode	D9H	CIDL	WDTE	–	–	–	CPS1	CPS0	ECF	00xxx000B
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt Enable 0	A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt Priority	B8H	–	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000B
			B7	B6	B5	B4	B3	B2	B1	B0	
IPH#	Interrupt Priority High	B7H	–	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	CEX4	CEX3	CEX2	CEX1	CEX0	ECI	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	RxD	FFH
PCON#[1]	Power Control	87H	SMOD1	SMOD0	–	POF	GF1	GF0	PD	IDL	00xxx000B

* SFRs are bit addressable.
\# SFRs are modified from or added to the 80C51 SFRs.
– Reserved bits.
1. Reset value depends on reset source.

SPECIAL FUNCTION REGISTERS (CONTINUED)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00000000B
RCAP2H#	Timer 2 Capture High	CBH									00H
RCAP2L#	Timer 2 Capture Low	CAH									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack Pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$	00H
T2MOD#	Timer 2 Mode Control	C9H	–	–	–	–	–	–	T2OE	DCEN	xxxxxx00B
TH0	Timer High 0	8CH									00H
TH1	Timer High 1	8DH									00H
TH2#	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2#	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	00H
WDTRST	Watchdog Timer Reset	A6H									

* SFRs are bit addressable.
\# SFRs are modified from or added to the 80C51 SFRs.
– Reserved bits.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. Minimum and maximum high and low times specified in the data sheet must be observed.

This device is configured at the factory to operate using 6 clock periods per machine cycle, referred to in this datasheet as "6 clock mode". (This yields performance equivalent to twice that of standard 80C51 family devices). It may be optionally configured on commercially-available EPROM programming equipment to operate at 12 clocks per machine cycle, referred to in this datasheet as "12 clock mode". Once 12 clock mode has been configured, it cannot be changed back to 6 clock mode.

RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (12 oscillator periods in 6 clock mode, or 24 oscillator periods in 12 clock mode), while the oscillator is running. To ensure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up. Ports 1, 2, and 3 will asynchronously be driven to their reset condition when a voltage above V_{IH1} (min.) is applied to RESET.

The value on the $\overline{EA}$ pin is latched when RST is deasserted and has no further effect.

National Semiconductor

February 1995

LM567/LM567C Tone Decoder

General Description

The LM567 and LM567C are general purpose tone decoders designed to provide a saturated transistor switch to ground when an input signal is present within the passband. The circuit consists of an I and Q detector driven by a voltage controlled oscillator which determines the center frequency of the decoder. External components are used to independently set center frequency, bandwidth and output delay.

Features

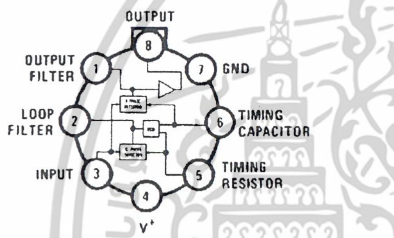
- 20 to 1 frequency range with an external resistor
- Logic compatible output with 100 mA current sinking capability
- Bandwidth adjustable from 0 to 14%
- High rejection of out of band signals and noise
- Immunity to false signals
- Highly stable center frequency
- Center frequency adjustable from 0.01 Hz to 500 kHz

Applications

- Touch tone decoding
- Precision oscillator
- Frequency monitoring and control
- Wide band FSK demodulation
- Ultrasonic controls
- Carrier current remote controls
- Communications paging decoders

Connection Diagrams

Metal Can Package

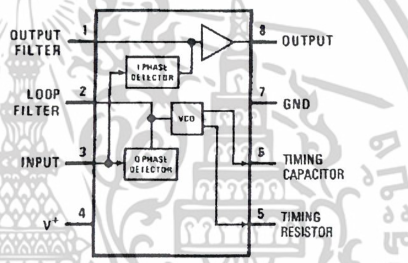


TL/H/6975–1

Top View

Order Number LM567H or LM567CH
See NS Package Number H08C

Dual-In-Line and Small Outline Packages



TL/H/6975–2

Top View

Order Number LM567CM
See NS Package Number M08A
Order Number LM567CN
See NS Package Number N08E

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage Pin	9V
Power Dissipation (Note 1)	1100 mW
V_8	15V
V_3	−10V
V_3	V_4 + 0.5V
Storage Temperature Range	−65°C to +150°C
Operating Temperature Range	
LM567H	−55°C to +125°C
LM567CH, LM567CM, LM567CN	0°C to +70°C

Soldering Information	
Dual-In-Line Package	
Soldering (10 sec.)	260°C
Small Outline Package	
Vapor Phase (60 sec.)	215°C
Infrared (15 sec.)	220°C

See AN-450 "*Surface Mounting Methods and Their Effect on Product Reliability*" for other methods of soldering surface mount devices.

Electrical Characteristics AC Test Circuit, T_A = 25°C, V^+ = 5V

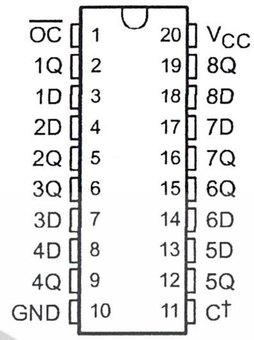
Parameters	Conditions	LM567			LM567C/LM567CM			Units
		Min	Typ	Max	Min	Typ	Max	
Power Supply Voltage Range		4.75	5.0	9.0	4.75	5.0	9.0	V
Power Supply Current Quiescent	R_L = 20k		6	8		7	10	mA
Power Supply Current Activated	R_L = 20k		11	13		12	15	mA
Input Resistance		18	20		15	20		kΩ
Smallest Detectable Input Voltage	I_L = 100 mA, $f_i = f_o$		20	25		20	25	mVrms
Largest No Output Input Voltage	I_C = 100 mA, $f_i = f_o$	10	15		10	15		mVrms
Largest Simultaneous Outband Signal to Inband Signal Ratio			6			6		dB
Minimum Input Signal to Wideband Noise Ratio	B_n = 140 kHz		−6			−6		dB
Largest Detection Bandwidth		12	14	16	10	14	18	% of f_o
Largest Detection Bandwidth Skew			1	2		2	3	% of f_o
Largest Detection Bandwidth Variation with Temperature			±0.1			±0.1		%/°C
Largest Detection Bandwidth Variation with Supply Voltage	4.75 − 6.75V		±1	±2		±1	±5	%V
Highest Center Frequency		100	500		100	500		kHz
Center Frequency Stability (4.75–5.75V)	$0 < T_A < 70$ $-55 < T_A < +125$		35 ± 60 35 ± 140			35 ± 60 35 ± 140		ppm/°C ppm/°C
Center Frequency Shift with Supply Voltage	4.75V − 6.75V 4.75V − 9V		0.5	1.0 2.0		0.4	2.0 2.0	%/V %/V
Fastest ON-OFF Cycling Rate			f_o/20			f_o/20		
Output Leakage Current	V_8 = 15V		0.01	25		0.01	25	μA
Output Saturation Voltage	e_i = 25 mV, I_8 = 30 mA e_i = 25 mV, I_8 = 100 mA		0.2 0.6	0.4 1.0		0.2 0.6	0.4 1.0	V
Output Fall Time			30			30		ns
Output Rise Time			150			150		ns

Note 1: The maximum junction temperature of the LM567 and LM567C is 150°C. For operating at elevated temperatures, devices in the TO-5 package must be derated based on a thermal resistance of 150°C/W, junction to ambient or 45°C/W, junction to case. For the DIP the device must be derated based on a thermal resistance of 110°C/W, junction to ambient. For the Small Outline package, the device must be derated based on a thermal resistance of 160°C/W, junction to ambient.

Note 2: Refer to RETS567X drawing for specifications of military LM567H version.

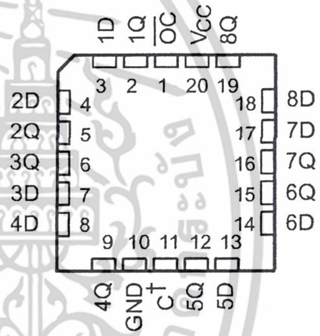
- **Choice of Eight Latches or Eight D-Type Flip-Flops in a Single Package**
- **3-State Bus-Driving Outputs**
- **Full Parallel Access for Loading**
- **Buffered Control Inputs**
- **Clock-Enable Input Has Hysteresis to Improve Noise Rejection ('S373 and 'S374)**
- **P-N-P Inputs Reduce DC Loading on Data Lines ('S373 and 'S374)**

SN54LS373, SN54LS374, SN54S373, SN54S374 . . . J OR W PACKAGE
SN74LS373, SN74LS374, SN74S374 . . . DW, N, OR NS PACKAGE
SN74S373 . . . DW OR N PACKAGE
(TOP VIEW)



† C for 'LS373 and 'S373; CLK for 'LS374 and 'S374.

SN54LS373, SN54LS374, SN54S373, SN54S374 . . . FK PACKAGE
(TOP VIEW)



† C for 'LS373 and 'S373; CLK for 'LS374 and 'S374.

description

These 8-bit registers feature 3-state outputs designed specifically for driving highly capacitive or relatively low-impedance loads. The high-impedance 3-state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pullup components. These devices are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches, meaning that while the enable (C or CLK) input is high, the Q outputs follow the data (D) inputs. When C or CLK is taken low, the output is latched at the level of the data that was set up.

The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs are set to the logic states that were set up at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines of the 'S373 and 'S374 devices simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output-control ($\overline{OC}$) input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or the high-impedance state. In the high-impedance state, the outputs neither load nor drive the bus lines significantly.

$\overline{OC}$ does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered, even while the outputs are off.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP – N	Tube	SN74LS373N	SN74LS373N
		Tube	SN74LS374N	SN74LS374N
		Tube	SN74S373N	SN74S373N
		Tube	SN74S374N	SN74S374N
	SOIC – DW	Tube	SN74LS373DW	LS373
		Tape and reel	SN74LS373DWR	
		Tube	SN74LS374DW	LS374
		Tape and reel	SN74LS374DWR	
		Tube	SN74S373DW	S373
		Tape and reel	SN74S373DWR	
		Tube	SN74S374DW	S374
		Tape and reel	SN74S374DWR	
	SOP – NS	Tape and reel	SN74LS373NSR	74LS373
		Tape and reel	SN74LS374NSR	74LS374
		Tape and reel	SN74S374NSR	74S374
–55°C to 125°C	CDIP – J	Tube	SN54LS373J	SN54LS373J
		Tube	SNJ54LS373J	SNJ54LS373J
		Tube	SN54LS374J	SN54LS374J
		Tube	SNJ54LS374J	SNJ54LS374J
		Tube	SN54S373J	SN54S373J
		Tube	SNJ54S373J	SNJ54S373J
		Tube	SN54S374J	SN54S374J
		Tube	SNJ54S374J	SNJ54S374J
	CFP – W	Tube	SNJ54LS373W	SNJ54LS373W
		Tube	SNJ54LS374W	SNJ54LS374W
		Tube	SNJ54S374W	SNJ54S374W
	LCCC – FK	Tube	SNJ54LS373FK	SNJ54LS373FK
		Tube	SNJ54LS374FK	SNJ54LS374FK
		Tube	SNJ54S373FK	SNJ54S373FK
		Tube	SNJ54S374FK	SNJ54S374FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Function Tables

'LS373, 'S373
(each latch)

INPUTS			OUTPUT
$\overline{OC}$	C	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

'LS374, 'S374
(each latch)

INPUTS			OUTPUT
$\overline{OC}$	CLK	D	Q
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

HM62256B Series

32,768-word × 8-bit High Speed CMOS Static RAM

HITACHI

ADE-203-135D (Z)
Rev. 4.0
Nov. 29, 1995

Description

The Hitachi HM62256B is a CMOS static RAM organized 32-kword × 8-bit. It realizes higher performance and low power consumption by employing 0.8 μm Hi-CMOS process technology. The device, packaged in 8 × 14 mm TSOP, 8 × 13.4 mm TSOP with thickness of 1.2 mm, 450-mil SOP (foot print pitch width), 600-mil plastic DIP, or 300-mil plastic DIP, is available for high density mounting. It offers low power standby power dissipation; therefore, it is suitable for battery back-up systems.

Features

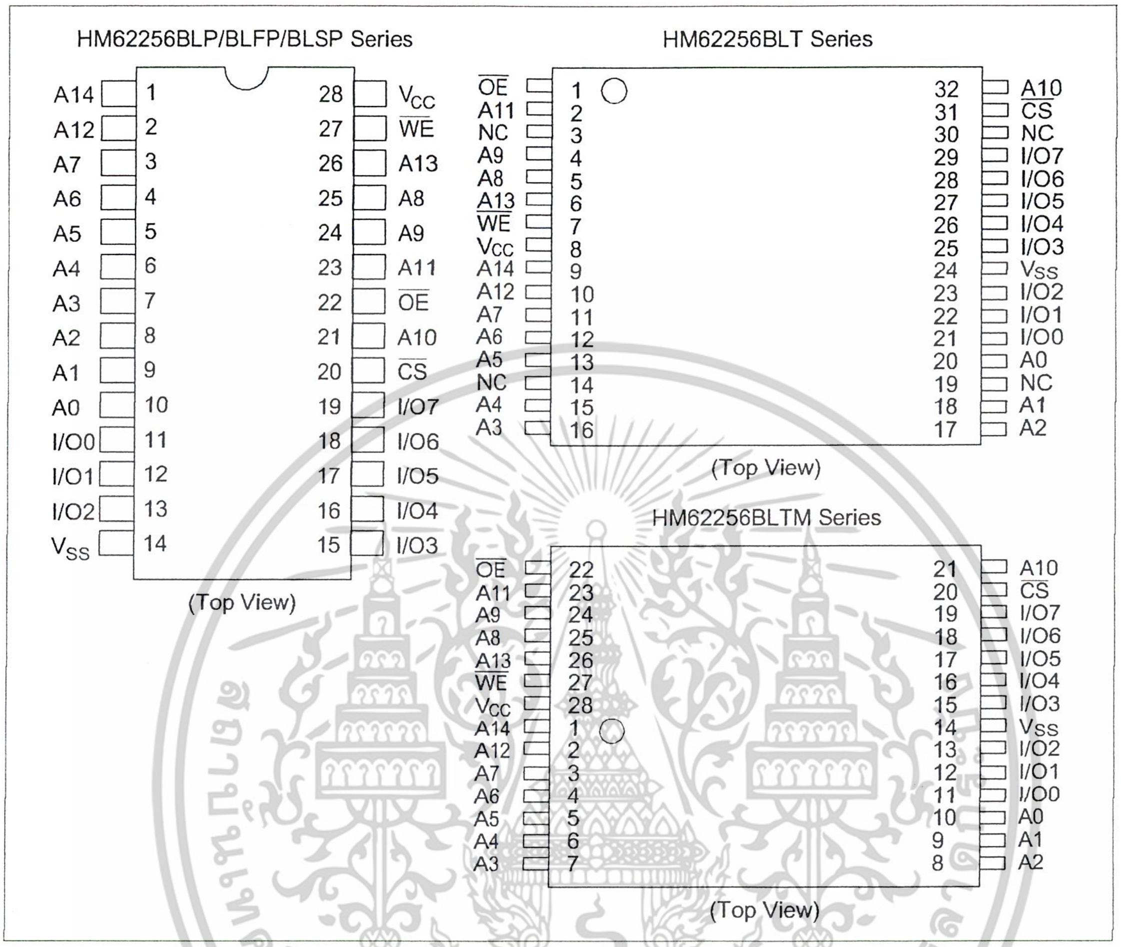
- High speed
  Fast access time: 45/55/70/85 ns (max)
- Low power
  Standby: 1.0 μW (typ)
  Operation: 25 mW (typ) (f = 1 MHz)
- Single 5 V supply
- Completely static memory
  No clock or timing strobe required
- Equal access and cycle times
- Common data input and output
  Three state output
- Directly TTL compatible
  All inputs and outputs
- Capability of battery back up operation

Ordering Information

Type No.	Access Time	Package
HM62256BLP-7	70 ns	600-mil 28-pin plastic DIP (DP-28)
HM62256BLP-7SL	70 ns	
HM62256BLSP-7	70 ns	300-mil 28-pin plastic DIP (DP-28NA)
HM62256BLSP-7SL	70 ns	
HM62256BLFP-7T	70 ns	450-mil 28-pin plastic SOP (FP-28DA)
HM62256BLFP-4SLT*1 HM62256BLFP-5SLT HM62256BLFP-7SLT	45 ns 55 ns 70 ns	
HM62256BLFP-7ULT	70 ns	
HM62256BLT-8	85 ns	8 mm × 14 mm 32-pin TSOP (TFP-32DA)
HM62256BLT-7SL	70 ns	
HM62256BLTM-8	85 ns	8 mm × 13.4 mm 28-pin TSOP (TFP-28DA)
HM62256BLTM-4SL*1 HM62256BLTM-5SL HM62256BLTM-7SL	45 ns 55 ns 70 ns	
HM62256BLTM-7UL	70 ns	

Note: 1. Under development

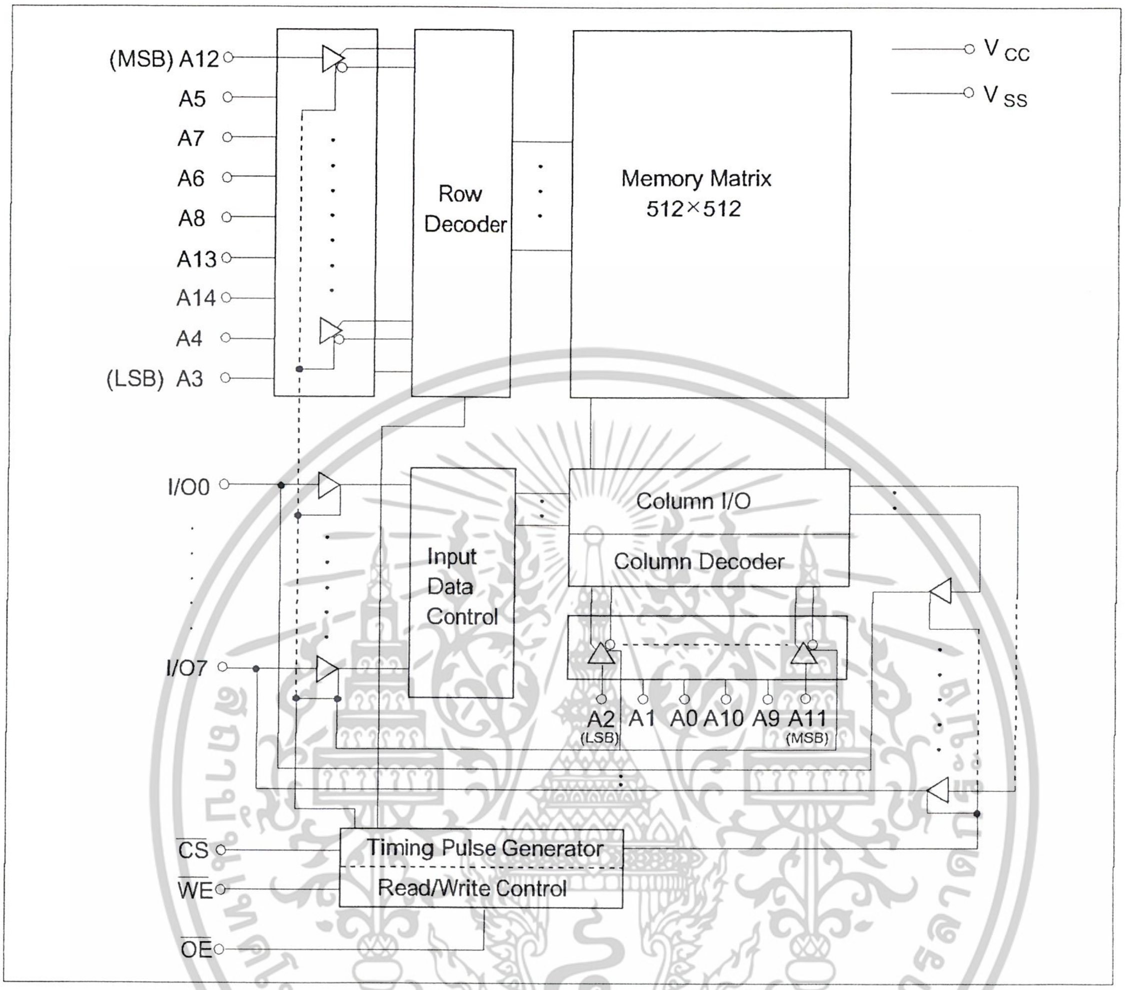
Pin Arrangement



Pin Description

Symbol	Function
A0 – A14	Address
I/O0 – I/O7	Input/output
$\overline{CS}$	Chip select
$\overline{WE}$	Write enable
$\overline{OE}$	Output enable
NC	No connection
V_{CC}	Power supply
V_{SS}	Ground

Block Diagram



Function Table

$\overline{WE}$	$\overline{CS}$	$\overline{OE}$	Mode	V_{cc} Current	I/O Pin	Ref. Cycle
X	H	X	Not selected	I_{SB}, I_{SB1}	High-Z	—
H	L	H	Output disable	I_{CC}	High-Z	—
H	L	L	Read	I_{CC}	Dout	Read cycle (1)–(3)
L	L	H	Write	I_{CC}	Din	Write cycle (1)
L	L	L	Write	I_{CC}	Din	Write cycle (2)

Note: X: H or L

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Power supply voltage[*1]	V_{CC}	–0.5 to +7.0	V
Terminal voltage[*1]	V_T	–0.5[*2] to V_{CC} + 0.3[*3]	V
Power dissipation	P_T	1.0	W
Operating temperature	Topr	0 to + 70	°C
Storage temperature	Tstg	–55 to +125	°C
Storage temperature under bias	Tbias	–10 to +85	°C

Notes: 1. Relative to V_{SS}
2. V_T min: –3.0 V for pulse half-width ≤ 50 ns
3. Maximum voltage is 7.0 V

Recommended DC Operating Conditions (Ta = 0 to +70°C)

Parameter	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{CC}	4.5	5.0	5.5	V
	V_{SS}	0	0	0	V
Input high (logic 1) voltage	V_{IH}	2.2	—	V_{CC}+0.3	V
Input low (logic 0) voltage	V_{IL}	–0.5[*1]	—	0.8	V

Note: 1. V_{IL} min: –3.0 V for pulse half-width ≤ 50 ns

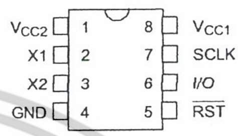
DALLAS SEMICONDUCTOR

DS1302
Trickle Charge Timekeeping Chip

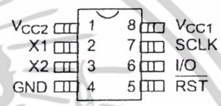
FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 31 x 8 RAM for scratchpad data storage
- Serial I/O for minimum pin count
- 2.5–5.5 volt full operation
  - *Optional 2.0–5.5 volt full operation also available*
- Uses less than 300 nA at 2.5 volts
- Single–byte or multiple–byte (burst mode) data transfer for read or write of clock or RAM data
- 8–pin DIP or optional 8–pin SOIC's for surface mount
- Simple 3–wire interface
- TTL–compatible (V_{CC} = 5V)
- Optional industrial temperature range –40°C to +85°C
- DS1202 compatible
- Added features over DS1202
  - *Optional trickle charge capability to* V_{CC1}
  - Dual power supply pins for primary and backup power supplies
  - Backup power supply pin can be used for battery or super cap input
  - Additional scratchpad memory (7 bytes)

PIN ASSIGNMENT



DS1302
8–PIN DIP (300 MIL)



DS1302S 8–PIN SOIC (200 MIL)
DS1302Z 8–PIN SOIC (150 MIL)

PIN DESCRIPTION

X1, X2	–	32.768 kHz Crystal Pins
GND	–	Ground
$\overline{RST}$	–	Reset
I/O	–	Data Input/Output
SCLK	–	Serial Clock
V_{CC1}, V_{CC2}	–	Power Supply Pins

ORDERING INFORMATION

PART #	DESCRIPTION
DS1302	Serial Timekeeping Chip; 8–pin DIP
DS1302S	Serial Timekeeping Chip; 8–pin SOIC (200 mil)
DS1302Z	Serial Timekeeping Chip; 8–pin SOIC (150 mil)

DESCRIPTION

The DS1302 Trickle Charge Timekeeping Chip contains a real time clock/calendar and 31 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The real time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24–hour or 12–hour format with an AM/PM indicator.

Interfacing the DS1302 with a microprocessor is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: (1) $\overline{RST}$ (Reset), (2) I/O (Data line), and (3) SCLK (Serial clock). Data can be transferred to and from the clock/RAM one byte at a time or in a burst of up to 31 bytes. The DS1302 is designed to operate on very low power and retain data and clock information on less than 1 microwatt.

The DS1302 is the successor to the DS1202. In addition to the basic timekeeping functions of the DS1202, the DS1302 has the additional features of dual power pins for primary and back–up power supplies, programmable trickle charger for V_{CC1}, and seven additional bytes of scratchpad memory.

OPERATION

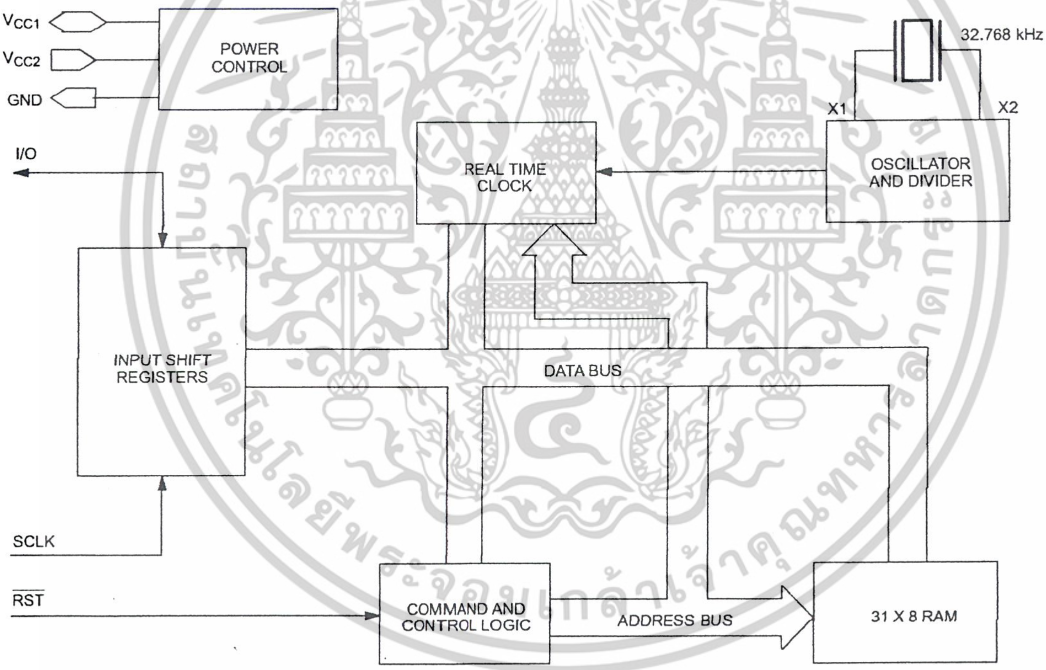
The main elements of the Serial Timekeeper are shown in Figure 1: shift register, control logic, oscillator, real time clock, and RAM. To initiate any transfer of data, $\overline{RST}$ is taken high and eight bits are loaded into the shift register providing both address and command information. Data is serially input on the rising edge of the SCLK. The first eight bits specify which of 40 bytes will be accessed, whether a read or write cycle will take place, and whether a byte or burst mode transfer is to occur.

After the first eight clock cycles have loaded the command word into the shift register, additional clocks will output data for a read or input data for a write. The number of clock pulses equals eight plus eight for byte mode or eight plus up to 248 for burst mode.

COMMAND BYTE

The command byte is shown in Figure 2. Each data transfer is initiated by a command byte. The MSB (Bit 7) must be a logic "1". If it is zero, writes to the DS1302 will be disabled. Bit 6 specifies clock/calendar data if logic "0" or RAM data if logic "1". Bits one through five specify the designated registers to be input or output, and the LSB (Bit 0) specifies a write operation (input) if logic "0" or read operation (output) if logic "1". The command byte is always input starting with the LSB (Bit 0).

DS1302 BLOCK DIAGRAM Figure 1



ADDRESS/COMMAND BYTE Figure 2

7	6	5	4	3	2	1	0
1	RAM / $\overline{CK}$	A4	A3	A2	A1	A0	RD / $\overline{W}$

RESET AND CLOCK CONTROL

All data transfers are initiated by driving the $\overline{RST}$ input high. The $\overline{RST}$ input serves two functions. First, $\overline{RST}$ turns on the control logic which allows access to the shift register for the address/command sequence. Second, the $\overline{RST}$ signal provides a method of terminating either single byte or multiple byte data transfer.

A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, data must be valid during the rising edge of the clock and data bits are output on the falling edge of clock. If the $\overline{RST}$ input is low all data transfer terminates and the I/O pin goes to a high impedance state. Data transfer is illustrated in Figure 3. At power–up, $\overline{RST}$ must be a logic "0" until $V_{CC} \geqq 2.5$ volts. Also SCLK must be at a logic "0" when $\overline{RST}$ is driven to a logic "1" state.

DATA INPUT

Following the eight SCLK cycles that input a write command byte, a data byte is input on the rising edge of the next eight SCLK cycles. Additional SCLK cycles are ignored should they inadvertently occur. Data is input starting with bit 0.

DATA OUTPUT

Following the eight SCLK cycles that input a read command byte, a data byte is output on the falling edge of the next eight SCLK cycles. Note that the first data bit to be transmitted occurs on the first falling edge after the last bit of the command byte is written. Additional SCLK cycles retransmit the data bytes should they inadvertently occur so long as $\overline{RST}$ remains high. This operation permits continuous burst mode read capability. Also, the I/O pin is tri–stated upon each rising edge of SCLK. Data is output starting with bit 0.

BURST MODE

Burst mode may be specified for either the clock/calendar or the RAM registers by addressing location 31 decimal (address/command bits one through five = logical one). As before, bit six specifies clock or RAM and bit 0 specifies read or write. There is no data storage capacity at locations 9 through 31 in the Clock/Calendar Registers or location 31 in the RAM registers. Reads or writes in burst mode start with bit 0 of address 0.

As in the case with the DS1202, when writing to the clock registers in the burst mode, the first eight registers must be written in order for the data to be transferred. However, when writing to RAM in burst mode it is not necessary to write all 31 bytes for the data to transfer. Each byte that is written to will be transferred to RAM regardless of whether all 31 bytes are written or not.

CLOCK/CALENDAR

The clock/calendar is contained in seven write/read registers as shown in Figure 4. Data contained in the clock/calendar registers is in binary coded decimal format (BCD).

CLOCK HALT FLAG

Bit 7 of the seconds register is defined as the clock halt flag. When this bit is set to logic "1", the clock oscillator is stopped and the DS1302 is placed into a low–power standby mode with a current drain of less than 100 nanoamps. When this bit is written to logic "0", the clock will start.

AM-PM/12-24 MODE

Bit 7 of the hours register is defined as the 12– or 24–hour mode select bit. When high, the 12–hour mode is selected. In the 12–hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24–hour mode, bit 5 is the second 10 hour bit (20 – 23 hours).

WRITE PROTECT BIT

Bit 7 of the control register is the write protect bit. The first seven bits (bits 0 – 6) are forced to zero and will always read a zero when read. Before any write operation to the clock or RAM, bit 7 must be zero. When high, the write protect bit prevents a write operation to any other register.

TRICKLE CHARGE REGISTER

This register controls the trickle charge characteristics of the DS1302. The simplified schematic of Figure 5 shows the basic components of the trickle charger. The trickle charge select (TCS) bits (bits 4 – 7) control the selection of the trickle charger. In order to prevent accidental enabling, only a pattern of 1010 will enable the trickle charger. All other patterns will disable the trickle charger. The DS1302 powers up with the trickle charger disabled. The diode select (DS) bits (bits 2 – 3) select whether one diode or two diodes are connected between V_{CC2} and V_{CC1}. If DS is 01, one diode is selected or if DS is 10, two diodes are selected. If DS is 00 or 11, the trickle charger is disabled independent of

TCS. The RS bits (bits 0 – 1) select the resistor that is connected between V_{CC2} and V_{CC1}. The resistor selected by the resistor select (RS) bits is as follows:

RS Bits	Resistor	Typical Value
00	None	None
01	R1	2KΩ
10	R2	4KΩ
11	R3	8KΩ

If RS is 00, the trickle charger is disabled independent of TCS.

Diode and resistor selection is determined by the user according to the maximum current desired for battery or super cap charging. The maximum charging current can be calculated as illustrated in the following example. Assume that a system power supply of 5V is applied to V_{CC2} and a super cap is connected to V_{CC1}. Also assume that the trickle charger has been enabled with 1 diode and resistor R1 between V_{CC2} and V_{CC1}. The maximum current I_{max} would therefore be calculated as follows:

$$I_{max} = (5.0V - \text{diode drop}) / R1$$
$$\sim (5.0V - 0.7V) / 2K\Omega$$
$$\sim 2.2\text{ mA}$$

Obviously, as the super cap charges, the voltage drop between V_{CC2} and V_{CC1} will decrease and therefore the charge current will decrease.

CLOCK/CALENDAR BURST MODE

The clock/calendar command byte specifies burst mode operation. In this mode the first eight clock/calendar registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

If the write protect bit is set high when a write clock/calendar burst mode is specified, no data transfer will occur to any of the eight clock/calendar registers (this includes the control register). The trickle charger is not accessible in burst mode.

RAM

The static RAM is 31 x 8 bytes addressed consecutively in the RAM address space.

RAM BURST MODE

The RAM command byte specifies burst mode operation. In this mode, the 31 RAM registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

REGISTER SUMMARY

A register data format summary is shown in Figure 4.

CRYSTAL SELECTION

A 32.768 kHz crystal can be directly connected to the DS1302 via pins 2 and 3 (X1, X2). The crystal selected for use should have a specified load capacitance (CL) of 6 pF.

POWER CONTROL

V_{CC1} provides low power operation in single supply and battery operated systems as well as low power battery backup.

V_{CC2} provides the primary power in dual supply systems where V_{CC1} is connected to a backup source to maintain the time and data in the absence of primary power.

The DS1302 will operate from the larger of V_{CC1} or V_{CC2}. When V_{CC2} is greater than V_{CC1} + 0.2V, V_{CC2} will power the DS1302. When V_{CC2} is less than V_{CC1}, V_{CC1} will power the DS1302.

19-0094; Rev 6; 7/95

MAXIM

Microprocessor Supervisory Circuits

MAX691A/MAX693A/MAX800L/MAX800M

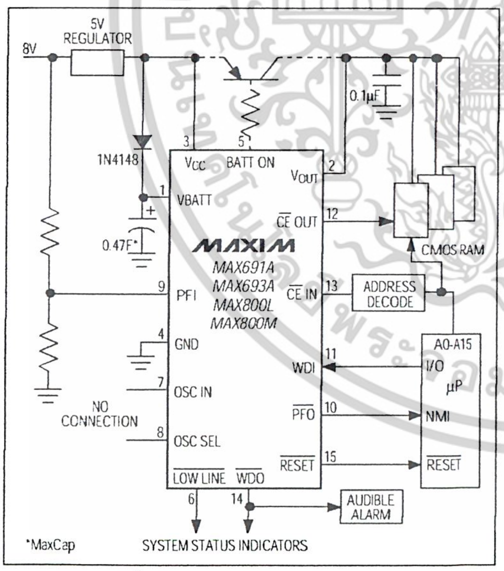
General Description

The MAX691A/MAX693A/MAX800L/MAX800M microprocessor (μP) supervisory circuits are pin-compatible upgrades to the MAX691, MAX693, and MAX695. They improve performance with 30μA supply current, 200ms typ reset active delay on power-up, and 6ns chip-enable propagation delay. Features include write protection of CMOS RAM or EEPROM, separate watchdog outputs, backup-battery switchover, and a $\overline{RESET}$ output that is valid with V_{CC} down to 1V. The MAX691A/MAX800L have a 4.65V typical reset-threshold voltage, and the MAX693A/MAX800M's reset threshold is 4.4V typical. The MAX800L/MAX800M guarantee power-fail accuracies to ±2%.

Applications

Computers
Controllers
Intelligent Instruments
Automotive Systems
Critical μP Power Monitoring

Typical Operating Circuit



Features

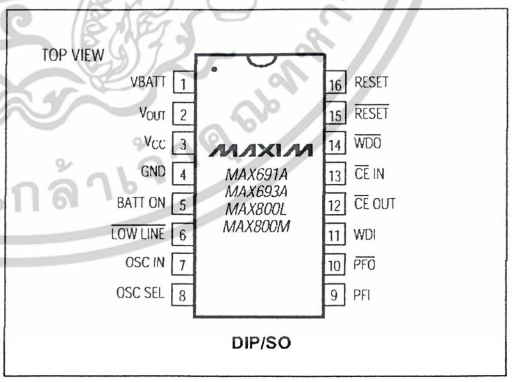
- ♦ 200ms Power-OK/Reset Timeout Period
- ♦ 1μA Standby Current, 30μA Operating Current
- ♦ On-Board Gating of Chip-Enable Signals, 10ns Max Delay
- ♦ MaxCap™ or SuperCap™ Compatible
- ♦ Guaranteed $\overline{RESET}$ Assertion to V_{CC} = 1V
- ♦ Voltage Monitor for Power-Fail or Low-Battery Warning
- ♦ Power-Fail Accuracy Guaranteed to ±2% (MAX800L/M)
- ♦ Available in 16-Pin Narrow SO and Plastic DIP Packages

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX691ACPE	0°C to +70°C	16 Plastic DIP
MAX691ACSE	0°C to +70°C	16 Narrow SO
MAX691ACWE	0°C to +70°C	16 Wide SO
MAX691AC/D	0°C to +70°C	Dice*
MAX691AEPE	-40°C to +85°C	16 Plastic SO
MAX691AESE	-40°C to +85°C	16 Narrow SO
MAX691AEWE	-40°C to +85°C	16 Wide SO
MAX691AEJE	-40°C to +85°C	16 CERDIP
MAX691AMJE	-55°C to +125°C	16 CERDIP

Ordering Information continued on last page.
** Dice are specified at T_A = +25 °C.*

Pin Configuration



Microprocessor Supervisory Circuits

ABSOLUTE MAXIMUM RATINGS

Terminal Voltage (with respect to GND)
- V_{CC} ....-0.3V to +6V
- VBATT...-0.3V to +6V
- All Other Inputs ....-0.3V to (V_{OUT} + 0.3V)

Input Current
- V_{CC} Peak...1.0A
- V_{CC} Continuous...250mA
- VBATT Peak 250mA
- VBATT Continuous 25mA
- GND, BATT ON 100mA
- All Other Outputs 25mA

Continuous Power Dissipation (T_A = +70°C)
- Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
- Narrow SO (derate 8.70mW/°C above +70°C) 696mW
- Wide SO (derate 9.52mW/°C above +70°C)...............762mW
- CERDIP (derate 10.00mW/°C above +70°C).............800mW

Operating Temperature Ranges
- MAX69_AC_ _/MAX800_C_ _ ....0°C to +70°C
- MAX69_AE_ _/MAX800_E_ _ ....-40°C to +85°C
- MAX69_AMJE ....-55°C to +125°C

Storage Temperature Range ....-65°C to +160°C

Lead Temperature (soldering, 10sec) ....+300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(MAX691A, MAX800L: V_{CC} = 4.75V to 5.5V, MAX693A, MAX800M: V_{CC} = 4.5V to 5.5V, VBATT = 2.8V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS			MIN	TYP	MAX	UNITS
Operating Voltage Range, V_{CC}, VBATT (Note 1)				0		5.5	V
V_{OUT} Output	V_{CC} = 4.5V	I_{OUT} = 25mA		V_{CC} - 0.05	V_{CC} - 0.02		V
		I_{OUT} = 250mA	MAX69_AC/AE, MAX800_C/E	V_{CC} - 0.3	V_{CC} - 0.2		
			MAX69_AM	V_{CC} - 0.40			
V_{CC}-to-V_{OUT} On-Resistance	V_{CC} = 4.5V	MAX69_AC/AE, MAX800_C/E			0.8	1.2	Ω
		MAX69_AM			0.8	1.6	
V_{OUT} in Battery-Backup Mode	VBATT = 4.5V, I_{OUT} = 20mA			VBATT - 0.3			V
	VBATT = 2.8V, I_{OUT} = 10mA			VBATT - 0.25			
	VBATT = 2.0V, I_{OUT} = 5mA			VBATT - 0.15			
VBATT-to-V_{OUT} On-Resistance	VBATT = 4.5V					15	Ω
	VBATT = 2.8V					25	
	VBATT = 2.0V					30	
Supply Current in Normal Operating Mode (Excludes I_{OUT})	V_{CC} > VBATT - 1V				30	100	µA
Supply Current in Battery-Backup Mode (Excludes I_{OUT}) (Note 2)	V_{CC} < VBATT - 1.2V, VBATT = 2.8V	T_A = +25°C			0.04	1	µA
		T_A = T_{MIN} + T_{MIN}				5	
VBATT Standby Current (Note 3)	VBATT + 0.2V ≤ V_{CC}	T_A = +25°C		-0.1		0.02	µA
		T_A = T_{MIN} + T_{MIN}		-1.0		0.02	
Battery Switchover Threshold	Power-up				VBATT + 0.3		V
	Power-down				VBATT - 0.3		
Battery Switchover Hysteresis					60		mV
BATT ON Output Low Voltage	I_{SINK} = 3.2mA				0.1	0.4	V
	I_{SINK} = 25mA				0.7	1.5	
BATT ON Output Short-Circuit Current	Sink current				60		mA
	Source current			1	15	100	µA

ELECTRICAL CHARACTERISTICS (continued)

(MAX691A, MAX800L: V_{CC} = 4.75V to 5.5V, MAX693A, MAX800M: V_{CC} = 4.5V to 5.5V, VBATT = 2.8V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
RESET AND WATCHDOG TIMER					
Reset Threshold Voltage	MAX691A, MAX800L	4.50	4.65	4.75	V
	MAX693A, MAX800M	4.25	4.40	4.50	
	MAX800L, T_A = +25°C, V_{CC} falling	4.55		4.70	
	MAX800M, T_A = +25°C, V_{CC} falling	4.30		4.45	
Reset Threshold Hysteresis			15		mV
V_{CC} to RESET Delay	Power-down		80		µs
$\overline{\text{LOWLINE}}$-to-$\overline{\text{RESET}}$ Delay			800		ns
Reset Active Timeout Period, Internal Oscillator	Power-up	140	200	280	ms
Reset Active Timeout Period, External Clock (Note 4)	Power-up		2048		Clock Cycles
Watchdog Timeout Period, Internal Oscillator	Long period	1.0	1.6	2.25	sec
	Short Period	70	100	140	ms
Watchdog Timeout Period, External Clock (Note 4)	Long Period		4096		Clock Cycles
	Short Period		1024		
Minimum Watchdog Input Pulse Width	V_{IL} = 0.8V, V_{IH} = 0.75 x V_{CC}	100			ns
$\overline{\text{RESET}}$ Output Voltage	I_{SINK} = 50µA, V_{CC} = 1V, VBATT = 0V, V_{CC} falling		0.004	0.3	V
	I_{SINK} = 3.2mA, V_{CC} = 4.25V		0.1	0.4	
	I_{SOURCE} = 1.6mA, V_{CC} = 5V	3.5			
$\overline{\text{RESET}}$ Output Short-Circuit Current	Output source current		7	20	mA
RESET Output Voltage Low (Note 5)	I_{SINK} = 3.2mA	0.1	0.4		V
$\overline{\text{LOW LINE}}$ Output Voltage	I_{SINK} = 3.2mA, V_{CC} = 4.25V			0.4	V
	I_{SOURCE} = 1µA, V_{CC} = 5V	3.5			
$\overline{\text{LOW LINE}}$ Output Short-Circuit Current	Output source current	1	15	100	µA
$\overline{\text{WDO}}$ Output Voltage	I_{SINK} = 3.2mA			0.4	V
	I_{SOURCE} = 500µA, V_{CC} = 5V	3.5			
$\overline{\text{WDO}}$ Output Short-Circuit Current	Output source current		3	10	mA
WDI Threshold Voltage (Note 6)	V_{IH}	0.75 x V_{CC}			V
	V_{IL}			0.8	
WDI Input Current	WDI = 0V	-50	-10		µA
	WDI = V_{OUT}		20	50	

ELECTRICAL CHARACTERISTICS (continued)

(MAX691A, MAX800L: V_{CC} = 4.75V to 5.5V, MAX693A, MAX800M: V_{CC} = 4.5V to 5.5V, VBATT = 2.8V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
POWER-FAIL COMPARATOR					
PFI Input Threshold	MAX69_AC/AE/AM, V_{CC} = 5V	1.2	1.25	1.3	V
	MAX800_C/E, V_{CC} = 5V	1.225	1.25	1.275	
PFI Leakage Current			±0.01	±25	nA
$\overline{PFO}$ Output Voltage	I_{SINK} = 3.2mA			0.4	V
	I_{SOURCE} = 1µA, V_{CC} = 5V	3.5			
$\overline{PFO}$ Output Short-Circuit Current	Output source current	1	15	100	µA
PFI-to-PFO Delay	V_{IN} = -20mV, V_{OD} = 15mV		25		µs
	V_{IN} = 20mV, V_{OD} = 15mV		60		
CHIP-ENABLE GATING					
$\overline{CE}$ IN Leakage Current	Disable mode		±0.005	±1	µA
$\overline{CE}$ IN-to-$\overline{CE}$ OUT Resistance (Note 7)	Enable mode		75	150	Ω
$\overline{CE}$ OUT Short-Circuit Current (Reset Active)	Disable mode, $\overline{CE}$ OUT = 0V	0.1	0.75	2.0	mA
$\overline{CE}$ IN-to-$\overline{CE}$ OUT Propagation Delay (Note 8)	50Ω source impedance driver, C_{LOAD} = 50pF		6	10	ns
$\overline{CE}$ OUT Output Voltage High (Reset Active)	V_{CC} = 5V, I_{OUT} = -100µA	3.5			V
	V_{CC} = 0V, VBATT = 2.8V, I_{OUT} = 1µA	2.7			
RESET-to-$\overline{CE}$ OUT Delay	Power-down		12		µs
INTERNAL OSCILLATOR					
OSC IN Leakage Current	OSC SEL = 0V		0.10	±5	µA
OSC IN Input Pull-Up Current	OSC SEL = V_{OUT} or floating, OSC IN = 0V		10	100	µA
OSC SEL Input Pull-Up Current	OSC SEL = 0V		10	100	µA
OSC IN Frequency Range	OSC SEL = 0V		50		kHz
OSC IN External Oscillator Threshold Voltage	V_{IH}	V_{OUT} - 0.4	V_{OUT} - 0.6		V
	V_{IL}		3.65	2.00	
OSC IN Frequency with External Capacitor	OSC SEL = 0V, COSC = 47pF		100		kHz

Note 1: Either V_{CC} or VBATT can go to 0V, if the other is greater than 2.0V.

Note 2: The supply current drawn by the MAX691A/MAX800L/MAX800M from the battery excluding I_{OUT} typically goes to 10µA when (VBATT - 1V) < V_{CC} < VBATT. In most applications, this is a brief period as V_{CC} falls through this region.

Note 3: "+" = battery-discharging current, "-" = battery-charging current.

Note 4: Although presented as typical values, the number of clock cycles for the reset and watchdog timeout periods are fixed and do not vary with process or temperature.

Note 5: RESET is an open-drain output and sinks current only.

Note 6: WDI is internally connected to a voltage divider between V_{OUT} and GND. If unconnected, WDI is driven to 1.6V (typ), disabling the watchdog function.

Note 7: The chip-enable resistance is tested with V_{CC} = 4.75V for the MAX691A/MAX800L and V_{CC} = 4.5V for the MAX693A/MAX800M. $\overline{CE}$ IN = $\overline{CE}$ OUT = V_{CC} / 2.

Note 8: The chip-enable propagation delay is measured from the 50% point at $\overline{CE}$ IN to the 50% point at $\overline{CE}$ OUT.

FAIRCHILD

SEMICONDUCTOR TM

August 1986

Revised March 2000

DM74LS138 • DM74LS139 Decoder/Demultiplexer

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The DM74LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The DM74LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design.

Features

- Designed specifically for high speed:
  - Memory decoders
  - Data transmission systems
- DM74LS138 3-to-8-line decoders incorporates 3 enable inputs to simplify cascading and/or data reception
- DM74LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)
  - DM74LS138 21 ns
  - DM74LS139 21 ns
- Typical power dissipation
  - DM74LS138 32 mW
  - DM74LS139 34 mW

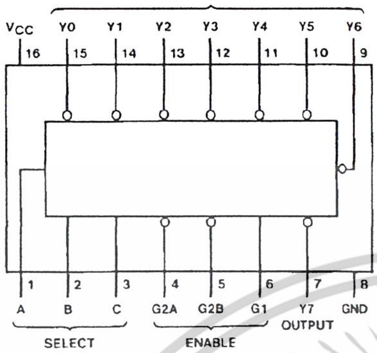
Ordering Code:

Order Number	Package Number	Package Description
DM74LS138M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS138SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS138N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
DM74LS139M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS139SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS139N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

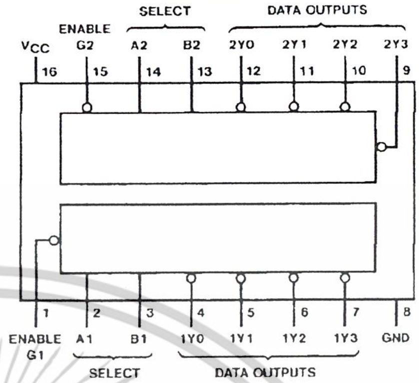
Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagrams

DM74LS138



DM74LS139



Function Tables

DM74LS138

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	YO	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

DM74LS139

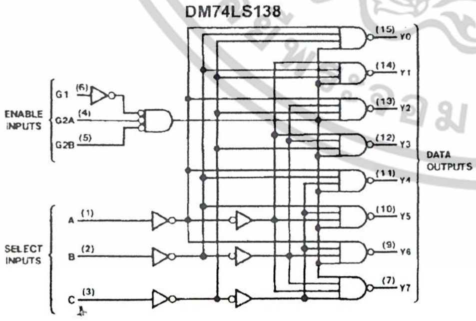
Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Level
L = LOW Level
X = Don't Care

Note 1: G2 = G2A + G2B

Logic Diagrams

DM74LS138



DM74LS139