

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ระบบฝากข้อความทางโทรศัพท์อัตโนมัติ

๑/๗.
๖๒๑๖
๒๕๓๗

นายจักรกฤษณ์ ประกอบการ
นางสาวนิชชินทร เกตุภูพงษ์
นางสาวหวานจิตร รัตนสนธิ

เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

๖ ๒๕๑๙๔๗

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา ๒๕๓๗

**DIGITAL VOICE MAILBOX SYSTEM
(DVMS)**

**Mr. Juckrit Prakopkhan
Miss. Nitcharinton Ketpupong
Miss. Hwanjit Rattanasonti**

**A Special Project Submitted in Partial Fulfillment of the Requirement
for the Degree of Bachelor of Science
Department of Applied Physics
Faculty of Science
King Mongkut 's Institute of Technology Ladkrabang
1994**

หัวข้อโครงการพิเศษ

โดย

ภาควิชา

อาจารย์ที่ปรึกษา

ระบบฝากข้อความทางโทรศัพท์อัตโนมัติ

นายจักรกฤษณ์ ประกอบการ

นางสาวนิชชินทร เกตุภูงษ์

นางสาวหวานจิตร์ รัตนสนธิ

ฟิสิกส์ประยุกต์

อ.วิชิต ศิริโชติ

อ.ธีรวัฒน์ ประกอบผล

ภาควิชาฟิสิกส์ประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

อนุมัติให้นำโครงการพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

..... หัวหน้าภาควิชาฟิสิกส์ประยุกต์

(ผศ.ดร. ปรีชา เทียนสมประสงค์)

คณะกรรมการโครงการพิเศษ

..... ประธานกรรมการ

(ผศ.ดร. บุญส่ง ศิวโมกษธรรม)



..... กรรมการ

(ผศ.ดร. อารีย์ วิเชียรฉาย)

..... กรรมการ

..... กรรมการ

(ดร. รัชภาคย์ จิตต์อารีย์)

..... กรรมการ

..... กรรมการ

..... กรรมการ

(อ. ธีรวัฒน์ ประกอบผล)

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงการพิเศษ	ระบบฝากข้อความทางโทรศัพท์อัตโนมัติ
โดย	นายจักรกฤษณ์ ประกอบการ
	นางสาวนิชกรินทร์ เกตุภูงษ์
	นางสาวหวานจิตร รัตนสนธิ
อาจารย์ที่ปรึกษา	อ. วิชิต ศิริโชติ
	อ. อธิวัฒน์ ประกอบผล
ภาควิชา	ฟิสิกส์ประยุกต์
ปีการศึกษา	2537

บทคัดย่อ

ระบบฝากข้อความทางโทรศัพท์อัตโนมัติเป็นอุปกรณ์ที่พัฒนาขึ้นเพื่อนำไปใช้ร่วมกับระบบโทรศัพท์หรือระบบอื่นๆที่เกี่ยวข้อง โดยมีสัญญาณจากคู่สายเรียกเข้า ระบบจะทำการยกหูโทรศัพท์และเข้าสู่ระบบฝากข้อความโดยอัตโนมัติ กรณีที่ผู้โทรเข้าเรียกใช้บริการนี้ระบบก็จะทำการบันทึกข้อความเสียงโดยแปลงข้อมูลเป็นดิจิตอลและทำการเก็บข้อมูลลงบนหน่วยความจำของเครื่องคอมพิวเตอร์ส่วนบุคคลที่เชื่อมต่อกับระบบนี้ เจ้าของMailbox สามารถเรียกฟังข้อความโดยการโทรศัพท์เข้าสู่ระบบและสอบถามข้อมูลได้ การควบคุมระบบการทำงานของระบบฝากข้อความทางโทรศัพท์ จะมีโปรแกรมควบคุมเป็น ภาษา C ซึ่งในส่วนของโปรแกรมสามารถพัฒนาให้มีประสิทธิภาพยิ่งขึ้นได้ นอกจากนี้ผู้ใช้สามารถเรียกดูรายละเอียดของวันและเวลาที่ทำการบันทึกหรือฝากข้อความผ่านทางจอมอนิเตอร์ได้ และมีระบบรักษาความปลอดภัยของข้อความที่ได้บันทึกไว้ โดยมีรหัสผ่าน (password) เป็นตัวตรวจสอบความถูกต้องในการเรียกฟังข้อความ

Special Project Title	DIGITAL VOICE MAILBOX SYSTEM (DVMS)
Name	Mr. Juckrit Prakopkhan Miss. Nitcharinton Ketpupong Miss. Hwanjit Rattanasonti
Special Project Advisor	Mr. Wichit Sirichote Mr. Teerawat Prakobphol
Department	Applied Physics
Academic Year	1994

Abstract

Digital Voice Mailbox System (DVMS) is a device that can be easily installed and used as an automatic voice-message recorder in a telephone system. The way this device operates is to automatically switch an incoming call from a subscriber line to an owner mailbox recorder when the call is unattended. This can be done by detecting a ringing tone with a ringing detector installed in the DVMS . The recorded voice message is kept in a mailbox in a digital format which is then transfered to a memory of an on-line PC(Personal Computer). The owner mailbox can retrieve (or playback) the message through a system installed in the DVMS. This system which is controlled by a programme by a program written in C language can offer many interesting features including descriptions of date-time of recording and data protection from unauthorised people.

กิตติกรรมประกาศ

โครงการพิเศษนี้ เสร็จสมบูรณ์ได้ดีด้วยความช่วยเหลือจากบุคคลต่างๆดังนี้

- | | |
|--|--|
| คุณพ่อและคุณแม่ | ที่ทำให้กำลังใจทั้งยังส่งเสริมและสนับสนุนให้การศึกษา
ด้วยดีตลอดมา |
| อ. วิจิต ศิริโชติ | ที่ให้คำปรึกษาในการทำโครงการพิเศษ |
| อ. อนุชิต จารุวนาววัฒน์ | ให้คำปรึกษาและชี้แนะแนวทางในการแก้ปัญหาด้วยดี
ตลอดมา |
| อ. ชีรวัฒน์ ประกอบผล | ที่ให้คำปรึกษาและเอื้อเฟื้ออุปกรณ์ต่างๆ |
| นายเจริญศักดิ์ มันทาดิลก | ที่ให้ความช่วยเหลือทางด้านข้อมูลและอุปกรณ์ในการ
ทดลอง |
| นายธนภัทร ฉายากุล | ที่ให้คำปรึกษาต่างๆ |
| และขอขอบคุณเพื่อนๆทุกคนที่ทำให้กำลังใจมาตลอด | |

นายจักรกฤษณ์ ประกอบการ
นางสาวนิชชินทร เกตุภู่พงษ์
นางสาวหวานจิตร รัตนสนธิ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญตาราง	ช
สารบัญรูป	ซ
 บทที่ 1 บทนำ	 1
 บทที่ 2 ทฤษฎีและหลักการทำงานของระบบ	 3
2.1 วิวัฒนาการการสื่อสาร	3
2.2 ระบบการสื่อสาร	5
2.3 ระบบสื่อสารทางโทรศัพท์	7
2.3.1 ระบบโทรศัพท์และชุมสายโทรศัพท์	7
2.3.2 การทำงานของโทรศัพท์เบื้องต้น	12
2.3.3 กลไกการเชื่อมต่อวงจร	14
2.3.4 การสนทนา	14
2.3.5 ระบบการส่งสัญญาณในสายส่ง	15
2.3.6 สัญญาณเสียงพูด	15
2.3.7 สัญญาณโทรศัพท์มาตรฐาน	17
2.3.8 สัญญาณรบกวน	22
2.3.9 สัญญาณของโทรศัพท์ระบบ DTMF และระบบ PULSE	22
2.3.10 ระบบจ่ายกำลังในโทรศัพท์	25
2.3.11 แรงดันคู่สายโทรศัพท์ในสถานะต่างๆ	27
2.4 การอินเทอร์เฟส IBM/PC	30
2.4.1 สัญญาณต่างๆบนสล็อตของ IBM/PC	30
2.4.2 การจัดแอดเดรสสำหรับหน่วยความจำและ I/O	37
2.4.3 การจัดการอินเทอร์รัพท์ของ IBM/PC	42

2.5	การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลและแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	68
2.5.1	การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	68
2.5.2	การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	74
2.5.3	เทคนิคเคล็ดตามอคูเลชั่น และ CVSD	77
2.6	ระบบฝากข้อความทางโทรศัพท์อัตโนมัติ	81
บทที่ 3	การออกแบบวงจรและการทำงานของระบบ	82
3.1	ระบบเชื่อมต่อกับโทรศัพท์	84
3.1.1	วงจรตรวจจับการวางหูโทรศัพท์	84
3.1.2	วงจรตรวจจับสัญญาณกระดิ่งและตัดต่อคู่สายโทรศัพท์	85
3.1.3	วงจรตัด-ต่อคู่สายโทรศัพท์	88
3.2	วงจรถอดรหัสความถี่	88
3.3	วงจรกรองความถี่และขยายสัญญาณเสียง	92
3.3.1	ภาคอินพุท	92
3.3.2	ภาคเอาต์พุท	92
3.4	วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลและแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	97
3.5	ระบบเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคล	100
3.5.1	อินพุท/เอาต์พุท พอร์ต 8255 PIO	100
3.5.2	การติดต่อกับสล็อตของ IBM/PC	102
บทที่ 4	ผลการทดลอง	119
4.1	ผลการทดลอง	119
4.2	สรุปผลการทดลอง	120

บทที่ 5	สรุปผลการทำงานและข้อเสนอแนะในการพัฒนาระบบ	123
5.1	สรุปผลการทำงาน	123
5.2	ข้อเสนอแนะในการพัฒนาระบบ	123

ภาคผนวก ก โปรแกรมที่ใช้ในระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ

ภาคผนวก ข Data Sheet ของ IC ต่าง ๆ ที่ใช้ในวงจร

เอกสารอ้างอิง

ประวัติผู้เขียน

สารบัญตาราง

	หน้า
2-1 ตารางแสดงลักษณะของสัญญาณต่างๆที่ใช้แจ้งสภาวะ การใช้งานโทรศัพท์	29
2-2 ตารางแสดงระดับสัญญาณระหว่างคู่สายโทรศัพท์ในช่วงการใช้งาน	29
2-3 แสดงค่าแอดเดรสที่เป็นไปได้ในการขออินเทอร์รัพท์ภายใน IBM/PC	51
3-1 ตารางแสดงค่าความถี่ของ input clock , filter cutoff และ data clock	98
4-1 แสดงจำนวนไบต์ที่ใช้เก็บข้อความต่าง ๆ	122

สารบัญรูป

	หน้า
2-1 การพัฒนาการสื่อสารระหว่างมนุษยชาติ	3
2-2 แผนผังแสดงการพัฒนาระบบการสื่อสาร	4
2-3 บล็อกไดอะแกรมของการสื่อสาร	5
2-4 บล็อกไดอะแกรมของการสื่อสารที่ละเอียดขึ้น	7
2-5 โครงสร้างพื้นฐานของระบบโทรศัพท์	8
2-6 การวางสายโทรศัพท์โดยตรงระหว่างเครื่องรับโทรศัพท์ที่สลับเปลี่ยน	9
2-7 การวางสายโทรศัพท์โดยอาศัยชุมสายโทรศัพท์	9
2-8 วงจรโทรศัพท์	11
2-9 แสดงการใช้งานโทรศัพท์และการส่งสัญญาณ	13
2-10 แสดงแบนด์วิดท์ของสัญญาณเสียง	16
2-11 แสดงสัญญาณ Dial Tone	18
2-12 แสดงสัญญาณ Busy Tone	19
2-13 แสดงสัญญาณ Ring Back Tone	19
2-14 แสดงสัญญาณ Ringing Tone	20
2-15 แสดงลักษณะต่างๆ ในโทรศัพท์	21
2-16 แสดงโทรศัพท์หน้าปิดแบบหมุน	23
2-17 แสดงสัญญาณของโทรศัพท์หน้าปิดแบบหมุนเมื่อหมุนหมายเลข 4	23
2-18 แสดงหน้าปิดแบบกดปุ่มและความถี่ที่ใช้	24
2-19 แสดงแบบจำลองการติดต่อสื่อสารทางโทรศัพท์	25
2-20 แสดงวงจรโทรศัพท์อย่างง่าย	26
2-21 แสดงแรงดันบนคู่สายขณะที่มีสัญญาณต่างๆ เข้ามา	28
2-22 ขาสัญญาณต่างๆ ใน I/O SLOT บนไมโครคอมพิวเตอร์	31
2-23 แสดงถึงการใช้งานแอดเดรสบิตในการอ้างแอดเดรสทางพอร์ทใน IBM/PC	38
2-24 แสดงการใช้งานแอดเดรสต่างๆ สำหรับพอร์ท I/O ของ IBM/PC	39
2-25 การใช้งานแอดเดรสสำหรับพอร์ท I/O บนการ์ดต่างๆ	40

2-27 แสดงบล็อกไดอะแกรมของวงจรในส่วนของการอินเทอร์รัพท์ใน IBM/PC	42
2-28 บล็อกไดอะแกรมและการจัดเรียงขาของ 8259A	43
2-29 ขั้นตอนที่เกิดขึ้นในขบวนการอินเทอร์รัพท์	46
2-30 การใช้งานหน่วยความจำ 1024 ไบต์ล่างของ IBM/PC	49
2-31 การใช้เวคเตอร์ของ 8259A ใน IBM/PC	51
2-32 ตัวอย่างการเปลี่ยนแปลงลำดับความสำคัญของแชนเนลต่าง ๆ	53
2-33 แสดงบิตต่าง ๆ ภายใน ICW1	57
2-34 แสดงบิตต่าง ๆ ภายใน ICW2	59
2-35 แสดงบิตต่าง ๆ ภายใน ICW3	60
2-36 แสดงบิตต่าง ๆ ภายใน ICW4	61
2-37 แสดงบิตต่าง ๆ ภายใน OCW1	63
2-38 แสดงบิตต่าง ๆ ภายใน OCW2	64
2-39 แสดงการจัดเรียงบิตต่าง ๆ ภายใน OCW3	66
2-40 บล็อกไดอะแกรมของ A/D Converter แบบ SAR	68
2-41 วงจร A/D Converter แบบไสลบเดี่ยว	70
2-42 วงจร A/D Converter แบบไสลบคู่	71
2-43 วงจร A/D แบบขนาน	74
2-44 วงจร DAC แบบ Binary Weighted	75
2-45 วงจร DAC แบบ Ladder	76
2-46 แสดงวงจรเบื้องต้นของเดคด้ามอดุลชันในส่วนแปลงสัญญาณเสียงเป็นดิจิตอล	77
2-47 แสดงวงจรที่ใช้แปลงกลับจากข้อมูลดิจิตอลเป็นสัญญาณเสียง	78
2-48 แสดงการเปรียบเทียบสัญญาณอินพุตกับข้อมูลที่ได้สัญญาณอนาลอกจากอินทิเกรเตอร์	78
2-49 แผนผังการทำงานของระบบ CVSD ในส่วนแปลงสัญญาณเสียงเป็นสัญญาณดิจิตอล (CVSD Encoder)	79
2-50 แผนผังการทำงานของระบบ CVSD ในส่วนของการแปลงกลับจากดิจิตอลเป็นสัญญาณเสียง (CVSD Decoder)	80
3-1 โครงสร้างของระบบฝากข้อความทางโทรศัพท์อัตโนมัติ	83
3-2 วงจรตรวจจับและแปลงสัญญาณกระดิ่ง	86

3-3 วงจรนับสัญญาณกระดิ่ง	87
3-4 วงจรตัดต่อคู่สายโทรศัพท์	90
3-5 วงจรถอดรหัสความถี่ DTMF	91
3-6 วงจรกรองความถี่และขยายสัญญาณเสียงภาคอินพุต	93
3-7 วงจรกรองความถี่และขยายสัญญาณเสียงภาคเอาต์พุต	94
3-8 วงจรตรวจสอบสัญญาณกระดิ่งและตัดต่อคู่สายโทรศัพท์	95
3-9 แสดงการเชื่อมต่อวงจร Detector กับ ภาคขยายสัญญาณอินพุต	96
3-10 แสดงบล็อกไดอะแกรมระบบการทำงานของ FX709	97
3-11 แสดงบล็อกไดอะแกรมของการเลือกค่า clock input , filter clock , data clock	98
3-12 แสดงบล็อกไดอะแกรมของการเลือกอินพุต เอาต์พุต A,B	99
3-13 แสดงความหมายของบิตต่าง ๆ ที่ใช้ในการกำหนดรหัสควบคุม	101
3-14 แสดงส่วนของวงจรที่เชื่อมต่อกับ SLOT ของ IBM/PC	104

บทที่ 1

บทนำ

ในปัจจุบันโลกมนุษย์เสมือนกับว่าถูกย่อให้เล็กลงด้วยความเจริญก้าวหน้าทางเทคโนโลยี การสื่อสารโทรคมนาคมซึ่งพัฒนาไปอย่างรวดเร็วและอย่างไม่หยุดยั้งไม่ว่าจะเป็น การสื่อสารทาง ด้านเสียง ข้อมูล ข้อความและภาพเป็นต้น บริการทางโทรศัพท์นับว่าเป็นบริการทางการสื่อสารด้าน เสียงที่เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์ในยุคโลกาภิวัตน์ (Globalization) มากขึ้นเนื่อง จากบริการทางโทรศัพท์มีเครือข่ายอยู่ทั่วโลก จึงทำให้เกิดความสะดวกรวดเร็วในการส่งข่าวสารข้อมูล ทำให้มนุษย์ไม่ว่าจะอยู่ใกล้หรือไกลกันเพียงใดก็สามารถติดต่อพูดคุยกันได้เสมือนนั่งคุยอยู่ใกล้ๆ กันและสามารถรับทราบข่าวสารเหตุการณ์ต่างๆที่เกิดขึ้นในแต่ละมุมโลกได้อย่างฉับไวและทันเหตุ การณ์ ทำให้การติดต่อธุรกิจค้าขายเป็นไปได้อย่างสะดวกสบายยิ่งขึ้น ดังนั้นจำนวนผู้ใช้บริการทาง โทรศัพท์มีจำนวนมากขึ้นเพื่อเป็นการตอบสนองความต้องการที่มากขึ้นนี้จึงจำเป็นต้องมีการพัฒนา เทคโนโลยี อุปกรณ์และระบบที่เกี่ยวข้องกับการใช้งานทางโทรศัพท์ให้ดีขึ้นตามไปด้วย

ระบบบันทึกข้อความทางโทรศัพท์อัตโนมัติก็เป็นอีกรูปแบบหนึ่งของการพัฒนาระบบที่ใช้ เชื่อมต่อกับโทรศัพท์ โดยเป็นระบบที่อำนวยความสะดวกต่อผู้ใช้บริการทางโทรศัพท์ในการรับฝาก ข้อความเสียง (Voice Message) แบบอัตโนมัติ และสืบเนื่องจากแนวโน้มของการใช้งานเครื่องคอมพิวเตอร์ส่วนบุคคล (Pesonal Computer PC) ไม่ว่าจะเป็นในส่วนงานทางราชการ รัฐวิสาหกิจ เอกชน ห้างร้านต่างๆ หรือใช้ส่วนตัวภายในบ้าน มีปริมาณเพิ่มขึ้นตลอดเวลา และจากปัจจัยทาง ด้านราคาของเครื่องคอมพิวเตอร์ส่วนบุคคลในปัจจุบันซึ่งมีราคาลดลงเมื่อเทียบกับในอดีตที่มี ราคาสูงมากทำให้ความสามารถในการซื้อเครื่องคอมพิวเตอร์มีปริมาณเพิ่มมากขึ้นตามไปด้วย จึง เป็นสาเหตุประการสำคัญที่จะนำระบบฝากข้อความทางโทรศัพท์มาเชื่อมต่อกับคอมพิวเตอร์โดยทำ การควบคุมการทำงานของระบบด้วยการเขียนโปรแกรมควบคุมการทำงานของระบบ นอกจากนี้ เครื่องฝากข้อความทางโทรศัพท์อัตโนมัติยังสามารถนำมาเชื่อมต่อกับระบบ PABX (Private Automatic Branch Exchange) ได้ด้วย ซึ่งนับได้ว่าเป็นระบบที่อำนวยความสะดวกในการติดต่อสื่อสารข้อมูลในกรณีที่ไม่สามารถติดต่อระหว่างกันได้ในทันที เช่น กรณีที่ไม่สามารถพูดสายได้ในขณะ นั้น หรือ ไม่มีผู้รับสายเป็นต้น ทำให้อาจเกิดการสูญเสียข้อมูลในช่วงเวลาเร่งด่วนได้ ระบบบันทึกข้อความทางโทรศัพท์จึงเป็นทางเลือกหนึ่งที่จะช่วยให้ข่าวสารข้อมูลต่าง ๆ สามารถส่งถึงกันได้ตลอดเวลา

1.1 วัตถุประสงค์ในการทำโครงการพิเศษ

- เพื่อเป็นการพัฒนาระบบ Mail Box โดยเป็นระบบที่บันทึกข้อมูลลงบนฮาร์ดดิสต์ของเครื่องคอมพิวเตอร์ ควบคุมการทำงานและพัฒนาระบบได้โดยการเขียนโปรแกรมควบคุม
- เพื่อสร้างระบบที่อำนวยความสะดวกแก่ผู้ใช้บริการโทรศัพท์ โดยเป็นระบบที่ทำการเชื่อมต่อ กับโทรศัพท์และคอมพิวเตอร์ซึ่งสามารถทำการบันทึกเสียงและเรียกฟังข้อความได้โดยอัตโนมัติ
- เพื่อพัฒนาระบบฝากข้อความทางโทรศัพท์อัตโนมัติให้สามารถนำไปเชื่อมต่อกับระบบ PABX ได้
- เพื่อศึกษา และออกแบบ ระบบฝากข้อความทางโทรศัพท์อัตโนมัติ โดยสามารถเขียนโปรแกรมควบคุมระบบการทำงานได้

1.2 ขอบข่ายการทำงานของระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ

- เป็นศูนย์รับฝากข้อความทางโทรศัพท์โดยผู้โทรเข้าสามารถฝากข้อความ หรือ ฟังข้อความ ได้
- ทุก Mail Box มีระบบรักษาความปลอดภัยของข้อมูลด้วยรหัสผ่าน (Password) และสามารถเปลี่ยนรหัสผ่านได้เมื่อต้องการ
- ข้อความที่ฟังแล้วสามารถเก็บไว้ฟังใหม่และลบออกได้เมื่อต้องการ
- ควบคุมระบบการทำงานได้โดยการเขียนโปรแกรมควบคุม
- ระบบสามารถนำไปประยุกต์ใช้งานเชื่อมต่อการสื่อสารทางด้านอื่น ๆ ได้

1.3 ประโยชน์ที่จะได้รับ

- ได้รับความรู้ และข้อมูลเพิ่มเติมเกี่ยวกับขั้นตอนการทำงาน และสัญญาณต่างๆ ในระบบโทรศัพท์
- สามารถเข้าใจระบบการทำงาน และการเชื่อมต่อพอร์ตภายนอกกับเครื่องคอมพิวเตอร์ โดยสามารถนำมาประยุกต์ใช้ได้อย่างมีประสิทธิภาพ
- สามารถสร้างระบบที่สามารถรับฝากข้อความทางโทรศัพท์อัตโนมัติ โดยสามารถนำไปใช้งานได้จริง และสามารถนำไปเชื่อมต่อกับเครื่องมือสื่อสารอื่นๆ ได้ เช่น ระบบ PABX เป็นต้น

บทที่ 2

ทฤษฎีและหลักการทำงานของระบบ

การสื่อสารนั้นจัดได้ว่าเป็นปัจจัยที่สำคัญอีกปัจจัยหนึ่งของมนุษย์นอกเหนือจากปัจจัยสี่ แม้ว่าการสื่อสารจะไม่เกี่ยวข้องกับความเป็นความตายของมนุษย์เหมือนกับปัจจัยสี่ แต่การที่จะได้มาซึ่งปัจจัยสี่นั้นย่อมต้องอาศัยการสื่อสารเป็นสื่อกลางอย่างแน่นอน มนุษย์อาศัยการสื่อสารเป็นเครื่องมือเพื่อบรรลุวัตถุประสงค์ในการดำเนินกิจกรรมนานาชนิดของตนและอยู่ร่วมกับคนอื่นในสังคม การสื่อสารเป็นพื้นฐานของการติดต่อของมนุษย์ และเป็นเครื่องมือของกระบวนการสังคมยิ่งสังคมที่มีความซับซ้อนมากเพียงใดและประกอบด้วยคนจำนวนมากเพียงใดการสื่อสารก็ยิ่งมีความสำคัญมากขึ้นเท่านั้น

การสื่อสารเป็นส่วนหนึ่งของชีวิตและสังคมจนมีคำกล่าวว่า การสื่อสาร คือ ชีวิต การสื่อสารคือสังคม และในทางกลับกันชีวิตและสังคมก็คือ การสื่อสารกับตนเองกับผู้อื่นกับสิ่งแวดล้อมและกับโลกนั่นเอง

2.1 วิวัฒนาการของการสื่อสาร

อักษรภาพอียิปต์โบราณ อักษรคูเนอ

การเขียนอักษรลงบนกระดาศาปาปรัส

อักษรกรีกโบราณ

ตัวเลขของชาวฮินดู

เครื่องพิมพ์ของ Gutenberg(1440)

โทรเลข(1844)

ทรานสมิตเตอร์และรีเซปเตอร์(1858)

โทรศัพท์(1876) เครื่องบันทึกเสียง(1877)

วิทยุโทรเลข(1895)

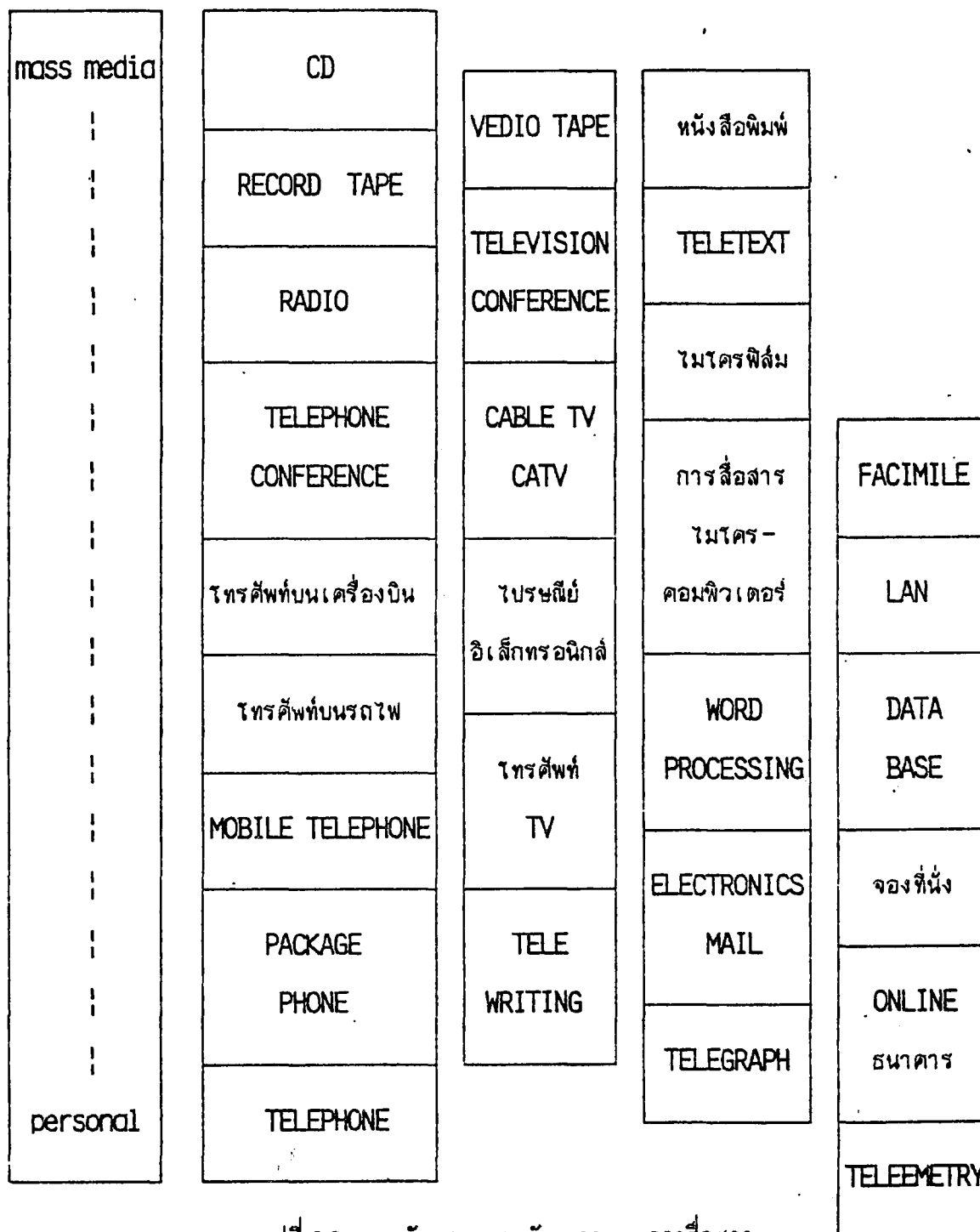
โทรทัศน์(1923)

สื่อสารดาวเทียม(1957)

รูปที่ 2-1 รวบรวมให้เห็นถึงการพัฒนาการสื่อสารระหว่างมนุษยชาติ

แผนผังแสดงพัฒนาการบริการการสื่อสาร

พิจารณาจากบนไปล่าง เริ่มการใช้งานในลักษณะ mass จนไปถึงการใช้งานส่วนบุคคล
ถ้าพิจารณาจากซ้ายไปขวาจะเรียงตามลักษณะของการสื่อสารที่ใช้ในการส่งสาร



รูปที่ 2-2 แผนผังแสดงการพัฒนาระบบการสื่อสาร

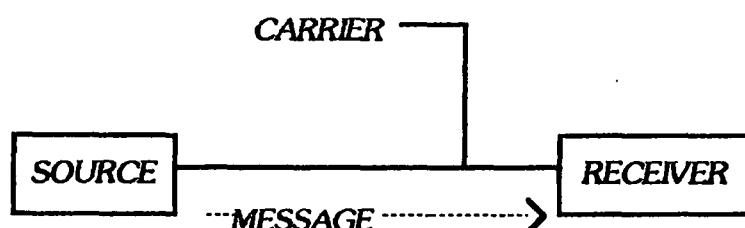
2.2 ระบบการสื่อสาร

คำว่า " การสื่อสาร " (Communion) มีคำจำกัดความหมายถึง การส่งถ่ายข้อมูลที่มีความหมายในตัวจากแหล่งข้อมูลหนึ่ง (ผู้ส่ง , เครื่องกำเนิดหรือเครื่องส่งข้อมูล) ไปสู่อีกแหล่งข้อมูลหนึ่ง (ผู้รับ , เครื่องรับสัญญาณปลายทางหรือเครื่องรับข้อมูล)

เครื่องส่ง หมายถึง แหล่งกำเนิดข้อมูลหรือเครื่องส่งสัญญาณต่าง ๆ สำหรับการสื่อสาร และคำว่า เครื่องรับ หมายถึง แหล่งรับข้อมูลหรือเครื่องรับสัญญาณที่จุดหมายปลายทาง

ในการสื่อสารนั้น สิ่งที่จะละเว้นการกล่าวถึงสิ่งนี้ได้เลยก็คือ ตัวกลาง ที่ใช้เพื่อเป็นทางข้อมูลจากเครื่องส่งไปยังเครื่องรับตัวกลางที่ให้นำข้อมูลนี้มีชื่อเรียกเฉพาะว่า " ช่องสัญญาณ "

' (Channel) " หรือ " ทรานสมิชัน " (Transmission) สามารถเขียนแผนผังการสื่อสารได้ดังรูปที่ 2-3

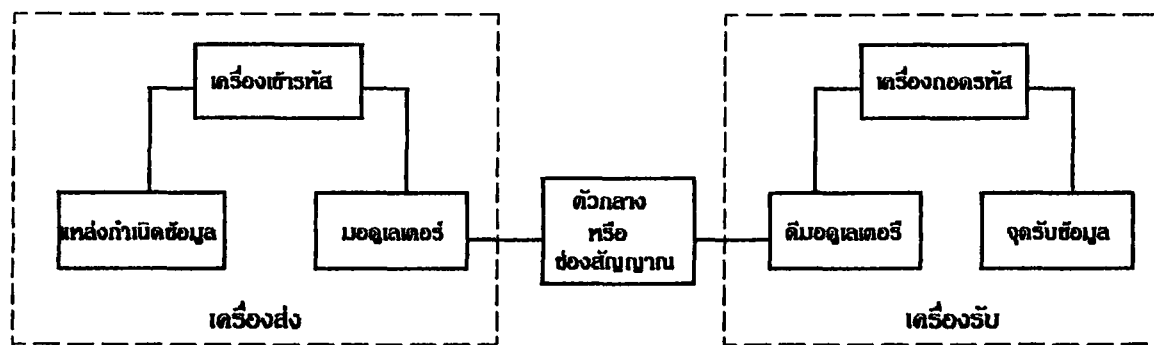


รูปที่ 2-3 บล็อกไดอะแกรมของการสื่อสาร

การสื่อสารอาจแบ่งเป็นประเภทใหญ่ ๆ ได้เป็น 2 ประเภทคือ การสื่อสารตามสาย และสายส่งสัญญาณ (Transmission) หรือเป็นบรรยากาศ (อาศัยใช้คลื่นแม่เหล็กไฟฟ้าส่งผ่าน บรรยากาศ การสื่อสารวิทยุทั้งนี้ขึ้นอยู่กับช่องสัญญาณหรือตัวกลางที่ใช้สำหรับส่งสัญญาณอยู่นั้นเป็นอากาศ)

ดังได้กล่าวมาแล้วว่า เพื่อทำการเพิ่มประสิทธิภาพของการสื่อสาร และอีกทั้งให้มีการรักษาความลับในการส่งข้อมูลขึ้นเมื่อจำเป็นเราสามารถทำได้โดยวิธีการจัดการเข้ารหัสให้กับข้อมูลนั้นโดยทางด้านเครื่องส่งจะใช้เครื่องเข้ารหัส (Encoder) เป็นตัวทำหน้าที่นี้ และทางด้านเครื่องรับก็จะใช้เครื่องถอดรหัส (Decoder) สำหรับทำการถอดรหัสนั้น ซึ่งตามธรรมดาแล้วเครื่องมือการสื่อสารส่วนใหญ่มักจะมีเครื่องส่งเครื่องรับอยู่ภายในเครื่องเดียวกัน ดังนั้นเมื่อจำเป็นจะต้องใช้เครื่องเข้ารหัสและเครื่องถอดรหัสในเครื่องมือการสื่อสาร มักจะถูกสร้างขึ้นคู่กันบรรจุอยู่ในเครื่องมือการสื่อสาร จึงทำให้เกิดมีคำศัพท์เฉพาะสำหรับเรียกชื่อรวมของเครื่องเข้ารหัส-ถอดรหัสนี้ว่า " โคเดค " (Codec) ซึ่งได้มาจากรากศัพท์เดิม Coder & Decoder นั่นเอง

ตามปรกตินั้น สัญญาณที่ได้จากตัวแหล่งกำเนิดข้อมูลโดยตรงหรือข้อมูลที่ได้ออกมาจากเครื่องเข้ารหัสนั้น มักจะมีคุณสมบัติไม่เหมาะสมที่จะใช้ส่งผ่านช่องสัญญาณไปยังเครื่องรับโดยทันที ในลักษณะรูปแบบของสัญญาณที่ได้ออกมานั้น ยกตัวอย่างเช่น เมื่อสัญญาณข้อมูล คือสัญญาณจากไมโครโฟนซึ่งกำเนิดมาจากเสียงต่าง ๆ นั้น ถ้าเราต้องการที่จะส่งสัญญาณนี้ผ่านสายอากาศแล้วผ่านช่องสัญญาณซึ่งเป็นบรรยากาศออกไปยังเครื่องรับแล้ว เราจะพบว่าคลื่นสัญญาณไฟฟ้าที่มีความถี่อยู่ในย่านความถี่เสียงนั้นถ้าจะแพร่กระจายออกไปจากสายอากาศได้แล้วก็จะต้องใช้สายอากาศที่มีความยาวมากมายหลายกิโลเมตรซึ่งเป็นการยากที่จะนำมาใช้ในการปฏิบัติงานจริงดังนั้น เพื่อให้เราสามารถส่งข้อมูลของเสียงผ่านบรรยากาศไปยังเครื่องรับได้สะดวกเราจะต้องอาศัยวิธีการบางอย่างมาช่วย เช่น การอาศัยคลื่นสัญญาณไฟฟ้าที่มีความถี่สูงพอสมควรมาเป็น คลื่นพาห้ (carrier) ชวนำข้อมูลเสียงนั้นผ่านบรรยากาศไปยังเครื่องรับดังนี้ เป็นต้น ทั้งนี้เพราะเราทราบกันดีอยู่แล้วว่า สายอากาศซึ่งใช้ช่วยกระจายคลื่นแม่เหล็กไฟฟ้าจะมีขนาดที่สั้นลงเมื่อความถี่ของคลื่นที่ใช้งานนั้นมีค่าสูงขึ้น กระบวนการที่ทำการฝากข้อมูลไปกับคลื่นพาห้นี้เป็นกระบวนการที่รู้จักกันดีในนามว่า "การมอดูเลชัน" (modulation) โดยเครื่องมือที่ใช้ทำหน้าที่นี้มีชื่อเรียกว่า "มอดูเลเตอร์" (modulator) และย่อมเป็นการแน่นอนว่าเมื่อทางด้านเครื่องส่งมีกระบวนการฝากข้อมูลไปกับคลื่นพาห้แล้วทางด้านเครื่องรับก็จะต้องมีกระบวนการที่จะแยกเอาข้อมูลออกมาจากคลื่นพาห้นั้นกระบวนการแยกข้อมูลออกจากคลื่นพาห้เรียกว่า "การดีมอดูเลชัน" (demodulation) ซึ่งในบางครั้งอาจจะเรียกว่า "การดีเทคชัน" (detection) เครื่องมือที่ใช้สำหรับทำหน้าที่นี้มีชื่อว่า "ดีมอดูเลเตอร์" (demodulator) หรือในบางครั้งก็เรียกว่า "ดีเทคเตอร์" (detector) มอดูเลเตอร์และดีมอดูเลเตอร์นั้นเมื่ออยู่ร่วมกันแล้วมีชื่อเรียกเป็นศัพท์โดยเฉพาะว่า "โมเด็ม" (modem)

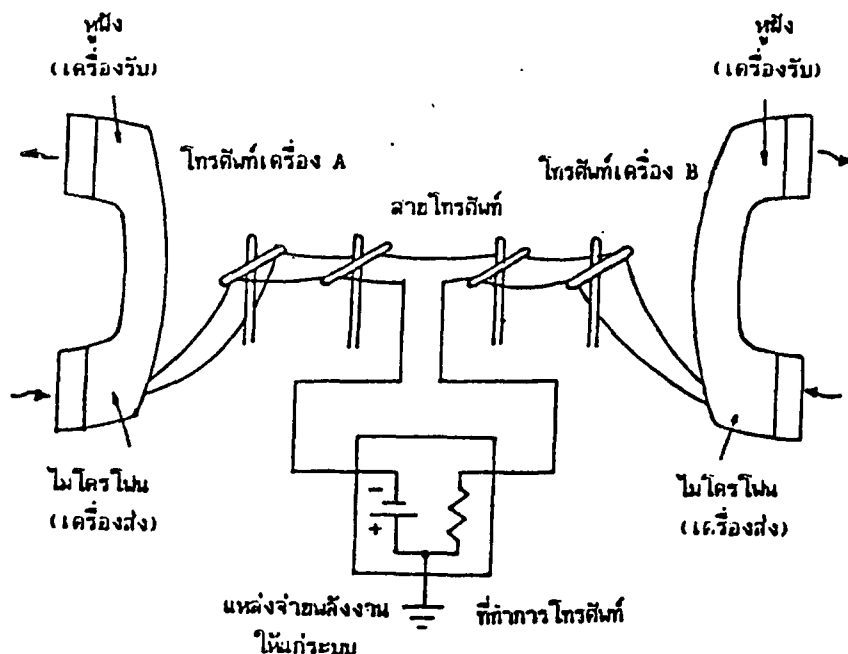


รูปที่ 2-4 บล็อกไดอะแกรมของการสื่อสารที่ละเอียดขึ้น

2.3 ระบบสื่อสารทางโทรศัพท์

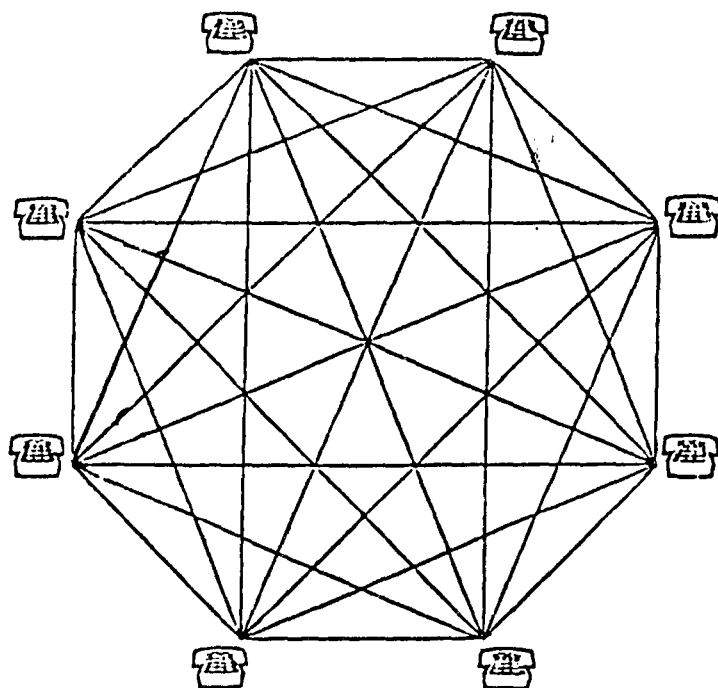
2.3.1 ระบบโทรศัพท์และชุมสายโทรศัพท์

ระบบโทรศัพท์เป็นระบบซึ่งพัฒนามาจากโทรเลขโดยใช้ไมโครโฟนแทนคีย์โทรเลขและใช้ลำโพงเล็กๆเป็นหูฟังแทนมูเซอร์ซึ่งวางจางๆ ของโทรศัพท์มีแสดงในรูปที่ 2-5 ไมโครโฟนจะทำหน้าที่เปลี่ยนสัญญาณเสียงให้เป็นสัญญาณไฟฟ้าส่งไปยังเครื่องรับซึ่งใช้ลำโพงเป็นตัวเปลี่ยนสัญญาณไฟฟ้ากลับเป็นสัญญาณเสียงตามเดิม ไมโครโฟนและลำโพงจะรวมอยู่ด้วยกันในส่วนที่เรียกว่า "หูโทรศัพท์" (subset) ภายในที่วางหูโทรศัพท์นั้นจะมีอุปกรณ์อย่างหนึ่งที่เรียกว่า "ไฮบริดทรานสฟอร์มเมอร์" (hybrid transformer) อยู่ ไฮบริดทรานสฟอร์มเมอร์นี้จะทำหน้าที่เป็นตัวแยกสัญญาณจากไมโครโฟนส่งออกไปยังสายโทรศัพท์และรับสัญญาณที่ส่งมาตามสายโทรศัพท์ส่งไปยังลำโพงหูโทรศัพท์และอุปกรณ์ต่างๆภายในที่วางหูโทรศัพท์รวมเรียกว่า "เครื่องรับโทรศัพท์" (hand set)

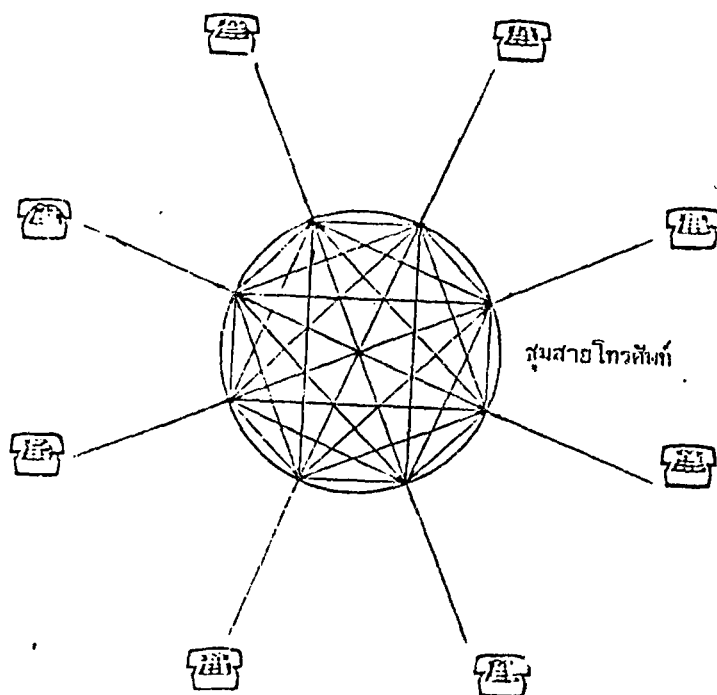


รูปที่ 2-5 โครงสร้างพื้นฐานของระบบโทรศัพท์

เมื่อจำนวนผู้ใช้โทรศัพท์มีมากขึ้นการที่จะวางคู่สายโทรศัพท์โดยตรงระหว่างเครื่องรับโทรศัพท์แต่ละเครื่องดังแสดงในรูปที่ 2-6 ย่อมเป็นการสิ้นเปลืองค่าใช้จ่ายมาก เพราะถ้ามีเครื่องรับโทรศัพท์อยู่ N เครื่องก็จะต้องใช้สายในการติดต่อกัน ทั้งหมดเป็นจำนวนถึง $N(N-1)/2$ คู่ ยกตัวอย่างเช่น ถ้ามีโทรศัพท์อยู่เพียง 100 เครื่องเท่านั้น เราก็จำเป็นจะต้องใช้สายทั้งหมดเป็นจำนวนถึง 4850 คู่ ดังนั้นเป็นต้น อย่างไรก็ตามเมื่อพิจารณาจากความจริงแล้วโทรศัพท์แต่ละเครื่องนั้นมิได้มีการใช้อยู่ตลอดเวลา คงใช้กันในระยะเวลานั้นๆ เท่านั้น ดังนั้นถ้าเราสามารถคาดคะเนได้ว่าการใช้โทรศัพท์ในเวลาเดียวกันนั้นมีจำนวนการใช้โดยเฉลี่ยเกิดขึ้นพร้อมๆ กันเท่าใดเราก็สามารถที่จะสร้างวงจรสลับสายขึ้น เพื่อให้โทรศัพท์ทั้งหลายสามารถใช้ร่วมกันอย่างมีประสิทธิภาพได้ตามรูปที่ 2-7 นั้นจะเห็นได้ว่า โดยอาศัยชุมสายหรือวงจรสลับสายนั้นสามารถลดจำนวนสายที่ใช้ในการติดต่อระหว่างเครื่องรับโทรศัพท์ให้เหลือเพียงจำนวนของเครื่องรับโทรศัพท์ทั้งหมดเท่านั้น



รูปที่ 2-6 การวางสายโทรศัพท์โดยตรงระหว่างเครื่องรับโทรศัพท์ที่เปลี่ยนแปลง



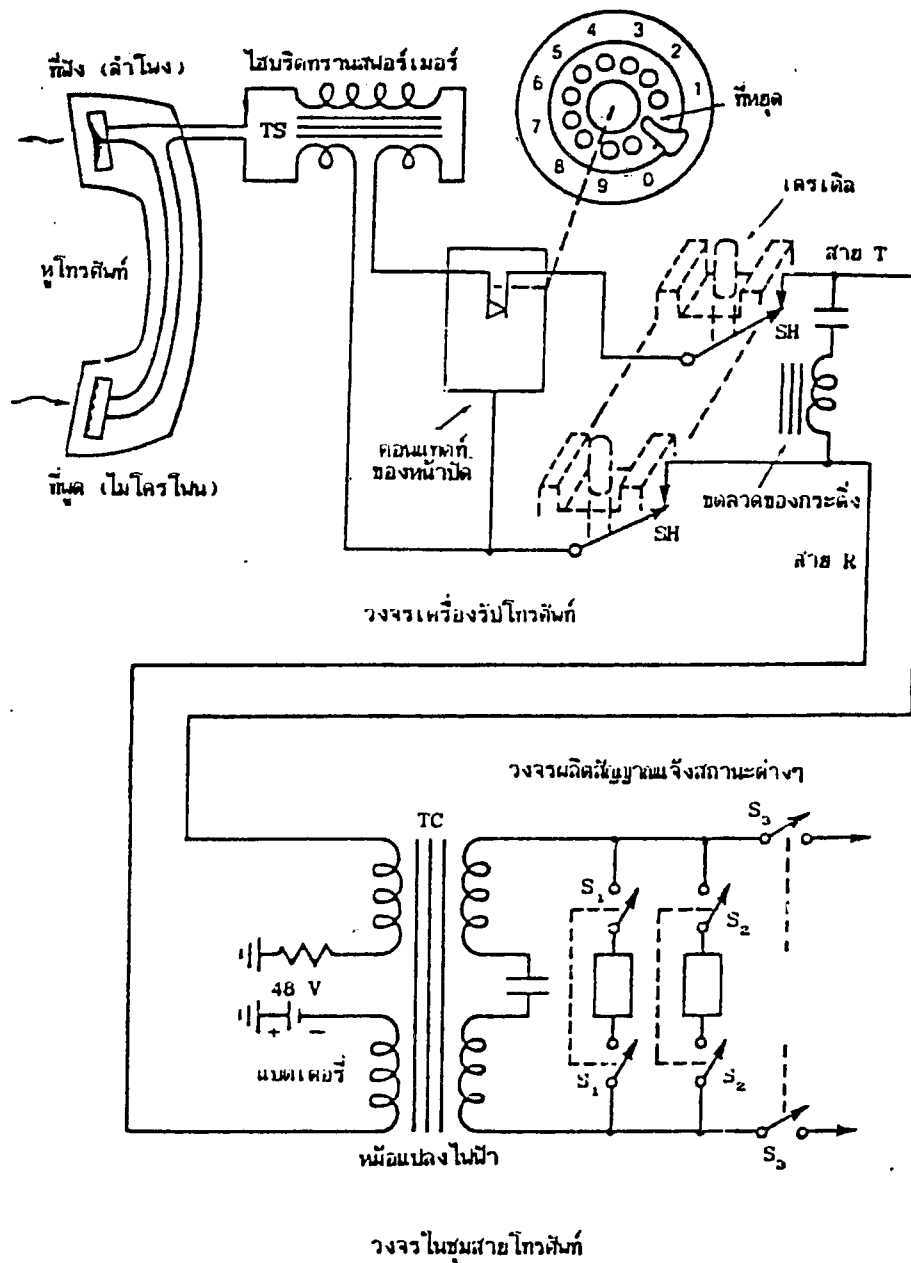
รูปที่ 2-7 การวางสายโทรศัพท์โดยอาศัยชุมสายโทรศัพท์

สถานที่ทำการสลับสายบริการให้แก่ผู้ใช้โทรศัพท์นั้น คือที่ทำการโทรศัพท์ หรือที่เรียกกันว่า "ชุมสายโทรศัพท์" สายที่ต่อระหว่างเครื่องรับโทรศัพท์กับเครื่องสลับสายภายในชุมสายโทรศัพท์นั้น ประกอบรวมกันเป็นวงจรที่เรียกว่า "ลูปผู้เช่า" (subscriber loop) สายที่ต่อเพื่อให้บริการระหว่างชุมสายนั้นเรียกว่า "ทังก์" (trunk) สิ่งที่สำคัญอย่างยิ่งสำหรับการบริการการสลับสาย ก็คือจะต้องมีการกำหนดหมายเลขประจำโทรศัพท์แต่ละเครื่องไว้ก่อน ทั้งนี้เพื่อเป็นการอ้างอิงบอกให้ทราบถึงตำแหน่งของโทรศัพท์เครื่องต่างๆ ที่มีอยู่นั้นว่าได้ถูกจัดเป็นระบบให้อยู่ในระบบการสลับสายนั้น อย่างไรก็ตามเครื่องสลับสายต้องมีความสามารถที่จะแยกได้ว่าโทรศัพท์เครื่องไหนได้ทำการติดต่อเข้ามา ในเวลาใด และต้องการติดต่อไปยังโทรศัพท์เครื่องใด วงจรสลับสายจะทำหน้าที่ต่อสายที่ว่างให้กับผู้ใช้โทรศัพท์ให้สามารถติดต่อกันได้

รูปที่ 2-8 แสดงส่วนประกอบง่ายๆ ของเครื่องรับโทรศัพท์ที่ต่ออยู่กับชุมสายโทรศัพท์ในตู้โทรศัพท์จะมีไมโครโฟนและลำโพง เมื่อยังไม่มีการใช้โทรศัพท์ หูโทรศัพท์ที่วางอยู่จะกดปุ่มเครเดิล (cradle) ที่อยู่ตรงที่วางหูโทรศัพท์ดังแสดงในรูปที่ 2-8 ให้ต่ำลงไปเปิดคอนแทคท์ของสวิตช์ SH ออกจากกัน สวิตช์ SH นี้จะทำหน้าที่ตัดวงจรหูโทรศัพท์ออกไปจากระบบโทรศัพท์ แต่อย่างไรก็ตามในเครื่องรับโทรศัพท์นั้นยังมีวงจรส่วนอื่นที่ยังต่ออยู่กับสายไปยังชุมสายโทรศัพท์ คือวงจรของกระดิ่งไฟฟ้า เมื่อชุมสายโทรศัพท์ต้องการเรียกติดต่อโทรศัพท์เข้ามาทางชุมสายโทรศัพท์สามารถที่จะส่งสัญญาณ AC เข้ามายังขดลวดโซลินอยด์ (solenoid) ได้ขอให้สังเกตว่า กระแส DC จากแบตเตอรี่ 48 V ภายในชุมสายจะไม่สามารถไหลเข้าไปในขดลวดโซลินอยด์ได้ ทั้งนี้เนื่องจากมีตัวเก็บประจุ C ทำหน้าที่กั้นกระแส DC ไว้

เมื่อผู้ใช้โทรศัพท์ยกหูโทรศัพท์ ปุ่มเครเดิลจะเป็นอิสระขึ้นทำให้คอนแทคท์ของสวิตช์ SH นั้นสัมผัสกันดังแสดงในรูปที่ 2-8 ทำให้กระแสไฟ DC จากแบตเตอรี่สามารถไหลเข้าเลี้ยงอุปกรณ์ในตู้โทรศัพท์ให้ทำงานได้ เครื่องสลับสายในชุมสายโทรศัพท์นั้นสามารถที่จะรู้ภาวะของเครื่องรับโทรศัพท์ว่ากำลังถูกใช้หรือไม่ โดยการตรวจสอบดูว่ามีกระแสไฟ DC ไหลอยู่ภายในตู้โทรศัพท์นั้นหรือไม่ ส่วนเครื่องสลับสายจะทำหน้าที่คอยส่งสัญญาณแจ้งภาวะที่เกี่ยวกับการต่อสายโทรศัพท์ที่เหมาะสมผ่านหม้อแปลงซึ่งอยู่ในชุมสายออกไปยังเครื่องรับโทรศัพท์ตามสภาวะของการใช้โทรศัพท์ที่เป็นอยู่ขณะนั้น สำหรับสัญญาณแจ้งภาวะการใช้โทรศัพท์ที่สำคัญแบ่งออกเป็น 3 ชนิด คือ

1. สัญญาณพร้อมให้หมุนโทรศัพท์ (Dial Tone)
2. สัญญาณกระดิ่ง (Ringing Tone)
3. สัญญาณแจ้งว่าสายไม่ว่าง (Busy Tone)



รูปที่ 2-8 วงจรโทรศัพท์

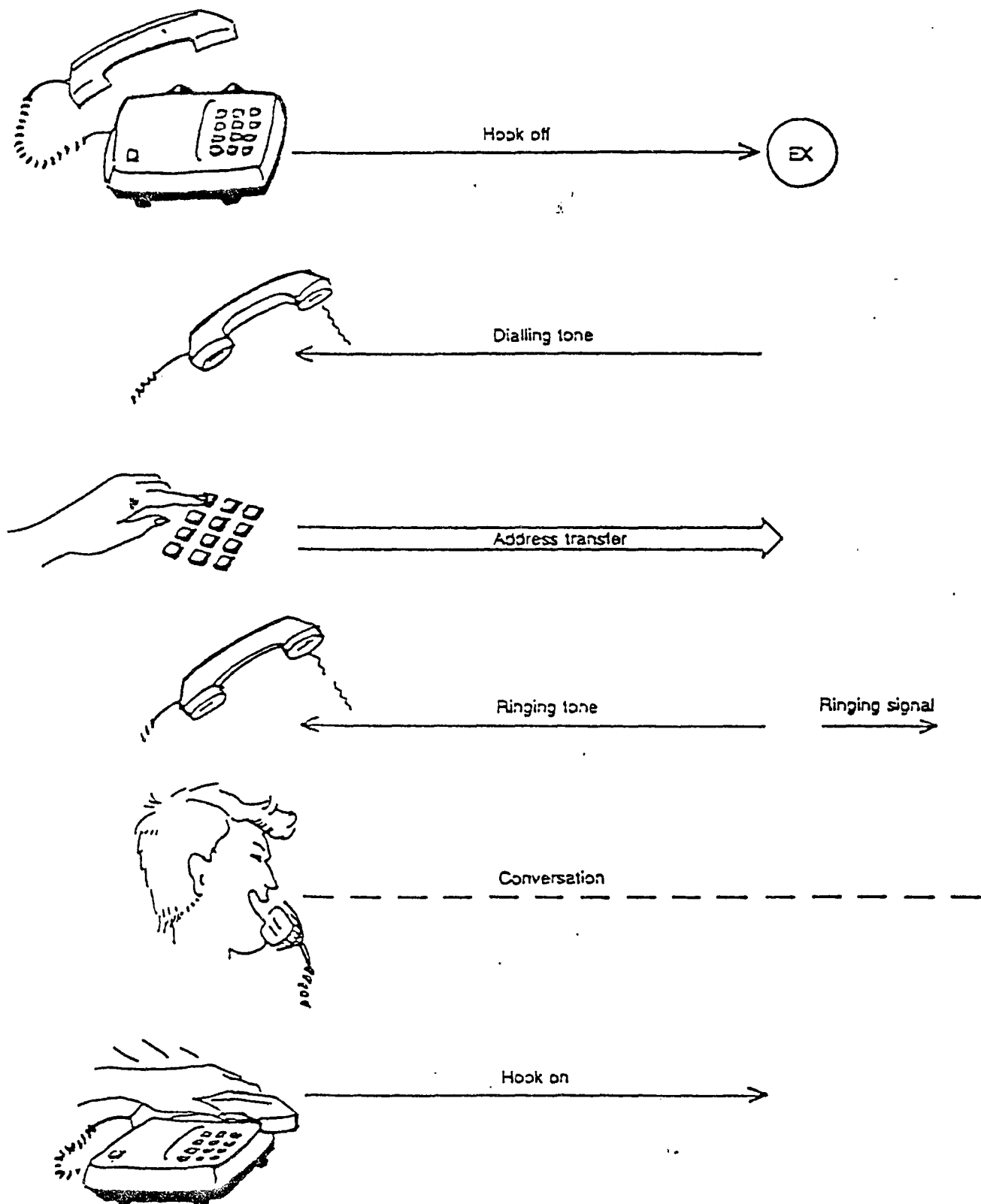
เนื่องจากโทรศัพท์ที่ใช้ระบบหน้าบิดหมุนสำหรับการติดต่อผ่านชุมสายนั้นไม่ค่อยจะ
อำนวยความสะดวกให้กับผู้ใช้โทรศัพท์เท่าใดนักเพราะมันเป็นระบบเชิงกลทำงานค่อนข้างช้า ดังนั้น
ต่อมาจึงได้เกิดมีการคิดต่อโทรศัพท์ชนิดกดปุ่มขึ้น ระบบโทรศัพท์ชนิดกดปุ่มนี้เรียกว่าระบบ DTMF

(Dual Tone MultiFrequency) ทั้งนี้เพราะการกดปุ่มหมายเลขแต่ละปุ่มบนหน้าปัดเครื่องรับโทรศัพท์ นั้น จะทำให้เกิดสัญญาณที่ประกอบขึ้นจากความถี่ 2 ความถี่ ส่งออกไปตามสายโทรศัพท์ไปยังชุมสายเพื่อใช้บอกให้ชุมสายรู้ว่า ผู้ใช้โทรศัพท์นั้นต้องการติดต่อกับโทรศัพท์เครื่องใดแทนที่จะใช้พัลส์ของกระแสจากการหมุนหน้าปัด

2.3.2 การทำงานของโทรศัพท์เบื้องต้น

โทรศัพท์เป็นเครื่องมือทางไฟฟ้า ที่ใช้ในการรับและการส่งข้อมูลข่าวสารโดยโทรศัพท์อาจแบ่งเป็น 2 ชนิด คือแบบกดปุ่มและแบบหมุน แต่หน้าที่ของทั้ง 2 ระบบก็เหมือน ๆ กันจะต่างกันที่แบบกดปุ่มจะส่งสัญญาณออกไปเป็นความถี่ที่แตกต่างกัน ส่วนแบบหมุนจะส่งสัญญาณเป็นจำนวนพัลส์หน้าที่หลัก ๆ ของโทรศัพท์ทั้ง 2 แบบที่เหมือนกันสามารถสรุปได้ดังนี้ คือ

- เครื่องโทรศัพท์จะรับรู้ว่ามิผู้ต้องการใช้โทรศัพท์ เมื่อมีการยกหูโทรศัพท์ขึ้น
- เครื่องใช้โทรศัพท์จะส่งสัญญาณที่เรียกว่า สัญญาณหมุน (dial tone) บอกว่าพร้อมที่จะให้ทำการกดหรือการหมุนหมายเลขที่จะติดต่อกันได้ ซึ่งก็คือ เสียงที่ได้ยินเมื่อเวลายกหูเป็นสัญญาณเสียงที่มีความถี่ 350 เฮิรตซ์ กับ 440 เฮิรตซ์ มอดูเลตรวมกัน
- เครื่องโทรศัพท์จะทำหน้าที่ส่งรหัสหมายเลขที่ผู้เรียกต้องการจะติดต่อด้วยไปยังชุมสาย
- เครื่องโทรศัพท์จะส่งสัญญาณบอกผู้เรียกว่า หมายเลขที่ต้องการจะติดต่อด้วยว่างหรือไม่ ถ้าว่างก็จะส่งสัญญาณกลับ (ring back) ซึ่งความถี่ 44 เฮิรตซ์ มอดูเลตกันมาโดยจะดัง 2 วินาทีแล้วเงียบ 4 วินาที สลับกันไป แต่ถ้าหมายเลขที่ต้องการจะเรียกไม่ว่างก็จะส่งสัญญาณความถี่ 480 เฮิรตซ์ กับ 620 เฮิรตซ์ มอดูเลตกันมา
- สามารถเปลี่ยนรูปพลังงานเสียงเป็นสัญญาณไฟฟ้า และสัญญาณไฟฟ้ากลับมาเป็นพลังงานเสียง
- เครื่องโทรศัพท์จะปรับระดับแรงดันอัตโนมัติในกรณีที่เกิดการเปลี่ยนแปลงของแรงดันขึ้น
- เครื่องโทรศัพท์จะส่งสัญญาณไปยังชุมสาย เพื่อแจ้งให้ทราบว่สิ้นสุดการใช้งานแล้วและให้ชุมสายเลิกทำการต่อกับอีกฝ่ายหนึ่งได้



รูปที่ 2-9 แสดงการใช้งานโทรศัพท์และการส่งสัญญาณ (signalling)

2.3.3 กลไกการเชื่อมต่อวงจร

วงจรพื้นฐานข้างในโทรศัพท์และการเชื่อมต่อกับชุมสายเบื้องต้นมีลักษณะดังนี้คือ ตามรูปที่ 2-8 จะเห็นว่าโทรศัพท์จะเชื่อมต่อกับชุมสายด้วยสายสองเส้นคือ T (TIP) และ R (RING) เมื่อผู้ใช้ยกหูโทรศัพท์ขึ้นแหล่งจ่ายไฟตรงของชุมสาย(48โวลต์) ก็จะถูกต่อเข้ากับวงจรของเครื่องโทรศัพท์โดย ฮุกสวิตช์ (hook-switch) ในส่วนที่เชื่อมต่อระหว่างหูฟัง (ซึ่งรวมทั้งไมโครโฟนด้วย) กับสายโทรศัพท์ก็จะต้องมีหม้อแปลงอัตโนมัติ (auto transformer) ทำหน้าที่ปรับอิมพีแดนซ์ของหูฟังและสายโทรศัพท์ให้สมดุลกันเพื่อให้การรับส่งสัญญาณมีประสิทธิภาพที่สุดรวมไปถึงการทำให้ผู้พูดได้ยินเสียงที่ตัวเองพูดไป (side tone) ในระดับที่เหมาะสมด้วย เมื่อมีการติดต่อกับเครื่องโทรศัพท์กับชุมสายแล้ว ก็จะมีการส่งสัญญาณไปยังอุปกรณ์สวิตช์ซึ่งเพื่อบอกให้รู้ว่าขณะนี้คู่สายนี้ไม่ว่างแล้ว

สำหรับการหมุนหมายเลขโทรศัพท์ก็คือ การส่งสัญญาณพัลส์ (pulse train) ตั้งแต่ 1 ถึง 10 พัลส์ เช่นถ้ามีการส่งพัลส์ 1 พัลส์ หมายถึง การหมุนหมายเลขศูนย์ส่ง 2 พัลส์ ก็หมายถึงเลขหนึ่ง ดังนั้นถ้าหมุนเลข 9 ก็จะมีการส่งพัลส์จำนวน 10 พัลส์ นั่นเอง และความเร็วในการส่งก็คือ 10 พัลส์ต่อวินาที

สำหรับโทรศัพท์ที่ใช้ในการกดปุ่มนั้น ก็จะส่งสัญญาณที่มีค่าความถี่ที่แตกต่างกันออกไปสำหรับแต่ละหมายเลขที่มีอยู่ 10 ตัว ความถี่ที่ส่งออกไปเป็นความถี่ที่อยู่ในย่านความถี่เสียงแต่ว่าในการกดหนึ่งครั้งจะมีสัญญาณเสียงที่มอดูเลตแล้วถูกส่งออกไป 2 ความถี่

ทางชุมสายเมื่อได้รับข้อมูลจากผู้เรียกแล้วก็จะแปลงสัญญาณที่ได้รับมาส่งให้อุปกรณ์สวิตช์ซึ่งทำงานเพื่อทำการต่อสายให้กับผู้เรียก ถ้าปลายสายที่ต้องการติดต่อกับไม่ว่าง ชุมสาย ก็จะส่งสัญญาณเรียก (ringing tone) ไปยังผู้เรียกเพื่อแจ้งให้ทราบว่าสามารถต่อวงจรให้ได้ตามต้องการแล้ว

2.3.4 การสนทนา

เมื่อปลายทางหรือผู้เรียกยกหูโทรศัพท์ขึ้นการทำงานในส่วนควบคุมของชุมสายโทรศัพท์ก็จะหยุดเพื่อที่จะรอทำงานให้กับผู้อื่นที่เรียกเข้ามาต่อไป แต่หน้าที่ของชุมสายสำหรับตอนนี้ก็คือการทำงานของมิเตอร์สำหรับเรียกเก็บค่าบริการในภายหลัง

ในระหว่างที่ทำการสนทนาอยู่เครื่องโทรศัพท์ก็จะทำงาน 2 โหมดไปพร้อมๆกัน คือ แปลงจากสัญญาณไฟฟ้าให้เป็นสัญญาณเสียง(acoustic energy)ซึ่งจะเรียกว่า โหมดรับสัญญาณ (receiver mode) และในทางกลับกัน โหมดที่ทำหน้าที่แปลงจากสัญญาณเสียงให้เป็นสัญญาณไฟฟ้า เรียกว่า โหมดส่งสัญญาณ (transmitter mode) ในโหมดนี้จะมีเรื่องของการป้อนกลับของสัญญาณ เข้ามา

เกี่ยวข้องนั้นก็คือ การที่ผู้พูดสามารถได้ยินเสียงของตนเองจากหูฟังด้วย เรียกเสียงนี้ว่า Side tone ซึ่งจำเป็นที่จะต้องป้อนกลับมา เพราะไม่เช่นนั้นจะไม่สามารถรู้ได้ว่าการพูดให้มีเสียงฟังในระดับใด จึงจะพอเหมาะที่คู่สนทนาได้ยินเสียงพูดของผู้เรียกอย่างชัดเจน

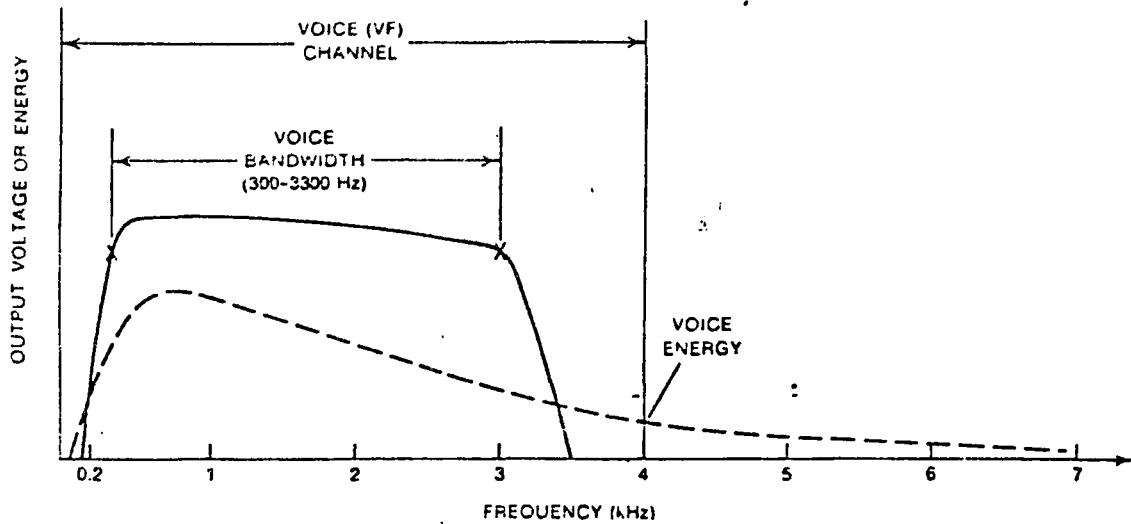
เมื่อสิ้นสุดการสนทนาทั้ง 2 ฝ่าย วางหูโทรศัพท์ลง สัญญาณจากฮุกสวิตช์ ก็จะบอกให้ชุมสายทำการเปิดวงจรที่ทำการติดต่ออยู่ออกอุปกรณ์ต่าง ๆ ก็จะว่างและพร้อมสำหรับการติดต่อครั้งต่อไป

2.3.5 ระบบการส่งสัญญาณในสายส่ง

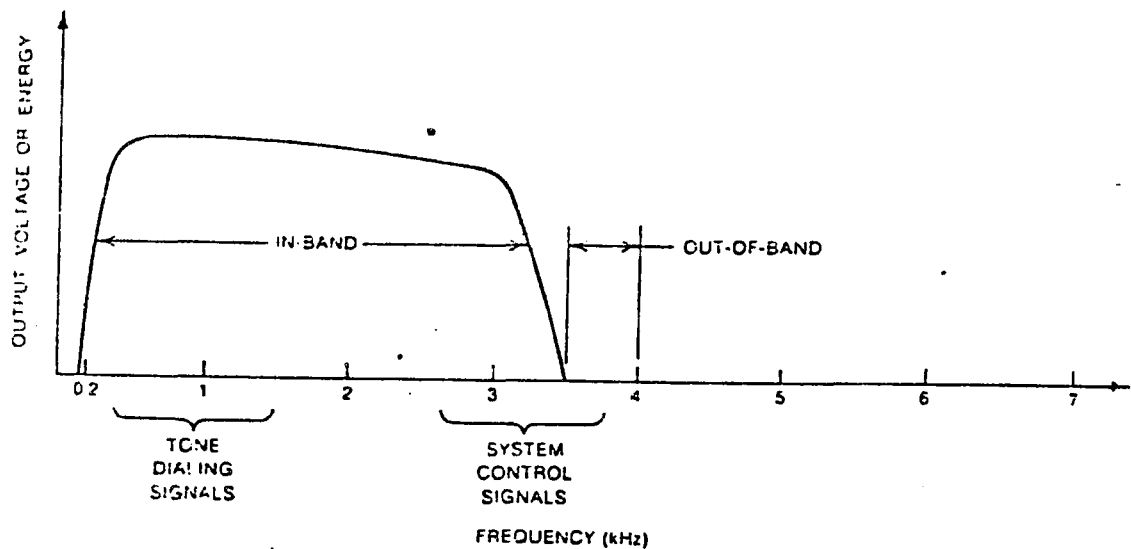
สัญญาณที่ปรากฏในสายส่งสามารถแยกได้เป็น 2 ชนิดใหญ่ ๆ คือ สัญญาณเสียงที่พูดคุยกัน และสัญญาณที่ใช้ในการควบคุมระบบสวิตชิง ซึ่งใช้ในการเชื่อมต่อวงจรระหว่างผู้เรียกและผู้ถูกเรียกนั่นเองรวมทั้งสัญญาณเรียกกลับสัญญาณบอกไม่ว่าง สัญญาณควบคุมอาจจะใช้ได้ทั้งสัญญาณอนาลอก หรือสัญญาณดิจิทัลก็ได้ เนื่องจากโทรศัพท์แบบหมุนกับแบบกดปุ่มมีการส่งรหัสหมายเลขโทรศัพท์ต่างกัน ดังนั้นในการส่งสัญญาณออกไปในสายส่งบางครั้งอาจมีการส่งทั้งสัญญาณดิจิทัลและสัญญาณอนาลอกไปพร้อม ๆ กันก็ได้

2.3.6 สัญญาณเสียงพูด

สัญญาณเสียงพูดจัดเป็นสัญญาณอะนาลอก ดูจากรูปที่ 2-9 เสียงพูดมีแบนด์วิดท์ ตั้งแต่ 100 เฮิรตซ์ ไปจนถึง 6 กิโลเฮิรตซ์ แต่ในความเป็นจริงแล้วเสียงพูดที่ทำให้คนทั้งสามารถฟังและจับใจความได้สบาย ๆ จะอยู่ในช่วง 200 - 4,000 เฮิรตซ์ เท่านั้น วงจรกรองความถี่จึงได้ถูกนำมาใช้เพื่อป้องกันสัญญาณที่ไม่ต้องการเข้ามาภายในระบบ โดยจะยอมให้สัญญาณที่มีความถี่ 0 - 4,000 เฮิรตซ์ สามารถผ่านเข้าไปในระบบได้ และย่านความถี่นี้มีชื่อเรียกว่า ช่องสัญญาณเสียงพูด (voice channel) หรือ VF channel แต่อย่างไรก็ตาม แบนด์วิดท์ของเสียงพูดในการส่งจริงจะอยู่ในช่วง 300 - 3,000 เฮิรตซ์ เท่านั้น ไม่ได้มีการใช้ช่องสัญญาณในการส่งเต็มย่านต่างๆ หลายสัญญาณไม่ว่าจะเป็นสัญญาณหมุน (dial tone) หรือสัญญาณควบคุมต่าง ๆ ก็จะถูกส่งไปในช่วงความถี่นี้ทั้งสิ้น



(ก)



(ข)

รูปที่ 2-10 แสดงแบนวิธของสัญญาณเสียง

(ก) การกระจายพลังงานของสัญญาณ เสียง

(ข) ช่วงของสัญญาณเสียงใน in-band และ out-band

เมื่อกล่าวถึงระดับความดังของเสียงที่ได้ยิน คือ ขนาดแอมพลิจูดของสัญญาณซึ่งสามารถอธิบายได้โดยสมการทางคณิตศาสตร์โดยอธิบายในรูปของพลังงานที่ปรากฏที่โหลด ดังรูปที่ 2-10 เช่น สายโทรศัพท์คู่หนึ่งมีอิมพีแดนซ์ 600 โอห์ม พลังงานที่ปรากฏที่โหลดก็คือ

$$P_{load} = \frac{e_s^2}{600}$$

โดย P_{load} คือ กำลังที่ตกคร่อมโหลด (วัตต์)
 e_s คือ ระดับแรงดันของสัญญาณที่ส่งไป (โวลต์)

ในระบบโทรศัพท์และวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้องกับเสียง มักจะการใช้การเปรียบเทียบ
 กับกำลังขนาด 1 มิลลิวัตต์อยู่เสมอ โดยอยู่ในรูปของเดซิเบล ซึ่งมีสมการดังนี้

$$dB = 10 \log_{10} (P_1/P_2)$$

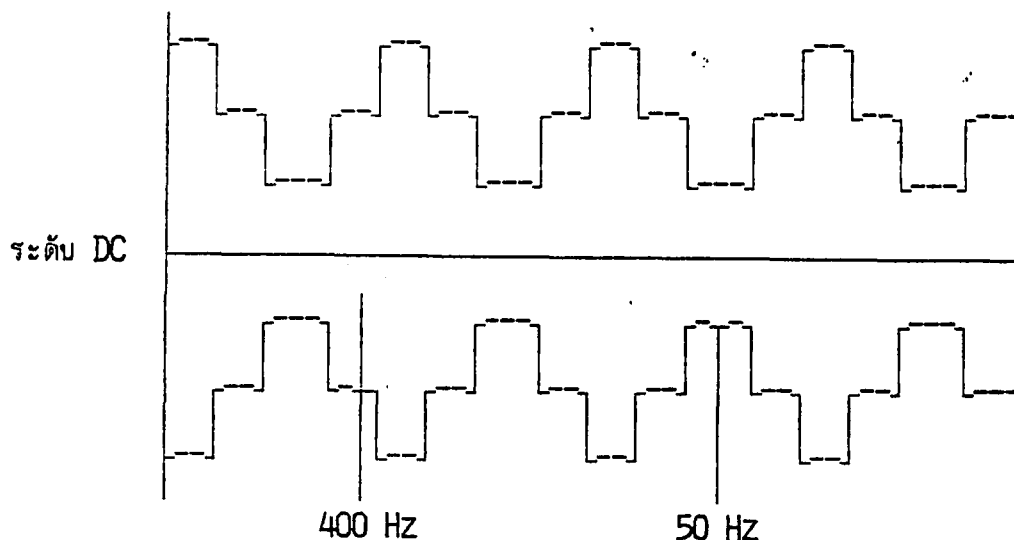
แต่เนื่องจากมักจะใช้ค่า 1 มิลลิวัตต์ เป็นค่าเปรียบเทียบ (ค่า $P_2 = 1$ มิลลิวัตต์ ในสมการ
 นั้นเอง) ก็จะใช้สัญลักษณ์เป็น dBm แทนซึ่งความหมาย คือ การเปรียบเทียบกำลังที่จุดใด ๆ กับกำลัง
 ขนาด 1 มิลลิวัตต์นั่นเอง ในระบบโทรศัพท์ที่ใช้กันจริง ๆ จะมีการกำหนดจุด ๆ หนึ่งในสายส่งให้มี
 ค่า $dB_m = 0$ ซึ่งเรียกจุดนี้ว่า Zero level transmission point (Zero LTP) ซึ่งประโยชน์ที่ได้จากการ
 กำหนดจุด ๆ นี้ขึ้นมาก็คือ สามารถทำให้ทราบที่ระยะต่าง ๆ ที่ห่างจากจุด Zero LTP มีค่ากำลัง
 ของสัญญาณกี่ dBm เมื่อทราบเพียงค่าแรงดันจากการวัดที่ระยะนั้น ๆ

2.3.7 สัญญาณโทรศัพท์มาตรฐาน (Subscriber signalling)

สัญญาณโทรศัพท์ที่เกิดขึ้นในสถานะต่าง ๆ กันจะมีความแตกต่างกัน และเพื่อให้โทรศัพท์
 ทั่วไปสามารถใช้งานร่วมกันได้ จึงมีการวัดมาตรฐานขึ้นให้โทรศัพท์ทั่วโลกใช้มาตรฐานเดียวกันหมด
 อาจเรียกได้ว่า Subscriber signal คือ เป็นสัญญาณที่ชุมสายโทรศัพท์จะแจ้งให้สมาชิกทราบว่าควร
 ปฏิบัติอย่างไรเมื่อได้ยินสัญญาณนั้น โดยทั่วไปแบ่งลักษณะสัญญาณได้ดังต่อไปนี้

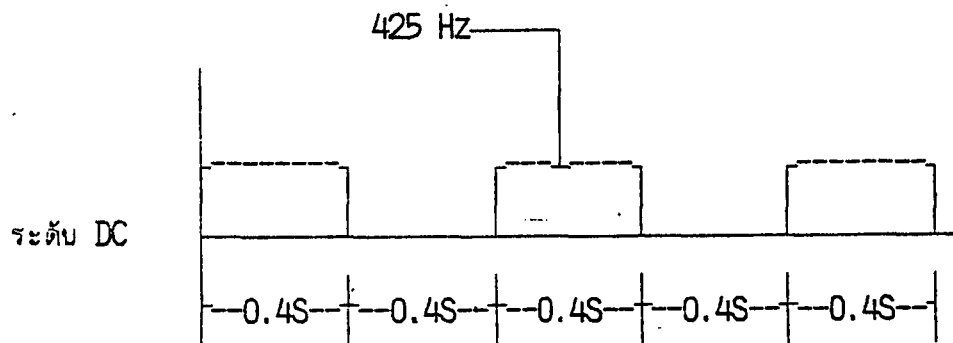
1) สัญญาณไทม์น (DT: Dial Tone) หรือสัญญาณแมวกรน CCITT กำหนดให้ลักษณะ
 สัญญาณเป็นเสียงต่อเนื่องกัน (Continuous) ความถี่ 400 ถึง 450 Hz ผสมกับความถี่ 50 Hz แบบ
 AM (Amplitud Modulation) จะเป็นการบอกให้ผู้เรียก (calling) ทราบว่าขณะนี้ชุมสายโทรศัพท์นั้น ๆ
 พร้อมที่จะรับรู้การหมุนหมายเลขเพื่อเรียกออกและทำการติดต่อออกได้เวลาที่ยกหูจนชุมสายส่ง dial
 tone ให้ใช้เวลาน้อยกว่า 1 วินาที และชุมสายจะส่ง dial tone ให้เป็นเวลา 5 วินาที ถ้าเกินกว่านี้ line
 subscriber จะกลายเป็น line - lockout ลักษณะของสัญญาณ แสดงดังรูปที่ 2-11 เพื่อปลด register

และ code receiver ไปใช้งานกับ line subscriber รายอื่นต่อไป เพราะในแต่ละชุมสาย register และ code receiver มีจำนวนจำกัด ขณะนี้เราจึงไม่สามารถแยกออกได้ ถ้าจะเรียกออกต้องวางหู (hook - on) และยกหู (hook - off) ก็จะได้ dial tone ใหม่



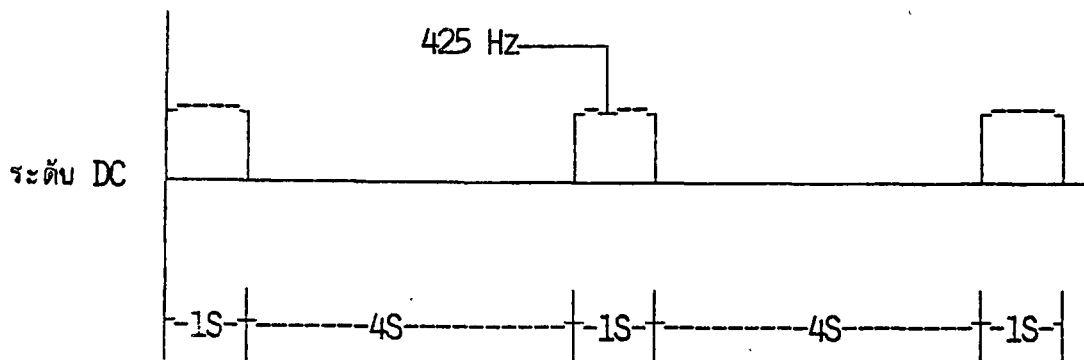
รูปที่ 2-11 แสดงสัญญาณ Dial Tone

2) สัญญาณไม่ว่าง (BT : Busy Tone) ลักษณะของสัญญาณจะดัง-หยุด-ดัง-หยุด เป็นช่วง ๆ เพื่อบอกให้ทราบว่าจะไม่สามารถติดต่อกับปลายทางตามหมายเลขที่หมุนกดได้ เราจึงต้องวางหูก่อน แล้วยกใหม่ เมื่อได้ยิน Dial tone จึงค่อยหมุน, กดหมายเลขที่ต้องการเรียกออกอีกครั้งลักษณะของสัญญาณเป็นความถี่ 400 Hz (เหมือน Dial tone) แต่ดัง 0.4 วินาที หยุด 0.4 วินาที ดังแสดงในรูปที่ 2-12



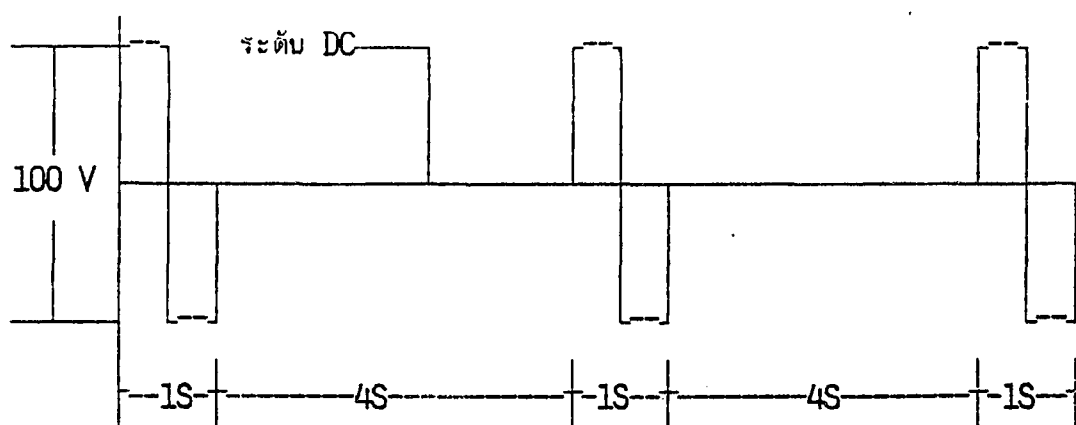
รูปที่ 2-12 แสดงสัญญาณ Busy Tone

3) สัญญาณเรียกกลับ (RBT : Ring Back Tone) เมื่อทำการเรียกออกแล้วชุมสายต่อสายได้เป็นผลสำเร็จ ชุมสายจะส่งสัญญาณไปเรียกทางผู้รับ (Called) คือทำให้กริ่งโทรศัพท์ของผู้รับดังขึ้นพร้อม ๆ กันนั้น ทางด้านผู้เรียกจะมีสัญญาณเรียกกลับจังหวะพร้อม ๆ กับสัญญาณกระดิ่งโดยเสียงนี้จะมีค่าความถี่ 400 Hz ดังนาน 1 วินาที หยุดเงียบ 4 วินาที และชุมสายจะตัดเป็นสัญญาณไม่ว่าง (busy tone) ถ้าภายใน 1.5 นาที ไม่มีคนรับสาย ลักษณะสัญญาณแสดงดังรูปที่ 2-13

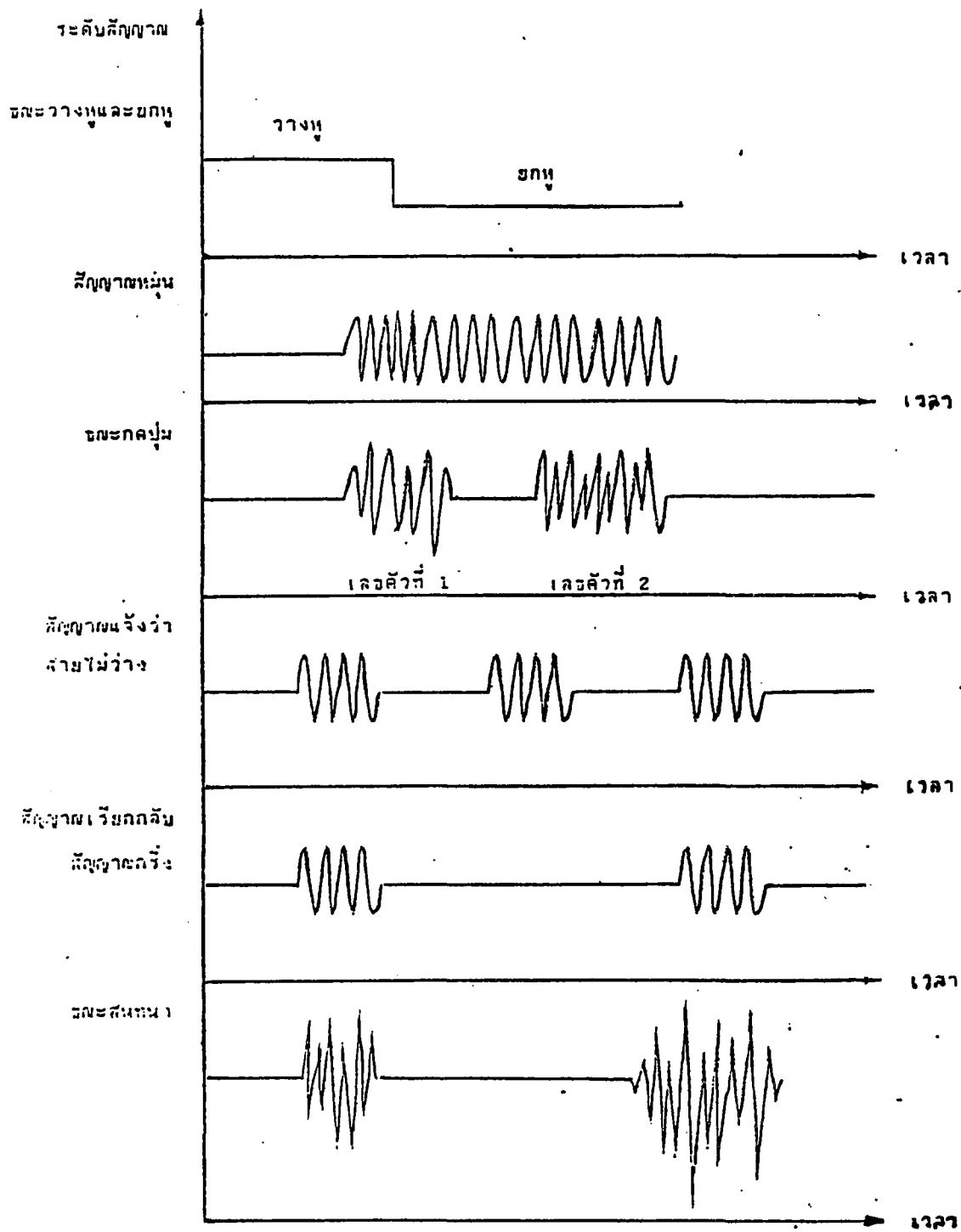


รูปที่ 2-13 แสดงสัญญาณ Ring Back Tone

4) สัญญาณกระดิ่ง (PGT : Ringing Tone) เป็นสัญญาณไฟฟ้าที่ขับให้กระดิ่งในเครื่องโทรศัพท์ดังขึ้นเพื่อให้ทราบว่ามีการโทรเข้า เนื่องจากเป็นสัญญาณที่ใช้ขับให้กระดิ่งดังจึงใช้แรงดันสูงพอสมควร คือ อยู่ในช่วง 90 - 150pp (มักใช้ 100 V) ความถี่ประมาณ 20 - 60 Hz (มักใช้ 25 Hz หรือ 16 Hz, 100 Hz ในบางประเภท) จังหวะการดังหยุดจะพร้อม ๆ กับ Ring Back Tone ของผู้ที่โทรเข้ามา คือดัง 1 วินาที หยุดเงียบ 4 วินาที ลักษณะสัญญาณแสดงดังรูปที่ 2-14



รูปที่ 2-14 แสดงสัญญาณ Ringing Tone



รูปที่ 2-15 แสดงลักษณะสัญญาณต่างๆ ในโทรศัพท์

2.3.8 สัญญาณรบกวน

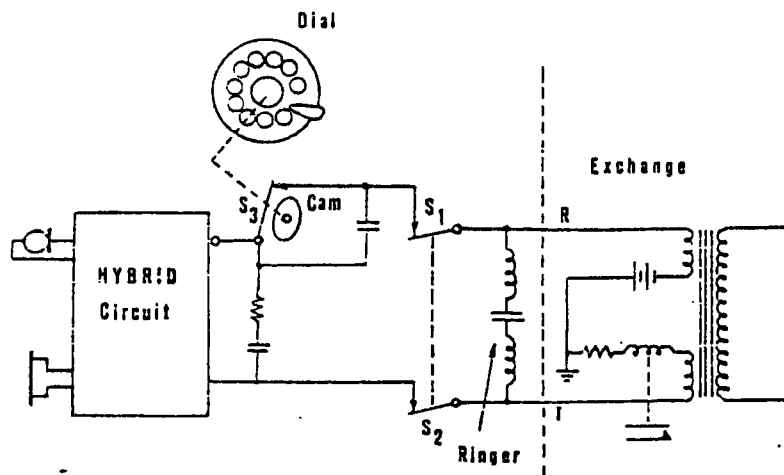
ในระบบใดๆ ของวงจรอิเล็กทรอนิกส์ย่อมจะต้องมีสัญญาณรบกวนเข้ามาเกี่ยวข้องด้วยเสมอ อาจทำให้เกิดความผิดเพี้ยน (distort) ของสัญญาณเสียงพูดได้และสิ่งทำให้เกิดสัญญาณรบกวนขึ้น ก็เป็นสิ่งแวดล้อมรอบ ๆ ข้างนั่นเอง เช่น ความร้อน การสวิตช์ของอุปกรณ์ไทรสเตอร์สายไฟฟ้ากำลัง สูงที่อยู่ใกล้ ๆ กับสายส่งสัญญาณ หรือแม้แต่ข้อต่อของสายที่บกพร่อง สิ่งเหล่านี้ล้วนมีผลให้ ประสิทธิภาพของสัญญาณโทรศัพท์ด้อยลงทั้งสิ้น นอกจากนั้นยังมีสัญญาณรบกวน อีกชนิดหนึ่ง คือ เสียงสะท้อน (echo) ในสายโทรศัพท์ สาเหตุของการเกิดเสียงสะท้อนก็คือ เกิดการไม่สมดุลกัน (mismatching) ระหว่างระบบโทรศัพท์ที่มี 2 สาย กับระบบที่มี 4 สาย และปัญหาเรื่องการสะท้อนจะ มากยิ่งขึ้นถ้าหากกระยะทางระหว่างจุดที่ทำให้เกิดเสียงสะท้อนไกลกันมากๆ แต่โดยปกติแล้วมักจะไม่น่า รู้สึกถึงการสะท้อนกลับของเสียงจนถึงขั้นที่ทำให้รู้สึกรำคาญ

2.3.9 สัญญาณของโทรศัพท์ระบบ DTMF และระบบ PULSE

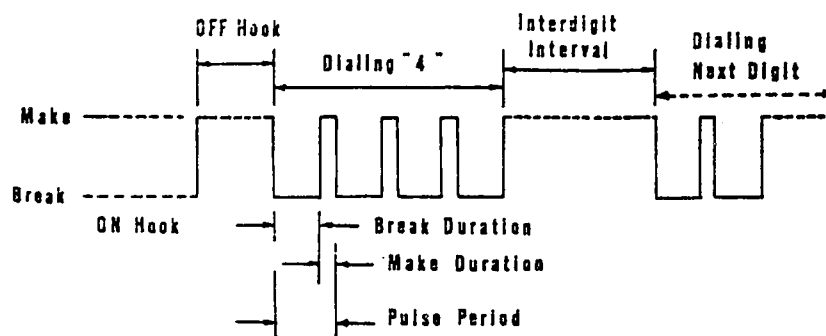
ในการติดต่อกับชุมสายเพื่อให้ทำการติดต่อกับปลายทางที่ผู้เรียกต้องการ ดังนั้นจึงต้องทำให้ชุมสายรับรู้ว่าผู้เรียกต้องการติดต่อกับโทรศัพท์หมายเลขใดโดยการส่งสัญญาณจากเครื่องโทรศัพท์ต้นทางออกไปที่ชุมสายซึ่งเป็นที่ตกลงกันไว้ว่าให้ตัวเลขหนึ่งแทนได้ด้วยสัญญาณไฟฟ้าอย่างหนึ่งเรียกว่า Dial Signalling โดยต้องส่งสัญญาณจนครบหมายเลขทุกตัว สัญญาณดังกล่าวนี้ได้กำหนดเป็นมาตรฐานไว้ 2 แบบ คือ แบบพัลส์ช ้นับดิโทรศัพท์จะเป็นแบบหมุน (Rotary dial) และแบบ 2 ความถี่เสียง (Touch Tone หรือ DTMF: Dual Tone Multi Frequency)

- หน้าปิดแบบหมุน

เมื่อเราหมุน Dial แล้วปล่อยมือ จะทำให้ Cam หรือ ลูกเบี้ยวหมุนส่งผลทำให้ S_3 เปิด - ปิด วงจรจำนวนครั้งเท่ากับเลขที่หมุน จึงทำให้ได้กระแส Impulse กระแสหยุดไหลเมื่อ S_3 ปิดวงจร และกระแสจะไหลเมื่อ S_3 ปิดวงจรนั้นคือ เกิด pulse ความเร็วหน้าปิดต้องอยู่ในมาตรฐานประกอบด้วยความเร็วของกระแส Impulse อัตราส่วนการตัด-ต่อ (Break - Make Ratio) ของ Contact และช่วงเวลาหยุดระหว่างเลขหมาย (Interdigit Interval) ดังรูปที่ 2-14 แสดงการเกิดพัลส์โดยการหมุนเลข 4 ปกติความเร็วของกระแส Impulse ใช้กันอยู่ 2 ค่าคือ 10 PPS (PPS : Pulse Per Second) มาตรฐานในการตัด - ต่อ คือ 2:1 ช่วงเวลาในการหยุดเลขหมายทั่วไปมีค่า 700 ms แต่อาจใช้ได้ในช่วง 600 ms ถึง 900 ms



รูปที่ 2-16 แสดงโทรศัพท์หน้าปัดแบบหมุน



No.4 Dial Pulse

รูปที่ 2-17 แสดงสัญญาณของโทรศัพท์หน้าปัดแบบหมุนเมื่อหมุนหมายเลข 4

- หน้าปัดแบบกดปุ่ม

โดยทั่วไปมี 12 ปุ่ม แบ่งเป็น 4 แถว (Rows) และ 3 หลัก (Columns) แต่สามารถเพิ่มขึ้นได้ถึง 16 ปุ่ม โดยเพิ่มแถวที่ 4 ขึ้นมาอีกได้ ความถี่ที่ใช้แต่ละ Row และ Column จะแตกต่างกัน ทั้ง 4 Rows เรียกว่า กลุ่มความถี่ต่ำและ 3 หรือ 4 Columns เรียกว่ากลุ่มความถี่สูง เมื่อทำการกดปุ่มจะมีความถี่ออกมา 2 ความถี่ เช่น เมื่อกดหมายเลข 7 ได้ความถี่ 852 Hz และ 1209 Hz เป็นต้น

ความผิดพลาดที่ยอมรับเกิดขึ้นได้สำหรับการผลิตความถี่ 1.5 %

ความผิดพลาดที่ยอมรับเกิดขึ้นได้สำหรับการรับเลขหมาย 2.0 %

ความถี่ DTMF ที่ใช้งาน ทดลองค้นคว้าโดย Bell Telephone System เมื่อปี ค.ศ. 1950 ความถี่ต่าง ๆ ที่ใช้งาน ดังรูปที่ 2-18

	1	2	3	A	697	
	4	5	6	B	770	
	7	8	9	C	852	กลุ่มความถี่ต่ำ
	*	0	#	D	941	
กลุ่มความถี่สูง	1209	1336	1477	1633	ความถี่ (Hz)	

รูปที่ 2-18 แสดงหน้าปัดแบบกดปุ่มและความถี่ที่ใช้

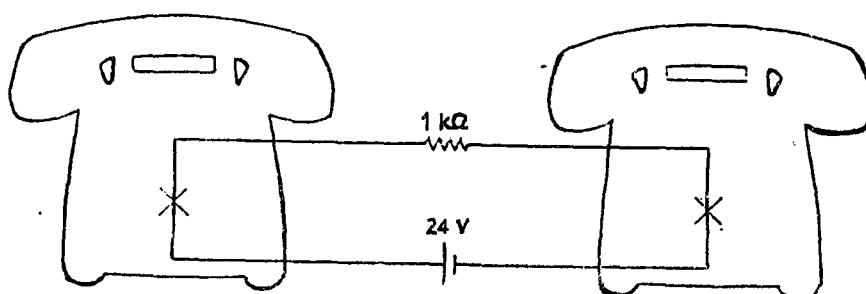
- ข้อเปรียบเทียบของระบบหมุน กับระบบกดปุ่ม

- ระบบ Pulse ใช้เวลาในการหมุนมาก และระยะเวลาในการหมุนแต่ละหลัก (digit) ต้องห่างอย่างน้อย 0.5 วินาที เพื่อให้หมุนสายสามารถแยกแยะตัวเลขที่หมุนได้ ทำให้ในการเรียกออกในครั้งหนึ่งจึงต้องใช้เวลานานในการเรียกออกจนครบทุกหมายเลข

- ระบบ DTMF ใช้เวลาในการกดปุ่มน้อยกว่า ระยะเวลาการกดแต่ละหลัก(digit)จะใช้เวลาอย่างน้อยประมาณ 0.25 วินาที ระยะเวลาห่างกันแต่ละหลักประมาณ 0.25 - 0.5 วินาที
- ความถี่ในระบบ DTMF ทุกความถี่จะอยู่ในย่านความถี่เสียง และเป็นความถี่ IN - BAND ที่ใช้ในระบบโทรศัพท์ ทำให้สามารถส่งไปตามสายโทรศัพท์ได้ดี และมีการสูญเสียน้อย คือ ในย่าน 300 Hz - 3.4 KHz นอกจากนี้ระบบ DTMF สามารถนำมาใช้ในการควบคุมพิเศษต่างๆ ได้ เช่น ระบบการส่งทางโทรศัพท์, ส่งรหัสทางโทรศัพท์ ฯลฯ ในขณะที่ระบบ Pulse ไม่สามารถทำได้

2.3.10 ระบบจ่ายกำลังในโทรศัพท์

โทรศัพท์ทุกเครื่องใช้แรงดันในช่วง 42 ถึง 52 Vdc (Typical 48) ในระยะแรก ๆ ที่เริ่มมีโทรศัพท์จะใช้แรงดันจ่ายที่โทรศัพท์แต่ละเครื่องของ Subscriber แต่ต่อมาได้พัฒนาขึ้นให้มีการจ่ายกำลังที่ Central Office (CE) Central Office นอกจากจะเป็นแหล่งจ่ายกำลังแล้วยังมีการจัดการในระบบส่งสัญญาณต่าง ๆ ด้วย โดยมีอุปกรณ์สำหรับ Transmission , Signalling และ Switching รูปที่ 2-19 แสดงแบบจำลองของการติดต่อสื่อสารทางโทรศัพท์ เมื่อสมาชิก(Subscriber) A ติดต่อกับสมาชิก B

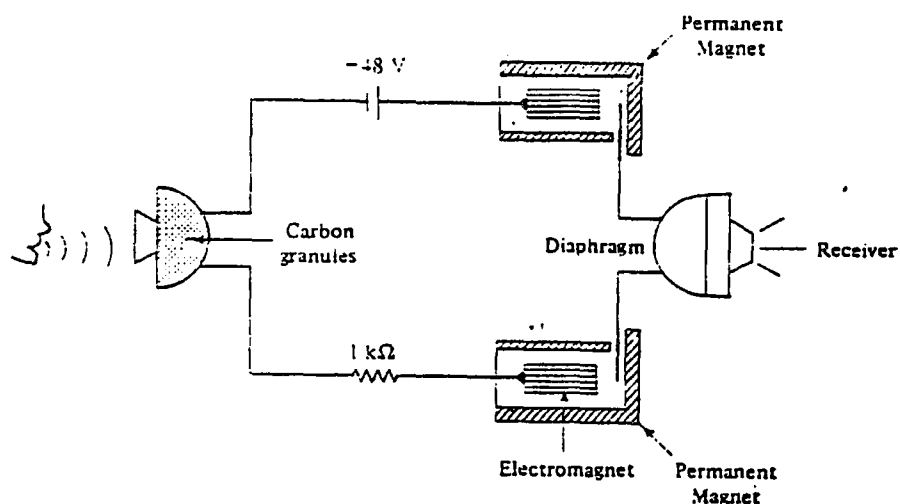


รูปที่ 2-19 แสดงแบบจำลองการติดต่อสื่อสารทางโทรศัพท์

เครื่องส่งโทรศัพท์เมื่อโทรศัพท์อยู่ในสถานะ "off hook" ไฟฟ้ากระแสตรงจาก Central Office จะไหลผ่านเครื่องรับ และเครื่องรับของโทรศัพท์ทั้งสอง เครื่องส่งจะประกอบด้วยแผ่นไดอะเฟรมที่ยึดติดอยู่ระหว่างสารคาร์บอน เมื่อมีการพูดแรงดันของอากาศจะไปสั่นสะเทือนแผ่นไดอะเฟรม

คาร์บอนจะเปลี่ยนค่าความต้านทานแปรไปตามค่าแรงดันของโมเลกุลอากาศซึ่งเกิดจากเสียงพูดนี้เอง เมื่อต่อวงจรดังรูปที่ 2-20 จะทำให้สัญญาณที่ได้จากขดลวดทุติยภูมิของหม้อแปลงเป็นแรงดันไฟฟ้าสลับตามคลื่นเสียง สรุปได้ว่า

1. เสียงที่เกิดจากการสั่นสะเทือนของอากาศ ถูกเปลี่ยนไปเป็นการเปลี่ยนแปลงทางไฟฟ้า
2. ขั้วสาร ข้อมูล ถูกส่งไปในรูปของไฟ AC ที่ควบคุมบนแรงดันไฟ DC
3. กระแสไฟฟ้าที่เปลี่ยนแปลงไปมา จะแปรผันไปตามเสียงของผู้พูด



รูปที่ 2-20 แสดงวงจรโทรศัพท์อย่างง่าย

2.3.11 แรงดันคู่สายโทรศัพท์ในสถานะต่าง ๆ

เมื่อรู้ลักษณะของสัญญาณของระบบโทรศัพท์แล้ว เราจำเป็นต้องรู้ขนาดของแรงดันบนสายโทรศัพท์ในทุกๆสถานะด้วย แล้วเราจะสามารถนำมาตรวจสอบใช้ประโยชน์ ดัดแปลงประยุกต์ใช้งานได้อย่างกว้างขวาง ปัจจุบันมีผู้นำไปใช้ได้อย่างมากมาย นอกจากทำ PABX แล้วยังมีอื่นๆ อีก เช่น ระบบการเตือนภัยทางโทรศัพท์ ระบบสั่งงานเครื่องใช้ไฟฟ้า ระบบแจ้งผลสอบทางโทรศัพท์ และฟังเสียงที่บ้านผ่านคู่สายโทรศัพท์ เป็นต้น

แรงดันต่าง ๆ เมื่อวัดคร่อมคู่สาย ประกติจะมีค่าดังต่อไปนี้

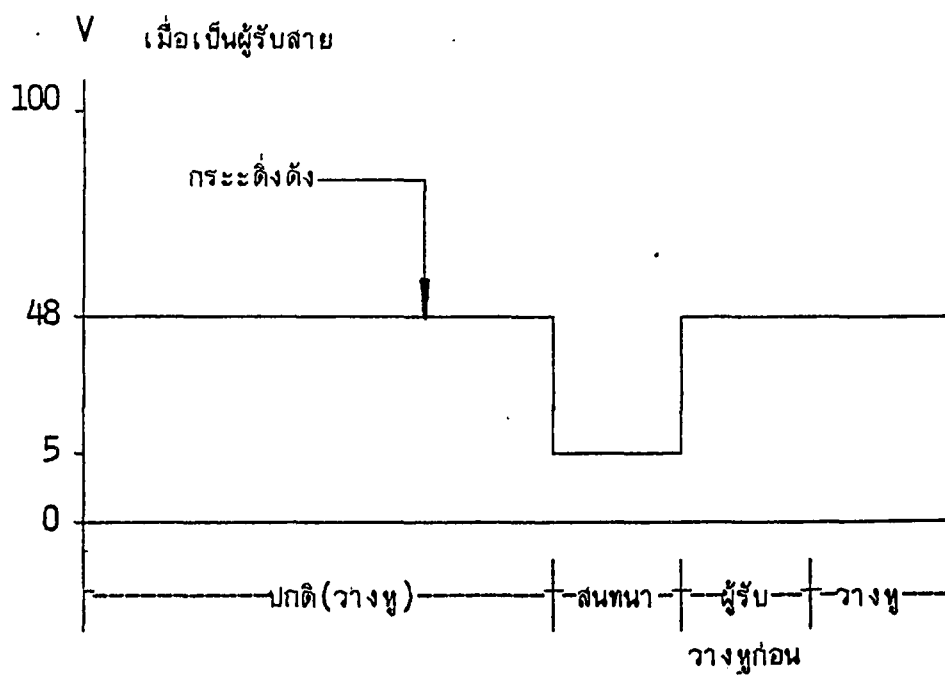
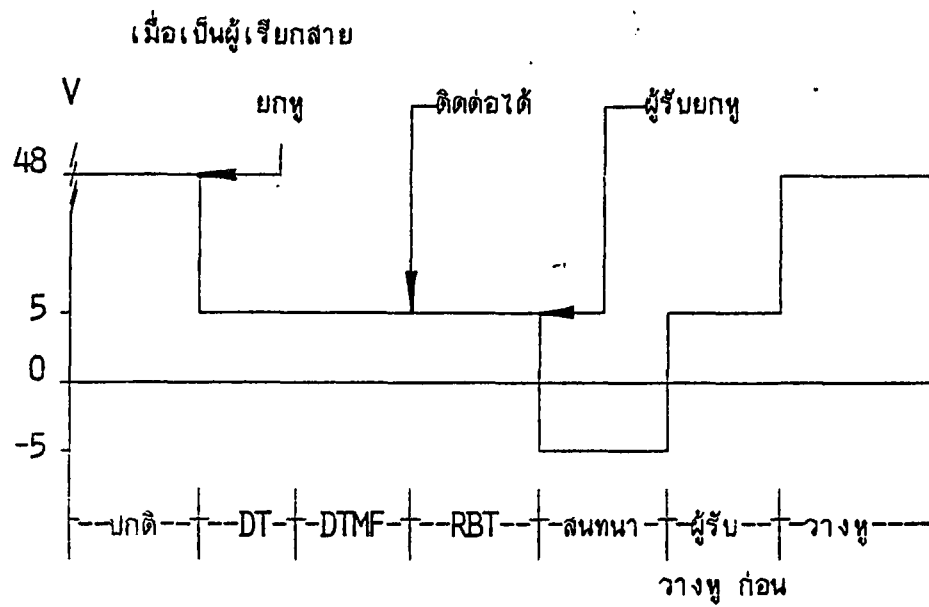
เมื่อวางหูโทรศัพท์ - จะมีแรงดัน DC ประมาณ 48 Volt

เมื่อยกหูโทรศัพท์ - จะมีแรงดัน DC ประมาณ 5-10 Volt ขณะสัญญาณ DT,BT,RTB

- จะมีแรงดัน DC ประมาณ 90-150 Volt ขณะสัญญาณ RTG

Max Loop Resistance = ความต้านทานของโทรศัพท์ + ความต้านทานของสายส่งมีค่าไม่เกิน 1800 โอห์ม

- ถ้าโทรศัพท์ติดต่อกันไม่ได้ สัญญาณจะเป็น BT
- ถ้าโทรศัพท์ติดต่อกันได้ แต่ไม่มีผู้รับสาย เรายังถืออยู่ประมาณ 90 วินาที
- สัญญาณ RTB จะเงียบและจะมีสัญญาณ BT มาแทน
- กรณีที่เราเรียกสายออก เมื่อการสนทนาสิ้นสุดลง ถ้าผู้รับวางหูก่อนในบางชุมสายจะให้สัญญาณ BT หลังจากผู้รับวางหูไปแล้ว 10 นาที แต่ในบางชุมสายจะให้เงียบไปเลย ๆ



รูปที่ 2-21 แสดงแรงดันบนคู่สาย ขณะที่สัญญาณต่างๆเข้ามา

ชนิดของสัญญาณ	การส่งสัญญาณ	ความถี่ (เฮิรตซ์)
สัญญาณพร้อมให้หมุน	ต่อเนื่องไม่ขาดหาย	350 มอดูเลตกับ 440
สัญญาณเรียกกลับ	ดัง 2 วินาที เงียบ 4 วินาที	25
สัญญาณกระดิ่ง	ดัง 2 วินาที เงียบ 4 วินาที	440 มอดูเลตกับ 480
สัญญาณแจ้งว่าสายไม่ว่าง	- ขาดหาย 30 ครั้งต่อนาที เมื่อสายในชุมสายไม่ว่าง - ขาดหาย 60 ครั้งต่อนาที เมื่อเครื่องรับโทรศัพท์ที่ต้องการติดต่อถูกใช้งานอยู่ - ขาดหาย 120 ครั้งต่อนาที เมื่อโทรศัพท์ไม่ว่าง	480 มอดูเลตกับ 620

ตารางที่ 2-1 แสดงลักษณะของสัญญาณต่างๆที่ใช้แจ้งสภาวะการใช้งานโทรศัพท์

ช่วงเวลาการใช้งาน	ระดับสัญญาณไฟกระแสดัง	ระดับสัญญาณไฟกระแลกลับ
ไม่ได้ใช้งาน (ไม่ได้ยกหูฟังขึ้น)	48 โวลต์	
ยกหูฟังขึ้น มีสัญญาณหมุน	10 โวลต์	600 มิลลิโวลต์
ขณะกดหมายเลข	10 โวลต์	ไม่เกิน 0.5 โวลต์
มีสัญญาณแจ้งว่าสายไม่ว่าง	10 โวลต์	400 มิลลิโวลต์
มีสัญญาณเรียกกลับ	10 โวลต์	400 มิลลิโวลต์
มีสัญญาณกระดิ่ง (สำหรับเครื่องผู้รับ)	48 โวลต์	110 โวลต์
มีการพูดระหว่างคู่สายโทรศัพท์	10 โวลต์	ไม่เกิน 1 โวลต์ (สัญญาณเสียง)

ตารางที่ 2-2 ระดับสัญญาณระหว่างคู่สายโทรศัพท์ในช่วงการใช้งาน

2.4 การอินเทอร์เฟส IBM/PC

2.4.1 สัญญาณต่าง ๆ บนสล๊อตของ IBM/PC

ภายใน IBM/PC ได้มีการออกแบบให้สามารถที่จะเพิ่มเติมวงจรอินเทอร์เฟสเข้าไปในภายหลังได้ โดยผ่านทางสล๊อตที่อยู่บนเมนบอร์ด สำหรับสล๊อตบนเมนบอร์ดนี้จะมีจำนวน 5 สล๊อต (สำหรับใน IBM PC/XT จะมี 8 สล๊อต) ซึ่งแต่ละสล๊อตจะมีจำนวนทั้งสิ้น 32 ขา แบ่งออกเป็น 2 ข้าง ๆ ละ 31 ขา ส่วนการเรียกตำแหน่งขาของสล๊อตเหล่านี้จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล๊อตโดยขาที่อยู่ทางด้านซ้ายของสล๊อตจะเรียกโดยใช้อักษร "B" นำหน้าเลขตำแหน่งของขา เช่น ขา B16 ก็คือขาทางด้านซ้ายของสล๊อต (นับจากทางด้านท้ายของเครื่อง) ส่วนขาทางที่อยู่ทางด้านขวาของสล๊อตจะเรียกโดยใช้อักษร "A"

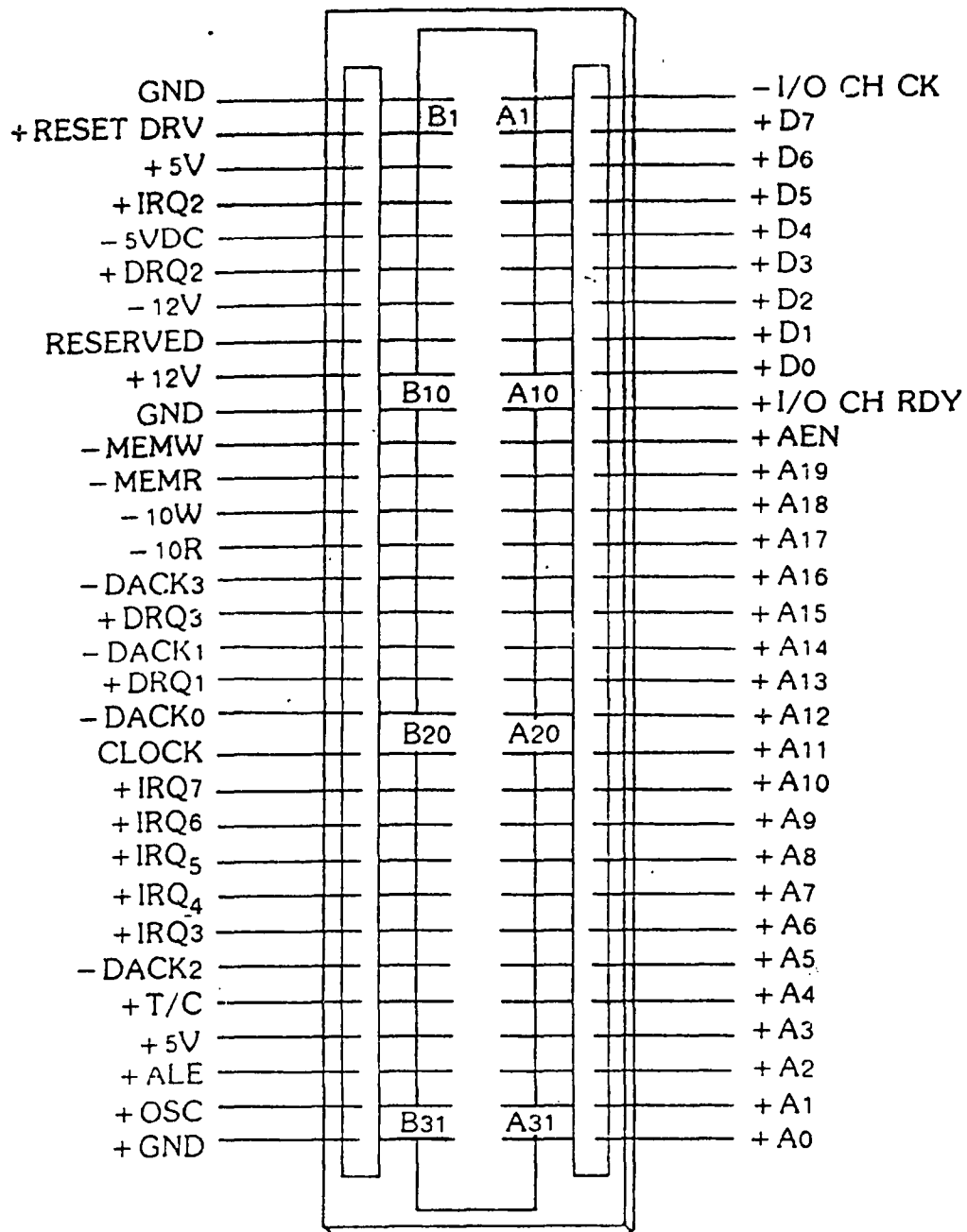
แต่ละขาของสล๊อตเหล่านี้จะเชื่อมต่อกับสัญญาณต่าง ๆ บนเมนบอร์ด ทำให้การสร้างวงจรอินเทอร์เฟสกับ IBM/PC สามารถทำได้สะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขาของสล๊อตเหล่านี้จะประกอบไปด้วย เส้นสัญญาณของบัสแอดเดรส (Address Bus) , บัสข้อมูล (Data BUs) , บัสควบคุมสำหรับการเขียน / อ่านข้อมูลจากหน่วยความจำหรือพอร์ต I/O , เส้นสัญญาณสำหรับการขออินเทอร์รัพท์สำหรับวงจรอินเทอร์เฟส , เส้นสัญญาณสำหรับการขอ DMA , สัญญาณฐานเวลา (Timing Signal) ต่าง ๆ ที่ใช้ในระบบ , เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำ และสัญญาณการตรวจสอบความผิดพลาด (I/O Check)

นอกจากเส้นสัญญาณเหล่านี้ สล๊อตบนเมนบอร์ดยังเชื่อมต่อกับแหล่งจ่ายไฟต่าง ๆ ที่ใช้ในระบบด้วย คือ +5Vdc , -5Vdc , +12Vdc , -12Vdc

- รายละเอียดเกี่ยวกับสัญญาณต่าง ๆ

OSC (Oscillator ขา B30)

ขานี้เป็นเอนาท์พุทท์ที่เชื่อมต่อกับสัญญาณที่มีความถี่สูงสุดบนเมนบอร์ด คือ 14.31818 Mhz ซึ่งมีคาบเวลาประมาณ 70 nanosec. และมี Duty Cycle (ช่วงเวลาใน 1 คาบที่สัญญาณคล็อกมีลอจิก เป็น "1" หารด้วยคาบเวลาทั้งหมด) ประมาณ 50% สัญญาณคล็อกอื่น ๆ ของระบบ เช่น คล็อกที่ป้อนให้กับ 8088 หรือ ชิพสัพพอร์ตต่าง ๆ นั้นจะถูกสร้างขึ้นโดยการหารสัญญาณคล็อกนี้ อย่างไรก็ตามสิ่งหนึ่งที่จะต้องคำนึงถึงในการหารสัญญาณคล็อกนี้ คือ สัญญาณคล็อกนี้จะไม่ Synchronize กับสัญญาณอื่น ๆ บนบัสของระบบ ดังนั้นจึงไม่ควรที่จะนำสัญญาณจากขา Osc นี้ไปใช้เป็นสัญญาณคล็อกสำหรับวงจรภายนอกอื่น ๆ ที่ทำงานร่วมกับระบบ



รูปที่ 2-22 ขาสัญญาณต่างๆใน I/O Slot บนไมโครคอมพิวเตอร์

CLK (Clock; ขา B20)

ขาสัญญาณนี้เป็นเอาต์พุตซึ่งต่อกับสัญญาณคล็อกที่ถูกสร้างขึ้นโดยการหารสัญญาณ OSC ด้วย 3 ทำให้ได้ความถี่ประมาณ 4.77 MHz หรือมีช่วงเวลาใน 1 คาบ (ช่วงเวลาของคล็อก 1 ลูก) เท่ากับ 210 nanosec สำหรับค่า Duty Cycle ของสัญญาณนี้จะมีค่าประมาณ 1/3 คือ ใน 1 คาบจะมีช่วงเวลาที่เป็นลอจิก "1" เท่ากับ 1/3 ของคาบเวลาทั้งหมด หรือประมาณ 70 nanosec และช่วงเวลาที่เป็นลอจิก "0" เท่ากับ 2/3 ของคาบเวลาทั้งหมด หรือประมาณ 140 nanosec. สัญญาณนี้เป็นสัญญาณที่ถูกใช้เป็นคล็อกของระบบ

RESET DRV (ขา B2)

ขาสัญญาณนี้เป็นเอาต์พุต ซึ่งจะเอกทีฟ (ลอจิก "1") ในช่วงที่เราเริ่มจ่ายไฟให้กับระบบ และจะยังคงเอกทีฟไปจนกว่าระบบต่าง ๆ ภายใน IBM/PC จะพร้อมที่จะทำงานได้จากนั้นสัญญาณนี้ก็เปลี่ยนกลับเป็นลอจิก "0" นอกจากนี้ในระหว่างการทำงานของ IBM/PC ถ้าระดับแรงดันของแหล่งจ่ายไฟตกลง สัญญาณนี้ก็จะถูกทำให้เอกทีฟเช่นกัน โดยทั่วไปแล้วสัญญาณนี้จะถูกนำไปใช้ในการรีเซ็ตวงจรอินเทอร์เฟซหรืออุปกรณ์ I/O ต่าง ๆ ในช่วงที่เริ่มจ่ายไฟให้กับระบบ ซึ่งจะเป็นการทำให้อุปกรณ์หรือวงจรเหล่านี้ถูกปรับให้อยู่ในสถานะที่แน่นอน ก่อนที่จะเริ่มต้นการทำงานในระบบ (สถานะนี้เป็นสถานะที่เราควรทราบ และต้องการให้วงจรทำงานในขณะที่ระบบถูกรีเซ็ต)

A0 - A19 (Address Bus; ขา A31-A12)

ขาสัญญาณทั้ง 20 ขานี้เป็นเอาต์พุต ซึ่งใช้ในสำหรับกำหนดแอดเดรสของหน่วยความจำหรืออุปกรณ์ I/O ที่ 8088 ต้องการติดต่อด้วย โดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำสุด (Least Significant Bit) และ A19 จะมีนัยสำคัญสูงสุด (Most Signification Bit) สำหรับค่าแอดเดรสบนบัสแอดเดรส A0 - A19 นี้ จะถูกกำหนดโดย 8088 ในระหว่างขบวนการอ่าน / เขียนหน่วยความจำหรืออุปกรณ์ I/O แต่ในช่วงของขบวนการ DMA นั้น DMA Controller จะเป็นผู้กำหนดค่าแอดเดรสบนบัสแอดเดรสเอง (ในระหว่างนี้ 8088 จะถูกตัดออกจากระบบ)

จะเห็นได้ว่าจำนวนเส้นแอดเดรสจะมีอยู่ 20 เส้น ซึ่งสามารถที่จะอ้างแอดเดรสของหน่วยความจำได้ถึง 1Mbyte แต่อย่างไรก็ตามจะมีแอดเดรสบางแอดเดรสที่ถูกใช้งานโดย IBM/PC อยู่แล้ว คือแอดเดรสของหน่วยความจำ RAM บนเมนบอร์ดที่ถูกใช้โดยระบบ จำนวน 64 Kbyte (สำหรับ IBM/XT จะเป็น 64 Kbyte) และแอดเดรสสำหรับหน่วยความจำ ROM อีก 48 Kbyte ซึ่งถูกใช้ในส่วนของแอดเดรสบนสุดใน 1 Mbyte คือ 0FC00H จนถึง 0FFFFH (สำหรับ IBM/XT จะเป็น 64 Kbyte)

สำหรับการอ้างแอดเดรสของพอร์ต I/O นั้น จะใช้แอดเดรสเพียง 16 เส้น คือ A0 - A15 ซึ่งจะทำให้อ้างแอดเดรสของพอร์ตได้ 64K พอร์ต โดยผ่านทางชุดคำสั่ง IN และ OUT ส่วนเส้น แอดเดรสที่เหลือ คือ A16 - A19 นั้นจะไม่ถูกใช้งาน อย่างไรก็ตามภายใน IBM/PC จะใช้เส้นแอดเดรสในการอ้างแอดเดรสของพอร์ตเพียง 10 เส้น คือ จาก A0 - A9 และค่าแอดเดรสที่ใช้งานจะต้องอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น

D0 - D7 (Data Bus : ขา A9-A2)

ขาสัญญาณนี้จะเป็นแบบ Bi-direction ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ต I/O กับ IBM/PC โดยบิต D0 จะมีนัยสำคัญต่ำสุดและบิต D7 จะมีนัยสำคัญสูงสุด

สำหรับในบัสไซเคิลของการเขียนข้อมูลที่สร้างขึ้นโดย 8088 นั้น ข้อมูลจะถูกส่งออกมาทางบัสข้อมูลก่อนที่สัญญาณ $\overline{IOW}$ (ในกรณีที่ต้องการส่งข้อมูลให้กับพอร์ต) หรือ $\overline{MEMW}$ (ในกรณีที่ต้องการส่งข้อมูลให้กับหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็น ลอจิก "1" (ขอบขาขึ้น) ซึ่งโดยทั่วไปขอบขาขึ้นของสัญญาณ $\overline{IOW}$ หรือ $\overline{MEMR}$ นี้ จะถูกใช้เพื่อสั่งให้พอร์ต I/O หรือหน่วยความจำที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลไปเก็บไว้

สำหรับในบัสไซเคิลของการอ่านข้อมูลที่สร้างขึ้นโดย 8088 นั้น พอร์ต I/O หรือหน่วยความจำที่ถูกอ้างถึง จะส่งข้อมูลออกมาบนบัสข้อมูลก่อนที่สัญญาณ $\overline{IOR}$ (ในกรณีที่ต้องการอ่านข้อมูลจาก พอร์ต) หรือสัญญาณ $\overline{MEMR}$ (ในกรณีที่ต้องการอ่านข้อมูลจากหน่วยความจำ) จะเปลี่ยนจาก ลอจิก "0" เป็นลอจิก "1" (ขอบขาขึ้น)

ALE (address Latch Enable : ขา B28)

ขาสัญญาณนี้เป็นสัญญาณเอาท์พุทที่ 8288 Bus Controller สร้างขึ้นเพื่อใช้สำหรับแสดงการเริ่มต้นของบัสไซเคิล และแสดงให้อุปกรณ์ภายนอกทราบว่าแอดเดรสที่ 8088 ต้องการติดต่อด้วยนั้นถูกส่งออกมาบนบัสแอดเดรสแล้ว โดยที่สัญญาณ ALE นี้ จะเปลี่ยนจากลอจิก "1" เป็น "0" เมื่อค่าแอดเดรสที่ต้องถูกส่งออกมาบนบัสข้อมูลเรียบร้อยแล้ว ดังนั้นขอบขาลงของสัญญาณ ALE นี้จะถูกใช้ในการแลทช์ค่าแอดเดรสจากบัสแอดเดรส / ข้อมูล (Address/Data Bus : AD0 - AD7) ของ 8088 ทำให้สามารถแยกค่า (A0 - A19) และข้อมูล (A0 - A7) ออกจากกันได้ อย่างไรก็ตามสัญญาณ ALE จะแอกทีฟเฉพาะในบัสไซเคิลที่สร้างขึ้นโดย 8088 เท่านั้น โดยจะไม่แอกทีฟในระหว่างขบวนการ DMA

I/O CHECK (I/O Channel Check : ขา A1)

ขาสัญญานนี้เป็นอินพุตที่ใช้ในการแสดงความผิดพลาดเกี่ยวกับพาริตี ที่เกิดขึ้นในการทำงานของวงจรอินเทอร์เฟซหรืออุปกรณ์ I/O เมื่อขาสัญญานนี้ได้รับลอจิก "0" จะทำให้ 8088 ถูกอินเทอร์รัพท์แบบ Non-Maskable (NMI) อย่างไรก็ตามเราสามารถที่จะทำให้วงจรภายในของ IBM/PC ทำการขออินเทอร์รัพท์ (เมื่อได้รับสัญญาณ I/O CHECK หรือไม่ก็ได้ โดยการกำหนดลอจิกของบิตข้อ มูลของพอร์ทที่ควบคุมการขออินเทอร์รัพท์ คือ บิต D7 ของพอร์ท 00A0H ในกรณีที่บิต D7 ของ พอร์ท 00A0H ถูกเซตเป็น "1" ทำให้วงจรภายนอกขออินเทอร์รัพท์แบบ NMI ได้ (Enable) แต่ถ้าบิต D7 ถูกเซตเป็น "0" ก็คือจะเป็นการดิสเอเบิล (Disable) การขออินเทอร์รัพท์แบบ NMI ดังนี้

Enable : ใช้คำสั่ง OUT ส่งข้อมูล 80H ไปยังพอร์ท 00A0H

Disabie : ใช้คำสั่ง OUT ส่งข้อมูล 00H ไปยังพอร์ท 00A0H

และเนื่องจากยังมีอุปกรณ์อื่นที่สามารถขออินเทอร์รัพท์แบบ NMI ได้อีก ดังนั้นซอฟต์แวร์ที่ใช้งานจะต้องสามารถตรวจสอบว่าการขออินเทอร์รัพท์นั้นเกิดจากแหล่งใดได้ด้วย

I/O CHRDY (I/O Channel Ready : ขา A10)

ขาสัญญานนี้เป็นอินพุตที่ใช้เพิ่มช่วงเวลาในบัสไซเคิลในกรณีที่อุปกรณ์ I/O หรือหน่วยความจำที่เกี่ยวข้องกับขบวนการในบัสไซเคิลที่เกิดขึ้นนั้น ไม่สามารถทำงานทันตามช่วงเวลาปกติของบัสไซเคิลนั้น ๆ ได้ (ช่วงเวลาของบัสไซเคิลที่เกี่ยวกับหน่วยความจำใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 4 ลูก หรือ 840 nanosec. ในขณะที่บัสไซเคิลที่เกี่ยวกับ I/O จะใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 5 ลูก หรือ 1.05 microsec.) เมื่ออุปกรณ์ I/O หรือหน่วยความจำต้องการที่จะเพิ่มช่วงเวลาในบัสไซเคิลให้นานขึ้นอีกนั้น จะสามารถทำได้โดยการป้อนลอจิก "0" ให้กับขา I/O CHRDY ในช่วงเวลาที่ I/O หรือ หน่วยความจำที่ถูกกำหนดนั้น ได้รับสัญญาณจากการดีโค้ดแอดเดรส และสัญญาณ MEMR , MEMW , IOR , IOW แยกที่พ

IRQ2-IRQ7 (interrupt request 2 through 7 : ขา B4 และ B25-B21)

ขาสัญญานทั้ง 6 นี้เป็นขาอินพุตที่ใช้สำหรับการขออินเทอร์รัพท์จาก 8088 โดยสัญญาณเหล่านี้จะต้องต่อเข้ากับ 8259A บนเมนบอร์ดโดยตรง โปรแกรมในส่วน BIOS ของ IBM/PC จะทำการโปรแกรม 8259A ให้ IRQ2 มีนัยสำคัญสูงสุด (Highest Priority) และ IRQ7 มีนัยสำคัญต่ำสุด (Lowest Priority) ในกรณีที่มีการขออินเทอร์รัพท์เกิดขึ้น คือ ระดับลอจิกที่ขา IRQ ขาใดขาหนึ่งถูกเปลี่ยนจากลอจิก "0" เป็น "1" (ขอบขาขึ้น) 8259A ก็จะมีการส่งสัญญาณ INT ให้กับ 8088 เพื่อทำการขออินเทอร์รัพท์

สิ่งสำคัญในการขออินเทอร์รัพต์ผ่านทาง IRQ2-IRQ7 นี้ คืออุปกรณ์ที่ทำการขออินเทอร์รัพต์ โดยผ่านทาง IRQ ใดก็จะต้องรักษาระดับสัญญาณที่ขา IRQ นั้น ให้แอกทีฟ (ลอจิก 1) อยู่จนกว่า จะได้รับสัญญาณ INTA (Interrupt Acknowledge) จาก 8088 เสียก่อน ถ้าไม่เช่นนั้นการขออินเทอร์รัพต์จะถูกยกเลิก และอินเทอร์รัพต์ Level 7 (IRQ 7) ก็จะถูกสร้างขึ้นโดยอัตโนมัติ ไม่ว่าการขออินเทอร์รัพต์ที่ถูกยกเลิกนั้น จะเป็นการขออินเทอร์รัพต์ใน Level หรือขาใด ๆ

$\overline{IOR}$ (I/O Read : ขา B14)

ขาสัญญาณนี้เป็นเอาต์พุตแอกทีฟที่ลอจิก 0 ที่สร้างขึ้นโดย 8088 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้ เป็นบัสไซเคิลของการอ่านข้อมูลจากพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูล โดยข้อมูลจะต้องถูกส่งออกมาบนบัสข้อมูลก่อนขอบขาขึ้นของสัญญาณ $\overline{IOR}$ ประมาณ 30 nanosec เพื่อให้มั่นใจว่า 8088 สามารถรับข้อมูลได้ถูกต้อง สำหรับขบวนการ DMA 8237A-5 DMA Controller จะทำการสร้างสัญญาณ $\overline{IOR}$ เอง โดยที่ค่าแอดเดรสที่อยู่บนบัสแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำ (แทนที่จะเป็นแอดเดรสของพอร์ท I/O) ที่พอร์ท I/O ที่ขอ DMA ต้องการจะนำข้อมูลไปเก็บ การที่พอร์ทใดจะส่งข้อมูลออกมาบนบัสข้อมูลนั้น จะอาศัยสัญญาณ DACK จาก DMA Controller เป็นตัวกำหนด เช่นกรณีที่สัญญาณ DACK1 แอกทีฟก็จะแสดงว่าพอร์ท I/O ที่ต้องจะต้องส่งข้อมูลออกมาบนบัสข้อมูลก็คือพอร์ท I/O ที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1)

$\overline{IOW}$ (I/O Write : ขา B13)

ขาสัญญาณนี้เป็นเอาต์พุตแอกทีฟที่ลอจิก 1 ซึ่งถูกสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้แสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้ อย่างไรก็ตามในช่วงเวลาที่สัญญาณ $\overline{IOW}$ นี้แอกทีฟ (ลอจิก 0) ข้อมูลบนบัสข้อมูลอาจจะยังไม่สมบูรณ์ ดังนั้นในการออกแบบจึงควรใช้ขอบขาขึ้นของสัญญาณ $\overline{IOW}$ แทนขอบขาลงในการทำให้พอร์ท I/O ที่เกี่ยวข้องรับข้อมูลไปเก็บไว้ เพื่อให้ข้อมูลบนบัสข้อมูลสมบูรณ์เสียก่อน สำหรับในขบวนการ DMA นั้น DMA-Controller จะทำการสร้างสัญญาณ $\overline{IOW}$ เอง โดยที่ค่าแอดเดรสที่อยู่บนบัส แอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำที่พอร์ท I/O ที่ขอ DMA ต้องการอ่านข้อมูล

$\overline{MEMW}$ (Memory Write : ขา B11)

ขานี้เป็นเอาต์พุตแอกทีฟที่ลอจิก "0" ซึ่ง 8288 สร้างขึ้นในระหว่างบัสไซเคิลในการเขียน

ข้อมูลลงในหน่วยความจำของ 8088 สัญญาณ $\overline{\text{MEMW}}$ นี้จะถูกส่งออกมาเพื่อให้หน่วยความจำที่แอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้ โดยทั่วไปหน่วยความจำจะรับข้อมูลในช่วงขอบขาขึ้นของสัญญาณ $\overline{\text{MEMW}}$

$\overline{\text{MEMR}}$ (Memory Read : ขา B12)

ขานี้เป็นเอาต์พุตจาก 8288 ซึ่งจะแอกทีฟ (ลอจิก 0) ในระหว่างบัสไซเคิลของการอ่านข้อมูลจากหน่วยความจำของ 8088 เพื่อให้หน่วยความจำที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้น ทำการส่งข้อมูลออกมาบนบัสข้อมูล

DRQ1-DRQ3 (DMA Request 1-3 : ขา B18 B6 และ B16)

ขาสัญญาณทั้ง 3 เป็นสัญญาณอินพุตแอกทีฟที่ลอจิก "1" ซึ่งอุปกรณ์ภายนอกสามารถใช้ในการขอ DMA จากระบบ โดยการป้อนระดับสัญญาณลอจิก "1" ให้กับขา DRQ ขาใดขาหนึ่ง

$\overline{\text{DACK0-DACK3}}$ (DMA Acknowledge 0-3 ขา B19 B17 และ B15)

สัญญาณทั้ง 4 นี้เป็นเอาต์พุตแอกทีฟที่ลอจิก "0" ซึ่ง 8237A-5 สร้างขึ้นเพื่อแสดงให้องค์กรภายนอกที่ขอ DMA นั้นได้รับการตอบสนองแล้ว และ 8537A-5 จะเข้าสู่ขบวนการ DMA เพื่อให้การส่งผ่านข้อมูลระหว่างอุปกรณ์ I/O ที่ขอ DMA กับหน่วยความจำเกิดขึ้นได้โดยตรง โดยสัญญาณ $\overline{\text{DACK}}$ นี้จะแอกทีฟในแชนแนลใดก็ขึ้นอยู่กับว่าขบวนการ DMA ที่เกิดขึ้นนั้นเป็นการตอบสนองต่อการขอ DMA ในแชนแนลใด

AEN(Address Enable : ขา A11)

สัญญาณนี้เป็นเอาต์พุตที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอกทีฟ(ลอจิก "1") นั้นเป็นบัสไซเคิลของขบวนการ DMA

สำหรับบนเมนบอร์ดของ IBM/PC นั้นจะใช้สัญญาณนี้ในการดิสเอเบิล 8255 และจะใช้ดิสเอเบิลพอร์ท I/O ต่าง ๆ ที่ไม่เกี่ยวข้องกับขบวนการ DMA ที่เกิดขึ้น ที่จำเป็นต้องทำเช่นนี้ก็เพราะในระหว่าง ขบวนการ DMA นั้น 8237A-5 จะส่งแอดเดรสของหน่วยความจำออกมาบนบัสแอดเดรส และจะทำให้สัญญาณ IOR หรือ $\overline{\text{IOW}}$ แอกทีฟด้วย ดังนั้นถ้าไม่ทำการดิสเอเบิลพอร์ท I/O ที่ไม่เกี่ยวข้องไว้ ก็จะทำให้พอร์ท I/O ที่มีค่าแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้นทำการอ่านหรือส่งข้อมูลออกมาบนบัสข้อมูลทำให้เกิดความผิดพลาดได้

T/C (Terminal Count : ขา B27)

สัญญาณนี้ถูกสร้างขึ้นจากการนำเอาสัญญาณเอาต์พุตที่ขา EOP ของ 8237A-5 มากลับลอจิก ทำให้สัญญาณ T/C นี้แอกทีฟที่ลอจิก "1"

บัสของแหล่งจ่ายไฟของระบบ

+5Vdc (ขา B3 และ B29)

ขาทั้งสองนี้ต่อกับแหล่งจ่ายไฟ DC +5V ของระบบ โดยจะมีค่าความเที่ยงตรงอยู่ในช่วง +4.75 ถึง +5.25 Vdc

+12Vdc (ขา B9)

ขานี้จะต่ออยู่กับแหล่งจ่ายไฟ DC +12V ของระบบ โดยจะมีค่าความเที่ยงตรงอยู่ในช่วง +11.4 ถึง +12.6 Vdc

-5Vdc (ขา B5)

ขานี้จะต่ออยู่กับแหล่งจ่ายไฟ DC -5V โดยจะมีค่าความเที่ยงตรงอยู่ในช่วง -5.5 ถึง -4.5 Vdc

-12Vdc (ขา B7)

ขานี้จะต่ออยู่กับแหล่งจ่ายไฟ DC -12V ของระบบ โดยจะมีค่าความเที่ยงตรงอยู่ในช่วง -13.2 ถึง -10.8Vdc

GND (ขา B1 B10 และ B31)

ขาทั้งสามจะต่อเข้ากราวด์ของระบบ

2.4.2 การจัดแอดเดรสสำหรับหน่วยความจำและ I/O

การอ้างแอดเดรสของพอร์ท

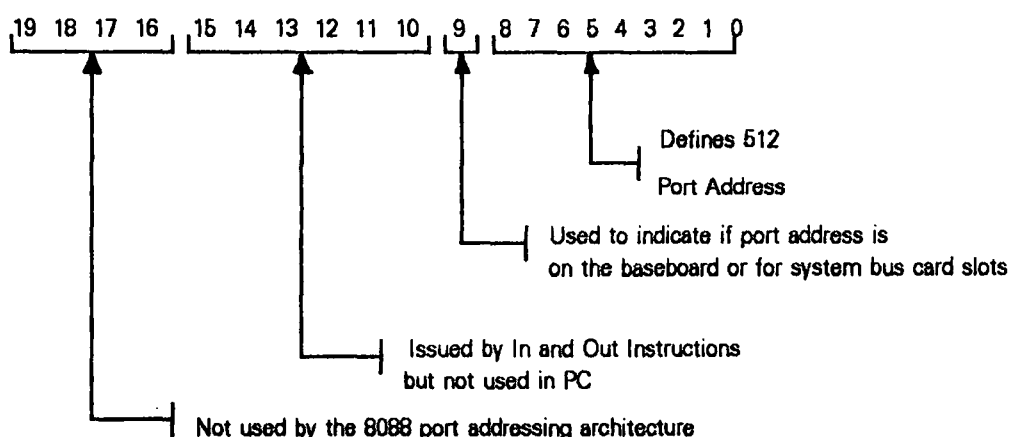
แอดเดรสของพอร์ท I/O ต่าง ๆ จะถูกสร้างขึ้นโดย 8088 ซึ่งแอดเดรสเหล่านี้เป็นแอดเดรสที่จัดไว้สำหรับพอร์ท I/O โดยเฉพาะ คือ แยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาด ส่วนการส่งข้อมูลให้กับพอร์ทเหล่านี้จะทำได้โดยการใช้คำสั่ง OUT ของ 8088 ส่งข้อมูลนั้นไปยังแอดเดรสของพอร์ทที่ต้องการ และการตรวจสอบหรือการอ่านข้อมูลจากพอร์ทก็ทำได้โดยการใช้คำสั่ง IN ของ 8088 อ่านข้อมูลจากแอดเดรสที่ต้องการเช่นกัน

ภายในตัว 8088 จะมีแอดเดรสสำหรับใช้กับพอร์ท I/O อยู่ 65,535 หรือ 64k แอดเดรส ซึ่งทำให้การอ้างแอดเดรสของพอร์ท I/O ที่ทำงานร่วมกับ 8088 ต้องใช้จำนวนเส้นแอดเดรส 16 เส้น คือ A0 - A15 แต่ใน IBM/PC นี้ถูกออกแบบมาให้ใช้เส้นแอดเดรสเฉพาะ 10 เส้นล่าง คือ A0 - A9 ดังนั้น

ในการอ้างแอดเดรสของพอร์ทของอุปกรณ์หรือชิพพอร์ทใด ๆ ที่ใช้ร่วมกับ IBM/PC จึงใช้จำนวนเส้นแอดเดรสเพียง 10 เส้น โดย A10 - A15 จะไม่ถูกนำไปใช้งานแต่ค่าแอดเดรสบนเส้นแอดเดรสเหล่านี้จะยังคงเปลี่ยนแปลงตามค่าแอดเดรสของพอร์ทที่กำหนดไว้ในคำสั่ง OUT หรือ IN อยู่ด้วย

เนื่องจากใน IBM/PC ได้ใช้งานเส้นแอดเดรสเพียง 10 เส้น ดังนั้นจึงสามารถอ้างแอดเดรสของพอร์ทได้สูงสุดเพียง 1024 พอร์ท นอกจากนี้ในกรณีที่เป็นการอ่านข้อมูลจากพอร์ทของ IBM/PC ข้อมูลในบิต A9 จะถูกจัดให้มีหน้าที่ในการแบ่งพอร์ททั้ง 1204 พอร์ทออกเป็น 2 ส่วน คือ ถ้าข้อมูลในบิต A9 เป็น "0" ก็จะทำให้การอ่านข้อมูลได้เฉพาะจากพอร์ทของอุปกรณ์หรือชิพพอร์ทต่าง ๆ ที่อยู่บนเมนบอร์ด แต่ถ้าข้อมูลในบิต A9 เป็น "1" ก็จะทำให้การอ่านข้อมูลได้เฉพาะจากพอร์ทที่อยู่บนการ์ดต่าง ๆ เท่านั้น

สำหรับรูปที่ 2-23 แสดงถึงการใช้งานแอดเดรสบิตต่าง ๆ ในการอ้างแอดเดรสทางพอร์ทใน IBM/PC



รูปที่ 2-23 แสดงถึงการใช้งานแอดเดรสบิตต่าง ๆ ในการอ้างแอดเดรสทางพอร์ทใน IBM/PC

การใช้งานแอดเดรสสำหรับพอร์ท I/O ใน IBM/PC

1.) การใช้งานของกลุ่มพอร์ทที่อยู่บนเมนบอร์ดของ IBM/PC ซึ่งมีแอดเดรสอยู่ในตำแหน่ง 0000H จนถึง 01FFH (บิต A9 เป็น "0") สำหรับแอดเดรสของพอร์ท I/O ในกลุ่มนี้จะถูกใช้งานในการอ้างแอดเดรสของพอร์ทและอุปกรณ์ที่เป็น I/O ต่าง ๆ บนเมนบอร์ดของ IBM/PC รูปที่ 2-24 จะแสดงถึงการใช้งานแอดเดรสต่าง ๆ ตั้งแต่ 0000H ถึง 01FFH

จากรูปที่ 2-24 จะเห็นว่าแอดเดรส 00C0H จนถึง 01FFH นั้นไม่ได้ถูกใช้งานบนเมนบอร์ดของ IBM/PC ดังนั้นจึงสามารถที่จะใช้งานแอดเดรสต่าง ๆ เหล่านี้ได้ อย่างไรก็ตามแอดเดรสเหล่านี้ยังถูกตีโค้ดให้เป็นแอดเดรสที่ใช้งานในการอ่านข้อมูลจากพอร์ท I/O บนเมนบอร์ดเท่านั้น ดังนั้นการกำรใช้ค่าแอดเดรส 00C0H ถึง 01FFH กับพอร์ท I/O บนการ์ดหรือวงจรรินเทอร์เฟซที่เราสร้างขึ้นต้องเป็นพอร์ทเอาต์พุตเพียงชนิดเดียว

2.) การใช้งานของกลุ่มพอร์ทที่อยู่บนการ์ดที่ใช้เสียบบนสล็อต ซึ่งมีแอดเดรสอยู่ในตำแหน่ง 0200H จนถึง 03FFH (A9 เป็น "1")

0000 H	512	Used on base system logic board
01FF H		
0200 H	512	Avariable in system bus card slots
03FF H		
0400 H	64512	Not used in PC Design
FFFF H		

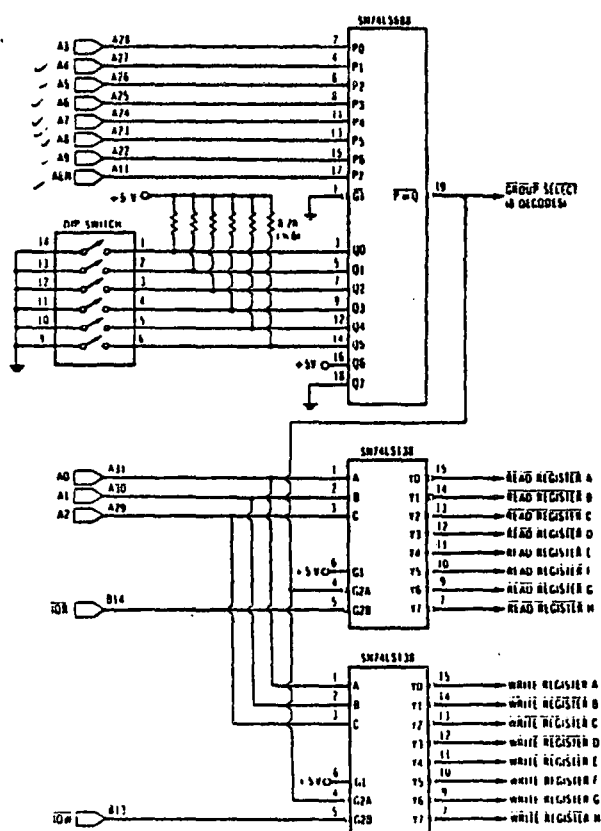
รูปที่ 2-24 การใช้งานแอดเดรสต่างๆ สำหรับพอร์ท I/O ของ IBM / PC

การใช้งานแอดเดรสในกลุ่มนี้จะเปลี่ยนแปลงได้ขึ้นอยู่กับการใช้การ์ดต่าง ๆ ร่วมกับ IBM/PC โดยการ์ดที่ถูกออกแบบนี้อาจจะใช้ค่าแอดเดรสต่างๆ ที่เหลืออยู่ก็ได้ ดังนั้นก่อนที่จะออกแบบวงจรอินเทอร์เฟซที่จำเป็นต้องใช้ค่าแอดเดรสสำหรับพอร์ท I/O จึงควรจะตรวจสอบดูว่าการ์ดต่าง ๆ ที่ใช้อยู่ในระบบของ IBM/PC ที่เราใช้งานอยู่นั้นมีการ์ดอะไรบ้าง และใช้งานอยู่ในแอดเดรสใดสำหรับการใช้งานแอดเดรสของพอร์ท I/O ในกลุ่มนี้จะแสดงได้ดังรูปที่ 2-25

	HEX ADDRESS	USES
1	0200H	NOT USED
1	0201H	GAME CONTROL ADAPTER
118	0202H-0277H	NOT USED
8	0278H-027FH	SECOND PRINTER PORT ADAPTER
120	0280H-02F7H	NOT USED
8	02F8H-02FFH	SECOND SERIAL PORT ADAPTER CARD
120	0300H-0377H	NOT USED
8	0378H-037FH	PRINTER PORT ADAPTER CARD
48	0380H-03AFH	NOT USED
16	03B0H-03BFH	MONOCHROME AND PRINTER ADAPTER
16	03C0H-03CFH	NOT USED
16	03D0H-03DFH	COLOR GRAPHICS ADAPTER
16	03E0H-03EFH	NOT USED
8	03F0H-03F7H	5 1/6 INCH DISKETTIE DRIVE ADAPTER
8	03F8H-03FFH	SERIAL PORT ADAPTER CARD

รูปที่ 2-25 การใช้งานแอดเดรสสำหรับพอร์ท I/O บนการ์ดต่างๆ

- เทคนิคการดีโค้ดแอดเดรสสำหรับพอร์ท I/O



รูปที่ 2-26 ตัวอย่างวงจรดีโค้ดโดยใช้สวิทช์เลือก

จากรูปเป็นวงจรที่ทำการดีโค้ดกลุ่มแอดเดรสขนาด 8 แอดเดรส ซึ่งการเลือกกลุ่มแอดเดรสที่จะทำการดีโค้ดนี้จะได้ทำได้โดยการเซต DIP Switch ที่ขา Q0-Q5 ของ 74LS688

สำหรับหน้าที่ของ 74LS688 นี้จะทำการเปลี่ยนเทียบค่าของอินพุต 2 ชุดที่ถูกส่งเข้ามาทางขา P0-P7 และขา Q0-Q7 ถ้าอินพุตทั้ง 2 ชุดนี้เท่ากันแล้ว เอาท์พุทที่ขา P=Q จะให้อาท์พุทเป็นลอจิก "0" จากในวงจรขา P0-P5 ของ 74LS688 ต่อกับแอดเดรสบิต A4-A9 ในขณะที่ขา Q0-Q5 ต่อกับทางด้านทานที่ทำหน้าที่รักษาระดับแรงดันให้เป็นลอจิก "1" ไว้ในกรณีที่ไม่มีอินพุตใด ๆ เข้ามา และขา Q0-Q5 นี้จะต่อกับปลายข้างหนึ่งของดิฟสวิทช์ ส่วนปลายอีกข้างหนึ่งของดิฟสวิทช์นั้นจะต่อลงกราวด์ไว้ ดังนั้นถ้าเราทำการ "ON" ดิฟสวิทช์ ที่ต่อกับขาใดขานั้นก็จะได้รับลอจิก "0" ในขณะที่ถ้าดิฟสวิทช์ที่ต่อกับขาใดถูก "OFF" ขานั้นก็จะได้รับลอจิก "1" และเนื่องจากอินพุตที่ขา P0-P5(แอดเดรส A3-A9) ต้องเท่ากับอินพุตที่ขา Q0-Q5 ดังนั้น ถ้าเราเปลี่ยนแปลงการเซตดิฟสวิทช์เหล่านี้ก็จะทำให้แอดเดรส A3-A5 ซึ่งต่อกับขา P0-P5ต้องเปลี่ยนแปลงตามไปด้วยจึงทำให้เอาท์พุทของ 74LS688 แอดคทีฟได้ สำหรับขา Q6 นั้นจะต่อกับลอจิก "1" และขา P5 ต่อกับแอดเดรส A9 ในกรณีเช่นนี้จึงเท่า

เท่ากับเป็นการบังคับให้แอดเดรสที่จะทำการดีโค๊ดได้นั้นจะต้องมีแอดเดรส A9 เป็น "1" เท่านั้น

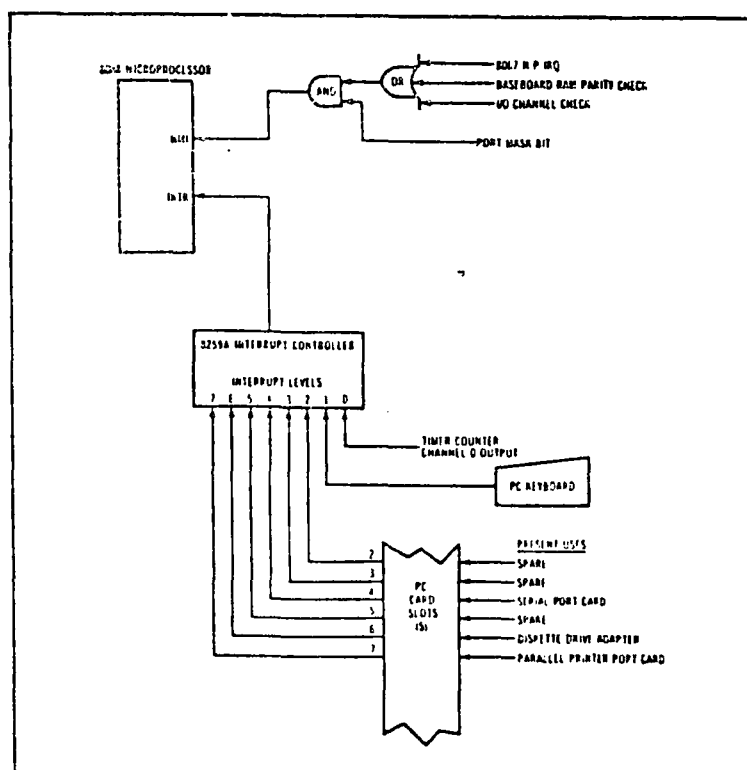
2.4.3 การจัดการอินเทอร์รัพท์ของ IBM/PC

การจัดการอินเทอร์รัพท์ใน IBM/PC จะแบ่งออกเป็น 9 ระดับ โดยสามารถแบ่งออกได้เป็น 2 กลุ่ม คือ การอินเทอร์รัพท์แบบนอนมาสค์เอเบิล (Non-Maskable Interrupt : NMI) และการอินเทอร์รัพท์แบบมาสค์เอเบิล (Maskable Interrupt : INT)

สำหรับการอินเทอร์รัพท์แบบ นอนมาสค์เอเบิลนั้น จะเกิดได้ใน 3 กรณี คือ

1. เมื่อเกิดความผิดพลาดในการเช็คพาริตีของข้อมูลในหน่วยความจำ
2. สัญญาณจากขา 32 ของซอกเก็ตที่ใช้สำหรับ 8087
3. สัญญาณ I/O CHECK จากสล็อต

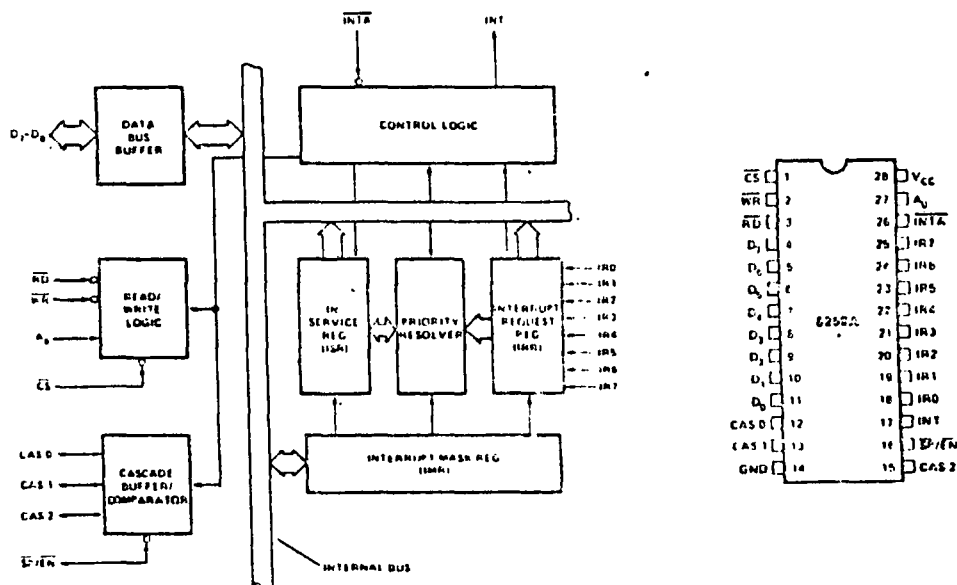
สำหรับการอินเทอร์รัพท์แบบมาสค์เอเบิล นั้น IBM/PC จะใช้ชิพพอร์ทเบอร์ 8259A Interrupt Controller เข้าช่วยในการจัดการอินเทอร์รัพท์แบบมาสค์เอเบิลของระบบ ซึ่งการใช้ 8259A เข้าช่วยนี้ทำให้ IBM/PC สามารถจัดแบ่งการอินเทอร์รัพท์ในแบบมาสค์เอเบิลออกได้ถึง 8 ระดับโดยผ่านแชนเนลการอินเทอร์รัพท์ทั้ง 8 แชนเนลของ 8259A คือ IR0-IR7



รูปที่ 2-27 แสดงบล็อกไดอะแกรมของวงจรในส่วนของการอินเทอร์รัพท์ใน IBM/PC

8259A Interrupt Controller

8259A เป็นชิพซีพพอร์ทที่ช่วยจัดการเกี่ยวกับการอินเทอร์รัพท์ของระบบ ซึ่งการใช้ 8259A นี้จะทำให้สามารถเพิ่มจำนวนระดับ (แชนแนล) ของการขออินเทอร์รัพท์แบบมาสคเเบิ้ลได้มากถึง 8 ระดับ โดยแต่ละระดับจะมีลำดับความสำคัญที่แตกต่างกันรูปที่ 2-28 จะแสดงถึงบล็อกไดอะแกรม การจัดเรียงขาและหน้าที่ของขาต่าง ๆ ของ 8259A



รูปที่ 2-28 บล็อกไดอะแกรมและการจัดเรียงขาของ 8259A

Vcc (Supply : ขา 28) เป็นเส้นทางเดินของแหล่งจ่ายไฟ +5 Volt

GND (Ground : ขา 14) กราวด์

$\overline{CS}$ (Chip Select : ขา 1)

เมื่อระดับสัญญาณที่ขานี้เป็นศูนย์ จะเป็นการอินนาเบิ้ล 8259A ให้สามารถติดต่อส่งผ่านข้อมูลกับ CPU ได้

$\overline{WR}$ (Write Signal : ขา 2)

เมื่อสัญญาณที่ขาได้รับลอจิก "0" และสัญญาณ CS นั้นแอกทีฟด้วย ก็จะเป็นการทำให้ 8259A รับข้อมูล (คำสั่ง) ที่ส่งมาจาก CPU

$\overline{RD}$ (Read Signal : ขา 3)

เมื่อสัญญาณที่ขาได้รับลอจิก "0" และสัญญาณ CS นั้นแอกทีฟด้วย ก็จะเป็นการทำให้ 8259A ส่งสถานะการทำงานออกมาบนบัสข้อมูลเพื่อส่งให้กับ CPU

D7-D0 (Bidirectional Data Bus : ขา 4-11)

ใช้ในการส่งผ่านข้อมูลและสถานะต่าง ๆ ระหว่าง CPU กับ 8259A

CAS0-CAS2 (Cascade Lines : ขา 12,13,15)

ใช้ในระบบที่มีการใช้ 8259A มากกว่า 1 ตัว สัญญาณทั้งสามนี้จะเป็นเอาต์พุตสำหรับ 8259A ที่เป็นมาสเตอร์ และจะเป็นอินพุตสำหรับ 8259A ที่เป็นสเลฟ

$\overline{SP}$ / $\overline{EN}$ (Slave Program / Enable Buffer : ขา 16)

กรณีที่ 8259A ทำงานในโหมดบัฟเฟอร์ ขานี้จะทำหน้าที่เป็นเอาต์พุตที่ใช้อินนาเบิลบัฟเฟอร์ต่าง ๆ และในกรณีที่ 8259A ไม่ได้ทำงานในโหมดบัฟเฟอร์ ขานี้จะทำหน้าที่เป็นอินพุตที่ใช้กำหนดว่า 8259A ตัวใดเป็นมาสเตอร์ตัวใดเป็นสเลฟ(มาสเตอร์ป้อนลอจิก "1" สเลฟป้อนลอจิก "0")

INT (Interrupt : ขา 17)

ขาสัญญาณนี้เป็นเอาต์พุตที่ใช้สำหรับส่งสัญญาณอินเทอร์รัพท์ให้กับ CPU โดยที่สัญญาณขานี้จะแอกทีฟ (ลอจิก 1) เมื่ออุปกรณ์ภายนอกส่งสัญญาณขออินเทอร์รัพท์มาให้กับขาอินพุต IRO-IR7 ของ 8259A

IRO-IR7 (Interrupt Request : ขา 18-25)

ขาสัญญาณอินพุตทั้ง 8 นี้ เป็นขาที่ใช้ในการรับสัญญาณขออินเทอร์รัพท์จากอุปกรณ์ภายนอก สำหรับการแอกทีฟของสัญญาณที่ป้อนให้กับอินพุตทั้ง 8 นี้ จะแบ่งออกได้เป็น 2 ลักษณะโดยขึ้นอยู่กับการโปรแกรม 8259A คือ

1. ถ้า 8259A ถูกโปรแกรมให้ทำงานในแบบ Edge Trigger สัญญาณจะแอกทีฟเมื่อระดับลอจิกของสัญญาณอินพุตถูกเปลี่ยนจาก 0 เป็น 1 (ขอบขาขึ้น) และสัญญาณนี้จะต้องเป็น 1 อยู่จนกระทั่งการขออินเทอร์รัพท์นั้นได้รับการตอบสนอง

2. ถ้า 8259A ถูกโปรแกรมให้ทำงานในแบบ Level Trigger สัญญาณจะแอกทีฟเมื่อสัญญาณอินพุตมีลอจิกเป็น 1

$\overline{INTA}$ (Interrupt Acknowledge : ขา 26)

สัญญาณที่ป้อนให้กับขานี้ จะเป็นสัญญาณที่ใช้รีนาเบิ้ลให้ 8259A ทำการส่งข้อมูลที่เป็นเวกเตอร์การอินเทอร์รัพท์ออกมาบนบัสข้อมูล สำหรับสัญญาณที่นำมาป้อนให้กับขา $\overline{INTA}$ จะเป็นสัญญาณตอบรับการขออินเทอร์รัพท์ที่ CPU สร้างขึ้น

A0 (A0 Address Line : ขา 27)

เป็นสัญญาณอินพุตซึ่งถูกใช้งานร่วมกับสัญญาณที่ขา CS , WR , RD เพื่อกำหนด Command Words หรือ Control Words ที่ CPU ต้องการจะส่งข้อมูลให้

ขั้นตอนของการอินเทอร์รัพท์

1. วงจรอินเทอร์เฟสทำการขออินเทอร์รัพท์ผ่านทาง 8259A โดยทำให้สัญญาณที่ขา IRO-IR7 ของ 8259A ขาใดขาหนึ่งแอกทีฟ (บนสล็อตของ IBM/PC จะเรียกชื่อขาเหล่านี้เป็น IRQ0-IRQ7 ตามลำดับ) ส่วนอุปกรณ์ใดจะทำการอินเทอร์รัพท์ผ่านทางขาใดนั้น ก็ขึ้นอยู่กับการออกแบบระบบว่า การอินเทอร์รัพท์ทางขาใดจะใช้สำหรับอุปกรณ์ใด

2. หลังจากที่ 8259A ได้รับสัญญาณขออินเทอร์รัพท์จากวงจรอินเทอร์เฟสแล้วจะทำการเซตบิต IRR ของ 8259A ที่สัมพันธ์กับสัญญาณที่ขออินเทอร์รัพท์ จากนั้น ในกรณีที่มีสัญญาณขออินเทอร์รัพท์เข้ามาพร้อมกันมากกว่า 1 สัญญาณ 8259A ก็จะเปรียบเทียบลำดับในแกนแนลที่มีลำดับความสำคัญสูงสุดก่อน จากนั้นจึงจะตอบสนองต่อการขออินเทอร์รัพท์ที่มีลำดับความสำคัญรองลงมา

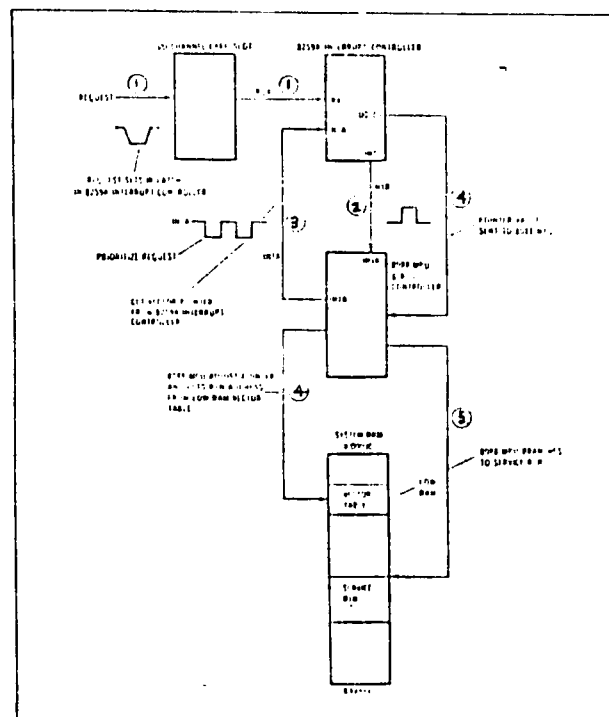
3. หลังจากที่ CPU ได้รับสัญญาณ INT แล้ว ก็จะตอบสนองโดยการส่งพัลส์ของสัญญาณ $\overline{INTA}$ (Interrupt Acknowledge , แอกทีฟที่ลอจิก 0) จำนวน 2 ลูก ให้กับ 8259A โดยพัลส์ลูกแรกจะเป็นการทำให้บิต ISR บิตใดบิตหนึ่งของ 8259A ถูกเซต สำหรับบิต ISR (In-Service Register : เป็นรีจิสเตอร์ที่ใช้แสดงการทำงานของ 8259A ว่าแกนแนลใดที่กำลังได้รับการตอบสนองอยู่) นี้จะมี

จำนวน 8 บิต คือ ISRO-ISR7 เท่ากับจำนวนแชนแนลและเท่ากับจำนวนบิต IRR (Interrupt Request Register : เป็นรีจิสเตอร์ที่ใช้แสดงว่า แชนแนลใดที่ทำการขออินเทอร์รัพท์เข้ามาแล้วยังไม่ได้รับการตอบสนอง) โดยบิตใดของ ISR จะถูกเซตนั้น ขึ้นอยู่กับแชนแนลของการอินเทอร์รัพท์ที่ 8259A จะทำการตอบสนอง (การตอบสนอง คือ การส่งสัญญาณ INT ให้กับ CPU รวมทั้งการส่งเวคเตอร์การอินเทอร์รัพท์สำหรับการขออินเทอร์รัพท์ในแชนแนลนั้นให้กับ CPU ด้วย) และในเวลานี้ 8259A ก็จะมีเซตบิต IRR ที่ตรงกับบิต ISR ที่ถูกเซตด้วย

4. CPU จะส่งพัลส์ลูกที่ 2 ของสัญญาณ INTA ให้กับ 8259A ในเวลานี้ 8259A จะทำการส่งเวคเตอร์ของการอินเทอร์รัพท์ออกมาบนบัสข้อมูล เพื่อให้ CPU นำไปเปิดตารางเวคเตอร์เพื่อหาตำแหน่งแอดเดรสเริ่มต้นของโปรแกรมที่ตอบสนองต่อการอินเทอร์รัพท์

เวคเตอร์ของการอินเทอร์รัพท์จะมีค่าแตกต่างกันไปตามแชนแนลที่ขออินเทอร์รัพท์ เพื่อให้ตำแหน่งของโปรแกรมที่ตอบสนองต่อการอินเทอร์รัพท์ของแต่ละแชนแนลแตกต่างกัน

5. หลังจากที่ CPU คำนวณค่าแอดเดรสของโปรแกรมที่ตอบสนองต่อการอินเทอร์รัพท์เรียบร้อยแล้ว CPU ก็จะเก็บค่ารีจิสเตอร์ CS (CODE Segment Register) , IP (Instruction Pointer or Program Counter) และ แฟล็ก (Flag) ลงบนหน่วยความจำที่เป็นสแต็ก (Stack) จากนั้นจึงทำการโหลดค่าแอดเดรสที่คำนวณได้นั้นไปเก็บไว้ในรีจิสเตอร์ CS , IP ซึ่งจะเป็นผลให้ 8088 กระโดดไปทำงานในโปรแกรมที่ตอบสนองต่อการอินเทอร์รัพท์ที่เกิดขึ้น



รูปที่ 2-29 ขั้นตอนการที่เกิดขึ้นในขบวนการอินเทอร์รัพท์

โปรแกรมตอบสนองการขออินเทอร์รัพท์

ในโปรแกรมตอบสนองการขออินเทอร์รัพท์ นอกจากจะประกอบด้วยส่วนของโปรแกรมที่ตอบสนองต่ออุปกรณ์ที่ขออินเทอร์รัพท์แล้วยังจำเป็นที่จะต้องเพิ่มโปรแกรมอีกส่วนเข้าไปเพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้เมื่อ 8088 กลับเข้าสู่การทำงานในโปรแกรมเดิมที่ทำงานอยู่ก่อนจะถูกอินเทอร์รัพท์ ส่วนของโปรแกรมที่เพิ่มเข้าไปนี้จะช่วยในการแก้ปัญหาที่อาจเกิดขึ้นได้ คือ ในกรณีที่ภายในโปรแกรมตอบสนองการอินเทอร์รัพท์ใช้งานรีจิสเตอร์เดียวกับที่โปรแกรมเดิมก่อนที่จะถูกอินเทอร์รัพท์ใช้อยู่ จะทำให้ข้อมูลภายในรีจิสเตอร์นั้นเปลี่ยนแปลงไป ซึ่งเป็นผลให้เกิดความผิดพลาดขึ้นเมื่อ 8088 กลับเข้าสู่การทำงานในโปรแกรมเดิม

เพื่อจะแก้ปัญหานี้ จำเป็นต้องเพิ่มส่วนของโปรแกรมที่สั่งให้ 8088 ทำการเก็บค่าของ รีจิสเตอร์ที่ถูกใช้งานในโปรแกรมตอบสนองการอินเทอร์รัพท์ไว้ก่อนที่จะเริ่มการทำงานในโปรแกรมที่ตอบสนองต่ออุปกรณ์ที่ขออินเทอร์รัพท์ และก่อนที่จะเริ่มการทำงานในส่วนของโปรแกรมเดิมก็จะต้องทำการดึงเอาข้อมูลที่เก็บไว้คืนให้กับรีจิสเตอร์ต่าง ๆ เหล่านั้นด้วย โดยทั่วไปแล้วจะใช้คำสั่ง PUSH และ POP

การจัดเตรียมอินเทอร์รัพท์ของระบบ

1. จัดเตรียมเนื้อที่ในหน่วยความจำ (เฉพาะส่วนที่เป็น RAM) ที่ต้องการใช้เป็นสแต็กโดยกำหนดค่าแอดเดรสของหน่วยความจำนั้นให้กับรีจิสเตอร์ SS (Stack Segment) และ SP (Stack Pointer)
2. จัดเตรียมเนื้อที่ในหน่วยความจำสำหรับเก็บตารางเวกเตอร์ เนื้อที่สำหรับหน่วยความจำในส่วนนี้จะใช้เนื้อที่ 1K ไบต์ล่างของหน่วยความจำส่วนที่เป็น RAM คือ จากแอดเดรส 00000H จนถึง 003FFH
3. ทำการโปรแกรม 8259A ตามลำดับขั้นตอนดังนี้
 - 3.1 ทำการส่งข้อมูลให้กับ Initialization Command Words (ICWs) ของ 8259A ซึ่งข้อมูลที่ส่งให้นี้ จะกำหนดการทำงานโดยปรกติของ 8259A
 - 3.2 ทำการส่งข้อมูลให้กับ Operation Command Words (OCWs) ของ 8259A ซึ่งข้อมูลที่ส่งให้นี้ จะกำหนดโหมดการอินเทอร์รัพท์ของ 8259A
4. ทำการอินทาบิลการอินเทอร์รัพท์ของ 8088 โดยการเซตแฟล็ก I ของรีจิสเตอร์แฟล็ก (คำสั่ง STI)

ตารางเวคเตอร์บน IBM/PC

8088 ใช้หน่วยความจำ 1024 ไบต์ล่าง คือจาก 00000H จนถึง 003FFH สำหรับการอินเทอร์รัพท์ของระบบ ซึ่งจะมีการอินเทอร์รัพท์ได้ทั้งสิ้น 256 Type โดยหน่วยความจำส่วนนี้จะถูกใช้ในการเก็บแอดเดรสของโปรแกรมย่อยที่ตอบสนองต่อการอินเทอร์รัพท์ทั้ง 256 Type นี้และ เนื่องจากการอ้างแอดเดรสในหน่วยความจำต้องกำหนดค่าให้กับรีจิสเตอร์ 2 ตัวคือ รีจิสเตอร์ IP และ CS ซึ่งเป็นรีจิสเตอร์ขนาด 16 บิตทั้งคู่ ดังนั้นหน่วยความจำที่ใช้เก็บแอดเดรสสำหรับการอินเทอร์รัพท์แต่ละ Type จะมีจำนวน 4 ไบต์ คือ 2 ไบต์ใช้สำหรับเก็บแอดเดรสที่จะป้อนให้กับ CS (คือ แอดเดรสเซก - เมนธ์) และอีก 2 ไบต์ใช้สำหรับเก็บค่าแอดเดรสที่จะป้อนให้กับ IP (คือ แอดเดรสออฟเซต) ซึ่งเมื่อนำแอดเดรสทั้งสองมาประกอบกันแล้วก็จะได้แอดเดรสของโปรแกรมที่ตอบสนองต่อการอินเทอร์รัพท์ที่เกิดขึ้น ซึ่งเมื่อรวมทั้งหมด 256 Type แล้ว ก็จะเป็นเนื้อที่ในหน่วยความจำเท่ากับ $256 * 4$ หรือ 1024 ไบต์พอดี

ในกรณีที่มีการขออินเทอร์รัพท์เกิดขึ้น 8088 จะใช้ตารางนี้เพื่อหาตำแหน่งแอดเดรสของโปรแกรมตอบสนองการอินเทอร์รัพท์ ซึ่งการอินเทอร์รัพท์ที่กล่าวถึงนี้อาจเป็นการอินเทอร์รัพท์ทางซอฟต์แวร์ หรือ ฮาร์ดแวร์ภายนอกก็ได้ รูปที่ 2-30 จะแสดงการใช้งานหน่วยความจำ 1024 ไบต์ล่างของ IBM/PC สำหรับการขออินเทอร์รัพท์ทั้ง 256 Type

X 003FF		TYPE 255	
			SOME USED BY BIOS OS AND BASIC SEE IBM MANUAL FOR PRESENT USAGE
	IRQ7	TYPE 16	
	IRQ6	TYPE 15	
	IRQ5	TYPE 14	
	IRQ4	TYPE 13	
	IRQ3	TYPE 12	8259A CONTROLLER INTERRUPTS
	IRQ2	TYPE 11	
	IRQ2	TYPE 10	
	IRQ1	TYPE 9	
	IRQ0	TYPE 8	
	RESERVED	TYPE 7	
	RESERVED	TYPE 6	USED BY BIOS
	PRINT SCREEN	TYPE 5	
	INT0	TYPE 4	INT0 INSTRUCTION
	INT	TYPE 3	INT INSTRUCTION
	NMI	TYPE 2	8088 MPU NMI INPUT DEDICATEED INTERRUPT POINTERS
	SINGLE STEP	TYPE 1	8088 MPU HARDWARE
X 00000	DIVIDE ERROR	TYPE 0	8088 MPU HARDWARE

NOTE :

ANY INTERRUPTS LEVEL(TYPE) CAN BE GENERERATED BY
USING THE 8088 MICROPROCESSOR INSTRUCTION INT(TYPE)

CS (BASE ADDRESS)
IP (OFFSET)

รูปที่ 2-30 การใช้งานหน่วยความจำ 1024 ไบต์ล่างของ IBM/PC

อุปกรณ์ภายนอกที่ขออินเทอร์รัพท์นั้นจะส่งข้อมูลที่ใช้เป็นพอยเตอร์ขนาด 8 บิต ออกมาบน บัสข้อมูล เพื่อส่งให้กับ 8088 หลังจากได้รับพัลส์ INTA ลูกที่ 2 ซึ่งพอยเตอร์ขนาด 8 บิต ที่ถูกส่งให้ กับ 8088 จะประกอบไปด้วย

บิต	D7	D6	D5	D4	D3	D2	D1	D0
การใช้งาน	A9	A8	A7	A6	A5	L3	L2	L1

ค่าพอยเตอร์นี้ จะถูกส่งออกมาโดย 8259A โดยที่ L1-L3 เป็นค่าที่ 8259A ทำการกำหนดให้ โดยอัตโนมัติ ค่านี้จะตรงกับแชนแนลที่ขออินเทอร์รัพท์อยู่ (เลขฐานสอง) สำหรับ A5-A9 เป็นค่าที่ ถูกโปรแกรมไว้ใน ICW2 ของ 8259A

จะเห็นได้ว่าค่าที่โปรแกรมให้กับ 8259A จะทำให้การอินเทอร์รัพท์ถูกแบ่งออกเป็น 32 ช่วง โดยแต่ละช่วงจะมีการอินเทอร์รัพท์อีก 8 Type (บิต L1-L3) ซึ่งจะครบ 256 Type พอดี

8088 จะใช้ค่าพอยเตอร์นี้ในการสร้างเป็นแอดเดรสขนาด 10 บิต เพื่อที่ใช้สำหรับอ้างแอด - เดรสของ RAM ในส่วนที่เป็นตารางเวคเตอร์ ซึ่ง 8088 จะสร้างแอดเดรสขนาด 10 บิตนี้ โดยการเพิ่ม บิตที่มีข้อมูลเป็น '0' อีก 2 บิตเข้าไปที่ท้ายของพอยเตอร์ที่ได้รับ

A9	A8	A7	A6	A5	L3	L2	L1	'0'	'0'
----	----	----	----	----	----	----	----	-----	-----

เหตุที่ 8088 เพิ่มบิตที่เป็น '0' เข้าไปเพื่อให้แอดเดรสของการอินเทอร์รัพท์แต่ละแบบนั้น ห่างกัน 4 ไบท์

ข้อมูลในบิต A5-A9 นั้น สามารถที่จะโปรแกรมได้โดยผ่านทาง ICW2 ดังนั้น เพื่อให้การ อินเทอร์รัพท์ภายใน IBM/PC เกิดขึ้นได้ตามที่กำหนดไว้ โปรแกรมส่วนที่เป็น BIOS จึงได้โปรแกรม 8259A ให้ข้อมูลในบิต A5-A9 มีค่าเป็น 00001 (เลขฐานสอง) ซึ่งจะทำให้ค่าแอดเดรสที่เป็นไปได้ใน กรณีนี้ คือ

แบบแผนที่ ขออินเทอร์รัพท์	แอดเดรสบิต A9-A5	แอดเดรสบิต A4-A2			บิต A1-A0	แอดเดรส (ฐานสิบหก)
		A4(L3)	A3(L2)	A2(L1)		
0	00001	0	0	0	00	0020
1	00001	0	0	1	00	0024
2	00001	0	1	0	00	0028
3	00001	0	1	1	00	002C
4	00001	1	0	0	00	0030
5	00001	1	0	1	00	0034
6	00001	1	1	0	00	0038
7	00001	1	1	1	00	003C

ตารางที่ 2-3

จะเห็นได้ว่าแอดเดรสของตารางเวคเตอร์สำหรับ IBM/PC ในกรณีนี้จะอยู่ระหว่าง 00020H จนถึง 0003FH ในรูปที่ 2-31 จะแสดงการใช้ตารางเวคเตอร์ในแอดเดรสดังกล่าวของ IBM/PC (อินเทอร์รัพท์ Type 8-15) ซึ่งถูกกำหนดโดยโปรแกรมในส่วนของ BIOS

HEX ADDRESS

X 0003C	IRQ7	TYPE 15 PARALLEL PRINTER PORT CARD
X 00038	IRQ6	TYPE 14 DISKETTE ADAPTER CARD
X 00034	IRQ5	TYPE 13 NOT USED
X 00030	IRQ4	TYPE 12 SERIAL PORT CARD
X 0002C	IRQ3	TYPE 11 NOT USED
X 00028	IRQ2	TYPE 10 NOT USED
X 00024	IRQ1	TYPE 9 KEYBOARD
X 00020	IRQ0	TYPE 8 TIMER COUNTER CHANNEL 0

รูปที่ 2-31 การใช้เวคเตอร์ของ 8259A ใน IBM/PC

อย่างไรก็ตามเราสามารถเปลี่ยนแปลงตำแหน่งตารางเวคเตอร์ของ IBM/PC ให้อยู่ในตำแหน่งอื่นได้ โดยการเปลี่ยนข้อมูลของบิต A9-A5 ใน ICW2 แต่สิ่งที่จะต้องคำนึงถึง คือ ตารางเวคเตอร์ที่ BIOS ใช้อยู่นั้น ยังคงอยู่ในตำแหน่งเดิม คือ แอดเดรส 00020H-0003FH

การโปรแกรม 8259A

ภายใน 8259A จะมีรีจิสเตอร์ที่ใช้ในการเก็บคำสั่งต่าง ๆ ซึ่งเรียกว่า Command Word อยู่ 2 ชนิด โดย Command Word นี้จะใช้ในการกำหนดลักษณะการทำงานต่าง ๆ ของ 8259A สำหรับ Command Word ทั้ง 2 ชนิดนี้ จะประกอบไปด้วย

1. Initialization Command Words (ICWs) : ก่อนที่ 8259A จะทำงานได้เราจะต้องทำการโปรแกรม ICW นี้ก่อน สำหรับ ICW นี้จะมีอยู่ด้วยกัน 4 ตัว คือ ICW1 , ICW2 , ICW3 , ICW4 โดยในการโปรแกรมนั้นอาจจะโปรแกรม ICW เพียง 2 , 3 หรือ 4 ตัวก็ได้

2. Operation Command Words (OCWs) : มีอยู่ด้วยกัน 3 ตัว คือ OCW1 , OCW2 , OCW3 ซึ่งจะใช้ในการกำหนดโหมดการทำงานในขบวนการอินเทอร์รัพท์ของ 8259A สำหรับโหมดการทำงานนี้จะประกอบไปด้วย

2.1) โหมด Fully Nested

ในกรณีที่เราไม่ได้ทำการโปรแกรมให้ 8259A ทำงานในโหมดใดโหมดหนึ่งโดยเฉพาะแล้ว 8259A ก็ จะเข้าสู่การทำงานในโหมด Fully Nested เองโดยอัตโนมัติ

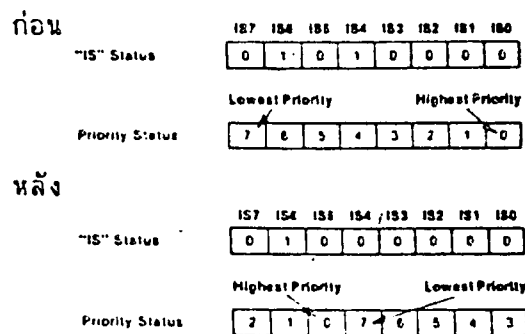
ภายในโหมดนี้ 8259A จะกำหนดให้การขออินเทอร์รัพท์ผ่านทางแชนแนล 0 มีลำดับความสำคัญสูงสุด ในขณะที่แชนแนล 7 มีลำดับความสำคัญต่ำสุด หลังจากที่ 8259A ทำการอินเทอร์รัพท์ CPU และได้รับสัญญาณ $\overline{INTA}$ ที่ CPU ส่งกลับมาแล้ว 8259A ก็ จะทำการส่งเวคเตอร์การอินเทอร์รัพท์ของแชนแนลที่ 8259A ทำการตอบสนองอยู่ออกมาบนบัสข้อมูล นอกจากนี้ข้อมูลของ รีจิสเตอร์ IS ในบิตที่ตรงกับแชนแนลที่ 8259A ทำการตอบสนองอยู่ก็ จะถูกเซตด้วย ซึ่งข้อมูลภายในบิตนี้จะยังคงถูกเซตจนกว่า CPU จะส่งคำสั่ง EOI (End Of Interrupt) มาให้กับ 8259A โดยที่คำสั่ง EOI นี้จะต้องถูกส่งให้กับ 8259A ก่อนที่ CPU จะเสร็จจากการทำงานในโปรแกรมตอบสนองการขออินเทอร์รัพท์ เนื่องจากการที่ข้อมูลในบิตใดบิตหนึ่งถูกเซตอยู่นั้น จะเป็นผลให้ 8259A ไม่ยอมรับการขออินเทอร์รัพท์ในแชนแนลที่มีลำดับความสำคัญเท่ากับหรือต่ำกว่าแชนแนลที่ตรงกับบิต ของ รีจิสเตอร์ IS ที่ถูกเซต

2.2) โหมด Rotating Priority

การทำงานในโหมดนี้จะช่วยให้สามารถเปลี่ยนแปลงลำดับความสำคัญของ แชนแนลการอินเทอร์รัพท์ทั้ง 8 แชนแนลของ 8259A ได้ สำหรับโหมดนี้ จะแบ่งออกได้เป็นอีก 2 โหมด คือ

a.) โหมด Automatic Rotation

ภายในโหมดนี้ 8259A จะทำการเปลี่ยนแปลงลำดับความสำคัญของแชนแนลการอินเทอร์รัพท์ทั้ง 8 โดยอัตโนมัติ คือ เมื่อ 8259A ทำการขออินเทอร์รัพท์ CPU ให้กับแชนแนลใดแล้ว แชนแนลนั้นก็จะถูกปรับให้มีลำดับความสำคัญต่ำสุด โดยที่ลำดับความสำคัญของแชนแนลอื่นๆ ก็จะถูกเปลี่ยนแปลงไปในลักษณะที่สัมพันธ์กัน ดังแสดงในตัวอย่างข้างล่าง



รูปที่ 2-32 ตัวอย่างการเปลี่ยนแปลงลำดับความสำคัญของแชนแนลต่าง ๆ

โดยทั่วไปโหมดนี้จะถูกใช้งานในกรณีที่อุปกรณ์ภายนอกที่เชื่อมต่อกับแชนแนลการอินเทอร์รัพท์ทั้ง 8 ของ 8259A มีลำดับความสำคัญเท่า ๆ กัน

b.) โหมด Specific Rotation

ภายในโหมดนี้ สามารถเปลี่ยนแปลงลำดับความสำคัญในการขออินเทอร์รัพท์ของแชนแนลต่าง ๆ ได้โดยการกำหนดแชนแนลที่ต้องการให้มีลำดับความสำคัญต่ำที่สุด ซึ่งจะมีผลให้ลำดับความสำคัญของแชนแนลอื่น ๆ ถูกเปลี่ยนไปในลักษณะที่สัมพันธ์กัน

2.3) โหมด Special Mask

การทำงานภายในโปรแกรมตอบสนองการอินเทอร์รัพท์ บางครั้งอาจจำเป็นต้องเปลี่ยนลำดับความสำคัญของแชนแนลการอินเทอร์รัพท์ของระบบ โดยอาจจะต้องดีสเอเบิลการขออินเทอร์รัพท์จากแชนแนลที่มีลำดับความสำคัญต่ำกว่าแชนแนลที่ได้รับการตอบสนองอยู่นั้นในช่วงเวลาหนึ่ง และทำการรีนาเบิลการขออินเทอร์รัพท์จากแชนแนลที่มีลำดับความสำคัญต่ำกว่าแชนแนลที่ได้รับการตอบสนองอยู่นี้ในอีกช่วงเวลาหนึ่ง ซึ่งจะต้องใช้โหมด Special Mask เข้าช่วย โดยโหมดนี้จะใช้งานร่วมกับ OCW1 ในการกำหนดว่าแชนแนลใดที่จะถูกรีนาเบิล และแชนแนลใดจะถูกดีสเอเบิลการขออินเทอร์รัพท์

2.4) โหมด Poll

ภายในโหมดนี้เอาต์พุตจากขา INT ของ 8259A จะไม่ถูกใช้งาน โดยโปรแกรมที่ส่งคำสั่ง Poll ให้กับ 8259A จะทำการตอบสนองต่อการขออินเทอร์รัพท์ของอุปกรณ์หรือวงจรภายนอกเอง ซึ่งคำสั่งนี้จะมีผลกับ 8259A เหมือนกับเมื่อ 8259A ได้รับพัลส์ลูกแรกของสัญญาณ $\overline{INTA}$ ซึ่งจะทำให้บิตของรีจิสเตอร์ IS ที่ตรงกับแชนแนลที่มีการขออินเทอร์รัพท์นั้นถูกเซต

หลังจากที่ 8259A ได้รับคำสั่ง Poll แล้ว 8259A จะเข้าสู่โหมด Poll โดยจะรอสัญญาณ RD จาก CPU ซึ่งเมื่อได้รับสัญญาณแล้ว 8259A ก็จะส่งข้อมูลแสดงสถานะการอินเทอร์รัพท์ออกมาบนบัสข้อมูลดังนี้

D7	D6	D5	D4	D3	D2	D1	D0
I	-	-	-	-	W2	W1	W0

โดยที่ W0-W2 : เป็นรหัสของแชนแนลที่ขออินเทอร์รัพท์และมีลำดับความสำคัญสูงสุด
(ในจำนวนแชนแนลที่ขออินเทอร์รัพท์)

I : ข้อมูลในบิตนี้จะถูกเซตเป็น 1 เมื่อมีการขออินเทอร์รัพท์จากอุปกรณ์ภายนอกผ่านทางแชนแนลการอินเทอร์รัพท์ (ขา IRO-IR7 ขาใดขาหนึ่ง หรือหลายขาพร้อมกัน)

โดยทั่วไปใหม่นี้ จะถูกใช้งานในกรณีที่โปรแกรมตอบสนองการอินเทอร์รัพท์เพียงโปรแกรมเดียว สามารถทำการตอบสนองให้กับการขออินเทอร์รัพท์ได้ในหลายๆ แชนแนล ซึ่งจะทำให้สะดวกกว่าการใช้อินเทอร์รัพท์ CPU โดยตรงและยังช่วยประหยัดเนื้อที่ในหน่วยความจำที่เก็บตารางได้ด้วย

การโปรแกรม 8259A จะต้องเริ่มจาก ICW ก่อนเสมอ ซึ่งการเริ่มต้นจะทำได้โดยใช้คำสั่ง OUT ไปยังแอดเดรสของ 8259A โดยให้เส้นแอดเดรส A0 ของ 8259A มีลอจิกเป็น "0" และบิต D4 ของคำสั่งที่ส่งให้กับ 8259A มีลอจิกเป็น "1"

ภายใน IBM/PC จะจัดให้แอดเดรสของ 8259A ตรงกับแอดเดรสของพอร์ทเพียง 2 พอร์ท คือ 0020H และ 0021H โดยในขั้นแรกจะใช้แอดเดรส 0020H สำหรับการเข้าสู่การโปรแกรม ICW1 จากนั้นจึงใช้แอดเดรส 0021H สำหรับการโปรแกรม ICW2 , 3 และ 4 ตามลำดับ (เมื่อเสร็จจากการโปรแกรม ICW2 แล้ว 8259A จะทราบได้เองว่าคำสั่งต่อไปที่ส่งเข้ามาทางพอร์ทนั้นเป็นข้อมูลที่ส่งให้กับ ICW3 และข้อมูลไบต์ต่อไปเป็นของ ICW4 ตามลำดับ) เมื่อเสร็จจากการโปรแกรม ICW แล้ว 8259A จะเข้าสู่การโปรแกรม OCW โดยอัตโนมัติ โดยการโปรแกรม OCW1 จะผ่านทางแอดเดรสของพอร์ท 0021H ส่วน OCW2 จะต้องโปรแกรมผ่านทางพอร์ท 0020H โดยบิต D3,D4 ของคำสั่งที่ส่งให้กับ 8259A จะต้องเป็น "0" ทั้งคู่ และ สำหรับ OCW3 ก็จะใช้โปรแกรมผ่านทางพอร์ท 0020H เช่นกันแต่บิต D3,D4 ของคำสั่งที่ส่งให้กับ 8259A จะต้องเป็น "1" , "0" ตามลำดับ

Port Address	Register
HEX 0020	ICW1
HEX 0021	ICW2
HEX 0021	ICW3*
HEX 0021	ICW4
HEX 0021	OCW1
HEX 0020	OCW2
HEX 0020	OCW3

* skipped in PC initialization.

** Note These addresses are only valid if executed in the order shown.

Initialization Command Word : (ICW)

การโปรแกรม ICW ของ 8259A จะเริ่มต้นจาก ICW1 โดยในกรณีของ IBM/PC จะต้องใช้คำสั่ง OUT ส่งข้อมูลที่มีบิต D4 เป็น 1 ไปยังแอดเดรส 0020H ซึ่งจะทำให้ 8259A เข้าสู่โหมดการเริ่ม ICW1 และจะเกิดขั้นตอนต่าง ๆ ดังนี้

1.) โหมด Edge-Sensitive จะถูกเซต ซึ่งจะส่งผลให้การขออินเทอร์รัพท์ (หลังจากที่มีการโปรแกรม 8259A แล้ว) ผ่านทางแชนแนลทั้ง 8 ของ 8259A จะเกิดได้ต่อเมื่อระดับสัญญาณที่แชนแนลนั้น ๆ เปลี่ยนระดับลอจิกจาก 0 เป็น 1 (ขอบขาขึ้น) เท่านั้น

2.) รีจิสเตอร์ IS จะถูกเคลียร์ (รีเซตเป็น 0) เพื่อให้ 8259A พร้อมทั้งจะทำการขออินเทอร์รัพท์ได้

3.) การขออินเทอร์รัพท์ผ่านแชนแนลที่ 7 (IR7) จะถูกจัดให้มีลำดับความสำคัญต่ำสุดในขณะที่การขออินเทอร์รัพท์ผ่านทางแชนแนลที่ 0 (IR0) จะมีลำดับความสำคัญสูงสุด

4.) แอดเดรสในโหมดสลาฟ (ในกรณีที่มีการใช้ 8259A มากกว่า 2 ตัว) จะถูกเซตเป็น 7

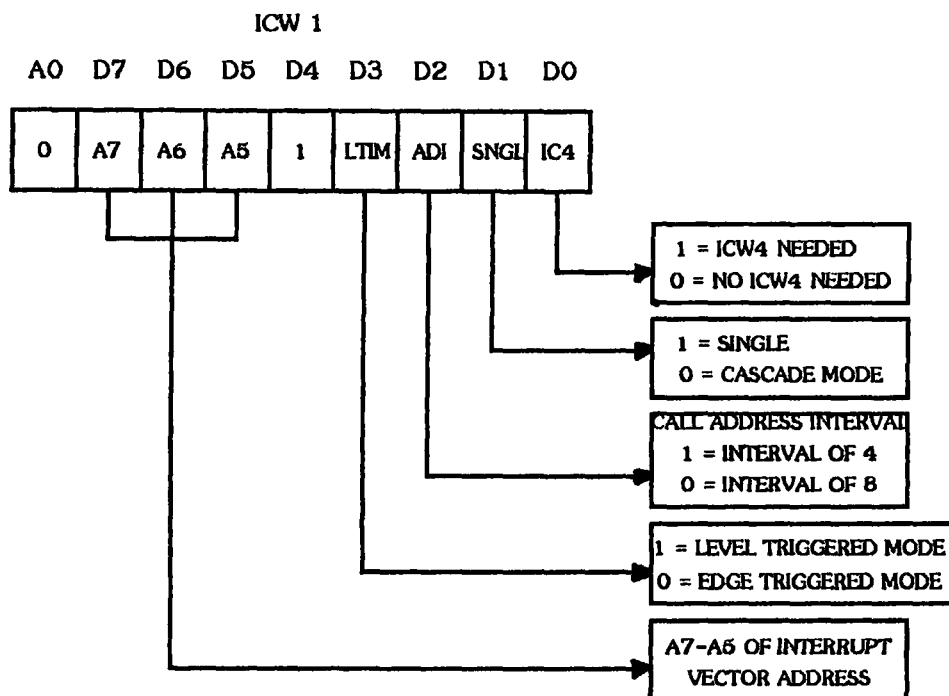
5.) โหมด Special Mask จะถูกเคลียร์ และ การอ่านสถานะของ 8259A จะถูกเซตให้เป็นการอ่านข้อมูลในรีจิสเตอร์ IR ของ 8259A

6.) ถ้าบิต IC4 ของ ICW1 ถูกเซตเป็น 0 ข้อมูลทุกบิตของ ICW4 จะถูกเซตเป็น 0 ทั้งหมด และการทำงานจะเป็นการทำงานร่วมกับ 8088 หรือ 8085

ต่อไปนี้จะกล่าวถึงรายละเอียดหน้าที่ของบิตต่าง ๆ ใน ICW ทั้ง 4 ตัว

ICW1 :

หน้าที่ของบิตต่าง ๆ ภายใน ICW1 นั้นแสดงได้ดังไดอะแกรมข้างล่างนี้



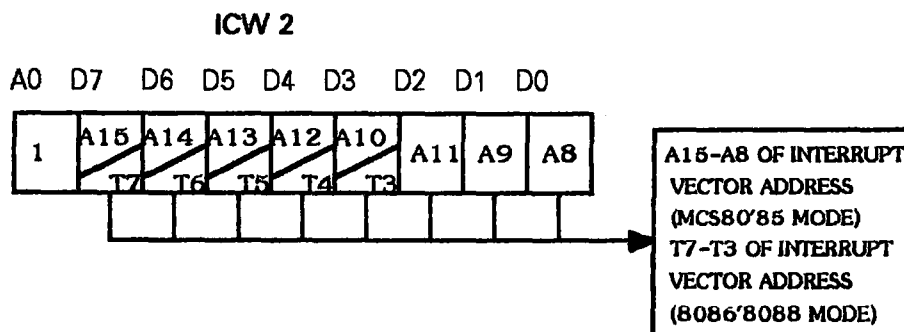
รูปที่ 2-33 แสดงบิตต่างๆภายใน ICW1

- บิต D0 : ข้อมูลในบิตนี้จะใช้กำหนดว่าในการโปรแกรม 8259A นี้ จะทำการโปรแกรม
(IC4) ICW4 ด้วยหรือไม่ ถ้าบิตนี้เป็น 0 ก็จะไม่มีการโปรแกรม ICW4 แต่ถ้าบิตนี้เป็น " 1 " ก็จะมีการโปรแกรม ICW4 ด้วย
- บิต D1 : ข้อมูลในบิตนี้จะถูกใช้ในการแสดงให้ 8259A ทราบว่า ภายในระบบมีการใช้
(SNGL) 8259A มากกว่า 1 ตัว (ต่อ Cascade) หรือไม่ ถ้าบิตนี้เป็น 1 แสดงว่าในระบบ มี 8259A เพียง 1 ตัว ถ้าบิตนี้เป็น 0 แสดงว่าในระบบมี 8259A มากกว่า 1 ตัว
- บิต D2 : จะถูกใช้งานในกรณีที่ทำงานร่วมกับ 8080 หรือ 8085 เท่านั้น ดังนั้น ภายใน
(ADI) IBM/PC จึงไม่มีการใช้งานบิตนี้ โดยจะถูกเซตเป็น 0

- บิต D3 : ข้อมูลในบิตนี้ถูกใช้ในการกำหนด หรือ เลือกโหมดการตรวจสอบการขอ
(LTIM) อินเทอร์รัพท์ ซึ่งโหมดการตรวจสอบการขออินเทอร์รัพท์ จะแบ่งออกเป็น 2 โหมด คือ
- 1.) โหมด Level-Trigger (ข้อมูลในบิตนี้จะถูกเซตเป็น 1) : ในโหมดนี้ 8259A จะรับทราบการขออินเทอร์รัพท์เมื่อระดับลอจิกของสัญญาณที่ขา IR เป็นลอจิก 1 และ ระดับลอจิกที่ขา IR นี้จะต้องเปลี่ยนเป็นลอจิก 0 ก่อนที่ 8259A จะได้รับคำสั่ง EOI หรือ ก่อนที่ CPU จะถูกอินาเบลให้ยอมรับการ ขออินเทอร์รัพท์ได้อีก
 - 2.) โหมด Edge-Trigger (ข้อมูลในบิตนี้จะถูกเซตเป็น 0) : ในโหมดนี้ 8259A จะรับทราบการขออินเทอร์รัพท์เมื่อระดับลอจิกที่ขา IR เปลี่ยนจากลอจิก 0 เป็น 1 (ขอบขาขึ้น) ดังนั้น ภายในโหมดนี้จึงไม่มีความจำเป็นที่ต้องยกเลิก ระดับลอจิก 1 ที่ขา IR ก่อนที่ 8259A จะได้รับคำสั่ง EOI หรือ ก่อนที่ 8259A จะถูกอินาเบล
- บิต D4 : ข้อมูลในบิตนี้จะถูกเซตเป็น 1 เพื่อแสดงว่าข้อมูลที่ส่งให้กับ 8259A นี้เป็นข้อมูลสำหรับการโปรแกรม ICW1
- บิต D5-D7 : บิตทั้ง 3 นี้จะถูกใช้ในระบบของ 8080/8085 เท่านั้น ดังนั้น ภายใน IBM/PC (A5-A7) ซึ่งใช้ 8088 เป็น CPU จะไม่ใช้งาน 3 บิตนี้ และจะเซตข้อมูลในบิตทั้ง 3 นี้เป็น 0

ICW2

หน้าที่ของบิตต่าง ๆ ภายใน ICW2 แสดงได้ดังไดอะแกรมข้างล่างนี้

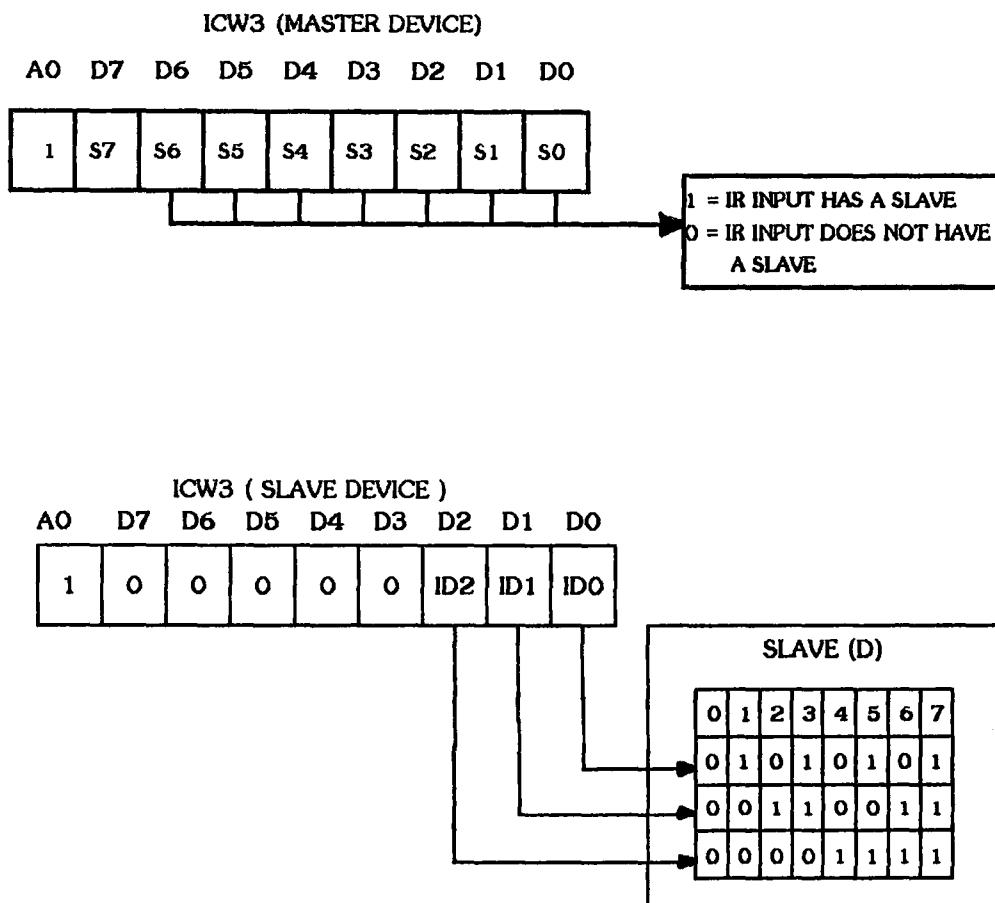


รูปที่ 2-34 แสดงบิตต่างๆภายใน ICW2

- บิต D0-D2 : ข้อมูลในบิตทั้ง 3 นี้ ถูกใช้งานในระบบที่ใช้ 8080/8085 เป็น CPU ดังนั้นใน (A8-A10) IBM/PC จึงเซตให้บิตทั้ง 3 เป็น 0
- บิต D3-D7 : ข้อมูลในบิตทั้ง 5 นี้ถูกใช้งานทั้งในระบบที่ใช้ 8080/8085 เป็น CPU และในระบบที่ใช้ 8088 เป็น CPU โดยในระบบที่ใช้ 8080/8085 นั้น บิตทั้ง 5 นี้ จะมีหน้าที่เป็น (T3-T7) A12-A15 และถูกใช้งานร่วมกับ บิต D0-D2 ของ ICW2 และ บิต D5-D7 ของ ICW1 เพื่อประกอบเป็นเวคเตอร์การอินเทอร์รัพท์สำหรับระบบที่ใช้ 8088 จะทำหน้าที่เป็น T3-T7 ซึ่งก็คือ A5-A9 ที่กล่าวถึงในหัวข้อตารางเวคเตอร์ของ IBM/PC นั่นเอง

ICW3 :

หน้าที่ของบิตต่าง ๆ ภายใน ICW3 จะแสดงตามไดอะแกรมข้างล่างนี้



รูปที่ 2-35 แสดงบิตต่างๆภายใน ICW3

ICW3 ถูกใช้งานเฉพาะภายในระบบที่มีการใช้ 8259A มากกว่า 1 ตัว มาต่อกันแบบ Cascade ดังนั้น ICW3 จะถูกใช้งานในกรณีที่บิต SNGL ของ ICW1 ถูกโปรแกรมให้เป็น 0 เท่านั้น ICW3 นี้แบ่งได้เป็น 2 ประเภท โดยขึ้นอยู่กับว่า 8259A ที่เราทำการโปรแกรมนั้นทำงานเป็น Master หรือ Slave

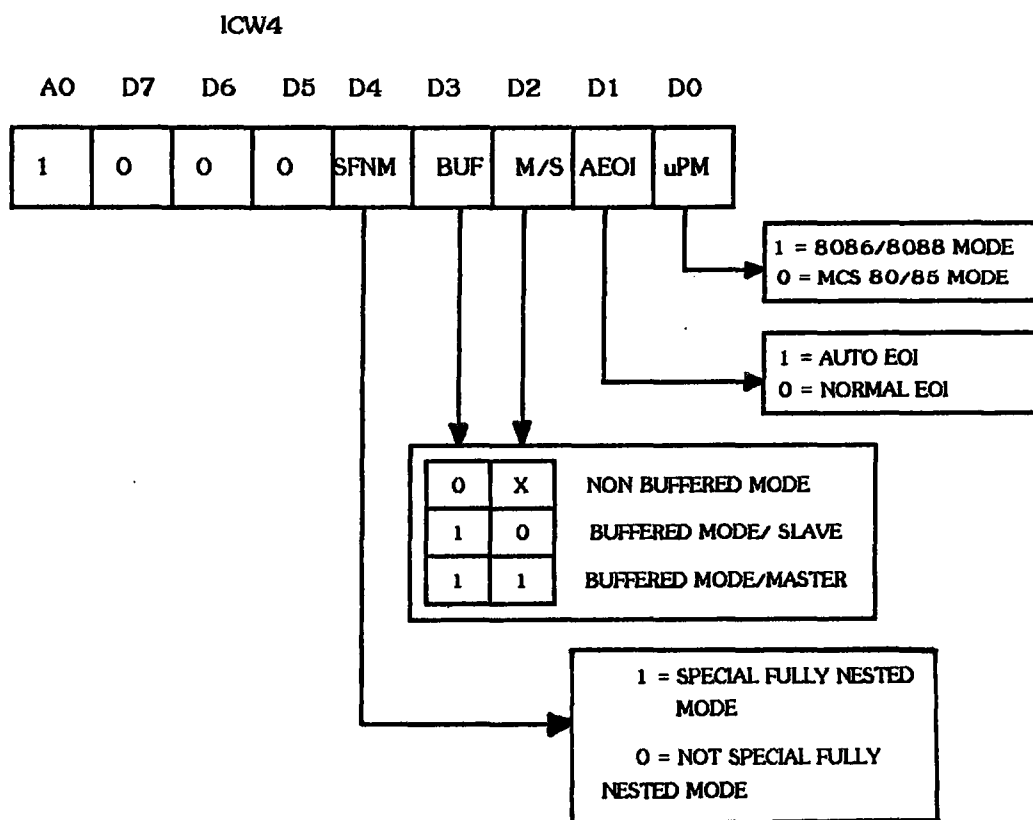
1.) **กรณีที่ทำงานเป็น Master** เมื่อมีการอินเทอร์รัพต์ผ่านทางแชนแนลการอินเทอร์รัพต์ของ 8259A 8259A จะตรวจสอบข้อมูลใน ICW3 ถ้าพบว่าบิตที่ตรงกับแชนแนลที่ทำการอินเทอร์รัพต์นั้น มีข้อมูลเป็น 1 ก็แสดงว่า การขออินเทอร์รัพต์นั้นเป็นการขออินเทอร์รัพต์ผ่านทาง

8259A ที่ทำหน้าที่เป็น Slave ในกรณีนี้ 8259A ที่ เป็น Master ก็จะส่งรหัส (เป็นเลขฐานสองที่ตรงกับ แชนแนลที่ถูกขออินเทอร์รัพท์) ออกทางบัส Cascade เพื่อให้ 8259A ที่ เป็น Slave ที่ทำการขอ อินเทอร์รัพท์นั้น ส่งเวคเตอร์อินเทอร์รัพท์ให้กับ 8088

2.) กรณีที่ทำงานเป็น Slave กรณีนี้บิต D0-D2 ของ ICW3 (บิต ID0-ID2) จะถูกโปรแกรมเพื่อกำหนดรหัสของ 8259A ที่ เป็น Slave โดยเมื่อ 8259A ที่ เป็น Slave ได้รับรหัสจาก 8259A ที่ เป็น Master แล้ว 8259A ที่ เป็น Slave ก็จะตรวจสอบรหัสที่ได้รับนี้กับรหัส (ข้อมูล) ในบิต ID0-ID2 ของ ICW3 ถ้าพบว่ารหัสทั้ง 2 นี้ตรงกัน 8259A ที่ เป็น Slave ก็จะทำการส่งเวคเตอร์อินเทอร์รัพท์ออกมาบนบัสข้อมูลเพื่อส่งให้กับ 8088

ICW4 :

หน้าที่ของบิตต่าง ๆ ภายใน ICW4 จะแสดงได้ดังไดอะแกรมข้างล่างนี้



รูปที่ 2-36 แสดงบิตต่างๆภายใน ICW4

- บิต D0 : ข้อมูลในบิตนี้จะแสดงให้ 8259A ทราบว่า CPU ที่ 8259A ทำงานร่วมด้วยนั้นเป็น
(uPM) 8082/8085 หรือ 8086/8088 โดยถ้าบิตนี้ถูกเซตเป็น 0 ก็จะทำให้เข้าสู่โหมดการทำงานร่วมกับ 8080/8085 แต่ถ้าบิตนี้ถูกเซตเป็น 1 ก็เข้าสู่โหมดการทำงานร่วมกับ 8086/8088
- บิต D1 : ใช้ในการเลือกโหมด EOI ของ 8259A ดังนี้
(AEOI) - ถ้าข้อมูลในบิตนี้ถูกเซตเป็น 0 8259A จะเข้าสู่โหมด Normal EOI โดยภายในโหมดนี้จะต้องทำการรีเซตบิตของรีจิสเตอร์ IS เอง โดยการส่งคำสั่ง EOI ให้กับ 8259A ก่อนที่จะสิ้นสุดการทำงานในโปรแกรมตอบสนองการอินเทอร์รัพท์ที่เกิดขึ้นและโหมดนี้ยังแบ่งได้เป็นอีก 2 โหมด ย่อย ๆ คือ
- 1.) โหมด Specific EOI : โหมดนี้จะสามารถกำหนดบิตของรีจิสเตอร์ที่ต้องการรีเซตได้
 - 2.) โหมด Non-Specific EOI : โหมดนี้เมื่อ 8259A ได้รับคำสั่ง EOI แล้ว ก็จะมีการเซตบิตของรีจิสเตอร์ IS ที่ถูกเซตโดยฮาร์ดแวร์ สำหรับในกรณีที่มีบิตของรีจิสเตอร์ IS ถูกเซตมากกว่า 1 บิต 8259A ก็จะมีการรีเซตเฉพาะบิตที่มีลำดับความสำคัญสูงสุดในจำนวนบิตที่ถูกเซตนั้น อย่างไรก็ตาม การทำงานในลักษณะข้างต้นเป็นการทำงานเมื่อ 8259A อยู่ในโหมด Fully Nested เท่านั้นและถ้า 8259A ทำงานอยู่ในโหมดอื่นที่มีโครงสร้างต่างไปจากโหมด Fully Nested ก็จะต้องโปรแกรมให้ 8259A ทำงานในโหมด Specific EOI
- ถ้าข้อมูลในบิตนี้ถูกเซตเป็น 1 8259A จะเข้าสู่โหมด AEOI ภายในโหมดนี้เมื่อ 8259A ได้รับพัลส์ลูกที่ 2 ของสัญญาณ INTA แล้วก็จะทำการรีเซตบิตของรีจิสเตอร์ IS ที่ถูกเซตโดยฮาร์ดแวร์ และในกรณีที่มีบิตของรีจิสเตอร์ IS ถูกเซตอยู่มากกว่า 1 บิต 8259A ก็จะทำกาเลือกรีเซตเฉพาะบิตที่มีลำดับความสำคัญสูงสุดในจำนวนบิตที่ถูกเซต สำหรับในกรณีที่มีการใช้งาน 8259A มากกว่า 1 ตัวนั้น โหมด AEOI นี้จะใช้เฉพาะ 8259A ที่เป็นตัว Master เท่านั้น
- บิต D2 : ใช้แสดงให้ 8259A ทราบว่า จะต้องทำงานเป็น Master หรือ Slave โดยถ้าเป็น 1
(M/S) ก็แสดงว่า 8259A นี้ทำงานเป็น Master และถ้าเป็น 0 ก็แสดงว่าทำงานเป็น Slave สำหรับหน้าที่ของบิตนี้ ขึ้นอยู่กับบิต D3 ของ ICW4 (บิต BUF) โดย ถ้าต้องการใช้งานบิต M/S นี้ บิต BUF จะต้องถูกเซตเป็น 1

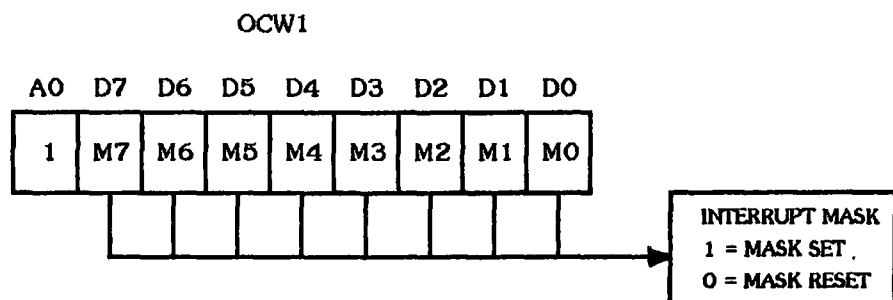
- บิต D3 : ถ้าบิตนี้ถูกเซตเป็น 1 8259A จะเข้าสู่การทำงานในโหมด Buffer (BUF)
- บิต D4 : ถ้าข้อมูลในบิตนี้มีค่าเป็น 1 ก็จะเป็นการสั่งให้ 8259A เข้าสู่โหมด Special Fully (SFNM) Nested ซึ่งเป็นโหมดที่ใช้กับระบบที่มีการใช้งาน 8259A หลายตัวร่วมกันในลักษณะของ Cascade สำหรับการโปรแกรม 8259A ให้เข้าสู่การทำงานในโหมดนี้ จะต้องทำการโปรแกรมเฉพาะ 8259A ที่ทำหน้าที่เป็น Master เท่านั้น
- บิต D5-D7 : บิตทั้ง 3 นี้ไม่ถูกใช้งาน และจะถูกเซตให้เป็น 0

Operation Command Word (OCW)

หลังจากที่ได้รับการโปรแกรม ICW เรียบร้อยแล้ว 8259A ก็พร้อมที่จะขอรับการอินเทอร์รัพท์ทางแชนแนลทั้ง 8 ได้แต่ก็ยังสามารถที่จะกำหนดการทำงานในลักษณะอื่น ๆ ของ 8259A ได้อีกโดยการโปรแกรม OCW ทั้ง 3 ของ 8259A

OCW1 :

จะใช้ในการอ่านาเบิลหรือดิสเอเบิลการขออินเทอร์รัพท์ทางแชนแนลทั้ง 8 ของ 8259A สำหรับการจับบิตต่าง ๆ ของ 8259A แสดงได้ดังไดอะแกรมข้างล่างนี้

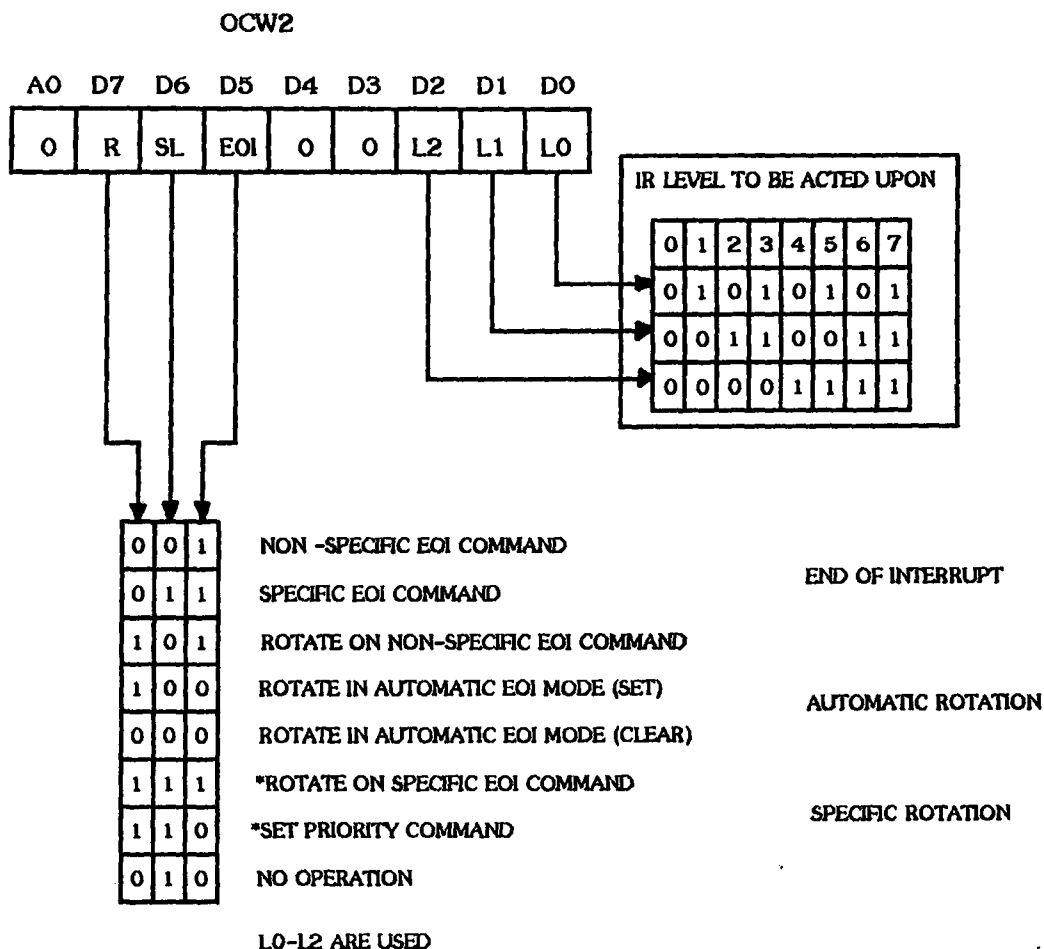


รูปที่ 2-37 แสดงบิตต่างๆภายใน OCW1

ข้อมูลในแต่ละบิตใน OCW1 จะถูกใช้ในการควบคุมการดิสเอเบิลหรือการอ่านาเบิล โดยบิตที่ตรงกับแชนแนลใดก็ควบคุมการดิสเอเบิลหรืออ่านาเบิลการขออินเทอร์รัพท์ของแชนแนลนั้น และถ้าข้อมูลในบิตใดเป็น 1 ก็จะเป็นการดิสเอเบิล แต่ถ้าข้อมูลในบิตใดเป็น 0 ก็จะเป็นการอ่านาเบิลสำหรับสถานะของ OCW1 นี้จะทราบได้โดยการอ่านข้อมูลผ่านทางพอร์ท 0021H เข้ามา

OCW2 :

ใช้สำหรับการเลือก และการเปลี่ยนแปลงลำดับความสำคัญของการอินเทอร์รัพท์ใน
 แชนแนลทั้ง 8 ของ 8259A รวมทั้งการเลือกโหมด EOI สำหรับการจับเรียงบิตต่าง ๆ แสดงดังได้
 เดอะแแกรมข้างล่างนี้



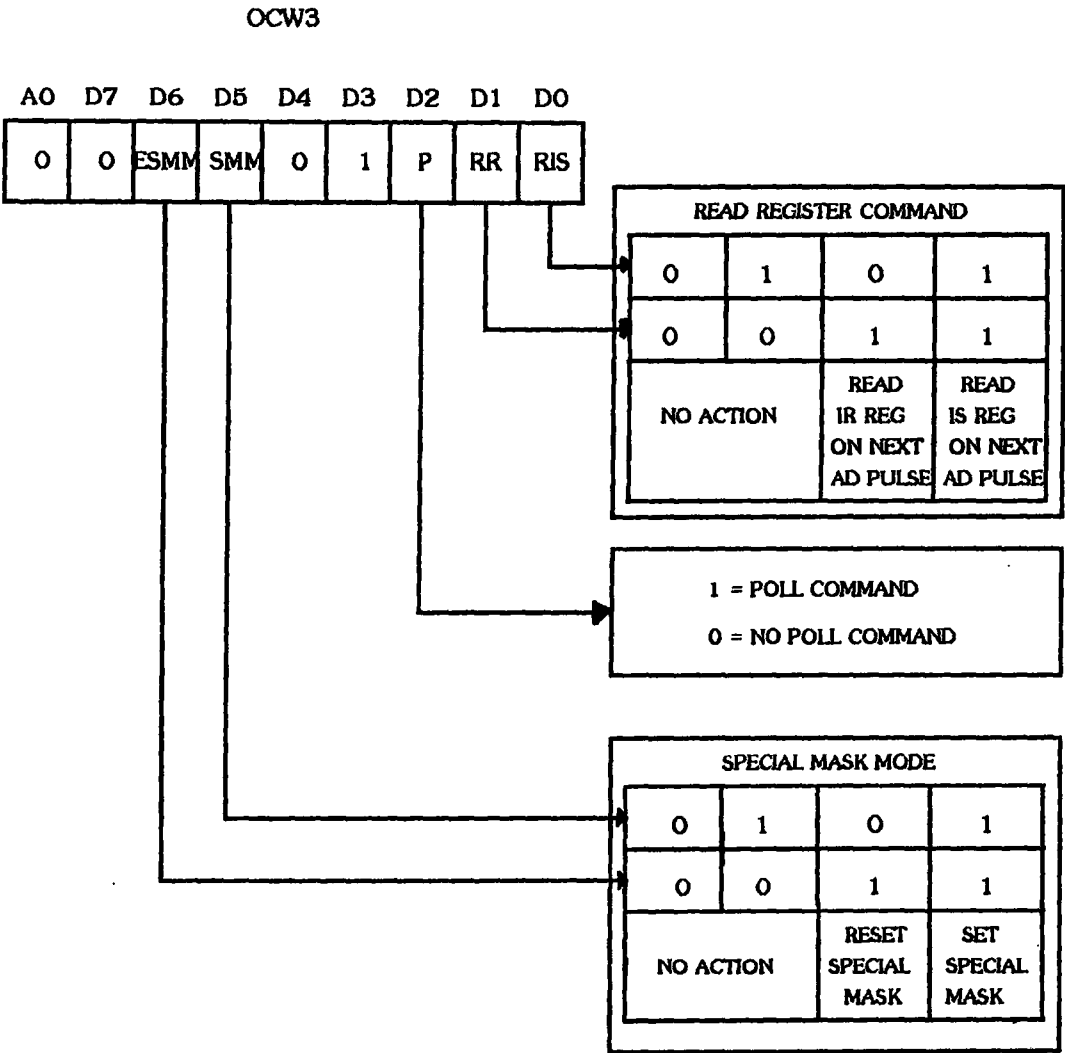
รูปที่ 2-38 แสดงบิตต่างๆภายใน OCW2

- บิต D0-D2 : ใช้สำหรับการกำหนดว่าคำสั่งที่ส่งให้กับ OCW2 นี้ จะมีผลกับแชนแนล (ขา IR)
 (L0-L2) ไດ ซึ่งข้อมูลดังกล่าวนี้ จะเป็นรหัสเลขฐานสองของแชนแนลที่ต้องการ อย่างไรก็ตาม บิตทั้ง 3 จะถูกใช้งานหรือไม่ ขึ้นอยู่กับข้อมูลในบิต D6 (บิต SL) ของ OCW2 ด้วย
- บิต D3-D4 : บิตทั้ง 2 จะไม่ถูกใช้งาน และจะถูกเซตให้เป็น 0
- บิต D5-D7 : ใช้กำหนดโหมดการทำงานต่าง ๆ ของ 8259A การนำเอาข้อมูลในบิตทั้ง 3 นี้มา ประกอบกัน จะทำให้เกิดการทำงานใน 8 ลักษณะที่แตกต่างกัน ดังนี้

D7	D6	D5	การทำงาน
0	0	1	คำสั่ง Non-Specific EOI : คำสั่งนี้จะถูกใช้ในการสั่งให้ 8259A ทำการรีเซ็ต(เคลียร์ให้มีค่าเป็น 0) บิตรีจิสเตอร์ IS ที่มีลำดับความสำคัญสูงสุดในกลุ่มของบิตที่ถูกเซ็ต โดยคำสั่งนี้จำเป็นต้องต้องใช้เมื่อบิต D1 ของ ICW4 มีค่าเป็น 0
0	1	1	คำสั่ง Specific EOI : ลักษณะการทำงานคล้ายกับในกรณีของ Non-Specific EOI แต่ในคำสั่งนี้จะสามารถกำหนดบิตของรีจิสเตอร์ IS ที่ต้องการจะรีเซ็ตได้ด้วย โดยกำหนดบิตที่ต้องการรีเซ็ตลงในบิต D0-D2 (เป็นเลขฐานสอง)
1	0	1	คำสั่ง Non-Specific EOI : ซึ่งทำการรีเซ็ตบิตของรีจิสเตอร์ IS ที่มีลำดับความสำคัญสูงสุดในกลุ่มของบิตที่ถูกเซ็ตและทำการเปลี่ยนแปลงลำดับความสำคัญของแชนแนลต่าง ๆ ใน 8259A โดยให้แชนแนลที่ตรงกับบิตที่ถูกรีเซ็ตนั้นมีลำดับความสำคัญต่ำสุด และเปลี่ยนแปลงลำดับความสำคัญของแชนแนลอื่น ๆ ในลักษณะที่สัมพันธ์กัน
1	0	0	โหมด Automatic Rotating Priority : เป็นคำสั่งให้ 8259A เปลี่ยนแปลงลำดับความสำคัญในการขออินเทอร์รัพท์ของแชนแนลต่าง ๆ ทั้ง 8 แชนแนลเมื่อเกิด AEOI ขึ้น
0	0	0	เป็นการยกเลิกโหมด Automatic Rotating Priority
1	1	1	คำสั่ง Specific EOI ซึ่งทำการรีเซ็ตบิตของรีจิสเตอร์ IS ที่ถูกกำหนดไว้ในบิต D0-D2 (ฐานสอง) และทำการเปลี่ยนแปลงลำดับความสำคัญของแชนแนลต่าง ๆ ด้วย โดยให้แชนแนลที่ตรงกับบิตที่ถูกรีเซ็ตมีลำดับความสำคัญต่ำสุดและเปลี่ยนแปลงลำดับความสำคัญของแชนแนลอื่น ๆ ในลักษณะที่สัมพันธ์กัน
1	1	0	เป็นคำสั่งที่ใช้ในการกำหนดลำดับความสำคัญให้กับแชนแนลต่าง ๆ ของ 8259A โดยบิต D0-D2 จะใช้กำหนด แชนแนลที่ต้องการให้มีลำดับความสำคัญต่ำสุด ซึ่งเป็นผลให้แชนแนลอื่น ๆ ถูกเปลี่ยนแปลงลำดับความสำคัญไปในลักษณะที่สัมพันธ์กันด้วย

OCW3 :

ไดอะแกรมข้างล่างแสดงการจัดเรียงบิตต่าง ๆ ภายใน OCW3 (เส้นแอดเดรส A0 ของ 8259A จะต้องมียกระดับลอจิกเป็น "0" และ บิต D3, D4 และ D7 จะต้องถูกเซ็ทเป็น "1" , "0" และ "0" ตามลำดับ)



รูปที่ 2-39 แสดงการจัดเรียงบิตต่างๆภายใน OCW3

บิต D0-D1 : ข้อมูลใน 2 บิตนี้ จะถูกใช้ในการอ่านสถานะของ 8259A โดยข้อมูลในบิต RIS (บิต D0) จะใช้สำหรับการเลือกรีจิสเตอร์สถานะของ 8259A คือ IR และ IS โดยถ้าข้อมูลในบิต RIS เป็น 1 ก็จะเป็นการเลือกรีจิสเตอร์ IS ถ้าเป็น 0 ก็จะเป็นการเลือกรีจิสเตอร์ IR อย่างไรก็ตามหน้าที่ของบิต RIS นี้ยังขึ้นอยู่กับข้อมูลในบิต RR (บิต D1) โดยถ้าข้อมูลในบิตนี้มีค่าเป็น 0 ก็จะไม่เกิดผลใด ๆ ขึ้นไม่ว่าข้อมูลในบิต RIS จะเป็น 0 หรือ 1 สำหรับข้อมูลในรีจิสเตอร์ทั้ง 2 นั้น จะทำหน้าที่ในการแสดงสถานะของ 8259A ที่แตกต่างกัน ดังนี้

รีจิสเตอร์ IR (Interrupt Request Register)

ใช้ในการแสดงว่าแชนแนลใดบ้างที่ได้รับสัญญาณขออินเทอร์รัพท์ CPU จากอุปกรณ์หรือวงจรรภายนอก

รีจิสเตอร์ IS (In-Service Register) :

ใช้ในการแสดงว่าแชนแนลใดบ้างที่ได้รับการตอบสนองอยู่ โดยข้อมูลภายในบิตต่าง ๆ ของรีจิสเตอร์ IS จะถูกเปลี่ยนแปลงเมื่อ 8259A ได้รับคำสั่ง EOI

บิต D2 : ในกรณีที่ข้อมูลในบิตนี้ถูกเซตเป็น 1 8259A จะเข้าสู่การทำงานในโหมด Poll (P)

บิต D3-D4 : บิตทั้ง 2 นี้ใช้ในการแสดงว่าข้อมูลที่ส่งให้กับ 8259A นี้เป็นข้อมูลที่ใช้ในการโปรแกรม OCW 3 โดยข้อมูลในบิตนี้จะต้องเป็น 1 และ 0 ตามลำดับ

บิต D5 : ในกรณีที่ข้อมูลในบิตนี้ถูกเซตเป็น 1 8259A จะเข้าสู่การทำงานในโหมด Specific (SMM) Mask อย่างไรก็ตาม ข้อมูลในบิต D5 นี้จะขึ้นอยู่กับข้อมูลในบิต D6 ด้วย

บิต D6 : ใช้งานร่วมกับบิต D5 ในการกำหนดโหมด Mask ของ 8259A โดยข้อมูลในบิต D5 จะมีผลต่อการทำงานของ 8259A เฉพาะในกรณีที่บิต D6 มีข้อมูลเป็น 1 เท่านั้น

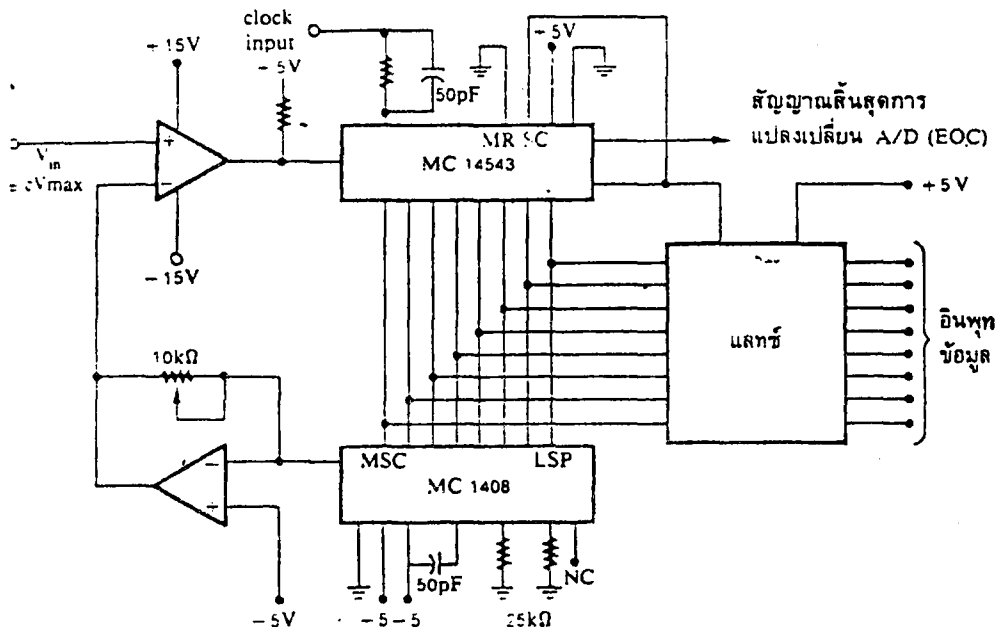
2.5 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลและแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก

2.5.1 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล

ส่วนสำคัญในการเชื่อมต่อหรือสื่อสารระหว่างเครื่องมือทางวิทยาศาสตร์กับคอมพิวเตอร์ก็คือการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล โดยที่เครื่องมือทางวิทยาศาสตร์ส่วนมาก จะให้เอาต์พุตเป็นสัญญาณอนาลอก ADC จะแปลงให้เป็นสัญญาณที่ไม่ต่อเนื่อง คือ สัญญาณดิจิทัล และส่งผ่านไปยังคอมพิวเตอร์ โดยในปัจจุบันได้มีการออกแบบวงจร ADC ออกมามากมายหลายประเภท ซึ่งแต่ละประเภทก็มีข้อดีจำกัดที่แตกต่างกัน

1. การประมาณค่าผลสำเร็จ (Successive Approximation , SA)

ADC ชนิดนี้มีการใช้งานกันอย่างกว้างขวาง เนื่องจากมีความละเอียดสูง (16 บิต) และความเร็วสูง (1 MHz) เทคนิคการแปลงชนิดนี้ อาศัยการเปรียบเทียบสัญญาณอินพุตที่ไม่ทราบค่า กับแรงดันหรือกระแสจากการแปลงสัญญาณดิจิทัลให้เป็นสัญญาณอนาลอก(ADC) ชนิดความแม่นยำสูงซึ่งอินพุตของ DAC ก็คือค่าดิจิทัลที่เอาต์พุตของ DAC หัวใจของการแปลงชนิดนี้คืออุปกรณ์ที่เรียกว่า รีจิสเตอร์ประมาณค่าแบบผลสำเร็จ (Successive Approximation Register , SAR) เป็นอุปกรณ์ที่ต่างจากวงจรนับทั่ว ๆ ดังรูปที่ 2-40



รูป 2-40 บล็อกไดอะแกรมของ A/D Converter แบบ SAR

การแปลงเริ่มต้นเมื่อสัญญาณอนาล็อกถูกป้อนให้กับคอนเวอร์เตอร์และพัลส์ของการคอนเวอร์เตอร์ได้เริ่มขึ้นถูกป้อนให้กับตัว SAR พัลส์สัญญาณนาฬิกาถูกแรกที่ป้อนให้กับตัว SAR จะ "ON" เอาท์พุทของบิตหนักสูงสุดจากนั้นจึงเป็นการปรับให้เอาท์พุทของ DAC เป็น 50% ของแรงดันของอินเวอร์เตอร์ตัว SAR จะมองไปยังเอาท์พุทของวงจรเปรียบเทียบว่า เอาท์พุทของ DAC มีค่ามากกว่าหรือน้อยกว่าสัญญาณอนาล็อกทางอินพุท ถ้าแรงดันของ DAC มีค่ามากกว่าวงจรเปรียบเทียบจะยังคงสถานะ "OFF" ดังนั้นตัว SAR จะ "OFF" บิตหนักสำคัญสูงสุดลง และให้ชื่อว่าสถานะ "0" ถ้าแรงดันของ DAC มีค่าน้อยกว่าสัญญาณอนาล็อกทางอินพุท วงจรเปรียบเทียบจะยังคงทำงานอยู่ ดังนั้นตัว SAR จะยังคงปล่อยให้บิตหนักสำคัญสูงสุด "ON" และเราเรียกสถานะนี้ว่า "1" ซึ่งสถานะทั้งสองนี้จะกระทำภายในพัลส์ของสัญญาณนาฬิกาเพียงพัลส์เดียว บนสัญญาณนาฬิกาถัดไป ตัว SAR จะ "ON" บิตหนักสำคัญสูงสุดอันดับสอง และทำการตรวจสอบผลลัพธ์ที่ได้อีกครั้งจากวงจรเปรียบเทียบ ถ้าสัญญาณจาก DAC ครั้งใหม่นี้มีค่ามากกว่าแรงดันอินพุท-เอาท์พุทของวงจรเปรียบเทียบจะไม่มี ดังนั้นตัว SAR จะ "OFF" บิตหนักสำคัญสูงสุดอันดับสอง และเรียกมันว่า "0" แต่ถ้าสัญญาณจาก DAC มีค่าน้อยกว่าวงจรเปรียบเทียบจะยังคงทำงาน และตัว SAR จะปล่อยให้ บิตหนักสำคัญสูงสุดอันดับสองนี้ "ON" ตัว SAR จะพิจารณาแต่ละบิตด้วยวิธีเดียวกันจนกระทั่งทุก บิตถูกพิจารณาหมด เนื่องจากหนึ่งบิตถูกหาค่าภายในหนึ่งพัลส์ ADC ขนาด 8 บิต จึงใช้สัญญาณนาฬิกาเพียง 8 พัลส์ก็สามารถทำการแปลงได้จนครบกระบวนการ เมื่อบิตหนักสำคัญต่ำสุดถูกพิจารณาเสร็จสิ้นแล้ว ตัว SAR จะส่งสัญญาณสิ้นสุดการแปลงไปทำการคำนวณผลลัพธ์ที่ได้ซึ่งเป็นเลขฐานสองทางเอาท์พุทไว้

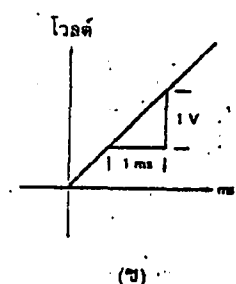
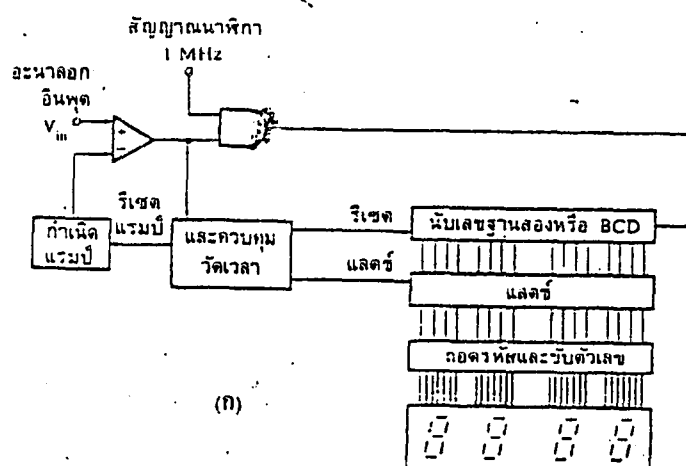
2. การอินทิเกรต (Intergration Conversion)

คอนเวอร์เตอร์ตระกูลนี้เป็นการแปลงสัญญาณแบบทางอ้อมโดยอันดับแรกจะแปลงเป็นฟังก์ชันของเวลา แล้วแปลงจากฟังก์ชันของเวลาให้อยู่ในรูปของดิจิตอลด้วยการใช้วงจรนับหรือเป็นการอินทิเกรตสัญญาณนั่นเอง แบ่งได้เป็น 3 แบบดังนี้

2.1) แบบสโลปเดียว (Single Slope or Single Ramp)

วงจร ADC ชนิดนี้ แสดงดังรูปที่ 2-41 ประกอบด้วยวงจรถ่ายค่าสัญญาณแรมป์ วงจรเปรียบเทียบวงจรรับเลขฐานสอง โดยเมื่อเริ่มทำการแปลงสัญญาณ สัญญาณแรมป์และวงจรรับจะถูกรีเซ็ตให้เป็น 0 แรงดันอนาล็อกถูกป้อนเข้าไปยังวงจรเปรียบเทียบทางขาอนอินเวอร์ตอินพุท เมื่อแรงดันอินพุทที่ขานี้เป็นบวกมากกว่าที่ขาอินเวอร์ตอินพุทวงจรเปรียบเทียบก็จะให้เอาท์พุทเป็น

ระดับ "high" ทำให้แอนดเกิตปล่อยสัญญาณนาฬิกาผ่านไปยังวงจรนับได้ และทำให้เริ่มเกิดสัญญาณแรมป์



รูป 2-41 วงจร A/D Converter แบบสโลปเดียว

(ก) บล็อกไดอะแกรม

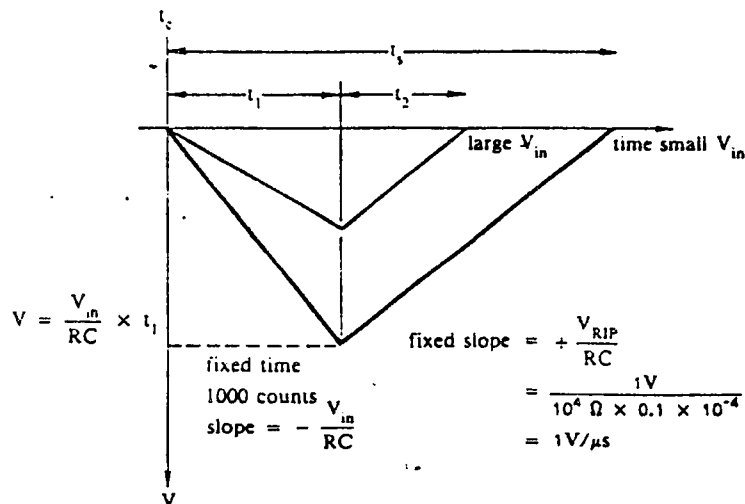
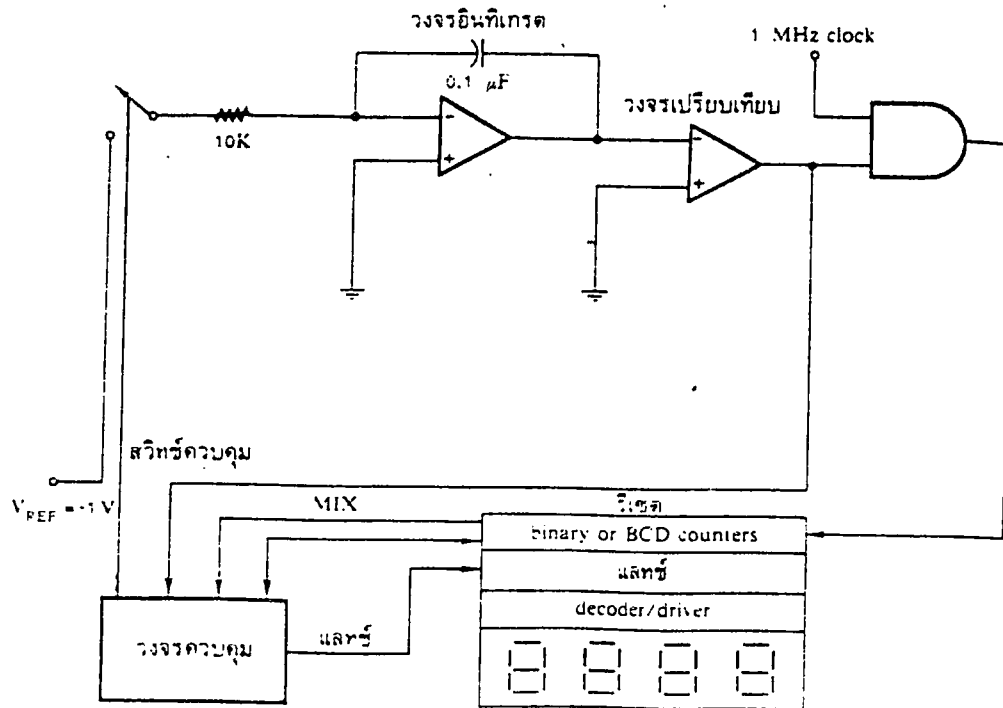
(ข) ความชันของสัญญาณแรมป์

สัญญาณแรมป์มีแรงดันเป็นบวกเพิ่มขึ้นเรื่อยๆจนมากกว่าระดับแรงดันอินพุต-เอาต์พุตของวงจรเปรียบเทียบจะตกลงมาเป็นระดับ "low" ทำให้แอนดเกิตปล่อยสัญญาณนาฬิกาผ่านไปยังวงจรนับได้ และทำให้เริ่มเกิดสัญญาณแรมป์

2.2) แบบสโลปคู่ (Dual Slope)

เทคนิคการแปลงแบบสโลปคู่เป็นเทคนิคที่ให้ข้อดีทางเสถียรภาพของการแปลงสัญญาณเมื่อสัญญาณอินพุตมีการเปลี่ยนแปลงอย่างรวดเร็ว วงจรผลิตสัญญาณแรมป์อ้างอิงได้ปรับปรุงขึ้นโดยตัดเอาผลกระทบของการเลื่อนไหล เมื่อใช้วงจรไปนาน ๆ ดังรูปที่ 2-42ก แสดงบล็อกไดอะแกรม

ของวงจร ADC แบบสไลบคู้ซึ่งวงจรส่วนใหญ่คล้ายกับแบบสไลบเดียวแต่มีสวิตช์ที่อินพุตเพิ่มขึ้นเพื่อทำการเลือกกระหว่างแรงดันอินพุตกับแรงดันอ้างอิง



รูป 2-42 วงจร A/D Converter แบบสไลบคู้

(ก) บล็อกไดอะแกรม

(ข) เอาท์พุทของวงจรอินทิเกรเตอร์เมื่อเทียบกับเวลา

ส่วนแรกของวงจร คือ วงจรกำเนิดสัญญาณแรมป์ ซึ่งใช้วงจรอินทิเกรเตอร์ที่อินเวอร์ตดิ้ง อินพุตของออปแอมป์มีสภาพเป็นกราวด์เสมือนถ้ามีแรงดันอินพุต 2 โวลต์ จะได้กระแสไหลผ่านตัวต้านทาน 10k เท่ากับ 0.2mA ไปยังจุดรวม (Summing point) เนื่องจากค่าความต้านทานอินพุตของ ออปแอมป์สูงมากกระแสที่ไหลจึงเกิดขึ้นผ่านตัวเก็บประจุขณะที่ตัวเก็บประจุทำการชาร์จ แรงดันที่เอาต์พุตของออปแอมป์ก็จะเป็นลบมากขึ้น เพื่อรักษาระดับกระแสให้คงที่ แรงดันคร่อมตัวเก็บประจุจึงเป็นสัญญาณแรมป์เชิงเส้น โดยถ้าแรงดันอินพุตเป็นบวก วงจรอินทิเกรเตอร์ จะให้เอาต์พุตเป็นสัญญาณแรมป์ทางลบ ดังแสดงไว้ในช่วง t_1 ดังรูปที่ 2-42x หากแรงดันอินพุตเป็นลบจะให้เอาต์พุตเป็นบวก ความชันของสัญญาณแรมป์ สามารถคำนวณได้จากความสัมพันธ์ของประจุ $q = cv$ และ $q = It$ โดยจับ 2 สมการเท่ากับได้

$$V/t = I/C$$

และ $I = V_{in} / R$

ดังนั้น $V/t = V_{in} / RC$

จากรูป 2-42ก ให้แรงดันอินพุตเป็น 2 โวลต์ จะได้ความชันของสัญญาณแรมป์ทางเอาต์พุตเท่ากับ -2 โวลต์ต่อมิลลิวินาที อธิบายวงจรได้คือ เมื่อสวิตช์ต่อกับสัญญาณอินพุตจะทำให้มีแรงดันบวกจากอินพุตป้อนเข้าสู่วงจรอินทิเกรเตอร์ ได้เอาต์พุตออกมาเป็นแรมป์ทางลบ วงจรเปรียบเทียบกับก็ได้แรงดันลบจากวงจรอินทิเกรเตอร์แล้วให้เอาต์พุตเป็นบวกทำการเปิดแอนด์เกตให้สัญญาณนาฬิกาผ่านเข้าไปสู่วงจรนับ วงจรนับจะนับไปยังค่าที่กำหนดไว้คงที่ (t_1) แล้วทำการสับสวิตช์ต่อเข้ากับแรงดันอ้างอิงในช่วงที่วงจรนับด้วยค่าคงที่นั้น วงจรอินทิเกรตจะให้สัญญาณแรมป์ทางลบที่มีค่าได้สูงสุดตามแต่ละระดับแรงดันอินพุต เมื่อทำการสวิตช์อินพุตของวงจรอินทิเกรเตอร์ให้ไปที่แรงดันอ้างอิงค่าลบ เอาต์พุตของวงจรจึงได้เป็นแรมป์ทางบวกคือช่วง t_2 รูปที่ 2-42x พร้อม ๆ กับรีเซ็ตค่าของวงจรนับรวมเป็น 0 เมื่อเริ่มนับใหม่

เมื่อเอาต์พุตของวงจรอินทิเกรเตอร์เพิ่มขึ้นถึงแรงดัน 0 อีกครั้ง เอาต์พุตของวงจรเปรียบเทียบกับก็จะเป็นลบหรือเป็น 0 วงจรควบคุมจับการเปลี่ยนแปลงอันนี้ได้ ก็ส่งสัญญาณสไตรปให้วงจรนับ เก็บที่ได้ไว้ในวงจรแลทช์ จากนั้นจึงรีเซ็ตให้เป็นศูนย์แล้วทำการสวิตช์ให้อินพุตของวงจรอินทิเกรเตอร์ต่อกับแรงดันอินพุต เป็นการเริ่มการแปลงสัญญาณอีกรอบหนึ่ง จำนวนที่นับได้ที่เก็บไว้

ในวงจรแลทซ์ก็จะเป็นสัดส่วนโดยตรงกับแรงดันอินพุตสัญญาณแรมป์ทางเอาต์พุตของวงจรอินทิเกรเตอร์ในช่วงเวลาคงที่ t_1 จะลดลงสู่ระดับ V ซึ่ง

$$V = (V_{in} * t_1) / RC$$

เพื่อให้กลับสู่ระดับ 0 หน้าที่ของวงจรอินทิเกรเตอร์ จึงต้องสร้างแรมป์ทั้งทางบวกและลบให้เพิ่มขึ้นเท่า ๆ กันในช่วงเวลา t (ที่เกิดแรงดันอินพุตอ้างอิง) ได้

$$V = (V_{ref} * t_2) / RC$$

สูตรทั้งสองของ V สามารถจับนำมาเท่ากันได้

$$(V_{in} * t_1) / RC = (V_{ref} * t_2) / RC$$

$$V_{in} * t_1 = V_{ref} * t_2$$

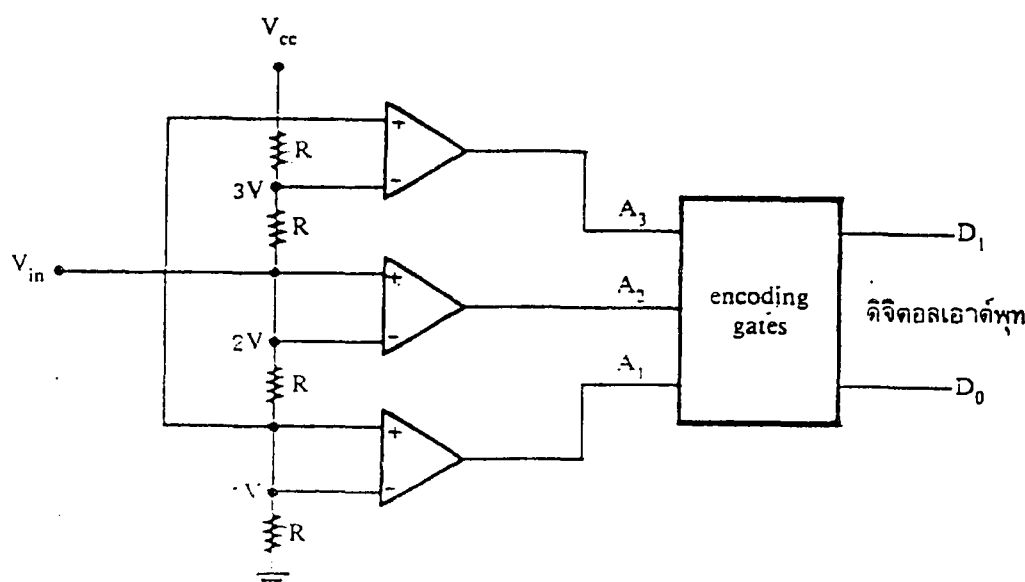
$$t_2 = t_1 * (V_{in} / V_{ref})$$

จะเห็นได้ว่า ถ้าใช้ค่าความต้านทานและค่าเก็บประจุในช่วงเวลาการอินทิเกรตสัญญาณ และช่วงเวลาอินทิเกรตอ้างอิง การเปลี่ยนแปลงของทั้งสองค่านี้ก็จะไม่มีผลต่อความถูกต้องของสัญญาณเอาต์พุต ซึ่งเป็นข้อดีที่เหนือกว่าแบบสโลปเดียว คือ ค่าที่ได้ไม่ขึ้นกับความถี่ของรอบการทำงาน ส่วนข้อเสีย คือ คาบเวลาที่ขยายออกไปที่ต้องการใช้ในการแปลงนั้นต้องใช้เวลามากกว่า 100 ms.

3. แบบขนานหรือแฟลช (Parallel or Flash)

วงจร ADC ชนิดนี้ใช้หลักการง่าย ๆ แต่เป็นวิธีที่เร็วที่สุด โดยใช้วงจรเปรียบเทียบขนานกันดังรูปที่ 2-43 ตัวต้านทานที่ต่ออนุกรมกันจะอยู่ในรูปวงจรแบ่งแรงดันที่ตกคร่อมตัวเปรียบเทียบแต่ละตัว แรงดันอินพุตสูงสุดจะขึ้นอยู่กับค่าของ V_{cc} สัญญาณเอาต์พุตจากตัวเปรียบเทียบแต่ละตัวจะ

เป็น "1" หรือ "0" จำนวนของวงจรเปรียบเทียบที่ต้องใช้ในวงจรขึ้นอยู่กับขนาดของสัญญาณอนาล็อกที่อินพุตและความละเอียด โดยสามารถหาจำนวนวงจรเปรียบเทียบได้จาก $2^N - 1$ เมื่อ N แทนจำนวนบิตหรือความละเอียดที่ต้องการ เทคนิคนี้มีข้อเสีย คือ ยังต้องการค่าความต้องใช้วงจรเปรียบเทียบสูงตาม (ความละเอียด 8 บิต ใช้วงจรเปรียบเทียบถึง 255 ตัว) นอกจากนี้เอาต์พุตก็ไม่ใช่เลขฐานสองจำเป็นต้องมีวงจรเพิ่มเติมไปทำการเข้ารหัส ส่วนข้อดี คือ มีความเร็วสูงมากในระดับนาโนวินาที



รูปที่ 2.43 วงจร A/D แบบขนาน

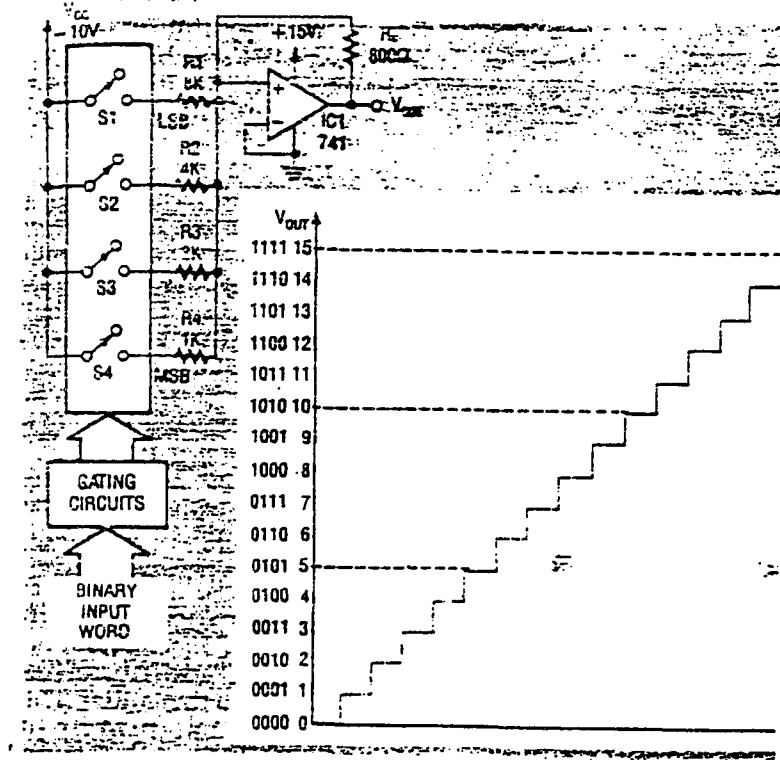
2.5.2 การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก

ตัวแปลงดิจิตอลเป็นอนาล็อกเป็นตัวแปลงรหัสเลขฐานสองจากคอมพิวเตอร์หรือวงจรดิจิตอลใดๆ ให้กลายเป็นระดับแรงดันอนาล็อกที่มีความสัมพันธ์กับระบบเลขฐานสอง โดยในการ

ทดลองทางวิทยาศาสตร์ จะใช้สัญญาณอนาล็อกที่ได้ไปควบคุมอุปกรณ์เครื่องมือต่างๆซึ่งหลักการใหญ่อยู่ 2 วิธี คือ

1. แบบใช้ตัวต้านทานหลายค่า (Binary - weighed resistor D/A)

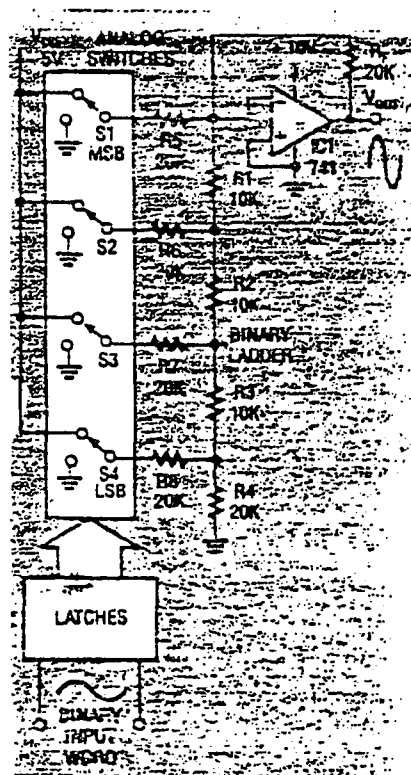
เป็นเทคนิคจัดน้ำหนักของรหัสไบนารี โดยใช้ตัวต้านทานค่า ต่าง ๆ และออปแอมป์เพื่อจัดระดับสัญญาณลอจิก 2 ระดับให้เป็นแรงดันที่ได้สัดส่วนกัน รูปที่ 2-44 แสดงวงจรเปลี่ยนสัญญาณดิจิตอลขนาด 4 บิต ออปแอมป์ที่ใช้มีอัตราขยายสูงมาก (มากกว่า 1000 เท่า) มีอินพุตอิมพีแดนซ์สูง เอาท์พุตอิมพีแดนซ์ต่ำมาก และสิ่งที่ควรพิจารณาก็คือ สัญญาณที่เอาท์พุตย้อนกลับมายังอินพุตแบบกลับเฟสเพื่อเปรียบเทียบกับสัญญาณที่ขาอินพุตอีกข้าง เอาท์พุตออปแอมป์จะเป็นตัวจ่ายหรือรับกระแส เพื่อให้แรงดันที่เปรียบเทียบนั้นมีค่าเดียวกัน ถึงแม้ว่าวิธีนี้จะมีลักษณะวงจรที่ง่าย ตรงไปตรงมา แต่ไม่สะดวกในการนำไปใช้งาน ถ้าต้องการความละเอียดของ DAC มากกว่า 4 บิต เพราะค่าของตัวต้านทานที่ใช้มากมายหลายค่าเกินไป



รูป 2-44 วงจร DAC แบบ Binary Weighted

2. แบบใช้ตัวต้านทาน 2 ค่า (R / 2R Ladder Network D/A)

เทคนิคนี้สามารถสร้างแรงดันตามน้ำหนักของรหัสไบนารี โดยอาศัยความต้านทานเพียง 2 ค่าที่จัดในลักษณะวงจรแบ่งแรงดัน หรือเรียกว่า binary ladder ดังแสดงในรูป 2-45 ซึ่งกระแสจะถูกเปลี่ยนเป็นค่าแรงดันด้วยออปแอมป์ และตัวต้านทานป้อนกลับแบบกลับเฟสลบ แม้ว่า DAC ชนิดนี้จะวิเคราะห์ยากกว่าแบบตัวต้านทานหลายค่าแต่สามารถออกแบบได้ง่ายเนื่องจากใช้ตัวต้านทานเพียง 2 ค่า เท่านั้น จึงสามารถเพิ่มจำนวนบิตได้โดยเพิ่มส่วนของ R / 2R ลงไป



รูป 2-45 วงจร DAC แบบ Ladder

2.5.3 เทคนิคเดลต้ามอดูเลชัน และ CVSD

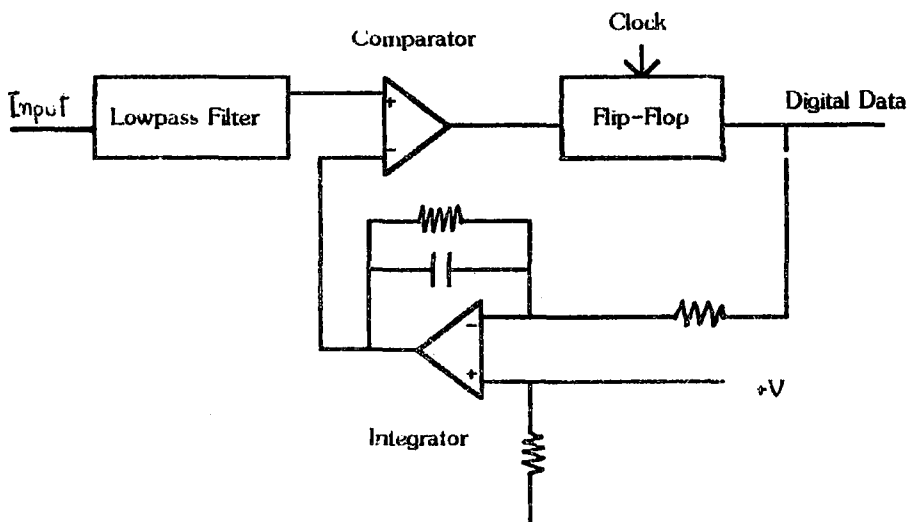
เทคนิค CVSD (Continuously Variable Slope Delta Modulation) เป็นการนำเอาสัญญาณอนาลอกมาเข้ารหัสให้เป็นสัญญาณดิจิทัลโดยมีประสิทธิภาพ เป็นรูปแบบการเข้ารหัสที่สามารถนำประโยชน์ในการส่งเคราะห์เสียงได้ดี

1. เดลต้ามอดูเลชัน

เทคนิคของเดลต้ามอดูเลชัน จะไม่ใช้การสุ่มสัญญาณหนึ่งจุด แล้วแปลงเป็นข้อมูลดิจิทัลหนึ่งเวิร์ดที่มีความละเอียดเป็นจำนวนบิตที่ต้องการ (ซึ่งเป็นหลักการของ PCM) แต่จะใช้วิธีเปรียบเทียบความสูงหรือการเปลี่ยนแปลงของสัญญาณเสียงแทน

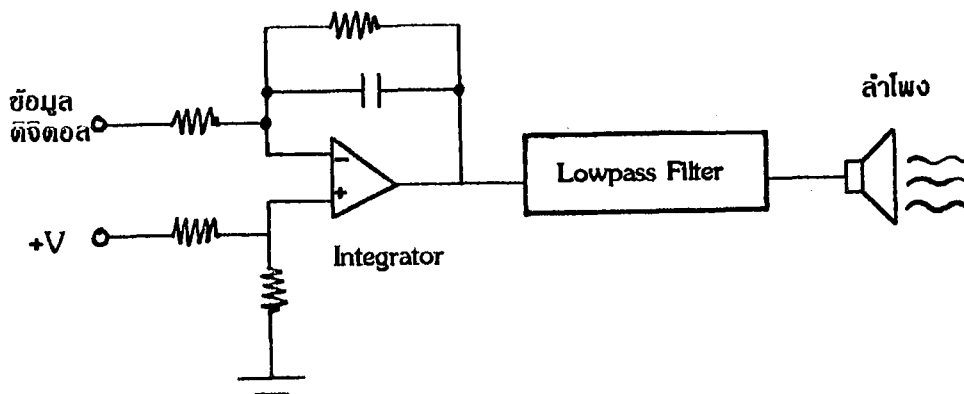
ข้อมูลที่ได้ก็คือทิศทางของการเปลี่ยนแปลง ซึ่งก็มีเพียงขึ้น หรือ ลง เท่านั้น ดังนั้นความกว้างของข้อมูลดิจิทัลจึงใช้เพียงบิตเดียวก็เพียงพอ ข้อดีของวิธีเดลต้ามอดูเลชันคือ ใช้หน่วยความจำน้อยกว่าวิธีการแบบอื่นๆ

รูปที่ 2-46 เป็นวงจรเบื้องต้นของเดลต้ามอดูเลชัน comparator จะทำหน้าที่เปรียบเทียบสัญญาณอินพุตปัจจุบันกับสัญญาณอินพุตก่อนหน้าซึ่งได้จากการป้อนกลับมายังอินทิเกรเตอร์ เอาท์พุทจากการเปรียบเทียบถูกป้อนผ่านฟลิปฟลอปที่ควบคุมด้วยสัญญาณ clock เพื่อให้ได้เป็นข้อมูลดิจิทัล ซึ่งก็คือการกำหนดอัตราการสุ่มสัญญาณนั่นเอง

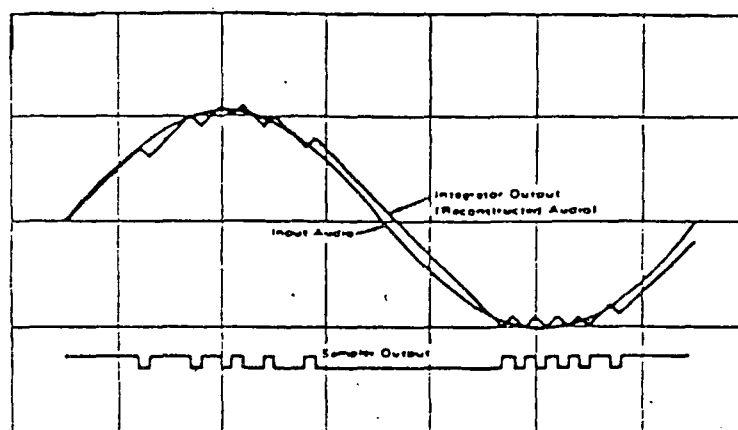


รูปที่ 2-46 แสดงวงจรเบื้องต้นของเดลต้ามอดูเลชันในส่วนแปลงสัญญาณเสียงเป็นดิจิทัล

สัญญาณที่ได้จากตัวเปรียบเทียบและจากการอินทิเกรเตอร์เปรียบเทียบกับสัญญาณอินพุต แสดงดังในรูปที่ 2-48 ลักษณะเช่นนี้จะพบว่า ยิ่งความถี่ของสัญญาณ clock มีค่าสูงก็ยิ่งสามารถบันทึกการเปลี่ยนแปลงที่แคบได้มากขึ้น ทำให้ได้คุณภาพเสียงที่ดีขึ้น แต่ก็สิ้นเปลืองหน่วยความจำมากขึ้นตามไปด้วย ความถี่เท่าใดจึงจะเพียงพอคงต้องใช้การทดลองโดยการนำเอาที่พูดสุดท้ายที่เป็นข้อมูลผ่านวงจรแปลงกลับในรูปที่ 2-47 แล้วฟังเสียงที่ได้หากฟังเป็นภาษามนุษย์รู้เรื่องก็ใช้ค่านั้น สำหรับเสียงพูดเกือบเท่าเสียงจากโทรศัพท์ซึ่งมีแถบความกว้างประมาณ 4 kHz ก็ใช้เพียง 16 kHz แต่ที่ความถี่ต่ำถึง 9.6 kHz ก็ยังพอฟังรู้เรื่อง ความถี่นี้จะเป็นตัวกำหนดอัตราข้อมูล (bit rate) ซึ่งที่ 16 kHz ก็เท่ากับ 1600 บิตต่อวินาที



รูปที่ 2-47 แสดงวงจรที่ใช้แปลงกลับจากข้อมูลดิจิตอลเป็นสัญญาณเสียง



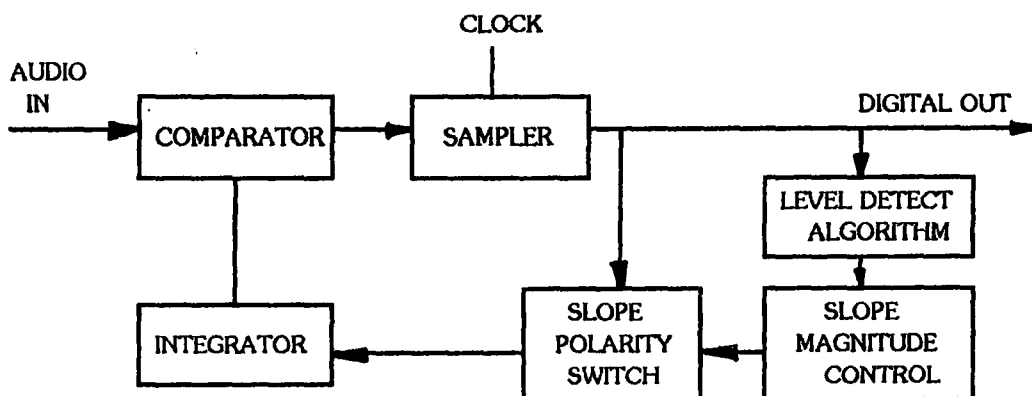
รูปที่ 2-48 แสดงการเปรียบเทียบสัญญาณอินพุตกับข้อมูลที่ได้สัญญาณอนาลอกจากอินทิเกรเตอร์

2. CVSD

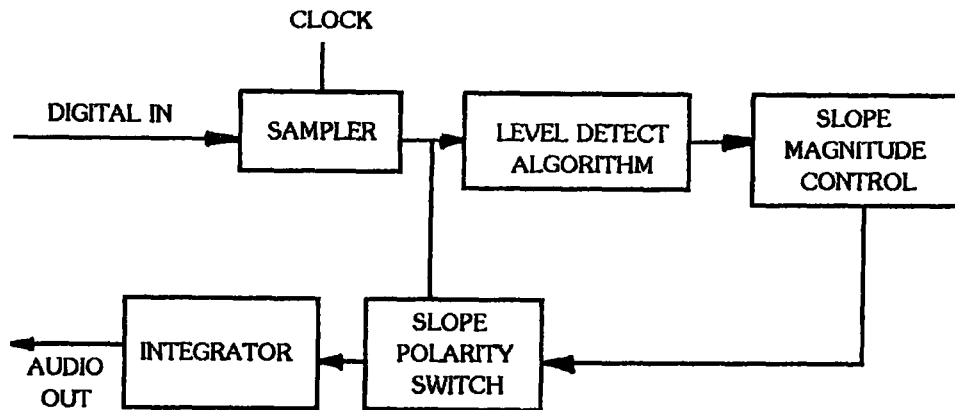
ข้อจำกัดของวิธีการเดลต้ามอดูเลชันก็คือ Bandwidth ที่ใช้งาน ซึ่งถูกจำกัดโดยความถี่ของ clock และจะสูงกว่าความถี่สูงสุดของสัญญาณอินพุตมากกว่า 2 เท่าขึ้นไป อีกอันหนึ่งคือ ความเร็วของการเปลี่ยนแปลงความสูงของสัญญาณ หรือ dynamic range ระบบเดลต้ามอดูเลชันธรรมดาที่มีค่า dynamic range ให้กว้างโดยการควบคุมอัตราขยายของอินทิเกรเตอร์ เพื่อให้ตอบสนองต่อสัญญาณที่มีความชันมากๆ ได้ทัน ระบบนี้มีชื่อเรียกใหม่ว่า "ระบบเดลต้ามอดูเลชันแบบเปลี่ยนแปลงความชันอย่างต่อเนื่อง" หรือ CVSD (Continuous Variable Slope Delta Modulation)

ระบบ CVSD ทั้งส่วนแปลงจากอนาลอกเป็นดิจิตอล และส่วนแปลงกลับจากดิจิตอลเป็นอนาลอก แสดงในรูปที่ 2-49 และ 2-50 ตามลำดับ วิธีการของ CVSD ก็คือ มีการตรวจระดับสัญญาณ โดยอาจใช้วิธีการจัดให้มีรีจิสเตอร์สำหรับเก็บข้อมูลดิจิตอลล่าสุดจำนวน 3 ถึง 4 บิต แล้วตรวจดูว่าเป็น 0 หกตหรือเป็น 1 หกตใช่หรือไม่ ถ้าใช่ แสดงว่าขณะนี้อัตราขยายของอินทิเกรเตอร์ต่ำเกินไป ตอบสนองต่อความชันของสัญญาณไม่ทัน ก็จะทำให้การเพิ่มอัตราขยายให้สูงขึ้นเฉพาะในช่วงนั้น

ในส่วนของการแปลงกลับนั้น ก็จะต้องมีการทำงานในลักษณะเดียวกัน คือมีรีจิสเตอร์ตรวจดูข้อมูลว่าเป็น 0 หกต หรือ 1 หกตหรือไม่ แล้วจัดการควบคุมอัตราการขยายของอินทิเกรเตอร์ให้สอดคล้องกัน



รูปที่ 2-49 แผนผังการทำงานของระบบ CVSD ในส่วนแปลงสัญญาณเสียงเป็นดิจิตอล
(CVSD Encoder)



รูปที่ 2-50 แผนผังการทำงานของระบบ CVSD ในส่วนการแปลงกลับจากดิจิตอลเป็นสัญญาณเสียง
(CVSD Decoder)

ระบบการมอดแบบเดลต้านี้ ประกอบด้วยตัวเปรียบเทียบในส่วนของทางควบคุมทางเดียว (forward path) และตัวอินทิเกรเตอร์ (integrator) ในส่วนของทางควบคุมทางกลับ (feedback path) สัญญาณอินพุตของตัวเปรียบเทียบเป็นสัญญาณอนาลอกที่เข้ามาเทียบกับสัญญาณเอาต์พุตของอินทิเกรเตอร์ สัญญาณเอาต์พุตของตัวเปรียบเทียบจะให้ค่าเครื่องหมายของความแตกต่างระหว่างแรงดันอินพุตกับเอาต์พุตของอินทิเกรเตอร์ซึ่งเป็นสัญญาณดิจิตอลที่ใช้ควบคุมทิศทางของสัญญาณแรมพ์ (ramp) ในตัวอินทิเกรเตอร์

โดยปกติ ตัวเปรียบเทียบจะให้สัญญาณซิงโครนัสกับสัญญาณ clock ถ้ากระแสบิตของสัญญาณนาฬิกาถูกรับส่งแบบอนุกรมไปที่ตัวอินทิเกรเตอร์ที่อยู่ไกลออกไป สัญญาณเอาต์พุตก็จะเหมือนเดียวกับสัญญาณเอาต์พุตของอินทิเกรเตอร์ในวงรอบของการควบคุมเดม นั่นคือตัวอินทิเกรเตอร์ที่ตำแหน่งของการส่งสามารถติดตามสัญญาณเอาต์พุตและตัวรับที่อยู่ไกลออกไปก็สามารถสร้างสัญญาณอินพุตขึ้นมาใหม่โดยผ่านวงจรของความถี่ต่ำที่ภาคตัวรับเพื่อจับคู่สัญญาณออก โดยอัตราการส่งสัญญาณนาฬิกา ต้องมากกว่าแถบความถี่ที่ใช้งานอย่างน้อย 2 เท่าของความถี่สูงสุด เช่น แถบความถี่เสียง คือ 4 kHz ควรใช้อัตราการส่งสัญญาณนาฬิกาที่จะนำไปชักค่าตัวอย่าง (Sampling) มีค่ามากกว่า 8 k ขึ้นไปจึงจะทำให้ การมอดแบบเดลต้า สามารถนำไปผ่านกระบวนการแปลงสัญญาณดิจิตอล (Digitizes) และส่ง (Transmit) สัญญาณ อนาลอกอินพุตไปที่ตัวรับที่อยู่ไกลๆได้ ขณะไม่มีสัญญาณอินพุตที่ตัวส่งจะมีสัญญาณเปลี่ยนแปลงทั้งขึ้น-ลง(1-0)อย่างต่อเนื่องถูกส่งออกไปแทนที่ระหว่างตัวอินทิเกรเตอร์ทั้งสองเกิดการรั่วไหลแล้ว เอาต์พุตของตัวรับจะลดลงเป็นศูนย์ และจะทำการรับใหม่เมื่อต้องการ

2.6 ระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ (DVMB)

เมื่อศึกษาลักษณะสัญญาณของระบบโทรศัพท์ในทุกๆสภาวะรวมทั้งระดับสัญญาณต่างๆในระบบ PABX (Private Automatic Branch Exchange) เพื่อนำไปสู่การพัฒนาระบบบันทึกข้อความทางโทรศัพท์อัตโนมัติโดยสามารถนำมาเชื่อมต่อกับระบบ PABX ได้และสอดคล้องกับการใช้งานในสำนักงาน หน่วยงานราชการ รัฐวิสาหกิจ ซึ่งในปัจจุบันมีการใช้ระบบ PABX กันอย่างกว้างขวาง

ระดับสัญญาณที่สภาวะต่าง ๆ สามารถนำมาดัดแปลงประยุกต์ในด้านการควบคุมระบบให้ทำงานตามความต้องการ โดยการออกแบบวงจรโดยการนำความรู้ในเรื่องลักษณะและระดับสัญญาณของโทรศัพท์หรือ PABX เป็นพื้นฐานในการออกแบบทางด้านฮาร์ดแวร์

สำหรับระบบรับฝากข้อความทางโทรศัพท์อัตโนมัตินี้ งานที่ต้องเกี่ยวข้องในการออกแบบวงจรทดลองระบบและการนำไปใช้งานจริงนั้น แบ่งออกได้ดังต่อไปนี้

1. ส่วนฮาร์ดแวร์(Hardware) เป็นส่วนของวงจรทั้งหมด ประกอบด้วย

- วงจรเชื่อมต่อกับโทรศัพท์จะทำการตรวจจับสัญญาณโทรศัพท์ (Detector)และ วงจรถอดรหัสความถี่ (DTMF)

- วงจรที่เกี่ยวข้องกับสัญญาณเสียง(กรองความถี่และกำจัดสัญญาณรบกวน) และส่วนของวงจรที่ทำการแปลงสัญญาณอนาลอกเป็นดิจิตอลและแปลงสัญญาณดิจิตอลเป็นอนาลอก

- วงจรอินเทอร์เฟซ เป็นส่วนที่เชื่อมต่อสัญญาณที่ตรวจจับมาแล้ว สัญญาณเสียงที่ทำการแปลงเป็นสัญญาณดิจิตอลเรียบร้อยแล้ว ส่งผ่านในลักษณะการสื่อสารแบบสล็อตเพื่อเชื่อมต่อกับคอมพิวเตอร์

2. ส่วนโปรแกรม(Software) ภาษาที่ใช้ในการเขียนโปรแกรมควบคุมระบบ คือ ภาษาซี (C Language) แบ่งเป็น 2 ลักษณะตามหน้าที่ คือ

- โปรแกรมควบคุมระบบการทำงานของระบบจะสามารถควบคุม โดยการเขียนโปรแกรมสั่งงานได้โดยตรง โดยรับสัญญาณต่าง ๆ มาจากวงจร Detector วงจรที่เกี่ยวข้องกับสัญญาณเสียง แล้วผ่านทางวงจรอินเทอร์เฟซ จากนั้นโปรแกรมจะรับข้อมูลผ่านทางสล็อตและควบคุมการทำงานของระบบตามโปรแกรมที่เขียนไว้

- โปรแกรมแสดงข้อมูลของผู้ใช้หรือเจ้าของ Mailbox และแสดงวัน และ เวลาที่ทำการบันทึกฝากข้อความหรือมีผู้ฝากข้อความถึงเจ้าของ Mailbox และสามารถเปลี่ยนแปลงข้อมูลส่วนตัวได้โดยการป้อนข้อมูลทางคีย์บอร์ด

สำหรับรายละเอียดของการออกแบบและการทำงานของวงจรต่างๆและการเขียนโปรแกรมควบคุมการทำงานของระบบจะอธิบายอย่างละเอียดในบทต่อไป

บทที่ 3

การออกแบบวงจรและการทำงานของระบบ

รายละเอียดในบทนี้เป็นการกล่าวถึงโครงสร้างของวงจรและโปรแกรมที่ใช้งานจริงทั้งหมด และแสดงบล็อกไดอะแกรมของระบบโดยรวมดังรูปที่ 3-1

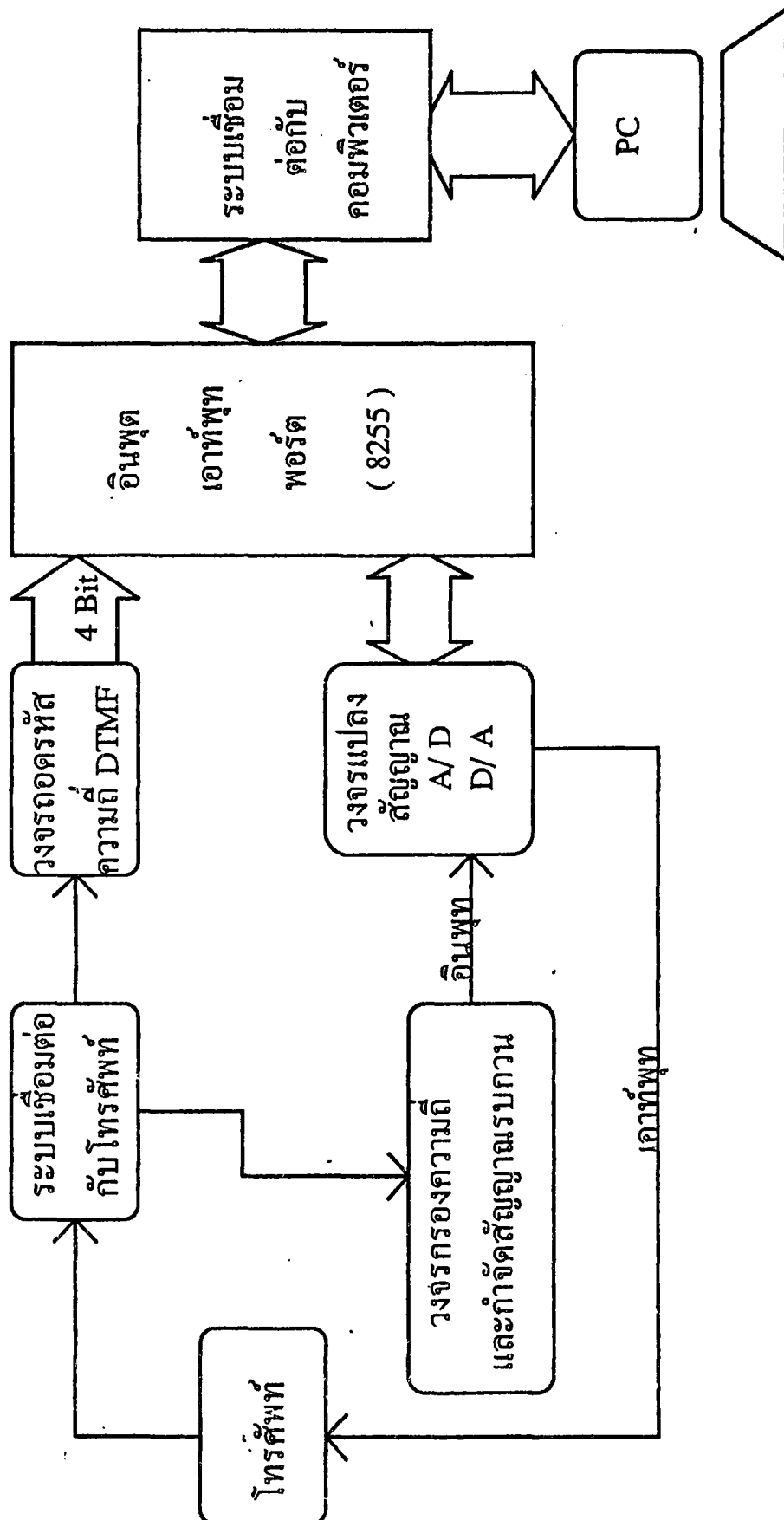
ระบบบันทึกข้อความทางโทรศัพท์อัตโนมัติสามารถแบ่งเป็นส่วนย่อยๆได้ดังนี้

- ระบบเชื่อมต่อกับโทรศัพท์
- วงจรถอดรหัสความถี่ (Dual Tone Multi Frequency, DTMF)
- วงจรกรองความถี่และขยายสัญญาณเสียง (Filter & Amplifier Audio)
- วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลและแปลงสัญญาณดิจิตอลเป็นสัญญาณ

อนาลอก (Analog to digital converter & Digital to analog converter)

- ระบบอินเทอร์เฟซกับเครื่องคอมพิวเตอร์ส่วนบุคคล

หลักการโดยรวมของระบบบันทึกข้อความทางโทรศัพท์อัตโนมัติ คือ เมื่อมีสัญญาณโทรศัพท์เข้ามาในระบบเชื่อมกับต่อโทรศัพท์จะทำการตรวจจับสัญญาณกระดิ่งแล้วทำการยกหูโทรศัพท์โดยอัตโนมัติในกรณีที่ผู้เรียกสายได้ทำการวางหูโทรศัพท์โดยอาจไม่ต้องการใช้บริการของระบบนี้จะมีวงจรที่ทำการตรวจจับการวางหูเพื่อออกจากระบบการทำงาน จากนั้นจะทำการตรวจจับสัญญาณที่มาจากการกดแป้นโทรศัพท์(สัญญาณDTMF)ซึ่งเป็นสัญญาณที่นำไปใช้ในการกำหนดฟังก์ชันการใช้งาน จากนั้นจะทำการแปลงสัญญาณอนาลอก(ซึ่งเป็นข้อมูลของเสียง) เป็นดิจิตอลและเก็บบันทึกข้อมูลลงบนหน่วยความจำภายในเครื่องคอมพิวเตอร์ การใช้งานและออกจากระบบบันทึกข้อความนี้จะมีส่วนของโปรแกรมควบคุมการทำงานทั้งหมดโดยนำสัญญาณต่างๆที่ได้ผ่านทางสล็อตของเครื่องคอมพิวเตอร์



รูปที่ 3.1 แสดงโครงสร้างของระบบฝากข้อความทางโทรศัพท์อัตโนมัติ

3.1. ระบบเชื่อมต่อกับโทรศัพท์ (Telephone Interface)

เป็นส่วนที่ทำหน้าที่ในการตรวจจับ, แปลงและนับสัญญาณกระดิ่งจากเครื่องโทรศัพท์โดยมีรายละเอียดดังต่อไปนี้ คือ

3.1.1 วงจรตรวจจับและแปลงสัญญาณกระดิ่งโทรศัพท์ (Ringing detector)

หน้าที่การทำงาน

วงจรตรวจจับสัญญาณกระดิ่งโทรศัพท์แสดงในรูปที่ 3-2

ตรวจสอบว่ามีสัญญาณกระดิ่งเข้ามาหรือไม่ โดยถ้ามีสัญญาณกระดิ่งซึ่งเป็นสัญญาณไฟสลับระดับ 100 V เรียกเข้ามาที่คู่สายโทรศัพท์ วงจรตรวจสอบสัญญาณกระดิ่งก็จะทำงานโดยจะแปลงสัญญาณกระดิ่งให้เป็นสัญญาณพัลส์ที่อยู่ในช่วง 0 - 5 V

หลักการทำงาน

เมื่อมีสัญญาณกระดิ่งซึ่งเป็นสัญญาณไฟสลับขนาด 110 V 25 Hz ครอบคลุมบนสัญญาณไฟตรง 48 V ผ่านส่วนแรกของวงจรตรวจสัญญาณกระดิ่งซึ่งมีตัวเก็บประจุต่ออยู่นั้น จะทำให้สัญญาณไฟตรงไม่สามารถที่จะผ่านไปได้และมีตัวต้านทานเป็นตัวจำกัดกระแสที่ป้อนเข้ามา ส่วนซีเนอร์ไดโอดสองตัวที่ต่อติดกันนั้นจะทำหน้าที่ลดแรงดันไฟฟ้าที่ครอบคลุมตัวมันให้มีค่าเท่ากับแรงดันไบอัสย้อนกลับสูงสุดของซีเนอร์ไดโอดเอง สัญญาณที่ได้ออกมาจะมีทั้งซีกบวกและซีกลบ แรงดันไฟฟ้าที่ได้จะต่อเข้ากับขา 1 และ 2 ของ ออปโต-คัปเปิลโดยมีไดโอดต่อครอบคลุมแบบย้อนกลับไว้กับขา 1 และ 2 ส่วนเอาต์พุตของออปโตคัปเปิลนั้นจะต่อครอบคลุมเข้ากับวงจรดังรูปที่ 3-2 ในกรณีที่ไม่มีสัญญาณกระดิ่งเรียกเข้ามาออปโตคัปเปิลจะยังไม่ทำงาน และที่ด้านเอาต์พุตของวงจรจะมีค่าเป็นสัญญาณไฟตรงเท่ากับ 5 V ถ้ามีสัญญาณกระดิ่งเรียกเข้ามา จะทำให้มีแรงดันตกคร่อมที่ออปโต คัปเปิลด้านอินพุตออปโตคัปเปิลจะทำงานโดยขา 4 และ 5 จะต่อเชื่อมกันได้เอาต์พุตของวงจรมีแรงดันเป็น 0 V โดยมีตัวต้านทานกับตัวเก็บประจุเป็นวงจรกรองทางด้านเอาต์พุต จากนั้นจะนำสัญญาณเข้าที่ขา 1 ของ ซีมิคอนดักเตอร์ทำให้เวลาที่สัญญาณกระดิ่งเข้ามาช่วงที่มีเสียงกระดิ่งดังจะได้สัญญาณเอาต์พุตของวงจรตรวจสอบสัญญาณกระดิ่งเป็นลอจิก "ต่ำ" ส่วนในกรณีที่ไม่มีสัญญาณกระดิ่งที่ด้านเอาต์พุตก็จะได้สัญญาณเป็นลอจิก "สูง" ดังนั้นเมื่อมีสัญญาณกระดิ่งเข้ามาสัญญาณที่

ด้านเอาต์พุตของวงจรตรวจสอบสัญญาณกระดิ่งก็จะมีลักษณะเป็นสัญญาณพัลส์ตามลักษณะของ ช่วงเสียงกระดิ่งที่ดังขึ้น

3.1.2 วงจรนับสัญญาณกระดิ่ง

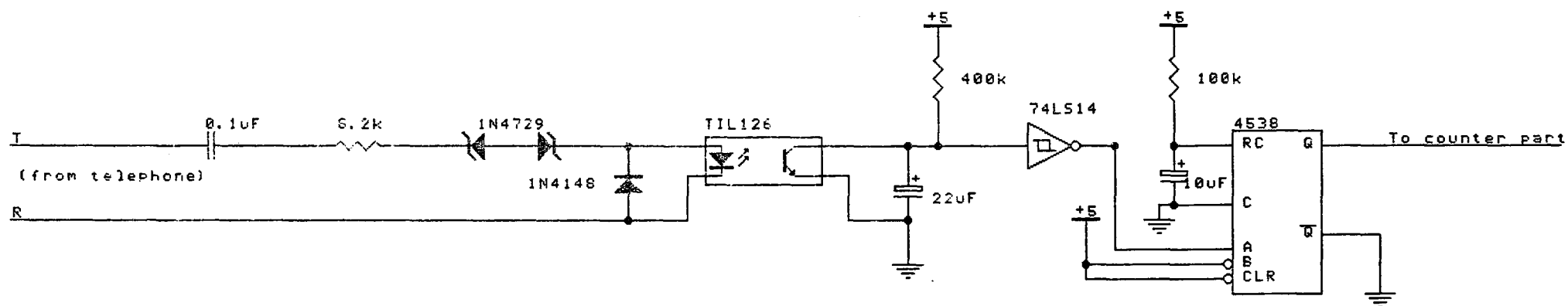
หน้าที่การทำงาน

วงจรนับสัญญาณกระดิ่งแสดงดังรูปที่ 3-3

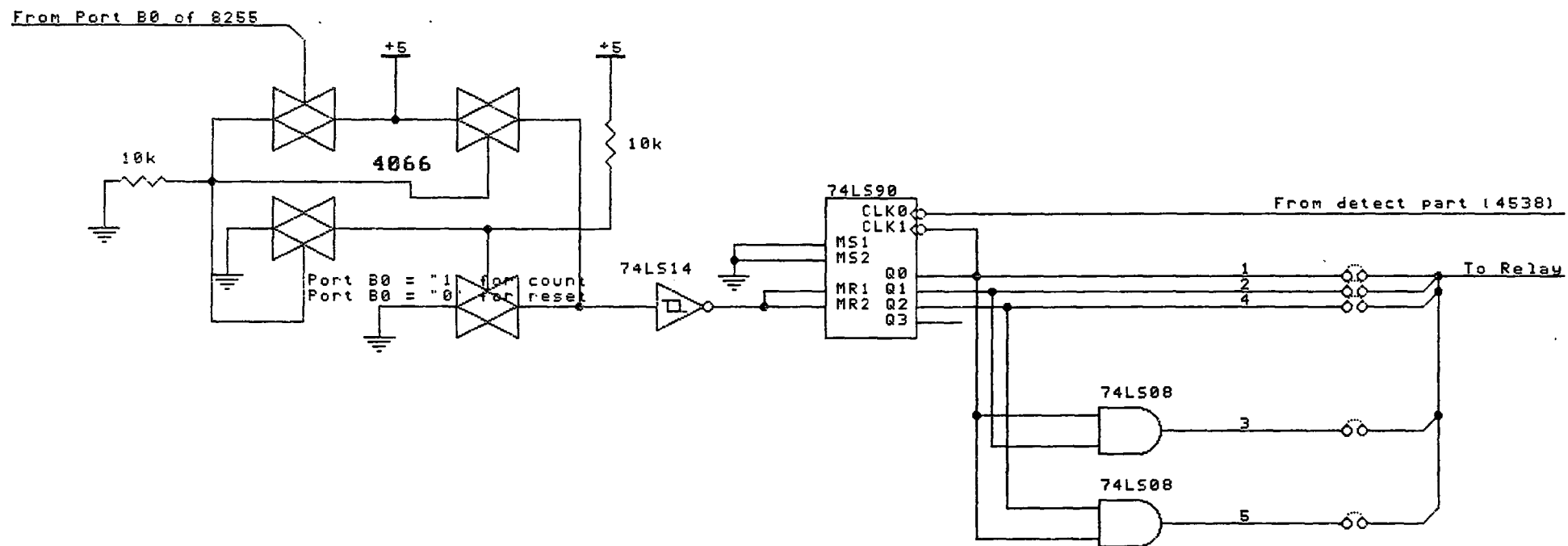
ทำการนับสัญญาณกระดิ่งโดยสามารถตั้งจำนวนครั้งที่ให้กระดิ่งดังได้ และส่งลอจิกทางเอาต์พุตเพื่อไปขับให้รีเลย์ 1 ทำงานต่อไป

หลักการทำงาน

นำสัญญาณทางด้านเอาต์พุตของวงจรแปลงสัญญาณกระดิ่งซึ่งมีลักษณะเป็นสัญญาณพัลส์ตามจังหวะที่กระดิ่งดังขึ้น โดยจะมีจำนวนพัลส์หลายลูกในแต่ละครั้งจึงจำเป็นต้องทำการหน่วงให้ได้พัลส์ 1 ลูกต่อเสียงกระดิ่งดัง 1 ครั้งเช่นกัน ซึ่งจะใช้วงจรโมโนสเตเบิลมัลติไวเบรเตอร์โดยช่วงเวลาทำการหน่วงจะนำไปใช้ในการกำหนดค่า R และ C ในวงจร สำหรับไอซีที่ใช้คือ MC 14538 แบบปริทริกเอเบิล สัญญาณพัลส์ที่ได้ทางเอาต์พุตจะเป็นอินพุตของวงจรรนับในที่นี้จะใช้ไอซีเบอร์ 74LS90 เป็นตัวนับสัญญาณพัลส์ที่เข้ามา ทางด้านเอาต์พุตของวงจรรนับจะได้จำนวนครั้งของกระดิ่งที่ดังในรูปของเลขฐานสอง 4 บิต คือ Q_D Q_C Q_B Q_A จากนั้นจะนำค่าต่างๆนี้ไปทำการดีโค้ดโดยสามารถเลือกได้ว่าจะให้กระดิ่งโทรศัพท์ดังเป็นจำนวนกี่ครั้ง จากนั้นจึงส่งลอจิก "1" ที่ได้จากการ ดีโค้ดเข้าที่ขาเบสของทรานซิสเตอร์ เบอร์ C458 เพื่อทำการขับให้รีเลย์ 1 ทำงานได้เพื่อทำการยกหูโทรศัพท์ต่อไป



รูปที่ 3-2 วงจรตรวจจับและแปลงสัญญาณกระดิ่ง



รูปที่ 3-3 วงจรนับสัญญาณกระตุ้น

3.1.3 วงจรตัด-ต่อคู่สายโทรศัพท์

หน้าที่การทำงาน

วงจรตัด-ต่อคู่สายโทรศัพท์แสดงในรูปที่ 3-4

ใช้ในการตัด-ต่อคู่สายโทรศัพท์ที่มีค่าความต้านทานขนาด 600 โอห์ม ครอบคลุมภายในคู่สายโทรศัพท์ เสมือนกับว่าทำการยกหู-วางหูโทรศัพท์

หลักการทำงาน

วงจรตัด-ต่อคู่สายโทรศัพท์เป็นวงจรขั้วรีเลย์ให้ทำงานในลักษณะตัดต่อตัวต้านทาน 600 โอห์มครอบคลุมคู่สายโทรศัพท์ โดยรีเลย์สามารถทำงานได้โดยจะใช้ทรานซิสเตอร์เป็นตัวขับให้รีเลย์ทำงานทำให้รีเลย์ต่อครบวงจร เสมือนกับว่าเกิดการยกหูโทรศัพท์ขึ้น โดยอินพุตของทรานซิสเตอร์มาจากเอาต์พุตของวงจรนับสัญญาณกระดิ่ง ทรานซิสเตอร์จะ ON ก็ต่อเมื่อ เมื่อให้สัญญาณลอจิก "1" ที่ขาเบสของทรานซิสเตอร์เท่านั้น และเมื่อทรานซิสเตอร์เริ่มทำงานจะมีกระแสไหลจากขาคอลเลคเตอร์ไปยังขาคีมิตเตอร์ โดยที่ รีเลย์จะมีไดโอดต่อไบอัสกลับอยู่ และในกรณีที่ผู้โทรเข้าได้ทำการวางหูโทรศัพท์ไปแล้ว จากนั้นระบบจะทำการรีเซ็ตวงจรนับและส่งเอาต์พุตที่มีสถานะลอจิกเป็น "0" ให้กับทรานซิสเตอร์ทำให้ทรานซิสเตอร์ OFF ซึ่งจะเป็นผลให้รีเลย์ต่อไม่ครบวงจรเสมือนกับว่าเป็นการวางหูโทรศัพท์โดยอัตโนมัติ

3.2 วงจรถอดรหัสความถี่ (Dual Tone Multi Frequency, DTMF)

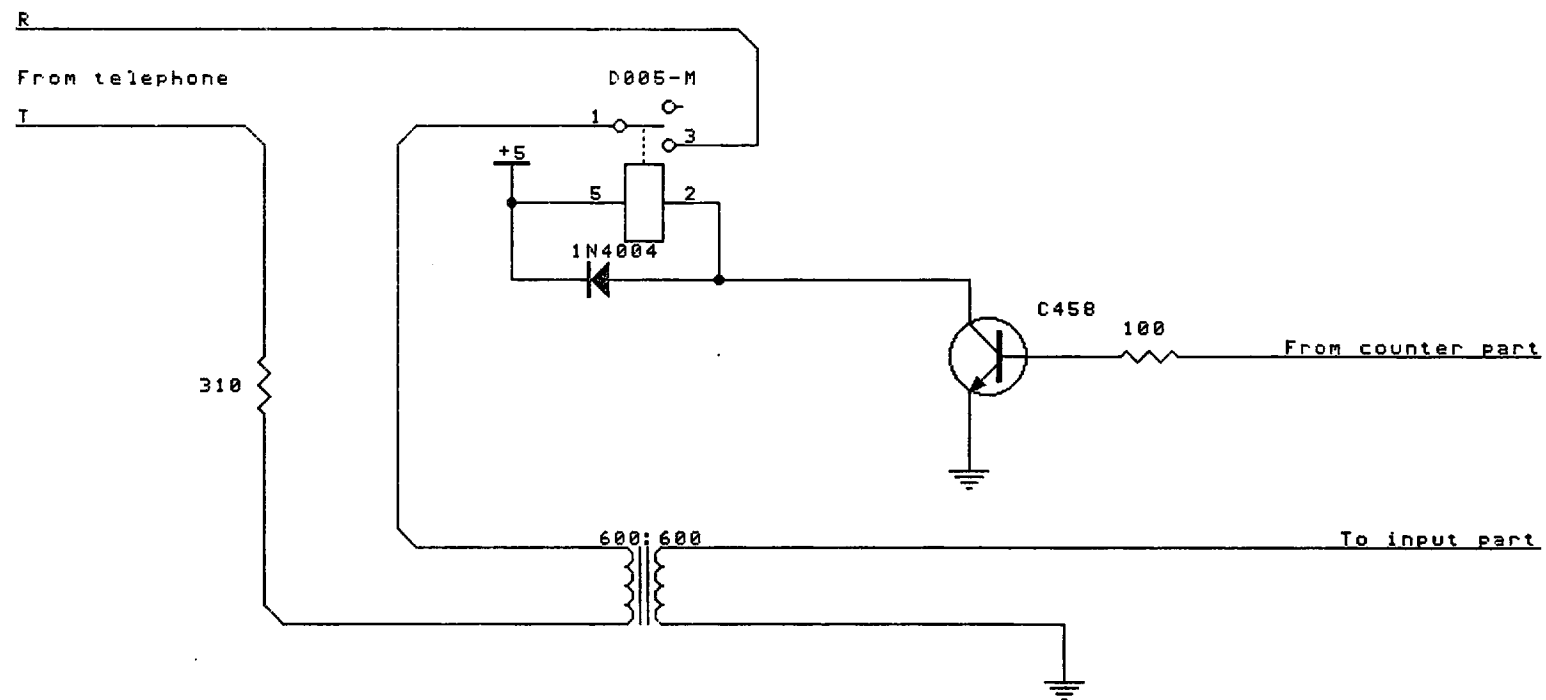
หน้าที่การทำงาน

รับสัญญาณ DTMF จากคู่สายแล้ว นำมาถอดรหัส DTMF ให้เป็นสัญญาณดิจิตอลซึ่งเป็นข้อมูล 4 บิต พร้อมสัญญาณดิจิตอลที่เป็นสัญญาณสโตรบอีก 1 บิต เพื่อใช้เป็นข้อมูลส่งให้แก่ส่วนการทำงานของอินพุตพอร์ตของ 8255

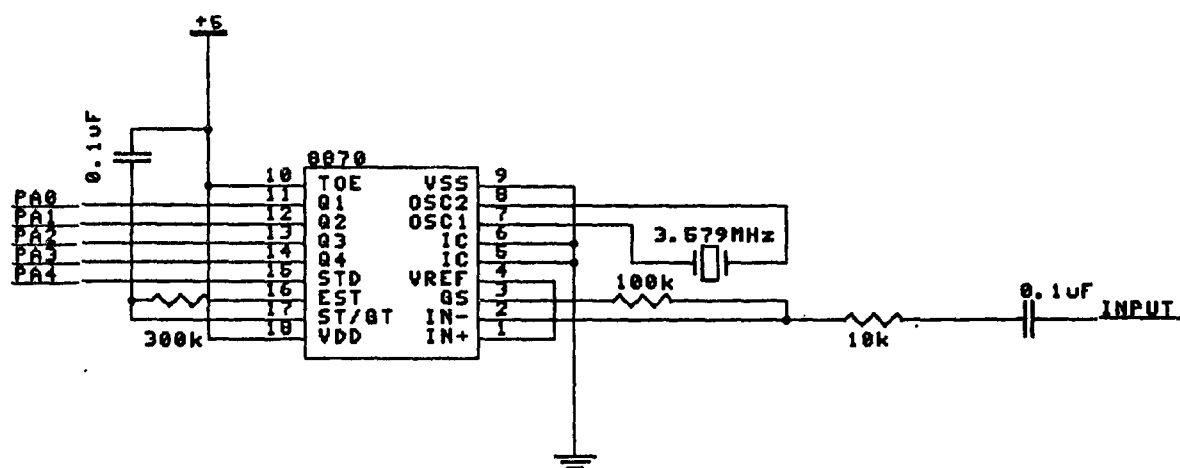
หลักการทำงาน

หลังจากวงจรนับสัญญาณกระดิ่งทำการต่อคู่สายโทรศัพท์เรียบร้อยแล้ว จะมีสัญญาณ DTMF ถูกส่งมาตามคู่สาย สัญญาณจากการกดแป้นโทรศัพท์นั้นจะเข้าที่อินพุตของไอซี MT8870 ซึ่งจะทำหน้าที่แปลงรหัส DTMF ให้เป็นสัญญาณดิจิตอลที่เป็นข้อมูล BCD 4 บิต พร้อมทั้งสัญญาณ

สโตรบอีก 1 บิท ตามจำนวนรหัส DTMF ซึ่งมีทั้งหมด 16 รหัส สัญญาณทั้งหมดนี้จะถูกส่งต่อไปยัง ส่วนการทำงานของอินพุทพอร์ตของ 8255 เพื่อเป็นข้อมูลในการประมวลผลและสั่งงานต่อไป รูปที่ 3-5 แสดงวงจรถอดรหัสความถี่ DTMF



รูปที่ 3-4 วงจรติดต่อคู่สายโทรศัพท์



รูปที่ 3-5 วงจรตรวจจับสัญญาณ DTMF

3.3 วงจรกรองความถี่และขยายสัญญาณเสียง (Filter & Amplifier Audio)

แบ่งการทำงานออกเป็น 2 ลักษณะ คือ ภาคอินพุต และ ภาคเอาต์พุต

3.3.1 ภาคอินพุต

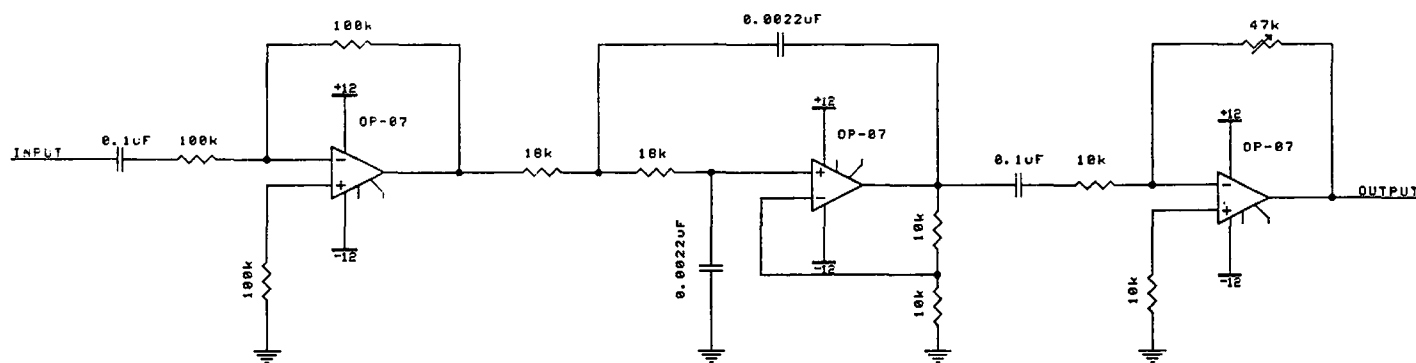
หน้าที่และการทำงาน

เมื่อวงจรตรวจนับสัญญาณกระดิ่งทำการติดต่อคู่สายโทรศัพท์แล้ว สัญญาณเสียงจากคู่สายโทรศัพท์ จะผ่านวงจรในส่วนที่เป็นบัฟเฟอร์เพื่อทำการปรับอินพุตอิมพีแดนซ์ให้เหมาะสมโดยมีอัตราขยาย (Gain) เท่ากับ 1 จากนั้นจะผ่านวงจรกรองความถี่แบบ Low Pass Filter ซึ่งเป็นวงจรที่กรองเอาความถี่สูงออกไปให้เหลือแต่เพียงความถี่ต่ำออกไปใช้งานโดยมีความถี่คัทออฟที่ 4 kHz จากนั้นจะนำสัญญาณที่ได้เข้าที่อินพุตของวงจร Audio Amplifier ที่เป็นแบบ Inverting Amplifier โดยมีอัตราขยายขึ้นกับค่าความต้านทาน R_1 และ R_2 (R_2/R_1) สำหรับการออกแบบวงจรกรองความถี่แบบ Low Pass Filter ที่ใช้ในโครงงานนี้จะเป็นแบบอันดับ 2 (Second Order) การเลือกใช้งานที่อันดับ 2 นี้ เนื่องจากเพื่อให้ได้วงจรกรองความถี่ที่ประสิทธิภาพดีขึ้นนั่นเอง แสดงวงจรในรูปที่ 3-6

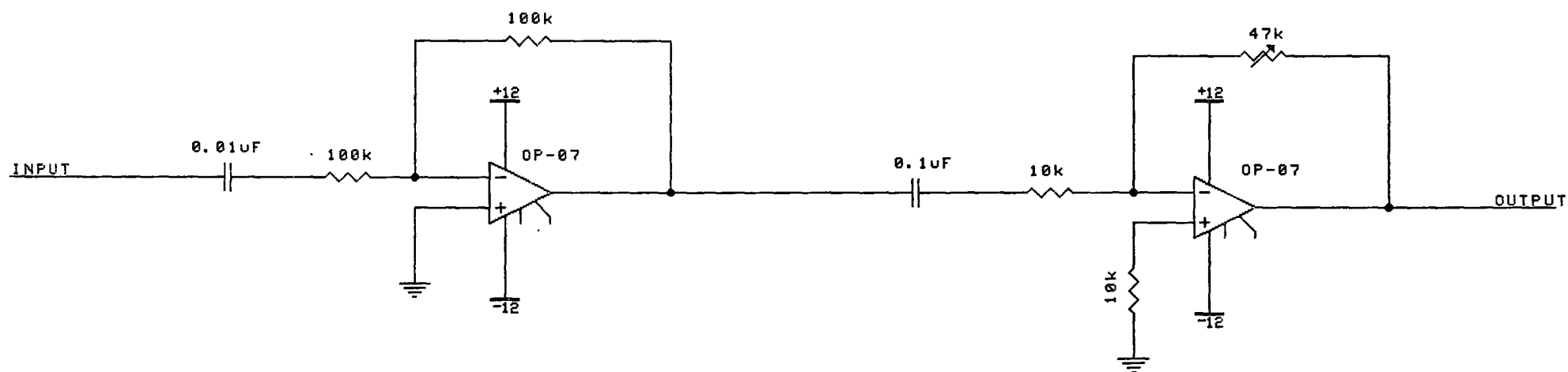
3.3.2 ภาคเอาต์พุต

หน้าที่และการทำงาน

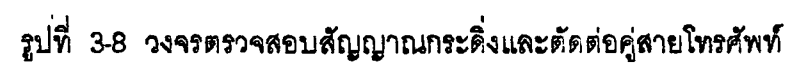
สัญญาณเสียงจากเอาต์พุตของวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอกนั้นจะผ่านวงจรส่วนที่เป็นบัฟเฟอร์ที่มีอัตราขยายเท่ากับ 1 และผ่านวงจรขยายสัญญาณเสียง (Audio Amplifier) เพื่อทำการขยายสัญญาณเสียงพูดให้ดังขึ้นโดยที่อัตราขยายสัญญาณมีค่าเท่ากับอัตราส่วน ระหว่าง R_1 และ R_2 (R_2/R_1) วงจรแสดงดังรูปที่ 3-7

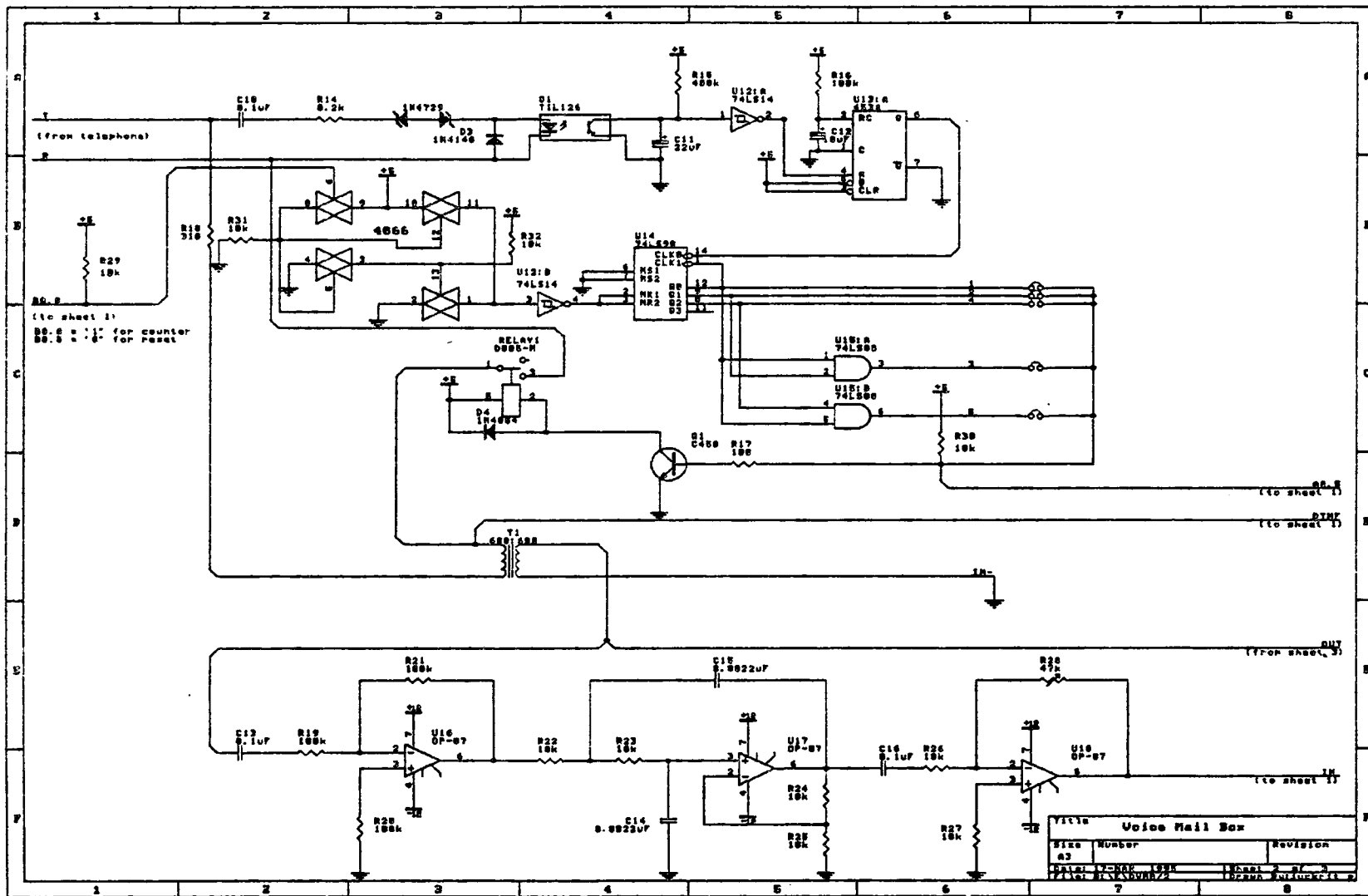


รูปที่ 3-6 วงจรกรองความถี่และขยายสัญญาณเสียงอินพุต



รูปที่ 3-7 วงจรกรองความถี่และขยายสัญญาณเสียงเอ๊าท์พุท

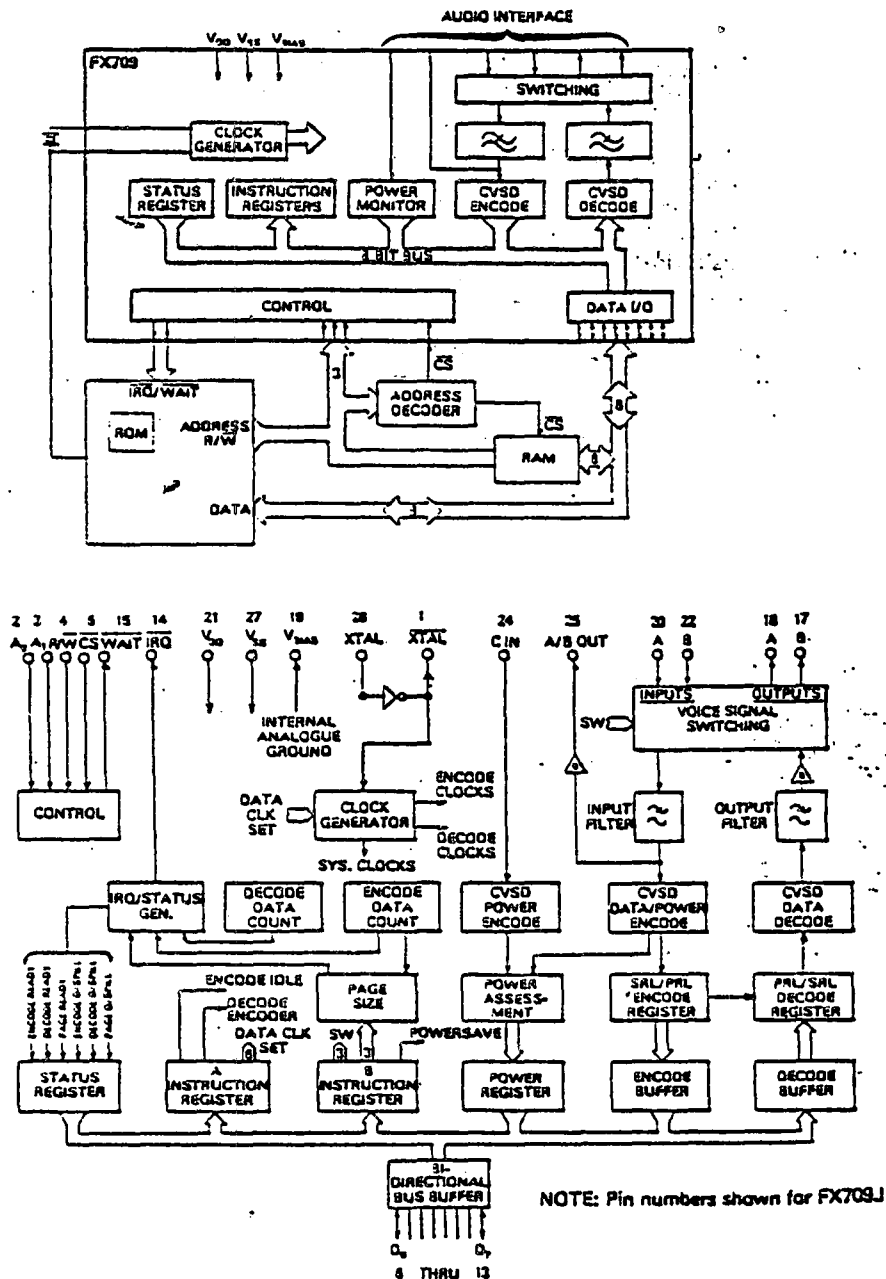




รูปที่ 3-9 แสดงการเชื่อมต่อวงจร Detector กับภาคขยายสัญญาณอินพุต

3.4 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลและแปลงดิจิตอลเป็นอนาลอก

สำหรับโครงการพิเศษนี้จะใช้ชิพสำเร็จรูปที่สามารถแปลงสัญญาณอนาลอกเป็นดิจิตอลและแปลงสัญญาณดิจิตอลเป็นอนาลอกภายในตัว โดยในที่นี้ได้เลือกใช้ **FX709 Voice Store Retrieve CVSD Codec** ซึ่งสะดวกในการใช้งานเป็นอย่างมาก โดยมีรายละเอียดของโครงสร้างและการใช้งานของ FX709 อยู่ในภาคผนวกท้ายเล่ม



รูปที่ 3-10 แสดงบล็อกไดอะแกรมระบบการทำงานของ FX 709

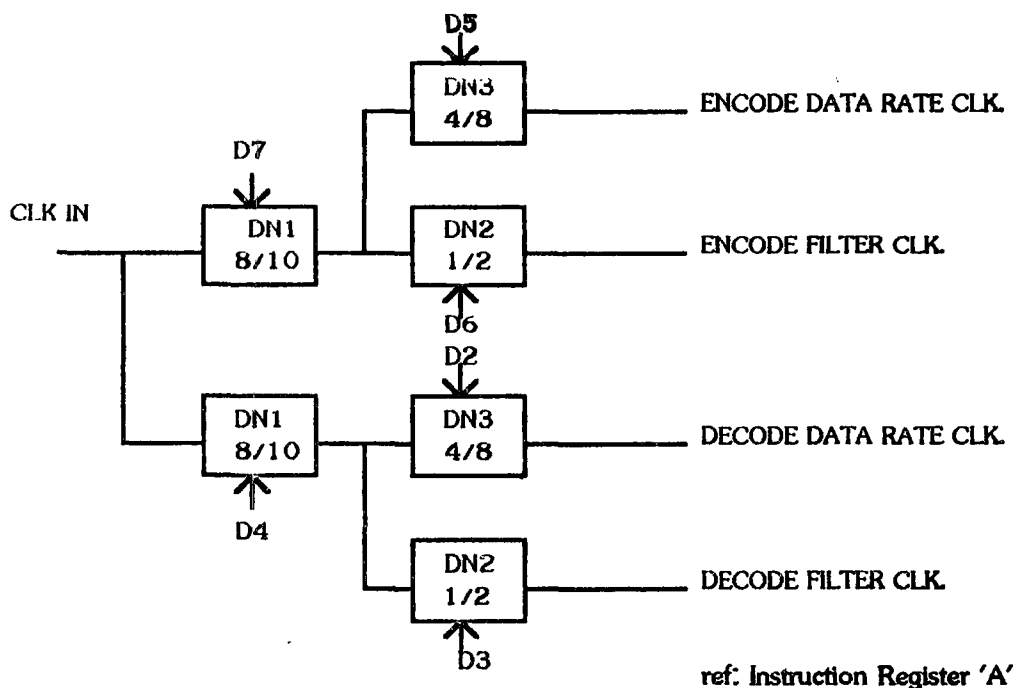
ลักษณะการทำงานอย่างคร่าวๆของ FX709 มีดังนี้

เป็นชิพที่สามารถทำการเก็บบันทึก(Record) และเรียกกลับ(Playback) สัญญาณเสียงได้โดยเมื่อทำงานอยู่ในโหมด Encode สัญญาณเสียงทางด้านอินพุตจะผ่านวงจรกรองความถี่ซึ่งอยู่ภายในตัวชิพนี้โดยที่สามารถเลือกค่าความถี่คutoffได้จากตารางที่ 3-1

CLOCK INPUT	N1	N2	FILTER CLOCK (Hz)	LOWPASS FILTER BW P8. $\pm 1dB$	N3	DATA CLOCK (kbs)
2MHz	8	2	125k	3320	4	62.5
"	8	2	125k	3320	8	31.25
"	10	2	100k	2656	4	50.0
"	10	2	100k	2656	8	25.0
1MHz	8	1	125k	3320	4	31.25
"	8	1	125k	3320	8	15.625
"	8	2	62.5k	1660	4	31.25*
"	8	2	62.5k	1660	8	15.625*
"	10	1	100k	2656	4	25.0
"	10	1	100k	2656	8	12.5
"	10	2	50k	1328	4	25.0*
"	10	2	50k	1328	8	12.5*
*Caution: Although possible, the Codec insertion loss is not according to the specification at these settings. (see Page 10).						
2.048MHz	8	2	128k	3400	4	64.0
"	8	2	128k	3400	8	32.0
"	10	2	102.4k	2720	4	51.2
"	10	2	102.4k	2720	8	25.6
1.024MHz	8	1	128k	3400	4	32.0
"	8	1	128k	3400	8	16.0
"	8	2	64k	1700	4	32.0*
"	8	2	64k	1700	8	16.0*
"	10	1	102.4k	2720	4	25.6
"	10	1	102.4k	2720	8	12.8
"	10	2	51.2k	1360	4	25.6*
"	10	2	51.2k	1360	8	12.8*
614.4kHz	8	1	76.8k	2040	8	9.6*
768.0kHz	10	1	76.8k	2040	8	9.6*

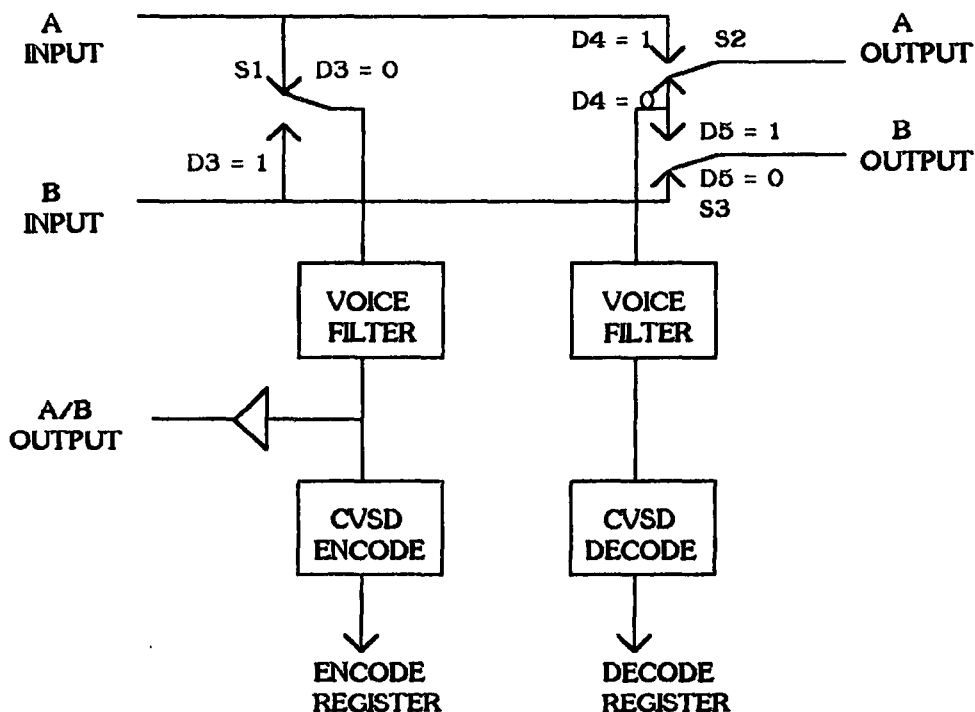
Table 1 Possible combinations of clock input frequency, filter cutoff (Hz) and Data Clock (kbs)

ตารางที่ 3-1 แสดงค่าความถี่ของ input clock , filter cutoff และ data clock



รูปที่ 3-11 แสดงบล็อกไดอะแกรมของการเลือกค่า clock input, filter clock, data clock

เมื่อสัญญาณเสียงผ่านวงจรกรองความถี่ต่ำแล้วจะถูกแปลงเป็นสัญญาณดิจิทัล 8 บิต โดย CVSD ENCODER ข้อมูลที่ได้จากการ Encode จะถูกส่งไปยัง I/O Bus เพื่อนำข้อมูลไปเก็บไว้ในหน่วยความจำภายนอกที่ทำการเชื่อมต่อ ส่วนการทำงานในโหมด Decode หน่วยความจำที่ทำการบันทึกข้อมูลไว้ จะทำการ Decode โดยทำข้อมูล 8 บิตแบบขนานผ่านทาง I/O Bus จากนั้นจะทำการแปลงข้อมูลโดย CVSD DECODER จะได้สัญญาณเสียงที่ทำการบันทึกในตอนแรกผ่านวงจรกรองความถี่ต่ำ และออกที่ขาเอาต์พุตของ FX 709 ซึ่งสามารถเลือกได้ว่าจะใช้งานอินพุต A หรือ B ได้โดยการโปรแกรมโหมดการทำงานของ FX 709 ดังรูปที่ 3-12



The input and output switching of analogue signals is controlled by the state of bit No.'s D3, D4, and D5 in Instruction Register B.

รูปที่ 3-12 แสดงบล็อกไดอะแกรมของการเลือกอินพุต เอาต์พุต A,B

3.5 ระบบเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคล

แบ่งเป็น 2 ส่วนที่สำคัญ คือ

3.5.1 อินพุท เอาท์พุทพอร์ท 8255 PIO

เป็นส่วนที่ทำการส่งและรับข้อมูล 8 บิตเพื่อนำไปตรวจสอบสภาวะการทำงานของฮาร์ดแวร์ โดยจะทำหน้าที่หลักด้วยกัน 2 อย่าง คือ

- เป็นอินพุทพอร์ท จะรับข้อมูลดิจิทัล 8 บิต จากการกดแป้นโทรศัพท์โดยสัญญาณ DTMF จะถูกแปลงเป็น BCD 4 บิต และ Std ส่งผ่านมายังอินพุทพอร์ทของ 8255

- เป็นเอาท์พุทพอร์ท จะส่งสภาวะลอจิก "1" หรือ "0" ไปควบคุมฮาร์ดแวร์

- การใช้งาน 8255

เป็นพอร์ทอินพุท-เอาท์พุทแบบขนาน (Parallel Input/Output Port) โดยกำหนดให้ทำงานในโหมดศูนย์ ซึ่งจะมีพอร์ท A เป็นอินพุทพอร์ท , พอร์ท B เป็นเอาท์พุทพอร์ท , พอร์ท C บนเป็นอินพุทพอร์ท ส่วนพอร์ท C ล่างนั้นไม่ได้ถูกใช้งาน โดย Control word ที่จะต้องส่งให้ 8255 คือ 91h

สถานะต่าง ๆ ในการควบคุมการทำงาน

- ต้องการให้อ่านข้อมูล ขา RD ต้องมีสภาวะลอจิกเป็น "0"
- ต้องการให้เขียนข้อมูล ขา WR ต้องมีสภาวะลอจิกเป็น "0"
- ขณะทำงาน ขา CS ต้องเป็น "0" และ ควรรีเซ็ต ขา RESET โดยจะแอกทีฟที่ลอจิก "1"
- การเลือกเบอร์พอร์ท ทำได้โดยการควบคุมที่ขา A0 และ A1

00 เลือกพอร์ท A

01 เลือกพอร์ท B

10 เลือกพอร์ท C

11 ให้เขียนรหัสควบคุม 8255 (มีความหมายเฉพาะขา WR แอคทีฟ)

รายละเอียดอื่น ๆ ของ 8255 PIO ดูได้จากภาคผนวกท้ายเล่ม

- การเขียนรหัสควบคุม จะต้องเขียนลงบนขา D0-D7 โดยแต่ละบิตจะมีความหมายแสดงในรูปที่ 3-13

D ₀	เลือกพอร์ต C ล่าง	
D ₁	เลือกพอร์ต B กลุ่ม B	
D ₂	เลือกโหมด	การเลือกโหมด
D ₃	เลือกพอร์ต C บน	(0)0 = โหมด 0 (1)0 = โหมด 1
D ₄	เลือกพอร์ต A	IX = โหมด 2
D ₅	เลือกโหมด	การเลือกพอร์ต
D ₆		0 = OUTPUT 1 = INPUT
D ₇	โหมดเปิดแฟล็ก '1' = แอคทีฟ	

รูปที่ 3-13 แสดงความหมายของบิตต่างๆที่ใช้ในการกำหนดรหัสควบคุม

3.5.2 การติดต่อกับสล็อตของ IBM/PC

การนำสัญญาณต่าง ๆ จากสล็อตมาใช้งานในการควบคุมและตรวจจับสัญญาณ

จากระบบ

สัญญาณจากสล็อตที่นำมาใช้งานจริง มีดังต่อไปนี้

- 1.) D0 - D7
- 2.) A0 - A9
- 3.) IRQ5
- 4.) I/O CHRDY
- 5.) IOR / IOW
- 6.) RESET
- 7.) AEN
- 8.) GND
- 9.) +5 , -5 Volt
- 10.) +12 , -12 Volt

จากรูปที่ 3-14 อธิบายการทำงานในแต่ละส่วนได้ดังนี้

ด้านซ้ายมือของรูป คือ ด้านที่จะเสียบเข้ากับสล็อตของเครื่องคอมพิวเตอร์ ของการ์ดที่เราสร้างขึ้น

- U1,U2 (74LS244) เป็นบัฟเฟอร์ทางเดียวที่ต่อขาอินพุตกับสล็อตกราวด์ ดังนั้นจึงแยกที่พอยต์ตลอดเวลา ใช้สำหรับรับสัญญาณจากขา RESET , IOR , IOW , AEN และ แอดเดรสบัส A0-A9

- U3 (74LS245) เป็นบัฟเฟอร์แบบ Tri-state ที่ทำงานได้สองทิศทาง โดยจะควบคุมทิศทางได้จากการใช้ลอจิกที่ขา G1 (DIR)

การทำงานของขาต่าง ๆ

- ขา RESET จากสล็อตจะเข้าไปที่ขา RESET ของ 8255 เพื่อใช้ Reset 8255

- ขา IOR , IOW จะถูกนำมาแนบ (NAND) กันเพื่อที่จะเป็นตัวให้ U0 (74LS138) ทำงาน เมื่อต้องการจะเขียนหรืออ่านข้อมูล และขา IOR จะส่งลอจิกออกไปที่ขา DIR ของ 74LS245 เพื่อใช้ในการกำหนดทิศทางของ 74LS245

- ขาแอดเดรส A0,A1 จะเป็นสัญญาณที่ถูกส่งไปที่ U9 (FX709) เพื่อเป็นตัวกำหนด สถานะการทำงานของ FX709 ว่าจะให้ทำงานในลักษณะใด

- ขาแอดเดรส A2,A3 เป็นสัญญาณที่ 74LS138 ใช้ในการ decode เพื่อจะเลือกที่จะใช้งานชิพตัวใดระหว่าง FX709 และ 8255 โดย

ถ้า A2,A3 เป็น 0,0 จะเป็นการเลือกใช้งาน FX709 โดยเอาท์พุทที่ขา Y0 ของ 74LS138 จะมีสถานะลอจิกเป็น "1"

ถ้า A2,A3 เป็น 1,0 จะเป็นการเลือกใช้งาน 8255 โดยเอาท์พุทที่ขา Y1 ของ 74LS138 จะมีสถานะลอจิกเป็น "1" ซึ่งที่ขา Y0 และ Y1 นี้จะทำการต่อเข้ากับขา CS ของ Fx709 และ 8255 ตามลำดับ

- ขา IRQ5 เป็นขาที่จะรับสัญญาณการขออินเทอร์รัพท์จากขา IRQ ของ FX709 ซึ่ง FX709 จะใช้สำหรับแจ้งให้ CPU ทราบว่าขณะนี้พร้อมที่จะรับหรือส่งข้อมูลกับ CPU แล้ว โดยข้อมูลทั้งหมดจะถูกส่งผ่านทาง ดาต้าบัส D0-D7 เหตุผลที่เลือกขา IRQ5 นี้เนื่องจากเป็นขาอินเทอร์รัพท์ที่ยังว่างอยู่

- ขา I/O CHRDY ใช้สำหรับรับสัญญาณ wait ที่จะส่งมาจาก FX709

- ขา Data Bus D0-D7 เป็นขาที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์กับฮาร์ดแวร์ที่สร้างขึ้น

- ขา แอดเดรสบิต A4-A9 และ AEN จากรูปเป็นวงจรที่ทำการ decode กลุ่มแอดเดรสขนาด 8 บิต ซึ่งการเลือกกลุ่มแอดเดรสที่จะทำการดีโค้ดนี้จะทำได้โดยการเช็ตดิฟลิวท์ที่ต่อตรงอยู่กับขา Q0-Q5 ของ 74LS688 สำหรับหน้าที่ของ 74LS688 นี้จะทำการเปรียบเทียบค่าของอินพุท 2 ชุดที่ถูกส่งเข้ามาทางขา P0-P7 และขา Q0-Q7 ถ้าอินพุทจากทั้ง 2 ชุดนี้มีค่าเท่ากันแล้ว เอาท์พุทที่ขา P=Q จะเป็นลอจิก "0" จากในวงจร ขา P0-P5 ของ 74LS688 ต่อเข้ากับแอดเดรสบิต A4-A9 จะเห็นได้ว่าเราสามารถเปลี่ยนแปลงค่า A4-A9 ได้โดยการเปลี่ยนค่าที่ดิฟลิวท์ ทำให้สามารถเปลี่ยนแปลงค่าแอดเดรสที่ต้องการจะ decode ได้โดยสะดวกขึ้น สำหรับขา Q6,Q7 นี้จะต่อลงกราวด์ (ลอจิก "0") และขา P5 ที่ต่อกับแอดเดรสบิต A9 ในกรณีที่ Q5 ถูกเช็ตเป็นลอจิก "1" จะเป็นการบังคับให้แอดเดรสที่จะ decode ได้นั้น จะต้องต้องมีแอดเดรสบิต A9 เป็น "1" เท่านั้น ส่วนขา P6 จะต่อกับสัญญาณ AEN และ P7 ต่อกับลอจิก "0" การต่อในลักษณะนี้เพื่อเป็นการป้องกันไม่ให้ 74LS688 ทำการ decode ในระหว่างขบวนการ DMA

เอาท์พุทจากขา P=Q ของ 74LS688 นี้ จะถูกนำไปใช้ในการอินาเบิล 74LS138 ซึ่งทำหน้าที่ในการดีโค้ดแอดเดรสเพื่อเลือกใช้งานระหว่าง FX709 และ 8255 อีกที่

นอกจากนี้แล้ว เอาท์พุทจากขา P=Q ของ 74LS688 ยังจะนำไปใช้ในการอินาเบิล 74LS245 โดยถ้า E มีลอจิกเป็น "1" 74LS245 จะหยุดการทำงาน คือ จะไม่มีการรับส่งข้อมูลใด ๆ

หลักการการทำงานของโปรแกรม

ในที่นี้จะขอกล่าวถึงหลักการการทำงานของโปรแกรมเป็นหัวข้อต่าง ๆ ที่สำคัญดังนี้

1.) ทำการโปรแกรม 8259A ซึ่งเป็นชิพที่ใช้ในการจัดระบบการขออินเทอร์รัพท์ CPU ใหม่ โดยการป้อนค่า ICW1 , ICW2 , ICW3 , ICW4 , OCW1 , OCW2 และ OCW3 ค่าใหม่ให้กับ 8259A ทำให้ CPU สามารถตอบรับการขออินเทอร์รัพท์จากขา IRQ5 ได้

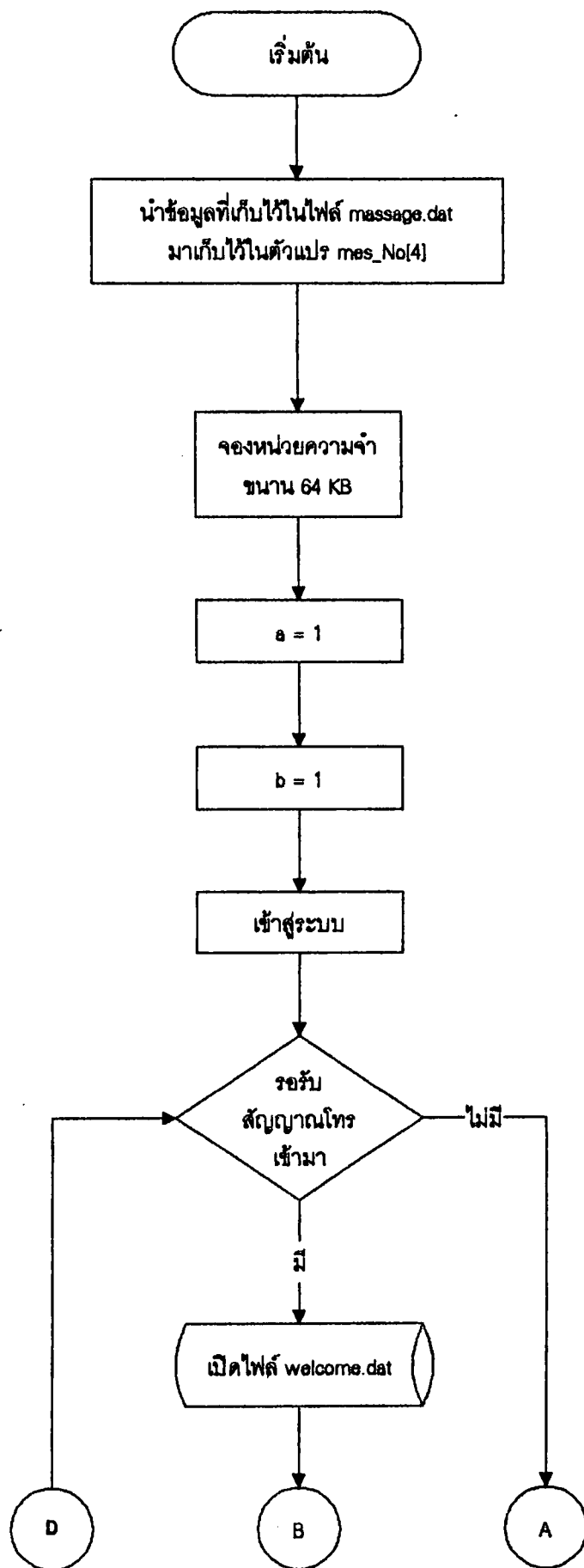
2.) ทำการจองพื้นที่หน่วยความจำในส่วนของ RAM

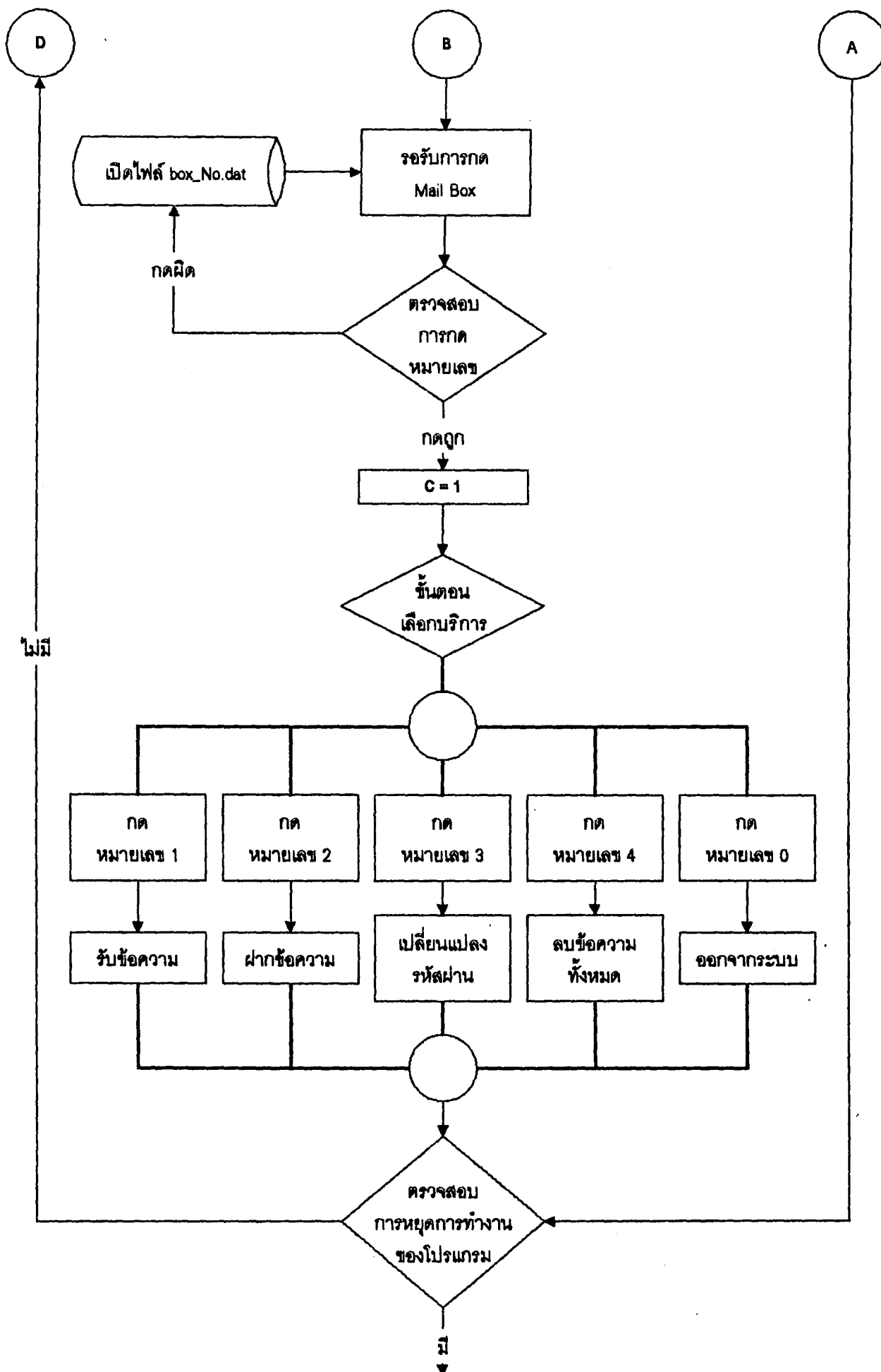
3.) ทำการโปรแกรมชิพ FX709 ให้ทำการแปลงข้อมูลเสียงเป็นข้อมูลดิจิตอล หรือแปลงข้อมูลดิจิตอลเป็นข้อมูลเสียงได้ เมื่อ FX709 ทำการแปลงข้อมูลเสร็จก็จะส่งสัญญาณการขออินเทอร์รัพท์ผ่านทางขา IRQ5 ไปบอก CPU ว่าทำการแปลงข้อมูลเรียบร้อยแล้ว

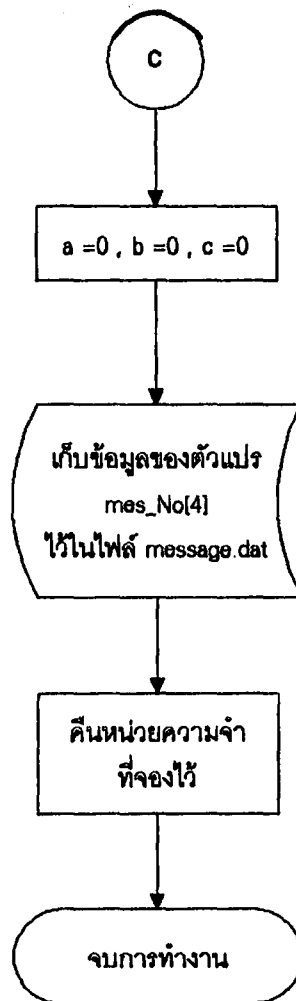
เมื่อ CPU ได้รับสัญญาณการขออินเทอร์รัพท์จากชิพ FX709 ก็จะมีการกระโดดไปทำงานในส่วนของโปรแกรมย่อย ซึ่งในโปรแกรมย่อยนี้จะมีคำสั่งให้ทำการเก็บข้อมูล หรือส่งข้อมูลระหว่างหน่วยความจำที่ทำการจองไว้กับวงจรที่ใช้ในการแปลงข้อมูลผ่านทาง Data Bus

4.) ทำการตรวจสอบสัญญาณการยกหูโทรศัพท์ (สัญญาณนี้ ใช้แสดงว่าได้มีการตัดเข้าสู่ระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติแล้ว)

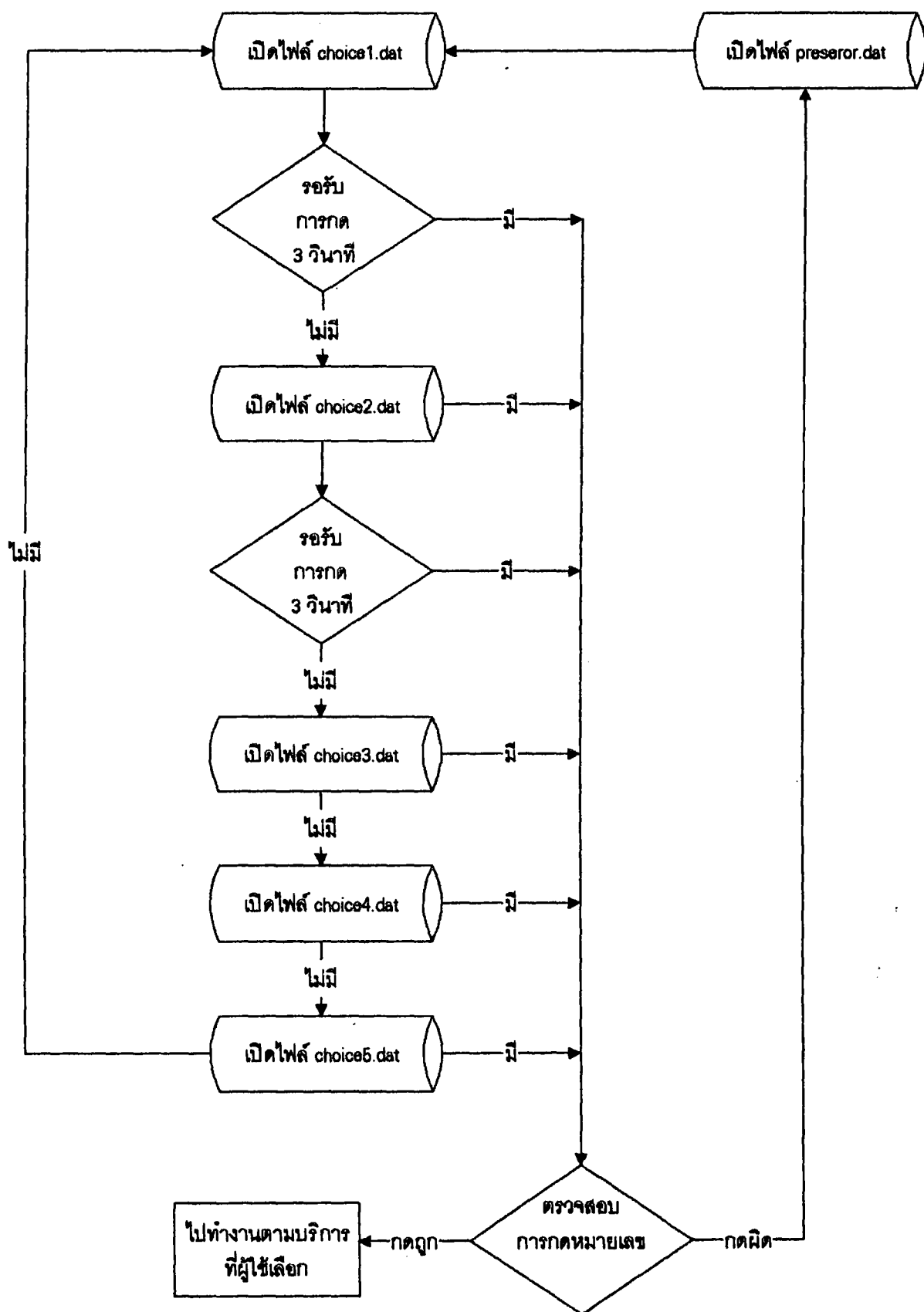
5.) ทำการตรวจสอบข้อมูลที่ได้จากการกดแป้นโทรศัพท์ (มีชิพ MT8870 ทำการแปลงสัญญาณ DTMF เป็นข้อมูลดิจิตอล 8 บิต) กับข้อมูลที่ได้กำหนดไว้ เพื่อใช้ในการตรวจสอบว่าในขั้นตอนต่อไป จะต้องไปทำงานในลักษณะใด



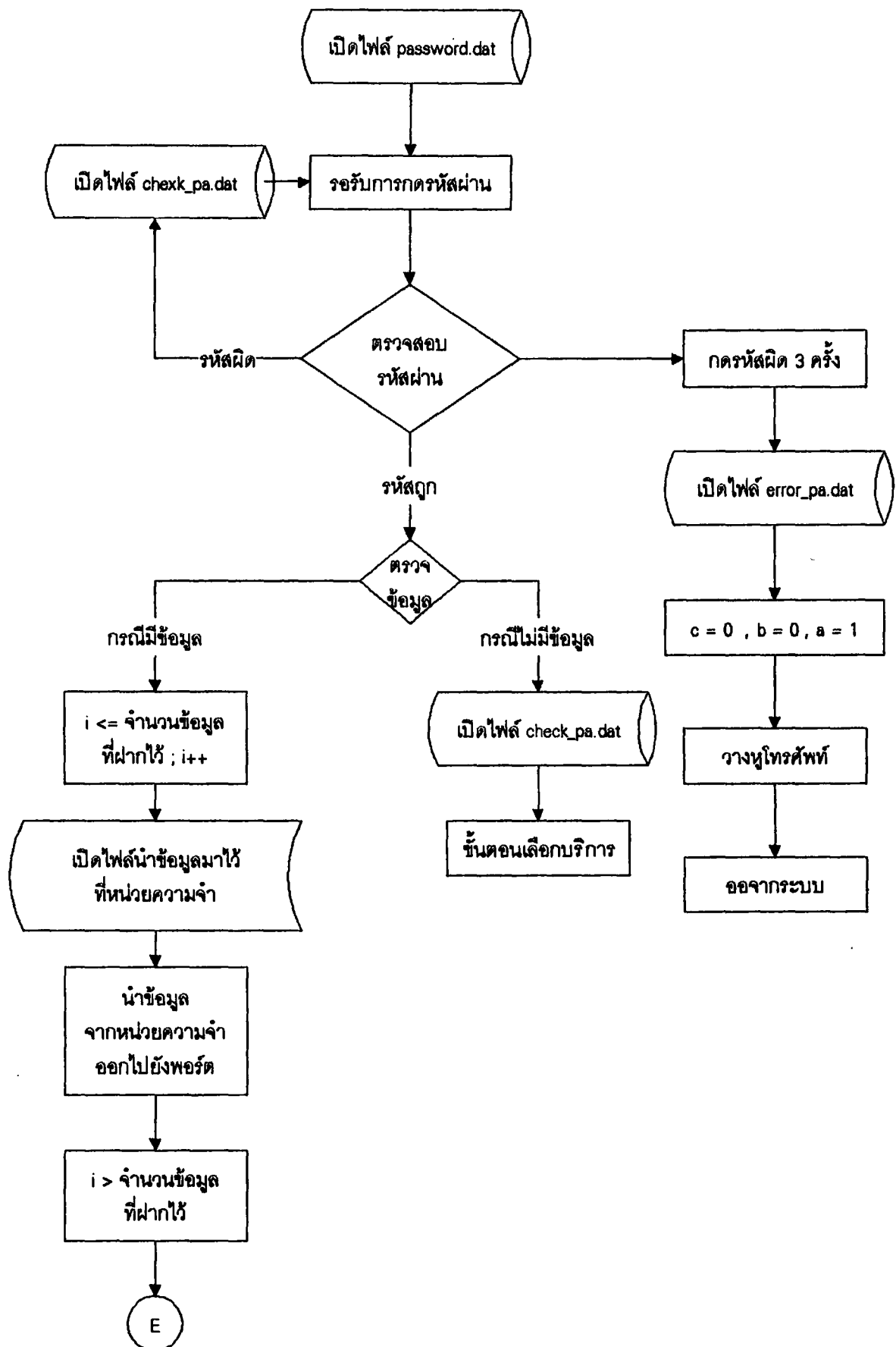


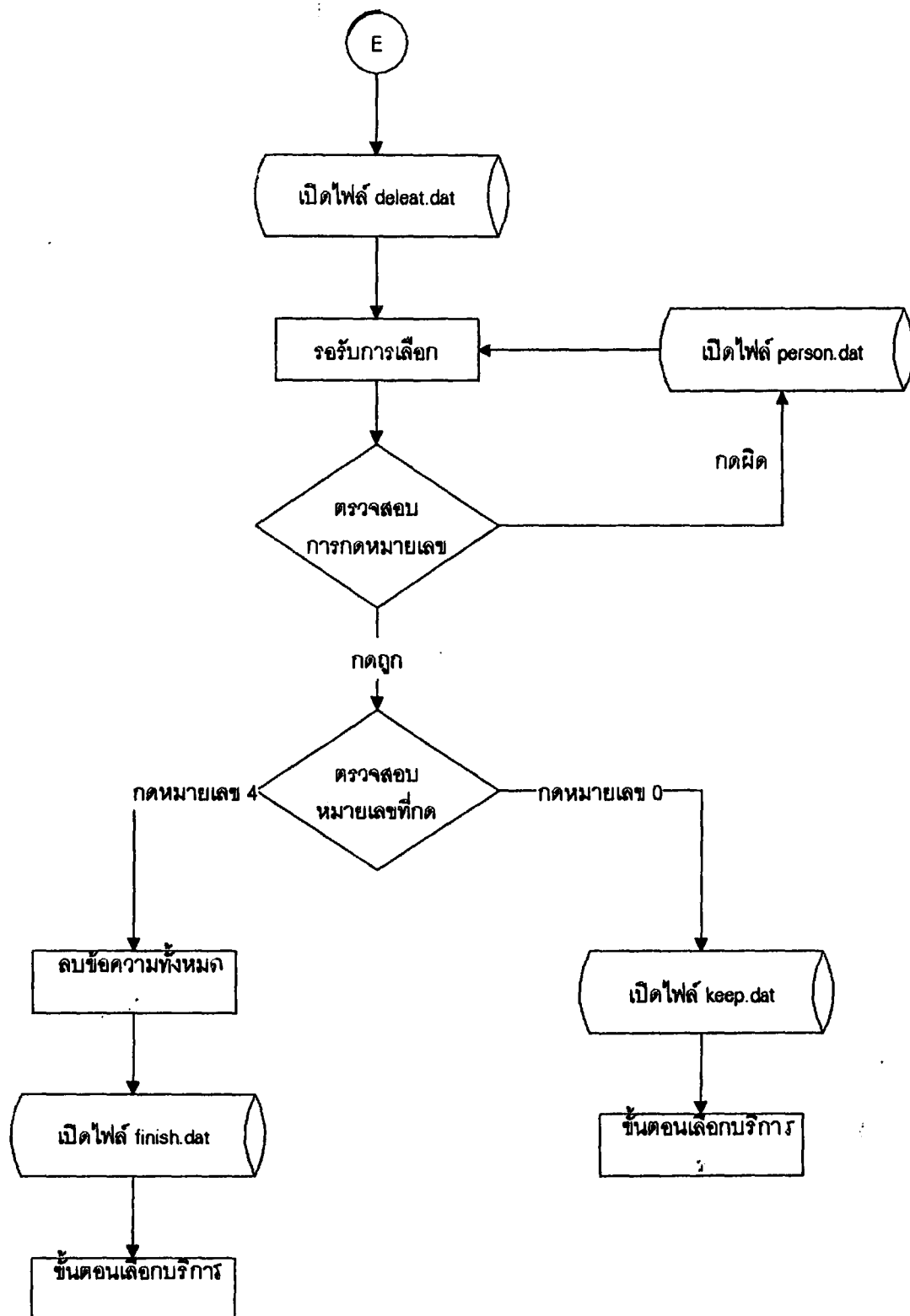


ขั้นตอนการเลือกบริการ

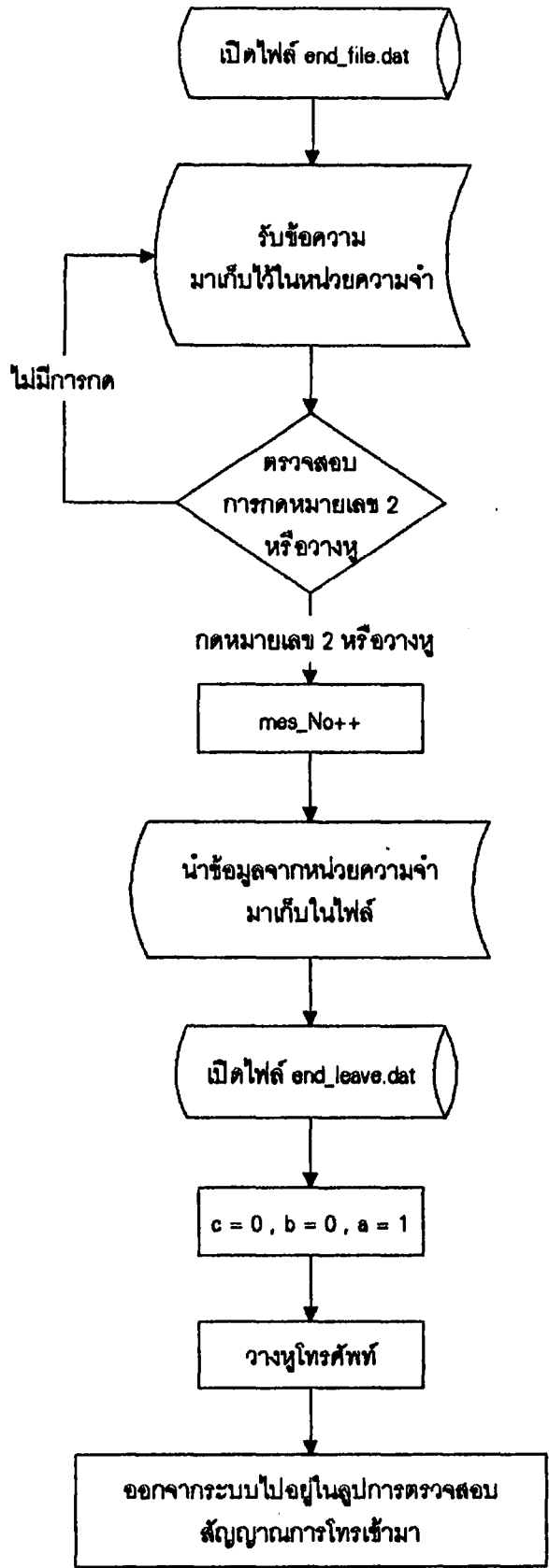


ขั้นตอนการรับข้อความ

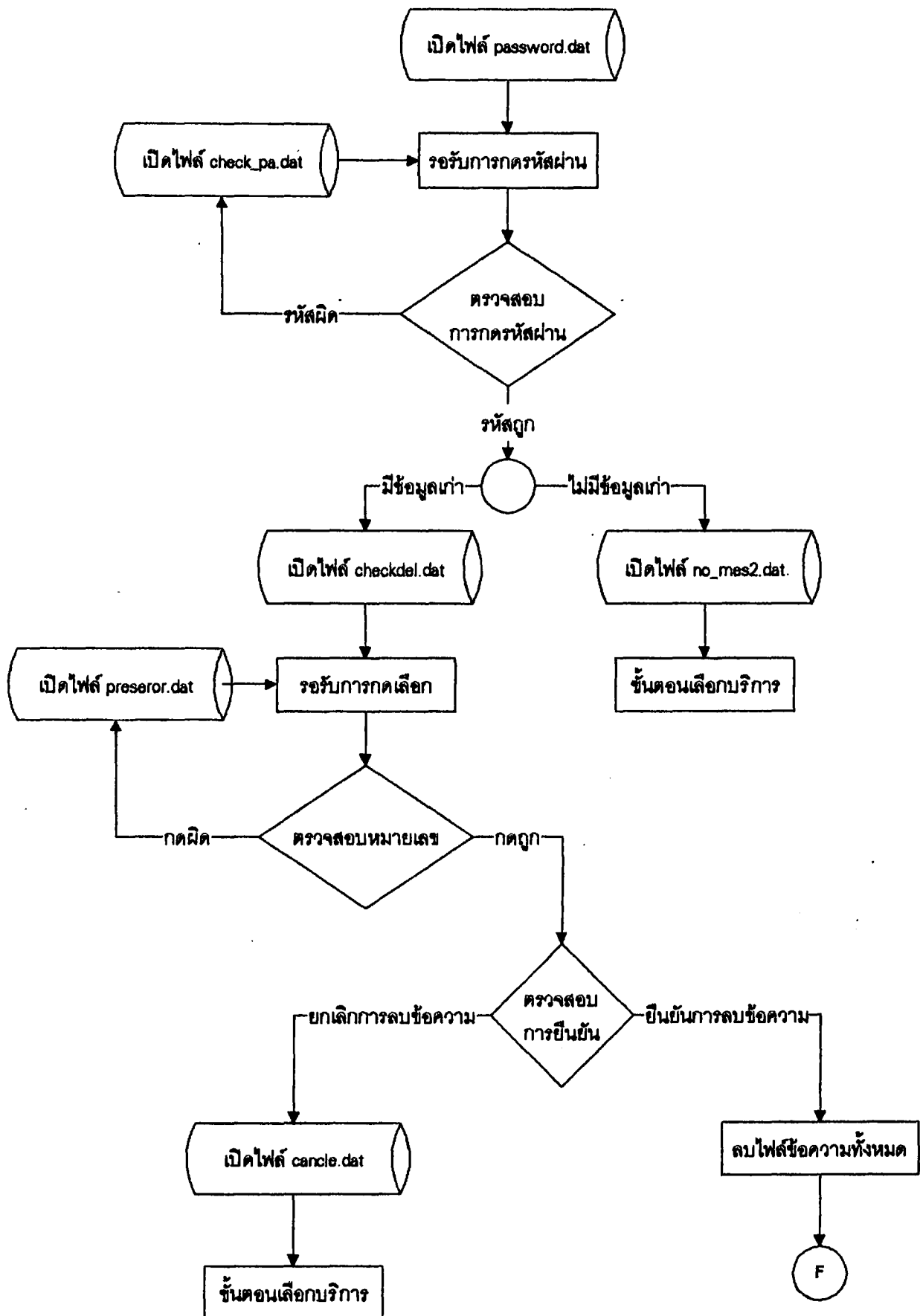


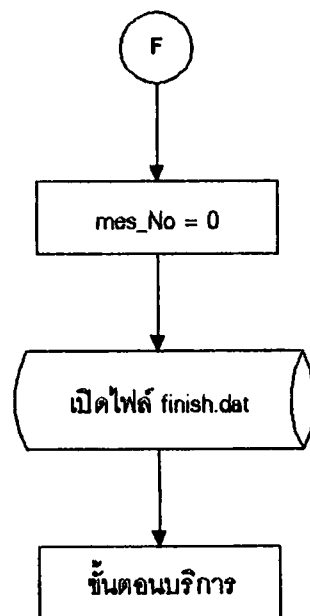


ขั้นตอนการฝากข้อความ

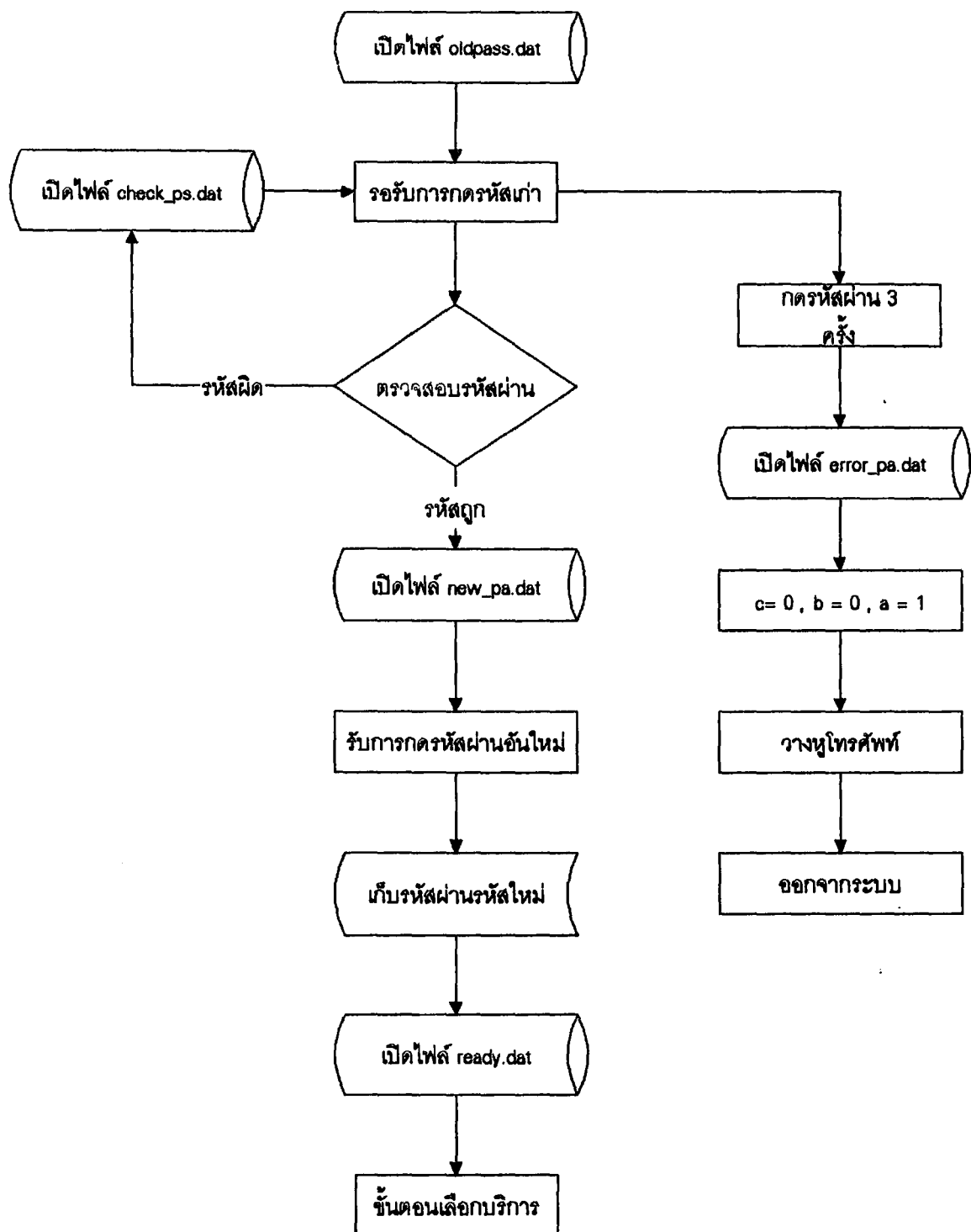


ขั้นตอนลบข้อความเก่าทั้งหมด

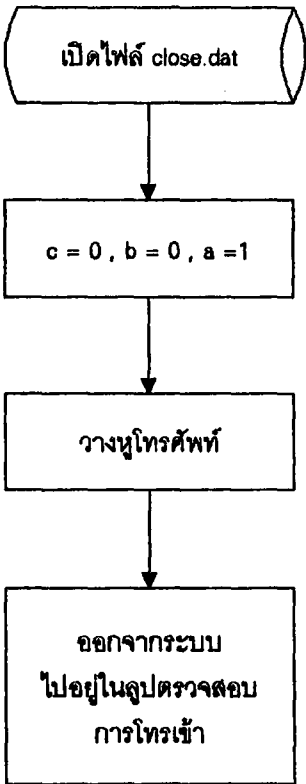




ขั้นตอนการเปลี่ยนแปลงรหัสผ่าน



ขั้นตอนการออกจากระบบ



บทที่ 4

ผลการทดลอง

เมื่อทำการทดสอบระบบฝากข้อความทางโทรศัพท์อัตโนมัติ โดยให้ผู้ทดลองอ่านคู่มือการใช้งานประกอบกับการใช้งานจริง พบว่าผู้ใช้งานส่วนใหญ่สามารถใช้งานได้อย่างสะดวก แสดงให้เห็นว่าระบบที่สร้างขึ้นนี้สามารถนำไปใช้งานได้จริง โดยมีหลักการทำงานและขั้นตอนการใช้งานดังจะกล่าวในหัวข้อต่อไป

4.1 การทำงานของระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ

การทำงานของระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติจะแบ่งได้เป็น 2 ส่วน คือ ส่วนที่ติดต่อกับผู้ใช้โดยตรง โดยในส่วนนี้จะเป็นการเขียนโปรแกรมเพื่อควบคุมวงจรให้สามารถบันทึกและส่งสัญญาณเสียงที่บันทึกไว้ออกไปให้ผู้รับทราบข้อมูลต่างๆ ในขณะที่ผู้ใช้งานกำลังติดต่อกับระบบอยู่ ก็จะมีเสียงโต้ตอบออกมาเพื่อบอกวิธีใช้งานให้ผู้รับทราบ ส่วนที่สอง คือ โปรแกรมแสดงข้อมูลต่างๆ ของเจ้าของ Mail Box และจะมีการบันทึกวันและเวลาที่มีการฝากข้อความ รายละเอียดของส่วนที่ติดต่อกับผู้ใช้โดยตรงนี้ จะมีแผนภาพการทำงานเป็นดังรูป ซึ่งรายละเอียดของส่วนต่างๆ จะเป็นดังนี้

1.1) Handshaking : เป็นส่วนที่จะคอยตรวจสอบว่ามีผู้เข้ามาติดต่อแล้วหรือยัง ถ้ามีการติดต่อเข้ามาแล้วก็จะทำการเข้าสู่ระบบทันทีซึ่งจะมีเสียงมาบอกให้ผู้รับทราบว่า " ขณะนี้ท่านกำลังเข้าสู่ระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ โปรดกด Mail Box ที่ต้องการติดต่อ "

1.2) การตรวจสอบการกด Mail Box : โปรแกรมส่วนนี้จะคอยตรวจว่าผู้ใช้งานกด Mail Box ถูกต้องตรงตามที่กำหนดไว้หรือยัง ถ้ายังไม่ถูกต้องจะมีเสียงมาเตือนผู้ใช้งานว่า " ท่านกด Mail Box ผิดผลขาด โปรดกดอีกครั้ง " เรื่อย ๆ ไปจนกว่าผู้ใช้งานจะกดถูกต้อง เมื่อผู้ใช้งานกดถูกต้องแล้วก็มีเสียงมาบอกให้ทราบว่าในระบบมีบริการอะไรบ้าง

1.3) การเลือกบริการต่าง ๆ : บริการต่าง ๆ ที่มีในระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ มีดังนี้

- | | |
|--------|----------------------|
| กด (1) | ฟังข้อความ |
| กด (2) | รับฝากข้อความ |
| กด (3) | เปลี่ยนแปลงรหัสผ่าน |
| กด (4) | ลบข้อความเก่าทั้งหมด |

กต (0) ออกจากกระบบฝากข้อความทางโทรศัพท์อัตโนมัติ

ในระหว่างประโยคจะมีช่วงเวลาเว้นไว้ประมาณ 3 วินาทีเพื่อรอการกดเลือกบริการจากผู้ใช้งาน ซึ่งถ้าไม่มีการกดแป้นใด ๆ ก็จะมีข้อความถัดไปบอกออกมา การทำงานจะเป็นเช่นนี้ไปเรื่อย ๆ จนกว่าจะมีการกดเลือกบริการจากผู้ใช้งาน และถ้าผู้ใช้งานกดตัวเลือกอื่นที่ไม่ตรงตามที่กำหนดไว้ ก็จะมีเสียงมาเตือนผู้ใช้งานว่า " **ไม่มีตัวเลือกที่ท่านกด โปรดกดอีกครั้ง** " จนกว่าตัวเลือกที่ผู้ใช้งานกดจะตรงกับตัวเลือกใดตัวเลือกหนึ่งที่ได้กำหนดไว้

1.4) การฟังข้อความ : เมื่อผู้ใช้งานที่เป็นเจ้าของ Mail Box ต้องการจะฟังข้อความที่ฝากถึงตน หลังจากที่ยกตัวเลือก (1) แล้ว จะมีเสียงมาบอกผู้ใช้งานว่า " **โปรดกดรหัสผ่านของท่าน** " ซึ่งถ้ารหัสผ่านที่กดไม่ถูกต้องก็จะมีเสียงมาเตือนว่า " **รหัสผ่านของท่านผิดพลาด โปรดกดอีกครั้ง** " และถ้ากดรหัสผ่านไม่ถูกต้องถึง 3 ครั้ง ก็จะมีเสียงมาบอกให้ผู้ใช้งานทราบว่า " **รหัสผ่านของท่านผิดพลาด ท่านกำลังจะออกจากกระบบฝากข้อความทางโทรศัพท์อัตโนมัติ สวัสดีค่ะ** " แล้วผู้ใช้งานนั้นก็จะถูกตัดจากระบบไป ถ้าจะเข้ามาในระบบใหม่ก็จะต้องโทรศัพท์เข้ามาอีกครั้ง

ถ้ากดรหัสผ่านถูกต้อง แล้วไม่มีข้อความฝากไว้เลย ก็จะมีเสียงมาบอกว่า " **ไม่มีข้อความฝากถึงท่านเลย** " จากนั้นก็จะกลับไปในขั้นตอนเลือกบริการต่าง ๆ อีกครั้ง

แต่ถ้ามีข้อความฝากไว้ก็จะทำการส่งข้อความที่ฝากไว้ทั้งหมดออกมา เมื่อฟังข้อความทั้งหมดเรียบร้อยแล้วก็จะมีเสียงมาถามว่า " **กต (4) เมื่อต้องการลบข้อความทั้งหมด กต (0) เมื่อต้องการเก็บข้อความทั้งหมดไว้** " ถ้าผู้ใช้งานกดเลข 4 ก็จะทำให้การลบข้อความทั้งหมดที่เก็บไว้ และจะมีเสียงมาบอกว่า " **ข้อความทั้งหมดได้ถูกลบเรียบร้อยแล้ว** " จากนั้นก็จะกลับเข้าสู่ขั้นตอนการเลือกบริการต่าง ๆ อีกครั้ง

ถ้าผู้ใช้งานกดเลข 0 ก็จะทำให้การเก็บข้อความทั้งหมดไว้ โดยจะมีเสียงมาบอกว่า " **ข้อความทั้งหมดได้ถูกเก็บไว้เรียบร้อยแล้ว** " จากนั้นก็จะกลับเข้าสู่ขั้นตอนการเลือกบริการต่าง ๆ อีกครั้ง

แต่ถ้าผู้ใช้งานกดเลขอื่น ๆ ที่ไม่ใช่เลข 4 และ เลข 0 ก็จะมีเสียงมาเตือนผู้ใช้งานว่า " **ไม่มีตัวเลือกที่ท่านกด** " ซึ่งผู้ใช้งานจะต้องทำการกดใหม่จนกว่าจะถูกตรงตามที่กำหนด

1.5) การฝากข้อความ : หลังจากที่ยกตัวเลือกเลข 2 ก็จะมีเสียงมาบอกผู้ใช้งานว่า " **ท่านได้เข้าสู่ระบบบันทึกข้อความ โปรดกด (2) เมื่อสิ้นสุดการบันทึก** " จากนั้นผู้ใช้งานก็ฝากข้อความถึงผู้ที่ต้องการได้ เมื่อผู้ใช้งานพูดจบแล้วกดเลข (2) ก็จะมีเสียงมาบอกว่า " **บันทึกข้อความเรียบร้อยแล้วท่าน กำลังจะออกจากกระบบฝากข้อความทางโทรศัพท์อัตโนมัติ สวัสดีค่ะ** " แล้วผู้ใช้งานนั้นก็จะถูกตัดออกจากระบบไป

1.6) การเปลี่ยนแปลงรหัสผ่าน : เมื่อผู้ใช้งานกดเลข (3) ก็จะทำให้เข้าสู่บริการการเปลี่ยนแปลงรหัสผ่าน โดยจะมีเสียงมาบอกว่า " **กรุณากดรหัสผ่านอันเก่าของท่าน** " ซึ่งการตรวจสอบรหัสผ่านก็จะ

เหมือนกับในขั้นตอนของการฟังข้อความเมื่อผู้ใช้กดรหัสผ่านถูกต้องแล้วก็จะมีเสียงมาบอกว่า "กรุณากดรหัสผ่านอันใหม่ของท่าน" และเมื่อผู้ใช้กดรหัสผ่านอันใหม่เรียบร้อยแล้วก็จะมีเสียงมาแจ้งให้ทราบว่า "เปลี่ยนแปลงรหัสผ่านเรียบร้อยแล้ว" จากนั้นก็จะกลับไปขั้นตอนการเลือกบริการต่าง ๆ อีกครั้ง

1.7) การลบข้อความเก่าทั้งหมด : หลังจากที่ใช้กดเลข (4) แล้วก็จะเข้าสู่บริการการลบข้อความเก่าทั้งหมด โดยในขั้นแรกจะมีเสียงมาบอกว่า "กรุณากดรหัสผ่านของท่าน" ซึ่งการตรวจสอบรหัสผ่านก็จะเหมือนกับในขั้นตอนของการฟังข้อความ เมื่อผู้ใช้กดรหัสผ่านถูกต้องแล้วก็จะมีเสียงมาถามผู้ใช้ว่า "ท่านต้องการลบข้อความเก่าทั้งหมดใช่ไหม ถ้าใช่กด (4) ไม่ใช่ กด (0)" " ถ้าผู้ใช้ยืนยันจะลบข้อความทั้งหมดก็จะกดเลข (4) แล้วจะมีเสียงมาแจ้งว่า " ข้อความทั้งหมดได้ถูกลบเรียบร้อยแล้ว " จากนั้นก็จะกลับเข้าสู่ขั้นตอนการเลือกบริการต่าง ๆ อีกครั้ง

ถ้าผู้ใช้ต้องการยกเลิกการลบข้อความเก่าก็จะกดเลข (0) แล้วจะมีเสียงมาบอกว่า " ยกเลิกการลบข้อความเก่า " ซึ่งข้อความทั้งหมดก็จะยังถูกเก็บไว้เหมือนเดิม แล้วผู้ใช้ก็จะกลับเข้าสู่ขั้นตอนการเลือกบริการต่าง ๆ อีกครั้ง

แต่ถ้าผู้ใช้กดเลขอื่น ๆ ที่ไม่ใช่เลข (0) และเลข (4) ก็จะมีเสียงมาเตือนว่า " ไม่มีตัวเลือกที่ท่านกด " ซึ่งผู้ใช้จะต้องทำการกดใหม่จนกว่าจะกดถูกต้อง

1.8) การออกจากระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ : เมื่อผู้ใช้กดเลข (0) ก็จะมีเสียงมาแจ้งว่า " ท่านกำลังจะออกจากระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ สวัสดีค่ะ " แล้วผู้ใช้ก็จะถูกตัดออกจากระบบไป ซึ่งถ้าต้องการเข้ามาในระบบใหม่ก็ทำได้โดยการโทรเข้ามาอีกครั้ง

4.2 ผลการทดลอง

การบันทึกข้อความโดยทำการบันทึกลงบนฮาร์ดดิสก์ในแต่ละประโยคจะใช้เนื้อที่แสดงดังตารางที่ 4-1

จากการทดลองใช้งานจริง ได้ผลสรุปดังนี้

- การเข้าสู่ระบบเป็นไปอย่างรวดเร็วและต่อเนื่องโดยมีข้อความเสียงอธิบายการใช้งานทุกขั้นตอน เพื่ออำนวยความสะดวกแก่ผู้ใช้
- มีความปลอดภัยในการเก็บรักษาข้อมูลของเจ้าของ Mail Box โดยระบบตรวจสอบรหัสผ่านสามารถทำงานได้อย่างมีประสิทธิภาพ

ข้อผิดพลาดที่เกิดขึ้นในการทดสอบระบบ

- คุณภาพของเสียงที่ทำการบันทึกในขณะที่ทำการเรียกกลับ (Playback) เพื่อเรียกฟังยังคงพบสัญญาณรบกวนอยู่บ้าง ทำให้ความคมชัดของเสียงยังไม่ดีเท่าที่ควร
- ผลจากการตรวจสอบสัญญาณ DTMF จากการกดแป้นโทรศัพท์ ในบางครั้งข้อมูลที่ได้ไม่ตรงกับตัวเลขที่กด ทำให้การประมวลผลของโปรแกรมที่เขียนทดสอบเกิดความผิดพลาด โดยมีเปอร์เซ็นต์ความผิดพลาดประมาณ 5%
- มีระยะเวลาที่ใช้ในการบันทึกข้อความประมาณ 15 วินาที ซึ่งในบางครั้งอาจไม่เพียงพอ กับความต้องการของผู้ใช้

จากตารางที่ 4-1 แสดงจำนวนไบต์ที่ใช้เก็บข้อความต่างๆ เจลีย์แล้วพบว่า ใน 1 พยางค์จะใช้จำนวนไบต์ในการเก็บประมาณ 2,528 ไบต์ เนื้อที่ทั้งหมดที่ใช้ในการบันทึกข้อความที่ใช้โต้ตอบกับผู้ใช้ คือ 725,032 ไบต์ และในการบันทึกเสียงพูดหรือฝากข้อความในแต่ละครั้งสามารถบันทึกได้นานประมาณ 15 วินาที

4.3 สรุปผลการทดลอง

ระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติที่สร้างขึ้นนี้ยังไม่สมบูรณ์เท่าที่ควรนักคือพบข้อผิดพลาดเมื่อนำไปใช้งานจริง เนื่องจากบางครั้งอาจเกิดความผิดพลาดจากส่วนที่เป็นฮาร์ดแวร์ และในบางครั้งก็อาจจะเกิดความผิดพลาดเนื่องจากโปรแกรมที่ควบคุมการทำงาน โดย สามารถสรุปสาเหตุที่สำคัญได้ดังนี้ คือ

ในส่วนของฮาร์ดแวร์

เกิดจากความผิดพลาดในการออกแบบวงจรในส่วนที่เป็นสัญญาณอนาล็อกเป็นส่วนใหญ่ จึงเป็นสาเหตุหลักที่ทำให้คุณภาพของเสียงที่ได้ด้อยลงไป สำหรับแนวทางแก้ไขที่ได้กระทำในโครงการพิเศษนี้ คือ การเชื่อมต่อวงจรระหว่างส่วนที่เป็นสัญญาณอนาลอกกับสัญญาณดิจิทัล จะไม่ใช้กราวด์ร่วมกัน และระมัดระวังกราวด์ลูปที่อาจเกิดขึ้นได้ ซึ่งสามารถลดปัญหาเรื่องสัญญาณรบกวนลงได้บ้าง

ในส่วนของโปรแกรม

เนื่องจากระบบนี้ทำงานโดยจะส่งสัญญาณอินเทอร์รัพท์ให้กับ CPU โดยเขียนโปรแกรมที่เป็นภาษา C ควบคุม ซึ่งในบางครั้งสัญญาณอินเทอร์รัพท์จากระบบที่สร้างขึ้นนี้จะไปรบกวนการทำงานของหลักของ CPU ทำให้บางครั้งระบบของคอมพิวเตอร์เกิดความผิดพลาดขึ้นได้ แนวทางแก้ไขปัญหาดังกล่าวนี้นี้ได้สร้างโปรแกรมควบคุม 8259A ขึ้นใหม่ และเลือกใช้อินเทอร์รัพท์หมายเลข 5 (IRQ5) ทำให้ CPU สามารถตอบสนองต่อการเรียกใช้อินเทอร์รัพท์ได้อย่างมีประสิทธิภาพมากขึ้น

ข้อความที่บันทึกเก็บไว้ในฮาร์ดดิสต์สำหรับใช้โต้ตอบกับผู้ใช้ระบบ	Bytes.
ขณะนี้ท่านกำลังเข้าสู่ระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ โปรดกด Mailbox ที่ต้องการติดต่อ	56,495
ท่านกด Mailbox ผิดผลลาด โปรดกดอีกครั้ง	28,230
กด 1 เพื่อฟังข้อความ	20,072
กด 2 เพื่อบันทึกข้อความ	22,492
กด 3 เพื่อเปลี่ยนแปลงรหัสผ่าน	24,547
กด 4 เพื่อลบข้อความเก่าทั้งหมด	27,224
กด 0 เพื่อออกจากระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ	33,311
ไม่มีตัวเลือกที่ท่านกด	16,358
โปรดกดรหัสผ่านของท่าน	20,290
รหัสผ่านของท่านผิดผลลาด โปรดกดอีกครั้ง	29,390
รหัสผ่านของท่านผิดผลลาด ท่านกำลังจะออกจากระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ สวัสดีค่ะ	38,596
ไม่มีข้อความฝากถึงท่านเลย	19,257
กด 4 เมื่อต้องการลบข้อความทั้งหมด กด 0 เมื่อต้องการเก็บข้อความทั้งหมดไว้	47,136
ข้อความทั้งหมดได้ถูกลบเรียบร้อยแล้ว	27,083
ข้อความทั้งหมดได้ถูกเก็บไว้เรียบร้อยแล้ว	26,485
ท่านได้เข้าสู่ระบบบันทึกข้อความ โปรดกด 2 เมื่อสิ้นสุดการบันทึก	42,093
บันทึกข้อความเรียบร้อยแล้ว ท่านกำลังจะออกจากระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ สวัสดีค่ะ	58,406
โปรดกดรหัสผ่านที่ท่านต้องการเปลี่ยนแปลง	31,166
โปรดกดรหัสผ่านอันใหม่	20,184
เปลี่ยนแปลงรหัสผ่านเรียบร้อยแล้ว	19,112
ท่านต้องการลบข้อความทั้งหมดใช่หรือไม่ ถ้าใช่กด 4 ไม่ใช่กด 0	56,353
ยกเลิกการลบข้อความเก่า	18,832
ท่านกำลังจะออกจากระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ สวัสดีค่ะ	42,920

ตารางที่ 4-1 แสดงจำนวนไบต์ที่ใช้เก็บข้อความต่าง ๆ

บทที่ 5

สรุปผลการทำงานและข้อเสนอแนะในการพัฒนาระบบ

5.1 สรุปผลการทำงาน

ระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติจะมีศูนย์กลางของระบบซึ่งประกอบไปด้วยเครื่องโทรศัพท์ที่สมาชิก(เจ้าของ Mail Box) หรือผู้ที่ต้องการติดต่อกับสมาชิกเรียกสายเข้าที่เลขหมายและคอมพิวเตอร์ส่วนบุคคลที่ติดตั้งการ์ดที่ถูกออกแบบมาสำหรับใช้งานในระบบนี้โดยเฉพาะ

การทำงานของระบบ คือ ที่ศูนย์กลางจะทำหน้าที่เป็นแหล่งเก็บข้อมูลให้แก่สมาชิกในกรณีที่มีผู้เรียกสายและต้องการฝากข้อความถึงเจ้าของ Mail Box ระบบจะทำการเก็บ และบันทึกข้อมูลทั้งหมดลงบนหน่วยความจำของเครื่องคอมพิวเตอร์ ซึ่งทำให้สามารถเก็บข้อมูลได้เป็นจำนวนมากโดยขึ้นกับขนาดของหน่วยความจำ และประสิทธิภาพของเครื่องคอมพิวเตอร์ที่นำมาใช้งาน

นอกจากการบริการรับฝากข้อความให้กับสมาชิกแล้ว ระบบยังมีฟังก์ชันการใช้งานเพิ่มเติมดังนี้

- สามารถตรวจสอบ วันและเวลา ที่มีการบันทึกข้อความโดยที่สมาชิกสามารถดูข้อมูลดังกล่าวได้ ด้วยการเรียกดูข้อมูลโดยที่ข้อมูลจะปรากฏบนหน้าจอคอมพิวเตอร์
- สมาชิกสามารถเปลี่ยนแปลงรหัสผ่านโดยการกดรหัสผ่านอันใหม่ผ่านทางแป้นกดของเครื่องโทรศัพท์ได้ ซึ่งเป็นการอำนวยความสะดวกแก่ผู้ใช้ในกรณีที่อยู่ไกลจากเครื่องคอมพิวเตอร์ที่ควบคุมระบบ หรืออาจเลือกใช้วิธีการเปลี่ยนแปลงรหัสผ่าน โดยการป้อนข้อมูลผ่านทางคีย์บอร์ดก็สามารถทำได้

5.2 ข้อเสนอแนะในการพัฒนาระบบ

เนื่องจากระบบที่สร้างขึ้นนี้มีขีดจำกัดในเรื่องเวลาที่ใช้ในการบันทึกมีช่วงเวลาที่ไม่นานมากนัก เนื่องจากโปรแกรมที่ควบคุมการทำงานมีความสามารถในการจองเนื้อที่บนหน่วยความจำจำกัดที่ 64 kbyte และประสิทธิภาพของเสียงขณะทำการเรียกฟังข้อความนั้นยังไม่คมชัดเท่าที่ควรจึงจำเป็นต้องพัฒนาระบบต่อไป โดยสามารถเพิ่มเติมส่วนที่พัฒนาคุณภาพเสียง เช่น ระบบ DPS และอาจลดข้อจำกัดในส่วนของการจองหน่วยความจำเพื่อใช้งานโดยอาจพัฒนาระบบในส่วนของการบีบอัดให้มีประสิทธิภาพดีขึ้น หรือนำระบบ Compress ซึ่งเป็นเทคโนโลยีอันหนึ่งที่นำศึกษาเป็นอย่างมากมายมาประยุกต์ใช้งานร่วมด้วย

นอกจากนี้ยังควรพัฒนาระบบให้มีความสะดวกในการใช้งานมากขึ้นโดยเพิ่มฟังก์ชันบริการมากขึ้น เช่น สามารถเลือกฟังข้อความที่ลำดับใดก็ได้โดยไม่จำเป็นต้องฟังข้อความที่ทำการบันทึกไว้ตั้งแต่ข้อความแรกเพื่อสะดวกและไม่เป็นการเสียเวลา

สามารถเลือกเก็บหรือลบข้อมูลที่ต้องการและสำคัญไว้ เนื่องจากในโครงการพิเศษนี้มีขีดความสามารถเพียงให้บริการลบข้อความเก่าโดยทำการลบทุกข้อความที่ได้ทำการบันทึกไว้พร้อมๆ กันและควรปรับปรุงให้มีส่วนที่ทำการลบข้อความเก่าที่ทำการบันทึกไว้นานเกินระยะเวลาที่กำหนดไว้ เป็นต้น

ภาคผนวก(Appendix)

ภาคผนวก ก โปรแกรมที่ใช้ในระบบรับฝากข้อความทางโทรศัพท์อัตโนมัติ

```
#include <alloc.h>
#include <conio.h>
#include <dos.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <8255.inc>
#include <choice.inc>
#include <password.inc>
#include <name.inc>
#include <m8259.inc>
#include <end.inc>
```

```
#define INTR 0X0d
```

```
void interrupt( *oldhandler )( void );
void recieve_8255( int * , int );
int enter_choice( int * );
int password( int * , int * );
void creat_filename( char * , int );
void mannage_8259( void );
int check_EOF( void );
```

```
char far *pt_const , far *data_pt;
long int count;
```

```

void interrupt input( void )      /* Service routine for IRQ5 */
{
    disable();
    *data_pt = inportb(0x302);
    data_pt++;
    count++;
    enable();
    outp(0x20,0x20);              /* End Of Interrupt (EOI) */
}

```

```

void interrupt output( void )    /* Service routine for IRQ5 */
{
    disable();
    data_pt++;
    enable();
    outp(0x20,0x20);              /* End Of Interrupt (EOI) */
}

```

```

int main(void)
{
    FILE *pass_pt;
    FILE *out;
    FILE *in;
    FILE *rec;
    FILE *mes;

    int a,b,c,i,j,k,chk,check_pass;
    int box_No[3],real_pass[4],mes_No[4];

```

```

int pass[4],check[2],choose[2],del[2];

unsigned char hook;

char passw[12],name[10],greet[11],confirm;

char far *file_pt;


passw[0] = 'p'; passw[1] = 'a'; passw[2] = 's'; passw[3] = 's';
passw[4] = 'w'; passw[5] = '_'; passw[7] = '.'; passw[8] = 'd';
passw[9] = 'a'; passw[10] = 't'; passw[11] = '\0';


if( (mes = fopen("message.dat","r")) == NULL )    /* Get mes_No[] */
{
    printf("Error in open file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
i = 0;
while( !feof(mes) )
{
    mes_No[i] = getc( mes );
    if( ferror( mes ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    i++;
}
fclose( mes );

```

```

if( ( pt_const = farmalloc(0xFFFF) ) == NULL )
{
    printf("Can't allocate memory\n");
    exit(1);
}

a = 1;
while( a )                                /* LOOP HANDCHECKING */
{
    clrscr();

    gotoxy( 34,12 ); printf("Handshaking...\n");
    gotoxy( 21,14 ); printf("Press anykey to stop voice mailbox system.");
    outputb( 0x307,0x90 );    _ /* Send control word to 8255 */

    b = 1;
    while( b )                            /* LOOP OPERATING */
    {
        outputb( 0x305,0x01 );
        hook = inputb( 0x304 );
        if( (hook & 0x20) == 0x20 )        /* check hook */
        {
            clrscr();

            gotoxy( 30,12); printf("*** OPERATING ***");
            data_pt = file_pt = pt_const;    /* welcome */
            if( (out = fopen("welcome.dat","rb")) == NULL )
            {
                printf("Error in open file.\n");
            }
        }
    }
}

```

```

    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
while( !feof(out) )
{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( out );
manpage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );          /*Select output voice*/
outportb( 0x300,0x48 );          /*Out Value to A Instruction*/
outportb( 0x301,0x3F );          /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );

```

```

a = 1;
while( a )                /* LOOP CHECK MAILBOX */
{
    outportb( 0x305,0x01 );    /* select input voice */
    recieve_8255( box_No,2 );
    if( box_No[0] == '1' )    /* check Mail Box No. */
    {
        name[0] = '1';
        if( box_No[1] == '1' )
        {
            passw[6] = '0'; name[1] = '1';
            j = 0; a = 0;
        }
        else if( box_No[1] == '2' )
        {
            passw[6] = '1'; name[1] = '2';
            j = 1; a = 0;
        }
        else if( box_No[1] == '3' )
        {
            passw[6] = '2'; name[1] = '3';
            j = 2; a = 0;
        }
        else if( box_No[1] == '4' )
        {
            passw[6] = '3'; name[1] = '4';
            j = 3; a = 0;
        }
    }
}

```

```

else
{
data_pt = file_pt = pt_const;    /*Wrong mailbox No.*/
if( (out = fopen("box_No.dat","rb")) == NULL )
{
printf("Error in open file.\n");
outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
exit(1);
}
while( !feof(out) )
{
*file_pt =getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
exit(1);
}
file_pt++;
}
fclose( out );

mannage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outputb( 0x305,0x03 );    /*Select output voice*/
outputb( 0x300,0x48 ); /*Out Value to A Instruction*/
outputb( 0x301,0x3F ); /*Out Valut to B Instruction*/

```

```

while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR, oldhandler );

a = 1;
    }
}
else if( box_No[0] == '0' )
{
    data_pt = file_pt = pt_const;      /* Out of process */
    if( (out = fopen("close.dat", "rb")) == NULL )
    {
        printf("Error in open file.\n");
        outportb( 0x305, 0x00 ); outportb( 0x305, 0x01 );
        exit(1);
    }
    while( !feof(out) )
    {
        *file_pt = getc( out );
        if( ferror( out ) )
        {
            printf("Error in reading file.\n");
            outportb( 0x305, 0x00 ); outportb( 0x305, 0x01 );

```

```

    exit(1);
}
file_pt++;
    }
    fclose( out );

    manpage_8259();
    oldhandler = getvect( INTR );
    setvect( INTR,output );
    outportb( 0x305,0x03 ); _ /*Select output voice*/
    outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
    outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/

    while( data_pt <= file_pt )
    {
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR,oldhandler );
    exit(0);
}

else
{
    data_pt = file_pt = pt_const; /* Wrong mailbox No.*/
    if( (out = fopen("box_No.dat","rb")) == NULL )
    {

```

```

printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
while( !feof(out) )
{
*file_pt =getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
file_pt++;
}
fclose( out );
manpage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 ); _ /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/

while( data_pt <= file_pt )
{
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;

```

```

    }

    setvect( INTR,oldhandler );

    a = 1;
}

}                                /* END OF LOOP CHECK MAILBOX */

c = 1;

****/ while( c )                /* LOOP CHOOSE */
{
    while( 1 )
    {
        data_pt = file_pt = pt_const;    /* Enter choice#1 */

        if( (out = fopen("choice1.dat","rb")) == NULL )
        {
            printf("Error in open file.\n");
            outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
            exit(1);
        }

        while( !feof(out) )
        {
            *file_pt = getc( out );

            if( ferror( out ) )
            {
                printf("Error in reading file.\n");
                outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
                exit(1);
            }

            file_pt++;
        }
    }
}

```

```

fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 ); _ /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Value to B Instruction*/
while( data_pt <= file_pt )
{
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );
outportb( 0x305,0x01 );
if( enter_choice(choose) == 1 )
break;
else
{
data_pt = file_pt = pt_const; /* Enter choice#2 */
if( (out = fopen("choice2.dat","rb")) == NULL )
{
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
__ while( !feof(out) )
__ {

```

```

*file_pt = getc( out );
if( ferror( out ) )
{
    printf("Error in reading file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
file_pt++;
}
fclose( out );
manpage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 ); /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Value to B Instruction*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
}
outportb( 0x305,0x01 );
if( enter_choice(choose) == 1 )
break;
else

```

```

{
data_pt = file_pt = pt_const;    /* Enter choice#3 */
if( (out = fopen("choice3.dat","rb")) == NULL )
{
    printf("Error in open file.\n");
    outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
    exit(1);
}
while( !feof(out) )
{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( out );
mannage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outputb( 0x305,0x03 ); /*Select output voice*/
outputb( 0x300,0x48 ); /*Out Value to A Instruction*/
outputb( 0x301,0x3F ); /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{

```

```

    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR, oldhandler );

    }

    outportb( 0x305, 0x01 );

    if( enter_choice(choose) == 1 )

break;

    else

    {
data_pt = file_pt = pt_const;    /* Enter choice#4 */
if( (out = fopen("choice4.dat", "rb")) == NULL )
{
    printf("Error in open file.\n");
    outportb( 0x305, 0x00 ); outportb( 0x305, 0x01 );
    exit(1);
}
while( !feof(out) )
{
    *file_pt =getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305, 0x00 ); outportb( 0x305, 0x01 );
        exit(1);
    }
    file_pt++;

```

```

}

fclose( out );

manage_8259();

oldhandler = getvect( INTR );
setvect( INTR,output );

outportb( 0x305,0x03 ); _ /*Select output voice*/

outportb( 0x300,0x48 ); /*Out Value to A Instruction*/

outportb( 0x301,0x3F ); /*Out Value to B Instruction*/

while( data_pt <= file_pt )
{
    inportb( 0x300 );

    outportb( 0x302, *data_pt );

    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );

    }

    outportb( 0x305,0x01 );

    if( enter_choice(choose) == 1 )

break;

    else

    {

data_pt = file_pt = pt_const;    /* Enter choice#5 */

if( (out = fopen("choice5.dat","rb")) == NULL )

{

    printf("Error in open file.\n");

    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

    exit(1);

}

```

```

while( !feof(out) )
{
    *file_pt =getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}

fclose( out );
mannage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 ); _ /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );
}

outportb( 0x305,0x01 );
if( enter_choice(choose) == 1 )

```

```

break;
}
/**/ if ( choose[0] == '1' )          /* GET MESSAGE */
{
    clrscr(); gotoxy( 30,12 ); printf("Playback...");
    data_pt = file_pt = pt_const;    /* Enter password */
    if( (out = fopen("password.dat","rb")) == NULL )
    {
printf("Error in open file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    while( !feof(out) )
    {
        *file_pt = getc( out );
        if( ferror( out ) )
        {
            printf("Error in reading file.\n");
            outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
            exit(1);
        }
        file_pt++;
    }
    fclose( out );
    manpage_8259();
    oldhandler = getvect( INTR );
    setvect( INTR,output );
    outportb( 0x305,0x03 );    /*Select output voice*/

```

```

    outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
    outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/
    while( data_pt <= file_pt )
    {
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR,oldhandler );
for( i=1; i<=3; i++ )
{
    outportb( 0x305,0x01 );
    recieve_8255( pass,4 );
    if( (pass_pt = fopen(passw,"r")) == NULL )
    {
        printf("Error in open file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    k = 0;
    while( !feof( pass_pt ) )
    {
        real_pass[k] = getc( pass_pt );
        if( ferror( pass_pt ) )
        {
printf("Error in read file.\n");
outportb( 0x305,0x01 ); outportb( 0x305,0x00 );
exit(1);

```

```

    }

    k++;
}

if( password( real_pass,pass ) == 1 ) break;

else if( i<=2 )

{
    /* Password Error. */

    data_pt = file_pt = pt_const;

    if( (out = fopen("check_pa.dat","rb")) == NULL )

    {

        printf("Error in open file check_pa.dat\n");

        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

        exit(1);

    }

    while( !feof(out) )

    {

        *file_pt = getc( out );

        if( ferror( out ) )

        {

printf("Error in reading file.\n");

outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

exit(1);

        }

        file_pt++;

    }

    fclose( out );

    manpage_8259();

    oldhandler = getvect( INTR );

    setvect( INTR,output );

```

```

    outportb( 0x305,0x03 );    /*Select input voice*/
    outportb( 0x300,0x48 );/*Out Value to A Instruct.*/
    outportb( 0x301,0x3F );/*Out Valut to B Instruct.*/
    while( data_pt <= file_pt )
    {
        inportb( 0x300 );
        outportb( 0x302, *data_pt );
        if( check_EOF() == 1 ) break;
    }
    setvect( INTR,oldhandler );
}

}

if( i == 4 )
{
    /* Password error out of program. */
    data_pt = file_pt = pt_const;
    if( (out = fopen("error_pa.dat","rb")) == NULL )
    {
        printf("Error in open file error_pa.dat\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    while( !feof(out) )
    {
        *file_pt =getc( out );
        if( ferrord( out ) )
        {
            printf("Error in reading file.\n");
            outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

```

```

        exit(1);
    }

    file_pt++;
}

fclose( out );
mannage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct.*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
c=0; b=0; a=1;
break;          /* out from loop choose */
}

if( mes_No[j] == 0 )
{
    /* No message */
    data_pt = file_pt = pt_const;
    if( (out = fopen("no_mes.dat","rb")) == NULL )
    {
        printf("Error in open file no_mes.dat\n");
    }
}

```

```

    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
while( !feof(out) )
{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Value to B Instruct.*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );

```

```

    c = 1;                /* goto choose again */
}
else
{
    for( i=1; i<= mes_No[j]; i++ )
    {
        data_pt = file_pt = pt_const;
        creat_filename( name,i );
        if( (out = fopen(name,"rb")) == NULL )
        {
            printf("Error in open file.\n");
            outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
            exit(1);
        }
        while( !feof(out) )
        {
            *file_pt =getc( out );
            if( ferror( out ) )
            {
                printf("Error in reading file.\n");
                outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
                exit(1);
            }
            file_pt++;
        }
        fclose( out );
        manpage_8259();
        oldhandler = getvect( INTR );
    }
}

```

```

setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
}
while( 1 )
{
    data_pt = file_pt = pt_const;    /* check delete */
    if( (out = fopen("delete.dat","rb")) == NULL )
    {
        printf("Error in open file error_pa.dat\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    while( !feof(out) )
    {
        *file_pt = getc( out );
        if( ferror( out ) )
        {
            printf("Error in reading file.\n");
            outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

```

```

exit(1);
    }
    file_pt++;
}
fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct.*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
outportb( 0x305,0x01 );
recieve_8255( check,1 );
if( check[0] == '4' )    /* DELETE MESSAGE */
{
    for( i=1; i<= mes_No[j]; i++ )
    {
creat_filename( name,i );
unlink( name );
    }
    mes_No[j] = 0;

```

```

        a = 1;          /* goto loop choose again */
/* complete delete message */
        data_pt = file_pt = pt_const;
        if( out = fopen("finish.dat","rb") == NULL )
        {
printf("Error in open file error_pa.dat\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
        }
        while( !feof(out) )
        {
/*file_pt = getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
file_pt++;
}
fclose( out );
mannage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 ); /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct*/
while( data_pt <= file_pt )

```

```

    {
inportb( 0x300 );
outportb( 0x302, *data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR, oldhandler );

    c = 1; break;      /* To loop choose again */
}                      /* END OF DELETE MESSAGE */

else if( check[0] == '0' )
{
    /* keep message */

    data_pt = file_pt = pt_const;

    if( (out = fopen("keep.dat", "rb")) == NULL )
    {
printf("Error in open file.\n");
outportb( 0x305, 0x00 ); outportb( 0x305, 0x01 );
exit(1);
    }

    while( !feof(out) )
    {
*file_pt = getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305, 0x00 ); outportb( 0x305, 0x01 );
exit(1);
}
file_pt++;
    }

```

```

fclose( out );

mannage_8259();

oldhandler = getvect( INTR );

setvect( INTR,output );

outportb( 0x305,0x03 ); /*Select output voice*/

outportb( 0x300,0x48 ); /*Out Value to A Instruct*/

outportb( 0x301,0x3F ); /*Out Valut to B Instruct*/

while( data_pt <= file_pt )

{

inportb( 0x300 );

outportb( 0x302, *data_pt );

if( check_EOF() == 1 ) break;

}

setvect( INTR,oldhandler );

break;

}

else

{

data_pt = file_pt = pt_const; /* press error */

if( (out = fopen("preserror.dat","rb")) == NULL )

{

printf("Error in open file.\n");

outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

exit(1);

}

while( !feof(out) )

{

*file_pt =getc( out );

```

```

if( ferrord( out ) )
{
    printf("Error in reading file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}

file_pt++;

    }

    fclose( out );

    manpage_8259();

    oldhandler = getvect( INTR );
    setvect( INTR,output );

    outportb( 0x305,0x03 );    /*Select output voice*/
    outportb( 0x300,0x48 ); /*Out Value to A Instruct*/
    outportb( 0x301,0x3F ); /*Out Valut to B Instruct*/
    while( data_pt <= file_pt )
    {
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR,oldhandler );
    }
}

c = 1;                /* goto loop choose again */

/**/    }                /* END OF GET MESSAGE */

/**/    else if( choose[0] == '2' )        /* LEAVE MESSAGE */

```

```

    {
clrscr();

gotoxy( 30,12); printf("Recording...");

data_pt = file_pt = pt_const;      /* check EOF */
if( (out = fopen("end_file.dat","rb")) == NULL )
{
    printf("Error in open file.\n");

    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

    exit(1);
}

while( !feof(out) )
{
    *file_pt = getc( out );

    if( ferror( out ) )
    {
        printf("Error in reading file.\n");

        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

        exit(1);
    }

    file_pt++;
}

fclose( out );

mannage_8259();

oldhandler = getvect( INTR );

setvect( INTR,output );

outportb( 0x305,0x03 );      /*Select output voice*/

outportb( 0x300,0x48 ); /*Out Value to A Instruction*/

outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/

```

```

while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR, oldhandler );
data_pt = file_pt = pt_const;
count = 0;
mannage_8259();          /* record data */
oldhandler = getvect( INTR );
setvect( INTR, input );
outportb( 0x305, 0x01 );    /*Select input voice*/
outportb( 0x300, 0x48 ); /*Out value to A Instruction*/
outportb( 0x301, 0x3F ); /*Out value to B Instruction*/
while ( count <= 0xFFFF )
{
    inportb(0x300);
    inportb(0x302);
    if ( check_EOF() == 1 ) break;
}

setvect(INTR, oldhandler);
mes_No[j]++;
creat_filename( name, mes_No[j] );
if( (in = fopen(name, "wb")) == NULL )
{
    printf("Error in open file.\n");
    outportb( 0x305, 0x01 ); outportb( 0x305, 0x00 );
}

```

```

    exit(1);
}
for( file_pt=pt_const; file_pt<=data_pt; )
{
    if( putc(*file_pt,in) == EOF )
    {
        printf("Error in writting file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( in );

/* recording complete */
data_pt = file_pt = pt_const;
if( (out = fopen("endleave.dat","rb")) == NULL )
{
    printf("Error in open file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
while( !feof(out) )
{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    }
}

```

```

    exit(1);
}

file_pt++;
}

fclose( out );

mannage_8259();

oldhandler = getvect( INTR );
setvect( INTR,output );

outportb( 0x305,0x03 ); _ /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct.*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );

outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

c=0; b=0; a=1; /* out from loop operate */

/**/ } /* END OF LEAVE MESSAGE */

/**/ else if( choose[0] == '3' ) /* CHANGE PASSWORD */
{
    clrscr(); gotoxy( 30,12 ); printf("Change password...");
    data_pt = file_pt = pt_const; /* Enter old password */
    if( (out = fopen("oldpass.dat","rb")) == NULL )
    {
        printf("Error in open file.\n");
    }
}

```

```

outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
while( !feof(out) )
{
*file_pt = getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
file_pt++;
}
fclose( out );
manpage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );      /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );

```

```

        for( i=1; i<=3; i++ )
        {
outputb( 0x305,0x01 );
recieve_8255( pass,4 );
if( (pass_pt = fopen(passw,"r")) == NULL )
{
    printf("Error in open file.\n");
    outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
    exit(1);
}
k = 0;
while( !feof( pass_pt ) )
{
    real_pass[k] = getc( pass_pt );
    if( ferror( pass_pt ) )
    {
        printf("Error in read file.\n");
        outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
        exit(1);
    }
    k++;
}
if( password( real_pass,pass ) == 1 ) break;
else if( i<=2 )
{
    data_pt = file_pt = pt_const; /* Password error */
    if( (out = fopen("check_pa.dat","rb")) == NULL )
    {

```

```

printf("Error in open file check_pa.dat\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
while( !feof(out) )
{
    *file_pt =getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select input voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct.*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}

```

```

    setvect( INTR,oldhandler );
}

}

if( i == 4 )
{
data_pt = file_pt = pt_const; /* Error out program */
if( (out = fopen("error_pa.dat","rb")) == NULL )
{
    printf("Error in open file error_pa.dat\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
while( !feof(out) )
{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( out );
mannage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/

```

```

outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
c=0; b=0; a=1;
break;          /* out from loop choose */
}

data_pt = file_pt = pt_const; /* Enter new password */
if( (out = fopen("newpass.dat","rb")) == NULL )
{
    printf("Error in open file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}

while( !feof(out) )
{
    *file_pt =getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
}

```

```

}
file_pt++;
}
fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );      /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Value to B Instruction*/
while( data_pt <= file_pt )
{
inportb( 0x300 );
outportb( 0x302, *data_pt );
if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
outportb( 0x305,0x01 );
receive_8255( pass,4 );
if( (pass_pt = fopen(passw,"w")) == NULL )
{
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
for( i=0; i<=3; i++ )
{
if( putc(pass[i],pass_pt) == EOF )

```

```

{
    printf("Error in writting file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}

}

/* record password ready */
data_pt = file_pt = pt_const;
if( (out = fopen("ready.dat","rb")) == NULL )
{
    printf("Error in open file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}

while( !feof(out) )
{
    *file_pt =getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}

fclose( out );
mannage_8259();
oldhandler = getvect( INTR );

```

```

    setvect( INTR,output );
    outportb( 0x305,0x03 );      /*Select output voice*/
    outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
    outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/
    while( data_pt <= file_pt )
    {
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR,oldhandler );

    c = 1;      /* To choose again */
/**/    }      /* END OF CHANGE PASSWORD */
/**/    if( choose[0] == '4' )      /* DELETE MESSAGE */
    {
        data_pt = file_pt = pt_const;      /* Enter password */
        if( (out = fopen("password.dat","rb")) == NULL )
        {
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
        }
        while( !feof(out) )
        {
*file_pt =getc( out );
if( ferrord( out ) )
{
printf("Error in reading file.\n");

```

```

    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
file_pt++;
}
fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );      /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Value to B Instruction*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
for( i=1; i<=3; i++ )
{
    outportb( 0x305,0x01 );
    receive_8255( pass,4 );
    if( (pass_pt = fopen(passw,"r")) == NULL )
    {
        printf("Error in open file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
}

```

```

}

k = 0;

while( !feof( pass_pt ) )
{
    real_pass[k] = getc( pass_pt );
    if( ferror( pass_pt ) )
    {
        printf("Error in read file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    k++;
}

if( password( real_pass,pass ) == 1 ) break;
else if( i<=2 )
{
    data_pt = file_pt = pt_const; /*Password error*/
    if( (out = fopen("check_pa.dat","rb")) == NULL )
    {
        printf("Error in open file check_pa.dat\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    while( !feof(out) )
    {
        *file_pt = getc( out );
        if( ferror( out ) )
        {

```

```

printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
    }
    file_pt++;
}
fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 ); /*Select input voice*/
outportb( 0x300,0x48 );/*Out Value to A Instruct*/
outportb( 0x301,0x3F );/*Out Valut to B Instruct*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
}
}
if( i == 4 )
{
    clrscr();
    data_pt = file_pt = pt_const; /* Error out program */
    if( (out = fopen("error_pa.dat","rb")) == NULL )
    {

```

```

printf("Error in open file error_pa.dat\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
while( !feof(out) )
{
    *file_pt =getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( out );
mannage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct.*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}

```

```

setvect( INTR,oldhandler );

outputb( 0x305,0x00 ); outputb( 0x305,0x01 );

c=0; b=0; a=1;          /* out from loop choose */
}

    while( 1 )
    {
data_pt = file_pt = pt_const;      /* check delete */
if( (out = fopen("ckeckdel.dat","rb")) == NULL )
{
    printf("Error in open file error_pa.dat\n");
    outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
    exit(1);
}
while( !feof(out) )
{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}
fclose( out );

mannage_8259();

oldhandler = getvect( INTR );
setvect( INTR,output );

```

```

outportb( 0x305,0x03 );      /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
outportb( 0x305,0x01 );
recieve_8255( del,1 );
if( del[0] == '4' )      /* delete all old file */
{
    for( i=1; i<= mes_No[j]; i++ )
    {
        creat_filename( name,i );
        unlink( name );
    }
    mes_No[j] = 0;
    data_pt = file_pt = pt_const; /*complete dalete file*/
    if( (out = fopen("finish.dat","rb")) == NULL )
    {
        printf("Error in open file error_pa.dat\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    while( !feof(out) )

```

```

{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}

fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Value to B Instruct.*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );
break;
}          /* complete delete file */

else if( del[0] == '0' )
{

```

```

a = 1;

data_pt = file_pt = pt_const; /*cancel delete file*/
if( (out = fopen("cancel.dat","rb")) == NULL )
{
    printf("Error in open file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}

while( !feof(out) )
{
    *file_pt = getc( out );
    if( ferror( out ) )
    {
        printf("Error in reading file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    file_pt++;
}

fclose( out );
manpage_8259();

oldhandler = getvect( INTR );
setvect( INTR,output );

outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruct.*/
outportb( 0x301,0x3F ); /*Out Valut to B Instruct.*/
while( data_pt <= file_pt )
{

```

```

    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );
break;
}

else
{
    data_pt = file_pt = pt_const;    /* press error */
    if( (out = fopen("preserror.dat","rb")) == NULL )
    {
        printf("Error in open file.\n");
        outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
        exit(1);
    }
    while( !feof(out) )
    {
        *file_pt = getc( out );
        if( ferror( out ) )
        {
            printf("Error in reading file.\n");
            outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
            exit(1);
        }
        file_pt++;
    }
    fclose( out );
}

```

```

mannage_8259());
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );    /*Select output voice*/
outportb( 0x300,0x48 );    /*Out Value to A Instruct*/
outportb( 0x301,0x3F );    /*Out Valut to B Instruct*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302, *data_pt );
    if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
}

    }

    o = 1;                /* Goto loop choose again */
/**/_      }            /* END OF DELETE MESSAGE */
/**/_      else if( choose[0] == '0' )        /* OUT OF PROCESS */
{
data_pt = file_pt = pt_const;
if( (out = fopen("close.dat","rb")) == NULL )
{
    printf("Error in open file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
while( !feof(out) )
{

```

```

*file_pt = getc( out );
if( ferror( out ) )
{
    printf("Error in reading file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}

file_pt++;
}

fclose( out );
manpage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 ); _ /*Select output voice*/
outportb( 0x300,0x48 ); /*Out Value to A Instruction*/
outportb( 0x301,0x3F ); /*Out Value to B Instruction*/
while( data_pt <= file_pt )
{
    inportb( 0x300 );
    outportb( 0x302,*data_pt );
    if( check_EOF() == 1 ) break;
}

setvect( INTR,oldhandler );
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
c=0; b=0; a=1; /* out from loop choose */
/**/_ } /* END LOOP OUT OF PROCESS */
/**/_ else if( choose[0] == '5' ) /* ERROR IN CHOOSE */
{

```

```

    data_pt = file_pt = pt_const;

    if( (out = fopen("presesor.dat","rb")) == NULL )
    {
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
    }

    while( !feof(out) )
    {
*file_pt =getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
file_pt++;
    }

    fclose( out );

    manpage_8259();

    oldhandler = getvect( INTR );
    setvect( INTR,output );

    outportb( 0x305,0x03 );      /*Select output voice*/
    outportb( 0x300,0x48 );     /*Out Value to A Instruction*/
    outportb( 0x301,0x3F );     /*Out Valut to B Instruction*/

    while( data_pt <= file_pt )
    {
inportb( 0x300 );

```

```

outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR,oldhandler );

    c = 1;

/**/    }                /* END OF ERROR IN CHOOSE */
/**/    else if( choose[0] == '6' )        /* ERROR IN CHOOSE */
    {
        data_pt = file_pt = pt_const;
        if( (out = fopen("preserror.dat","rb")) == NULL )
        {
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
        }
        while( !feof(out) )
        {
*file_pt = getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
file_pt++;
        }
        fclose( out );
        manpage_8259();

```

```

oldhandler = getvect( INTR );
setvect( INTR,output );
outputb( 0x305,0x03 );      /*Select output voice*/
outputb( 0x300,0x48 );      /*Out Value to A Instruction*/
outputb( 0x301,0x3F );      /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{
inportb( 0x300 );
outputb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );
c = 1;
/**/      }      /* END OF ERROR IN CHOOSE */
/**/      else if( choose[0] == '7' )      /* ERROR IN CHOOSE */
{
data_pt = file_pt = pt_const;
if( (out = fopen("presesor.dat","rb")) == NULL )
{
printf("Error in open file.\n");
outputb( 0x305,0x00 ); outputb( 0x305,0x01 );
exit(1);
}
while( !feof(out) )
{
*file_pt =getc( out );
if( ferror( out ) )
{

```

```

printf("Error in reading file.\n");

outportb( 0x305,0x00 ); outportb( 0x305,0x01 );

exit(1);
}

file_pt++;

}

fclose( out );

manage_8259();

oldhandler = getvect( INTR );

setvect( INTR,output );

outportb( 0x305,0x03 );      /*Select output voice*/

outportb( 0x300,0x48 );      /*Out Value to A Instruction*/

outportb( 0x301,0x3F );      /*Out Valut to B Instruction*/

while( data_pt <= file_pt )

{

inportb( 0x300 );

outportb( 0x302,*data_pt );

if( check_EOF() == 1 ) break;

}

setvect( INTR,oldhandler );

c = 1;

/**/      }      /* END OF ERROR IN CHOOSE */

/**/      else if( choose[0] == '8' )      /* ERROR IN CHOOSE. */

{

data_pt = file_pt = pt_const;

if( (out = fopen("presesor.dat","rb")) == NULL )

{

printf("Error in open file.\n");

```

```

outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
while( !feof(out) )
{
*file_pt = getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
file_pt++;
}
fclose( out );
manage_8259();
oldhandler = getvect( INTR );
setvect( INTR,output );
outportb( 0x305,0x03 );      /*Select output voice*/
outportb( 0x300,0x48 );      /*Out Value to A Instruction*/
outportb( 0x301,0x3F );      /*Out Valut to B Instruction*/
while( data_pt <= file_pt )
{
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
}
setvect( INTR,oldhandler );

```

```

        c = 1;

/**/    }                                /* END OF ERROR IN CHOOSE */

/**/    else if( choose[0] == '9' )        /* ERROR IN CHOOSE */
    {
        data_pt = file_pt = pt_const;
        if( (out = fopen("preserror.dat","rb")) == NULL )
        {
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
        }
        while( !feof(out) )
        {
*file_pt = getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}
file_pt++;
        }
        fclose( out );
        manpage_8259();
        oldhandler = getvect( INTR );
        setvect( INTR,output );
        outportb( 0x305,0x03 );          /*Select output voice*/
        outportb( 0x300,0x48 );          /*Out Value to A Instruction*/

```

```

        outportb( 0x301,0x3F );    /*Out Valut to B Instruction*/
        while( data_pt <= file_pt )
        {
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
        }

        setvect( INTR,oldhandler );

        c = 1;
/**/    }                                /* END OF ERROR IN CHOOSE */
/**/    else if( choose[0] == '*' )      /* ERROR IN CHOOSE */
    {
        data_pt = file_pt = pt_const;
        if( (out = fopen("preserror.dat","rb")) == NULL )
        {
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
        }

        while( !feof(out) )
        {
*file_pt = getc( out );
if( ferror( out ) )
{
printf("Error in reading file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
}

```

```

file_pt++;
    }

    fclose( out );

    manpage_8259();

    oldhandler = getvect( INTR );

    setvect( INTR,output );

    outportb( 0x305,0x03 );      /*Select output voice*/

    outportb( 0x300,0x48 );     /*Out Value to A Instruction*/

    outportb( 0x301,0x3F );     /*Out Valut to B Instruction*/

    while( data_pt <= file_pt )
    {
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR,oldhandler );

    c = 1;

/**/      }      /* END OF ERROR IN CHOOSE */

/**/      else if( choose[0] == '#' )      /* ERROR IN CHOOSE */
    {
        data_pt = file_pt = pt_const;

        if( (out = fopen("presesor.dat","rb")) == NULL )
        {
printf("Error in open file.\n");
outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
exit(1);
        }

        while( !feof(out) )

```

```

    {
*file_pt = getc( out );
if( ferror( out ) )
{
    printf("Error in reading file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
file_pt++;
    }

    fclose( out );
    manpage_8259();
    oldhandler = getvect( INTR );
    setvect( INTR,output );
    outportb( 0x305,0x03 );      /*Select output voice*/
    outportb( 0x300,0x48 );     /*Out Value to A Instruction*/
    outportb( 0x301,0x3F );     /*Out Valut to B Instruction*/
    while( data_pt <= file_pt )
    {
inportb( 0x300 );
outportb( 0x302,*data_pt );
if( check_EOF() == 1 ) break;
    }

    setvect( INTR,oldhandler );
    c = 1;
/**/    }          /* END OF ERROR IN CHOOSE */
    }          /* END OF LOOP CHOOSE */
}          /* check hook */

```

```

else if( kbhit() )
{
    clrscr();
    gotoxy( 20,12 );
    printf("DO YOU WANT TO STOP VOICE MALIBOX SYSTEM. (Y/N)");
    chk = 1;
    while( chk )
    {
        confirm = getch();
        if( toupper(confirm) == 'Y' )
            { chk = 0; b = 0; a = 0; }
        else if( toupper(confirm) == 'N' )
            { chk = 0; b = 0; a = 1; }
        else  chk = 1;
    }
}

}          /* END OF LOOP OPERATING (b)*/

}          /* END OF LOOP HANDCHECKING (a)*/

farfree( pt_const );

if( (mes = fopen("message.dat","w")) == NULL )    /*Save mes_No[]*/
{
    printf("Error in open file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}

for( i=0; i<=3; i++ )
{

```

```
if( putc(mes_No[i],mes) == EOF )      /* record message No. */
{
    printf("Error in writting file.\n");
    outportb( 0x305,0x00 ); outportb( 0x305,0x01 );
    exit(1);
}
}
return 0;
}
```

ภาคผนวก ข Data Sheet ของ IC ต่าง ๆ ที่ใช้ในวงจร

MT 8870 ไอซีถอดรหัสความถี่โทรศัพท์ (Integrated DTMF Receiver)

การถอดรหัสความถี่โทรศัพท์ หมายถึง การแปลงสัญญาณความถี่ ซึ่งเกิดจากการกดปุ่มตัวเลข ของโทรศัพท์ชนิดกดปุ่ม (ชนิด Tone หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิทัล ซึ่งไอซี MT 8870 ใช้แปลงความถี่ DTMF ให้เป็นเลขฐานสอง ขนาด 4 บิต

สมัยก่อนการออกแบบวงจรถอดรหัสความถี่โทรศัพท์ มักใช้ไอซีจำพวกเฟลลอปซึ่งสร้างปัญหาในการทำงานมากมาย เช่น เรื่องความถี่ที่เปลี่ยนแปลงไป การปรับแต่งวงจรขนาดของไอซีที่ใหญ่ เพราะต้องใช้ไอซีจำนวนมาก เป็นต้น ปัจจุบัน ด้วยเทคโนโลยีขั้นสูง มีหลายบริษัทที่ผลิตไอซีสำหรับใช้งาน ถอดรหัสความถี่โทรศัพท์ DTMF โดยเฉพาะ ออกสู่ท้องตลาด เช่น MITEL ได้แก่อิซีเบอร์ MT8870 SAMSUNG เบอร์ KT3170 เป็นต้น

คุณสมบัติของ MT8870

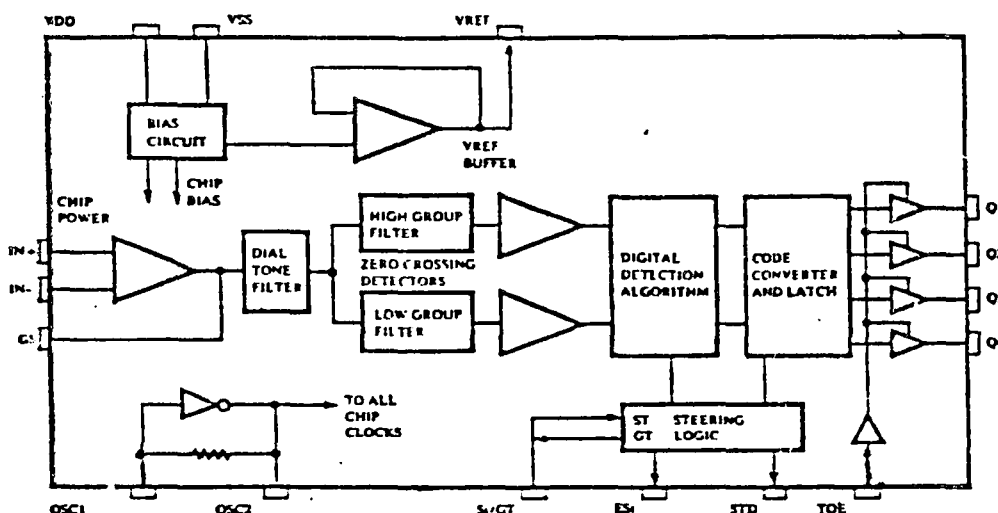
- เป็นตัวรับ และถอดรหัสความถี่ (DTMF Receiver)
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดไทม์ (Guard time)
- เป็นไอซีคุณภาพสูง

การนำ MT8870 ไปใช้งาน

- นำไปใช้งานด้านรีโมทคอนโทรล
- เครื่องป้องกันโทรศัพท์ทางไกล
- ใช้ในงานเกี่ยวกับเครดิตการ์ด
- ใช้งานร่วมกับคอมพิวเตอร์
- ใช้ในเครื่องชุมสายขนาดเล็ก หรือ PABX
- ใช้กับงานทางด้านโทรศัพท์ทั่วไป
- เครื่องกันขโมย
- การควบคุมอุปกรณ์ทางโทรศัพท์
- ใช้ทำเครื่องสอบถามทางโทรศัพท์

โครงสร้างของ MT8870

ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังก์ชันทางดิจิทัล เป็นไอซีที่สร้างขึ้นโดยใช้เทคโนโลยี *ISO2-CMOS* ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิทช์คาปาซิเตอร์ฟิลเตอร์เตอร์พลเตอร์สำหรับกรองความถี่สูงและกรองความถี่ต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิทัลเพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิต และเช็คช่วงเวลาสัญญาณเข้ามาส่วนภาคอินพุตเป็นออปแอมป์ซึ่งสามารถปรับอัตราขยายได้โดย ต่ออุปกรณ์ภายนอกเอาท์พุต เป็นวงจรแลทช์ 3 สถานะ รูป A.1 แสดงขาของ MT8870 ส่วนรูป A.2 แสดงโครงสร้างภายในของ MT8870



รูปที่ A.2 แสดงโครงสร้างภายในของ MT8870

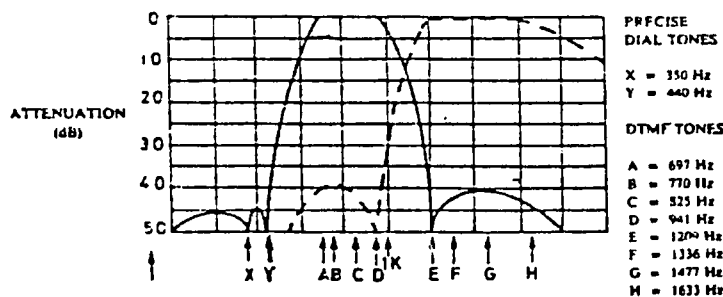
ฟังก์ชันการทำงานภายในของ MT8870

ภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

- ภาคกรองสัญญาณความถี่ (filter section)
- ภาคถอดรหัส (decoder section)
- ภาคตรวจสอบสัญญาณ (steering circuit)
- ภาคขยายสัญญาณความแตกต่าง (differential input)
- ภาคกำเนิดความถี่ (oscillator)

1. ภาคกรองสัญญาณความถี่ (filter section)

ในส่วนนี้ จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิดสวิตช์คาปาซิเตอร์ (six order switched capacitor band pass filter) ซึ่งความถี่ที่แยกได้ มี 2 ช่วง คือช่วงความถี่สูงและช่วงความถี่ต่ำ



รูปที่ A.3 แสดงความถี่ที่ได้จากภาคกรองความถี่

2. ภาคถอดรหัส (decoder section)

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคการนับแบบดิจิตอลและมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา Est (early steering) ก็จะแอกทีฟ สำหรับค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ นั้น แสดงในรูปที่ A.4

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

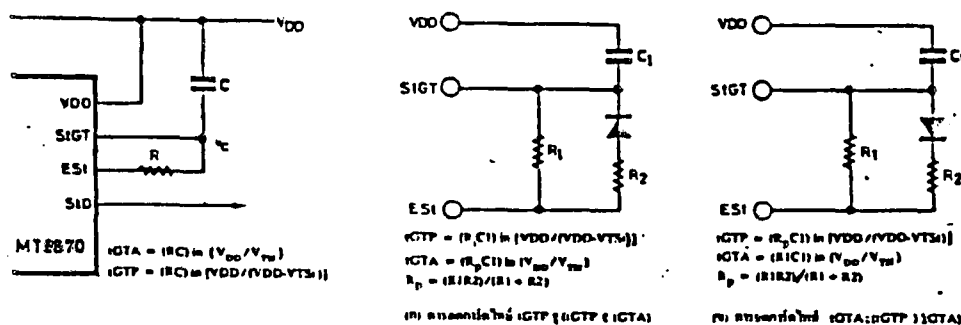
รูปที่ A.4 แสดงค่าที่ถอดรหัสได้จากความถี่ที่ถูกต้อง

3. ภาคตรวจสอบสัญญาณ (steering circuit)

ก่อนที่จะมีการถอดรหัสความถี่ ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลานานเท่าใด สามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา Est จะเป็น "High" ทำให้ V_C สูงขึ้น ตัวเก็บประจุ C จะคายประจุ ทำให้แรงดัน V_C สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสจึงจะถอดรหัส ออกเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงาน ถ้าได้ดูจากแผนภูมิ หรือ ไทม์มิงไดอะแกรม (timing diagram) ในรูป A.9 จะเข้าใจได้ดียิ่งขึ้น

ถูกต้องหรือพูดได้ว่าเวลาที่เรที่ตั้งไว้โดย RC ก็คือ Guard Time นั้นเอง เมื่อสัญญาณความถี่เข้ามานานเท่าหรือมากกว่าเวลาที่ตั้งไว้ จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่า ก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลา และการคำนวณเวลาดูได้จากรูปที่ A.5

A.5



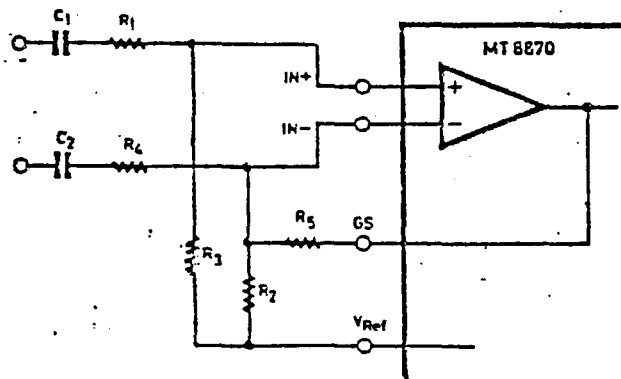
รูปที่ A.5 แสดงวงจรตรวจสอบสัญญาณอย่างง่ายและแสดงการกำหนดเวลาการ์ดใหม่

4. ภาควิชาสัญญาณความแตกต่าง (differential input)

วงจรส่วนอินพุทของ MT8870 เป็นภาควิชาออปแอมป์ ที่สามารถปรับอัตราขยาย โดยต่อวงจรภายนอกเพิ่มเข้าไป รูปที่ A.6 แสดงการต่อวงจรภายนอกเข้ากับอินพุท ซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอินพุท และอิมพีแดนซ์ ได้ดังนี้

$$\text{อัตราขยาย } (A_v \text{ diff}) = R_5 / R_1$$

$$\text{อินพุทอิมพีแดนซ์ } (Z_{in} \text{ diff}) = 2\sqrt{(R_1)^2 + (1/\omega_c)^2}$$



ลักษณะการต่อวงจรอินพุต

$$C_1 = C_2 = 10 \text{ nF}$$

$$R_1 = R_2 = R_3 = 100 \text{ K}\Omega \quad \text{ค่าผิดพลาด } \pm 1\%$$

$$R_4 = 60 \text{ K}\Omega, R_5 = 37.5 \text{ K}\Omega \quad \text{ค่าผิดพลาด } \pm 5\%$$

$$R_6 = \frac{R_1 R_2}{R_1 + R_2}$$

$$\text{อัตราขยายแรงดัน (A_v diff)} = \frac{R_6}{R_1}$$

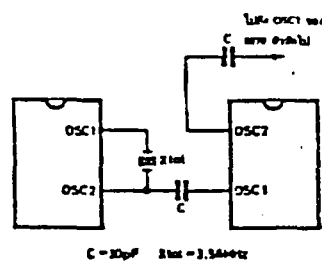
อินพุตที่สมดุล

$$(Z_{INDIFF}) = 2 \sqrt{R_1^2 + \left(\frac{1}{\omega C}\right)^2}$$

รูปที่ A.6 แสดงการต่อวงจรภาคอินพุต

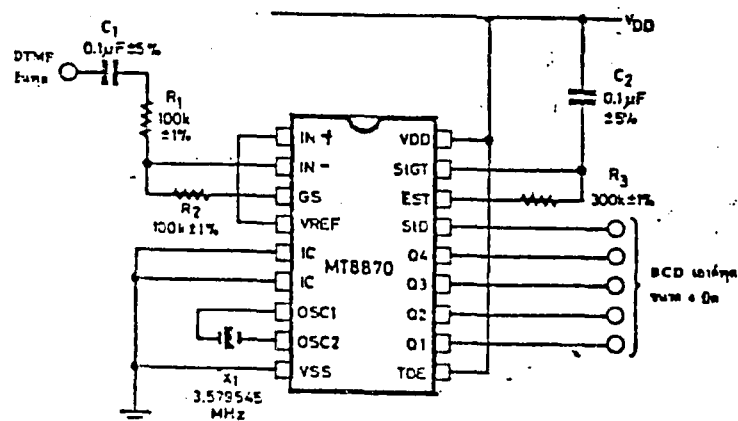
5. ภาคกำเนิดความถี่ (oscillator)

ในภาคนี้ ภายในไอซีจะมีวงจรเวลาอยู่ภายใน เพียงแต่ต่อแร่คริสตอลขนาด 3.58 MHz ก็สามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่ แสดงในรูปที่ A.7



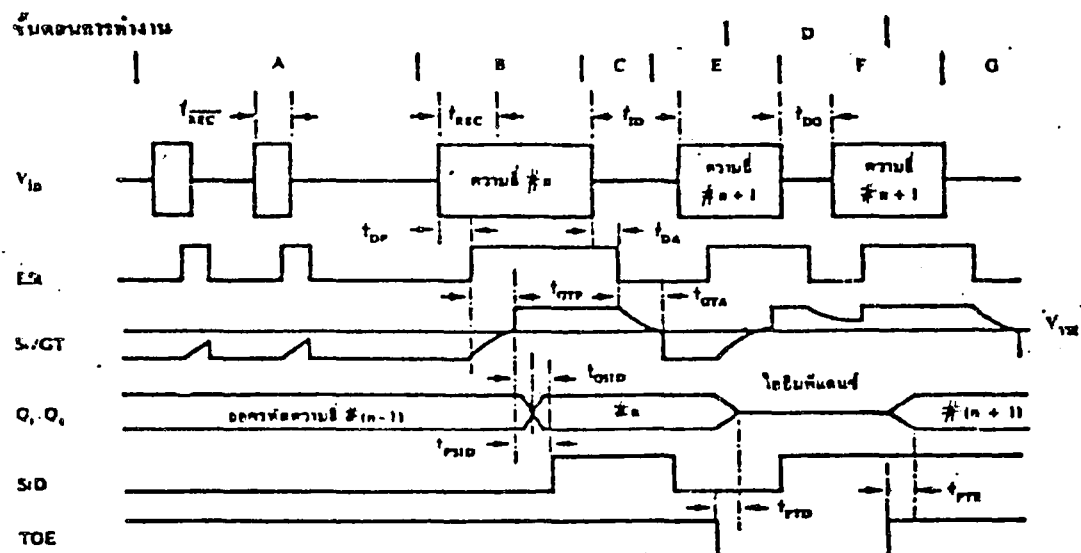
$$C = 30 \text{ pF} \quad f_{osc} = 3.58 \text{ MHz}$$

รูปที่ A.7 การต่อวงจรผลิตความถี่



รูปที่ A.8 แสดงวงจรใช้งานเบื้องต้นของ MT8870

คำอธิบาย Timing Diagram ดูที่หน้าถัดไป



รูปที่ A.9 แสดงแผนภูมิเวลา(Timing Diagram) ของ MT8870

อธิบายขั้นตอนการทำงาน

- A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาต์พุตไม่เปลี่ยน
- B - ความถี่ # n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแสดงไว้ที่เอาต์พุต
- C - จบความถี่ # n ช่วงห่างถูกต้อง เอาต์พุตยังคงแสดงอยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่
- D - เอาต์พุตเปลี่ยนเป็นไฮอิมพีแดนซ์
- E - ความถี่ # n + 1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแสดงไว้
- F - ความถี่ # n + 1 หายไป ช่วงห่างไม่ถูกต้อง เอาต์พุตยังคงแสดงอยู่
- G - จบความถี่ # n + 1 ช่วงห่างถูกต้อง เอาต์พุตยังคงแสดงอยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

- V_{in} - สัญญาณความถี่ DTMF ที่เข้ามา
- ESi - Early Steering output ใช้แสดงความถี่ที่ถูกต้อง
- Si/GT - Steering input/Guard Time output สำหรับต่อกับ RC ภายนอก
- Q_1-Q_4 - เอาต์พุต BCD ขนาด 4 บิต
- SiD - Delayed Steering output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ
- TOE - Tone Output Enable (input) ใช้ควบคุม Q_1-Q_4 ให้เป็นไฮอิมพีแดนซ์
- t_{REC} - คาบเวลานานสุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง
- t_{REC} - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง
- t_{ID} - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ
- t_{DO} - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง
- t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง
- t_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง
- t_{GTP} - ทรานส์ไทม์ของการปรากฏความถี่ DTMF
- t_{GTA} - ทรานส์ไทม์ของการหายไปของความถี่ DTMF

รูปที่ 10 แสดงคำอธิบาย Timing Diagram ของ MT8870

เกี่ยวกับ DTMF

การเลือกความถี่ของสัญญาณ DTMF นั้น มิใช่มาจากการสุ่มตัวเลข หรือกำหนดตามอำเภอใจ แต่ได้มาจากการศึกษาถึงผลดี - ผลเสียอย่างรอบคอบ โดยเราจะเลือกความถี่ที่อยู่ในย่านความถี่ของสัญญาณโทรศัพท์ (300 - 3400Hz) และจะต้องไม่มีโทนที่เป็น Harmonic หรือใกล้เคียง Harmonic ซึ่งกันและกัน และปกติ สัญญาณความถี่ DTMF ต้องเป็น pure sine wave จึงมีจุด

ฮาร์โมนิก ที่ N เกิดไปตรงกับอีกความถี่หนึ่ง ในกรณีที่เกิดความเพี้ยนจากวงจรผลิตความถี่ ไม่ให้ความถี่เป็น pure sine wave จริง

ความถี่ DTMF ที่เรานำมาใช้ ทดลองค้นคว้าโดย Bell Telephone System เมื่อปี ค.ศ. 1950 ความถี่ต่าง ๆ ที่ใช้งานมีดังนี้

1	2	3	A	697
4	5	6	B	770
7	8	9	C	852
*	0	#	D	941
1209	1336	1477	1633	ความถี่ (Hz)

ในการผลิตความถี่สัญญาณ DTMF นั้น ในไอซีผลิตสัญญาณ DTMF รุ่นใหม่ๆ จะผลิตความถี่มาตรฐานขึ้นมาค่าหนึ่ง แล้วทำการหารความถี่ลงไป และจะใช้วงจร counter กับ clock มาช่วย ตัวอย่างเช่น IC Motorola MC14403 ใช้ความถี่ 500 kHz จาก XTAL ใช้ไฟเลี้ยงไอซี 1.4 V จาก Local loop

สำหรับไอซี MT8870 ที่เรานำมาใช้งานผลิตจาก XTAL ที่ใช้เป็น Chroma Crystal ของ TV คือความถี่ 3.579545 MHz นอกจากจะหาง่ายและราคาถูกแล้ว ยังได้ค่าความผิดพลาดน้อยที่สุด เมื่อหารด้วยค่า N ต่างๆ เช่น ค่า $N = 4672$ MHz เราจะได้ความถี่ 766.17 MHz ซึ่งใกล้เคียงสัญญาณความถี่ DTMF มาตรฐานคือ 770 MHz ความถี่ที่เราผลิต เพื่อนำมาตรวจสอบ สำหรับการถอดรหัส DTMF ทางอินพุตนำมาสรุปได้ค่าต่างๆ ทั้งหมดดังนี้

เมื่อ

$$f_{osc} = 3,579,545$$

$$f_{actual} = f_{osc} / N$$

<i>input</i>	<i>Nominal</i> (Hz)	<i>Divider ratio</i> N	<i>Actual</i> (Hz)	<i>Deviation</i> (%)
R_1	697	5,120	699.13	+0.3
R_2	770	4,672	766.17	-0.5
R_3	852	4,224	847.43	-0.5
R_4	941	3,776	947.97	+0.74
C_1	1,209	2,944	1,215.88	+0.57
C_2	1,336	2,688	1,331.68	-0.32
C_3	1,477	2,432	1,471.85	-0.35
C_4	1,633	2,176	1,645.01	+0.74



Silicon Gate MOS 8255

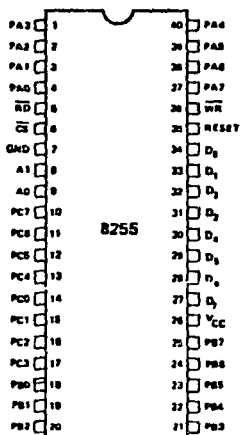
PROGRAMMABLE PERIPHERAL INTERFACE

- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with MCS™-8 and MCS™-80 Microprocessor Families
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40 Pin Dual In-Line Package
- Reduces System Package Count

The 8255 is a general purpose programmable I/O device designed for use with both the 8008 and 8080 microprocessors. It has 24 I/O pins which may be individually programmed in two groups of twelve and used in three major modes of operation. In the first mode (Mode 0), each group of twelve I/O pins may be programmed in sets of 4 to be input or output. In Mode 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining four pins three are used for handshaking and interrupt control signals. The third mode of operation (Mode 2) is a Bidirectional Bus mode which uses 8 lines for a bidirectional bus, and five lines, borrowing one from the other group, for handshaking.

Other features of the 8255 include bit set and reset capability and the ability to source 1mA of current at 1.5 volts. This allows darlington transistors to be directly driven for applications such as printers and high voltage displays.

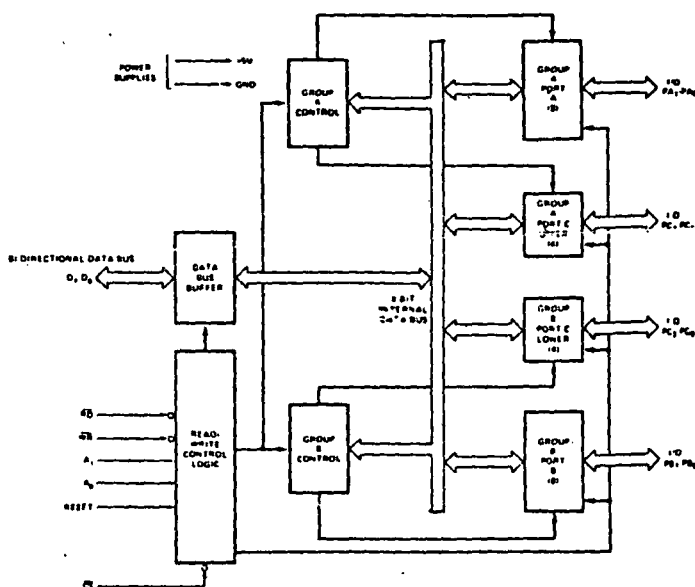
PIN CONFIGURATION



PIN NAMES

D ₇ -D ₀	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
AD, A1	PORT ADDRESS
PA7-PA0	PORT A (8BIT)
PB7-PB0	PORT B (8BIT)
PC7-PC0	PORT C (8BIT)
Vcc	+5 VOLTS
GND	0 VOLTS

8255 BLOCK DIAGRAM



SILICON GATE MOS 8255

8255 BASIC FUNCTIONAL DESCRIPTION

General

The 8255 is a Programmable Peripheral Interface (PPI) device designed for use in 8080 Microcomputer Systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the 8080 system bus. The functional configuration of the 8255 is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state, bi-directional, eight bit buffer is used to interface the 8255 to the 8080 system data bus. Data is transmitted or received by the buffer upon execution of INPUT or OUTPUT instructions by the 8080 CPU. Control Words and Status information are also transferred through the Data Bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the 8080 CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select: A "low" on this input pin enables the communication between the 8255 and the 8080 CPU.

($\overline{RD}$)

Read: A "low" on this input pin enables the 8255 to send the Data or Status information to the 8080 CPU on the Data Bus. In essence, it allows the 8080 CPU to "read from" the 8255.

($\overline{WR}$)

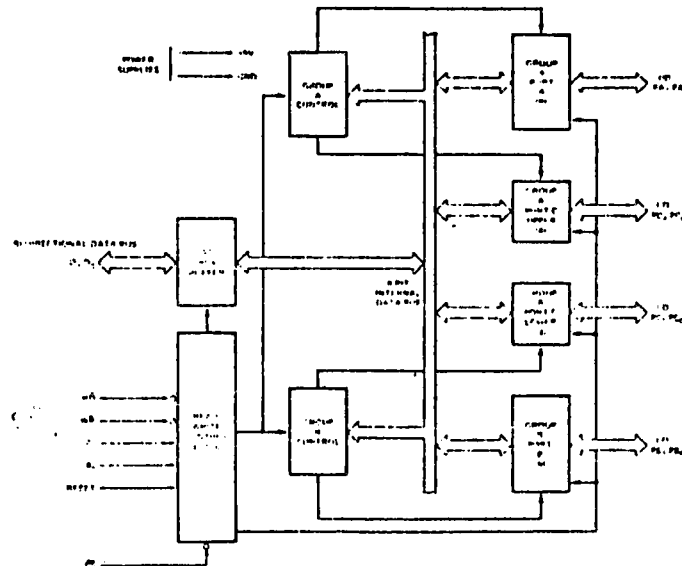
Write: A "low" on this input pin enables the 8080 CPU to write Data or Control words into the 8255.

(A_0 and A_1)

Port Select 0 and Port Select 1: These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the Control Word Register. They are normally connected to the least significant bits of the Address Bus (A_0 and A_1).

8255 BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A = DATA BUS
0	1	0	1	0	PORT B = DATA BUS
1	0	0	1	0	PORT C = DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS = PORT A
0	1	1	0	0	DATA BUS = PORT B
1	0	1	0	0	DATA BUS = PORT C
1	1	1	0	0	DATA BUS = CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS = 3-STATE
1	1	0	1	0	ILLEGAL CONDITION



8255 Block Diagram

8255 DATA SHEET

SILICON GATE MOS 8255

(RESET)

Reset: A "high" on this input clears all internal registers including the Control Register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the 8080 CPU "outputs" a control word to the 8255. The control word contains information such as "mode", "bit set", "bit reset" etc. that initializes the functional configuration of the 8255.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A — Port A and Port C upper (C7-C4)

Control Group B — Port B and Port C lower (C3-C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

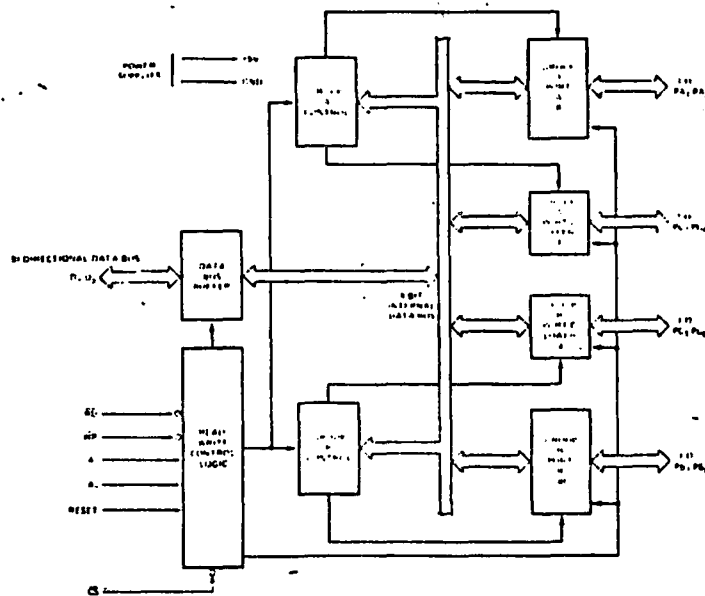
The 8255 contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255.

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch.

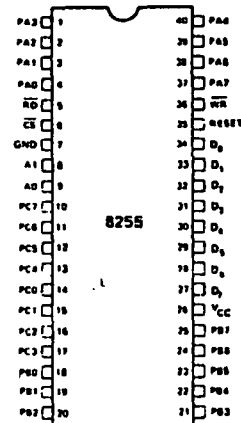
Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with Ports A and B.

8255 BLOCK DIAGRAM



PIN CONFIGURATION



PIN NAMES

D ₇ -D ₀	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
Vcc	+5 VOLTS
GND	0 VOLTS

8255 DATA SHEET

appendix C-4

SILICON GATE MOS 8255

8255 DETAILED OPERATIONAL DESCRIPTION

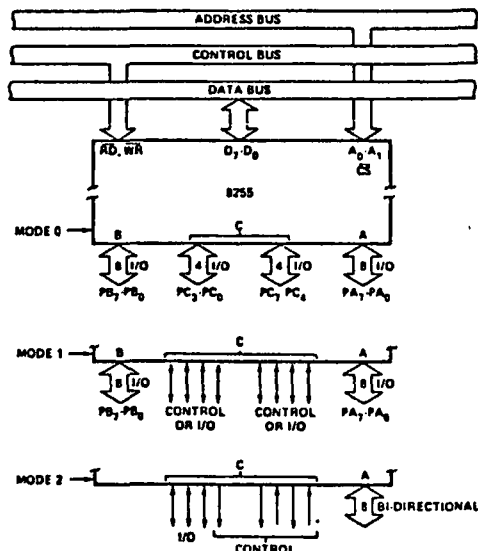
Mode Selection

There are three basic modes of operation that can be selected by the system software:

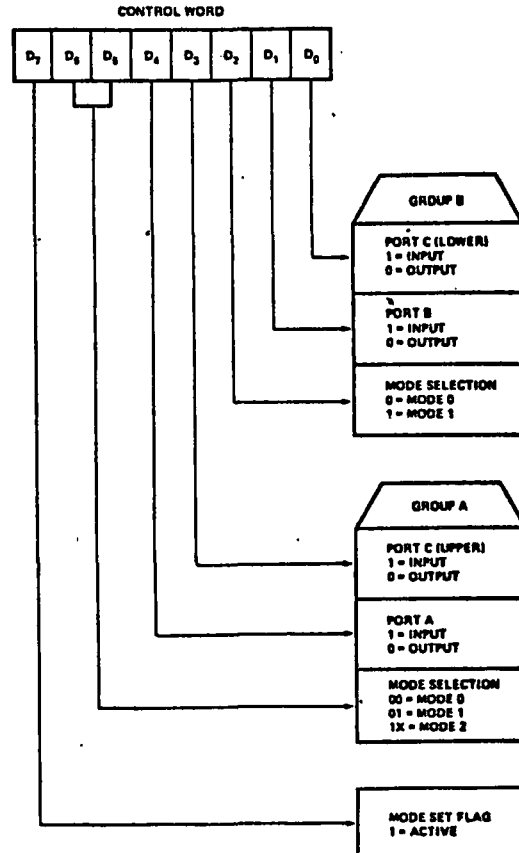
- Mode 0 – Basic Input/Output
- Mode 1 – Strobed Input/Output
- Mode 2 – Bi-Directional Bus

When the RESET input goes "high" all ports will be set to the Input mode (i.e., all 24 lines will be in the high impedance state). After the RESET is removed the 8255 can remain in the Input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single OUTput instruction. This allows a single 8255 to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.



Basic Mode Definitions and Bus Interface



Mode Definition Format

The Mode definitions and possible Mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255 has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

SILICON GATE MOS 8255

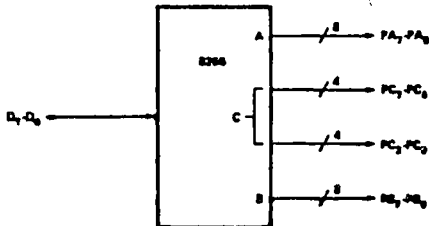
MODE 0 PORT DEFINITION CHART

A		B		GROUP A			GROUP B	
D ₄	D ₃	D ₁	D ₀	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 CONFIGURATIONS

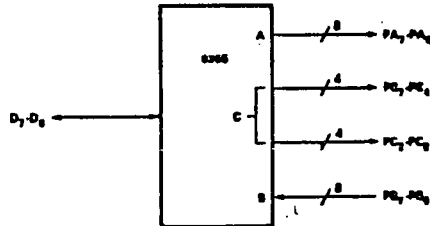
CONTROL WORD #0

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	0	0



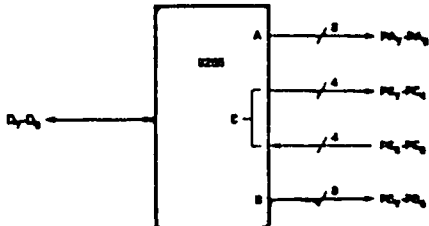
CONTROL WORD #2

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	1	0



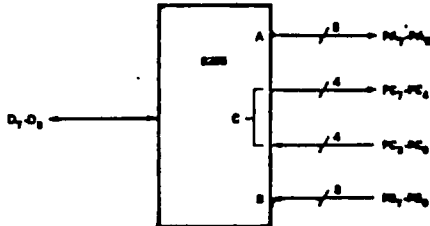
CONTROL WORD #1

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	0	1



CONTROL WORD #3

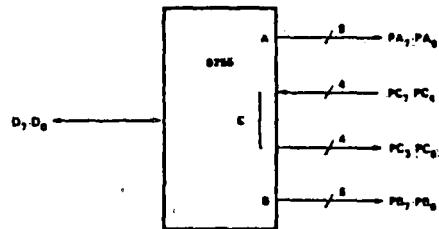
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	1	1



SILICON GATE MOS 8255

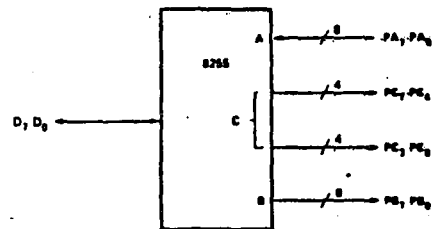
CONTROL WORD #4

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	0	0



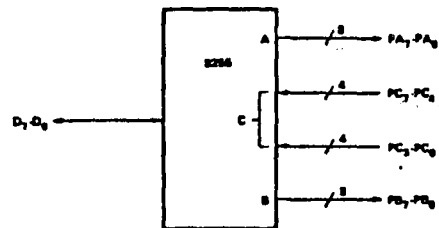
CONTROL WORD #8

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	0	0



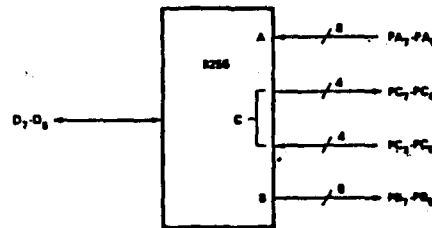
CONTROL WORD #6

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	0	1



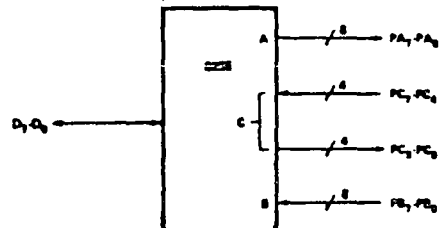
CONTROL WORD #9

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	0	1



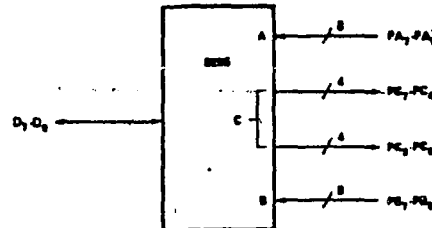
CONTROL WORD #5

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	1	0



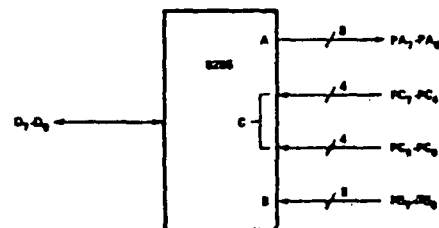
CONTROL WORD #10

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	1	0



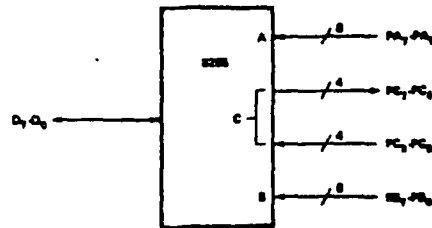
CONTROL WORD #7

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	1	1



CONTROL WORD #11

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	1	1



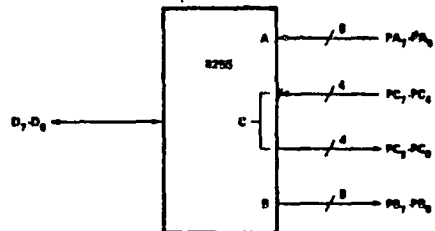
8255 DATA SHEET

appendix C-7

SILICON GATE MOS 8255

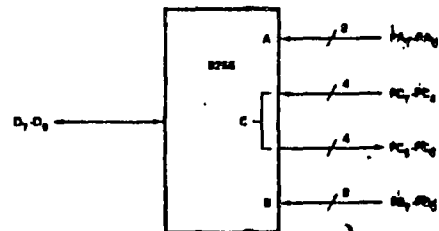
CONTROL WORD #12

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	0	0



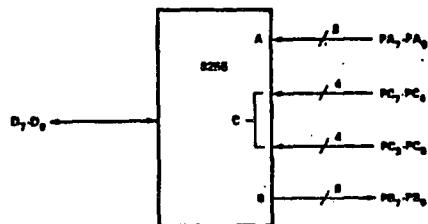
CONTROL WORD #14

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	1	0



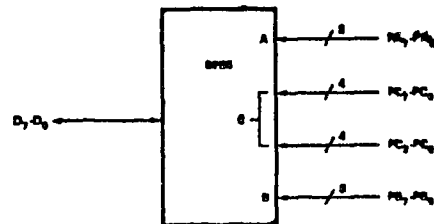
CONTROL WORD #13

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	0	1



CONTROL WORD #15

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	1	1



8255 DATA SHEET

appendix C-8

NOTES:



ISO²-CMOS MT8870C/MT8870C-1
Integrated DTMF Receiver

9161-002-054-NA ISSUE 1 November 1987

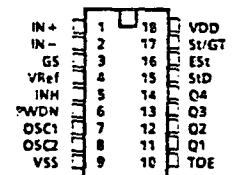
Features

- Complete DTMF Receiver
- Low Power Consumption
- Internal Gain Setting Amplifier
- Adjustable Guard Time
- Central Office Quality
- Power-down Mode
- Inhibit Mode

Applications

- Receiver System for British Telecom (BT) or CEPT Spec (MT8870C-1)
- Paging Systems
- Repeater Systems/Mobile Radio
- Credit Card System
- Remote Control
- Personal Computers
- Telephone Answering Machine

Pin Connections



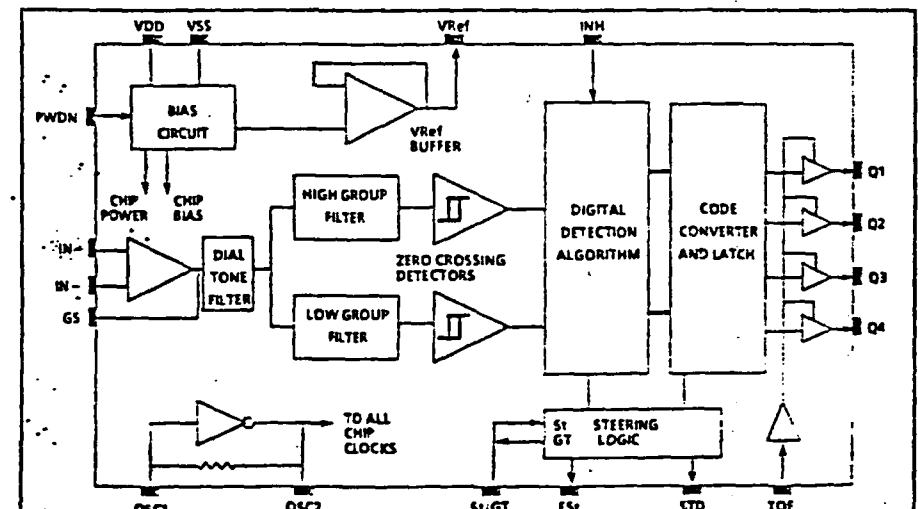
Ordering Information

MT8870CE/MT8870CE-1 Plastic Dip
MT8870CC/MT8870CC-1 Cerdip
MT8870CS/MT8870CS-1 SOIC
-40°C to +85°C

Description

The MT8870C/MT8870C-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions, fabricated in Mitel's double poly ISO²-CMOS technology. The filter section uses switched capacitor techniques for

high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone-pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.



not guaranteed and not subject to change without notice. VDD = 5.0V

1. Referenced to lowest level frequency component in tone
2. For guard time only

Absolute Maximum Ratings¹

	Parameter	Symbol	Min	Max	Units
1	DC Power Supply Voltage	V _{DD}		6	V
2	Voltage on any pin	V _i	V _{SS} - 0.3	V _{DD} + 0.3	V
3	Current at any pin (other than supply)	I _i		10	mA
4	Storage temperature	T _{STG}	-65	-150	°C
5	Package power dissipation	P _D		1000	mW

¹ Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied. Derate above 75°C at 16 mW/°C. All leads soldered to board.

Recommended Operating Conditions - Voltages are with respect to ground (V_{SS}) unless otherwise stated.

	Parameter	Sym	Min	Typ ¹	Max	Units	Test Conditions
1	DC Power Supply Voltage	V _{DD}	4.75	5.0	5.25	V	
2	Operating Temperature	T _O	-40		+85	°C	
3	Crystal/Clock Frequency	f _c		3.579545		MHz	
4	Crystal/Clock Freq. Tolerance	Δf _c		±0.1		%	

¹ Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

DC Electrical Characteristics - V_{DD} = 5.0V ± 5%, V_{SS} = 0V, -40°C ≤ T_O ≤ +85°C, unless otherwise stated.

	Characteristics	Sym	Min	Typ ¹	Max	Units	Test Conditions
1	Standby supply current	I _{DDQ}			100	μA	PWDN = V _{DD}
2	Operating supply current	I _{DD}		3.0	9.0	mA	
3	Power consumption	P _O		15	50	mW	f _c = 3.579545 MHz
4	High level input	V _{IH}	3.5			V	V _{DD} = 5.0V
5	Low level input voltage	V _{IL}			1.5	V	V _{DD} = 5.0V
6	Input leakage current	I _{IH} /I _{IL}		0.1		μA	V _{IH} = V _{SS} or V _{DD}
7	Pull up (source) current	I _{SO}		7.5	20	μA	TOE (pin 10) = 0, V _{DD} = 5.0V
8	Pull down (sink) current	I _{SI}		15	45	μA	INH = 5.0V, PWDN = 5.0V, V _{DD} = 5.0V
9	Input impedance (IN+, IN-)	R _{IN}		10		MΩ	@ 1 kHz
10	Steering threshold voltage	V _{TS1}	2.2		2.5	V	V _{DD} = 5.0V
11	Low level output voltage	V _{OL}			V _{SS} - 0.03	V	No load
12	High level output voltage	V _{OH}	V _{DD} - 0.03			V	No load
13	Output low (sink) current	I _{OL}	1.0	2.5		mA	V _{OUT} = 0.4 V
14	Output high (source) current	I _{OH}	0.4	0.8		mA	V _{OUT} = 4.6 V
15	V _{REF} output voltage	V _{REF}	2.4		2.7	V	No load, V _{DD} = 5.0V
16	V _{REF} output resistance	R _{OR}		10		kΩ	

¹ Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

Operating Characteristics - V_{DD} = 5.0V ± 5%, V_{SS} = 0V, -40°C ≤ T_O ≤ +85°C, unless otherwise stated. Gain Setting Amplifier

	Characteristics	Sym	Min	Typ ¹	Max	Units	Test Conditions
1	Input leakage current	I _{IN}			100	nA	V _{SS} ≤ V _{IN} ≤ V _{DD}
2	Input resistance	R _{IN}	10			MΩ	
3	Input offset voltage	V _{OS}			25	mV	
4	Power supply rejection	PSRR	50			dB	1 kHz
5	Common mode rejection	CMRR	40			dB	-3.0V ≤ V _{IN} ≤ 3.0V
6	DC open loop voltage gain	A _{VOL}	32			dB	
7	Unity gain bandwidth	f _c	0.30			MHz	
8	Output voltage swing	V _O	4.0			V _{DD}	Load ≥ 100 kΩ to V _{SS}
9	Maximum capacitive load (GS)	C _L			100	pF	
10	Maximum resistive load (GS)	R _L			50	kΩ	
11	Common mode range	V _{CM}	2.5			V _{DD}	No Load

MT8870C AC Electrical Characteristics - V_{DD} = 5.0V ± 5%, V_{SS} = 0V, -40°C ≤ T_O ≤ +85°C, using Test Circuit and with Figure 2.

	Characteristics	Sym	Min	Typ ¹	Max	Units	Notes ²
1	Valid input signal levels (each tone of composite signal)		-29			dBm	1, 2, 3, 5, 6, 9
			-27.5			mV _{RMS}	1, 2, 3, 5, 6, 9
					-1	dBm	1, 2, 3, 5, 6, 9
					869	mV _{RMS}	1, 2, 3, 5, 6, 9
2	Negative twist accept				6	dB	2, 3, 6, 9
3	Positive twist accept				6	dB	2, 3, 6, 9
4	Frequency deviation accept		±1.5%	±2 Hz			2, 3, 5, 9
5	Frequency deviation reject		±3.5%				2, 3, 5, 9
6	Third tone tolerance				-16	dB	2, 3, 4, 5, 9, 10
7	Noise tolerance				-12	dB	2, 3, 4, 5, 7, 9, 10
8	Dial tone tolerance				+22	dB	2, 3, 4, 5, 8, 9, 11

¹ Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

NOTES:

- dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
- Digit sequence consists of all DTMF tones.
- Tone duration: 40 ms, tone pause = 40 ms.
- Signal condition consists of nominal DTMF frequencies.
- Both tones in composite signal have an equal amplitude.
- Tone pair is deviated by ±1.5% ± 2 Hz.
- Bandwidth limited (3 kHz) Gaussian noise.
- The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2%.
- For an error rate of better than 1 in 10,000.
- Referenced to lowest level frequency component in DTMF signal.
- Referenced to the minimum valid accept level.
- For guard time calculation purposes.



MT8870C-1 AC Electrical Characteristics¹ - $V_{DD} = 5.0V \pm 5\%$, $V_E = 0V$, $-40^\circ C \leq T_C \leq 85^\circ C$, using Test Circuit shown in Figure 2.

	Characteristics	Sym	Min	Typ ²	Max	Units	Notes ³
1	Valid input signal levels (each tone of composite signal)		-31		+1	dBm	$V_{DD} = 5.0V$ 1, 2, 3, 5, 6, 9
			21.8		86.9	mV _{RMS}	
2	Input Signal Level Reject		-37			dBm	$V_{DD} = 5.0V$ 1, 2, 3, 5, 6, 9
			10.9			mV _{RMS}	
3	Negative twist accept				6	dB	2, 3, 6, 9
4	Positive twist accept				6	dB	2, 3, 6, 9
5	Frequency deviation accept		$\pm 1.5\% \pm 2 \text{ Hz}$				2, 3, 5, 9
6	Frequency deviation reject		$\pm 3.5\%$				2, 3, 5, 9
7	Third tone tolerance		-18.5			dB	2, 3, 4, 5, 9, 13
8	Noise tolerance			-12		dB	2, 3, 4, 5, 7, 9, 10
9	Dial tone tolerance			+22		dB	2, 3, 4, 5, 8, 9, 11

¹Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

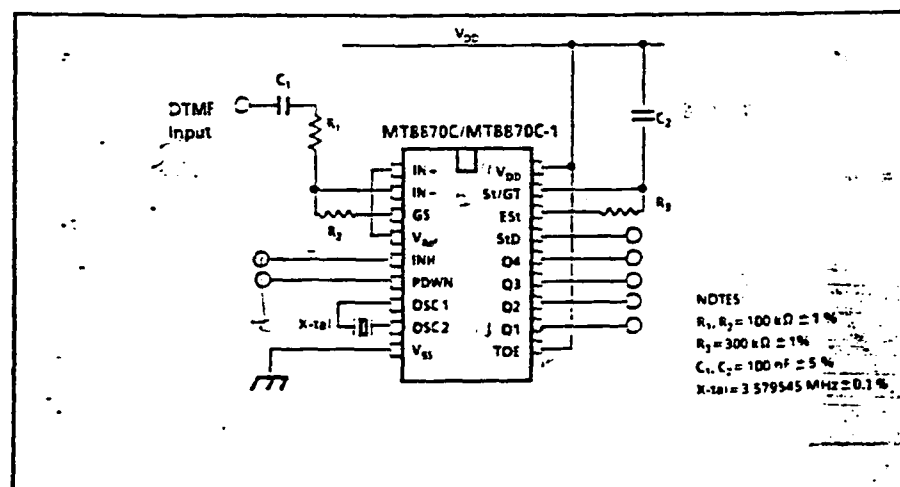
NOTES

- dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
- Digit sequence consists of all DTMF tones.
- Tone duration = 40 ms, tone pause = 40 ms.
- Signal condition consists of nominal DTMF frequencies.
- Both tones in composite signal have an equal amplitude.
- Tone pair is deviated by $\pm 1.5\% \pm 2 \text{ Hz}$.
- Bandwidth limited (3 kHz) Gaussian noise.
- The precise dial tone frequencies are (350 Hz and 440 Hz) $\pm 2\%$.
- For an error rate of better than 1 in 10,000.
- Referenced to lowest level frequency component in DTMF signal.
- Referenced to the minimum valid accept level.
- For guard time calculation purposes.
- Referenced to Fig. 10 input DTMF tone level at -25dBm (-28dBm at GS Pin) interference frequency range between 480-3400Hz.

AC Electrical Characteristics - $V_{DD} = 5.0V \pm 5\%$, $V_E = 0V$, $-40^\circ C \leq T_C \leq 85^\circ C$, using Test Circuit shown in Figure 2.

	Characteristics	Sym	Min	Typ ²	Max	Units	Conditions
1 2 3 4 5 6 T I M I N G	Tone present detect time	t_{DP}	5	13	14	ms	Note 12
	Tone absent detect time	t_{DA}	0.5	4	8.5	ms	Note 12
	Tone duration accept	t_{REC}			40	ms	User adjustable
	Tone duration reject	t_{REE}	20			ms	User adjustable
	Interdigit pause accept	t_{ID}			40	ms	User adjustable
	Interdigit pause reject	t_{IDR}	20			ms	User adjustable
7 8 9 10 11 O U T P U T S	Propagation delay (St to Q)	t_{PD}		8	11	μs	TOE = V_{DD}
	Propagation delay (St to StD)	t_{PS1D}		12	16	μs	TOE = V_{DD}
	Output data set up (Q to StD)	t_{QStD}		3.4		μs	TOE = V_{DD}
	Propagation delay (TOE to Q ENABLE)	t_{PTE}		50		ns	load of 10 k Ω , 50 pF
	Propagation delay (TOE to Q DISABLE)	t_{PTD}		300		ns	load of 10 k Ω , 50 pF
	Crystal/clock frequency	f_C	3.5759	3.5795	3.5831	MHz	
12 13 14 15 16 C L O C K	Clock input rise time	t_{rHCL}			110	ns	Ext. clock
	Clock input fall time	t_{fHCL}			110	ns	Ext. clock
	Clock input duty cycle	DC_{CL}	40	50	60	%	Ext. clock
	Capacitive load (OSC2)	C_{LO}			30	pF	

²Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.



Pin Description

Pin #	Name	Description
1	IN+	Non-Inverting Op-Amp (Input).
2	IN-	Inverting Op-Amp (Input).
3	GS	Gain Select. Gives access to output of front end differential amplifier for connection of feedback resistor.
4	V _{ref}	Reference Voltage (Output). Nominally V _{DD} /2 is used to bias inputs at mid-rail (see Fig.2).
5	INH	Inhibit (Input). Logic high inhibits the detection of tones representing characters A, B, C and D. This pin input is internally pulled down.
6	PWDN	Power Down (Input). Active high. Powers down the device and inhibits the oscillator. This pin input is internally pulled down.
7	OSC1	Clock (Input).
8	OSC2	Clock (Output). A 3.579545 MHz crystal connected between pins OSC1 and OSC2 completes the internal oscillator circuit.
9	V _{SS}	Negative Power Supply (Input).
10	TOE	Three State Output Enable (Input). Logic high enables the outputs Q1-Q4. This pin is pulled up internally.
11-14	Q1-Q4	Three State Data (Output). When enabled by TOE, provide the code corresponding to the last valid tone-pair received (see Table 1). When TOE is logic low, the data outputs are high impedance.
15	StD	Delayed Steering (Output). Presents a logic high when a received tone-pair has been registered and the output latch updated; returns to logic low when the voltage on St/GT falls below V _{SS} .
16	Est	Early Steering (Output). Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause Est to return to a logic low.
17	St/GT	Steering Input/Guard time (Output) Bidirectional. A voltage greater than V _{SS} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V _{SS} frees the device to accept a new tone pair. The GT output acts to reset the external steering time-constant; its state is a function of Est and the voltage on St.
18	V _{DD}	Positive power supply (Input).

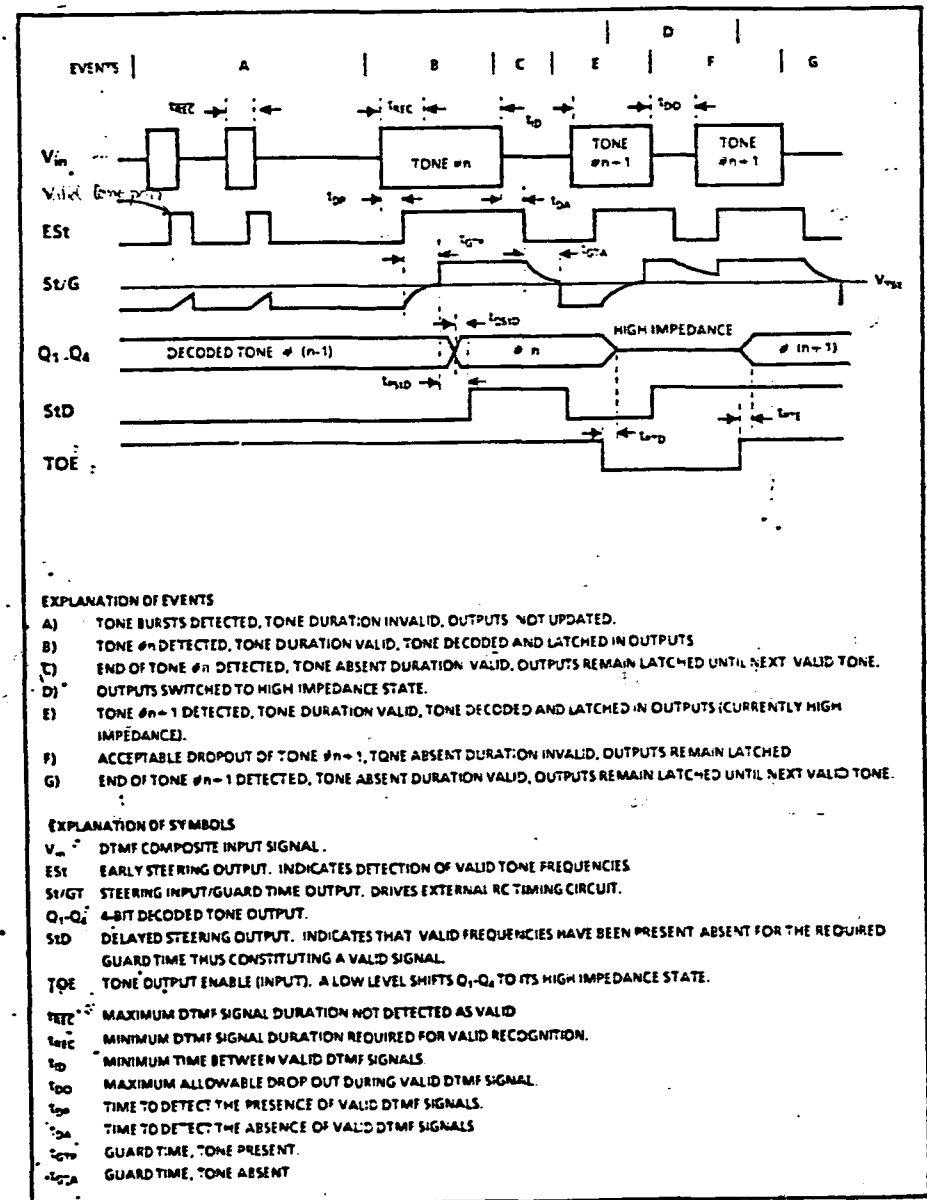


Figure 3- Timing Diagram

Functional Description

The MT8870C/MT8870C-1 monolithic DTMF receiver offers small size, low power consumption and high performance. Its architecture consists of a bandsplit filter section, which separates the high and low group tones, followed by a digital counting section which verifies the frequency and duration of the received tones before passing the corresponding code to the output bus.

Filter Section

Separation of the low-group and high group tones is achieved by applying the DTMF signal to the inputs of two sixth-order switched capacitor bandpass filters, the bandwidths of which correspond to the low and high group frequencies. The filter section also incorporates notches at 350 and 440 Hz for exceptional dial tone rejection (see Figure 4). Each filter output is followed by a single order switched capacitor filter section which smooths the signals prior to limiting. Limiting is performed by high-gain comparators which are provided with hysteresis to prevent detection of unwanted low-level signals. The outputs of the comparators provide full rail logic swings at the frequencies of the incoming DTMF signals.

Decoder Section

Following the filter section is a decoder employing digital counting techniques to determine the frequencies of the incoming tones and to verify that they correspond to standard DTMF frequencies. A complex averaging algorithm protects against tone

simulation by extraneous signals such as voice while providing tolerance to small frequency deviations and variations. This averaging algorithm has been developed to ensure an optimum combination of immunity to talk-off and tolerance to the presence of interfering frequencies (third tones) and noise. When the detector recognizes the presence of two valid tones (this is referred to as the "signal condition" in some industry specifications) the "Early Steering" (EST) output will go to an active state. Any subsequent loss of signal condition will cause EST to assume an inactive state (see "Steering Circuit").

Steering Circuit

Before registration of a decoded tone pair, the receiver checks for a valid signal duration (referred to as character recognition condition). This check is performed by an external RC time constant driven by EST. A logic high on EST causes v_c (see Figure 5) to rise as the capacitor discharges. Provided signal condition is maintained (EST remains high) for the validation period (t_{GT}), v_c reaches the threshold (V_{TS}) of the steering logic to register the tone pair, latching its corresponding 4-bit code (see Table 1) into the output latch. At this point the GT output is activated and drives v_c to V_{DD} . GT continues to drive high as long as EST remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (StD) goes high, signalling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the three state control input (TOE) to a logic high. The steering circuit works in reverse to validate

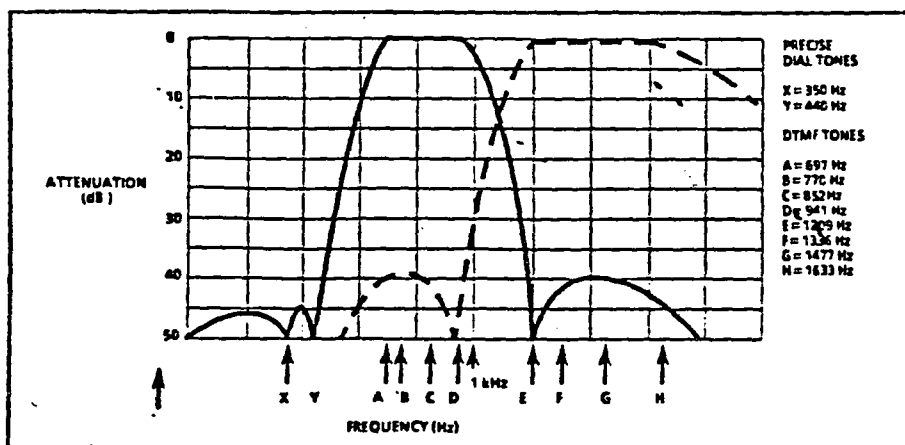


Figure 4- Filter Response

Digit	TOE	INH	EST	Q ₄	Q ₃	Q ₂	Q ₁
ANY	L	X	-	Z	Z	Z	Z
1	H	L	H	0	0	0	1
2	H	L	H	0	0	1	0
3	H	L	H	0	0	1	1
4	H	L	H	0	1	0	0
5	H	L	H	0	1	0	1
6	H	L	H	0	1	1	0
7	H	L	H	0	1	1	1
8	H	L	H	1	0	0	0
9	H	L	H	1	0	0	1
0	H	L	H	1	0	1	0
*	H	L	H	1	0	1	1
#	H	L	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
1	H	H	H	0	0	0	1
2	H	H	H	0	0	1	0
3	H	H	H	0	0	1	1
4	H	H	H	0	1	0	0
5	H	H	H	0	1	0	1
6	H	H	H	0	1	1	0
7	H	H	H	0	1	1	1
8	H	H	H	1	0	0	0
9	H	H	H	1	0	0	1
0	H	H	H	1	0	1	0
*	H	H	H	1	0	1	1
#	H	H	H	1	1	0	0
A	H	H	L	undetected, the output code will remain the same as the previous detected code			
B	H	H	L				
C	H	H	L				
D	H	H	L				

L = LOGIC LOW, H = LOGIC HIGH, Z = HIGH IMPEDANCE

Table 1 - Functional Decode Table

the interdigit pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (dropouts) too short to be considered a valid pause. This facility, together with the capability of selecting the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

Guard Time Adjustment

In many situations not requiring selection of tone duration and interdigit pause, the simple steering

circuit shown in Figure 5 is applicable. Component values are chosen according to the formula:

$$t_{REC} = t_{DP} + t_{GTP}$$

$$t_{ID} = t_{DA} + t_{GTA}$$

The value of t_{DP} is a device parameter (see Figure 3) and t_{GTP} is the minimum signal duration to be recognized by the receiver. A value for C of 0.1 μ F is recommended for most applications, leaving R to be selected by the designer.

Different steering arrangements may be used to select independently the guard times for tone present (t_{GTP}) and tone absent (t_{GTA}). This may be necessary to meet system specifications which place both accept and reject limits on both tone duration and interdigit pause. Guard time adjustment also allows the designer to tailor system parameters such as talk off and noise immunity. Increasing t_{REC} improves talk-off performance since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. Alternatively, a relatively short t_{REC} with a long t_{ID} would be appropriate for extremely noisy environments where fast acquisition time and immunity to tone drop-outs are required. Design information for guard time adjustment is shown in Figure 6.

Power-down and Inhibit Mode

A logic high applied to pin 6 (PWDN) will power down the device to minimize the power consumption in a standby mode. It stops the oscillator and the functions of the filters.

Inhibit mode is enabled by a logic high input to the pin 5 (INH). It inhibits the detection of tones representing characters A, B, C and D. The output code will remain the same as the previous detected code (see Table 1).

Differential Input Configuration

The input arrangement of the MT8870C/MT8870C-1 provides a differential-input operational amplifier as well as a bias source (V_{BI}) which is used to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the op-amp output (G5) for adjustment of gain. In a single-ended configuration, the input pins are connected as shown in Figure 2 with the op-amp connected for unity gain and V_{BI} biasing the input at $\frac{1}{2}V_{DD}$. Figure 7 shows the differential configuration, which permits the adjustment of gain with the feedback resistor R_5 .

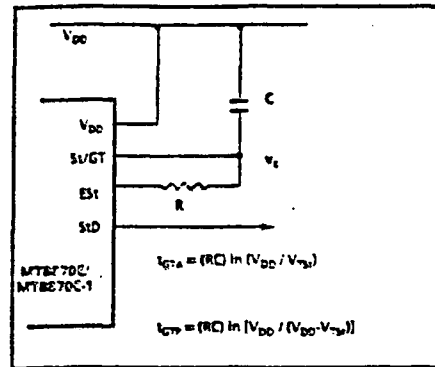
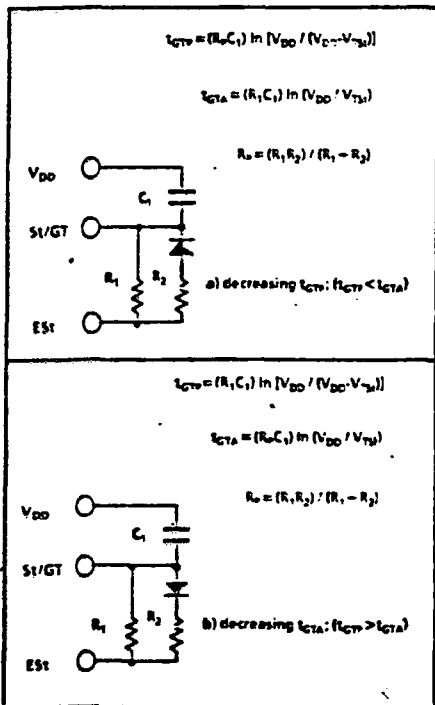
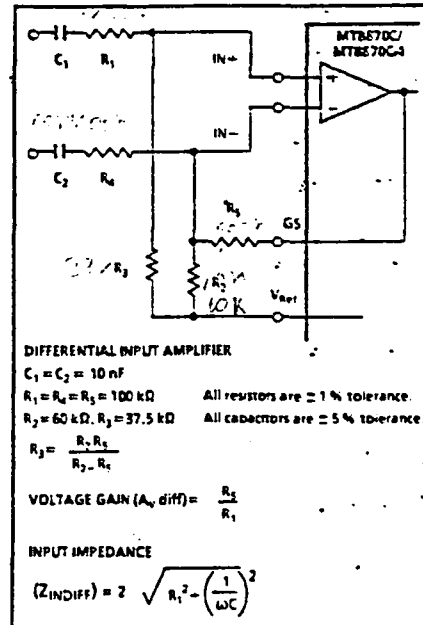
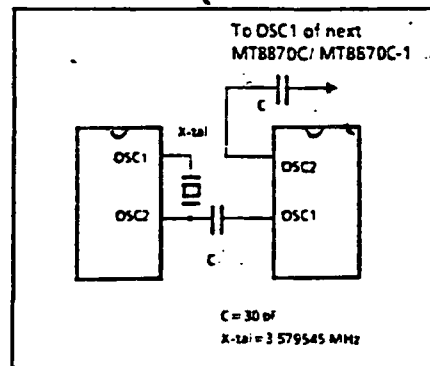


Figure 5- Basic Steering Circuit


Figure 6- Guard Time Adjustment
Crystal Oscillator

The internal clock circuit is completed with the addition of an external 3.579545 MHz crystal and is normally connected as shown in Figure 2 (Single Ended Input Configuration). However, it is possible to configure several MT8870C/MT8870C-1 devices


Figure 7- Differential Input Configuration
employing only a single oscillator crystal. The oscillator output of the first device in the chain is coupled through a 30 pF capacitor to the oscillator input (OSC1) of the next device. Subsequent devices are connected in a similar fashion. Refer to Figure 8 for details. The problems associated with unbalanced loading are not a concern with the arrangement shown, i.e. precision balancing capacitors are not required.


APPLICATION

RECEIVER SYSTEM FOR BRITISH TELECOM SPEC FOR 1151

The circuit shown in Fig. 10 illustrates the use of MT8870C-1 device in a typical receiver system. BT Spec defines the input signals less than -34 dBm as the non-operate level. This condition can be attained by choosing a suitable values of R_1 and R_2 to provide 3 dB attenuation, such that -34 dBm input signal will correspond to -37 dBm at the gain setting pin GS of MT8870C-1. As shown in the diagram, the component values of R_3 and C_2 are the guard time requirements when the total component tolerance is 6%. For better performance, it is recommended to use the non-symmetric guard time circuit in Fig. 9.

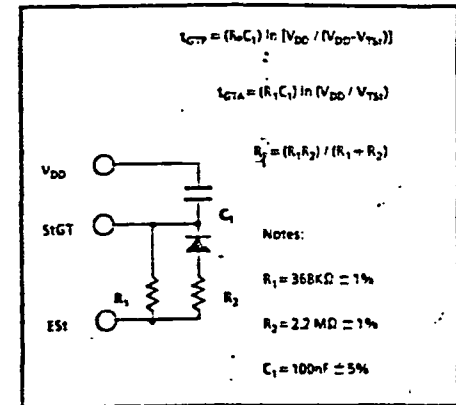


Figure 9 - Non-Symmetric Guard Time Circuit

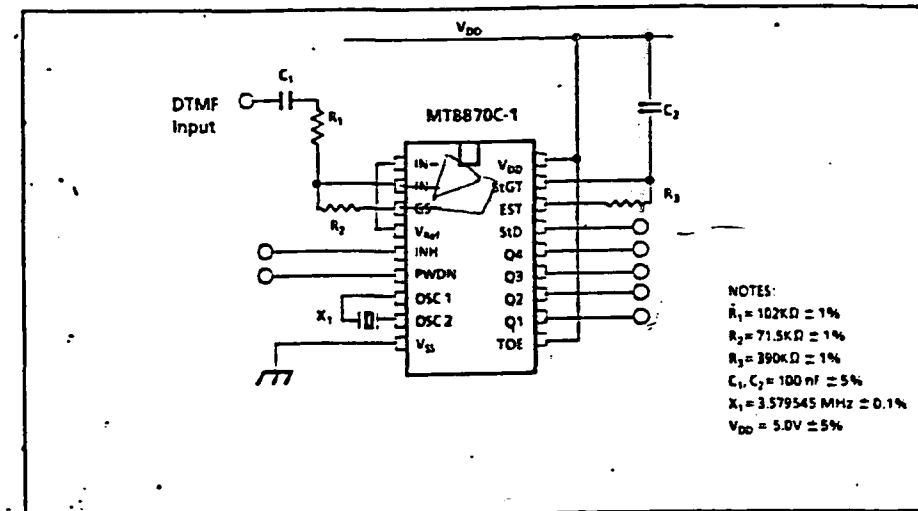


Figure 10 - Single-Ended Input Configuration for BT or CEPT Spec

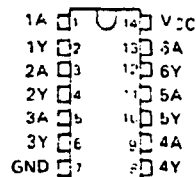
SN5414, SN54LS14, SN7414, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

DECEMBER 1963 REVISED MARCH 1988

- Operation from Very Slow Edges
- Improved Line-Receiving Characteristics
- High Noise Immunity

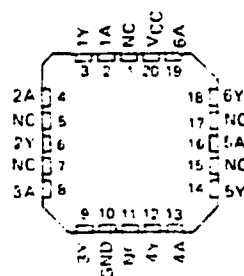
SN5414, SN54LS14 J OR N PACKAGE
SN7414 D OR N PACKAGE
SN74LS14 D OR N PACKAGE

(TOP VIEW)



SN54LS14 FK PACKAGE

(TOP VIEW)



NC = No Internal Connection

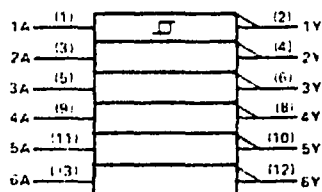
description

Each circuit functions as an inverter, but because of the Schmitt action, it has different input threshold levels for positive (V_{T+}) and for negative going (V_{T-}) signals.

These circuits are temperature-compensated and can be triggered from the slowest of input ramps and still give clean, jitter-free output signals.

The SN5414 and SN54LS14 are characterized for operation over the full military temperature range of -55°C to 125°C . The SN7414 and the SN74LS14 are characterized for operation from 0°C to 70°C .

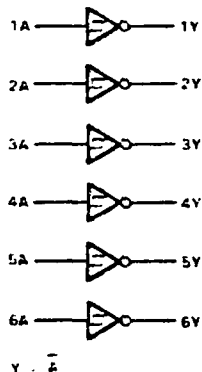
logic symbol¹



¹This symbol is in accordance with ANSI/IEEE Std 91-1964 and IEC Publication 617-12.

Pin numbers shown are for C, J, N, and V packages.

logic diagram (positive logic)



PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

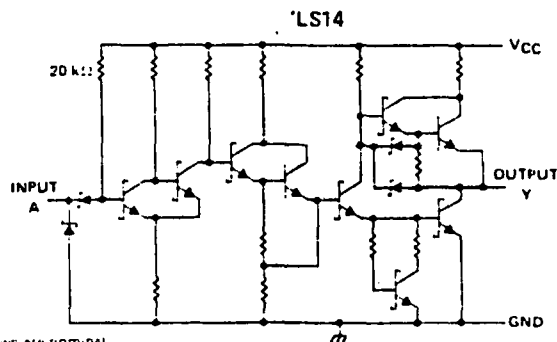
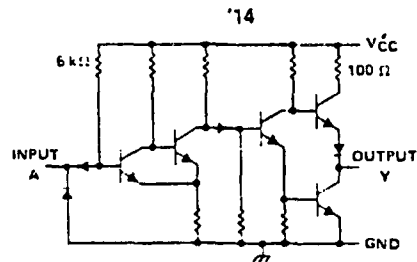
TEXAS
INSTRUMENTS

POST OFFICE BOX 655312 • DALLAS, TEXAS 75265

2
TTL Devices

SN5414, SN54LS14, SN7414, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

schematics



Resistor values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage: '14	5.5 V
'LS14	7 V
Operating free-air temperature: SN54'	-55°C to 125°C
SN74'	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

SN5414, SN7414 HEX SCHMITT-TRIGGER INVERTERS

recommended operating conditions

	SN5414			SN7414			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I _{OH} High-level output current			-0.5			-0.5	mA
I _{OL} Low-level output current			16			16	mA
T _A Operating free-air temperature	-55		125	0		70	°C

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

PARAMETER	TEST CONDITIONS ¹	MIN	TYP ²	MAX	UNIT
V _{T+}	V _{CC} = 5 V	1.5	1.7	2	V
V _{T-}	V _{CC} = 5 V	0.6	0.9	1.1	V
Hysteresis (V _{T+} - V _{T-})	V _{CC} = 5 V	0.4	0.8		V
V _{IK}	V _{CC} = MIN., I _I = -12 mA			-1.5	V
V _{OH}	V _{CC} = MIN., V _I = 0.6 V, I _{OH} = -0.8 mA	2.4	3.4		V
V _{OL}	V _{CC} = MIN., V _I = 2 V, I _{OL} = 16 mA		0.2	0.4	V
I _{T+}	V _{CC} = 5 V, V _I = V _{T+}		-0.43		mA
I _{T-}	V _{CC} = 5 V, V _I = V _{T-}		-0.56		mA
I _I	V _{CC} = MAX., V _I = 5.5 V			1	mA
I _{IH}	V _{CC} = MAX., V _{IH} = 2.4 V			40	μA
I _{IL}	V _{CC} = MAX., V _{IL} = 0.4 V		-0.8	-1.2	mA
I _{OS} ³	V _{CC} = MAX.	-18		-55	mA
I _{CC}	V _{CC} = MAX.		22	36	mA
I _{CCL}	V _{CC} = MAX.		39	60	mA

¹ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

² All typ. val. values are at V_{CC} = 5 V, T_A = 25°C.

³ Not more than one output should be shorted at a time.

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

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PLH}	A	Y	R _L = 400 Ω, C _L = 15 pF	15	22		ns
t _{PHL}				15	22		ns

TTL Devices

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN54LS14, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

recommended operating conditions

	SN54LS14			SN74LS14			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I _{OH} High-level output current			-0.4			-0.4	mA
I _{OL} Low-level output current			4			8	mA
T _A Operating free-air temperature	-55		125	0		70	°C

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

PARAMETER	TEST CONDITIONS ¹	SN54LS14			SN74LS14			UNIT
		MIN	TYP ²	MAX	MIN	TYP ²	MAX	
V _{T+}	V _{CC} 5V	1.4	1.6	1.9	1.4	1.6	1.9	V
V _{T-}	V _{CC} 5V	0.5	0.8	1	0.5	0.8	1	V
V _{THRESH} (V _{T+} - V _{T-})	V _{CC} 5V	0.4	0.8		0.4	0.8		V
V _{IH}	V _{CC} MIN, I _I = 10 nA			1.5			1.5	V
V _{OH}	V _{CC} MIN, V _I = 0.5 V, I _{OH} = -0.4 mA	2.5	3.4		2.7	3.4		V
V _{OL}	V _{CC} MIN, V _I = 1.4 V	I _{OL} 4 mA			I _{OL} 4 mA			V
		I _{OL} 8 mA			I _{OL} 8 mA			
I _{T+}	V _{CC} 5V, V _I = V _{T+}	0.14			0.14			mA
I _{T-}	V _{CC} 5V, V _I = V _{T-}	0.18			0.18			mA
I _I	V _{CC} MAX, V _I = 7 V	0.1			0.1			mA
I _{IH}	V _{CC} MAX, V _{IH} = 2.7 V	20			20			nA
I _{IL}	V _{CC} MAX, V _{IL} = 0.4 V	0.4			0.4			nA
I _{OS}	V _{CC} MAX	100			100			nA
I _{CCM}	V _{CC} MAX	8.6			8.6			mA
I _{CC1}	V _{CC} MAX	12			12			mA

¹ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

² A typical value at V_{CC} = 5 V, T_A = 25 °C.

³ Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

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

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS		MIN	TYP	MAX	UNIT
t _{PLH}	A	Y	P _L = 2 nA	C _L = 15 pF	15	22		ns
t _{PHL}								

TTL Devices

NOTES A. All diodes are 1N3064 or equivalent.
B. C_L includes probe and jig capacitance.
C. Generator characteristics and reference voltage are

	Generator Characteristics				Reference Voltages		
	Z _{out}	PRR	t _r	t _f	V _{I ref(H)}	V _{I ref(L)}	V _{O ref}
SN54LS74, SN74LS74	50 Ω	1 MHz	10 ns	10 ns	1.7 V	0.9 V	1.5 V
SN54ALS74, SN74ALS74	50 Ω	1 MHz	15 ns	6 ns	1.6 V	0.8 V	1.3 V

POSITIVE-GOING THRESHOLD VOLTAGE VS FREE-AIR TEMPERATURE

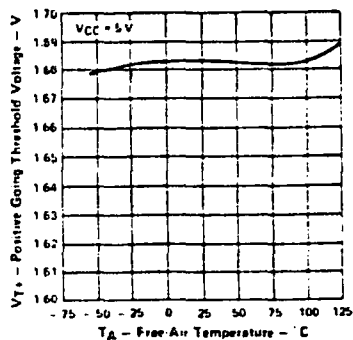


FIGURE 1

INTEGRATIVE-GOING THRESHOLD VOLTAGE
VS
FREE AIR TEMPERATURE

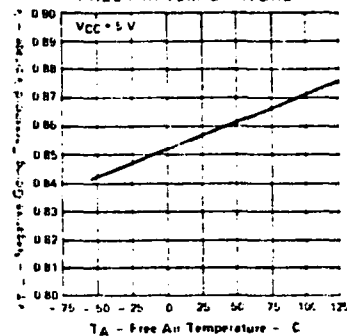


FIGURE 2

HYSTERESIS
vs
FREE AIR TEMPERATURE

$V_{GS} - V_{TH} - \text{Hysteresis} - \text{mV}$

$T_A - \text{Free Air Temperature} - ^\circ\text{C}$

FIGURE 3

Data for temperatures below 0 °C and 70 °C and supply voltages below 4.75 V and above 5.25 V are applicable for SN5414 only.

SN5414, SN7414 HEX SCHMITT-TRIGGER INVERTERS

TYPICAL CHARACTERISTICS OF '14 CIRCUITS

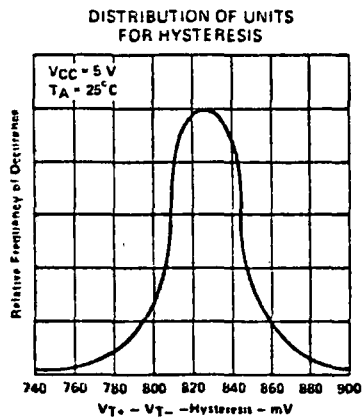


FIGURE 4

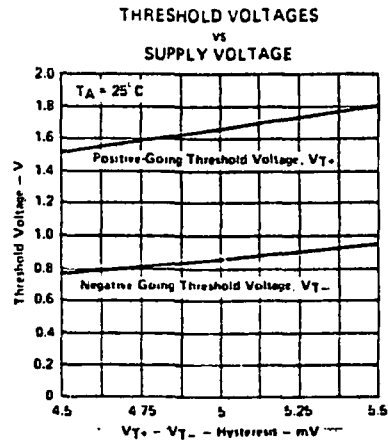


FIGURE 5

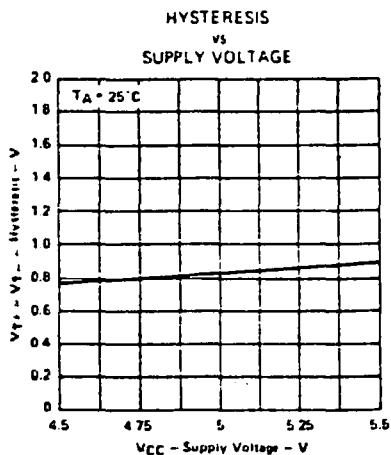


FIGURE 6

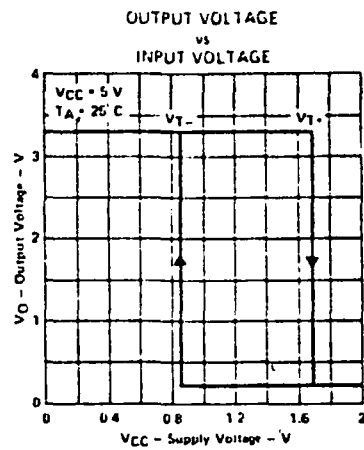


FIGURE 7

Data for temperatures below 0°C and 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN5414 only.

SN54LS14, SN74LS14
HEX SCHMITT-TRIGGER INVERTERS

TYPICAL CHARACTERISTICS OF 'LS14 CIRCUITS

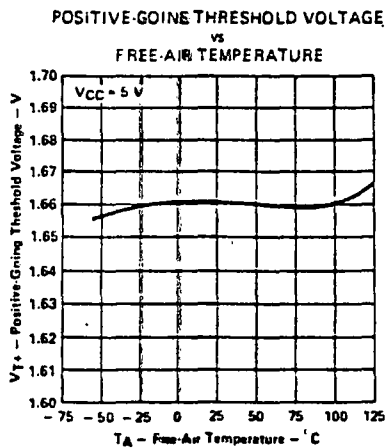


FIGURE 8

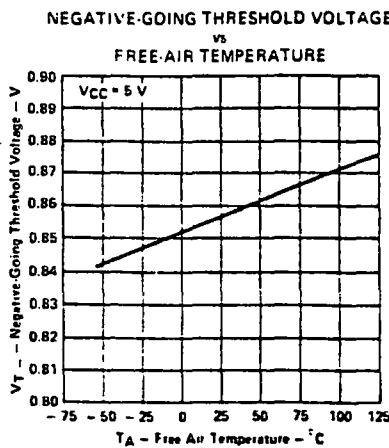


FIGURE 9

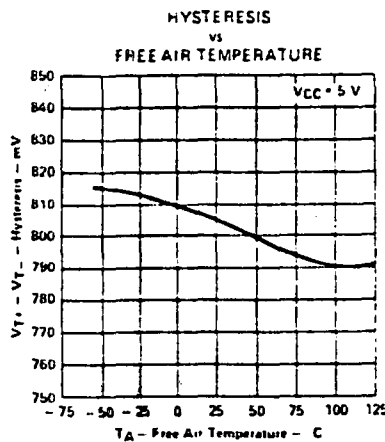


FIGURE 10

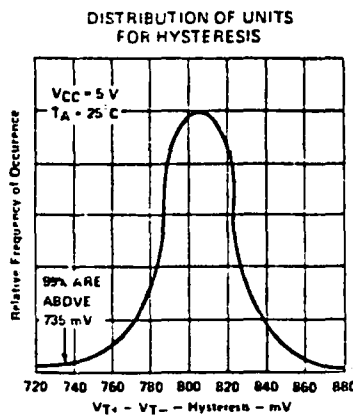


FIGURE 11

Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN54LS14 only.

2
TTL Devices

SN54LS14, SN74LS14
HEX SCHMITT-TRIGGER INVERTERS

TYPICAL CHARACTERISTICS OF 'LS14 CIRCUITS

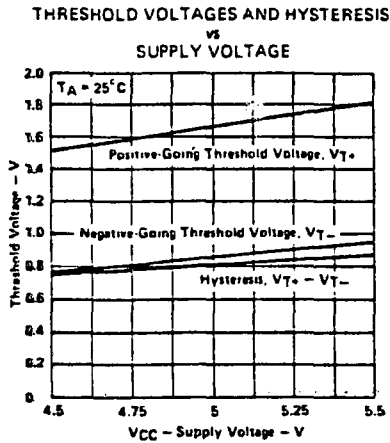


FIGURE 12

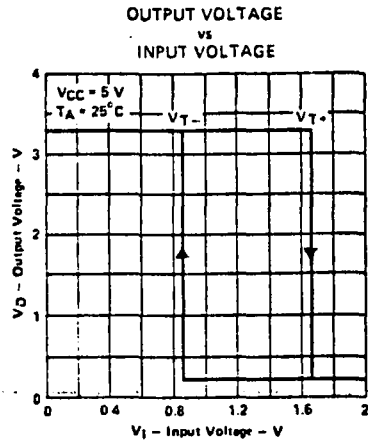


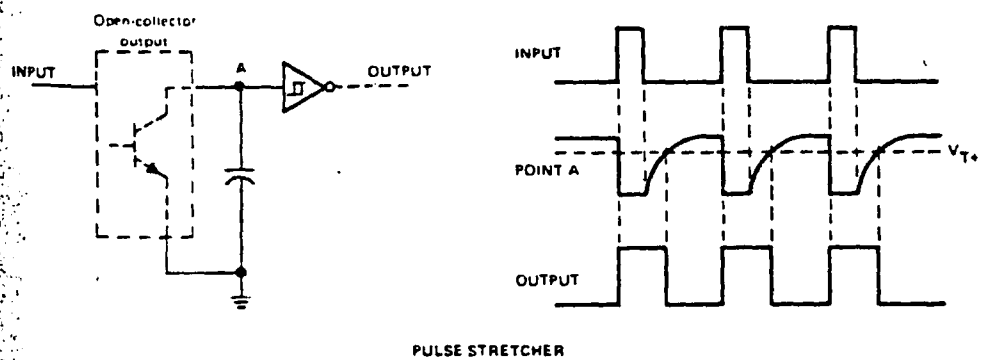
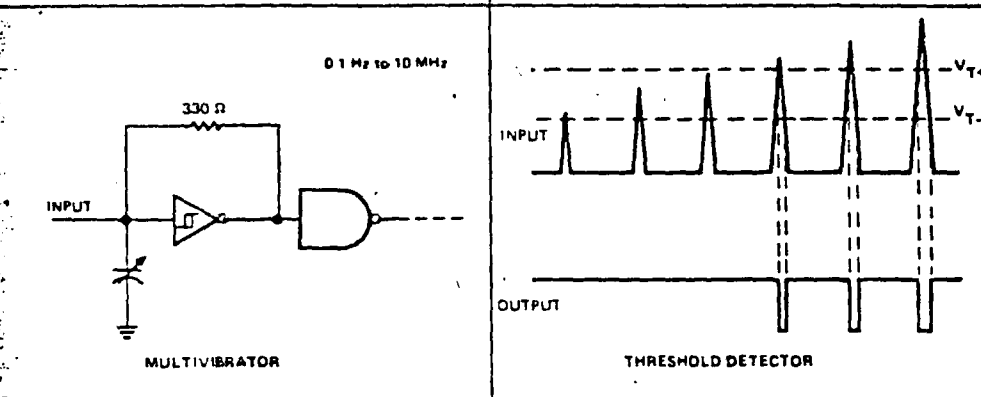
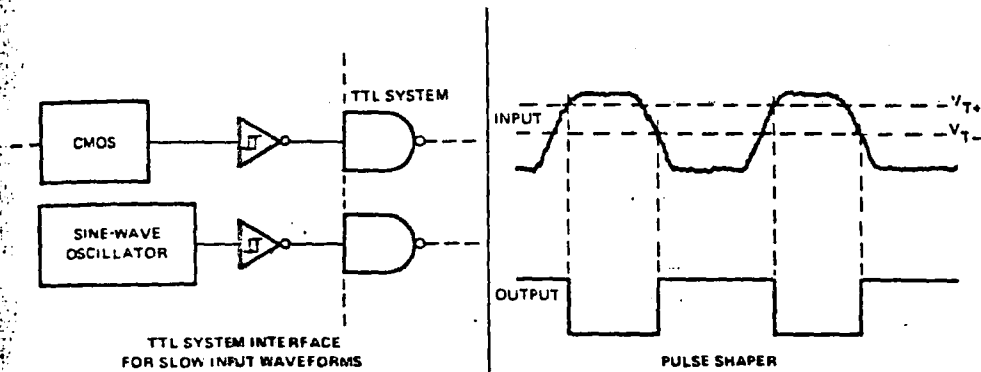
FIGURE 13

Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN54LS14 only.

TTL Devices

SN5414, SN54LS14
SN7414, SN74LS14
HEX SCHMITT-TRIGGER INVERTERS

TYPICAL APPLICATION DATA



TTL Devices

**SN5490A, SN5492A, SN5493A, SN54LS90, SN54LS92, SN54LS93,
SN7490A, SN7492A, SN7493A, SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE AND BINARY COUNTERS**

MARCH 1974, REVISED MARCH 1986

'90A, 'LS90 . . . Decade Counters

'92A, 'LS92 . . . Divide By-Twelve Counters

'93A, 'LS93 . . . 4-Bit Binary Counters

TYPES	TYPICAL POWER DISSIPATION
'90A	145 mW
'92A, '93A	130 mW
'LS90, 'LS92, 'LS93	45 mW

description

Each of these monolithic counters contains four master-slave flip-flops and additional gating to provide a divide-by-two counter and a three-stage binary counter for which the count cycle length is divide-by-five for the '90A and 'LS90, divide-by-six for the '92A and 'LS92, and the divide-by-eight for the '93A and 'LS93.

All of these counters have a gated zero reset and the '90A and 'LS90 also have gated set-to-nine inputs for use in BCD nine's complement applications.

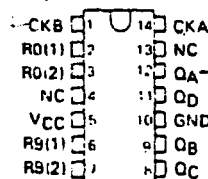
To use their maximum count length (decade, divide-by-twelve, or four-bit binary) of these counters, the CKB input is connected to the QA output. The input count pulses are applied to CKA input and the outputs are as described in the appropriate function table. A symmetrical divide-by-ten count can be obtained from the '90A or 'LS90 counters by connecting the QD output to the CKA input and applying the input count to the CKB input which gives a divide-by-ten square wave at output QA.

SN5490A, SN54LS90 . . . J OR W PACKAGE

SN7490A . . . N PACKAGE

SN74LS90 . . . D OR N PACKAGE

(TOP VIEW)

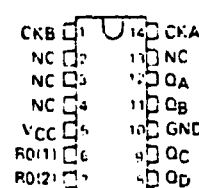


SN5492A, SN54LS92 . . . J OR W PACKAGE

SN7492A . . . N PACKAGE

SN74LS92 . . . D OR N PACKAGE

(TOP VIEW)

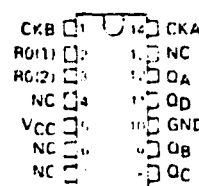


SN5493A, SN54LS93 . . . J OR W PACKAGE

SN7493 . . . N PACKAGE

SN74LS93 . . . D OR N PACKAGE

(TOP VIEW)



NC No internal connection

2

TTL Devices

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

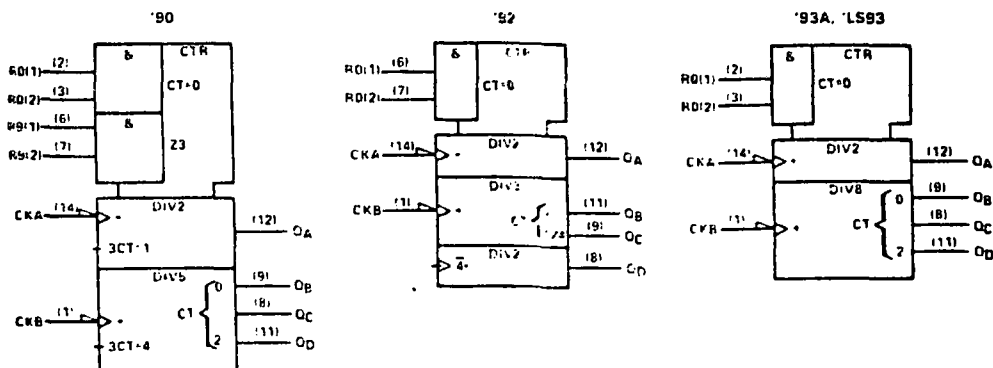
**TEXAS
INSTRUMENTS**

POST OFFICE BOX 155502 • DALLAS, TEXAS 75265

2-277

SN5490A, '92A, '93A, SN54LS90, 'LS92, 'LS93,
SN7490A, '92A, '93A, SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

logic symbols†



†These symbols are in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.

TTL Devices

SN5490A, '92A, '93A, SN54LS90, 'LS92, 'LS93,
SN7490A, '92A, '93A, SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

'90A, 'LS90
BCD COUNT SEQUENCE
(See Note A)

COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

'90A, 'LS90
BI-QUINARY (5-2)
(See Note B)

COUNT	OUTPUT			
	Q _A	Q _D	Q _C	Q _B
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

'92A, 'LS92
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	H	L	L	L
7	H	L	L	H
8	H	L	H	L
9	H	L	H	H
10	H	H	L	L
11	H	H	L	H

'90A, 'LS90
RESET COUNT FUNCTION TABLE

RESET INPUTS				OUTPUT			
R ₀₍₁₎	R ₀₍₂₎	R ₉₍₁₎	R ₉₍₂₎	Q _D	Q _C	Q _B	Q _A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

'92A, 'LS92, '93A, 'LS93
RESET COUNT FUNCTION TABLE

RESET INPUTS		OUTPUT			
R ₀₍₁₎	R ₀₍₂₎	Q _D	Q _C	Q _B	Q _A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

- NOTES A Output Q_D is connected to input C₀ for BCD count
B Output Q_D is connected to input C₀ for bi-quinary count
C Output Q_D is connected to input C₀ B
D H = high level, L = low level, X = irrelevant

'93A, 'LS93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

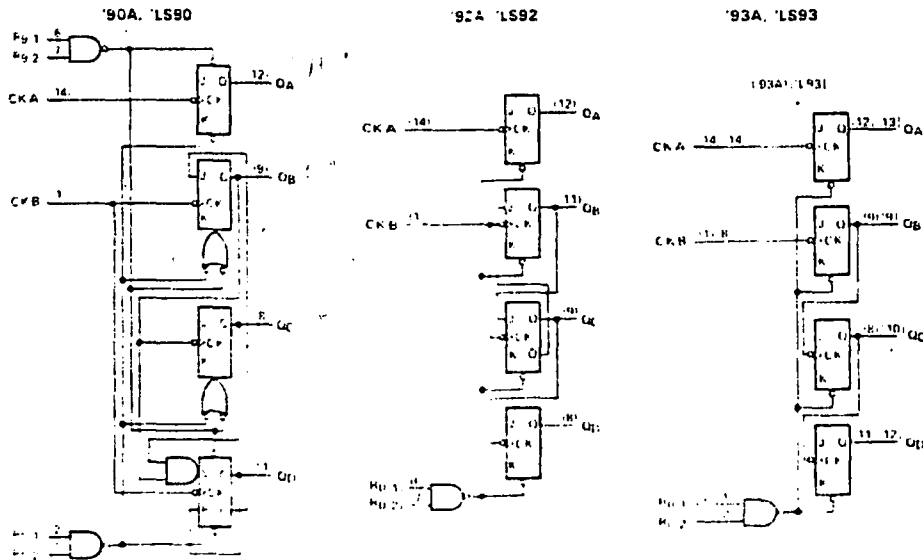
TTL Devices

TEXAS
INSTRUMENTS

POST OFFICE BOX 65570 • DALLAS, TEXAS 75265

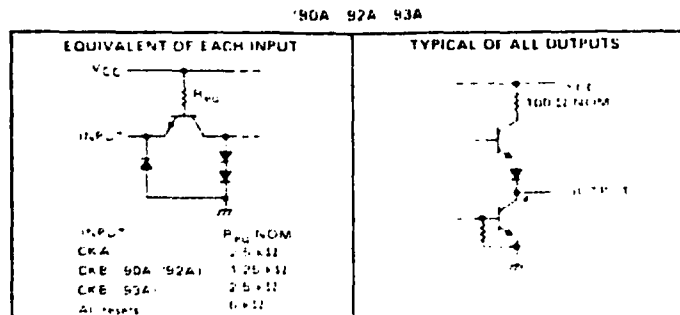
**SN5490A, '92A, '93A, SN54LS90, 'LS92, 'LS93,
SN7490A, '92A, '93A, SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS**

logic diagrams (positive logic)



The logic diagrams shown in this document are for reference only and are not to be used as a basis for design. Pin numbers shown in parentheses for the 'LS90 and '90A and pin numbers shown in parentheses for the '92, '93.

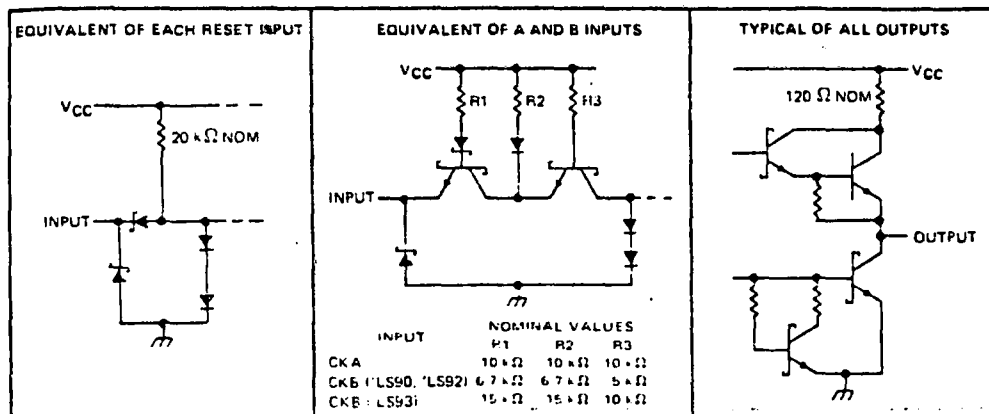
schematics of inputs and outputs



SN54LS90, 'LS92, 'LS93,
SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

schematics of inputs and outputs (continued)

'LS90, 'LS92, 'LS93



TTL Devices

SN5490A, SN5492A, SN5493A, SN7490A, SN7492A, SN7493A DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	5.5 V
Intermittent voltage (see Note 2)	5.5 V
Operating free air temperature range: SN5490A, SN5492A, SN5493A	-55 C to 125 C
SN7490A, SN7492A, SN7493A	0 C to 70 C
Storage temperature range	-65 C to 150 C

NOTES: 1. Voltage values except intermittent voltage are with respect to network ground terminal.
2. This is the voltage between two emitters of a multi-emitter transistor. For these circuits, this rating applies between the two R_D inputs and for the 90A circuit, it also applies between the two R_D inputs.

recommended operating conditions

	SN5490A, SN5492A SN5493A			SN7490A, SN7492A SN7493A			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			800			800	μ A
Low-level output current, I_{OL}			16			16	mA
Count frequency, f_{COUNT} (see Figure 1)	A input	0	32	0	32		MHz
	B input	0	16	0	16		
Pulse width, t_w	A input	15		15			ns
	B input	30		30			
	Reset inputs	15		15			
High-to-low state setup time, t_{su}		25		25			ns
Operating free air temperature, T_A	-55		125	0		70	C

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

PARAMETER ^a	TEST CONDITIONS ^b	90A			92A			93A		UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	
V_{IH} High-level input voltage		2			2			2		V
V_{IL} Low-level input voltage				0.8			0.8			0.8 V
V_{OL} Input clamp voltage	V_{CC} MIN, $I_L = 12$ mA			1.5			1.5			1.5 V
V_{OH} High-level output voltage	V_{CC} MIN, $V_{IH} = 2$ V, $V_{IL} = 0.8$ V, $I_{OH} = 800$ μ A	2.4	3.4		2.4	3.4		2.4	3.4	V
V_{OL} Low-level output voltage	V_{CC} MIN, $V_{IH} = 2$ V, $V_{IL} = 0.8$ V, $I_{OL} = 16$ mA ^c	0.2	0.4		0.2	0.4		0.2	0.4	V
I_I Input current at maximum input voltage	V_{CC} MAX, $V_I = 5.5$ V			1			1			1 mA
I_{IH} High-level input current	Any reset			40			40			40 μ A
	CKA			80			80			80 μ A
	CKB			120			120			120 μ A
I_{IL} Low-level input current	Any reset			-16			-16			-16 mA
	CKA			-32			-32			-32 mA
	CKB			-48			-48			-48 mA
I_{OS} Short-circuit output current ^d	V_{CC} MAX	SN54	20	57	20	57	20	57		mA
		SN74	18	57	18	57	18	57		
I_{CC} Supply current	V_{CC} MAX, See Note 3		29	42		26	39		26	39 mA

^a Parameters are shown as MIN or MAX, use the appropriate value for the parameter. Recommended operating conditions.

^b All typical values are at $V_{CC} = 5$ V, $T_A = 25^\circ$ C.

^c Not more than one output should be shorted at a time.

^d C_A outputs are tested at $I_{OL} = 16$ mA plus the limit value for I_{IL} for the CKB input. This permits driving the CKB input while maintaining full fan-out capability.

NOTE 3: I_{CC} is measured with all outputs driven, both R_D inputs grounded following momentary connection to 4.5 V, and all other inputs grounded.

SN5490A, SN5492A, SN5493A, SN7490A, SN7492A, SN7493A
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

switching characteristics, $V_{CC} = 5V$, $T_A = 25^\circ C$

PARAMETER ¹	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'90A			'92A			'93A			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}	CKA	QA	$C_L = 15 pF$, $R_L = 400 \Omega$, See Figure 1	32	42		32	42		32	42		MHz
	CKB	QB		16			16			16			
t_{PLH}	CKA	QA		10	16		10	16		10	16		ns
t_{PHL}				12	18		12	18		12	18		
t_{PLH}	CKA	QD		32	48		32	48		46	70		ns
t_{PHL}				34	50		34	50		46	70		
t_{PLH}	CKB	QB		10	16		10	16		10	16		ns
t_{PHL}				14	21		14	21		14	21		
t_{PLH}	CKB	QC		21	32		10	16		21	32		ns
t_{PHL}				23	35		14	21		23	35		
t_{PLH}	CKB	QD		21	32		21	32		34	51		ns
t_{PHL}				23	35		23	35		34	51		
t_{PHL}	SWING	Any		26	40		26	40		26	40		ns
t_{PLH}	SWING	QA, QD		20	30								ns
t_{PHL}		QB, QC		20	40								

¹ f_{max} = maximum count frequency

t_{PLH} = propagation delay time: low to high level output

t_{PHL} = propagation delay time: high to low level output

TTL Devices

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

2-283

SN54LS90, SN54LS92, SN54LS93, SN74LS90, SN74LS92, SN74LS93 DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage: R inputs	7 V
A and B inputs	5.5 V
Operating free-air temperature range: SN54LS [†] Circuits	-55°C to 125°C
SN74LS [‡] Circuits	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1 Voltage values are with respect to network ground terminal

recommended operating conditions

		SN54LS90 SN54LS92 SN54LS93			SN74LS90 SN74LS92 SN74LS93			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5 [*]	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}				-400			-400	μ A
Low-level output current, I_{OL}				4			8	mA
Count frequency, f_{COUNT} (see Figure 3)	A input	0	32	0	0	32		MHz
	B input	0	16	0	0	16		
Pulse width, t_{PW}	A input	15		15				ns
	B input	30		30				
	Reset inputs	30		30				
Reset inactive-state setup time, t_{SU}		25		25				ns
Operating free-air temperature, T_A		55	125	0	70			°C

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

PARAMETER		TEST CONDITIONS ¹		SN54LS90 SN54LS92 SN54LS93			SN74LS90 SN74LS92 SN74LS93			UNIT
				MIN	TYP ²	MAX	MIN	TYP ²	MAX	
V_{IH}	High-level input voltage			2			2			V
V_{IL}	Low-level input voltage				0.7			0.8		V
V_{IK}	Input clamp voltage	$V_{CC} = \text{MIN.}$	$I_I = -1E-4$		1.5			-1.5		
V_{OH}	High-level output voltage	$V_{CC} = \text{MIN.}$ $V_{IL} = V_{IL \text{ max.}}$	$V_{IH} = 2 \text{ V}$ $I_{OH} = -400 \mu\text{A}$	2.5	3.4		2.7	3.4		V
V_{OL}	Low-level output voltage	$V_{CC} = \text{MIN.}$ $V_{IL} = V_{IL \text{ max.}}$	$V_{IH} = 2 \text{ V}$ $I_{OL} = 4 \text{ mA}$ $I_{OL} = 8 \text{ mA}$	0.25	0.4		0.25	0.4		V
I_I	Input current at maximum input voltage	Any reset	$V_{CC} = \text{MAX.}$		0.1			0.1		mA
		CKA	$V_I = 7 \text{ V}$		0.2			0.2		
		CKB	$V_{CC} = \text{MAX.}$		0.4			0.4		
I_{IH}	High-level input current	Any reset	$V_{CC} = \text{MAX.}$		20			20		μ A
		CKA	$V_I = 2.7 \text{ V}$		40			40		
		CKB			80			80		
I_{IL}	Low-level input current	Any reset	$V_{CC} = \text{MAX.}$		0.4			-0.4		mA
		CKA	$V_I = 0.4 \text{ V}$		2.4			-2.4		
		CKB			-3.2			-3.2		
I_{OS}	Short-circuit output current ³	$V_{CC} = \text{MAX.}$		-70	100		70	-100		mA
I_{CC}	Supply current	$V_{CC} = \text{MAX.}$	See Note 3		9	15		9	15	mA
					9	15		9	15	

¹ For conditions shown as MIN. or MAX., use the appropriate value specified under recommended operating conditions.

² All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

³ Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

⁴ QA outputs are tested at specified I_{OL} plus the limit value of I_{IL} for the CKB input. This permits driving the CKB input while maintaining full fan-out capability.

NOTE 3 I_{CC} is measured with all outputs open, both RC inputs grounded following momentary connection to 4.5 V, and all other inputs grounded.

**SN54LS90, SN54LS92, SN54LS93,
SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS**

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

PARAMETER		TEST CONDITIONS ¹	SN54LS93			SN74LS93			UNIT
			MIN	TYP ²	MAX	MIN	TYP ²	MAX	
V _{IH}	High-level input voltage		2			2			V
V _{IL}	Low-level input voltage				0.7			0.8	V
V _{IK}	Input clamp voltage	V _{CC} = MIN, I _I = 18 mA			1.1			1.5	V
V _{OH}	High-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{IL} max, I _{OH} = -400 μ A	2.5	3.4		2.7	3.4		V
V _{OL}	Low-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{IL} max	I _{OL} = 4 mA ³			0.25 0.4			V
		I _{OL} = 8 mA ³			0.35 0.5				
I _I	Input current at maximum input voltage	Any reset	V _{CC} = MAX, V _I = 7 V			0.1			mA
		CKA or CKB	V _{CC} = MAX, V _I = 5.5 V			0.2			
I _{IH}	High-level input current	Any reset	V _{CC} = MAX, V _I = 7 V			20			μ A
		CKA or CKB	V _{CC} = MAX, V _I = 7 V			40			
I _{IL}	Low-level input current	Any reset	V _{CC} = MAX, V _I = 0.4 V			0.4			mA
		CKA	V _{CC} = MAX, V _I = 0.4 V			-2.4			
		CKB	V _{CC} = MAX, V _I = 0.4 V			-1.6			
I _{OS}	Short-circuit output current	V _{CC} = MAX	20		-100	20		-100	mA
I _{CC}	Supply current	V _{CC} = MAX, See Note 3	9	15		9	15		mA

¹For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

²All typical values are at V_{CC} = 5 V, T_A = 25 $^{\circ}$ C.

³Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

⁴O_A outputs are tested at specified I_{OL} plus the limit value for I_{IL} for the CKB input. This permits driving the CKB input while maintaining full fan out capability.

NOTE 3: I_{CC} is measured with all outputs open (both R₀ inputs grounded following momentary connection to 4.5 V, and all other inputs grounded).

switching characteristics, V_{CC} = 5 V, T_A = 25 $^{\circ}$ C

PARAMETER ¹	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	LS90			LS92			LS93			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f _{max}	CKA	O _A	C _L = 15 pF R _L = 2 k Ω See Figure 1	12	47		12	42		12	42		MHz
	CKB	O _B		10			10			10			
t _{PLH}	CKA	O _A		10	16		10	16		10	16		ns
		O _B		12	18		12	18		12	18		
		O _C		12	48		12	48		12	48		ns
		O _D		12	48		12	48		12	48		
t _{PHL}	CKB	O _B		10	16		10	16		10	16		ns
		O _C		14	21		14	21		14	21		
		O _D		21	32		21	32		21	32		ns
		O _A		23	35		23	35		23	35		
t _{PLH}	CKB	O _B		21	32		21	32		21	32		ns
		O _C		21	32		21	32		21	32		
		O _D		23	35		23	35		23	35		ns
		O _A		23	35		23	35		23	35		
t _{PHL}	Set In. 0	Any		26	40		26	40		26	40		ns
		O _A , O _B		20	10								
	Set In. 9	O _C , O _D		26	40								ns
		O _A , O _B		26	40								

¹f_{max} = maximum count frequency

t_{PLH} = propagation delay time, low to high-level output

t_{PHL} = propagation delay time, high to low-level output

**TEXAS
INSTRUMENTS**

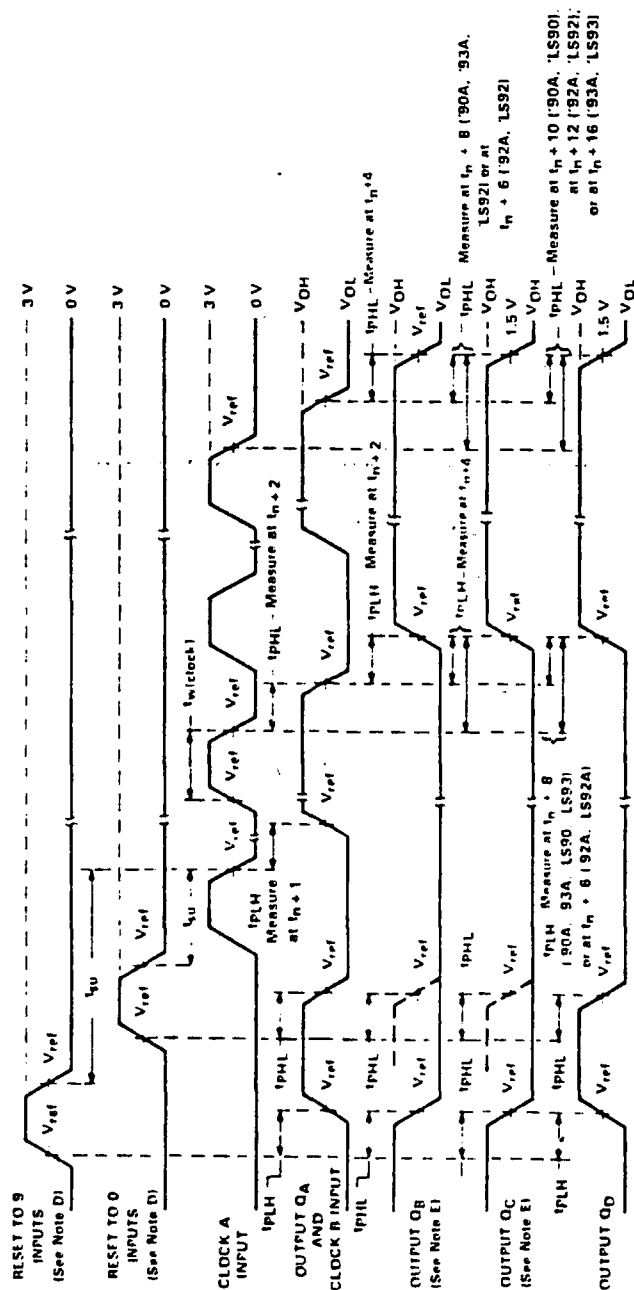
POST OFFICE BOX 655512 • DALLAS, TEXAS 75265

2-285

TTL Devices

2

PARAMETER MEASUREMENT INFORMATION



NOTES

A Input pulses are supplied by a generator having the following characteristics: for '90A', '92A', '93A', $t_r \leq 5$ ns, $t_f \leq 5$ ns, $\text{PWR} = 1$ MW, duty cycle = 50%; for 'LS90', 'LS92', 'LS93', $t_r \leq 15$ ns, $t_f \leq 5$ ns, $\text{PWR} = 1$ MW, duty cycle = 50%, $Z_{\text{out}} = 50$ ohms.

B C_L includes probe and jig capacitance.

C All diodes are 1N3064 or equivalent.

D Each reset input is tested separately with the other reset at 4 V.

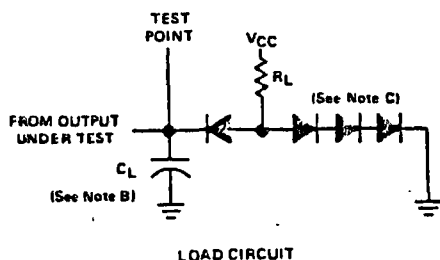
E Reference waveforms are shown with dashed lines.

F for '90A', '92A', and '93A', $V_{\text{ref}} = 1.5$ V for 'LS90', 'LS92', and 'LS93', $V_{\text{ref}} = 3$ V.

FIGURE 1A

**SN5490A, SN5492A, SN5493A, SN54LS90, SN54LS92, SN54LS93;
SN7490A, SN7492A, SN7493A, SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS**

PARAMETER MEASUREMENT INFORMATION



- TES:** A. Input pulses are supplied by a generator having the following characteristics:
for '90A, '92A, '93A, $t_i \leq 5$ ns, $t_f \leq 5$ ns, PRR = 1 MHz, duty cycle = 50%, $Z_{out} = 50$ ohms.
for 'LS90, 'LS92, 'LS93, $t_i \leq 15$ ns, $t_f \leq 5$ ns, PRR = 1 MHz, duty cycle = 50%, $Z_{out} = 50$ ohms
- B. C_L includes probe and jig capacitance.
- C. All diodes are 1N3064 or equivalent.
- D. Each reset input is tested separately with the other reset at 4.5 V.
- E. Reference waveforms are shown with dashed lines.
- F. For '90A, '92A, and '93A, $V_{ref} = 1.5$ V. For 'LS90, 'LS92, and 'LS93, $V_{ref} = 1.3$ V.

FIGURE 1B

TTL Devices

**SN5490A, SN5492A, SN5493A, SN54LS90, SN54LS92, SN54LS93,
SN7490A, SN7492A, SN7493A, SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE AND BINARY COUNTERS**
MARCH 1974—REVISED MARCH 1988

'90A, 'LS90 . . . Decade Counters

'92A, 'LS92 . . . Divide By-Twelve Counters

'93A, 'LS93 . . . 4-Bit Binary Counters

TYPES	TYPICAL POWER DISSIPATION
'90A	145 mW
'92A, '93A	130 mW
'LS90, 'LS92, 'LS93	45 mW

description

Each of these monolithic counters contains four master-slave flip-flops and additional gating to provide a divide-by-two counter and a three-stage binary counter for which the count cycle length is divide-by-five for the '90A and 'LS90, divide-by-six for the '92A and 'LS92, and the divide-by-eight for the '93A and 'LS93.

All of these counters have a gated zero reset and the '90A and 'LS90 also have gated set-to-nine inputs for use in BCD nine's complement applications.

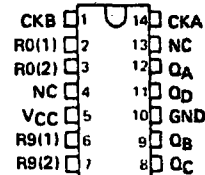
To use their maximum count length (decade, divide-by-twelve, or four-bit binary) of these counters, the CKB input is connected to the QA output. The input count pulses are applied to CKA input and the outputs are as described in the appropriate function table. A symmetrical divide-by-ten count can be obtained from the '90A or 'LS90 counters by connecting the QD output to the CKA input and applying the input count to the CKB input which gives a divide-by-ten square wave at output QA.

SN5490A, SN54LS90 . . . J OR W PACKAGE

SN7490A . . . N PACKAGE

SN74LS90 . . . D OR N PACKAGE

(TOP VIEW)

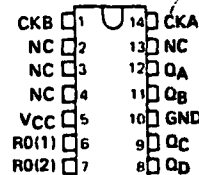


SN5492A, SN54LS92 . . . J OR W PACKAGE

SN7492A . . . N PACKAGE

SN74LS92 . . . D OR N PACKAGE

(TOP VIEW)

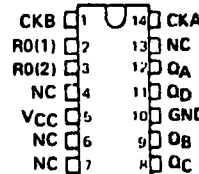


SN5493A, SN54LS93 . . . J OR W PACKAGE

SN7493 . . . N PACKAGE

SN74LS93 . . . D OR N PACKAGE

(TOP VIEW)



NC—No internal connection



TTL Devices

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

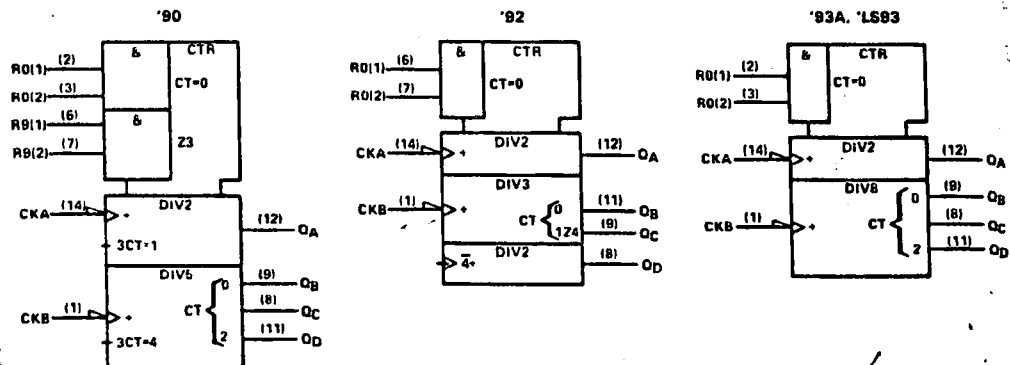
**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

2-277

**SN5490A, '92A, '93A, SN54LS90, 'LS92, 'LS93,
SN7490A, '92A, '93A, SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS**

logic symbols¹



¹These symbols are in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.



TTL Devices

SN5490A, '92A, '93A, SN54LS90, 'LS92, 'LS93,
SN7490A, '92A, '93A, SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

'90A, 'LS90
BCD COUNT SEQUENCE
(See Note A)

COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

'92A, 'LS92
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	H	L	L	L
7	H	L	L	H
8	H	L	H	L
9	H	L	H	H
10	H	H	L	L
11	H	H	L	H

'92A, 'LS92, '93A, 'LS93
RESET/COUNT FUNCTION TABLE

RESET INPUTS		OUTPUT			
R ₀ (1)	R ₀ (2)	Q _D	Q _C	Q _B	Q _A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

NOTES: A. Output Q_A is connected to input CKB for BCD count.
B. Output Q_D is connected to input CKA for bi-quinary count.
C. Output Q_A is connected to input CKB.
D. H = high level, L = low level, X = irrelevant

'90A, 'LS90
BI-QUINARY (5-2)
(See Note B)

COUNT	OUTPUT			
	Q _A	Q _D	Q _C	Q _B
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

'90A, 'LS90
RESET/COUNT FUNCTION TABLE

RESET INPUTS				OUTPUT			
R ₀ (1)	R ₀ (2)	R ₉ (1)	R ₉ (2)	Q _D	Q _C	Q _B	Q _A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

'93A, 'LS93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

2

TTL Devices

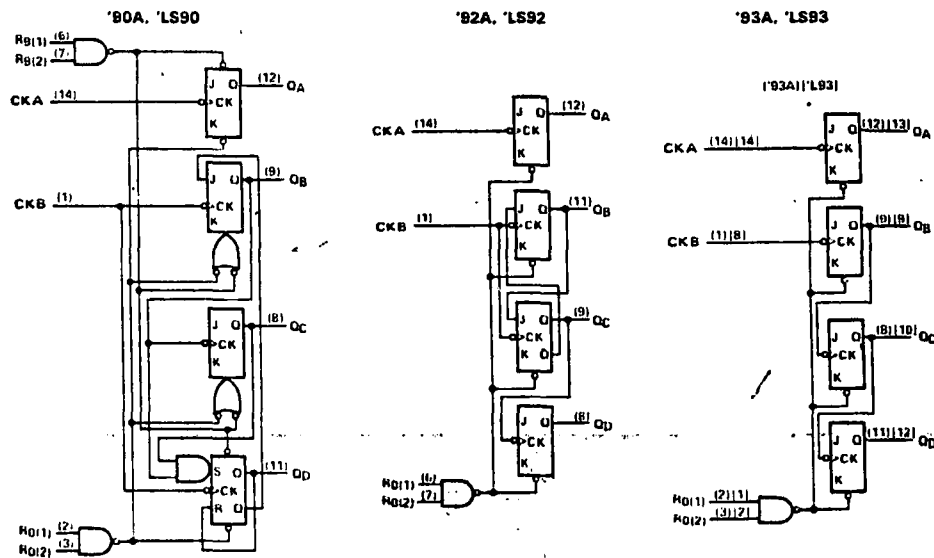
TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75266

2-279

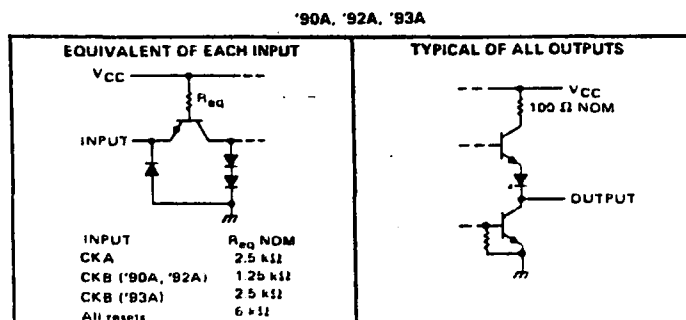
**SN5490A, '92A, '93A, SN54LS90, 'LS92, 'LS93,
SN7490A, '92A, '93A, SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS**

logic diagrams (positive logic)



The J and K inputs shown without connection are for reference only and are functionally at a high level.
Pin numbers shown in () are for the 'LS93 and '93A and pin numbers shown in [] are for the 54LS93.

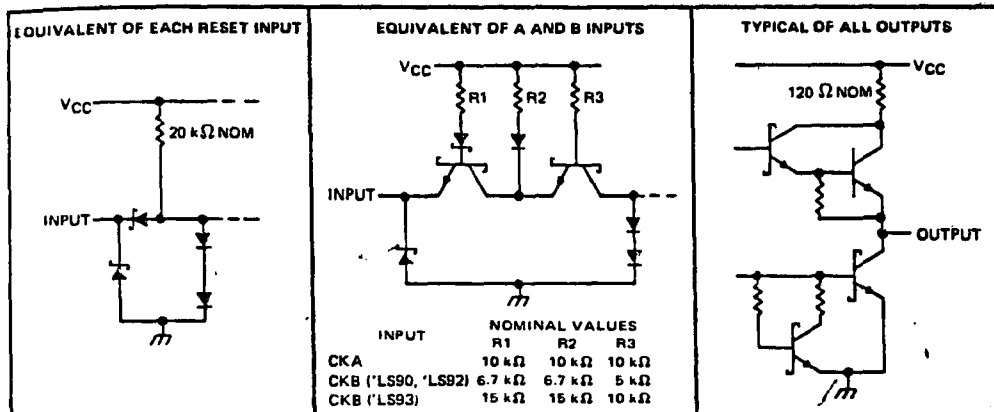
schematics of inputs and outputs



SN54LS90, 'LS92, 'LS93,
SN74LS90, 'LS92, 'LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

schematics of inputs and outputs (continued)

'LS90, 'LS92, 'LS93



TTL Devices

SN5490A, SN5492A, SN5493A, SN7490A, SN7492A, SN7493A DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	5.5 V
Intermitter voltage (see Note 2)	5.5 V
Operating free-air temperature range	SN5490A, SN5492A, SN5493A -55°C to 125°C SN7490A, SN7492A, SN7493A 0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTES: 1. Voltage values, except intermitter voltage, are with respect to network ground terminal.
2. This is the voltage between two emitters of an emitter-coupled transistor. For these inputs, this rating applies between the two R_B inputs, and for the 90A output, it also applies between the two R_B inputs.

recommended operating conditions

		SN5490A, SN5492A			SN7490A, SN7492A			UNIT
		SN5493A			SN7493A			
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}				800			800	μ A
Low-level output current, I_{OL}				16			16	mA
Count frequency, f_{count} (see Figure 1)	A input	0		32	0		32	MHz
	B input	0		16	0		16	
Pulse width, t_w	A input	15			15			ns
	B input	30			30			
	Reset inputs	15			15			
Reset inactive state setup time, t_{su}		75			25			ns
Operating free-air temperature, T_A		55		125	0		70	°C

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

PARAMETER ¹		TEST CONDITIONS ¹		'90A		'92A		'93A		UNIT		
				MIN	TYP	MAX	MIN	TYP	MAX		MIN	TYP
V _{IH}	High-level input voltage					2		2		V		
V _{IL}	Low-level input voltage				0.8		0.8		0.8	V		
V _{IC}	Input clamp voltage	V _{CC} MIN, I _I 12 mA			1.5		1.5		1.5	V		
V _{OH}	High-level output voltage	V _{CC} MIN, V _{IH} 2 V, V _{IL} 0.8 V, I _{OH} 800 μA		24	34		24	34		V		
V _{OL}	Low-level output voltage	V _{CC} MIN, V _{IH} 2 V, V _{IL} 0.8 V, I _{OL} 16 mA ²		0.2	0.4		0.2	0.4		V		
I _I	Input current at maximum input voltage	V _{CC} MAX, V _I 5.5 V			1		1		1	mA		
I _{IH}	High-level input current	Any reset			40		40		40	μA		
		CKA	V _{CC} MAX, V _I 2.4 V		80		80		80			
		CKB			120		120		120			
I _{IL}	Low-level input current	Any reset			16		16		16	mA		
		CKA	V _{CC} MAX, V _I 0.4 V		32		32		32			
		CKB			48		48		48			
I _{OS}	Short-circuit output current	V _{CC} MAX	SN54	25	57	20	57	20	57	mA		
			SN74	18	57	18	57	18	57			
I _{CC}	Supply current	V _{CC} MAX, See Table 3		29	42		26	39		26	39	mA

¹ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

All typical values are at V_{CC} 5 V, T_A 25°C.

Not more than one output should be shorted at a time.

² OA outputs are tested at I_{OL} 16 mA plus the unit value for I_{OL} for the CKB input. This permits driving the CKB input while maintaining full fan-out capability.

NOTE 3: I_{CC} is measured with all outputs open, R_B inputs shorted following momentary connection to 4.5 V, and all other inputs grounded.

SN5490A, SN5492A, SN5493A, SN7490A, SN7492A, SN7493A
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

switching characteristics, VCC = 5 V, TA = 25 C

PARAMETER ¹	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'90A			'92A			'93A			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}	CKA	QA	CL 15 pF. RI 400 Ω See Figure 1	32	42		32	42		32	42		MHz
	CKB	QH		16			16			16			
tPLH	CKA	QA		10	16		10	16		10	16		ns
tPHL				12	18		12	18		12	18		
tPLH	CKA	QD		32	48		32	48		46	70		ns
tPHL				34	50		34	50		46	70		
tPLH	CKB	QB		10	16		10	16		10	16		ns
tPHL				14	21		14	21		14	21		
tPLH	CKB	QC		21	32		10	16		21	32		ns
tPHL				23	35		14	21		23	35		
tPLH	CKB	QD		21	32		21	32		34	51		ns
tPHL				23	35		23	35		34	51		
tPLH	Set to 0	Any		26	40		26	40		26	40		ns
tPHL	Set to 0	QA, QD		20	30								
tPHL				QB, QC	26	40							

¹f_{max} = maximum count frequency

t_{PLH} = propagation delay time, low to high level output

t_{PHL} = propagation delay time, high to low level output

SN54LS90, SN54LS92, SN54LS93,
SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage: R inputs	7 V
A and B inputs	5.5 V
Operating free-air temperature range: SN54LS [†] Circuits	-55°C to 125°C
SN74LS [†] Circuits	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal

recommended operating conditions

		SN54LS90 SN54LS92 SN54LS93			SN74LS90 SN74LS92 SN74LS93			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
		4.5	5	5.5	4.75	5	5.25	
Supply voltage, V_{CC}								V
High-level output current, I_{OH}				-400			-400	μ A
Low-level output current, I_{OL}				4			8	mA
Count frequency, f_{COUNT} (see Figure 1)	A input	0	32		0	32		MHz
	B input	0	16		0	16		
Pulse width, t_{PW}	A input	15		15				ns
	B input	30		30				
	Reset inputs	30		30				
Reset inactive state setup time, t_{SU}		25		25				ns
Operating free-air temperature, T_A		55	125		0	70		°C

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

PARAMETER		TEST CONDITIONS [†]		SN54LS90 SN54LS92			SN74LS90 SN74LS92			UNIT
				MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V_{IH}	High-level input voltage			2			2			V
V_{IL}	Low-level input voltage					0.7			0.8	V
V_{IK}	Input clamp voltage	V_{CC} MIN, $I_I = 18$ mA				-1.5			-1.5	V
V_{OH}	High-level output voltage	V_{CC} MIN, $V_{IH} = 2$ V, $V_{IL} = V_{IL\max}$, $I_{OH} = 400$ μ A		2.5	3.4		2.5	3.4		V
V_{OL}	Low-level output voltage	V_{CC} MIN, $V_{IH} = 2$ V, $V_{IL} = V_{IL\max}$, $I_{OL} = 4$ mA [*] $I_{OL} = 8$ mA [*]		0.25	0.4		0.25	0.4		V
I_I	Input current at maximum input voltage	Any reset	$V_{CC} = \text{MAX}$, $V_I = 7$ V			0.1			0.1	mA
		CKA	$V_{CC} = \text{MAX}$, $V_I = 5$ V			0.2			0.2	
		CKB	$V_{CC} = \text{MAX}$, $V_I = 5$ V			0.4			0.4	
I_{IH}	High-level input current	Any reset	$V_{CC} = \text{MAX}$, $V_I = 2.7$ V			20			20	μ A
		CKA	$V_{CC} = \text{MAX}$, $V_I = 2.7$ V			40			40	
		CKB	$V_{CC} = \text{MAX}$, $V_I = 2.7$ V			80			80	
I_{IL}	Low-level input current	Any reset	$V_{CC} = \text{MAX}$, $V_I = 0.4$ V			-0.4			-0.4	mA
		CKA	$V_{CC} = \text{MAX}$, $V_I = 0.4$ V			-2.4			-2.4	
		CKB	$V_{CC} = \text{MAX}$, $V_I = 0.4$ V			-3.2			-3.2	
I_{OS}	Short-circuit output current [§]	$V_{CC} = \text{MAX}$		-20	-100	-20	-100			mA
I_{CC}	Supply current	$V_{CC} = \text{MAX}$, See Note 3	LS90		9	15		9	15	mA
			LS92		9	15		9	15	

[†]For conditions shown as MIN, or MAX, use the appropriate value specified under recommended operating conditions

[‡]All typical values are at $V_{CC} = 5$ V, $T_A = 25^\circ\text{C}$.

[§]Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one pulse width.

^{*} Q_A outputs are tested at specified I_{OL} plus the limit value of I_{OL} for the CKB input. This permits driving the CKB input while maintaining full fan-out capability.

NOTE 3: I_{CC} is measured with all outputs open, both R_C inputs grounded following momentary connection to 4.5 V, and all other inputs grounded.

**SN54LS90, SN54LS92, SN54LS93
SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS**

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

PARAMETER	TEST CONDITIONS ¹	SN54LS93			SN74LS93			UNIT
		MIN	TYP ²	MAX	MIN	TYP ²	MAX	
High-level input voltage		2			2			V
Low-level input voltage				0.7			0.8	V
Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = 18 \text{ mA}$			1.5			-1.5	V
High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V.}$ $V_{IL} = V_{IL \text{ max.}}$, $I_{OH} = 400 \mu\text{A}$	2.5	3.4		2.7	3.4		V
Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V.}$ $V_{IL} = V_{IL \text{ max.}}$							V
	$I_{OL} = 4 \text{ mA}^4$		0.25	0.4		0.25	0.4	
Low-level output voltage	$I_{OL} = 8 \text{ mA}^4$					0.35	0.5	V
Input current at maximum output voltage	Any reset $V_{CC} = \text{MAX.}$, $V_I = 7 \text{ V}$			0.1			0.1	mA
High-level input current	CKA or CKB $V_{CC} = \text{MAX.}$, $V_I = 5.5 \text{ V}$			0.2			0.2	
Low-level input current	Any reset $V_{CC} = \text{MAX.}$, $V_I = 2.7 \text{ V}$			20			20	μA
	CKA or CKB $V_{CC} = \text{MAX.}$, $V_I = 2.7 \text{ V}$			40			80	
Short-circuit output current ³	Any reset $V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$			0.4			0.4	mA
	CKA $V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$			-2.4			-2.4	
Supply current	CKB $V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$			1.6			1.6	mA
	Short-circuit output current ³ $V_{CC} = \text{MAX.}$	20		-100	20		-100	
Supply current	$V_{CC} = \text{MAX.}$, See Note 3		9	15		9	15	mA

¹ Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

² Typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

³ No more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

⁴ Outputs are tested at specified I_{OL} plus the limit value for I_{IL} for the CKB input. This permits driving the CKB input while maintaining full output capability.

⁵ I_{CC} is measured with all outputs open, both R_D inputs grounded following recommended connection to 4.7 Ω , and all other inputs grounded.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER ¹	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	LS90			LS92			LS93			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}	CKA	Q_A	$C_L = 15 \text{ pF}$ $R_L = 2 \text{ k}\Omega$ See Figure 1	32	42		32	42		32	42		MHz
	CKB	Q_B		16			16			16			
t_{PLH}	CKA	Q_A		10	16		10	16		10	16		ns
t_{PHL}	CKA	Q_A		12	18		12	18		12	18		
t_{PLH}	CKA	Q_D		32	48		32	48		46	70		ns
t_{PHL}	CKA	Q_D		34	50		34	50		46	70		
t_{PLH}	CKB	Q_B		10	16		10	16		10	16		ns
t_{PHL}	CKB	Q_B		14	21		14	21		14	21		
t_{PLH}	CKB	Q_C		21	32		10	16		21	32		ns
t_{PHL}	CKB	Q_C		23	35		14	21		23	35		
t_{PLH}	CKB	Q_D		21	32		21	32		34	51		ns
t_{PHL}	CKB	Q_D		23	35		23	35		34	51		
t_{PLH}	Set to 0	Any		26	40		26	40		26	40		ns
t_{PHL}	Set to 0	Q_A , Q_D		20	30								
t_{PHL}	Set to 0	Q_B , Q_C		26	40								ns

¹ f_{max} = maximum count frequency

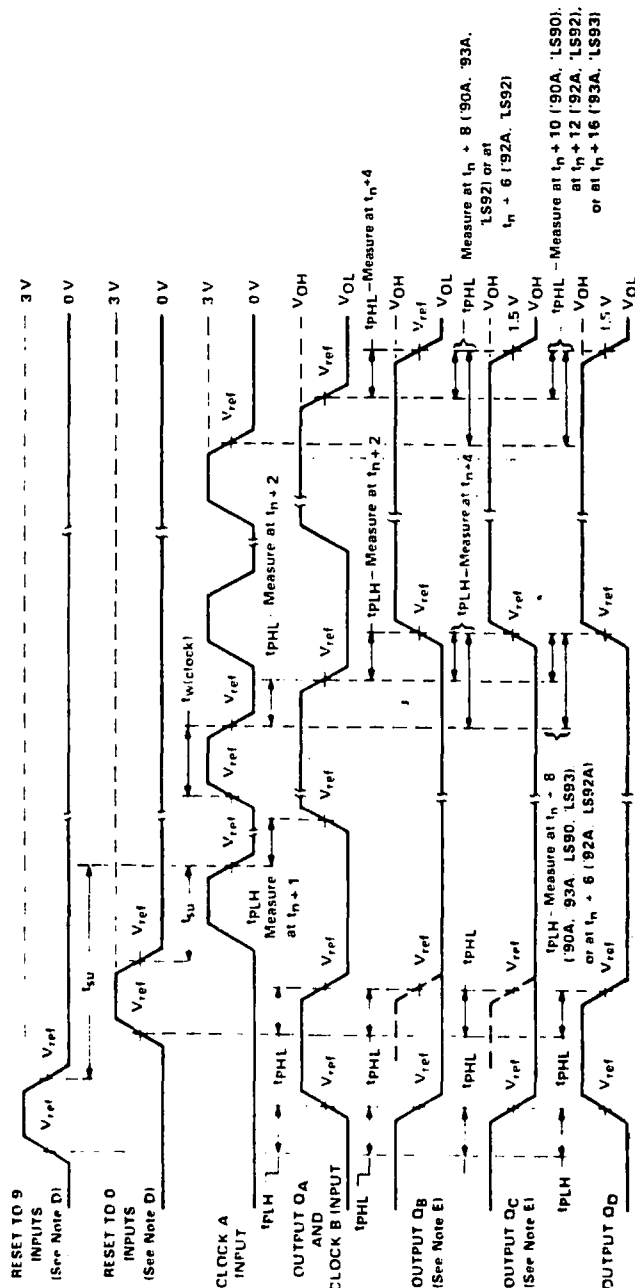
t_{PLH} = propagation delay time, low to high-level output

t_{PHL} = propagation delay time, high to low-level output

SN5490A, SN5492A, SN5493A, SN54LS90, SN54LS92, SN54LS93,
SN7490A, SN7492A, SN7493A, SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

TTL Devices

PARAMETER MEASUREMENT INFORMATION

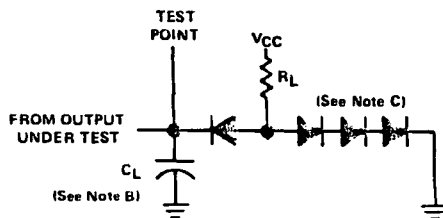


- NOTES
- Input pulses are supplied by a generator having the following characteristics:
for 90A, 92A, 93A $t_r \leq 5$ ns, $t_f \leq 5$ ns, PRR = 1 MHz, duty cycle = 50%, $Z_{out} = 50$ ohms.
for LS90, LS92, LS93 $t_r \leq 15$ ns, $t_f \leq 5$ ns, PRR = 1 MHz, duty cycle = 50%, $Z_{out} = 50$ ohms.
 - C_L includes probe and jig capacitance.
 - All diodes are 1N3064 or equivalent.
 - Each reset input is tested separately with the other reset at 4.5 V.
 - Reference waveforms are shown with dashed lines.
 - For 90A, 92A, and 93A $V_{ref} = 1.5$ V. For LS90, LS92, and LS93 $V_{ref} = 1.3$ V.

FIGURE 1A

**SN5490A, SN5492A, SN5493A, SN54LS90, SN54LS92, SN54LS93,
SN7490A, SN7492A, SN7493A, SN74LS90, SN74LS92, SN74LS93
DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS**

PARAMETER MEASUREMENT INFORMATION



LOAD CIRCUIT

- NOTES: A Input pulses are supplied by a generator having the following characteristics:
for '90A, '92A, '93A, $t_r \leq 5$ ns, $t_f \leq 5$ ns, PRR = 1 MHz, duty cycle = 50%, $Z_{out} \leq 50$ ohms
for 'LS90, 'LS92, 'LS93, $t_r \leq 15$ ns, $t_f \leq 5$ ns, PRR = 1 MHz, duty cycle = 50%, $Z_{out} \leq 50$ ohms
B C_L includes probe and jig capacitance
C All diodes are 1N3064 or equivalent
D Each reset input is tested separately with the other reset at 4.5 V
E Reference waveforms are shown with dashed lines
F For '90A, '92A, and '93A, $V_{ref} = 1.5$ V. For 'LS90, 'LS92, and 'LS93, $V_{ref} = 1.3$ V

FIGURE 1B



TTL Devices

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

2-287

SN54LS138, SN54S138, SN74LS138, SN74S138A 3-LINE TO 8-LINE DECODERS/DEMULPLEXERS

DECEMBER 1972—REVISED MARCH 1988

- Designed Specifically for High-Speed: Memory Decoders Data Transmission Systems
- 3 Enable Inputs to Simplify Cascading and/or Data Reception
- Schottky-Clamped for High Performance

description

These Schottky-clamped TTL MSI circuits are designed to be used in high-performance memory decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems, these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast enable circuit, the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the Schottky-clamped system decoder is negligible.

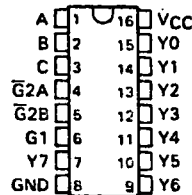
The 'LS138, SN54S138, and SN74S138A decode one of eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

All of these decoder/demultiplexers feature fully buffered inputs, each of which represents only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and to simplify system design.

The SN54LS138 and SN54S138 are characterized for operation over the full military temperature range of -55°C to 125°C . The SN74LS138 and SN74S138A are characterized for operation from 0°C to 70°C .

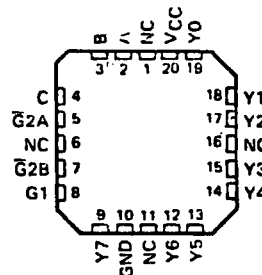
SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138A . . . D OR N PACKAGE

(TOP VIEW)



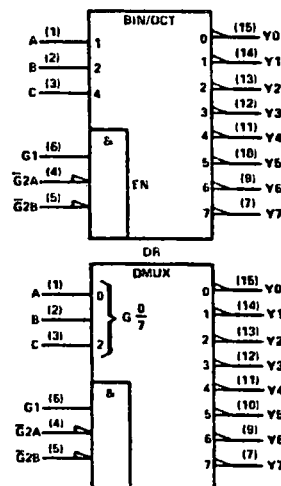
SN54LS138, SN54S138 . . . FK PACKAGE

(TOP VIEW)



NC—No internal connection

logic symbols†



†These symbols are in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

Pin numbers shown are for D, J, N, and W packages.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

Copyright © 1972, Texas Instruments Incorporated

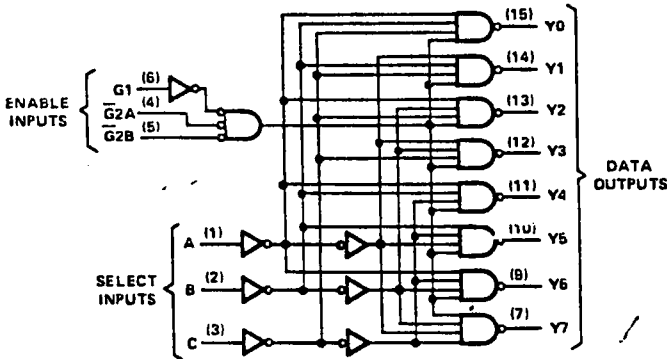
2-425

TTL Devices

SN54LS138, SN54S138, SN74LS138, SN74S138A
3-LINE-TO 8-LINE DECODERS/DEMULTIPLEXERS

logic diagram and function table

'LS138, SN54S138, SN74S138A



Pin numbers shown are for D, J, N, and W packages.

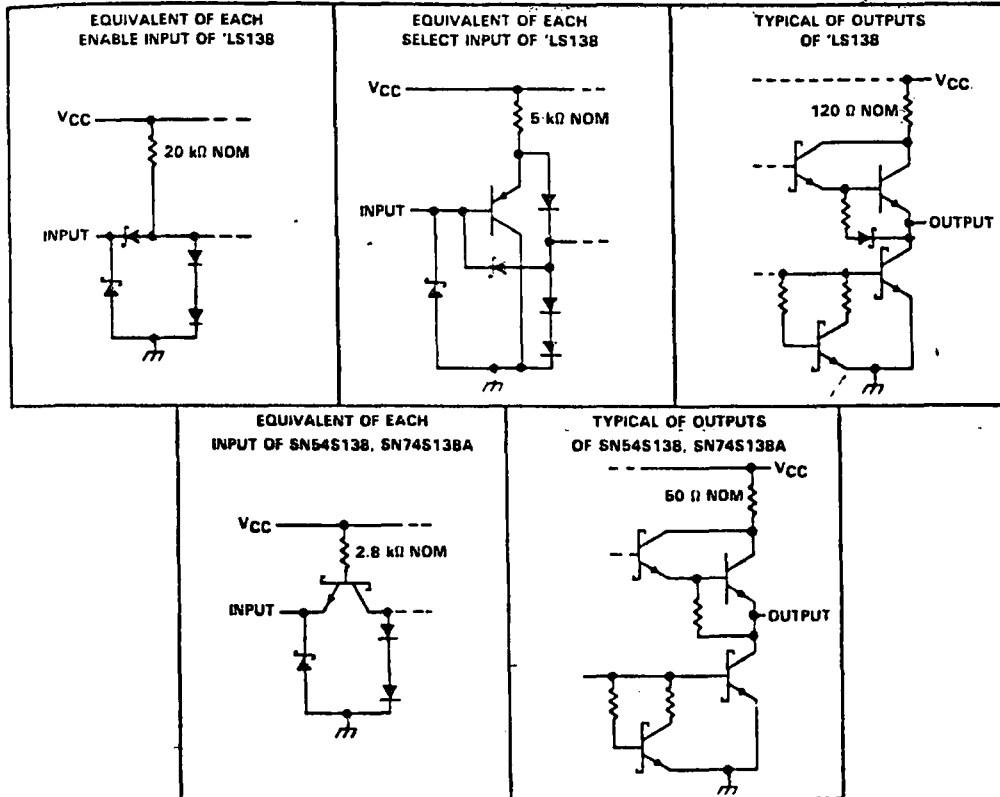
'LS138, SN54S138, SN74S138A
 FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	L	H	H	H	H
H	L	H	L	H	H	H	H	L	H	H	H	H
H	L	H	H	L	H	H	H	H	L	H	H	H
H	L	H	H	H	H	H	H	H	H	L	H	H

* $G2 = G2A + G2B$
 H = high level, L = low level, X = irrelevant

SN54LS138, SN54S138, SN74LS138, SN74S138A 3-LINE TO 8-LINE DECODERS/DEMULTIPLEXERS

schematics of inputs and outputs



TTL Devices

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V _{CC} (see Note 1)	7 V
Input voltage	7 V
Operating free-air temperature range: SN54LS138, SN54S138	-55°C to 125°C
SN74LS138, SN74S138A	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

SN54LS138, SN74LS138 3-LINE TO 8-LINE DECODERS/DEMULTIPLEXERS

recommended operating conditions

		SN54LS138			SN74LS138			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH}	High-level input voltage	2			2			V
V _{IL}	Low-level input voltage			0.7			0.8	V
I _{OH}	High-level output current			-0.4			-0.4	mA
I _{OL}	Low-level output current			4			8	mA
T _A	Operating free-air temperature	-55		125	0		70	°C

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

PARAMETER	TEST CONDITIONS ¹	SN54LS138			SN74LS138			UNIT
		MIN	TYP ²	MAX	MIN	TYP ²	MAX	
V _{IK}	V _{CC} = MIN, I _I = -18 mA			-1.5			-1.5	V
V _{OH}	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = MAX, I _{OH} = -0.4 mA	2.5	3.4		2.7	3.4		V
V _{OL}	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = MAX			0.25			0.25	V
I _I	V _{CC} = MAX, V _I = 7 V			0.1			0.1	mA
I _{IH}	V _{CC} = MAX, V _I = 2.7 V			20			20	µA
I _{IL}	V _{CC} = MAX, V _I = 0.4 V			-0.4			-0.4	mA
I _{OS} ³	V _{CC} = MAX	-20		-100	-20		-100	mA
I _{CC}	V _{CC} = MAX, Outputs enabled and open	6.3	10		6.3	10		mA

¹ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

² All typical values are at V_{CC} = 5 V, T_A = 25°C.

³ Not more than one output should be shorted at a time, and duration of the short-circuit test should not exceed one second.

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

PARAMETER ¹	FROM (INPUT)	TO (OUTPUT)	LEVELS OF DELAY	TEST CONDITIONS	SN54LS138 SN74LS138			UNIT
					MIN	TYP	MAX	
t _{PLH}	Binary Select	Any	2	R _L = 2 kΩ, C _L = 15 pF, See Note 2		11	20	ns
t _{PHL}						18	41	ns
t _{PLH}			3			21	27	ns
t _{PHL}						20	39	ns
t _{PLH}	Enable	Any	2			12	18	ns
t _{PHL}						20	32	ns
t _{PLH}			3			14	26	ns
t _{PHL}						13	38	ns

¹ t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

NOTE 2: Load circuits and voltage waveforms are shown in Section 1.

SN54S138, SN74S138A
3-LINE TO 8-LINE DECODERS/DEMULPLEXERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, VCC (see Note 1)	7 V
Input voltage	5.5 V
Operating free-air temperature range: SN54S138	-55°C to 125°C
SN74S138A	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

		SN54S138			SN74S138A			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
VCC	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH}	High-level input voltage	2			2			V
V _{IL}	Low-level input voltage			0.8			0.8	V
I _{OH}	High-level output current			-1			-1	mA
I _{OL}	Low-level output current			20			20	mA
T _A	Operating free-air temperature	-55		125	0		70	°C

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

PARAMETER	TEST CONDITIONS ¹	SN54S138 SN74S138A			UNIT
		MIN	TYP ²	MAX	
V _{IK}	VCC = MIN, I _I = -18 mA			-1.2	V
V _{OH}	VCC = MIN, V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OH} = -1 mA	SN54S ³	2.5	3.4	V
		SN74S ³	2.7	3.4	
V _{OL}	VCC = MIN, V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OL} = 20 mA			0.5	V
I _I	VCC = MAX, V _I = 5.5 V			1	mA
I _{IH}	VCC = MAX, V _I = 2.7 V			50	μA
I _{IL}	VCC = MAX, V _I = 0.5 V			-2	mA
I _{OS} ⁴	VCC = MAX		-40	-100	mA
I _{CC}	VCC = MAX, Outputs enabled and open		49	74	mA

¹ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

² All typical values are at VCC = 5 V, T_A = 25°C.

³ Not more than one output should be shorted at a time, and duration of the short circuit test should not exceed one second.



TTL Devices

SN54S138, SN74S138A
3-LINE TO 8-LINE DECODERS/DEMULTIPLEXERS

switching characteristics, VCC = 5 V, TA = 25°C

PARAMETER†	FROM (INPUT)	TO (OUTPUT)	LEVELS OF DELAY	TEST CONDITIONS	SN54S138 SN74S138A			UNIT	
					MIN	TYP	MAX		
tPLH	Binary Select	Any	2	RL = 280 Ω, CL = 15 pF, See Note 2	4.5	7		ns	
tPHL					7	10.5		ns	
tPLH			3		7.5	12		ns	
tPHL					8	12		ns	
tPLH	Enable	Any	2		5	8		ns	
tPHL					7	11		ns	
tPLH			3		7	11		ns	
tPHL					7	11		ns	

¹t_{PLH} = propagation delay time, low-to-high-level output
¹t_{PHL} = propagation delay time, high-to-low-level output
NOTE 2: Load circuits and voltage waveforms are shown in Section 1.

2
TTL Devices

SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244, SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244

OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

APRIL 1985—REVISED MARCH 1988

- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- PNP Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

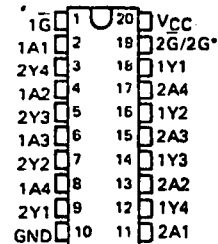
description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical $\bar{G}$ (active-low output control) inputs, and complementary $\bar{G}$ and $\bar{G}$ inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

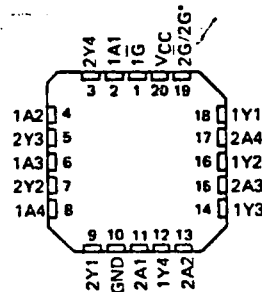
The SN54' family is characterized for operation over the full military temperature range of -55°C to 125°C . The SN74' family is characterized for operation from 0°C to 70°C .

SN54LS', SN54S' ... J OR W PACKAGE
SN74LS', SN74S' ... DW OR N PACKAGE

(TOP VIEW)

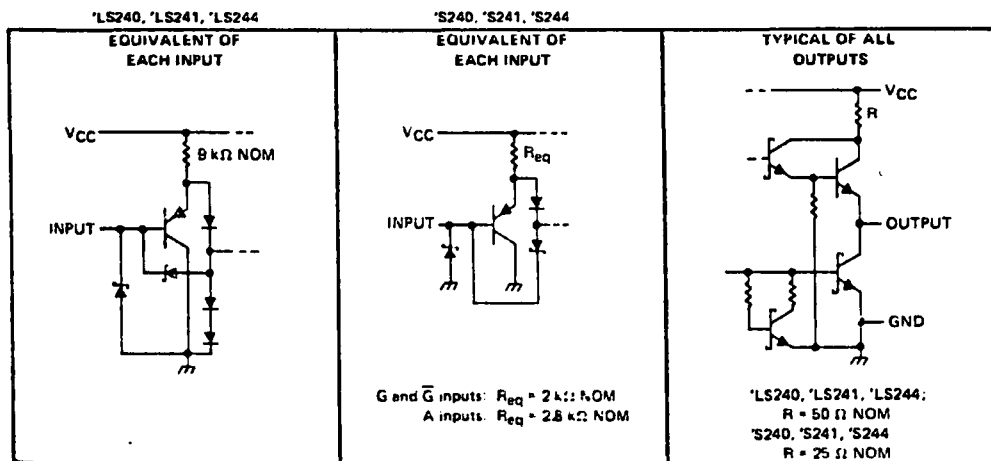


SN54LS', SN54S' ... FK PACKAGE
(TOP VIEW)



*2G for 'LS241 and 'S241 or 2G for all other drivers.

schematics of inputs and outputs



PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

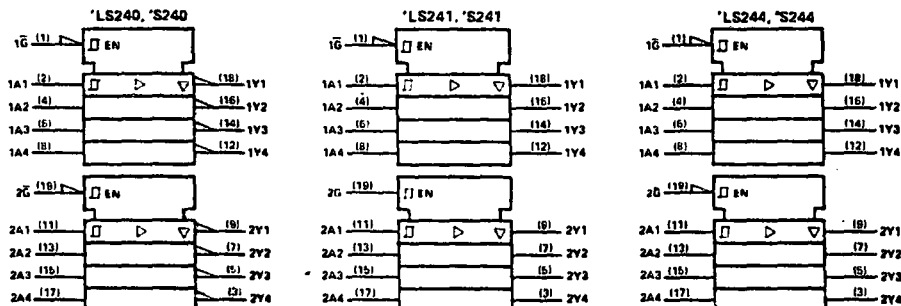
2-691



TTL Devices

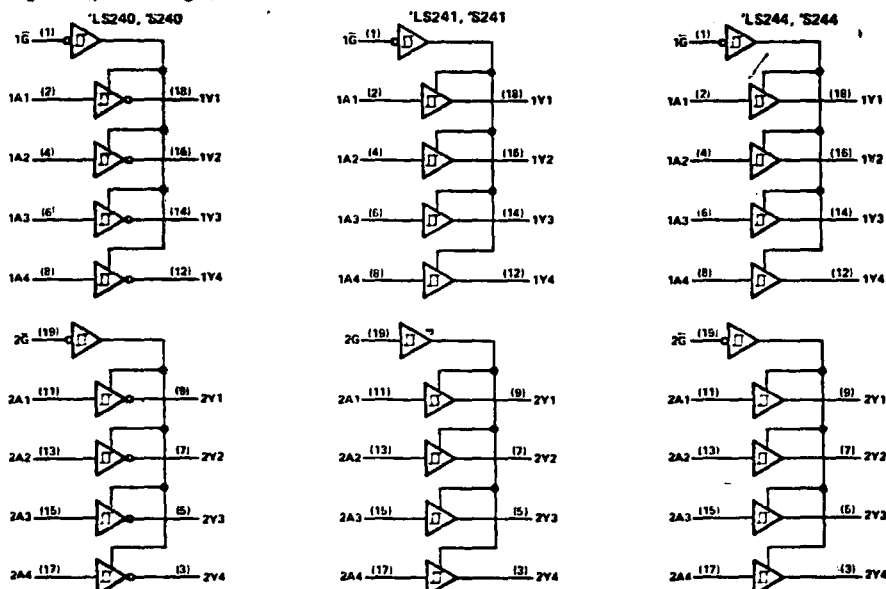
**SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244,
SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS**

logic symbols†



†These symbols are in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.

logic diagrams (positive logic)



Pin numbers shown are for DW, J, N, and W packages.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage: 'LS Circuits	7 V
'S Circuits	5.5 V
Off-state output voltage	5.5 V
Operating free-air temperature range: SN54LS', SN54S' Circuits	- 55 C to 125 C
SN74LS', SN74S' Circuits	0 C to 70 C
Storage temperature range	- 65 C to 150 C

NOTE 1: Voltage values are with respect to network ground terminal.

SN54LS240, SN54LS241, SN54LS244, SN74LS240, SN74LS241, SN74LS244 OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

recommended operating conditions

PARAMETER	SN54LS ¹			SN74LS ¹			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC} Supply voltage (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
V _{IH} High-level input voltage	2			2			V
V _{IL} Low-level input voltage			0.7			0.8	V
I _{OH} High-level output current			-12			-15	mA
I _{OL} Low-level output current			12			24	mA
T _A Operating free-air temperature	-65		125	0		70	°C

NOTE 1: Voltage values are with respect to network ground terminal.

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

PARAMETER	TEST CONDITIONS [†]			SN54LS ¹			SN74LS ¹			UNIT
				MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V _{IK}	V _{CC} = MIN.	I _I = -18 mA				-1.5			-1.5	V
Hysteresis (V _{T+} - V _{T-})	V _{CC} = MIN			0.2	0.4		0.2	0.4		V
V _{OH}	V _{CC} = MIN.	V _{IH} = 2 V.	V _{IL} = MAX.	2.4	3.4		2.4	3.4		V
	V _{CC} = MIN.	V _{IH} = 2 V.	V _{IL} = 0.5 V.	2			2			V
V _{OL}	V _{CC} = MIN.	V _{IH} = 2 V.	I _{OL} = 12 mA			0.4			0.4	V
	V _{IL} = MAX		I _{OL} = 24 mA						0.5	V
I _{OZH}	V _{CC} = MAX.	V _{IH} = 2 V.	V _O = 2.7 V			20			20	μA
I _{OZL}	V _{IL} = MAX		V _O = 0.4 V			-20			-20	μA
I _I	V _{CC} = MAX.	V _I = 7 V				0.1			0.1	mA
I _{IH}	V _{CC} = MAX.	V _I = 2.7 V				20			20	μA
I _{IL}	V _{CC} = MAX.	V _{IL} = 0.4 V				-0.2			-0.2	mA
I _{OS} [§]	V _{CC} = MAX			-40		-225	-40		-225	mA
I _{CC}	Outputs high		All	17	27		17	27		mA
	Outputs low	V _{CC} = MAX.	'LS240	26	44		26	44		
		Output open	'LS241, 'LS244	27	46		27	46		
	All outputs disabled		'LS240	29	50		29	50		
			'LS241, 'LS244	32	54		32	54		

[†] For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡] All typical values are at V_{CC} = 5 V, T_A = 25°C.

[§] Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

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

PARAMETER	TEST CONDITIONS			'LS240			'LS241, 'LS244			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
t _{PLH}	R _L = 667 Ω, See Note 2	C _L = 45 pF.		9	14		12	18		ns
t _{PHL}				12	18		12	18		ns
t _{PZL}				20	30		20	30		ns
t _{PZH}				15	23		15	23		ns
t _{PLZ}	R _L = 667 Ω, See Note 2	C _L = 5 pF.		10	20		10	20		ns
t _{PHZ}				15	25		15	25		ns

NOTE 2: Load circuits and voltage waveforms are shown in Section 1.

TTL Devices

SN54S240, SN54S241, SN54S244, SN74S240, SN74S241, SN74S244, OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

recommended operating conditions

PARAMETER	SN54S*			SN74S*			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC} Supply voltage, (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
V _{IH} High-level input voltage	2			2			V
V _{IL} Low-level input voltage			0.8			0.8	V
I _{OH} High-level output current			-12			-15	mA
I _{OL} Low-level output current			48			64	mA
External resistance between any input and V _{CC} or ground			40			40	k Ω
T _A Operating free-air temperature (see Note 3)	-55		125	0		70	°C

NOTES: 1. Voltage values are with respect to network ground terminal.
3. An SN54S241J operating at free-air temperature above 115°C requires a heat sink that provides a thermal resistance from case to free-air R_{θCA} of not more than 40°C/W.

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

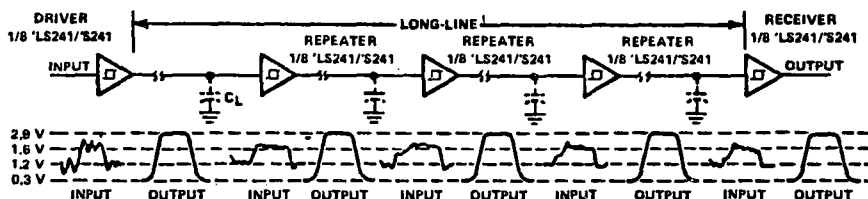
PARAMETER	TEST CONDITIONS†		SN54S*		SN74S*		UNIT
			MIN	MAX	MIN	MAX	
V _{IK}	V _{CC} = MIN, I _I = -18 mA			-1.2		-1.2	V
Hysteresis (V _{T+} - V _{T-})	V _{CC} = MIN		0.2	0.4	0.2	0.4	V
V _{OH}	V _{CC} = MIN, I _{OH} = -1 mA, V _{IH} = 2 V, V _{IL} = 0.8 V				3.7		V
	V _{CC} = MIN, I _{OH} = -3 mA, V _{IH} = 2 V, V _{IL} = 0.8 V		2.4	3.4	2.4	3.4	
	V _{CC} = MIN, I _{OH} = MAX, V _{IH} = 2 V, V _{IL} = 0.5 V		2		2		
V _{OL}	V _{CC} = MIN, I _{OL} = MAX, V _{IH} = 2 V, V _{IL} = 0.8 V			0.55		0.55	V
I _{OZH}	V _{CC} = MAX, V _{IH} = 2 V, V _O = 2.4 V			50		50	μA
I _{OZL}	V _{IL} = 0.8 V, V _O = 0.5 V			-50		-50	
I _I	V _{CC} = MAX, V _I = 5.5 V			1		1	mA
I _{IH}	V _{CC} = MAX, V _I = 2.7 V			50		50	μA
I _{IL}	Any A			-400		-400	μA
	Any G			-2		-2	
I _{OS} ‡	V _{CC} = MAX		-50	-225	-50	-225	mA
I _{CC}	Outputs high	V _{CC} = MAX, Outputs open	'S240	80	123	80	mA
			'S241, 'S244	95	147	95	
	Outputs low		'S240	100	145	100	
			'S241, 'S244	120	170	120	
	Outputs disabled		'S240	100	145	100	
			'S241, 'S244	120	170	120	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

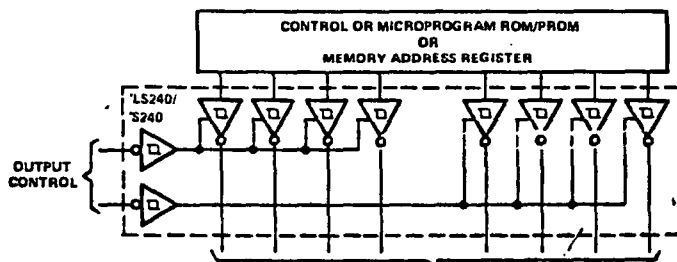
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

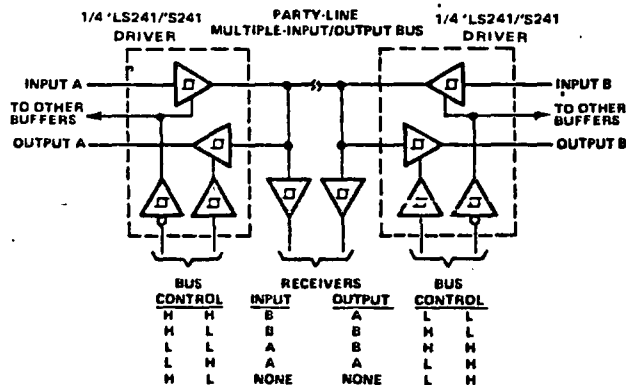
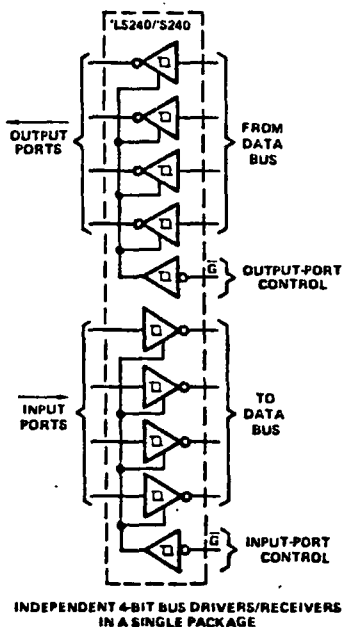
**SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244,
SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS**



'LS241, 'S241 USED AS REPEATER/LEVEL RESTORER



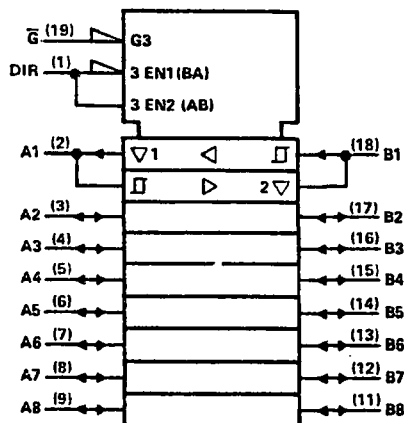
'LS240/'S240 USED AS SYSTEM AND/OR MEMORY BUS DRIVER—4-BIT ORGANIZATION CAN BE APPLIED TO HANDLE BINARY OR BCD



PARTY-LINE BUS SYSTEM WITH MULTIPLE INPUTS, OUTPUTS, AND RECEIVERS

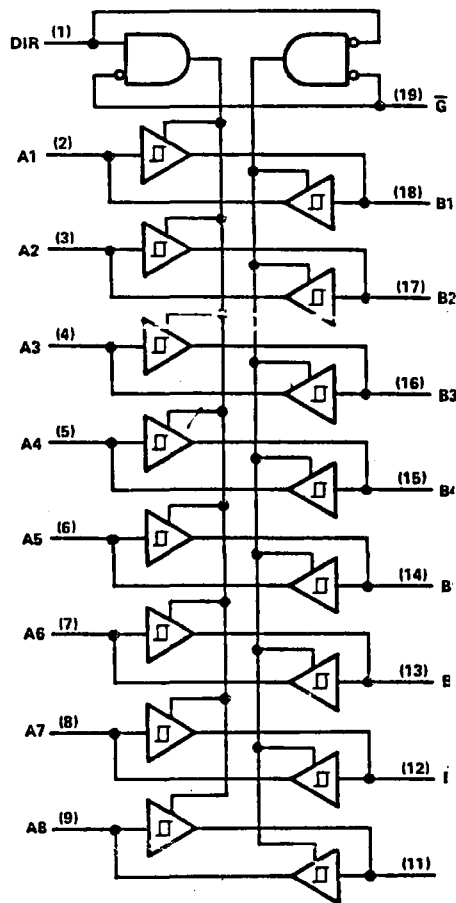
SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

logic symbol†



†This symbol is in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.
 Pin numbers shown are for DW, J, N, and W packages.

logic diagram (positive logic)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	5.0
Input voltage	5.0
Off-state output voltage	5.0
Operating free-air temperature range: SN54LS245	-55°C to 125°C
SN74LS245	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

SN54LS245, SN74LS245 OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

recommended operating conditions

PARAMETER	SN54LS245			SN74LS245			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-12			-15	mA
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	-55		125	0		70	°C

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

PARAMETER		TEST CONDITIONS†		SN54LS245		SN74LS245		UNIT
				MIN	TYP‡	MAX	MIN	
V _{IH}	High-level input voltage			2			2	V
V _{IL}	Low-level input voltage					0.7		0.8
V _{IK}	Input clamp voltage	V _{CC} = MIN,	I _I = -18 mA			-1.5		-1.5
Hysteresis (V _{T+} - V _{T-}) A or E input		V _{CC} = MIN		0.2	0.4		0.2	0.4
V _{OH}	High-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{IL max}	I _{OH} = -3 mA	2.4	2.4		2.4	2.4
			I _{OH} = MAX	2		2		
V _{OL}	Low-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{IL max}	I _{OL} = 12 mA			0.4		0.4
			I _{OL} = 24 mA				0.5	
I _{OZH}	Off-state output current, high-level voltage applied	V _{CC} = MAX, G at 2 V	V _O = 2.7 V		20		20	μA
I _{OZL}	Off-state output current, low-level voltage applied		V _O = 0.4 V		-200		-200	
I _I	Input current at maximum input voltage	A or B DIR or G	V _{CC} = MAX,	V _I = 5.5 V		0.1		0.1
				V _I = 7 V		0.1		0.1
I _{IH}	High-level input current	V _{CC} = MAX,	V _{IH} = 2.7 V		20		20	μA
I _{IL}	Low-level input current	V _{CC} = MAX,	V _{IL} = 0.4 V		-0.2		-0.2	mA
I _{OS}	Short-circuit output current‡	V _{CC} = MAX		-40	-225	-40	-225	mA
I _{CC}	Supply current	Total, outputs high	V _{CC} = MAX, Outputs open		48	70	48	70
		Total, outputs low			62	90	62	90
		Outputs at Hi-Z			64	95	64	95

† If conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH} Propagation delay time, low-to-high-level output	$C_L = 45 \text{ pF}$, $R_L = 667 \Omega$, See Note 2		8	12	ns
t_{PHL} Propagation delay time, high-to-low-level output			8	12	ns
t_{PZL} Output enable time to low level			27	40	ns
t_{PZH} Output enable time to high level			25	40	ns
t_{PLZ} Output disable time from low level	$C_L = 5 \text{ pF}$, $R_L = 667 \Omega$, See Note 2		15	25	ns
t_{PHZ} Output disable time from high level			15	28	ns

NOTE 2: Load circuits and voltage waveforms are shown in Section 1.



TTL Devices

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

2-703

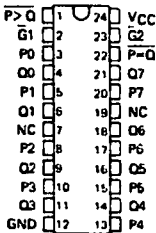
SN54LS682, SN54LS684, SN54LS685, SN54LS687, SN54LS688, SN74LS682, SN74LS684 THRU SN74LS688 8-BIT MAGNITUDE/IDENTITY COMPARATORS

D2617, JANUARY 1981—REVISED MARCH 1986

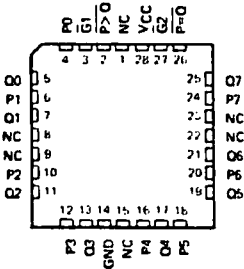
- Compares Two-8-Bit Words
- Choice of Totem-Pole or Open-Collector Outputs
- Hysteresis at P and Q Inputs
- 'LS682 has 20-kΩ Pullup Resistors on the Q Inputs
- SN74LS686 and 'LS687 . . . JT and NT 24-Pin, 300-Mil Packages

TYPE	P = Q	P > Q	OUTPUT ENABLE	OUTPUT CONFIGURATION	20-kΩ PULLUP
'LS682	yes	yes	no	totem-pole	yes
'LS684	yes	yes	no	totem-pole	no
'LS685	yes	yes	no	open-collector	no
SN74LS686	yes	yes	yes	totem-pole	no
'LS687	yes	yes	yes	open-collector	no
'LS688	yes	no	yes	totem-pole	no

SN54LS687 . . . JT PACKAGE
SN74LS686, SN74LS687 . . . DW OR NT PACKAGE
(TOP VIEW)

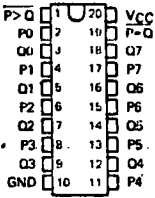


SN54LS687 . . . FK PACKAGE
(TOP VIEW)

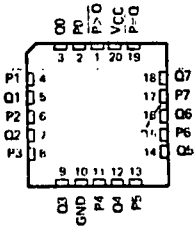


NC—No internal connection

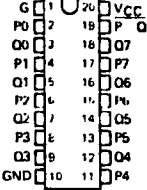
SN54LS682, SN54LS684, SN54LS685 . . . J PACKAGE
SN74LS682, SN74LS684, SN74LS685 . . . DW OR N PACKAGE
(TOP VIEW)



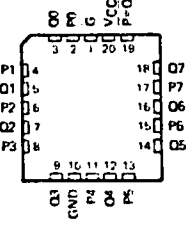
SN54LS682, SN54LS684, SN54LS685 . . . FK PACKAGE
(TOP VIEW)



SN54LS688 . . . J PACKAGE
SN74LS688 . . . DW OR N PACKAGE
(TOP VIEW)



SN54LS688 . . . FK PACKAGE
(TOP VIEW)



PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

TTL Devices

SN54LS682, SN54LS684, SN54LS685, SN54LS687, SN54LS688
SN74LS682, SN74LS684 THRU SN74LS688
8-BIT MAGNITUDE/IDENTITY COMPARATORS

description

These magnitude comparators perform comparisons of two eight-bit binary or BCD words. All types provide $P = Q$ outputs and all except 'LS688 provide $P > Q$ outputs as well. The 'LS682, 'LS684, 'LS686, and 'LS688 have totem-pole outputs, while the 'LS685 and 'LS687 have open-collector outputs. The 'LS682 features 20-k Ω pullup termination resistors on the Q inputs for analog or switch data.

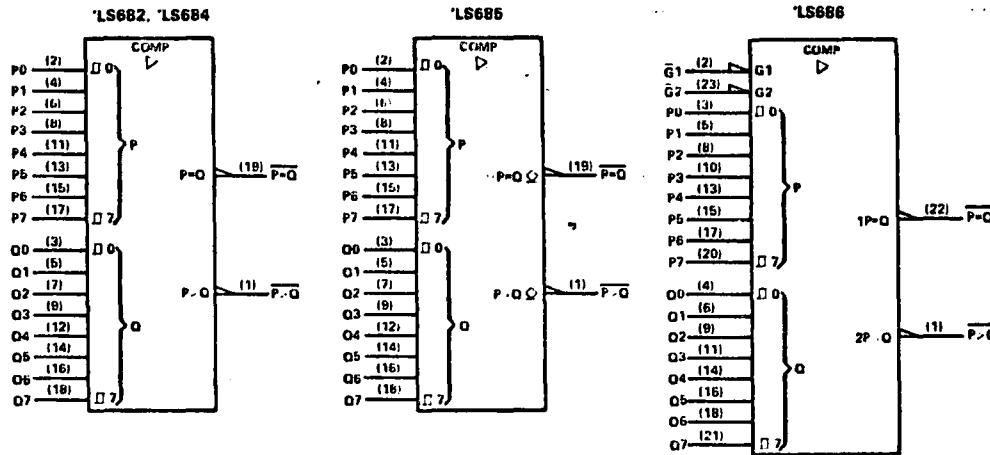
FUNCTION TABLE

INPUTS			OUTPUTS	
DATA	ENABLES		$P = Q$	$P > Q$
P, Q	$\bar{G}, G1$	$G2$		
$P = Q$	L	X	L	H
$P > Q$	X	L	H	L
$P < Q$	X	X	H	H
$P = Q$	H	X	H	H
$P > Q$	X	H	H	H
X	H	H	H	H

- NOTES: 1. The last three lines of the function table applies only to the devices having enable inputs, i.e., 'LS686 thru 'LS688.
2. The $P < Q$ function can be generated by applying the $P = Q$ and $P > Q$ outputs to a 2-input NAND gate.
3. For 'LS686 and 'LS687, $\bar{G}1$ enables $P = Q$ and $G2$ enables $P > Q$.

TTL Devices

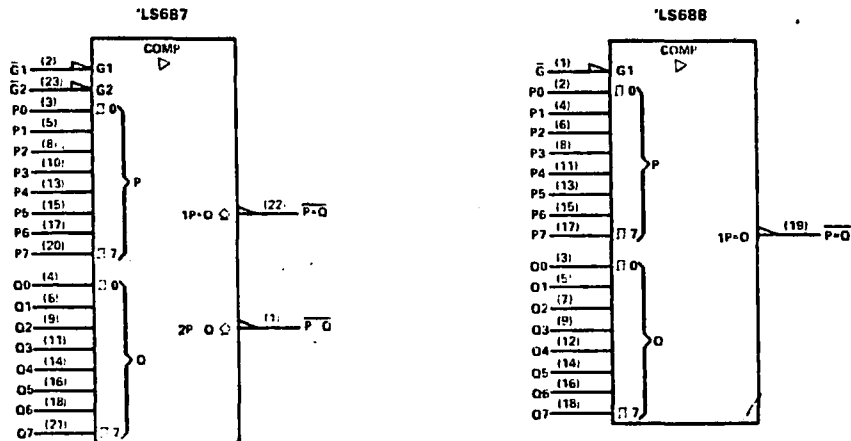
logic symbols†



†These symbols are in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for DW, J, JT, N, and NT packages.

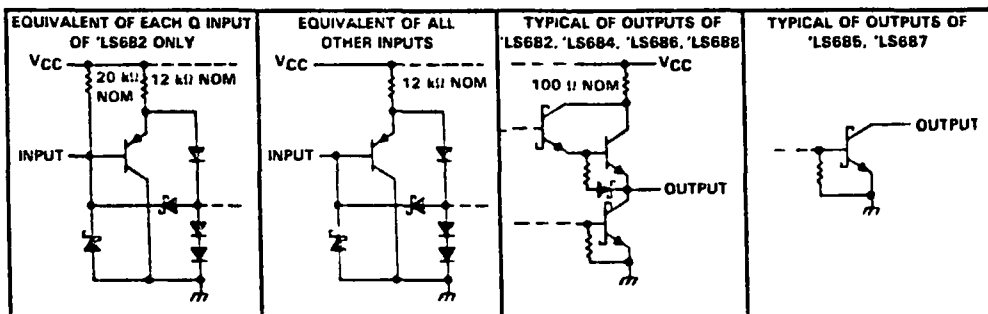
**SN54LS682, SN54LS684, SN54LS685, SN54LS687, SN54LS688,
SN74LS682, SN74LS684 THRU SN74LS688
8-BIT MAGNITUDE/IDENTITY COMPARATORS**

logic symbols† (continued)



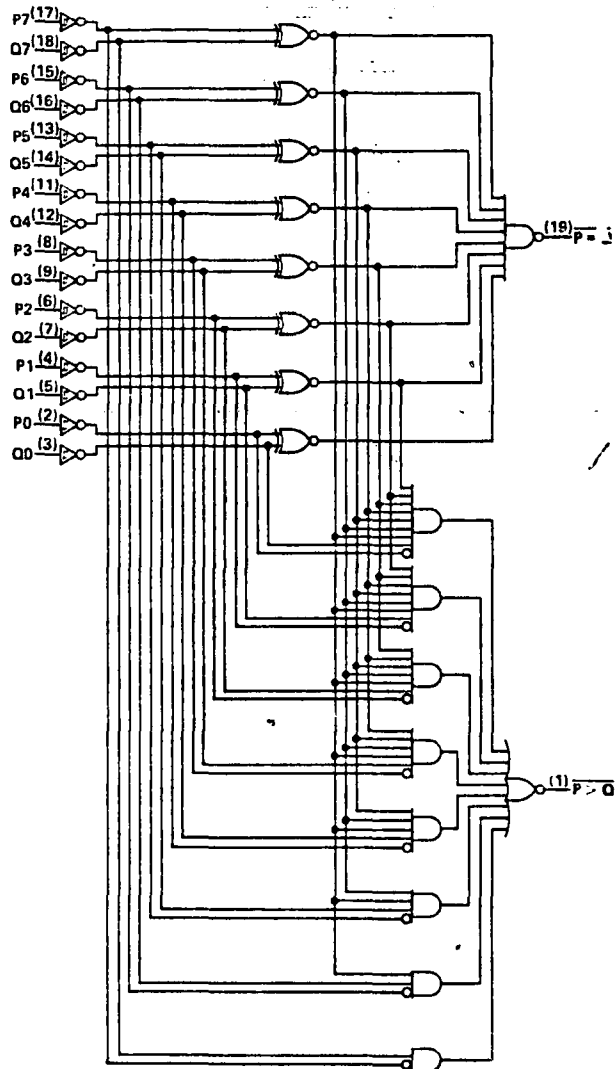
†These symbols are in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for DW, J, JT, N, and L package

schematics of inputs and outputs



**SN54LS682, SN54LS684, SN54LS685
SN74LS682, SN74LS684, SN74LS685
8-BIT MAGNITUDE/IDENTITY COMPARATORS**

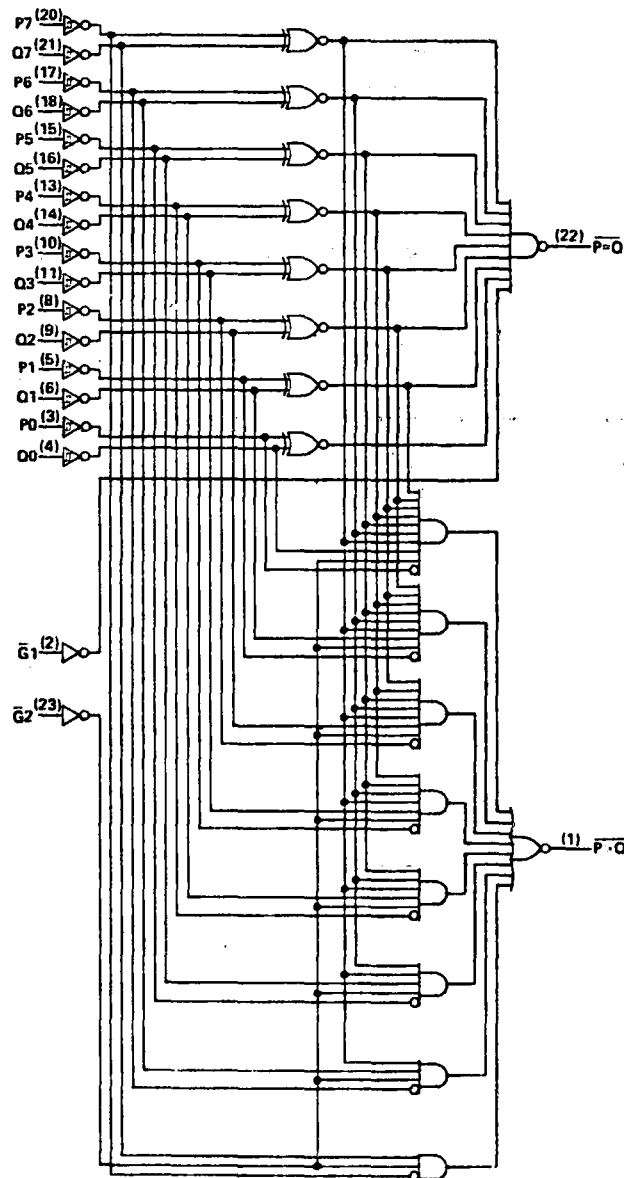
LS682, LS684, LS685 logic diagram (positive logic)



Pin numbers shown are for DW, J, and N packages.

SN54LS687
SN74LS686, SN74LS687
8-BIT MAGNITUDE/IDENTITY COMPARATORS

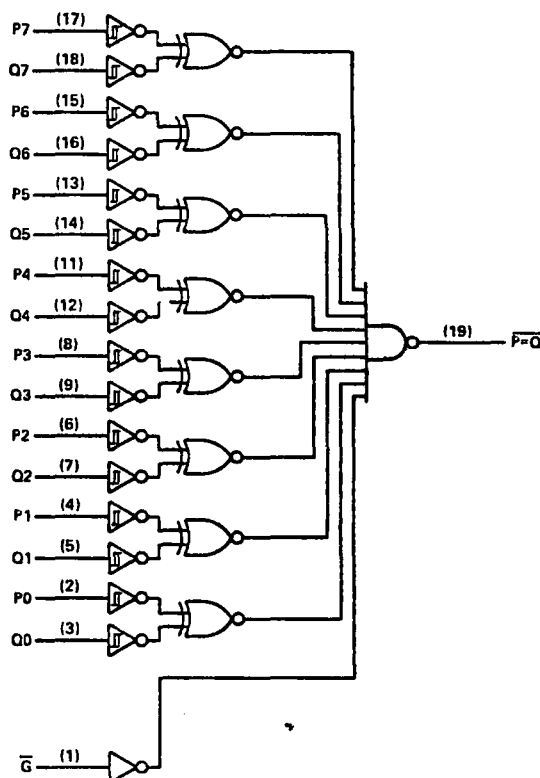
'LS686, 'LS687 logic diagram (positive logic)



Pin numbers shown are for DW, JT, and NT packages.

**SN54LS682, SN54LS684, SN54LS685, SN54LS687, SN54LS688
SN74LS682, SN74LS684 THRU SN74LS688
8-BIT IDENTITY COMPARATORS**

'LS688 logic diagram (positive logic)



Pin numbers shown are for DW, J, and N packages.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, VCC (see Note 1)	7 V
Input voltage: Q inputs of 'LS682	5.5 V
All other inputs	7 V
Off-state output voltage: 'LS685, 'LS687	7 V
Operating free-air temperature range:	
SN54LS682, SN54LS684, SN54LS685, SN54LS687, SN54LS688	-55°C to 125°C
SN74LS682, SN74LS684 thru SN74LS688	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

SN54LS682, SN54LS684, SN54LS688
SN74LS682, SN74LS684, SN74LS686, SN74LS688
8-BIT MAGNITUDE/IDENTITY COMPARATORS WITH TOTEM-POLE OUTPUTS

recommended operating conditions

	SN54LS ¹			SN74LS ¹			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.85	5	5.25	V
High-level output current, I_{OH}			-400			-400	μ A
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	-55		125	0		70	$^{\circ}$ C

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

PARAMETER		TEST CONDITIONS [†]	SN54LS ¹			SN74LS ¹			UNIT
			MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V_{IH}	High-level input voltage		2			2			V
V_{IL}	Low-level input voltage			0.7			0.8		V
$V_T + -V_T$	Hysteresis	P or Q inputs		0.4			0.4		V
V_{IK}	Input clamp voltage	$V_{CC} = \text{MIN}, I_I = -18 \text{ mA}$		-1.5			-1.5		V
V_{OH}	High-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = V_{IL \text{ max}}, I_{OH} = -400 \mu\text{A}$	2.5			2.7			V
V_{OL}	Low-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = V_{IL \text{ max}}$							V
		$I_{OL} = 12 \text{ mA}$	0.25	0.4		0.25	0.4		
		$I_{OL} = 24 \text{ mA}$				0.35	0.5		
I_I	Input current at maximum input voltage	Q inputs, 'LS682							
		All other inputs							
		$V_{CC} = \text{MAX}, V_I = 5.5 \text{ V}$		0.1			0.1		mA
		$V_{CC} = \text{MAX}, V_I = 7 \text{ V}$							
I_{IH}	High-level input current	$V_{CC} = \text{MAX}, V_I = 2.7 \text{ V}$		20			20		μ A
I_{IL}	Low-level input current	Q inputs, 'LS682		-0.4			-0.4		mA
		All other inputs		-0.2			-0.2		
$I_{OS}^{\S}$	Short-circuit output current	$V_{CC} = \text{MAX}, V_O = 0$	-20	-100		-20	-100		mA
I_{CC}	Supply current	'LS682	42	70		42	70		mA
		'LS684	40	65		40	65		
		'LS686	44	75		44	75		
		'LS688	40	65		40	65		

[†] For conditions shown as MIN or MAX, use the appropriate values specified under recommended operating conditions.

[‡] All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$.

[§] Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

NOTE 1: I_{CC} is measured with any $\bar{Q}$ inputs grounded, all other inputs at 4.5 V, and all outputs open.

TTL Devices

SN54LS682, SN54LS684, SN54LS688
SN74LS682, SN74LS684, SN74LS686, SN74LS688
8-BIT MAGNITUDE/IDENTITY COMPARATORS WITH TOTEM-POLE OUTPUTS

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER ¹	FROM (INPUTS)	TO (OUTPUT)	TEST CONDITIONS	'LS682			'LS684			'LS686			'LS688			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
t _{PLH}	P	$\overline{P} = \overline{Q}$	R _L = 667 Ω, C _L = 45 pF, All other inputs low, See Note 2	13	25		15	25		13	25		12	18	ns	
t _{PHL}				15	25		17	25		20	30		17	23		
t _{PLH}	Q	$\overline{P} = \overline{Q}$		14	25		16	25		13	25		12	18	ns	
t _{PHL}				15	25		15	25		21	30		17	23		
t _{PLH}	G, G ₁	$\overline{P} = \overline{Q}$								11	20		12	18	ns	
t _{PHL}										19	30		13	20		
t _{PLH}	P	$\overline{P} > \overline{Q}$			20	30		22	30		19	30				ns
t _{PHL}					15	30		17	30		15	30				
t _{PLH}	Q	$\overline{P} > \overline{Q}$			21	30		24	30		18	30				ns
t _{PHL}					19	30		20	30		19	30				
t _{PLH}	G ₂	$\overline{P} > \overline{Q}$									21	30				ns
t _{PHL}												16	25			

¹ t_{PLH} = propagation delay time, low-to-high-level outputs; t_{PHL} = propagation delay time, high-to-low-level output.
NOTE 2: Load circuits and voltage waveforms are shown in Section 1.

TTL Devices

**SN54LS685, SN54LS687
SN74LS685, SN74LS687, SN74LS688
8-BIT MAGNITUDE/IDENTITY COMPARATORS WITH TOTEM-POLE OUTPUTS**

recommended operating conditions

	SN54LS ¹			SN74LS ¹			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.85	5	5.25	V
High-level output current, V_{OH}			5.5			5.5	V
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	-55		125	0		70	°C

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

PARAMETER		TEST CONDITIONS ¹	SN54LS ¹			SN74LS ¹			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
V _{IH}	High-level input voltage		2			2			V
V _{IL}	Low-level input voltage				0.7			0.8	V
V _{T+} - V _{T-}	Hysteresis	P or Q inputs	0.4			0.4			V
V _{IK}	Input clamp voltage	V _{CC} = MIN, I _I = -18 mA	-1.5			-1.5			V
I _{OH}	High-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{ILmax} , V _{OH} = 5.5 V	250			100			μA
V _{OL}	Low-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{ILmax} , I _{OL} = 12 mA	0.25	0.4		0.25	0.4		V
		V _{IL} = V _{ILmax} , I _{OL} = 24 mA				0.35	0.5		
I _I		V _{CC} = MAX, V _I = 7 V	0.1			0.1			mA
I _{IH}	High-level input current	V _{CC} = MAX, V _I = 2.7 V	20			20			μA
I _{IL}	Low-level input current	V _{CC} = MAX, V _I = 0.4 V	-0.2			-0.2			mA
I _{CC}	Supply current	'LS685	40			40			mA
		'LS687	44			44			
		V _{CC} = MAX, See Note 1							

¹For conditions shown as MIN or MAX, use the appropriate values specified under recommended operating conditions.

²All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^\circ\text{C}$.

NOTE 1: I_{CC} is measure with any $\bar{G}$ inputs grounded, all other inputs at 4.5 V, and all outputs open.



TTL Devices

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

2-1137

SN54LS685, SN54LS687
SN74LS685, SN74LS687
8-BIT MAGNITUDE/IDENTITY COMPARATORS WITH OPEN-COLLECTOR OUTPUTS

switching characteristics, VCC = 5 V, TA = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'LS685			'LS687			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
tPLH	P	$\overline{P} = \overline{Q}$	R _L = 667 Ω, C _L = 45 pF, All other inputs low, See Note 2	30	45		24	35		ns
tPHL				19	35		20	30		
tPLH	Q	$\overline{P} = \overline{Q}$		24	45		24	35		ns
tPHL				23	35		20	30		
tPLH	$\overline{G}, \overline{G1}$	$\overline{P} = \overline{Q}$					21	35		ns
tPHL							18	30		
tPLH	P	$\overline{P} > \overline{Q}$		32	45		24	35		ns
tPHL				16	35		16	30		
tPLH	Q	$\overline{P} > \overline{Q}$		30	45		24	35		ns
tPHL				20	35		16	30		
tPLH	$\overline{G2}$	$\overline{P} > \overline{Q}$					24	35		ns
tPHL							15	30		

1tPLH = propagation delay time, low-to-high-level outputs; tPHL = propagation delay time, high-to-low-level output.
 NOTE 2: Load circuits and voltage waveforms are shown in Section 1.

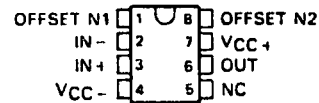
TTL Devices

OP-07C, OP-07D, OP-07E ULTRA-LOW-OFFSET-VOLTAGE OPERATIONAL AMPLIFIERS

D2757, OCTOBER 1983—REVISED JUNE 1988

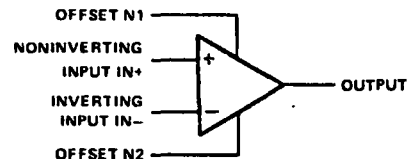
- Ultra-Low Offset Voltage . . .30 μ V Typ (OP-07E)
- Ultra-Low Offset Voltage Temperature Coefficient . . . 0.3 μ V/ $^{\circ}$ C Typ (OP-07E)
- Ultra-Low Noise
- No External Components Required
- Replaces Chopper Amplifiers at a Lower Cost
- Single-Chip Monolithic Fabrication
- Wide Input Voltage Range
0 to $\pm$ 14 V Typ
- Wide Supply Voltage Range
 $\pm$ 3 V to $\pm$ 18 V
- Essentially Equivalent to Fairchild μ A714 Operational Amplifiers
- Direct Replacement for PMI OP-07C, OP-07D, OP-07E

D, JG, OR P PACKAGE
(TOP VIEW)



NC—No internal connection

symbol



description

These devices represent a breakthrough in operational amplifier performance. Low offset and long-term stability are achieved by means of a low-noise, chopperless, bipolar-input-transistor amplifier circuit. For most applications, external components are required for offset nulling and frequency compensation. The true differential input, with a wide input voltage range and outstanding common-mode rejection, provides maximum flexibility and performance in high-noise environments and in noninverting applications. Low bias currents and extremely high input impedances are maintained over the entire temperature range. The OP-07 is unsurpassed for low-noise, high-accuracy amplification of very-low-level signals.

These devices are characterized for operation from 0 $^{\circ}$ C to 70 $^{\circ}$ C.

AVAILABLE OPTIONS

T _A	V _{IO} MAX at 25 $^{\circ}$ C	PACKAGE		
		SMALL OUTLINE (D)	CERAMIC DIP (JG)	PLASTIC DIP (P)
0 $^{\circ}$ C to 70 $^{\circ}$ C	150 μ V	OP-07CD OP-07DD	OP-07CJG OP-07DJG	OP-07CP OP-07DP
	75 μ V	OP-07ED	OP-07EJG	OP-07EP

The D package is available taped and reeled. Add the suffix R to the device type when ordering. (e.g., OP-07CDR)

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

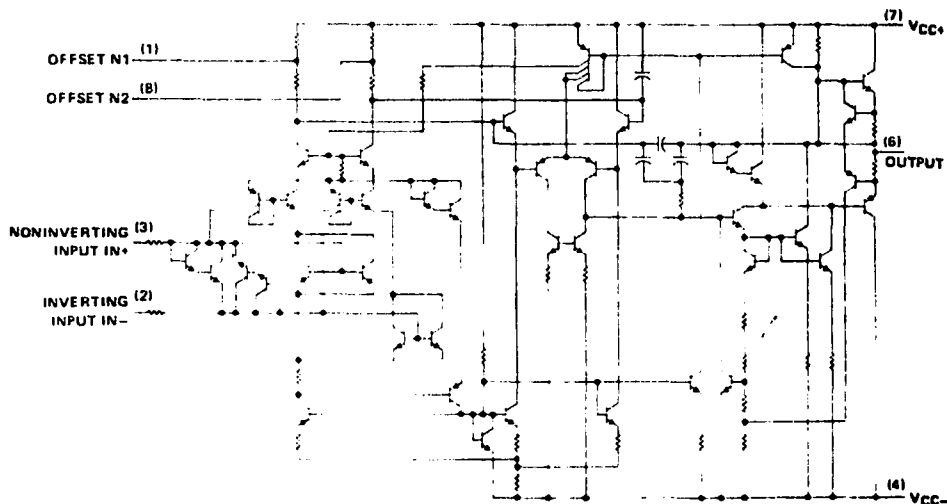
TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

Copyright $\copyright$ 1983, Texas Instruments Incorporated

OP-07C, OP-07D, OP-07E
ULTRA-LOW-OFFSET VOLTAGE OPERATIONAL AMPLIFIERS

schematic



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage V_{CC+} (see Note 1)	22 V
Supply voltage V_{CC-}	-22 V
Differential input voltage (see Note 2)	± 30 V
Input voltage (either input, see Note 3)	± 22 V
Duration of output short circuit (see Note 4)	unlimited
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 5)	500 mW
Operating free-air temperature range	0°C to 70°C
Storage temperature range	-65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds: JG package	300°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: D or P package	260°C

- NOTES 1. All voltage values, unless otherwise noted, are with respect to the midpoint between V_{CC+} and V_{CC-} .
2. Differential voltages are at the noninverting input terminal with respect to the inverting input terminal.
3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 volts, whichever is less.
4. The output may be shorted to ground or either power supply.
5. For operation above 64°C free air temperature, derate the D package to 464 mW at 70°C at the rate of 5.8 mW/°C.

OP-07C, OP-07D, OP-07E
ULTRA-LOW-OFFSET-VOLTAGE OPERATIONAL AMPLIFIERS

CC
PUT
22 V
22 V
30 V
22 V
limited
0 mW
70°C
150°C
300°C
260°C
°C

electrical characteristics at specified free-air temperature. VCC = ± 15 V (unless otherwise noted)

PARAMETER	TEST CONDITIONS	OP 7C			OP 7D			OP 7E			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V _{IO} Input offset voltage	V _O = 0, R _S = 50 Ω	60	150	150	60	150	150	30	75	75	μV
Temperature coefficient of V _{IO}	0°C to 70°C	85	250	250	85	250	250	45	130	130	μV/°C
Long term drift of input offset voltage	V _O = 0, R _S = 50 Ω	0.5	1.8	2.5	0.7	2.5	2.5	0.3	1.3	1.3	μV/mo
Offset adjustment range	See Note 6	0.4			0.5			0.3			mV
I _{IO} Input offset current	R _S = 20 kΩ, See Figure 1	2.4			±4			±4			nA
Temperature coefficient of I _{IO}	25°C	0.8	6	6	0.8	6	6	0.5	3.8	3.8	nA/°C
Input bias current	0°C to 70°C	1.6	6	6	1.6	6	6	0.9	5.3	5.3	nA
Temperature coefficient of input bias current	0°C to 70°C	12	50	50	12	50	50	8	35	35	pA/°C
I _{IB} Input bias current	25°C	1.8	7	7	1.7	12	12	1.2	14	14	nA
Temperature coefficient of I _{IB}	0°C to 70°C	2.2	9	9	2.3	14	14	1.5	15.5	15.5	nA/°C
I _{IB} Input bias current	0°C to 70°C	18	50	50	18	50	50	13	35	35	pA/°C
Common-mode input voltage range	25°C	±13	±14		±13	±14		±13	±14		V
V _{ICR} Common-mode input voltage range	0°C to 70°C	±13	±13.5		±13	±13.5		±13	±13.5		V
V _{OM} Peak output voltage	R _L ≥ 10 kΩ	±12	±13		±12	±13		±12.5	±13		V
	R _L ≥ 2 kΩ	±11.5	±12.8		±11.5	±12.8		±10.5	±12		V
	R _L ≥ 1 kΩ	±12			±12			±10.5	±12		V
	R _L ≥ 2 kΩ	±11	±12.6		±11	±12.6		±12	±12.6		V
Large-signal differential voltage amplification	VCC = ±3 V, V _O = 0.5 V, R _L ≥ 500 kΩ	100	400		400			150	400		V/mV
B ₁ Unity gain bandwidth	V _O = ±10 V, R _L = 2 kΩ	120	400		120	400		200	500		MHz
f ₁ Input resistance	25°C	0.4	0.6		0.4	0.6		0.4	0.6		MΩ
CMRR Common-mode rejection ratio	V _{IC} = ±13 V, R _S = 50 Ω	97	120		94	110		106	123		dB
SVS Supply voltage sensitivity	VCC = ±3 V to ±18 V, R _S = 50 Ω	7	32		7	32		5	20		μV/V
P _D Power dissipation	V _O = 0, No load	80	150		80	150		75	120		mW
	VCC = ±3 V, V _O = 0, No load	4	8		4	8		4	8		mW

NOTE 6: All characteristics are measured under open-loop conditions with zero common-mode input voltage unless otherwise noted. Since long-term drift cannot be measured on the individual devices, this specification is not intended to be a guarantee or warranty. It is an engineering estimate of the averaged trend line of drift versus time over a minimum of 1000 hours at the test temperature.

OP-07C, OP-07D, OP-07E ULTRA-LOW-OFFSET VOLTAGE OPERATIONAL AMPLIFIERS

operating characteristics at specified free-air temperature, $V_{CC\pm} = \pm 15\text{ V}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		OP-7C			OP-7D			OP-7E			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V _n	Equivalent input noise voltage	T _A = 25°C	f = 10 Hz	10.5			10.5			10.3			nV/√Hz
			f = 100 Hz	10.2			10.3			10.0			
			f = 1 kHz	9.8			9.8			9.6			
V _{NPP}	Peak-to-peak equivalent input noise voltage	f = 0.1 Hz to 10 Hz, T _A = 25°C		0.38			0.38			0.35			μV
I _n	Equivalent input noise current	T _A = 25°C	f = 10 Hz	0.35			0.35			0.32			pA/√Hz
			f = 100 Hz	0.15			0.15			0.14			
			f = 1 kHz	0.13			0.13			0.12			
I _{NPP}	Peak-to-peak equivalent input noise current	f = 0.1 Hz to 10 Hz, T _A = 25°C		15			15			14			pA
SR	Slew rate	R _L ≥ 2 kΩ, T _A = 25°C		0.3			0.3			0.3			V/μs

†All characteristics are measured under open loop conditions with zero common mode input voltage unless otherwise specified.

TYPICAL APPLICATION DATA

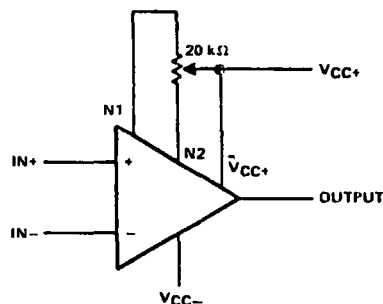


FIGURE 1. INPUT OFFSET VOLTAGE NULL CIRCUIT

ELECTRICAL CHARACTERISTICS

STATIC CHARACTERISTICS^{1,3}

PARAMETER		CONDITIONS	V _{SS} (Vdc)	V _{DD} (Vdc)	T _{LOW} ²		25°C			T _{HIGH} ²		Units	
					Min.	Max.	Min.	Typ.	Max.	Min.	Max.		
QUIESCENT DEVICE CURRENT	I _{DD}	V _{IN} = V _{SS} or V _{DD} All valid input combinations	0	5	—	0.05	—	0.0005	0.05	—	1.5	μAdc	
			0	10	—	0.1	—	0.001	0.1	—	3.0		
			0	15	—	0.2	—	0.002	0.2	—	6.0		
MINIMUM INPUT HIGH VOLTAGE (Control Input)	V _{IH}	V _{IS} = V _{SS} V _{OS} = V _{DD} I _{OS} = 10μA	0	5	—	4.0	—	2.75	4.0	—	4.0	Vdc	
			0	10	—	8.0	—	5.5	8.0	—	8.0		
			0	15	—	12.0	—	8.25	12.0	—	12.0		
MAXIMUM INPUT LOW VOLTAGE (Control Input)	V _{IL}	V _{IS} = V _{SS} V _{OS} = V _{DD} I _{OS} = 10μA	0	5	1.0	—	1.0	2.25	—	1.0	—	Vdc	
			0	10	2.0	—	2.0	4.5	—	2.0	—		
			0	15	3.0	—	3.0	6.75	—	3.0	—		
SWITCH INPUT/OUTPUT LEAKAGE	I _{OFF}	V _C = V _{SS} V _{IS} = ±7.5Vdc	-7.5	+7.5	—	±100	—	±0.01	±100	—	±200	nAdc	
ON-RESISTANCE C,D,F,H device	R _{ON}	V _C = V _{DD} V _{SS} < V _{IS} < V _{DD} R _L = 10kΩ	-7.5	+7.5	—	220	—	80	280	—	320	Ω	
			0	+15	—								
			-5	+5	—	310	—	120	400	—	550	Ω	
			0	+10	—								
			-2.5	+2.5	—	2000	—	270	2500	—	3500	Ω	
			0	+5	—								
	E device	R _{ON}	V _C = V _{DD} V _{SS} < V _{IS} < V _{DD} R _L = 10kΩ	-7.5	+7.5	—	250	—	80	280	—	300	Ω
				0	+15	—							
				-5	+5	—	330	—	120	400	—	520	Ω
				0	+10	—							
				-2.5	+2.5	—	2100	—	270	2500	—	3200	Ω
				0	+5	—							
ON-RESISTANCE MATCH (Same package)	ΔR _{ON}	V _C = V _{DD} V _{SS} < V _{IS} < V _{DD} R _L = 10kΩ	-7.5	+7.5	—	—	—	5	—	—	—	Ω	
			0	+15	—								
			-5	+5	—	—	—	10	—	—	—	Ω	

NOTES: ¹ Remaining Static Electrical Characteristics are listed under "SCL4000B Series Family Specifications"² T_{LOW} = -55°C for C, D, F, H device.

= -40°C for E device.

T_{HIGH} = +125°C for C, D, F, H device.

= + 85°C for E device.

³ This device has been designed for balanced output drive current specifications. Consult Family Specifications.

SCL4066B



CMOS QUAD ANALOG SWITCH

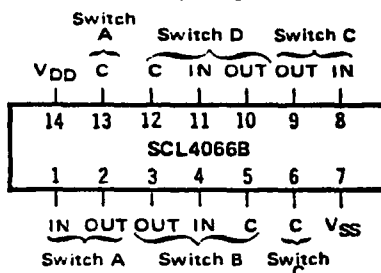
FEATURES

- Transmission or Multiplexing of Analog or Digital Signals
- 80Ω Typical ON-Resistance for 15-Volt operation
- Switch ON-Resistance Matched to within 5Ω over 15-Volt Signal-Input Range
- ON-Resistance Flat over Full Peak-to-Peak Signal Range
- High Degree of Linearity:
 - ≤ 0.5% Distortion (typ) @ $f_{is} = 1\text{kHz}$, $V_{is} = 5\text{V}_{p-p}$, $V_{DD} - V_{SS} \geq 10\text{V}$, $R_L = 10\text{k}\Omega$
- Extremely Low OFF switch Leakage Resulting in very Low Offset Current and High Effective OFF Resistance:
 - 10pA (typ) @ $V_{DD} - V_{SS} = 10\text{V}$, $T_A = 25^\circ\text{C}$
- Extremely High Control Input Impedance (Control Circuit Isolated from Signal Circuit):
 - $10^{12}\Omega$ (typ)
- Low Crosstalk between Switches:
 - 50dB (typ) @ $f_{is} = 0.9\text{MHz}$, $R_L = 1\text{k}\Omega$
- Matched Control-Input to Signal-Output Capacitance Reduces Output Signal Transients
- Frequency Response, Switch ON = 40MHz (typ)

DESCRIPTION

The SCL4066B is a Quad Bilateral Switch intended for the transmission or multiplexing of analog or digital signals. It is pin-for-pin compatible with the SCL4016B, but exhibits a much lower ON-resistance. In addition, the ON-resistance is relatively constant over the full input signal range. The SCL4066 consists of four independent bilateral switches. A single control signal is required per switch. Both the P and the N device in a given switch are biased ON or OFF simultaneously by the control signal. As shown below, the well of the N-channel device on each switch is either tied to the input when the switch is ON or to V_{SS} when the switch is OFF. This configuration minimizes the variation of the switch-transistor threshold

CONNECTION DIAGRAM (all packages)



Add suffix to package:

C	14-pin Cerdip	F	14-pin Flat
D	14-pin Ceramic	H	Chip
E	14-pin Epoxy		

RECOMMENDED OPERATING CONDITIONS

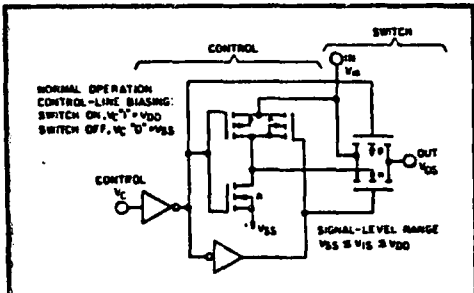
For maximum reliability:

DC Supply Voltage	$V_{DD} - V_{SS}$	3 to 15	Vdc
Operating Temperature	T_A	-55 to +125	°C
C, D, F, H Device		-40 to +85	°C
E Device			

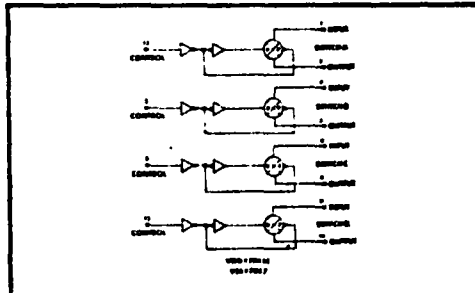
voltage with input-signal, and thus keeps the ON-resistance low over the full operating range.

The advantages over single-channel switches include peak input-signal voltage swings equal to the full supply voltage, and more constant ON-impedance over the input-signal range. For sample-and-hold applications, however, the SCL4016 is recommended.

SCHEMATIC DIAGRAM (one of four switches)



LOGIC DIAGRAM



ELECTRICAL CHARACTERISTICS

STATIC CHARACTERISTICS ^{1,3}

PARAMETER		CONDITIONS	V _{SS} (Vdc)	V _{DD} (Vdc)	T _{LOW} ²		25°C			T _{HIGH} ²		Units	
					Min.	Max.	Min.	Typ.	Max.	Min.	Max.		
QUIESCENT DEVICE CURRENT	I _{DD}	V _{IN} = V _{SS} or V _{DD}	0	5	—	0.05	—	0.0005	0.05	—	1.5	μA _{dc}	
		All valid input combinations	0	10	—	0.1	—	0.001	0.1	—	3.0		
			0	15	—	0.2	—	0.002	0.2	—	6.0		
MINIMUM INPUT HIGH VOLTAGE (Control Input)	V _{IH}	V _{IS} = V _{SS}	0	5	—	4.0	—	2.75	4.0	—	4.0	Vdc	
		V _{OS} = V _{DD}	0	10	—	8.0	—	5.5	8.0	—	8.0		
		I _{OS} = 10μA	0	15	—	12.0	—	8.25	12.0	—	12.0		
MAXIMUM INPUT LOW VOLTAGE (Control Input)	V _{IL}	V _{IS} = V _{SS}	0	5	1.0	—	1.0	2.25	—	1.0	—	Vdc	
		V _{OS} = V _{DD}	0	10	2.0	—	2.0	4.5	—	2.0	—		
		I _{OS} = 10μA	0	15	3.0	—	3.0	6.75	—	3.0	—		
SWITCH INPUT/OUTPUT LEAKAGE	I _{OFF}	V _C = V _{SS} V _{IS} = ±7.5Vdc	-7.5	+7.5	—	±100	—	±0.01	±100	—	±200	nA _{dc}	
ON-RESISTANCE C,D,F,H device	R _{ON}	V _C = V _{DD} V _{SS} < V _{IS} < V _{DD} R _L = 10kΩ	-7.5	+7.5	—	220	—	80	280	—	320	Ω	
			0	+15									
			-5	+5	—	310	—	120	400	—	550	Ω	
	0	+10											
	-2.5	+2.5	—	2000	—	270	2500	—	3500	Ω			
	0	+5											
E device	R _{ON}	V _C = V _{DD} V _{SS} < V _{IS} < V _{DD} R _L = 10kΩ	-7.5	+7.5	—	250	—	80	280	—	300	Ω	
			0	+15									
			-5	+5	—	330	—	120	400	—	520	Ω	
0	+10												
-2.5	+2.5	—	2100	—	270	2500	—	3200	Ω				
0	+5												
ON-RESISTANCE MATCH (Same package)	ΔR _{ON}	V _C = V _{DD} V _{SS} < V _{IS} < V _{DD} R _L = 10kΩ	-7.5	+7.5	—	—	—	5	—	—	—	Ω	
			0	+15									
			-5	+5	—	—	—	10	—	—	—	Ω	
0	+10												
-2.5	+2.5	—	—	—	10	—	—	—	Ω				
0	+5												

NOTES: ¹ Remaining Static Electrical Characteristics are listed under "SCL4000B Series Family Specifications"² T_{LOW} = -55°C for C, D, F, H device.

= -40°C for E device.

T_{HIGH} = +125°C for C, D, F, H device.

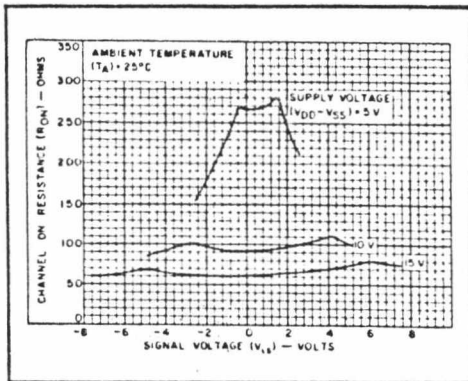
= + 85°C for E device.

³ This device has been designed for balanced output drive current specifications. Consult Family Specifications.

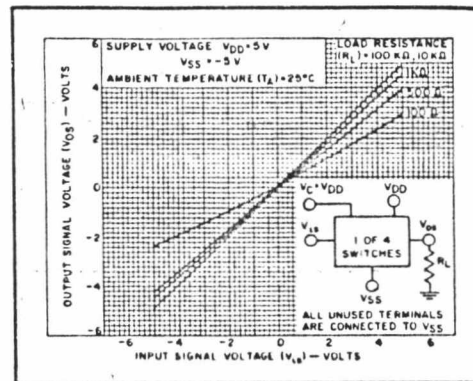
ELECTRICAL CHARACTERISTICS (Continued)

DYNAMIC CHARACTERISTICS ($C_L = 50\text{pF}$, $T_A = 25^\circ\text{C}$)

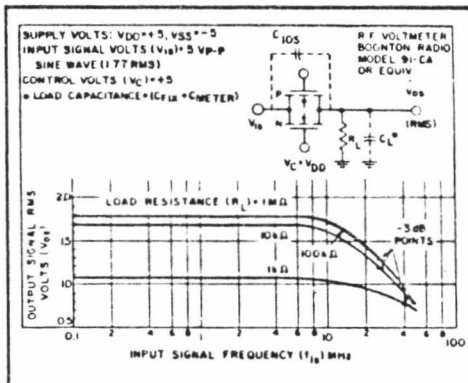
PARAMETER	CONDITIONS	V _{SS} (Vdc)	V _{DD} (Vdc)	Min.	Typ.	Max.	Units
SIGNAL INPUTS (V _{is}) AND OUTPUTS (V _{os})							
PROPAGATION DELAY TIME Signal Input to Signal Output	t _{PLH} , t _{PHL} V _c =V _{DD} V _b =Square Wave R _L = 10kΩ	0 0 0	5 10 15	— — —	20 10 7.5	40 20 15	ns
BANDWIDTH (-3dB) (Sine Wave)	BW V _c =V _{DD} V _b =5V _{p-p} centered @ 0.0Vdc R _L 1kΩ 10kΩ 100kΩ 1MΩ	-5	+5	— — — —	54 40 38 37	— — — —	MHz
INSERTION LOSS (= 20 · log ₁₀ $\frac{V_{os}}{V_b}$)	V _c =V _{DD} V _b =5V _{p-p} centered @ 0.0Vdc R _L 1kΩ 10kΩ 100kΩ 1MΩ	-5	+5	— — — —	2.3 0.2 0.1 0.05	— — — —	dB
SIGNAL DISTORTION (Sine Wave)	V _c =V _{DD} V _b =5V _{p-p} centered @ 0.0Vdc f _{is} =1.0kHz R _L = 10kΩ	-5	+5	—	0.16	—	%
FEEDTHROUGH (-50dB)	V _c =V _{SS} V _b =5V _{p-p} centered @ 0.0Vdc R _L 1kΩ 10kΩ 100kΩ 1MΩ	-5	+5	— — — —	1250 140 18 2	— — — —	kHz
CROSSTALK (-50dB) Between two switches	V _c (A)=V _{DD} V _c (B)=V _{SS} V _b (A)=5V _{p-p} centered @ 0.0Vdc R _L = 10kΩ	-5	+5	—	0.9	—	MHz
CAPACITANCE Input	C _{in}			—	8	—	pF
Output	C _{os}	V _C = V _{SS}	-5	+5	—	8	pF
Feedthrough	C _{ios}			—	0.5	—	pF
CONTROL INPUT (V _c)							
PROPAGATION DELAY TIME Turn on	t _{PC} V _{SS} <V _b <V _{DD} R _L = 10kΩ	0 0 0	5 10 15	— — —	50 25 20	100 50 40	ns
MAXIMUM INPUT FREQUENCY	f _c V _{SS} <V _b <V _{DD} R _L = 1.0kΩ	0 0 0	5 10 15	— — —	5 10 12	— — —	MHz
CROSSTALK (To signal port)	V _c = Square Wave R _L = 10kΩ R _{in} = 1.0kΩ	0 0 0	5 10 15	— — —	30 50 100	— — —	mV



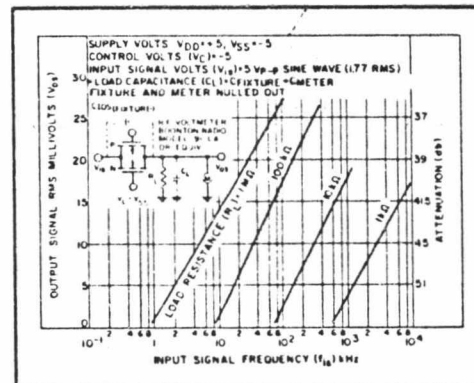
Typical channel ON resistance vs. signal voltage for three values of supply voltage ($V_{DD} - V_{SS}$)



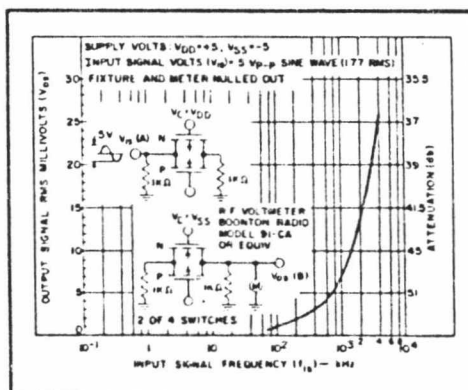
Typical ON characteristics for 1 of 4 channels.



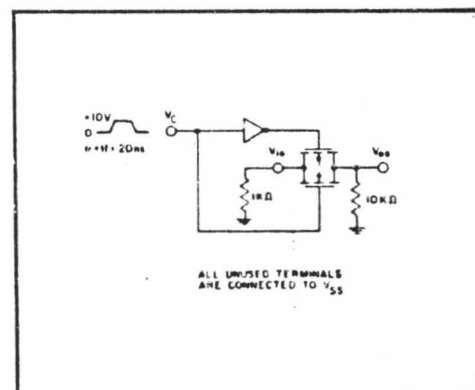
Typ. switch frequency response - switch "ON"



Typ. feedthru vs. freq. - switch "OFF"



Typ. crosstalk between switch circuits in the same package

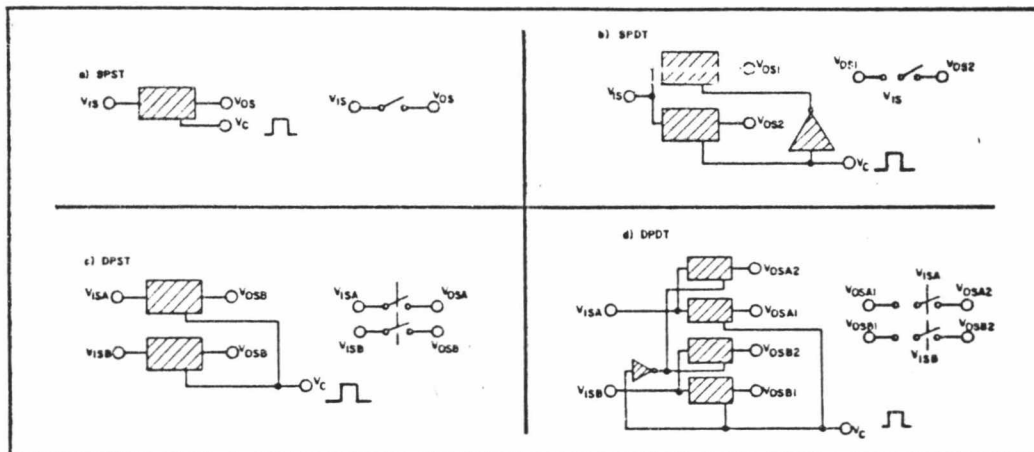


Test circuit, crosstalk-control input to signal output

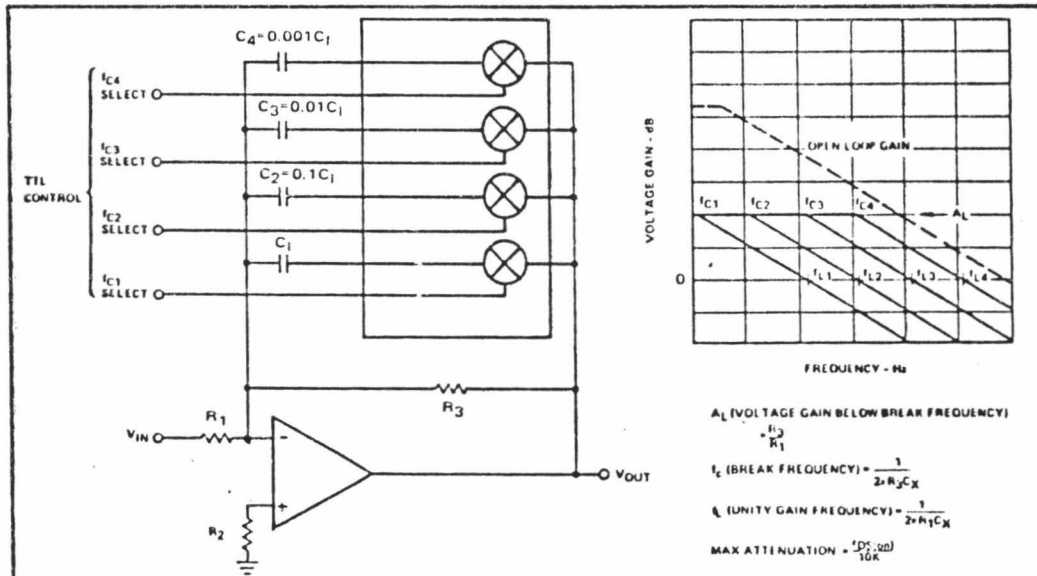
SPECIAL CONSIDERATIONS – SCL4066B

1. In applications where separate power sources are used to drive V_{DD} and the signal inputs, the V_{DD} current capability should exceed V_{DD}/R_L (R_L = effective external load of the 4 SCL4066B bilateral switches). This provision avoids any permanent current flow or clamp action on the V_{DD} supply when power is applied or removed from SCL4066B.
2. In certain applications, the external load-resistor current may include both V_{DD} and signal-line components. To avoid drawing V_{DD} current when switch current flows into terminals 1, 4, 8, or 11, the voltage drop across the bidirectional switch must not exceed 0.8 volt (calculated from R_{ON} values shown). No V_{DD} current will flow through R_L if the switch current flows into terminals 2, 3, 9, or 10. Failure to observe this condition may result in distortion of the signal.

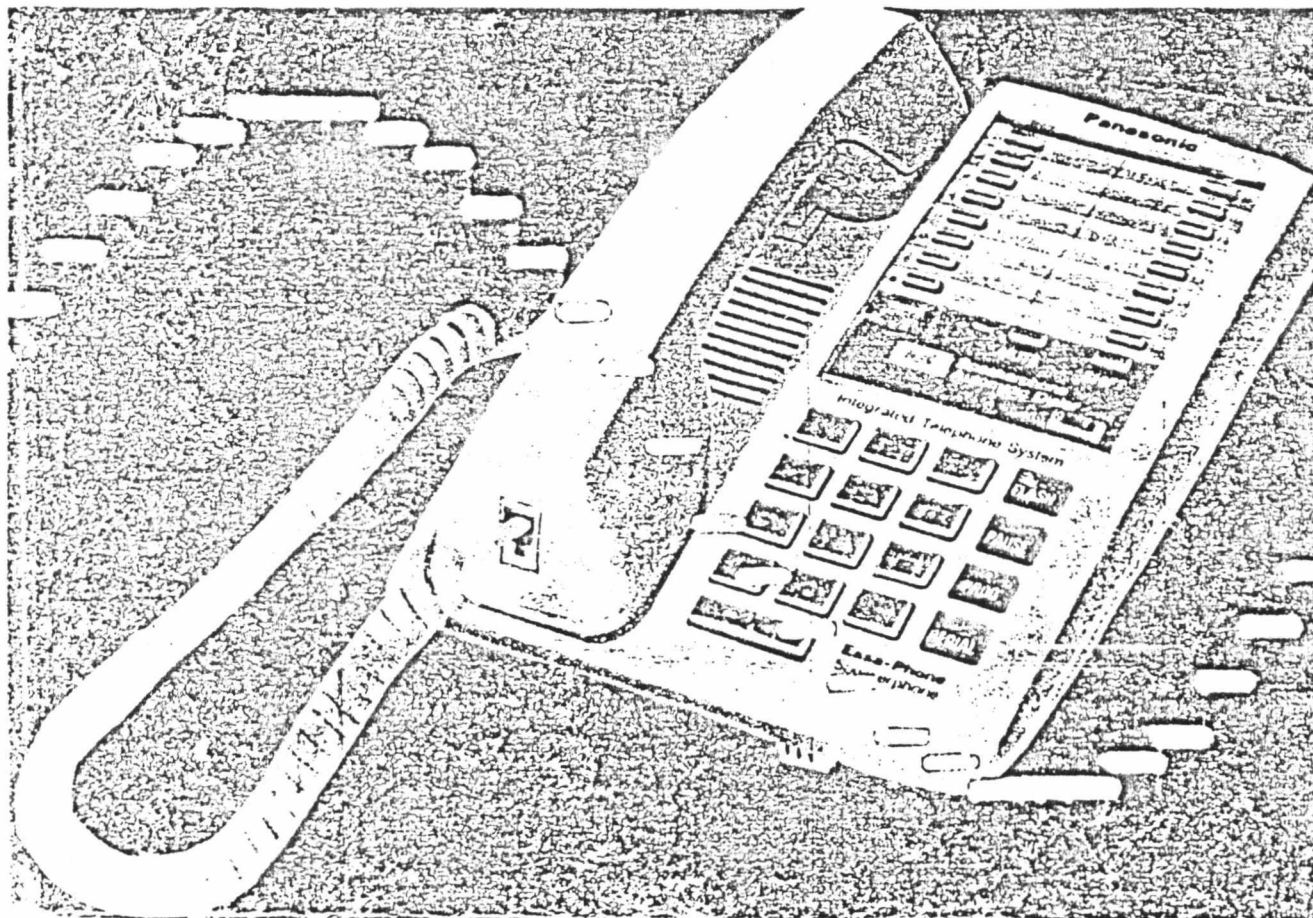
APPLICATIONS INFORMATION



Basic Switch Functions using the SCL4066B



Active Low Pass Filter with Digitally Selected Break Frequency



DTMF Encoding and Decoding

Thanks to new low-cost DTMF encoders and decoders, the world of DTMF signalling now is available for use in your next project.

DALE NASSAR

(MULTIFREQUENCY MULTIFREQUENCY) signalling was developed about two decades ago by Bell Labs as a faster (by a factor of about 10), more versatile, and more reliable telephone-dialing scheme than the old pulse or rotary dialing technique. The DTMF method is often referred to as tone dialing or *Touch-Tone* (note that *Touch-Tone* is a trademark of AT&T) and is used with push-button telephones and other equipment.

A standard DTMF signal consists of a pair of audio tones chosen from a group of eight standard frequencies. Those frequencies are divided into two groups: a low-tone group of four frequencies and a high-tone group of four frequencies. A standard DTMF signal consists of the algebraic sum of one tone from the low group and one tone from the high group. There are therefore $16 (4 \text{ low} \times 4 \text{ high})$

possible DTMF signals that can be encoded with the eight frequencies. The four standard low frequencies are 697, 770, 852, and 941 Hz, and are referred to as row frequencies R1, R2, R3, and R4, respectively. The four standard high frequencies are 1209, 1336, 1477, and 1633 Hz, and are referred to as column frequencies C1, C2, C3, and C4, respectively. Any combination of DTMF tone can be generated using a 4×4 keypad switch matrix as shown in Fig. 1. The DTMF frequencies and the keypad layout of Fig. 1 are international standards. The frequencies produced by DTMF generators are allowed a $\pm 1.5\%$ deviation from the listed standards. Note that all of those tones are well within the telephone system's voice band.

The choice of the standard DTMF frequencies was by no means an arbitrary

one. The designers of the DTMF system used a great deal of care in selecting the particular frequencies. Other tones that may appear on the telephone line such as dial tones and power-line noise must not fall in the DTMF frequency band. Further, the standard frequencies must have no harmonic interaction, thus the highest standard frequency (1633 Hz) is lower than the third harmonic of the lowest standard frequency (697 Hz).

Conventional telephones that use DTMF signalling are usually equipped with a standard 3×4 keypad matrix for representing the digits 0-9, and two spare symbols, * (star or asterisk), and # (pound or octothorpe), which can be used for various purposes. That 3×4 matrix represents all four row frequencies (R1-R4), and the three lowest column frequencies (C1-C3). Some special-purpose

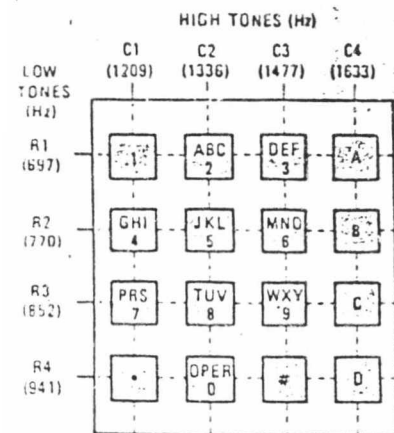


FIG. 1—STANDARD DTMF KEYPAD layout. The DTMF row and column frequencies are as shown.

telephones use the fourth column (C4) to represent four additional symbols (shown as A, B, C, and D in Fig. 1) in order to encode all of the sixteen possible DTMF signals.

If you have a tone-dial phone you can listen to a DTMF signal by simply picking up the telephone handset and pressing one of the buttons. For example, pressing the 8 key generates a 852-Hz tone (R3) and a 1336-Hz tone (C2) simultaneously. Those signals are processed and decoded by a DTMF receiver at the telephone company's central office.

The central office contains the switching equipment that provides local-exchange telephone service for a given geographical area. That area is designated by the first three digits of the telephone number. After the connection is established between the called and the calling parties, the DTMF receiver at the central office is no longer active and the connected parties are free to use the keypad-generated signals for station-to-station (end-to-end) signalling.

Until very recently, the DTMF encoders used by the telephone companies exclusively used large and bulky transistorized LC-tuned oscillator circuits to generate the tones. Many such LC circuits are still in use. Such rugged circuits were used by the telephone company because they were extremely dependable. They were designed to withstand the worst of operating conditions. For the hobbyist, limited parts availability makes building that type of circuit almost impractical. Fortunately it is also unnecessary, as DTMF generators are available in IC form.

Further, until just a few years ago the experimenter had to settle for a not-so-reliable IC decoding (receiving) system. The decoding circuitry had to be built up using a number of simple IC's. For instance, a separate 567 phase-locked-loop tone-detector IC was required for each

frequency used, for a total of eight. Additionally, each tone detector had to be tuned by critical external timing components. Because each DTMF signal received activated two detector outputs (one for each frequency received), a logic circuit had to be added to convert those outputs into a usable format. The net result was a complex circuit that was time-consuming to build and difficult to align. Also, performance was often unsatisfactory. True, performance could be improved with the addition of pre-filtering at the inputs of each tone detector. But the active-filter circuitry required for that

made an already complex circuit even more so.

Fortunately, those days are gone forever. With the new DTMF IC's available today, a complete and extremely reliable DTMF-encoding and -decoding system can be breadboarded in less than 10 minutes. Also, the built-in features of those decoding IC's usually include pre-filtering, complex processing, signal validation, etc., making possible a high degree of efficiency and reliability. In addition, no external tuning components are required, keeping the parts count minimal. DTMF IC's are manufactured by Nation-

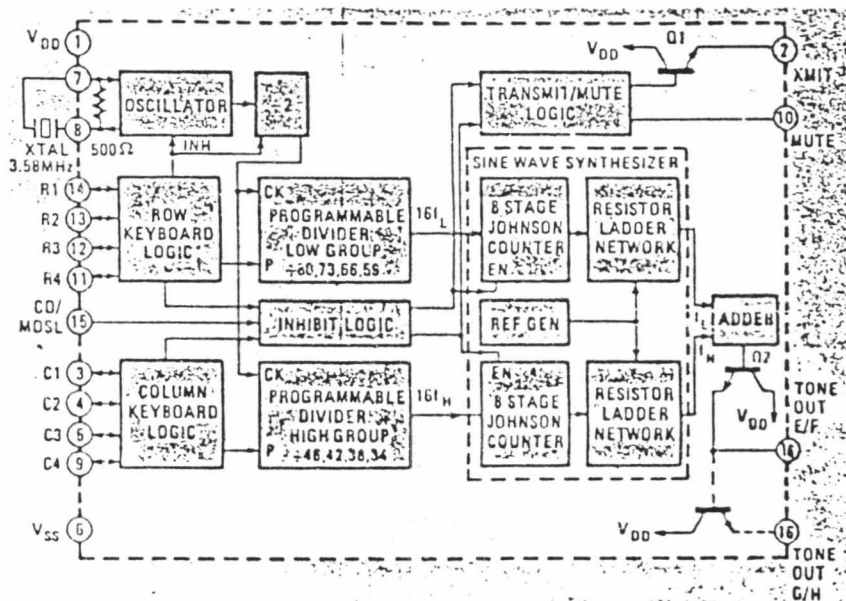


FIG. 2—INSIDE A DTMF ENCODER. The S2559E DTMF generator IC is shown here in block-diagram form.

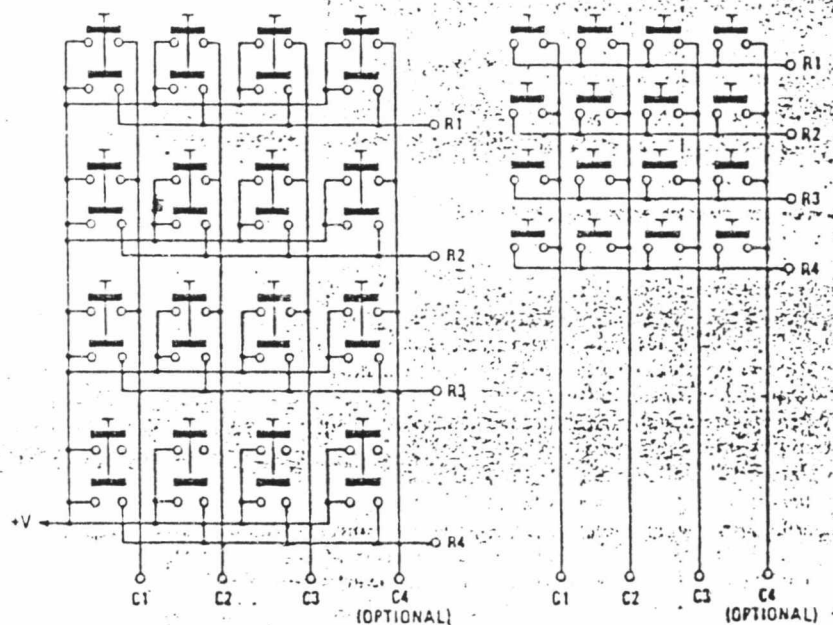


FIG. 3—TWO TYPES OF KEYPADS can be used with DTMF encoders. The one shown in a is a standard telephone tone-dialing keypad and uses DPST switches. The one shown in b is a calculator-type keypad and uses simple SPST switches, but it can not be used with all encoder IC's.

7. Silicon Systems, Mostek, Motorola, AMI, and Teltone.

Although the DTMF system was originally designed for telephone dialing, it is extremely useful as the basis for remote-control systems. In this article we'll describe some of the DTMF-encoding and decoding IC's that are commercially available, and how they can be used in remote-control applications. After reading this article, you should have no trouble choosing and successfully using the DTMF IC's that best fit your needs.

DTMF encoding

The heart of a DTMF encoder is a DTMF tone-generator IC. Those IC's are extremely easy to use and are very low in cost. Some DTMF tone generators are available for less than \$2.00 in single quantities! They generate the desired DTMF signals by dividing a crystal-generated reference frequency. The oscillator is on-board the IC; the crystal is simply connected across two terminals of the IC. The most-common crystal frequency is 3.579545 MHz; that's the TV color-burst frequency, so those crystals are readily available and low in cost. However, as we will see shortly, other frequency references may be used for special purposes.

A block diagram of a typical tone-encoder IC is shown in Fig. 2. The IC illustrated there is a Gould AMI (3800 Homestead Rd., Santa Clara, CA 95051) S2559E. The desired DTMF signals are activated by a twelve-key (3 × 4) or sixteen-key (4 × 4) matrix keypad that is connected directly to the row- and column-input pins of the tone-generator IC. Two major types of keypads are used: One is the standard telephone pushbutton keypad. They are used with IC's that generate tones whenever the corresponding row and column pins are pulled high. As shown in Fig. 3-a, that keypad consists of a series of DPST momentary switches with a common line that simultaneously pulls the corresponding row and column outputs high when pressed. Note that some encoders are active low. For those, the keypad common line is connected to ground. Then, the appropriate row and column outputs are grounded when a key is pressed. The other, and simpler, keypad arrangement is shown in Fig. 3-b. Referred to as a calculator-type or X-Y keypad, it consists of SPST momentary switches and can be built easily. However, it can only be used with tone generators that use calculator-type scanning circuitry to detect switch closures. The S2559E contains such circuitry. Generally, the datasheet of a particular tone encoder will specify the type of keypad required.

A simple DTMF encoder is shown in Fig. 4. It mainly consists of a 16-key SPST keypad like the one shown in Fig. 3-b, and the S2559E tone-encoder IC. Power can be supplied by a small power

TABLE 1—PIN FUNCTIONS

Pin	Function
SIGNAL IN	DTMF input. Internally biased so that the input signal may be AC coupled. SIGNAL IN also permits DC coupling as long as the input voltage does not exceed the positive supply.
12/T8	DTMF-signal detection control. When 12/T8 is at logic 1, the M-957 detects the 12 most commonly used DTMF signals (1 through 8). When 12/T8 is at logic 0, the M-957 detects all 16 DTMF signals (1 through 0).
A, B	Binary DTMF signal-sensitivity control inputs. A and B select the sensitivity of the SIGNAL IN input to a maximum of -31 dBm.
D3, D2, D1, D0	Data outputs. When enabled by the OE input, the data outputs provide the code corresponding to the detected digit in the format programmed by the HEX pin. The data outputs become valid after a tone pair has been detected and are cleared when a valid pause is timed.
OE	Output enable. When OE is at logic 1, the data outputs are in the CMOS push/pull state and represent the contents of the output register. When OE is driven to logic 0, the data outputs are forced to the high-impedance or "third" state.
HEX	Binary output format control. When HEX is at logic 1, the output of the M-957 is full 4-bit binary. When HEX is at logic 0, the output is binary-coded 2-of-8.
STROBE	Valid data indication. STROBE goes to logic 1 after a valid tone pair is sensed and decoded at the data outputs. STROBE remains at logic 1 until a valid pause occurs or the CLEAR input is driven to logic 1, whichever is earlier.
CLEAR	STROBE control. Driving CLEAR to logic 1 forces the STROBE output to logic 0. When CLEAR is at logic 0, STROBE is forced to logic 0 only when a valid pause is detected. Tie to VNA or VND when not used.
BD	Early signal presence output. BD indicates that a possible signal has been detected and is being validated.
XIN, XOUT	Crystal connections. When an auxiliary clock is used, XIN should be tied to logic 1.
OSC/CLK	Time base control. When osc/CLK is at logic 1, the output of the M-957's internal oscillator is selected as the time base. When osc/CLK is at logic 0 and XIN is at logic 1, the AUXCLK input is selected as the time base.
AUXCLK	Auxiliary clock input. When osc/CLK is at logic 0 and XIN is at logic 1, the AUXCLK input is selected as the M-957's time base. The auxiliary input must be 3.58 MHz divided by 8 for the M-957 to operate to specifications. If unused, AUXCLK should be left open.
VNA, VND	Negative analog and digital power supply connections. Separated on the chip for greater system flexibility, VNA and VND should be at equal potentials.
VP	Positive power supply connection.

supply or by a conventional 9-volt battery. Because the S2559E is a CMOS device, power consumption is low. Typically, the circuit shown will draw 5 mA during encoding and 7 μ A when idle. Since the device is CMOS, be sure to observe all of the standard precautions when handling the IC.

Encoder output

The output of the encoder consists of two of the eight DTMF frequencies. Figure 5 shows an oscilloscope display of the row-3 signal (852 Hz) and Fig. 6 shows the column-2 signal (1336 Hz). The DTMF output is produced by adding the two signals together. The resulting signal,

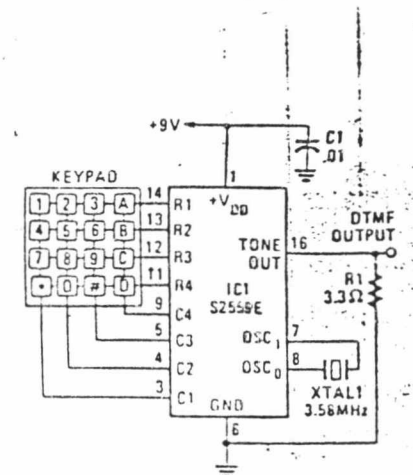


FIG. 4—A COMPLETE DTMF ENCODER requires just a keypad, an encoder IC, a crystal, and two additional components.

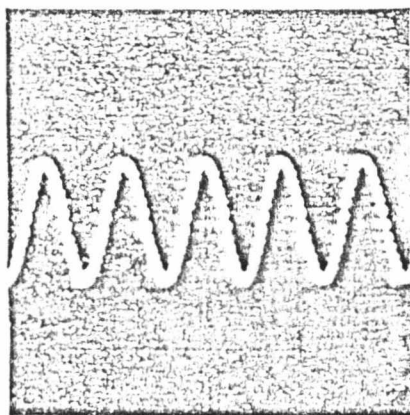


FIG. 5—A ROW 3, 852 Hz, DTMF signal.

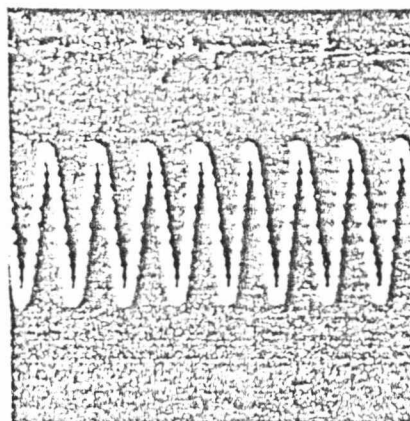


FIG. 6—A COLUMN 2, 1336 Hz DTMF signal.

which would be generated by pressing the "8" key, is shown in Fig. 7. Note that the output of the S2559E is not a pure sine-wave. Instead the output is a digitally synthesized approximation, as shown in Fig. 8.

The S2559E also is capable of generating single-frequency tones. To place the IC in the single-frequency mode, pin 15, the mode-select pin (MSEL), is either tied high or left floating; for DTMF operation, that pin is grounded. Once in the single-frequency mode, a single frequency is output by pressing two keys in the appropriate row or column. For instance, simul-

taneously pressing the 4 and 5 keys (in row 2) will result in a 770-Hz output. The single-frequency mode is used primarily for testing.

The S2559E, as well as most other encoder IC's, have mute and transmit pins (MUTE and XMIT). In the S2559E, when no keys are pressed, the MUTE pin is low and the XMIT output is enabled and can

source current to an external load. When a key is pressed, the XMIT output goes into a high-impedance state and the MUTE output goes high. Those pins are used in telephone applications. For instance, the MUTE pin is used to mute the telephone receiver during dialing so that the user does not hear the DTMF signals at full volume. The enterprising experimenter will doubtless find many other uses for those handy outputs.

To make the output of the encoder circuit audible, a speaker or some other transducer must be driven by the output signal. The S2559E output must be buffered to drive an 8-ohm speaker, but other high-impedance speakers can be driven by the IC directly. For example, the author has driven the earpiece from an old telephone headset by adding a 330-ohm resistor in series with the earpiece to prevent loading the encoder's output as well as to increase battery life.

We've been discussing the S2559E thus far, but there are three other members of that IC family. They are the S2559F, G, and H. Those four devices have replaced the earlier A, B, C, and D versions and feature extended operating voltage (2.5 to

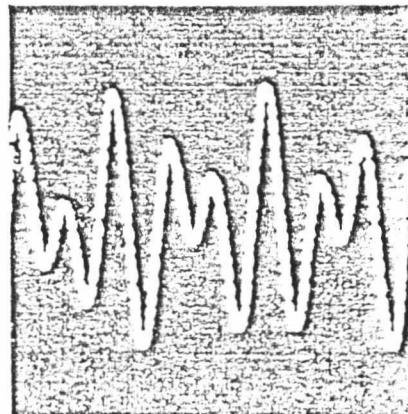


FIG. 7—WHEN THE DIGIT 8 is pressed on the keypad, the encoder generates the signal shown here. It consists of the sum of the row 3 (Fig. 5) and the column-2 (Fig. 6) signals.

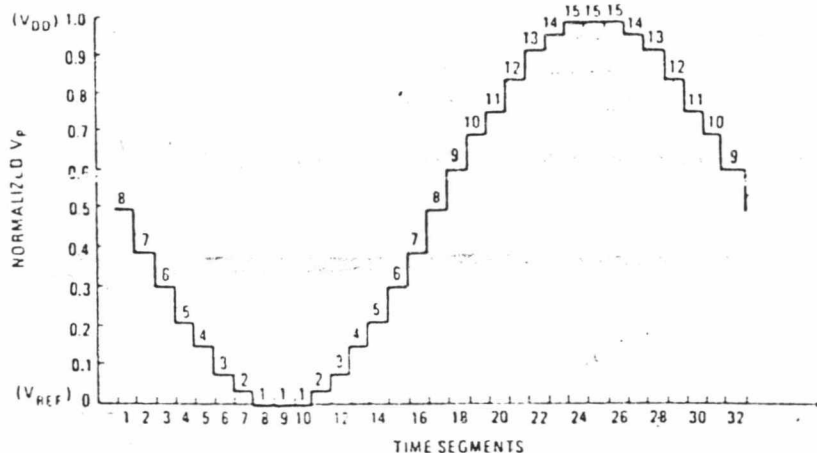


FIG. 8—THE OUTPUT OF THE S2559E is not a pure sine-wave. Instead it is a digitally synthesized waveform. The staircase-shape of such a waveform is shown here.

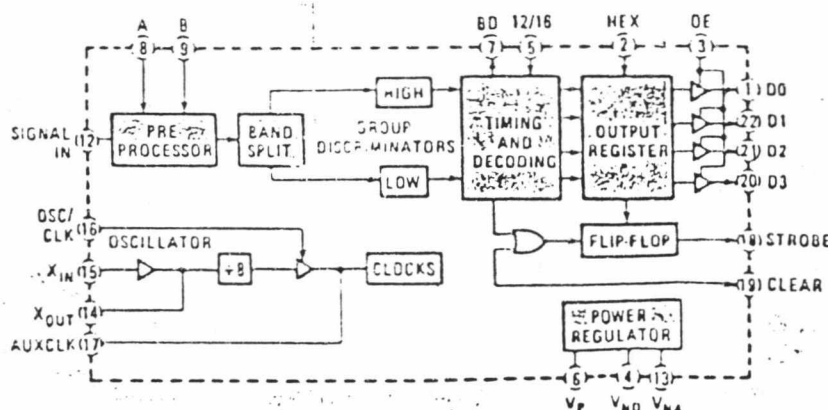


FIG. 9—INSIDE A DTMF DECODER. The S2559 DTMF decoder IC is shown here in block-diagram form.

TABLE 2—DTMF TO BINARY DECODING

Signal	Low-Frequency Component (Hz)	High-Frequency Component (Hz)	Hex Output Format	2-Of-8 Output Format
			3210	3210
1	697	1209	0001	0000
2	697	1336	0010	0001
3	697	1477	0011	0010
4	770	1209	0100	0100
5	770	1336	0101	0101
6	770	1477	0110	0110
7	852	1209	0111	1000
8	852	1336	1000	1001
9	852	1477	1001	1010
0	941	1336	1010	1101
*	941	1209	1011	1100
#	941	1477	1100	1110
A	697	1633	1101	0011
B	770	1633	1110	0111
C	852	1633	1111	1011
D	941	1633	0000	1111

Note: The M-957 detects signals A through D only when the 12/16 input is at logic 0.

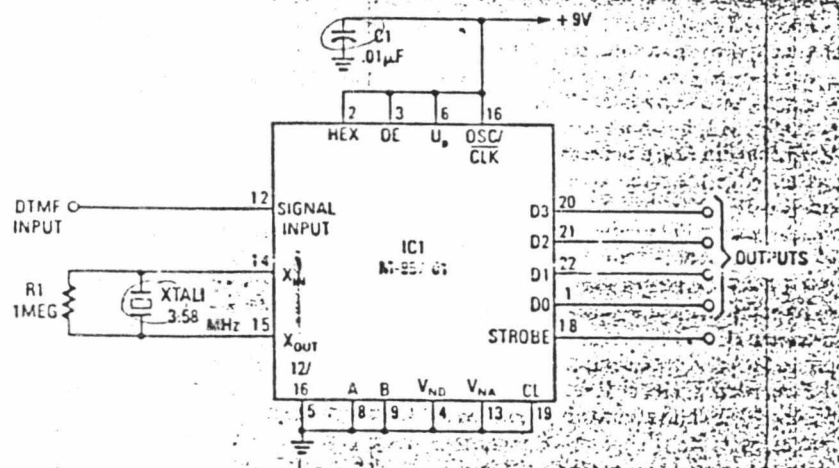


FIG. 10—USING THE M-957, building a DTMF decoder requires just the IC, a crystal, and a capacitor.

10 Volts), improved tone fidelity, and an on-chip oscillator bias-resistor. In the S2995F, the MDSL function of pin 15 is replaced by a chip disable (CD) function that is active high. When that pin is tied high, the row and column inputs are placed in a high-impedance state, the tone output is tied to ground, the oscillator is inhibited, the MUTE pin is tied high, and the XMIT pin is enabled. Essentially, the effect is that the IC is electronically disconnected from the keypad. That allows one keypad to be shared by several different devices.

The S2559G and H are identical to the S2559E and F, respectively, except that the output transistor has been replaced by a Darlington pair. In some applications that eliminates the need for an external transistor amplifier stage in the telephone circuit.

DTMF decoding

DTMF decoding is considerably more complex than DTMF encoding. The most involved function of the detector is to determine whether a received signal within the DTMF frequency band (697–1633 Hz) is a true DTMF signal or merely noise or speech. The detector must also be capable of detecting a DTMF signal that is combined with such noise. The DTMF detector should recognize any valid DTMF signal that is within $\pm 2\%$ of the standard value. The detector's job is made somewhat easier by the fact that a DTMF signal must have a minimum duration of 40 ms, and that each DTMF signal must be separated from others by at least 35 ms.

Somewhat surprisingly, most of the circuitry required to decode DTMF signals is now available in IC form. Therefore, despite its greater complexity, an entire 16-digit de-

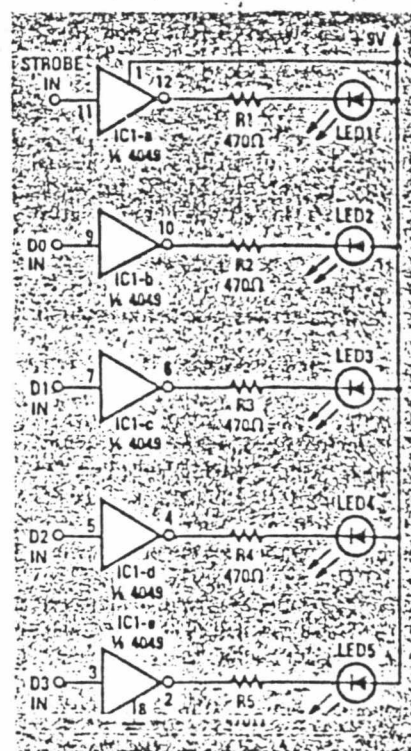


FIG. 11—USE THIS LED MONITOR CIRCUIT to verify that the encoder and decoder are operating correctly.

coder can be built as easily and as simply as a 16-digit encoder.

DTMF decoders are often referred to by manufacturers as DTMF receivers. Those devices have only recently become commonly available at affordable prices. Some can be purchased for under \$15.00 in single-unit quantities. Just a few years ago, when the first IC encoders became available, those devices cost about \$100, and required external filters. The IC's on the market today are extremely sophisticated signal-processing devices with switched-capacitor filtering that use digital frequency-detection techniques. They can reliably detect DTMF signals with no need for pre-filtering.

The decoder that we'll use in our circuit is the M-957 from Teltone (P.O. Box 657, 10801-120th Ave. N.E., Kirkland, WA 98033). That CMOS device can be powered by a DC power supply or batteries. There are two versions of the M-957: the M-957-01, which can accept voltages of 5 to 12, and the M-957-02, which is designed for 5-volt operation only.

A block diagram of the M-957 is shown in Fig. 9. The function of each pin is outlined in Table 1. The pre-processing stages of the M-957 filter out noise and split the received DTMF signal into its high and low-frequency-group components, and limit each component to provide automatic gain control. The individual tones are then detected. The decoded output of the M-957-01 is a 4-bit binary code appearing at the D0–D3 output. The output code format can be selected via pin 2, HEX. When that pin is

pin (logic 1), the output format is 4-bit decimal; when the pin is low (logic 0), the output format is binary-coded 2-8.

Putting it together

It takes very little in the way of external circuitry to put the M-957 to work. Adding just a single capacitor and a crystal as shown in Fig. 10 yields a functional DTMF receiver/decoder.

Now that we have an encoder and a decoder, the next step is to verify that both work as intended. The easiest way to do that is to wire the output of the encoder (Fig. 4) to the input of the decoder. If you are using separate sources (batteries or DC power supplies) to power the circuits, be sure to tie their grounds together.

To monitor the output states of the encoder during testing you can build a simple monitor circuit like the one shown in Fig. 11. That circuit uses $\frac{1}{4}$ of a 4049 hex inverter as a buffer to drive five indicating LED's. Those five LED's show the states of the four data outputs as well as state of the STROBE output. Table 2 shows the correspondence between the DTMF signal received and the state of the data outputs. The strobe output should be high, as indicated by a lighted LED, any time that a valid DTMF signal is received and decoded by the circuit.

Once you are sure that the decoder is operational it is time to think about adding to its usefulness and versatility. For one thing, the outputs could be further decoded to provide a 1-of-16 output. A circuit for doing that is shown in Fig. 12.

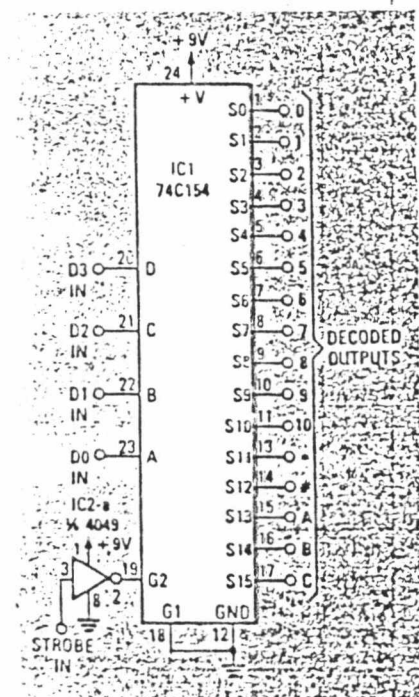


FIG. 12—THE FOUR-BIT OUTPUT of the M-957 is decoded using this circuit. With it, each DTMF code can be used to address one of the 16 outputs.

ORDERING INFORMATION

The following is available from High Technology Semiconductors, 2512 Chambers Road, Suite 204, Tustin, CA 92680, (714) 259-7733; Telex M-957-N, \$11.35; Stantel STC-5089-N (which is pin-for-pin compatible with the AMI S2559E), \$2.10; 3.58-MHz crystal, \$1.25. Also available is a kit of parts, TRK-957-N, which consists of the M-957, STC-5089, 3.58-MHz crystal, a 22-pin DIP socket, and a 1-megohm resistor for \$14.95. Please add \$2.30 shipping to all orders. MasterCard, Visa, and COD orders accepted.

Built around a 74C154 4-to-16 decoder/multiplexer, it provides 16 separate output lines. Each of the 16 DTMF signals will enable only one of the circuit's normally high outputs. For instance, if a DTMF 9 is received, only the 9 output, pin 10 of IC1, will go low. That output will remain low as long as a valid DTMF 9 is being received by the circuit.

Another useful enhancement would be to add some type of latched output. That means that once the appropriate DTMF signal is received, the output would remain either high or low until the next time the same DTMF signal is received. Such operation approximates the on/off action of a toggle or pushbutton switch.

A circuit for adding latched outputs is shown in Fig. 13. It is built around half of a 74C73 dual flip-flop that is configured to act as an edge-triggered binary divider (divide-by-2). When the circuit is used as shown, no external debounce circuitry is required. The input is shown as a DTMF D, but it could be any of the DTMF signals. Two complementary latched outputs are available. Use whichever output is appropriate for your application. Tie all of the IC's unused inputs (IC1-b) to ground to prevent oscillation and unnecessary current drain.

Switch S1 is used to clear both outputs to zero. That switch is not needed for all applications and can be eliminated if desired. Conversely, the circuit can be set up

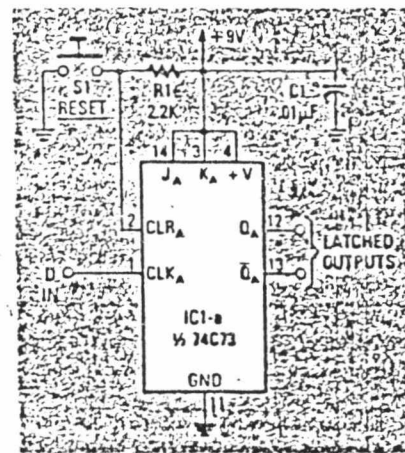


FIG. 13—LATCHED OUTPUTS can be provided using this circuit.

for remote reset. That is done by eliminating the switch and the 2.2K resistor (R1) and tying one of the momentary outputs of the 74C154 to the CLR pin of the 74C73. For example, if pin 17 of the 74C154 is connected to pin 2 of the 74C73, the latch will be reset anytime a DTMF C is received. If no reset function is desired, the CLR pin must be tied high.

Up to sixteen devices may be independently controlled by the outputs of the circuit in Fig. 12. If the controlled device is digital and if it is voltage-compatible with the decoder output, direct connection to that device is possible. If heavy driving currents are required, that current can be supplied by transistor switches located at the decoder outputs. If the voltages are not directly compatible, matching can be done using optocouplers or power-driver IC's. Also solid-state relays may be used to interface the digital signals with high-voltage, high-current loads, such as 117-volt AC household appliances, or even industrial devices with larger power requirements.

Going farther

If a wireless data link is desired, any simple, single-channel radio or infrared communications link may be used. For example, a toy walkie-talkie set or a low-cost FM wireless-microphone/FM radio system may be used.

Many DTMF tone generators can be driven by logic-level signals. That allows direct control of DTMF signalling by a microprocessor or ROM circuit. The S2559E requires active high logic-levels at all of its row and column inputs. That means that an 8-bit signal or some type of external driving circuitry is required for digital control of the IC. Other DTMF devices are better suited to digital control. One such device is AMI's S2579 DTMF tone generator with binary input. That device is designed so that a 4-bit digital signal can be used to encode all 16 DTMF signals.

The DTMF IC's will function with crystal frequencies other than those specified for DTMF operation. However, the frequencies that will be generated or decoded will differ from the standard DTMF ones. If a higher crystal frequency is used, all tones will be correspondingly higher in frequency; if a lower crystal frequency is used, all tones will be lower in frequency. That effect can be useful for applications such as when a private communications code is desired.

In this article we've presented some of the basics of DTMF communications. We've also presented some possible applications of that technology. For the enterprising experimenter there are countless more. Now that the cost of the required encoding and decoding IC's is so low, the only limit to their use is your own imagination.

R-E

ประวัติผู้เขียน

นายจักรกฤษณ์ ประกอบการ เกิดวันที่ 10 มกราคม พ.ศ. 2517 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นและมัธยมศึกษาตอนปลายจากโรงเรียนเบญจมราชรังสฤษฎิ์ จະเซียงเทร่า จังหวัดจะเซียงเทร่าจนจบการศึกษา แล้วเข้าศึกษาต่อในระดับในระดับปริญญาตรี สาขาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง และสำเร็จการศึกษาในปีการศึกษา 2537

นางสาวนิชกรินทร์ เกตุภุพงษ์ เกิดวันที่ 28 พฤษภาคม พ.ศ. 2517 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นและมัธยมศึกษาตอนปลายจากโรงเรียนราชินีบน จังหวัดกรุงเทพฯจนจบการศึกษา แล้วเข้าศึกษาต่อในระดับปริญญาตรี สาขาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง และสำเร็จการศึกษาในปีการศึกษา 2537

นางสาวหวานจิตร์ รัตนสนธิ เกิดวันที่ 9 พฤศจิกายน พ.ศ. 2516 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นและมัธยมศึกษาตอนปลายจากโรงเรียนเตรียมอุดมศึกษาพัฒนาการ จังหวัด กรุงเทพฯ จนจบการศึกษาแล้วเข้าศึกษาในระดับปริญญาตรี สาขาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง และสำเร็จการศึกษาในปีการศึกษา 2537

เอกสารอ้างอิง

รศ. ประกิจ ตั้งติสานนท์ (B.Eng., M.Eng.(JAPAN)), "วิศวกรรมการสื่อสารไฟฟ้าอิเล็กทรอนิกส์,"
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร.

ดร. วิวัฒน์ กิรานนท์, "พื้นฐานการสื่อสาร," สำนักพิมพ์สุภา, กรุงเทพมหานคร.

ธานินทร์ ถาวรศาสนวงศ์ (B.Eng.(KMIT-LADKRABANG)), ทินกร ดูก (B.Eng. (KMIT-LADKRABANG)), "การอินเทอร์เฟสไอพีเอ็มพีซี," สำนักพิมพ์ฟิสิกส์เซ็นเตอร์การพิมพ์,
กรุงเทพมหานคร.

ประดลย์ พึ่งเงิน, นิรันดร์ อินทวงศ์, ณรงค์ คำภู, "วิทยานิพนธ์เครื่องบันทึกโทรศัพท์อัตโนมัติ," ภาควิชาโทรคมนาคม, คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีเจ้าคุณทหาร
ลาดกระบัง, กรุงเทพมหานคร.