



เครื่องเลือกสัญญาณภาพและเสียง
สำหรับระบบโทรทัศน์วงจรปิด

AV SWITCHER FOR CCTV

โดย

นายไกรวิทย์ เอก
นายนพพร ศิริกาญจนสุข

วัน เดือน ปี..... 14.ค.ค.2541
เลขทะเบียน..... 038899
เลขเรียกหนังสือ..... T.401๗๑.ก.๑๗๔.๑

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038899

หัวข้อปริญญาบัตร

เครื่องเลือกสัญญาณภาพและเสียงสำหรับระบบโทรทัศน์วงจรปิด

โดย

นายไกรวิทย์ เอกา

อาจารย์ที่ปรึกษา

นายนพพร ศิริกาญจนสุข

ภาควิชา

อาจารย์ประดิษฐ์ วัชรพิบูลย์

ปีการศึกษา

เทคนิคอุตสาหกรรม

2540

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

อนุมัติปริญญาบัตรฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร

คณะกรรมการการสอบปริญญาบัตร

-----ประธานกรรมการ

()

-----กรรมการ

()

-----กรรมการ

()

-----กรรมการ

()

-----กรรมการ

()

-----กรรมการ

()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการปริญญาโทฉบับนี้สามารถประสบความสำเร็จได้ด้วยดี ก็เพราะได้รับคำแนะนำปรึกษาที่ดี ตลอดจนเครื่องมือเครื่องใช้และอุปกรณ์ที่สำคัญต่างๆ รวมทั้งสถานที่ทำงานที่อบอุ่นจาก อาจารย์ประดิษฐ์ วัชรพิบูลย์ รวมทั้งคำติชมจากอาจารย์ประจำภาควิชาทุกท่าน , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ศูนย์นนทบุรี และแหล่งข้อมูลในการค้นคว้าข้อมูลต่างๆที่ไม่ได้กล่าวถึง คณะผู้จัดทำจึงขอขอบคุณในความกรุณาของทุกท่านมา ณ.ที่นี้ด้วย

คณะผู้จัดทำ

นายไกรวิทย์ เอกา

นายนพพร ศิริกาญจนสุข



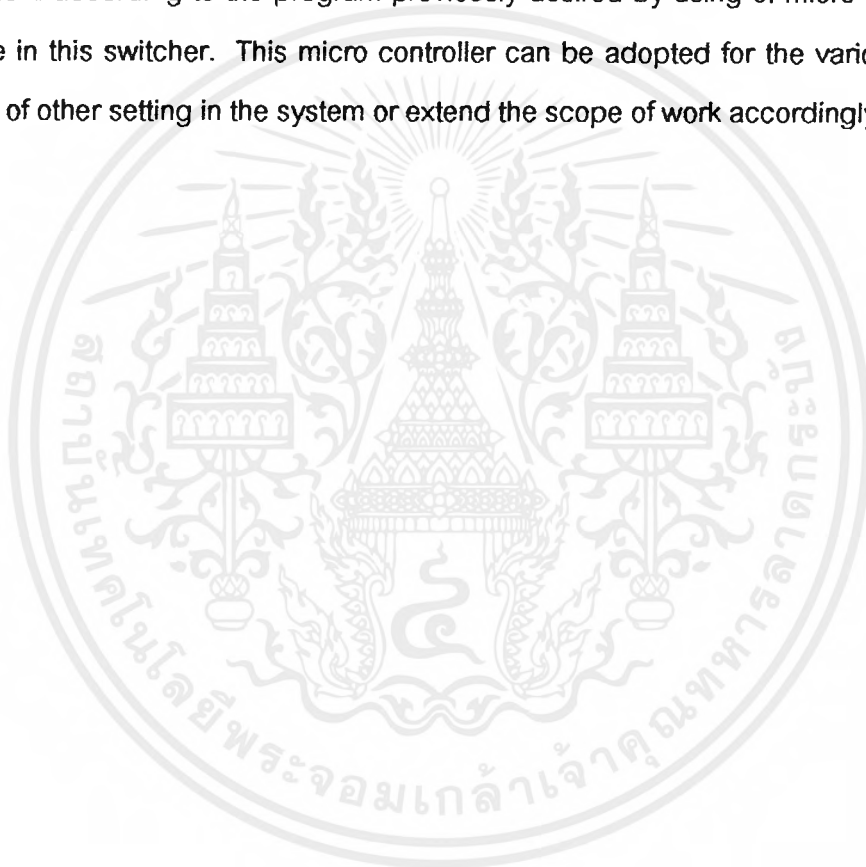
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ
เครื่องควบคุมสัญญาณภาพและเสียง
สำหรับระบบโทรทัศน์วงจรปิด
AV SWITCHER FOR CCTV

เครื่องควบคุมสัญญาณนี้ถูกออกแบบมาเพื่อควบคุมสัญญาณวิดีโอหรือสัญญาณภาพ ในระบบของโทรทัศน์วงจรปิด ซึ่งโดยทั่วไปแล้วในระบบโทรทัศน์วงจรปิดนั้น ต้องการมอนิเตอร์ 1 ตัวต่อกล้องวิดีโอ 1 ตัว ดังนั้นถ้าเป็นระบบที่ใหญ่แล้วจะมีราคาที่สูงมาก ถ้าหากว่าเราสามารถลดจำนวนของจอมอนิเตอร์ที่จะใช้ก็เท่ากับเป็นการลดค่าใช้จ่าย หรือลดต้นทุนของโทรทัศน์วงจรปิดได้ เครื่องควบคุมของสัญญาณวิดีโอนี้ เป็นอุปกรณ์ที่สามารถทำให้จอมอนิเตอร์ 1 ตัว สามารถที่จะเลือกแสดงผลของสัญญาณภาพใดๆจากกล้องวิดีโอได้หลาย ๆ ตัว ได้ โดยในโครงงานนี้ออกแบบมาให้รับกล้องวิดีโอได้ 4 ตัว แสดงผลออกที่จอมอนิเตอร์ 1 ตัว โดยที่ผู้ใช้งานสามารถตั้งโหมดการทำงานได้หลายแบบ โดยมีทั้งหมด MANUAL ผู้ใช้งานสามารถเลือกสัญญาณวิดีโอจากกล้องวิดีโอใดๆให้แสดงผลได้ตามต้องการและในโหมด AUTO SCAN เครื่องก็จะทำการ SCAN เลือกช่องสัญญาณเรียงลำดับอย่างอัตโนมัติตามโปรแกรมที่ได้ออกแบบไว้โดยอุปกรณ์ที่เป็นหัวใจของเครื่องควบคุมสัญญาณวิดีโอนี้ก็คือ ไมโครคอนโทรลเลอร์ซึ่งเป็นอุปกรณ์ที่สามารถประยุกต์ใช้งานได้มากมาย การใช้โปรแกรมในการควบคุมทำให้มีความยืดหยุ่นในการปรับเปลี่ยนระบบต่าง ๆ หรือขยายขอบเขตของงานได้อย่างสะดวก

ABSTRACT

The AV SWITCHER is designed to select a video signal from any TV camera in close circuit television system. In general, in close circuit television system one video camera requires one TV monitor. However, the cost of the system is proportion to its size. Thus, if the numbers of TV monitors can be reduced the cost of the close circuit the television system will be reduce as well. In this project this AV SWITCHER is designed to display the picture signal from 4 video cameras according to the need of the users. The users can set the mode of display, namely manual and auto scan mode according to their desire. In the auto scan mode, it is capable to scan in sequence from the first to the fourth camera according to the program previously desired by using of micro controller which is the key device in this switcher. This micro controller can be adopted for the various purposes and allow flexibility of other setting in the system or extend the scope of work accordingly



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1

บทนำ	1
ความสามารถของเครื่องเลือกสัญญาณภาพและเสียง	2

บทที่ 2

2.1 ระบบเลือกสัญญาณภาพโดยใช้สวิตช์	4
2.2 แผนผังแสดงภายในของเครื่องเลือกสัญญาณภาพ	4
การทำงานของเครื่อง	7
การทำงานในโหมด manual	7
การทำงานในโหมดสแกนอัตโนมัติ	7
ภาค VIDEO SELECTOR	8
ภาค AUDIO SELECTOR	11

บทที่ 3

จอแสดงผลแบบ LCD	17
3.1 รายละเอียดของคำสั่ง HD44780	17
3.2 ส่วนประกอบของโปรแกรม	23
3.3 การเขียนข้อมูล CHARACTER GENERATOR	23

บทที่ 4

ไมโครคอนโทรลเลอร์ตระกูล 8051	25
4.1 คุณลักษณะพื้นฐานของ 8051	25
4.2 โปรแกรมสั่งงานของไมโครคอนโทรลเลอร์	26
4.3 โปรแกรมแอสเซมบลี	27
4.4 ขั้นตอนของการพัฒนาโปรแกรม	28
4.5 โครงสร้างของโปรแกรมภาษาแอสเซมบลี	29
4.6 โครงสร้างภายในของ 8051	31
4.7 ชุดคำสั่งของ 8051	36

บทที่ 5

โปรแกรมสำหรับเครื่องเลือกสัญญาณ

50

บทที่ 6

ผลการทดลอง

59

ขั้นตอนการใช้เครื่อง

61

รายการอุปกรณ์ของภาคต่างๆ

63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

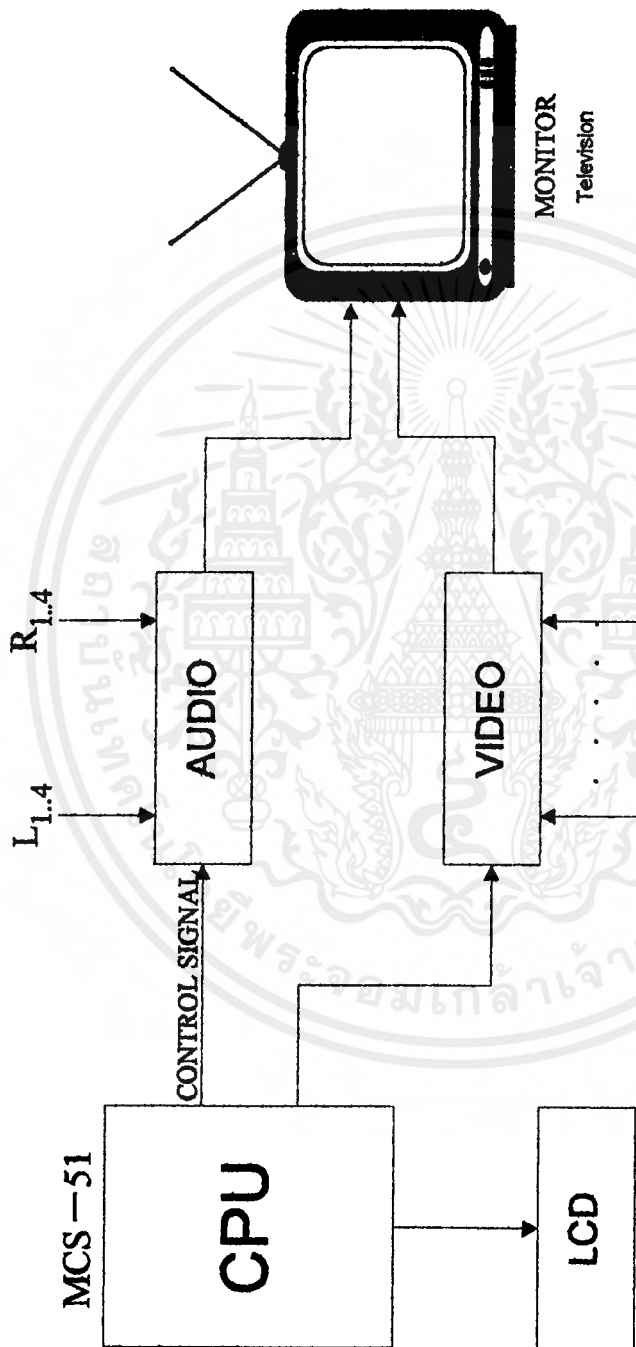
บทนำ

ในปัจจุบันระบบโทรทัศน์วงจรปิดมีการใช้กันอย่างแพร่หลายเมื่อเทียบกับในอดีต อันเนื่องมาจากหลายสาเหตุ โดยเฉพาะในการรักษาความปลอดภัยภายในตัวอาคาร ระบบจะมีขนาดใหญ่หรือเล็ก ขึ้นอยู่กับจำนวนจุดที่จะติดตั้งกล้องโทรทัศน์ ส่วนความซับซ้อนของระบบขึ้นอยู่กับขีดความสามารถของผู้ใช้ เช่น จะดูภาพจากกล้องแต่ละตัว จากจอมอนิเตอร์แต่ละตัวที่ต่อโดยตรงกับกล้องโทรทัศน์นั้นๆ ถ้าเป็นเช่นนี้ราคาของระบบจะแพงมากอันเนื่องมาจากจำนวนมอนิเตอร์หลายตัวเท่าจำนวนกล้อง ถ้าจะให้ประหยัดและไม่เปลืองเนื้อที่ ในการตั้งจอมอนิเตอร์ก็จะเป็นระบบที่ภาพจากกล้องโทรทัศน์ทั้งหมดมาเรียงกัน แสดงออกมาบนจอมอนิเตอร์เพียงจอเดียว โดยผ่านเครื่องแบ่งภาพโทรทัศน์ซึ่งมีราคาค่อนข้างสูงเช่นกัน ดังนั้นในระบบที่ไม่ใหญ่นักและมีงบประมาณน้อยจะใช้วิธีเลือกสัญญาณภาพจากแต่ละกล้องมาดูบนจอมอนิเตอร์จอเดียวที่ละกล้อง โดยใช้สวิตช์หรือเครื่องเลือกสัญญาณภาพซึ่งเป็นหัวข้อของโครงการนี้



ความสามารถของเครื่องเลือกสัญญาณภาพและเสียงของโครงการนี้

- สามารถต่อเข้ากับกล้องวิดีโอได้ 4 กล้อง
- สามารถแสดงผลโดยใช้มอนิเตอร์เพียงจอเดียว
- สัญญาณภาพแต่ละช่องจะผลัดเปลี่ยนกันแสดงผลที่จะมอนิเตอร์แต่ละช่องในโหมด SCAN อัตโนมัติ
- สามารถตั้งเวลาในการแสดงผลของสัญญาณภาพออกทางจอมอนิเตอร์ของแต่ละช่องสัญญาณได้
- สัญญาณภาพสามารถแสดงผลในแต่ละช่องที่เราต้องการได้ในโหมด MANUAL ซึ่งจะมีสวิตช์เลือกช่องสัญญาณในแต่ละช่อง
- แสดงผลออกทางจอ LCD DOT MATRIX
- หากช่องสัญญาณในไม่มีสัญญาณภาพจะไม่มีภาพ SCAN. หากช่องแต่จะยังคงภาพมืดไว้เพื่อประโยชน์ในงานด้านการรักษาความปลอดภัย เช่น ถ้ามีขโมยเข้ามาตัดสายสัญญาณที่กล้องวิดีโอตัวใดตัวหนึ่ง มอนิเตอร์ก็จะแสดงผลภาพมืดเอาไว้เพื่ออำนวยความสะดวกของเจ้าหน้าที่ว่ามีช่องสัญญาณใดเกิดความผิดพลาด
- ในโหมด MANUAL สามารถเลือกช่องสัญญาณในการแสดงผลทางจอมอนิเตอร์ทั้งภาพและเสียงได้อย่างอิสระ



BLOCK DIAGRAM

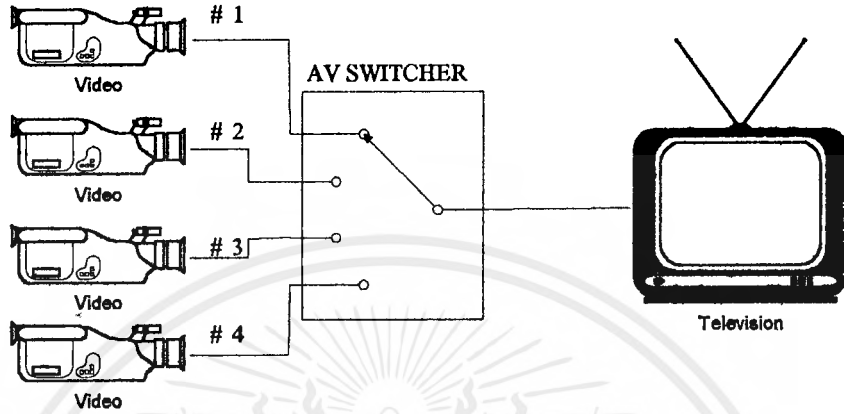
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

เครื่องเลือกสัญญาณภาพและเสียง

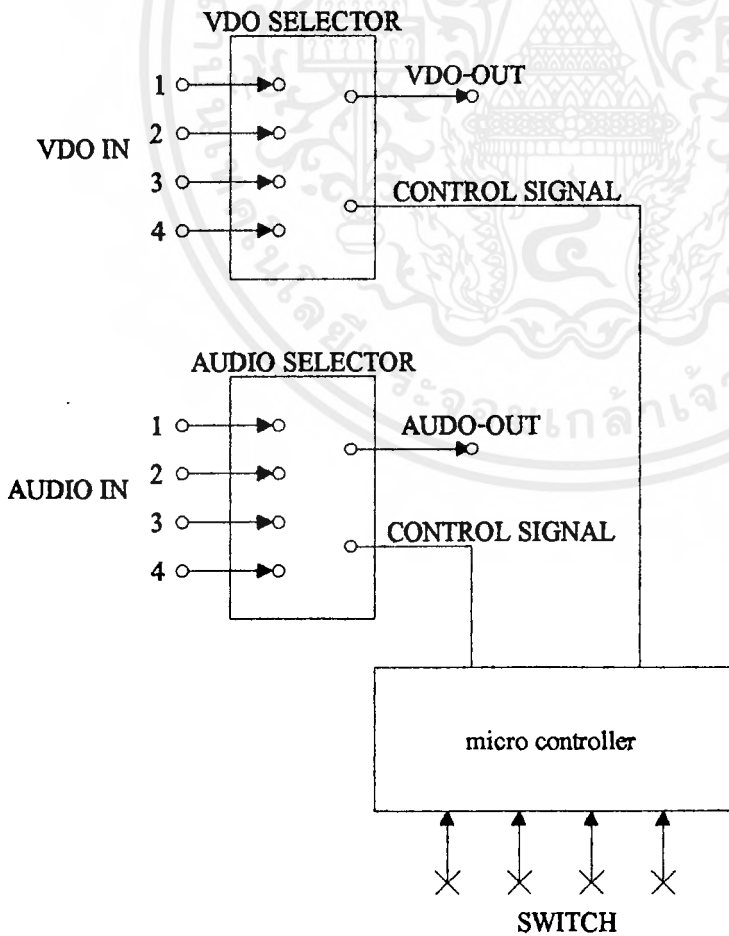
2.1 ระบบเลือกสัญญาณภาพโดยใช้สวิตช์

แสดงดังรูปที่ 2-1



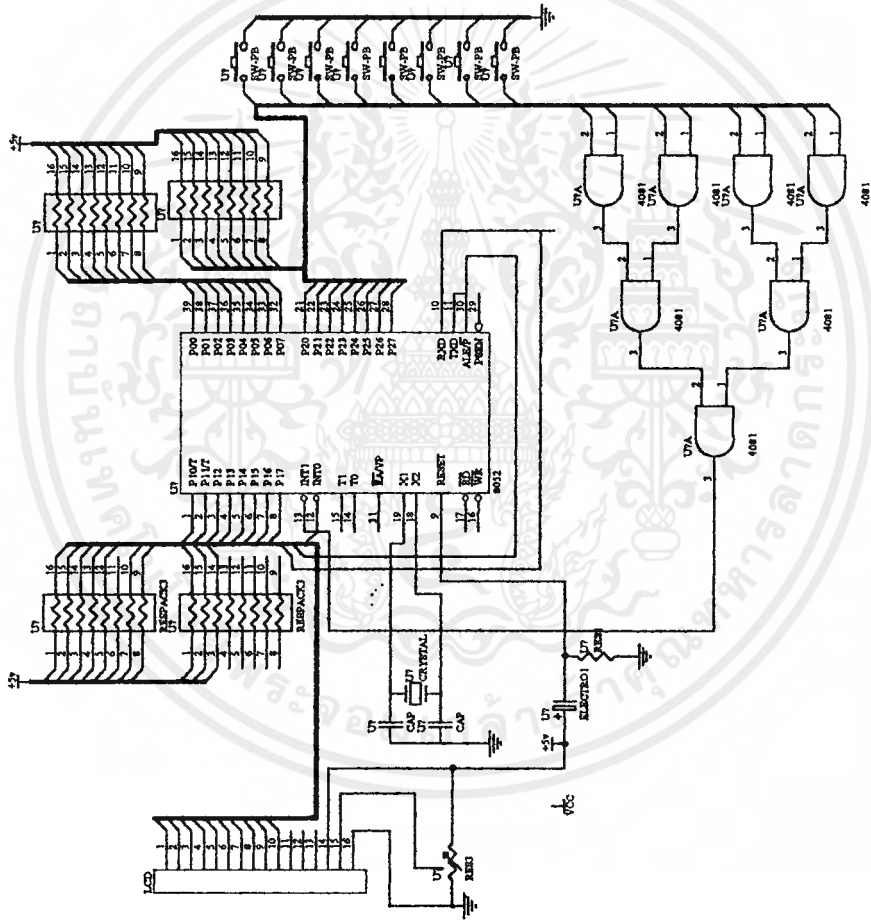
รูปที่ 2-1 การต่อสัญญาณภาพจากกล้อง 4 ตัว เข้ากับเครื่องเลือกสัญญาณภาพออกจอคอมพิวเตอร์โทรทัศน์ 1 เครื่อง

2.2 แผนผังแสดงภายในของเครื่องเลือกสัญญาณภาพ



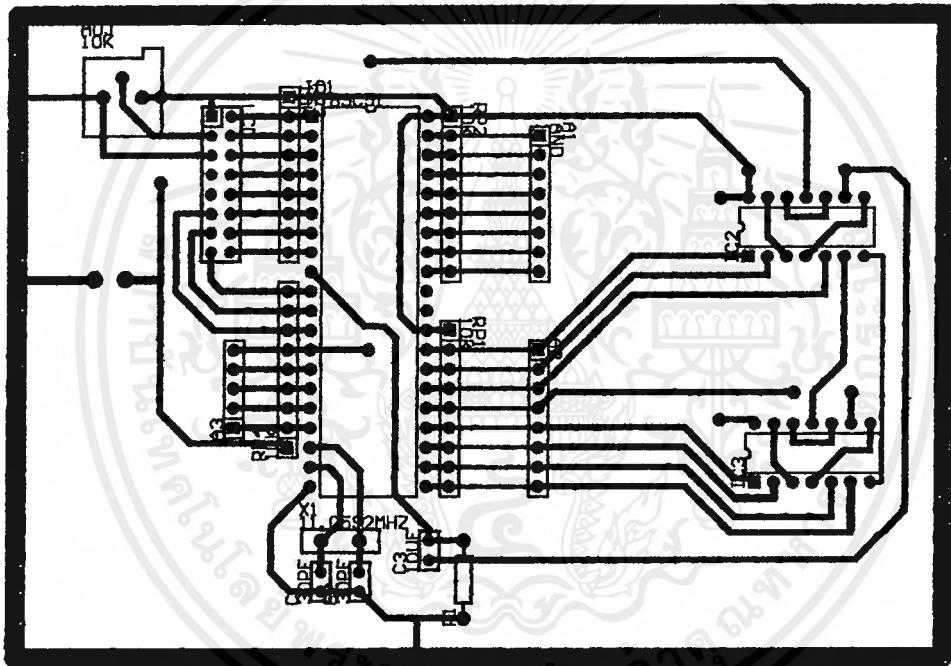
รูปที่ 2-2 แผนผังของเครื่องเลือกสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



File	Revision
Size	Number
Date	21 Sep 1999
File	C:\CIB\CIB\MSH
Sheet of	Drawn By
7	

รูปที่ 2-3 วงจรภาคไมโครคอนโทรลเลอร์



รูปที่ 2-4 ลายทองแดงของวงจรไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของเครื่อง

สิ่งที่สำคัญที่สุดในการทำงานก็คือ ส่วนที่เป็นตัวควบคุมการทำงานของระบบโดยในระบบนี้ใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS 51 ในการควบคุมการทำงานของระบบทั้งหมดและด้วยความสามารถของไมโครคอนโทรลเลอร์ที่สามารถประยุกต์ใช้งานได้หลายรูปแบบ จึงเป็นการง่ายที่จะนำมาใช้เป็นวงจรควบคุม โดยออกแบบให้มีพอร์ทอินพุทเพื่อรับสัญญาณจากสวิตต์ต่าง ๆ และพอร์ททางเข้าที่พหุมีไว้สำหรับส่งสัญญาณทางเข้าที่พหุที่ได้จากการประมวลผลใน CPU ไปทำการควบคุมชุดอนาล็อกสวิตต์ (ANALOG SWITCH) ให้เปิดปิดตามที่เรทำได้ทำการออกแบบไว้ ซึ่งการทำงานจะมีอยู่ 2 โหมด โดยมีสวิตต์ไว้สำหรับกดให้เลือกได้ดังนี้

การทำงานในโหมด MANUAL

สัญญาณภาพและสัญญาณเสียง เมื่อเราได้ทำการเลือกสัญญาณจากช่องใดช่องหนึ่งแล้ว CPU ก็ จะทำการรับสัญญาณ แล้วนำสัญญาณที่รับจากสวิตต์ส่งสัญญาณออกทางพอร์ทเข้าที่พหุตามช่องสัญญาณที่เราได้ทำการเลือกกด สัญญาณนี้ก็จะถูกส่งไปขับชุด ANALOG SWITCH และเช่นเดียวกันเราก็สามารถเลือกช่องสัญญาณเสียงได้โดยการเลือกจากสวิตต์เช่นเดียวกัน ซึ่งในโหมด MANUAL นี้ ผู้ใช้งานสามารถเลือกช่องของสัญญาณได้ตามต้องการ

การทำงานในโหมดสแกนอัตโนมัติ (AUTOMATIC SCANNING)

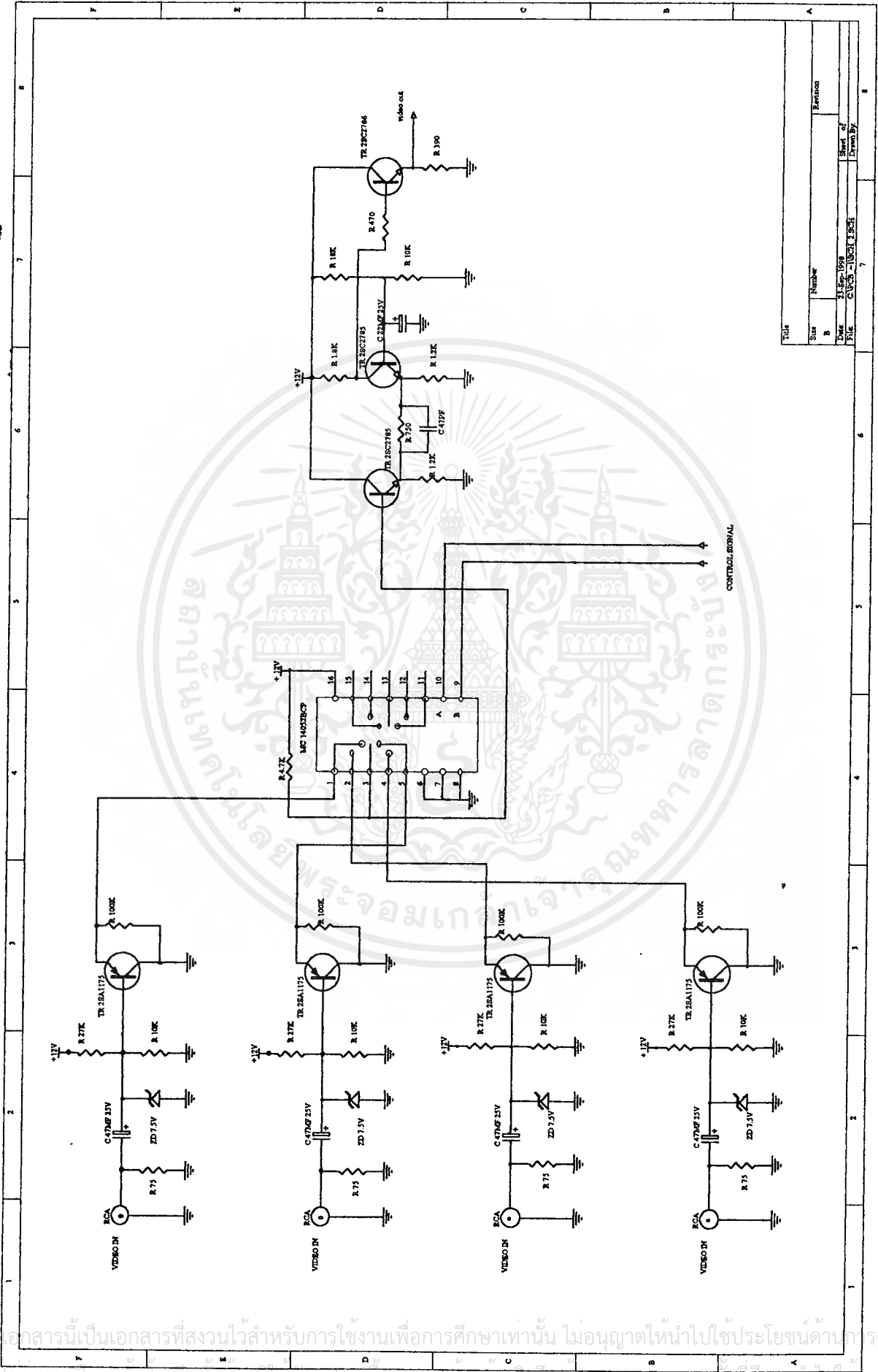
เมื่อเราเลือกใช้งานในโหมด AUTO SCAN CPU จะทำการส่งสัญญาณทาง OUTPUT ออกไปที่ PORT OUTPUT เพื่อนำไปควบคุมชุด ANALOG SWITCH โดยไม่จำเป็นต้องรอสัญญาณจากการเลือกช่องสัญญาณภาพและเสียง โดย CPU จะทำงานด้วยโปรแกรมที่โปรแกรมเมอร์ได้ทำการออกแบบไว้ และในขณะเดียวกัน CPU จะทำการส่งสัญญาณไปขับ ANALOG SWITCH ตัวที่ 1 CPU ก็ จะทำการส่งสัญญาณไปขับ ANALOG SWITCH ตัวที่ 5 ด้วย เพื่อทำให้เกิดภาพและเสียงที่ได้จากกล้องวีดีโอตัวเดียวกัน จากนั้นจะทำการหน่วงเวลาไว้สักพักหนึ่ง ซึ่งคาบของเวลานี้กำหนดได้โดยการโปรแกรมและก็จะเริ่มดำเนินการเช่นนี้กับช่องสัญญาณที่ 2 ที่ 3 และที่ 4 จนครบทุกช่องสัญญาณ และจะวงไปเริ่มที่ช่องสัญญาณที่ 1 ต่อไปอีก โดยการทำงานจะวนเช่นนี้ต่อไปเรื่อย ๆ แล้วแต่โปรแกรมที่เราได้ทำการออกแบบไว้ ภาคแสดงผล

ในภาคแสดงผลจะใช้จอแสดงผลเป็นแบบ LCD DOT MATRIX ขนาด 16 ตัวอักษร 1 บรรทัดไว้ทำการแสดงข้อความให้ผู้ใช้งานได้รู้ว่าเครื่องกำลังทำงานในโหมดการทำงานใด และบอกว่ากำลังแสดงผลของช่องสัญญาณใด และขณะนี้เป็นเวลาเท่าใด จอ LCD DOT MATRIX นี้ทำการเชื่อมต่อกับ CPU โดยผ่าน PORT 8255 ซึ่งเป็น PORT แบบขนานทำหน้าที่ในการเชื่อมต่อรับส่งข้อมูลระหว่าง CPU กับ LCD โดยอาศัยโปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

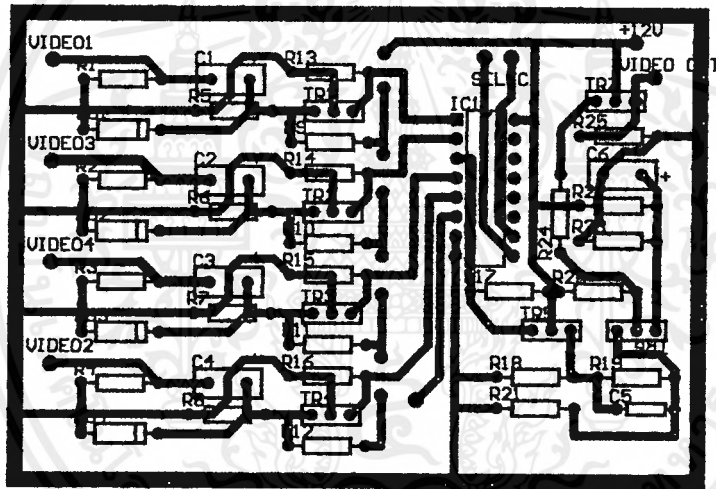
ภาค VIDEO SELECTOR

ภาค VIDEO SELECTOR มีหน้าที่ในการเลือกสัญญาณภาพ ซึ่งการเลือกสัญญาณนั้นถูกควบคุมด้วย CPU ซึ่งการควบคุมการทำงานขึ้นอยู่กับการออกแบบของผู้ใช้งานว่าจะทำการโปรแกรมให้ทำการรับสัญญาณภาพในช่องใด โดยสัญญาณควบคุมจาก CPU นั้นจะเข้ามาที่ขาที่ 9 และขาที่ 10 ของ IC เบอร์ MC 14052 ซึ่งทำหน้าที่เป็น IC ANALOG SWITCH ซึ่งวงจรที่ผู้ออกแบบได้ออกแบบมานั้นได้ทำการออกแบบมาให้สามารถรับสัญญาณภาพได้ 4 CHANNEL ซึ่ง IC เบอร์ MC 14052 มีโครงสร้างภายในเป็น SWITCH สามารถเลือกได้ 4 ตำแหน่งซึ่งมีอยู่ 2 ชุด โดยผู้ออกแบบได้ป้องกันสัญญาณ VIDEO INPUT เข้ามาที่ขา 1, ขาที่ 2, ขาที่ 4, ขาที่ 5 และสัญญาณ VIDEO ที่ได้จะออกที่ขา 3 โดยที่ขา 3 ผู้ออกแบบได้ต่อความต้านทานค่า 4.7 กิโลโอห์ม ต่อไว้เพื่อขจัดระดับสัญญาณ VIDEO ให้อยู่บนแรงดันไฟ DC ขาที่ 16 ต่อไฟ VCC และขาที่ 6, ขาที่ 7, ขาที่ 8 ต่อลง GROUND เมื่อได้รับสัญญาณ VIDEO ตามที่ต้องการแล้วก็จะถูกส่งออกมาที่ ขา 3 เพื่อป้องกันสัญญาณเข้าที่ขา BASE ของ TRANSISTOR เบอร์ 2 SC 2785 ซึ่งผลที่ได้จากการทดลองนั้นโดยทำการวัดรูปสัญญาณที่ออกจากขา EMITTER ของ TRANSISTOR นั้นมีการลดทอนของสัญญาณเพียงเล็กน้อยเท่านั้นและสัญญาณที่ได้ไม่มีการผิดเพี้ยน สำหรับสัญญาณที่ขา COLLECTOR นั้นจะมีก็แต่เพียงแรงดันไฟตรงเท่านั้น สำหรับ CAPACITOR ค่า 47 PF ที่ทำการต่อคร่อมความต้านทานค่า 750 โอห์ม จากวงจรที่แสดงไว้ที่ผู้ออกแบบได้ออกแบบไว้ที่ค่า C 47 PF แต่เมื่อทำการเพิ่มค่าของ CAPACITOR ไปเป็น 100 PF ทำให้ภาพที่แสดงผลออกทาง MONITOR สว่างขึ้น และเมื่อผู้ออกแบบได้ลองทำการเพิ่มแรงดันไฟที่ป้องกันเข้าวงจรอีกประมาณ 1-2 โวลต์ ทำให้ภาพที่ปรากฏบน MONITOR มีความคมชัดมากยิ่งขึ้น สำหรับสัญญาณภาพที่ผู้ออกแบบได้ลองทำการวัดที่ขา EMITTER ของ TRANSISTOR ตัวที่ 2 นั้นจะมีค่าแรงดันมากกว่า INPUT ถึง 10 เท่า และเมื่อผู้ออกแบบได้ทำการวัดสัญญาณที่ออกจากขา COLLECTOR สัญญาณที่ได้จะมากกว่า INPUT ถึง 2.5 เท่า เช่นเมื่อ CPU ส่งคำสั่งควบคุมมาที่ขา 9 และขาที่ 10 เป็น A=0, B=0 ที่ขา ENABLE ให้ LOGIC=0 ตำแหน่งของ SWITCH ก็จะเลื่อนลงมาที่ตำแหน่ง YO หรือขาที่ 1 ของ IC ถ้า CPU ส่งคำสั่งควบคุมให้ A=0, B=1 ตำแหน่งของ SWITCH ก็จะเลื่อนมาที่ SWITCH ตัวที่ 4 ถ้าคำสั่งควบคุมให้ A=1, B=0 ก็จะเลื่อนมาที่ SWITCH ตัวที่ 2 ถ้าคำสั่งควบคุมให้ A=1, B=1 ก็จะเลื่อนมาที่ SWITCH ตัวที่ 3 ก็จะได้ช่องสัญญาณตามที่ได้ทำการออกแบบมาแล้วนั่นเอง สำหรับ ZENER DIODE 7.5 VOLT นั้นเป็นตัวควบคุมขาแรงดัน 7.5 โวลต์ที่ขา BASE ของ TRANSISTOR



Title			
Size	Number	Revision	
B			
Name	21.5.1988	Sheet of	7
File	C:\MSD\105\11.85\1	Drawn By	

รูปที่ 2-5 วงจรเลือกสัญญาณภาพ

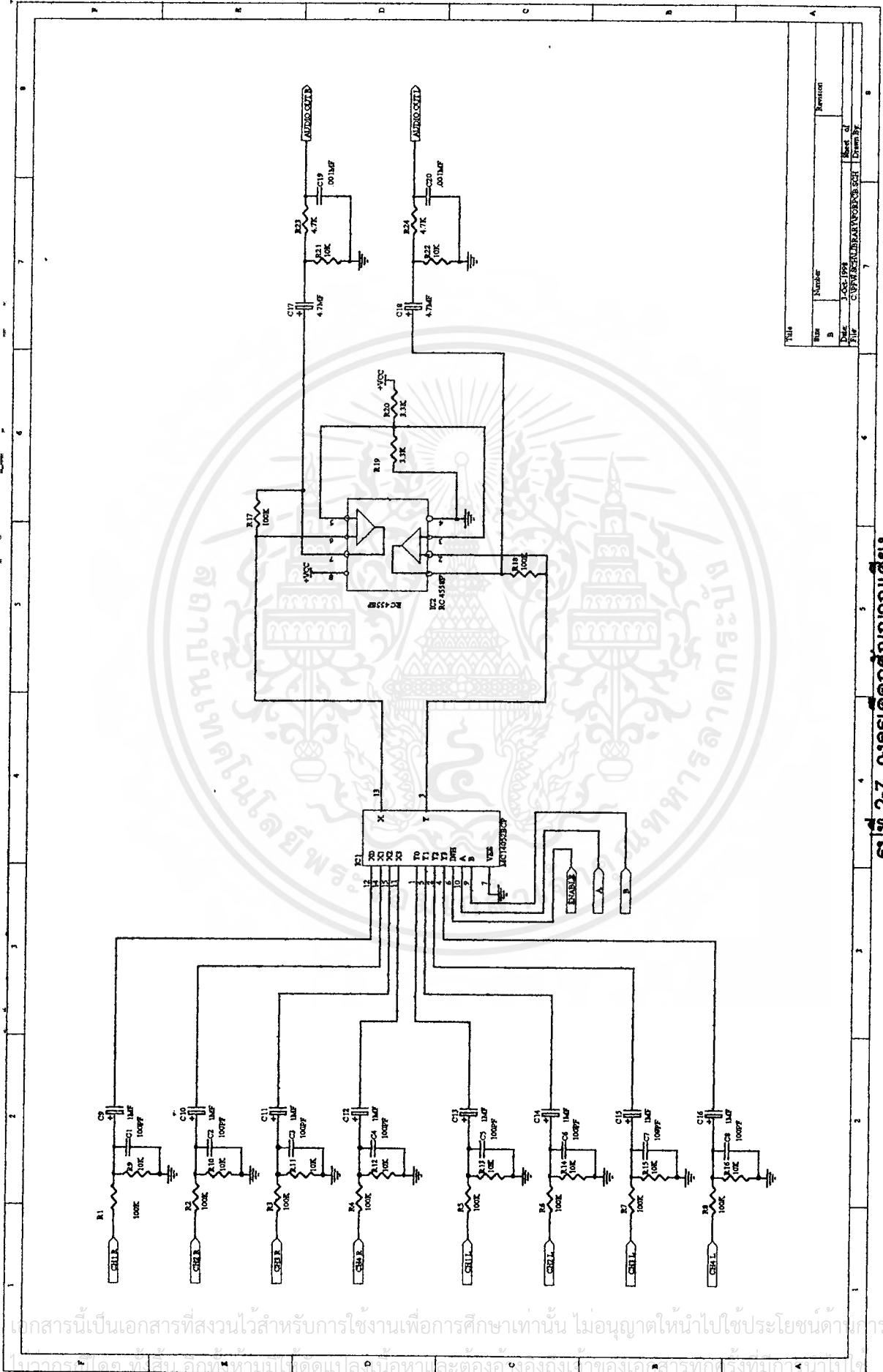


รูปที่ 2-6 ลายทองแดงของวงจรเลือกสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

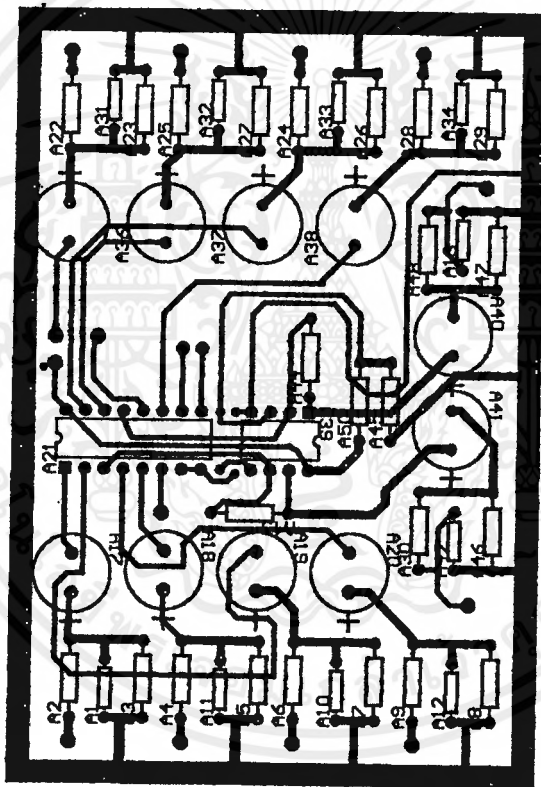
ภาค AUDIO SELECTOR

ภาค AUDIO SELECTOR มีหน้าที่ในการเลือกสัญญาณเสียง โดยทำการเลือกที่ละช่องสัญญาณเสียง โดยสัญญาณเสียงที่เลือกจะควบคุมมาจาก CPU ซึ่งขึ้นอยู่กับการออกแบบของผู้ออกแบบว่าจะใช้งานในช่องสัญญาณใด สัญญาณจาก CPU จะถูกส่งมาที่ขา 9 และขาที่ 10 ของ IC ที่ผู้ออกแบบให้ใช้เบอร์ MC 14052 โดย IC เบอร์นี้ทำหน้าที่เป็น IC ANALOG SWITCH ซึ่งวงจรที่ออกแบบมานั้นผู้ออกแบบได้ออกแบบมาเพื่อให้รับสัญญาณทาง INPUT ได้ 4 ช่องสัญญาณ ภายในตัว IC นี้มีสวิตช์อยู่ 2 ชุด แต่ละชุดประกอบด้วยสวิตช์ 4 ตำแหน่ง ซึ่งผู้ออกแบบได้ออกแบบมาเพื่อรับสัญญาณ INPUT เข้ามาทางขาที่ 1 ขาที่ 2 ขาที่ 4 และขาที่ 5 โดยกำหนดให้เป็นช่องสัญญาณทางด้านขวา และสัญญาณที่เข้าทางขา 11 และขาที่ 12 ขาที่ 14 และขาที่ 15 ทำการออกแบบมาให้เป็นช่องสัญญาณทางด้านซ้าย สำหรับขาที่ 6 ขาที่ 7 และ 8 ต่อดลง GROUND และมีขาที่ 16 ต่อกับไฟตรง DC เช่น เมื่อผู้ออกแบบทำการโปรแกรมที่ CPU โดยกำหนดมาให้ CHANNEL SELECT INPUT เป็น A=0 B=0 และขา ENABLE เป็น 0 ตำแหน่งของสวิตช์ก็จะเลื่อนมาที่ตำแหน่ง Y0 หรือขาที่ 1 สัญญาณ AUDIO INPUT ก็จะเป็นช่องสัญญาณที่ 1 ตามที่ผู้ออกแบบได้ออกแบบไว้ ถ้าผู้ออกแบบโปรแกรมที่ CPU A=0 B=1 สวิตช์ก็จะเลื่อนตำแหน่งมาที่ Y4 หรือขา 5 ถ้าผู้ออกแบบโปรแกรมให้ A=1 B=0 สวิตช์ก็จะเลื่อนขาที่ตำแหน่ง Y2 หรือขาที่ 2 ถ้าให้ A=1 B=1 สวิตช์ก็จะเลื่อนมาที่ตำแหน่ง Y3 หรือขาที่ 4 ตามที่ผู้ออกแบบได้ออกแบบไว้ทุกประการ เมื่อสัญญาณที่ถูกเลือกได้ตามต้องการแล้วสัญญาณก็เข้ามาที่ BUFFER โดยที่ผู้ออกแบบมาให้ใช้เบอร์ RC 458 ซึ่งออกแบบมาเพื่อมิให้เกิดการดึงกระแสจาก IC 14052 โดยตรง



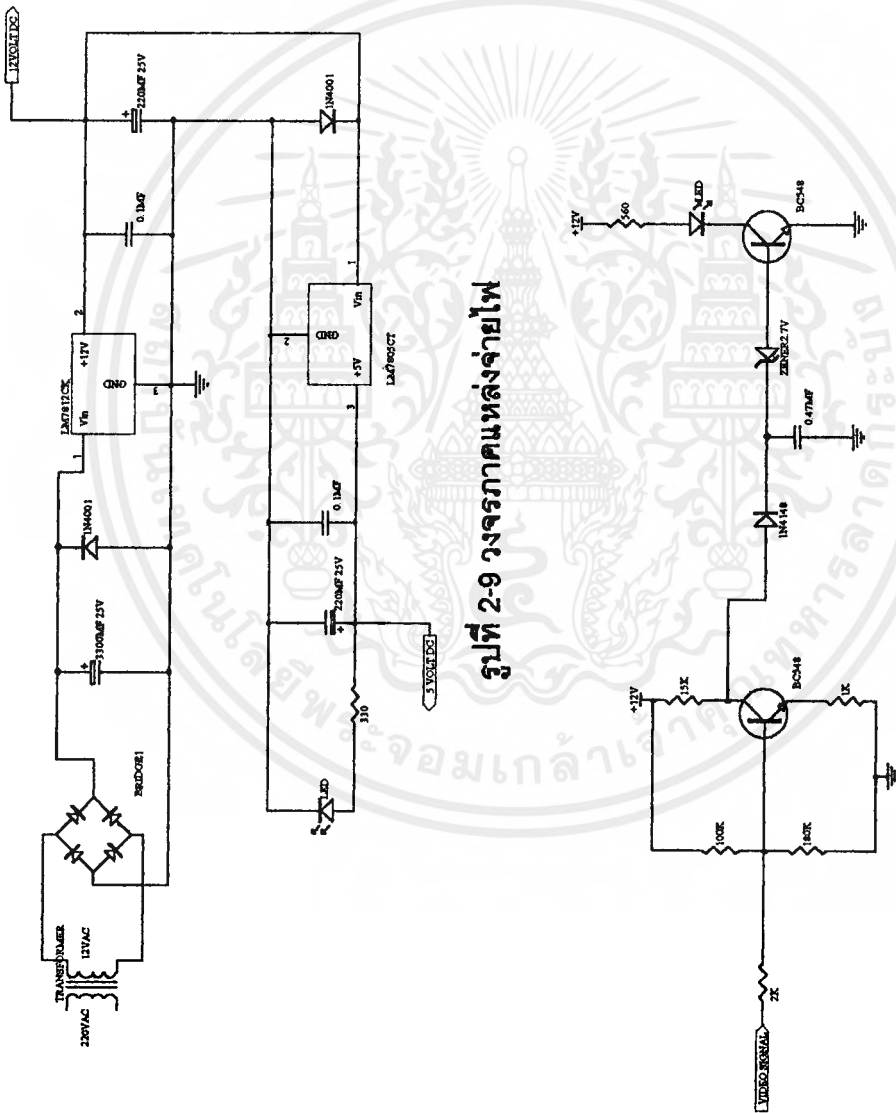
File	Number	Revision
B		
Date	3-Oct-1999	Sheet of
File	C:\PWA\BENCH\LAB\KAWAIVOXPCB.SCH	Drawn By

รูปที่ 2-7 วงจรเลือกสัญญาณเสียง



รูปที่ 2-8 ลายทองแดงของวงจรเลือกสัญญาณเสียง

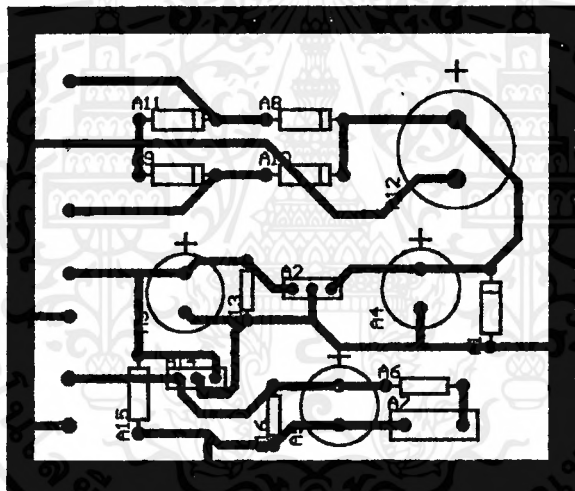
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-9 วงจรภาคแหล่งจ่ายไฟ

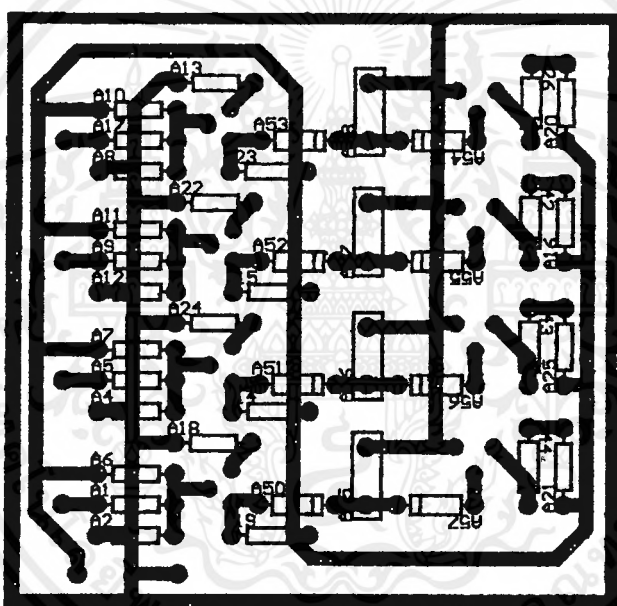
รูปที่ 2-10 วงจรภาคแสดงสถานะของอินพุตสัญญาณภาพ

Title	
Size	Number
B	
Date	1-Oct-1998
File	C:\PCB - POWER.ECH
Drawn By	
Sheet of	7
Drawn By	



รูปที่ 2-11 ลายทองแดงของวงจรแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-12 ลายทองแดงของวงจรแสดงสถานะของอินพุตสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

DOT MATRIX LCD MODULE

อุปกรณ์ในปัจจุบันนี้ในส่วนแสดงผลนั้นจะใช้ LCD เสียเป็นส่วนใหญ่ ไม่ว่าจะเป็นเครื่องเล่นวีดีโอเทป เครื่องถ่ายภาพเอกสาร เครื่องมือวัดคุมต่าง ๆ เครื่องคอมพิวเตอร์ หรือแม้แต่เพจเจอร์ เราพอจะแบ่ง DOT MATRIX LCD MODULE นี้ออกเป็นพวก ๆ ดังนี้

1. CHARACTER LCD MODULE
2. GRAPHIC LCD MODULE
3. SEGMENT DISPLAY TYPE LCD MODULE

โดยในแต่ละแบบนี้จะมีส่วนประกอบใหญ่ ๆ แบ่งได้เป็น

1. DOT MATRIX LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการเปิดและปิดกับแสง ก็คือส่วนที่เป็นกระจกบรรจุผลึก
2. DRIVER เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD อีกทีหนึ่งโดยมีเบอร์ที่นิยมใช้ใน LCD MODULE เช่น HD44100H, MSM5259
3. CONTROLLER เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุม LCD MODULE ให้ทำงานแสดงผลต่าง ๆ เช่น การลบจอภาพ, การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ IC ที่นิยมใช้กันคือ HD4478 ซึ่งจะใช้ในแบบ CHARACTER LCD MODULE เป็นส่วนใหญ่และ เบอร์ IC HD61830 จะใช้ในแบบ GRAPHIC LCD MODULE

ในการศึกษาการทำงานและใช้งาน LCD MODULE นั้นไม่ใช่เรื่องยากเลยถ้าเราสามารถทำความเข้าใจในส่วน ของ CONTROLLER ได้ก็เพียงพอแล้วและโดยมาก LCD MODULE ในแต่ละบริษัทแล้วจะใช้ตัวอักษรหรือ จำนวนบรรทัดก็มีหลักการทำงานแบบเดียวกันทั้งหมด IC ที่นิยมมากที่สุดตัวหนึ่งที่เป็น CONTROLLER LCD ก็คือ เบอร์ HD44780 โดยรูปแบบการทำงานของมันเป็นมาตรฐานให้กับ CONTROLLER LCD ตัวอื่น ๆ ด้วย

HD44780 เป็นไอซี LSI ตัวหนึ่งใช้ควบคุม LCD โดยแสดงผลในรูปตัวอักษรหรือสัญลักษณ์ต่าง ๆ ตัวมันเองสามารถต่อใช้งานแบบ 4 BIT หรือ 8 BIT ก็ได้ โดยถ้าเราต่อแบบ 4 BIT จะต่อใช้งานที่ DB7-DB4 เท่านั้น โดยข้อมูลครั้งแรกที่ส่งนั้น DH44780 จะถือเป็นข้อมูล 4 BIT บน และข้อมูลที่ส่งต่อมานั้นเป็นข้อมูล 4 BIT ล่าง

3-1 รายละเอียดของคำสั่ง HD44780

1. CLEAR DISPLAY

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2 RETURN HOME

RS	R/W	DB7							DB0
0	0	0	0	0	0	0	0	0	0

.Noeffect

คำสั่งนี้จะทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพข้อมูลในจอภาพไม่เปลี่ยน

3 ENTRY MODE SET

RS	R/W	DB7							DB0
0	0	0	0	0	0	0	0	I/D	S

BIT I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM ADDRESS เพิ่มขึ้นหรือลดลงหนึ่งโดย

1 = เพิ่ม

0 = ลดลงหนึ่ง

BIT S : โดยจะเป็นตัวกำหนดแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัว CURSOR อยู่ที่ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่กับที่ตัว CURSOR จะถูกดันไปทางขวามือ

4 DISPLAY ON/OFF CONTROL

RS	R/W	DB7							DB0	
0	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 และจะอยู่ที่ LINE ที่ 11 ในแบบ 5X10 DOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปเผยแพร่ภายนอก

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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

5 CURSOR OR DISPLAY SHIFT

0	0	0	0	0	1	S/C	R/L	.	.
---	---	---	---	---	---	-----	-----	---	---

เป็นคำสั่งกำหนดให้ตำแหน่ง CURSOR หรือข้อมูลไปเกิดทางซ้ายหรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่าน โดย

S/C	R/L	
0	0	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปขวามือ 1 ตำแหน่ง
1	0	เป็นการดันตัวอักษรที่เกิดไปทางซ้าย
1	1	เป็นการดันตัวอักษรที่เกิดไปทางขวามือ

6 FUNCTION SET

RS	RW	DB7	DB0					
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

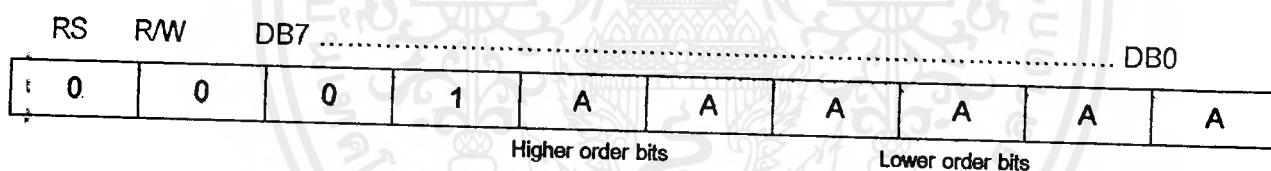
F: เป็นการ SET ขนาด DOT การแสดงผล 5X7 หรือ 5 X 10 โดย

F=0 เป็นแบบ 5X7 และ F=1 เป็นแบบ 5X10

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

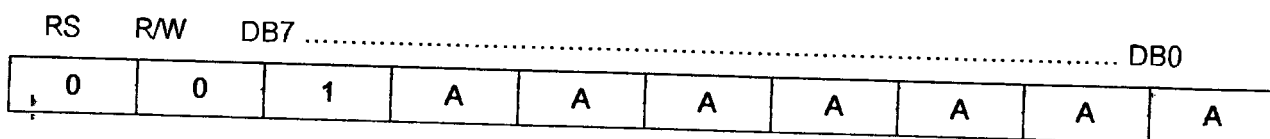
. No effect

7 SET CG RAM ADDRESS



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



เป็นคำสั่ง SET ค่า ADDRESS ใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM (DD RAM คือส่วนที่จะแสดงผลหน้าจอ LCD โดยจำนวน ADDRESS ที่จะเกิดขึ้นบนจอ LCD จะอยู่กับ SET ค่า N ด้วย
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

แบบการจัด ADDRESS ของ DD RAM หน้าจอ LCD แบบ 16 ตัวอักษร 1 บรรทัด 16 ตัว
อักษร 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

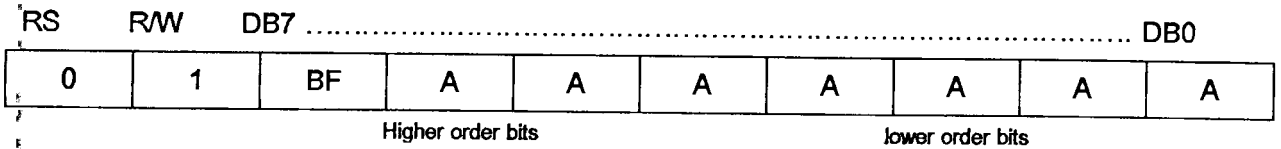
20 ตัวอักษร 1 บรรทัด

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
9. READ BUSY FLAG AND ADDRESS
ไม่ว่ากรณีใดๆ พงษ์สัน อภิสิทธิ์หามมเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เป็นคำสั่งอ่านค่า BUSY FLAG ซึ่งจะเป็นตัวบอกว่าตัว HD44780 นี้ อยู่ในขบวนการทำงานภายในอยู่หรืออยู่ในสภาพพร้อมจะรับข้อมูล โดย

BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

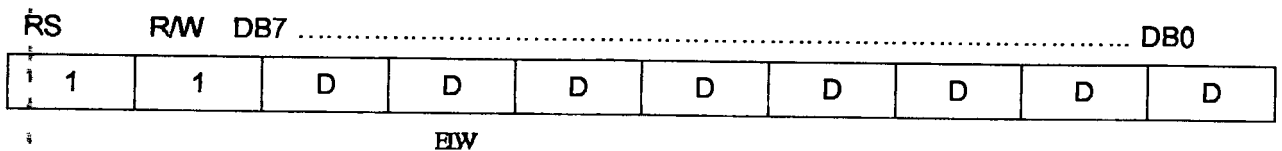
และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล ADDRESS ของ CG RAM หรือ DD RAM ด้วย

10. WRITE DATA TO CG หรือ DD RAM



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 ขึ้นมาก่อนจะเขียนข้อมูล

11. READ DATA FROM CG OR DD RAM



เป็นคำสั่งอ่านค่าข้อมูลจาก CG RAM หรือ DD RAM โดยก่อนอ่านค่าจาก DD RAM หรือ CG RAM นี้ควรจะใช้คำสั่ง SET ADDRESS ก่อนเพื่อให้รู้ว่าข้อมูลที่อ่านได้นั้นเป็น DD หรือ CG RAM

จากตารางการทำงานจะเห็นว่าการใช้งาน LCD MODULE นั้นง่ายเพียงแต่เราส่งคำสั่งเริ่มแรกและ

SET ความต้องการในขนาดตัวอักษร CURSOR หลังจากนั้นเราก็สามารถเขียนตัวอักษรเข้าไปใน DD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RAM ตามตารางตัวอักษรที่ให้มานั้นก็จะเกิดอักษรในจอภาพ LCD เรายังสามารถกำหนดตำแหน่งตัวอักษรที่จะให้เกิดบนจอได้โดยการ SET DD RAM ADDRESS ตามตารางที่ให้มาในหัวข้อ SET DD RAM ADDRESS ขอให้ทดสอบทำความเข้าใจกับตัวโปรแกรมที่ใช้กับ ET-BOARD V3.0, V3.5 นี้ที่ให้มาจะเห็นว่ามีส่วนเริ่มต้นก็คือ ส่วนการ INITIAL LCD เพื่อกำหนดหน้าที่การทำงานต่าง ๆ

3-2 ส่วนประกอบของโปรแกรม

EPLUSE จะเป็นส่วนกำเนิดสัญญาณ ENABLE SIGNAL โดยการใช้ PORT B BIT ที่ 2 กำเนิด PULSE สัญญาณ ENABLE ขึ้น

GOTO จะเป็นส่วนกำหนดตำแหน่งของส่วน DD RAM ADDRESS ที่จะเขียนข้อมูล โดยจากโปรแกรม INITIAL ที่เรา SET ไว้เมื่อเขียนข้อมูลเข้าไปใน DD RAM แล้ว ADDRESS ของ DD RAM จะเพิ่มขึ้น 1 โดยทันที

WRBYTE เป็นส่วนเขียนข้อมูล 1 BYTE เข้าไปในตำแหน่ง ADDRESS ของ DD RAM ขณะนั้น ๆ

WRLINE เป็นส่วนในการเขียนข้อมูลที่ละ 1 LINE เพราะตำแหน่ง DD RAM ที่เกิดบนจอภาพ LCD นั้นแต่ละตำแหน่งจะไม่ต่อกันไปในแต่ละบรรทัด

จากตัวอย่างที่ให้จะมี LCD แบบ 20 ตัวอักษร 2 บรรทัด และ 16 ตัวอักษร 4 บรรทัด

นอกจากนี้ LCD MODULE (HD44780) นี้จะยังมีส่วนหนึ่งของ CHARACTER GENERATOR ที่เราสามารถเขียนข้อมูลในการเกิดตัวอักษรขึ้นได้เองจากตารางตัวอักษร 5X7 DOT นั้นจะเห็นว่าเป็นตำแหน่งในตาราง 00H ถึง 07H ส่วนตำแหน่ง 08H-0FH จะเป็นตำแหน่งเดียวกับ 00H-07H จะเห็นว่าจะมี CHARACTER GENERATOR 8 ตัวที่เราสามารถเขียนข้อมูลกำหนดเองได้และถ้าเป็นแบบ 5X10 DOT จะเขียนได้ 4 ตัวอักษร ซึ่งจากข้อพิเศษนี้ทำให้เราสามารถเขียนตัวอักษรและสัญลักษณ์หรืออักษรภาษาไทยได้

3-3 การเขียนข้อมูล CHARACTER GENERATOR

เราสามารถเขียนข้อมูลได้โดยกำหนด ADDRESS ของ CG RAM ก่อนโดยเขียนได้ 64 ตำแหน่ง BIT 5-BIT 0 และเมื่อกำหนด ADDRESS แล้วก็จะทำการเขียนข้อมูลลงใน CG RAM โดยเป็นลักษณะ BIT ต่อ BIT บนจอ 1 ตัวอักษร คือ 5X7 DOT นั้นจะใช้ข้อมูล BIT 4 ถึง BIT 0 ต่อ 1 BYTE เท่านั้น 1 ตัวอักษรจะใช้ข้อมูล 8 BYTE ด้วยกันให้ดูจากตารางประกอบไปด้วยและเมื่อเขียนข้อมูลลงใน CG RAM แล้วเวลาเราจะใช้งานก็ให้เขียนข้อมูลใน DD RAM คือข้อมูลตำแหน่งในตาราง CHARACTER ที่ตำแหน่ง 00H-07H ตัวอย่างโปรแกรมการเขียนข้อมูลตัวหนังสือภาษาไทยเป็นตัว (อ), (ท) และตัว () เข้าไปใน CG RAM ตำแหน่งที่ 00H, 01H และ 02H และนำมาแสดงผลทางจอ LCD โดยใช้ 2 บรรทัดในการแสดงผล

สรุป การใช้งาน LCD MODULE นั้นที่สำคัญคือต้องเข้าใจในตัว CONTROLLER ของ LCD MODULE นั้น โดย CONTROLLER ทุก ๆ บริษัทจะมีการทำงานที่เหมือน ๆ กันเป็นส่วนใหญ่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ไมโครคอนโทรลเลอร์ตระกูล 8051

ไมโครคอนโทรลเลอร์เป็นไมโครโปรเซสเซอร์ประเภทหนึ่งที่ได้รับการออกแบบมาเพื่อใช้งานกับระบบควบคุมที่มีขนาดเล็ก โดยภายในมีไอซีคอนโทรลเลอร์หนึ่งตัวจะประกอบด้วยหน่วยการทำงานหลักของคอมพิวเตอร์ครบถ้วน เช่น หน่วยประเมินผลกลางหรือ (CPU) หน่วยความจำ พอร์ทในการติดต่อหรือควบคุมอุปกรณ์ต่างๆ เป็นต้นซึ่งหากว่าเป็นการใช้งานไมโครโปรเซสเซอร์ทั่วไปก็จะต้องใช้ไอซีภายนอกมาประกอบเพื่อทำหน้าที่เหล่านี้ ดังนั้นจึงอาจกล่าวได้ว่าไมโครคอนโทรลเลอร์เป็นระบบคอมพิวเตอร์เพียงงานควบคุมที่สมบูรณ์ โดยบรรจุอยู่ในตัวไอซีเพียงหนึ่งตัวเท่านั้น ในบางครั้งจึงอาจพบว่ามีเรียกไมโครคอนโทรลเลอร์ว่าเป็น ระบบไมโครคอมพิวเตอร์ชนิดเดียว

ในบรรดาไมโครคอนโทรลเลอร์ที่นิยมใช้กันอย่างแพร่หลายคือตระกูล 8051 ของบริษัทอินเทล (intel corporation)

4-1 คุณลักษณะพื้นฐานของ 8051

จากแผนภาพในรูปที่ 1.1 แสดงให้เห็นถึงหน่วยการทำงานพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ต่างๆ ที่จัดอยู่ภายในตระกูล MCS-51 นี้ ประกอบด้วย

หน่วยประมวลผลกลางขนาด 8 บิต

หน่วยประมวลผลสำหรับข้อมูลแบบบิต (Boolean Processor)

ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์

ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์

หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ EPROM (เบอร์ 8751) หรือแบบ ROM (เบอร์ 8051)

หน่วยความจำแบบ RAM ภายในจำนวน 128 ไบต์

พอร์ทอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ

วงจรรับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร

วงจรถูกส่งแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)

วงจรถอบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับ

ความสำคัญได้สองระดับ

วงจรออสซิลเลเตอร์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้ มักจะมีรูปร่างของไอซีเป็นแบบ DIP ขนาด 40 ขา ดังแสดงเป็นแผนภาพในรูปที่ 1.2 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ ชื่อย่อที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่า Alternate Functions ในรูปที่ 1.2) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ตัวอย่างเช่น ขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อเป็น P3.0) อาจจะใช้เป็นขาสัญญาณเอาต์พุต หรืออินพุตตามปกติ หรืออาจทำหน้าที่เป็นขาสัญญาณอินพุตของข้อมูลสื่อสารแบบอนุกรม (RxD) ให้กับวงจรสื่อสารแบบอนุกรมของ 8051 ได้ ซึ่งการกำหนดว่าจะทำงานในลักษณะใดก็ขึ้นอยู่กับ การเชื่อมต่อวงจรเข้ากับขาสัญญาณและโปรแกรมควบคุมของระบบนั้น

4-2 โปรแกรมสั่งงานของไมโครคอนโทรลเลอร์

โดยทั่วไปไมโครคอนโทรลเลอร์จะมีชุดคำสั่ง (Instruction Set) อยู่จำนวนหนึ่งเพื่อที่ผู้ใช้งานจะได้ทำการสั่งให้มีการดำเนินงานบางอย่างตามที่ได้กำหนดไว้ ซึ่งตามความเป็นจริงแล้วคำสั่งเหล่านี้เป็นรูปแบบข้อมูลตัวเลข ฐานสอง (Binary bit Pattern) ซึ่งจะนำไปส่งให้กับหน่วยประมวลผลกลางหรือซีพียูภายในไมโครคอนโทรลเลอร์เอง ในช่วงเวลาที่เหมาะสม ตัวอย่างเช่นข้อมูลในรูปที่ 1.3

การนำคำสั่งหลายๆ คำสั่งมารวมเข้าด้วยกันนั้นเพื่อทำงานอย่างใดอย่างหนึ่งจะใช้คำศัพท์ว่าโปรแกรม (Program) ดังนั้นการนำรูปแบบของข้อมูลเลขฐานสองหลายๆ ชุดมาเรียงลำดับติดต่อกันไปดังตัวอย่างในรูปที่ผ่านมา จะเรียกว่า โปรแกรมภาษาเครื่อง (Machine Language Program) โดยใช้ในความหมายที่เสมือนกับว่าได้มีการใช้ภาษาเพื่อติดต่อให้กับไมโครคอนโทรลเลอร์ทำงานและรูปแบบของภาษานี้ ไมโครคอนโทรลเลอร์ก็สามารถที่จะทำความเข้าใจได้โดยตรงว่าจะต้องมีการดำเนินการอย่างไรบ้าง

อย่างไรก็ตามการสร้างโปรแกรมด้วยภาษาเครื่องมีความไม่สะดวกหลายประการอาทิเช่น

1. ความลำบากต่อการทำความเข้าใจโปรแกรมในภายหลัง
2. การใช้เวลาในการใส่ข้อมูลภายในโปรแกรม เนื่องจากต้องใส่ค่าของบิตข้อมูลที่ละตัวๆ เป็น

ลำดับ

3. ไม่สามารถสื่อความหมายในลักษณะของตัวหนังสือที่ใช้กันในชีวิตประจำวันได้
4. เมื่อเกิดความผิดพลาดขึ้นภายในโปรแกรม จะค้นหาตำแหน่งผิดพลาดได้ยาก เนื่องจากมี

เพียงตัวเลข 1 และ 0 วางสลับกันไปมาในรูปแบบที่แตกต่างกัน

ดังนั้นในเวลาต่อมาจึงได้มีการปรับปรุงให้ใช้ตัวเลขฐานสิบหก (Hexadecimal numbers) เข้ามาทดแทนการใช้ตัวเลขฐานสองโดยตรง ตัวอย่างโปรแกรมในรูปที่ 1.3 เมื่อเขียนอยู่ในลักษณะของเลขฐานสิบหกจะทำให้มีความกระชับและสร้างโอกาสของความผิดพลาดให้น้อยลงได้มาก ตามลักษณะในรูปที่ 1.4 อย่างไรก็ตามการตัวเลขนี้ก็เพียง เฉพาะช่วงของการสร้างเป็นโปรแกรมนั้นขั้นตอนที่ไม่ใคร่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลเลอร์จะทำความเข้าใจก็ยังคงเป็นรหัสตัวเลขฐานสองอยู่เช่นเดิม ซึ่งหน้าที่เหล่านี้ จะมีโปรแกรมคอมพิวเตอร์เพื่อทำหน้าที่แปลเลขฐานสิบหกไปเป็นเลขฐานสองอีกต่อหนึ่ง

อย่างไรก็ตาม แม้ว่าจะได้ใช้ระบบเลขฐานสิบหกมาทดแทนการป้อนข้อมูลเป็นเลขฐานสองแต่ ก็ยังคงสร้างความยุ่งยากในการจดจำรหัสตัวเลขที่เทียบเท่ากับการใช้คำสั่งต่างๆ ในระยะต่อมาจึงได้ใช้วิธีการสร้างเป็นชื่อของคำสั่งที่เรียกว่า นิโมนิก (Mnemonics) ซึ่งชื่อแต่ละชื่อของรหัสนิโมนิกนั้นจะมีความหมายใกล้เคียงกับการทำงานของคำสั่งนั้นมาก อย่างไรก็ตามไมโครคอนโทรลเลอร์แต่ละตระกูลก็ มักจะมีการกำหนดชื่อของรหัสนิโมนิกที่แตกต่างกันออกไป ซึ่งเป็นหน้าที่ของผู้ใช้ที่จะต้องค้นหาจาก เอกสารหรือคู่มือใช้งานเองว่า ได้มีการกำหนดชื่อไว้ได้อย่างไร (ขอให้ดูรายการชุดคำสั่งของ 8051 ได้ จากภาคผนวก ก) ดังนั้นเมื่อโปรแกรมในรูปที่ 1.3 ได้นำมาเขียนอยู่ในรูปแบบของรหัสนิโมนิกของ 8051 แล้ว จะมีลักษณะดังในรูปที่ 1.6 ซึ่งโปรแกรมในลักษณะเช่นนี้จะเรียกว่า โปรแกรมภาษาแอสเซมบลี (Assembly Language Program)

4-3 โปรแกรมแอสเซมเบลอร์

เมื่อโปรแกรมได้รับการพัฒนาสร้างขึ้นด้วยภาษาแอสเซมบลี ซึ่งเป็นรหัสอักษรที่สร้างขึ้นมา ภายหลัง เพื่อให้จะให้มนุษย์เราสามารถสั่งงานหรือตรวจสอบความผิดพลาดของคำสั่งได้สะดวกมากยิ่งขึ้น ดังนั้นเมื่อนำโปรแกรมไปใช้งานจึงจำเป็นต้องทำการแปลโปรแกรมภาษาแอสเซมบลีนี้ไปเป็น ภาษาเครื่อง หรือตัวเลขฐานสิบหก ซึ่งไมโครคอนโทรลเลอร์จะสามารถเข้าใจได้เท่านั้นลักษณะของการ แปลความหมายของรหัสนิโมนิกที่ง่ายที่สุด ทำได้โดยการเปิดตารางของชุดคำสั่ง(เช่นในภาคผนวก ข) และแทนที่ด้วยตัวเลขฐานสิบหกของคำสั่งนั้นๆ การแปลความหมายของการแอสเซมบลีด้วยตนเอง (Hand Asseblly) เช่นนี้ ถือว่าเป็นวิธีที่ง่ายที่สุดและไม่จำเป็นต้องมีเครื่องมือช่วยงานใดๆ ทั้งสิ้น แต่ อย่างไรก็ตามเนื่องจากว่าเป็นงานที่ไม่น่าสนใจมีความซ้ำซากและมีโอกาสผิดพลาดได้ง่าย ดังนั้นจึงได้มีการพัฒนาโปรแกรมบนเครื่องไมโครคอมพิวเตอร์เพื่อช่วยการทำงานในลักษณะเช่นนี้แทน ซึ่งเรียกว่า โปรแกรมแอสเซมเบลอร์(Assembler Program)

การทำงานของโปรแกรมแอสเซมเบลอร์จะมีการแปลความโปรแกรมภาษาแอสเซมบลี ที่ผู้ใช้ สร้างขึ้นซึ่ง ก็มักจะเรียกว่าเป็น โปรแกรมต้นฉบับ (Source Program) รหัสนิโมนิกชุดคำสั่งของไมโคร คอนโทรลเลอร์จะได้รับการแปลไปเป็นภาษาเครื่องและนำไปเก็บไว้ยังอีกโปรแกรมหนึ่ง ซึ่งจะเรียกกัน ว่า โปรแกรมออบเจกต์ (Object Program) โปรแกรมออบเจกต์นี้สามารถนำไปใช้งานต่อไปได้อีกหลาย ลักษณะ เช่น นำไปรวมกับโปรแกรมออบเจกต์อื่นๆ โดยใช้โปรแกรมลิงเกอร์(Linker Program) หรืออาจ จะสร้างเป็นโปรแกรมรหัสเลขฐานสิบหก (Hexadecimal Format) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณลักษณะเพิ่มเติมของโปรแกรมแอสเซมเบลอร์

โดยปกติแล้วหน้าที่ของโปรแกรมแอสเซมเบลอร์ ก็จะมีหน้าที่เฉพาะการแปลความจากรหัสนี้ ไม่นิกของชุดคำสั่งไมโครคอนโทรลเลอร์และวีจิสเตอร์ไปเป็นรหัสเลขฐานสองซึ่งเทียบเท่ากัน แต่ในระยะต่อมาได้มีการเพิ่มเติมคุณลักษณะให้กับโปรแกรมมากขึ้น ตัวอย่างเช่น

1. ยินยอมให้ผู้ใช้สามารถใช้ชื่อที่ตั้งขึ้นมาเพื่ออ้างถึงตำแหน่งของหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุต แทนการใช้หมายเลขแอดเดรสแต่เพียงลักษณะเดียว
2. ยินยอมให้มีการใช้ชื่อของตัวอักษร เพื่อแทนกลุ่มของชุดคำสั่งหลายๆ คำสั่ง
3. สามารถเพิ่มการคำนวณทางคณิตศาสตร์เข้าไปในการเขียนคำสั่งรหัสนี้ได้อีก
4. ยินยอมให้ผู้ใช้สามารถควบคุมรูปแบบการแสดงผลลัพธ์ ตลอดจนกำหนดเงื่อนไขในการแอสเซมเบลอร์ได้

ข้อดีของโปรแกรมภาษาแอสเซมบลี

ลักษณะที่เป็นอุปสรรคสำคัญในการพัฒนาสร้างโปรแกรมด้วยภาษาแอสเซมบลี คือ จะต้องมีความรู้และความเข้าใจเกี่ยวกับสถาปัตยกรรมของไมโครคอนโทรลเลอร์ที่ใช้งานอยู่ อาทิเช่น ในบรรดาชุดคำสั่งต่างๆ มีหน้าที่หรือการทำงานอย่างไร และเมื่อทำงานเสร็จสิ้นจะมีผลต่อวีจิสเตอร์ใดบ้าง นอกจากนี้ยังจะต้องทราบถึงวิธีการกำหนดแอดเดรสเพื่อบอกตำแหน่งที่ติดต่อกัน เป็นต้น ซึ่งหากว่าเป็นการพัฒนาโปรแกรมภาษาอื่นๆ ที่มีอยู่บนเครื่องไมโครคอมพิวเตอร์ ก็ไม่จำเป็นต้องให้ความสนใจในสิ่งต่างๆ เหล่านี้เลย อุปสรรคอีกประการหนึ่งซึ่งเป็นลักษณะของโปรแกรมแอสเซมบลีเองซึ่งไม่มีความสามารถของการโอนย้ายโปรแกรม (Non-portable Program) กล่าวคือโปรแกรมแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูลใด ก็จะไม่สามารถที่จะนำไปใช้งานได้กับตระกูลอื่นๆ ได้เนื่องจากมีลักษณะสถาปัตยกรรมและรหัสนี้ไมนิกที่แตกต่างกันดังได้กล่าวในหัวข้อที่ผ่านมาแล้วข้างต้น

4-4 ขั้นตอนของการพัฒนาโปรแกรม

เครื่องไมโครคอมพิวเตอร์เป็นเครื่องมือช่วยพัฒนา เริ่มต้นผู้ใช้ควรจะต้องทำการเขียนโปรแกรม

มาก่อนล่วงหน้าบนกระดาษ และนำมาจัดเก็บเป็นไฟล์ในเครื่องคอมพิวเตอร์ด้วยการพิมพ์ เข้าไปอีกคู่หนึ่ง โปรแกรมบนเครื่องคอมพิวเตอร์ที่ได้รับการออกแบบมาเพื่อให้เราสามารถพิมพ์ข้อความและนำไปใช้ภายในเครื่องคอมพิวเตอร์ได้นั้น เรียกว่า โปรแกรมเอดิเตอร์ (Editor) และโปรแกรมชุดคำสั่งที่เราสร้างขึ้นเพื่อสั่งงานไมโครคอนโทรลเลอร์นั้น จะเรียกว่าโปรแกรมต้นฉบับ (Source Program) ซึ่งจะถูกนำไปใช้งานโดยโปรแกรมอื่นๆ ต่อไป

เนื่องจากชุดคำสั่งที่เราเขียนเพื่อสั่งงานนั้นยังไม่สามารถนำไปใช้สั่งงานไมโครคอนโทรลเลอร์ได้โดยตรง จำเป็นต้องทำการเปลี่ยนรูปแบบคำสั่งดังกล่าวไปเป็นรหัสตัวเลขภาษาเครื่อง ในไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบรหัสข้อมูลเลขฐานสอง หรือเลขฐานสิบหก โดยใช้โปรแกรมแอสเซมเบลอร์ ซึ่งในบางครั้งขั้นตอนการทำงานนี้อาจจะต้องการโปรแกรมประเภทลิงเกอร์ (Linker) เพื่อทำการเชื่อมโยงข้อมูลบางอย่างร่วมด้วย ผลที่ได้รับภายหลังจากการแปลความนี้จะได้โปรแกรมออกมาหลายประเภท เพื่อนำไปใช้ประโยชน์ต่างๆ กัน เช่น โปรแกรมรูปแบบข้อมูลเลขฐานสิบหก (Hex Format Program) หรือ โปรแกรมภาษาเครื่องแบบเลขฐานสอง เป็นต้น

โดยปกติแล้ว ก่อนที่จะนำโปรแกรมภาษาเครื่องที่ได้ไปทดลองในระบบจริง ก็อาจจะสามารถใช้โปรแกรมซิมูเลเตอร์ (Simulator) เป็นโปรแกรมช่วยงานประเภทหนึ่งที่จะทำการจำลองแบบการทำงานของไมโครคอนโทรลเลอร์บนเครื่องไมโครคอมพิวเตอร์ ซึ่งจะมีประโยชน์อย่างมาก เนื่องจาก จะทำให้สามารถทดสอบการทำงานของโปรแกรมได้ โดยไม่จำเป็นต้องนำไปทดสอบบนระบบที่แท้จริงเลย ความสามารถของโปรแกรมซิมูเลเตอร์นี้จะช่วยให้ผู้ใช้สามารถติดตามการทำงานของโปรแกรมที่สร้างขึ้นได้อย่างใกล้ชิด สามารถทำการตรวจดูค่าหรือสถานะต่างๆ ภายในไอซีไมโครคอนโทรลเลอร์ได้ นอกจากนี้ยังมีความสามารถที่สำคัญอีกอย่างหนึ่ง คือการกำหนดจุดหยุด (Breakpoints) เมื่อไมโครคอนโทรลเลอร์ได้ประมวลผลมาถึงแอดเดรสที่เราระบุไว้ก็จะหยุดการทำงานโดยอัตโนมัติ ซึ่งเราก็จะมักตรวจดูค่าของตัวแปรภายในหน่วยความจำ หลังจากนั้นก็สามารถสั่งให้โปรแกรมทำงานที่ตำแหน่งหลังจากบรรทัด ที่มีการหยุดค้างนั้นไว้ต่อไปได้ ในกรณีที่มีการใช้วงจรราร์ดแวร์เข้ามาประกอบกับโปรแกรมซิมูเลเตอร์นี้ จะเป็นลักษณะเครื่องมืออีกประเภทหนึ่งที่เรียกว่า อิมูเลเตอร์ (Emulator) ซึ่งแทนที่จะเป็นการทดลองแบบการทำงาน ก็สามารถรับ หรือ ประมวลผลข้อมูลจากสภาพการทำงานจริงได้ทันที

หลังจากที่โปรแกรมที่สร้างขึ้นได้รับการตรวจสอบการทำงานแล้วว่าถูกต้อง และเป็นไปตามความต้องการแล้ว ก็จะต้องนำโปรแกรมนี้ไปบรรจุไว้ในไอซีหน่วยความจำ EPROM โดยใช้เครื่องมือพิเศษที่ได้รับการออกแบบมาเพื่อทำหน้าที่ดังกล่าวนี้ เรียกว่า เครื่องโปรแกรมอีพ롬 (EPROM Programmer) ซึ่งก็มักจะนำข้อมูลภายในโปรแกรมภาษาเครื่องทั้งแบบที่เป็นข้อมูลเลขฐานสองหรือเลขฐานสิบหกมาจัดเก็บภายใน EPROM ซึ่งสามารถนำไปใส่ภายในระบบที่ออกแบบไว้ได้ตามความต้องการ

4-5 โครงสร้างของโปรแกรมภาษาแอสเซมบลี

การเขียนโปรแกรมภาษาแอสเซมบลีในแต่ละบรรทัด จะต้องปฏิบัติตามรูปแบบหรือกฎเกณฑ์ที่ได้มีการกำหนดไว้ เพื่อที่ว่าโปรแกรมแอสเซมเบลอร์จะสามารถทำการแปลงความได้อย่างถูกต้องซึ่งโดยปกติจะประกอบด้วยกลุ่มของข้อมูลจำนวนสี่กลุ่ม หรือกล่าวอีกลักษณะว่าจัดแบ่งออกเป็นสี่คอลัมน์ (Column) การแยกช่องคอลัมน์นี้สามารถกระทำได้ ทั้งการใช้ช่องว่าง (Space) จำนวนหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวหรือมากกว่านั้น หรืออาจจะเป็นอักขรแท็บ (Tab) ได้ตามต้องการ ลองพิจารณารูปแบบตัวอย่างของบรรทัดในโปรแกรมภาษาแอสเซมบลีต่อไปนี้

```
NAME: MOV A,10H ;assign a value to register A from memory
```

ในคอลัมน์แรกจะเป็นกลุ่มตัวอักษร (Symbolic Name field) ซึ่งจากตัวอย่างนี้เป็นกลุ่มอักษร Name โปรแกรมแอสเซมเบลอร์จะทราบได้ว่าสัญลักษณ์นี้จะต้องเป็นชื่อของตำแหน่งแอดเดรสที่เก็บชุดของคำสั่ง MOV A,10H นี้ ซึ่งคำศัพท์ที่เรามักจะใช้กันจะเรียกกันว่า ลาเบล (Label) ดังนั้นการใช้ลาเบลก็เพื่อจุดประสงค์ในการอ้างอิงถึงบรรทัดใดบรรทัดหนึ่งของโปรแกรมตัวอย่างเช่น หากมีชุดคำสั่งในบรรทัดที่เขียนว่า

```
JMP NAME
```

ก็มีความหมายให้ 8051 กระโดด (ตามคำสั่ง Jump) มายังคำสั่งแอดเดรสของ NAME นี้ในบางครั้งเราอาจจะพบว่า มีบางบรรทัดภายในโปรแกรมนั้นเว้นว่างไว้ ไม่มีเขียนข้อความใดๆไว้เลยก็ได้ โดยทั่วไปกฎเกณฑ์ที่ระบุไว้สำหรับการตั้งชื่อของลาเบล จะใช้อักษรตัวแรกได้เฉพาะตัวอักษรหรือตัวอักษรพิเศษบางตัว เช่น @ หรือ ? นอกจากนี้แล้วชื่อที่ตั้งขึ้นมานี้จะต้องไม่ไปซ้ำกับชุดคำสั่งหรือชื่อของรีจิสเตอร์ ซึ่งชื่อพวกนี้ถือเป็นคำสงวน (Reserved words) ไม่สามารถนำไปตั้งชื่อของลาเบล

คอลัมน์ที่สองจะเป็นส่วนที่บอกให้แอสเซมเบลอร์ทราบว่าต้องดำเนินการอย่างไร (Action field) ซึ่งตามตัวอย่างนี้เป็นข้อความว่า MOV ซึ่งตรงกับรหัสนี้โมนิกชุดคำสั่งของ 8051 เพื่อที่จะทำการโอนย้ายข้อมูล ในบางครั้งในคอลัมน์นี้อาจเป็นข้อมูลอื่นๆที่ไม่ตรงกับรหัสนี้โมนิกชุดคำสั่งของ 8051 ก็ได้ เช่น คำแนะนำในการแปลความสำหรับแอสเซมเบลอร์ในการแปลความเป็นต้น ซึ่งรายละเอียดในส่วนนี้จะได้กล่าวถึงต่อไปในหัวข้อถัดไป

คอลัมน์ที่สามจะเป็นส่วนที่เรียกว่า โอเปอเรนด์ (Operand Filed) ซึ่งมีความหมายถึงส่วนเสริมของข้อมูลในคอลัมน์ที่ผ่านมา ตามตัวอย่างข้อมูลของคอลัมน์นี้ประกอบด้วยโอเปอเรนด์สองตัว คือ A และ 10 H ทำให้ 8051 ทราบได้ว่าต้นทางของข้อมูลที่จะทำการโอนย้ายอยู่ที่หน่วยความจำ 10H และปลายทางที่จะนำข้อมูลไปเก็บเป็นรีจิสเตอร์ A และขอให้สังเกตว่ากรณีนี้มีโอเปอเรนด์สองตัวจะต้องแยกกันด้วยเครื่องหมายคอมมา (Comma) ซึ่งใช้สัญลักษณ์เป็นเครื่องหมาย โดยไม่ช่องว่างแทรกอยู่ระหว่างโอเปอเรนด์ทั้งสอง สำหรับคำสั่งของ 8051 นั้นอาจจะมีจำนวนของโอเปอเรนด์ได้ตั้งแต่ 0 ถึง 2 ตัว ตัวอย่างเช่น คำสั่ง RET จะไม่มีโอเปอเรนด์เลย

สำหรับคอลัมน์สุดท้ายเรียกว่า คอมเมนต์ (Comment) ซึ่งจะเริ่มต้นด้วยอักษร Semicolon ใช้สัญลักษณ์เป็น ; ดังนั้นเมื่อแอสเซมเบลอร์พบเครื่องหมายนี้ที่ใดก็จะไม่แปลความข้อมูลที่เหลือภายในบรรทัดนั้นอีกเลย จุดประสงค์ของการมีส่วนคอมเมนต์นี้เพื่อเป็นการอธิบายให้กับผู้เขียนโปรแกรมเองหรือผู้อื่นที่จะมาศึกษาการทำงานของโปรแกรมในภายหลังให้สามารถทำความเข้าใจได้โดยง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดที่เกี่ยวข้องกับการสร้างโปรแกรม

ในการสร้างเขียนโปรแกรมภาษาแอสเซมบลี มีข้อควรคำนึงบางประการที่จำเป็นต้องให้ความสนใจเกี่ยวกับชื่อของตัวอักษร สัญลักษณ์ หรือตัวเลขแอดเดรสที่อยู่ในส่วนของโอเปอเรนด์ดังต่อไปนี้

ค่าของตัวเลข

โดยปกติแล้วหากมีค่าของตัวเลขโดยลำพัง ก็จะมีความหมายถึงตัวเลขฐานสิบ (Decimal Numbers) แต่หากต้องการที่ใช้ตัวเลขในระบบเลขฐานอื่น ก็จะต้องต่อท้ายตัวเลขชุดนั้นด้วยตัวอักษรหนึ่งตัวต่อไปนี้

หมายเหตุ ขอให้สังเกตว่าในการใช้ตัวเลขที่เป็นฐานสิบหกนั้น หากว่าเลขตัวแรกเป็นค่าตั้งแต่ A ถึง F แล้ว มักจะต้องนำหน้าตัวเลขชุดนี้ด้วยตัวเลขศูนย์ ทั้งนี้เพื่อหลีกเลี่ยงความสับสนระหว่างค่าตัวเลขกับชื่อของลาเบล

การอ้างถึงตำแหน่งปัจจุบันของโปรแกรมเคาน์เตอร์

สัญลักษณ์ตัวอักษร \$ (Dollar sign) ที่ปรากฏอยู่ในส่วนของโอเปอเรนด์ เป็นการแทนความหมายของค่าปัจจุบันของตำแหน่งแอดเดรสที่เก็บอยู่ในรีจิสเตอร์ PC (โปรแกรมเคาน์เตอร์) ซึ่งจะพบกันมากในการใช้ร่วมกับชุด คำสั่งที่เกี่ยวข้องกับการกระโดด ดังตัวอย่าง

SJMP \$

ซึ่งมีความหมายถึง ให้โปรแกรมกระโดดไปยังตำแหน่ง ของแอดเดรสที่เก็บคำสั่งไบต์แรกในบรรทัดปัจจุบัน (คำสั่ง SJMP) หากจะเทียบเท่ากับการเขียนในลักษณะปกติจะมีลักษณะดังนี้

HERE: SJMP HER

ดังนั้น ผลการทำงานของคำสั่งจะเหมือนกับการวนรอบไม่รู้จบ (Endless Loop)

4-6 โครงสร้างภายในของ 8051

ภายในไมโครคอนโทรลเลอร์ 8051 ซึ่งประกอบด้วยหน่วยการทำงานต่างๆ ภายในไอซี 8051 จำนวนมาก โดยแต่ละบล็อกซึ่งเป็นวงจรควบคุม รีจิสเตอร์(Register) หรือ หน่วยความจำภายในของ 8051 จะถูกเชื่อมต่อเข้าด้วยกันผ่านทางเส้นสัญญาณที่เรียกว่า บัสข้อมูลภายใน รีจิสเตอร์และหน่วยความจำเหล่านี้จะถูกนำไปใช้ในระหว่างการประมวลผลคำสั่ง หน้าที่ของโปรแกรมที่ผู้ใช้สร้างขึ้นมา ก็เป็นการควบคุมการรับหรือส่งข้อมูลระหว่างรีจิสเตอร์เหล่านี้ ซึ่งอาจจะมีการดำเนินการร่วมกับหน่วยการดำเนินงานประมวลผลทางคณิตศาสตร์และลอจิก (ALU: Arithmetic and Logic Unit) ด้วย

4-6-1 รีจิสเตอร์ภายใน

ในความเป็นจริงแล้ว รีจิสเตอร์อาจจะพิจารณาได้ว่าเป็นหน่วยความจำ RAM ประเภทหนึ่งที่อยู่ภายในไมโครคอนโทรลเลอร์ และสามารถนำข้อมูลเข้าไปเก็บไว้ภายในรีจิสเตอร์ หรืออ่านค่าข้อมูลนี้ออกมาในภายหลังได้เช่นเดียวกับหน่วยความจำทุกประการ ความแตกต่างระหว่างหน่วยความจำและเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่เป็นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ที่สำคัญเป็นลักษณะของ การติดต่อจากหน่วยประมวลผลกลาง โดยหากว่าเป็นการติดต่อกับ รีจิสเตอร์ก็มักจะใช้ชุดคำสั่งที่เกี่ยวข้องกับชื่อของรีจิสเตอร์นั้นโดยตรง ไม่จำเป็นต้องให้ความสนใจว่า ตำแหน่งของรีจิสเตอร์นั้นจะอยู่ที่บริเวณใด ซึ่งเป็นลักษณะที่ตรงข้ามกับหน่วยความจำภายในที่จะต้องระบุถึงตำแหน่งโดยชัดเจน นอกจากนี้แล้วยังอาจมีชุดคำสั่ง บางอย่างที่สามารถดำเนินการได้เฉพาะ กับรีจิสเตอร์เท่านั้นด้วย หากจะจำแนกกลุ่มของรีจิสเตอร์ตามลักษณะของการนำไปใช้งานแล้วจะแยก ออกได้เป็นสามลักษณะ คือ รีจิสเตอร์ที่เกี่ยวข้องกับแอดเดรสรีจิสเตอร์ใช้งานทั่วไป และรีจิสเตอร์ฟังก์ชัน หน้าที่พิเศษ

รีจิสเตอร์ที่เกี่ยวข้องกับแอดเดรส (Address Registers)

รีจิสเตอร์ในกลุ่มนี้จะเป็นรีจิสเตอร์ขนาด 16 บิตที่ใช้งานเพื่อเก็บข้อมูลแอดเดรสเป็นสำคัญโดย ค่าที่อยู่ภายในแอดเดรสนี้ จะนำไปเป็นค่าของข้อมูลที่ส่งออกไปทางบัลลแอดเดรสของไมโคร คอนโทรลเลอร์ เพื่อบอกตำแหน่งที่ต้องการติดต่อ รีจิสเตอร์ที่จัดในกลุ่มนี้ประกอบด้วย

โปรแกรมเคาน์เตอร์ (Program Counter)

เป็นรีจิสเตอร์ที่ใช้ในการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรมซึ่งจะต้องไปนำคำสั่ง มาประมวลผลในลำดับถัดไป โดยเมื่อถึงจังหวะเวลาที่ 8051 ต้องการคำสั่งที่เก็บไว้ภายในหน่วยความ จำแล้ว หน่วยประมวลผลของ 8051 ก็จะส่งค่าข้อมูลที่อยู่ภายในรีจิสเตอร์นี้ออกมายังบัลลแอดเดรส ซึ่ง หน่วยความจำที่ต่ออยู่กับบัลลนี้ก็จะรับทราบ และนำค่าที่เก็บอยู่ในส่วนของหน่วยความจำ ในตำแหน่ง ที่ต้องการส่งกลับมาให้กับ 8051 ภายหลังจากนี้แล้วค่าข้อมูลภายในรีจิสเตอร์ นี้จะเพิ่มขึ้นโดยอัตโนมัติ เพื่อบอกตำแหน่งของคำสั่งถัดไป การใช้งานภายในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

สแต็กพอยน์เตอร์ (Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งแอดเดรสหน่วยความจำของบริเวณที่ใช้งาน เป็นสแต็ก (Stack) สำหรับเก็บข้อมูลแอดคิวมูลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม (ซึ่ง ลักษณะการทำงานของสแต็กจะกล่าวโดยละเอียดต่อไป) โดยปกติแล้วเมื่อทำการเริ่มต้นระบบใหม่ภาย หลังจากรับจ่ายไฟฟ้าหรือมีการรีเซ็ตเกิดขึ้น ค่าภายในสแต็กพอยน์เตอร์จะเป็นค่า 07H ซึ่งเป็นแอด เดรสภายในบริเวณเนื้อที่ 128 ไบต์แรกของหน่วยความจำภายใน การใช้งานภายในโปรแกรมเรียกว่า รีจิสเตอร์ SP

ดาต้าพอยน์เตอร์ (Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรี จิสเตอร์ขนาด 8 บิตจำนวนสองตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าของแอดเดรสของ หน่วยความจำหรืออุปกรณ์ที่จะต้องใช้งานภายในโปรแกรม การใช้รีจิสเตอร์ DPTR นี้ นับว่ามีประโยชน์ เนื่องจากทำให้หน่วยประมวลผลกลางสามารถใช้เทคนิคทางฮาร์ดแวร์ได้

รีจิสเตอร์ใช้งานทั่วไป (General-Purpose Registers)

รีจิสเตอร์ในกลุ่มนี้จัดเป็นพื้นที่หน่วยความจำที่ใช้ในการสนับสนุนในการประมวลผลการทำงาน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า จากหน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) เพื่อให้สามารถจัดการข้อมูลให้เร็วที่สุดนอก ไม่วารณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนี้โปรแกรมที่ไม่ได้ใช้คำสั่งเหล่านี้ก็ยังใช้เป็นการเก็บข้อมูลตัวแปรภายในโปรแกรมได้รีจิสเตอร์ในกลุ่มนี้มีขนาดเพียง 8 บิตมีชื่อเรียกว่า รีจิสเตอร์ R0,R1,R2,R3,R4,R5,R6 และ R7 ซึ่งตำแหน่งของรีจิสเตอร์เหล่านี้อยู่ภายในบริเวณของหน่วยความจำข้อมูลภายใน (Internal RAM) เช่นบริเวณแอดเดรส 00H ถึง 07H โดยจำแนกออกเป็นกลุ่ม (หรือ BANK) ของกลุ่มรีจิสเตอร์

จะเห็นได้ว่าชื่อของรีจิสเตอร์ไม่ว่าจะอยู่ในรีจิสเตอร์แบงก์ใด ก็จะมีชื่อ R0 ถึง R7 เหมือนกันทั้งสิ้น ดังนั้นในการใช้งานผู้ใช้จะต้องให้ความระมัดระวังว่า ต้องการรีจิสเตอร์นั้นๆจากแบงก์ใด ซึ่งการกำหนดเลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในรีจิสเตอร์ PSW เท่านั้น

อย่างไรก็ตามโดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0 - R7 เฉพาะในแบงก์ 0 เท่านั้น ดังนั้นพื้นที่ของแบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วยความจำ RAM เช่นปกติ

รีจิสเตอร์หน้าที่พิเศษ

รีจิสเตอร์หน้าที่พิเศษ(SFR)เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่และการทำงานของอุปกรณ์หรือพอร์ตของ 8051 ทั้งหมด ตำแหน่งของรีจิสเตอร์เหล่านี้จะจัดอยู่ในบริเวณแอดเดรส 80H - FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้ สามารถทำได้ทั้งการระบุถึงชื่อของรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้ ในรูปที่ 3.2 (b) ได้แสดงให้เห็นถึงการจัดพื้นที่หน่วยความจำสำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อสังเกตว่ารีจิสเตอร์ที่อยู่ในตำแหน่งแอดเดรสที่เป็นจำนวนทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H 88H 90H A0H A8H B0H B8H D0H E0H และ F0H)

แอกคิวมูเลเตอร์ (Accumulator) หรือ ACC

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การทำงานของรีจิสเตอร์นี้ มีลักษณะเช่นเดียวกับตัวแอกคิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานภายในโปรแกรมจะเรียกเป็น รีจิสเตอร์ A

รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณและหารตัวเลข ในกรณีที่ไม่ใช่ในการคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

โปรแกรมสแตตัสเวิร์ด (PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์ (Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต/เอาต์พุตโดยตรง ซึ่งแต่ละตัวจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะของการอินพุตหรือการเอาต์พุตข้อมูลได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูอาจารย์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ในสื่อออนไลน์ การดำเนินการใดๆ ที่เกี่ยวข้องกับพอร์ตทั้งสิ้นจะมีผลทำให้ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนไม่ทราบใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไขคัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้

แปลงไปเช่นกัน นอกจากนี้พอร์ต P0 และ P2 ยังสามารถนำมาใช้ในการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเดรส 8 บิตบนของหน่วยความจำ ส่วนพอร์ต P0 นั้นในช่วงเริ่มแรกจะเป็นค่าของแอดเดรส 8 บิตล่างของหน่วยความจำ ช่วงเวลาต่อมาจึงจะนำพอร์ต P0 ไปใช้เป็นบัสสำหรับการรับหรือส่งข้อมูลกับหน่วยอุปกรณ์ภายนอก สำหรับพอร์ต P3 นั้นนอกเหนือจากจะใช้ในฐานะของพอร์ตอินพุต/เอาต์พุตเช่นปกติแล้ว ยังนำมาใช้ในฐานะของบัสควบคุมเกี่ยวกับสัญญาณอินเตอร์รัปต์ได้อีกด้วย

รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์นี้มีอยู่ด้วยกันสองชุดและแยกจากกันอย่างชัดเจน สำหรับการส่งและการรับโดยซีพียูจะทำการจัดการเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมหน้าที่การทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรเซสเซอร์ (บิต IDL และ PD) กำหนดอัตราการทวิตคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรม (บิต SMOD) และแฟล็กสถานะสำหรับการใช้งานทั่วไป (บิต GR0 และ GR1)

4-6-2 หน่วยประมวลผลทางคณิตศาสตร์และลอจิก

ในการทำชุดคำสั่งที่เกี่ยวข้องกับการคำนวณทางคณิตศาสตร์ หรือการดำเนินการแบบลอจิกนั้น 8051 จะดำเนินการโดยใช้หน่วยประมวลผลทางคณิตศาสตร์และลอจิก (Arithmetic and Logic Unit) หรือที่เรียกกันว่า ALU ซึ่งจะมีรีจิสเตอร์ที่เข้ามาเกี่ยวข้องด้วย คือ รีจิสเตอร์เก็บข้อมูลชั่วคราว (Temporary Register) เพื่อเก็บข้อมูลตัวเลขที่จะมาดำเนินการนั้น ดังนั้น เมื่อเราได้สั่งคำสั่งที่ต้องดำเนินการโดย AUL ก็มักจะส่งค่าของข้อมูลมาทางรีจิสเตอร์ A และรีจิสเตอร์อื่นๆ ด้วยเสมอ จากนั้นหน่วยประมวลผลกลางของ 8051 จะเป็นตัวดำเนินการเพื่อย้ายข้อมูลเข้าไปยังรีจิสเตอร์เก็บข้อมูลชั่วคราวนี้โดยอัตโนมัติ และเมื่อเสร็จสิ้นการดำเนินการแล้ว จึงจะนำผลลัพธ์ที่ได้ออกมาเก็บไว้ในรีจิสเตอร์ A เพื่อให้ผู้ใช้งานไปประมวลผลในโปรแกรมต่อไป นอกจากนี้ 8051 ยังมีคำสั่งที่เกี่ยวข้องกับการคูณหรือการหารด้วย ซึ่งก็มีการดำเนินการที่เหมือนกันเพียงแต่จะต้องนำข้อมูลมาไว้ในรีจิสเตอร์ A และรีจิสเตอร์ B เท่านั้น

รีจิสเตอร์อีกตัวหนึ่งที่ข้อมูลของมันเป็นผลมาจากการทำงานของหน่วย ALU คือ รีจิสเตอร์ PSW ซึ่งทำหน้าที่เก็บสถานะที่เกี่ยวข้องกับผลลัพธ์จากการดำเนินการทางคณิตศาสตร์หรือลอจิกเราสามารถพิจารณาได้ว่าเป็นกลุ่มของแฟล็ก (Flags) เพื่อบอกผลของการทำคำสั่ง ตัวอย่างเช่น การบวกค่าตัวเลขและมีการทดค่าเกิดขึ้น ก็จะมีแฟล็กทด (Carry Flag) ที่มีสถานะตามการทดตัวเลขนั้น เป็นต้น

เอกสาร 4-6-3 หน่วยความจำของ 8051 ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ตระกูล 8051 แยกการจัดการหน่วยความจำออกเป็นสองส่วนอย่างชัดเจน คือ หน่วยความจำโปรแกรม (Program Memory) และหน่วยความจำข้อมูล (Data Memory) หน่วยความจำทั้งสองนี้ มีหน้าที่ที่แตกต่างไปจากกัน และใช้วิธีการอ้างแอดเดรส สัญญาณการติดต่อแยกออกจากกัน

หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมของ 8051 เป็นบริเวณหน่วยความจำ สำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบ ข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหายโครงสร้างของหน่วยความจำโปรแกรม มีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วย ความจำประเภทต่างๆ เช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory) เป็นต้น

8051 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์เอง และหน่วยความจำโปรแกรมภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

ขนาดหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์เบอร์ต่างๆ ภายในตระกูล 8051 จะมีความแตกต่างกันออกไป เพื่อความเหมาะสมกับการนำไปใช้งานลักษณะต่างๆ กัน เช่น

8051 และ 8052 มีหน่วยความจำแบบ ROM ขนาด 4 และ 8 กิโลไบต์ ตามลำดับ

8751 มีหน่วยความจำแบบ EPROM ขนาด 4 กิโลไบต์ ข้อมูลที่จัดเก็บภายในนี้สามารถใช้แสงอัลตราไวโอเล็ตลบและนำกลับไปบรรจุโปรแกรมใหม่ได้อีกครั้งหนึ่ง

8031 และ 8032 ไม่มีหน่วยความจำโปรแกรมอยู่ภายในตัวไอซีเลย ดังนั้นในการนำไปใช้งานจึงจำเป็นต้องอาศัยหน่วยความจำโปรแกรมภายนอกเสมอ

ไมโครคอนโทรลเลอร์เบอร์ต่างๆ ของตระกูล 8051 นี้ สามารถขยายให้ใช้งานหน่วยความจำภายนอกได้ทั้งสิ้น โดยกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้ว การอ้างตำแหน่งแอดเดรสที่มีทั้งในหน่วยความจำโปรแกรมภายในและภายนอกนั้น จะต้องทำการควบคุมระดับลอจิกของสัญญาณ EA ในขณะนั้นด้วย

หน่วยความจำข้อมูล

หน่วยความจำข้อมูล (Data Memory) ซึ่งโดยพื้นฐานแล้วเป็นหน่วยความจำ RAM สามารถเขียนหรืออ่านข้อมูลได้ (Read or Write Memory) ใช้สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว ซึ่งโดยพื้นฐานแล้วหน่วยความจำข้อมูลจัดเป็นหน่วยความจำ RAM แบบสแตติก ดังนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในหน่วยความจำนี้สูญไป พื้นที่ของหน่วยความจำข้อมูลของ 8051 นี้สามารถมีได้ สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น ตามลักษณะแผนภาพในรูปที่ 3.7 คือ หน่วยความจำโปรแกรมภายใน (Internal Data Memory) ซึ่งก็เป็น RAM ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำ ไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

4-7 ชุดคำสั่งของ 8051

ไมโครคอนโทรลเลอร์ 8051 ประกอบด้วยคำสั่งทั้งหมดจำนวนมาก ซึ่งนำมาแสดงไว้ในตารางของภาคผนวก ก ซึ่งสามารถจะจัดกลุ่มคำสั่งเหล่านี้ตามลักษณะหน้าที่การทำงานที่คล้ายคลึงกัน เพื่อความสะดวกต่อการศึกษา ทำความเข้าใจและใช้งาน ดังนี้

1. กลุ่มคำสั่งในการโอนย้ายข้อมูลทำหน้าที่ในการโอนย้ายข้อมูลระหว่างรีจิสเตอร์หรือหน่วยความจำ
2. กลุ่มคำสั่งดำเนินการแบบลอจิก ทำหน้าที่เกี่ยวกับการประมวลผลแบบลอจิกต่างๆ เช่น การ AND OR หรือ EX-OR ระหว่างข้อมูลในรีจิสเตอร์ A กับรีจิสเตอร์ทั่วไป และการทำคำสั่งเกี่ยวกับการเลื่อนหรือหมุนบิตข้อมูลภายในรีจิสเตอร์ A เป็นต้น
3. กลุ่มคำสั่งในการกระโดดไปยังตำแหน่งต่างๆ ภายในโปรแกรม ซึ่งจะเปลี่ยนลำดับของการประมวลผลภายในโปรแกรมไปยังส่วนต่างๆ แทนที่จะดำเนินการไปเป็นลำดับต่อเนื่องเช่นปกติ ซึ่งสามารถจะกระทำการกระโดดได้ทั้งแบบมีหรือไม่มีเงื่อนไข นอกจากนั้นกลุ่มคำสั่งนี้ยังรวมถึงการเรียกใช้งานโปรแกรมย่อย และการกลับมาทำโปรแกรมหลักด้วย
4. กลุ่มคำสั่งการคำนวณทางคณิตศาสตร์ เช่น การบวก ลบ คูณ และหารข้อมูลภายในรีจิสเตอร์ต่างๆ
5. กลุ่มคำสั่งการประมวลผลแบบบิต ซึ่งเป็นความสามารถของ 8051 ที่จะดำเนินการประมวลผลแบบบิต แทนที่จะเป็นข้อมูลทั้งไบต์เช่นปกติ โดยมีชุดคำสั่งที่จัดการโดยตรง

สำหรับค่าเลขฐานสิบหกของรหัสนิมิกชุดคำสั่งทั้งหมดนั้น ได้แสดงไว้ในภาคผนวก ข พร้อมกันนี้ได้แสดงจำนวนไบต์ข้อมูลที่จะเกิดขึ้น หากใช้รูปแบบของคำสั่งนิมิกในแต่ละบรรทัดซึ่งมีตั้งแต่ 1-3 ไบต์ สำหรับผู้ที่ใช้ในการป้อนข้อมูลโปรแกรมโดยการเปิดตารางและพิจารณาค่าของเลขฐานสิบหกจากแต่ละบรรทัดของโปรแกรมเอง โดยไม่ได้มีการใช้โปรแกรมแอสเซมเบลอร์ ขอให้ศึกษารายละเอียดเพิ่มเติมจากคู่มือใช้งานจากบริษัทผู้ผลิตโดยตรง สำหรับการศึกษานี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาของการประมวลผลชุดคำสั่ง

เวลาในการประมวลผลคำสั่งหนึ่งจนเสร็จสิ้นของ 8051 นั้น จะนับเป็นหน่วยของ แมชชีนไซเคิล (Machin cycle) ซึ่งจากข้อมูลจำนวนของแมชชีนไซเคิลในตารางของภาคผนวก ก จะเห็นว่ามีค่าที่แตกต่างกันไป โดยค่าเวลา 1 แมชชีนไซเคิลนั้นจัดว่าเป็นช่วงเวลาที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่ง ซึ่งหากว่าเป็นคำสั่งที่ซับซ้อนมากก็จะต้องใช้เวลาานสองถึงสามแมชชีนไซเคิล

การคำนวณหาว่าเวลาที่ใช้ในการทำคำสั่งใดจนเสร็จสิ้น จะต้องดูว่าคำสั่งนั้นใช้จำนวนแมชชีนไซเคิลเป็นเท่าไรในการประมวลผลเวลาที่จะใช้คำนวณตามสูตร

รูปแบบการอ้างตำแหน่งแอดเดรสของ 8051

การอ้างตำแหน่งแอดเดรส (Addressing Modes) เป็นวิธีการที่ไม่ใครคอนโทรลเลอร์จะสามารถระบุตำแหน่งของข้อมูลที่จะดำเนินการ เมื่อมีการใช้ชุดคำสั่งต่างๆ ได้ (โอเปอร์เรนด์ต้นทางและปลายทาง) ลักษณะการอ้างแอดเดรสของ 8051 มีทั้งหมด 5 ลักษณะ คือ

1. การอ้างแอดเดรสจากค่าตำแหน่งทันที (Immediate Addressing)
2. การอ้างแอดเดรสจากรีจิสเตอร์ (Register Addressing)
3. การอ้างแอดเดรสจากหน่วยความจำโดยตรง (Direct Addressing)
4. การอ้างแอดเดรสโดยข้ามผ่านรีจิสเตอร์ (Register-indirect Addressing)
5. การอ้างแอดเดรสโดยข้ามผ่านค่าฐานภายในรีจิสเตอร์ร่วมกับค่าดัชนีในอีกรีจิสเตอร์ (Base-register Plus index register Addressing)

การอ้างแอดเดรสจากค่าตำแหน่งทันที (Immediate Addressing)

การอ้างแอดเดรสด้วยวิธีการนี้เป็นลักษณะพื้นฐานที่สุด ซึ่งบางครั้งการใช้คำสั่งที่มีการอ้างแอดเดรสแบบนี้อาจเรียกว่าเป็น คำสั่งที่อ้างอิงกับหน่วยความจำ (Memory Reference Instruction) โดยโอเปอร์เรนด์ที่ใช้กับชุดคำสั่งอาจจะเป็นตัวเลขหนึ่งหรือสองไบต์ เพื่อบอกค่าที่จะไปนำค่ามาเก็บไว้ในรีจิสเตอร์อย่างชัดเจน โดย 8051 มีการกำหนดไว้ว่าจะต้องมีเครื่องหมาย # (pound sign) นำหน้าค่าตัวเลขของโอเปอร์เรนด์ต้นทาง

ตัวอย่างคำสั่ง

```
MOV DPTR,#0B0CH
```

เป็นการบอกว่าข้อมูลต้นทางที่จะนำมาเก็บไว้ในรีจิสเตอร์ DPTE เป็นค่าข้อมูล 0B0CH โดยสามารถแสดงเป็นแผนภาพได้ดังนี้

การอ้างแอดเดรสจากรีจิสเตอร์ (Register Addressing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการอ้างตำแหน่งของข้อมูลที่จะดำเนินการมาจากรีจิสเตอร์ ดังนั้นจึงมีการใช้ชื่อของรีจิสเตอร์ ใน ส่วนของโอเพอร์แอนด์ต้นทางหรือปลายทางของข้อมูล อย่างไรก็ตามรีจิสเตอร์ที่จะสามารถใช้กับการ อ้างแอดเดรสลักษณะนี้จะมีเฉพาะ รีจิสเตอร์ A รีจิสเตอร์ DPTR และรีจิสเตอร์ R0 - R7 สำหรับรีจิส เตอร์ฟังก์ชันพิเศษอื่นๆ จะต้องใช้วิธีการอ้างแอดเดรสแบบอื่น คือ การอ้างแอดเดรสจากหน่วยความจำ โดยตรงเท่านั้น

ตัวอย่างคำสั่ง

```
MOV A,R0
```

เป็นการบอกว่าข้อมูลต้นที่จะนำมาเก็บไว้ในรีจิสเตอร์ R0 มาเก็บไว้ในรีจิสเตอร์ A โดยสามารถ แสดงเป็นแผนภาพได้ดังนี้

การอ้างแอดเดรสจากหน่วยความจำโดยตรง (Direct Addressing)

เป็นการอ้างตำแหน่งของข้อมูลที่จะดำเนินการ โดยให้หมายเลขตำแหน่งแอดเดรสของหน่วย ความจำมาโดยชัดเจน และกำหนดด้วยว่าจะต้องเป็นเฉพาะหน่วยความจำข้อมูลภายใน 8051 เท่านั้น (ซึ่งรวมถึงบริเวณรีจิสเตอร์ใช้งานพิเศษ (SFR) ด้วย) ทำให้ค่าตัวเลขของแอดเดรสที่ใช้ในการเขียนโอ เพอร์แอนด์ของชุดคำสั่งเป็นได้เพียงขนาดหนึ่งไบต์

แอดเดรสของหน่วยความจำข้อมูลภายใน 8051 มีค่าได้ตั้งแต่ 00H -7FH และมีค่าแอดเดรส หน่วยความจำที่เป็นรีจิสเตอร์ฟังก์ชันพิเศษอยู่ในช่วงตั้งแต่ 80H -FFH ดังแสดงในรูปที่ 4.1

ตัวอย่างคำสั่ง

```
MOV A,R0
```

เป็นการบอกว่าข้อมูลต้นที่จะนำมาเก็บไว้ในรีจิสเตอร์ R0 มาเก็บไว้ในรีจิสเตอร์ A โดยสามารถ แสดงเป็นแผนภาพได้ดังนี้

การอ้างแอดเดรสจากหน่วยความจำโดยตรง (Direct Addressing)

เป็นการอ้างตำแหน่งของข้อมูลที่จะดำเนินการ โดยให้หมายเลขตำแหน่งแอดเดรสของหน่วย ความจำมาโดยชัดเจน และกำหนดด้วยว่าจะต้องเป็นเฉพาะหน่วยความจำข้อมูลภายใน 8051 เท่านั้น (ซึ่งรวมถึงบริเวณรีจิสเตอร์ใช้งานพิเศษ (SFR) ด้วย) ทำให้ค่าตัวเลขของแอดเดรสที่ใช้ในการเขียนโอ เพอร์แอนด์ของชุดคำสั่งเป็นได้เพียงขนาดหนึ่งไบต์

แอดเดรสของหน่วยความจำข้อมูลภายใน 8051 มีค่าได้ตั้งแต่ 00H -7FH และค่าแอดเดรส หน่วยความจำที่เป็นของรีจิสเตอร์ฟังก์ชันพิเศษอยู่ในช่วงตั้งแต่ 80H -FFH ดังแสดงในรูปที่ 4.1

ตัวอย่างคำสั่ง

```
MOV A,20H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการบอกถึงตำแหน่งแอดเดรสที่จะไปนำข้อมูลต้นทางมา ซึ่งเป็นแอดเดรส 20H ของหน่วยความจำ ข้อมูลภายใน 8051 มาเก็บไว้ยังรีจิสเตอร์ A โดยสามารถแสดงเป็นแผนภาพได้ดังนี้

การอ้างแอดเดรสโดยอ้อมผ่านรีจิสเตอร์ (Register-indirect Addressing)

การอ้างแอดเดรสโดยวิธีการนี้ เป็นการนำค่าแอดเดรสของตำแหน่งหน่วยความจำที่ต้องการจัดเก็บไว้ภายในรีจิสเตอร์ ดังนั้นเมื่อมีความต้องการที่จะติดต่อกับหน่วยความจำก็เพียงแต่ใช้ค่าที่เก็บอยู่ในรีจิสเตอร์เท่านั้น การอ้างแอดเดรสในลักษณะเช่นนี้อาจเรียกได้ว่า รีจิสเตอร์ ทำหน้าที่เป็นพอยน์เตอร์ (Pointer) หรือตัวชี้ ไปยังแอดเดรสของหน่วยความจำ

รีจิสเตอร์ของ 8051 ที่สามารถใช้การอ้างแอดเดรสด้วยแบบนี้ได้จะเป็นรีจิสเตอร์ R0 หรือ R1 ซึ่งเป็นรีจิสเตอร์ขนาด 8 บิต ทำให้ตำแหน่งของหน่วยความจำที่ใช้เป็นเพียงค่าตั้งแต่ 0 - FFH แต่หากจะต้องการอ้างถึงหน่วยความจำได้ในขอบเขตที่มากกว่านี้ ก็จะต้องใช้รีจิสเตอร์ DPTR ซึ่งเป็นรีจิสเตอร์ขนาด 16 บิตแทน นอกจากนี้แล้วการใช้ รีจิสเตอร์เพื่อทำงานลักษณะนี้จะต้องนำหน้าชื่อของรีจิสเตอร์ด้วยเครื่องหมาย @ (อ่านว่า at sign)

ตัวอย่างคำสั่ง

```
MOV A,@R0
```

เป็นการบอกถึงตำแหน่งแอดเดรสที่จะไปนำข้อมูลต้นทางมา โดยใช้ค่าที่เก็บอยู่ในรีจิสเตอร์ R0 เป็นแอดเดรสต้นทางของหน่วยความจำมาเก็บไว้ยังรีจิสเตอร์ A โดยสามารถแสดงเป็นแผนภาพได้ดังนี้

การอ้างแอดเดรสโดยอ้อมผ่านค่าฐานภายในรีจิสเตอร์ร่วมกับค่าดัชนีในอีกรีจิสเตอร์ (Base-register Plus Index register Addressing)

การอ้างแอดเดรสทางอ้อมในอีกลักษณะหนึ่งของ 8051 ซึ่งใช้รีจิสเตอร์ทำหน้าที่เป็นตัวชี้ตำแหน่งข้อมูลเช่นกัน เพียงแต่มีการใช้รีจิสเตอร์ถึงสองตัวเข้ามามีในการกำหนดเป็นค่าแอดเดรสของตำแหน่งที่ต้องการ โดยรีจิสเตอร์ตัวหนึ่งจะทำหน้าที่เก็บค่าเริ่มต้นของหน่วยความจำ (Base-register) ซึ่งจะต้องไปรวมกับค่าขนาด 8 บิตที่อยู่ในรีจิสเตอร์อีกตัวหนึ่ง (Index register) วิธีการอ้างแอดเดรสแบบนี้ จะทำให้การเขียนโปรแกรมในลักษณะที่เป็นการติดต่อข้อมูลจากโครงสร้างที่เป็นตาราง (Lookup table) ทำได้โดยสะดวกมากขึ้น

ตัวอย่างคำสั่ง

```
MOV A,@A+DPTR
```

เป็นการบอกถึงตำแหน่งแอดเดรสที่จะไปนำข้อมูลต้นทางมา โดยคำนวณจากค่า 16 บิต ที่เก็บอยู่ในรีจิสเตอร์ DPTR บวกกับค่าที่เก็บอยู่ในรีจิสเตอร์ A เมื่อไปนำข้อมูลที่ตำแหน่งแอดเดรสนี้แล้วให้นำมาเก็บไว้ยังรีจิสเตอร์ A โดยสามารถแสดงเป็นแผนภาพได้ดังนี้

การโอนย้ายข้อมูลกับหน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายใน (Internal Data Memory) เป็นหน่วยความจำ RAM ที่ประกอบอยู่ในตัวไอซีของ 8051 เอง การโอนย้ายข้อมูลที่อยู่ในส่วนของหน่วยความจำภายในนี้ สามารถใช้วิธีการอ้างแอดเดรสที่กล่าวมาได้หลายแบบ ยกเว้นการอ้างแอดเดรสโดยข้ามผ่านค่าฐานภายในรีจิสเตอร์รวมกับค่าดัชนีในอีกรีจิสเตอร์เท่านั้น เนื่องจากมีขนาดหน่วยความจำเพียง 256 ไบต์ แสดงให้เห็นถึงการทำคำสั่งในการโอนย้ายข้อมูลที่อยู่ในหน่วยความจำข้อมูลภายใน ซึ่งก็จะเป็นการดำเนินการกับข้อมูลขนาดหนึ่งไบต์ระหว่างรีจิสเตอร์ A รีจิสเตอร์ R0 - R7 รีจิสเตอร์หน้าที่พิเศษ (SFR) และหน่วยความจำ RAM เป็นที่สังเกตว่าการโอนย้ายข้อมูลระหว่างหน่วยความจำ ภายในด้วยกันนั้น สามารถทำได้โดยไม่ต้องใช้รีจิสเตอร์ R0 - R7 หรือ รีจิสเตอร์ A เลยส่วนข้อมูลที่โอนย้ายให้กับรีจิสเตอร์ DPTR ซึ่งเป็นรีจิสเตอร์ขนาดสองไบต์ (16 บิต) นั้นก็ใช้ค่าตัวเลขขนาดสองไบต์ (Immediate Data) ส่งให้กับรีจิสเตอร์ได้ นอกจากนี้แล้วยังมีการดำเนินการกับข้อมูลแบบบิตด้วย ซึ่งกระทำระหว่างแฟล็กทต (Carry register) กับหน่วยความจำที่ใช้งานแบบบิตได้โดยตรง

ตัวอย่างคำสั่ง

ความหมาย

MOV A,#0F1H

ให้นำค่าตัวเลข F1H ไปไว้ยังรีจิสเตอร์ A

MOV 20H,#0A

ให้นำค่าตัวเลข 0AH ไปเก็บไว้ยังหน่วยความจำภายในแอดเดรส 20H

MOV DPTR,#1234H

ให้นำค่าตัวเลข 16 บิตที่มีค่า 1234 H ไปไว้ยังรีจิสเตอร์ DPTR

MOV A,R0

เป็นการนำค่าข้อมูลจากรีจิสเตอร์ไปเก็บยังรีจิสเตอร์ A

MOV 40H,A

ให้นำข้อมูลจากรีจิสเตอร์ A ไปเก็บหน่วยความจำภายในแอดเดรส 40H

MOV 20H,40H

ให้นำข้อมูลจากหน่วยความจำภายในแอดเดรส 40H ไปเก็บไว้ยังหน่วยความจำภายในแอดเดรส 20H

MOV R0,12H

ให้นำข้อมูลจากหน่วยความจำภายในแอดเดรส 12H ไปเก็บยังรีจิสเตอร์ R0

MOV 8,R7

ให้นำข้อมูลจากรีจิสเตอร์ R7 มาเก็บไว้หน่วยความจำภายในแอดเดรส 08H

MOV @R1,#35H

ให้นำข้อมูลจากหน่วยความจำภายในแอดเดรส 35 H มาเก็บไว้ยังแอดเดรสที่อยู่ภายในรีจิสเตอร์ R1

MOV @R1,80H

ให้นำข้อมูลจากหน่วยความจำภายในแอดเดรส 80H มาเก็บไว้ที่หน่วยความจำที่มีค่าแอดเดรสอยู่ในรีจิสเตอร์ R1

MOV A,@R0

ให้นำข้อมูลที่อยู่ภายในหน่วยความจำที่มีค่าแอดเดรสอยู่ในรีจิสเตอร์ R0 ไปเก็บไว้ในรีจิสเตอร์ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโอนย้ายข้อมูลกับหน่วยความจำข้อมูลภายนอก

หน่วยความจำ RAM ที่อยู่ภายนอกตัวไอซี 8051 จะเรียกว่า หน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งขนาดสูงสุดของหน่วยความจำประเภทนี้จะมีค่าได้สูงสุด 64 กิโลไบต์ (แอดเดรสจาก 0000 - FFFFH) การโอนย้ายข้อมูลกับหน่วยความจำข้อมูลภายนอกของ 8051 นั้นจะใช้คำสั่ง MOVX ซึ่งหลักการคล้ายคลึงกับการใช้คำสั่ง MOV ในหัวข้อที่ผ่านมาทุกประการ (โดยตัวอักษร X มีความหมายถึง External) ดังนั้นเมื่อใช้การอ้างแอดเดรสโดยอ้อมผ่านรีจิสเตอร์ DPTR ก็จะได้ตำแหน่งของหน่วยความจำที่ครบถ้วนตามความสามารถของ 8051 นอกจากนั้นยังสามารถใช้รีจิสเตอร์ R0 และ R1 ได้เช่นกัน เพียงแต่จะอ้างถึงตำแหน่งได้ในขอบเขตที่จำกัดเพียงแอดเดรส 00 - FFH เนื่องจากรีจิสเตอร์ทั้งสองตัวเป็นขนาด 8 บิตเท่านั้น ดังได้แสดงเป็นแผนภาพในรูปที่ 4.3

ตารางต่อไปนี้จะแสดงให้เห็นถึงรูปแบบทั่วไปแบบต่างๆ ของคำสั่งการโอนย้ายข้อมูลกับหน่วยความจำข้อมูลภายนอกโดยพื้นฐาน

ตัวอย่างคำสั่ง	ความหมาย
MOVX A,@R1	ให้นำข้อมูลจากหน่วยความจำภายนอกที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ R1 มาเก็บไว้ยังรีจิสเตอร์ A
MOVX A,@DPTR	ให้นำข้อมูลจากหน่วยความจำภายนอกที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ DPTR มาเก็บไว้ยังรีจิสเตอร์ A
MOVX @R0,A	ให้นำข้อมูลจากรีจิสเตอร์ A มาเก็บไว้ยังตำแหน่งของหน่วยความจำที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ R0
MOVX @DPTR,A	ให้นำข้อมูลจากรีจิสเตอร์ A มาเก็บไว้ยังตำแหน่งของหน่วยความจำที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ DPTR

การโอนย้ายข้อมูลจากหน่วยความจำโปรแกรมภายนอก

หน่วยความจำโปรแกรมของ 8051 เป็นหน่วยความจำที่สามารถอ่านข้อมูลได้อย่างเดียวเท่านั้น เช่นในลักษณะของหน่วยความจำ EPROM ดังนั้นการโอนย้ายข้อมูลจึงเป็นเฉพาะการอ่านข้อมูลมาจากหน่วยความจำโปรแกรมแต่เพียงทางเดียวเท่านั้น การทำงานลักษณะเช่นนี้จะนำมาใช้ประโยชน์ในการเขียนโปรแกรมประเภทที่มี การอ่านข้อมูลที่กำหนดมาล่วงหน้า และไม่สามารถเปลี่ยนแปลงค่าได้ เช่น ตารางค่ารูปแบบของการเปลี่ยนข้อมูลรหัส ASCII ไปเป็นรูปแบบที่แสดงผลบน LED 7 เซกเมนต์ เป็นต้น

การทำงานเพื่อที่จะนำข้อมูลจากหน่วยความจำโปรแกรมนี้อาศัยคำสั่ง MOVX (โดยตัวอักษร C มีความหมายถึง Code) โดยใช้การอ้างแอดเดรสหน่วยความจำแบบโดยอ้อมผ่านค่าฐานภายในไมโครคอนโทรลเลอร์ ซึ่งอีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ P0 หรือ DPTR รวมกับค่าดัชนีในรีจิสเตอร์ A ดังที่ได้กล่าวในหัวข้อที่ผ่านมา เมื่อไปนำค่าจากตำแหน่งดังกล่าวแล้ว ก็จะนำมาจัดเก็บไว้ในรีจิสเตอร์ A อีกครั้งหนึ่ง

ตัวอย่างคำสั่ง

ความหมาย

MOV DPTR,#1000H กำหนดค่าตัวเลขเริ่มต้น 1000H ให้กับรีจิสเตอร์ DPTR

MOV A,#2 กำหนดค่าตัวเลขเริ่มต้น 2H ให้กับรีจิสเตอร์ A

MOVC A,@A+DPTR เป็นการนำค่าจากหน่วยความจำโปรแกรมที่แอดเดรส 1002 มาเก็บไว้ในรีจิสเตอร์ A

MOVC A,@A+PC เป็นการนำค่าจากหน่วยความจำโปรแกรมที่แอดเดรส 502AH มาเก็บไว้ในรีจิสเตอร์ A (สมมติว่าค่ารีจิสเตอร์ PC เป็น 5000H และค่าในรีจิสเตอร์ A เป็น 2AH)

การแลกเปลี่ยนข้อมูลจากหน่วยความจำ

การแลกเปลี่ยนข้อมูล (Data Exchanges) เป็นการโอนย้ายข้อมูลลักษณะหนึ่ง เพียงแต่เมื่อเสร็จสิ้นการทำคำสั่งแล้ว ข้อมูลจากตำแหน่งต้นทางจะถูกย้ายไปเก็บที่ตำแหน่งปลายทาง และข้อมูลตำแหน่งปลายทางจะถูก ย้ายไปเก็บที่ตำแหน่งต้นทาง หรืออาจจะกล่าวโดยง่ายว่ามีการสลับข้อมูลกันระหว่างข้อมูลในตำแหน่งต้นทางและปลายทาง ดังนั้นการแลกเปลี่ยนข้อมูลกับการโอนย้ายข้อมูลจะแตกต่างกันตรงที่ เมื่อภายหลังการทำคำสั่งโอนย้ายข้อมูลแล้วที่ตำแหน่งต้นทางและปลายจะมีข้อมูลเหมือนกันทุกประการ

แผนภาพในรูปที่ 4.5 แสดงให้เห็นถึงการแลกเปลี่ยนข้อมูลของ 8051 ซึ่งดำเนินการได้เฉพาะกับรีจิสเตอร์ A รีจิสเตอร์ R0 - R7 และหน่วยความจำภายในเท่านั้น ซึ่งอาจจะเป็นการอ้างถึงค่าของแอดเดรสโดยตรงหรือโดยอ้อมผ่านรีจิสเตอร์ R0 กับ R1 ได้ ในกรณีหลังนั้นนอกจากจะทำการแลกเปลี่ยนข้อมูลทั้งไบต์เช่นปกติแล้ว ยังสามารถแลกเปลี่ยนเฉพาะข้อมูล 4 บิตล่าง (Low Nibble) กับข้อมูลในรีจิสเตอร์ A ได้อีกด้วย

ตารางต่อไปนี้จะแสดงให้เห็นถึงรูปแบบโดยพื้นฐานของคำสั่งการแลกเปลี่ยนข้อมูล โดยใช้คำสั่ง XCH (ซึ่งย่อมาจากคำว่า Exchange)

ตัวอย่างคำสั่ง

ความหมาย

XCH A,R5 ให้ทำการสลับค่าข้อมูลจากรีจิสเตอร์ R5 กับรีจิสเตอร์ A

XCH A,@R1 ให้ทำการสลับค่าข้อมูลทั้งไบต์จากหน่วยความจำ RAM ที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ R1 กับรีจิสเตอร์ A

XCHD A,@R0 ให้ทำการสลับค่าข้อมูลเฉพาะ 4 บิตล่างจากหน่วยความจำ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำ RAM ที่มีค่าเก็บอยู่ภายในรีจิสเตอร์ R1 กับรีจิสเตอร์

A

XCH A,80H

ให้ทำการสลับค่าข้อมูลระหว่างตำแหน่งหน่วยความจำ
RAM ภายในที่แอดเดรส 80H กับรีจิสเตอร์ A

การโอนย้ายข้อมูลกับสแต็ก

เทคนิควิธีอย่างหนึ่งที่นิยมนำมาใช้ในเพื่อการจัดการข้อมูลภายในโปรแกรม คือ การใช้งานสแต็ก (Stack) ซึ่งโดยหลักการแล้วเป็นการจองพื้นที่หน่วยความจำบริเวณหนึ่งขึ้นมา และจดจำตำแหน่ง เหล่านี้ไว้ว่าเป็นตำแหน่งสำหรับการเก็บข้อมูล เมื่อนำข้อมูลมาเก็บไว้ยังพื้นที่สแต็กก็จะดูค่าภายในตัวชี้ (Pointer) ซึ่งจะบอกว่าเป็นตำแหน่งใดที่จะนำข้อมูลไปเก็บไว้ได้ เมื่อนำข้อมูลมาเก็บไว้แล้วค่าภายในตัวชี้ก็จะเพิ่มขึ้นเพื่อเลื่อนไปยังตำแหน่งถัดไป ดังนั้นหากจะนำข้อมูลต่อไปมาจัดเก็บไว้ก็จะเป็นตำแหน่งแอดเดรสที่อยู่ถัดไป

การทำงานในลักษณะเช่นนี้เปรียบเทียบกับการวางแผ่นกระดาษไว้ในถาดกล่าวคือกระดาษแผ่นที่อยู่ในลำดับล่างสุดจะเป็นแผ่นที่ถูกวางเป็นแผ่นแรกกระดาษแผ่นต่อไปก็จะวางอยู่เหนือกระดาษแผ่นแรกถัดขึ้นไปและจะเป็นเช่นนี้ไปเรื่อยๆ จนสิ้นสุดเมื่อต้องการกระดาษแผ่นใดก็จะต้องนำกระดาษที่วางซ้อนอยู่ก่อนนั้นออกไปจนกระทั่งถึงแผ่นที่ต้องการ

รีจิสเตอร์ของ 8051 ทำหน้าที่เป็นตัวชี้ตำแหน่งหน่วยความจำของสแต็กที่จะเก็บข้อมูล เข้าไปมีชื่อว่า รีจิสเตอร์ SP (Stack Pointer) โดยค่าภายในรีจิสเตอร์นี้จะเป็นค่าของแอดเดรสของตำแหน่งดังกล่าว การโอนย้ายข้อมูล กับสแต็กจะใช้คำสั่ง PUSH และ POP โดยคำสั่ง PUSH เป็นการนำข้อมูลจากโอเพอร์แอนด์ต้นทางเข้าไปเก็บยังสแต็ก ซึ่งจะมีผลทำให้ค่าแอดเดรสภายในรีจิสเตอร์ SP เพิ่มขึ้น สำหรับคำสั่ง POP ก็จะมีลักษณะการทำงานที่ตรงข้ามกัน

เมื่อเริ่มต้นการทำงานของ 8051 โดยอาจจะเป็นการเริ่มจ่ายไฟหรือการรีเซตเกิดขึ้น 8051 จะทำการกำหนดพื้นที่สแต็ก ไว้บริเวณหน่วยความจำข้อมูลภายในที่แอดเดรส 07H ทำให้ค่าเริ่มต้นของรีจิสเตอร์ SP จึงมีค่าเป็น 07H เพื่อชี้ไปยังตำแหน่งดังกล่าว อย่างไรก็ตามหน่วยความจำข้อมูลภายในที่แอดเดรสนี้เป็นพื้นที่ของรีจิสเตอร์ใช้งาน R0 - R7 ในแบงก์ที่ 1 ดังนั้นหากว่ามีการใช้หน่วยความจำนี้ในฐานะของรีจิสเตอร์ R0 - R7 แล้ว ก็จะต้องกำหนดค่าให้กับรีจิสเตอร์ SP ใหม่เพื่อย้ายไปใช้หน่วยความจำบริเวณอื่นเป็นพื้นที่สแต็กแทน รายละเอียดของสแต็กนี้จะได้นำไปกล่าวโดยละเอียดอีกครั้งในบทที่ 7 เพื่อสนับสนุนโครงสร้างการทำงานของโปรแกรมย่อยขอให้ดูรูปแบบโดยพื้นฐานของการใช้งานคำสั่ง PUSH และ POP จากตารางต่อไปนี้

ตัวอย่างคำสั่ง	ความหมาย
MOV 81H,#40H	กำหนดค่าให้กับตำแหน่งรีจิสเตอร์ SP (ซึ่งอยู่ตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับตำแหน่ง 81H) ให้ชี้ไปยังตำแหน่งแอดเดรสให้นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

H ซึ่งจะจองไว้เป็นพื้นที่สแต็ก

MOV R0,#22H	กำหนดค่าตัวเลข 22H ให้กับรีจิสเตอร์ R0
PUSH 00H	รีจิสเตอร์ SP มีค่า 41H และแอดเดรส 41H , มีค่าข้อมูล 22H
POP 07H	รีจิสเตอร์ SP มีค่า 40H และรีจิสเตอร์ R7 มีค่าข้อมูล 22H

หมายเหตุ จากตัวอย่างข้างต้นนี้แสดงให้เห็นถึงการโอนย้ายข้อมูลจากรีจิสเตอร์ R0 ไปยังรีจิสเตอร์ R7 ผ่านทางหน่วยความจำของสแต็กซึ่งอยู่บริเวณแอดเดรส 40H นอกจากนี้จะเห็นว่าในการใช้รูปแบบของคำสั่ง PUSH และ POP นั้นไม่สามารถจะอ้างถึงชื่อรีจิสเตอร์ได้โดยตรง จึงจำเป็นต้องใช้ค่าแอดเดรสของหน่วยความจำ RAM ที่ ตรงกันกับรีจิสเตอร์แทน

การเพิ่มและลดตำแหน่งหน่วยความจำ

การดำเนินงานเกี่ยวกับหน่วยความจำเพื่อเคลื่อนย้ายหรือโอนถ่ายข้อมูล มักจะมีการอ้างถึงตำแหน่งหน่วยความจำแอดเดรสที่อยู่ใกล้เคียงกัน เช่น อาจจะเป็นตำแหน่งที่อยู่ก่อนหน้าหรืออยู่หลังจากแอดเดรสปัจจุบันไปหนึ่งตำแหน่ง เป็นต้น ซึ่งหากพิจารณาโดยพื้นฐานจะมีลักษณะเหมือนกับการลบค่าตัวเลข 1 หรือบวกค่าตัวเลข 1 เข้ากับหมายเลขแอดเดรสในปัจจุบันนั่นเอง คำสั่งพื้นฐานของ 8051 ที่สนับสนุนการทำงานเช่นนี้ ได้แก่ คำสั่ง INC และ DEC ซึ่งมักจะนำมาใช้ประกอบกับการอ้างแอดเดรสหลายๆ ลักษณะ ทั้งแบบการอ้างตำแหน่งแอดเดรสโดยอ้อมหรือโดยตรงอย่างไรก็ตามคำสั่งทั้งสองนี้ก็ยังสามารถนำไปใช้ในวัตถุประสงค์เพื่อลดหรือเพิ่มค่าภายในรีจิสเตอร์ต่างๆ ได้

การโอนย้ายข้อมูลจากตาราง

โครงสร้างของข้อมูลซึ่งมีการจัดการให้อยู่ในลักษณะของตาราง ประกอบด้วยกลุ่มของข้อมูลที่มีการจัดเรียงไว้เป็นลำดับอนุกรมต่อเนื่อง โดยสมาชิกตัวแรกของตาราง คือตัวที่ศูนย์ ตัวถัดมาเป็นตัวที่หนึ่ง, สอง,.....เป็นลำดับไป หากว่าตำแหน่งของตารางเริ่มต้นที่แอดเดรส 100H ดังนั้น

ข้อมูลตัวแรกจะอยู่ที่แอดเดรส $100+0 = 100H$,

ข้อมูลตัวแรกจะอยู่ที่แอดเดรส $100+1 = 101H$,

ข้อมูลตัวแรกจะอยู่ที่แอดเดรส $100+2 = 102H$,

ข้อมูลตัวแรกจะอยู่ที่แอดเดรส $100+3 = 103H$, เรื่อยไป

โครงสร้างข้อมูลแบบตารางนี้มักจะนิยมนำมาใช้ในการช่วยแก้ปัญหาให้ง่ายขึ้น โดยทำการคิดคำนวณผลลัพธ์ไว้ล่วงหน้า แล้วนำไปเก็บยังตารางไว้ จากนั้นในขณะเมื่อประมวลผลโปรแกรมก็เพียงไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เลือกจับคู่กับค่าที่เหมาะสมภายในตารางเท่านั้น ตัวอย่างเช่น การคำนวณหาค่าของ $\sin x$ ก็อาจจะทำการคำนวณค่า \sin เมื่อ x เป็นค่าตัวเลขจำนวนเต็มต่างๆ ไว้เรียบร้อยแล้ว และนำมาจัดเก็บเป็นโครงสร้างลักษณะแบบตาราง เมื่อต้องการหาค่าของ $\sin 2$ เพียงแต่ไปนำคำตอบมาจากข้อมูลตัวที่สามของตารางเท่านั้น ไม่ต้องมีการคำนวณใดๆ เกิดขึ้นทั้งสิ้น (ดูรูปที่ 4.9) ซึ่งจะมีผลทำให้โปรแกรมลดความซับซ้อนลงได้อย่างมาก และใช้เวลาในการประมวลผลได้รวดเร็วมากยิ่งขึ้น ขอให้ทำความเข้าใจในการนำไปใช้งานจากตัวอย่างโปรแกรมในหัวข้อถัดไป

คำสั่งการใช้งานพอร์ตอินพุต/เอาต์พุต

เมื่อพิจารณาถึงวิธีการส่งข้อมูลภายในพอร์ตจะสามารถแยกประเภทพอร์ตออกได้เป็นพอร์ตแบบขนาน (Parallel port) ซึ่งทำการส่งจำนวนบิตข้อมูลทั้งหมดออกมา หรือนำเข้าไปพร้อมกันในคราวเดียว และพอร์ตแบบอนุกรม (Serial port) ซึ่งทำการโอนย้ายข้อมูลคราวละบิตๆ จนครบจำนวน แต่สำหรับในบทนี้จะได้กล่าวถึงเฉพาะในส่วนของพอร์ตแบบขนานเท่านั้น เนื่องจาก 8051 ใช้หลักการที่เรียกว่า Memory mapped system กล่าวคือ การอ้างถึงพอร์ต รีจิสเตอร์หรืออุปกรณ์ต่างๆ ภายในระบบจะเป็นการติดต่อกับหน่วยความจำตำแหน่งหนึ่งเท่านั้น ดังนั้นในการดำเนินการเพื่อนำเข้าหรือส่งออกข้อมูลกับพอร์ต จึงใช้คำสั่งการอ่านค่าจากหน่วยความจำซึ่งถูกออกแบบให้เป็นตำแหน่งของพอร์ต หรือคำสั่งการเขียนค่าข้อมูลไปยังตำแหน่งหน่วยความจำนั้นแทน ดังนั้นจะสังเกตเห็นได้ว่าในตารางชุดคำสั่งของ 8051 จะไม่มีคำสั่งที่เกี่ยวกับการทำงาน พอร์ตแต่ประการใด เช่น คำสั่ง IN (นำเข้าข้อมูลจากพอร์ต) หรือ คำสั่ง OUT (ส่งข้อมูลออกทางพอร์ต) เป็นต้น นอกจากนี้ยังสามารถใช้คำสั่งจัดการข้อมูลแบบบิตได้โดยตรง (Single-bit Operation) เพื่อมาจัดการพอร์ตอินพุต/เอาต์พุตทั้งหมดแบบเส้นสัญญาณเดียวได้ โดยการใช้คำสั่ง SETB เพื่อกำหนดค่าเป็น 1 หรือคำสั่ง CLR เพื่อทำให้บิตมีค่าเป็น 0 คำสั่งเหล่านี้มีประโยชน์ และทำให้ลดความซับซ้อนในการใช้คำสั่งภายในโปรแกรมลงได้มาก การพิจารณาตรวจสอบค่าบิตของแต่ละพอร์ตอินพุต/เอาต์พุตนั้นก็สามารุณใช้ชุดคำสั่ง ในการทดสอบบิตของพอร์ตได้โดยตรง โดยไม่ต้องมีการย้ายค่าของพอร์ตไปยังรีจิสเตอร์ก่อนแต่ประการใด

พอร์ตแบบขนานของ 8051

8051 มีโครงสร้างของพอร์ตที่สามารถใช้งานแบบขนานได้จำนวนทั้งหมดสี่พอร์ต เรียกชื่อเรียงตามลำดับว่า พอร์ต 0 1 2 และ 3 และเป็นพอร์ตขนาดแปดบิตทั้งหมด การใช้งานพอร์ตสามารถทำได้ทั้งในลักษณะของเส้นสัญญาณเดี่ยวๆ หรือกลุ่มของสัญญาณได้ นอกจากนี้พอร์ต 0 2 และ 3 ยังสามารถนำไปใช้งานอื่นๆ ที่ไม่ใช่เป็นพอร์ตอินพุต/เอาต์พุตได้ โดยพอร์ต 0 จะทำหน้าที่มัลติเพลกซ์ระหว่างบัสแอดเดรสในบิตต่ำและบัสข้อมูลสำหรับการติดต่อกับวงจรมอบรวมกับข้อมูลไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัสด์แอดเดรสไบต์สูงซึ่งจะส่งออกมาทางพอร์ต 2 สำหรับพอร์ต 3 นั้นนอกเหนือไปจากความสามารถ เช่นพอร์ตปกติแล้ว สามารถนำไปเป็นขาสัญญานของการอินเตอร์รัปต์ต่างๆ ซึ่งรวมทั้งการสร้าง สัญญานควบคุม RD\ และ WR\ เพื่อทำหน้าที่อ่านหรือเขียนหน่วยความจำข้อมูลภายนอกด้วยการใช้งานพอร์ตลักษณะงานแบบอื่นๆ ที่ไม่ใช่เป็นพอร์ตแบบอินพุต/เอาต์พุตนี้จะดำเนินการโดย 8051 เองโดยอัตโนมัติ

โครงสร้างการอินเตอรัปต์ของ 8051

จากแผนภาพโครงสร้างระบบอินเตอรัปต์ของ 8051 ในรูปที่ 10.2 แสดงให้เห็นว่า สัญญานที่เข้ามาทำการอินเตอรัปต์ 8051 นั้นเกิดขึ้นได้ห้าลักษณะตามตารางข้อมูลในรูปที่ 10.3 โดยจะเห็นได้ว่าจะสามารถที่จะกำหนดเลือกเพื่อยินยอม (หรืออีนาเบิล: Enable) และห้าม (หรือดิสเอเบิล: Disable) ไม่ให้มีการอินเตอรัปต์แต่ละประเภทได้ โดยการกำหนดบิตของข้อมูลที่เกี่ยวข้องซึ่ง มักจะอยู่ภายในรีจิสเตอร์ TCON และ SCON นอกจากนี้ยังมีตำแหน่งบิตภายในรีจิสเตอร์ IE (Intereupt Enable Register) ซึ่งทำหน้าที่เสมือนกับเป็นสวิตช์หลักที่เกี่ยวข้องกับสัญญาณอินเตอรัปต์ทั้งหมดโดยหากว่ากำหนดให้ห้ามการเกิดอินเตอรัปต์แล้ว การกำหนดบิตเพื่อห้ามหรือยินยอม ของแต่ละอินเตอรัปต์ก็จะมีผลใดๆ เกิดขึ้น นอกจากนี้แผนภาพในรูปที่ 5.1 ยังแสดงให้เห็นด้วยว่าสัญญาณอินเตอรัปต์แต่ละประเภทยังสามารถกำหนด ระดับความสำคัญ (Priority) ของการอินเตอรัปต์ได้สองลักษณะ คือ ระดับความสำคัญสูงหรือต่ำ (Highor Low Priority) กล่าวคือ ขณะที่กำลังประมวลผลอยู่ภายในส่วนของโปรแกรมย่อยบริการอินเตอรัปต์ของสัญญาณที่มีระดับความสำคัญต่ำอยู่ ก็อาจจะถูกขัดจังหวะให้ไปประมวลผลของสัญญาณอินเตอรัปต์ ที่มีระดับความสำคัญสูงกว่าได้ แต่หากว่าเป็นสัญญาณอินเตอรัปต์ที่มีระดับความสำคัญต่ำเช่นเดียวกันแล้วก็จะต้องรอให้เสร็จสิ้นการประมวลผลที่ดำเนินการอยู่ก่อน

การควบคุมอินเตอรัปต์

ตามโครงสร้างด้านการจัดการอินเตอรัปต์ของ 8051 สามารถกำหนดเลือกเพื่อยินยอมหรือ ไม่ยินยอม (Enable/Disable) ให้มีการอินเตอรัปต์ของแต่ละสัญญาณได้ โดยใช้วิธีการกำหนดค่าของบิตภายในรีจิสเตอร์ IE (ดูรูปที่ 10.4) ซึ่งจะมีทั้งแบบที่ระบุถึงอินเตอรัปต์โดยรวมทั้งหมด (บิตที่ 7) และอินเตอรัปต์แต่ละประเภทได้ ในกรณีที่กำหนดค่าข้อมูลเป็น 1 ให้กับบิตจะมีความหมายถึงการยินยอมให้มีการอินเตอรัปต์เกิดขึ้นได้ และจะเป็นกรณีตรงข้ามกันสำหรับการกำหนดค่าข้อมูลที่เป็น 0 หากลองย้อนกลับไปพิจารณาแผนภาพในรูปที่ 10.2 อีกครั้ง จะเห็นว่าจะต้องทำการกำหนดให้ยินยอมการอินเตอรัปต์ทั้งหมดให้เกิดขึ้นก่อน จึงจะมีผลทำให้การกำหนดบิตเพื่อยินยอมของแต่ละอินเตอรัปต์มีผลขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ใดๆ ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับความสำคัญของการอินเทอร์รัปต์

การกำหนดระดับความสำคัญให้กับสัญญาณอินเทอร์รัปต์แต่ละประเภทนั้น สามารถทำได้ โดยการกำหนดข้อมูลที่มีค่าเป็น 1 หรือ 0 ให้กับบิตภายในรีจิสเตอร์ IP (Interrupt Priority) ดังแสดงในตารางของรูปที่ 10.5 โดยหากว่ามีค่าเป็น 1 ก็จะทำให้สัญญาณอินเทอร์รัปต์นั้นๆ มีระดับความสำคัญสูง และเป็นกรณีตรงข้ามกันสำหรับการกำหนดค่าเป็น 0

กรณีที่สัญญาณที่เข้ามาอินเทอร์รัปต์มีระดับความสำคัญเดียวกัน เกิดขึ้นพร้อมกัน ก็อาจจะทำให้เกิดปัญหาขึ้นได้ แต่อย่างไรก็ตาม 8051 ก็มีโครงสร้างทางด้านฮาร์ดแวร์ในการพิจารณาตามลำดับของตารางดังต่อไปนี้

การรีเซ็ต

โดยความหมายของการรีเซ็ต (Reset) เป็นการบังคับให้มีการเริ่มต้นใหม่อีกครั้งหนึ่ง ซึ่งมักจะกระทำโดยการกำหนดสถานะของสัญญาณที่ขา RESET ของตัวไอซี 8051 ให้เป็นระดับลอจิกที่เหมาะสมเท่านั้น การรีเซ็ตนี้ถือว่าการอินเทอร์รัปต์อย่างหนึ่งได้ แต่จะมีลักษณะต่างออกไปจากอินเทอร์รัปต์แบบอื่นๆ คือไม่มีตำแหน่งบิตภายในรีจิสเตอร์ ที่สามารถจะห้ามไม่ให้มีการอินเทอร์รัปต์ของสัญญาณนี้ได้ ซึ่งมีศัพท์เฉพาะเรียกว่า Non-maskable Interrupt นอกจากนี้การดำเนินการของโปรแกรมก็จะแตกต่างออกไปด้วย โดยจะไม่มีเก็บค่าของคำสั่งที่กำลังจะไปทำในลำดับต่อไปภายในรีจิสเตอร์ PC เมื่อมีการรีเซ็ตเกิดขึ้น โปรแกรมจะถูกส่งให้กระโดดไปยังแอดเดรส 0000 แทนที่ ซึ่งตำแหน่งนี้จะเป็นตำแหน่งเริ่มต้นของการทำงานของ 8051 เมื่อเริ่มจ่ายไฟให้กับระบบเมื่อใดก็ตามที่มีการรีเซ็ตเกิดขึ้น ค่าสถานะต่างๆภายในไมโครคอนโทรลเลอร์จะถูกกำหนดกลับไปเป็นค่าเริ่มต้นใหม่อีกครั้ง ดังตาราง แต่ขอให้สังเกตด้วยว่าข้อมูลภายในหน่วยความจำข้อมูลภายในจะไม่โดนทำลายหรือรีเซ็ตตามไปด้วย จึงยังคงมีค่าข้อมูลเก่าอยู่เช่นเดิม นอกจากนั้นยังทำให้กลับไปใช้รีจิสเตอร์ใช้งานทั่วไป R0 - R7 จากแบงก์แรกอีกด้วยเนื่องจากค่าของรีจิสเตอร์ PSW โดยรีเซ็ตกลับไปเป็นค่าศูนย์

รีจิสเตอร์	ค่าเริ่มต้น(ฐานสิบหก)
PC	0000
DPTR	0000
A	00
B	00
SP	07
PSW	00
P0-P3	FF
IP	XXX00000 (ฐานสอง)
IE	0XX00000 (ฐานสอง)
TCON	00
TMOD	00
THQ	00
TLO	00
THI	00
TLI	00
SCON	00
SBUF	XX
PCON	0XXXXXXXX (ฐานสอง)

ชุดคำสั่งเกี่ยวกับการอินเตอร์รัปต์

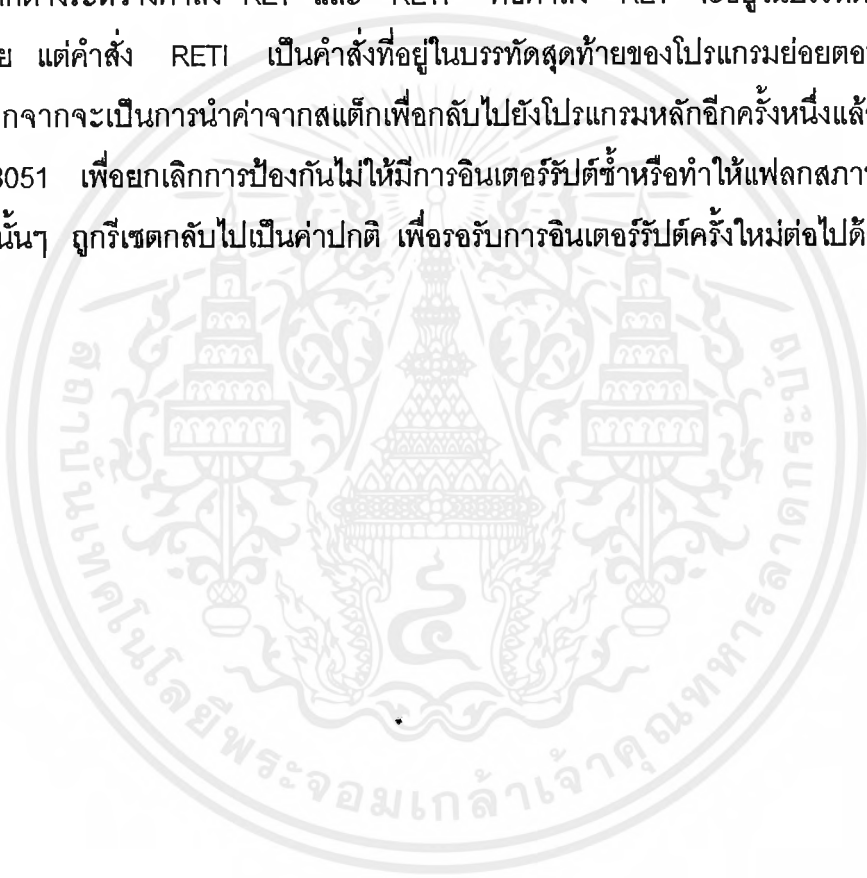
จากลักษณะการทำงานเมื่อมีการอินเตอร์รัปต์เกิดขึ้นจะมีความคล้ายคลึงกับการเรียกใช้งานโปรแกรมย่อยตามที่ได้กล่าวในบทที่ผ่านมา เพียงแต่การเรียกโปรแกรมย่อยนี้เป็นการเรียกมาจากวงจรรีอาร์ตแวร์ของไมโครคอนโทรลเลอร์ และตำแหน่งของโปรแกรมย่อยนั้นก็ได้มีการระบุตำแหน่งไว้แน่นอนแล้ว (ดูรูปที่ 10.6) เมื่อมีการอินเตอร์รัปต์เกิดขึ้น 8051 จะทำการเซตวงจรมายในไว้เพื่อป้องกันไม่ให้เกิดการอินเตอร์รัปต์ซ้ำ จนกว่าจะได้กระทำคำสั่ง RETI ในส่วนสิ้นสุดของโปรแกรมย่อยตอบสนองการอินเตอร์รัปต์เสียก่อน ขอให้ดูความหมายของคำสั่งนี้ได้จากตารางต่อไปนี้

การกลับไปยังโปรแกรมหลักภายหลังการอินเตอร์รัปต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่ง	ความหมาย
RETI	.ให้ทำการนำค่าข้อมูลสองไบต์ออกจากสแต็ก (เช่นเดียวกับการ Pop) และนำมาเก็บไว้ภายในรีจิสเตอร์ PC และรีเซตวงจรรกการห้ามการอินเตอร์รัปต์ให้กลับเป็นปกติ

ความแตกต่างระหว่างคำสั่ง RET และ RETI คือคำสั่ง RET จะอยู่ในบรรทัดสุดท้ายภายในส่วนของโปรแกรมย่อย แต่คำสั่ง RETI เป็นคำสั่งที่อยู่ในบรรทัดสุดท้ายของโปรแกรมย่อยตอบสนองการอินเตอร์รัปต์ ซึ่งนอกจากจะเป็นการนำค่าจากสแต็กเพื่อกลับไปยังโปรแกรมหลักอีกครั้งหนึ่งแล้วยังจะมีผลต่อวงจรรีเซ็ตแวนซ์ของ 8051 เพื่อยกเลิกการป้องกันไม่ให้เกิดการอินเตอร์รัปต์ซ้ำหรือทำให้แฟล็กสถานะที่เกี่ยวข้องกับการอินเตอร์รัปต์นั้นๆ ถูกรีเซตกลับไปเป็นค่าปกติ เพื่อรอรับการอินเตอร์รัปต์ครั้งใหม่ต่อไปด้วย



บทที่ 5

โปรแกรมสำหรับเครื่องเลือกสัญญาณ

```

00F8=      1      COMMANDS EQU 11111000B
00F9=      2      WRD          EQU 11111001B
007F=      3      DELAY_SCAN EQU 07FH
0086=      4      ENABLE     EQU P0.6
0084=      5      C_BIT0     EQU P0.4
0085=      6      C_BIT1     EQU P0.5
0002=      7      DELAY_SCANX EQU 2
           8
0000           9      ORG 0000H
0000 0201D0  10      JMP START
0013           11     ORG 0013H
0013 020016  12      JMP ISR1
0016           13     ISR1:
0016 C2AF    14      CLR EA
0018 C003    15      PUSH 03
           16
001A AE72    17      MOV R6,72H
001C BEAA68  18      CJNE R6,#0AAH,NORMAL_NORMAL ;KEY_FROM
           19      ;FUNCTION S
           20
           21
           22      ;FUNCTION ON SCAN
001F 20A435  23      SETY:   JB P2.4,KEY_STOP ;SET MODE
0022 30A4FD  24      SETX:   JNB P2.4,SETX
0025           25     SETYK:
0025 9002DC  26      MOV DPTR,#TSET
0028 120233  27      CALL IN_MEM
002B 7430    28      MOV A,#30H
002D 2573    29      ADD A,73H
002F F54B    30      MOV 4BH,A
0031 120200  31      CALL SCAN
0034           32     XXX3:
0034 20A50F  33      JB P2.5,NKSET
0037 30A5FD  34      KSELECT: JNB P2.5,KSELECT
003A 0573    35      INC 73H
003C AE73    36      MOV R6,73H
003E BE05E4  37      CJNE R6,#5,SETYK
0041 757301  38      MOV 73H,#1
0044 80DF    39      JMP SETYK
           40
0046           41     NKSET:
0046 20A4EB  42      JB P2.4,XXX3
0049 742F    43      MOV A,#2FH
004B 12029F  44      CALL DELAY_MS
004E 30A4FD  45      NKSETY: JNB P2.4,NKSETY
0051 90011A  46      MOV DPTR,#FUNCTION_SCAN
0054 020099  47      JMP GO_GO ;END SETM
           48
           49
0057           50     KEY_STOP:
0057 20A623  51      JB P2.6,KEY1TO4X

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

52
005C 900107 53 MOV DPTR,£FUNCTION_SCAN
005F 0200A4 54 JMP GO_GO
55
56
0062 57 KEY_STOP:
0062 20A623 58 JB P2.6,KEY1TO4X
8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
nop7.asm

0065 742F 59 MOV A,£2FH
0067 120284 60 CALL DELAY_MS
006A 30A6FD 61 STOPX: JNB P2.6,STOPX
62
006D 9002E1 63 MOV DPTR,£SOSCAN
0070 120218 64 CALL IN_MEM
0073 1201E5 65 CALL SCAN
0076 74FF 66 MOV A,£0FFH
0078 120284 67 CALL DELAY_MS
68
007B 9000C7 69 MOV DPTR,£SENDXX
007E C2AF 70 CLR EA
0080 C2AA 71 CLR EX1
0082 757200 72 MOV 72H,£00
0085 0200A4 73 JMP GO_GO
74
75
0088 76 KEY1TO4X:
0088 9000CA 77 MOV DPTR,£KEY1TO4
008B C2AF 78 CLR EA
008D C2AA 79 CLR EX1
008F 0200A4 80 JMP GO_GO
81
0092 82 NORMAL_NORMAL:
0092 20A71B 83 JB P2.7,OUTISRX
0095 7571AA 84 MOV 71H,£0AAH
0098 30A7FD 85 NP1: JNB P2.7,NP1
009B 9000C5 86 MOV DPTR,£KEY_START
009E 0200A4 87 JMP GO_GO
88
89
00A1 90 NEXT_KEY1:
00A1 900000 91 MOV DPTR,£00H
92
00A4 93 GO_GO:
00A4 758109 94 MOV SP,£09H
95 ; MOV 07H,£00
00A7 858208 96 MOV 08H,DPL
00AA 858309 97 MOV 09H,DPH
98
00AD 99 OUTISR:
00AD D2AF 100 SETB EA
00AF 32 101 RETI
00B0 102 OUTISRX:
00B0 D2AF 103 SETB EA

```

```

00B2 D003      104      POP 03
00B4 32        105      RETI
                106
                107
00B5          108      KEY_STARTI:
00B5 120218    109      CALL IN_MEM
00B8 7430      110      MOV A, f30H
00BA 2B        111      ADD A, R3
00BB F542      112      MOV 42H, A
00BD 1201E5    113      CALL SCAN
00C0 7401      114      MOV A, f1
00C2 120299    115      CALL DELAY_SEC
                116
                117

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
nop7.asm

```

00C5          118      KEY_START:
00C5 7B01      119      MOV R3, f1
00C7          120      SENDXX:
00C7 12015C    121      CALL SEND_CONTROL
                122      ;-----
                123
                124
00CA          125      KEY1TO4:
00CA 43800F    126      ORL P0, f00001111B
00CD E580      127      MOV A, P0
00CF          128      NSY:
                129
00CF 20A007    130      K1:      JB P2.0, K2
00D2 30A0FD    131      K11:     JNB P2.0, K11
00D5 7B01      132      MOV R3, f1
00D7 80EE      133      JMP SENDXX
                134
                135
00D9 20A107    136      K2:      JB P2.1, K3
00DC 30A1FD    137      K22:     JNB P2.1, K22
00DF 7B02      138      MOV R3, f2
00E1 80E4      139      JMP SENDXX
                140
                141
00E3 20A207    142      K3:      JB P2.2, K4
00E6 30A2FD    143      K33:     JNB P2.2, K33      ;CHEC
00E9 7B03      144      MOV R3, f3
00EB 80DA      145      JMP SENDXX
                146
00ED 20A307    147      K4:      JB P2.3, KSCAN
00F0 30A3FD    148      K44:     JNB P2.3, K44
00F3 7B04      149      MOV R3, f4
00F5 80D0      150      JMP SENDXX
                151
                152
00F7 20A606    153      KSCAN:   JB P2.6, KMMM
00FA 30A6FD    154      KSCAN1: JNB P2.6, KSCAN1
00FD 020107    155      JMP FUNCTION_SCAN      ;CHEC

```

```

0100          156
0100 80C8     157
              158
              159
0102          160
0102 D2AA     161
0104 02018C   162
              163
              164
              165
0107          166
              167
0107 9002D1   168
010A 120218   169
010D 1201E5   170
0110 74FF     171
0112 120284   172
              173
0115 7572AA   174
0118 D2AF     175
011A D2AA     176
8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
nop7.asm

              177
011C AC70     178
011E BC0403   179
0121 02014E   180
0124 BC0303   181
0127 020144   182
012A BC0203   183
012D 02013A   184
              185
0130          186
              187
0130 7B01     188
0132 12015C   189
0135 E573     190
0137 120299   191
013A          192
013A 7B02     193
013C 12015C   194
013F E573     195
0141 120299   196
              197
0144          198
0144 7B03     199
0146 12015C   200
0149 E573     201
014B 120299   202
014E          203
014E 7B04     204
0150 12015C   205
0153 E573     206
0155 120299   207

KMMM:
      JMP KEY1TO4

EXIT_KEY_START:
      SETB EX1
      JMP STARTX

FUNCTION_SCAN:
      MOV DPTR, fSTSCAN
      CALL IN_MEM
      CALL SCAN
      MOV A, fOFFH
      CALL DELAY_MS

      MOV 72H, f0AAH
      SETB EA
      SETB EX1

      MOV R4, 70H
      CJNE R4, f4, NCHH3
      JMP CCH4
NCHH3: CJNE R4, f3, NCHH2
      JMP CCH3
NCHH2: CJNE R4, f2, CCH1
      JMP CCH2

CCH1:

      MOV R3, f1
      CALL SEND_CONTROL
      MOV A, 73H
      CALL DELAY_SEC

CCH2:
      MOV R3, f2
      CALL SEND_CONTROL
      MOV A, 73H
      CALL DELAY_SEC

CCH3:
      MOV R3, f3
      CALL SEND_CONTROL
      MOV A, 73H
      CALL DELAY_SEC

CCH4:
      MOV R3, f4
      CALL SEND_CONTROL
      MOV A, 73H
      CALL DELAY_SEC

```

```

0158 80D6      208      JMP CCH1
015A 80FE      209      JMP $
                210
                211
                212
015C          213      SEND_CONTROL:
                214
015C 8B70      215      MOV 70H,R3      ;SEND CONTROL
015E 12025D    216      CALL INIT
0161 9002B1    217      MOV DPTR, fCH_ON
0164 120218    218      CALL IN_MEM
0167 E570      219      MOV A,70H
0169 2430      220      ADD A, f30H
016B F542      221      MOV 42H,A
                222
016D 1201E5    223      CALL SCAN      ;SHOW MOINTOR
                224
                ;-----
0170 74FF      225      MOV A, f0FFH
0172 1570      226      DEC 70H
0174 5570      227      ANL A,70H
                228
0176 FF        229      MOV R7,A
0177 5403      230      ANL A, f00000011B
0179 F4        231      CPL A
017A 5307FC    232      ANL 07H, f11111100B
017D 4F        233      ORL A,R7
                234
017E 33        235      RLC A
8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
nop7.asm
017F 33        236      RLC A
0180 33        237      RLC A
0181 33        238      RLC A
                239
0182 F580      240      MOV P0,A
0184 C286      241      CLR P0.6
0186 0570      242      INC 70H
0188 22        243      RET
                244
                245
                246
                247
0189          248      START:
0189 757301     249      MOV 73H, f1      ;DELAY
018C          250      STARTX:
018C 757100     251      MOV 71H, f0
018F 757200     252      MOV 72H, f0
0192 7580FF     253      MOV P0, f0FFH
0195 75B0FF     254      MOV P3, f0FFH
0198 75A0FF     255      MOV P2, f0FFH      ;SET "1
019B 12025D     256      CALL INIT        ;INIT L
                257
019E 75A884     258      MOV IE, f84H      ;ENABLE
01A1 D2AA      259      SETB EX1

```

```

01A3 C28A      260          CLR IT1
                261
01A5          262          DEMO:
01A5 12025D    263          CALL INIT
01A8 E580      264          MOV A,P0          ;LOAD D
01AA 7B01      265          MOV R3,£1        ;CHECK
                266
01AC          267          START_CHECK:
01AC 13        268          RRC A
01AD 500C      269          JNC SET
                270
01AF          271          RESETY:
01AF C0E0      272          PUSH ACC
01B1 7401      273          MOV A,£1
01B3 1201D3    274          CALL SHOW
01B6 D0E0      275          POP ACC
01B8 0201C4    276          JMP END_CHECK
                277
01BB          278          SET:
01BB C0E0      279          PUSH ACC
01BD 7400      280          MOV A,£0
01BF 1201D3    281          CALL SHOW
01C2 D0E0      282          POP ACC
                283
01C4          284          END_CHECK:
01C4 C0E0      285          PUSH ACC
01C6 7401      286          MOV A,£1
01C8 120299    287          CALL DELAY_SEC
01CB D0E0      288          POP ACC
                289
01CD 0B        290          INC R3
01CE BB05DB    291          CJNE R3,£5,START_CHECK
01D1 80D2      292          JMP DEMO
                293
                294
8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
nop7.asm
                295
01D3          296          SHOW:
01D3 B40103    297          CJNE A,£1,NO_READY
01D6 0201D9    298          JMP READYX
                299          NO_READY:
01D9          300          READYX:
01D9 120218    301          CALL IN_MEM
                302
01DC 7430      303          MOV A,£30H
01DE 2B        304          ADD A,R3
01DF F542      305          MOV 42H,A
                306
01E1 1201E5    307          CALL SCAN
01E4 22        308          RET
                309          SCAN:
01E5 C083      310          PUSH DPH
01E7 C082      311          PUSH DPL

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
nop7.asm

```

01E9 C000      312      PUSH 00
01EB 7480      313      MOV A,£80H
01ED 120225    314      CALL BLOCK2
                                315
01F0 75F008    316      MOV B,£8
01F3 7840      317      MOV R0,£40H
01F5           318      L1:
01F5 E6        319      MOV A,@R0
01F6 120235    320      CALL WRITE
01F9 08        321      INC R0
01FA D5F0F8    322      DJNZ B,L1
                                323
01FD 74C0      324      MOV A,£0C0H
01FF 120225    325      CALL BLOCK2
                                326
0202 75F008    327      MOV B,£8
0205 7848      328      MOV R0,£48H
0207 7400      329      LX: MOV A,£0
0209 E6        330      MOV A,@R0
020A 120235    331      CALL WRITE
020D 08        332      INC R0
020E D5F0F6    333      DJNZ B,LX
                                334
0211 D000      335      POP 00
0213 D082      336      POP DPL
0215 D083      337      POP DPH
0217 22        338      RET
                                339
0218           340
                                341      IN_MEM:
                                342
0218 7A10      343      MOV R2,£16
021A 7840      344      MOV R0,£40H
                                345
021C 7400      346      IN1: MOV A,£0
021E 93        347      MOVC A,@A+DPTR
021F F6        348      MOV @R0,A
0220 A3        349      INC DPTR
0221 08        350      INC R0
0222 DAF8      351      DJNZ R2,IN1
                                352
0224 22        353      RET

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
nop7.asm

```

                                354
0225           355      BLOCK2:
0225 C0E0      356      PUSH ACC
0227 75B0F8    357      MOV P3,£COMMANDS
022A D0E0      358      POP ACC
022C F590      359      MOV P1,A
022E 120245    360      CALL EPLUSE
0231 120253    361      CALL WAITBF
0234 22        362      RET
0235           363      WRITE:

```

```

0235 C0E0      364      PUSH ACC
0237 75B0F9   365      MOV P3, fWRD
023A D0E0      366      POP ACC
023C F590      367      MOV P1, A
023E 120245   368      CALL EPLUSE
0241 120253   369      CALL WAITBF
0244 22        370      RET
              371
0245 D2B2      372      EPLUSE: SETB P3.2
0247 C0E0      373      PUSH ACC
0249 7401      374      MOV A, f01
024B 120284   375      CALL DELAY_MS
024E C2B2      376      CLR P3.2
0250 D0E0      377      POP ACC
0252 22        378      RET
              379
              380
0253          381      WAITBF:
0253 C0E0      382      PUSH ACC
0255 7404      383      MOV A, f4
0257 120284   384      CALL DELAY_MS
025A D0E0      385      POP ACC
025C 22        386      RET
              387
025D          388      INIT:
025D 75B0F8   389      MOV P3, fCOMMANDS
              390
0260 759038   391      MOV P1, f38H      ;DISPLAY 5*7
0263 5145     392      CALL EPLUSE
0265 5153     393      CALL WAITBF
              394
              395
0267 75900F   396      MOV P1, f0FH
026A 5145     397      CALL EPLUSE
026C 5153     398      CALL WAITBF
              399
026E 759006   400      MOV P1, f6
0271 5145     401      CALL EPLUSE
0273 5153     402      CALL WAITBF
              403
0275 759001   404      MOV P1, f1
0278 5145     405      CALL EPLUSE
027A 5153     406      CALL WAITBF
              407
027C 759080   408      MOV P1, f80H
027F 5145     409      CALL EPLUSE
0281 5153     410      CALL WAITBF
0283 22        411      RET
              412

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
 nop7.asm

```

413      ;PROGRAM DELAY TIME
414      ;USED REGISTER A
415      ;SUB DELAY_SEC AND DELAY_MS

```

```

416 ;PROGRAM BY RATCHATAPAK PATTWONG. KMIT
417
0284 418 DELAY_MS:
0284 COE0 419 PUSH ACC
0286 COF0 420 PUSH B
0288 75F000 421 MOV B,£0
028B 422 DD:
028B D5F0FD 423 DJNZ B,$ ;500 uS AT 12
028E D5F0FD 424 DJNZ B,$ ;500 uS AT 12
0291 D5E0F7 425 DJNZ ACC,DD
426
0294 DOF0 427 POP B
0296 DOE0 428 POP ACC
0298 22 429 RET
430
0299 431 DELAY_SEC:
0299 COE0 432 PUSH ACC
029B COF0 433 PUSH B
029D F5F0 434 MOV B,A
029F 435 DDD:
029F 74FA 436 MOV A,£250
02A1 5184 437 CALL DELAY_MS ;250 mS
02A3 5184 438 CALL DELAY_MS ;500 mS
02A5 5184 439 CALL DELAY_MS ;750 mS
02A7 5184 440 CALL DELAY_MS ;1000 mS
02A9 D5F0F3 441 DJNZ B,DDD
02AC DOF0 442 POP B
02AE DOE0 443 POP ACC
02B0 22 444 RET
445
02B1 43485820 446 CH_ON: DB "CHX MONITOR "
02B5 2020204D
02B9 4F4E4954
02BD 4F522020
02C1 44454C41 447 TSET: DB "DELAY TIME05 "
02C5 59205449
02C9 4D453035
02CD 20202020
02D1 20205343 448 STSCAN: DB " SCAN READY "
02D5 414E2052
02D9 45414459
02DD 20202020
02E1 20205343 449 SOSCAN: DB " SCAN STOP "
02E5 414E2053
02E9 544F5020
02ED 20202020
0000= 450 END

```

บทที่ 6

ผลการทดลอง

ทำการทดสอบโดยใช้สัญญาณวีดีโอจากเครื่องเล่นวีดีโอโดยขั้นแรกทำการป้อนสัญญาณเข้าที่ช่องสัญญาณ โดยที่ยังไม่ได้ต่อเข้ากับไมโครคอนโทรลเลอร์ โดยที่เราทำการจำลองสัญญาณควบคุมที่จะถูกส่งโดยไมโครคอนโทรลเลอร์ขึ้นมาโดยใช้แรงดันไฟตรงบวก 12 โวลท์ แทนด้วยลอจิก "1" แรงดันไฟตรง 0 โวลท์ แทนด้วยลอจิก "0" แล้วทำการป้อนสัญญาณภาพและเสียงจากเครื่องเล่นวีดีโอเข้ามาทางอินพุทของเครื่อง และเอาท์พุทต่อไปแสดงผลยังจอมอนิเตอร์ ทำการเปลี่ยนแปลงสัญญาณควบคุมไปเป็นค่าต่างๆดังแสดงในตาราง

CONTROL SIGNAL	OUT PUT
00	CH.1
01	CH.2
10	CH.3
11	CH.4

โดยภายในตัวเครื่องมีหลอดจอร์ LED แสดงผลให้ทราบว่ามียินพุทเข้ามาหรือไม่ คือถ้ามียินพุทเข้ามาจะทำให้ LED ติด โดยจะทำการตรวจเช็คควบคู่ไปกับการป้อนสัญญาณภาพและเสียง ถ้าเราทำการป้อนสัญญาณทุกๆช่อง LED ก็จะมีติดทุกดวง ที่จอ LCD ก็จะมีการแสดงผลว่าเครื่องกำลังทำงานอยู่ในช่องใด และอยู่ในโหมดการทำงานใด โดยการเลือกโหมดการทำงาน , ช่องของสัญญาณ , เวลาในการทำงานของโหมดสแตนด์บายโนมีตี จะทำการดูได้อย่างละเอียดในขั้นตอนการใช้เครื่อง

ผลการทดลอง

จากการทดลองสรุปได้ว่า เมื่อทำการป้อนสัญญาณเข้ามาทางอินพุตทุกๆช่อง แล้วทำการส่งสัญญาณควบคุมจาก CPU ส่งไปควบคุมชุดวีดีโอและอะดิโอสวิตช์ ผลปรากฏว่าในการทดลองครั้งแรก สัญญาณควบคุมจาก CPU ไม่สามารถควบคุมชุดวีดีโอและอะดิโอได้เนื่องมาจาก CPU ส่งสัญญาณ "1" มาเป็นแรงดัน 5 โวลท์ ซึ่งชุดอนาล็อกสวิตช์จะมองแรงดัน 5 โวลท์นี้เป็น "0" ต้องมีการเพิ่มแรงดันเป็น 12 โวลท์จึงสามารถควบคุมชุดวีดีโอและอะดิโอได้ แต่ถ้าเราไม่ป้อนสัญญาณทุกช่องจะพบว่ามีการกวนสัญญาณข้ามช่องกัน โดยจะพบในช่องที่ไม่มีการป้อนของสัญญาณแต่อย่างไรก็ตามเมื่อป้อนสัญญาณครบทุกช่องการกวนก็จะมีผลทำให้ภาพและเสียงผิดเพี้ยนแต่อย่างไรก็ตามเพราะว่าสัญญาณที่ทำการกวนมาเป็นเพียงระดับสัญญาณเล็กๆ

อุปสรรคหรือปัญหาในการสร้าง

มักเกิดจากความไม่พร้อมในเรื่องเครื่องมือที่ใช้ในการวัดและทดสอบวงจร อาทิเช่น ออสซิลอสโคป , แพทเทินเจนเนอเรเตอร์ เป็นต้น และการหาอุปกรณ์ที่นำมาใช้ในการประกอบวงจร

ขั้นตอนในการใช้เครื่อง

1. สวิตช์ เปิด-ปิด ไฟที่ทำการจ่ายให้กับวงจรภายในตัวเครื่อง โดยตัวเครื่องนี้ได้ทำการติดตั้งสวิทช์สายไฟเป็นแบบหลอดนีออนเรืองแสงเพื่อให้ผู้ใช้งานได้ทราบว่าเครื่องกำลังทำงานอยู่
2. ปุ่มควบคุมการทำงาน ซึ่งออกแบบให้เป็นสวิตช์แบบกดติดปลั๊กยดัดบ ซึ่งมีอยู่ทั้งหมด 8 ปุ่ม มีการทำงานต่างๆ กันดังนี้
 - 2.1 ปุ่มเลือกการแสดงผลสัญญาณภาพและการแสดงผลสัญญาณเสียง ช่องที่ 1 โดยทำงานอยู่ในโหมด manual
 - 2.2 ปุ่มเลือกการแสดงผลสัญญาณภาพและการแสดงผลสัญญาณเสียง ช่องที่ 2 โดยทำงานอยู่ในโหมด manual
 - 2.3 ปุ่มเลือกการแสดงผลสัญญาณภาพและการแสดงผลสัญญาณเสียง ช่องที่ 3 โดยทำงานอยู่ในโหมด manual
 - 2.4 ปุ่มเลือกการแสดงผลสัญญาณภาพและการแสดงผลสัญญาณเสียง ช่องที่ 4 โดยทำงานอยู่ในโหมด manual
 - 2.5 ปุ่ม set time ผู้ใช้กดเพื่อตั้งเวลาในการแสดงผลภาพและเสียงของแต่ละช่อง ซึ่งปุ่มนี้สามารถกดได้เมื่อเครื่องทำงานอยู่ในโหมดสแกนอัตโนมัติ โดยทำงานร่วมกับปุ่ม select time
 - 2.6 ปุ่ม select time กดเมื่อต้องการเปลี่ยนค่าเวลาการแสดงผลภาพและเสียงของแต่ละช่อง โดยสามารถเลือกค่าเวลาได้ 4 ค่า คือ 5, 10 ,15, 20 วินาทีตามลำดับ
 - 2.7 ปุ่มเลือกโหมดการทำงานซึ่งผู้ใช้สามารถเลือกใช้ได้ 2 โหมดการทำงาน คือ โหมดสแกน และโหมด manual โดยกดครั้งหนึ่งจะเป็นโหมดสแกน กดซ้ำจะเป็นโหมด manual
 - 2.8 ปุ่ม start เป็นปุ่มสั่งให้เริ่มการทำงาน
3. จอ LCD แสดงข้อมูลต่างๆ ขณะที่เครื่องกำลังทำงาน
4. ปลั๊กเสียบสายสัญญาณ input และ output ของสัญญาณภาพและเสียง
5. กระจกฟิวส์ บรรจุฟิวส์ขนาด 1 แอมป์

เมื่อเริ่มจ่ายไฟเข้าเครื่อง CPU จะเริ่มตรวจสอบสัญญาณ VDO ว่าจะมีสัญญาณต่อเข้ามาที่ input ในช่องสัญญาณใดบ้าง โดยจอ LCD จะทำการแสดงข้อความว่า CH..Ready หากว่าช่องใดไม่มีสัญญาณ VDO ต่อเข้ามาทาง input จอ LCD ก็จะแสดงข้อความว่า CH..No signal โดยจอ LCD จะวนแสดงสถานะของแต่ละช่องสลับกันไป และภายในตัวเครื่อง จะมี LED สีแดง 4 ตัว ซึ่งออกแบบมาเพื่อแสดงสถานะของสัญญาณ VDO แต่ละช่อง ซึ่งจะมีประโยชน์เมื่อผู้ใช้งานในระยะไกล ซึ่งก็จะสามารถมองเห็นได้ว่า ในขณะที่ CPU จะรอการเริ่มทำงาน โดยผู้ใช้งานจะสามารถเริ่มการทำงานโดยการกดปุ่ม start เมื่อกดแล้ว เครื่องก็จะทำการเข้าไปทำงานในโหมด manual โดยต่อสัญญาณภาพไปแสดงที่จอมอนิเตอร์ ในโหมดการทำงานนี้ สามารถให้แสดงภาพและเสียงของช่องใดก็ได้ ที่มีสัญญาณต่อเข้ามายัง input ของเครื่อง โดยผู้ใช้เลือกกดปุ่ม 2.1-2.4 ได้ตามต้องการ หรือหากผู้ใช้งานต้องการที่จะเปลี่ยนโหมดการทำงานไปเป็นโหมดสแกนอัตโนมัติ ก็สามารถทำได้โดยการกดปุ่ม 2:7 เมื่อกดแล้วเครื่องจะเข้าไปทำงานในโหมดสแกนอัตโนมัติ โดยจะทำการแสดงภาพและเสียงของช่องต่างๆ วนกันไปเรื่อยๆ ตามที่ได้ทำการโปรแกรมเอาไว้ โดยระยะเวลาในการแสดงภาพและเสียงของแต่ละช่อง ตามที่เราได้ตั้งค่าไว้ โดยการตั้งค่านี้นี้เราจะกดปุ่ม 2.5 ซึ่งเป็นปุ่มตั้งค่าเวลาของการแสดงภาพและเสียงของแต่ละช่อง โดยจะทำงานร่วมกับปุ่ม 2.6 โดยปุ่ม 2.6 เป็นปุ่มในการเปลี่ยนค่าเวลาตามที่เราต้องการ เมื่อเลือกค่าเวลาได้แล้ว ก็กดปุ่ม 2.5 ซ้ำอีกครั้ง เครื่องก็จะไปทำงานในโหมดสแกนต่อไป.

รายการอุปกรณ์ของภาคต่างๆ

ภาคตัดต่อสัญญาณภาพ(VIDEO SWITCH)

ตัวต้านทานขนาด1/4วัตต์ ค่าผิดพลาด 5%

75Ω	4	ตัว
27Ω	4	ตัว
10KΩ	5	ตัว
100KΩ	4	ตัว
4.7KΩ	1	ตัว
750Ω	1	ตัว
1.2KΩ	1	ตัว
1.8KΩ	1	ตัว
18KΩ	1	ตัว
1.2KΩ	2	ตัว
470Ω	1	ตัว
390Ω	1	ตัว
ตัวเก็บประจุ		
47MF 25V	4	ตัว
22MF 25V	1	ตัว
47 pF	1	ตัว
ซีเนอร์ไดโอด		
7.5 V 1/4W	4	ตัว
ทรานซิสเตอร์		
2SA1175	4	ตัว
2SC2785	3	ตัว
ไอซี		
MC14052BCP	1	ตัว

ภาคตัดต่อสัญญาณเสียง(AUDIO SWITCH)

ตัวต้านทานขนาด1/4W ค่าผิดพลาด 5%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

100KΩ 10 ตัว

10K Ω	10	ตัว
3.3K Ω	2	ตัว
4.7K Ω	1	ตัว

ตัวเก็บประจุ

1MF 25V	8	ตัว
4.7MF 25V	2	ตัว
100pF	8	ตัว
0.001MF	2	ตัว

ไอซี

MC14052BCP	1	ตัว
RC4558P	1	ตัว

ภาคไมโครคอนโทรลเลอร์(MICROCONTROLLER)

ตัวต้านทานขนาด1/4W ค่าผิดพลาด5%

8.2K Ω	1	ตัว
---------------	---	-----

ตัวต้านทานปรับค่าได้

10K Ω แบบเก็อกม้า	1	ตัว
--------------------------	---	-----

ตัวต้านทานแบบ NETWORK

10K Ω 9PIN COMMON	4	ตัว
--------------------------	---	-----

ตัวเก็บประจุ

30pF	2	ตัว
------	---	-----

ตัวกำเนิดความถี่

11.592Mz	1	ตัว
----------	---	-----

ไอซี

4081	2	ตัว
------	---	-----

AT89C51	1	ตัว
---------	---	-----

คอนเน็คเตอร์แถวเดี่ยว	2	แถว
-----------------------	---	-----

จอแสดงผล DOTMTRIC LCD 1แถว 16 คอลัมน์	1	ตัว
---------------------------------------	---	-----

สวิทช์กดติดปล่อยดับ	8	ตัว
---------------------	---	-----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคแสดงสถานะสัญญาณภาพอินพุท

ตัวต้านทานขนาด1/4Wค่าผิดพลาด5%

2K Ω	4	ตัว
100K Ω	4	ตัว
15K Ω	4	ตัว
180K Ω	4	ตัว
1N4148	4	ตัว
LED	4	ตัว
1K Ω	4	ตัว
560 Ω	4	ตัว
ไดโอด		
1N4148	4	ตัว
ซีเนอร์ไดโอด		
2.7V 1/4W	8	ตัว
ตัวเก็บประจุ		
0.47MF	4	ตัว
ทรานซิสเตอร์		
BC548	8	ตัว

ภาคแหล่งจ่ายไฟ(POWER SUPPLY)

ตัวต้านทานขนาด1/4Wค่าผิดพลาด5%

330 Ω	1	ตัว
ไดโอด		
1N4001	2	ตัว
แบบBRIDGE	1	ตัว
LED	1	ตัว
ตัวเก็บประจุ		
3300MF 25V	1	ตัว

220MF 25V

2

ตัว

0.1MF

2

ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการวิชาการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซี

LM7812	1	ตัว
LM7805	1	ตัว
หม้อแปลง 12VAC500mA	1	ตัว
ฟิวส์และกระบอกฟิวส์	1	ตัว
สายไฟAC	1	เส้น
สวิตช์ACมีหลอดไฟในตัว	1	อัน

กล่องลงอุปกรณ์ 1 ใบ
แผ่นปรินท์
ชอคเก็ตไอซี





ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

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

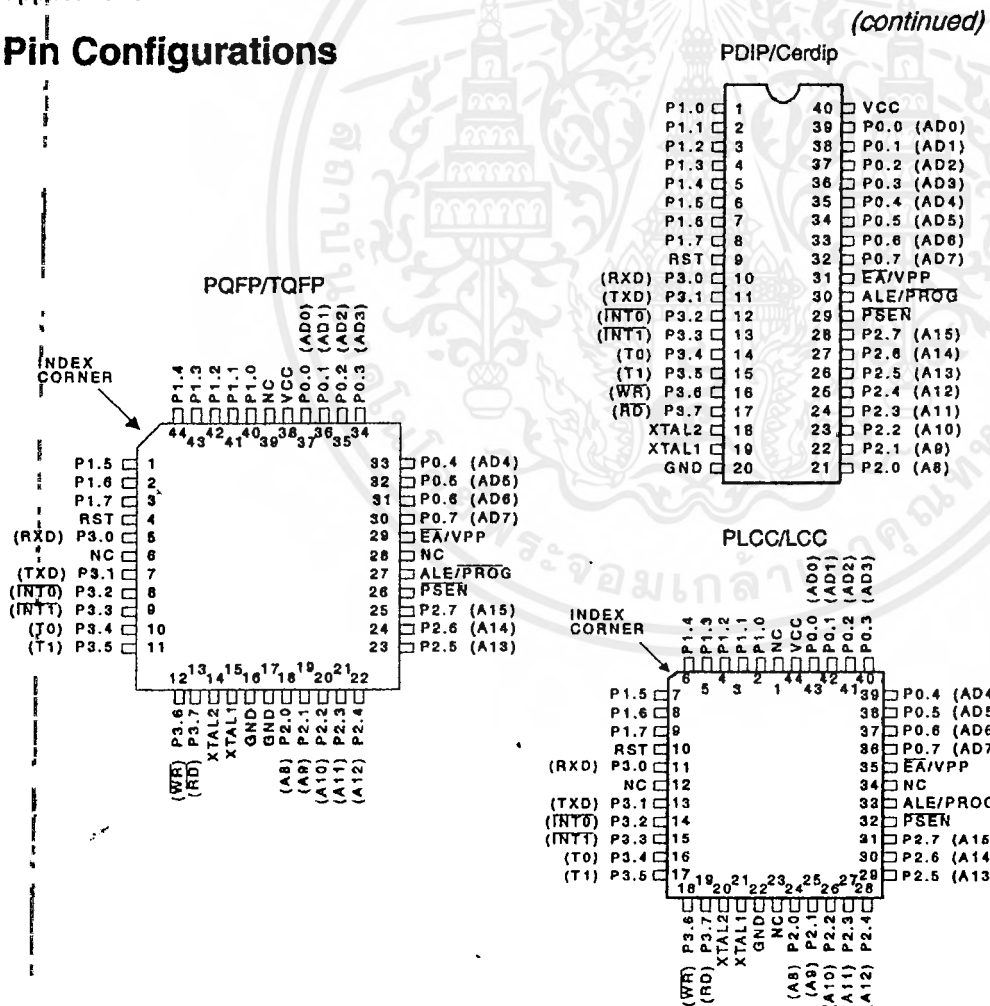
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 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 MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

8-Bit Microcontroller with 4 Kbytes Flash

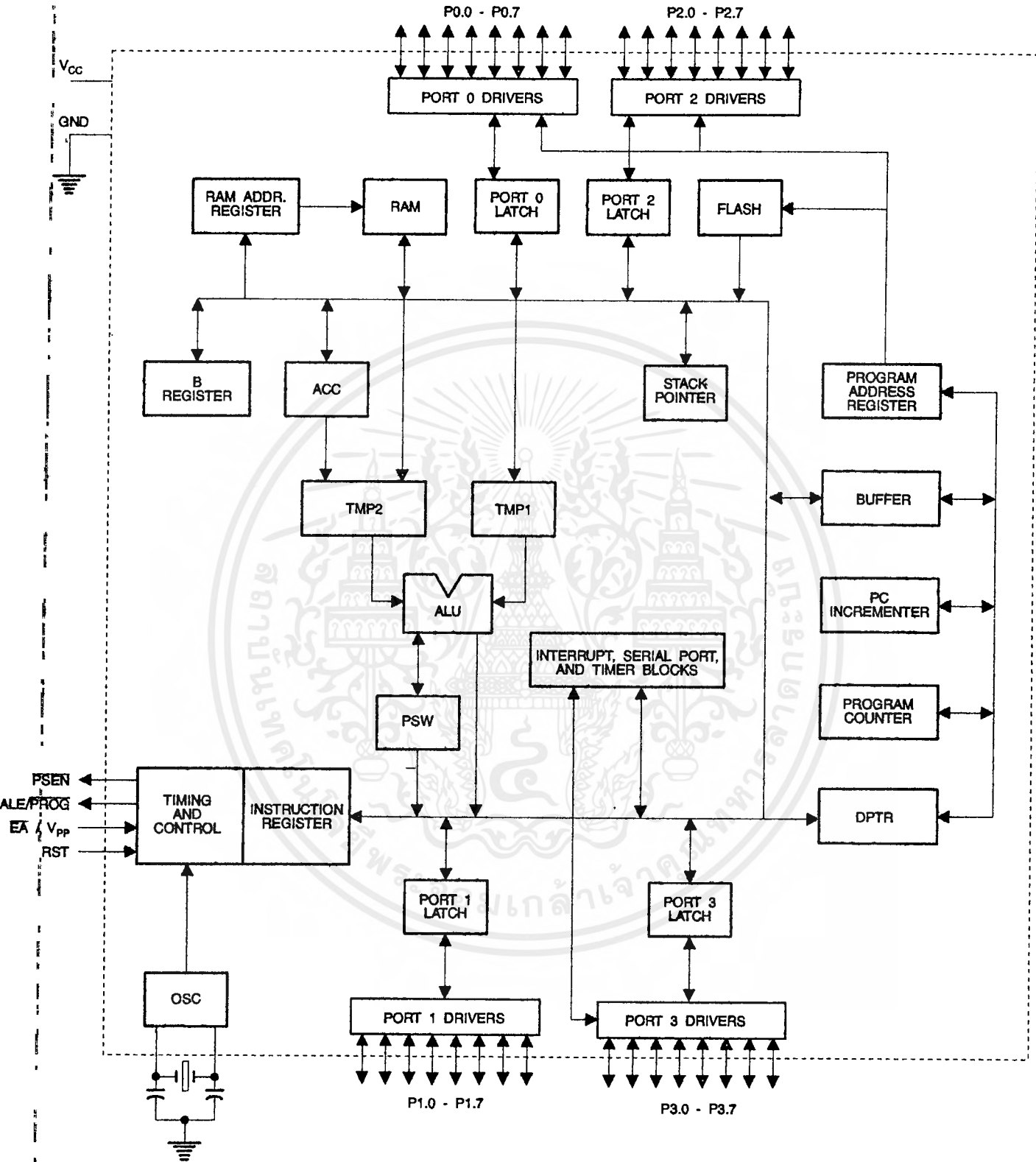
AT89C51

Pin Configurations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Description (Continued)

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

Pin Description

V_{CC}
Supply voltage.

GND
Ground.

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

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

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

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

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

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

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX

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

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

Port 3
Port 3 is an 8-bit 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 listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	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/PROG
Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

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

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

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

(continued)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

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

\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, for parts that require 12-volt V_{PP} .

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

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

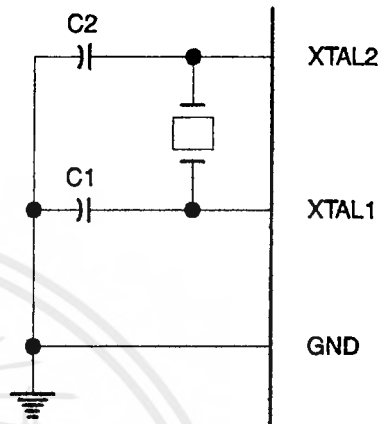
Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this

mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

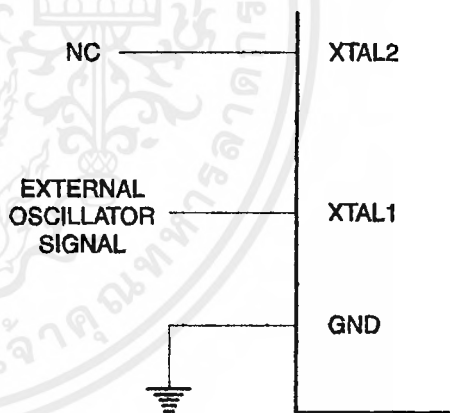
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hard-

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	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

ware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

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 Vcc

is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

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

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

Lock Bit Protection Modes

Program Lock Bits				Protection Type
LB1	LB2	LB3		
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 is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

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

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

	Vpp = 12 V	Vpp = 5 V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

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

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

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise EA/Vpp to 12 V for the high-voltage programming mode.
5. Pulse ALE/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 AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an at-



Programming the Flash (Continued)

tempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

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

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

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

Flash Programming Modes

Mode	RST	PSEN	ALE/ PROG	EA/ V _{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V ⁽¹⁾	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	H	L		H/12V	H	H	H	H
				H/12V	H	H	L	L
				H/12V	H	L	H	L
Chip Erase	H	L		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 $\overline{\text{PROG}}$ pulse.

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

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel

(031H) = 51H indicates 89C51

(032H) = FFH indicates 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.

Figure 3. Programming the Flash

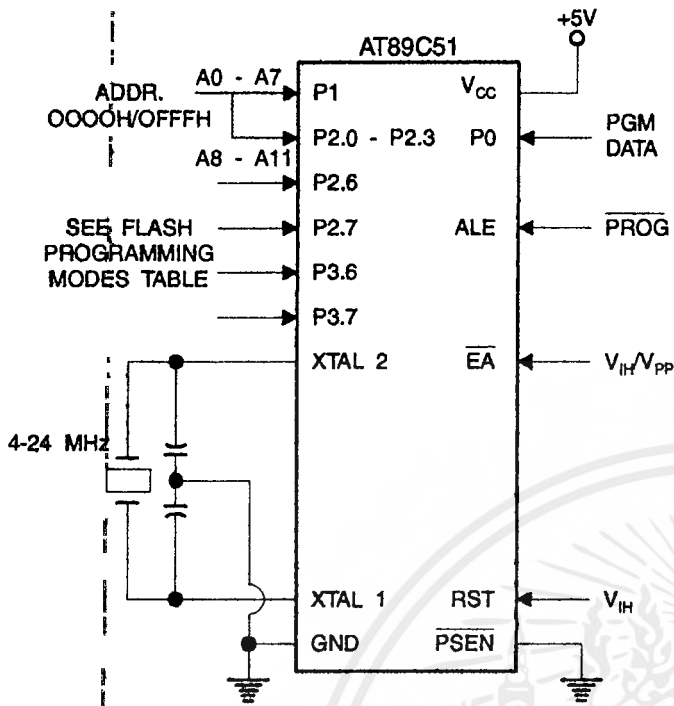
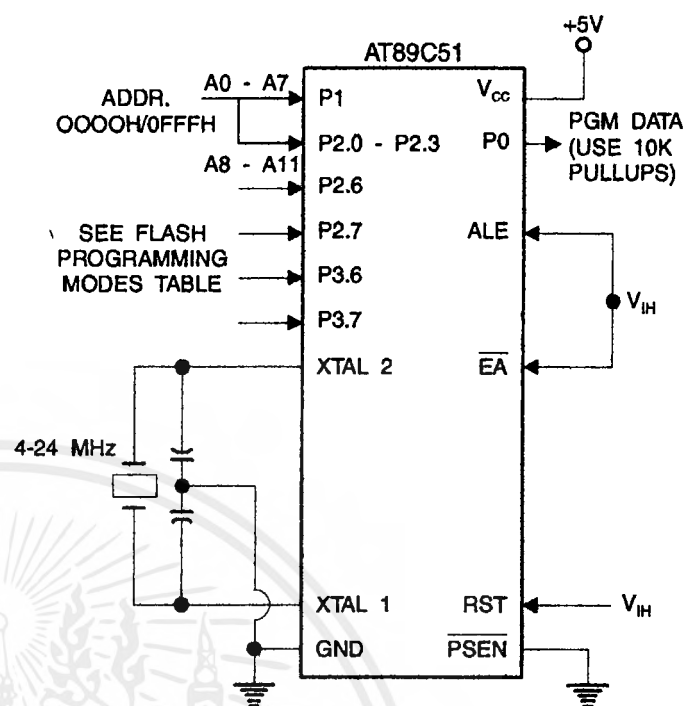


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

$T_A = 21^\circ\text{C to } 27^\circ\text{C}$, $V_{CC} = 5.0 \pm 10\%$

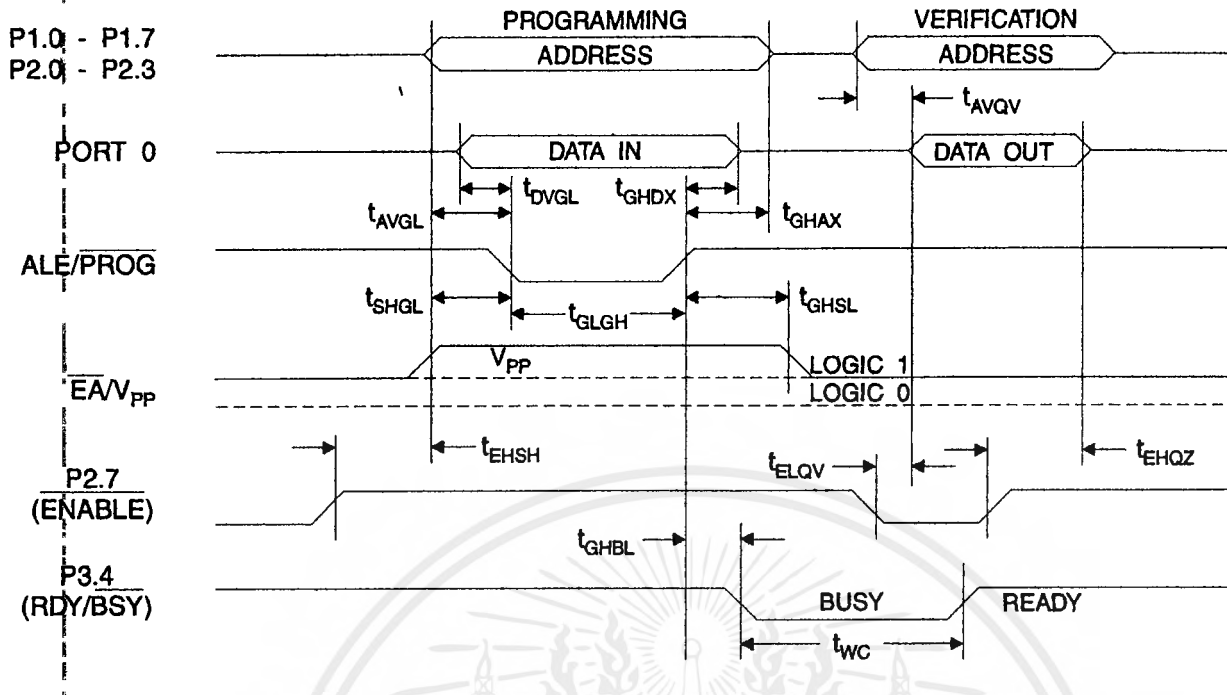
Symbol	Parameter	Min	Max	Units
$V_{PP}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}^{(1)}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 (ENABLE) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
$t_{GHSL}^{(1)}$	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	ENABLE Low to Data Valid		$48t_{CLCL}$	
t_{EHQV}	Data Float After ENABLE	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms

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

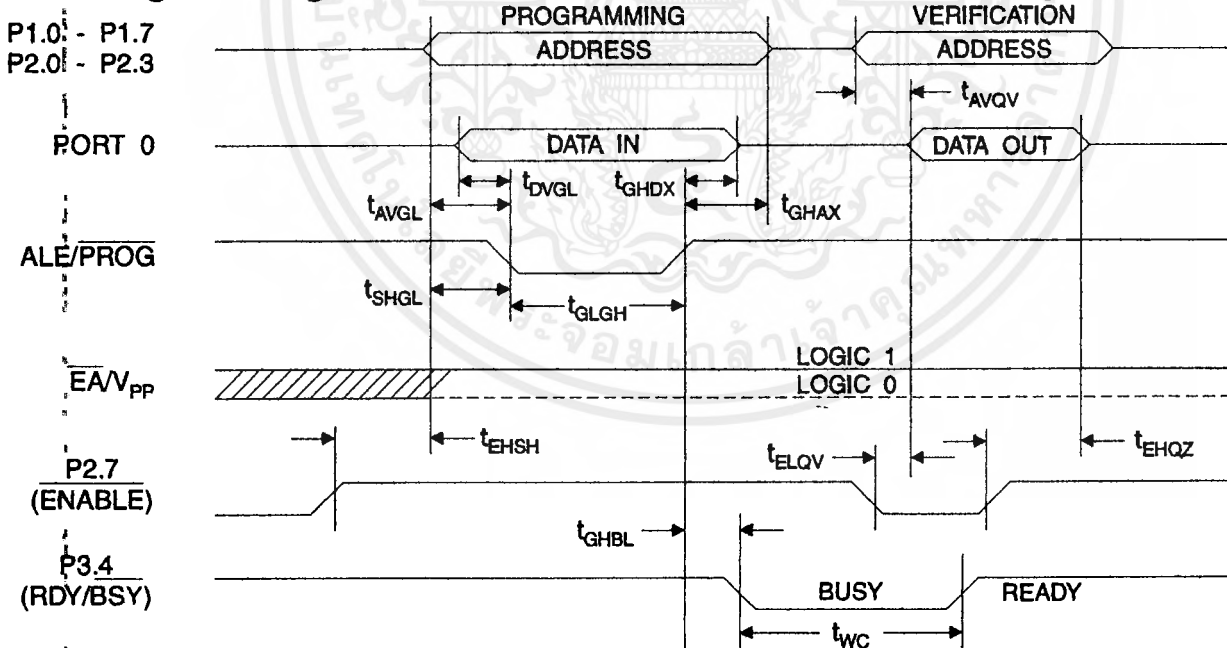


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 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

T_A = -40°C to 85°C, 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, PSEN)	I _{OL} = 3.2 mA		0.45	V
V _{OH}	Output High Voltage (Ports 1,2,3, ALE, 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 _I	Input Leakage Current (Port 0; 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		20	mA
		Idle Mode, 12 MHz		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, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
 2. Minimum V_{CC} for Power Down is 2 V.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 *ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

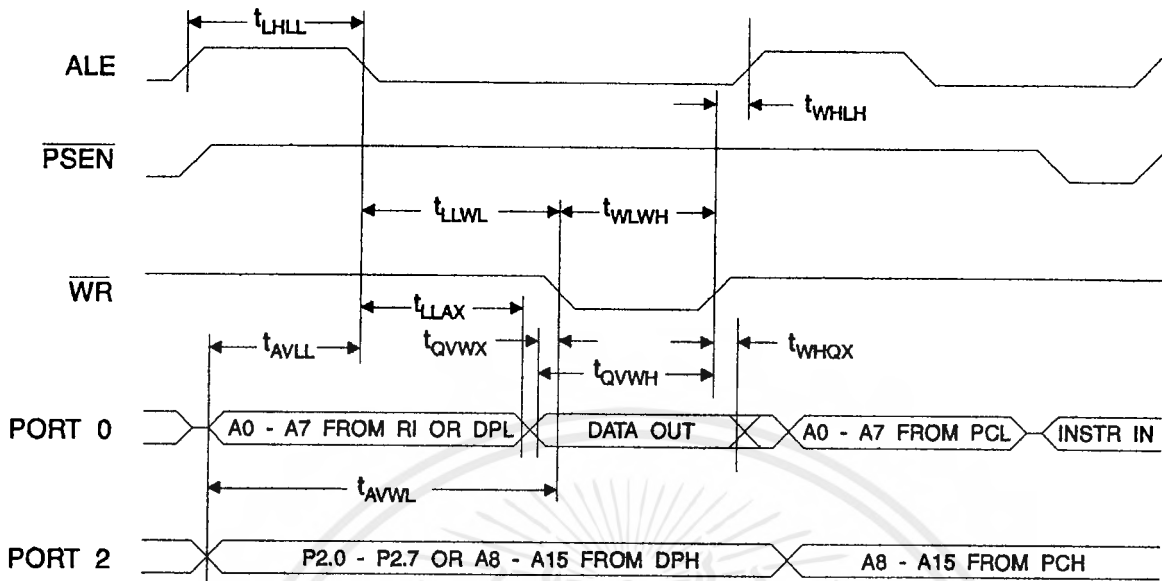
A.C: Characteristics

(Under Operating Conditions; Load Capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; Load Capacitance for all other outputs = 80 pF)

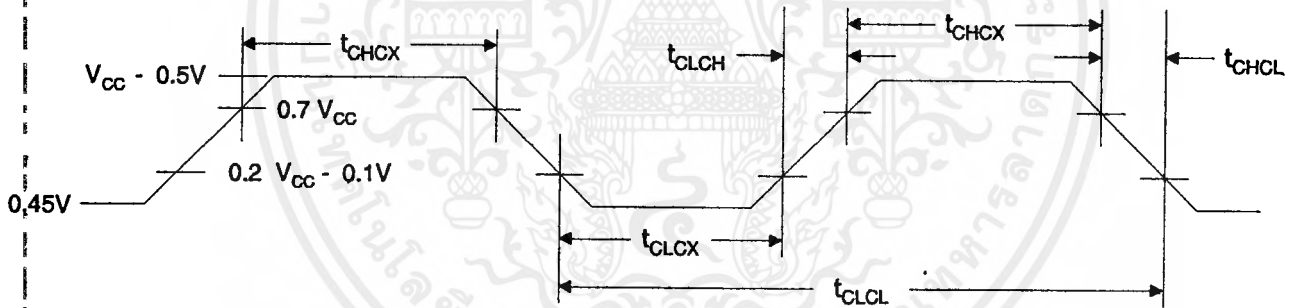
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
1/tCLCL	Oscillator Frequency			0	24	MHz
tLHL	ALE Pulse Width	127		2tCLCL-40		ns
tAVLL	Address Valid to ALE Low	28		tCLCL-13		ns
tLLAX	Address Hold After ALE Low	48		tCLCL-20		ns
tLLIV	ALE Low to Valid Instruction In		233		4tCLCL-65	ns
tLLPL	ALE Low to PSEN Low	43		tCLCL-13		ns
tPLPH	PSEN Pulse Width	205		3tCLCL-20		ns
tPLIV	PSEN Low to Valid Instruction In		145		3tCLCL-45	ns
tpXIX	Input Instruction Hold After PSEN	0		0		ns
tpXIZ	Input Instruction Float After PSEN		59		tCLCL-10	ns
tpXAV	PSEN to Address Valid	75		tCLCL-8		ns
tAVIV	Address to Valid Instruction In		312		5tCLCL-55	ns
tPLAZ	PSEN Low to Address Float		10		10	ns
tRLRH	RD Pulse Width	400		6tCLCL-100		ns
tWLWH	WR Pulse Width	400		6tCLCL-100		ns
tRLDV	RD Low to Valid Data In		252		5tCLCL-90	ns
tRHDZ	Data Hold After RD	0		0		ns
tRHDZ	Data Float After RD		97		2tCLCL-28	ns
tLLDV	ALE Low to Valid Data In		517		8tCLCL-150	ns
tAVDV	Address to Valid Data In		585		9tCLCL-165	ns
tLLWL	ALE Low to RD or WR Low	200	300	3tCLCL-50	3tCLCL+50	ns
tAVWL	Address to RD or WR Low	203		4tCLCL-75		ns
tQVWX	Data Valid to WR Transition	23		tCLCL-20		ns
tQVWH	Data Valid to WR High	433		7tCLCL-120		ns
tWHQX	Data Hold After WR	33		tCLCL-20		ns
tRLAZ	RD Low to Address Float		0		0	ns
tWHLH	RD or WR High to ALE High	43	123	tCLCL-20	tCLCL+25	ns

External Data Memory Cycle



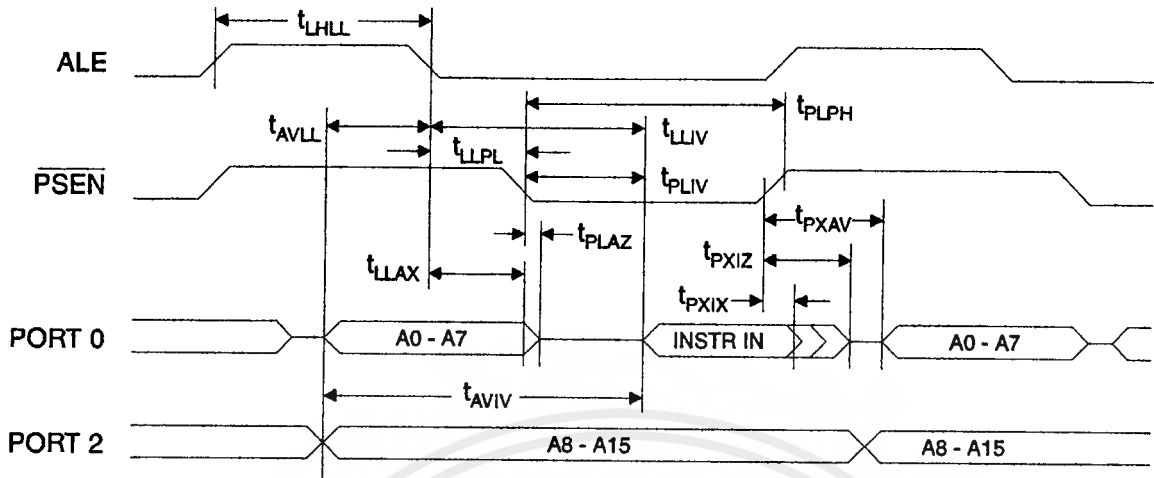
External Clock Drive Waveforms



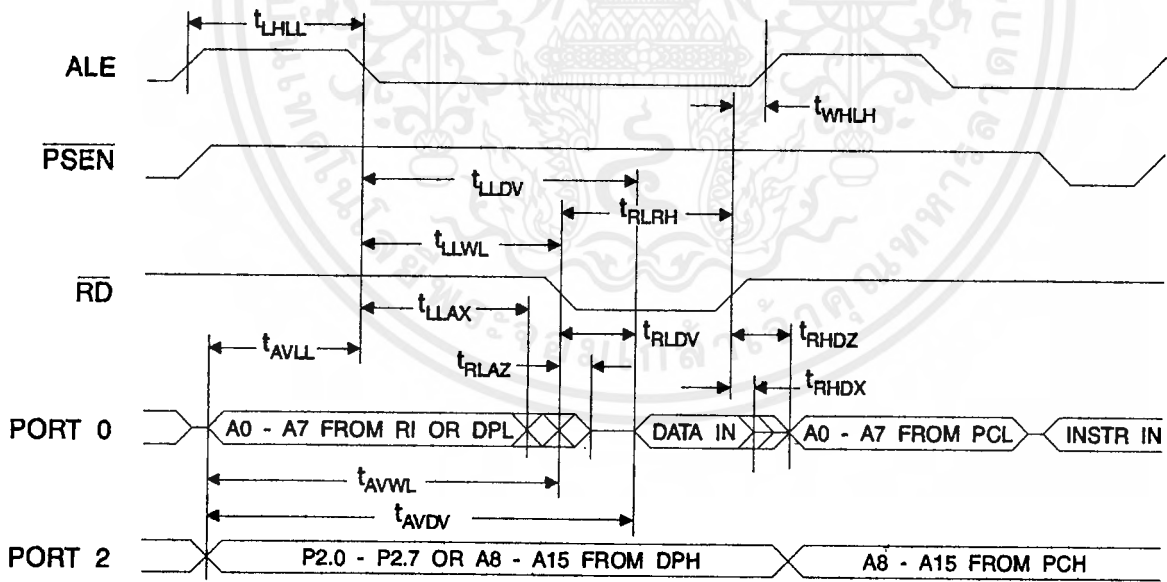
External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLL}$	Oscillator Frequency	0	24	MHz
t_{CLL}	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

External Program Memory Read Cycle



External Data Memory Read Cycle



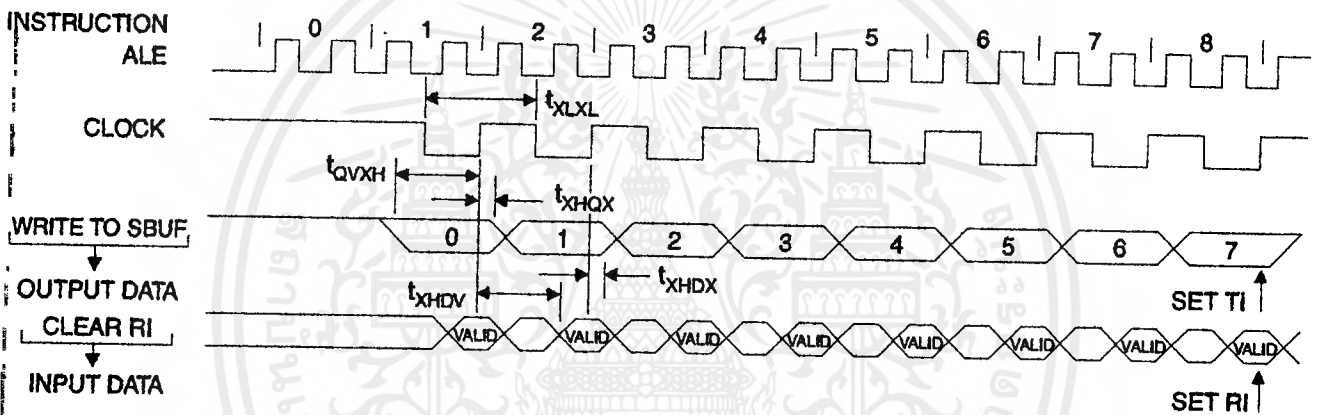
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

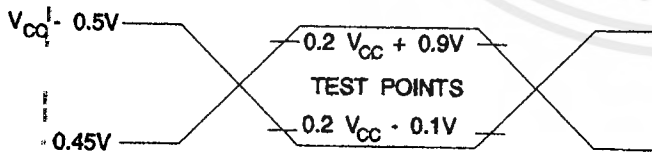
($V_{CC} = 5.0\text{ V} \pm 20\%$; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHGX}	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

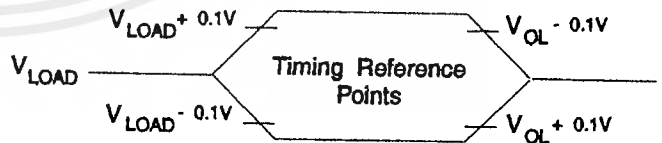
Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms ⁽¹⁾



Float Waveforms ⁽¹⁾



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

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



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range	
12	5 V ± 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)	
		AT89C51-12JC	44J		
		AT89C51-12PC	40P6		
		AT89C51-12QC	44Q		
		AT89C51-12AI	44A		
		AT89C51-12JI	44J		
	5 V ± 10%	AT89C51-12PI	40P6	Industrial (-40°C to 85°C)	
		AT89C51-12QI	44Q		
		AT89C51-12AA	44A		Automotive (-40°C to 125°C)
		AT89C51-12JA	44J		
		AT89C51-12PA	40P6		
		AT89C51-12QA	44Q		
AT89C51-12DM	40D6				
AT89C51-12LM	44L				
16	5 V ± 20%	AT89C51-12DM/883	40D6	Military (-55°C to 125°C)	
		AT89C51-12LM/883	44L		
		AT89C51-16AC	44A		Commercial (0°C to 70°C)
		AT89C51-16JC	44J		
		AT89C51-16PC	40P6		
		AT89C51-16QC	44Q		
	AT89C51-16AI	44A			
	AT89C51-16JI	44J			
	5 V ± 20%	AT89C51-16PI	40P6	Industrial (-40°C to 85°C)	
		AT89C51-16QI	44Q		
		AT89C51-16AA	44A		Automotive (-40°C to 125°C)
		AT89C51-16JA	44J		
AT89C51-16PA		40P6			
AT89C51-16QA		44Q			
20	5 V ± 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)	
		AT89C51-20JC	44J		
		AT89C51-20PC	40P6		
		AT89C51-20QC	44Q		
		AT89C51-20AI	44A	Industrial (-40°C to 85°C)	
		AT89C51-20JI	44J		
		AT89C51-20PI	40P6		
		AT89C51-20QI	44Q		

AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

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



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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Advance Information

Analog Multiplexers/
Demultiplexers

High-Performance Silicon-Gate CMOS

The MC54/74HC4051, MC54/74HC4052, and MC54/74HC4053 utilize silicon-gate CMOS technology to achieve fast propagation delays, low ON resistances, and low OFF leakage currents. These analog multiplexers/demultiplexers control analog voltages that may vary across the complete power supply range (from V_{CC} to V_{EE}).

The HC4051, HC4052, and HC4053 are identical in pinout to the metal-gate MC14051B, MC14052B, and MC14053B. The Channel-Select inputs determine which one of the Analog Inputs/Outputs is to be connected, by means of an analog switch, to the Common Output/Input. When the Enable pin is high, all analog switches are turned off.

The Channel-Select and Enable inputs are compatible with standard CMOS outputs; with pulldown resistors, they are compatible with LSTTL outputs.

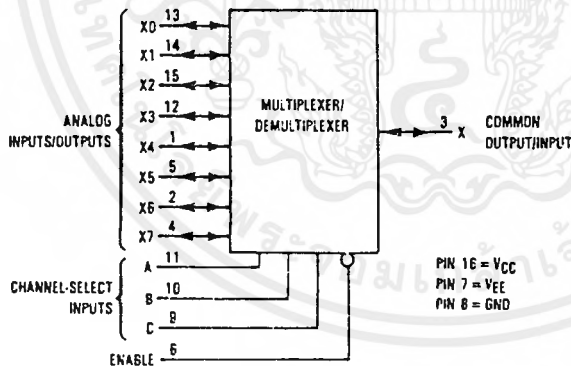
These devices have been designed so that the ON resistance (R_{ON}) is more linear over input voltage than R_{ON} of metal-gate CMOS analog switches.

For multiplexers/demultiplexers with channel select latches, see HC4351, HC4352, and HC4353.

- Fast Switching and Propagation Speeds
- Low Crosstalk Between Switches
- Diode Protection on All Inputs/Outputs
- Analog Power Supply Range ($V_{CC} - V_{EE}$) = 2.0 to 12.0 V
- Digital (Control) Power Supply Range ($V_{CC} - GND$) = 2.0 to 6.0 V
- Improved Linearity and Lower ON Resistance than Metal-Gate Counterparts
- Low Noise
- In Compliance with the Requirements Defined by JEDEC Standard No. 7A
- Chip Complexity: HC4051—184 FETs or 46 Equivalent Gates
HC4052—168 FETs or 42 Equivalent Gates
HC4053—156 FETs or 39 Equivalent Gates

LOGIC DIAGRAM
MC54/74HC4051

Single-Pole, 8-Position Plus Common Off



PIN 16 = V_{CC}
PIN 7 = V_{EE}
PIN 8 = GND

MC54/74HC4051
MC54/74HC4052
MC54/74HC4053



J SUFFIX
CERAMIC
CASE 620-09



N SUFFIX
PLASTIC
CASE 648-08



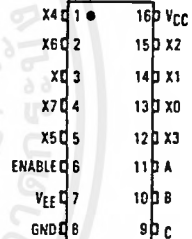
DW SUFFIX
SOIC
CASE 751G-01

ORDERING INFORMATION

MC74HCXXXXN Plastic
MC54HCXXXXJ Ceramic
MC74HCXXXXDW SOIC

$T_A = -55^\circ$ to 125°C for all packages.
Dimensions in Chapter 6.

PIN ASSIGNMENT
MC54/74HC4051



FUNCTION TABLE
MC54/74HC4051

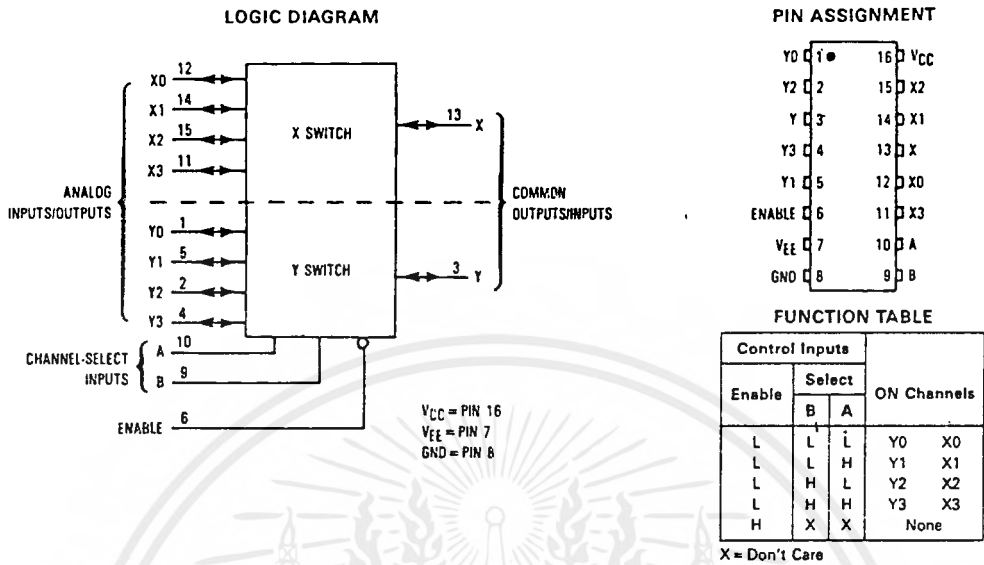
Control Inputs				ON Channels
Enable	Select			
	C	B	A	
L	L	L	L	X0
L	L	L	H	X1
L	L	H	L	X2
L	L	H	H	X3
L	H	L	L	X4
L	H	L	H	X5
L	H	H	L	X6
L	H	H	H	X7
H	X	X	X	None

X = don't care

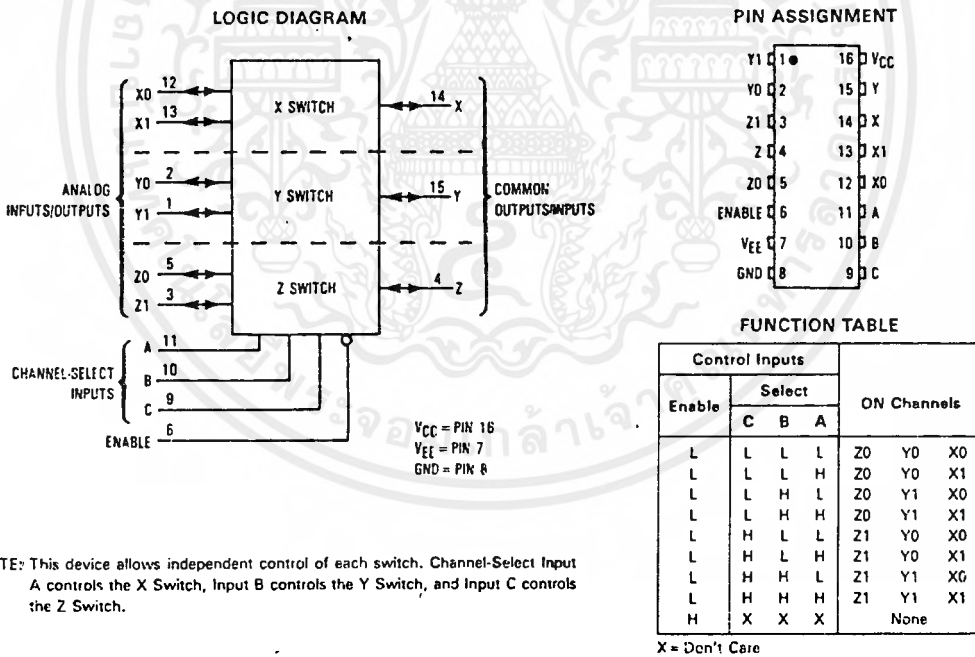
This document contains information on a new product. Specifications and information herein are subject to change without notice.

MC54/74HC4051 • MC54/74HC4052 • MC54/74HC4053

MC54/74HC4052
Double-Pole, 4-Position
Plus Common Off



MC54/74HC4053
Triple Single-Pole, Double-Position
Plus Common Off



5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC54/74HC4051 • MC54/74HC4052 • MC54/74HC4053

MAXIMUM RATINGS*

Symbol	Parameter	Value	Unit
V _{CC}	Positive DC Supply Voltage (Ref. to GND) (Ref. to V _{EE})	-0.5 to +7.0 -0.5 to 14.0	V
V _{EE}	Negative DC Supply Voltage (Ref. to GND)	-7.0 to +0.5	V
V _{IS}	Analog Input Voltage	V _{EE} - 0.5 to V _{CC} + 0.5	V
V _{in}	Digital Input Voltage (Ref. to GND)	-1.5 to V _{CC} + 1.5	V
I	DC Current Into or Out of Any Pin	±25	mA
P _D	Power Dissipation in Still Air Plastic or Ceramic DIP† SOIC Package†	750 500	mW
T _{stg}	Storage Temperature	-65 to +150	°C
T _L	Lead Temperature, 1 mm from Case for 10 Seconds (Plastic DIP or SOIC Package) (Ceramic DIP)	260 300	°C

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{in} and V_{out} should be constrained to the ranges indicated in the Recommended Operating Conditions.

Unused digital input pins must be tied to an appropriate logic voltage level (e.g., either GND or V_{CC}). Unused Analog I/O pins may be left open or terminated. See Applications Information.

*Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the Recommended Operating Conditions.

†Power Dissipation Temperature Derating:

Plastic "N" Package: -10 mW/°C from 65° to 85°C
Ceramic "J" Package: -10 mW/°C from 100° to 125°C
SOIC "D" Package: -7 mW/°C from 65° to 85°C

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit	
V _{CC}	Positive DC Supply Voltage (Ref. to GND) (Ref. to V _{EE})	2.0 2.0	6.0 12.0	V	
V _{EE}	Negative DC Supply Voltage (Ref. to GND)	-6.0	GND	V	
V _{IS}	Analog Input Voltage	V _{EE}	V _{CC}	V	
V _{in}	Digital Input Voltage (Ref. to GND)	GND	V _{CC}	V	
V _{IO} *	Static or Dynamic Voltage Across Switch	-	1.2	V	
T _A	Operating Temperature, All Package Types	-55	+125	°C	
t _r , t _f	Input Rise and Fall Time, (Channel Select or Enable Inputs)	V _{CC} = 2.0 V V _{CC} = 4.5 V V _{CC} = 6.0 V	0 0 0	1000 500 400	ns

*For voltage drops across the switch greater than 1.2 V (switch on), excessive V_{CC} current may be drawn; i.e., the current out of the switch may contain both V_{CC} and switch input components. The reliability of the device will be unaffected unless the Maximum Ratings are exceeded.

DC ELECTRICAL CHARACTERISTICS Digital Section (Voltages Referenced to GND) V_{EE} = GND, Except Where Noted

Symbol	Parameter	Test Conditions	V _{CC} V	Guaranteed Limit			Unit
				25°C to -55°C	≤85°C	≤125°C	
V _{IH}	Minimum High-Level Input Voltage, Channel-Select or Enable Inputs	R _{on} = Per Spec	2.0	1.5	1.5	1.5	V
			4.5	3.15	3.15	3.15	
			6.0	4.2	4.2	4.2	
V _{IL}	Maximum Low-Level Input Voltage, Channel-Select or Enable Inputs	R _{on} = Per Spec	2.0	0.3	0.3	0.3	V
			4.5	0.9	0.9	0.9	
			6.0	1.2	1.2	1.2	
I _{in}	Maximum Input Leakage Current, Channel-Select or Enable Inputs	V _{in} = V _{CC} or GND, V _{EE} = -6.0 V	6.0	±0.1	±1.0	±1.0	μA
I _{CC}	Maximum Quiescent Supply Current (per Package)	Channel Select = V _{CC} or GND Enable = V _{CC} or GND V _{IS} = V _{CC} or GND V _{IO} = 0 V					μA
			V _{EE} = GND	6.0	2	20	
		V _{EE} = -6.0	6.0	8	80	160	

NOTE: Information on typical parametric values can be found in Chapter 4.

5

MOTOROLA HIGH-SPEED CMOS LOGIC DATA

5-470

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC54/74HC4051 • MC54/74HC4052 • MC54/74HC4053

DC ELECTRICAL CHARACTERISTICS Analog Section

Symbol	Parameter	Test Conditions	V _{CC}	V _{EE}	Guaranteed Limit			Unit	
					25°C to -55°C	≤ 85°C	≤ 125°C		
R _{on}	Maximum "ON" Resistance	V _{in} = V _{IL} or V _{IH} V _{IS} = V _{CC} to V _{EE} I _S ≤ 2.0 mA (Figures 1, 2)	4.5	0.0	190	240	280	Ω	
			4.5	-4.5	120	150	170		
			6.0	-6.0	100	125	140		
		4.5	0.0	150	190	230			
		4.5	-4.5	100	125	140			
		6.0	-6.0	80	100	115			
ΔR _{on}	Maximum Difference in "ON" Resistance Between Any Two Channels in the Same Package	V _{in} = V _{IL} or V _{IH} V _{IS} = 1/2 (V _{CC} - V _{EE}) I _S ≤ 2.0 mA	4.5	0.0	30	35	40	Ω	
			4.5	-4.5	12	15	18		
			6.0	-6.0	10	12	14		
I _{off}	Maximum Off-Channel Leakage Current, Any One Channel	V _{in} = V _{IL} or V _{IH} V _{IO} = V _{CC} - V _{EE} Switch Off (Figure 3)	6.0	-6.0	0.1	0.5	1.0	μA	
			Maximum Off-Channel Leakage Current, Common Channel						
	HC4051	6.0	-6.0	0.2	2.0	4.0			
	HC4052	6.0	-6.0	0.1	1.0	2.0			
I _{on}	Maximum On-Channel Leakage Current, Channel to Channel	V _{in} = V _{IL} or V _{IH} Switch to Switch = V _{CC} - V _{EE} (Figure 5)	6.0	-6.0	0.2	2.0	4.0	μA	
			HC4052	6.0	-6.0	0.1	1.0		2.0
			HC4053	6.0	-6.0	0.1	1.0		2.0
			6.0	-6.0	0.1	1.0	2.0		

AC ELECTRICAL CHARACTERISTICS (C_L = 50 pF, Input t_r = t_f = 6 ns)

Symbol	Parameter	V _{CC} V	Guaranteed Limit			Unit	
			25°C to -55°C	≤ 85°C	≤ 125°C		
t _{PLH} , t _{PHL}	Maximum Propagation Delay, Channel-Select to Analog Output (Figure 9)	2.0	370	465	550	ns	
		4.5	74	93	110		
		6.0	63	79	94		
t _{PLH} , t _{PHL}	Maximum Propagation Delay, Analog Input to Analog Output (Figure 10)	2.0	60	75	90	ns	
		4.5	12	15	18		
		6.0	10	13	15		
t _{PLZ} , t _{PHZ}	Maximum Propagation Delay, Enable to Analog Output (Figure 11)	2.0	290	364	430	ns	
		4.5	58	73	86		
		6.0	49	62	73		
t _{PZL} , t _{PZH}	Maximum Propagation Delay, Enable to Analog Output (Figure 11)	2.0	345	435	515	ns	
		4.5	69	87	103		
		6.0	59	74	87		
C _{in}	Maximum Input Capacitance, Channel-Select or Enable Inputs	-	10	10	10	pF	
C _{I/O}	Maximum Capacitance	Analogue I/O	-	35	35	35	pF
		Common O/I: HC4051 HC4052 HC4053	-	130	130	130	
			-	80	80	80	
			-	50	50	50	
Feedthrough	-	1.0	1.0	1.0			

NOTES:

- For propagation delays with loads other than 50 pF, see Chapter 4.
- Information on typical parametric values can be found in Chapter 4.

C _{PD}	Power Dissipation Capacitance (Per Package) (Figure 13) Used to determine the no-load dynamic power consumption: $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$ For load considerations, see Chapter 4.	Typical @ 25°C, V _{CC} = 5.0 V, V _{EE} = 0 V			pF
		45 (HC4051)	80 (HC4052)	45 (HC4053)	

5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC54/74HC4051 • MC54/74HC4052 • MC54/74HC4053

ADDITIONAL APPLICATION CHARACTERISTICS (GND = 0.0 V)

Symbol	Parameter	Test Condition	VCC V	VEE V	Limit*			Unit
					25°C 54/74HC			
BW	Maximum On-Channel Bandwidth or Minimum Frequency Response (Figure 6)	$f_{in} = 1$ MHz Sine Wave Adjust f_{in} Voltage to Obtain 0 dBm at V_{OS} Increase f_{in} Frequency Until dB Meter Reads -3 dB $R_L = 50 \Omega, C_L = 10$ pF	2.25	-2.25	80	95	120	MHz
			4.50	-4.50	80	95	120	
-	Off-Channel Feedthrough Isolation (Figure 7)	$f_{in} =$ Sine Wave Adjust f_{in} Voltage to Obtain 0 dBm at V_{IS} $f_{in} = 10$ kHz, $R_L = 600 \Omega, C_L = 50$ pF	2.25	-2.25				dB
			4.50	-4.50			-50	
		$f_{in} = 1.0$ MHz, $R_L = 50 \Omega, C_L = 10$ pF	2.25	-2.25			-40	
			4.50	-4.50			-40	
-	Feedthrough Noise, Channel Select Input to Common O/I (Figure 8)	$V_{in} \leq 1$ MHz Square Wave ($t_r = t_f = 6$ ns) Adjust R_L at Setup so that $I_S = 0$ A Enable = GND $R_L = 600 \Omega, C_L = 50$ pF	2.25	-2.25			25	mVpp
			4.50	-4.50			105	
		$R_L = 10$ k $\Omega, C_L = 10$ pF	2.25	-2.25			35	
			4.50	-4.50			145	
-	Crosstalk Between Any Two Switches (Figure 12) (Test does not apply to HC4051)	$f_{in} =$ Sine Wave Adjust f_{in} Voltage to Obtain 0 dBm at V_{IS} $f_{in} = 10$ kHz, $R_L = 600 \Omega, C_L = 50$ pF	2.25	-2.25			-50	dB
			4.50	-4.50			-50	
		$f_{in} = 1$ MHz, $R_L = 50 \Omega, C_L = 10$ pF	2.25	-2.25			-60	
			4.50	-4.50			-60	
THD	Total Harmonic Distortion (Figure 14)	$f_{in} = 1$ kHz, $R_L = 10$ k $\Omega, C_L = 50$ pF THD = THD _{Measured} - THD _{Source} $V_{IS} = 4.0$ Vpp sine wave $V_{IS} = 8.0$ Vpp sine wave $V_{IS} = 11.0$ Vpp sine wave	2.25	-2.25			0.10	
			4.50	-4.50			0.08	
			6.00	-6.00			0.05	

*Limits not tested. Determined by design and verified by qualification.

5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

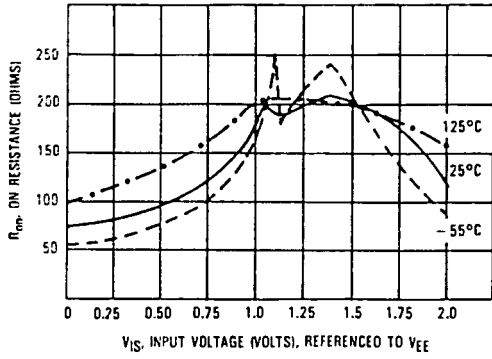


Figure 1a. Typical On Resistance, $V_{CC} - V_{EE} = 2.0$ V

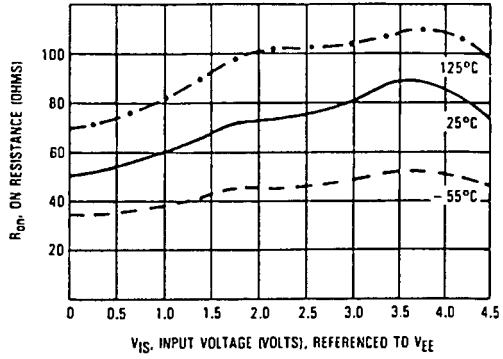


Figure 1b. Typical On Resistance, $V_{CC} - V_{EE} = 4.5$ V

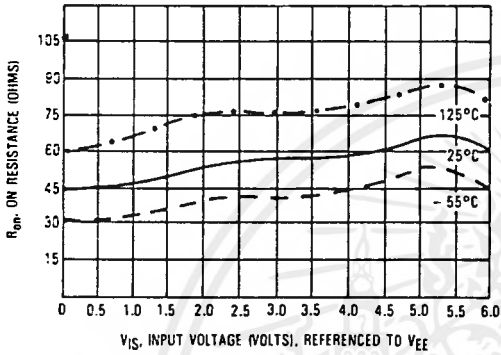


Figure 1c. Typical On Resistance, $V_{CC} - V_{EE} = 6.0$ V

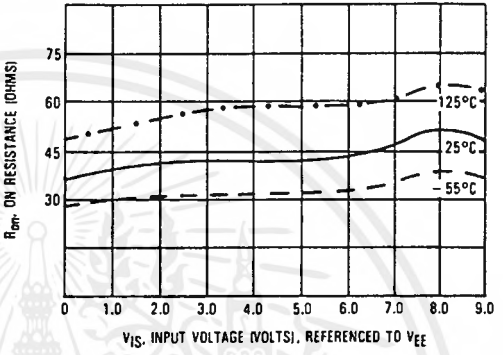


Figure 1d. Typical On Resistance, $V_{CC} - V_{EE} = 9.0$ V

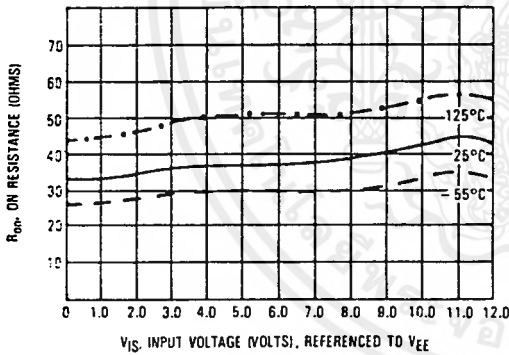


Figure 1e. Typical On Resistance, $V_{CC} - V_{EE} = 12.0$ V

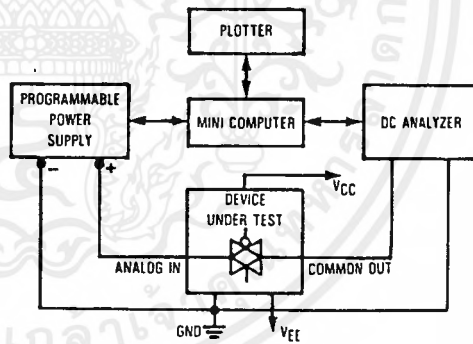


Figure 2. On Resistance Test Set-Up

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC54/74HC4051 • MC54/74HC4052 • MC54/74HC4053

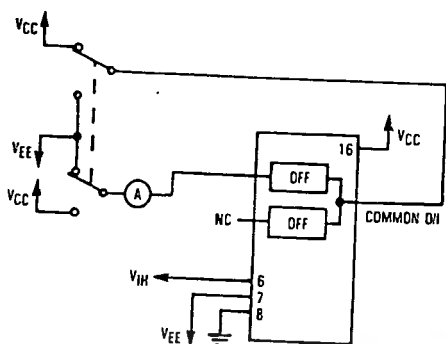


Figure 3. Maximum Off Channel Leakage Current, Any One Channel, Test Set-Up

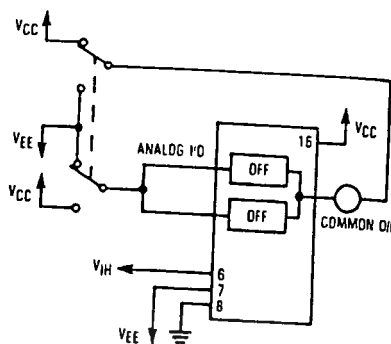


Figure 4. Maximum Off Channel Leakage Current, Common Channel, Test Set-Up

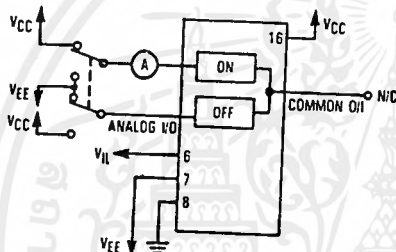
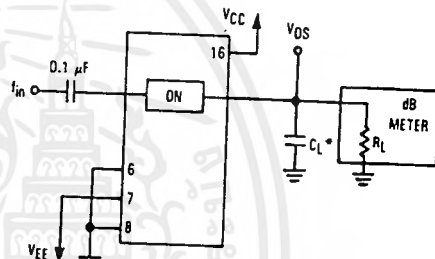
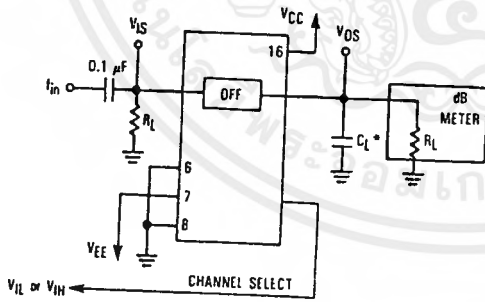


Figure 5. Maximum On Channel Leakage Current, Channel to Channel, Test Set-Up



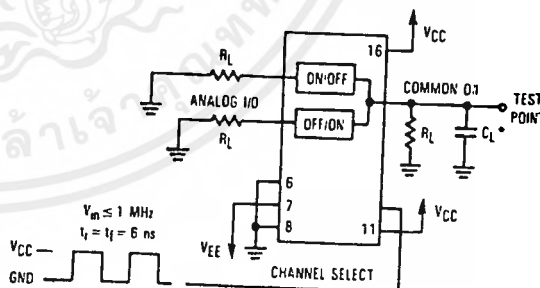
*Includes all probe and jig capacitance.

Figure 6. Maximum On-Channel Bandwidth, Test Set-Up



*Includes all probe and jig capacitance.

Figure 7. Off-Channel Feedthrough Isolation, Test Set-Up



*Includes all probe and jig capacitance.

Figure 8. Feedthrough Noise, Channel Select to Common Out, Test Set-Up

5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC54/74HC4051 • MC54/74HC4052 • MC54/74HC4053

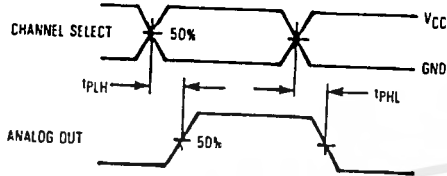
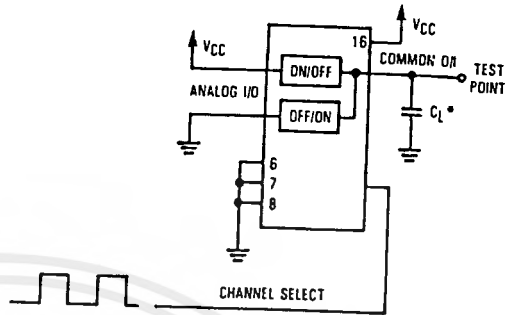


Figure 9a. Propagation Delays, Channel Select to Analog Out



*Includes all probe and jig capacitance.

Figure 9b. Propagation Delay, Test Set-Up Channel Select to Analog Out

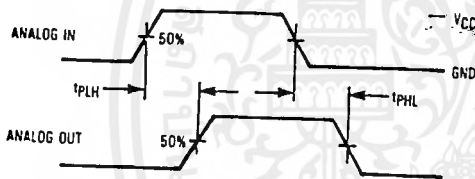
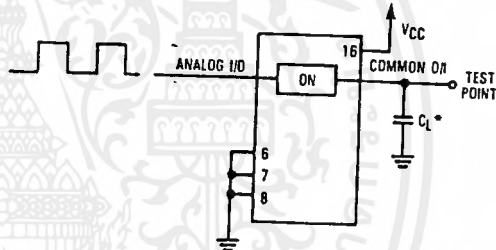


Figure 10a. Propagation Delays, Analog In to Analog Out



*Includes all probe and jig capacitance.

Figure 10b. Propagation Delay, Test Set-Up Analog In to Analog Out

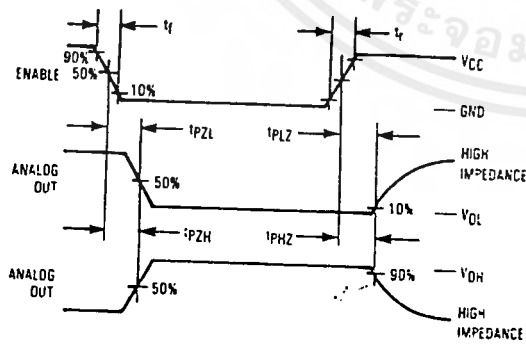


Figure 11a. Propagation Delays, Enable to Analog Out

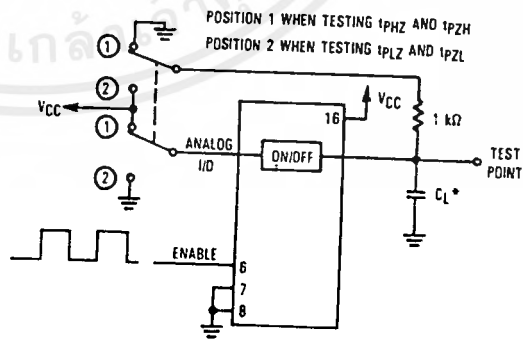
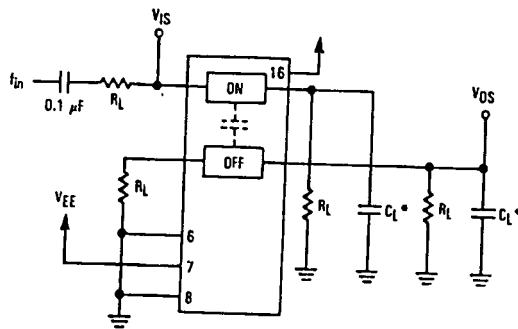


Figure 11b. Propagation Delay, Test Set-Up Enable to Analog Out

5

MC54/74HC4051 • MC54/74HC4052 • MC54/74HC4053

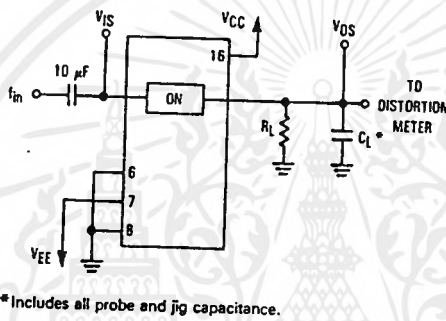


*Includes all probe and jig capacitance.

Figure 12. Crosstalk Between Any Two Switches, Test Set-Up



Figure 13. Power Dissipation Capacitance, Test Set-Up



*Includes all probe and jig capacitance.

Figure 14a. Total Harmonic Distortion, Test Set-Up

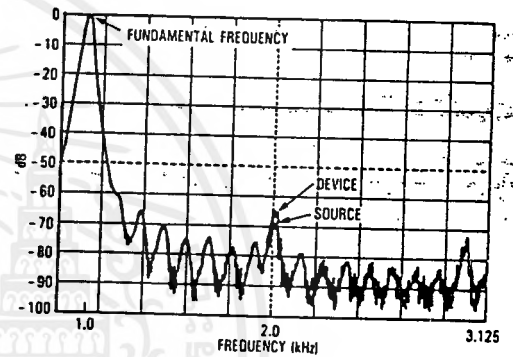


Figure 14b. Plot, Harmonic Distortion

5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM1558/LM1458 Dual Operational Amplifier

General Description

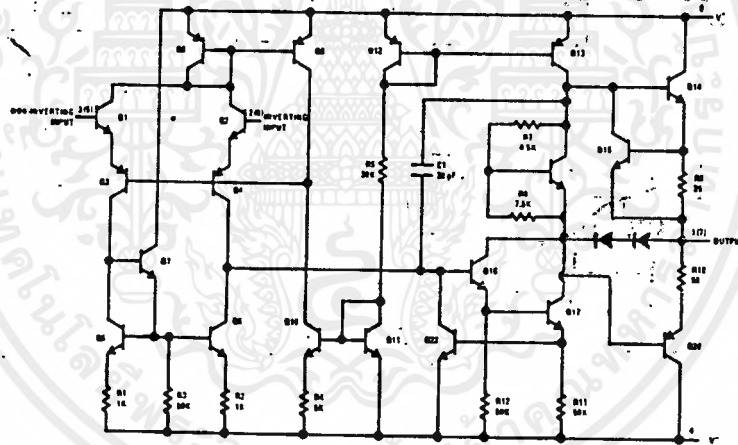
The LM1558 and the LM1458 are general purpose dual operational amplifiers. The two amplifiers share a common bias network and power supply leads. Otherwise, their operation is completely independent. Features include:

- No frequency compensation required
- Short-circuit protection
- Wide common-mode and differential voltage ranges

- Low-power consumption
- 8-lead TO-5 and 8-lead mini DIP
- No latch up when input common mode range is exceeded

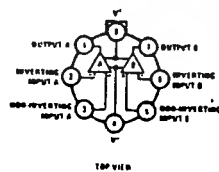
The LM1458 is identical to the LM1558 except that the LM1458 has its specifications guaranteed over the temperature range from 0°C to 70°C instead of -55°C to +125°C.

Schematic and Connection Diagrams



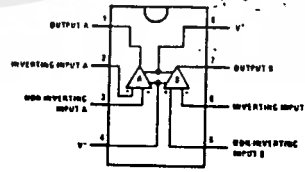
Note: Numbers in parentheses are pin numbers for amplifier B.

Metal Can Package



Order Number LM1558H
or LM1458H
See NS Package H08C

Dual-In-Line Package



Order Number LM1558J
or LM1458J
See NS Package J08A

Order Number LM1458N
See NS Package N08B



LM1558/LM1458

Absolute Maximum Ratings

Supply Voltage LM1558	±22V	Output Short-Circuit Duration	Indefinite
LM1458	±18V	Operating Temperature Range LM1558	-55°C to 125°C
Power Dissipation (Note 1) LM1558H/LM1458H	500 mW	LM1458	0°C to 70°C
LM1458N	400 mW	Storage Temperature Range	-65°C to 150°C
Differential Input Voltage	±30V	Lead Temperature (Soldering, 10 sec)	300°C
Input Voltage (Note 2)	±15V		

Electrical Characteristics (Note 3)

PARAMETER	CONDITIONS	LM1558			LM1458			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Input Offset Voltage	$T_A = 25^\circ\text{C}, R_S \leq 10\text{ k}\Omega$		1.0	5.0		1.0	6.0	mV
Input Offset Current	$T_A = 25^\circ\text{C}$		80	200		80	200	nA
Input Bias Current	$T_A = 25^\circ\text{C}$		200	500		200	500	nA
Input Resistance	$T_A = 25^\circ\text{C}$	0.3	1.0		0.3	1.0		M Ω
Supply Current Both Amplifiers	$T_A = 25^\circ\text{C}, V_S = \pm 15\text{V}$		3.0	5.0		3.0	5.6	mA
Large Signal Voltage Gain	$T_A = 25^\circ\text{C}, V_S = \pm 15\text{V}$ $V_{\text{OUT}} = \pm 10\text{V}, R_L \geq 2\text{ k}\Omega$	50	160		20	160		V/mV
Input Offset Voltage	$R_S \leq 10\text{ k}\Omega$			6.0			7.5	mV
Input Offset Current				500			300	nA
Input Bias Current				1.5			0.8	μA
Large Signal Voltage Gain	$V_S = \pm 15\text{V}, V_{\text{OUT}} = \pm 10\text{V}$ $R_L \geq 2\text{ k}\Omega$	25			15			V/mV
Output Voltage Swing	$V_S = \pm 15\text{V}, R_L = 10\text{ k}\Omega$ $R_L = 2\text{ k}\Omega$	±12 ±10	±14 ±13		±12 ±10	±14 ±13		V V
Input Voltage Range	$V_S = \pm 15\text{V}$	±12			±12			V
Common Mode Rejection Ratio	$R_S \leq 10\text{ k}\Omega$	70	90		70	90		dB
Supply Voltage Rejection Ratio	$R_S \leq 10\text{ k}\Omega$	77	96		77	96		dB

Note 1: The maximum junction temperature of the LM1558 is 150°C, while that of the LM1458 is 100°C. For operating at elevated temperatures, devices in the TO-5 package must be derated based on a thermal resistance of 150°C/W, junction to ambient or 45°C/W, junction to case. For the DIP the device must be derated based on a thermal resistance of 187°C/W, junction to ambient.

Note 2: For supply voltages less than ±15V, the absolute maximum input voltage is equal to the supply voltage.

Note 3: These specifications apply for $V_S = \pm 15\text{V}$ and $-55^\circ\text{C} \leq T_A \leq 125^\circ\text{C}$, unless otherwise specified. With the LM1458, however, all specifications are limited to $0^\circ\text{C} \leq T_A < 70^\circ\text{C}$ and $V_S = \pm 15\text{V}$.