สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมปิด_เปิดไฟอัตโนมัติ

T033926

นายต่อตระกูล ใจสุข 39013272
นายสมคิด ไชยยศ 39013291

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน.....33926
วัน, เดือน, ปี 20 ก.ย. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATION ELECTRIC CONTROL DEVICE

Mr . Totragool Jaisook 39013272

Mr . Somkit Chaiyos 39013291

Project Report Submitted in Partial Fulfillment of the Requirement

for the Bachelor 's Degree

Department of Industrial Technology

Faculty of Engineering

King Mongkut 's Institute of Technology Ladkrabang

1998

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องควบคุมปิด-เปิดไฟอัตโนมัติ
ชื่อนักศึกษา	นายต่อตระกูล ใจสุข
	นายสมคิด ไชยยศ
อาจารย์ที่ปรึกษา	ผศ.ประดิษฐ์ วัชรพิบูลย์
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2541

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

.....หัวหน้าภาควิชาเทคนิคอุตสาหกรรม

( )

คณะกรรมการสอบ

.....ประธานคณะกรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Report	Automation Electric contron divice
By	Mr . Totrogool Jaisook
	Mr . Somkit Chaiyos
Department	Industrial Technology
Project Report Advisor	Asst.Prof . Pradit Watcharapiboon

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

. . . . . Chairman
( )

Project Report Committee

. . . . . Member
( )

. . . . . Member
( )

. . . . . Member
( )

. . . . . Member
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องควบคุมปิด-เปิดไฟอัตโนมัติ
ชื่อนักศึกษา	นายต่อตระกูล ใจสุข
	นายสมคิด ไชยยศ
อาจารย์ที่ปรึกษา	ผศ.ประดิษดิ์ วัชรพิบูลย์
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2541

บทคัดย่อ

ขณะที่เรานอนอยู่แล้วต้องตื่นแต่เช้ามืด แสงสว่างเป็นสิ่งจำเป็น เราต้องเดินไปเปิดไฟเพื่อทำกิจกรรมต่างๆ แต่บ่อยครั้งที่เราต้องเดินชนกับสิ่งของต่างๆที่วางไม่เป็นระเบียบ แต่ถ้าหากเราตื่นมาโดยไม่ต้องเดินไปเปิดไฟและขณะที่นอนอยู่ก็ไม่มีแสงไฟมารบกวนก็น่าจะดีทีเดียว

โครงงานนี้เป็นเครื่องควบคุมการ ปิด-เปิด ไฟฟ้าอัตโนมัติ ควบคุมการทำงานด้วย Microcontroller สามารถตั้งเวลา ปิด-เปิด ได้โดยผ่านทาง Keyboard สามารถควบคุมอุปกรณ์ได้ถึง 4 อุปกรณ์ ไม่จำเป็นต้องเป็นไฟฟ้าแสงสว่างอย่างเดียว แต่สามารถใช้ได้กับเครื่องใช้ไฟฟ้าทุกชนิดที่ใช้ไฟฟ้า 220 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Report Title	AUTOMATION ELELTRIC CONTROL DEVICE
Name	Mr . Totragool Jaisook
	Mr . Somkit Chaiyos
Project Report Advisor	Asst.Prof . Pratid Watcharapiboon
Department	Industrial Technology
Academic Year	1998

Abstract

When you are sleeping and to wake up in dawn. The light are neccessaries. You need it for one's daily tasks but many time you may be to collide with some thing. This problem can be to mend

This project is automation electric control device. Controled by micro controler MCS51 family. It can be control device with time. You are who set time. You can applied to use with any device on 220Vac

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

เนื่องจากเครื่องควบคุมปิด-เปิดไฟอัตโนมัติ นี้ประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ฮาร์ดแวร์ และ ซอฟท์แวร์ จึงจำเป็นต้องใช้ความรู้และการค้นคว้าอย่างมาก ซึ่งความซับซ้อนทางกลวิธีทางการโปรแกรมนั้นมากเป็นพิเศษ ซึ่งคณะผู้จัดทำมีความรู้ในด้านนี้น้อยมาก ทั้งนี้ผู้จัดทำได้ความอนุเคราะห์ด้านคำปรึกษาชี้แนะในแนวทางที่ถูกต้องของอาจารย์ผู้มีพระคุณ ซึ่งนอกจากจะได้รับการสั่งสอน ถ่ายทอดความรู้แล้ว ยังได้รับความห่วงใยต่อคณะผู้จัดทำฉันท์ศิษย์และอาจารย์อีกด้วย

ฉะนั้น ในโอกาสอันเหมาะสมนี้ คณะผู้จัดทำขอได้กล่าวคำขอบพระคุณในความรู้ความห่วงใย และปรารถนาดี ที่คณะผู้จัดทำได้รับ จาก ท่านอาจารย์ ประดิษดิ์ วัชรพิบูลย์ ณ. ที่นี้ด้วยความลำนึกในพระคุณอย่างสูง โดยส่วนดีของปริญญานิพนธ์นี้ที่อาจได้มีโอกาสเผยแพร่ ไปยังผู้แสวงหาความรู้ในด้านนี้ คณะผู้จัดทำขอมอบให้เป็นเกียรติต่อ ท่านอาจารย์ และคณะผู้จัดทำยังขอกล่าวคำขอบคุณอย่างจริงใจต่อบุคคลที่มีส่วนช่วยเหลือต่อผลงานชิ้นนี้ และขออภัยที่มิได้เอ่ยนาม ณ. ที่นี้ ส่วนข้อผิดพลาดหรือแนวทางที่ผิดนั้น คณะผู้จัดทำขอน้อมรับไว้เพื่อการแก้ไขต่อไปในอนาคต

คณะผู้จัดทำ

26 มีนาคม 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทคัดย่อ		ก
กิตติกรรมประกาศ		ข
สารบัญ		ค
บทที่ 1	บทนำ	1
1.1	วัตถุประสงค์	1
1.2	ขอบเขตของโครงงาน	1
1.3	แนวโน้มในการประยุกต์ใช้งานอื่น	2
บทที่ 2	ทฤษฎีและการใช้งาน 8255 เบื้องต้น	3
2.1	ลักษณะเบื้องต้น	3
2.2	หน้าที่ของขาต่างๆ	4
2.3	การใช้งานของ 8255	5
บทที่ 3	ทฤษฎีของไมโครคอนโทรลเลอร์ AT89C52	6
3.1	สถาปัตยกรรมของ 89C52	8
3.2	การทำงานของ 89C52	17
3.3	ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ	21
3.4	การรีเซ็ท	25
3.5	รีจีสเตอร์ของ 89C52	26
3.6	พื้นที่หน่วยความจำแฝงของ AT89C52	30
3.7	รีจิสเตอร์ฟังก์ชั่นพิเศษ (SFR)	30
3.8	การรับ-ส่งข้อมูลทางพอร์ทอนุกรม	52
3.9	การขัดจังหวะ	55
3.10	เมโมรีแม็พไอโอเทคนิค	57
บทที่ 4	ส่วนประกอบของโครงงาน	58
4.1	ส่วนประกอบสำคัญ	58
4.2	คุณลักษณะของเครื่อง	62
4.3	วงจรและการทำงาน	62
4.4	วิธีการใช้งาน	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 สรุปผลและวิจารณ์ 65
ภาคผนวก ก. ชุดคำสั่งของโครงงาน และการใช้งาน
ภาคผนวก ข. ข้อมูลของอุปกรณ์ที่ใช้ และข้อมูลทางไฟฟ้าของ IC TTL
บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

< Automation Electric control device >

บทนำ

ในปัจจุบันวัฒนาการของโลกก้าวไปไกลไม่หยุดยั้ง อุปกรณ์ไฟฟ้าต่างๆได้เข้ามามีบทบาทสำคัญต่อการดำรงชีวิตประจำวันของมนุษย์อย่างมากมาย อาทิเช่น แอร์ ทีวี เครื่องซักผ้า ระบบไฟฟ้า วิทยุ และอื่นๆ อีกมากมาย ซึ่งในอดีตการควบคุมอุปกรณ์เหล่านี้ยังต้องใช้ตัวคนทำควบคุม ซึ่งอุปกรณ์บางชนิด ซึ่งทำงานเป็นประจำ เราจึงต้องคอยควบคุมอยู่เป็นประจำหรือแม้แต่ในบางครั้งเราต้องการควบคุมเหล่านั้นตามเวลาที่กำหนด เราก็ต้องคอยควบคุมด้วยคน

แต่ในปัจจุบันนี้เราสามารถควบคุมอุปกรณ์เหล่านั้นได้ โดยไม่ต้องใช้คนเป็นคนควบคุม โดยเครื่องควบคุมการปิด-เปิดไฟอัตโนมัติ (Electric Control Device) ซึ่งมันจะคอยอำนวยความสะดวก และเป็นประโยชน์อย่างยิ่งในการควบคุมอุปกรณ์เหล่านี้

เครื่องควบคุมการปิด-เปิดไฟอัตโนมัติ เป็นอุปกรณ์ที่ใช้งานง่ายการควบคุมไม่ซับซ้อน มีขนาดเล็กสามารถเคลื่อนย้ายได้สะดวก ระบบการติดตั้งง่าย ผู้ใช้งานไม่ต้องมีความรู้มากก็สามารถใช้งานได้ ดังนั้นจึงเหมาะกับทุกสภาพการใช้งาน

เครื่องควบคุมการปิด-เปิดไฟอัตโนมัตินี้ เป็นการนำเอาวงจรที่ใช้ไมโครคอนโทรลเลอร์ (Micro controller) เป็นหัวใจสำคัญในการประมวลผลหรือเป็นตัวควบคุมอุปกรณ์ไฟฟ้าต่างๆ

1.1 วัตถุประสงค์

1. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ AT89C52 และสามารถนำไปประยุกต์ใช้งานร่วมกับ Chip Support
2. เพื่อนำความรู้ทางด้านทฤษฎีมาประยุกต์ใช้งานตามทฤษฎีได้จริง
3. เพื่อให้สามารถนำอุปกรณ์ทางอิเล็กทรอนิกส์ ทำ พัฒนาให้มีประสิทธิภาพ และคุณค่าสูงสุด
4. เพื่อสร้างเครื่องควบคุมทางอิเล็กทรอนิกส์ ที่สามารถนำไปใช้ควบคุมอุปกรณ์ไฟฟ้าได้

1.2 ขอบเขตของโครงงาน

1. ออกแบบวงจร เครื่องควบคุมอุปกรณ์การปิด-เปิดไฟอัตโนมัติ
2. สร้างโปรแกรมแปลภาษาบูลลีนขั้นพื้นฐาน ให้เป็นภาษาแอสเซ็มบลีของไมโครคอนโทรลเลอร์ AT89C51
3. ทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 แนวโน้มในการประยุกต์ใช้งาน

จากการศึกษาวงจรควบคุมการทำงานของอุปกรณ์ไฟฟ้า สามารถนำไปควบคุมอุปกรณ์ไฟฟ้าต่างๆ ได้ ซึ่งในส่วนของ ฐานเวลามาตรฐานยังไม่สามารถผ่าน Y2K ดังนั้นควรจะมีการพัฒนาเรื่องของฐานเวลา

ซึ่งจากวงจรปุ่ม key จะเหมือนการ key telephone ดังนั้นสามารถนำไปพัฒนาเพื่อควบคุมโทรศัพท์ต่อไป

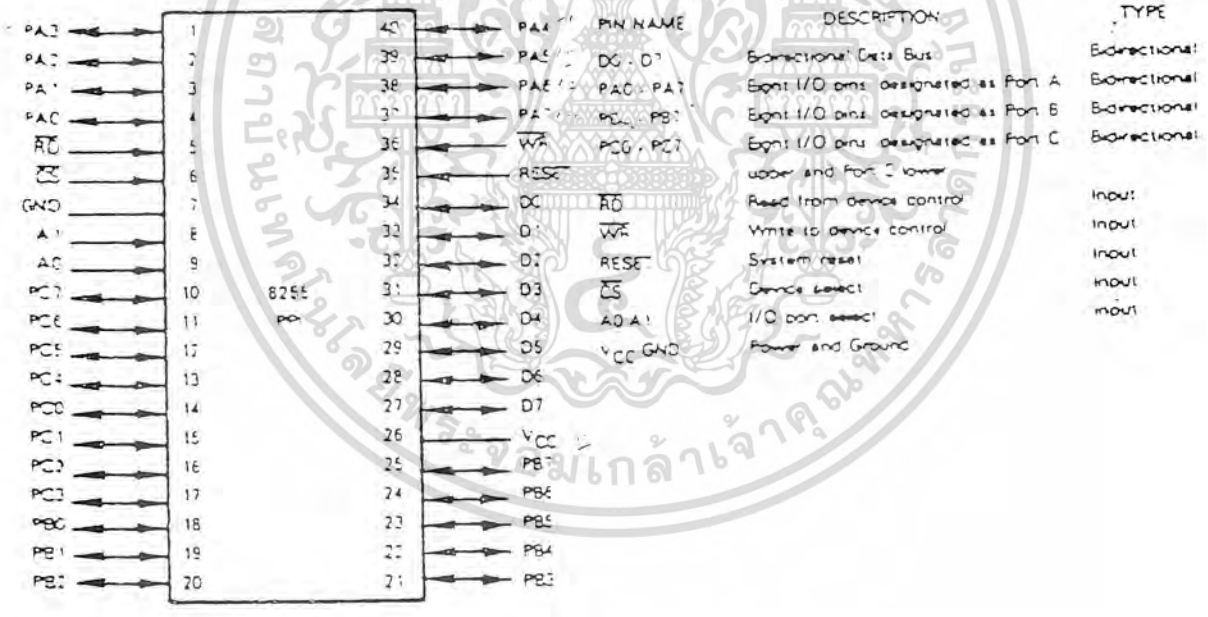
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและการใช้งาน 8255 เบื้องต้น

2.1 ลักษณะเบื้องต้น

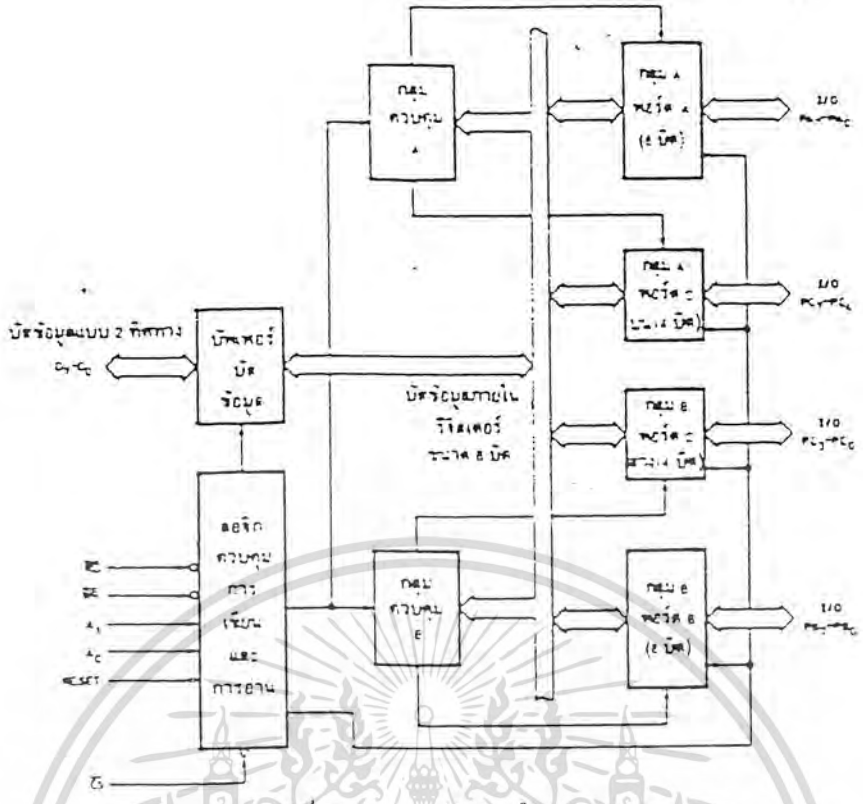
8255 นั้นเป็นไอซี LSI ขนาด 40 ขา ซึ่งทำหน้าที่ขยาย port จากรูปที่ 2.1 แสดงตำแหน่งของขาต่างๆทั้ง 4 ขา

ส่วนรูปที่ 2.2 แสดงแผนผังภายในของ 8255 ซึ่ง 8255 นี้มีพอร์ตสำหรับรับส่งข้อมูลอยู่ด้วยกัน 3 พอร์ต มีชื่อดังนี้คือ A , B และ C โดยมีพอร์ต C นี้จะแบ่งออกเป็น 2 ส่วนคือ พอร์ต C ล่าง (CLO) กับ C บน (CHI) นอกจากนี้แล้วยังมีพอร์ทอีกพอร์ทหนึ่งซึ่งทำหน้าที่ควบคุมการทำงานของพอร์ท A ,B และ C โดยการรับคำสั่งมาจาก CPU พอร์ตนี้เราเรียกว่าพอร์ตควบคุม (Control port) พอร์ตนี้จะใช้งานก็ต่อเมื่อ CPU ต้องการกำหนดลักษณะการทำงานของพอร์ต A ,B และ C หรือต้องการเปลี่ยนแปลงหลังจากที่กำหนดไว้เดิม CPU จะส่งรหัสควบคุมมาทางดาต้าบัส (Data Bus) ให้แก่พอร์ตควบคุมนี้



รูปที่ 2.1 ตำแหน่งขาต่างๆ ของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แผนผังภายในของ 8255

การกำหนดรหัสที่ใช้ในการควบคุมพอร์ตต่างๆ นี้จะกล่าวในตอนต่อไป ในทางปฏิบัติผู้ออกแบบระบบต้องนำรหัสควบคุมที่ได้มาตามข้อกำหนดของ 8255 นี้ไปใส่ในโปรแกรมเพื่อให้ CPU ทำการส่งรหัสควบคุมนี้มายังพอร์ตควบคุมเมื่อระบบนั้นเริ่มต้นทำงาน

2.2 หน้าที่ของขาต่างๆ

ก่อนที่จะกล่าวถึงการนำ 8255 ไปใช้งานควรทราบถึงหน้าที่ของขาต่างๆ ของ 8255 ทั้ง 40 ขาเสียก่อน จะทำให้เข้าใจถึงวิธีการใช้งานได้ดียิ่งขึ้น ขาต่างๆ ของ 8255 สามารถแบ่งออกได้ดังนี้

$\overline{CS}$ (Chip Select) ขานี้ใช้สำหรับรับสัญญาณจากภายนอกเพื่อใช้ในการเลือกว่าจะให้ 8255 ตัวนี้ทำงานหรือไม่ โดยแต่ถ้าขานี้ได้รับลอจิก "0" จะทำให้ 8255 เชื่อมต่อเข้าระบบบัสต่างๆ ของ CPU และพร้อมที่จะติดต่อกับ CPU ได้แต่เป็นลอจิก "1" มันก็จะปลดตัวเองออกจากระบบบัสของ CPU (โดยการเป็น Hi-z)

$\overline{RD}$ (Read Enable) เป็นขาอินพุตที่รับสัญญาณจาก CPU ถ้าขานี้ได้รับลอจิก "0" และขณะนั้นขา $\overline{CS}$ ต้องเป็นลอจิก "0" ด้วย 8255 จะทำการส่งข้อมูลจากพอร์ตที่ CPU ต้องการติดต่อด้วยนั้นได้แก่ CPU ทางดาต้าบัส

$\overline{WR}$ (Write Enable) มีหน้าที่การทำงานตรงกันข้าม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับขา $\overline{RD}$ คือถ้า $\overline{WR}$ นี้ได้รับลอจิก "0" ($\overline{CS}$ ต้องเป็น "0" ด้วยเช่นกัน) 8255 จะรับข้อมูลจากดาต้าบัสของ CPU ส่งออกไปยังพอร์ตที่ CPU ส่งออกไปยังพอร์ตที่ CPU กำหนดไว้

RESET คือ ขาที่ทำหน้าที่ RESET8255 เมื่อได้ที่ 8255 ได้รับสัญญาณ RESET มันจะกลับเข้าสู่โหมดอินพุตคือทุกๆ พอร์ตจะเป็นอินพุตพอร์ต ขา RESET นี้ใช้เมื่อต้องการเคลียร์สถานะต่างๆ ของ 8255

D_0-D_7 คือขาข้อมูลที่ใช้ในการติดต่อรับส่งข้อมูลกับ CPU โดยขา D_0-D_7 นี้จะต่อเข้ากับดาต้าบัสของ CPU เพื่อให้ CPU ส่งข้อมูลออกไปยังพอร์ตหรือรับข้อมูลจากพอร์ตส่งให้แก่ CPU ผ่านทาง D_0-D_7 นี้

A_0-A_1 คือขาแอดเดรสที่ใช้ในการเลือกพอร์ตที่ CPU ต้องการจะติดต่อด้วยซึ่งมีความเป็นไปได้ทั้งหมด 4 ค่า ดังนี้คือ

00 = พอร์ต A
01 = พอร์ต B
10 = พอร์ต C
11 = พอร์ตควบคุม

PA_0-PA_7 เป็นขาสัญญาณของพอร์ต A ใน 8255 ซึ่งจะถูกเลือกโดยค่าของ A_0-A_1 และเมื่อพอร์ตนี้ถูกเลือกใช้ข้อมูลต่างๆ ก็จะถูกส่งผ่าน PA_0-PA_7 นี้ไปยัง D_0-D_7 (กรณีที่ให้พอร์ต A นี้ เป็นอินพุตพอร์ต) หรือจาก D_0-D_7 มายัง D_0-D_7 (กรณีที่เป็นเอาท์พุตพอร์ต)

PB_0-PB_7 เป็นขาสัญญาณของพอร์ต B ซึ่งจะถูกเลือกโดยลอจิกที่ A_0-A_1 เช่น กันกับพอร์ต A และพอร์ต B นี้มีข้อจำกัดการรับส่งข้อมูลที่ต่างจากพอร์ต A ในบางกรณี

PC_0-PC_7 เป็นสายสัญญาณของพอร์ต C ซึ่งแบ่งออกเป็น 2 กลุ่ม คือ PC_0-PC_3 และ PC_4-PC_7 โดยแต่ละกลุ่มสามารถแยกกันทำงานได้โดยอิสระกลุ่มหนึ่งอาจจะเป็นอินพุตพอร์ตในขณะที่อีกกลุ่มหนึ่งเป็นเอาต์พุตพอร์ตได้ แต่จะทำงานพร้อมๆ กันโดยการเลือกด้วยลอจิกที่ A_0-A_1

2.3 การใช้งาน 8255

8255 นั้นแบ่งลักษณะการทำงานออกเป็น 3 โหมด (Mode) ด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โหมด 0 เป็นโหมดอินพุตหรือเอาต์พุตพอร์ตอย่างใดอย่างหนึ่ง ซึ่งทั้ง 3 พอร์ตคือ A , B และ C สามารถทำงานในโหมดนี้ได้

- โหมด1 เป็นโหมดอินพุตหรือเอาต์พุตพอร์ตอย่างใดอย่างหนึ่งเช่นกัน แต่จะมีลักษณะการทำงานเป็นลักษณะ handshaking ซึ่งจะกล่าวรายละเอียดในภายหลังในโหมดนี้กาทำงานได้เฉพาะพอร์ต A และ B

- โหมด 2 เป็นโหมด Bi-directional คือเป็นได้ทั้งอินพุตและเอาต์พุตพอร์ตในเวลาเดียวกันและทำงานแบบ handshaking เช่นเดียวกับโหมด 1 ในโหมดนี้ใช้ได้เฉพาะพอร์ต A เท่านั้น

การกำหนดโหมดการทำงานของ 8255 นั้นทำได้โดย CPU ทำการส่งรหัสควบคุมผ่านทางดาต้าบัสมายังพอร์ตควบคุม (Control port) ของ 8255 รหัสควบคุมนี้จะมีขนาด 1 ไบต์ เรียกว่า Control Byte และในแต่ละบิตของ Control Byte (1 Byte = 8 bit) นั้นจะมีความหมายเฉพาะของตัวเองดังแสดงในรูปที่ 2 ซึ่งจะอธิบายได้ดังนี้

บิต D_7 เป็นบิตที่แสดงว่า Byte นี้เป็นรหัสควบคุม (Control Byte) ที่จะมีผลต่อการกำหนดโหมดการทำงานของ 8255

บิต D_7 และ D_5 มีความหมายในการเลือกโหมดของพอร์ต A ซึ่งสามารถทำงานได้ทั้ง 3 โหมด โดยลอจิกที่ D_6 และ D_5 จะมีความหมายดังนี้

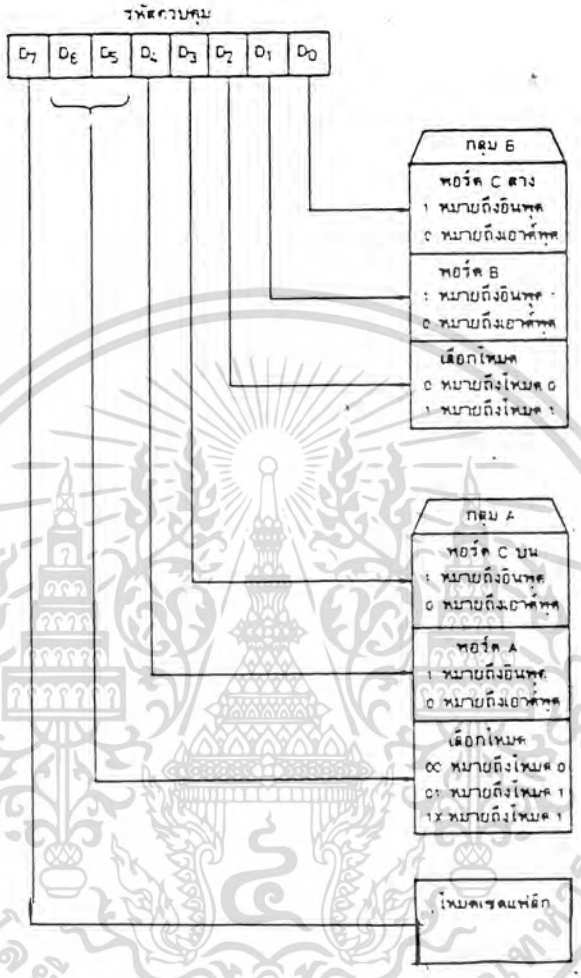
00 = โหมด 0
01 = โหมด 1
10 = โหมด 2
11 = โหมด 2

บิต D_4 ถ้าเป็นลอจิก "0" หมายถึงสั่งให้พอร์ต A ทำหน้าที่เป็นเอาต์พุตพอร์ต แต่ถ้าเป็นลอจิก "1" พอร์ต A จะเป็นอินพุตพอร์ต บิตนี้จะมีความหมายว่าเมื่อเราให้ 8255 ทำงานในโหมด 0 หรือโหมด 1 เท่านั้น เพราะในโหมดที่ 2 พอร์ต A จะเป็นอินพุต และเอาต์พุตพอร์ตในเวลาเดียวกัน

บิต D_3 เป็นบิตที่กำหนดการทำงานของพอร์ต C บน (PC_4-PC_7) ถ้าบิตนี้เป็นลอจิก "0" พอร์ต C บน นี้จะเป็นเอาต์พุตพอร์ต ถ้าเป็น "1" จะเป็นอินพุตพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต D_2 เป็นบิตที่ใช้สำหรับกำหนดโหมดการทำงานของพอร์ต B ถ้าเป็นลอจิก "0" หมายถึงให้พอร์ต B ทำงานในโหมด 0 ถ้าลอจิก "1" จะทำงานในโหมด 1



รูปที่ 2.3 แสดงความหมายของแต่ละบิตในรหัสควบคุม

บิต D_1 เป็นการกำหนดให้พอร์ต B เป็นอินพุตหรือเอาต์พุตพอร์ต ถ้า D_1 เป็นลอจิก "0" จะเป็นเอาต์พุตพอร์ต แต่ถ้าลอจิก "1" จะเป็นอินพุตพอร์ต

บิต D_0 เป็นบิตที่กำหนดการเป็นอินพุตหรือเอาต์พุตของพอร์ต C ล่าง (PC_0-PC_3) ถ้าบิตนี้เป็น "0" จะเป็นเอาต์พุตพอร์ต ถ้าเป็น "1" จะเป็นอินพุต

1

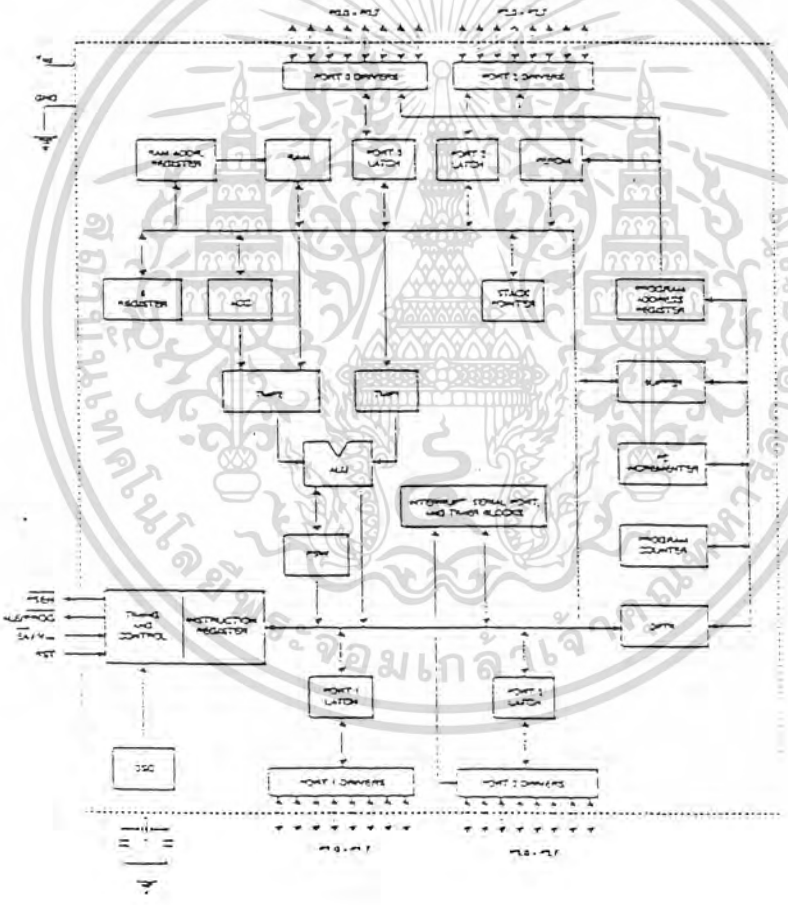
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ทฤษฎีของไมโครคอนโทรลเลอร์ 89C52

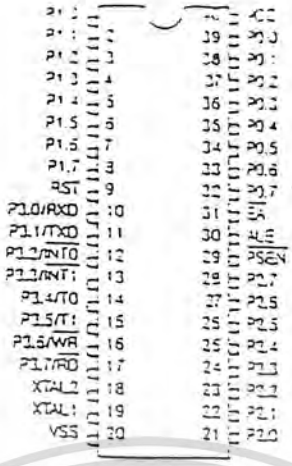
3.1 สถาปัตยกรรมของ 89C52

89C52 ใช้สถาปัตยกรรมเดียวกันกับ 8052 แต่ภายในของ 89C52 จะเปลี่ยนแปลงจากบล็อกที่เป็น ROM (Read Only Memory) ของ 8052 เป็นบล็อก PE ROM ซึ่งมีขนาด 8 Kbyte ฉะนั้นเราจึงขอกล่าวว่า 89C52 เป็นตระกูลเดียวกันกับ 8052 จากรูปที่ 3.1 จะเป็นสถาปัตยกรรมภายในของ 89C52 ซึ่งจะอธิบายถึงส่วนย่อย ๆ ของภายใน 89C52 เพียงชิพเดียว และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา (Pin) ของ 89C52 ที่มีอยู่ 40 ขา ดังรูปที่ 3.1



รูปที่ 3.1 สถาปัตยกรรมภายในของ 89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ไดอะแกรมขาของ 89C52 แบบ DIP

89C52 เป็นไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 89C52 มีขาอยู่ข้างละ 20 ขา รวมกันทั้งหมด 2 ข้าง ได้ 40 ขา มีการใช้งานต่าง ๆ กัน ดังต่อไปนี้

VCC (ที่ขา 40)

เป็นขาที่ต้องใช้ไฟเลี้ยง + 5 โวลท์ เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลเตจของลอจิก 0 และ 1 ของ 89C52 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

VSS (ที่ขา 20)

เป็นขาที่ต้องต่อกับกราวด์ (Ground)ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้อง มีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน

Port 0 (ที่ขา 32 - 39)

เป็นพอร์ทขนานขนาด 8 บิท อยู่ที่ขา 32 ถึง 32 เริ่มจากบิท 0 ถึงบิท 7 ตามลำดับ

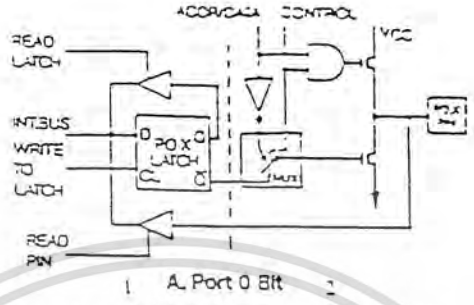
ดังในรูปที่ 3.2 แต่ละขาจะเขียนว่า P0.0 , P0.1 , ... , P0.7 P0.7 นั้น หมายถึงบิท 7 ของพอร์ท

0 ซึ่งเป็นบิทที่มีนัยสำคัญสูงสุด (Most Significant) และ P0.0 คือบิท 0 ของพอร์ท 0 เป็นบิท

ที่มีนัยสำคัญต่ำสุด (Least Significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับ-ส่งตำแหน่งและข้อมูลกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ หรือใช้เป็นพอร์ทรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ท 0 จะถูกแลตช์ (Latch) ไว้ที่ขาของพอร์ท โครงสร้างแต่ละบิทของพอร์ท 0 เป็น แบบ Open Drain Bidirec- tional ดังรูปที่ 3.3



รูปที่ 3.3 โครงสร้างของพอร์ท 0

ในรูปที่ 3.3 เมื่อเปรียบเทียบกับรูปที่ 3.1 ส่วนที่ 1 ของรูปที่ 3.3 ก็คือ Port 0 Latch ในรูปที่ 3.1 และส่วนที่ 2 ของรูปที่ 3.3 ก็คือ Port 0 Driver ของรูปที่ 3.1 นั่นเอง

จากโครงสร้างในรูปที่ 3.3 เมื่อมีคำสั่งของการเขียนข้อมูลมายังพอร์ท 0 ข้อมูลจาก Internal Data Bus จะถูก Latch ไว้ที่ D-FF โดยสัญญาณ " Write to Latch " ที่ถูกสร้างมาจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ท 0 จะอ่านได้ 2 แบบ คือ การอ่านข้อมูลที่ส่งไปเก็บไว้ที่พอร์ทก็จะมีสัญญาณ Read Latch มาเพื่ออ่านข้อมูลจาก D-FF กลับเข้าไปยัง Internal Data Bus การอ่านข้อมูลอีกแบบก็คือ การอ่านสภาวะของสัญญาณที่เข้ามาทางพอร์ท 0 ก็จะมีสัญญาณ Read Pin มาควบคุมการอ่าน พอร์ท 0 จะใช้งานหลายอย่างดังนี้

1. ใช้สำหรับส่งค่าตำแหน่ง หน่วยความจำภายนอกที่ต้องการติดต่อด้วยตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อได้ก็คือ 64 Kbyte จึงมีค่าตำแหน่งหน่วยความจำ 16 บิท ของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิทล่างจะถูกส่งออกไปทางพอร์ท 0 และ 8 บิทบนจะส่งออกไปทางพอร์ท 2

2. ใช้รับ-ส่ง ข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory

3. ใช้รับ-ส่ง ข้อมูลผ่านทางพอร์ทโดยตรง ในกรณีที่ไม่มีการใช้หน่วยความจำของ Program Memory หรือ Data Memory ภายนอก

วงจรภายในส่วน Timing and Control จะเป็นตัวสร้างสัญญาณมาควบคุมวงจรในรูปที่ 3.3 เพื่อให้การทำงานแต่ละอย่างข้างต้น เมื่อแต่ละบิทของพอร์ท 0 ทำงานตามข้อ 1 และ 2 ข้างต้น วงจร Timing and Control จะทำให้สภาวะลอจิกของขา Control เป็น 1 ซึ่งทำให้สวิทช์ MUX อยู่ในตำแหน่งข้างบน เมื่อพอร์ท 0 จะส่งข้อมูลซึ่งเป็นค่าตำแหน่ง

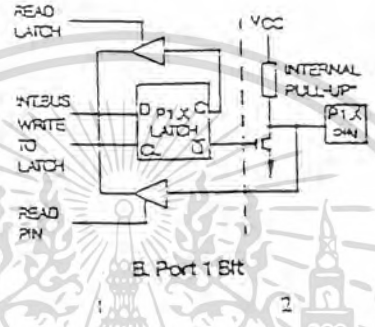
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช

หน่วยความจำ หรือข้อมูลที่จะเขียนออกไปยังหน่วยความจำภายนอก ก็จะส่งค่าดังกล่าวมายัง ADDR/DATA ถ้าข้อมูลที่ส่งมาเป็น 1 จะทำให้สัญญาณออกจาก AND GATE เป็น 1 และสัญญาณที่ออกจาก Inverter เป็น 0 ดังนั้น FET ตัวบน ON (สภาวะ ON ของ FET คือความต้านทานระหว่างขา D กับ S มีค่าต่ำมากเสมือนกับเป็นวงจรปิด) ส่วน FET ตัวล่าง OFF (สภาวะ OFF ของ FET คือความต้านทานระหว่างขา D กับ S มีค่าสูงมาก เสมือนกับเป็นวงจรเปิด) สภาวะลอจิกที่ขา P0.X PIN จะเป็น 1 แต่ถ้าข้อมูลที่ส่งออกมายัง ADDR/DATA เป็น 0 ก็จะทำให้สัญญาณจาก AND GATE เป็น 0 และสัญญาณที่ออกจาก Inverter เป็น 1 ดังนั้น FET ตัวบนจะ OFF ส่วน FETตัวล่างจะ ON ทำให้สภาวะลอจิกที่ขา P0.X PIN เป็น 0 เมื่อ 89C52 ต้องการใช้พอร์ท 0 สำหรับการอ่านข้อมูลจากหน่วยความจำภายนอก หรือใช้ทำงานในข้อ 3 ข้างบน ก็จะทำได้โดยวงจร Timing and Control ทำให้สภาวะลอจิกของสัญญาณ Control ในรูปเป็น 0 ทำให้เอาต์-พุตจาก AND GATE เป็น 0 FET ตัวบนจะ OFF และสวิตช์ MUX จะอยู่ในตำแหน่งข้างล่าง
ดังนั้น FET ตัวล่างจะ ON หรือ OFF ก็แล้วแต่ข้อมูลที่ขา Q ของ D-FF เมื่อมีการเขียนข้อมูลจาก Internal Data Bus มายัง D-FF ก็จะมีสัญญาณ Write to Latch มายัง D-FF ด้วย ถ้าข้อมูลที่เขียนมาเป็น 1 ก็จะทำให้ขา Q มีสภาวะลอจิกเป็น 0 ทำให้ FET ตัวล่าง OFF ดังนั้นขา P0.X จะอยู่ในสภาวะอิมพีแดนซ์สูง (High Impedance) เพราะ FET ทั้ง 2 ตัว OFF แต่ถ้าข้อมูลที่เขียนมายัง D-FF เป็น 0 จะทำให้ FET ตัวล่าง ON แต่ตัวบน OFF ทำให้สภาวะลอจิกที่ขา P0.X เป็น 1 ดังนั้น PORT 0 เมื่อให้ทำงานเป็นพอร์ทส่งข้อมูล (ไม่ใช่ส่งตำแหน่งหน่วยจำ) จะไม่สามารถแสดงสภาวะลอจิก 1 ได้จึงต้องต่อตัวต้านทาน Pull Up ไว้ภายนอก ระหว่างขา P0.X กับไฟเลี้ยงวงจร ถ้าจะใช้พอร์ท 0 สำหรับรับข้อมูลเข้าจะต้องเขียน 1 มาเก็บไว้ยัง D-FF เสียก่อนเพื่อให้ขา P0.X อยู่ในสภาวะ High Impedance แล้วจึงใช้คำสั่งอ่านสภาวะลอจิกเข้าไปยัง Internal Data Bus ต่อไป โดยคำสั่งอ่านสภาวะลอจิกทางพอร์ท 0 ก็จะทำให้วงจร Timing and Control สร้างสัญญาณ Read Pin สำหรับการอ่านสภาวะลอจิกข้างต้น ถ้าไม่เขียน 1 มาเก็บไว้ยัง D-FF ก่อนที่จะอ่านข้อมูลแล้วอาจมีข้อมูลค้างอยู่ที่ D-FF ทำให้ Q เป็น 0 และ Q เป็น 1 ซึ่งทำให้ FET ตัวล่าง ON สัญญาณที่ต่อเข้ามาที่ขา P0.X ไม่ว่าจะมีสภาวะลอจิกใดจะถูกดึงลงกราวด์ ดังนั้นเมื่ออ่านข้อมูลไปก็จะพบว่าเป็น 0 เสมอ ในการอ่านข้อมูลจากหน่วยความจำภายนอกนั้นวงจร Timing and Control ก็จะเขียนข้อมูลมายัง D-FF ให้เป็น 1 และสร้างสัญญาณ Control ให้มีลอจิกเป็น 0 ก่อนจะอ่านข้อมูลเข้าไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 1 (ที่ขา 1 - 8)

เป็นพอร์ทขนานขนาด 8 บิท ในรูปที่ 3.2 คือขา P1.0 ถึง P1.7 (ขา 1 - 8) P1.0 หมายถึงบิท 0 ของพอร์ท 1 ซึ่งเป็นบิท Least Significant Bit และบิท P1.7 หมายถึงบิทที่ 7 ของพอร์ท 1 ซึ่งเป็น Most Significant bit โครงสร้างของพอร์ท 1 แต่ละบิทมีดังรูปที่ 3.4



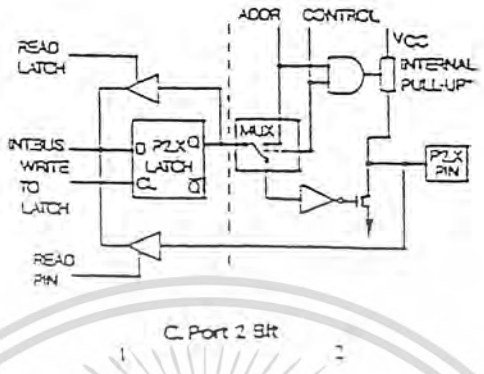
รูปที่ 3.4 โครงสร้างของพอร์ท 1

ส่วนที่ 1 คือ Port 1 Latch ในรูปที่ 3.1 ซึ่งจะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ท 0 ในรูปที่ 3.3 ส่วนที่ 2 คือ Port 1 Driver ในรูปที่ 3.1 Port 1 Driver นี้จะมีตัวต้าน-ทานต่ออยู่เป็น Internal Pull Up พอร์ท 1 นี้จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ท 1 จะถูก Latch ไว้แล้วส่งออกไปทางแต่ละขา ก่อนที่จะอ่านข้อมูลเข้าไปทาง พอร์ท 1 จะต้องเขียน 1 ไปยังทุกบิทของพอร์ท 1 เสียก่อน เพื่อให้ FET อยู่ในสภาวะ OFFก่อน มิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งออกมาค้างอยู่ที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่ขานี้ก็จะถูกลัดวงจรลงกราวด์ โดยไม่สนใจว่าสภาวะลอจิกของสัญญาณที่เข้ามาจะเป็นอะไร ข้อมูลที่อ่านเข้าไปจึงจะเป็น 0 เสมอ

Port 2 (ที่ขา 21 - 28)

พอร์ทขนานขนาด 8 บิท คือขา P2.0 ถึง P2.7 (บิท 0 ถึงบิท 7 ของพอร์ท 2) ในรูปที่ 3.3 โครงสร้างของพอร์ท 2 แต่ละบิทจะมีดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 โครงสร้างของพอร์ท 2

ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันที่ใน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะคือ

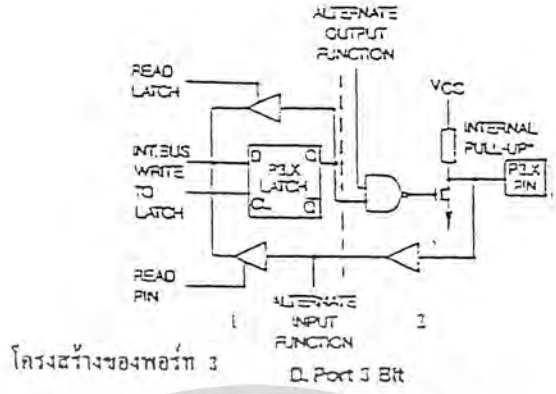
1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิท บนของค่าตำแหน่ง
2. ใช้เป็นพอร์ทรับ และส่งข้อมูลกับภายนอก

ดังนั้นภาค Driver ของพอร์ท 2 จึงแตกต่างจาก Driver ของพอร์ท 0 โดยที่ใน พอร์ท 2 นั้นจะมีเฉพาะ ADDR (ตำแหน่งหน่วยความจำ) เข้ามาที่ MUX (Multiplexer) เท่านั้น นอกนั้นแล้วการทำงานจะเหมือนกัน และที่เอาต์พุตของพอร์ท 2 จะมี Internal pull-up ซึ่งเป็นตัวต้านทาน และจะทำให้เอาต์พุตของพอร์ท 2 แสดงสถานะลอจิกเป็น 1 ได้ ถ้า FET อยู่ในสภาวะ OFF บางครั้งเรียกว่า " Quasi-bidirectional " เมื่อใช้เป็นพอร์ทอินพุตก็สามารถทำได้โดยการต่อสัญญาณภายนอกเข้ามาโดยตรง ถ้าสัญญาณภายนอกเป็น 0 ก็จะมีกระแสไหลออกจากพอร์ท (Source Current) ในการที่จะใช้พอร์ทนี้เป็นพอร์ทรับข้อมูลเข้า จะต้องเขียน 1 ไปยังแต่ละบิทของพอร์ทเสียก่อน ดังได้อธิบายในเรื่อง Port 0 และ Port 1

Port 3 (ที่ขา 10 - 17)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือขา P3.0 ถึง P3.7 หรือขา 10 - 17 ตามลำดับ ในรูปที่ 3.2 พอร์ทนี้มีโครงสร้างดังรูปที่ 3.6



รูปที่ 3.6 โครงสร้างของพอร์ท 3

ส่วนที่ 1 ในรูปที่ 3.6 เป็นส่วน Latch ข้อมูลที่เขียนมายังพอร์ท 3 ทาง Internal Bus เหมือนกับพอร์ทอื่น ๆ และพอร์ท 3 จะมี Internal pull up อยู่ทุกบิท แต่พอร์ท 3 นี้แต่ละบิทจะใช้การทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Alterna-tive Output Function เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ท 3 ทำงานในฟังค์ชั่นอื่น และจุด Alternative Output Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่นตามการทำงานของบิทนั้น

แต่ละบิทของพอร์ท 3 จะมีฟังก์ชั่นอื่นดังนี้

P3.0/ RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/ TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/ INT0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/ INT1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/ TO (Time / Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Time /Counter 0 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ TO นี้ หรือสัญญาณนาฬิกาก็ได้

P3.5/ T1 (Timer / Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร Time / Counter 1 ซึ่งมีการทำงานเหมือนกับ To

P3.6/ $\overline{WR}$ (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 89C52

P3.7/ $\overline{RD}$ (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RST (ที่ขา 9)

ขารีเซ็ทขานี้จะใช้ทำการรีเซ็ทการทำงานของ 89C52 ที่ขา RST ภายใน 89C52 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ (Ground) ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการรีเซ็ทการทำงานของ 89C52 ดังนั้นจึงสามารถต่อตัวเก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง + 5 โวลท์ เพื่อให้เกิดการรีเซ็ท เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 89C52 ซึ่งเรียกว่า Power on reset การรีเซ็ทจะทำให้ค่าในรีจิสเตอร์ต่าง ๆ เปลี่ยนแปลงไปเป็นค่าหนึ่งดังตารางรูปที่ 3.7

รูปที่ 3.7 ค่าในรีจิสเตอร์เมื่อเกิดการรีเซ็ท 89C52

ในตารางรูปที่ 3.7 ช่องทางขวาเป็นค่าของรีจิสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการรีเซ็ทในรีจิสเตอร์ SBUF เมื่อสิ้นสุดการรีเซ็ทจะมีค่าที่ไม่แน่นอน และพอร์ทจะอยู่ในสภาวะลอจิก 1 ทุกบิทตลอดเวลาที่สัญญาณของขา RST เป็น HIGH อยู่

เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซ็ท 89C52 จะเริ่มทำงานจาก คำสั่งที่อยู่ใน Program memory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงาน ถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้จะต้องเขียนโปรแกรมมาเก็บไว้ที่ตำแหน่ง 0000H ในเครื่องไมโครคอมพิวเตอร์แบบบอร์ดเดี่ยว (Single Board Micro- computor) จะมีโปรแกรมที่เขียนเก็บไว้เริ่มจากตำแหน่ง 0000H นี้เรียกว่า มอนิเตอร์โปรแกรม (Monitor program) ที่จะคอยรับการกดแป้นพิมพ์ (Keyboard) และแสดงผลทางตัวแสดงผล (Display) แบบ 7 Segment

ALE (ที่ขา 30)

Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์ สัญญาณนี้จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความ จำสำหรับข้อมูลภายนอก 89C52 สัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอก 89C52 ว่าขณะนี้สัญญาณนี้ Active (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิทล่างของตำแหน่งหน่วยความจำภายนอก 89C52 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุต

$\overline{\text{PSEN}}$ (ที่ขา 29)

Program Store Enable เป็นขาที่ 29 ในรูปที่ 3.2 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 89C52 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปร- แกรมภายใน 89C52 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา $\overline{\text{PSEN}}$ นี้สามารถต่อไปยังขาอินพุตของ TTL ชนิด LS ได้ถึง 8 อินพุต

$\overline{\text{EA}}$ (ที่ขา 31)

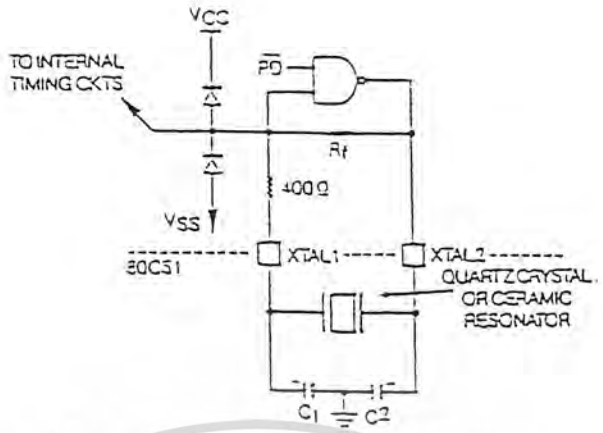
External Access ขา 31 ของรูปที่ 3.2 ขานี้เป็นขาอินพุตที่ต่อเข้าไปยังวงจร Timing 0 and Control ในรูปที่ 3.1 เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา $\overline{\text{EA}}$ นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 89C52 จะต้องสร้างสัญญาณ $\overline{\text{PSEN}}$ ออกไปยังภายนอก เพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา $\overline{\text{EA}}$ เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง0FFFH ถูกเก็บไว้ภายใน 89C52 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่าง ๆ จาก ROM ภายใน 89C52

XTAL 1 (ที่ขา 19)

ขาที่ 19 ของรูปที่ 3.2 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิสเลเตอร์ ในรูปที่ 3.8 จะเห็นวงจรภายในของออสซิลเล-เตอร์ NAND Gate จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิสเลตหรือไม่ก็ขึ้นกับสัญญาณ PD ซึ่งต่อมาจากบิท PD ของรีจิสเตอร์ PCON ถ้าต้องการใช้สัญญาณนาฬิกา (Clock Signal) จากภายนอกมาเป็นสัญญาณนาฬิกา มาควบคุมการทำงานของ 89C52 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้วงจรออสซิสเล-เตอร์ภายในก็ให้ต่อ Crytal หรือเซรามิคเรโซเนเตอร์ ดังรูปที่ 3.8 คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 PF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

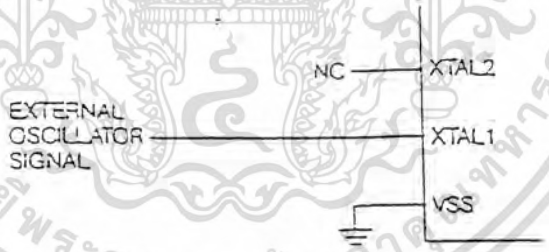
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 3.8 วงจรออสซิสเลเตอร์ภายใน 89C52

XTAL 2 (ที่ขา 18)

ขาที่ 18 ของรูปที่ 3.2 ขานี้เป็นจุดเอาต์พุตของวงจรขยายแบบกลับสัญญาณที่ประกอบเป็นวงจรออสซิสเลเตอร์ (อินพุตคือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็นสัญญาณนาฬิกาของ 89C52 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 3.9



รูปที่ 3.9 89C52 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

3.2 การทำงานของ 89C52

คอมพิวเตอร์จะทำงานด้วยวงจรที่เรียกว่าฮาร์ดแวร์ (Hardware) ประกอบขึ้นมาเพียงอย่างเดียวไม่ได้ จะต้องมีโปรแกรม หรือคำสั่งที่จัดเรียงกันไว้ให้คอมพิวเตอร์ทำงานตามลำดับใน 89C52 ก็เช่นกัน ผู้ใช้ต้องเขียนโปรแกรมเป็นภาษาเครื่อง ซึ่งอยู่ในรูปของเลขฐาน 2 เก็บไว้ในหน่วยความจำประเภท Program Memory แต่ละคำสั่งของ 89C52 อาจประกอบด้วย 1 , 2 หรือ 3 ไบท์ แล้วแต่ว่าจะเป็นคำสั่งให้ทำงานอะไร คอมพิวเตอร์ก็จะเหมือนกับคนที่จะต้องทำงานตามคำสั่ง เมื่อรับคำสั่งแล้วก็จะไปทำตามคำสั่งนั้นเสร็จสิ้นแล้วก็กลับมารับคำสั่งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

33926

จากรูปที่ 3.1 เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 89C52 ซึ่งมีวงจรนับ Power on reset ต่ออยู่จะมีการรีเซ็ทเกิดขึ้น การทำงานภายใน 89C52 จะเริ่มจากบล็อก Program Counter ซึ่งเป็นวงจรนับ (Counter Circuit) ชนิดหนึ่ง ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนบัส (BUS) หมายเลข 1 บัสนี้มีขนาด 16 บิท ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ 0 Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำ จะปรากฏที่บนบัส 16 บิท หมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซ็ท ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็น ROM ภายใน หรือภายนอก 89C52 โดยการป้อนสภาวะลอจิกเข้าไปที่ 89C52 ทางขา EA ซึ่งต่ออยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส (Decoder) แล้วสร้างสัญญาณควบคุมต่อไป

ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา $\overline{EA}$ จะเป็นการเลือกใช้ ROM ภายใน 89C52 โดยที่วงจรTiming and Control จะสร้างสัญญาณไปยัง ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมาทางบัสหมายเลข 2 ข้อมูลจาก ROM จะถูกส่งลงไปยังบัสหมายเลข 3ที่เรียกว่า Internal Data Bus แล้วนำไปเก็บไว้ที่ Instruction Register (เป็นวงจร Latch) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่น ๆ ต่อไป

แล้วแต่ว่าเป็นคำสั่งให้ทำงานอะไร ในกรณีที่เลือก ROM ภายนอก 89C52 โดยป้อนสัญญาณลอ-จิก 1 เข้าไปที่ขา EA จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ท 0 และพอร์ท

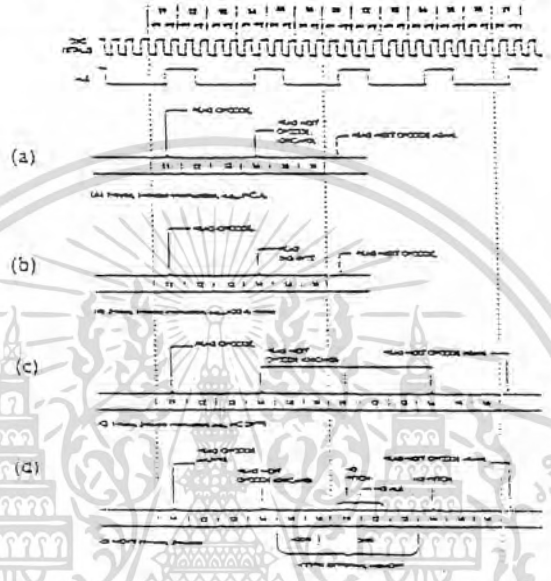
2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ท 0 ไปยัง Internal Data Bus แล้วนำไปเก็บไว้ที่ Instruction Register เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจาก ROM ภายในการทำงานในช่วงส่งค่าตำแหน่งหน่วยความจำไปยังหน่วยความจำ แล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ใน Instruction Register เรียกว่าเป็นช่วงของการ Fetch (Fetch Cycle) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่นถ้าเป็นคำสั่งให้บวกข้อมูลในรีจิสเตอร์ Accu-mulator กับข้อมูลจากหน่วยความจำ Data Memory ภายใน RAM ตำแหน่ง 23H วงจร Timing and Control ก็จะส่งสัญญาณให้ Instruction Register ส่งค่าตำแหน่งหน่วยความจำ 23 H ลงไปยัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อใช้ชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่งให้ RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1 (วงจร Latch) ขณะเดียวกันวงจร Timing and Control ก็จะส่งสัญญาณไปยัง ACC ให้ส่งข้อมูลมายัง TMP2 (วงจร Latch) วงจร ALU ซึ่งโครงสร้างเป็นวงจรทำการคำนวณทางคณิตศาสตร์ (บวก , ลบ , คูณ , หาร) และยังสามารถทำงานทางลอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จิก (AND , OR , NOT , XOR) จะทำการบวกเลขจาก TMP1 และ TMP2 เข้าด้วยกัน ผลลัพธ์ที่ได้จะส่งผ่าน Internal Data Bus กลับไปเก็บยัง ACC

PSW (Program Status Word) ซึ่งจะทำหน้าที่เก็บสถานะผลลัพธ์ของการทำงานใน ALU เช่น ผลลัพธ์การบวกมีค่าเกิน 8 บิท ก็จะทำให้บิทหนึ่งใน PSW ถูก SET เป็น 1

การทำงานที่กล่าวมาข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and Control และสัญญาณที่สร้างขึ้นนี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างมาจากวงจร Oscillator ทำให้การทำงานต่าง ๆ เป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ ดังในรูปที่ 3.10



รูปที่ 3.10 ลำดับสถานะการทำงานใน MCS-51

คำสั่งแต่ละคำสั่งของ 89C52 จะใช้เวลาทำงาน 1 , 2 หรือ 3 ไซเคิลของเครื่อง (Mac- hine Cycle) แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 ไซเคิลของเครื่องจะใช้เวลา 12 ไซเคิล ของสัญญาณนาฬิกา ดังนั้นแต่ละคำสั่งของ 89C52 จะใช้เวลาการทำงาน 12 , 24 หรือ 36 ไซเคิล ของสัญญาณนาฬิกานั้นเอง แต่ละไซเคิลของเครื่องจะถูกแบ่งออกเป็น 6 State คือ S1 , S2 , S3 , S4 , S5 , และ S6 แต่ละ State จะประกอบด้วย 2 ไซเคิล ของสัญญาณนาฬิกา ในไซเคิลแรกจะเรียกว่าเฟส 1 (P1) และไซเคิลที่ 2 เรียกว่าเฟส 2 (P2) ในแต่ละเฟสจะนับตั้งแต่ขอบขาลงของสัญญาณนาฬิกา ถึงขอบขาลงของสัญญาณนาฬิกาที่อยู่ถัดไป ดังในรูปที่ 3.10 เมื่อ 89C52 ทำงานเสร็จ 1 ไซเคิลของเครื่องก็จะเริ่มทำงาน State 1 Phase 1 (S1 P1) ของไซเคิลต่อไป

ใน 1 ไซเคิล ของเครื่องวงจร Timing and Control จะสร้างสัญญาณ ALE ออกมา 2 ไซเคิล เพื่อ Fetch คำสั่งเข้าไป 2 ครั้งเสมอ ที่บริเวณขอบขาขึ้นของสัญญาณ ALE คำสั่งใดจะมีกี่ไบท์

หรือใช้เวลาทำงานกี่ไซเคิล จะดูได้จากตารางชุดคำสั่ง 89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งประเภท 1 ไบท์ 1 ไซเคิลของเครื่องได้แก่คำสั่ง INC A จะมีการอ่านคำสั่งจากหน่วยความจำสำหรับโปรแกรม 2 ครั้ง ที่เวลาประมาณขอบขาขึ้นของสัญญาณ ALE เมื่อคำสั่งแรกถูกอ่านเข้าไปที่เวลา ที่เวลาขอบขาขึ้นของสัญญาณ ALE แรก แล้วนำไปเก็บที่ Instruction Register เพื่อให้วงจร Timing and Control ถอดรหัสแล้วเข้าอยู่การ Execute ขณะเดียวกันก็จะเริ่มต้นการ Fetch คำสั่งที่อยู่ในหน่วยความจำคำสั่งถัดไปเข้ามา และคำสั่งที่ 2 จะถูกอ่านเข้ามาที่เวลาขอบขาขึ้นของสัญญาณ ALE ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็จะทราบว่า การทำงานคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่งเพียง 1 ไบท์ ดังนั้นคำสั่งที่ถูกอ่านมาไบท์ที่ 2 จะไม่ถูกนำมาทำงาน เพียงแต่อ่านเข้ามาแล้วทิ้งไป (Discard) ดังในรูปที่ 3.10a

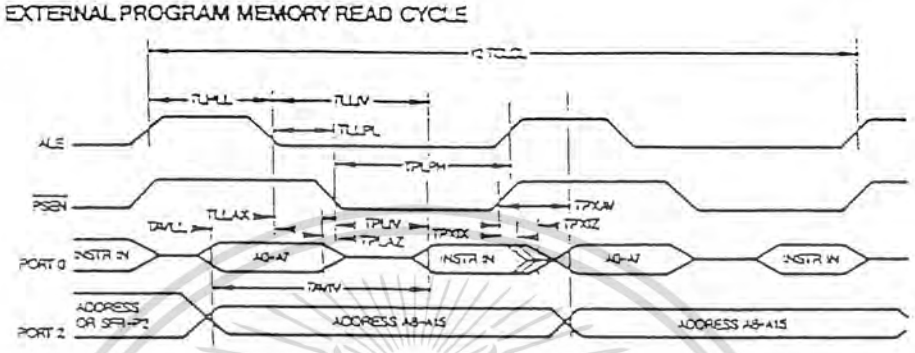
คำสั่งประเภท 2 ไบท์ และใช้เวลาเพียง 1 ไซเคิล ของเครื่องได้แก่คำสั่ง ADD A # Data ในหนึ่งไซเคิลของเครื่องนี้ จะมีการอ่านคำสั่งเข้ามา 2 ไบท์ เหมือนกับคำสั่งประเภท 1 ไบท์ 1 ไซเคิลของเครื่อง แตกต่างกันที่ไบท์ที่ 2 จะถูกนำมาใช้งานด้วยไม่ได้ถูกทิ้งไปดังในรูปที่ 3.10b ตัวอย่างของคำสั่ง ADD A , # 33H จะเขียนเป็นภาษาเครื่องได้ 2 ไบท์ คือ 24 33 เมื่ออ่านคำสั่งไบท์แรกคือ 24 เข้าไปไว้ที่ Instruction Register และ Timing and Control จะถอดรหัสพบว่าเป็นคำสั่งบวกเลข ก็จะส่งสัญญาณไปยัง Accumulator ให้เอาข้อมูลไปไว้ที่ TMP1 เมื่อคำสั่งที่ 2 ถูกอ่านเข้ามาที่ Instruction Register และ Timing and Control จะส่งให้เอาข้อมูลไบท์ที่ 2 ส่งลงไปยัง Internal Data Bus ไปเก็บยัง TMP1 จากนั้นวงจร ALU จะนำเอาข้อมูล TMP1 และ TMP2 มาบวกกัน ผลลัพธ์ที่ได้จะส่งออกมาจาก ALU ไปยัง Internal Data Bus แล้วไปเก็บไว้ที่ Accumulator

คำสั่งประเภท 1 , 2 หรือ 3 ไบท์ ที่ใช้เวลาทำงาน 2 ไซเคิล ของเครื่อง เช่น คำสั่ง INC DPTR จะมีการอ่านคำสั่งเข้าไป 4 ครั้ง ทุก ๆ ขอบขาขึ้นของสัญญาณ ALE ที่มี 2 ครั้งต่อ 1 ไซเคิล ของเครื่อง ถ้าเป็นคำสั่งประเภท 1 , 2 หรือ 3 ไบท์ วงจร Timing and Control จะเอาคำสั่ง 1 , 2 หรือ 3 ไบท์ แรกเท่านั้นไปทำงาน ส่วนคำสั่งที่เหลือทิ้งไปดังในรูปที่ 3.10c คำสั่ง 1ไบท์ ที่ใช้เวลาทำงาน 2 ไซเคิล ของเครื่องที่กล่าวมาแล้วจะไม่รวมถึงคำสั่ง MOVX ซึ่งใช้ในการอ่าน หรือเขียนข้อมูลกับหน่วยความจำ Data Memory ภายนอก การทำงานของคำสั่งนี้จะมีการ Fetch คำสั่งเข้าไป 2 ไบท์ ในไซเคิลของเครื่องแรก ในไซเคิลของเครื่องที่ 2 จะไม่มีการ Fetch คำสั่งเข้าไป แต่จะเป็นช่วงเวลาของการอ่าน หรือเขียนข้อมูลกับ Data Memory ภายนอก สัญญาณ ALE ซึ่งปกติจะเปลี่ยนเป็น 1 ที่ S1 P2 ก็จะไม่เปลี่ยนเป็น 1 ในไซเคิลของเครื่องที่ 2 โดยจะเป็น 0 อยู่จนกว่าจะถึงเวลา S4 P2 ของไซเคิลของเครื่องที่ 2 สัญญาณ ALE จะเปลี่ยนเป็น 1 เพื่อการอ่าน หรือเขียนข้อมูลกับ Data Memory ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

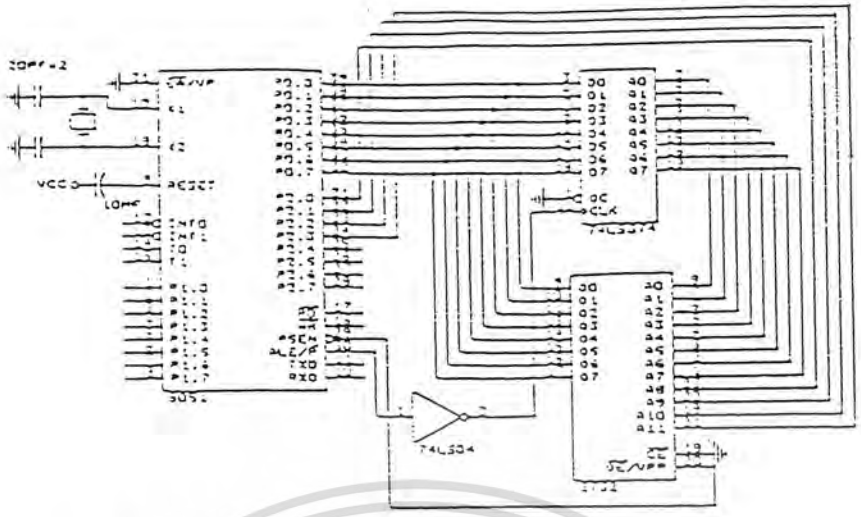
การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 89C52 นั้น ลำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่ง ดังรูปที่ 3.11



รูปที่ 3.11 Timing Diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก

การอ่านคำสั่ง (Fetch) จาก Program area ภายนอก จะเริ่มจาก 89C52 ส่งสัญญาณ ลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณที่ขา $\overline{\text{PSEN}}$ จะเป็น 1 จากนั้น Port 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่าง และพอร์ท 2 จะส่งตำแหน่งหน่วยความจำ 8 บิทบนออกมา แล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาลงของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 ไว้ จากนั้นพอร์ท 0 ก็จะยกเลิกการส่งค่าตำแหน่งหน่วยความจำ เข้าสู่สภาวะ High Impedance และสัญญาณ $\overline{\text{PSEN}}$ จะเป็น 0 เพื่อเตรียมรับคำ-สั่ง ที่ส่งออกจากหน่วยความจำภายนอก เข้าไปยัง 89C52 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register (ดูรูปที่ 3.1) แล้ว สัญญาณ PSEN จะกลับเป็น 1 พร้อมกับสัญญาณ ALE ก็จะกลับเป็น High เพื่อการอ่านคำสั่งต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลา ตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่ง จากนั้นจะเริ่มลำดับการ Fetch ข้อมูลไบท์ที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งจะมีการเปลี่ยนแปลงของสัญญาณตามเวลา เหมือนกับการ Fetch ไบท์แรกนั้นเอง จาก TimigDiagram ดังกล่าว จะออกแบบวงจรที่มี Program Memory อยู่ภายนอก 89C52 ได้ดังตัวอย่างในรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 วงจรที่มี Program Memory อยู่ภายนอก 89C52

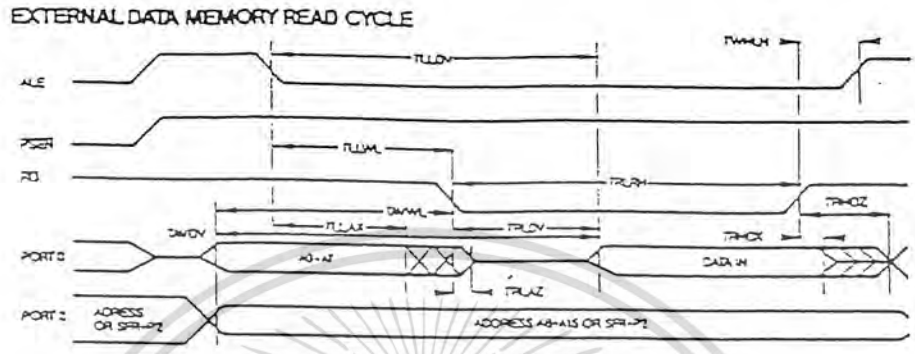
74LS374 ในรูปที่ 3.12 จะทำหน้าที่ Latch ตำแหน่งในหน่วยความจำ 8 บิทล่าง ที่
เวลาขอบขาลงของสัญญาณ ALE ซึ่งสัญญาณ ALE จะถูกกลับให้เป็นตรงกันข้ามโดย Inverter 74LS04 ก่อนที่จะป้อนให้กับขา CK ของ 74LS374 และที่ขอบขาขึ้นของสัญญาณที่ออกจาก 74LS04 และ Latch ตำแหน่งหน่วยความจำ ข้อมูลที่ออกจาก 74LS374 จะเป็นค่า 8 บิทล่างของตำแหน่งหน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อค่าตำแหน่งหน่วยความจำ 8 บิท เข้ากับ A0ถึง A7 ของ EPROM และข้อมูลจากพอร์ท 2 บิท P2.0 ถึง P2.3 จะต่อเข้ากับ A8 - A11 ของ EPROM โดยตรง เพราะค่าตำแหน่งหน่วยความจำ 8 บิท บนที่ออกมาจากพอร์ท 2 จะคงที่ตลอดเวลา ขา $\overline{\text{PSEN}}$ ของ 89C52 จะถูกต่อเข้ากับขา $\overline{\text{OE}}$ ของ EPROM 2716 ดังนั้นเมื่อสัญญาณ $\overline{\text{PSEN}}$ มีสภาวะลอจิกเป็น 0 ก็ส่งคำสั่งที่เก็บไว้ใน EPROM ณ ตำแหน่งที่ชี้โดยข้อมูลที่ขา A0 ถึง A11 ออกมายังพอร์ท 0 และถูก 89C52 เก็บไปทำงานต่อไป

การอ่าน - เขียนข้อมูลกับหน่วยความจำสำหรับข้อมูลภายนอก 89C52

การอ่าน - เขียนข้อมูลกับ Data Memory ภายใน 89C52 นั้นจะมีสัญญาณสร้างมาจากส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นที่จะต้องทำความเข้าใจ แต่การอ่าน - เขียนข้อมูลกับ Data Memory ภายนอก อันเนื่องมาจากคำสั่ง MOVX นั้น เมื่อคำสั่งดังกล่าวถูกอ่านเข้ามายัง Instruction Register แล้ว Timing and Control จะทำการถอดรหัส และสร้างสัญญาณควบคุมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านข้อมูลจาก External Data Memory จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 3.13

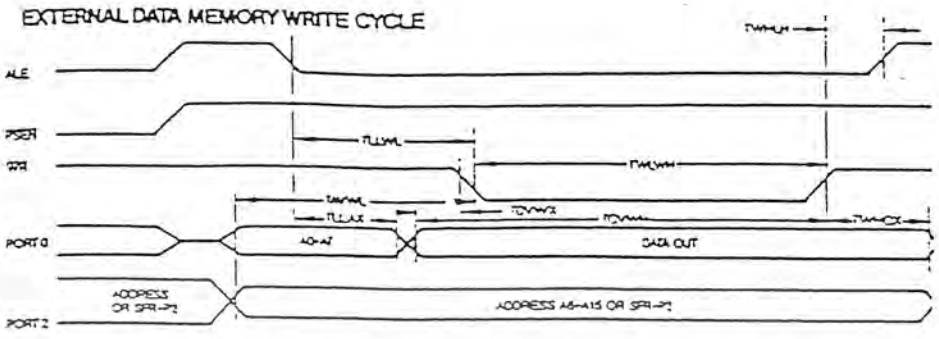


รูปที่ 3.13 Timing diagram ของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูล ภายนอก 89C52

การทำงานจะเริ่มจากการส่งค่าตำแหน่งหน่วยความจำภายนอก 8 บิทล่าง ออกพอร์ท 0 และ 8 บิทบนออกทางพอร์ท 2 เมื่อส่งค่าตำแหน่งแล้ว สัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับมาเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามารถ Latch ตำแหน่งหน่วยความจำไว้เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 89C52 เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลทางพอร์ท 0 จะเปลี่ยนแปลงไปก็ยังมีค่าตำแหน่งหน่วยความจำส่งไปยังหน่วยความจำในระหว่างการติดต่อกับ Data Memory นี้สัญญาณ PSEN จะเป็น 1 ตลอดเพราะสัญญาณ PSEN จะเป็น Active (เป็น 0) ก็ต่อเมื่อเป็นการติดต่อหน่วยความจำ สำหรับโปรแกรมภายนอก 89C52 เท่านั้น 89C52 จะส่งสัญญาณ ลอจิก 0 ออกมาทางขวา RD (P3.7) เพื่อบอกกับหน่วยความจำภายนอกว่าต้องการอ่านข้อมูลเข้าไปเมื่อ 89C52 ส่งสัญญาณ RD เป็นลอจิก 0 ทำให้พอร์ท 0 เข้าสู่สถานะ High Impe- dance พร้อมที่จะให้หน่วยความจำภายนอกส่งข้อมูลมาบนพอร์ท 0 ซึ่งส่งมาจากหน่วยความจำภายนอกจะถูกอ่านเข้าไปเก็บที่เวลาขอบขาขึ้นของสัญญาณ RD จากนั้นสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไปในระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภานนอกนี้ พอร์ท 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทบนออกมาตลอดเวลา

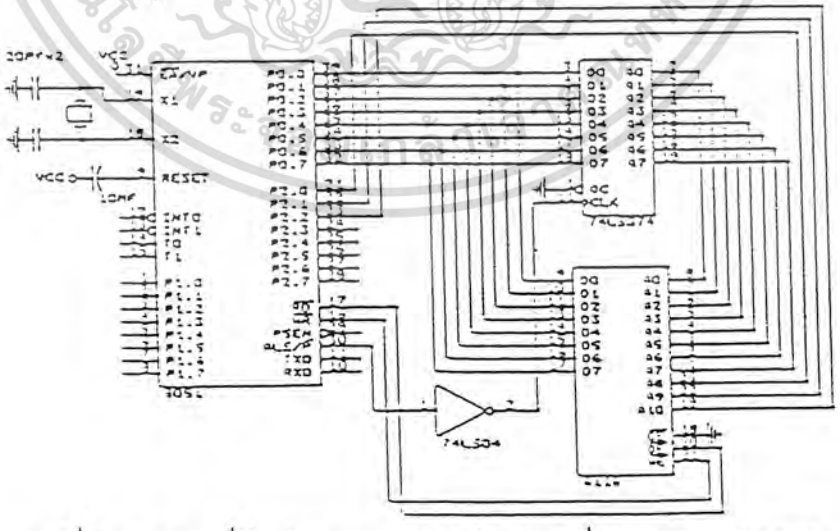
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 89C52 จะมีไดอะแกรมสัญญาณตามเวลาดังรูป 1.14



รูปที่ 3.14 Timing diagram ของการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 89C52

เมื่อ 89C52 ส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างไปทางพอร์ท 0 และหน่วยความจำ 8บิท บนลงไปทางพอร์ท 2 แล้ว สัญญาณ ALE ก็จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำบนพอร์ท 0 เหมือนกับการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก เมื่อสัญญาณ ALE เป็น 0 แล้ว 89C52 จะส่งข้อมูลที่ต้องการเขียนไปยัง
พอร์ท 0 แล้วจะให้สัญญาณ $\overline{WR}$ เปลี่ยนสภาวะลอจิกเป็น 0 ขณะนี้หน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้น สัญญาณ $\overline{WR}$ จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลแล้วสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อ Fetch คำสั่งต่อไปมาทำงาน หน่วยความจำสำหรับข้อมูลภายนอก ที่สามารถอ่านและเขียนข้อมูลได้ดังรูปที่ 3.15



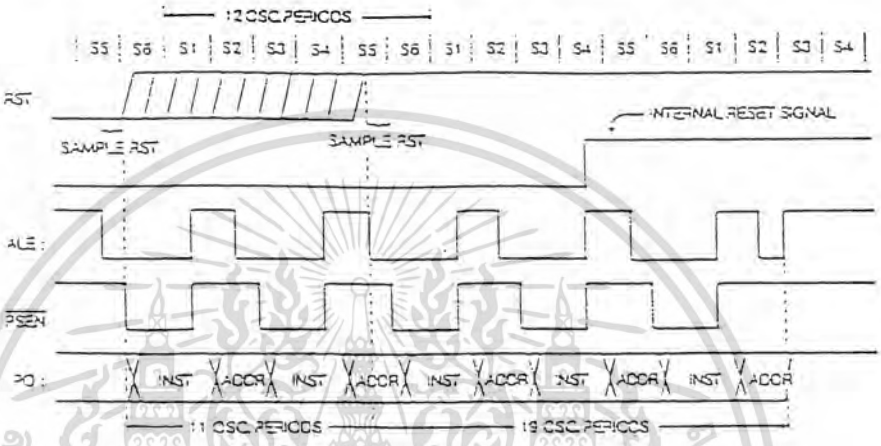
รูปที่ 3.15 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่ภายนอก 89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

74LS374 ในรูปจะใช้สำหรับ Latch ค่าตำแหน่งหน่วยความจำ 8 บิทล่างไว้ แม้ว่าข้อมูลบนพอร์ท 2 จะเปลี่ยนไป สัญญาณ RD และ WR จะอ่านหรือเขียนข้อมูลจากหน่วยความจำภายนอก 6116 เป็นหน่วยความจำแบบ RAM ที่สามารถจะอ่านและเขียนข้อมูลได้

3.4 การรีเซ็ท

เมื่อป้อนสัญญาณที่มีสภาวะลอจิกเป็น 1 เข้าไปทางขวา RST จะไม่ได้เกิดการรีเซ็ทขึ้นทันทีทันใด แต่ลำดับการเกิดรีเซ็ทจะแสดงได้ดังไดอะแกรมตามเวลาในรูปที่ 3.16



รูปที่ 3.16 ไดอะแกรมตามเวลาของการรีเซ็ท

ในรูปที่ 3.16 เป็น Timing Diagram ของการรีเซ็ท สภาวะลอจิกของสัญญาณที่ขา RST ที่ถูกอ่านเข้ามาที่เวลา S5P2 (เฟส 2 State 5) ของทุก ๆ ไซเคิลของเครื่อง ในกรณีที่เป็นคำสั่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิล ของเครื่องก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาในไซเคิลที่ 2 ของการทำงาน ดังนั้นในการรีเซ็ท จะต้องป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขาเป็นเวลาอย่างน้อย 2 ไซเคิลของเครื่องหรือ 24 ไซเคิล ของสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์ภายใน 89C52 เพื่อให้แน่ใจว่าสัญญาณรีเซ็ท จะถูกอ่านเข้าไปตรวจสอบและทำงาน ขณะที่ทำงานรีเซ็ท 89C52 ออสซิลเลเตอร์จึงจะต้องทำงานอยู่ด้วย เมื่อ 89C52 สุ่มข้อมูลที่ขา RST แล้วตรวจพบว่าเป็นสภาวะลอจิก 1 ก็จะสร้างสัญญาณรีเซ็ทขึ้นภายใน ที่เวลา S2P4 ของไซเคิลเครื่องถัดไป ข้อมูลแต่ละพอร์ทที่ส่งออกมา จะยังคงปรากฏที่พอร์ทจนกว่าจะเกิดการรีเซ็ทขึ้นซึ่งต้องใช้เวลา 19 ไซเคิลของสัญญาณจากออสซิลเลเตอร์นับตั้งแต่เวลา S5P2 ในไซเคิลของเครื่องที่พบสัญญาณรีเซ็ท ในระหว่างเวลา 19 ไซเคิล นี้จะยังคงมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

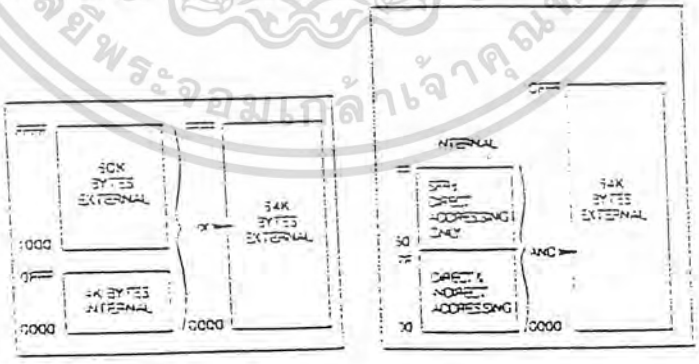
สภาวะของสัญญาณลอจิกที่ขา RST จะถูกอ่านเข้าไปตรวจสอบที่เวลา S5P2 ของทุก ๆ ไซเคิล ของเครื่อง ดังนั้นถึงแม้ว่าสัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาก่อนก็ยังไม่เกิดการตรวจสอบสัญญาณรีเซ็ท ดังในรูปที่ 3.16 สัญญาณที่ขา RST อาจเป็น 1 มาตั้งแต่ State ที่ 6 ก็จะไม่เกิดอะไรขึ้นจนกระทั่ง 1 ไซเคิลของออสซิลเลเตอร์ต่อมาซึ่งเป็นเวลา S5P2 จึงจะเกิดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการทำงานมากกว่า 1 ไซเคิลของเครื่อง 89C52 ก็จะต้องทำงานในคำสั่งนั้นให้เสร็จสิ้นเสียก่อนจึงจะเริ่มการรีเซ็ทได้ โดย 89C52 จะดูสภาวะของสัญญาณที่ขา RST ของ S5P2 ในไซเคิลของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไซเคิลเครื่องแรก ๆ ในคำสั่งอาจมีสภาวะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไซเคิลเครื่องสุดท้าย มีสภาวะลอจิกที่ขา RST เป็น 0 ก็จะไม่เกิดการรีเซ็ทขึ้นที่เวลา S5P2 เมื่อตรวจสอบสภาวะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอไปจนถึงเวลา S4P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของไซเคิลเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซ็ทภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซ็ท โดยการเขียนข้อมูล 0 ไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังตารางในรูปที่ 3.7 ระหว่างนี้ข้อมูลใน RAM ภายใน 89C52 จะไม่เปลี่ยนแปลงในระหว่างการเขียนข้อมูลลงไปยัง SFR จะยังมีการ Fetch คำสั่งเข้ามาทำงานอีก 1 คำสั่งจนกว่าจะถึง S3P1 ของไซเคิลที่ 2 (นับแต่ไซเคิลของเครื่องที่ตรวจพบลอจิก 1 ที่ขา RST) ก็จะทำให้สภาวะลอจิกที่ขา ALE และ PSEN ค้างอยู่ที่สภาวะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสภาวะลอจิกที่ขา RST เป็น 0 เวลาตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ PSEN ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคิลของออสซิลเลเตอร์เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 89C52 จะรออีก 1 ถึง 2 ไซเคิลของเครื่องสัญญาณ ALE และ PSEN จะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานเริ่มจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

3.5 รีจิสเตอร์ของ 89C52

หน่วยความจำของ 89C52 แบ่งออกเป็น 2 แบบ คือ หน่วยความจำสำหรับโปรแกรม (Program Area) และหน่วยความจำสำหรับเก็บข้อมูล (Data Area) ดังแสดงในไดอะแกรมรูปที่ 3.17



รูปที่ 3.17 ไดอะแกรมภาพของหน่วยความจำ 89C52

หน่วยความจำสำหรับโปรแกรมเป็นหน่วยความจำที่ 89C52 ใช้สำหรับโปรแกรมภาษาเครื่องที่ 89C52 จะทำงานเมื่อเริ่มป้อนไฟเลี้ยงให้ 89C52 หรือมีการรีเซ็ท (Reset) 89C52 จะทำให้เริ่มการทำงานจากคำสั่งในโปรแกรมตำแหน่งที่ 0000H เมื่อทำงาน 1 คำสั่งก็จะทำให้รีจิสเตอร์ PC ที่ชี้ตำแหน่งโปรแกรมมีค่าเพิ่มขึ้นเพื่อชี้ตำแหน่งของคำสั่งต่อไป ตำแหน่งสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท้ายของหน่วยความจำคือ FFFFH หน่วยความจำสำหรับโปรแกรมนี้สามารถที่จะเลือกได้ว่าเป็นหน่วยความจำที่อยู่ภายใน 89C52 หรือภายนอก 89C52 ก็ได้ หน่วยความจำสูงสุดสำหรับโปรแกรมภายนอก 89C52 มีได้ถึง 64 KByte ทำให้สามารถใช้งานได้อย่างกว้างขวาง หน่วยความจำในช่วงนี้ 89C52 สามารถอ่านข้อมูลได้อย่างเดียว ไม่สามารถเขียนข้อมูลเข้าไปได้ระหว่างการทำงาน

หน่วยความจำสำหรับข้อมูลเป็นหน่วยความจำที่ 89C52 ใช้สำหรับเก็บหรือพักข้อมูลระหว่างที่ทำงาน หน่วยความจำสำหรับข้อมูลมี 2 แบบ แบบหนึ่งมีขนาด 128 ไบท์อยู่ภายใน 89C52 หน่วยความจำอีกแบบหนึ่งจะมีขนาด 64 กิโลไบท์ (Kbyte) ต้องต่อเพิ่มเติมเข้าไปภายนอก 89C52 หน่วยความจำภายในตำแหน่ง 0 ถึง 7FH นี้สามารถอ้างถึงได้โดยตรงคือมีการสั่งให้อ่านหรือเขียนข้อมูลไปยังตำแหน่งนั้นได้โดยตรง แต่หน่วยความจำตำแหน่ง 80H ถึง FFH นั้นเป็นรีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register , SFR) หน่วยความจำภายในช่วงนี้ใช้เป็นรีจิสเตอร์สำหรับงานเฉพาะอย่าง

หน่วยความจำสำหรับข้อมูลภายใน 89C52 ช่วง 00H ถึง 07FH สามารถแบ่งออกได้เป็น 3 กลุ่มคือ

1. Register bank 0.3 อยู่ในหน่วยความจำช่วงตำแหน่ง 00H ถึง 1FH หน่วยความจำนี้แบ่ง ออกเป็น 4 ชุด ชุดละ 8 ไบท์ แต่ละชุดเราเรียกว่า BANK แต่ละไบท์ใน 1 BANK จะมีชื่อของรีจิสเตอร์ว่า R0,R1,R2,R3,R4,R5,R6 และR7 รีจิสเตอร์เหล่านี้จะเรียกใช้งานในระหว่างการทำงานของโปรแกรมได้อย่างสะดวก และรีจิสเตอร์เหล่านี้จะเป็นชื่อซ้ำกันในทุก BANK การใช้งานจึงต้องเรียกใช้งานทีละ BANK เท่านั้น โดยการกำหนดให้รีจิสเตอร์ PSW ที่จะกล่าวถึงต่อไปในบทนี้ เมื่อมีการ Reset การทำงานของ 89C52 จะเริ่มการใช้งานรีจิสเตอร์ R0 ถึง R7 ที่ BANK 0 ซึ่งรีจิสเตอร์ R0 ถึง R7 ในแต่ละ BANK นั้นจะอ้างอิงในหน่วยความจำสำหรับข้อมูลภายใน 89C52 ดังในตาราง

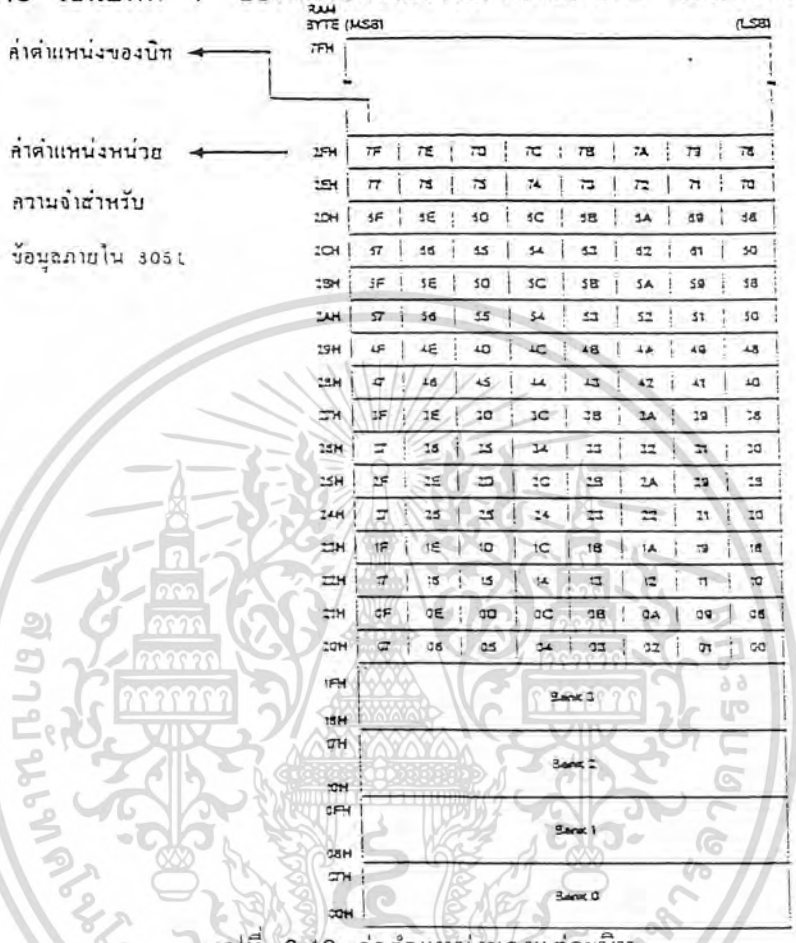
รีจิสเตอร์	ตำแหน่งหน่วยความจำ			
	BANK 0	BANK 1	BANK 2	BANK 3
R0	0	8	10	18
R1	1	9	11	19
R2	2	A	12	1A
R3	3	B	13	1B
R4	4	C	14	1C

ตัวอย่าง เมื่อกำลังมีการใช้งานในหน่วยความจำ BANK 1 และมีการอ้างถึงรีจิสเตอร์ R7 เช่นคำสั่ง MOV A , R7 (รหัสภาษาเครื่องคือ EFH)

การทำงานของคำสั่งนี้คือการเอาข้อมูลจากตำแหน่ง FH ของหน่วยความจำภายใน 89C52 ไปไว้ยังรีจิสเตอร์ A นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 Bit Address Area เป็นหน่วยความจำในช่วงตำแหน่ง 20H ถึง 2FH หน่วยความจำแต่ละบิท ในช่วงของหน่วยความจำดังกล่าวจะสามารถตรวจสอบหรือตั้งค่าเป็น 1 หรือ 0 ได้โดยการโปรแกรมภาษาเครื่อง แต่ละบิทของหน่วยความจำช่วงนี้ จะมีค่าของตำแหน่งดังใน Memory Map รูปที่ 3.18 เช่นบิทที่ 7 ของหน่วยความจำในตำแหน่ง 2FH จะมีค่าตำแหน่งเป็น 7FH นั่นเอง



รูปที่ 3.18 ค่าตำแหน่งของแต่ละบิท

ในรูปที่ 3.18 ตัวเลขทางซ้ายเป็นค่าตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 89C52 ซึ่งแต่ละบิทในตำแหน่งนั้นจะมีค่าเป็นเลขฐาน 16 ที่จะใช้เป็นค่าอ้างอิงในคำสั่งจัดการกับข้อมูลบิทนั้น

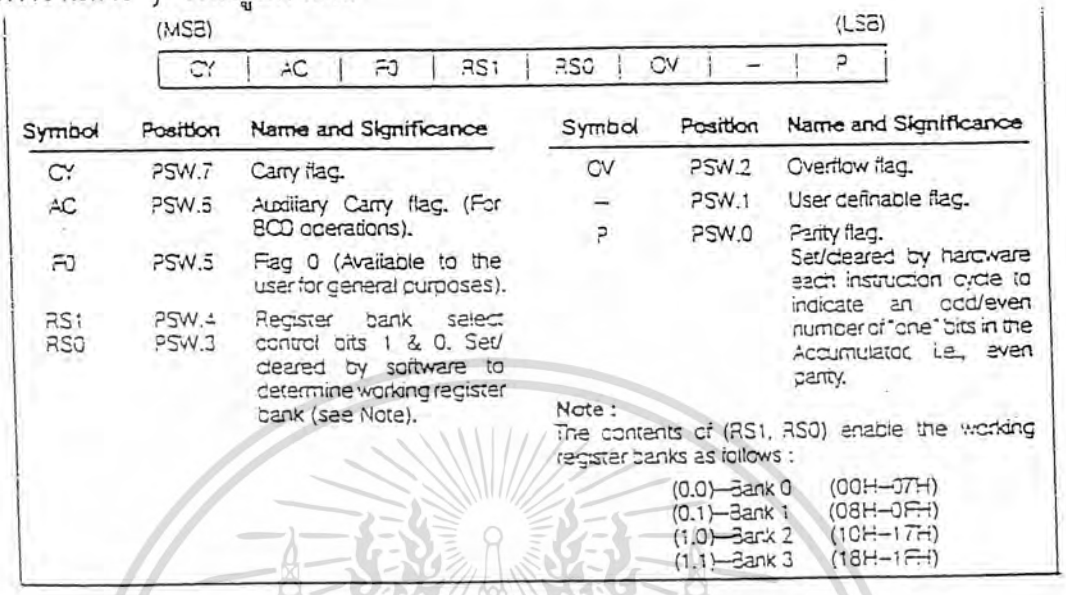
3. Scratched Pod Area เป็นช่วงของหน่วยความจำตำแหน่ง 30H ถึง 7FH หน่วยความจำช่วงนี้จะใช้สำหรับเก็บข้อมูลทั่วไป ถ้ารีจิสเตอร์ Stack Pointer ชี้มายังหน่วยความจำช่วงนี้จะต้องระวังไม่ให้เกิดการเขียนทับของข้อมูลอันจะทำให้การทำงานของโปรแกรมผิดพลาดได้

ที่กล่าวมาแล้วคงจะพอเข้าใจถึงลักษณะการจัดการหน่วยความจำของ 89C52 ได้พอที่จะเริ่มเขียนโปรแกรมภาษาแอสเซมบลีของ 89C52 สรุปชุดคำสั่งของ 89C52 จะอยู่ในภาคผนวก ก

จากตารางคำสั่งในหน้า 12-13 ของภาคผนวก ก จะมีคำอธิบายการใช้งานหรือการทำงานของแต่ละคำสั่งใน 89C52 ไว้ด้วย คำสั่งของ 89C52 เป็นคำสั่งที่ประสิทธิภาพการทำงานสูงมาก ในขณะที่ 89C52 ทำงานจะมีรีจิสเตอร์ตัวหนึ่งที่เก็บสภาวะ (Flag) ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างการคำนวณเช่น ตัวทด (Carry) หรือจะใช้เลือก BANK ของรีจิสเตอร์ภายใน 8052 ก็ได้ รีจิสเตอร์นี้คือ Program Status Word (PSW) มีขนาด 8 บิท แต่ละบิทจะใช้เก็บสถานะการทำงานต่าง ๆ ไว้ดังรูปที่ 3.19



รูปที่ 3.19 Program Status Word (PSW)

PSW.0 บิท 0 เรียกว่าบิทพาริตี้บิทนี้จะบอกว่าในรีจิสเตอร์ Accumulator หรือรีจิสเตอร์ A มี 1 เป็นจำนวนคี่หรือคู่ เช่นในรีจิสเตอร์ A ขนาด 8 บิทมี 1 อยู่ 3 ตัวและมี 0 อยู่ 5 ตัวก็จะทำให้บิท PSW.0 นี้มีค่าเป็น 1 ถ้าใน Accumulator มี 1 อยู่เป็นจำนวนคู่ก็จะทำให้บิทนี้มีค่าเป็น 0

PSW.1 บิท 1 บิทนี้ไม่มีการใช้งาน

PSW.2 บิท 2 เรียกว่า Overflow Flag เป็นบิทที่บอกการคำนวณนั้นทำให้เกิดตัวทดขึ้นในระหว่างการคำนวณ ตัวทดนี้เป็นตัวทดที่เกิดจากบิท 6 ไปยังบิท 7 มีประโยชน์เมื่อทำการคำนวณแบบ Signed Integer

PSW.3 , PSW.4 บิท 3 และ 4 2 บิทนี้จะใช้งานร่วมกันเพื่อเป็นตัวบอกว่าขณะนี้ใช้รีจิสเตอร์ R0 ถึง R7 ใน BANK ได้ดังตาราง

บิท 4 (RB1)	บิท 3 (RB0)	Register bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น บิท 4 และบิท 3 มีค่าเป็น 10_2 เป็นการเลือกรีจิสเตอร์ BANK 2 หมายความว่าในรหัสคำสั่งช่วยจำที่อ้างอิงถึง R0 ก็จะอ้างถึงหน่วยความจำภายในที่ตำแหน่ง 10H

PSW.5 บิท 5 เรียกว่าบิทเอนกประสงค์เป็นบิทที่ผู้ใช้สามารถใช้คำสั่งกำหนดค่าให้เป็น 0 หรือ 1 ก็ได้ โดยที่การทำงานของคำสั่งอื่น จะไม่ทำให้บิทนี้มีค่าเปลี่ยนแปลง บิทนี้มีประโยชน์สำหรับในการส่งสถานะของโปรแกรมระหว่างเรียกการทำงานของโปรแกรมย่อย (Subroutine)

PSW.6 บิท 6 เรียกว่า Auxiliary Carry Flag เป็นบิทที่ใช้สำหรับเก็บตัวทดที่เกิดขึ้นระหว่างการคำนวณ โดยตัวทดนี้เป็นตัวทดที่เกิดการคำนวณของบิท 3 ข้ามไปยังบิท 4

PSW.7 บิท 7 เรียกว่า Carry Flag เป็นบิทที่บอกสถานะการคำนวณทางคณิตศาสตร์ว่าผลลัพธ์นั้นทำให้เกิดตัวทดขึ้นหรือไม่ เช่นการบวกเลข 2 จำนวนเข้าด้วยกันแล้วผลลัพธ์มีค่ามากกว่า 255 ก็จะทำให้เกิดตัวทดขึ้น เนื่องมาจากว่า Accumulator ที่ทำการบวกนี้สามารถเก็บข้อมูลได้เพียง 8 บิทเท่านั้น และทำให้บิทนี้มีค่าเป็น 1

3.6 พื้นที่หน่วยความจำข้อมูลแฝงของ AT89C52

เป็นพื้นที่ที่อยู่ซ้อนทับกับค่าตำแหน่งเดียวกับ หมายเลขรีจิสเตอร์ฟังชั่นพิเศษเช่นเดียวกับ ไมโครคอนโทรลเลอร์ 8052 การเข้าถึงจะสามารถเข้าถึงโดยอ้อม (indirect) มีขนาด 128 ไบต์อยู่ที่ตำแหน่ง 80H - FFH

3.7 รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register , SFR)

ใน 89C52 จะใช้วิธีการกำหนดชื่อให้กับตำแหน่งของหน่วยความจำ สำหรับข้อมูลภายใน(Internal Data Memory) ที่เรียกว่า Symbolize เช่น การให้ชื่อหน่วยจำแต่ละตำแหน่งในแต่ละ Bank ซึ่งอยู่ในช่วงหน่วยความจำตำแหน่ง 00H ถึง 1FH แล้วในคำสั่งจะอ้างอิงหน่วยความจำแต่ละตำแหน่งโดยการใช้ชื่อ R0 , R1 , R2 , R3 , R4 , R5 , R6 และ R7 หน่วยความจำตำแหน่งเหล่านี้ จะเรียกอีกอย่างหนึ่งว่าเป็นรีจิสเตอร์ ซึ่งมีหน้าที่ในการเก็บหรือพักข้อมูล หรือใช้สำหรับการกระทำบางอย่าง รีจิส-เตอร์บางอย่างใน 89C52 ที่เรียกว่า Specail Function Register (SFR) เป็นรีจิสเตอร์ที่ใช้สำหรับงานเฉพาะ คือข้อมูลที่นำไปเก็บไว้รีจิสเตอร์เหล่านี้จะมีความหมายเฉพาะตัวของรีจิสเตอร์ ที่แต่ละตำแหน่งของ SFR อาจจะไม่ใช้เป็นหน่วยความจำ (RAM) แต่อาจเป็นตัวนับ (Register Counter) , Shift Register หรือ Latch ซึ่งการอ้างอิงข้อมูลในแต่ละตำแหน่งนั้น 89C52 จะถือเสมือนว่าเป็นหน่วยความจำตำแหน่งหนึ่ง จึงเรียกการมองข้อมูลแต่ละตำแหน่งนี้ว่า Memory Map I/O รีจิสเตอร์กลุ่มนี้มีดังในรูป ที่ 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

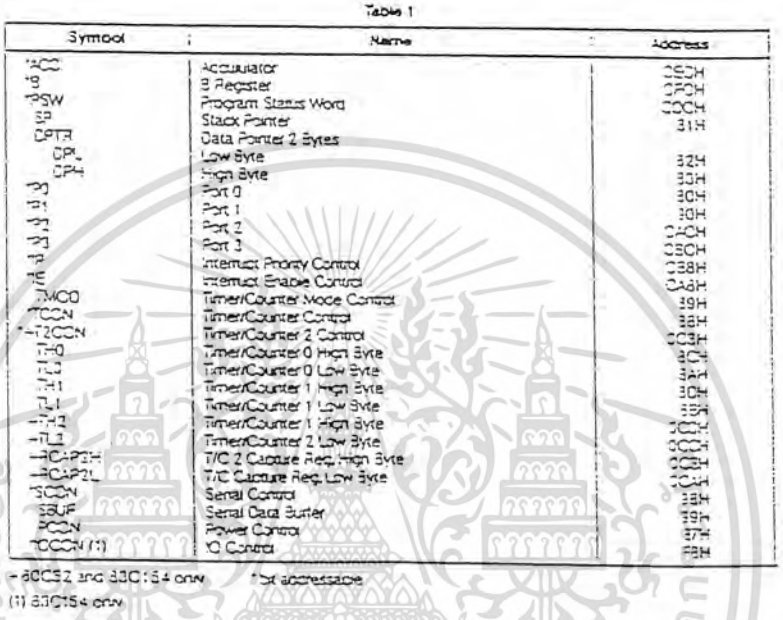


Table 1

Symbol	Name	Address
ACC	Accumulator	0E0H
B	B Register	0F0H
PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
P0	Port 0	80H
P1	Port 1	90H
P2	Port 2	0A0H
P3	Port 3	0B0H
IP	Interrupt Priority Control	0B8H
IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
TCON	Timer/Counter Control	88H
*T2CON	Timer/Counter 2 Control	0C8H
(T)H0	Timer/Counter 0 High Byte	8CH
(T)L0	Timer/Counter 0 Low Byte	8AH
(T)H1	Timer/Counter 1 High Byte	8DH
(T)L1	Timer/Counter 1 Low Byte	8BH
*(T)H2	Timer/Counter 1 High Byte	0CDH
*(T)L2	Timer/Counter 2 Low Byte	0CCH
*RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
*RCAP2L	T/C Capture Reg. Low Byte	0CAH
SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H
*CCON (1)	IO Control	0D8H

* 80C52 and 83C154 only * bit addressable
(1) 83C154 only

รูปที่ 3.20 Special Function Register (SFR)

ในรูปที่ 3.20 ช่อง Symbol ทางซ้ายจะเป็นสัญลักษณ์ของรีจิสเตอร์ ในช่องถัดมาคือชื่อของรี-จิสเตอร์ตามสัญลักษณ์ที่อยู่ทางซ้าย ในช่องขวาสุดจะเป็นตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 89C52 ที่แทนด้วยชื่อหรือสัญลักษณ์ทางซ้ายนั่นเอง เช่น ในบรรทัดแรกคือรีจิสเตอร์ชื่อ Accummulater ที่มีสัญลักษณ์ ACC รีจิสเตอร์นี้คือ หน่วยความจำสำหรับข้อมูลภายใน 89C52 ที่ตำแหน่ง 0E0H การอ่านหรือเขียนข้อมูลกับรีจิสเตอร์เหล่านี้ สามารถทำได้โดยการใช้คำสั่งในกลุ่มการย้ายข้อมูล (เช่น MOV A , # 25H หรือ MOV 0E0H , #25H) และรีจิสเตอร์บางตัวในกลุ่มนี้ ยังสามารถใช้คำสั่งในกลุ่ม Boolean Instruction เพื่อการทำงานในแต่ละบิทในรีจิสเตอร์เหล่านั้นได้ จากตารางในรูปที่ 3.20 รีจิสเตอร์ที่มีเครื่องหมาย * อยู่ข้างหน้าสามารถจะใช้คำสั่งในกลุ่ม Boolean Instruction จัดการกับแต่ละบิทได้ รีจิสเตอร์ที่มีเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมาย + นำหน้าหมายความว่า รีจิสเตอร์นั้นมีเฉพาะใน 80C52 และ 83C154 เท่านั้นไม่มีใน 89C52

รูปที่ 3.21 แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิท

รูปที่ 3.21 ในช่องสี่เหลี่ยมเล็ก ๆ จะเป็นตำแหน่งของบิทนั้นในแต่ละรีจิสเตอร์ เช่น ในช่องซ้ายของรีจิสเตอร์ TCON มีค่า ซึ่งเป็นตำแหน่งค่าบิท 7 ของหน่วยความจำ 88H ถ้าต้องการให้บิทนี้มีค่าเป็น 0 ก็สามารถทำได้โดยใช้คำสั่ง

CLR 8FH

หรือจะทำให้เป็น 1ก็ทำได้โดยคำสั่ง

SETB 8FH

Special Function Register

รีจิสเตอร์ในกลุ่ม Special Function Register มีดังนี้

1. Accumulator ตำแหน่งหน่วยความจำภายในเท่ากับ 0E0H

รีจิสเตอร์นี้มีขนาด 8 บิท เป็นรีจิสเตอร์ที่ใช้มาก ซึ่งในรหัสคำสั่งช่วยจำจะอ้างถึงรีจิสเตอร์นี้โดยใช้สัญลักษณ์ A เช่น MOV A , #15H คำสั่งที่จะอ่านหรือเก็บข้อมูลกับหน่วยความจำภายนอกจะต้องกระทำผ่านรีจิสเตอร์นี้เท่านั้น เช่น MOVX @R0 , A หรือ MOVX A ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

@R0 เป็นต้น และข้อมูลที่อยู่ภายในรีจิสเตอร์นี้ ก็สามารถที่จะให้โปรแกรมตรวจสอบเพื่อกระโดดการทำงานไปยังตำแหน่งอื่นได้ เช่น JZ rel

2. B Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0F0H

เป็นรีจิสเตอร์ขนาด 8 บิท ที่ใช้ในคำสั่งการคูณ (MUL AB) และคำสั่งการหาร (DIV AB) เท่านั้น โดยรีจิสเตอร์ B นี้จะเก็บตัวคูณและผลลัพธ์บิท 8 ถึง 15 ในคำสั่งการคูณ ส่วนในคำสั่งการหารนั้นรีจิสเตอร์ B จะเก็บตัวหารและผลการหาร การเขียนข้อมูลไปยังรีจิสเตอร์นี้จะต้องใช้คำสั่งเคลื่อนย้ายข้อมูลไปยังตำแหน่ง 0F0H เช่น MOV 0F0H , 25H จะเป็นการกำหนดค่า 25H ให้กับรี-จีสเตอร์ B

3. Program status word ตำแหน่งหน่วยความจำภายในเท่ากับ 0D0H

เป็นรีจิสเตอร์ขนาด 8 บิท ที่แต่ละบิทจะบอกสภาวะต่าง ๆ แต่ละบิทของ PSW จะสามารถกำหนดให้เป็น 1 หรือ 0 ได้ด้วยคำสั่ง SETB หรือ CLR ตามลำดับค่าตำแหน่งบิท 0 ถึงบิท 7 ของรีจิสเตอร์ PSW เท่ากับ D0H ถึง D7H ตามลำดับ

4. Stack Pointer ตำแหน่งหน่วยความจำในเท่ากับ 081H

เป็นรีจิสเตอร์ขนาด 8 บิท รีจิสเตอร์นี้จะใช้ตำแหน่งหน่วยความจำภายใน 89C52 ที่ใช้เก็บตำแหน่ง (Address) เดิมของโปรแกรมก่อนทำงานคำสั่ง CALL หรือตำแหน่งที่จะใช้เก็บข้อมูลด้วยคำสั่ง PUSH และ ตำแหน่งที่จะอ่านข้อมูลออกมาในคำสั่ง POP เมื่อทำการรีเซ็ท 89C52 โดยการป้อนสัญญาณสภาวะลอจิก 1 เข้าไปที่ขา RST ของ 89C52 จะทำให้ข้อมูลในรีจิสเตอร์นี้มีค่าเป็น 07H หมายความว่ารีจิสเตอร์ SP ชี้หน่วยความจำภายใน 89C52 ที่ตำแหน่ง 07H ค่าของ SP จะเปลี่ยนแปลงไปโดยการใช้คำสั่งเคลื่อนย้ายข้อมูลหรือการทำงานของคำสั่ง PUSH , POP และ CALL

5. Data Pointer Register ตำแหน่งหน่วยความจำภายในเท่ากับ 82H และ 83H

รีจิสเตอร์ DTPR มีขนาด 16 บิท หน้าที่ของรีจิสเตอร์นี้ก็คือใช้สำหรับชี้ตำแหน่งในหน่วยความจำรีจิสเตอร์ DPTR นี้สามารถใช้อ้างถึงตำแหน่งหน่วยความจำได้สูงสุด 60×1024 ตำแหน่ง เช่นคำสั่ง MOVX A , @DPTR หรือ ใช้ชี้ตำแหน่งโปรแกรมที่ต้องการกระโดดข้ามไปทำงาน เช่นคำสั่ง JMP @A+DPTR รีจิสเตอร์ DPTR นี้ประกอบด้วยรีจิสเตอร์ขนาด 8 บิท 2 ตัวคือ DPH ซึ่งอยู่ที่ตำแหน่ง 83H และ DPL ซึ่งอยู่ตำแหน่ง 82H ในหน่วยความจำสำหรับข้อมูลภายใน 89C52 ดังนั้นการแก้ไขข้อมูลในรีจิสเตอร์ DPTR จึงทำได้ทั้งทีละ 16 บิท เช่น คำสั่ง MOV DPTR , #DATA16 หรือจัดการทีละ 8 บิทโดยการแก้ไขข้อมูลใน DPH หรือ DPL ด้วยคำสั่ง MOV 83H , #DATA8 หรือ MOV 83H , #DATA8

6. PORT 0 ถึง 3 ตำแหน่งหน่วยความจำภายในเท่ากับ 80H, 90H, 0A0H, 0B0H

Special Function Register ชื่อ P0 , P1 , P2 และ P3 เป็นรีจิสเตอร์ขนาด 8 บิท ของหน่วยความจำสำหรับข้อมูลภายใน 89C52 ที่ตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ทนั้น ๆ ของ 89C52 ข้อมูลที่เขียนออกไปจะถูก LATCH ค้างไว้และปรากฏที่แต่ละบิทของพอร์ท เช่น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV 80H , #18H จะปรากฏสภาวะลอจิก LLLHHLLL ที่ขาบิท 7 ถึง 0 ของพอร์ท 0 ตามลำดับ ในการอ่านข้อมูลจาก
รีจิสเตอร์แต่ละตัวก็จะเป็นการอ่านสภาวะลอจิกของสัญญาณที่ปรากฏอยู่ที่แต่ละขาของพอร์ทนั้น เช่น MOV A , 80H เป็นการอ่านสภาวะลอจิกจากพอร์ท 0 เข้ามายัง Accumulator การอ่านข้อมูลจากพอร์ทจะต้องเขียนข้อมูล 11111111B ไปไว้ที่พอร์ทนั้น ๆ เสียก่อน ทุกบิทของพอร์ท 0 ถึง 3 จะสามารถแก้ไขเปลี่ยนแปลงได้โดยใช้คำสั่ง SETB bit และ CLR bit

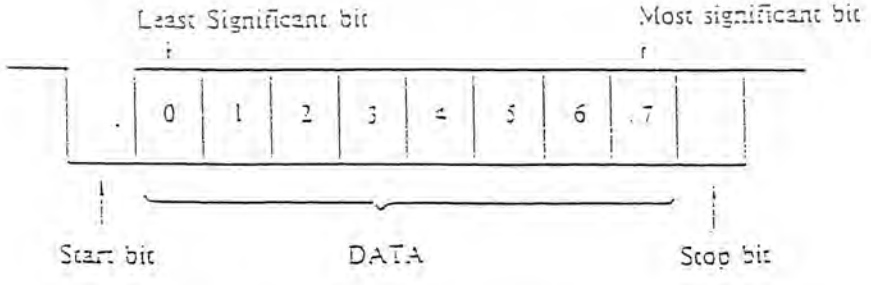
7. Serial Data Buffer ตำแหน่งหน่วยความจำภายในเท่ากับ 99H

รีจิสเตอร์นี้มีขนาด 8 บิทและมีตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 89C52 เท่ากับ 99H โครงสร้างภายในแล้วรีจิสเตอร์นี้มี 2 ตัว ที่มีชื่อเดียวกัน ตัวหนึ่งสำหรับเก็บข้อมูลที่จะส่งแบบอนุกรมออกจาก 89C52 และอีกตัวหนึ่งสำหรับข้อมูลแบบอนุกรมที่เข้ามา ดังนั้น Serial Port ของ 89C52 จึงเรียกว่ามีการทำงานแบบ Full Duplex เพราะสามารถส่งและรับข้อมูลได้ในเวลาเดียวกันเนื่องจากรีจิสเตอร์สำหรับส่งและรับแยกออกจากกัน ข้อมูลที่ต้องการจะส่งออกก็ให้เขียนไปยังรีจิส-เตอร์ SBUF แล้วสั่งงานให้ส่งข้อมูลออกมา ข้อมูลในรีจิสเตอร์จะเริ่มส่งออกโดยเริ่มจากบิท 0 ถึง 7 ตามลำดับ ถ้าข้อมูลมีข้อมูลเข้ามาทางขา RXD ก็จะถูกเก็บไปไว้ในรีจิสเตอร์นี้โดยถือว่าข้อมูลบิทแรกที่เข้ามาคือบิท 0

Serial Port จะสามารถกำหนดให้การทำงานรับ-ส่งข้อมูลแบบอนุกรมได้ 4 โมด (MODE) โดยการกำหนดในรีจิสเตอร์ SCON (Serial Port Control Register) ซึ่งจะอธิบายต่อไปในข้อ 5.18 แต่ละโมดการทำงานของ Serial Port มีดังนี้

MODE 0 : ในโมดนี้ จะมีการรับหรือส่งข้อมูลแบบอนุกรมทางขา RXD และ ขา TXD จะส่งสัญญาณ Clock ที่ใช้สำหรับเลื่อน (Shift) ข้อมูล 1 ชุด ของข้อมูลจะประกอบด้วยข้อมูล 8 บิทเท่านั้น และจะเริ่มการรับ-ส่งข้อมูลจากบิท 0 จนถึงบิท 7 ตามลำดับ อัตราการส่งข้อมูลแบบอนุกรมจะเท่ากับ 1/12 เท่าของความถี่สัญญาณนาฬิกาที่ใช้กับ 89C52

MODE 1 : ข้อมูลที่รับ-ส่ง 1 ชุดในโมดนี้จะมี 10 บิท ผ่านทางขา RXD และ TXD ตามลำดับ เริ่มต้นการรับส่งข้อมูลด้วย Start bit 1 บิท (ลอจิกเป็น 0) , ข้อมูล 8 บิท (เริ่มจากบิท 0) , Stop bit 1 บิท (ลอจิก 0) การส่งข้อมูลโมดนี้มีดังรูปที่ 3.22

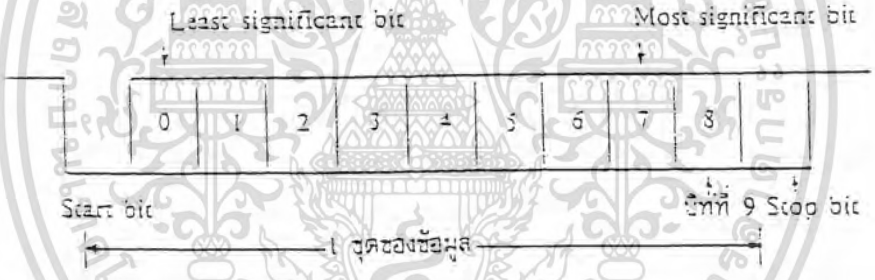


รูปที่ 3.22 ชุดข้อมูลอนุกรมในโมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรับข้อมูลอนุกรมเข้ามาข้อมูล 8 บิทจะถูกเก็บในรีจิสเตอร์ SBUF และ Stop bit จะถูกเก็บไปที่บิท RB8 ในรีจิสเตอร์ SCON ในการส่งข้อมูลออกก็จะเขียนข้อมูลที่ต้องการส่งไปยังรีจิสเตอร์ SBUF อัตราการส่งข้อมูลในโมดนี้ สามารถกำหนดได้ตามต้องการโดยจะขึ้นกับการเกิด Overflow ใน Timer 1

MODE 2 : การรับ-ส่งข้อมูลของโมด 2 1 ชุดจะมี 11 บิท ข้อมูลจะส่งออกผ่านทางขา TXD และรับเข้ามาทางขา RXD ข้อมูลแต่ละชุดจะเริ่มต้นด้วย Start bit 1 บิท , ข้อมูล 8 บิท (เริ่มจาก
บิท 0) , ข้อมูลบิทที่ 9 จำนวน 1 บิท และ Stop bit อีก 1 บิทข้อมูลบิทที่ 9 ที่จะส่งออกนี้สามารถกำหนดได้ว่าจะให้เป็น 1 หรือ 0 พาริตี้ของข้อมูลไปเป็นบิทที่ 9 เพื่อว่าปลายทางรับข้อมูลแล้วจะได้ใช้ตรวจสอบว่าข้อมูลที่รับเข้ามา 8 บิทมีพาริตี้บิทตรงกับบิทที่ 9 หรือไม่ ถ้าไม่ตรงก็แสดงว่ามีข้อผิดพลาดเกิดขึ้นระหว่างการส่งข้อมูลเข้ามานั้นข้อมูลบิทที่ 9 ก็จะถูกนำไปเก็บในบิท RB8 ของรีจิสเตอร์ SCON ชุดข้อมูลที่รับ-ส่งจะมีดังรูปที่ 3.23



รูปที่ 3.23 ชุดข้อมูลอนุกรมในโมด 2

อัตราการส่งข้อมูลจะกำหนดให้เป็น 1/32 หรือ 1/64 เท่า ของความถี่สัญญาณนาฬิกาที่ใช้กับ 89C52 โดยการกำหนดบิท SMOD ในรีจิสเตอร์ PCON

MODE 3 : การส่งข้อมูลโมดนี้ 1 ชุดมี 11 บิท เหมือนกับโมด 2 ทุกประการแตกต่างกันตรงอัตราการส่งข้อมูลเท่านั้น คืออัตราการส่งข้อมูลในโมด 3 นี้ สามารถกำหนดได้ตามต้องการโดยจะขึ้นกับการเกิด Overflow ใน Timer 1 เหมือนกับโมด 1

8. SCON (Serial Port Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 98H

รีจีสเตอร์ SCON มีขนาด 8 บิท ใช้สำหรับควบคุมการส่งและรับข้อมูลผ่านทาง Serial Port แต่ละบิทของข้อมูลในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCON : SERIAL PORT CONTROL REGISTER, BIT ADDRESSABLE

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier (NOTE 1).
SM1	SCON.6	Serial Port mode specifier (NOTE 1).
SM2	SCON.5	Enables the multiprocessor communication feature in mode 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 3).
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1 :

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc/12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc/64 OR Fosc/32
1	1	3	9-Bit UART	Variable

SERIAL PORT SET-UP : Table 9

MODE	SCON	SM2 VARIATION
0	10H	Single Processor Environment (SM2 = 0)
1	50H	
2	90H	
3	D0H	
0	NA	Multiprocessor Environment (SM2 = 1)
1	70H	
2	B0H	
3	F0H	

รูปที่ 3.24 Serial Port Control Register (SCON)

ในรูปที่ 3.24 บิท R1 จะเป็นชื่อของบิท 0 และ SNO จะเป็นบิท 7 ของรีจิสเตอร์ SCON ซึ่งความหมายหรือการทำงานของแต่ละบิทมีดังนี้

RI Receive Interrupt Flag

บิทนี้จะถูกกำหนดโดยฮาร์ดแวร์ให้มีค่าเป็น 0 หรือ 1 โดยที่ในการรับข้อมูลโมด 0 นั้น บิท RB8 จะมีค่าเป็น 1 เมื่อมีข้อมูลเข้ามาครบทั้ง 8 บิท ส่วนในโมดอื่นบิท RB8 จะเป็น 1 ก็ต่อเมื่อข้อมูลเข้ามาถึงเวลาครึ่งหนึ่งของ Stop bit (ยกเว้นบางกรณีให้ดูที่เรื่องบิท SM2 ของรีจิสเตอร์ SCON) บิทนี้จะสามารถ Clear ให้มีค่าเป็น 0 ได้ โดยใช้คำสั่ง CLR bit โดยค่าตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของบิทมีค่าเท่ากับ 98H บิทนี้มีประโยชน์ให้รู้ว่าข้อมูลได้เข้ามาอยู่ใน SBUF ครบทั้งชุดแล้ว พร้อมที่ CPU จะอ่านไปเก็บในหน่วยความจำต่อไป หรืออาจกำหนดค่าในรีจิสเตอร์ IE และ IP เพื่อเมื่อมีข้อมูลเข้ามาทางพอร์ทอนุ-กรมแล้วจะทำให้เกิดการขัดจังหวะ (Interrupt) การทำงานของโปรแกรมหลัก (Main Program) แล้วกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ (Interrupt Service Routine)

TI Transmit Interrupt Flag

ค่าในบิท TI จะถูกกำหนดให้เป็น 1 หรือ 0 ด้วยฮาร์ดแวร์ โดยในการส่งข้อมูลแบบอนุกรมโมด 0 บิทนี้จะเป็น 1 เพื่อจะบอกว่าการส่งข้อมูลในรีจิสเตอร์ SBUF ออกไปทางพอร์ทอนุกรมครบทั้ง 8 บิท แต่ถ้าเป็นการส่งข้อมูลแบบอนุกรมในโมดอื่นจะทำให้ข้อมูลในบิท TI เป็น 1 เมื่อเริ่มการส่ง Stop bit ข้อมูลบิทนี้จะสามารถ Clear เป็น 0 ได้ด้วยคำสั่ง CLR bit โดยที่ค่าตำแหน่งของบิทนี้เท่ากับ 99H บิทนี้ยังมีประโยชน์เพื่อบอกว่าการส่งข้อมูลจาก SBUF ออกไปทางพอร์ทอนุกรมนั้นสิ้นสุดแล้วพร้อมที่จะให้โปรแกรมเขียนข้อมูลลงไปยัง SBUF สำหรับการส่งออกต่อไปได้ นอกจากนี้การกำหนดค่าในรีจิส-เตอร์ IE และ IP ยังสามารถที่จะกำหนดให้เกิดการขัดจังหวะการทำงานของโปรแกรมได้เมื่อบิทนี้ถูกฮาร์ดแวร์ทำให้มีค่าเป็น 1

RB8

เมื่อมีการกำหนดให้รับข้อมูลในโมด 2 และ 3 จะใช้บิทนี้สำหรับเก็บข้อมูลบิทที่ 9 ที่เข้ามาทางพอร์ทอนุกรม ส่วนในโมด 1 นั้นบิทนี้จะเก็บ Stop bit ซึ่งมีค่าเป็น 1 นั่นเอง ในโมด 0 บิทนี้จะไม่ถูกใช้งาน ค่าตำแหน่งของบิทนี้คือ 9AH

TB8

ในการส่งข้อมูลแบบอนุกรมโมด 2 และ 3 จะใช้บิทนี้เก็บข้อมูลบิทที่ 9 ส่วน โมดอื่นจะไม่ใช้งานบิทนี้การกำหนดค่าในบิทนี้สามารถทำได้โดยใช้คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่งของบิทนี้คือ 9BH

REN Receive Enable

เป็นบิทที่จะใช้กำหนดให้ทำการรับข้อมูลเข้ามาจากทางพอร์ทอนุกรม (Serial Port) หรือไม่ถ้าบิทนี้เป็น 1 ก็จะรับข้อมูลเข้ามา แต่ถ้าเป็น 0 ก็จะไม่รับข้อมูลที่ขา RXD เข้ามา การให้บิทนี้เป็น 1 หรือ 0 ทำได้โดยใช้ คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่งของบิทนี้คือ 9CH

SM2

เป็นบิทสำหรับควบคุมการทำงานของฮาร์ดแวร์ที่จะทำให้บิท RI เป็น 1 หรือไม่ ในกรณีที่บิท SM2 เป็น 0 ค่าในบิท RI ก็จะเป็นไปตามที่ได้อธิบายมาแล้วในเรื่องบิท RI แต่ถ้าบิท SM2 = 1

โมด 2 และ 3 ซึ่งปกติแล้วบิท RI จะเป็น 1 เมื่อข้อมูลบิทที่ 9 เข้ามา แต่เมื่อ SM2 เป็น 1 แล้ว RI จะเป็น 1 ก็ต่อเมื่อข้อมูลบิทที่ 9 ที่เข้ามามีค่าเป็น 1 ถ้าข้อมูลบิทที่ 9 เข้ามาเป็น 0 จะไม่ทำให้บิท RI มีค่าเป็น 1 (คือบิท RI จะเป็น 0)

ในโมด 1 บิท RI มีค่าเป็น 1 เมื่อข้อมูล Stop bit เข้ามายังพอร์ตอนุกรมถูกต้องแต่ถ้า Stop bit ไม่เข้ามายังพอร์ตอนุกรมอันอาจเกิดจากปัญหาในการส่งข้อมูลแล้วบิท RI จะมีค่าเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM0, SM1

เป็น 2 บิท ที่ใช้งานร่วมกันเพื่อกำหนดโมดของการรับ-ส่งของพอร์ตอนุกรม ค่าใน 2 บิทนี้จะกำหนดโมดได้ดังนี้

SM0	SM1	MODE	Description
0	0	0	Shift register
0	1	1	8-bit UART
1	0	2	9-bit UART
1	1	3	9-bit UART

การทำงานของแต่ละโมดจะมีดังในข้อ 5.1.7

9. TIMER Register TH0, TL0, TH1, TL1

ตำแหน่งหน่วยความจำภายในเท่ากับ 8CH, 8AH, 8DH 8BH

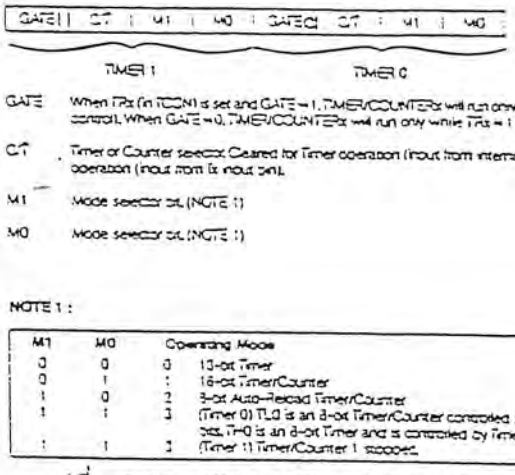
ใน 89C52 จะมีวงจร Timer อยู่ 2 ชุด คือ Timer 0 และ Timer 1 (8052 จะมี Timer 2 อีก 1 ชุด) ใน Timer แต่ละชุดจะมีรีจิสเตอร์ขนาด 8 บิทอยู่ 2 ตัว เพื่อเก็บค่าการนับของ Timer ได้สูงสุดถึง 16 บิทใน Timer 0 รีจิสเตอร์นี้คือ TH0 , TL0 และ Timer 1 คือรีจิสเตอร์ TH1 , TL1 TLx (x หมายถึง 0 หรือ 1) จะเก็บค้าของการนับ 8 บิทล่าง และ THx จะเก็บค่าของการนับ 8 บิทบน ผู้ใช้จะสามารถการทำงานของวงจร Timer ในโมด Timer หรือโมด Counter ได้โดยการกำหนดในรีจิสเตอร์ชื่อ TMOD (Time/Counter Mode Control Register) การทำงานเป็น Timer นั้นจะให้รีจิสเตอร์ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิล (Cycle) ของสัญญาณนาฬิกาที่ผ่านวงจรหาร 12 ดังรูปที่ 3.26 เมื่อการนับครบถึงค่าสูงสุดที่รีจิสเตอร์ TLx และ THx จะเก็บได้คือค่า FFFFH แล้วยังนับต่อไปค่าที่ได้จากการนับจะเป็น 0000H ทำให้เกิดการ SET บิทบางบิทในรีจิสเตอร์ TCON เพื่อบอกสภาวะ Timer Overflow นี้ในการให้วงจร Timer ทำงานเป็น Counter ก็คือการใช้รีจิสเตอร์ TLx และ THx ทำการนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 ของ 89C52 สัญญาณที่เข้ามาทางขาลอจิก 0 หรือ 1 เป็นแบบ TTL คือลอจิก 0 จะต้องมีโวลเตจไม่เกิน 0.6 โวลท์ และลิจิก 1 จะต้องมีโวลเตจมากกว่า 2.4 โวลท์

10. TMOD Timer/ Counter mode register

ตำแหน่งหน่วยความจำภายในเท่ากับ 89H

TMOD เป็นรีจิสเตอร์ขนาด 8 บิทที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer1 แต่ละบิทในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 TMOD Timer/ Counter mode register

ในรูปที่ 3.25 M0 เป็นชื่อของบิท 0 และ GATE ทางซ้ายสุดเป็นชื่อของบิท 7 รีจิสเตอร์นี้แบ่งข้อมูลออกเป็น 2 ชุด ชุดละ 4 บิท คือบิท 0-3 ใช้สำหรับควบคุมการทำงานของ Timer 0 และบิท 4-7 ใช้ควบคุมการทำงานของ Timer 1 หน้าที่ในการควบคุม Timer ของแต่ละบิทที่มีชื่อเดียวกันจะเหมือนกัน

GATEx เป็นบิทที่ใช้ควบคุมให้ Timer x ทำงานหรือไม่ ถ้าบิทนี้ของ Timer x ถูกตั้งเป็น 1 จะทำให้ Timer ทำงานก็ต่อเมื่อที่ขา INTx มีสภาวะลอจิกเป็น 1 และบิท TRx ในรีจิสเตอร์ TCON เป็น 1 ด้วย

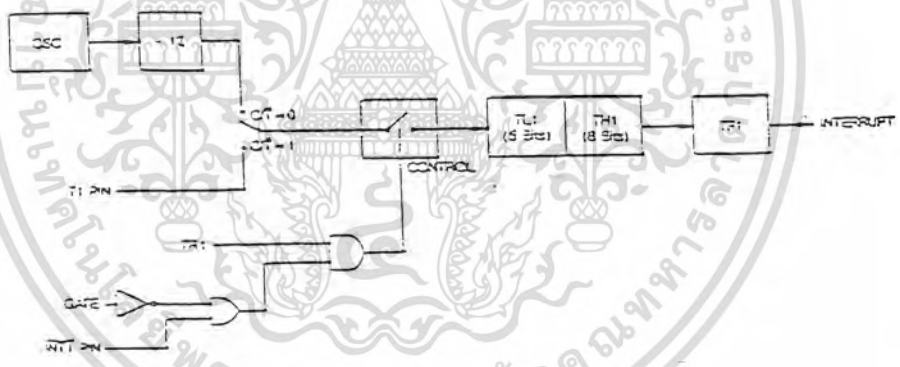
$\overline{C/T}$ บิทนี้ใช้สำหรับเลือกการทำงานของ Timer ว่าจะใช้เป็น Timer หรือ Counter ถ้าบิทนี้เป็น 1 ก็หมายความว่าเลือกการทำงานเป็น Counter ซึ่งจะนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา Tx

M1, M0 เป็น 2 บิทที่ใช้ร่วมกันเพื่อเลือกโมดการทำงานของ Timer การทำงานโมด 0 , 1 และ 2 ของ Timer 0 จะเหมือนกับ Timer 1 แต่ในโมด 3 การทำงานของทั้งสองจะต่างกัน ค่าใน M1 และ M0 จะเลือกโมดการทำงานดังนี้

M1	M0	การทำงาน
0	0	โมด 0 รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 13 บิท ค่าจากการนับ 8 บิทบนมาจาก 8 บิทของ Thx และอีก 5 บิทล่างมาจากค่า 5 บิทล่างของรีจิสเตอร์ TLx โดยที่ 3 บิทบน ของ TLx จะไม่ต้องสนใจเลย
0	1	โมด 1 รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 16 บิท ค่าจากการนับ 8 บิทบนมาจาก 8 บิทของ THx และค่าจากการนับ 8 บิทล่างอยู่ในรีจิสเตอร์ TLx
1	0	โมด 2 ในการนับของรีจิสเตอร์ TLx ขนาด 8 บิทเมื่อนับถึงค่าสูงสุดคือ FFH เมื่อทำการนับต่อไปจะเกิดการ Overflow แล้วก็จะ "Reload"เอาข้อมูลจาก THx เข้าไปยัง TLx เพื่อเป็นค่าเริ่มต้นในการนับครั้งต่อไป
1	1	โมด 3 การทำงานของ Timer 0 และ timer 1 จะต่างกันดังที่จะกล่าวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของแต่ละโมดจะมีรายละเอียดดังนี้



รูปที่ 3.26 Timer mode 0 : 13 bit count

รูปที่ 3.26 เป็นไดอะแกรมของวงจร Timer ภายใน 89C52 ที่ทำงานในโมด 0 ซึ่ง Timer 0 และ Timer 1 ก็จะมีการทำงานเหมือนกันทุกประการ ในการอธิบายนี้จะใช้วงจรของ Timer 1 จากรูปจะเห็นสวิตช์ $C/\overline{T}$ ซึ่งถ้ากำหนดค่าในบิท $C/\overline{T}$ ของ TMOD เป็น 0 จะทำให้สวิตช์อยู่ในตำแหน่งบนเพื่อให้สัญญาณนาฬิกาที่ออกจากวงจรออสซิลเลเตอร์ผ่านวงจรหาร 12 ไปยังสวิตช์

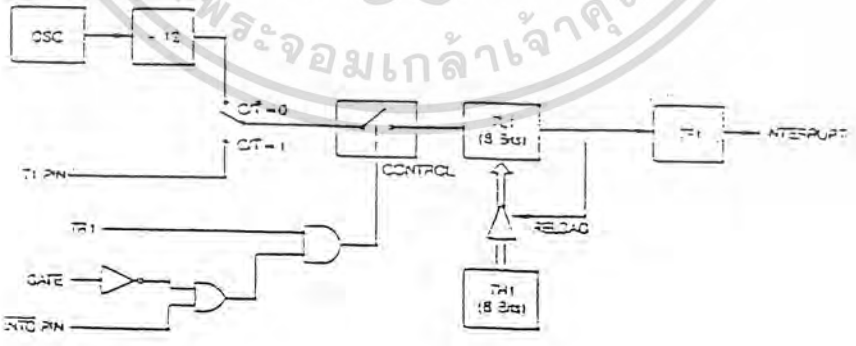
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control ถ้าออสซิลเลเตอร์ผลิตสัญญาณนาฬิกาความถี่ 12 เมกกะเฮิรตซ์ก็จะมีสัญญาณความถี่ 1 เมกกะเฮิรตซ์ ออกจากวงจรหาร 12 ถ้าบิท C/$\overline{T}$ เป็น 1 จะทำให้สวิตช์ C/$\overline{T}$ อยู่ในตำแหน่งข้างล่าง เพื่อให้สัญญาณที่เข้ามาทาง T (หรือ T0 ถ้าเป็น Timer 0) ผ่านไปยังสวิตช์ Control สัญญาณที่เข้ามายังสวิตช์ Control จะส่งผ่านไปยังวงจรนับหรือไม่ก็ขึ้นอยู่กับสัญญาณที่ออกมาจาก AND GATE ถ้าบิท TR1 (หรือ TR0 ถ้าเป็น Timer 0) ในรีจิสเตอร์ TCON เป็น 0 จะทำให้สภาวะของสัญญาณที่ออกจาก AND GATE เป็น 0 เสมอ และจะไม่มีสัญญาณใดออกจากสวิตช์ Control ไปยังวงจรนับเลย รีจิสเตอร์ TL1 และ TH1 จะไม่ทำงาน แต่ถ้าบิท TR1 เป็น 1 จะทำให้สภาวะของสัญญาณออกจาก AND GATE ไปควบคุมสวิตช์ Control ขึ้นอยู่กับสภาวะของสัญญาณ $\overline{INT1}$ (หรือ $\overline{INT0}$ ถ้าเป็น Timer 0) และข้อมูลที่บิท GATE ของรีจิสเตอร์ TMOD ถ้าบิท GATE เป็น 0 หรือสัญญาณที่ขา INT1 มีสภาวะลอจิกเป็น 1 จะทำให้สัญญาณควบคุมสวิตช์ Control เป็น 1 ทำให้มีสัญญาณออกไปยังตัวนับรีจิสเตอร์ TL1 และ TH1 (หรือ TL0 และ TH0 ถ้าเป็น Timer 0) รีจิสเตอร์ TL1 จะทำการนับโดยมีการนับเพียง 5 บิท เท่านั้น (ทำหน้าที่เป็นวงจรPrescalar ขนาด 5 บิท) ซึ่งนับได้ตั้งแต่ 0 ถึง 31 เมื่อ TL1 นับสัญญาณที่ออกจากสวิตช์ Control ครบ 32ไซเคิลจะมีสัญญาณส่งไปยัง TH1 ไซเคิล บิท 5 ถึง 7 ของ TLx ที่ไม่ได้ใช้งานก็จะไม่ต้องสนใจการทำงานของ Timer 0 และ Timer 1 ในโมดนี้จะเหมือนกับการทำงานของ Timer ในไมโครคอนโทรลเลอร์เบอร์ 8084 ทุกประการ

โมด 1

ในโมดนี้จะมีการทำงานของวงจรภายในของ Timer 0 หรือ 1 เหมือนกับโมด 0 ทุกประการแตกต่างกันที่ TLx จะถูกใช้งานทั้ง 8 บิท ทำให้ผลการนับใน TLx และ THx จะมีถึง 16 บิท

โมด 2



รูปที่ 3.27 Timer mode 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

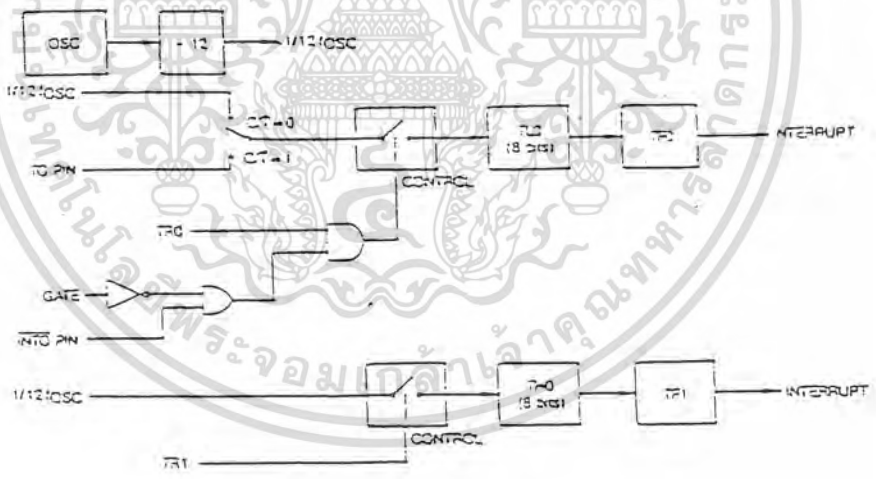
ในรูปที่ 3.27 เป็นไดอะแกรมของวงจร Timer 1 ใน 89C52 ที่ทำงานโมด 2 Timer 0 และ Timer 1 มีการทำงานในโมด 2 เหมือนกันโดยจะสามารถกำหนดให้ทำหน้าที่เป็น Timer หรือ Counter ได้โดย

บิท C/T และควบคุมการนับได้โดยข้อมูลในบิท TR1 และ GATE ในรีจิสเตอร์ TMOD กับสัญญาณที่เข้า INTx เมื่อเริ่มการทำงานข้อมูลในรีจิสเตอร์ TH1 จะถูกโหลด (Load) ไปยังรีจิสเตอร์ TL1 ทำให้

รีจิสเตอร์ TH1 และ TL1 มีค่าเหมือนกัน เมื่อเกิดการนับจำนวนไซเคิลของสัญญาณที่ออกจากสวิตช์Control จะทำให้ค่าจากกการนับในรีจิสเตอร์ TL1 เพิ่มขึ้นไปเรื่อย ๆ ทีละ 1 จนถึง 0FFH ในการนับครั้งต่อไปจะทำให้บิท TF1 ในรีจิสเตอร์ TCON ไม่เป็น 1 และข้อมูลในรีจิสเตอร์ TH1 จะถูกโหลดไปยังรีจิสเตอร์ TL1 เพื่อเป็นค่าเริ่มต้นการนับต่อไป

โมด 3

การทำงานโมด3ของTimer 0 และ1 ต่างกันTimer1ในโมด3 จะไม่ทำงานtimer0 ในโมด 3 จะทำงานเป็นตัวนับที่เสมือนมีตัวนับ 8 บิทอยู่ 2 ตัว คือ TL0 และ TH0 ทำงานแยกกันดังรูปที่ 3.28



รูปที่ 3.28 Timer 0 mode 3

รู

จิสเตอร์ TL0 จะเป็นตัวนับ 8 บิท ที่มีการนับสัญญาณจากออสซิลเลเตอร์หารด้วย 12 หรือนับสัญญาณที่เข้ามาทางขา T0 ขึ้นกับบิท ในรีจิสเตอร์ TMOD และการนับจะควบคุมโดยบิท TR0 และ GATE ในรีจิสเตอร์ TMOD กับสภาวะลอจิกของสัญญาณที่ขา INT0 เหมือนกับในการทำงานโมด 0 , 1 และ 2 แต่ค่าจากการนับนี้สูงสุดจะเพียง 255 เท่านั้น เมื่อค่าการนับเปลี่ยนจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0FFH เป็น 00H คือเกิดการ overflow จะทำให้บิท TF0 ถูก SET เป็น 1 และอาจเกิดการขัดจังหวะ(Interrupt) การทำงานของโปรแกรมได้ถ้ามีการกำหนดค่าในรีจิสเตอร์ IE และ IP ตัวนับอีกตัวคือ รีจิสเตอร์ TH0 จะทำงานในโมดของ Timer เท่านั้น คือจะนับจำนวนไซเคิลของสัญญาณที่ออกจากออสซิลเลเตอร์แล้วหารด้วย 12 การนับจะควบคุมได้ด้วยบิท TR1 ในรีจิสเตอร์ TMOD ถ้าบิทนี้เป็น 1 ก็จะมีสัญญาณเข้าไปยัง TH0 แต่ถ้าบิทนี้เป็น 0 ก็จะไม่มีสัญญาณเข้าไปยัง TH0

11. TCON Timer Control Regster ตำแหน่งหน่วยความจำภายในเท่ากับ 088H

รีจิสเตอร์ ขนาด 8 บิทนี้ใช้ควบคุมการทำงาน และบอกสภาวะของ Timer 0 และ Timer 1 แต่บิทของรีจิสเตอร์นี้ จะทำงานต่างกันดังรูปที่ 3.29

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External interrupt.
IE0	TCON.1	External interrupt 0 edge flag. Set by hardware when External interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External interrupt.

รูปที่ 3.29 TCON Timer Control Register

ในรูปที่ 3.29 T0 เป็นชื่อของบิท 0 และ TF เป็นชื่อของบิท 7 ในรีจิสเตอร์ TCON แต่ละบิทมีหน้าที่การทำงานดังนี้

IT0 Interrupt 0 เป็นบิทที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องจากสภาวะของสัญญาณที่เข้ามาทางขา $\overline{\text{INT0}}$

ถ้า IT0 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา $\overline{\text{INT0}}$ เปลี่ยนจาก 1 เป็น 0

ถ้า IT0 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา $\overline{\text{INT0}}$ เปลี่ยนเป็น 0

IE0 บิทนี้จะเป็น 1 ถ้าสัญญาณที่เข้ามาทางขา INT0 มีสภาวะลอจิกของสัญญาณตามที่กำหนดในบิท IT0 แล้วทำให้เกิดการขัดจังหวะโปรแกรม เมื่อเกิดการกระโดด ไปทำงานยังโปรแกรมตอบสนองการขัดจังหวะแล้ว จะทำให้บิทนี้กลับเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IT1 Interrupt 1 เป็นบิทที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องจากสภาวะของสัญญาณที่เข้ามาทางขา $\overline{INT1}$

ถ้า IT1 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา $\overline{INT1}$เปลี่ยนจาก 1 เป็น 0

ถ้า IT1 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT1 เป็น 0

IE0 บิทนี้จะเป็น 1 ถ้าสัญญาณที่เข้ามาทางขา $\overline{INT1}$ มีสภาวะลอจิกของสัญญาณ ตามที่กำหนดในบิท IT1 แล้วทำให้เกิดการขัดจังหวะโปรแกรม เหมือนกับ IT0 ที่ทำงานกับสัญญาณ $\overline{INT0}$

TR0 Timer 0 Run Control Bit บิทนี้ถ้าเป็น 0 Timer 0 ไม่ทำการนับสัญญาณไม่ว่ากรณี ใด ๆ ทั้งสิ้น แต่ถ้าบิทนี้เป็น 1 จะทำให้ Timer 0 ทำงานโดยขึ้นกับสัญญาณ GATE $\overline{INT0}$ ข้อมูลในบิทนี้จะสามารถ Set เป็น 1 หรือ Clear เป็น 0

TF0 Timer 0 Overflow flag บิทนี้จะเป็น 1 เมื่อการนับของรีจิสเตอร์ใน Timer 0 (TL0 หรือ TH0 ขึ้นอยู่กับโมดของการทำงาน) เกิด Overflow ขึ้น คือเอาการนับเพิ่มไปจนถึงค่าสูงสุดแล้วนับต่อไปทำให้ค่าการนับกลับมาเริ่มต้นใหม่ที่ 0 หรือค่า Reload เมื่อ 89C52 กระโดดไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะทำให้บิทนี้กลับเป็น 0

TR1 Timer 1 Run Control Bit การทำงานจะเหมือนกับการทำงานของบิท TR0 แต่บิทนี้จะทำงานกับ Timer 1

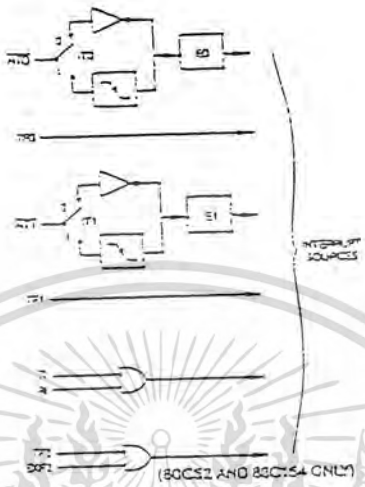
TF1 Timer 1 Overflow flag บิทนี้เหมือนกับบิท TF0 ต่างกันที่ขึ้นกับการทำงานของ Timer1

4 บิทแรกที่กล่าวมานั้นจะเกี่ยวข้องกับการขัดจังหวะ (Interrupt) ส่วน 4 บิทหลังนั้นได้กล่าวมาแล้วอย่างละเอียดในเรื่องโมดการทำงานของ Timer

ในขณะที่ Timer ทำงานในโมดของ Timer นั้น รีจิสเตอร์ที่ทำหน้าที่เป็นตัวนับจะมีค่าเพิ่มขึ้น 1 ทุก ๆ 1 ไซเคิลของเครื่อง ซึ่งเท่ากับ 12 คาบของสัญญาณจากออสซิลเลเตอร์ ในกรณีที่ Timer ทำงานเป็น Counter เพื่อนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 รีจิสเตอร์จะเพิ่มค่าไป 1 เมื่อมีการเปลี่ยนสถานะของสัญญาณที่ขาดังกล่าวจาก 1 เป็น 0 โดยวงจรภายใน 89C52 จะตรวจสอบสถานะของสัญญาณที่ขาดังกล่าวในช่วงเวลาเฟส 2 ของ Stat5 (S5P2) ในทุก ๆ 1 ไซเคิลของเครื่อง เช่นในเวลา S5P2 ครั้งหนึ่งพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 1 และในเวลา S5P2 ของไซเคิลของเครื่องถัดมาพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 0 ก็จะทำให้ค่าในรีจิสเตอร์ตัวนับเพิ่มค่าไป 1 แต่ละไซเคิลของเครื่องจะเท่ากับ 12 ไซเคิลของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณออสซิลเลเตอร์ ดังนั้นสัญญาณที่จะนับได้จะต้องเป็น 1 อย่างน้อยให้ถูกจับได้ใน 1 ไซเคิลของเครื่องและเป็น 0 อย่างน้อย ก็ต้องให้ถูกตรวจสอบได้



รูปที่ 3.30 แหล่งกำเนิดสัญญาณจังหวะ

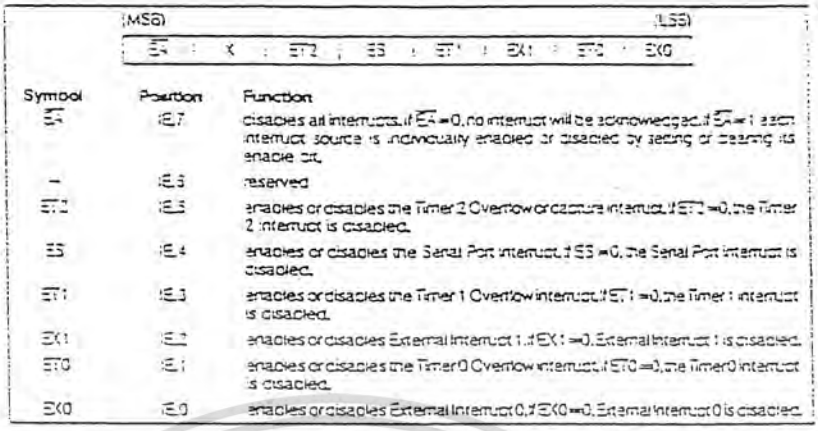
ในรูปที่ 3.30 จะสามารถทำให้เกิดการขัดจังหวะได้ 2 วิธี คือมีข้อมูลเข้ามาทางพอร์ทอนุกรมเก็บอยู่ที่รีจิสเตอร์ SBUF และกรณีที่ข้อมูลใน SBUF ส่งออกไปทางอนุกรมหมดแล้ว ไม่ว่าเกิดกรณีใด ๆ ก็ทำให้เกิดการขัดจังหวะขึ้น

สัญญาณภายนอกที่เข้ามายัง 89C52 ทางขา INT0 และ INT1 จะสามารถทำให้เกิดการขัดจังหวะการทำงาน 89C52 ได้ (สัญญาณที่ 1 และ 3 ในรูปที่ 3.30)โดยสภาวะของสัญญาณนั้นเปลี่ยนจาก 1 เป็น 0 หรือเมื่อสัญญาณนั้นเป็น 0 แล้วแต่การกำหนดในบิท IT0 และ IT1 ของรีจิสเตอร์ TCON จะทำให้บิท IE0 กับ IE1 เป็นตัวสร้างสัญญาณขัดจังหวะต่อไป

จาก Timer 0 และ Timer 1 เมื่อค่าการนับในแต่ละโหมดถึงค่าสูงสุดในโหมดนั้นแล้ว เมื่อทำการนับต่อไป ค่าการนับต่อไปจะเป็น 0 (หรืออาจเป็นค่าที่ Reload จาก THx ในโหมด 2) และทำให้บิท TF0 , TF1 เป็น 1 ซึ่งสัญญาณจาก 2 บิทนี้จะสามารถทำให้เกิดการขัดจังหวะได้เช่นกัน ดังเช่นสัญญาณขัดจังหวะที่ 2 และ 4 ในรูปที่ 3.30

แหล่งกำเนิดสัญญาณทั้ง 6 ที่สามารถทำให้เกิดการขัดจังหวะได้ 5 แบบนี้ ผู้ใช้สามารถกำหนดให้สัญญาณใดบ้างเกิดการขัดจังหวะเรียกว่า Enable หรือไม่ให้เกิดการขัดจังหวะเรียกว่า Disable โดยการกำหนดในรีจีสเตอร์ IE (Interrupt Enable Register) ซึ่งมี 8 บิท แต่ละบิทสามารถ Enable ให้ขัดจังหวะได้จากแต่ละสัญญาณ ดังรูปที่ 3.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(MSB)							(LSB)
EA	-	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
EA	IE.7	disables all interrupts. if EA=0, no interrupt will be acknowledged. if EA=1 each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
-	IE.6	reserved
ET2	IE.5	enables or disables the Timer 2 Overflow or capture interrupt. if ET2=0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt. if ES=0, the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. if ET1=0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External Interrupt 1. if EX1=0, External Interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. if ET0=0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt 0. if EX0=0, External Interrupt 0 is disabled.

รูปที่ 3.31 Interrupt Enable Register

ถ้าต้องการ Enable บิทใดก็ให้โปรแกรมกำหนดค่าในบิทนั้นเป็น 1 ถ้าค่าในบิทนั้นเป็น 0 หมายถึง Disable การ Disable จะทำให้มีการขัดจังหวะการทำงานของโปรแกรมเนื่องจากสัญญาณขอขัดจังหวะนั้น ๆ EX0 เป็นชื่อบิท 0 และ EA เป็นชื่อของบิท 7

EX0 บิทนี้ใช้สำหรับการ Enable สัญญาณที่เข้ามาทางขา INTO ให้เกิดการขัดจังหวะหรือไม่

ET0 Timer 0 Interrupt Enable Bit ข้อมูลบิทนี้จะใช้ Enable Disable สัญญาณขัดจังหวะที่มาจากวงจร Timer 0 (TF0)

EX1 บิทนี้จะใช้ Enable หรือ Disable สัญญาณที่เข้ามาทางขา INT1 ให้เกิดการขัดจังหวะหรือไม่

ET1 Timer 1 Interrupt Enable Bit บิทนี้จะใช้ Enable หรือ Disable สัญญาณขัดจังหวะจาก Timer 1 (TF1)

ES ข้อมูลในบิทนี้จะ Disable หรือ Enable การขัดจังหวะจาก Serial Port อันเนื่องจากมีข้อมูลเข้ามายัง SBUF หรือข้อมูลจาก SBUF ได้ส่งออกหมดแล้ว

ET2 Timer 2 Internal Enable Bit จะใช้เฉพาะใน 8052 และ 83152 เท่านั้น บิทนี้จะใช้ Enable หรือ Disable สัญญาณขอขัดจังหวะที่มาจาก Timer 2 (สัญญาณที่ 6 ในรูปที่ 3.30)

EA บิทนี้จะควบคุมทั้ง 6 บิทที่กล่าวมาแล้ว ถ้าข้อมูลในบิทนี้เป็น 0 จะเป็นการ Disable ทุกบิทที่กล่าวมาแล้ว ทำให้ไม่เกิดการขัดจังหวะโปรกแรมได้เลย แต่ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทนี้ป็น 1 การ Enable/Disable ใน 6 บิทที่กล่าวมาแล้วจะขึ้นกับข้อมูล รีจิสเตอร์นั้นคือ IP Interrupt Priority Register

การกำหนดให้บิทใด Enable หรือ Diasble นั้นจะเป็นไปโดยอิสระไม่ขึ้นแก่กัน จึงสามารถกำหนดให้บิทใดหรือมากกว่า 1 บิท Enable ก็ได้ ดังนั้น 89C52 จึงมี Register อีกตัวที่ใช้เลือกว่าถ้ามีสัญญาณขอการขัดจังหวะโปรแกรมเข้ามาพร้อมกันมากกว่า 1 แล้วจะทำให้โปรแกรมตอบสนองการขัดจังหวะอันใดก่อน รีจิสเตอร์นั้น คือ IP Interrupt Priority Register

13. IP Interrupt Priority register ตำแหน่งหน่วยความจำภายในเท่ากับ 0B8H

ในการตอบสนองต่อสัญญาณขัดจังหวะของ 89C52 นั้น ถ้าสัญญาณขัดจังหวะทั้งหมดเข้ามาพร้อมกัน 89C52 จะต้องเลือกทำงานโปรแกรมตอบสนองการขัดจังหวะโดยการตรวจสอบสัญญาณเรียงตามลำดับ ซึ่งเรียกว่าวิธีการ Polling สัญญาณจังหวะหนึ่งจะถูกตรวจสอบก่อนแล้วสัญญาณอื่น ๆ จะถูกตรวจสอบต่อมา ถ้าสัญญาณนั้นขอขัดจังหวะ 89C52 จะสร้างคำสั่ง CALL เป็นพิเศษขึ้นมาเพื่อไปทำงานโปรแกรมตอบสองการขัดจังหวะของสัญญาณนั้น เมื่อเสร็จสิ้นแล้วจะกลับมาทำงานในโปรแกรมเดิมก่อนการขัดจังหวะ ทำให้เสมือนว่าสัญญาณแต่ละสัญญาณมีลำดับความสำคัญไม่เท่ากัน สัญญาณขัดจังหวะจะมีลำดับความสำคัญดังนี้ โดยเรียงจากลำดับความสำคัญสูงสุดถึงต่ำสุด

1. 1E0
2. TF0
3. IE1
4. TF1
5. RI + TI

แต่ถ้าในการใช้งานบางครั้งจำเป็นที่จะต้องให้สัญญาณใดสัญญาณหนึ่งมีลำดับความสำคัญสูงสุด (Highest Priority) เพื่อจะทำงานโปรแกรมตอบสนองการขัดจังหวะได้ก่อนการขัดจังหวะของสัญญาณอื่น จะสามารถกำหนดลำดับความสำคัญของการขัดจังหวะได้ใหม่โดยการกำหนดข้อมูลในบิทของรีจิสเตอร์ IP (Interrupt Priority Register) ตามตำแหน่งของแต่ละบิทในรูปที่ 3.32

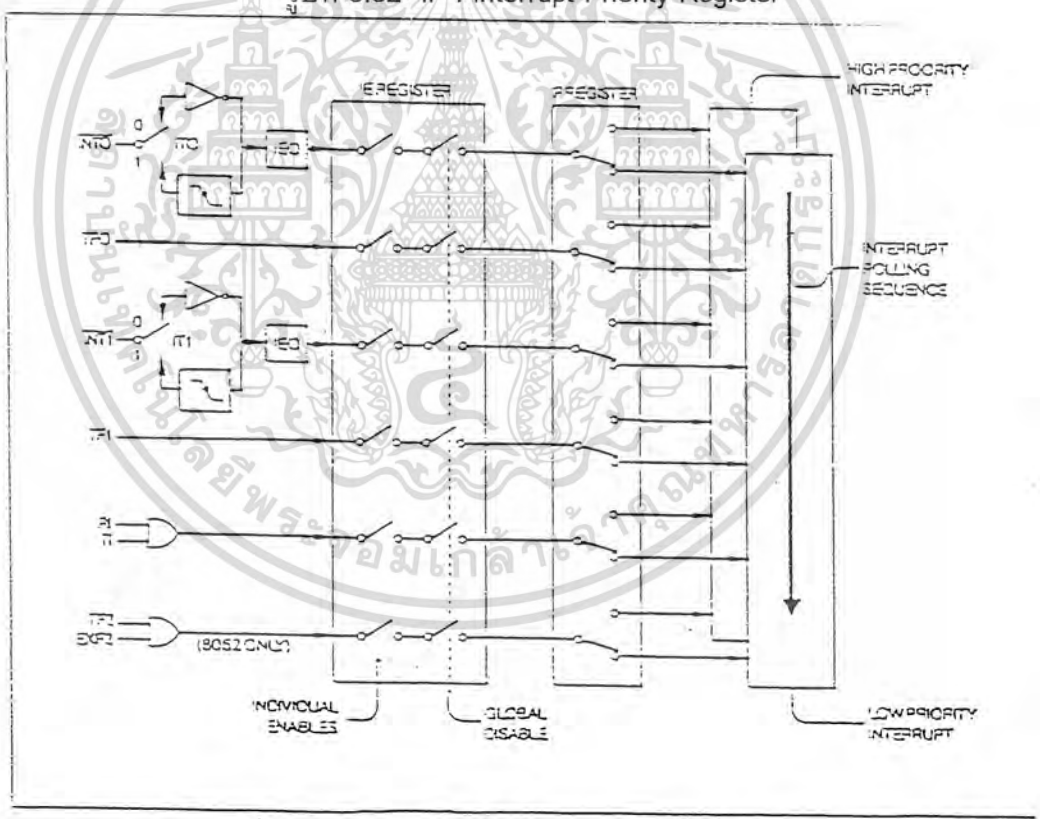
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(MSB) (LSB)

X	X	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
PCT	IP.7	PCT = 1, only one level
–	IP.6	reserved
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

รูปที่ 3.32 IP : Interrupt Priority Register



รูปที่ 3.33 ระบบการขัดจังหวะของ AT89C52

รูปที่ 3.33 เป็นแผนภาพแสดงระบบขัดจังหวะของ 8052 ซึ่งแตกต่างจากของ 89C52 ตรงที่ 8052 จะมีสัญญาณขัดจังหวะมาจาก TF2 , EFX2 คือชุดล่างในภาพ

ในรูปจะเห็นว่าแต่ละสัญญาณจะมีสวิตช์ควบคุมอยู่ 3 ตัว 2 ตัวแรกอยู่ในกรอบสี่เหลี่ยม IE Register และอีก 1 สวิตช์อยู่ในกรอบ IP Register สวิทช์ตัวแรกทางซ้ายสุดจะควบคุมข้อมูลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลแต่ละบิท บิท 0 ถึงบิท 5 ของรีจิสเตอร์ IE ถ้าข้อมูลเป็น 1 จะทำให้สวิตช์นั้นปิดวงจร (Closed circuit) การควบคุมสวิตช์ทางซ้ายสุดของแต่ละสัญญาณจะไม่ขึ้นแก่กัน (Individual) สวิตช์ที่ 2 ถัดมาของทุกสัญญาณจะควบคุมร่วมกันด้วยบิท EA ในรีจิสเตอร์ IE ถ้าบิทนี้เป็น 0 สวิตช์ที่ 2 ของทุกสัญญาณจะเปิดวงจร (Opened Circuit) ทำให้ไม่มีสัญญาณขอขัดจังหวะผ่านไปได้ สวิตช์ที่ 3 ทางขวาสุดจะใช้สำหรับเลือกว่าสัญญาณนั้นจะอยู่ในกลุ่มลำดับความสำคัญสูง (High Priority Interrupt) หรือลำดับความสำคัญต่ำ (Low Priority Interrupt)

ถ้าต้องการให้สัญญาณใดมีลำดับความสำคัญสูงก็ให้กำหนดบิทนั้นในรีจิสเตอร์ IP เป็น 1 สวิตช์ที่ 3 จะเลื่อนไปอยู่ในตำแหน่งบน ถ้าไม่ต้องการก็กำหนดให้บิทนั้นเป็น 0 บิทใดเป็น 1 เรียกว่าสัญญาณนั้นจะอยู่ในกลุ่มลำดับความสำคัญสูงและบิทใดเป็น 0 เรียกว่าสัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญต่ำ ถ้าในกลุ่มลำดับความสำคัญสูงมีเพียง 1 สัญญาณก็จะเรียกว่าสัญญาณนั้นมีลำดับความสำคัญสูงสุด ในกลุ่มลำดับความสำคัญเดียวกันก็จะมีการจัดลำดับความสำคัญเฉพาะกลุ่มโดยวิธี Polling เหมือนเดิม เช่นกรณีที่มีการกำหนดในบิทของรีจิสเตอร์ IP ให้มีลำดับความสำคัญสูงหรือต่ำเหมือนกันแล้ว เกิดมีความต้องการขอการขัดจังหวะจากสัญญาณนั้น ๆ มาพร้อมกัน 89C52 ก็จะทำงานในโปรแกรมตอบสนองการขัดจังหวะของ Timer 1 , External interrupt 1 และ Timer 0 พร้อมกัน 89C52 กำลังทำงานตอบสนองการขัดจังหวะของสัญญาณขัดจังหวะที่ลำดับความสำคัญต่ำอยู่ ถ้ามีสัญญาณขัดจังหวะที่มีลำดับความสำคัญสูงกว่าเกิดขึ้น การทำงานของโปรแกรมก็จะกระโดดไปทำงานในตำแหน่งโปรแกรมตอบสนองการขัดจังหวะของสัญญาณที่มีลำดับความสำคัญสูง เสร็จแล้วจึงกลับมาทำงานที่โปรแกรมตอบสนองการขัดจังหวะลำดับความสำคัญต่ำต่อไป แต่ละบิทของรีจิสเตอร์ IP นั้นจะบอกลำดับความสำคัญของแหล่งกำเนิดสัญญาณขัดจังหวะดังนี้

PX0 บิท 0 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะภายนอก 89C52 คือ INT0

PT0 บิท 1 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 0

PX1 บิท 2 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะภายนอกของ 89C52 คือ INT1

PT1 บิท 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 1

PT2 บิท 5 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 2 บิทนี้ใช้เฉพาะใน 8052 ที่มี Timer 2

PS บิท 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Serial Port ในกรณีที่มีข้อมูลเข้ามาหรือส่งข้อมูลออกสิ้นสุดแล้ว

บิทที่เหลือกจะไม่มีการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของการขัดจังหวะจะกล่าวต่อไปในหัวข้อ 3.9

14. PCON (Power Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 87H

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1,2 or 3.
HPD	PCON.5 (83C154 only)	Hard Power Down bit. Setting this bit allows CPU to enter in Power Down state on an external event (1 to 0 transition) on bit P1 (p.3-5) the CPU quit the Hard Power Down mode when bit P1 (p. 3-5) go high or when reset is activated.
SPD	PCON.5 (83C154 only)	Recover from idle or Power Down bit. When 0 SPD has no effect. When 1, SPD permits to exit from idle or Power Down with any non enabled interrupt source (except annex 2). In this case the program start at the next address. When interrupt is enabled the appropriate interrupt routine is serviced.
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

รูปที่ 3.34 PCON : Power Control Register

89C52 เป็นไมโครโปรดอนโทรลเลอร์ที่สร้างขึ้นด้วยเทคโนโลยี่ทั้งแบบ CHMOS และ HMOS ซึ่งแบบ CHMOS มีข้อดีตรงที่ใช้กำลังไฟต่ำกว่าแบบ HMOS ดังนั้นต่อไปในอนาคตจึงจะมีแต่เฉพาะรุ่น CHMOS เท่านั้น นอกจากนี้แล้ว 89C52 ยังมีข้อดีอีกตรงที่สามารถลดการใช้กำลังไฟลงได้โดยการทำงานใน Idle Mode และ Power Down Mode ใน Idle Mode นั้นสัญญาณนาฬิกาจากออสซิลเลเตอร์จะป้อนให้เฉพาะส่วน Interrupt , Serial Port และ Timer ในส่วนอื่นจะไม่มีสัญญาณนาฬิกาไปเลี้ยงแต่มีไฟเลี้ยงให้กับทุกส่วนในวงจร การใช้กำลังไฟจึงลดลงมาก ส่วนใน Power Down Mode นั้น ออสซิลเลเตอร์จะหยุดทำงานทำให้ไม่มีสัญญาณนาฬิกาไปเลี้ยงส่วนใด ๆ ในวงจรเลยแต่ข้อมูลภายในรีจิสเตอร์จะยังคงอยู่ไม่สูญหายไป รายละเอียดของแต่ละโหมดจะได้กล่าวต่อไป

การสั่งงานให้ 89C52 ทำงานในโหมดของ Idle หรือ Power Down จะสามารถทำได้โดยใช้กำหนดค่าในรีจิสเตอร์ PCON (Power Control Register) แต่ละบิทในรีจิสเตอร์ PCON มีดังนี้

IDC บิท 0 ถ้าบิทนี้ถูก Set ให้เป็น 1 89C52 จะเข้าสู่การทำงานใน Idle Mode ทันที

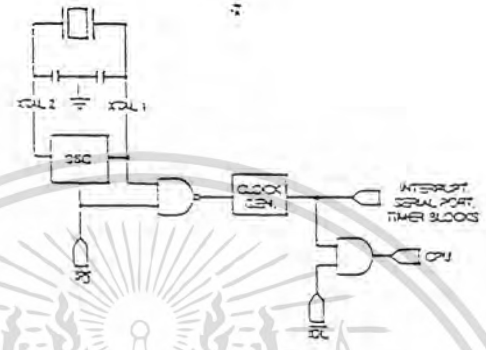
PD บิท 1 ถ้าบิทนี้ถูก Set ให้เป็น 1 89C52 จะเข้าสู่การทำงานใน Power Down ทันที

GF0, GF1 บิท 2,3 เป็น General Purpose Flag bit บิทนี้สามารถกำหนดให้มี ค่าเป็น 0 หรือ 1 ได้ โดยโปรแกรม เพื่อส่งผ่านสถานะการทำงานของ 89C52 ระหว่างแต่ละโปรแกรมย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMOD บิท 7 เป็นบิทที่ใช้ร่วมในการกำหนดส่งข้อมูล (Baud Rate) ผ่านทางพอร์ทอนุกรม ซึ่งในการรับ-ส่งข้อมูลผ่านทางพอร์ทอนุกรมโหมด 1 และ 3 จะสามารถกำหนดอัตราการส่งข้อมูลได้ตามอัตราการเกิด Overflow ใน Timer 1 ถ้าบิทนี้เป็น 1 จะทำให้อัตราการส่งข้อมูล เพิ่มขึ้น 2 เท่า รายละเอียดการส่งข้อมูลผ่านพอร์ทอนุกรมดูในข้อการรับส่งข้อมูลทางพอร์ทอนุกรม

บิท 4,5,6 ไม่ได้ใช้งาน



รูปที่ 3.35 Power down และ Idle mode

Idle Mode

ในรูปที่ 3.35 ขณะที่ 89C52 ทำงานตามปกติไปจนถึงคำสั่งที่ทำให้บิท 0 ของรีจิสเตอร์ PMOD มีค่าเป็น 1 ก็จะเข้าสู่การทำงานใน Idle Mode โดยสัญญาณ IDL จะเป็น LOW (สัญญาณจะตรงข้ามกับข้อมูลในบิท 0) ขณะนี้สัญญาณนาฬิกาออสซิลเลเตอร์ไม่ออกจาก AND GATE ไปยังส่วน CPU โดยจ่ายเฉพาะส่วน Interrupt , Timer และ Serial Port ในขณะนี้ 89C52 จะเสมือนหยุดการทำงานโดยข้อมูลใน Stack Pointer , Program Counter , Program Status Word , Accumulator และรีจิสเตอร์อื่น ๆ จะไม่เปลี่ยนแปลงข้อมูลพอร์ทต่าง ๆ จะยังคงค่าเดิมไว้เหมือนกับก่อนเข้าสู่ Idle mode และสัญญาณ ALE กับ PSEN จะเป็นลอจิก High ขณะนี้การใช้กระแสไฟของ 89C52 จะไม่มีการเปลี่ยนสถานะลอจิก การที่จะออกจาก Idle Mode ทำได้ 2 วิธี

วิธีที่ 1 โดยการขัดจังหวะจากสัญญาณขัดจังหวะทั้ง 6 ที่กล่าวมาแล้ว เมื่อมีสัญญาณขอขัดจังหวะจากแหล่งใดก็ตาม จะทำให้บิท 0 ของรีจิสเตอร์ PCON มีค่าเป็น 0 และการทำงานของ 89C52 จะออกจาก Idle Mode โดยกระโดดไปทำงานยังตำแหน่งของโปรแกรมตอบสนองการขัดจังหวะนั้น ๆ เมื่อเสร็จสิ้นการทำงานของโปรแกรมตอบสนองการขัดจังหวะโดยการทำงานคำสั่ง RETI ก็จะกลับมาทำงานยังคำสั่งที่อยู่ต่อจากคำสั่งที่ทำให้บิท 0 ของรีจิสเตอร์ PMOD เป็น 1 ซึ่งทำให้การทำงานเข้าสู่ Idle Mode เช่นคำสั่งที่ตำแหน่ง 200H คือ MOV PCON , #1H ที่เป็นคำสั่งที่ทำให้บิท IDC มีค่า 1 ดังนั้นเมื่อทำงานที่คำสั่งนี้เสร็จสิ้นก็จะหยุดการทำงาน และเมื่อเกิดการขัดจังหวะ เนื่องจากสัญญาณขัดจังหวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใด ๆ ก็ตาม 89C52 จะออกจาก Idle Mode ไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะเมื่อเสร็จสิ้นการทำงานโปรแกรมตอบสนองการขัดจังหวะแล้วจะกระโดดมาทำงานที่ตำแหน่งของคำสั่ง MOV PCON , #1H

วิธีที่ 2 ก็คือการป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปยังขา RST เพื่อทำการรีเซ็ท 89C52 สัญญาณรีเซ็ทนี้จะต้องมีลอจิกเป็น 1 ในระหว่างนี้ 89C52 จะทำงานในคำสั่งต่อจากคำสั่งที่ทำให้บิท 0 ของ ของ PCON เป็น 1 เข้าสู่ Idle Mode ต่อไปอีก 2-3 คำสั่ง ก่อนที่ทุกอย่างจะเข้าสู่การรีเซ็ท (ดูรายละเอียดการรีเซ็ทในหัวข้อ 3.4) ดังนั้นจะต้องระวังคำสั่งที่อยู่ต่อจากคำสั่งที่ทำให้เข้าสู่ Idle Mode อาจทำให้ข้อมูลบนพอร์ทเปลี่ยนแปลงจนทำให้อุปกรณ์ที่มาต่อเสียหายเมื่อกลับออกจาก Idle Mode

ในวิธีที่ 1 นั้นแสดงว่าการเข้าสู่โปรแกรมตอบสนองการขัดจังหวะจะเป็นได้ 2 กรณีคือขณะที่ทำงานตามปกติแล้วมีสัญญาณขัดจังหวะก็จะกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ หรือในกรณีที่อยู่ใน Idle Mode แล้วมีสัญญาณขัดจังหวะก็จะกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ จึงอาจทำให้โปรแกรมตอบสนองการขัดจังหวะนั้นมาจากกรณีใด

Power Down Mode

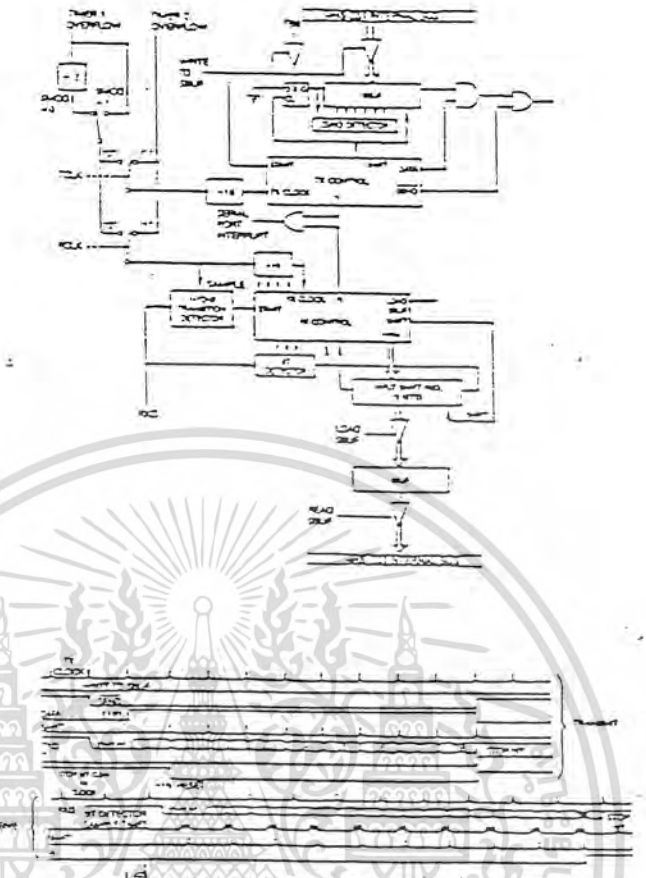
ในการเข้าสู่ Power down mode นั้นจะทำได้โดยการใช้โปรแกรมกำหนดให้บิท PD หรือบิท 1 ของรีจิสเตอร์ PCON มีค่าเป็น 1 เช่น MOV PCON , #2 เมื่อ 89C52 ทำงานที่คำสั่งนี้เสร็จสิ้นสัญญาณ PD ในรูปที่ 3.35 จะเป็น 0 เพราะจะตรงข้ามกับข้อมูลในบิท PD ทำให้การทำงานจะเข้าสู่ Power Down Mode ทันที ในโหมดนี้ออสซิลเลเตอร์จะหยุดการทำงานทำให้มีสัญญาณนาฬิกาไปยังส่วนต่าง ๆ ภายใน 89C52 ดังนั้นจะไม่มีการทำงานใด ๆ รวมทั้งข้อมูลในรีจิสเตอร์ทุกตัวจะไม่เปลี่ยนแปลง และข้อมูลใน แรมภายใน ก็จะไม่เปลี่ยนแปลง ขณะนี้สัญญาณออกจากขา ALE และ PSEN จะเป็น 0 การใช้กำลังไฟของ AT89C52 จะต่ำมากอีกทั้งสามารถลดไฟเลี้ยงวงจรที่ขา VCC ลงได้จนถึง 2 โวล์ทโดยไม่ทำให้ข้อมูลใด ๆ ใน AT89C52 สูญหายไป การออกจาก Power Down Mode ทำได้วิธีเดียวคือ การป้อนสัญญาณลอจิก 1 เข้าไปยังขา RST ซึ่งทำให้เข้าสู่สภาวะการรีเซ็ท AT89C52 แต่จะทำให้ข้อมูลใน SFR เปลี่ยนแปลงไป ถ้าในขณะที่อยู่ใน Power Down Mode มีการลดไฟเลี้ยงวงจรจะต้องให้ไฟเลี้ยงวงจรกับมาอยู่ที่ 5 โวล์ท ก่อนที่จะเข้าสู่การรีเซ็ท

3.8 การรับ-ส่งข้อมูลทางพอร์ทอนุกรม

ในการรับ-ส่งข้อมูลแบบอนุกรมผ่านทางพอร์ทอนุกรมนั้น จะต้องมีการกำหนดโหมดการทำงานในรีจิสเตอร์ SCON และในบางโหมดของการทำงานจะสามารถกำหนดอัตราการส่งข้อมูลได้โดยการโปรแกรมใน Timer ข้อมูลที่จะส่งออกหรือรับเข้าทางพอร์ทอนุกรมจะอยู่ที่รีจิสเตอร์ SBUF การทำงานของวงจรภายในแต่ละโหมดมีดังนี้ โดยแสดงการทำงานในโหมด 1 เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานโหมด 1



รูปที่ 3.36 Serial Port mode 1

จากรูปที่ 3.36 บิท SMOD จะเป็นตัวเลือกว่า สัญญาณ Timer 1 Overflow ที่ส่งไปยังวงจรหาร 16 จะถูกหาร 2 ก่อนหรือไม่ ถ้า SMOD เป็น 1 สัญญาณ Timer 1 จะไม่ถูกหาร แต่ถ้า SMOD เป็นสัญญาณ 1 Timer Overflow จะถูกหาร 2 ก่อนที่จะเข้าวงจรหาร 16 การส่งข้อมูลจะเริ่มจากการที่มีคำสั่งเขียนข้อมูลไปยังรีจิสเตอร์ SBUF จะมีสัญญาณ Write to SBUF เกิดขึ้นเพื่อรับข้อมูลจาก Internal Bus ด้านบนไปเก็บยังรีจิสเตอร์ SBUF และทำให้เอาต์พุตของ D FLIP FLOP ทางซ้ายของ SBUF มีค่าเป็น 1 และเป็นบิทที่ 9 ของการส่งข้อมูล สัญญาณ Write to SBUF ยังส่งไปยัง TX control ด้วย ขณะนี้ข้อมูลในวงจรหาร 16 มีค่าเป็นอะไรไม่ทราบจึงจะรอจนกว่าข้อมูลในวงจรหาร 16 นับเพิ่มขึ้นจนถึงค่าสูงสุดแล้ววนกลับมาเป็น 0 คือเกิดการวนกลับทำให้เริ่มการส่งข้อมูลที่เวลา S1P1 ของไซเคิลเครื่องถัดไป (การส่งข้อมูลออกจะสัมพันธ์กับการเกิด Overflow ในวงจรหาร 16) สัญญาณ SEND จาก TX Control เปลี่ยนสภาวะลอจิกเป็น 0 แล้วเริ่มส่งข้อมูลที่เป็น Start bit (0) ออกไป เมื่อส่ง Start Bit ออกไปแล้ววงจร Tx Control ก็จะทำให้สัญญาณ DATA เป็น 1 เพื่อเลื่อนข้อมูลใน SBUF ออกไป เริ่มจากบิท 0 จนถึงบิทที่ 7 การส่งข้อมูลนี้จะเกิดขึ้นเมื่อสัญญาณ Tx Clock เปลี่ยนสถานะจาก 0 เป็น 1 ดังในรูปที่ 3.36 ขณะที่ข้อมูลถูกเลื่อนออกไปนั้นจะมี 0 ถูกเลื่อนเข้ามาทางซ้ายของรีจิสเตอร์ เมื่อข้อมูลเลื่อนออกไปทั้ง 8 บิทแล้วบิทที่ 9 ซึ่งเป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และตอนต้นอยู่ทางซ้ายจะถูกเลื่อนมาอยู่ในตำแหน่งสุดท้ายทางขวาของรีจิสเตอร์ SBUF และทางซ้ายของหลักนี้จะมี 0 อยู่ทั้ง 8 บิทใน SBUF ทำให้ Zero Detector ส่งสัญญาณ Shift ออกไปเป็นการส่งข้อมูลบิทสุดท้าย (บิท 7) ออกไป ก็จะรออีก 1 TX Clock (Bit Clock) ก็จะทำให้ขา TXD ส่งข้อมูล Stop Bit (1) ออกมา สัญญาณ DATA ซึ่งมีสภาวะลอจิกเป็น 1 มาตั้งแต่เริ่มส่งข้อมูลบิท 0 ก็จะกลับเป็น 0 และบิท TI จะเป็น 1 เพื่อบอกการสิ้นสุดการส่งข้อมูลทั้งหมดจะสิ้นสุดเมื่อสัญญาณ TX Clock ไซเคิลที่ 10 นับตั้งแต่สัญญาณ SEND เปลี่ยนสภาวะลอจิกเป็น 0

การรับข้อมูล

การรับข้อมูลจะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 แล้วหาร 2 หรือไม่ขึ้นกับค่าของบิท SMOD สัญญาณนี้จะไปเข้าวงจรหาร 15 และเป็นตัวกำหนดอัตราการรับข้อมูล การรับข้อมูลจะเริ่มจากวงจร 1-TO-0 Transition Detector พบว่าสัญญาณที่ขา RXD เปลี่ยนจาก 1 เป็น 0 ซึ่งหมายถึงข้อมูล Start bit เข้ามา การตรวจสอบนี้จะกระทำด้วยอัตราเดียวกับสัญญาณที่เข้าวงจรหาร 16 เมื่อพบการเปลี่ยนสถานะลอจิกที่ขา RXD ก็จะเริ่มการับข้อมูล ขณะนี้จะรีเซ็ทวงจรหาร 16 ให้มีค่าเป็น 0 เพื่อสร้างสัญญาณ RX Clock ให้เข้าจังหวะ (Synchronous) กับข้อมูลที่เข้ามาโดยสัญญาณ RX Clock จะเป็น 1 เมื่อการนับของวงจรหาร 16 มีค่าเป็น 15 ขณะที่วงจรหาร 16 นับถึง 7 , 8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD เพื่อเป็นการตรวจข้อมูลนั้นเป็นอะไร ถ้าอย่างน้อยข้อมูล 2 ใน 3 เป็นค่าใดก็จะถือว่าข้อมูลที่เข้ามาเป็นค่านั้น ถ้าในการตรวจสอบ Start Bit แล้วพบว่าผิดพลาด คือไม่เป็น 0 ก็จะรีเซ็ทการทำงานเพื่อไปตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ของข้อมูลที่ขา RXD ใหม่ แต่ถ้าพบ Start Bit ก็จะเก็บข้อมูลทั้งหมดที่เข้ามาโดยเลื่อนข้อมูลเข้าไปยัง Input Shift Register ที่มีสัญญาณควบคุมการเลื่อนข้อมูล (Shift) ส่งมาจาก RX Control ในเริ่มต้นการรับข้อมูลจะมีการเขียนข้อมูล 1FFH ไปเก็บใน Input Shift Register ขณะที่ข้อมูลถูกเลื่อนเข้าไปทางขวาของ Input Shift Register ก็จะมี 1 ถูกเลื่อนออกไปทางซ้ายทุกครั้งที่มีข้อมูลเข้ามา เมื่อ Start Bit ที่รับเข้ามาถูกเลื่อนไปทางซ้ายสุดของ Input Shift Register ก็จะมีสัญญาณไปบอก RX Control Block หลังจากข้อมูลบิทสุดท้ายเข้ามาแล้วก็จะโหลด (Load) เอาข้อมูล 8 บิทไปเก็บในรีจิสเตอร์ SBUF พร้อมทั้ง Set ค่าในบิท RI และRB8 ของรีจิสเตอร์ SCON แต่การโหลดข้อมูลไปเก็บนี้จะเกิดขึ้นได้ก็ต่อเมื่อ

1. RI = 0 และ
2. SM2 = 0 หรือถ้า SM2 = จะต้องได้รับ stop bit เป็น 1

ถ้าไม่มีสภาวะใดสภาวะหนึ่งดังกล่าวแล้ว ข้อมูลที่รับเข้ามาจะถูกทิ้งไปคือ ไม่โหลดไปเก็บในรีจิสเตอร์ SBUF ถ้ามีสภาวะดังกล่าวถูกต้อง stop bit จะถูกนำไปเก็บในรีจิสเตอร์ SBUF และบิท RI จะเป็น 1

แต่ไม่ว่าทั้ง 2 กรณีจะเกิดหรือไม่ก็จะกลับไปสู่การตรวจสภาวะเปลี่ยนจาก 1 เป็น 0 ที่ขา RXD เพื่อรับข้อมูลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับข้อมูลแบบอนุกรมโหมด 1 นี้อัตราการส่งข้อมูลแต่ละบิทท (Band Rate) จะขึ้นกับอัตราการเกิด overflow ใน Timer 1 ดังสมการ

$$Baudrate = \frac{2^{SMOD}}{32} \times (Timer_1 OverflowRate)$$

ในขณะที่ใช้ Timer 1 เป็นตัวกำหนด Baud Rate นี้จะต้อง Disable ไม่ให้เกิดการขัดจังหวะเนื่องมาจากการ Overflow Timer 1 อาจใช้ในโหมดของ Timer หรือ Counter ก็ได้ ซึ่งเมื่อการนับในรีจิสเตอร์ตัวนับมีค่าสูงสุดแล้วกลับมาเป็น 0 ก็จะเกิด Overflow เช่นเดียวกัน แต่โดยปกติแล้วจะใช้ Timer 1 นี้ในโหมดของ Timer ที่มีการทำงานแบบ Auto Reload โหมด 2 เพื่อว่าเมื่อค่าในการนับโดย

รีจิสเตอร์ TL1 ถึงค่าสูงสุดก็จะโหลดค่าในรีจิสเตอร์ TH1 มาไว้ใน TL1 สำหรับเป็นค่าเริ่มต้นการนับต่อไป ซึ่ง Baud rate จะมีค่า

$$Baudrate = \frac{2^{SMOD}}{32} \times \frac{OscillatorFrequency}{12 \times [256 - (TH1)]}$$

โดยที่ SMOD เป็นบิทหนึ่งในรีจิสเตอร์ PCO

3.9 การขัดจังหวะ (Interrupt)

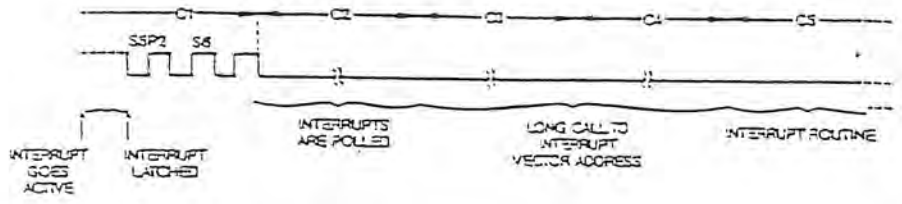
การขัดจังหวะคือสภาวะหนึ่งที่คอมพิวเตอร์กำลังทำงานอยู่แล้วถูกขัดจังหวะด้วยสัญญาณหรือคำสั่งพิเศษที่ทำให้คอมพิวเตอร์ต้องละจากงานที่กำลังทำอยู่ ไปทำงานในโปรแกรมตอบสนองการขัดจังหวะนั้น เมื่อเสร็จแล้วก็จะกลับมาทำงานเดิมต่อไปได้ ใน 89C52 จะสามารถขัดจังหวะกาทำงานได้ 6 แหล่งคือ

1. INT0 , INT1 เป็น 2 ขาของ 89C52 ที่จะรับสัญญาณจากภายนอก การขัดจังหวะจะเกิดขึ้น ถ้าสัญญาณที่ขาดังกล่าวมีสภาวะลอจิกเป็น 0 หรือเปลี่ยนจาก 1 เป็น 0 โดยเลือกด้วยการกำหนดในบิท IT0 หรือ IT1 ในรีจิสเตอร์ TCON

2. TF0 , TF1 เป็นบิทหนึ่งที่จะบอกการทำงานของ Timer 0 , Timer 1 เมื่อเกิด Overflow ขึ้นใน Timer จะทำให้บิทนี้เป็น 1 และเกิดการขัดจังหวะการทำงานของ 89C52 ได้

3. TI , RI เป็น 2 บิท ในรีจิสเตอร์ SCON ถ้าบิทนี้ถูกเซตให้เป็น 1 โดยฮาร์ดแวร์ อันเนื่องจากเสร็จสิ้นการส่งหรือรับข้อมูลจะสามารถทำให้เป็นการขัดจังหวะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or an access to IE or IP.

รูปที่ 3.37 ไดอะแกรมเวลาของการตอบสนองการขัดจังหวะ

89C52 จะทำการอ่านสัญญาณจากทั้ง 6 แหล่งที่เวลา S5P2 ของทุก ๆ ไซเคิลของเครื่อง (Machine Cycle) เข้ามาเก็บและในช่วงของไซเคิลของเครื่องถัดไปก็จะตรวจสอบสถานะของสัญญาณทั้ง 6 ที่เก็บเข้ามา ถ้าสัญญาณนั้นมีการขัดจังหวะที่ถูกต้อง 89C52 ก็จะละทิ้งการทำงานเดิมไว้ชั่วคราวแล้วสร้างคำสั่ง LCALL ขึ้นมาภายใน 89C52 เพื่อไปทำงานในโปรแกรมตอบสนองการขัดจังหวะแต่ละสัญญาณนั้น เมื่อทำงานในโปรแกรมตอบสนองการขัดจังหวะเสร็จสิ้นก็จะสามารถกลับมาทำงานเดิมได้ โดยคำสั่ง RET1 เป็นคำสั่งสุดท้ายในโปรแกรมตอบสนองการขัดจังหวะ สัญญาณขัดจังหวะจากแต่ละแหล่งจะมีตำแหน่งหน่วยความจำที่จะเก็บโปรแกรมตอบสนองการขัดจังหวะไว้ต่างกันดังนี้

สัญญาณที่ขอขัดจังหวะ		ตำแหน่งเริ่มต้นโปรแกรมตอบสนองการขัดจังหวะ
1	INT0	0003H
2	TFO	000BH
3	INT1	0013H
4	TF1	001BH
5	TI,RI	0023H

ตำแหน่งเริ่มต้นโปรแกรมนี้เป็นตำแหน่งใน Program area เช่น ถ้ามีสัญญาณของ INT0 เข้ามาแล้ว 89C52 ตรวจสอบว่ามีการขอขัดจังหวะถูกต้อง ก็จะละทิ้งการทำงานเดิม แล้วไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะที่มีตำแหน่งเริ่มต้นอยู่ที่ตำแหน่ง 0003H เมื่อเสร็จสิ้นการทำงานของโปรแกรมตอบสนองการขัดจังหวะจะต้องมีคำสั่ง RETI อยู่เพื่อกลับมาสู่การทำงานเดิมได้ 89C52 จะทำการตรวจสอบสัญญาณดังกล่าวว่ามีสัญญาณใดขอการขัดจังหวะมาบ้างได้โดยวิธี Polling คือการตรวจสอบเรียงตามลำดับจาก 1 , 2 , 3 , 4 และ 5 ตามลำดับ ดังนั้นถ้ามีการขอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขัดจังหวะเข้ามาพร้อม ๆ กัน AT89C52 ซึ่งตรวจสอบการขอขัดจังหวะแบบ Polling จะพบว่าสัญญาณมีการขอขัดจังหวะจากสัญญาณต้น ๆ ก่อนจึงตอบสนองต่อการขอขัดจังหวะของสัญญาณต้น ๆ ก่อนหรืออีกนัยหนึ่งก็คือ สัญญาณขอการขัดจังหวะต้น ๆ จะมีลำดับความสำคัญสูงสุด (Highest Priority) และในสัญญาณที่ 5 จะมีลำดับความสำคัญต่ำที่สุด (Lowest Priority) อย่างไรก็ตามสามารถที่จะจัดลำดับความสำคัญของสัญญาณขัดจังหวะนี้ใหม่ เพื่อให้มีการตอบสนองการขัดจังหวะสัญญาณขอการขัดจังหวะลำดับหลังได้

โดยการโปรแกรมในรีจิสเตอร์ IP (Interrupt Priority Register) และสามารถกำหนดว่าจะให้ทำโปรแกรมตอบสนองการขัดจังหวะเมื่อมีสัญญาณขอขัดจังหวะเข้ามาหรือไม่ก็ได้ โดยการโปรแกรมในรีจิสเตอร์ IE (Interrupt Enable Register)

เมื่อ 89C52 ทำการตรวจสัญญาณขอการขัดจังหวะที่เก็บเข้ามาเมื่อเวลา S5P2 แล้วพบว่ามีการขอขัดจังหวะนั้น แม้ว่ามีการ Enable ในรีจิสเตอร์ IE ถูกต้อง แต่จะต้องมีเงื่อนไขดังนี้ด้วย

1. ไม่ได้กำลังทำงานในโปรแกรมตอบสนองการขัดจังหวะของสัญญาณ ที่มีลำดับความสำคัญสูงกว่าหรือเท่ากัน เช่น กำลังทำงานในโปรแกรมตอบสนองการขัดจังหวะของสัญญาณ INT0 อยู่ แล้วมีการขอขัดจังหวะจากสัญญาณ INT1 อีก จะไม่เกิดการทิ้งงานเดิม คือไม่มีการไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะของสัญญาณ INT1

2. เนื่องจากการส่งสัญญาณเข้าไปเพื่อตรวจสอบนั้นจะทำให้เวลา S5P5 ของในไซเคิลสุดท้ายของคำสั่ง และคำสั่งที่อยู่ถัดมาจะต้องในเวลาทำงาน 2 ไซเคิลของเครื่อง ดังนั้นการตรวจสอบจะกระทำในไซเคิลแรก แม้ว่าจะมีการขอการขัดจังหวะเข้ามา ก็จะไม่ทำโปรแกรมตอบสนองการขัดจังหวะ จะต้องอ่านสัญญาณที่เวลา S5P2 อีกครั้งแล้วไปตรวจสอบที่ไซเคิลที่ 2 ของคำสั่ง ถ้ามีการขอขัดจังหวะถูกต้องจึงจะข้ามไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ

3. คำสั่งที่กำลังทำงานอยู่ขณะที่ตรวจสอบสัญญาณการขอขัดจังหวะ จะต้องไม่ใช่คำสั่ง RET หรือคำสั่งใด ๆ ก็ตามที่พยายามเขียนข้อมูลไปยังรีจิสเตอร์ IE หรือ IP

สัญญาณขอขัดจังหวะที่ถูกอ่านเข้าไปที่เวลา S5P2 นี้ไม่ว่าได้รับการตอบสนองหรือไม่จะถูกทิ้งไป แล้วอ่านเข้าไปใหม่ทุกเวลา S5P2

3.10 เมโมรีแม็พไอโอเทคนิค

เป็นวิธีการเพิ่มหมายเลขพอร์ตอินพุตเอาต์พุต โดยการกำหนดหมายเลขพอร์ตให้แทนที่ที่ตำแหน่งของหน่วยความจำ ตำแหน่งใดตำแหน่งหนึ่ง ซึ่งเป็นหน่วยความจำประเภทหน่วยความจำภายนอก ซึ่ง ณ. ตำแหน่งนั้นไม่มีหน่วยความจำจริงต่ออยู่ เพียงแต่ใช้หมายเลขของหน่วยความจำเป็นตัวอ้างอิงในการติดต่อ ในกรณีของ ชุดจำลอง พีแอลซี จะต้องใช้หมายเลขพอร์ตซึ่งได้จากวิธีการดังกล่าวนี้เป็นหมายเลขที่แน่นอน (แอ็คทีฟที่หมายเลขเพียงหมายเลขเดียว เพื่อป้องกันความผิดพลาดที่จะเกิดขึ้นกับอุปกรณ์ภายนอก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4

ส่วนประกอบของโครงงาน

ส่วนประกอบแบ่งออกเป็น 2 ส่วนใหญ่ๆคือ

1.Hardware แสดงดังรูปในภาคผนวก แบ่งออกเป็นส่วนต่างๆดังนี้

1.1 Controler เป็นส่วนที่สำคัญมากในโครงงานนี้โดยใช้ Microtroler ตระกูล MCS51 เบอร์ที่ใช้เป็นเบอร์ AT89C52 ทฤษฎีและการทำงานได้อธิบายไว้แล้วในบทที่ 2 AT89C52 จะเป็นตัวประมวลผลต่างๆที่เราต้องการให้ทำงานตามที่เราสั่งเข้าไปทาง Keybord แล้วแสดงผลออกทาง อุปกรณ์ Out put เช่น จอแสดงผล (LCD) หรืออุปกรณ์ที่เราต้องการควบคุม

1.1 Memory จำเป็นอย่างยิ่งที่ Controler จะต้องใช้ควบคู่กันไปในการประมวลผลเพื่อใช้เป็นที่เก็บผลลัพธ์ต่างๆที่ประมวลผลได้ ในโครงงานนี้ใช้ IC 6264 เป็น Memory ขนาด 32 Kb

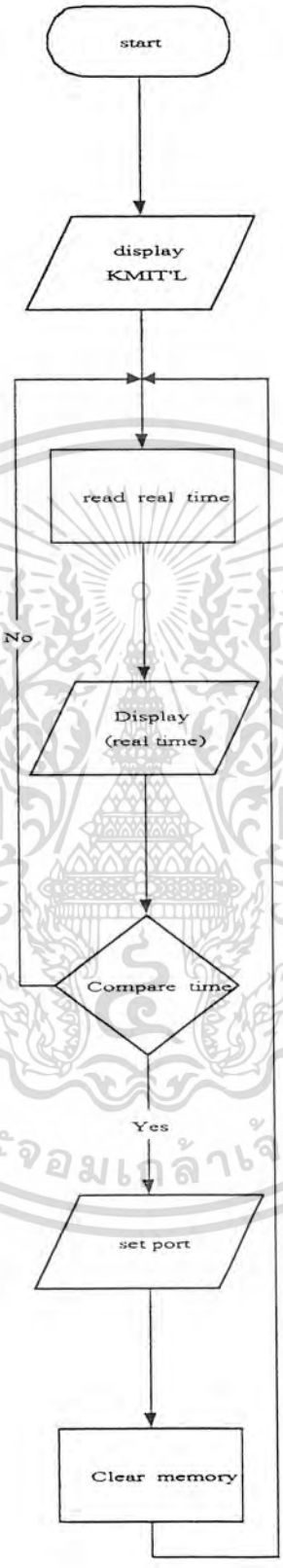
1.2 Kebord เป็นอุปกรณ์ In put ซึ่งเราสามารถป้อนคำสั่งต่างๆเพื่อกำหนดการควบคุมการทำงานของเครื่อง

1.3 จอแสดงผล (LCD Module) ใช้เบอร์ SC1601KU เป็น LCD 1 แถว 16 ตัวอักษรใช้ในการแสดงผลของการป้อนคำสั่งต่างๆ เช่น เวลา ปิด-เปิด หรือ ช่องการควบคุม

1.4 Driver เป็นวงจรที่ใช้ขับเครื่องใช้ไฟฟ้า โดยใช้ Transister BC547 ร่วมกับ Relay 12V 1 Contrac

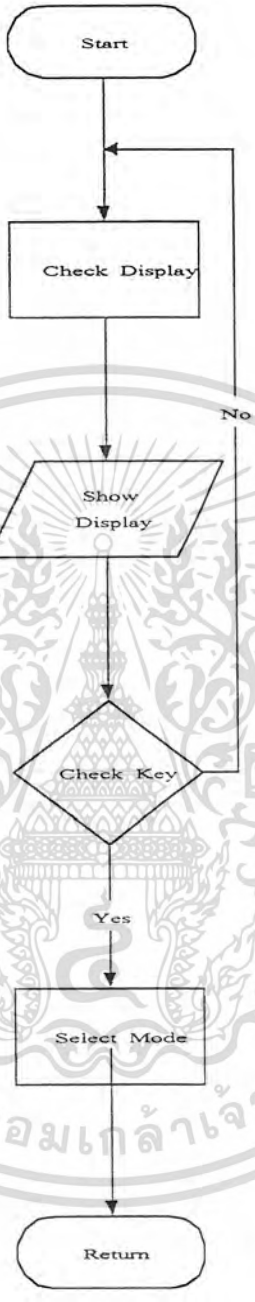
2. Software เขียนด้วยภาษา แอสแซมบลี เป็นคำสั่งต่างที่ใช้ควบมคุม Controler เพื่อให้ทำงานตามที่เราสั่ง Software จะทำงานร่วมกับ Hardware กล่าวคือในการป้อนคำสั่งต่างๆทาง Keybord หรือแสดงผลออกทาง Display หรือควบคุมการปิด-เปิดอุปกรณ์ภายนอก ก็จะมี Software นี้เป็นคำสั่งผ่านอุปกรณ์ให้ทำงานตามแต่ละขั้นตอนให้เป็นไปอย่างเป็นระเบียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 การทำงานโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แผนผังการทำงานของ Keyboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แผนผังการทำงานการ set port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณลักษณะของเครื่อง

เครื่องควบคุมการ ปิด-เปิด ไฟฟ้าอัตโนมัติมีคุณลักษณะดังนี้

1.ใช้กับไฟฟ้า 220 V

2.CPU ตระกูล MCS51 เบอร์ AT89C52

3.สามารถควบคุมอุปกรณ์ได้ 4 ตัว

4.ในการควบคุมสามารถควบคุมจาก ปี-เดือน-วัน-ชั่วโมง-นาที-วินาที

5.สามารถควบคุมอุปกรณ์ที่ใช้ไฟฟ้า 220V ได้ทุกชนิด

6.จอแสดงผลเป็น LCD และมี LED แสดงสถาณะของแต่ละช่องว่า ปิด หรือ เปิด อยู่

หลักการพื้นฐานของระบบ

จุดประสงค์ของเครื่องควบคุมนี้ก็คือการใช้ Keybord สั่งงานเพื่อควบคุมการ ปิด-เปิด ของอุปกรณ์ไฟฟ้า โดยเริ่มแรก Controler จะตรวจสอบว่ามีการกด Keybord หรือไม่ ถ้าหากมีการกด Keyใดๆก็ตาม Controler จะรับรู้ว่ามีการต้องการที่จะใช้ Controler เพื่อควบคุมการปิด-เปิด อุปกรณ์ไฟฟ้า Controler ก็จะรอรับคำสั่งต่างๆตามวิธีการใช้งาน ขณะที่ Controler ทำงานก็จะแสดงข้อความต่างๆที่เราได้สั่งออกทาง LCD
การที่จะควบการปิด-เปิด ทำได้โดยการกด Keybord เลือกช่องของอุปกรณ์แล้วตั้งเวลาที่ต้องการจะปิดหรือเปิด

วงจรและการทำงาน

การทำงานและรายละเอียดของวงจรที่สมบูรณ์ของเครื่องควบคุมแสดงดังรูปในภาคผนวค จากวงจรจะสังเกตุว่าพอที่จะแยกได้อยู่ 4 ส่วน

ส่วนที่ 1 เป็นอุปกรณ์ทางด้าน Input คือ Keybord ขณะที่เรากด Keybord IC MC145412 จะเป็นตัวสร้างสัญญาน DTMF โดยแต่ละปุ่มจะให้ความถี่ของสัญญาน DTMF ที่แตกต่างกัน สัญญาน DTMF ที่ได้นี้จะถูกถอดรหัสให้เป็น BCD Code โดย IC MT8870 BCD Code จะถูกส่งไปเข้าระบบ Microcontroler ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 2 ระบบ Micro controler เป็นหัวใจของระบบโดยใช้ Controler ตระกูล MCS51 นั่นก็คือ IC AT89C52 โดยมี IC 6264 เป็น External RAM และ Program บรรจุอยู่ภายใน ใการส่งคำสั่งเพื่อควบคุมอุปกรณ์ภายนอก Controler จะอาศัย IC 8255 เป็นทางผ่านในการควบคุม ว่าจะให้อุปกรณ์ตัวไหน ปิด-เปิดที่วเลาใดๆตามที่เรา Program เอาไว้ หรือแม้แต่การแสดงข้อความบน LCD

ส่วนที่ 3 เป็นส่วนของ Out put LCD เป็นตัวแสดงผล ขนาด 1 แถว 16 ตัวอักษร และชุด Driver เพื่อเป็นตัวขับอุปกรณ์ภายนอกให้ทำงาน ใช้ Transister เบอร์ BC547 กับ Relay 12 V 1 contrac ทำงานร่วมกัน คือ ถ้าหากมีสัญญานมา Trig ที่ขา B ของTransister ซึ่งจะเป็นสัญญานที่ได้มาจาก Controler ผ่านทาง 8255 ก็จะทำให้ Transister ทำหน้าที่เป็น Switch มีกระแสไหลผ่าน Relay ซึ่งปกติจะอยู่ในสภาวะ NO เมื่อมีกระแสไหลผ่าน coil ของ Relay ก็จะอยู่ในสภาวะ NC ตัดต่อให้มีกระแสไฟฟ้า 220 V ไหลผ่านอุปกรณ์ไฟฟ้า เป็นการเปิดให้ทำงาน และถ้าหากมีการ Teig อีกครั้ง Relay ก็จะอยู่ในสภาวะ NO เป็นการปิดไฟฟ้า

ส่วนที่ 4 เป็นชุด Real time ใช้ IC DS1202 เป็นฐานเวลาให้กับระบบ โดยมี Battery Back up อยู่ ทำเครื่องรับรู้อยู่ตลอดเวลาว่าขณะนี้เป็นเวลาเท่าใด แม้จะไม่มีไฟเลี้ยงวงจร

วิธีการใช้งาน

1. เมื่อติดตั้งเครื่องเรียบร้อยแล้ว On power switch ที่หน้าจอจะแสดงข้อความว่า KMIT'L 1999 ประมาน 2 วินาทีแล้วจะปรากฏเวลาปัจจุบัน เรียงจาก ปี เดือน วัน ชั่วโมง นาที วินาที เช่น 99 04 01 12 1500 หมายถึง ปี ค.ศ 1999 เดือน เมษายน วันที่ 29 เวลา 1:12:15:00 นาฬิกา

1.1 ถ้าหากเวลาไม่ตรงกับเวลาปัจจุบันให้ตั้งเวลาใหม่ ดังนี้

1.1a กด Key ใดๆก็ได้

1.1b ที่หน้าจอจะมีข้อความว่า "Insert year" ให้ใส่ปีโดยป้อนเลข 2 ตัวท้าย

"Insert month" ป้อนเดือนด้วยเลข 2 หลัก

"Insert day" ป้อนวันด้วยเลข 2 หลัก

"Insert hour" ป้อนชั่วดมงด้วยเลข 2 หลัก

"Insert min" ป้อนนาทีด้วยเลข 2 หลัก

"Insert sec" ป้อนวินาทีด้วยเลข 2 หลัก

"Insert channal ป้อนอะไรก็ได้

"Insert on,off" ป้อนอะไรก็

เมื่อป้อน เสร็จที่หน้าจอจะแสดงข้อความว่า "Insert mode"ให้ป้อน mode ต่างๆ ตามที่ต้องการซึ่ง มี อยู่ 3 mode คือ

1. *# คือ mode set real time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ** คือ mode cancel

3. ## คือ mode set เวลา ปิด-เปิด

จากข้อ 1.1 เราต้องการ set real time ให้กด *# ที่หน้าจอจะแสดงเวลาที่เรา set เอาไว้และรอให้เราป้อนคำสั่งต่อไป

2. ถ้าหากเวลาตรงกับเวลาปัจจุบันแล้วสามารถตั้งเวลาควบคุมได้เลยโดย

2.1 กด key ใดๆก็ได้

2.2 ทำตามขั้นตอน 1.1b แต่พอถึงตอนใส่ค่า Insert channal ให้ใส่ช่องที่ต้องการควบคุม โดยจะมี 4 ช่องให้เลือกคือ 01,02,03,04 ส่วนการ insert on,off คือการสั่ง ปิด-หรือ เปิด โดย 00 จะเป็นการปิด 01 จะเป็นการเปิด

2.3 สำหรับ insert mode ให้กด ## ที่หน้าจอจะปรากฏข้อความ set time complete ประมาณ 1 วินาที แล้วจะแสดงเวลาปัจจุบัน

3. ถ้าต้องการปิดหรือเปิดโดยไม่ต้องรอเวลาที่ตั้งเอาไว้สามารถกดปุ่ม invert ได้เลย ที่หน้าจอจะปรากฏขอ้ความ insert channal ให้ใส่ช่องที่ต้องการจะควบคุม โดยใส่เลข 1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5

สรุปและวิจารณ์

จากการทำ Project นี้สามารถใช้เครื่องควบคุมการปิด เปิดไฟฟ้าอัตโนมัติได้ตามวัตถุประสงค์ที่กำหนดเอาไว้คือ ในการควบคุมอุปกรไฟฟ้าสามารถตั้งเวลาควบคุมการปิด เปิด ได้ตามเวลาที่เรากำหนดและสามารถควบคุมอุปกรณ์ไฟฟ้าได้ถึง 4 channal ในการตั้งเวลาสามารถทำได้โดยการกด key bord ตั้งเวลาได้จาก ปี เดือน วัน ชั่วโมง นาที วินาที และยังสามารถตรวจสอบได้ด้วยว่าขณะนี้อุปกรณ์อยู่ในสภาวะปิด หรือ เปิด อยู่ และถ้าหากต้องการกลับสภาวะก็สามารถทำได้ทันที

ปัญหาที่พบ

ปัญหาที่พบมีมากมาย ตั้งแต่การเริ่มทำ hardware เราไม่สามารถหาอุปกรณ์ที่มีคุณสมบัติตามที่ต้องการดังนั้นเราจึงดัดแปลงและหาอุปกรณ์ที่สามารถใช้งานได้ใกล้เคียงกับอุปกรณ์ต้นแบบ เมื่ออุปกรณ์ตัวหนึ่งเปลี่ยนไป อุปกรณ์ตัวอื่นก็ต้องเปลี่ยนตามไปด้วย หรือต้องมีการเพิ่มเติมเข้ามาอีกประการหนึ่ง project นี้ใช้หลายๆวงจรรวมกันหากวงจรใดวงจรหนึ่งมีการเปลี่ยนแปลง วงจรอื่นๆก็ต้องปรับเปลี่ยนตามไปด้วยจึงเกิดความล่าช้าไม่เป็นไปตามขั้นตอนที่วางแผนเอาไว้

ปัญหาอีกอย่างที่เราพบก็คือ ขณะที่มีการเปลี่ยนแปลงทางด้าน hardware ด้าน software ก็ต้องมีการแก้ไขตามไปด้วยดังนั้นเราจึงไม่สามารถเขียน program ตาม โฟลชาร์จ ที่เราเขียนเอาไว้ตั้งแต่ต้นได้

หลังจากที่เราสามารถทำเครื่องควบคุมขึ้นมาแล้วจากการทดลอง แต่พอถึงขั้นตอนการใช้งานจริงก็เจอปัญหาอีก เช่น ขณะที่ทดลองเราใช้ LED แทนอุปกรณ์ไฟฟ้าซึ่งสามารถควบคุมได้ดีมากไม่มีปัญหาแต่พอเอาอุปกรณ์ไฟฟ้าจริงๆมาทดลองกลับมีปัญหาตามมาอีกคือ หากมีการควบคุมมากกว่า 1 channal จะเกดการรวนของ relay ซึ่งเป็นอุปกรณ์ในการขับอุปกรณ์ไฟฟ้าสาเหตุเกิดจากการกระเพื่อมของสัญญาณขณะที่เราสั่งให้มีการปิด เปิด เป็นผลให้ relay ตัวข้างๆได้รับผลกระทบไปด้วย การแก้ไข ได้ใช้ capacitor ต่อคล่อม switchเพื่อลดการกระเพื่อมของสัญญาณ

ข้อเสนอแนะ

project นี้สามารถที่จะพัฒนาได้อีกโดยการใช้ โทรศัพท์สั่งงานในการควบคุม เพียงแต่เพิ่มอุปกรณ์ตรวจสอบสัญญาณกระดิ่งของโทรศัพท์เข้าไปแล้วใช้สัญญาณ DTMF ที่ได้จากการกด key ของโทรศัพท์ไปควบคุมแทนการกด key bord แต่ก็ต้องเขียน program เพิ่มเติมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งของโครงงาน และการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;***************************
;* Program Electric contron*
;***************************
;
          newkey  equ    45h
          newflg  bit    f0
          YEAR          EQU   30H
          MONTH          EQU   31H
          DATE          EQU   32H
          HR           EQU   33H
          MIN           EQU   34H
          SEC           EQU   35H
          TEMP          EQU   49H
          RTC_DATA       EQU   P1.0
          RTC_CLK        EQU   P1.2
          RTC_RST        EQU   P1.1
PORTA     EQU   8000H          ;LCD DATA (PA0-PA7)
PORTB     EQU   8001H          ;LCD CONTROL (PB0-PB2)
                    ;PB0 -> RS
                    ;PB1 -> R/W
                    ;PB2 -> E
PORTC     EQU   8002H
PORTP     EQU   8003H          ;8255 CONTROL PORT
LCDLINE1 EQU    80h
LCDLINE2 EQU    0c0h


        org 0000h
       jmp main


        org 0003h
        jmp interrupt_key

        org 0013h
        jmp set_channal

;*************************
;*  CONTROC KEY      *
;*************************
;

interrupt_key:
        lcall copy1
        push acc
        push dpl
        push dph
        push psw
        lcall lcd_data
        mov r2,#5
        lcall delaylcd
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov r0,#50h
        mov r5,#12h
        mov r6,#2
        MOV R7,#12H
keyint:  NOP
check_lcd:    cjne r7,#18,lcd1
              lcall lcd_year
    lcd1:     cjne r7,#16,lcd2
              lcall lcd_month
    lcd2:     cjne r7,#14,lcd3
              lcall lcd_date
    lcd3:     cjne r7,#12,lcd4
              lcall lcd_hour
    lcd4:     cjne r7,#10,lcd5
              lcall lcd_min
    lcd5:     cjne r7,#8,lcd6
              lcall lcd_sec
    lcd6:     cjne r7,#6,lcd7
              lcall lcd_channal
    lcd7:     cjne r7,#4,lcd8
              lcall lcd_on_off
    lcd8:     cjne r7,#2,lcd11
              lcall lcd_mode
    lcd11:    cjne r7,#17,lcd22
              lcall lcd_year1
    lcd22:    cjne r7,#15,lcd33
              lcall lcd_month1
    lcd33:    cjne r7,#13,lcd44
              lcall lcd_date1
    lcd44:    cjne r7,#11,lcd55
              lcall lcd_hour1
    lcd55:    cjne r7,#9,lcd66
              lcall lcd_min1
    lcd66:    cjne r7,#7,lcd77
              lcall lcd_sec1
    lcd77:    cjne r7,#5,lcd88
              lcall lcd_channal1
    lcd88:    cjne r7,#3,lcd99
              lcall lcd_on_off1
    lcd99:     cjne r7,#1,lcd10
              lcall lcd_mode1
    lcd10:    jmp lcd_ent
              ;ret


        ;lcall check_lcd
lcd_ent:
        DEC R7
tim0:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov a,p1
        anl a,#00001000b
        cjne a,#00H,good
        jmp tim0
good2:
        mov a,#00h
        jmp good1
good:   mov a,p1
        anl a,#0f0h
        cjne a,#0a0h,good1
        jmp good2
good1:  mov newkey,a
        setb newflg
tim1:
        mov a,p1
        anl a,#00001000b
        CJNE a,#00H,tim1
        ;JMP GI2
simulate:
        Jbc newflg,add_mem
        Jmp keyint
add_mem:
        djnz r6,add1
        jmp add2

 add1:
        lcall show_3
        mov a,newkey
        lcall show_1
        MOV TEMP,A
        lcall lcdwd
        mov a,newkey
        swap a
        ANL A,#0FH
        swap a
        mov @r0,a
        inc r0
        MOV A,R5
        DEC A
        JZ MODE
        MOV R5,A
        JMP KEYINT
        ;djnz r3,keyint
        ;jmp mode
 ADD2:
        lcall show_4
        mov a,newkey
        lcall show_1
        lcall lcdwd
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov r2,#5
        lcall delaylcd
        mov a,newkey
        swap a
        ANL A,#0FH
        dec r0
        add a,@r0
        mov @r0,a
        inc r0
        mov r6,#02h
        MOV A,R5
        DEC A
        JZ MODE
        MOV R5,A
        JMP KEYINT
        ;djnz r5,keyint
        ;jmp mode


;**********************
;*     mode           *
;**********************
 mode:
      mov a,58h
      cjne a,#0bbh,mode1
      lcall lcd_cancel
      jmp gi3
mode1:
      cjne a,#0bch,mode2
      lcall lcd_real_time
      ;lcall lcd_completed
      lcall  set_time_ds1202
      jmp gi2
mode2:
      cjne a,#0cch,mode3
      lcall lcd_set_time
      jmp give_mem
mode3:
      jmp gi2
;********************
;    GIVE  MEM     *
;********************
give_mem:mov r7,#08
        mov r1,#50h
        mov b,#08
        mov a,20h
        mul ab
        mov 46h,a
        mov b,#08
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov a,21h
        mul ab
        mov 47h,a
        mov a,b
        add a,46h
        mov 46h,a
        mov dpl,47h
        mov dph,46h
 gi:    mov a,@r1
        movx @dptr,a
        inc dptr
        inc r1
        djnz r7,gi
        mov a,21h
        addc a,#01
        mov 21h,a
        jc gi1
        jmp gi2
 gi1:   inc 20h
        clr c
 gi2:
        lcall lcd_completed
 gi3:
        lcall copy2
        pop psw
        pop dph
        pop dpl
        pop acc
        clr ie0
        reti


;**********************
;*   SET  CHANNAL     *
;**********************
set_channal:
        CLR EA
        lcall copy1
        push acc
        push dpl
        push dph
        push psw
        lcall lcd_invert_ch
tim00:  clr ex0
        mov a,p1
        anl a,#00001000b
        cjne a,#00H,goodd
        jmp tim00
good22:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov a,#00h
        jmp good11
goodd:   mov a,p1
        anl a,#0f0h
        cjne a,#0a0h,good11
        jmp good22
good11:  mov newkey,a
        setb newflg
tim11:
        mov a,p1
        anl a,#00001000b
        CJNE a,#00H,tim11
keyintt:
      mov a,newkey
      lcall show_4
      mov a,newkey
      lcall show_1
      MOV TEMP,A
      lcall lcdwd
      mov r2,#5
      lcall delaylcd
      mov a,temp
      cjne a,#31h,p11_off
      call por_con
      mov a,#01h
      movx @dptr,a
      mov r2,#02h
      lcall delaylcd
      mov a,#0
      lcall lcdwd1
      mov r2,#5
      lcall delaylcd
      jmp clear_mem1
  p11_off:
      cjne a,#31h,p22_on
      call por_con
      mov a,#01h
      movx @dptr,a
      mov r2,#02h
      lcall delaylcd
      mov a,#0
      lcall lcdwd1
      jmp clear_mem1
  p22_on:
      cjne a,#32h,p22_off
      call por_con
      mov a,#02h
      movx @dptr,a
      mov r2,#02h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    jmp clear_mem1
p22_off:
    cjne a,#32h,p33_on
    call por_con
    mov a,#02h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    jmp clear_mem1
p33_on:
    cjne a,#33h,p33_off
    call por_con
    mov a,#04h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    jmp clear_mem1
p33_off:
    cjne a,#33h,p44_on
    call por_con
    mov a,#04h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    jmp clear_mem1
p44_on:
    cjne a,#34h,p44_off
    call por_con
    mov a,#08h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    jmp clear_mem1
p44_off:
    cjne a,#34h,p55_on
    call por_con
    mov a,#08h
    movx @dptr,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      mov r2,#02h
      lcall delaylcd
      mov a,#0
      lcall lcdwd1
      jmp clear_mem1
  p55_on: nop
clear_mem1:
       lcall lcd_completed
       mov r2,#5
       lcall delaylcd
       lcall copy2
       clr ie1
       clr ie0
       pop psw
       pop dph
       pop dpl
       pop acc
       reti

;*********************
;*       MAIN        *
;*********************
 main:
      LCALL lcd_kmitl
 main2:
      mov r2,#00h
      mov r3,#00h
      mov 24h,#00h
      mov 23h,#00h
      mov 20h,#00h
      mov 25h,#00h
      mov 21h,#00h
      mov 22h,#00h
      mov 26h,#00h
      mov 27h,#00h
      mov 3ah,#00h
      mov 3bh,#00h
      mov 40h,#00h
      mov 41h,#00h
      mov 70h,#00h
      mov 71h,#00h
 main1:
      clr ea
      call read_time
      LCALL LCDMAIN
      LCALL LCD_DIS_MEM
      setb ea
      setb ex0
      setb ex1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov tcon,#05h
        mov ip,#04h
        mov dptr,#0000h
        mov r0,#30h
        mov 3dh,21h
        mov 3ch,20h
        mov a,21h
        add a,20h
        jz main1
        ;JMP MAIN1
loop_next:
        ;clr ea
        ;call read_time
        ;LCALL LCDMAIN
        ;LCALL LCD_DIS_MEM
        ;setb ea
        ;setb ex0
        mov r1,#30h
        mov r4,#5
        mov a,41h
        add a,#01
        mov 41h,a
        jc nnext1
        jmp loop
nnext1:inc 40h
        clr c

loop:
        movx a,@dptr
        mov 22h,a
        mov a,@r1
        cjne a,22h,next
        inc r1
        inc dptr
        djnz r4,loop
        mov r1,#30h
        jmp check
next:
          mov b,#08
          mov a,40h
          mul ab
          mov 72h,a
          mov b,#08
          mov a,41h
          mul ab
          mov 73h,a
          mov a,b
          add a,72h
          mov 72h,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov dpl,73h
        mov dph,72h
        mov a,41h
        cjne a,21h,loop_next
        mov a,40h
        cjne a,20h,loop_next
        mov 40h,#00h
        mov 41h,#00h
       jmp main1
;************************
;*  Check port contron  *
;************************
Check: inc dptr
      mov b,#02h
      movx a,@dptr
      mul ab
      mov 27h,a
      inc dptr


      mov a,dpl
      mov 75h,a
      mov a,dph
      mov 74h,a


      movx a,@dptr
      add a,27h
port_contron:
      cjne a,#03h,p1_off
      call por_con
      mov a,#01h
      movx @dptr,a
      mov r2,#02h
      lcall delaylcd
      mov a,#0
      lcall lcdwd1
      mov r2,#5
      lcall delaylcd
      ;lcall por_con1
      jmp clear_mem
 p1_off:
      cjne a,#02h,p2_on
      call por_con
      mov a,#01h
      movx @dptr,a
      mov r2,#02h
      lcall delaylcd
      mov a,#0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p2_on:
    cjne a,#05h,p2_off
    call por_con
    mov a,#02h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p2_off:
    cjne a,#04h,p3_on
    call por_con
    mov a,#02h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p3_on:
    cjne a,#07h,p3_off
    call por_con
    mov a,#04h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p3_off:
    cjne a,#06h,p4_on
    call por_con
    mov a,#04h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p4_on:
    cjne a,#09h,p4_off
```

เอกสารนี้เป็นการเผยแพร่เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    call por_con
    mov a,#08h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p4_off:
    cjne a,#08h,p5_on
    call por_con
    mov a,#08h
    movx @dptr,a
    mov r2,#02h
    lcall delaylcd
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem

p5_on:
    cjne a,#0bh,p5_off
    call por_con
    setb acc.4
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p5_off:
    cjne a,#0ah,p6_on
    call por_con
    clr acc.4
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p6_on:
    cjne a,#0dh,p6_off
    call por_con
    setb acc.5
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p6_off:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    cjne a,#0ch,p7_on
    call por_con
    clr acc.5
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p7_on:
    cjne a,#0fh,p7_off
    call por_con
    setb acc.6
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p7_off:
    cjne a,#0eh,p8_on
    call por_con
    clr acc.6
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p8_on:
    cjne a,#11h,p8_off
    call por_con
    setb acc.7
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
p8_off:
    cjne a,#10h,go
    call por_con
    clr acc.7
    movx @dptr,a
    mov a,#0
    lcall lcdwd1
    ;lcall por_con1
    jmp clear_mem
 tt1:
    dec 3ch
    clr c
    jmp tt2
go:   jmp clear_mem
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************
; 8255 contron portc output*
;*****************************
por_con:
        ;push acc
        ;push dpl
        ;push dph
       mov dptr,#portp
       mov a,#80h
       movx @dptr,a
       mov r2,#1
       lcall delaylcd
       mov dptr,#portc
       movx a,@dptr
       ret
por_con1:
        pop dph
        pop dpl
        ;pop acc
        ret


;*****************************
;*  Clear and Shife memory*
;*****************************
clear_mem:
      mov a,21h
      cjne a,#01h,pass
      mov a,20h
      cjne a,#00h,pass
      jmp main2


pass:
      mov a,75h
      mov dpl,a
      mov a,74h
      mov dph,a

      mov a,3dh
      subb a,#01
      mov 3dh,a
      jc tt1
  tt2:
      mov a,3dh
      add a,3ch
      cjne a,#00,send
      jmp main1
 send:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      mov a,21h
      subb a,41h
      mov 43h,a
      jc aaa
bbb:
      mov a,20h
      subb a,40h
      mov 42h,a
      add a,43h
      jz tto2
      mov 3bh,21h
      mov 3ah,20h
      jmp shife
aaa:
     dec 20h
     clr c
     jmp bbb
tto2:
     mov 3bh,43h
     mov 3ah,42h
     jmp shife

shife:
      mov r7,#8
      inc dptr
      mov a,3bh
      cjne a,#00h,to1
      mov a,3ah
      cjne a,#00h,to1
      mov 3bh,#01h

to1:
      movx a,@dptr
      mov 36h,a
      mov a,dpl
      subb a,#08
      mov dpl,a
      jc to9
to10:
      mov a,36h
      movx @dptr,a
      inc dptr
      mov a,#08
      addc a,dpl
      mov dpl,a
      jc to11
to12:
      djnz r7,to1
      mov a,dpl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    subb a,#01
    mov dpl,a
    jc to13
to14:
    mov a,3bh
    subb a,#01
    mov 3bh,a
    jc to17
to18:
    mov a,3bh
    add a,3ah
    cjne a,#00,shife
    mov a,21h
    subb a,#01
    mov 21h,a
    jc to19
to20:
    mov 3dh,21h
    mov 3ch,20h
    mov 3eh,#00
    mov 3fh,#00
    mov 40h,#00
    mov 41h,#00
    jmp main1
 to9:
    mov a,dph
    dec a
    clr c
    mov dph,a
    jmp to10
to11:
    mov a,dph
    inc a
    mov dph,a
    clr c
    jmp to12
to13:
    mov a,dph
    dec a
    mov dph,a
    clr c
    jmp to14
to17:
    dec 3ah
    clr c
    jmp to18
to19:
    dec 20h
    clr c
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      jmp to20
;******************
;*  LCD REAL TIME *
;******************


LCDMAIN: MOV   DPTR,#PORTP      ;SET 8255 CONTROL PORT
        MOV   A,#80H
        MOVX  @DPTR,A
        MOV   R2,#2           ;FIRST DELAY
        LCALL  DELAYLCD
        MOV   A,#00111000B     ;FUNCTION SET
        LCALL  LCDWI
        MOV   A,#00001100B     ;DISPLAY ON/OFF
        LCALL  LCDWI
        MOV   A,#01H          ;CLEAR
        LCALL  LCDWI
        RET
LCDMAIN1: MOV   DPTR,#PORTP      ;SET 8255 CONTROL PORT
        MOV   A,#80H
        MOVX  @DPTR,A
        MOV   R2,#2           ;FIRST DELAY
        LCALL  DELAYLCD
        MOV   A,#00111000B     ;FUNCTION SET
        LCALL  LCDWI
        MOV   A,#00001111B     ;DISPLAY ON/OFF
        LCALL  LCDWI
        RET

; ********** LCD_dis_mem **********
; LOAD DATA TO LCD-MODULE
; IN  = DPTR  START BLOCK (16 BYTE)
; REG = A,R2,DPTR

LCD_dis_mem:
            mov    a,#080h
            lcall  lcdwi
            mov    a,year
            lcall show_1
            lcall  LCDWD
            lcall show_2
            lcall  LCDWD
            mov   a,#83h
            lcall lcdwi
            mov   a,month
            lcall show_1
            lcall lcdwd
            lcall show_2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall lcdwd
        mov  a,#086h
        lcall lcdwi
        mov  a,date
        lcall show_1
        lcall lcdwd
        lcall show_2
        lcall lcdwd
        mov a,#0c1h
        lcall lcdwi
        mov a,hr
        lcall show_1
        lcall lcdwd
        lcall show_2
        lcall lcdwd

        mov    a,#0c3h
        lcall  lcdwi
        mov    a,#0a0h
        lcall show_1
        lcall  lcdwd

        mov a,#0c4h
        lcall lcdwi
        mov a,min
        lcall show_1
        lcall lcdwd
        lcall show_2
        lcall lcdwd
        mov a,#0c6h
        lcall lcdwi
        mov a,sec
        lcall show_1
        lcall lcdwd
        lcall show_2
        lcall lcdwd
        RET
;***************************************
;*          DATA LCD                   *
;***************************************

kmitl:        DB    "  KMIT'L 1999   "
ins_year:     DB    "Insert year     "
ins_month:    DB     "Insert month    "
ins_date:     DB    "Insert date     "
ins_hour:     DB    "Insert hour     "
ins_min:      DB    "Insert min      "
ins_sec:      DB    "Insert sec      "
ins_channal:  DB     "Insert channal  "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ins_on_off:    DB    "Insert on,off  "
ins_mode:      DB    "Insert mode    "
ins_data:      DB    "Ins. data please"
cancel:        DB    "   cancel      "
real_time:     DB    " set real time "
set_time:      DB    "   set time    "
completed:     DB    "  completed    "
invert_ch:     DB    " Invert Ch     "
;**********************
;*     Load LCD     *
;**********************
 lcd_invert_ch:
            lcall lcdmain
            mov dptr,#invert_ch
            lcall lcdld
            lcall show_4
            mov r2,#5
            lcall delaylcd
            ret

 lcd_kmitl:    lcall lcdmain
            mov dptr,#kmitl
            lcall lcdld
            mov r2,#10
            lcall delaylcd
            ret
  lcd_year:    lcall lcdmain
            mov dptr,#ins_year
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
 lcd_month:    lcall lcdmain
            mov dptr,#ins_month
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
  lcd_date:    lcall lcdmain
            mov dptr,#ins_date
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
  lcd_hour:    lcall lcdmain
            mov dptr,#ins_hour
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_min:    lcall lcdmain
            mov dptr,#ins_min
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_sec:    lcall lcdmain
            mov dptr,#ins_sec
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_channal:    lcall lcdmain
            mov dptr,#ins_channal
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_on_off:    lcall lcdmain
            mov dptr,#ins_on_off
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_mode:    lcall lcdmain
            mov dptr,#ins_mode
            lcall lcdld
            lcall show_3
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_cancel:    lcall lcdmain
            mov dptr,#cancel
            lcall lcdld
            mov r2,#5
            lcall delaylcd
            ret
lcd_real_time:  lcall lcdmain
            mov dptr,#real_time
            lcall lcdld
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov r2,#5
            lcall delaylcd
            ret
lcd_set_time:   lcall lcdmain
            mov dptr,#set_time
            lcall lcdld
            mov r2,#5
            lcall delaylcd
            ret
lcd_completed:  lcall lcdmain
            mov dptr,#completed
            lcall lcdld
            mov r2,#2
            lcall delaylcd
            ret
lcd_data:       lcall lcdmain
            mov dptr,#ins_data
            lcall lcdld
            ;mov r2,#10
            ;lcall delaylcd
            ret
 lcd_year1:    lcall lcdmain
            mov dptr,#ins_year
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
            ret
 lcd_month1:    lcall lcdmain1
            mov dptr,#ins_month
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
            ret
 lcd_date1:    lcall lcdmain
            mov dptr,#ins_date
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL LCDWD

            lcall show_4
            ;mov r2,#10
            ;lcall delaylcd
            ret
lcd_hour1:  lcall lcdmain
            mov dptr,#ins_hour
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_min1:   lcall lcdmain
            mov dptr,#ins_min
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_sec1:   lcall lcdmain
            mov dptr,#ins_sec
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
            ret
lcd_channal1:   lcall lcdmain
            mov dptr,#ins_channal
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ret
lcd_on_off1:    lcall lcdmain
            mov dptr,#ins_on_off
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
            ret
 lcd_mode1:    lcall lcdmain
            mov dptr,#ins_mode
            lcall lcdld
            MOV A,#0C6H
            LCALL LCDWI
            MOV A,TEMP
            LCALL LCDWD
            lcall show_4
            ;mov r2,#1
            ;lcall delaylcd
            ret


show_1:  mov    b,a
      swap   a
      anl    a,#0fh
      add    a,#30h
      ret

show_2:  mov    a,b
      anl    a,#0fh
      add    a,#30h
      ret
show_3:  lcall lcdmain1
      mov a,#0c6h
      lcall lcdwi
      ret
show_4:  lcall lcdmain1
      mov a,#0c7h
      lcall lcdwi
      ret

; ********** DELAY SUB **********
; DELAY SUBROUTINE
; IN  = R2
; REG = R2,R3,R4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DELAYLCD:  MOV   R3,#0
DELAY1:  MOV   R4,#0
      DJNZ  R4,$
      DJNZ  R3,DELAY1
      DJNZ  R2,DELAYLCD
      RET

; ********** LCDWI **********
; LCD WRITE INSTRUCTION (RS=0)
; IN  = A
; REG = A,DPTR

LCDWI:  MOV   DPTR,#PORTA     ;DATA TO PORTA
      MOVX  @DPTR,A
      MOV   DPTR,#PORTB     ;PORTB READ MODIFY WRITE
      MOVX  A,@DPTR
      CLR   ACC.0          ;RS=0
      CLR   ACC.1          ;R/W=0
      CLR   ACC.2          ;ENABLE=0
      MOVX  @DPTR,A
      SETB  ACC.2          ;ENABLE=1
      MOVX  @DPTR,A
      CLR   ACC.2          ;ENABLE=0
      MOVX  @DPTR,A
      MOV   A,#0           ;DELAY
LCDWI1:  DEC   A
      JNZ   LCDWI1
      RET

; ********** LCDWD **********
; LCD WRITE DATA (RS=1)
; IN  = A
; REG = A,DPTR

LCDWD:  MOV   DPTR,#PORTA     ;DATA TO PORTA
      MOVX  @DPTR,A
      MOV   DPTR,#PORTB     ;PORTB READ MODIFY WRITE
      MOVX  A,@DPTR
      SETB  ACC.0          ;RS=1
      CLR   ACC.1          ;R/W=0
      CLR   ACC.2          ;ENABLE=0
      MOVX  @DPTR,A
      SETB  ACC.2          ;ENABLE=1
      MOVX  @DPTR,A
      CLR   ACC.2          ;ENABLE=0
      MOVX  @DPTR,A
      MOV   A,#0           ;DELAY
LCDWD1:  DEC   A
      JNZ   LCDWD1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET

; ********** LCDDIS **********
; LOAD DATA TO LCD-MODULE
; IN  = DPTR  START BLOCK (40 BYTE)
; REG = A,R2,DPTR

LCDLD:  MOV    A,#80H          ;SET ADDRESS LINE 1
        LCALL  LCDLDS
        mov    a,#0c0h
        lcall  lcdlds
        RET

LCDLDS:  PUSH   DPH             ;LOAD SUB.
        PUSH   DPL
        LCALL  LCDWI           ;WRITE ADDRESS
        POP    DPL
        POP    DPH
        MOV    R2,#8           ;16 CHAR.
LCDLDS1: CLR    A
        MOVC   A,@A+DPTR       ;MOVC FOR CODE / MOVX FO
R DATA
        PUSH   DPH
        PUSH   DPL
        LCALL  LCDWD           ;WRITE DATA
        POP    DPL
        POP    DPH
        INC    DPTR
        DJNZ   R2,LCDLDS1
        RET

;***************
;* DELAY 2 SEC *
;***************

DEBOUCE:
            MOV       R5,#0EH ;(1)
LOOP3:      MOV       R6,#00H ;(1)
LOOP2:      MOV       R7,#00H ;(1)
LOOP1:      DJNZ      R7,LOOP1;(2)
            DJNZ      R6,LOOP2;(2)
            DJNZ      R5,LOOP3;(2)
            RET               ;(2)

;*************
;* READ TIME *
;*************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
READ_TIME:
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------READ YEAR--------------

            MOV         R2,#10001101B
            CALL        RTC_READ
            MOV         YEAR,R3

;-------------READ MONTH--------------

            MOV         R2,#10001001B
            CALL        RTC_READ
            MOV         MONTH,R3

;-------------READ DATE---------------

            MOV         R2,#10000111B
            CALL        RTC_READ
            MOV         DATE,R3

;-------------READ HOUR--------------

            MOV         R2,#10000101B
            CALL        RTC_READ
            MOV         HR,R3

;-------------READ MIN--------------

            MOV         R2,#10000011B
            CALL        RTC_READ
            MOV         MIN,R3

;-------------READ SEC--------------

            MOV         R2,#10000001B
            CALL        RTC_READ
            MOV         SEC,R3
            RET
RTC_WRITE_8bits:
            MOV         R4,#08H             ;8
 bits transfer
WR_8bit1:
            RRC         A
            MOV         RTC_DATA,C
            SETB        RTC_CLK             ;R
ising edge clock
            CALL        DELAY
            CLR         RTC_CLK
            CALL        DELAY
            DJNZ        R4,WR_8bit1
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;************************
;BYTE READ  DS1202     *
; R2=COMMAND           *
; R3=DATA              *
;************************

RTC_READ:
        CLR     RTC_CLK
        CALL    DELAY
        SETB    RTC_RST
        CALL    DELAY
        MOV     A,R2
        CALL    RTC_WRITE_8bits
        MOV     R4,#08H             ;T
HEN READ DATA BYTE
        CLR     A
RD1:    CLR     RTC_CLK
        CALL    DELAY
        MOV     C,RTC_DATA
        RRC     A
        SETB    RTC_CLK
        CALL    DELAY
        DJNZ    R4,RD1
        MOV     R3,A
        CLR     RTC_RST
        CALL    DELAY
        RET

DELAY:
        NOP
        RET
;********************
;* Set Time DS1202 *
;********************

SET_TIME_DS1202:

;************************
;* Open write protect *
;************************
        MOV     R2,#10001110B       ;
CONTROL COMMAND
        MOV     R3,#00H
        CALL    RTC_WRITE
;***************
;* write time *
;***************
;----------Write Year------------------
```

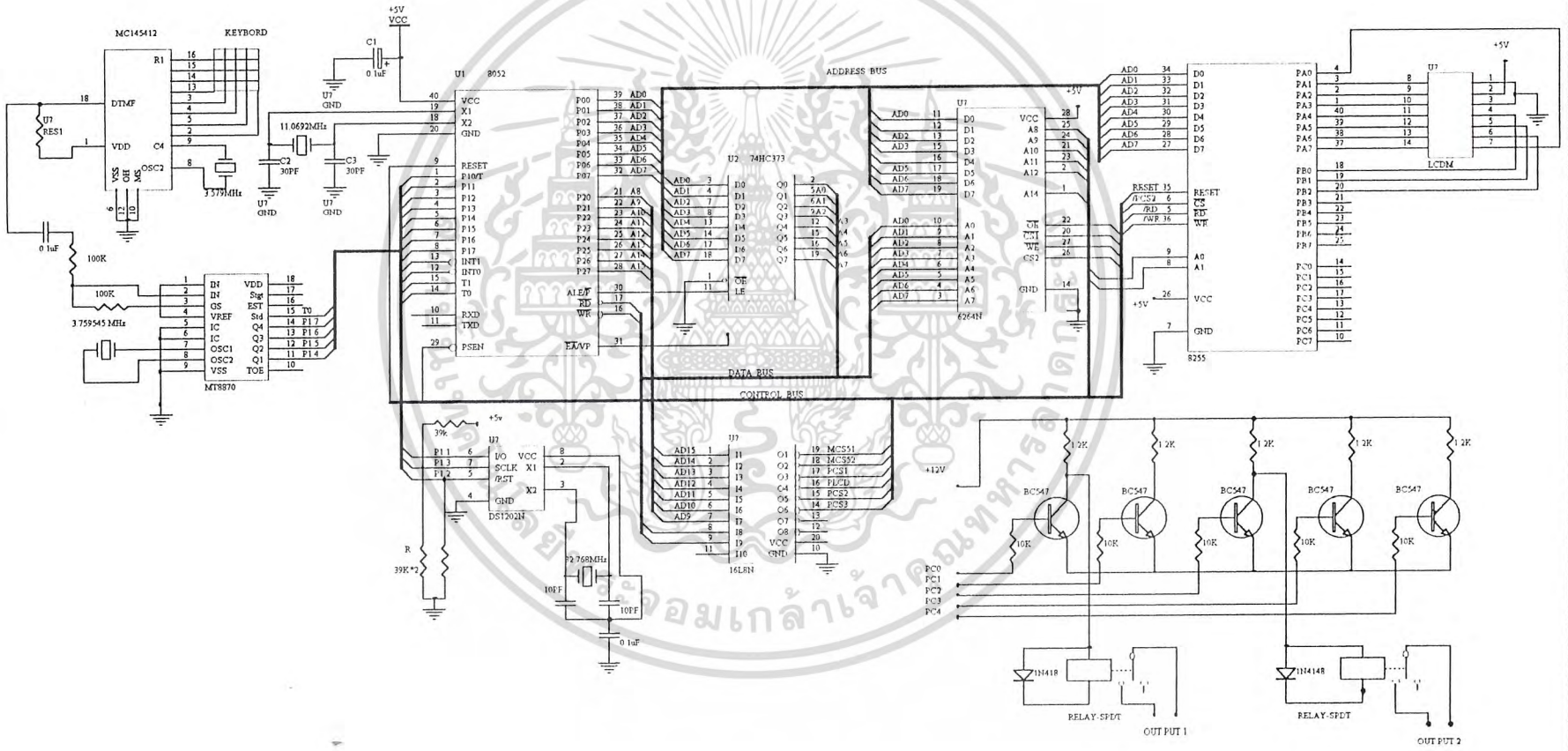
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV         R2,#10001100B
            MOV         R3,50h
            CALL        RTC_WRITE

;----------Write Month----------------

            MOV         R2,#10001000B
            MOV         R3,51h
            CALL        RTC_WRITE
;----------Write Date-----------------

            MOV         R2,#10000110B
            MOV         R3,52h
            CALL        RTC_WRITE

;----------Write Hour-----------------

            MOV         R2,#10000100B
            MOV         R3,53h
            CALL        RTC_WRITE

;----------Write Min------------------

            MOV         R2,#10000010B
            MOV         R3,54h
            CALL        RTC_WRITE

;----------Write Sec------------------

            MOV         R2,#10000000B
            MOV         R3,55h
            CALL        RTC_WRITE

;****************
; write protect *
;****************

            MOV         R2,#10001110B
            MOV         R3,#80H
            CALL        RTC_WRITE
            RET
;************************
; BYTE WRITE  DS1202    *
; R2 = COMMAND          *
; R3 = DATA             *
;************************

RTC_WRITE:
            CLR         RTC_CLK         ;C
```

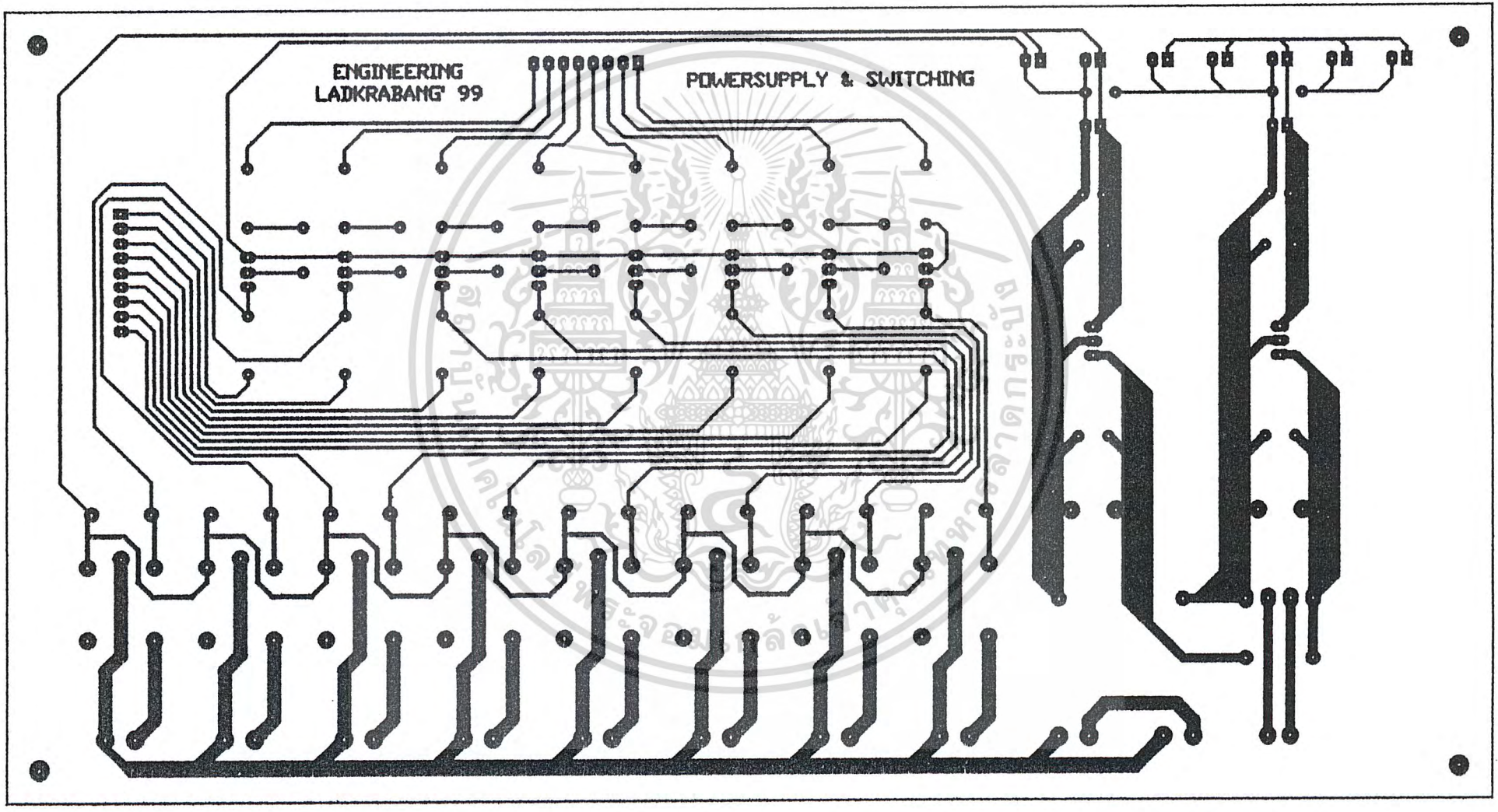
เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

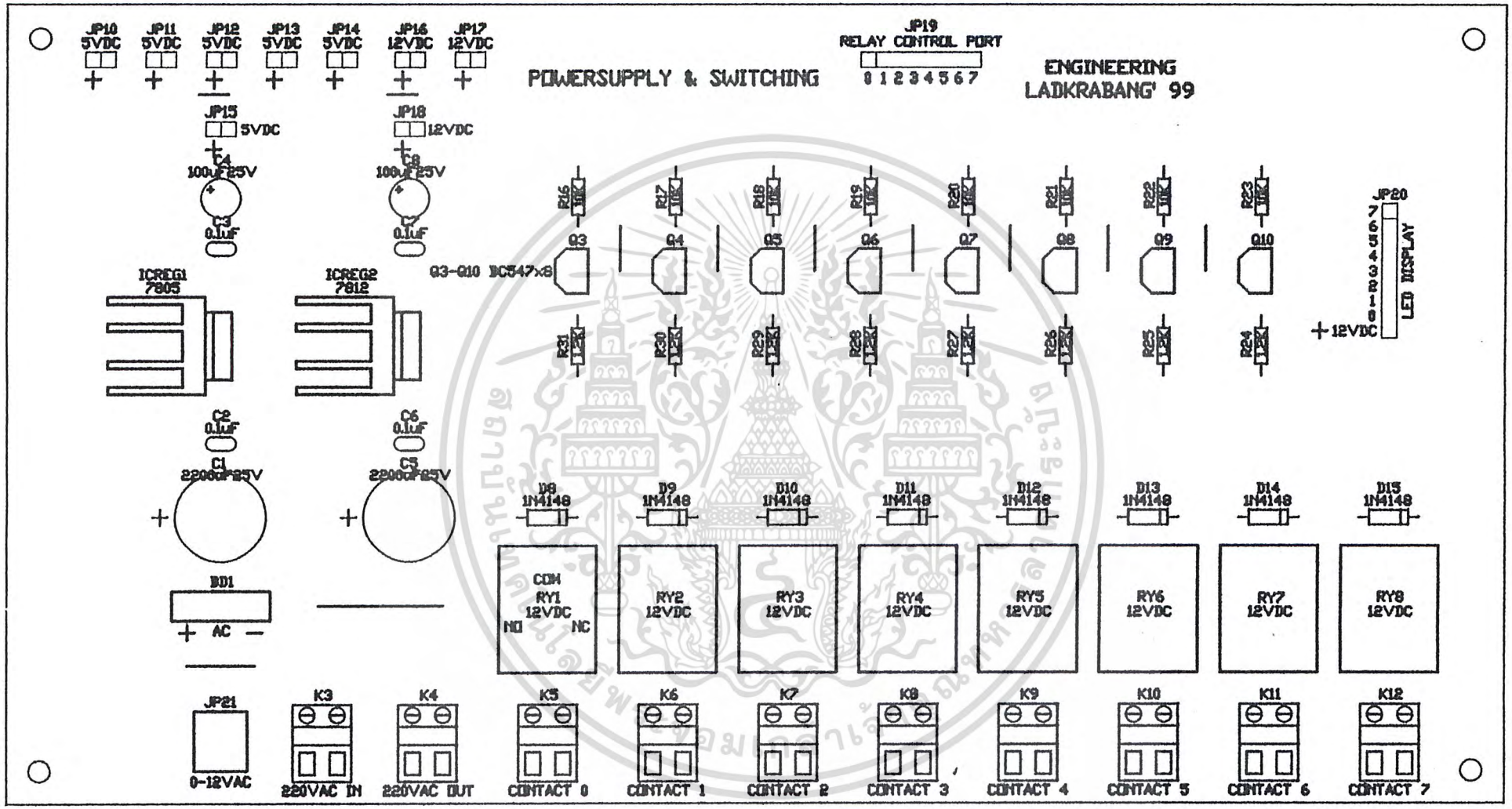
```
LK = "0"
        CALL        DELAY
        SETB        RTC_RST          ;R
ST = "1"
        CALL        DELAY
        MOV         A,R2             ;W
RITE COMMAND BYTE
        CALL        RTC_WRITE_8bits
        MOV         A,R3             ;W
RITE DATA BYTE
        CALL        RTC_WRITE_8bits
        CLR         RTC_RST          ;R
ST = "0"
        CALL        DELAY
        ;jmp gi2
        RET

copy1:  mov 60h,r1
        mov 61h,r2
        mov 62h,r3
        mov 63h,r4
        mov 64h,r5
        mov 65h,r6
        mov 66h,r7
        ret
copy2:  mov r1,60h
        mov r2,61h
        mov r3,62h
        mov r4,63h
        mov r5,64h
        mov r6,65h
        mov r7,66h
        ret

      end
```

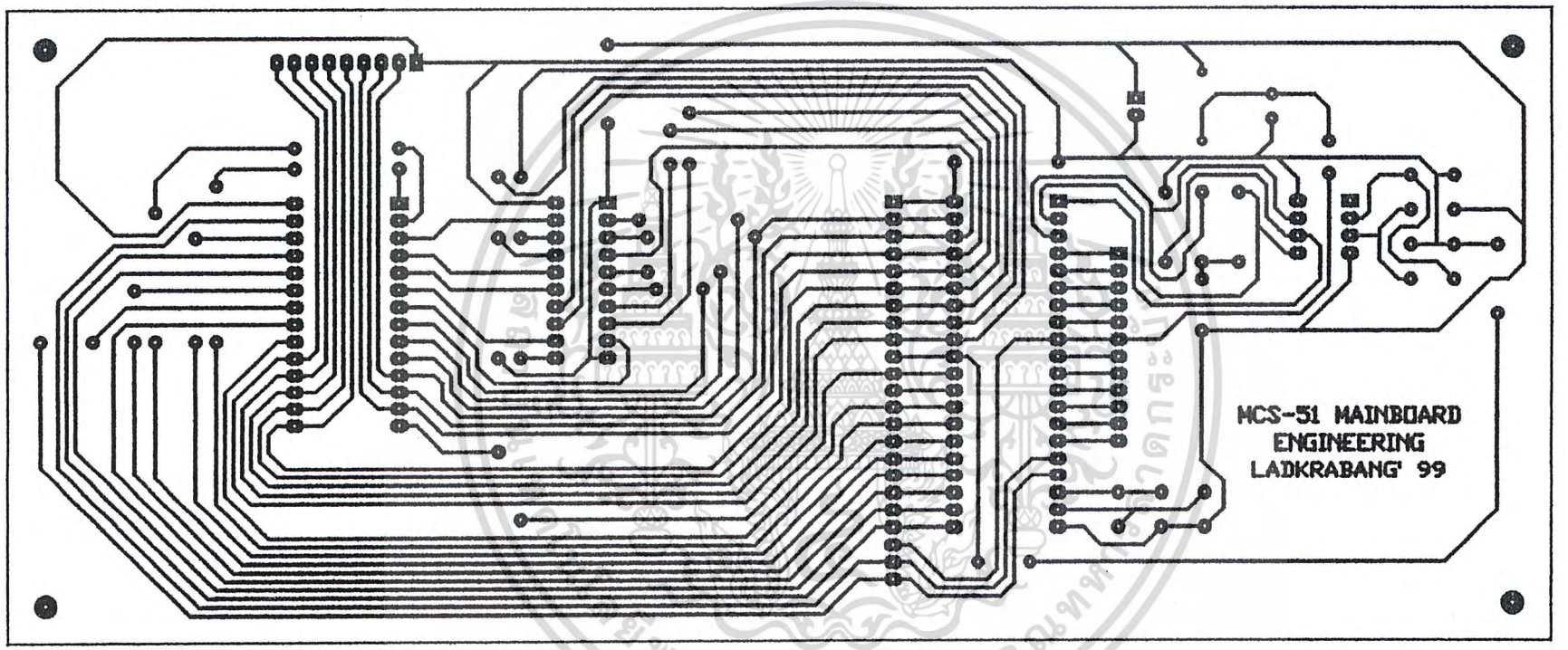
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

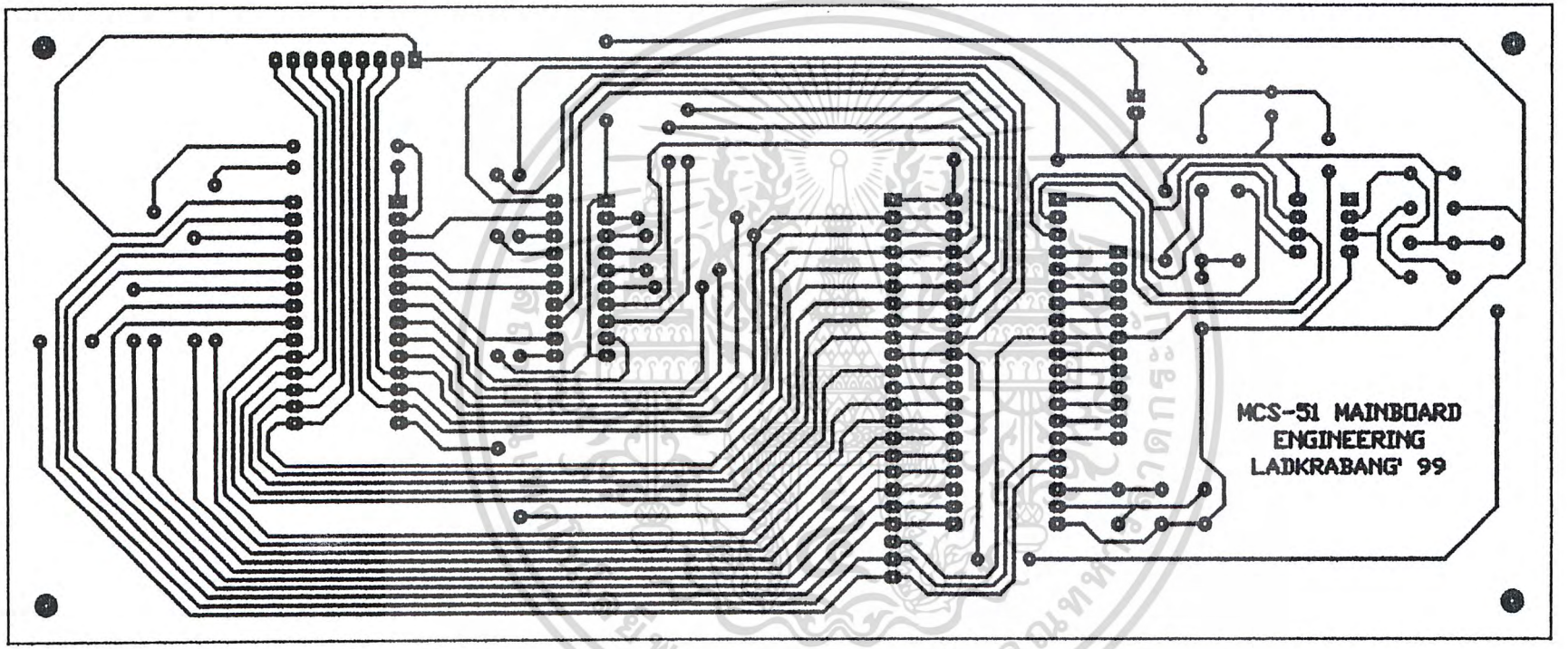


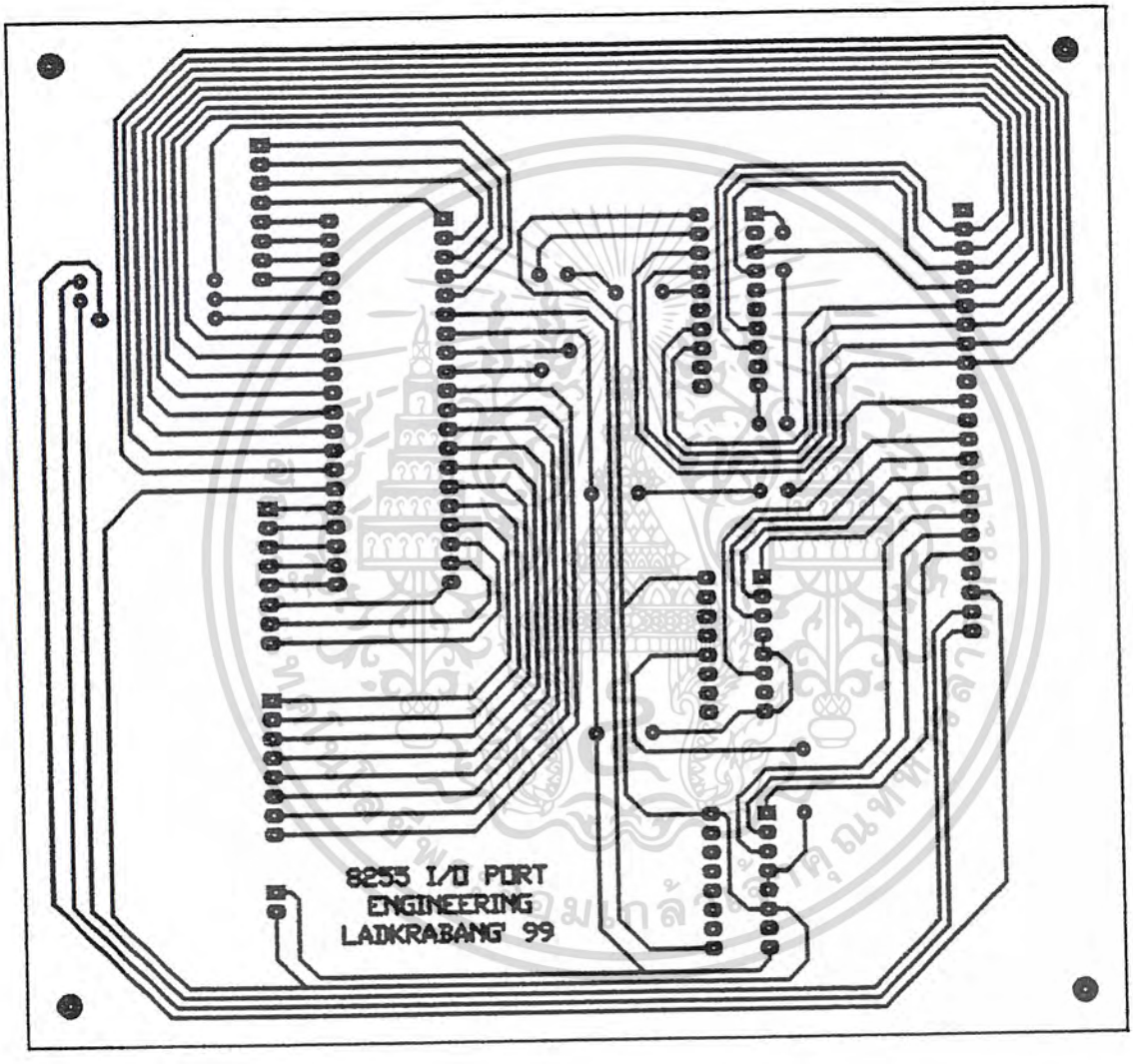
Title		
Size: B	Number	Revision
Date: 28-Mar-1999		Sheet of
File: AVROJ1.SCH		Drawn By


ENGINEERING
LADKRABANG' 99
POWERSUPPLY & SWITCHING

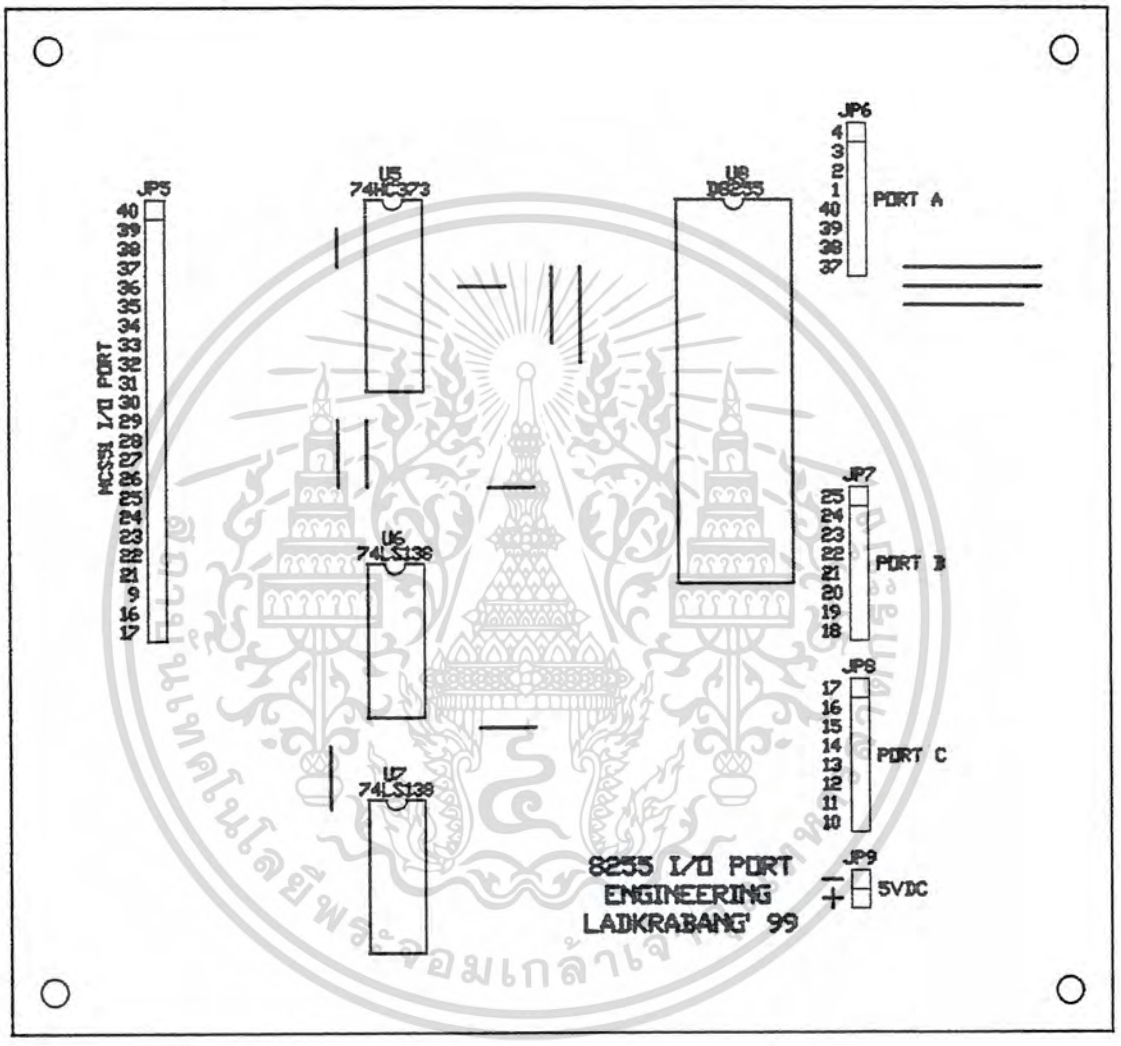

JP10 5VDC
JP11 5VDC
JP12 5VDC
JP13 5VDC
JP14 5VDC
JP16 12VDC
JP17 12VDC
POWERSUPPLY & SWITCHING
JP19
RELAY CONTROL PORT
0 1 2 3 4 5 6 7
ENGINEERING
LADKRABANG' 99
JP15 5VDC
JP18 12VDC
C4 100uF25V
C8 100uF25V
C3 0.1uF
C7 0.1uF
ICREG1 7805
ICREG2 7812
Q3-Q10 BC547x8
R16 10K
R17 10K
R18 10K
R19 10K
R20 10K
R21 10K
R22 10K
R23 10K
R31 12K
R30 12K
R29 12K
R28 12K
R27 12K
R26 12K
R25 12K
R24 12K
JP20
LED DISPLAY
+12VDC
C2 0.1uF
C6 0.1uF
C1 2200uF25V
C5 2200uF25V
BD1
AC
D8 1N4148
D9 1N4148
D10 1N4148
D11 1N4148
D12 1N4148
D13 1N4148
D14 1N4148
D15 1N4148
COM
RY1 12VDC
NO NC
RY2 12VDC
RY3 12VDC
RY4 12VDC
RY5 12VDC
RY6 12VDC
RY7 12VDC
RY8 12VDC
JP21
0-12VAC
K3
220VAC IN
K4
220VAC OUT
K5
CONTACT 0
K6
CONTACT 1
K7
CONTACT 2
K8
CONTACT 3
K9
CONTACT 4
K10
CONTACT 5
K11
CONTACT 6
K12
CONTACT 7


MCS-51 MAINBOARD
ENGINEERING
LADKRABANG' 99


MCS-51 MAINBOARD
ENGINEERING
LADKRABANG' 99



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลของอุปกรณ์ที่ใช้ และข้อมูลทางไฟฟ้าของ IC TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

DS1202, DS1202S
Serial Timekeeping Chip

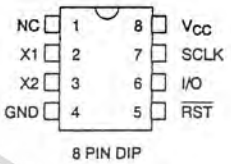
FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation
- 24 x 8 RAM for scratchpad data storage
- Serial I/O for minimum pin count
- 2.0-5.5 volt full operation
- Uses less than 300 nA at 2 volts
- Single-byte or multiple-byte (burst mode) data transfer for read or write of clock or RAM data
- 8-pin DIP or optional 16-pin SOIC for surface mount
- Simple 3-wire interface
- TTL-compatible (V_{CC} = 5V)
- Optional industrial temperature range -40°C to +85°C

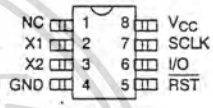
ORDERING INFORMATION

DS1202	8–pin DIP
DS1202S	16–pin SOIC
DS1202S8	8–pin SOIC

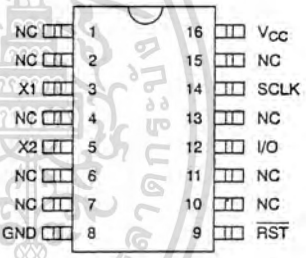
PIN ASSIGNMENT



8 PIN DIP



8–PIN SOIC
(208 mil)



16–PIN SOIC

PIN DESCRIPTION

NC	–	No Connection
X1, X2	–	32.768 KHz Crystal Input
GND	–	Ground
$\overline{RST}$	–	Reset
I/O	–	Data Input/Output
SCLK	–	Serial Clock
V_{CC}	–	Power Supply Pin

DESCRIPTION

The DS1202 Serial Timekeeping Chip contains a real time clock/calendar and 24 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The real time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Interfacing the DS1202 with a microprocessor is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: (1) $\overline{RST}$ (Reset), (2) I/O (Data line), and (3) SCLK (Serial clock). Data can be transferred to and from the clock/

Copyright © 1993 Dallas Semiconductor Corporation
Printed in U.S.A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RAM one byte at a time or in a burst of up to 24 bytes. The DS1202 is designed to operate on very low power and retain data and clock information on less than 1 microwatt.

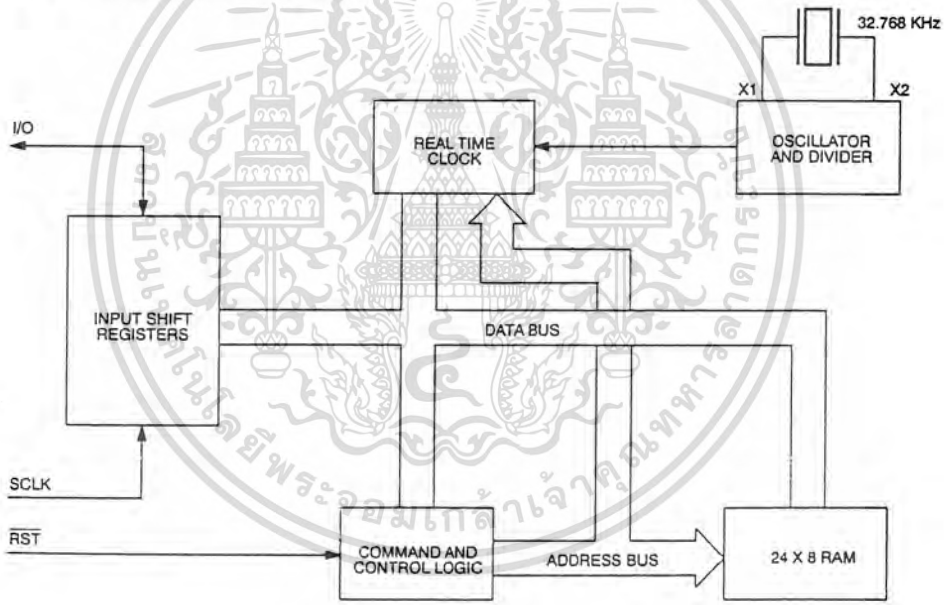
OPERATION

The main elements of the Serial Timekeeper are shown in Figure 1: shift register, control logic, oscillator, real time clock, and RAM. To initiate any transfer of data, $\overline{RST}$ is taken high and eight bits are loaded into the shift register providing both address and command information. Data is serially input on the rising edge of the SCLK. The first eight bits specify which of 32 bytes will be accessed, whether a read or write cycle will take place, and whether a byte or burst mode transfer is to occur. After the first eight clock cycles have occurred which load the command word into the shift register, additional clocks will output data for a read or input data for a write. The number of clock pulses equals eight plus eight for byte mode or eight plus up to 192 for burst mode.

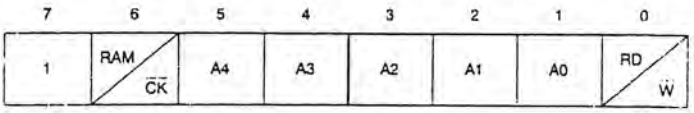
COMMAND BYTE

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

DS1202 BLOCK DIAGRAM Figure 1



ADDRESS/COMMAND BYTE Figure 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESET AND CLOCK CONTROL

All data transfers are initiated by driving the $\overline{RST}$ input high. The $\overline{RST}$ input serves two functions. First, $\overline{RST}$ turns on the control logic which allows access to the shift register for the address/command sequence. Second, the $\overline{RST}$ signal provides a method of terminating either single byte or multiple byte data transfer. A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, data must be valid during the rising edge of the clock and data bits are output on t he falling edge of clock. All data transfer terminates if the $\overline{RST}$ input is low and the I/O pin goes to a high impedance state. Data transfer is illustrated in Figure 3.

DATA INPUT

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

DATA OUTPUT

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

BURST MODE

Burst mode may be specified for either the clock/calendar or the RAM registers by addressing location 31 decimal (address/command bits one through five = logical one). As before, bit six specified clock or RAM and bit 0 specifies read or write. There is no data storage capacity at locations 8 through 31 in the Clock/Calendar Registers or locations 24 through 31 in the RAM registers. When writing to the clock registers in the burst mode, the first eight registers must be written in order for the data to be transferred.

However, when writing to RAM in burst mode it is not necessary to write all 24 bytes for the data to transfer. Each byte that is written to will be transferred to RAM regardless of whether all 24 bytes are written or not.

CLOCK/CALENDAR

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

CLOCK HALT FLAG

Bit 7 of the seconds register is defined as the clock halt flag. When this bit is set to logic 1, the clock oscillator is stopped and the DS1202 is placed into a low-power standby mode with a current drain of not more than 100 nanoamps. When this bit is written to logic 0, the clock will start.

AM-PM/12-24 MODE

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

WRITE PROTECT REGISTER

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

CLOCK/CALENDAR BURST MODE

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

RAM

The static RAM is 24 x 8 bytes addressed consecutively in the RAM address space.

RAM BURST MODE

The RAM command byte specifies burst mode operation. In this mode, the 24 RAM registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REGISTER SUMMARY

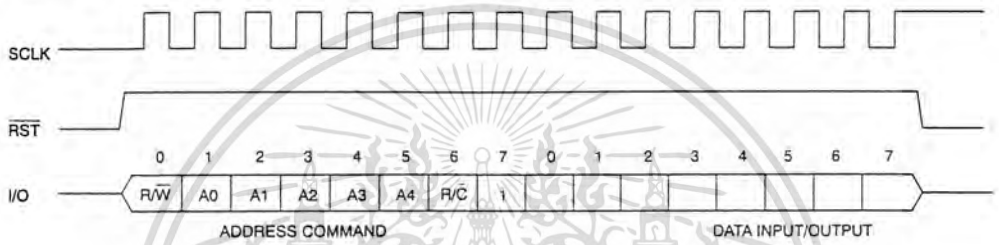
A register data format summary is shown in Figure 4.

CRYSTAL SELECTION

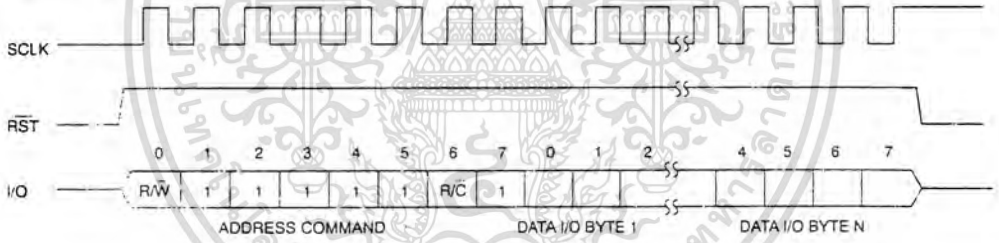
A 32.768 KHz crystal, Daiwa Part No. DT26S, Seiko Part No. DS-VT-200 or equivalent, can be directly connected to the DS1202 via pins 2 and 3 (X1, X2). The crystal selected for use should have a specified load capacitance (CL) of 6 pF. The crystal is connected directly to the X1 and X2 pins. There is no need for external capacitors or resistors. Note: X1 and X2 are very high impedance nodes. It is recommended that they and the crystal be guard–ringed with ground and that high frequency signals be kept away from the crystal area. For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks". Crystals can be ordered from Dallas Semiconductor. Order part number DS9032.

DATA TRANSFER SUMMARY Figure 3

SINGLE BYTE TRANSFER



BURST MODE TRANSFER



FUNCTION	BYTE N	SCLK n
CLOCK	8	72
RAM	24	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REGISTER ADDRESS/DEFINITION Figure 4

REGISTER ADDRESS
A. CLOCK

	7	6	5	4	3	2	1	0
SEC	1	0	0	0	0	0	0	RD/$\overline{W}$
MIN	1	0	0	0	0	0	1	RD/$\overline{W}$
HR	1	0	0	0	0	1	0	RD/$\overline{W}$
DATE	1	0	0	0	0	1	1	RD/$\overline{W}$
MONTH	1	0	0	0	1	0	0	RD/$\overline{W}$
DAY	1	0	0	0	1	0	1	RD/$\overline{W}$
YEAR	1	0	0	0	1	1	0	RD/$\overline{W}$
CONTROL	1	0	0	0	1	1	1	RD/$\overline{W}$
CLOCK BURST	1	0	1	1	1	1	1	RD/$\overline{W}$

REGISTER DEFINITION

00–59	CH	10 SEC				SEC
00–59	0	10 MIN				MIN
01–12 00–23	12/24	0	10 A/P	HR		HR
01–28/29 01–30 01–31	0	0	10 DATE			DATE
01–12	0	0	0	10 M		MONTH
01–07	0	0	0	0	0	DAY
0–99	10 YEAR					YEAR
	WP	FORCED TO ZERO				

B. RAM

RAM 0	1	1	0	0	0	0	0	RD/$\overline{W}$
RAM 23	1	1	1	0	1	1	1	RD/$\overline{W}$
RAM BURST	1	1	1	1	1	1	1	RD/$\overline{W}$

RAM DATA 0

RAM DATA 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	–0.3V to +7.0V
Operating Temperature	0°C to 70°C
Storage Temperature	–55°C to +125°C
Soldering Temperature	260°C for 10 seconds

* 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.

RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C)

PARAMETER	SYMBOL		MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}		2.0		5.5	V	1
Logic 1 Input	V_{IH}		2.0		V_{CC}+0.3	V	1
Logic 0 Input	V_{IL}	V_{CC}=2.0V	–0.3		+0.3	V	1
		V_{CC}=5V	–0.3		+0.8		

DC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; V_{CC} = 2.0 to 5.5V*)

PARAMETER	SYMBOL		MIN	TYP	MAX	UNITS	NOTES
Input Leakage	I_{LI}				+500	µA	6
I/O Leakage	I_{LO}				+500	µA	6
Logic 1 Output	V_{OH}	V_{CC}=2V	1.6			V	2
		V_{CC}=5V	2.4				
Logic 0 Output	V_{OL}	V_{CC}=2V			0.4	V	3
		V_{CC}=5V			0.4		
Active Supply Current	I_{CC}	V_{CC}=2V			.4	mA	5
		V_{CC}=5V			1.2		
Timekeeping Current	I_{CC1}	V_{CC}=2V			0.3	µA	4
		V_{CC}=5V			1		
Leakage Current	I_{CC2}	V_{CC}=2V			100	nA	10
		V_{CC}=5V			100		

*Unless otherwise noted.

CAPACITANCE

(t_A = 25°C)

PARAMETER	SYMBOL	CONDITION	TYP	MAX	UNITS	NOTES
Input Capacitance	C_I		5		pF	
I/O Capacitance	$C_{I/O}$		10		pF	
Crystal Capacitance	C_X		6		pF	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS

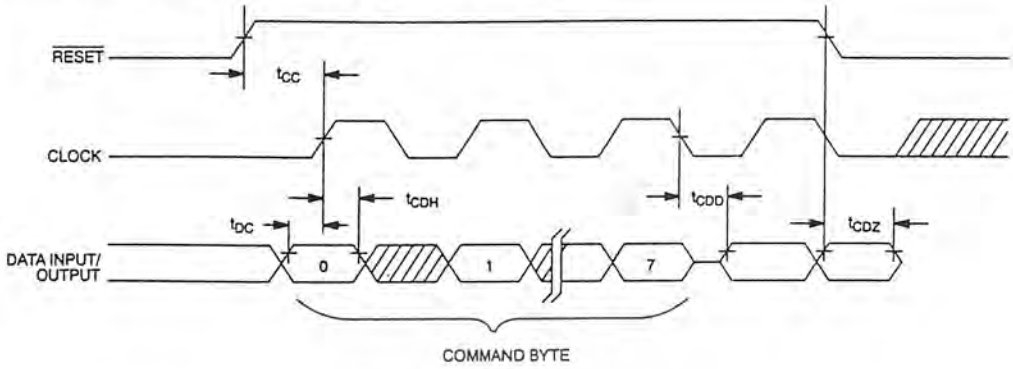
(0°C to 70°C; V_{CC} = 2.0 to 5.5V*)

PARAMETER	SYMBOL		MIN	TYP	MAX	UNITS	NOTES
Data to CLK Setup	t_{DC}	V_{CC}=2V	200			ns	7
		V_{CC}=5V	50				
CLK to Data Hold	t_{CDH}	V_{CC}=2V	280			ns	7
		V_{CC}=5V	70				
CLK to Data Delay	t_{CDD}	V_{CC}=2V			800	ns	7, 8, 9
		V_{CC}=5V			200		
CLK Low Time	t_{CL}	V_{CC}=2V	1000			ns	7
		V_{CC}=5V	250				
CLK High Time	t_{CH}	V_{CC}=2V	1000			ns	7, 12
		V_{CC}=5V	250				
CLK Frequency	f_{CLK}	V_{CC}=2V			0.5	MHz	7, 12
		V_{CC}=5V	DC		2.0		
CLK Rise and Fall	t_R, t_F	V_{CC}=2V			2000	ns	
		V_{CC}=5V			500		
RST to CLK Setup	t_{CC}	V_{CC}=2V	4			µs	7
		V_{CC}=5V	1				
CLK to RST Hold	t_{CCH}	V_{CC}=2V	1000			ns	7
		V_{CC}=5V	250				
RST Inactive Time	t_{CWH}	V_{CC}=2V	4			µs	7
		V_{CC}=5V	1				
RST to I/O High Z	t_{CDZ}	V_{CC}=2V			280	ns	7
		V_{CC}=5V			70		

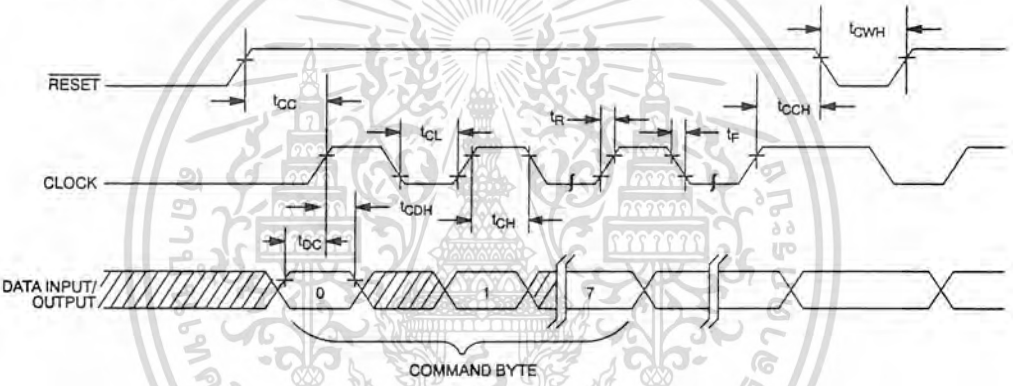
*Unless otherwise noted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING DIAGRAM: READ DATA TRANSFER Figure 5



TIMING DIAGRAM: WRITE DATA TRANSFER Figure 6



NOTES:

1. All voltages are referenced to ground.
2. Logic one voltages are specified at a source current of 1 mA at V_{CC}=5V and .4 mA at V_{CC}=2V, V_{OH}=V_{CC} for capacitive loads.
3. Logic zero voltages are specified at a sink current of 4 mA at V_{CC}=5V and 1.5 mA at V_{CC}=2V.
4. I_{CC1} is specified with I/O open, $\overline{RST}$ set to a logic 0, and clock halt flag=0 (oscillator enabled).
5. I_{CC} is specified with the I/O pin open, $\overline{RST}$ high, SCLK=2 MHz at V_{CC}=5V; SCLK=500 KHz, V_{CC}=2V and clock halt flag=0 (oscillator enabled).
6. $\overline{RST}$, SCLK, and I/O all have 40KΩ pulldown resistors to ground.
7. Measured at V_{IH}=2.0V or V_{IL}=0.8V and 10 ms maximum rise and fall time.
8. Measured at V_{OH}=2.4V or V_{OL}=0.4V.
9. Load capacitance = 50 pF.

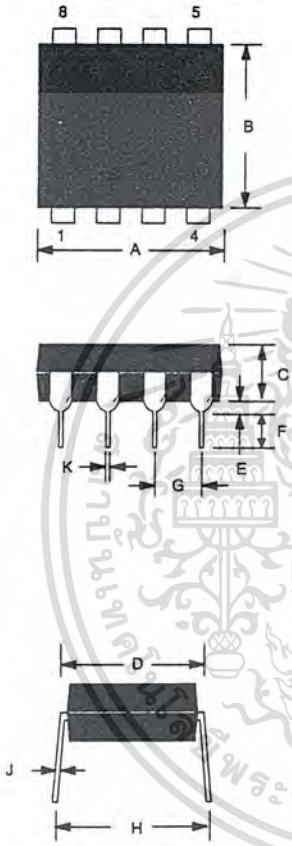
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. I_{CC2} is specified with $\overline{RST}$, I/O, and SCLK open. The clock halt flag must be set to logic one (oscillator disabled).

11. At power–up, $\overline{RST}$ must be at a logic 0 until $V_{CC} \geq 2$ volts. Also, SCLK must be at a logic 0 when $\overline{RST}$ is driven to a logic one state.

12. If t_{CH} exceeds 100 ms with $\overline{RST}$ in a logic one state, then I_{CC} may briefly exceed I_{CC} specification.

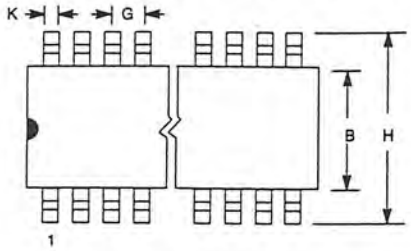
DS1202 SERIAL TIMEKEEPER 8–PIN DIP

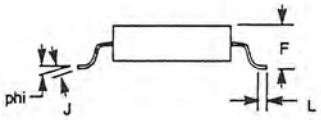


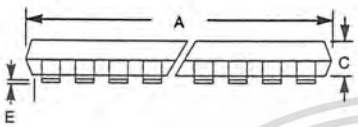
PKG	8–PIN	
DIM	MIN	MAX
A IN. MM	.360	.400
B IN. MM	.240	.260
C IN. MM	.120	.140
D IN. MM	.300	.325
E IN. MM	.015	.040
F IN. MM	.110	.140
G IN. MM	.090	.110
H IN. MM	.320	.370
J IN. MM	.008	.012
K IN. MM	.015	.021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1202S SERIAL TIMEKEEPER 16–PIN SOIC



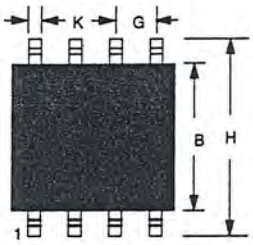


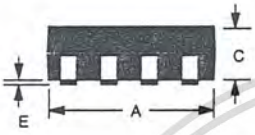


PKG	16-PIN	
DIM	MIN	MAX
A IN. MM	0.500 12.70	0.511 12.99
B IN. MM	0.290 .7.37	0.300 7.65
C IN. MM	0.089 2.26	0.095 2.41
E IN. MM	0.004 0.102	0.012 0.30
F IN. MM	0.094 2.38	0.105 2.68
G IN. MM	0.050 BSC 1.27 BSC	
H IN. MM	0.398 10.11	0.416 10.57
J IN. MM	0.009 0.229	0.013 0.33
K IN. MM	0.013 0.33	0.019 0.48
L IN MM	0.016 0.406	0.040 1.20
phi	0°	8°

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1202S8 8–PIN SOIC 200 MIL





PKG	8-PIN	
DIM	MIN	MAX
A IN. MM	0.203 5.16	0.213 5.41
B IN. MM	0.203 5.16	0.213 5.41
C IN. MM	0.070 1.78	0.074 1.88
E IN. MM	0.004 0.102	0.010 0.390
F IN. MM	0.074 1.88	0.84 2.13
G IN. MM	0.050 BSC 1.27 BSC	
H IN. MM	0.302 7.67	0.318 8.07
J IN. MM	0.006 0.152	0.010 0.254
K IN. MM	0.013 0.33	0.020 0.508
L IN. MM	0.19 4.83	0.030 0.762

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

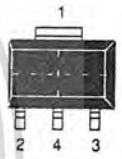
DS1233D
5V EconoReset

FEATURES

- Automatically restarts microprocessor after power failure
- Maintains reset for 350 ms after V_{CC} returns to an in-tolerance condition
- Accurate 5%, 10% or 15% microprocessor 5V power supply monitoring
- Reduces need for discrete components
- Precision temperature-compensated voltage reference and voltage sensor
- Low-cost TO-92 package or surface mount SOT-223 package
- Internal 5 KΩ pull-up resistor
- Compatible with Motorola 68XXX series and HC16 Microprocessors
- Operating temperature of –40°C to +85°C

PIN ASSIGNMENT

TO-92 Package
See Mech. Drawing
Pg. 486



SOT-223 Package
See Mech. Drawing
Pg. 490

PIN DESCRIPTION

PIN 1	GROUND
PIN 2	$\overline{\text{RESET}}$
PIN 3	V_{CC}
PIN 4	GROUND (SOT–223 ONLY)

DESCRIPTION

The DS1233D EconoReset uses a precision temperature compensated reference and comparator circuit to monitor the status of the power supply (V_{CC}). When an out-of-tolerance condition is detected, an internal power fail signal is generated which forces reset to the active state. When V_{CC} returns to an in-tolerance condition, the reset signal is kept in the active state for approximately 350 ms to allow the power supply and processor to stabilize.

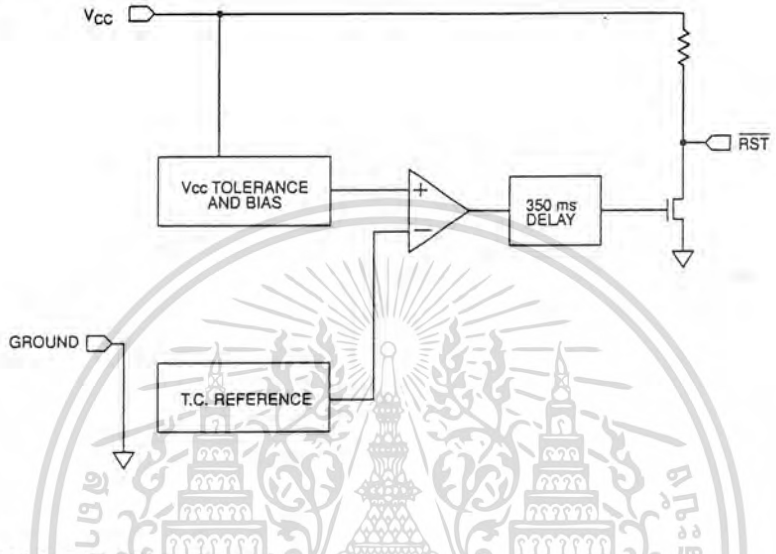
Copyright © 1993 Dallas Semiconductor Corporation
Printed in U.S.A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

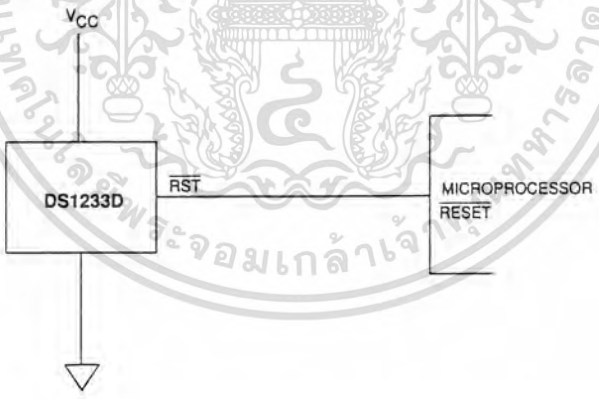
OPERATION - POWER MONITOR

The DS1233D provides the functions of detecting out-of-tolerance power supply conditions and warning a processor-based system of impending power failure. When V_{CC} is detected as out-of-tolerance, as defined by the tolerance of the part selected, the $\overline{RST}$ signal is asserted. On power-up, $\overline{RST}$ is kept active for approximately 350 ms after the power supply has reached the selected tolerance. This allows the power supply and microprocessor to stabilize before $\overline{RST}$ is released.

BLOCK DIAGRAM Figure 1

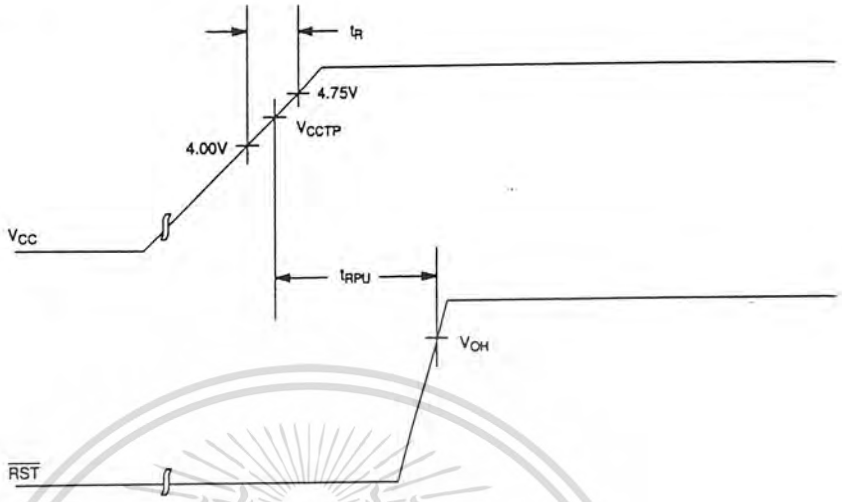


APPLICATION EXAMPLE Figure 2

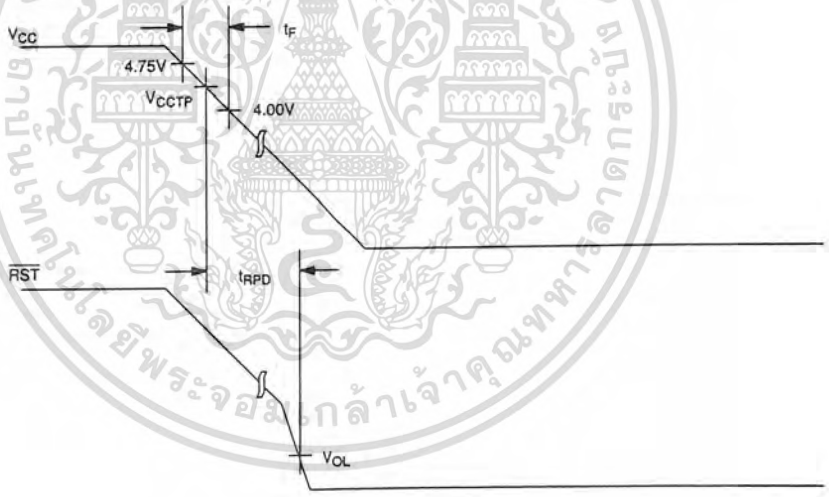


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POWER UP Figure 3



POWER DOWN Figure 4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Voltage on V_{CC} Pin Relative to Ground	-0.5V to +7.0V
Voltage on I/O Relative to Ground	–0.5V to V_{CC} + 0.5V
Operating Temperature	–40°C to +85°C
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds

* 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.

RECOMMENDED DC OPERATING CONDITIONS

(–40°C to +85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	1.2	5.0	5.5	V	1

DC ELECTRICAL CHARACTERISTICS

(–40°C to +85°C; V_{DD} = 5V ± 10%)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Low Level @ $\overline{RST}$	V_{OL}			0.4	V	1
Output Voltage @ –500 µA	V_{OH}	V_{CC}–0.5V	V_{CC}–0.1V		V	1
Output Current @ 2.4V	I_{OH}		350		µA	
Output Current @ 0.4V	I_{OL}	+10			mA	2
Operating Current	I_{CC}			50	µA	
V_{CC} Trip Point 5%	V_{CCTP1}	4.5	4.625	4.74	V	1
V_{CC} Trip Point 10%	V_{CCTP2}	4.25	4.375	4.49	V	1
V_{CC} Trip Point 15%	V_{CCTP3}	4.0	4.125	4.24	V	1
Output Capacitance	C_{OUT}			10	pF	
Internal Pull-Up Resistor	R_P	3.75	5	6.25	KΩ	

AC ELECTRICAL CHARACTERISTICS

(–40°C to +85°C; V_{CC} = 5V ± 10%)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Reset Active Time	t_{RST}	250	350	450	ms	
V_{CC} Detect to $\overline{RST}$	t_{RPD}			100	ns	
V_{CC} Slew Rate (4.75V - 4.00V)	t_F	300			µs	
V_{CC} Slew Rate (4.00V - 4.75V)	t_R	0			ns	
V_{CC} detect to RST	t_{RPU}	250	350	450	ms	

NOTES:

1. All voltages are referenced to ground.
2. A 1KΩ external resistor may be required for proper operation of the microprocessor reset control circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ECONORESET SELECTION GUIDE

		VCC TRIP POINT			PUSHBUTTON DETECT		
		MIN	TYP	MAX	MIN	TYP	MAX
5V	DS1233-15	4.0	4.125	4.24	2.4	–	3.3
	DS1233-10	4.25	4.375	4.49	2.4	–	3.3
	DS1233-5	4.5	4.625	4.75	2.4	–	3.3
	DS1233D–15	4.0	4.125	4.24	N/A		N/A
	DS1233D–10	4.25	4.375	4.49	N/A		N/A
	DS1233D–5	4.5	4.625	4.75	N/A		N/A
	DS1833–15	4.0	4.125	4.24	N/A		N/A
	DS1833–10	4.25	4.375	4.49	N/A		N/A
	DS1833–5	4.5	4.625	4.75	N/A		N/A
3.3V	DS1233A-15	2.64	2.72	2.80	1.8	–	3.0
	DS1233A-10	2.8	2.88	2.97	1.8	–	3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง HD44780

■ Instructions

Instruction	Code										Description	Execution Time (max) (when fcp or fosc is 250 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DD RAM address 0 in address counter.	1.64 ms
Return Home	0	0	0	0	0	0	0	0	1	*	Sets DD RAM address 0 in address counter. Also returns display being shifted to original position. DD RAM contents remain unchanged.	1.64 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies shift of display. These operations are performed during data write and read.	40μs
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of entire display (D), cursor ON/OFF (C), and blink of cursor position character (B).	40μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves cursor and shifts display without changing DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL), number of display lines (L) and character font (F).	40μs
Set CG RAM Address	0	0	0	1	ACG						Sets CG RAM address. CG RAM data is sent and received after this setting.	40μs
Set DD RAM Address	0	0	1	ADD							Sets DD RAM address. DD RAM data is sent and received after this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0μs
Write Data to CG or DD RAM	1	0	Write Data								Writer data into DD RAM or CG RAM.	40μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	40μs
	I/D=1: Increment I/D=0: Decrement S =1: Accompanies display shift S/C=1: Display shift S/C=0: Cursor move R/L=1: Shift to the right R/L=0: Shift to the left DL=1: 8 bits, DL=0: 4 bits N =1: 2 lines, N=0: 1 line F =1: 5×10 dots, F=0: 5×7 dots BF=1: Internally operating BF=0: Can accept instruction										DD RAM: Display data RAM CG RAM: Character generator RAM Acc: CG RAM address Adc: DD RAM Address: Corresponds to cursor address AC: Address counter used for both DD and CG RAM address.	Execution time changes when frequency changes Example: When fcp or fosc is 270 kHz: $40\mu s \times \frac{250}{270} = 37\mu s$

* No effect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของคำสั่ง HD44780

1. CLEAR DISPLAY

Clear display

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	0	0	0	0	1

คำสั่งนี้จะเป็นการเขียนช่องว่างหรือ SPACE (ASCII 20H) เข้าไปใน DD RAM ทั้งหมดและทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ ตำแหน่งบนสุดซ้ายมือของจอภาพ SET I/D = 1, S ไม่มีการเปลี่ยน

2. RETURN HOME

Return home

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	0	0	0	0	*

* No effect

คำสั่งนี้จะทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพข้อมูลในจภาพไม่เปลี่ยน

3. ENTRY MODE SET

Entry mode set

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	0	0	0	I/D	S

BIT I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM ADDRESS เพิ่มขึ้นหนึ่งหรือลดลงหนึ่งโดย

1 = เพิ่ม

0 = ลดลงหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT S : เป็นกำหนดแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัว CURSOR อยู่กับที่ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะ อยู่กับที่ตัว CURSOR จะถูกดันไปทางขวามือ

4. DISPLAY ON/OFF CONTROL

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	0	0	1	D	C	B

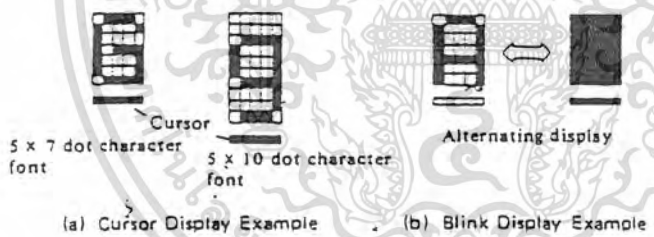
BIT D : เป็น BIT ให้เปิดปิดหน้าจอภาพโดยถ้า

D = 1 จะ ON และ

D = 0 จะ OFF

BIT C : จะให้แสดง CURSOR ให้ BIT C = 1 และถ้าไม่ต้องการแสดง CURSOR BIT C = 0 โดยตัว CURSOR จะอยู่ที่ LINE ที่ 8 ในแบบ 5X7 DOT และจะอยู่ LINE ที่ 11 ในแบบ 5X10 DOT

BIT B : เป็น BIT SET การกระพริบของ CURSOR โดย B = 1 การกระพริบ B = 0 ไม่มีการกระพริบ โดยมีระยะเวลาการกระพริบประมาณ 379.2 ms



5. CURSOR OR DISPLAY SHIFT

Cursor or display shift

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	0	1	S/C	R/L	*	*

* No effect

เป็นคำสั่งกำหนดให้ตำแหน่ง CURSOR หรือข้อมูลไปเกิดทางซ้ายหรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่าน โดย

S/C	R/L	
0	0	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปขวามือ 1 ตำแหน่ง
1	0	เป็นการค้นตัวอักษรที่เกิดไปทางซ้าย
1	1	เป็นการค้นตัวอักษรที่เกิดไปทางขวามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. FUNCTION SET

Function set

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	1	DL	N	F	*	*

* No effect

BIT DL : เป็นการ SET การติดต่อว่าจะให้เป็นแบบ 8 BIT หรือ 4 BIT
โดยถ้าต้องการติดต่อ 4 BIT DL = 0 และ 8 BIT DL = 1

N : เป็นการ SET บรรทัดการแสดงผล N = 0 แสดง 1 บรรทัด
N = 1 แสดง 2 บรรทัด ในกรณีมากกว่า 2 บรรทัด ก็ให้ SET N = 1

F : เป็นการ SET ขนาด DOT การแสดงผล 5X7 หรือ 5X10 โดย
F = 0 เป็นแบบ 5X7 และ F = 1 เป็นแบบ 5X10

N	F	No. of display lines	Character font	Duty factor	Remarks
0	0	1	5 x 7 dots	1/8	
0	1	1	5 x 10 dots	1/11	
1	*	2	5 x 7 dots	1/16	Cannot display 2 lines with 5 x 10 dot character font.

* No effect

7. SET CG RAM ADDRESS

Set CG RAM address

	RS	R/W	DB_7						DB_0
Code	0	0	0	1	A	A	A	A	A	A

← Higher Order Bits — Lower Order Bits →

ใน HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือ DISPLAY DATA RAM (DD RAM) จำนวน 80X8 BIT และ CHARACTER GENERATOR ROM CG RAM จำนวน 512 BIT และ 7200 BIT คำสั่งนี้จะเป็นการ SET ADDRESS ใน CG RAM โดยต้องทำการ SET ADDRESS ก่อนเขียนหรืออ่านข้อมูลจาก CG RAM ด้วย

8. SET DD RAM ADDRESS

Set DD RAM address

	RS	R/W	DB_7						DB_0
Code	0	0	1	A	A	A	A	A	A	A

← Higher Order Bits — Lower Order Bits →

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่ง SET ค่า ADDRESS ใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM(DD RAM คือ ส่วนที่จะแสดงผลหน้าจอ LCD) โดยจำนวน ADDRESS ที่จะเกิดขึ้นบนจอ LCD จะอยู่กับ SET ค่า N ด้วย

N = 0 (1 บรรทัด) ADDRESS จะอยู่ 00H-4FH

N = 1 (2 บรรทัด) ADDRESS จะอยู่ 00H-27H สำหรับบรรทัดที่ 1 และ 40H-67H สำหรับ บรรทัดที่ 2

แบบการจัด ADDRESS ของ DD RAM หน้าจอ LCD แบบ 16 ตัวอักษร 1 บรรทัด ,16 ตัวอักษร 2 บรรทัด ,16 ตัวอักษร 4 บรรทัด , 20 ตัวอักษร 1 บรรทัด, 20 ตัวอักษร 2 บรรทัด และ 40 ตัวอักษร 2 บรรทัด

16 ตัวอักษร 1 บรรทัด

00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

16 ตัวอักษร 2 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

16 ตัวอักษร 4 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

20 ตัวอักษร 1 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13

20 ตัวอักษร 2 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. READ BUSY FLAG AND ADDRESS

Read busy flag & address

	RS	R/W	DB_7							DB_0
Code	0	1	BF	A	A	A	A	A	A	A

← Higher Order Bits Lower Order Bits →

เป็นคำสั่งอ่านค่า BUSY FLAG ซึ่งจะเป็นตัวบอกว่าตัว HD44780 นี้อยู่ในขบวนการทำงานภายในอยู่หรืออยู่ในสภาพพร้อมจะรับข้อมูล โดย

BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล ADDRESS ของ CG RAM หรือ DD RAM ด้วย

10. WRITE DATA TO CG หรือ DD RAM

Write data to CG or DD RAM

	RS	R/W	DB_7							DB_0
Code	1	0	D	D	D	D	D	D	D	D

← Higher Order Bits Lower Order Bits →

เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อเขียนข้อมูลและ ADDRESS จะเพิ่มหรือลดโดยอัตโนมัติตามคำสั่งที่ SET ใน ENTRY MODE ข้อกำหนดที่จะรู้ว่าเป็นการเขียนข้อมูลของ CG RAM หรือ DD RAM ทำได้โดยการ SET ADDRESS ของ CG RAM หรือ DD RAM ขึ้นมาก่อนจะเขียนข้มูล

11. READ DATA FROM CG OR DD RAM

Read data from CG or DD RAM

	RS	R/W	DB_7							DB_0
Code	1	1	D	D	D	D	D	D	D	D

← Higher Order Bits Lower Order Bits →

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

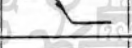
ขาต่างๆในการต่อใช้งาน HD44780

1. RS (REGISTOR SELECTION) จะเป็นขาเลือก REGISTOR ภายในซึ่งมี อยู่ 2 ตัวคือ **INSTRUCTION REGISTOR (IR)** และ **DATA REGISTOR (DR)** โดยถ้าเป็น 1 จะเป็นการเลือก DATA และถ้าเป็น 0 จะเป็นการเลือก INSTRUCTION

2. R/W (READ/WRITE) เป็นตัวเลือกว่าจะเขียนหรือจะอ่านข้อมูลจากตัว IC โดยอ่า นข้อมูล = 1, เขียนข้อมูล = 0

3. E (ENA BLE SIGNAL) เป็นขากำหนดสภาพการรับเขียนอ่านข้อมูล

Register selection

RS	R/W	E	Operation
0	0		IR write as internal operation (Display clear, etc.)
0	1		Read busy flag (DB_7) and address counter (DB_0 – DB_6)
1	0		DR write as internal operation (DR to DD or CG RAM)
1	1		DR read as internal operation (DD or CG RAM to DR)

4. DB0-DB7 เป็นขารับส่งข้อมูลจากตัว IC

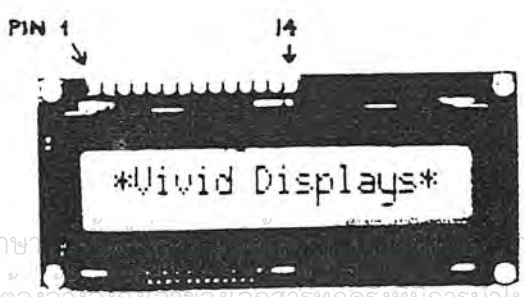
5. VDD ไฟเลี้ยงตัววงจร +5V

6. VSS เป็นขา GND

7. VO เป็นขารับ VOLTAGE ในการขับ LCD ให้สว่างหรือมืด

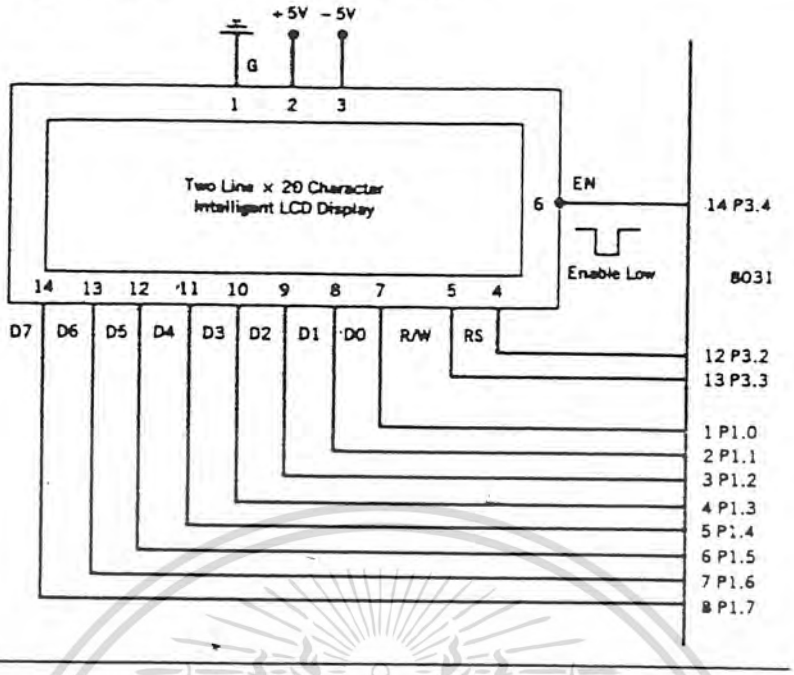
PIN CONNECTION

Pin No	Symbol	Level	Function
1	V_{SS}	—	0V
2	V_{DD}	—	+5V
3	Vo	—	—
4	RS	H/L	L: Instruction code input H: Data input
5	R/W	H/L	H: Data read (LCD module → MPU) L: Data write (LCD module ← MPU)
6	E	H, H→L	Enable signal
7	DB0	H/L	Data bus line Note (1), (2)
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่อนุญาตให้นำไปใช้เพื่อการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Intelligent LCD Circuit for "Lcdisp" Program




```
ADDRESS   MNEMONIC        COMMENT
            org 0000h
lcdisp:     clr p3.2        ;select the command register
            clr p3.3        ;select write level
            mov a,#3fh      ;command 8 bits/char. 2 rows, 5 x 10
            acall strobe    ;strobe command to display
            mov a,#0eh      ;command screen and cursor on, no blink
            acall strobe
            mov a,#06h      ;command cursor right as data displayed
            acall strobe
            mov a,#01h      ;clear all and home cursor
            acall strobe
            setb p3.2       ;select display data RAM register
            mov a,#'h'      ;say "hello"
            acall strobe
            mov a,#'e'
            acall strobe
            mov a,#'l'
            acall strobe
            acall strobe
            mov a,#'o'
            acall strobe
here:       sjmp here       ;message sent
;
;the subroutine "strobe" is used to check for a display busy
;condition, and pulse P3.3 high-low-high to enable the display
;write or read
;
strobe:     mov p1,#0ffh    ;configure port 1 as an input
            setb p3.3       ;set read level
wait:       setb p3.4       ;generate read strobe
            clr p3.4        ;enable the display
            jb p1.7,wait    ;check for busy when BF = 1
            setb p3.4       ;end of read strobe
            clr p3.3        ;write character to display
            setb p3.2       ;choose data RAM
            mov p1,a        ;character to port 1
            clr p3.4        ;generate write strobe
            setb p3.4
            clr p3.2        ;return with display as before call
            ret
            end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งห้ามมิให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with 80C51 and 80C52 Products
- 8 Kbytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
  - Data Retention: 10 Years
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-Bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

Description

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8 Kbytes 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 80C51 and 80C52 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 AT89C52 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

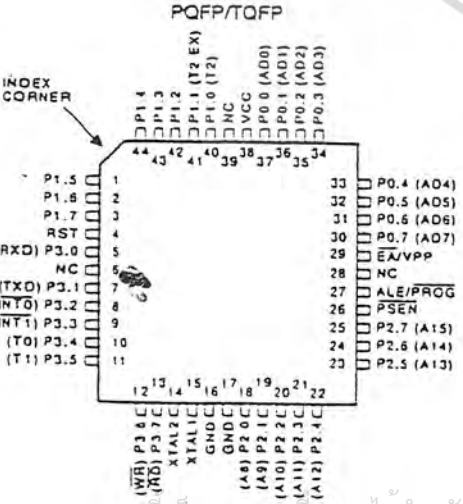
The AT89C52 provides the following standard features: 8 Kbytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is

continued

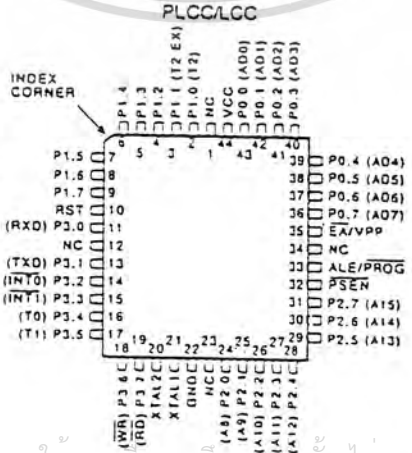
Pin Configurations

PDIP/Cerdip

PQFP/TQFP



PLCC/LCC

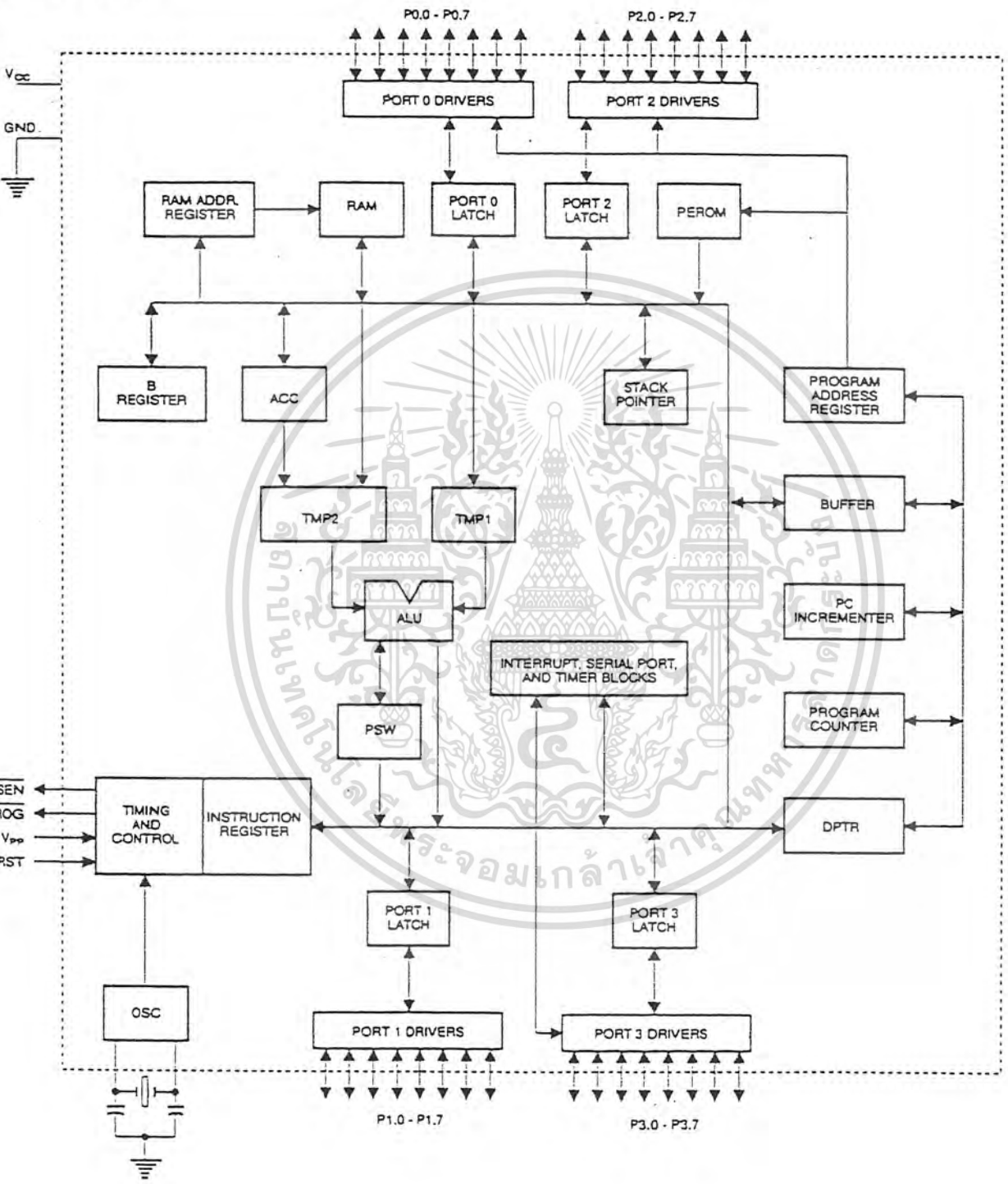


8-Bit Microcontroller with 8 Kbytes Flash

AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ck Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Description (Continued)

esigned with static logic for operation down to zero frequency nd supports two software selectable power saving modes. The dle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. he Power Down Mode saves the RAM contents but freezes the scillator, disabling all other chip functions until the next hardware reset.

Pin Description

CC

upply voltage.

ND

round.

ort 0

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

ort 0 can also be configured to be the multiplexed low-order ddress/data bus during accesses to external program and data emory. In this mode, P0 has internal pullups.

ort 0 also receives the code bytes during Flash programming nd outputs the code bytes during program verification. Exterl pullups are required during program verification.

ort 1

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

addition, P1.0 and P1.1 can be configured to be the mer/counter 2 external count input (P1.0/T2) and the mer/counter 2 trigger input (P1.1/T2EX), respectively, as own in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

ort 1 also receives the low-order address bytes during Flash ogramming and program verification.

ort 2

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

ort 2 emits the high-order address byte during fetches from exmal program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 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 bidirectional 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 shown in the following table.

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 programming 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 is an 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 microcrontroller is in external execution mode.

$\overline{PSEN}$

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

When the AT89C52 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{EA}$/V_{PP}

External Access Enable. $\overline{EA}$ 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, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ 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 when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode. *(continued)*

Table 1. AT89C52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP X0000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 00000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			PCON 00000000	87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

e 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

Bit	7	6	5	4	3	2	1	0
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

ecial Function Registers (Continued)

rupt Registers The individual interrupt enable bits are in E register. Two priorities can be set for each of the six interrupt sources in the IP register.

ta Memory

AT89C52 implements 256 bytes of on-chip RAM. The r 128 bytes occupy a parallel address space to the Special tion Registers. That means the upper 128 bytes have the addresses as the SFR space but are physically separate SFR space.

n an instruction accesses an internal location above address the address mode used in the instruction specifies whether CPU accesses the upper 128 bytes of RAM or the SFR e. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timer 0 and 1

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Table 3. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-Bit Auto-Reload
0	1	1	16-Bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-Reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

(continued)

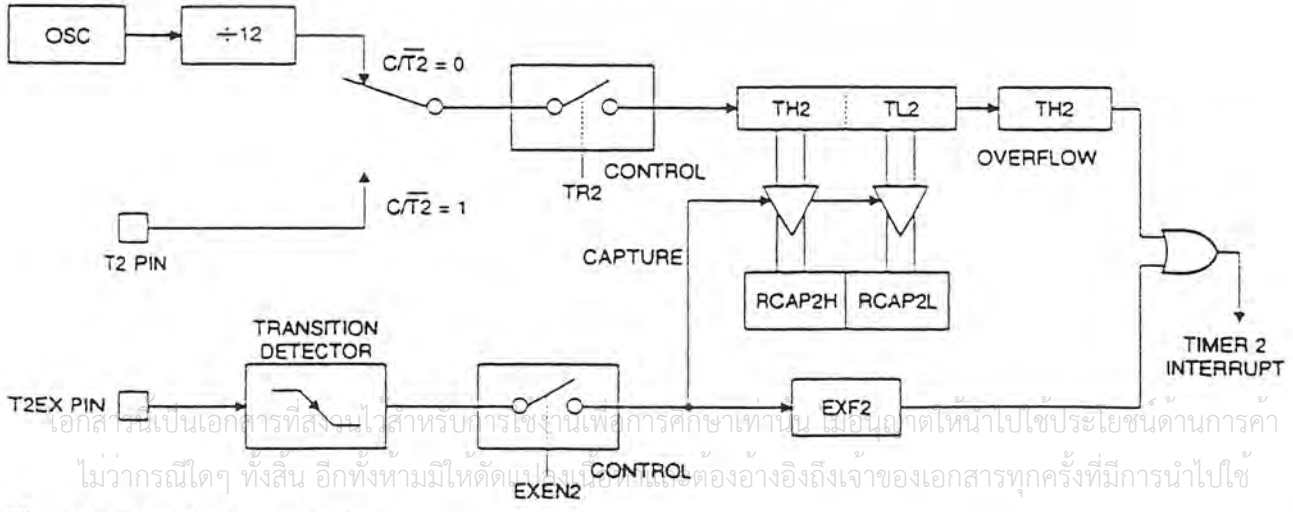


Figure 1. Timer 2 in Capture Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Auto-Reload (Up or Down Counter) (Continued)

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

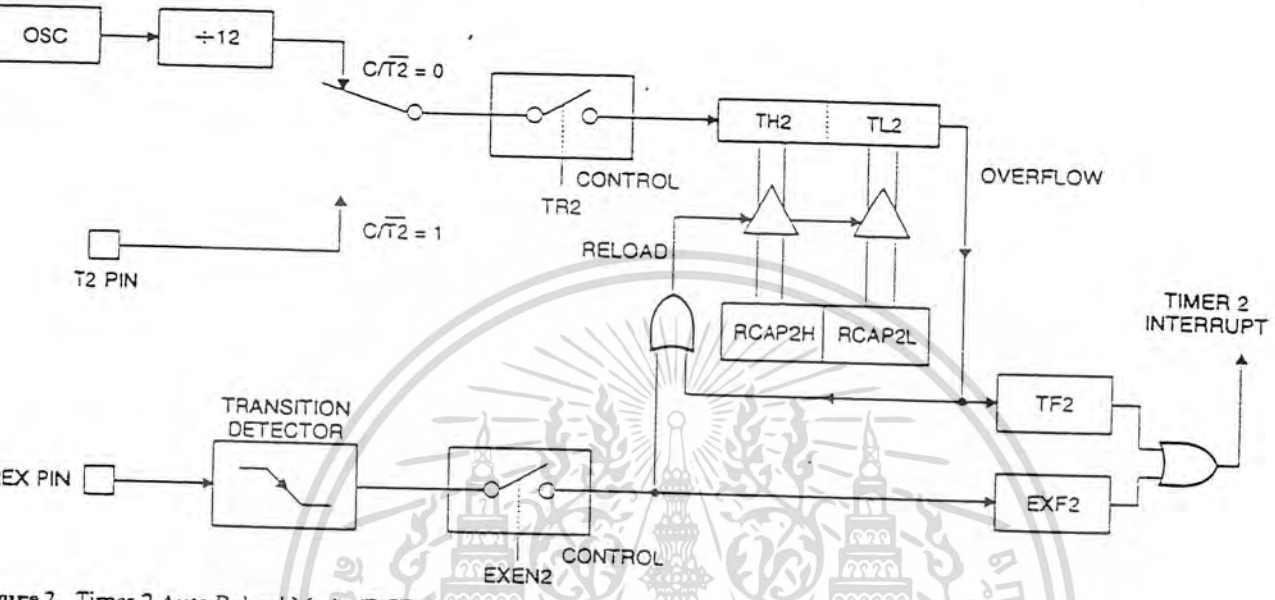


Figure 2. Timer 2 Auto Reload Mode (DCEN = 0)

Table 4. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H

Reset Value = XXXX XX00B

Not Bit Addressable

	—	—	—	—	—	—	T20E	DCEN
Bit	7	6	5	4	3	2	1	0

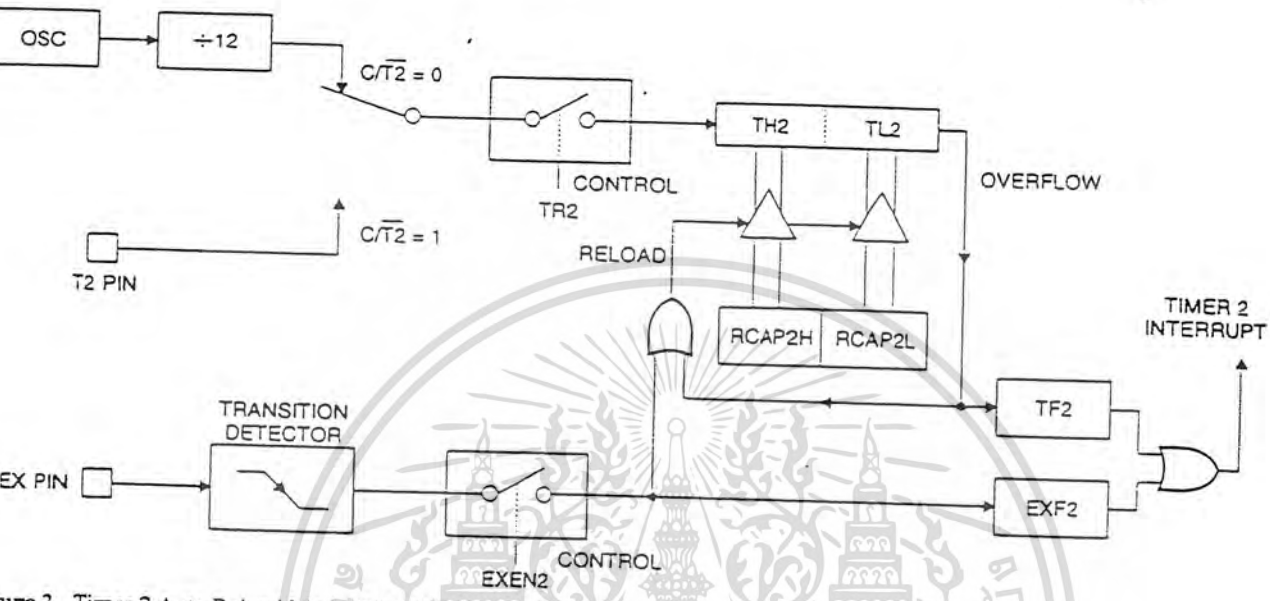
Symbol	Function
—	Not implemented, reserved for future use.
T20E	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Auto-Reload (Up or Down Counter) (Continued)

logic 0 at T2EX makes Timer 2 count down. The timer under-ows when TH2 and TL2 equal the values stored in RCAP2H nd RCAP2L. The underflow sets the TF2 bit and causes FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.



ure 2. Timer 2 Auto Reload Mode (DCEN = 0)

ble 4. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H

Reset Value = XXXX XX00B

Not Bit Addressable

	—	—	—	—	—	—	T20E	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
—	Not implemented, reserved for future use.
T20E	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

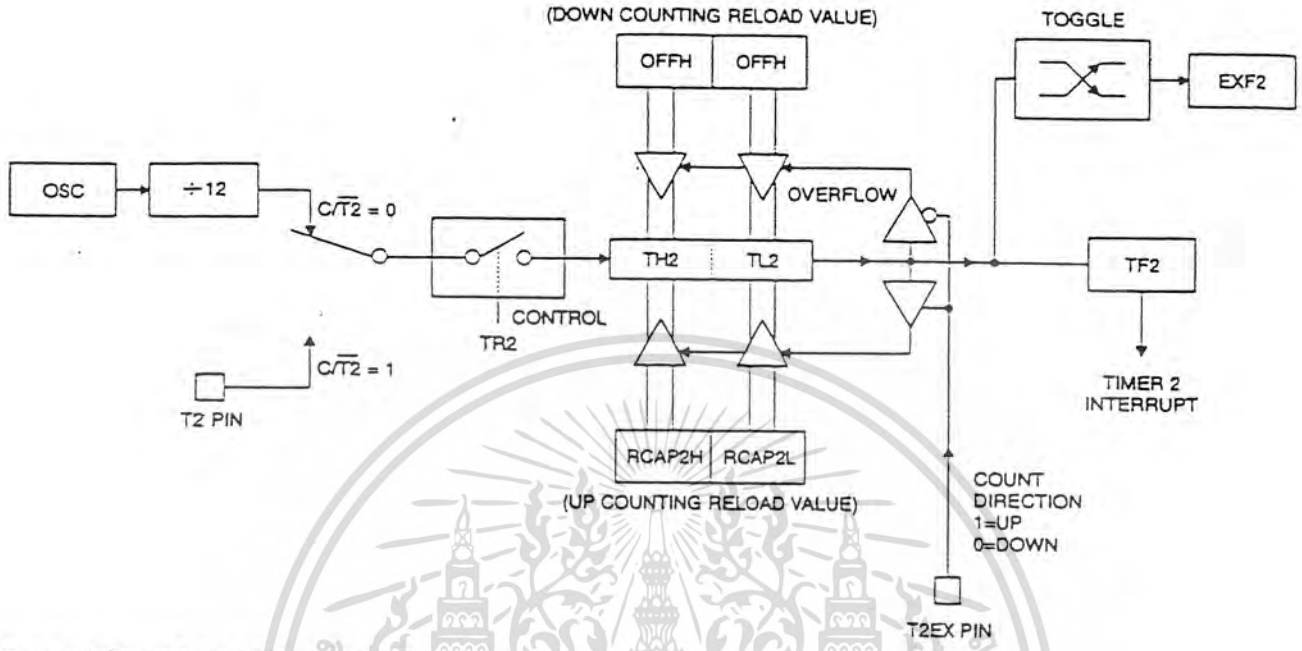


Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

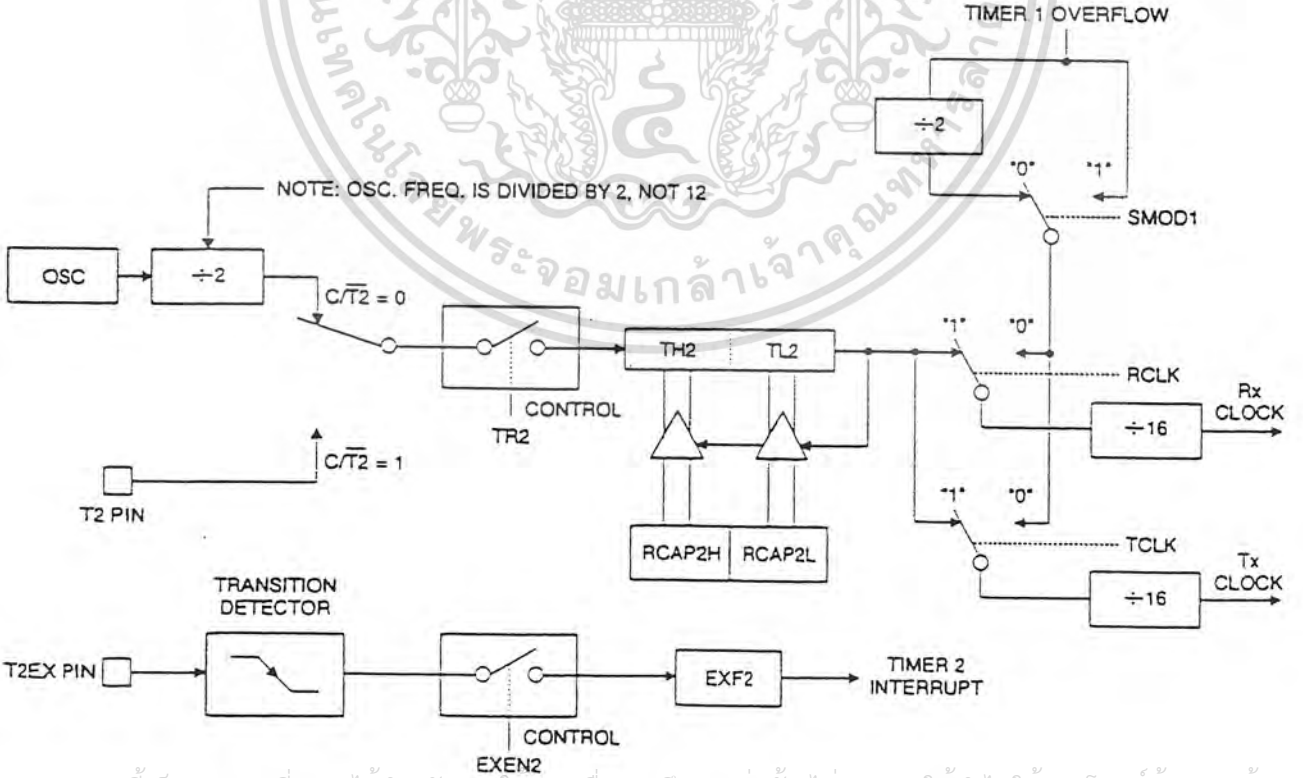


Figure 4. Timer 2 in Baud Rate Generator Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

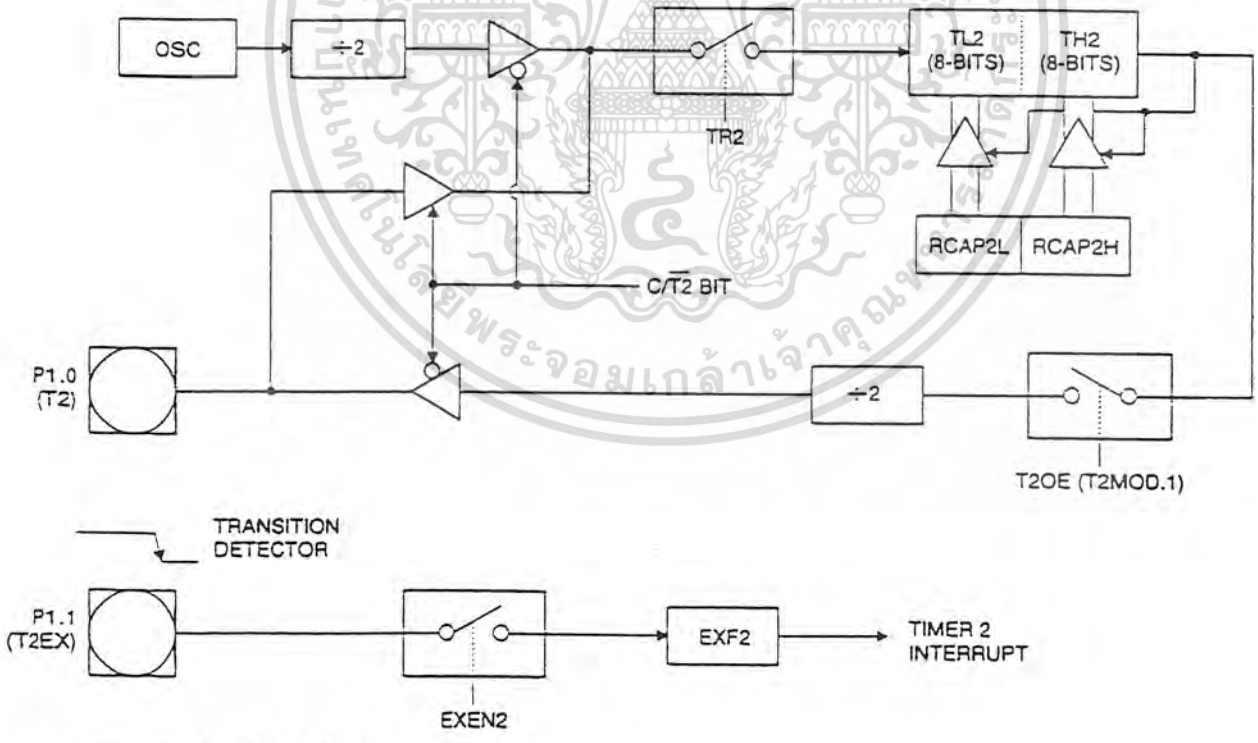


Figure 5. Timer 2 in Clock-Out Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rogrammable Clock Out

50% duty cycle clock can be programmed to come out on 1.0, as shown in Figure 5. This pin, besides being a regular I/O n, has two alternate functions. It can be programmed to input e external clock for Timer/Counter 2 or to output a 50% duty ycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating equency.

o configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be t. Bit TR2 (T2CON.2) starts and stops the timer.

he clock-out frequency depends on the oscillator frequency nd the reload value of Timer 2 capture registers (RCAP2H, CAP2L), as shown in the following equation.

$$\text{lock–Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

the clock-out mode, Timer 2 roll-overs will not generate an nterrupt. This behavior is similar to when Timer 2 is used as a aud-rate generator. It is possible to use Timer 2 as a baud-rate enerator and a clock generator simultaneously. Note, however, hat the baud-rate and clock-out frequencies cannot be deter-ined independently from one another since they both use CAP2H and RCAP2L.

UART

he UART in the AT89C52 operates the same way as the ART in the AT89C51.

nterrupts

he AT89C52 has a total of six interrupt vectors: two external nterrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, nd 2), and the serial port interrupt. These interrupts are all own in Figure 6.

ach of these interrupt sources can be individually enabled or isabled by setting or clearing a bit in Special Function Register E. IE also contains a global disable bit, EA, which disables all nterrupts at once.

ote that Table 5 shows that bit position IE.6 is unimplemented. the AT89C51, bit position IE.5 is also unimplemented. User oftware should not write 1s to these bit positions, since they ay be used in future AT89 products.

imer 2 interrupt is generated by the logical OR of bits TF2 and XF2 in register T2CON. Neither of these flags is cleared by ardware when the service routine is vectored to. In fact, the ervice routine may have to determine whether it was TF2 or XF2 that generated the interrupt, and that bit will have to be leared in software.

he Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of he cycle in which the timers overflow. The values are then olled by the circuitry in the next cycle. However, the Timer 2 ag, TF2, is set at S2P2 and is polled in the same cycle in which he timer overflows.

Table 5. Interrupt Enable (IE) Register

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

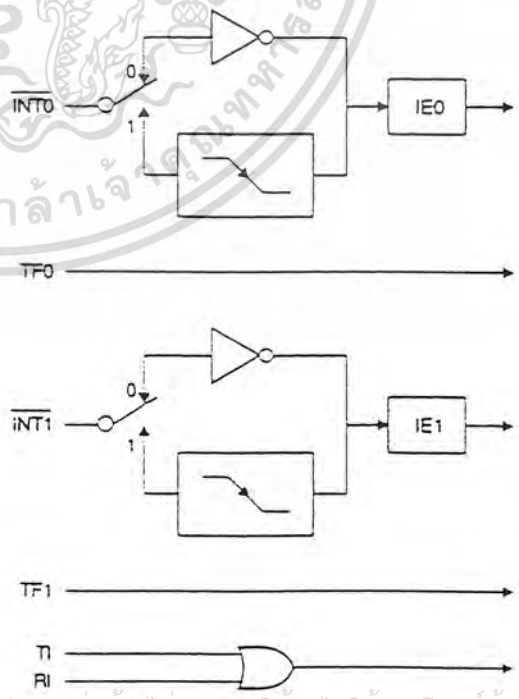


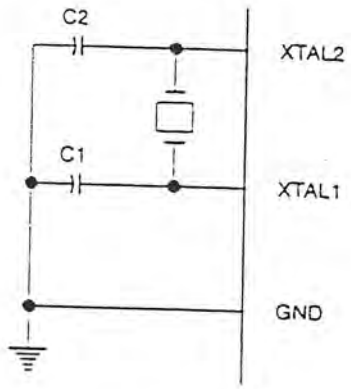
Figure 6. Interrupt Sources

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. 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 8. 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.

Figure 7. Oscillator Connections



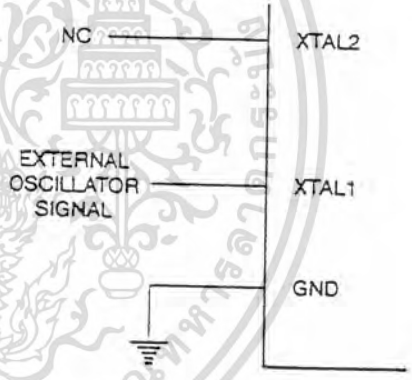
Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

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.

Note that when idle mode is terminated by a hardware 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 mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Figure 8. 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.

Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง เผยแพร่ หรือทำซ้ำ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Memory Lock Bits

The AT89C52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

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. The latched value of $\overline{EA}$ must agree 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, EA is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

Programming the Flash

The AT89C52 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 AT89C52 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C52 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} = 12 V	V_{PP} = 5 V
Top-Side Mark	AT89C52 xxxx yyww	AT89C52 xxxx-5 yyww
Signature	(030H)=1EH (031H)=52H (032H)=FFH	(030H)=1EH (031H)=52H (032H)=05H

The AT89C52 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 AT89C52, the address, data and control signals should be set up according to the Flash programming mode table and Figures 9 and 10. To program the AT89C52, 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 12 V 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.

Data Polling: The AT89C52 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 data on P0.7. Once the write cycle has been completed, true data is 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 $\overline{BUSY}$. 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 1s. The chip erase operation must be executed before the code memory can be reprogrammed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,

ห้องสมุดกลาง ... เพื่อการศึกษา ... ไม่แสวงหากำไร ... ห้ามดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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) = 52H indicates 89C52
(032H) = FFH indicates 12 V programming
(032H) = 05H indicates 5 V 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	$\overline{PSEN}$	ALE/$\overline{PROG}$	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L	⎍	H/12V(1)	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	⎍(2)	H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.

2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

Figure 9. Programming the Flash Memory

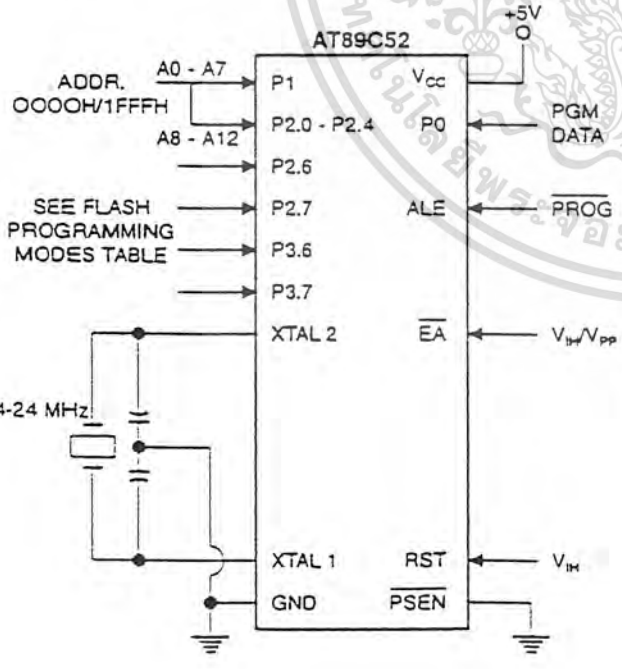
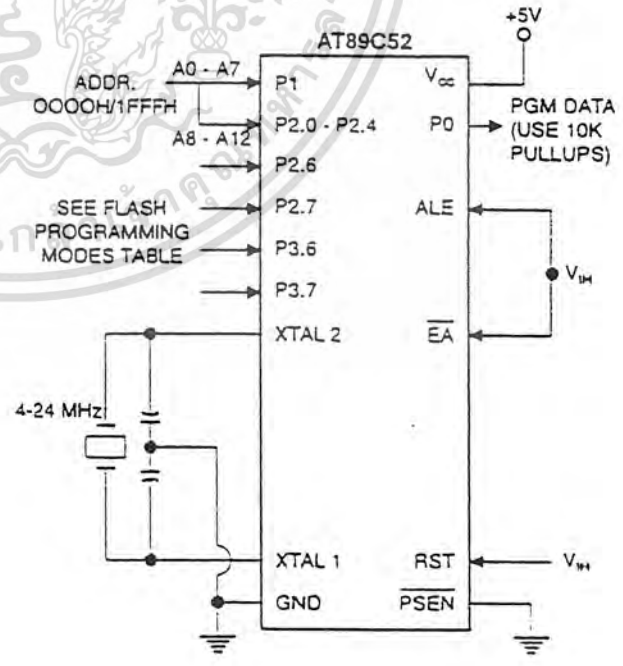


Figure 10. Verifying the Flash Memory



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง เผยแพร่ หรือสำเนาต่อ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Characteristics

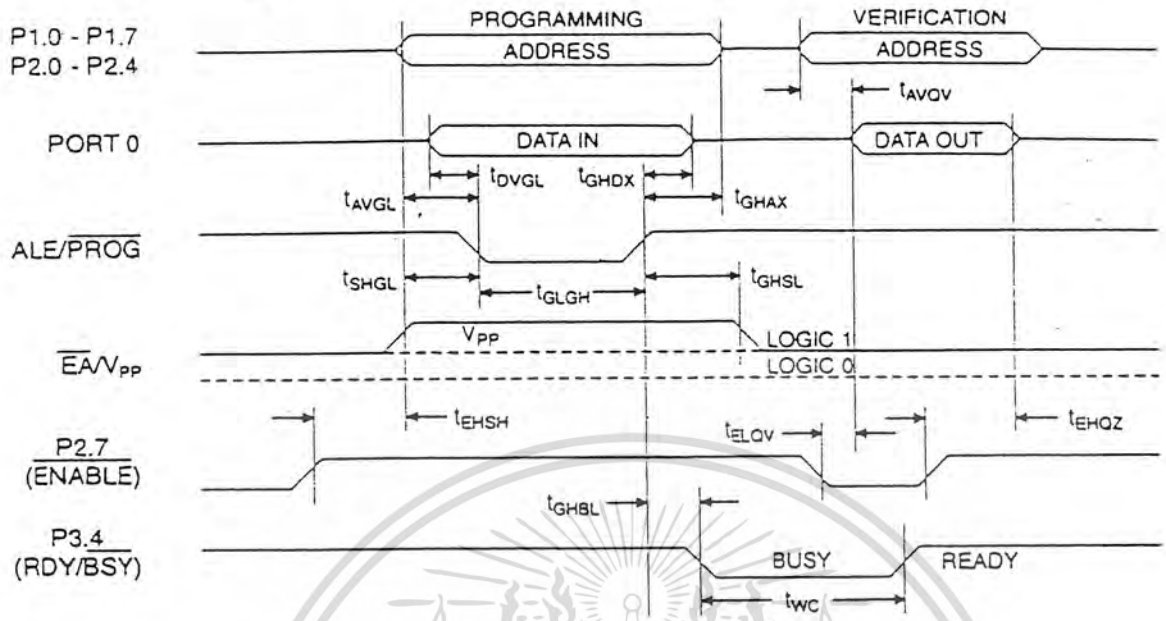
T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

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	4	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_{EHQV}	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

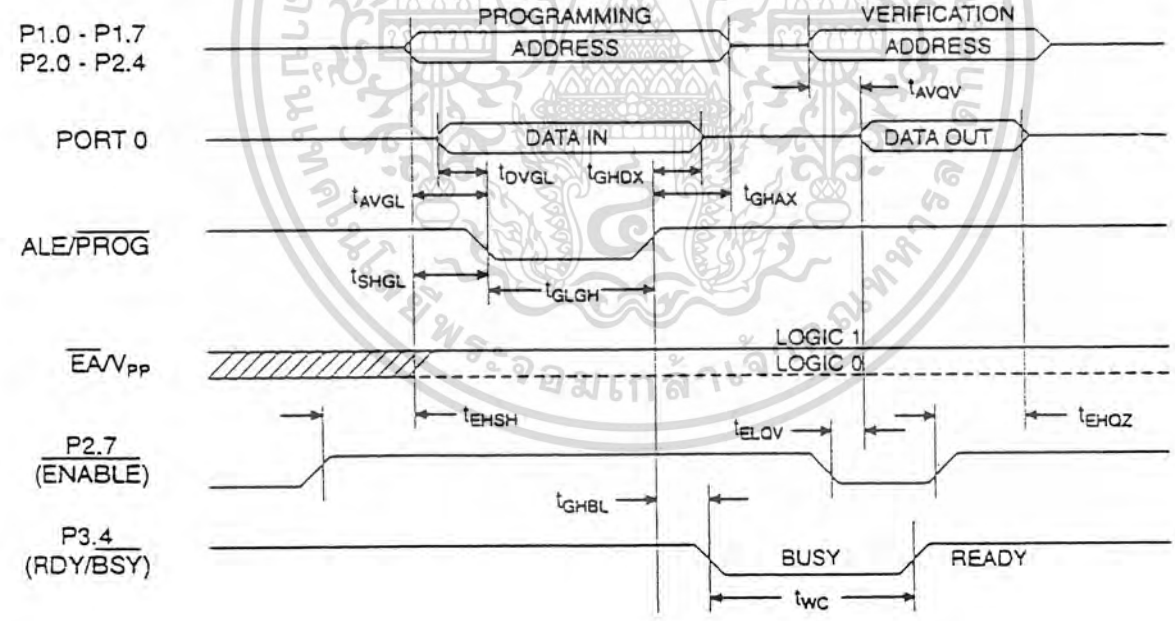
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

lash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
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.

D.C. Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 5.0 V ± 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} = 5 V ± 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} = 5 V ± 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} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2 V		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown 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		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode[2]	V_{CC} = 6 V		100	µA
		V_{CC} = 3 V		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 IOL for all output pins:71 mA
If IOL exceeds the test condition, VOL 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 2 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.C. Characteristics

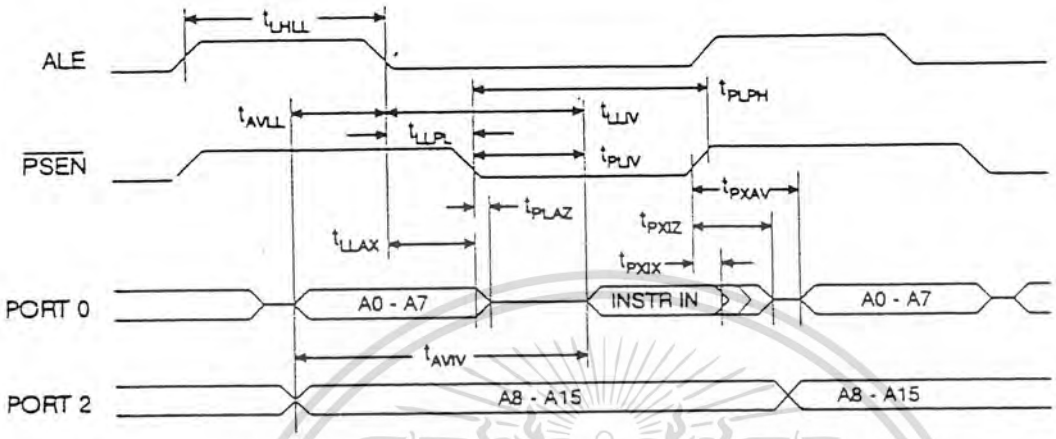
ıder operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all ıer outputs = 80 pF.

xternal Program and Data Memory Characteristics

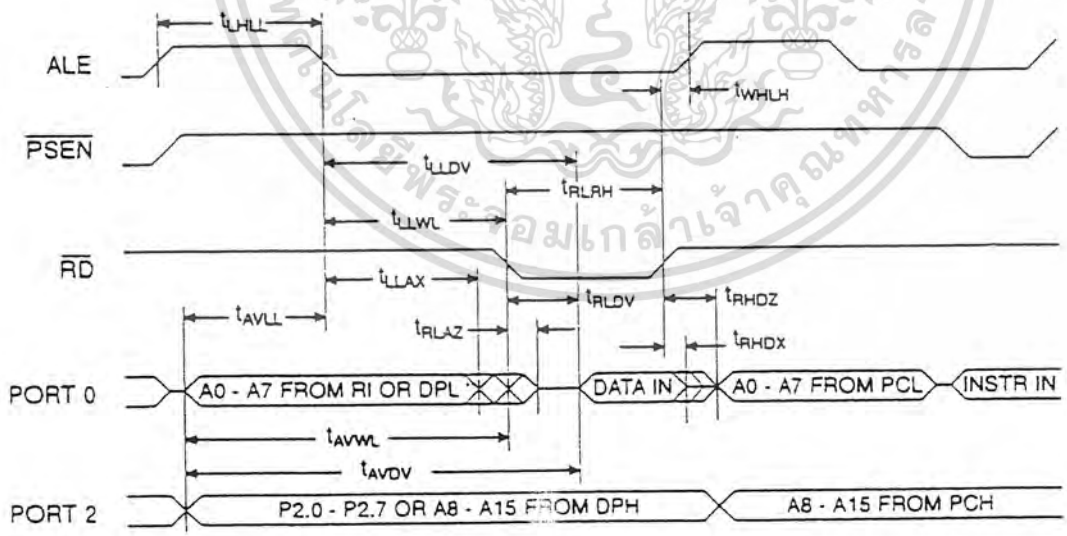
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
/tCLCL	Oscillator Frequency			0	24	MHz
LHLL	ALE Pulse Width	127		2tCLCL-40		ns
AVLL	Address Valid to ALE Low	28		tCLCL-13		ns
LLAX	Address Hold After ALE Low	48		tCLCL-20		ns
LLIV	ALE Low to Valid Instruction In		233		4tCLCL-65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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

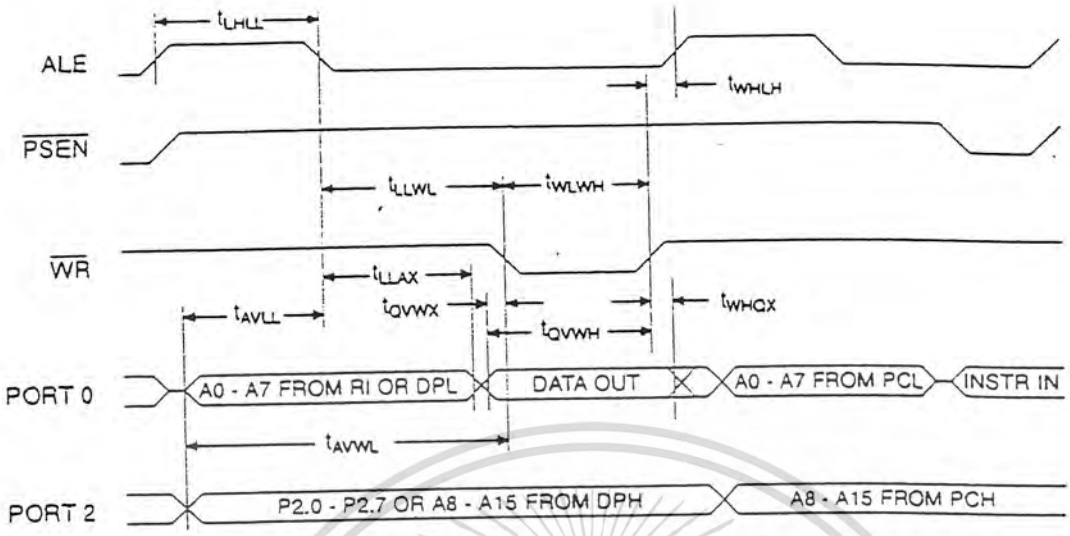


External Data Memory Read Cycle

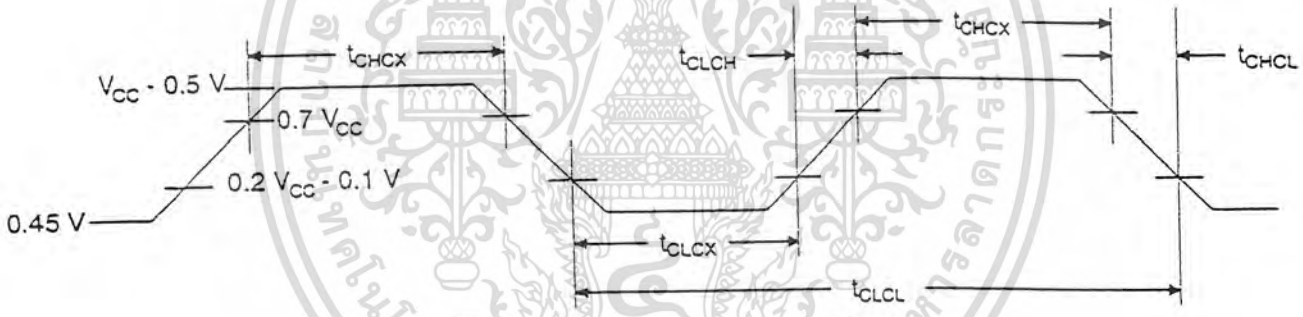


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory 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

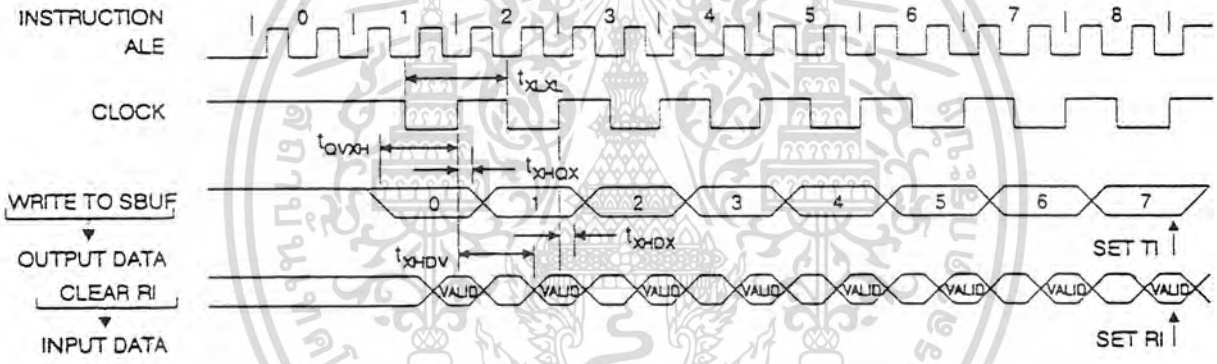
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

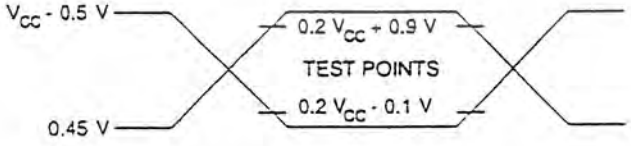
The values in this table are valid for V_{CC} = 5.0 V ± 20% and Load Capacitance = 80 pF.

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		$2t_{CLCL}$-33		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

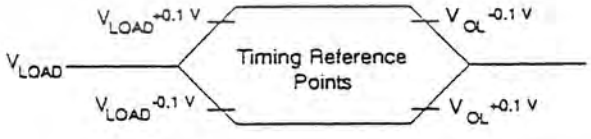


AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V 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 whe a 100-mV change from load voltage occurs. A port pin begins to float when a 100-mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

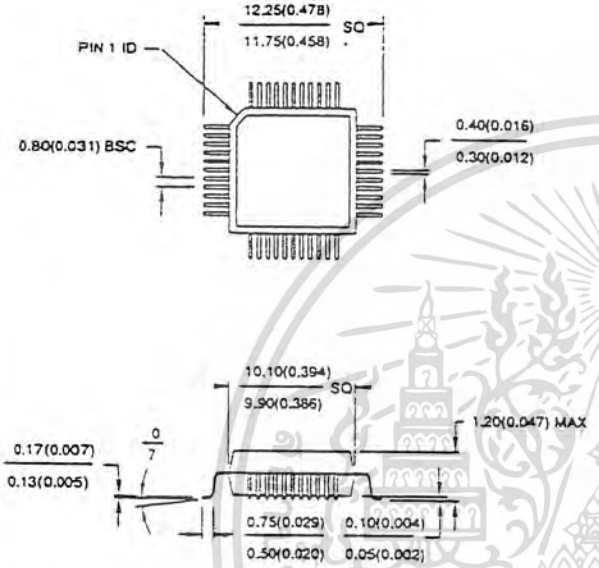
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5 V ± 10%	AT89C52-12DM AT89C52-12LM	40D6 44L	Military (-55°C to 125°C)
		AT89C52-12DM/883 AT89C52-12LM/883	40D6 44L	Military/883C Class B, Fully Compliant (-55°C to 125°C)
16	5 V ± 20%	AT89C52-16AA AT89C52-16JA AT89C52-16PA AT89C52-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
20	5 V ± 20%	AT89C52-20AC AT89C52-20JC AT89C52-20PC AT89C52-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C52-20AI AT89C52-20JI AT89C52-20PI AT89C52-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
24	5 V ± 20%	AT89C52-24AC AT89C52-24JC AT89C52-24PC AT89C52-24QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C52-24AI AT89C52-24JI AT89C52-24PI AT89C52-24QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
40D6	40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
44L	44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

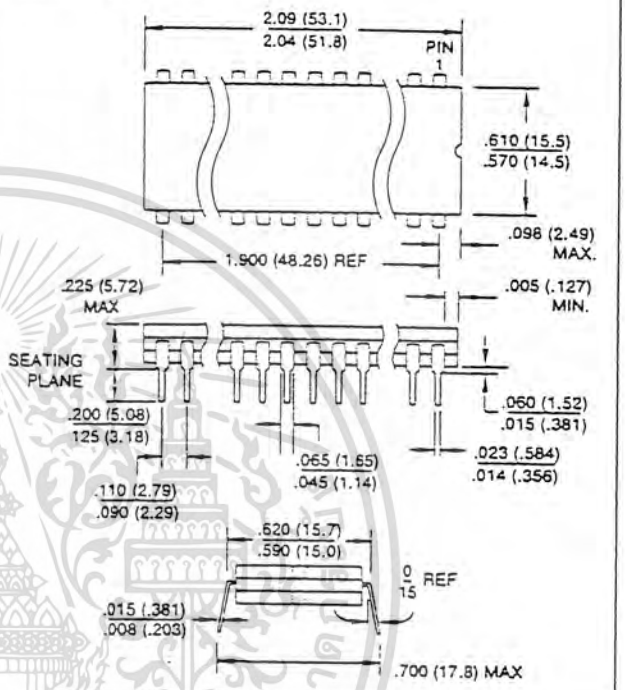
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

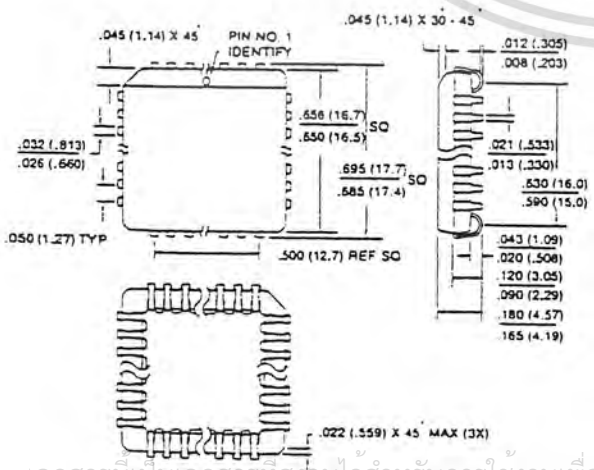
44A, 44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)



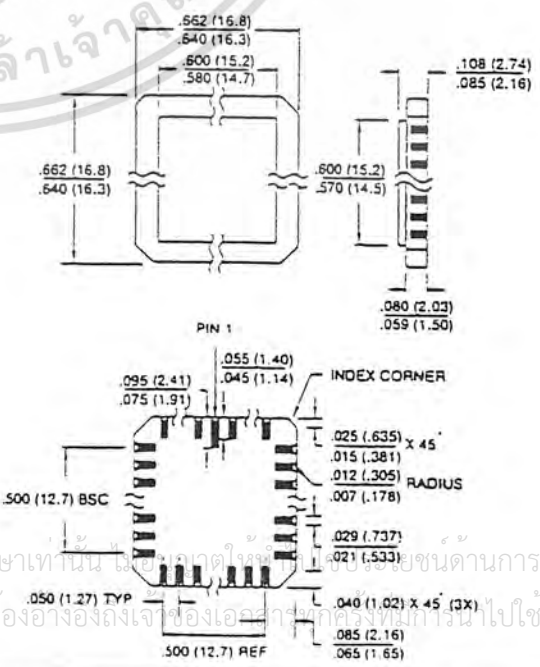
40D6, 40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
Dimensions in Inches and (Millimeters)
MIL-M-38510 D-5 CONFIG 1



44J, 44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC OUTLINE MO-047 AC



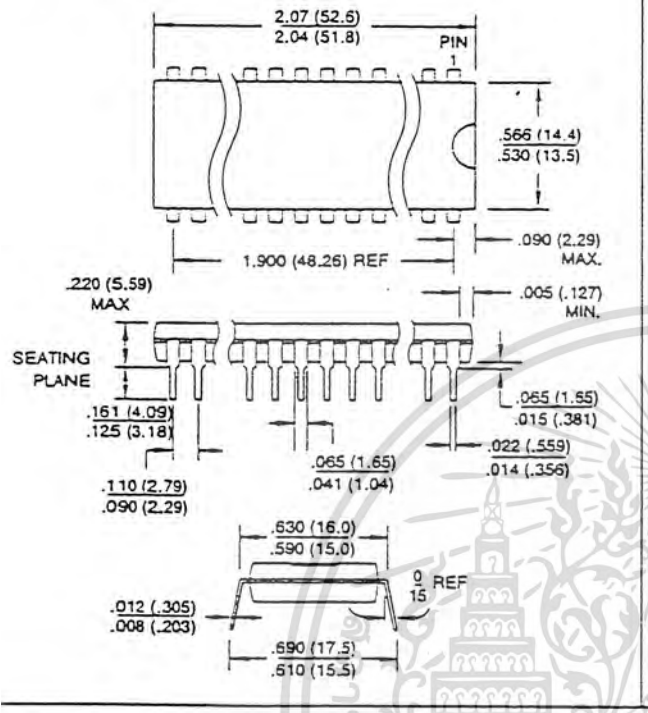
44L, 44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
Dimensions in Inches and (Millimeters)



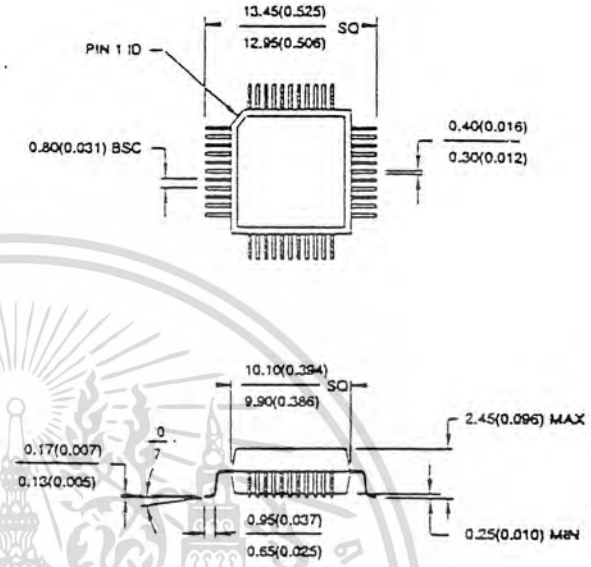
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

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



44Q, 44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)
Dimensions in Millimeters and (Inches)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง เนื้อหา ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54373 / 74373 Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power T				
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Pa			
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C			
T.I.	SN54S373	J (20)									SN54LS373	J (20)													
	SN74S373	J (20), N									SN74LS373	J (20), N													
FAIRCHILD																									
MOTOROLA																									
N. S. C.																									
PHILIPS																									
SIGNETICS																									
SIEMENS																									
FUJITSU																									
HITACHI																									
MITSUBISHI																									
NEC																									
TOSHIBA																									

Electrical Characteristics SN54LS373/SN74LS373

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

recommend operating conditions

		SN54LS373			SN74LS373			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}				−1			−2.6	mA
High-level output voltage, V_{OH}				5.5			5.5	V
Pulse width, t_w	Clock enable high	15			15			ns
	Clock enable high	15			15			
Setup time, t_{SUTUP}		0			0			ns
Hold time, t_{HOLD}		10			10			ns
Operating free-air temperature, T_A		−55		125	0		70	°C

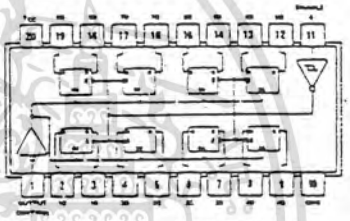
electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS‡		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = −18mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max, I_{OH} = MAX		2.4	3.1		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max	I_{OL} = 24mA		0.35	0.5	V
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX, V_{IH} = 2V, V_O = 2.7V				20	μA
I_{OZL}	Off-state output current, low-level voltage applied	V_{CC} = MAX, V_{IH} = 2V, V_O = 0.4V				−20	μA
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7V				0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7V				20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4V				−0.4	mA
I_{OS}	Short-circuit output current §	V_{CC} = MAX		−30		−130	mA
I_{CC}	Supply current	V_{CC} = MAX, Output control at 4.5V	LS373		24	40	mA

switching characteristics, V_{CC} = 5V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{max}							MHz
t_{PLH}	Data	Any Q	C_L = 45pF, R_L = 667Ω, See Notes 2 and 3		12	18	ns
t_{PHL}					12	18	
t_{PLH}	Clock or enable	Any Q			20	30	ns
t_{PHL}					18	30	
t_{PZH}	Output control	Any Q			15	25	ns
t_{PZL}					25	35	
t_{PHZ}	Output control	Any Q	C_L = 5pF, R_L = 667Ω, See Notes 2 and 3		12	20	ns
t_{PLZ}					15	25	

Pin Assignments (Top View)



SN54LS373 (J)
SN54S373 (J)

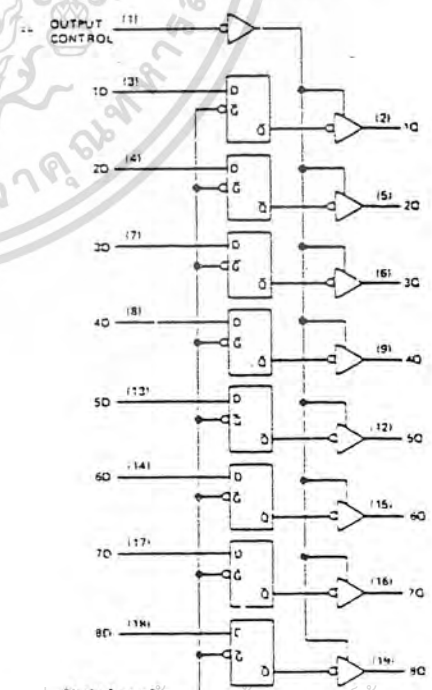
SN74LS373 (J, N)
SN74S373 (J, N)

LS373, S373 FUNCTION TABLE

OUTPUT CONTROL	ENABLE G	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q0
H	X	X	Z

LS373, S373 TRANSPARENT LATCHES

LS373, S373 TRANSPARENT LATCHES



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

245 74245 Octal Bus Transceivers with 3-state Outputs

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL			
	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF
											SN54LS245				W									
											SN74LS245				N									
ILD																								
OLA																								
S																								
ICS																								
NS																								
SU																								
HI																								
BISHI																								
BA																								

ctrical Characteristics SN54LS245/SN74LS245

absolute maximum ratings over operating free-air temperature range

oltage, VCC	7 V	Operating free-air temperature range	SN54LS	55°C	to 125°C
tage	7 V		SN74LS	0°C	to 70°C
		Storage temperature range		65°C	to 150°C

recommended operating conditions

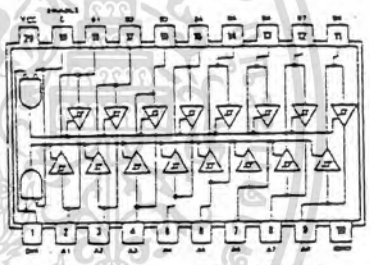
	SN54LS245			SN74LS245			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
oltage, VCC	4.5	5	5.5	4.75	5	5.25	V
h output current, IOH			-12			-15	mA
w output current, IOL			12			24	mA
g free-air temperature, TA	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS †		SN74LS245 MIN	TYP‡	MAX	UNIT
High-level input voltage			2			V
Low-level input voltage					0.8	V
Input clamp voltage	VCC = MIN,	II = -18mA			-1.5	V
Hysteresis (VT+ - VT-) A or B input	VCC = MIN		0.2	0.4		V
High-level output voltage	VCC = MIN, VIH = 2V, VIL = VIL max	IOH = -3mA	2.4	3.4		V
		IOH = MAX	2			
Low-level output voltage	VCC = MIN, VIH = 2V, VIL = VIL max	IOL = 12mA			0.4	V
		IOL = 24mA			0.5	
Off-state output current, high-level voltage applied	VCC = MAX, G at 2V	VO = 2.7 V			10	μA
Off-state output current, low-level voltage applied		VO = 0.4V			-200	
Input current at maximum input voltage	A or B	VCC = MAX,	VI = 5.5V			0.1	mA
	DIR or G		VI = 7V			0.1	
High-level input current	VCC = MAX,	VIH = 2.7V			20	μA
Low-level input current	VCC = MAX,	VIL = 0.4V			-0.2	mA
Short-circuit output current	VCC = MAX		-40		-225	mA
Supply current	Total, outputs high	VCC = MAX, Outputs open		48	70	mA
	Total, outputs low			62	90	
	Outputs at Hi-Z			64	95	

switching characteristics, VCC 5V, TA 25C

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Propagation delay time, low-to-high-level output	CL = 45pF, RL = 667Ω, See Note 2		8	12	ns
Propagation delay time, high-to-low-level output			8	12	ns
Output enable time to low level			27	40	ns
Output enable time to high level			25	40	ns
Output disable time from low level	CL = 5pF, RL = 667Ω, See Note 2		15	25	ns
Output disable time from high level			15	25	ns



SN54LS245 (J) SN74LS245 (J, N)

conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
typical values are at VCC = 5V, TA = 25°C.
more than one output should be shorted at a time, and duration of the short-short should not not exceed one second

เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ปรเมษฐ์ ประมยานันท์ และ ปิยะพงศ์ เผ่าวณิช , " คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51 " (หน้า 9-94 , 110-116) , บริษัทซีเอ็ดยูเคชัน จำกัด มหาชน , 2537
2. ไพบูลย์ บุปผา , " เทคนิคการใช้แอลซีดีโมดูล " , เซมิคอนดัคเตอร์อิเล็กทรอนิกส์ (หน้า 48-55 ฉบับที่ 166) ธันวาคม 2539
3. สุนทร วิทูสุรพจน์ , " การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 " , บริษัทซีเอ็ดยูเคชัน จำกัด มหาชน , 2537
4. ไกรวุฒิ โรจน์ประเสริฐสุด "ไมโครโปรเซสเซอร์2" , บริษัทซีเอ็ดยูเคชัน จำกัดมหาชน ,2539
5. Seth Bergmann , " Complier Design " , WM.C. Brown Publisher (pages 30-67)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้