# เครื่องบันทึกเสียงในคอมพิวเตอร์

# VOICE RECORDER


โดย

นางสาว กลัญชละ วิทยะสิรินันท์

นางสาว พิมลจิตต์ ธรรมสารสุนทร



วัน เดือน ปี 14 ม.ค. 2539
เลขทะเบียน 034121
เลขเรียกหนังสือ T 371021 ก4


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์สำหรับภาคการศึกษาที่ 2 ปีการศึกษา 2537

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

# เครื่องบันทึกเสียงในคอมพิวเตอร์

# VOICE RECORDER


ผู้จัดทำ

นางสาว กลัญชละ วิทยะสิรินันท์

นางสาว พิมลจิตต์ ธรรมสารสุนทร


อาจารย์ที่ปรึกษา

( อ. ขนิษฐา แซ่ตั้ง )


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เครื่องบันทึกเสียงในคอมพิวเตอร์

นางสาว กลัญชละ วิทยะสิรินันท์
นางสาว พิมลจิตต์ ธรรมสารสุนทร

อ. ขนิษฐา แซ่ตั้ง อาจารย์ที่ปรึกษา
ปีการศึกษา 2537

**บทคัดย่อ**

เครื่องบันทึกเสียงในคอมพิวเตอร์นี้ แบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนบันทึกเสียงลงในคอมพิวเตอร์ และส่วนนำข้อมูลที่เก็บไว้แปลงกลับเป็นสัญญาณเสียง ซึ่งในส่วนการบันทึกเสียงสามารถบันทึกข้อมูลได้โดยมีความยาวไม่จำกัด และเก็บข้อมูลได้หลายชุด ทั้งนี้ความยาวของข้อมูลจะขึ้นกับขนาดของหน่วยความจำ ที่ใช้เก็บข้อมูลในส่วนการนำข้อมูลออกมา มีทั้งหมด 8 ช่องสัญญาณเสียง การทำงานของแต่ละ ช่องสัญญาณจะเป็นอิสระแก่กัน สามารถเรียกใช้งาน ช่องสัญญาณเสียงใดเมื่อใดก็ได้

โครงงานนี้ใช้การเข้ารหัสแบบเดลต้ามอดูเลชั่น ในการแปลงสัญญาณระหว่างสัญญาณอนาลอก(สัญญาณเสียง) และสัญญาณดิจิตอล วิธีนี้ทำให้ประหยัดเนื้อที่หน่วยความจำ โดยใช้ IC เบอร์ MC3418 ซึ่งเป็น A/DและD/A ในชิพตัวเดียวกัน ในส่วนของซอฟแวร์ควบคุมการจัดการข้อมูลจะใช้หลักการ คือการวนตรวจสอบ(Polling) ว่ามี ช่องสัญญาณใด ต้องการข้อมูลแล้วจึงส่งข้อมูลให้ทีละ 1 ไบต์ ในขณะเดียวกันก็จะตรวจสอบความต้องการขอเริ่มใช้ข้อมูลและขอเลิกการใช้ จากแต่ละช่องสัญญาณในขณะเดียวกันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# VOICE RECORDER

MISS KALANCHALA VIDHAYASIRINUN

MISS PIMOLJIT THAMMASARNSOONTHORN

ADVISOR MISS KANITTHA SAETANG

ACADEMIC YEAR 2537

## ABSTRACT

VOICE RECORDER is divided into 2 parts ,voice recording part and voice playing part. The recording part can record unlimited data and many files .It is depend on an amount of data storage(Harddisk) . The playing part consists of 8 voice channels. Each of channel works indepentdent ,they are able to be used at anytimes.

Conversion of analog signal and digital signal of this project uses Delta Modulation for encoding and decoding data. This method uses less memories than others. So,IC MC 3418s are used as A/D and D/A converters. Software for controlling data management uses polling method in order to check which channel needs data . Then ,computer will transfer a byte of data to that channel and it always checks the request signal for connection from another unused channel .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ต้องขอกราบขอบพระคุณ อาจารย์ ขนิษฐา แซ่ตั้ง และอาจารย์ อัครินทร์ คุณกิตติ ที่ได้กรุณาเป็นที่ปรึกษา ให้ความรู้และคำแนะนำ ตลอดจนให้ความอนุเคราะห์ด้านอุปกรณ์ เครื่องมือเครื่องใช้ และห้องปฏิบัติงาน

และขอขอบคุณรุ่นพี่และเพื่อนๆที่ได้ ให้คำแนะนำ ให้กำลังใจ และให้ความช่วยเหลือเป็นอย่างมากจนปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| | | |
|---|---|---|
| **บทที่ 1** | **บทนำ** | 1 |
| **บทที่ 2** | **หลักการเบื้องต้นของการเข้ารหัสสัญญาณแบบดิจิตอล** | 4 |
| **บทที่ 3** | **การบันทึกเสียงระบบดิจิตอล** | 17 |
| **บทที่ 4** | **หลักการทำงาน** | 21 |
| 4.1 | ส่วนฮาร์ดแวร์ | 21 |
| 4.2 | ส่วนซอฟแวร์ | 22 |
| **บทที่ 5** | **รายละเอียดการออกแบบและการทำงาน** | 24 |
| 5.1 | ฮาร์ดแวร์ | 24 |
| 5.1.1 | การบันทึกเสียง | 24 |
| 5.1.2 | การกระจายเสียง | 34 |
| 5.2 | ซอฟแวร์ควบคุมการทำงานของวงจร | 41 |
| **บทที่ 6** | **สรุปโครงงาน** | 47 |
| **ภาคผนวก** | | |
| **บรรณานุกรม** | | |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

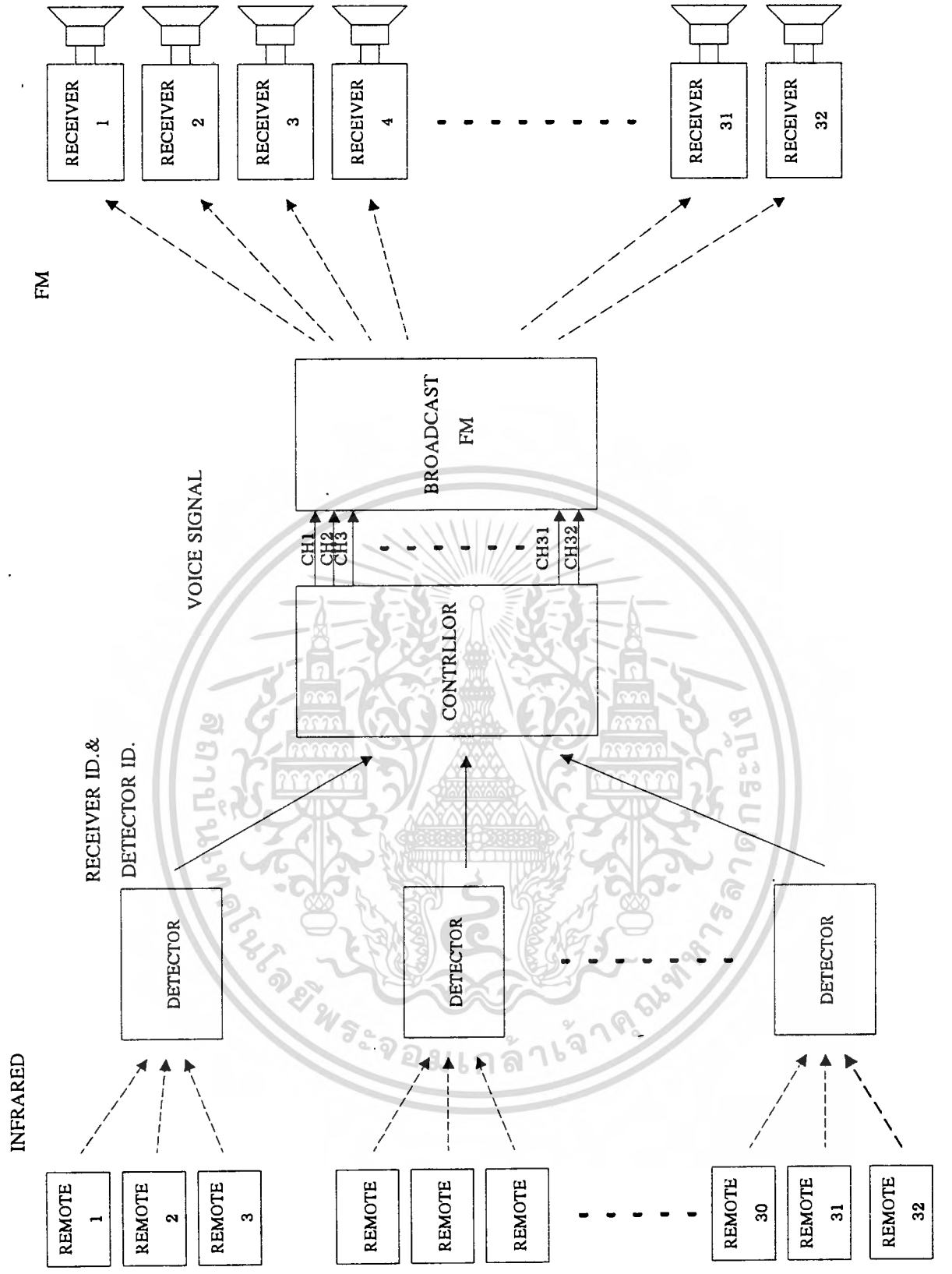
ตามที่ได้ทราบกันดีอยู่แล้วว่า ในการจัดงานนิทรรศการ พิพิธภัณฑ์ งานแสดงสินค้า จะต้องใช้เจ้าหน้าที่หรือ presenter จำนวนมาก เพื่อบรรยายรายละเอียดของสิ่งที่แสดงในงานนั้นๆ ทั้งๆที่ในปัจจุบันนี้เทคโนโลยีได้ก้าวหน้ามาถึงระดับที่สามารถนำมาประยุกต์ใช้กับงานที่กล่าวมาแล้วเป็นอย่างดี ดังนั้นทางคณะผู้จัดทำจึงคิดประดิษฐ์ Automatic Presentation Machine ขึ้นเพื่อใช้ในงานบรรยายเหล่านั้นแทนคนซึ่งเป็นทรัพยากรที่มีคุณค่า การใช้ Automatic Presentation Machine สามารถกำจัดข้อเสียในการใช้เจ้าหน้าที่บรรยายได้ ซึ่งก็คือ การที่ผู้ร่วมงานต้องรอให้ถึงรอบการบรรยายใหม่เพื่อจะได้ฟังเรื่องราวตั้งแต่ต้นจนจบ การที่ต้องรอให้มีผู้ชมจำนวนมากพอเจ้าหน้าที่จึงจะเริ่มบรรยาย และการที่มีเสียงรบกวนทั้งจากผู้บรรยายหัวข้ออื่นและจากผู้ชมเอง ทำให้ผู้ที่รับฟังการบรรยายอยู่ไม่สามารถรับฟังการบรรยายได้อย่างชัดเจน Automatic Presentation Machine จะบรรยายผ่านทางหูฟังของผู้เข้าชมงาน โดยเครื่องรับที่จะแจกให้กับผู้เข้าชมงานจะประกอบด้วยหูฟังและรีโมตคอนโทรล ซึ่งสามารถใช้งานได้ง่ายเพียงผู้เข้าชมงานใส่หูฟังและเมื่อต้องการจะทราบเรื่องราวในหัวข้อใดก็เพียงแต่กดรีโมต ผู้เข้าชมงานก็จะได้รับฟังการบรรยายเรื่องราวที่ต้องการทราบทันที

โครงงานนี้ประกอบด้วยการทำงาน 3 ส่วนใหญ่ๆ คือ

1. ส่วนรีโมต (ทั้งเครื่องส่งและเครื่องรับ)
2. ส่วนการจัดการข้อมูล (บันทึกเสียงเก็บลงในคอมพิวเตอร์และนำข้อมูลที่เก็บไว้แปลงกลับเป็นเสียง)
3. ส่วนกระจายเสียง (ทั้งเครื่องส่งและเครื่องรับ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่** 1.1 BLOCK DIAGRAM แสดงโครงสร้างของโครงงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของโครงงาน

เครื่องรับแต่ละเครื่องจะประกอบด้วย 2 ส่วน คือ รีโมตและเครื่องรับฟัง จาก block diagram รูปที่ 1.1 ด้านบนสามารถอธิบายการทำงานได้ดังนี้

เมื่อมีการกดรีโมตไปที่ DETECTOR DETECTORจะ detect สัญญาณว่าส่งมาจากเครื่องรับที่มี ID เบอร์อะไร แล้วจะส่งเบอร์ ID ของเครื่องรับและ ID ประจำ DETECTOR ไปให้ส่วน CONTROL เพื่อบอกว่าเครื่องรับเครื่องใดต้องการทราบข้อมูลเรื่องใด ส่วน CONTROL จะนำข้อมูลที่เก็บอยู่ใน harddisk ซึ่งอยู่ในรูปสัญญาณ digital ออกมา แล้วแปลงกลับเป็นสัญญาณเสียง ( สัญญาณ analog ) ส่งไปให้กับส่วน BROADCAST (โดยที่เครื่องรับแต่ละเครื่องจะใช้ความถี่พาห์เฉพาะของเครื่องนั้นๆ) ส่วน BROADCAST จะกระจายเสียงแบบ FM ด้วยความถี่พาห์ของเครื่องรับที่ขอข้อมูลมา

โปรเจคที่จะกล่าวถึงต่อไปนี้ คือส่วนของการจัดการข้อมูล ทั้งการบันทึกเสียงเก็บลงในคอมพิวเตอร์ และการนำข้อมูลจากคอมพิวเตอร์ออกมาแปลงเป็นเสียง ซึ่งการทำงานของโปรเจคชิ้นนี้สามารถเก็บข้อมูลได้ยาวไม่จำกัด (ขึ้นกับขนาดของหน่วยความจำ) สามารถเก็บข้อมูลได้หลายข้อมูล และสามารถนำข้อมูลที่เก็บไว้ออกมาได้พร้อมๆกันทีละหลายๆข้อมูล

คุณสมบัติของเครื่องบันทึกเสียง

1. สามารถบันทึกข้อมูลได้มากกว่า 1 ชุด
2. สามารถบันทึกข้อมูลเสียงที่มีความยาวไม่จำกัด
3. มี 8 ช่องสัญญาณเสียงให้เลือกกระจายเสียง
4. การทำงานของแต่ละช่องสัญญาณเสียงทำงานเป็นอิสระ (ไม่ขึ้นกับช่องสัญญาณเสียงอื่นๆ)
5. แต่ละช่องสัญญาณเสียงสามารถเลือกข้อมูลที่เก็บไว้ได้ตามต้องการ
6. การเลือกใช้งานช่องสัญญาณเสียงแต่ละช่องสัญญาณสามารถเริ่มและหยุดได้ทันทีเมื่อต้องการ

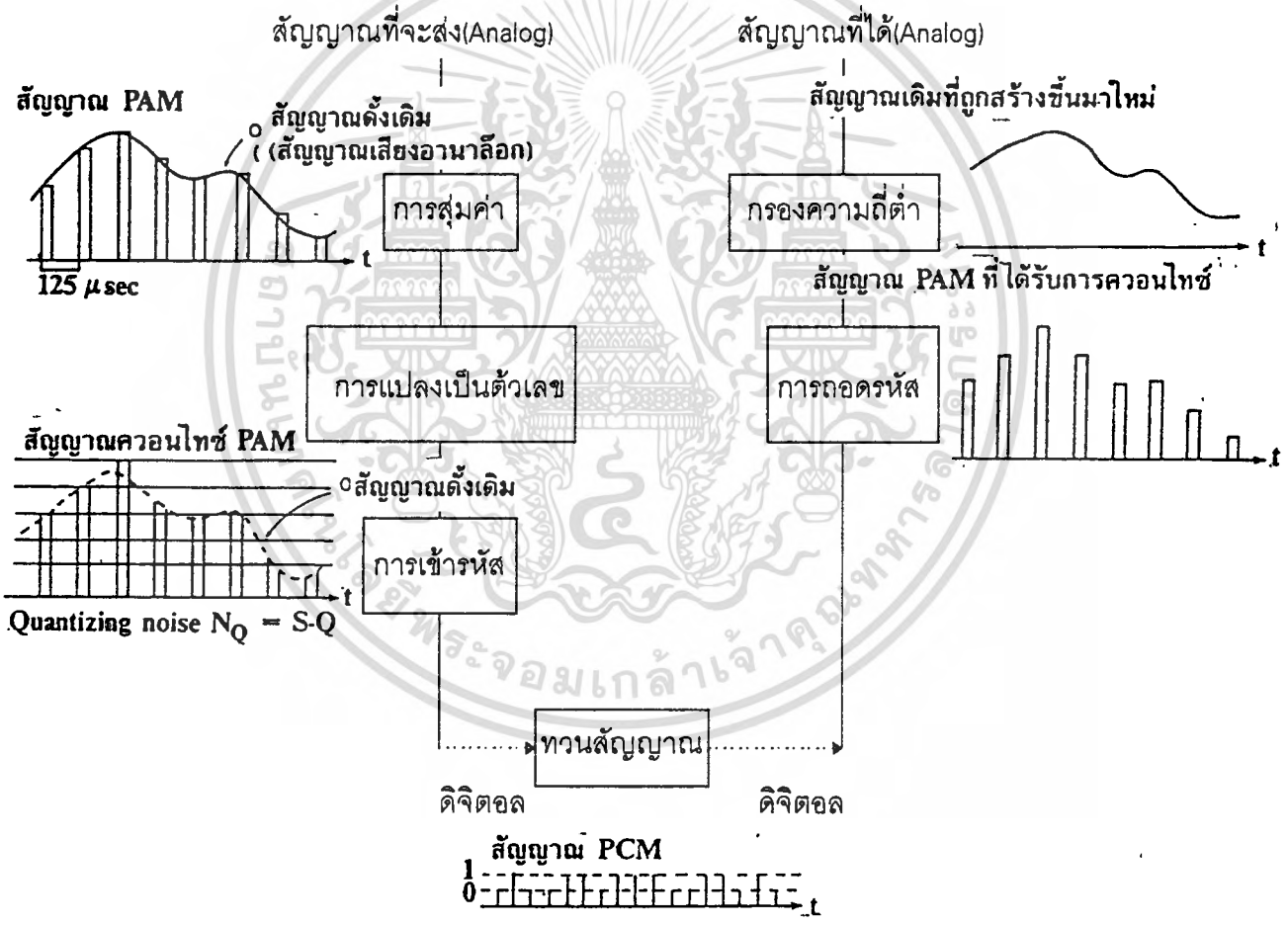


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

## หลักการเบื้องต้นของการเข้ารหัสสัญญาณแบบดิจิตอล

ระบบการสื่อสารปัจจุบัน การมอดดูเลชั่นของสัญญาณดิจิตอล ถูกนำมาใช้แทนการมอดดูเลชั่นแบบอนาลอก นั่นคือสัญญาณที่เป็นอนาลอกจะถูกเปลี่ยนเป็นสัญญาณดิจิตอลก่อน แล้วทำการมอดดูเลชั่นสัญญาณที่ได้ เพื่อทำการส่งผ่านทางสายส่งหรือคลื่นวิทยุ หลังจากนั้นทางด้านรับก็จะทำการแปลงสัญญาณดิจิตอลกลับมาเป็นสัญญาณอนาลอกตามเดิม




รูปที่ 2.1 แสดงขั้นตอนการแปลงสัญญาณ Analog ไปเป็นสัญญาณ Digital

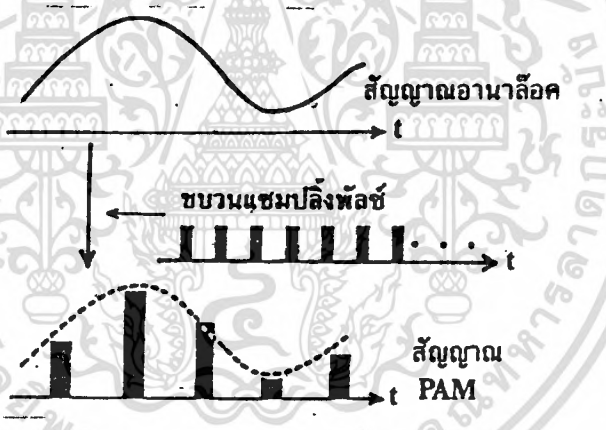


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการของระบบการมอดดูเลชั่นสัญญาณดิจิตอล จะต้องผ่านขบวนการที่สำคัญ 3 ขบวนการคือ

1. การสุ่มตัวอย่าง (Sampling)
2. การแบ่งแรงดันออกเป็นระดับต่างๆ
3. การเข้ารหัสตามระดับแรงดัน

สัญญาณอนาลอกจะเป็นสัญญาณที่ต่อเนื่องกันตลอด ตามแกนของเวลา การสุ่มค่า (Sampling) . คือขบวนการนำค่าแอมปลิจูดของสัญญาณอนาลอกบางค่าในช่วงเวลาที่ห่างกันคงที่มาเรียงต่อกันซึ่งเสมือนการมอดดูเลททางแอมปลิจูด โดยคลื่นพาหะที่เป็นขบวนพัลส์ซึ่งมีคาบเวลาระหว่างพัลส์ที่คงที่ โดยผลลัพธ์ที่ได้จะเป็นสัญญาณที่ไม่ต่อเนื่องตามแกนเวลาเรียกว่า Pulse Amplitude Modulation (PAM) ดังรูปที่ 2.2 .

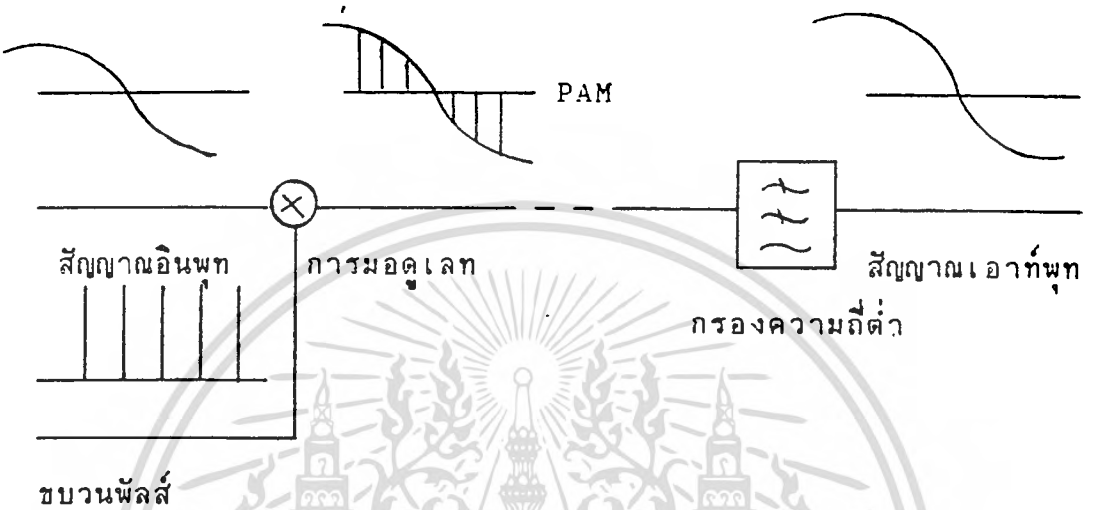



รูปที่ 2.2 การมอดดูเลชั่นระหว่างสัญญาณอนาลอกกับสัญญาณพัลส์

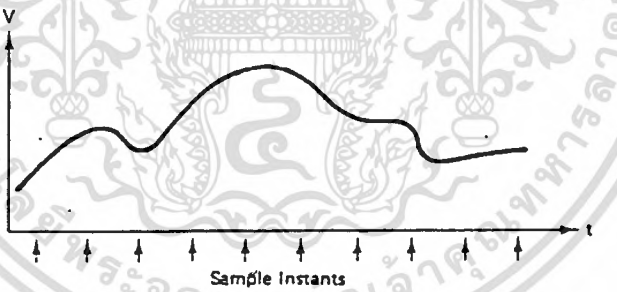
สิ่งที่จะต้องคำนึงถึงในการสุ่มสัญญาณคือ จะใช้ความถี่เท่าไรจึงจะเหมาะสมนั่นคือ ในการสุ่มค่าจะอาศัยทฤษฎีการสุ่มค่าหรือ ทฤษฎีการ Sampling (Sampling Theorem) ซึ่งได้ผ่านการพิสูจน์ทางคณิตศาสตร์แล้วโดยมีแนวความคิดเบื้องต้นที่ว่า ขบวนการพัลส์ที่เอาท์พุทของมอดดูเลเตอร์ จะเปลี่ยนแปลงไปตามระดับของสัญญาณอินพุทดังรูปที่ 2.3 สำหรับทางด้านรับนั้น เมื่อองค์ประกอบความถี่สูงของขบวนพัลส์ ถูกจำกัดออกมาโดยวงจรกรองความถี่ต่ำ (Lowpass Filter) แล้วจะได้รับสัญญาณเดิม หรือกล่าวอีกนัยหนึ่งว่าถ้าหากทำการแซมปลิ้งสัญญาณอินพุทด้วยระยะห่างที่เท่ากัน

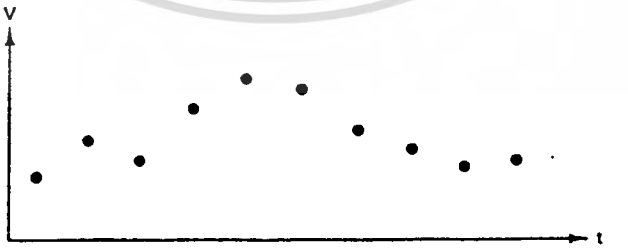
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้ว ก็สามารถจะทำให้เกิดสัญญาณเดิมได้อย่างสมบูรณ์ทางด้านรับ ซึ่งตามทฤษฎีการแซมปลิ้ง กล่าวไว้ว่า " ถ้าข่าวสารในสัญญาณถูกจำกัดให้มี ความถี่สูงเป็น$f_m$แล้วหากใช้ขบวนพัลส์ที่มีความถี่เท่ากับหรือมากกว่า$2f_m$ ทำการแซมปลิ้งแล้ว ก็จะสามารถเก็บข่าวสารได้อย่างสมบูรณ์ "ระยะห่างของแต่ละจุดที่ทำการแซมปลิ้ง คือ $1/2f_m$เรียกว่า Nyquist Interval




รูปที่ 2.3 แสดงแนวความคิดในทฤษฎีการแซมปลิ้ง



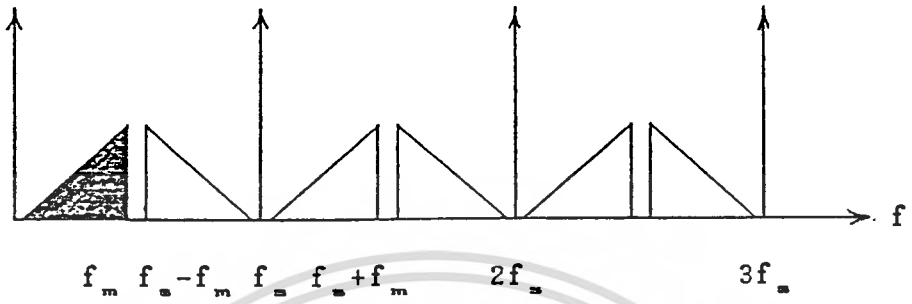




รูปที่ 2.4 แสดงระยะห่างของการแซมปลิ้ง

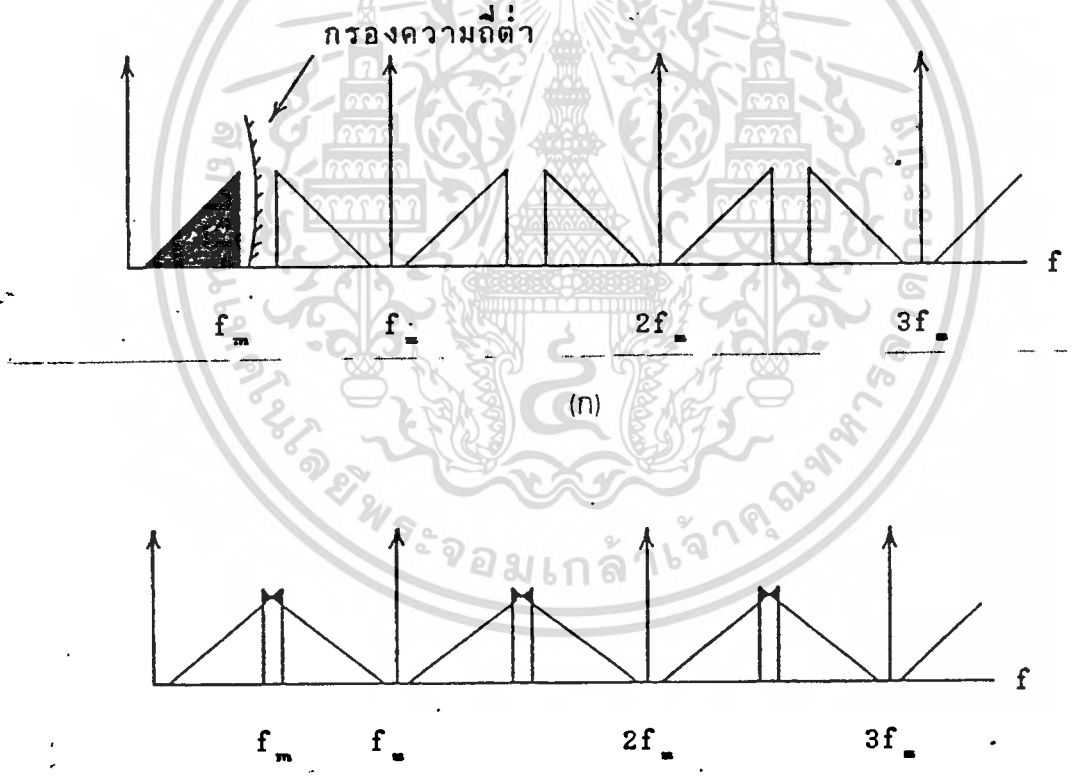


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผ่านการแซมปลิ้งแล้ว ความถี่สูงของสัญญาณอนาลอก($f_m$)และความถี่ของการแซมปลิ้ง ($f_s$) ให้แถบความถี่( FrequencySpectrum)ของสัญญาณPAMดังรูปที่2.5จากรูปจะพบว่ามันจะประกอบด้วย ฮาร์โมนิคของสัญญาณคลื่นรูปซายน์ ที่มีความถี่เป็นทวีคูณของความถี่พัลส์รวมกันอยู่ (0, $f_s$,$2f_s$,..)




รูปที่ 2.5 แสดงPAM ของสัญญาณที่ผ่านการแซมปลิ้งแล้ว




รูปที่ 2.6 แสดงผลของความถี่สัญญาณสุ่มค่าและสัญญาณอนาลอก

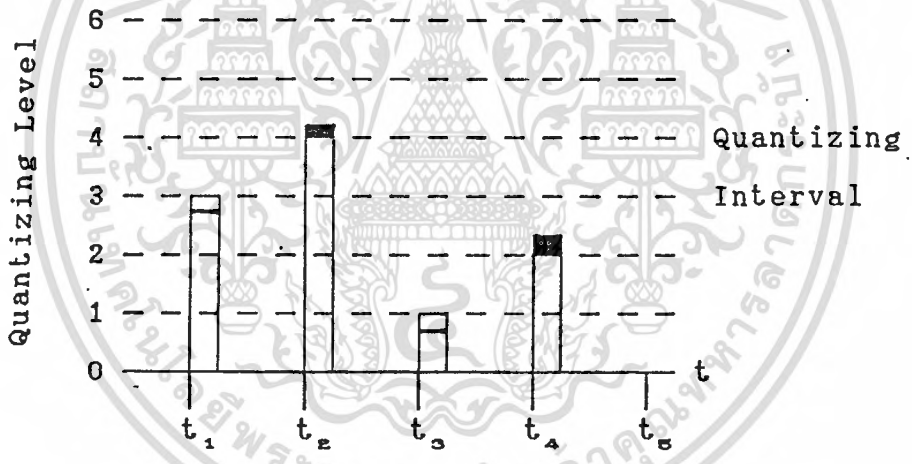


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางด้านรับนั้น จะสามารถนำสัญญาณอนาลอกกลับคืนมาได้ โดยวงจรกรองความถี่ต่ำ ซึ่งแสดงได้ดัง รูป 2.6

รูป2.6ก ถ้าความถี่ $f_s$ มากกว่าสองเท่าของความถี่ $f_m$ แล้ว จะไม่มีการซ้อนทับกันของความถี่ข้างเคียงซึ่งตรงข้ามกับรูปที่ 2.6ข ที่ใช้ความถี่สุ่มค่าน้อยกว่าสองเท่าของความถี่เสียง ที่ไม่สามารถสร้างสัญญาณอนาลอกกลับคืนมาได้ดังเดิม

ในระบบโทรศัพท์ ที่มีแถบความถี่เสียงจำกัด มีค่าอยู่ระหว่าง 0.3kHz ถึง 3.4kHz ดังนั้น ความถี่ที่ใช้ในการแซมปลิ้งต้องมีค่าไม่ต่ำกว่า 6.8kHz ในทางปฏิบัติจะใช้ 8kHzเพื่อให้การสร้างวงจรสัญญาณง่ายขึ้นดังนั้นช่วงเวลาที่ใช้ในการแซมปลิ้งแต่ละครั้งจะเท่ากับ1/8000 วินาที(125ไมโครวินาที) เมื่อสุ่มค่าได้สัญญาณ PAM แล้วจะถูกส่งไปผ่านการแปลงค่าเป็นตัวเลข โดยการแบ่งขนาดออกเป็นช่วงๆ ดังรูปที่2.7




รูปที่ 2.7 แสดงการจัดระดับสัญญาณ

จากรูปที่ 2.7 ขนาดของสัญญาณ (Amplitude) ที่ถูกสุ่มค่าจะถูกจัดให้เป็นระดับเรียกว่าระดับ ควอนไตซ์ (QuantizingLevel)โดยมีระยะห่างแต่ละระดับเรียกว่า Quantizing Intervalหรือ QuantizingStep ที่เท่ากันซึ่งเป็นการจัดระดับแบบยูนิฟอร์ม ขนาดของสัญญาณสุ่มค่า (Sample) ทุกตัวจะแสดงด้วยค่าระดับควอนไตซ์ที่ใกล้เคียงที่สุด เช่น การจัดให้มีช่วงการควอนไตซ์ห่างกันอยู่ 1 ถ้าสัญญาณสุ่มค่าเป็น 2.8 ก็จะถูกจัดให้เป็นระดับ 3 ดังนั้น PAM ที่ถูกจัดระดับแล้วจะเป็นเพียงค่า

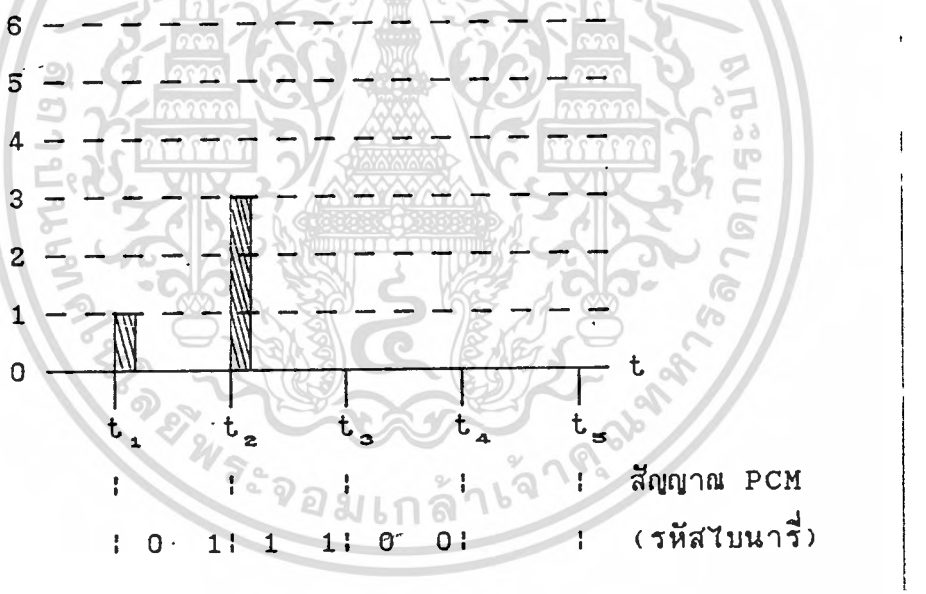


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมาณ ของสัญญาณอนาลอกเท่านั้น ส่วนที่เกินและส่วนเกินของการจัดระดับจึงเป็นค่าผิดพลาดของสัญญาณเดิม กับค่าที่ได้จัดระดับ เรียกค่านี้ว่า ควอนไตซิ่งนอยส์(Quantizing Noise)

ในทางปฏิบัติ เราไม่อาจเลี่ยงควอนไตซิ่งนอยส์ได้แต่สามารถรักษาคุณภาพของเสียงได้โดยการลดควอนไตซิ่งอินเตอร์วัลให้แคบลงหลังจากสัญญาณได้รับการจัดระดับแล้วก็จะถูกนำไปเข้ารหัสโดยเปลี่ยนเป็นรหัสเลขฐานสองดังรูปที่ 2.8

ในการเข้ารหัสอาจเป็นการเข้ารหัสแบบธรรมดาแบบเกรย์ หรือแบบสมมาตรก็ได้ซึ่งจำนวนบิทที่ใช้ในการเข้ารหัสอาจใช้เป็นหลายๆบิท สัญญาณที่ได้นี้เรียกว่า สัญญาณPCM (Pulse Code Modulation)โดยระดับค่าตัวเลขจะถูกกำหนดโดยจำนวนบิทของเลขฐานสองของแต่ละค่าเช่นใช้ 7 บิท จะได้ค่าตัวเลข 128ระดับนั่นคือจำนวนค่าตัวเลขเท่ากับ $2^n$ เมื่อ n คือจำนวนบิท




รูปที่ 2.8 การนำสัญญาณ PAM มาเข้ารหัส

จากทฤษฎีสายส่งสัญญาณ ทำให้ทราบว่าหากทำการส่งสัญญาณPCM ไปตามสายส่งสัญญาณจะเกิดการผิดเพี้ยนของสัญญาณได้เนื่องจากสัญญาณรบกวน,การแทรกสอด (Interference) ระหว่างทางและ การตอบสนองทางความถี่ (Frequency response)ของสายส่งซึ่งมีการลดทอนมากสำหรับสัญญาณความถี่สูงและลดทอนน้อยสำหรับความถี่ต่ำ เนื่องจากค่าของตัวเหนี่ยวนำ และตัว



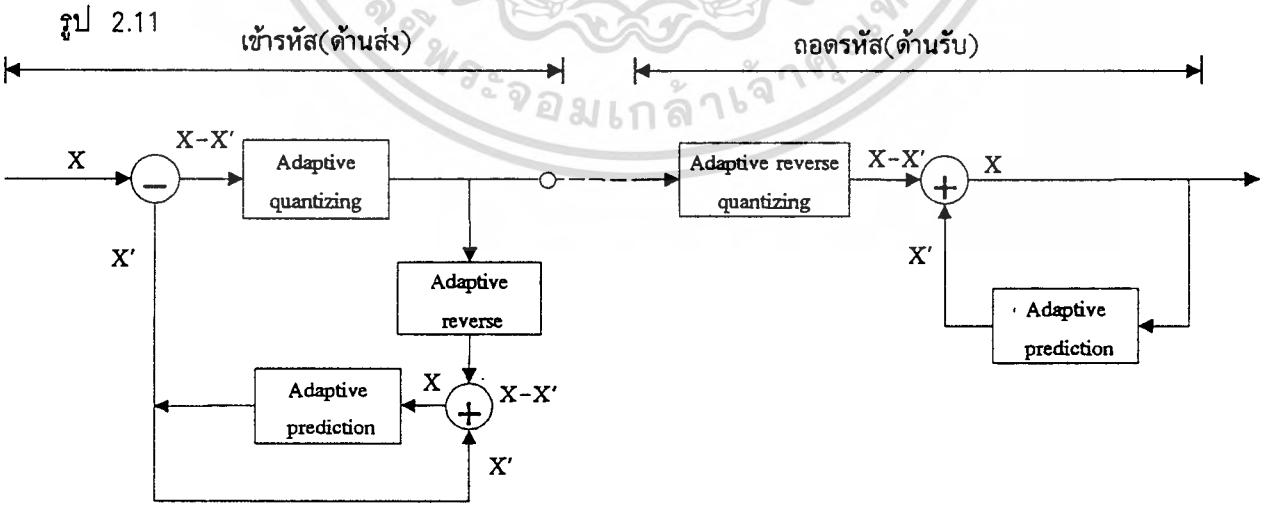
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านรับเมื่อเครื่องรับได้สัญญาณดิจิตอลPCM ก็จะแปลงกลับไปเป็นสัญญาณอนาลอก(ดูรูปที่ 2.10 ) โดยผ่านการถอดรหัส และการกรองสัญญาณซึ่งเรียกขั้นตอนทั้งสองรวมกันว่า การเปลี่ยนสัญญาณดิจิตอลไปเป็นสัญญาณอนาลอก(D/A Conversion)

การถอดรหัสนั้นก็ตรงข้ามกับการเข้ารหัส (ในรูปที่ 2.10)โดยเริ่มต้นจาก รหัสฐานสองที่มาจากเครื่องรับสัญญาณPCMจะถูกนำมาคำนวณและสร้างเป็นระดับค่าตัวเลขและสัญญาณสุ่มค่าจะถูกสร้างขึ้นใหม่ซึ่งสอดคล้องกับระดับที่คำนวณได้จากข้อมูลฐานสองที่ได้รับนี้สัญญาณ PCM ที่ถูกแปลงเป็นตัวเลขแล้ว ที่ด้านส่งก็จะถูกสร้างขึ้นใหม่ที่ด้านรับซึ่งสัญญาณที่ได้ก็ยังคงมีสัญญาณรบกวนจากการแปลงเป็นตัวเลขเช่นเดียวกับทางด้านส่ง สัญญาณPCM ที่สร้างขึ้นใหม่ที่ด้านรับ ก็จะถูกส่งผ่านไปยังวงจรกรองผ่านความถี่ต่ำ ก็จะได้สัญญาณเสียงอนาลอกต่อเนื่องตามแกนเวลา

<u>การเข้ารหัสแบบอแด็พทิพว์ดิฟเฟอเรนเชียลพีซีเอ็ม</u>( Adaptive Differential Pluse Code Modulation )

ในระบบ PCM ที่กล่าวมาแล้วนั้นจะจัดระดับของสัญญาณเสียงจากแซมเปิ้ลโดยตรง แต่สำหรับในระบบ ADPCM จะจัดระดับโดยใช้ผลต่างระหว่างค่าแซมเปิ้ล เนื่องจากค่าแซมเปิ้ลซึ่งอยู่ใกล้เคียงกันจะมีลักษณะคล้ายคลึงกัน ดังนั้นจึงคาดคะเนสัญญาณอินพุทที่เข้ามาก่อน แล้วจัดระดับโดยใช้ผลต่างระหว่างสัญญาณที่คาดคะเนและสัญญาณปัจจุบันเพื่อทำรหัสส่งออกไปและผลต่างนี้มีค่าน้อย จึงทำให้จำนวนบิทในการเข้ารหัสลดลงได้ โครงสร้างเบื้องต้นของ ADPCM แสดงไว้ดังรูป 2.11



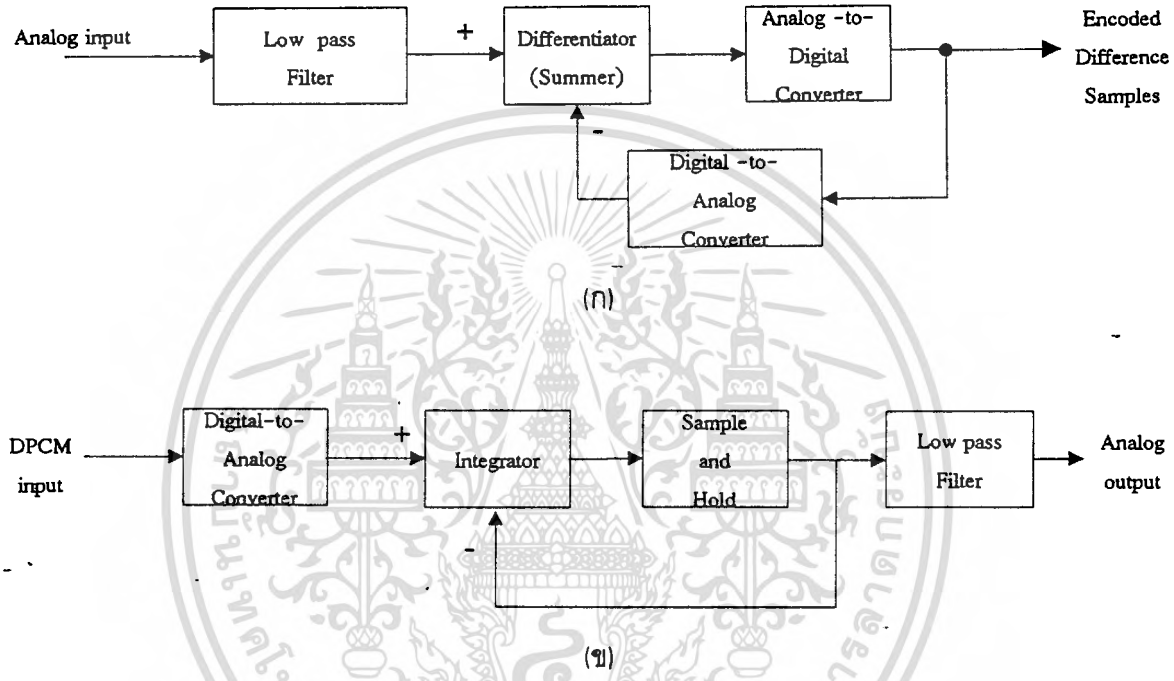

รูปที่ 2.11 โครงสร้างของระบบ ADPCM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.11 ขั้นแรก ที่วงจรเข้ารหัส (ด้านส่ง)จะสร้างผลต่าง X-X'ระหว่างสัญญาณอินพุท X และสัญญาณที่คาดคะเน X' จากอินพุทตัวก่อน ๆ โดยวงจรดิฟเฟอเรนติเอเตอร์(Differentiator) ต่อจากนั้น จะนำผลต่าง X-X' ของสัญญาณไปจัดระดับ การที่จะทำให้สัญญาณซึ่งผิดพลาดนี้มีการเปลี่ยนแปลงของระดับอย่างรวดเร็วนั้นจะไม่ใช้การจัดระดับแบบยูนิฟอร์ม แต่จะใช้วิธีจัดระดับเพื่อให้เพิ่มขึ้นหรือลดลงโดยอาศัยแซมเปิ้ลตัวก่อน (Adaptive Quantizing)วิธีนี้จะปรับปรุงคุณภาพให้ดีขึ้น

วิธีการเข้ารหัสแบบ ADPCM นี้จะใช้เพียง 4 bit (32 kb/s) ซึ่งมีคุณภาพใกล้เคียงกับระบบ ซึ่งใช้ 7 bit (56 kb/s)




รูปที่ 2.12 การเข้ารหัสแบบ Adaptive Delta Modulation

(ก)การเข้ารหัส (ข)การถอดรหัส

<u>การเข้ารหัสแบบอแด็พทิฟว์เดลต้ามอดูเลชั่น</u> ( Adaptive Delta Modulation )

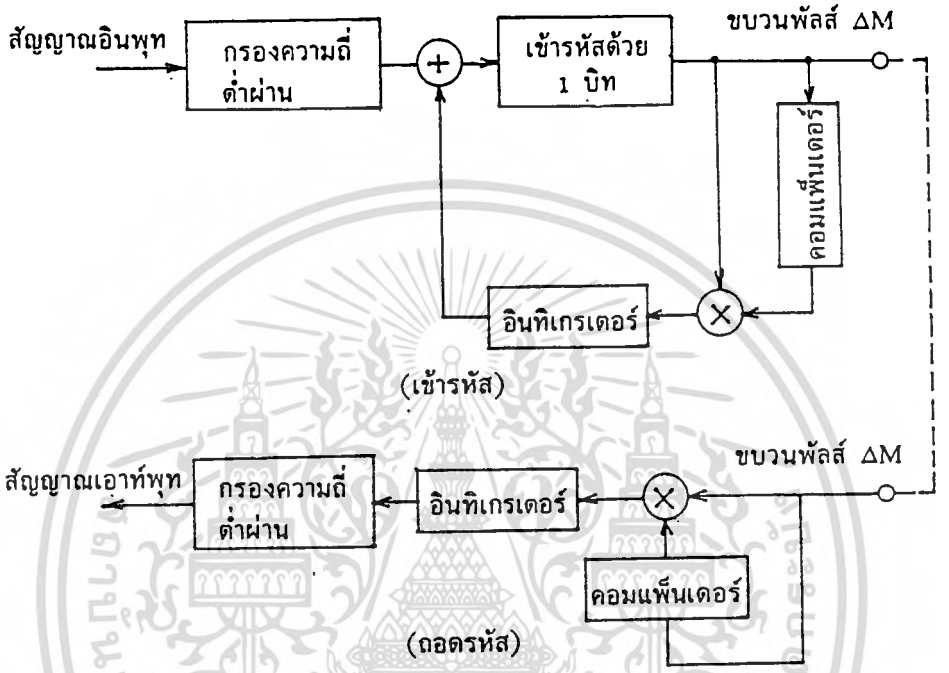
ในขณะที่ความถี่ในการแซมปลิ้งสูงขึ้น ผลต่างระหว่างค่าแซมเปิ้ลข้างเดียวจะน้อยลง ระบบการเข้ารหัสแบบ ADM จะพิจารณาจุดนี้ คือ ใช้ความถี่ในการแซมเปิ้ลให้สูงขึ้น และการเข้ารหัสของผลต่างของสัญญาณ เพื่อส่งออกไปด้วย 1 บิท ตามรูปที่ 2.13 ซึ่งแสดงโครงสร้างของการเข้ารหัส/ถอดรหัสแบบ ADM ประกอบด้วยคอมพาเรเตอร์(Comparator) อินทิเกรเตอร์(Integrator) D/A



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนเวอร์เตอร์(Convertor) และวงจรลอจิกที่จำเป็นบางวงจร เนื่องจากมีขนาดเล็กและการสูญเสียพลังงานน้อย จึงสามารถสร้างได้โดยง่าย

การที่เข้ารหัสโดยใช้เพียง 4 บิท จะกำหนดขนาดของ step จากขบวนพัลส์ที่เข้ามาก่อน (ขบวนพัลส์ ΔM) กล่าวคือ ตามรูปที่ 2.14




รูปที่ 2.13 โครงร่างของระบบ ADM

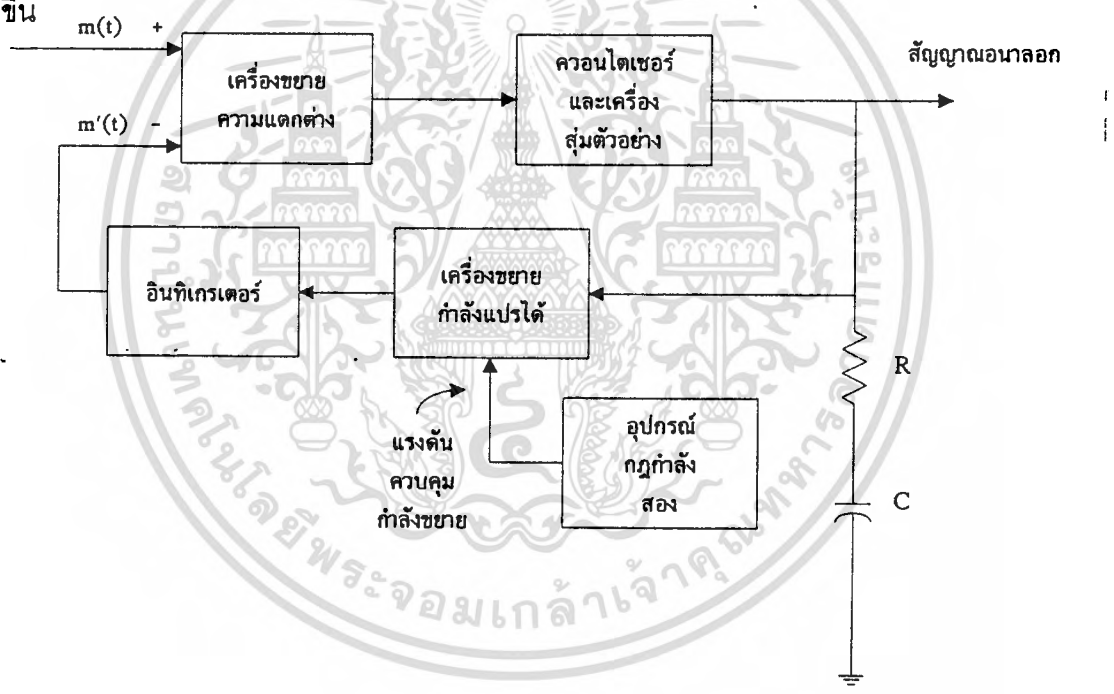
รูปที่ 2.14 หลักการของการเปลี่ยน Stepในการจัดระดับของ ADM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ได้หลังจากผ่านวงจร อินทิเกรเตอร์จะเป็นพัลส์ที่เปลี่ยนแปลงตามสัญญาณอนาลอก(อินพุท) อย่างใกล้ชิด และมีรูปร่างคล้ายสัญญาณอนาลอกมาก ยกเว้นในกรณีที่ สัญญาณอนาลอกเปลี่ยนแปลงขนาดอย่างรวดเร็ว หรือกล่าวได้ว่า ความชันของสัญญาณสูงมาก ๆ ทำให้วงจรส่วน เดลต้ามอดูเลชั่นไม่สามารถสร้างพัลส์ไล่ทันสัญญาณอนาลอก ในกรณีนี้เราเรียกว่า เกิดการเกินภาระของความชัน (Slope Overload ) การแก้ปัญหานี้สามารถทำได้โดย การปรับขนาดของ step แต่จริง ๆ แล้วขนาดของ step ควรจะให้เล็กเมื่อมีการเปลี่ยนแปลงของสัญญาณเกิดขึ้นช้า ๆ และเพิ่มขนาดของ step เพื่อที่จะหลีกเลี่ยง slope overload เมื่อสัญญาณเปลี่ยนแปลงมาก

Block Diagram แบบหนึ่งของระบบ ADM ที่สามารถปรับขนาด step ตามลักษณะของสัญญาณ แสดงได้ดังในรูปที่ 2.15 ขนาดของ step จะเปลี่ยนแปลงได้โดยการควบคุม gain ของอินทิเกรเตอร์ ซึ่ง gain จะมีค่าต่ำเมื่อ voltage เป็น 0 และ gain จะเพิ่มขึ้นเมื่อ voltage ควบคุมเพิ่มมากขึ้น




รูปที่ 2.15 Block Diagram ระบบ ADM

วงจรควบคุม gain ประกอบด้วย RC และ อุปกรณ์ square law เมื่อสัญญาณอินพุทคงที่ หรือเปลี่ยนแปลงอย่างช้า ๆ DM จะตามได้ทัน และเอาท์พุทของ modulator จะเป็นพัลส์ บวก และ ลบ สลับกันตลอดเวลา pluse เหล่านี้เมื่ออินทิเกรตโดย RC แล้วจะได้เอาท์พุทออกมาเฉลี่ยเป็น 0 V จึงต่ำ ดังนั้น step ของ accumulator จึงต่ำด้วย

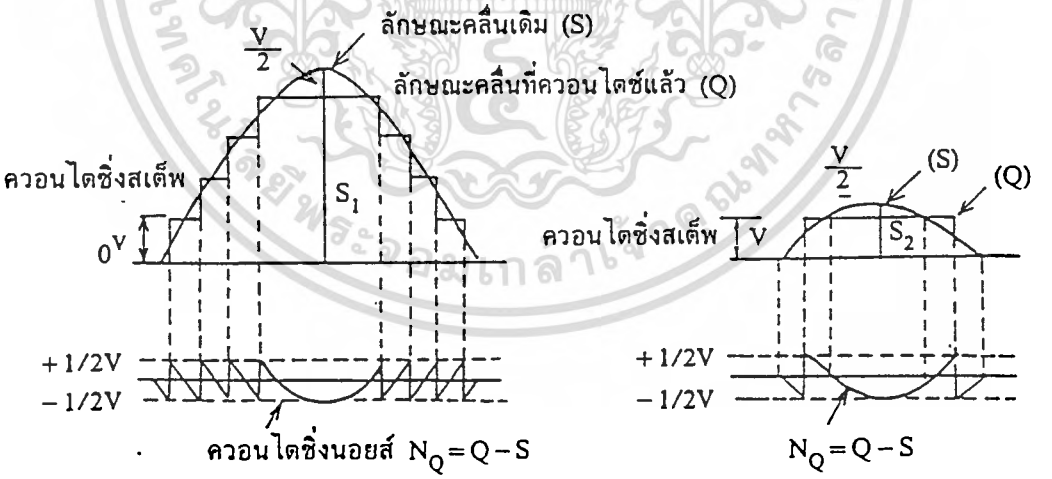


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่จะเกิด slope overload เอาท์พุทของ modulator จะเป็น บวกหรือ ลบตลอด ตอนนี้ที่อินทิเกรเตอร์ จะอินทิเกรตให้เอาท์พุท voltage ออกมาจึงไปเพิ่ม gain ของ amplifier ทำให้มีขนาด step เพิ่มขึ้น จึงทำให้ไม่เกิด slope overload หรือเกิดเพียงเล็กน้อย สำหรับ demodulator ในระบบ ADM จะมีวงจรเหมือน adaptive encoder ทุกประการ

**คอมแพนดิ้ง**(Companding)

ตามที่ได้กล่าวมาแล้วว่า เราไม่สามารถหลีกเลี่ยงควอนไตซิ่งนอยส์ที่เกิดขึ้นได้ แต่จะต้องทำให้ลดลง โดยการลดควอนไตซิ่งอินเทอร์วัล หรือการเพิ่มจำนวนระดับนั่นเอง แต่เมื่อเพิ่มจำนวนระดับขึ้นแล้ว จำนวนบิทที่จะใช้เพิ่มขึ้น จึงจำเป็นต้องใช้ความเร็วในการส่ง สัญญาณดิจิตอลให้สูงขึ้น ตามปกติควอนไตซิ่งนอยส์จะเกิดขึ้นอย่างสม่ำเสมอในทุกอินเทอร์วัล โดยไม่เกี่ยวข้องกับแอมปลิจูดของสัญญาณเดิม หรือ กล่าวอีกในหนึ่งคือ พาวเวอร์ของควอนไตซิ่งนอยส์เกือบจะคงที่โดยไม่ขึ้นอยู่กับสัญญาณ และในการวัดคุณภาพของการเข้ารหัสของสัญญาณจะใช้อัตราส่วนของสัญญาณ S ต่อควอนไตซงนอยส์ $N_Q$ เมื่อเป็นเช่นนี้จะเข้าใจได้ว่า ในกรณีที่ สัญญาณมีระดับสูง $S/N_Q$ จะดีกว่ากรณีของสัญญาณซึ่งมีระดับต่ำ ดังนั้นจึงจำเป็นต้องพิจารณาควอนไตซิ่งนอยส์ในบริเวณที่สัญญาณมีระดับต่ำ อย่างเช่นตามรูปที่ 2.16 กรณีที่เป็นการจัดระดับแบบยูนิฟอร์ม จะเห็นได้ว่า เมื่อสัญญาณมีพาวเวอร์ ต่ำ นอยส์จะมีระดับสูงเมื่อเปรียบเทียบกับระดับของสัญญาณจึงทำให้ $S/N_Q$ เลวลง ด้วย




$$\frac{V/2}{S_1} < \frac{V/2}{S_2}$$

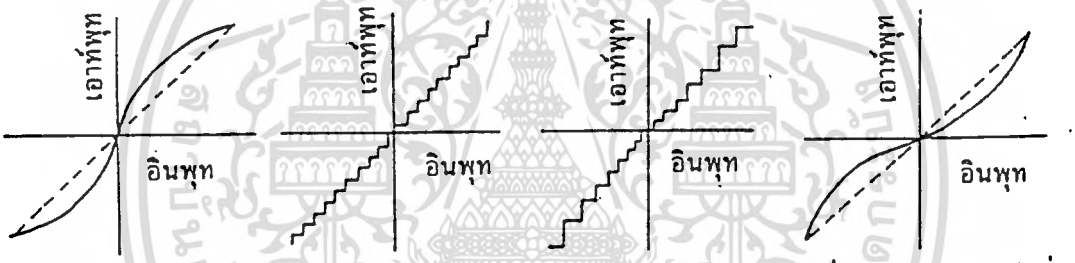
รูปที่ 2.16 การควอนไตซ์และควอนไตซิ่งนอยส์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุนี้จึงใช้การจัดระดับโดยวิธีอื่น กล่าวคือ แบบนอนยูนิฟอร์ม(Non-Uniform Quantizing) คือบริเวณที่สัญญาณมีแอมปลิจูดต่ำจะใช้ควอนไตซิ่งสเต็พแบบแคบๆ และในทางตรงกันข้ามบริเวณที่สัญญาณมีแอมปลิจูดสูงจะใช้ควอนไตซิ่งสเต็พกว้าง ๆ ซึ่งการทำให้เป็นแบบนอนยูนิฟอร์มนั้น จะใช้หลักการของ Companding เข้าช่วย

Companding เป็นชื่อรวมของวงจรคอมเพรสเซอร์ซึ่งใช้สำหรับด้านส่งและวงจรเอ็กซ์แพนเดอร์ซึ่งใช้สำหรับด้านรับ อย่างไรก็ตามตัวอย่างการจัดระดับควอนไตซ์แบบนอนยูนิฟอร์มได้แสดงไว้ในรูป 2.17 ซึ่งมีขั้นตอน คือ ก่อนที่จะทำการจัดระดับจะผ่านสัญญาณไปยังวงจรคอมเพรสเซอร์ ซึ่งมีคุณลักษณะของอินพุท/เอาท์พุท ตามรูปที่ 2.17(ก) แล้วทำการจัดระดับแบบยูนิฟอร์มตามรูป 2.17(ข) ก็จะได้การจัดระดับแบบนอนยูนิฟอร์มตามรูป 2.17(ค) สำหรับทางด้านรับนั้น เมื่อสัญญาณดิจิตอลผ่านขั้นตอนการถอดรหัสแล้ว จะผ่านไปยังวงจรเอ็กซ์แพนเดอร์ ซึ่งมีคุณลักษณะตรงกันข้ามกับคอมเพรสเซอร์ ตามรูป 2.17(ง)




(ก) คอมเพรสชั่น (ข) ยูนิฟอร์มควอนไตซิ่ง (ค) น็อนยูนิฟอร์มควอนไตซิ่ง (ง) เอ็กซ์แพ็นชั่น

รูปที่ 2.17 คุณลักษณะการจัดระดับด้วยคอมเพรสเซอร์และเอ็กซ์เพนเดอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3

## การบันทึกเสียงระบบดิจิตอล

การบันทึกเสียงสำหรับนำมาใช้ใหม่โดยใช้วิธีทางดิจิตอลทำให้ไม่จำเป็นต้องใช้กลไกเคลื่อนไหวใดๆ ดังเช่นแบบเดิม แต่จะใช้อุปกรณ์หน่วยความจำเก็บข้อมูลเสียงแทนเส้นเทปหรือแผ่นเสียง เทคนิคการบันทึกเสียงด้วยระบบดิจิตอลมีด้วยกันหลายวิธี ในที่นี้จะทำการศึกษาและออกแบบระบบดิจิตอลออดิโอที่ ใช้วิธี CVSD เพื่อเป็นแนวทางในการศึกษาและพัฒนาต่อไป

ในการบันทึกเสียงระบบดิจิตอลไม่ว่าจะใช้วิธีการอะไร จะประกอบด้วย 3 ส่วนสำคัญคือ ส่วนแรกทำหน้าที่แปลงสัญญาณเสียงที่เป็นอะนาลอกให้เป็นข้อมูลดิจิตอล แล้วจึงนำไปบันทึกลงในหน่วยความจำซึ่งเป็นหน่วยที่สอง และส่วนสุดท้ายคือส่วนที่ทำหน้าที่แปลงข้อมูลดิจิตอลจากหน่วยความจำออกมาเป็นสัญญาณอะนาลอก ต่อไปนี้เราจะมาศึกษาการทำงานของระบบที่ทำหน้าที่ในการแปลงสัญญาณ

### เดลต้ามอดูเลชั่น

เดลต้ามอดูเลชั่น เป็นเทคนิคของการโมดูเลชั่นอย่างหนึ่งที่สัญญาณอนาลอกสามารถนำมาเข้ารหัสเป็นไบนารีดิจิต หรือบิทโดยตรง ไม่ได้ใช้การสุ่มสัญญาณหนึ่งจุดแล้วแปลงเป็นข้อมูลดิจิตอลหนึ่งเวิร์ด ที่มีความละเอียดเป็นจำนวนบิตที่ต้องการ แต่จะใช้วิธีเปรียบเทียบความสูงหรือการเปลี่ยนแปลงของสัญญาณ เสียงแทน

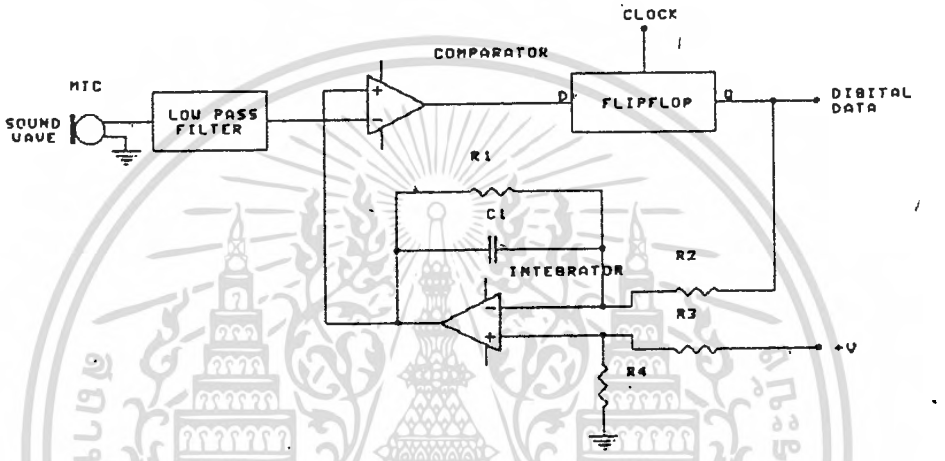
ข้อมูลที่ได้ก็คือทิศทางของการเปลี่ยนแปลง ซึ่งก็มีเพียง ขึ้น หรือ ลง เท่านั้น ดังนั้นความกว้างของข้อมูลดิจิตอลจึงใช้เพียงบิตเดียวก็เพียงพอ ข้อดีของเดลต้ามอดูเลชั่นก็คือใช้หน่วยความจำน้อยกว่าวิธีแบบอื่น ๆ

รูปที่ 3.1 เป็นวงจรเบื้องต้นของเดลต้ามอดูเลชั่นคอมพาราเตอร์ จะทำหน้าที่เปรียบเทียบสัญญาณอินพุทปัจจุบันกับสัญญาณอินพุทก่อนหน้า ซึ่งได้จากการป้อนกลับมายังอินทิเกรเตอร์ เอาท์พุตจากการเปรียบเทียบ ถูกป้อนผ่านฟลิปฟลอปที่ควบคุมด้วยสัญญาณนาฬิกาเพื่อให้ได้เป็น

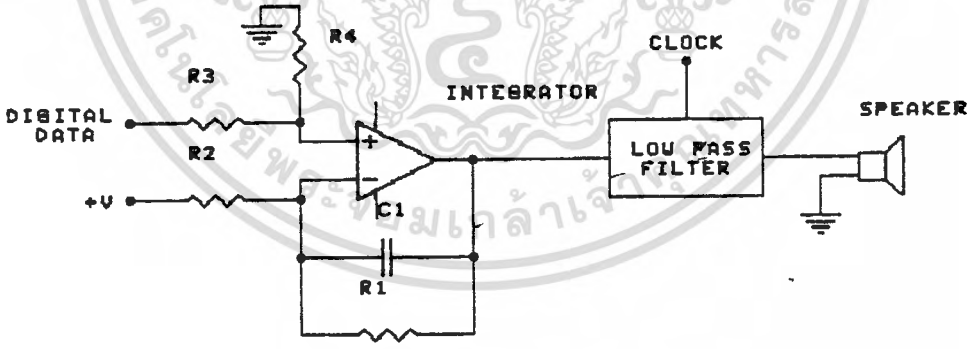


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลดิจิตอล ซึ่งก็คือกำหนดอัตราการสุ่มสัญญาณนั่นเอง สัญญาณที่ได้หลังจากผ่านวงจรอินทิเกรเตอร์จะเป็นพัลส์ที่เปลี่ยนแปลงตามสัญญาณอนาลอก(อินพุท)อย่างใกล้ชิด และมีรูปร่างคล้ายสัญญาณอนาลอกมาก ยกเว้นในกรณีที่สัญญาณอนาลอกเปลี่ยนแปลงขนาดอย่างรวดเร็ว หรือกล่าวได้ว่าความชันของสัญญาณสูงมากๆ ทำให้วงจรส่วนเดลต้ามอดูเลชั่นไม่สามารถสร้างพัลส์ไล่ทันสัญญาณอนาลอก ในกรณีนี้เราเรียกว่า เกิดการเกินภาระของความชัน (Slope Overload) การแก้ปัญหานี้สามารถทำได้ โดยการปรับขนาดของ step แต่จริง ๆ แล้วขนาดของ ควรจะให้เล็กเมื่อมีการเปลี่ยนแปลงของสัญญาณเกิดขึ้นช้า ๆ และเพิ่มขนาดของ step เพื่อที่จะหลีกเลี่ยง slope overload เมื่อสัญญาณเปลี่ยนแปลงมาก




รูปที่ 3.1 วงจรเบื้องต้นของเดลต้ามอดูเลชั่นในส่วนของการแปลงจากสัญญาณเสียงเป็นดิจิตอล



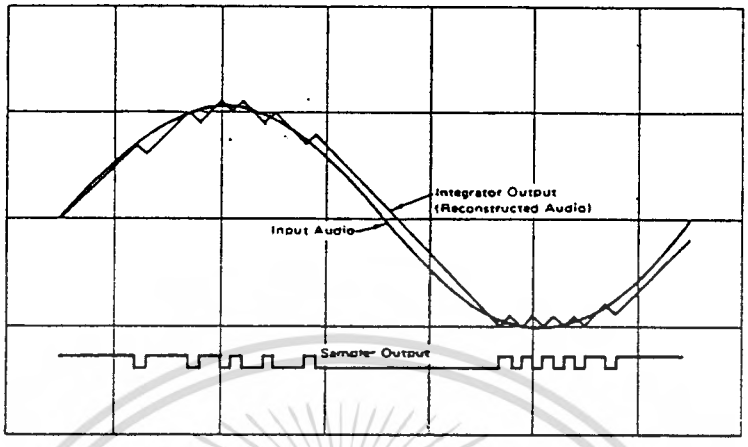

รูปที่ 3.2 วงจรที่ใช้แปลงกลับจากข้อมูลดิจิตอลเป็นสัญญาณเสียง

สัญญาณที่ได้จากตัวเปรียบเทียบและจากอินทิเกรเตอร์เปรียบเทียบกับสัญญาณอินพุตแสดงในรูปที่ 3.3 ลักษณะเช่นนี้จะพบว่า ยิ่งความถี่ของสัญญาณนาฬิกามีค่าสูง ก็ยิ่งสามารถบันทึกการเปลี่ยนแปลงที่แคบได้มากขึ้นทำให้ได้คุณภาพเสียงที่ดีขึ้นแต่ก็สิ้นเปลืองหน่วยความจำมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามไปด้วย ความถี่เท่าใดจึงจะเพียงพอ ทฤษฎีการสุ่มตัวอย่างกล่าวไว้ว่า ไม่น้อยกว่าสองเท่าของความถี่สูงสุดที่นำมาสุ่ม ซึ่งจะเป็นตัวกำหนดอัตราเร็วของข้อมูล (bit rate) ซึ่งที่ 100 kHz ก็เท่ากับ 100000 บิตต่อวินาที




รูปที่ 3.3 เปรียบเทียบสัญญาณอินพุตกับข้อมูลที่ได้สัญญาณอนาลอกจากอินทิเกรเตอร์

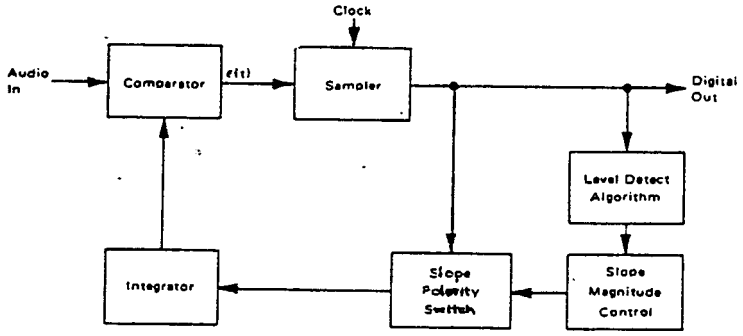
CVSD

ข้อจำกัดของวิธีการเดลต้ามอดูเลชั่นก็คือ แถบกว้างความถี่ใช้งาน ซึ่งถูกจำกัดโดยความถี่สัญญาณนาฬิกาและจะสูงกว่าความถี่สูงสุดของสัญญาณอินพุตมากกว่าสองเท่าขึ้นไป อีกอันหนึ่งคือความเร็วของการเปลี่ยนแปลงความสูงของสัญญาณ หรือ *ไดนามิกเรนจ์* ระบบเดลต้ามอดูเลชั่นธรรมดามีค่าไดนามิกเรนจ์ที่แคบ จำเป็นต้องมีส่วนเพิ่มเติมทำหน้าที่ขยายไดนามิกเรนจ์ให้กว้าง โดยการควบคุมอัตราขยายของ อินทิเกรเตอร์ เพื่อให้ตอบสนองต่อสัญญาณที่มีความชันมาก ๆ ได้ทัน ระบบนี้มีชื่อเรียกใหม่ว่า *ระบบเดลต้ามอดูเลชั่นแบบเปลี่ยนแปลงความชันต่อเนื่อง* หรือ CVSD (continous variable slopedelta modulation)

ระบบ CVSD ทั้งส่วนแปลงจากอะนาลอกเป็นดิจิตอล และส่วนแปลงกลับจากดิจิตอลเป็นอนาลอกแสดงในรูปที่ 3.4 และ 3.5 ตามลำดับ วิธีการของ CVSD ก็คือมีการตรวจระดับสัญญาณโดยอาจใช้วิธีการจัดให้มีรีจิสเตอร์สำหรับเก็บข้อมูลดิจิตอลล่าสุดจำนวน 3 ถึง 4 บิตแล้วตรวจดูว่าเป็น "0" หมด หรือ "1" หมดหรือไม่ถ้าใช่แสดงว่า ขณะนี้อัตราขยายของอินทิเกรเตอร์ต่ำเกินไป ตอบสนองต่อความชันของสัญญาณไม่ทัน ก็จะทำการเพิ่มอัตราขยายให้สูงขึ้นเฉพาะในช่วงนั้น

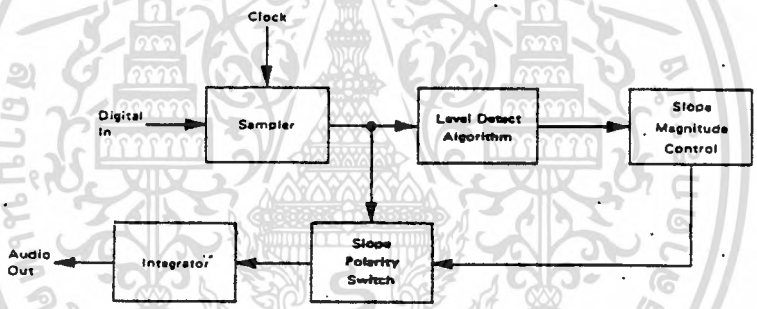


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.4 แผนผังการทำงานของระบบ CVSD ในส่วนของการแปลงสัญญาณเสียงเป็นข้อมูลดิจิตอล

ในส่วนของการแปลงกลับก็จะต้องมีการทำงานในลักษณะเดียวกัน คือมีรีจีสเตอร์ตรวจดูข้อมูลว่าเป็น "0"หมด หรือ "1" หมดหรือไม่ แล้วจัดการควบคุมอัตราขยายของอินทิเกรเตอร์ให้สอดคล้องกัน




รูปที่ 3.5 แผนผังการทำงานของระบบ CVSD ในส่วนแปลงกลับจากดิจิตอลเป็นสัญญาณเสียง

เมื่อรู้จักทฤษฎีการทำงานของการแปลงสัญญาณเสียงโดยใช้เทคนิค CVSD แล้วก็มารู้จักกับไอซีสำเร็จที่ทำหน้าที่ดังกล่าวเบอร์หนึ่ง ซึ่งออกแบบมาสำหรับงานแปลงเสียงพูดและแปลงกลับโดยเฉพาะคือเบอร์ MC3417 และ MC3418 ซึ่งเป็น CVSD มอดูเลเตอร์/ดีมอดดูเลเตอร์ ของโมโตโรล่า ตัวเดียวทำหน้าที่ทั้งแปลงจากสัญญาณอนาลอกเป็นดิจิตอล และแปลงกลับจากดิจิตอลเป็นอนาลอก ออกแบบมาใช้กับระบบสื่อสารสมัยใหม่ที่เป็นระบบดิจิตอล แต่สามารถนำมาดัดแปลงใช้งานในด้านที่ต้องการศึกษานี้ได้พอดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

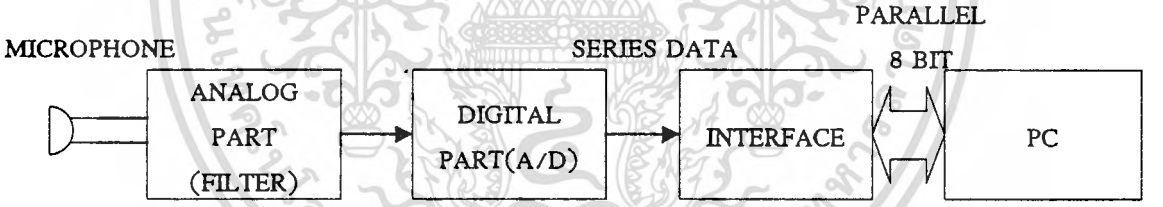
## หลักการทำงาน

การทำงานแบ่งเป็น 2 ส่วนใหญ่ๆ คือ ส่วนฮาร์ดแวร์ และ ส่วนซอฟแวร์

### 4.1 ส่วนฮาร์ดแวร์ (Hardware Part)

หมายถึง ส่วนของวงจรทั้งหมดที่จะออกแบบให้สามารถทำงานได้ตามคำสั่งของซอฟแวร์ ซึ่งการทำงานในส่วนฮาร์ดแวร์ของโครงงานนี้แบ่งเป็น 2 ส่วน คือ การบันทึกเสียงลงในคอมพิวเตอร์ และการนำเสียงที่ถูกบันทึกอยู่ในคอมพิวเตอร์มากระจายเสียงออกทางลำโพง ซึ่งทั้ง 2 ส่วนนี้มีหลักการทำงานที่เหมือนกันดังจะอธิบายต่อไปนี้

4.1.1 การบันทึกเสียงเก็บลงในคอมพิวเตอร์




รูปที่ 4.1

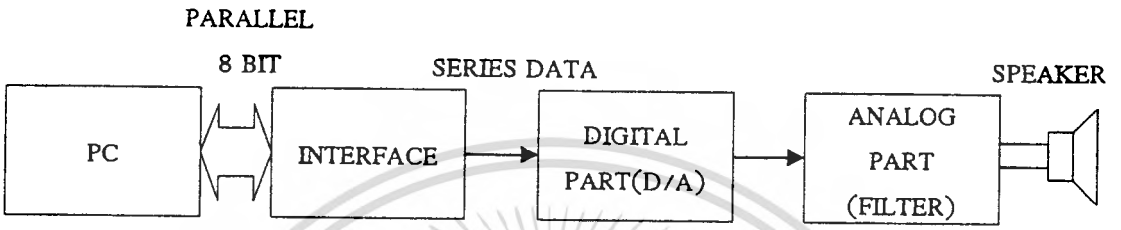
จากรูปที่ 4.1 สัญญาณเสียงจากไมโครโฟนผ่านเข้าวงจรอนาลอก หรือวงจร filter เพื่อกรองสัญญาณที่ไม่ต้องการออกให้เหลือแต่ความถี่ที่ต้องการ แล้วผ่านวงจร A/D converter เพื่อทำการแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอลแบบ Series แล้วนำสัญญาณที่ได้ผ่านวงจร Digital เพื่อทำการแปลงสัญญาณSeriesให้เป็นสัญญาณ Parallel ขนาด 8 bit และวงจรในส่วนนี้จะสร้าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ Digital บางสัญญาณที่จำเป็นต้องใช้ในการทำงานของวงจร สัญญาณ Parallel ที่ได้จะถูกส่งผ่านวงจร Interface เพื่อที่จะนำเข้าไปในคอมพิวเตอร์แล้วเก็บไว้เป็น file ใน Harddisk

4.1.2 การกระจายเสียงออกทางลำโพง




รูปที่ 4.2

จากรูปที่ 4.2 การทำงานในส่วนนี้มีหลักการเหมือนกับส่วนการบันทึกเสียง เพียงแต่การทำงานในส่วนนี้จะเริ่มจากการนำข้อมูลที่เก็บอยู่ใน file ซึ่งอยู่ใน Harddiskออกมาผ่านส่วน Interface แล้วนำมาผ่านวงจรDigital,วงจร D/A converter และวงจร Analog (filter) เพื่อกรองเฉพาะสัญญาณเสียงออกทางลำโพง

4.2 ส่วนซอฟแวร์ (Software)

เป็นส่วนของโปรแกรมที่ใช้ควบคุมการทำงานของวงจร สามารถแบ่งออกเป็น

1. โปรแกรมควบคุมการจัดการข้อมูลเพื่อส่งให้ส่วนกระจายเสียง

มีข้อกำหนดเบื้องต้นสำหรับโปรแกรมในส่วนนี้ดังนี้

-สามารถรับการขอติดต่อ(Connection) จาก channel ต่าง ๆ ได้ตลอดเวลา

-สามารถทำการส่งข้อมูลให้กับ channel ที่ติดต่ออยู่ได้อย่างต่อเนื่อง พร้อมกันทุก channel

2. โปรแกรมสนับสนุนการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วย

-โปรแกรมบันทึกสัญญาณเสียง

-โปรแกรมบันทึกและแก้ไขข้อมูล

ซึ่งจะกล่าวถึงรายละเอียดในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

## รายละเอียดการออกแบบและการทำงาน

### 5.1 ฮาร์ดแวร์

วงจรทั้งหมดจะใช้เครื่อง IBM PC เป็นส่วนควบคุมหลัก โดยได้รับการสนับสนุนจากส่วนซอฟแวร์ จาก บทที่ 4 จะเห็นว่า การทำงานของส่วนบันทึกเสียงและส่วนกระจายเสียงมีหลักการที่เหมือนกัน ทั้งสองวงจรจะแบ่งการทำงานออกเป็น 3 ส่วน

5.1.1 การบันทึกเสียง แบ่งการทำงานออกเป็น 3 ส่วนคือ

ก) วงจร Analog และ วงจร A/D converter

วงจรส่วนนี้จะทำหน้าที่ ในการกรองสัญญาณเสียงที่เข้ามาทางลำโพงให้ได้เฉพาะความถี่ที่ต้องการ(ช่วงความถี่เสียงคน คือ 300-3k Hz ) และแปลงสัญญาณที่กรองได้นั้น(สัญญาณ Analog )ให้เป็นสัญญาณ Digitalโดยในส่วนนี้วงจรจะทำการแปลงสัญญาณเสียงโดยใช้วิธีของเดลต้ามอดูเลชั่น ซึ่งเป็นวิธีการแปลงสัญญาณ Analog ให้เป็นสัญญาณ Digital แบบ Series ดังที่ได้กล่าวมาแล้ว

หัวใจสำคัญของวงจรในส่วนนี้คือ ไอซีเบอร์ MC3418 (Contineously Variable Slope DeltaModulator/Demodulator) ของโมโตโรล่า (Motorola) ซึ่งจะทำหน้าที่มอดดูเลทสัญญาณเสียงให้เป็น สัญญาณดิจิตอลแบบอนุกรม และดีมอดดูเลทสัญญาณดิจิตอลแบบอนุกรมเป็นสัญญาณเสียง ซึ่งสามารถทำได้ทั้ง 2 ฟังก์ชันในตัวเดียวกัน โดยการควบคุมที่ขา 15 (Encode/Decode) ซึ่งลักษณะการต่อของวงจรจะเป็นดังรูปที่ 5.2

รายละเอียดของไอซี MC3418 เป็นดังนี้

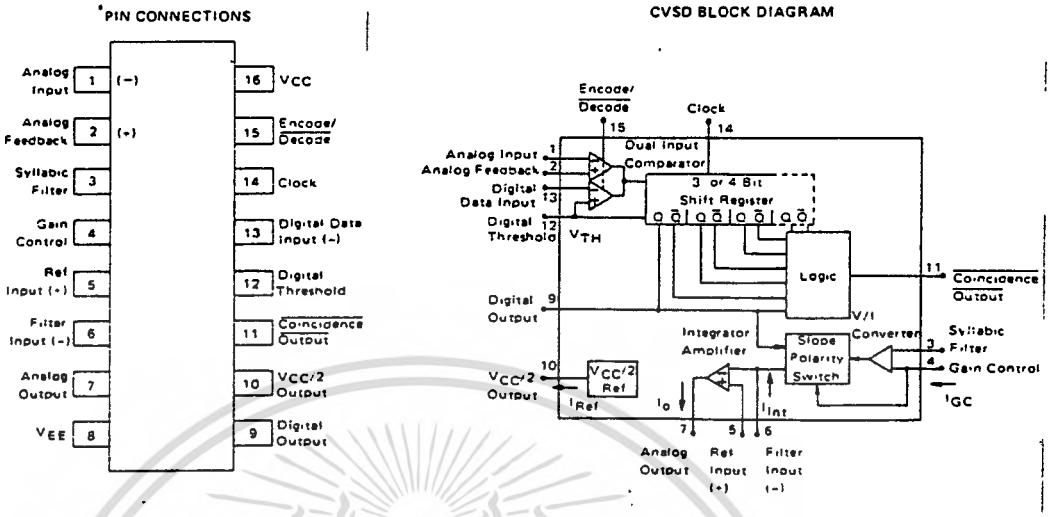
- เป็นไอซีที่สร้างขึ้นเพื่อใช้มอดดูเลทสัญญาณในช่วงความถี่เสียง
- มีระบบมอดดูเลชั่นและดีมอดดูเลชั่นในตัวเดียวกัน
- สามารถใช้งานร่วมกับไอซีประเภทซีมอส-(CMOS) และทีทีแอล ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การจ่ายไฟเลี้ยงสามารถกระทำได้ 2 แบบ คือ

1. ไฟเลี้ยงคู่ (Dual Supply) คือ ใช้ไฟบวกร่วมกับไฟลบ
2. ไฟเลี้ยงเดี่ยว (Single Supply) คือ ใช้ไฟบวกกับกราวนด์




(ก) การวางขา (ข) แผนผังภายใน

รูปที่ 5.1 รายละเอียดของไอซี MC3418

จากรูปข้างต้นสามารถอธิบายถึงรายละเอียดและหน้าที่ของสัญญาณในแต่ละขาได้ดังนี้

ขา 1 : Analog Input: เป็นขาที่ใช้รับสัญญาณเสียง ซึ่งจากแผนผังภายในจะเห็นได้ว่าเป็นขาอินเวอร์ทติ้งของวงจรเปรียบเทียบแรงดัน(Voltage Comparator)

ขา 2 : Analog Feedback เป็นขานอนอินเวอร์ทติ้งของวงจรเปรียบเทียบแรงดัน ซึ่งทำหน้าที่ในการนำสัญญาณกลับมาเปรียบเทียบกับสัญญาณอินพุต ที่จะเข้ามาในช่วงนาฬิกาถัดไป

ขา 3 : Syllabic Filter เป็นขาอินพุตที่รับสัญญาณจากวงจรกรองสัญญาณภายนอก เพื่อนำมาควบคุมขนาดของสัญญาณในแต่ละขั้นของส่วนที่เป็นอินทิเกรเตอร์

ขา 4 : Gain Control Input เป็นขาอินพุต ที่รับสัญญาณ เพื่อควบคุมอัตราการขยายของออปแอมป์ภายในตัวไอซี

ขา 5 : Reference Input เป็นขานอนอินเวอร์ทติ้งของออปแอมป์เพื่อใช้เป็นแรงดันอ้างอิงทางดีซี ของสัญญาณเอาท์พุต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 6 : Filter Input เป็นขานอนอินเวอร์ทติ้งของออปแอมป์ เพื่อใช้ต่อกับวงจรภายนอก สำหรับสร้างเป็นวงจรอินทิเกรเตอร์

ขา 7 : Analog Outputเป็นขาเอาท์พุตของวงจรอินทิเกรเตอร์ที่สร้างขึ้น ซึ่งจะใช้เป็นสัญญาณเอาท์พุต ในขณะที่ไอซีตัวนี้ทำงานเป็นวงจรถอดรหัส และ เป็นสัญญาณป้อนกลับเข้าขา 2 ในกรณีที่ทำงานเป็นวงจรเป็นวงจรเข้ารหัส

ขา 8 : $V_{EE}$ เป็นขาที่ใช้สำหรับต่อกับไฟเลี้ยงที่มีค่าต่ำสุดที่จ่ายให้กับไอซีตัวนี้

ขา 9 : Digital Outputเป็นขาเอาท์พุตที่ได้มาจากการมอดดูเลทสัญญาณอนาลอกซึ่งสัญญาณนี้ จะมีการแกว่ง (Swing) อยู่ระหว่าง $V_{CC}$ กับ $V_{EE}$

ขา 10 : $V_{CC}/2$ Outputเป็นขาเอาท์พุตที่ถูกสร้างขึ้น เพื่อให้เกิดแรงดันที่มีค่าครึ่งหนึ่งของแรงดัน $V_{CC}$ ซึ่งจะถูกนำไปใช้งานในกรณีของการจ่ายไฟเลี้ยงเดี่ยว

ขา 11 : Coincidence Output เป็นขาเอาท์พุตที่แสดงว่าชิฟรีจิสเตอร์ (Shift Register) ภายในถูกเก็บด้วย 0 หรือ 1 ทั้งหมด ซึ่งสัญญาณนี้จะถูกนำไปใช้ ในการกำหนดอัตราการขยายสัญญาณของวงจรสร้างระดับสัญญาณ

ขา 12 : Digital Threshold เป็นขาอินพุตเพื่อใช้เลือกว่าจะใช้งานกับไอซีประเภท ใดๆ

ขา 13 : Digital DataInputเป็นขาอินพุต เพื่อรับสัญญาณดิจิตอลแบบอนุกรมจากภายนอกใน การแปลงเป็นสัญญาณอะนาลอก ใช้ความถี่ 16 kHz

ขา 14 : Clock Input เป็นขาอินพุตที่รับสัญญาณนาฬิกา เพื่อใช้กำหนดจังหวะการทำงาน ภายในตัวไอซี ซึ่งความถี่ของสัญญาณนาฬิกานี้จะเป็นความถี่ที่ใช้สำหรับการแซมปลิ้งสัญญาณอะนาลอกที่ต้องการมอดดูเลท

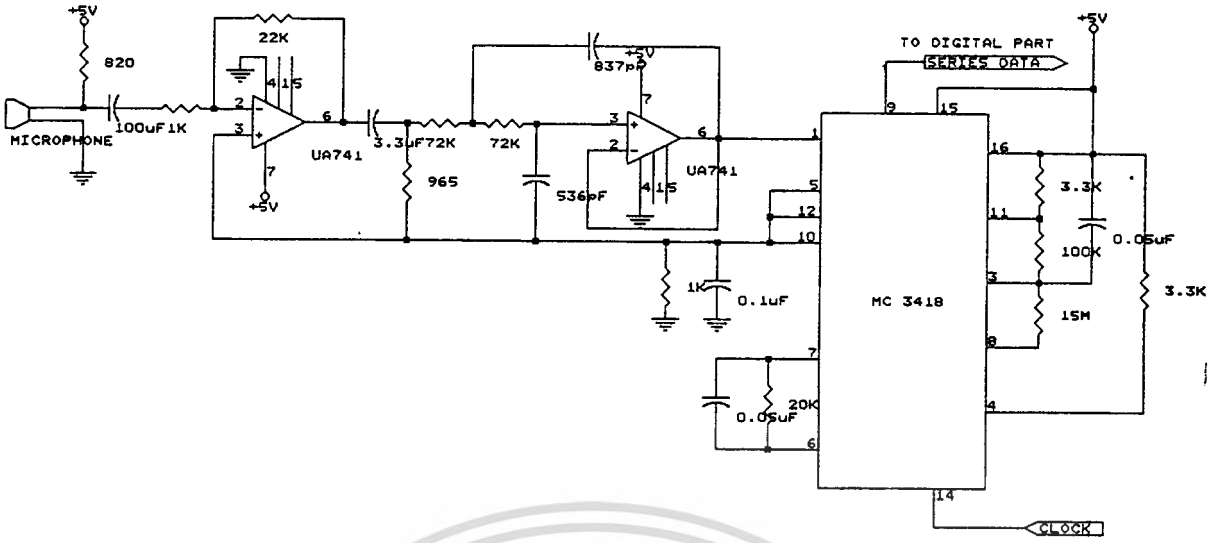
ขา 15 : Encode/Decode เป็นขาที่ใช้ในการควบคุมการทำงานของไอซี ถ้าเป็น 1ก็จะทำงานเป็นตัวเข้ารหัส ถ้าเป็น 0 ก็จะทำงานเป็นตัวถอดรหัส

ขา 16 : $V_{CC}$ เป็นขาที่รับแรงดันไฟเลี้ยงค่าสูงสุดของไอซี ซึ่งจะมีค่าอยู่ระหว่าง 4.75 ถึง 16.5 โวลท์

สำหรับวงจรมอดดูเลเตอร์เพื่อนำไปใช้งาน เป็นดังรูปที่ 5.2



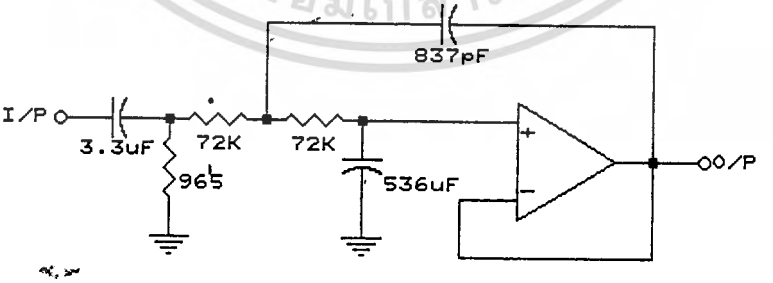
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 5.2 วงจรส่วน เดลต้า มอดดูเลเตอร์

จากรูปข้างต้น สามารถอธิบายการทำงานได้ดังนี้ คือ

ใช้สำหรับการแปลงสัญญาณเสียงเป็นสัญญาณดิจิตอลซึ่งขั้นตอนการทำงานจะเป็น สัญญาณเสียงจะถูกส่งผ่านทางไมโครโฟนเข้าสู่วงจรปรีไมค์ ซึ่งใช้ออปแอมป์ เบอร์uA741 ต่อในวงจรขยายสัญญาณแบบกลับเฟส (Inverting Amplifier) ซึ่งมีอัตราการขยายแรงดัน(Voltage Gain)ประมาณ 22 เท่า เพื่อทำการขยายสัญญาณ หลังจากนั้นสัญญาณที่ได้จะถูกนำมากรองเพื่อกำจัดสัญญาณที่ไม่ต้องการออกโดยใช้วงจรกรองความถี่แบนด์พาสแบบแอคทีฟ (Active Band Pass Filter) ซึ่งมีความถี่คัทออฟที่ 50 Hz (ซึ่งเป็นความถี่ของสัญญาณไฟบ้าน) และ 3.3 KHz (ซึ่งเป็นความถี่สูงสุดที่ในการส่งสัญญาณเสียงทางคู่ สายโทรศัพท์)ซึ่งวงจรกรองความถี่ดังกล่าวเป็นดังรูปที่ 5.3




รูปที่ 5.3 วงจรกรองความถี่สัญญาณทางภาคอินพุท



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรกรองความถี่สัญญาณ ดังรูป 5.3 เราจะสามารถแยกการพิจารณาวงจรออกได้เป็น 2 ส่วน (วงจรสองส่วนนี้มีการต่อกันแบบคาสเคด (Cascade) ) ดังนี้

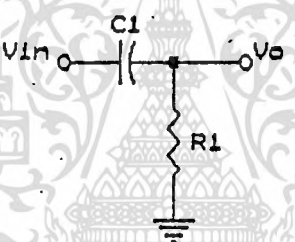
1) วงจรส่วนไฮพาส

เป็นวงจรกรองความถี่แบบพาสซิฟอันดับที่1 ประกอบด้วย ตัวเก็บประจุไฟฟ้า $C_1$และตัวต้านทาน $R_1$ ซึ่งต่อขนานอยู่กับค่าความต้านทานทางด้านอินพุทของวงจรในส่วนที่ 2 ซึ่งมีค่ามากกว่า $R_1$ มากจึงไม่ นำมาคิด ดังนั้นจะได้ว่า

สมการโนด (Node Equation)

$$V_o(sC_1+(1/R_1)) - V_{in}(sC_1) = 0$$

ทรานเฟอร์ฟังก์ชั่น $V_o = sC_1 = s$ ---- (5.1)




รูปที่ 5.4 วงจรกรองความถี่ไฮพาสที่ใช้

$$V_{in} = (sC_1+(1/R_1))/ C_1 = s+(1/C_1R_1)$$ ---- (5.2)

ซึ่งจะได้ความถี่คัตออฟที่ $f_H = 1/2\prod(\sqrt{C_1R_1})$ ---- (5.3)

แทนค่า $= 1/2\prod(\sqrt{3.3x10^{-6}})(965)$

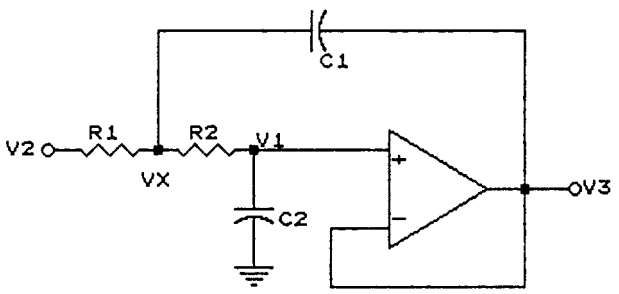
$= 49.98 = 50$ Hz

2) วงจรส่วนโลว์พาส

เป็นวงจรกรองความถี่สัญญาณแบบอันดับที่ 2 ประกอบด้วยออปแอมป์ และวงจร RC ดังรูปที่ 5.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 5.5 วงจรกรองความถี่โลว์พาสที่ใช้

พิจารณาโดยใช้การวิเคราะห์แบบโนด (Node Analysis) จะได้สมการดังนี้

$$\begin{bmatrix} \frac{1}{R_1}+\frac{1+sC_1}{R_2} & \frac{-1}{R_2} \\ \frac{1}{R_2} & -\frac{(1+sC)}{R_2} \end{bmatrix} \begin{bmatrix} V_x \\ V_1 \end{bmatrix} = \begin{bmatrix} sC_1 & \frac{1}{R_1} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_3 \\ V_2 \end{bmatrix}$$

จะได้ทรานเฟอร์ฟังก์ชั่นฟีดแบค (Feedback Transfer Function)

$$T_{FB} = \left.\frac{V_1}{V_3}\right|_{V_2=0} = \frac{s/R_2C_2}{s^2+s(1/R_1C_1)+(1/R_2C_1)+(1/R_2C_2)+(1/R_1R_2C_1C_2)} \quad \text{---- (5.4)}$$

ทรานเฟอร์ฟังก์ชั่นฟีดฟอร์เวอร์ด (Feedforward Transfer Function)

$$T_{FF} = \left.\frac{V_1}{V_2}\right|_{V_3=0} = \frac{1/R_1R_2C_1C_2}{s^2+s(1/R_1C_1)+(1/R_2C_1)+(1/R_2C_2)+(1/R_1R_2C_1C_2)} \quad \text{---- (5.5)}$$

ซึ่งจะได้ทรานเฟอร์ฟังก์ชั่นของวงจรดังสมการ

$$T_V = \frac{kN_{FF}}{D-kN_{FB}} \quad \text{---- (5.6)}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $N_{FB}$,$N_{FF}$ คือ เศษ (Numerator) ของ $T_{FB}$ ,$T_{FF}$ ตามลำดับ

D คือ ส่วน (Denomenator) ของทั้ง $T_{FB}$ และ $T_{FF}$

k คือ อัตราการขยายมีค่าประมาณ $1+(r_2/r_1)$ ในที่นี้มีค่าเท่ากับ 1

ดังนั้น จะได้ว่า

$$T_V = \frac{k/R_1R_2C_1C_2}{s^2+s(1/R_1C_1)+(1/R_2C_1)+((1-k)/R_2C_2)+(1/R_1R_2C_1C_2)} \quad \text{---- (5.7)}$$

จากฟังก์ชั่นของอัตราการขยายแรงดันอันดับ 2 ของวงจรกรองความถี่แบบโลว์พาส

$$\frac{V_o}{V_{in}} = \frac{w_p^2}{s^2+(w_p/Q_p)s+w_p^2} \quad \text{---- (5.8)}$$

ซึ่งความถี่คัตออฟมีค่าเท่ากับความถี่โพล

$$w_L = w_p = 1/\sqrt{R_1R_2C_1C_2} \quad \text{---- (5.9)}$$

$$f_L = 1/2\Pi(\sqrt{R_1R_2C_1C_2}) \quad \text{---- (5.10)}$$

แทนค่า $= 1/2\Pi\sqrt{(72x10^3)(72x10^3)(837x10^{-12})(536x10^{-12})}$

$= 3.3$ kHz

หลังจากสัญญาณเสียงถูกกรองความถี่เรียบร้อยแล้วก็จะถูกมอดดูเลทโดยไอซีเบอร์ MC3418 โดยผ่านเข้าทางขาที่ 1(Analog Input) ซึ่งขาที่ 15 จะต้องถูกควบคุมให้มีสถานะเป็น'1' (Encode) จะทำให้ได้สัญญาณเอาท์พุท(Output)ในลักษณะของสัญญาณดิจิตอลแบบอนุกรม ออก ทางขาที่ 9 (Digital Output)

ข) <u>วงจร Digital</u>

วงจรในส่วนนี้มีหน้าที่ในการแปลงสัญญาณ Digital แบบ Series ให้เป็นสัญญาณแบบ Parallel และสร้างสัญญาณต่างๆที่จำเป็นในการทำงานของวงจร

จากหน้าที่ดังกล่าวข้างต้น สามารถออกแบบวงจรได้ดังรูปที่ 5.6

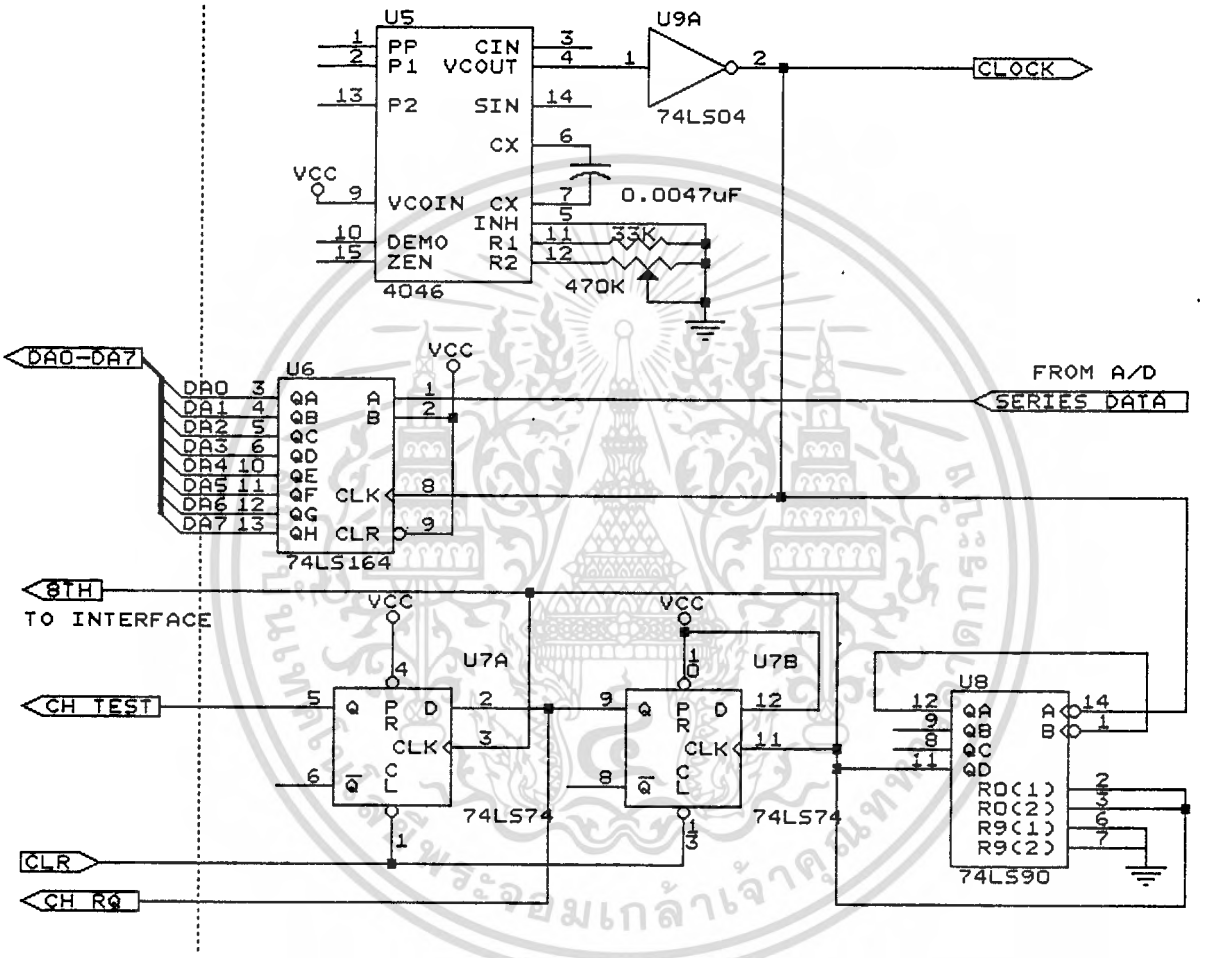
จากรูปที่ 5.6 วงจรส่วนนี้ประกอบด้วยอุปกรณ์ต่างๆ ซึ่งมีหน้าที่การทำงานดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 74LS164 เป็นอุปกรณ์ในการแปลงสัญญาณ Series ให้เป็นสัญญาณ Parallel ซึ่งมีการทำงานดังนี้

ข้อมูลสัญญาณ Series จากวงจร Analog จะเข้าทางขา A ของ 74LS164 ซึ่ง 74LS164 นี้จะทำงานตามจังหวะ สัญญาณ clock โดยจะ shift ข้อมูลเข้าทีละ bit โดยจะได้ output เป็นข้อมูล Parallel ขนาด 8 bit เมื่อสัญญาณ clock นับครั้งที่ 8




รูปที่ 5.6 วงจรส่วน Digital

- MC14046 เป็นอุปกรณ์ที่ใช้สร้างสัญญาณ clock
- 74LS04 เป็น Inverter ซึ่งในวงจรนี้ใช้เพื่อขับกระแส
- 74LS90 เป็น Decade counter ซึ่งนำมาต่อเป็นวงจรนับ 8
- 74LS74 เป็น D-Flip Flop ซึ่งใช้เป็น registor ขนาด 1 bit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจร สามารถอธิบายการทำงานได้ดังนี้

MC14046 สร้าง clock ความถี่ 16KHz เพื่อเป็นฐานเวลาให้กับส่วนต่างๆ ของวงจร โดยต่อเข้ากับ 74LS04 เพื่อช่วยขับกระแส 74LS164 จะรับข้อมูล Series โดยทำงานตามจังหวะสัญญาณ clock เมื่อได้ข้อมูล output เป็นข้อมูล Parallel ครบ 8 bit แล้วจะส่งข้อมูลให้วงจรส่วน Interface PC โดยมี74LS90 ทำการนับตามจังหวะสัญญาณ clock เมื่อนับถึง 8 (แสดงว่าข้อมูลครบ 8 bit แล้ว สัญญาณที่ขา $Q_D$จะเป็น "1" ทำให้ 74LS74 ซึ่งทำหน้าที่เป็น register ได้รับสัญญาณที่ขา clock ทำให้ output ของ74LS74 เป็น "1" (สัญญาณ 8TH เป็น '1'เพื่อแสดงให้ส่วน Interface ทราบว่า ข้อมูลครบ 8 bitแล้ว ส่วน Interface จะทำการรับข้อมูลพร้อมทั้งทำการ clear 74LS74 เพื่อให้พร้อมที่จะรับข้อมูลชุดต่อไป แต่ถ้าข้อมูล 8 บิท ชุดที่ 2 เข้ามาขณะที่ข้อมูลชุดแรกยังไม่ถูกส่วน Interface รับไป (ข้อมูลชุดแรกจะหายไป หรือถูกข้อมูลชุดที่ 2 ทับนั่นเอง) สัญญาณ Test จะเปลี่ยนเป็น '1'เพื่อบอกให้ส่วน Interface รู้ว่า เกิด over run ขึ้น

ค) <u>ส่วน Interface</u>

จากวงจรทั้งสองส่วนที่ออกแบบไว้แล้วจะติดต่อกับ PC โดยผ่านพอร์ท 3 พอร์ท ดังนี้

1. พอร์ทอินพุทสัญญาณควบคุม (Control Signal Input Port) ใช้เป็นพอร์ทสำหรับรับสัญญาณ 'DATA RQ' และ สัญญาณ 'DATA TEST'

2.พอร์ทอินพุทข้อมูล (Data Input Port) ใช้เป็นพอร์ทสำหรับรับข้อมูลจากส่วน Digitalให้ PC

จากการกำหนดข้างต้น จะต้องทำการเลือกช่วงของพอร์ทแอดเดรสที่ว่างจากการใช้งานของ IBMPC ตามมาตราฐานจากหนังสือ IBMPC Techincal Reference เมื่อพิจารณาพอร์ทแอดเดรสแมพ (Port Address Map) สำหรับ IBM PC แล้วจึงทำการเลือกพอร์ทแอดเดรสในช่วง 283h-284h ซึ่งเป็นช่วงที่ว่างจากการใช้งานของ IBM PC

ดังนั้นจึงสามารถกำหนดพอร์ทแอดเดรสสำหรับพอร์ททั้ง 2 ได้ดังนี้

พอร์ท 283h - พอร์ทอินพุทสัญญาณควบคุม

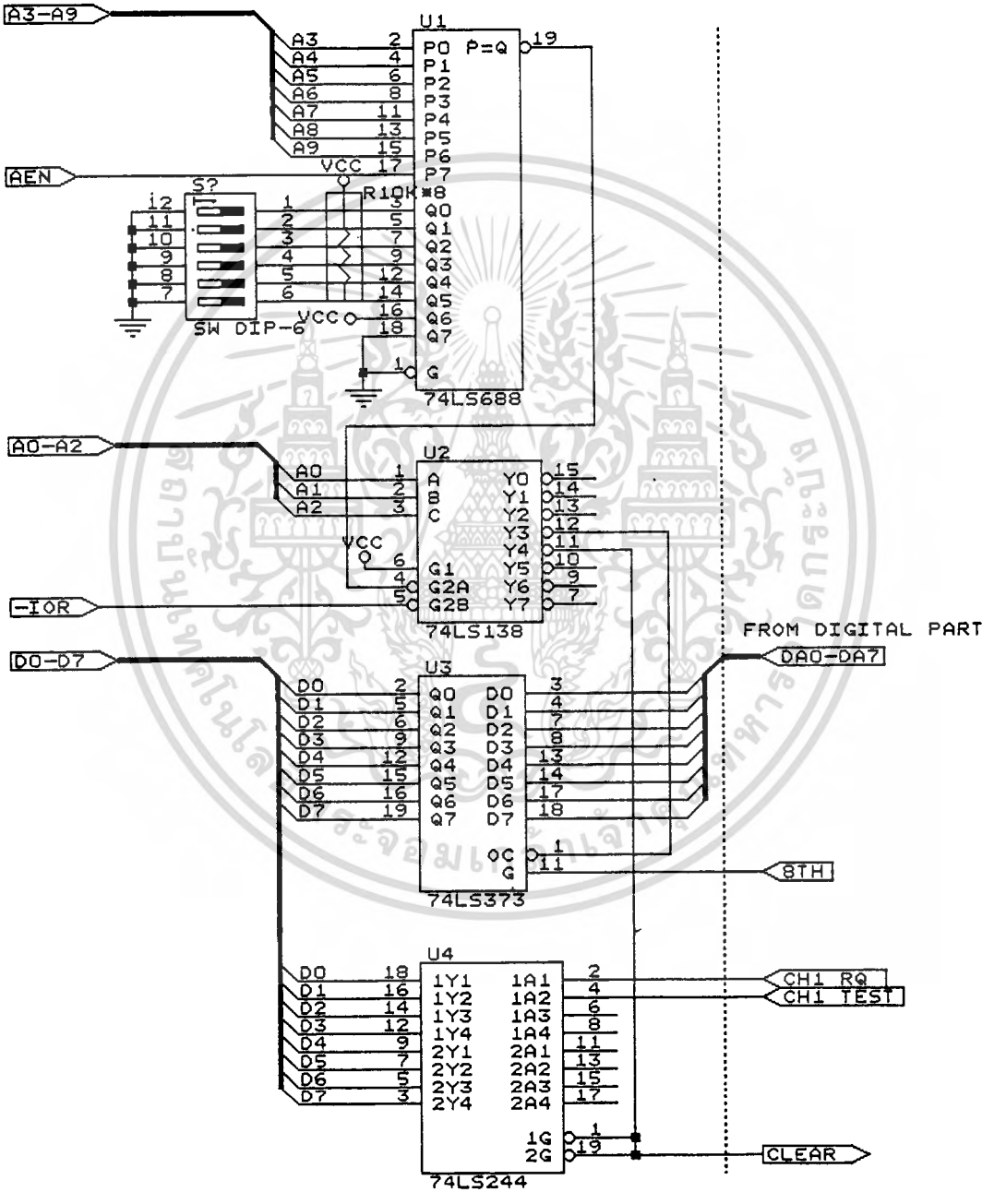
พอร์ท 284h - พอร์ทอินพุทข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยวงจรในส่วนนี้สามารถออกแบบได้ดังรูปที่ 5.7

จากรูปบัสแอดเดรสของเครื่อง pc จำนวน 10 เส้นและสัญญาณ AEN จะถูกนำมาดีโค้ทแอดเดรสร่วมกันด้วย 74LS688 และ 74LS138 ให้พอร์ท 283h และพอร์ท 284h เป็นพอร์ทอินพุท โดยใช้สัญญาณ IORในการควบคุมการอินพอร์ท พอร์ท 283h ใช้ 74LS373 เป็นอุปกรณ์ I/O ในการ inport data ส่วนพอร์ท 284h ใช้ 74LS244 เป็นอุปกรณ์I/O

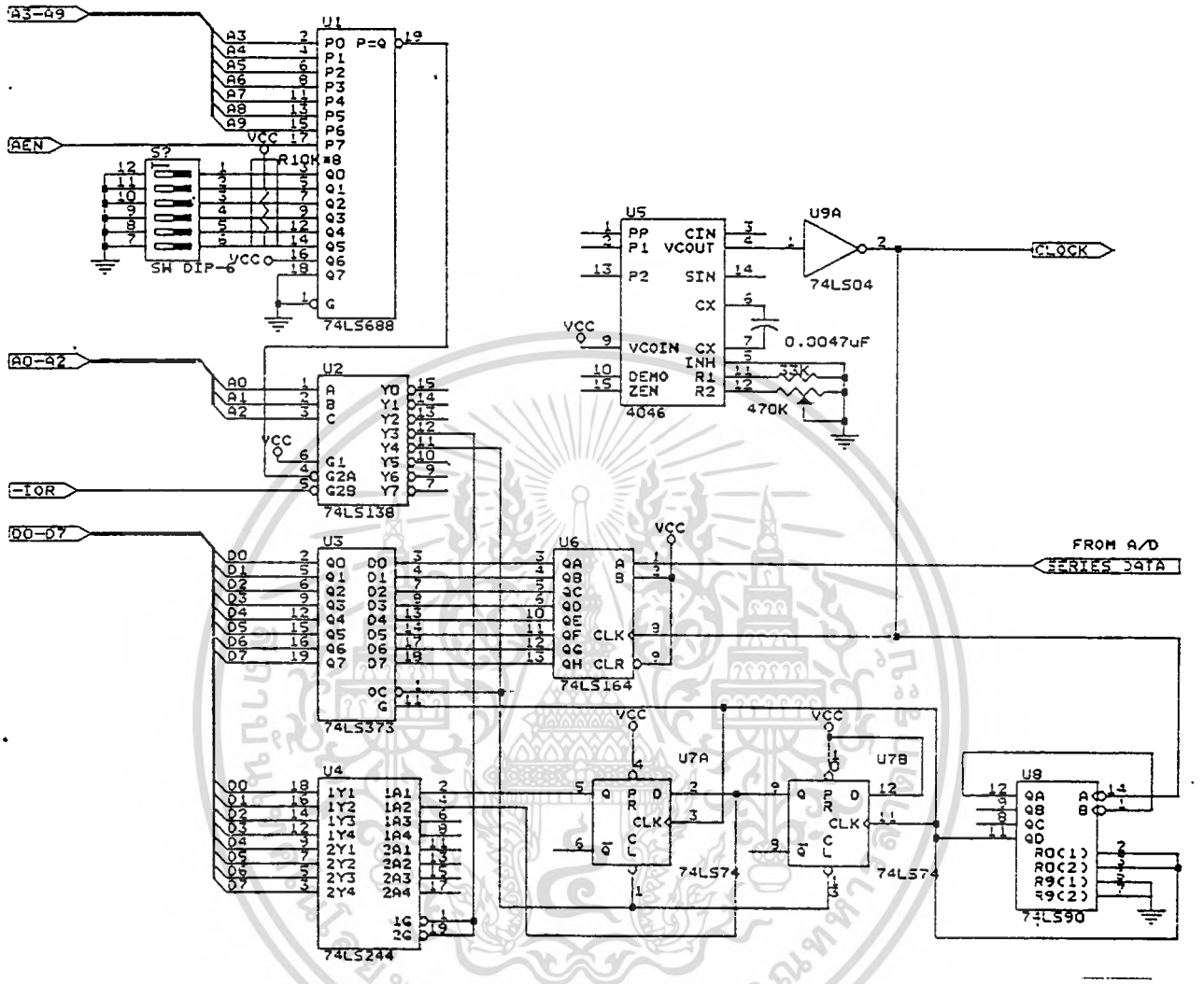



รูปที่ 5.7 วงจร Interface ในส่วนการบันทึกเสียง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรทั้งสามส่วนสามารถรวมเป็นวงจรทั้งหมดได้ดังรูปที่ 5.8




รูปที่ 5.8 วงจรบันทึกเสียง

5.1.2 <u>การกระจายเสียง</u> แบ่งการทำงานออกเป็น 3 ส่วน คือ

ก) <u>วงจร Analog และวงจร D/A converter</u>

วงจรส่วนนี้จะทำหน้าที่ในการแปลงสัญญาณ Digital แบบ Series ให้เป็นสัญญาณ Analog

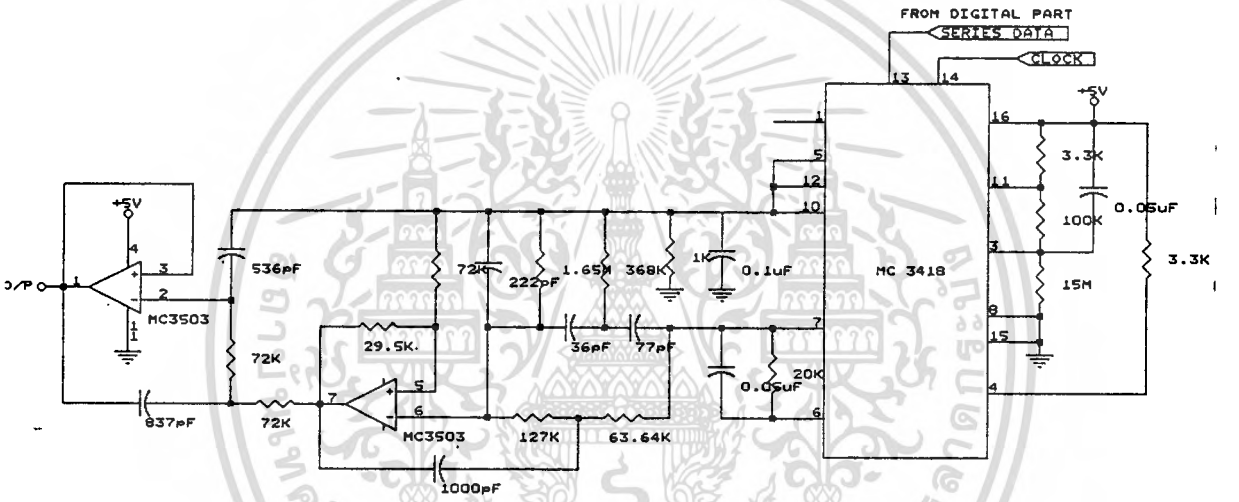


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

( ซึ่งสัญญาณที่ได้จะมีลักษณะเป็นขั้นสามเหลี่ยม ) แล้วผ่านวงจร filter เพื่อกรองความถี่ที่ไม่ต้องการออกให้ได้สัญญาณoutput ที่มีลักษณะเรียบ แล้วนำสัญญาณที่ได้ออกลำโพง โดยในส่วนนี้วงจรจะทำการแปลงสัญญาณ Digitalแบบ Series โดยใช้วิธีของเดลต้า ดีมอดูเลชั่น ซึ่งจะแปลงเป็นสัญญาณ Analog

วงจรส่วนนี้จะใช้ไอซี MC3418 ในการแปลงสัญญาณ Digital แบบ Series ไปเป็นสัญญาณ Analogโดยใช้ไอซี MC3418 (เช่นเดียวกับการบันทึกเสียง) โดยใช้วิธีดีมอดูเลทด้วยวิธีของเดลต้ามอดูเลชั่น

สำหรับวงจรดีมอดูเลเตอร์ที่จะนำไปใช้งาน เป็นดังรูปที่ 5.9



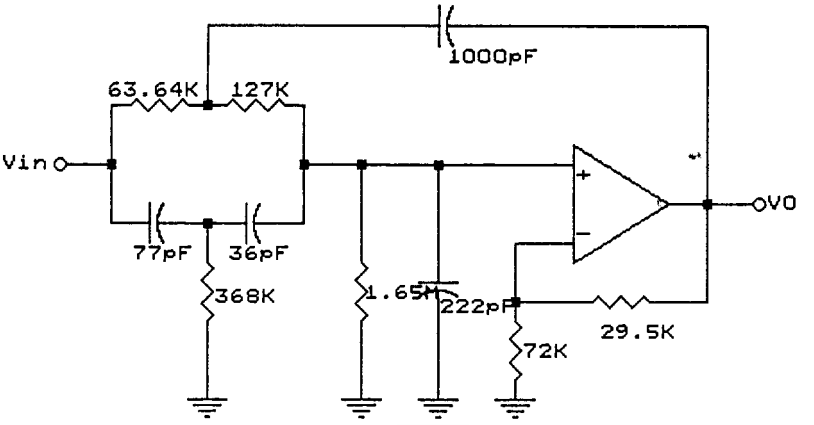

รูปที่ 5.9 วงจรส่วนเดลต้า ดีมอดูเลเตอร์

จากรูปข้างต้น สามารถอธิบายการทำงานได้ดังนี้ คือ

สัญญาณดิจิตอลที่ถูกส่งมาแบบอนุกรมจะถูกแปลงเป็นสัญญาณอนาลอกโดยเข้าทางขา 13 (Digital Input) ซึ่งเป็นอินพุทแบบกลับเฟส โดยขาที่ 15 จะต้องมีสถานะเป็น '0' (Decode) สัญญาณอนาลอกที่ได้จะถูกส่งออกทางขา 7 (Analog Output) ซึ่งสัญญาณที่ได้นี้ ตามคุณสมบัติของเดลต้ามอดดูเลชั่น จะทำให้สัญญาณความถี่สูงปะปนอยู่ ดังนั้นจึงจำเป็นต้องทำการกรองสัญญาณความถี่สูงนี้ออกไป โดยใช้วงจรกรองความถี่ดังรูปที่ 5.10



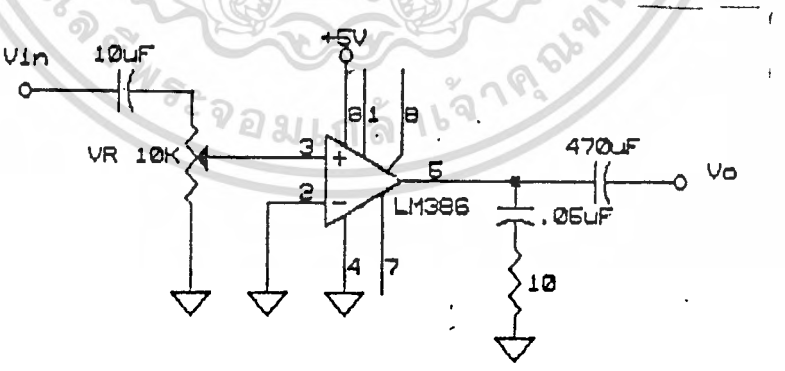
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 5.10 วงจรกรองความถี่สำหรับสัญญาณจาก CVSD

วงจรข้างต้น จะทำการกรองสัญญาณที่ออกจากขา 7 ของ MC3418 ซึ่งมีสัญญาณเป็นแบบสามเหลี่ยมให้มีความเรียบขึ้นจากนั้นสัญญาณเสียงที่ได้จะถูกกรองสัญญาณอีกครั้งด้วย วงจรกรองความถี่สัญญาณแบนด์พาสเพื่อให้สัญญาณมีความถี่อยู่ในช่วง 50 Hz ถึง 3.3 kHz ซึ่งวงจรและรายละเอียดจะเหมือนกับทางภาคอินพุท

ภาคสุดท้ายจะใช้ Op Amp ต่อเป็น Buffer เพื่อขับกระแส output ให้กับส่วนที่จะนำไปใช้งานจริงต่อไป แต่ในการทดลองจะทำการต่อกับส่วน Power Amp ทำการขยายกำลัง เพื่อใช้ในการขับลำโพง ซึ่งใช้ไอซี LM386 (AudioPower Amplifier) ซึ่งวงจรขยายกำลังที่ใช้มีอัตราการขยายเท่ากับ 20 เท่า ดังแสดงในรูปที่ 5.11




รูปที่ 5.11 แสดงวงจรขยายกำลังด้วย LM 386



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข) วงจร Digital

วงจรในส่วนนี้มีหน้าที่ในการแปลงสัญญาณ Parallel ให้เป็นสัญญาณ Digital แบบ Series และสร้างสัญญาณต่างๆที่จำเป็นในการทำงานของวงจร

จากหน้าที่ดังกล่าวข้างต้น สามารถออกแบบวงจรได้ดังรูปที่ 5.12

จากรูป วงจรส่วนนี้ประกอบด้วยอุปกรณ์ต่างๆ ซึ่งมีหน้าที่การทำงานดังนี้

- 74LS74 เป็น D-Flip Flop ซึ่งใช้เป็น register ขนาด 1 bit
- 74LS273 เป็น latch ขนาด 8 bit
- 74LS165 เป็นอุปกรณ์ในการแปลงสัญญาณแบบ Parallel ให้เป็นแบบ Series ซึ่งมีการทำงานดังนี้

เมื่อสัญญาณ SH/LD เป็น "0" 74LS165 จะทำการโหลดข้อมูลขนาด 8 bit เข้ามาเก็บไว้ในอินพุท register ภายในไอซี

เมื่อสัญญาณ SH/LD เป็น "1" 74LS165 จะทำการ shift ข้อมูลที่อยู่ในอินพุทregister ออกมาทีละ bit ตามจังหวะสัญญาณ clock ซึ่งสัญญาณที่ขา INHจะต้องเป็น "0"

- 74LS90 ใช้เป็นวงจรนับ 8
- MC14046 ทำหน้าที่สร้าง clock ให้กับส่วนต่างๆ ของวงจร

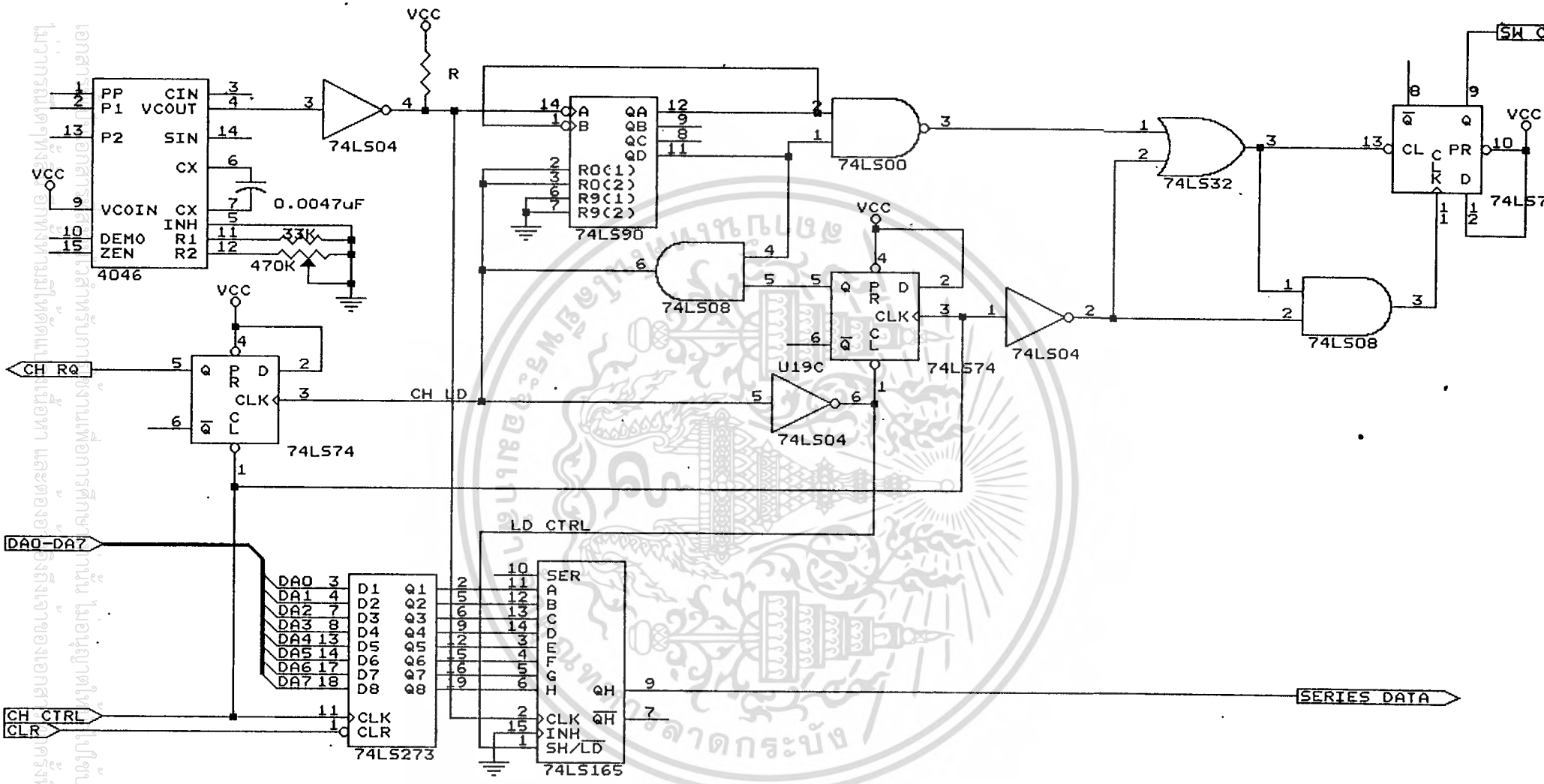
โดยจะอธิบายการทำงานโดยแบ่งเป็น 3 ส่วนย่อยๆ คือ

1. ส่วนสร้างสัญญาณที่จำเป็น

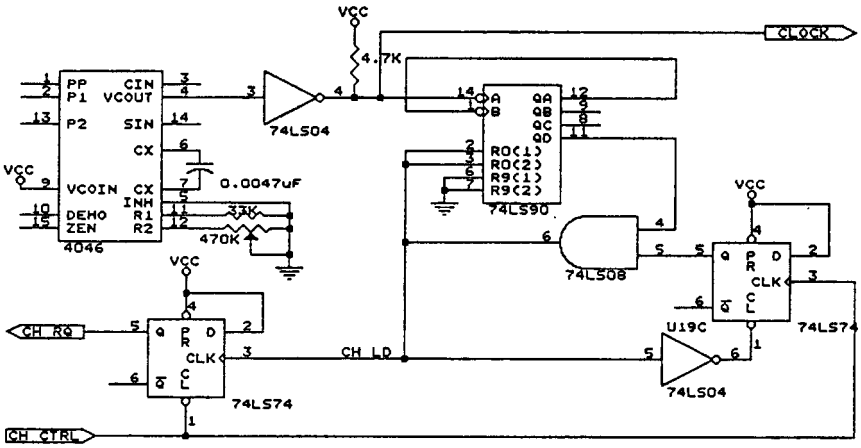
จากรูป 5.13 วงจรส่วนนี้ทำหน้าที่ในการสร้างสัญญาณที่จำเป็นในวงจร MC14046 สร้าง clock เป็นฐานเวลาให้ส่วนต่างๆ ของวงจร 74LS90 จะทำการนับตามจังหวะสัญญาณ clock เมื่อนับถึง 8 สัญญาณที่ขา$Q_D$จะเป็น "1" โดยนำสัญญาณ "CH CTRL" ( เป็นสัญญาณจากส่วน Inferface ที่ใช้ในการเลือก channel ที่จะ out ข้อมูลออกมา )ที่ถูกเก็บไว้ใน 74LS74 มาANDกับสัญญาณจากขา $Q_D$ ของ74LS90จะได้เป็นสัญญาณ"CH LD" ถ้าสัญญาณ "CH LD"เป็น "1" จะ reset 74LS90ให้เริ่มนับใหม่ ทำการ clear register ที่เก็บสัญญาณ 'CH CTRL' และจะทำให้ได้สัญญาณ 'CH RQ'เป็น 1 เพื่อทำการ บอกส่วน Interface ว่าต้องการข้อมูลชุดต่อไปเมื่อส่วน Interfaceส่งข้อมูลชุดใหม่ออกมา 'CH CTRL' จะเป็น 0 register ที่เก็บสัญญาณ 'CH RQ' จะถูก clear



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

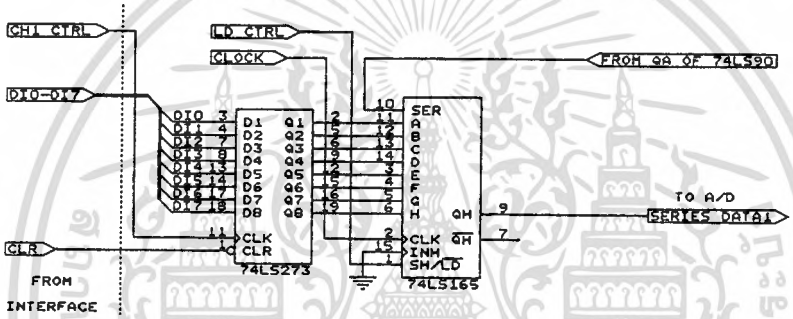



รูปที่ 5.12 วงจรส่วนดิจิตอล




รูปที่ 5.13

2.ส่วนควบคุมการ latch ข้อมูล และแปลงสัญญาณ digital




รูปที่ 5.14

จากรูป 5.14 วงจรส่วนนี้ทำหน้าที่ในการ latch ข้อมูล และแปลงสัญญาณจากสัญญาณ DigitalแบบParallelเป็นแบบ Series เมื่อสัญญาณ "CH CTRL" จากส่วน Interface เป็น "1" 74LS273 จะทำการlatchข้อมูลแบบ Parallel 8 bit จากส่วน Interface ไว้ และเมื่อสัญญาณ "LD CTRL" เป็น "1" 74LS165 จะ load ข้อมูลจาก 74LS273 เข้ามาแล้วทำการ shift ออกทีละ bit ตามจังหวะสัญญาณ clock

ค) ส่วน Interface

จากวงจรทั้งสองส่วนที่ออกแบบไว้แล้วจะติดต่อกับ PC โดยผ่านพอร์ท 9 พอร์ท ดังนี้

1. พอร์ทเอาท์พุทข้อมูล (Data Output Port) ใช้เป็นพอร์ทสำหรับส่งข้อมูลที่เก็บไว้ ให้กับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

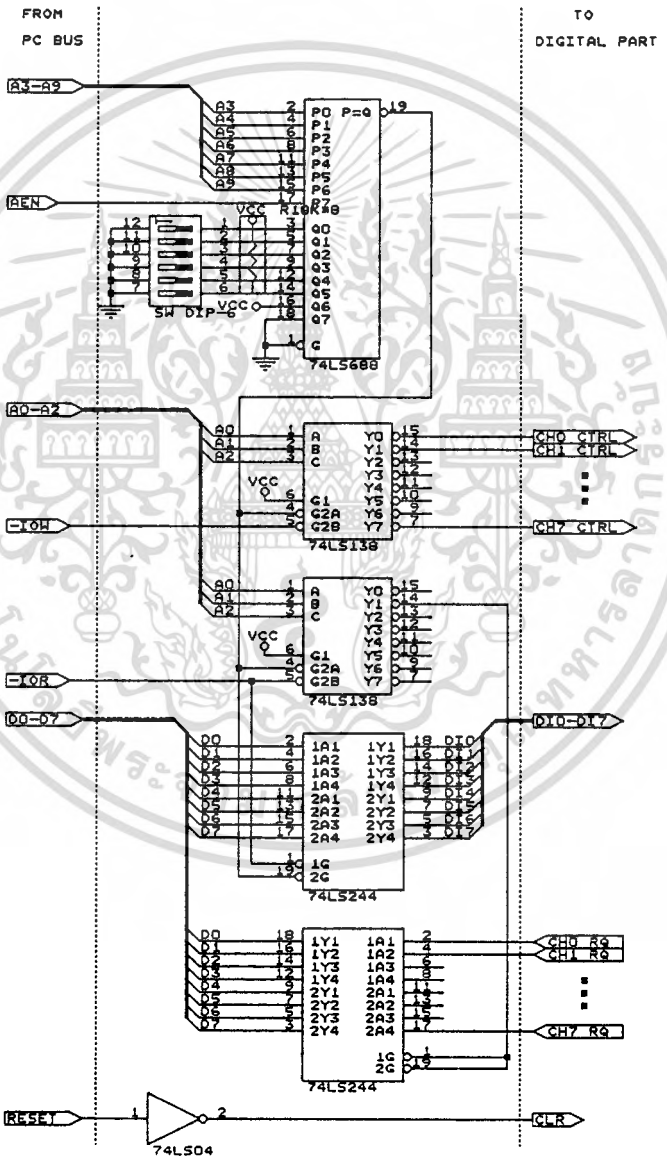
วงจรส่วน Digital มีทั้งหมด 8 channel (8 port)

2.พอร์ทอินพุทสัญญาณควบคุม (Control Signal Output Port) ใช้เป็นพอร์ทสำหรับรับสัญญาณ "CH RQ" จากส่วน Digital ให้ PC

เมื่อพิจารณาพอร์ทแอดเดรสแมพ ( Port Address Map) สำหรับ IBM PCแล้วจึงทำการเลือกพอร์ทแอดเดรสในช่วง 280h-287h ซึ่งเป็นช่วงที่ว่างจากการใช้งานของ IBM PC ดังนั้นจึงกำหนดให้

พอร์ท 280h - 287h เป็น พอร์ทเอาท์พุทข้อมูล

พอร์ท 281h เป็น พอร์ทอินพุทสัญญาณควบคุม




5.16 วงจร Interface ในส่วนกระจายเสียง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยวงจรส่วน interface สามารถออกแบบได้ดังรูปที่ 5.16

จากรูปบัสแอดเดรสของเครื่อง PC จำนวน 10 จะถูกนำมาดีโค้ทร่วมกับสัญญาณ AEN เพื่อใช้ควบคุมการทำงานโดยใช้ 74LS168 และ 74LS138 ในการดีโค้ทแอดเดรส ให้พอร์ท 280h ถึง 287h เป็นพอร์ทเอาท์พุท มีสัญญาณ IOW เป็นสัญญาณควบคุมพอร์ทเอาท์พุท โดย74LS138 ทำหน้าที่ในการ decode port ซึ่งก็คือ ทำหน้าที่ ในการสร้างสัญญาณ 'CH 0 CTRL'-' CH7 CTRL'เพื่อใช้ในการเลือก channel ที่ จะ out data ออกไป โดยมี 74LS244 ทำหน้าที่เป็น Buffer เพื่อขับกระแสในการ outport ข้อมูล และให้พอร์ท 281h เป็นพอร์ทอินพุท มีสัญญาณ IOR เป็นสัญญาณควบคุมพอร์ทอินพุทโดยมี 74LS244 เป็นอุปกรณ์ I/O ในการ in data

## 5.2 ซอฟแวร์ควบคุมการทำงานของวงจร

จากที่ได้กล่าวมาแล้วข้างต้นว่า ในส่วน นี้สามารถแบ่งการทำงานได้เป็น 2 ส่วนใหญ่ ๆ ซึ่งอธิบายรายละเอียดแต่ละส่วนได้ดังนี้

1.โปรแกรมควบคุมการจัดการข้อมูลเพื่อส่งข้อมูลให้ส่วนกระจายเสียง จะทำหน้าที่

-ตรวจสอบการขอติดต่อ (Connection)จาก channel ต่าง ๆ และเตรียมข้อมูลให้กับ channel ที่เริ่มทำการติดต่อ

-ตรวจสอบการขอข้อมูลจาก channel ที่ติดต่ออยู่ และส่งข้อมูลให้ได้อย่างต่อเนื่อง

2.โปรแกรมสนับสนุนการทำงาน ประกอบด้วย

-โปรแกรมบันทึกสัญญาณเสียง

-โปรแกรมบันทึกและแก้ไขข้อมูล

ในโปรแกรมใช้งานกับข้อมูลเสียงจำเป็นต้องมีหน่วยความจำที่มีจำนวนมากเพราะข้อมูลเสียงมักมีขนาดใหญ่มาก (ใน project นี้ ถ้าต้องการใช้ข้อมูลเสียงที่มีความยาว 30 วินาที จะต้องใช้หน่วยความจำที่มีขนาด 60 kbytes ) ซึ่งการนำข้อมูลเสียงทั้งหมดมาใช้งานในคราวเดียวจะกระทำได้ยากมาก (ไม่สามารถจองหน่วยความจำที่มีขนาดตามที่ต้องการได้ ) ดังนั้นในโปรแกรมนี้จึงได้ทำการแบ่งข้อมูลเสียงออกเป็นส่วน ๆ เพื่อความสะดวกในการใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมควบคุมการจัดการข้อมูลเพื่อส่งให้ส่วนกระจายเสียง

อัลกอริธึม (Algorithm)สำหรับการทำงานของโปรแกรมในส่วนนี้ คือ จะทำการรับข้อมูลเริ่มการติดต่อ (ในการทดลอง จะใช้ การรับค่าจาก keyboard) คอยตรวจสอบสถานะการขอข้อมูลชุดต่อไปโดยการ in data จาก port 281h และส่งข้อมูลออกไปให้ channel ที่ขอข้อมูลโดยการ out data ออก port (280h-287h) สามารถอธิบายหลักการคร่าว ๆ ได้ดังนี้

1.ตรวจสอบข้อมูลที่แสดงความต้องการขอเริ่มการติดต่อ ซึ่งข้อมูลจะประกอบด้วยหมายเลขช่องสัญญาณ และหมายเลขเรื่องข้อมูล(โดยตรวจสอบทุกครั้งที่มีการกด keyboard ) ในขั้นนี้จะทำเมื่อ PC พร้อมที่จะทำการเตรียมข้อมูลชุดใหม่ให้กับช่องสัญญาณที่ขอเข้ามาใหม่เท่านั้น

2.จัดเตรียมข้อมูลให้กับ channel ที่ขอใช้ข้อมูล แต่การเตรียมข้อมูลในครั้งแรก จะไม่สามารถทำให้เสร็จในครั้งเดียว เนื่องจากจะต้องตรวจสอบข้อมูลที่รับเข้ามา เตรียมพารามิเตอร์ที่จำเป็นในการตรวจสอบสถานะการทำงานต่าง ๆ และทำการเปิดไฟล์ loadข้อมูล ซึ่งเมื่อรวมแล้วจะต้องใช้เวลานาน อาจมีผลต่อการส่งข้อมูลชุดต่อไปให้กับchannel ที่ใช้งานอยู่ จึงต้องทำการแบ่งงานทั้งหมดออกเป็นงานย่อย ๆ แล้วทำทีละส่วนทุกครั้งที่วนloop การทำงานมาถึง

3.รับค่าที่แสดงสถานะการขอข้อมูลชุดต่อไปของช่องสัญญาณที่อยู่ในระหว่างการใช้งาน โดยจะทำการตรวจสอบค่าแสดงสถานะทีละบิท

-ถ้า channel ใดต้องการข้อมูลชุดต่อไป สัญญาณที่บิทแสดงสถานะของ channel นั้นจะเป็น '1' PC จะส่งข้อมูลชุดต่อไปออกไปให้และตรวจสอบblockข้อมูล

-ถ้าไม่มี channel ใดต้องการข้อมูลชุดต่อไปเลย ข้อมูลแสดงสถานะจะมีค่าเป็น '0' PC จะกลับไปตรวจสอบการขอเริ่มการติดต่อจาก อื่น ๆ (ตรวจสอบการกดkeyboard ) หรือเตรียมข้อมูลที่ยังทำไม่เสร็จทันที

4.ตรวจสอบการสิ้นสุดblock ข้อมูลและการสิ้นสุดของไฟล์ข้อมูลโดยที่

-ถ้าสิ้นสุด blockข้อมูล จะทำการ load ข้อมูล เพิ่ม

-ถ้าสิ้นสุด block ข้อมูลและเป็น blockข้อมูลชุดสุดท้าย(สิ้นสุดไฟล์) จะทำการ ยกเลิกการติดต่อ (Disconnection) channel นั้น

-ถ้ายังไม่สิ้นสุด blockข้อมูล PCจะทำงานขั้นต่อไป

5.ทำการตรวจสอบจำนวนchannel ที่ติดต่ออยู่ โดยที่

-ถ้ายังมีการ connection กับ channel อยู่จะเริ่มทำตามขั้นที่ 1 ใหม่

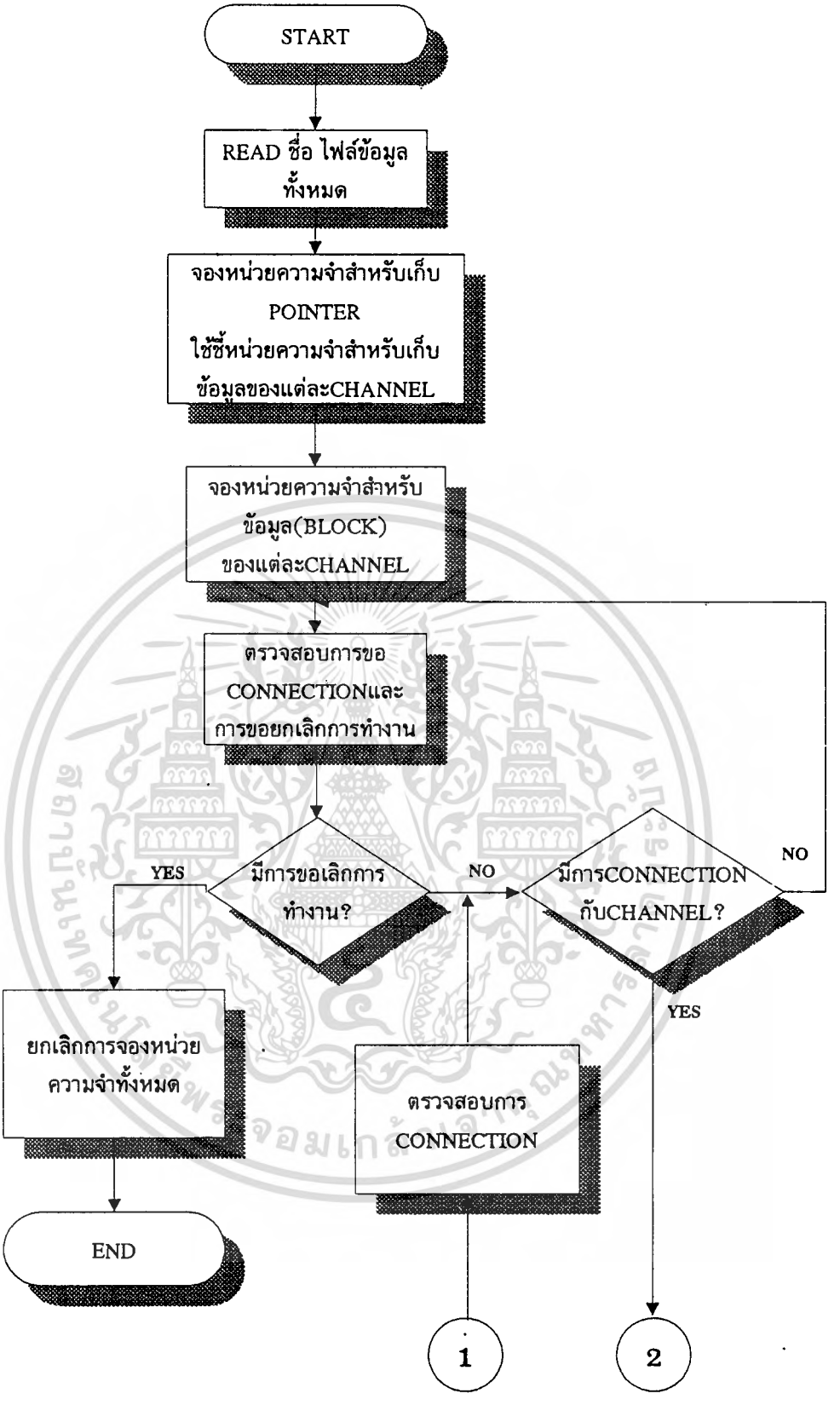


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ถ้าไม่มีการ connection แล้ว จะอยู่ในสภานะรอ การขอconnection หรือ การยกเลิก

ซึ่งแผนผังแสดงการทำงานเป็นดังรูป 5.17

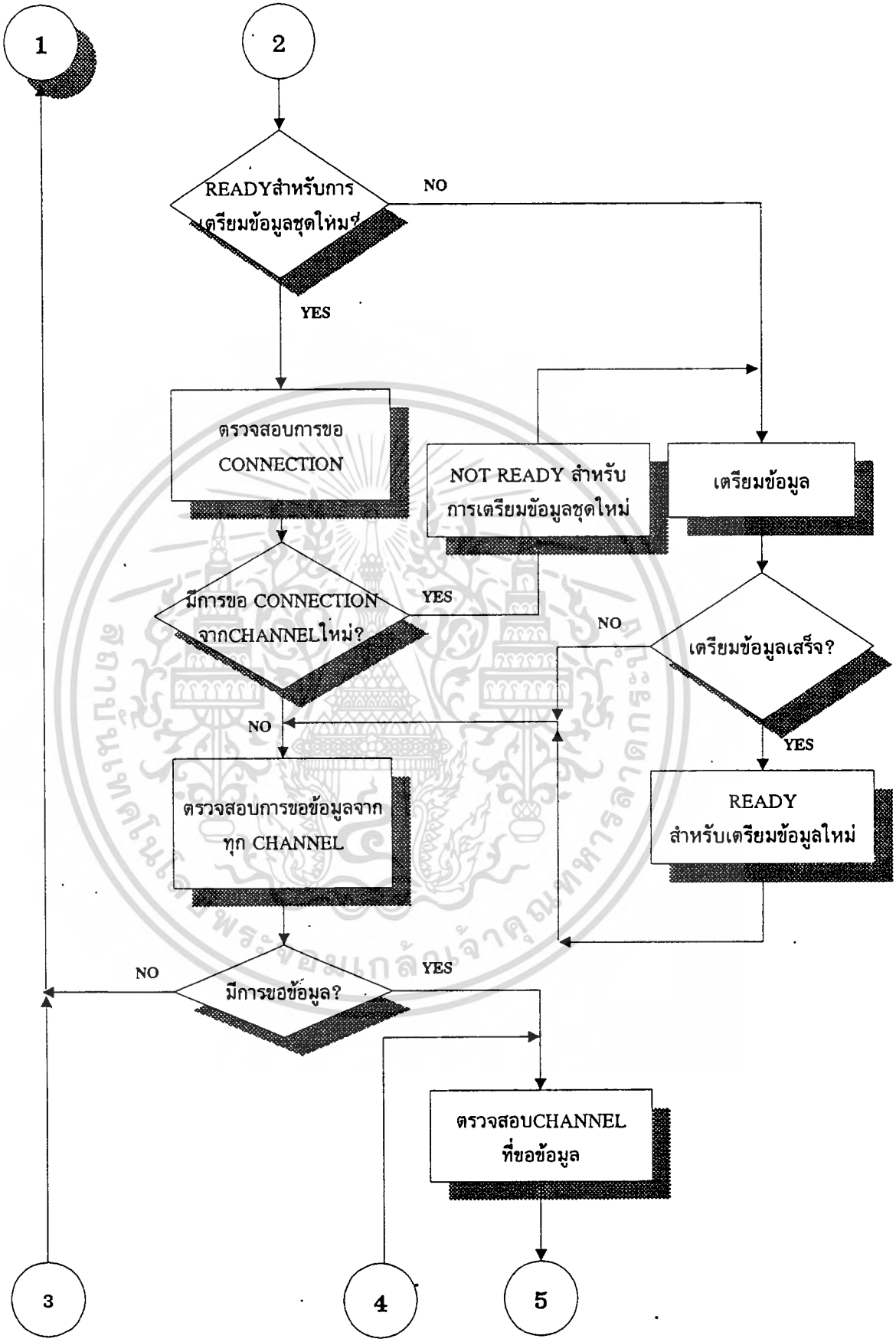
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



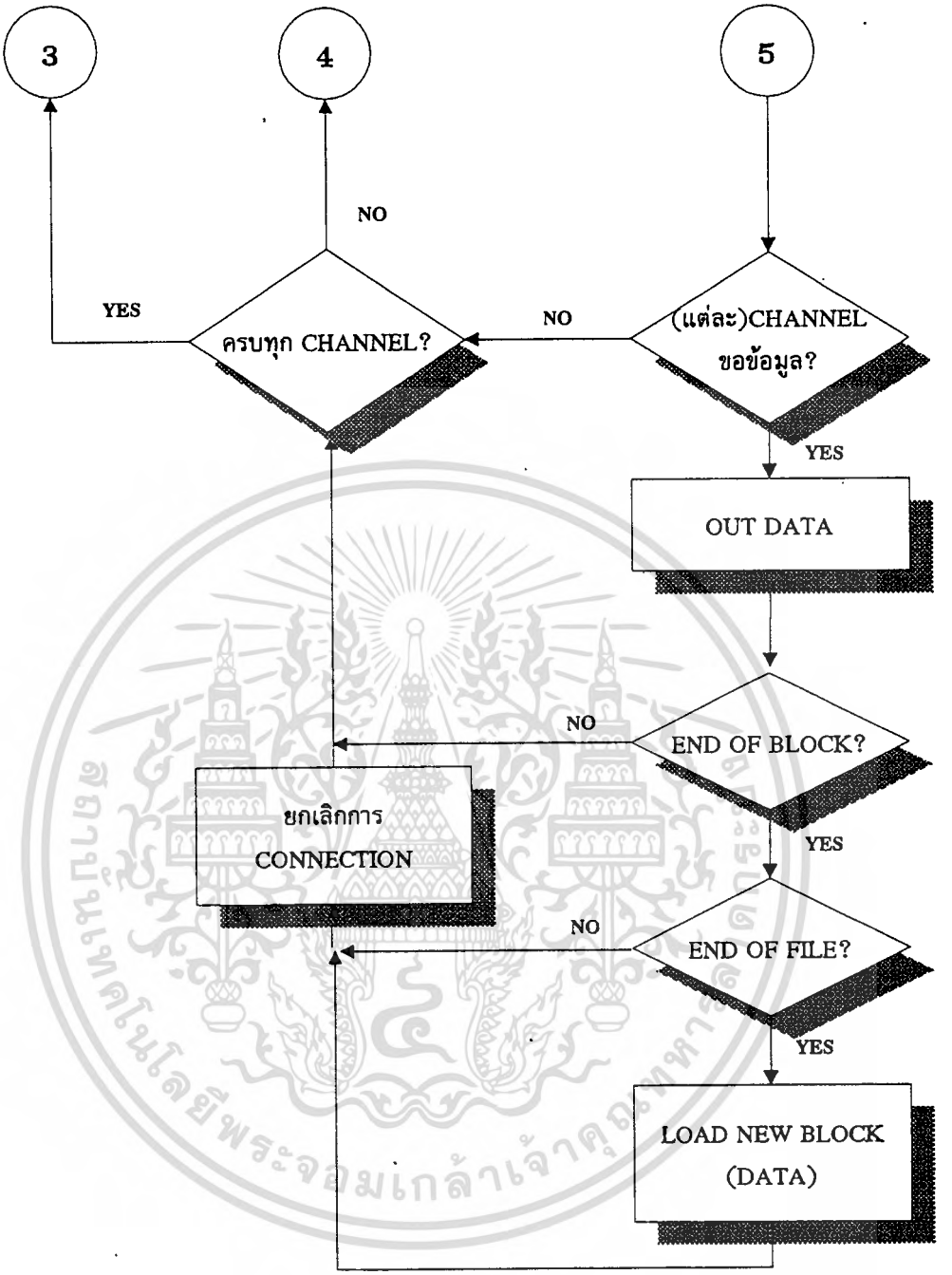

รูปที่ 5.17 Flowchart แสดงการทำงานของโปรแกรมควบคุมการจัดการข้อมูล



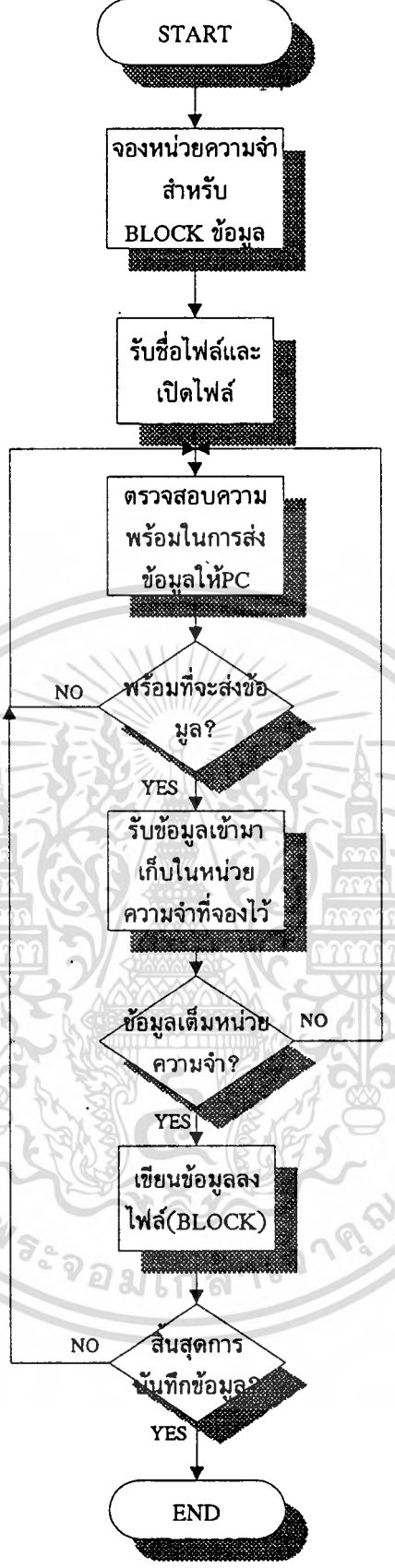
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า




รูปที่ 5.18 Flowchart แสดงการทำงานของโปรแกรมบันทึกข้อมูลเสียง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**โปรแกรมสนับสนุนการทำงาน**

ตามที่ได้กล่าวมาแล้ว ว่า จะเป็น โปรแกรมสำหรับบันทึกข้อมูลเสียง และโปรแกรมบันทึกและแก้ไขข้อมูล ซึ่งได้ทำการรวมโปรแกรมทั้งสองหน้าที่นี้ไว้ใน โปรแกรมเดียว เป็น Menu เลือกการทำงาน โดยยังอาศัยหลักการใช้งานข้อมูลเสียงแบบเดิม คือ ต้องแบ่งข้อมูลออกเป็นส่วน ๆ เพื่อหลีกเลี่ยงปัญหาการจองหน่วยความจำขนาดใหญ่ หลักการทำงานของโปรแกรมบันทึกเสียงสามารถแสดงได้ดังรูป 5.18





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6

## สรุป

ผลจากการทำโครงงานตาม แนวความคิดที่ได้ออกแบบไว้ทั้งในส่วนที่เป็น วงจรและโปรแกรมควบคุมการทำงาน สามารถพิสูจน์ได้ว่า วงจรที่ได้ออกแบบไว้สามารถทำงานได้ตามที่ต้องการ คือ ในส่วน Hardware จะส่งสัญญาณขอข้อมูลชุดต่อไปทันทีที่ได้มีการรับข้อมูลไปจาก PC เพื่อให้สามารถส่งข้อมูลเสียงออกไปได้อย่างต่อเนื่อง ทำการแปลงข้อมูลที่รับมาให้อยู่ในรูปที่ Converter(D/A) สามารถนำไปแปลงได้ และทำการเปิด analog switch เมื่อมีการส่งข้อมูลออกไป ปิด switch เมื่อไม่มีข้อมูล(ไม่ใช้งาน) หรือ ข้อมูลที่ต้องการส่งมาไม่ทัน ในขณะที่ส่วน Software จะทำการตรวจสอบสัญญาณขอข้อมูลและทำการส่งข้อมูลออกไปให้กับช่องสัญญาณที่ขอ รวมถึงการตรวจสอบการขอเริ่มการติดต่อ (ใช้งาน)

อย่างไรก็ตาม ถึงแม้วงจรจะทำงานได้ตามคุณสมบัติที่กล่าวไว้ในบทนำ แต่ยังคงมีปัญหาสัญญาณรบกวนที่เกิดจาก

-ไฟบ้าน มีความถี่ 50 Hz

-ไฟเลี้ยง คอมพิวเตอร์

-การเปิด ปิด analog switch จะทำให้ เกิดเสียงดังเป็นระยะ ๆ สาเหตุเนื่องมาจากขณะที่ PC ทำการโหลดข้อมูลจาก Harddisk จะทำให้การส่งข้อมูลมายังช่องสัญญาณที่ขอเกิดการไม่ต่อเนื่อง(ในกรณีนี้ จะมีการปิด analog switch และจะเปิด เมื่อมีข้อมูลมาแล้ว ) จากสาเหตุข้างต้น ได้ทำการแก้ไขโดยไม่ใช้ analog switch และป้อน clock ที่มีความถี่เป็นครึ่งหนึ่งของความถี่ที่ใช้เป็นฐานเวลาในวงจรให้กับขา series input ของ IC เบอร์ 74LS165 ซึ่งเสมือนเป็นการป้อนข้อมูลหลอกให้แก่ IC 74LS165ทำให้ช่วงเวลาที่ PC ทำการโหลดข้อมูลอยู่นั้น output จาก IC 3418 ไม่ตกลงเป็น 0 ทันที แต่จะค่อยๆตกลงจนเป็น 0 ดังนั้นเสียงที่ได้รับจะมีความต่อเนื่องมากขึ้น อย่างไรก็ตาม การส่งข้อมูลไม่ทันนี้เกิดเป็นระยะเวลาสั้น ๆ เท่านั้น ทำให้สัญญาณเสียงที่ได้รับ จะมีคุณภาพที่ยอมรับได้

<u>แนวทางแก้ไขปัญหา</u>

เลือกใช้ ส่วนประมวลผลที่เป็น Multiprocessor หรือ เครื่อง PC ที่มี CPU ความเร็วสูง เพื่อแก้ปัญหาการส่งข้อมูลให้ช่องสัญญาณที่ขอ ไม่ทันเนื่องจาก เสียเวลาส่วนหนึ่งในการ load Block ข้อมูลใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "mouse.c"
#include "function.c"
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <bios.h>

#define UP    0x48
#define DOWN  0x50
#define ENTER 0x1C
#define chan  8
#define nport 0x280

int nfile,i;
char fname[chan][11];
FILE *file;

void main(void){

        int actx,acty,act,wait,ascii,scan;
        char option[4][10];

        strcpy(option[0],"PLAY");
        strcpy(option[1],"RECORD");
        strcpy(option[2],"DELETE");
        strcpy(option[3],"QUIT");

        clrscr();
        if ((file=fopen("recfile","rt"))==NULL){
           square(0,0,80,25,CYAN);
           square(23,6,59,16,BLACK);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    WINDOW(22,5,58,15,BLUE,YELLOW);
    gotoxy(27,10);
    cprintf("THERE 'S NO RECORD\n");
    getch();
}
else{
    while(!feof(file)){
            if (fread(&fname[nfile],sizeof(fname[nfile]),1,file)==0)
           break;
            nfile++;
    }
    fclose(file);
}

do{
/*make menu*/
wait=1;
WINDOW(0,0,80,25,LIGHTGRAY,LIGHTGRAY);
WINDOW(20,5,60,20,LIGHTCYAN,MAGENTA);
gotoxy(35,7);
textcolor(YELLOW);textbackground(MAGENTA);
cprintf(" MENU \n");
barbox(33,10,option[0]);
nobarbox(33,13,option[1]);
nobarbox(33,16,option[2]);
nobarbox(33,19,option[3]);
installmouse();
actx=33;
acty=10;
act=0;

do{
        if(left_button()){ /*for use mouse*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nobarbox(33,10,option[0]);
getposition(&actx,&acty);
actx=actx/8+1;acty=acty/8+1;
gotoxy(actx,acty);

if((actx>=33)&&(actx<=43)){ /*position mouse right*/
        switch(acty){
                case 10:actx=33;acty=10;act=0;
                        barbox(actx,acty,option[act]);
                        while(left_button());
                        wait=0;hidemouse();
                        play();
                        break;

                case 13:actx=33;acty=13;act=1;
                        barbox(actx,acty,option[act]);
                        while(left_button());
                        wait=0;hidemouse();
                        record();
                        break;

                case 16:actx=33;acty=16;act=2;
                        barbox(actx,acty,option[act]);
                        while(left_button());
                        wait=0;hidemouse();
                        del();
                        break;

                case 19:square(0,0,80,25,BLACK);
                        textcolor(WHITE);
                        hidemouse();exit(1);

                default:barbox(33,10,option[0]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        break;
                    }/*end switch case*/
            }/*end  for position right*/
}/*end for use mouse*/

if(bioskey(1)){/*use keyboard*/
        scan=0;ascii=getch();
        if(ascii==0x0d)
                if(acty==10){
                  play();
                  wait=0;
                }
                else if(acty==13){
                  wait=0;
                  record();
                }
                else if(acty==16){
                  wait=0;
                  del();
                }
                else if(acty==19) {
                  square(0,0,80,25,BLACK);
                  textcolor(WHITE);
                  hidemouse();
                  exit(1);
                }
        if((ascii==0xe0)||(ascii==0x00)){/*shift cursor bar*/
                scan=getch();
                nobarbox(actx,acty,option[act]);
                switch (scan){
                        case UP  :if(acty==10) {
                                  acty=19; act=3;
                                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    else{
                                      acty=acty-3;act=act-1;
                                    }
                                    break;

                            case DOWN:if(acty==19){
                                      acty=10; act=0;
                                      }
                                      else {
                                          acty=acty+3;act=act+1;
                                      }
                                      break;

                                }
                            barbox(actx,acty,option[act]);
                        }
                    }
                }while(wait);
                }while(1);
}/*main*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <alloc.h>
#include <string.h>
#include <io.h>

#define chan 8
#define nport 0x280
#define BSIZE 500
#define mxchandgt 1
#define mxfiledgt 1

#define  IDLE          0
#define  GETCHAN       1
#define  GETCHANRDY    2
#define  CVERTCHAN     3
#define  CHECKCHAN     4
#define  GETFILE       5
#define  CHECKCANCEL   6
#define  GETFILERDY    7
#define  CVERTFILE     8
#define  CHECKFILE     9
#define  CLOSEUSEDFILE 10
#define  OPENFILE      11
#define  NOFILE        12
#define  SIZEFILE      13
#define  READFILE      14
#define  CALSIZE       15
#define  SETPARA       16
#define  OUTDATA       17
#define  SHOW          18
#define  CLOSEFILE     19
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define  CLOSEFILE2     20
#define  READY          21

void play(void);
void record(void);
void del(void);
void delword(int length,int color);
void WINDOW(int x1,int y1,int x2,int y2,int tcolor,int bcolor);
void square(int x1,int y1,int x2,int y2,int color);
void table(int x1,int y1,int b1,int b2,int line,int space,int linecol,int backcolor);

void play(void){
        extern char fname[chan][11];
        extern int nfile;
        int i,chn,rec,nrec=0,status=IDLE,digit=0,end[chan];
        char keyin[5];
        unsigned int d[chan],j;
        unsigned char **datap,condat,chctrl,chreq;
        unsigned long ssize[chan];
        FILE *load[chan],*file;

        if((datap=(unsigned char **)calloc(chan+1,sizeof(unsigned char*)) )==NULL){
          square(0,0,80,25,CYAN);
          square(23,6,59,16,BLACK);
          WINDOW(22,5,58,15,BLUE,YELLOW);
          gotoxy(25,10);
          cprintf("READY ALLOCATION MEMORY FOR POINTER\n");
          getch();
          return;
        }

        for(i=0;i<chan;i++){
          ssize[i]=0;d[i]=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end[i]=0;
    if( (datap[i]=(unsigned char *)calloc(BSIZE,sizeof(unsigned char)) )==NULL){
            square(0,0,80,25,CYAN);
            square(23,6,59,16,BLACK);
            WINDOW(22,5,58,15,BLUE,YELLOW);
            gotoxy(25,10);
            cprintf("READY ALLOCATION MEMORY FOR DATA\n");
            getch();

            for(j=0;j<i;j++)
              if(datap[j]!=NULL)
                free(datap[j]);
            free(datap);
            return;
    }
}

inportb(0x280);

  clrscr();
  square(0,0,80,25,CYAN);
  square(17,6,61,24,BLACK);
  table(16,5,9,30,9,1,YELLOW,GREEN);
  textcolor(WHITE);
  gotoxy(18,6);
  cprintf("CHANNEL");
  gotoxy(38,6);
  cprintf("DATA");
  textcolor(BLUE);

  for(i=0;i<chan;i++){
            gotoxy(23,8+2*i);
            cprintf("%d",i);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
gotoxy(70,23);
cprintf("NOW :0");

while(1){
   if(nrec==0){
         WINDOW(10,1,70,4,GREEN,MAGENTA);
          gotoxy(26,2);
          cprintf("WAITING FOR CONNECTION REQUEST");
          gotoxy(25,3);
          cprintf("PRESS 'q' to return to main menu");

        if( getch()=='q'){
                for(i=0;i<chan;i++)
                     if (datap[i]!=NULL)
                          free(datap[i]);
                free(datap);
                return;
        }
        else status=GETCHAN;
}

 gotoxy(26,2);
 delword(30,MAGENTA);
 gotoxy(25,3);
 delword(35,MAGENTA);
 gotoxy(35,3);
 textcolor(GREEN);
 cprintf("ON USE");
 gotoxy(20,24);
 textbackground(GREEN);
 textcolor(BLUE);
 gotoxy(20,24);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    cprintf("PREPARING BLOCK DATA");

while(nrecliistatus){
    textbackground(GREEN);
    textcolor(BLUE);

    if( status==READY){
                gotoxy(20,24);
                delword(20,BLACK);
    }

    if( (status==IDLE)&&(kbhit()) ){
                gotoxy(20,24);
                cprintf("PREPARING BLOCK DATA");
                status=GETCHAN;
    }

    if(status){
        switch(status){

            case GETCHAN        : if (kbhit()){
                                  keyin[digit]=getch();
                                  status=GETCHANRDY;
                                  }
                                  break;

            case GETCHANRDY : if((digit>mxchandgt)||(keyin[digit]==0x0d)){
                                  status=CVERTCHAN;
                                  digit=0;
                                  }
                                  else{
                                  digit++;
                                  status=GETCHAN;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                    break;

case CVERTCHAN : chn=atoi(keyin);
                    status=CHECKCHAN;
                    break;

case CHECKCHAN : if(chn<0 || chn>chan )
                            status=READY;
                    else{ status=GETFILE;
                    gotoxy(1,1);cprintf("chn=%d\n",chn);
                    }
                    break;

case GETFILE    :  if(kbhit()){
                    keyin[digit]=getch();
                    status=CHECKCANCEL;
                    }
                    break;

case CHECKCANCEL : if(keyin[digit]=='q'){
                    if(ssize[chn]!=0){
                            status=CLOSEFILE;
                            ssize[chn]=0;
                    }
                    else status=READY;
                }
                    else status=GETFILERDY;
                    break;

case GETFILERDY : if((digit==mxfiledgt)||(keyin[digit]==0x0d)){
                    status=CVERTFILE;
                    digit=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                    else{
                        digit++;
                        status=GETFILE;
                     }
                        break;

case CVERTFILE   : rec=atoi(keyin);
                    status=CHECKFILE;
                    break;

case CHECKFILE   : if((rec<0) || (rec>nfile))
                        status=READY;
                     else{
                        status=CLOSEUSEDFILE;
                        gotoxy(1,2);cprintf("FILE:%s\n",fname[rec]);
                     }
                    break;

case CLOSEUSEDFILE :if(ssize[chn]!=0){
                        fclose(load[chn]);
                        gotoxy(30,8+2*chn);
                        delword (10,GREEN);
                        }
                        else nrec++;
                        status=OPENFILE;
                        break;

case OPENFILE :  if((load[chn]=fopen(fname[rec],"rb"))==NULL){
                        status=NOFILE;
                        nrec--;
                    }
                        else status=SIZEFILE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;

case NOFILE  :  fclose(load[chn]);
                status=READY;
                break;

case SIZEFILE :  ssize[chn]=filelength( fileno(load[chn]) );
                 status=READFILE;
                 break;

case READFILE : fread(datap[chn],sizeof(unsigned char),BSIZE,load[chn]);
                 status=CALSIZE;
                 break;

case CALSIZE :  ssize[chn]=ssize[chn]-BSIZE;
                 status=SETPARA;
                 break;

case SETPARA:end[chn]=0;d[chn]=0;
                      status=OUTDATA;
                      break;

case OUTDATA : outportb(nport+chn,datap[chn][d[chn]]);
                 status=SHOW;
                 d[chn]++;
                 break;

case SHOW  :   gotoxy(30,8+2*chn);
                 cprintf(fname[rec]);
                 status=READY;
                 break;

case CLOSEFILE : fclose(load[chn]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    nrec--;
                    status=CLOSEFILE2;
                    break;

            case CLOSEFILE2 :ssize[chn]=0;
                     gotoxy(30,8+2*chn);
                    delword(10,GREEN);
                    status=READY;
                    break;

            case READY :  status=IDLE;
                    digit=0;
                    textbackground(GREEN);
                    gotoxy(75,23);
                    cprintf("%d",nrec);
                        break;
}
}
condat=inportb(0x281);

if(condat!=0){
        for(chctrl=0x01,j=0;j<chan;j++){
        chreq=condat&chctrl;
        if(chreq!=0&&ssize[j]!=0){
                outportb(nport+j,datap[j][d[j]]);
                d[j]++;
                if(d[j]==ssize[j]&&end[j]==1){
                 if(d[j]==0){
                   printf("error condat channel %d\n",j);
                   getch();
                 }
                 fclose(load[j]);
                 ssize[j]=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    nrec--;
                    gotoxy(30,8+2*j);
                    delword(10,GREEN);
                    gotoxy(75,23);
                    cprintf("%d",nrec);
                   }
                   else if(d[j]==BSIZE){
                    if(ssize[j]>BSIZE){
                     fread(datap[j],sizeof(unsigned char),BSIZE,load[j]);
                     ssize[j]=ssize[j]-BSIZE;
                     d[j]=0;
                    }
                    else{ fread(datap[j],sizeof(unsigned char),ssize[j],load[j]);
                    d[j]=0;end[j]=1;
                    }
                    }

                  }/*chreq!=0*/
                  chctrl<<=1;
                 } /*shift chctrl*/
               }/*end condat!=0*/
              }/*end while (nrec)*/
             }/*end while(1)*/
}


void delword(int length,int color){
        int i;

        textbackground(color);
        for(i=0;i<length;i++)
                putch(0x20);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void square(int x1,int y1,int x2,int y2,int color){
        int x,y;

        textbackground(color);
        for(y=y1;y<=y2;y++){
                for(x=x1;x<=x2;x++){
                        gotoxy(x,y);
                        putch(0x20);
                }
        }
}

void table(int x1,int y1,int b1,int b2,int line,int space,int linecol,int bcol){
        int i,x,y,x2,y2;

        textcolor(linecol);
        textbackground(bcol);
        x2=x1+b1+b2+2;
        y2=y1+(space+1)*line;
        for(y=y1+1;y<y2;y++){
                for(x=x1+1;x<x2;x++){
                        gotoxy(x,y);
                        putch(0x20);
                }
                gotoxy(x1,y);       putch(0xb3);
                gotoxy(x2,y);       putch(0xb3);
        }
        for(y=y1+1;y<y2;y++){
                gotoxy(x1+b1+1,y);
                putch(0xb3);
        }
        for(i=0;i<line+1;i++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for(y=y1;y<=y2;y=y+(space+1)){
                        gotoxy(x1,y);           putch(0xc3);
                        gotoxy(x1+1,y);
                        for(x=x1+1;x<x2;x++)
                                if(x==b1+x1+1)      putch(0xc5);
                                else        putch(0xc4);
                        gotoxy(x2,y);           putch(0xb4);
                }
        }
        gotoxy(x1,y1);          putch(0xda);
        gotoxy(x1+b1+1,y1);             putch(0xc2);
        gotoxy(x2,y1);          putch(0xbf);
        gotoxy(x1,y2);          putch(0xc0);
        gotoxy(x1+b1+1,y2);             putch(0xc1);
        gotoxy(x2,y2);          putch(0xd9);
}

void WINDOW(int x1,int y1,int x2,int y2,int tcolor,int bcolor){
        int x,y;

        textcolor(tcolor);
        textbackground(bcolor);
        for(y=y1+1;y<y2;y++){
                for(x=x1+1;x<x2;x++){
                        gotoxy(x,y);
                        putch(0x20);
                }
                gotoxy(x1,y);putch(0xb3);
                gotoxy(x2,y);putch(0xb3);
        }
        gotoxy(x1+1,y1);
        for(x=x1+1;x<x2;x++)putch(0xc4);
        gotoxy(x1+1,y2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for(x=x1+1;x<x2;x++)putch(0xc4);
        gotoxy(x1,y1);putch(0xda);
        gotoxy(x2,y1);putch(0xbf);
        gotoxy(x1,y2);putch(0xc0);
        gotoxy(x2,y2);putch(0xd9);
}

void record(void){

        extern int nfile;
        extern char fname[chan][11];
        char ch,sname[11];
        unsigned char *datar,condat,chdat;
        unsigned int i;
        FILE *file;

        clrscr();
        sname[0]=10;
        square(0,0,80,25,CYAN);
        WINDOW(20,4,60,6,GREEN,BLUE);
        gotoxy(22,5);
        cprintf("RECORD NAME :");
        gotoxy(38,5);
        i=0;
        do{
           sname[i]=getche();
           i++;
         }while((sname[i-1]!='\r')&&(i<=10));

        sname[i-1]='\0';
        strcpy(fname[nfile],sname);
        file=fopen(fname[nfile],"wb");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WINDOW(25,7,55,10,BLUE,WHITE);
gotoxy(27,8);
cprintf("PRESS ANY KEY WHEN READY");
gotoxy(30,9);
cprintf("PRESS 'Q' TO QUIT");
ch=getch();

if (ch=='q')
        return;
gotoxy(27,8);
delword(25,WHITE);
gotoxy(30,9);
delword(20,WHITE);
gotoxy(35,8);
cprintf("RECORDING");

datar=(unsigned char*)calloc(BSIZE,sizeof(unsigned char));
do{
        for(i=0;i<BSIZE;i++){
                outportb(nport+4,0xff);
                datar[i]=inportb(nport+4);
                do{
                        condat=(inportb(nport+3)&0xff);
                        chdat=condat&0x01;
                }while(chdat==0);
                if (condat==0x03){
                        gotoxy(38,24);
                        cprintf("\n Overrun data error!!\n");
        }
    fwrite(datar,sizeof(unsigned char),BSIZE,file);
}while(!kbhit());

gotoxy(35,8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            delword(10,WHITE);
            gotoxy(30,8);
            cprintf("SUCCESS RECORD");
            fclose(file);
            file=fopen("recfile","at");
            fwrite(&fname,sizeof(fname[nfile]),1,file);
            fclose(file);
            nfile++;
            getch();
}

void del(void){
            extern int nfile;
            extern char fname[chan][11];
            int i;
            char ch;
            FILE *file;

            clrscr();
            square(0,0,80,25,CYAN);
            if(nfile==0){
                        WINDOW(20,8,60,13,BLUE,WHITE);
                        gotoxy(30,10);
                        cprintf("THERE 'S NO RECORD");
                        getch();
                        return;
            }
            WINDOW(20,2,60,24,BLUE,WHITE);
            gotoxy(30,3);
            cprintf("RECORD");
            for(i=0;i<nfile;i++){
                        gotoxy(25,4+i);
                        cprintf("%d.%s\n",i+1,fname[i]);
```

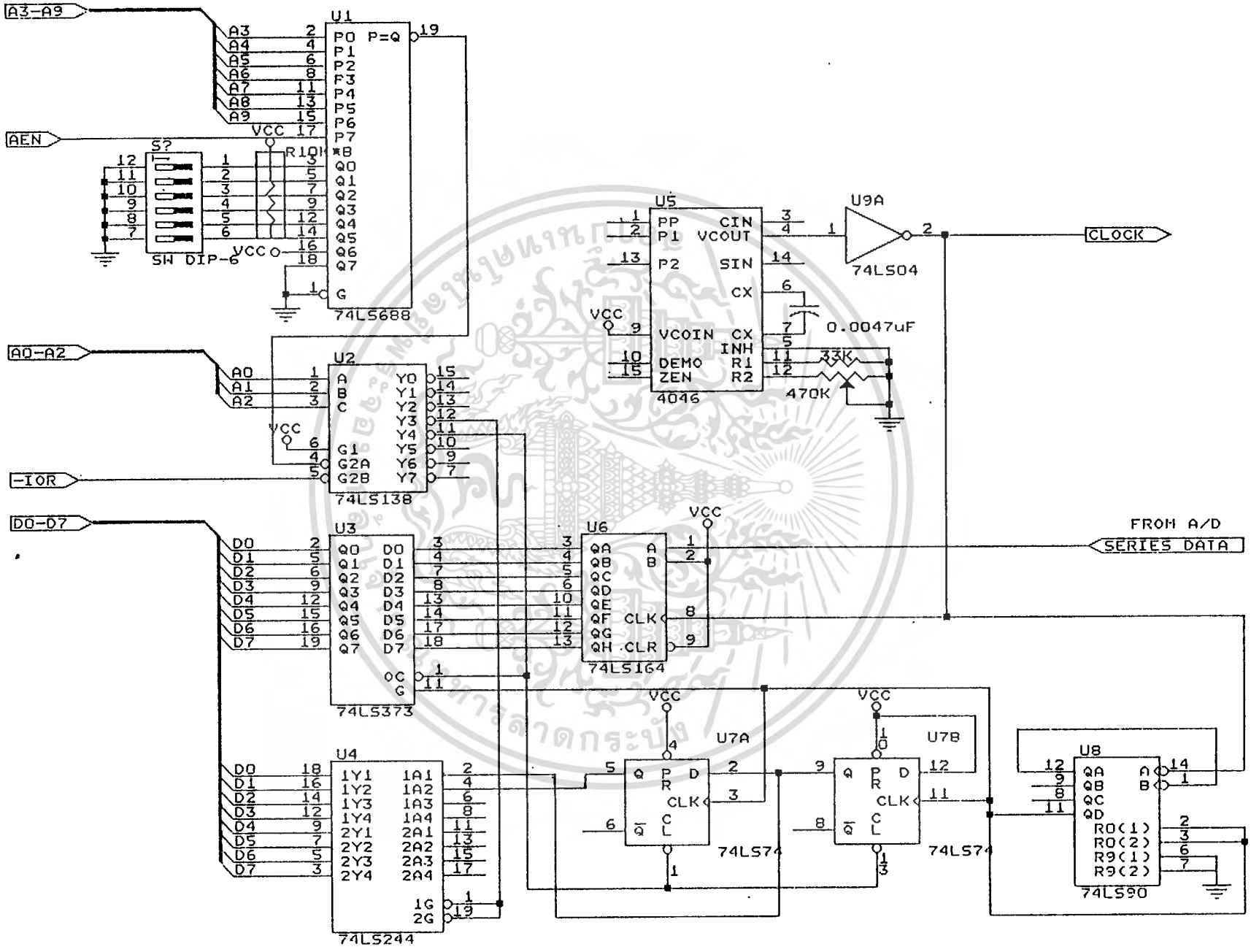
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

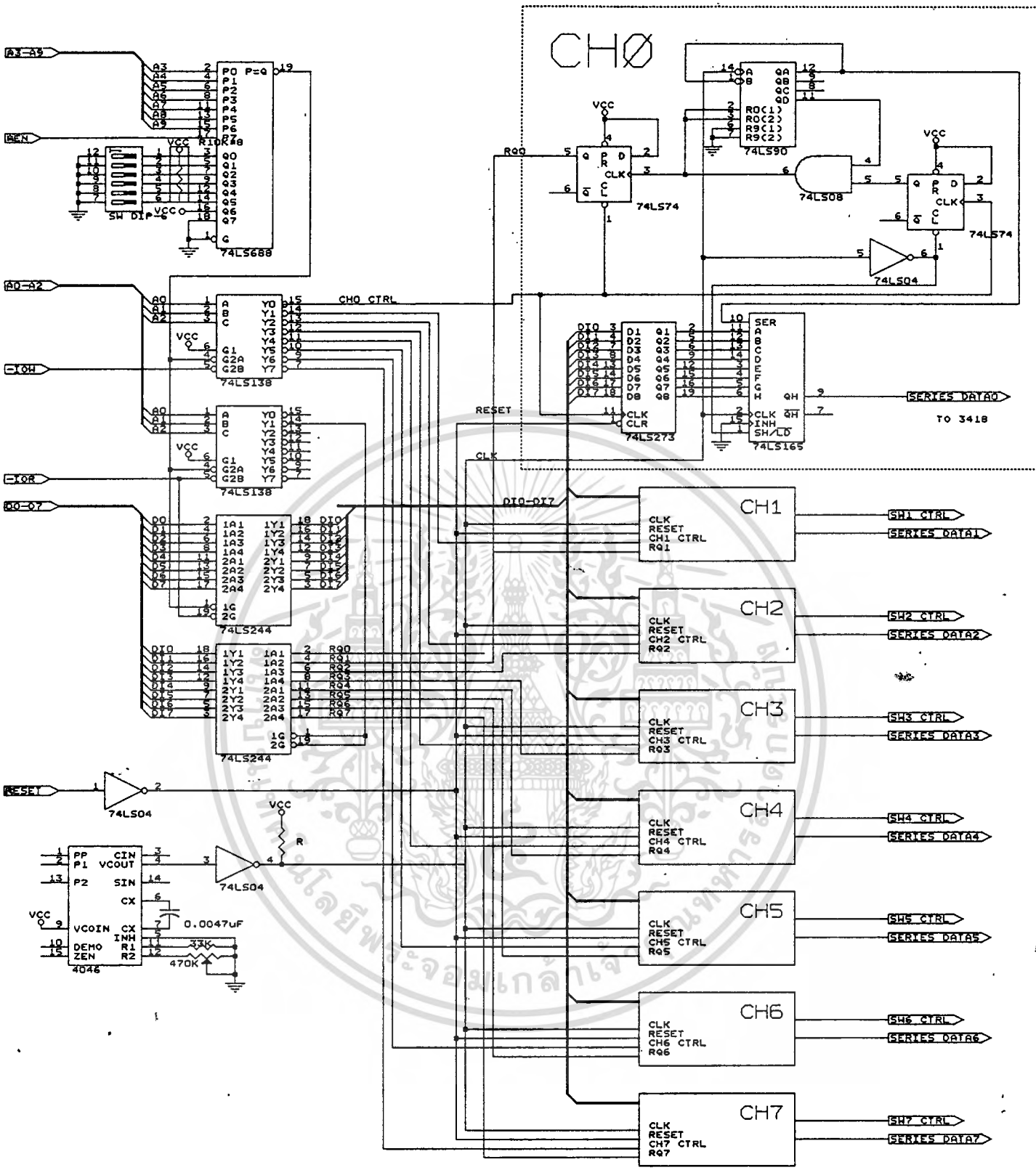
```
}
gotoxy(30,20);
cprintf("SELECT NO.OF RECORD");
gotoxy(28,21);
cprintf("PRESS 'Q' TO RETURN TO MENU");
ch=getch();
if (ch=='q'||ch=='Q')
        return;
ch=ch-'0';
if (ch<0||ch>nfile){
        square(20,2,60,24,CYAN);
        WINDOW(20,8,60,13,BLUE,WHITE);
        cprintf("ERROR SELECT FILE");
        getch();
        return;
}
square(30,3,60,21,WHITE);
remove(fname[ch-1]);
for(i=0;i<(nfile-ch);i++)
        strcpy(fname[ch-1+i],fname[ch+i]);
nfile--;
cprintf("RECORD");
for(i=0;i<nfile;i++){
        gotoxy(25,4+i);
        cprintf("%d.%s\n",i+1,fname[i]);
}
if((file=fopen("recfile","wt"))==NULL)
        printf("ERROR WRITING RECORD fILE\n");
else
        for(i=0;i<nfile;i++)
                fwrite(&fname[i],sizeof(fname[i]),1,file);
fclose(file);
return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void barbox(int x1,int y1,char *opt){
        int x;

        textcolor(WHITE);
        textbackground(BLUE);
        gotoxy(x1,y1);
        for(x=x1;x<x1+2;x++)  putch(0x20);
        cprintf(opt);
        for(x=wherex()+1;x<=43;x++) putch(0x20);
}

void nobarbox(int x1,int y1,char *opt){
        int x;

        textcolor(LIGHTGRAY);
        textbackground(BROWN);
        gotoxy(x1,y1);
        for(x=x1;x<x1+2;x++)  putch(0x20);
        cprintf(opt);
        for(x=wherex()+1;x<=43;x++) putch(0x20);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


A3-A9
U1
P=Q
AEN
S?
R10K
SW DIP-6
74LS688
A0-A2
U2
74LS138
-IOR
D0-D7
U3
74LS373
U4
74LS244
U5
4046
U9A
74LS04
CLOCK
0.0047uF
33K
470K
U6
74LS164
FROM A/D
SERIES DATA
U7A
U7B
74LS74
U8
74LS90
VCC




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# MC3417, MC3517
# MC3418, MC3518

2

## Specifications and Applications Information

CONTINUOUSLY VARIABLE SLOPE DELTA MODULATOR/DEMODULATOR

LASER-TRIMMED INTEGRATED CIRCUIT

### CONTINUOUSLY VARIABLE SLOPE DELTA MODULATOR/DEMODULATOR

Providing a simplified approach to digital speech encoding/decoding, the MC3517/18 series of CVSDs is designed for military secure communication and commercial telephone applications. A single IC provides both encoding and decoding functions.

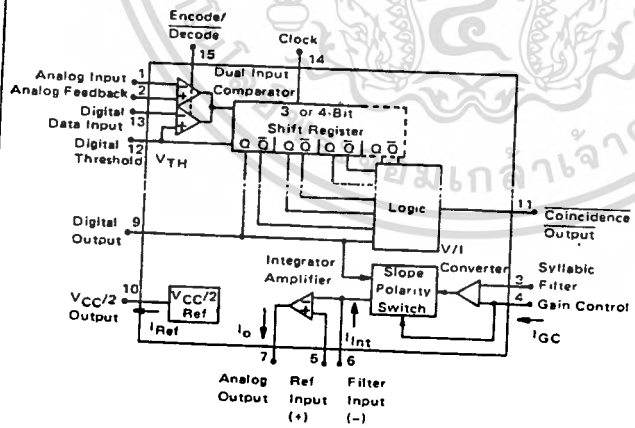
- Encode and Decode Functions on the Same Chip with a Digital Input for Selection
- Utilization of Compatible $I^2L$ – Linear Bipolar Technology
- CMOS Compatible Digital Output
- Digital Input Threshold Selectable ($V_{CC}/2$ reference provided on chip)
- MC3417/MC3517 has a 3-Bit Algorithm (General Communications)
- MC3418/MC3518 has a 4-Bit Algorithm (Commercial Telephone)

**L SUFFIX**
CERAMIC PACKAGE
CASE 620-10

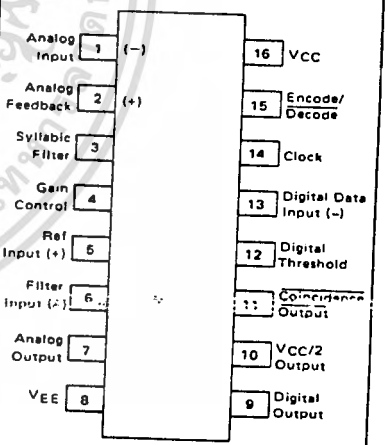
**P SUFFIX**
PLASTIC PACKAGE
CASE 648-08

**DW SUFFIX**
PLASTIC PACKAGE
CASE 751G-01
SO-16L

CVSD BLOCK DIAGRAM




PIN CONNECTIONS




ORDERING INFORMATION

| Device | Package | Temperature Range |
|---|---|---|
| MC3417L | Ceramic DIP | 0°C to +70°C |
| MC3418DW | Plastic SOIC | 0°C to +70°C |
| MC3418L | Ceramic DIP | 0°C to +70°C |
| MC3418P | Plastic DIP | 0°C to +70°C |
| MC3517L | Ceramic DIP | −55°C to +125°C |
| MC3518L | Ceramic DIP | −55°C to +125°C |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## MAXIMUM RATINGS

(All voltages referenced to $V_{EE}$, $T_A$ = 25°C unless otherwise noted.)

| Rating | Symbol | Value | Unit |
|---|---|---|---|
| Power Supply Voltage | $V_{CC}$ | −0.4 to +18 | Vdc |
| Differential Analog Input Voltage | $V_{ID}$ | ±5.0 | Vdc |
| Digital Threshold Voltage | $V_{TH}$ | −0.4 to $V_{CC}$ | Vdc |
| Logic Input Voltage (Clock, Digital Data, Encode/Decode) | $V_{Logic}$ | −0.4 to +18 | Vdc |
| Coincidence Output Voltage | $V_{O(Con)}$ | −0.4 to +18 | Vdc |
| Syllabic Filter Input Voltage | $V_{I(Syl)}$ | −0.4 to $V_{CC}$ | Vdc |
| Gain Control Input Voltage | $V_{I(GC)}$ | −0.4 to $V_{CC}$ | Vdc |
| Reference Input Voltage | $V_{I(Ref)}$ | $V_{CC}/2$ − 1.0 to $V_{CC}$ | Vdc |
| $V_{CC}/2$ Output Current | $I_{Ref}$ | −25 | mA |

## ELECTRICAL CHARACTERISTICS

($V_{CC}$ = 12 V, $V_{EE}$ = Gnd, $T_A$ = 0°C to +70°C for MC3417/18, $T_A$ = −55°C to +125°C for MC3517/18 unless otherwise noted.)

| Characteristic | Symbol | MC3417/MC3517 Min | MC3417/MC3517 Typ | MC3417/MC3517 Max | MC3418/MC3518 Min | MC3418/MC3518 Typ | MC3418/MC3518 Max | Unit |
|---|---|---|---|---|---|---|---|---|
| Power Supply Voltage Range (Figure 1) | $V_{CCR}$ | 4.75 | 12 | 16.5 | 4.75 | 12 | 16.5 | Vdc |
| Power Supply Current (Figure 1) (@ Idle Channel) | $I_{CC}$ | | | | | | | mA |
| ($V_{CC}$ = 5.0 V, All except MC3418P,DW) | | — | 3.7 | 5.0 | — | 3.7 | 5.0 | |
| ($V_{CC}$ = 5.0 V, MC3418P,DW) | | — | — | — | — | 3.7 | 5.5 | |
| ($V_{CC}$ = 15 V, All except MC3418P,DW) | | — | 6.0 | 10 | — | 6.0 | 10 | |
| ($V_{CC}$ = 15 V, MC3418P,DW) | | — | — | — | — | 6.0 | 11 | |
| Gain Control Current Range (Figure 2) | $I_{GCR}$ | 0.002 | — | 3.0 | 0.002 | — | 3.0 | mA |
| Analog Comparator Input Range (Pins 1 and 2) ($4.75 V \le V_{CC} \le 16.5 V$) | $V_I$ | 1.3 | — | $V_{CC}-1.3$ | 1.3 | — | $V_{CC}-1.3$ | Vdc |
| Analog Output Range (Pin 7) ($4.75 V \le V_{CC} \le 16.5 V$, $I_O$ = ±5.0 mA) | $V_O$ | 1.3 | — | $V_{CC}-1.3$ | 1.3 | — | $V_{CC}-1.3$ | Vdc |
| Input Bias Currents (Figure 3) (Comparator in Active Region) | $I_{IB}$ | | | | | | | μA |
| Analog Input (I1) | | — | 0.5 | 1.5 | — | 0.25 | 1.0 | |
| Analog Feedback (I2) | | — | 0.5 | 1.5 | — | 0.25 | 1.0 | |
| Syllabic Filter Input (I3) | | — | 0.06 | 0.5 | — | 0.06 | 0.3 | |
| Reference Input (I5) | | — | −0.06 | −0.5 | — | −0.06 | −0.3 | |
| Input Offset Current (Comparator in Active Region) | $I_{IO}$ | | | | | | | μA |
| Analog Input/Analog Feedback \|I1 − I2\| — Figure 3 | | — | 0.15 | 0.6 | — | 0.05 | 0.4 | |
| Integrator Amplifier \|I5 − I6\| — Figure 4 | | — | 0.02 | 0.2 | — | 0.01 | 0.1 | |
| Input Offset Voltage V/I Converter (Pins 3 and 4) — Figure 5 | $V_{IO}$ | — | 2.0 | 6.0 | — | 2.0 | 6.0 | mV |
| Transconductance | gm | | | | | | | mA/mV |
| V/I Converter, 0 to 3.0 mA | | 0.1 | 0.3 | — | 0.1 | 0.3 | — | |
| Integrator Amplifier, 0 to ±5.0 mA Load | | 1.0 | 10 | — | 1.0 | 10 | — | |
| Propagation Delay Times (Note 1) | | | | | | | | μs |
| Clock Trigger to Digital Output | $t_{PLH}$ | — | 1.0 | 2.5 | — | 1.0 | 2.5 | |
| ($C_L$ = 25 pF to Gnd) | $t_{PHL}$ | — | 0.8 | 2.5 | — | 0.8 | 2.5 | |
| Clock Trigger to Coincidence Output | $t_{PLH}$ | — | 1.0 | 3.0 | — | 1.0 | 3.0 | |
| ($C_L$ = 25 pF to Gnd) ($R_L$ = 4.0 kΩ to $V_{CC}$) | $t_{PHL}$ | — | 0.8 | 2.0 | — | 0.8 | 2.0 | |
| Coincidence Output Voltage — Low Logic State ($I_{OL(Con)}$ = 3.0 mA) | $V_{OL(Con)}$ | — | 0.12 | 0.25 | — | 0.12 | 0.25 | Vdc |
| Coincidence Output Leakage Current — High Logic State ($V_{OH}$ = 15 V, 0°C ≤ $T_A$ ≤ 70°C) | $I_{OH(Con)}$ | — | 0.01 | 0.5 | — | 0.01 | 0.5 | μA |

NOTE 1. All propagation delay times measured 50% to 50% from the negative going (from $V_{CC}$ to +0.4 V) edge of the clock.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2

**ELECTRICAL CHARACTERISTICS (continued)**

| Characteristic | Symbol | MC3417/MC3517 Min | MC3417/MC3517 Typ | MC3417/MC3517 Max | MC3418/MC3518 Min | MC3418/MC3518 Typ | MC3418/MC3518 Max | Unit |
|---|---|---|---|---|---|---|---|---|
| Applied Digital Threshold Voltage Range (Pin 12) | $V_{TH}$ | +1.2 | — | $V_{CC}$ – 2.0 | +1.2 | — | $V_{CC}$ – 2.0 | Vdc |
| Digital Threshold Input Current ($1.2\ V \leq V_{th} \leq V_{CC} - 2.0\ V$) | $I_{I(th)}$ | | | | | | | μA |
| ($V_{IL}$ applied to Pins 13, 14 and 15) | | — | — | 5.0 | — | — | 5.0 | |
| ($V_{IH}$ applied to Pins 13, 14 and 15) | | — | –10 | –50 | — | –10 | –50 | |
| Maximum Integrator Amplifier Output Current | $I_O$ | ±5.0 | — | — | ±5.0 | — | — | mA |
| $V_{CC}/2$ Generator Maximum Output Current (Source only) | $I_{Ref}$ | +10 | — | — | +10 | — | — | mA |
| $V_{CC}/2$ Generator Output Impedance (0 to +10 mA) | $z_{Ref}$ | — | 3.0 | 6.0 | — | 3.0 | 6.0 | Ω |
| $V_{CC}/2$ Generator Tolerance ($4.75\ V \leq V_{CC} \leq 16.5\ V$) | $e_r$ | — | — | ±3.5 | — | — | ±3.5 | % |
| Logic Input Voltage (Pins 13, 14 and 15) | | | | | | | | Vdc |
| Low Logic State | $V_{IL}$ | Gnd | — | $V_{th}$ – 0.4 | Gnd | — | $V_{th}$ – 0.4 | |
| High Logic State | $V_{IH}$ | $V_{th}$ + 0.4 | — | 18 | $V_{th}$ + 0.4 | — | 18 | |
| Dynamic Total Loop Offset Voltage (Note 2) — Figures 3, 4 and 5 | $\Sigma V_{offset}$ | | | | | | | mV |
| $I_{GC}$ = 12 μA, $V_{CC}$ = 12 V | | | | | | | | |
| $T_A$ = 25°C (All except 3418P,DW) | | — | — | — | — | ±0.5 | ±1.5 | |
| (MC3418P,DW) | | — | — | — | — | ±0.5 | ±3.0 | |
| $0°C \leq T_A \leq +70°C$ (MC3417/18L) | | — | — | — | — | ±0.75 | ±2.3 | |
| (MC3418P,DW) | | — | — | — | — | ±0.75 | ±3.8 | |
| $-55°C \leq T_A \leq +125°C$ (MC3517/18) | | — | — | — | — | ±1.5 | ±4.0 | |
| $I_{GC}$ = 33 μA, $V_{CC}$ = 12 V | | | | | | | | |
| $T_A$ = 25°C | | — | ±2.5 | ±5.0 | — | — | — | |
| $0°C \leq T_A \leq +70°C$ (MC3417/18) | | — | ±3.0 | ±7.5 | — | — | — | |
| $-55°C \leq T_A \leq +125°C$ (MC3517/18) | | — | ±4.5 | ±10 | — | — | — | |
| $I_{GC}$ = 12 μA, $V_{CC}$ = 5.0 V | | | | | | | | |
| $T_A$ = 25°C (All except MC3418P,DW) | | — | — | — | — | ±1.0 | ±2.0 | |
| (MC3418P,DW) | | — | — | — | — | ±1.0 | ±3.5 | |
| $0°C \leq T_A \leq +70°C$ (MC3417/18L) | | — | — | — | — | ±1.3 | ±2.8 | |
| (MC3418P,DW) | | — | — | — | — | ±1.3 | ±4.3 | |
| $-55°C \leq T_A \leq +125°C$ (MC3517/18) | | — | — | — | — | ±2.5 | ±5.0 | |
| $I_{GC}$ = 33 μA, $V_{CC}$ = 5.0 V | | | | | | | | |
| $T_A$ = 25°C | | — | ±4.0 | ±6.0 | — | — | — | |
| $0°C \leq T_A \leq +70°C$ (MC3417/18) | | — | ±4.5 | ±8.0 | — | — | — | |
| $-55°C \leq T_A \leq +125°C$ (MC3517/18) | | — | ±5.5 | ±10 | — | — | — | |
| Digital Output Voltage | | | | | | | | Vdc |
| ($I_{OL}$ = 3.6 mA) | $V_{OL}$ | — | 0.1 | 0.4 | — | 0.1 | 0.4 | |
| ($I_{OH}$ = –0.35 mA) | $V_{OH}$ | $V_{CC}$ – 1.0 | $V_{CC}$ – 0.2 | — | $V_{CC}$ – 1.0 | $V_{CC}$ – 0.2 | — | |
| Syllabic Filter Applied Voltage (Pin 3) (Figure 2) | $V_{I(Syl)}$ | +3.2 | — | $V_{CC}$ | +3.2 | — | $V_{CC}$ | Vdc |
| Integrating Current (Figure 2) | $I_{Int}$ | | | | | | | |
| ($I_{GC}$ = 12 μA) | | 8.0 | 10 | 12 | 8.0 | 10 | 12 | μA |
| ($I_{GC}$ = 1.5 mA) (All except 3418P,DW) | | 1.45 | 1.5 | 1.55 | 1.45 | 1.5 | 1.55 | mA |
| (MC3418P,DW) | | — | — | — | 1.42 | 1.5 | 1.58 | mA |
| ($I_{GC}$ = 3.0 mA) | | 2.75 | 3.0 | 3.25 | 2.75 | 3.0 | 3.25 | mA |
| Dynamic Integrating Current Match ($I_{GC}$ = 1.5 mA) Figure 6 | $V_{O(Ave)}$ | | | | | | | mV |
| (All except MC3418P,DW) | | — | ±100 | ±250 | — | ±100 | ±250 | |
| (MC3418P,DW) | | — | — | — | — | ±100 | ±280 | |
| Input Current — High Logic State ($V_{IH}$ = 18 V) | $I_{IH}$ | | | | | | | μA |
| Digital Data Input | | — | — | +5.0 | — | — | +5.0 | |
| Clock Input | | — | — | +5.0 | — | — | +5.0 | |
| Encode/Decode Input | | — | — | +5.0 | — | — | +5.0 | |
| Input Current — Low Logic State ($V_{IL}$ = 0 V) | $I_{IL}$ | | | | | | | μA |
| Digital Data Input | | — | — | –10 | — | — | –10 | |
| Clock Input | | — | — | –360 | — | — | –360 | |
| Encode/Decode Input | | — | — | –36 | — | — | –36 | |
| Clock Input, $V_{IL}$ = 0.4 V | | — | — | –72 | — | — | –72 | |

NOTE 2. Dynamic total loop offset ($\Sigma V_{offset}$) equals $V_{IO}$ (comparator) (Figure 3) minus $V_{IOX}$ (Figure 5). The input offset voltages of the analog comparator and of the integrator amplifier include the effects of input offset current through the input resistors. The slope polarity switch current mismatch appears as an average voltage across the 10 k integrator resistor. For the MC3417/MC3517, the clock frequency is 16 kHz. For the MC3418/MC3518, the clock frequency is 32 kHz. Idle channel performance is guaranteed if this dynamic total loop offset is less than one-half of the change in integrator output voltage during one clock cycle (ramp step size). Laser trimming is used to insure good idle channel performance.

## DEFINITIONS AND FUNCTION OF PINS

**Pin 1 — Analog Input**

This is the analog comparator inverting input where the voice signal is applied. It may be ac or dc coupled depending on the application. If the voice signal is to be level shifted to the internal reference voltage, then a bias resistor between Pins 1 and 10 is used. The resistor is used to establish the reference as the new dc average of the ac coupled signal. The analog comparator was designed for low hysteresis (typically less than 0.1 mV) and high gain (typically 70 dB).

**Pin 2 — Analog Feedback**

This is the noninverting input to the analog signal comparator within the IC. In an encoder application it should be connected to the analog output of the encoder circuit. This may be Pin 7 or a low pass filter output connected to Pin 7. In a decode circuit Pin 2 is not used and may be tied to $V_{CC}/2$ on Pin 10, ground or left open.

The analog input comparator has bias currents of 1.5 $\mu$A max, thus the driving impedances of Pins 1 and 2 should be equal to avoid disturbing the idle channel characteristics of the encoder.

**Pin 3 — Syllabic Filter**

This is the point at which the syllabic filter voltage is returned to the IC in order to control the integrator step size. It is an NPN input to an op amp. The syllabic filter consists of an RC network between Pins 11 and 3. Typical time constant values of 6.0 ms to 50 ms are used in voice codecs.

**Pin 4 — Gain Control Input**

The syllabic filter voltage appears across $C_S$ of the syllabic filter and is the voltage between $V_{CC}$ and Pin 3. The active voltage to current (V−I) converter drives Pin 4 to the same voltage at a slew rate of typically 0.5 V/$\mu$s. Thus the current injected into Pin 4 ($I_{GC}$) is the syllabic filter voltage divided by the $R_x$ resistance. Figure 7 shows the relationship between $I_{GC}$ (x-axis) and the integrating current, $I_{Int}$ (y-axis). The discrepancy, which is most significant at very low currents, is due to circuitry within the slope polarity switch which enables trimming to a low total loop offset. The $R_x$ resistor is then varied to adjust the loop gain of the codec, but should be no larger than 5.0 kΩ to maintain stability.

**Pin 5 — Reference Input**

This pin is the noninverting input of the integrator amplifier. It is used to reference the dc level of the output signal. In an encoder circuit it must reference the same voltage as Pin 1 and is tied to Pin 10.

**Pin 6 — Filter Input**

This inverting op amp input is used to connect the integrator external components. The integrating current ($I_{Int}$) flows into Pin 6 when the analog input (Pin 1) is high with respect to the analog feedback (Pin 2) in the encode mode or when the digital data input (Pin 13) is high in the decode mode. For the opposite states, $I_{Int}$ flows out of Pin 6. Single integration systems require a capacitor and resistor between Pins 6 and 7. Multipole configurations will have different circuitry. The resistance between Pins 6 and 7 should always be between 8.0 kΩ and 13 kΩ to maintain good idle channel characteristics.

**Pin 7 — Analog Output**

This is the integrator op amp output. It is capable of driving a 600-ohm load referenced to $V_{CC}/2$ to +6.0 dBm and can otherwise be treated as an op amp output. Pins 5, 6, and 7 provide full access to the integrator op emp for designing integration filter networks. The slew rate of the internally compensated integrator op amp is typically 0.5 V/$\mu$s. Pin 7 output is current limited for both polarities of current flow at typically 30 mA.

**Pin 8 — $V_{EE}$**

The circuit is designed to work in either single or dual power supply applications. Pin 8 is always connected to the most negative supply.

**Pin 9 — Digital Output**

The digital output provides the results of the delta modulator's conversion. It swings between $V_{CC}$ and $V_{EE}$ and is CMOS or TTL compatible. Pin 9 is inverting with respect to Pin 1 and non-inverting with respect to Pin 2. It is clocked on the falling edge of Pin 14. The typical 10% to 90% rise and fall times are 250 ns and 50 ns respectively for $V_{CC}$ = 12 V and $C_L$ = 25 pF to ground.

**Pin 10 — $V_{CC}/2$ Output**

An internal low impedance mid-supply reference is provided for use of the MC3417/18 in single supply applications. The internal regulator is a current source and must be loaded with a resistor to insure its sinking capability. If a +6.0 dBmo signal is expected across a 600 ohm input bias resistor, then Pin 10 must sink 2.2 V/600 Ω = 3.66 mA. This is only possible if Pin 10 sources 3.66 mA into a resistor normally and will source only the difference under peak-load. The reference load resistor is chosen accordingly. A 0.1 $\mu$F bypass capacitor from Pin 10 to $V_{EE}$ is also recommended. The $V_{CC}/2$ reference is capable of sourcing 10 mA and can be used as a reference elsewhere in the system circuitry.

**Pin 11 — $\overline{\text{Coincidence Output}}$**

The duty cycle of this pin is proportional to the voltage across $C_S$. The coincidence output will be low whenever the content of the internal shift register is all 1s or all 0s. In the MC3417 the register is 3 bits long while the MC3418 contains a 4 bit register. Pin 11 is an open collector of an NPN device and requires a pull-up resistor.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2

If the syllabic filter is to have equal charge and discharge time constants, the value of $R_P$ should be much less than $R_S$. In systems requiring different charge and discharge constants, the charging constant is $R_S C_S$ while the decaying constant is $(R_S + R_P)C_S$. Thus longer decays are easily achievable. The NPN device should not be required to sink more than 3.0 mA in any configuration. The typical 10% to 90% rise and fall times are 200 ns and 100 ns respectively for $R_L$ = 4.0 kΩ to +12 V and $C_L$ = 25 pF to ground.

**Pin 12 — Digital Threshold**

This input sets the switching threshold for Pins 13, 14, and 15. It is intended to aid in interfacing different logic families without external parts. Often it is connected to the $V_{CC}/2$ reference for CMOS interface or can be biased two diode drops above $V_{EE}$ for TTL interface.

**Pin 13 — Digital Data Input**

In a decode application, the digital data stream is applied to Pin 13. In an encoder it may be unused or may be used to transmit signaling message under the control of Pin 15. It is an inverting input with respect to Pin 9. When Pins 9 and 13 are connected, a toggle flip-flop is formed and a forced idle channel pattern can be transmitted. The digital data input level should be maintained for 0.5 μs before and after the clock trigger for proper clocking.

**Pin 14 — Clock Input**

The clock input determines the data rate of the codec circuit. A 32K bit rate requires a 32 kHz clock. The switching threshold of the clock input is set by Pin 12. The shift register circuit toggles on the falling edge of the clock input. The minimum width for a positive-going pulse on the clock input is 300 ns, whereas for a negative-going pulse, it is 900 ns.

**Pin 15 — Encode/$\overline{\text{Decode}}$**

This pin controls the connection of the analog input comparator and the digital input comparator to the internal shift register. If high, the result of the analog comparison will be clocked into the register on the falling edge at Pin 14. If low, the digital input state will be entered. This allows use of the IC as an encoder/decoder or simplex codec without external parts. Furthermore, it allows non-voice patterns to be forced onto the transmission line through Pin 13 in an encoder.

**Pin 16 — $V_{CC}$**

The power supply range is from 4.75 to 16.5 volts between Pin $V_{CC}$ and $V_{EE}$.

FIGURE 1 – POWER SUPPLY CURRENT

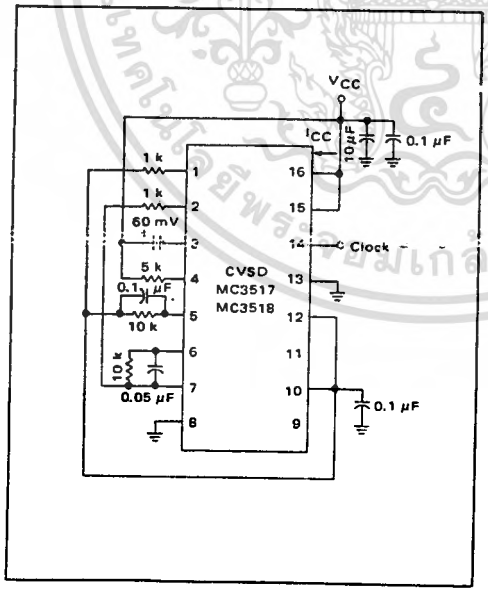
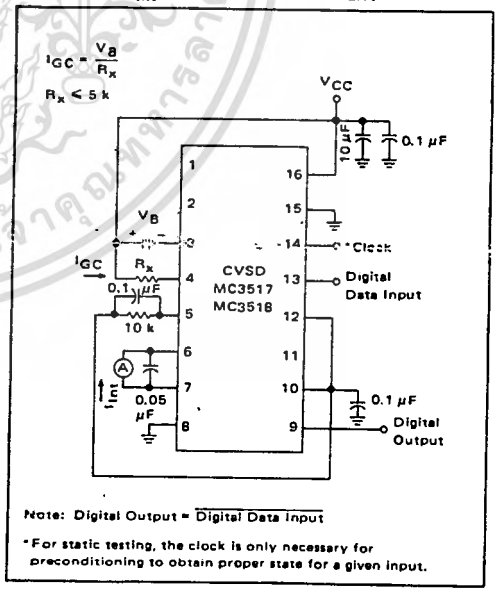


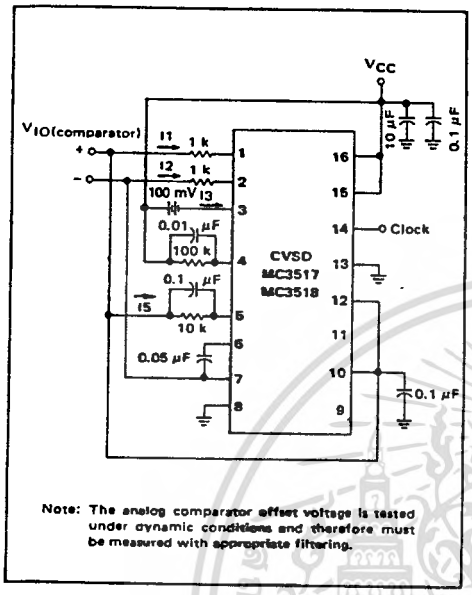

FIGURE 2 – $I_{GCR}$, GAIN CONTROL RANGE and $I_{Int}$ – INTEGRATING CURRENT




Note: Digital Output = $\overline{\text{Digital Data Input}}$

*For static testing, the clock is only necessary for preconditioning to obtain proper state for a given input.

FIGURE 3 – INPUT BIAS CURRENTS, ANALOG COMPARATOR OFFSET VOLTAGE AND CURRENT




Note: The analog comparator offset voltage is tested under dynamic conditions and therefore must be measured with appropriate filtering.

FIGURE 4 – INTEGRATOR AMPLIFIER OFFSET VOLTAGE AND CURRENT

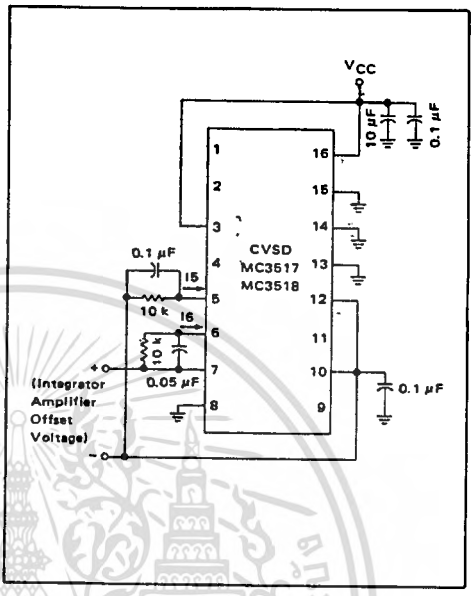
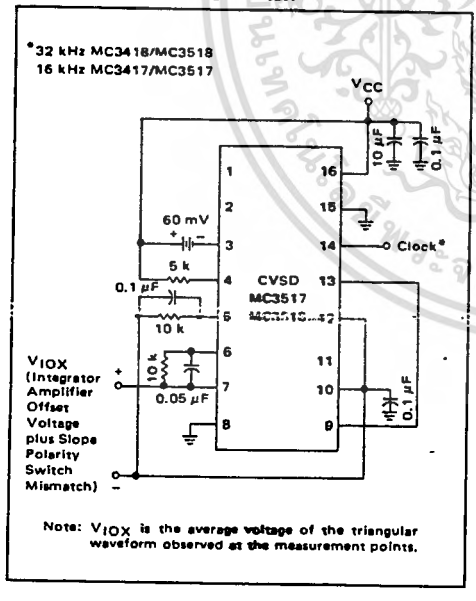
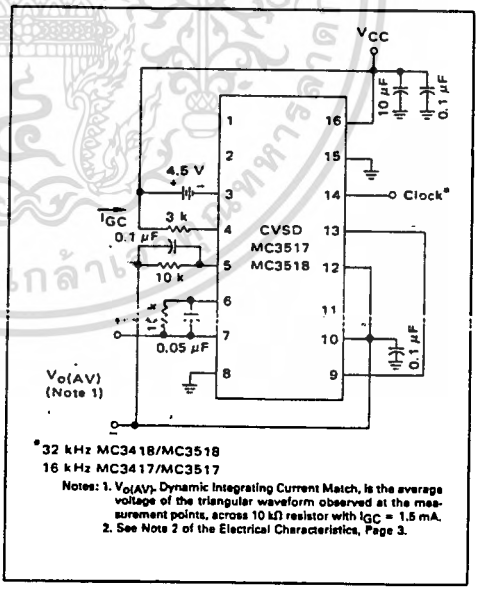



FIGURE 5 – V/I CONVERTER OFFSET VOLTAGE, $V_{IO}$ and $V_{IOX}$




Note: $V_{IOX}$ is the average voltage of the triangular waveform observed at the measurement points.

FIGURE 6 – DYNAMIC INTEGRATING CURRENT MATCH




*32 kHz MC3418/MC3518
16 kHz MC3417/MC3517

Notes: 1. $V_{O(AV)}$, Dynamic Integrating Current Match, is the average voltage of the triangular waveform observed at the measurement points, across 10 kΩ resistor with $I_{GC}$ = 1.5 mA.
2. See Note 2 of the Electrical Characteristics, Page 3.

2

## TYPICAL PERFORMANCE CURVES

FIGURE 7 – TYPICAL $I_{Int}$ versus $I_{GC}$ (Mean ± 2σ)

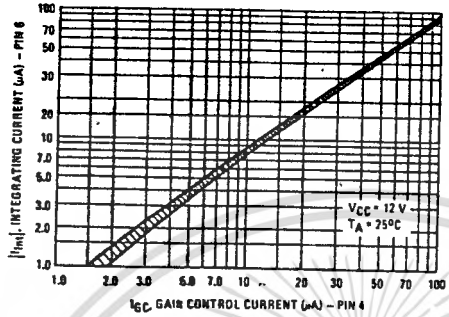



FIGURE 8 – NORMALIZED DYNAMIC INTEGRATING CURRENT MATCH versus $V_{CC}$

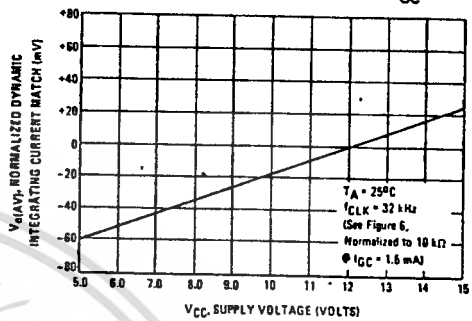



FIGURE 9 – NORMALIZED DYNAMIC INTEGRATING CURRENT MATCH versus CLOCK FREQUENCY

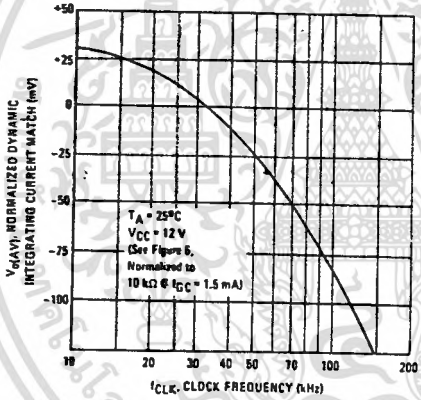



FIGURE 10 – DYNAMIC TOTAL LOOP OFFSET versus CLOCK FREQUENCY

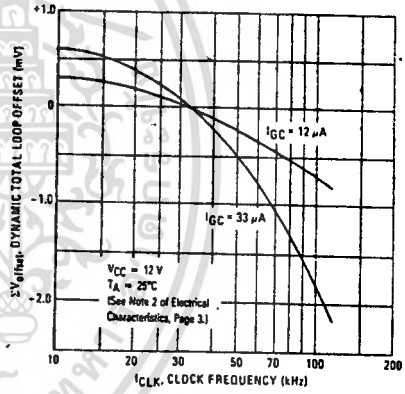



FIGURE 11 – BLOCK DIAGRAM OF THE CVSD ENCODER

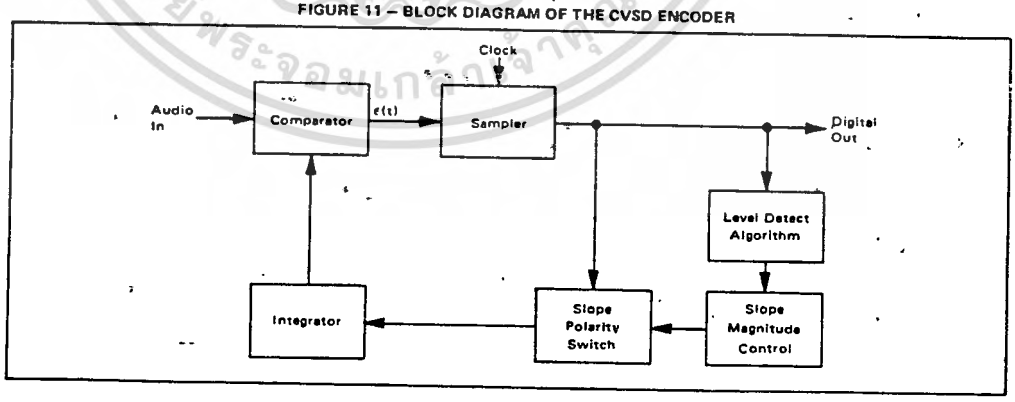


FIGURE 12 – CVSD WAVEFORMS

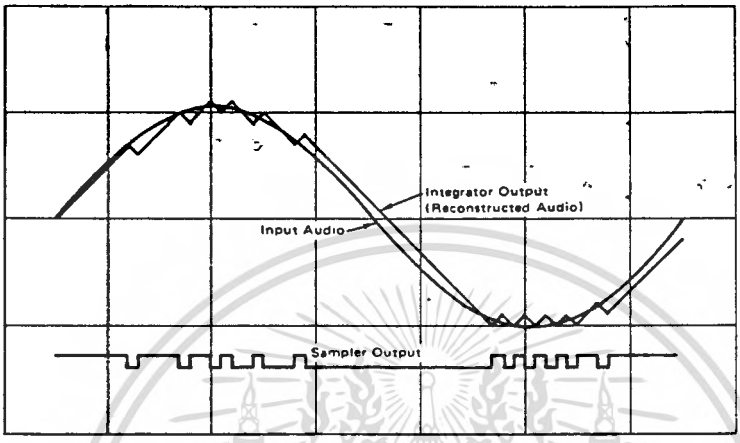
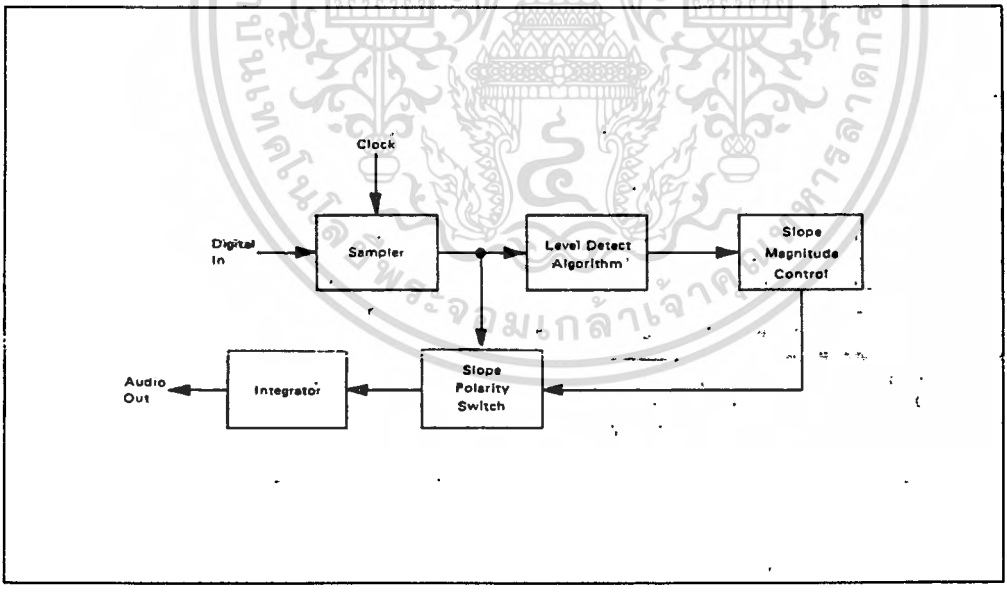
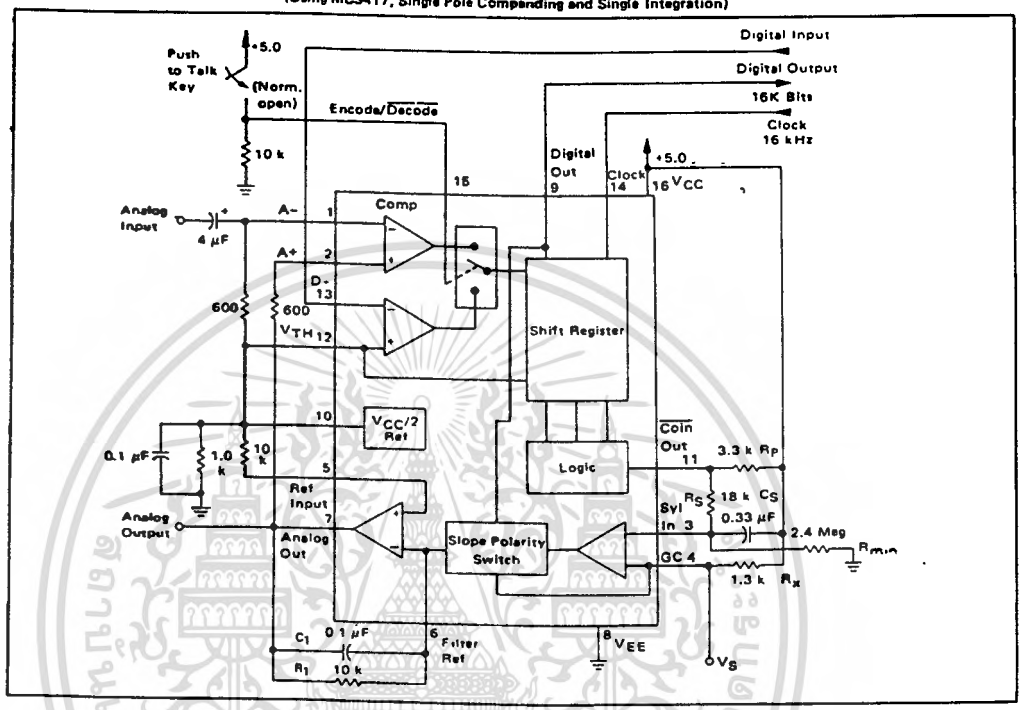



FIGURE 13 – BLOCK DIAGRAM OF THE CVSD DECODER



2

FIGURE 14 – 16 kHz SIMPLEX VOICE CODEC
(Using MC3417, Single Pole Companding and Single Integration)




## CIRCUIT DESCRIPTION

The continuously variable slope delta modulator (CVSD) is a simple alternative to more complex conventional conversion techniques in systems requiring digital communication of analog signals. The human voice is analog, but digital transmission of any signal over great distance is attractive. Signal/noise ratios do not vary with distance in digital transmission and multiplexing, switching and repeating hardware is more economical and easier to design. However, instrumentation A to D converters do not meet the communications requirements. The CVSD A to D is well suited to the requirements of digital communications and is an economically efficient means of digitizing analog inputs for transmission.

### The Delta Modulator

The innermost control loop of a CVSD converter is a simple delta modulator. A block diagram CVSD Encoder is shown in Figure 11. A delta modulator consists of a comparator in the forward path and an integrator in the feedback path of a simple control loop. The inputs to the comparator are the input analog signal and the integrator output. The comparator output reflects the sign of the difference between the input voltage and the integrator output. That sign bit is the digital output and also controls the direction of ramp in the integrator. The comparator is normally clocked so as to produce a synchronous and band limited digital bit stream.

If the clocked serial bit stream is transmitted, received, and delivered to a similar integrator at a remote point, the remote integrator output is a copy of the transmitting control loop integrator output. To the extent that the integrator at the transmitting locations tracks the input signal, the remote receiver reproduces the input signal. Low pass filtering at the receiver output will eliminate most of the quantizing noise, if the clock rate of the bit stream is an octave or more above the bandwidth of the input signal. Voice bandwidth is 4 kHz and clock rates from 8 k and up are possible. Thus the delta modulator digitizes and transmits the analog input to a remote receiver. The serial, unframed nature of the data is ideal for communications networks. With no input at the transmitter, a continuous one zero alternation is transmitted. If the two integrators are made leaky, then during any loss of contact the receiver output decays to

## CIRCUIT DESCRIPTION (continued)

zero and receive restart begins without framing when the receiver reacquires. Similarly a delta modulator is tolerant of sporadic bit errors. Figure 12 shows the delta modulator waveforms while Figure 13 shows the corresponding CVSD decoder block diagram.

**The Companding Algorithm**

The fundamental advantages of the delta modulator are its simplicity and the serial format of its output. Its limitations are its ability to accurately convert the input within a limited digital bit rate. The analog input must be band limited and amplitude limited. The frequency limitations are governed by the nyquist rate while the amplitude capabilities are set by the gain of the integrator.

The frequency limits are bounded on the upper end; that is, for any input bandwidth there exists a clock frequency larger than that bandwidth which will transmit the signal with a specific noise level. However, the amplitude limits are bounded on both upper and lower ends. For a signal level, one specific gain will achieve an optimum noise level. Unfortunately, the basic delta modulator has a small dynamic range over which the noise level is constant.

The continuously variable slope circuitry provides increased dynamic range by adjusting the gain of the integrator. For a given clock frequency and input bandwidth the additional circuitry increases the delta modulator's dynamic range. External to the basic delta modulator is an algorithm which monitors the past few outputs of the delta modulator in a simple shift register. The register is 3 or 4 bits long depending on the application. The accepted CVSD algorithm simply monitors the contents of the shift register and indicates if it contains all 1s or 0s. This condition is called coincidence. When it occurs, it indicates that the gain of the integrator is too small. The coincidence output charges a single pole low pass filter. The voltage output of this syllabic filter controls the integrator gain through a pulse amplitude modulator whose other input is the sign bit or up/down control.

The simplicity of the all ones, all zeros algorithm should not be taken lightly. Many other control algorithms using the shift register have been tried. The key to the accepted algorithm is that it provides a measure of the average power or level of the input signal. Other techniques provide more instantaneous information about the shape of the input curve. The purpose of the algorithm is to control the gain of the integrator and to increase the dynamic range. Thus a measure of the average input level is what is needed.

The algorithm is repeated in the receiver and thus the level data is recovered in the receiver. Because the algorithm only operates on the past serial data, it changes the nature of the bit stream without changing the channel bit rate.

The effect of the algorithm is to compand the input signal. If a CVSD encoder is played into a basic delta modulator, the output of the delta modulator will reflect the shape of the input signal but all of the output will be at an equal level. Thus the algorithm at the output is needed to restore the level variations. The bit stream in the channel is as if it were from a standard delta modulator with a constant level input.

The delta modulator encoder with the CVSD algorithm provides an efficient method for digitizing a voice input in a manner which is especially convenient for digital communciations requirements.

## APPLICATIONS INFORMATION

### CVSD DESIGN CONSIDERATIONS

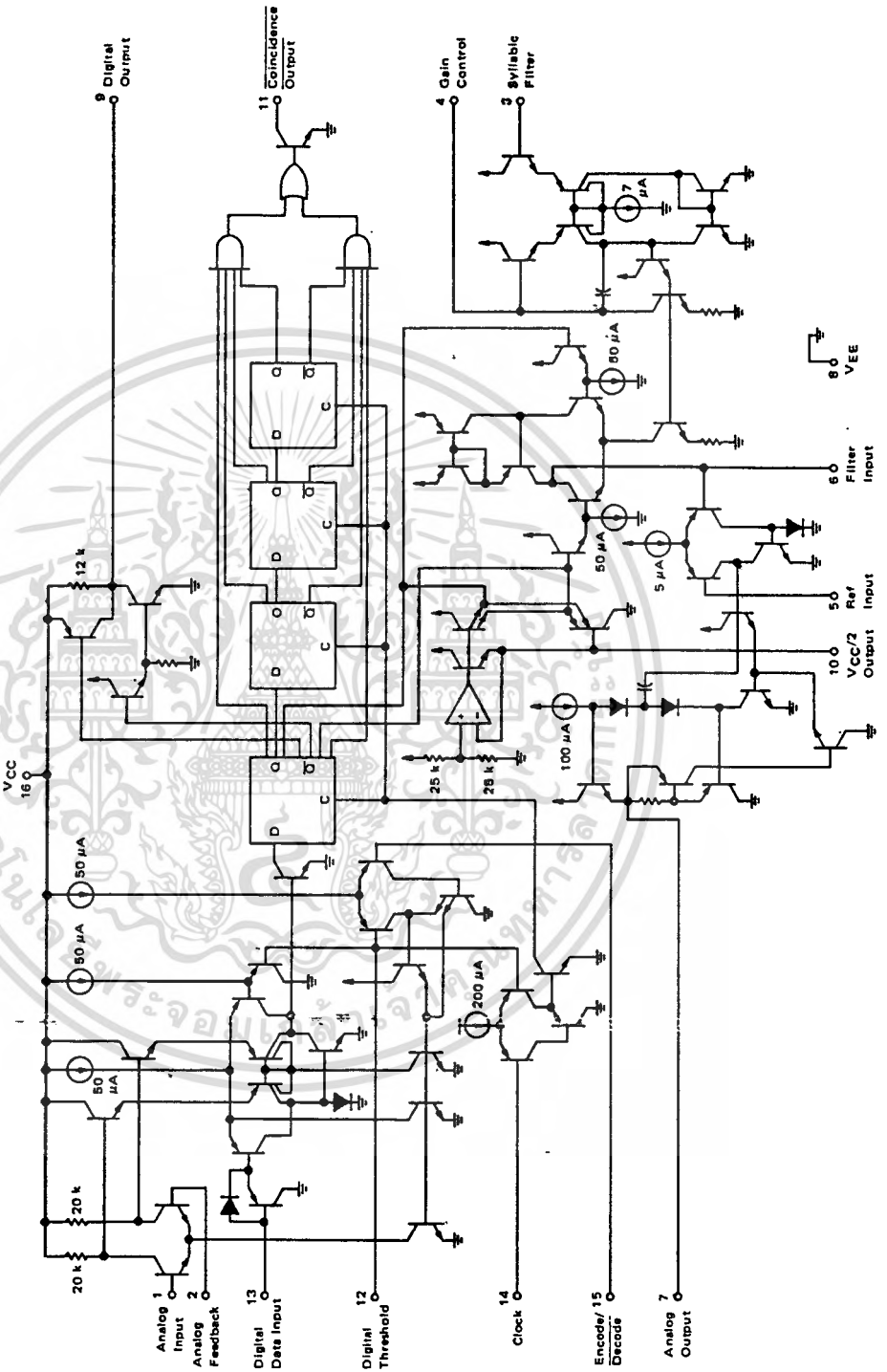
A simple CVSD encoder using the MC3417 or MC3418 is shown in Figure 14. These ICs are general purpose CVSD building blocks which allow the system designer to tailor the encoder's transmission characteristics to the application. Thus, the achievable transmission capabilities are constrained by the fundamental limitations of delta modulation and the design of encoder parameters. The performance is not dictated by the internal configuration of the MC3417 and MC3418. There are seven design considerations involved in designing these basic CVSD building blocks into a specific codec application, and they are as follows:

1. Selection of clock rate
2. Required number of shift register bits
3. Selection of loop gain
4. Selection of minimum step size
5. Design of integration filter transfer function
6. Design of syllabic filter transfer function
7. Design of low pass filter at the receiver

The circuit in Figure 14 is the most basic CVSD circuit possible. For many applications in secure radio or other intelligible voice channel requirements, it is entirely sufficient. In this circuit, items 5 and 6 are reduced to their simplest form. The syllabic and integration filters are both single pole networks. The selection of items 1 through 4 govern the codec performance.

2

CVSD CIRCUIT SCHEMATIC



## CVSD DESIGN CONSIDERATIONS (continued)

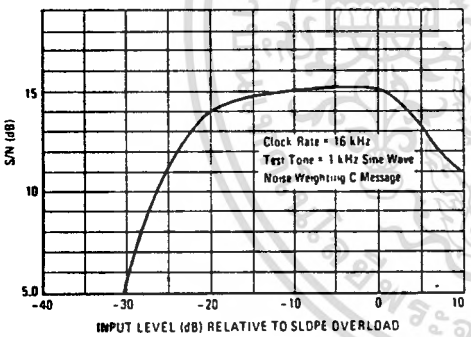
### Layout Considerations

Care should be exercised to isolate all digital signal paths (Pins 9, 11, 13, and 14) from analog signal paths (Pins 1–7 and 10) in order to achieve proper idle channel performance.

### Clock Rate

With minor modifications the circuit in Figure 14 may be operated anywhere from 9.6 kHz to 64 kHz clock rates. Obviously the higher the clock rate the higher the S/N performance. The circuit in Figure 14 typically produces the S/N performance shown in Figure 15. The selection of clock rate is usually dictated by the bandwidth of the transmission medium. Voice bandwidth systems will require no higher than 9600 Hz. Some radio systems will allow 12 kHz. Private 4-wire telephone systems are often operated at 16 kHz and commercial telephone performance can be achieved at 32K bits and above. Other codecs may use bit rates up to 200K bits/sec.

FIGURE 15 – SIGNAL-TO-NOISE PERFORMANCE OF MC3417 WITH SINGLE INTEGRATION, SINGLE-POLE AND COMPANDING AT 16K BITS – TYPICAL




### Shift Register Length (Algorithm)

The MC3417 has a three-bit algorithm and the MC3418 has a four-bit algorithm. For clock rates of 16 kHz and below, the 3-bit algorithm is well suited. For 32 kHz and higher clock rates, the 4-bit system is preferred. Since the algorithm records a fixed past history of the input signal, a longer shift register is required to obtain the same internal hsitory. At 16 bits and below, the 4-bit algorithm will produce a slightly wider dynamic range at the expense of level change response. Basically the MC3417 is designed for low bit rate systems and the MC3418 is intended for high performance, high bit rate system. At bit rates above 64K bits either part will work well.

### Selection of Loop Gain

The gain of the circuit in Figure 14 is set by resistor $R_X$. $R_X$ must be selected to provide the proper integrator step size for high level signals such that the companding ratio does not exceed about 25%. The companding ratio is the active low duty cycle of the coincidence output on Pin 11 of the codec circuit. Thus the system gain is dependent on:

1. The maximum level and frequency of the input signal.
2. The transfer function of the integration filter.

For voice codecs the typical input signal is taken to be a sine wave at 1 kHz of 0 dBmo level. In practice, the useful dynamic range extends about 6 dB above the design level. In any system the companding ratio should not exceed 30%.

To calculate the required step size current, we must describe the transfer characteristics of the integration filter. In the basic circuit of Figure 14, a single pole of 160 Hz is used.

$$R_1 = 10\ k\Omega,\ C_1 = 0.1\ \mu F$$

$$\frac{V_o}{I_i} = \frac{1}{C(S + 1/RC)} \equiv \frac{K}{S + \omega_o}$$

$$\omega_o = 2\pi f$$

$$10^3 = \omega_o = 2\pi f$$

$$f = 159.2\ Hz$$

Note that the integration filter produces a single-pole response from 300 to 3 kHz. The current required to move the integrator output a specific voltage from zero is simply:

$$I_i = \frac{V_o}{R_1} + \left(C_1 \times \frac{dV_o}{dt}\right)$$

Now a 0 dBmo sine wave has a peak value of 1.0954 volts. In 1/8 of a cycle of a sine wave centered around the zero crossing, the sine wave changes by approximately its peak value. The CVSD step should trace that change. The required current for a 0 dBm 1 kHz sine wave is:

$$I_i = \frac{1.1\ V}{^*2(10\ k\Omega)} + \frac{0.1\ \mu F(1.1)}{0.125\ ms} = 0.935\ mA$$

*The maximum voltage across $R_1$ when maximum slew is required is:

$$\frac{1.1\ V}{2}$$

Now the voltage range of the syllabic filter is the power supply voltage, thus:

$$R_X = 0.25(V_{CC})\ \frac{1}{0.935\ mA}$$

A similar procedure can be followed to establish the proper gain for any input level and integration filter type.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## CVSD DESIGN CONSIDERATIONS (continued)

### Minimum Step Size

The final parameter to be selected for the simple codec in Figure 14 is idle channel step size. With no input signal, the digital output becomes a one-zero alternating pattern and the analog output becomes a small triangle wave. Mismatches of internal currents and offsets limit the minimum step size which will produce a perfect idle channel pattern. The MC3417 is tested to ensure that a 20 mVp-p minimum step size at 16 kHz will attain a proper idle channel. The idle channel step size must be twice the specified total loop offset if a one-zero idle pattern is desired. In some applications a much smaller minimum step size (e.g., 0.1 mV) can produce quiet performance without providing a 1-0 pattern.

To set the idle channel step size, the value of $R_{min}$ must be selected. With no input signal, the slope control algorithm is inactive. A long series of ones or zeros never occurs. Thus, the voltage across the syllabic filter capacitor ($C_S$) would decay to zero. However, the voltage divider of $R_S$ and $R_{min}$ (see Figure 14) sets the minimum allowed voltage across the syllabic filter capacitor. That voltage must produce the desired ramps at the analog output. Again we write the filter input current equation:

$$I_i = \frac{V_o}{R_1} + C\frac{dV_o}{dt}$$

For values of $V_o$ near $V_{CC}/2$ the $V_o/R$ term is negligible; thus

$$I_i = C_S\frac{\Delta V_o}{\Delta T}$$

where $\Delta T$ is the clock period and $\Delta V_o$ is the desired peak-to-peak value of the idle output. For a 16K-bit system using the circuit in Figure 14

$$I_i = \frac{0.1\,\mu F \;\; 20\,mV}{62.5\,\mu s} = 33\,\mu A$$

The voltage on $C_S$ which produces a 33 $\mu$A current is determined by the value of $R_x$.

$$I_i R_x = V_{Smin};\ \text{for } 33\,\mu A,\ V_{Smin} = 41.6\,mV$$

In Figure 14 $R_S$ is 18 kΩ. That selection is discussed with the syllabic filter considerations. The voltage divider of $R_S$ and $R_{min}$ must produce an output of 41.6 mV.

$$V_{CC}\frac{R_S}{R_S + R_{min}} = V_{Smin} \qquad R_{min} \simeq 2.4\,M\Omega$$

Having established these four parameters — clock rate, number of shift register bits, loop gain and minimum step size — the encoder circuit in Figure 14 will function at near optimum performance for input levels around 0 dBm.

## INCREASING CVSD PERFORMANCE

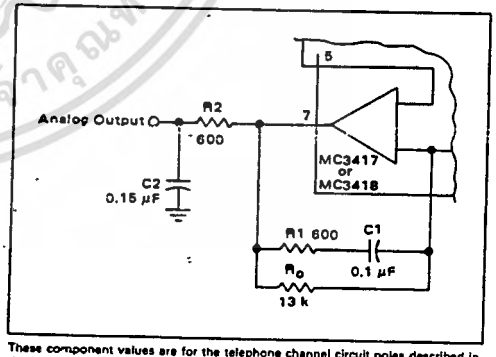
### Integration Filter Design

The circuit in Figure 14 uses a single-pole integration network formed with a 0.1 $\mu$F capacitor and a 10 kΩ resistor. It is possible to improve the performance of the circuit in Figure 14 by 1 or 2 dB by using a two-pole integration network. The improved circuit is shown.

The first pole is still placed below 300 Hz to provide the 1/S voice content curve and a second pole is placed somewhere above the 1 kHz frequency. For telephony circuits, the second pole can be placed above 1.8 kHz to exceed the 1633 touchtone frequency. In other communication systems, values as low as 1 kHz may be selected. In general, the lower in frequency the second pole is placed, the greater the noise improvement. Then, to ensure the encoder loop stability, a zero is added to keep the phase shift less than 180°. This zero should be placed slightly above the low-pass output filter break frequency so as not to reduce the effectiveness of the second pole. A network of 235 Hz, 2 kHz and 5.2 kHz is typical for telephone applications while 160 Hz, 1.2 kHz and 2.8 kHz might be used in voice only channels. (Voice only channels can use an output low-pass filter which breaks at about 2.5 kHz.) The two-pole network in Figure 16 has a transfer function of:

$$\frac{V_o}{I_i} = \frac{R_0R_1\left(S + \frac{1}{R_1C_1}\right)}{R_2C_2(R_0 + R_1)\left(S + \frac{1}{(R_0 + R_1)C_1}\right)S + \left(\frac{1}{R_2C_2}\right)}$$

FIGURE 16 — IMPROVED FILTER CONFIGURATION




These component values are for the telephone channel circuit poles described in the text. The R2, C2 product can be provided with different values of R and C. R2 should be chosen to be equal to the termination resistor on Pin 1.

## INCREASING CVSD PERFORMANCE (continued)

Thus the two poles and the zero can be selected arbitrarily as long as the zero is at a higher frequency than the first pole. The values in Figure 16 represent one implementation of the telephony filter requirement.

The selection of the two-pole filter network effects the selection of the loop gain value and the minimum step size resistor. The required integrator current for a given change in voltage now becomes:

$$I_i = \frac{V_o}{R_0} + \left(\frac{R_2C_2}{R_0} + \frac{R_1C_1}{R_0} + C_1\right)\frac{\Delta V_o}{\Delta T} +$$

$$\left(R_2C_2C_1 + \frac{R_1C_1R_2C_2}{R_0}\right)\frac{\Delta V_o 2}{\Delta T2}$$

The calculation of desired gain resistor $R_X$ then proceeds exactly as previously described.

### Syllabic Filter Design

The syllabic filter in Figure 14 is a simple single-pole network of 18 kΩ and 0.33 μF. This produces a 6.0 ms time constant for the averaging of the coincidence output signal. The voltage across the capacitor determines the integrator current which in turn establishes the step size. The integrator current and the resulting step size determine the companding ratio and the S/N performance. The companding ratio is defined as the voltage across $C_S/V_{CC}$.

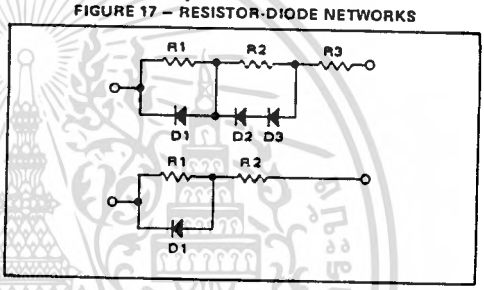
The S/N performance may be improved by modifying the voltage to current transformation produced by $R_X$. If different portions of the total $R_X$ are shunted by diodes, the integrator current can be other than $(V_{CC}-V_S)/R_X$. These breakpoint curves must be designed experimentally for the particular system application. In general, one would wish that the current would double with input level. To design the desired curve, supply current to Pin 4 of the codec from an external source. Input a signal level and adjust the current until the S/N performance is optimum. Then record the syllabic filter voltage and the current. Repeat this for all desired signal levels. Then derive the resistor diode network which produces that curve on a curve tracer.

Once the network is designed with the curve tracer, it is then inserted in place of $R_X$ in the circuit and the forced optimum noise performance will be achieved from the active syllabic algorithm.

Diode breakpoint networks may be very simple or moderately complex and can improve the usable dynamic range of any codec. In the past they have been used in high performance telephone codecs.

Typical resistor-diode networks are shown in Figure 17.

FIGURE 17 – RESISTOR-DIODE NETWORKS




If the performance of more complex diode networks is desired, the circuit in Figure 18 should be used. It simulates the companding characteristics of nonlinear $R_X$ elements in a different manner.

### Output Low Pass Filter

A low pass filter is required at the receiving circuit output to eliminate quantizing noise. In general, the lower the bit rate, the better the filter must be. The filter in Figure 20 provides excellent performance for 12 kHz to 40 kHz systems.

## TELEPHONE CARRIER QUALITY CODEC USING MC3418

Two specifications of the integrated circuit are specifically intended to meet the performance requirements of commercial telephone systems. First, slope polarity switch current matching is laser trimmed to guarantee proper idle channel performance with 5 mV minimum step size and a typical 1% current match from 15 μA to 3 mA. Thus a 300 to 1 range of step size variation is possible. Second, the MC3418 provides the four-bit algorithm currently used in subscriber loop telephone systems. With these specifications and the circuit of Figure 18, a telephone quality codec can be mass produced.

The circuit in Figure 18 provides a 30 dB S/Nc ratio over 50 dB of dynamic range for a 1 kHz test tone at a 37.7K bit rate. At 37.7K bits, 40 voice channels may be multiplexed on a standard 1.544 megabit T1 facility. This codec has also been tested for $10^{-7}$ error rates with asynchronous and synchronous data up to 2400 baud and for reliable performance with DTMF signaling. Thus, the design is applicable in telephone quality subscriber loop carrier systems, subscriber loop concentrators and small PABX installations.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## TELEPHONE CARRIER QUALITY CODEC USING MC3418 (continued)

### The Active Companding Network

The unique feature of the codec in Figure 18 is the step size control circuit which uses a companding ratio reference, the present step size, and the present syllabic filter output to establish the optimum companding ratios and step sizes for any given input level. The companding ratio of a CVSD codec is defined as the duty cycle of the coincidence output. It is the parameter measured by the syllabic filter and is the voltage across $C_S$ divided by the voltage swing of the coincidence output. In Figure 18, the voltage swing of Pin 11 is 6.0 volts. The operating companding ratio is analoged by the voltage between Pins 10 and 4 by means of the virtual short across Pins 3 and 4 of the V to I op amp within the integrated circuit. Thus, the instantaneous companding ratio of the codec is always available at the negative input of A1.

The diode D1 and the gain of A1 and A2 provide a companding ratio reference for any input level. If the output of A2 is more than 0.7 volts below $V_{CC}/2$, then the positive input of A1 is ($V_{CC}/2$ − 0.7). The on diode drop at the input of A1 represents a 12% companding ratio (12% = 0.7 V/6.0 V).

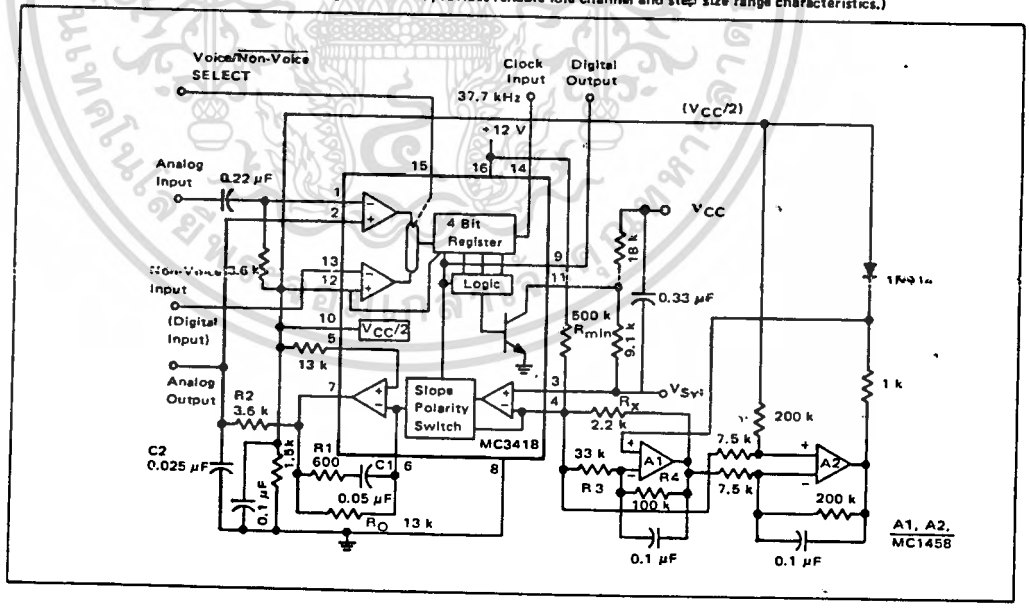
The present step size of the operating codec is directly related to the voltage across $R_X$, which established the integrator current. In Figure 18, the voltage across $R_X$ is amplified by the differential amplifier A2 whose output is single ended with respect to Pin 10 of the IC.

For large signal inputs, the step size is large and the output of A2 is lower than 0.7 volts. Thus D1 is fully on. The present step size is not a factor in the step size control. However, the difference between 12% companding ratio and the instantaneous companding ratio at Pin 4 is amplified by A1. The output of A1 changes the voltage across $R_X$ in a direction which reduces the difference between the companding reference and the operating ratio by changing the step size. The ratio of R4 and R3 determines how closely the voltage at Pin 4 will be forced to 12%. The selection of R3 and R4 is initially experimental. However, the resulting companding control is dependent on $R_X$, R3, R4, and the full diode drop D1. These values are easy to reproduce from codec to codec.

For small input levels, the companding ratio reference becomes the output of A2 rather than the diode drop. The operating companding ratio on Pin 4 is then compared to a companding ratio smaller than 12% which is determined by the voltage drop across $R_X$ and the gain of A2 and A1. The gain of A2 is also experimentally determined, but once determined, the circuitry is easily

FIGURE 18 — TELEPHONE QUALITY DELTAMOD CODER

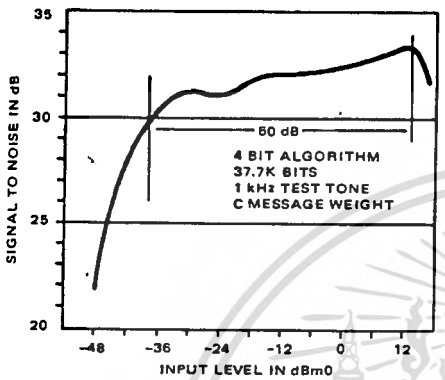
(Both double integration and active companding control are used to obtain improved CVSD performance. Laser trimming of the integrated circuit provides reliable idle channel and step size range characteristics.)



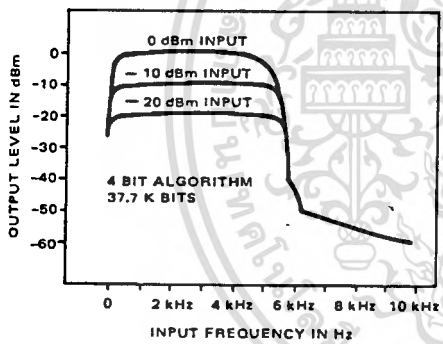
TELEPHONE CARRIER QUALITY CODEC USING MC3418 (continued)

FIGURE 19 – SIGNAL-TO-NOISE PERFORMANCE AND FREQUENCY RESPONSE
(Showing the improvement realized with the circuit in Figure 18.)

a. SIGNAL-TO-NOISE PERFORMANCE OF TELEPHONY QUALITY DELTAMODULATOR




b. FREQUENCY RESPONSE versus INPUT LEVEL (SLOPE OVERLOAD CHARACTERISTIC)




repeated.

With no input signal, the companding ratio at Pin 4 goes to zero and the voltage across $R_X$ goes to zero. The voltage at the output of A2 becomes zero since there is no drop across $R_X$. With no signal input, the actively controlled step size vanished.

The minimum step size is established by the 500 k resistor between $V_{CC}$ and $V_{CC}/2$ and is therefore independently selectable.

The signal to noise results of the active companding network are shown in Figure 19. A smooth 2 dB drop is realized from +12 dBm to –24 under the control of A1. At –24 dBm, A2 begins to degenerate the companding reference and the resulting step size is reduced so as to extend the dynamic range of the codec by 20 dBm.

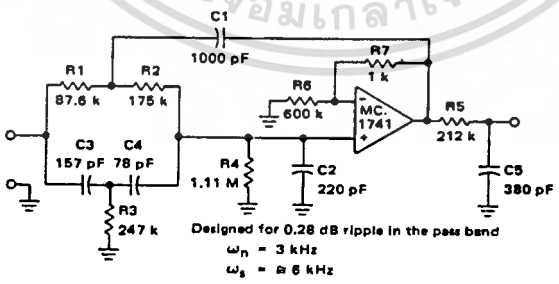
The slope overload characteristic is also shown. The active companding network produces improved performance with frequency. The 0 dBm slope overload point is raised to 4.8 kHz because of the gain available in controlling the voltage across $R_X$. The curves demonstrate that the level linearity has been maintained or improved.*

The codec in Figure 18 is designed specifically for 37.7K bit systems. However, the benefits of the active companding network are not limited to high bit rate systems. By modifying the crossover region (changing the gain of A2), the active technique may be used to improve the performance of lower bit rate systems.

The performance and repeatability of the codec in Figure 18 represents a significant step forward in the art and cost of CVSD codec designs.

*A larger value for C2 is required in the decoder circuit than in the encoder to adjust the level linearity with frequency. In Figure 18, 0.050 μF would work well.

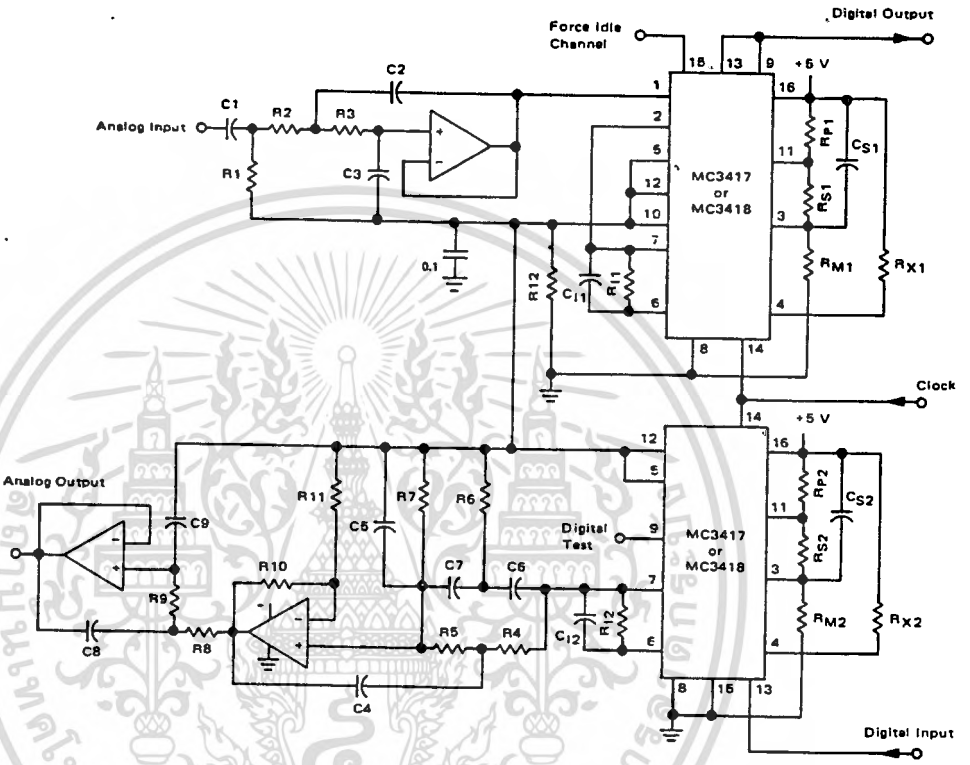
FIGURE 20 – HIGH PERFORMANCE ELLIPTIC FILTER FOR CVSD OUTPUT






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 21 – FULL DUPLEX/32K BIT CVSD VOICE CODEC USING MC3517/18 AND MC3503/6 OP AMP




**Codec Components**

$R_{X1}$, $R_{X2}$ – 3.3 kΩ
$R_{P1}$, $R_{P2}$ – 3.3 kΩ
$R_{S1}$, $R_{S2}$ – 100 kΩ
$R_{I1}$, $R_{I2}$ – 20 kΩ
$R_{12}$ – 1 kΩ
$R_{M1}$, $R_{M2}$ – 5 MΩ (MC3417)
Minimum step size = 20 mV
$R_{M1}$, $R_{M2}$ – 15 MΩ (MC3418)
Minimum step size = 6 mV

$C_{S1}$, $C_{S2}$ – 0.05 μF
$C_{I1}$, $C_{I2}$ – 0.05 μF

2 MC3417 (or MC3418)
1 MC3403 (or MC3406)

Note: All Res. 5%
All Cap. 5%

**Input Filter Specifications**

12 dB/Octave Rolloff above 3.3 kHz
6 dB/Octave Rolloff below 50 Hz

**Output Filter Specifications**

Break Frequency – 3.3 kHz
Stop Band – 9 kHz
Stop Band Atten. – 50 dB
Rolloff – > 40 dB/Octave

**Filter Components**

| | |
|---|---|
| R1 – 965 Ω | C1 – 3.3 μF |
| R2 – 72 kΩ | C2 – 837 pF |
| R3 – 72 kΩ | C3 – 536 pF |
| R4 – 63.46 kΩ | C4 – 1000 pF |
| R5 – 127 kΩ | C5 – 222 pF |
| R6 – 365.6 kΩ | C6 – 77 pF |
| R7 – 1.645 MΩ | C7 – 38 pF |
| R8 – 72 kΩ | C8 – 837 pF |
| R9 – 72 kΩ | C9 – 536 pF |
| R10 – 29.5 kΩ | |
| R11 – 72 kΩ | |

Note: All Res. 0.1% to 1%.
All Cap. 1.0%

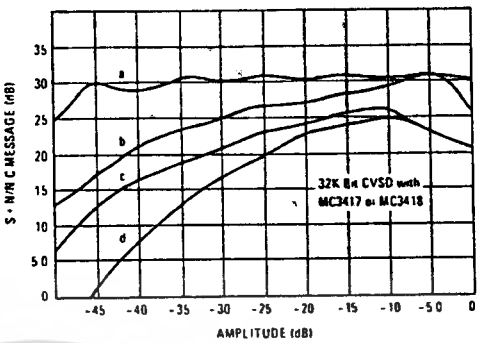
## COMPARATIVE CODEC PERFORMANCE

The salient feature of CVSD codecs using the MC3517 and MC3518 family is versatility. The range of codec complexity tradeoffs and bit rate is so wide that one cannot grasp the interdependency of parameters for voice applications in a few pages.

Design of a specific codec must be tailored to the digital channel bandwidth, the analog bandwidth, the quality of signal transmission required and the cost objectives. To illustrate the choices available, the data in Figure 22 compares the signal-to-noise ratios and dynamic range of various codec design options at 32K bits. Generally, the relative merits of each design feature will remain intact in any application. Lowering the bit rate will reduce the dynamic range and noise performance of all techniques. As the bit rate is increased, the overall performance of each technique will improve and the need for more complex designs diminishes.

Non-voice applications of the MC3517 and MC3518 are also possible. In those cases, the signal bandwidth and amplitude characteristics must be defined before the specification of codec parameters can begin. However, in general, the design can proceed along the lines of the voice applications shown here, taking into account the different signal bandwidth requirements.

FIGURE 22 – COMPARATIVE CODEC PERFORMANCE – SIGNAL-TO-NOISE RATIO FOR 1 kHz TEST TONE




These curves demonstrate the improved performance obtained with several codec designs of varying complexity.

Curve a — Complex companding and double integration (Figure 18 — MC3418)
Curve b — Double integration (Figure 14 using Figure 16 — MC3418)
Curve c — Single integration (Figure 14 — MC3418) with 6.0 mV step size
Curve d — Single integretion (Figure 14 — MC3417) with 25 mV step size



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

# บรรณานุกรม


1. ประกิจ ตังติสานนท์ ,วิศวกรรมสื่อสาร, คณะวิศวกรรมศาสตร์ สจล. .
2. ถวิล กิ่งทอง ,เทคโนโลยี การส่งสัญญาณดิจิตอล, คณะวิศวกรรมศาสตร์ สจล.
3. ประเสริฐโรจน์สุธีวัฒน์,เครื่องบันทึกเสียงพูดระบบดิจิตอล,เซมิคอนดัคเตอร์ อิเลคทรอนิคส์ ฉบับที่ 84 ก.พ.-มี.ค.
4. TEXAS INSTRUMENTS INC., The TTL Data Book for Design Engineering ,Second Edition
5. บริษัท ซีเอ็ดยูเคชั่น ,คู่มือไอซี CMOS ,2534
6. ธานินทร์ ถาวรศาสนวงศ์ ,ทินกร ดุ๊ก ,การอินเตอร์เฟส IBM PC
7. Motorola,Telecommunication Device Data,Series C ,1989
8. ณัฐพล มงคลประเสริฐ ,หัสรังษี ศิริวิมลวรรณ,อนุรักษ์ ตัณฑ์พูนเกียรติ,เครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ,วิทยานิพนธ์ คณะวิศวกรรมศาสตร์ สจล. ,2533
9. ธันวา ศรีประโมง ,การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม ,พิมพ์ครั้งที่ 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้