

เครื่องควบคุมการให้สารละลาย
ในเส้นเลือดดำโดยใช้ไมโครคอนโทรลเลอร์
INFUSION PUMP



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญาโท

ปีการศึกษา 2537

ภาควิชา

วิศวกรรมอิเล็กทรอนิกส์

คณะ

วิศวกรรมศาสตร์

เรื่อง

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เครื่องควบคุมการให้สารละลายในเส้นเลือดดำโดยใช้ไมโครคอนโทรลเลอร์

INFUSION PUMP

ผู้จัดทำ

1. นางสาวสุชาดา เตโชติรส รหัสประจำตัวนักศึกษา 34108427

2. นายสุธี เวชพฤษพิทักษ์ รหัสประจำตัวนักศึกษา 34108437

3. นางสาวโสภา โผ้วงพันธ์ รหัสประจำตัวนักศึกษา 34108469



P. Phrasart
.....
(อาจารย์พลศาสตร์ เลิศประเสริฐ)

อาจารย์ที่ปรึกษา

เครื่องควบคุมการให้สารละลาย
ในเส้นเลือดดำโดยใช้ไมโครคอนโทรลเลอร์

น.ส.สุชาดา เตโชติรส

นายสุธี เวชพุกษ์พิทักษ์

น.ส.โสภา โฝพ่วงพันธ์ุ

อาจารย์พลศาสตร์ เลิศประเสริฐ อาจารย์ที่ปรึกษา

บทคัดย่อ

ในปัจจุบันนี้การควบคุมอัตราการไหลของสารละลาย เช่น น้ำเกลือหรือสารละลายอื่นๆเพื่อใช้ในการรักษาพยาบาลผู้ป่วยมีความจำเป็นมาก ดังนั้นเครื่องควบคุมการให้สารละลายในเส้นเลือดดำจึงถูกนำมาใช้กันมากขึ้น ผู้ใช้สามารถตั้งปริมาตรตามที่ต้องการและสามารถตั้งอัตราการไหลให้เหมาะกับผู้ป่วยในแต่ละโรคได้ อุปกรณ์นี้ถูกควบคุมโดยไมโครโปรเซสเซอร์ MCS-51 โดยการเขียนโปรแกรมควบคุมมอเตอร์ในการขับหยดน้ำ โดยมอเตอร์จะเกิดการหมุน ตามโปรแกรมที่ตั้งไว้ และกดสายน้ำเกลือทำให้สารละลายสามารถหยดออกมาได้ ดังนั้นจึงเห็นว่าสารละลายสามารถไหลได้ตามอัตราต่างๆที่ผู้ใช้ตั้งไว้ ถ้าสารละลายในสายน้ำเกลือไม่สามารถไหลได้หรือเกิดการอุดตัน เครื่องจะส่งเสียงเตือนและไฟสีแดงจะติด เมื่อปริมาตรที่ผู้ใช้ตั้งไว้เท่ากับปริมาตรของสารละลายที่เครื่องขับออกมาเครื่องจะส่งเสียงร้องบอกและไฟสีเขียวจะติด ขณะที่เครื่องยังคงทำงานตามปกติไฟสีเขียวจะติดเพื่อแสดงว่า ยังคงมีการไหลของสารละลายอยู่แต่ไฟสีเขียวจะดับเมื่อเกิดการอุดตัน หรือในกรณีที่การทำงานของเครื่องสมบูรณ์ และดับในกรณีที่ผู้ใช้กดปุ่มหยุดการทำงาน ดังนั้นจึงเห็นว่าเครื่องควบคุมการให้สารละลายในเส้นเลือดดำนี้มีต่อผู้ใช้ประโยชน์มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INFUSION PUMP

Miss Suchada Taechotirote

Mr. Sutee Vajchapraukpitak

Miss Sopa Popongpun

Abstract

Nowaday it is necessary to control flow rate of Sodium chloride and another type of solution before injection. Therefore infusion pump have to be used. An user can set Total volume and flow rate for each case of patients in suitable way. This equipment is controlled by Microprocessor MCS-51 by programming for controlling motor driving. MCS-51 will be linked with a motor when MCS-51 receive input from an user, it will order motor to rotate and press tube of solution . So that solution in a tube can flow in each rate that set by an user. If solution can't flow or occlusion occur , infusion pump will alarm with siren sound and RED LED will be lighted. When the delivered volume equal the total volume that set by an user, infusion pump will alarm and YELLOW LED will be lighted . In normal working , GREEN LED will be lighted all time for show that infution pump is flowing . If infution pump work completely or work in error state (occlusion) or stop button is pressed , GREEN LED will deodorize . Therefor this equipment is very useful for each case of patients.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและแนวความคิดพื้นฐานในการออกแบบ	2
2.1 แนวคิดทั่วไปเกี่ยวกับเครื่องควบคุมการให้สารละลาย	2
2.2 การเลือกใช้ Microcontroller	2
2.3 การออกแบบทางด้านกลศาสตร์	4
2.4 การทำงานของวงจร SENSOR	13
บทที่ 3 วงจรเพิ่มเติมเพื่อให้การทำงานของเครื่องมีประสิทธิภาพมากขึ้น	15
3.1 วงจร Buffer	15
3.2 วงจรสร้างความถี่ 100 Hz	16
3.3 วงจรเสียงเตือนเมื่อเกิดการอุดตัน (Occlusion)	18
3.4 ส่วนแสดงผลว่ายังคงมีสารละลายไหลอยู่ (Flowing)	20
3.5 วงจรเสียงเตือนเมื่อการให้ของเหลวเรียบร้อยแล้ว	21
บทที่ 4 การควบคุมการทำงานโดยใช้ Microcontroller	23
4.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51	23
4.2 สถาปัตยกรรมของ 8051	30
4.3 รีจิสเตอร์ของ 8051	36
4.4 กลุ่มคำสั่งใน MCS-51	40
4.5 การเลือกใช้พอร์ทของ 8031	41
4.6 การใช้พอร์ทของ 8255	42
บทที่ 5 โปรแกรมการควบคุมการทำงานโดยใช้ Microcontroller	47
5.1 แนวความคิดในการออกแบบโปรแกรมควบคุมการทำงาน	47
5.2 Flowchart แสดงการทำงานของเครื่องควบคุมการให้สารละลาย	48
5.3 แนวคิดเกี่ยวกับการทำงานของเครื่องที่นำมาใช้พัฒนาโปรแกรม	49
5.3.1 แนวคิดในการออกแบบโปรแกรมหลัก	49
5.3.2 แนวคิดในการออกแบบโปรแกรมการหาร	51
5.3.3 แนวคิดในการออกแบบโปรแกรมการกด Switch เพื่อเพิ่มค่าหรือลดค่าของ ปริมาตรและอัตราการไหล (Volume & Rate)	55
5.4 การเชื่อมต่อ Display LCD เพื่อแสดงผล	58
บทที่ 6 การทดลอง	60
6.1 การทดลองที่ 1 การตรวจสอบความถูกต้องของวงจร SENSOR	60
6.2 การทดลองที่ 2 การตรวจสอบความถูกต้องของเครื่องว่าเป็นไปตามโปรแกรมหรือไม่	61
6.3 การทดลองที่ 3 เพื่อตรวจสอบดูความถูกต้อง ของการทำงานของเครื่อง Infusion Pump	63
ภาคผนวก	65
กิตติกรรมประกาศ	106
บรรณานุกรม	107

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันปัญหาที่พบบ่อยๆในการให้สารละลายแก่ผู้ป่วย เช่น หน้าเกลือ, ยา ก็คือ การทำให้อัตราการไหลของสารละลาย เข้าสู่เส้นเลือดของผู้ป่วยเป็นไปอย่างสม่ำเสมอที่สุด ปกติแล้วสามารถทำได้โดยการปรับแถบบังคับจำนวนหยดที่ติดมากับ *เซทสายหน้าเกลือ* ซึ่งถ้าสายถูกบีบมาก อัตราการไหลของสารละลายก็จะช้า แต่ถ้าสายถูกบีบน้อยอัตราการไหลของสารละลายก็จะเร็วขึ้น สาเหตุที่ทำให้อัตราการไหลของสารละลายไม่คงที่เมื่อเวลาผ่านไปก็คือ การไหลของสารละลายเกิดจากแรงดึงดูดของโลกเพียงอย่างเดียวดังนั้นอัตราการไหลของสารละลายจะขึ้นอยู่กับปริมาณของสารละลายภายในขวดเท่านั้น และอีกสาเหตุหนึ่งก็คือ มันจะขึ้นอยู่กับลักษณะของเส้นเลือดบริเวณที่ทำกรเจาะให้สารละลายของตัวผู้ป่วย ซึ่งอาจพบว่ามีความดันเลือดไม่คงที่หรือเส้นเลือดมีการอุดตัน

ดังนั้นเครื่องควบคุมการให้สารละลายสามารถที่จะลดปัญหาดังกล่าวได้ ทั้งนี้เนื่องจากหยดแต่ละหยดของสารละลายที่ไหลออกมา นั้น เกิดจากการทำงานของระบบบีบ ที่ถูกควบคุมด้วยไมโครคอมพิวเตอร์ดังนั้นค่าแฟคเตอร์ตัวงวที่เป็นสาเหตุที่ทำให้อัตราการไหลของสารละลายไม่คงที่ก็จะหมดไป นอกจากนี้การที่เราสามารถควบคุมจำนวนหยดของสารละลายได้ ทำให้เราสามารถควบคุมปริมาณความเข้มข้นหรือเจือจางของสารละลายที่เราต้องการให้เข้าสู่ตัวผู้ป่วยได้อีกด้วย นอกจากนี้ยังได้ทำการออกแบบเพิ่มเติมในส่วนของการป้องกันความผิดพลาดที่อาจเกิดขึ้นเช่น วงจรตรวจจับฟองอากาศ ซึ่งการให้สารละลายเข้าสู่ตัวผู้ป่วยนั้นจะปล่อยให้ฟองอากาศเข้าสู่ตัวผู้ป่วยไม่ได้ หรือ สัญญาณเตือนภัยที่จะร้องดังขึ้นเมื่อสารละลายภายในขวดหมด เพื่อให้ทำการเปลี่ยนขวดใหม่หรือจัดการอย่างใดอย่างหนึ่งต่อไป

จึงหวังว่าเครื่องควบคุมการให้สารละลายดังกล่าวนี้ สามารถที่จะอำนวยความสะดวกและทำให้เกิดประโยชน์แก่ผู้นำไปใช้ได้บ้างไม่มากนัก

จุดประสงค์

เพื่อให้สามารถควบคุมการให้สารละลายแก่ผู้ป่วยได้อย่างมีประสิทธิภาพ มีความแน่นอนเกี่ยวกับการตั้งจำนวนหยดต่อเวลา และปริมาตรที่ต้องการควบคุม สามารถอำนวยความสะดวกให้แก่ผู้ใช้ได้อย่างเหมาะสม

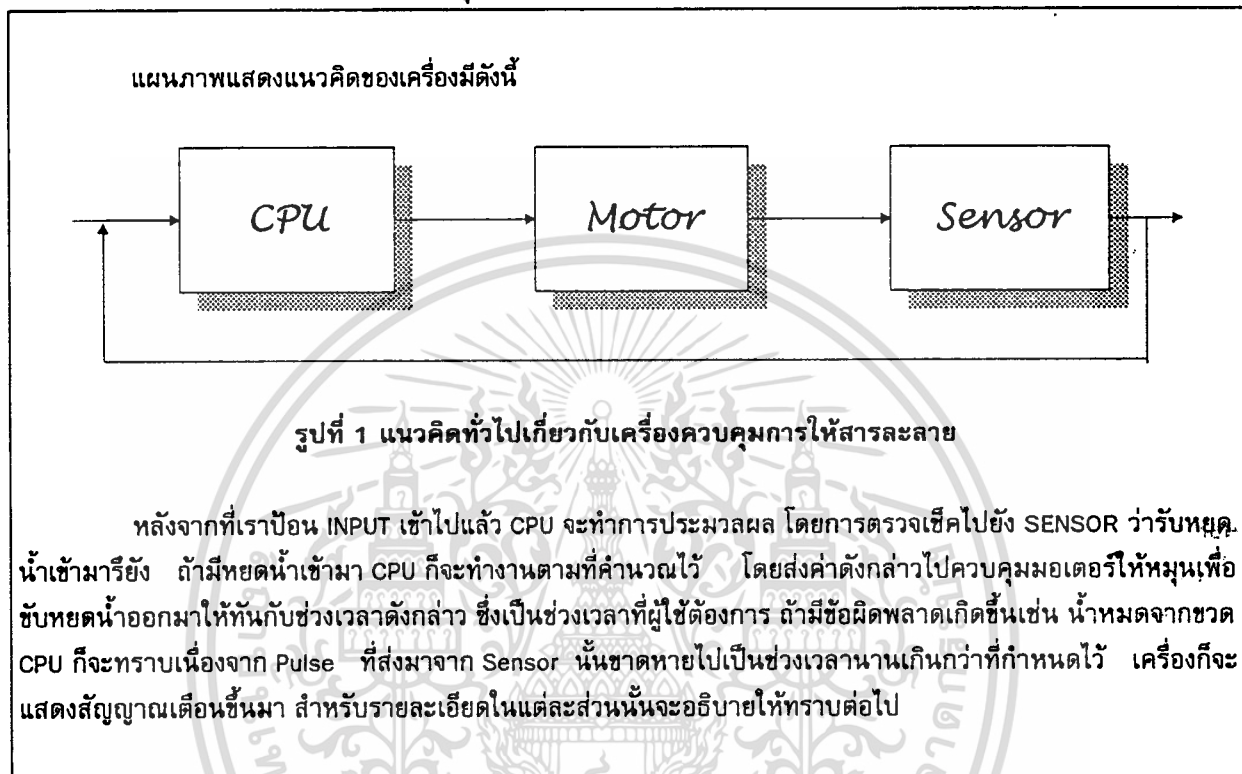
คุณสมบัติ

- สามารถตั้งจำนวนหยดต่อเวลาได้ตามต้องการ โดยผู้ใช้สามารถเลือกจำนวนหยดได้ตามความเหมาะสม
- สามารถควบคุมปริมาตรของสารละลายที่ต้องการจะใช้ได้ โดยแสดงผลให้เห็นทาง Display
- มีระบบเตือนภัยแสดงให้ผู้ใช้ได้ทราบเมื่อเครื่องทำงานผิดพลาด เพื่อให้จัดการอย่างใดอย่างหนึ่งต่อไป

บทที่ 2

ทฤษฎีและแนวความคิดพื้นฐานในการออกแบบ

2.1 แนวคิดทั่วไปเกี่ยวกับเครื่องควบคุมการให้สารละลาย



2.2 การเลือกใช้ Microcontroller

สำหรับเครื่องควบคุมการให้สารละลายเราได้เลือกใช้ *Microcontroller* ตระกูล MCS-51 ซึ่งเป็นชิปที่เหมาะสมกับระบบขนาดกลางที่มีความซับซ้อนและต้องการความเร็วในการทำงานสูงพอสมควร ได้รับความนิยมแพร่หลายมากที่สุด สามารถควบคุมการทำงานได้อย่างมีประสิทธิภาพ ชิปไมโครคอนโทรลเลอร์ทุกตัวในตระกูลนี้จะมีการทำงานพื้นฐานที่เหมือนกันทุกประการ มีขนาดของชิปเท่ากันเพื่อให้สามารถแทนใช้กันได้ ส่วนรายละเอียดที่แตกต่างกันของชิปแต่ละเบอร์ในตระกูลนี้ก็ได้ออกมาจำนวนคำสั่งที่เพิ่มมากขึ้นเพื่ออำนวยความสะดวกในการเขียนโปรแกรม ความเร็วในการประมวลผลมากขึ้น ใช้พลังงานน้อยลงกว่าเดิม มีวงจรภายในชิปเพิ่มมากขึ้น เช่น มีวงจรมีเมมโมรี่หรือแคชเพิ่ม มีวงจรรีบและส่งข้อมูลแบบอนุกรมเพิ่มขึ้น สามารถรับและจัดการสัญญาณอินพุตภายนอกได้มากขึ้น มีหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายในชิปเพิ่มขึ้น ทั้งนี้เพื่อให้เหมาะสมกับงานหรือระบบที่ต้องการนำไปใช้ควบคุมให้มากที่สุด สำหรับเครื่องควบคุมการให้สารละลายนี้ได้เลือกใช้ชิปไมโครคอนโทรลเลอร์เบอร์ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ MCS-51

คุณสมบัติที่สำคัญของชิปไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังนี้

- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ในชิป จำนวน 4 กิโลไบต์ (เบอร์ 8031, 8032 ไม่มีหน่วยความจำส่วนนี้)
- มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป (RAM) อยู่ในชิปจำนวน 128 ไบต์ (ใน 8031, 8051) หรือ 256 ไบต์ (ในเบอร์ 8032, 8052)
- สามารถใช้หน่วยความจำสำหรับโปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้อย่างละ 64 กิโลไบต์แยกจากกัน
- คำสั่งส่วนใหญ่ใช้เวลาเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิร์ตซ์
- มีพอร์ตที่สามารถรับหรือส่งข้อมูลได้ทั้งสองทิศทาง จำนวน 4 พอร์ตละ 8 บิต หรือสามารถใช้งานเป็นพอร์ตขนาด 1 บิตแยกจากกัน ทำให้เสมือนมีพอร์ตขนาด 1 บิตใช้งานรวมทั้งสิ้น 32 พอร์ต
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราการเร็วในการรับและส่งข้อมูล (baud rate) ได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที
- จัดลำดับความสำคัญของสัญญาณอินเทอร์รัปต์ได้ 2 ระดับ
- มีรีจิสเตอร์สำหรับใช้งานเป็นไทม์เมอร์หรือเคาน์เตอร์เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิปหรือนับการเปลี่ยนแปลงของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา (ในเบอร์ 8052 จะมี 3 ตัว)
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิตเพื่อให้ออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น
- มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
- สามารถประมวลผลแบบบิตเพื่อใช้ในงานควบคุมโดยเฉพาะ
- ใช้โปรแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-48 (upwardly compatible) ได้
- เบอร์ 8031 เบอร์นี้ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป แต่สามารถใช้หน่วยความจำเพื่อเก็บโปรแกรมที่อยู่ภายนอกได้มากถึง 64 กิโลไบต์ ซึ่งอาจใช้เป็น ROM, PROM, EPROM ตามความต้องการของผู้ผลิต เบอร์ 8031 นี้มีไว้ใช้ในกรณีที่โปรแกรมมีขนาดเล็กกว่า 4 กิโลไบต์ หรือมากกว่า 4 กิโลไบต์

สำหรับรูปแสดงตำแหน่งของขาต่างๆใน MCS-51 จะแสดงไว้ที่ภาคผนวก

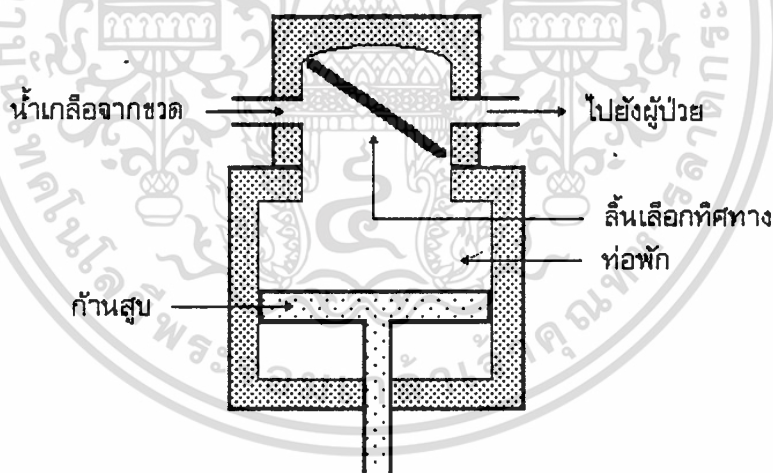
2.3 การออกแบบทางด้านกลศาสตร์

แบบของกลไกชุดรีดน้ำเกลือ

เครื่องควบคุมการให้สารละลายหรือ Infusion Pump ที่มีใช้ตามโรงพยาบาลต่างๆ ในปัจจุบันนั้น มีรูปแบบต่างๆ กัน ซึ่งรูปแบบต่างๆ กันนี้ส่วนมากแล้วจะต่างกันที่ การออกแบบทางกลศาสตร์ในการรีดสารละลายให้ออกมา ซึ่งในส่วนนี้จะมีลักษณะการทำงานที่แตกต่างกัน แต่ถึงอย่างไรหลักการทำงานของเครื่องนี้ก็คือ การใช้ส่วนของจอร์อิเล็กทรอนิกส์ในการควบคุมการทำงาน ให้กลไกในการรีดสารละลายเป็นไปตามที่เราต้องการ ดังนั้นสิ่งแรกที่เราจะคำนึงถึงในการออกแบบก็คือ เราต้องการจะให้มีกลไกในการรีดสารละลายอย่างไร ซึ่งจากการศึกษากลไกของเครื่องควบคุมการให้สารละลายที่มีใช้อยู่ในปัจจุบันและจากวารสารทางวิชาการแพทย์พบว่า มีด้วยกัน 4 แบบ แต่สามารถแบ่งเป็นลักษณะในการทำได้เป็น 2 กลุ่มใหญ่ๆ

1. กลไกในการให้สารละลายที่มีลักษณะเหมือนการบีบ

กลไกในการให้สารละลายในลักษณะนี้ จะมีลักษณะเหมือนกับบีบน้ำซึ่งจะสูบน้ำจากแหล่งจ่ายน้ำไปเก็บไว้ในถังพักน้ำ และจะจ่ายน้ำออกไปให้กับผู้ใช้เมื่อถึงคราวที่ต้องการ รูปแสดงกลไกอย่างคร่าวๆ แสดงได้ดังรูปที่ 2

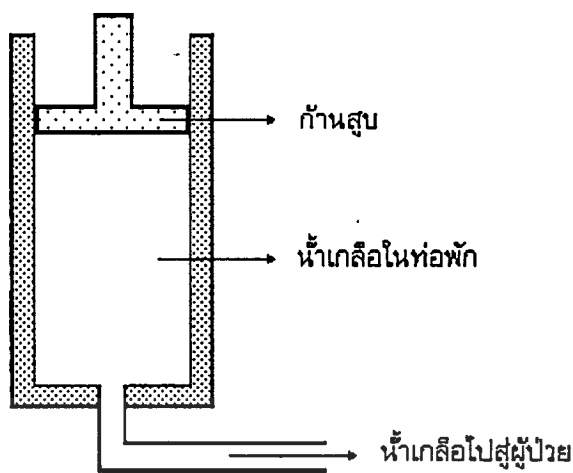


รูปที่ 2 กลไกในการรีดสารละลายที่มีการทำงานคล้ายบีบน้ำ

เมื่อเริ่มต้นการทำงานสารละลายจะถูกสูบเข้ามาเก็บไว้ในท่อพักจนเต็มโดยใช้เวลาในช่วงนี้ไม่นานนัก จากนั้นจอร์ทางอิเล็กทรอนิกส์ก็จะควบคุมการทำงานของก้านสูบโดยให้ก้านสูบค่อยๆ ดันสารละลายออกจากท่อพักเข้าสู่ตัวผู้ป่วย ซึ่งความเร็วในการดันของก้านสูบจะแปรผันตามอัตราการให้สารละลายที่แพทย์หรือพยาบาลตั้งไว้ในตอนต้น เมื่อสารละลายหมดจากท่อพักแล้วก็จะมีการสูบเข้ามาให้เต็มท่อพักใหม่อีกและการทำงานก็จะเป็นอย่างนี้เรื่อยไป

อีกรูปแบบหนึ่งของกลไกที่มีลักษณะแบบการบีบเช่นกัน แสดงได้ดังรูปที่ 3 ซึ่งแบบนี้สารละลายทั้งหมดจะถูกบรรจุไว้ในท่อพักก่อน ซึ่งท่อพักนี้มีขนาดใหญ่พอที่จะบรรจุปริมาตรสารละลายที่ให้อุปกรณ์ เนื่องจากวิธีการนี้จะไม่มีการสูบน้ำสารละลายเข้ามายังท่อพักเมื่อสารละลายหมดจากท่อพักแล้ว

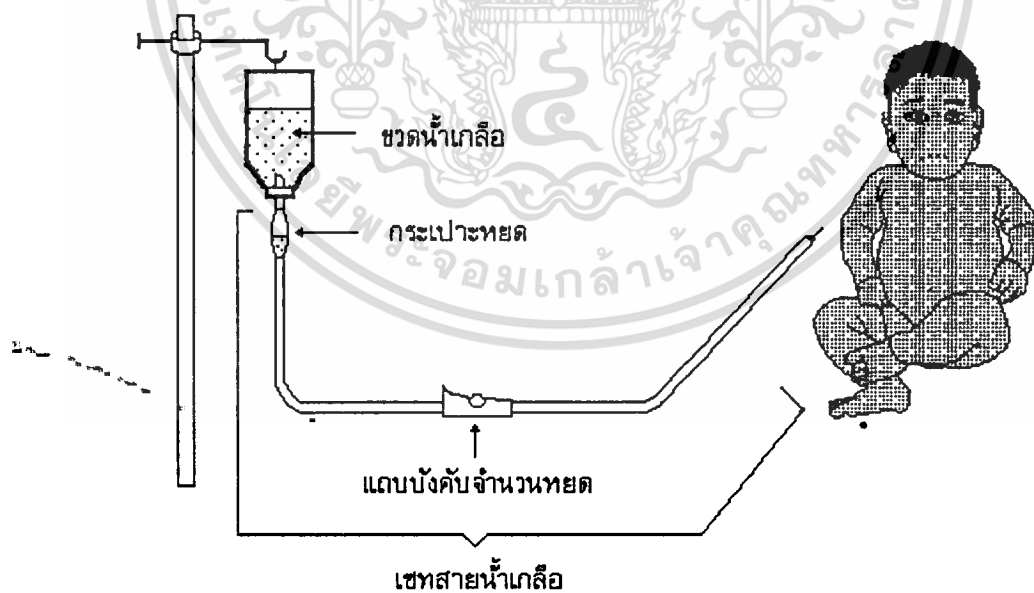
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 กลไกการให้สารละลายแบบบีบอีกรูปแบบหนึ่ง

2. กลไกในการให้สารละลายที่มีลักษณะแบบการรีด

กลไกลักษณะนี้จะมีใช้ตามโรงพยาบาลมากกว่าแบบที่แล้ว ทั้งนี้อาจจะเป็นเพราะสามารถใช้งานได้สะดวก และคล่องตัวกว่า ซึ่งถ้าเราเคยเห็นผู้ป่วยที่ต้องให้น้ำเกลือจะเห็นว่า ชุดการให้น้ำเกลือจะประกอบไปด้วยขวดน้ำเกลือ และเซทสายน้ำเกลือ ดังแสดงในรูปที่ 4

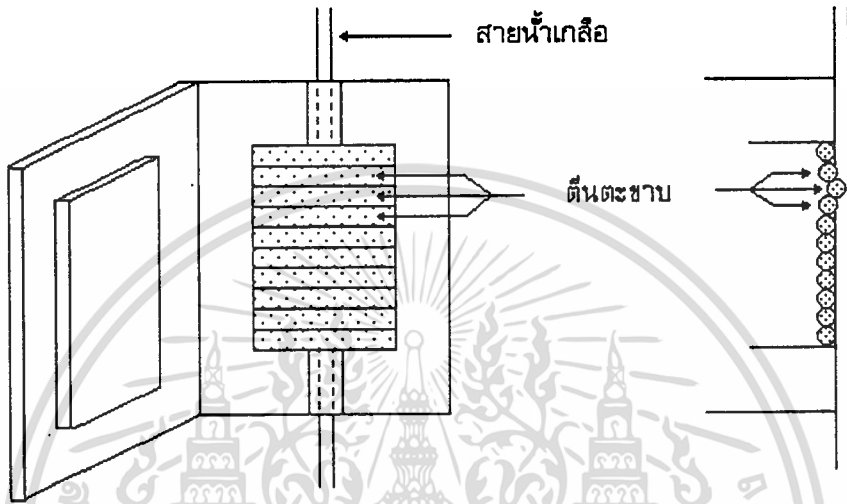


รูปที่ 4 แสดงชุดการให้น้ำเกลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.กลไกการรีดแบบดินตะขาบ

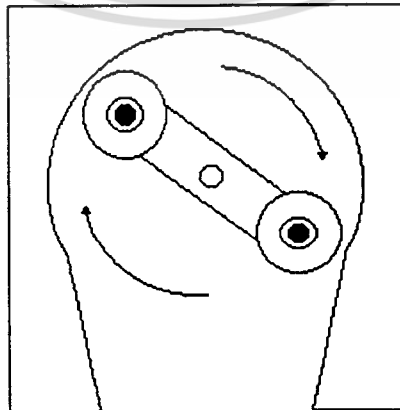
เครื่องควบคุมการให้สารละลาย ที่มีชุดการรีดแบบดินตะขาบนี้เป็นที่นิยมใช้กันมากที่สุด เพราะเนื่องจากสามารถนำไปใช้งานได้สะดวกที่สุด กลไกแบบดินตะขาบแสดงให้เห็นด้วยคร่าวๆ ดังรูปที่ 5 การทำงานของกลไกแบบนี้ก็คือ ดินตะขาบแต่ละอันที่วางเรียงเป็นหน้ากระดานนั้นจะเคลื่อนตัวออกไปกดทับสายน้ำเกลือในลักษณะที่ผลัดกันกดแบบต่อเนื่อง ซึ่งมีลักษณะเหมือนกับล้อที่กดทับสายน้ำเกลือ แล้วค่อยๆ เคลื่อนลงมาทำให้เกิดการรีดสารละลายให้ออกมาจากสายที่ระเหยตามที่เราต้องการ



รูปที่ 5 แสดงกลไกการรีดลักษณะแบบดินตะขาบ

4.กลไกการรีดแบบวงกลม

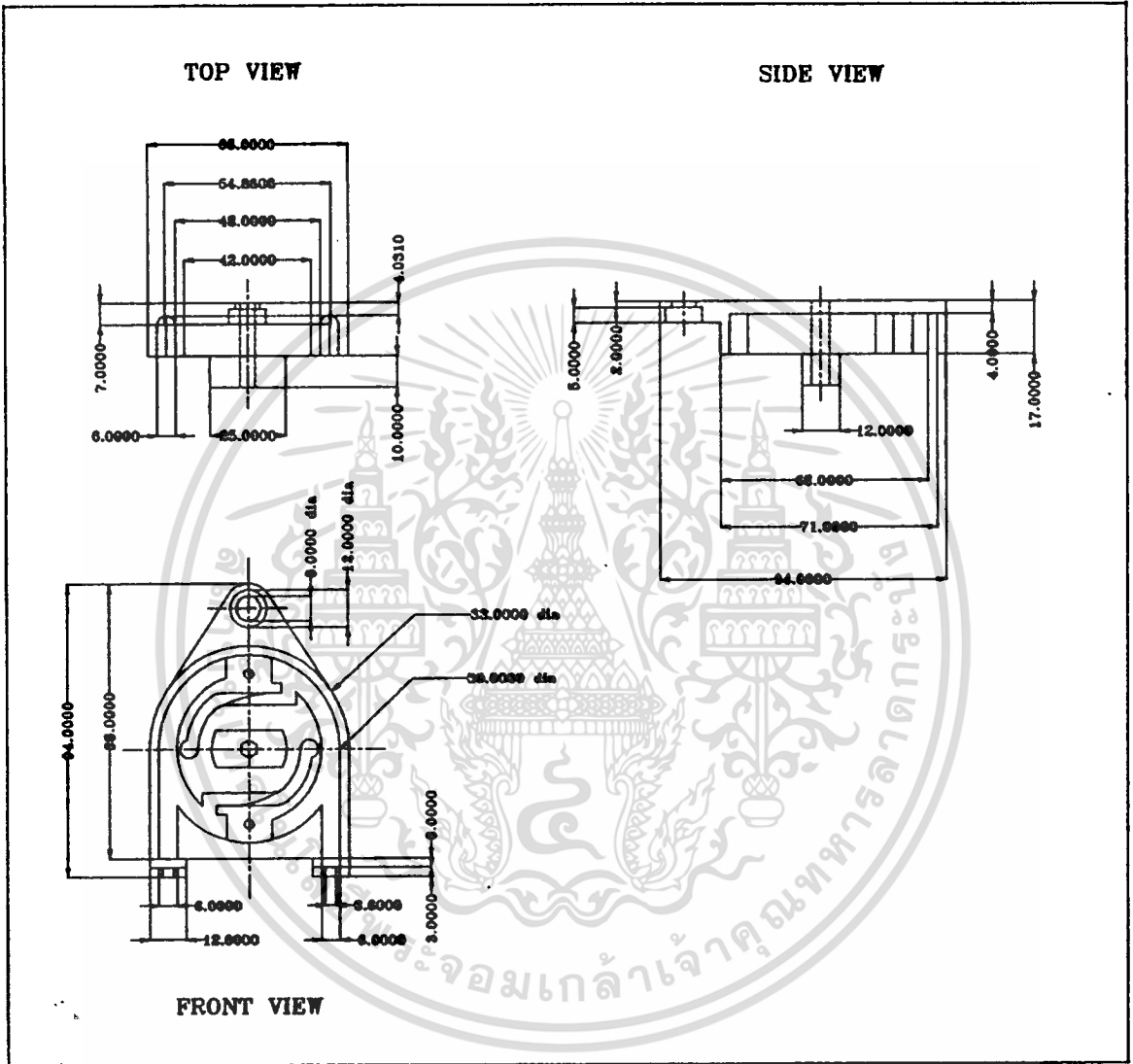
รูปที่ 6 แสดงให้เห็นถึงชุดรีดสารละลายที่มีโครงสร้างเป็นแบบวงกลม การทำงานของชุดการรีดแบบนี้ก็คือ ล้อใดล้อหนึ่งจะกดทับสายน้ำเกลือตลอดเวลา เพื่อบังคับให้สารละลายส่งผ่านไปยังผู้ป่วยเมื่อแกนรีดนี้หมุนไปเท่านั้น



รูปที่ 6 แสดงชุดการรีดสารละลายแบบวงกลม

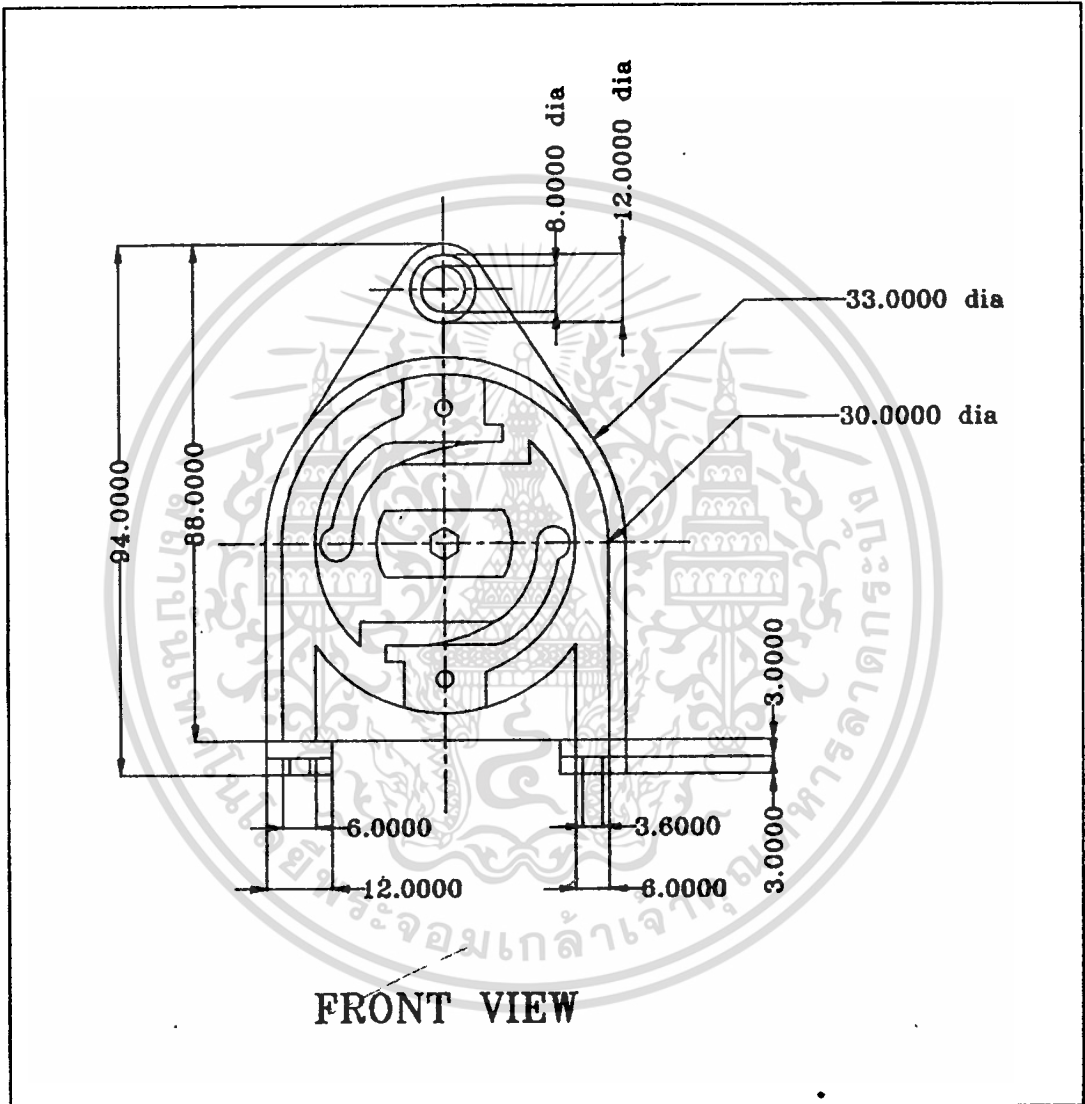
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7 กลไกการรีดที่เลือกใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8 กลไกการรีดที่เลือกใช้ (ต่อ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เราอาจจะมองเห็นได้ว่า ชุดการรีดสารละลายแบบวงกลมกับชุดการรีดสารละลายแบบตีนตะขาบมีลักษณะที่คล้ายคลึงกัน เพราะถ้าเราลองยึดชุดการรีดสารละลายแบบวงกลมออกให้อยู่ในลักษณะแนวเส้นตรง ก็จะได้การทำงานที่เหมือนกับชุดการรีดสารละลายแบบตีนตะขาบ

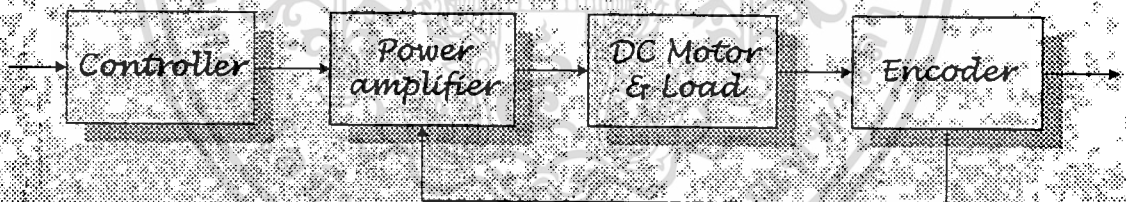
ในการที่จะเลือกกลไกการรีดสารละลายที่กล่าวไปแล้วทั้งหมดมาอย่างใดอย่างหนึ่ง เพื่อมาทำการวิจัยและพัฒนาต่อไปนั้น เราควรคำนึงถึงความเป็นไปได้ในความสามารถที่จะสร้างชุดการรีดสารละลายขึ้นมา ถ้าเราจะเลือกใช้กลไกการรีดแบบบีบ แบบนี้ผู้ใช้อาจจะไม่นิยมใช้กันมากเท่าไรนัก ถ้าจะใช้กลไกการรีดแบบตีนตะขาบ ถึงแม้ว่าจะเหมาะสมในการใช้งานมากที่สุดแต่ก็พบว่ากลไกนี้มีความซับซ้อนพอสมควรจึงได้ตัดสินใจเลือกใช้ชุดการรีดแบบวงกลมซึ่งมีการออกแบบง่ายกว่าแต่สามารถรีดสารละลายได้มีประสิทธิภาพสูงทีเดียว ซึ่งจะได้กล่าวถึงการออกแบบต่อไป

เมื่อได้แนวความคิดเกี่ยวกับชุดรีดน้ำเกลือหรือสารละลายเรียบร้อยแล้ว ส่วนที่จะต้องคำนึงถึงต่อไปคือส่วนที่เกี่ยวข้องกับวงจรอิเล็กทรอนิกส์ เพื่อใช้ในการควบคุมมอเตอร์เพื่อควบคุมการทำงานของชุดรีดน้ำเกลือให้เป็นไปตามที่เราต้องการ

ระบบการควบคุมความเร็วของ DC Motor

ส่วนประกอบพื้นฐานของระบบการควบคุมดีซีมอเตอร์แสดงได้ดังบล็อกไดอะแกรมซึ่งประกอบด้วย

1. ตัวคอนโทรลเลอร์ (Controller)
2. วงจรไดรเวอร์ (Driver) หรือเพาเวอร์แอมพลิไฟร์ (Power amplifier)
3. ฟีดแบ็คทรานสดิวเซอร์ (Feedback transducer) หรือเอนโคเดอร์ (Encoder)
4. ดีซีมอเตอร์ (D.C. Motor) และโหลด (Load)



รูปที่ 9 ระบบการควบคุมความเร็วของ DC Motor

รายละเอียดและหน้าที่ของส่วนประกอบแต่ละส่วนอธิบายได้ดังนี้

1. ตัวคอนโทรลเลอร์

เป็นส่วนของระบบที่ทำให้เกิดสัญญาณควบคุมไปยังการทำงานของดีซีมอเตอร์และโหลด และยังสามารถแบ่งออกได้อีกเป็น 2 ประเภทตามลักษณะของสัญญาณที่ใช้ควบคุม คือ

1. อนาลอกคอนโทรลเลอร์ เป็นสัญญาณควบคุมในลักษณะของอนาลอก
2. ดิจิตอลคอนโทรลเลอร์ เป็นสัญญาณควบคุมในลักษณะของดิจิตอลหรือไบนารีโค้ด (Binary code)

2. วงจรไดรเวอร์

ทำหน้าที่ขยายสัญญาณควบคุมให้มีขนาดเหมาะสม ก่อนที่จะป้อนเข้าไปขับดีซีมอเตอร์และโหลด วงจรไดรเวอร์ยังแบ่งออกเป็น 2 แบบใหญ่ๆ คือ

1. แบบลิเนียร์แอมพลิไฟเออร์ (Linear Amplifier)
2. พัลส์วิตมอดูเลชันแอมพลิไฟเออร์ (Pulse-width modulation Amplifier)

3. ฟีดแบ็คทรานสดิวเซอร์หรือเอนโคเดอร์

เป็นอุปกรณ์ที่ใช้ในการรับรู้หรือดีเทค (Detect) สัญญาณเอาต์พุตที่ต้องการ โดยไม่มีผลของการโหลดตั้ง (Loading) สัญญาณที่ ดีเทคได้นี้จะป้อนกลับไปเปรียบเทียบกับสัญญาณอ้างอิง ทำให้ได้สัญญาณที่คลาดเคลื่อนกัน (Error) ฟีดแบ็คทรานสดิวเซอร์ แบ่งออกได้เป็น 2 แบบใหญ่ๆ คือ

1. อนุาลอกทรานสดิวเซอร์ ได้แก่พวก ทาโตเซนเนอร์เรเตอร์ โปเทนทีโอมิเตอร์ และ ซิงโคร เป็นต้น
2. ดิจิตอลทรานสดิวเซอร์ ได้แก่พวก อินครีเมนท์เอนโคเดอร์รีโซลเวอร์ แมคนติกฟลักซ์ เป็นต้น

4. ดีซีมอเตอร์และโหลด

เป็นระบบที่ถูกควบคุมหรือส่วนที่ออกแรงทำงานซึ่งจะเป็นเครื่องจักรกล (ดีซีมอเตอร์) หรืออะไรก็ตามที่ให้

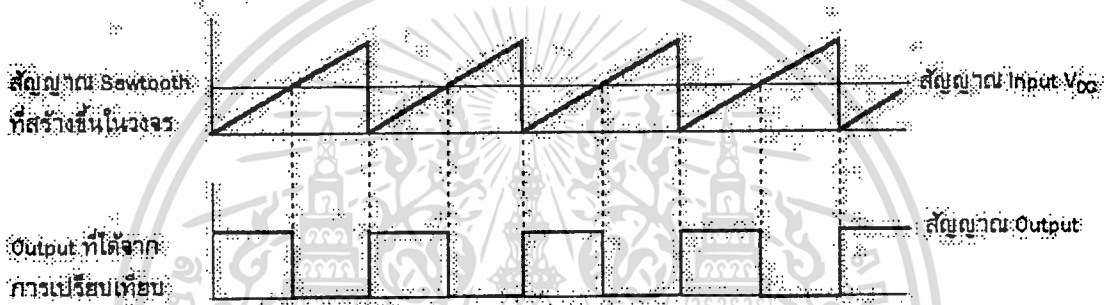
ตัวแปร



สำหรับเครื่องควบคุมการให้สารละลายนี้ได้เลือกใช้การควบคุมดีซีมอเตอร์โดยใช้ Pulse width Modulation เป็นหลัก จึงขออธิบายเฉพาะในส่วนของ Pulse width Modulation เท่านั้น

Pulse Width Modulation

หลักการทํางานของส่วนนี้เป็นการ Compare กันระหว่างสัญญาณไฟ DC ที่เป็น Input เข้าไป และสัญญาณ (Saw tooth) ที่สร้างขึ้นในวงจร ซึ่งความถี่ของ Saw tooth สามารถปรับได้โดยการปรับค่า R vary แล้ว สัญญาณทั้งสองนี้ หัวไปเข้า Op-Amp ซึ่งใช้เป็น Comparator Op-Amp ทำให้ได้ Output ออกมา อธิบายได้ดังรูป



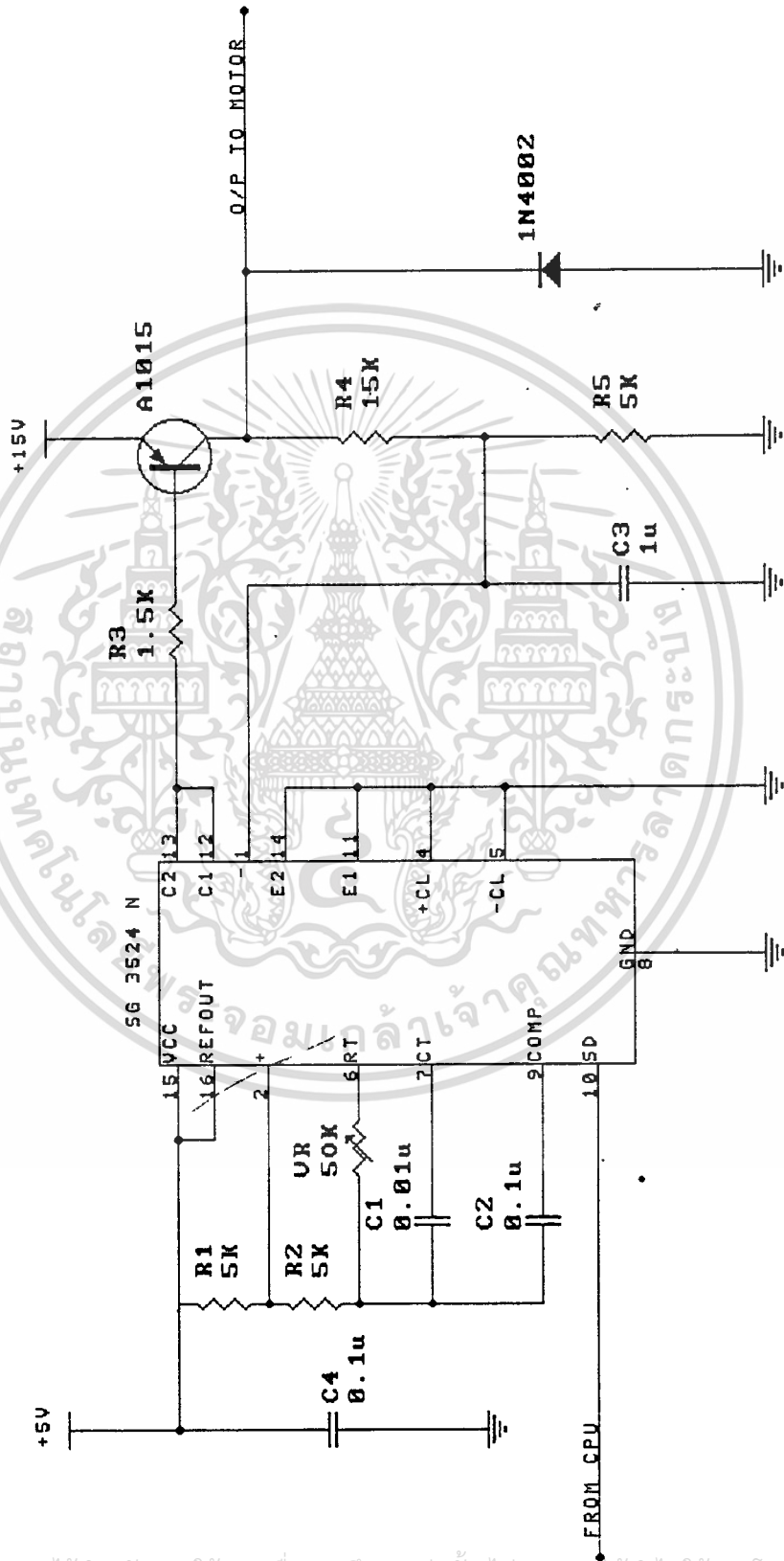
รูปที่ 10 แสดงการเปรียบเทียบสัญญาณเมื่อใช้ Pulse Width Modulation

จากรูปจะเห็นได้ว่า เราสามารถเปลี่ยนความถี่หรือ Duty cycle ได้โดยการปรับความถี่ของ Saw tooth (ปรับค่า V_{R1}) สามารถปรับความกว้างของ Pulse โดยที่ความถี่เท่าเดิม โดยการปรับที่ V_{DC} Input

จากหลักการดังกล่าว เราจึงนำเทคนิคนี้มาใช้ในส่วนขับ DC Motor ช่วยในการลดกระแสที่ผ่าน Transistor Switching และยังทำให้ไม่ต้องใช้ Supply ขนาดใหญ่มาก ประโยชน์ที่ได้ก็คือ เราสามารถปรับ ความเร็วของ Motor ได้โดยที่แรงบิดในการหมุนของ Motor มีค่าคงที่และ Supply ที่ให้ Motor มีค่าคงที่ด้วยนั่นก็คือการปรับ Duty cycle ของ Pulse นั้นเอง

ถ้าความกว้างของ Duty cycle ของ Pulse มวลก็จะทำให้ Motor มีความเร็วสูง ถ้าความกว้างของ Duty cycle ของ Pulse น้อยก็จะทำให้ Motor มีความเร็วต่ำ

รูปที่ 11 รูปแสดงวงจร Pulse Width Modulation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การทำงานของวงจร SENSOR

จากรูปแสดงการทำงานของวงจรส่วน SENSOR สามารถอธิบายได้ดังนี้ $C=22\mu F$ ที่มีในวงจรจะช่วยกันสัญญาณรบกวน (noise) ทำให้สัญญาณดีขึ้น ส่วนตัวต้านทาน (R) ที่มีอยู่ในวงจรจะช่วยแบ่งกระแส Divider สำหรับ $\mu A 741$ ใช้เป็นตัว Comparator โดยเลือกให้ขา 3 (+) , ขา 2 (-) Compare กันโดยให้ขา 3 มี Voltage คงที่ซึ่งวัดได้ประมาณ 3 V สำหรับขา 2 จะเห็นว่าต่อกับ Diode ที่เป็นตัวรับหยดน้ำ ดังนั้นขณะที่มีหยดน้ำผ่าน ก็จะเสมือนมีอะไรมากระทบสัญญาณระหว่างตัวรับและตัวส่ง ขณะที่หยดน้ำผ่านตัวรับและตัวส่งนั้น จะทำให้กระแสไม่สามารถไหลผ่าน Diode ตัวรับได้ เพราะมีความต้านทานสูงมาก ดังนั้น V ที่ขา 2 จึงสูงกว่า ขา 3 มากเนื่องจาก ขา 2 เป็นลบ OUTPUT ที่ออกมาจึงเป็นลบ

ส่วนขณะที่ไม่มีหยดน้ำผ่าน Diode ตัวรับ จะรับสัญญาณจากภาคส่งได้ตลอด มีกระแสไหลผ่าน Diode ซึ่งเสมือนมีความต้านทานน้อย จึงทำให้ Voltage ตกคร่อม ขา 2 น้อยประมาณ 1.5V ทำให้ V ที่ ขา 2 ต่ำกว่าขา 3 เนื่องจาก ขา 3 เป็นบวก OUTPUT ที่ออกมาจึงเป็นบวก ดังนั้นขณะที่มีหยดน้ำผ่านจะได้สัญญาณเป็นลบ ดังรูป



ช่วงที่มีหยดน้ำผ่าน

รูปที่ 12 แสดงลักษณะสัญญาณขณะที่มีหยดน้ำผ่านจะได้สัญญาณเป็นลบ

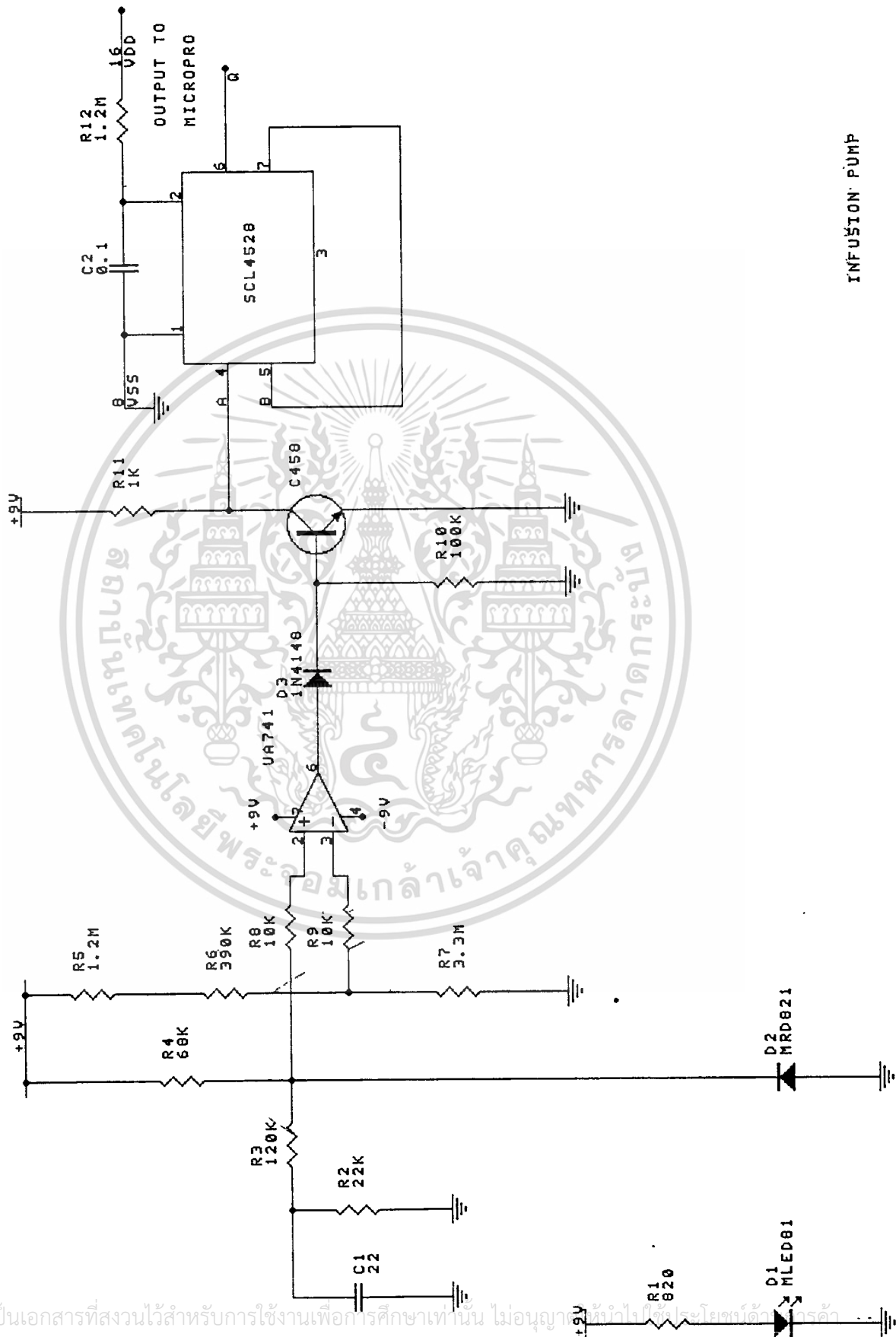
OUTPUT ที่ออกมาจากขา 6 จะผ่าน Signal Diode เพื่อให้ Switch สัญญาณได้เร็วขึ้นและกันไฟลช ไม่ให้ผ่านผ่านได้เฉพาะกราวด์ (GND) กับไฟบวก เท่านั้น จึงทำให้สัญญาณเราดีขึ้นแต่สัญญาณที่เราต้องการ คือต้องการให้มี Pulse เป็นบวกเมื่อมีหยดน้ำผ่าน ดังนั้นจึงนำสัญญาณที่ได้มากลับเฟสโดยเลือกใช้ Transistor เบอร์ C 458 มาช่วยกลับเฟสสัญญาณ โดยถ้ามีสัญญาณที่เป็น Volt บวก ออกมาจาก OUTPUT ขา 6 ของ $\mu A 741$ จะทำการไบอัส Transistor ทำให้ Transistor ทำงานโดยมีกระแสไหลจาก Collector ไปยัง Emitter ทำให้ Volt ที่จุด B เป็นกราวด์ (GND) เพราะเปรียบเสมือนการ Short Circuit แต่พอมี OUTPUT ซึ่งเป็นลบที่ขา 6 ทำให้ไม่สามารถไบอัส Transistor ได้ Transistor จึงไม่นำกระแสเปรียบเสมือนเป็น High Impedance Volt ที่จุด B จึงสูงทำให้ OUTPUT ที่จุด B เป็นบวก ทำให้สามารถกลับเฟสสัญญาณได้ หลังจากนั้นสัญญาณที่ได้จะผ่านเข้า IC SCL 4528 เพื่อสร้าง Pulse ในช่วงเวลาที่ต้องการ จาก $T = RC$ โดย SCL 4528 นี้จะเป็น IC ที่เราเลือกมาใช้งานเพื่อให้ทำงานเป็นแบบ Non-retigable คือจะ Trig สัญญาณที่ Pulse ขาขึ้น ซึ่งถ้ามี Pulse เล็กๆ ซึ่งเกิดจาก Noise มันก็จะไม่ Trig อีก ดังนั้นเราก็จะได้ Pulse บวกเป็นจำนวนหยดน้ำที่ Sensor จับได้

ช่วงที่มีหยดน้ำผ่าน



รูปที่ 13 แสดงลักษณะสัญญาณขณะที่มีหยดน้ำผ่านจะได้สัญญาณเป็นบวก

รูปที่ 14 รูปแสดงการทำงานของวงจร SENSOR



INFUSION PUMP

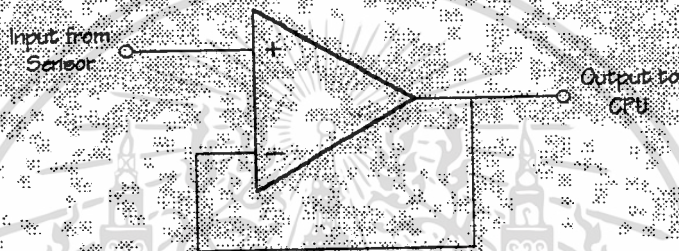
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ได้
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วงจรเพิ่มเติมเพื่อให้การทำงานของเครื่องมีประสิทธิภาพมากขึ้น

3.1 วงจร Buffer

เนื่องจากสัญญาณที่มาจากวงจร SENSOR มีกำลังอ่อนมากจึงต้องนำมานำวงจร Buffer เพื่อช่วยขยายกระแสก่อนที่จะนำสัญญาณนี้ไปเป็นอินพุตเข้าสู่พอร์ตของ CPU ถ้าเราไม่ใช้ Buffer สัญญาณจะ Drop ท้ายไป Voltage จะตกลงจนเกือบเป็น 0 โวลท์ CPU จึงไม่สามารถรับสัญญาณจาก SENSOR เข้าไปประมวลผลได้ วงจร Buffer สามารถสร้างได้ง่ายๆ โดยใช้ Op-Amp เบอร์ 741 ต่อเป็นวงจร Buffer ได้ดังรูป

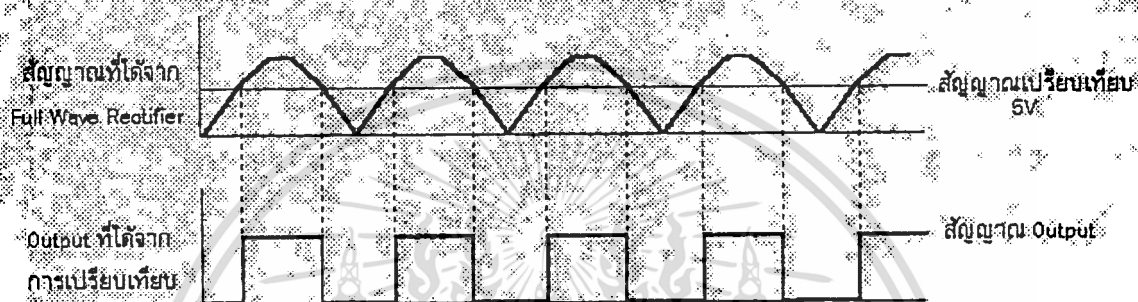


รูปที่ 1.5 รูปแสดงวงจร Buffer

ครั้งแรกที่เรายังไม่ได้อิงวงจร Buffer ลงไป เมื่อทำการทดสอบดูก็ปรากฏว่า CPU ไม่สามารถรับสัญญาณที่เราต้องการส่งไปจาก SENSOR ได้เลย ทั้งๆที่ได้ลองตรงจับค่า Voltage ของหยตหน้าเมื่อผ่าน SENSOR ดูแล้วก็เห็นว่า มีระดับ Voltage ที่สูงมากกว่า 5 โวลท์ ซึ่งเป็นระดับที่ CPU น่าจะสามารถเข้าใจได้แต่เมื่อลองไปวัด Voltage ที่ขาของ CPU ดู ปรากฏว่า Voltage ที่เข้ามา Drop ท้ายไปเกือบหมด จึงได้ลองต่อวงจรขยายกระแสเข้าไป เมื่อกลับมาทดลองอีกครั้งหนึ่งก็ปรากฏว่า CPU สามารถรับสัญญาณจาก SENSOR ได้

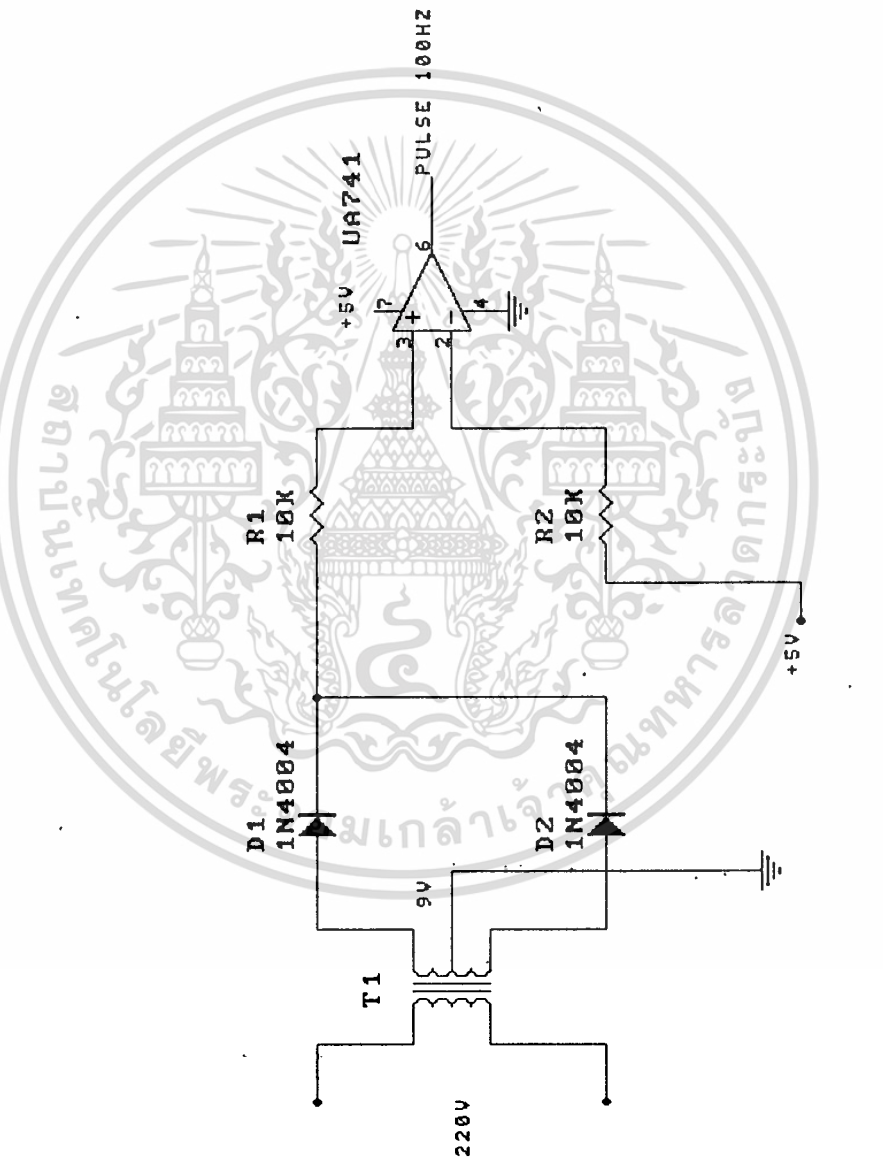
3.2 วงจรสร้างความถี่ 100 Hz

การที่เราต้องสร้างวงจรมีขึ้นมาก็เพื่อใช้เป็น Time Base ป้อนให้กับ CPU เพื่อจะได้ใช้เป็นค่าเปรียบเทียบกับอินพุตที่ผู้ใช้ป้อนเข้ามา ซึ่งจะได้กล่าวถึงการใช้งาน Time Base วงจรสร้างความถี่ 100 Hz นี้จะใช้ Diode 2 ตัวต่อแบบ Full-Wave Rectifiers รับไฟบ้าน 220 V ความถี่ 50 Hz เข้ามาเมื่อผ่านวงจร Full-Wave Rectifiers ความถี่ที่ได้จะเป็น 100 Hz มีเฟสของคลื่นเป็นบวกทั้งหมด หลังจากนั้นเรานำ Op-Amp 741 มาใช้เป็นตัวเปรียบเทียบ โดยป้อนความถี่ 100Hz ที่ได้เข้าทางขาบวกของ Op-Amp 741 และป้อนไฟ 5 V เข้าทางขาลบ ถ้าคลื่นที่เข้ามามีศักย์สูงกว่า 5 V เหนือที่พหุของ 741 ก็จะทำให้ค่า +5 V ออกมา ถ้าคลื่นที่เข้ามามีศักย์ต่ำกว่า 5 V เหนือที่พหุของ 741 ก็จะให้ค่า 0 ออกมา แสดงได้ดังรูปต่อไปนี้



รูปที่ 16. รูปแสดงสัญญาณที่ได้จากวงจรสร้างความถี่ 100 Hz

รูปที่ 17 รูปร่างสร้างควมถี่ 100 Hz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

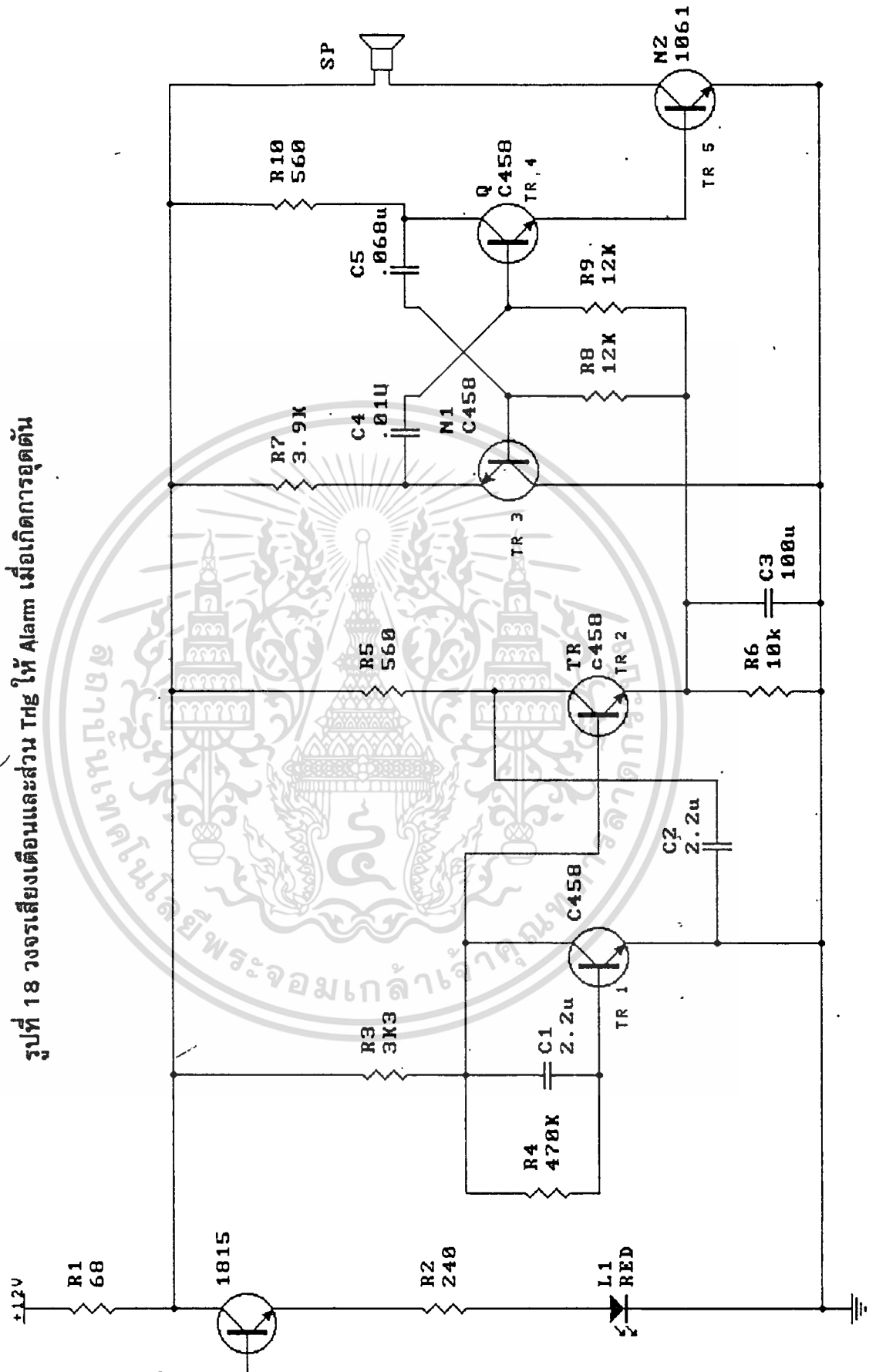
3.3 วงจรเสียงเตือนเมื่อเกิดการอุดตัน (Occlusion)

จากรูปจะเห็นว่า TR1, TR2 ต่อเป็นวงจรกำเนิดความถี่ต่ำ โดยความถี่นี้ขึ้นอยู่กับ R3, R4, R5 และ C 2.2 โดยความถี่นี้ จะส่งออกจากขา E ของ TR2 โดยมี R6 ต่อวงกราวด์และมี C 100/10 ทำหน้าที่ให้เกิดทางเสียง และที่ขา E ของ TR2 จะต่อไปควบคุมชุดกำเนิดความถี่สูง ทำให้เสียงที่ได้ดังเป็นท่วงๆ และมีท่วงเสียงเหมือนไซเรนจริงๆ โดยชุดกำเนิดความถี่สูงประกอบด้วย TR3, TR4, R7, R8, R9, R10 และ C 0.047 ทั้งสองตัวความถี่ที่ขึ้นสูงๆ จะส่งออกไปขยายให้แรงขึ้นโดย TR5 เพื่อส่งออกลำโพงต่อไป

ส่วน Trig ให้ Alarm เมื่อเกิดการอุดตัน (occlusion)

ส่วนนี้ต่อเข้ากับวงเสียงเตือนเมื่อเกิดการอุดตัน การที่เพิ่มวงจรเสียงเตือนเมื่อเกิดการอุดตันเข้าไปนี้ เพื่อให้สะดวกในการใช้ ผู้ใช้จะทราบได้ทันทีที่มีการอุดตันหรือน้ำเกลือไม่สามารถไหลได้ จะเกิดเสียงร้องเตือนและไฟสีแดงจะติด ในส่วนนี้จะต้องได้รับคำสั่งจาก CPU ส่งมาเพื่อให้ Trig ขา Base ของ Transistor โดยสัญญาณจาก CPU จะมีค่าประมาณ 5 โวลท์ ซึ่งพอที่จะ Trig ขา Base ให้ Transistor on ได้ และเมื่อ Transistor on แล้วก็จะทำให้ LED สีแดงสว่าง และวงจเสียงเตือนออกลำโพง แล้วทั้ง LED และเสียงเตือนจะดับเมื่อเรากดปุ่ม STOP วงจรเสียงเตือนและส่วน Trig ให้ Alarm แสดงได้ดังรูปต่อไปนี้

รูปที่ 18 วงจรเสียงเตือนและส่วน Tng ให้ Alarm เมื่อเกิดการอุดตัน

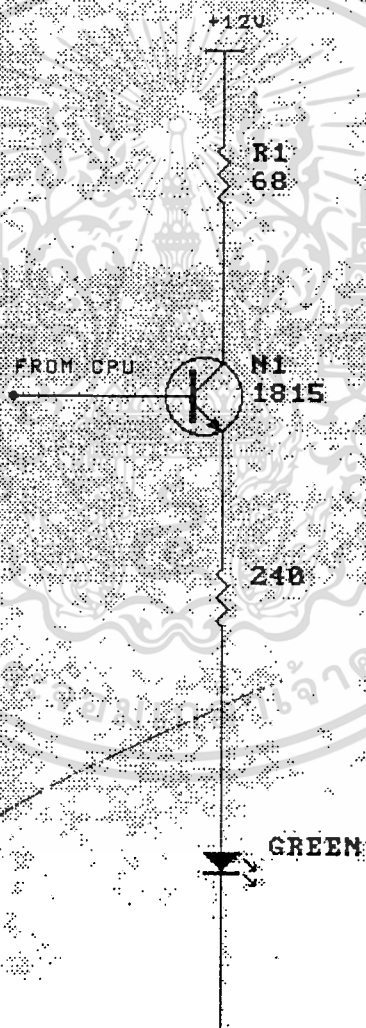


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนแสดงผลว่ายังคงมีสารละลายไหลอยู่ (Flowing)

ส่วน Flowing จะได้รับสัญญาณจาก CPU เช่นกัน ในขณะที่เครื่องกำลังทำงาน ก็คือในขณะที่มอเตอร์หมุน และช่องเหลวยังไหลอยู่ CPU ก็จะส่งสัญญาณ 5 โวลต์มา Triga Base ของ Transister ให้ on เพื่อให้ไฟ LED สีเขียวสว่าง ซึ่งแสดงว่าเครื่องกำลังทำงานเป็นปกติอยู่ แล้วไฟ LED จะดับก็ต่อเมื่อเกิดการอุดตันหรือให้ของเหลวจนหมดครบตามที่ตั้งไว้แล้ว ซึ่งต้องควบคุมโดย CPU เท่านั้น

รูปที่ 19 รูปแสดงการต่อ LED เมื่อยังคงมีการให้สารละลายอยู่เป็นปกติ



3.5 วงจรเสียงเตือนเมื่อการให้ของเหลวเรียบร้อยแล้ว

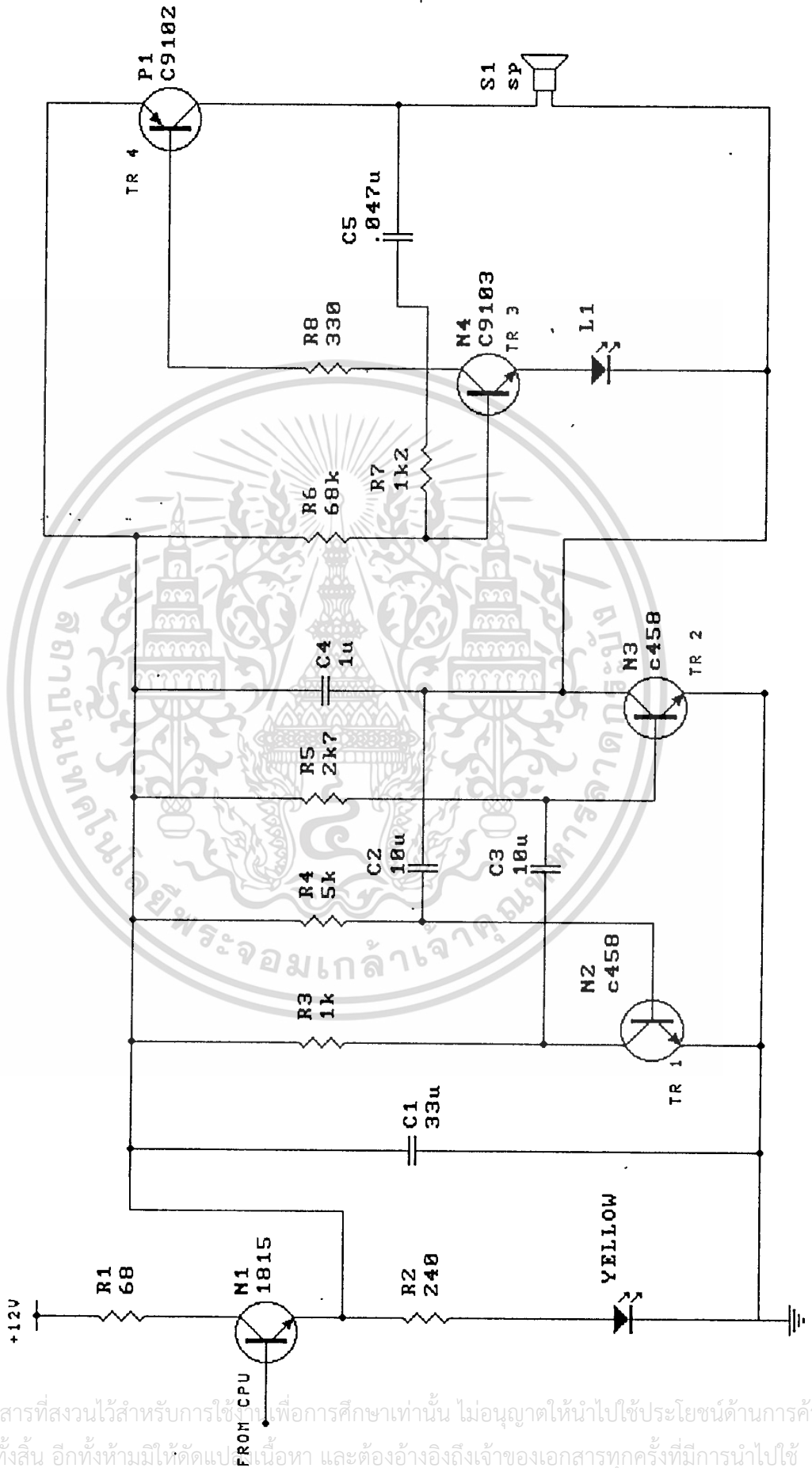
การทำงานของวงจร จะเห็นว่า TR3 , TR4 คือเป็นวงจรกำเนิดความถี่เสียงออกลำโพง ความถี่ชุดนี้จะถูกควบคุมอีกทีด้วย TR1 TR2 ซึ่ง TR1, TR2 คือเป็นวงจรกำเนิดความถี่ต่ำ โดยจะสลับกันทำงานทุกครั้งที่ TR2 ทำงาน ชุดกำเนิดความถี่เสียงจะทำงานด้วยทุกครั้ง

นั่นก็เมื่อ TR1 on จะทำให้ขา C ของ TR1 มีศักดาต่ำมากเป็นกราวด์ (Ground) ส่งผลให้ TR2 off เพราะไม่ได้รับการ Bias และเมื่อ TR2 off ศักดาที่ขา C ของมันก็จะขึ้นเป็นศักดาสูง ทำให้ TR3 และ TR4 off ไปด้วยเพราะที่ขา Cathode ของ Diode มีศักดาสูงกระแสจึงไม่สามารถไหลได้ ไฟลำโพงไม่ดังและ LED ไม่สว่าง แต่เมื่อ TR1 off ที่ขา C ของมันจะมีศักดาสูงส่งผลให้เกิดการ Bias ที่ TR2 ทำให้ศักดาที่ขา E ของ TR2 เป็นกราวด์ ผลอันนี้จะทำให้ ขา Cathode ของ Diode มีศักดาต่ำ กระแสก็สามารถไหลได้ TR3 และ TR4 ก็จะ on เสร็จแล้ว LED ก็จะสว่างและลำโพงจะดัง ดังนั้นการ on ของ TR3 ,TR4 จะทำงานพร้อมกัน TR2 เสมอ

ส่วน Trig เพื่อแสดงว่าการให้ของเหลวเรียบร้อยแล้ว

สัญญาณที่มาจาก Trig ขา Base ของ Transister จะเป็นคำสั่งที่ CPU ส่งมาเป็นสัญญาณ 5 โวลท์ เมื่อ CPU ประมวลผลแล้วว่า การให้ของเหลวครบตามที่ผู้ใช้กำหนดไว้ หรือของเหลวหมดแล้ว ก็จะ Trig ให้ Transister on เพื่อให้ LED สีเหลืองติด และมีเสียงเตือนเช่นกัน โดยที่ทั้ง LED และเสียงเตือนจะดับเมื่อกดปุ่ม STOP รูปแสดงวงจรเสียงเตือนเมื่อการให้ของเหลวเรียบร้อยแล้วและส่วน Trig เพื่อแสดงผลเป็นดังนี้

รูปที่ 20 รูปร่างเสียงเตือนและส่วน TRg เพื่อแสดงว่าการให้ของเหลวเรียบร้อยแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การควบคุมการทำงานโดยใช้ Microcontroller

4.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) คือไมโครคอมพิวเตอร์แบบที่มีขนาดเล็ก โดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิพเดียว เหมาะสำหรับงานควบคุมอุปกรณ์อื่น ๆ แบบอัตโนมัติ เพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51 หรือ MCS51 อินเทลแกเบอร์ 8051 และ 8052 จะมีโครงสร้างต่างกันเล็กน้อยดังแสดงในรูป

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

รูปที่ 21 ตารางของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวในตระกูล 51

จากตารางในรูป แต่ละคอลัมน์จะบอกถึงคุณสมบัตินี้หรือโครงสร้างของไมโครคอนโทรลเลอร์แต่ละเบอร์ในตระกูล MCS-51 เช่นมี ROM หรือ RAM ภายในเท่าใด ถ้าเป็นรุ่นที่ไม่มี ROM อยู่ภายในจะเป็นเบอร์อะไรหรือถ้าเป็นรุ่นที่มีหน่วยความจำสำหรับโปรแกรมเป็นแบบ EPROM จะเป็นเบอร์อะไร เช่นในบรรทัดแรกจะบอกว่า 8051 มี ROM อยู่ภายในขนาด 4 กิโลไบต์ แต่ถ้าเป็นเบอร์ 8031 จะไม่มี ROM ขนาด 4 กิโลไบต์อยู่ภายใน นอกจากนี้ในตารางยังจะบอกว่า ไมโครคอนโทรลเลอร์เบอร์นั้น มีพอร์ทสำหรับอ่านเขียนขนาด 8 บิตอยู่ที่ชุด (8 Bit I/O PORT), มี Timer/Counter ขนาด 16 บิตที่ชุด (16 Bit Timer/Counter) และยังบอกถึงคุณสมบัตินี้ๆอีก ทำให้ผู้ใช้สามารถเลือกใช้ไมโครคอนโทรลเลอร์แต่ละเบอร์ที่เหมาะสมกับงานใช้งานได้อย่างดีที่สุด

MCS-51 ผลิตโดยบริษัท Intel มีการทำงานเป็นแบบ 8 บิตหมายความว่าส่วนที่ทำหน้าที่ในการคำนวณ (Arithmetic Logic Unit, ALU) จะทำงานสูงสุดทีละ 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

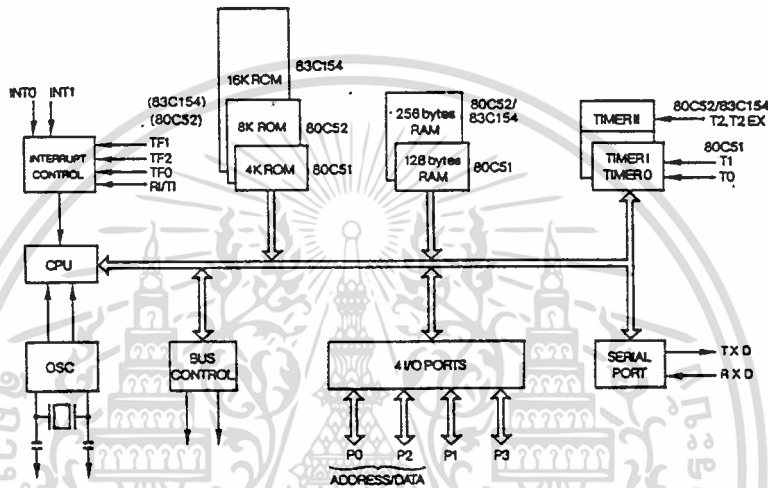
ข้อดีของ MCS-51

- สามารถนำข้อมูลมา AND , OR กันหรือทำ Complement ทั้งทีละ 8 บิตและ 1 บิต
- สามารถใช้กับหน่วยความจำสำหรับโปรแกรม (Program Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับเก็บชุดคำสั่งที่จะให้ MCS-51 ทำงานได้สูงสุด 64 กิโลไบต์(64*1024 ไบต์) ทำให้เขียนโปรแกรมควบคุมการทำงานได้มาก
- สามารถติดต่อกับหน่วยความจำสำหรับข้อมูล (Data Memory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรมได้สูงสุด 64 กิโลไบต์
- ใน 8051 และ 8751 มีหน่วยความจำสำหรับโปรแกรมจำนวน 4 กิโลไบต์ (ใน 8052 และ 8752 หน่วยความจำสำหรับโปรแกรมจำนวน 8 กิโลไบต์) อยู่ภายในวงจรรวมทำให้ไม่ต้องต่อหน่วยความจำสำหรับโปรแกรมอยู่ภายนอก ระบบรวมทั้งหมดจึงมีขนาดเล็กและสัญญาณรบกวนจากภายนอกจะทำให้ MCS-51 ทำงานผิดพลาดได้ยาก
- มีพอร์ตแบบขนาน (Parallel Port) สำหรับข้อมูลเข้าและออกจำนวน 32 บิตที่ข้อมูลแต่ละบิตเป็นอิสระต่อกัน
- มีวงจร Timer/Counter ขนาด 16 บิต 2 ชุด (8032 มี 3 ชุด) ที่ทำงานในโหมดต่างๆได้ถึง 4 โหมด
- มี Universal Asynchronous Receiver Transmitter (UART) สำหรับรับ-ส่งข้อมูลอนุกรม(Serial) แบบ Full Duplex ที่สามารถเลือกรูปแบบการรับ-ส่งข้อมูลได้ 4 แบบ
- มีแหล่งกำเนิดสัญญาณขอขัดจังหวะการทำงานของโปรแกรม(Interrupt Request Signal) 6 แหล่ง ซึ่งสามารถกระโดดไปทำงานตอบสนองการขัดจังหวะ (Interrupt Service Routine) ได้ต่าง ๆ กัน 5 ตำแหน่ง
- สามารถเลือกการทำงานให้อยู่ในโหมดของ Idle และ Power Down ซึ่งจะประหยัดการใช้กำลังไฟในการทำงาน

ซึ่งจากข้อดีดังกล่าว จึงทำให้ MCS-51 เป็นที่นิยมนำมาใช้ในการควบคุมระบบอัตโนมัติมากคุณสมบัติดังกล่าวบรรจุไว้ในวงจรรวมเดียว (Single Chip) ขนาด 40 ขา ดังนั้นจึงสามารถออกแบบให้ระบบทั้งหมดมีขนาดเล็กและการที่ทั้งหมดบรรจุอยู่ในวงจรรวมเดียวจึงทำให้การตรวจสอบหาข้อผิดพลาดในระบบง่ายไม่สลับซับซ้อน รวมทั้งลดปัญหาเรื่องการที่มีสัญญาณรบกวนในระบบจนทำให้การทำงานผิดพลาดไป แต่การที่จะนำเอา MCS-51 มาใช้งานได้จำเป็นที่จะต้องศึกษาและทำความเข้าใจถึงโครงสร้างและองค์ประกอบของ MCS-51 เสียก่อน แล้วถึงเขียนโปรแกรมควบคุมการทำงานของ MCS-51 ให้เป็นไปตามต้องการ

โครงสร้างของ 8051

ภายใน 8051 จะประกอบขึ้นด้วย GATE ต่างๆ เช่น AND, OR, NOT ซึ่ง GATE เหล่านี้จะถูกนำเอามาออกแบบให้มีหน้าที่การทำงานต่างๆเช่นวงจรถอดรหัสคำสั่ง (Instruction Decoder), วงจรสร้างสัญญาณนาฬิกา (Clock Signal Generator) โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆดังในโคแอสแกรมนี้



รูปที่ 22 โคแอสแกรมโครงสร้างของ 8051

โคแอสแกรมรูปนี้เป็นโครงสร้างใหญ่ๆของ 8051 เนื่องจากลักษณะของ 8051 เป็นคอมพิวเตอรืจึงประกอบด้วย 3 ส่วนหลักๆคือ

ส่วนที่ 1 คือ CPU(Central Processing Unit) หรือตัวประมวลผล ส่วนนี้จะมีส่วนที่ทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่นๆเรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่สัญญาณสำหรับการติดต่อกับหน่วยความจำ, อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออกจากตัว 8051 ซึ่งส่วนควบคุมการขัดจังหวะ (Interrupt Control) และส่วนควบคุมบัส (Bus Control) ก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณควบคุมจากส่วน CPU, นี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่ง(Instruction) ตามที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมา จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์ เพื่อให้ทุกๆส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

ในส่วน CPU นี้ ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (Arithmetic Logic Unit)ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูล เช่น การบวก,ลบ,คูณหรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์หรือหน่วยความจำที่ต้องการ

ส่วนที่ 2 คือหน่วยความจำ (Memory) มีไว้สำหรับจัดจำข้อมูล ถ้าจะให้เห็นภาพพจน์ของหน่วยความจำได้ดีก็คือ หน่วยความจำเปรียบเหมือนกล่องเก็บเอกสารจำนวนมากที่นำมาต่อเรียงกันไว้ แต่ละกล่องก็มีเอกสาร 1 แผ่น ดังในรูปมีกล่องเอกสารทั้งหมด 15 กล่อง

1	2	3	4	15
---	---	---	---	-------	----

รูปที่ 23 ภาพเสมือนของหน่วยความจำ

ถ้าต้องการเอาเอกสารออกจากกล่องใด หรือเอาเอกสารไปเก็บที่กล่องใด จะต้องรู้หมายเลขกล่องข้อมูลเสียก่อน ซึ่งถ้าเป็นหน่วยความจำแล้วหมายเลขของกล่องก็คือตำแหน่งของหน่วยความจำหรือแอสแตรส (Address) นั่นเอง การเอาข้อมูลไปเก็บในหน่วยความจำเรียกว่าการเขียน (Write) และการเอาข้อมูลออกจากหน่วยความจำเรียกว่า การอ่าน (Read) ข้อมูล ซึ่งแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลได้เพียงค่าเดียวเท่านั้น ในไมโครโปรเซสเซอร์ทั่วไปรวมทั้ง 8051 นั้น ข้อมูลในแต่ละตำแหน่งของหน่วยความจำจะมีค่าได้เพียง 8 หลักของเลขฐาน 2 (8 บิตเท่ากับ 1 ไบต์) ดังนั้นแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลมีค่าได้ระหว่าง 0 ถึง 255 (00000000 ถึง 11111111 ในเลขฐาน 2) แต่จำนวนตำแหน่งที่จะเก็บข้อมูลได้ขึ้นกับไมโครโปรเซสเซอร์แต่ละเบอร์ การติดต่อกับหน่วยความจำต้องมีสัญญาณ 3 กลุ่มคือ

1. แอสแตรสหรือค่าตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ใน 8051 จะติดต่อกับหน่วยความจำประเภท Program Memory หรือ Data Memory ได้สูงสุดชนิดละ 65536 ตำแหน่ง ดังนั้นการอ้างอิงแต่ละตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น (2^{16} เท่ากับ $64 \times 1024 = 65536$)

2. ข้อมูลที่จะอ่านหรือเขียนกับหน่วยความจำที่ตำแหน่งในข้อ 1

3. สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำ เพื่อขอกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล

สัญญาณเหล่านี้จะถูกวงจรควบคุมภายใน 8051 สร้างมาจากวงจรลอจิกของคำสั่งที่ 8051 อ่านจากหน่วยความจำ Program Memory เข้าไปทำงานนั่นเอง ในรูปไดอะแกรมโครงสร้างของ 8051 หน่วยความจำได้แก่ 4K ROM และ 128 Byte RAM ซึ่งขนาดของหน่วยความจำนี้มีขนาดต่าง ๆ กันตามเบอร์ของไมโครคอนโทรลเลอร์ซึ่งจะอธิบายโดยละเอียดต่อไป

ส่วนที่ 3 อุปกรณ์อินพุตและเอาต์พุต (Input/Output Device) เป็นส่วนที่ใช้ส่งข้อมูลเข้าหรือออกจาก 8051 ติดต่อกับภายนอกได้ อุปกรณ์อินพุตและเอาต์พุตได้แก่ 4 I/O Port, Timer 0, Timer 1, Serial Port

1. 4 I/O Port: คำว่าพอร์ทหมายถึงจุดที่จะติดต่อกับส่วนที่อยู่ภายนอก 4 I/O Port ของ 8051 เป็นที่ใช้สำหรับรับ-ส่งข้อมูล ซึ่งเป็นสัญญาณดิจิทัลเข้าหรือออกจากตัว MCS-51 พอร์ททั้งหมดมี 4 พอร์ท โดยแต่ละพอร์ทจะรับ-ส่งข้อมูลได้ 8 บิต มีพอร์ท P0, P1, P2 และ P3 บางพอร์ทจะใช้ทำงานมากกว่า 1 อย่างก็ได้ เช่นพอร์ท P0 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P2 จะใช้สำหรับถาวรส่งค่าตำแหน่ง (Address) ของหน่วยความจำที่ต้องการจะติดต่อ และพอร์ท PO จะใช้รับส่งข้อมูล เมื่อติดต่อกับหน่วยความจำได้ด้วยแต่สิ่งเหล่านี้ไม่ได้เกิดขึ้นในเวลาเดียวกันแต่จะใช้วิธีทำงานตามลำดับ โดยควบคุมจากสัญญาณควบคุม (Control) ที่ถอดรหัสมาจากแต่ละคำสั่งที่ให้อุปกรณ์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงจากสัญญาณนาฬิกา

2. *Timer 0 และ Timer 1* เป็นวงจรนับที่สามารถกำหนดให้ทำการนับจำนวนไชนีเซลของสัญญาณที่ต่อจากภายนอก 8051 หรือจำนวนไชนีเซลของสัญญาณนาฬิกาภายใน 8051 ก็ได้ ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย CPU

3. *Serial Port หรือพอร์ทอนุกรม* CPU จะอ่านและเขียนข้อมูลกับ Serial Port เป็นแบบ 8บิต แต่ข้อมูลจะถูกส่งออกจาก 8051 เรียงไปที่ละบิตออกจากขา TXD และถาวรรับข้อมูลก็จะรับเข้ามาที่ละบิตทางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิต เพื่อให้ CPU อ่านไปใช้งานต่อไป

8051 มีพอร์ทให้ใช้งานได้หลายแบบ ทำให้สะดวกแก่การนำไปใช้งานตงๆมากมาย การจะนำพอร์ทเหล่านี้ไปใช้งานได้จะต้องเขียนโปรแกรมขึ้นมาควบคุมที่จะได้กล่าวต่อไป

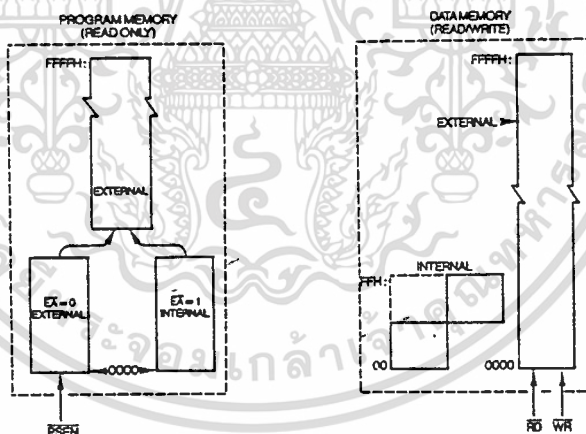


การจัดหน่วยความจำของ 8051

หน่วยความจำของ 8051 แบ่งออกได้เป็น 2 แบบตามลักษณะการใช้งานคือ

1. **Program Memory:** เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปรหัสภาษาเครื่อง (Machine Language) ซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่น ๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้จะต้องเป็นแบบ Read Only Memory (ROM) และผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ (หน่วยความจำแบบ ROM เป็นแบบ Non-volatile ซึ่งเมื่อปิดไฟแล้วข้อมูลก็จะไม่มีการสูญหาย)

การเขียนข้อมูลลงไปบน ROM จะต้องใช้เครื่องมือพิเศษ ในระหว่างการทำงานของ 8051 ผู้ใช้จะไม่สามารถใช้คำสั่งทำการเขียนข้อมูลลงในหน่วยความจำแบบนี้ได้ จำนวนตำแหน่งสูงสุดของหน่วยความจำแบบนี้ที่ 8051 จะใช้งานได้คือ 65536 ตำแหน่ง ค่าของตำแหน่ง (Address) จะเขียนเป็นเลขฐาน 16 ได้ตั้งแต่ 0000H ถึง FFFFH หน่วยความจำตำแหน่ง 0000H ถึง 0FFFH จำนวน 4 กิโลไบต์นั้น ผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอก 8051 (ไมโครคอนโทรลเลอร์เบอร์ เช่น 8052 จะมีขนาดของ ROM ส่วนนี้ได้ถึง 8 กิโลไบต์ ตำแหน่ง 0000H ถึง 1FFFH) ถ้าต้องการให้ 8051 ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน 8051 ก็ให้พินสัญญาณสถานะลอจิก High (1) เข้าที่ขา EA ของ 8051 ส่วนหน่วยความจำที่ตำแหน่ง 1FFFH ถึง FFFFH จะต้องต่ออยู่ภายนอก 8051 เสมอ ดังในแผนภูมิหน่วยความจำในรูป



รูปที่ 24 แผนภูมิหน่วยความจำของ 8051

Internal Memory: หมายถึงหน่วยความจำที่อยู่ภายใน 8051 ส่วน **External Memory** หมายถึงหน่วยความจำที่อยู่ภายนอก 8051

ไมโครคอนโทรลเลอร์เบอร์ 8031, 8051 และ 8751 นั้น โดยโครงสร้างและรหัสคำสั่งจะเหมือนกันทุกประการแตกต่างกันที่

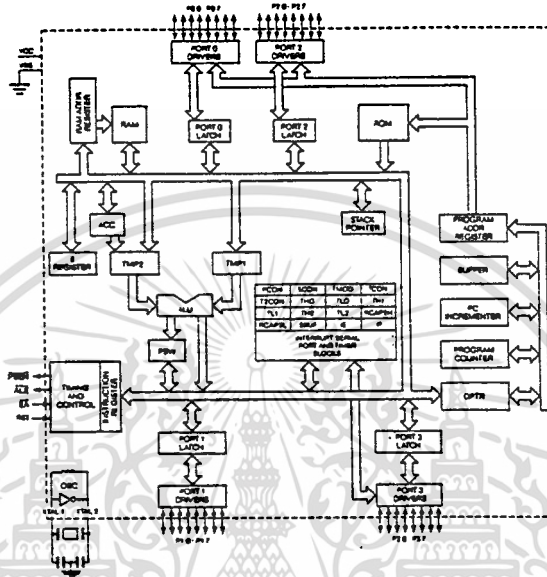
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 8031 จะไม่มี ROM ขนาด 4 กิโลไบต์อยู่ภายใน ผู้ใช้จะต้องเลือกการใช้งาน Program Memory อยู่ภายนอกวงจรทั้งหมด 64 กิโลไบต์
- 8051 จะมี ROM ขนาด 4 กิโลไบต์ อยู่ภายใน ถ้าต้องการเก็บคำสั่งควบคุมการทำงาน ไว้ในหน่วยความจำส่วนนี้ จะส่งโปรแกรมคำสั่งไปให้โรงงานผู้ผลิตทำการเขียนใส่ใน ROM ให้ตั้งแต่ในขั้นตอนของการผลิตวงจรรวม ผู้ใช้ไม่สามารถแก้ไขโปรแกรมได้เอง ถ้าจะนำมาใช้งานโดยเก็บโปรแกรมไว้ในหน่วยความจำช่วง 4 กิโลไบต์แรกอยู่ภายนอกก็สามารถทำได้ โดยการต่อ ROM ไว้ภายนอก แล้วต่อขา EA ของ 8051 ไว้กับสัญญาณที่สภาวะลอจิกเป็น 0
- 8751 จะมีหน่วยความจำ ขนาด 4 กิโลไบต์ เป็นแบบ EPROM (Erasable Program Read Only Memory) อยู่ภายในวงจรรวมเอาไว้ใช้แก้ไขโปรแกรมที่จะให้ 8751 ทำงาน ผู้ใช้สามารถเขียนคำสั่งลงไปให้ EPROM ได้เองโดยใช้เครื่องมือที่เรียกว่าเครื่องโปรแกรม EPROM (EPROM Programmer) และผู้ใช้สามารถแก้ไขโปรแกรมที่อยู่ใน EPROM ได้โดยการล้างข้อมูลในทุกตำแหน่งของ EPROM ออก ด้วยการฉายแสงอุลตราไวโอเล็ต (Ultraviolet) ผ่านกระจกใสของวงจรรวมเข้าไปยังวงจรรวมภายใน ตามเวลาที่กำหนดในคู่มือเฉพาะ (Data sheet) ของ 8751 จากนั้นก็ใช้เครื่องโปรแกรม EPROM เขียนโปรแกรมลงไปใหม่ 8751 นี้จะสะดวกมากสำหรับการพัฒนาโปรแกรม

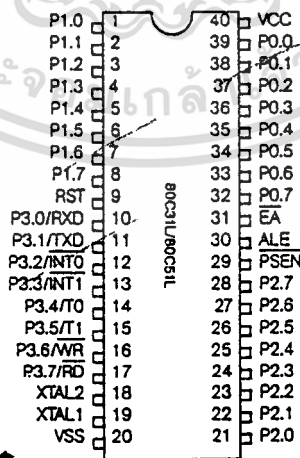
2. Data Memory เป็นหน่วยความจำที่ 8051 จะใช้สำหรับพัก เก็บข้อมูล แล้วเรียกมาใช้ใหม่ในระหว่างการทำงานของ 8051 การอ่านหรือเขียนข้อมูลจากหน่วยความจำ จะกระทำโดยคำสั่งที่เก็บไว้ใน Program Memory หน่วยความจำนี้เป็นประเภท Random Access Memory (RAM) ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้จะไม่สูญหาย แต่ถ้าปิดเครื่องหรือไม่จ่ายไฟให้แก่ RAM แล้ว ข้อมูลใน RAM ก็จะไม่สูญหาย การสูญหายของข้อมูลไม่ได้หมายความว่าอะไรอยู่เลย แต่เป็นกรณีที่ข้อมูลใหม่ซึ่งไม่ใช่ข้อมูลที่เก็บไว้เดิมเข้ามาอยู่แทนที่ เช่นเดิมเก็บข้อมูล 18H ไว้ที่ตำแหน่ง 1900H เมื่อปิดไฟแล้วเปิดใหม่ ข้อมูลที่ตำแหน่ง 1900H จะไม่ใช่ 18H อาจเป็นอะไรก็ได้ ซึ่งเรียกการเกิดลักษณะแบบนี้ว่าข้อมูลสูญหายไป หน่วยความจำแบบ Data Memory ของ 8051 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายใน 8051 จำนวน 128 ไบต์ ที่ตำแหน่ง 00H ถึง 7FH (เบอร์ 8052 จะมี 256 ไบต์ อยู่ที่ตำแหน่ง 00H ถึง FFH) และอีกชุดหนึ่งจะต้องต่ออยู่ภายนอกวงจรรวม 8051 มีได้สูงสุด 65536 ไบต์ (64 กิโลไบต์) อยู่ที่ตำแหน่ง 0000H ถึง FFFFH ดังแสดงในแผนภูมิหน่วยความจำของ 8051 หน่วยความจำแบบ Data Memory ภายใน 8051 ที่ตำแหน่ง 80H ถึง FFH นั้นไม่ได้มีอยู่ที่ตำแหน่ง จะมีเฉพาะในบางตำแหน่งซึ่งเรียกหน่วยความจำบางตำแหน่งนี้ว่า Special Function Register (SFR) เพราะจะใช้หน่วยความจำเหล่านี้สำหรับงานพิเศษเท่านั้น แต่ละตำแหน่งของหน่วยความจำแบบ SFR นี้ อาจเป็น RAM หรือวงจรรนับ (Counter) วงจรตั้งเวลา (Timer) ก็ได้ เช่นเป็น Timer 0, Timer 1 ดังนั้นใน 8051 จึงไม่ถือว่า SFR เป็น Data Memory ถ้าเป็น 8052 ซึ่งมี Data Memory ขนาด 256 ไบต์ จะใช้บางตำแหน่งของหน่วยความจำช่วงตำแหน่ง 80H ถึง FFH เป็น SFR ส่วนตำแหน่งอื่นที่เหลือก็เป็น RAM เหมือนกับหน่วยความจำช่วง 00H ถึง 7FH นั่นเอง

4.2 สถาปัตยกรรมของ 8051

รูปข้างล่างต่อไปนี้ จะเป็นสถาปัตยกรรมภายในของ 8051 ซึ่งจะอธิบายถึงส่วนย่อยๆของภายใน 8051 เพียงชีพเดียว และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา (Pin) ของ 8051 ที่มีอยู่ 40 ขา ดังรูป จะแสดง โดอะแกรมขาของ 8051 แบบ DIP



รูปที่ 25 สถาปัตยกรรมภายในของ 8051

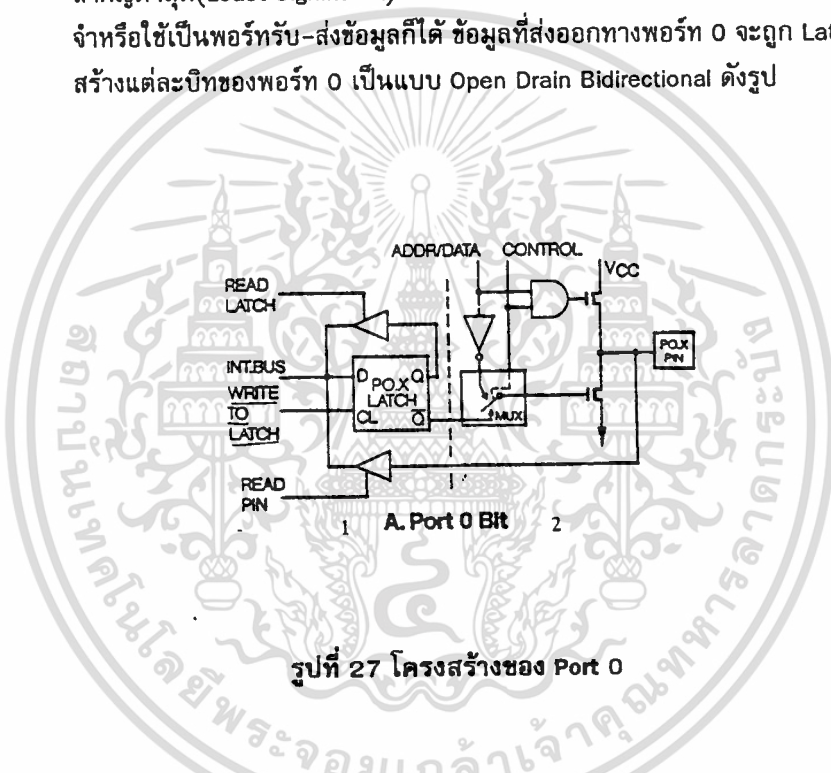


รูปที่ 26 โดอะแกรมขาของ 8051 แบบ DIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาตให้หน้าไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8051 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 จะมีขาอยู่ข้างละ 20 ขา ขารวมทั้งหมด 40 ขานั้นจะใช้งานต่าง ๆ กันดังนี้คือ

- V_{cc} ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลเตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง
- V_{ss} ขา 20 เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน
- Port 0 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ ดังแสดงในรูปที่แล้ว แต่ละขาจะเขียนว่า PO.0 , PO.1 ,, PO.7 นั้น PO.7 หมายถึงบิต 7 ของพอร์ท 0 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด (Most Significant) และ PO.0 คือบิต 0 ของพอร์ท 0 เป็นบิตที่มีนัยสำคัญต่ำสุด (Least Significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับ-ส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ทรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ท 0 จะถูก Latch ไว้ที่ขาพอร์ท โครงสร้างแต่ละบิตของพอร์ท 0 เป็นแบบ Open Drain Bidirectional ดังรูป



รูปที่ 27 โครงสร้างของ Port 0

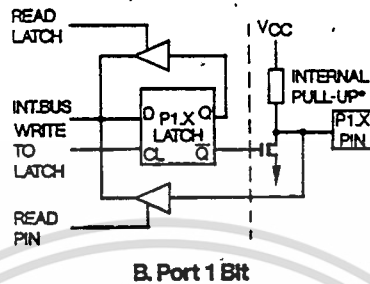
ในรูปนี้ เมื่อมีคำสั่งการเขียนข้อมูลมายังพอร์ท 0 ข้อมูลจาก Internal Data Bus จะถูก Latch ไว้ที่ D-FF โดยสัญญาณ "Write to Latch" ที่ถูกสร้างจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ท 0 จะอ่านได้ 2 แบบ คือการอ่านข้อมูลที่ส่งไปเก็บไว้ที่พอร์ท ก็จะมีสัญญาณ Read Latch มาเพื่ออ่านข้อมูลจาก D-FF กลับเข้าไปยัง Internal Data Bus การอ่านข้อมูลอีกแบบก็คือ การอ่านสถานะของสัญญาณที่เข้ามาทางพอร์ท 0 ก็จะมีสัญญาณ Read Pin มาควบคุมการอ่าน พอร์ท 0 จะใช้งานได้หลายอย่างดังนี้

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำภายนอก ที่ต้องการติดต่อกับ ตำแหน่งหน่วยความจำที่สูงที่สุดที่จะติดต่อได้ก็คือ 64 kbyte จึงมีค่าตำแหน่งหน่วยความจำ 16 บิตของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิตล่างจะถูกส่งออกไปทางพอร์ท 0 และ 8 บิตบนจะส่งออกไปทางพอร์ท 2
2. ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory
3. ใช้รับ-ส่งข้อมูลผ่านทางพอร์ทโดยตรงในกรณีที่ไม่มีการใช้หน่วยความจำของ Program Memory หรือ

Data Memory ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า -
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

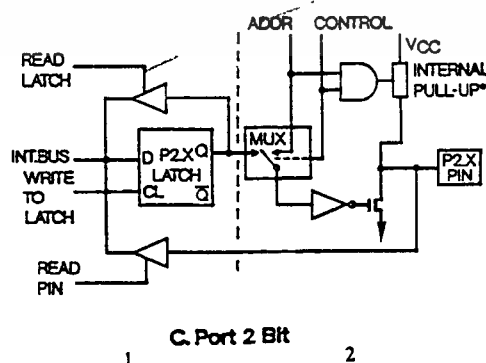
-Port 1 เป็นพอร์ตขนานขนาด 8 บิต คือขา P1.0 ถึง P1.7 (ขา1-ขา8) P1.0 หมายถึงบิต 0 ของพอร์ต 1 ซึ่งเป็นบิต Least Significant Bit และบิต P1.7 หมายถึงบิตที่ 7 ของพอร์ต 1 ซึ่งเป็นบิต Most Significant Bit โครงสร้างของพอร์ต 1 แต่ละบิตมีดังรูป



รูปที่ 28 โครงสร้างของ Port 1

ส่วนที่ 1 คือ Port 1 Latch ในรูปที่ 25 ซึ่งมีการทำงานเหมือนส่วนที่ 1 ของพอร์ต 0 ในรูปที่ 27 ส่วนที่ 2 คือ Port 1 Driver ในรูปที่ 1.5. Port 1 Driver นี้จะมีตัวต้านทานต่ออยู่เป็น Internal Pull Up พอร์ต 1 นี้จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ต 1 จะถูก Latch ไว้แล้วถูกส่งออกไปทางแต่ละขา ก่อนที่จะอ่านข้อมูลเข้าไปทางพอร์ต 1 จะต้องเขียน 1 ไปยังทุกบิตของพอร์ต 1 เสียก่อน เพื่อให้ FET อยู่ในสภาวะ OFF ก่อนมิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งออกมาค้างอยู่ที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่ขานี้ ก็จะถูกลัดวงจรลงกราวด์ โดยไม่สนใจว่าสภาวะของลอจิกสัญญาณที่เข้ามาจะเป็นอะไร ข้อมูลที่อ่านเข้าไปจึงเป็น 0 เสมอ

-Port 2 พอร์ตขนานขนาด 8 บิตคือขา P2.0 ถึง P2.7 (แทนบิต 0 ถึงบิต 7 ของพอร์ต 2) โครงสร้างของพอร์ต 2 แต่ละบิตมีดังรูป



รูปที่ 29 โครงสร้างของ Port 2

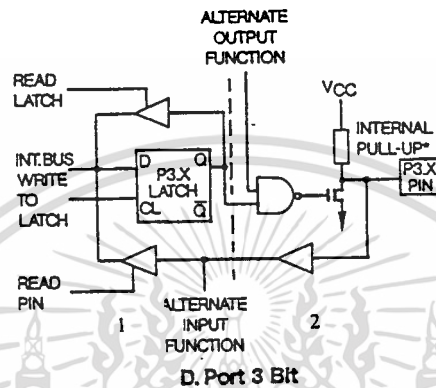
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะโครงสร้างจะเหมือนกับพอร์ท 0 แตกต่างกันในพอร์ท 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะ

คือ

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิตบนของค่าตำแหน่ง
2. ใช้เป็นพอร์ทรับและส่งข้อมูลกับภายนอก

-Port 3 คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับ พอร์ทนี้มีโครงสร้างดังรูป



รูปที่ 30 โครงสร้างของ Port 3

ส่วนที่ 1 ในรูปเป็นส่วน Latch ข้อมูลที่เขียนมายังพอร์ท 3 ทาง Internal Bus ซึ่งเหมือนกับพอร์ทอื่นๆ และพอร์ท 3 จะมี Internal Pull Up อยู่ทุกบิต แต่พอร์ท 3 นี้แต่ละบิตจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Alternative Output Function ที่สร้างมาจากส่วน Timing and Control สัญญาณ Alternative Output Function เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ท 3 ทำงานในฟังก์ชันอื่นและจุด Alternative Input Function เป็นจุดที่จะเอาสัญญาณไปเข้าส่วนอื่นตามการทำงานของบิตนั้น แต่ละบิตของพอร์ท 3 มีฟังก์ชันอื่นๆดังตารางต่อไปนี้

ขา PORT	ขา	การทำงานตามฟังก์ชันพิเศษ
P3.0	10	RXD เป็นขาที่ใช้รับข้อมูลแบบอนุกรม
P3.1	11	TXD เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม
P3.2	12	INT0 เป็นรับสัญญาณขัดจังหวะจากภายนอก
P3.3	13	INT1 เป็นรับสัญญาณขัดจังหวะจากภายนอก
P3.4	14	T0 ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0
P3.5	15	T1 ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 1
P3.6	16	WR ขาสัญญาณควบคุมการเขียนข้อมูลเข้าไปยังหน่วยความจำภายนอก 8051
P3.7	17	RD ขาสัญญาณควบคุมการอ่านข้อมูลเข้าไปยังหน่วยความจำภายนอก 8051

ตารางที่ 1 ตารางแสดงการทำงานตามฟังก์ชันพิเศษของ PORT 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-RST ขารี่เซ็ทนี้จะใช้ทำการรีเซ็ทการทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขาเข้ากับกราวด์ (Ground) ถ้าป้อนสัญญาณที่สภาวะลอจิก 1 เข้าไปที่ขานี้ จะเป็นการรีเซ็ทการทำงานของ 8051 ดังนั้นจึงสามารถต่อตัวเก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง +5 โวลท์ เพื่อให้เกิดการรีเซ็ทเมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power on reset การรีเซ็ทจะทำให้ค่าในรีจิสเตอร์ต่างๆเปลี่ยนไปเป็นค่าหนึ่งดังในตารางรูปต่อไป

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	00H
DPTR	0000H
P0-P3	0FFH
IP	00H
IE	0X000000B
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TLO	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H

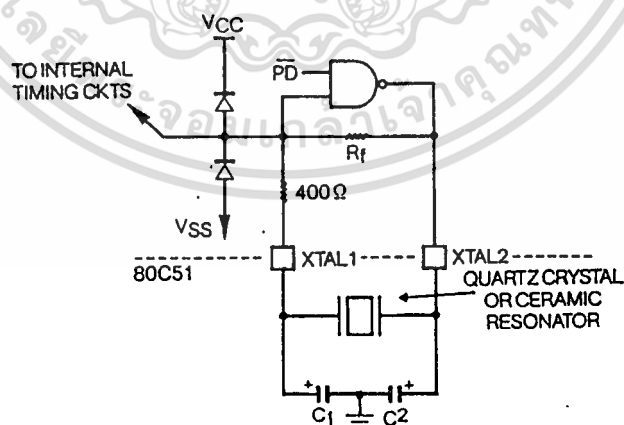
ตารางที่ 2 ตารางค่าของรีจิสเตอร์เมื่อเกิดการรีเซ็ท 8051

ในตารางนี้ช่องทางขา เป็นค่าของรีจิสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการรีเซ็ท ในรีจิสเตอร์ SBUP เมื่อสิ้นสุดการรีเซ็ทจะมีค่าที่ไม่แน่นอน และพอร์ทจะอยู่ในสภาวะลอจิก 1 ทุกบิตตลอดเวลาที่สัญญาณของขา RST เป็น High อยู่ เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซ็ท 8051 จะเริ่มทำงานจากคำสั่งที่อยู่ใน Program Memory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงานถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้จะต้องเขียนโปรแกรมมาเก็บไว้ที่ตำแหน่ง 0000H ในเครื่องไมโครคอมพิวเตอร์แบบบอร์ดเดี่ยว (Single Board Microcomputer) จะมีโปรแกรมที่เขียนเก็บไว้เริ่มจากตำแหน่ง 0000H นี้เรียกว่ามอนิเตอร์โปรแกรม (Monitor Program) จะคอยรับการกดแป้นพิมพ์ (Keyboard) และแสดงผลทางตัวแสดงผล (Display) แบบ 7 Segment

-ALE Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์ สัญญาณนี้จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะใช้บอกกับอุปกรณ์ 8051 ว่าขณะนี้สัญญาณนี้ Active (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิตล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เฉพาะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS

เอกสารนี้เป็นเอกสาร ได้ถึง 8 อินพุตการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

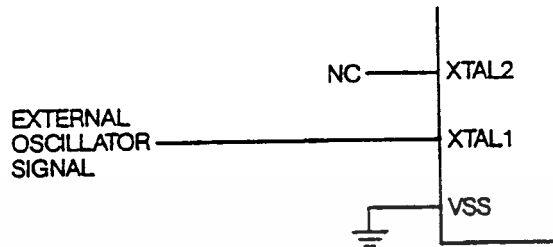
- PSEN** Program Store Enable เป็นขาที่ 29 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปรแกรมภายใน แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา PSEN นี้สามารถต่อไปยังขาอิพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท
- EA** External Accessเป็นขาที่ 31ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA นี้ แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 8051 จะต้องสร้างสัญญาณ PSEN ออกไปยังภายนอกเพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา EA เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ใน 8051 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่าง ๆ จาก ROM ภายใน 8051
- XTAL 1** เป็นขาที่ 19 ขานี้จะต่อเข้ากับ ขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบด้วยวงจรออสซิลเลเตอร์ ดังในรูปจะเห็นวงจรภายในของออสซิลเลเตอร์ NAND Gate จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิลเลตหรือไม่ ก็ขึ้นกับสัญญาณ PD ของรีจิสเตอร์ PCON ถ้าต้องการใช้สัญญาณนาฬิกา (Clock Signal) จากภายนอกมาเป็นสัญญาณนาฬิกา ควบคุมการทำงานของ 8051 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้ออสซิลเลเตอร์ภายใน ก็ให้ต่อ Crystal หรือเซรามิกเรโซเนเตอร์ดังรูปเข้าไป คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 PF



รูปที่ 31 วงจรออสซิลเลเตอร์ภายใน 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

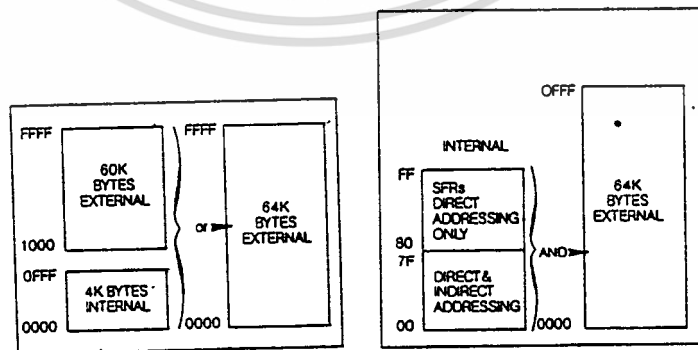
-XTAL2 เป็นขาที่ 18 ขานี้เป็นจุดเอาต์พุทของวงจรขยายแบบกลับเฟสสัญญาณ ที่ประกอบเป็นวงจรออสซิลเลเตอร์ (อินพุทคือขา XTAL1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอก มาเป็นสัญญาณนาฬิกาของ 8051 แล้วให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูป



รูปที่ 32 8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

4.3 รีจิสเตอร์ของ 8051

หน่วยความจำของ 8051 แบ่งออกเป็น 2 แบบคือ หน่วยความจำสำหรับโปรแกรม(Program Area) และ หน่วยความจำสำหรับเก็บข้อมูล(Data Area) ดังแสดงในไดอะแกรมรูปต่อไปนี้



รูปที่ 33 ไดอะแกรมภาพของหน่วยความจำ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำสำหรับโปรแกรมเป็นหน่วยความจำที่ 8051 ใช้สำหรับเก็บโปรแกรมภาษาเครื่องที่ 8051 จะทำงานเมื่อเริ่มป้อนไฟเลี้ยงให้ 8051 หรือมีการรีเซ็ต (Reset) 8051 จะทำให้เริ่มการทำงานจากคำสั่งในโปรแกรม ตำแหน่งที่ 0000H เมื่อทำงาน 1 คำสั่งก็จะทำให้รีจิสเตอร์ PC ที่ชี้ตำแหน่งโปรแกรมมีค่าเพิ่มขึ้นเพื่อชี้ตำแหน่งของคำสั่งต่อไป ตำแหน่งสุดท้ายของหน่วยความจำคือ FFFFH หน่วยความจำสำหรับโปรแกรมนี้อาจเลือกได้ว่าเป็นหน่วยความจำที่อยู่ใน 8051 หรือภายนอก 8051 ก็ได้ หน่วยความจำสูงสุดสำหรับโปรแกรมภายนอก 8051 มีได้ถึง 64 กิโลไบต์ ทำให้สามารถใช้งานได้อย่างกว้างขวาง หน่วยความจำในช่วงนี้ 8051 สามารถอ่านข้อมูลได้อย่างเดียว ไม่สามารถเขียนข้อมูลเข้าไปได้ระหว่างการทำงาน

หน่วยความจำสำหรับข้อมูล เป็นหน่วยความจำที่ 8051 ใช้สำหรับเก็บหรือพักข้อมูลระหว่างที่ทำงาน หน่วยความจำสำหรับข้อมูลมี 2 แบบ แบบหนึ่งมีขนาด 128 ไบต์อยู่ภายใน 8051 หน่วยความจำอีกแบบหนึ่งจะมีขนาด 64 กิโลไบต์ ต้องต่อเพิ่มเติมเข้าไปภายนอก 8051 หน่วยความจำภายในตำแหน่ง 0 ถึง 7FH นี้ สามารถอ้างถึงได้โดยตรง แต่หน่วยความจำตำแหน่ง 80H ถึง FFH นั้นเป็นรีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register, SFR) หน่วยความจำภายในช่วงนี้ใช้เป็นรีจิสเตอร์สำหรับงานเฉพาะอย่าง

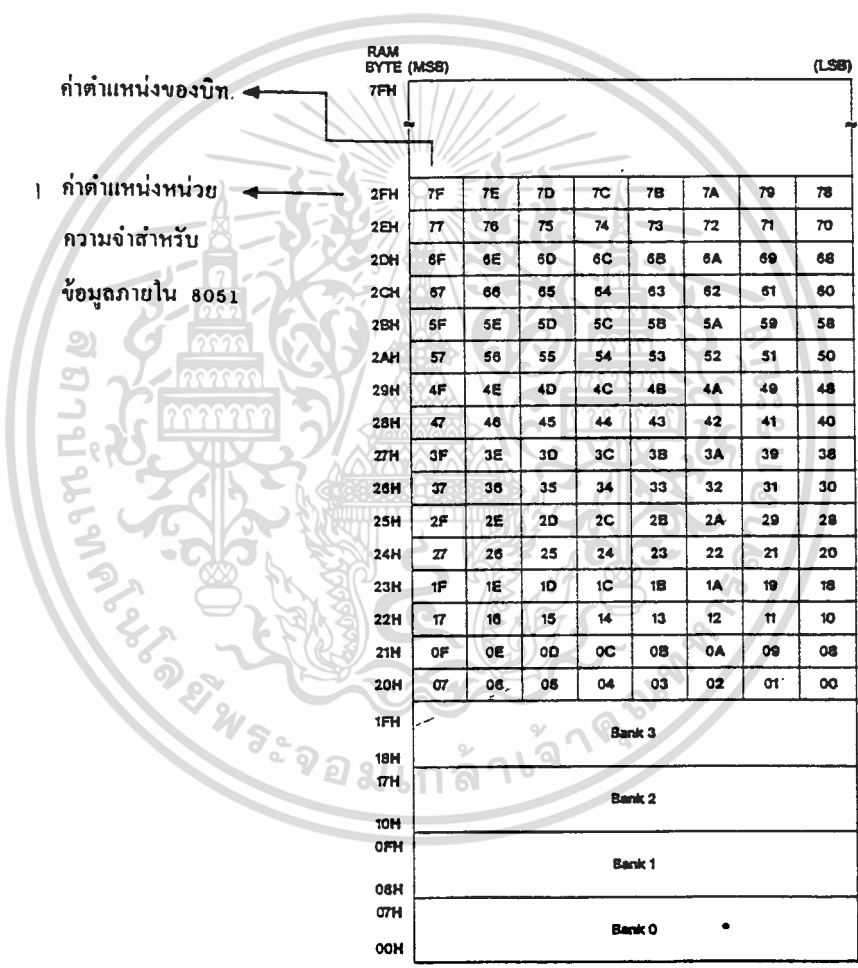
หน่วยความจำสำหรับข้อมูลภายใน 8051 ช่วง 00H ถึง 07FH สามารถแบ่งออกได้เป็น 3 กลุ่ม คือ

1. Register Blank 0.3 อยู่ในหน่วยความจำช่วงตำแหน่งที่ 00H ถึง 1FH หน่วยความจำนี้จะแบ่งออกเป็น 4 ชุด ชุดละ 8 ไบต์ แต่ละชุดเราเรียกว่า BANK แต่ละไบต์ใน 1 BANK จะมีชื่อของรีจิสเตอร์ว่า R0, R1, R2, R3, R4, R5, R6 และ R7 รีจิสเตอร์เหล่านี้จะเรียกใช้งานในระหว่างการทำงานของโปรแกรมได้อย่างสะดวก และรีจิสเตอร์เหล่านี้จะเป็นชื่อซ้ำกันในทุก BANK การใช้งานจึงต้องเรียกใช้งานทีละ BANK เท่านั้น โดยการกำหนดในรีจิสเตอร์ PSW ที่จะกล่าวถึงต่อไป เมื่อมีการรีเซ็ตการทำงาน ของ 8051 จะเริ่มการ ใช้งานรีจิสเตอร์ R0 ถึง R7 ที่ BANK 0 ซึ่งรีจิสเตอร์ R0 ถึง R7 ในแต่ละ BANK นั้นจะอ้างอิงในหน่วยความจำสำหรับข้อมูลภายใน 8051 ดังในตาราง

รีจิสเตอร์	ตำแหน่งหน่วยความจำ			
	BANK 0	BANK 1	BANK 2	BANK 3
R0	0	8	10	18
R1	1	9	11	19
R2	2	A	12	1A
R3	3	B	13	1B
R4	4	C	14	1C
R5	5	D	15	1D
R6	6	E	16	1E
R7	7	F	17	1F

ตารางที่ 3 ตารางอ้างอิงในหน่วยความจำสำหรับข้อมูลภายใน 8051

2. Bit Address Area เป็นหน่วยความจำในช่วงตำแหน่ง 20H ถึง 2FH หน่วยความจำแต่ละบิต ในช่วงของหน่วยความจำดังกล่าวจะสามารถตรวจสอบ หรือตั้งค่าเป็น 1 หรือ 0 ได้โดยการโปรแกรมภาษาเครื่อง แต่ละบิตของข้อมูลในหน่วยความจำช่วงนี้ จะมีค่าของตำแหน่งตั้งใน Memory Map เช่นบิตที่ 7 ของหน่วยความจำในตำแหน่ง 2FH จะมีค่าตำแหน่งเป็น 7FH นั่นเอง ในรูปข้างล่างนี้ ตัวเลขทางซ้ายเป็นค่าตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 8051 ซึ่งแต่ละบิตในตำแหน่งนั้นจะมีค่าเป็นเลขฐาน 16 ที่จะใช้เป็นค่าอ้างอิงในคำสั่งจัดการกับข้อมูลบิตนั้น



รูปที่ 34 แสดงค่าตำแหน่งของแต่ละบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. *Scratched Pod Area* เป็นช่วงของหน่วยความจำตำแหน่ง 30H ถึง 7FH หน่วยความจำช่วงนี้จะใช้สำหรับเก็บข้อมูลทั่วไป ถ้ารีจิสเตอร์ Stack Pointer ชี้อย่างหน่วยความจำช่วงนี้ จะต้องระวังไม่ให้เกิดการเขียนทับของข้อมูล อันจะทำให้การทำงานของโปรแกรมผิดพลาดได้

จากตารางคำสั่งในภาคผนวก จะมีคำอธิบายการใช้งานหรือการทำงานของแต่ละคำสั่งใน 8051 ไว้ด้วย คำสั่งของ 8051 เป็นคำสั่งที่มีประสิทธิภาพการทำงานสูงมาก ในขณะที่ 8051 ทำงานจะมีรีจิสเตอร์ ตัวหนึ่งที่เก็บสถานะ (Flag) ที่เกิดขึ้นระหว่างการคำนวณ เช่น ตัวทด (Carry) หรือจะใช้เลือก BANK ของรีจิสเตอร์ภายใน 8051 ก็ได้ รีจิสเตอร์นี้คือ Program Status Word (PSW) มีขนาด 8 บิตแต่ละบิตใช้เก็บสถานะการทำงานต่างๆไว้ดังรูป

(MSB)				(LSB)				
CY	AC	FO	RS1	RS0	OV	—	P	
Symbol	Position	Name and Significance	Symbol	Position	Name and Significance			
CY	PSW.7	Carry flag.	OV	PSW.2	Overflow flag.			
AC	PSW.6	Auxiliary Carry flag. (For BCD operations).	—	PSW.1	User definable flag.			
FO	PSW.5	Flag 0 (Available to the user for general purposes).	P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the Accumulator, i.e., even parity.			
RS1	PSW.4	Register bank select control bits 1 & 0. Set/cleared by software to determine working register bank (see Note).	Note : The contents of (RS1, RS0) enable the working register banks as follows :					
RS0	PSW.3							
					(0.0)—Bank 0	(00H—07H)		
					(0.1)—Bank 1	(08H—0FH)		
					(1.0)—Bank 2	(10H—17H)		
					(1.1)—Bank 3	(18H—1FH)		

รูปที่ 35 Program Status Word (PSW)

- PSW.0 บิต 0 เรียกว่าพาริตีบิต บิตนี้จะบอกว่าในรีจิสเตอร์ Accumulator หรือรีจิสเตอร์ A มี 1 เป็นจำนวนคี่หรือคู่ เช่นในรีจิสเตอร์ A ขนาด 8 บิต มี 1 อยู่ 3 ตัว และมี 0 อยู่ 5 ตัว ก็จะทำให้บิต PSW.0 นี้มีค่าเป็น 1 ถ้าใน Accumulator มี 1 อยู่เป็นจำนวนคู่ ก็จะทำให้บิตนี้มีค่าเป็น 0
- PSW.1 บิต 1 บิตนี้ไม่มีการใช้งาน
- PSW.2 บิต 2 บิตนี้เรียกว่า Overflow Flag เป็นบิตที่บอกว่าการคำนวณนั้นทำให้เกิดตัวทดขึ้นระหว่างการคำนวณ ตัวทอนี้เป็นตัวทดที่เกิดจากบิต 6 ไปยังบิต 7 มีประโยชน์เมื่อทำการคำนวณแบบ Signed Integer
- PSW.3 , PSW.4 เป็นบิตที่ 3 และบิตที่ 4 ทั้ง 2 บิตนี้จะใช้งานร่วมกันเพื่อเป็นตัวบอกว่าขณะนี้ใช้รีจิสเตอร์ R0 ถึง R7 ใน BANK ได้ ตัวอย่างเช่น บิต 4 และบิต 3 มีค่าเป็น 10₂ เป็นการเลือกรีจิสเตอร์ BANK 2 หมายความว่าในรหัสคำสั่งช่วยจำที่อ้างอิงถึง R0 ก็จะอ้างอิงหน่วยความจำ ภายในตำแหน่ง 10H
- PSW.5 บิต 5 เรียกว่า บิตเอนกประสงค์เป็นบิตที่ผู้ใช้สามารถใช้คำสั่งกำหนดค่าให้เป็น 0 หรือ 1 ก็ได้ โดยที่การทำงานของคำสั่งอื่นจะไม่ทำให้บิตนี้มีค่าเปลี่ยนแปลง บิตนี้มีประโยชน์สำหรับการส่งสถานะของโปรแกรมระหว่างเรียกการทำงานของโปรแกรมย่อย (Subroutine)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PSW.6 บิท 6 เรียกว่า Auxiliary Carry Flag เป็นบิทที่ใช้สำหรับเก็บตัวทศที่เกิดขึ้นระหว่างการคำนวณโดยตัวทศนี้เป็นตัวทศที่เกิดการคำนวณของบิท 3 ข้ามไปยังบิท 4
- PSW.7 บิท 7 เรียกว่า Carry Flag เป็นบิทที่บอกสถานะการคำนวณทางคณิตศาสตร์ว่าผลลัพธ์นั้นทำให้เกิดตัวทศขึ้นหรือไม่ เช่นการบวกเลข 2 จำนวนเข้าด้วยกันแล้วผลลัพธ์มีค่ามากกว่า 255 ก็จะทำให้เกิดตัวทศขึ้นเนื่องมาจาก Accumulator ที่ทำการบวกนี้สามารถเก็บข้อมูลได้เพียง 8 บิทเท่านั้นและทำให้บิทนี้มีค่าเป็น 1

4.4 กลุ่มคำสั่งใน MCS-51

ในหัวข้อนี้จะขอกล่าวถึงเพียงคร่าว ๆ เท่านั้น คำสั่งที่ใช้ควบคุมการทำงานของ MCS-51 มี 2 ประเภท คือ คำสั่งที่ต้องการข้อมูลมาดำเนินการเพื่อให้ได้ผลลัพธ์ และคำสั่งที่ไม่ต้องการข้อมูลมาดำเนินการ คำสั่งที่ต้องการข้อมูลจะมีวิธีเข้าถึงข้อมูลหลายวิธีดังนี้

วิธีการเข้าถึงข้อมูลในคำสั่ง

- วิธีการเข้าถึงข้อมูลโดยตรง (Direct Addressing) เป็นวิธีกำหนดตำแหน่งหน่วยความจำโดยตรงในคำสั่ง
- วิธีการเข้าถึงข้อมูลโดยทางอ้อม (Indirect Addressing) เป็นวิธีการเข้าถึงข้อมูลโดยค่าตำแหน่งหน่วยความจำจะอยู่ในรีจิสเตอร์เฉพาะบางตัว นั่นคือวิธีนี้จะใช้ค่าในรีจิสเตอร์เป็นตัวชี้ตำแหน่งหน่วยความจำ
- วิธีการเข้าถึงข้อมูลในรีจิสเตอร์ใช้งานโดยทั่วไป (Register Instruction) ข้อมูลที่ต้องการจะอยู่ในรีจิสเตอร์ใช้งานทั่วไป R0-R7 ของแต่ละกลุ่มรีจิสเตอร์ที่ถูกเลือกใช้งานในขณะนั้น
- วิธีการเข้าถึงข้อมูลในรีจิสเตอร์เฉพาะของคำสั่ง (Register-Specific) คำสั่งบางคำสั่งของ MCS-51 จะระบุไว้แล้วว่าต้องดำเนินการกับข้อมูลในรีจิสเตอร์ตัวใด เช่น Accumulator, DPTR, SP MCS-51 จะรู้ได้เองว่าจะประมวลผลกับรีจิสเตอร์ตัวใด
- วิธีการเข้าถึงข้อมูลที่กำหนดเองโดยตรง (Immediate Constant) เป็นการกำหนดข้อมูลให้กับคำสั่งโดยตรง ทั้งนี้ต้องใช้เครื่องหมาย “#” ระบุหน้าข้อมูลที่ต้องการ
- วิธีการเข้าถึงข้อมูลโดยใช้ตัวชี้อ้างอิง (Indexed Addressing) ข้อมูลที่ใช้วิธีการอ้างอิงแบบนี้จะเป็นข้อมูลที่อยู่ในหน่วยความจำสำหรับเก็บโปรแกรมภายในหรือภายนอกชิปเท่านั้น จุดประสงค์มีไว้เพื่อใช้ในการเปิดหาค่าข้อมูลที่เก็บไว้ในหน่วยความจำสำหรับเก็บโปรแกรมซึ่งข้อมูลไม่สูญหายแม้ไม่มีพลังงาน การอ้างอิงจะใช้ค่าของรีจิสเตอร์ใช้งานเฉพาะ DPTR หรือ PC มารวมกับค่าในรีจิสเตอร์ A เพื่อชี้ตำแหน่ง

คำสั่งใน MCS-51

คำสั่งทั้งหมดใน MCS-51 สามารถแยกเป็นประเภทตามลักษณะการทำงานได้ดังนี้

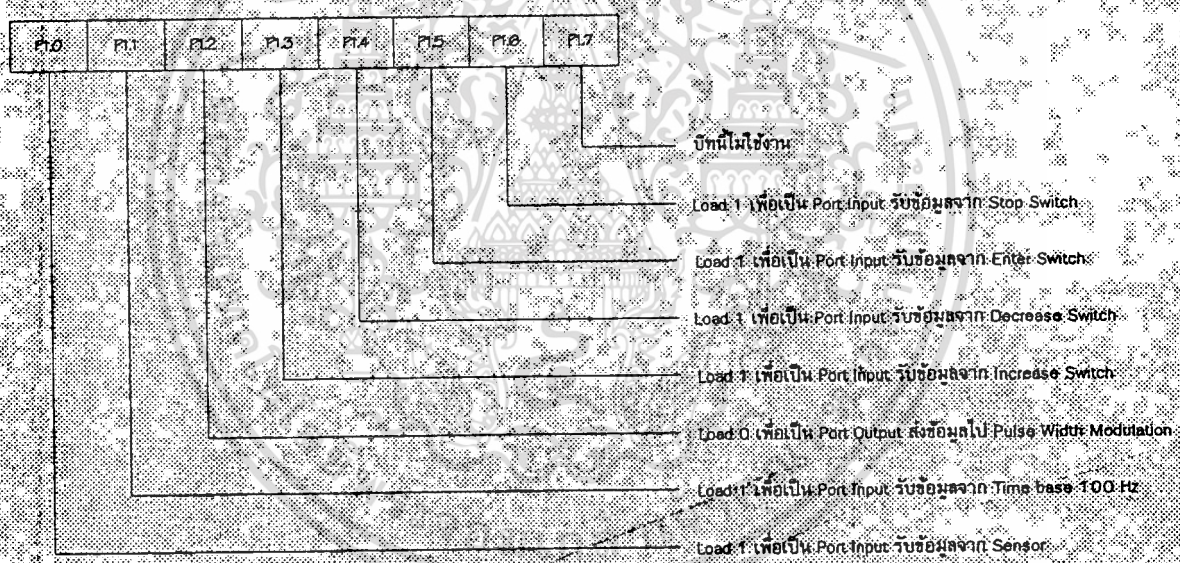
- กลุ่มคำสั่งทางคณิตศาสตร์ (Arithmetic Instruction) ประกอบด้วยคำสั่งในการบวก ลบ คูณ หารรวมทั้งคำสั่งในการเพิ่มค่าข้อมูล
- กลุ่มคำสั่งทางตรรกศาสตร์ (Logical Instruction) ประกอบด้วยกลุ่มคำสั่งทางตรรกศาสตร์ เช่น AND, OR, Exclusive-OR, Complement รวมทั้งคำสั่งสำหรับการเลื่อนบิตข้อมูลไปทางซ้ายหรือขวาโดยผ่านบิต Carry Flag หรือไม่ก็ได้
- กลุ่มคำสั่งเคลื่อนย้ายข้อมูล (Data Transfer Instruction) ใช้เพื่อนำข้อมูลที่ได้จากการประมวลผลไปเก็บยังหน่วยความจำบริเวณใดบริเวณหนึ่ง หรือเพื่อย้ายข้อมูลไปยังรีจิสเตอร์ เพื่อประมวลผล

- กลุ่มคำสั่งในการควบคุมลำดับการทำงานของโปรแกรม (Program Control Instruction) กลุ่มนี้ยังแบ่งออกเป็นคำสั่งที่ควบคุมลำดับการทำงานโดยมีเงื่อนไขและไม่มีเงื่อนไข ทำให้การเขียนโปรแกรมสะดวกมากขึ้น
- กลุ่มคำสั่งสำหรับการประมวลผลแบบบูลีน (Boolean Instruction) MCS-51 มีความสามารถพิเศษในการประมวลผลแบบบูลีนซึ่งไม่มีในไมโครโปรเซสเซอร์ชนิดอื่น การประมวลผลแบบนี้มีไว้สำหรับงานทางด้านควบคุมโดยเฉพาะ จึงทำให้ MCS-51 เหมาะสมสำหรับที่จะนำไปใช้กับอุปกรณ์ต่างๆได้เป็นอย่างดี

คำสั่งต่างๆที่ใช้กับ MCS-51 จะมีแสดงไว้ในตารางที่ภาคผนวก จะเห็นว่าจากความรู้คร่าวๆเรื่อง MCS-51 ดังที่ได้กล่าวมาแล้ว เราสามารถนำข้อมูลต่างๆไปเขียนโปรแกรมเพื่อควบคุมการใช้งานได้ตามที่เราต้องการ สำหรับเครื่องควบคุมการใช้สารละลายโดยใช้ Microcontroller ได้เลือกใช้พอร์ทของ Single Board ดังนี้

4.5 การเลือกใช้พอร์ทของ 8031

ในเครื่องนี้เราเลือกใช้ Port 1 ของ 8031 โดยเลือกใช้ 6 บิต ถ้าต้องการให้บิตใดเป็นอินพุตพอร์ท (Input Port) ต้อง Load ค่า 1 ส่งไปยังบิตนั้น ส่วนถ้าเราต้องการให้พอร์ทไหนเป็นเอาต์พุตพอร์ท (Output Port) สามารถเลือกใช้ได้เลยโดยไม่ต้อง Load ค่าอะไรส่งไปยังพอร์ท



หมายเหตุ ถ้าต้องการเซตให้ Port ใดเป็น Input Port ก็ Load ค่า 1 ส่งไปที่ Port นั้น
ถ้าต้องการเซตให้ Port ใดเป็น Output Port ก็ตามการใช้งานได้เลย

รูปที่ 36 รูปแสดงการเซต Port ของ 8031

เราเลือกใช้พอร์ท P1.0 รับ Input เข้ามาจาก SENSOR โดย Load ค่า 1 โยเก็บไว้ก่อน สัญญาณที่รับมาจาก SENSOR จะเป็นชบวน Pulse ที่แต่ละ Pulse จะแทนหยดน้ำ 1 หยดเพื่อให้ CPU นำไปประมวลผลต่อไป เลือกใช้พอร์ท P1.1 รับ Time Base ที่เราสร้างขึ้นโดย Time Base นี้มีความถี่ประมาณ 100 Hz เราจะใช้ Time Base นี้ส่งให้ CPU นำไปเปรียบเทียบกับปริมาตรที่ผู้ใช้ป้อนข้อมูลเข้ามา กำหนดให้พอร์ท P1.2 เป็น Output ส่งคำสั่งจาก CPU ไปยัง Pulse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Width Modulation เพื่อควบคุมการหมุนของมอเตอร์ ให้พอร์ต P1.3 รับ INPUT เข้ามาจากผู้ใช้เมื่อผู้ใช้ต้องการเพิ่ม ปริมาตรหรืออัตราการไหลให้เป็นไปตามที่ต้องการ สำหรับพอร์ต P1.4 จะรับคำสั่งจากผู้ใช้อีกเช่นกันเมื่อมีการตั้ง ปริมาตรหรืออัตราการไหลเกินกว่าที่ต้องการก็สามารถลดจำนวนที่ตั้งไปแล้วลงได้โดยส่งคำสั่งเข้ามาทางพอร์ตนี้ ส่วน พอร์ต P1.5 จะรับคำสั่งเพื่อให้เครื่องควบคุมการให้สารละลายทำงาน พอร์ต P1.6 จะรับคำสั่งให้หยุดการทำงานคือ มอเตอร์หยุดหมุน รวมทั้งหยุดเสียงสัญญาณเตือนต่าง ๆ

4.6 การใช้พอร์ตของ 8255

สำหรับเครื่องควบคุมการให้สารละลายนี้ได้เลือกใช้พอร์ตของ 8255 มาใช้เป็น Output Port โดย 8255 จะมีสาย PA0-PA7, PBO-PB7 และ PC0-PC7 เป็นทางผ่านของข้อมูล ระหว่างอุปกรณ์ภายในบอร์ด 8255 สายสัญญาณ เหล่านี้ถูกแบ่งออกเป็น 3 I/O Port ได้แก่ พอร์ต A (PA), พอร์ต B (PB) และพอร์ต C (PC) พอร์ตเหล่านี้แต่ละพอร์ต สามารถเป็นได้ทั้งพอร์ตอินพุตและเอาต์พุต แต่ละพอร์ตจะมีสายสัญญาณเชื่อมเข้ากับบัสข้อมูลภายในของ 8255 นอกจากนั้นยังมีกลุ่มของ Group A control และ Group B Control ซึ่งจะเป็นตัวกำหนดลักษณะการทำงานของทั้ง 3 I/O พอร์ต (8255 มีลักษณะการทำงานที่แตกต่างกันอยู่ 3 โหมด สามารถกำหนดได้โดยการโปรแกรมส่ง Control Word ให้กับ 8255 ซึ่งจะกล่าวถึงภายหลัง) นอกจากนี้ 8255 ยังมีส่วนที่ติดต่อกับ CPU โดยมี Data Bus Buffer ซึ่งจะ เป็นบัสไฟเออร์ให้กับบัสข้อมูลของ CPU และมีส่วน Read/Write Control Logic เป็นส่วนที่ควบคุมให้ข้อมูลเข้าหรือออกจาก รีจิสเตอร์ภายในตัวที่ถูกต้องและในเวลาที่เหมาะสม

รายละเอียดการจัดเรียงขาของ 8255

ในส่วนนี้จะพิจารณาหน้าที่ของขาแต่ละขาของ 8255 ซึ่งข้อมูลเหล่านี้จะมีประโยชน์ในการเชื่อมต่อเข้ากับ ระบบบัสของ CPU แต่ใน ET-Board ที่เราใช้ได้มีการต่อ 8255 ไว้ให้เลือกใช้งานเรียบร้อยแล้ว

ขาต่างๆของ 8255 มีดังนี้

DO-D7 : เป็นสายข้อมูลอินพุต/เอาต์พุตแบบ 2 ทิศทาง (Bi-Directional Bus) จะเป็นทางผ่านของข้อมูล ระหว่างพอร์ตต่างๆของ 8255 กับบัสข้อมูลของ CPU

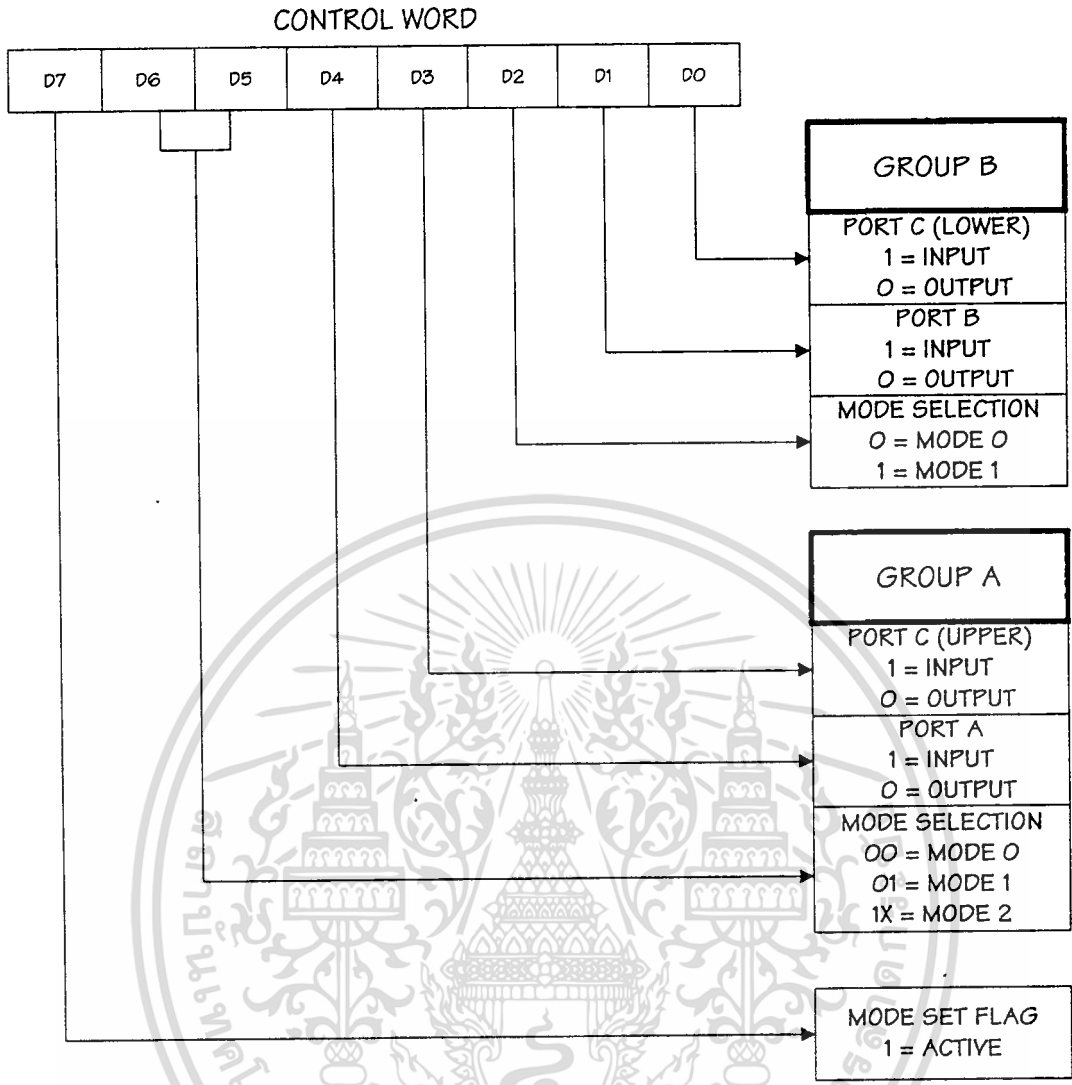
CS (Chip Select Input) : เมื่อขานี้มีสถานะลอจิกเป็น 0 CPU จะสามารถที่จะอ่านหรือเขียนข้อมูลกับ 8255 ได้

RD (Read Input) : เมื่อขานี้มีสถานะลอจิกเป็น 0 และสัญญาณ CS มีลอจิกเป็น 0 ข้อมูลของ 8255 จะ ปรากฏสู่ระบบบัสข้อมูล, CPU ก็จะสามารถอ่านข้อมูลออกไปได้ (ในการตั้งชื่อของขาสัญญาณนี้ จะถือเอา CPU เป็นหลัก)

WR (Write Input) : เมื่อขานี้มีสถานะลอจิกเป็น 0 และขาสัญญาณ CS มีลอจิกเป็น 0 ข้อมูลจากระบบบัส ข้อมูลจะถูกเขียนเข้าไปยัง 8255 ได้

เอกสารนี้เป็นเอา AO-A1 (Address Input) : จะเป็นตัวกำหนดการเลือกใช้รีจิสเตอร์ภายใน 8255 ซึ่งจะกล่าวถึงภายหลัง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 38 รูปแสดงรายละเอียดแต่ละบิตของรีจิสเตอร์ควบคุมของ 8255

โหมด 0: Basic Register I/O

ในการใช้ 8255 ให้อยู่ในโหมด 0 นั้นเราต้องส่งคำสั่งควบคุม(Control Word) ให้แก่รีจิสเตอร์ควบคุมเสียก่อน คำสั่งควบคุมนี้จะกำหนดลักษณะการทำงานของให้แก่แต่ละพอร์ตของ 8255 ตัวอย่างเช่น

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

จากรูปจะเห็นว่า

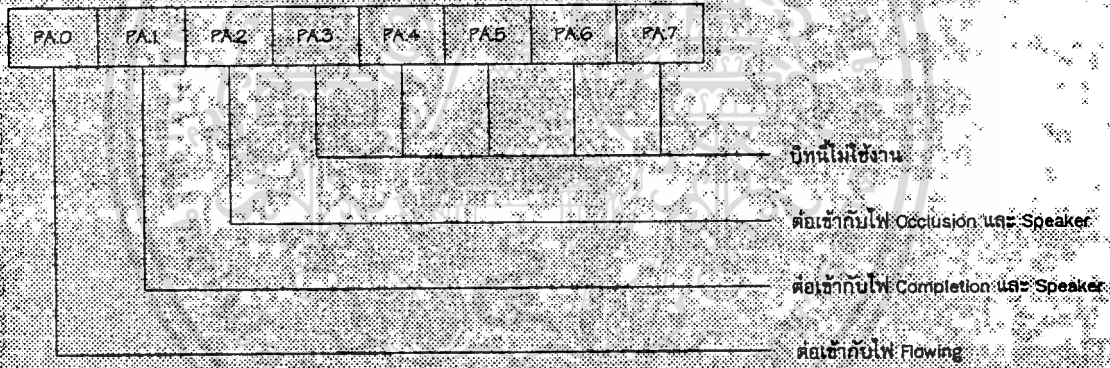
- บิต D7 เป็นตัวกำหนดว่าเป็นคำสั่งควบคุม (Control Word)
- บิต D6 และ D5 กำหนดโหมดการทำงานของพอร์ต A D6,D5 มีค่าเป็น 0 แสดงว่าอยู่ในโหมด 0
- บิต D4 = 0 กำหนดให้พอร์ต A เป็นพอร์ตเอาต์พุต
- บิต D3 = 0 เซ็ตพอร์ต C 4 บิตบนเป็นพอร์ตเอาต์พุต

- บิต D2 = 0: เซ็ตโหมดของพอร์ต B ให้พอร์ต B อยู่ในโหมด 0
- บิต D1 = 0: เซ็ตพอร์ต B เป็นพอร์ตเอาต์พุต
- บิต D0 = 0: เซ็ตพอร์ต C ให้ 4 บิตล่างเป็นพอร์ตเอาต์พุต

ในเครื่องควบคุมการให้สารละลายนี้ใช้การทำงานในโหมด 0 ของ 8255 โดยส่งคำสั่งควบคุมออกไปคำสั่งควบคุมนี้จะกำหนดให้พอร์ตทั้ง 3 ของ 8255 ทำงานอยู่ในโหมด 0 และเป็นพอร์ตเอาต์พุตซึ่งจะได้สายสัญญาณซึ่งสามารถติดต่อกับอุปกรณ์ภายนอกได้ถึง 24 สาย คำสั่งของ CPU ที่จะเซ็ทให้ 8255 อยู่ในลักษณะดังกล่าวได้แก่

```
LD A, 80H          ; เซ็ตคำสั่งควบคุม
OUT (13H), A      ; ส่งคำสั่งควบคุมให้ 8255
```

ซึ่งแอดเดรส 13H เป็นแอดเดรสที่สมมติขึ้นเท่านั้นและเลือกใช้พอร์ต A 3 พอร์ต คือ PA.0, PA.1 และ PA.2 โดยให้พอร์ต PA.0 ต่อเข้ากับไฟแสดงการไหล (Flowing) ถ้าเครื่องยังทำงานเป็นปกติขุมมีการให้สารละลายกับผู้ป่วยเป็นปกติไฟ Flowing จะติด ให้พอร์ต PA.1 ต่อเข้ากับไฟที่แสดงว่าเครื่องทำงานเสร็จสมบูรณ์แล้ว ถ้าทำงานเสร็จเรียบร้อยแล้วไฟดวงนี้จะติดพร้อมก็มีเสียงร้องบอก ส่วนพอร์ต PA.2 จะเข้ากับไฟที่แสดงว่าเกิดการอุดตัน (Occlusion) สารละลายไม่ไหลเป็นเวลานานเกินกว่าที่ตั้งไว้ ซึ่งอาจเป็นอันตรายกับผู้ป่วยได้ ไฟดวงนี้จะติดและมีเสียงสัญญาณเตือนร้องดังขึ้น โดยเราเซ็ทให้พอร์ต A ทั้ง 3 นี้เป็นพอร์ตเอาต์พุตดังรูป



รูปที่ 39 รูปแสดงการเซ็ท Port ของ 8255

การทำงานในโหมด 1

การทำงานของ 8255 ในโหมด 1 นี้เป็นการทำงานในลักษณะของการ HANDSHAKE พอร์ต A และ พอร์ต B จะเป็นพอร์ตข้อมูล ส่วนพอร์ต C นี้จะถูกใช้เป็นที่สัญญาณ HANDSHAKE โดย 4 บิตจะเป็นสัญญาณ HANDSHAKE ให้กับพอร์ต A และ 4 บิตล่างจะเป็นสัญญาณ HANDSHAKE ให้กับพอร์ต B

หลักการรับ-ส่งข้อมูลในวิธีการของ HANDSHAKE นี้คือ การให้อุปกรณ์ภายนอกส่งสัญญาณแสดงสถานะความพร้อมให้กับ 8255 วิธีการทำ HANDSHAKE มีประโยชน์มากในกรณีที่อุปกรณ์ภายนอกทำงานช้ากว่าระบบไมโครโปรเซสเซอร์ ด้วยวิธีการนี้ CPU สามารถที่จะส่งข้อมูลให้กับ 8255 แล้วไปทำงานอื่นได้ จนกว่าข้อมูลภายใน 8255 ถูกส่งออกไปแล้ว CPU จึงส่งข้อมูลใหม่ออกไปให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในโหมด 2

โหมดนี้จะเป็นการใช้งานในลักษณะที่ให้พอร์ท A เป็นพอร์ทข้อมูลแบบ 2 ทิศทาง คือใช้พอร์ท A เป็นอินพุตและเอาต์พุตแลช (เอาต์พุตแลช (Output Latch) หมายถึง การเก็บข้อมูลเอาไว้เพื่อรออุปกรณ์ภายนอกมารับเอาข้อมูลออกไป ส่วนอินพุตแลช (Input Latch) หมายถึงการเก็บข้อมูลที่อุปกรณ์ภายนอกส่งเข้ามาเพื่อรอให้ CPU อ่านเข้าไป)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

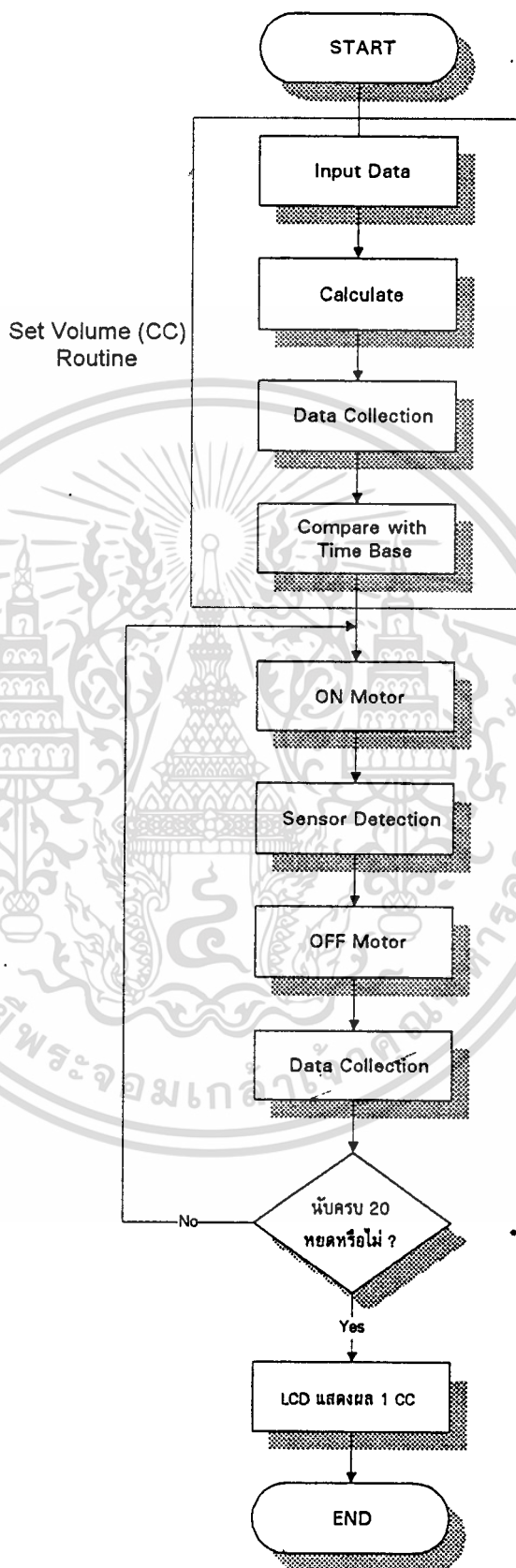
โปรแกรมการควบคุมการทำงานโดยใช้ Microcontroller

5.1 แนวความคิดในการออกแบบโปรแกรมควบคุมการทำงาน

ก่อนที่เราจะศึกษาถึง รายละเอียดการเขียนโปรแกรมควบคุมการทำงานของแต่ละส่วน ควรเข้าใจหลักการสำคัญของเครื่องก่อนคือ เมื่อผู้ใช้ได้ Set จำนวนซีซีที่ต้องการเรียบร้อยแล้ว Motor จะ On เพื่อให้ชุดรีดน้ำเคลื่อนทำงาน ที่ถาวรรับ Input ที่เข้ามาจาก Sensor และทำการประมวลผล แต่ถ้าตรวจจับได้ว่ามีหยดน้ำผ่านเข้ามาภายในเวลาที่กำหนด ก็จะไป Off Motor แล้วเก็บข้อมูลที่เป็น Data ของจำนวนหยดไว้ ถ้านับได้ครบ 15 หยดเมื่อไรก็จะไปแสดงผลบน LCD ว่าเป็น 1 CC และเพิ่มจำนวนซีซีขึ้นเรื่อยๆถ้าครบ 15 หยดครั้งต่อไป ซึ่ง Flow chart แสดงการทำงานแสดงได้ดังนี้



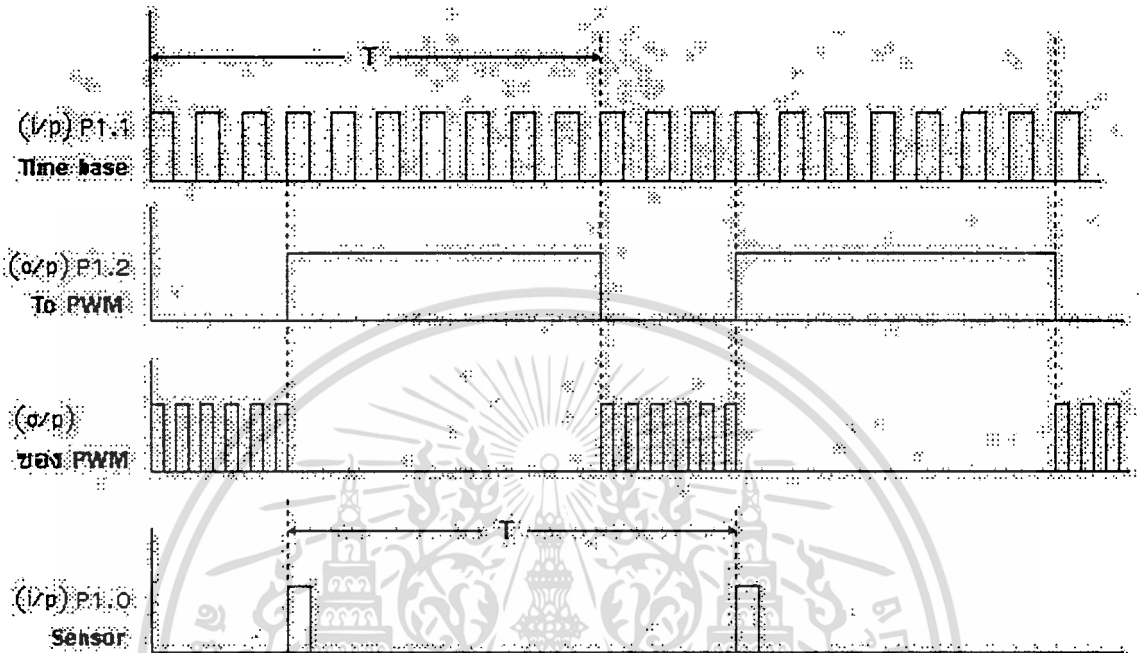
5.2 Flowchart แสดงการทำงานของเครื่องควบคุมการให้สารละลาย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวคิดเกี่ยวกับการทำงานของเครื่องที่นำมาใช้พัฒนาโปรแกรม

5.3.1 แนวคิดในการออกแบบโปรแกรมหลัก

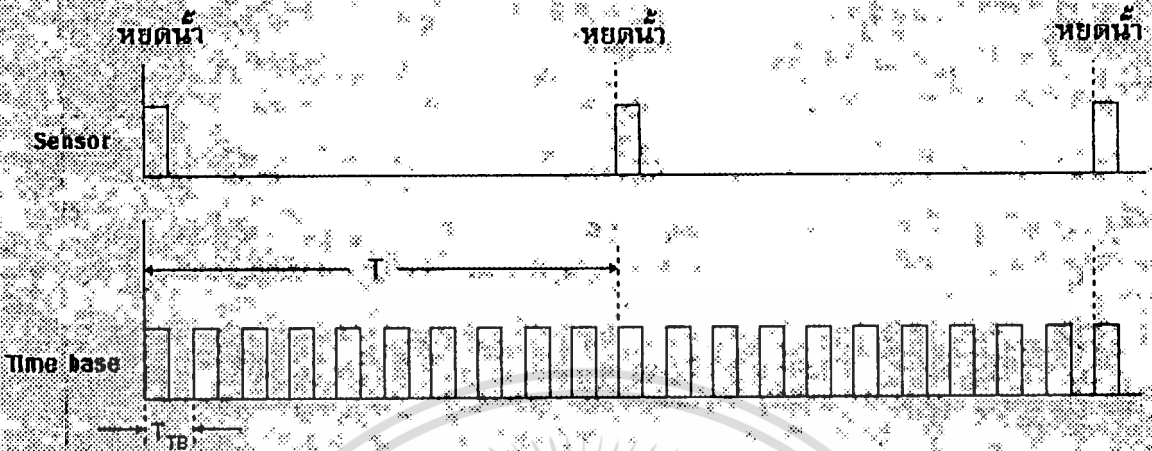


รูปที่ 40 แนวคิดเกี่ยวกับการทำงานของเครื่องที่นำมาใช้พัฒนาโปรแกรม

ในการทำงาน CPU จะใช้เวลาในการประมวลผลเร็วมากดังนั้นเราจึงใช้หลักการแบ่งเวลาเพื่อให้ CPU วนเช็คพอร์ทแต่ละพอร์ทเป็น Loop ตอนแรกเราส่งให้ Time Base เริ่มนับแล้วส่งให้ P1.2 ซึ่งต่อกับ Pulse Width Modulation มีสถานะเป็น Low เพื่อให้มอเตอร์หมุนขยับหน้า แล้วให้ CPU วนไปเช็คที่พอร์ท P1.0 ซึ่งรับอินพุตเข้ามาจาก SENSOR ว่ามีสถานะเป็น High หรือยัง (คือตรวจจับหยุดหน้าได้) ถ้ายังไม่หยุดหน้า ก็ให้วนกลับไปเช็คพอร์ท P1.1 วนนับจำนวน Pulse ได้เท่ากับ $24000 \times X$ หรือไม่ ซึ่งค่านี้เป็นค่าที่ CPU กำหนดได้เป็นค่าระยะห่างระหว่างหยุดหน้าแต่ละหยุด (T) ส่วน X เป็นอัตราค่าที่ผู้ใช้ตั้งเข้ามา ถ้ายังไม่หยุดหน้า Pulse ไม่ได้เท่ากับค่านี้ก็ให้วนเช็คแบบนี้ไปเรื่อยๆ เพื่อเช็คสถานะของแต่ละพอร์ท เมื่อมีหยุดหน้าเข้ามา SENSOR จะตรวจจับได้ทำให้พอร์ท P1.0 เป็น High CPU ก็จะทำการ Load ค่า 1 ไปที่พอร์ท P1.2 ซึ่งจะทำการหยุดมอเตอร์ และพอร์ท P1.1 จะนับ Time Base ไปเรื่อยๆ เมื่อครบ $24000 \times X$ พัลส์ก็จะสั่งให้พอร์ท P1.2 เป็น Low อีกครั้งหนึ่งเพื่อให้มอเตอร์หมุนขยับหน้า จะเป็นเช่นนี้ไปเรื่อยๆ จนปริมาณครบตามที่ผู้ใช้ตั้งไว้ แต่ถ้าพอร์ท P1.1 นับ Time Base จนครบ $24000 \times X$ แล้ว แต่ยังไม่มีส่วนไหนที่เป็น High จาก SENSOR เข้ามา (พอร์ท P1.2 ยังคงสถานะ Low อยู่) CPU ก็จะส่งสัญญาณทำให้เกิดการอุดตัน (Occlusion) เพื่อให้ผู้ใช้แก้ไขต่อไป

สำหรับเครื่องควบคุมสารละลายเครื่องนี้ เราได้เลือกใช้เซ็นเซอร์น้ำเกลือที่เป็นแบบ 15 หยดต่อ 1 CC นำมาติดตั้งและใช้ปริมาณนี้ในการเขียนโปรแกรมคำนวณ สำหรับ Flow Chart แสดงการทำงานของเครื่องแสดงได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 รูปแสดงคาบเวลาของหยดน้ำเมื่อเทียบกับ Time base

$$\begin{aligned}
 \text{จำนวน Pulse} &= \frac{3600 \left(\frac{\text{Sec.}}{\text{Drop}} \right)}{15X \left(\frac{\text{Drop}}{\text{Pulse}} \right)} \\
 &= \frac{1 \left(\frac{\text{Sec.}}{\text{Pulse}} \right)}{100 \left(\frac{\text{Pulse}}{\text{Drop}} \right)} \\
 &= \frac{3600 * 100 \left(\frac{\text{Pulse}}{\text{Drop}} \right)}{15X} \\
 T &= \frac{24000 \left(\frac{\text{Pulse}}{\text{Drop}} \right)}{X}
 \end{aligned}$$

แต่ในโปรแกรม DEVIDE เราจะให้ $T = \frac{24000}{X} = \left(\frac{250}{X} \right) (96)$ เพราะต้องใช้ การหาร 8 บิต (คือค่าตั้งแต่ 0-255) ก็คือเราจะทำการหาร 250 ด้วย X ก่อน (เพราะ 250 แปลงเป็นเลขฐาน 2 แล้วไม่เกิน 8 บิต) สำหรับค่า X ผู้ใช้สามารถตั้งได้ตั้งแต่ค่า 0-200 ml/hr และเมื่อได้ผลการแล้วจึงนำไปคูณกับ 96 คือเป็นการคูณ หาร ทีละ 8 บิตนั่นเอง แล้วเก็บผลลัพธ์ที่หารได้ Byte สูงไว้ในหน่วยความจำชื่อ TB High และเก็บผลการ Byte ต่ำไว้ที่ TB Low

ตัวอย่างการหารโดยใช้โปรแกรม Device

ถ้าผู้ใช้ป้อนค่า X = 11 ml/hr โปรแกรมนี้จะทำงานโดยมีลำดับการทำงานดังต่อไปนี้

$$\frac{24000}{X} = \left(\frac{250}{11} \right) (96)$$

โดยที่ $\frac{250}{11} = 22 \text{ เศษ } 8$

เอกสารนี้เป็นเอกสารจากนั้นจึงนำ $22 * 96 = 2112$ ศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ (1) การการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0000 จริงๆแล้วการนับค่า CC หรือนับค่า Time Base เหมือนกับนาฬิกาจับเวลาแบบถอยหลัง มีการคำนวณค่า Time Base และปริมาตรเป็น CC เก็บไว้ในรีจิสเตอร์ เราใช้ค่า Time Base เพื่อให้ CPU ใช้ในการเปรียบเทียบ นับจำนวนพัลส์ให้ตามที่คำนวณ มอเตอร์จะถูกสั่งให้หมุนและหยุดหมุนโดยได้รับคำสั่งจาก CPU อ่านค่าพัลส์ที่นับได้ เช่น ถ้า CPU คำนวณได้ว่าหยุดน้ำแต่ละหยุดห่างกัน 10 พัลส์ พอ CPU นับพัลส์ของ Time Base ได้ 1 ก็จะไปลดค่าในรีจิสเตอร์ที่เก็บค่า 10 ลง 1 รีจิสเตอร์นั้นก็เก็บค่า 9 ไว้แทน เมื่อค่าในรีจิสเตอร์นี้ลดลงจนเหลือ 0 CPU ก็จะสั่ง Drive ครึ่งหนึ่งพร้อมกับ Load ค่าที่คำนวณได้ไปเก็บไว้ในรีจิสเตอร์ตามเดิม คือรีจิสเตอร์นั้นก็เก็บค่า 10 ไว้อีก เป็นเช่นนี้ไปเรื่อยๆ ส่วนปริมาตรก็เช่นกัน เมื่อมีพัลส์ จาก SENSOR เข้ามาจนครบ 15 หยุด CPU ก็จะไปลดค่าในรีจิสเตอร์ที่เก็บค่าปริมาตรที่ใช้ตั้งไว้ลง 1 ก็จะลดค่าในรีจิสเตอร์นี้ลง 1 ทุกครั้งที่มีพัลส์จาก SENSOR เข้ามาครบ 15 พัลส์ (15 หยุด) เมื่อค่าในรีจิสเตอร์ลดลงจนเหลือ 0 ก็จะแสดงผลบอกผู้ใช้มีไฟที่แสดงว่าการทำงานสมบูรณ์ดีดี และมีเสียงร้องบอกผู้ใช้

5.3.2 แนวคิดในการออกแบบโปรแกรมการหาร

เครื่องควบคุมสารละลายนี้ ความสำคัญอยู่ที่โปรแกรม Delay เพราะเราต้องการให้มีการหน่วงเวลาให้ได้หลายค่าตามอัตราการ SET RATE ตามที่ผู้ใช้ต้องการ ซึ่ง RATE ที่ใช้มีตั้งแต่ 1-200 ml/hr เราใช้หลักการในการกำหนดค่าความถี่ขึ้นมาค่าหนึ่ง ซึ่งเราใช้เป็นฐานเวลา (Time Base) แล้วผผันโปรแกรมคำนวณค่าออกมาเป็นจำนวน Pulse ของความถี่ที่สัมพันธ์กับ RATE นั้นๆ ซึ่งจำนวนพัลส์ที่เป็นค่า Delay คำนวณได้โดย

เราเลือกใช้ Time Base ที่มีความถี่ 100Hz

$$T_{TB} = \frac{1}{100} \left(\frac{\text{Sec}}{\text{Pulse}} \right)$$

ถ้าให้ Rate ที่ Set = X ml/hr

เลือกใช้ชุดสายน้ำเกลือแบบ 15 หยุด/ml

เราต้องการให้อยู่ในหน่วย Sec./หยุด เราจะแปลงหน่วยโดย

$$\begin{aligned} & \left(X \frac{\text{ml}}{\text{hr}} \right) \left(15 \frac{\text{Drop}}{\text{ml}} \right) \left(\frac{1}{3600} \frac{\text{hr}}{\text{Sec}} \right) \\ &= \frac{15X}{3600} \left(\frac{\text{Drop}}{\text{Sec}} \right) \\ &= \frac{3600}{15X} \left(\frac{\text{Sec}}{\text{Drop}} \right) \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{8 * 96}{11} = \frac{768}{11} = 69 \text{ เศษ } 9 \quad \text{ซึ่งผลลัพธ์ที่ได้จะปัดขึ้นเป็น } 70 \dots(2)$$

เนื่องจากเราตั้งว่าถ้าเศษที่ได้เกิน $\frac{X}{2}$ ให้ปัดขึ้นซึ่ง 9 นั้นมากกว่า $11/2$ มากจึงปัดขึ้นอีก 1 ผลลัพธ์จึงกลายเป็น 70 หลังจากนั้นจึงนำ (1) กับ (2) มารวมกัน

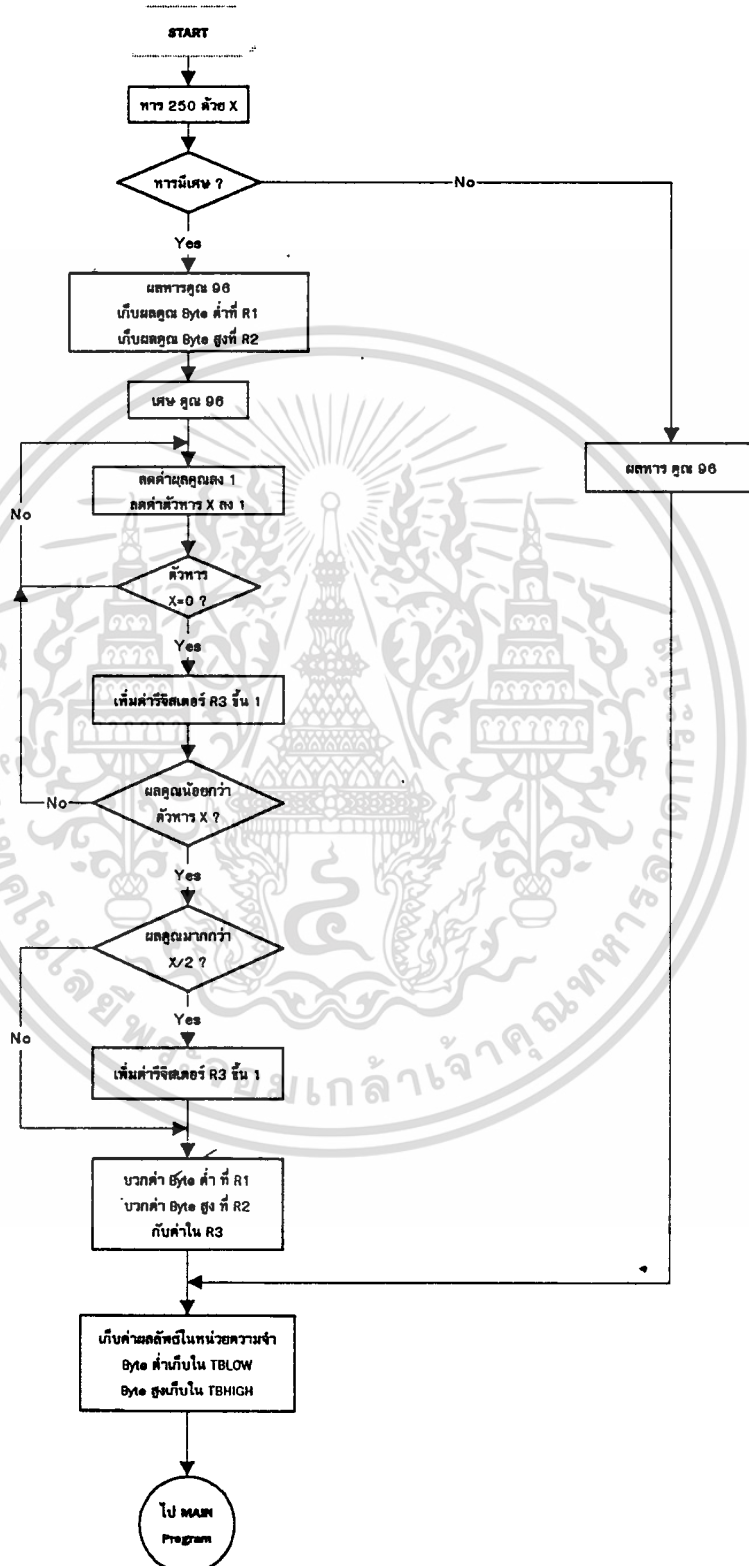
$$2112 + 70 = 2182$$

แล้วจึงนำผลลัพธ์นี้ไปเก็บที่ได้ไว้ใน TB High และ TB Low ตามลำดับ จะเห็นว่าวิธีที่ใช้คิดคำนวณนั้นยุ่งยากพอสมควร แต่ที่ต้องทำเพื่อใช้รีจิสเตอร์ 8 บิตที่มีทำการคำนวณให้ได้ผลลัพธ์ถูกต้องตามที่ต้องการ สำหรับ Flow Chart แสดงการทำงานของโปรแกรมการหารแสดงได้ดังต่อไปนี้



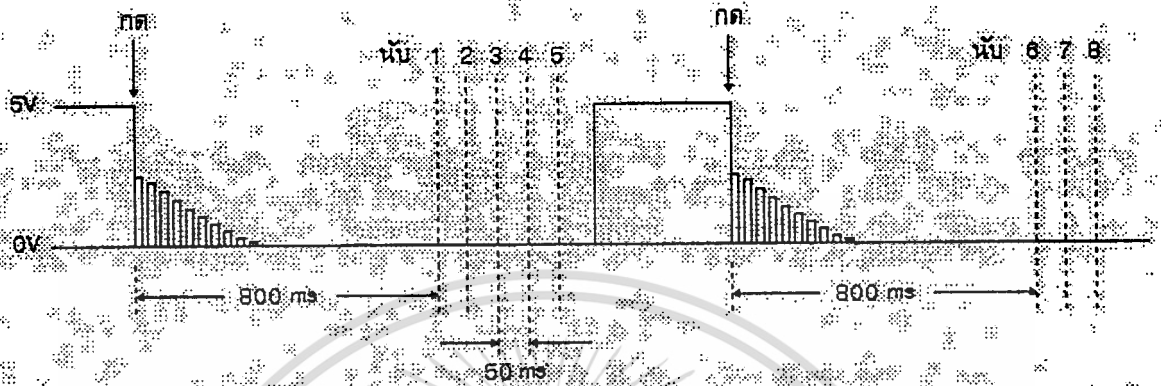
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOWCHART โปรแกรม DEVIDE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

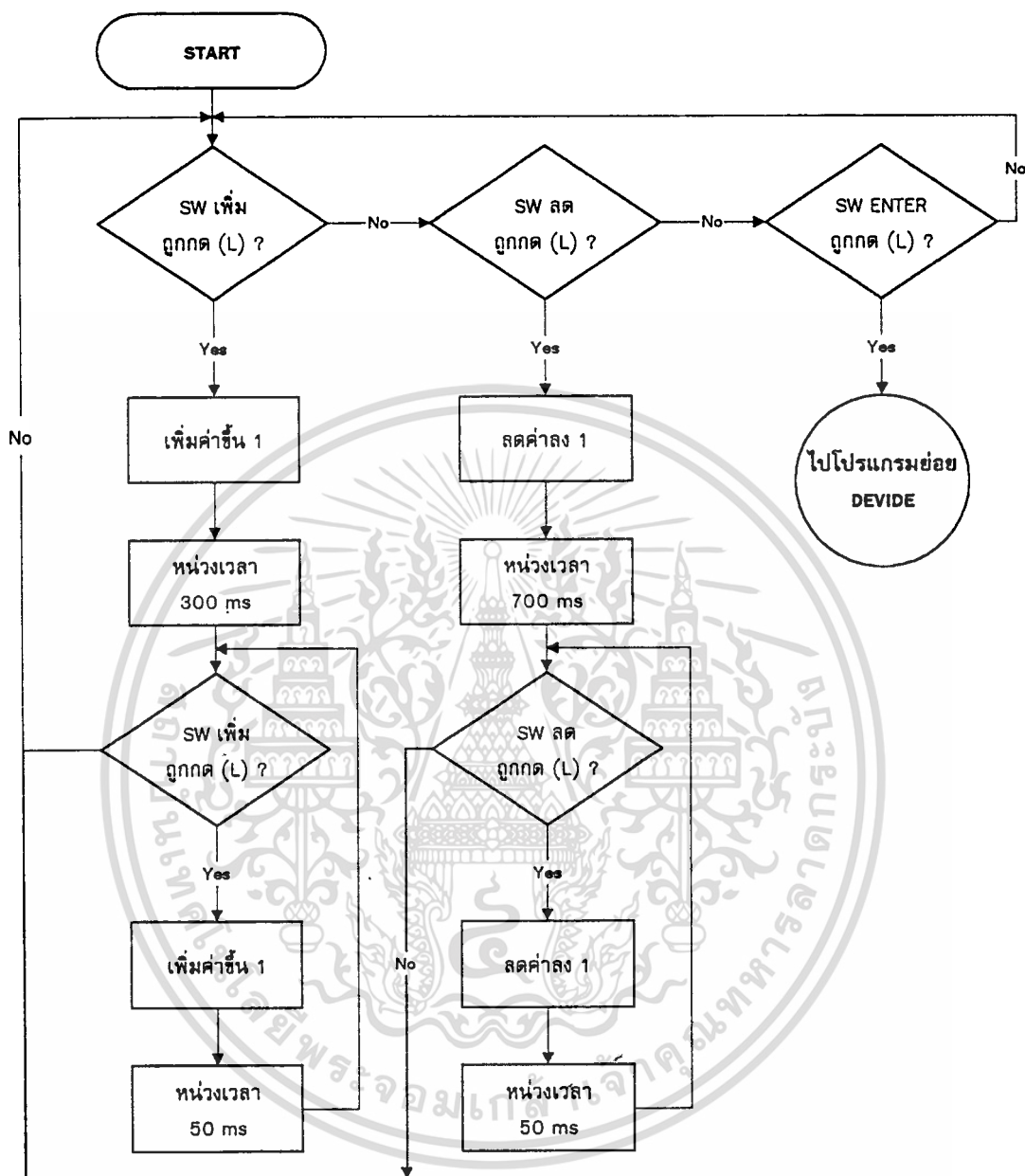
5.3.3 แนวคิดในการออกแบบโปรแกรมการกด Switch เพื่อเพิ่มค่าหรือลดค่าของปริมาตรและอัตราการไหล (Volume & Rate)



รูปที่ 42 รูปแนวคิดในการออกแบบโปรแกรมการกด Switch เพื่อเพิ่มค่าหรือลดค่าของปริมาตรและอัตราการไหล (Volume & Rate)

ปกติสถานะของ Switch จะถูกต่อกับตัวต้านทาน R-Pull Up ไว้ จะมีสถานะตอนปกติเป็น High และสถานะตอนถูกกดเป็น Low ซึ่งโปรแกรมจะทำการ Scan Switch ทุกตัวไปตามลำดับเรื่อยๆ โดยเช็คว่าผู้ใช้กด Switch เพิ่ม (Increase Switch) หรือกด Switch ลด (Decrease Switch) หรือกด Switch Enter เพื่อให้เครื่องเริ่มการทำงาน ถ้า Switch ใดถูกกดก็จะแยกไปทำงานตาม Block ที่แสดงไว้ ถ้าสมมติว่า Switch เพิ่มถูกกด (กดแล้วเป็น Low) ก็จะเพิ่มค่าการนับขึ้น 1 และหน่วงเวลาประมาณ 800 ms และถ้า Switch ลดนี้ยังถูกกดอยู่ก็จะเพิ่มค่าขึ้น 1 อีกทุกๆ 50 ms แต่ถ้ากดแล้วปล่อย ภายในเวลา 800 ms ก็จะเป็นการเพิ่มค่าขึ้นทีละ 1 อย่างช้าๆ ตามอัตราการกด ซึ่ง Switch ลดก็มีการทำงานเช่นเดียวกันดังแสดงใน Flow Chart การทำงานรวม

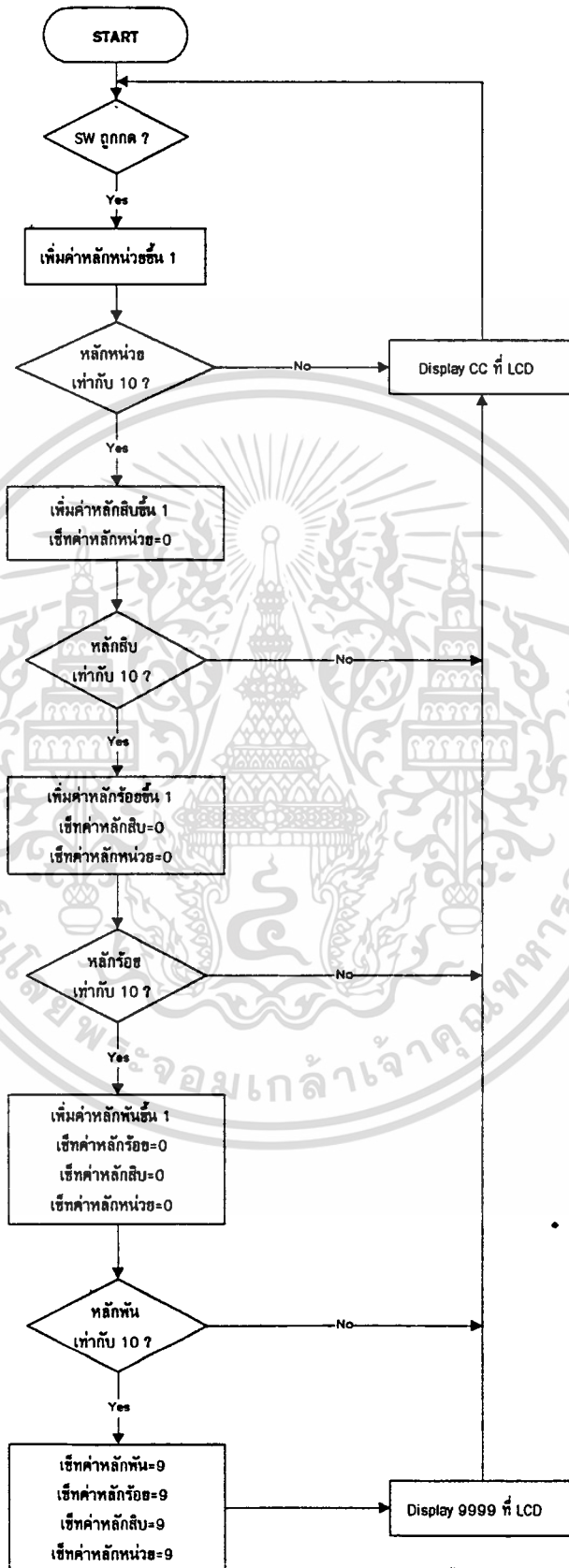
FLOWCHART โปรแกรม INC/DEC



Flow Chart ที่แสดงไปข้างต้นเป็น Flow Chart แสดงหลักการทำงานของทั้ง 3 Switch สำหรับแต่ละ Switch จะถูกตั้งไว้ให้สามารถเพิ่มหรือลดค่าไว้เป็นค่าใดค่าหนึ่ง อย่างเช่นเวลาที่เราจะทำการตั้งปริมาตร (Set Volume) เราสามารถเซ็ค่าได้ตั้งแต่ 0-9999 ml จะเห็นว่าเป็นเลข 4 หลัก การเพิ่มค่าเราต้องเขียนโปรแกรมให้ทำการเพิ่มค่าทีละหลัก โดยเริ่มจากหลักหน่วยก่อน แต่เนื่องจากรีจิสเตอร์ที่เราสามารถใช้ได้ใน ET-Board เป็นรีจิสเตอร์ 8 บิตเมื่อเพิ่มค่าจาก 0 ไปถึง 255 ก็จะเต็ม ดังนั้นหลักการเขียนโปรแกรมจะสามารถแสดงได้ดัง Flow Chart ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOWCHART โปรแกรมการเพิ่มค่า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเพิ่มปริมาตร (Increase Volume) จะใช้หลักการเดียวกับการเพิ่มอัตราการไหล (Increase Rate of Flowing) แต่อัตราการไหลจะขีดได้แค่ 0-200 ml/hr เท่านั้นแต่การขีดปริมาตรสามารถขีดได้ตั้งแต่ 0-9999 ml ส่วนหลักการลดปริมาตรก็ใช้หลักการเดียวกันแต่เป็นการลดค่าลงมา

5.4 การขีด Display LCD เพื่อแสดงผล

ในเครื่องนี้เราจะขีด LCD เพื่อให้แสดงผลเราใช้ LCD 2 บรรทัดเพื่อแสดงผล โดยตั้งให้ Cursor ที่อยู่บนจอ LCD ไม่กระพริบ แสดงผลแล้ว DRAM ที่ค่าขึ้นอีก 1 ดังรูป

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1-line	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
2-line	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	51	53

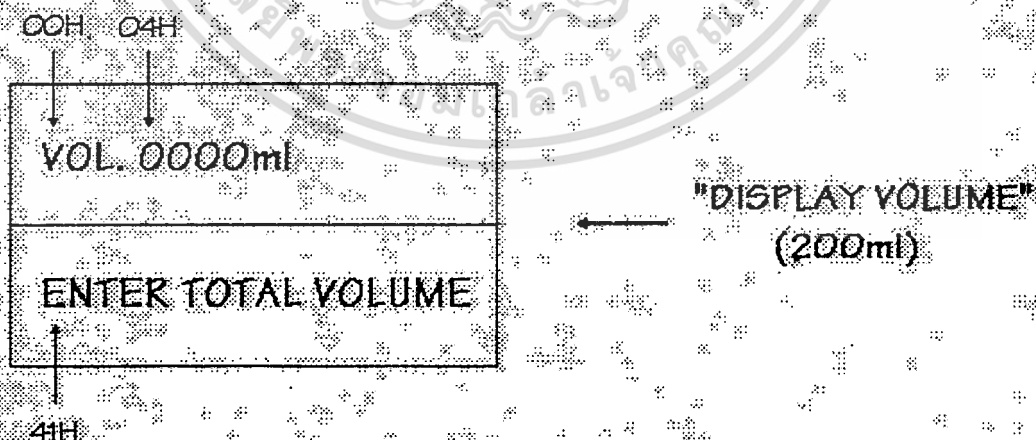
ตารางที่ 4 ตารางแสดงตำแหน่งต่าง ๆ บนจอ LCD 2 บรรทัด

จากตารางข้างบนเป็นตำแหน่งต่าง ๆ บนจอ LCD 2 บรรทัดที่เราใช้ เป็นตำแหน่งของ DRAM ถ้าเราต้องการให้มีตัวอักษรใดไปปรากฏบนจอ LCD ก็ไปเปิดตาราง Characteristic ว่าตัวอักษรหรือตัวเลขนั้นตรงกับ Code อะไรแล้วไปเลือกตำแหน่งที่ต้องการบนจอ LCD

เราจะตั้งให้สามารถ Display Volume ได้ตั้งแต่ 0-9999

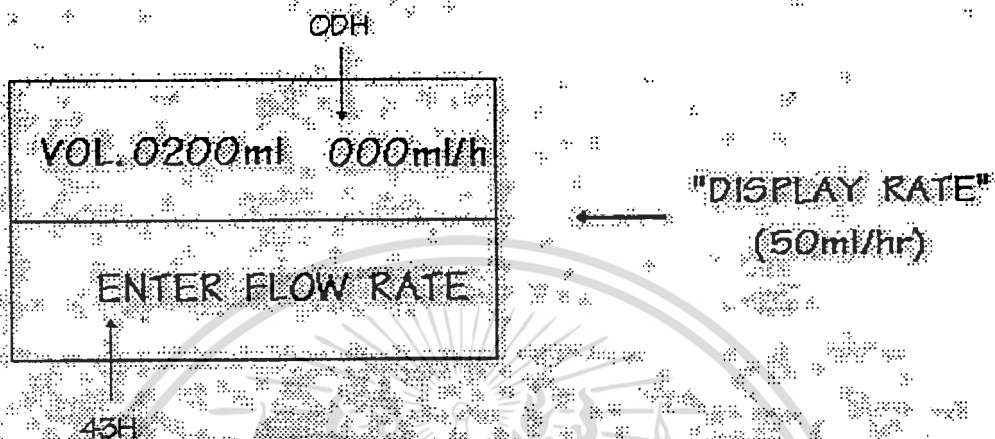
Display Rate of Flowing ได้ตั้งแต่ 0-200

ในตอนแรกที่เปิดเครื่องบนจอ LCD จะแสดง Display ดังรูปต่อไปนี้



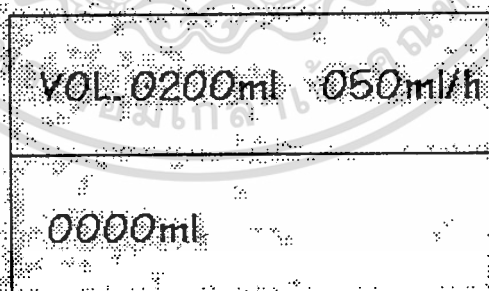
รูปที่ 43 แสดง DISPLAY บนจอ LCD ขณะ SET VOLUME

ผู้ใช้จะต้องใส่ค่าตัวเลขหรือปริมาณที่ต้องการลงไปโดยใช้ปุ่ม Increase Switch หรือ Decrease Switch เมื่อได้ปริมาณตามที่ต้องการแล้วให้กดปุ่ม Enter แล้ว Display บนจอ LCD จะเปลี่ยนไปเป็นดังรูป



รูปที่ 44 แสดง DISPLAY บนจอ LCD ขณะ SET RATE

สมมติว่าผู้ใช้ตั้งปริมาตรไว้ที่ 100 ml เมื่อกดปุ่ม Enter แล้ว ผู้ใช้ต้องใส่ค่าอัตราการไหลที่ต้องการลงไปโดยใช้ปุ่ม Increase Switch หรือ Decrease Switch เช่นเดิมแล้วกดปุ่ม Enter อีกครั้งหนึ่งเครื่องก็จะแสดง Display ดังรูป และเครื่องก็จะเริ่มทำงานเมื่อ SENSOR ตรวจจับหยดน้ำได้ครบ 15 หยด ซึ่งแต่ละหยด CPU ก็จะนำไปทำการประมวลผล พอครบ 15 หยดค่าของปริมาตรบนจอ LCD บรรทัดที่ 2 ก็จะเพิ่มขึ้น 1



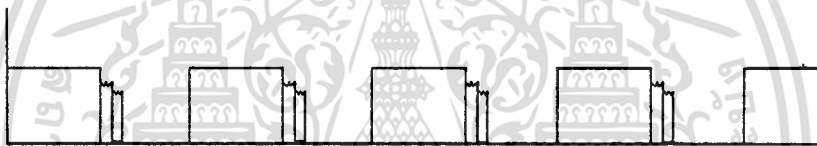
รูปที่ 45 แสดง DISPLAY บนจอ LCD หลังจากที่ทำกา SET เรียบร้อยแล้ว

บทที่ 6

การทดลอง

6.1 การทดลองที่ 1 การตรวจสอบความถูกต้องของวงจร SENSOR

ในช่วงแรกของการทดลอง เครื่องควบคุมการให้สารละลาย ได้ทำการทดลองในส่วนของ Sensor ก่อน โดยพยายามใช้ Sensor ตรวจจับหยดน้ำให้ได้ โดยใช้ลองวงจรรีโมทคอนโทรล ซึ่งมี LED เป็นตัวรับและตัวส่ง มาวางไว้ระหว่างกระเปาะสายน้ำเกลือ ป้อนไฟเลี้ยงเข้าไปเพื่อให้ภาครับและภาคส่งทำงาน ถ้าไม่มีหยดน้ำหยดลงมาระหว่าง LED ตัวรับและตัวส่ง LED จะรับ-ส่งสัญญาณกันตลอด แต่ถ้ามีหยดน้ำหยดลงมาก็เหมือนมีอะไรมาบังสัญญาณระหว่าง LED ทำให้รับ-ส่งกันไม่ได้ เมื่อลองใช้ออสซิลโลสโคปตรวจจับดู ปรากฏว่ามีสัญญาณ Pulse ที่แสดงให้เห็นว่า LED สามารถตรวจจับหยดน้ำได้จริง แต่ยังมี Noise ปนอยู่บ้าง จึงต่อตัวเก็บประจุ (Capaciter) เพิ่มเข้าไป เพื่อกรองเอาสัญญาณรบกวนออก แต่ก็ยังมีปัญหาเมื่อทดลองปล่อยให้น้ำหยดไปเรื่อยๆ จะปรากฏว่ามี Pulse เดี่ยวๆ เข้ามาแทรกซ้อนกับ Pulse หลัก ซึ่งอาจทำให้เกิดความผิดพลาดขึ้นได้ ถ้าเราใช้ Counter นับ Pulse จึงแก้ปัญหาโดยใช้ IC SCL 4528 เพื่อสร้าง Pulse ในช่วงเวลาที่ต้องการ โดย SCL 4528 นี้จะทำงานแบบ Non retrigable คือจะ Trig ที่สัญญาณ Pulse ขาขึ้น ซึ่งถ้ามี Pulse เล็กๆ ซึ่งเกิดจาก Noise ก็จะไม่ Trig อีก



รูปที่ 46 รูปแสดงความผิดปกติของ Pulse ที่ตรวจจับได้

นอกจากนั้นแล้ววงจร Sensor ที่เราเลือกใช้ Output ที่ได้ขณะที่มีหยดน้ำผ่านเป็นค่าลบเราต้องการทำให้ Output ที่ได้มีค่าบวก จึงนำสัญญาณที่เราต้องการมากลับเฟส โดยเลือกใช้ Transister เบอร์ C 458 มาช่วยกลับเฟสสัญญาณ โดยได้อธิบายการทำงานของวงจรไปแล้วในส่วนของ Sensor คือ ถ้ามีสัญญาณที่เป็น V+ ออกมาจาก Output ขา 6 ของ IC 741 จะทำให้เกิดการ Bias Transister ทำให้ Transister ทำงานและมีกระแสไหลจากขา Collector ไปยังขา Emitter ทำให้ V ที่จุด B เป็นศูนย์ (Ground) เพราะเปรียบเสมือนการ Short Circuit แต่พอมี Output ที่เป็นลบที่ขา 6 ทำให้ไม่สามารถ Bias Transister ได้ Transister จึงไม่นำกระแส เปรียบเสมือนเป็น High Impedance คักคาที่จุด B จึงสูงให้ Output ที่จุด B เป็นบวก จึงทำให้สามารถกลับเฟสสัญญาณได้

หลังจากที่เราได้ทำการทดลองในส่วนของ Sensor แล้ว ก็หันมาทดลองในส่วนควบคุม Motor เพื่อบังคับ Motor ให้ไปขับชุดรีดน้ำเกลือ เพื่อที่จะให้สารละลายหยดได้ตามเวลาที่เรต้องการ โดยเราเลือกที่จะควบคุม Motor โดยใช้ Pulse Width Modulation (PWM) เนื่องจากเป็นหลักการที่ง่าย ช่วยลดกระแสที่จะไหลผ่าน Transister เพราะเป็นการ Switching และยังทำให้ไม่ต้องใช้ Supply ที่มีขนาดใหญ่มาก ประโยชน์ที่ได้ก็คือเราสามารถปรับความเร็วของ Motor ได้โดยที่แรงบิดในการหมุนของ Motor มีค่าคงที่และ Supply ที่ให้ Motor มีค่าคงที่ด้วย นั่นก็คือการปรับความถี่ของ Pulse นั้นเอง

สรุปและวิจารณ์ผลการทดลองตอนที่ 1

จากการทดลองเครื่องควบคุมการให้สารละลายในหลอดเลือดดำนี้ ปรากฏว่าเมื่อมีหยดน้ำผ่าน วงจร Sensor ที่เราเลือกใช้ จะเห็นว่า Sensor ที่ใช้สามารถตรวจจับหยดน้ำได้จริง โดยที่การหักเหของแสงจาก ตัวส่งไปยังตัวรับไม่มีปัญหาแต่อย่างใด เมื่อส่วนของวงจร Sensor ทำงานได้แล้ว ก็ได้หันมาทดลองในส่วนของ Motor โดยพยายามหาวิธีที่จะควบคุม Motor ได้อย่างมีประสิทธิภาพโดยเลือกใช้การควบคุม Motor แบบ Pulse Width Modulation (PWM) ในส่วนของ Pulse Width Modulation นั้น เราสามารถปรับความถี่ได้ตามต้องการ โดยการปรับค่า R Vary ทำให้การ On-Off เร็วขึ้น ทำให้การควบคุมสามารถทำได้ละเอียดขึ้น Motor จะหมุนและหยุดหมุนได้ตามที่เราต้องการ ขึ้นกับช่วงห่างของ Pulse ที่เราตั้งความถี่ไว้

6.2 การทดลองที่ 2 การตรวจสอบความถูกต้องของเครื่องว่าเป็นไปตามโปรแกรมหรือไม่

จุดประสงค์การทดลอง

เพื่อตรวจสอบดูความถูกต้อง ของการทำงานของไมโครคอนโทรลเลอร์ ว่าเป็นไปตามที่เราเขียนโปรแกรมหรือไม่

วิธีการทดลอง

1. ต่อขุอสายน้ำเกลือเข้ากับเครื่อง Infusion Pump โดยให้สายน้ำเกลือพาดเข้าร่องของ Pump ที่ติดกับ Motor
2. กด Switch เปิดเครื่อง จอ LCD จะปรากฏคำว่า “ ENTER TOTAL VOLUME ”
3. เชื่ค่า TOTAL VOLUME โดยกดปุ่ม Δ และ ∇ ตามค่าที่กำหนดไว้ในตาราง แล้วกดปุ่ม ENTER จอ LCD จะปรากฏคำว่า “ ENTER FLOW RATE ”
4. เชื่ค่า FLOW RATE โดยกดปุ่ม Δ และ ∇ เช่นเดียวกันเมื่อเชื่ค่าเรียบร้อยแล้ว กดปุ่ม ENTER
5. เครื่องจะปรากฏบนจอ LCD ว่า “FLOWING ml ”
6. จับเวลา 1 นาทีว่าของเหลวไหลไปที่ ml บันทึกค่าลงในตารางเปรียบเทียบกับค่าที่เชื่ไว้ในข้อ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกผลการทดลองที่ 2

FLOW RATE (ml/hr)	จำนวนหยดจากการ คำนวณ (หยด/นาที)	จำนวนหยดจากการทดลอง	
		ดูสัญญาณจาก Scope (หยด)	ดูจำนวนหยดจากการนับ (หยด)
40	10	10	12
60	15	15	16
80	20	20	20
100	25	25	27
120	30	30	33
140	35	35	39
160	40	40	42
180	45	45	48
200	50	50	52

การคำนวณจำนวนหยดที่ถูกต้อง

เราทราบว่า 1 millilite (ml) = 15 drop

อัตราการไหล FLOW RATE ml/hr ต้องการทำให้อยู่ในรูป drop/minute

$$\text{FLOW RATE ml/hr} = \text{FLOW RATE} \left(\frac{15 \text{ Drops}}{60 \text{ Minutes}} \right)$$

$$\begin{aligned} \therefore \text{จำนวนหยดของแต่ละ FLOW RATE ที่ตั้งไว้} &= \text{FLOW RATE} * \left(\frac{15 \text{ Drops}}{60 \text{ Minutes}} \right) \\ &= \left(\frac{\text{FlowRate}}{4} \right) \left(\frac{\text{Drops}}{\text{Minute}} \right) \end{aligned}$$

สรุปและวิจารณ์ผลการทดลองตอนที่ 2

จากผลการทดลองตอนที่ 2 เมื่อดูสัญญาณหยดน้ำจาก Sensor และสัญญาณ Drive Motor จาก PWM โดยใช้ Oscilloscope จะได้ผลการทดลองที่ตรงกับการคำนวณ แสดงว่า MCS-51 สามารถทำงานตามคำสั่งที่โปรแกรมไว้ได้อย่างถูกต้อง แต่จากการทดลองโดยการนับหยดน้ำจะมีความผิดพลาดเกิดขึ้นคือมีหยดน้ำเกินมาประมาณ 2-4 หยด เนื่องจากชุดรีดสายน้ำเกลือ ซึ่งเป็นส่วนประกอบทางด้าน Mechanic มีความบกพร่อง ในบางมุมจะกดรีดสายไม่แน่น ทำให้หยดน้ำไหลออกมา โดยแรงโน้มถ่วงของโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทดลองที่ 3 เพื่อตรวจสอบความถูกต้อง ของการทำงานของเครื่อง Infusion Pump

จุดประสงค์การทดลอง

เพื่อตรวจสอบความถูกต้อง ของการทำงานของเครื่อง Infusion Pump ที่อัตราการไหลต่างๆ
วิธีการทดลอง

1. Set ค่า Total Volume ให้เท่ากับ 250 ml
2. Set ค่า Flow Rate ที่อัตราต่างๆ ตามตาราง
3. ปลอ่ยให้หยดใส่ภาชนะตวงเป็นเวลา 30 นาที
4. ตวงของเหลวที่หยดออกมานั้นด้วยเครื่องมือตวงหน่วยเป็น ml แล้วคำนวณออกมาเป็น ml/hr
5. บันทึกค่าที่ได้ลงในตารางผลการทดลอง แล้วคำนวณหาอัตราการผิดพลาด

ตารางบันทึกผลการทดลองที่ 3

FLOW RATE (ml/hr)	ปริมาตรใน 1 ชั่วโมง (ml)	ปริมาตรที่ได้จากการวัด ใน 1 ชั่วโมง (ml)	อัตราความผิดพลาด (%)
40	40	40.6	1.5
60	60	58.8	2.0
80	80	80.8	1.0
100	100	105.0	3.5
120	120	115.2	4.0
140	140	136.0	2.86
160	160	155.0	3.1
180	180	173.3	3.7
200	200	195.0	2.5

สรุปและวิจารณ์ผลการทดลองตอนที่ 3

จะเห็นว่า เครื่องควบคุมการให้สารละลายโดยใช้ไมโครคอนโทรลเลอร์นี้ มีความผิดพลาดอยู่เหมือนกัน สาเหตุประการแรกอาจเนื่องมาจากจำนวนครั้งที่เราทำการทดลองอาจน้อยเกินไป จึงยังไม่สามารถวัดได้อย่างละเอียดมากนัก รวมทั้งอุปกรณ์การวัดที่ใช้ยังขาดความแม่นยำพอ ไม่สามารถวัดอย่างละเอียดได้เครื่องจึงเกิดการผิดพลาดบ้าง แต่ในทางการแพทย์แล้วการผิดพลาดที่น้อยกว่า 3 เปอร์เซ็นต์ถือว่ายอมรับได้ แต่ยังคงต้องปรับปรุงให้เครื่องมีการทำงานที่ถูกต้องแม่นยำมากกว่านี้

สรุปและวิจารณ์ผลการทดลองเครื่องควบคุมการให้สารละลายในเส้นเลือดดำโดยใช้ไมโครคอนโทรลเลอร์

จากการทดลองจะเห็นว่าเครื่องสามารถทำงานได้ ในระดับหนึ่งถึงแม้ว่าจะยังไม่ถูกต้องทั้งหมดแต่ก็พอจะเห็นได้ว่า ถ้าผู้ใช้เซ็ทปริมาตรค่าน้อยๆ จะเกิดความผิดพลาดน้อยมาก แต่ถ้าเซ็ทค่าปริมาตรมากจะเกิดความผิดพลาดเพิ่มขึ้น ซึ่งความผิดพลาดของเครื่อง อาจเกิดจากเซ็ทสายน้ำเกลือหรือส่วนประกอบของเครื่องทางด้าน Mechanic รวมทั้งอุปกรณ์ที่ใช้ทำการวัดปริมาตรที่ต้องการความละเอียดมากกว่านี้ ซึ่งคงต้องทำการตรวจสอบและแก้ไขต่อไป แต่การทำงานของเครื่องโดยรวมจะทำให้ผู้ใช้เกิดความสะดวกมากขึ้น ซึ่งโดยปกติแล้ว การให้น้ำเกลือหรือสารละลายเข้าไปในร่างกายของผู้ป่วยจะเป็นการประมาณโดยคร่าวๆ เท่านั้น เป็นการกะประมาณที่มีความละเอียดน้อย แต่การใช้เครื่องควบคุมการให้สารละลายจะมีความสะดวกและละเอียดมากกว่า ในกรณีที่ผู้ป่วยต้องการให้น้ำหรือสารละลายเข้าไปในร่างกายในอัตราคงที่ อุปกรณ์นี้จะมีความสะดวกและเป็นประโยชน์กับผู้ป่วยมาก

แนวทางการพัฒนาการทำงานเครื่องต่อไป

1. ควรเพิ่มวงจรตรวจจับฟองอากาศเข้าไปด้วย เนื่องจากการให้สารละลายเข้าไปในตัวผู้ป่วยต้องไม่มีฟองอากาศ ซึ่งนำเอาออกซิเจนจากภายนอกเข้าไปในร่างกาย มิฉะนั้นจะเกิดอันตรายต่อผู้ป่วยมาก
2. กรณีที่สายน้ำเกลือหลุดออกจากกลไกการให้สารละลายที่ติดอยู่กับแกนมอเตอร์ จะทำให้ SENSOR ยังคงตรวจจับหยดน้ำได้ตลอดและส่งสัญญาณไปให้ CPU ประมวลผล แต่ไม่มีสัญญาณเตือนดังขึ้นซึ่งจะเป็นอันตรายกับผู้ป่วยมาก เนื่องจากสารละลายจะไหลเข้าสู่ตัวผู้ป่วยตลอดเวลา
3. ควรมีการโยกสายสัญญาณไปยังจุดอื่นด้วย เพื่อว่าเมื่อเกิดการ Alarm แพทย์หรือพยาบาลที่อยู่ ณ จุดอื่นซึ่งไม่ได้อยู่ในห้องผู้ป่วยจะได้เข้ามาดูและทำการแก้ไขได้ทัน

ภาคผนวก

pump.lst

pump.lst

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 1

```

pump.asm
2200      1      ORG 2200H
E8C0=    2      COMMAND EQU 0E0C0H
E8C1=    3      READBUSY EQU 0E0C1H
E8C2=    4      WRITEDATA EQU 0E0C2H
E8C3=    5      READDATA EQU 0E0C3H
E8E3=    6      CTRL EQU 0E0E3H
E8E9=    7      PORTA EQU 0E0E9H
3081=    8      TBLOW EQU 3000H
3081=    9      TBHIGH EQU 3001H
0890=   10      SENSOR EQU P1.0
0891=   11      TIMEBASE EQU P1.1
0892=   12      PULSEWIDTH EQU P1.2
0893=   13      INCSW EQU P1.3
0894=   14      DECSW EQU P1.4
0895=   15      ENTERSW EQU P1.5
0896=   16      STOPSW EQU P1.6
17
18      ;-----
19      ;***** MAIN program *****
*****
20
2200 1226D7 21      START: LCALL INITLCD ; set init
ial LCD
2203 1223C4 22      LCALL SETPORT ; set P1 &
PORTA(o/p)
2206 D292 23      SETB PULSEWIDTH ; stop mot
or
2208 1223CE 24      LCALL SETVOLUME ; set tota
l volume
220B 1225CB 25      LCALL DELAY2
220E 00 26      WAIT1: NOP
220F 3095FC 27      JNB ENTERSW,WAIT1
2212 1224E0 28      AEK: LCALL SETRATE ; set flow
rate ml/h
2215 1225CB 29      LCALL DELAY2
2218 00 30      WAIT2: NOP
2219 3095FC 31      JNB ENTERSW,WAIT2
221C 8D0002 32      CJNE R5,#0,DEVI
221F 80F1 33      SJMP AEK
2221 122611 34      DEVI: LCALL DEVIDE ; calculat
e
2224 7C38 35      MOV R4,#30H
2226 7B30 36      MOV R3,#30H
2228 7A30 37      MOV R2,#30H
222A 7938 38      MOV R1,#30H
222C 022332 39      LJMPL DISPC2
222F 90E0E0 40      MAIN: MOV DPTR,#PORTA ; move dro
p's delay
2232 7400 41      MOV A,#00H ; from mem
ory
2234 F0 42      MOVX @DPTR,A
2235 903000 43      MOV DPTR,#TBLOW ; low byte
to R5
2238 E0 44      MOVX A,@DPTR
2239 FD 45      MOV R5,A
223A 903001 46      MOV DPTR,#TBHIGH ; high byt
e to R
223D E0 47      MOVX A,@DPTR
223E F5F0 48      MOV B,A
2240 2091FD 49      TBASE: JB TIMEBASE,$
2243 90E0E0 50      MOV DPTR,#PORTA ; set PA.0
2246 7401 51      MOV A,#01H ; ( flowing
LED )
2248 F0 52      MOVX @DPTR,A

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 2

```

pump.asm
2249 C292 59      PWM: CLR PULSEWIDTH ; drive mo
tor
224B 209151 60      CHECKTBH: JB TIMEBASE,CHECKTBL
224E 209153 61      JB 01H,COUNTTTB
2251 209003 62      CHECKSEN: JB SENSOR,LOOPN
2254 0222EE 63      LJMPL FLAGSEN
2257 0292 64      LOOPN: SETB PULSEWIDTH
2259 D202 65      SETB 02H
225B 309003 66      JNB SENSOR,COUNTSEN
225E 0222CE 67      LJMPL CHECKSTOP
2261 08 68      COUNTSEN: INC R0 ; detect 1
sensor
2262 C202 69      CLR 02H
2264 880F67 70      CJNE R0,#15,CHECKSTOP ; 15 drop
= i ml
2267 7400 71      MOV A,#00H
2269 F8 72      MOV R0,A
226A 8E0106 73      CJNE R5,#1,LOOPE
226D 8F0007 74      CJNE R7,#0,LOOPF
2270 02227D 75      LJMPL LASTDROP
2273 1E 76      LOOPE: DEC R6
2274 02230E 77      LJMPL COUNTCC
2277 1F 78      LOOPF: DEC R7
2278 7EFF 79      MOV R6,#OFFH
227A 02230E 80      LJMPL COUNTCC
227D 09 81      LASTDROP: INC R1
227E 893A18 82      CJNE R1,#3AH,DISPCC3
2281 0A 83      INC R2
2282 7930 84      MOV R1,#30H
2284 BA3A12 85      CJNE R2,#3AH,DISPCC3
2287 0B 86      INC R3
2288 7A30 87      MOV R2,#30H
228A 7930 88      MOV R1,#30H
228C 8B3A0A 89      CJNE R3,#3AH,DISPCC3
228F 0C 90      INC R4
2290 7B30 91      MOV R3,#30H
2292 7A30 92      MOV R2,#30H
2294 7930 93      MOV R1,#30H
2296 8C3A00 94      CJNE R4,#3AH,DISPCC3
2299 122338 95      DISPCC3: LCALL DISPLAYCC
229B 122338 96      LJMPL COMPLETE
229C 022360 97      CHECKTBL: SETB 01H
229F D201 98      JB TIMEBASE,CHECKSEN
22A1 2091AD 99      COUNTTB: CLR 01H
22A4 C201 100      CJNE R5,#0,LOOPG
22A6 8D0018 101      XCH A,B
22A9 C5F0 102      CJNE A,#0,LOOPH
22AB 8A0017 103      XCH A,B
22AE C5F0 104      MOV DPTR,#TBLOW
22B0 903000 105      MOVX A,@DPTR
22B3 E0 106      MOV R5,A
22B4 FD 107      MOV DPTR,#TBHIGH
22B5 903001 108      MOVX A,@DPTR
22B8 E0 109      MOV B,A
22B9 F5F0 110      MOVX A,@DPTR
22BB 309216 111      JNB 0,A
22BE 022249 112      LJMPL PWH

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 3

pump.asm

```

22C1 10      118
22C2 022251 119 LOOPG: DEC R5
                L JMP CHECKSEN
                120
                121
22C5 C5F0   122 LOOPH: XCH A,B
22C7 15F0   123         DEC B
22C9 70FF   124         MOV R5,#0FFH
22CB 022251 125         L JMP CHECKSEN
                126
22CE 309623 127 CHECKSTOP: JNB STOPSW,LOOPI
22D1 022248 128         L JMP CHECKTBN
                129
22D4 0292   130 OCCLUSION: SETB PULSEWIDTH ; stop mo
tor
22D6 90E0E0 131         MOV DPTR,#PORTA
22D9 7404   132         MOV A,#04H
22DB F0     133         MOVX @DPTR,A
22DC 2096FD 134         JB ENTERSV,$
22DF 1225C8 135         L CALL DELAY2
22E2 3096FD 136         JNB ENTERSV,$
22E5 90E0E0 137         MOV DPTR,#PORTA
22E8 7400   138         MOV A,#04H
22EA F0     139         MOVX @DPTR,A
22EB 022240 140         L JMP TBASE
                141
22EE 3002D0 142 FLAGSEN: JNB 02H,CHECKSTOP
22F1 022251 143         L JMP COUNTSEN
                144
22F4 1225C8 145 LOOPI: LCALL DELAY2
22F7 3096FD 146         JNB STOPSW,$
22FA 0292   147         SETB PULSEWIDTH
22FC 90E0E0 148         MOV DPTR,#PORTA
22FF 7400   149         MOV A,#04H
2301 F0     150         MOVX @DPTR,A
2302 2096FD 151         JB ENTERSV,$
2305 1225C8 152         L CALL DELAY2
2308 3096FD 153         JNB ENTERSV,$
230B 022240 154         L JMP TBASE
                155
COUNTCC: INC R1
230E 09     156         CJNE R1,#3AH,DISPCC1
230F 893A1A 157         INC R2
2312 0A     158         MOV R1,#30H
2313 7930   159         CJNE R2,#3AH,DISPCC1
2315 8A3A14 160         INC R3
2318 0B     161         MOV R2,#30H
2319 7A30   162         CJNE R3,#3AH,DISPCC1
231B 7930   163         MOV R1,#30H
231D 8B3A0C 164         CJNE R3,#3AH,DISPCC1
2320 0C     165         INC R4
2321 7830   166         MOV R3,#30H
2323 7A30   167         MOV R2,#30H
2325 7930   168         MOV R1,#30H
2327 8C3A02 169         CJNE R4,#3AH,DISPCC1
232A 8000   170         SJMP DISPCC1
                171
DISPCC1: LCALL DISPLAYCC
232C 122338 172         L JMP CHECKSTOP
                173
DISPCC2: LCALL DISPLAYCC
232F 0222CE 174         L JMP MAIN
                175
2332 122338 175
2335 02222F 176

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 4

pump.asm

```

2338 C082   177 DISPLAYCC: PUSH DPL
233A C083   178         PUSH DPH
233C 74C0   179         MOV A,#0C0H
233E 90E0C0 180         MOV DPTR,#COMMAND
2341 F0     181         MOVX @DPTR,A
2342 1226C7 182         L CALL WAITBF
2345 EC     183         MOV A,R4
2346 1226B7 184         L CALL WRITE
2349 EB     185         MOV A,R3
234A 1226B7 186         L CALL WRITE
234D EA     187         MOV A,R2
234E 1226B7 188         L CALL WRITE
2351 E9     189         MOV A,R1
2352 1226B7 190         L CALL WRITE
2355 7460   191         MOV A,#6DH
2357 1226B7 192         L CALL WRITE
235A 746C   193         MOV A,#6CH
235C 1226B7 194         L CALL WRITE
235F 780E   195         MOV R0,#14
2361 74A0   196         MOV A,#0A0H
2363 1226B7 197         L CALL WRITE
2366 08F9   198         DJNZ R0,LOOPM
2368 0083   199         POP DPH
236A 0082   200         POP DPL
236C 22     201         RET
                202
                203
236D 90E0E0 204 COMPLETE: MOV DPTR,#PORTA
2370 7402   205         MOV A,#02H
2372 F0     206         MOVX @DPTR,A
2373 70C0   207         MOV R5,#0C0H
2375 75F05D 208         MOV B,#5DH
2378 C201   209         CLR 01H
237A 2091FD 210         JB TIMEBASE,$
237D C292   211         PMW1: CLR PULSEWIDTH
237F 20911D 212 CHECKTBH1: JB TIMEBASE,CHECKTBL1
2382 20011F 213         JB 01H,COUNTTB1
2385 309002 214 CHECKSEN1: JNB SENSOR,SILEN
2388 0292   215         SETB PULSEWIDTH
238A 309602 216 SILEN: JNB STOPSW,LOOPJ
238D 80F0   217         SJMP CHECKTBH1
                218
238F 1225C8 219 LOOPJ: LCALL DELAY2
2392 3096FD 220         JNB STOPSW,$
2395 0292   221         SETB PULSEWIDTH
2397 90E0E0 222         MOV DPTR,#PORTA
239A 7400   223         MOV A,#04H
239C F0     224         MOVX @DPTR,A
239D 8023   225         SJMP STOP
                226
239F 0201   227 CHECKTBL1: SETB 01H
23A1 2091E1 228         JB TIMEBASE,CHECKSEN1
23A4 C201   229 COUNTTB1: CLR 01H
23A6 80000C 230         CJNE R5,#0,LOOPK
23A9 C5F0   231         XCH A,B
23AB 840008 232         CJNE A,#0,LOOPL
23AE C5F0   233         XCH A,B
23B0 70C0   234         MOV R5,#0C0H
23B2 75F05D 235         MOV B,#5DH

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 5

pump.asm

```

2385 1D      236 LOOPK: DEC    R5
2386 022385  237 LJMPL      CHECKSEN1
                238
2389 C5F0    239 LOOPL: XCH   A,B
238B 19F0    240 DEC        B
238D 7DFF    241 MOV       R5,#0FFH
238F 022385  242 LJMPL      CHECKSEN1
                243
23C2 80FE    244 STOP: SJMP $
                245
                -----
                247 ;***** " SETPORT " *****
*****
23C4 75907B  248 SETPORT: MOV   P1,#7BH      ;01111011B
23C7 90E0E3  249 DPTR,#CTRL
23CA 7483    250 MOV       A,#3AH
23CC F0     251 MOVX    #DPTR,A
23CD 22     252 RET
                -----
                256 ;***** " SETVOLUME " *****
*****
23CE 7814    257 SETVOLUME: MOV  R0,#20
23D0 7930    258 MOV       R1,#30H
23D2 7A30    259 MOV       R2,#30H
23D4 7B30    260 MOV       R3,#30H
23D6 7C30    261 MOV       R4,#30H
23D8 7D00    262 MOV       R5,#0
23DA 7E00    263 MOV       R6,#0
23DC 7F00    264 MOV       R7,#0
23DE 74C0    265 MOV       A,#00H      ;DD RAN=
40H
23E0 90E0C0  267 MOV       DPTR,#COMMAND
23E3 F0     268 MOVX    #DPTR,A
23E4 1226C7  269 LCALL   WAITBF
23E7 9026FB  270 MOV       DPTR,#TAB
23EA E0     271 DISENTERV: MOVX A,#DPTR
23EB 1226B7  272 LCALL   WRITE
23EE A3     273 INC      DPTR
23EF 08F9    274 DJNZ   R0,DISENTERV
23F1 800A    275 SJMP   DISPLAY1
                -----
23F3 30930C  277 SCANSW1: JNB  INCSW,INCVOL
23F6 309457  278 SCANSW2: JNB  DECSW,DECVOL
23F9 2096F7  279 SCANSW3: JB   ENTERSW,SCANSW1
23FC 22     280 RET
                -----
23FD 1224A0  282 DISPLAY1: LCALL DISPVOL
2400 80F1    283 SJMP   SCANSW1
                -----
2402 200409  285 INCVOL: JB   04H,COUNTV
2405 8EFF05  286 CJNE   R6,#0FFH,REGIS
2408 0F     287 INC      R7
2409 7E00    288 MOV       R6,#00H
240B 8001    289 SJMP   COUNTV
                -----
240D 8E     291 REGIS: INC  R6
240E 09     292 COUNTV: INC R1
240F 893A24  293 CJNE   R1,#3AH,DISPLAY2
2412 0A     294 INC      R2

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 6

pump.asm

```

2413 7930    295 MOV       R1,#30H
2415 BA3A1E  296 CJNE   R2,#3AH,DISPLAY2
2418 0B     297 INC      R3
2419 7A30    298 MOV       R2,#30H
241B 7930    299 MOV       R1,#30H
241D BB3A16  300 CJNE   R3,#3AH,DISPLAY2
2420 0C     301 INC      R4
2421 7B30    302 MOV       R3,#30H
2423 7A30    303 MOV       R2,#30H
2425 7930    304 MOV       R1,#30H
2427 BC3A0C  305 CJNE   R4,#3AH,DISPLAY2
242A 7C39    306 MOV       R4,#39H
242C 7B39    307 MOV       R4,#39H
242E 7A39    308 MOV       R2,#39H
2430 7939    309 MOV       R1,#39H
2432 0204    310 SETB   04H
2434 8000    311 SJMP   DISPLAY2
                -----
2436 400F    313 DISPLAY2: JC   DISPLAY3
2438 1224A0  314 LCALL   DISPVOL
243B 1225B1  315 LCALL   LCALL DELAY1
243E 309303  316 INCAGAIN: JNB INCSW,LOOP5
2441 0223F6  317 LJMPL      SCANSW2
                -----
2444 D3     318 LOOP5: SETB  C
2445 80BB    319 SJMP   INCVOL
                -----
2447 1224A0  322 DISPLAY3: LCALL DISPVOL
244A 1225CB  323 LCALL   DELAY2
244D C3     324 CLR      C
244E 80EE    325 SJMP   INCAGAIN
                -----
2450 C204    326 DECVOL: CLR  04H
2452 BE000A  328 CJNE   R6,#0,REGIS1
2455 BF0002  329 CJNE   R7,#0,REGIS2
2458 8022    330 SJMP   ZEROSETV
                -----
245A 1F     332 REGIS2: DEC  R7
245B 7EFF    333 MOV       R6,#0FFH
245D 8001    334 SJMP   COUNTV1
                -----
245F 1E     336 REGIS1: DEC  R6
2460 19     337 COUNTV1: DEC R1
2461 B92F22  338 CJNE   R1,#2FH,DISPLAY4
2464 1A     339 DEC      R2
2465 7939    340 MOV       R1,#39H
2467 BA2F1C  341 CJNE   R2,#2FH,DISPLAY4
246A 1B     342 DEC      R3
246B 7A39    343 MOV       R2,#39H
246D 7939    344 MOV       R1,#39H
246F 8B2F14  345 CJNE   R3,#2FH,DISPLAY4
2472 1C     346 DEC      R4
2473 7B39    347 MOV       R3,#39H
2475 7A39    348 MOV       R2,#39H
2477 7939    349 MOV       R1,#39H
2479 BC2F0A  350 CJNE   R4,#2FH,DISPLAY4
247C 7C30    351 ZEROSETV: MOV R4,#30H
247E 7B30    352 MOV       R3,#30H
2480 7A30    353 MOV       R2,#30H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ออก... ศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการศึกษา

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 7

pump.asm

```

2482 7930 354 MOV R1,#30H
2484 8000 355 SJMP DISPLAY4
356
2486 400F 357 DISPLAY4: JC DISPLAY5
2488 1224A0 358 LCALL DISPVOL
248B 1225E1 359 LCALL DELAY3
248E 389403 360 DECAGAIN: JNB DECSW_LOOP6
2491 0223F9 361 LJMP SCANSW3
362
2494 03 363 LOOP6: SETB C
2495 8089 364 SJMP DECVOL
365
2497 1224A0 366 DISPLAY5: LCALL DISPVOL
249A 1225F8 367 LCALL DELAY4
249D 03 368 CLR C
249E 80EE 369 SJMP DECAGAIN
370
24A0 C082 371 DISPVOL: PUSH DPL
24A2 C083 372 PUSH DPH
24A4 7480 373 MOV A,#80H ;DD RAM=
00H
24A6 90E0C0 374 MOV DPTR,#COMMAND
24A9 F0 375 MOVX @DPTR,A
24AA 1226C7 376 LCALL WAITBF
24AD 7456 377 MOV A,#5EH ;# v #
24AF 1226B7 378 LCALL WRITE
24B2 744F 379 MOV A,#4FH ;# o #
24B4 1226B7 380 LCALL WRITE
24B7 744C 381 MOV A,#4CH ;# l #
24B9 1226B7 382 LCALL WRITE
24BC 742E 383 MOV A,#2EH ;# . #
24BE 1226B7 384 LCALL WRITE
24C1 EC 385 MOV A,R4
24C2 1226B7 386 LCALL WRITE
24C5 EB 387 MOV A,R3
24C6 1226B7 388 LCALL WRITE
24C9 EA 389 MOV A,R2
24CA 1226B7 390 LCALL WRITE
24CD E9 391 MOV A,R1
24CE 1226B7 392 LCALL WRITE
24D1 7460 393 MOV A,#6DH ;# m #
24D3 1226B7 394 LCALL WRITE
24D6 746C 395 MOV A,#6CH ;# l #
24D8 1226B7 396 LCALL WRITE
24DB 0083 397 POP DPH
24DD 0082 398 POP DPL
24DF 22 399 RET
400
-----
401
*****
402 ;***** " SETRATE " *****
403
404 SETRATE: MOV R0,#20
405 MOV R1,#30H
406 MOV R2,#30H
407 MOV R3,#30H
408 MOV R5,#00H
409 PUSH DPL
410 PUSH DPH
411 MOV A,#0C0H ;DD RAM=
40H
24F0 90E0C0 412 MOV DPTR,#COMMAND

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 8

pump.asm

```

24F3 F0 413 MOVX @DPTR,A
24F4 1226C7 414 LCALL WAITBF
24F7 0083 415 POP DPH
24F9 0082 416 POP DPL
24FB E0 417 DISENTERR: MOVX A,@DPTR
24FC 1226B7 418 LCALL WRITE
24FF A3 419 INC DPTR
2500 08F9 420 DJNZ R0,DISENTERR
2502 800D 421 SJMP DISPLAY1
422
2504 1225B1 423 LCALL DELAY1
2507 30930C 424 SCANSWR1: JNB INCSW,INCRATE
250A 309438 425 SCANSWR2: JNB DECSW,DECRATE
250D 2095F7 426 SCANSWR3: JB ENTERSW,SCANSWR1
2510 22 427 RET
428
2511 12257F 429 DISPLAY1: LCALL DISPRATE
2514 80F1 430 SJMP SCANSWR1
431
2516 00 432 INCRATE: INC R5
2517 09 433 INC R1
2518 893A10 434 CJNE R1,#3AH,DISPLAY2
251B 0A 435 INC R2
251C 7930 436 MOV R1,#30H
251E 8A3A0A 437 CJNE R2,#3AH,DISPLAY2
2521 0B 438 INC R3
2522 7A30 439 MOV R2,#30H
2524 7930 440 MOV R1,#30H
2526 8B3302 441 CJNE R3,#33H,DISPLAY2
2529 8000 442 SJMP DISPLAY2
443
252B 400F 444 DISPLAY2: JC DISPLAY3
252D 12257F 445 LCALL DISPRATE
2530 1225B1 446 LCALL DELAY1
2533 309303 447 INCAGAIN1: JNB INCSW,LOOP7
2536 02250A 448 LJMP SCANSWR2
449
2539 03 450 LOOP7: SETB C
253A 80DA 451 SJMP INCRATE
452
253C 12257F 453 DISPLAY3: LCALL DISPRATE
253F 1225CB 454 LCALL DELAY2
2542 C3 455 CLR C
2543 80EE 456 SJMP INCAGAIN1
457
2545 800002 458 DECRATE: CJNE R5,#0,DECREASE
2548 8013 459 SJMP ZEROSETR
460
254A 10 461 DECREASE: DEC R5
254B 13 462 COUNTR: DEC R1
254C 892F16 463 CJNE R1,#2FH,DISPLAY4
254F 1A 464 DEC R2
2550 7939 465 MOV R1,#33H
2552 8A2F10 466 CJNE R2,#2FH,DISPLAY4
2555 1B 467 DEC R3
2556 7A39 468 MOV R2,#39H
2558 7939 469 MOV R1,#39H
255A 8B2F08 470 CJNE R3,#2FH,DISPLAY4
255D 7B30 471 ZEROSETR: MOV R3,#30H

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
Page 11

pump.asm

```

2626 800006 590 CJNE R0,#0,LOOPB
2629 75F060 591 MOV B,#96
262C A4 592 MUL AB
262D 8073 593 SJMP KEEPDATA
594
262F 75F060 595 LOOPB: MOV B,#96
2632 A4 596 MUL AB
2633 F9 597 MOV R1,A
2634 AAF0 598 MOV R2,B
2636 E8 599 MOV A,R0
2637 75F060 600 MOV B,#96
263A C0F0 601 PUSH B
263C 8DF0 602 MOV B,R5
263E A9F0 603 MOV R0,B
2640 D0F0 604 POP B
2642 A4 605 MUL AB
2643 B40009 606 CHECKA: CJNE A,#0,LOWBYTE
2646 C5F0 607 XCH A,B
2648 B40008 608 XCH A,#0,HIGHBYTE
2648 C5F0 609 XCH A,B
264D 8037 610 SJMP ADDDATA
611
264F 18 612 LOWBYTE: DEC R0
2650 14 613 DEC A
2651 8009 614 SJMP CHECKR0
615
2653 C5F0 616 HIGHBYTE: XCH A,B
2655 18 617 DEC R0
2656 15F0 618 DEC B
2658 74FF 619 MOV A,#0FFH
265A 8009 620 SJMP CHECKR0
621
265C B800E4 622 CHECKR0: CJNE R0,#0,CHECKA
265F 0B 623 INC R3
2660 C0F0 624 PUSH B
2662 8DF0 625 MOV B,R5
2664 A8F0 626 MOV R0,B
2666 D0F0 627 POP B
2668 C5F0 628 XCH A,B
266A B40002 629 CJNE A,#0,LOOPXCH
266D 8004 630 SJMP LOOPQ
631
266F C5F0 632 LOOPXCH: XCH A,B
2671 80D0 633 SJMP CHECKA
634
2673 C5F0 635 LOOPQ: XCH A,B
2675 C0F0 636 PUSH B
2677 F5F0 637 MOV B,A
2679 C3 638 CLR C
267A 98 639 A,R0
267B E5F0 640 MOV A,B
267D D0F0 641 POP B
267F 50C2 642 JNC CHECKA
2681 C3 643 CLR C
2682 9C 644 SUBB A,R4
2683 4001 645 JC ADDDATA
2685 0B 646 INC R3
2686 B9FF02 647 ADDDATA: CJNE R1,#0FFH,SNDDATA1
2689 8009 648 SJMP SNDDATA2

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
Page 12

pump.asm

```

649
2688 80002 650 SNDDATA1: CJNE R3,#0,LOOPC
268E 800F 651 SJMP MOVDATA
652
2690 09 653 LOOPC: INC R1
2691 1B 654 DEC R3
2692 80F2 655 SJMP ADDDATA
656
2694 80002 657 SNDDATA2: CJNE R3,#0,LOOPD
2697 8006 658 SJMP MOVDATA
659
2699 0A 660 LOOPD: INC R2
269A 7900 661 MOV R1,#0
269C 1B 662 DEC R3
269D 80E7 663 SJMP ADDDATA
664
269F E9 665 MOVDATA: MOV A,R1
26A0 8AF0 666 MOV B,R2
26A2 C082 667 KEEPDATA: PUSH DPL
26A4 C083 668 PUSH DPH
26A6 903000 669 MOV DPTR,#TBLOW
26A9 F0 670 MOVX @DPTR,A
26AA E5F0 671 MOV A,B
26AC 903001 672 MOV DPTR,#TBHIGH
26AF F0 673 MOVX @DPTR,A
26B0 E9 674 MOV A,R1
26B1 D083 675 POP DPH
26B3 D082 676 POP DPL
26B5 00 677 POP: NOP
26B6 22 678 RET
679
680 ;
681 ;***** " WRITE ASCII TO LCD " *****
682
26B7 C082 683 WRITE: PUSH DPL
26B9 C083 684 PUSH DPH
26BB 90E0C2 685 MOV DPTR,#WRITEDATA
26BE F0 686 MOVX @DPTR,A
26BF 1226C7 687 LCALL WAITBF
26C2 D083 688 POP DPH
26C4 D082 689 POP DPL
26C6 22 690 RET
691
692 ;
693 ;***** " WAIT BUSY FLAG " *****
694
26C7 C082 695 WAITBF: PUSH DPL
26C9 C083 696 PUSH DPH
26CB 90E0C1 697 MOV DPTR,#READBUSY
26CE E0 698 RDY1: MOVX A,@DPTR
26CF 20E7FC 699 JB ACC.7,RDY1
26D2 D083 700 POP DPH
26D4 D082 701 POP DPL
26D6 22 702 RET
703
704 ;
705 ;***** " SET LCD " *****
706
26D7 C082 707 INITLCD: PUSH DPL

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 9

```

pump.asm
255F 7A30 472 MOV R2,#30H
2561 7930 473 MOV R1,#30H
2563 8000 474 SJMP DISPLAYR4
475
2565 406F 476 DISPLAYR4: JC DISPLAYR5
2567 12257F 477 LCALL DISPRATE
256A 1225E1 478 LCALL DELAY3
256D 309403 479 DECAGAIN1: JNB DECSV_LOOP8
2570 02250D 480 LJMPC SCANSVR3
481
2573 03 482 LOOP8: SETB C
2574 80CF 483 SJMP DECRATE
484
2576 12257F 485 DISPLAYR5: LCALL DISPRATE
2579 1225FB 486 LCALL DELAY4
257C 03 487 CLR C
257D 80EE 488 SJMP DECAGAIN1
489
257F C082 490 DISPRATE: PUSH DPL
2581 C083 491 PUSH DPH
2583 748D 492 MOV A,#8DH ;DD RAM=
ODH
2585 90E0C0 493 MOV DPTR,#COMMAND
2588 F0 494 @DPTR,A
2589 1226C7 495 LCALL WAITBF
258C E8 496 MOV A,R3
258D 1226B7 497 LCALL WRITE
2590 EA 498 MOV A,R2
2591 1226B7 499 LCALL WRITE
2594 E9 500 MOV A,R1
2595 1226B7 501 LCALL WRITE
2598 748D 502 MOV A,#8DH ;* * *
259A 1226B7 503 LCALL WRITE ;* * *
259D 746C 504 MOV A,#8CH ;* * *
259F 1226B7 505 LCALL WRITE ;* * *
25A2 742F 506 MOV A,#2FH ;* / *
25A4 1226B7 507 LCALL WRITE ;* / *
25A7 7468 508 MOV A,#68H ;* h *
25A9 1226B7 509 LCALL WRITE ;* h *
25AC D083 510 POP DPH
25AE D082 511 POP DPL
25B0 22 512 RET
513
-----
514
*****
515 ;***** DELAY 1 *****
DELAY1: PUSH DPL ;750ms
MOV R0,#100
LOOP1: MOV DPH,#255
LOOP2: NOP
DJNZ DPL_LOOP2
DJNZ DPH_LOOP1
DJNZ R0_LOOP
POP DPH
RET
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 10

```

pump.asm
*****
531 ;***** DELAY 2 *****
25C8 C082 532 DELAY2: PUSH DPL ;100ms
25CD C083 533 PUSH DPH
25CF 758382 534 MOV DPH,#130
25D2 758282 535 LOOP3: MOV DPL,#130
25D5 00 537 LOOP4: NOP
25D6 D582FC 538 DJNZ DPL_LOOP4
25D9 D583F6 539 DJNZ DPH_LOOP3
25DC D083 540 POP DPH
25DE D082 541 POP DPL
25E0 22 542 RET
543
-----
544
*****
545 ;***** DELAY 3 *****
25E1 C082 546 DELAY3: PUSH DPL ;390ms
25E3 C083 548 PUSH DPH
25E5 7802 549 MOV R0,#2
25E7 7583FF 550 K1: MOV DPH,#255
25EA 7582FF 551 K2: MOV DPL,#255
25ED 00 552 K3: NOP
25EE D582FC 553 DJNZ DPL_K3
25F1 D583F6 554 DJNZ DPH_K2
25F4 D8F1 555 DJNZ R0,K1
25F6 D083 556 POP DPH
25F8 D082 557 POP DPL
25FA 22 558 RET
559
-----
560
*****
561 ;***** DELAY 4 *****
25FB C082 562 DELAY4: PUSH DPL
25FD C083 564 PUSH DPH
25FF 758301 565 MOV DPH,#1
2602 758201 566 K4: MOV DPL,#1
2605 00 567 K5: NOP
2606 D582FC 568 DJNZ DPL_K5
2609 D583F6 569 DJNZ DPH_K4
260C D083 570 POP DPH
260E D082 571 POP DPL
2610 22 572 RET
573
-----
574
*****
575 ;***** "DEVIDE" *****
*****
576 ;*** PROGRAM (24,000/RATE) = (250/RATE). (9
6) ***
2611 800003 577 DEVIDE: CJNE R5,#0,LOOPA
2614 022685 579 LJMPC POH
580
2617 7800 581 LOOPA: MOV R3,#00H
2619 ED 582 MOV A,R5
261A 75F002 583 MOV B,#2
261D 84 584 DIV AB
261E FC 585 MOV R4,A
261F 80F0 586 MOV B,R5
2621 74FA 587 MOV A,#250 ; 100 Hz TIMEBAS
E
2623 84 588 DIV AB
2624 A8F0 589 MOV R0,B

```

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
                                Page 13
pump.asm
26D9 C083 708 PUSH DPH
26DB 90E0C0 709 MOV DPTR,#COMMAND ;FUNCTION S
ET
26DE 7438 710 MOV A,#38H ;2 LINE DIS
PLAY
26E0 F0 711 MOVX @DPTR,A
26E1 1226C7 712 LCALL WAITBF
26E4 740C 713 MOV A,#0CH ;DISPLAY 0
N OFF CONTROL
26E6 F0 714 MOVX @DPTR,A
26E7 1226C7 715 LCALL WAITBF
26EA 7406 716 MOV A,#6 ;ENTRY MODE
SET
26EC F0 717 MOVX @DPTR,A
26ED 1226C7 718 LCALL WAITBF
26F0 7401 719 MOV A,#1 ;CLEAR AND
HOME
26F2 F0 720 MOVX @DPTR,A
26F3 1226C7 721 LCALL WAITBF
26F6 0083 722 POP DPH
26F8 0082 723 POP DPL
26FA 22 724 RET
725
726 ;-----
727
26FB 20454E54 728 TAB: DB " ENTER TOTAL VOLUME "
26FF 4552054
2703 4F54414C
2707 20584F4C
2708 55404520
270F 2020464E 729 DB " ENTER FLOW RATE "
2713 54455220
2717 464C4F57
2718 20524154
271F 45202020
0000= 730
731
END

```

```

279 426
flaglen = 22EE 142 63
highbyte = 2653 616 608
incagain = 243E 316 325
incagainf = 2533 447 456
incrate = 2516 432 424 451
incsw = 0093 13 277 316 424 447
incvol = 2402 285 277 320

```

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
                                Page 1
pump.asm
adddata = 2686 647 610 645 655 663
aek = 2212 31 37
checka = 2643 606 622 633 642
checkr0 = 265C 622 614 620
checksen = 2251 62 104 120 125
checksen1 = 2385 214 228 237 242
checkst0p = 22CE 127 68 72 142 173
checktbh = 2248 60 128
checktbh1 = 237F 212 217
checktbl = 229F 103 60
checktbl1 = 239F 227 212
command = E0C0 2 181 267 374 412 493 709
complete = 236D 204 101
countcc = 230E 156 80 84
countr = 2548 462
countsen = 2261 70 67 143
counttb = 22A4 105 61
counttb1 = 23A4 229 213
countv = 240E 292 285 289
countv1 = 2460 337 334
ctrl = E0E3 6 250
decagain = 248E 360 369
decagain1 = 256D 479 488
decrate = 2545 458 425 483
decrease = 254A 461 458
decsw = 0094 14 278 360 425 479
decvol = 2450 327 278 364 446
delay1 = 2581 517 315 423 135 145 152 219
delay2 = 25CB 533 27 32
323 454
delay3 = 25E1 547 359 478
delay4 = 25FB 563 367 486
devi = 2221 39 36
devide = 2611 578 39
disenterr = 24FB 417 420
disenterv = 23EA 271 274
dispcc1 = 232C 172 157 160 164 169 170
dispcc2 = 2332 175 44
dispcc3 = 2299 100 87 90 94 99
display1 = 23FD 202 275
display2 = 2436 313 293 296 300 305 311
display3 = 2447 322 313
display4 = 2486 357 338 341 345 350 355
display5 = 2497 366 357
displaycc = 2338 178 100 172 175
displayr1 = 2511 429 421
displayr2 = 2528 444 434 437 441 442
displayr3 = 253C 453 444
displayr4 = 2565 476 463 466 470 474
displayr5 = 2576 485 476
disprate = 257F 490 429 445 453 477 485
dispvol = 24A0 371 282 314 322 358 366
entersw = 0095 15 30 35 134 136 151 153

```

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology
                                Page 2
pump.asm
initcd = 2607 707 21
k1 = 25E7 550 555
k2 = 25EA 551 554
k3 = 25E0 552 553
k4 = 2602 566 569
k5 = 2605 567 568
keepdata = 26A2 667 593
lastdrop = 2270 86 77
loop = 2587 520 525
loop1 = 258A 521 524
loop2 = 2580 522 523
loop3 = 2502 536 539
loop4 = 2506 537 538
loop5 = 2444 319 316
loop6 = 2494 363 360
loop7 = 2539 450 447
loop8 = 2573 482 479
loopa = 2617 581 578
loopb = 262F 595 590
loopc = 2630 653 650
loopd = 2639 660 657
loope = 2273 79 75
loopf = 2277 82 76
loopg = 22C1 119 106
loopi = 22C5 122 108
loopj = 22F4 145 127
loopk = 238F 219 216
loopl = 2385 236 230
loopm = 2389 239 232
loopn = 2361 197 199
loopo = 2257 65 62
loopq = 2673 635 630
loopxch = 266F 632 629
lowbyte = 264F 612 606
main = 222F 46 176
movdata = 269F 665 651 658
occlusion = 22D4 130 116
pom = 2685 677 579
porta = E0E0 7 46 56 131 137 148 204
222
pulsewidth = 0092 12 25 59 65 116 130 147
211 215 221
pwm = 2249 59 117
pwm1 = 2370 211
rdyl = 26CE 698 699
readbusy = E0C1 3 697
readdate = E0C3 5
regis1 = 2400 291 286
regis2 = 245F 336 328
regis3 = 245A 332 329
scanswr1 = 2507 424 426 430
scanswr2 = 250A 425 448
scanswr3 = 2500 426 480

```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

scansw1 = 23F3 277 279 283
scansw2 = 23F6 278 317
scansw3 = 23F9 279 361
sensor = 0090 10 62 67 214
setport = 23C4 249 23
setrate = 24E0 404 31
setvolume = 23CE 258 26
silen = 238A 216 214
    
```

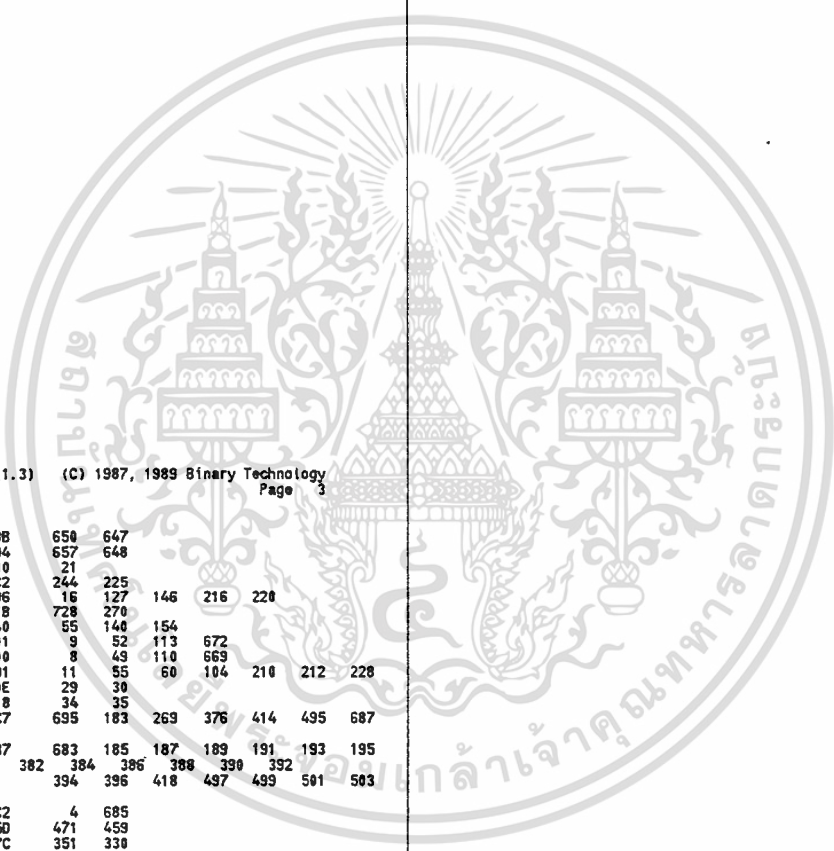
16

8651 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology Page 3

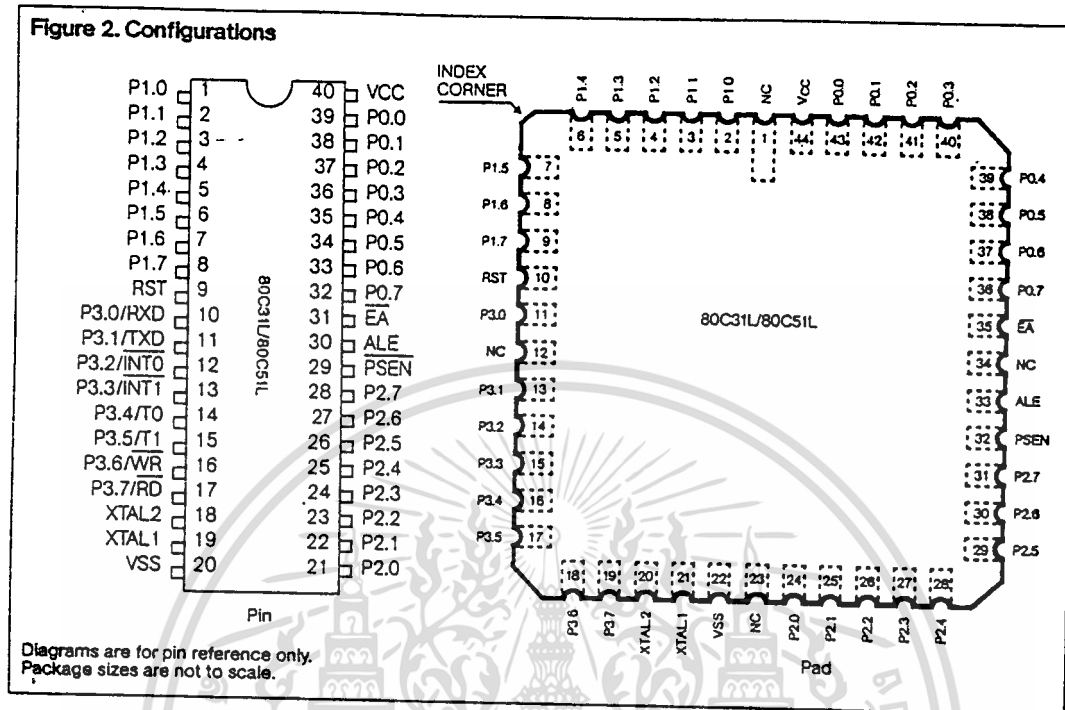
pump.asm

```

saddata1 = 260B 650 647
saddata2 = 2694 657 648
start = 2200 21
stop = 23C2 244 225
stopsw = 0096 16 127 146 216 220
tab = 26F8 728 270
tbase = 2240 55 140 154
tbhigh = 3001 9 52 113 672
tblow = 3000 8 49 110 669
timebase = 0091 11 55 60 104 210 212 228
wait1 = 220E 29 30
wait2 = 2218 34 35
waitbf = 26C7 695 183 269 376 414 495 687
712 715 718 721
write = 2687 683 185 187 189 191 193 195
198 272 378 380 382 384 386 388 390 392
505 507 509 394 396 418 437 499 501 503
writedata = E0C2 4 685
zerosetr = 255D 471 459
zerosetv = 247C 351 330
    
```

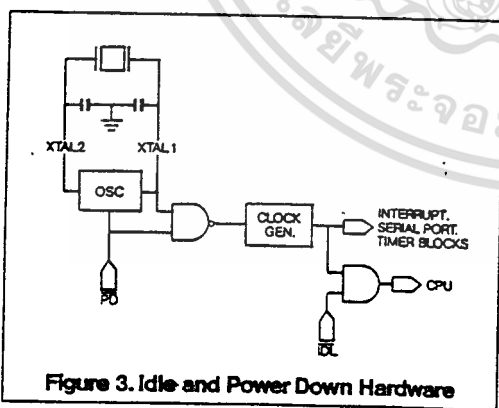


80C51-L/80C31-L



IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off. These special modes are activated by software via the Special Function Register, its hardware address is 87H. PCON is not bit addressable.



PCON: Power Control Register

SMOD	-	-	-	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

Symbol Position Name and Function

SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
-	PCON.6	(Reserved)
-	PCON.5	(Reserved)
-	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX0000).



80C51-L/80C31-L

Table 1. Status of the external pins during Idle and Power Down modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. Table 1 describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiates the Special Function Register (see Table 1).

In the Power Down mode, VCC may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in Figure 4.

STOP CLOCK MODE

Due to static design, the MHS 80C31/C51 clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

80C51 I/O PORTS

The I/O port drive of the 80C51 is similar to the 8051. The I/O buffers for Ports 1, 2, and 3 are implemented as shown in figure 4.

When the port latch contains a 0, all pFETs in figure 4 are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the IOH source current. This inverter and T3 form a latch which holds the 1 and is supported by T2. When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C. Specifications. When the input goes below approximately 2V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

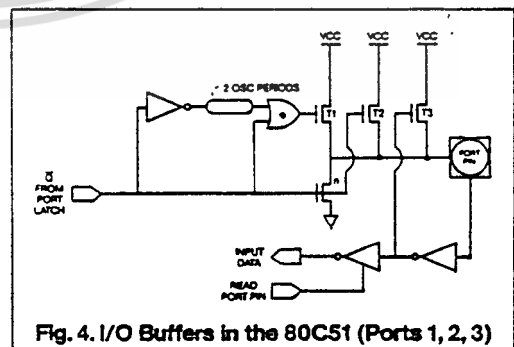


Fig. 4. I/O Buffers in the 80C51 (Ports 1, 2, 3)

80C51-L / 80C31-L

80C51 PIN DESCRIPTIONS

VSS

Circuit ground potential

VCC

Supply voltage during normal, Idle, and Power Down operation.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification. In the 80C51, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) 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 1's. During accesses to external Data Memory that uses 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C51. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the functions of various special features of the MCS-51 Family, as listed below.

Port Pin	Alternate Function
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	TO (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 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pull-down resistor permits Power-On reset using only a capacitor connected to VCC.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

PSEN

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

EA

When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

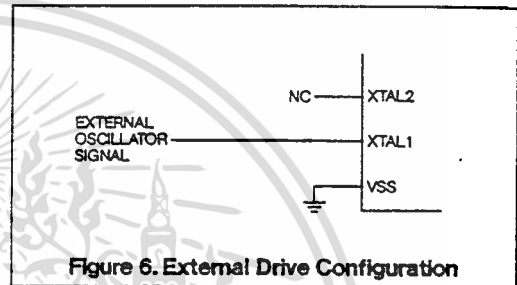
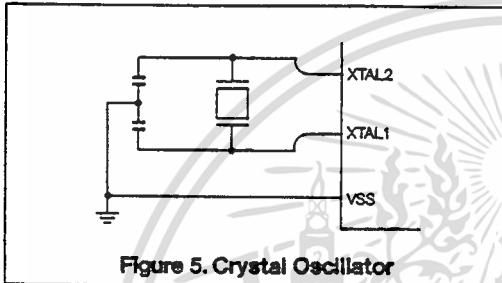
Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

80C51-L/80C31-L

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in figure 5. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left

unconnected as shown in figure 6. 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 high and low times specified on the Data Sheet must be observed.



80C51-L/80C31-L

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias:
 Commercial 0°C to 70°C
 Industrial - 40°C to 85°C
 Storage Temperature - 65°C to + 150°C
 Voltage on VCC to VSS - 0.5V to +7V
 Voltage on Any Pin to VSS - 0.5V to VCC + 0.5V
 Power Dissipation 1W*

*This value is based on the maximum allowable die temperature and the thermal resistance of the package.

***NOTICE:**

Stresses at or above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

TA = - 40°C to 85°C; VCC = 2.7V to 6V; VSS = 0V; F = 0 to 6 MHz

Symbol	Parameter	Min	Max	Unit	Test Conditions
VIL	Input Low Voltage	-0.5	0.2VCC -0.1	V	
VIH	Input High Voltage (Except XTALs and RST)	0.2VCC +0.9	VCC +0.5	V	
VIH1	Input High Voltage to RST for Reset	0.7VCC	VCC +0.5	V	
VIH2	Input High Voltage To XTAL 1	0.7VCC	VCC +0.5	V	
VPD	Power Down Voltage To VCC in PD Mode	2.0	6.0	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6mA (note 1)
VOL1	Output Low Voltage Port 0, ALE, PSEN		0.45	V	IOL = 3.2mA (note 1)
VOH	Output High Voltage Ports 1, 2, 3	0.9VCC		V	IOH = -10µA
		2.4		V	IOH = -60µA VCC = 5V ± 10%
VOH1	Output High Voltage (Port 0 in External in External Bus Mode), ALE, PSEN	0.9VCC		V	IOH = -40µA
		2.4		V	IOH = -400µA VCC = 5V ± 10%
IIL	Logical 0 Input Current Ports 1,2,3		-50	µA	Vin = 0.45V
ILI	Input Leakage Current		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		-500	µA	Vin = 2.0V
ICCPD	Power Supply Current (Power Down Mode)	50	10	µA	VCC = 2.0V to 5.5V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	fc = 1MHz, TA = 25°C

Note 1:

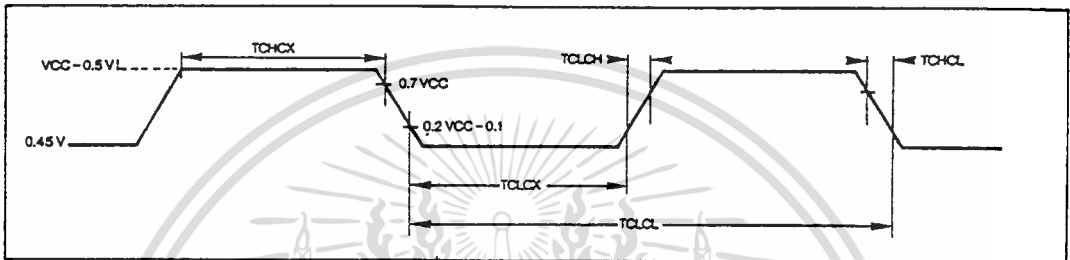
Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0

transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45V with maxi VOL peak 0.6V. A Schmitt Trigger use is not necessary.

80C51-L/80C31-L

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL 1)

Symbol	Parameter	Variable Clock freq = 0 to 6 MHz		Unit
		Min	Max	
TCLCL	Oscillator Period	166		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



AC CHARACTERISTICS

($T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 2.7\text{V}$ to 6V , $V_{SS} = 0\text{V}$)

(Load Capacitance for Port 0, ALE, and PSEN = 100pf; Load Capacitance for All Other Outputs = 80pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

Symbol	Parameter	Min	Max	Units
TLHLL	ALE Pulse Width	$2TCLCL - 40$		ns
TAVLL	Address Valid to ALE	$TCLCL - 55$		ns
TLLAX	Address Hold After ALE	$TCLCL - 35$		ns
TLLIV	ALE to Valid Instr In		$4TCLCL - 170$	ns
TLLPL	ALE to PSEN	$TCLCL - 25$		ns
TPLPH	PSEN Pulse Width	$3TCLCL - 35$		ns
TPLIV	PSEN to Valid Instr In		$3TCLCL - 220$	ns
TPXIX	Input Instr Hold After PSEN	0		ns
TPXIZ	Input Instr Float After PSEN		$TCLCL - 20$	ns
TPXAV	PSEN to Address Valid	$TCLCL - 8$		ns
TAVIV	Address to Valid Instr In		$5TCLCL - 220$	ns
TPLAZ	PSEN Low to Address Float		0	ns

See next page for External Data Memory Characteristics.

EXTERNAL DATA MEMORY CHARACTERISTICS

Symbol	Parameter	Min	Max	Units
TRLRH	RD Pulse Width	6TCLCL - 100		ns
TWLWH	WR Pulse Width	6TCLCL - 100		ns
TLLAX	Data Address Hold After ALE	TCLCL - 35		ns
TRLDV	RD to Valid Data In		5TCLCL - 165	ns
TRHDX	Data Hold After RD	0		ns
TRHDZ	Data Float After RD		2TCLCL - 70	ns
TLLDV	ALE to Valid Data In		8TCLCL - 150	ns
TAVDV	Address to Valid Data In		9TCLCL - 165	ns
TLLWL	ALE to WR or RD	3TCLCL - 50	3TCLCL + 50	ns
TAVWL	Address to WR or RD	4TCLCL - 130		ns
TQVWX	Data Valid to WR Transition	TCLCL - 60		ns
TQVWH	Data Setup to WR High	7TCLCL - 150		ns
TWHQX	Data Hold After WR	TCLCL - 50		ns
TRLAZ	RD Low to Address Float		0	ns
TWHLH	RD or WR High to ALE High	TCLCL - 40	TCLCL + 40	ns

MAXIMUM ICC (mA)

Freq. VCC	Operating (Note 3)			Idle (Note 4)		
	2.7V	5V	6V	2.7V	5V	6V
1 MHz	0.8 mA	1.5 mA	1.8 mA	400 μ A	800 μ A	1 mA
6 MHz	4 mA	8 mA	10 mA	1.2 mA	3.5 mA	3.8 mA

Note 2:

Power Down ICC is measured with all output pins disconnected; EA = Port 0 = VCC; XTAL2 N.C.; RST = VSS

Note 3:

ICC is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + 0.5V; VIH = VCC - 0.5V; XTAL2 N.C.; EA = RST = Port 0 = VCC. ICC would be slightly higher if a crystal oscillator used.

Note 4:

Idle ICC is measured with all output pins disconnected; XTAL1 driven TCLCH, TCHCL = 5 ns, VIL = VSS + 0.5V; VIH = VCC - 0.5V; XTAL2 N.C.; Port 0 = VCC; EA = RST = VSS.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list all the characters and what they stand for.

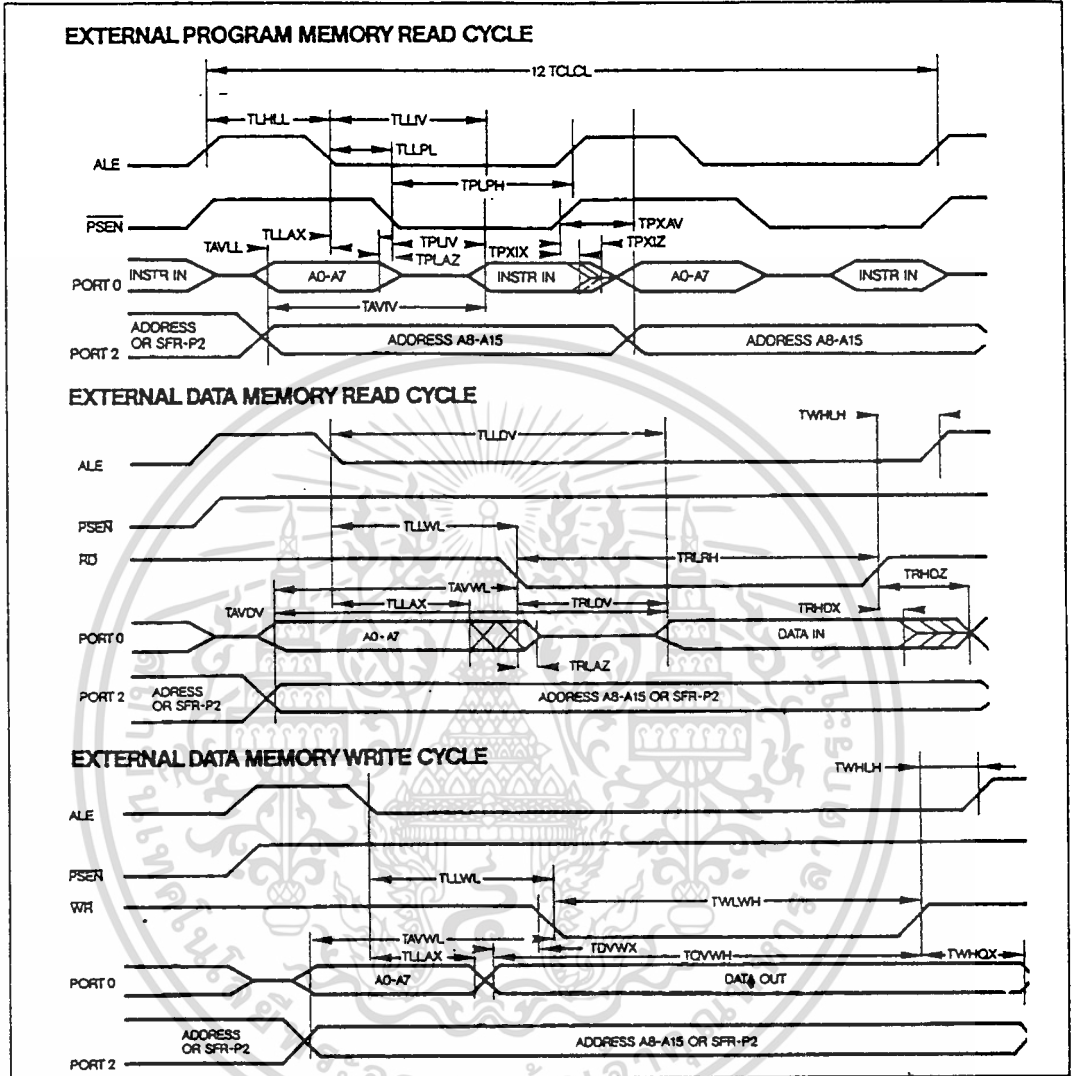
EXAMPLE:

TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

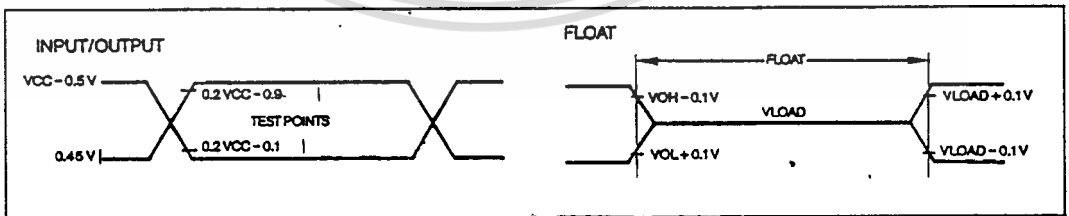
A: Address.
C: Clock.
D: Input data.
H: Logic level HIGH.
I: Instruction (program memory contents).
L: Logic level LOW, or ALE.
P: PSEN

Q: Output data.
R: READ signal.
T: Time.
V: Valid.
W: WRITE signal
X: No longer a valid logic level.
Z: Float.

AC TIMING DIAGRAMS



AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS

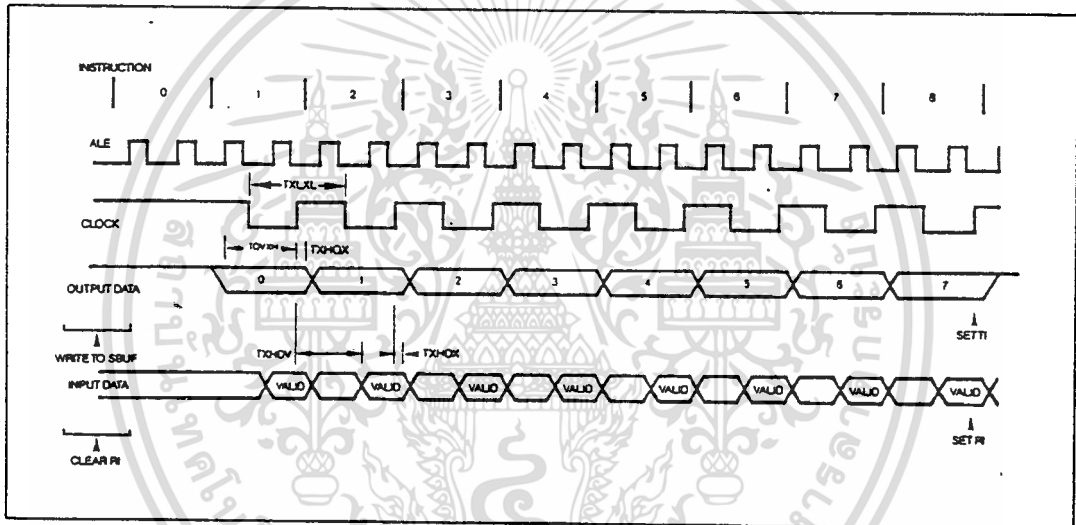


AC inputs during testing are driven at V_{CC} - 0.5 for a logic "1" and 0.45V for a logic "0". Timing measurements are made at V_{IH} min for a logic "1" and V_{IL} max for a logic "0". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. I_{ol}/I_{oh} ≥ ± 20 Ma.

SERIAL PORT TIMING - SHIFT REGISTER MODE**A.C. CHARACTERISTICS:**

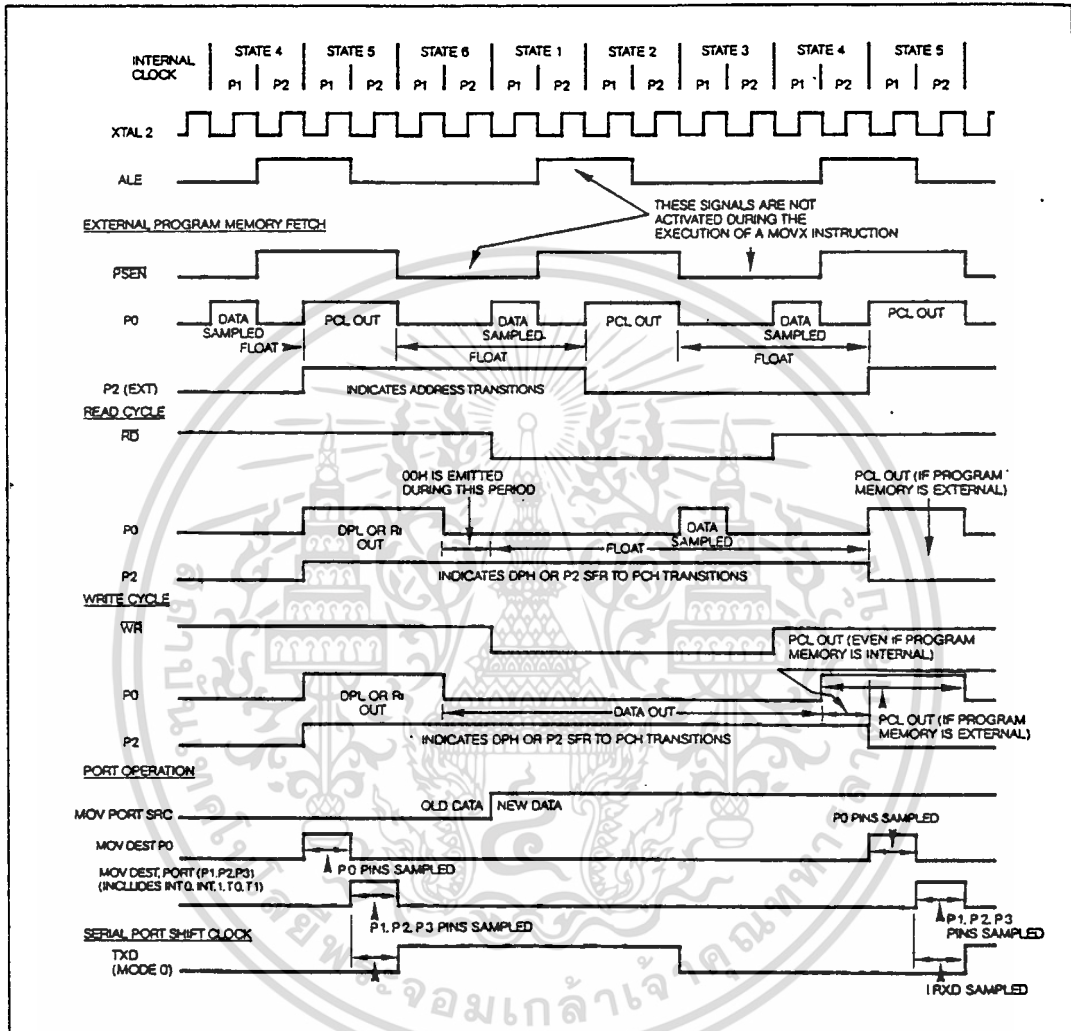
(TA = 0°C to 70°C; VSS = 0V; VCC = 2.7V to 6V; Load Capacitance = 80 pF)

Symbol	Parameter	Min	Max	Units
TXLXL	Serial Port Clock Cycle Time	12TCLCL		μ S
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold After Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TCLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS

80C51-L/80C31-L

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though ($T_A = 25^\circ\text{C}$ fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

Table 1. MCS[®]-51 Instruction Set Description

ARITHMETIC OPERATIONS				
Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
Mnemonic		Destination	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

80C51-L/80C31-L

Table 1. (Cont.)

DATA TRANSFER				
Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTRA	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte form stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1
BOOLEAN VARIABLE MANIPULATION				
Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,1 bit	AND complement of direct bit to Carry	2	2
ORL	C/bit	OR direct bit to Carry flag	2	2
ORL	C,1 bit	OR complement of direct bit to Carry	2	2
MOV	C/bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
Mnemonic		Description	Byte	Cyc
ACALL	addr 11	Absolute Subroutine Call	2	2
LCALL	addr 16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr 11	Absolute Jump	2	2
LJMP	addr 16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

80C51-L/80C31-L

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)				
Mnemonic		Description	Byte	Cyc
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data,rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

- Rn - Working register R0-R7
 direct - 128 internal RAM locations, any I/O port, control or status register
 @Ri - Indirect internal RAM location addressed by register R0 or R1
 #data - 8-bit constant included in instruction
 #data 16 - 16-bit constant included as bytes 2 & 3 of instruction
 bit - 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

- addr 16 - Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
 Addr 11 - Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
 rel - SJMP and all conditional jumps include an 8-bit offset byte. Range is +127-128 bytes relative to first byte of the following instruction.

All mnemonics copyrighted © Intel Corporation 1979

Table 2. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP		33	1	RLC	A
01	2	AJMP	code addr	34	2	ADDC	A,#data
02	3	LJMP	code addr	35	2	ADDC	A,data addr
03	1	RR	A	36	1	ADDC	A,@R0
04	1	INC	A	37	1	ADDC	A,@R1
05	2	INC	data addr	38	1	ADDC	A,R0
06	1	INC	@R0	39	1	ADDC	A,R1
07	1	INC	@R1	3A	1	ADDC	A,R2
08	1	INC	R0	3B	1	ADDC	A,R3
09	1	INC	R1	3C	1	ADDC	A,R4
0A	1	INC	R2	3D	1	ADDC	A,R5
0B	1	INC	R3	3E	1	ADDC	A,R6
0C	1	INC	R4	3F	1	ADDC	A,R7
0D	1	INC	R5	40	2	JC	code addr
0E	1	INC	R6	41	2	AJMP	code addr
0F	1	INC	R7	42	2	ORL	data addr,A
10	3	JBC	bit addr,code addr	43	3	ORL	data addr,#data
11	2	ACALL	code addr	44	2	ORL	A,#data
12	3	LCALL	code addr	45	2	ORL	A,data addr
13	1	RRC	A	46	1	ORL	A,@R0
14	1	DEC	A	47	1	ORL	A,@R1
15	2	DEC	data addr	48	1	ORL	A,R0
16	1	DEC	@R0	49	1	ORL	A,R1
17	1	DEC	@R1	4A	1	ORL	A,R2
18	1	DEC	R0	4B	1	ORL	A,R3
19	1	DEC	R1	4C	1	ORL	A,R4
1A	1	DEC	R2	4D	1	ORL	A,R5
1B	1	DEC	R3	4E	1	ORL	A,R6
1C	1	DEC	R4	4F	1	ORL	A,R7
1D	1	DEC	R5	50	2	JNC	code addr
1E	1	DEC	R6	51	2	ACALL	code addr
1F	1	DEC	R7	52	2	ANL	data addr,A
20	3	JB	bit addr,code addr	53	3	ANL	data addr,#data
21	2	AJMP	code addr	54	2	ANL	A,#data
22	1	RET		55	2	ANL	A,data addr
23	1	RL	A	56	1	ANL	A,@R0
24	2	ADD	A,data	57	1	ANL	A,@R1
25	2	ADD	A,data addr	58	1	ANL	A,R0
26	1	ADD	A,@R0	59	1	ANL	A,R1
27	1	ADD	A,@R1	5A	1	ANL	A,R2
28	1	ADD	A,R0	5B	1	ANL	A,R3
29	1	ADD	A,R1	5C	1	ANL	A,R4
2A	1	ADD	A,R2	5D	1	ANL	A,R5
2B	1	ADD	A,R3	5E	1	ANL	A,R6
2C	1	ADD	A,R4	5F	1	ANL	A,R7
2D	1	ADD	A,R5	60	2	JZ	code addr
2E	1	ADD	A,R6	61	2	AJMP	code addr
2F	1	ADD	A,R7	62	2	XRL	data addr A
30	3	JNB	bit addr,code addr	63	3	XRL	data addr,#data
31	2	ACALL	code addr	64	2	XRL	A,#data
32	1	RETI		65	2	XRL	A,data addr

80C51-L/80C31-L

Table 2. (Cont.)

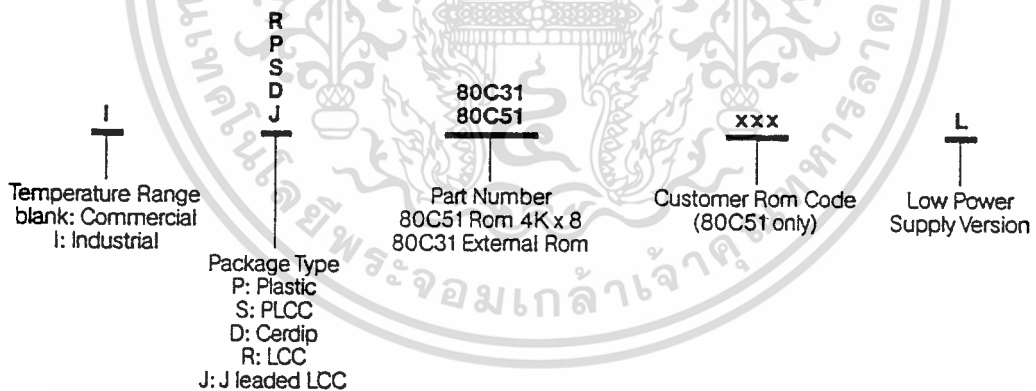
Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0	99	1	SUBB	A,R1
67	1	XRL	A,@R1	9A	1	SUBB	A,R2
68	1	XRL	A,R0	9B	1	SUBB	A,R3
69	1	XRL	A,R1	9C	1	SUBB	A,R4
6A	1	XRL	A,R2	9D	1	SUBB	A,R5
6B	1	XRL	A,R3	9E	1	SUBB	A,R6
6C	1	XRL	A,R4	9F	1	SUBB	A,R7
6D	1	XRL	A,R5	A0	2	ORL	C,bit addr
6E	1	XRL	A,R6	A1	2	AJMP	code addr
6F	1	XRL	A,R7	A2	2	MOV	C,bit addr
70	2	JNZ	code addr	A3	1	INC	DPTR
71	2	ACALL	code addr	A4	1	MUL	AB
72	2	ORL	C,bit addr	A5		reserved	
73	1	JMP	@A+DPTR	A6	2	MOV	@R0,data addr
74	2	MOV	A,#data	A7	2	MOV	@R1,data addr
75	3	MOV	data addr,#data	A8	2	MOV	R0,data addr
76	2	MOV	@R0,#data	A9	2	MOV	R1,data addr
77	2	MOV	@R1,#data	AA	2	MOV	R2,data addr
78	2	MOV	R0,#data	AB	2	MOV	R3,data addr
79	2	MOV	R1,#data	AC	2	MOV	R4,data addr
7A	2	MOV	R2,#data	AD	2	MOV	R5,data addr
7B	2	MOV	R3,#data	AE	2	MOV	R6,data addr
7C	2	MOV	R4,#data	AF	2	MOV	R7,data addr
7D	2	MOV	R5,#data	B0	2	ANL	C,bit addr
7E	2	MOV	R6,#data	B1	2	ACALL	code addr
7F	2	MOV	R7,#data	B2	2	CPL	bit addr
80	2	SJMP	code addr	B3	1	CPL	C
81	2	AJMP	code addr	B4	3	CJNE	A,#data,code addr
82	2	ANL	C,bit addr	B5	3	CJNE	A,data addr,code addr
83	1	MOVC	A,@A+PC	B6	3	CJNE	@R0,#data,code addr
84	1	DIV	AB	B7	3	CJNE	@R1,#data,code addr
85	3	MOV	data addr,data addr	B8	3	CJNE	R0,#data,code addr
86	2	MOV	data addr,@R0	B9	3	CJNE	R1,#data,code addr
87	2	MOV	data addr,@R1	BA	3	CJNE	R2,#data,code addr
88	2	MOV	data addr,R0	BB	3	CJNE	R3,#data,code addr
89	2	MOV	data addr,R1	BC	3	CJNE	R4,#data,code addr
8A	2	MOV	data addr,R2	BD	3	CJNE	R5,#data,code addr
8B	2	MOV	data addr,R3	BE	3	CJNE	R6,#data,code addr
8C	2	MOV	data addr,R4	BF	3	CJNE	R7,#data,code addr
8D	2	MOV	data addr,R5	C0	2	PUSH	data addr
8E	2	MOV	data addr,R6	C1	2	AJMP	code addr
8F	2	MOV	data addr,R7	C2	2	CLR	bit addr
90	3	MOV	DPTR,#data	C3	1	CLR	C
91	2	ACALL	code addr	C4	1	SWAP	A
92	2	MOV	bit addr,C	C5	2	XCH	A,data addr
93	1	MOVC	A,@A+DPTR	C6	1	XCH	A,@R0
94	2	SUBB	A,#data	C7	1	XCH	A,@R1
95	2	SUBB	A,data addr	C8	1	XCH	A,R0
96	1	SUBB	A,@R0	C9	1	XCH	A,R1
97	1	SUBB	A,@R1	CA	1	XCH	A,R2
98	1	SUBB	A,R0	CB	1	XCH	A,R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

80C51-L/80C31-L

Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4	E6	1	MOV	A,@R0
CD	1	XCH	A,R5	E7	1	MOV	A,@R1
CE	1	XCH	A,R6	E8	1	MOV	A,R0
CF	1	XCH	A,R7	E9	1	MOV	A,R1
D0	2	POP	data addr	EA	1	MOV	A,R2
D1	2	ACALL	code addr	EB	1	MOV	A,R3
D2	2	SETB	bit addr	EC	1	MOV	A,R4
D3	1	SETB	C	ED	1	MOV	A,R5
D4	1	DA	A	EE	1	MOV	A,R6
D5	3	DJNZ	data addr,code addr	EF	1	MOV	A,R7
D6	1	XCHD	A,@R0	F0	1	MOVX	@DPTR,A
D7	1	XCHD	A,@R1	F1	2	ACALL	code addr
D8	2	DJNZ	R0,code addr	F2	1	MOVX	@R0,A
D9	2	DJNZ	R1,code addr	F3	1	MOVX	@R1,A
DA	2	DJNZ	R2,code addr	F4	1	CPL	A
DB	2	DJNZ	R3,code addr	F5	2	MOV	data addr,A
DC	2	DJNZ	R4,code addr	F6	1	MOV	@R0,A
DD	2	DJNZ	R5,code addr	F7	1	MOV	@R1,A
DE	2	DJNZ	R6,code addr	F8	1	MOV	R0,A
DF	2	DJNZ	R7,code addr	F9	1	MOV	R1,A
E0	1	MOVX	A,DPTR	FA	1	MOV	R2,A
E1	2	AJMP	code addr	FB	1	MOV	R3,A
E2	1	MOVX	A,@R0	FC	1	MOV	R4,A
E3	1	MOVX	A,@R1	FD	1	MOV	R5,A
E4	1	CLR	A	FE	1	MOV	R6,A
E5	2	MOV	A,data addr	FF	1	MOV	R7,A





MC14528B

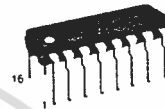
DUAL MONOSTABLE MULTIVIBRATOR

The MC14528B is a dual, retriggerable, resettable monostable multivibrator. It may be triggered from either edge of an input pulse, and produces an output pulse over a wide range of widths, the duration of which is determined by the external timing components, C_X and R_X .

- Separate Reset Available
- Diode Protection on All Inputs
- Triggerable from Leading or Trailing Edge Pulse
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-power TTL Loads or One Low-power Schottky TTL Load Over the Rated Temperature Range
- Pin-for-Pin Replacement with the MC14538B



L SUFFIX
CERAMIC
CASE 620



P SUFFIX
PLASTIC
CASE 648



D SUFFIX
SOIC
CASE 751B

MAXIMUM RATINGS* (Voltages Referenced to V_{SS})

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage	-0.5 to +18.0	V
V_{in}, V_{out}	Input or Output Voltage (DC or Transient)	-0.5 to V_{DD} +0.5	V
I_{in}, I_{out}	Input or Output Current (DC or Transient), per Pin	±10	mA
P_D	Power Dissipation, per Package†	500	mW
T_{stg}	Storage Temperature	-65 to +150	°C
T_L	Lead Temperature (8-Second Soldering)	260	°C

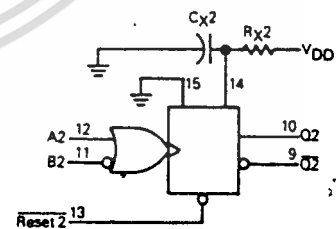
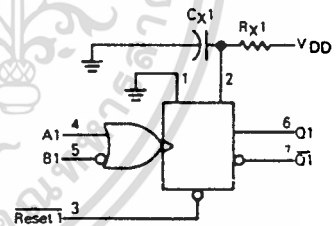
*Maximum Ratings are those values beyond which damage to the device may occur.
†Temperature Derating: Plastic "P" and D/DW" Packages: -7.0 mW/°C From 65°C To 125°C
Ceramic "L" Packages: -12 mW/°C From 100°C To 125°C

ORDERING INFORMATION

MC14XXBCP Plastic
MC14XXBCL Ceramic
MC14XXBD SOIC

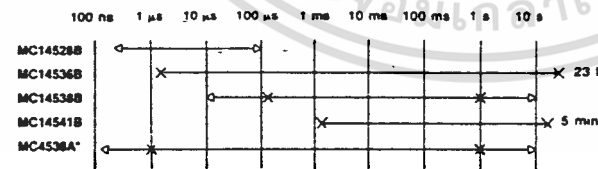
$T_A = -55$ to 125 C for all packages.

BLOCK DIAGRAM



V_{DD} = Pin 16
 V_{SS} = Pin 1, Pin 8, Pin 15
 R_X and C_X are external components

ONE-SHOT SELECTION GUIDE



*LIMITED OPERATING VOLTAGE (2-6V)

Total Output Pulse Width Range
Recommended Pulse Width Range

MC14528B

ELECTRICAL CHARACTERISTICS (Voltages Referenced to V_{SS})

Characteristic	Symbol	V _{DD} Vdc	-55°C		25°C			125°C		Unit
			Min	Max	Min	Typ #	Max	Min	Max	
Output Voltage V _{in} = V _{DD} or 0	"0" Level VOL	5.0	—	0.05	—	0	0.05	—	0.05	Vdc
		10	—	0.05	—	0	0.05	—	0.05	
		15	—	0.05	—	0	0.05	—	0.05	
	"1" Level VOH	5.0	4.95	—	4.95	5.0	—	4.95	—	Vdc
		10	9.95	—	9.95	10	—	9.95	—	
		15	14.95	—	14.95	15	—	14.95	—	
Input Voltage (V _O = 4.5 or 0.5 Vdc) (V _O = 9.0 or 1.0 Vdc) (V _O = 13.5 or 1.5 Vdc)	"0" Level VIL	5.0	—	1.5	—	2.25	1.5	—	1.5	Vdc
		10	—	3.0	—	4.50	3.0	—	3.0	
		15	—	4.0	—	6.75	4.0	—	4.0	
	"1" Level VIH	5.0	3.5	—	3.5	2.75	—	3.5	—	Vdc
		10	7.0	—	7.0	5.50	—	7.0	—	
		15	11	—	11	8.25	—	11	—	
Output Drive Current (V _{OH} = 2.5 Vdc) (V _{OH} = 4.6 Vdc) (V _{OH} = 9.5 Vdc) (V _{OH} = 13.5 Vdc)	Source IOH	5.0	-1.2	—	-1.0	-1.7	—	-0.7	—	mAdc
		10	-0.64	—	-0.51	-0.88	—	-0.36	—	
		15	-1.6	—	-1.3	-2.25	—	-0.9	—	
	Sink IOL	5.0	0.64	—	0.51	0.88	—	0.36	—	mAdc
		10	1.6	—	1.3	2.25	—	0.9	—	
		15	4.2	—	3.4	8.8	—	2.4	—	
Input Current	I _{in}	15	—	-0.1	—	0.00001	0.1	—	1.0	μAdc
Input Capacitance (V _{in} = 0)	C _{in}	—	—	—	—	5.0	7.5	—	—	pF
Quiescent Current (Per Package)	I _{DD}	5.0	—	5.0	—	0.005	5.0	—	150	μAdc
		10	—	10	—	0.010	10	—	300	
		15	—	20	—	0.015	20	—	600	
**Total Supply Current at an external load Capacitance (C _L) and at external timing capacitance (C _X). use the formula —	I _T	—	$I_T(C_L, C_X) = [(C_L + 0.36C_X)V_{DD}] + 2 \times 10^{-8} R_X C_X (V_{DD} - 2)^2 \times 10^{-3}$ where: I _T in μA (per circuit), C _L and C _X in pF, R _X in megohms, V _{DD} in Vdc, f in kHz is input frequency.						—	μAdc

*Data labelled "Typ" is not to be used for design purposes but is intended as an indication of the IC's potential performance.

**The formulas given are for the typical characteristics only at 25°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 range V_{SS} ≤ (V_{in} or V_{out}) ≤ V_{DD}. Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}). Unused outputs must be left open.

PIN ASSIGNMENT

V _{SS}	1	16	V _{DD}
C _{X1} /R _{X1}	2	15	V _{SS}
Reset	3	14	C _{X2} /R _{X2}
A1	4	13	Reset 2
B1	5	12	A2
O1	6	11	B2
O1	7	10	O2
V _{SS}	8	9	O2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC14528B

SWITCHING CHARACTERISTICS** (C_L = 50 pF, T_A = 25°C)

Characteristic	Symbol	C _X pF	R _X kΩ	V _{DD} Vdc	Min	Typ #	Max	Unit
Output Rise and Fall Time t _{TLH} , t _{FHL} = (1.5 ns/pF) C _L + 25 ns t _{TLH} , t _{FHL} = (0.75 ns/pF) C _L + 12.5 ns t _{TLH} , t _{FHL} = (0.55 ns/pF) C _L + 9.5 ns	t _{TLH} , t _{FHL}	—	—	5.0 10 15	— — —	100 50 40	200 100 80	ns
Turn-Off, Turn-On Delay Time — A or B to Q or Q̄ t _{PLH} , t _{PHL} = (1.7 ns/pF) C _L + 240 ns t _{PLH} , t _{PHL} = (0.66 ns/pF) C _L + 87 ns t _{PLH} , t _{PHL} = (0.5 ns/pF) C _L + 65 ns	t _{PLH} , t _{PHL}	15	5.0	5.0 10 15	— — —	325 120 90	650 240 180	ns
Turn-Off, Turn-On Delay Time — A or B to Q or Q̄ t _{PLH} , t _{PHL} = (1.7 ns/pF) C _L + 620 ns t _{PLH} , t _{PHL} = (0.66 ns/pF) C _L + 257 ns t _{PLH} , t _{PHL} = (0.5 ns/pF) C _L + 185 ns	t _{PLH} , t _{PHL}	1000	10	5.0 10 15	— — —	705 290 210	— — —	ns
Input Pulse Width — A or B	t _{WH}	15	5.0	5.0 10 15	150 75 55	70 30 30	— — —	ns
	t _{WL}	1000	10	5.0 10 15	— — —	70 30 30	— — —	ns
Output Pulse Width — Q or Q̄ (For C _X > 0.01 μF use graph for appropriate V _{DD} level)	t _W	15	5.0	5.0 10 15	— — —	550 350 300	— — —	ns
Output Pulse Width — Q or Q̄ (For C _X > 0.01 μF use formula: t _W = 0.2 R _X C _X Ln [(V _{DD} - V _{SS})]	t _W	10,000	10	5.0 10 15	15 10 15	30 50 55	45 90 95	μs
Pulse Width Match between Circuits in the same package	t ₁ - t ₂	10,000	10	5.0 10 15	— — —	6.0 8.0 8.0	25 35 35	%
Reset Propagation Delay — Reset to Q or Q̄	t _{PLH} , t _{PHL}	15	5.0	5.0	—	325	600	ns
				10	—	90 60	225 170	
		1000	10	5.0	—	1000	—	ns
				10	—	300	—	
		15	5.0	5.0	0	—	—	ns
				10	0	—	—	
		1000	10	5.0	0	—	—	ns
				10	0	—	—	
		—	—	5.0	—	—	1000	kΩ
				—	—	—	—	
External Timing Capacitance	C _X	—	—	—	—	No Limits†	—	μF

† R_X is in Ohms, C_X is in farads, V_{DD} and V_{SS} in volts, PW_{out} in seconds.
 * If C_X > 15 μF, Use Discharge Protection Diode D_X, per Fig. 9.

**The formulas given are for the typical characteristics only at 25°C

‡ Data labelled "Typ" is not to be used for design purposes but is intended as an indication of the IC's potential performance

FUNCTION TABLE

Reset	Inputs		Outputs	
	A	B	Q	Q̄
H				
H	L	H		
H		L	Not Triggered	Not Triggered
H	H		Not Triggered	Not Triggered
H	L, H,	H	Not Triggered	Not Triggered
H	L	L, H,	Not Triggered	Not Triggered
L	X	X	L	H
	X	X	Not Triggered	Not Triggered

MC14528B

FIGURE 1 - OUTPUT SOURCE CURRENT TEST CIRCUIT

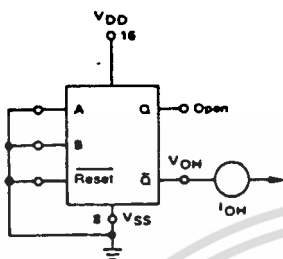


FIGURE 2 - OUTPUT SINK CURRENT TEST CIRCUIT

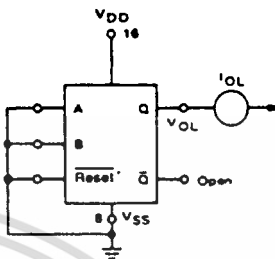


FIGURE 3 - POWER DISSIPATION TEST CIRCUIT AND WAVEFORMS

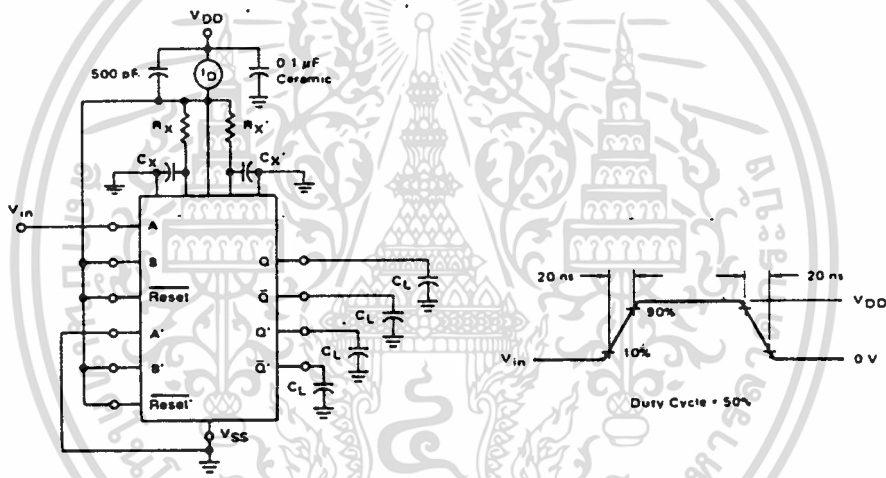
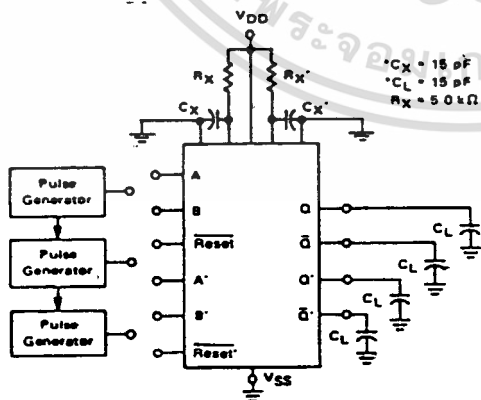


FIGURE 4 - AC TEST CIRCUIT

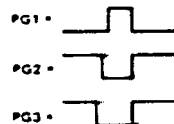


*C_X = 15 pF
*C_L = 15 pF
*R_X = 50 Ω

INPUT CONNECTIONS

CHARACTERISTICS	Reset	A	B
t _{PLH} , t _{PHL} , t _{TLH} , t _{THL} t _w	V _{DD}	PG1	V _{DD}
t _{PLH} , t _{PHL} , t _{TLH} , t _{THL} t _w	V _{DD}	V _{SS}	PG2
t _{PLH(R)} , t _{PHL(R)} , t _w	PG3	PG1	PG2

*Includes capacitance of probes, wiring, and fixture parasitic.
NOTE AC test waveforms for PG1, PG2, and PG3 on next page.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC14528B

FIGURE 5 - AC TEST WAVEFORMS

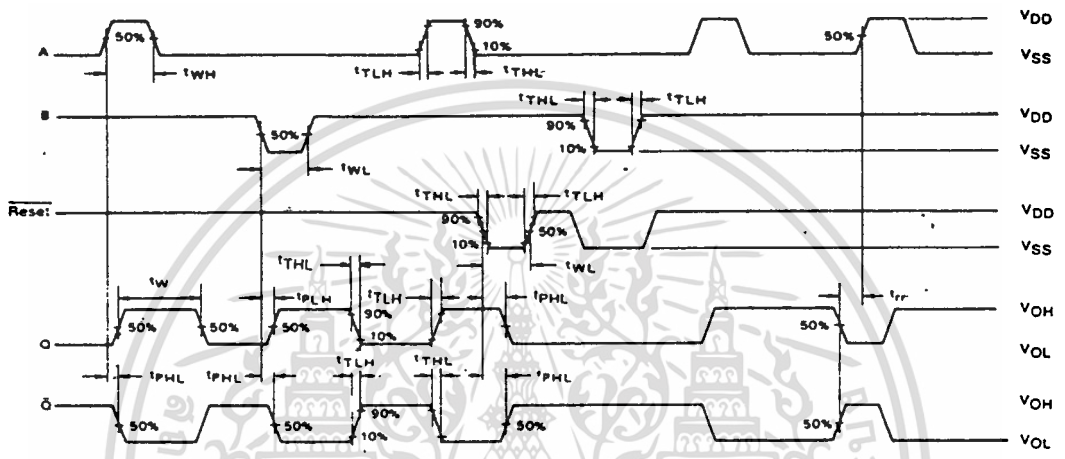
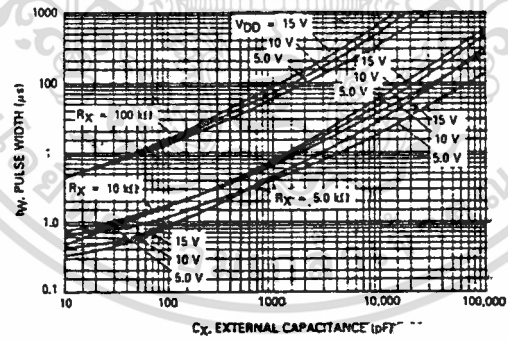


FIGURE 6 - PULSE WIDTH versus C_X



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC14528B

TYPICAL APPLICATIONS

FIGURE 7 — RETRIGGERABLE MONOSTABLES CIRCUITRY

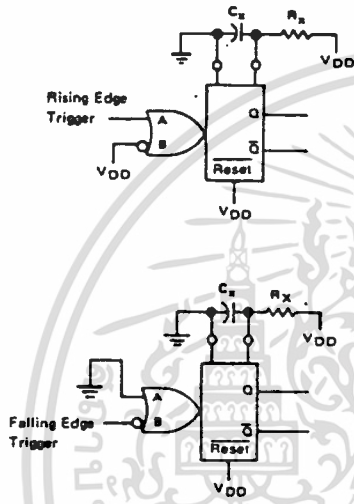


FIGURE 8 — NON-RETRIGGERABLE MONOSTABLES CIRCUITRY

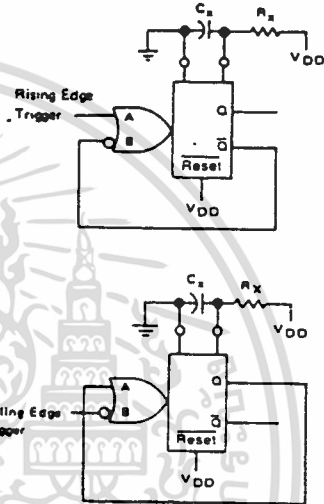


FIGURE 9 — USE OF A DIODE TO LIMIT POWER DOWN CURRENT SURGE

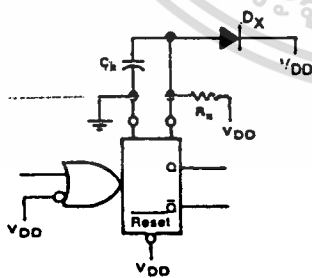
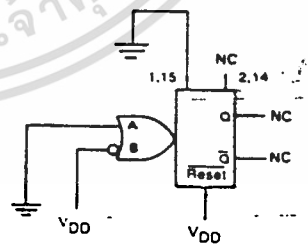


FIGURE 10 — CONNECTION OF UNUSED SECTIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**SG1525A/SG1527A
SG2525A/SG2527A
SG3525A/SG3527A**

PULSE WIDTH MODULATOR CONTROL CIRCUITS

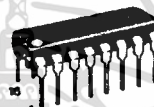
The SG1525A/1527A series of pulse width modulator control-circuits offer improved performance and lower external parts count when implemented for controlling all types of switching power supplies. The on-chip +5.1 volt reference is trimmed to $\pm 1\%$ and the error amplifier has an input common-mode voltage range that includes the reference voltage, thus eliminating the need for external divider resistors. A sync input to the oscillator enables multiple units to be slaved or a single unit to be synchronized to an external system clock. A wide range of dead time can be programmed by a single resistor connected between the CT and Discharge pins. These devices also feature built-in soft-start circuitry, requiring only an external timing capacitor. A shutdown pin controls both the soft-start circuitry and the output stages, providing instantaneous turn off through the PWM latch with pulsed shutdown, as well as soft-start recycle with longer shutdown commands. The under voltage lockout inhibits the outputs and the changing of the soft-start capacitor when VCC is below nominal. The output stages are totem-pole design capable of sinking and sourcing in excess of 200 mA. The output stage of the SG1525A series features NOR Logic resulting in a low output for an off state while the SG1527A series utilizes OR Logic which gives a high output when off. The devices are available in Military, Industrial and Commercial temperature ranges.

- 8.0 to 35 Volt Operation
- 5.1 Volt $\pm 1.0\%$ Trimmed Reference
- 100 Hz to 400 kHz Oscillator Range
- Separate Oscillator Sync Pin
- Adjustable Dead Time Control
- Input Undervoltage Lockout
- Latching PWM to Prevent Multiple Pulses
- Pulse-by-Pulse Shutdown
- Dual Source/Sink Outputs: ± 400 mA Peak

PULSE WIDTH MODULATOR CONTROL CIRCUITS

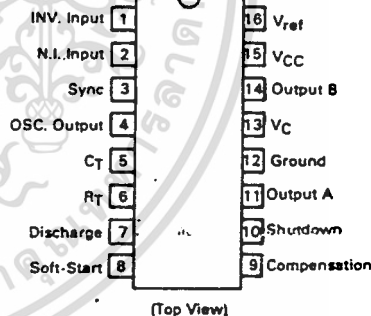
SILICON MONOLITHIC INTEGRATED CIRCUITS

J SUFFIX
CERAMIC PACKAGE
CASE 620-02

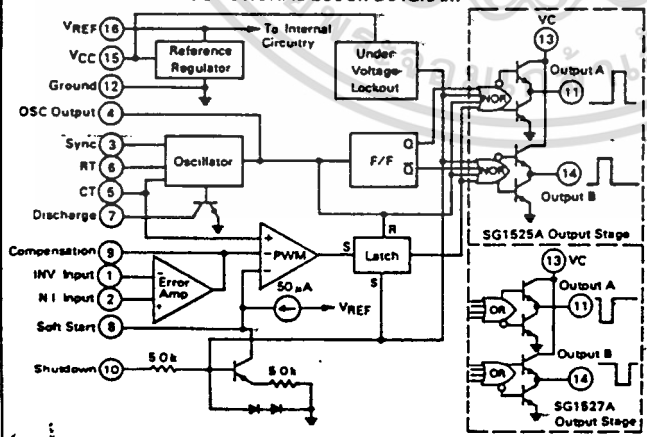


N SUFFIX
PLASTIC PACKAGE
CASE 648-05

PIN CONNECTIONS



FUNCTIONAL BLOCK DIAGRAM



ORDERING INFORMATION

Device	Temperature Range	Package
SG1525AJ	-65 to +125°C	Ceramic DIP
SG1527AJ	-65 to +125°C	Ceramic DIP
SG2525AJ	-25 to +85°C	Ceramic DIP
SG2525AN	-25 to +85°C	Plastic DIP
SG2527AJ	-25 to +85°C	Ceramic DIP
SG2527AN	-25 to +85°C	Plastic DIP
SG3525AJ	0 to +70°C	Ceramic DIP
SG3525AN	0 to +70°C	Plastic DIP
SG3527AJ	0 to +70°C	Ceramic DIP
SG3527AN	0 to +70°C	Plastic DIP

MOTOROLA LINEAR/INTERFACE DEVICES

4-193

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SG1525A, SG1527A, SG2525A, SG2527A, SG3525A, SG3527A

MAXIMUM RATINGS (Note 1)

Rating	Symbol	Value	Unit
Supply Voltage	V _{CC}	+40	V _{dc}
Collector Supply Voltage	V _C	+40	V _{dc}
Logic Inputs	—	-0.3 to +5.5	V
Analog Inputs	—	-0.3 to V _{CC}	V
Output Current, Source or Sink	I _O	±500	mA
Reference Output Current	I _{ref}	50	mA
Oscillator Charging Current	—	5.0	mA
Power Dissipation (Plastic & Ceramic Package) Note 2, T _A = +25°C Note 3, T _C = +25°C	P _D	1000 2000	mW
Thermal Resistance Junction to Air Plastic and Ceramic Package	R _{θJA}	100	°C/W
Thermal Resistance Junction to Case Plastic and Ceramic Package	R _{θJC}	60	°C/W
Operating Junction Temperature	T _J	+150	°C
Storage Temperature Range Ceramic Package Plastic Package	T _{stg}	-65 to +150 -55 to +125	°C
Lead Temperature (Soldering, 10 Seconds)	T _{Solder}	+300	°C

NOTES:

1. Values beyond which damage may occur
2. Derate at 10 mW/°C for ambient temperatures above +50°C
3. Derate at 16 mW/°C for case temperatures above +25°C

RECOMMENDED OPERATING CONDITIONS

Characteristic	Symbol	Min.	Max.	Unit
Supply Voltage	V _{CC}	+8.0	+35	V _{dc}
Collector Supply Voltage	V _C	+4.5	+35	V _{dc}
Output Sink/Source Current (Steady State) (Peak)	I _O	0 0	±100 ±400	mA
Reference Load Current	I _{ref}	0	20	mA
Oscillator Frequency Range	f _{osc}	0.1	400	kHz
Oscillator Timing Resistor	R _T	2.0	150	kΩ
Oscillator Timing Capacitor	C _T	0.001	0.2	μF
Deadtime Resistor Range	R _D	0	500	Ω
Operating Ambient Temperature Range SG1525A, SG1527A SG2525A, SG2527A SG3525A, SG3527A	T _A	-65 -25 0	+125 +85 +70	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SG1525A, SG1527A, SG2525A, SG2527A, SG3525A, SG3527A
ELECTRICAL CHARACTERISTICS ($V_{CC} = +20 \text{ Vdc}$, $T_A = T_{\text{low}}$ to T_{high} (Note 4), unless otherwise specified)

Characteristic	Symbol	SG1525A/2525A SG1527A/2527A			SG3525A SG3527A			Unit
		Min	Typ	Max	Min	Typ	Max	
REFERENCE SECTION								
Reference Output Voltage ($T_J = +25^\circ\text{C}$)	V_{ref}	5.05	5.10	5.15	5.00	5.10	5.20	Vdc
Line Regulation ($+8.0 \text{ V} \leq V_{CC} \leq +35 \text{ V}$)	$R_{regline}$	—	10	20	—	10	20	mV
Load Regulation ($0 \text{ mA} \leq I_L \leq 20 \text{ mA}$)	$R_{regload}$	—	20	50	—	20	50	mV
Temperature Stability	$\Delta V_{ref}/\Delta T$	—	20	50	—	20	50	mV
Total Output Variation Includes Line and Load Regulation over Temperature	ΔV_{ref}	5.00	—	5.20	4.95	—	5.25	Vdc
Short Circuit Current ($V_{ref} = 0 \text{ V}$, $T_J = +25^\circ\text{C}$)	I_{SC}	—	80	100	—	80	100	mA
Output Noise Voltage ($10 \text{ Hz} \leq f \leq 10 \text{ kHz}$, $T_J = +25^\circ\text{C}$)	V_n	—	40	200	—	40	200	μV_{rms}
Long Term Stability ($T_J = +125^\circ\text{C}$) (Note 5)	S	—	20	50	—	20	50	mV/kyr
OSCILLATOR SECTION (Note 6, unless otherwise specified)								
Initial Accuracy ($T_J = +25^\circ\text{C}$)	—	—	± 2.0	± 6.0	—	± 2.0	± 6.0	%
Frequency Stability with Voltage ($+8.0 \text{ V} \leq V_{CC} \leq +35 \text{ V}$)	$\frac{\Delta f_{osc}}{\Delta V_{CC}}$	—	± 0.3	± 1.0	—	± 1.0	± 2.0	%
Frequency Stability with Temperature	$\frac{\Delta f_{osc}}{\Delta T}$	—	± 3.0	± 6.0	—	± 3.0	± 6.0	%
Minimum Frequency ($R_T = 150 \text{ k}\Omega$, $C_T = 0.2 \mu\text{F}$)	f_{min}	—	—	100	—	—	100	Hz
Maximum Frequency ($R_T = 2.0 \text{ k}\Omega$, $C_T = 1.0 \text{ nF}$)	f_{max}	400	—	—	400	—	—	kHz
Current Mirror ($I_{RT} = 2.0 \text{ mA}$)	—	1.7	2.0	2.2	1.7	2.0	2.2	mA
Clock Amplitude	—	3.0	3.5	—	3.0	3.5	—	V
Clock Width ($T_J = +25^\circ\text{C}$)	—	0.3	0.5	1.0	0.3	0.5	1.0	μs
Sync Threshold	—	1.2	2.0	2.8	1.2	2.0	2.8	V
Sync Input Current (Sync Voltage = +3.5 V)	—	—	1.0	2.5	—	1.0	2.5	mA
ERROR AMPLIFIER SECTION ($V_{CM} = +5.1 \text{ V}$)								
Input Offset Voltage	V_{IO}	—	0.5	5.0	—	2.0	10	mV
Input Bias Current	I_{IB}	—	1.0	10	—	1.0	10	μA
Input Offset Current	I_{IO}	—	—	1.0	—	—	1.0	μA
DC Open Loop Gain ($R_L \geq 10 \text{ M}\Omega$)	A_{VOL}	60	75	—	60	75	—	dB
Low Level Output Voltage	V_{OL}	—	0.2	0.5	—	0.2	0.5	V
High Level Output Voltage	V_{OH}	3.8	5.6	—	3.8	5.6	—	V
Common Mode Rejection Ratio ($+1.5 \text{ V} \leq V_{CM} \leq +5.2 \text{ V}$)	CMRR	60	75	—	60	75	—	dB
Power Supply Rejection Ratio ($+8.0 \text{ V} \leq V_{CC} \leq +35 \text{ V}$)	PSRR	50	60	—	50	60	—	dB
PWM COMPARATOR SECTION								
Minimum Duty Cycle	DC_{min}	—	—	0	—	—	0	%
Maximum Duty Cycle	DC_{max}	45	49	—	45	49	—	%
Input Threshold, Zero Duty Cycle (Note 6)	V_{TH}	0.6	0.9	—	0.6	0.9	—	V
Input Threshold, Maximum Duty Cycle (Note 6)	V_{TH}	—	3.3	3.6	—	3.3	3.6	V
Input Bias Current	I_{IB}	—	0.05	1.0	—	0.05	1.0	μA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SG1525A, SG1527A, SG2525A, SG2527A, SG3525A, SG3527A

ELECTRICAL CHARACTERISTICS (Continued)

Characteristic	Symbol	SG1525A/2526A SG1527A/2527A			SG3525A SG3527A			Unit
		Min	Typ	Max	Min	Typ	Max	
SOFT-START SECTION								
Soft-Start Current ($V_{\text{shutdown}} = 0 \text{ V}$)	—	25	50	80	25	50	80	μA
Soft-Start Voltage ($V_{\text{shutdown}} = 2.0 \text{ V}$)	—	—	0.4	0.6	—	0.4	0.6	V
Shutdown Input Current ($V_{\text{shutdown}} = 2.5 \text{ V}$)	—	—	0.4	1.0	—	0.4	1.0	mA
OUTPUT DRIVERS (Each Output, $V_{\text{CC}} = +20 \text{ V}$)								
Output Low Level ($I_{\text{sink}} = 20 \text{ mA}$) ($I_{\text{sink}} = 100 \text{ mA}$)	V_{OL}	— —	0.2 1.0	0.4 2.0	— —	0.2 1.0	0.4 2.0	V
Output High Level ($I_{\text{source}} = 20 \text{ mA}$) ($I_{\text{source}} = 100 \text{ mA}$)	V_{OH}	18 17	19 18	— —	18 17	19 18	— —	V
Under Voltage Lockout (V_8 and $V_9 = \text{High}$)	V_{UL}	6.0	7.0	8.0	6.0	7.0	8.0	V
Collector Leakage, $V_{\text{C}} = +35 \text{ V}$ (Note 7)	$I_{\text{C(leak)}}$	—	—	200	—	—	200	μA
Rise Time ($C_{\text{L}} = 1.0 \text{ nF}$, $T_{\text{J}} = 25^\circ\text{C}$)	t_{r}	—	100	600	—	100	600	ns
Fall Time ($C_{\text{L}} = 1.0 \text{ nF}$, $T_{\text{J}} = 25^\circ\text{C}$)	t_{f}	—	50	300	—	50	300	ns
Shutdown Delay ($V_{\text{SD}} = +3.0 \text{ V}$, $C_{\text{S}} = 0$, $T_{\text{J}} = +25^\circ\text{C}$)	t_{ds}	—	0.2	0.5	—	0.2	0.5	μs
Supply Current, ($V_{\text{CC}} = +35 \text{ V}$)	I_{CC}	—	14	20	—	14	20	mA

NOTES:

- $T_{\text{low}} = -55^\circ\text{C}$ for SG1525A/1527A
 -25°C for SG2525A/2527A
 0°C for SG3525A/3527A
- $T_{\text{high}} = +125^\circ\text{C}$ for SG1525A/1527A
 $+85^\circ\text{C}$ for SG2525A/2527A
 $+70^\circ\text{C}$ for SG3525A/3527A
- Since long term stability cannot be measured on each device before shipment, this specification is an engineering estimate of average stability from lot to lot.
- Tested at $f_{\text{osc}} = 40 \text{ kHz}$ ($R_{\text{T}} = 1.6 \text{ k}\Omega$, $C_{\text{T}} = 0.01 \mu\text{F}$, $R_{\text{D}} = 0 \Omega$).
- Applies to SG1525A/2525A/3525A only, due to polarity of output pulses.

APPLICATION INFORMATION

SHUTDOWN OPTIONS

(See Block Diagram, front page)

Since both the compensation and soft-start terminals (Pins 9 and 8) have current source pull-ups, either can readily accept a pull-down signal which only has to sink a maximum of $100 \mu\text{A}$ to turn off the outputs. This is subject to the added requirement of discharging whatever external capacitance may be attached to these pins.

An alternate approach is the use of the shutdown circuitry of Pin 10 which has been improved to enhance the available shutdown options. Activating this circuit by applying a positive signal on Pin 10 performs two

functions: the PWM latch is immediately set providing the fastest turn-off signal to the outputs; and a $150 \mu\text{A}$ current sink begins to discharge the external soft-start capacitor. If the shutdown command is short, the PWM signal is terminated without significant discharge of the soft-start capacitor, thus, allowing, for example, a convenient implementation of pulse-by-pulse current limiting. Holding Pin 10 high for a longer duration, however, will ultimately discharge this external capacitor, recycling slow turn-on upon release.

Pin 10 should not be left floating as noise pickup could conceivably interrupt normal operation.

SG1525A, SG1527A, SG2525A, SG2527A, SG3525A, SG3527A

TYPICAL CHARACTERISTICS

FIGURE 1 — SG1525A OSCILLATOR SCHEMATIC

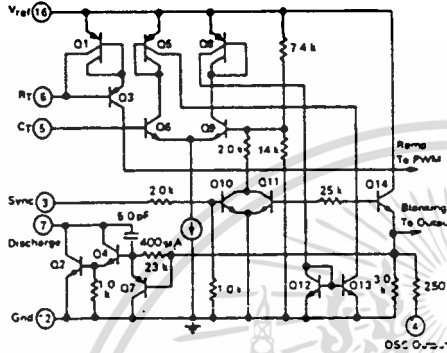


FIGURE 2 — OSCILLATOR CHARGE TIME versus RT

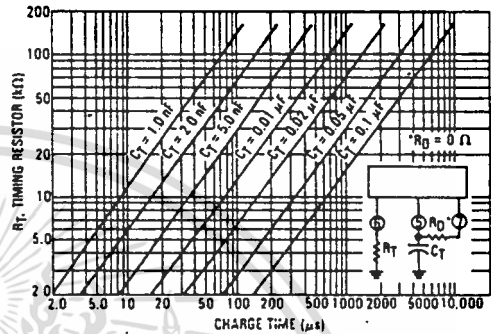


FIGURE 3 — OSCILLATOR DISCHARGE TIME versus RD

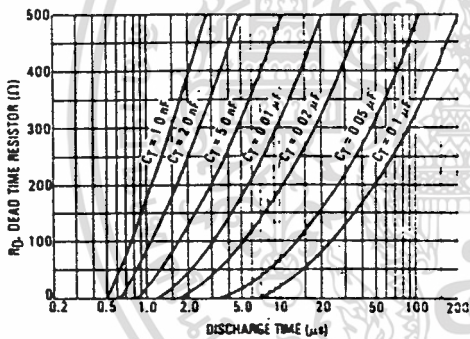


FIGURE 4 — SG1525A ERROR AMPLIFIER SCHEMATIC

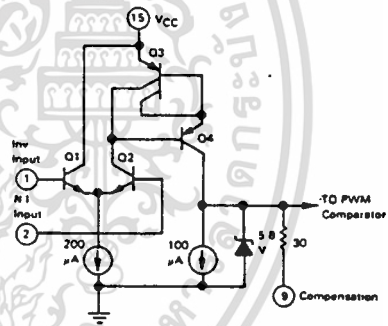


FIGURE 5 — ERROR AMPLIFIER OPEN-LOOP FREQUENCY RESPONSE

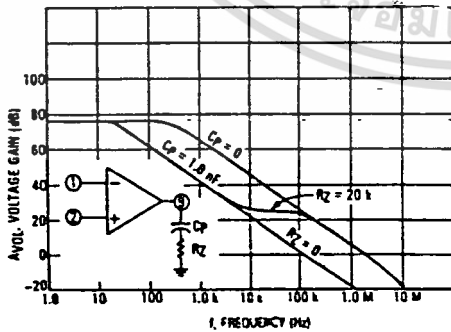
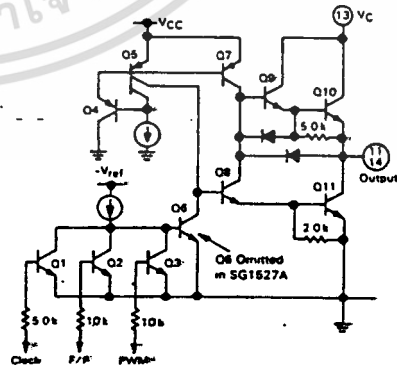


FIGURE 6 — SG1525A OUTPUT CIRCUIT (1/2 CIRCUIT SHOWN)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SG1525A, SG1527A, SG2525A, SG2527A, SG3525A, SG3527A

**FIGURE 7 — SG1525A/2525A/3525A
OUTPUT SATURATION CHARACTERISTICS**

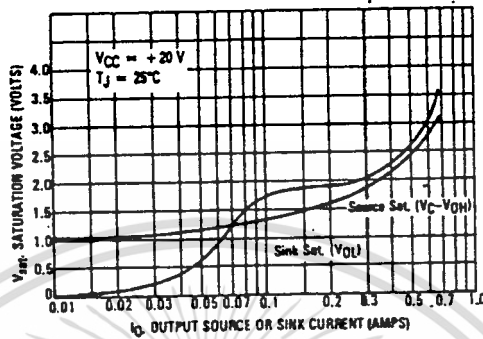
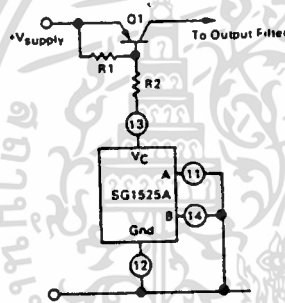
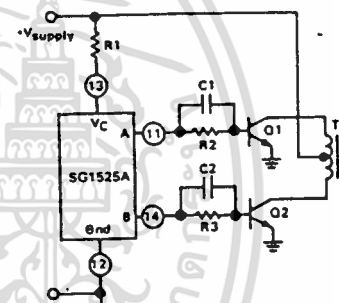


FIGURE 8 — SINGLE ENDED SUPPLY



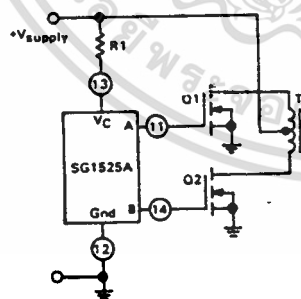
For single-ended supplies, the driver outputs are grounded. The Vc terminal is switched to ground by the totem-pole source transistors on alternate oscillator cycles.

FIGURE 9 — PUSH-PULL CONFIGURATION



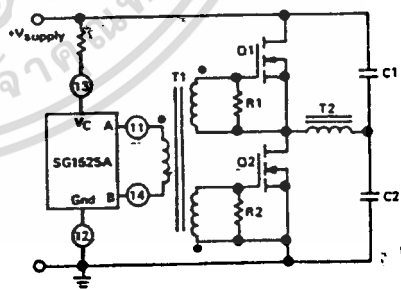
In conventional push-pull bipolar designs, forward base drive is controlled by R1-R3. Rapid turn-off times for the power devices are achieved with speed-up capacitors C1 and C2.

FIGURE 10 — DRIVING POWER FETS



The low source impedance of the output drivers provides rapid charging of power FET input capacitance while minimizing external components.

FIGURE 11 — DRIVING TRANSFORMERS IN A HALF-BRIDGE CONFIGURATION

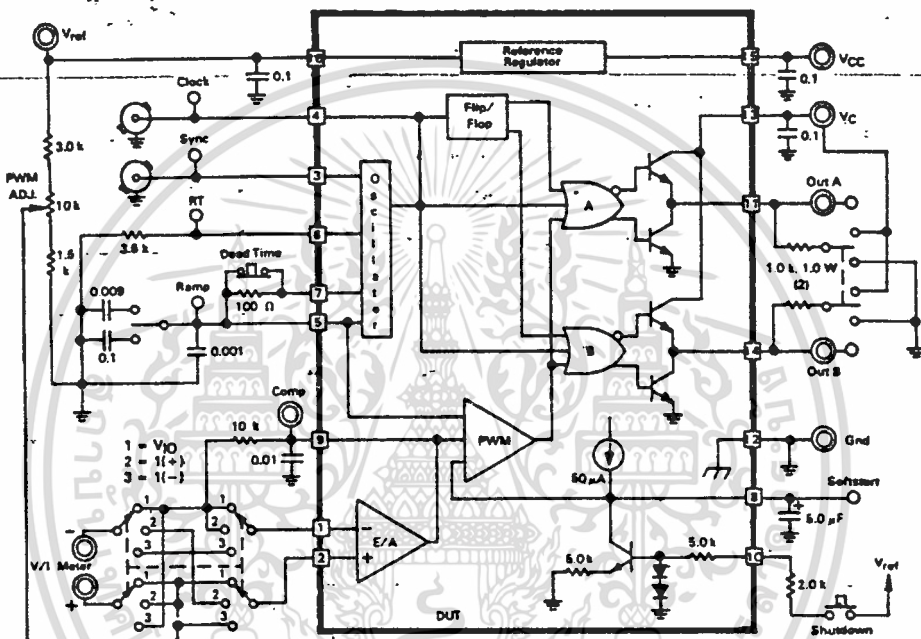


Low power transformers can be driven directly by the SG1525A. Automatic reset occurs during deadtime, when both ends of the primary winding are switched to ground.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SG1525A, SG1527A, SG2525A, SG2527A, SG3525A, SG3527A

FIGURE 12 -- LAB TEST FIXTURE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



LM741/LM741A/LM741C/LM741E Operational Amplifier

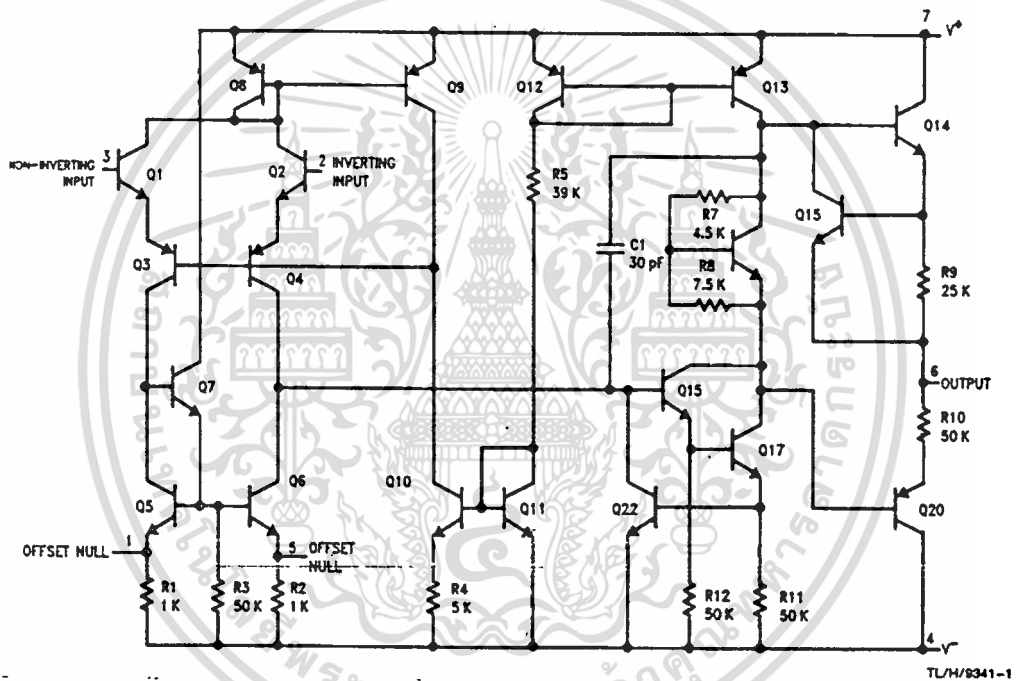
General Description

The LM741 series are general purpose operational amplifiers which feature improved performance over industry standards like the LM709. They are direct, plug-in replacements for the 709C, LM201, MC1439 and 748 in most applications. The amplifiers offer many features which make their application nearly foolproof: overload protection on the input and

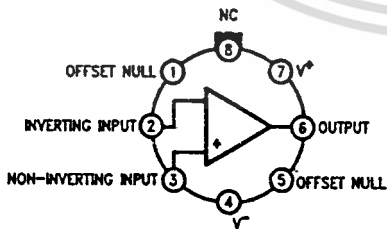
output, no latch-up when the common mode range is exceeded, as well as freedom from oscillations.

The LM741C/LM741E are identical to the LM741/LM741A except that the LM741C/LM741E have their performance guaranteed over a 0°C to +70°C temperature range, instead of -55°C to +125°C.

Schematic and Connection Diagrams (Top Views)



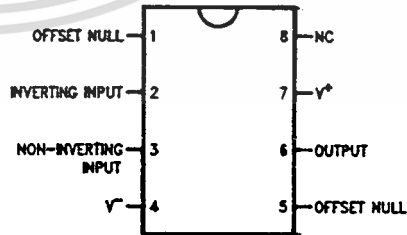
Metal Can Package



Order Number LM741H, LM741AH,
LM741CH or LM741EH
See NS Package Number H08C

TL/H/8341-2

Dual-In-Line or S.O. Package



Order Number LM741CJ, LM741CM,
LM741CN or LM741EN
See NS Package Numbers J08A, M08A or N08E

TL/H/8341-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Note 3) (Continued)

Parameter	Conditions	LM741A/LM741E			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Voltage Swing	$V_S = \pm 20V$ $R_L \geq 10\text{ k}\Omega$ $R_L \geq 2\text{ k}\Omega$	± 16									V V
	$V_S = \pm 15V$ $R_L \geq 10\text{ k}\Omega$ $R_L \geq 2\text{ k}\Omega$				± 12 ± 10	± 14 ± 13		± 12 ± 10	± 14 ± 13		V V
Short Circuit Current	$T_A = 25^\circ\text{C}$ $T_{AMIN} \leq T_A \leq T_{AMAX}$	10 10	25	35 40		25			25		mA mA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$ $R_S \leq 10\text{ k}\Omega$, $V_{CM} = \pm 12V$ $R_S \leq 50\text{ k}\Omega$, $V_{CM} = \pm 12V$					70	90		70	90	
Voltage Rejection	$T_{AMIN} \leq T_A \leq T_{AMAX}$ $V_S = \pm 20V$ to $V_S = \pm 5V$ $R_S \leq 50\text{ k}\Omega$ $R_S \leq 10\text{ k}\Omega$		86	96							dB dB
	$T_A = 25^\circ\text{C}$, Unity Gain										
Slew Rate	$T_A = 25^\circ\text{C}$, Unity Gain		0.25	0.8		0.3			0.3		μs %
			6.0	20		5			5		
Bandwidth (Note 4)	$T_A = 25^\circ\text{C}$	0.437	1.5								MHz
Slew Rate	$T_A = 25^\circ\text{C}$, Unity Gain	0.3	0.7			0.5			0.5		V/ μs
Supply Current	$T_A = 25^\circ\text{C}$					1.7	2.8		1.7	2.8	mA
Power Consumption	$T_A = 25^\circ\text{C}$ $V_S = \pm 20V$ $V_S = \pm 15V$			80	150						mW mW
	$V_S = \pm 20V$ $T_A = T_{AMIN}$ $T_A = T_{AMAX}$				165						mW mW
LM741E	$V_S = \pm 20V$ $T_A = T_{AMIN}$ $T_A = T_{AMAX}$				150						mW mW
LM741	$V_S = \pm 15V$ $T_A = T_{AMIN}$ $T_A = T_{AMAX}$					60	100				mW mW

Note 1: For operation at elevated temperatures, these devices must be derated based on thermal resistance, and T_J max. (listed under "Absolute-Maximum Ratings"). $T_J = T_A + (\theta_{JA} P_{D})$.

Thermal Resistance	Cardip (J)	DIP (N)	TO-5 (H)	SO-8 (M)
θ_{JA} (Junction to Ambient)	100°C/W	100°C/W	150°C/W	195°C/W
θ_{JC} (Junction to Case)	N/A	N/A	80°C/W	N/A

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

Note 3: Unless otherwise specified, these specifications apply for $V_S = \pm 15V$, $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$ (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$.

Note 4: Calculated value from: $\text{BW (MHz)} = 0.35/\text{Rise Time}(\mu\text{s})$.

Note 5: For military specifications see RETS741X for LM741 and RETS741AX for LM741A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributor for availability and specifications.
(Note 5)

	LM741A	LM741E	LM741	LM741C
Supply Voltage	±22V	±22V	±22V	±18V
Power Dissipation (Note 1)	500 mW	500 mW	500 mW	500 mW
Differential Input Voltage	±30V	±30V	±30V	±30V
Input Voltage (Note 2)	±15V	±15V	±15V	±15V
Output Short Circuit Duration	Indefinite	Indefinite	Indefinite	Indefinite
Operating Temperature Range	-55°C to +125°C	0°C to +70°C	-55°C to +125°C	0°C to +70°C
Storage Temperature Range	-65°C to +150°C	-65°C to +150°C	-65°C to +150°C	-65°C to +150°C
Junction Temperature	150°C	100°C	150°C	100°C
Soldering Information				
N-Package (10 seconds)	260°C	260°C	260°C	260°C
J- or H-Package (10 seconds)	300°C	300°C	300°C	300°C
M-Package				
Vapor Phase (60 seconds)	215°C	215°C	215°C	215°C
Infrared (15 seconds)	215°C	215°C	215°C	215°C

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

Electrical Characteristics (Note 3)

Parameter	Conditions	LM741A/LM741E			LM741			LM741C		
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
Input Offset Voltage	$T_A = 25^\circ\text{C}$ $R_S \leq 10\text{ k}\Omega$ $R_S \leq 50\Omega$		0.8	3.0		1.0	5.0		2.0	6.0
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ $R_S \leq 50\Omega$ $R_S \leq 10\text{ k}\Omega$			4.0			6.0			7.5
Average Input Offset Voltage Drift				15						
Input Offset Voltage Adjustment Range	$T_A = 25^\circ\text{C}$, $V_S = \pm 20\text{V}$	±10				±15			±15	
Input Offset Current	$T_A = 25^\circ\text{C}$		3.0	30		20	200		20	200
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$			70		85	500			300
Average Input Offset Current Drift				0.5						
Input Bias Current	$T_A = 25^\circ\text{C}$		30	80		80	500		80	500
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$			0.210			1.5			0.8
Input Resistance	$T_A = 25^\circ\text{C}$, $V_S = \pm 20\text{V}$	1.0	6.0		0.3	2.0		0.3	2.0	
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$, $V_S = \pm 20\text{V}$	0.5								
Input Voltage Range	$T_A = 25^\circ\text{C}$							±12	±13	
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$					±12	±13			
Large Signal Voltage Gain	$T_A = 25^\circ\text{C}$, $R_L \geq 2\text{ k}\Omega$ $V_S = \pm 20\text{V}$, $V_O = \pm 15\text{V}$ $V_S = \pm 15\text{V}$, $V_O = \pm 10\text{V}$	50				50	200		20	200
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$, $R_L \geq 2\text{ k}\Omega$, $V_S = \pm 20\text{V}$, $V_O = \pm 15\text{V}$ $V_S = \pm 15\text{V}$, $V_O = \pm 10\text{V}$	32								
	$V_S = \pm 5\text{V}$, $V_O = \pm 2\text{V}$	10				25			15	

กิตติกรรมประกาศ

ขอขอบคุณผู้ให้คำปรึกษา แนะนำ ตลอดทั้งความช่วยเหลือ ดังต่อไปนี้

- | | |
|---------------------------------|---|
| 1. อาจารย์พลศาสตร์ เลิศประเสริฐ | อาจารย์ที่ปรึกษา ภาควิชาอิเล็กทรอนิกส์ |
| 2. นายอิทธิพล พจนสัง | นักศึกษาปริญญาโท ภาควิชาอิเล็กทรอนิกส์ |
| 3. นายกมล เกี้ยวศรีกุล | บัณฑิตรุ่น 29 ภาควิชาวิศวกรรมระบบควบคุม |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. DON LANCASTER, "TTL COOKBOOK", HOWARD W. SAMS & CO., INC.
2. JACOB MILLMAN, ARVIN GRABEL, "MICROELECTRONICS", McGRAW-HILL INC.
3. MOTOROLA, "CMOS APPLICATION-SPECIFIC STANDARD ICs", MOTOROLA INC. ,1991
4. MOTOROLA, "LINEAR AND INTERFACE INTEGRATED CIRCUITS", MOTOROLA INC. ,1985
5. SAVANT, RODEN, CARPENTER, "ELECTRONIC DESIGN CIRCUITS AND SYSTEM", THE BENJAMIN/CUMMINGS PUBLISHING COMPANY, LTD.
6. STEPHEN R. FLEEMAN, "ELECTRONIC DEVICES", PRENTICE HALL INC.
7. ชูชัย ชนสารตั้งเจริญ, ดนัย แสงสุริยศิลป์, ทินกร ดูก, ชงชัย อุดมกิจโกศล, ธานีินทร์ ถาวรศาสนวงศ์, "การใช้งาน Z80", ชุจก. สำนักพิมพ์ Physics Center
8. ประเมษฐ์ ประยานันท์, ปิยพงษ์ เผ่าวนิช, "คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51" สำนักพิมพ์ซีเอ็ด
9. ผศ. พิพัฒน์ เลาทสงคราม, "การทดลองไมโครคอนโทรลเลอร์ MCS-51"
10. สุเจตน์ จันทพงษ์, "ไมโครคอนโทรลเลอร์ชิปเดี่ยว 8051", โครงการตำราวิชาการวิทยาลัยมหานคร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้