B

ปริญญานิพนธ์

เครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

AUTOMATIC PAGER CALL TONE & VOICE SYSTEM

นายกังวาล สุรินทราบูรณ์

นายโชติ สุภาการ

นางสาวทิพวัลย์ สุวรรณชมพู

นายสุพล สืบจากดี

A021052

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชา ครุศาสตร์วิศวกรรม

คณะ ครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เลขหมู่.....

เลขทะเบียน 1284

วัน เดือน ปี -3 พ.ย. 2538

021052

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อปริญญานิพนธ์ เครื่องเรียกวิทยุติดตามตัวระบบ โทนและวอยซ์ แบบอัตโนมัติ

AUTOMATIC PAGER CALL TONE & VOICE SYSTEM

ชื่อนักศึกษา

1. นายกังวาล	สุรินทราบูรณ์	รหัสประจำตัว	36031401
2. นายโชติ	สุภาการ	รหัสประจำตัว	36031410
3. นางสาวทิพวัลย์	สุวรรณชมพู	รหัสประจำตัว	36031411
4. นายสุพล	สืบจากดี	รหัสประจำตัว	36031431

อาจารย์ผู้ควบคุมปริญญานิพนธ์

1. อาจารย์กิติพงค์ มะโน
2. อาจารย์วรวิทย์ สมหา
3. อาจารย์สันติ ตันตระกูล

คณะกรรมการสอบปริญญานิพนธ์		ลายเซ็น
1.อาจารย์กิติพงค์	มะโน	.....
2.อาจารย์วรวิทย์	สมหา	.....
3.อาจารย์สันติ	ตันตระกูล	.....
4.อาจารย์พีระวุฒิ	สุวรรณจันทร์	.....
5.อาจารย์สุชิน	อาจหาญ	.....

วัน/เดือน/ปี ที่สอบ 22 ธันวาคม 2537 **เวลา** 13.00 น. **ถึง** 15.00 น.

สถานที่สอบ ห้อง ค.301 คณะครุศาสตร์อุตสาหกรรม

.....

(ผศ.ดร.ธีระพล [illegible])

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

AUTOMATIC PAGER CALL TONE & VOICE SYSTEM

ผู้จัดทำ

นายกังวาล สุรินทราบูรณ์

นายโชติ สุภาการ

นางสาวทิพวัลย์ สุวรรณชมพู

นายสุพล สืบจากดี

อาจารย์ที่ปรึกษา

ลงนาม.....

(อาจารย์กิติพงค์ มะโน)

ลงนาม.....

(อาจารย์วรวิทย์ สมหา)

ลงนาม.....

(อาจารย์สันติ ตันตระกูล)

หัวหน้าภาควิชา

ลงนาม.....

(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องเรียกวิทยุติดตามตัวระบบ โทนและวอยซ์แบบอัตโนมัติ

AUTOMATIC PAGER CALL TONE & VOICE SYSTEM

จุดประสงค์

1. เพื่อศึกษาการนำระบบคอมพิวเตอร์มาใช้ในการสื่อสารโทรคมนาคม
2. เพื่อสร้างเครื่องเรียกวิทยุติดตามตัวระบบเสียงพูดอัตโนมัติ (ต้นแบบ)
3. เพื่อศึกษาการเขียนโปรแกรมควบคุมการทำงานทางด้าน Micro Controller
4. เพื่อนำเครื่องต้นแบบมาสร้างเป็นเครื่องใช้งานจริง ในบริการของการสื่อสารแห่งประเทศไทย

ประโยชน์ที่คาดว่าจะได้รับ

1. ทราบการสร้างสัญญาณ Encoder แบบ Two-Tone ในการส่งออกอากาศ
2. ทราบหลักการของการเขียนโปรแกรมควบคุมการทำงาน Micro Controller
3. ทราบถึงการนำวงจร DAC มาใช้งาน
4. ทราบถึงการนำวงจรวิเคราะห์เสียงพูด มาประยุกต์ใช้งาน
5. ทราบถึงการต่อวงจรอินเตอร์เฟสเข้ากับคู่สายโทรศัพท์
6. การสื่อสารแห่งประเทศไทย สามารถนำไปใช้งานในบริการของการสื่อสารแห่งประเทศไทย
7. เป็นเครื่องต้นแบบที่ใช้ในการศึกษา การนำเอาระบบคอมพิวเตอร์มาประยุกต์ใช้ในการสื่อสารโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

AUTOMATIC PAGER CALL TONE & VOICE SYSTEM

นายกังวาล สุรินทราบูรณ์

นายโชติ สุภาการ

นางสาวทิพวัลย์ สุวรรณชมพู

นายสุพล สืบจากดี

อาจารย์ที่ปรึกษา

อาจารย์ กิติพงศ์ มะโน

อาจารย์ วรวิทย์ สมหา

อาจารย์ สันติ ตันตระกูล

ปีการศึกษา 2537

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการนำเสนอเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ โดยใช้ไมโครโปรเซสเซอร์เป็นอุปกรณ์ควบคุมระบบ ซึ่งเป็นเครื่องต้นแบบที่จะนำไปสร้างใช้งานในบริการของการสื่อสารแห่งประเทศไทยต่อไป การพัฒนาโปรแกรมที่ใช้กับเครื่องนี้ ใช้ภาษาซี และภาษาแอสเซมบลี้ ในการพัฒนาโดยการใช้โปรแกรม AVC51 ในการ COMPILE และ LINK ซึ่งโปรแกรมที่เขียนส่วนใหญ่จะใช้ภาษาซี จะมีเพียงบางส่วนที่ต้องการความเร็วในการทำงานสูงๆ จะใช้ภาษาแอสเซมบลี้ การพัฒนาเครื่องนี้ขึ้นมาเพื่อนำไปแทนระบบเก่าที่ยังใช้บุคคลากรในการทำงานอยู่ และเครื่องนี้ ยังคงสามารถทำงานในระบบแบบเก่าได้ด้วย (ระบบที่ใช้พนักงานเป็นโอเปอร์เรเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC PAGER CALL TONE & VOICE SYSTEM

MR. KANGWAN SURINTARABOOL
MR. CHOTE SUPAKARN
Miss. TIPPAWAN SUWANCHOMPU
MR. SUPHOL SUEBJARKDEE

ADVISOR

MR. KITIPONG MANO
MR. WORAWIT SOMHA
MR. SUNTI TUNTRAKOOL

1994

ABSTRACT

THIS THESIS PRESENT AN AUTOMATIC PAGER CALL TONE & VOICE SYSTEM THAT CONTROLED BY MICROPROCESSOR. IT CAN USED ORIGINAL MACHINE IN ORDER TO APPLY AND SERVICE WORK OF COMMUNICATION AUTHORITY OF THAILAND (CAT.) WE USE C AND ASSEMBLY LANGUAGE FOR PROGRAM APPLICATION, COMPILE AND LINK BY USED AVC 51. GENERAL WRITEN BY C LANGUAGE, SOMETIME THIS PROGRAM WANT TO VERY QUICK SPEED SO WE USE ASSEMBLY LANGUAGE, THIS MACHINE APPLICATION FOR SERVE IN PLACE OF OLD SYSTEM WHICH PASS EMPLOYEE AND IT CAN WORK IN OLD SYSTEM (MANUAL OPERATOR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ ฉบับนี้สำเร็จลุล่วงไปด้วยดี ก็ด้วยความช่วยเหลือในการให้คำแนะนำ คำปรึกษาตลอดจนอุปกรณ์ เครื่องมือ เครื่องใช้ และสถานที่ต่างๆในการปฏิบัติงานจาก คณาจารย์ ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน โดยเฉพาะอย่างยิ่ง อ. วรวิทย์ สมหา สำนักวิจัยและบริการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การสื่อสารแห่งประเทศไทย รวมไปถึงเพื่อน ๆ ครุ.วศ.รุ่นที่ 15 และ พี่ๆในสำนักวิจัยวิจัยและบริการคอมพิวเตอร์ทุกท่านที่ให้ความช่วยเหลือ ร่วมมือ ร่วมใจ และขอขอบคุณ บิดา มารดา ผู้ซึ่งให้กำเนิดและสนับสนุนทางด้านกำลังใจและกำลังทรัพย์ จึงขอขอบพระคุณทุกท่านที่กล่าวมาข้างต้น ด้วยความจริงใจอย่างสุดซึ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 เคลค้ามอดูเลชั่น	2
2.2 อุปกรณ์ที่ใช้	8
2.2.1 โฟโต้ทรานซิสเตอร์	8
2.2.2 OPTOCOUPLER	10
2.3 MT 8870 ไอซีถอดรหัสความถี่	18
2.3.1 คุณสมบัติของ MT8870	19
2.3.2 การนำ MT 8870 ไปใช้งาน	19
2.3.3 โครงสร้างของ MT 8870	20
2.3.4 ฟังก์ชั่นการทำงานภายใน MT 8870	20
2.4 การใช้ทรานซิสเตอร์เป็นสวิทชิ่ง	26
2.4.1 การใช้ทรานซิสเตอร์เป็นสวิทชิ่งในอุดมคติ	26
2.4.2 การใช้ทรานซิสเตอร์เป็นสวิทชิ่งในความเป็นจริง	27
2.4.3 เงื่อนไขที่ทำให้ทรานซิสเตอร์ ON หรือ อิ่มตัว	29
2.4.4 เงื่อนไขที่ทำให้ทรานซิสเตอร์ ON หรือ OFF	29
2.5 แนวทางการใช้งาน LCD MODULE	29
2.5.1 การต่อเข้ากับระบบไมโคร	30
2.6 ทฤษฎีไมโครโปรเซสเซอร์ตระกูล MCS 51	39
2.6.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยว	39
2.6.2 โครงสร้างของ 8051	41
2.6.3 การจัดการหน่วยความจำของ 8051	44
2.6.4 สถาปัตยกรรมของ 8051	46
2.6.5 การทำงานของ 8051	57
2.6.6 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ	60
2.6.7 การรีเซท	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.9 Power Supply 139
3.2 การออกแบบ Soft Ware 139
3.3 ระบบการทำงานและการใช้งานเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ 140
3.3.1 Edit Status 141
3.3.2 Auto Pages 142
3.3.3 Munual Pages 145
3.3.4 Send To PC 147
3.3.5 โหมดการ Clear ข้อมูล 147
3.3.6 การบันทึกข้อความลงในวงจร Play & Record 148

บทที่ 4 การทดลองและผลการทดลอง 152
4.1 ผลการทดลอง 152
4.1.1 การสร้างสัญญาณ Sine Wave 152
4.1.2 การทดลองตั้งสถานะของเครื่อง 152
4.1.3 การทดลองในโหมด Automatic 153
4.1.4 การทดลองในโหมด Manual 153
4.1.5 การทดลองส่งจำนวนครั้งของการใช้งานไปบน PC 153
4.1.6 การทดลอง Clear ข้อมูล 154
4.1.7 การทดลอง Clear สถานะและข้อมูล 154
4.2 ปัญหาที่เกิดขึ้น 154
4.3 การแก้ไข 154

บทที่ 5 บทสรุปและวิจารณ์ 156
5.1 บทสรุป 156
5.2 ประโยชน์ที่ได้รับจากโครงงาน 156
5.3 แนวทางการพัฒนาโครงงาน 157
5.4 งบประมาณค่าใช้จ่าย 157

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก. เครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ 159
ภาคผนวก ข. บล็อคไดอะแกรมและวงจรเครื่องเรียกวิทยุติดตามตัว
ระบบโทนและวอยซ์ แบบอัตโนมัติ 168
ภาคผนวก ค. โฟล์วชาร์ทการทำงานของเครื่องเรียกวิทยุติดตามตัว
ระบบโทนและวอยซ์แบบอัตโนมัติ 173
ภาคผนวก ง. โปรแกรมการทำงานของเครื่องเรียกวิทยุติดตามตัว
ระบบโทนและวอยซ์แบบอัตโนมัติ 183

บรรณานุกรม 220

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปภาพ		หน้า
รูปภาพที่ 2.1	วงจรเบื้องต้นของเดลต้ามอดูเลชั่น	2
รูปภาพที่ 2.2	วงจรที่ใช้แปลงกลับจากข้อมูลดิจิตอลเป็นสัญญาณเสียง	3
รูปภาพที่ 2.3	การเปรียบเทียบสัญญาณอินพุท กับข้อมูลที่ได้สัญญาณอนาลอกจากอินทิเกรเตอร์	3
รูปภาพที่ 2.4	แผนผังการทำงานของระบบ CVSD ในส่วนของการแปลงสัญญาณเสียงเป็นข้อมูลดิจิตอล	5
รูปภาพที่ 2.5	แผนผังการทำงานของระบบ CVSD ในส่วนแปลงกลับจากดิจิตอลเป็นสัญญาณเสียง	5
รูปภาพที่ 2.6	วงจรสมบูรณ์ของเครื่องบันทึกเสียงพูด	6
รูปภาพที่ 2.7	รายละเอียดของขา IC T6668	7
รูปภาพที่ 2.8	การต่ออินเตอร์เฟส T6668 เข้ากับ CPU เพื่อควบคุมการทำงาน	9
รูปภาพที่ 2.9	แสดงสัญลักษณ์โครงสร้างของไฟโต้ ทรานซิสเตอร์	9
รูปภาพที่ 2.10	ตัวเชื่อมโยงทางแสงแบบทั่วๆ ไป	11
รูปภาพที่ 2.11	เป็นชิ้นส่วนอินพุทของตัวเชื่อมโยงทางแสง	12
รูปภาพที่ 2.12	วงจรกระตุ้นอินพุทของเกทแบบ ทีทีแอล	15
รูปภาพที่ 2.13	ขาเบสของตัวเชื่อมโยงทางแสงเบอร์ AN26	16
รูปภาพที่ 2.14	วงจรไม่กลับสัญญาณ	18
รูปภาพที่ 2.15	แสดงรายละเอียดขาของ MT8870	19
รูปภาพที่ 2.16	แสดงโครงสร้างภายใน MT8870	20
รูปภาพที่ 2.17	แสดงความถี่ที่ได้จากภาคกรองความถี่	21
รูปภาพที่ 2.18	แสดงวงจรตรวจสอบสัญญาณ	22
รูปภาพที่ 2.19	แสดงการต่อวงจรภาคอินพุท	23
รูปภาพที่ 2.20	แสดงการต่อวงจรผลิตความถี่	24
รูปภาพที่ 2.21	แสดงวงจรใช้งานเบื้องต้นของ MT8870	24
รูปภาพที่ 2.22	แสดงแผนภูมิเวลา	25
รูปภาพที่ 2.23	วงจรทรานซิสเตอร์ชนิดคอมมอนอิมิตเตอร์	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ หน้า

รูปภาพที่ 2.24 กราฟลักษณะสมบัติของทรานซิสเตอร์ 28
รูปภาพที่ 2.25 การต่อแบบ MEMORY MAP 32
รูปภาพที่ 2.26 การต่อแบบ I/O PORT 32
รูปภาพที่ 2.27 ตำแหน่ง address ของ LCD MODULE รุ่น DMC 202 37
รูปภาพที่ 2.28 ตำแหน่ง address ของ LCD MODULE รุ่น DMC164 37
รูปภาพที่ 2.29 ไมโครคอนโทรเลอร์ แบบชิพเดี่ยวในตระกูล 51 39
รูปภาพที่ 2.30 ไดอะแกรมโครงสร้างของ 8051 42
รูปภาพที่ 2.31 ภาพเสมือนของหน่วยความจำ 42
รูปภาพที่ 2.32 แผนภูมิหน่วยความจำของ 8051 45
รูปภาพที่ 2.33 สถาปัตยกรรมภายในของ 8051 48
รูปภาพที่ 2.34 ไดอะแกรมขาของ 8051 แบบ DIP 49
รูปภาพที่ 2.35 โครงสร้างของพอร์ท 0 49
รูปภาพที่ 2.36 โครงสร้างของพอร์ท 1 51
รูปภาพที่ 2.37 โครงสร้างของพอร์ท 2 51
รูปภาพที่ 2.38 โครงสร้างของพอร์ท 3 52
รูปภาพที่ 2.39 ค่าของรีจิสเตอร์เมื่อเกิดการรีเซท 8051 54
รูปภาพที่ 2.40 วงจรออสซิลเลเตอร์ภายใน 8051 56
รูปภาพที่ 2.41 8051 ทำงานโดยสัญญาณที่มาจากภายนอก 56
รูปภาพที่ 2.42 ลำดับสถานะการทำงานใน MCS-51 59
รูปภาพที่ 2.43 Timing diagram ของการอ่านโปรแกรมจากหน่วย ความจำภายนอก 60
รูปภาพที่ 2.44 วงจรที่มี Program memory อยู่ภายนอก 8051 61
รูปภาพที่ 2.45 Timing diagram ของการอ่านข้อมูลจากหน่วยความจำสำหรับ ข้อมูลภายนอก 8051 62
รูปภาพที่ 2.46 timing diagram ของการเขียนข้อมูลไปยังหน่วยความจำสำหรับ ข้อมูลภายนอก 8051 63
รูปภาพที่ 2.47 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่ภายนอก 8051 64
รูปภาพที่ 2.48 ไดอะแกรมตามเวลาของรีเซท 64
รูปภาพที่ 2.49 Special Function Register (SFR) 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ		หน้า
รูปภาพที่ 2.50	แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิท	68
รูปภาพที่ 2.51	ชุดข้อมูลอนุกรมในโหมด 1	71
รูปภาพที่ 2.52	ชุดข้อมูลอนุกรมในโหมด 2	72
รูปภาพที่ 2.53	Serial Port Control Register (SCON)	73
รูปภาพที่ 2.54	TMOD Timer / Counter Mode Register	77
รูปภาพที่ 2.55	Timer Mode 0:13 bit count	78
รูปภาพที่ 2.56	Timer Mode 2	80
รูปภาพที่ 2.57	Timer 0 Mode 3	81
รูปภาพที่ 2.58	TCON Timer Control Register	82
รูปภาพที่ 2.59	แหล่งกำเนิดสัญญาณขัดจังหวะ	84
รูปภาพที่ 2.60	Interrupt Enable Register	85
รูปภาพที่ 2.61	IP : Interrupt Prioriry Register	87
รูปภาพที่ 2.62	ระบบการขัดจังหวะของ 8052 และ 83154	88
รูปภาพที่ 2.63	PCON : Power Control Register	90
รูปภาพที่ 2.64	Power down and Idel Mode	91
รูปภาพที่ 2.65	Serial Port Mode 0	93
รูปภาพที่ 2.66	Serial Port Mode 1	96
รูปภาพที่ 2.67	Setial Port Mode 2	99
รูปภาพที่ 2.68	Serial Port Mode 3	100
รูปภาพที่ 2.69	วงจรขับสัญญาณ และ รับสัญญาณที่ใช้กับมาตรฐาน RS-232-C	104
รูปภาพที่ 2.70	วงจรขับ และรับสัญญาณที่ใช้กับมาตรฐาน RS-423	105
รูปภาพที่ 2.71	วงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-422	106
รูปภาพที่ 2.72	วงจรที่ใช้การอินเตอร์เฟสแบบ Current - Loop	107
รูปภาพที่ 2.73	ลักษณะสวิทซ์ที่ใช้ส่งระดับสัญญาณTTL	107
รูปภาพที่ 2.74	วงจรแก้การ Bounce ของสวิทซ์	108
รูปภาพที่ 2.75	วงจรขับ LED	109
รูปภาพที่ 2.76	วงจรขับหลอดแสดงผลที่กินกระแสมาก	110
รูปภาพที่ 2.77	การทดลองวงจร D/A แบบรวมกระแส	112
รูปภาพที่ 2.78	ตัวอย่างวงจร D/A แบบรวมกระแส	113

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพที่ 2.79 ตัวอย่างวงจร D/A แบบวงจรบันไดของตัวต้านทาน 114
รูปภาพที่ 2.80 วงจร D/A ที่ใช้กับเลขฐานสิบรหัส BCD 2 หลัก 115
รูปภาพที่ 2.81 บล็อคไดอะแกรม ของ IC8 117
รูปภาพที่ 2.82 Circuit Diagram ของ IC8 118
รูปภาพที่ 2.83 รูปคลื่นแรงดันของ IC8 119
รูปภาพที่ 2.84 เซอร์กิตไดอะแกรมของ IC9 120
รูปภาพที่ 2.85 วงจรขยายสัญญาณขา 7 ของ IC 9 120
รูปภาพที่ 2.86 วงจรควบคุมปริมาตร 121
รูปภาพที่ 2.87 Plan C Code Waveform 122
รูปภาพที่ 3.1 Block Diagram Automatic Pager System 124
รูปภาพที่ 3.2 วงจร Extermal Line Interface 125
รูปภาพที่ 3.3 วงจร Digital Play & Recore 126
รูปภาพที่ 3.4 วงจร DTMF Decorder 128
รูปภาพที่ 3.5 วงจร Matrix Keyboard 4*4 129
รูปภาพที่ 3.6 วงจร LCD MODULE 130
รูปภาพที่ 3.7 วงจร Sinewave Genalate 131
รูปภาพที่ 3.8 ภาพแสดงบอร์ดและตำแหน่งจัมพ์เปอร์ 132
รูปภาพที่ 3.9 ภาพแสดงรายละเอียดของคอนเนคเตอร์ และ ชิพต่างๆ 133
รูปภาพที่ 3.10 ภาพแสดงวงจรทั้งหมดของบอร์ด ANT - 32 134
รูปภาพที่ 3.11 ภาพแสดงลักษณะสัญญาณ RS 232 และ การต่อสาย 135
รูปภาพที่ 3.12 ANT - 32 MEMORY MAP 136
รูปภาพที่ 3.13 แสดงลักษณะการเขียน CONTROL WORD 137
รูปภาพที่ 3.14 วงจร POWER SUPPLY 139
รูปภาพที่ 3.15 Avcase Intergrate Environment 140
รูปภาพที่ 3.16 MAIN MENU 141
รูปภาพที่ 3.17 EDIT STATUS 141
รูปภาพที่ 3.18 สถานะและจำนวนการใช้งานของหมายเลข 142
รูปภาพที่ 3.19 AUTO MATIC MODE 143

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพที่ 3.20 AUTO MATIC MODE ขณะที่ยังไม่ป้อนหมายเลข 143
รูปภาพที่ 3.21 AUTO MATIC MODE ขณะที่ป้อนหมายเลข 144
รูปภาพที่ 3.22 MANUAL MODE ขณะยังไม่ป้อนหมายเลข 145
รูปภาพที่ 3.23 MANUAL MODE ขณะที่ป้อนหมายเลข 146
รูปภาพที่ 3.24 MODE ส่งข้อมูลสู่ PC 147
รูปภาพที่ 3.25 ตัวอย่างข้อมูลที่พิมพ์ออกมาทางเครื่องพิมพ์ 148
รูปภาพที่ 3.26 แสดงตำแหน่ง DIP SWITCH และ SWITCH บนแผ่นวงจรที่ใช้งาน 149
รูปภาพที่ 4.1 รายงานจำนวนครั้งของการใช้เครื่องที่มีสถานะเป็น ON บน PC 153

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง		หน้า
ตารางที่ 2.1	การกำหนดชนิด และ จำนวนแรม ที่ใช้	8
ตารางที่ 2.2	การกำหนดอัตราเร็วข้อมูลหรือบิท	8
ตารางที่ 2.3	แสดงคุณสมบัติของ 4N33	13
ตารางที่ 2.4	แสดงคุณสมบัติของ 4N26	17
ตารางที่ 2.5	แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ	22
ตารางที่ 2.6	ขาสัญญาณของ LCD MODULE	31
ตารางที่ 2.7	ชุดคำสั่งควบคุมและการแสดงข้อความ	33
ตารางที่ 2.8	แสดงค่าความถี่ของ Plan C กลุ่มต่างๆ	122
ตารางที่ 2.9	Plan C Code Table	123
ตารางที่ 3.1	การกำหนดข้อความในช่องต่างๆ ของวงจร Digital Play & record	127
ตารางที่ 3.2	8255 Mode 0 Configutation	138
ตารางที่ 3.3	ตำแหน่งของ DIP - SW บนแผงวงจร	150
ตารางที่ 3.4	ตำแหน่งช่องข้อมูลของ DIP - SW	150
ตารางที่ 4.1	แสดงผลค่าความถี่ของตาราง Plan C ที่ได้จากการทดลอง	152

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ปัจจุบันเป็นยุคของการติดต่อสื่อสาร และการสื่อสารก็มีหลายรูปแบบต่างๆ มากมาย ระบบวิทยุติดตามตัวก็เป็นระบบติดต่อสื่อสารระบบหนึ่งที่สะดวก รวดเร็ว และง่ายต่อการใช้งาน ซึ่งการสื่อสารแห่งประเทศไทยก็เป็นองค์กรหนึ่งที่มีการให้บริการวิทยุติดตามตัวระบบโทนและวอยซ์ แต่เนื่องจากปัจจุบันยังใช้ระบบที่ผู้ใช้บริการที่ต้องการเรียกเครื่องวิทยุติดตามตัวต้องเรียกผ่านพนักงาน (ผ่านศูนย์กลาง) จึงเป็นการไม่สะดวกต่อผู้ใช้บริการ ซึ่งบางครั้งอาจจะมีการส่งข้อความไปยังเครื่องวิทยุติดตามตัวหมายเลขนั้นๆ ล่าช้า

ด้วยเหตุผลดังกล่าว จึงทำให้เกิดแนวความคิดที่จะพัฒนาจากระบบเดิมที่ต้องผ่านพนักงานในการเรียกติดต่อเครื่องวิทยุติดตามตัว มาเป็นให้ผู้ใช้บริการเรียกติดต่อไปยังเครื่องวิทยุติดตามตัวหมายเลขที่ต้องการเองโดยอัตโนมัติ โดยอาศัย Frequency Tone จากการกดคีย์ที่หน้าปัทม์โทรศัพท์มาเป็นตัวกำหนดหมายเลขที่ต้องการเรียกซึ่งจะทำให้ผู้ใช้บริการมีความสะดวกมากยิ่งขึ้นในการใช้บริการ และข่าวสารที่ผู้ใช้บริการส่งไปนั้นก็จะถึงผู้รับในเวลาขณะนั้นทันที ซึ่งเป็นความรวดเร็วในการติดต่อสื่อสารข้อความอีกทางหนึ่ง

ในปริญญานิพนธ์ฉบับนี้ จะกล่าวถึงการพัฒนาเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ ที่ใช้ไมโครโปรเซสเซอร์เป็นส่วนควบคุมการทำงานของระบบ ซึ่งส่งผลให้ระบบมีความถูกต้อง รวดเร็ว และมีประสิทธิภาพที่ดี ตัวเครื่องมีขนาดเล็ก การติดตั้งและการเคลื่อนย้ายทำได้ง่าย มีราคาถูกเมื่อเทียบกับขีดความสามารถ สามารถเรียกเครื่องวิทยุติดตามตัวได้จำนวน 900 เลขหมาย และยังสามารถแก้ไขเพิ่มให้เรียกได้ถึง 1900 เลขหมาย โดยกรรมวิธีทาง Software การ Generate Code จะใช้มาตราฐานตาม Plan C โดยจะกล่าวถึงการ Generate Code ในบทถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

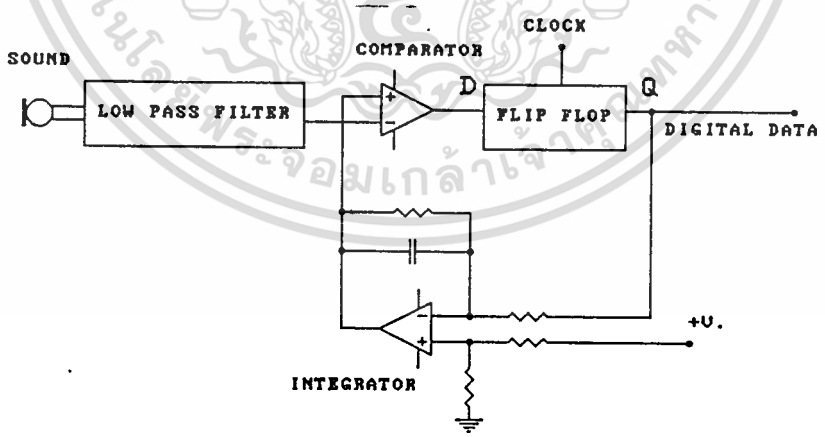
บทที่ 2
ทฤษฎีและหลักการ

2.1 เดลต้ามอดูเลชั่น

เทคนิคของเดลต้ามอดูเลชั่นจะไม่ใช้การสุ่มสัญญาณหนึ่งจุดแล้วแปลงเป็นข้อมูลดิจิตอลหนึ่งเวิร์ด ที่มีความละเอียดเป็นจำนวนบิตที่ต้องการ แต่จะใช้วิธีเปรียบเทียบความสูงหรือการเปลี่ยนแปลงของสัญญาณเสียงแทน

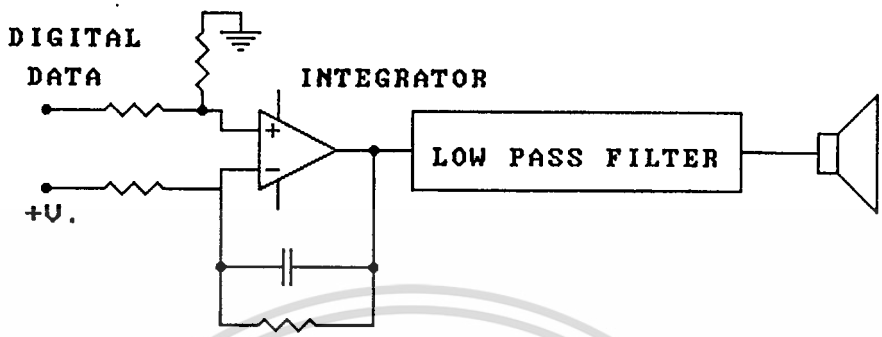
ข้อมูลที่ได้ก็คือทิศทางของการเปลี่ยนแปลง ซึ่งก็มีเพียง ขึ้น หรือ ลง เท่านั้น ดังนั้นความกว้างของข้อมูลดิจิตอลจึงใช้เพียงบิทเดียวก็เพียงพอ ข้อดีของวิธีการเดลต้ามอดูเลชั่นก็คือ ใช้หน่วยความจำน้อยกว่าวิธีการแบบอื่นๆ

รูปที่ 2.1 เป็นวงจรเบื้องต้นของเดลต้ามอดูเลชั่น คอมพาราเตอร์จะทำหน้าที่เปรียบเทียบสัญญาณอินพุทปัจจุบันกับสัญญาณอินพุทก่อนหน้า ซึ่งได้จากการป้อนกลับมายังอินทิเกรเตอร์ เอาท์พุทจากการเปรียบเทียบถูกป้อนผ่านฟลิปฟลอปที่ควบคุมด้วยสัญญาณนาฬิกาเพื่อให้ได้เป็นข้อมูลดิจิตอล ซึ่งก็คือการกำหนดอัตราการสุ่มสัญญาณนั่นเอง

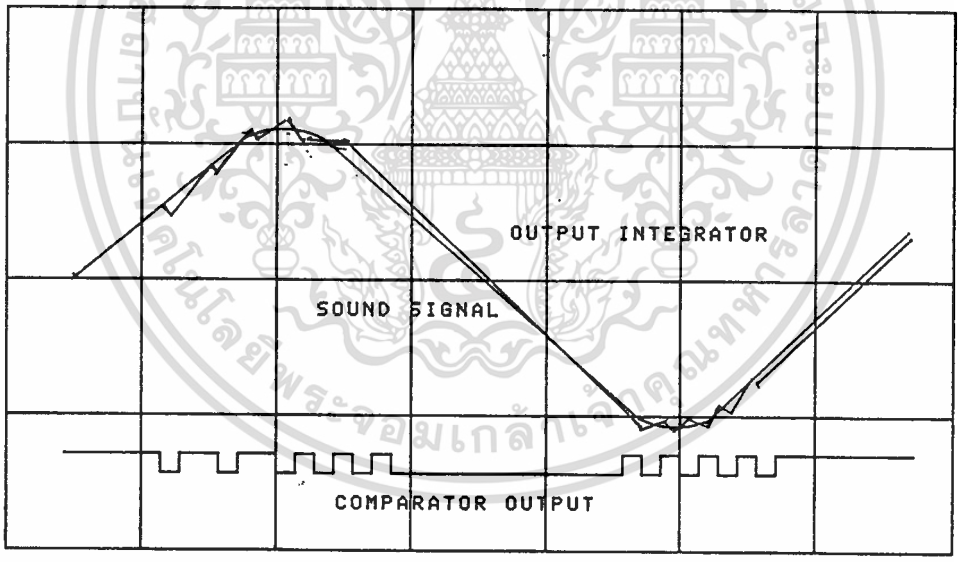


รูปที่ 2.1 วงจรเบื้องต้นของเดลต้ามอดูเลชั่นในส่วนของการแปลงจากสัญญาณเสียงเป็นดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 วงจรที่ใช้แปลงกลับจากข้อมูลดิจิตอลเป็นสัญญาณเสียง



รูปที่ 2.3 เปรียบเทียบสัญญาณอินพุทกับข้อมูลที่ได้สัญญาณอนาลอกจากอินทิเกรเตอร์

สัญญาณที่ได้จากตัวเปรียบเทียบและจากอินทิเกรเตอร์ เปรียบเทียบกับสัญญาณอินพุท แสดงในรูปที่ 2.3 ลักษณะเช่นนี้จะพบว่า ยิ่งความถี่ของสัญญาณนาฬิกามีค่าสูงก็ยิ่งสามารถบันทึกการเปลี่ยนแปลงที่แคบได้มากขึ้น ทำให้ได้คุณภาพเสียงที่ดีขึ้น แต่ก็สิ้นเปลืองหน่วยความจำมากขึ้นตามไปด้วย ความถี่เท่าใดจึงจะเพียงพอคงต้องใช้การทดลอง โดยการนำเอาท์พุทสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท้ายที่เป็นข้อมูลดิจิตอลผ่านวงจรแปลงกลับ ในรูปที่ 2.2 แล้วฟังเสียงที่ได้ หากฟังเป็นภาษามนุษย์รู้เรื่องก็ใช้ที่ค่านั้น สำหรับเสียงพูดคุณภาพเทียบเท่าเสียงจากโทรศัพท์ซึ่งมีแถบกว้างประมาณ 4 kHz ก็ใช้เพียง 16 kHz แต่ที่ความถี่ต่ำถึง 9.6 kHz ก็ยังฟังรู้เรื่อง ความถี่นี้จะเป็นตัวกำหนดอัตราเร็วข้อมูล (bit rate) ซึ่งที่ 16 kHz ก็เท่ากับ 1,600 บิตต่อวินาที

CVSD

ข้อจำกัดของวิธีการเดลต้ามอดูเลชั่นก็คือ แถบกว้างความถี่ใช้งาน ซึ่งถูกจำกัดโดยความถี่สัญญาณนาพิกา และจะสูงกว่าความถี่สูงสุดของสัญญาณอินพุทมากกว่า 2 เท่าขึ้นไป อีกอันหนึ่ง คือ ความเร็วของการเปลี่ยนแปลงความสูงของสัญญาณหรือไดนามิกเรนจ์ ระบบเดลต้ามอดูเลชั่นธรรมดามีค่าไดนามิกเรนจ์ที่แคบ จำเป็นต้องมีส่วนเพิ่มเติมทำหน้าที่ขยายไดนามิกเรนจ์ให้กว้าง โดยการควบคุมอัตราขยายของอินทิเกรเตอร์ เพื่อให้ตอบสนองต่อสัญญาณที่มีความชันมากๆ ได้ทัน ระบบนี้มีชื่อเรียกใหม่ว่า ระบบเดลต้ามอดูเลชั่นแบบเปลี่ยนแปลงความชันต่อเนื่อง หรือ CVSD (Continuous Variable Slope Delta Modulation)

ระบบ CVSD ทั้งส่วนแปลงจากอนาลอกเป็นดิจิตอลและส่วนแปลงกลับจากดิจิตอลเป็นอนาลอก แสดงในรูปที่ 2.4 และ 2.5 ตามลำดับ วิธีการของ CVSD ก็คือมีการตรวจระดับสัญญาณโดยอาจใช้วิธีการจัดให้มีรีจิสเตอร์สำหรับเก็บข้อมูลดิจิตอลล่าสุดจำนวน 3 ถึง 4 บิท แล้วตรวจดูว่าเป็น "0" หมดหรือ "1" หมดหรือไม่ ถ้าใช่แสดงว่าขณะนี้อัตราขยายของอินทิเกรเตอร์ต่ำเกินไป ตอบสนองต่อความชันของสัญญาณไม่ทัน ก็จะทำการเพิ่มอัตราขยายให้สูงขึ้นเฉพาะในช่วงนั้น

ในส่วนของการแปลงกลับก็จะต้องมีการทำงานในลักษณะเดียวกัน คือ มีรีจิสเตอร์ตรวจดูข้อมูลว่าเป็น "0" หมด หรือ "1" หมดหรือไม่ แล้วจัดการควบคุมอัตราขยายของอินทิเกรเตอร์ให้สอดคล้องกัน

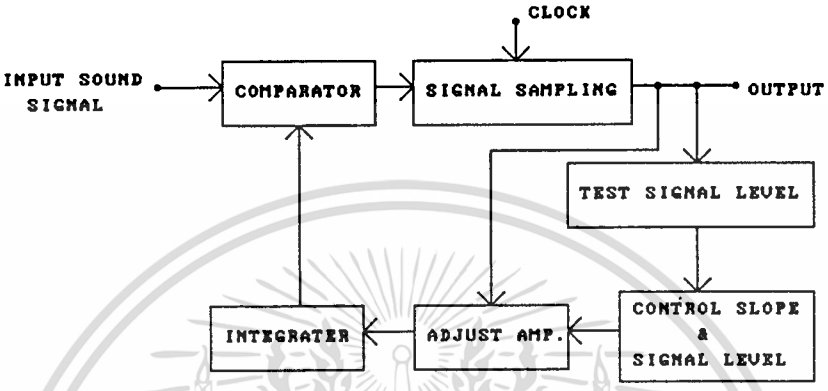
ไอซีบันทึกเสียงสำเร็จรูป

เมื่อรู้จักทฤษฎีการทำงานของการบันทึกเสียงโดยใช้เทคนิค CVSD แล้วก็มารู้จักกับไอซีสำเร็จที่ทำหน้าที่ดังกล่าวกัน ซึ่งออกแบบมาสำหรับงานบันทึกเสียงพูดและเล่นกลับ โดยมีวงจรจัดการเกี่ยวกับหน่วยความจำให้ด้วยเสร็จสรรพ

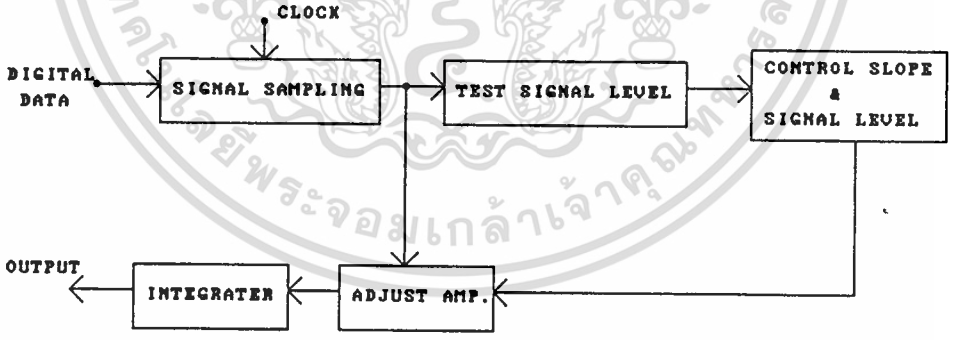
จากที่ได้กล่าวมา บิทเรตหรืออัตราเร็วของข้อมูลสำหรับวิธีการเดลต้ามอดูเลชั่นมีค่าเท่ากับความถี่สัญญาณนาพิกา เสียงพูดคุณภาพเท่าระบบโทรศัพท์มีแถบกว้างความถี่ 4kHz ต้องใช้ความถี่นาพิกา 16 kHz ได้ข้อมูลดินิตอลที่บิตเรต 16 K บิทต่อวินาที หรือพูดอีกอย่างได้ว่า การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบันทึกเสียง 1 วินาทีต้องใช้หน่วยความจำ 16K บิท ถ้าต้องการเวลาบันทึกยาวนานขึ้นก็ต้องใช้หน่วยความจำมากขึ้นเป็นทวีคูณ จึงหลีกไม่พ้นที่จะต้องใช้ไดนามิกแรมซึ่งมีความจุสูง



รูปที่ 2.4 แผนผังการทำงานของระบบ CVSD ในส่วนของการแปลงจากสัญญาณเสียงเป็นข้อมูลดิจิตอล



รูปที่ 2.5 แผนผังการทำงานของระบบ CVSD ในส่วนแปลงกลับจากดิจิตอลเป็นสัญญาณเสียง

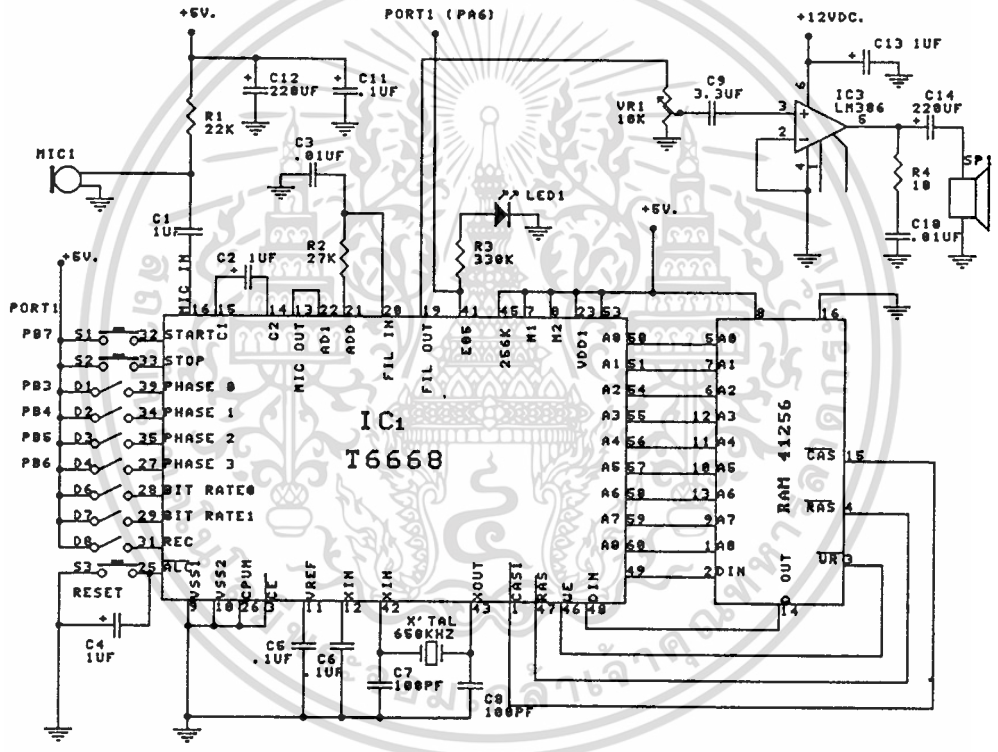
T 6668

T-6668 เป็นของโตชิบา มีฟังก์ชั่นใช้งานค่อนข้างสมบูรณ์ ประยุกต์ใช้งานได้กว้างและสะดวกกว่ามีรูปร่างภายนอกเป็นแบบติดตั้งบนผิวหรือเซอร์เฟสเมาต์ขนาด 60 ขา ต่อเข้ากับหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำชนิดไดนามิก ขนาด 64K*1 บิต หรือ 256K * 1 บิต โดยตรง 4 ตัว ใช้คริสตอลควบคุมความถี่สัญญาณนาฬิกา เปลี่ยนบิทเรตโดยใช้ดิปสวิทช์เลือกหน้าของหน่วยความจำ 256K*1 บิท (41256) จำนวน 4 ตัวที่บิทเรต 16K จะบันทึกได้นาน 64 วินาที หรือนาทีเศษๆ

T6668 นี้สมบูรณ์ในตัวทางด้านอินพุทสามารถต่อไมโครโฟนเข้ากับไอซีได้เลยด้านเอาท์พุทก็เพิ่มภาคขยายอีกส่วนเดียว เนื่องจาก T6668 เป็นแบบเซอร์เฟสเมาต์ มีขาชิดกันมาก และจำเป็นต้องใช้แผ่นวงจรพิมพ์แบบสองหน้าชนิดเพลตทรูโฮลด์



รูปที่ 2.6 วงจรสมบรูณ์ของเครื่องบันทึกเสียงพูดที่ใช้ T6668 บันทึกได้นาน 16 วินาทีต่อแรม 1 ตัว (สูงสุด 4ตัว) แบ่งหน้าบันทึกได้

รายละเอียดขาใช้งานของ T6668

A_0- A_8 : ขาแอดเดรสต่อกับแรม

D_{in},D_{out} : ขาดาต้าต่อกับแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{RES}$,WE : สัญญาณควบคุมแรม

CAS_1 - CAS_4 : ขาเลือกแรมแต่ละตัว รวม 4 ตัว

M_1,M_2 : ใช้กำหนดจำนวนแรมที่ใช้ ดูตารางที่ 2.1

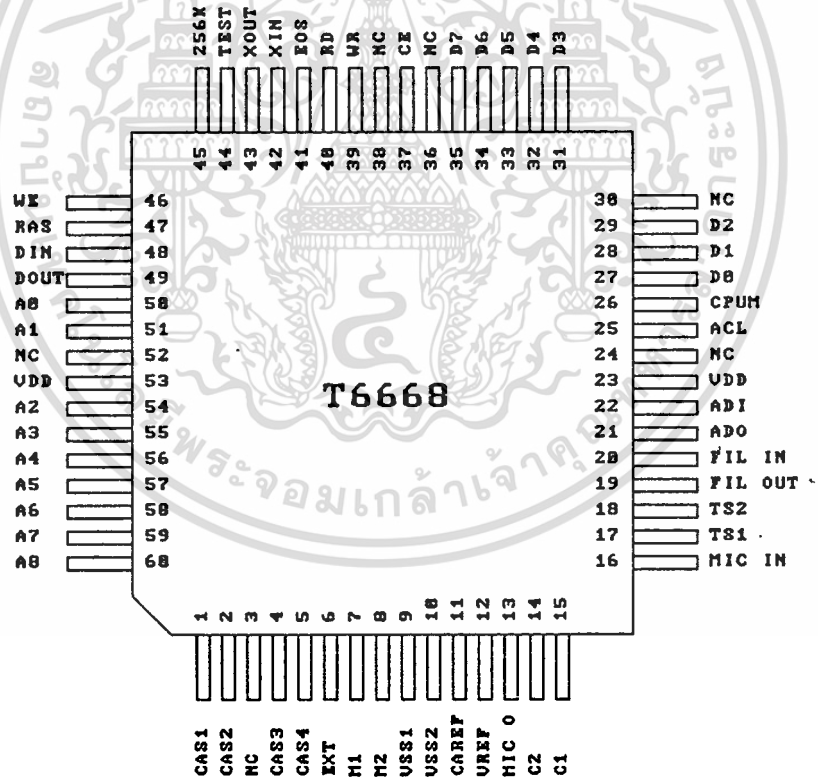
256K : เลือกขนาดแรม ต่อกราวด์ใช้ 4164 ต่อไฟบวกใช้ 41256

EOS : เอาท์พุท เป็นไฮเมื่อจบข้อความที่บันทึก

MIC_{in},MIC_{out} : อินพุทและเอาท์พุทของภาคขยายส่วนหน้า

AD_I,AD_O : อินพุทสัญญาณอนาลอกที่จะนำไปแปลงเพื่อบันทึก และเอาท์พุท อนาลอกที่ได้จากการอ่าน

FIL_{in},FIL_{out} : วงจรกรองความถี่ต่ำผ่าน



รูปที่ 2.7 รายละเอียดของขาไอซี T 6668

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	256 K(ขา 45)	M_2	M_1
4164*1	0	0	0
4164*2	0	0	1
4164*3	0	1	0
4164*4	0	1	1
41256*1	1	0	0
41256*3	1	0	1
41256*3	1	1	0
41256*4	1	1	1

ตารางที่ 2.1 การกำหนดชนิดและจำนวนแรมที่ใช้

บิตเรต	D7 (ขา35)	D6 (ขา34)
8 K	0	0
11 K	0	1
16 K	1	0
32 K	1	1

ตารางที่ 2.2 การกำหนดอัตราเร็วข้อมูลหรือบิท

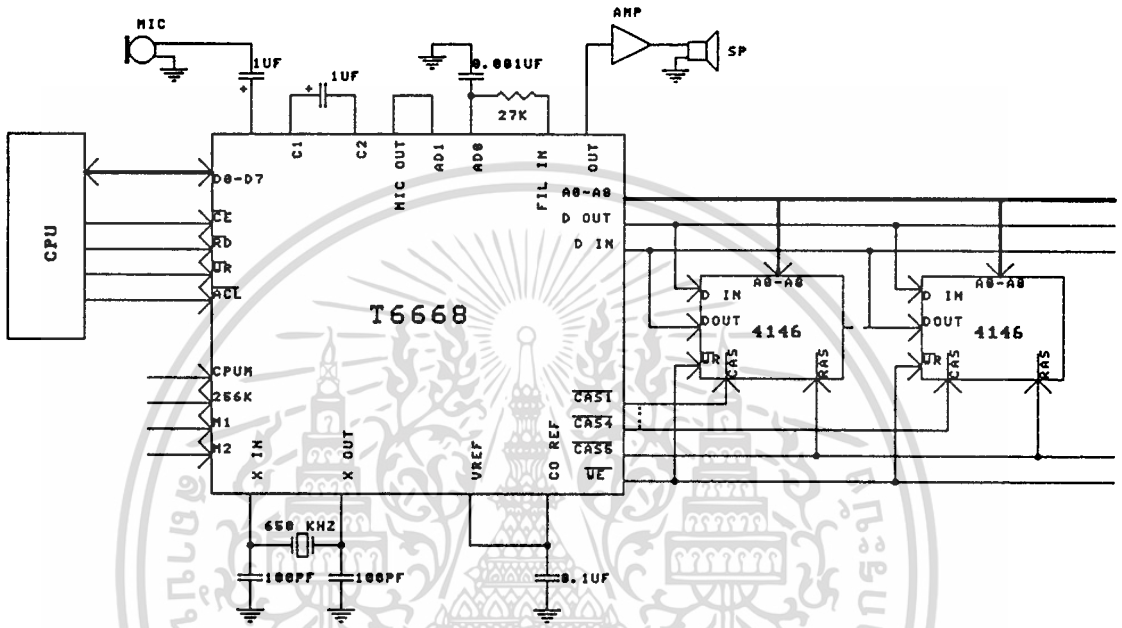
2.2 อุปกรณ์ที่ใช้

2.2.1. โฟโต้ทรานซิสเตอร์

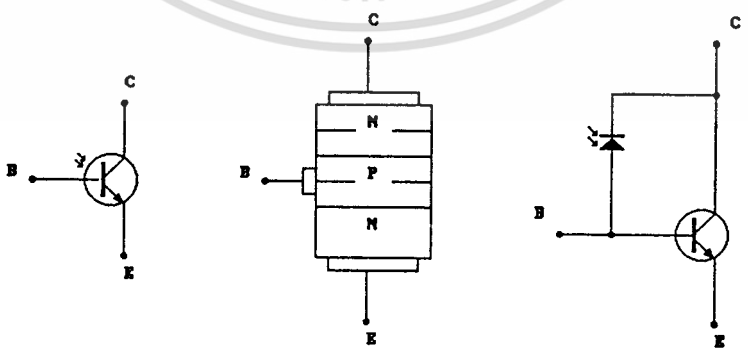
โฟโต้ทรานซิสเตอร์เป็นอุปกรณ์รับแสง ซึ่งอุปกรณ์รับแสงหมายถึง อุปกรณ์ที่สามารถเปลี่ยนพลังงานแสงให้แปลค่ากับค่าของพลังงานทางไฟฟ้าได้ โดยอุปกรณ์จะประกอบด้วยสารกึ่งตัวนำ ซึ่งอาจจะนำมาต่อเชื่อมให้เกิดรอยต่อหรือเป็นเนื้อสารกึ่งตัวนำอย่างเดียวก็ได้ในภาวะปกติของการใช้งานของโฟโต้ทรานซิสเตอร์ รอยต่อระหว่างเบสอิมิตเตอร์จะต่อไบอัสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนรอยต่อระหว่างเบสคอลเลคเตอร์จะต่อไบอัสกลับที่รอยต่อนี้เอง เป็นส่วนทำให้เกิดการแปลค่ากระแสที่ขึ้นกับแสง



รูปที่ 2.8 การต่ออินเตอร์เฟส T6668 เข้ากับ CPU เพื่อควบคุมการทำงาน



รูปที่ 2.9 แสดงสัญลักษณ์โครงสร้างของโฟโต้ทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อไบอัสกลับที่รอยต่อ ระหว่างเบสคอลเลคเตอร์และมีแสงฉายมาที่บริเวณรอยต่อ กระแสเบสจะถูกขยายด้วยอัตราการขยายทรานซิสเตอร์เป็นกระแสอิมิตเตอร์ และถ้าไบอัสขั้วเบสด้วยกระแสเบสภายนอกก็จะถูกขยายรวมกับกระแสเนื่องจากแสงด้วย

ถ้าให้ I_P = กระแสที่เกิดขึ้นเนื่องจากแสง

I_B = กระแสเบสที่มาจากภายนอก

I_E = กระแสอิมิตเตอร์

I_{FE} = อัตราขยายของทรานซิสเตอร์

จากสมการของทรานซิสเตอร์ คือ $I_C = h_{FE}I_B$ และ $I_E = I_C + (I_B + I_F)$ เราจะได้

$$I_E = (I_F + I_B)(h_{FE} + 1)$$

จะเห็นว่ากระแส I_E เปลี่ยนแปลงตามกระแส I_F ด้วยอัตราการขยายถึง $h_{FE} + 1$ เท่า ซึ่งถ้าเป็น I_F มีค่าเปลี่ยนแปลงจาก 1 ถึง 10uA และให้ h_{FE} มีค่าประมาณ 100 เราจะได้ค่า I_E เปลี่ยนแปลงจาก 100uA ถึง 1mA

ถ้าอัตราขยายยิ่งสูง (A_1 มาก) จะทำให้ผลตอบสนองต่อแสงจะไวขึ้น ค่า h_{FE} สูงๆ จะต้องทำให้รอยต่อเบส - คอลเลคเตอร์มีพื้นที่มาก แต่ก็ทำให้กระแสรั่วไหลสูงขึ้นด้วย เพราะรอยต่อจะถูกไบอัสกลับ

2.2.2. OPTOCOUPLER

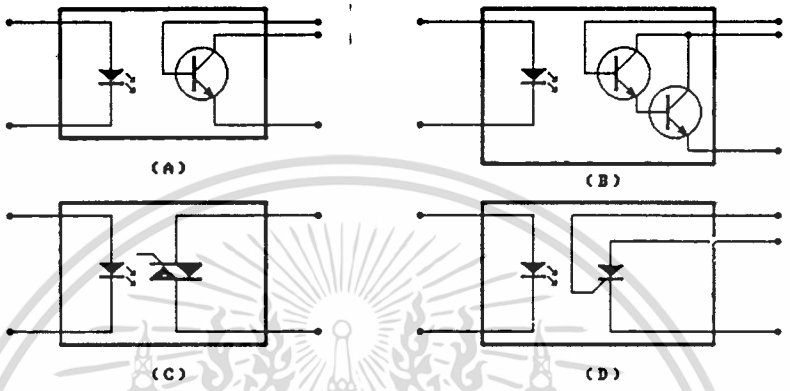
2.2.2.1 พื้นฐานของตัวเชื่อมโยงทางแสง

ตัวเชื่อมโยงทางแสง บางครั้งเรียกว่า ตัวแยกโดยใช้แสง (OPTOISOLATOR) เป็นอุปกรณ์เดียวที่ประกอบด้วยแหล่งกำเนิดแสงและตัวตรวจรับแสง โดยที่ทั้งสองชิ้นส่วนนี้แยกจากกันโดยมีฉนวนที่โปร่งใสคั่นกลาง และชิ้นส่วนทั้งหมดจะถูกบรรจุอยู่ในตัวถังทึบแสง

แหล่งกำเนิดแสง สำหรับตัวเชื่อมโยงทางแสง ส่วนมากแล้วจะใช้ไดโอดเปล่งแสงอินฟาเรด (INFARED EMITTING DIODE) ที่ทำจากสารแกลเลี่ยม อาร์เซไนด์ ส่วนตัวตรวจรับหรืออุปกรณ์ภาคเอาท์พุทนั้น อาจจะเป็นโฟโต้ทรานซิสเตอร์, โฟโต้ดาร์ลิงตัน, สวิทช์สองทิศทาง ซึ่งทำงานเมื่อมีแสงมากระตุ้น และ SCR ที่ถูกกระตุ้นด้วยแสง ในรูปที่ 2.10 ได้แสดงให้เห็นถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์ของวงจรชนิดต่างๆ ที่กล่าวมาแล้ว ถึงแม้ว่าจะมีหลายชนิดมากกว่านี้ แต่รูปที่แสดงก็นำมาเฉพาะที่พบเห็นกันบ่อยๆ เท่านั้น



รูปที่ 2.10 ตัวเชื่อมโยงทางแสงแบบทั่วๆ ไป

สัญญาณจะถูกส่งระหว่างชิ้นส่วนทั้งสองส่วนที่แยกจากกันทางไฟฟ้า โดยอยู่ในรูปของสัญญาณแสง โดยที่ชิ้นส่วนทั้งสองนี้ไม่สามารถสลับหน้าที่กันได้ และไม่มีการเชื่อมโยงทางไฟฟ้าระหว่างชิ้นส่วนทั้งสอง สัญญาณที่ถูกส่งผ่านจึงได้ทิศทางเดียวเท่านั้น

2.2.2.2 คุณสมบัติของตัวเชื่อมโยงทางแสง

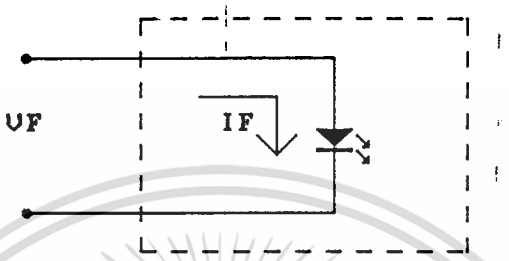
ในการออกแบบ ตัวเชื่อมโยงทางแสงควรเข้าใจถึงตัวแปรต่างๆ ที่สำคัญ เนื่องจากสนใจเฉพาะวงจรความถี่ต่ำ ตัวแปรทางด้านไฟฟ้ากระแสตรงสามารถแบ่งออกเป็น อินพุท เอาท์พุท และอัตราส่วนของการส่งผ่านกระแส (Current Transfer Ratio)

อัตราส่วนของการส่งผ่านกระแส หรือ CTR นั้น เป็นอัตราส่วนระหว่างกระแสอินพุทต่อกระแสเอาท์พุทของตัวเชื่อมโยงทางแสง (เรียกว่า Bias แทนด้วยตัวอีต้า η) ซึ่งค่านี้จะขึ้นอยู่กับประสิทธิภาพของไดโอดเปล่งแสงอินฟาเรดและช่องว่างระหว่างชิ้นส่วนทางอินพุทและเอาท์พุท โดยที่พื้นที่ ความไว (Sensitivity) และอัตราการขยายตัวของตัวตรวจรับ ก็มีบทบาทที่สำคัญเช่นกัน

ตัวแปรอินพุททางด้านไฟฟ้ากระแสตรง ซึ่งเป็นตัวกำหนดตัวแปรทางด้านไฟฟ้าของไดโอดเปล่งแสงอินฟาเรด ได้แก่ กระแสของไดโอดเมื่อได้รับไบอัสตรง (I_F) แรงดันตกคร่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดโอดเมื่อได้รับไบอัสตรง (V_F) และแรงดันสูงสุดที่ทนได้ เมื่อได้รับไบอัสกลับ (V_R) ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 เป็นชิ้นส่วนอินพุทของตัวเชื่อมโยงทางแสง

เนื่องจากตัวแปรเอาท์พุททางด้านไฟฟ้ากระแสตรงและตัวแปรส่งถ่าย (TRANSFER PARAMETER) จะแตกต่างกันโดยขึ้นอยู่กับชนิดของชิ้นส่วนที่เป็นตัวตรวจรับที่ใช้ในตัวเชื่อมโยงทางแสง ตัวเชื่อมโยงทางแสงที่ใช้โฟโต้ทรานซิสเตอร์ และโฟโต้คาร์ลิงตันนั้น มีหลักการทำงานเหมือนกัน รอยต่อระหว่างขาคอลเลคเตอร์กับขาเบสถูกทำให้กว้างขึ้น แสงที่ตกกระทบรอยต่อจะทำให้เกิดคู่ของอิเลคตรอนและโฮลขึ้นมา เกิดการนำกระแสไฟ ซึ่งตัวแปรของทั้งสองแบบมีดังนี้

I_C : เป็นกระแสสูงสุดที่ไหลต่อเนื่องผ่านขาคอลเลคเตอร์ (output)

$V_{(BR)CBO}$: เป็นแรงดันพังทลายสูงสุดจากขาคอลเลคเตอร์ไปยังขาเบส

$V_{(BR)CEO}$: เป็นแรงดันพังทลายสูงสุดจากขาคอลเลคเตอร์ไปยังขาอิมิตเตอร์

$V_{(BR)ECO}$: เป็นแรงดันพังทลายสูงสุดจากขาอิมิตเตอร์ปยังขาคอลเลคเตอร์

ตัวเชื่อมโยงทางแสง ที่ใช้สวิทช์สองทิศทางที่ทำงานเมื่อมีแสงมากระตุ้นเป็นภาคเอาท์-พุทนั้น ถูกออกแบบมาสำหรับใช้ในงานซึ่งต้องการการแยก การทริกหรือกระตุ้นตัวไตรแอค การแยกการสวิทช์ทางด้านไฟฟ้ากระแสสลับที่มีขนาดกระแสต่ำ และการแยกทางไฟฟ้าที่มีค่าสูง มีตัวแปรที่สำคัญดังนี้

$I_{T(RMS)}$: เป็นค่ากระแส rms สูงสุดขณะอยู่ในสถานะที่ทำงาน (on-state)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุท	I_r			50	mA
	V_r (IF =10 mA)		1.2	1.5	V
	V_r		3	3	V
เอาท์พุท	I_T (RMS)			100	mA
	V_{DMR}		2.5	250	V
	V_{TM} (I_T =100 mA)			3.0	V
ตัวแปร	I_F		8.0	15	mA
เชื่อมโยง	I_B		100		A

ตารางที่ 2.3 แสดงคุณสมบัติของ 4N33

2.2.2.3 วงจรเปลี่ยนระดับแรงดัน

เมื่อต้องการวงจรทางด้านดิจิตอลรับสัญญาณอินพุทจากอุปกรณ์ภายนอก ซึ่งมีค่าแรงดันสูง จึงจำเป็นที่จะต้องเปลี่ยนระดับแรงดันให้อยู่ในระดับลอจิก 5 โวลท์ ถ้าสัญญาณอินพุทเป็นไฟตรงก็สามารถเชื่อมต่อกับวงจรลอจิกโดยใช้ตัวเชื่อมโยงทางแสง โดยไม่ต้องเกี่ยวข้องกันทางไฟฟ้าระหว่างวงจรทั้งสองส่วน

ประโยชน์ของการแยกวงจรทั้งสองออกจากกันนี้ ก็คือ สัญญาณรบกวนใดๆ หรือแรงดันที่เป็นยอดแหลม (Spike Voltage) ที่เกิดจากสายกราวด์ของวงจรภายนอกจะไม่เข้าไปเกิดขึ้นในสายกราวด์ของวงจรลอจิกเลย นอกจากนี้ตัวเชื่อมโยงทางแสงยังสามารถใช้เพื่อเปลี่ยนไฟสลับเป็นสัญญาณลอจิก 5 โวลท์ได้ ในขณะเดียวกันก็จะแยกวงจรลอจิกออกจากไฟสลับที่มีค่าสูงด้วย

V_{DRM} : เป็นค่าแรงดันซ้ำๆ ระหว่างขั้วเอาท์พุท เมื่ออยู่ในสถานะหยุดทำงาน (repetitive off-state output terminal voltage)

V_{TM} : เป็นแรงดันยอดสูงสุด (peak voltage) เมื่ออยู่ในสถานะที่ทำงาน

ตัวเชื่อมโยงทางแสงที่ใช้ SCR ที่ถูกกระตุ้นด้วยแสงนั้น ออกแบบมาสำหรับใช้ในงานที่ต้องการแยกกันทางไฟฟ้าที่มีค่าสูงระหว่างวงจรด้านแรงดันต่ำ (ใช้ไอซี) และทางด้านไฟฟ้ากระแสสลับแรงดันสูงตัวแปรที่สำคัญสำหรับอุปกรณ์ที่ใช้ SCR มีดังนี้

$L_{T(RMS)}$: เป็นกระแส rms สูงสุดเมื่ออยู่ในสถานะทำงาน

V_{DRM} : เป็นค่าแรงดันซ้ำๆระหว่างขั้วเอาท์พุท เมื่ออยู่ในสถานะหยุดทำงาน

V_{RM} : เป็นค่าแรงดันย้อนกลับสูงสุด

ตัวแปรของการส่งผ่านของตัวเชื่อมโยงทางแสงเป็นการวัดอัตราส่วนของการส่งกระแสระหว่างชิ้นส่วนอินพุทและเอาท์พุท สำหรับตัวเชื่อมโยงทางแสงมีค่าตัวแปรที่สำคัญดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C_{TR} : เป็นอัตราส่วนต่ำสุดระหว่างกระแสเอาท์พุทของคอลเลคเตอร์สูงสุดต่อกระแสไดโอดที่ค่า V_{CE} และ I_F ที่กำหนด

$V_{CE(SAT)}$: เป็นแรงดันอิ่มตัวระหว่างขาคอลเลคเตอร์กับขาอิมิตเตอร์

สำหรับตัวเชื่อมโยงทางแสงที่ใช้สวิทซ์สองทิศทาง ซึ่งทำงานเมื่อมีแสงมากระตุ้นและแบบที่ใช้ SCR นั้นมีตัวแปรดังนี้

I_{FT} : เป็นค่ากระแสกระตุ้นไดโอดเปล่งแสงอินฟาเรดสูงสุด ซึ่งต้องการใช้เพื่อคงสถานะเอาไว้ได้

I_H : เป็นค่ากระแสยึด (holding current) ซึ่งต้องการสำหรับเอาท์พุทเพื่อที่จะคงสถานะให้เอาท์พุทค้าง (latch) ไว้

คุณสมบัติของตัวเชื่อมโยงทางแสง 3 ชนิด คือ เบอร์ 4N33, 4N26 และ MOC3010 ได้แสดงในตารางที่ 2.3,2.4,2.5 ตามลำดับ

ในรูปที่ 2.12 แสดงถึงการใช้งานของตัวเชื่อมโยงทางแสงซึ่งอินพุทเป็นไฟตรง 12 โวลท์ จะถูกเปลี่ยนให้อยู่ในระดับลอจิก 5 โวลท์ จะเห็นว่าในวงจรใช้เบอร์ 4N33 (ดูตารางที่ 2.3) เมื่ออินพุทมีแรงดัน 12 โวลท์ จะทำให้เอาท์พุทเป็นระดับลอจิก "1" ในขณะเดียวกันสัญญาณรบกวนที่มีลักษณะเหมือนกัน (common-mode noise) จะถูกกำจัดไปโดยไดโอดที่อินพุทของตัวเชื่อมโยงทางแสง

เมื่อสัญญาณ 12 โวลท์ถูกป้อนเข้าสู่อินพุทของตัวเชื่อมโยงทางแสงจะเกิดกระแสไหลผ่าน R_1 และไดโอดเปล่งแสงอินฟาเรด กระแสที่เกิดขึ้นจะไปทำให้ไดโอดสว่าง และแสงที่เกิดขึ้นจะไปตกกระทบรอยต่อระหว่างคอลเลคเตอร์และเบสของโฟโต้ดาร์ลิงตัน ทำให้มันทำงานขึ้นมา สาเหตุที่ใช้เอาท์พุทเป็นโฟโต้ดาร์ลิงตัน ก็เนื่องจากมันมีค่าอัตราการส่งผ่านกระแสสูง ทำให้มีกระแสเพียงพอไหลผ่านตัวต้านทาน R_2 เพื่อให้เกิดเอาท์พุทที่ระดับลอจิก "1" โดยที่สัญญาณเอาท์พุทสามารถใช้ขับอินพุทของลอจิกเกทต่างๆ ได้

เมื่อนำเอาแรงดัน 12 โวลท์ ออกจะทำให้โฟโต้ดาร์ลิงตันหยุดทำงาน และ R_2 จะดึงให้เอาท์พุทมีค่าเป็น "0"

เมื่อออกแบบวงจรที่คล้ายๆกันนี้สำหรับอินพุทที่เป็นระดับสัญญาณไฟตรงหลายระดับค่าแรงดัน กำหนดด้วยตัวแปรอินพุทของวงจรลอจิกเกทที่จะป้อนเข้าไป ค่าของ R_2 หาได้โดย

$$R_2 \quad < \quad V_{IL} / I_{IL}$$

โดยที่ V_{IL} เป็นค่าแรงดันของเกทเมื่อระดับแรงดันอินพุทอยู่ในลอจิก "0" และ I_{IL} เป็นกระแสเกท เมื่ออินพุทอยู่ในลอจิก "0" เช่นกัน

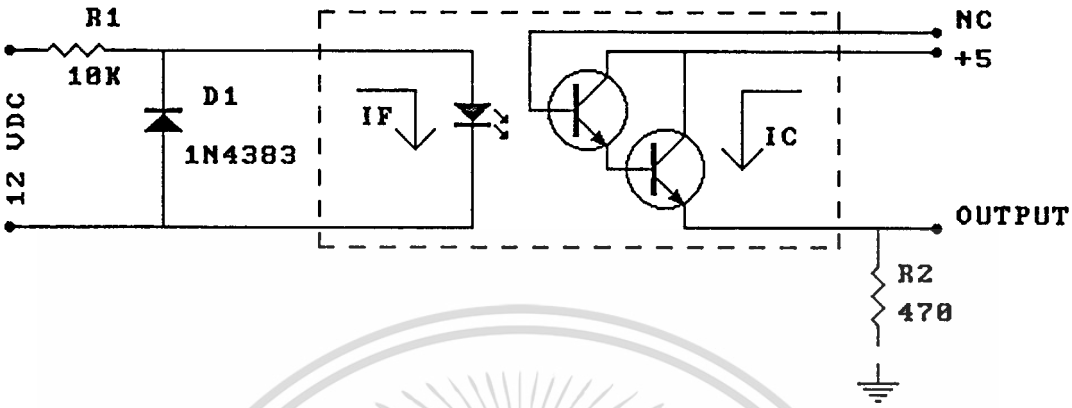
ค่าของ R_1 หาได้จากการหากระแสคอลเลคเตอร์ (I_C) ก่อน

โดย $\quad I_C \quad = \quad V_{IH} / R_2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก379 ค 2537

ห้องสมุด
คณะครุศาสตร์อุตสาหกรรม สจล.



รูปที่ 2.12 วงจรกระตุ้นอินพุทของเกทแบบทีทีแอล

โดยที่ V_{IH} เป็นระดับแรงดันอินพุทลอจิก "1" เพื่อป้อนเข้าสู่เกท และ R_2 เป็นค่าความต้านทาน มีหน่วยเป็นโอห์ม ขั้นต่อไปก็หากระแสของไดโอดเมื่อได้รับไบอัสตรง (I_F)

$$I_F = I_C / \eta$$

เมื่อ IF เป็นอัตราส่วนผ่านกระแสของตัวเชื่อมโยงทางแสง สามารถหาได้จากตารางที่ 2.3 จะได้ว่า

$$= I_C / I_F$$

$$= 50 / 10 = 5$$

ดังนั้นจะหาค่า IF ได้และจะได้ค่าของ R1 โดยได้จาก

$$R_1 = (V_{IH} - V_F) / I_F$$

ตัวอย่าง เช่น จงคำนวณหาค่า R_1 และ R_2 ในวงจรรูปที่ 2.12 โดยสมมติว่า ต่อลอจิกเกทตระกูล 7400 สำหรับเกทตระกูลนี้มีค่าตัวแปรอินพุทคือ V_{IL} = 0.8 โวลท์, V_{IH} = 2 โวลต์ และ IL = -1.6 มิลลิแอมป์ ดังนั้นจะหาค่า R_2 ได้จาก

$$R_2 < V_L / I_L = 0.8 / (1. * 10^{-3}) < 500$$

จะต้องเลือกค่าของ R_2 เป็นค่าของตัวต้านทานที่มีค่า มาตรฐานใกล้เคียง โดยที่มีค่าน้อยกว่า 500 โอห์ม ซึ่งหาได้เท่ากับ 470 โอห์ม เมื่อหาค่า R_2 ได้แล้ว เราจะใช้ค่านี้ในการหาค่ากระแสคอลเลคเตอร์ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1284 021052

$$I_C = V_{IH} / R_2 = 2 / 470 = 4.3 \text{ mA}$$

เมื่อระดับแรงดันอินพุทลอจิก "1" เท่ากับ 2 โวลท์ และค่า R2 เท่ากับ 470 โอห์ม ค่าของ I_F หาได้จาก

$$I_F = I_C / \eta = 4.3 / 5 = 1 \text{ mA}$$

ดูในตารางที่ 2.3 ที่ค่า I_F ในช่องอินพุท และ Ic ในช่องเอาท์พุท ค่าทั้งสองตัวนี้จะต้องไม่เกินอัตราสูงสุดของเบอร์ 4N33

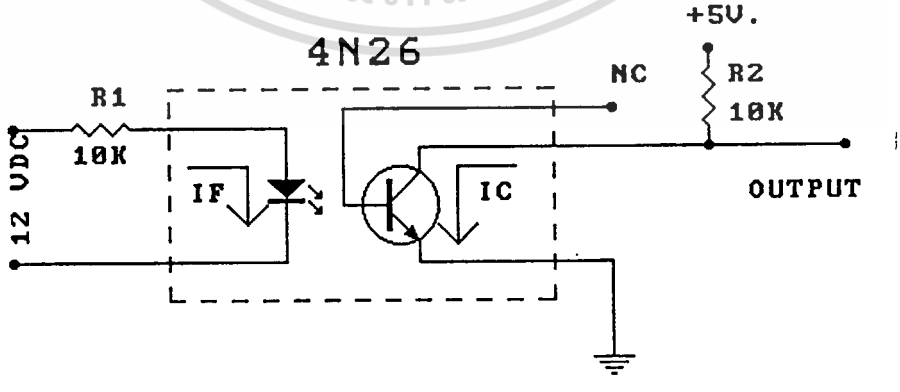
ค่าของ R_1 หาได้จาก

$$R_1 = (V_{IN} - V_F) / I_F = (12 - 2) / 1*10^{-3} = 10.8K$$

ค่าของตัวต้านทานทั่วไปที่ใกล้เคียงที่คำนวณได้ คือ 10K

โดยการลดค่าของ R_1 จะเป็นการเพิ่มผลของการถ่วง (loading effect) สัญญาณจากแหล่งกำเนิด และลดค่าประสิทธิภาพของการส่งผ่าน เช่น ในวงจรคล้ายกันเมื่อเลือกค่า R_1 ที่ให้ค่ากระแส I_F เท่ากับ 20 mA อัตราส่วนของการส่งผ่านกระแสจะเหลือเพียง 46 % เท่านั้น

วงจรในรูปที่ 2.13 จะเปลี่ยนอินพุท 24 โวลท์ ไปเป็นเอาท์พุท 5 โวลท์ ที่มีขั้วกลับกัน (invert) ซึ่งหมายความว่า เมื่ออินพุทมีค่าสูงจะทำให้เอาท์พุทมีค่าต่ำ เมื่อป้อนสัญญาณ 24 โวลท์เข้าไปจะเกิดกระแสไหลผ่านไดโอดเปล่งแสงอินฟาเรด แสงที่เปล่งออกมาจะไปกระทบกับโฟโต้ทรานซิสเตอร์ ทำให้เกิดการนำกระแสขึ้น เนื่องจากเอาท์พุทนั้นต่อจากขาคอลเลคเตอร์จึงได้ค่าแรงดันลอจิก "0" เมื่อต่อไปเข้าอินพุทของลอจิกเกท



รูปที่ 2.13 ขาเบสของตัวเชื่อมโยงทางแสงเบอร์ 4N26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำสัญญาณอินพุทออกไปจะทำให้โฟโต้ทรานซิสเตอร์หยุดนำกระแส และ R_2 จะดึงให้เอาท์พุทเป็นลอจิก "1" การใช้เบอร์ 4N26 แทนเบอร์ 4N33 ก็เนื่องจากมีค่าแรงดันอิ่มตัวระหว่างขาคอลเลคเตอร์และอิมิตเตอร์ VCE(SAT) ต่ำ ในตารางที่ 2.4 จะแสดงให้เห็นคุณสมบัติต่างๆ ของเบอร์ 4N26

		ค่าต่ำสุด	ค่าปกติ	ค่าสูงสุด	หน่วย
อินพุท	I_r			50	mA
	V_r (IF =10 mA)		1.2	1.5	V
	V_r			3	V
เอาท์พุท	I_T (RMS)			100	mA
	V_{DMR}			250	V
	V_{TM} (I_T =100 mA)		2.5	3.0	V
ตัวแปร	I_F		8.0	15	mA
เชื่อมโยง	I_B		100		A

ตารางที่ 2.4 แสดงคุณสมบัติของ 4N26

ค่าความต้านทาน R_2 ไม่ได้จำเพาะเจาะจงนัก ส่วนค่าของ R_1 ที่กำหนดมาสำหรับอินพุทที่มีระดับแรงดันใดๆ หาได้จาก $R_1 = (V_{in} - V_F) / I_F$

เมื่อ

$$I_F = ((V_{CC} - V_{CE(sat)} / R_2) - I_{IL}) / \eta$$

ค่าของ I_F จะเป็นตัวรับประกันว่า โฟโต้ทรานซิสเตอร์จะเกิดการอิ่มตัวขึ้น ค่าของ R_1 เมื่ออินพุทมีแรงดัน 24 โวลต์ สามารถหาได้ง่าย ลองสมมติว่าเกทที่จะนำไปขับเป็นไอซีทีทีแอล 7400 เนื่องจาก R_2 มีค่า 10k และเกทต้องการกระแสอินพุท I_{IL} เท่ากับ -1.5mA ดังนั้น

$$I_F = ((5 - 0.4) / 10{,}000) + 0.0016) / 0.2 = 10 \text{ mA}$$

เมื่อได้ค่า I_F แล้วสามารถคำนวณหาค่า R_1 ได้

$$R_1 = (24 - 1.1) / 10*10\text{-}3 = 2.3k$$

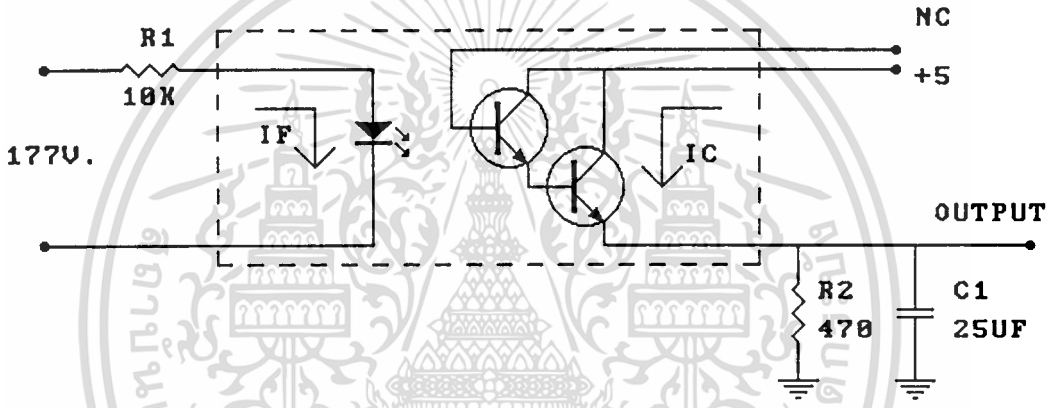
ค่ามาตรฐานที่มีใกล้เคียง คือ 2.2k

วงจรไม่กลับสัญญาณ (non - inverting circuit) ซึ่งจะเปลี่ยนสัญญาณไฟสลับขนาด 117 โวลท์ให้เป็นระดับลอจิก 5 โวลท์ แสดงให้เห็นในรูปที่ 2.14 เมื่อป้อนอินพุทขนาด 117 โวลท์เข้าไป จะเกิดกระแสไหลผ่านไดโอดเปล่งแสงอินฟาเรดในช่วงครึ่งไซเคิลบวก และผ่านไดโอด D1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในช่วงครึ่งไซเกิลลบ นอกจากนี้ D1 ยังทำหน้าที่จำกัดแรงดันไบอัสกลับ (VR) ไม่ให้เกิน 0.7 โวลต์ อีกด้วย

ในระหว่างแต่ละครึ่งไซเกิลบวกนั้น โฟโตดาร์ลิงตันจะนำกระแส ซึ่งจะทำให้เกิดแรงดันไฟตรง แบบกระเพื่อมตกคร่อม R_2 ซึ่งจะถูกกรองให้เรียบโดยตัวเก็บประจุ C_1 แรงดันที่ตกคร่อม C_1 จะทำให้อินพุทของเกทมีค่าสูง และเมื่ออินพุทไฟสลับถูกนำออกไป โฟโต้ดาร์ลิงตันจะหยุดนำกระแส แรงดันที่ตกคร่อม C_1 จะตกลงไปเนื่องจากตัวเก็บประจุจะคายประจุผ่าน R_2 ตอนนี้ R_2 จะดึงให้อินพุทของเกทมีค่าต่ำ



รูปที่ 2.14 วงจรไม่กลับสัญญาณซึ่งเปลี่ยนวงจรไฟสลับ 117 โวลท์มาเป็นระดับลอจิก 5 โวลท์

2.3 MT8870 ไอซีถอดรหัสความถี่

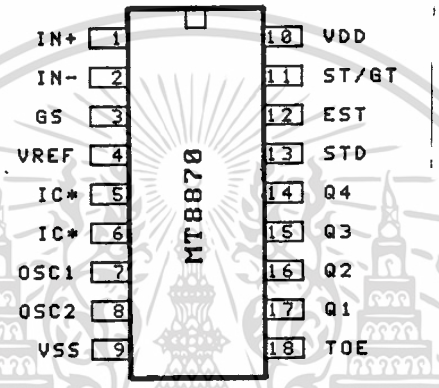
โทรศัพท์ (Integrated DTMF Receiver)

ไอซีทางโทรศัพท์ไม่ค่อยมีผู้นำเข้ามาจำหน่ายหายากคู่มือไม่มี ขาดการเผยแพร่ ทั้งๆที่ในปัจจุบันโทรศัพท์มีส่วนสำคัญกับชีวิตประจำวันของเราอย่างมากมาย และนับวันก็จะยิ่งทวีความสำคัญเพิ่มมากขึ้นทุกที และอุตสาหกรรมทางด้านโทรศัพท์ และการสื่อสารก็ขยายตัวขึ้นอย่างรวดเร็ว ดังนั้นช่างหรือวิศวกรอย่างพวกเราก็น่าจะมาศึกษาเอาไว้ เพื่อจะได้นำมาใช้งานได้ในโอกาสต่อไป ก่อนอื่นขอให้ความหมายของคำว่า ถอดรหัสความถี่โทรศัพท์ อันหมายถึงการแปลงสัญญาณความถี่ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่ม (ชนิด Tone หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DTMF) ให้เป็นระบบตัวเลขทางดิจิตอล ซึ่งไอซี MT8870 ใช้แปลงความถี่โทรศัพท์ให้เป็นเลขฐานสองขนาด 4 บิท

ในยุคก่อนการออกแบบวงจรถอดรหัสความถี่ของโทรศัพท์ มักใช้ไอซีจำพวกเฟสล็อกลูป ซึ่งสร้างปัญหามาก ไม่ว่าเรื่องของความถี่ที่เปลี่ยนแปลงไป การปรับแต่งวงจร ขนาดของวงจรที่ใหญ่ เพราะต้องใช้ไอซีจำนวนมาก



รูปที่ 2.15 แสดงรายละเอียดขาของ MT8870

2.3.1 คุณสมบัติของ MT8870

- เป็นตัวรับและถอดรหัสความถี่ (DTMF receiver)
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดไทม์ (Guard time) ได้
- เป็นไอซีคุณภาพสูง

2.3.2 การนำ MT8870 ไปใช้งาน

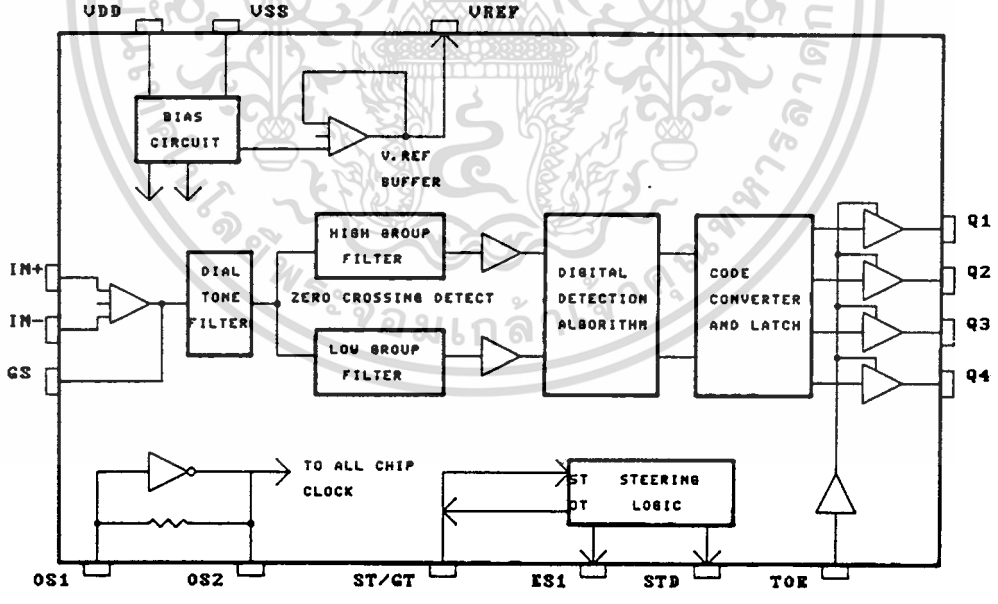
- นำไปใช้งานด้านรีโมตคอนโทรล
- เครื่องป้องกันโทรศัพท์ทางไกล
- ใช้ในงานเกี่ยวกับเครดิตการ์ด
- ใช้งานร่วมกับคอมพิวเตอร์
- ใช้โนเครื่องชุมสายขนาดย่อยหรือ PABX
- ใช้กับงานทางด้านโทรศัพท์ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องกันขโมย
- การควบคุมอุปกรณ์ทางโทรศัพท์
- ใช้ทำเครื่องสอบถามทางโทรศัพท์

2.3.3 โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังก์ชันทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO-CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิทช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอลเพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ ออกเป็นเลขฐานสองขนาด 4 บิท และเช็คช่วงเวลาที่สัญญาณเข้ามา ส่วนภาคอินพุทเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอก เอาต์พุทเป็นวงจรแลตช์ 3 สถานะ รูปที่ 2.15 แสดงขาของ MT8870 และรูปที่ 2.16 แสดงโครงสร้างภายในของ MT8870



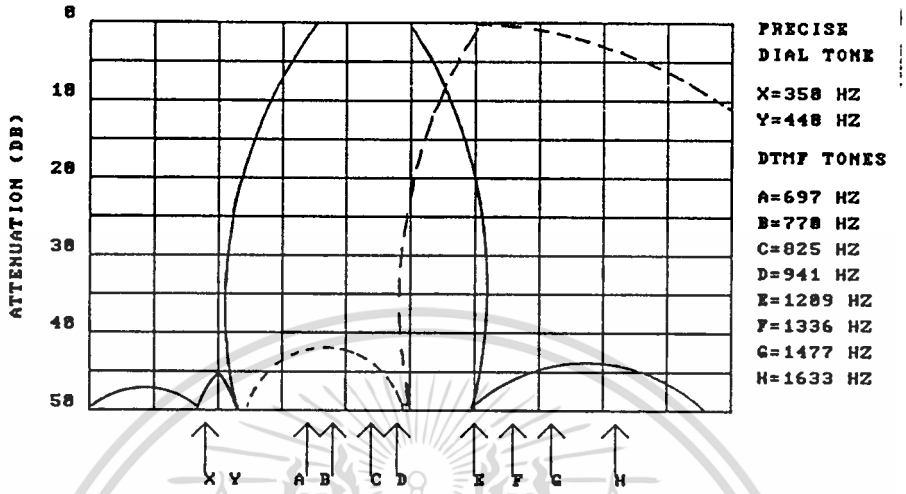
รูปที่ 2.16 แสดงโครงสร้างภายใน MT8870

2.3.4 ฟังก์ชันการทำงานภายใน MT8870

ภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

2.3.4.1 ภาคกรองความถี่ (filter section)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงความถี่ที่ได้จากภาคกรองความถี่

2.3.4.2 ภาคถอดรหัส (decoder section)

2.3.4.3 ภาคตรวจสอบสัญญาณ (steering circuit)

2.3.4.4 ภาคขยายสัญญาณความแตกต่าง (differential input)

2.3.4.5 ภาคกำเนิดความถี่ (oscillator)

2.3.4.1 ภาครองสัญญาณความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิด สวิทช์คาปาซิเตอร์ (six-order switched capacitor band pass filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วง คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ

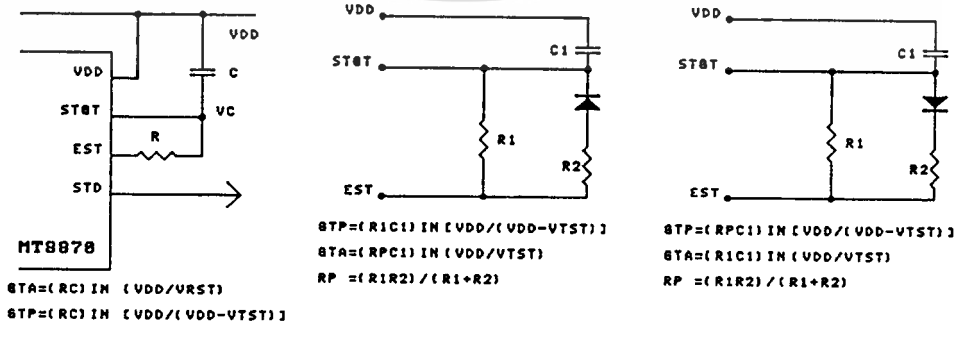
2.3.4.2 ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิตอล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสมเมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา ESt (early steering) ก็จะแอคทีฟ สำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆ นั้น แสดงในตารางที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

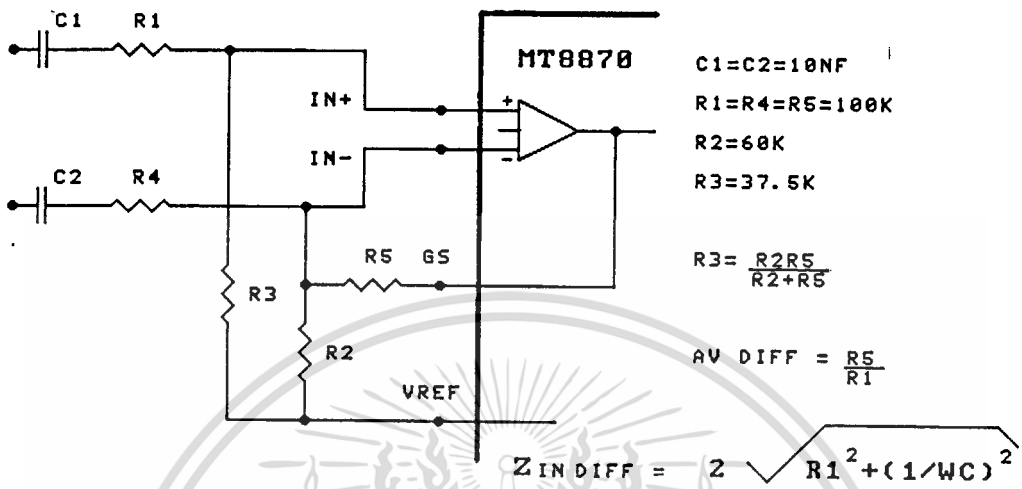
F_{LOW}	F_{HIGH}	NO	TOE	Q_4	Q_3	Q_2	Q_1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

ตารางที่ 2.5 แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ



รูปที่ 2.18 แสดงวงจรตรวจสอบสัญญาณอย่างง่ายและแสดงการกำหนดเวลาการ์ดไทม์ (gard time) พร้อมวิธีคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 แสดงการต่อวงจรภาคอินพุท

2.3.4.3 การตรวจสอบสัญญาณ

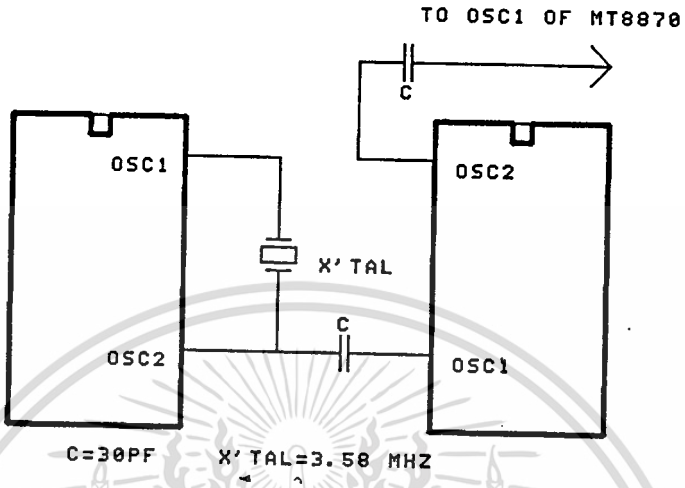
ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาท์พุทจะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา ESt จะเป็น "High" นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 2.17 เมื่อขา ESt เป็น "High" ทำให้ V_C สูงขึ้น ตัวเก็บประจุ C จะคายประจุทำให้แรงดัน V_C สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัส จึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิท รายละเอียดการทำงานขอให้ดูจากแผนภูมิเวลาหรือไทมิ่งไดอะแกรม (timing diagram) ในรูปที่ 2.22 จะเข้าใจได้ง่ายกว่า

สำหรับคำว่าการ์ดไทม์ (gard time) นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่เราตั้งไว้โดย RC ก็คือการ์ดไทม์นั้นเอง เมื่อสัญญาณความถี่เข้ามานานเท่าหรือมากกว่าเวลาที่ตั้งไว้จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสเป็นรหัสตัวเลขออกไป การตั้งเวลาและการคำนวณเวลาดูได้จากรูปที่ 2.18

2.3.4.4 ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุทของ MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยายโดยต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

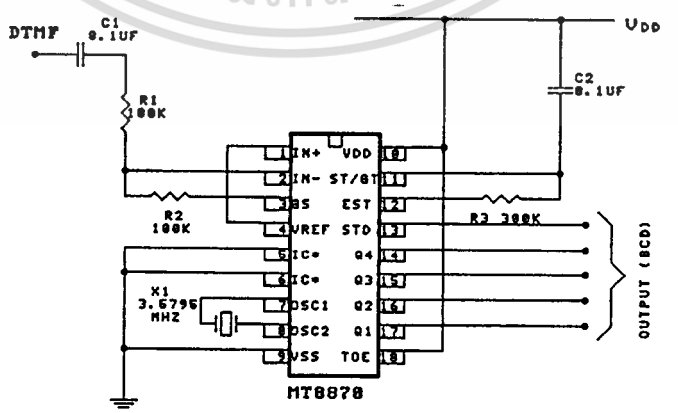


รูปที่ 2.20 แสดงการต่อวงจรผลิตความถี่

วงจรภายนอกเพิ่มเข้าไป รูปที่ 2.19 แสดงการต่อวงจรภายนอกเข้ากับอินพุท ซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอินพุทและอิมพีแดนซ์ได้ ดังนี้

อัตราขยาย $(A_V diff) = R_5 / R_1$

อินพุทอิมพีแดนซ์ $(Z_{in} diff) = \sqrt{R1^2 + \left(\frac{1}{WC}\right)^2}$

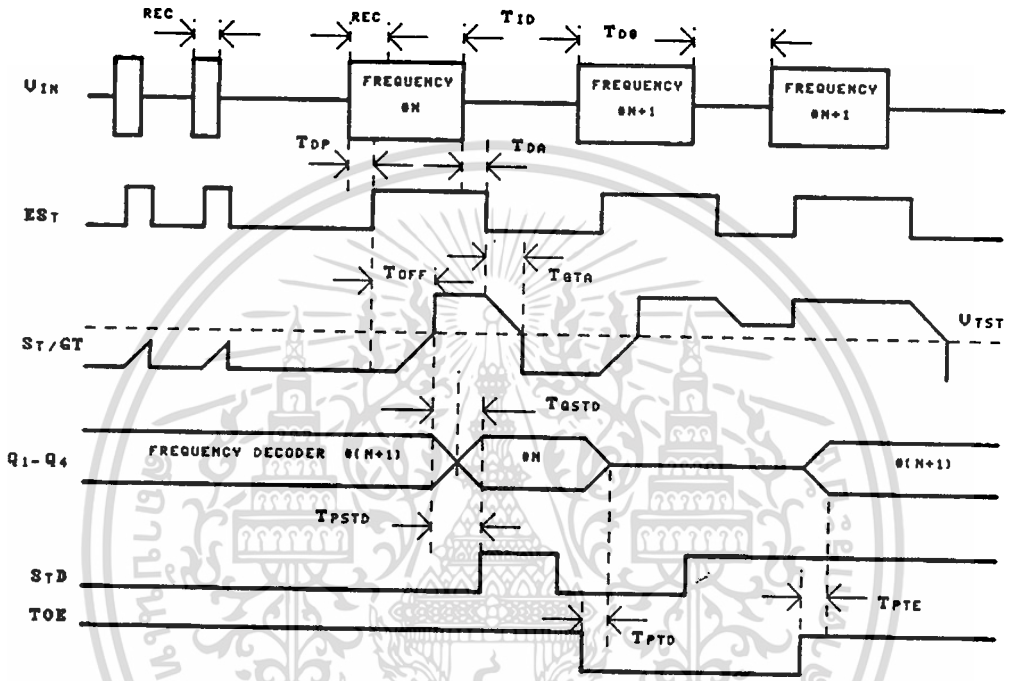


รูปที่ 2.21 แสดงวงจรใช้งานเบื้องต้นของ MT8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.5 ภาคกำเนิดความถี่

ในภาคนี้ภายในไอซีจะมีวงจรเวลาอยู่ภายใน เพียงแต่ต่อแร่คริสตอลขนาด 3.58 MHz ก็สามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่แสดงในรูปที่ 2.20



รูปที่ 2.22 แสดงแผนภูมิเวลา (timing diagram) ของ MT8870

อธิบายขั้นตอนการทำงาน

A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาท์พุทไม่เปลี่ยน

B - ความถี่ #n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแลตช์ไว้ที่เอาท์พุท

C - จบความถี่ #n ช่วงห่างถูกต้อง เอาท์พุทยังคงแลตช์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D - เอาท์พุทเปลี่ยนเป็นไฮอิมพีแดนซ์

E - ความถี่ #n+1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถอดรหัสและแลตช์ไว้

F - ความถี่ #n+1 หายไป ช่วงห่างไม่ถูกต้อง เอาท์พุทยังคงแลตช์อยู่

G - จบความถี่ #n+1 ช่วงห่างถูกต้อง เอาท์พุทยังแลตช์อยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vin - สัญญาณความถี่ DTMF ที่เข้ามา

ESt - Early Sterring output ใช้แสดงความถี่ที่ถูกต้อง

St/GT - Steering input/Guard Time output สำหรับต่อกับ RC ภายนอก

Q_1-Q_4 - เอาท์พุท BCD ขนาด 4 บิต

TOE - Tone Output Enable (input) ใช้ควบคุม Q_1-Q_4 ให้เป็นไฮอิมพีแดนซ์

t_{REC} - คาบเวลานานสุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง

t_{ID} - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

t_{DO} - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{GTP} - การ์ดไทม์ของการปรากฏความถี่ DTMF

2.4 การใช้ทรานซิสเตอร์เป็นสวิทช์

ทรานซิสเตอร์มีลักษณะสมบัติการทำงานได้สามสถานะ คือ แอคทีฟ คัทออฟ และอิ่มตัว ในที่นี้จะกล่าวถึงการทำงานของทรานซิสเตอร์อยู่ในสภาวะคัทออฟและอิ่มตัวนั่นคือทรานซิสเตอร์มีคุณสมบัติเป็นสวิทช์

2.4.1 การใช้ทรานซิสเตอร์เป็นสวิทช์ในอุดมคติ

ในรูปที่ 2.23 เป็นวงจรทรานซิสเตอร์ชนิดคอมมอนอิมิตเตอร์ที่ทำหน้าที่เป็นสวิทช์ โดยตัวทรานซิสเตอร์ เสมือนเป็นสวิทช์ให้กับโหลด R1 ที่ระหว่าง Vcc กับขา C ของทรานซิสเตอร์ส่วนขา E ต่อลงกราวด์ ทรานซิสเตอร์จะทำตัวคล้ายเป็นสวิทช์ ซึ่งจะปิด-เปิดระหว่างขา E กับขา C ส่วนแรงดันอินพุทที่ป้อนเข้าที่ขา B ทำตัวเป็นส่วนควบคุมการทำงานของส่วนสวิทช์ ถ้าพิจารณาตามกฎของเคียชโฮฟจะพบว่า ส่วนของแรงดัน Vce สามารถหาได้จาก

$$Vce = Vcc - Ic\ RL$$

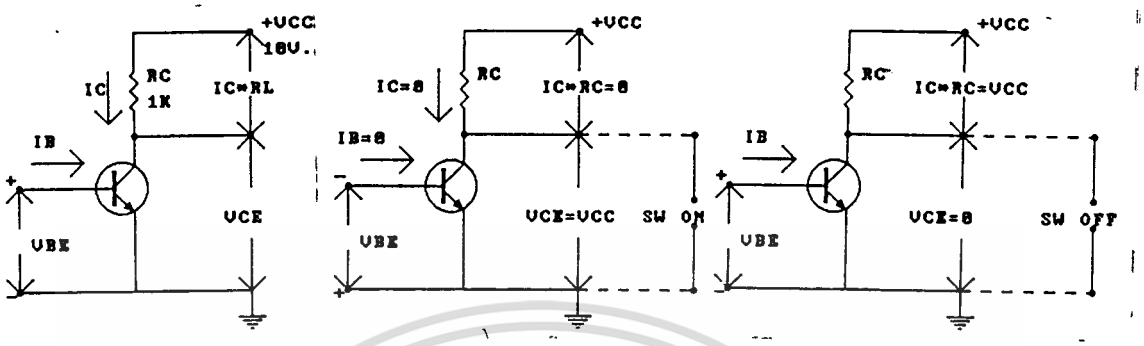
เมื่อแรงดันที่ B - E มีค่าเป็นศูนย์ ส่วนของกระแส IB จะมีค่าเป็นศูนย์ด้วย กระแส Ic ก็มีค่าเป็นศูนย์ด้วย ทรานซิสเตอร์อยู่ในสภาวะคัทออฟ นั่นคือแรงดันตกคร่อมโหลดจะมีค่าเป็นศูนย์

แรงดันตกคร่อม VCE หาได้จาก

$$V_{CE} = V_{CC} - (0 * R_L)$$

$$= V_{CC}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 วงจรทรานซิสเตอร์ชนิดคอมมอนอิมิตเตอร์ที่ทำหน้าที่เป็นสวิทช์

เมื่อให้แรงดันที่เบสเป็นบวก หรือป้อนกระแส IB จนทำให้ทรานซิสเตอร์อิ่มตัวไหลได้เป็นจำนวนมากมาย ดังนั้นแรงดันตกคร่อมโหลดจะมีค่าเป็น I_CR_L ซึ่งมีค่าประมาณได้เท่ากับ VCC และจากสมการสามารถหา VCE ได้เป็น

$$V_{CE} = V_{CC} - V_{CC}$$
$$= 0$$

หรืออาจกล่าวได้ว่าเมื่อทรานซิสเตอร์อิ่มตัวมีผลทำให้ VCE มีค่าประมาณเท่ากับศูนย์ จากคุณสมบัติทางอุดมคติ ทำให้กำลังงานสูญเสียที่เกิดขึ้นกับทรานซิสเตอร์พลอยมีค่าเป็นศูนย์ด้วย ทั้งนี้เพราะ

$$P_D = I_C * V_{CE}$$

เมื่อทรานซิสเตอร์อยู่ในสภาวะคัทออฟ IF = 0

$$P_D = 0 * V_{CE} = 0$$

เมื่อทรานซิสเตอร์อยู่ในสภาวะอิ่มตัว VCE = 0

$$P_D = I_C * 0 = 0$$

2.4.2 การใช้ทรานซิสเตอร์เป็นสวิทช์ในความเป็นจริง

พิจารณากราฟลักษณะสมบัติของทรานซิสเตอร์ จากสมการเดิม เมื่อ $I_C = 0$ และ $V_{CE} = V_{CC}$ ซึ่งค่า V_{CE} จะมีค่าเท่ากับ V_{CC} = 10 โวลท์ (ดูรูปที่ 2.23) ร่างจุด A ลงบนกราฟที่ $I_C = 0$ และ V_{CE} = 10 โวลท์ แต่เมื่อ V_{CE} = 10 โวลท์

เราสามารถหาค่า IC ได้ดังนี้

$$0 = V_{CC} - I_CR_L$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_C = V_{CC} / R_L$$
$$= 10\ V / 1\ K$$
$$= 10\ mA$$

ร่างจุด B ลงบนกราฟที่ $V_{CE} = 0$ และ $I_C = 10$ mA ลากเส้นสมการโหลดเส้นเชื่อมต่อ AB (ดูรูปที่ 2.24)

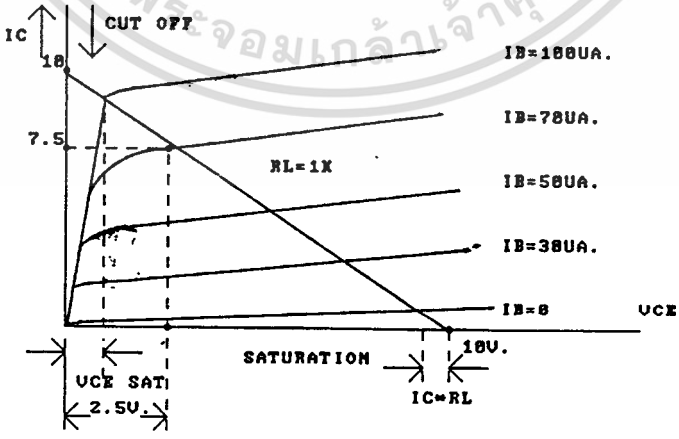
เส้นสมการโหลดนี้จะเป็นบอกค่าความสัมพันธ์ของกระแสและแรงดัน นั่นคือที่จุดค่ากระแส I_C มีค่าต่างๆ แรงดัน V_{CC} นั่นคือจะมีกระแสบางส่วนไหลผ่านทรานซิสเตอร์ได้ บ้างกระแสส่วนนี้เรียกว่า กระแสรั่วไหล (ICEO) ดังนั้นค่าแรงดัน V_{CE} จริงๆ สามารถคำนวณได้จาก

$$V_{CE} = V_{CC} - I_{CEO}R_L$$

เมื่อ $V_{CC} = 10$ โวลท์, $R_L = 1$ K และ $I_{CEO} = 1$ uA

$$V_{CE} = 10\ V - (1\ uA * 1\ K)$$
$$= 9.999 \text{ โวลท์}$$
$$\approx V_{CC}$$

เมื่อทรานซิสเตอร์เข้าสู่สภาวะอิ่มตัว แรงดัน V_{CE} ที่เกิดขึ้นยังไม่มีค่าเป็นศูนย์ จะมีค่าประมาณเท่ากับ V_{CEsat} (ประมาณ 0.2 โวลต์)



รูปที่ 2.24 กราฟลักษณะสมบัติของทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 เงื่อนไขของการทำให้ทรานซิสเตอร์ออนหรืออิ่มตัว

2.4.3.1 ถ้าเราทำการออกแบบ หรือคำนวณค่า I_B และ I_C ได้จากวงจร ค่าที่ปรากฏเมื่อ $I_B > I_C / hFE$ จะทำให้ทรานซิสเตอร์อิ่มตัว

2.4.3.2 ทรานซิสเตอร์จะอิ่มตัวเมื่อให้ไบอัสตรงระหว่างขา B และ E

2.4.4 เงื่อนไขการทำให้ทรานซิสเตอร์ออนหรือออฟ

2.4.4.1 รอยต่อระหว่าง B และ E อยู่ในลักษณะไบอัสกลับ หรือแรงดัน VBE มีค่าเป็นศูนย์

2.4.4.2 กระแสเบสมีค่าเท่ากับศูนย์

ในการออกแบบทรานซิสเตอร์เป็นสวิทช์ต้องคำนึงถึง การกำหนดค่า hFEmin หรือค่าที่ต่ำที่สุดที่เป็นไปได้ ทรานซิสเตอร์จะอิ่มตัว เมื่อกำหนดอัตราของกระแส $I_C / I_B = h_{FEmin}$

2.5 แนวทางการใช้งาน LCD MODULE

ปัจจุบัน LCD เป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องมือเครื่องใช้ต่างๆ ทั้งนี้เนื่องจากมีความเหมาะสมด้วยประการทั้งปวง ทั้งในด้านของการกินกระแสต่ำ สามารถแสดงผลเป็นตัวอักษรและตัวเลข หรือแสดงเป็นกราฟฟิกได้ (เฉพาะรุ่น) จะติดปัญหาก็คือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อน และหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD MODULE ออกมา คือ เป็น MODULE ที่มีตัว LCD และวงจรควบคุมมาให้พร้อม (เรียกว่า LCM) ซึ่งทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครได้ง่าย และสะดวกสำหรับการเขียนโปรแกรม รวมทั้งมีจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม ทำให้ผู้ใช้ทางด้านไมโครหันมาใช้แผงแสดงด้วย LCD MODULE กันมากขึ้น

LCD MODULE มีอยู่มากมายหลายรุ่น และมีคุณสมบัติเดกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือ แบบ DOT MATRIX และ GRAPHIC โดยแบบ DOT MATRIX จะแสดงผลเป็นตัวอักษรขนาด 5 * 8 DOT และมีจำนวนอักษรและบรรทัดแตกต่างกันไปในแต่ละรุ่น ส่วนแบบ GRAPGIC จะสามารถแสดงผลในแบบ BIG-MAP คือจะสร้างเป็นภาพใดๆ ก็ได้ตามต้องการ แนวทางในการใช้งานของทั้ง 2 แบบ จะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบ DOT MATRIX มากกว่า เนื่องจากมีราคาถูก และเพียงพอต่องานส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหญ่ และคู่มือเล่มนี้จะกล่าวถึงการใช้งานกับ DOT MATRIX เท่านั้น คุณสมบัติของ DOT MATRIX LCD MODULE สามารถสรุปเป็นข้อๆ ได้ดังนี้

1. มีให้เลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป

2. ตัวอักษรแสดงด้วย DOT MATRIX ขนาด 5*8 DOT

3. สามารถต่อเข้ากับระบบไมโครได้ 2 ลักษณะ คือ แบบ MEMORY MAP (20-PIN LCD BUS) และแบบผ่าน 8255 PORT (26-PIN 8255 BUS) โดยกรณี 26-PIN 8255 BUS จะใช้แผ่น PCB (DMCAD) เป็นตัว ADAPTER ทำให้เป็น 8255 BUS อีกที

4. การใช้งานง่ายและสะดวก ระบบไมโครเพียงแค่ส่งข้อมูลให้กับ LCD MODULE เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และจะคงค้างไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโคร

5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR DISPLAY, HOME CURSOR, ON OFF CURSOR, BLINK CHARCTER และอื่นๆ อีก

6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว

7. กินกระแสน้อย และมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5V เท่านั้น

2.5.1 การต่อเข้ากับระบบไมโคร

LCD MODULE จะต่อเข้ากับระบบไมโครได้ 2 ลักษณะ คือ แบบ MEMORY MAP โดยผ่าน LCD BUS ขนาด 20 PIN และแบบ I/O PORT โดยผ่าน 8255 BUS ขนาด 26 PIN ซึ่งทั้งสองแบบนี้จะมีข้อดีและข้อด้อยแตกต่างกันไป โดยแต่ละแบบจะมีหลักการดังนี้

การต่อแบบ MEMORY MAP

- สามารถต่อเข้ากับ CHIP เบอร์ทั่วๆ ไปได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครมองเห็น LCD MODULE ในลักษณะของ MEMORY ได้ทันที

- ผู้ใช้สามารถเขียนและอ่านข้อมูลจาก LCD MODULE ได้ ทำให้มองเสมือนว่าเป็น MEMORY BUFFER ไปในตัว

- เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบ FLAG ความพร้อมในขณะที่ LCD MODULE กำลังทำงานได้

- ใช้ได้กับบอร์ดที่มี LCD BUS มาให้พร้อมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำให้กันพื้นที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการ DECODE ละเอียดพอควร

- การจัดขาสัญญาณจะต้องเป็นไปตามแบบของ CHIP แต่ละเบอร์ด้วย

PIN	SYMBOL	LEVEL	FUNCTION
1	Vss	---	0 V GND
2	Vcc	---	+ 5 V POWER SUPPLY
3	Vee	---	+ V FOR LIQUID CRYSTAL DRIVE
4	RS	H/L	REGISTER SELECT H: DATA INPUT L: INSTRUCTION
5	R/W	H/L	INPUT
6	E	H	H: DATA READ L: DATA WRITE
7	DB 0	H/L	ENABLE SIGNAL (L -> H)
8	DB 1	H/L	DATA BUS BIT 0
9	DB 2	H/L	DATA BUS BIT 1
10	DB 3	H/L	DATA BUS BIT 2
11	DB 4	H/L	DATA BUS BIT 3
12	DB 5	H/L	DATA BUS BIT 4
13	DB 6	H/L	DATA BUS BIT 5
14	DB7	H/L	DATA BUS BIT 6 DATA BUS BIT 7

ตารางที่ 2.6 ขาสัญญานของ LCD MODULE

การต่อแบบ I/O PORT

1. สามารถต่อเข้ากับ I/O PORT ใดๆ ก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมาให้ตรงกับข้อกำหนดของ LCD MODULE

2. ผู้ใช้จะเขียนข้อมูลใช้ LCD MODULE ได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนด MEMORY ส่วนหนึ่งให้เป็นเสมือน BUFFER ให้กับ LCD MODULE อีกที

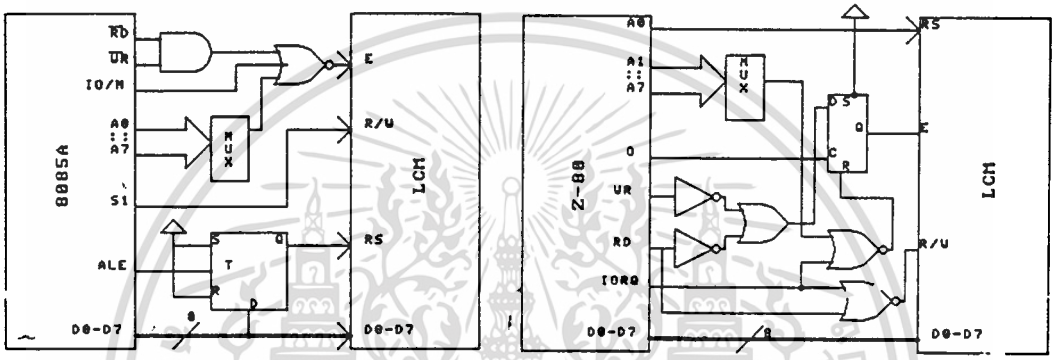
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบไมโครเอง เพื่อรอให้ LCD MODULE กระทำขบวนการต่างๆ

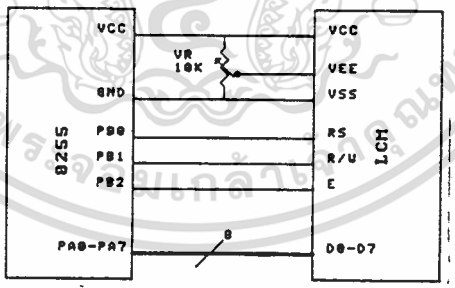
4. ใช้ได้กับบอร์ดทั่วๆ ไปที่มี PORT

5. ไม่เปลืองส่วนของ MEMORY ในการใช้งาน

6. การจัดขาสัญญาณกระทำได้อย่างอิสระ



รูปที่ 2.25 การต่อแบบ MEMORY MAP



รูปที่ 2.26 การต่อแบบ I/O PORT

2.5.1.3 ชุดคำสั่งควบคุมและการแสดงข้อความ

การเขียนหรืออ่านข้อมูลกับ LCD MODULE ก็คือ การกำหนดคุณสมบัติต่างๆ ในการใช้งานของ LCD ตามชุดคำสั่งควบคุม และรวมถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCTION	RS	R/W	DATA BIT 7	6	5	4	3	2	1	0	EXE TIME ( )
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS						40
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS							40
BUSY,ADD. READ	0	1	BF	ADDRESS							0
CGRAM,DDRAM WR	1	0	WRITE DATA								40
CGRAM, DDRAM RD	1	1	READ DATA								40

ตารางที่ 2.7 ชุดคำสั่งควบคุมและการแสดงข้อความ

2.5.1.4 ความเข้าใจพื้นฐาน

- การเขียนข้อมูลให้กับ LCD MODULE จะแบ่งเป็น 2 ลักษณะคือ INSTRUCTION และ DATA โดยจะกำหนดด้วยขาสัญญาณ RS คือถ้า RS=0 จะหมายถึงส่งสัญญาณควบคุม (INSTRUCTION) หรืออ่านค่า FLAG สภาพการทำงานของ LCD MODULE และถ้า RS=1 จะหมายถึงการเขียนหรืออ่าน DATA กับ LCD MODULE

- หลักการในการเขียนข้อมูลให้ LCD MODULE นี้ คือ เมื่อมีการเขียนข้อมูลไปแล้ว ตัว LCD MODULE จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง (ตามค่า EXECUTE TIME ในตาราง) ซึ่งระบบไมโครสามารถตรวจสอบได้จาก BUSY FLAG (BF) และถ้าเรียบร้อยแล้วจึงจะสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบ I/O PORT คือ ไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครก็จะต้องใช้วิธีการหน่วงเวลาแทน

- การเขียนข้อมูลให้กับ LCD MODULE นี้ สามารถทำได้ทั้งแบบ 8 BIT และ 4 BIT โดยกรณี 4 BIT จะใช้สายสัญญาณ DATA เพียง 4 เส้น คือ DB4 - DB7 (ใช้สำหรับระบบไมโครแบบ 4BIT หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 BIT เพียงแต่ให้เขียน 2 ครั้ง คือ DB4 - DB7 ก่อน แล้วตามด้วย DB0 - DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่ง FUNCTION SET ด้วย

- DRAM (DISPLAY DATA RAM) คือ หน่วยความจำภายในตัว LCD MODULE ที่เป็น BUFFER ของข้อมูล โดยถ้าเขียนรหัส ASCII ใดๆ ลงไปในหน่วยความจำนี้ ก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปรากฏเป็นตัวอักษรที่แผงแสดงทันที

- CGRAM (CHARACTER GENERATOR RAM) คือ หน่วยความจำภายในตัว LCD MODULE สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โดยจะอ้าง ADDRESS ได้ทั้งหมด 64 BYTE คือ ตัวอักษร คูณกับ 8 ROW

2.5.1.5 รายละเอียดของแต่ละคำสั่ง

- CLEAR DISPLAY

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

สำหรับการ CLEAR DISPLAY โดยจะทำการเขียนตัวอักษร SPACE ลงใน DDRAM ทั้งหมด และกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะกลับไปที่ตำแหน่งซ้ายบนสุดของจอภาพ

- CURSOR AT HOME

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

สำหรับกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

- ENTRY MODE SET

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

I/D = 0 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ DECREMENT

I/D = 1 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ INCREMENT

S = 0 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะถูกเลื่อนไปทิศทางตามค่า I/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S = 1 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะอยู่กับที่ และตัวอักษรจะถูกดันไปทิศทางตามค่า I/D

การกำหนด I/D และ S นี้ ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้ว จะต้องไม่ใช้คำสั่ง CLEAR DISPLAY อีก

- DISPLAY ON/OFF

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

D = 0 กำหนดให้ OFF DISPLAY

D = 1 กำหนดให้ ON DISPLAY

C = 0 กำหนดให้ OFF CURSOR

C = 1 กำหนดให้ ON CURSOR โดย CURSOR จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่ง CURSOR

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่ง CURSOR (กระพริบเป็นรูปสี่เหลี่ยม)

- DISPLAY SHIFT

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

S/C = 0 กำหนดให้เลื่อน CURSOR ตามทิศทาง R/L ไป 1 ตำแหน่ง

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

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

- FUNCTION SET

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DL = 0 กำหนดให้การติดต่อกับ LCD MODULE เป็นแบบ 4 BIT

DL = 1 กำหนดให้การติดต่อกับ LCD MODULE เป็นแบบ 8 BIT

จะสังเกตุว่าการกำหนดค่า DL นี้ สามารถกระทำได้ที่ DB4 - DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 BIT ตั้งแต่ครั้งแรก หลังจากจ่ายไฟเลี้ยง ก็จะทำให้ LCD MODULE มีการรับข้อมูลแบบ 4 BIT ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 DUTY และ 1/11 DUTY

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 DUTY

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5*7 DOTS

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5*10 DOTS (กรณีที่ LCD MODULE เป็นแบบ 5*7 อยู่แล้ว ก็จะไม่มีผลอะไร)

- SET CGRAM ADDRESS

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

สำหรับการกำหนด ADDRESS ของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที

- SET DDRAM ADDRESS

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

สำหรับการกำหนด ADDRESS ของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที ตำแหน่งของ ADDRESS ในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนอักษรต่อบรรทัดไม่เท่ากัน ซึ่งแสดงดังภาพต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่น DMC202

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

รูปที่ 2.27 ตำแหน่ง ADDRESS ของ LCD MODULE รุ่น DMC 202

รุ่น DMC164

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

รูปที่ 2.28 ตำแหน่ง ADDRESS ของ LCD MODULE รุ่น DMC 164

- BUSY FLAG AND ADDRESS READ

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

สำหรับการอ่านค่า BF (BUSY FLAG) ซึ่งบอกถึงความพร้อมของ LCD MODULE ในการรับข้อมูล ถ้า BF = 0 หมายถึงว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายถึงว่ายังไม่พร้อม นอกจากนี้ยังเป็นการอ่านค่า ADDRESS ของ CGRAM หรือ DDRAM ด้วย

2.5.1.6 การอ่านและเขียนข้อมูลกับ DDRAM / CGRAM

- WRITE DATA TO DDRAM OR CGRAM

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเขียนข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

- READ DATA FROM DDRAM OR CGRAM

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

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

2.5.1.7 แนวทางการเขียนโปรแกรมควบคุม

1. เมื่อจ่ายไปเลี้ยงให้กับ LCD MODULE ครั้งแรก ภายในจะมีการ RESET ระบบโดยอัตโนมัติ ซึ่งจะใช้เวลา 10mS หลังจากที่ระดับแรงไฟขึ้นถึง 4.5V แล้ว ทั้งนี้ระบบ RESET ดังกล่าวจะกระทำสิ่งต่างๆ ดังต่อไปนี้

- ทำการ CLEAR จอภาพทั้งหมด (CLEAR DISPLAY)
- กำหนดคุณสมบัติด้วยคำสั่ง FUNCTION SET คือ DL = 1 (ติดต่อกับระบบไมโครในแบบ 8 BIT) , N = 0 (แสดงข้อมูล 1 บรรทัด) , = 0 (กำหนดตัวอักษร แบบ 5*7 DOTS)
- กำหนดคุณสมบัติด้วยคำสั่ง DISPLAY ON/OFF คือ D = 0 (ไม่แสดงข้อมูล) , C = 0 (CURSOR OFF) , B = 0 (BLINK OFF)
- กำหนดคุณสมบัติด้วยคำสั่ง ENTRY MODE SET คือ I/D = 1 (INCREMENT) , S = 0 (NO SHIFT)

การใช้งาน LCD MODULE ต้องรอให้ขบวนการ RESET ภายในทำงานเรียบร้อยก่อน ซึ่งจะตรวจสอบได้ด้วย BF (BUSY FLAG) หรืออาจจะใช้การหน่วงเวลาก็ได้

2. การใช้งาน LCD MODULE จะเกี่ยวข้องกับทางด้านโปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่างๆ รวมทั้งการอ่านหรือเขียนข้อมูลนั้น จะถูกกำหนดด้วยขาสัญญาณทั้งหมดที่มีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติโปรแกรมจะต้องกำหนดคุณสมบัติต่างๆ ที่ต้องการไว้ที่ส่วนต้น และจากนั้นก็จะเป็นการอ่านและเขียนข้อมูลลงใน DDRAM ซึ่งก็คือข้อความที่จะให้แสดงนั่นเอง

2.6 ทฤษฎีไมโครโปรเซสเซอร์ ตระกูล MCS 51

2.6.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51

(Single Chip Microcontroller system 51 family Architectural)

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) คือไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิพเดี่ยวเหมาะสำหรับงานควบคุมอุปกรณ์อื่น ๆ แบบอัตโนมัติ เพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการ ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51 หรือ MAS 51 อันได้แก่ เบอร์ 8051 และ 8052 ซึ่งมีโครงสร้างและชุดคำสั่งแตกต่างกันเพียงเล็กน้อยดังตารางในรูปที่ 2.29

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
8051	8031	—	4K	128	4	2		✓				
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓				
8052AH	8032AH	8752BH	8K	256	4	3		✓				
80C51BH	80C31BH	87C51	4K	128	4	2		✓				
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓				
83C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓				
83C51GA	80C51GA	87C51GA	4K	128	4	2		✓	✓			8
83C152JA	80C152JA	—	8K	256	5	2		✓		✓	2	
—	80C152JB	—	—	256	7	2		✓		✓	2	
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2	
—	80C152JD	—	—	256	7	2		✓		✓	2	
83C451	80C451	—	4K	128	7	2		✓				
83C452	80C452	87C452P	8K	256	5	2		✓				

รูปที่ 2.29 ตารางของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวในตระกูล 51

จากตารางในรูปที่ 2.29 แต่ละคอลัมน์จะบอกถึงคุณสมบัติหรือโครงสร้างของไมโคร-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทลเลอร์แต่ละเบอร์ในตระกูล MCS51 เช่นมี RAM ถ้าเป็นรุ่นที่ไม่มี ROM อยู่ภายในจะเป็นเบอร์อะไร หรือถ้าเป็นรุ่นที่มีหน่วยความจำสำหรับโปรแกรมเป็นแบบ EPROM จะเป็นเบอร์อะไร เช่นในบรรทัดแรกจะบอกว่า 8051 มี ROM อยู่ภายในขนาด 4 กิโลไบท์ แต่ถ้าเป็นเบอร์ 8031 จะไม่มี ROM ขนาด 4 กิโลไบท์อยู่ภายในนอกจากนี้ในตารางยังจะบอกว่าไมโครคอนโทรลเลอร์นั้นพอร์ทสำหรับอ่านเขียนข้อมูลขนาด 8 บิท อยู่กี่ชุด (8 bit I/O Port) มี Timer/Counters ขนาด 16 บิทกี่ชุด (16 Bit Timer/Counters) และยังบอกถึงคุณสมบัติอื่น ๆ อีก ทำให้ผู้ใช้สามารถเลือกใช้ไมโครคอนโทรลเลอร์แต่ละเบอร์ให้เหมาะสมกับการใช้งานได้อย่างดีที่สุด

MCS-51 ผลิตโดยบริษัท Intel มีการทำงานเป็นแบบ 8 บิท หมายความว่าส่วนที่ทำหน้าที่ในการคำนวณ (Arithmetic Logic Unit, ALU) จะทำงานสูงสุดทีละ 8 บิท

MCS-51 มีข้อดีดังนี้

- สามารถนำเอาข้อมูลมา AND, OR หรือทำComplement ทั้งแบบทีละ 8บิทและ 1บิท

- สามารถใช้กับหน่วยความจำสำหรับโปรแกรม (Program Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับเก็บคำสั่งชุดที่จะให้ MCS-51 ทำงาน ได้สูงสุด 64 กิโลไบท์ (Kilobyte) (61 x 11024 ไบท์) ทำให้เขียนโปรแกรมควบคุมการทำงานได้มาก

- สามารถต่อกับหน่วยความจำสำหรับข้อมูล (Data Merory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรมได้สูงสุด 64 กิโลไบท์

- ใน 8051 และ 8751 มีหน่วยความจำสำหรับโปรแกรมจำนวน 4 กิโลไบท์ (ใน 8052 และ 8752 มีหน่วยความจำสำหรับโปรแกรมจำนวน 8 กิโลไบท์) อยู่ภายในวงจรรวมทำให้ไม่ต้องต่อหน่วยความจำสำหรับโปรแกรมอยู่ภายนอก ระบบรวมทั้งหมดจึงมีขนาดเล็กและสัญญาณรบกวนจากภายนอกจะทำให้ MCS-51 ทำงานผิดพลาดได้ยาก

- มีพอร์ทแบบขนาน (Parallel Port) สำหรับข้อมูลเข้าและออกจำนวน 32 บิท ที่มีข้อมูลอยู่แต่ละบิทเป็นอิสระต่อกัน

- มีวงจร Timer/Counter ขนาด 16 บิท 2 ชุด (8052 มี 3 ชุด) ที่ทำงานในโหมดต่าง ๆ ได้ถึง 4 โหมด

- มี Universal Asynchonous Receiver Transmitter (UART) สำหรับรับ - ส่งข้อมูลอนุกรม (Serial)แบบ Full duplex ที่สามารถเลือกรูปแบบการรับ-ส่งข้อมูลได้ 4 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีแหล่งกำเนิดสัญญาณของขัดจังหวะการทำงานของโปรแกรม (Interupt Request Signal) 6 แหล่ง ซึ่งสามารถทำกระโดดไปทำงานตอบสนองการขัดจังหวะ (Interrupt Service Routine) ได้ต่าง ๆ กัน 5 ตำแหน่ง

- สามารถเลือกการทำงานให้อยู่ในโหมดของ Idell และ Power Down ซึ่งจะประหยัดการใช้กำลังไฟในการทำงาน ซึ่งจากข้อดีดังกล่าว จึงทำให้ MCS-51 เป็นที่นิยมนำมาใช้ในการควบคุมระบบอัตโนมัติมากคุณสมบัติดังกล่าวบรรจุไว้ในวงจรรวมเดี่ยว (Single Chip) ขนาด 40 ขา ดังนั้นจึงสามารถออกแบบให้ระบบทั้งหมดมีขนาดเล็ก และการที่ทั้งหมดบรรจุอยู่ภายในวงจรรวมเดี่ยวจึงทำให้การตรวจสอบหาข้อผิดพลาดในระบบง่ายไม่สลับซับซ้อน รวมทั้งลดปัญหาเรื่องการที่มีสัญญาณรบกวนในระบบจนทำให้การทำงานผิดพลาดไป แต่การที่จะนำเอา MCS-51 มาใช้ในงานได้จำเป็นที่จะต้องศึกษาและทำความเข้าใจถึงโครงสร้างองค์ประกอบของ MCS-51 เสียก่อนแล้วถึงจะเขียนโปรแกรมเพื่อควบคุมการทำงานของ MCS-51 โดยใช้ 8051 เป็นตัวอธิบาย เพราะไมโครคอนโทรลเลอร์ในตระกูลนี้จะแตกต่างกันเพียงเล็กน้อยเท่านั้น

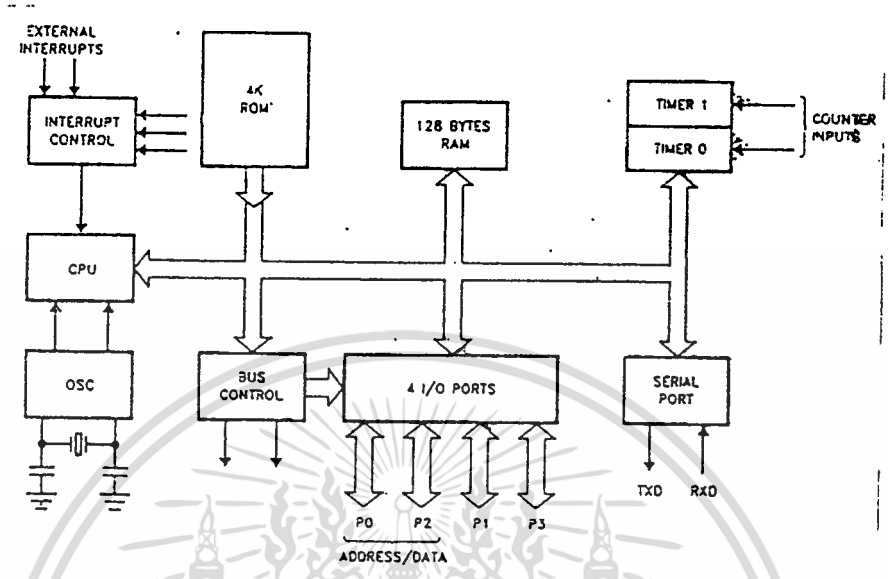
2.6.2 โครงสร้างของ 8051

ภายใน 8051 จะประกอบขึ้นด้วย GATE ต่าง ๆ เช่น AND, OR, NOT ซึ่ง GATE เหล่านี้จะถูกนำเอามาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่นวงจรถอดรหัสคำสั่ง (Insturction Decoder) วงจรสร้างสัญญาณนาฬิกา (Clock Signal Generator) โครงสร้างภายในของ 8051 จะประกอบขึ้นด้วยส่วนย่อย ๆ ดังไดอะแกรมในรูปที่ 2.30

ไดอะแกรมในรูปที่ 2.30 เป็นโครงสร้างใหญ่ ๆ ของ 8051 เนื่องจากลักษณะของ 8051 เป็นคอมพิวเตอร์จึงประกอบด้วย 3 ส่วนหลัก ๆ คือ

ส่วนที่ 1 คือ CPU (Central Processing Unit) หรือตัวประมวลผล ส่วนนี้จะมีวงจรที่ทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่น ๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่สัญญาณสำหรับการติดต่อกับหน่วยความจำ,อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออกจากตัว 8051 ซึ่งส่วนควบคุมด้วยการสร้างสัญญาณควบคุมจากส่วน CPU นี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่ง (Instrution) ตามที่กำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์เพื่อให้ทุก ๆ ส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

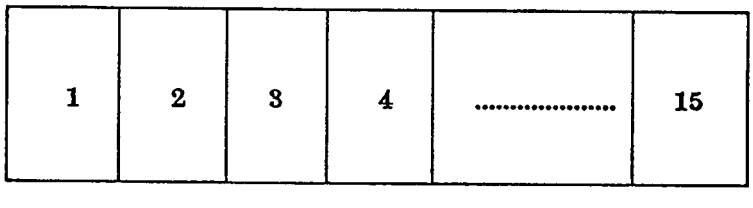
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 ไดอะแกรมโครงสร้างของ 8051

ใน CPU นี้ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (Arithmetic Logic Unit) ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูลเช่น การบวก, ลบ, คูณ หรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์หรือหน่วยความจำที่ต้องการ

ส่วนที่ 2 คือ หน่วยความจำ (Memory) มีไว้สำหรับจดจำข้อมูล ถ้าจะให้เห็นภาพพจน์ของหน่วยความจำได้ดีก็คือ หน่วยความจำเปรียบเหมือนกล่องเก็บเอกสารจำนวนมากที่นำมาต่อเรียงกันไว้ แต่ละกล่องก็มีเอกสาร 1 แผ่น ดังรูปที่ 2.31 มีกล่องเอกสารทั้งหมด 15 กล่อง



รูปที่ 2.31 ภาพเสมือนของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการเอาเอกสารจากกล่องใด หรือเอาเอกสารไปเก็บที่กล่องใด จะต้องรู้หมายเลขของกล่องข้อมูลเสียก่อนซึ่งถ้าเป็นหน่วยความจำ แล้วหมายเลขกล่องก็คือตำแหน่งความจำหรือแอสเดรส (Address) นั่นเอง การเอาข้อมูลไปเก็บในหน่วยความจำเรียกว่าการเขียน (Write) ข้อมูล และการเอาข้อมูลออกจากหน่วยความจำจะเรียกว่าการอ่าน (Read) ข้อมูล ซึ่งแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลได้เพียงค่าเดียวเท่านั้น ในไมโครโปรเซสเซอร์ทั่วไป รวมทั้ง 8051 นั้นข้อมูลในแต่ละตำแหน่งของหน่วยความจำจะมีค่าได้เพียง 8 หลัก ของเลขฐาน 2 (8 บิท เท่ากับ 1 ไบท์) แต่จำนวนตำแหน่งที่จะเก็บข้อมูลมีค่าได้ระหว่าง 0 ถึง 255 (00000000 ถึง 1111111 ในเลขฐาน 2) แต่จำนวนตำแหน่งที่จะเก็บข้อมูลได้ขึ้นกับไมโครโปรเซสเซอร์แต่ละเบอร์ การติดต่อกับหน่วยความจำจะต้องมีสัญญาณ 3 กลุ่ม คือ

- แอสเดรสหรือค่าตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ใน 8051 จะติดต่อกับหน่วยความจำประเภท Program Memory หรือ Data Memory ได้สูงสุดชนิดละ 65536 ตำแหน่ง ดังนั้นการอ้างอิงแต่ละตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น (2^{16} เท่ากับ 64 x 1024)
- ข้อมูลที่จะอ่านหรือเขียนกับหน่วยความจำที่ตำแหน่งในข้อ 1
- สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำเพื่อบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล

สัญญาณเหล่านี้จะถูกวงจรควบคุมภายใน 8051 สร้างมาจากวงจรถอดรหัสของคำสั่งที่ 8051 อ่านจากหน่วยความจำ Program Memory เข้าไปทำงานนั่นเอง ในรูปที่ 2.29 หน่วยความจำได้แก่ 4K ROM และ 128 Byte RAM ซึ่งขนาดของหน่วยความจำนี้มีขนาดต่าง ๆ กันตามเบอร์ไมโครคอนโทรลเลอร์ และจะอธิบายโดยละเอียดในข้อ 2.6.3

ส่วนที่ 3 อุปกรณ์อินพุทและเอาท์พุท (Input/Output Device) เป็นส่วนที่จะใช้ส่งข้อมูลเข้าหรือออกจาก 8051 ทำให้ 8051 ติดต่อกันภายนอกได้ ดังในไดอะแกรมรูปที่ 2.29 อุปกรณ์อินพุทและเอาท์พุทได้แก่ 4 I/O Port, Timer 0, Timer 1, Serial Port การทำงานของแต่ละส่วนมีดังนี้

- 4 I/O Port คือ คำว่าพอร์ทหมายถึงจุดที่จะติดต่อกับส่วนที่อยู่ภายนอก 4 I/O Port ของ 8051 เป็นที่ใช้สำหรับรับ-ส่งข้อมูลซึ่งเป็นสัญญาณดิจิตอลเข้าหรือออกจากตัว MCS-51 พอร์ทมีทั้งหมด 4 พอร์ท โดยแต่ละพอร์ทจะรับ-ส่งข้อมูลได้ 8 บิท มีพอร์ท P0,P1,P2 และ P3 บางพอร์ทจะใช้ทำงานมากกว่า 1 อย่างก็ได้ เช่น พอร์ท P0 และ P2 จะใช้รับส่งข้อมูล เมื่อติดต่อกับหน่วยความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดขึ้นที่เวลาเดียวกัน แต่จะใช้วิธีการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามลำดับโดยควบคุมจากสัญญาณควบคุม (Control) ที่ถอดรหัสมาจากแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงกับจากสัญญาณนาฬิกา

- Timer 0 และ Timer 1 เป็นวงจรนับที่สามารถกำหนดให้ทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอก 8051 หรือจำนวนไซเคิลของสัญญาณนาฬิกาภายใน 8051 ก็ได้ ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย CPU

- Serial Port หรือพอร์ทอนุกรม CPU จะอ่านและเขียนข้อมูลกับ Serial Port เป็นแบบ 8 บิท แต่ข้อมูลจะถูกส่งออกจาก 8051 เรียงไปทีละบิทออกจากขา TXD และในการรับข้อมูลเข้าก็จะรับเข้ามาทีละบิททางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิท เพื่อให้ CPU อ่านไปใช้งานต่อไป

8051 มีพอร์ทให้ใช้งานได้หลายแบบทำให้สะดวกแก่การนำไปใช้งานต่าง ๆ มากมาย การจะนำพอร์ทเหล่านี้ไปใช้งานได้จะต้องเขียนโปรแกรมขึ้นมาควบคุมที่จะได้กล่าวต่อไป

2.6.3 การจัดการหน่วยความจำของ 8051

หน่วยความจำของ 8051 แบ่งออกไว้เป็น 2 แบบ ตามลักษณะของการใช้งานคือ

2.6.3.1 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 volatite ซึ่งเมื่อปิดไฟแล้วข้อมูลก็ไม่มีการสูญหาย) 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 กิโลไบท์ นั้นผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอก 8051 (ไมโครคอนโทรลเลอร์เบอร์อื่น ๆ เช่น 8052 จะมีขนาดของ ROM ส่วนนี้ได้ถึง 8 กิโลไบท์ ตำแหน่ง 0000H ถึง 1FFFH) ถ้าต้องการให้ 8051 ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน 8051 ก็ให้ป้อนสัญญาณสภาวะลอจิก High (1) เข้าที่ขา EA ของ 8051 แต่ถ้าต้องการให้ทำงานในโปรแกรมที่เก็บไว้ใน ROM ภายนอก 8051 ก็ให้ต่อลอจิก

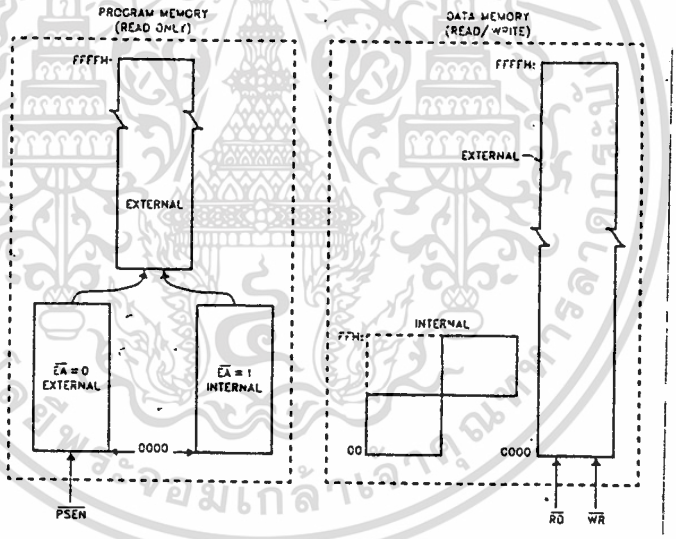
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low (0) เข้าที่ขา EA ของ 8051 ส่วนหน่วยความจำที่ตำแหน่ง 1FFFH ถึง FFFFH จะต้องอยู่ภายนอก 8051 เสมอดังแสดงในแผนภูมิหน่วยความจำ (Memory Map) ในรูปที่ 2.31

Internal Memory หมายถึง หน่วยความจำนั้นอยู่ภายใน 8051 นั้น โดยโครงสร้างและรหัสคำสั่งจะเหมือนกันทุกประการแตกต่างกันที่

- 8031 จะไม่มี ROM ขนาด 4 กิโลไบท์อยู่ภายใน ผู้ใช้จะต้องเลือกการใช้งาน Program Memory อยู่ภายนอกวงจรรวมทั้งหมด 64 กิโลไบท์

- 8051 จะมี ROM ขนาด 4 กิโลไบท์อยู่ภายใน ถ้าต้องการเก็บคำสั่งควบคุมการทำงานไว้ในหน่วยความจำส่วนนี้ จะต้องส่งโปรแกรมคำสั่งไปให้โรงงานผู้ผลิตทำการเขียนใส่ใน ROM ให้ตั้งในขั้นตอนของการผลิตวงจรรวม ผู้ใช้ไม่สามารถแก้ไขโปรแกรมได้เอง โดยการต่อ ROM ไว้ภายนอก แล้วต่อขา EA ของ 8051 ไว้กับสัญญาณที่มีสภาวะลอจิกเป็น 0



รูปที่ 2.32 แผนภูมิหน่วยความจำของ 8051

- 8751 จะมีหน่วยความจำขนาด 4 กิโลไบท์ เป็น ERPOM (Erasable Program Read Only Memory) อยู่ภายในวงจรรวมเอาไว้ ใช้เก็บโปรแกรมคำสั่งที่จะให้ 8751 ทำงาน ผู้ใช้สามารถเขียนคำสั่งลงไปใน EPROM ได้เองโดยใช้เครื่องมือที่เรียกว่าเครื่องโปรแกรม EPROM (EPROM Programmer) และผู้ใช้สามารถแก้ไขโปรแกรมที่อยู่ใน EPROM ได้โดยการล้างข้อมูลในทุกตำแหน่งของ EPROM ออกด้วยการฉายแสงอุลตร้าไวโอเลต (Ultraviolet) ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระจกใสบนวงจรรวมเข้าไปยังวงจรภายใน ตามเวลาที่กำหนดในคู่มือเฉพาะ (Data sheet) ของ 8751 จากนั้นก็ใช้เครื่องโปรแกรม EPROM เขียนโปรแกรมลงไปใหม่ 8751 นี้จะแสดงมากสำหรับการพัฒนาโปรแกรม

2.6.3.2 Data Memory เป็นหน่วยความจำที่ 8051 จะใช้สำหรับพัก, เก็บข้อมูล แล้วเรียกมาใช้ใหม่ในระหว่างการทำงานของ 8051 การอ่านหรือเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ใน Program Memory หน่วยความจำแบบนี้เป็นประเภท RAM (Random Access Memory) ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้จะไม่สูญหาย แต่ถ้าปิดเครื่องไม่จ่ายไฟให้แก่ RAM แล้ว ข้อมูลใน RAM ก็จะสูญหายไป การสูญหายของข้อมูลไม่ได้หมายความว่าไม่มีอยู่เลยแต่เป็นการที่มีข้อมูลใหม่ซึ่งไม่ใช่ข้อมูลที่เก็บไว้เดิมเข้ามาอยู่แทนที่ เช่นเดิมเก็บข้อมูล 18H ไว้ที่ตำแหน่ง 1900H เมื่อปิดไฟแล้วเปิดใหม่ข้อมูลที่ตำแหน่ง 1900H จะไม่ใช่ 18H อาจเป็นค่าอะไรก็ได้ ซึ่งเรียกการเกิดลักษณะแบบนี้ว่าข้อมูลสูญหายไป หน่วยความจำแบบ Data Memory ของ 8051 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายใน 8051 จำนวน 128 ไบท์ที่ตำแหน่ง 00H ถึง 9FH (เบอร์ 8052 จะมี 256 ตำแหน่ง 00H ถึง FFH) และอีกชุดหนึ่งจะต้องต่ออยู่ภายนอกของวงจรรวม 8051 มีได้สูงสุด 65536 ไบท์ (64 กิโลไบท์) อยู่ที่ตำแหน่ง 0000H ถึง FFFFH ดังแสดงในรูปที่ 31 หน่วยความจำแบบ Data Memory ภายใน 8051 ที่ตำแหน่ง 80H ถึง FFH นั้นไม่ได้มีอยู่ทุกตำแหน่ง จะมีเฉพาะในบางตำแหน่งซึ่งเรียกหน่วยความจำบางตำแหน่งนี้ว่า Special Function Register (SFR) เพราะจะใช้หน่วยความจำเหล่านี้สำหรับงานพิเศษเท่านั้น แต่ละตำแหน่งของหน่วยความจำแบบ SFR นี้อาจเป็น RAM หรือวงจรนับ (Counter) วงจรตั้งเวลา (Timer) ก็ได้เช่นเป็น Timer 0, Timer 1 ดังนั้นใน 8051 จึงไม่ถือว่า SFR เป็น Data Memory ถ้าเป็น 8052 ซึ่งมี Data Memory ขนาด 256 ไบท์ จะใช้บางตำแหน่งของหน่วยความจำช่วงตำแหน่ง 80H ถึง FFH เป็น SFR ส่วนตำแหน่งอื่นที่เหลือก็เป็น RAM เหมือนกับหน่วยความจำช่วง 00H ถึง 7FH นั่นเอง

2.6.4 สถาปัตยกรรมของ 8051

ในตอนนี้ ได้กล่าวถึงไดอะแกรมภายในของ 8051 อย่างกว้าง ๆ ซึ่งพอจะบอกได้โดยสังเขปว่าประกอบด้วยส่วนใหญ่ ๆ อะไรบ้าง ในรูปที่ 2.33 เป็นสถาปัตยกรรมภายในของ 8051 ซึ่งจะอธิบายถึงส่วนย่อย ๆของภายใน 8051 เพียงชิพเดียว และสัญญาณจากภายในจะต่อสู่ภายนอกทางขา (Pin) ของ 8051 ที่มีอยู่ 40 ขา ดังรูปที่ 2.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8051 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 มีขาอยู่ข้างละ 20 ขานั้นจะใช้งานต่าง ๆ กันดังนี้คือ

Vcc

ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลท์เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลเตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

Vss

ขา 20 เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน

Port 0

เป็นพอร์ทขนานขนาด 8 บิท อยู่ที่ขา 39 ถึง 32 เริ่มจากบิท 0 ถึง บิท 7 ตามลำดับในรูปที่ 2.41 แต่ละขาจะเขียนว่า P0.0, P0.1,.... P0.7 นั้น P0.7 หมายถึงบิท 7 ของพอร์ท 0 ซึ่งเป็นบิทที่มีนัยสำคัญสูงสุด (Most Significant) และ P0.0 คือ บิท 0 ของพอร์ท 0 เป็นบิทที่มีนัยสำคัญต่ำสุด (Least Significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับ - ส่ง ตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ทรับ - ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ท 0 จะถูก Latch ไว้ที่ขาของพอร์ท โครงสร้างแต่ละบิทของพอร์ท 0 เป็นแบบ Open Drain Bidirectional ดังรูปที่ 2.35

ในรูปที่ 2.35 เมื่อเปรียบเทียบกับรูปที่ 2.33 ส่วนที่ 1 ของรูป 2.35 ก็คือ Port 0 Latch ในรูปที่ 2.33 และส่วนที่ 2 ของรูป 2.35 ก็คือ Port 0 Driver ของรูปที่ 2.33 นั่นเอง

จากโครงสร้างในรูปที่ 2.35 เมื่อมีคำสั่งการเขียนข้อมูลมายังพอร์ท 0 ข้อมูลจาก Internal Data Bus จะถูก Latch ไว้ที่ D-FF โดยสัญญาณ "Write to Latch" ที่ถูกสร้างมาจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ท 0 จะอ่านได้ 2 แบบ คือการอ่านข้อมูลที่ส่งไปไว้ที่พอร์ทก็จะมีสัญญาณ Read Latch มาเพื่ออ่านข้อมูลจาก D-FF กลับเข้ามายัง Internal Data Bus การอ่านข้อมูลอีกแบบก็คือการอ่านสภาวะของสัญญาณที่เข้ามาทางพอร์ท 0 ก็จะมีสัญญาณ Read Pin มาควบคุมการอ่านพอร์ท 0 จะใช้งานหลายอย่างดังนี้

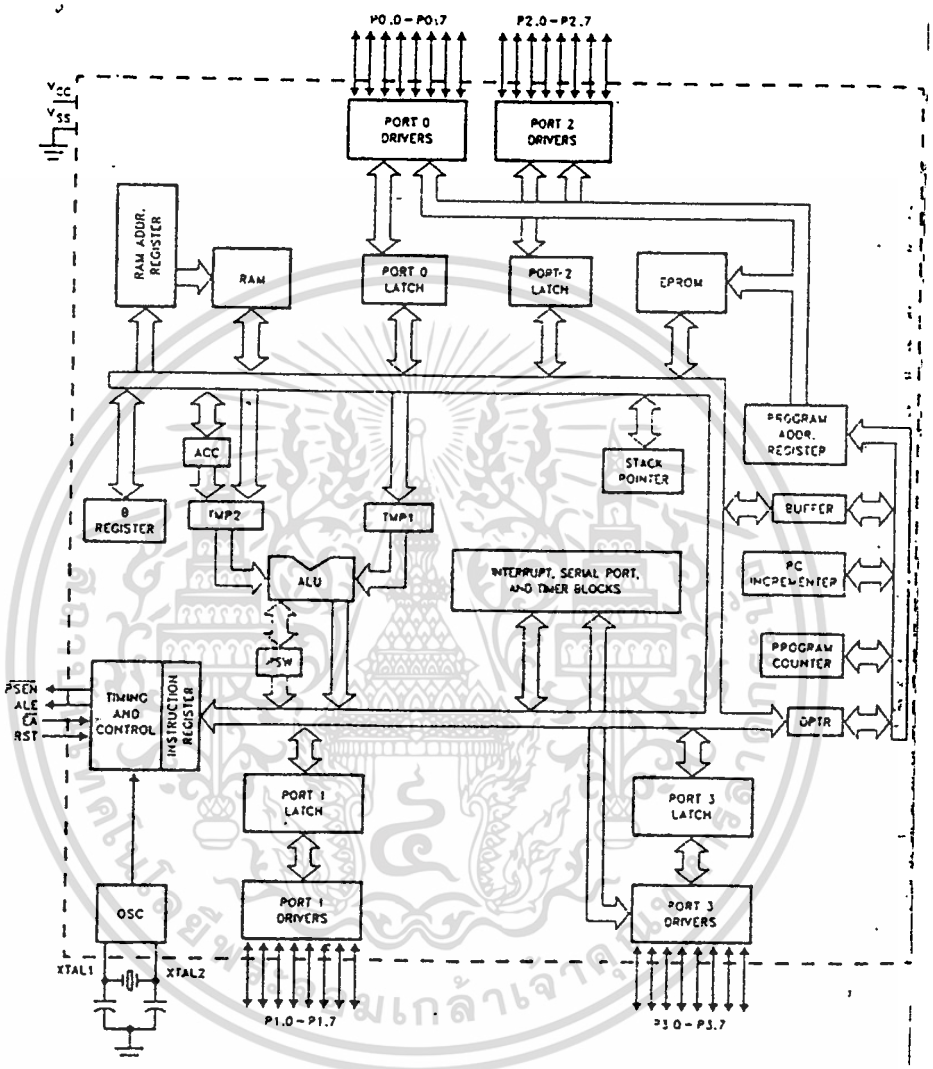
1. ใช้สำหรับส่งตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อด้วย ตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อได้ก็คือ 64 kbyte จึงมีค่าตำแหน่งหน่วยความจำ 16 บิท ของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิทล่างจะถูกส่งไปทางพอร์ท 0 และ 8 บิทบนจะส่งออกไปทางพอร์ท 2

2. ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory

3. ใช้รับ - ส่งข้อมูลผ่านทางพอร์ทโดยตรง ในกรณีที่ไม่มีการใช้หน่วยความจำของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

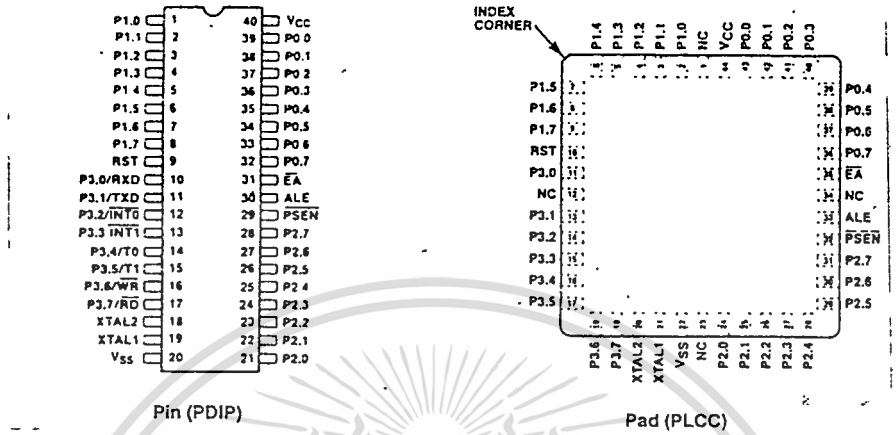
Program Memory หรือ Data Memory ภายนอก



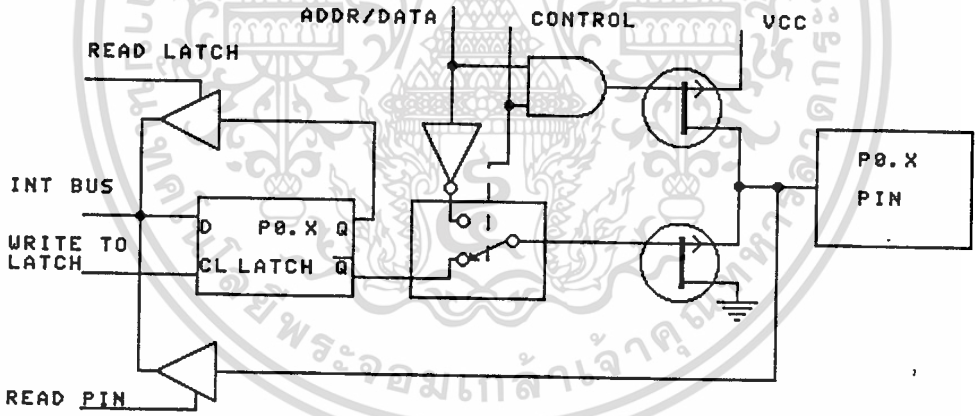
รูปที่ 2.33 สถาปัตยกรรมภายในของ 8051

วงจรภายในส่วน Timing and Control จะเป็นตัวสร้างสัญญาณมาควบคุมวงจร ในรูปที่ 2.35 เพื่อให้การทำงานแต่ละอย่างข้างต้น เมื่อแต่ละบิทของพอร์ท 0 ทำงานตามข้อ 1 และ 2 ข้างต้น วงจร Timing and Control จะทำให้สภาวะลอจิกของขา Control เป็น 1 ซึ่งทำให้สวิทช์ MUX อยู่ในตำแหน่งข้างบน เมื่อพอร์ท 0 จะถูกส่งข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำ หรือข้อมูลที่จะเขียนออกไปยังหน่วยความจำภายนอก ก็จะส่งค่าดังกล่าวมายัง ADDR/DATA ถ้าข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 ไดอะแกรมขาของ 8051 แบบ DIP



รูปที่ 2.35 โครงสร้างของพอร์ท 0

มูลที่ส่งมาเป็น 1 จะทำให้สัญญาณออกจาก AND GATE เป็น 1 และสัญญาณที่ออกจาก Inverter เป็น 0 ดังนั้น FET ตัวบน ON (สภาวะ ON ของ FET คือความต้านทานระหว่างขา D กับ S มีค่าสูงมากเสมือนกับเป็นวงจรเปิด) สภาวะลอจิกที่ขา PO.X PIN จะเป็น 1 แต่ถ้าข้อมูลที่ส่งออกมาขัง ADDR/DATA เป็น 0 ก็จะทำให้สัญญาณจาก AND GATE เป็น 0 และสัญญาณที่ออกจาก Inverter เป็น 1 ดังนั้น FET ตัวบนจะ OFF ส่วน FET ตัวล่างจะ ON ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

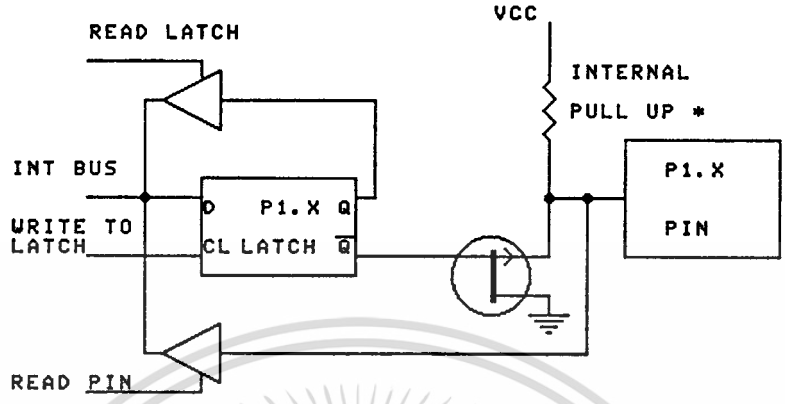
สภาวะลอจิกที่ขา P0.X Pin เป็น 0 เมื่อ 8051 ต้องการใช้พอร์ท 0 สำหรับการอ่านข้อมูลจากหน่วยความจำภายนอก หรือใช้ทำงานในข้อ 3 ข้างบน ก็จะทำได้โดยวงจร Timing and Control ทำให้สภาวะลอจิกของสัญญาณ Control ในรูปเป็น 0 ทำให้เอาท์พุทจาก AND GATE เป็น 0 FET ตัวบนจะ OFF และสวิทช์ MUX จะอยู่ในตำแหน่งข้างล่าง ดังนั้น FET ตัวล่างจะ ON หรือ OFF ก็แล้วแต่ขาแต่ละข้อมูลที่ขา Q ของ D-FF เมื่อมีการเขียนข้อมูลจาก Internal Data Bus มายัง D-FF ก็จะมีสัญญาณ Write to Latch มายัง D-FF ด้วย ถ้าข้อมูลที่เขียนมาเป็น 1 ก็จะทำให้ขา Q มีสภาวะลอจิกเป็น 0 ทำให้ FET ตัวล่าง OFF ดังนั้นขา P0.X จะอยู่ในภาวะอิมพีแดนซ์สูง (High Impedance) เพราะ FET ทั้ง 2 ตัว OFF แต่ถ้าข้อมูลที่เขียนมายัง D-FF เป็น 0 จะทำให้ FET ตัวล่าง ON แต่ตัวบน OFF ทำให้สภาวะลอจิกที่ขา P0.X เป็น 1 ดังนั้น Port 0 เมื่อทำงานเป็นพอร์ทส่งข้อมูล (ไม่ใช่ส่งตำแหน่งหน่วยความจำ) จะไม่สามารถแสดงสภาวะลอจิก 1 ได้จึงต้องต่อตัวต้านทาน Pull Up ไว้ภายนอก ระหว่างขา P0.X กับไฟเลี้ยงวงจรถ้าจะใช้พอร์ท 0 สำหรับข้อมูลเข้าจะต้องเขียน 1 มาเก็บไว้ยัง D-FF เสียก่อนเพื่อให้ขา P0.X อยู่ในสภาวะ High Impedance แล้วจึงใช้คำสั่งอ่านสภาวะลอจิกเข้าไปยัง Internal Data Bus ต่อไป โดยคำสั่งอ่านสภาวะลอจิกทางพอร์ท 0 ก็จะทำให้วงจร Timing and Control สร้างสัญญาณ Read Pin สำหรับการอ่านสภาวะลอจิกข้างต้น ถ้าไม่เขียน 1 มาเก็บไว้ยัง D-FF ก่อนที่จะอ่านข้อมูลแล้วอาจมีข้อมูลค้างอยู่ที่ D-FF ทำให้ Q เป็น 0 และ Q เป็น 1 ซึ่งทำให้ FET ตัวล่าง ON สัญญาณที่ต่อเข้ามาที่ขา P0.X ไม่ว่าจะมีสภาวะลอจิกใดจะถูกดึงลงกราวด์ ดังนั้นเมื่ออ่านข้อมูลเข้าไปก็จะพบว่าเป็น 0 เสมอ ในการอ่านข้อมูลเข้าไปก็จะพบว่าเป็น 0 เสมอ ในการอ่านข้อมูลจากหน่วยความจำภายนอกนั้นวงจร Timing and Control ก็จะเขียนข้อมูลมายัง D-FF ให้เป็น 1 และสร้างสัญญาณ Control ให้มีลอจิกเป็น 0 ก่อนจะอ่านข้อมูลเข้าไปด้วย

Port 1

เป็นพอร์ทขนานขนาด 8 บิท ในรูปที่ 2.34 คือขา P1.0 ถึง P1.7 ขา (ขา 1-8) P1.0 หมายถึงบิท 0 ของพอร์ท 1 ซึ่งเป็นบิท Least Significant Bit และบิท P1.7 หมายถึงบิทที่ 7 ของพอร์ท 1 ซึ่งเป็นบิท Most significant bit โครงสร้างของพอร์ท 1 แต่ละบิทมีดังรูปที่ 2.36

ส่วนที่ 1 คือ Port 1 Latch ในรูปที่ 2.33 ซึ่งจะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ทที่ 0 ในรูปที่ 2.34 ส่วนที่ 2 คือ Port 1 Driver ในรูปที่ 2.33 Port 1 driver นี้จะมีตัวต้านทานต่ออยู่เป็น Internal Pull พอร์ท 1 นี้จะทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ท 1 จะถูก Latch ไว้แล้วส่งออกไปทางแต่ละขา ก่อนที่จะอ่านข้อมูลเข้าไปทางพอร์ท 1 จะต้องเขียน 1 ไปยังทุกบิทของพอร์ท 1 เสียก่อนเพื่อให้ FET อยู่ในสภาวะ OFF ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

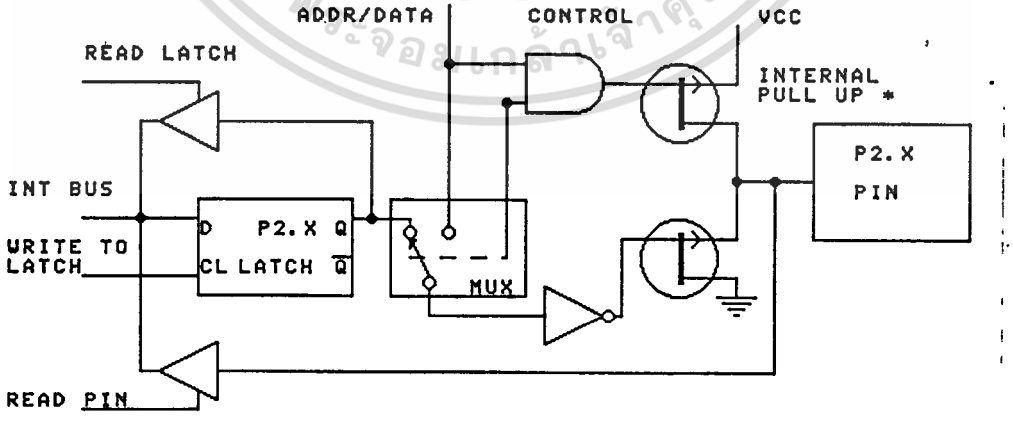


รูปที่ 2.36 โครงสร้างของพอร์ท 1

มิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งออกมาค้างอยู่ที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่ขานี้ก็จะถูกลัดวงจรลงกราวด์ โดยไม่สนใจว่าสภาวะลอจิกของสัญญาณที่เข้ามาจะเป็นอะไร ข้อมูลที่อ่านเข้าไปจึงจะเป็น 0 เสมอ

Port 2

พอร์ทขนานขนาด 8 บิท คือขา P2.0 ถึง P2.7 (บิท 0 ถึง 7 ของพอร์ท 2) ในรูปที่ 2.34 โครงสร้างของพอร์ท 2 แต่ละบิทจะมีดังรูปที่ 2.35



รูปที่ 2.37 โครงสร้างของพอร์ท 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

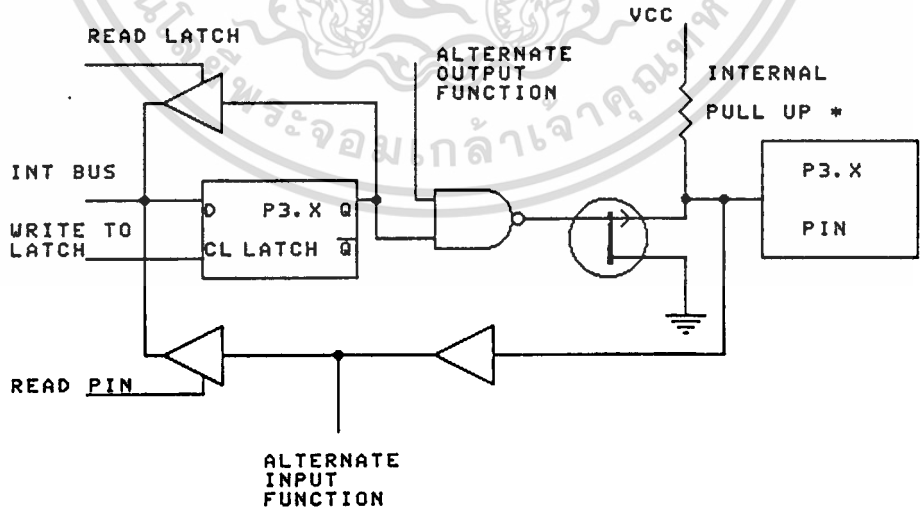
ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันที่โน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะ คือ

- ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิท บนของค่าตำแหน่ง

- ใช้เป็นพอร์ทรับและส่งข้อมูลภายนอก ดังนั้นภาค Driver ของพอร์ท 2 จึงแตกต่างจาก Driver ของพอร์ท 0 โดยที่ในพอร์ท 2 นั้นจะมีเฉพาะ ADDR (ตำแหน่งหน่วยความจำ) เข้ามาที่ MUX (Multiplexer) เท่านั้น นอกนั้นแล้วการทำงานจะเหมือนกันและที่เอาท์พุทของพอร์ท 2 จะมี Internal pull-up ซึ่งเป็นตัวต้านทานและจะทำให้เอาท์พุทของพอร์ท 2 แสดงสถานะลอจิกเป็น 1 ได้ ถ้า FET อยู่ในสภาวะ OFF บางครั้งเรียกว่า "Quasi-bidirectional" เมื่อใช้เป็นพอร์ทอินพุทก็สามารถทำได้โดยการต่อสัญญาณภายนอกเข้ามาโดยตรง ถ้าสัญญาณภายนอกเป็น 0 ก็จะมีกระแสไหลออกจากพอร์ท (Source Current) ในการที่จะใช้พอร์ทนี้เป็นพอร์ทรับข้อมูลเข้า จะต้องเขียน 1 ไปยังแต่ละบิทของพอร์ทเสียก่อน ดังได้อธิบายในเรื่อง Port 0 และ Port 1

Port 3

คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับในรูปที่ 2.34 พอร์ทนี้มีโครงสร้างดังรูปที่ 2.38



รูปที่ 2.38 โครงสร้างของพอร์ท 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 ในรูปที่ 2.38 เป็นส่วน Latch ข้อมูลที่เขียนมายังพอร์ท 3 ทาง Internal Bus เหมือนกับพอร์ทอื่น ๆ และพอร์ท 3 จะมี Internal pull up อยู่ทุกบิท แต่พอร์ท 3 นี้แต่ละบิทจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Alternative Output Function ที่สร้างมาจากส่วน Timing and Control สัญญาณ Alternative Output Function เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ท 3 ทำงานในฟังก์ชั่นอื่น และจุด Alternative Input Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่นตามต้องการทำงานของบิทนั้น แต่ละบิทของพอร์ท 3 จะมีฟังก์ชั่นอื่นดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/INTO (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/INTI (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/TO (Timer/Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Couter 0 ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ TO นี้หรือสัญญาณนาฬิกาก็ได้

P3.5/T1 (Timer/Counter 1 External) ขารับสัญญาณเข้าไปยัง Timer/Counter ซึ่งมีการทำงานเหมือนกับ TO

P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

RST

ขารีเซทขานี้จะใช้ทำงานการรีเซทการทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ (Ground) ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการรีเซทการทำงานของ 8051 ดังนั้นจึงสามารถต่อตัวเก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง +5 โวลท์ เพื่อให้เกิดการรีเซทเมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power on reset การรีเซทจะทำให้ค่าในการรีจิสเตอร์ต่าง ๆ เปลี่ยนไปเป็นค่าหนึ่งดังในตารางรูปที่ 2.39

ในตารางรูปที่ 2.39 ช่องทางขวาเป็นค่าของรีจิสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการรีเซท ในรีจิสเตอร์ SBUF เมื่อสิ้นสุดการรีเซทจะมีค่าที่ไม่แน่นอน และพอร์ทจะอยู่ในสภาวะลอจิก 1 ทุกบิทตลอดเวลาที่สัญญาณของขา RST เป็น HIGH อยู่ เมื่อสัญญาณที่ขา RST กลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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
THO	00H
TLO	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H

รูปที่ 2.39 ค่าของรีจิสเตอร์เมื่อเกิดการรีเซท 8051

เป็น 0 ก็จะออกจากการรีเซท 8051 จะเริ่มทำงานจากคำสั่งที่อยู่ใน Program memory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงานถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้จะต้องเขียนโปรแกรมไว้ที่ตำแหน่ง 0000H ในเครื่องไมโครคอมพิวเตอร์แบบบอร์ดเดี่ยว (Single Board Microcomputer) จะมีโปรแกรมที่เขียนเก็บไว้เริ่มจากตำแหน่ง 0000H นี้เรียกว่ามอนิเตอร์โปรแกรม (Monitor program) ที่จะคอยรับการกดแป้นพิมพ์ (Keyboard) และแสดงผลทางตัวแสดงผล (Display) แบบ 7 Segment

ALE

Address Latch Enable ขานี้จะส่งสัญญาณที่ความถี่ 1/6 เท่าของสัญญาณนาฬิกาจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออสซิลเลเตอร์สัญญาณนี้จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอก 8051 ว่าขณะนี้สัญญาณนี้ Active (เป็นลอจิก) จะมีการส่งข้อมูลที่เป็น 8 บิทล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Lacth ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท

PSEN

Program Store Enable เป็นขาที่ 29 ในรูปที่ 2.34 ขานี้จะปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปรแกรมภายใน 8051 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา PSEN นี้สามารถต่อไปยังขาอินพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท

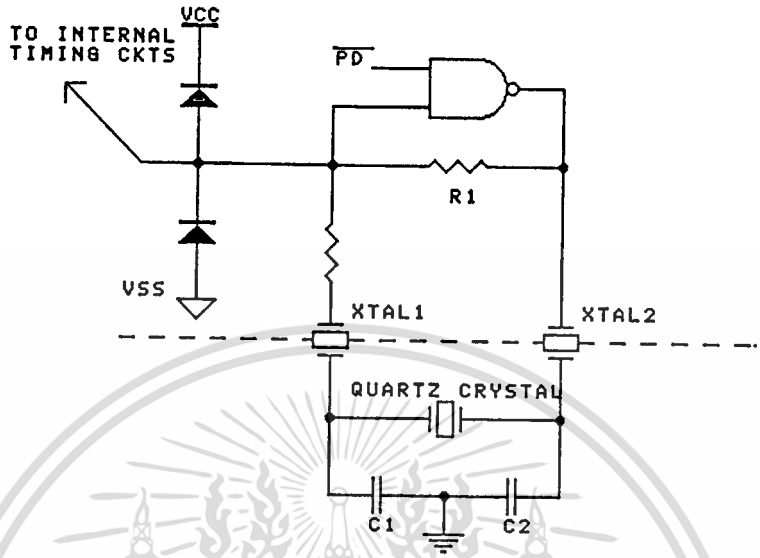
EA

External Access ขา 31 ของรูปที่ 2.34 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control ในรูปที่ 2.33 เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 8051 จะต้องสร้างสัญญาณ PSEN ออกไปยังภายนอก เพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่สัญญาณที่ป้อนให้ขา EA เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน 8051 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่าง ๆ จาก ROM ภายใน 8051

XTAL 1

ขาที่ 19 ของรูปที่ 2.34 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 2.40 จะเห็นวงจรภายในของออสซิลเลเตอร์ NAND GATE จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิลเลตหรือไม่ก็ขึ้นกับสัญญาณ PD ซึ่งต่อมาจากบิท PD ของรีจิสเตอร์ PCON ถ้าต้องการใช้สัญญาณนาฬิกา (Clock Signal) จากภายนอกมาเป็นสัญญาณนาฬิกาควบคุมการทำงานของ 8051 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อ Crytal หรือเซรามิคเรโซเนเตอร์ดังรูปที่ 2.40 คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 PF

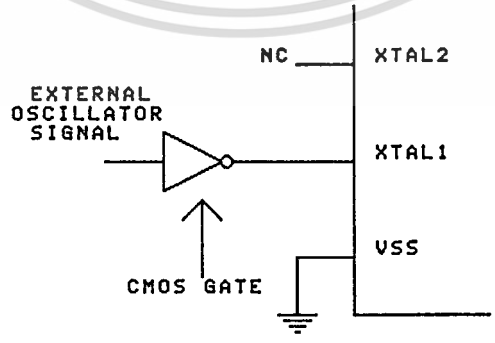
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.40 วงจรออสซิลเลเตอร์ภายใน 8051

XTAL 2

ขาที่ 18 ของรูปที่ 2.34 ขานี้เป็นจุดเอาท์พุทของวงจรขยายแบบกลับเฟสสัญญาณที่ประกอบเป็นวงจรออสซิลเลเตอร์ (อินพุท คือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกออกมาเป็นสัญญาณนาฬิกาของ 8051 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 2.41



รูปที่ 2.41 8051 ทำงานโดยสัญญาณที่มาจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5 การทำงานของ 8051

คอมพิวเตอร์จะทำงานด้วยวงจรที่เรียกว่าฮาร์ดแวร์ (Hardware) ประกอบขึ้นมาเพียงอย่างเดียวไม่ได้จะต้องมีโปรแกรมหรือคำสั่งที่จัดเรียงกันไว้ให้คอมพิวเตอร์ทำงานตามลำดับใน 8051 ก็เช่นกัน ผู้ใช้จะต้องเขียนโปรแกรมเป็นภาษาเครื่อง ซึ่งอยู่ในรูปของเลขฐาน 2 เก็บไว้ในหน่วยความจำประเภท Program Memory แต่ละคำสั่งของ 8051 อาจประกอบด้วย 1,2 หรือ 3 ไบท์ แล้วแต่ว่าจะเป็นคำสั่งให้ทำงานอะไร คอมพิวเตอร์ก็จะเหมือนกับคนที่จะต้องทำงานตามคำสั่ง เมื่อรับคำสั่งแล้วก็จะไปทำตามคำสั่งนั้นเสร็จสิ้นแล้วก็กลับมารับคำสั่งต่อไป

จากรูปที่ 2.33 เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งมีวงจร Power on reset ต่ออยู่จะมีการรีเซทเกิดขึ้นการทำงานภายใน 8051 จะเริ่มจากบล็อก Program Counter ซึ่งเป็นวงจรนับ (Counter Circuit) ชนิดหนึ่งส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนบัส (Bus) หมายเลข 1 บัสนี้มีขนาด 16 บิท ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR Register ที่เป็นวงจร Lacth ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำจะปรากฏที่บนบัส 16 บิทหมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซท ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็น ROM ภายในหรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา EA ซึ่งต่ออยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส (Decoder) แล้วสร้างสัญญาณควบคุมต่อไปถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมาทางบัสหมายเลข 2 ข้อมูลจาก ROM จะถูกส่งลงไปยังบัสหมายเลข 3 ที่เรียกว่า Internal Data Bus แล้วนำไปเก็บไว้ที่ Intruction Rigister (เป็นวงจร Latch) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่น ๆ ต่อไปแล้วแต่ว่าเป็นคำสั่งให้ทำงานอะไร ในกรณีที่เลือก ROM ภายนอก 8051 โดยป้อนสัญญาณลอจิก 1 เข้าไปที่ ขา EA จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ท 0 และพอร์ท 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ท 0 ไปยัง Internal Data Bus แล้วไปเก็บที่ Intruction Register เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจาก Rom ภายในการทำงานในช่วงส่งค่าตำแหน่งหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ใน Intruction Register เรียกว่าเป็นช่วงของการ Fetch (Fetch Cycle) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่นถ้าเป็นคำสั่งให้บวกข้อมูลในรีจิสเตอร์ Accumulator กับข้อมูลจากหน่วยความจำ Data Mormoory ภายใน RAM ตำแหน่ง 23H วงจร Timing and Control ก็จะส่งสัญญาณให้ Intruction Register ส่งค่าตำแหน่งความจำ 23H ลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อใช้ชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่งให้ RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1 (วงจร Lacth) ขณะเดียวกับวงจร Timing and Control ที่จะส่งสัญญาณไปยัง ACC ให้ส่งข้อมูลมายัง TMP2 (วงจร Lacth) วงจร ALU ซึ่งโครงสร้างเป็นวงจรทำการคำนวณทางคณิตศาสตร์ (บวก, ลบ, คูณ, หาร) และยังสามารถทำงานทางลอจิก (AND, OR, NOT, XOR) จะทำการบวกเลขจาก TMP1 และ TMP2 เข้าด้วยกันผลลัพธ์ที่ได้จะส่งผ่าน Internal data Bus กลับไปเก็บยัง ACC PSW (Program Status Word) ซึ่งจะทำหน้าที่เก็บสถานะผลลัพธ์ของการทำงานใน ALU เช่น ผลลัพธ์การบวกมีค่าเกิน 8 บิท ก็จะทำให้บิทหนึ่งใน PSW ถูก SET เป็น 1

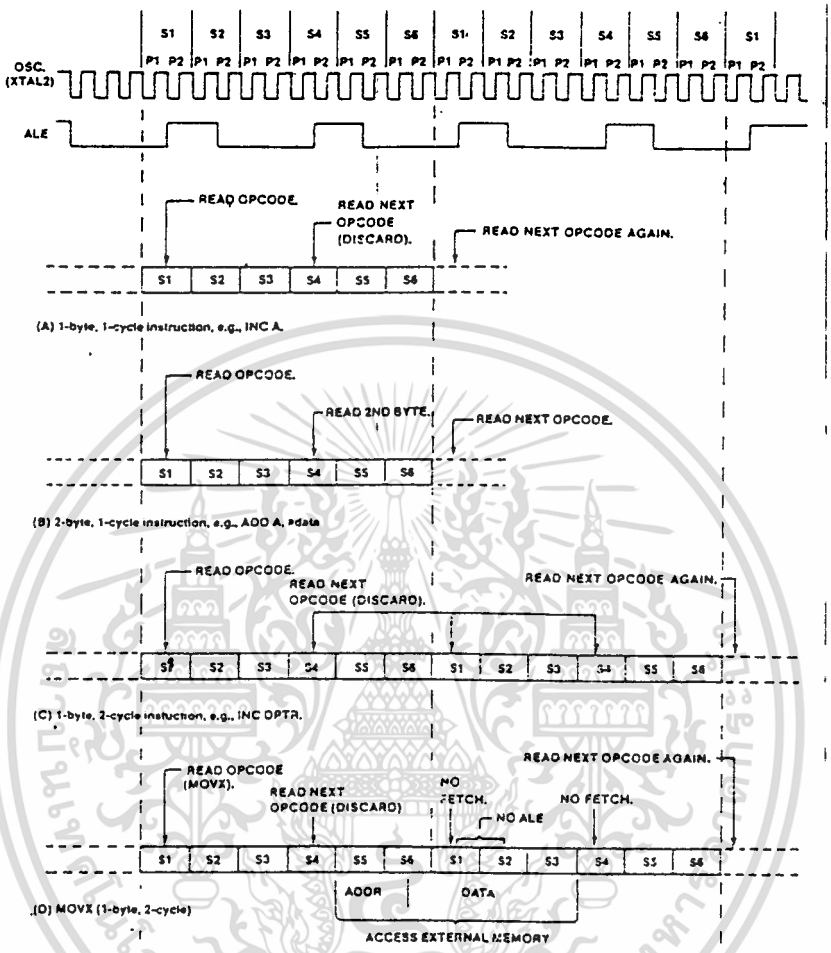
การทำงานที่กล่าวมาข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and Control และสัญญาณที่สร้างขึ้นนี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างมาจาก Oscillator ทำให้การทำงานต่าง ๆ เป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ ดังในรูปที่ 2.42

คำสั่งแต่ละคำสั่งของ 8051 จะใช้เวลาทำงาน 1,2 หรือ 3 ไซเคิลของเครื่อง (Machine Cycle) แล้วแต่ว่าเป็นคำสั่งประเภทใด

คำสั่งประเภท 1 ไบท์ 1 ไซเคิลของเครื่องได้แก่คำสั่ง INC A จะมีการอ่านคำสั่งจากหน่วยความจำสำหรับโปรแกรม 2 ครั้ง ที่เวลาประมาณขอบขาขึ้นของสัญญาณ ALE แรก แล้วนำไปเก็บที่ Intruction Register เพื่อให้วงจร Timing and Control ถอดรหัสแล้วเข้าอยู่การ Execute ขณะเดียวกันก็จะเริ่มต้นการ Fetch คำสั่งที่อยู่ในหน่วยความจำตำแหน่งถัดไปเข้ามา และคำสั่งที่ 2 จะถูกอ่านเข้ามาที่เวลาขอบขาขึ้นของสัญญาณ ALE ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็จะทราบว่าการทำงานคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่ง เพียง 1 ไบท์ ดังนั้นคำสั่งที่ถูกอ่านมาไบท์ที่ 2 จะไม่ถูกนำมาทำงาน เพียงแต่อ่านเข้ามาแล้วทิ้งไป (Discard) ดังในรูปที่ 2.42a

คำสั่งประเภท 2 ไบท์และใช้เวลา 1 ไซเคิลของเครื่องได้แก่ คำสั่ง ADD A,#data ในหนึ่งไซเคิลของเครื่องนี้จะมีการอ่านคำสั่งเข้ามา 2 ไบท์เหมือนกับคำสั่งประเภท 1 ไบท์ 1ไซเคิลของเครื่อง แตกต่างกันที่ไบท์ที่ 2 จะถูกนำมาใช้งานด้วยไม่ถูกทิ้งไปดังรูปที่ 2.42b ตัวอย่างของคำสั่ง ADD A,#33H จะเขียนเป็นภาษาเครื่องได้ 2 ไบท์ คือ 24 33 เมื่ออ่านคำสั่งไบท์แรกคือ 24 เข้าไปไว้ที่ Inturction Register และ Timing and Coutrol จะถอดรหัสพบว่าเป็นคำสั่งบวกเลข ก็จะส่งสัญญาณไปยัง Accumulator ให้เอาข้อมูลไปไว้ที่ IMP1 เมื่อคำสั่งที่ 2 ถูกอ่านเข้ามาที่ Intruction register แล้ว Timing and Control จะสั่งให้เอาข้อมูลไบท์ที่ 2 ส่งลงไปยัง Internal Data Bus ไปเก็บยัง TMP1 จากนั้นวงจร ALU จะนำเอาข้อมูล TMP1 และ TMP2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.42 ลำดับสถานะการทำงานใน MCS-51

มาบวกกับผลลัพธ์ที่ได้จะส่งออกจาก ALU ไปยัง Internal Data Bus แล้วไปเก็บไว้ที่ Accumulator

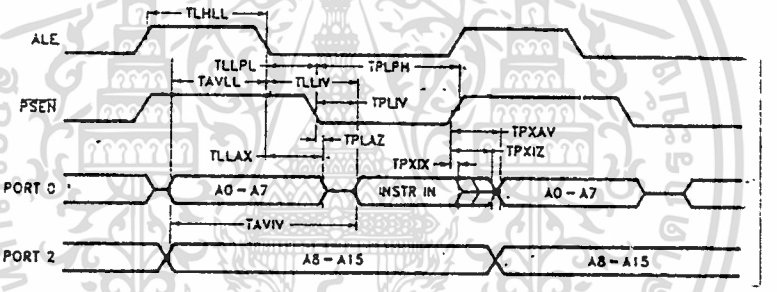
คำสั่งประเภท 1,2 หรือ 3 ไบท์ ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่องเช่นคำสั่ง INC DPTR จะมีการอ่านคำสั่งเข้าไป 4 ครั้งทุก ๆ ขอบขาขึ้นของสัญญาณ ALE ที่มี 2 ครั้งต่อ 1 ไซเคิลของเครื่อง ถ้าเป็นคำสั่งประเภท 1,2 หรือ 3 ไบท์ วงจร Timing and Control จะเอาคำสั่ง 1,2 หรือ 3 ไบท์แรกเท่านั้นไปทำงานส่วนคำสั่งที่เหลือจะทิ้งไปดังรูปที่ 2.42c คำสั่ง 1 ไบท์ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่องที่กล่าวมาแล้วจะไม่รวมถึงคำสั่ง MOVX ซึ่งใช้ในการอ่านหรือเขียนข้อมูลกับหน่วยความจำ Data Memory ภายนอก การทำงานของคำสั่งนี้จะมีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fetch คำสั่งเข้าไป 2 ไบท์ ในไซเคิลของเครื่องแรก ในไซเคิลของเครื่องที่ 2 จะไม่มีการ Fetch คำสั่งเข้าไปแต่จะเป็นช่วงเวลาของการอ่านหรือเขียนข้อมูลกับ Data memory ภายนอกสัญญาณ ALE ซึ่งปกติจะเปลี่ยนเป็น 1 ที่ S1P2 ก็จะไม่เปลี่ยนเป็น 1 ไซเคิลของเครื่องที่ 2 โดยจะเป็น 0 อยู่จนกว่าจะถึงเวลา S4P2 ของไซเคิลของเครื่องที่ 2 สัญญาณ ALE จะเปลี่ยนเป็น 1 เพื่อทำการอ่านหรือเขียนข้อมูลกับ Data Memory ภายนอก

2.6.6 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 นั้น ลำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่งมีดังรูปที่ 2.43

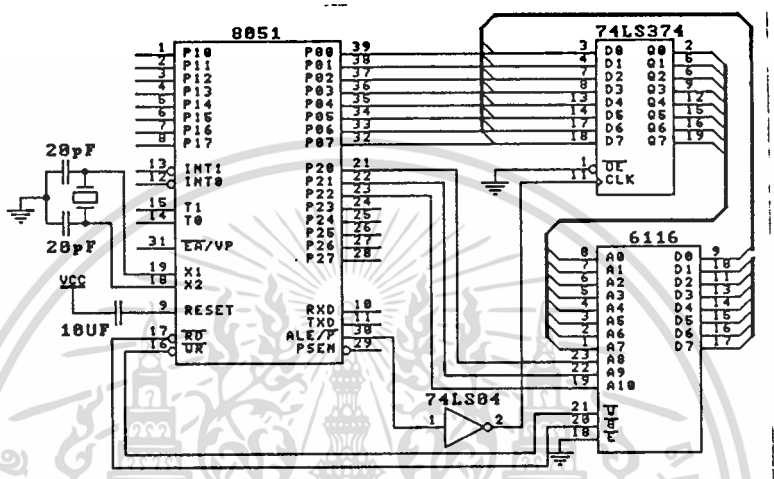


รูปที่ 2.43 Timing diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก

การอ่านคำสั่ง (Fetch) จาก Program area ภายนอกจะเริ่มจาก 8051 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณที่ขา PSEN จะเป็น 1 จากนั้น Port 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างและพอร์ท 2 จะส่งตำแหน่งหน่วยความจำ 8 บิท บนออกมาแล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาลงของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 ไว้ จากนั้นพอร์ท 0 ก็จะยกเลิกการส่งค่าตำแหน่งหน่วยความจำเข้าสู่ภาวะ High Impedance และสัญญาณ PSEN จะเป็น 0 เพื่อเตรียมรับคำสั่งที่ส่งออกจากหน่วยความจำภายนอกเข้าไปยัง 8051 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register (ดูรูป 2.33) แล้วสัญญาณ PSEN จะกลับเป็น 1 พร้อมสัญญาณ ALE ก็จะกลับเป็น High เพื่อการอ่านคำสั่งต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลา ตั้งแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่งจากนั้นจะเริ่มลำดับการ Fetch ไบท์แรกนั้นเองจาก Timing Diagram ดังกล่าวจะออกแบบวงจรที่มี Program Memory อยู่ภายนอก 8051 ได้ดังตัวอย่างในรูปที่ 2.44



รูปที่ 2.44 วงจรที่มี Program memory อยู่ภายนอก 8051

74LS374 ในรูปที่ 2.44 จะทำหน้าที่ Latch ตำแหน่งในหน่วยความจำ 8 บิท บิทล่างที่เวลาขอบขาลงของสัญญาณ ALE ซึ่งสัญญาณ ALE จะถูกกลับให้เป็นตรงข้ามโดย Inverter 74LS04 ก่อนที่จะป้อนให้กับขา CK ของ 74374 และที่ขอบขาขึ้นของสัญญาณที่ออกจาก 74LS04 จะ Latch ตำแหน่งหน่วยความจำ ข้อมูลที่ออกจาก 74LS374 จะเป็นค่า 8 บิทล่างของตำแหน่งหน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อค่าตำแหน่งหน่วยความจำ 8 บิทเข้ากับ A0 ถึง A7 ของ EPROM และข้อมูลจากพอร์ท 2 บิท P2.0 ถึง P2.3 จะต่อเข้ากับ A8-A11 ของ EPROM โดยตรงเพราะค่าตำแหน่งหน่วยความจำ 8 บิทบนที่ออกมาจากพอร์ท 2 จะคงที่ตลอดเวลา ขา PSEN ของ 8051 จะถูกต่อเข้ากับขา 0E ของ EPROM 2716 ดังนั้นเมื่อสัญญาณ PSEN มีสภาวะลอจิกเป็น 0 ก็จะสั่งคำสั่งที่เก็บใน EPROM ณ ตำแหน่งที่ชี้โดยข้อมูลที่ขา A0 ถึง A11 ออกมายังพอร์ท 0 และถูก 8051 เก็บไปทำงานต่อไป

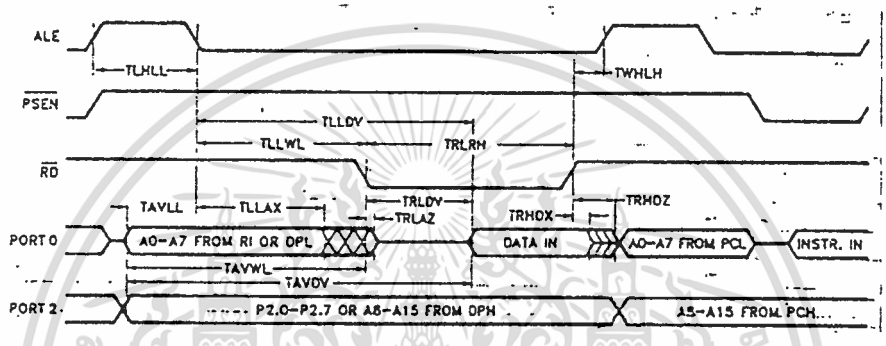
การอ่าน-เขียนข้อมูลกับหน่วยความจำสำหรับข้อมูลภายนอก 8051

การอ่าน-เขียนข้อมูลกับ Data Memory ภายใน 8051 นั้นจะมีสัญญาณสร้างมาจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นที่จะต้องทำความเข้าใจ แต่การอ่าน-เขียนข้อมูลกับ Data Memory อันเนื่องมาจากคำสั่ง MOVX นั้น เมื่อคำสั่งดังกล่าวถูกอ่านเข้ามายัง Intruction Register แล้วTiming and Control จะทำการถอดรหัสแล้วสร้างสัญญาณควบคุมดังนี้

การอ่านข้อมูลจาก External Data Memory จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 2.45



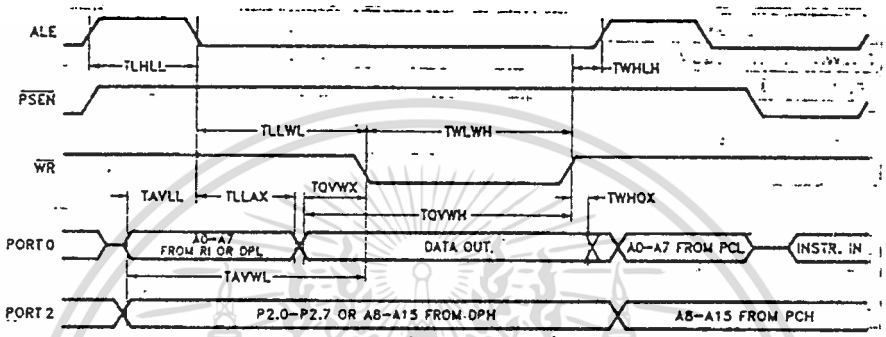
รูปที่ 2.45 Timing diagram ของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 8051

การทำงานเริ่มจากการส่งค่าตำแหน่งหน่วยความจำภายนอก 8 บิทล่างออกทางพอร์ท 0 และ 8 บิทบนออกทางพอร์ท 2 เมื่อส่งค่าตำแหน่งแล้วสัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับมาเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามารถ Latch ตำแหน่งหน่วยความจำไว้เหมือนกับการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลบนพอร์ท 0 จะเปลี่ยนแปลงไปก็จะมีค่าตำแหน่งหน่วยความจำส่งไปยังหน่วยความจำในระหว่างการติดต่อกับ Data Memory นี้สัญญาณ PSEN จะเป็น 1 ตลอดเพราะสัญญาณ PSEN จะ Active (เป็น 0) ก็ต่อเมื่อการติดต่อกับหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เท่านั้น 8051 จะส่งสัญญาณลอจิก 0 ออกมาทางขา RD (P3.7) เพื่อบอกกับหน่วยความจำภายนอกว่าต้องการอ่านข้อมูลเข้าไปเมื่อ 8051 ส่งสัญญาณ RD เป็นลอจิก 0 จะทำให้พอร์ท 0 เข้าสู่สถานะ High Impedance พร้อมที่จะให้น่วยความจำภายนอกส่งข้อมูลมาบนพอร์ท 0 ข้อมูลบนพอร์ท 0 ซึ่งส่งมาจากหน่วยความจำภายนอกจะถูกอ่านเข้าไปเก็บที่เวลาขอบขาขึ้นของสัญญาณ RD จากนั้นสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไปใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับภายนอกนี้ พอร์ท 2 จะส่งค่าตำแหน่งความจำ 8 บิทบนออกมาตลอดเวลา

การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051 จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 2.46



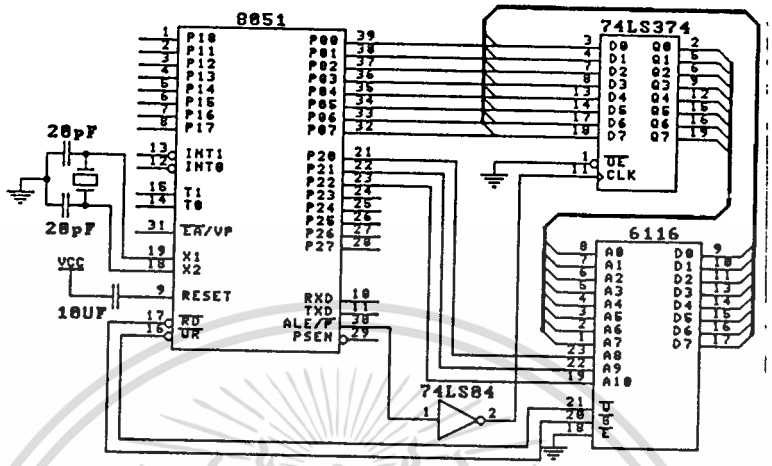
รูปที่ 2.46 Timing diagram ของการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

เมื่อ 8051 ส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างไปยังพอร์ท 0 และ 8 บิทบนลงไปทางพอร์ท 2 แล้ว สัญญาณ ALE จะกลับเป็นอุปกรณ์ภายนอกจะสามารถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำบนพอร์ท 0 เหมือนกับการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก เมื่อสัญญาณ ALE เป็น 0 แล้ว 8051 จะส่งข้อมูลที่ต้องการเขียนไปยังพอร์ท 0 แล้วจะให้สัญญาณ WR เปลี่ยนสภาวะลอจิกเป็น 0 ขณะนี้หน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้นสัญญาณ WR จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลแล้วสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อ Fetch คำสั่งต่อไปมาทำงาน

หน่วยความจำสำหรับข้อมูลภายนอกที่สามารถอ่านและเขียนข้อมูลได้ จะสามารถเขียนเป็นวงจรได้ดังรูปที่ 2.47

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

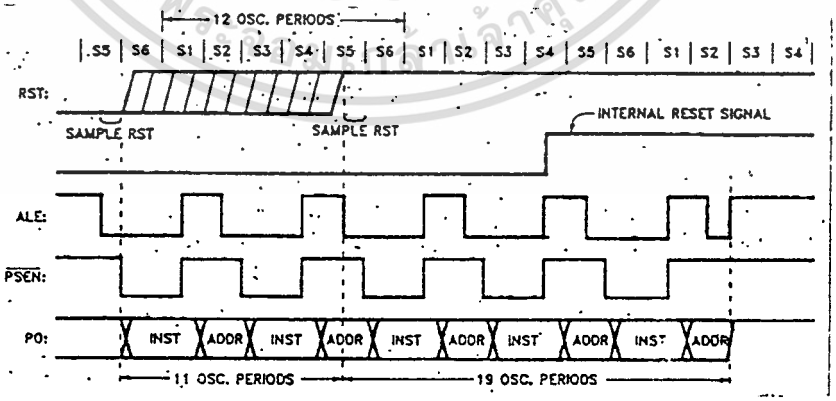
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.47 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่ภายนอก 8051

2.6.7 การรีเซท

เมื่อป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปทาง RST จะไม่ได้เกิดการรีเซทขึ้นทันที ทันใด แต่ลำดับการเกิดรีเซทจะแสดงได้ดังไดอะแกรมตามเวลาในรูปที่ 2.48



รูปที่ 2.48 ไดอะแกรมตามเวลาของรีเซท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.48 เป็น Timing Diagram ของการรีเซท สภาวะลอจิกของสัญญาณที่ขา RST จะถูกอ่านเข้ามาที่เวลา S5P2 (เฟส 2 State 5) ของทุก ๆ ไซเคิลของเครื่อง ในกรณีที่เป็นคำสั่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิลของเครื่องก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาในไซเคิลที่ 2 ของการทำงาน ดังนั้นในการรีเซทจะต้องป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปในขาที่เป็นเวลาอย่างน้อย 2 ไซเคิลของเครื่อง 24 ไซเคิลของสัญญาณนาฬิกาที่สร้างจากออสซิลเลเตอร์ภายใน 8051 เพื่อให้แต่ใจว่าสัญญาณรีเซทจะถูกอ่านเข้าไปตรวจสอบและทำงานขณะที่ทำการรีเซท 8051 ออสซิลเลเตอร์จะต้องทำงานอยู่ด้วย เมื่อ 8051 สุ่มข้อมูลที่ขา RST แล้ว ตรวจพบว่าเป็นสภาวะลอจิก 1 ก็จะสร้างสัญญาณรีเซทขึ้นภายใน ที่เวลา S2P4 ของไซเคิลเครื่องถัดไป ข้อมูลที่แต่ละพอร์ทส่งออกมายังคงปรากฏที่พอร์ทจนกว่าจะเกิดการรีเซทขึ้นซึ่งต้องใช้เวลา 19 ไซเคิล ของสัญญาณจากออสซิลเลเตอร์นับตั้งแต่เวลา S5P2 ในไซเคิลของเครื่องที่พบสัญญาณรีเซท ในระหว่างเวลา 19 ไซเคิลนี้จะยังคงมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

สภาวะของสัญญาณลอจิกที่ขา RST จะถูกอ่านเข้าไปตรวจสอบที่เวลา S5P2 ของทุก ๆ ไซเคิลของเครื่องดังนั้นถึงแม้ว่าสัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาก่อนก็จะยังไม่เกิดการตรวจสอบสัญญาณรีเซท ดังในรูปที่ 2.48 สัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาตั้งแต่ State ที่ 6 ก็จะไม่เกิดอะไรขึ้นจนกระทั่ง 1 ไซเคิลของออสซิลเลเตอร์ต่อมาซึ่งเป็นเวลา S5P2 จึงจะเกิดการตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการทำงานมากกว่า 1 ไซเคิลของเครื่อง 8051 ก็จะต้องทำงานในคำสั่งนั้นให้เสร็จสิ้นเสียก่อนจึงจะเริ่มการรีเซทได้ โดย 8051 จะดูสภาวะของสัญญาณที่ขา RST ของ S5P2 ในไซเคิลของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไซเคิลเครื่องแรก ๆ ในคำสั่งอาจมีสภาวะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไซเคิลของเครื่องสุดท้าย มีสภาวะลอจิกที่ขา RST เป็น 0 ก็จะไม่เกิดการรีเซท

ขึ้นที่เวลา S5P2 เมื่อตรวจสอบสภาวะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอไปจนถึงเวลา S4P2 ของไซเคิลของเครื่องถัดไปจึงจะทำให้สัญญาณรีเซทภายในเปลี่ยนสภาวะลอจิกจาก 0 เป็น 1 ในระหว่างเวลา S5P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของไซเคิลของเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซทภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซท โดยการเขียนข้อมูล 0 ไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังตารางในรูปที่ 2.39 ระหว่างนี้ข้อมูลใน RAM ภายใน 8051 จะไม่เปลี่ยนแปลงข้อมูลในระหว่างการเขียนข้อมูลลงไปยัง SFR จะยังมีการ Fetch คำสั่งเข้ามาทำงานอีก 1 คำสั่งจนกว่าจะถึง S3P1 ของไซเคิลของที่ 2 (นับแต่ไซเคิลของเครื่องตรวจพบลอจิก 1 ที่ขา RST)ก็จะทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สภาวะลอจิกที่ขา ALE และ PSEN ค้างอยู่ที่สภาวะลอจิก 1 และจะเป็นอย่างนี้จนกว่าสภาวะลอจิกที่ขา RST เป็น 0 เวลานับตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ PSEN ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคิลของออสซิลเลเตอร์เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 8051 จะรออีก 1 ถึง 2 ไซเคิลของเครื่องสัญญาณ ALE และ PSEN จะเริ่มเปลี่ยนแปลงเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานเริ่มจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

2.6.8 Special Function Register

ใน 8051 จะใช้วิธีการกำหนดชื่อให้กับตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน (Internal Data Memory) ที่เรียกว่า การ Symbolize เช่นการให้ชื่อหน่วยความจำแต่ละหน่วยในแต่ละ Bank ซึ่งอยู่ในช่วงหน่วยความจำตำแหน่ง 00H ถึง 1FH แล้วในคำสั่งจะอ้างอิงหน่วยความจำแต่ละหน่วย โดยการใช้ชื่อ R0, R1, R2, R3, R4, R5, R6 และ R7 หน่วยความจำตำแหน่งเหล่านี้ จะเรียกอีกอย่างหนึ่งว่าเป็นรีจิสเตอร์ ซึ่งมีหน้าที่ในการเก็บหรือพักข้อมูล หรือใช้สำหรับการกระทำบางอย่าง รีจิสเตอร์กลุ่มหนึ่งใน 8051 ที่เรียกว่า Special Function Register (SFR) เป็นรีจิสเตอร์ที่ใช้สำหรับงานเฉพาะ คือข้อมูลที่ถูกนำไปเก็บไว้ในรีจิสเตอร์เหล่านี้จะมีความหมายเฉพาะตัวของรีจิสเตอร์ ที่แต่ละตำแหน่งของ SFR อาจจะไม่ใช้เป็นหน่วยความจำ (RAM) แต่อาจเป็นตัวนับหน่วยความจำตำแหน่งหนึ่ง จึงเรียกการมองข้อมูลแต่ละตำแหน่งนี้ว่า Memory Map I/O รีจิสเตอร์กลุ่มนี้มีดังในรูปที่ 2.49

ในรูปที่ 2.49 ช่อง Symbol ทางด้านซ้ายจะเป็นลักษณะของรีจิสเตอร์ในช่องถัดมาคือชื่อของรีจิสเตอร์ตามสัญลักษณ์ที่อยู่ทางซ้าย ในช่องขวาสุดจะเป็นตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่แทนด้วยชื่อ หรือสัญลักษณ์ทางซ้ายนั่นเอง เช่น ในบรรทัดแรกคือรีจิสเตอร์ชื่อ Accumulator ที่มีสัญลักษณ์ ACC รีจิสเตอร์นี้คือหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่ตำแหน่ง 0E0H การอ่านหรือเขียนข้อมูลกับรีจิสเตอร์เหล่านี้สามารถทำได้โดยการใช้คำสั่งในกลุ่มการเคลื่อนย้ายข้อมูล (เช่น MOV A, #25H หรือ MOV 0E0H, #25H) และรีจิสเตอร์บางตัวในกลุ่มนี้สามารถใช้คำสั่งในกลุ่ม Boolean Instruction เพื่อการทำงานกับแต่ละบิทในรีจิสเตอร์เหล่านี้ได้ จากตารางในรูปที่ 2.49 รีจิสเตอร์ที่มีเครื่องหมาย * อยู่ข้างหน้าจะสามารถใช้คำสั่งในกลุ่ม Boolean Instruction จัดการกับแต่ละบิทได้ รีจิสเตอร์ที่มีเครื่องหมาย + นำหน้าหมายความว่า รีจิสเตอร์นั้นมีเฉพาะใน 80C52 และ 83C154 เท่านั้น ไม่มีใน 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

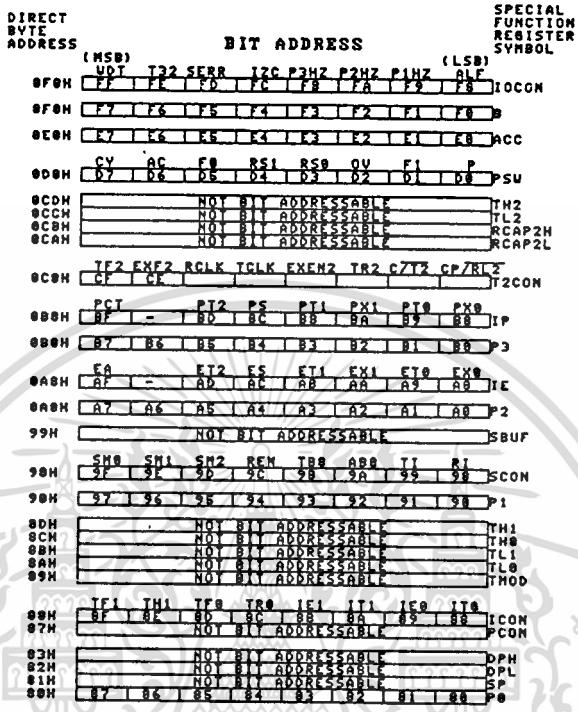
~ Symbol	Name	Adderss
*ACC	Accumulator	CE0H
*B	B Register	CFOH
*PSW	Program Status Word	CDOH
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	CA0H
*P3	Port 3	CB0H
*IP	Interrupt Prority Control	CB8H
*IE	Interrupt Enable Control	CA8H
TMOD	Timer/Counter Mode Control	89H
*TCCN	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H
*IOCON (1)	IO Control	F8H

+80C52 and 83c154 only *bit addressable

(1) 83c154 only

รูปที่ 2.49 Special Function Register (SFR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.50 แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิท

รูปที่ 2.50 ในช่องสี่เหลี่ยมเล็ก ๆ จะเป็นตำแหน่งของบิทนั้นในแต่ละรีจิสเตอร์ เช่น ในช่องซ้ายสุดของรีจิสเตอร์ TCOM มีค่า 8FH ซึ่งเป็นค่าตำแหน่งบิท 7 ของหน่วยความจำตำแหน่ง 88H ถ้าต้องการทำให้บิทนี้มีค่าเป็น 0 ก็สามารถทำได้โดยใช้คำสั่ง

CLR 8FH

หรือจะทำให้บิทนี้เป็น 1 ก็ทำได้โดยคำสั่ง

SETB 8FH

รีจิสเตอร์ในกลุ่ม Special Function Register มีดังนี้

1. Accumulator ตำแหน่งหน่วยความจำภายในเท่ากับ 0E0H

รีจิสเตอร์นี้มีขนาด 8 บิท เป็นรีจิสเตอร์ที่ใช้มาก ซึ่งในรหัสคำสั่งช่วยจำจะอ้างอิงถึงรีจิสเตอร์นี้ โดยใช้สัญลักษณ์ A เช่น MOV A, #15H คำสั่งที่จะอ่านหรือเก็บข้อมูลกับหน่วยความจำภายนอกจะต้องกระทำผ่านรีจิสเตอร์นี้เท่านั้น เช่น MOVX @ RO, A หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOVX A, @RO เป็นต้น และข้อมูลที่อยู่ภายในรีจิสเตอร์นี้ก็สามารถที่จะให้โปรแกรมตรวจสอบเพื่อกระโดดการทำงานไปยังตำแหน่งอื่นได้ เช่น JZ rel

2. B Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0F0H

เป็นรีจิสเตอร์ขนาด B บิทที่ใช้ในคำสั่งการคูณ (MUL AB) และคำสั่งการหาร (DIV AB) เท่านั้น โดยรีจิสเตอร์ B นี้จะเป็นตัวเก็บคูณ และผลลัพธ์บิท 8 ถึง 15 ในคำสั่งการคูณ ส่วนในคำสั่งการหารนั้นรีจิสเตอร์ B จะเก็บตัวหารและผลการหาร การเขียนข้อมูลไปยังรีจิสเตอร์นี้จะต้องใช้คำสั่งเคลื่อนย้ายข้อมูลไปยังตำแหน่ง 0F0H เช่น MOV 0F0H, 25H จะเป็นการกำหนดว่า 25H ให้กับรีจิสเตอร์ B

3. Program status word ตำแหน่งหน่วยความจำภายในเท่ากับ 0D0H

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

4. Stack Pointer ตำแหน่งหน่วยความจำภายในเท่ากับ 081H

เป็นรีจิสเตอร์ 8 บิท รีจิสเตอร์นี้จะใช้ชี้ตำแหน่งหน่วยความจำภายใน 8051 ที่ใช้เก็บตำแหน่ง (Adress) เดิมของโปรแกรมก่อนทำงานคำสั่ง CALL หรือตำแหน่งที่จะใช้เก็บข้อมูลด้วยคำสั่ง PUSH และตำแหน่งที่จะอ่านข้อมูลออกมาในคำสั่ง POP เมื่อทำการรีเซท 8051 โดยการป้อนสัญญาณสภาวะลอจิก 1 เข้าไปที่ขา RST ของ 8051 จะทำให้ข้อมูลในรีจิสเตอร์นี้มีค่าเป็น 07H หมายความว่ารีจิสเตอร์ SP ชี้หน่วยความจำภายใน 8051 ที่ตำแหน่ง 07H ค่าของ SP จะเปลี่ยนแปลงไปโดยการใช้คำสั่งเคลื่อนย้ายข้อมูลหรือการทำงานของคำสั่ง PUSH, POP และ CALL

5. Data Pointer Register ตำแหน่งหน่วยความจำภายในเท่ากับ 82H และ 83H

รีจิสเตอร์ DPTR มีขนาด 16 บิท หน้าที่ของรีจิสเตอร์นี้ก็คือใช้สำหรับชี้ตำแหน่งความจำรีจิสเตอร์ DPTR นี้สามารถใช้อ้างอิงตำแหน่งหน่วยความจำได้สูงสุด 60 x 1024 ตำแหน่ง เช่นคำสั่ง MOVX A, @DPTR หรือใช้ชี้ตำแหน่งโปรแกรมที่ต้องการกระโดดข้ามไปทำงานเช่นคำสั่ง JMP @A+DPTR รีจิสเตอร์ DPTR นี้ประกอบด้วยรีจิสเตอร์ขนาด 8 บิท 2 ตัว คือ DPH ซึ่งอยู่ในตำแหน่ง 83H และ DPL ซึ่งอยู่ในตำแหน่ง 82H ในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับข้อมูลภายใน 8051 ดังนั้นการแก้ไขข้อมูลในรีจิสเตอร์ DPTR จึงทำได้ทั้งทีละ 16 บิท เช่นคำสั่ง MOV OPTR, #data 16 หรือจัดการทีละ 8 บิท โดยการแก้ไขข้อมูลใน DPH หรือ DPL ด้วยคำสั่ง MOV 83H, #data8 หรือ MOV 82H, #data8

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

Special Function Register ชื่อ P0, P1, P2 หรือ P3 เป็นรีจิสเตอร์ขนาด 8 บิท ของหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่ตำแหน่ง 80H, 90H, 0A0H และ 0B0H ตามลำดับ การเขียนข้อมูลลงไปยังหน่วยความจำแต่ละตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ทนั้น ๆ ของ 8051 ข้อมูลที่เขียนออกไปจะถูก LATCH ค้างไว้และปรากฏที่แต่ละบิทของพอร์ท เช่น MOV 80H, #18H จะปรากฏสภาวะลอจิก LLLHHLLL ที่ขาบิท 7 ถึง 0 ของพอร์ท 0 ตามลำดับในการอ่านข้อมูลจากรีจิสเตอร์แต่ละตัวก็จะเป็นการอ่านสภาวะลอจิกของสัญญาณที่ปรากฏอยู่ที่แต่ละขาของพอร์ทนั้น ๆ เช่น MOV A, B0H เป็นการอ่านสภาวะลอจิกจากพอร์ท 0 เข้ามายัง Accemulator การอ่านข้อมูลจากพอร์ทจะต้องเขียนข้อมูล 11111111B ไปไว้ที่พอร์ทนั้น ๆ เสียก่อน ของพอร์ท 0 ถึง 3 จะสามารถแก้ไขเปลี่ยนแปลงได้โดยคำสั่ง SET B bit และ CLR bit

7. Serial Data Buffer ตำแหน่งหน่วยความจำภายในเท่ากับ 99H

รีจิสเตอร์นี้มีขนาด 8 บิท และมีตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 8051 เท่ากับ 99H โครงสร้างภายในแล้วรีจิสเตอร์นี้มี 2 ตัวที่มีชื่อเดียวกัน ตัวหนึ่งสำหรับเก็บข้อมูลที่จะส่งแบบอนุกรมออกจาก 8051 จึงเรียกว่ามีการทำงานแบบ Full Duplex เพราะสามารถส่งและรับข้อมูลได้ในเวลาเดียวกัน เนื่องจากมีรีจิสเตอร์สำหรับส่งและรับแยกออกจากกัน ข้อมูลที่ต้องการจะส่งออกก็ให้เขียนไปยังรีจิสเตอร์ SBUF แล้วสั่งงานให้ส่งข้อมูลออกมา ข้อมูลในรีจิสเตอร์จะเริ่มส่งออก โดยเริ่มจากบิท 0 ถึง 7 ตามลำดับ ถ้าข้อมูลมีข้อมูลเข้ามาทางขา RXD ก็จะถูกเก็บไปไว้ในรีจิสเตอร์นี้โดยถือว่าข้อมูลบิทแรกที่เข้ามา คือ บิท 0

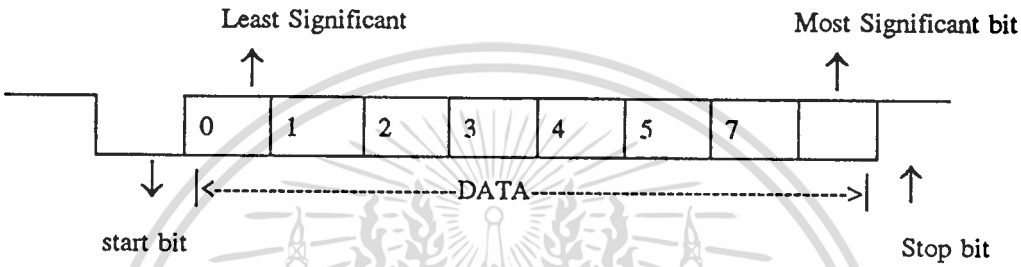
Serial Port จะสามารถกำหนดให้การทำงานรับ-ส่งข้อมูลแบบอนุกรมได้ 4 โหมด (MODE) โดยการกำหนดให้รีจิสเตอร์ SCON (Serial Control Register) ซึ่งจะอธิบายต่อไปในข้อ 8 แต่ละโหมดการทำงานของ Serial Port มีดังนี้

MODE 0 : ในโหมดนี้จะมีการรับหรือส่งข้อมูลแบบอนุกรมทางขา RXD และขา TXD จะส่งสัญญาณ Clock ที่ใช้สำหรับเลื่อน (Shift) ข้อมูล 1 ชุดของข้อมูลจะประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยข้อมูล 8 บิทเท่านั้นและจะเริ่มการรับ-ส่งข้อมูลจากบิท 0 จนถึงบิท 7 ตามลำดับ อัตราการส่งข้อมูลแบบอนุกรมจะเท่ากับ 1/12 เท่าของความถี่สัญญาณนาฬิกาที่ใช้กับ 8051

MODE 1 : ข้อมูลที่รับ-ส่ง 1 ชุดในโหมดนี้จะมี 10 บิท ผ่านทางขา RXD และ TXD ตามลำดับ เริ่มต้นการรับส่งข้อมูลด้วย Start bit 1 บิท (ลอจิกเป็น 0), ข้อมูล 8 บิท (เริ่มจากบิท 0), Stop bit 1 บิท (ลอจิก 0) การส่งข้อมูลโหมดนี้มีดังรูปที่ 2.51



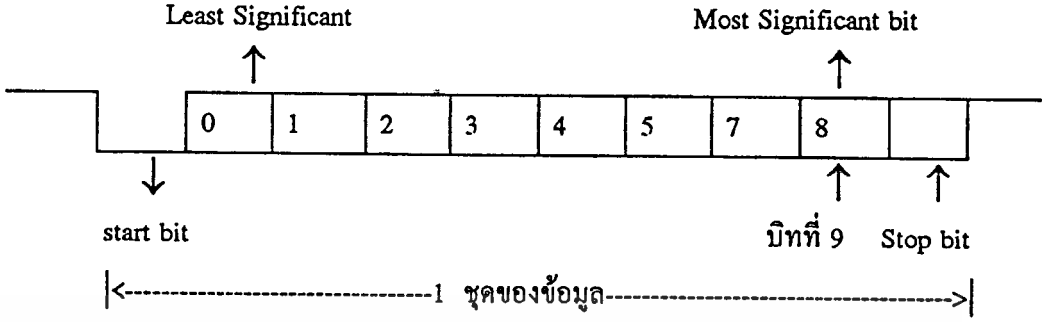
รูปที่ 2.51 ชุดข้อมูลอนุกรมในโหมด 1

เมื่อรับข้อมูลอนุกรมเข้ามาข้อมูล 8 บิท จะถูกเก็บในรีจิสเตอร์ SBUF และ Stop Bit จะถูกเก็บไปที่บิท RB8 ในรีจิสเตอร์ SCON ในการส่งข้อมูลออกก็จะเขียนข้อมูลที่ต้องการส่งไปยังรีจิสเตอร์ SBUF อัตราการส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ โดยจะขึ้นกับการเกิด Overflow ใน Timer 1

MODE 2 : การรับ-ส่งข้อมูลของโหมด 2 1 ชุดจะมี 11 บิท จะส่งออกผ่านทางขา TXD และรับเข้าทางขา RXD ข้อมูลแต่ละชุดจะเริ่มต้นด้วย Start bit 1 บิท, ข้อมูล 8 บิท (เริ่มจากบิท 0), ข้อมูลบิทที่ 9 จำนวน 1 บิท และ Stop Bit อีก 1 บิท ข้อมูลบิทที่ 9 ที่จะส่งออกนี้สามารถกำหนดได้ว่าจะให้เป็น 1 หรือ 0 โดยการกำหนดในบิท RB8 ของรีจิสเตอร์ SCON บิทนี้มีประโยชน์มากในการส่งข้อมูลแบบอนุกรมเช่นอาจส่งค่าพาริตี้ของข้อมูลไปเป็นบิทที่ 9 เพื่อว่าเมื่อปลายทางรับข้อมูลแล้วจะได้ใช้ตรวจสอบว่าข้อมูลที่รับเข้ามา 8 บิทมีพาริตี้บิท ตรงกับบิทที่ 9 หรือไม่ ถ้าไม่ตรงก็แสดงว่ามีข้อผิดพลาดเกิดขึ้นระหว่างการส่งข้อมูล เมื่อรับข้อมูลเข้ามานั้นข้อมูลบิทที่ 9 ก็จะถูกนำไปเก็บในบิท RB8 ของรีจิสเตอร์ SCON ชุดข้อมูลที่รับ-ส่งมีดังรูปที่ 2.52

อัตราการส่งข้อมูลจะกำหนดให้เป็น 1/32 หรือ 1/64 เท่าของความถี่สัญญาณนาฬิกาที่ใช้กับ 8051 โดยการกำหนดบิท SM0D ในรีจิสเตอร์ PCON MODE 3 : การส่งข้อมูลโหมดนี้ 1 ชุดมี 11 บิท เหมือนกับโหมด 2 ทุกประการแตกต่างกันตรงอัตราการส่งข้อมูลเท่านั้น คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.52 ชุดข้อมูลอนุกรมในโหมด 2

อัตราการส่งข้อมูลในโหมด 3 นี้ สามารถกำหนดได้ตามต้องการ โดยจะขึ้นกับการเกิด Overflow ใน Timer 1 เหมือนกับโหมด 1

8. SCON (Serial Port Control Register).

ตำแหน่งหน่วยความจำภายในเท่ากับ 98H

รีจิสเตอร์ SCON มีขนาด 8 บิท ใช้สำหรับควบคุมการส่งและรับข้อมูลผ่านทาง Serial Port แต่ละบิทของข้อมูลในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 2.53

SCON : SERIAL PORT CONTROL REGISTER BIT ADDERSSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier. (NOTE 1).
SM1	SCON.6	Serial Port mode specifier. (NOTE 1).
SM2	SCON.5	Enables the Muitiprocessor communication feature in mode 2&3. in mode 2 or 3. if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. in mode 1. if SM2=1 then RI will not be activated if a valid stop bit was not received in mode 0. SM2 snould be 0. (See Table 9).
REN	SCON.4	Set/cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2&3. Set/Cleared by software.
RB8	SCON.2	In moeds 2 & 3. is the 9th data bit that was recerved. In mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.if SM2=0.RB8 is the stop bit that was ercerved. in mode 0. RB8 is not used.

TI SCON.1 Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0. or at the beginning of the stop bit in the other modes. Must be cleared by software.

RI SCON.0 Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0. or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1 :

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

SERIAL PORT SET-UP :

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

รูปที่ 2.53 Serial Port Control Register (SCON) (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.53 บิท RI จะเป็นชื่อของบิท 0 และ SMO จะเป็นบิท 7 ของรีจิสเตอร์ SCON ซึ่งความหมายหรือการทำงานของแต่ละบิทมีดังนี้

R1 Receive Interrupt Flag

บิทนี้จะถูกกำหนดโดยฮาร์ดแวร์ให้มีค่าเป็น 0 หรือ 1 โดยที่ในการรับข้อมูลโหมด 0 นั้นบิท RB8 จะมีค่าเป็น 1 เมื่อมีข้อมูลเข้ามาครบทั้ง 8 บิท ส่วนในโหมดอื่นบิท RB8 จะเป็น 1 ก็ต่อเมื่อข้อมูลเข้ามาถึงเวลาครึ่งหนึ่งของ Stop Bit (ยกเว้นบางกรณีให้ดูที่เรื่องบิท SM2 ของรีจิสเตอร์ SCON) บิทนี้จะสามารถ Clear ให้มีค่าเป็น 0 ได้ โดยใช้คำสั่ง CLR bit โดยค่าตำแหน่งของบิทมีค่าเท่ากับ 98H บิทนี้มีประโยชน์ให้รู้ว่าข้อมูลได้เข้ามาอยู่ใน SBUF ครบทั้งชุดแล้วพร้อมที่ CPU จะอ่านไปเก็บในหน่วยความจำต่อไป หรืออาจกำหนดค่าในรีจิสเตอร์ IE และ IP เพื่อเมื่อมีข้อมูลมาทางพอร์ทอนุกรมแล้วจะทำให้เกิดการขัดจังหวะ (Interrupt) การทำงานของโปรแกรมหลัก (Main Program) แล้วกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ (Interrupt Service Routine)

TI Transmit Interrupt Flag

ค่าในบิท TI จะถูกกำหนดให้เป็น 1 หรือ 0 ด้วยฮาร์ดแวร์ โดยในการส่งข้อมูลแบบอนุกรมโหมด 0 บิทนี้ จะเป็น 1 เพื่อจะบอกว่าการส่งข้อมูลในรีจิสเตอร์ SBUF ออกไปทางพอร์ทอนุกรมครบทั้ง 8 บิท แต่ถ้าเป็นการส่งข้อมูลแบบอนุกรมในโหมดอื่นจะทำให้ข้อมูลในบิท TI เป็น 1 เมื่อเริ่มการส่ง Stop bit ข้อมูลบิทนี้จะสามารถ Clear เป็น 0 ได้ด้วยคำสั่ง CLR bit โดยที่ค่าตำแหน่งของบิทนี้เท่ากับ 99H บิทนี้ยังมีประโยชน์เพื่อบอกว่าการส่งข้อมูลจาก SBUF ออกไปทางพอร์ทอนุกรมนั้นสิ้นสุดแล้วพร้อมที่จะให้โปรแกรมเขียนข้อมูลลงไปยัง SBUF สำหรับการส่งออกไปได้ นอกจากนี้การกำหนดค่าในรีจิสเตอร์ IE และ IP ยังสามารถที่จะกำหนดให้เกิดการจังหวะการทำงานของโปรแกรมได้เมื่อบิทนี้ถูกฮาร์ดแวร์ทำให้มีค่าเป็น 1

RB8

เมื่อมีการกำหนดให้รับข้อมูลในโหมด 2 และ 3 จะใช้บิทนี้สำหรับเก็บข้อมูลบิทที่ 9 ที่เข้ามาทางพอร์ทอนุกรม ส่วนในโหมด 1 นั้นบิทนี้จะเก็บ Stop bit ซึ่งมีค่าเป็น 1 นั่นเอง ในโหมด 0 บิทนี้จะไม่ถูกใช้งาน ค่าตำแหน่งของบิทนี้คือ 9AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TB8

ในการส่งข้อมูลแบบอนุกรมโหมด 2 และ 3 จะใช้บิทนี้เก็บข้อมูลบิทที่ 9 ส่วนโหมดอื่นจะไม่ใช้งานบิทนี้ การกำหนดค่าในบิทนี้สามารถทำได้โดยใช้คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่งของบิทนี้คือ 9BH

REN Receive Enable

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

SM2

เป็นบิทสำหรับควบคุมการทำงานของฮาร์ดแวร์ที่จะทำให้บิท RI เป็น 1 หรือไม่ ในกรณีที่บิท SM2 เป็น 0 ค่าในบิท RI ก็จะเป็นไปตามที่ได้อธิบายมาแล้วในเรื่องบิท RI แต่ถ้าบิท SM2 = 1 โหมด 2 และ 3 ซึ่งปกติแล้วบิท RI จะเป็น 1 เมื่อข้อมูลบิทที่ 9 เข้ามาเป็น 0 จะไม่ทำให้บิท RI มีค่าเป็น 1 (คือบิท RI จะเป็น 0)

ในโหมด 1 บิท RI มีค่าเป็น 1 เมื่อข้อมูล Stop Bit เข้ามายังพอร์ทอนุกรมถูกต้องแต่ถ้า Stop bit ไม่เข้ามายังพอร์ทอนุกรมอันอาจเกิดจากปัญหาในการส่งข้อมูลแล้วบิท RI จะมีค่าเป็น 0

ในโหมด 0 บิทนี้จะมีค่าเป็น 0 เสมอ

SM0, SM1

เป็น 2 บิทที่ใช้งานร่วมกันเพื่อกำหนดโหมดของการรับ-ส่งข้อมูลของพอร์ทอนุกรม ค่าใน 2 บิทนี้จะกำหนดโหมดได้ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. TIMER Register THO, TLO, THI, TLI

ตำแหน่งหน่วยความจำภายในเท่ากับ 8CH, 8AH, 8DH, 8BH

ใน 8051 จะมีวงจร Timer อยู่ 2 ชุด คือ Timer 1 (8052 จะมี Timer 2 อีก 1 ชุด) ใน Timer แต่ละชุดจะมีรีจิสเตอร์ขนาด 8 บิท อยู่ 2 ตัว เพื่อเก็บค่าการนับของ Timer ได้สูงสุดถึง 16 บิท ใน Timer 0 รีจิสเตอร์ TH1, TL1 TLx (x หมายถึง 0 หรือ 1) จะเก็บค่าของการนับ 8 บิทล่างและ THx จะเก็บค่าของการนับ 8 บิทบน ผู้ใช้จะสามารถทราบการทำงานของวงจร Timer ในโหมด Timer หรือโหมด Counter ได้โดยการกำหนดในรีจิสเตอร์ชื่อ TMOD (Timer/Counter Mode Control Register) การทำงานเป็น Timer นั้นจะให้รีจิสเตอร์ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิล (Cycle) ของสัญญาณนาฬิกาที่ผ่านวงจรหาร 12 ดังรูปที่ 2.55 เมื่อการนับครบถึงค่าสูงสุดที่รีจิสเตอร์ TLx และ THx จะเก็บได้คือค่า FFFFH แล้วยังนับต่อไป ค่าที่ได้จากการนับจะเป็น 0000H ทำให้เกิดการ Set บิทบางบิทในรีจิสเตอร์ TCON เพื่อบอกสภาวะ Timer Overflow นี้ ในการให้วงจร Timer ทำงานเป็น Counter ก็คือการใช้รีจิสเตอร์ THx และ TLx ทำการนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 ของ 8051 สัญญาณที่เข้ามาทางขา T0 หรือ T1 อาจจะมาจากอุปกรณ์ตรวจจับ (Sensor) ก็ได้แต่สภาวะของสัญญาณนี้จะต้องมีระดับโวลท์เตจของสภาวะลอจิก 0 หรือ 1 เป็นแบบ TTL คือลอจิก 0 จะต้องมีโวลท์เตจไม่เกิด 0.6 โวลท์ และลอจิก 1 จะต้องมีโวลท์เตจมากกว่า 2.4 โวลท์

10. TMOD Timer/Counter mode register

ตำแหน่งหน่วยความจำภายในเท่ากับ 89H

TM0D เป็นรีจิสเตอร์ขนาด 8 บิท ที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer 1 แต่ละบิท ในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 2.54

GATE	C/T	M1	M0	GATE	C/T	M1	M0

|<----------Timer 1---------->|<----------Timer 0---------->|

GATE When TRx (in TCON) is set and GATE=1. TIMER/COUNTERx will run only while

iNTx pin is high (hardware control). When GATE=0. TIMER/COUNTERx will run only while TRx=1 (software control).

C/T Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1 :

M1	M0	Operating Mode	
0	0	0	13-bit Timer
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0)TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

รูปที่ 2.54 TM0D Timer/Counter Mode Register

ในรูปที่ 2.54 M0 เป็นชื่อของบิท 0 และ GATE ทางซ้ายสุดเป็นชื่อของบิท 7 รีจิสเตอร์นี้แบ่งข้อมูลออกเป็น 2 ชุด ชุดละ 4 บิท คือบิท 0-3 ใช้สำหรับควบคุมการทำงานของ Timer 0 และบิท 4-7 ใช้ควบคุมการทำงานของ Timer 1 หน้าที่ในการควบคุม Timer ของแต่ละบิทที่มีชื่อเดียวกันจะเหมือนกัน

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

C/T บิทนี้ใช้สำหรับเลือกการทำงานของ Timer ว่าจะใช้เป็น Timer หรือ Counter ถ้าบิทนี้เป็น 1 ก็หมายความว่าเลือกการทำงานเป็น Counter ซึ่งจะนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา Tx

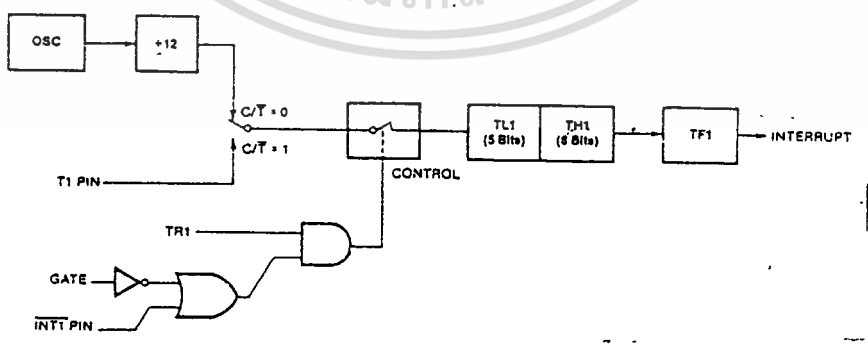
M1, M0 เป็น 2 บิทที่ใช้ร่วมกันเพื่อเลือกโหมดการทำงานของ Timer การทำงานโหมด 0, 1 และ 2 ของ Timer จะเหมือนกับ Timer 1 แต่ในโหมด 3 การทำงานของทั้งสองจะต่างกัน ค่าใน M1 และ M0 จะเลือกโหมดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

การทำงานของแต่ละโหมดจะมีรายละเอียดดังนี้

โหมด 0



รูปที่ 2.55 Timer Mode 0 : 13 bit count

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.55 เป็นไดอะแกรมของวงจร Timer ภายใน 8051 ที่ทำงานในโหมด 0 ซึ่ง Timer 0 และ Timer 1 ก็จะมีการทำงานเหมือนกันทุกประการ ในการอธิบายนี้จะใช้วงจรของ Timer 1 จากรูปจะเห็นสวิทช์ C/T ซึ่งถ้ากำหนดค่าในบิท C/T ของ TMOD เป็น 0 จะทำให้สวิทช์อยู่ในตำแหน่งบนเพื่อให้สัญญาณนาฬิกาที่ออกจากวงจรออสซิลเลเตอร์ผ่านวงจรหาร 12 ไปยังสวิทช์ Control ถ้าออสซิลเลเตอร์ผลิตสัญญาณนาฬิกาความถี่ 12 เมกกะเฮิรตซ์ก็จะมีสัญญาณความถี่ 1 เมกกะเฮิรตซ์ออกจากวงจรหาร 12 ถ้าบิท C/T เป็น 1 จะทำให้สวิทช์ C/T อยู่ในตำแหน่งข้างล่าง เพื่อให้สัญญาณที่เข้ามาทาง T1 (หรือ T0 ถ้าเป็น Timer 0) ผ่านไปยังสวิทช์ Control สัญญาณที่เข้ามายังสวิทช์ Control จะส่งผ่านไปยังวงจรนับหรือไม่ก็ขึ้นอยู่กับสัญญาณควบคุมที่ออกมาจาก AND GATE ถ้าบิท TR1 (หรือ TR0 ถ้าเป็น Timer 0) ในรีจิสเตอร์ TCON เป็น 0 จะทำให้สภาวะของสัญญาณที่ออกจาก AND GATE เป็น 0 เสมอ และจะไม่มีสัญญาณใดออกจากสวิทช์ Control ไปยังวงจรนับเลย รีจิสเตอร์ TL1 และ TH1 จะไม่ทำงานแต่ถ้าบิท TR1 เป็น 1 จะทำให้สภาวะของสัญญาณออกจาก AND GATE ไปควบคุมสวิทช์ Control ขึ้นกับสภาวะของสัญญาณ INT1 (หรือ INT0 ถ้าเป็น Timer 0) และข้อมูลที่บิท GATE ของรีจิสเตอร์ TM0D ถ้าบิท GATE เป็น 0 หรือสัญญาณที่ขา INT1 มีสภาวะลอจิกเป็น 1 จะทำให้สัญญาณควบคุมสวิทช์ Control เป็น 1 ทำให้มีสัญญาณออกไปยังตัวนับรีจิสเตอร์ TL1 และ TH1 (หรือ TL0 และ TH0 ถ้าเป็น Timer 0) รีจิสเตอร์ TI1 จะทำการนับโดยมีการนับเพียง 5 บิทเท่านั้น (ทำหน้าที่เป็นวงจร Prescaler ขนาด 5 บิท) ซึ่งนับได้ตั้งแต่ 0 ถึง 31 เมื่อ TL1 นับสัญญาณที่ออกจากสวิทช์ Control ครบ 32 ไซเคิลจะมีสัญญาณส่งไปยัง TH1 ไซเคิล บิท 5 ถึง 7 ของ TLx ที่ไม่ได้ใช้งานก็จะไม่ต้องสนใจ การทำงานของ Timer 0 และ 1 ในโหมดนี้จะเหมือนกับการทำงานของ Timer ในไมโครคอนโทรลเลอร์เบอร์ 8048 ทุกประการ

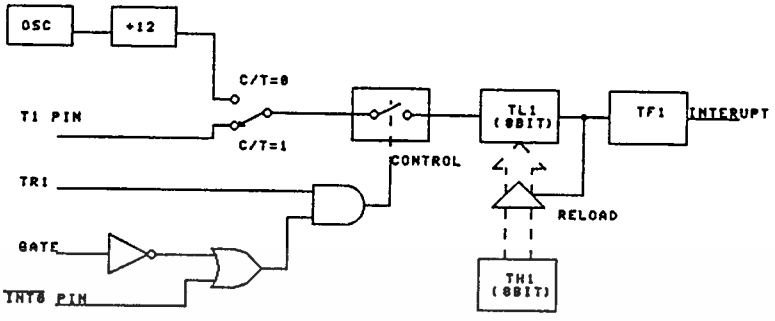
โหมด 1

ในโหมดนี้จะมีการทำงานของวงจรภายในของ Timer 0 หรือ 1 เหมือนกับโหมด 0 ทุกประการแตกต่างกันที่ TLx จะถูกใช้งานทั้ง 8 บิท ทำให้ผลการนับใน TLx และ THx จะมีถึง 16 บิท

โหมด 2

ในรูปที่ 2.56 เป็นไดอะแกรมของวงจร Timer 1 ใน 8051 ที่ทำงานโหมด 2 Timer 0 และ Timer 1 มีการทำงานในโหมด 2 เหมือนกันโดยจะสามารถกำหนดให้ทำหน้าที่เป็น Timer หรือ Counter ได้โดยบิท C/T และควบคุมการนับได้โดยข้อมูลในบิท TR1 และ GATE ในรีจิสเตอร์ TM0D กับสัญญาณที่ขา INTx เมื่อเริ่มการทำงานข้อมูลในรีจิสเตอร์ TH1 จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.56 Timer Mode 2

โหลด (Load) ไปยังรีจิสเตอร์ TL1 ทำให้รีจิสเตอร์ TH1 และ TL1 มีค่าเหมือนกัน เมื่อเกิดการนับจำนวนไซเคิลของสัญญาณที่ออกจากสวิทช์ Control จะทำให้ค่าจากการนับเป็นรีจิสเตอร์ TL1 เพิ่มขึ้นไปเรื่อย ๆ ทีละ 1 จนถึง 0FFH ในการนับครั้งต่อไปจะทำให้บิท TF1 ในรีจิสเตอร์ TCON ไม่เป็น 1 และข้อมูลในรีจิสเตอร์ TH1 จะถูกโหลดไปยังรีจิสเตอร์ TL1 เพื่อเป็นค่าเริ่มต้นการนับต่อไป

โหมด 3

การทำงานโหมด 3 ของ Timer 0 และ 1 จะต่างกัน

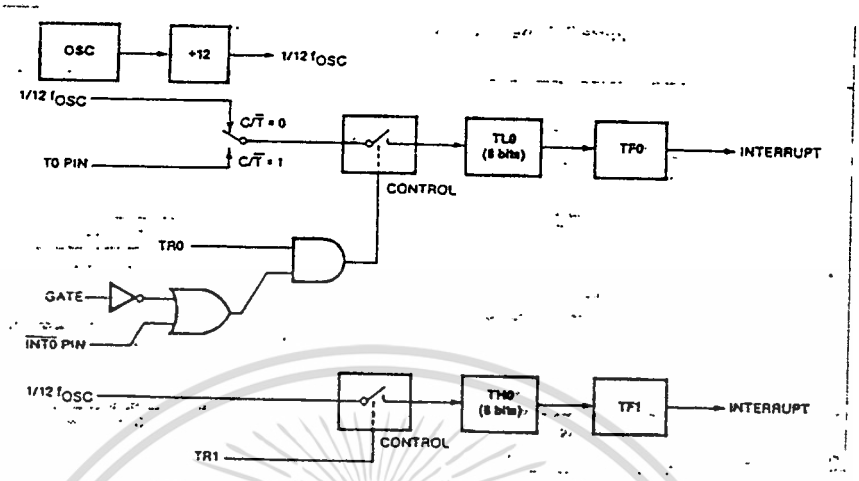
Timer 1 ในโหมด 3 จะไม่ทำงาน

Timer 0 ในโหมด 3 จะทำงานเป็นตัวนับที่เสมือนมีตัวนับ 8 บิทอยู่ 2 ตัวคือ TL0 และ TH0 ทำงานแยกกันดังรูปที่ 2.56

รีจิสเตอร์ TL0 จะเป็นตัวนับ 8 บิท ที่มีการนับสัญญาณจากออสซิลเลเตอร์หารด้วย 12 หรือนับสัญญาณที่เข้ามาทางขา T0 ขึ้นกับบิท C/T ในรีจิสเตอร์ TM0D และการนับจะควบคุมโดยบิท TR0 และ GATE ในรีจิสเตอร์ TM0D กับสภาวะลอจิกของสัญญาณที่ขา INT0 เหมือนกับในการทำงานโหมด 0, 1 และ 2 แต่ค่าจากการนับสิ้นสุดจะมีเพียง 255 เท่านั้น เมื่อค่าการนับเปลี่ยนจาก 0FFH เป็น 00H คือเกิดการ overflow จะทำให้บิท TF0 ถูก SET เป็น 1 และอาจเกิดการขัดจังหวะ (Interrupt) การทำงานของโปรแกรมได้ถ้ามีการกำหนดค่าในรีจิสเตอร์ IE และ IP

ตัวนับอีกตัวคือรีจิสเตอร์ TH0 จะทำงานในโหมดของ Timer เท่านั้น คือจะนับจำนวนไซเคิลของสัญญาณที่ออกจากออสซิลเลเตอร์แล้วหารด้วย 12 การนับจะควบคุมได้ด้วยบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.57 Timer 0 Mode 3

TR1 ในรีจิสเตอร์ TM0D ถ้าบิทนี้เป็น 1 ก็จะมีสัญญาณเข้าไปยัง TH0 แต่ถ้าบิทนี้เป็น 0 ก็จะไม่มีสัญญาณเข้าไปยัง TH0

11. TCON Timer Control Register ตำแหน่งหน่วยความจำภายในเท่ากับ 088H

รีจิสเตอร์ขนาด 8 บิทนี้ใช้ควบคุมการทำงานและบอกสภาวะของ Timer 0 และ Timer 1 แต่บิทของรีจิสเตอร์จะทำงานต่างกันดังรูปที่ 2.58

ในรูปที่ 2.58 T0 เป็นชื่อของบิท 0 และ TF เป็นชื่อของบิท 7 ในรีจิสเตอร์ TCON แต่ละบิทมีหน้าที่การทำงานดังนี้

IT0 Interrupt 0 เป็นบิทที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องจากสภาวะของสัญญาณที่เข้ามาทางขา INTO

ถ้า IT0 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT0 เปลี่ยนจาก 1 เป็น 0

ถ้า IT0 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INTO เป็น 0

IT1 Interrupt 1 เป็นบิทที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรมอันเนื่องจากสภาวะของสัญญาณที่เข้ามาทางขา INT1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1ONOFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify failling edge/flow level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by nardware when interrupt is processed.
ITO	TCON.0	Interupt 0 type control bit. Set/cleared by software to specify failling edge/low level triggered External Interrupt.

รูปที่ 2.58 TCON Timer Control register

ถ้า IT1 เป็น 1 จะเกิดการขัดจังหวะโปรแกรมถ้าสภาวะของสัญญาณที่ขา INT1 เปลี่ยนจาก 1 เป็น 0

ถ้า IT1 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT1 เป็น 0

IE1 บิทนี้จะเป็น 1 ถ้ามีสัญญาณเข้ามาทางขา INT1 มีสภาวะลอจิกของสัญญาณ

TR0 Timer 0 Run Control Bit บิทนี้ถ้าเป็น 0 Timer 0 ไม่ทำการนับสัญญาณ ไม่ว่ากรณีใด ๆ ทั้งสิ้น แต่ถ้าบิทนี้เป็น 1 จะทำให้ Timer 0 ทำงาน โดยขึ้นกับสัญญาณ GATE, INTO ข้อมูลในบิทนี้จะสามารถ Set เป็น 1 หรือ Clear เป็น 0

TF0 Timer 0 Overflow flag บิทนี้จะเป็น 1 เมื่อการนับของรีจิสเตอร์ใน Timer 0 (TL0 หรือ TH0 ขึ้นกับโหมดของการทำงาน) เกิด Overflow ขึ้น คือเอาการนับเพิ่มไปจนถึงค่า กระโดดไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะจะทำให้บิทนี้กลับเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TR1 Timer Run Control Bit การทำงานจะเหมือนกับการทำงานของบิท TR0 แต่บิทนี้จะทำงานกับ Timer 1

TF1 Timer 1 Overflow Flag บิทนี้เหมือนกับบิท TF0 ต่างกันที่ขึ้นอยู่กับการทำงานของ Timer 1

4 บิทแรกที่กล่าวมานั้นจะเกี่ยวข้องกับการขัดจังหวะ (Interrupt) ส่วน 4 บิทหลังนั้นได้กล่าวมาแล้วอย่างละเอียดในเรื่องโหมดการทำงานของ Timer

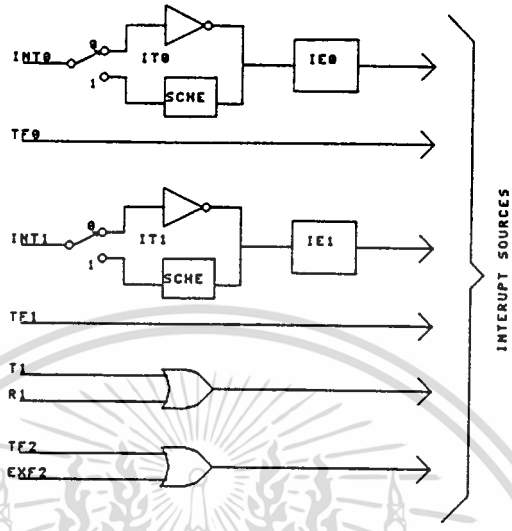
ในขณะที่ Timer ทำงานในโหมดของ Timer นั้น รีจิสเตอร์ที่ทำหน้าที่เป็นตัวนับจะมีค่าเพิ่มขึ้น 1 ทุก ๆ 1 ไซเคิลของเครื่อง ซึ่งเท่ากับ 12 คาบของสัญญาณจากออสซิลเลเตอร์ ในกรณีที่ Timer ทำงานเป็น Counter เพื่อนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 รีจิสเตอร์จะเพิ่มค่าไป 1 เมื่อมีการเปลี่ยนสภานะของสัญญาณที่ขาดังกล่าวจาก 1 เป็น 0 โดยวงจรภายใน 8051 จะตรวจสอบสถานะของสัญญาณที่ขาดังกล่าวในช่วงเวลาเฟส 2 ของ State5 (S5P2) ในทุก ๆ 1 ไซเคิลของเครื่อง เช่นในเวลา S5P2 ครั้งหนึ่งพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 1 และในเวลา S5P2 ของไซเคิลของเครื่องถัดมาพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 0 ก็จะทำให้ค่าในรีจิสเตอร์ตัวนับเพิ่มค่าไป 1 แต่ละไซเคิลของเครื่องจะเท่ากับ 12 ไซเคิลของสัญญาณจากออสซิลเลเตอร์ ดังนั้นสัญญาณที่จะนับได้จะต้องเป็น 1 อย่างน้อยให้ถูกจับได้ใน 1 ไซเคิลของเครื่องและเป็น 0 อย่างน้อยก็ต้องให้ถูกตรวจจับได้ใน 1 ไซเคิลของเครื่องเช่นกัน ทำให้ความถี่สูงสุดของสัญญาณที่ขา T0 หรือ T1 ที่จะนับได้ต้องไม่เกิน 1/24 เท่าของความถี่สัญญาณจากออสซิลเลเตอร์

12. IE Interrupt Enable Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0A8H

การขัดจังหวะการทำงาน (Interrupt) เป็นการที่มีสัญญาณหนึ่งหรือคำสั่งหนึ่งที่ (ไม่ใช่คำสั่ง CALL หรือ JMP) ที่จะทำให้การทำงานการปกติของโปรแกรมถูกขัดจังหวะ แล้วข้ามไปทำงานยังตำแหน่งหนึ่งตำแหน่งใดที่กำหนดไว้ เมื่อทำงานในโปรแกรมขัดจังหวะเสร็จสิ้นก็จะกลับมาทำงานในโปรแกรมที่ตำแหน่งก่อนจะไปทำงานยังโปรแกรมขัดจังหวะโปรแกรมที่ถูกกระโดดไปทำงานเรียกว่า โปรแกรมตอบสนองการขัดจังหวะ (Interrupt Service Routine) ใน 8051 จะสามารถขัดจังหวะด้วยสัญญาณจาก 6 แหล่งดังรูปที่ 2.59 ถ้าเป็น 8051 หรือ 83154 สามารถขัดจังหวะได้ด้วยสัญญาณจาก 8 แหล่งคือสัญญาณในชุดของรูปที่ 2.59

สัญญาณขัดจังหวะที่ 5 ในรูปที่ 2.59 จะสามารถทำให้เกิดการขัดจังหวะได้ 2 วิธีคือ มีข้อมูลเข้ามาทางพอร์ทอนุกรมเก็บอยู่ที่รีจิสเตอร์ SBUF และกรณีที่ข้อมูลใน SBUP ส่งออกไปทางพอร์ทอนุกรมหมดแล้ว ไม่ว่ากรณีใด ๆ ก็ทำให้เกิดการขัดจังหวะขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.59 แหล่งกำเนิดสัญญาณขัดจังหวะ

สัญญาณภายนอกที่เข้ามายัง 8051 ทางขา INT0 และ INT1 จะสามารถทำให้เกิดการขัดจังหวะการทำงาน 8051 ได้ (สัญญาณที่ 1 และ 3 ในรูปที่ 2.59) โดยสภาวะของสัญญาณนั้นเปลี่ยนจาก 1 เป็น 0 หรือเมื่อสัญญาณนั้นเป็น 0 แล้วแต่การกำหนดในบิท IT0 และ IT1 ของรีจิสเตอร์ TCON จะทำให้บิท IE0 กับ IE1 เป็นตัวสร้างสัญญาณขัดจังหวะต่อไป

จาก Timer 0 และ Timer 1 เมื่อค่าการนับในแต่ละโหมดถึงค่าสูงสุดในโหมดนั้นแล้ว เมื่อทำการนับต่อไปค่าการนับต่อไปจะเป็น 0 (หรืออาจเป็นค่าที่ Reload จาก THx ในโหมด 2) และทำให้บิท TF0, TF1 เป็น 1 ซึ่งสัญญาณจาก 2 บิทนี้จะสามารถทำให้เกิดการขัดจังหวะได้เช่นกัน ดังเช่น สัญญาณขัดจังหวะที่ 2 และ 4 ในรูปที่ 2.59

แหล่งกำเนิดสัญญาณทั้ง 6 ที่สามารถทำให้เกิดการขัดจังหวะได้ 5 แบบนี้ผู้ใช้สามารถกำหนดให้สัญญาณใดบ้างเกิดการขัดจังหวะเรียกว่า Enable หรือไม่ให้เกิดการขัดจังหวะ เรียกว่า Disable โดยการกำหนดในรีจิสเตอร์ IE (Interrupt Enable Register) ซึ่งมี 8 บิท แต่ละบิทสามารถ Enable ให้ขัดจังหวะได้จากแต่ละสัญญาณดังรูปที่ 2.60

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(MSB) (LSB)

$\overline{EA}$	X	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
$\overline{EA}$	IE.7	disables all interrupts.if $\overline{EA}$=0. no interrupt will be acknowledged.if $\overline{EA}$=1 each interrupt source is individuaily enabled or disabled by setung or clearing its enable bit.
ET2	IE.6	reserved
ES	IE.5	enables or disables the Timer 2 Overflow or capture interrupt.it ET2=0. the Timer 2 interrupt is disabled.
ET1	IE.4	enables or disables the Serial Port interrupt il ES=0. the Serial Port interrupt is disabled.
EX1	IE.3	enablesor disables the Timer 1 Overflow interrupt if .ET1=0.the Timer 1 interrupt is disabled.
ET0	IE.2	enables or disables Externalinterrupt 1 if EX1=0. Externalinterrupt 1 is disabled.
EX0	IE.1	enables or disables the Timer 0 Overflow interrup. it ET0=0 Overflow interrupt. if ET0=0. the Timer 0 interrupt is disabled.
	IE.0	enables or disables Externalinterrupt 0. it EX0=0. Externalinterrupt 0 is disabled.

รูปที่ 2.60 Interrupt Enable Register

EX0 บิทนี้ใช้สำหรับการ Enable สัญญาณที่เข้ามาทางขา INT0 ให้เกิดการขัดจังหวะหรือไม่

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

EX1 บิทนี้จะใช้ Enable หรือ Disable สัญญาณที่เข้ามาทางขา INT1 ให้เกิดการขัดจังหวะหรือไม่

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EX0 บิทนี้ใช้สำหรับการ Enable สัญญาณที่เข้ามาทางขา INT0 ให้เกิดการขัดจังหวะหรือไม่

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

EX1 บิทนี้จะใช้ Enable หรือ Disable สัญญาณที่เข้ามาทางขา INT1 ให้เกิดการขัดจังหวะหรือไม่

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

ES ข้อมูลในบิทนี้จะ Disable หรือ Enable การขัดจังหวะจาก Serial Port อันเนื่องจากมีข้อมูลเข้ามายัง SBUF หรือข้อมูลจาก SBUF ได้ส่งออกไปทาง Serial Port หมดแล้ว

ET2 Timer 2 Internal Enable bit จะใช้งานเฉพาะใน 8052 และ 83152 เท่านั้นบิทนี้จะใช้ Enable หรือ Disable สัญญาณของการขัดจังหวะที่มาจาก Timer 2 (สัญญาณที่ 6 ในรูปที่ 2.58)

EA บิทนี้จะควบคุมทั้ง 6 บิทที่กล่าวมาแล้ว ถ้าข้อมูลในบิทนี้เป็น 0 จะเป็นการ Disable ทุกบิทที่กล่าวมาแล้ว ทำให้ไม่เกิดการขัดจังหวะโปรแกรมได้เลยแต่ถ้าบิทนี้เป็น 1 การ Enable/Disable ใน 6 บิทที่กล่าวมาแล้วจะขึ้นกับข้อมูลในแต่ละบิทนั้นบิทอื่นนอกจากนี้จะไม่มีการใช้งาน

การกำหนดให้บิทใด Enable หรือ Disable นั้นจะเป็นการไปโดยอิสระไม่ขึ้นแก่กันจึงสามารถกำหนดให้บิทใดหรือมากกว่า 1 บิท Enable ก็ได้ ดังนั้น 8051 จึงมีรีจิสเตอร์อีกตัวที่ใช้เลือกว่าถ้ามีสัญญาณของการขัดจังหวะโปรแกรมเข้ามาพร้อมกันมากกว่า 1 แล้วจะทำโปรแกรมตอบสนองการขัดจังหวะอันใดก่อน รีจิสเตอร์นั้นคือ IP Interrupt Priority Register

13. IP Interrupt Proiority register

ตำแหน่งหน่วยความจำภายในเท่ากับ 0B8H

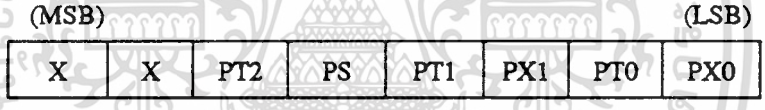
ในการตอบสนองต่อสัญญาณขัดจังหวะของ 8051นั้น ถ้าสัญญาณขัดจังหวะทั้งหมดเข้ามาพร้อมกัน 8051 จะต้องเลือกทำงานโปรแกรมตอบสนองการขัดจังหวะโดยการตรวจสอบสัญญาณเรียงตามลำดับ ซึ่งเรียกว่าวิธีการ Polling สัญญาณขัดจังหวะหนึ่งจะถูกตรวจสอบก่อนแล้วสัญญาณอื่น ๆ จะถูกตรวจสอบก่อน ถ้าสัญญาณนั้นขอขัดจังหวะ 8051 จะสร้างคำสั่ง CALL เป็นพิเศษขึ้นมาเพื่อไปทำงานโปรแกรมตอบสนองการขัดจังหวะของสัญญาณนั้น เมื่อเสร็จสิ้นแล้วก็จะกลับมาทำงานในโปรแกรมเดิมก่อนการขัดจังหวะ ทำให้เสมือนว่าสัญญาณแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณมีลำดับความสำคัญไม่เท่ากัน สัญญาณขัดจังหวะจะมีลำดับความสำคัญดังนี้ โดยเรียงจากลำดับความสำคัญสูงสุดถึงต่ำสุด

1. IE0
2. TF0
3. IE1
4. TF1
5. RI + TI

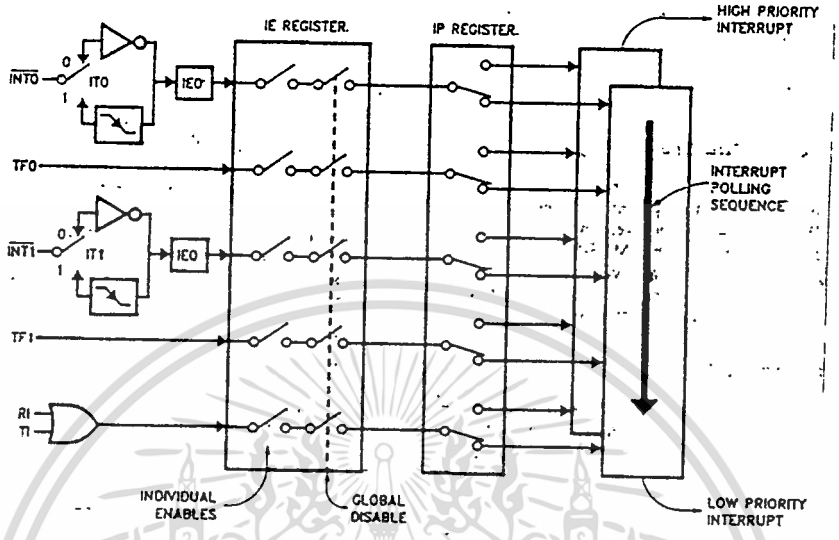
แต่ในการใช้งาน บางครั้งจำเป็นที่จะต้องให้สัญญาณใดสัญญาณหนึ่งมีลำดับความสำคัญสูงสุด (Highest Priority) เพื่อจะทำงานโปรแกรมตอบสนองการขัดจังหวะได้ก่อนการขัดจังหวะของสัญญาณอื่น จะสามารถกำหนดลำดับความสำคัญของการขัดจังหวะได้ใหม่โดยการกำหนดข้อมูลในบิทของรีจิสเตอร์ IP (Interrupt Priority Register) ตามตำแหน่งของแต่ละบิทในรูปที่ 2.60



Symbol	Position	Function
PCT	IP.7	PCT=1,only one level
-	IP.6	reserved
PE2	IP.5	defines the Timer2 interrupt priority level.PT2=1 programs it to the higher priority level.
PS	IP.4	efines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.1	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer0 interrupt priority level.PT0=1 programs it to the higher priority level.
PX0	IP.0	defines the External interrupt 0 prionty level. PX0 = 1 programs it to the higher priority level.

รูปที่ 2.61 IP : Interrupt Prioriry Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.62 ระบบการขัดจังหวะของ 8052 และ 83154

รูปที่ 2.62 เป็นแผนภาพแสดงระบบขัดจังหวะของ 8052 ซึ่งแตกต่างจากของ 8051 ตรงที่ 8052 จะมีสัญญาณขัดจังหวะมาจาก TF2, EFX2 คือชุดล่างในภาพ

ในรูปจะเห็นว่าแต่ละสัญญาณจะมีสวิทช์ควบคุมอยู่ 3 ตัว 2 ตัวแรกอยู่ในกรอบสี่เหลี่ยม IE Register และอีก 1 สวิทช์อยู่ในกรอบ IP Register สวิทช์ตัวแรกทางซ้ายสุดจะควบคุมด้วยข้อมูลแต่ละบิท บิท 0 ถึงบิท 5 ของรีจิสเตอร์ IE ถ้าข้อมูลเป็น 1 จะทำให้สวิทช์นั้นปิดวงจร (Closed circuit) การควบคุมสวิทช์ทางซ้ายสุดของแต่ละสัญญาณจะไม่ขึ้นแก่กัน (Individual) สวิทช์ที่ 2 ถัดมาของทุกสัญญาณจะควบคุมร่วมกันด้วยบิท EA ในรีจิสเตอร์ IE ถ้าบิทนี้เป็น 0 สวิทช์ที่ 2 ของทุกสัญญาณจะเปิดวงจร (Opened Circuit) ทำให้ไม่มีสัญญาณขอขัดจังหวะผ่านไปได้สวิทช์ที่ 3 ทางขวาสุดจะใช้สำหรับเลือกว่าสัญญาณนั้นจะอยู่ในกลุ่มลำดับความสำคัญสูง (High Priority Interrupt) หรือลำดับความสำคัญต่ำ (Low Priority Interrupt)

ถ้าต้องการสัญญาณใดมีลำดับความสำคัญสูงก็ให้กำหนดบิทนั้นในรีจิสเตอร์ IP เป็น 1 สวิทช์ที่ 3 จะเลื่อนไปอยู่ในตำแหน่งบน ถ้าไม่ต้องการก็กำหนดให้บิทนั้นเป็น 0 บิทใดเป็น 1 เรียกว่าสัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญสูงและบิทใดเป็น 0 เรียกว่าสัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญต่ำ ถ้าในกลุ่มลำดับความสำคัญสูงมีเพียง 1 สัญญาณก็จะเรียกว่าสัญญาณนั้นมีลำดับความสำคัญสูงสุด ในกลุ่มลำดับความสำคัญเดียวกันก็จะมีการจัดลำดับความสำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉพาะกลุ่มโดยวิธี Polling เหมือนเดิม เช่นกรณีที่มีการกำหนดในบิทของรีจิสเตอร์ IP ให้มีลำดับความสำคัญสูงหรือต่ำเหมือนกันแล้วเกิดมีความต้องการขอการขัดจังหวะจากสัญญาณนั้น ๆ มาพร้อมกัน 8051 ก็จะทำงานในโปรแกรมตอบสนองการขัดจังหวะเรียงตามลำดับความสำคัญ 5 ลำดับที่กล่าวมาแล้วข้างต้น เช่นให้ PT1, PX1 และ PT0 เป็น 1 เมื่อมีสัญญาณขอขัดจังหวะการทำงานมาจาก Timer1, External interrupt 1 และ Timer 0 พร้อมกัน 8051 ก็จะทำงานตอบสนองการขัดจังหวะของ Timer 0, External interrupt 1 และ Timer 1 ตามลำดับ ในขณะที่ 8051 กำลังทำงานโปรแกรมตอบสนองการขัดจังหวะของสัญญาณขัดจังหวะที่มีลำดับความสำคัญต่ำอยู่ ถ้ามีสัญญาณขัดจังหวะที่มีลำดับความสำคัญสูงกว่าเกิดขึ้น การทำงานของโปรแกรมก็จะกระโดดไปทำงานในตำแหน่งโปรแกรมตอบสนองการขัดจังหวะของสัญญาณที่มีลำดับความสำคัญสูงเสร็จแล้วจึงกลับมาทำงานที่โปรแกรมตอบสนองการขัดจังหวะลำดับความสำคัญต่อไป แต่ละบิทของรีจิสเตอร์ IP นั้นจะบอกลำดับความสำคัญของแหล่งกำเนิดสัญญาณขัดจังหวะดังนี้

PX0 บิท 0 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจากภายนอก 8051 คือ INT0

PT0 บิท 1 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 0

PX1 บิท 2 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจากภายนอก 8051 คือ INT1

PT1 บิท 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 1

PT2 บิท 5 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 2 บิทนี้ใช้เฉพาะใน 8052 ที่มี Timer 2

PS บิท 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะ Serial Port ในกรณีที่มีข้อมูลเข้ามาหรือส่งข้อมูลออกสิ้นสุดแล้ว

14. PCON (Power Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 87H

8051 เป็นไมโครคอนโทรลเลอร์ที่สร้างขึ้นด้วยเทคโนโลยีทั้งแบบ CHMOS และ HMOS ซึ่งแบบ CHMOS มีข้อดีตรงที่ใช้กำลังไฟต่ำกว่าแบบ HMOS ดังนั้นต่อไปในอนาคตจึงจะมีแต่เฉพาะรุ่น CHMOS เท่านั้น นอกจากนี้แล้ว 8051 ยังมีข้อดีอีกตรงที่สามารถลดการใช้กำลังไฟลงได้โดยการทำงานใน idle Mode และ Power Down Mode ใน Idle Mode นั้น สัญญาณนาฬิกาจากออสซิลเลเตอร์จะป้อนให้เฉพาะส่วน Interrupt, Serial Port และ Timer ในส่วนอื่นจะไม่มีสัญญาณนาฬิกาไปเลี้ยงแต่มีไฟเลี้ยงให้กับทุกส่วนในวงจร การใช้กำลังไฟจึงลดลงมาก ส่วนใน Power Down Mode นั้น ออสซิลเลเตอร์จะหยุดทำงานทำให้ไม่มีสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปเลี้ยงส่วนใด ๆ ในวงจรเลยแต่ข้อมูลภายในรีจิสเตอร์จะยังคงอยู่ไม่สูญหายไป รายละเอียดของแต่ละโหมดจะได้กล่าวต่อไป

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 bort is being used in either modes 1,2 or 3
HPD	PCON.6 (83C154 only)	Hard Power Down bit.Setting this bit allows CPU to enter in Power Down state on an external event(1 to 0 transition)on bit T1(p.3-5)the CPU quit the Hard Power Down mode when bit T1(p.3-5)fo high or when reset is activated.
RPD	PCON.5 (83C154 only)	Recover from idle or Power Down bit.When 0 RPD has no erfect.When 1,RPO permits to exit from idle or Power Down with any non enabled interrupt source (except times 2). In this case the program start at the next address. When interrupt is enabled the appropriate interrupt routine is serviced.
-		(Reserved)
GF1	PCON.4	General-purpose flag bit.
GF0	PCON.3	General-purpose flag bit.
PD	PCON.2	Power Down bit.Setting this bit activates power down operation.
IDL	PCON.1	Ice mode bit.Setting this bit activates idle mode operation.
	PCON.0	

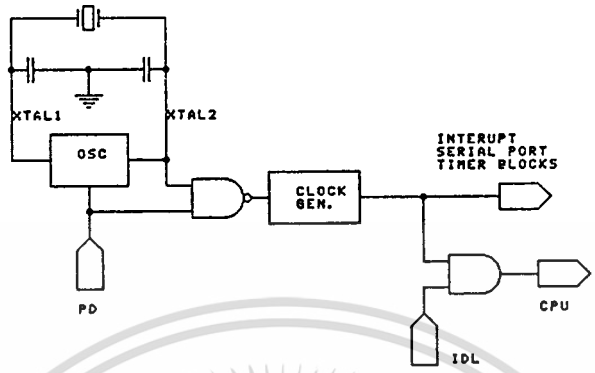
รูปที่ 2.63 PCON : Power Control Register

การสั่งงานให้ 8051 ทำงานในโหมดของ Idle หรือ Power Down จะสามารถทำได้โดยใช้กำหนดค่าในรีจิสเตอร์ PCON (Power Control Register) แต่ละบิทในรีจิสเตอร์ PCON มีดังนี้

IDC บิท 0 ถ้าบิทนี้ถูก Set ให้เป็น 1 8051 จะเข้าสู่การทำงานใน Idel Mode ทันที

PD บิท 1 ถ้าบิทนี้ถูก Set ให้เป็น 1 จะเข้าสู่การทำงานใน Power Down Mode ทันที GF0, GF1 บิท 2,3 เป็น General Purpose Flag Bit บิทนี้สามารถกำหนดให้มีค่าเป็น 0 หรือ 1 ได้โดยโปรแกรม เพื่อส่งผ่านสถานะการทำงานของ 8051 ระหว่างแต่ละโปรแกรมย่อย SM0D บิท 7 เป็นบิทที่ใช้ร่วมในการกำหนดอัตราการส่งข้อมูล (Baud Rate) ผ่านทางพอร์ทอนุกรม ซึ่งในการรับ-ส่งข้อมูลผ่านทางพอร์ทอนุกรม โหมด 1 และ 3 จะสามารถกำหนดอัตราการส่งข้อมูลได้ตามอัตราการเกิด Overflow ใน Time 1 ถ้าบิทนี้เป็น 1 จะทำให้อัตราการส่งข้อมูลเพิ่มขึ้น 2 เท่า บิท 4,5,6 ไม่ได้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.64 Power down และ Idle Mode Idle Mode

ในรูปที่ 2.64 ขณะที่ 8051 ทำงานตามปกติไปจนถึงคำสั่งที่ทำให้บิท 0 ของรีจิสเตอร์ PM0D มีค่าเป็น 1 ก็จะเข้าสู่การทำงานใน Idle Mode โดยสัญญาณ IDL จะเป็น LOW สัญญาณจะตรงข้ามกับข้อมูลในบิท (0) ขณะนี้สัญญาณนาฬิกาจากออสซิลเลเตอร์จะไม่ออกจาก AND GATE ไปยังส่วน CPU โดยจ่ายเฉพาะส่วน Interrupt, Timer และ Serial Port ในขณะนี้ 8051 จะเสมือนหยุดการทำงานโดยข้อมูลใน Stack Pointer, Program Counter, Program Status Word, Accumulator และรีจิสเตอร์อื่น ๆ จะไม่เปลี่ยนแปลงข้อมูลที่พอร์ทต่าง ๆ จะยังคงค่าเดิมไว้เหมือนกับก่อนเข้าสู่ Idle mode และสัญญาณ ALE กับ PSEN จะเป็นลอจิก Hight ขณะนี้การใช้กระแสไฟของ 8051 จะลดต่ำลงมาก เนื่องจากภายใน 8051 จะไม่มีการเปลี่ยนสถานะลอจิกการที่จะออกจาก Idle Mode ทำได้ 2 วิธี

วิธีที่ 1 โดยการขัดจังหวะจากสัญญาณจังหวะทั้ง 6 ที่กล่าวมาแล้ว เมื่อมีสัญญาณขอขัดจังหวะจากแหล่งใดก็ตาม จะทำให้บิท 0 ของรีจิสเตอร์ PCON มีค่าเป็น 0 และการทำงานของ 8051 จะออกจาก Idle mode โดยกระโดดไปทำงานยังตำแหน่งของโปรแกรมตอบสนองการขัดจังหวะนั้น ๆ เมื่อเสร็จสิ้นการทำงานในโปรแกรมตอบสนองการขัดจังหวะโดยการทำงานคำสั่ง RET1 ก็จะกลับมาทำงานยังคำสั่งที่อยู่ต่อจากคำสั่งที่ทำให้บิท 0 ของรีจิสเตอร์ PMOD เป็น 1 ซึ่งทำให้การทำงานเข้าสู่ Idle Mode เช่นคำสั่งที่ตำแหน่ง 2000H คือ MOV PCON,#1H ที่เป็นคำสั่งที่ทำให้บิท IDC มีค่า 1 ดังนั้นเมื่อทำงานที่คำสั่งนี้เสร็จสิ้นก็จะหยุดการทำงาน และเมื่อเกิดการขัดจังหวะ เนื่องจากสัญญาณขัดจังหวะใด ๆ ก็ตาม 8051 จะออกจาก Idle Mode ไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะเมื่อเสร็จสิ้นการทำงานโปรแกรมตอบสนองการขัดจังหวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วจะกระโดดมาทำงานที่ตำแหน่งของคำสั่งต่อจากคำสั่ง MOV PCON, #1H

วิธีที่ 2 ก็คือการป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปยังขา RST เพื่อทำการรีเซท 8051 สัญญาณรีเซทนี้จะต้องมีลอจิกเป็น 1 ในระหว่างนี้ 8051 จะทำงานในคำสั่งต่อจากคำสั่งที่ทำให้บิท 0 ของ PCON เป็น 1 เข้าสู่ Idle Mode ต่อไปอีก 2-3 คำสั่ง ก่อนที่ทุกอย่างจะเข้าสู่การรีเซท ดังนั้นจะต้องระวังคำสั่งที่อยู่ต่อจากคำสั่งที่ทำให้เข้าสู่ Idle Mode อาจทำให้ข้อมูลบนพอร์ทเปลี่ยนแปลงจนทำให้อุปกรณ์ที่มาต่อเสียหายเมื่อกลับออกจาก Idle Mode

ในวิธีที่ 1 นั้นแสดงว่าการเข้าสู่โปรแกรมตอบสนองการขัดจังหวะจะเป็นได้ 2 กรณี คือ ขณะที่ทำงานตามปกติแล้วมีสัญญาณขัดจังหวะก็จะกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ หรือในกรณีที่อยู่ใน Idle Mode แล้วมีสัญญาณขัดจังหวะก็จะกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ จึงอาจให้โปรแกรมกำหนดข้อมูลในบิท GF0 หรือ GF1 หรือทั้งสองเพื่อให้โปรแกรมตอบสนองการขัดจังหวะรู้ได้ว่า การเข้าสู่โปรแกรมตอบสนองการขัดจังหวะนั้นมาจากกรณีใด

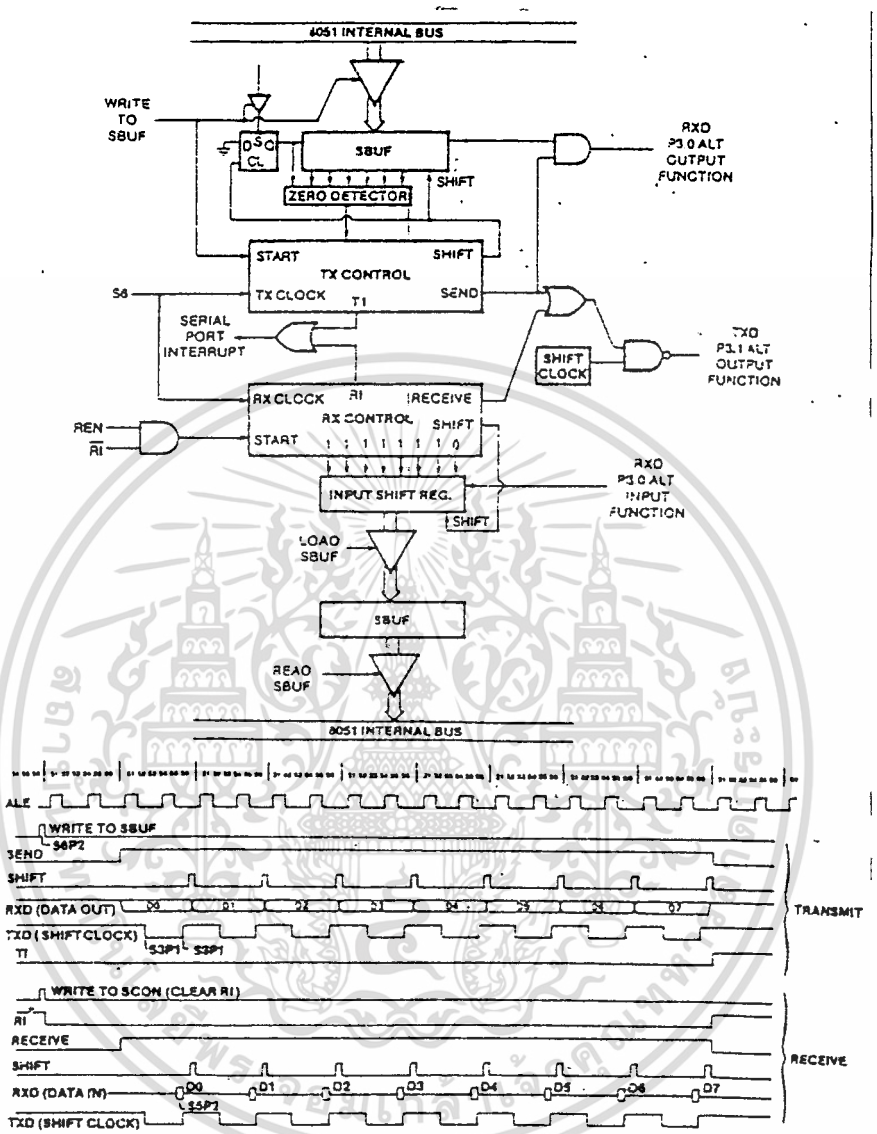
Power Down Mode

ในการเข้าสู่ Power down mode นั้นจะทำได้โดยการใช้โปรแกรมกำหนดให้บิท PD หรือบิท 1 ของรีจิสเตอร์ PCON มีค่าเป็น 1 เช่น MOV PCON, #2 เมื่อ 8051 ทำงานที่คำสั่งนี้เสร็จสิ้น สัญญาณ PD ในรูปที่ 2.64 จะเป็น 0 เพราะจะตรงข้ามกับข้อมูลในบิท PD ทำให้การทำงานจะเข้าสู่ Power Down Mode ทันที ในโหมดนี้ออสซิลเลเตอร์จะหยุดการทำงานทำให้ไม่มีสัญญาณนาฬิกาไปยังส่วนต่าง ๆ ภายใน 8051 ดังนั้น จะไม่มีการทำงานใด ๆ รวมทั้งข้อมูลในรีจิสเตอร์ทุกตัวจะไม่เปลี่ยนแปลง และข้อมูลใน RAM ภายในก็จะไม่เปลี่ยนแปลง ขณะนี้สัญญาณออกจากขา ALE และ PSEN จะเป็น 0 การใช้กำลังไฟของ 8051 จะต่ำมาก อีกทั้งสามารถลดไฟเลี้ยงวงจรที่ขา VCC ลงได้จนถึง 2 โวลท์ โดยไม่ทำให้ข้อมูลใด ๆ ใน 8051 สูญหายไปการออกจาก Power Down Mode ทำได้วิธีเดียวคือ ไปยังขา RST ของ 8051 ซึ่งทำให้เข้าสู่การรีเซท 8051 แต่จะทำให้ข้อมูลใน SFR เปลี่ยนแปลงไป ถ้าในขณะที่อยู่ใน Power Down Mode มีการลดไฟเลี้ยงวงจรจะต้องให้ไฟเลี้ยงวงจรกลับมาอยู่ที่ 5 โวลท์ก่อนที่จะเข้าสู่การรีเซท

2.6.9 การรับ-ส่งข้อมูลทางพอร์ทอนุกรม

รูปที่ 2.65 เป็นไดอะแกรมที่ทำงานในการรับ - ส่งข้อมูลทางพอร์ทอนุกรม โหมด 0 และไดอะแกรมสัญญาณตามเวลาของสัญญาณที่จุดต่าง ๆ ในวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.65 Serial Port Mode 0

ในการรับ-ส่งข้อมูลแบบอนุกรมผ่านทางพอร์ทอนุกรมนั้น จะต้องมีการกำหนดโหมดการทำงานในรีจิสเตอร์ SCON และในบางโหมดของการทำงานจะสามารถกำหนดอัตราการส่งข้อมูลได้โดยการโปรแกรมใน Timer ข้อมูลที่จะส่งออกหรือรับเข้าทางพอร์ทอนุกรมจะอยู่ที่รีจิสเตอร์ SBUF การทำงานของวงจรภายในแต่ละโหมดมีดังนี้

การทำงานโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูล

การส่งข้อมูลจะเริ่มจากการทำงานของคำสั่งให้เคลื่อนย้ายข้อมูลเข้ามายังรีจิสเตอร์ SBUF โดยจะมีสัญญาณ Write to SBUF ซึ่งเกิดขึ้นในช่วงเวลา S6P2 สัญญาณนี้จะทำให้ข้อมูลจากบัสภายใน 8051 (8051 Internal Bus) ทางด้านบนถึงนำไปเก็บที่รีจิสเตอร์ SBUF อีกทั้งยังทำการ Set ให้ D FLIF-FLOP ที่อยู่ทางซ้ายของ SBUF มีสภาวะลอจิกที่ขา Q เป็น 1 สัญญาณ Write to SBUF ซึ่งต่อเข้าไปยังขา Start ของ TX Control จะบอกให้ TX Control เริ่มส่งข้อมูล โดยจะหน่วงเวลาไว้ 1 ไซเคิลของเครื่องเมื่อผ่านเวลา 1 ไซเคิลของเครื่องไปแล้วขึ้นไซเคิลใหม่ สัญญาณ SEND จะเป็นสภาวะลอจิกจาก 0 เป็น 1 และเริ่มส่งข้อมูลบิท 0 จากนี้ในทุกเวลา S6P2 สัญญาณ SHIFT จะเปลี่ยนเป็น 1 ออกจากวงจร TX Control (ที่เวลาอื่นสัญญาณนี้จะเป็น 0) ทำให้ข้อมูลใน SBUF ถูกเลื่อนออกไปทีละบิทตรงขอบขาลงของสัญญาณ Shift ในทุก ๆ ไซเคิลของเครื่อง ขณะที่ข้อมูลถูกเลื่อนออกไปทางขวานี้ ข้อมูลจาก D-FF จะเลื่อนเข้ามาทางซ้าย ข้อมูลจาก D-FF บิทแรกที่เลื่อนเข้ามาจะเป็น 1 เพราะถูก Set ไว้ตอนแรก แต่บิทต่อมาจะเป็น 0 เพราะขา D ถูกต่อไว้ที่กราวด์ ขณะที่สัญญาณ SEND มีลอจิกเป็น 1 จะทำให้เอาท์พุทของ OR Gate มีลอจิกเป็น 1 ดังนั้นสัญญาณจากวงจร Shift Clock จะถูกส่งออกไปทางขา P3.1 (TXD) โดยสัญญาณนี้จะเป็น 1 ในช่วงเวลา S6P1 ถึง S2P2 ของไซเคิลเครื่องถัดไป และจะเป็น 0 ในช่วงเวลา S3P1 ถึง S5P2 ในไซเคิลของเครื่องเดียวกัน สัญญาณจากขา TXD ที่ส่งออกไปนี้ก็เพื่อให้อุปกรณ์ปลายทางสามารถรับข้อมูลได้ถูกต้องเพราะถูกส่งออกไปพร้อมกับข้อมูลใน SBUF ข้อมูลจำนวน 8 บิท จะถูกเลื่อนออกไปทางขา RXD จนครบทั้ง 8 บิท เมื่อข้อมูลบิทสุดท้ายถูกส่งออกไปจะทำให้ 1 ที่เกิดขึ้นจาก D-FF ในตอนเริ่มต้นการส่งข้อมูลถูกเลื่อนมาอยู่ทางขวาสุดของรีจิสเตอร์ SBUF และทางซ้ายทั้งหมดจะเป็น 0 ทำให้วงจร Zero Detector ซึ่งตรวจสอบค่า 0 นี้ส่งสัญญาณไปบอกวงจร Tx Control ว่าสิ้นสุดการส่งข้อมูลออก เมื่อสิ้นสุดการส่งข้อมูล สัญญาณ SEND จะเปลี่ยนสภาวะลอจิก 1 เป็น 0 ที่ขอบขาลงของสัญญาณ SHIFT และการส่งออกข้อมูลบิทสุดท้ายออกไปจะทำให้บิท T1 ในรีจิสเตอร์ SCON เปลี่ยนจาก 0 เป็น 1 บอกการสิ้นสุดของการส่งข้อมูล

การรับข้อมูล

การรับข้อมูลทางพอร์ทอนุกรมในโหมด 0 จะเริ่มต้นรับข้อมูลเข้ามาทางขา RXD ก็ต่อเมื่อมีการทำงานของคำสั่ง Set ค่าบิท REN เป็น 1 และ Clear บิท R1 เป็น 0 ในรีจิสเตอร์ SCON ซึ่งการทำงานของคำสั่งนี้ จะทำให้เริ่มรับข้อมูลที่เวลา S6P2 เสร็จแล้วจะหน่วงเวลาไปจนถึง S6P2 ในไซเคิลของเครื่องถัดไปก็จะทำให้สภาวะลอจิกของสัญญาณ Receive เปลี่ยนจาก 1 เป็น 0 เพื่อเริ่มส่งสัญญาณ Clock จากวงจร Shift Clock ถูกส่งออกไปทางขา TXD โดยสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

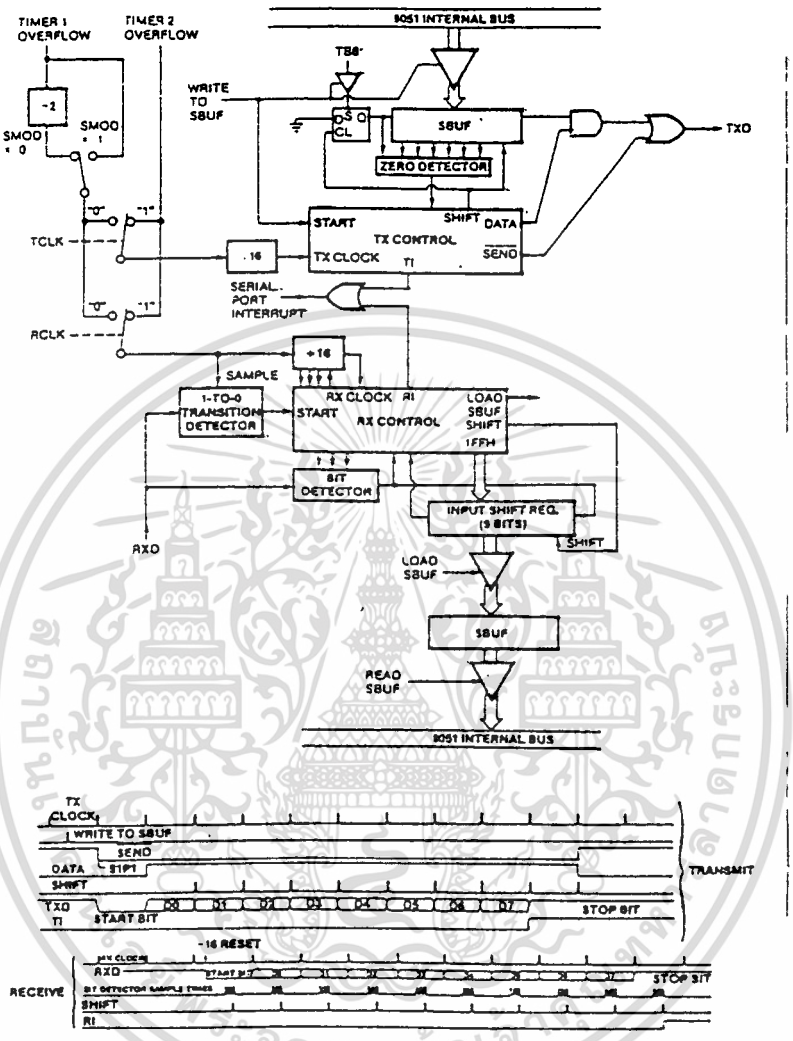
Clock ที่ขานี้จะมีสภาวะลอจิกเป็น 1 ตั้งแต่ S6P1 จนถึง S2P2 ของไซเคิลเครื่องถัดไปและมีสภาวะลอจิกเป็น 0 ในช่วงเวลา S3P1 ถึง S5P2 แต่ก่อนที่สัญญาณ Receive จะเปลี่ยนเป็น 1 นั้น วงจร RX Control จะเขียนข้อมูล 11111110H เข้าไปยัง Input Shift Register จากนั้นในทุก ๆ เวลา S6P2 ซึ่งสัญญาณ Shift มีสภาวะลอจิกเป็น 1 (นอกนั้นที่ เวลาอื่นจะมีลอจิกเป็น 0) จะเลื่อนข้อมูลใน Input Shift Register ไปทางซ้ายทีละ 1 บิท เมื่อครบ 7 ครั้ง ข้อมูล 0 ซึ่งอยู่ทางขวาสุดของรีจิสเตอร์ในตอนเริ่มต้นจะเลื่อนมาอยู่ที่ตำแหน่งซ้ายสุดและบอกให้กับ RX control ว่ามีข้อมูลบิทสุดท้ายอีก 1 บิทที่จะเข้ามา เมื่อข้อมูลสุดท้ายเข้ามาที่เวลา S5P2 ของไซเคิลของเครื่องที่ 9 (ข้อมูลที่เข้ามายังพอร์ทอนุกรมจะถูกอ่านเข้าไปที่เวลา S5P2 แต่จะถูกเก็บเข้าไปที่ Input shift register ที่เวลา S6) นับตั้งแต่เริ่มรับข้อมูลจะทำให้ที่เวลา S1P1 ของไซเคิลของเครื่องที่ 10 นับตั้งแต่เริ่มมีสัญญาณ Write to SCON (Clear R1) จะเกิดการส่งข้อมูลจาก Input Shift Register ไปเก็บยังรีจิสเตอร์ SBUF ต่อไปและในขณะเดียวกันก็จะทำให้บิท R1 ซึ่งถูก Clear ตั้งแต่เริ่มต้นรับข้อมูลถูก Set ให้เป็น 1 และจะทำให้สัญญาณ Receive เป็น 0 ทำให้ไม่มีสัญญาณ Clock ออกไปในโหมดนี้จะมีการส่งข้อมูลชุดละ 10 บิท คือ Start bit 1 บิท ข้อมูล 8 บิทและ Stop 1 บิท อัตราการส่งข้อมูลในโหมดนี้จะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 ข้อมูลนี้จะถูกส่งออก 1 บิท ทุก 16 หรือ 32 ครั้งของการเกิด Overflow ขึ้นใน Timer 1 การ Overflow คือ การที่ข้อมูลใน Timer Register มีค่าเพิ่มขึ้นจนถึงค่าสูงสุด เมื่อเพิ่มค่าไปอีกก็จะกลับเป็น 0 ใน 8052 จะสามารถกำหนดให้อัตราการส่งข้อมูลขึ้นกับอัตราการเกิด Overflow ของ Timer 2 ได้ดังในรูปที่ 2.66

การทำงานโหมด 1

การส่งข้อมูล

จากรูปที่ 2.66 บิท SMOD จะเป็นตัวเลือกว่าสัญญาณ Timer1 Overflow ที่ส่งไปยังวงจรหาร 16 จะถูกหาร 2 ก่อนหรือไม่ ถ้า SMOD เป็น 1 สัญญาณ Timer 1 Overflow จะไม่ถูกหาร แต่ถ้า SMOD เป็น 1 สัญญาณ Timer Overflow จะถูกหาร 2 ก่อน ที่จะเข้าวงจรหาร 16 การส่งข้อมูลจะเริ่มจากการที่มีคำสั่งเขียนข้อมูลไปยังรีจิสเตอร์ SBUF จะมีสัญญาณ Write to SBUF เกิดขึ้นเพื่อรับข้อมูลจาก Internal Bus ด้านบนไปเก็บยังรีจิสเตอร์ SBUF และทำให้เอาท์พุทของ D FLIP-FLOP ทางซ้ายของ SBUF มีค่าเป็น 1 และเป็นบิทที่ 9 ของการส่งข้อมูล สัญญาณ Write to SBUF ยังส่งไปยัง TX control ด้วย ขณะนี้ข้อมูลในวงจรหาร 16 มีค่าเป็นอะไรไม่ทราบจึงจะรอจนกว่าข้อมูลในวงจรหาร 16 นับเพิ่มขึ้นจนถึงค่าสูงสุดแล้ววนกลับเป็น 0 คือเกิดการวนกลับทำให้เริ่มการส่งข้อมูลที่เวลา S1P ของไซเคิลเครื่องถัดไป (การส่งข้อมูลออกจะสัมพันธ์กับการเกิด Overflow ในวงจรหาร 16) สัญญาณ SEND จาก TX Control เปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.66 Serial Port Mode 1

สภาวะลอจิกเป็น 0 แล้วเริ่มส่งข้อมูลที่เป็น Start bit (0) ออกไป เมื่อส่ง Strart Bit ออกไปแล้ว วงจร Tx Control ก็จะทำให้สัญญาณ DATA เป็น 1 เพื่อเลื่อนข้อมูลใน SBUF ออกไปเรื่อยจาก บิท 0 จนถึงบิทที่ 7 การส่งข้อมูลนี้จะเกิดขึ้นเมื่อสัญญาณ Tx Clock เปลี่ยนสถานะจาก 0 เป็น 1 ดังในรูปที่ 2.66 ขณะที่ข้อมูลถูกเลื่อนออกไปนั้นจะมี 0 ถูกเลื่อนเข้ามาทางซ้ายของรีจิสเตอร์ SBUF เมื่อข้อมูลเลื่อนออกไปทั้ง 8 บิทแล้วบิทที่ 9 ซึ่งเป็น 1 และตอนต้นอยู่ทางซ้ายสุดจะถูกเลื่อนมาอยู่ในตำแหน่งสุดท้ายทางขวาของรีจิสเตอร์ SBUF และทางซ้ายของหลักนี้จะมี 0 อยู่ทั้ง 8 บิทใน SBUF ทำให้ Zero Detector รู้ว่าเป็นข้อมูลบิทสุดท้ายแล้วที่ส่งออกโดยจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณมาบอกกับวงจร Tx Control ด้วยเมื่อ Tx Control ส่งสัญญาณ Shift ออกไปเป็นการส่งข้อมูลบิทสุดท้าย (บิท 7) ออกไปก็จะรออีก 1 TX Clock Bit clock ก็จะทำให้ขา TXD ส่งข้อมูล Stop Bit(1) ออกมา สัญญาณ DATA ซึ่งมีสภาวะลอจิกเป็น 1 มาตั้งแต่เริ่มส่งข้อมูลบิท 0 ก็จะกลับเป็น 0 และบิท T1 จะเป็น 1 เพื่อบอกการสิ้นสุดการส่งข้อมูลทั้งหมดจะสิ้นสุดลงเมื่อสัญญาณ TX Clock ไซเคิลที่ 10 นับตั้งแต่สัญญาณ SEND เปลี่ยนสภาวะลอจิกเป็น 0

การรับข้อมูล

การรับข้อมูลจะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 แล้วหาร 2 หรือไม่ขึ้นกับค่าของบิท SMOD สัญญาณนี้จะไปเข้าวงจรหาร 16 และเป็นตัวกำหนดอัตราการรับข้อมูล การรับข้อมูลจะเริ่มจากวงจร 1-T0-0 Transition Detector พบว่าสัญญาณที่ขา RXD เปลี่ยนจาก 1 เป็น 0 ซึ่งหมายถึงมีข้อมูล Start bit เข้ามาการตรวจสอบนี้จะกระทำด้วยอัตราเดียวกับสัญญาณที่เข้าวงจรหาร 16 เมื่อพบการเปลี่ยนสถานะลอจิกที่ขา RXD ก็จะเริ่มการรับข้อมูล ขณะนี้จะรีเซทวงจรหาร 16 ให้มีค่าเป็น 0 เพื่อสร้างสัญญาณ RX Clock ให้เข้าจังหวะ (Synchronous) กับข้อมูลที่เข้ามาโดยสัญญาณ RX Clock จะเป็น 1 เมื่อการนับของวงจรหาร 16 มีค่าเป็น 15 ขณะที่วงจรหาร 16 นับถึง 7, 8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD เพื่อเป็นการตรวจว่าข้อมูลนั้นเป็นอะไร ถ้าอย่างน้อยข้อมูล 2 ใน 3 เป็นค่าใดก็จะถือว่าข้อมูลที่เข้ามาเป็นค่านั้น ถ้าในการตรวจสอบ Start Bit แล้วพบว่าผิดพลาด คือไม่เป็น 0 ก็จะรีเซทการทำงานเพื่อไปตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ของข้อมูลที่ขา RXD ใหม่ แต่ถ้าพบ Start bit ก็จะเก็บข้อมูลทั้งหมดที่เข้ามาโดยเลื่อนข้อมูลเข้าไปยัง Input Shift Register ที่มีสัญญาณควบคุมการเลื่อนข้อมูล (Shift) ส่งมาจาก RX control ในตอนเริ่มต้นการรับข้อมูลจะมีการเขียนข้อมูล 1FFH ไปเก็บใน Input Shift Register ขณะที่ข้อมูลถูกเลื่อนเข้าไปทางขวาของ Input Shift Register ก็จะมี 1 ถูกเลื่อนออกไปทางซ้ายทุกครั้งที่มีข้อมูลเข้ามา เมื่อ Start bit ที่รับเข้ามาถูกเลื่อนไปถึงซ้ายสุดของ Input Shift Register ก็จะมีสัญญาณไปบอก RX Control Block หลังจากข้อมูลบิทสุดท้ายเข้ามาแล้วก็จะโหลด (Load) เอาข้อมูล 8 บิทไปเก็บในรีจิสเตอร์ SBUF พร้อมทั้ง Set ค่าในบิท RI และ RB8 ของรีจิสเตอร์ SCON แต่การโหลดข้อมูลไปเก็บนี้จะเกิดขึ้นได้ก็ต่อเมื่อ

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

ถ้าไม่มีสภาวะใดสภาวะหนึ่งดังกล่าวแล้ว ข้อมูลที่รับเข้ามาก็จะถูกทิ้งไปคือไม่โหลดไปเก็บในรีจิสเตอร์ SBUF ถ้ามีสภาวะดังกล่าวถูกต้อง stop bit จะถูกนำไปเก็บในรีจิสเตอร์ SBUF และบิท RI จะเป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ไม่ว่าทั้ง 2 กรณีจะเกิดหรือไม่ก็จะกลับไปสู่การตรวจสภาวะเปลี่ยนจาก 1 เป็น 0 ที่ขา RXD เพื่อรับข้อมูลต่อไป

ในการรับข้อมูลแบบอนุกรมโหมด 1 นี้ อัตราการส่งข้อมูลแต่ละบิท (Baud Rate) จะขึ้นอยู่กับอัตราการเกิด overflow ใน Timer 1 ดังสมการ

$$\text{Baud rate} = 2^{SMOD}/32 \times (\text{Timer 1 Overflow Rate})$$

ในขณะที่ใช้ Timer 1 เป็นตัวกำหนด Baud Rater นี้จะต้อง Disable ไม่ให้เกิดการขัดจังหวะเนื่องมาจากการ Overflow Timer 1 อาจใช้ในโหมดของ Timer หรือ Counter ก็ได้ ซึ่งเมื่อการนับในรีจิสเตอร์ตัวนับมีค่าสูงสุดแล้วกลับมาเป็น 0 ก็จะเกิด Overflow เช่นเดียวกัน แต่โดยปกติแล้วจะใช้ Timer 1 นี้ในโหมดของ Timer ที่มีการทำงานแบบ Auto Reload โหมด 2 เพื่อว่าเมื่อค่าในการนับโดยรีจิสเตอร์ TL1 ถึงค่าสูงสุดก็จะโหลดค่าในรีจิสเตอร์ TH1 มาไว้ใน TL1 สำหรับเป็นค่าเริ่มต้นการนับต่อไป ซึ่ง Baud rate จะมีค่า

$$\text{Baud rate} = \frac{2^{SMOD}}{32} \times \frac{Oscillator\ \ frequency}{12 \times [256 - (TH\ 1)]}$$

โดยที่ SMOD เป็นบิทหนึ่งในรีจิสเตอร์ PCON

เช่นความถี่ของออสซิลเลเตอร์เท่ากับ 11.059 MHz บิท SMOD = 0 รีจิสเตอร์ TH1 มีค่า E8H, Timer 1 ทำงานในโหมด 2 จะได้ว่าอัตราการส่ง-รับข้อมูลแบบอนุกรม

$$= \frac{1}{32} \times \frac{11.059 \times 10^{6}}{12 \times 24} \qquad = \frac{2^{0}}{32} \times \frac{11.059 \times 10^{6}}{12 \times [256 - 232]}$$

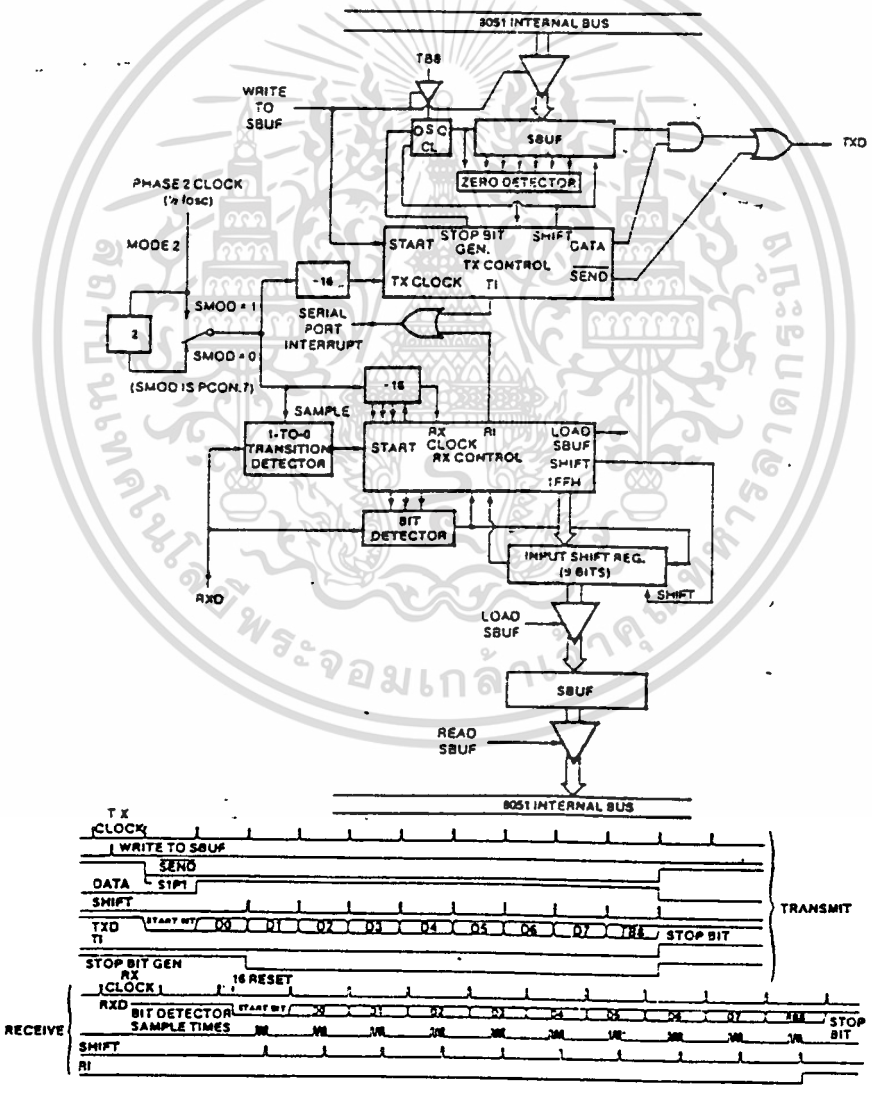
$$= \frac{1}{32} \times \frac{11.059 \times 10^{6}}{12 \times 24}$$

1200 บิท/วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

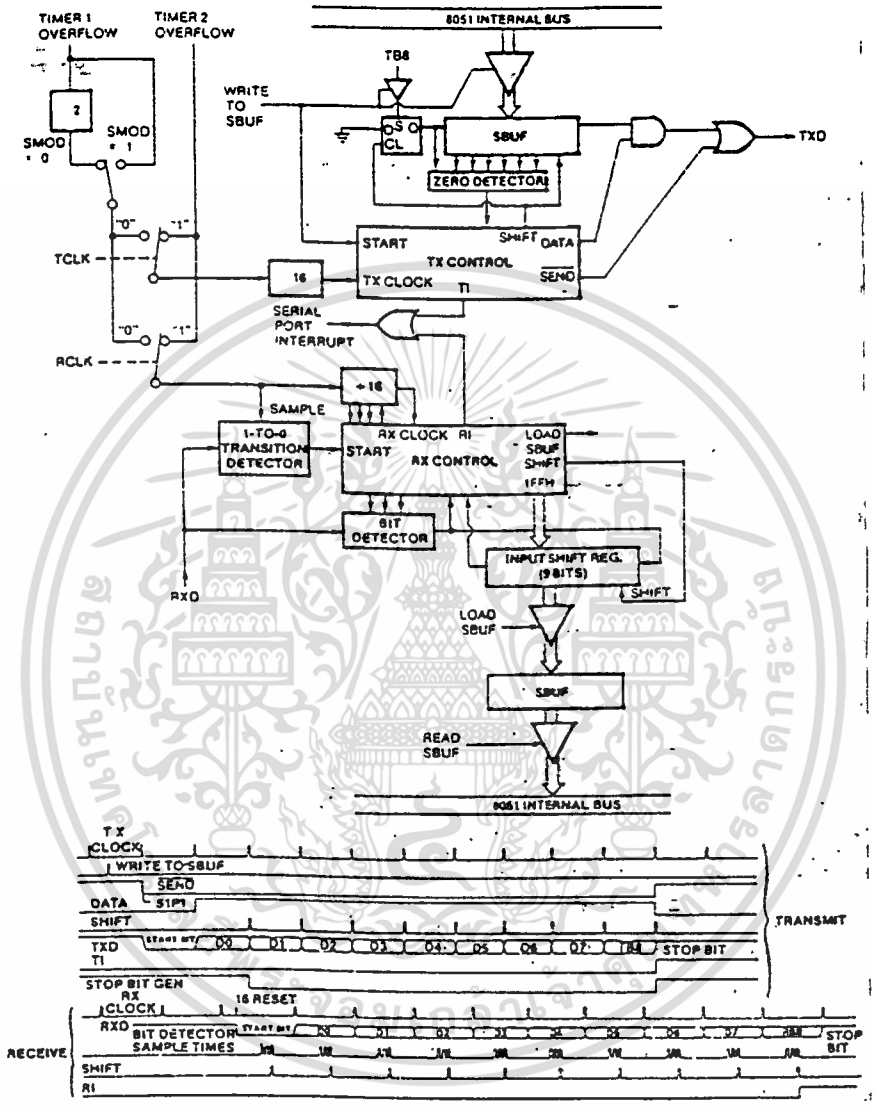
การทำงานโหมด 2 และโหมด 3

โหมด 2 และโหมด 3 ของการรับ-ส่งข้อมูลทางพอร์ทอนุกรมจะเหมือนกันแตกต่างเฉพาะตรงที่อัตราการส่งข้อมูลเท่านั้น ในโหมด 2 จะเลือกอัตราการรับ-ส่งได้เป็น 1/32 หรือ 1/64 เท่าของความถี่สัญญาณออสซิลเลเตอร์ ส่วนในโหมด 3 จะสามารถกำหนดอัตราการส่งข้อมูลได้โดยอัตราการส่งข้อมูลจะขึ้นกับอัตราการเกิด Overflow ของ Timer เช่นเดียวกับการรับ - ส่งข้อมูลโหมด 1



รูปที่ 2.67 Serial Port Mode 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.68 Serial Port Mode 3

ในรูปที่ 2.68 จะเห็นว่าสัญญาณจากเฟส 2ของสัญญาณนาฬิกาซึ่งมีความถี่เท่ากับ 1/2 ของสัญญาณออสซิลเลเตอร์จะถูกเลือกโดยสวิทช์ SMOD ว่าจะให้หาร 2 หรือไม่ สวิทช์ SMOD นี้จะสามารถกำหนดตำแหน่งให้อยู่ข้างบนหรือข้างล่างโดยการกำหนดค่าของบิท SMOD ใน รีจิสเตอร์ PCON ให้เป็น 1 หรือ 0 ตามลำดับ สัญญาณที่ออกจากวงจรหาร 16 จะมีความถี่ เป็น 1/32 หรือ 1/64 เท่าของสัญญาณจากออสซิลเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ Timer 1 Overflow อันเกิดจากการนับถึงค่าสูงสุดใน Timer1 แล้วค่าการนับจะกลับเป็น 0 ใหม่ สัญญาณนี้จะถูกเลือกโดยสวิทช์ SMOD ให้หาร 2 หรือไม่ ก่อนที่จะส่งไปยังวงจรหาร 16 เพื่อสร้างเป็นสัญญาณควบคุมการเลื่อนข้อมูลเข้าหรือออกทางพอร์ทอนุกรมต่อไป การเลือกตำแหน่งของสวิทช์ SMOD ก็โดยการกำหนดค่าในบิท SMOD ของรีจิสเตอร์ PCON ถ้าบิทนี้เป็น 1 สัญญาณจะถูกหาร 2 ก่อนแล้วจึงส่งเข้าไปยังวงจรหาร 16 การกำหนดอัตราการส่งของข้อมูลจะทำได้เหมือนกับในการรับ-ส่งข้อมูลพอร์ทอนุกรมโหมด 1 ที่กล่าวมาแล้วทุกประการ

การรับ-ส่งข้อมูลในโหมด 2 และ 3 นั้น 1 ชุดของข้อมูลจะประกอบด้วย 1 Start Bit, 9 Data Bit 1 Stop Bit ทั้งหมดนี้จะส่งออกหรือรับเข้าทางพอร์ทอนุกรมทีละบิทเรียงตามลำดับ จากรูปที่ 2.67 และ 2.68 จะสามารถอธิบายการทำงานภายใน 8051 ในการส่งหรือรับข้อมูลได้ดังนี้

การส่งข้อมูล

การส่งข้อมูลจะเริ่มจากการมีการเขียน (Write) ข้อมูลลงไปยังรีจิสเตอร์ SBUF ทำให้มีสัญญาณ Write to SBUF เกิดขึ้น แล้วข้อมูล 8 บิทจาก Internal Data ของ 8051 ถูกเขียนลงไปในรีจิสเตอร์ SBUF และข้อมูลจากบิท TB8 ของรีจิสเตอร์ SCON จะเขียนลงไปยัง D FLIP FLOP เช่นกัน รวมเป็น 9 บิท ดังรูป ที่เวลา S1P1 ของไซเคิลเครื่องแรกหลังจากสัญญาณ TX Clock นับครบ 1 รอบจะทำให้สัญญาณ SEND ที่ออกจากวงจร TX Control เปลี่ยนสภาวะลอจิกเป็น 0 เพื่อเริ่มการส่งข้อมูล โดยจะเริ่มส่ง Star Bit (0) ออกไปทางขา TXD เมื่อสัญญาณ TX Clock เปลี่ยนเป็น 1 ครั้งต่อไปก็จะเริ่มส่งข้อมูลบิท 0 ออกไปและสัญญาณ DATA จะเป็น 1 ที่เวลานี้ด้วย TX Control จะส่งสัญญาณ Shift ไปยังรีจิสเตอร์ SBUF ทุกครั้งที่สัญญาณ TX Clock เป็น 1 เพื่อเลื่อนข้อมูลออกไปทางขวา และจะเลื่อน 0 เข้ามาทางซ้ายขณะที่บิท TB8 อยู่ทางขวาสุดเตรียมส่งออกนั้น ข้อมูลทางซ้ายทุกบิทจะเป็น 0 วงจร Zero Detector จะมีสัญญาณไปบอก TX Control ในการส่งข้อมูลบิทสุดท้าย คือ TB8 ออกไปตามเวลาของสัญญาณ TX Clock (Bit time) เมื่อส่งข้อมูลบิท TB8 ออกไปเสร็จสิ้นแล้วก็จะรอเวลา 1 Bit Time สัญญาณ SEND ก็จะกลับเป็น 1 และบิท T1 ในรีจิสเตอร์ SCON จะเป็น 1 เพื่อบอกสิ้นสุดการส่งข้อมูลและสัญญาณ DATA ก็จะกลับเป็น 0 จากนั้นสัญญาณที่ออกจาก TXD ก็คือ Stop Bit นั่นเอง

การรับข้อมูล

การรับข้อมูลที่เข้ามาทางขา RXD ของ 8051 จะเริ่มต้นเมื่อวงจร 1-to-0 Transition Detector ทำการตรวจสอบข้อมูลที่ขา RXD 16 เท่าของอัตราการส่งข้อมูลแล้วพบว่าสัญญาณที่ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RXD เปลี่ยนจาก 1 เป็น 0 ทำให้วงจรหาร 16 ถูกรีเซทไปด้วยและจะเขียนข้อมูลในรีจิสเตอร์ SBUF เป็น 1FFH (มี 9 บิท) เมื่อ Counter ในวงจรหาร 16 นับถึง 7, 8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD ถ้า 2 ใน 3 เหมือนกันก็ถือว่าข้อมูลที่เข้ามาคือค่านั้น ถ้าข้อมูลที่รับเข้ามาบิทแรกไม่เป็น 0 คือไม่ใช่ Start Bit (เพราะอาจตรวจพบการเปลี่ยนจากสัญญาณจากขา RXD แต่การตรวจสอบที่เวลา 7, 8 และ 9 ส่วนใหญ่เป็น 1) ก็จะเกิดการรีเซทส่วนรับข้อมูลเข้ามาทีละ 1 บิทข้อมูลนี้จะถูกเลื่อนเข้าไปเก็บทางขวาของ input Shift Register และ 1 จะถูกเลื่อนออกไปทางซ้ายโดยสัญญาณ Shift จนกระทั่ง สัญญาณ Start Bit ซึ่งเป็น 0 ถูกเลื่อนเข้ามาถึงซ้ายสุดของ Input Shift Register จะบอกให้กับ RX Control รู้ว่าจะต้องมีข้อมูลบิทสุดท้ายเข้ามาอีก 1 บิท เมื่อข้อมูลบิทสุดท้ายเข้ามาจะเกิดการโหลดข้อมูลจาก Input Shift Register ไปยัง SBUF, RB8 และเซต RI ให้เป็น 1 แต่สิ่งนี้จะเกิดขึ้นได้ต้องมีสภาวะดังนี้ด้วย

1. บิท RI เป็น 0 เสมอ
2. SM2 = 0 หรือถ้า SM2 = 1 ข้อมูลบิทที่ 9 จะต้องเป็น 1

ถ้าไม่มีสภาวะดังกล่าวทั้ง 2 ข้อมูลที่รับมาจะถูกทิ้งไปไม่ถูกนำไปเก็บที่ SBUF และ RI ก็จะไม่ถูก Set แต่ถ้าเกิดขึ้นทั้ง 2 กรณี ข้อมูลจะถูกนำไปเก็บที่รีจิสเตอร์ SBUF 8 บิท และบิทสุดท้ายจะนำไปเก็บที่บิท RB8 ของรีจิสเตอร์ SCON หลังจากนั้น 1 Bit Time ไม่ว่าจะมีการเก็บข้อมูลหรือ ก็จะเข้าสู่การตรวจสอบสถานะการเปลี่ยนสัญญาณจาก 1 เป็น 0 เพื่อเตรียมรับข้อมูลต่อไป

2.6.10 ลักษณะสัญญาณที่ใช้ในการอินเตอร์เฟสตามมาตรฐานต่าง ๆ

ปกติการอินเตอร์เฟสที่ออกแบบ และจัดให้ใช้ควบคุมและตรวจสอบ (sense) ข้อมูลจะเป็นการอินเตอร์เฟสที่ใช้สัญญาณดิจิตอล เป็นหลักสัญญาณพวกนี้จะถูกส่งและรับมาจากรีจิสเตอร์อินพุท/เอาท์พุท, อินพุทของ interrupt-request, พอร์ทที่ใช้ในการทำ DMA และไทม์เมอร์/เคาน์เตอร์ การอินเตอร์เฟสที่กล่าวมานี้ทุกตัวเป็นการใช้สัญญาณดิจิตอลที่มีระดับสัญญาณแบบ TTL ในการอินเตอร์เฟส ดังนั้นถ้าเรานำอุปกรณ์อื่นที่ใช้ระดับสัญญาณ TTL ในการอินเตอร์เฟสด้วยเช่นกัน การทำการอินเตอร์เฟสเข้ากับเครื่อง PC จะทำได้โดยตรง แต่ในหลายกรณีสัญญาณที่ใช้ไม่ได้เป็นระดับสัญญาณ TTL หรือไม่ได้เป็นสัญญาณดิจิตอล และระยะทางที่ทำการอินเตอร์เฟสก็มีระยะห่างมากซึ่งก่อให้เกิดปัญหาหลายประการ เราสามารถแก้ปัญหาเหล่านี้ได้โดยการใช้การอินเตอร์เฟสแบบต่าง ๆ ที่กล่าวไว้ในบทนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเตอร์เฟสตามมาตรฐาน RS-232C

มาตรฐาน RS-323C เป็นมาตรฐานที่ได้รับการพัฒนามานานแล้วและถูกใช้งานกันอย่างแพร่หลาย เราใช้ RS-232C เชื่อมต่อ DTE (Data Terminal Equpment) เข้ากับ DCE (Data Communication Equipment) เช่น การต่อเทอร์มินัลเข้ากับไมเด็ม มาตรฐาน RS-232C กล่าวถึงลักษณะทางกล, ลักษณะของสัญญาณไฟฟ้าและลักษณะการทำงานที่ใช้ในการอินเตอร์เฟส ตัวอย่างของอุปกรณ์ที่ใช้ในการอินเตอร์เฟสตามมาตรฐาน RS-232C ได้แก่ เทอร์มินัล, พล็อตเตอร์ (plotter), ลอจิกอนาไลเซอร์ (logic analyzer) และเครื่องพิมพ์ ถ้าการประยุกต์ใช้งานของเราต้องการอินเทอร์เฟสอุปกรณ์เข้ากับการอินเตอร์เฟสตามมาตรฐาน RS-232C เราจำเป็นต้องแปลงระดับสัญญาณ TTL ให้เป็นระดับสัญญาณแบบอื่น ซึ่งรายละเอียดของระดับสัญญาณที่ใช้ได้กล่าวไว้ในมาตรฐาน RS-232C สำหรับตัวอย่างของการใช้งานขอให้ดูจากการใช้ IBM PC serial-interface feature card ไดอะแกรมของวงจรและลักษณะการทำงานแสดงไว้ในหนังสือ "IBM System-Reference Manual"

ลักษณะสัญญาณที่ใช้ในการอินเตอร์เฟส

มาตรฐาน RS-232C ใช้สายสัญญาณเพียงเส้นเดียวในการส่งสัญญาณโดยสัญญาณที่ส่งไปได้ทิศทางเดียว ในกรณีที่อัตราเร็วในการส่งข้อมูลมีค่าเท่ากับ 20 K bps (กิโลบิทต่อวินาที) ซึ่งค่านี้เป็นค่าสูงสุดที่ใช้ในการส่งข้อมูล ระยะทางที่ใช้ส่งข้อมูลไม่ควรเกิน 50 ฟุต (ตามข้อกำหนดในมาตรฐาน) สำหรับการแทนแรงดันของระดับสัญญาณ (ในบทนี้เราแทนระดับสัญญาณด้วยลอจิกบวก) มีข้อกำหนดดังนี้ "1" แทนระดับแรงดันที่มีค่าระหว่าง +5 โวลท์ ถึง +15 โวลท์ "0" แทนระดับแรงดันที่มีค่าระหว่าง -5 โวลท์ ถึง -15 โวลท์

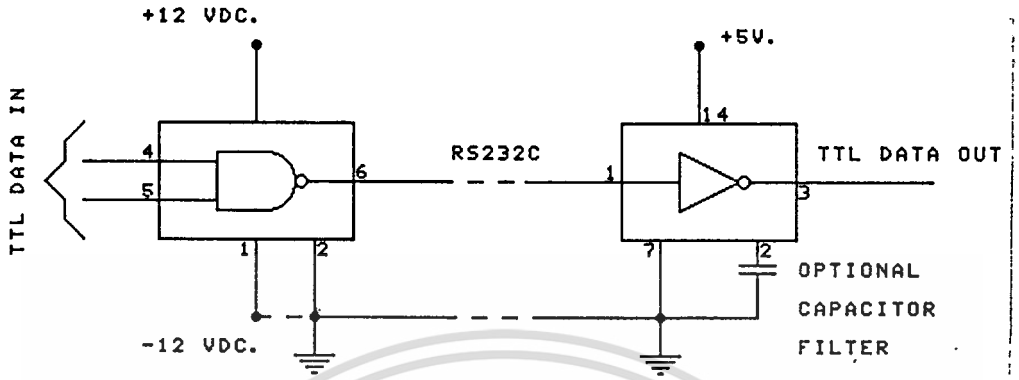
วงจรแปลงระดับแรงดัน

ตัวอย่างของวงจรที่ใช้แปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณที่กำหนดไว้ในมาตรฐาน RS-232C และแปลงกลับจากระดับแรงดันในมาตรฐาน RS-232C ไปเป็นระดับสัญญาณ TTL แสดงไว้ในรูปที่ 2.69

ลักษณะการอินเตอร์เฟสที่ใช้มาตรฐาน RS-423

มาตรฐาน RS-423 เป็นมาตรฐานที่ได้รับการพัฒนามาจากมาตรฐาน RS-232C อุปกรณ์ที่ผลิตขึ้นมาใหม่ ๆ มักจะใช้ในการอินเตอร์เฟสแบบนี้ โดยเฉพาะอุปกรณ์ที่ต้องการให้อัตราเร็วในการส่งข้อมูลมีค่าสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.69 วงจรขับสัญญาณและรับสัญญาณที่ใช้กับมาตรฐาน RS-232-C

ลักษณะสัญญาณที่ใช้ในการอินเตอร์เฟส

มาตรฐาน RS-423 ใช้สายสัญญาณเส้นเดียวในการส่งสัญญาณ โดยสัญญาณที่ส่ง ๆ ไปได้ทิศทางเดียว อัตราเร็วในการส่งข้อมูลมีค่าสูงถึง 100 K bps ที่ระยะห่าง 40 ฟุต ตัวรับข้อมูลเป็นแบบ balanceed-line ดังนั้นตัวรับข้อมูล (receiver) จึงรับข้อมูลแบบขยายความแตกต่างของสัญญาณระหว่างสายกราวด์กับตัวขับสัญญาณ (driver) การทำเช่นนี้ ช่วยแก้ปัญหาในกรณีที่เกิดความแตกต่างระหว่างแรงดันที่กราวด์ของตัวรับข้อมูลกับตัวขับสัญญาณ สำหรับการแทนระดับแรงดันนั้น ลอจิก "1" แทนระดับแรงดันที่อยู่ระหว่าง +4 โวลท์ ถึง +6 โวลท์ ส่วนลอจิก "0" แทนระดับแรงดันที่มีค่าระหว่าง -4 โวลท์ ถึง -6 โวลท์

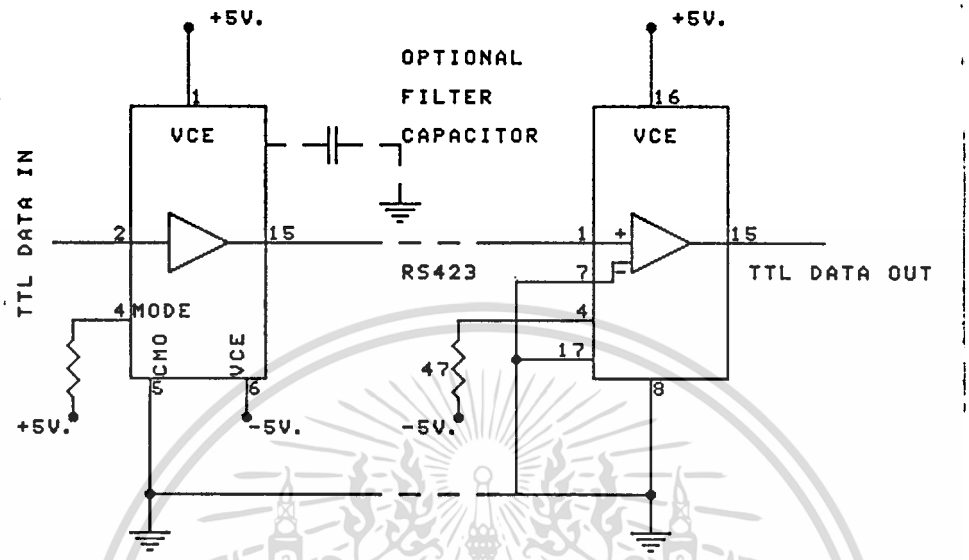
วงจรแปลงระดับแรงดันของมาตรฐาน RS-423

วงจรที่ใช้แปลงระดับแรงดันจากระดับสัญญาณ TTL ไปเป็นระดับสัญญาณที่ใช้ในมาตรฐาน RS-423 และแปลงระดับสัญญาณจาก RS-423 กลับไปเป็นระดับสัญญาณ TTL แสดงไว้ในรูปที่ 2.70

ลักษณะการอินเตอร์เฟสที่ใช้มาตรฐาน RS - 422

มาตรฐาน RS-422 ได้พัฒนามาจากมาตรฐาน RS-423 ทำให้อัตราเร็วในการส่งข้อมูลมีค่าสูงขึ้นและระยะทางที่ใช้ส่งข้อมูลระหว่างตัวส่งและตัวรับมีระยะทางไกลขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.70 วงจรขับ และ รับสัญญาณที่ใช้กับมาตรฐาน RS - 423

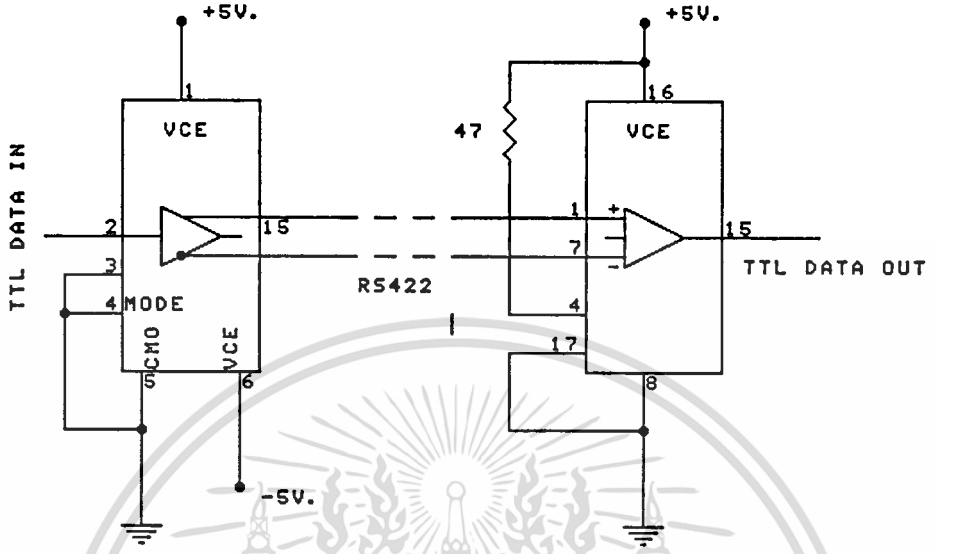
ลักษณะสัญญาณที่ใช้ในการอินเตอร์เฟส

มาตรฐาน RS-422 ใช้การส่งข้อมูลในลักษณะของ one-war belanced-line โดยอัตราเร็วในการส่งข้อมูลมีค่าสูงถึง 10M bps ที่ระยะห่างเท่ากับ 1000 ฟุต ในกรณีที่ส่งข้อมูลในอัตราเร็วที่ต่ำกว่า 10M bps ระยะห่างที่ใช้ในการส่งข้อมูลสามารถขยายได้ถึง 4000 ฟุต ระดับแรงดันที่ส่งจากตัวขับสัญญาณจะมีค่าระหว่าง +2 โวลท์ ถึง +6 โวลท์ นอกจากนี้ตัวขับสัญญาณสามารถจับสัญญาณที่มีระดับต่ำถึง 200 mV ได้

วงจรแปลงระดับแรงดันที่ใช้ในมาตรฐาน RS - 422

วงจรที่ใช้แปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณที่ใช้ในมาตรฐาน RS-422 และแปลงจาก RS-422 กลับไปเป็นระดับสัญญาณ TTL นอกจากนี้ผู้ผลิตบางบริษัทได้ทำวงจรขับสัญญาณเป็นแบบ tri-state ทำให้เราสามารถส่งข้อมูลได้สองทิศทางบนสายคู่เดียว (single pair) คุณสมบัติข้อนี้ทำให้เราสามารถใช้มาตรฐาน RS-422 ในเน็ทเวิร์คที่มีโครงสร้างเป็นแบบ multidrop ซึ่งอุปกรณ์หลาย ๆ ตัวสามารถรับและส่งข้อมูลแบบ half duplex บนสายคู่เดียวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.71 วงจรขับ และ รับสัญญาณที่ใช้กับมาตรฐาน RS - 422

การส่งข้อมูลโดยใช้การอินเตอร์เฟสแบบ current-loop

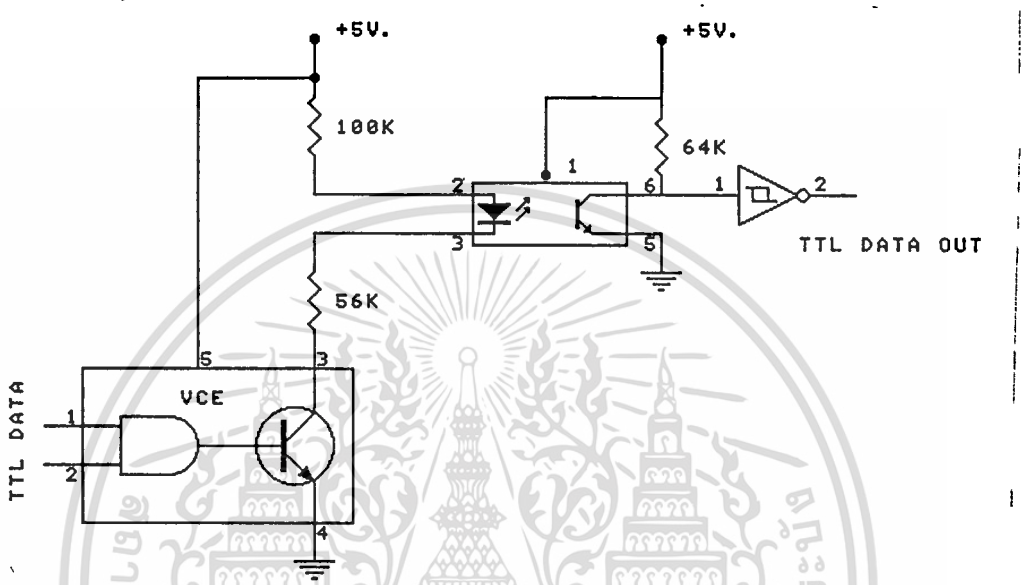
การอินเตอร์เฟสแบบ current-loop มีข้อดีคือสามารถส่งข้อมูลได้ในระยะทางไกล ๆ และค่าใช้จ่ายในการอินเตอร์เฟสแบบนี้มีราคาไม่สูงนัก หลักการของการอินเตอร์เฟสแบบนี้มีดังนี้ เมื่อลูปถูกวงจร (closed-loop) ระดับแรงดันถูกเปลี่ยนเป็นกระแสตามสมการ V = IR เมื่อลูปถูกปิดวงจรจะไม่มีกระแสไหลในลูป เนื่องจากวงจรของอินเตอร์เฟสแบบนี้มีค่าอิมพีแดนซ์ต่ำดังนั้นจึงทนต่อสัญญาณรบกวนได้ดี เรามักใช้วงจรของการอินเตอร์เฟสแบบนี้ในกรณีที่ต้องเดินสายผ่านบริเวณที่มีสัญญาณรบกวนมาก ๆ นอกจากนี้เราสามารถใช้การอินเทอร์เฟสแบบนี้แยกกราวด์ของสองระบบออกจากกัน วงจรนี้สามารถใช้ส่งข้อมูลได้ในอัตราเร็ว 50 K bps โดยระยะทางที่ใช้ส่งข้อมูลมีค่าได้ไม่เกิน 3000 ฟุต ตัวจำกัดระยะทางที่ใช้ในการส่งข้อมูลคือ ความต้านทานของสายที่ประกอบกันเป็นลูป ซึ่งไม่ควรมีค่าเกิน 30 โอห์ม (สำหรับวงจรในรูป 2.72)

การดักจับสัญญาณที่ส่งจากสวิทช์ (SWITCH SENSING)

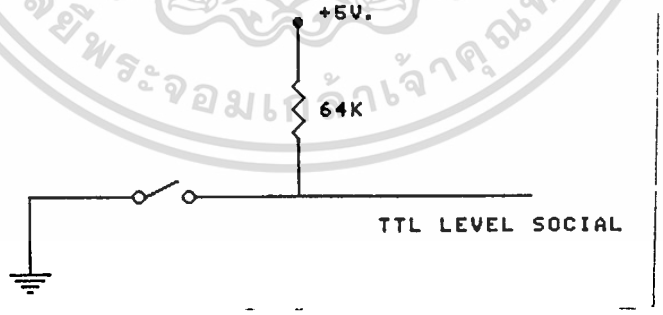
บ่อยครั้งทีเดียวการใช้สวิทช์เป็นตัวระบุสภาวะการทำงานของระบบสามารถใช้ประโยชน์ได้อย่างกว้างขวาง เช่น ใช้ในการควบคุมระยะในการเคลื่อนที่ (limit switch) เราสามารถสร้างสวิทช์ที่สามารถส่งสัญญาณในระดับสัญญาณ TTL ไปยังพอร์ทอินพุทที่ใช้กับระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ TTL ได้ ดังวงจรในรูปที่ 2.73 เมื่อสวิทช์เปิดวงจร ระดับแรงดัน +5 โวลท์จะถูกส่งไปยังวงจรขับสัญญาณเมื่อสวิทช์ปิดวงจร ระดับแรงดันที่ส่งไปจะเป็นแรงดันที่กราวด์แทน



รูปที่ 2.72 วงจรที่ใช้การอินเตอร์เฟสแบบ Current - Loop

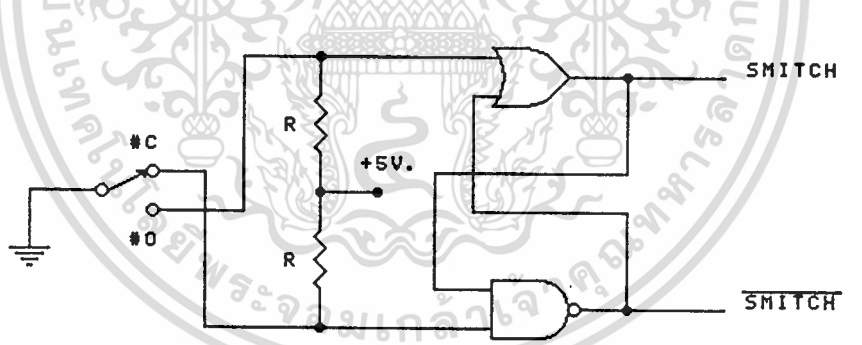


รูปที่ 2.73 ลักษณะสวิทช์ที่ใช้ส่งระดับสัญญาณ TTL

ปัญหาที่เกิดขึ้นมากในการอินเตอร์เฟสโดยใช้สวิทช์คือ switch bounce เมื่อสวิทช์ทางกล (mechanical switch) ถูกเปิดหรือปิดจะมีการ bounce ที่หน้าคอนแท็กขึ้น การ bounce นี้ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดสัญญาณรบกวนขึ้นซึ่งสัญญาณรบกวนนี้จะถูกส่งไปยังวงจรรับสัญญาณทำให้วงจรรับสัญ - ญาณรับข้อมูลในลักษณะที่ว่าสวิทช์ถูกเปิดและปิดอย่างรวดเร็วหลาย ๆ ครั้ง การแก้ปัญหาข้อนี้สามารถทำได้หลายวิธี ๆ ที่ใช้โดยทั่วไปคือ ทำการ sampling สภาพสัญญาณอย่างรวดเร็วหลาย ๆ ครั้ง การตรวจสอบนี้ทำเพื่อยืนยันว่า สภาพของสัญญาณเข้าสู่สภาวะคงตัวแล้วหรือยัง การตรวจสอบแบบนี้มักใช้กับระบบที่สามารถสูญเสียเวลาไปกับซอฟท์แวร์เพื่อการใช้การตรวจสอบได้ สำหรับอีกวิธีหนึ่งเป็นการใช้ฮาร์ดแวร์ในการแก้ไขการ bounce ของสวิทช์ดังแสดงในรูป 2.74 สวิทช์ประกอบด้วยคอนแท็กต่อไปนี้ normally open (NO) และ normally closed (NC) เมื่อคอนแท็ก NO ถูกต่อลงกราวนด์ bounce ลูกแรกที่เกิดขึ้นจะเป็นตัวเซ็ทวงจรแลทช์ ดังนั้น bounce ลูกต่อ ๆ ไปจะไม่ทำให้เอาท์พุทของแลทช์เปลี่ยนแปลงไป เช่นเดียวกันเมื่อคอนแท็ก NC ถูกต่อกราวนด์ซึ่งก็คือการรีเซทแลทช์ bounce ลูกแรกจะเป็นตัวรีเซทแลทช์ทำให้เอาท์พุทของแลทช์มีค่าคงที่ วงจรในรูปที่ 2.74 จะทำงานได้เฉพาะกับสวิทช์ที่ประกอบด้วยคอนแท็กแบบ normally oper และ normally closed ซึ่งการสับสวิทช์จาก NC ไปเป็น NO จะช่วยทำให้มีการหน่วงเวลาเกิดขึ้นระยะหนึ่ง ดังแสดงไว้ในไทม์มิ่งไดอะแกรมของรูปที่ 2.74



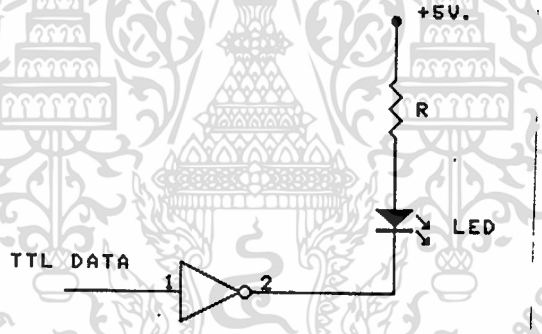
รูปที่ 2.74 วงจรแก้การ Bounce ของสวิทช์

INDICATOR DRIVER

ในบางครั้งถ้าระบบสามารถแสดงตำแหน่งของสภาวะการทำงานของมันต่อผู้ใช้งานได้จะช่วยให้การตรวจสอบและแก้ไขระบบสามารถทำได้ง่ายและรวดเร็ว ตัวอย่างของอุปกรณ์ชิ้นหนึ่งที่ใช้ในการแสดงสภาวะการทำงานของระบบคือ indicator ตัวอย่างที่จะอธิบายต่อไปเป็นตัวอย่างของการใช้ระดับสัญญาณ TTL เป็นตัวขับ indicator แบบต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรามักใช้ LED (Light-Emitting-Diode) แสดงสภาวะของระบบในกรณีที่ระดับความสว่างของ indicator ไม่ต้องสว่างมากนัก ผู้ผลิตบางบริษัทได้สร้าง LED ที่ถูกขับโดยตรงจากไอซี TTL แบบมาตรฐาน (standard TTL) หรือแบบ S(Schottky) ซึ่งสำหรับแบบ LS (Low-Power Schottky) มักไม่มีกำลังพอที่จะขับ LED ได้ นอกจากนี้เราต้องพิจารณาให้ดีก่อนที่จะต่อ LED เข้ากับอุปกรณ์พวก NMOS LS1 ไม่มีกำลังพอในการขับ LED อนึ่งถ้า LED ถูกต่อกับเอาท์พุทของอุปกรณ์พวก TTL เอาท์พุทนั้นไม่ควรนำไปใช้ขับวงจรอื่นอีกเพราะ LED สามารถ drop ระดับแรงดันเอาท์พุทของอุปกรณ์ TTL ทำให้ระดับลอจิกเปลี่ยนแปลงไป ซึ่งเป็นสาเหตุทำให้การทำงานของวงจรเกิดผิดพลาดขึ้น ด้วยเหตุนี้สัญญาณทุกตัวที่ถูก indicator ใช้ในการแสดงผลจะต้องถูกบัฟเฟอร์ไว้ก่อนด้วยวงจรขับสัญญาณ สำหรับการใช้ไอซี TTL ในการขับ LED ได้แสดงไว้ในรูป 2.75

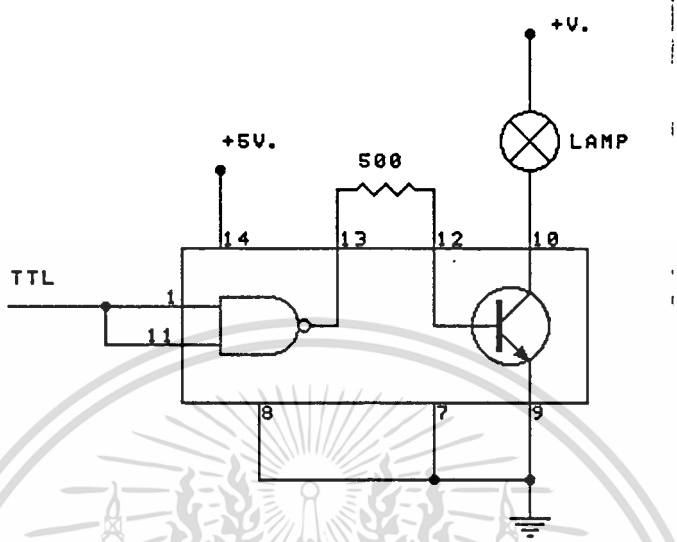


รูปที่ 2.75 วงจรขับ LED

เมื่อเราต้องการใช้ indicator ที่ให้ความสว่างมากกว่า LED เราสามารถใช้หลอด incancescent แทนได้ แต่หลอดแบบนี้กินกระแสในการทำงานสูงกว่า LED มาก ดังนั้นวงจรขับกระแสต้องมีกำลังพอที่จะจ่ายกระแสให้แก่หลอด incandescent ได้ สำหรับวงจรที่ใช้ในการขับหลอด incandescent ด้วยสัญญาณ TTL แสดงไว้ในรูปที่ 2.75

ในกรณีที่เราต้องการแสดงข้อมูลเป็นตัวเลข หรือคาร์แรกเตอร์ในเลขฐานสิบหกอุปกรณ์แสดงผลจำเป็นต้องซับซ้อนขึ้น บริษัทผู้ผลิตได้ผลิตอุปกรณ์แสดงผลแบบ dip-type ซึ่งสามารถขับได้โดยตรงจากอุปกรณ์ TTL รวมทั้งการถอดรหัสเพื่อแสดงผลเป็นตัวเลขฐานสิบหรือฐานสิบหก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.76 วงจรขับหลอดแสดงผลที่กินกระแสมาก

RELAY DRIVING

ในการทำการอินเตอร์เฟสร่วมกับรีเลย์ หรือขดโซลินอยด์ เพื่อใช้ในการควบคุมอุปกรณ์ที่ต้องการกำลังสูง ๆ เราต้องใช้อุปกรณ์พิเศษบางตัวเพิ่มเข้าไป อุปกรณ์พวกนี้ไม่สามารถขับได้โดยตรงจากอุปกรณ์ TTL เนื่องจากกระแสที่จ่ายจากอุปกรณ์ TTL ไม่มีกำลังพอที่จะทำให้รีเลย์ หรือขดลวดโซลินอยด์ทำงานได้ นอกจากนี้อุปกรณ์ที่ใช้ในการขับรีเลย์หรือขดโซลินอยด์ต้องเพิ่มอุปกรณ์ที่ใช้เป็นตัวป้องกันการเกิด inductive kickback ที่เกิดขึ้นในรีเลย์หรือขดลวดโซลินอยด์ ตัวอย่างของอุปกรณ์ที่ใช้ในการอินเตอร์เฟสแบบนี้ได้แก่ ไอซีเบอร์ SN75475 dual perripheral-driver ซึ่งผลิตโดยบริษัท Texas Instruments Incorported อุปกรณ์ตัวนี้สามารถจ่ายกระแสได้ถึง 300 mA และใช้ได้กับระดับแรงดันที่สูงถึง 100 โวลท์ นอกจากนี้มักสามารถถูกขับจากวงจร TTL แบบมาตรฐาน (standard TTL circuit) และในตัวชิพไอซีมีไดโอดที่ใช้ในการลด transient ที่เกิดจาก inductive load รวมไว้ด้วย

2.7 วงจร D/A

วงจร D/A มีด้วยกันหลายแบบด้วยกัน เราคงจะไม่สามารถพูดอธิบายได้ทุกแบบ ในที่นี้จะขอกล่าวเฉพาะวงจร D/A แบบรวมกระแสกับแบบวงจรบันไดของตัวต้านทาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 วงจร D/A แบบรวมกระแส

วงจรในรูปที่ 2.77 เป็นหลักการของวงจร D/A แบบรวมกระแส (Current Summing Type) นี้ เราจะใช้สวิทช์ 4 ตัว ใช้ตัวต้านทาน 5 กิโลโอห์ม จำนวน 15 ตัว และมิเตอร์วัดกระแสพิสัยไม่เกิน 20 มิลลิแอมป์ สวิทช์ทั้ง 4 ตัว นี้คือ SW_A,SW_B, ซึ่งเปรียบเสมือนปริมาณดิจิตอลแต่ละบิท เมื่อปิดสวิทช์หมายถึง "1" เปิดหมายถึง "0" เมื่อสวิทช์ทั้งสี่แทนเลขฐานสอง SW_A จะเป็นบิทนัยสำคัญต่ำสุด (บิทต่ำสุดทางขวามือ) และ SW_D จะเป็นบิทนัยสำคัญสูงสุด

เมื่อบิทสวิทช์แต่ละตัวจะมีกระแสขนาดต่างกันไหลมาที่มิเตอร์วัดกระแส เช่น บิทสวิทช์ SW_A จะมีกระแสไหล 1 มิลลิแอมป์ ปิดสวิทช์ SW_B จะมีกระแสไหล 2 มิลลิแอมป์ เป็นต้น ถ้าปิดสวิทช์หลายสวิทช์พร้อมกันจะมีกระแสไหลจากแต่ละกิ่งมารวมกันแล้วไหลไปยังมิเตอร์ กระแสที่ไหลมาที่มิเตอร์นี้คือปริมาณอนาลอกที่แปลงมาได้จากวงจรนี้นั่นเอง

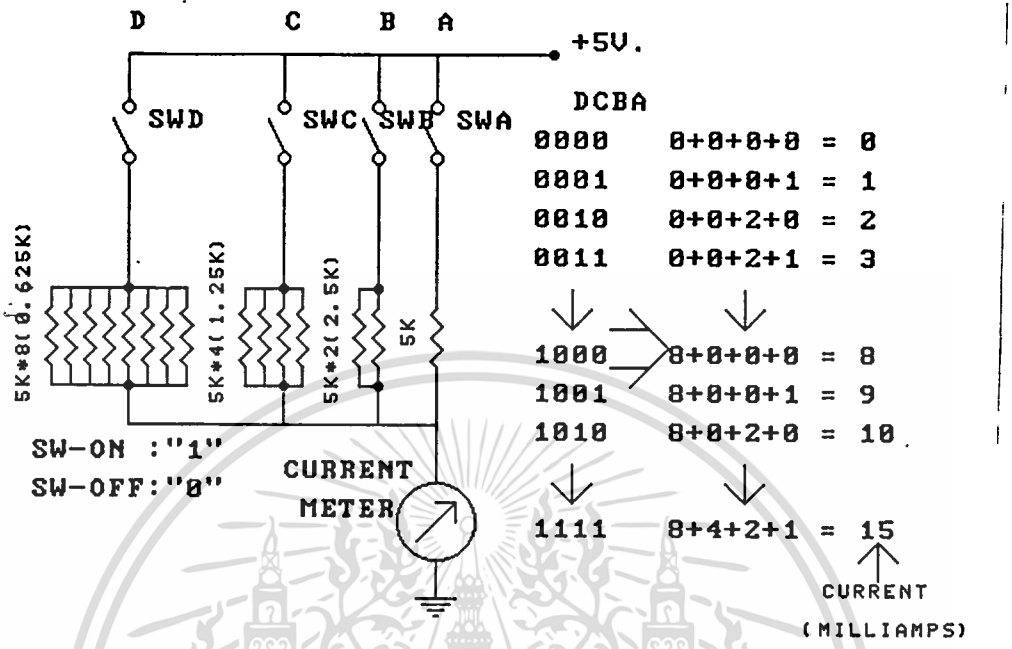
สวิทช์ทั้งสี่เรียงกันเป็นเลขฐานสอง 4 บิท ดังนั้นตัวเลขที่สามารถแสดงได้ คือ 0 ถึง 15 มิลลิแอมป์ เลขฐานสองเพิ่มทีละ 0001 ในขณะที่กระแสเพิ่มทีละ 1 มิลลิแอมป์ สวิทช์แต่ละตัวจะมี "น้ำหนัก" หรือขนาดกระแสที่ผลิตออกมาได้ต่างกัน คือ SW_A มีน้ำหนัก 1, SW_B มีน้ำหนัก 2,SW_C มีน้ำหนัก 4 และ SW_D มีน้ำหนัก 8 น้ำหนักนี้จะเพิ่มทีละเท่าขึ้นมาเรื่อยๆ ตามหลักการของเลขฐานสองนั่นเอง

เราใช้ตัวต้านทานขนาด 5 กิโลโอห์ม เพราะเราใช้แหล่งจ่ายไฟ 5 โอห์ม และต้องการให้กระแสไหลผ่านตัวต้านทานแต่ละตัวเพียง 1 มิลลิแอมป์เท่าๆ กัน เลือกใช้ตัวต้านทานเหมือนกัน 15 ตัว เพื่อให้ทดลองได้ง่ายและเข้าใจง่าย จะใช้ตัวต้านทาน 2.5 กิโลโอห์มเพียงตัวเดียวต่อที่ SW_B หรือ 1.25 กิโลโอห์มต่อที่ SW_C ก็ได้ แต่ค่าความต้านทานเหล่านี้หาได้ลำบากมาก

ลองเปิดปิดสวิทช์ SW_A จนถึง SW_D ตามเลขฐานสองจะได้กระแสไหลผ่านมิเตอร์ตามที่แสดงในตารางในรูปที่ 1 เช่น ถ้าปิด SW_A จะอ่านมิเตอร์ได้ 1 มิลลิแอมป์ ปิด SW_A และ SW_B จะได้ 3 มิลลิแอมป์ และถ้าปิดสวิทช์ทุกตัวจะได้ 15 มิลลิแอมป์ เป็นต้น

การทดลองนี้ใช้สวิทช์เพียง 4 ตัวเท่านั้น ที่จริงเราสามารถเพิ่มสวิทช์ขึ้นไปอีกเรื่อยๆ ค่าความต้านทานของสวิทช์ถัดมาจะน้อยลงไปทีละเท่า แต่ไม่ควรเพิ่มสวิทช์มาก เพราะกระแสจะไหลมากเกินไปจนเกินพิสัยของมิเตอร์ ถ้าต้องการทดลองกับสวิทช์หลายๆ ตัวจริงๆ ควรลดขั้นกระแสลงจาก 1 มิลลิแอมป์เป็น 0.1 หรือ 0.01 มิลลิแอมป์ก็ได้ โดยการเพิ่ม R จาก 5 กิโลโอห์ม เป็น 50 หรือ 500 กิโลโอห์มใช้สวิทช์ดังในรูปที่ 2.77 อาจจะยังมองไม่เห็นการใช้งานที่เป็นจริง ลองดูวงจรในรูปที่ 2.78 วงจรนี้ใช้หลักการเดียวกันคือการรวมกระแส แต่ใช้เอาท์พุทของวงจรนับชนิด CMOS แทนสวิทช์ทั้งสี่ เอาท์พุท Q_A ถึง Q_D ของวงจรนับ เมื่อเป็น "1" จะจ่ายกระแสเข้าไปในวงจรเช่นเดียวกับการปิดสวิทช์ SW_A ถึง SW_D ปกติ CMOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.77 การทดลองวงจร D/A แบบรวมกระแส

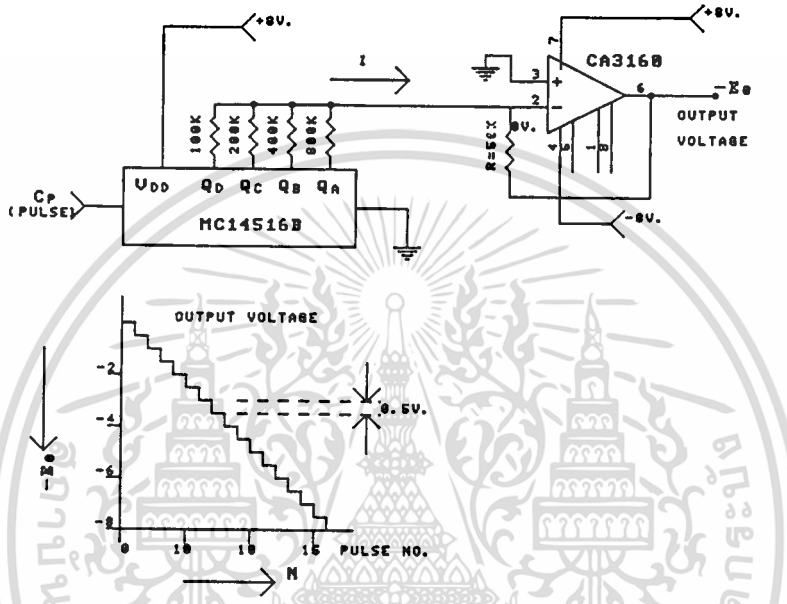
จ่ายกระแสได้ไม่มากนัก เพราะ จะทำให้แรงดันที่เอาท์พุทตกต่ำลง เราจึงเลือกใช้ตัวต้านทานในวงจรที่มีค่าใหญ่ๆ เพื่อให้กระแสไหลน้อยหน่อย และแทนที่จะใช้มิเตอร์วัดกระแสต่อโดยตรงเราจะใช้ออปแอมป์ต่อรับกระแสแทนและแหล่งกระแสเป็นแรงดันออกทางเอาท์พุทของออปแอมป์ด้วย

ขั้วลบ (ขา 2) ของออปแอมป์จะมีศักย์ไฟฟ้าเท่ากับขั้วบวก (ขา 3) ตามหลักการทำงานของออปแอมป์ ขั้วลบจึงมีศักย์ไฟฟ้าเป็นศูนย์ จึงทำให้กระแส I ที่ไหลจากตัวต้านทานคำนวณให้ได้เหมือนรูปที่ 2.77 นอกจากนั้นกระแส I นี้จะไม่ไหลเข้าออปแอมป์เลย เพราะออปแอมป์มีความต้านทานสูงมาก กระแสทั้งหมดจะไหลผ่านตัวต้านทาน r ไปยังเอาท์พุท แรงดันขาออกที่เอาท์พุทจึงเท่ากับแรงดันตกคร่อม r นั่นเอง แรงดัน E_O นี้จะเป็นค่าลบและจะมีขนาด 0.5 โวลท์ต่อการเปลี่ยนแปลงสัญญาณดิจิตอล 1 ขั้น ถ้าเปลี่ยนค่า r ให้เล็กหรือใหญ่ขึ้นก็สามารถเปลี่ยนขนาดแรงดันขาออกนี้ได้

เมื่อเราป้อนพัลส์เข้าวงจรนับ เอาท์พุท Q_A จนถึง Q_D จะเริ่มเปลี่ยนแปลง และเพิ่มค่าตามหลักการของเลขฐานสอง เอาท์พุทใดเป็น "1" ก็จะจ่ายกระแสออกไป กระแสที่จ่ายออกไปนี้จะมีขนาดตาม "น้ำหนัก" ของบิทนั้น ถ้านำมัลติมิเตอร์มาวัดแรงดันขาออกของออปแอมป์ในขณะที่วงจรนับกำลังนับพัลส์ จะเห็นเข็มมิเตอร์เพิ่มขึ้นเรื่อยๆ จาก 0 โวลท์ เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-0.5,-1,-1.5,-2.0 โวลท์ โดยเพิ่มทีละ -0.5 โวลท์แบบนี้ไปเรื่อยๆ จนถึง -7.5 โวลท์ ในที่สุด จากนั้นจะตกกลับเป็น 0 โวลท์ใหม่แล้วจึงค่อยๆ เพิ่มแรงดันขึ้นไปอีกถ้าใช้ออสซิลโลสโคปรับรูปคลื่น E_O จะเห็นรูปคลื่นคล้ายขั้นบันไดตามในรูปที่ 2.78



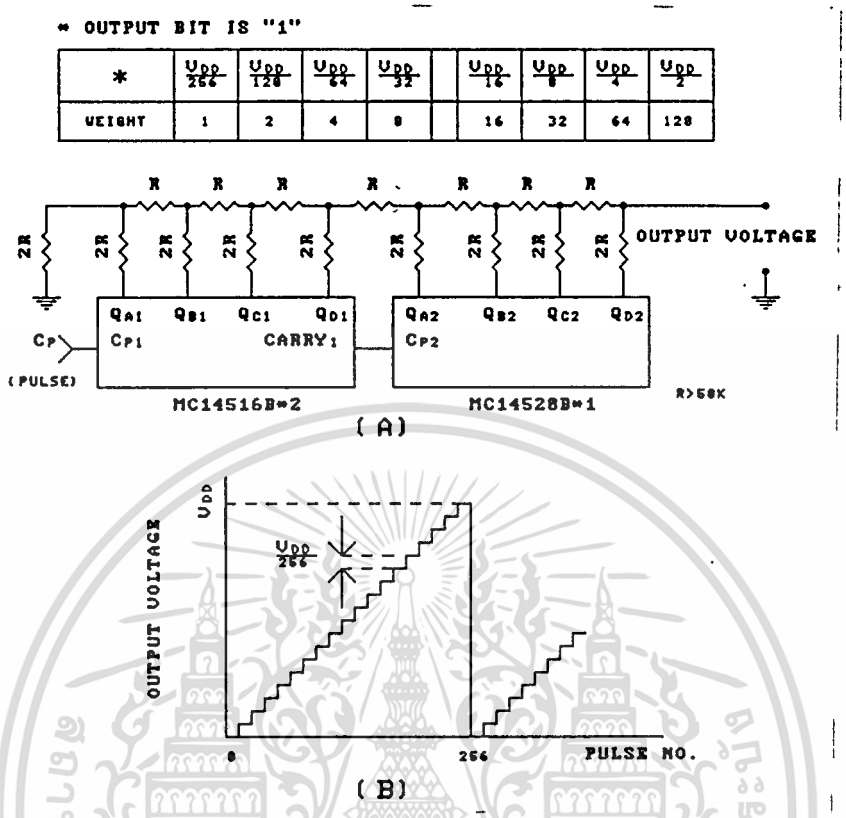
รูปที่ 2.78 ตัวอย่างวงจร D/A แบบรวมกระแส

2.7.2 วงจร D/A แบบวงจรบันไดของตัวต้านทาน

วงจร D/A แบบแรกเป็นแบบรวมกระแส วงจร D/A ที่จะกล่าวต่อไปเป็นแบบรวมแรงดัน ใช้ตัวต้านทานเพียง 2 ค่า คือ R และ 2R (ตัวต้านทานที่มีค่า 2*R โอห์ม) ต่อสลับไปมาเหมือนขั้นบันไดหรือเรียกว่า วงจร D/A แบบวงจรบันไดของตัวต้านทาน (R.2R ladder type ตามที่แสดงในรูปที่ 2.79 (ก) ในรูปเป็นวงจร D/A ขนาด 8 บิท ใช้วงจรนับ 4 บิท 2 ตัวต่ออนุกรมกันและใช้เอาท์พุทของวงจรนับทำหน้าที่เป็นสวิทช์ป้อนเลขฐานสองเข้าวงจร D/A

วงจรบันไดของตัวต้านทานนี้ผู้ออกแบบครั้งแรกนับว่ามีความสามารถมาก เพราะไม่ว่าจะวัดที่จุดไหนของแต่ละขั้นบันไดจะได้ความต้านทานรวมลงมาถึงปลายบันไดเท่ากับ R เสมอ ดังนั้นถ้าบิทสูงสุดคือ Q_{D2} เป็น "1" จะทำให้แรงดันขาออกเป็นครึ่งหนึ่งของแหล่งจ่ายไฟ คือ $V_{DD/2}$ ถ้าบิทถัดไป คือ Q_{C2} เป็น "1" ก็จะให้แรงดันขาออกลดลงไปครึ่งหนึ่ง คือ $V_{DD/4}$ บิทถัดๆ ไปก็จะทำให้แรงดันลดไป ทีละครึ่งเรื่อยๆ จนถึงบิทสุดท้าย คือ Q_{A1} จะให้แรงดันที่เอาท์พุทเป็น $V_{DD/256}$ ถ้านับแรงดัน $V_{D/256}$ เป็นแรงดัน 1 หน่วย แต่ละบิทก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.79 ตัวอย่างวงจร D/A แบบวงจรบันไดของตัวต้านทาน

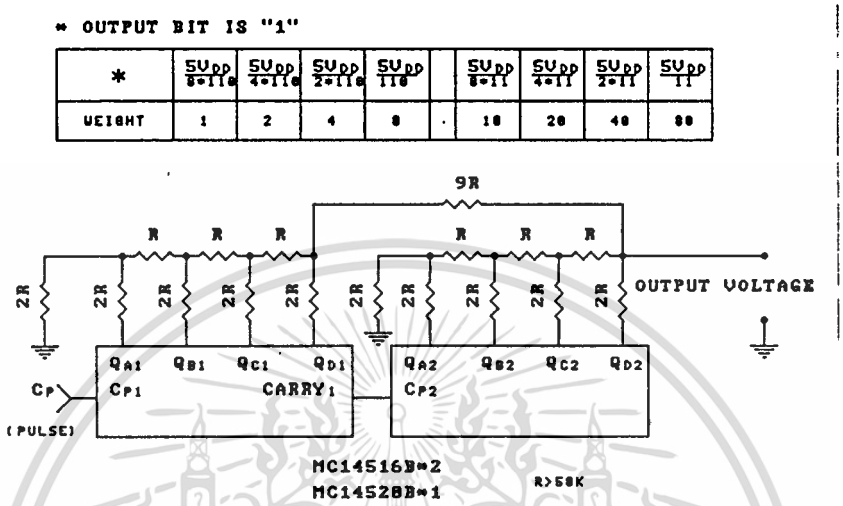
ให้แรงดันคิดเป็นน้ำหนักเรียงจากบิทต่ำสุดไปสูงสุด คือ 1,2,3,...128 เหมือนกับแบบรวมกระแสตามที่ได้อธิบายมาแล้ว

ถ้าป้อนพัลส์ให้วงจรนับแล้ววัดแรงดันขาออกด้วยมัลติมิเตอร์ ก็จะเห็นเข็มมิเตอร์ค่อยๆ กระดิกขึ้น หรือถ้าใช้ออสซิลโลสโคปจับดูรูปคลื่นก็จะเห็นเป็นรูปคลื่นฟันเลื่อยแบบขั้นบันไดเหมือนกัน ตัวต้านทาน R และ 2R ที่ใช้ในวงจรควรเป็นตัวต้านทานที่มีความแม่นยำสูง เช่น ความผิดพลาดไม่เกิน 1 เปอร์เซ็นต์ ถ้าผิดพลาดมากจะทำให้สัญญาณอนาลอกที่ได้ผิดพลาดได้

วงจรในรูปที่ 2.79 เป็นวงจร D/A ที่ใช้กับเลขฐานสอง ถ้าเป็นเลขฐานสิบรหัส BCD ซึ่งเรานิยมใช้กันมากนั้นก็จำเป็นต้องดัดแปลงวงจรบันไดของตัวต้านทานใหม่เป็นอย่างในรูปที่ 2.80 วงจร D/A นี้มีวงจรบันไดอยู่ 2 วงจร เป็ยของหลักหน่วยวงจรหนึ่งและเป็นของหลักสิบอีกวงจรหนึ่ง แต่ละวงจรใช้ตัวต้านทาน R และ 2R เหมือนกัน สองวงจรนี้ต่อเชื่อมกันด้วยตัวต้านทานที่มีขนาด 9R เพื่อให้แรงดันขาออกของวงจรบันไดหลักหน่วยมีขนาดเล็กลง 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่า เมื่อมารวมรับแรงดันขาออกของวงจรบันไดหลักสิบ ที่ใช้ค่า 9R เพราะความต้านทานขาออกของวงจรบันไดมีค่า R พอดีตามที่ได้อธิบายไปแล้ว



รูปที่ 2.80 วงจร D/A ที่ใช้กับเลขฐานสิบรหัส BCD 2 หลัก

วงจรบันไดของตัวต้านทานนี้สามารถใช้งานในทางปฏิบัติได้ถึง 3 หลักของเลขฐานสิบ ถ้าเกินกว่า 3 หลักแล้ว จะต้องคัดเลือกค่าความต้านทานให้แม่นยำ มิฉะนั้นจะเกิดความคลาดเคลื่อนมาก วงจรบันไดของตัวต้านทนก็มีที่ทำเป็นไอซีสำเร็จรูปขาย ไอซีบางตัวที่เป็นวงจร D/A นั้น แท้จริงภายในจะมีวงจรบันไดของตัวต้านทานและแหล่งจ่ายกระแสที่ถูกควบคุมด้วยสัญญาณ "0" และ "1" ของอินพุทเท่านั้น ไอซี D/A ที่นิยมใช้กันมากได้แก่ เบอร์ MC1408 ของบริษัทโมโตโรลา เป็นวงจร D/A ขนาด 8 บิท

2.8 ส่วนประกอบของวงจรเครื่องรับ

ในวงจรภาครับจะประกอบด้วย วงจร ANT (Antenna), วงจรขยาย HF, วงจรผลิตสัญญาณตัวที่ 1, วงจรมิกเซอร์ตัวที่ 1, วงจรผลิตสัญญาณตัวที่ 2, วงจรมิกเซอร์ตัวที่ 2, วงจรขยายสัญญาณ IF ตัวที่ 2, ตัวแยกความถี่, วงจรขยายสัญญาณ AF (1), วงจรขยายสัญญาณ AF (2), วงจรดีโคดเดอร์ (ถอดรหัส), วงจรผลิตสัญญาณนาฬิกา

2.8.1 วงจรสายอากาศ

วงจรนี้จะทำให้เกิดสัญญาณที่ดี ถ้าใช้กับสายอากาศเฟอร์ไรท์ (L1), ตัวเก็บประจุที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปรับค่าได้ (CV1) และ C1 และจะส่งสัญญาณทั้งหมดไปยังวงจรขยาย HF (IC1)

2.8.2 วงจรขยายสัญญาณ HF

วงจรนี้ประกอบด้วยวงจรจูนนิ่ง T1-T3 และวงจรขยายสัญญาณ HF (IC1) ด้วยเหตุนี้เราจะเอาแบบจำลองผลตอบสนองของ IF ตัวที่ 1 มาใช้ สัญญาณจะถูกขยายและส่งไปที่เบสของมิกเซอร์ตัวที่ 1 (IC2)

2.8.3 ส่วนของออสซิเลเตอร์ตัวที่ 1 และมัลติไพล์เออร์

วงจรนี้ประกอบด้วย IC3 (MC-4168), T4, T5 และ T7 ความถี่ของวงจรออสซิเลเตอร์หาได้จาก

$$FL = (FS - 10.7) / 3 \text{ (MHz)}$$

เมื่อ

FL = ความถี่ออสซิเลเตอร์

FS = ความถี่พาหะ

ฮาโมนิคที่ 3 (3FL) ของความถี่ FL ที่ผลิตขึ้นนี้จะถูกปรับปรุงโดยวงจร Tank ซึ่งประกอบด้วย T4 และ C10 และถูกนำไปยังเบสของ IC2 โดยตัวเก็บประจุ C9

2.8.4 วงจรมิกเซอร์ตัวที่ 1

วงจรนี้ประกอบด้วย IC2 (MC-4168) และ T6 ประกอบด้วยความถี่ 10.7 MHz ซึ่งเป็นความแตกต่างระหว่างความถี่พาหะ FS และความถี่ออสซิเลเตอร์ตัวที่ 1 (3FL), ซึ่งจะถูกทำให้ดีขึ้นโดย T6 และพาไปที่ F1 หลังจากการแมทชิ่งอิมพีแดนซ์

2.8.5 มิกเซอร์และออสซิเลเตอร์ตัวที่ 2

วงจรนี้ประกอบด้วย IC4 (MC-4169) และ X2 ความถี่ออสซิเลเตอร์ คือ 10.245 MHz เอาท์พุทที่เกิดขึ้นจะผ่านไปที่มิกเซอร์ตัวที่ 2 โดย C19 และถูกประยุกต์ใช้เป็น BPF (F2) ซึ่งประกอบด้วยความถี่ 455 kHz

2.8.6 วงจรขยาย IF ตัวที่ 2, ตัวแยกความถี่

วงจรขยาย IF ตัวที่ 2 และตัวแยกความถี่ ประกอบด้วย IC5 และ IC6 มีอัตราขยายแรงดันประมาณ 120 dB เอาท์พุทของ IC6 จะถูกรวมเข้ากับตัวแยกความถี่ ซึ่งประกอบด้วย T9,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

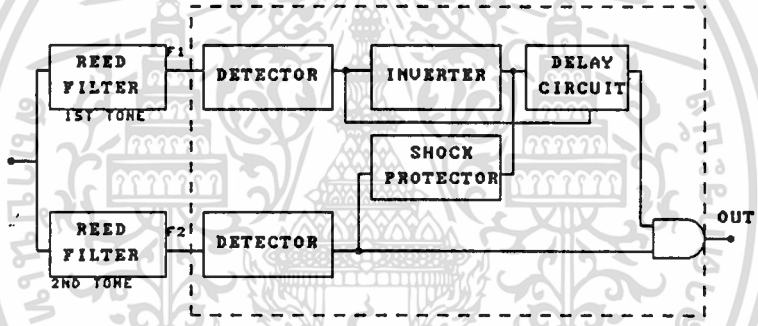
C24, C25, C26, D1 และ D2 ใช้สำหรับการ Demodulate

2.8.7 ส่วนขยาย AF

ส่วนขยายนี้ประกอบด้วย วงจรขยาย AF (IC7) และ T10, IC7 จะเป็นตัวขยายสัญญาณที่ทำการ Demodulate แล้ว และถูกนำไปกรอง 2 ครั้งที่เวลาเดียวกัน ซึ่งมันจะทำให้เกิดการแมทชิ่งอิมพีแดนซ์โดย T10 และถูกนำไปยังขาเบสของ TR1 และ TR2

2.8.8 Decoder

ส่วนของดีโค้ดเดอร์ ประกอบด้วย IC8 (MC-4173)



รูปที่ 2.81 บล็อคไดอะแกรมของ IC8

สัญญาณ AC จะถูกแยกโดยฟิลเตอร์ (F1, F2) คือ จะถูกแปลงเป็นแรงดันทาง DC โดยตัวดีเทคเตอร์พัลซ์สัญญาณเสียงตัวที่ 1 จะถูกนำไปผ่าน AND GATE โดยผ่านอินเวอร์เตอร์ และวงจรหน่วง เอาท์พุทเกทจะเกิดขึ้นเมื่อพัลซ์สัญญาณเสียงถูกส่งไปยัง AND GATE ตัวป้องกันการสะเทือนของสัญญาณจะเกิดขึ้นเมื่อป้องกันสัญญาณที่ไม่ต้องการเซอร์กิตไดอะแกรมของ IC8 และรูปคลื่นแรงดันถูกแสดงในรูปที่ 2.82 และรูปที่ 2.83 ตามลำดับ

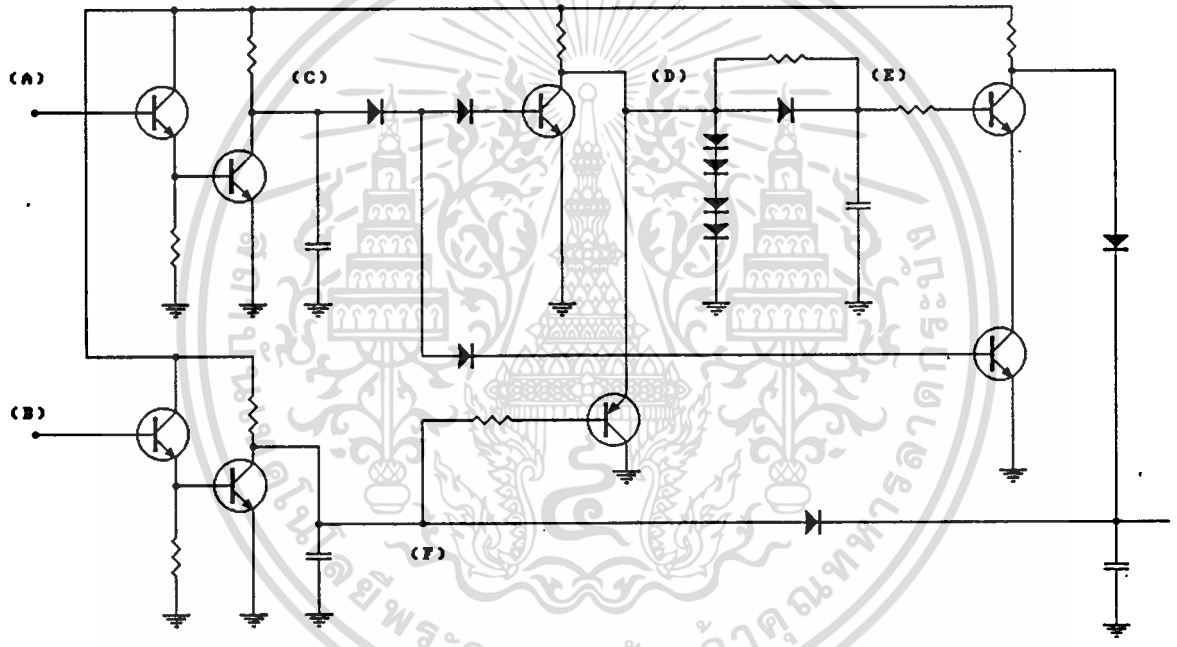
2.8.9 Timer

วงจร Timer ประกอบด้วย IC9 (MC-8007) และเป็นตัวไบอัสให้กับวงจรขยาย AF ตัวที่ 1 และ 2 ออสซิเลเตอร์และวงจรควบคุมปริมาตร รูปที่ 2.84 แสดงเซอร์กิตไดอะแกรมของ IC9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

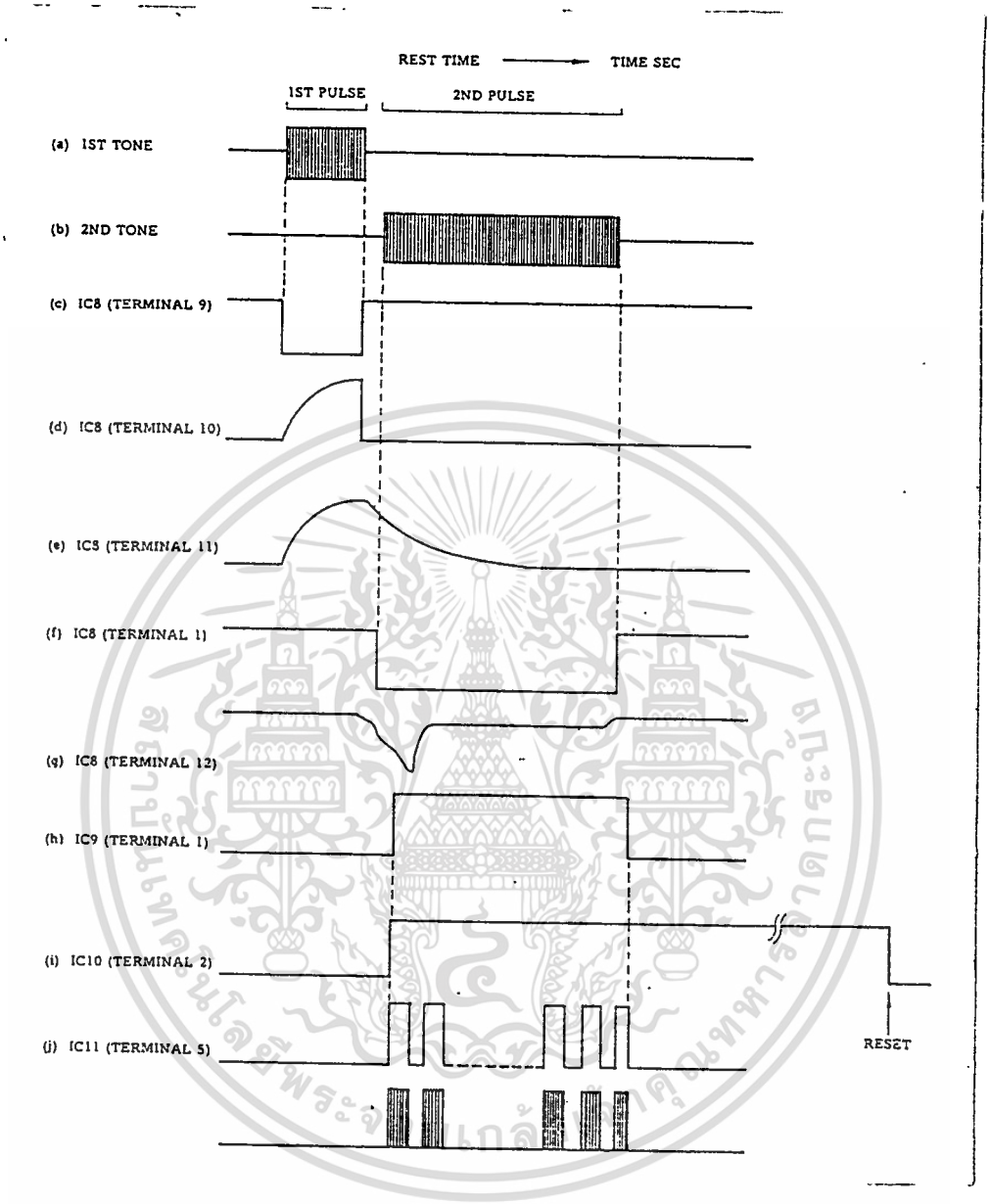
แรงดันจะเกิดขึ้นที่ขา 1 และ 7 ของ IC9 เมื่อเอาท์พุท (ขณะที่ไม่มีพัลซ์) ของ IC9 เป็นตัวดึงสัญญาณไฟที่ขา 3 โดยผ่าน D9 หลังจากนั้นประมาณ 3 วินาที แรงดันที่ขา 1 ก็จะถูกสร้างขึ้นเหมือนเดิม และขา 7 จะรักษาสภาวะเดิมจนกระทั่งขาที่ 7 รีเซท แรงดันที่ขา 1 จะถูกส่งผ่านไปยังออสซิเลเตอร์ IC10 (MC-4109) ผ่าน F17 และแยกความถี่ผ่าน R26 และ D7

ในระหว่างการทำงานนี้ ฟังก์ชั่นของการ Demodulate ของตัวแยกความถี่ จะไม่ถูกป้องกันการ Demodulate สัญญาณจากการถูกรวมสัญญาณกับเสียงที่มีความไว แรงดันที่ขา 7 จะถูกแยกไปแต่ละวงจรซึ่งแสดงในรูปที่ 2.85



รูปที่ 2.82 Circuit Diagram ของ IC8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

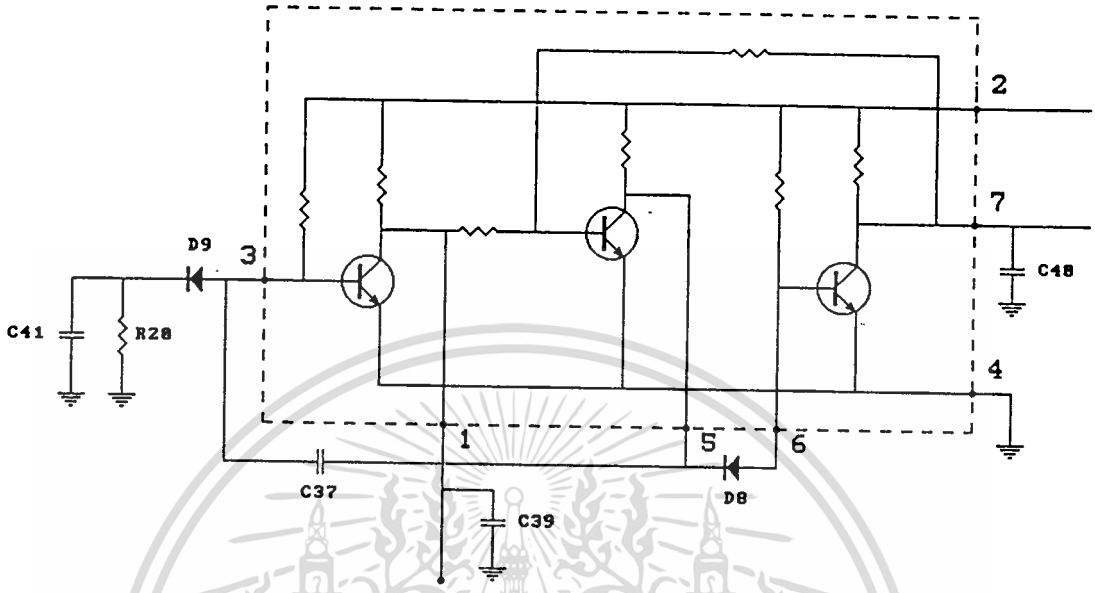


รูปที่ 2.83 รูปคลื่นแรงดันของ IC8

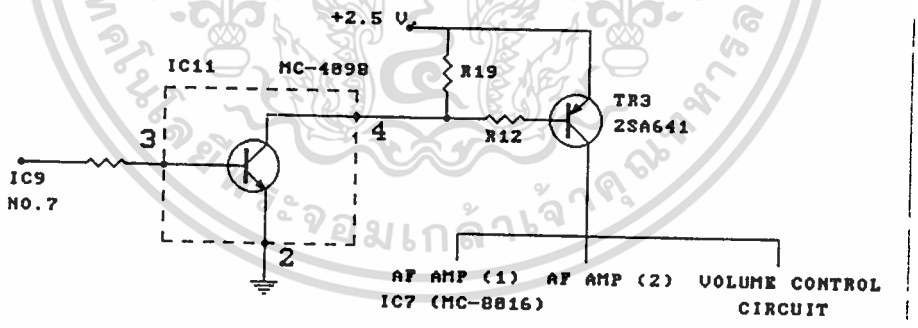
วงจรควบคุมปริมาตรจะถูกจัดหาสำหรับควบคุมสัญญาณอินพุทของวงจรขยาย AF ตัวที่ 1 และ IC7 (MC-8016)

เมื่อ D3 ไม่สามารถไบอัสผ่าน R13 และ D4, เสียงที่ถูก Demodulate จะถูกส่งผ่านไปวงจรขยาย AF IC7 โดยไม่คำนึงถึงตำแหน่งของการควบคุมว่าเป็น "H" หรือ "L" (จนกระทั่งอิมพีแดนซ์ของไดโอด D3 เป็น "High") ถ้า IC9 ทำงานและ D3 ถูก Bias อินพุทที่ไปที่ IC7 จะลดลงเป็นเพราะตำแหน่งของการควบคุม "H" และ "L"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.84 เซอร์กิตไดอะแกรมของ IC9

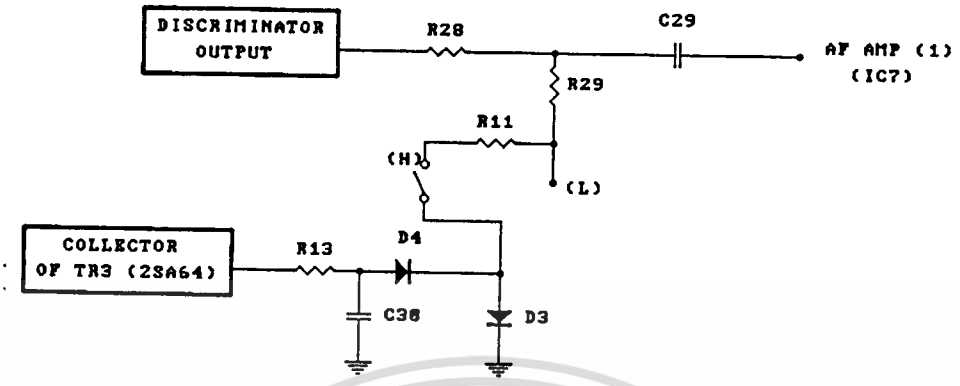


รูปที่ 2.85 วงจรขยายสัญญาณขา 7 ของ IC9

2.8.10 วงจรขยาย AF ตัวที่ 2

วงจรขยาย AF ตัวที่ 2 จะเป็นตัว พุช-พูล การขยายกำลัง ซึ่งประกอบด้วย TR1, TR2, T10 และ T11 วงจรขยายนี้จะไม่ทำหน้าที่เป็นวงจรขยายภายใต้สภาวะ Stand-by จนกระทั่งไบอัสจะไม่เกิดขึ้นที่ขาเบสของ TR1 และ TR2 เพราะฉนั้นจะไม่ได้ยินเสียงจากลำโพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.86 วงจรควบคุมปริมาตร

ถ้าเครื่องรับถูกเรียก ไทม์เมอร์ (IC9) จะทำการดีโคดเดอร์เอาท์พุท และแรงดัน 2.4V จะมาที่ขาคอลเลคเตอร์ของ TR3 แรงดันนี้จะผ่านมาทาง R28 เป็นไบอัสโวลท์เตจที่ขาเบสของ TR1 และ TR2 และเสียงที่มีความไวและข่าวสารจะเปล่งเสียงจากลำโพงโดยไบอัสโวลท์เตจนี้

2.8.11 ออสซิเลเตอร์

วงจรออสซิเลเตอร์ประกอบด้วย IC10 และ IC11 แรงดันเอาท์พุทของ IC9 (ขา 1) จะถูกส่งไปยัง IC10 (MC-4109) ผ่าน R17 IC10 จะเป็น Multivibrator ซึ่งผลิตความถี่ประมาณ 3 Hz เอาท์พุทของ IC10 (ขา 2) จะถูกส่งไปยังขา 4 ของ IC11 (MC-4098) และถูกกระตุ้นโดย Multivibrator การออสซิเลทความถี่ของออสซิเลเตอร์นี้ประมาณ 1.6 kHz เอาท์พุทของ IC11 ถูกส่งไปที่ IC7 โดยผ่าน R33 หลังจากที่เกิดการขยาย และจะเกิดเสียงจากลำโพงเหมือนกับความไวเสียง

2.8.12 แจ็ค "S" และ "A"

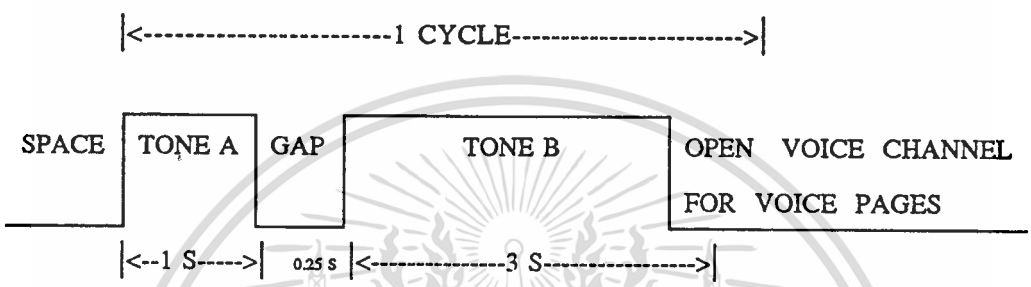
หนึ่งในแจ็ค "S" และ "A" จะถูกส่งผ่านจากสายอากาศ L1 โดย C45 ความไวจะมีมากขึ้นเมื่อหูฟังและสายอากาศถูกใช้ (วงจรเสียงพูดจะไม่ถูกปิดเมื่อสายอากาศถูกใช้)

2.9 การ Generate Code ตามมาตราฐาน Plan C

เครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์ นั้น การเรียกไปยังเครื่องวิทยุติดตามตัวจะใช้สัญญาณซายน์เวฟ 2 ความถี่ ในการมอดูเลชั่นไปกับคลื่นพาห์ ความถี่ 170.900 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวคือ เป็นเครื่องสร้างสัญญาณลักษณะซายน์เวฟที่ถูกกำหนดความถี่คงที่ไว้เป็นชุด แต่ละชุดจะมีความถี่ 2 ความถี่ ค่าของความถี่แต่ละชุดที่ต้องการจะถูกกำหนดไว้เป็นรหัส เรียกว่า รหัส Plan C คือ จะส่งสัญญาณโทนที่หนึ่งเป็นเวลาประมาณ 1 วินาที, หยุดส่งสัญญาณ 0.25 วินาที และส่งสัญญาณโทนที่สองอีกประมาณ 3 วินาที ดังรูปที่ 2.87



รูปที่ 2.87 Plan C Code Waveform

ความถี่หลักของเครื่องแบบ Plan C จะมีทั้งสิ้น 30 ชุด กำหนดเป็นกลุ่มย่อยๆ กลุ่มละ 10 ชุด ได้เป็น 3 กลุ่ม คือ A,B และ C ตามตารางที่ 2.8

ที่	กลุ่ม A		กลุ่ม B		กลุ่ม C	
	โมโดโรล่าร์	ความถี่(Hz)	โมโดโรล่าร์	ความถี่(Hz)	โมโดโรล่าร์	ความถี่(Hz)
0	140	321.7	110	330.5	120	569.1
1	141	339.6	111	349.0	121	600.9
2	142	358.6	112	368.5	122	634.5
3	143	378.6	113	389.0	123	669.9
4	144	399.8	114	410.8	124	707.3
5	145	422.1	115	433.7	125	746.8
6	146	445.7	116	457.9	126	788.5
7	147	470.5	117	483.5	127	832.5
8	148	496.8	118	510.5	128	879.0
9	149	524.6	119	539.0	129	928.1

ตารางที่ 2.8 แสดงค่าความถี่ของ Plan C กลุ่มต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเข้ารหัส Plan C จะกำหนดโปรแกรมให้ค่าของตัวเลขหลักร้อยเป็นตัวกำหนดว่าจะสร้างความถี่ชุดใด (แบ่งเป็น 9 ชุด ตามรหัสเรียกขาน 100 - 900) เมื่อเลือกชุดความถี่ได้แล้ว ก็จะนำเอาค่าตัวเลขหลักสิบเป็นตัวชี้ค่าความถี่ของสัญญาณโทนชุด A และตัวเลขหลักหน่วยเป็นตัวชี้ค่าความถี่ของสัญญาณโทนชุด B

หลักร้อย	หมายเลขเรียกขาน	กลุ่มความถี่
1	100 -199	B - B
2	200 -299	C - C
3	300 -399	B - C
4	400 - 499	A - A
5	500 - 599	B - A
6	600 - 699	C - B
7	700 - 799	A - B
8	800 - 899	C - A
9	900 - 999	A - C

ตารางที่ 2.9 Plan C Code Table

สมมุติว่า ต้องการเรียกวิทยุติดตามตัวหมายเลข "456" อันดับแรก เครื่องจะนำหลักร้อย คือ เลข "4" มาเลือกหาชุดความถี่ในตารางที่ 2.2 ซึ่งตรงกับชุดที่ 4 กำหนดให้ใช้ความถี่กลุ่ม A กับ A จากนั้นจะนำค่าหมายเลขที่จะเรียกหลักสิบ คือ เลข "5" มาเป็นตัวชี้ค่าความถี่กลุ่ม A ลำดับที่ 5 จะได้ความถี่โทน A เป็น 422.1 Hz และขั้นตอมาจะนำค่าหลักหน่วยของหมายเลขที่จะเรียก คือ เลข "6" มาเป็นตัวชี้ค่าความถี่ กลุ่ม A ลำดับที่ 6 ก็จะได้ความถี่ของโทน B เป็น 445.7 Hz นั่นคือ เมื่อเราต้องการเรียกเครื่องวิทยุติดตามตัวหมายเลข "456" เครื่องก็จะสร้างสัญญาณซายน์เวฟความถี่ 422.1 Hz และ 445.7 Hz เป็นสัญญาณโทน A และสัญญาณโทน B ตามลำดับ

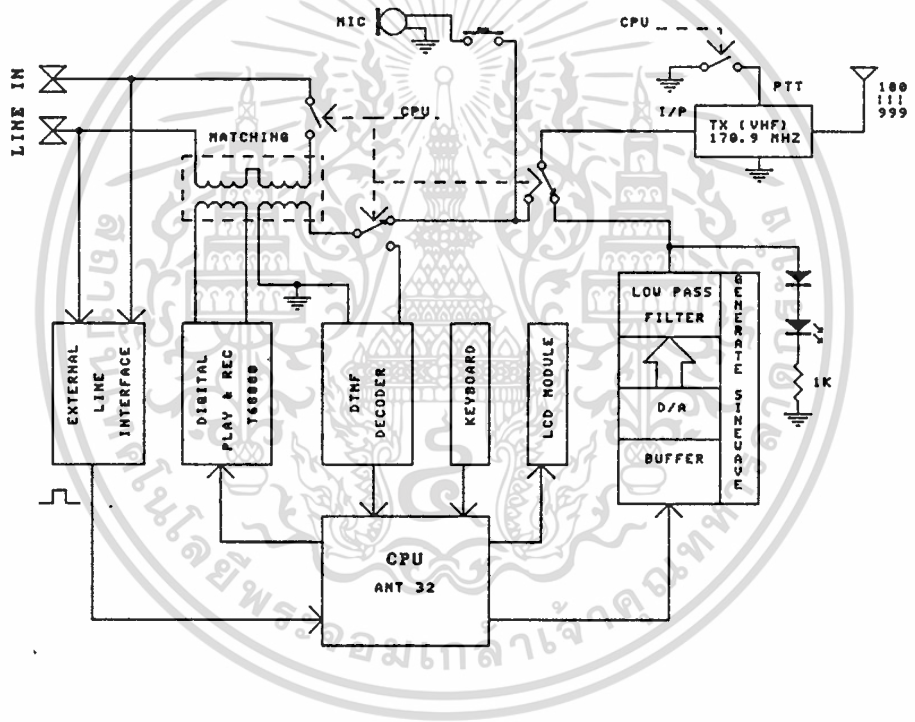
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ

การออกแบบเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

Automatic Pager Call Tone & Voice System



รูปที่ 3.1 Block Diagrame Automatic Pager System

3.1 การออกแบบ HARD WARE

การออกแบบเครื่องเรียกวิทยุติดตามตัวแบบอัตโนมัตินี้ การสร้างทำการประกอบวงจรลงบนแผนปริ๊นท์เอนกประสงค์ เนื่องจากโครงงานอยู่ในช่วงทดลอง เพื่อความสะดวกในการปรับปรุงแก้ไข

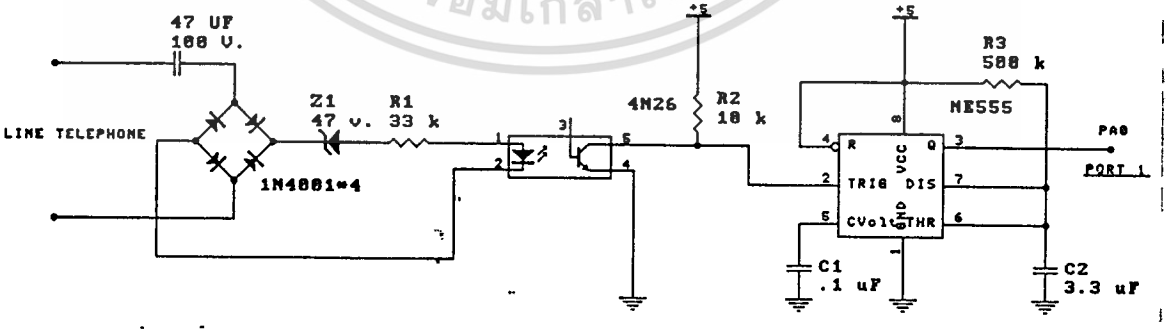
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการทำงานของเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ ดังรูปที่ 3.1 ประกอบด้วยวงจรที่เป็นส่วนสำคัญ 9 ส่วน ดังนี้

1. External Line Interface
2. Digital Paly & Rec. (T68000)
3. DTMF Decorder (Dual Tone Multi Frequecy Decorder
4. MATRIX Keyboard
5. LCD Modute
6. Sinewave Genalate
7. CPU (ANT-32)
8. การโปรแกรม IC 8255 บนบอร์ด ANT-32 ให้อยู่ในโหมดต่างๆ
9. Power Supply

3.1.1 External Line Interface

วงจรนี้ทำหน้าที่รับสัญญาณเรียก (Ringing Tone) ที่มาจากการเรียกเข้ามาของผู้ใช้แล้ว ทำการเปลี่ยนสัญญาณที่เข้ามาเป็นสัญญาณพัลซ์ จากนั้นจะส่งสัญญาณพัลซ์นี้ไปยัง CPU เพื่อให้ CPU รู้ว่ามีการติดต่อเข้ามา โดยผ่านทางพอร์ท 8255 จากนั้น CPU จะทำการส่งสัญญาณไปยัง Q1(C1061) เพื่อให้ RY1 ทำงาน ซึ่งเสมือนเป็นการสั่งให้ยกหูโทรศัพท์ เพื่อรอรับขั้นตอนการทำงานต่อไป



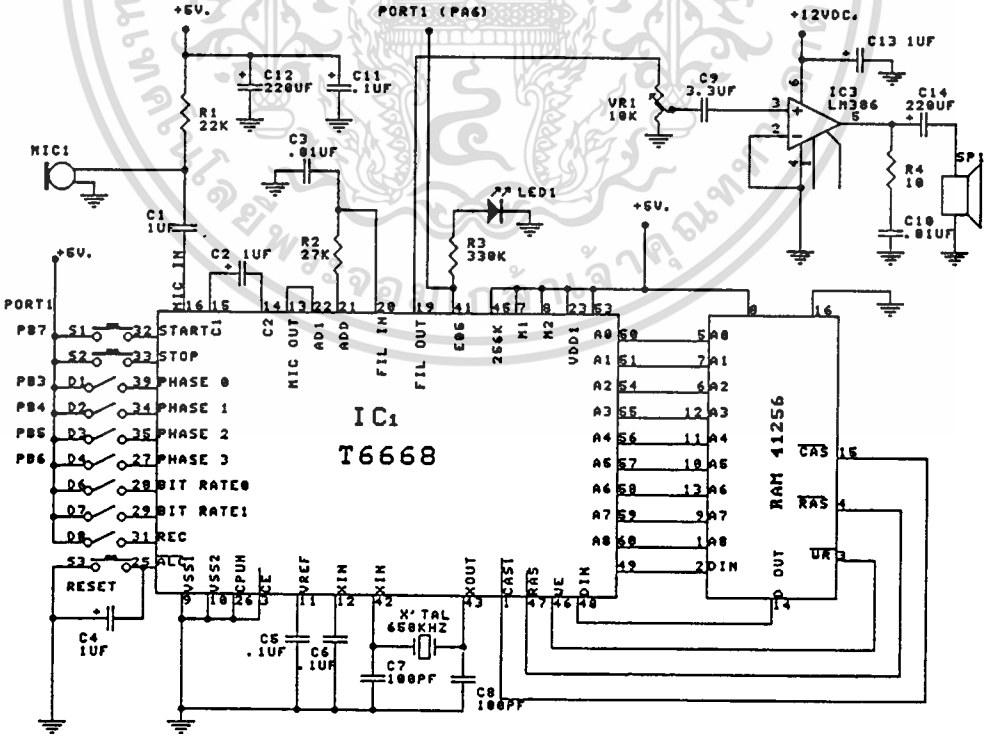
รูปที่ 3.2 วงจร External Line Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรรูปที่ 3.2 สัญญาณเรียกที่ส่งมาจากชุมสายขององค์การโทรศัพท์ จะมีแรงดันประมาณ 100Vp-p (ถ้ามีการเรียกเข้ามา) ที่ Line Telephone จะผ่าน C 47 uF/100V และผ่านวงจร Bridge Rectifier ผ่าน Zener Diode 47 V ผ่าน R 33KΩ ผ่าน Opto Isolater (4N26) เมื่อมีสัญญาณเรียกเข้ามาจะทำให้ LED ของ Opto Isolate เปล่งแสง ทำให้ทรานซิสเตอร์ใน Opto Isolate นำกระแสมีผลทำให้ที่ขา 2 ของ IC555 มีลอจิกเป็น "0" ซึ่ง IC555 นี้ทำการต่อเป็นวงจรโมโนสเตเบิ้ล เมื่อขา Trig (ขา 2) ของ IC555 เปลี่ยนสถานะจาก "1" เป็น "0" ทำให้ที่ขา O/P (ขา 3) ของ IC555 เกิดพัลซ์ชั่วขณะหนึ่ง เพื่อส่งไปให้ CPU ทราบว่ามีการเรียกเข้ามาโดยผ่านทาง Port 1 (PA0) ของ 8255 จำนวน 1 Bit

3.1.2 Digital Play & Record

วงจรนี้ทำหน้าที่ ตอบรับโทรศัพท์ที่เรียกเข้ามา โดยเป็นเสียงพูดออกไปยังผู้ที่เรียกเข้ามา โดยการเลือกข้อความที่ต้องการตอบกลับไป ซึ่งการเลือกข้อความจะใช้ CPU เป็นตัวเลือกโดยผ่านมาทาง Port 1 (PB3 - PB7) ของ 8255 จำนวน 5 Bit



รูปที่ 3.3 วงจร Digital Play & Record

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจร รูปที่ 3.3 เป็นเสมือน Tape ทั่วไป คือ มีการบันทึกลงไปแล้วนำข้อความเหล่านั้นมาเล่นกลับ แต่วงจรนี้ทำงานในระบบดิจิตอล (การใช้งานโดยละเอียดดูได้จากทฤษฎีในบทที่ 2) ข้อความที่บันทึกไว้ในช่วงต่างๆ คือ

D3	D2	D1	D0	ข้อความ
0	0	0	0	เว้นว่างไว้
0	0	0	1	กรุณากดหมายเลข Pager ที่ท่านต้องการเรียก
0	0	1	0	ไม่สามารถติดต่อหมายเลขนี้ได้
0	0	1	1	กรุณาพูดข้อความท่านมีเวลาพูด 20 วินาที
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	ขอขอบคุณที่ใช้บริการ ของ กสท.

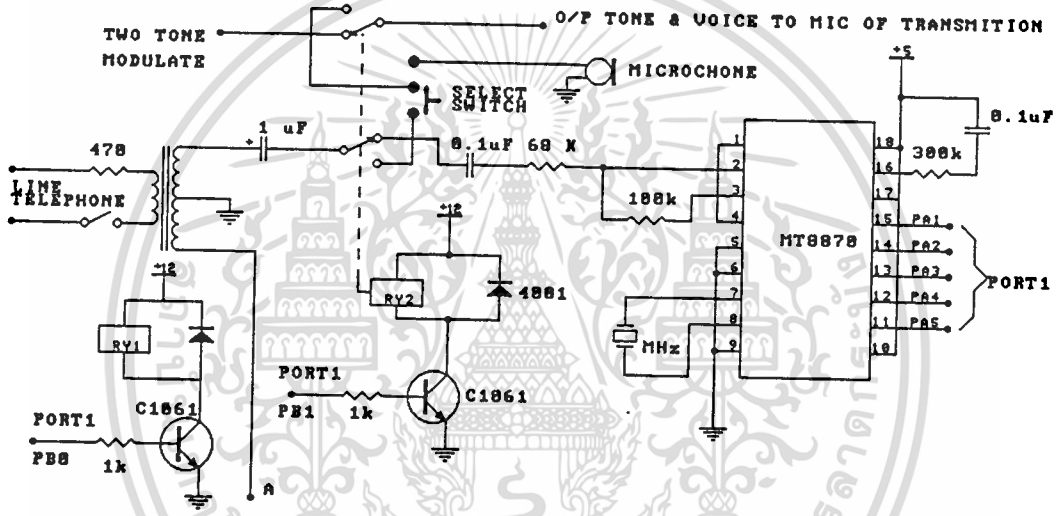
ตารางที่ 3.1 การกำหนดข้อความในช่องต่างๆ ของวงจร Digital Play & record

เมื่อเราต้องการที่จะนำข้อความที่อยู่ในช่องใดออกมาตอบกลับไปยังผู้เรียก ก็จะสั่งให้ CPU กำหนดสัญญาณที่ Port 1 ขา PB3 - PB6 (D0 - D3) ของ 8255 ค้างไว้ตามตาราง จากนั้นก็จะส่งสัญญาณมา Start วงจร Digital Play & Record โดยผ่านทาง Port 1 ขา PB7 ของ 8255 เพื่อนำข้อความที่ต้องการตอบกลับไปยังผู้เรียกใช้บริการ

3.1.3 DTMF Decorder (Dual Tone Multi Frequecy Decorder)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรในส่วนนี้ทำหน้าที่ถอดรหัสสัญญาณ DTMF ใช้ IC DTMF Decorder, MT 8870 ซึ่งจะทำการถอดรหัสสัญญาณ DTMF อันเกิดจากการกดคีย์ที่หน้าปัทม์ของเครื่องโทรศัพท์ที่ใช้เรียกเข้ามา โดยที่วงจรนี้จะทำหน้าที่การรับสัญญาณเข้ามา จากนั้นจะทำการถอดรหัสสัญญาณเป็นรหัสไบนารี่ 4 บิท และในขณะที่มีการกดคีย์โทรศัพท์ จะทำให้ขาสโตป (ขา 15 ของ MT8870) ซึ่งเป็นขาเอาท์พุท จะส่งสัญญาณไปให้ CPU รับรู้ว่ามีการกดคีย์โทรศัพท์เข้ามาโดยผ่านทาง Port 1 (PA1) ของ 8255 ทันทีที่มีการกดคีย์ จากสัญญาณพัลส์ที่ขาสโตปพร้อมกับรับข้อมูล ซึ่งเป็นรหัสไบนารี่ไปทำงานตามคีย์ที่กดมาต่อไป



รูปที่ 3.4 วงจร DTMF Decorder

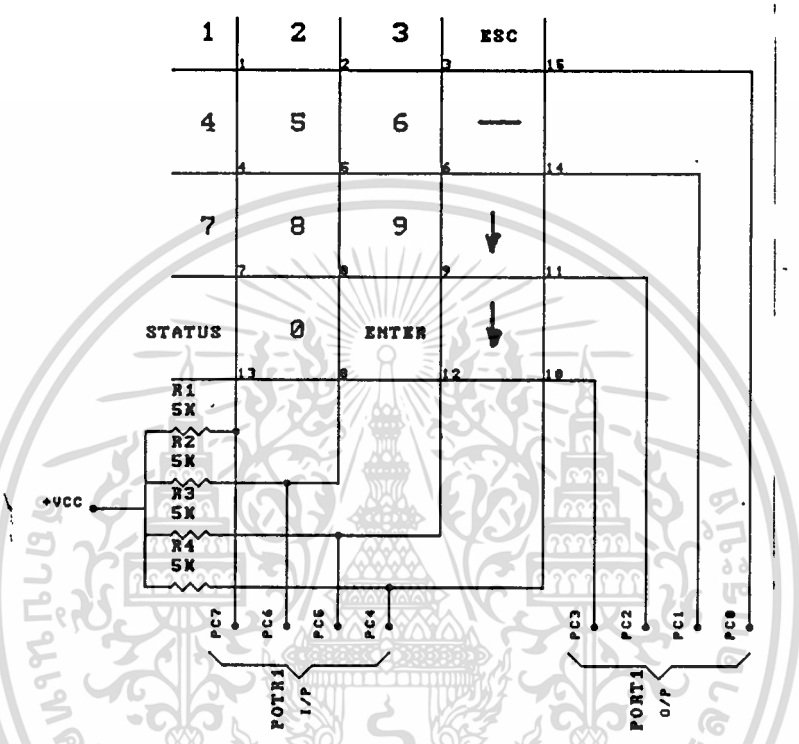
รูปที่ 3.4 แสดงวงจร DTMF Decorder วงจรนี้ใช้ IC MT8870 ทำหน้าที่เปลี่ยนสัญญาณ Tone อันเกิดจากการกดคีย์โทรศัพท์ เปลี่ยนมาเป็นสัญญาณไบนารี่ 4 บิท อินพุทของวงจรจะต่อกับ Line โทรศัพท์ โดยผ่าน Matching Transformer เพื่อรอรับการกดคีย์หมายเลขของ Pager ขา Strob ของ MT8870 จะต่อกับ CPU โดยผ่านทาง Port 1 (PA1) ของ IC8255 ของบอร์ด ANT32, ขาเอาท์พุทของ MT8870 (PC0-PC3) จะต่อเข้ากับ CPU ทาง Port 1 (PA2-PA5) ของ IC 8255 ของบอร์ด ANT32

3.1.4 MATRIX Keyboard

เป็นวงจร Matrix Keyboard ขนาด 4*4 = 16 Key ใช้ในการเลือกเมนูการทำงานต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และใช้เรียก Pager ในระบบ Manual โดยต่อกับ CPU โดยผ่านทาง Port 1 (PC0-PC3) ของ IC8255 ของบอร์ด ANT32 เซ็ทเป็น Out Port และผ่านทาง Port 1 (PC4-PC7) ของ IC8255 ของบอร์ด ANT32 เซ็ทเป็น In Port โดยมีการวางคีย์การใช้งานดังรูปที่ 3.5



รูปที่ 3.5 วงจร Matrix Keyboard 4*4

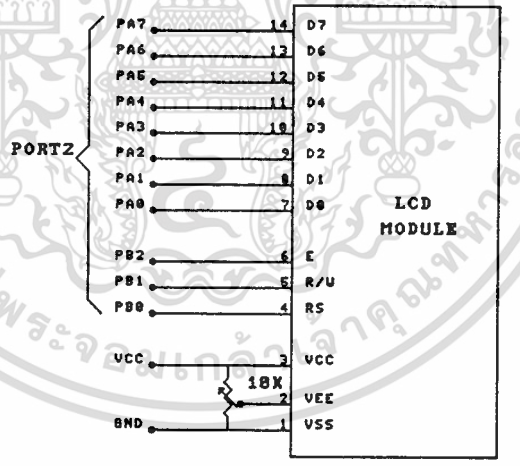
3.1.5 LCD Module

เป็น Monitor ของเครื่อง ในที่นี้ใช้ขนาด 16 ตัวอักษร * 4 แถว โดยการควบคุมจาก CPU โดยผ่านทาง Port 2 (PB0-PB2) ของ IC8255 เป็นบิทควบคุม 3 บิท และผ่านทาง Port 2 (PA0-PA7) ของ IC 8255 เป็นบิทข้อมูล 8 บิท (การเขียนโปรแกรมควบคุมและการใช้งานดูจากรายละเอียดทฤษฎีในบทที่ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 Sinewave Genalate

วงจรส่วนนี้ทำหน้าที่ผลิตสัญญาณ Sinewave เพื่อใช้เป็นสัญญาณในการเรียกเครื่อง Pager แต่ละตัว ซึ่ง Pager แต่ละเครื่องนั้นจะมีหมายเลขที่ไม่เหมือนกัน ดังนั้นการส่งสัญญาณที่ไปเรียกนั้นก็จะมีความถี่ที่ไม่เหมือนกัน ความถี่ที่ใช้ทั้งหมดจะมีอยู่ทั้งหมด 30 ความถี่ ดังที่กล่าวไว้ในบทที่ 2 ในวงจรนี้จะใช้วงจรดิจิตอลแปลงสัญญาณเป็นอนาล็อกขนาด 8 บิท โดยใช้วงจร แล็กเดอร์ เป็นวงจร DAC โดยการกำเนิดสัญญาณดิจิตอลจาก CPU โดยผ่านทาง Port 2 (PC0-PC7) ของ IC 8255 บนบอร์ด ANT32 ต่อเข้ากับ Buffer โดยใช้ IC 74LS244 เพื่อขยายกระแสให้สูงขึ้น จากนี้จะผ่านวงจรแล็กเดอร์ เพื่อแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก จากสัญญาณอนาล็อกที่ได้จะนำไปผ่านวงจรกรองความถี่ต่ำผ่าน เพื่อให้ได้สัญญาณความถี่ที่เป็นรูปคลื่น Sinewave เพื่อนำไปใช้งานเป็นสัญญาณ Two-Tone และนำไป Mod เพื่อส่งออกอากาศต่อไป โดยมีการต่อวงจรใช้งานดังรูปที่ 3.7



รูปที่ 3.6 วงจร LCD Module

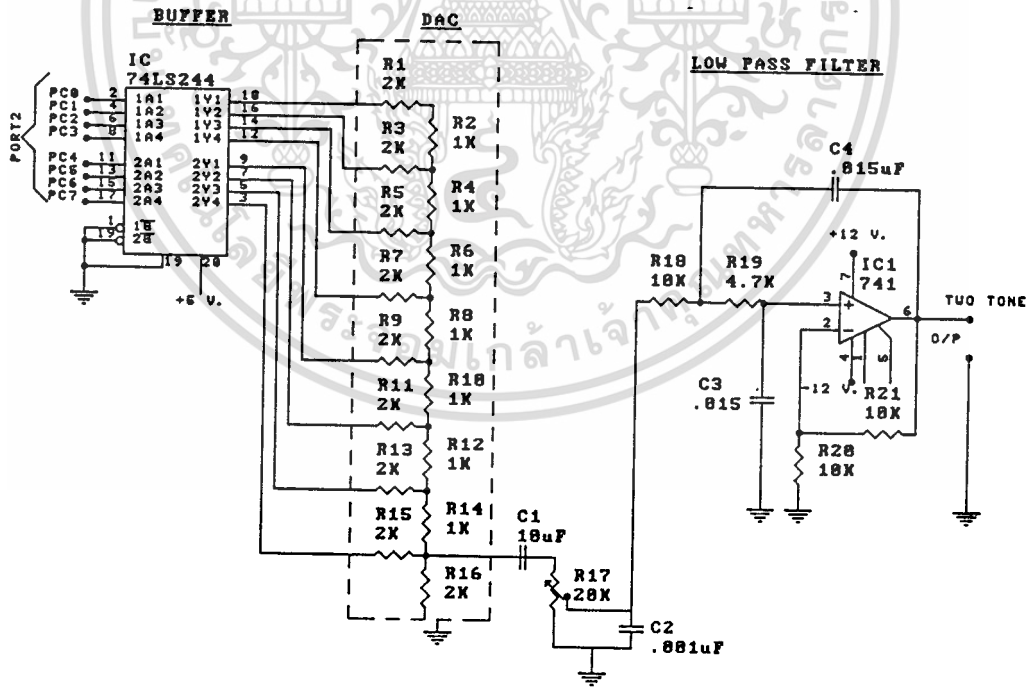
3.1.7 CPU (Center Processin Unit)

ในส่วนนี้เราใช้บอร์ด ANT32 ของบริษัท ศิลารีเสิร์ช จำกัด เป็นบอร์ดไมโคร-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

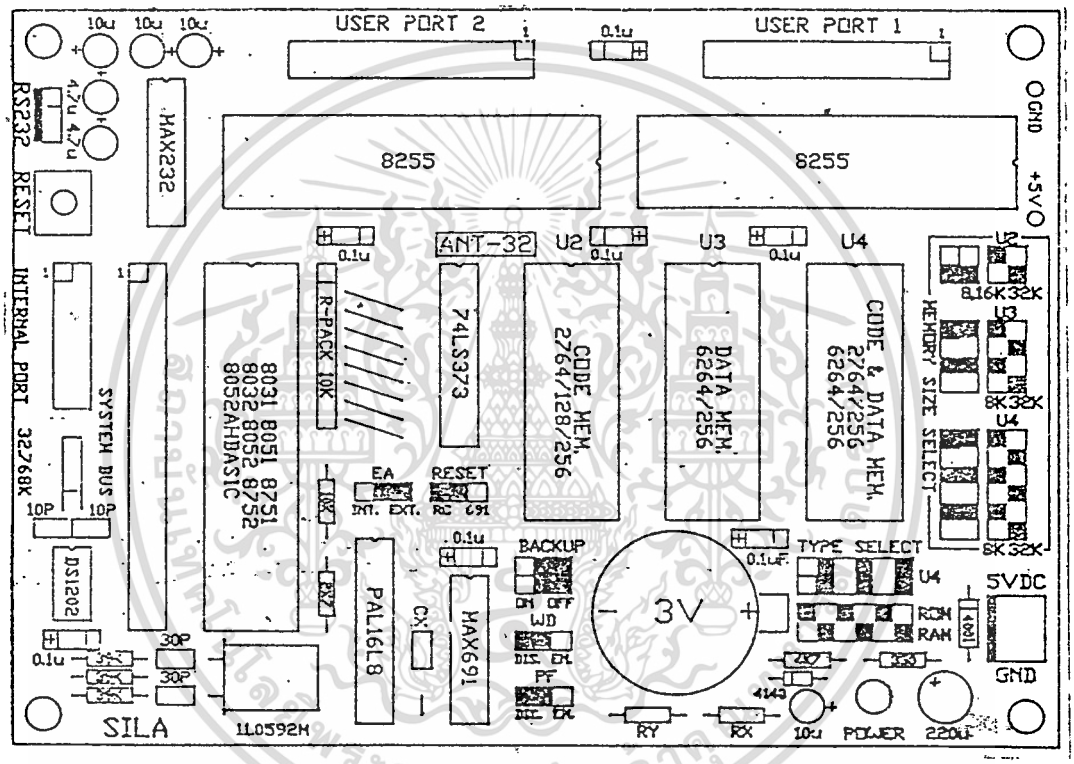
คอนโทรลเลอร์สำเร็จรูป ซึ่งเป็นบอร์ดที่นำไปใช้งานในลักษณะ Embedded Controller กล่าวคือ เป็นบอร์ดที่ถูกออกแบบมาเพื่อควบคุมงานโดยเฉพาะ โดยติดตั้งอยู่ในเครื่องจักรกล เครื่องใช้ไฟฟ้า รวมทั้งระบบอัตโนมัติต่างๆ ซึ่งบอร์ดนี้จะใช้ CPU เบอร์ 8032 ซึ่งอยู่ในตระกูล MCS51 ซึ่งได้ถูกนำมาประยุกต์ใช้เป็นตัวควบคุมในระบบเครื่องเรียกวิทยุติดตามตัวแบบอัตโนมัตินี้ ซึ่งเบอร์ ANT32 นี้ได้ถูกออกแบบมา ซึ่งประกอบด้วย Watchdog Timer, Battery Backup และ Power Fail Detector ซึ่งใช้ชิพ Max 691 วงจร Real Timer Clock ใช้ชิพ DS1202 และบอร์ดสื่อสารอนุกรมใช้ชิพ Max 232 รูปที่ 3.8 แสดงบอร์ดและตำแหน่งจัมพ์เปอร์, รูปที่ 3.9 แสดงรายละเอียดของคอนเนคเตอร์ และชิพเบอร์ต่างๆ ,รูปที่ 3.10 แสดงวงจรทั้งหมดของบอร์ด ANT -32 ,รูปที่ 3.11 แสดงลักษณะสัญญาณ RS232 จากชิพ MAX 232 และการต่อสาย, รูปที่ 3.12 แสดงการกำหนดตำแหน่ง Memory ของบอร์ด ANT-32

การพัฒนาโปรแกรมบนบอร์ด ANT- 32 นี้ใช้ภาษาแอสแซมบลี้ และภาษาซี โดยใช้โปรแกรม AVC51 ในการ COMPILE และ LINK โปรแกรมซึ่งการใช้ภาษาซีนั้นจะมีความยืดหยุ่นมากในการเขียนโปรแกรม



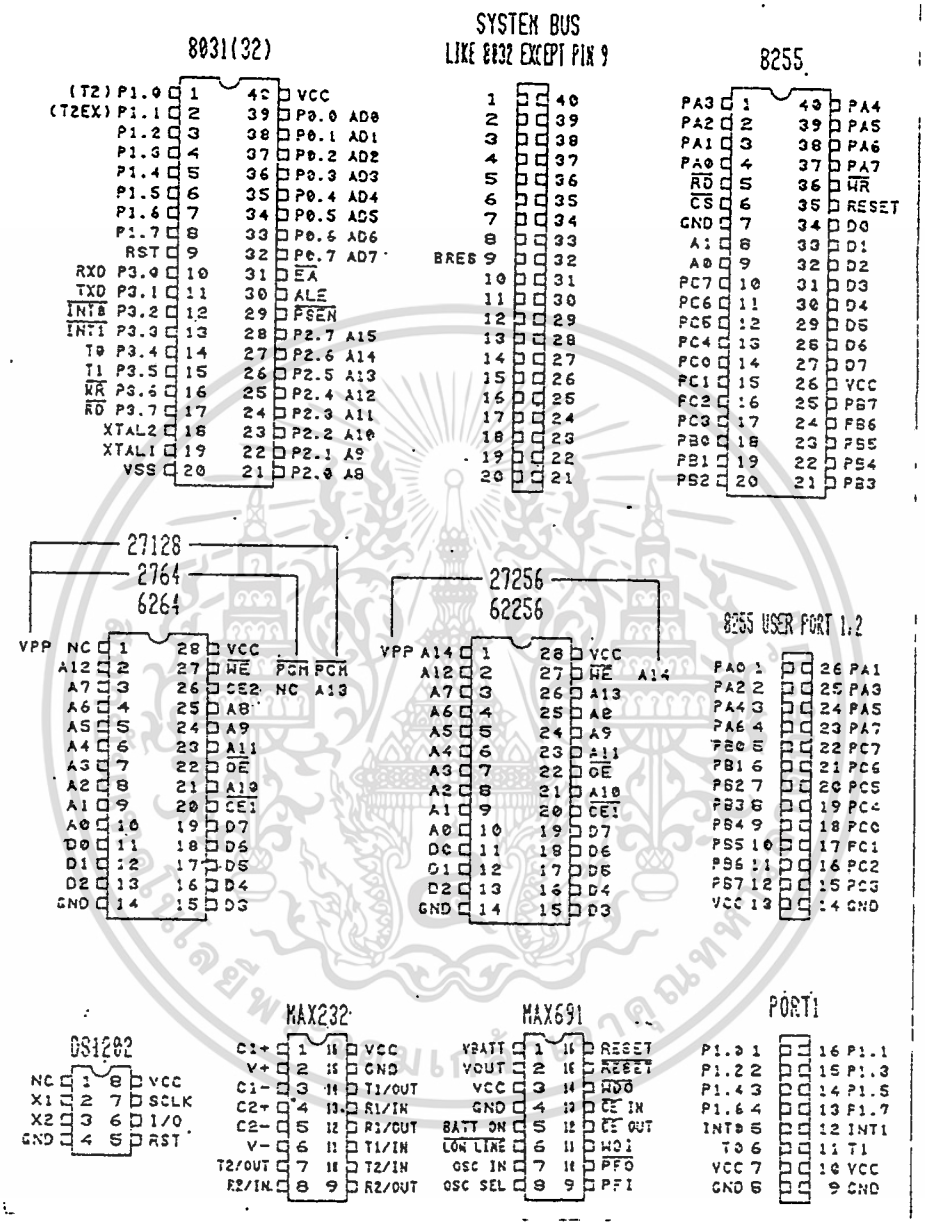
รูปที่ 3.7 วงจร Sinewave Generate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



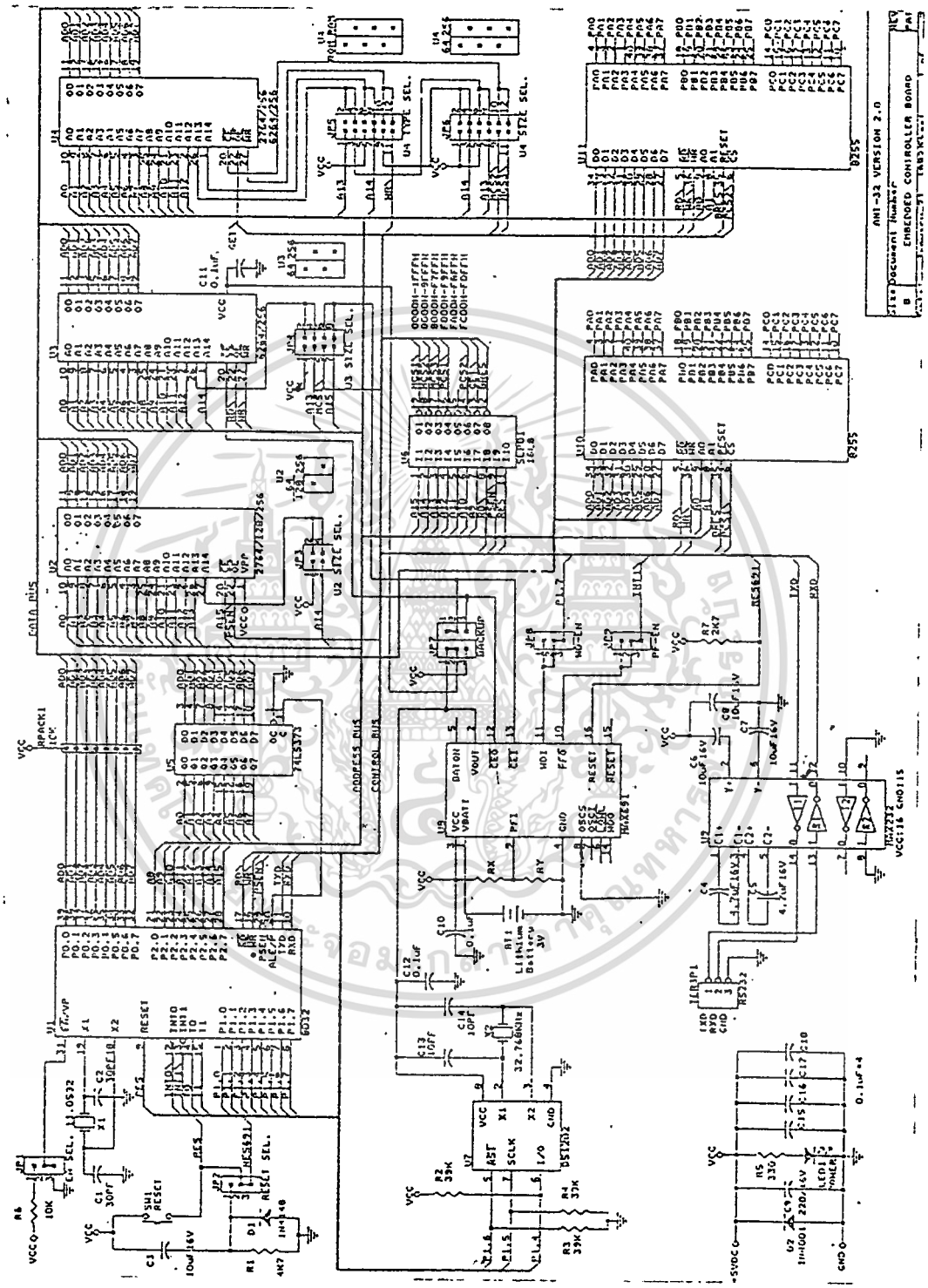
รูปที่ 3.8 ภาพแสดงบอร์ดและตำแหน่งจัมพ์เปอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



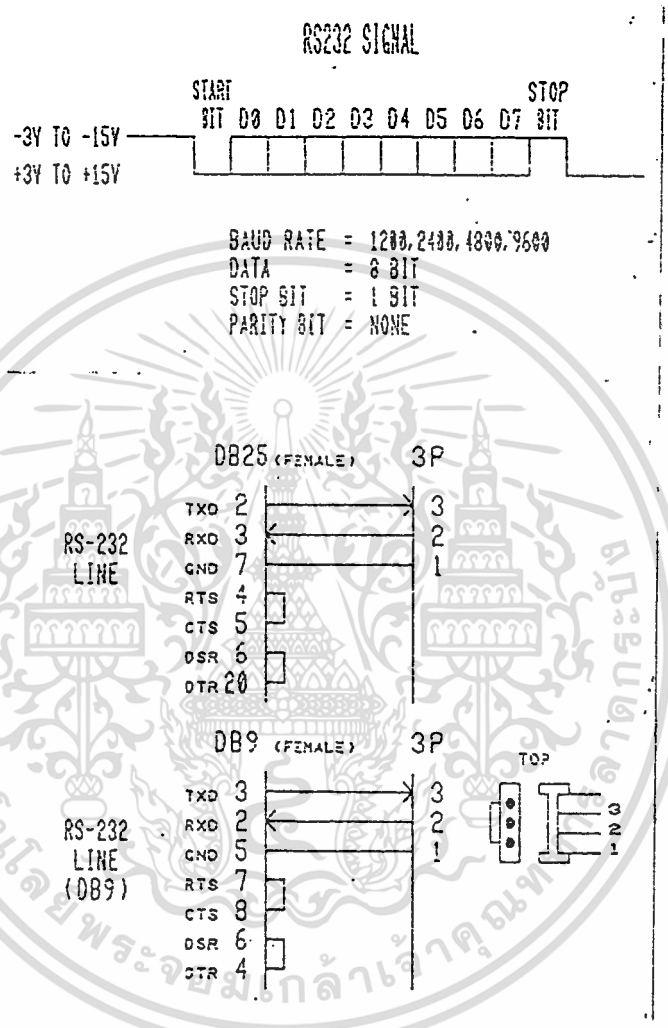
รูปที่ 3.9 ภาพแสดงรายละเอียดของคอนเนคเตอร์และชิพต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



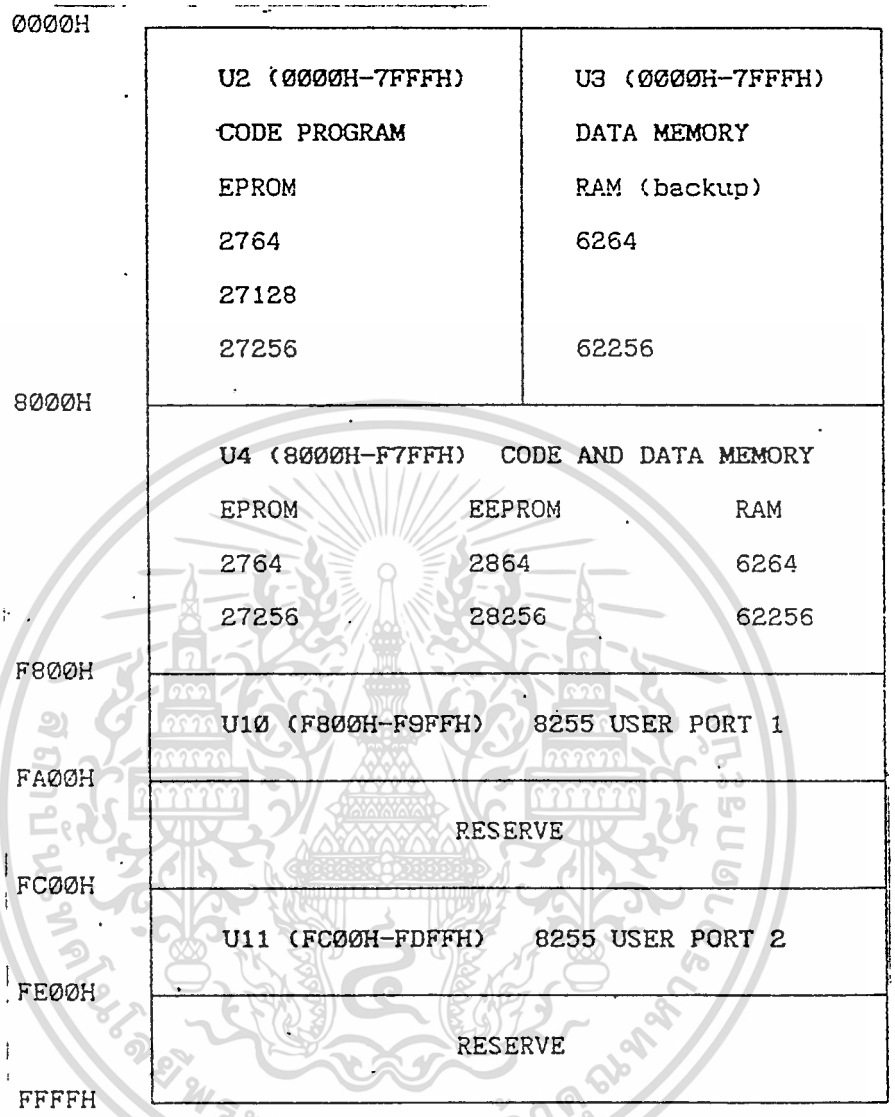
รูปที่ 3.10 ภาพแสดงวงจรทั้งหมดของบอร์ด ANT-32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ภาพแสดงลักษณะสัญญาณ RS 232 และการต่อสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ANT-32 MEMORY MAP

3.1.8 การโปรแกรม IC 8255 บนบอร์ด ANT32 ให้อยู่ในโหมดต่างๆ

8255 Programmable Peripheral Interface (PPI) เป็นชิพพอร์ทแบบขนาน สำหรับบอร์ด ANT32 ใช้ 8255 จำนวน 2 ตัว โดยมี User Port 1 และ 2 มีตำแหน่งแอดเดรส คือ

User Port 1 (U10) แอดเดรส F800H + 8255 offset addr = actual addr

Port A	ตำแหน่งแอดเดรส	F800H + 00H = F800H
Port B	ตำแหน่งแอดเดรส	F800H + 01H = F801H
Port C	ตำแหน่งแอดเดรส	F800H + 02H = F802H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode Port ตำแหน่งแอดเดรส F800H + 03H = F803H

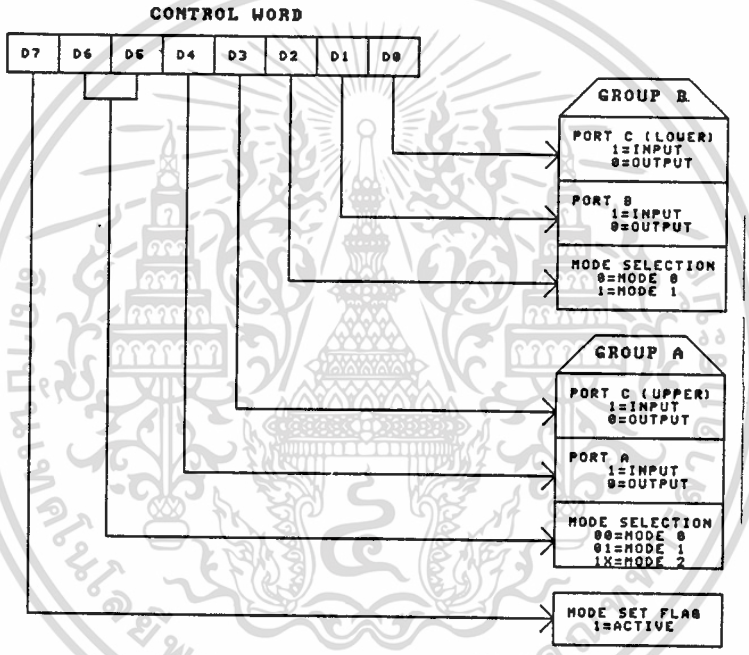
User Port 2 (U11) แอดเดรส FC00H + 8255 offset addr = actual addr

Port A ตำแหน่งแอดเดรส FC00H + 00H = FC00H

Port B ตำแหน่งแอดเดรส FC00H + 01H = FC01H

Port C ตำแหน่งแอดเดรส FC00H + 02H = FC02H

Mode Port ตำแหน่งแอดเดรส FC00H + 03H = FC03H



รูปที่ 3.13 แสดงลักษณะการเขียน Control Word

การใช้งานพอร์ท 8255 ต้องกำหนดโหมดการทำงานของพอร์ท A,B และ C ให้เป็นพอร์ทอินพุทหรือเอาท์พุท โดยการเขียนค่า Control Code ไปที่ Mode Port (Mode Port นี้สามารถเขียนได้เท่านั้นไม่สามารถอ่านได้) ในปริญญานิพนธ์นี้ เราใช้การทำงานในโหมด 0 โดยมีค่า Control Code ดังตารางที่ 3.2

ซึ่งเครื่องนี้ได้เซ็ท Control Code ดังนี้

User Port 1 = 98H

Port A = Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port B = Output

Port C บน = Input

Port C ล่าง = Output

User Port 2 = 80H

Port A = Output

Port B = Output

Port C บน = Output

Port C ล่าง = Output

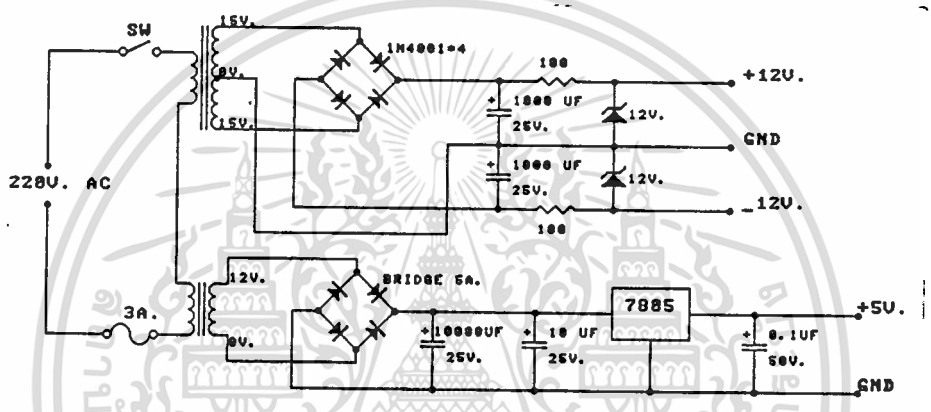
PORT A (PA0-PA7)	PORT C บน (PC4-PC7)	PORT B (PA0-PA7)	PORT C ล่าง (PA0-PA7)	Control Code (hex)
Output	Output	Output	Output	80H
Output	Output	Output	Input	81H
Output	Output	Input	Output	82H
Output	Output	Input	Input	83H
Output	Input	Output	Output	88H
Output	Input	Output	Input	89H
Output	Input	Input	Output	8AH
Output	Input	Input	Input	8BH
Input	Output	Output	Output	90H
Input	Output	Output	Input	91H
Input	Output	Input	Output	92H
Input	Output	Input	Input	93H
Input	Input	Output	Output	98H
Input	Input	Output	Input	99H
Input	Input	Input	Output	9AH
Input	Input	Input	Input	9BH

ตารางที่ 3.2 8255 Mode 0 Configulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.9 POWER SUPPLY

วงจรส่วนนี้ทำหน้าที่เป็นแหล่งจ่ายกำลังไฟฟ้ากระแสตรง ให้กับวงจรภายในเครื่องทั้งหมดซึ่งประกอบด้วยแหล่งจ่ายไฟฟ้าดังนี้ 5 VOLT DC, 12 VOLT DC โดยมีวงจรตามรูปที่ 3.14



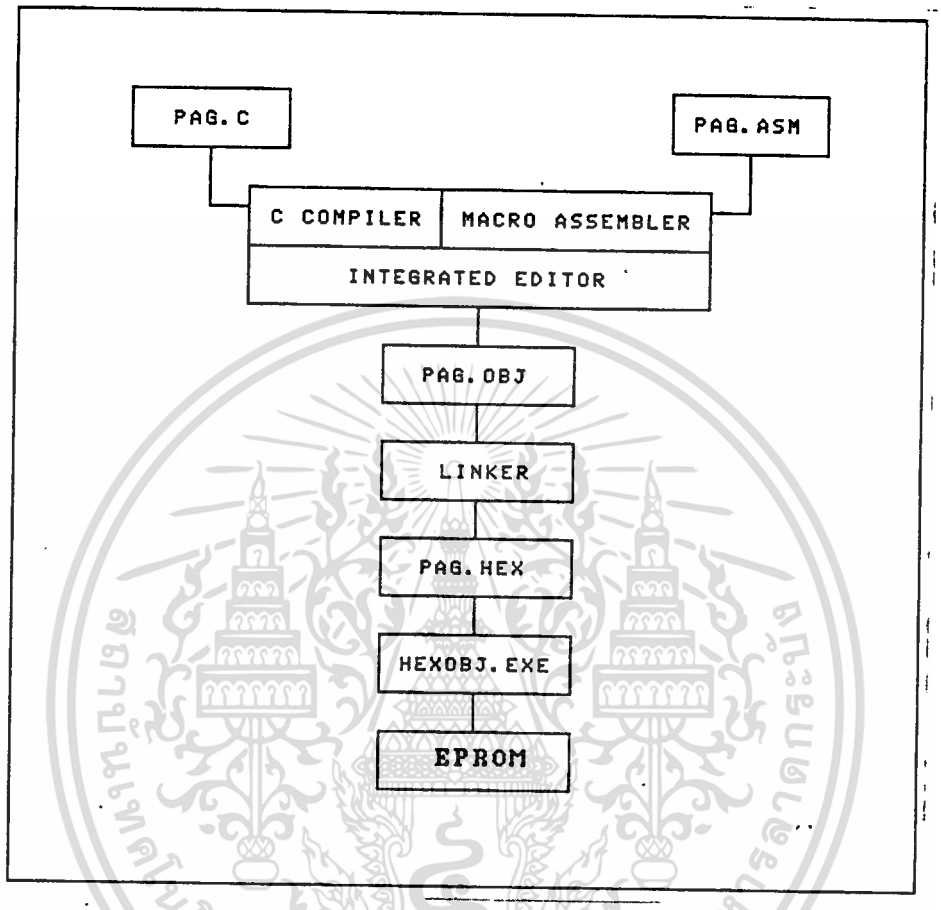
รูปที่ 3.14 วงจร POWER SUPPLY

3.2 การออกแบบ SOFT WARE

SOFT WARE ที่ใช้เป็นภาษา C ร่วมกับภาษา Assembly โดยใช้โปรแกรม AvCase 8051 C compiler เป็นโปรแกรม compile และ link โดยจะได้เป็น Hex Format File ซึ่งเป็น File ที่มีนามสกุลเป็น .HEX และเป็น Format ที่ใช้กับไมโครโปรเซสเซอร์ตระกูล MCS 51 การอัดโปรแกรมลง EPROM จะใช้โปรแกรม HEXOBJ.EXE ในการแปลง Hex Format File เพื่อให้ได้ File ที่สามารถอัดลง EPROM ได้

ภาษาที่ใช้เขียนโปรแกรมส่วนใหญ่ใช้ภาษา C ส่วนภาษา Assembly นั้นจะใช้ในการสร้างสัญญาน Sinewave ซึ่งต้องการความเร็วในการทำงานสูง ซึ่งนอกนั้นจะใช้ภาษา C ทั้งหมด ซึ่งจะแยกออกเป็น 2 File คือ PAG.C และ PA.ASM แล้วนำ File ทั้งสองมาทำการ compile และ link ด้วยโปรแกรม AvCase 8051 C compiler ซึ่งมีรูปแบบการทำงานดังรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 Avcase Integrate Environment

3.3 ระบบการทำงานและการใช้งานเครื่องเรียกวิทยุติดตามตัว ระบบโทนและวอยซ์แบบอัตโนมัติ

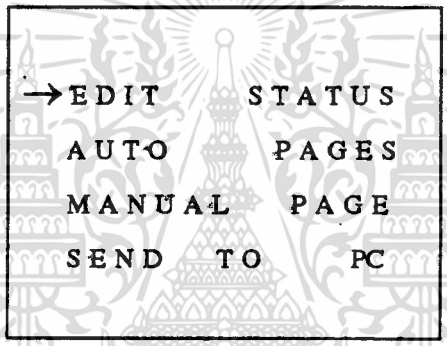
ระบบการทำงานของเครื่องทำตามเมนูการใช้ได้ 4 กรณี ดังนี้

1. EDIT_____STATUS
2. AUTO_____PAGES
3. MANUAL__PAGE
4. PRINT_____USED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

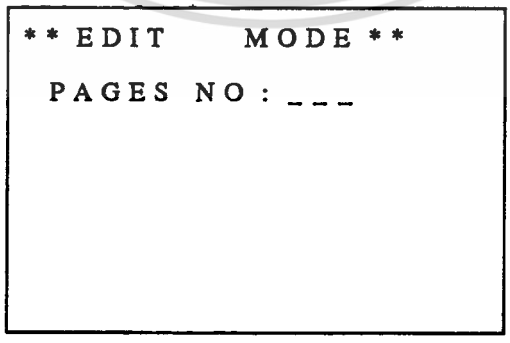
3.3.1 EDIT STATUS

เป็นการกำหนดสถานะของเครื่องวิทยุติดตามตัว ว่าจะให้สามารถเรียกได้หรือไม่ได้ ดังนั้นในกรณีที่เริ่มใช้เครื่องเรียกวิทยุติดตามตัวใหม่ โปรแกรมจะกำหนดให้ทุกเครื่องมีสถานะ: OFF ทั้งหมด เราจึงต้องมีการกำหนดสถานะของเครื่องวิทยุติดตามตัวที่ใช้บริการ ให้อยู่ในสถานะ ON โดยการเข้าสู่การทำงานของ EDIT MODE โดยกดคีย์ ESC ให้โปรแกรมการทำงานกลับสู่ MAIN MENU ดังรูปที่ 3.16



รูปที่ 3.16 MAIN MENU

จากนั้นใช้คีย์ เลื่อน ไปยังหัวข้อ EDIT STATUS แล้วกดคีย์ Enter ซึ่งเป็นการเข้าสู่การทำงานใน EDIT MODE จากนั้นจอภาพจะปรากฏดังรูป 3.17



รูปที่ 3.17 EDIT STATUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะปรากฏ Cursor กระพริบรอรับการป้อนหมายเลขเครื่องวิทยุติดตามตัวที่ต้องการกำหนดสถานะเป็น ON หรือ OFF สมมุติว่าป้อนหมายเลข "543" เมื่อเราป้อนหมายเลขที่จะเรียกจนครบ 3 หลักแล้ว โปรแกรมก็จะแสดงสถานะของเครื่องหมายเลขดังกล่าวออกมา พร้อมแสดงจำนวนครั้งของการใช้งานที่บรรทัดสุดท้าย (ซึ่งเป็นจำนวนครั้งที่ผู้ใช้บริการเรียกเครื่องวิทยุติดตามตัว โดยจะเพิ่มค่าขึ้นทุกครั้งที่มีการเรียกติดต่อ) ดังรูปที่ 3.18

```
** EDIT MODE **
PAGES NO : 543
STATUS   : OFF
USED     : 0
```

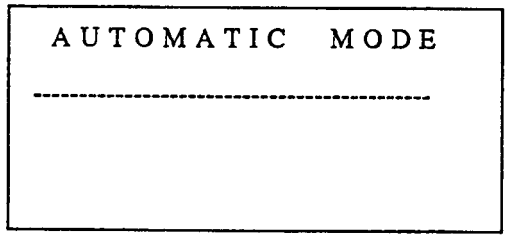
รูปที่ 3.18 สถานะและจำนวนการใช้งานของหมายเลข

จากนั้นจะเป็นขั้นตอนของการกำหนดสถานะโดยการกดคีย์ status บนแป้นคีย์บอร์ด การกดคีย์ status 1 ครั้งก็จะเปลี่ยนสถานะจาก ON เป็น OFF ขณะนี้เราก็สามารถกำหนดสถานะของเครื่องวิทยุติดตามตัวได้แล้ว ซึ่งขณะที่โปรแกรมทำงานอยู่ในขั้นตอนที่เราสามารถใช้คีย์ลูกศรขึ้นและลง (↑ , ↓) เพิ่มหมายเลขเครื่องวิทยุติดตามตัวหรือลดหมายเลขเครื่องวิทยุติดตามได้ได้ครั้งละ 1 เพื่อแสดงและแก้ไขสะถานะของหมายเลขวิทยุติดตามตัวหมายเลขอื่น ๆ เมื่อเราจบการกำหนดสถานะให้เครื่องวิทยุติดตามตัวแล้ว เราจะต้องกลับไปยัง MAIN MENU ด้วยการกดคีย์ Esc เพื่อเข้า MODE การทำงานอื่นต่อไป

3.3.2 AUTO PAGES

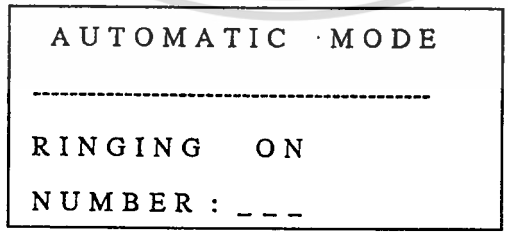
เป็นโปรแกรมที่ให้เครื่องเรียกวิทยุติดตามตัวทำงานโดยอัตโนมัติ ซึ่งอาศัยสัญญาน Ringing จากคู่สายโทรศัพท์เป็นสัญญานสั่งให้โปรแกรมเริ่มทำงาน โดยเมื่อเราอยู่ที่ MAIN MENU ต้องเลื่อน Cursor (→) มายังเมนูหัวข้อ AUTO PAGES แล้วกด Enter เข้าสู่ระบบ AUTOMATIC MODE ซึ่งหน้าจอจะปรากฏดังรูปที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 Automatic Mode

เมื่อมีผู้ใช้บริการเรียกเข้ามาก็จะมีการส่ง Ringing Tone เข้ามายังวงจร External Line Interface สัญญาน Ringing จะผ่านวงจร Rectifier ออกมาเป็นสัญญานดีซีพัลส์ ผ่านเข้ามารับ Opto Transister ทำให้ขาอินพุทของ IC 555 เปลี่ยนสถานะจากบวกเป็นลบ ตามสัญญาน Ringing ที่เข้ามา เมื่อวงจรโมโนสเตเบิ้ลถูกทริก จะทำให้เอาท์พุท (ขา 3 IC 555) ซึ่งปกติจะเป็นลอจิก "0" เปลี่ยนสถานะไปเป็นลอจิก "1" เพียงชั่วระยะเวลาหนึ่ง เอาท์พุทของวงจรโมโนสเตเบิ้ลก็จะเปลี่ยนสถานะกลับมาเป็นลอจิก "0" เหมือนเดิม สัญญานนี้ก็จะถูกส่งไปยัง CPU โดยผ่านทางพอร์ท 1 (PA0) ของ 8255 เพื่อให้ CPU ทราบว่ามีการติดต่อเข้ามา จากนั้น CPU จะสั่งต่อวงจรแล้วก็เสมือนกับการยกหูรับโทรศัพท์ที่เรียกเข้ามา (สัญญาน Ringing จะหายไป) จากนั้น CPU ก็จะไปสั่งให้ IC T6668 ตอบรับโทรศัพท์ โดยขั้นแรก CPU จะส่งสัญญานไปเลือกแชนแนลของข้อความก่อนจำนวน 4 บิท โดยส่งผ่าน 8255 ทางพอร์ท 1 (Pb3-Pb6) ซึ่งในขั้นนี้จะเลือกข้อความช่องที่ 1 "กรุณากดหมายเลขที่ท่านต้องการเรียก" โดย CPU จะส่งสัญญานไป START IC T6668 ให้ตอบข้อความในช่องที่ 1 โดยสัญญาน START จะส่งผ่านทางพอร์ท 1 (Pb7) ของ 8255 ซึ่งในขณะเดียวกับจอภาพก็จะแสดงว่ามีการเรียกเข้ามาพร้อมกับมี CURSOR กระพริบเพื่อรอการแสดงหมายเลขที่ผู้ใช้บริการกดเข้ามา ดังรูปที่ 3.20



รูปที่ 3.20 Automatic Mode ขณะยังไม่ป้อนหมายเลข

เมื่อผู้ใช้บริการกดหมายเลขที่จะเรียก(โดยใช้คีย์โทรศัพท์) สัญญานที่ผ่านเข้ามาจะถูกถอดรหัสเป็นสัญญานไบนารี 4 บิท โดยวงจร DTMF DECORDER วงจร DTMF นี้จะส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณพัลซ์ที่ขาสโตป (ขา 15) ทุกครั้งที่มีการกดคีย์พร้อมกับส่งสัญญาณรหัสไบนารี 4 บิท เพื่อแจ้งให้ CPU ทราบว่ามีการกดคีย์หมายเลขเข้ามาแล้ว เมื่อผู้ใช้กดหมายเลขเข้ามาจนครบ 3 หลัก CPU ก็จะนำหมายเลขที่ผู้ใช้ต้องการติดต่อมาตรวจเช็คสถานะ และจอภาพก็จะแสดงสถานะของเครื่องหมายเลขที่ผู้ใช้ติดต่อเข้ามาดังรูปที่ 3.21

```
AUTOMATIC MODE
---------------------------------
STATUS : ON
NUMBER : 543
```

รูปที่ 3.21 Automatic Mode ขณะป้อนหมายเลข

ถ้าสถานะเครื่องที่เรียกเป็น ON CPU ก็จะสั่งให้รีเลย์ (Ry3) ต่อวงจร ซึ่งเป็นการสั่งให้ทำการกดคีย์เครื่องส่งวิทยุ (ซึ่งมีความถี่ในการส่งออกอากาศ 170.900 MHz เตรียมพร้อมที่จะส่งสัญญาณ Tone A และ Tone B ออกอากาศ เพื่อเรียกเครื่องวิทยุติดตามตัว) ซึ่งสั่งให้รีเลย์ (Ry3) ต่อวงจรโดยผ่าน 8255 ทาง Port 1(PB2) จากนั้น CPU ก็นำหมายเลขที่เรียกเข้ามา มาเปิดตารางโค้ด Plan C เพื่อกำหนดความถี่ของ Tone A และ B เมื่อหาความถี่ได้แล้ว CPU ก็จะผลิตความถี่ Tone A เป็นเวลา 1 วินาที แล้วหยุดผลิต 0.25 วินาที แล้วผลิต Tone B เป็นเวลา 3 วินาที จะได้สัญญาณดิจิตอลขนาด 8 บิท ออกมาทางพอร์ท 2 (PC0-PC7) ของ 8255 จากนั้นจะนำสัญญาณไปผ่าน Buffer,DAC และ LPF เพื่อสร้างสัญญาณ Sinewave และจะได้สัญญาณ Sinewave ของ Tone A และ B ทางขา 6 ของ IC 741 แล้วนำสัญญาณ Tone A และ B นี้ไปเข้ายัง Mic ของเครื่องส่งเพื่อส่งออกอากาศ ซึ่งขณะนี้เครื่องรับวิทยุติดตามตัวหมายเลขที่ผู้ใช้เรียกจะมีสัญญาณดังขึ้น (แสดงว่ามีผู้เรียกเข้ามา) จากนั้น CPU ก็จะสั่งให้ IC T6668 ตอบโทรศัพท์อีกครั้งโดยเลือกข้อความในช่วงที่ 2 "กรุณาพูดข้อความท่านมีเวลาพูด 20 วินาที" ซึ่งเมื่อ T6668 ส่งข้อความจนหมดแล้ว ก็จะส่งสัญญาณไปให้ CPU ทราบว่าพูดข้อความจบแล้ว (ทางขา 41 (EOS)) (ซึ่งในขณะที่ IC T6668 เล่นกลับอยู่ ลอจิกที่ขานี้จะมีสถานะเป็น "0" เมื่อสิ้นสุดข้อความที่ขานี้จะมีสถานะเป็น "1") ซึ่งต่อกับ CPU โดยผ่าน 8255 พอร์ท 1(PA6) จากนั้น CPU ก็จะมาสั่งให้รีเลย์ (Ry2) ต่อวงจรโดยผ่าน 8255 พอร์ท 1 (PB1) ซึ่งก็จะทำให้ผู้ที่ต้องการเรียกวิทยุติดตามตัว พูดข้อความเข้ามายังคู่สายโทรศัพท์ข้อความนั้นก็จะเข้ามายังขด Primary ของหม้อแปลง T1 และออกไปยังขด Secondary ของหม้อแปลง T1 ผ่าน C 1μF และต่อไปยัง Mic ของเครื่องส่งวิทยุออกอากาศไปยังเครื่องรับวิทยุติดตามตัว เครื่องรับก็จะรับข้อความที่ผู้ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พูดมาในโทรศัพท์ ซึ่งจากการตอบข้อความในช่องที่ 2 จบลง CPU ก็จะเริ่มจับเวลาเมื่อครบ 20 วินาที CPU ก็จะตัดระบบการติดต่อทั้งหมด แล้วเพิ่มจำนวนครั้งของการใช้วิทยุติดตามตัวขึ้นไปอีก 1 ครั้ง และทำการ Reset ระบบให้กลับสู่การเริ่มต้นของ MODE AUTOMATIC ใหม่ เพื่อรอรับการติดต่อเข้ามาใหม่ (รอรับสัญญาณ Ringing)

ถ้าตรวจสอบสถานะของเครื่องวิทยุฯ ที่ถูกเรียกพบว่าเป็น OFF CPU ก็จะส่งสัญญาณไปเลือกแชนแนลข้อความของ IC T6668 ช่องถี่ 3 "ไม่สามารถให้บริการหมายเลขที่ท่านต้องการติดต่อได้" ตอบกลับไปยังผู้ใช้บริการที่ติดต่อเข้ามา แล้วทำการ Reset ระบบให้กลับสู่การเริ่มต้นของ MODE AUTOMATIC ใหม่

ขั้นตอนการทำงานที่กล่าวมาเป็นขั้นตอนการทำงานของเครื่องในหัวข้อเมนู AUTO PAGES ซึ่งการยกเลิกการทำงานในโหมดนี้ทำได้โดยกดคีย์ Esc โปรแกรมการทำงานก็จะกลับสู่ MAIN MENU (ปกติจะตั้งการทำงานไว้ที่โหมดนี้ตลอดเวลานอกจากจะไปเข้าโหมดอื่นชั่วคราว)

3.3.3 MANUAL PAGE

การทำงานในโหมดนี้จะเป็นระบบที่เรียกวิทยุติดตามตัว โดยใช้คีย์บอร์ดของเครื่องในการกำหนดหมายเลขที่จะเรียก เมื่อเราเข้าสู่ Mode แล้วจะปรากฏข้อความบนจอภาพดังรูปที่ 3.22

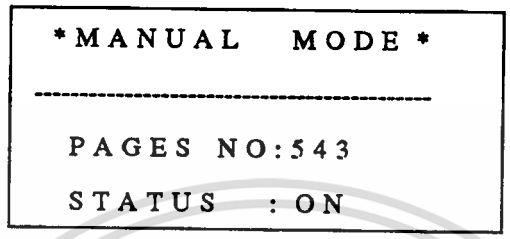
```
*MANUAL  MODE *
--------------------------------
 PAGES NO:___
--------------------------------
```

รูปที่ 3.22 Manual Mode ขณะยังไม่ป้อนหมายเลข

ขณะเดียวกัน CPU ก็จะสั่งให้รีเลย์ (Ry1) ต่อวงจร ซึ่งเป็นเสมือนการยกหูโทรศัพท์ไว้ เพื่อป้องกันการติดต่อโทรศัพท์เข้ามา ซึ่งเมื่อมีผู้ใช้บริการต่อโทรศัพท์เข้ามาในขณะนี้ก็จะได้รับสัญญาณสายไม่ว่าง (Busy Tone) จากชุมสายโทรศัพท์กลับไป จากนั้น CPU ก็จะมารอรับการกดหมายเลขจากคีย์บอร์ดของเครื่องโดยจะแสดงภาพการกระพริบของ CURSOR อยู่บนจอภาพในบรรทัดที่ 3 เพื่อรอแสดงหมายเลขจากการกดคีย์บอร์ด เมื่อกดหมายเลขที่ต้องการเรียกจากคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดครบ 3 หลักแล้ว CPU จะนำหมายเลขมาตรวจสอบสถานะว่าเป็น ON หรือ OFF เมื่อกดหมายเลขครบแล้ว จอภาพจะแสดงสถานะให้ทราบดังรูปที่ 3.23



รูปที่ 3.23 Manual Mode ขณะป้อนหมายเลข

ถ้าหมายเลขที่เรียกมีสถานะเป็น ON CPU ก็จะทำการเพิ่มจำนวนการเรียกวิทยุหมายเลขนี้ขึ้นไปอีก 1 ครั้ง CPU ก็จะสั่งให้รีเลย์ (Ry3) ต่อวงจรซึ่งเป็นการกดคีย์ (PTT) ของเครื่องส่ง จากนั้น CPU จะนำหมายเลขที่เรียบเข้ามามาเปิดตารางโค้ด Plan C เพื่อกำหนดความถี่ของ TONE A และ B เมื่อ CPU กำหนดความถี่ได้แล้ว CPU ก็จะผลิตความถี่ TONE A เป็นเวลา 1 วินาที หยุดผลิต 0.25 วินาที และผลิตความถี่ TONE B เป็นเวลา 3 วินาที ซึ่งได้สัญญาณดิจิตอลขนาด 8 บิท ออกมาทางพอร์ท 2 (PC0-PC7) ของ 8255 จากนั้นจะนำสัญญาณไปผ่าน Buffer, DAC และ LPF เพื่อสร้างสัญญาณ Sinewave และจะได้ความถี่ TONE A และ B เป็นสัญญาณ Sinewave นำไป Mod เข้ากับเครื่องส่งทาง Microphone เพื่อส่งออกอากาศ ขณะนี้เครื่องรับวิทยุติดตามตัวหมายเลขที่ถูกเรียกจะมีสัญญาณดังขึ้น (แสดงว่ามีการเรียกเข้ามา) จากนั้น CPU ก็จะสั่งให้รีเลย์ (Ry2) ต่อวงจรทางพอร์ท 1 (Pb1) การทำงานใน Mode นี้เราจะต้องเลื่อน Select Switch ไปยังตำแหน่ง Mic เนื่องจากเราจะต้องพูดเข้า Mic เพื่อส่งข้อความที่ต้องการไปยังเครื่องรับวิทยุติดตามตัวที่ต้องการติดต่อ เมื่อส่งข้อความเรียบร้อยแล้วต้องการยกเลิกก็กดคีย์ Esc CPU ก็จะ Reset ระบบทั้งหมดแล้วกลับไปรอรับหมายเลขที่คีย์บอร์ดใหม่จอภาพก็จะปรากฏดังรูปที่ 3.22 ถ้าเราต้องการเรียกเครื่องวิทยุติดตามตัวหมายเลขอื่น ก็ทำตามขั้นตอนที่กล่าวมาถ้าต้องการจะกลับไปที่ MAIN MENU ก็กด Esc อีกครั้ง โปรแกรมก็จะกลับไปสู่ MAIN MENU เพื่อรอเลือกการทำงานใน Mode อื่นต่อไป

ถ้าหมายเลขที่เรียกมีสถานะเป็น OFF โปรแกรมก็จะแสดงสถานะเป็น OFF ให้ทราบทางจอภาพ ดังรูปที่ 3.23 และจะแสดงค้างไว้อย่างนั้นจนกว่าจะมีการกดคีย์ Esc เพื่อกลับไปเริ่มรับหมายเลขจากคีย์บอร์ดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 PRINT USED

การทำงานในโหมดนี้ เป็นการแสดงจำนวนครั้งของการใช้งาน ของเครื่องวิทยุติดตามตัว ออกแสดงผลทางเครื่องพิมพ์โดยการติดต่อกับเครื่องพิมพ์ทางพอร์ท RS232 ซึ่งใช้คอนเน็คเตอร์ DB25 ต่อกับเครื่องพิมพ์ที่มีพอร์ทแบบอนุกรม RS232 โดยเครื่องพิมพ์จะพิมพ์ หมายเลขเครื่อง และจำนวนครั้งของการใช้เครื่องเฉพาะเครื่องที่มีสถานะเป็น ON เท่านั้น เมื่อเราเข้าสู่การทำงานในโหมดนี้ โปรแกรมจะรอรับคีย์บอร์ด ถ้ากดคีย์ Esc จะยกเลิกแล้วกลับเข้าสู่ MAIN MENU แต่ถ้ากดคีย์ Enter โปรแกรมก็จะทำงานต่อไป ซึ่งจอภาพจะปรากฏการใช้คีย์ดัง รูปที่ 3.24

```
**********************
*   SEND  TO  PC     *
**********************
```

รูปที่ 3.24 โหมดส่งข้อมูลสู่ PC

เมื่อกดคีย์ Enter บนบอร์ด CPU ก็จะส่งข้อมูลออกมายังขา TXD (ขา 11) ผ่าน IC MAX232 ซึ่งเป็น IC พอร์ทอนุกรม 2 ชุด แต่บนบอร์ด ANT-32 นั่นจะใช้ชุดเดียว จากนั้นจะได้เอาท์พุทที่เป็นสัญญาณ RS232 ทางขา 14 ของ IC MAX 232 ส่งไปยังเครื่องพิมพ์ที่มีพอร์ท RS232 เมื่อเครื่องพิมพ์ได้รับสัญญาณทางพอร์ทอนุกรมก็จะพิมพ์ข้อมูลออกดังตัวอย่างในรูปที่ 3.25

เมื่อ CPU ส่งข้อความมาพิมพ์จนหมดแล้ว CPU ก็จะ Reset ระบบแล้วกลับไปสู่โปรแกรม MAIN MENU เพื่อรอรับการเรียกใช้งานในเมนูอื่นต่อไป

3.3.5 โหมดการ CLEAR ข้อมูล

การทำงานในโหมดนี้จะเป็นการ CLEAR สถานะของเครื่องและการ CLEAR จำนวนครั้งของการใช้งาน ซึ่งมีอยู่ 2 ลักษณะดังนี้

1. CLEAR สถานะของเครื่องทั้งหมด (100-999) ให้เป็น OFF พร้อมทั้ง CLEAR จำนวนครั้งของการใช้เครื่องแต่ละเลขหมายให้เป็นศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. CLEAR เฉพาะจำนวนครั้งของการใช้เครื่องแต่ละเลขหมายให้เป็นศูนย์

AUTOMATIC PAGER CALL SYSTEM

NUMBER	USED
100	5
160	10
245	10
543	10
699	10
800	2

รูปที่ 3.25 ตัวอย่างข้อมูลที่ส่งขึ้นไปยังเครื่องคอมพิวเตอร์

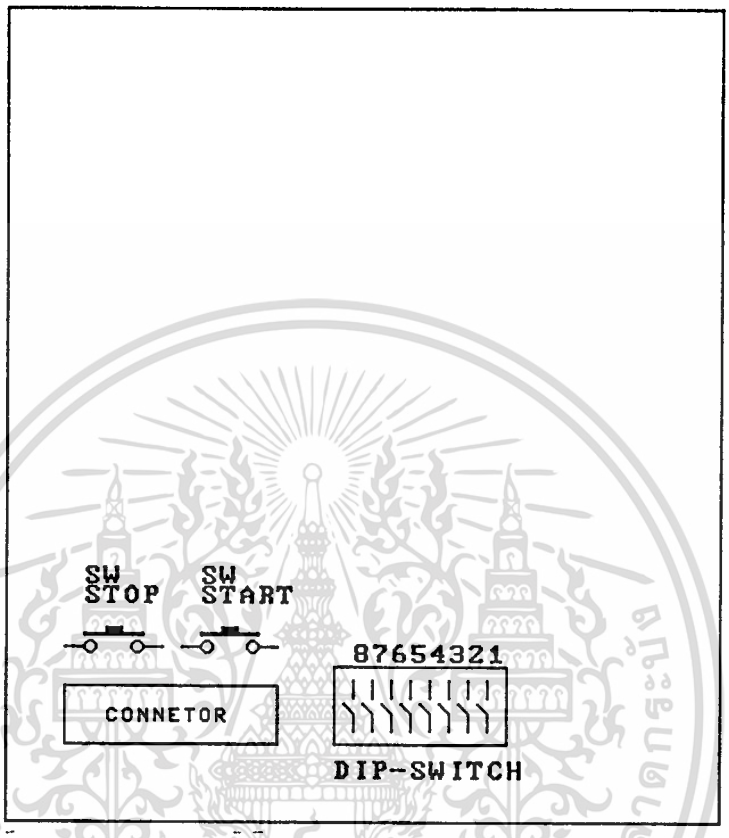
การเข้าสู่การ CLEAR วิธีที่ 1 ทำได้โดยปิดเครื่องเรียกวิทยุติดตามตัว กดคีย์ Esc ที่คีย์บอร์ดค้างไว้ จากนั้นเปิดเครื่องเรียกวิทยุติดตามตัว เมื่อเวลาผ่านไป 3 วินาที ให้ปล่อยคีย์ Esc ที่กดไว้โปรแกรมก็จะเข้าสู่ MAIN MENU ซึ่งขณะนี้สถานะของเครื่องและจำนวนครั้งของการใช้งานของทุกเครื่องจะถูก CLEAR เป็น OFF และศูนย์ครั้งตามลำดับ เป็นการเสร็จสิ้นการ CLEAR ข้อมูลในวิธีแรก

การเข้าสู่การ CLEAR วิธีที่ 2 ทำได้โดยเมื่อโปรแกรมอยู่ที่ MAIN MENU ให้กดคีย์ Clear ที่คีย์บอร์ด จากนั้นโปรแกรมจะมารอรับคีย์เพื่อสั่งงานต่อไป ถ้ากดคีย์ Esc โปรแกรมก็จะยกเลิกการทำงานในโหมดนี้ ไม่มีการ CLEAR จำนวนครั้งของการใช้งานและกลับสู่ MAIN MENU แต่ถ้ากดคีย์ Enter โปรแกรมก็จะทำการ CLEAR จำนวนครั้งของการใช้งานให้เป็นศูนย์ทุกเครื่อง เมื่อ CLEAR เสร็จแล้วโปรแกรมก็จะกลับสู่ MAIN MENU เพื่อรอรับการทำงานอื่นต่อไป

3.3.6 การบันทึกข้อความลงในวงจร Play & Record

เป็นการบันทึกข้อความที่ต้องการตอบรับโทรศัพท์กลับไปยังผู้ที่ติดต่อเข้ามาโดยเครื่องนี้มีการบันทึกทั้งหมด 4 ข้อความ 4 ช่อง เก็บสัญญาณโดยมีขั้นตอนการบันทึกข้อความอยู่ 7 ขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 แสดงตำแหน่ง DIP SWITCH และ SWITCH บนแผ่นวงจรที่ใช้งาน

1. เลื่อน Select Switch บนหน้าปัทม์ของเครื่องไปที่ตำแหน่ง Mic
2. เลื่อน DIP-SW เลข 1 มายังตำแหน่ง ON
3. เลื่อน DIP-SW เลข 8 มายังตำแหน่ง ON
4. กด SW. Start (จะสังเกตเห็นว่า LED บนวงจรบันทึกเสียงพูดจะดับ)
5. พูดข้อความของช่องที่ 1 "กรุณากดหมายเลขที่ท่านต้องการเรียก"
6. กด SW.Stop (สังเกตุเห็นว่า LED บนวงจรบันทึกเสียงพูดจะติด)
7. เลื่อน DIP-SW 1,8 มายังตำแหน่ง OFF

ซึ่งบนแผงวงจรจะมีตำแหน่ง DIP-SW อยู่ 8 ตำแหน่งซึ่งแต่ละตำแหน่งจะต่อกับ IC T6668 และมีหน้าที่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่ง DIP-SW	IC T6668	หน้าที่
1	DO	DO - D3 ใช้กำหนดช่องเก็บ
2	D1	ข้อมูล เก็บได้ทั้งหมด 16 ช่อง
3	D2	0000 - 1111
4	D3	
5	ไม่ได้	bit rate0,1 ใช้กำหนดความเร็ว
6	bit rate0	ของเสียงพูด
7	bit rate1	ใช้กำหนดการบันทึกหรือเล่น
8	$\overline{WR}$	กลับ

ตารางที่ 3.3 ตำแหน่ง DIP-SW บนแผงวงจร

จากขั้นตอนทั้ง 7 ที่กล่าวมาเป็นการบันทึกข้อความลงหน่วยความจำช่องที่ 1 การเปลี่ยนช่องเก็บข้อมูลทำได้โดยกำหนด DIP-SW. ที่ 1-4 ซึ่งมีตำแหน่งดังนี้ (โดย 0 = DIP-SW OFF และ 1 = DIP-SW ON)

DIP - SW				เก็บข้อความ
4	3	2	1	
0	0	0	0	ช่องที่ 0
0	0	0	1	ช่องที่ 1
0	0	1	0	ช่องที่ 2
0	0	1	1	ช่องที่ 3
0	1	0	0	ช่องที่ 4
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
1	1	1	1	ช่องที่ 15

ตารางที่ 3.4 ตำแหน่งช่องข้อมูลDIP-SW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในการบันทึกข้อความช่องต่อไป ก็จะทำตามขั้นตอนดังกล่าว แต่ต้องเปลี่ยนตำแหน่ง DIP-SW. ในข้อ 2 ให้ตรงตามช่องที่จะใช้เก็บข้อมูลด้วยเมื่อบันทึกข้อความเรียบร้อยแล้ว จะต้องปรับ DIP-SW. ทุกตัวให้อยู่ตำแหน่ง OFF ยกเว้น DIP-SW7 จะอยู่ในตำแหน่ง ON และข้อความที่บันทึกได้จะเก็บอยู่ใน RAM โดยจะมีถ่านแบตเตอรี่ต่อ Backup เอาไว้ 4 ก้อน เพื่อมิให้ข้อความถูกลบไป แล้วปรับ Select Switch กลับมายังตำแหน่ง Tel ด้วย

** หมายเหตุ ไฟล์วชาร์จการทำงานของเครื่องอยู่ในภาคผนวก

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดลอง

4.1.1 การสร้างสัญญาณ Sinewave

ผลการทดลองในการสร้างสัญญาณ Sinewave เพื่อกำหนดความถี่ตามตาราง PlanC จำนวน 30 ความถี่ ซึ่งผลที่ได้จะมีความคลาดเคลื่อน 0 Hz จนถึง 0.8 Hz แต่ยังคงสามารถทำงานได้ โดยผลที่ได้ แสดงตามตารางที่ 4.1

CODE	ความถี่กลุ่ม A	ความถี่กลุ่ม B	ความถี่กลุ่ม C
0	312.7 Hz	330.6 Hz	569.3 Hz
1	339.6 Hz	349.1 Hz	600.5 Hz
2	358.6 Hz	368.4 Hz	634.5 Hz
3	378.7 Hz	388.9 Hz	669.5 Hz
4	399.7 Hz	410.8 Hz	707.5 Hz
5	422.1 Hz	433.6 Hz	746.6 Hz
6	445.8 Hz	457.8 Hz	788.9 Hz
7	470.4 Hz	483.7 Hz	833.1 Hz
8	496.8 Hz	510.6 Hz	879.3 Hz
9	524.5 Hz	539.3 Hz	928.9 Hz

ตารางที่ 4.1 แสดงผลค่าความถี่ของตาราง Plan C ที่ได้จากการทดลอง

4.1.2 การทดลองตั้งสถานะของเครื่อง

ผลการทดลองสามารถตั้งสถานะของเครื่องให้ ON และ OFF ได้โดยทดลองเรียกเครื่องรับวิทยุติดตามตัวหมายเลข 160 , 245 , 543 และ 699 ซึ่งผลการทดลองสามารถทำงานได้ตามที่กำหนดสถานะไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การทดลองในโหมด Automatic

เมื่อตั้งเครื่องให้อยู่ในโหมดอัตโนมัติ แล้วทำการทดลองเรียกเครื่องหมายเลข 543 โดยเมื่อได้รับสัญญาณตอบรับ "กรุณากดหมายเลขที่ต้องการเรียก" จบลงแล้วกดคีย์ 5..4..3 บนเครื่องโทรศัพท์ ขณะนี้เครื่องรับหมายเลข 543 จะมีสัญญาณดังขึ้น (เพื่อแสดงว่ามีการเรียกติดต่อเข้ามา) และเครื่องโทรศัพท์ที่เรียกเข้ามาจะได้รับสัญญาณตอบรับ"กรุณาพูดข้อความ ท่านมีเวลาพูด 20 วินาที" จากนั้นทดลองพูดข้อความผ่านคู่สายโทรศัพท์เพื่อส่งไปยังเครื่องรับ ซึ่งข้อความเสียงพูดที่ได้รับได้มีความชัดเจนดีมาก เป็นที่น่าพอใจ เมื่อเวลาผ่านไป 20 วินาที เครื่องก็จะกลับมาเริ่มต้นรอรับการเรียกจากโทรศัพท์ใหม่

4.1.4 การทดลองในโหมด Manual

เมื่อตั้งเครื่องให้ทำงานอยู่ในโหมด Manual แล้วทำการทดลองเรียกเครื่องหมายเลข 543 โดยกดหมายเลข 5..4..3 จากคีย์บอร์ดของเครื่อง เมื่อกดคีย์ 5..4..3แล้ว เครื่องรับจะมีเสียงดังขึ้น (เพื่อแสดงว่ามีการเรียกติดต่อเข้ามา) จากนั้นทดลองพูดข้อความผ่านไมโครโฟน (ปรับตำแหน่ง Select Switch ไปที่ Mic) เพื่อส่งข้อความไปยังเครื่องรับ ซึ่งข้อความเสียงพูดที่รับได้มีความแรงสัญญาณและความชัดเจนดีเป็นที่พอใจ เมื่อส่งข้อความหมดลง แล้วกดคีย์ ESC เครื่องก็จะมารอรับการเรียกหมายเลขอื่นต่อไป

4.1.5 การทดลองส่งจำนวนครั้งของการใช้งานขึ้นไปบน PC

ซึ่งผลการทดลองที่ได้ จะเป็นดังรูป 4.1 โดยจะมีการแสดงผลเฉพาะเครื่องที่มีสถานะเป็น ON เท่านั้น

AUTOMATIC PAGER CALL SYSTEM

NUMBER	USED	NUMBER	USED
160	10	543	10
245	10	699	15
500	15	909	12

รูปที่ 4.1 รายงานจำนวนครั้งของการใช้เครื่องที่มีสถานะเป็น ON บน PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 การทดลอง CLEAR ข้อมูล

เมื่อเครื่องทำงานอยู่ที่ MAIN MENU แล้วกด CLEAR ที่เครื่อง แล้วกดคีย์ ENTER จากนั้นตรวจสอบข้อมูลจากโหมด EDIT ซึ่งผลที่ได้คือ เครื่องทุกเครื่องมีจำนวนครั้งของการใช้เป็นศูนย์ ตรงตามวัตถุประสงค์

4.1.7 การทดลอง CLEAR สถานะและข้อมูล

ปิดเครื่อง กดคีย์ Esc ค้างไว้ แล้วเปิดเครื่อง โดยกดคีย์ Esc ค้างไว้เป็นเวลา 3 วินาที แล้วปล่อยคีย์ จากนั้นทำการตรวจสอบสถานะและจำนวนครั้งของการใช้ทุกเครื่อง ผลที่ได้คือ ทุกเครื่องมีสถานะเป็น OFF และจำนวนครั้งของการใช้เป็นศูนย์ ตรงตามวัตถุประสงค์

4.2 ปัญหาที่เกิดขึ้น

4.2.1 การประกอบวงจร ใช้แผ่นวงจรเอนกประสงค์ ทำให้การ wide สาย เกิดการผิดพลาดง่าย และดูไม่เรียบร้อยนัก

4.2.2 เสียเวลาในการทดลองวงจรมาก กว่าจะได้วงจรที่สมบูรณ์และใช้งานได้

4.2.3 เครื่องคอมพิวเตอร์ที่ใช้เขียนโปรแกรมติดไวรัส ทำให้โปรแกรมที่ใช้งานเสียหายซึ่งโปรแกรมที่เขียนไว้ยังไม่ได้ทำการ backup จึงต้องเสียเวลาเขียนโปรแกรมใหม่

4.2.4 ใช้เวลาในการเขียนโปรแกรมสร้างสัญญาณ Sinewave มาก เนื่องจากต้องหาวิธีการที่ดีที่สุด เพื่อให้สามารถกำเนิดสัญญาณได้ไว้

4.3 การแก้ไข

4.3.1 เนื่องจากวงจรที่ประกอบนี้ต้องใช้แผ่นวงจรพิมพ์เอนกประสงค์เพราะยังเป็นเครื่องต้นแบบ ซึ่งยังอยู่ในช่วงการทดลอง แต่ถ้านำไปสร้างใช้งานจริงจะต้องทำแผ่นวงจรสำเร็จออกมาเมื่อคาดว่าวงจรที่ทดลองแล้วนั้นสมบูรณ์ใช้งานได้

4.3.2 ฟอร์แมท Hard disk และติดตั้งโปรแกรมใหม่ทั้งหมด เริ่มเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหม่แล้วทำการ Backup Source โปรแกรมไว้

4.3.3 ค้นหาเอกสารและหนังสือที่เกี่ยวข้องกับเทคนิคในการเขียนโปรแกรมทั้งภาษา C และ ภาษาแอสแซมบลี้ เพื่อใช้แก้ปัญหาในการเขียนโปรแกรม

4.3.4 ทดลองวงจรแต่ละส่วนบนโฟโต้บอร์ดก่อน เมื่อทดลองใช้ได้จึงนำอุปกรณ์มาประกอบบนแผ่นวงจรเอนกประสงค์ โดยการ Wide สาย เป็นส่วนๆ แล้วนำแต่ละส่วนมาต่อใช้งานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

5.1 บทสรุป

การได้ทำปริญญานิพนธ์เรื่องเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

สามารถสรุปได้เป็นข้อ ๆ ดังนี้

1. ทางด้าน Hard Ware เนื่องจากเครื่องนี้เป็นเครื่องต้นแบบทำให้สามารถแก้ไขปรับปรุงง่าย แต่ก็มีปัญหาบ้างเกี่ยวกับการจัดหาอุปกรณ์ที่นำมาประกอบ
2. ทางด้าน Software สามารถทำงานได้สมบูรณ์ตามวัตถุประสงค์ที่วางไว้
3. การนำไปใช้งานจากผลการทดลองสามารถนำไปใช้งานได้ดีและจะนำไปใช้เพื่อใช้งานจริงในส่วนบริการโทรคมนาคมของการสื่อสารแห่งประเทศไทย (กสท.)

ทางคณะผู้จัดทำได้หวังเป็นอย่างยิ่งว่า คงมีผู้สนใจเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ เพื่อจะคิดค้นและพัฒนาขีดความสามารถของเครื่องนี้ให้มีความสามารถที่ดีขึ้นจากเดิมเพื่อเป็นประโยชน์ และเป็นชื่อเสียงแก่ภาควิชาครุศาสตร์วิศวกรรม แก่สถาบันฯ แก่ประเทศชาติรวมถึงคณะผู้จัดทำเองที่สำคัญที่สุดจะเป็นประโยชน์ทางด้านเทคโนโลยีการสื่อสาร จึงขอฝากแนวความคิดไว้แก่รุ่นน้องๆ ที่ต้องการที่พัฒนาเครื่องนี้ต่อไป

5.2 ประโยชน์ที่ได้รับจากการทำงาน

1. ได้รับความรู้ความเข้าใจในการสร้างสัญญาณ Encoder แบบ Two-tone ในการส่งออกอากาศ เพื่อใช้เรียกเครื่องวิทยุติดตามตัวระบบโทนและวอยซ์
2. ได้รับความรู้ความเข้าใจ ในหลักการของการเขียนโปรแกรมควบคุมการทำงานบน Micro Controller และด้าน Hardware ของเครื่อง
3. ได้เรียนรู้การทำงานร่วมกันเป็นกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ได้รู้จักการแก้ปัญหาที่ถูกต้อง
5. เป็นการสร้างเสริมประสบการณ์ในการทำงานอย่างมีระบบ
6. ทำให้เกิดความคิดสร้างสรรค์ในการทำงาน

5.3 แนวทางในการพัฒนาโครงงาน

1. การเข้ารหัสเพื่อเรียกเครื่องวิทยุติดตามตัว ซึ่งปกติเครื่องที่สร้างนี้สามารถเข้ารหัสเรียกเครื่องวิทยุติดตามตัวได้จำนวน 900 หมายเลข ซึ่งสามารถ แก้ไขทางด้าน Soft ware ให้สามารถเข้ารหัสเรียกได้ ถึง 1900 หมายเลข

2. การเก็บข้อความที่มีการติดต่อระหว่างผู้เรียก กับผู้รับ ข้อความที่ผู้เรียนส่งไปยัง เครื่องรับวิทยุติดตามนั้น ในขณะนี้ยังคงไม่มีการบันทึกไว้ ซึ่งในส่วน ของการบันทึกข้อความนั้น จะต้องอาศัยเครื่องคอมพิวเตอร์ ในการบันทึก และใช้การ์ดซาวน์ Bluaster ในการบันทึกเสียงพูด แล้วนำมาบันทึกไว้ในรูปของ File เพื่อการค้นหา และตอบกลับไปยังผู้ใช้บริการที่ต้องการทราบข้อความที่ติดต่อมายังหมายเลขของตนที่ผ่านมา ซึ่งต้องเพิ่มเติมในส่วนของ Software และ Hardware ทั้งบนเครื่องเรียกวิทยุติดตามตัวนี้และบนเครื่องไมโครคอมพิวเตอร์

5.4 งบประมาณค่าใช้จ่าย

1. บอร์ดไมโครคอนโทรลเลอร์ ANT -32	2,150	บาท
2. คีย์บอร์ด	500	บาท
3. LCD Module แสดงผล	1,150	บาท
4. อุปกรณ์ ฮาร์ดแวร์ประกอบวงจร + แผ่นวงจร	2,500	บาท
5. กล่องใส่เครื่อง	120	บาท
	รวม 6,420	บาท

** หมายเหตุ ค่าใช้จ่ายทั้งหมดเบิกจากการสื่อสารแห่งประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ภาคผนวก ก. เครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

ภาคผนวก ข. บล็อกไดอะแกรมและวงจรเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

ภาคผนวก ค. โฟล์วชาร์จการทำงานของเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

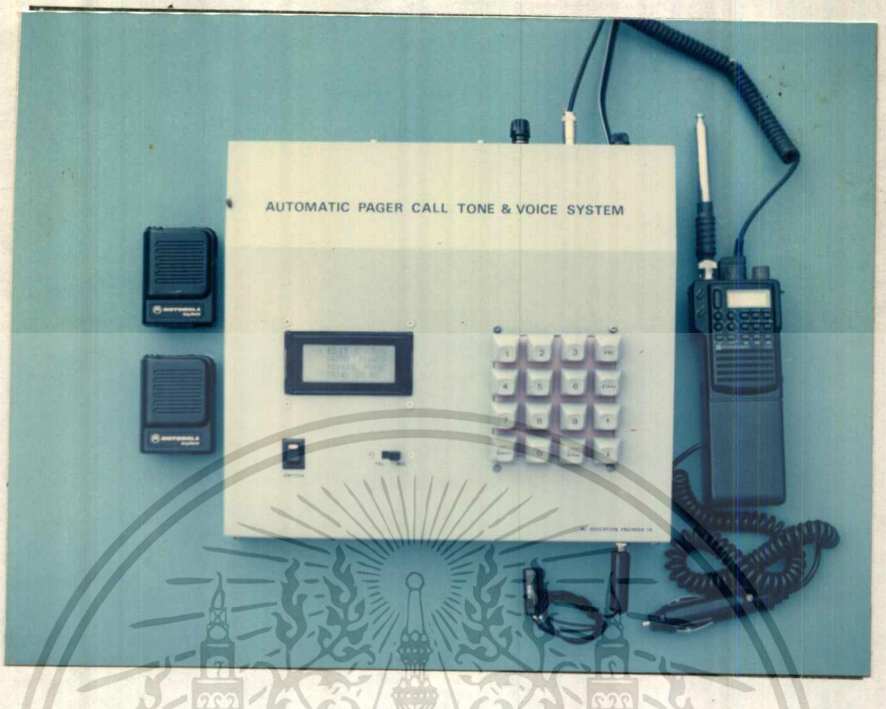
ภาคผนวก ง. โปรแกรมการทำงานของเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์แบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผนวก ก.

เครื่องเรียกวิทยุติดตามตัวระบบ โทนและวอยซ์ แบบอัตโนมัติ

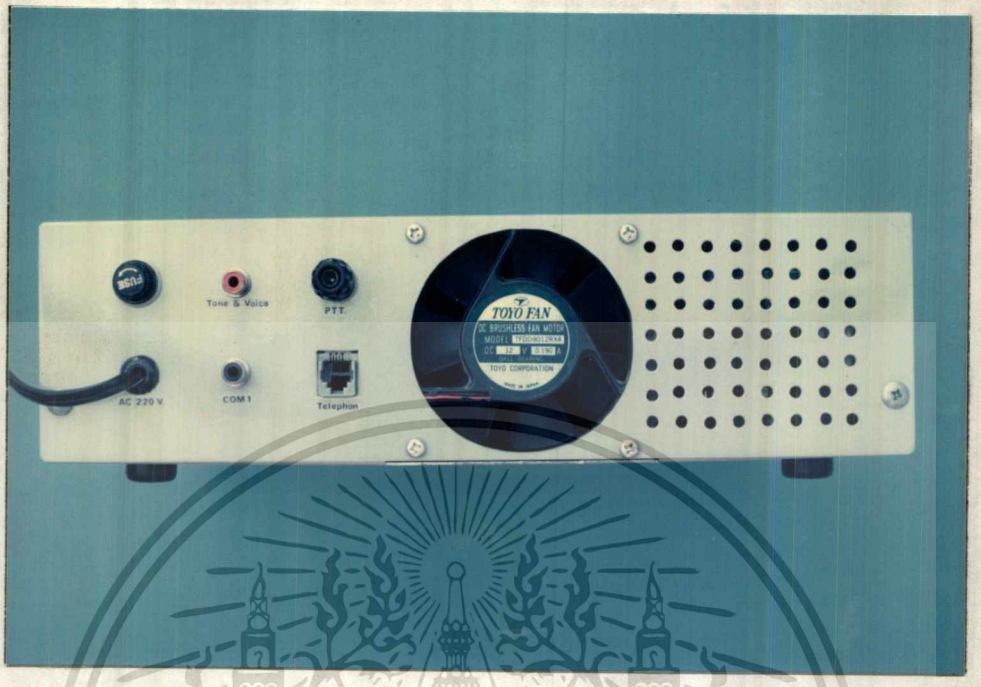
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



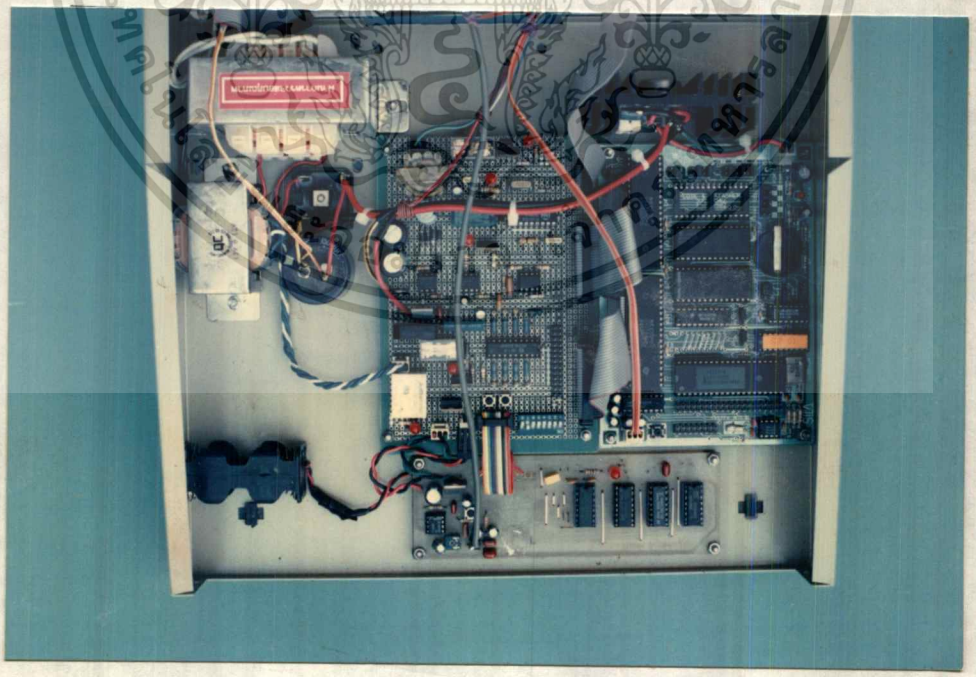
แสดงเครื่องต้นแบบที่สร้างขึ้น

แสดงจอภาพ,คีย์บอร์ด และสวิทช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงจุดต่ออินพุทและเอาท์พุท

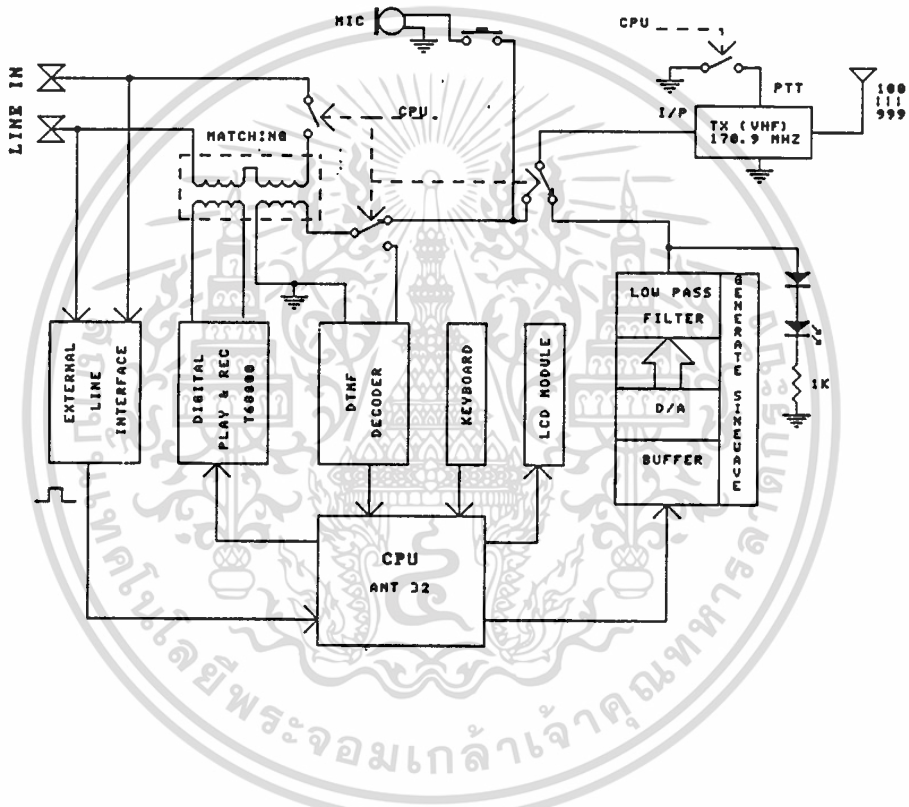


แสดงการต่อวงจรภายในเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

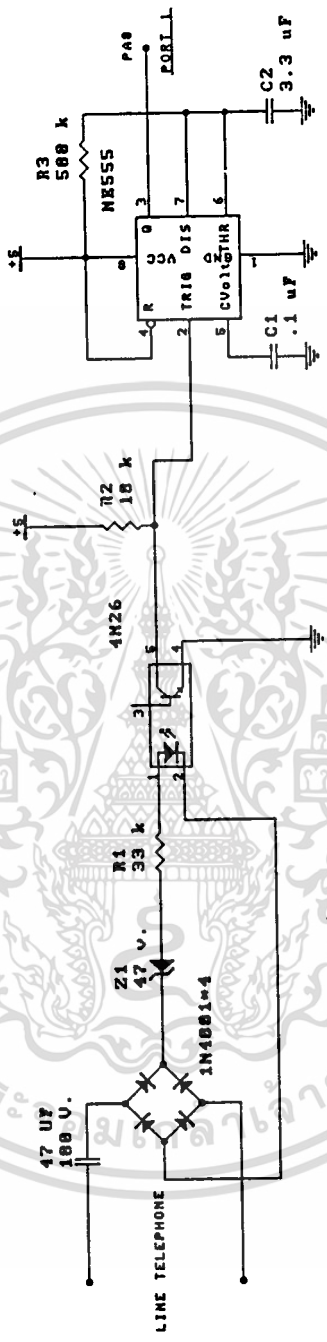
ผนวก ข.
บล็อคไดอะแกรมและวงจรเครื่องเรียกวิทยุติดตามตัวระบบโทนและวอยซ์ แบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



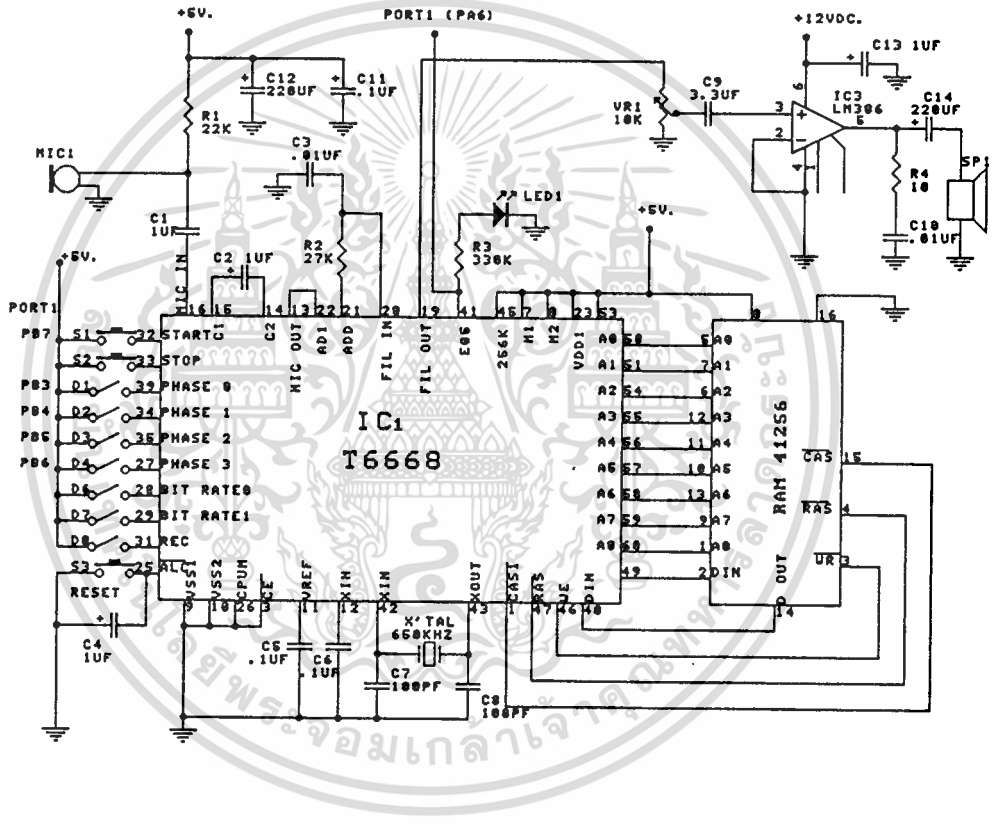
บล็อคไดแกรมของเครื่องเรียกวิทยุคิดตามตัวระบบโทนและวอยซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



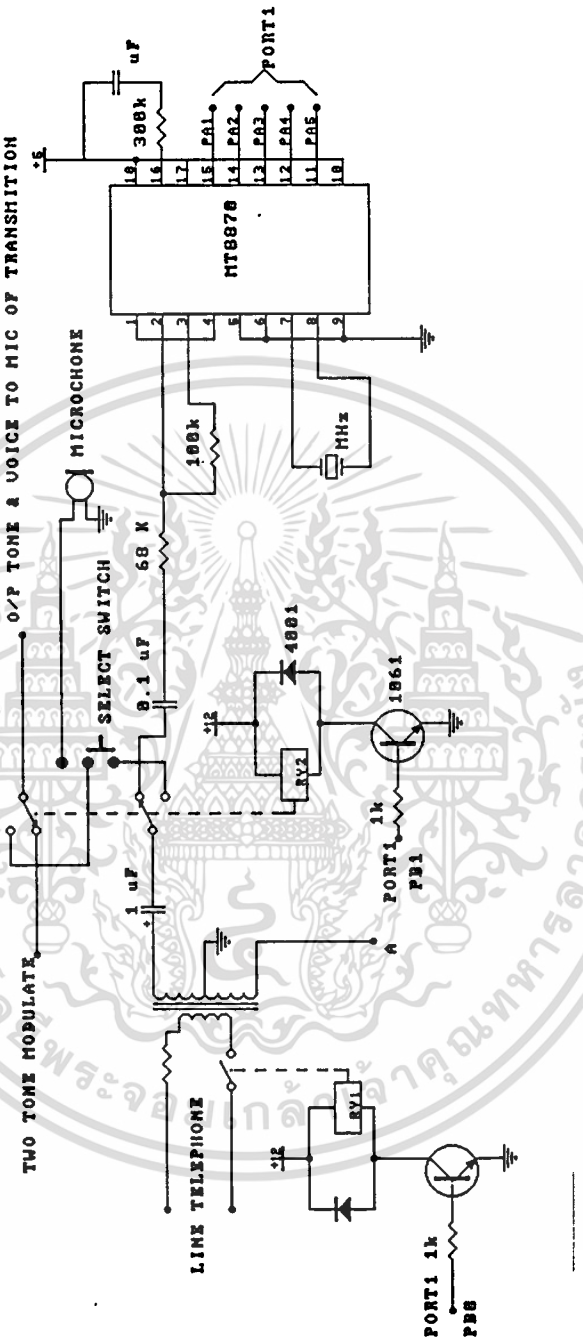
วงจร External line Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



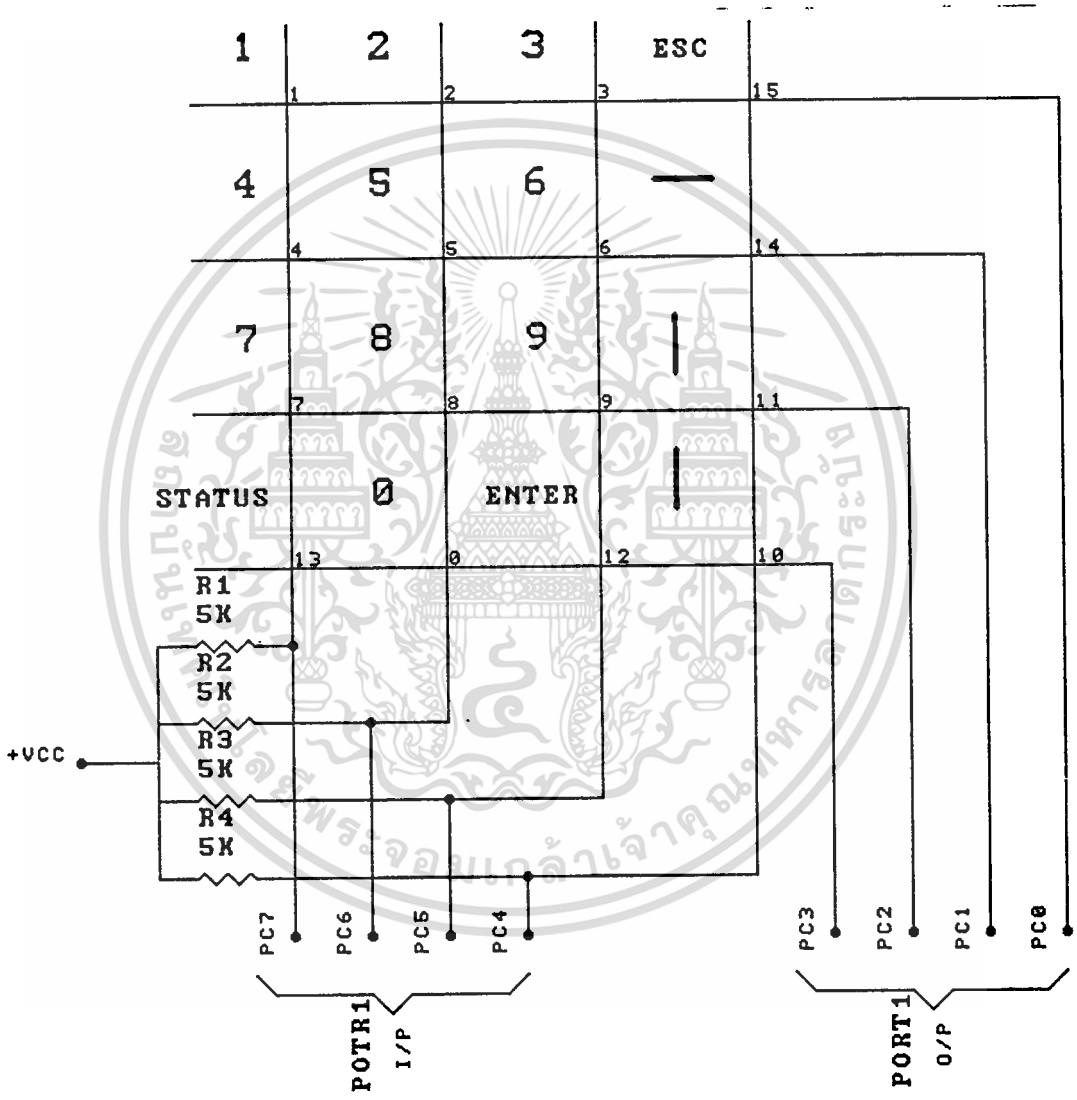
วงจร Digital Play & Record (T6668)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



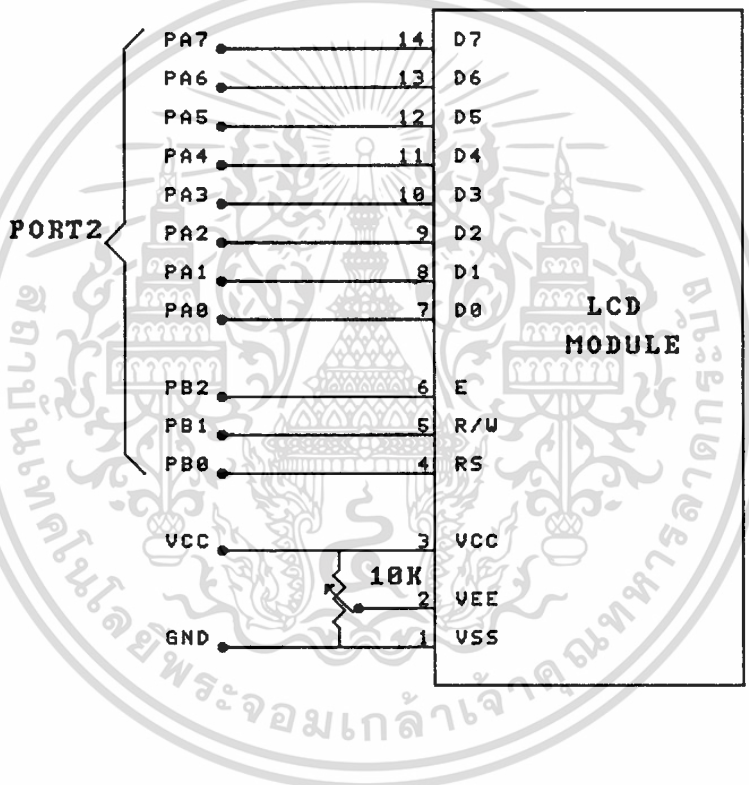
รูปที่ DTMF Decoder (Dual Tone Multi Frequency Decoder)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



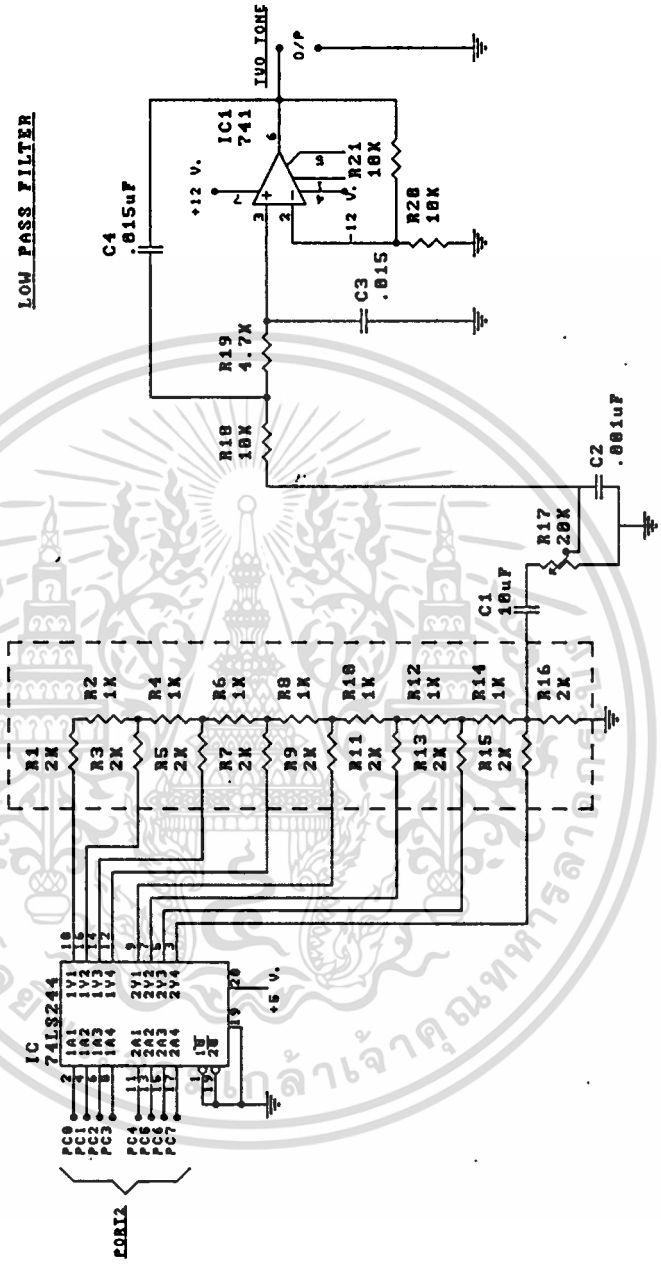
วงจร Matrix Keybord

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



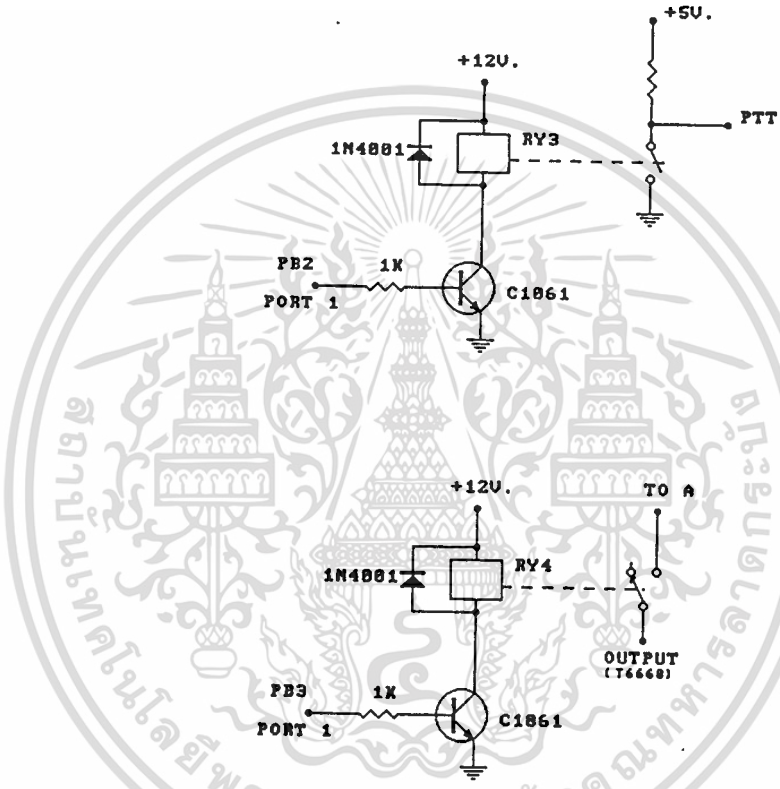
รูปที่ LCD Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



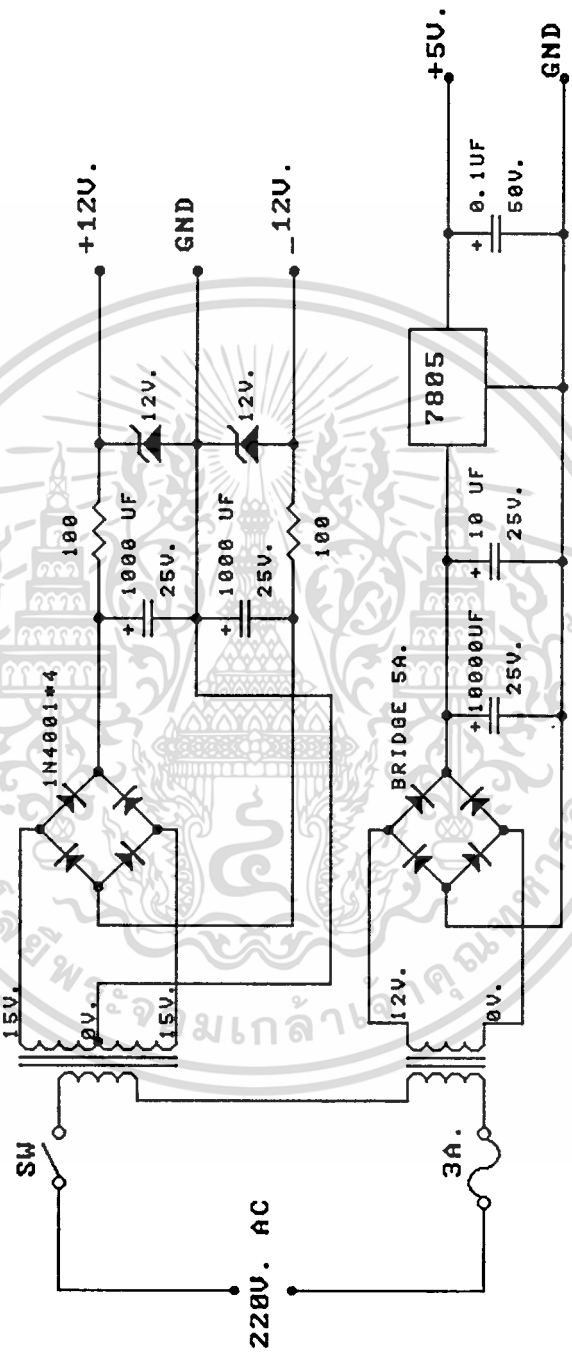
วงจร Sinewave Generate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



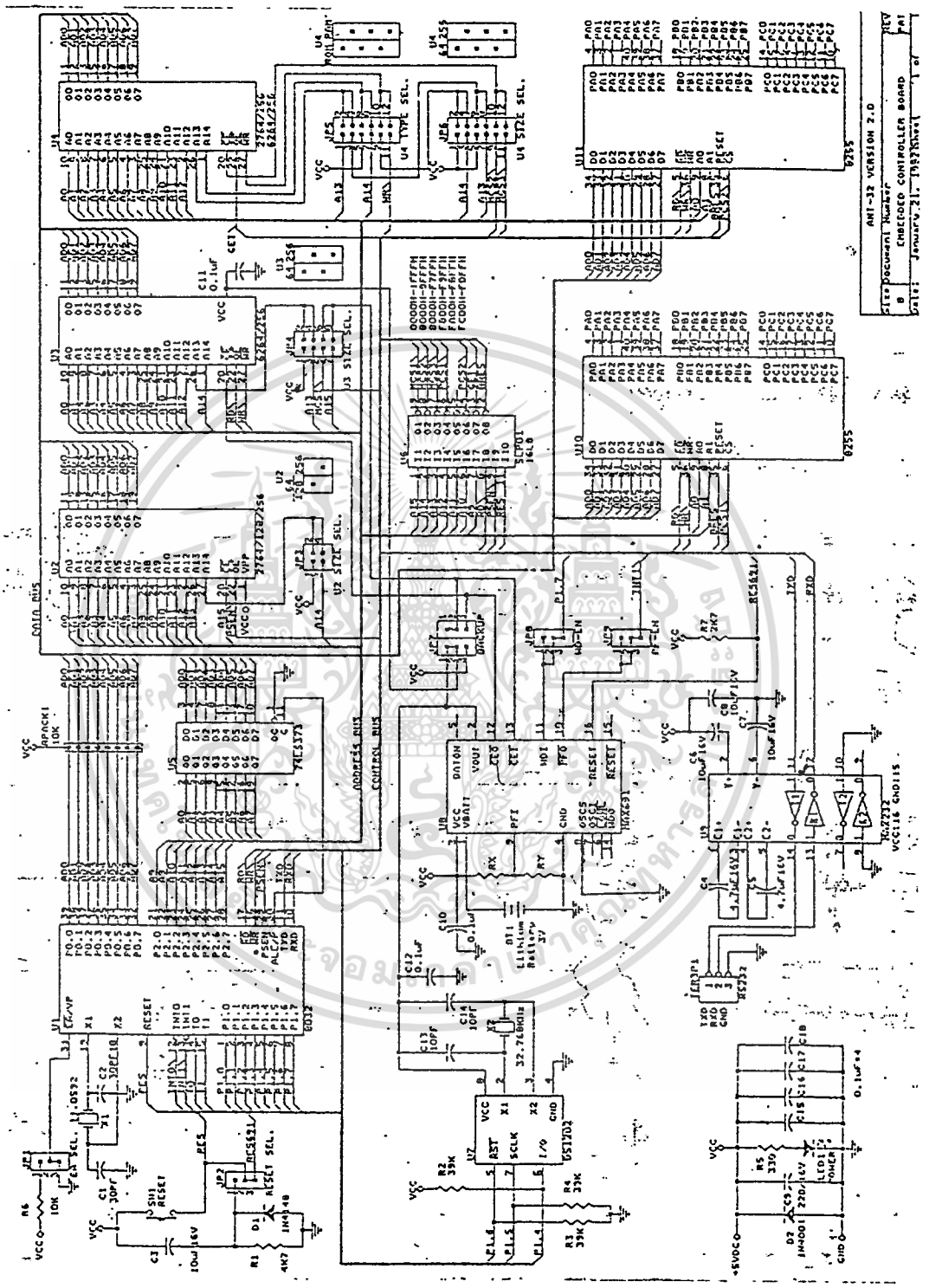
วงจร KEY PTT และวงจร RY4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจร Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

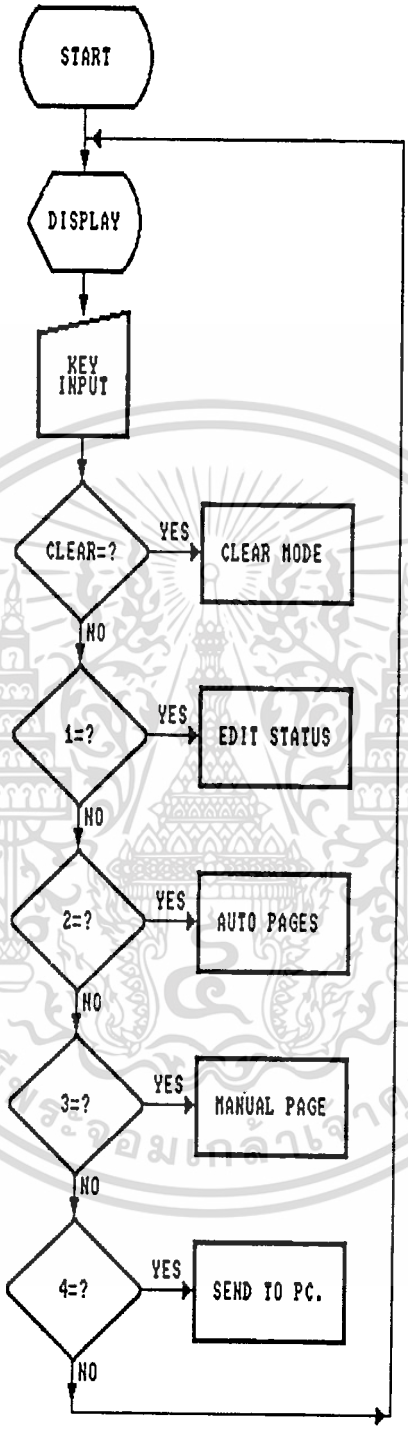


วงจรบอร์ด ANT -32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

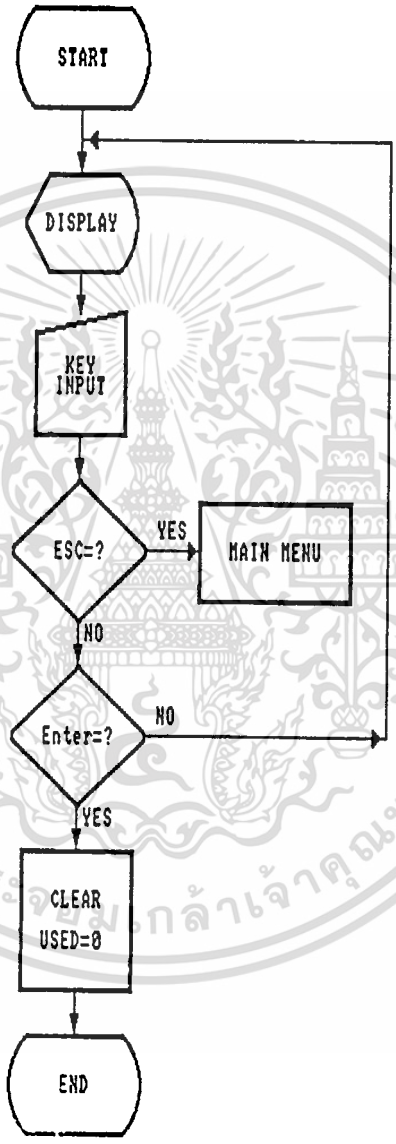
ผนวก ค.
โฟล์วชาร์จการทำงานของเครื่องเรียกวิทยุติดตามตัว ระบบ โทนและวอยซ์ แบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



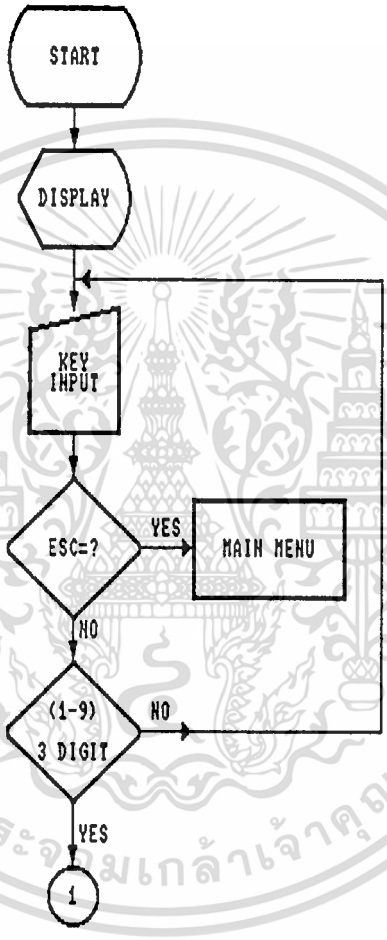
โฟล์วชาร์จแสดงโปรแกรมหลัก (MAIN MENU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



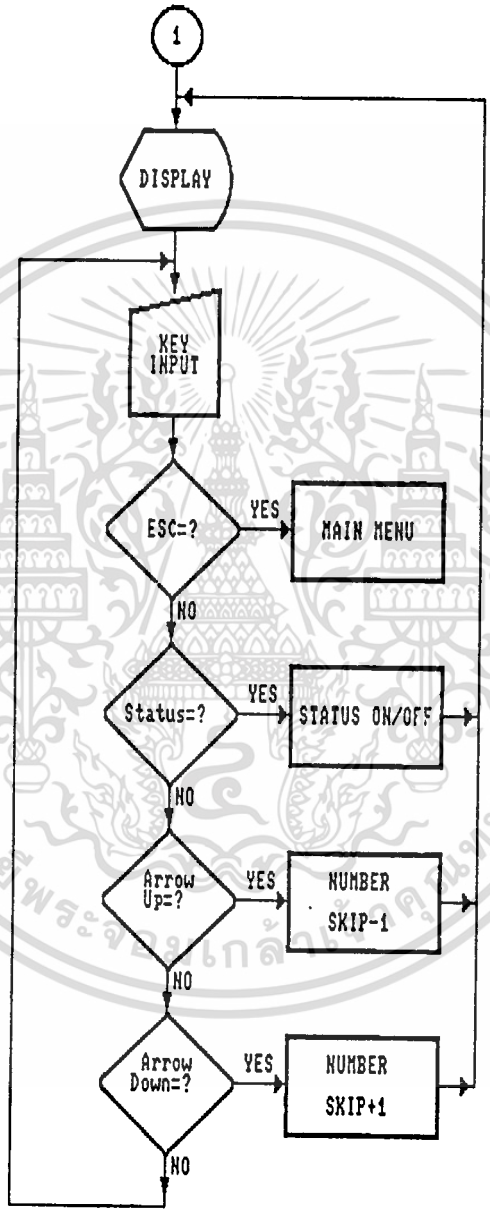
โฟล์วชาร์จแสดงโปรแกรมเคลียร์จำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



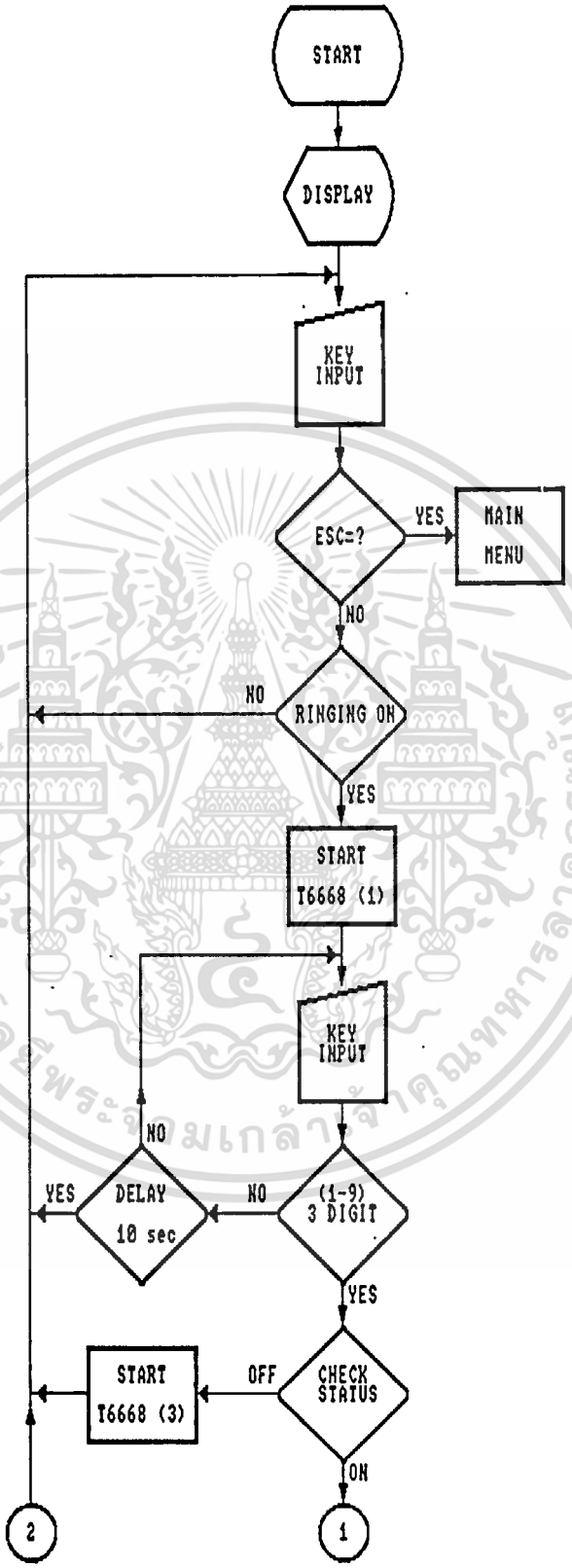
โฟล์วชาร์จแสดงโปรแกรมแก้ไขสถานะ (EDIT MODE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



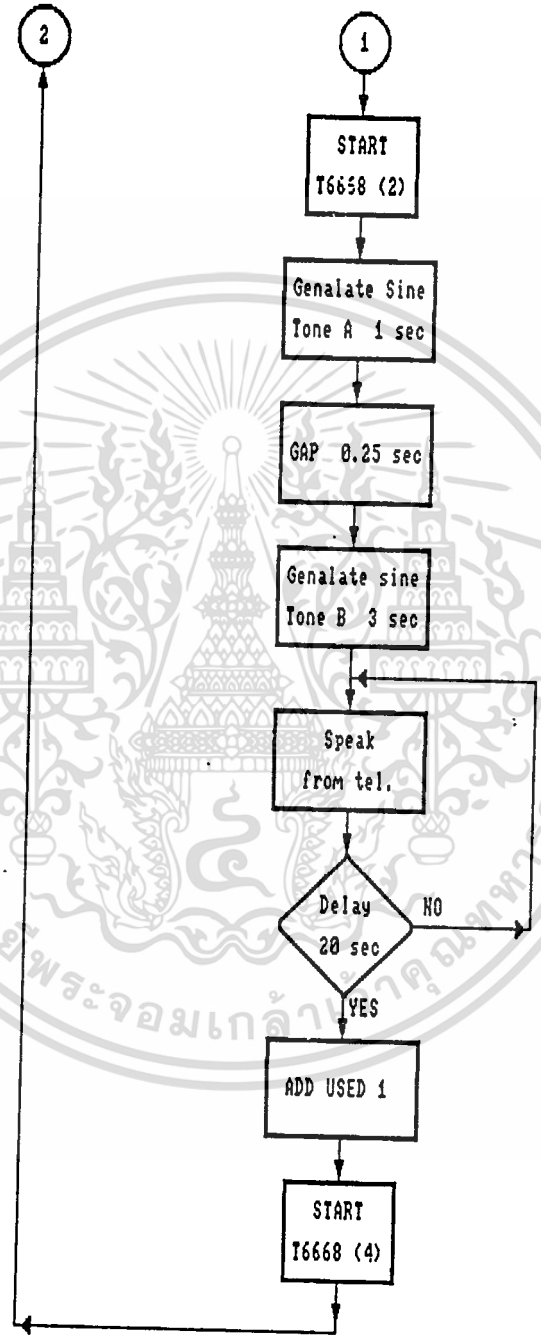
โฟล์วชาร์จแสดงโปรแกรมแก้ไขสถานะ (EDIT MODE) (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



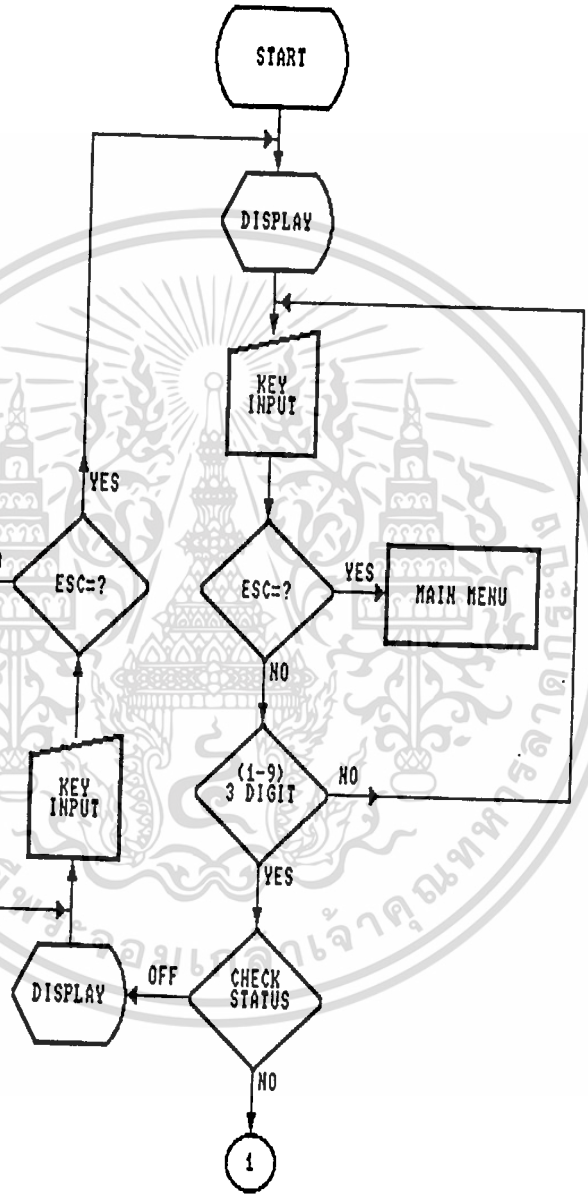
โฟล์วชาร์จแสดงโปรแกรมเครื่องเรียกวิทยุติดตามตัวแบบ AUTOMATIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โฟล์วชาร์จแสดงโปรแกรมเครื่องเรียกวิทยุติดตามตัวแบบ AUTOMATIC (AUTOMATIC PAGER) (ต่อ)

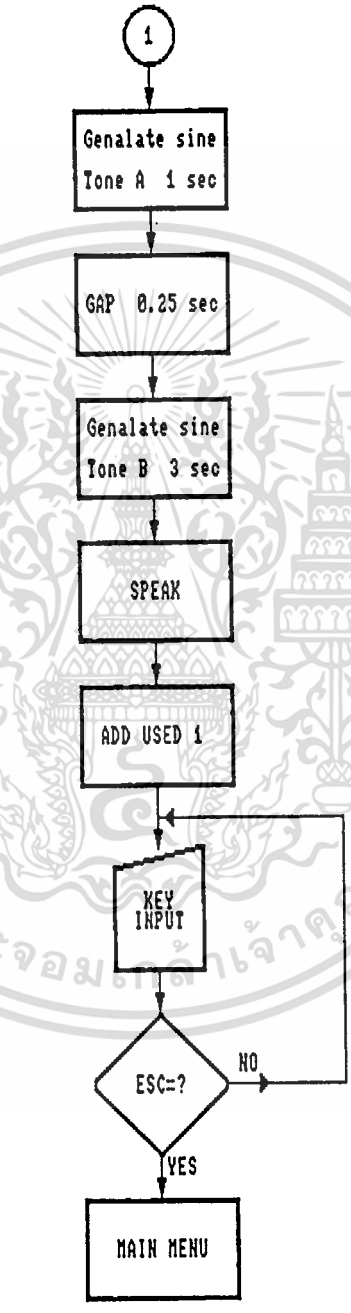
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โฟล์วชาร์จแสดงโปรแกรมเครื่องเรียกวิทยุติดตามตัวแบบ Manual

(MANUAL PAGER)

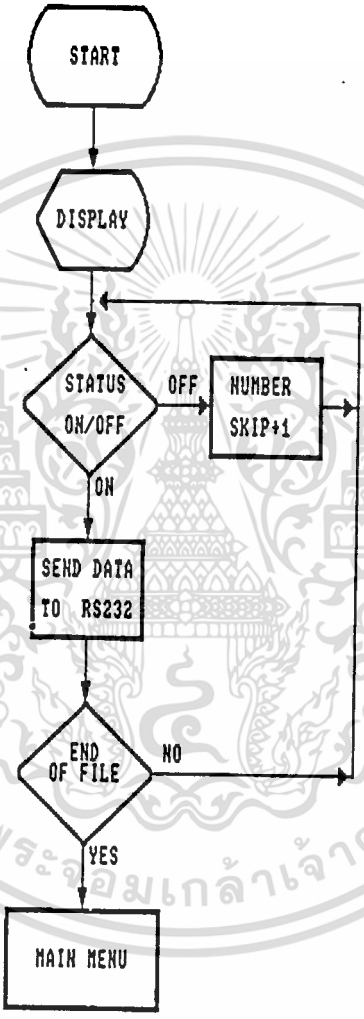
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ไฟล์วชาร์จแสดงโปรแกรมเครื่องเรียกวิทยุติดตามตัวแบบ Manual

(MANUAL PAGER) (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โฟล์วชาร์จแสดงโปรแกรมส่งข้อมูลไปคอมพิวเตอร์ (SEND TO PC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผนวก ง.
โปรแกรมการทำงานของเครื่องเรียกวิทยุติตามตัวระบบโทนและวอยซ์ แบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************/
/* Program PAG.C for                             */
/* Automatic Pager Call System ( Tone & Voice )  */
/* Compiler and Linker By AvCase51 C Compile     */
/*                                               */
/* c:\avc51\avc51 pag.c pa.asm                   */
/*                                               */
/* By EDUCATION ENGINEER 15 Date 25-10-94        */
/*************************************************/

#include   <conio.h>
#include   <stdio.h>
#include   <8051.h>
#include   <stdlib.h>
#include   <intrpt.h>
#include   <sys.h>

#define    PA1 ((char *) 0xF800)
#define    PB1 ((char *) 0xF801)
#define    PC1 ((char *) 0xF802)
#define    PD1 ((char *) 0xF803)
#define    PA2 ((char *) 0xFC00)
#define    PB2 ((char *) 0xFC01)
#define    PC2 ((char *) 0xFC02)
#define    PD2 ((char *) 0xFC03)

extern     sine_gen ();

/* Data Generate Sinewave */

unsigned   char  fa1[]={76,71,66,62,58,54,51,47,44,41};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned    char  fa2[]={ 3, 8,16,12,12,15, 5,15,11,10};
unsigned    char  fb1[]={73,69,64,60,56,52,49,46,43,39};
unsigned    char  fb2[]={13, 3,14,10,15,18,10, 4, 2,18};
unsigned    char  fc1[]={37,34,32,29,27,25,23,21,19,17};
unsigned    char  fc2[]={ 5,11, 2,12, 7, 5, 4, 5, 8,12};

char        xx[80], real[20], nn[5], yy[80], page[50][16];
int         o, z, tim;
int         *exx, *exx1;

/*  Main program  */

main()
{
    int     a=0;
    unsigned    char    b, f;

    *PD1 = 0x98;            /* Initialized System (8255) */
    *PD2 = 0x80;

    P1=1;
    P3=0xFF;

    set_vector (TIMER0, tm0 );
    TMOD=1;
    TH0=0;
    TL0=0;
    TCON=0x10;
    IE=0x82;

    InitLCD ();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OffCursor ();
Delay (2000);
Prints (0, 0, "****************");
Prints (1, 0, "*  AUTOMATIC   *");
Prints (2, 0, "*    PAGER     *");
Prints (3, 0, "****************");
Delay (25000);

exx=malloc (4000);
exx1=exx;

  *PC1=0xFE;
  a=*PC1;
  a&=0xF0;
  if (a==0xE0)
  {
     for (a=0; a<4000; a++, exx1++)
        *exx1=0;
  }

f=0;

for (;;)                              /* Main Menu */
{
   if  (f==0)
   {
      Clrsc ();
      Prints (0, 0, " -> EDIT STATUS");
      Prints (1, 0, "    AUTO  PAGES");
      Prints (2, 0, "    MANUAL PAGE");
      Prints (3, 0, "    SEND TO PC");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            a=0;
            f=1;
          }

        b=KBscan ();

      if (b==14)
      {
        Clear ();
        f=0;
        continue;    }

        if  (b==11 && a!=0)
        {   Prints (a, 1, "  ");
            a--;
            Prints (a, 1, "->");    }

        if  (b==10 && a!=3)
        {   Prints (a, 1, "  ");
            a++;
            Prints (a, 1, "->");    }

        if  (b==12)
        {
            Enter (a);
            f=0;    }
    }
}

/*  Enter function  */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Enter (a)
{
    switch (a)
    {
        case  0:
        edit ();
        break;

        case  1:
        am ();
        break;

        case  2:
        manu ();
        break;

        case  3:
        pcsend ();
        break;

    }
}

/*  Printf function  */

Printsf (x, y, msg, b)

char    msg[16];
{
    sprintf (xx, msg, b);
    Prints (x, y, xx);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/*  Print to LCD.  */

Print (line, tab, word)
{
    static  char    tbl [] = { 0x00, 0x40, 0x10, 0x50 };
    char    addr, i;

    addr=tbl[line];
    addr+= tab;
    GOTOLCD (addr);
    *PB2 = 0x01;
    *PA2 = word;
     EPULSE();
}

/*  Print to LCD.  */

Prints (line, tab, word)

char    word[16];
{
    static  char    tbl [] = { 0x00, 0x40, 0x10, 0x50 };
    char    addr, i;

    addr=tbl[line];
    addr+= tab;
    GOTOLCD (addr);
    i=0;
    while (word[i] != '\0')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        *PB2 = 0x01;
        *PA2 = word[i];
        EPULSE();
        i++;
        if (i+tab == 16)
            break;
    }
}

GOTOLCD (addr)
{
    addr |= 0x80;
    *PA2 = addr;
    *PB2 = 0;
    EPULSE ();
}

/* Initialized LCD */

InitLCD ()
{
    *PB2 = 0;
    *PA2 = 0x38;
    EPULSE ();
    Delay (50);
    *PA2 = 0x0F;
    EPULSE ();
    *PA2 = 0x06;
    EPULSE ();
    *PA2 = 0x01;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    EPULSE ();
   Delay (50);
}

/*  Clr LCD  */

Clrsc ()
{
    *PB2 = 0;
    *PA2 = 0x01;
    EPULSE ();
    Delay (50);
}

/*  On cursor  */

OnCursor ()
{
    *PB2 = 0;
    *PA2 = 0x0F;
    EPULSE ();
    Delay (50);
}

/*  Off cursor  */

OffCursor ()
{
    *PB2 = 0;
    *PA2 = 0x0C;
    EPULSE ();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Delay (50);
}

/*  Generate pulse  */

EPULSE ()
{
    int  a,b;
    int i;

    a = *PB2;
    al= 0x04;
    *PB2 = a;
    for (i = 0; i < 10 ; i++);
    a &= 0xFB;
    *PB2 = a;
}

/*  Goto XY function  */

gotoXY (x, y)
{
    static  char    tbl [] = { 0x00, 0x40, 0x10, 0x50 };
    char    addr;

    addr=tbl[x];
    addr+= y;
    GOTOLCD (addr);
}

/*  Delay function  */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Delay (z)
{
   int x,y;

   for (x = 0 ; x <= z ; x++)
      P1^=0x80;
}

/*  Initialized Serial Port  */

Inituart ()
{
   TMOD = 0x20;              /* timer 1 auto-reload mode */
   TL1 = -3;
   TH1 = -3;           /* about 9600 baud at 10.695MHz */
   TR1 = 1;                    /* enable timer 1 */
   TCON=0x40;
   SCON = 0x52;              /* mode 1, receiver enable */
}

/*  Send DATA to Serial Port  */

Send(char c)
{
    while(!TI)
      continue;
    TI = 0;
    SBUF = c;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Read keyboard */

KBscan ()
{
   static  char    dat [] = { 0xF7, 0xFB, 0xFD, 0xFE };
   static  char    dat1 [] = { 13, 0, 12, 10, 7, 8, 9, 11,
                               4, 5, 6, 14, 1, 2, 3, 15 };
   unsigned    char    a, b, c, d;

   for (;;)
   {   P1^=0x80;
       *PC1=0;
       a=*PC1;
       a&=0xF0;
       if  (a==0xF0)
           break;
   }

   for (;;)                        /* Start scan key */
   {   P1^=0x80;
       b=0;
       for (d=0; d<4; d++)
       {   *PC1=dat[d];
           a=*PC1;
           a&=0xF0;
           if  (a!=0xF0)
               break;
           b+=4;   }
       if  (a!=0xF0)
           break;    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if  (a==0xB0)                    /* Encode KEY */
        b+=1;
    if  (a==0xD0)
        b+=2;
    if  (a==0xE0)
        b+=3;

    a=dat1[b];

    return (a);
}

/*  Clear Memory  */

Clear ()
{
   unsigned    char  a, b;

   Clrsc ();

   Prints (0, 0, "** CLEAR MODE **");
   Prints (1, 0, "----------------");
   Prints (2, 0, "ESC. TO CANCEL  ");
   Prints (3, 0, "Ent. TO CONTINUE");

   while (1)
   {
      a=KBscan ();
      if (a==15)
         return;
      if (a==12)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
    }

    exx1=exx;
    exx1++;
    for   (z=0; z<4000; z+=2, exx1+=2)
        *exx1=0;

}

/*  Edit Function */

edit ()
{
    char    a, b, c;

    Clrsc ();
    Prints (0, 0, "** EDIT  MODE **");
    Prints (1, 0, " PAGES No :");
    OnCursor ();

    gotoXY (1, 12);
    a=0;
    for (;;)
    {
        b=KBscan ();

        if  (b==15)
        {
            OffCursor ();
            return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

    if (b>9)
        continue;
    if (b<1 && a==0)
        continue;
    xx[a]=b+0x30;
    Print (1, a+12, b+0x30);
    a++;
    gotoXY (1, a+12);
    if (a==3)
    {
        xx[3]=0;
        break;
    }
}

sscanf (xx, "%d", &o);
o-=100;
o*=2;
exx1=exx;
exx1+=o;
o/=2;
o+=100;
Prints (2, 0, " STATUS   :");
z=*exx1;
if (z==0)
   Prints (2, 12, "OFF");
else
   Prints (2, 12, "ON ");
exx1++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
z=*exx1;
exx1--;
Printsf (3, 0, " USED     : %d", z);
OffCursor ();

for (;;)
{
   b=KBscan ();

   if  (b==15)
       break;

   if  (b==10 && o!=999)
   {
       o++;
       exx1+=2;
       Prints (2, 0, " STATUS   :");
       z=*exx1;
       if  (z==0)
           Prints (2, 12, "OFF");
       else
           Prints (2, 12, "ON ");
       Printsf (1, 12, "%d", o);
   }

   if  (b==11 && o!=100)
   {
       o--;
       exx1-=2;
       Prints (2, 0, " STATUS   :");
       z=*exx1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (z==0)
               Prints (2, 12, "OFF");
            else
               Prints (2, 12, "ON ");
            Printsf (1, 12, "%d", o);
         }

         if (b==13)
         {
            z=*exx1;
            if (z==0)
            {
               *exx1=1;
               Prints (2, 12, "ON ");
            }
            else
            {
               *exx1=0;
               Prints (2, 12, "OFF");
            }
         }

         exx1++;
         z=*exx1;
         exx1--;
         Printsf (3, 0, " USED     : %d", z);
      }
   }

   /* Automatic Mode */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (tim==100)                    /* Delay Key 10 sec */
{
   a=200;
   break;
}
a=*PA1;
a&=2;
if (a==0)
   continue;
for   (;;)
{
   a=*PA1;
   a&=2;
   if (a==0)
      break;
}

a=*PA1;
a>>=2;
a&=0x0f;
if (a==10)
   a=0;
if (a>9)
   continue;
a+=0x30;
xx[b]=a;
b++;
Print (3, 10+b, a);
if (b==3)
   break;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (a==200)
{
  a=*PB1;
  a&=0xFE;
  *PB1=a;
  OffCursor ();
  continue;
}

xx[3]=0;                          /* Check Status */
sscanf (xx, "%d", &o);
o-=100;
o*=2;
exx1=exx;
exx1+=o;
o/=2;
o+=100;
Prints (2, 0, "  STATUS :  ");
z=*exx1;
if  (z==0)                        /* Status OFF */
{
   Prints (2, 11, "OFF");

  a=*PB1;                  /* Start T6668 Data 0011 */
  a&=0x87;
  al=0x98;
  *PB1=a;
  Delay (2000);
  a&=0x7F;
  *PB1=a;
  OffCursor ();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  a=*PB2;                          /* On Ry4 */
  al=8;
  *PB2=a;
  while (1)          /* Check Data End (0011) T6668 */
  {
    a=*PA1;
    a&=0x40;
    if (a!=0)
      break;
  }
  a=*PB2;                          /* Off Ry4 */
  a&=0xF7;
  *PB2=a;
  a=*PB1;                          /* Off Ry1 */
  a&=0xFE;
  *PB1=a;

  continue;
}
else                               /* Status ON */
{
   exx1++;
   z=*exx1;
   z++;
   *exx1=z;
   Prints (2, 11, "ON ");

   a=*PB1;                  /* Start T6668 Data 0010 */
   a&=0x87;
   al=0x90;
   *PB1=a;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Delay (2000);
  a&=0x7F;
  *PB1=a;
  OffCursor ();
  a=*PB2;                          /* On Ry4 */
  al=8;
  *PB2=a;
}

xx[0]-=0x30;          /* Check Number for Gen Sinewave */
xx[1]-=0x30;
xx[2]-=0x30;

x=xx[1];
y=xx[2];
if (xx[0]==1)
{
  a=fb1[x];
  b=fb2[x];
  c=fb1[y];
  d=fb2[y];
}
if (xx[0]==2)
{
  a=fc1[x];
  b=fc2[x];
  c=fc1[y];
  d=fc2[y];
}
if (xx[0]==3)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a=fb1[x];
    b=fb2[x];
    c=fc1[y];
    d=fc2[y];
  }
  if (xx[0]==4)
  {
    a=fa1[x];
    b=fa2[x];
    c=fa1[y];
    d=fa2[y];
  }
  if (xx[0]==5)
  {
    a=fb1[x];
    b=fb2[x];
    c=fa1[y];
    d=fa2[y];
  }
  if (xx[0]==6)
  {
    a=fc1[x];
    b=fc2[x];
    c=fb1[y];
    d=fb2[y];
  }
  if (xx[0]==7)
  {
    a=fa1[x];
    b=fa2[x];
    c=fb1[y];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   d=fb2[y];
}
if (xx[0]==8)
{
   a=fc1[x];
   b=fc2[x];
   c=fa1[y];
   d=fa2[y];
}
if (xx[0]==9)
{
   a=fa1[x];
   b=fa2[x];
   c=fc1[y];
   d=fc2[y];
}

x=*PB1;                          /* On Ry2 (PTT) */
xl=4;
*PB1=x;

tim=0;
while (tim<20)                   /* Send Tone A  1 sec */
{
     sine_gen (a, b);
}
tim=0;
while (tim<5)                    /* Send Gap  0.25 sec */
{
}
tim=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while (tim<50)                  /* Send Tone B  3 sec */
{
     sine_gen (c, d);
}

while (1)               /* Check Data End (0010) T6668 */
{
   a=*PA1;
   a&=0x40;
   if (a!=0)
      break;
}
a=*PB2;                                 /* Off Ry4 */
a&=0xF7;
*PB2=a;
a=*PB1;                                 /* On Ry2 */
al=2;
*PB1=a;

tim=0;                                  /* Delay 20 sec */
while (tim<360);

a=*PB1;                                 /* Off Ry2,Ry3 */
a&=0xF9;
*PB1=a;
a=*PB1;                         /* Start T6668 Data 0100 */
a&=0x87;
al=0xA0;
*PB1=a;
Delay (2000);
a&=0x7F;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  *PB1=a;
  a=*PB2;                              /* On Ry4 */
  a|=8;
  *PB2=a;
  while (1)              /* Check Data End (0100) T6668 */
  {
    a=*PA1;
    a&=0x40;
    if (a!=0)
      break;
  }
  a=*PB2;                              /* Off Ry4 */
  a&=0xF7;
  *PB2=a;
  a=*PB1;                          /* Off Ry1,Ry2,Ry3 */
  a&=0xF8;
  *PB1=a;

  }
}

/*  Interrupt TIMMER 0 function  */

void interrupt tm0 (void)
{
   tim++;
   TH0=0x3c;
   TL0=0xaf;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*  Manual function  */

manu ()
{
  unsigned    char    a, b, c, d, x, y;

  while (1)
  {

   a=*PB1;                              /* On Ry1 */
   al=1;
   *PB1=a;

   Clrsc ();
   Prints (0, 0, "* MANUAL  MODE *");
   Prints (1, 0, "----------------");
   Prints (2, 0, " PAGES No :");
   Prints (3, 0, "----------------");
   OnCursor ();

   gotoXY (2, 12);
   a=0;
   for (;;)
   {
      b=KBscan ();

      if  (b==15)                  /*  Cancle Key ESC */
      {
         OffCursor ();
         a=*PB1;                          /* Off Ry1 */
         a&=0xF8;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        *PB1=a;
        return;
    }

    if (b>9)
        continue;
    if (b<1 && a==0)
        continue;
    xx[a]=b+0x30;
    Print (2, a+12, b+0x30);
    a++;
    gotoXY (2, a+12);
    if (a==3)
    {
        xx[3]=0;
        break;
    }
}

sscanf (xx, "%d", &o);
o-=100;
o*=2;
exx1=exx;
exx1+=o;
o/=2;
o+=100;
Prints (3, 0, " STATUS   :     ");
z=*exx1;
if (z==0)
{
  Prints (3, 12, "OFF");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  OffCursor ();
 while (KBscan ()!=15);
  continue;
}
else
{
   exx1++;
   z=*exx1;
   z++;
   *exx1=z;
    Prints (3, 12, "ON ");
  OffCursor ();
}

a=*PB1;                    /* On Ry3 (PTT) */
al=4;
*PB1=a;

xx[0]-=0x30;          /* Check Number for Gen Sinewave */
xx[1]-=0x30;
xx[2]-=0x30;

x=xx[1];
y=xx[2];
if (xx[0]==1)
{
  a=fb1[x];
  b=fb2[x];
  c=fb1[y];
  d=fb2[y];
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (xx[0]==2)
{
   a=fc1[x];
   b=fc2[x];
   c=fc1[y];
   d=fc2[y];
}
if (xx[0]==3)
{
   a=fb1[x];
   b=fb2[x];
   c=fc1[y];
   d=fc2[y];
}
if (xx[0]==4)
{
   a=fa1[x];
   b=fa2[x];
   c=fa1[y];
   d=fa2[y];
}
if (xx[0]==5)
{
   a=fb1[x];
   b=fb2[x];
   c=fa1[y];
   d=fa2[y];
}
if (xx[0]==6)
{
   a=fc1[x];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    b=fc2[x];
    c=fb1[y];
    d=fb2[y];
  }
  if (xx[0]==7)
  {
    a=fa1[x];
    b=fa2[x];
    c=fb1[y];
    d=fb2[y];
  }
  if (xx[0]==8)
  {
    a=fc1[x];
    b=fc2[x];
    c=fa1[y];
    d=fa2[y];
  }
  if (xx[0]==9)
  {
    a=fa1[x];
    b=fa2[x];
    c=fc1[y];
    d=fc2[y];
  }

  tim=0;
  while (tim<20)                /* Send Tone A  1 sec */
  {
      sine_gen (a, b);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    tim=0;
    while (tim<5)                    /* Send Gab  0.25 sec */
    {
    }
    tim=0;
    while (tim<50)                   /* Send Tone B  3 sec */
    {
        sine_gen (c, d);
    }
   OffCursor ();
   a=*PB1;                                  /* On Ry2 */
   al=2;
   *PB1=a;

   while (KBscan ()!=15);                /* Check Key ESC */

    a=*PB1;                                /* Off Ry1,Ry2,Ry3 */
    a&=0xF8;
    *PB1=a;
    }
}

/*  Send Masege  */

s1 (char aa[100])
{
   int   a;
   char  z;

   a=0;
   for   (;;)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      if (aa[a]==0)
        break;
      else
        Send (aa[a]);
      Delay (500);
      a++;
    }

    Send (10);
    Delay (500);
    Send (13);
}

/*  Send to pc  */

pcsend ()
{
    int   a, b, c;

    Inituart ();
    Clrsc ();
    Prints (0, 0, "*****************");
    Prints (1, 0, "*  Send TO PC  *");
    Prints (2, 0, "*****************");

    s1 ("AUTOMATIC PAGER CALL SYSTEM ( Tone & Voice )");
    s1 ("  NUMBER     :     USED");
    exx1=exx;
    b=100;
    for   (a=0; a<900; a++, b++, exx1+=2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      if (*exx1==0)
         continue;
      sprintf (xx, "   %3d              %d", b, *(exx1+1));
      s1 (xx);
    }
    Send (0x1a);
    Delay (10000);
    timer ();
}

/*  Set timer  */

timer ()
{
    TMOD=1;
    IE=0x82;
    TH0=0;
    TL0=0;
    TCON=0x10;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;---------------------------------------------
; Program  PA.ASM  for
; Automatic Pager Call System ( Tone & Voice )
; Compiler and Linker   By  AvCase51 C Compile
;
; c:\avc51\avc51  pag.c  pa.asm
;
; By  EDUCATION  ENGINEER  15   Date  25-10-94
;---------------------------------------------
            defseg  c_text,class=CODE
            global  stack_internal
            global  _sine_gen

            seg     c_text

_sine_gen:    mov     a,sp
            add     a,#-2
            mov     r1,a
            mov     5,@r1
            add     a,#-2
            mov     r1,a
            mov     4,@r1
            mov     r0,#16
            mov     dptr,#sine
s1:         mov     a,#0
            movc    a,@a+dptr
            push    dph
            push    dpl
            mov     dptr,#0fc02h
            movx    @dptr,a
            pop     dpl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        pop     dph
        inc     dptr
        mov     r3,5
        mov     r2,4
s2:     djnz    r3,s2
        djnz    r0,s1
s3:     djnz    r2,s3
        ret

sine:   db      127, -81, -40, -12, -3, -12, -40, -81
        db      127, 79, 38, 10, 1, 10, 38, 79

        end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. เครื่องบันทึกเสียงพูดระบบดิจิตอล , วารสารเซมิคอนดัคเตอร์อิเล็กทรอนิกส์ , ฉบับที่ 84 ก.พ.-มี.ค. 2531,หน้า 144-151
2. พิพัฒน์ เลาหสงคราม , " ไมโครคอลโทรเลอร์ ", โครงการตำราสถาบันเทคโนโลยี-พระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะวิศวกรรมศาสตร์
3. ยืน ภู่วรวรรณ, และคณะ ,"เทคโนโลยีไมโครคอมพิวเตอร์ 16 บิท ", ซีเอ็ดยูเคชั่น,2531
4. ยืน ภู่วรวรรณ,"ทฤษฎีและการใช้งานอิเล็กทรอนิกส์ เล่ม 3", ซีเอ็ดยูเคชั่น,2525
5. ยืน ภู่วรวรรณ,"ทฤษฎีและการใช้งานอิเล็กทรอนิกส์ เล่ม 2", ซีเอ็ดยูเคชั่น,2525
6. ยืน ภู่วรวรรณ,วัฒนา เชียงกุล,"ไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์",บริษัท ซีเอ็ดยูเคชั่น จำกัด,พ.ศ.2521
7. "ไอซีน่าสน:MT8870", วารสารเซมิคอนดัคเตอร์อิเล็กทรอนิกส์,ฉบับที่ 81,2530, หน้า 226-237
8. HANDBOOK, " INSTRUCTIN MANUAL FOR TYPE PR-150 V2-5 VHF RADIO PAGING PECEIVER (TONE AND VOICE)" NIPPON ELECTRIC CO.,LTD.,TOKYO JAPAN
9. HANDBOOK,"MCS-51 MICROCONTROLLERS",ETT CO.,LTD.
10. Team NS."DATA ACQUIISITION LINERA DEVICE DATA BOOK 1989 EDITION", USA,NATIONAL SEMICONDUCTOR COPERATION,1989.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้