สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบอาคารจอดรถ

PARKING LOT



โดย

นายบัณฑิต วิเชียร

นายปกรณ์ ดาราฉาย

เลขหมู่.....
เลขทะเบียน 45720
วัน, เดือน, ปี 13 ก.พ. 2546

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบอาคารจอดรถ

ผู้จัดทำ

1. นาย บัณฑิต วิเชียร 42015268
2. นาย ปกรณ์ ดาราฉาย 42015269

.....อาจารย์ที่ปรึกษา

(รศ.ดร. จงกล งามวิวิทย์)

ระบบอาคารจอดรถ

โดย
นาย บัณฑิต วิเชียร
นาย ปกรณ์ ดาราฉาย
รศ.ดร. จงกล งามวิวิทย์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

อาคารจอดรถ เป็นระบบจอดรถที่บริการตัวเอง โดยใช้บัตรแถบขาวดำ เพื่อตรวจสอบว่ารถเข้าไปจอดในช่องใดและคิดเวลาจอดเมื่อดึงบัตรออกจากกล่อง บัตรนี้จะใช้คำนวณค่าจอดที่ชั้นหนึ่งของอาคารโดยชำระด้วยเหรียญ และปลดล็อคเมื่อเสียบบัตรเข้ากล่องที่ช่องจอดรถ

สำหรับโครงงานนี้ประกอบด้วยซอฟต์แวร์และฮาร์ดแวร์ ในส่วนของซอฟต์แวร์ได้พัฒนาโปรแกรมด้วยภาษาแอซแซมบลีของไมโครคอนโทรลเลอร์ MCS-51 เพื่อแสดงผลบริเวณทางเข้าออกของอาคารว่าขณะนี้เหลือที่ว่างสำหรับจอดรถจำนวนกี่คัน คำนวณเวลาที่ใช้ในการจอดรถและค่าจอดรถ ซึ่งถ้าจอดรถชั้นสูงจะราคาถูก ถ้าจอดรถชั้นต่ำจะราคาแพง และจะแสดงค่าบริการจอดรถด้วยจอแสดงผลแอลซีดี ในส่วนของฮาร์ดแวร์ประกอบด้วยแบบจำลองอาคารจอดรถ และวงจรต่างๆ เช่น วงจรอินฟาเรดอ่านค่าบัตร วงจรขับมอเตอร์ เป็นต้น

PARKING LOT

By

Bandit Wichian

Pakorn Darachai

Associate Professor Jongkol Ngamwiwit Advisor

Academie Year : 2002

Abstract

Parking lot is the self-service parking system that uses a code card for checking the parking car in the slot and calculates the parking time when push back the card from the box. This card will show the parking fee which will be paid by coins at the first floor, and the car will be unlocked when insert the card in the box at parking slot.

This project composes of software and hardware. In the part of software; the program has been developed with assembly language of microcontroller MCS-51 for showing the vacant space for car that can park, calculating the parking time and parking fee which are shown by LCD. The parking fee depends on parking time and floor of parking area. The fee of higher floor of parking area will be paid lower than the lower floor of parking area. In the part of hardware,compose of the parking model and several circuits such as Infrared circuit for checking the card and motor drive circuit.

สารบัญ

	หน้า
บทคัดย่อ	
สารบัญภาพ	๑
สารบัญตาราง	๒
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1ทฤษฎีการอ่านค่าบัตร	2
2.2ภาคตรวจจับ	2
2.3สเต็ปมอเตอร์แบบยูนิโพลาร์	4
2.4ทฤษฎีแสดงผลแบบผลึกเหลว	6
2.5โครงงานสร้าง LCD	7
2.6คำสั่งควบคุมโมดูล LCD	8
2.7ทฤษฎี DS1307	12
2.8ไมโครคอนโทรลเลอร์	16
2.9Special Function Register	28
2.10การคำนวณความเร็วรับส่งข้อมูลแบบอนุกรม	35
2.11เบื้องต้น LED 7 ส่วน	37
บทที่ 3 การคำนวณและการสร้าง	39
3.1ขั้นตอนการปฏิบัติงาน	39
3.2ขั้นตอนการทำงานทั้งหมดของระบบ	39
บทที่ 4 ผลการทดลอง	43
บทที่ 5 บทวิจารณ์และสรุป	45
กิตติกรรมประกาศ	48
หนังสืออ้างอิง	49
ภาคผนวก	

สารบัญภาพ

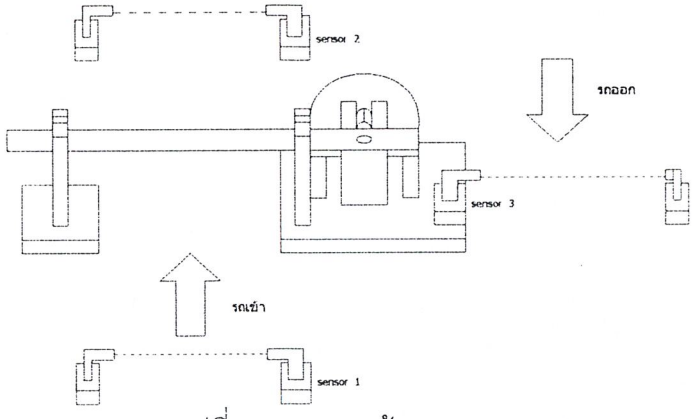
รูปที่	หน้า
1.แสดงทางเข้าออกของอาคารจอดรถ	1
2.วงจรเซนเซอร์	4
3.วงจรขับสเต็ปมอเตอร์โดย MCS 51	6
4.จอแสดงผล LCD	8
5. วงจรภายใน DS 1307	13
6.การจัดสรรหน่วยความจำ	14
7.โครงงานสร้างของ MCS 8051	17
8.สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ 8051	19
9.แสดงขาต่างๆของ 8051	21
10.โครงสร้างของพอร์ต 8051	21
11.การจัดพื้นที่หน่วยความจำสำหรับการโปรแกรมของ 8051	22
12.การจัดพื้นที่หน่วยความจำของข้อมูล	23
13.ผังเวลาของไมโครคอนโทรลเลอร์เบอร์ 8051	23
14.แสดงหน่วยความจำข้อมูลภายใน	25
15.รายละเอียดของ Speacial Function Register	26
16.ตำแหน่งการอ้างอิงระดับบิตของรีจีสเตอร์ SFR	27
- โฟวล์ชาร์ท	41
17.แสดงการทำงานทางเข้าออกของอาคาร	45
18.แสดงการทำงานของคานเมื่อทำการกดรหัสและดึงบัตรแล้ว	45
19.แสดงตัวบัตรที่ใช้ในการจอดรถและกล่องที่ใช้อ่านค่าบัตร	46
20.แสดงส่วนของการชำระค่าจอดรถ	46
21.แสดงภาพโดยรวมแบบอาคารจอดรถ	47

สารบัญตาราง

ตารางที่	หน้า
1.ไมโครคอนโทรลเลอร์ตระกูล 8051	16
2.การตอบรับการอินเตอร์รัพในตำแหน่งของหน่วยความจำ	31
3.บิตเลือกการทำงานของ TIME/COUNTER	34
4.แสดงการทำงานของ SM0 SM1	34
5.แสดงอัตราการส่งรับข้อมูลแบบอนุกรม	36
6.แสดงค่า SFR โหมด 3	36

บทที่ 1
บทนำ

ในปัจจุบันทั่วโลกต่างแข่งขันกันทางด้านอุตสาหกรรมยานยนต์อย่างมากมาย ทำให้มีรถยนต์รุ่นใหม่ออกมามากเช่นเดียวกัน สำหรับในประเทศไทย ปัจจุบันบนท้องถนนมีรถยนต์รุ่นใหม่ออกมามากรวมทั้งรุ่นเก่าด้วย และมีแต่จะเพิ่มขึ้นทุกวัน แต่ในทางตรงกันข้ามถนนที่จะรองรับรถเหล่านั้นไม่เพียงพอนั่นเป็นปัญหาหนึ่ง อีกปัญหาคือที่จอดรถไม่เพียงพอกับจำนวนรถที่เพิ่มขึ้น จึงมีอาคารสำหรับจอดรถ แต่มีปัญหาเรื่องการชำระค่าจอด ความสะดวกสบายในการจอดรถ ดังนั้นระบบจอดรถที่อยู่ภายในอาคาร เป็นลักษณะการบริการตัวเอง มีการแสดงจำนวนที่จอดรถเหลือ คิดเงินตามเวลาหลังจากรถเข้าจอด และแบ่งอัตราค่าจอดตามชั้นจอด ถ้าจอดชั้นสูงอัตราค่าจอดจะถูกกว่าอัตราค่าจอดชั้นต่ำ การชำระเงินจะชำระเงินที่ชั้นล่างก่อนจะไปนำรถออกจากที่จอด



รูปที่ 1 แสดงทางเข้าออกของอาคารจอดรถ

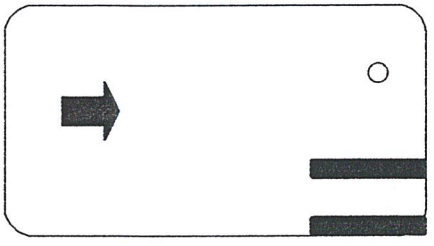
ทางเข้าออกของอาคารจอดรถแสดงดังรูปที่ 1 ทางเข้าออกประกอบด้วยเซนเซอร์(sensor) ซึ่งจะทำหน้าที่คอยเช็คว่ามีรถเข้าออกหรือยัง เมื่อมีรถเข้าเซนเซอร์ตัวที่ 1 จะส่งค่าไปบอกหน่วยประมวลผลกลางว่าขณะนี้มีรถเข้ามาแล้วประตูที่กั้นรถก็จะเปิด เมื่อรถวิ่งผ่านเซนเซอร์ตัวที่ 2 ประตูทางเข้าก็จะปิดพร้อมทั้งแสดงผลด้านทางเข้าว่าสามารถจอดรถได้อีกกี่คัน เมื่อรถออกวิ่งผ่านเซนเซอร์ตัวที่ 3 แล้ว เซนเซอร์ตัวที่ 3 จะส่งค่าไปบอกหน่วยประมวลผลกลางว่าขณะนี้มีรถออกมาจากอาคารจอดรถแล้ว หน่วยประมวลผลกลางก็จะสั่งให้จอแสดงผลลดค่าจำนวนรถที่สามารถจอดได้จำนวน 1 คัน

เมื่อขับรถผ่านทางเข้ามาแล้ว วนหาช่องจอดรถที่ว่างและมาเข้าจอด ซึ่งรายละเอียดในส่วนต่อไป จะอธิบายใน บทที่ 2 เป็นรายละเอียดเกี่ยวกับทฤษฎีการทำงานในส่วนต่างๆ บทที่ 3 จะเป็นขั้นตอนในการทำงานทั้งหมด บทที่ 4 จะกล่าวถึงผลการทดลองและสรุป

บทที่ 2
ทฤษฎีและหลักการ

2.1 ทฤษฎีการอ่านค่าบัตร

การทำงานที่สำคัญคือ ความสามารถทำการอ่านค่าบนตัวบัตร โดยภาคตรวจจะทำการอ่านรูปแบบของบัตรส่งไปให้หน่วยประมวลผลกลาง หลังจากนั้นหน่วยประมวลผลกลางจะนำรูปแบบที่ได้นี้ ไปทำการวิเคราะห์จัดเก็บเพื่อรับคำสั่งต่อไป



2.2 ภาคตรวจจับ

2.2.1 คุณสมบัติของการตรวจจับ

ตัวตรวจจับเป็นส่วนสำคัญของภาคตรวจจับ ซึ่งมีข้อกำหนดว่าต้องสามารถแยกแยะความแตกต่างระหว่างบัตร โดยอาศัยหลักการที่ว่า พื้นสีดำและพื้นสีขาวนั้นมีอัตราการดูดกลืนแสงที่ไม่เท่ากัน นั่นคือ สีดำมีอัตราการดูดกลืนแสงที่มากกว่า ทำให้แสงมาตกกระทบจะมีแสงออกมาน้อยกว่าพื้นสีขาว

2.2.2 หลักการของอุปกรณ์ตรวจจับ

อุปกรณ์ตรวจจับสามารถแบ่งตามการทำงานพื้นฐานได้ 2 ประเภท คือ ใช้หลักการสะท้อนลำแสง และหลักการของการตัดลำแสง ซึ่งการนำไปใช้งานต้องคำนึงถึงลักษณะของการทำงาน และความเหมาะสมด้วย การทำงานของอุปกรณ์ตรวจจับมีดังนี้

แบบใช้หลักการสะท้อนแสง

อุปกรณ์ตรวจจับแบบใช้หลักการสะท้อนของแสงมีข้อดีการตรวจจับสามารถกำหนดพื้นที่การทำงานได้แน่นอน ทำให้การออกแบบระบบได้ง่าย การตรวจจับจะอาศัยวัสดุที่สะท้อนแสงได้ ลำแสงอินฟาเรด(infrared)จาก LED สะท้อนกลับมายังโฟโต้ทรานซิสเตอร์(phototransistor)ทำให้ได้ระดับลอจิกที่แน่นอนและเปลี่ยนแปลงระดับลอจิกได้อย่างรวด เร็วหรือทำให้เกิด ฮิสเตอรีซีส (hyteresis) น้อยมากทำให้รูปสัญญาที่ได้ไม่เกิดการบิดเบี้ยว

แบบใช้หลักการของการตัดกันของลำแสง

แตกต่างกับแบบแรกตรงที่การจัดวางตำแหน่งของ LED อินฟาเรดและโฟโต้ทรานซิสเตอร์ ซึ่งจะวางหันหน้าเข้าหากัน โดยมีช่องว่างแคบๆ ลำแสงอินฟาเรดจาก LED จะส่งผ่านช่องว่างมากระทบกันกับ โฟโต้ทรานซิสเตอร์ ให้ทำงานตลอดเวลา แต่เมื่อใดที่วัตถุทึบแสงมาตัดผ่านช่องว่างจะทำให้แสงที่ตกกระทบกับ โฟโต้ทรานซิสเตอร์ ขาดหายไป และทำให้หยุดการทำงานทันที

การใช้ตัวตรวจจับแบบนี้ต้องใช้กับจานเข้ารหัสที่มีขนาดพอเหมาะและไม่บิดเบี้ยว เพื่อป้องกันการเสียดสีระหว่างจานเข้ารหัส และตัวตรวจจับในช่องแคบ จานเข้ารหัสที่ใช้จะเป็นวัสดุโปร่งแสง การทำงานของตัวตรวจจับแบบใช้หลักการตัดของลำแสงจะให้ความแม่นยำสูง เพราะไม่ต้องอาศัยการสะท้อนแสงแต่อย่างไรก็ตามการประยุกต์การใช้งานทำได้ด้อยกว่าแบบใช้หลักการสะท้อนของแสง เพราะติดตั้งในลักษณะเดียวคือ คร่อมระหว่างด้านหน้า ด้านหลังของจานเข้ารหัส

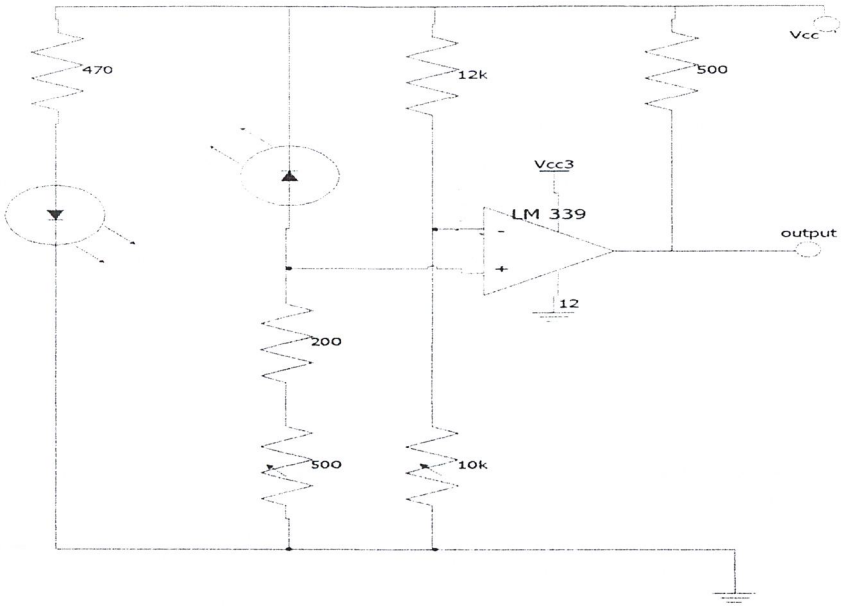
อุปกรณ์ตรวจจับที่ใช้หลักการของแสง มีข้อเสียที่ไม่สามารถทำงานในที่ที่มีความสว่างๆได้ เพราะแสงจากแหล่งอื่นมาตกกระทบ โฟโต้ทรานซิสเตอร์ จนทำให้ไม่สามารถตรวจจับแสงได้ทั้งประเภทส่งแสงโดยตรง และแบบสะท้อน เพื่อแก้ไขปัญหานี้ อุปกรณ์ตรวจจับที่ใช้หลักการของแสงจึงต้องมี ชีล(shield) เพื่อป้องกันแสงจากภายนอกด้วย

2.2.3 การเลือก เซนเซอร์

สำหรับตัวตรวจจับจะใช้หลักการสะท้อนของแสงซึ่งเหมาะที่จะใช้ แยกความแตกต่างของพื้นสีขาวและดำ ซึ่งมีความแตกต่างเรื่องของสัมประสิทธิ์ การสะท้อนแสงค่อนข้างมากและเลือกใช้การรับส่งแสงแบบอินฟาเรด เพื่อลดปัญหาการรบกวนของแสงสว่างภายในห้องลงไปได้ยิ่งไปกว่านั้น วงจรใช้งานของระบบอินฟาเรดยังเป็นวงจรที่ง่าย ไม่ซับซ้อนและมีความน่าเชื่อถืออีกด้วยแต่ในทางปฏิบัติ ดูแล้วว่าถ้าใช้ในรูปแบบของบัตรมีความเป็นไปได้ต่ำเพราะตัวอ่านมีลักษณะเป็นกล่อง เพราะแสงภายในมืดทำให้การอ่านค่านั้นผิดพลาดได้จึงต้องจำลองแสงภายในกล่องขึ้นมาเพื่อการอ่ารค่าได้อย่างถูกต้อง

ภาควงจร

เป็นวงจรแบบง่ายๆที่ประยุกต์มาจากไมโครเมาท์ ใช้ IC LM 339 ใช้หลักการเปรียบเทียบ (comparators) จ่ายไฟ 9 โวลท์ ภายในวงจรมี ตัวรับส่งอินฟาเรด วงจรละ 2 ตัว ซึ่งในแถบบัตรเราใช้เลขฐาน 2 ,3 หลักจึงใช้ทั้งหมด 6 ตัว และมีไฟจำลองแสง เพื่อความถูกต้องในการอ่านค่า ภายในวงจรยังประกอบด้วย ตัวต้านทานแบบธรรมดาและแบบปรับค่าได้ โดยเอาท์พุทออกมาประมาณ 8 โวลท์ เมื่อตรวจจับสีขาวและ สีดำประมาณ 0 โวลท์



รูปที่ 2 วงจรเซนเซอร์

2.3 สเต็ปมอเตอร์แบบยูนิโพล่าร์

สเต็ปมอเตอร์(stepmotor)ได้รับการพัฒนาอย่างต่อเนื่อง จนปัจจุบันสเต็ปมอเตอร์ที่นิยมใช้กันอย่างแพร่หลายที่สุด และหาได้ง่ายที่สุดคือสเต็ปเปอร์มอเตอร์แบบยูนิโพล่าร์(unipolar) มีลักษณะการพันขดลวดของมอเตอร์แบบพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์(stater) แต่ละขดแบ่งเป็น 2 เฟส(phase) รวมมอเตอร์ทั้งตัวจะมี 4 เฟส คือ เฟส 1,2,3,และ4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำให้สเต็ปมอเตอร์แบบนี้มีทั้ง แบบ 5 สาย 6 สาย ถ้าเป็นแบบ 5 สาย จะเป็นการนำสายไฟเลี้ยงขดลวดทั้ง 2 มาต่อรวมกันเป็นสายเดียว

การกระตุ้นและการควบคุมการหมุนของสเต็ปเปอร์มอเตอร์

การกระตุ้นและการควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีเควนเชียลในรูปแบบที่ถูกต้อง ด้วยสามารถแบ่งได้เป็น 3 รูปแบบคือ แบบ 1 เฟส 2 เฟส และแบบครึ่งสเต็ป (half step)

แบบหนึ่งเฟสหรือแบบฟูลสเต็ป (full step)เป็นการกระตุ้นที่มีรูปแบบง่ายที่สุด โดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป เช่น เริ่มต้นที่ ขด 1,2,3,4แล้ววนกลับมาที่ขดที่ 1 วนไปเรื่อยๆ หรือเริ่มที่ขดที่ 4,3,2 แล้วกลับมายังขดที่ 1 อีกครั้ง ซึ่งทำให้ทิศทางการหมุนสวนทางกัน ในการกระตุ้นรูปแบบนี้จึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบเวฟจึงมีราคาถูกและหาง่าย ขั้นตอนการทำงานต่างๆแสดงด้านล่าง

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

แบบ 2 เฟสเป็นการกระตุ้นซึ่งคล้ายกับแบบเฟส 1 แต่การกระตุ้นแบบนี้จะทำการกระตุ้นโดยจ่ายกำลังไฟฟ้าไปที่ขดลวดสองขด ที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไป เช่นเดียวกับแบบเวฟ ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ หนึ่ง และ สอง ตามด้วย การกระตุ้นขดที่ สอง สาม ต่อไปขดที่ สาม สี่ ถัดไปขดที่ หนึ่ง สอง วนไปตามลำดับเช่นนี้ หรือเริ่มที่ขดหนึ่ง สี่ ตามด้วยสาม สี่ สอง หนึ่ง ลักษณะนี้ จะสวนทางกัน การกระตุ้นเสต็ปมอเตอร์แบบนี้จะเพิ่มแรงบิดได้มากกว่าแบบเวฟ(wave)โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอีกจากสองขดลวดถัดไป สำหรับข้อเสียการกระตุ้นแบบนี้จะใช้กำลังไฟฟ้าที่มากขึ้น ขั้นตอนการทำงานแสดงในตารางด้านล่าง

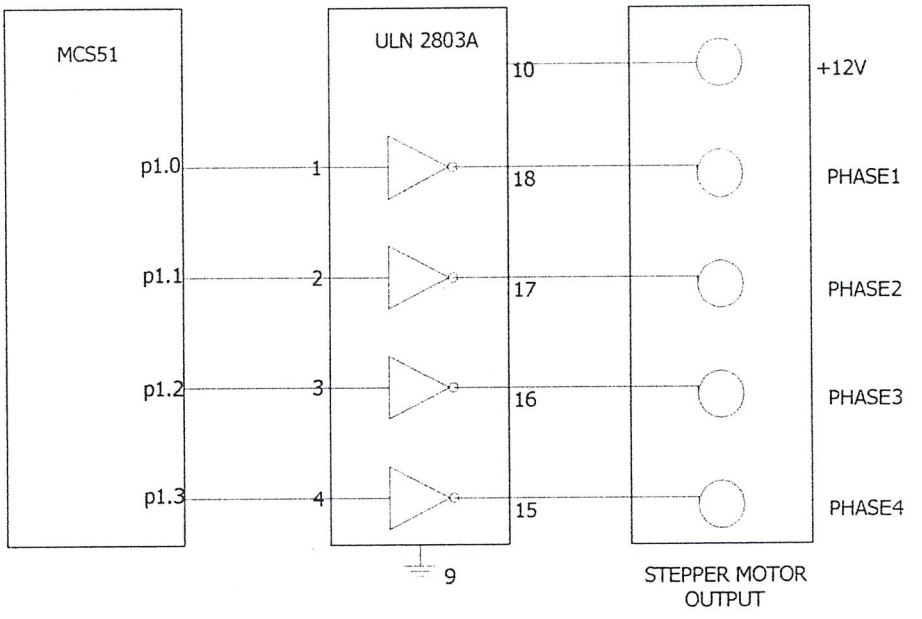
สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

แบบครึ่งสเต็ปเป็นรูปแบบที่ผสมผสานระหว่างการกระตุ้นแบบเวฟและแบบ สอง เฟส เพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ เริ่มจากขดลวดที่ 1,1 และ 2,2,2 และ 3,3,3 และ 4,4,4 และ 1 วนแล้วกลับมายังขดลวดที่ 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขด ที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องพึงระวังไว้อีกประการหนึ่งว่า เมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องทำการหมุนถึง 2 สเต็ปจึงจะได้เท่า 1 สเต็ปของการควบคุมในใบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าใช้ขนาดเท่ากับแบบ 2 เฟส เป็นอย่างน้อย จึงจะเพียงพอ ขั้นตอนการทำงานต่างๆแสดงดังตารางด้านล่าง

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-

6	-	-	ทำงาน	ทำงาน
สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

แสดงลำดับการทำงานของขดลวดในแต่ละเฟสมอเตอร์ เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป



รูปที่ 3 ตัวอย่างวงจรขับสเต็ปเปอร์มอเตอร์ของไมโครคอนโทรลเลอร์ MCS-51

ตัวอย่างวงจรขับสเต็ปเปอร์มอเตอร์ของไมโครคอนโทรลเลอร์ MCS-51 จะใช้พอร์ต 0 ส่งข้อมูลเข้าไปยัง IC ไดรฟ์เวอร์กระแสสูงแบบคอลเลคเตอร์เปิดเบอร์ ULN 2803A ด้วนการเลือก IC แบบนี้ทำให้สามารถเลือแรงดันสำหรับขับสเต็ปเปอร์มอเตอร์ได้กว้งพอสมควรตั้งแต่ 5-12 V เมื่อต้องการให้มอเตอร์หมุนให้ส่งข้อมูล 1 ไล่เรียงตามลำดับจาก P1.0 ถึง P1.3 แล้ววนกลับมาไปที่ P1.0 ใหม่ หากต้องการให้มอเตอร์หมุนกลับทิศทางก็ให้ส่งข้อมูลย้อนกลับ โดยเริ่มจาก P1.3 ก่อนแล้วสิ้นสุดรอบที่ P1.0 แล้ววนกลับไปที่ P1.3 ใหม่ เมื่อ ULN 2803A ได้รับข้อมูล 1 ก็จะทำการกลับลอจิก ทำให้เกิดกระแสไฟฟ้าไหลผ่านขดลวดของมอเตอร์ที่ต่อเข้าอยู่กับเอาท์พุทที่ทำงานส่งผลทำให้เกิดการเคลื่อนที่ของแกนมอเตอร์ขึ้น

2.4 ทฤษฎีการขับโมดูลแสดงผลแบบผลึกเหลว (LCD)

ตัวแสดงผล (display) ภายในเป็นผลึกเหลวสามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอกดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของ

โมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนที่เคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะที่นิยม คือ เบอร์ HD44780 จะใช้ควบคุม LCD แบบอักขระ

ตัวขับ (driver) เป็นตัวรับสัญญานจากตัวควบคุมมาขับให้ตัวแสดงผลข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่นี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

2.5 โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งานโมดูล (modul) LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน ในที่นี้ขอยกตัวอย่างโมดูล LCD แบบอักขระเพราะสามารถเข้าใจได้ง่ายจะเป็นบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780 ซึ่งใช้ในโมดูล LCD แบบอักขระประกอบด้วย

บัฟเฟอร์อินเอาต์พุท เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจีสเตอร์คำสั่ง เป็นรีจีสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมแสดงผล

รีจีสเตอร์ข้อมูล เป็นรีจีสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตังอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจีสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง(Look up-lable)ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงผลที่ตัวแสดงผล

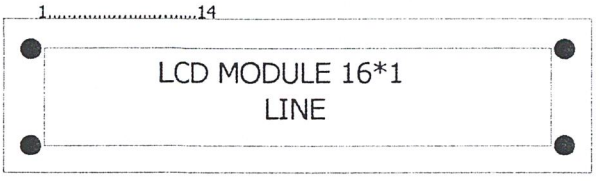
รอมเก็บตัวอักษร เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่อ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร เป็นความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอมีขนาด 512 บิต การเขียนการอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด (LCD*1) สำหรับโมดูล LCD ที่ยกมาใช้ในการเรียนรู้ในการทดลอง เป็นขนาด 16 ตัวอักษร 1 บรรทัด เนื่องจากราคาถูกหาง่าย และเป็นโมดูล LCD ที่มีโครงสร้างเป็นมาตรฐานมีผู้ผลิตหลายราย และมีการระบุเบอร์แตกต่างกันออกไปตามผู้

ผลิตอาทิ LM020L ของฮิตาชิ ของครอปเท็กซ์แต่อย่างไรก็ตามคอนโทรลเลอร์ที่ใช้คือ HD44750 ของฮิตาชิโมดูล LCD ขนาด 16*1 มีขาต่อใช้งานทั้งสิ้น14 ขามีรายละเอียดการทำงานของแต่ละขา



รูปที่ 4 จอแสดงผล LCD

Vss 1/ต่อกราวด์

Vdd 2/ต่อไฟเลี้ยง+5V

Vo 3/เป็นขาอินพุทรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS 4/เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจีสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจีสเตอร์ DR โดยถ้าขานี้เป็น 0 ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น 1 ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/W 5/เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น 0 เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น 1 จะเป็นการอ่านข้อมูล

E 6/ เป็นขาอีนาเบิล LCD ให้ทำงาน

2.6 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น 0 แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญ มี 10 คำสั่ง

2.6.1 คำสั่งเคลียร์ตัวแสดงผล

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ SPACE เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซีคิวคำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D ให้เป็น 1

2.6.2 คำสั่ง return home

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น 1 เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลงนั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

2.6.3 คำสั่งเลือกโหมดการป้อนข้อมูล(Entry mode Set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูลถ้าหากบิต S เป็น 1เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต้ถ้าหากบิตนี้เป็น 0 เมื่อเกิดข้อมูลใหม่ตัวเคอร์จะเลื่อยไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น 1 แอดเดรสของ DDRAM จะเพิ่มขึ้นแต่ถ้าเป็น 0 แอดเดรสจะลดลง

ดังนั้นข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H และที่ใช้บ่อยคือ 06H หมายถึงกำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือและแอดเดรสของ DDRAM เพิ่มขึ้น

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอ ถ้าบิตนี้เป็น 1 จะเป็นการเปิดจอแสดงผล ถ้าเป็น 0 จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น 1 ถ้ากำหนดให้เป็น 0 จะเป็นการปิดเคอร์เซอร์หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น 1 เคอร์เซอร์นี้จะกระพริบ

ดังนั้นจะมีข้อมูลคำสั่งตั้งแต่ 08H-0FH ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผลแต่ไม่แสดงเคอร์เซอร์ และ 0FH *เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ*

2.6.4 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1C-1FH

2.6.5 กำหนดฟังก์ชันการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	1	DL	N	F	*	*

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น 0 จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น 1 จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผลถ้าเป็น 0 จะแสดงผล 1 บรรทัดถ้าเป็น 1 จะแสดง 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงมากกว่า 2 บรรทัด ก็กำหนดบิต N ให้เป็น 1

บิต F ใช้เลือกความละเอียดของตัวอักษรให้แสดงผล ถ้าบิตนี้เป็น 0 จะเป็นการแสดงผลแบบ 5*7 จุด และถ้าเป็น 1 จะแสดงผลเป็นแบบ 5*10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5*7จุด

จุดที่น่าสังเกตคือ โมดูล LCD แบบ 16 ตัวอักษร 1 บรรทัด แม้จะมีบรรทัดการแสดงผลเพียง 1 บรรทัด แต่จะต้องกำหนด N ให้เป็น 1 เนื่องจากแอดเดรสของ DDRAM แบ่งเป็น 2 ช่องคือ 00H และ 40 H

2.6.6 คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดสรของ CGRAM ต้องกำหนดให้บิต 7 เป็น 0 บิต 6 เป็น 1 ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00-3H

2.6.7 คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น 1 และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น 0 แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CH และถ้าบิต N เป็น 1 แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH-87H และ 0C0H-0C7H

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้จะเป็นบอกสถานะของตัวควบคุมLCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น 0 แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือ

คำสั่ง แต่ถ้าเป็น 1 แสดงว่า ขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น 1 ด้วย แต่สัญญาณที่ RS ยังต้องเป็น 0 อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต0-บิต6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.6.8 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมดการทำงานให้แก่โมดูลของLCDก่อนจากนั้นจึงค่อยส่งข้อมูล(data) ที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้นคือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก 0 หมายความว่าข้อมูลที่ป้อนให้กับโมดูล LCD ขณะนี้เป็นคำสั่งในทางตรงข้ามหากขา RS ได้รับลอจิก 1 ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการอ่านหรือเขียนข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น 1 เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านหรือเขียนข้อมูลต้องกำหนดให้ขา R/W เป็น 1 ข้อมูลขนาด 8 บิต หรือ 4 บิต ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

2.6.9 จังหวะการทำงานของ LCD โมดูล

ในการติดต่อกับโมดูล LCD จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูลเนื่องจากต้องรอให้คอนโทรลเลอร์ภายใน LCD โมดูล แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยเสียก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

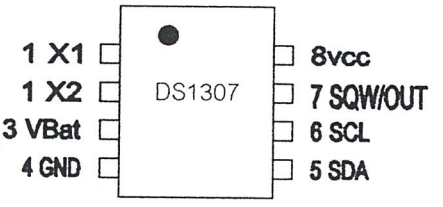
ดังนั้นในการใช้งานโมดูล LCD ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้โมดูล LCD พร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่โมดูล LCD ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้โมดูล LCD ทำการเตรียมความพร้อมหรืออินนิเชียล หลังจากนั้นก็กำหนดลอจิกให้แก่ขา RS ของโมดูล LCD แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาที เพื่อให้คอนโทรลเลอร์ใน LCD โมดูลแปลความหมายของขาลอจิกที่ขา RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิลโมดูล LCD ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของโมดูล LCD ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

ทั้งหมดที่กล่าวมาคือขั้นตอนการและจังหวะการทำงาน 1 รอบของโมดูล LCD จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรม อินนิเซียล LCD โปรแกรมหน่วงเวลา และโปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิลโมดู LCD

2.7 ทฤษฎี DS1307 ไอซีสร้างฐานเวลาหรือเรียลไทม์คล็อก(real time clock)

ผู้ผลิตคือ เดลลัสเซมิคอนดักเตอร์ ที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์ โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที นา ชั่งโมง วันที่ วันในสัปดาห์ เดือน ปี โดยสามารถปรับเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง คุณสมบัติทางเทคนิคมีดังนี้

- เป็นไอซีเรียลไทม์คล๊อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี
- มีหน่วยความจำนอนโวลาไทแรม 56 ไบต์อยู่ภายในสามารถเก็บข้อมูลทั่วไป
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ สามารรักษาข้อมูลไว้ได้แม้ไม่มีไฟเลี้ยง



รายละเอียดขาต่อใช้งานของ DS1307

Vcc.GNDขา 8,4 ต่อไฟเลี้ยง 5 V

V_{BAT} ขา 3 ใช้ต่อกับแบตเตอรี่ 3 V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไป แม้ว่าจะไม่มีไฟเลี้ยงให้แก่ DS 1307 ชนิดของแบตเตอรี่ที่เหมาะสมคือแบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40 mAhr หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปีที่อุณหภูมิ 25 องศาเซลเซียส

SDA , SCL ขา 5,6 เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัส I^2C

SQW/OUT ขา 7 ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1Hz,4.096kHz,8.192kHz,และ32kHz ในการใช้งานต้องต่อตัวต้านทาน 1k พูลอัปที่ขานี้ด้วย

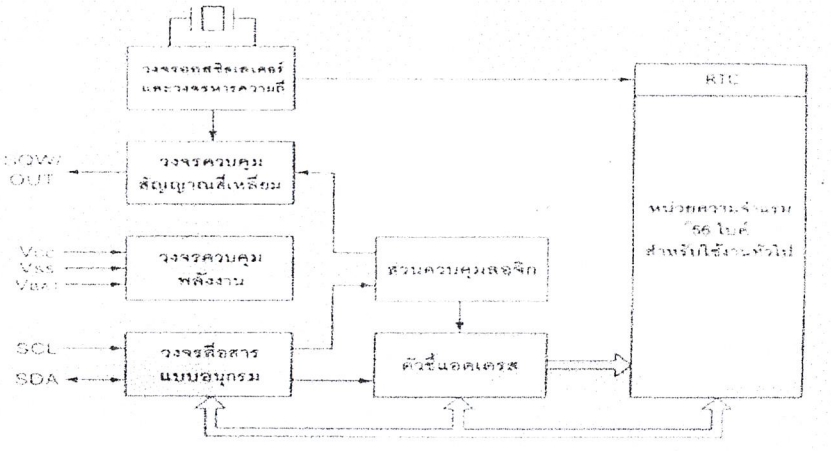
X1,X2 ขา1,2 ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768kHz เพื่อใช้เป็นฐานเวลาเพื่อใช้ในการสร้างเวลาจริงในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่ำๆประมาณ 15 pF คร่อมกับขากราวด์ด้วย

2.7.1 การทำงานของ DS1307

ไอซี DS 1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I^2C ในรูปแสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS 1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS 1307 ทำงานที่ขา

SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลากรณีที่มีการอีนาเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจีสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่วยความจำนอนโวลาไทแรมซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปอีก 56 ไบต์

วงจรควบคุมพลังงานไฟฟ้า จะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า 1.25 x V_{BAT} ก็จะควบคุมให้ DS 1307 หยุดการทำงานรีเซตค่าตัวนับแอดเดรสภายในทำให้ไม่สามารถติดต่อกับ DS 1307 ได้ดังนั้นในการใช้งาน DS 1307 ต้องระวังอย่าให้ไฟเลี้ยงต่ำกว่า 1.25 x V_{BAT} หรือประมาณ 3.75 v ในกรณีที่ใช้ V_{BAT} เท่ากับ 3 v ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} ไอซี DS 1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS 1307 ก็สามารถให้ค่าของเวลาที่เป็นจริงได้

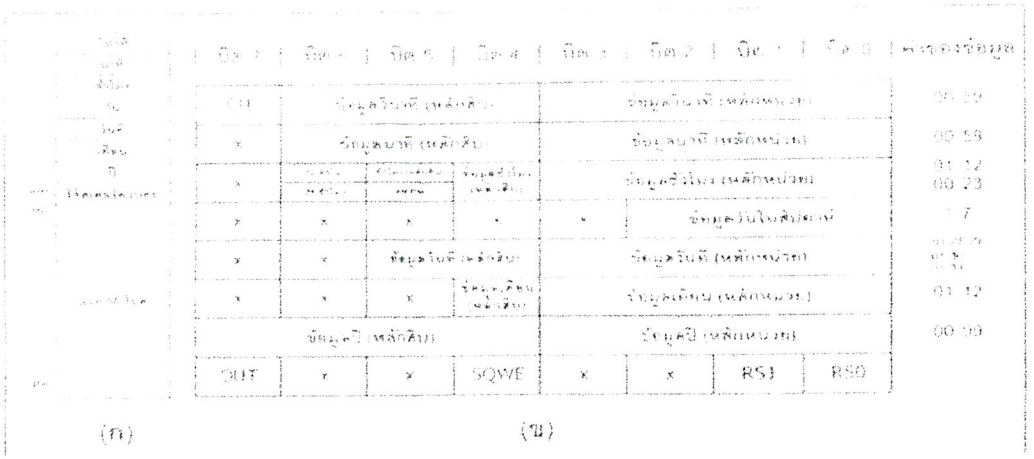


รูปที่ 5 วงจรภายใน DS1307

วงจรสื่อสารอนุกรมภายใน DS 1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS 1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส I^2C

2.7.2 การจัดสรรหน่วยความจำใน DS 1307

ในรูป ก แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์แรกตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจีสเตอร์ค่าเวลาใช้ในเก็บข้อมูลเกี่ยวกับเวลา ไบต์ต่อมาที่แอดเดรส 07 H เป็นพื้นของรีจีสเตอร์ควบคุมการทำงานของ DS1307 ในรูป ข แสดงรายละเอียดของรีจีเตอร์ค่าเวลาและรีจีสเตอร์ควบคุมของ DS1307



รูปที่ 6 แสดงการจัดสรรหน่วยความจำ

ด้วยการจัดสรรพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามที่ต้องโดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐาน 10 สำหรับการแสดงเวลาในรูปของชั่วโมงสามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24โดยกำหนดที่ บิต6 ของแอดเดรส 02 H และเมื่อเลือกแบบ 12 ชั่วโมงที่บิต 5 ในแอดเดรสเดียวกันจะใช้ในการแสดงค่า AM/PM โดยถ้าบิตนี้เป็น 1 หมายถึง ค่าชั่วโมงในขณะนี้เป็นช่วงเวลาหลังเที่ยงวัน ในกรณีที่เป็นแบบ 24 ชั่วโมงบิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง

2.7.3 รีจีสเตอร์ควบคุม

มีแอดเดรสอยู่ที่ 07 H มีรายละเอียดดังนี้

OUT (Outputcontrol) *ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่ดิสเอเบิลการกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น 1 ที่ขา SQW/OUT ก็จะเป็น 1 ถ้าบิตนี้เป็น 0 ก็จะเป็น 0 ด้วยเช่นเดียวกัน*

SQWE (Square Wave Enable) *ใช้ในการอีนาเบิลวงจรกำเนิดสัญญาณสี่เหลี่ยมที่ขา SQW/OUT ถ้าต้องการให้มีสัญญาณสี่เหลี่ยมออกมาต้องกำหนดให้บิตนี้เป็น 1*

RS1,RS0 *ใช้ในการเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SQW/OUT ดังมีรายละเอียดดังนี้*

RS1	RS0	ค่าความถี่สัญญาณ
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

2.7.4 โหมดการทำงานของ DS1307

มีด้วยกัน 2 โหมดคือโหมดเขียนข้อมูลและโหมดอ่านข้อมูล ในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

2.7.5 โหมดการเขียนข้อมูล

เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ทำการกำหนด สภาวะเริ่มต้นจากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียนจากนั้นรอการตอบรับจาก DS1307เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรสหลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องรอการตอบรับจาก DS1307 ทกครั้งจึงสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสภาวะหยุด เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล

2.7.6 โหมดการอ่านข้อมูล

เริ่มต้นทำงานเหมือนกับโหมดเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่านซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยแล้ว DS1307 จะทยอยส่งข้อมูลออกมาให้คอนโทรลเลอร์คราวละ 1 แอดเดรส หรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆคือเข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูล ให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจาแอดเดรสที่กำหนดไว้ก่อนหน้านี้

2.7.8 การเชื่อมต่อ DS1307 กับไมโครคอนโทรลเลอร์ MCS-51

แสดงดังรูป จะเห็นได้ว่ามีลักษณะการต่อเหมือนกับอุปกรณ์บนระบบบัส I^2C ตัวอื่นๆทุกประการ และสามารถที่จะต่อไอซีทั้งหมดร่วมกันบนสาย SDA และ SCL ได้ เป็นการย้ำให้เห็นถึงความสามารถพิเศษของระบบบัส I^2C ที่ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ที่มีความต่างกันในหน้าที่การทำงานบนสายสัญญาณเดียวกันได้ ถึงการทดลองนี้ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ระบบบัสได้คือ I^2C ได้ถึง 3 ตัว 3 ลักษณะการทำงาน โดยใช้สายสัญญาณเพียง 2 เส้น

ทฤษฎีการใช้งานไมโครคอนโทรลเลอร์ MCS-51

2.8 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ที่มีขนาด 8 บิตประกอบด้วยไมโครคอนโทรล-เลอร์เบอร์ต่างๆ ทุกๆ เบอร์จะมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดหน่วยความจำ-ภายใน และ ภายนอกที่แตกต่างกัน เพื่อความเหมาะสมในการนำเอาไปใช้งานตามความต้องการต่างๆ แต่เดิม 8051 ถูกสร้างด้วยวิธี HMOS I แต่ในปัจจุบันได้สร้างด้วยวิธี HMOS II จึงมีชื่อเป็น 8051AH ไมโครคอนโทรลเลอร์ในตระกูล 51 นั้น ถึงแม้ว่าจะมีหลายเบอร์แต่เราก็จะเรียกว่าเป็น "8051" ซึ่งหมายถึงไมโครคอนโทรลเลอร์ตระกูล51 นั้น ส่วนเบอร์ 8032 และ 8052 มีหน่วยความจำภายในเพิ่มขึ้นและมีวงจรนับ/จับเวลา ขนาด 16 บิตเพิ่มขึ้นมาดังตารางที่ 1

ตารางที่ 1 ไมโครคอนโทรลเลอร์ตระกูล 8051

เบอร์	หน่วยความจำภายใน		จำนวนไทเมอร์/เคาน์เตอร์	อินเตอร์รัพต์หมายเลข
	เก็บโปรแกรม	เก็บข้อมูล		
8052H	8K x 8 ROM	256 x 8 ROM	3 x 16-Bit	6
8051H	4K x 8 ROM	128 x 8 ROM	2 x 16 Bit	5
8051	4K x 8 ROM	256 x 8 ROM	2 x 16 Bit	5
8032AH	ไม่มี	128 x 8 ROM	3 x 16-Bit	6
8031AH	ไม่มี	128 x 8 ROM	2 x 16 Bit	5
8031	ไม่มี	128 x 8 ROM	2 x 16 Bit	5
8751H	4K x 8 EPROM	128 x 8 ROM	2 x 16 Bit	5
80751H-12	4K x 8 EPROM	128 x 8 ROM	2 x 16 Bit	5

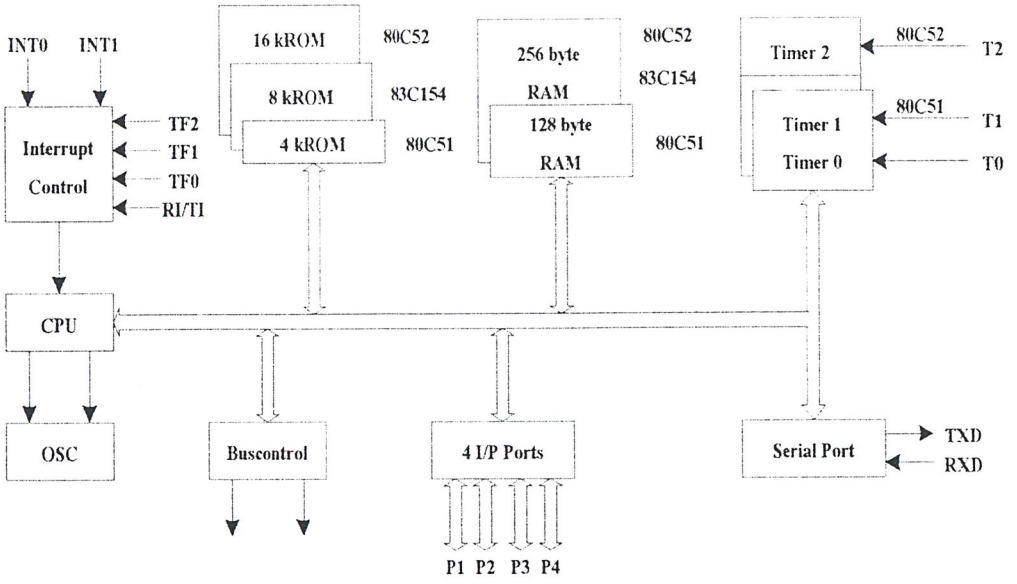
2.8.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

1. ต้องการแหล่งจ่ายไฟ + 5 V. ชุดเดียว
2. มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 กิโลไบต์
3. มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 ขึ้นไปมีถึง 256 ไบต์
4. มีหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
5. มีไทเมอร์เคาน์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับเบอร์ 8052 มี 3 ชุด) ทำงานได้ 4โหมด
6. รับอินเตอร์รัพท์ได้ 6 แหล่ง 5 เวคเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6 เวคเตอร์
7. มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ตแบบ Full Duplex เลือกรูปได้ 4 โหมด
8. มีคำสั่งในการทำ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิตและ 1 บิต
9. มีวงจรออสซิลเลเตอร์ภายใน

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.8.2 โครงสร้างของไมโครคอนโทรเลอร์ตระกูล MCS-51

ภายใน 8051 จะประกอบด้วยเกจชนิดต่างๆ เช่น AND, OR, NOT ซึ่งแต่ละเกจเหล่านี้จะนำมาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง, วงจรสัญญาณนาฬิกา เป็นต้น โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆ ดังรูป 7



รูปที่ 7 โครงสร้างของ 8051

2.8.3 โครงสร้างของ 8051 จะประกอบด้วย 3 ส่วน หลัก ๆ ดังนี้

1. **ซีพียู** (Central Processing Unit) ส่วนนี้จะทำหน้าที่สร้างสัญญาณควบคุมการติดต่อกับส่วนอื่นๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุม ได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ,อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออก ซึ่งส่วนควบคุมการขัดจังหวะ และส่วนควบคุมบัสก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณวงจรควบคุมจากซีพียูนี้ จะทำการสร้างสัญญาณ โดยการถอดรหัสจากคำสั่งที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกา ที่สร้างขึ้นจากวงจรออสซิลเลเตอร์เพื่อให้ทุกๆ ส่วนทำงานประสานกันอย่างถูกต้องในซีพียูยังประกอบด้วยส่วนประมวลผล (Arithmetic Logic Unit) ทำหน้าที่ประมวลผลข้อมูล เช่น การบวก, ลบ, คูณ หรือหาร ข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในหน่วยความจำที่ต้องการ

2. **หน่วยความจำ** (Memory) มีไว้สำหรับจดจำข้อมูล ซึ่งในการนำข้อมูลเข้าและออกจากหน่วยความจำ เราจำเป็นต้องรู้ตำแหน่งของหน่วยความจำ (Address) ในการนำข้อมูลเข้าไปเก็บในหน่วยความจำเรียกว่า การเขียนข้อมูล และการนำข้อมูลออกจากหน่วยความจำ เรียกว่าการอ่านข้อ-มูล ในไมโครคอนโทรลเลอร์ 8051 ข้อมูล ในแต่ละตำแหน่งจะมีขนาด 8 บิต ดังนั้น

45720

แต่ละตำแหน่งของหน่วยความจำสามารถเก็บความจำข้อมูลมีค่าได้ระหว่าง 00000000_2 ถึง 11111111_2 หรือ 00H ถึง 0FFH ในการติดต่อหน่วยความจำจะต้องมีสัญญาณสามกลุ่มคือ

2.1 ตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ซึ่งในไมโครคอนโทรลเลอร์8051 จะมีหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลที่มีขนาดสูงสุดชนิดละ 65536 ตำแหน่ง (64 กิโลไบต์) ดังนั้นการอ้างตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น (2^{16} เท่ากับ 65536)

2.2 ข้อมูลที่อ่านหรือเขียนกับหน่วยความจำในตำแหน่งที่เราต้องการ

2.3 สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำเพื่อจะบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล โดยวงจรถอดรหัสคำสั่ง จะทำการสร้างสัญญาณควบคุมจากคำสั่งที่อ่านเข้ามาจากหน่วยความจำโปรแกรม

3. อุปกรณ์อินพุต/เอาท์พุต (Input / Output Device) เป็นส่วนใช้ส่งข้อมูลเข้า หรือนำข้อมูลออกจากไมโครคอนโทรลเลอร์ 8051 ทำให้สามารถติดต่อกับอุปกรณ์ภายในอุปกรณ์อินพุต/เอาท์พุต ได้แก่ อินพุต/เอาท์พุตพอร์ตแบบขนาน วงจรนับ/จับเวลา 0 วงจรนับ/จับเวลา 1 พอร์ตสื่อสารอนุกรม

3.1 พอร์ตแบบขนาน เป็นที่สำหรับใช้รับส่งข้อมูลซึ่งเป็นสัญญาณดิจิตอลเข้าหรือออกจากตัวไมโครคอนโทรลเลอร์เบอร์ 8051 มีทั้งหมด 4 พอร์ต โดยแต่ละพอร์ตจะรับส่งข้อมูลได้ 8 บิต มีพอร์ต P0, P1, P2 และ P3 บางพอร์ตใช้งานได้มากกว่า 1 อย่าง

3.2 วงจรนับเวลา/จับเวลา 0 และวงจรนับ/จับเวลา 1 เป็น วงจรที่สามารถทำการนับจำนวนไซเคิลของสัญญาณที่ต่อ จากภายนอกของไมโครคอนโทรลเลอร์เบอร์ 8051หรือจำนวนของสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์เบอร์ 8051 ก็ได้ สามารถตั้งค่าเริ่มต้นของการนับและอ่านค่าการนับได้โดยซีพียู

3.3 พอร์ตอนุกรม ซีพียูจะอ่านและเขียนข้อมูล พอร์ตอนุกรมเป็นแบบ 8บิต แต่ละข้อมูลจะถูกส่งออกมาจากไมโครคอนโทรลเลอร์เบอร์ 8051 เรียงไปทีละบิตออกจากขา TXD และ ในการรับข้อมูลก็จะรับเข้ามาทีละบิตทางขา RXD และ จัดเรียงใหม่เป็น 8 บิต เพื่อให้ซีพียูอ่านไปใช้งานต่อไปในไมโครคอนโทรลเลอร์เบอร์ 8051 มีพอร์ตใช้งานได้หลายแบบทำให้สะดวกแก่การนำไปใช้งานต่างๆ ได้มากมาย การนำพอร์ตไปใช้งานจะต้องเขียนโปรแกรมขึ้นมาควบคุม

2.8.4 สถาปัตยกรรมภายในของ 8051

MCS-51 ใช้เทคโนโลยีในการผลิตแบบ NMOS และ CMOS เบอร์ 8032 และ 8052 จะมี ROM BASIC อยู่ภายในจึงสะดวกสำหรับโปรแกรมเมอร์ที่จะเขียนโปรแกรมด้วยภาษาเบสิค โครงสร้างภายในสำหรับเบอร์ 8051 ดังแสดงดังรูปที่ 2 ซึ่งจะอธิบายถึงส่วนประกอบย่อยๆ ภาย

ในตัวของไมโครคอนโทรลเลอร์เบอร์ 8051 และ สัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา ของไมโครคอนโทรลเลอร์ 8051

ไมโครคอนโทรลเลอร์เบอร์ 8051บรรจุอยู่ภายในวงจรรวมแบบDIP (Dual Inline Package) แบบ 40 ขาดังนี้

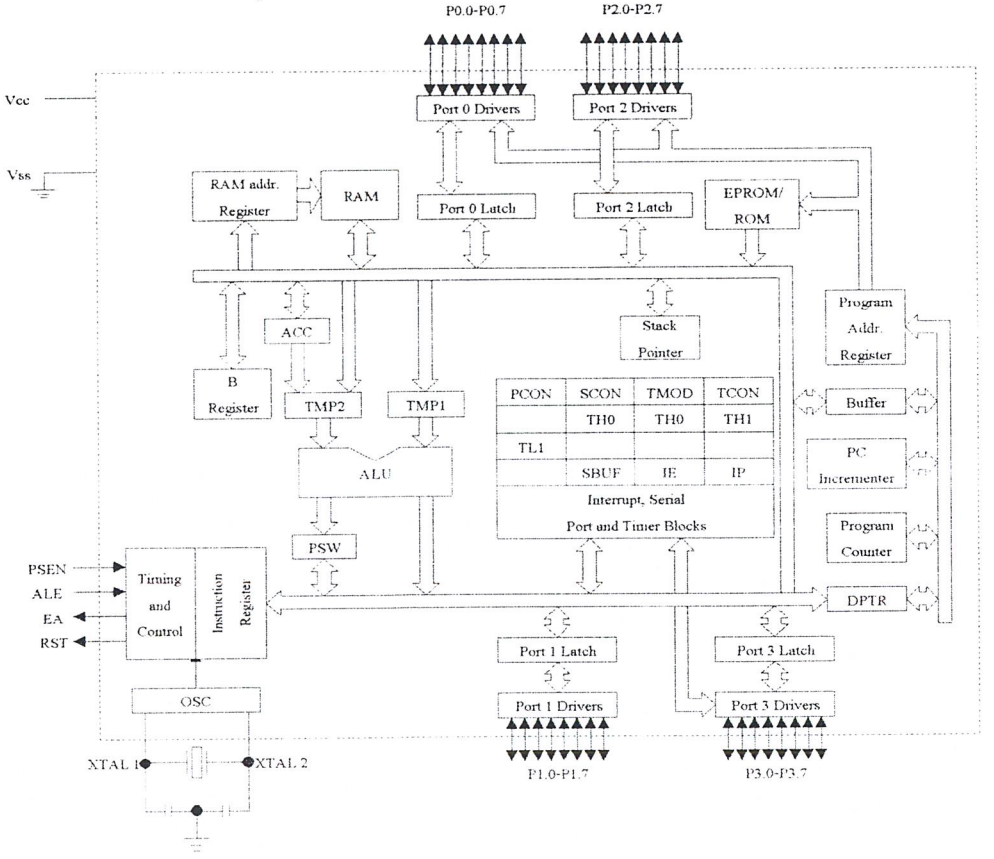
Vcc (ขา 40) ต่อกับไฟ +5V.

Vss (ขา 20) เป็นขา GND

Port 0 (ขา 32-39) มี 8 บิต คือ P0.0 – P0.7 มีโครงสร้างแบบ Open-Drain Bi-Directionalโดยสามารถใช้งานได้ 2 หน้าที่คือAddress Bus และ Data Bus นอกจากนี้ยังใช้งานเป็นอินพุตเอาท์พุตได้

Port 1 (ขา 1-8) มี 8 บิต คือ P1.0 – P1.8 ใช้งานเป็นอินพุตและเอาท์พุตพอร์ตทั่วไป

Port 2 (ขา 21-28) มี 8 บิต คือ P2.0 – P2.7 ใช้งานเป็นอินพุต/เอาท์พุตพอร์ตทั่วไปแล้วยังใช้ เป็นตัวส่ง Address ไบต์สูง เพื่อติดต่อกับหน่วยความจำภายนอก



รูปที่ 8 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ เบอร์ 8051

Port 3 (ขา 10-17) มี 8 บิต คือ P3.0 – P3.7 ใช้งานเป็นอินพุต/เอาท์พอร์ตทั่วไป และใช้งานในหน้าที่พิเศษดังนี้

3.0/RXD (Serial Input Port) ใช้รับข้อมูลแบบอนุกรม

3.1/TXD (Serial Output Port) ใช้ส่งข้อมูลแบบอนุกรม

3.2/INT0 (External Interrupt 0) ใช้รับสัญญาณขัดจังหวะจากภายนอกเบอร์ 0

3.3/INT1 (External Interrupt 1) ใช้รับสัญญาณขัดจังหวะจากภายนอกเบอร์1

3.4/T0 (Counter 0 External Input) ใช้เป็นอินพุตให้วงจรนับ / จับเวลา ชุดที่ 0

3.5/T1 (Counter 1 External Input) ใช้เป็นอินพุตให้วงจรนับ / จับเวลา ชุดที่ 1

3.6/WR (External Data Memory Write Strobe) ขาควบคุมการเขียนข้อ-มูลจากหน่วยความจำภายนอก

3.7/RD (External Data Memory Read Strobe) ขาควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอก

RST (ขา 9) Reset ใช้สำหรับรีเซ็ตวงจรภายในชิพ เพื่อเริ่มต้นการทำงานใหม่ในการรีเซ็ตต้องป้อนลอจิก “1” นานอย่างน้อย 2 แมชชีนไซเคิล

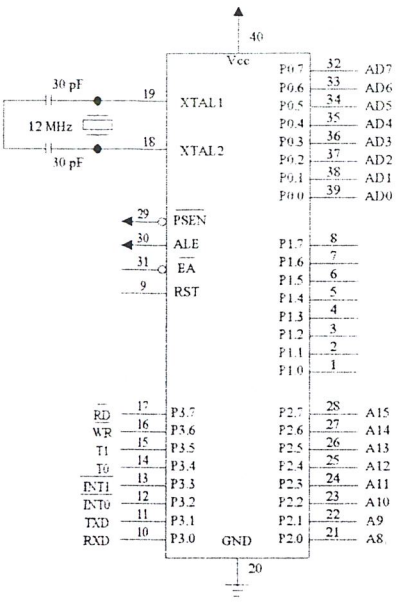
ALE (ขา 30) Address Latch Enable เป็นขาส่งสัญญาณออกไปภายนอก เพื่อควบคุมการ Latch ค่า Address ไบต์ต่ำจากพอร์ต 0

PSEN (ขา 29) Program Strobe Enable เป็นสัญญาณเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อขานี้ Active มีลอจิกเป็น “0” จะอ่านโปรแกรมจาก หน่วยความจำภายนอก และ ถ้าเป็นการอ่านโปรแกรมภายในขานี้จะไม่มีการ Active

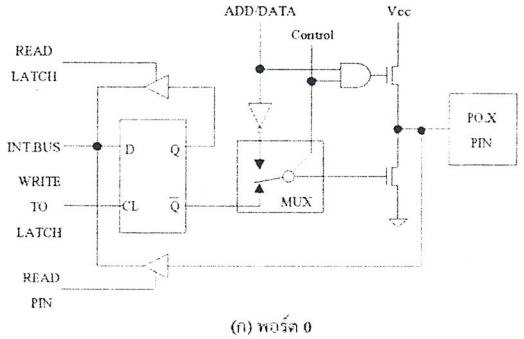
EA (ขา 31) External Access เป็น ขาที่ใช้สำหรับเลือกว่าให้ทำงานจากหน่วยความจำโปรแกรมภายในหรือหน่วยความจำภายนอกชิพ เมื่อขานี้ Active จะมีลอจิกเป็น “0” จะ เป็นการทำงานตามคำสั่งในหน่วยความจำโปรแกรมภายนอก

XTAL1 (ขา 19) ใช้ต่อคริสตัลภายนอก โดยอินพุตเข้าสู่วงจรออสซิลเลเตอร์

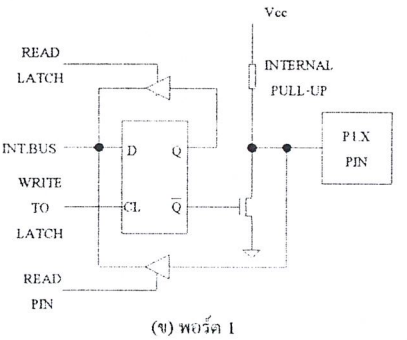
XTAL2 (ขา 18) ใช้ต่อคริสตัลภายนอก โดยเอาท์พุตออกจากวงจรออสซิลเลเตอร์



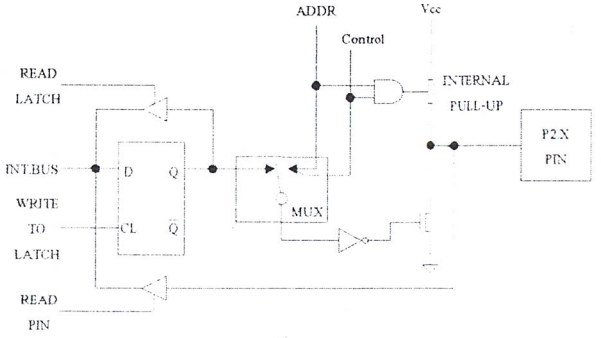
รูปที่ 9,10 ขาต่างๆ ของ 8051, โครงสร้างของพอร์ตต่างๆ ของ 8051



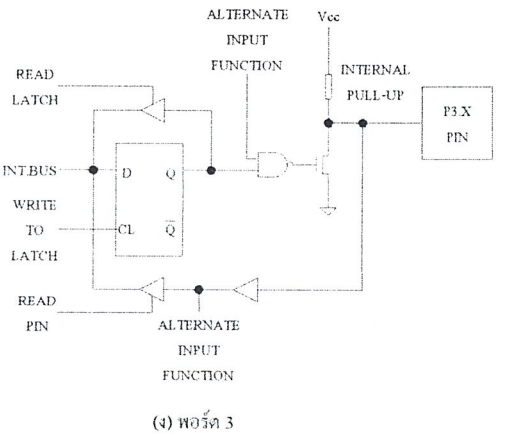
(ก) พอร์ต 0



(ข) พอร์ต 1



(ค) พอร์ต 2

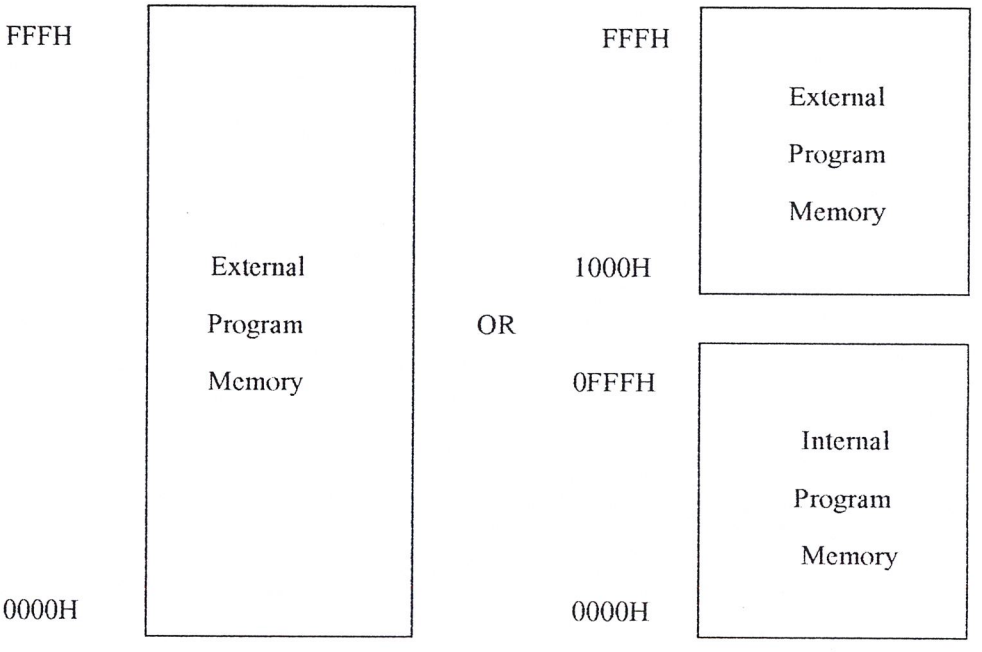


(ง) พอร์ต 3

2.8.5 การจัดการหน่วยความจำของ 8051

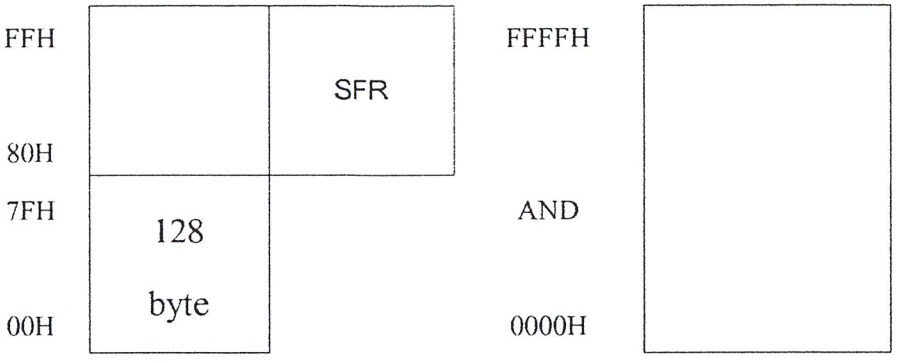
หน่วยความจำของ 8051 แบ่งออกเป็น 2 แบบตามลักษณะการใช้งาน ดังนี้

1. หน่วยความจำโปรแกรม (Program Memory) เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปของภาษาเครื่องซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำโปรแกรมไปทำการถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำนี้เป็นแบบรอม และ ผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ ส่วนที่เป็นหน่วยความจำโปรแกรมก็ คือ รอมขนาด 4 กิโลไบต์ ดังรูป



รูปที่ 11 การจัดพื้นที่หน่วยความจำสำหรับโปรแกรมของ 8051

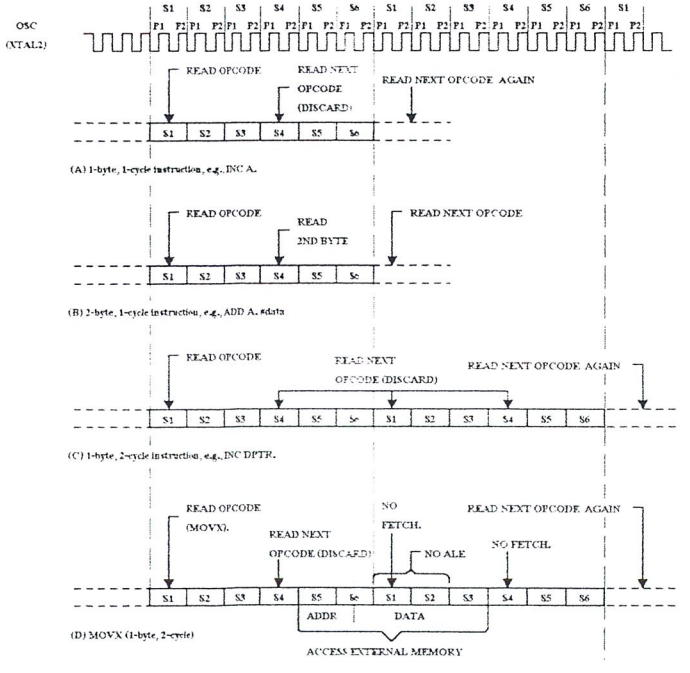
2. **หน่วยความจำข้อมูล** (Data Memory) เป็นหน่วยความจำที่ใช้เก็บข้อมูล สามารถอ่านและเขียนข้อมูลได้ ซึ่งหน่วยความจำภายในมีขนาดเพียง 128 ไบต์ ส่วนหน่วยความจำภายนอกไอซีมีขนาด 64 กิโลไบต์ ดังรูป



รูปที่ 12. การจัดพื้นที่หน่วยความจำข้อมูล

2.8.6 ฐานเวลาในการทำงานของไมโครคอนโทรลเลอร์

แมชชีนไซเคิล (Machine Cycle) คือรอบการทำงานของคำสั่ง เป็นค่าที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่ง ถ้าเป็นคำสั่งที่ซับซ้อนมากก็ต้องใช้เวลานาน 2-3 แมชชีนไซเคิล1 แมชชีนไซเคิล จะประกอบด้วยสัญญาณนาฬิกาจำนวน 12 ลูก โดยสัญญาณนาฬิกาแต่ละลูกเรียกว่า "เฟส" (Phase) สัญญาณนาฬิกา 2 เฟส รวมกันเป็น 1 สเตท (State) เพราะฉะนั้นใน 1 แมชชีนไซเคิลจึงมี 6 สเตท



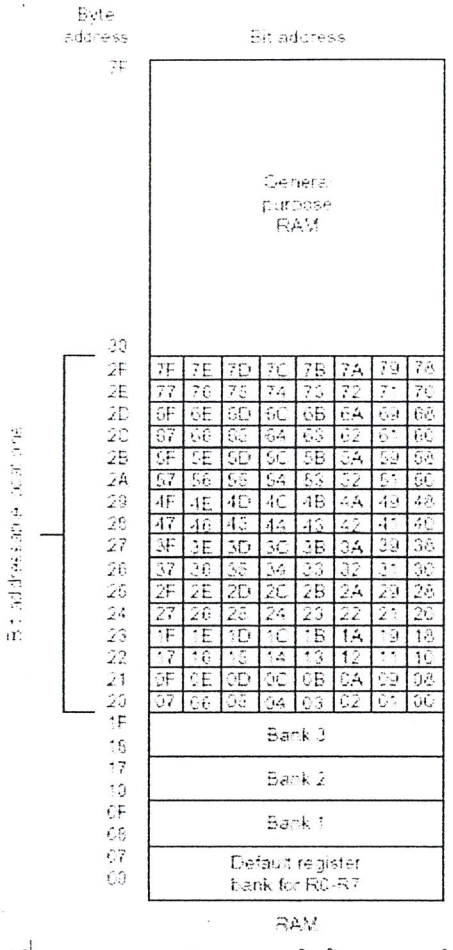
รูปที่ 13 ผังเวลาของไมโครคอนโทรลเลอร์เบอร์ 8051

2.8.7 การทำงานของ 8051

เมื่อป้อนไฟเลี้ยงให้กับไมโครคอนโทรลเลอร์เบอร์ 8051 ซึ่งมีวงจรรีเซ็ตเมื่อปิดเครื่องจะเกิดการรีเซ็ตการทำงานภายในไมโครคอนโทรลเลอร์เบอร์ 8051 เริ่มจากภาคโปรแกรมเคาน์เตอร์ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนเส้นทางหมายเลข 1 เส้นทางนี้ มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้ถูกส่งไปเก็บไว้ที่ Program ADDR Register ค่าตำแหน่งหน่วยความจำจะปรากฏลงบนบัส 16 บิต หมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากการรีเซ็ต ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมการเลือกได้ว่าเป็นรอมภายในหรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา EA ซึ่งต่ออยู่กับส่วนของวงจรเวลาและควบคุม ถ้าป้อนสัญญาณลอจิก 0 เข้าที่ขา EA เป็นการเลือกใช้รอมภายใน 8051 โดยที่วงจรเวลา และ ควบคุมจะสร้างสัญญาณไปยังรอมภายใน ให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมายังเส้นทางหมายเลข 2 ข้อมูลจากรอมถูกส่งไปยังเส้นทางหมายเลข 3 ที่เรียกว่าเส้นทางข้อมูลภายใน แล้วนำไปเก็บไว้ที่รีจิสเตอร์ IR (Instruction Register) เพื่อส่งไปให้กับวงจรเวลาและควบคุมทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆ ต่อไป ในกรณีที่เลือกรอมภายนอก โดยป้อนลอจิก 1 เข้าที่ขา EA จะทำให้วงจรเวลาและควบคุมส่งสัญญาณไปยังพอร์ต 0 และพอร์ต 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนเส้นทางหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ต 0 ไปยังเส้นทางข้อมูลภายในแล้วไปเก็บไว้ที่รีจิสเตอร์ IR เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจากรอมภายใน การทำงานในช่วงค่าตำแห่งในหน่วยความจำไปยังหน่วยความจำ แล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ในรีจิสเตอร์ IR เรียกว่า “ช่วงของการเฟตช์” (Fetch) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า “Execute Cycle”

ในส่วนของหน่วยความจำข้อมูลภายในที่อ้างอิงแบบ direct และ Indirect นั้นจะสามารถแบ่งออกได้ 3 ส่วน ดังแสดงในรูปที่ 8 โดยมีรายละเอียดดังนี้

- ส่วนที่ 1 เรียกกว่า Register Banks 0-3 ซึ่งอยู่ที่ตำแหน่งหน่วยความจำข้อมูลภายใน ตั้งแต่ 00H ถึง 1FH จำนวน 32 ไบต์โดยจะแบ่งออกเป็นชุด ชุดละ 8 ไบต์จำนวน 4 ชุด ซึ่งแต่ละชุดจะมีชื่อเรียกเป็น R0 ถึง R7 จะเป็น Register ที่ใช้งาน โดยเมื่อ 8051 ถูกรีเซ็ต Register Bank 0 จะถูกเลือกใช้
- ส่วนที่ 2 เรียกว่า Bit Addressable Area ซึ่งมีขนาด 16 ไบต์ที่ตำแหน่งหน่วยความจำข้อมูล 20H ถึง 2FH ในส่วนนี้สามารถที่จะอ้างอิงข้อมูลได้เป็นระดับบิตถึง 128 บิต โดยการอ้างอิงตำแหน่งโดยตรงในลักษณะบิต ตั้งแต่ตำแหน่ง 00H ถึง 7FH



รูปที่ 14 แสดงหน่วยความจำข้อมูลภายใน

- ส่วนที่ 3 เรียกว่า Scratch Pad Area จะอยู่ที่ตำแหน่งตั้งแต่ 30H ถึง 7FH ซึ่งเป็นบริเวณหน่วยความจำข้อมูลภายในเอนกประสงค์ที่ผู้ใช้สามารถใช้ได้โดยตรง นอกจากนี้ยังสามารถใช้หน่วยความจำข้อมูลบริเวณนี้สำหรับการเก็บข้อมูลแบบ Stack ได้ด้วย

ในส่วนของหน่วยความจำข้อมูลภายในที่ใช้อ้างอิงแบบ Direct เพียงอย่างเดียวหรือที่เรียกว่า SFR ซึ่งเป็นส่วนสำหรับเก็บหรือกำหนดการทำงานภายในของ 8051 ดังแสดงในรูปที่ 8

ในส่วนของบริเวณนี้จะมีขนาด 128 ไบต์แต่ในการใช้งานนั้นใช้ได้เฉพาะตำแหน่งซึ่งแสดงไว้ในรูปที่ 9 เท่านั้น หากผู้ใช้อ้างตำแหน่งนอกเหนือจากนี้นั้นจะได้ข้อมูลที่คาดเดาไม่ได้ โดยแต่ละตำแหน่งจะมีหน้าที่ดังนี้

ACC : เป็น Accumulator ซึ่งเป็นรีจีสเตอร์สำหรับการประมวลผลทางคณิตศาสตร์และลอจิก โดยผู้ใช้สามารถอ้างอิงได้ในรูปแบบของไบต์หรือระดับบิตได้

B : เป็นรีจีสเตอร์พิเศษสำหรับใช้กับคำสั่งในการคูณหรือหาร นอกจากนี้ยังใช้เป็นรีและลอจิก โดยผู้ใช้สามารถอ้างอิงได้ในรูปแบบของไบต์หรือระดับบิตได้

PSW : เป็นรีจีสเตอร์ Program Status Word หรือ แฟลกจะแสดงสถานะการทำงานของ 8051 สำหรับการตรวจสอบซึ่งจะอธิบายรายละเอียดในภายหลัง

8 Bytes

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW[1]							D7
C8	T2CON[1,2]	T2MOD[2]	RCAP2L[2]	RCAP2H[2]	TL2[2]	TH2[2]		CF
C0								C7
B8	IP[1]							BF
B0	P3							B7
A8	IE[1]							AF
A0	P2							A7
98	SCON[1]	SBUF						9F
90	P1							97
88	TCON[1]	TMOD[1]	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH				PCON[1] 87

[1] Bit Addressable

Notes 1. SFRs converting mode or control bits
2. AT89C52 only

รูปที่ 15 แสดงรายละเอียดของ Special Function Register

SP : เป็นรีจีสเตอร์สำหรับชี้หน่วยความจำข้อมูลภายในสำหรับการเก็บแบบ Stack

DPTR : เป็นรีจีสเตอร์ขนาด 16 บิต โดยแบ่งเป็น 8 บิตบนและ 8 บิตล่าง ให้สำหรับชี้ตำแหน่งของหน่วยความจำข้อมูลภายนอกหรือสำหรับการอ่านตารางข้อมูลของหน่วยความจำโปรแกรม

P0 : เป็นรีจีสเตอร์สำหรับพอร์ต 0 ของ 8051

P1 : เป็นรีจีสเตอร์สำหรับพอร์ต 1 ของ 8051

P2 : เป็นรีจีสเตอร์สำหรับพอร์ต 2 ของ 8051

P3 : เป็นรีจีสเตอร์สำหรับพอร์ต 3 ของ 8051

IP : เป็นรีจีสเตอร์สำหรับกำหนดลำดับความสำคัญของการอินเตอร์รัพท์ ของ 8051

IE : เป็นรีจีสเตอร์สำหรับกำหนดการรับหรือไม่รับการอินเตอร์รัพท์ ของ 8051

TMOD : เป็นรีจีสเตอร์สำหรับควบคุมหน้าที่ของ Timer/Counter ของ 8051

TCON : เป็นรีจีสเตอร์สำหรับควบคุมการทำงานของ Timer/Counter ของ 8051

T2CON : เป็นรีจีสเตอร์สำหรับควบคุมการทำงานของ Timer/Counter 2 ของ 8052

TH0 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 0 8 บิตบน

TL0 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 0 8 บิตล่าง

TH1	:	เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 1 8 บิตบน
TL1	:	เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 1 8 บิตล่าง
TH2	:	เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของTimer/Counter 2 8 บิตบนของ8052
TL2	:	เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 2 8 บิตล่างของ 8052
RCAP2H	:	เป็น Capture Register ของ Timer/Counter 2 8 บิตบนของ 8052
RCAP2L	:	เป็น Capture Register ของ Timer/Counter 2 8 บิตล่างของ 8052
SCON	:	เป็นรีจีสเตอร์สำหรับควบคุมการรับส่งข้อมูลแบบอนุกรมของ 8051
SBUF	:	เป็นรีจีสเตอร์สำหรับเก็บพักข้อมูลที่ได้จากการรับส่งข้อมูลแบบอนุกรมของ 8051
PCON	:	เป็นรีจีสเตอร์สำหรับควบคุมการทำงานของ 8051 ด้านเกี่ยวกับการใช้กำลังไฟฟ้า

ในส่วนของรีจีสเตอร์ SFR นี้สามารถที่จะอ้างอิงในระดับบิตได้โดยตำแหน่งการอ้างอิงระดับบิตแสดงไว้ในตารางต่อไปนี้

Byte address	Bit address								
F8									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E8									
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D8									
D0	D7	D6	D5	D4	D3	D2	—	D0	PSW
B8	—	—	—	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	—	—	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	not bit addressable								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	not bit addressable								TH1
8C	not bit addressable								TH0
8B	not bit addressable								TL1
8A	not bit addressable								TL0
89	not bit addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	not bit addressable								PCON
83	not bit addressable								DPH
82	not bit addressable								DPL
81	not bit addressable								SP
80	87	86	85	84	83	82	81	80	P0

รูปที่ 16 แสดงตำแหน่งการอ้างอิงระดับบิตของรีจีสเตอร์ SFR

2.9 Special Function Register ที่ใช้สำหรับการควบคุมในหน้าที่ต่าง ๆ ของ 8051 มีรายละเอียดดังนี้

PSW

เป็นรีจีสเตอร์ขนาด 8 บิตที่ใช้สำหรับแสดงสถานะการทำงานของ 8051 ที่สามารถตรวจสอบได้โดยการตรวจสอบนั้นสามารถการอ้างอิงได้ในระดับบิต โดยมีความหมายของแต่ละบิตดังนี้

CY	AC	F0	RS1	RS0	OV	-	P

CY หรือสามารถอ้างอิงระดับบิตเป็น PSW.7 เป็นบิตแสดงสถานะของตัวทดเมื่อมีการทำงานคำสั่งที่เกี่ยวกับคณิตศาสตร์และลอจิกโดยที่ถ้าเป็น “1” แล้วค่าของแอกคิวมูลเลเตอร์เกิน 255 (ฐานสิบ)หรือ FFH

AC หรือสามารถอ้างอิงระดับบิตเป็น PSW.6 เป็นบิตแสดงสถานะของตัวทดเสริมเป็น “1” เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์แล้วทำให้เกิดการทดข้ามจากบิต 3 มายังบิต 4 มักใช้ในการแปลงค่าเป็นเลขฐานสิบ

F0 หรือสามารถอ้างอิงระดับบิตเป็น PSW.5 เป็นบิตเอนกประสงค์ที่ผู้ใช้สามารถกำหนดสถานะการทำงานของโปรแกรมที่เขียนขึ้นได้ซึ่งจะไม่มีการเปลี่ยนแปลงใดๆขณะการทำงานของคำสั่ง

RS1, RS0 หรือสามารถอ้างอิงระดับบิตเป็น PSW.4 และ PSW.3 ตามลำดับ เป็นบิตสำหรับกำหนดเลือกชุดของรีจีสเตอร์ของ 8051 ซึ่งมี 4 ชุดที่ผู้ใช้สามารถเลือกใช้งานได้

OV หรือสามารถอ้างอิงระดับบิตเป็น PSW.2 เป็น “1” เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วทำให้เกิดการทดข้ามจากบิต 6 มายังบิต 7 ของแอกคิวมูลเลเตอร์หรือแอกคิวมูลเลเตอร์มีค่าเกิน 127 (ฐานสิบ)หรือสามารถอ้างอิงระดับบิตเป็น PSW.1 เป็นบิตเอนกประสงค์ที่ผู้ใช้สามารถใช้กำหนดสถานะของโปรแกรมเองได้อย่างอิสระ

P หรือสามารถอ้างอิงระดับบิตเป็น PSW. 0 เป็นบิตใช้สำหรับการตรวจสอบจำนวนของ “1” ในข้อมูลของแอกคิวมูลเลเตอร์ ถ้าแอกคิวมูลเลเตอร์มีจำนวนบิตที่เป็น “1” รวมกันเป็นเลขคู่บิตนี้จะเป็น “0” เป็นเลขคี่จะเป็น“1”

PCON

เป็นรีจีสเตอร์สำหรับควบคุมการทำงานของ 8051 ซึ่งเป็นรีจีสเตอร์ที่ไม่สามารถอ้างอิงระดับบิตได้ ดังมีรายละเอียดดังนี้

SMOD	-	-	-	GF1	GF0	PD	IDL

SMOD : เป็นบิตกำหนดอัตราความเร็วของการรับส่งข้อมูลแบบอนุกรมที่ใช้ Timer 1 เช่น ถ้า เวลาในการรับส่งเมื่อ SMOD = 1 จะทำให้ความเร็วในการรับส่งแบบอนุกรมเป็น 2 เท่า เมื่อเลือกการใช้งานใน Mode 1,2,3 ของการรับส่งแบบอนุกรม เป็นบิตที่ไม่ใช้งานซึ่งสงวนไว้สำหรับหน้าที่เพิ่มเติมในอนาคตของ 8051 ดังนั้นผู้ใช้จึงไม่ควรกำหนดในบิตนี้เป็น 1

GF0-GF1 : เป็นบิตเอนกประสงค์ที่ผู้ใช้สามารถกำหนดการทำงานได้

PD : เป็นบิตสำหรับให้ 8051 เข้าสู่การทำงานใน Power Down Mode ซึ่งจะมีเฉพาะเบอร์ที่มีโครงสร้างภายในเป็น CMOS เท่านั้น ซึ่งจะทำงานเมื่อบิตนี้เป็น 1

IDL : เป็นบิตสำหรับกำหนดให้ 8051 เข้าสู่การทำงานใน Idle Mode ซึ่งจะมีเฉพาะเบอร์ที่มีโครงสร้างภายในเป็น CMOS เท่านั้น ซึ่งจะทำงานเมื่อบิตนี้เป็น 1

ถ้าบิต PD และ IDL เป็น 1 ทั้งคู่ในเวลาเดียวกัน 8051 จะทำงานใน Power Down Mode ในการทำงานทั้งสองModeนี้เป็นการทำงานเพื่อประหยัดพลังงานของ 8051 โดยมีข้อแตกต่างดังนี้

Power Down Mode เมื่อมีการกำหนดให้บิตนี้เป็น 1 8051 จะทำงานอีกหนึ่งคำสั่งก่อนที่จะเข้าสู่ Power Down Mode โดยที่ส่วนสร้างความถี่อ้างอิงภายในจะหยุดทำงาน การทำงานทุกอย่างของ 8051 จะหยุด ข้อมูลต่าง ๆ ของหน่วยความจำภายใน รวมทั้ง SFR จะถูกรักษาไว้ ที่สัญญาณ ALE และ PSEN จะเป็นลอจิก 0 เมื่อเข้าสู่ Power Down Mode แรงดันไฟฟ้าที่จ่ายให้กับ 8051 สามารถลดลงเป็น 2โวลท์ ได้ อย่างไรก็ตามแรงดันไฟฟ้าจะต้องไม่ลดลงก่อนที่จะเข้าสู่ Power Down Mode และแรงดันไฟฟ้าจะต้องเข้าสู่ระดับปกติก่อนที่จะออกจาก Power Down Mode การออกจาก Power Down Mode จะกระทำได้เฉพาะทำการรีเซ็ต 8051 เท่านั้น ซึ่งจะทำให้ค่าของ SFR เปลี่ยนแปลง แต่ข้อมูลในส่วนของหน่วยความจำภายในไม่เปลี่ยนแปลง

Idle Mode เมื่อมีการกำหนดให้บิตนี้เป็น 1 8051 จะทำงานอีกหนึ่งคำสั่งก่อนเข้าสู่ Idle Mode โดยที่ความถี่การทำงานภายใน CPU ของ 8051 จะหยุดทำงานยกเว้นที่ให้กับส่วนของการอินเตอร์รัพท์ Timer และการรับส่งแบบอนุกรม ค่าต่าง ๆ ของรีจีสเตอร์จะถูกรักษาไว้ และสัญญาณของ ALE และ PSEN จะอยู่ที่ลอกจิก 1 ในการออกจาก Idle Mode จะสามารถทำได้สองวิธีได้แก่การรีเซ็ตและการอินเตอร์รัพท์

IE เป็นรีจีสเตอร์สำหรับการควบคุมการตอบรับการอินเตอร์รัพท์ของ 8051 ซึ่งสามารถอ้างอิงได้เป็นระดับบิตได้ โดยที่เมื่อให้ลอจิก 0 ที่บิตใด ๆ หมายความว่ายกเลิกการตอบ

รับการอินเตอร์รัพท์ของ 8051 และให้ลอจิก 1 ที่บิตใด ๆ เป็นการกำหนดให้ 8051 ตอบรับการอินเตอร์รัพท์นั้น ๆ ได้ โดยมีรายละเอียดดังนี้

EA	---	ET2	ES	ET1	EX1	ET0	EX0

EA หรือสามารถอ้างอิงระดับบิตได้เป็น IE.7 สำหรับบิตนี้จะควบคุมการตอบรับการอินเตอร์รัพท์ทั้งหมดของ 8051 โดยที่ถ้าเป็นลอจิก 0 8051 จะไม่ตอบรับการอินเตอร์รัพท์ทั้งหมดและลอจิก 1 เป็นการกำหนดให้สามารถควบคุมการตอบรับการอินเตอร์รัพท์ได้อิสระในแต่ละส่วนของการอินเตอร์รัพท์นั้นๆ หรือสามารถอ้างอิงระดับบิตได้เป็น IE.6 เป็นบิตที่สงวนไว้สำหรับอนาคตของ 8051 ซึ่งผู้ใช้กำหนดลอจิก 1 ไม่ได้

ET2 หรือสามารถอ้างอิงระดับบิตได้เป็น IE.5 เป็นบิตสำหรับควบคุมการตอบรับการอินเตอร์รัพท์ที่เกิดจาก Timer 2 หรือจาก Capture Interrupt ซึ่งมีเฉพาะเบอร์ 8052 เท่านั้น

ES หรือสามารถอ้างอิงระดับบิตได้เป็น IE.4 เป็นบิตสำหรับควบคุมการตอบรับการอินเตอร์รัพท์ในส่วนของการรับส่งแบบอนุกรม

ET1 หรือสามารถอ้างอิงระดับบิตได้เป็น IE.3 เป็นบิตสำหรับควบคุมการตอบรับการอินเตอร์รัพท์ที่เกิดจาก Timer 1

EX1 หรือสามารถอ้างอิงระดับบิตได้เป็น IE.2 เป็นบิตสำหรับควบคุมการตอบรับการอินเตอร์รัพท์ที่เกิดจากขา INT1

ET0 หรือสามารถอ้างอิงระดับบิตได้เป็น IE.1 เป็นบิตสำหรับควบคุมการตอบรับการอินเตอร์รัพท์ที่เกิดจาก Timer 0

EX0 หรือสามารถอ้างอิงระดับบิตได้เป็น IE.0 เป็นบิตสำหรับควบคุมการตอบรับการอินเตอร์รัพท์ที่ เกิดจากขา INT0

สำหรับการกำหนดการอินเตอร์รัพท์ของ 8051 มีลำดับการกำหนดได้ดังนี้

1. กำหนดให้บิต EA ใน IE รีจีสเตอร์เป็น 1

2. กำหนดลอจิก 1 ให้กับบิตที่ควบคุมการตอบรับการอินเตอร์รัพท์ที่จะใช้งานใน IE รีจีสเตอร์

3. เขียนโปรแกรมที่จะใช้ในการตอบรับการอินเตอร์รัพท์ในตำแหน่งของหน่วยความจำโปรแกรมต่อไปนี้ตามตารางที่ 2

ตารางที่ 2

แหล่งการอินเตอร์รัพท์	ตำแหน่งของหน่วยความจำ
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI และ TI	0023H
TF2 และ EXF2	002BH

สำหรับการอินเตอร์รัพท์ที่เกิดจากขา INT0 และ INT1 (P3.2 และ P3.3) สามารถกำหนดการตอบรับการอินเตอร์รัพท์ที่ระดับสัญญาณ หรือขอบของสัญญาณได้จากบิต IT0 และ IT1 ในรีจีสเตอร์ TCON ซึ่งจะมีรายละเอียดในส่วนของรีจีสเตอร์ TCON

IP

เป็นรีจีสเตอร์สำหรับการกำหนดลำดับความสำคัญของการอินเตอร์รัพท์ ซึ่ง 8051 มีลำดับความสำคัญการตอบรับการอินเตอร์รัพท์สองลำดับ โดยที่ถ้าเป็นลอจิก 1 จะมีลำดับสูงและลอจิก 0 จะมีลำดับต่ำ ดังรายละเอียดต่อไปนี้

—	—	PT2	PS	PT1	PX1	PT0	PX0

- หรือสามารถอ้างอิงระดับบิตเป็น IP.7 และ IP.6 เป็นบิตที่สงวนสำหรับอนาคต
- ผู้ใช้ไม่ควรกำหนดเป็นลอจิก 1

PT2 หรือสามารถอ้างอิงระดับบิตได้เป็น IP.5 เป็นบิตกำหนดลำดับความสำคัญการอินเตอร์รัพท์ที่ได้จาก Timer 2 ซึ่งจะมีเฉพาะเบอร์ 8052

PS หรือสามารถอ้างอิงระดับบิตได้เป็น IP.4 เป็นบิตกำหนดลำดับความสำคัญการอินเตอร์รัพท์ที่เกิดจากการรับส่งข้อมูลแบบอนุกรม

PT1 หรือสามารถอ้างอิงระดับบิตได้เป็น IP.3 เป็นบิตกำหนดลำดับความสำคัญการอินเตอร์รัพท์ที่เกิดจาก Timer 1

PX1 หรือสามารถอ้างอิงระดับบิตได้เป็น IP.2 เป็นบิตกำหนดลำดับความสำคัญการอินเตอร์รัพท์ที่เกิดจากขา INT1

PT0 หรือสามารถอ้างอิงระดับบิตได้เป็น IP.2 เป็นบิตกำหนดลำดับความสำคัญของอินเตอร์รัพท์ที่เกิดจาก Timer 0

PX0 หรือสามารถอ้างอิงระดับบิตได้เป็น IP.0 เป็นบิตกำหนดลำดับความสำคัญทางอินเตอร์รัพท์ที่เกิดจากขา INT0

ในกรณีที่มีการอินเตอร์รัพท์เกิดขึ้นพร้อมกัน ถ้ามีลำดับความสำคัญต่างกัน 8051 จะตอบรับการอินเตอร์รัพท์ที่มีลำดับสูงก่อน แต่ถ้าลำดับความสำคัญการอินเตอร์รัพท์เท่ากัน 8051 จะมีลำดับการอินเตอร์รัพท์จากมากไปน้อยดังนี้

IE0

TF0

IE1

TF1

IR และ TI

TF2 หรือ EXF2

TCON

เป็นรีจีสเตอร์ใช้สำหรับควบคุมการทำงานของ Timer/Counter และระดับการตอบรับการอินเตอร์รัพท์ที่เกิดจากขาภายนอก ซึ่งสามารถอ้างอิงระดับบิตได้ มีรายละเอียดดังนี้

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.7 เป็นบิตแสดงสถานะเมื่อเกิด Over Flow ที่เกิดจาก Timer 1 จะเป็นลอจิก 1 เมื่อเกิด Over Flow และเป็นลอจิก 0 เมื่อเข้าสู่โปรแกรมการตอบรับการอินเตอร์รัพท์

TR1 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.6 เป็นบิตสำหรับควบคุมการทำงานของ Timer 1โดยที่ลอจิก 1 ให้ Timer 1 ทำงานและหยุดเมื่อลอจิก 0

TF0 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.5 เป็นบิตแสดงสถานะเมื่อเกิด Over Flowที่เกิดจาก Timer 0 จะเป็นลอจิก 1 เมื่อเกิด Over Flow และเป็นลอจิก 0 เมื่อเข้าสู่โปรแกรมการตอบรับการอินเตอร์รัพท์

TR0 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.4 เป็นบิตสำหรับควบคุมการทำงานของTimer 0 โดยที่ลอจิก 1 ให้ Timer 0 ทำงานและหยุดเมื่อลอจิก 0

IE1 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.3 เป็นบิตแสดงสถานะการอินเตอร์รัพท์ที่เกิดจากขอบของสัญญาณอินเตอร์รัพท์ที่เกิดจากขา INT1 โดยจะเป็นลอจิก 1 เมื่อตรวจพบขอบของสัญญาณอินเตอร์รัพท์ภายนอก และเป็นลอจิก 0 เมื่อตอบรับการอ่านอินเตอร์รัพท์

IT1 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.2 เป็นบิตสำหรับกำหนดการตอบรับสัญญาณอินเตอร์รัพท์ที่เกิดจากขา INT1 ถ้าลอจิก 1 จะเป็นการตอบรับการอินเตอร์รัพท์ที่ขอบขาลงของสัญญาณอินเตอร์รัพท์และลอจิก 0 จะเป็นการตอบรับการอินเตอร์รัพท์ที่ระดับลอจิก 0

IE0 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.1 เป็นบิตแสดงสถานะการอินเตอร์รัพท์ที่เกิดจากขอบของสัญญาณอินเตอร์รัพท์ที่เกิดจากขา INT0 โดยจะเป็นลอจิก 1 เมื่อตรวจพบขอบของสัญญาณอินเตอร์รัพท์ภายนอก และเป็นลอจิก 0 เมื่อมีการตอบรับการอินเตอร์รัพท์

IT0 หรือสามารถอ้างอิงระดับบิตได้เป็น TCON.0 เป็นบิตสำหรับกำหนดการตอบรับสัญญาณการอินเตอร์รัพท์ที่เกิดจากขา INT0 ถ้าลอจิก 1 จะเป็นการตอบรับการอินเตอร์รัพท์ที่ขอบของขาลงของสัญญาณอินเตอร์รัพท์ และลอจิก 0 จะเป็นการตอบรับการอินเตอร์รัพท์ที่ระดับลอจิก 0

TMOD

เป็นรีจีสเตอร์สำหรับควบคุม Mode การทำงานของ Timer/Counter และไม่สามารถอ้างอิงระดับบิตได้ ซึ่งจะแบ่งเป็นสองส่วน โดยที่สี่บิตแรกของรีจีสเตอร์นี้ใช้สำหรับ Timer/Couter 0 และสี่บิตบนสำหรับ Timer/Counter 1 มีรายละเอียดดังนี้

Timer 1				Timer 0			
GATE	C/T	M1	M0	GATE	C/T	M1	M0

GATE เป็นบิตใช้สำหรับการควบคุม Timer/Counter ถ้าเป็นลอจิก 1 Timer/Counter นั้นจะทำงานเมื่อขาสัญญาณที่ขา INTx เป็นลอจิก 1 เท่านั้น และบิต TRx ใน TCON ต้องเป็นลอจิก 1 ด้วย ซึ่งจะเป็นการควบคุมด้วย Hardware เมื่อเป็นลอจิก 0 การควบคุม Timer/Counterนั้น ๆ จะเป็นการควบคุมด้วยโปรแกรมที่บิต TRx ของรีจนสเตอร์ TCON เท่านั้น

C/T เป็นบิตสำหรับเลือกการทำงานของ Timer/Counter เป็น Timer หรือ Counter โดยที่ถ้าเป็นลอจิก 1 จะทำหน้าที่เป็น Counter โดยรับสัญญาณจากขา Tx ของ 8051 และลอจิก 0 เป็น Timer โดยความถี่ที่นับได้จากความถี่ภายในของ 8051

M0,M1 เป็นบิตสำหรับเลือกการทำงานของ Timer/Counter ดังแสดงในตารางที่ 3 ต่อไปนี้

M1	M0	การทำงาน
0	0	เป็น Timer ขนาด 13 บิต เมื่อการทำงานของตระกูล 48
0	1	เป็น Timer/Counter ขนาด 16 บิต
1	0	เป็น Timer/Counter ขนาด 8 บิตที่สามารถตั้งค่าใหม่อัตโนมัติ
1	1	Timer/Counter 0 จะถูกแบ่งออกเป็น Timer/Counter 2 ชุดขนาด 8 บิต โดยที่ TL0 จะเป็น Timer/Counter ขนาด 8 บิตที่ถูกควบคุมของ Timer/Counter 0 และ TH0 จะเป็น Timer/Counter ขนาด 8 บิต ที่ควบคุมโดยชุดควบคุมของ Timer/Counter 1
1	1	Timer/Counter 1 จะหยุดทำงาน

T2CON

เป็นรีจีสเตอร์ของ 8052 จะไม่กล่าวถึงรายละเอียด

SCON

เป็นรีจีสเตอร์ใช้สำหรับควบคุมการรับส่งข้อมูลแบบอนุกรมซึ่งสามารถอ้างอิงระดับบิตได้โดยมีรายละเอียดดังนี้

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0-SM1 หรือสามารถอ้างอิงระดับบิตได้เป็น SCON.7 และ SCON.6 ตามลำดับ เป็นบิตสำหรับเลือกหน้าที่การทำงานของการรับส่งข้อมูลแบบอนุกรม

SM0	SM1	Mode	รายละเอียด	ความเร็วการรับส่ง
0	0	0	SHIFT REGISTER ขนาด 8 บิต	Fosc/12
0	1	1	8 บิต UART	ตั้งค่าได้
1	0	2	9 บิต UART	Fosc/64 หรือFosc/32
1	1	3	9 บิต UART	ตั้งค่าได้

ตารางที่ 4 แสดงการทำงาน SM0 SM1

SM2 หรือสามารถอ้างอิงระดับบิตได้เป็น SCON.5 เป็นบิตสำหรับกำหนดการทำงานในลักษณะ Multiprocessor ใน Mode 2 และ Mode 3 ถ้าเป็นลอจิก 1 แล้ว RI จะไม่เปลี่ยนแปลง ถ้าข้อมูลบิตที่ 9 (RB8) เป็น 0 สำหรับ Mode 1 RI จะไม่เปลี่ยนแปลง ถ้า STOP บิต ไม่ได้รับถ้าเลือกการทำงาน Mode 0 แล้ว SM2 จะต้องเป็นลอจิก 0

REN หรือสามารถอ้างอิงระดับบิตได้เป็น SCON.4 เป็นบิตสำหรับใช้ควบคุมการรับโดยโปรแกรมโดยที่ลอจิก 1 จะมีการรับและลอจิก 0 จะไม่รับ

TB8 หรือสามารถอ้างอิงระดับบิตได้เป็น SCON.3 เป็นบิตที่ 9 ในการส่งข้อมูลใน Mode 2 และ Mode 3 ซึ่งสามารถกำหนดได้โดยโปรแกรมการทำงานRB8หรือสามารถอ้างอิงระดับบิตได้เป็น SCON.2 ใน Mode 2 และ Mode 3 ใช้สำหรับรับข้อมูลบิตที่ 9 สำหรับใน Mode 1 ถ้า SM2 = 0 บิตนี้จะเป็นการรับ STOP บิต และ Mode 0 จะไม่ใช้งานบิตนี้

TI หรือสามารถอ้างอิงระดับบิตได้เป็น SCON.1 เป็นบิตบอกสถานะในการส่ง โดยที่จะเป็นลอจิก 1 เมื่อสิ้นสุดการส่งบิตที่ 8 ใน Mode 0 หรือสิ้นสุดการส่ง STOP บิตใน Mode อื่น และจะต้องกำหนดเป็นลอจิก 0 ด้วยโปรแกรม

RI หรือสามารถอ้างอิงระดับบิตได้เป็น SCON.0 เป็นบิตบอกสถานะการรับ จะเป็นลอจิก 1 เมื่อสิ้นสุดการรับบิตที่ 8 ใน Mode 0 หรือครึ่งช่วงเวลาของ STOP บิต สำหรับ Mode อื่น และจะต้องกำหนดให้เป็นลอจิก 0 ด้วยโปรแกรม

2.10 การคำนวณความเร็วการรับส่งข้อมูลแบบอนุกรม (Generating Baud Rate)

การกำหนดความเร็วการรับส่งข้อมูลแบบอนุกรมสามารถแบ่งออกได้ตาม Modeการทำงานดังนี้ **Mode 0**

ความเร็วการรับส่งข้อมูลแบบอนุกรมใน Mode นี้จะกำหนดอัตราการรับส่งตายตัวเท่ากับ 1/12 ของความถี่ของชุดกำเนิดความถี่อ้างอิงของ 8051 และจะไม่ใช้ Timer/Counter ดังนั้นเพียงกำหนดที่รีจีสเตอร์ SCON ก็เพียงพอ จะได้

$$\text{Baud Rate} = \frac{\text{Osc. Freq}}{12}$$

Mode 1

ในการกำหนดความเร็วการรับส่งข้อมูลแบบอนุกรมใน Mode 1 นี้จะใช้ Timer 1 เป็นฐานเวลาของการทำงาน โดยจะใช้การทำงานของ Timer 1 ใน Mode 2 (Auto-Reload) โดยสามารถคำนวณได้ดังนี้

$$\text{Baud Rate} = \frac{K * \text{Osc. Freq}}{32*12*[256-(TH1)]}$$

K = 1 เมื่อ SMOD ในรีจีสเตอร์ PCON = 0

K = 2 เมื่อ SMOD ในรีจีสเตอร์ PCON = 1

ส่วนมากแล้ว ผู้ใช้จะทราบค่าของ Baud Rate ที่จะส่งได้นั้น จะได้ค่าของ Time 1 สำหรับ Reload ได้เป็น

$$TH1 = 256 - \frac{K * \text{Osc. Freq.}}{384 * \text{Baud Rate}}$$

จากตารางต่อไปนี้ แสดงค่า Baud Rate ต่าง ๆ และค่า Reload ของ Timer 1

Baud Rate	f_{osc}	SMOD	Timer 1		
			C/T	Mode	Reload Value
Mode 0 Max: 1 MHZ	12 MHZ	X	X	X	X
Mode 2 Max. 375K	12 MHZ	1	X	X	X
Modes 1, 3: 62.5K	12 MHZ	1	0	2	FFH
19.2K	11.059 MHZ	1	0	2	FDH
9.6K	11.059 MHZ	0	0	2	FDH
4.8K	11.059 MHZ	0	0	2	FAH
2.4K	11.059 MHZ	0	0	2	F4H
1.2K	11.059 MHZ	0	0	2	E8H
137.5	11.986 MHZ	0	0	2	1DH
110	6 MHZ	0	0	2	72H
110	12 MHZ	0	0	1	FEEBH

ตารางที่ 5 แสดงอัตราการส่งข้อมูลแบบอนุกรม

Mode 2 ความเร็วการรับส่งใน Mode นี้จะเป็นค่าคงที่ซึ่งมี 2 ค่า ขึ้นกับค่า SMOD ในรีจิสเตอร์ PCON ดังนี้

เมื่อ SMOD = 1 Baud Rate = 1/32 Osc. Freq.

เมื่อ SMOD = 0 Baud Rate = 1/64 Osc. Freq.

Mode 3 การกำหนดความเร็วการรับส่งใน Mode 3 จะคิดเช่นเดียวกับการคิดใน Mode 1 สำหรับค่าเริ่มต้นของรีจีสเตอร์ SFR เมื่อ 8051 ถูกรีเซ็ต จะมีค่าดังแสดงในตารางที่ 6 ต่อไปนี้

SFR Name	Reset Value
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0–P3	FFH
IP (8051)	XXX00000B
IP (8052)	XX000000B
IE (8051)	0XX00000B
IE (8052)	0X000000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2 (8052)	00H
TL2 (8052)	00H
RCAP2H (8052)	00H
RCAP2L (8052)	00H
SCON	00H
SBUF	Indeterminate
PCON (HMOS)	0XXXXXXXB
PCON (CHMOS)	0XXX0000B

ตารางที่ 6 แสดงค่า SFR ในโหมด 3

2.11 ความรู้เบื้องต้นเกี่ยวกับ LED ตัวเลข 7 ส่วน

LED ตัวเลข 7 ส่วนประกอบขึ้นจาก LED จำนวน 7 ตัวที่บรรจุอยู่ในตัวถังเดียวกันและได้รับการจัดเรียงเป็นรูปตัวเลข LED แต่ละตัวจะถูกเรียกว่า ส่วน หรือ เซกเมนต์ (segment) แต่ละส่วนเซหรือเซกเมนต์มีชื่อเรียกแต่ต่างกันตามตำแหน่งที่ได้รับการจัดวางคือ a,b,c,d,e,f และ g ดังแสดงในรูป ส่วน dp เป็น LED อีก 1 ตัวที่บรรจุอยู่ใน LED ตัวเลข 7 ส่วนนี้ใช้เป็นตัวแสดงจุดทศนิยมในกรณีที่มีการแสดงผลในลักษณะเลขที่มีทศนิยม

LED ทุกตัวที่บรรจุอยู่ใน LED ตัวเลข 7 ส่วนนี้มีขาต่อร่วมกัน ซึ่งก็มีทั้งแบบต่อขาแคโทดร่วมกันเรียกว่า แบบแคโทดร่วม (common cathode) และแบบต่อขาแอโนดร่วมกันเรียกว่า แบบแอโนดร่วม (common anode) การขับให้ LED ตัวเลข 7 ส่วนแบบแคโทดร่วมสว่าง จะต้องจ่ายไฟลบเข้าที่ขาร่วม แล้วจ่ายไฟบวกเข้าที่ขาแดโนด ซึ่งก็คือขาของแต่ละเซกเมนต์ นั่นเอง ดังแสดงในรูปที่ P13-2 (ก) ในขณะที่ LED ตัวเลข 7 ส่วนแบบแอโนดร่วมจะต้องจ่ายไฟบวกเข้าที่ขาร่วม แล้วจ่ายไฟลบเข้าที่ขาแคโทด ซึ่งเป็นขาของแต่ละเซกเมนต์

LED ตัวเลข 7 ส่วนมีจำหน่ายทั้งแบบตัวเดี่ยว, ตัวคู่ และแบบที่มีมากกว่า 2 หลัก แต่ที่นิยมใช้งานและหาได้ง่ายมี 2 แบบคือ แบบตัวเดี่ยวและแบบตัวคู่ โดยมีการจัดขาดังในรูปที่ P13-3 จะเห็นได้ว่า LED ตัวเลข 7 ส่วนแบบตัวเดี่ยวมีขาต่อใช้งาน 10 ขา คือ ขา a,b,c,d,e,f,g,dp และขาร่วม (common) ซึ่งมี 2 ขา ถ้าเป็น LED ตัวเลข 7 ส่วนแบบตัวคู่มีขาต่อใช้งาน 20 ขา แบ่งเป็นขา a,b,c,d,e,f,g, และ dp อย่างละ 2 ขา รวม 16 ขาและขาร่วมอีกหลักละ 2 ขา การต่อขาร่วมของแต่ละหลักทั้ง 2 ขานั้น สามารถต่อใช้งานเพียงขาเดียวได้ เนื่องจากในโครงสร้างภายในของ LED ตัวเลข 7 ส่วน ขาร่วมนี้ต่อถึงกันอยู่แล้ว

การขับ LED ตัวเลข 7 ส่วนแบบหลักเดียว

ไมโครคอนโทรลเลอร์ MCCS-51 ไม่ควรนำมาขับ LED 7 ส่วนโดยตรง เนื่องจากความสามารถในการจ่ายแระแสเอาต์พุตร่วมไม่สูงมากนัก จึงต้องอาศัยไอซีบัฟเฟอร์มาช่วยในการขับ LED อาทิไอซีเบอร์ 74HC541 และที่เอาต์พุตของไอซีบัฟเฟอร์ที่ต่อกับ LED ตัเลข 7 ส่วนต้องมีตัวต้านทานจำกัดกระแสให้แก่ LED ในทุกเซกเมนต์ การกำหนดให้ LED ตัวเลข 7 ส่วนแสดงข้อมูลเป็นตัวเลขหรือเป็นสัญลักษณ์ใด ๆ ก็ตาม ต้องมีการกำหนดรูปแบบการแสดงผลของเซกเมนต์ต่างๆ ด้วยข้อมูลแต่ละบิตของไมโครคอนโทรลเลอร์แล้วใช้วิธีการเปิดตารางหรือ look up table

การเขียนโปรแกรมเพื่อขับ LED ตัวเลข 7 ส่วน ควรใช้การเปิดตารางข้อมูล (look up table) ช่วย แล้วใช้โปรแกรมหน่วงเวลาเพื่อให้ LED ในเซกเมนต์ที่ถูกสั่งให้ทำงานนั้นติดสว่างนานพอให้ผู้ใช้งานเห็นข้อมูลที่นำมาแสดงผลที่ LED ตัวเลข 7 ส่วนนั้น

การขับ LED ตัวเลข 7 ส่วนแบบมัลติเพล็กซ์

ในกรณีที่ต้องการให้ไมโครคอนโทรลเลอร์ MCS-51ขับ LED ตัวเลข 7 ส่วนมากกว่า 1 หลัก จะต้องใช้เทคนิคที่เรียกว่า การแสดงผลแบบมัลติเพล็กซ์ (multiplex) อันเป็นวิธีการขับให้ LED สว่างทีละหลักด้วยอัตราเร็วที่ตาของมนุษย์ไม่สามารถตรวจจับได้ทัน จึงดูเหมือนว่า LED ตัวเลข 7 ส่วนทุกหลักติดสว่างในเวลาเดียวกัน

การแสดงผลแบบมัลติเพล็กซ์นี้มีประโยชน์หลายประการดังนี้

1. ช่วยลดพลังงานไฟฟ้าที่ใช้ ทำให้ขนาดของแหล่งจ่ายกำลังงานไฟฟ้าเล็กลง ส่งผลให้ขนาดโดยรวมของระบบเล็กลงด้วย
2. ช่วยให้ไมโครคอนโทรเลอร์สามารถขับ LED ตัวเลข 7 ส่วนได้มากกว่า 1 หลัก โดยที่ใช้จำนวนพอร์ตเพิ่มเติมเฉพาะการติดต่อกับขาคอมมอนเท่านั้น
3. ลดจำนวนตัวต้านทานที่ใช้ในการจำกัดกระแสของ LED ในแต่ละเซกเมนต์ ยกตัวอย่าง LED ตัวเลข 7 ส่วนหนึ่งหลักต้องใช้ตัวต้านทานจำกัดกระแส 8 ตัว ถ้าหากขับ LED ตัวเลข 7 ส่วน 4 หลักโดยตรง ต้องใช้ตัวต้านทานมากถึง 32 ตัว ในขณะที่หากใช้วิธีการแสดงผลแบบมัลติเพล็กซ์ ยังคงใช้ตัวต้านทานเพื่อจำกัดกระแสให้ LED ในแต่ละเซกเมนต์เพียง 8 ตัวไม่ว่าจะขับ LED ตัวเลข 7 ส่วนกี่หลักก็ตาม

การขับ LED ตัวเลข 7 ส่วนแบบมัลติเพล็กซ์ จะทำการต่อขาของแต่ละเซกเมนต์ร่วมกันคือ เซกเมนต์ a ของทุกหลักจะต่อถึงกันไล่เรียงไปจนถึงเซกเมนต์ g ในบางงานที่ต้องใช้จุด dp ก็ต้องต่อขาของจุด dp รวมกันด้วย การควบคุมให้ LED ตัวเลข 7 ส่วนหลักใดติดสว่าง ทำได้โดยการจ่ายไฟเข้าที่ขาร่วมของ LED ตัวเลข 7 ส่วนหลักนั้น ๆ ยกตัวอย่างหาก LED ตัวเลข 7 ส่วนที่ใช้เป็นแบบแคโทดร่วม หากต้องการให้ LED ตัวเลข 7 ส่วนหลักที่ 3 ติดสว่างก็ให้ต่อขาร่วมของหลักที่ 3 ลงกราวด์หรือจ่ายไฟลบ LED ตัวเลข 7 ส่วนหลักที่ 3 ก็จะติดสว่างตามข้อมูลที่ส่งเข้ามายังขาของแต่ละเซกเมนต์

บทที่ 3
การคำนวณและการสร้าง

3.1 ขั้นตอนการปฏิบัติงาน

1. เริ่มต้นจากการออกแบบว่าเราต้องการให้อาคารจอดรถนี้มีการทำงานเป็นอย่างไร
2. ศึกษาองค์ประกอบต่างๆของโครงงานนี้ เช่น ด้านฮาร์ดแวร์ ซอฟแวร์
3. เริ่มต้นจากด้านทางเข้าอาคาร สร้างแบบจำลอง ศึกษาคำนวณพัลส์มอเตอร์ให้เคลื่อนที่ได้องศาที่ต้องการ ทำกล่องที่เสียบบัตรจอดรถ การวางตำแน่งของอุปกรณ์
4. จับวางเซนเซอร์ในจุดต่าง ทางเข้า 2 ตัว ทางออก 1 ตัวและบริเวณที่จอดอย่างละ1 ตัว
5. ประกอบ LED 7 segment ตรงด้านทางเข้า
6. ศึกษาวงจรที่ใช้ทั้งหมด เช่น วงจรอินฟาเรดที่ใช้แยกระหว่างสี ขาว กับ ดำ ประยุกต์มาจากวงไมโครเมาท์
7. เขียนโปรแกรมควบคุมในส่วนต่าง เช่น ตรวจสอบรถเข้าออก นำค่าบัตรที่ได้ไปใช้ในการเริ่มต้นจับเวลาจอด คำนวณราคาค่าจอดรถ
8. ทดสอบระบบทั้งดูข้อผิดพลาด และทำการแก้ไข
9. ติดตั้งอุปกรณ์ทั้งลงในแบบจำลองที่สร้างไว้
10. แล้วทำการทดสอบอีกครั้ง

3.2 ขั้นตอนการทำงานทั้งหมดของระบบ

เริ่มต้นการทำงานจาก อุปกรณ์เซนต์เซอร์จะทำหน้าที่ตรวจเช็คตลอดเวลาว่ามีรถเข้ามาจอดแล้วหรือยัง ซึ่งอุปกรณ์เซนต์เซอร์นี้จะถูกติดตั้งไว้กับพื้นตรงช่องที่รถจะเข้ามาจอด หลักทำงานของมันจะอาศัยแสงจากภายนอกหรือแสงจากแสงไฟบนเพดานมาตกกระทบที่ตัวมันทำให้มันทำงานและส่งค่าออกทางเอาท์พุตเป็น 1 ไปบอกหน่วยประมวลผลว่าขณะนี้ยังไม่มีรถเข้ามาจอด แต่เมื่อใดที่มีรถเข้ามาจอดบังแสง มันจะส่งค่าเอาท์พุตที่เป็น 0 ไปให้กับหน่วยประมวลผลเพื่อบอกว่าขณะนี้มีผู้ใช้บริการนำรถเข้ามาจอดแล้วชุดอุปกรณ์สำหรับกดรหัสก็จะทำงาน

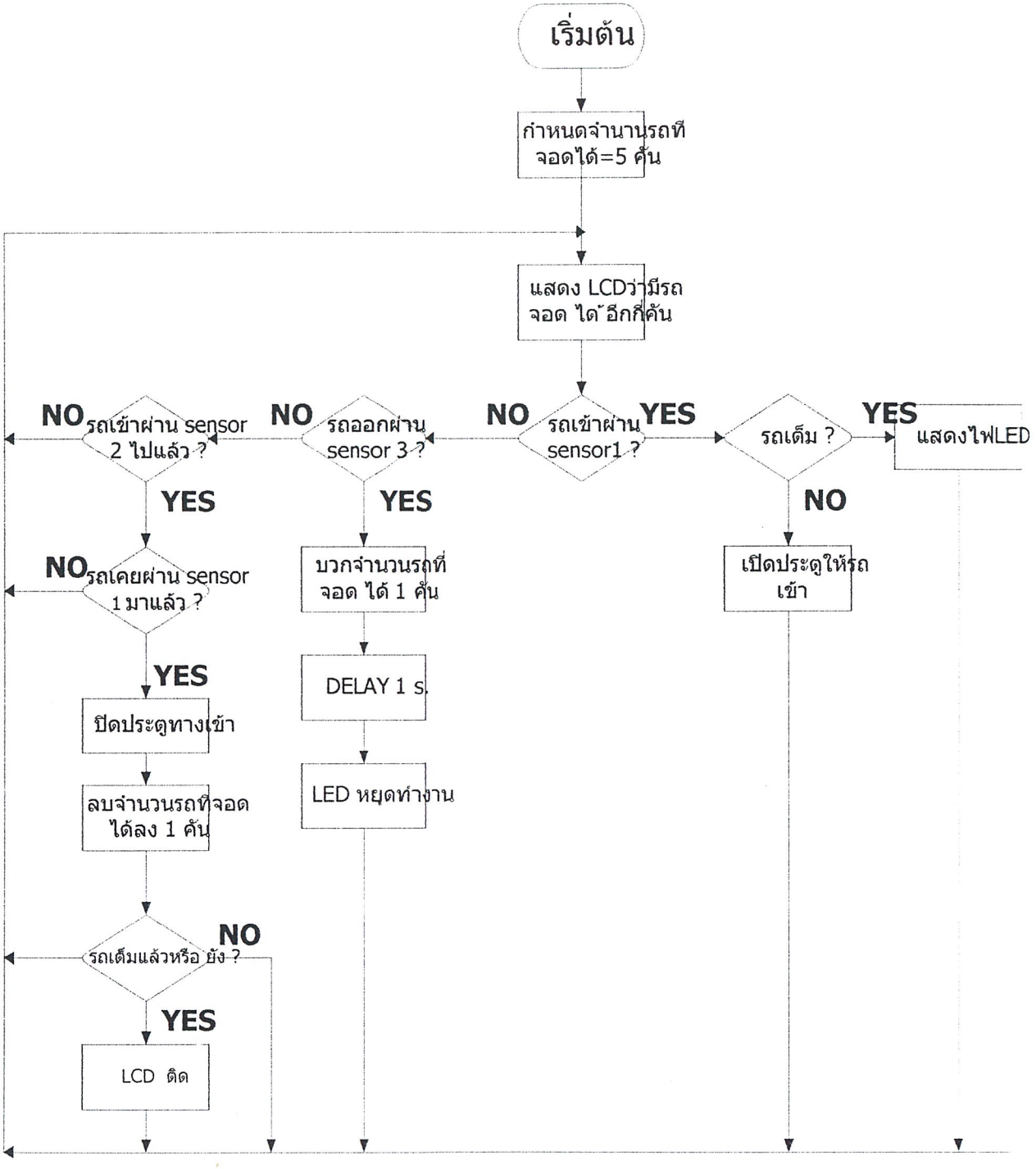
จากนั้นผู้ใช้บริการจะต้องเดินมากดรหัส ตัวเลข 3 ตัวที่เครื่องป้อนรหัสซึ่งเป็นตัวเลขใดก็ได้ เครื่องป้อนรหัสจะส่งค่าไปให้กับหน่วยประมวลผลรับรู้เพื่อเก็บค่าตัวเลข 3 ตัวที่ผู้ใช้บริการกด เมื่อผู้ใช้บริการกดรหัสแล้วโซลินอยด์ที่ทำหน้าที่ล็อคบัตรที่ติดตั้งอยู่ที่เครื่องตรวจเช็คบัตรก็จะทำงานโดยการปลดล็อคบัตรออก จากนั้นหน่วยประมวลผลจะทำหน้าที่เช็คว่าบัตรถูกดึงออกไปแล้วหรือยัง เมื่อบัตรถูกดึงออกไปโดยผู้ใช้บริการแล้ว มอเตอร์เกียร์จะหมุนคานที่ใช้ล็อครถขึ้นมาล็อครถ ซึ่งคานล็อครถจะเคลื่อนที่ขึ้นมาเรื่อยๆ จนชนลิมิตสวิตช์มอเตอร์เกียร์ก็จะหยุดหมุนจากนั้น

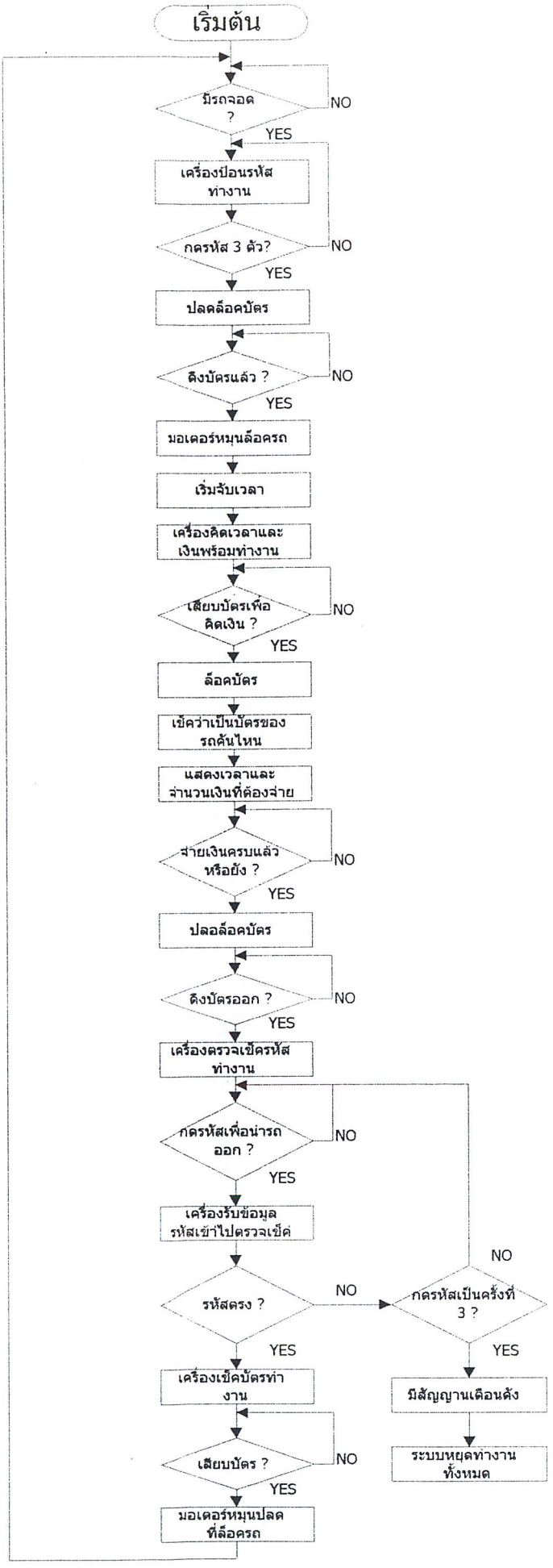
อุปกรณ์สำหรับจับเวลากับอุปกรณ์คิดเวลาและเงินก็เริ่มทำงานพร้อมรอผู้ใช้บริการกลับมาเสียบบัตรเพื่อคิดเงินเมื่อทำธุระเสร็จแล้ว อุปกรณ์คิดเวลาและเงินจะทำหน้าที่คอยเช็คตลอดว่าผู้ใช้บริการเสียบบัตรเพื่อคิดเงินในช่องเสียบแล้วหรือยัง

เมื่อผู้ใช้บริการทำธุระเสร็จแล้วกลับมานำบัตรมาเสียบเพื่อคิดเงิน โซลินอยด์จะล็อคบัตรไว้จนกว่าผู้ใช้บริการจะจ่ายเงินครบ จากนั้นหน่วยประมวลผลจะทำหน้าที่ตรวจสอบว่าบัตรที่มาเสียบว่าเป็นบัตรของช่องจอดรถใด เมื่อตรวจสอบเรียบร้อยแล้วจะแสดงผลเวลาและจำนวนเงินค่าบริการที่ต้องชำระให้กับผู้ใช้บริการทราบ จากนั้นจะตรวจสอบด้วยว่าผู้ใช้บริการชำระเงินค่าบริการครบตามที่แสดงผลไว้แล้วหรือยัง เมื่อผู้ใช้บริการชำระเงินเรียบร้อยแล้วโซลินอยดีก็จะปลดล็อคบัตร จากนั้นหน่วยประมวลผลจะทำหน้าที่คอยเช็คด้วยว่าบัตรถูกดึงออกไปแล้วหรือยัง เมื่อบัตรถูกดึงออกไปโดยผู้ใช้บริการแล้วเครื่องตรวจสอบรหัสก็พร้อมที่จะทำงานและทำการตรวจสอบด้วยว่าผู้ใช้บริการกดรหัสแล้วหรือยัง

เมื่อผู้ใช้บริการกดรหัสแล้วหน่วยประมวลผลจะนำรหัสที่ผู้ใช้บริการกดไปตรวจสอบกับรหัสที่กดไว้ในตอนแรกว่าตรงกันหรือไม่ ถ้ากดรหัสไม่ตรง 3 ครั้งสัญญาณจะดังเตือนที่ห้องควบคุมทำให้ทราบว่าขณะนี้กำลังมีปัญหาซึ่งผู้ใช้บริการอาจจะลืมรหัสหรือ อาจจะมีคนมาขโมยรถก็ได้ จากนั้นระบบจะหยุดทำงานทั้งหมดรอให้ผู้ดูแลไปตรวจสอบว่าเกิดจากสาเหตุใด แต่ถ้ากดรหัสถูกต้องเครื่องเช็คบัตรจะพร้อมทำงานและทำการตรวจสอบว่าเสียบบัตรเพื่อนำรถออกแล้วหรือยัง เมื่อผู้ใช้บริการนำบัตรมาเสียบโซลินอยด์ที่ติดตั้งอยู่ในเครื่องตรวจสอบบัตรจะล็อคบัตร จากนั้นมอเตอร์เกียร์จะหมุนปลดล็อครถ ผู้ใช้บริการก็สามารถนำรถออกไปได้ ก็จะจบขั้นตอนการทำงาน

โฟลว์ชาร์ทแสดงการทำงานของด้านทางเข้า




เริ่มต้น
มีรถจอด ?
NO
YES
เครื่องป้อนรหัสทำงาน
กดรหัส 3 ตัว?
NO
YES
ปลดล็อคบัตร
ดึงบัตรแล้ว ?
NO
YES
มอเตอร์หมุนล็อครถ
เริ่มจับเวลา
เครื่องคิดเวลาและเงินพร้อมทำงาน
เสียบบัตรเพื่อคิดเงิน ?
NO
YES
ล็อคบัตร
เช็คว่าเป็นบัตรของรถคันไหน
แสดงเวลาและจำนวนเงินที่ต้องจ่าย
จ่ายเงินครบแล้วหรือยัง ?
NO
YES
ปลอล็อคบัตร
ดึงบัตรออก ?
NO
YES
เครื่องตรวจเช็คหรัสทำงาน
กดรหัสเพื่อนำรถออก ?
NO
YES
เครื่องรับข้อมูลรหัสเข้าไปตรวจเช็ค
รหัสตรง ?
NO
YES
กดรหัสเป็นครั้งที่ 3 ?
NO
YES
มีสัญญานเตือนดัง
ระบบหยุดทำงานทั้งหมด
เครื่องเช็คบัตรทำงาน
เสียบบัตร ?
NO
YES
มอเตอร์หมุนปลดที่ล็อครถ

บทที่ 4
ผลการทดลอง

จากผลการทดลองในการเขียนโปรแกรมควบคุม โดยใช้โปรแกรมภาษาแอสแซมบลีของไมโครคอนโทรเลอร์ MCS-51 ควบคุมชุดอุปกรณ์การทำงาน 8 ชุด ผลที่ได้มีดังนี้

1. การควบคุมชุดอุปกรณ์เซนต์เซอร์สำหรับตรวจเช็ครถ ผลที่ได้ก็คือด้านโปรแกรมนั้นไม่มีปัญหาในส่วนใดแต่ปัญหายังมีในส่วนของอุปกรณ์คือ อุปกรณ์เซนต์เซอร์จะอาศัยหลักการทำงานโดยอาศัยแสงมาตกกระทบที่ตัวมันทำให้ได้ค่าเอาท์พุตออกมาเป็น 1 แต่เมื่อมีรถเข้ามาจอดบังแสงที่มาตกกระทบตัวมัน มันจะไม่ทำงานและให้ค่าเอาท์พุตออกมาเป็น 0 แล้วจะส่งค่าไปบออกหน่วยประมวลผลกลางว่าขณะนี้มีรถเข้ามาจอดในช่องจอดรถแล้ว ปัญหาที่พบในส่วนนี้คือเรื่องของแสงที่มาตกกระทบตัวเซนต์เซอร์ไม่เพียงพอทำให้มันไม่สามารถทำงานได้ ดังนั้นจึงต้องมีการเพิ่มแสงไฟเข้าไปในส่วนช่องของแบบจำลองอาคารจอดรถ เพื่อให้แสงที่มาตกกระทบตัวมันเพียงพอกับความต้องการของมันพร้อมทั้งปรับความไวในการรับแสงของเซนต์เซอร์ด้วย

2. การควบคุมมอเตอร์เกียร์ ผลที่ได้ก็คือด้านโปรแกรมนั้นไม่มีปัญหาในส่วนใดแต่ปัญหาอื่นที่พบจากการทดลองก็คือ ไม่สามารถขับมอเตอร์เกียร์ให้หมุนได้ตลอด เนื่องจากกระแสจากไมโครคอนโทรเลอร์ที่จ่ายให้กับอุปกรณ์ที่มาขับมอเตอร์เกียร์ไม่เพียงพอ ดังนั้นจึงได้เพิ่มไอซีบัฟเฟอร์ 74LS244 เข้าไปเพื่อเพิ่มกระแสให้พอเพียงที่จะขับมอเตอร์ได้ เมื่อเพิ่มเข้าไปแล้วปรากฏว่าใช้งานได้ดีมาก ปัญหาในส่วนนี้จึงหมดไป



3. การควบคุมชุดอุปกรณ์สำหรับใส่รหัส ผลที่ได้ก็คือด้านโปรแกรมนั้นไม่มีปัญหาในส่วนใดแต่ปัญหาอื่นที่พบจากการทดลองก็คือ อุปกรณ์ยังไม่ทำงานและได้ศึกษาเพิ่มเติมพบว่าอุปกรณ์นี้ต้องใส่ความต้านทานเพิ่มเข้าไป เพื่อให้มันสามารถทำงานได้ จึงได้ใส่ค่าความต้านทาน 10 กิโลโอห์มเพิ่มเข้าไประหว่างขาคอนโทรเลอร์กับไฟบวก 5 โวลท์ อุปกรณ์นี้จึงทำงานได้และไม่มีปัญหาในส่วนใด

4. การควบคุมอุปกรณ์สำหรับตรวจเช็คบัตร ผลที่ได้ก็คือด้านโปรแกรมนั้นไม่มีปัญหาในส่วนใดแต่ปัญหาอื่นที่พบจากการทดลองก็คือ เนื่องจากอุปกรณ์ชุดนี้อยู่ในกล่องซึ่งภายในจึงมีความมืดอุปกรณ์เซนต์เซอร์ไม่สามารถทำงานได้จึงต้องติดไฟเป็นหลอด LED เพิ่มเข้าไปเพื่อเพิ่มความสว่างให้กับข้างใน เพื่อให้เซนต์เซอร์ทำงานตามปกติ

5. การควบคุมชุดโซลินอยด์สำหรับล็อคบัตร ซึ่งหลักการทำงานของขดลวดโซลินอยด์ก็คือ เมื่อจ่ายกระแสไฟฟ้าให้กับมันจะเกิดสนามแม่เหล็กดึงแกนเหล็กขึ้น และเมื่อหยุดจ่ายกระแสไฟฟ้าแกนเหล็กจะกลับลงมายังตำแหน่งเดิม ผลที่ได้ก็คือด้านโปรแกรมนั้นไม่มีปัญหาในส่วนใดแต่ปัญหาอื่นที่พบจากการทดลองก็คือ เมื่อหยุดจ่ายกระแสไฟฟ้าให้กับโซลินอยด์ แกนเหล็กไม่กลับคืนลงมาสู่สภาวะเดิม ซึ่งสาเหตุที่ทำให้แกนเหล็กไม่สามารถกลับลงมายังตำแหน่งเดิมได้เนื่องจาก ตอนที่เราหยุดจ่ายกระแสไฟฟ้าให้กับโซลินอยด์นั้นยังมีกระแสไฟฟ้าตกค้างอยู่ที่ขดลวด จึงทำให้เกิดสนามแม่เหล็กที่ดึงขดลวดตกค้างอยู่ จากการทดลองแก้ไขโดยการนำเอาไดโอดมาต่อคร่อมที่ขั้วของขดลวดโซลินอยด์ เพื่อเป็นการกำจัดกระแสไฟฟ้าที่ยังค้างอยู่ให้หมดไป โซลินอยด์จึงทำงานตามปกติ

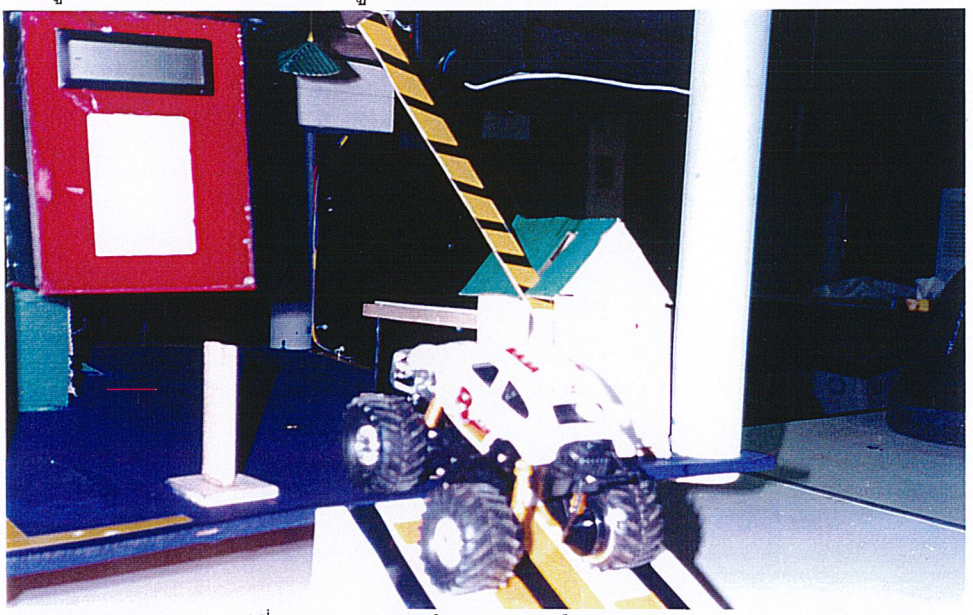
6. การควบคุมชุดอุปกรณ์สำหรับแสดงผลจำนวนเงินและเวลา ผลที่ได้ก็คือด้านโปรแกรมนั้นมีปัญหาค่อนข้างยุ่งยากมากเนื่องจากการเขียนโปรแกรมนั้นไม่สามารถทำงานควบคุมอุปกรณ์ตรงตามที่เขียนไว้ ต้องแก้ไขทีละขั้นและทดสอบหลายครั้งกว่าจะแสดงผลเวลาและจำนวนเงินที่ตั้งไว้ได้ ส่วนทางด้านอุปกรณ์ที่ใช้แสดงผลและคำนวณเวลานั้นไม่มีปัญหาในส่วนใด

7. การควบคุมชุดอุปกรณ์สำหรับหยอดเหรียญชำระเงิน จากการได้ทดสอบอุปกรณ์ชุดนี้ปรากฏว่าไม่มีปัญหาในส่วนใด ชุดลิมิตสวิตช์ที่ใช้เช็คเหรียญเมื่อเหรียญตกลงมาและโดนตัวมันทำให้มันส่งค่าไปให้กับหน่วยประมวณผลกลางสามารถทำงานได้ดี

8. ชุดอุปกรณ์สำหรับแสดงผลเมื่อกดรหัสผิด จากการได้ทดสอบอุปกรณ์ชุดนี้ปรากฏว่าไม่มีปัญหาในส่วนใด เสียงไซเรนที่ใช้ทำงานได้ดีตามปกติเมื่อมีการกดรหัสผิด

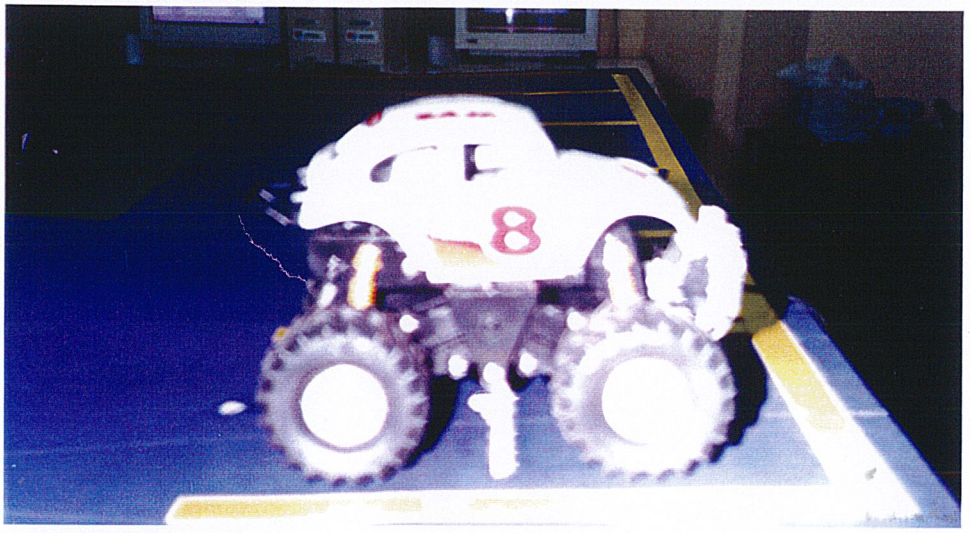
บทที่ 5
บทวิจารณ์และสรุป

จากผลการทดลองสรุปได้ว่า ในส่วนของชุดตรวจรถเข้าออกพร้อมแสดงผลเราใช้ LDR ในการตรวจเช็คจะติดตั้งบริเวณพื้น แต่จะมีปัญหาเรื่องความเม่นยำในการตรวจเช็ค เนื่องจากแสงแต่ละที่ไม่เท่ากัน จึงเพิ่มแสงเข้าไปในจุดอับต่างๆ ประกอบกับการปรับค่าความต้านทานเพื่อให้การอ่านค่าได้ถูกต้องยิ่งขึ้น จะแสดงได้ดังรูปที่ 17



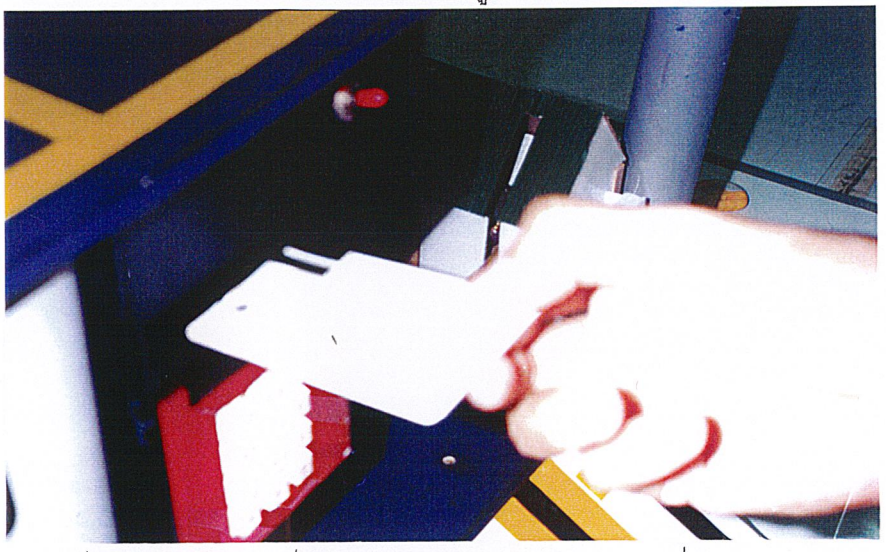
รูปที่ 17 แสดงการทำงานทางเข้าอาคารจอดรถ

แต่ในการประยุกต์ใช้งานจริงอาจใช้เป็นสวิทช์น้ำหนักแทน LDR ดังจะเห็นได้ในอาคารจอดรถบางแห่ง จะไม่มีปัญหาเรื่องแสงเข้ามาเกี่ยวข้อง ต่อมาในส่วนของตัวล็อครถจะเป็นคานยื่นออกมาตรงช่วงกลางของท้องรถ จะแสดงได้ดังรูปที่ 18



รูปที่ 18 แสดงการทำงานของคานเมื่อทำการกดรหัสและดึงบัตรแล้ว

ใช้มอเตอร์เกียร์ต่อเกลียวตรงส่วนปลายของมอเตอร์และต้องเพิ่มไอซีบัฟเฟอร์ 74LS244 เข้าไปเพื่อเพิ่มกระแสในการขับมอเตอร์เกียร์ แต่การยกขึ้นลงอาจช้าหน่อย เพราะเกลียวที่ต่อกับมอเตอร์เกียร์มีความละเอียดมาก ส่วนของกล่องที่ใช้วงจรอินฟาเรดในการอ่านค่าบัตรก็เพิ่มแสงเข้าไปด้วย เพื่อการอ่านค่าบัตรที่แม่นยำขึ้น จะแสดงได้ดังรูปที่ 19



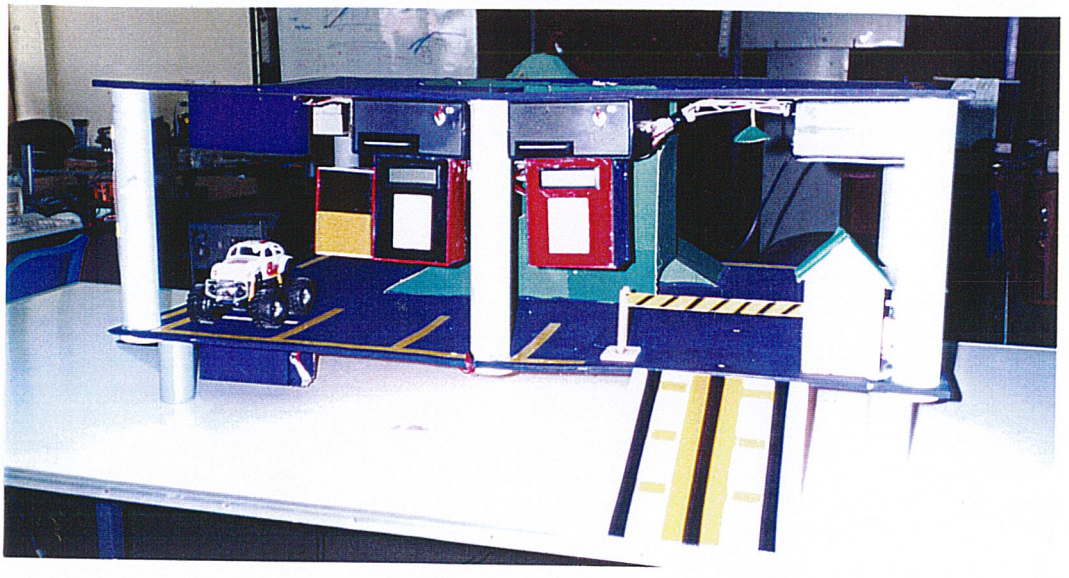
รูปที่ 19 แสดงตัวบัตรที่ใช้ในการในการจอดรถและกล่องที่ใช้อ่านค่าบัตร

อนาคตอาจพัฒนาให้มีการรีเซตการทำงานของวงจรภายในกล่องเมื่อผู้จอดรถกดรหัสผิด ส่วนตัวโซลินอยล์ที่ทำหน้าที่ล็อคบัตรต้องต่อไดโอดคร่อมที่เข้าของโซลินอยล์ เพื่อกำจัดกระแสไฟฟ้าที่ยังตกค้างอยู่ให้หมดไป ซึ่งจะทำให้เวลาหยุดจ่ายกระแสไฟฟ้าให้กับโซลินอยล์แกนเหล็กจะหล่นลงมาทันที ส่วนอุปกรณ์ใส่รหัสเราเพิ่มความต้านทาน 10 กิโลโอห์ม เข้าไปเพื่อให้มันทำงานได้ตามปกติ ส่วนอุปกรณ์แสดงผลจำนวนเงินและเวลา เป็นไปตามเป้าหมายที่โปรแกรมไว้ การชำระเงินใช้เหรียญ 5 บาทอย่างเดียวใช้ลิมิตสวิทช์ต่อเข้าที่รูหยอดเหรียญ จะแสดงได้ดังรูปที่ 20



รูปที่ 20 แสดงส่วนของการชำระค่าจอดรถ

แต่เวลาหยอดจะต้องหยอดช้าๆทีละเหรียญ ถ้าหยอดเร็วไปหรือช้าเกินไปอาจทำให้เกิดการผิดพลาดได้ ซึ่งในอนาคตอาจพัฒนาชนิดของเหรียญที่ใช้ชำระค่าจอดรถได้มากชนิดขึ้น



รูปที่ 21 แสดงภาพโดยรวมแบบจำลองอาคารจอดรถ

เนื่องจากว่าถ้าเกิดมีการกดรหัสผิดในตอนที่มาเอารถจะมีสัญญาณเตือนดังสั้นๆ แต่ถ้าหากกดผิดในครั้งที่ 3 สัญญาณเตือนจะดังยาวให้ผู้ที่ควบคุมรู้ว่าขณะนี้เกิดเหตุการณ์ที่ผิดปกติแล้ว ซึ่งในอนาคตอาจเปลี่ยนการกดรหัสมาเป็นการสแกนลายนิ้วมือหรือม่านตาก็เป็นได้ สำหรับในโครงงานได้ทำการจำลองระบบจอดรถจำนวน 2 คัน ดังนั้นกล่องที่ใช้วงจรอินฟาเรดในการอ่านค่าบัตรจึงมี 2 กล่อง และจะมีกล่องเช็คเวลาและค่าจอดรถด้วยอีก 1 กล่อง รวมเป็น 3 กล่อง ซึ่งในการเช็คเวลาและค่าจอด ณ คันใดคันหนึ่งจะต้องมีการต่อสายเข้ากับโซลิลอย์ที่กล่องเช็คเวลาและค่าจอดที่เวลาทำการเช็ค จะต่อกล่องวงจรอินฟาเรดทั้ง 2 กล่อง เข้ากับโซลินอยล์ที่กล่องเช็คเวลาและค่าจอดพร้อมกันไม่ได้ เพราะจะทำให้การทำงานผิดพลาดได้ในที่นี้ใช้เป็นสวิทช์ โครงงานนี้เป็นการประยุกต์นำเอาวงจรของโครงงานในเทอมที่แล้วมาใช้งานเป็นวงจรที่ใช้ในการอ่านค่าความแตกต่างระหว่างสีขาวและสีดำ โดยใช้หลักการสะท้อนของแสงอินฟาเรดโดยสีขาวและสีดำมีการดูดกลืนแสงไม่เท่ากันจะเป็นการประหยัดค่าใช้จ่ายได้มาก ในการประยุกต์ใช้งานจริงถ้ามีจำนวนรถที่เข้าจอดมากก็ทำการเพิ่มจำนวนแถบได้ตามความเหมาะสม โครงงานนี้สามารถนำไปประยุกต์ใช้งานได้จริง และทำการค้นคว้าพัฒนาต่อไป

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ดีจากการช่วยเหลือจากอาจารย์และบุคคลหลายท่าน

ขอขอบคุณ รศ.ดร. จงกล งามวิวิทย์ ที่ได้ให้คำปรึกษาชี้แจงแนวทางในการแก้ไขปัญหาต่างๆ อย่างทุ่มเทตลอดจนสถานที่ใช้ในโครงงานและปริญญานิพนธ์ และความเอาใจใส่อย่างดีเสมอมา

ขอขอบคุณภาควิชาวิศวกรรมระบบควบคุม อาจารย์ และเจ้าหน้าที่ทุกท่านที่อำนวยความสะดวกเอื้อเฟื้ออุปกรณ์และให้คำปรึกษาที่ดีเสมอมา

ขอขอบคุณเพื่อนๆ ทุกคน ที่ให้ความช่วยเหลือด้านอุปกรณ์ ลายวงจรต่างๆ และให้กำลังใจตลอดการทำงาน

ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับประสบการณ์ความรู้ ต่างๆ ตลอดระยะเวลา 3 ปี

ขอขอบคุณ พ่อ แม่ และครอบครัวของพวกเรา ที่คอยห่วงใยและให้กำลังใจตลอดจนความช่วยเหลือในทุกๆ เรื่องเสมอมาโดยไม่เปลี่ยนแปลง

นาย บัณฑิต วิเชียร

นาย ปกรณ์ ดาราฉาย

หนังสืออ้างอิง

1. Ayala,K.J., "The 8051 Microcontroller Architecture, Programming and Applications," West Publishing Company, 1991.
2. Intel Corporation, "Microcontroller Handbook," Intel Corporation, 1992.
3. MacKenzie,I.Scott., "The 8051 Microcontroller," Macmillan Publishing Company, 1992

ภาคผนวก

PROGRAM

```
MULTIPLE        EQU     10000
OUTPUT          EQU     20H
FLAG            EQU     21H             ;INSERT  KEYWORD
SS              EQU     49H
KEY             EQU     4CH
OVER            EQU     FLAG.2
BELL_ON         EQU     FLAG.4

                MOV     P0,#0FFH
                MOV     P1,#0
                MOV     P2,#0
                MOV     P3,#0

KPAD_ROW0       BIT     P0.0
KPAD_ROW1       BIT     P0.1
KPAD_ROW2       BIT     P0.2
KPAD_ROW3       BIT     P0.3
KPAD_COL2       BIT     P0.4
KPAD_COL1       BIT     P0.5
KPAD_COL0       BIT     P0.6
KPAD_DATA       EQU     4DH
SDA             BIT     P3.7;P1.0       ; SDA I2C Bus
SCL             BIT     P3.6;P1.1       ; SCL I2C Bus
FLAG1           EQU     02FH            ; User FLAG
I2C_ACK         BIT     FLAG1.0         ; Define I2C Acknowledge
as bit

TIME            EQU     031H
MONEY           EQU     032H
I2C_ADDR        EQU     033H            ; For keep I2C Address
I2C_DATA        EQU     034H            ; For keep I2C Data
SECONDS         EQU     043H            ; For keep Seconds
MINUTES         EQU     049H            ; For keep Minutes
HOURS           EQU     04AH            ; For keep Hours
CONTROL         EQU     047H            ; For keep Control Byte
LCD_DATA        EQU     048H
RTC_ID          EQU     11010000B       ; RTC Slave Address
RS              EQU     P3.0
RW              EQU     P3.1
EN              EQU     P3.2
DATA            EQU     P1

;*********************************************************************
;WAIT THE CAR
;*********************************************************************
                SETB    SDA
                SETB    SCL
                LCALL   WAIT_LCD
                LCALL   INIT_LCD
                LCALL   CLEAR_LCD
                LCALL   DELAY_10MS
MAIN:           CLR     P2.1            ;LOCK  CARD
LOOP:           JNB     P2.0,LOOP       ;CHECK THE CAR
;*********************************************************************
;INPUT KEYWORD
;*********************************************************************
```

```
KEYWORD:        LCALL   SCAN            ;INPUT KEYWORD
                MOV     40H,KPAD_DATA
                LCALL   SCAN
                MOV     41H,KPAD_DATA
                LCALL   SCAN
                MOV     42H,KPAD_DATA
                LCALL   DELAY_1MS
;*********************************************************************
                SETB    P2.1            ;CLEAR CARD
LOOP1:          JB      P2.2,LOOP1      ;CHECK OUT OF CARD
                SETB    P2.5            ;MOTOR GEAR TURN RIGTH
WAIT:           JNB     P2.6,WAIT       ;CHECK  LIMIT SWITCH
                CLR     P2.5
                LCALL   RTC_WR

                MOV     A,#0
                SETB    P2.4            ;SET FOR FREE CARD
LOOP2:          JNB     P2.3,LOOP2      ;CHECK  CARD 2
                LCALL   DELAY_1S        ;P2.3 CHECK CARD IN
                CLR     P2.4            ;LOCK CARD

                LCALL   RTC_RD
                LCALL   DELAY_10MS
                LCALL   WAIT_LCD
                LCALL   INIT_LCD
                LCALL   CLEAR_LCD
               ;LCALL   LCD_OFF
;*********************************************************************
;WRITE  LCD
;*********************************************************************
                SETB    EN
                CLR     RS
                MOV     DATA,#00H
                CLR     EN
                LCALL   WAIT_LCD

                MOV     A,#'T'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'I'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'M'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'E'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#' '
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK
                MOV     LCD_DATA,MINUTES
                LCALL   BCD2LCD         ;SHOW THE TIME IS MINUITS
```

```
                MOV     A,#':'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK
                MOV     LCD_DATA,SECONDS
                LCALL   BCD2LCD         ;SHOW THE TIME IS
SECOUNDS

                MOV     A,#' '
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'H'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'O'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'U'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'R'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'S'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK
;*********************************************************************
;WRITE LCD  ROW TWO
;*********************************************************************
                SETB    EN
                CLR     RS
                MOV     DATA,#0C0H
                CLR     EN
                LCALL   WAIT_LCD

                MOV     A,#'A'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'M'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'O'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'U'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'N'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK
```

```
                MOV     A,#'T'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#' '
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK
;*****************************************************************
                MOV     A,MINUTES
                LCALL   DISHEX
J:              LJMP    AAA
;*****************************************************************
DISHEX:
                CJNE    A,#00H,TE_1
                MOV     A,SECONDS
                LCALL   DISHEX0
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#' '
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#1
                LCALL   DELAY_10MS
        RET
TE_1:
                CJNE    A,#01H,TE_2
                MOV     A,SECONDS
                LCALL   DISHEX1
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#4
                LCALL   DELAY_10MS
        RET
TE_2:           CJNE    A,#02H,TE_3
                MOV     A,SECONDS
                LCALL   DISHEX2
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'4'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#8
                LCALL   DELAY_10MS
        RET

TE_3:           CJNE    A,#03H,TE_4
                MOV     A,SECONDS
                LCALL   DISHEX3
```

```
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'6'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#12
                LCALL   DELAY_10MS
        RET

TE_4:           CJNE    A,#04H,TE_5
                MOV     A,SECONDS
                LCALL   DISHEX4
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'8'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#16
                LCALL   DELAY_10MS
        RET

TE_5:           CJNE    A,#05H,TE_6
                MOV     A,SECONDS
                LCALL   DISHEX5
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#20
                LCALL   DELAY_10MS
        RET

TE_6:           CJNE    A,#06H,TE_7
                MOV     A,SECONDS
                LCALL   DISHEX6
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#24
                LCALL   DELAY_10MS
        RET

TE_7:           CJNE    A,#07H,TE_8
                MOV     A,SECONDS
                LCALL   DISHEX7
                MOV     A,#'1'
                LCALL   TEST
```

```
                MOV     A,#'4'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#28
                LCALL   DELAY_10MS
        RET

TE_8:           CJNE    A,#08H,TE_9
                MOV     A,SECONDS
                LCALL   DISHEX8
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'6'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#32
                LCALL   DELAY_10MS

TE_9:           CJNE    A,#09H,TE_10
                MOV     A,SECONDS
                LCALL   DISHEX9
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'8'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#36
                LCALL   DELAY_10MS
        RET

TE_10:          CJNE    A,#10H,TE_11
                MOV     A,SECONDS
                LCALL   DISHEX10
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                LCALL   DELAY_10MS
                MOV     R1,#40
                LCALL   DELAY_10MS
        RET

TE_11:  RET
TEST:           LCALL   DELAY_10MS
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK
        RET
;*****************************************************************
DISHEX0:
```

```
                CJNE    A,#00H,T_1
        RET

T_1:            CJNE    A,#15H,T_2
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#' '
                LCALL   TEST
                MOV     R1,#1
                LJMP    J

T_2:            CJNE    A,#30H,T_3
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#2
                LJMP    J

T_3:            CJNE    A,#45H,T_4
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#3
                LJMP    J

T_4:            MOV     A,#15H
T_4_1:          DJNZ    SECONDS,T_4_2
        RET

T_4_2:          CJNE    A,SECONDS,T_5
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#' '
                LCALL   TEST
                MOV     R1,#1
                LJMP    J

T_5:            MOV     A,#30H
                CJNE    A,SECONDS,T_6
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#2
                LJMP    J
```

```
T_6:            MOV     A,#45H
                CJNE    A,SECONDS,T_7
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#3
                LJMP    J

T_7:            LJMP    T_4

;*********************************************************************
DISHEX1:
                CJNE    A,#00H,T_8
        RET

T_8:            CJNE    A,#15H,T_9
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#5
                LJMP    J

T_9:            CJNE    A,#30H,T_10
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#6
                LJMP    J

T_10:           CJNE    A,#45H,T_11
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#7
                LJMP    J

T_11:           MOV     A,#15H
T_11_1:         DJNZ    SECONDS,T_11_2
        RET

T_11_2:         CJNE    A,SECONDS,T_12
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'5'
```

```
                LCALL   TEST
                MOV     R1,#5
                LJMP    J

T_12:           MOV     A,#30H
                CJNE    A,SECONDS,T_13
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#6
                LJMP    J

T_13:           MOV     A,#45H
                CJNE    A,SECONDS,T_14
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#7
                LJMP    J

T_14:           LJMP    T_11
;*****************************************************************
DISHEX2:
                CJNE    A,#00H,T_15
        RET

T_15:           CJNE    A,#15H,T_16
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'4'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#9
                LJMP    J

T_16:           CJNE    A,#30H,T_17
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#10
                LJMP    J

T_17:           CJNE    A,#45H,T_19
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#'5'
```

```
                LCALL   TEST
                MOV     R1,#11
                LJMP    J

T_19:           MOV     A,#15H
T_19_1:         DJNZ    SECONDS,T_19_2
        RET

T_19_2:         CJNE    A,SECONDS,T_20
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'4'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#9
                LJMP    J

T_20:           MOV     A,#30H
                CJNE    A,SECONDS,T_21
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#10
                LJMP    J

T_21:           MOV     A,#45H
                CJNE    A,SECONDS,T_22
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#11
                LJMP    J

T_22:           LJMP    T_19
;*****************************************************************
DISHEX3:
                CJNE    A,#00H,T_23
        RET

T_23:           CJNE    A,#15H,T_24
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'6'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#13
                LJMP    J

T_24:           CJNE    A,#30H,T_25
                MOV     A,#' '
```

```
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#14
                LJMP    J

T_25:           CJNE    A,#45H,T_26
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#15
                LJMP    J

T_26:           MOV     A,#15H
T_26_1:         DJNZ    SECONDS,T_26_2
        RET

T_26_2:         CJNE    A,SECONDS,T_27
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'6'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#13
                LJMP    J

T_27:           MOV     A,#30H
                CJNE    A,SECONDS,T_28
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#14
                LJMP    J

T_28:           MOV     A,#45H
                CJNE    A,SECONDS,T_29
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#15
                LJMP    J

T_29:           LJMP    T_26
;*****************************************************************
DISHEX4:
                CJNE    A,#00H,T_30
```

```
        RET

T_30:           CJNE    A,#15H,T_31
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'8'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#17
                LJMP    J

T_31:           CJNE    A,#30H,T_32
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#18
                LJMP    J

T_32:           CJNE    A,#45H,T_33
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#19
                LJMP    J

T_33:           MOV     A,#15H
T_33_1:         DJNZ    SECONDS,T_33_2
        RET

T_33_2:         CJNE    A,SECONDS,T_34
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'8'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#17
                LJMP    J

T_34:           MOV     A,#30H
                CJNE    A,SECONDS,T_35
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#18
                LJMP    J

T_35:           MOV     A,#45H
```

```
                CJNE    A,SECONDS,T_36
                MOV     A,#' '
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#19
                LJMP    J

T_36:           LJMP    T_33
;*****************************************************************
DISHEX5:
                CJNE    A,#00H,T_37
        RET

T_37:           CJNE    A,#15H,T_38
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#21
                LJMP    J

T_38:           CJNE    A,#30H,T_39
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#22
                LJMP    J

T_39:           CJNE    A,#45H,T_40
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#23
                LJMP    J

T_40:           MOV     A,#15H
T_40_1:         DJNZ    SECONDS,T_40_2
        RET

T_40_2:         CJNE    A,SECONDS,T_41
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#21
```

```
                LJMP    J

T_41:           MOV     A,#30H
                CJNE    A,SECONDS,T_42
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#22
                LJMP J

T_42:           MOV     A,#45H
                CJNE    A,SECONDS,T_43
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#23
                LJMP    J

T_43:           LJMP    T_40
;*****************************************************************
DISHEX6:
                CJNE    A,#00H,T_44
        RET

T_44:           CJNE    A,#15H,T_45
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#25
                LJMP    J

T_45:           CJNE    A,#30H,T_46
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#26
                LJMP    J

T_46:           CJNE    A,#45H,T_47
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#27
```

```
                LJMP    J

T_47:           MOV     A,#15H
T_47_1:         DJNZ    SECONDS,T_47_2
        RET

T_47_2:         CJNE    A,SECONDS,T_48
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#25
                LJMP    J

T_48:           MOV     A,#30H
                CJNE    A,SECONDS,T_49
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#26
                LJMP J

T_49:           MOV     A,#45H
                CJNE    A,SECONDS,T_50
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'3'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#27
                LJMP    J

T_50:           LJMP    T_47
;*****************************************************************
DISHEX7:
                CJNE    A,#00H,T_51
        RET

T_51:           CJNE    A,#15H,T_52
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'4'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#29
                LJMP    J

T_52:           CJNE    A,#30H,T_53
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
```

```
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#30
                LJMP    J

T_53:           CJNE    A,#45H,T_54
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#31
                LJMP    J

T_54:           MOV     A,#15H
T_54_1:         DJNZ    SECONDS,T_54_2
        RET

T_54_2:         CJNE    A,SECONDS,T_55
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'4'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#29
                LJMP    J

T_55:           MOV     A,#30H
                CJNE    A,SECONDS,T_56
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#30
                LJMP    J

T_56:           MOV     A,#45H
                CJNE    A,SECONDS,T_57
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#31
                LJMP    J

T_57:           LJMP    T_54
;*****************************************************************
DISHEX8:
                CJNE    A,#00H,T_58
        RET
```

```
T_58:           CJNE    A,#15H,T_59
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'6'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#33
                LJMP    J

T_59:           CJNE    A,#30H,T_60
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#34
                LJMP    J

T_60:           CJNE    A,#45H,T_61
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#35
                LJMP    J

T_61:           MOV     A,#15H
T_61_1:         DJNZ    SECONDS,T_61_2
        RET

T_61_2:         CJNE    A,SECONDS,T_62
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'6'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#33
                LJMP    J

T_62:           MOV     A,#30H
                CJNE    A,SECONDS,T_63
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#34
                LJMP    J

T_63:           MOV     A,#45H
                CJNE    A,SECONDS,T_64
                MOV     A,#'1'
```

```
                LCALL   TEST
                MOV     A,#'7'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#35
                LJMP    J

T_64:           LJMP    T_61
;*****************************************************************
DISHEX9:
                CJNE    A,#00H,T_65
        RET

T_65:           CJNE    A,#15H,T_66
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'8'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#37
                LJMP    J

T_66:           CJNE    A,#30H,T_67
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#38
                LJMP    J

T_67:           CJNE    A,#45H,T_68
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#39
                LJMP    J

T_68:           MOV     A,#15H
T_68_1:         DJNZ    SECONDS,T_68_2
        RET

T_68_2:         CJNE    A,SECONDS,T_69
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'8'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#37
                LJMP    J
```

```
T_69:           MOV     A,#30H
                CJNE    A,SECONDS,T_70
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#38
                LJMP    J

T_70:           MOV     A,#45H
                CJNE    A,SECONDS,T_71
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'9'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#39
                LJMP    J

T_71:           LJMP    T_68
;*****************************************************************
DISHEX10:
                CJNE    A,#00H,T_72
        RET

T_72:           CJNE    A,#15H,T_73
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#41
                LJMP    J

T_73:           CJNE    A,#30H,T_74
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#42
                LJMP    J

T_74:           CJNE    A,#45H,T_75
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#43
                LJMP    J
```

```
T_75:           MOV     A,#15H
T_75_1:         DJNZ    SECONDS,T_75_2
        RET

T_75_2:         CJNE    A,SECONDS,T_76
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#41
                LJMP    J

T_76:           MOV     A,#30H
                CJNE    A,SECONDS,T_77
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'0'
                LCALL   TEST
                MOV     R1,#42
                LJMP    J

T_77:           MOV     A,#45H
                CJNE    A,SECONDS,T_78
                MOV     A,#'2'
                LCALL   TEST
                MOV     A,#'1'
                LCALL   TEST
                MOV     A,#'5'
                LCALL   TEST
                MOV     R1,#43
                LJMP    J

T_78:           LJMP    T_75
;*****************************************************************

AAA:            MOV     A,#' '
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'B'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'A'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'H'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK

                MOV     A,#'T'
                LCALL   WRITE_TEXT
                LCALL   LCD_BLINK
```

```
                MOV         A,#' '
                LCALL       WRITE_TEXT
                LCALL       LCD_BLINK

;*********************************************************************
;CHECK  MONEY
;*********************************************************************
CHECK:          JNB         P2.2,CHECK
                LCALL       DELAY_10MS
CHECK1:         JB          P2.2,CHECK1
                LCALL       DELAY_10MS
                DJNZ        R1,CHECK
;*********************************************************************
                LCALL       WAIT_LCD
                LCALL       INIT_LCD
                LCALL       CLEAR_LCD
                LCALL       DELAY_1S
                SETB        P2.4            ;CLEAR CARD
WAIT4:          JB          P2.3,WAIT4      ;CHECK OUT OF CARD
                CLR         P2.4
                MOV         R2,#3           ;CHECK IF INPUT KEYWORD
OVER THREE TIMES

                LCALL       CONFIRM
LOOP3:          JNB         P2.2,LOOP3      ;FINAL OF CHECK CARD
                SETB        P2.7            ;MOTOR GEAR TURN LEFT
                LCALL       DELAY_1S
                SETB        P3.5
                LCALL       DELAY1
WAIT3:          JNB         P3.3,WAIT3      ;CHECK LIMIT SWITCH
                CLR         P3.5
                LCALL       DELAY_1S
                CLR         P2.7
                LJMP        MAIN
;*********************************************************************
SCAN:           MOV         P0,#0FFH
                MOV         KPAD_DATA,#0

CHK_COL0:       CLR         KPAD_COL0
                MOV         A,P0
                ANL         A,#00FH
                CJNE        A,#00FH,COL0_DETECT
                AJMP        CHK_COL1
COL0_DETECT:    MOV         KPAD_DATA,#01
                AJMP        GET_ROW

CHK_COL1:       SETB        KPAD_COL0
                CLR         KPAD_COL1
                MOV         A,P0
                ANL         A,#00FH
                CJNE        A,#00FH,COL1_DETECT
                AJMP        CHK_COL2
COL1_DETECT:    MOV         KPAD_DATA,#02
                AJMP        GET_ROW

CHK_COL2:       SETB        KPAD_COL1
                CLR         KPAD_COL2
```

```
                    MOV     A,P0
                    ANL     A,#00FH
                    CJNE    A,#00FH,COL2_DETECT
                    SJMP    SCAN

COL2_DETECT:        MOV     KPAD_DATA,#03
                    AJMP    GET_ROW

GET_ROW:            CLR     KPAD_COL0
                    CLR     KPAD_COL1
                    CLR     KPAD_COL2

                    JB      KPAD_ROW0,CHK_ROW1
W0:                 MOV     P0,#00FH
                    CLR     P0.4
                    CLR     P0.5
                    CLR     P0.6
                    MOV     A,P0
                    ANL     A,#00FH
                    CJNE    A,#00FH,W0
                    RET

CHK_ROW1:           JB      KPAD_ROW1,CHK_ROW2
                    MOV     A,KPAD_DATA
                    ADD     A,#3
                    MOV     KPAD_DATA,A
W2:                 MOV     P0,#00FH
                    CLR     P0.4
                    CLR     P0.5
                    CLR     P0.6
                    MOV     A,P0
                    ANL     A,#00FH
                    CJNE    A,#00FH,W2
          RET

CHK_ROW2:           JB      KPAD_ROW2,CHK_ROW3
                    MOV     A,KPAD_DATA
                    ADD     A,#6
                    MOV     KPAD_DATA,A
W3:                 MOV     P0,#00FH
                    CLR     P0.4
                    CLR     P0.5
                    CLR     P0.6
                    MOV     A,P0
                    ANL     A,#00FH
                    CJNE    A,#00FH,W3
          RET

CHK_ROW3:           MOV     A,KPAD_DATA
                    ADD     A,#9
                    MOV     KPAD_DATA,A
W4:                 MOV     P0,#00FH
                    CLR     P0.4
                    CLR     P0.5
                    CLR     P0.6
                    MOV     A,P0
                    ANL     A,#00FH
                    CJNE    A,#00FH,W4
```

```
        RET
;*********************************************************************
CONFIRM:        MOV     KPAD_DATA,#00H
                LCALL   SCAN
                MOV     44H,KPAD_DATA
                LCALL   SCAN
                MOV     45H,KPAD_DATA
                LCALL   SCAN
                MOV     46H,KPAD_DATA
                CLR     C
                MOV     A,44H
                SUBB    A,40H
                JNZ     ERROR
                CLR     C
                MOV     A,45H
                SUBB    A,41H
                JNZ     ERROR
                CLR     C
                MOV     A,46H
                SUBB    A,42H
                JNZ     ERROR
                CLR     C
        RET
;*********************************************************************
********
ERROR:          MOV     R3,#4           ;DELAY 3_SEC.
ERROR1:         SETB    P3.4            ;SHOW ALARM WHEN PASSWORD
IS WRONG
                LCALL   DELAY1
                DJNZ    R3,ERROR1
                CLR     P3.4
                DJNZ    R2,CONFIRM
                LJMP    MISSTAKE_PASSWORD
;*********************************************************************
********
MISSTAKE_PASSWORD:SETB  P3.4            ;SHOW LED WHEN PASS WORD
IS MISSTAKE
                LJMP    DEND
;*********************************************************************
********
DELAY_1MS:      MOV     6,#0E6H
DELAY_1MS_1:    NOP
                NOP
                DJNZ    R6,DELAY_1MS_1
DELAY_10MS:     MOV     7,#010H
DELAY_10MS_1:   MOV     6,#0E6H
DELAY_10MS_2:   NOP
                NOP
                DJNZ    R6,DELAY_10MS_2
                DJNZ    R7,DELAY_10MS_1
                RET
DELAY_1S:       MOV     5,#100
DELAY_1S_1:     ACALL   DELAY_10MS
                DJNZ    R5,DELAY_1S_1
                RET
;*********************************************************************
*********
```

```
DELAY1:         MOV     R4,#MULTIPLE
AGAIN:          MOV     TH0,#-100
                SETB    TCON.4      ;TR0
WAIT1:          JNB     TCON.5,WAIT1
                CLR     TCON.5      ;TF0
                CLR     TCON.4
                DJNZ    R4,AGAIN
          RET
;*********************************************************************
;SHOW LCD
;*********************************************************************
WAIT_LCD:

                SETB    EN              ;START LCD COMMAND
                CLR     RS              ;IT'S A COMMAND
                SETB    RW              ;IT'S A READ COMMAND
                MOV     DATA,#0FFh      ;SET ALL PINS TO FF
INNITIALLY
                MOV     A,DATA          ;READ THE RETURN VALUE
                JB      ACC.7,WAIT_LCD ;IF BIT 7 HIGH LCD STILL
BUSSY
                CLR     EN              ;FINISH THE COMMAND
                CLR     RW              ;TURN OFF RW FIR FUTURE
COMMAND
        RET

INIT_LCD:

                SETB    EN
                CLR     RS
                MOV     DATA,#38h
                CLR     EN
                LCALL   WAIT_LCD
                SETB    EN
                CLR     RS
                MOV     DATA,#0Eh
                CLR     EN
                LCALL   WAIT_LCD
                SETB    EN
                CLR     RS
                MOV     DATA,#06h
                CLR     EN
                LCALL   WAIT_LCD
        RET

CLEAR_LCD:

                SETB    EN
                CLR     RS
                MOV     DATA,#01h
                CLR     EN
                LCALL   WAIT_LCD
        RET

WRITE_TEXT:

                SETB    EN
                SETB    RS
```

```
                MOV         DATA,A
                CLR         EN
                LCALL       WAIT_LCD
        RET

LCD_BLINK:      SETB        EN
                CLR         RS
                MOV         DATA,#00001111B
                CLR         EN
                ACALL       WAIT_LCD
        RET

LCD_OFF:        SETB        EN
                CLR         RS
                MOV         DATA,#00001000B
                CLR         EN
                ACALL       WAIT_LCD
        RET

;*********************************************************************
; I2C RTC Read
;*********************************************************************
RTC_RD:         MOV         I2C_ADDR,#RTC_ID;Set RTC as I2C Write
Slav
                LCALL       I2C_SLAVE       ; Connect Slave

                MOV         I2C_DATA,#000H ; Set Slave Address 00H
                LCALL       I2C_DATA_WR     ; Write Data to Slave

                MOV         I2C_ADDR,#RTC_ID+1 ; Set RTC as I2C Read
Slave
                LCALL       I2C_SLAVE       ; Connect Slave
                LCALL       I2C_DATA_RD     ; Read Data from Slave

                MOV         SECONDS,I2C_DATA ; Read Data to SECONDS
                LCALL       I2C_ACK_BIT     ; Send Acknowledge

                LCALL       I2C_DATA_RD     ; Read Data from Slave
                MOV         MINUTES,I2C_DATA ; Read Data to MINUTES
                LCALL       I2C_ACK_BIT     ; Send Acknowledge

                LCALL       I2C_DATA_RD     ; Read Data from Slave
                MOV         HOURS,I2C_DATA ; Read Data to HOURS
                LCALL       I2C_ACK_BIT     ; Send Acknowledge


                LCALL       I2C_DATA_RD     ; Read Data from Slave
                MOV         CONTROL,I2C_DATA ; Read Data to CONTROL
                LCALL       I2C_NACK_BIT    ; Send Not Acknowledge

                LCALL       I2C_STOP        ; Send Stop Condition
        RET                                 ; Return

;*********************************************************************
; I2C RTC Write
;*********************************************************************
RTC_WR:         MOV         I2C_ADDR,#RTC_ID ; Set RTC as I2C Write
Slave
```

```
                LCALL       I2C_SLAVE       ; Connect Slave

                MOV         I2C_DATA,#000H ; Set Slave Address 00H
                LCALL       I2C_DATA_WR     ; Write Data to Slave

                MOV         I2C_DATA,#0;SECONDS ; Write SECONDS to
RTC
                LCALL       I2C_DATA_WR
                MOV         I2C_DATA,#0;MINUTES ; Write MINUTES to
RTC
                LCALL       I2C_DATA_WR

                MOV         I2C_DATA,#0;HOURS ; Write HOURS to RTC
                LCALL       I2C_DATA_WR
                MOV         I2C_DATA,#5;DAY   ; Write DAY to RTC
                LCALL       I2C_DATA_WR

                MOV         I2C_DATA,#6;DATE  ; Write DATE to RTC
                LCALL       I2C_DATA_WR

                MOV         I2C_DATA,#7;MONTH ; Write MONTH to RTC
                LCALL       I2C_DATA_WR
                MOV         I2C_DATA,#1;YEAR  ; Write YEAR to RTC
                LCALL       I2C_DATA_WR

                MOV         I2C_DATA,CONTROL  ; Write CONTROL to
RTC
                LCALL       I2C_DATA_WR

                LCALL       I2C_STOP          ; Send Stop Condition
                                              ; Return
        RET

;*********************************************************************
; I2C Data Write
; I/P:          I2C_DATA
; Reserve:  R5
;*********************************************************************
I2C_DATA_WR:    PUSH        ACC               ; Push ACC.
                SETB        I2C_ACK           ; Set ACK. bit
                MOV         A,I2C_DATA        ; Get Data
                MOV         R5,#008           ; Set loop 8 times
I2C_DATA_WR_1:  RLC         A                 ; Rotate ACC. to Left
with Carry
                MOV         SDA,C             ; Move Carry Flag to
SDA
                LCALL       I2C_CLK           ; Pulse I2C Clock
                DJNZ        R5,I2C_DATA_WR_1  ; Do until 8 times
                SETB        SDA               ; Set SDA
                LCALL       I2C_DELAY         ; Delay
                SETB        SCL               ; Set SCL
                LCALL       I2C_DELAY         ; Delay
                JB          SDA,I2C_DATA_WR_2 ; Check Acknowledge
from Slave
                CLR         I2C_ACK           ; Clear ACK. bit
I2C_DATA_WR_2:  CLR         SCL               ; Clear SCL
                POP         ACC               ; Pop ACC.
                                              ; Return
        RET
```

```
;*********************************************************************
; I2C Data Read
; O/P:              I2C_DATA
; Reserve:  R5
;*********************************************************************
I2C_DATA_RD:        PUSH        ACC                 ; Push ACC.
                    CLR         A                   ; Clear ACC.
                    MOV         R5,#008             ; Set loop 8 times
I2C_DATA_RD_1:      ACALL       I2C_DELAY           ; Delay
                    SETB        SCL                 ; Set SCL
                    ACALL       I2C_DELAY           ; Delay
                    MOV         C,SDA               ; Get SDA to Carry
Flag
                    RLC         A                   ; Rotate ACC. to Left
with Carry
                    CLR         SCL                 ; Clear SCL
                    DJNZ        R5,I2C_DATA_RD_1; Do until 8 times
                    MOV         I2C_DATA,A          ; Move Data to
I2C_DATA
                    POP         ACC                 ; Pop ACC.
        RET                                         ; Return

;*********************************************************************
; I2C Slave Connect
; I/P:              I2C_ADDR
; O/P Flag: I2C_ACK
; Reserve:  R5
;*********************************************************************
I2C_SLAVE:          PUSH        ACC                 ; Push ACC.
                    SETB        I2C_ACK             ; Set ACK. bit
                    MOV         A,I2C_ADDR          ; Get Slave Address
                    LCALL       I2C_START           ; Send Start
Condition
                    MOV         R5,#008             ; Set loop 8 times
I2C_SLAVE_1:        RLC         A                   ; Rotate ACC. to Left
with Carry
                    MOV         SDA,C               ; Move Carry Flag to
SDA
                    ACALL       I2C_CLK             ; Pulse I2C Clock
                    DJNZ        R5,I2C_SLAVE_1      ; Do until 8 times

                    SETB        SDA                 ; Set SDA
                    ACALL       I2C_DELAY           ; Delay
                    SETB        SCL                 ; Set SCL
                    ACALL       I2C_DELAY           ; Delay
                    JB          SDA,I2C_SLAVE_2     ; Check Acknowledge
from Slave
                    CLR         I2C_ACK             ; Clear ACK.
I2C_SLAVE_2:        CLR         SCL                 ; Clear SCL
                    POP         ACC                 ; Pop ACC.
            RET                                     ; Return

;*********************************************************************
; I2C Start Condition
;*********************************************************************
I2C_START:          JNB         SCL,I2C_START_1     ; Check current SCL
set?
                    CLR         SCL                 ; Clear SCL
```

```
I2C_START_1:    SETB    SDA             ; Set SDA
                SETB    SCL             ; Set SCL
                LCALL   I2C_DELAY       ; Delay
                CLR     SDA             ; Clear SDA during
SCL set
                LCALL   I2C_DELAY       ; Delay
                CLR     SCL             ; Clear SCL
        RET                             ; Return

;*********************************************************************
; I2C Stop Condition
;*********************************************************************
I2C_STOP:       JNB     SCL,I2C_STOP_1  ; Check current SCL
set?
                CLR     SCL             ; Clear SCL

I2C_STOP_1:     CLR     SDA             ; Clear SDA
                ACALL   I2C_DELAY       ; Delay
                SETB    SCL                 ; Set SCL
                ACALL   I2C_DELAY       ; Delay
                SETB    SDA             ; Set SDA during SCL
set
        RET                             ; Return

;*********************************************************************
; I2C Clock
;*********************************************************************
I2C_CLK:        ACALL   I2C_DELAY       ; Pulse SCL
                SETB    SCL
                ACALL   I2C_DELAY
                CLR     SCL
        RET                             ; Return

;*********************************************************************
; I2C Acknowledge
;*********************************************************************
I2C_ACK_BIT:    CLR     SDA             ; Clear SDA
                ACALL   I2C_DELAY       ; Delay
                ACALL   I2C_CLK         ; Pulse I2C Clock
                SETB    SDA
        RET                             ; Return

;*********************************************************************
; I2C Not Acknowledge
;*********************************************************************
I2C_NACK_BIT:   SETB    SDA             ; Set SDA
                ACALL   I2C_DELAY       ; Delay
                ACALL   I2C_CLK         ; Pulse I2C Clock
                SETB    SCL
        RET                             ; Return
BCD2LCD:        PUSH    ACC             ; Push ACC. To Stack
                PUSH    B               ; Push B to Stack
                MOV     A,LCD_DATA      ; Get input data
value
                MOV     B,A             ; Copy to B
                ANL     A,#11110000B    ; Get higher 4 bit
                SWAP    A               ; Swap nibble
```

```
                ADD         A,#030H             ; Convert to ASCII
                LCALL       WRITE_TEXT
                LCALL       LCD_BLINK
                MOV         A,B                 ; Restore value
                ANL         A,#00001111B        ; Get lower 4 bit
                ADD         A,#030H             ; Convert to ASCII
                LCALL       WRITE_TEXT
                LCALL       LCD_BLINK
                POP         B                   ; Pop B from Stack
                POP         ACC                 ; Pop ACC. from Stack
        RET                                     ; Return

;*********************************************************************
; Dummy Delay time I2C_DELAY, LCD_DELAY, 10m, 100m, 1s
;*********************************************************************
I2C_DELAY:      MOV         R6,#00CH            ; Each loop = 50 us
I2C_DELAY_1:    NOP
                NOP
                DJNZ        R6,I2C_DELAY_1
                RET
DELAY_500ms:    MOV         5,#50
DELAY_500ms_1:  LCALL       DELAY_10ms
                DJNZ        R5,DELAY_500ms_1
            RET


DEND:
                END
```

Features

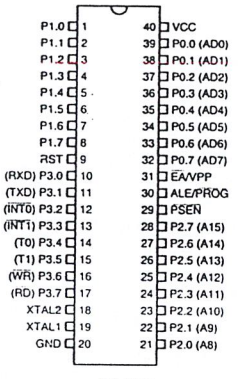
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

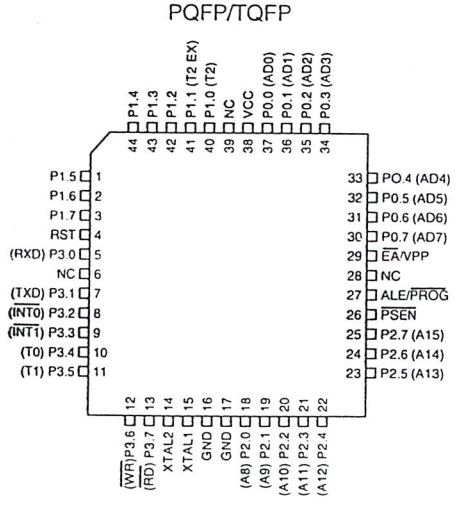
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations

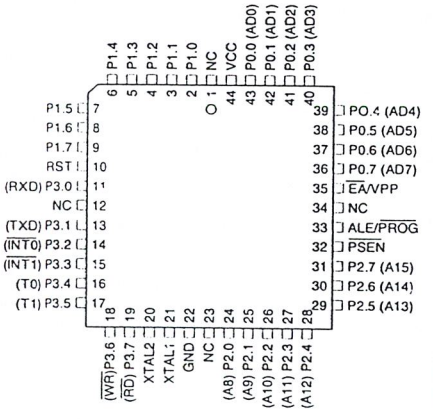
PDIP



PQFP/TQFP



PLCC

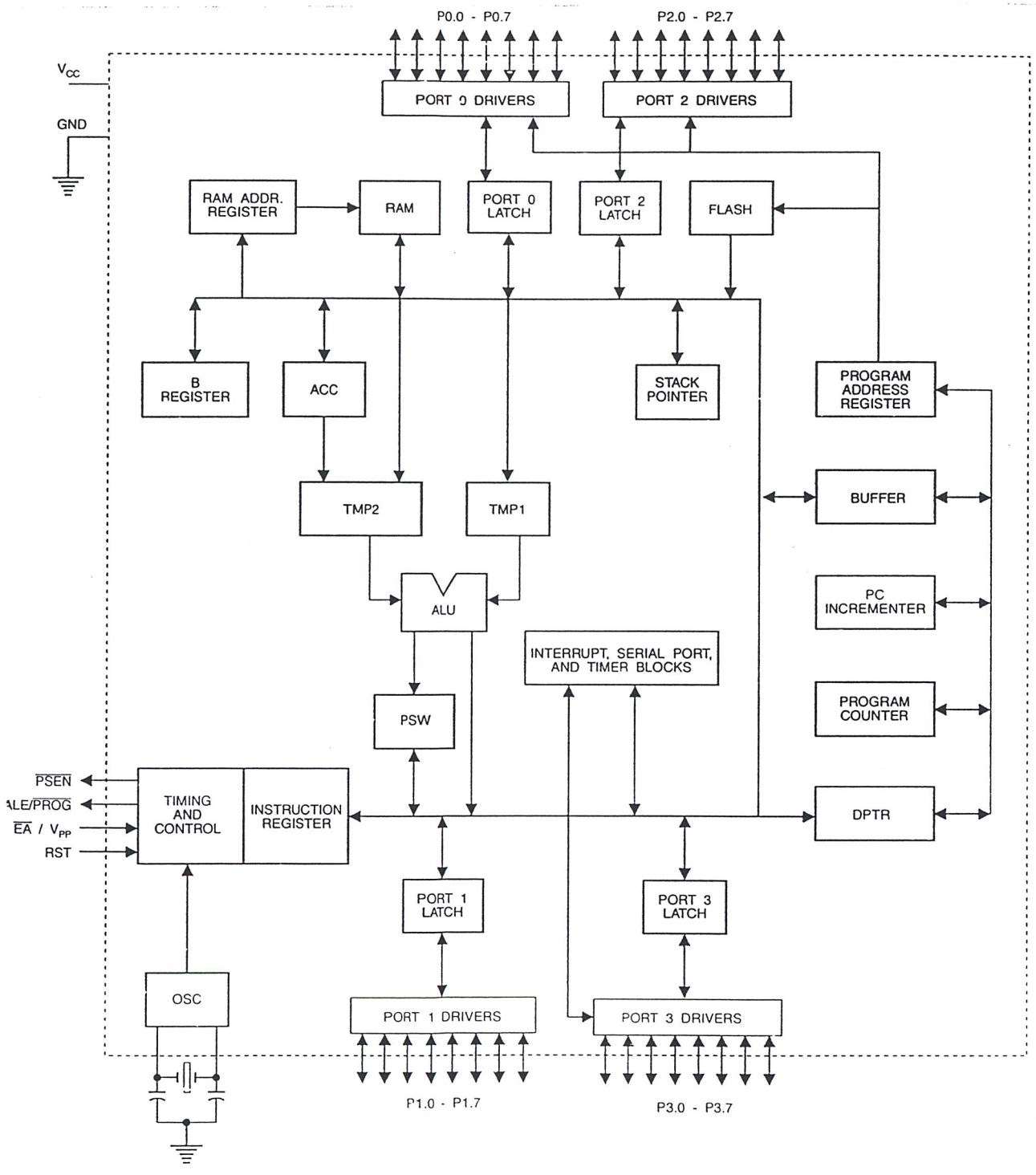


8-bit Microcontroller with 4K Bytes Flash

AT89C51

Rev. 0265G–02/00

Block Diagram



The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

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 pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE

pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/VPP

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

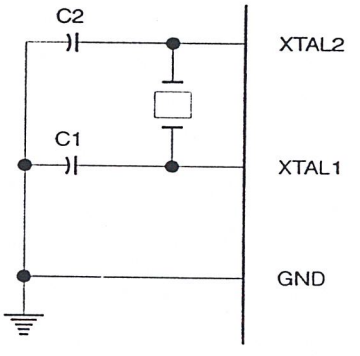
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections

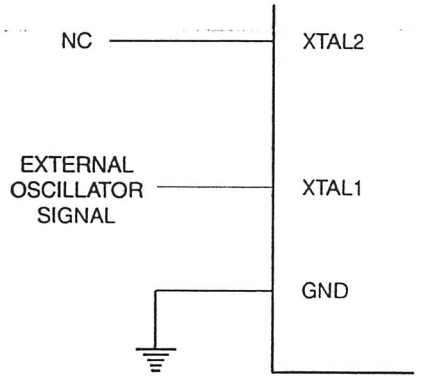


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) =F FH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

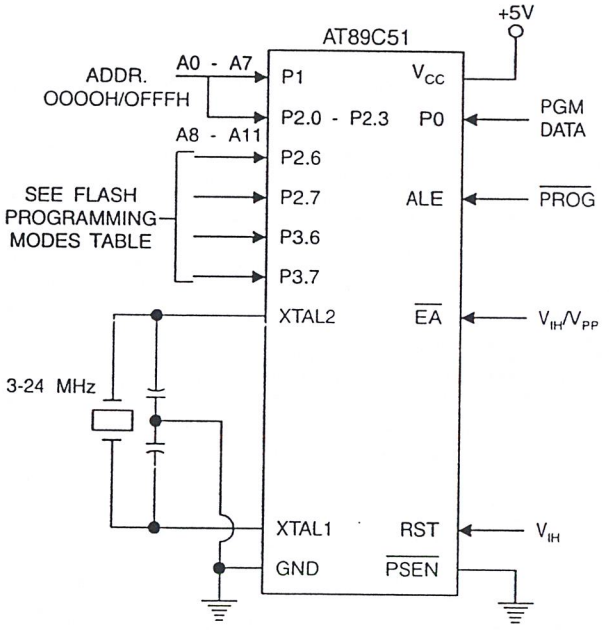
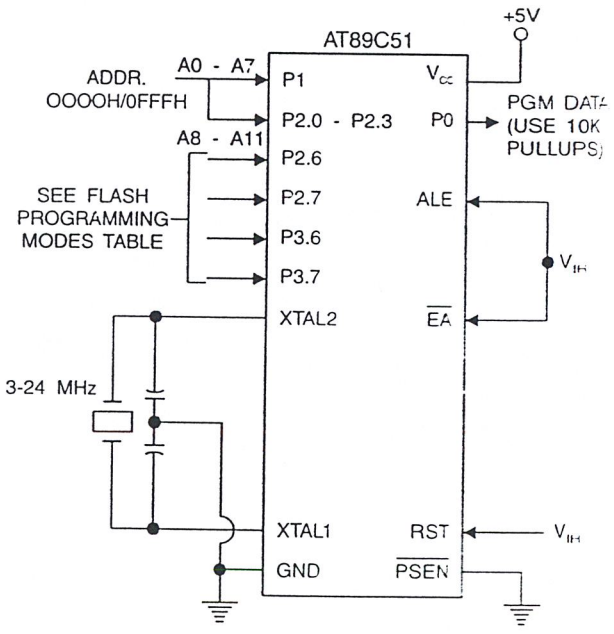
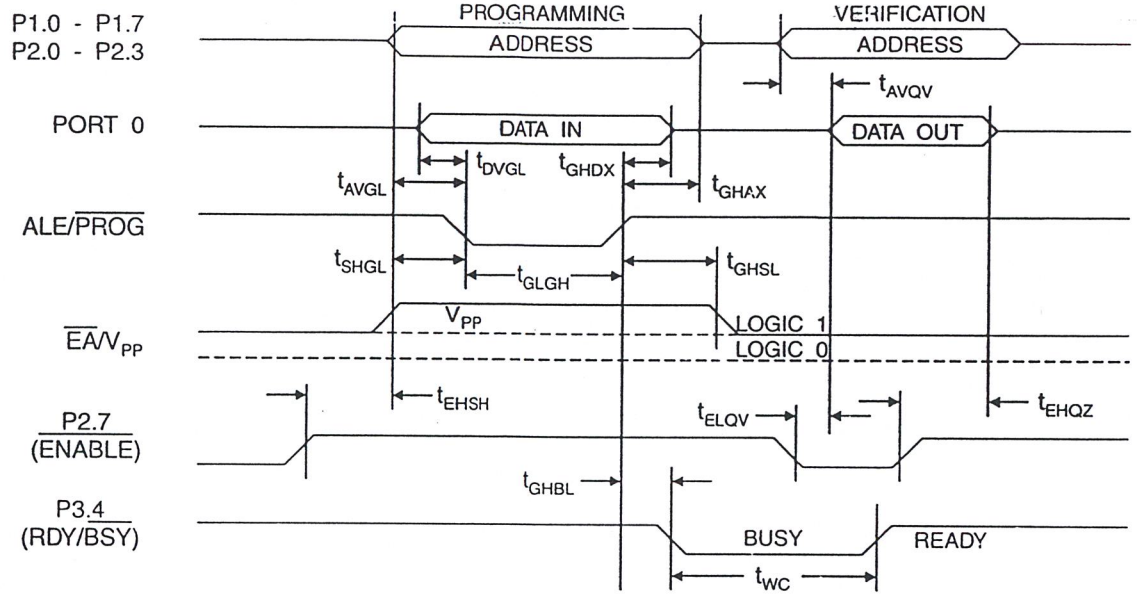


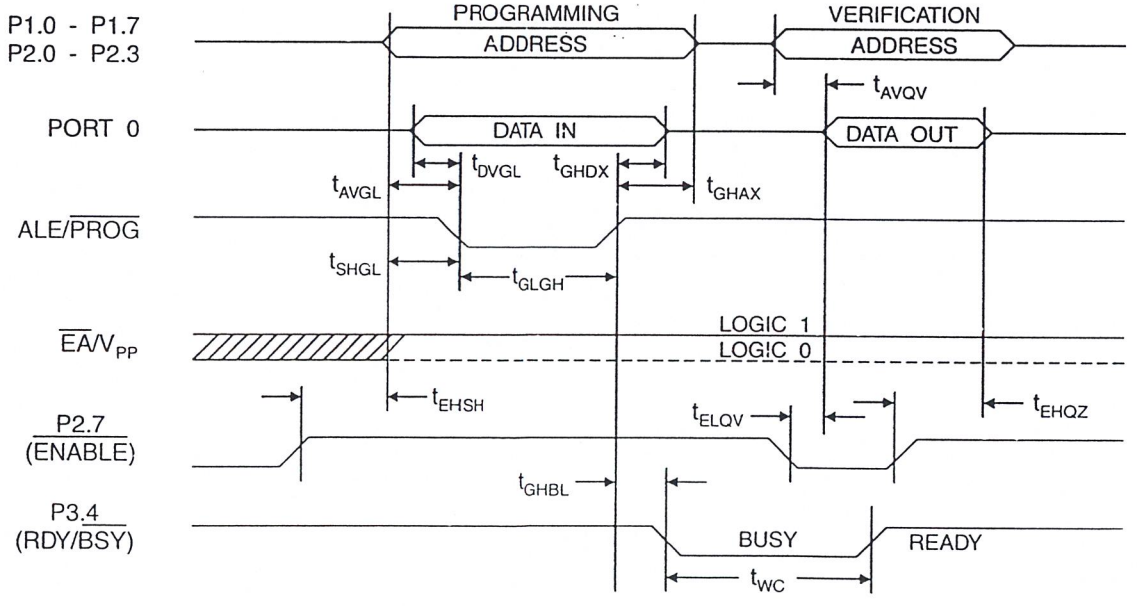
Figure 4. Verifying the Flash



Flash Programming and Verification Waveforms - High-voltage Mode (V_{PP} = 12V)



Flash Programming and Verification Waveforms - Low-voltage Mode (V_{PP} = 5V)



Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}[1]	Programming Enable Voltage	11.5	12.5	V
I_{PP}[1]	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}[1]	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low-voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode(2)	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

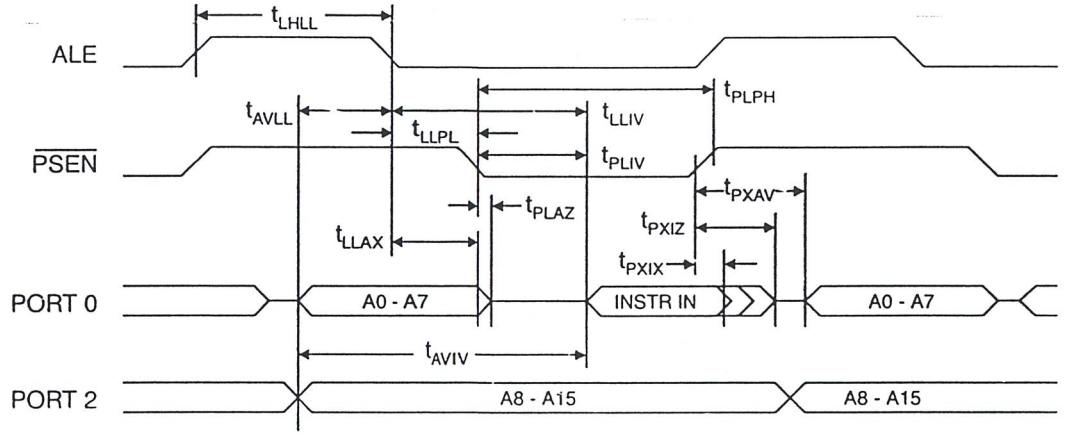
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

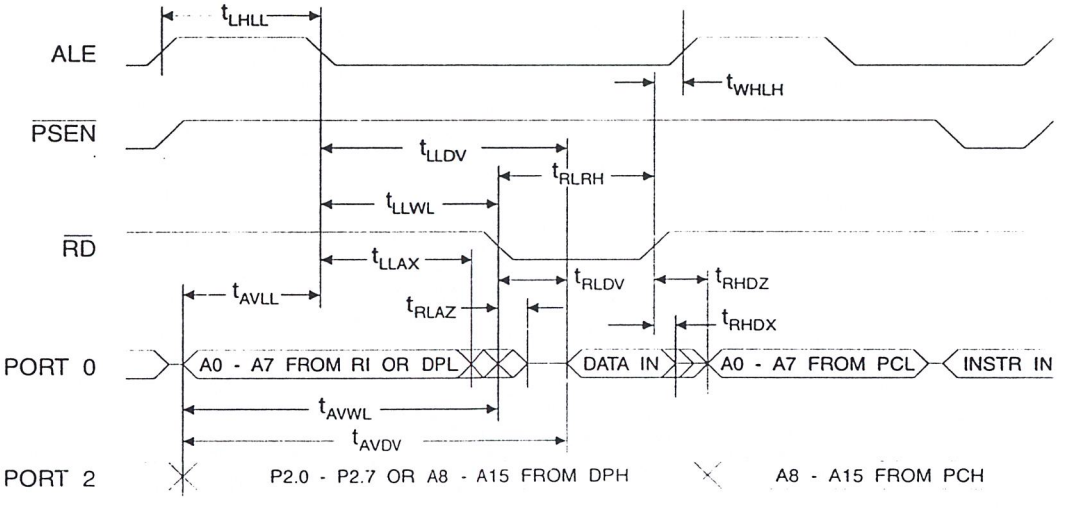
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{CLCL}-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL}-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		$t_{CLCL}-13$		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}-20$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		$t_{CLCL}-10$	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}-165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}-75$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		$t_{CLCL}-20$		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		$t_{CLCL}-20$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CLCL}-20$	$t_{CLCL}+25$	ns

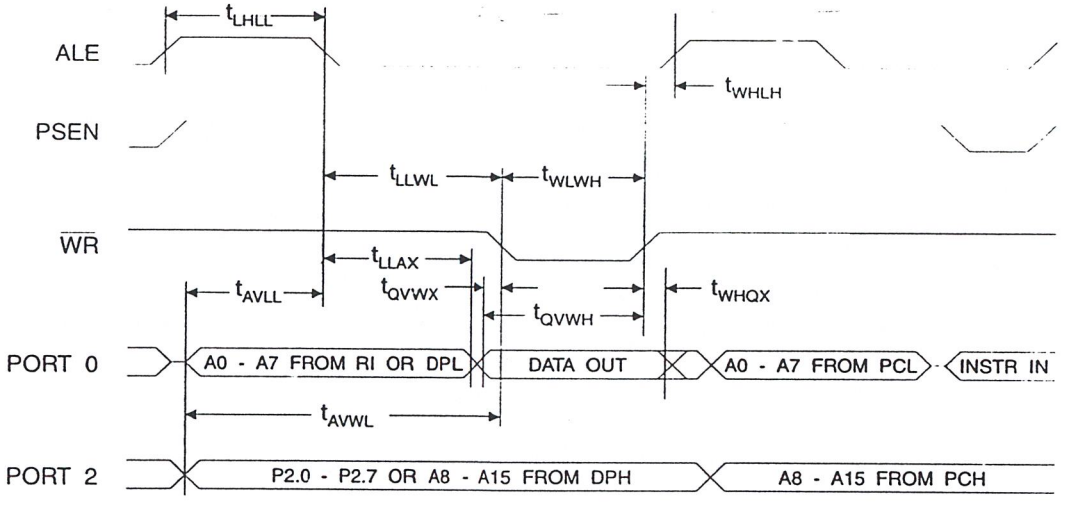
External Program Memory Read Cycle



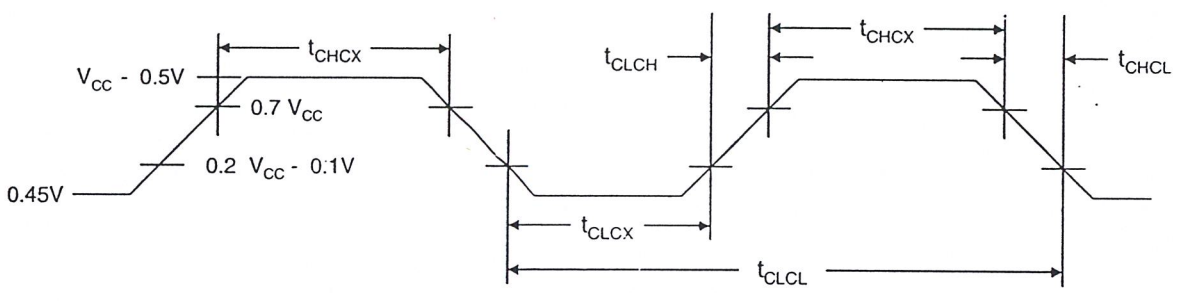
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

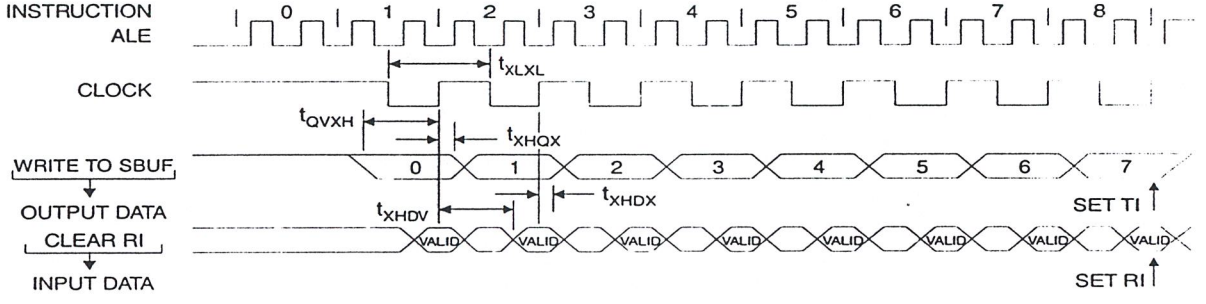
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

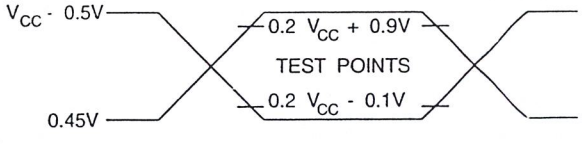
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

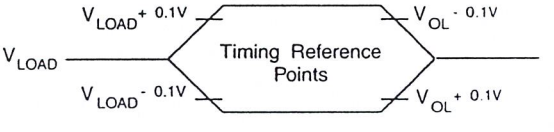


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

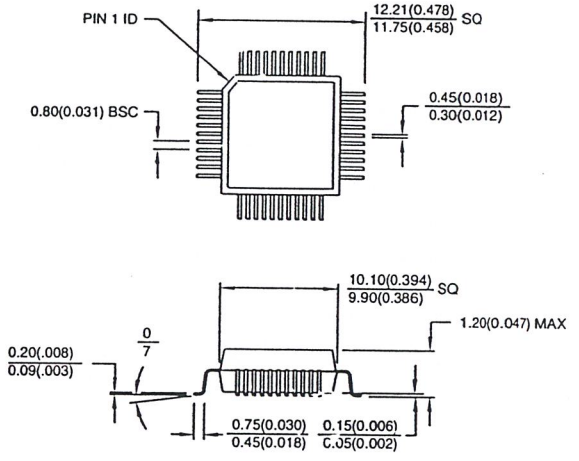
Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial
		AT89C51-12JC	44J	(0°C to 70°C)
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial
		AT89C51-12JI	44J	(-40°C to 85°C)
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial
		AT89C51-16JC	44J	(0°C to 70°C)
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial
		AT89C51-16JI	44J	(-40°C to 85°C)
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial
		AT89C51-20JC	44J	(0°C to 70°C)
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial
		AT89C51-20JI	44J	(-40°C to 85°C)
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
24	5V ± 20%	AT89C51-24AC	44A	Commercial
		AT89C51-24JC	44J	(0°C to 70°C)
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial
		AT89C51-24JI	44J	(-40°C to 85°C)
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

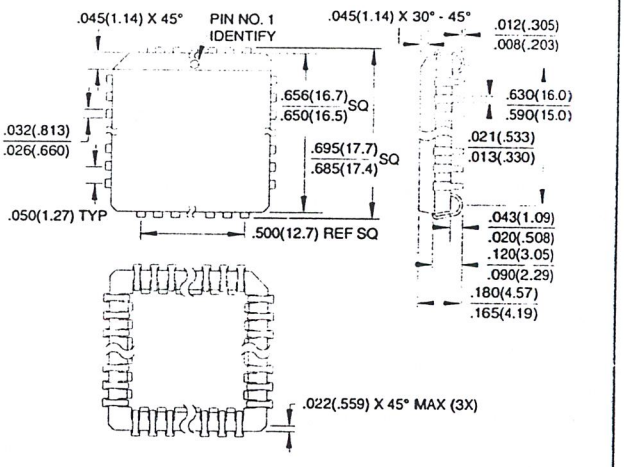
Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

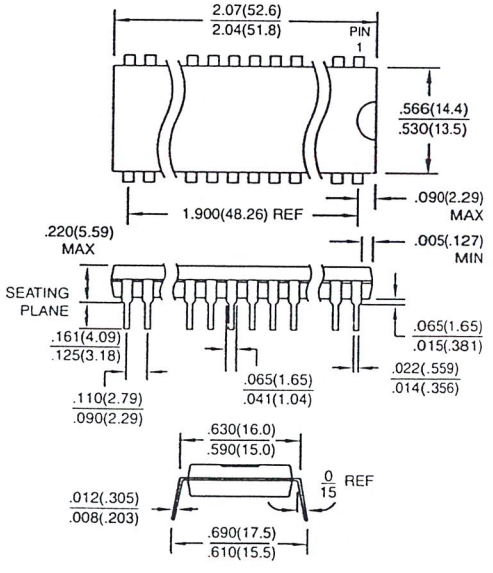


Controlling dimension: millimeters

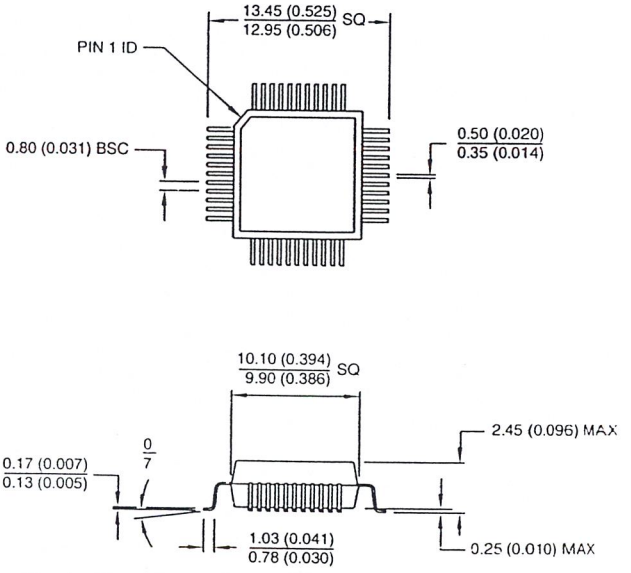
44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC



40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

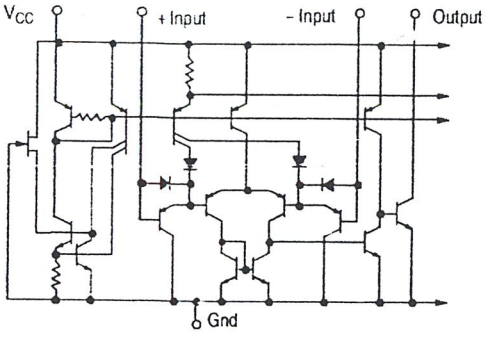
© Atmel Corporation 2000.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

0265G–02/00/xM



NOTE: Diagram shown is for 1 comparator.

Figure 1. Circuit Schematic

ELECTRICAL CHARACTERISTICS (V_{CC} = +5.0 Vdc, T_A = +25°C, unless otherwise noted)

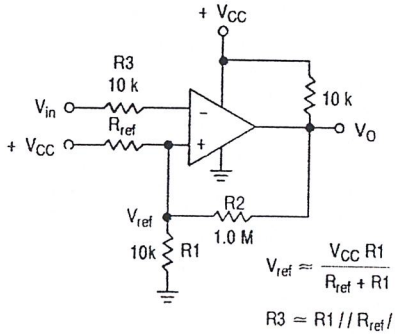
Characteristic	Symbol	LM239/339			LM2901/2901V			MC3302			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage (Note 3.)	V_{IO}	–	±2.0	±5.0	–	±2.0	±7.0	–	±3.0	±20	mVdc
Input Bias Current (Notes 3., 4.) (Output in Analog Range)	I_{IB}	–	25	250	–	25	250	–	25	500	nA
Input Offset Current (Note 3.)	I_{IO}	–	±5.0	±50	–	±5.0	±50	–	±3.0	±100	nA
Input Common Mode Voltage Range	V_{ICMR}	0	–	V_{CC} −1.5	0	–	V_{CC} −1.5	0	–	V_{CC} −1.5	V
Supply Current R_L = ∞ (For All Comparators) R_L = ∞, V_{CC} = 30 Vdc	I_{CC}	 – –	 0.8 1.0	 2.0 2.5	 – –	 0.8 1.0	 2.0 2.5	 – –	 0.8 1.0	 2.0 2.5	mA
Voltage Gain R_L ≥ 15 kΩ, V_{CC} = 15 Vdc	A_{VOL}	50	200	–	25	100	–	25	100	–	V/mV
Large Signal Response Time V_I = TTL Logic Swing, V_{ref} = 1.4 Vdc, V_{RL} = 5.0 Vdc, R_L = 5.1 kΩ	–	–	300	–	–	300	–	–	300	–	ns
Response Time (Note 5.) V_{RL} = 5.0 Vdc, R_L = 5.1 kΩ	–	–	1.3	–	–	1.3	–	–	1.3	–	µs
Output Sink Current V_I (−) ≥ +1.0 Vdc, V_I(+) = 0, V_O ≤ 1.5 Vdc	I_{Sink}	6.0	16	–	6.0	16	–	6.0	16	–	mA
Saturation Voltage V_I(−) ≥ +1.0 Vdc, V_I(+) = 0, I_{sink} ≤ 4.0 mA	V_{sat}	–	130	400	–	130	400	–	130	500	mV
Output Leakage Current V_I(+) ≥ +1.0 Vdc, V_I(−) = 0, V_O = +5.0 Vdc	I_{OL}	–	0.1	–	–	0.1	–	–	0.1	–	nA

2. (LM239) T_{low} = −25°C, T_{high} = +85°
(LM339) T_{low} = 0°C, T_{high} = +70°C
(MC3302) T_{low} = −40°C, T_{high} = +85°C
(LM2901) T_{low} = −40°C, T_{high} = +105°
(LM2901V) T_{low} = −40°C, T_{high} = +125°C
(NCV2901) T_{low} = −40°C, T_{high} = +125°C. Guaranteed by design. NCV prefix is for automotive and other applications requiring site and change control.
3. At the output switch point, V_C = 1.4 Vdc, R_S ≤ 100 Ω 5.0 Vdc ≤ V_{CC} ≤ 30 Vdc, with the inputs over the full common mode range (0 Vdc to V_{CC} −1.5 Vdc).
4. The bias current flows out of the inputs due to the PNP input stage. This current is virtually constant, independent of the output state.
5. The response time specified is for a 100 mV input step with 5.0 mV overdrive. For larger signals, 300 ns is typical.

PERFORMANCE CHARACTERISTICS (V_{CC} = +5.0 Vdc, T_A = T_{low} to T_{high} [Note 6.])

Characteristic	Symbol	LM239/339			LM2901/2901V			MC3302			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage (Note 7.)	V_{IO}	–	–	±9.0	–	–	±15	–	–	±40	mVdc
Input Bias Current (Notes 7., 8.) (Output in Analog Range)	I_{IB}	–	–	400	–	–	500	–	–	1000	nA
Input Offset Current (Note 7.)	I_{IO}	–	–	±150	–	–	±200	–	–	±300	nA
Input Common Mode Voltage Range	V_{ICMR}	0	–	V_{CC} –2.0	0	–	V_{CC} –2.0	0	–	V_{CC} –2.0	V
Saturation Voltage $V_I(-) \geq +1.0$ Vdc, $V_I(+) = 0$, $I_{sink} \leq 4.0$ mA	V_{sat}	–	–	700	–	–	700	–	–	700	mV
Output Leakage Current $V_I(+) \geq +1.0$ Vdc, $V_I(-) = 0$, $V_O = 30$ Vdc	I_{OL}	–	–	1.0	–	–	1.0	–	–	1.0	µA
Differential Input Voltage All $V_I \geq 0$ Vdc	V_{ID}	–	–	V_{CC}	–	–	V_{CC}	–	–	V_{CC}	Vdc

6. (LM239) T_{low} = –25°C, T_{high} = +85°
   (LM339) T_{low} = 0°C, T_{high} = +70°C
   (MC3302) T_{low} = –40°C, T_{high} = +85°C
   (LM2901) T_{low} = –40°C, T_{high} = +105°
   (LM2901V) T_{low} = –40°C, T_{high} = +125°C
   (NCV2901) T_{low} = –40°C, T_{high} = +125°C. Guaranteed by design. NCV prefix is for automotive and other applications requiring site and change control.
7. At the output switch point, $V_O \simeq$ 1.4 Vdc, $R_S \leq 100\ \Omega$ 5.0 Vdc ≤ V_{CC} ≤ 30 Vdc, with the inputs over the full common mode range (0 Vdc to V_{CC} –1.5 Vdc).
8. The bias current flows out of the inputs due to the PNP input stage. This current is virtually constant, independent of the output state.
9. The response time specified is for a 100 mV input step with 5.0 mV overdrive. For larger signals, 300 ns is typical.



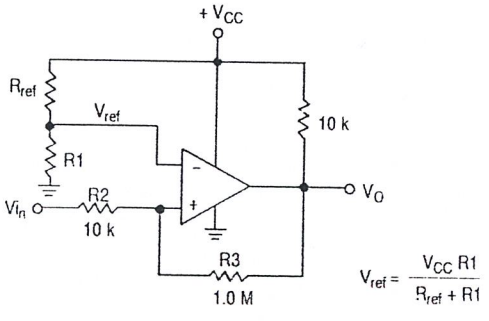
$$V_{ref} \simeq \frac{V_{CC}\,R1}{R_{ref} + R1}$$

$$R3 \simeq R1\,//\,R_{ref}\,//\,R2$$

$$V_H = \frac{R1\,//\,R_{ref}}{R1\,//\,R_{ref} + R2}\,[V_{O(max)} - V_{O(min)}]$$

$$R2 \gg Rref\,//\,R1$$

Figure 2. Inverting Comparator with Hystersis



$$V_{ref} = \frac{V_{CC}\,R1}{R_{ref} + R1}$$

$$R2 \simeq R1\,//\,R_{ref}$$

Amount of Hysteresis V_H

$$V_H = \frac{R2}{R2 + R3}\,[(V_{O(max)} - V_{O(min)}]$$

Figure 3. Noninverting Comparator with Hysteresis

Typical Characteristics

(V_{CC} = 15 Vdc, T_A = +25°C (each comparator) unless otherwise noted.)

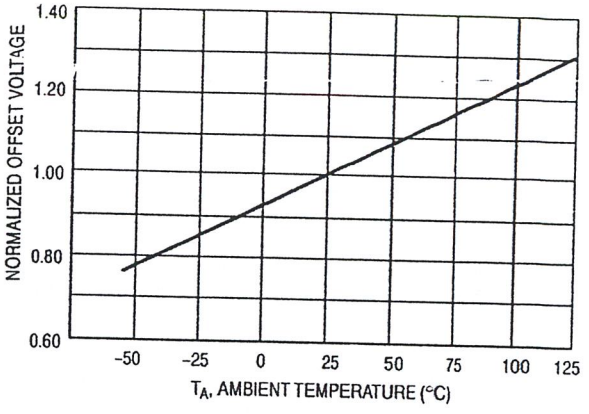


Figure 4. Normalized Input Offset Voltage

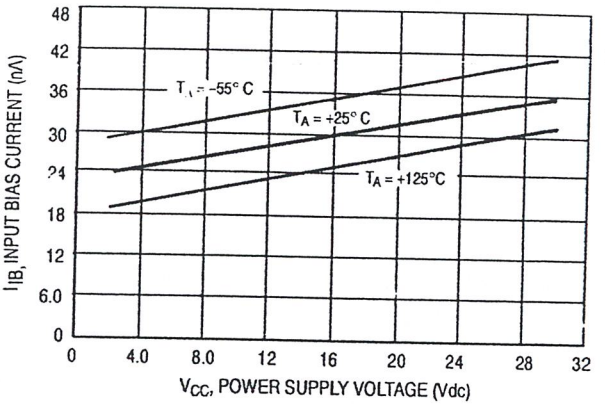


Figure 5. Input Bias Current

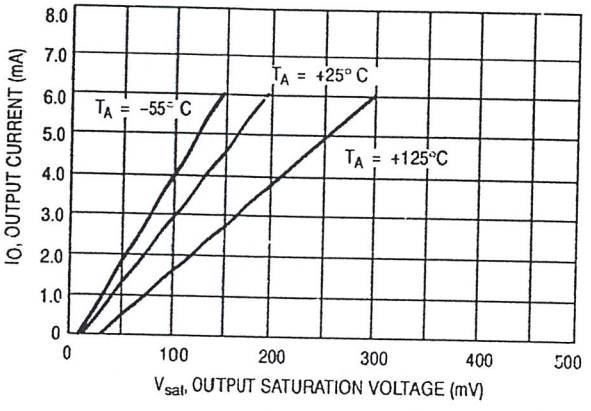
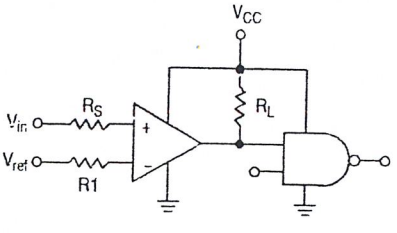


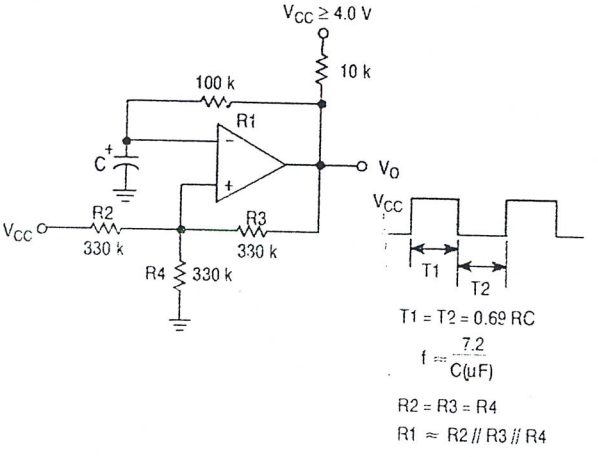
Figure 6. Output Sink Current versus Output Saturation Voltage



R_S = Source Resistance
$R1 \simeq R_S$

Logic	Device	V_{CC} (V)	R_L kΩ
CMOS	1/4 MC14001	+15	100
TTL	1/4 MC7400	+5.0	10

Figure 7. Driving Logic



T1 = T2 = 0.69 RC

$f \simeq \frac{7.2}{C(\mu F)}$

R2 = R3 = R4

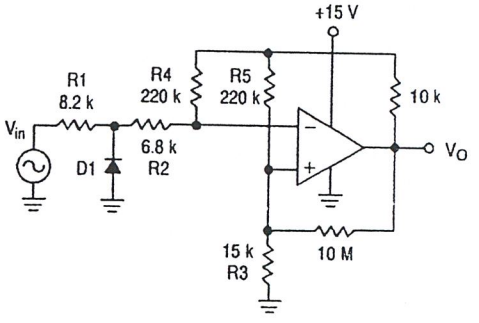
R1 ≈ R2 // R3 // R4

Figure 8. Squarewave Oscillator

APPLICATIONS INFORMATION

These quad comparators feature high gain, wide bandwidth characteristics. This gives the device oscillation tendencies if the outputs are capacitively coupled to the inputs via stray capacitance. This oscillation manifests itself during output transitions (V_{OL} to V_{OH}). To alleviate this situation input resistors < 10 kΩ should be used. The addition of positive feedback (< 10 mV) is also recommended. It is good design practice to ground all unused input pins.

Differential input voltages may be larger than supply voltages without damaging the comparator's inputs. Voltages more negative than −300 mV should not be used.



D1 prevents input from going negative by more than 0.6 V.

$$R1 + R2 = R3$$

$$R3 \leq \frac{R5}{10}$$ for small error in zero crossing

Figure 9. Zero Crossing Detector (Single Supply)

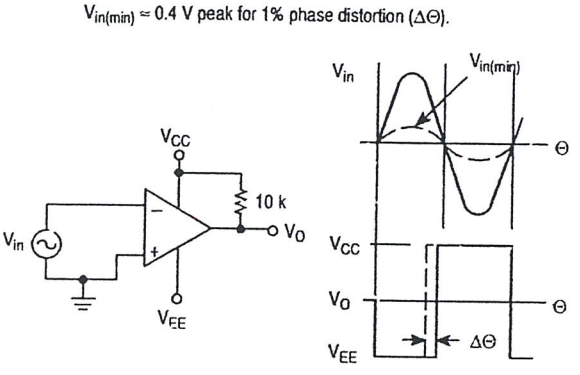


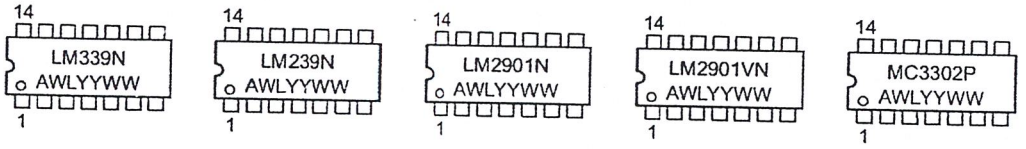
Figure 10. Zero Crossing Detector (Split Supplies)

ORDERING INFORMATION

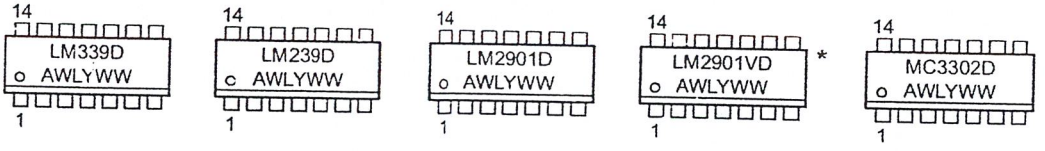
Device	Package	Shipping
LM239D	SO–14	55 Units/Rail
LM239DR2	SO–14	2500 Units/Tape & Reel
LM239N	PDIP–14	25 Units/Rail
LM339D	SO–14	55 Units/Rail
LM339DR2	SO–14	2500 Units/Tape & Reel
LM339N	PDIP–14	25 Units/Rail
LM2901D	SO–14	55 Units/Rail
LM2901DR2	SO–14	2500 Units/Tape & Reel
LM2901N	PDIP–14	25 Units/Rail
LM2901VDR2	SO–14	2500 Units/Tape & Reel
LM2901VN	PDIP–14	25 Units/Rail
NCV2901DR2	SO–14	2500 Units/Tape & Reel
MC3302D	SO–14	55 Units/Rail
MC3302DR2	SO–14	2500 Units/Tape & Reel
MC3302P	PDIP–14	25 Units/Rail

MARKING DIAGRAMS

PDIP–14
N, P SUFFIX
CASE 646



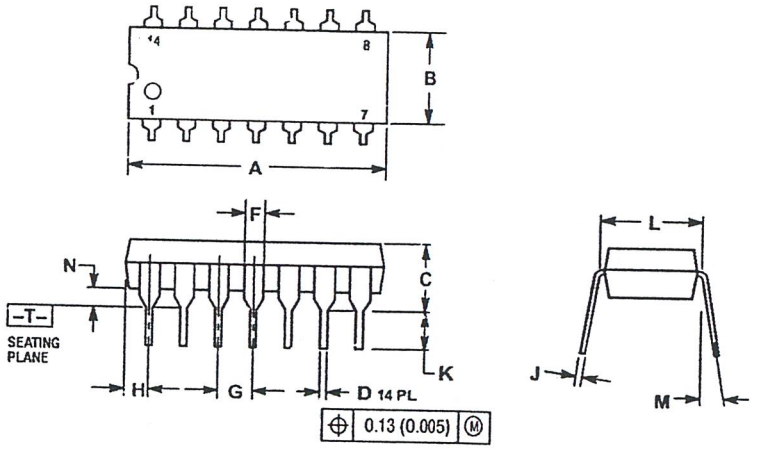
SO–14
D SUFFIX
CASE 751A



A = Assembly Location
WL = Wafer Lot
YY, Y = Year
WW = Work Week
*This marking diagram also applies to NCV2901.

PACKAGE DIMENSIONS

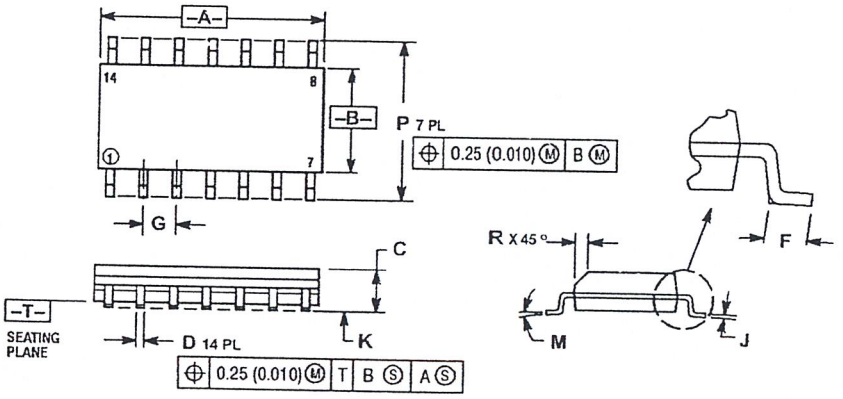
PDIP–14
P SUFFIX
CASE 646–06
ISSUE M



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
5. ROUNDED CORNERS OPTIONAL.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.715	0.770	18.16	18.80
B	0.240	0.260	6.10	6.60
C	0.145	0.185	3.69	4.69
D	0.015	0.021	0.38	0.53
F	0.040	0.070	1.02	1.78
G	0.100 BSC		2.54 BSC	
H	0.052	0.095	1.32	2.41
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.290	0.310	7.37	7.87
M	---	10°	---	10°
N	0.015	0.039	0.38	1.01

SO–14
D SUFFIX
CASE 751A–03
ISSUE F



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.127 (0.005) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	8.55	8.75	0.337	0.344
B	3.80	4.00	0.150	0.157
C	1.35	1.75	0.054	0.068
D	0.35	0.49	0.014	0.019
F	0.40	1.25	0.016	0.049
G	1.27 BSC		0.050 BSC	
J	0.19	0.25	0.008	0.009
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	5.80	6.20	0.228	0.244
R	0.25	0.50	0.010	0.019

SN74LS240, SN74LS244

Octal Buffer/Line Driver with 3-State Outputs

The SN74LS240 and SN74LS244 are Octal Buffers and Line Drivers designed to be employed as memory address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC board density.

- Hysteresis at Inputs to Improve Noise Margins
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- Input Clamp Diodes Limit High-Speed Termination Effects

GUARANTEED OPERATING RANGES

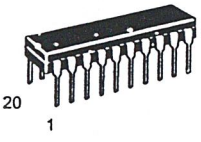
Symbol	Parameter	Min	Typ	Max	Unit
V_{CC}	Supply Voltage	4.75	5.0	5.25	V
T_A	Operating Ambient Temperature Range	0	25	70	°C
I_{OH}	Output Current – High			–3.0	mA
				–15	mA
I_{OL}	Output Current – Low			24	mA

ON Semiconductor

http://onsemi.com

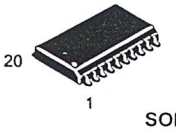
LOW POWER SCHOTTKY

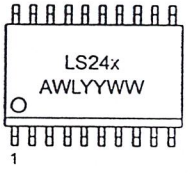
MARKING DIAGRAMS



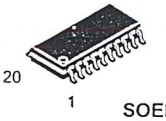


PDIP–20
N SUFFIX
CASE 738





SOIC–20
DW SUFFIX
CASE 751D





SOEIAJ–20
M SUFFIX
CASE 967

x = 0 or 4
A = Assembly Location
WL = Wafer Lot
YY = Year
WW = Work Week

ORDERING INFORMATION

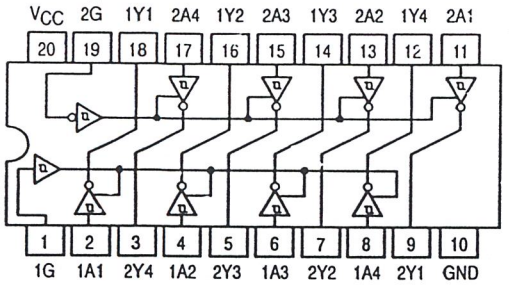
See detailed ordering and shipping information in the package dimensions section on page 5 of this data sheet.

© Semiconductor Components Industries, LLC, 2001
October, 2001 – Rev. 8

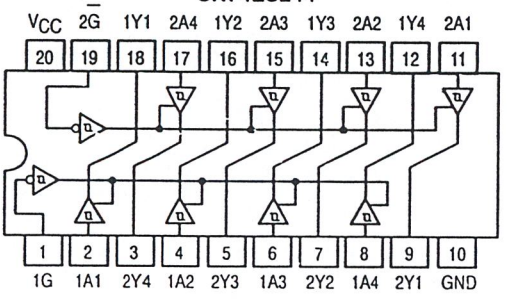
Publication Order Number:
SN74LS240/D

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)

SN74LS240



SN74LS244



TRUTH TABLES

SN74LS240

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	H
L	H	L
H	X	(Z)

SN74LS244

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	L
L	H	H
H	X	(Z)

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = HIGH Impedance

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Typ	Max	Unit	Test Conditions	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage				0.8	V	Guaranteed Input LOW Voltage for All Inputs	
$V_{T+}-V_{T-}$	Hysteresis		0.2	0.4		V	V_{CC} = MIN	
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage		2.4	3.4		V	V_{CC} = MIN, I_{OH} = −3.0 mA	
			2.0			V	V_{CC} = MIN, I_{OH} = MAX	
V_{OL}	Output LOW Voltage			0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
				0.35	0.5	V	I_{OL} = 24 mA	
I_{OZH}	Output Off Current HIGH				20	μA	V_{CC} = MAX, V_{OUT} = 2.7 V	
I_{OZL}	Output Off Current LOW				−20	μA	V_{CC} = MAX, V_{OUT} = 0.4 V	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.2	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Output Short Circuit Current (Note 1)		−40		−225	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current Total, Output HIGH				27	mA	V_{CC} = MAX	
	Total, Output LOW	LS240			44			
		LS244			46			
	Total at HIGH Z	LS240			50			
		LS244			54			

1. Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits Min	Typ	Max	Unit	Test Conditions
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS240		9.0 12	14 18	ns	C_L = 45 pF, R_L = 667 Ω
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS244		12 12	18 18	ns	
t_{PZH}	Output Enable Time to HIGH Level		15	23	ns	
t_{PZL}	Output Enable Time to LOW Level		20	30	ns	
t_{PLZ}	Output Disable Time from LOW Level		15	25	ns	C_L = 5.0 pF, R_L = 667 Ω
t_{PHZ}	Output Disable Time from HIGH Level		10	18	ns	

AC WAVEFORMS

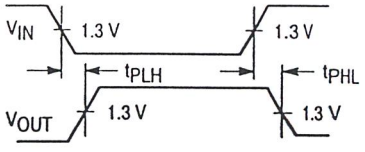


Figure 1.

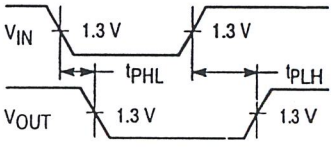


Figure 2.

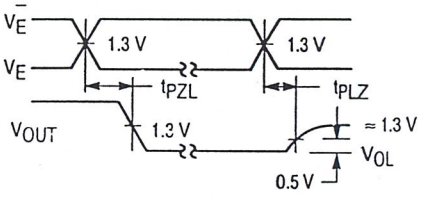


Figure 3.

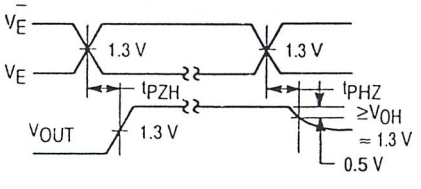
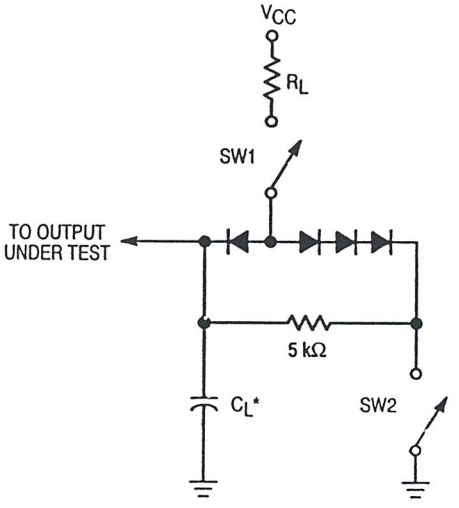


Figure 4.



SWITCH POSITIONS

SYMBOL	SW1	SW2
t_{PZH}	Open	Closed
t_{PZL}	Closed	Open
t_{PLZ}	Closed	Closed
t_{PHZ}	Closed	Closed

Figure 5.

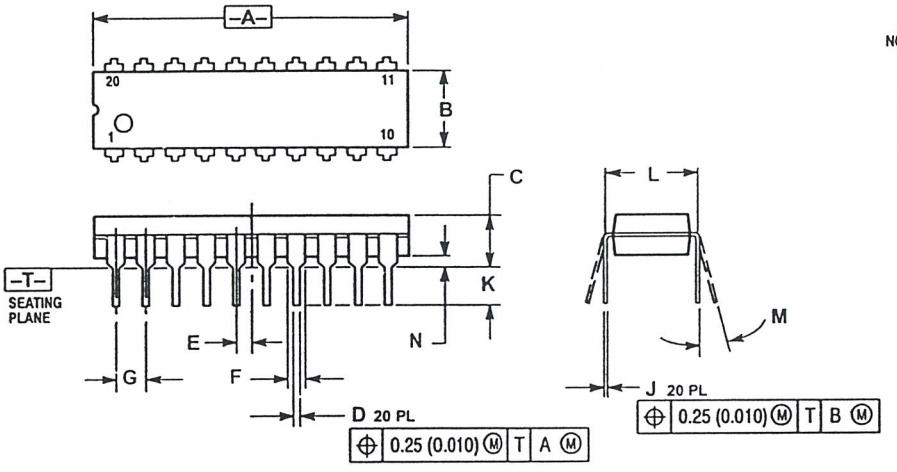
DEVICE ORDERING INFORMATION

Device Order Number	Package Type	Tape and Reel Size
SN74LS240N	PDIP–20	1440 Units/Box
SN74LS240DW	SOIC–WIDE	38 Units/Rail
SN74LS240DWR2	SOIC–WIDE	2500/Tape and Reel
SN74LS240M	SOEIAJ–20	See Note 2
SN74LS240MEL	SOEIAJ–20	See Note 2
SN74LS244N	PDIP–20	1440 Units/Box
SN74LS244DW	SOIC–WIDE	38 Units/Rail
SN74LS244DWR2	SOIC–WIDE	2500/Tape and Reel
SN74LS244M	SOEIAJ–20	See Note 2
SN74LS244MEL	SOEIAJ–20	See Note 2

2. For ordering information on the EIAJ version of the SOIC package, please contact your local ON Semiconductor representative.

PACKAGE DIMENSIONS

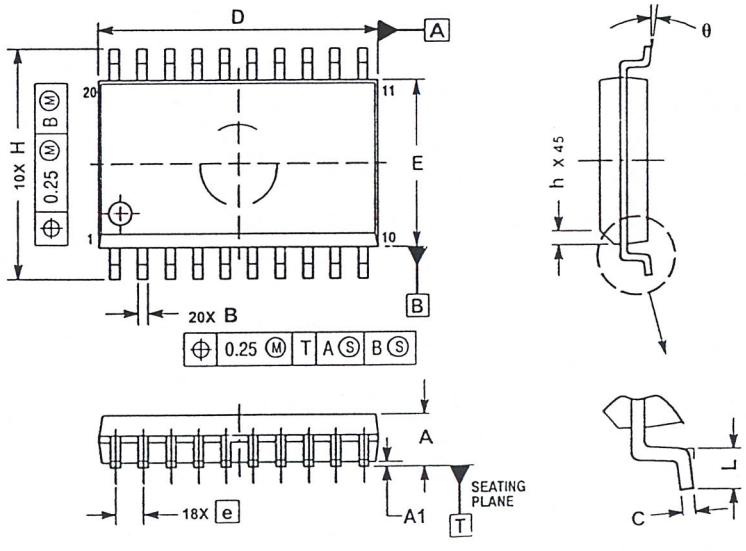
N SUFFIX
PLASTIC PACKAGE
CASE 738–03
ISSUE E



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.010	1.070	25.66	27.17
B	0.240	0.260	6.10	6.60
C	0.150	0.180	3.81	4.57
D	0.015	0.022	0.39	0.55
E	0.050 BSC		1.27 BSC	
F	0.050	0.070	1.27	1.77
G	0.100 BSC		2.54 BSC	
J	0.008	0.015	0.21	0.38
K	0.110	0.140	2.80	3.55
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.01

D SUFFIX
PLASTIC SOIC PACKAGE
CASE 751D–05
ISSUE F

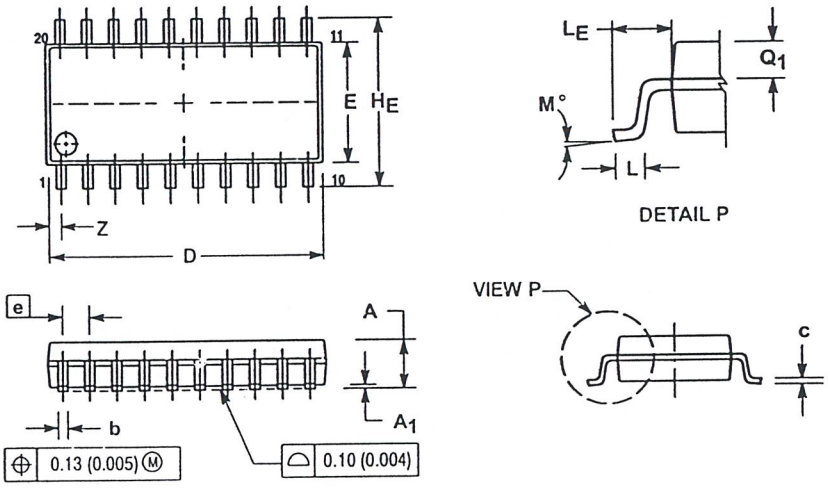


NOTES:
1. DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M, 1994.
3. DIMENSIONS D AND E DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 PER SIDE.
5. DIMENSION B DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE PROTRUSION SHALL BE 0.13 TOTAL IN EXCESS OF B DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS	
DIM	MIN	MAX
A	2.35	2.65
A1	0.10	0.25
B	0.35	0.49
C	0.23	0.32
D	12.65	12.95
E	7.40	7.60
e	1.27 BSC	
H	10.05	10.55
h	0.25	0.75
L	0.50	0.90
θ	0°	7°

PACKAGE DIMENSIONS

M SUFFIX
SOEIAJ PACKAGE
CASE 967–01
ISSUE O



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS D AND E DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS AND ARE MEASURED AT THE PARTING LINE. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.15 (0.006) PER SIDE.
4. TERMINAL NUMBERS ARE SHOWN FOR REFERENCE ONLY.
5. THE LEAD WIDTH DIMENSION (b) DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE LEAD WIDTH DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSIONS AND ADJACENT LEAD TO BE 0.46 (0.018).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	---	2.05	---	0.081
A_1	0.05	0.20	0.002	0.008
b	0.35	0.50	0.014	0.020
c	0.18	0.27	0.007	0.011
D	12.35	12.80	0.486	0.504
E	5.10	5.45	0.201	0.215
e	1.27 BSC		0.050 BSC	
H_E	7.40	8.20	0.291	0.323
L	0.50	0.85	0.020	0.033
L_E	1.10	1.50	0.043	0.059
M	0 °	10 °	0 °	10 °
Q_1	0.70	0.90	0.028	0.035
Z	---	0.81	---	0.032

DALLAS SEMICONDUCTOR

DS1307/DS1308
64 X 8 Serial Real Time Clock

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
- 56 byte nonvolatile RAM for data storage
- 2–wire serial interface
- Programmable squarewave output signal
- Automatic power fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode with oscillator running
- Optional industrial temperature range –40°C to +85°C (IND) available for DS1307 and DS1308
- DS1307 available in 8–pin DIP or SOIC
- DS1308 available in 36–pin SMD BGA (Ball Grid Array)
- DS1308 accuracy is better than ±2 minute/month at 25°C

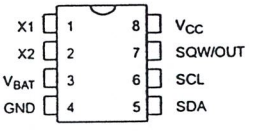
ORDERING INFORMATION

DS1307	Serial Timekeeping Chip; 8–pin DIP
DS1307Z	Serial Timekeeping Chip; 8–pin SOIC (150 mil)
DS1307N	8–pin DIP (IND)
DS1307ZN	8–pin SOIC (IND)
DS1308	36–pin BGA
DS1308N	36–pin BGA(IND)

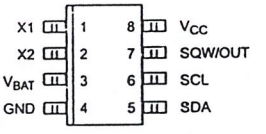
DESCRIPTION

The DS1307 Serial Real Time Clock is a low power, full BCD clock/calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2–wire bi–directional bus. The 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 AM/PM indicator. The DS1307 has a built–in power sense circuit which detects power failures and automatically switches to the battery supply.

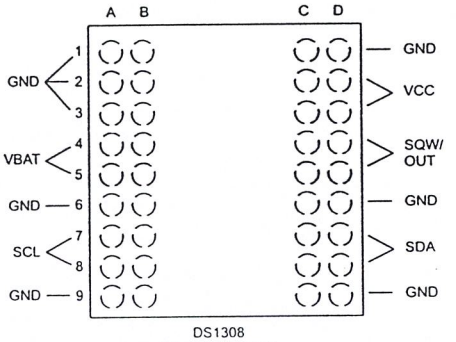
PIN ASSIGNMENT



DS1307 8–PIN DIP (300 MIL)



DS1307Z 8–PIN SOIC (150 MIL)



DS1308
36–PIN SMD BGA
(TOP VIEW)

PIN DESCRIPTION DS1307/DS1308

V_{CC}	– Primary Power Supply
X1, X2	– 32.768 kHz Crystal Connection
V_{BAT}	– +3 Volt Battery Input
GND	– Ground
SDA	– Serial Data
SCL	– Serial Clock
SQW/OUT	– Square wave/Output Driver

DS1308 PIN IDENTIFIER

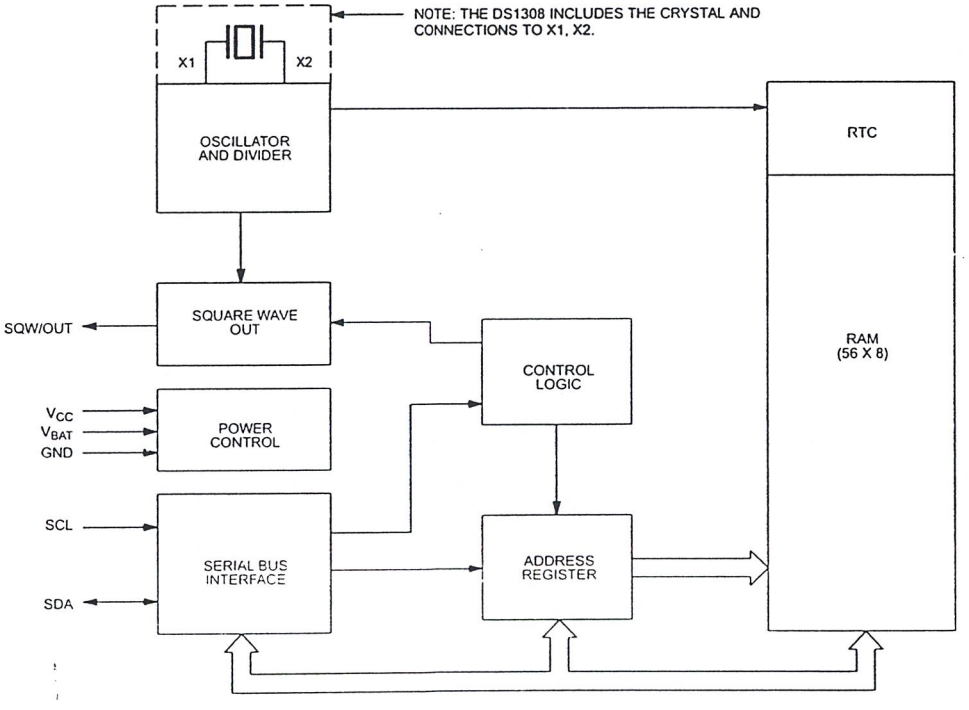
V_{CC}	– C2, C3, D2, D3
V_{BAT}	– A4, A5, B4, B5
SDA	– C7, C8, D7, D8
SCL	– A7, A8, B7, B8
SQW/OUT	– C4, C5, D4, D5
GND	– All Remaining Balls

The DS1308 incorporates the DS1307 chip with a 32.768 kHz crystal in a surface mountable, 36–pin ball grid array package (BGA). The close proximity of the embedded crystal to the high impedance crystal input pins on the DS1307 minimizes capacitive loading and noise injection problems associated with many other oscillator designs. The total area required for installation is less than that of 1 United States dime: thus, minimizing PCB space required.

OPERATION

The DS1307/1308 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT} the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low current battery backup mode. Upon power up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT}+0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock.

DS1307/DS1308 BLOCK DIAGRAM Figure 1



CONTROL REGISTER

The DS1307/DS1308 Control Register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	X	X	SQWE	X	X	RS1	RS0

OUT (Output control): This bit controls the output level of the SQW/OUT pin when the square wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0.

SQWE (Square Wave Enable): This bit, when set to a logic 1, will enable the oscillator output. The frequency of the square wave output depends upon the value of the RS0 and RS1 bits.

RS (Rate Select): These bits control the frequency of the square wave output when the square wave output has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

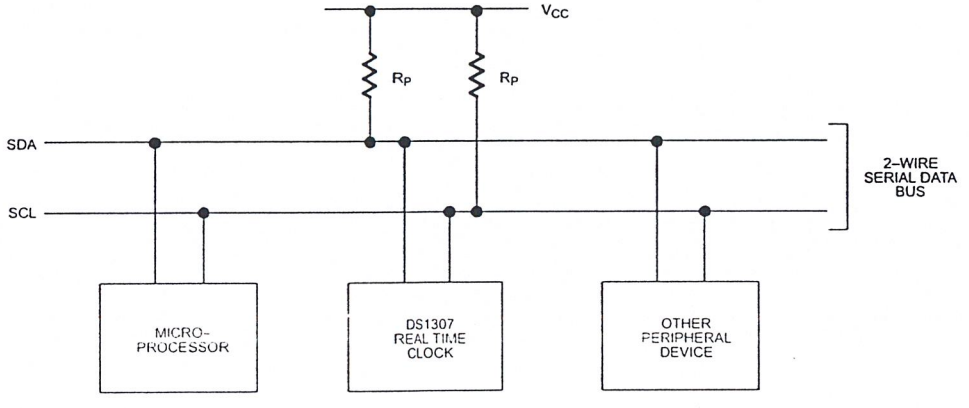
SQUAREWAVE OUTPUT FREQUENCY Table 1

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

2–WIRE SERIAL DATA BUS

The DS1307 supports a bi–directional 2–wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307/DS1308 operates as a slave on the 2–wire bus. A typical bus configuration using this 2–wire protocol is show in Figure 4.

TYPICAL 2–WIRE BUS CONFIGURATION Figure 4



Figures 5, 6, and 7 detail how data is transferred on the 2–wire bus.

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

SIGNAL DESCRIPTIONS

V_{CC}, GND – DC power is provided to the device on these pins. V_{CC} is the +5 volt input. When 5 volts is applied within normal limits, the device is fully accessible and data can be written and read. When a 3 volt battery is connected to the device and V_{CC} is below 1.25 x V_{BAT}, reads and writes are inhibited. However, the Timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below V_{BAT} the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at V_{BAT}.

V_{BAT} – Battery input for any standard 3 volt lithium cell or other energy source. Battery voltage must be held between 2.0 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which access to the real time clock and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A Lithium battery with 48 mAhr. or greater will back up the DS1307/DS1308 for more than 10 years in the absence of power at 25 degrees C.

SCL (Serial Clock Input) – SCL is used to synchronize data movement on the serial interface.

SDA (Serial Data Input/Output) – SDA is the input/output pin for the 2–wire serial interface. The SDA pin is open drain which requires an external pull–up resistor.

SQW/OUT (Square Wave/ Output Driver) – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1 Hz, 4 KHz, 8 KHz, 32 KHz). The SQW/OUT pin is open drain which requires an external pull–up resistor.

NOTE: X1, X2 are not applicable for the DS1308 or DS1308N.

X1, X2 – Connections for a standard 32.768 KHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

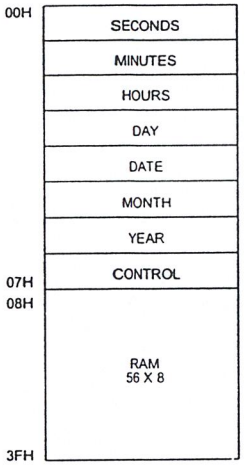
For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks". The DS1307 can also be driven by an external 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

Please review Application Note 95, "Interfacing the DS1307/DS1308 with a 8051–Compatible Microcontroller" for additional information.

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307/DS1308 is shown in Figure 2. The real time clock registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi–byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

DS1307 ADDRESS MAP Figure 2



CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The real time clock registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the Binary–Coded Decimal (BCD) format. Bit 7 of Register 0 is the Clock Halt (CH) bit. When this bit is set to a one, the oscillator is disabled. When cleared to a zero, the oscillator is enabled.

Please note that the initial power on state of all registers is not defined. Therefore it is important to enable the oscillator (CH bit=0) during initial configuration.

The DS1307/DS1308 can be run in either 12–hour or 24–hour mode. Bit 6 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).

DS1307/DS1308 TIMEKEEPER REGISTERS Figure 3

	BIT7							BIT0	
00H	CH	10 SECONDS			SECONDS				00–59
	X	10 MINUTES			MINUTES				00–59
	X	12/24	10 HR A/P	10 HR	HOURS				01–12 / 00–23
	X	X	X	X	X	DAY			1–7
	X	X	10 DATE		DATE				01–28/29 / 01–30 / 01–31
	X	X	X	10 MONTH	MONTH				01–12
	10 YEAR				YEAR				00–99
07H	OUT	X	X	SQWE	X	X	RS1	RS0	

Start data transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

Stop data transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

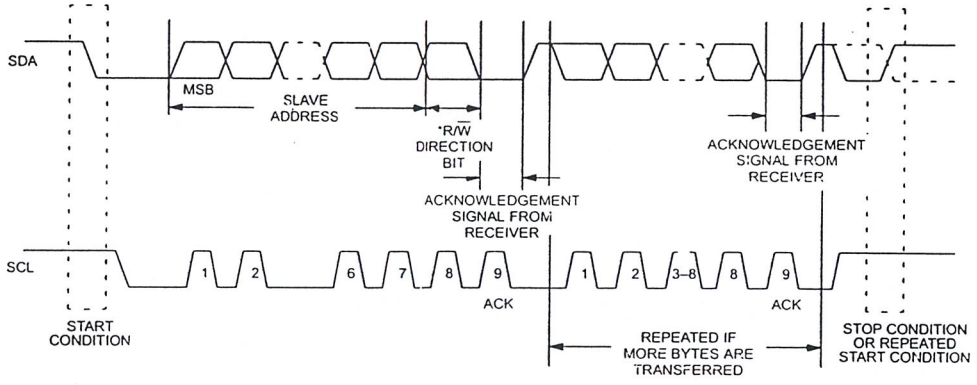
Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte–wise and each receiver acknowledges with a ninth bit.

Within the 2–wire bus specifications a regular mode (100 kHz clock rate) and a fast mode (400 kHz clock rate) are defined. The DS1307/DS1308 operates in the regular mode (100 kHz) only.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition

DATA TRANSFER ON 2–WIRE SERIAL BUS Figure 5



Depending upon the state of the R/$\overline{W}$ bit, two types of data transfer are possible:

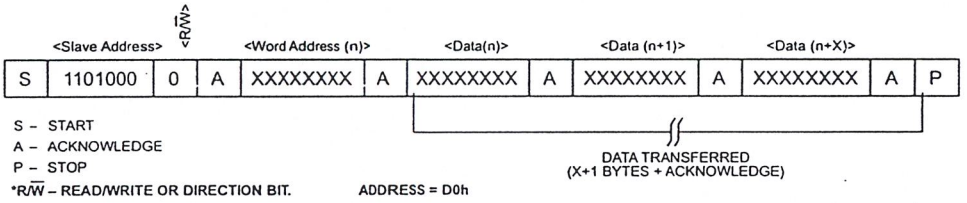
1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a 'not acknowledge' is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307/DS1308 may operate in the following two modes:

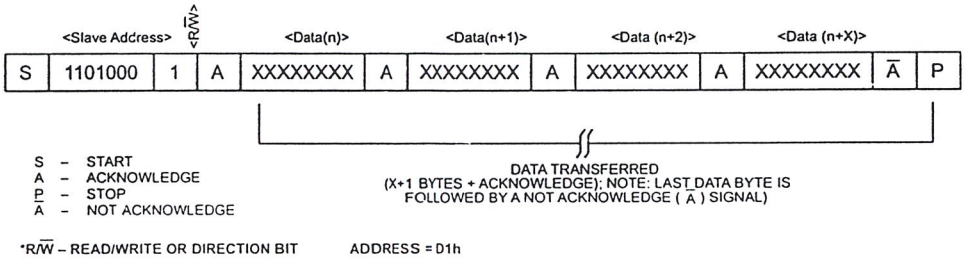
1. **Slave receiver mode (DS1307/DS1308 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and *direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307/DS1308 address, which is 1101000, followed by the *direction bit ($R/\overline{W}$) which, for a write, is a 0. After receiving and decoding the address byte the device outputs an acknowledge on the SDA line. After the DS1307/DS1308 acknowledges the slave address + write bit, the master transmits a register address to the DS1307/DS1308 This will set the register pointer on the DS1307/DS1308. The master will then begin transmitting each byte of data with the DS1307/DS1308 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

DATA WRITE – SLAVE RECEIVER MODE Figure 6



2. **Slave transmitter mode (DS1307/DS1308 read mode)**: The first byte is received and handled as in the slave receiver mode. However, in this mode, the *direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307/DS1308 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7-bit DS1307/DS1308 address, which is 1101000, followed by the *direction bit (R/$\overline{W}$) which, for a read, is a 1. After receiving and decoding the address byte the device inputs an acknowledge on the SDA line. The DS1307/DS1308 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307/DS1308 must receive a Not Acknowledge to end a read.

DATA READ – SLAVE TRANSMITTER MODE Figure 7



ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature	0°C to 70°C
Storage Temperature	–55°C to +125°C
Soldering Temperature	260°C for 10 seconds (See NOTE 12)

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

The Dallas Semiconductor DS1307/DS1308 is built to the highest quality standards and manufactured for long term reliability. All Dallas Semiconductor devices are made using the same quality materials and manufacturing methods. However, standard versions of the DS1307/DS1308 are not exposed to environmental stresses, such as burn–in, that some industrial applications require. Products which have successfully passed through this series of environmental stresses are marked IND or N, denoting their extended operating temperature and reliability rating. For specific reliability information on this product, please contact the factory at (972) 371–4448.

RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	1
Logic 1	V_{IH}	2.2		V_{CC}+0.3	V	1
Logic 0	V_{IL}	–0.3		+0.8	V	1
V_{BAT} Battery Voltage	V_{BAT}	2.0		3.5	V	1

DC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; V_{CC}=4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage	I_{LI}			1	µA	10
I/O Leakage	I_{LO}			1	µA	11
Logic 0 Output	V_{OL}			0.4	V	2
Active Supply Current	I_{CCA}			1.5	mA	9
Standby Current	I_{CCS}			200	µA	3
Battery Current (OSC ON); SQW/OUT OFF	I_{BAT1}		300	500	nA	4
Battery Current (OSC ON); SQW/OUT ON (32 KHz)	I_{BAT2}		480	800	nA	4

AC ELECTRICAL CHARACTERISTICS

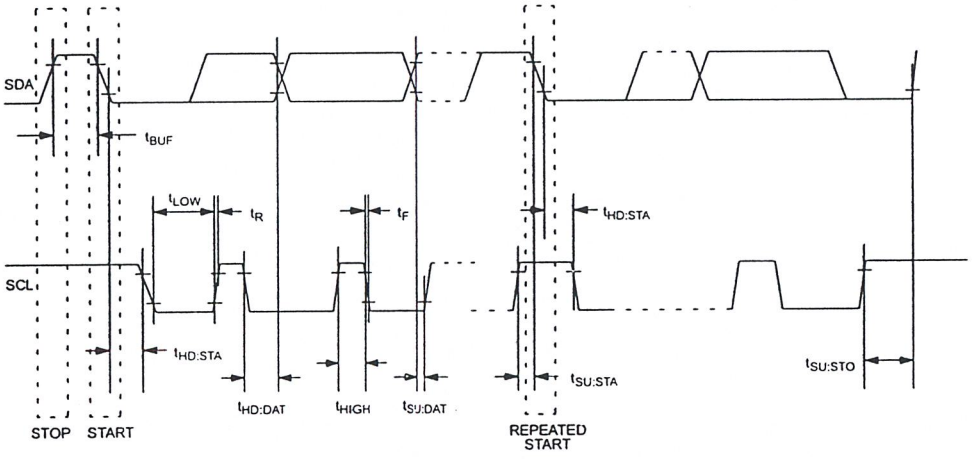
(0°C to 70°C; V_{CC}=4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	kHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			µs	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			µs	5
LOW Period of SCL Clock	t_{LOW}	4.7			µs	
HIGH Period of SCL Clock	t_{HIGH}	4.0			µs	
Set–up Time for a Repeated START Condition	$t_{SU:STA}$	4.7			µs	
Data Hold Time	$t_{HD:DAT}$	0			µs	6, 7
Data Set–up Time	$t_{SU:DAT}$	250			ns	
Rise Time of Both SDA and SCL Signals	t_R			1000	ns	
Fall Time of Both SDA and SCL Signals	t_F			300	ns	
Set–up Time for STOP Condition	$t_{SU:STO}$	4.7			µs	
Capacitive Load for each Bus Line	C_B			400	pF	8
I/O Capacitance	$C_{I/O}$		10		pF	
Crystal Specified Load Capacitance			12.5		pF	

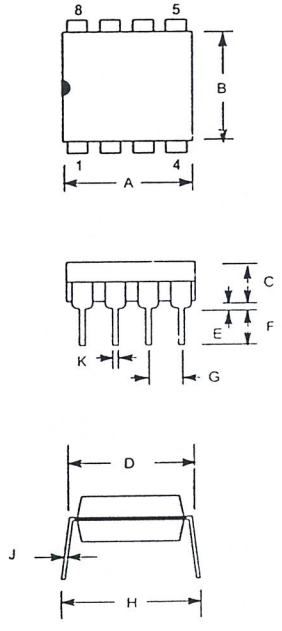
NOTES:

1. All voltages are referenced to ground.
2. Logic zero voltages are specified at a sink current of 5 mA at V_{CC}=4.5V, V_{OL}=GND for capacitive loads.
3. I_{CCS} specified with V_{CC}=5.0V and SDA, SCL=5.0V.
4. V_{CC}=0V, V_{BAT}=3V.
5. After this period, the first clock pulse is generated.
6. A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHMIN} of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
7. The maximum $t_{HD:DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.
8. C_B – total capacitance of one bus line in pF.
9. I_{CCA} – SCL clocking at max frequency = 100 kHz.
10. SCL only.
11. SDA and SQW/OUT
12. The DS1308 is designed to be subjected to no more than two (2) passes through a solder reflow process to limit premature crystal aging effects and maintain a reasonable accuracy of ±2 minutes/month at 25 degrees C (Worst case).

TIMING DIAGRAM Figure 8

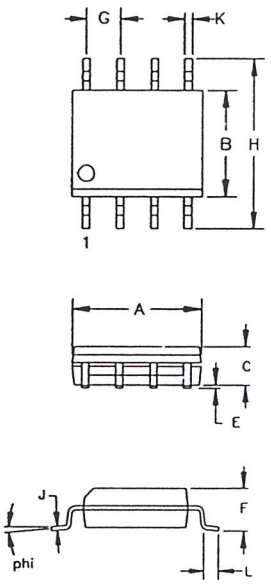


DS1307 64 X 8 SERIAL REAL TIME CLOCK 8–PIN DIP MECHANICAL DIMENSIONS



PKG	8–PIN	
DIM	MIN	MAX
A IN. MM	0.360 9.14	0.400 10.16
B IN. MM	0.240 6.10	0.260 6.60
C IN. MM	0.120 3.05	0.140 3.56
D IN. MM	0.300 7.62	0.325 8.26
E IN. MM	0.015 0.38	0.040 1.02
F IN. MM	0.120 3.04	0.140 3.56
G IN. MM	0.090 2.29	0.110 2.79
H IN. MM	0.320 8.13	0.370 9.40
J IN. MM	0.008 0.20	0.012 0.30
K IN. MM	0.015 0.38	0.021 0.53

DS1307Z 64 X 8 SERIAL REAL TIME CLOCK 8–PIN SOIC (150 MIL) MECHANICAL DIMENSIONS

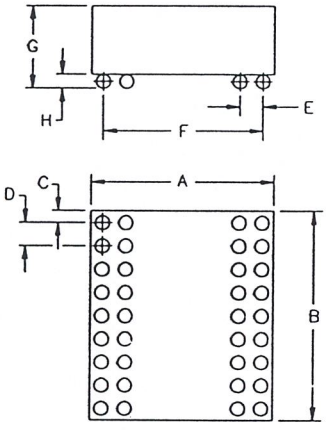


PKG	8–PIN (150 MIL)	
DIM	MIN	MAX
A IN. MM	0.188 4.78	0.196 4.98
B IN. MM	0.150 3.81	0.158 4.01
C IN. MM	0.048 1.22	0.062 1.57
E IN. MM	0.004 0.10	0.010 0.25
F IN. MM	0.053 1.35	0.069 1.75
G IN. MM	0.050 BSC 1.27 BSC	
H IN. MM	0.230 5.84	0.244 6.20
J IN. MM	0.007 0.18	0.011 0.28
K IN. MM	0.012 0.30	0.020 0.51
L IN. MM	0.016 0.41	0.050 1.27
phi	0°	8°

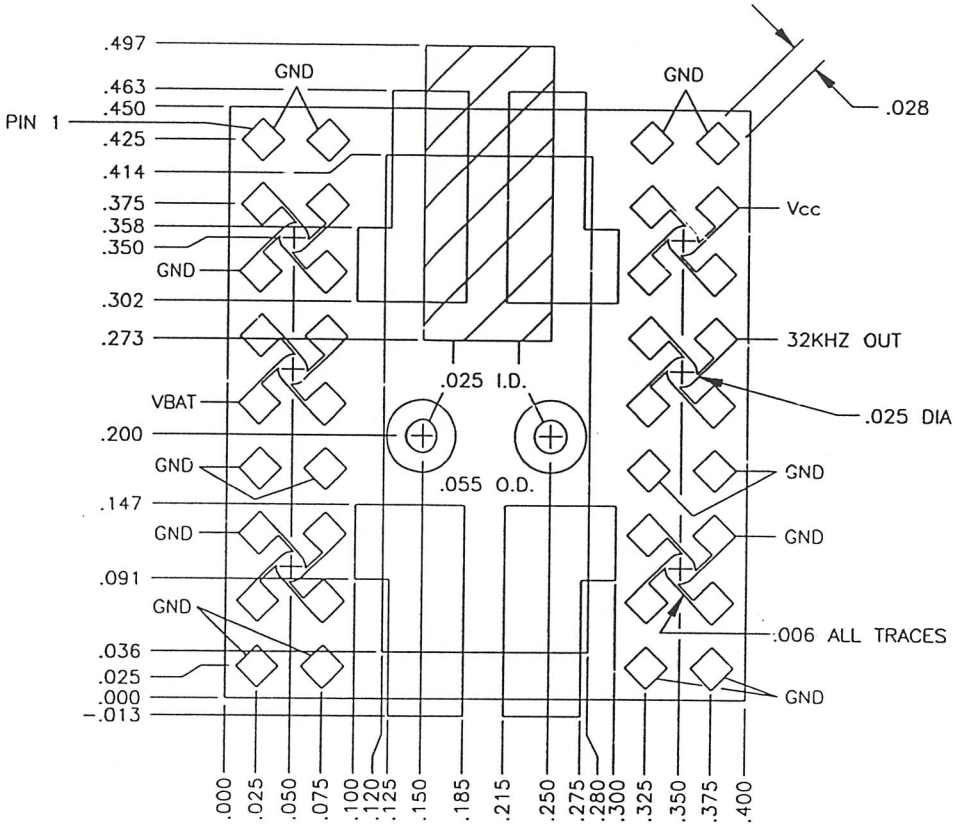
56–G2008–001

DS1308 MECHANICAL DIMENSIONS

PKG	36–PIN BALL GRID	
DIM	MIN	MAX
A IN. MM	0.395	0.405
B IN. MM	0.445	0.455
C IN. MM	0.022	0.028
D IN. MM	0.047	0.053
E IN. MM	0.047	0.053
F IN. MM	0.347	0.353
G IN. MM	0.170	0.190
H IN. MM	0.025	0.030



DS1308 RECOMMENDED LAYOUT LAND PATTERN

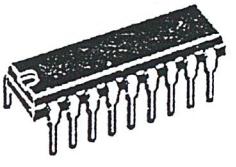


SGS-THOMSON MICROELECTRONICS

ULN2801A ULN2802A - ULN2803A ULN2804A - ULN2805A

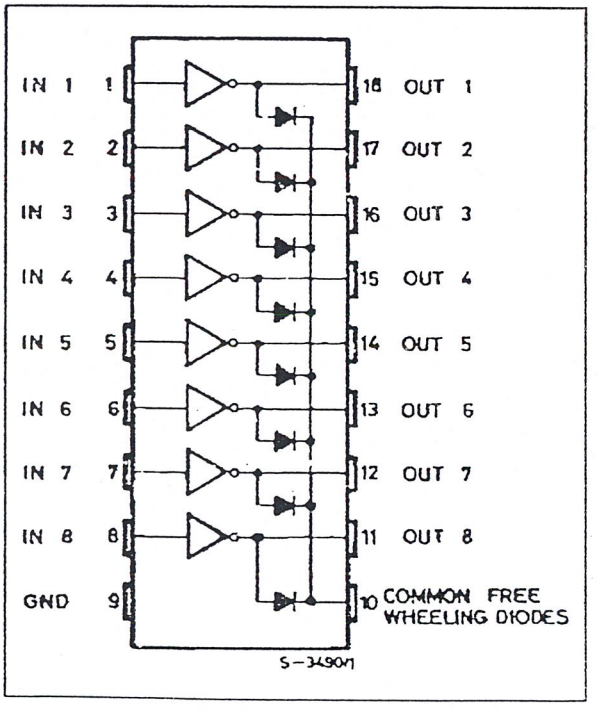
EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



DIP18

PIN CONNECTION (top view)



DESCRIPTION

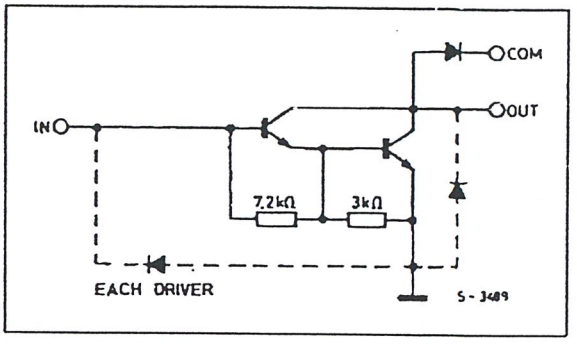
The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

Five versions are available to simplify interfacing to standard logic families : the ULN2801A is designed for general purpose applications with a current limit resistor; the ULN2802A has a 10.5kΩ input resistor and zener for 14-25V PMOS ; the ULN2803A has a 2.7kΩ input resistor for 5V TTL and CMOS ; the ULN2804A has a 10.5kΩ input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

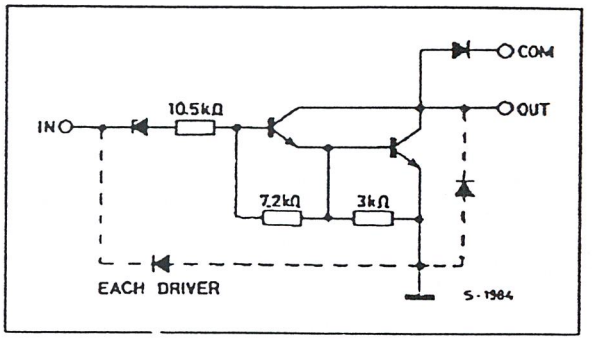
All types are supplied in a 18-lead plastic DIP with a copper lead frame and feature the convenient input-opposite-output pinout to simplify board layout.

SCHEMATIC DIAGRAM AND ORDER CODES

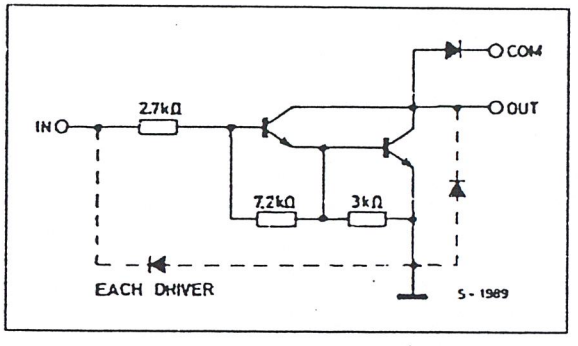
For ULN2801A (each driver for PMOS-CMOS)



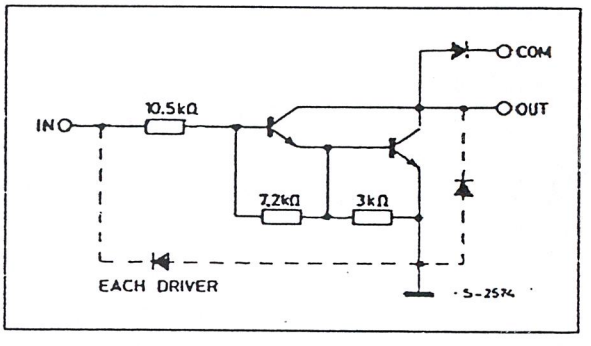
For ULN2802A (each driver for 14-15 V PMOS)



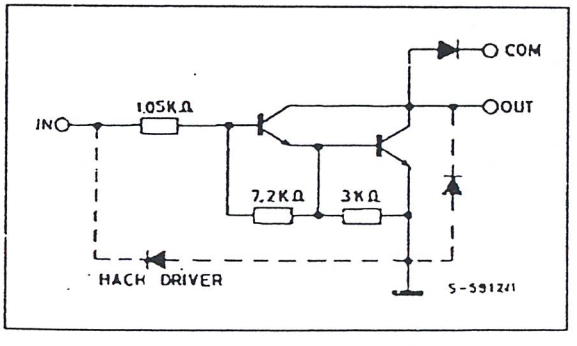
For ULN2803A (each driver for 5 V, TTL/CMOS)



For ULN2804A (each driver for 6-15 V CMOS/PMOS



For ULN2805A (each driver for high out TTL)



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_i	Input Voltage for ULN2802A, UL2803A, ULN2804A for ULN2805A	 30 15	V
I_C	Continuous Collector Current	500	mA
I_B	Continuous Base Current	25	mA
P_{tot}	Power Dissipation (one Darlingtcn pair) (total package)	 1.0 2.25	W
T_{amb}	Operating Ambient Temperature Range	– 20 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to 150	°C
Tj	Junction Temperature Range	– 20 to 150	°C

THERMAL DATA

Symbol	Parameter		Value	Unit
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	55	°C/W

ELECTRICAL CHARACTERISTICS (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V			50	µA	1a
		T_{amb} = 70°C, V_{CE} = 50V			100	µA	1a
		T_{amb} = 70°C for ULN2802A V_{CE} = 50V, V_i = 6V			500	µA	1b
		for ULN2804A V_{CE} = 50V, V_i = 1V			500	µA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250µA		0.9	1.1	V	2
		I_C = 200mA, I_B = 350µA		1.1	1.3	V	
		I_C = 350mA, I_B = 500µA		1.3	1.6	V	
$I_{i(on)}$	Input Current	for ULN2802A V_i = 17V		0.82	1.25	mA	3
		for ULN2803A V_i = 3.85V		0.93	1.35	mA	
		for ULN2804A V_i = 5V		0.35	0.5	mA	
		V_i = 12V		1	1.45	mA	
		for ULN2805A V_i = 3V		1.5	2.4	mA	
$I_{i(off)}$	Input Current	T_{amb} = 70°C, I_C = 500µA	50	65		µA	4
$V_{i(on)}$	Input Voltage	V_{CE} = 2 V for ULN2802A I_C = 300mA			13	V	5
		for ULN2803A I_C = 200mA			2.4	V	
		I_C = 250mA			2.7	V	
		I_C = 300mA			3	V	
		for ULN2804A I_C = 125mA			5	V	
		I_C = 200mA			6	V	
		I_C = 275mA			7	V	
		I_C = 350mA			8	V	
		for ULN2805A I_C = 350mA			2.4	V	
h_{FE}	DC Forward Current Gain	for ULN2801A V_{CE} = 2V, I_C = 350mA	1000			–	2
C_i	Input Capacitance			15	25	pF	–
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	–
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	–
I_R	Clamp Diode Leakage Current	V_R = 50V			50	µA	6
		T_{amb} = 70°C, V_R = 50V			100	µA	6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

Figure 8 : Collector Current as a Function of Saturation Voltage.

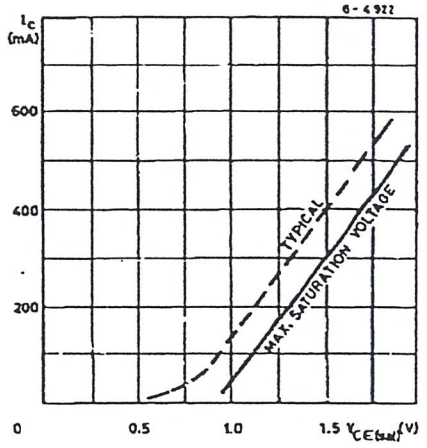


Figure 9 : Collector Current as a Function of Input Current.

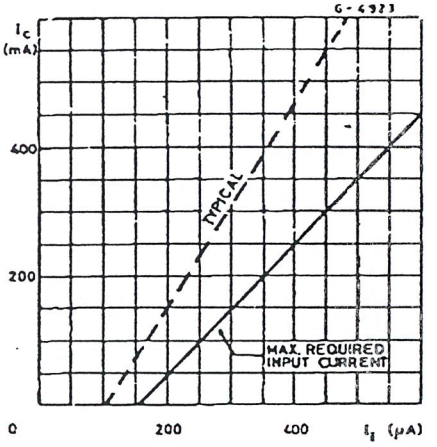


Figure 10 : Allowable Average Power Dissipation as a Function of Ambient Temperature.

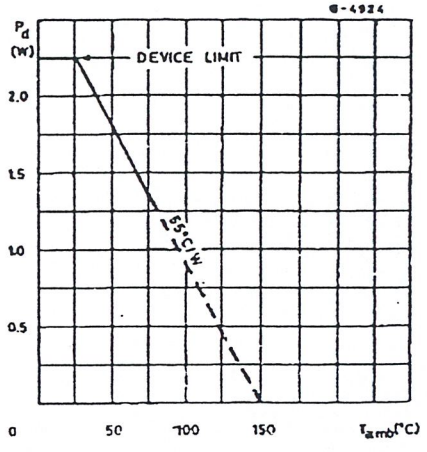


Figure 11 : Peak Collector Current as a Function of Duty Cycle.

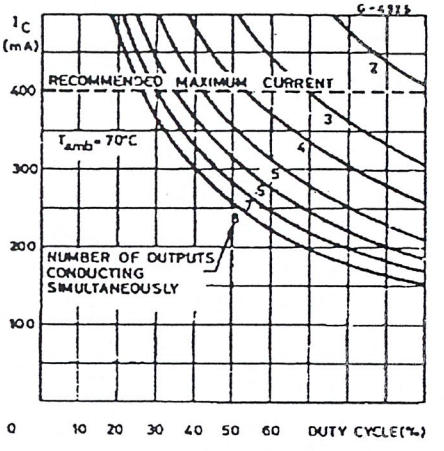


Figure 12 : Peak Collector Current as a Function of Duty.

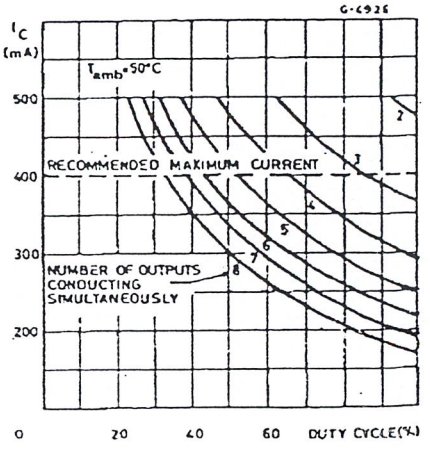


Figure 13 : Input Current as a Function of Input Voltage (for ULN2802A).

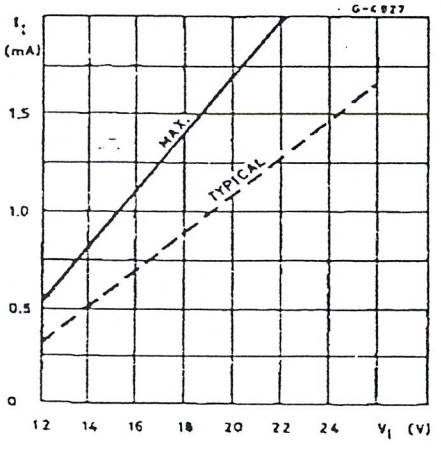


Figure 14 : Input Current as a Function of Input Voltage (for ULN2804A)

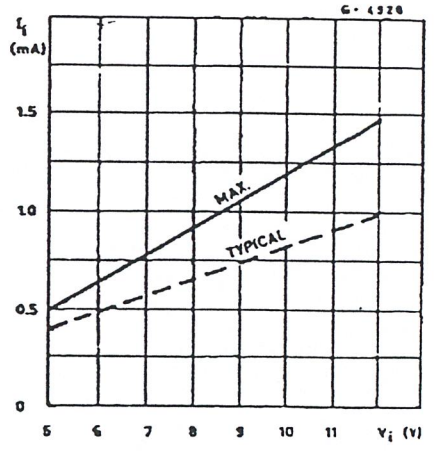


Figure 15 : Input Current as a Function of Input Voltage (for ULN2803A)

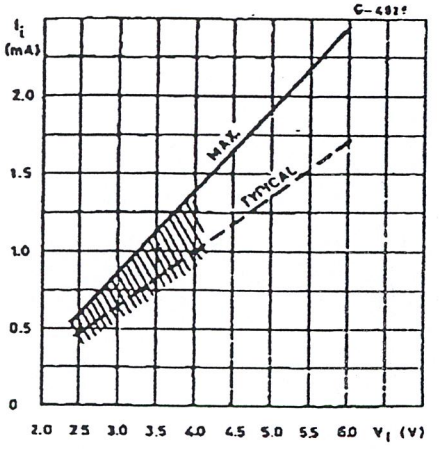
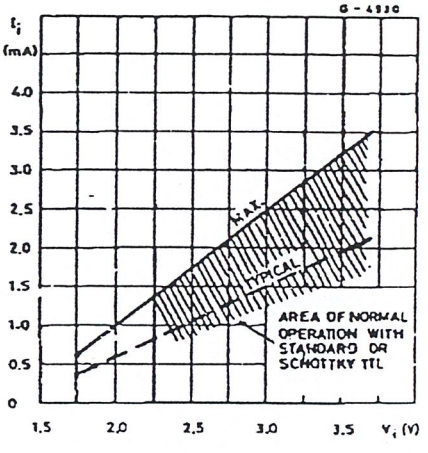
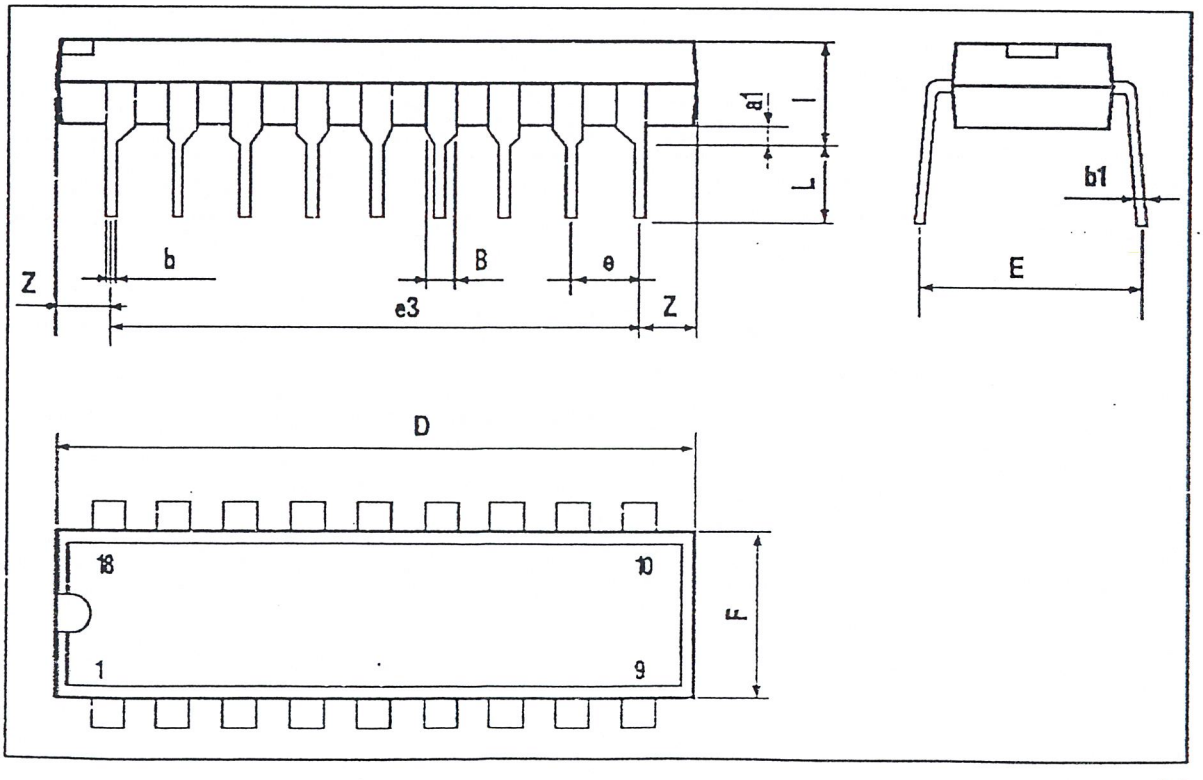


Figure 16 : Input Current as a Function of Input Voltage (for ULN2805A)



DIP18 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.254			0.010		
B	1.39		1.65	0.055		0.065
b		0.46			0.018	
b1		0.25			0.010	
D			23.24			0.915
E		8.5			0.335	
e		2.54			0.100	
e3		20.32			0.800	
F			7.1			0.280
I			3.93			0.155
L		3.3			0.130	
Z		1.27	1.59		0.050	0.063



TEST CIRCUITS

Figure 1a.

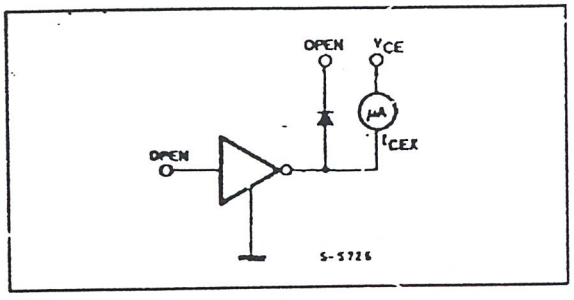


Figure 1b.

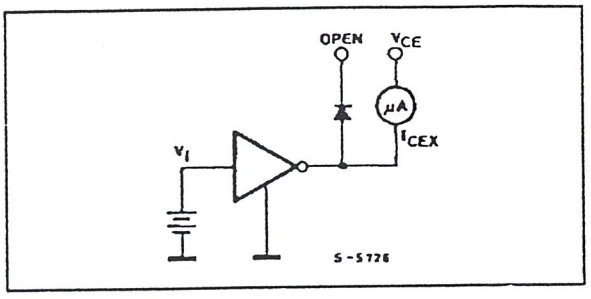


Figure 2.

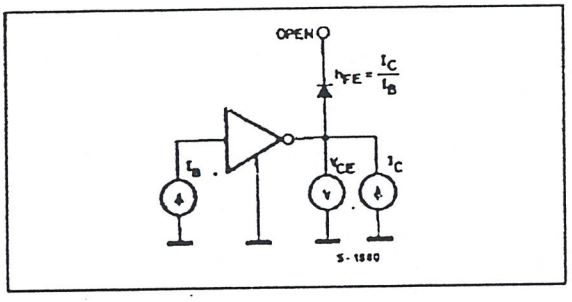


Figure 3.

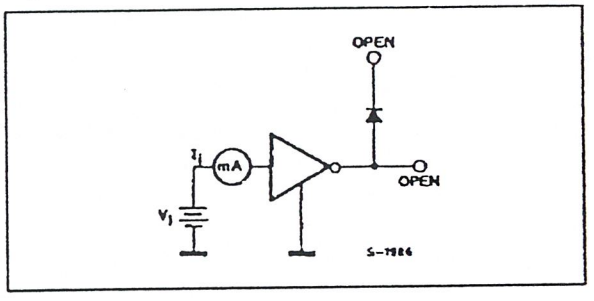


Figure 4.

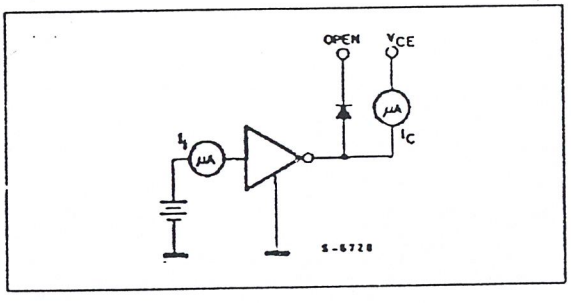


Figure 5.

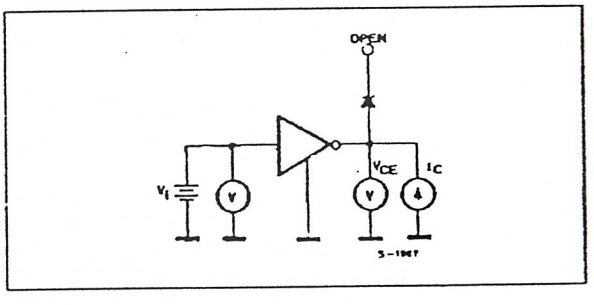


Figure 6.

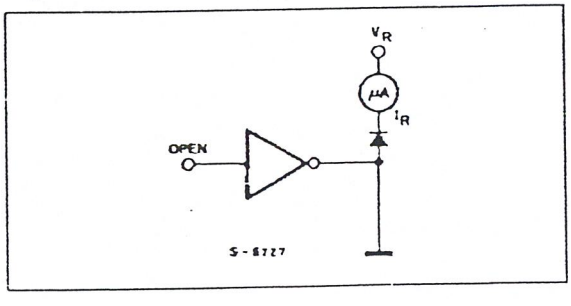


Figure 7.

